

**16****M16C/26A** Group

(M16C/26A, M16C/26B, M16C/26T)

Hardware Manual

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER

M16C FAMILY / M16C/Tiny SERIES

Hardware Manual

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

Rev. 2.00  
Revision Date: Feb.15, 2007

RenesasTechnology  
[www.renesas.com](http://www.renesas.com)

## Notes regarding these materials

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

## General Precautions in the Handling of MPU/MCU Products

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

### 1. Handling of Unused Pins

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

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

### 2. Processing at Power-on

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

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.  
In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.  
In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

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

### 4. Clock Signals

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

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

### 5. Differences between Products

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

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

# How to Use This Manual

## 1. Purpose and Target Readers

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

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

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

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

The following documents apply to the M16C/26A Group (M16C/26A, M16C/26B, and M16C/26T). Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

Document Type	Description	Document Title	Document No.
Hardware manual	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) Hardware Manual	This hardware manual
Software manual	Description of CPU instruction set	M16C/60, M16C/20, M16C/Tiny Series Software Manual	REJ09B0137
Application note	Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C	Available from Renesas Technology Web site.	
Renesas technical update	Product specifications, updates on documents, etc.		

## 2. Notation of Numbers and Symbols

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

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

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

Examples the PM03 bit in the PM0 register  
P3\_5 pin, VCC pin

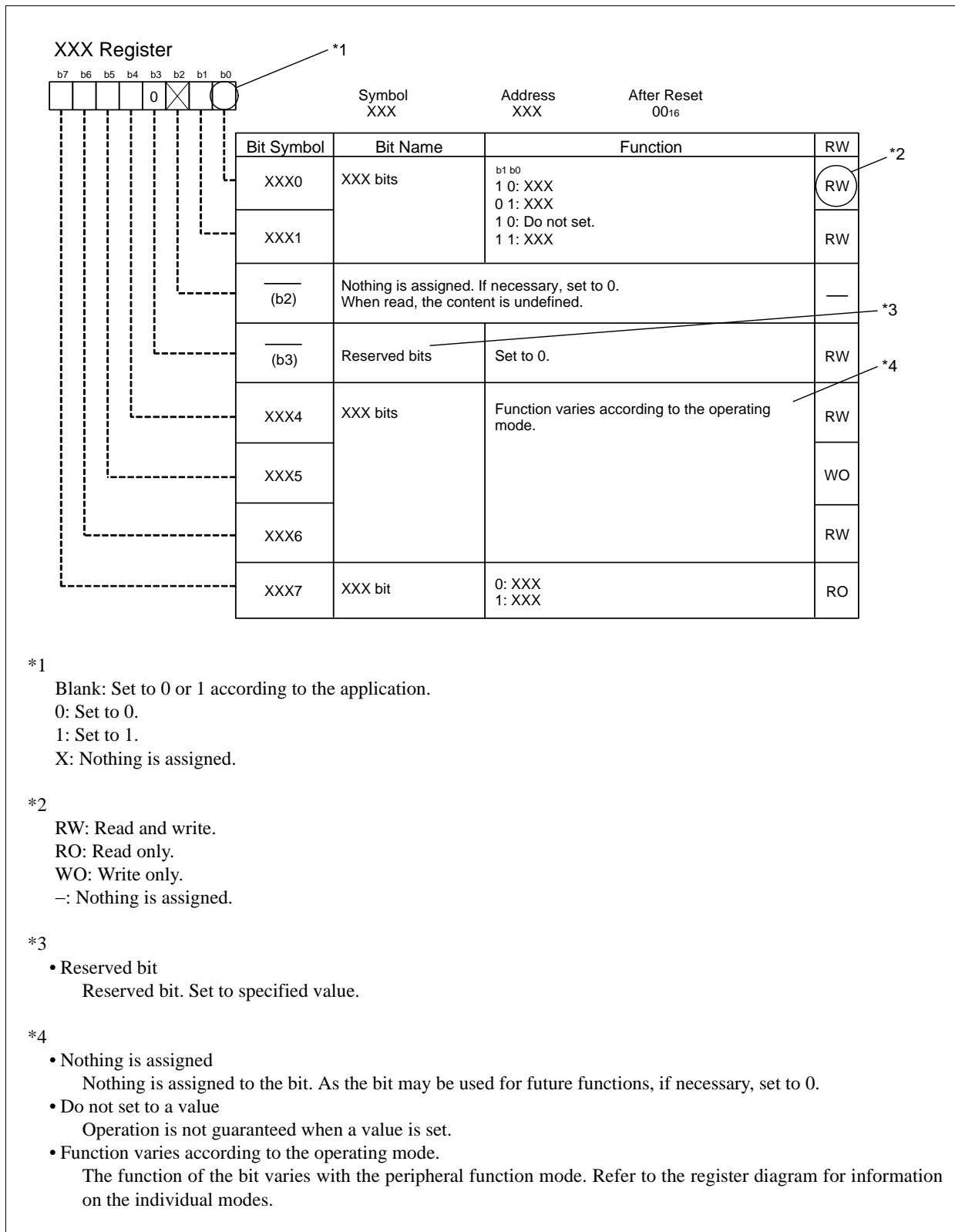
(2) Notation of Numbers

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

Examples Binary: 11<sub>2</sub>  
Hexadecimal: EFA0<sub>16</sub>  
Decimal: 1234

### 3. Register Notation

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



#### 4. List of Abbreviations and Acronyms

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

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

# Table of Contents

Quick Reference by Address _____	B-1
<b>1. Overview _____</b>	<b>1</b>
1.1 Applications .....	1
1.2 Performance Outline .....	2
1.3 Block Diagram .....	4
1.4 Product List .....	6
1.5 Pin Assignments .....	11
1.6 Pin Description .....	15
<b>2. Central Processing Unit (CPU) _____</b>	<b>17</b>
2.1 Data Registers (R0, R1, R2 and R3) .....	17
2.2 Address Registers (A0 and A1) .....	17
2.3 Frame Base Register (FB) .....	18
2.4 Interrupt Table Register (INTB) .....	18
2.5 Program Counter (PC) .....	18
2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) .....	18
2.7 Static Base Register (SB) .....	18
2.8 Flag Register (FLG) .....	18
2.8.1 Carry Flag (C Flag) .....	18
2.8.2 Debug Flag (D Flag) .....	18
2.8.3 Zero Flag (Z Flag) .....	18
2.8.4 Sign Flag (S Flag) .....	18
2.8.5 Register Bank Select Flag (B Flag) .....	18
2.8.6 Overflow Flag (O Flag) .....	18
2.8.7 Interrupt Enable Flag (I Flag) .....	18
2.8.8 Stack Pointer Select Flag (U Flag) .....	18
2.8.9 Processor Interrupt Priority Level (IPL) .....	18
2.8.10 Reserved Area .....	18
<b>3. Memory _____</b>	<b>19</b>
<b>4. Special Function Registers (SFRs) _____</b>	<b>20</b>



<b>5. Reset</b>	<b>26</b>
5.1 Hardware Reset	26
5.1.1 Hardware Reset 1	26
5.1.2 Hardware Reset 2	26
5.2 Software Reset	27
5.3 Watchdog Timer Reset	27
5.4 Oscillation Stop Detection Reset	27
5.5 Voltage Detection Circuit	29
5.5.1 Voltage Down Detection Interrupt	32
5.5.2 Limitations on Exiting Stop Mode	34
5.5.3 Limitations on Exiting Wait Mode	34
<b>6. Processor Mode</b>	<b>35</b>
<b>7. Clock Generation Circuit</b>	<b>38</b>
7.1 Main Clock	45
7.2 Sub Clock	46
7.3 On-chip Oscillator Clock	47
7.4 PLL Clock	47
7.5 CPU Clock and Peripheral Function Clock	49
7.5.1 CPU Clock	49
7.5.2 Peripheral Function Clock (f1, f2, f8, f32, f1SIO, f2SIO, f8SIO, f32SIO, fAD, fc32)	49
7.5.3 ClockOutput Function	49
7.6 Power Control	50
7.6.1 Normal Operation Mode	50
7.6.2 Wait Mode	51
7.6.3 Stop Mode	53
7.7 System Clock Protective Function	57
7.8 Oscillation Stop and Re-oscillation Detect Function	57
7.8.1 Operation When the CM27 bit is set to "0" (Oscillation Stop Detection Reset) ...	58
7.8.2 Operation When the CM27 bit is set to "1"	
(Oscillation Stop and Re-oscillation Detect Interrupt)	58
7.8.3 How to Use Oscillation Stop and Re-oscillation Detect Function	59
<b>8. Protection</b>	<b>60</b>
<b>9. Interrupt</b>	<b>61</b>
9.1 Type of Interrupts	61
9.1.1 Software Interrupts	62
9.1.2 Hardware Interrupts	63

9.2	Interrupts and Interrupt Vector .....	64
9.2.1	Fixed Vector Tables .....	64
9.2.2	Relocatable Vector Tables .....	65
9.3	Interrupt Control .....	66
9.3.1	I Flag .....	69
9.3.2	IR Bit .....	69
9.3.3	ILVL2 to ILVL0 Bits and IPL .....	69
9.4	Interrupt Sequence .....	70
9.4.1	Interrupt Response Time .....	71
9.4.2	Variation of IPL when Interrupt Request is Accepted .....	71
9.4.3	Saving Registers .....	72
9.4.4	Returning from an Interrupt Routine .....	74
9.5	Interrupt Priority .....	74
9.5.1	Interrupt Priority Resolution Circuit .....	74
9.6	$\overline{\text{INT}}$ Interrupt .....	76
9.7	$\overline{\text{NMI}}$ Interrupt .....	77
9.8	Key Input Interrupt .....	77
9.9	Address Match Interrupt .....	78
<b>10.</b>	<b>Watchdog Timer</b> .....	<b>80</b>
10.1	Count Source Protective Mode .....	81
<b>11.</b>	<b>DMAC</b> .....	<b>82</b>
11.1	Transfer Cycles .....	87
11.2	DMA Transfer Cycles .....	89
11.3	DMA Enable .....	90
11.4	DMA Request .....	90
11.5	Channel Priority and DMA Transfer Timing .....	91
<b>12.</b>	<b>Timer</b> .....	<b>92</b>
12.1	Timer A .....	94
12.1.1	Timer Mode .....	97
12.1.2	Event Counter Mode .....	98
12.1.3	One-shot Timer Mode .....	103
12.1.4	Pulse Width Modulation (PWM) Mode .....	105
12.2	Timer B .....	108
12.2.1	Timer Mode .....	111
12.2.2	Event Counter Mode .....	112
12.2.3	Pulse Period and Pulse Width Measurement Mode .....	113
12.2.4	A/D Trigger Mode .....	115

12.3 Three-phase Motor Control Timer Function .....	117
12.3.1 Position-data-retain Function .....	128
12.3.2 Three-phase/Port Output Switch Function .....	130
<b>13. Serial I/O .....</b>	<b>132</b>
13.1. UARTi (i=0 to 2) .....	132
13.1.1. Clock Synchronous serial I/O Mode .....	142
13.1.2. Clock Asynchronous Serial I/O (UART) Mode .....	150
13.1.3 Special Mode 1 (I <sup>2</sup> C bus mode)(UART2) .....	158
13.1.4 Special Mode 2 (UART2) .....	168
13.1.5 Special Mode 3 (IE Bus mode )(UART2) .....	173
13.1.6 Special Mode 4 (SIM Mode) (UART2) .....	175
<b>14. A/D Converter .....</b>	<b>180</b>
14.1 Operation Modes .....	186
14.1.1 One-Shot Mode .....	186
14.1.2 Repeat mode .....	188
14.1.3 Single Sweep Mode .....	190
14.1.4 Repeat Sweep Mode 0 .....	192
14.1.5 Repeat Sweep Mode 1 .....	194
14.1.6 Simultaneous Sample Sweep Mode .....	196
14.1.7 Delayed Trigger Mode 0 .....	199
14.1.8 Delayed Trigger Mode 1 .....	205
14.2 Resolution Select Function .....	211
14.3 Sample and Hold .....	211
14.4 Power Consumption Reducing Function .....	211
14.5 Output Impedance of Sensor under A/D Conversion .....	212
<b>15. CRC Calculation Circuit .....</b>	<b>213</b>
15.1. CRC Snoop .....	213
<b>16. Programmable I/O Ports .....</b>	<b>216</b>
16.1 Port Pi Direction Register (PDi Register, i = 1, 6 to 10) .....	216
16.2 Port Pi Register (Pi Register, i = 1, 6 to 10) .....	216
16.3 Pull-up Control Register 0 to Pull-up Control Register 2 (PUR0 to PUR2 Registers) ...	216
16.4 Port Control Register .....	217
16.5 Pin Assignment Control register (PACR) .....	217
16.6 Digital Debounce function .....	217
<b>17. Flash Memory Version .....</b>	<b>230</b>
17.1 Flash Memory Performance .....	230
17.1.1 Boot Mode .....	231

17.2 Memory Map .....	232
17.3 Functions To Prevent Flash Memory from Rewriting.....	235
17.3.1 ROM Code Protect Function .....	235
17.3.2 ID Code Check Function .....	235
17.4 CPU Rewrite Mode .....	237
17.4.1 EW0 Mode .....	238
17.4.2 EW1 Mode .....	238
17.5 Register Description .....	239
17.5.1 Flash memory control register 0 (FMR0) .....	239
17.5.2 Flash memory control register 1 (FMR1) .....	240
17.5.3 Flash memory control register 4 (FMR4) .....	240
17.6 Precautions in CPU Rewrite Mode .....	245
17.6.1 Operation Speed .....	245
17.6.2 Prohibited Instructions .....	245
17.6.3 Interrupts .....	245
17.6.4 How to Access .....	245
17.6.5 Writing in the User ROM Space .....	245
17.6.6 DMA Transfer .....	246
17.6.7 Writing Command and Data .....	246
17.6.8 Wait Mode .....	246
17.6.9 Stop Mode .....	246
17.6.10 Low Power Consumption Mode and On-chip Oscillator-Low Power Consumption Mode ...	246
17.7 Software Commands .....	247
17.7.1 Read Array Command (FF16) .....	247
17.7.2 Read Status Register Command (7016) .....	247
17.7.3 Clear Status Register Command (5016) .....	248
17.7.4 Program Command (4016) .....	248
17.7.5 Block Erase .....	249
17.8 Status Register .....	251
17.8.1 Sequence Status (SR7 and FMR00 Bits ) .....	251
17.8.2 Erase Status (SR5 and FMR07 Bits) .....	251
17.8.3 Program Status (SR4 and FMR06 Bits) .....	251
17.8.4 Full Status Check .....	252
17.9 Standard Serial I/O Mode .....	254
17.9.1 ID Code Check Function .....	254
17.9.2 Example of Circuit Application in Standard Serial I/O Mode .....	258
17.10 Parallel I/O Mode .....	260
17.10.1 ROM Code Protect Function .....	260

**18. Electrical Characteristics** \_\_\_\_\_ **261**

- 18.1. M16C/26A, M16C/26B (Normal version) ..... 261
- 18.2. M16C/26T (T version) ..... 280

**19. Usage Notes** \_\_\_\_\_ **299**

- 19.1 SFR ..... 299
  - 19.1.1 Precaution for 48-pin package ..... 299
  - 19.1.2 Precaution for 42-pin package ..... 299
  - 19.1.3 Register Setting ..... 299
- 19.2 PLL Frequency Synthesizer ..... 300
- 19.3 Power Control ..... 301
- 19.4 Protect ..... 303
- 19.5 Interrupts ..... 304
  - 19.5.1 Reading address 0000016 ..... 304
  - 19.5.2 Setting the SP ..... 304
  - 19.5.3 The  $\overline{\text{NMI}}$  Interrupt ..... 304
  - 19.5.4 Changing the Interrupt Generation Factor ..... 304
  - 19.5.5  $\overline{\text{INT}}$  Interrupt ..... 305
  - 19.5.6 Rewrite the Interrupt Control Register ..... 306
  - 19.5.7 Watchdog Timer Interrupt ..... 306
- 19.6 DMAC ..... 307
  - 19.6.1 Write to DMAE Bit in DMiCON Register ..... 307
- 19.7 Timer ..... 308
  - 19.7.1 Timer A ..... 308
  - 19.7.2 Timer B ..... 311
  - 19.7.3 Three-phase Motor Control Timer Function ..... 312
- 19.8 Serial I/O ..... 313
  - 19.8.1 Clock-Synchronous Serial I/O ..... 313
  - 19.8.2 Serial I/O (UART Mode) ..... 314
- 19.9 A/D Converter ..... 315
- 19.10 Programmable I/O Ports ..... 317
- 19.11 Electric Characteristic Differences Between Mask ROM  
and Flash Memory Version Microcomputers ..... 318
- 19.12 Mask ROM Version ..... 319
  - 19.12.1 Internal ROM area ..... 319
  - 19.12.2 Reserve bit ..... 319

19.13 Flash Memory Version .....	320
19.13.1 Functions to Inhibit Rewriting Flash Memory .....	320
19.13.2 Stop mode .....	320
19.13.3 Wait mode .....	320
19.13.4 Low power dissipation mode, on-chip oscillator low power dissipation mode ...	320
19.13.5 Writing command and data .....	320
19.13.6 Program Command .....	320
19.13.7 Operation speed .....	320
19.13.8 Instructions prohibited in EW0 Mode .....	320
19.13.9 Interrupts .....	321
19.13.10 How to access .....	321
19.13.11 Writing in the user ROM area .....	321
19.13.12 DMA transfer .....	321
19.13.13 Regarding Programming/Erase Times and Execution Time .....	321
19.13.14 Definition of Programming/Erase Times .....	322
19.13.15 Flash Memory Version Electrical Characteristics 10,000 E/W cycle product ..	322
19.13.16 Boot Mode .....	322
19.14 Noise .....	323
19.15 Instruction for a Device Use .....	324
<b>Appendix 1. Package Dimensions _____</b>	<b>325</b>
<b>Appendix 2. Functional Difference _____</b>	<b>326</b>
Appendix 2.1 Differences between M16C/26A, M16C/26B, and M16C/26T .....	326
Appendix 2.2 Differences between M16C/26A Group and M16C/26 Group .....	327
<b>Register Index _____</b>	<b>328</b>

## Quick Reference by Address

Address	Register	Symbol	Page	Address	Register	Symbol	Page
0000 <sub>16</sub>				0040 <sub>16</sub>			
0001 <sub>16</sub>				0041 <sub>16</sub>			
0002 <sub>16</sub>				0042 <sub>16</sub>			
0003 <sub>16</sub>				0043 <sub>16</sub>			
0004 <sub>16</sub>	Processor mode register 0	PM0	35	0044 <sub>16</sub>	INT3 interrupt control register	INT3IC	67
0005 <sub>16</sub>	Processor mode register 1	PM1	35	0045 <sub>16</sub>			
0006 <sub>16</sub>	System clock control register 0	CM0	40	0046 <sub>16</sub>			
0007 <sub>16</sub>	System clock control register 1	CM1	41	0047 <sub>16</sub>			
0008 <sub>16</sub>				0048 <sub>16</sub>	INT5 interrupt control register	INT5IC	67
0009 <sub>16</sub>	Address match interrupt enable register	AIER	79	0049 <sub>16</sub>	INT4 interrupt control register	INT4IC	67
000A <sub>16</sub>	Protect register	PRCR	60	004A <sub>16</sub>	UART2 Bus collision detection interrupt control register	BCNIC	67
000B <sub>16</sub>				004B <sub>16</sub>	DMA0 interrupt control register	DM0IC	67
000C <sub>16</sub>	Oscillation stop detection register	CM2	42	004C <sub>16</sub>	DMA1 interrupt control register	DM1IC	67
000D <sub>16</sub>				004D <sub>16</sub>	Key input interrupt control register	KUPIC	67
000E <sub>16</sub>	Watchdog timer start register	WDTS	81	004E <sub>16</sub>	A/D conversion interrupt control register	ADIC	67
000F <sub>16</sub>	Watchdog timer control register	WDC	81	004F <sub>16</sub>	UART2 transmit interrupt control register	S2TIC	67
0010 <sub>16</sub>				0050 <sub>16</sub>	UART2 receive interrupt control register	S2RIC	67
0011 <sub>16</sub>	Address match interrupt register 0	RMAD0	79	0051 <sub>16</sub>	UART0 transmit interrupt control register	S0TIC	67
0012 <sub>16</sub>				0052 <sub>16</sub>	UART0 receive interrupt control register	S0RIC	67
0013 <sub>16</sub>				0053 <sub>16</sub>	UART1 transmit interrupt control register	S1TIC	67
0014 <sub>16</sub>	Address match interrupt register 1	RMAD1	79	0054 <sub>16</sub>	UART1 receive interrupt control register	S1RIC	67
0015 <sub>16</sub>				0055 <sub>16</sub>	Timer A0 interrupt control register	TA0IC	67
0016 <sub>16</sub>				0056 <sub>16</sub>	Timer A1 interrupt control register	TA1IC	67
0017 <sub>16</sub>				0057 <sub>16</sub>	Timer A2 interrupt control register	TA2IC	67
0018 <sub>16</sub>				0058 <sub>16</sub>	Timer A3 interrupt control register	TA3IC	67
0019 <sub>16</sub>	Voltage detection register 1	VCR1	30	0059 <sub>16</sub>	Timer A4 interrupt control register	TA4IC	67
001A <sub>16</sub>	Voltage detection register 2	VCR2	30	005A <sub>16</sub>	Timer B0 interrupt control register	TB0IC	67
001B <sub>16</sub>				005B <sub>16</sub>	Timer B1 interrupt control register	TB1IC	67
001C <sub>16</sub>	PLL control register 0	PLC0	44	005C <sub>16</sub>	Timer B2 interrupt control register	TB2IC	67
001D <sub>16</sub>				005D <sub>16</sub>	INT0 interrupt control register	INT0IC	67
001E <sub>16</sub>	Processor mode register 2	PM2	36, 43	005E <sub>16</sub>	INT1 interrupt control register	INT1IC	67
001F <sub>16</sub>	Voltage down detection interrupt register	D4INT	30	005F <sub>16</sub>	INT2 interrupt control register	INT2IC	67
0020 <sub>16</sub>				0060 <sub>16</sub>			
0021 <sub>16</sub>	DMA0 source pointer	SAR0	81	0061 <sub>16</sub>			
0022 <sub>16</sub>				0062 <sub>16</sub>			
0023 <sub>16</sub>				0063 <sub>16</sub>			
0024 <sub>16</sub>	DMA0 destination pointer	DAR0	86	0064 <sub>16</sub>			
0025 <sub>16</sub>				0065 <sub>16</sub>			
0026 <sub>16</sub>				0066 <sub>16</sub>			
0027 <sub>16</sub>				0067 <sub>16</sub>			
0028 <sub>16</sub>	DMA0 transfer counter	TCR0	86	0068 <sub>16</sub>			
0029 <sub>16</sub>				0069 <sub>16</sub>			
002A <sub>16</sub>				006A <sub>16</sub>			
002B <sub>16</sub>				006B <sub>16</sub>			
002C <sub>16</sub>	DMA0 control register	DM0CON	85	006C <sub>16</sub>			
002D <sub>16</sub>				006D <sub>16</sub>			
002E <sub>16</sub>				006E <sub>16</sub>			
002F <sub>16</sub>				006F <sub>16</sub>			
0030 <sub>16</sub>				0070 <sub>16</sub>			
0031 <sub>16</sub>	DMA1 source pointer	SAR1	86	0071 <sub>16</sub>			
0032 <sub>16</sub>				0072 <sub>16</sub>			
0033 <sub>16</sub>				0073 <sub>16</sub>			
0034 <sub>16</sub>	DMA1 destination pointer	DAR1	86	0074 <sub>16</sub>			
0035 <sub>16</sub>				0075 <sub>16</sub>			
0036 <sub>16</sub>				0076 <sub>16</sub>			
0037 <sub>16</sub>				0077 <sub>16</sub>			
0038 <sub>16</sub>	DMA1 transfer counter	TCR1	86	0078 <sub>16</sub>			
0039 <sub>16</sub>				0079 <sub>16</sub>			
003A <sub>16</sub>				007A <sub>16</sub>			
003B <sub>16</sub>				007B <sub>16</sub>			
003C <sub>16</sub>	DMA1 control register	DM1CON	85	007C <sub>16</sub>			
003D <sub>16</sub>				007D <sub>16</sub>			
003E <sub>16</sub>				007E <sub>16</sub>			
003F <sub>16</sub>				007F <sub>16</sub>			

**NOTE:**

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

## Quick Reference by Address

Address	Register	Symbol	Page
0080 <sub>16</sub>			
0081 <sub>16</sub>			
0082 <sub>16</sub>			
0083 <sub>16</sub>			
0084 <sub>16</sub>			
0085 <sub>16</sub>			
0086 <sub>16</sub>			
01B0 <sub>16</sub>			
01B1 <sub>16</sub>			
01B2 <sub>16</sub>			
01B3 <sub>16</sub>	Flash memory control register 4 (2)	FMR4	242
01B4 <sub>16</sub>			
01B5 <sub>16</sub>	Flash memory control register 1 (2)	FMR1	241
01B6 <sub>16</sub>			
01B7 <sub>16</sub>	Flash memory control register 0 (2)	FMR0	241
01B8 <sub>16</sub>			
01B9 <sub>16</sub>			
01BA <sub>16</sub>			
01BB <sub>16</sub>			
01BC <sub>16</sub>			
01BD <sub>16</sub>			
01BE <sub>16</sub>			
01BF <sub>16</sub>			
0250 <sub>16</sub>			
0251 <sub>16</sub>			
0252 <sub>16</sub>			
0253 <sub>16</sub>			
0254 <sub>16</sub>			
0255 <sub>16</sub>			
0256 <sub>16</sub>			
0257 <sub>16</sub>			
0258 <sub>16</sub>			
0259 <sub>16</sub>			
025A <sub>16</sub>	Three phase protect control register	TPRC	131
025B <sub>16</sub>			
025C <sub>16</sub>	On-chip oscillator control register	ROCR	41
025D <sub>16</sub>	Pin assignment control register	PACR	139, 226
025E <sub>16</sub>	Peripheral clock select register	PCLKR	43
025F <sub>16</sub>			
02E0 <sub>16</sub>			
02E1 <sub>16</sub>			
02E2 <sub>16</sub>			
02E3 <sub>16</sub>			
02E4 <sub>16</sub>			
02E5 <sub>16</sub>			
02E6 <sub>16</sub>			
02E7 <sub>16</sub>			
02E8 <sub>16</sub>			
02E9 <sub>16</sub>			
033D <sub>16</sub>			
033E <sub>16</sub>	NMI digital debounce register	NDDR	227
033F <sub>16</sub>	P17 digital debounce register	P17DDR	227

### NOTES:

1. The blank areas are reserved and cannot be accessed by users.
2. This register is included in the flash memory version.

Address	Register	Symbol	Page
0340 <sub>16</sub>			
0341 <sub>16</sub>			
0342 <sub>16</sub>	Timer A1-1 register	TA11	122
0343 <sub>16</sub>			
0344 <sub>16</sub>	Timer A2-1 register	TA21	122
0345 <sub>16</sub>			
0346 <sub>16</sub>	Timer A4-1 register	TA41	122
0347 <sub>16</sub>			
0348 <sub>16</sub>	Three-phase PWM control register 0	INVC0	119
0349 <sub>16</sub>	Three-phase PWM control register 1	INVC1	120
034A <sub>16</sub>	Three-phase output buffer register 0	IDB0	121
034B <sub>16</sub>	Three-phase output buffer register 1	IDB1	121
034C <sub>16</sub>	Dead time timer	DTT	121
034D <sub>16</sub>	Timer B2 interrupt occurrence frequency set counter	ICTB2	122
034E <sub>16</sub>	Position-data-retain function control register	PDRF	129
034F <sub>16</sub>			
0350 <sub>16</sub>			
0351 <sub>16</sub>			
0352 <sub>16</sub>			
0353 <sub>16</sub>			
0354 <sub>16</sub>			
0355 <sub>16</sub>			
0356 <sub>16</sub>			
0357 <sub>16</sub>			
0358 <sub>16</sub>	Port function control register	PFGR	131
0359 <sub>16</sub>			
035A <sub>16</sub>			
035B <sub>16</sub>			
035C <sub>16</sub>			
035D <sub>16</sub>			
035E <sub>16</sub>	Interrupt request cause select register 2	IFSR2A	68
035F <sub>16</sub>	Interrupt request cause select register	IFSR	68, 76
0360 <sub>16</sub>			
0361 <sub>16</sub>			
0362 <sub>16</sub>			
0363 <sub>16</sub>			
0364 <sub>16</sub>			
0365 <sub>16</sub>			
0366 <sub>16</sub>			
0367 <sub>16</sub>			
0368 <sub>16</sub>			
0369 <sub>16</sub>			
036A <sub>16</sub>			
036B <sub>16</sub>			
036C <sub>16</sub>			
036D <sub>16</sub>			
036E <sub>16</sub>			
036F <sub>16</sub>			
0370 <sub>16</sub>			
0371 <sub>16</sub>			
0372 <sub>16</sub>			
0373 <sub>16</sub>			
0374 <sub>16</sub>	UART2 special mode register 4	U2SMR4	141
0375 <sub>16</sub>	UART2 special mode register 3	U2SMR3	141
0376 <sub>16</sub>	UART2 special mode register 2	U2SMR2	140
0377 <sub>16</sub>	UART2 special mode register	U2SMR	140
0378 <sub>16</sub>	UART2 transmit/receive mode register	U2MR	137
0379 <sub>16</sub>	UART2 bit rate generator	U2BRG	136
037A <sub>16</sub>			
037B <sub>16</sub>	UART2 transmit buffer register	U2TB	136
037C <sub>16</sub>	UART2 transmit/receive control register 0	U2C0	138
037D <sub>16</sub>	UART2 transmit/receive control register 1	U2C1	139
037E <sub>16</sub>			
037F <sub>16</sub>	UART2 receive buffer register	U2RB	136



## Quick Reference by Address

Address	Register	Symbol	Page
0380 <sub>16</sub>	Count start flag	TABSR	95, 110, 124
0381 <sub>16</sub>	Clock prescaler reset flag	CPSRF	r 0)
0382 <sub>16</sub>	One-shot start flag	ONSF	96
0383 <sub>16</sub>	Trigger select register	TRGSR	96, 124
0384 <sub>16</sub>	Up-down flag	UDF	95
0385 <sub>16</sub>			
0386 <sub>16</sub>	Timer A0 register	TA0	95
0387 <sub>16</sub>			
0388 <sub>16</sub>	Timer A1 register	TA1	95, 122
0389 <sub>16</sub>			
038A <sub>16</sub>	Timer A2 register	TA2	95, 122
038B <sub>16</sub>			
038C <sub>16</sub>	Timer A3 register	TA3	95
038D <sub>16</sub>			
038E <sub>16</sub>	Timer A4 register	TA4	95, 122
038F <sub>16</sub>			
0390 <sub>16</sub>	Timer B0 register	TB0	110
0391 <sub>16</sub>			
0392 <sub>16</sub>	Timer B1 register	TB1	110
0393 <sub>16</sub>			
0394 <sub>16</sub>	Timer B2 register	TB2	110, 124
0395 <sub>16</sub>			
0396 <sub>16</sub>	Timer A0 mode register	TA0MR	94
0397 <sub>16</sub>	Timer A1 mode register	TA1MR	94, 125
0398 <sub>16</sub>	Timer A2 mode register	TA2MR	94, 125
0399 <sub>16</sub>	Timer A3 mode register	TA3MR	94
039A <sub>16</sub>	Timer A4 mode register	TA4MR	94, 125
039B <sub>16</sub>	Timer B0 mode register	TB0MR	109
039C <sub>16</sub>	Timer B1 mode register	TB1MR	109
039D <sub>16</sub>	Timer B2 mode register	TB2MR	109, 125
039E <sub>16</sub>	Timer B2 special mode register	TB2SC	123, 185
039F <sub>16</sub>			
03A0 <sub>16</sub>	UART0 transmit/receive mode register	U0MR	137
03A1 <sub>16</sub>	UART0 bit rate generator	U0BRG	136
03A2 <sub>16</sub>	UART0 transmit buffer register	U0TB	136
03A3 <sub>16</sub>			
03A4 <sub>16</sub>	UART0 transmit/receive control register 0	U0C0	138
03A5 <sub>16</sub>	UART0 transmit/receive control register 1	U0C1	139
03A6 <sub>16</sub>	UART0 receive buffer register	U0RB	136
03A7 <sub>16</sub>			
03A8 <sub>16</sub>	UART1 transmit/receive mode register	U1MR	137
03A9 <sub>16</sub>	UART1 bit rate generator	U1BRG	136
03AA <sub>16</sub>	UART1 transmit buffer register	U1TB	136
03AB <sub>16</sub>			
03AC <sub>16</sub>	UART1 transmit/receive control register 0	U1C0	138
03AD <sub>16</sub>	UART1 transmit/receive control register 1	U1C1	139
03AE <sub>16</sub>	UART1 receive buffer register	U1RB	136
03AF <sub>16</sub>			
03B0 <sub>16</sub>	UART transmit/receive control register 2	UCON	138
03B1 <sub>16</sub>			
03B2 <sub>16</sub>			
03B3 <sub>16</sub>			
03B4 <sub>16</sub>	CRC snoop address register	CRCSAR	214
03B5 <sub>16</sub>			
03B6 <sub>16</sub>	CRC mode register	CRCMR	214
03B7 <sub>16</sub>			
03B8 <sub>16</sub>	DMA0 request cause select register	DM0SL	84
03B9 <sub>16</sub>			
03BA <sub>16</sub>	DMA1 request cause select register	DM1SL	85
03BB <sub>16</sub>			
03BC <sub>16</sub>	CRC data register	CRCD	214
03BD <sub>16</sub>			
03BE <sub>16</sub>	CRC input register	CRCIN	214
03BF <sub>16</sub>			

**NOTE:**

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

Address	Register	Symbol	Page
03C0 <sub>16</sub>	A/D register 0	AD0	184
03C1 <sub>16</sub>			
03C2 <sub>16</sub>	A/D register 1	AD1	184
03C3 <sub>16</sub>			
03C4 <sub>16</sub>	A/D register 2	AD2	184
03C5 <sub>16</sub>			
03C6 <sub>16</sub>	A/D register 3	AD3	184
03C7 <sub>16</sub>			
03C8 <sub>16</sub>	A/D register 4	AD4	184
03C9 <sub>16</sub>			
03CA <sub>16</sub>	A/D register 5	AD5	184
03CB <sub>16</sub>			
03CC <sub>16</sub>	A/D register 6	AD6	184
03CD <sub>16</sub>			
03CE <sub>16</sub>	A/D register 7	AD7	184
03CF <sub>16</sub>			
03D0 <sub>16</sub>			
03D1 <sub>16</sub>			
03D2 <sub>16</sub>	A/D trigger control register	ADTRGCON	183
03D3 <sub>16</sub>	A/D convert status register 0	ADSTAT0	184
03D4 <sub>16</sub>	A/D control register 2	ADCON2	182
03D5 <sub>16</sub>			
03D6 <sub>16</sub>	A/D control register 0	ADCON0	182
03D7 <sub>16</sub>	A/D control register 1	ADCON1	182
03D8 <sub>16</sub>			
03D9 <sub>16</sub>			
03DA <sub>16</sub>			
03DB <sub>16</sub>			
03DC <sub>16</sub>			
03DD <sub>16</sub>			
03DE <sub>16</sub>			
03DF <sub>16</sub>			
03E0 <sub>16</sub>			
03E1 <sub>16</sub>	Port P1 register	P1	224
03E2 <sub>16</sub>			
03E3 <sub>16</sub>	Port P1 direction register	PD1	223
03E4 <sub>16</sub>			
03E5 <sub>16</sub>			
03E6 <sub>16</sub>			
03E7 <sub>16</sub>			
03E8 <sub>16</sub>			
03E9 <sub>16</sub>			
03EA <sub>16</sub>			
03EB <sub>16</sub>			
03EC <sub>16</sub>	Port P6 register	P6	224
03ED <sub>16</sub>	Port P7 register	P7	224
03EE <sub>16</sub>	Port P6 direction register	PD6	223
03EF <sub>16</sub>	Port P7 direction register	PD7	223
03F0 <sub>16</sub>	Port P8 register	P8	224
03F1 <sub>16</sub>	Port P9 register	P9	224
03F2 <sub>16</sub>	Port P8 direction register	PD8	223
03F3 <sub>16</sub>	Port P9 direction register	PD9	223
03F4 <sub>16</sub>	Port P10 register	P10	224
03F5 <sub>16</sub>			
03F6 <sub>16</sub>	Port P10 direction register	PD10	223
03F7 <sub>16</sub>			
03F8 <sub>16</sub>			
03F9 <sub>16</sub>			
03FA <sub>16</sub>			
03FB <sub>16</sub>			
03FC <sub>16</sub>	Pull-up control register 0	PUR0	225
03FD <sub>16</sub>	Pull-up control register 1	PUR1	225
03FE <sub>16</sub>	Pull-up control register 2	PUR2	225
03FF <sub>16</sub>	Port control register	PCR	226

# M16C/26A Group (M16C/26A, M16C/26B, M16C/26T)

## SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

---

### 1. Overview

The M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) is a single-chip control MCU, fabricated using high-performance silicon gate CMOS technology, embedding the M16C/60 Series CPU core. The M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) is housed in 42-pin and 48-pin plastic molded packages. This MCU combines advanced instruction manipulation capabilities to process complex instructions by less bytes and execute instructions at higher speed. The M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) has a multiplier and DMAC adequate for office automation, communication devices and industrial equipment, and other high-speed processing applications. The M16C/26A and M16C/26B have normal version. The M16C/26T has T version and V version.

#### 1.1 Applications

Audio, cameras, office/communications/portable/ equipment, air-conditioning equipment, home appliances, etc.

## 1.2 Performance Outline

Table 1.1 and 1.2 outline performance overview of the M16C/26A Group (M16C/26A, M16C/26B, M16C/26T).

**Table 1.1. M16C/26A Group(M16C/26A, M16C/26B, M16C/26T) Performance (48-Pin Package)**

Item		Specification
CPU	Basic instructions	91 instructions
	Minimum instruction execution time	41.7 ns (f(BCLK) = 24MHz <sup>(3)</sup> , Vcc = 4.2 to 5.5 V) (M16C/26B)
		50 ns (f(BCLK) = 20MHz, Vcc = 3.0 to 5.5 V) (M16C/26A, M16C/26B, M16C/26T(T-ver.))
		100 ns (f(BCLK) = 10MHz, Vcc = 2.7 to 5.5 V) (M16C/26A, M16C/26B)
		50 ns (f(BCLK) = 20MHz, Vcc = 4.2 to 5.5 V -40 to 105°C) (M16C/26T(V-ver.))
		62.5 ns (f(BCLK) = 16MHz, Vcc = 4.2 to 5.5 V -40 to 125°C) (M16C/26T(V-ver.))
Operating mode	Single-chip mode	
Address space	1 Mbyte	
Memory capacity	ROM/RAM: See <b>1.4 Product Information</b>	
Peripheral Function	I/O ports	39 I/O pins
	Multifunction timers	TimerA:16 bits x 5 channels, TimerB:16 bits x 3 channels Three-phase motor control timer
	Serial I/O	2 channels (UART, clock synchronous serial I/O) 1 channel (UART, clock synchronous, I <sup>2</sup> C bus, or IEBus <sup>(1)</sup> )
	A/D converter	10 bit A/D Converter : 1 circuit, 12 channels
	DMAC	2 channels
	CRC calculation circuit	1 circuit (CRC-CCITT and CRC-16) with MSB/LSB selectable
	Watchdog timer	15 bits x 1 channel (with prescaler)
	Interrupts	20 internal and 8 external sources, 4 software sources, Interrupt priority level: 7
	Clock generation circuit	4 circuits Main clock oscillation circuit(*), Sub-clock oscillation circuit(*) On-chip oscillator, PLL frequency synthesizer (* )Equipped with a built-in feedback resistor.
	Oscillation stop detection	Main clock oscillation stop, re-oscillation detection function
	Voltage detection circuit	On-chip (M16C/26A, M16C/26B), not on-chip (M16C/26T)
Electrical Characteristics	Power supply voltage	Vcc = 4.2 to 5.5 V (f(BCLK) = 24 MHz) <sup>(3)</sup> (M16C/26B)
		Vcc = 3.0 to 5.5 V (f(BCLK) = 20 MHz) (M16C/26A, M16C/26B)
		Vcc = 2.7 to 5.5 V (f(BCLK) = 10 MHz)
		Vcc = 3.0 to 5.5 V (M16C/26T(T-ver.))
	Vcc = 4.2 to 5.5 V (M16C/26T(V-ver.))	
Power consumption	20 mA (Vcc = 5 V, f(BCLK) = 24 MHz) (M16C/26B) 16 mA (Vcc = 5 V, f(BCLK) = 20 MHz) 25 μA (f(XCIN) = 32 KHz on RAM) 3 μA (Vcc = 3 V, f(XCIN) = 32 KHz, in wait mode) 0.7 μA (Vcc = 3 V, in stop mode)	
Flash Memory Version	Programming /erasure voltage	2.7 to 5.5 V (M16C/26A, M16C/26B) 3.0 to 5.5 V (M16C/26T(T-ver.)) 4.2 to 5.5 V (M16C/26T(V-ver.))
	Programming /erasure endurance	100 times (all area) or 1,000 times (block 0 to 3) / 10,000 times (block A, block B) <sup>(2)</sup>
Operating Ambient Temperature		-20 to 85°C / -40 to 85°C <sup>(2)</sup> (M16C/26A, M16C/26B)
		-40 to 85°C (M16C/26T(T-ver.))
		-40 to 105°C / -40 to 125°C (M16C/26T(V-ver.))
Package	48-pin plastic molded QFP	

**NOTES:**

1. IEBus is a trademark of NEC Electronics Corporation.
2. See **Tables 1.7 to 1.10 Product Code** for the program and erase endurance, and operating ambient temperature.
3. The PLL frequency synthesizer is used to run the M16C/26B at f(BCLK) = 24 MHz.

**Table 1.2. Performance outline of M16C/26A group (M16C/26A, M16C/26B) (42-pin package)**

	Item	Performance
CPU	Basic instructions	91 instructions
	Minimum instruction execution time	41.7 ns (f(BCLK) = 24 MHz <sup>(3)</sup> , VCC = 4.2 to 5.5 V) (M16C/26B)
		50 ns (f(BCLK) = 20 MHz, Vcc = 3.0 to 5.5 V) (M16C/26A, M16C/26B)
		100 ns (f(BCLK) = 10 MHz, Vcc = 2.7 to 5.5 V) (M16C/26A, M16C/26B)
	Operation mode	Single-chip mode
Address space	1M byte	
Memory capacity	ROM/RAM: See <b>1.4 Product Information</b>	
Peripheral function	Port	33 I/O pins
	Multifunction timer	Timer A: 16 bits x 5 channels, Timer B: 16 bits x 3 channels Three-phase motor control timer
	Serial I/O	1 channel (UART, clock synchronous serial I/O)
		1 channel (UART, clock synchronous, I <sup>2</sup> C bus, or IEBus <sup>(1)</sup> )
	A/D converter	10 bit A/D converter: 1 circuit, 10 channels
	DMAC	2 channels
	CRC calculation circuit	1 circuits (CRC-CCITT and CRC-16) with MSB/LSB selectable
	Watchdog timer	15 bits x 1 channel (with prescaler)
	Interrupt	18 internal and 8 external sources, 4 software sources, Interrupt priority level: 7
	Clock generation circuit	4 circuits Main clock(*), Sub-clock(*) On-chip oscillator, PLL frequency synthesizer (* )Equipped with a built-in feedback resistor.
Oscillation stop detection		Main clock oscillation stop, re-oscillation detection function
Voltage detection circuit	On-chip	
	Supply voltage	Vcc = 4.2 to 5.5 V (f(BCLK) = 24 MHz) <sup>(3)</sup> (M16C/26B) Vcc = 3.0 to 5.5 V (f(BCLK) = 20 MHz) (M16C/26A, M16C/26B) Vcc = 2.7 to 5.5 V (f(BCLK) = 10 MHz)
Electrical Characteristics	Power Consumption	20 mA (Vcc = 5 V, f(BCLK) = 24 MHz) (M16C/26B)
		16 mA (Vcc = 5 V, f(BCLK) = 20 MHz)
		25 μA (f(XCIN) = 32 KHz on RAM)
		3 μA (Vcc = 3 V, f(XCIN) = 32 KHz, in wait mode)
		0.7 μA (Vcc = 3 V, in stop mode)
Flash memory	Programming/erasure voltage	2.7 to 5.5 V
	Programming/erasure endurance	100 times (all area) or 1,000 times (block 0 to 3) / 10,000 times (block A, block B) <sup>(2)</sup>
Operating Ambient Temperature		-20 to 85°C / -40 to 85°C <sup>(2)</sup>
Package		42-pin plastic molded SSOP

## NOTES:

- IEBus is a trademark of NEC Electronics Corporation.
- See **Tables 1.7 and 1.8 Product Code** for the program and erase endurance, and operating ambient temperature.
- The PLL frequency synthesizer is used to run the M16C/26B at f(BCLK) = 24 MHz.

### 1.3 Block Diagram

Figure 1.1 and 1.2 show block diagrams of the M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) 48-pin package and 42-pin package.

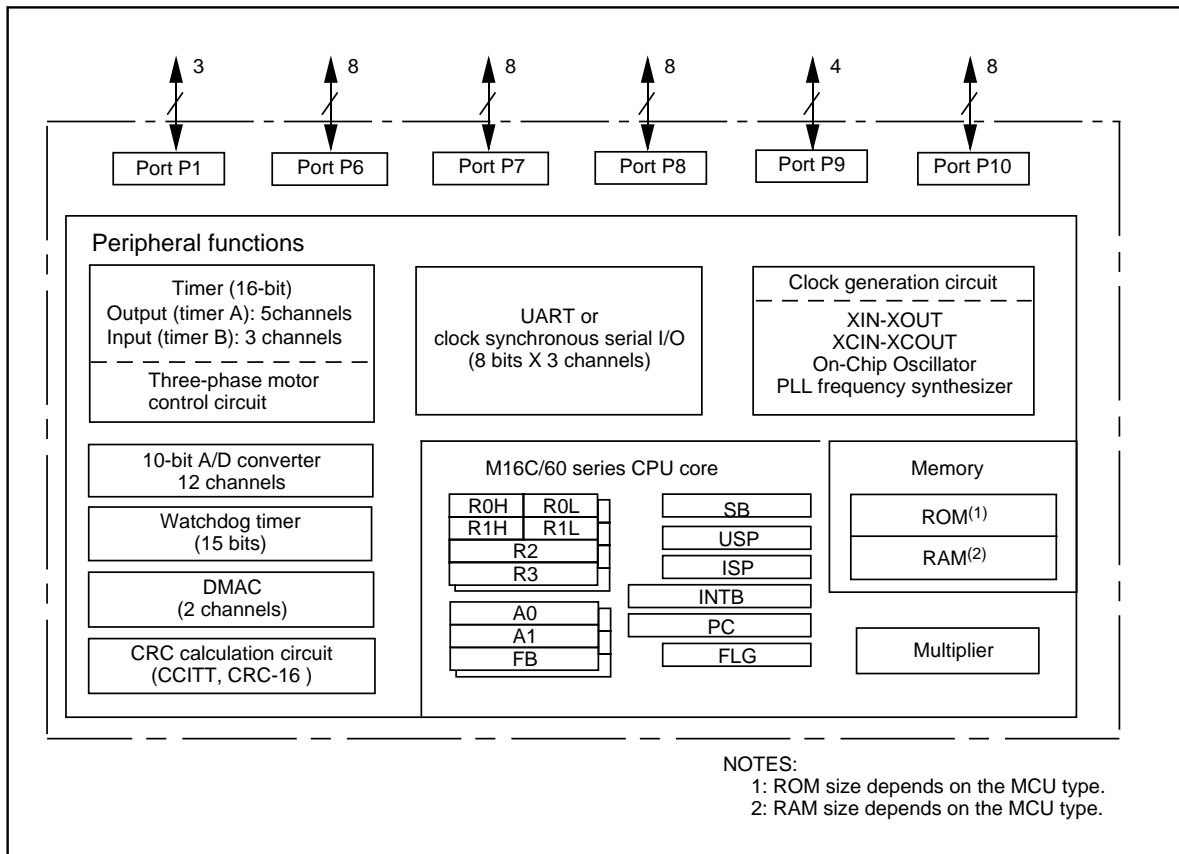


Figure 1.1 Block Diagram(48-pin Package)

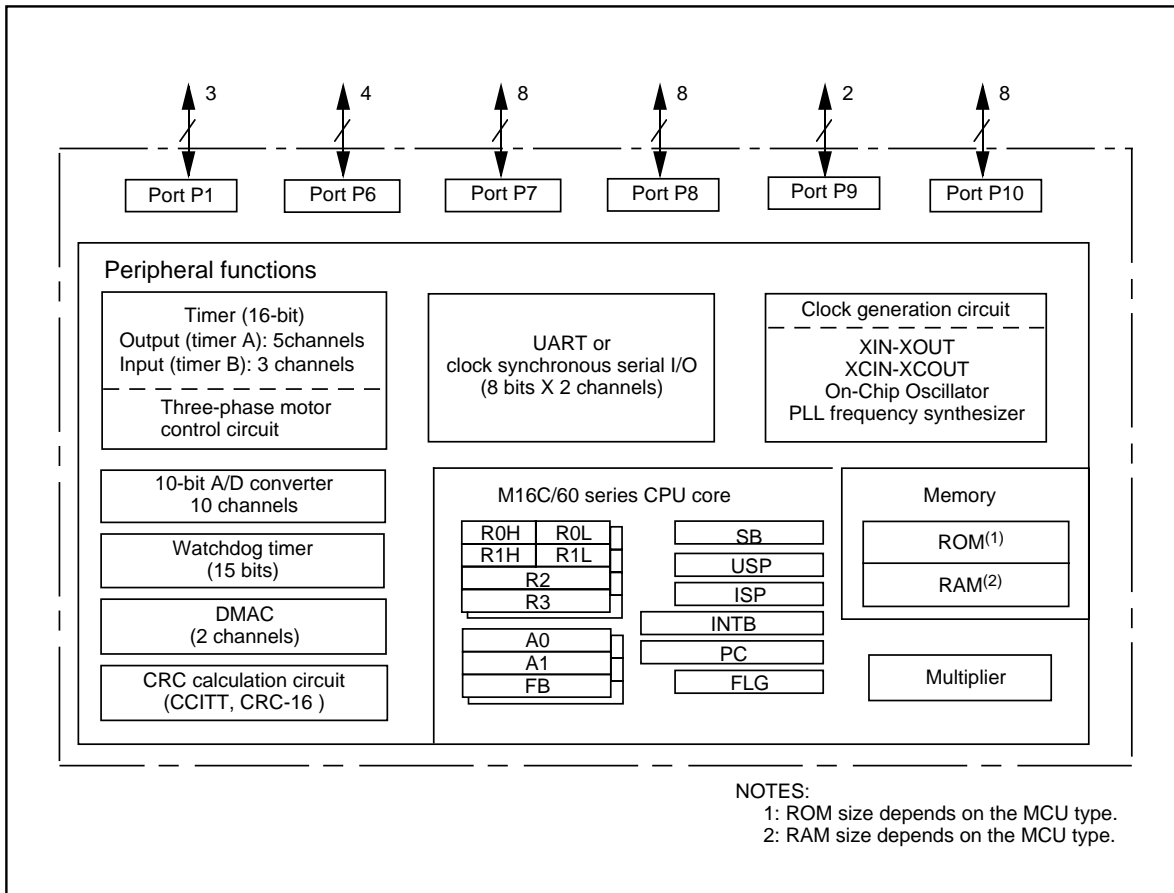


Figure 1.2 Block Diagram( 42-pin Package)

## 1.4 Product List

Tables 1.3 to 1.6 lists product information, Figure 1.3 shows a product numbering system, Table 1.7 lists the product code, and Figure 1.4 shows the marking.

**Table 1.3 M16C/26A**

**Current as of Feb., 2007**

Type Number	ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30260F3AGP (N)	24K + 4K	1K	PLQP0048KB-A (48P6Q-A)	Flash memory	U3, U5, U7, U9
M30260F6AGP (N)	48K + 4K	2K			
M30260F8AGP (N)	64K + 4K	2K			
M30263F3AFP (N)	24K + 4K	1K	PRSP0042GA-B (42P2R)		U5, U9
M30263F6AFP (N)	48K + 4K	2K			
M30263F8AFP (N)	64K + 4K	2K			
M30260M3A-XXXGP (N)	24K	1K	PLQP0048KB-A (48P6Q-A)	Mask ROM	U3, U5
M30260M6A-XXXGP (N)	48K	2K			
M30260M8A-XXXGP (N)	64K	2K			
M30263M3A-XXXFP (N)	24K	1K	PRSP0042GA-B (42P2R)		U5
M30263M6A-XXXFP (N)	48K	2K			
M30263M8A-XXXFP (N)	64K	2K			

(N): New

**Table 1.4 M16C/26B**

**Current as of Feb., 2007**

Type Number	ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30260F8BGP (N)	64K + 4K	2K	PLQP0048KB-A (48P6Q-A)	Flash memory	U7
M30263F8BFP (N)	64K + 4K	2K	PRSP0042GA-B (42P2R)		U9

(N): New

**Table 1.5 M16C/26T T-ver.**

**Current as of Feb., 2007**

Type Number	ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30260F3TGP	24K + 4K	1K	PLQP0048KB-A (48P6Q-A)	Flash memory	U3, U7
M30260F6TGP	48K + 4K	2K			
M30260F8TGP	64K + 4K	2K			

NOTE:

1. Available in flash memory version only.

**Table 1.6 M16C/26T V-ver.**

**Current as of Feb., 2007**

Type Number	ROM Capacity	RAM Capacity	Package	Remarks	Product Code
M30260F8VGP	64K + 4K	2K	PLQP0048KB-A (48P6Q-A)	Flash memory	U3, U7

NOTE:

1. Available in flash memory version only.

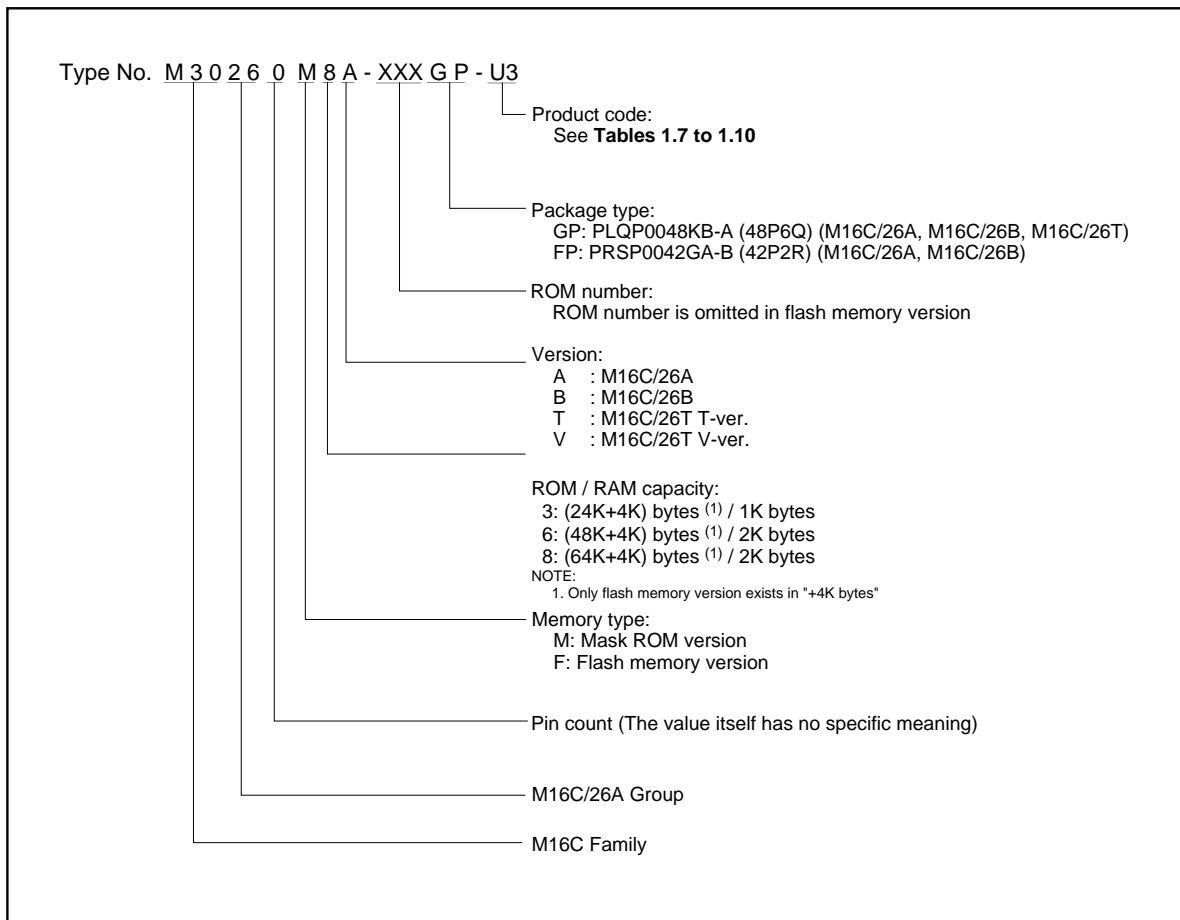


Figure 1.3 Product Numbering System



**Table 1.7 Product Code (Flash Memory Version) - M16C/26A, M16C/26B**

Product Code	Package	Internal ROM (Program Space: Blocks 0 to 3)		Internal ROM (Data Space: Blocks A and B)		Operating Ambient Temperature
		Program and Erase Endurance	Temperature Range	Program and Erase Endurance	Temperature Range	
U3	Lead free	100	0 to 60°C	100	0 to 60°C	-40 to 85°C
U5					-20 to 85°C	
U7		1,000		10,000	-40 to 85°C	-40 to 85°C
U9				-20 to 85°C	-20 to 85°C	

**Table 1.8 Product Code (Mask ROM Version - M16C/26A)**

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

## NOTE:

- The lead contained products, D3, D5, D7, and D9 are put together with U3, U5, U7, and U9 respectively. Lead-free products can be mounted by both conventional Sn-Pb paste and Lead-free paste (Sn-Ag-Cu plating).

**Table 1.9 Product Code (Flash Memory Version) - M16C/26T T-ver.**

Product Code	Package	Internal ROM (Program Space: Blocks 0 to 3)		Internal ROM (Data Space: Blocks A and B)		Operating Ambient Temperature
		Programming and erasure endurance	Temperature range	Programming and erasure endurance	Temperature range	
U3	Lead free	100	0°C to 60°C	100	-40°C to 85°C	-40°C to 85°C
U7		1,000		10,000		

**Table 1.10 Product Code (Flash Memory Version) - M16C/26T V-ver.**

Product Code	Package	Internal ROM (Program Space: Blocks 0 to 3)		Internal ROM (Data Space: Blocks A and B)		Operating Ambient Temperature
		Programming and erasure endurance	Temperature range	Programming and erasure endurance	Temperature range	
U3	Lead free	100	0°C to 60°C	100	-40°C to 125°C	-40°C to 125°C
U7		1,000		10,000		

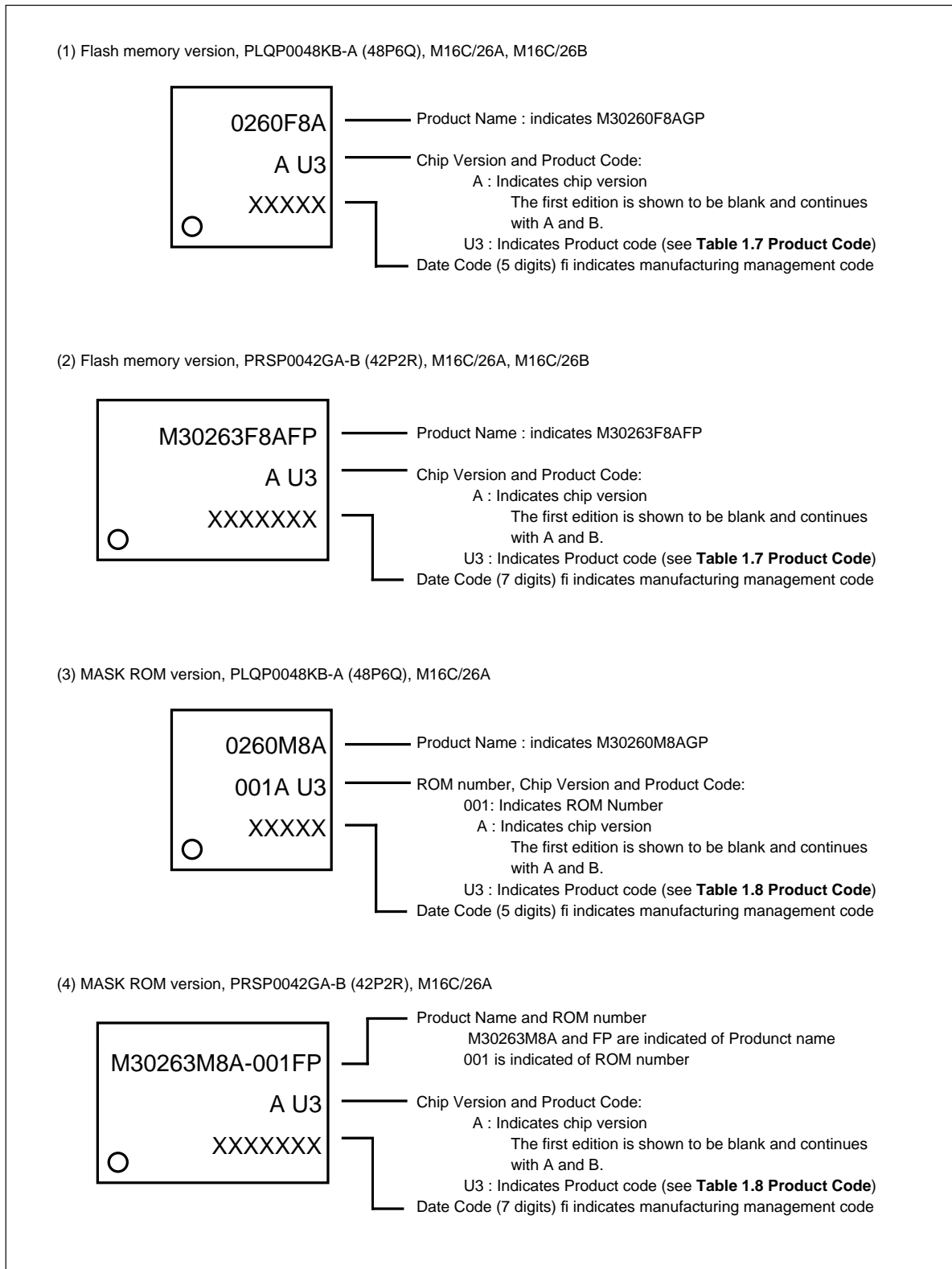


Figure 1.4 Marking Diagram (M16C/26A , M16C/26B)

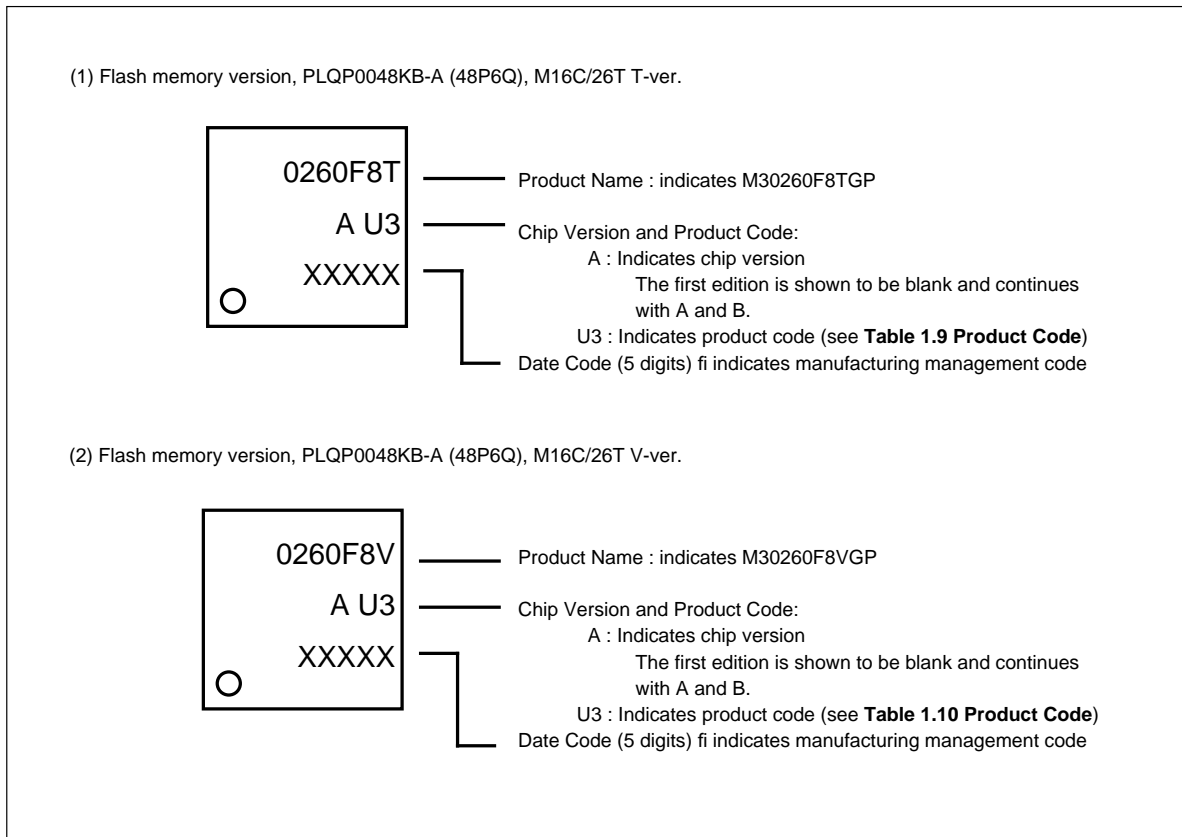


Figure 1.5 Marking Diagram (M16C/26T)

## 1.5 Pin Assignments

Figures 1.6 and 1.7 show the Pin Assignments (top view).

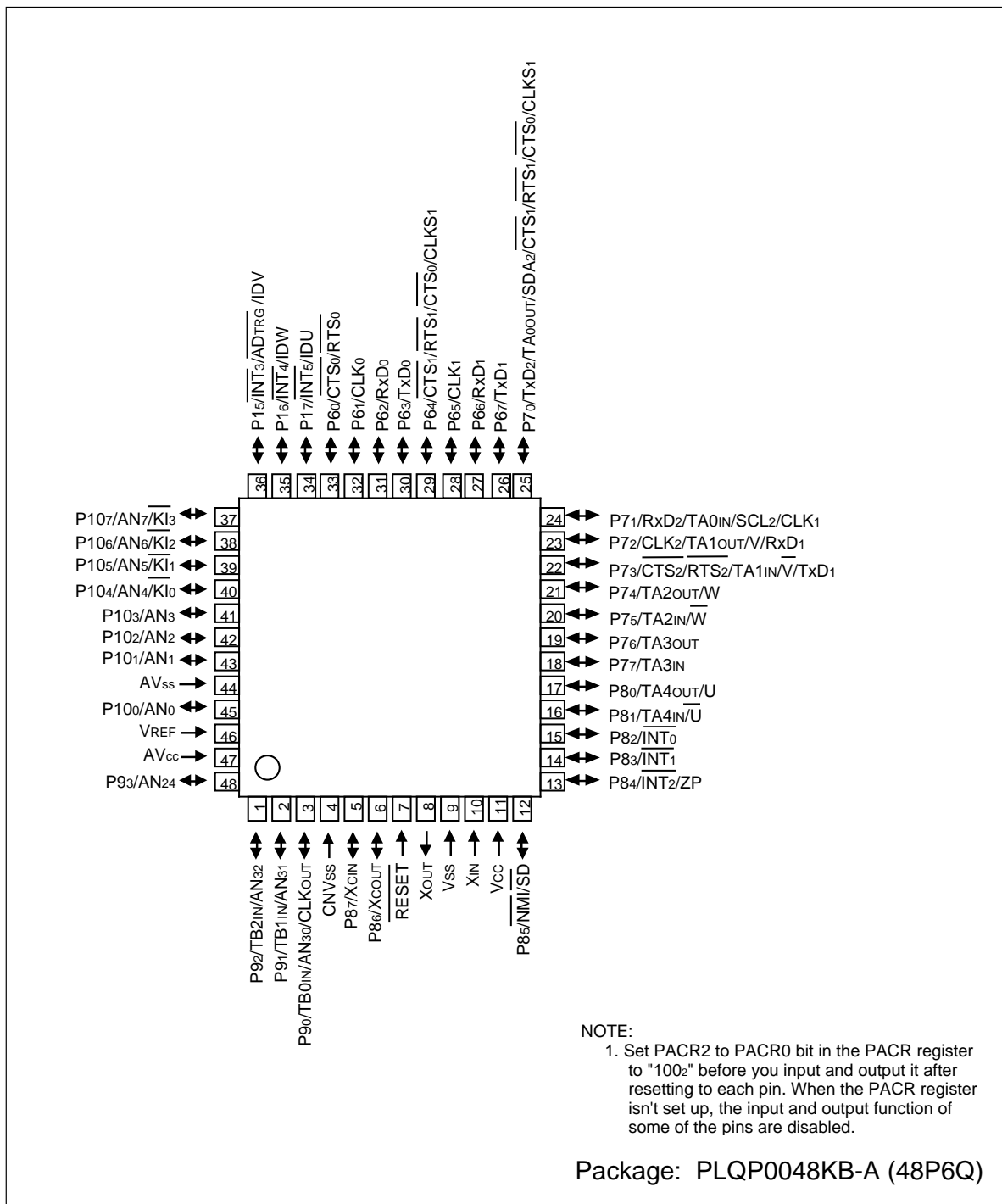


Figure 1.6 Pin Assignment for 48-Pin Package (Top View)

Table 1.11 Pin Characteristics for 48-Pin Package

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin
1		P92		TB2IN		AN32
2		P91		TB1IN		AN31
3		P90		TB0IN	CLKOUT	AN30
4	CNVss					
5	XCIN	P87				
6	XCOU	P86				
7	RESET					
8	XOUT					
9	Vss					
10	XIN					
11	Vcc					
12		P85	NMI	SD		
13		P84	INT2	ZP		
14		P83	INT1			
15		P82	INT0			
16		P81		TA4IN / U		
17		P80		TA4OUT / U		
18		P77		TA3IN		
19		P76		TA3OUT		
20		P75		TA2IN / W		
21		P74		TA2OUT / W		
22		P73		TA1IN / V	CTS2 / RTS2 / TxD1	
23		P72		TA1OUT / V	CLK2 / RxD1	
24		P71		TA0IN	RxD2 / SCL2 / CLK1	
25		P70		TA0OUT	TxD2 / SDA2 / RTS1 / CTS1 / CTS0 / CLKS1	
26		P67			TxD1	
27		P66			RxD1	
28		P65			CLK1	
29		P64			RTS1 / CTS1 / CTS0 / CLKS1	
30		P63			TxD0	
31		P62			RxD0	
32		P61			CLK0	
33		P60			RTS0 / CTS0	
34		P17	INT5	IDU		
35		P16	INT4	IDW		
36		P15	INT3	IDV		ADTRG
37		P107	KI3			AN7
38		P106	KI2			AN6
39		P105	KI1			AN5
40		P104	KI0			AN4
41		P103				AN3
42		P102				AN2
43		P101				AN1
44	AVss					
45		P100				AN0
46	VREF					
47	AVcc					
48		P93				AN24

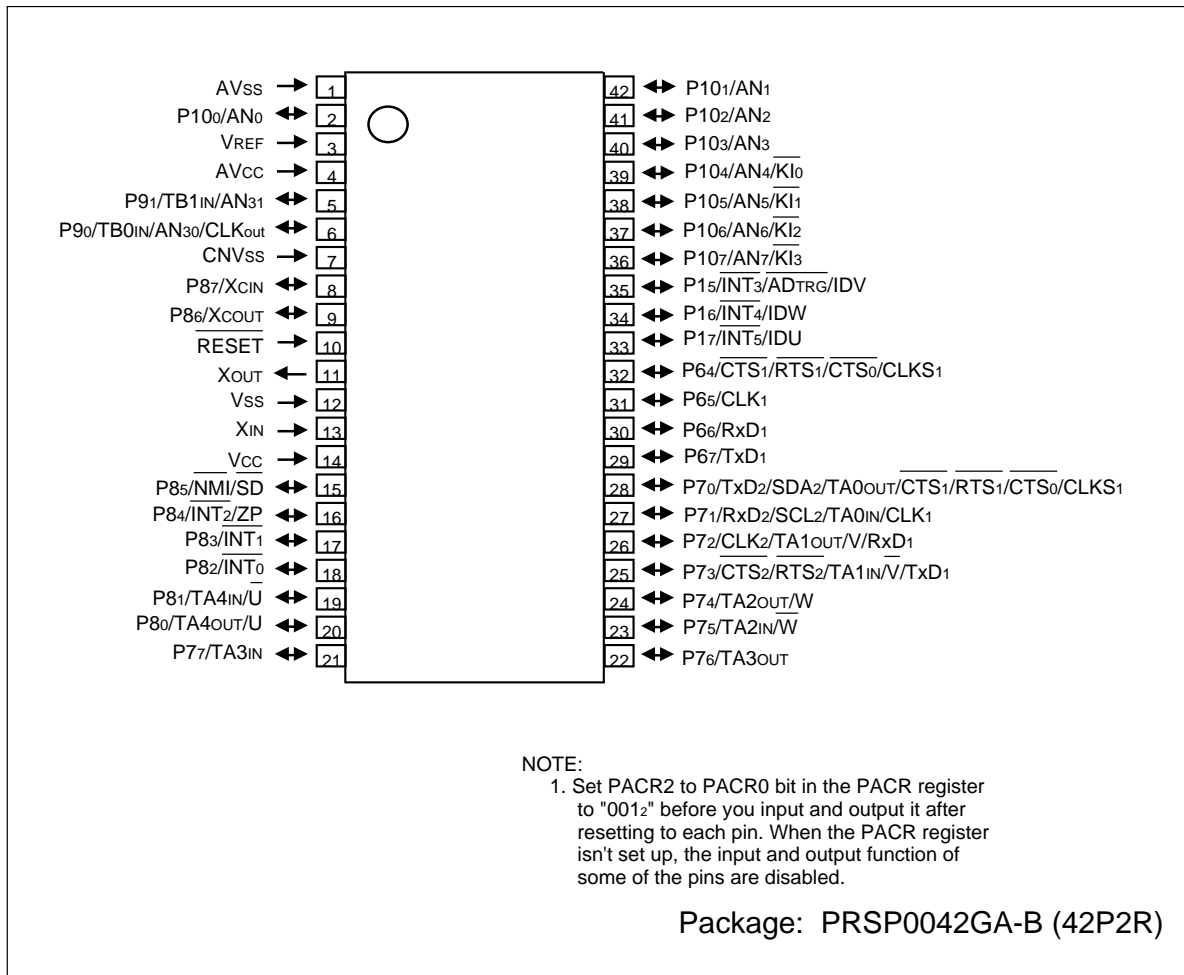


Figure 1.7 Pin Assignment for 42-Pin Package (Top View)

Table 1.12 Pin Characteristics for 42-Pin Package

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin
1	AVss					
2		P100				AN0
3	VREF					
4	AVcc					
5		P91		TB1IN		AN31
6		P90		TB0IN	CLKOUT	AN30
7	CNVss					
8	XCIN	P87				
9	XCOUT	P86				
10	RESET					
11	XOUT					
12	Vss					
13	XIN					
14	Vcc					
15		P85	NMI	SD		
16		P84	INT2	ZP		
17		P83	INT1			
18		P82	INT0			
19		P81		TA4IN / U		
20		P80		TA4OUT / U		
21		P77		TA3IN		
22		P76		TA3OUT		
23		P75		TA2IN / W		
24		P74		TA2OUT / W		
25		P73		TA1IN / V	CTS2 / RTS2 / TxD1	
26		P72		TA1OUT / V	CLK2 / RxD1	
27		P71		TA0IN	RxD2 / SCL2 / CLK1	
28		P70		TA0OUT	TxD2 / SDA2 / RTS1 / CTS1 / CTS0 / CLKS1	
29		P67			TxD1	
30		P66			RxD1	
31		P65			CLK1	
32		P64			RTS1 / CTS1 / CTS0 / CLKS1	
33		P17	INT5	IDU		
34		P16	INT4	IDW		
35		P15	INT3	IDV		ADTRG
36		P107	KI3			AN7
37		P106	KI2			AN6
38		P105	KI1			AN5
39		P104	KI0			AN4
40		P103				AN3
41		P102				AN2
42		P101				AN1

## 1.6 Pin Description

**Table 1.13 Pin Description (48-Pin and 42-Pin Packages)**

Classification	Pin Name	I/O Type	Description
Power Supply	Vcc, Vss	I	Apply 0V to the Vss pin. Apply following voltage to the Vcc pin. 2.7 to 5.5 V (M16C/26A, M16C/26B), 3.0 to 5.5 V (M16C/26T T-ver.), 4.2 to 5.5 V (M16C/26T V-ver.)
Analog Power Supply	AVcc AVss	I	Supplies power to the A/D converter. Connect the AVcc pin to Vcc and the AVss pin to Vss
Reset Input	RESET	I	The MCU is in a reset state when "L" is applied to the RESET pin
CNVSS	CNVss	I	Connect the CNVss pin to Vss
Main Clock Input	XIN	I	I/O pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT. To apply external clock, apply it to XIN and leave XOUT open. If XIN is not used (for external oscillator or external clock), connect XIN pin to VCC and leave XOUT open
Main Clock Output	XOUT	O	
Sub Clock Input	XCIN	I	I/O pins for the sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOUT
Sub Clock Output	XCOUT	O	
Clock Output	CLKOUT	O	Outputs the clock having the same frequency as f1, f8, f32, or fc
INT Interrupt Input	INT0 to INT5	I	Input pins for the INT interrupt. INT2 can be used for Timer A Z-phase function
NMI Interrupt Input	NMI	I	NMI interrupt input pin. NMI cannot be used as I/O port while the three-phase motor control is enabled. Apply a stable "H" to NMI after setting it's direction register to "0" when the three-phase motor control is enabled
Key Input Interrupt	KI0 to KI3	I	Input pins for the key input interrupt
Timer A	TA0OUT to TA4OUT	I/O	I/O pins for the timer A0 to A4
	TA0IN to TA4IN	I	Input pins for the timer A0 to A4
	ZP	I	Input pin for Z-phase
Timer B	TB0IN to TB1IN	I	Timer B0 to B1 input pins
Three-Phase Motor Control Timer Output	U, U, V, V, W, W	O	Output pins for the three-phase motor control timer
	IDU, IDW, IDV, SD	I/O	I/O pins for the three-phase motor control timer
Serial I/O	CTS1 to CTS2	I	Input pins to control data transmission
	RTS1 to RTS2	O	Output pins to control data reception
	CLK1 to CLK2	I/O	Inputs and outputs the transfer clock
	RxD1 to RxD2	I	Inputs serial data
	TxD1 to TxD2	O	Outputs serial data
	CLKS1	O	Output pin for transfer clock
Reference Voltage Input	VREF	I	Applies reference voltage to the A/D converter
A/D Converter	AN0 to AN7 AN30 to AN31	I	Analog input pins for the A/D converter
	ADTRG	I	Input pin for an external A/D trigger
I/O Ports	P15 to P17	I/O	I/O ports for CMOS. Each port can be programmed for input or output under the control of the direction register. An input port can be set, by program, for a pull-up resistor available or for no pull-up resistor available in 3-bit units
	P64 to P67 P70 to P77 P80 to P87 P100 to P107 P90 to P91	I/O	CMOS I/O ports which have a direction register determines an individual pin used as an input port or an output port. A pull-up resistor is selectable for every 4 input ports

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



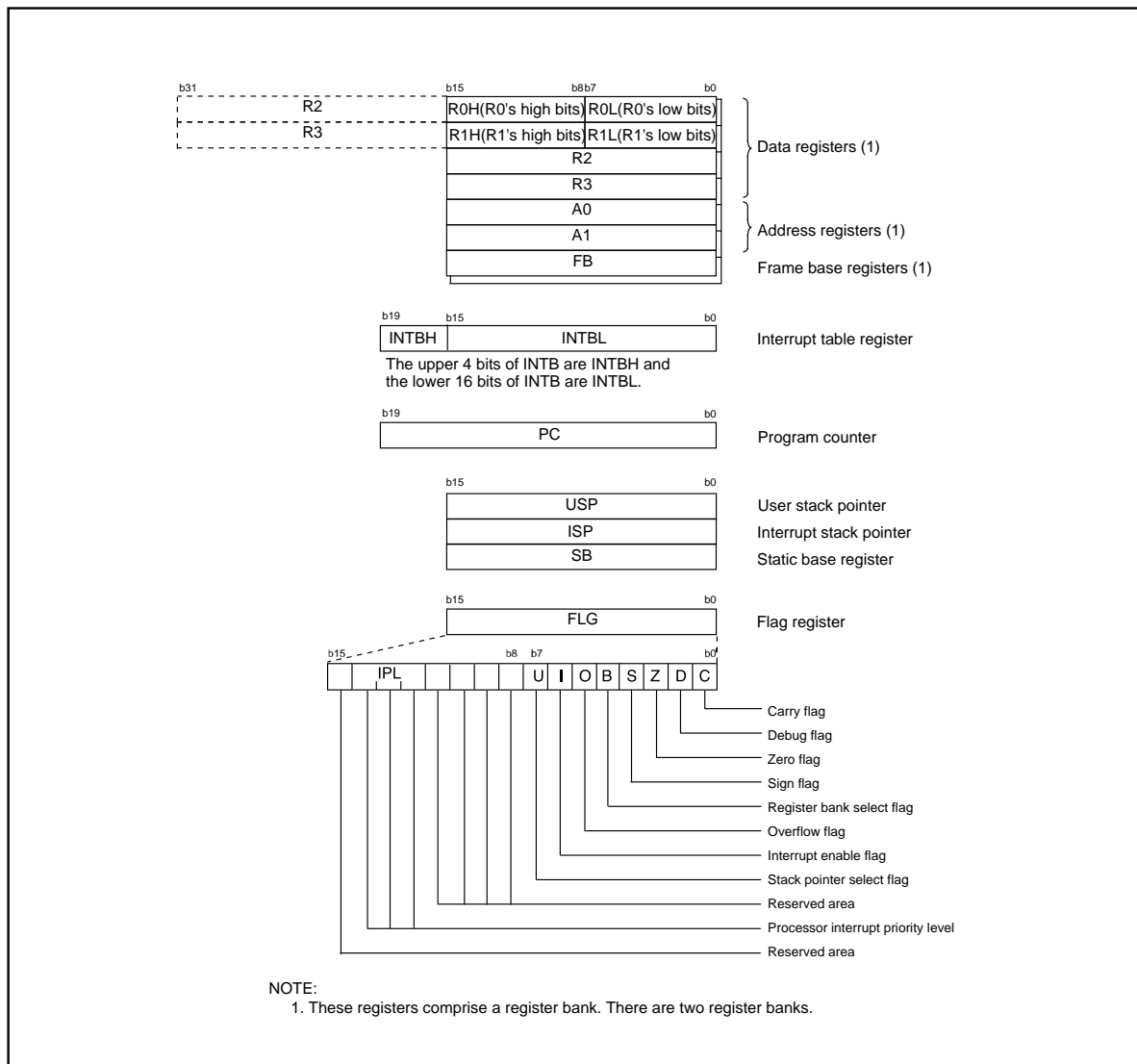
**Table 1.13 Pin Description ( 48-pin packages only) (Continued)**

Classification	Pin Name	I/O Type	Description
Serial I/O	CTS0	I	Inputs pin to control data transmission
	RTS0	O	Output pin to control data reception
	CLK0	I/O	Inputs and outputs the transfer clock
	RxD0	I	Inputs serial data
	TxD0	O	Outputs serial data
Timer B	TB2IN	I	Timer B2 input pin
A/D Converter	AN24	I	Analog input pins for the A/D converter
	AN32		
I/O Ports	P60 to P63 P92 to P93	I/O	CMOS I/O ports which have a direction register determines an individual pin used as an input port or an output port. A pull-up resistor is selectable for every 4 input ports

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

## 2. Central Processing Unit (CPU)

**Figure 2.1** shows the CPU registers. The register bank is comprised of seven registers (R0, R1, R2, R3, A0, A1 and FB) out of 13 registers. There are two sets of register bank.



**Figure 2.1. CPU Register**

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

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

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

### 2.2 Address Registers (A0 and A1)

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

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

## 2.3 Frame Base Register (FB)

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

## 2.4 Interrupt Table Register (INTB)

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

## 2.5 Program Counter (PC)

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

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

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

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

## 2.7 Static Base Register (SB)

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

## 2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

### 2.8.1 Carry Flag (C Flag)

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

### 2.8.2 Debug Flag (D Flag)

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

### 2.8.3 Zero Flag (Z Flag)

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

### 2.8.4 Sign Flag (S Flag)

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

### 2.8.5 Register Bank Select Flag (B Flag)

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

### 2.8.6 Overflow Flag (O Flag)

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

### 2.8.7 Interrupt Enable Flag (I Flag)

This flag enables a maskable interrupt.

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

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

### 2.8.8 Stack Pointer Select Flag (U Flag)

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

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

### 2.8.9 Processor Interrupt Priority Level (IPL)

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

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

### 2.8.10 Reserved Area

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

### 3. Memory

**Figure 3.1** is a memory map of the M16C/26A Group (M16C/26A, M16C/26B, M16C/26T). The M16C/26A Group provides 1-Mbyte address space addresses  $00000_{16}$  to  $FFFFFF_{16}$ .

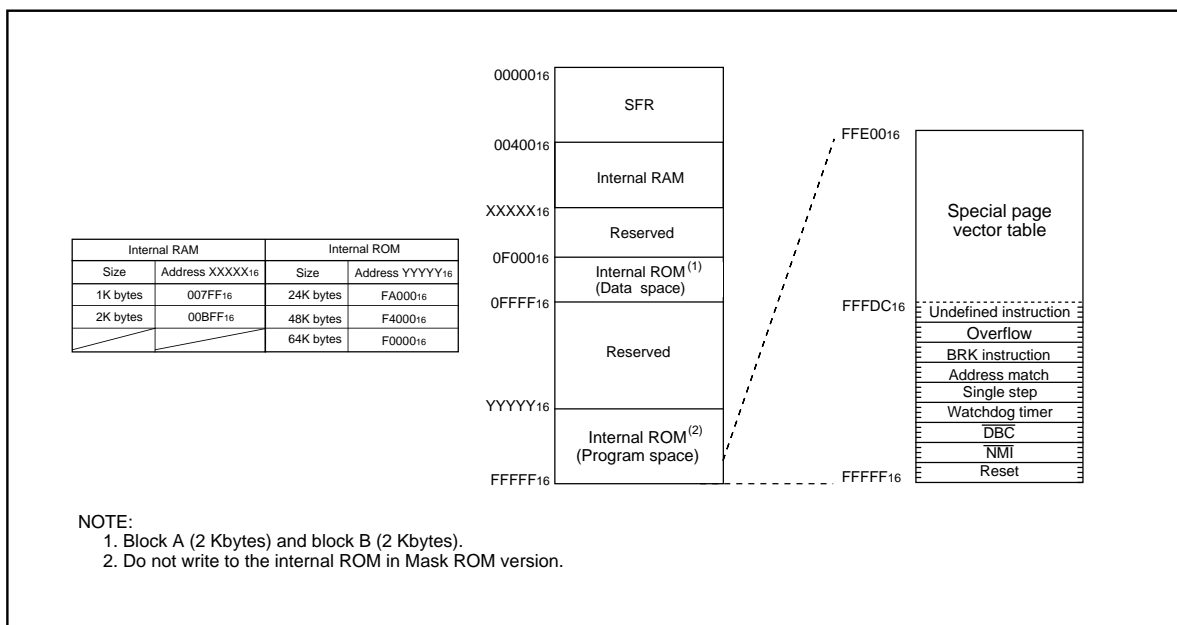
The internal ROM is allocated lower address, beginning with address  $FFFFFF_{16}$ . For example, a 64-Kbyte internal ROM area is allocated in addresses  $F0000_{16}$  to  $FFFFFF_{16}$ . The flash memory version has two sets of 2-Kbyte internal ROM area, block A and block B, for data space. These blocks are allocated addresses  $F000_{16}$  to  $FFFF_{16}$ .

The fixed interrupt vectors are allocated addresses  $FFFDC_{16}$  to  $FFFFFF_{16}$  and they store the start address of each interrupt routine.

The internal RAM is allocated higher addresses, beginning with address  $00400_{16}$ . For example, a 1-Kbyte internal RAM area is allocated in addresses  $00400_{16}$  to  $007FF_{16}$ . The internal RAM is used for temporarily storing data. The area is also used as stacks when subroutines are called or interrupt requests are acknowledged.

The SFR is allocated addresses  $00000_{16}$  to  $003FF_{16}$ . The peripheral function control registers are allocated here. All blank spaces within SFR location are reserved and cannot be accessed by users.

The special page vectors are allocated addresses  $FFE00_{16}$  to  $FFFDB_{16}$ . They are used for the JMPS instruction and JSRS instruction. Refer to the Renesas publication **M16C/60 and M16C/20 Series Software Manual** for details.



**Figure 3.1** Memory Map

## 4. Special Function Registers (SFRs)

Table 4.1 SFR Information(1)(1)

Address	Register	Symbol	After reset
0000 <sub>16</sub>			
0001 <sub>16</sub>			
0002 <sub>16</sub>			
0003 <sub>16</sub>			
0004 <sub>16</sub>	Processor mode register 0	PM0	00 <sub>16</sub>
0005 <sub>16</sub>	Processor mode register 1	PM1	00001000 <sub>2</sub>
0006 <sub>16</sub>	System clock control register 0	CM0	01001000 <sub>2</sub> <sup>(5)</sup> 01101000 <sub>2</sub> (M16C/26T)
0007 <sub>16</sub>	System clock control register 1	CM1	00100000 <sub>2</sub>
0008 <sub>16</sub>			
0009 <sub>16</sub>	Address match interrupt enable register	AIER	XXXXXX00 <sub>2</sub>
000A <sub>16</sub>	Protect register	PRCR	XX000000 <sub>2</sub>
000B <sub>16</sub>			
000C <sub>16</sub>	Oscillation stop detection register <sup>(2)</sup>	CM2	0X000000 <sub>2</sub>
000D <sub>16</sub>			
000E <sub>16</sub>	Watchdog timer start register	WDTS	XX <sub>16</sub>
000F <sub>16</sub>	Watchdog timer control register	WDC	00XXXXXX <sub>2</sub>
0010 <sub>16</sub>	Address match interrupt register 0	RMAD0	00 <sub>16</sub>
0011 <sub>16</sub>			00 <sub>16</sub>
0012 <sub>16</sub>			X0 <sub>16</sub>
0013 <sub>16</sub>			
0014 <sub>16</sub>	Address match interrupt register 1	RMAD1	00 <sub>16</sub>
0015 <sub>16</sub>			00 <sub>16</sub>
0016 <sub>16</sub>			X0 <sub>16</sub>
0017 <sub>16</sub>			
0018 <sub>16</sub>			
0019 <sub>16</sub>	Voltage detection register 1 <sup>(3, 4)</sup>	VCR1	00001000 <sub>2</sub>
001A <sub>16</sub>	Voltage detection register 2 <sup>(3, 4)</sup>	VCR2	00 <sub>16</sub>
001B <sub>16</sub>			
001C <sub>16</sub>	PLL control register 0	PLC0	0001X010 <sub>2</sub>
001D <sub>16</sub>			
001E <sub>16</sub>	Processor mode register 2	PM2	XXX00000 <sub>2</sub>
001F <sub>16</sub>	Low voltage detection interrupt register <sup>(4)</sup>	D4INT	00 <sub>16</sub>
0020 <sub>16</sub>	DMA0 source pointer	SAR0	XX <sub>16</sub>
0021 <sub>16</sub>			XX <sub>16</sub>
0022 <sub>16</sub>			XX <sub>16</sub>
0023 <sub>16</sub>			
0024 <sub>16</sub>	DMA0 destination pointer	DAR0	XX <sub>16</sub>
0025 <sub>16</sub>			XX <sub>16</sub>
0026 <sub>16</sub>			XX <sub>16</sub>
0027 <sub>16</sub>			
0028 <sub>16</sub>	DMA0 transfer counter	TCR0	XX <sub>16</sub>
0029 <sub>16</sub>			XX <sub>16</sub>
002A <sub>16</sub>			
002B <sub>16</sub>			
002C <sub>16</sub>	DMA0 control register	DM0CON	00000X00 <sub>2</sub>
002D <sub>16</sub>			
002E <sub>16</sub>			
002F <sub>16</sub>			
0030 <sub>16</sub>	DMA1 source pointer	SAR1	XX <sub>16</sub>
0031 <sub>16</sub>			XX <sub>16</sub>
0032 <sub>16</sub>			XX <sub>16</sub>
0033 <sub>16</sub>			
0034 <sub>16</sub>	DMA1 destination pointer	DAR1	XX <sub>16</sub>
0035 <sub>16</sub>			XX <sub>16</sub>
0036 <sub>16</sub>			XX <sub>16</sub>
0037 <sub>16</sub>			
0038 <sub>16</sub>	DMA1 transfer counter	TCR1	XX <sub>16</sub>
0039 <sub>16</sub>			XX <sub>16</sub>
003A <sub>16</sub>			
003B <sub>16</sub>			
003C <sub>16</sub>	DMA1 control register	DM1CON	00000X00 <sub>2</sub>
003D <sub>16</sub>			
003E <sub>16</sub>			
003F <sub>16</sub>			

## NOTES:

1. The blank spaces are reserved. No access is allowed.
2. Bits CM27, CM21, and CM20 do not change at oscillation stop detection reset.
3. The VCR1 and VCR2 registers do not change at software reset, watchdog timer reset, and oscillation stop detection reset.
4. Registers VCR1, VCR2, and D4INT cannot be used in M16C/26T.
5. M16C/26A, M16C/26B

X : Undefined

Table 4.2 SFR Information(2)(1)

Address	Register	Symbol	After reset
0040 <sub>16</sub>			
0041 <sub>16</sub>			
0042 <sub>16</sub>			
0043 <sub>16</sub>			
0044 <sub>16</sub>	INT3 interrupt control register	INT3IC	XX00X000 <sub>2</sub>
0045 <sub>16</sub>			
0046 <sub>16</sub>			
0047 <sub>16</sub>			
0048 <sub>16</sub>	INT5 interrupt control register	INT5IC	XX00X000 <sub>2</sub>
0049 <sub>16</sub>	INT4 interrupt control register	INT4IC	XX00X000 <sub>2</sub>
004A <sub>16</sub>	UART2 Bus collision detection interrupt control register	BCNIC	XXXXX000 <sub>2</sub>
004B <sub>16</sub>	DMA0 interrupt control register	DM0IC	XXXXX000 <sub>2</sub>
004C <sub>16</sub>	DMA1 interrupt control register	DM1IC	XXXXX000 <sub>2</sub>
004D <sub>16</sub>	Key input interrupt control register	KUPIC	XXXXX000 <sub>2</sub>
004E <sub>16</sub>	A/D conversion interrupt control register	ADIC	XXXXX000 <sub>2</sub>
004F <sub>16</sub>	UART2 transmit interrupt control register	S2TIC	XXXXX000 <sub>2</sub>
0050 <sub>16</sub>	UART2 receive interrupt control register	S2RIC	XXXXX000 <sub>2</sub>
0051 <sub>16</sub>	UART0 transmit interrupt control register	S0TIC	XXXXX000 <sub>2</sub>
0052 <sub>16</sub>	UART0 receive interrupt control register	S0RIC	XXXXX000 <sub>2</sub>
0053 <sub>16</sub>	UART1 transmit interrupt control register	S1TIC	XXXXX000 <sub>2</sub>
0054 <sub>16</sub>	UART1 receive interrupt control register	S1RIC	XXXXX000 <sub>2</sub>
0055 <sub>16</sub>	TimerA0 interrupt control register	TA0IC	XXXXX000 <sub>2</sub>
0056 <sub>16</sub>	TimerA1 interrupt control register	TA1IC	XXXXX000 <sub>2</sub>
0057 <sub>16</sub>	TimerA2 interrupt control register	TA2IC	XXXXX000 <sub>2</sub>
0058 <sub>16</sub>	TimerA3 interrupt control register	TA3IC	XXXXX000 <sub>2</sub>
0059 <sub>16</sub>	TimerA4 interrupt control register	TA4IC	XXXXX000 <sub>2</sub>
005A <sub>16</sub>	TimerB0 interrupt control register	TB0IC	XXXXX000 <sub>2</sub>
005B <sub>16</sub>	TimerB1 interrupt control register	TB1IC	XXXXX000 <sub>2</sub>
005C <sub>16</sub>	TimerB2 interrupt control register	TB2IC	XXXXX000 <sub>2</sub>
005D <sub>16</sub>	INT0 interrupt control register	INT0IC	XX00X000 <sub>2</sub>
005E <sub>16</sub>	INT1 interrupt control register	INT1IC	XX00X000 <sub>2</sub>
005F <sub>16</sub>	INT2 interrupt control register	INT2IC	XX00X000 <sub>2</sub>
0060 <sub>16</sub>			
0061 <sub>16</sub>			
0062 <sub>16</sub>			
0063 <sub>16</sub>			
0064 <sub>16</sub>			
0065 <sub>16</sub>			
0066 <sub>16</sub>			
0067 <sub>16</sub>			
0068 <sub>16</sub>			
0069 <sub>16</sub>			
006A <sub>16</sub>			
006B <sub>16</sub>			
006C <sub>16</sub>			
006D <sub>16</sub>			
006E <sub>16</sub>			
006F <sub>16</sub>			
0070 <sub>16</sub>			
0071 <sub>16</sub>			
0072 <sub>16</sub>			
0073 <sub>16</sub>			
0074 <sub>16</sub>			
0075 <sub>16</sub>			
0076 <sub>16</sub>			
0077 <sub>16</sub>			
0078 <sub>16</sub>			
0079 <sub>16</sub>			
007A <sub>16</sub>			
007B <sub>16</sub>			
007C <sub>16</sub>			
007D <sub>16</sub>			
007E <sub>16</sub>			
007F <sub>16</sub>			

## NOTE:

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

X: Undefined

Table 4.3 SFR Information(3)(1)

Address	Register	Symbol	After reset
0080 <sub>16</sub>			
0081 <sub>16</sub>			
0082 <sub>16</sub>			
0083 <sub>16</sub>			
0084 <sub>16</sub>			
0085 <sub>16</sub>			
0086 <sub>16</sub>			
⋮			⋮
01B0 <sub>16</sub>			
01B1 <sub>16</sub>			
01B2 <sub>16</sub>			
01B3 <sub>16</sub>	Flash memory control register 4 (2)	FMR4	01000000 <sub>2</sub>
01B4 <sub>16</sub>			
01B5 <sub>16</sub>	Flash memory control register 1 (2)	FMR1	000XX0X <sub>2</sub>
01B6 <sub>16</sub>			
01B7 <sub>16</sub>	Flash memory control register 0 (2)	FMR0	01 <sub>16</sub>
01B8 <sub>16</sub>			
01B9 <sub>16</sub>			
01BA <sub>16</sub>			
01BB <sub>16</sub>			
01BC <sub>16</sub>			
01BD <sub>16</sub>			
01BE <sub>16</sub>			
01BF <sub>16</sub>			
⋮			⋮
0250 <sub>16</sub>			
0251 <sub>16</sub>			
0252 <sub>16</sub>			
0253 <sub>16</sub>			
0254 <sub>16</sub>			
0255 <sub>16</sub>			
0256 <sub>16</sub>			
0257 <sub>16</sub>			
0258 <sub>16</sub>			
0259 <sub>16</sub>			
025A <sub>16</sub>	Three phase protect control register	TPRC	00 <sub>16</sub>
025B <sub>16</sub>			
025C <sub>16</sub>	On-chip oscillator control register	ROCR	X000010 <sub>12</sub>
025D <sub>16</sub>	Pin assignment control register	PACR	00 <sub>16</sub>
025E <sub>16</sub>	Peripheral clock select register	PCLKR	0000001 <sub>12</sub>
025F <sub>16</sub>			
⋮			⋮
0330 <sub>16</sub>			
0331 <sub>16</sub>			
0332 <sub>16</sub>			
0333 <sub>16</sub>			
0334 <sub>16</sub>			
0335 <sub>16</sub>			
0336 <sub>16</sub>			
0337 <sub>16</sub>			
0338 <sub>16</sub>			
0339 <sub>16</sub>			
033A <sub>16</sub>			
033B <sub>16</sub>			
033C <sub>16</sub>			
033D <sub>16</sub>			
033E <sub>16</sub>	NMI digital debounce register	NDDR	FF <sub>16</sub>
033F <sub>16</sub>	Port17 digital debounce register	P17DDR	FF <sub>16</sub>

## NOTES:

- Blank spaces are reserved. No access is allowed.
- This register is included in the flash memory version.

X: Undefined

Table 4.4 SFR Information(4)(1)

Address	Register	Symbol	After reset
0340 <sub>16</sub>			
0341 <sub>16</sub>			
0342 <sub>16</sub>	Timer A1-1 register	TA11	XX <sub>16</sub>
0343 <sub>16</sub>			XX <sub>16</sub>
0344 <sub>16</sub>	Timer A2-1 register	TA21	XX <sub>16</sub>
0345 <sub>16</sub>			XX <sub>16</sub>
0346 <sub>16</sub>	Timer A4-1 register	TA41	XX <sub>16</sub>
0347 <sub>16</sub>			XX <sub>16</sub>
0348 <sub>16</sub>	Three phase PWM control register 0	INVC0	00 <sub>16</sub>
0349 <sub>16</sub>	Three phase PWM control register 1	INVC1	00 <sub>16</sub>
034A <sub>16</sub>	Three phase output buffer register 0	IDB0	00111111 <sub>2</sub>
034B <sub>16</sub>	Three phase output buffer register 1	IDB1	00111111 <sub>2</sub>
034C <sub>16</sub>	Dead time timer	DTT	XX <sub>16</sub>
034D <sub>16</sub>	Timer B2 Interrupt occurrence frequency set counter	ICTB2	XX <sub>16</sub>
034E <sub>16</sub>	Position-data-retain function control register	PDRF	XXXX0000 <sub>2</sub>
034F <sub>16</sub>			
0350 <sub>16</sub>			
0351 <sub>16</sub>			
0352 <sub>16</sub>			
0353 <sub>16</sub>			
0354 <sub>16</sub>			
0355 <sub>16</sub>			
0356 <sub>16</sub>			
0357 <sub>16</sub>			
0358 <sub>16</sub>	Port function control register	PFCR	00111111 <sub>2</sub>
0359 <sub>16</sub>			
035A <sub>16</sub>			
035B <sub>16</sub>			
035C <sub>16</sub>			
035D <sub>16</sub>			
035E <sub>16</sub>	Interrupt request cause select register 2	IFSR2A	XXXXXXX <sub>2</sub> (2)
035F <sub>16</sub>	Interrupt request cause select register	IFSR	00 <sub>16</sub>
0360 <sub>16</sub>			
0361 <sub>16</sub>			
0362 <sub>16</sub>			
0363 <sub>16</sub>			
0364 <sub>16</sub>			
0365 <sub>16</sub>			
0366 <sub>16</sub>			
0367 <sub>16</sub>			
0368 <sub>16</sub>			
0369 <sub>16</sub>			
036A <sub>16</sub>			
036B <sub>16</sub>			
036C <sub>16</sub>			
036D <sub>16</sub>			
036E <sub>16</sub>			
036F <sub>16</sub>			
0370 <sub>16</sub>			
0371 <sub>16</sub>			
0372 <sub>16</sub>			
0373 <sub>16</sub>			
0374 <sub>16</sub>	UART2 special mode register 4	U2SMR4	00 <sub>16</sub>
0375 <sub>16</sub>	UART2 special mode register 3	U2SMR3	000X0X <sub>2</sub>
0376 <sub>16</sub>	UART2 special mode register 2	U2SMR2	X0000000 <sub>2</sub>
0377 <sub>16</sub>	UART2 special mode register	U2SMR	X0000000 <sub>2</sub>
0378 <sub>16</sub>	UART2 transmit/receive mode register	U2MR	00 <sub>16</sub>
0379 <sub>16</sub>	UART2 bit rate register	U2BRG	XX <sub>16</sub>
037A <sub>16</sub>	UART2 transmit buffer register	U2TB	XXXXXXXX <sub>2</sub>
037B <sub>16</sub>			XXXXXXXX <sub>2</sub>
037C <sub>16</sub>	UART2 transmit/receive control register 0	U2C0	00001000 <sub>2</sub>
037D <sub>16</sub>	UART2 transmit/receive control register 1	U2C1	00000010 <sub>2</sub>
037E <sub>16</sub>	UART2 receive buffer register	U2RB	XXXXXXXX <sub>2</sub>
037F <sub>16</sub>			XXXXXXXX <sub>2</sub>

## NOTE:

- Blank spaces are reserved. No access is allowed.
- Write "1" to bit 0 after reset.

X : Undefined



Table 4.5 SFR Information(5)<sup>(1)</sup>

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

## NOTE:

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

X : Undefined

Table 4.6 SFR Information(6)<sup>(1)</sup>

Address	Register	Symbol	After Reset
03C0 <sub>16</sub> 03C1 <sub>16</sub>	A/D register 0	AD0	XXXXXXXX <sub>2</sub> XXXXXXXX <sub>2</sub>
03C2 <sub>16</sub> 03C3 <sub>16</sub>	A/D register 1	AD1	XXXXXXXX <sub>2</sub> XXXXXXXX <sub>2</sub>
03C4 <sub>16</sub> 03C5 <sub>16</sub>	A/D register 2	AD2	XXXXXXXX <sub>2</sub> XXXXXXXX <sub>2</sub>
03C6 <sub>16</sub> 03C7 <sub>16</sub>	A/D register 3	AD3	XXXXXXXX <sub>2</sub> XXXXXXXX <sub>2</sub>
03C8 <sub>16</sub> 03C9 <sub>16</sub>	A/D register 4	AD4	XXXXXXXX <sub>2</sub> XXXXXXXX <sub>2</sub>
03CA <sub>16</sub> 03CB <sub>16</sub>	A/D register 5	AD5	XXXXXXXX <sub>2</sub> XXXXXXXX <sub>2</sub>
03CC <sub>16</sub> 03CD <sub>16</sub>	A/D register 6	AD6	XXXXXXXX <sub>2</sub> XXXXXXXX <sub>2</sub>
03CE <sub>16</sub> 03CF <sub>16</sub>	A/D register 7	AD7	XXXXXXXX <sub>2</sub> XXXXXXXX <sub>2</sub>
03D0 <sub>16</sub>			
03D1 <sub>16</sub>			
03D2 <sub>16</sub>	A/D trigger control register	ADTRGCON	00 <sub>16</sub>
03D3 <sub>16</sub>	A/D status register 0	ADSTAT0	00000X00 <sub>2</sub>
03D4 <sub>16</sub> 03D5 <sub>16</sub>	A/D control register 2	ADCON2	00 <sub>16</sub>
03D6 <sub>16</sub>	A/D control register 0	ADCON0	00000XX <sub>2</sub>
03D7 <sub>16</sub> 03D8 <sub>16</sub>	A/D control register 1	ADCON1	00 <sub>16</sub>
03D9 <sub>16</sub>			
03DA <sub>16</sub>			
03DB <sub>16</sub>			
03DC <sub>16</sub>			
03DD <sub>16</sub>			
03DE <sub>16</sub>			
03DF <sub>16</sub>			
03E0 <sub>16</sub>			
03E1 <sub>16</sub>	Port P1 register	P1	XX <sub>16</sub>
03E2 <sub>16</sub>			
03E3 <sub>16</sub>	Port P1 direction register	PD1	00 <sub>16</sub>
03E4 <sub>16</sub>			
03E5 <sub>16</sub>			
03E6 <sub>16</sub>			
03E7 <sub>16</sub>			
03E8 <sub>16</sub>			
03E9 <sub>16</sub>			
03EA <sub>16</sub>			
03EB <sub>16</sub>			
03EC <sub>16</sub>	Port P6 register	P6	XX <sub>16</sub>
03ED <sub>16</sub>	Port P7 register	P7	XX <sub>16</sub>
03EE <sub>16</sub>	Port P6 direction register	PD6	00 <sub>16</sub>
03EF <sub>16</sub>	Port P7 direction register	PD7	00 <sub>16</sub>
03F0 <sub>16</sub>	Port P8 register	P8	XX <sub>16</sub>
03F1 <sub>16</sub>	Port P9 register	P9	XXXXXXXX <sub>2</sub>
03F2 <sub>16</sub>	Port P8 direction register	PD8	00 <sub>16</sub>
03F3 <sub>16</sub>	Port P9 direction register	PD9	XXXX0000 <sub>2</sub>
03F4 <sub>16</sub> 03F5 <sub>16</sub>	Port P10 register	P10	XX <sub>16</sub>
03F6 <sub>16</sub>	Port P10 direction register	PD10	00 <sub>16</sub>
03F7 <sub>16</sub>			
03F8 <sub>16</sub>			
03F9 <sub>16</sub>			
03FA <sub>16</sub>			
03FB <sub>16</sub>			
03FC <sub>16</sub>	Pull-up control register 0	PUR0	00 <sub>16</sub>
03FD <sub>16</sub>	Pull-up control register 1	PUR1	00 <sub>16</sub>
03FE <sub>16</sub>	Pull-up control register 2	PUR2	00 <sub>16</sub>
03FF <sub>16</sub>	Port control register	PCR	00 <sub>16</sub>

## NOTE:

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

X: Undefined

## 5. Reset

There are four types of resets: a hardware reset, a software reset, an watchdog timer reset, and an oscillation stop detection reset.

### 5.1 Hardware Reset

There are two types of hardware resets: a hardware reset 1 and a hardware reset 2.

#### 5.1.1 Hardware Reset 1

A reset is applied using the  $\overline{\text{RESET}}$  pin. When an “L” signal is applied to the  $\overline{\text{RESET}}$  pin while the power supply voltage is within the recommended operating condition, the pins are initialized (see Table 5.1.1.1 Pin Status When  $\overline{\text{RESET}}$  Pin Level is “L”). The internal on-chip oscillator is initialized and used as CPU clock.

When the input level at the  $\overline{\text{RESET}}$  pin is released from “L” to “H”, the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. The internal RAM is not initialized. If the  $\overline{\text{RESET}}$  pin is pulled “L” while writing to the internal RAM, the internal RAM becomes indeterminate.

Figure 5.1.1.1 shows the example reset circuit. Figure 5.1.1.2 shows the reset sequence. Table 5.1.1.1 shows the status of the other pins while the  $\overline{\text{RESET}}$  pin is “L”. Figure 5.1.1.3 shows the CPU register status after reset. Refer to “SFR Map” for SFR status after reset.

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

2. Power on
  - (1) Apply an “L” signal to the  $\overline{\text{RESET}}$  pin.
  - (2) Let the power supply voltage increase until it meets the recommended operating condition.
  - (3) Wait  $t_d(\text{P-R})$  or more until the internal power supply stabilizes.
  - (4) Wait  $t_d(\text{ROC})$  or more.
  - (5) Apply an “H” signal to the  $\overline{\text{RESET}}$  pin.

#### 5.1.2 Hardware Reset 2

##### Note

**M16C/26T does not use this function.**

This reset is generated by the microcomputer’s internal voltage detection circuit. The voltage detection circuit monitors the voltage supplied to the VCC pin.

If the VC26 bit in the VCR2 register is set to “1” (reset level detection circuit enabled), the microcomputer is reset when the voltage at the VCC input pin drops below  $V_{\text{det3}}$ .

Conversely, when the input voltage at the VCC pin rises to  $V_{\text{det3r}}$  or more, the pins and the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. It takes about  $t_d(\text{S-R})$  before the program starts running after  $V_{\text{det3r}}$  is detected. The initialized pins and registers and the status thereof are the same as in hardware reset 1.

The microcomputer cannot exit stop mode by voltage down detection reset (hardware reset 2).

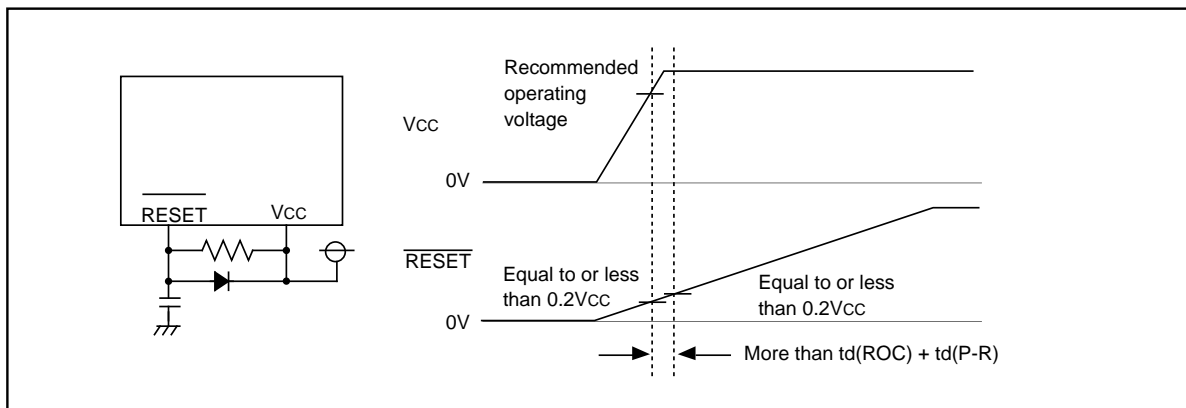


Figure 5.1.1.1. Example Reset Circuit

## 5.2 Software Reset

When the PM03 bit in the PM0 register is set to “1” (microcomputer reset), the microcomputer has its pins, CPU, and SFR initialized. Then the program is executed starting from the address indicated by the reset vector.

The device will reset using on-chip oscillator as the CPU clock.

At software reset, some SFR’s are not initialized. Refer to “SFR”.

## 5.3 Watchdog Timer Reset

When the PM12 bit in the PM1 register is “1” (reset when watchdog timer underflows), the microcomputer initializes its pins, CPU and SFR if the watchdog timer underflows.

The device will reset using on-chip oscillator as the system clock. Then the program is executed starting from the address indicated by the reset vector.

At watchdog timer reset, some SFR’s are not initialized. Refer to “SFR”.

## 5.4 Oscillation Stop Detection Reset

When the CM20 bit in the CM2 register is set to “1” (oscillation stop, re-oscillation detection function enabled) and the CM27 bit is set to “0” (reset at oscillation stop detection), the microcomputer initializes its pins, CPU and SFR, coming to a halt if it detects main clock oscillation circuit stop. Refer to the section “oscillation stop, re-oscillation detection function”.

At oscillation stop detection reset, some SFR’s are not initialized. Refer to the section “SFR”.

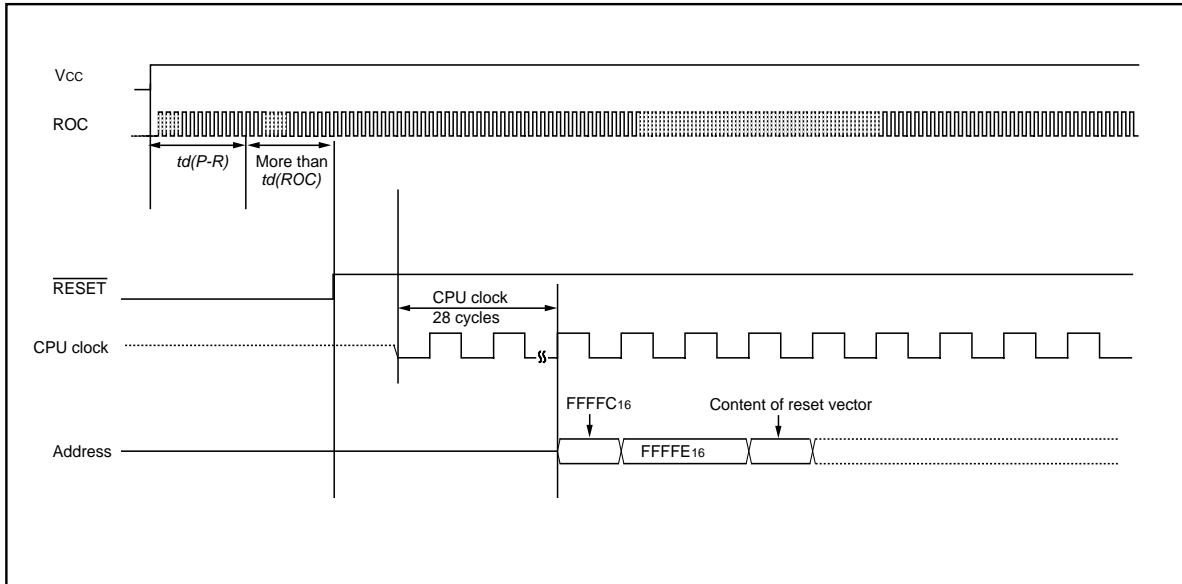


Figure 5.1.1.2. Reset Sequence

Table 5.1.1.1. Pin Status When RESET Pin Level is “L”

Pin name	Status
P1, P6 to P10	Input port (high impedance)

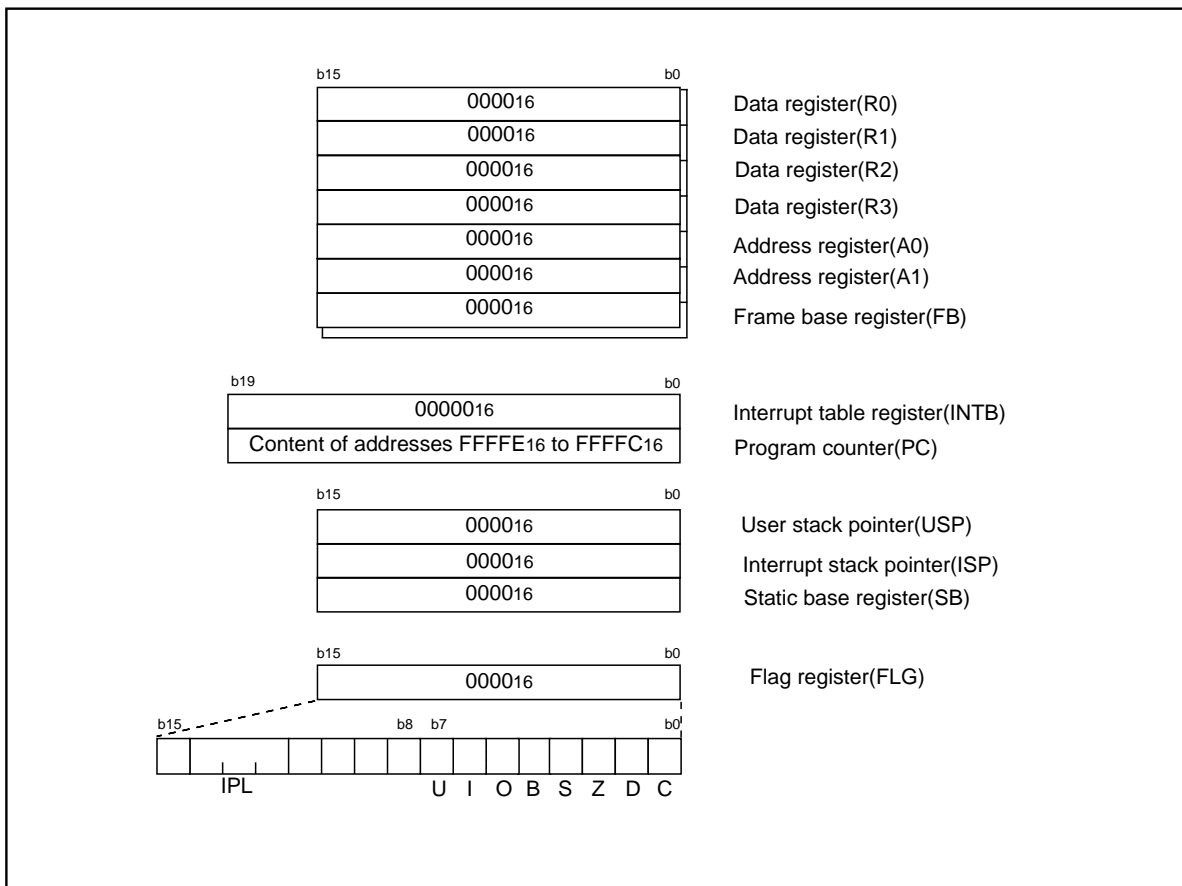


Figure 5.1.1.3. CPU Register Status After Reset

## 5.5 Voltage Detection Circuit

### Note

VCC=5 V is assumed. Voltage Detection Circuit is not available in M16C/26T.

The voltage detection circuit has circuits to monitor the input voltage at the VCC pin, each checking the input voltage with respect to Vdet3, and Vdet4, respectively. Use the VC26 to VC27 bits in the VCR2 register to select whether or not to enable these circuits.

Use the reset level detection circuit for hardware reset 2.

The voltage down detection circuit can be set to detect whether the input voltage is equal to or greater than Vdet4 or less than Vdet4 by monitoring the VC13 bit in the VCR1 register. Furthermore, a voltage down detection interrupt can be generated.

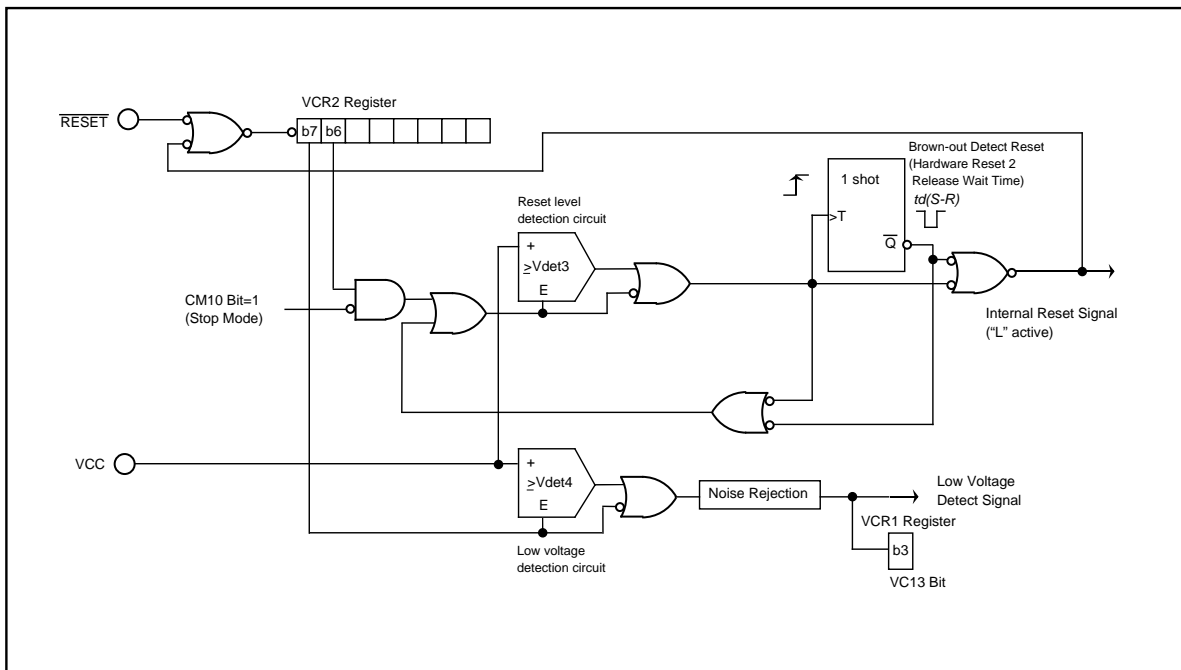


Figure 5.5.1. Voltage Detection Circuit Block

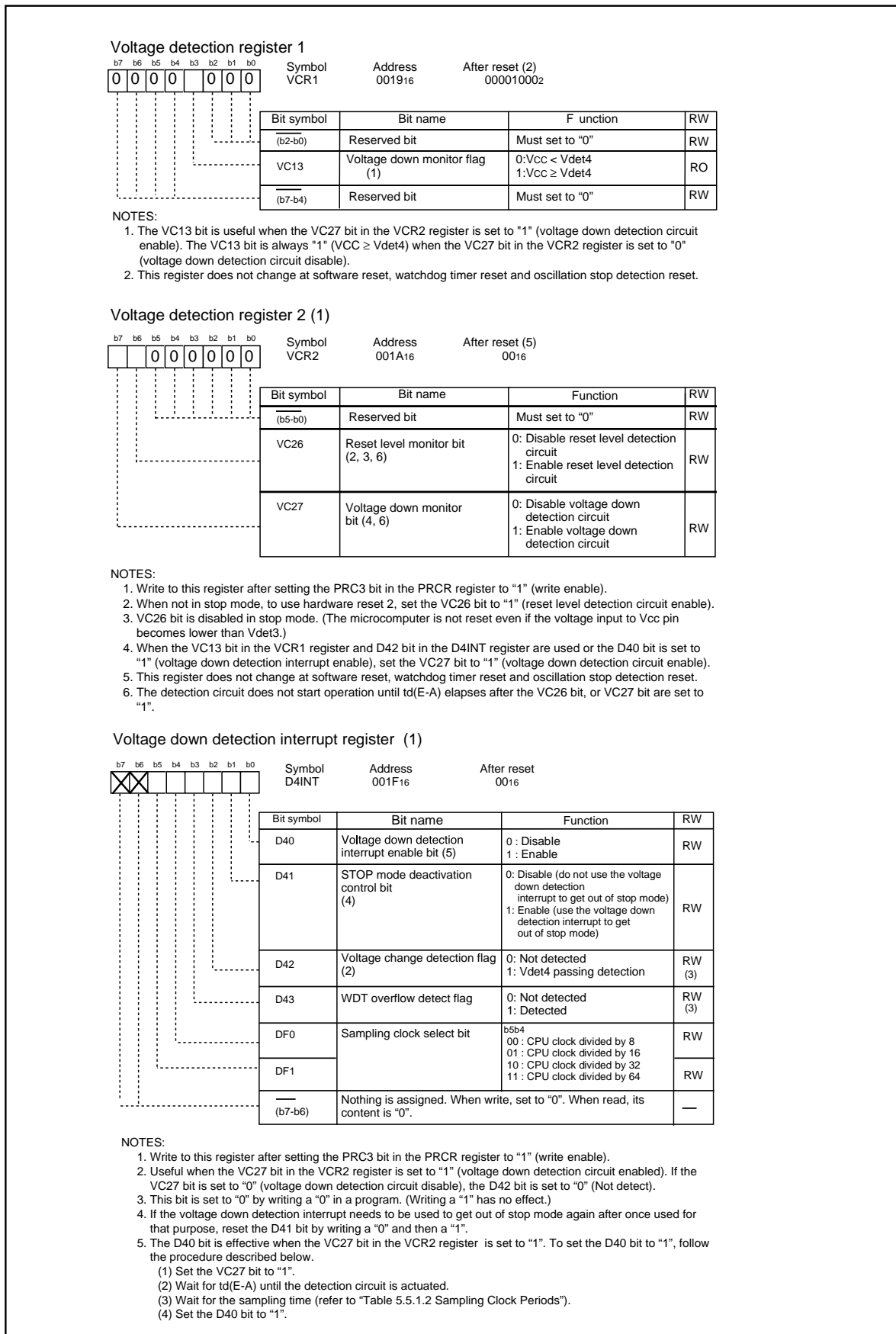


Figure 5.5.2. VCR1 Register, VCR2 Register, and D4INT Register

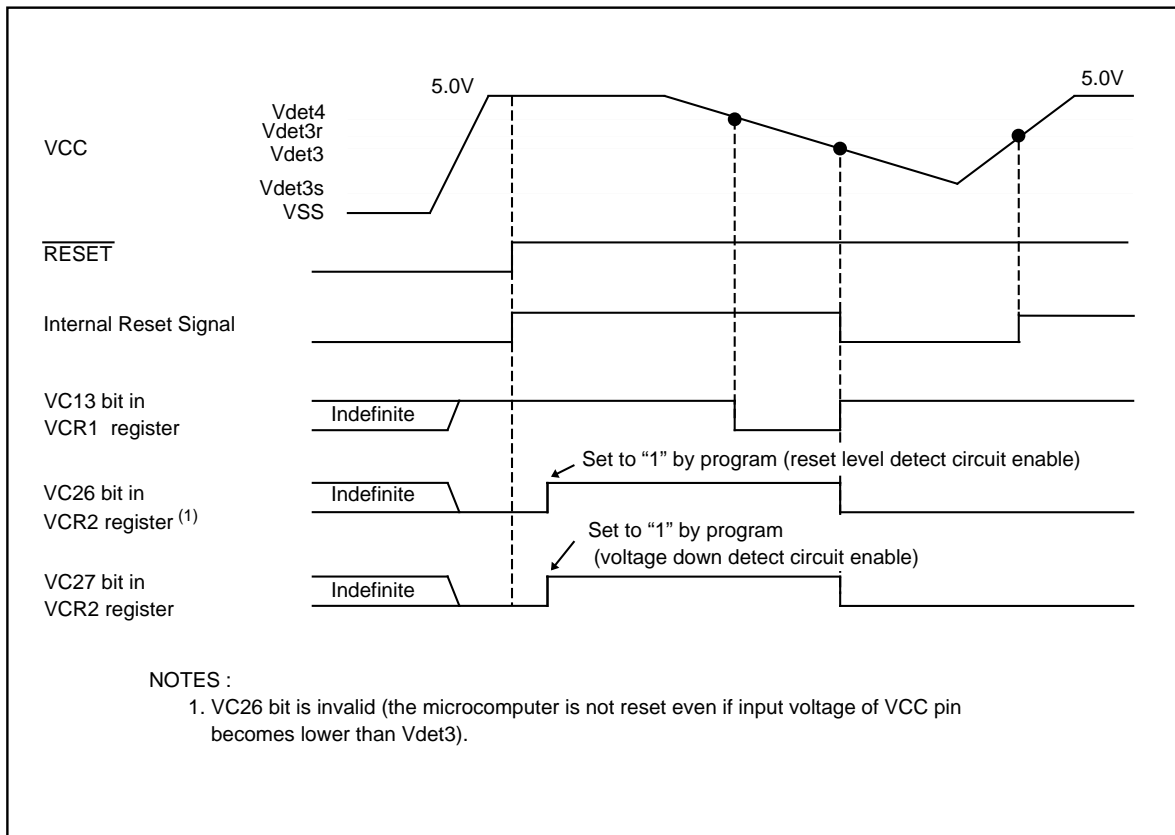


Figure 5.5.3. Typical Operation of Hardware Reset 2



### 5.5.1 Voltage Down Detection Interrupt

If the D40 bit in the D4INT register is set to “1” (voltage down detection interrupt enabled), the voltage down detection interrupt request is generated when the voltage applied to the VCC pin crosses the Vdet4 voltage level. The voltage down detection interrupt shares the same interrupt vector with the watchdog timer interrupt and oscillation stop, re-oscillation detection interrupt.

Set the D41 bit in the D4INT register to “1” (enabled) to use the voltage down detection interrupt to exit stop mode.

The D42 bit in the D4INT register is set to “1” as soon as the voltage applied to the VCC pin reaches Vdet4 due to the voltage rise and voltage drop. When the D42 bit changes “0” to “1”, the voltage down detection interrupt request is generated. Set the D42 bit to “0” by program. However, when the D41 bit is set to “1” and the microcomputer is in stop mode, the voltage down detection interrupt request is generated regardless of the D42 bit state if the voltage applied to the VCC pin is detected to be above Vdet4. The microcomputer then exits stop mode.

Table 5.5.1.1 shows how the voltage down detection interrupt request is generated.

The DF1 to DF0 bits in the D4INT register determine the sampling period that detects the voltage applied to the VCC pin reaches Vdet4. Table 5.5.1.2 shows the sampling periods.

**Table 5.5.1.1 Voltage Down Detection Interrupt Request Generation Conditions**

Operation Mode	VC27 Bit	D40 Bit	D41 Bit	D42 Bit	CM02 Bit	VC13 Bit
Normal Operation Mode <sup>(1)</sup>	1	1	—	0 to 1	—	0 to 1 <sup>(3)</sup>
						1 to 0 <sup>(3)</sup>
Wait Mode <sup>(2)</sup>			—	0 to 1	0	0 to 1 <sup>(3)</sup>
					1	1 to 0 <sup>(3)</sup>
Stop Mode <sup>(2)</sup>			1	—	0	0 to 1
						0 to 1

— : “0” or “1”

**NOTES:**

1. The status except the wait mode and stop mode is handled as the normal mode. (Refer to 7. **Clock generating circuit**)
2. Refer to 5.5.2 **Limitations on stop mode**, 5.5.3 **Limitations on wait mode**.
3. An interrupt request for voltage reduction is generated a sampling time after the value of the VC13 bit has changed. See the **Figure 5.5.1.2 Voltage Down Detection Interrupt Generation Circuit Operation Example** for details.

**Table 5.5.1.2 Sampling Periods**

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

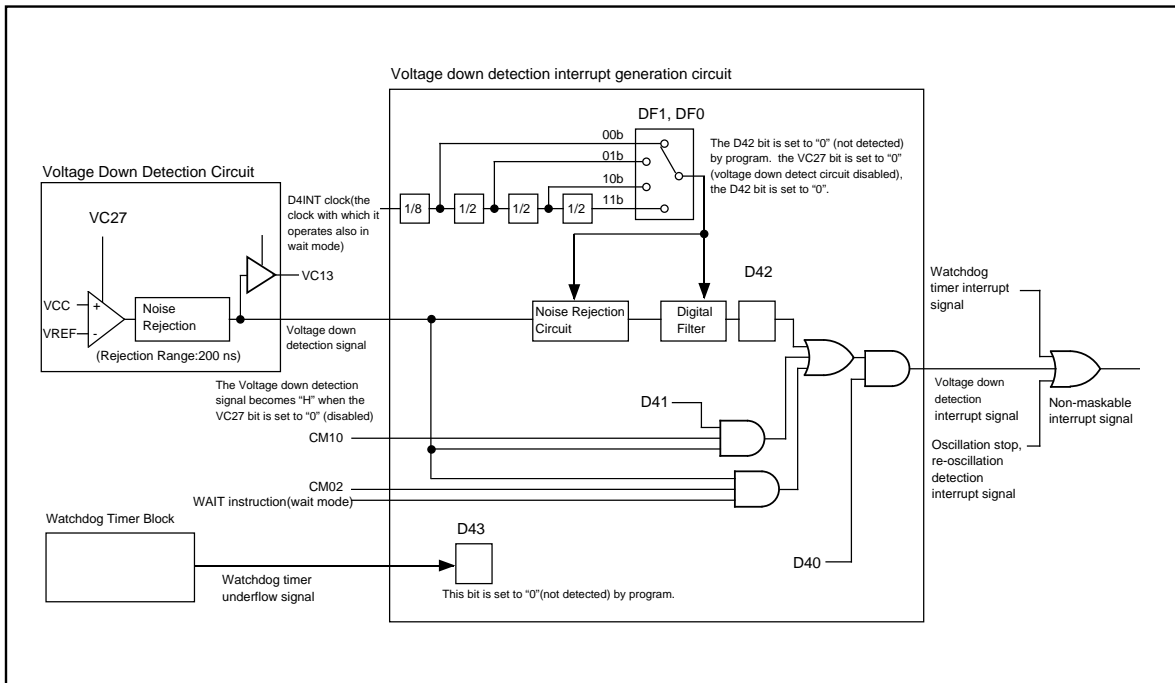


Figure 5.5.1.1 Power Supply Down Detection Interrupt Generation Block

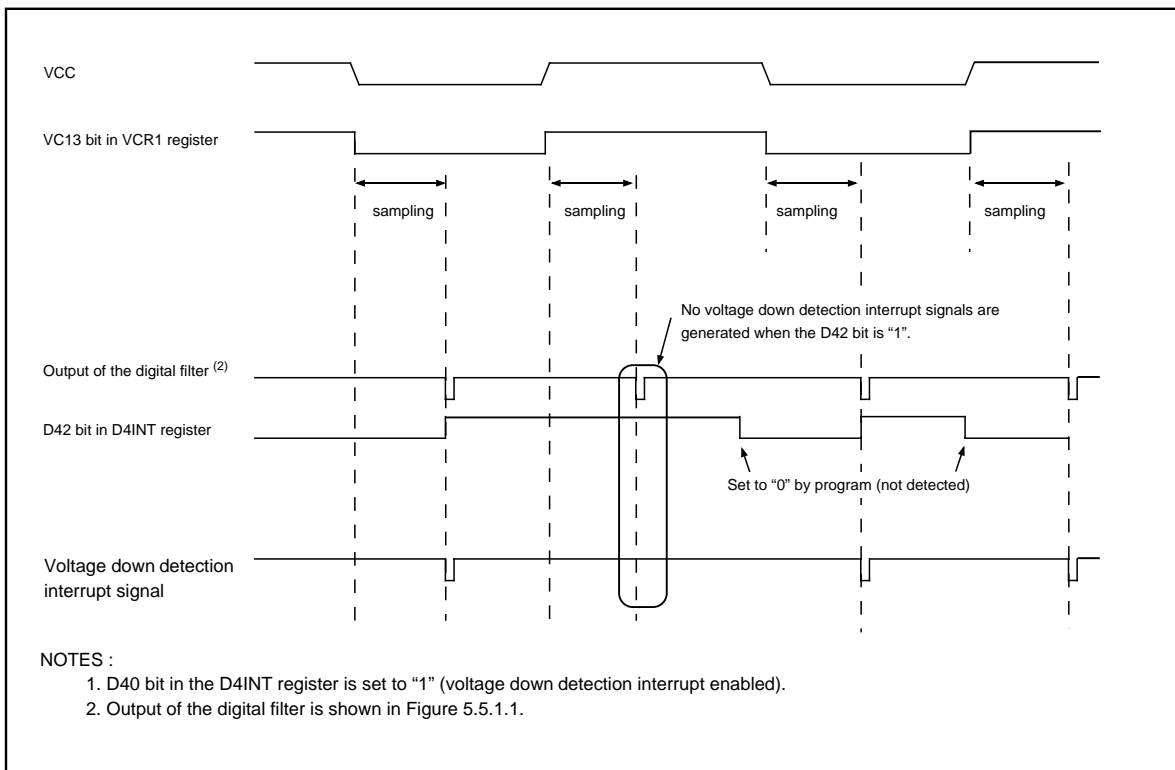


Figure 5.5.1.2 Power Supply Down Detection Interrupt Generation Circuit Operation Example

### 5.5.2 Limitations on Exiting Stop Mode

The voltage down detection interrupt is immediately generated and the microcomputer exits stop mode if the CM10 bit in the CM1 register is set to “1” under the conditions below.

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

If the microcomputer is set to enter stop mode when the voltage applied to the VCC pin drops below Vdet4 and to exit stop mode when the voltage applied rises to Vdet4 or above, set the CM10 bit to “1” when VC13 bit is “0” ( $VCC < Vdet4$ ).

### 5.5.3 Limitations on Exiting Wait Mode

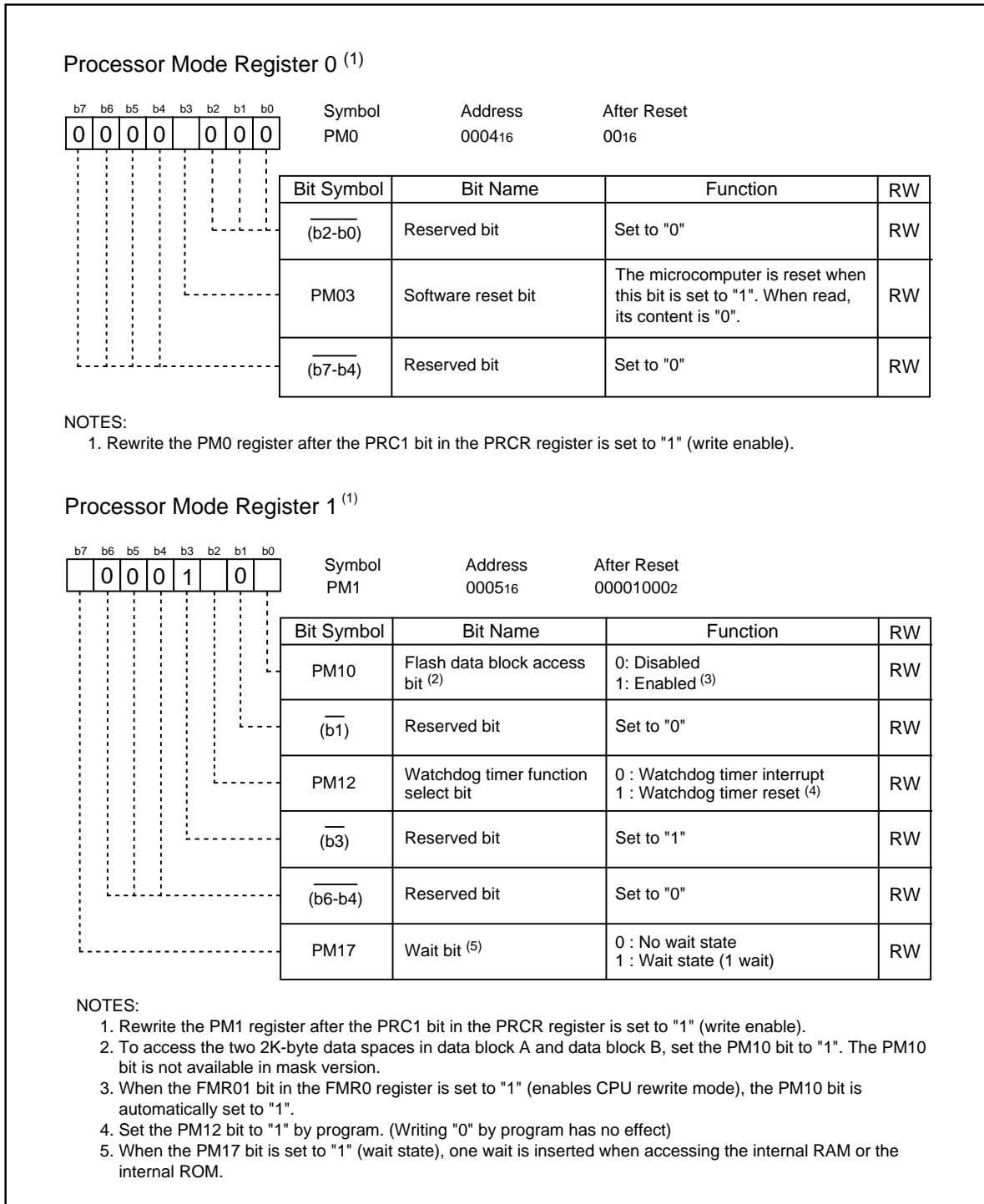
The voltage down detection interrupt is immediately generated and the microcomputer exits wait mode if WAIT instruction is executed under the conditions below.

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

If the microcomputer is set to enter wait mode when the voltage applied to the VCC pin drops below Vdet4 and to exit wait mode when the voltage applied rises to Vdet4 or above, perform WAIT instruction when VC13 bit is “0” ( $VCC < Vdet4$ ).

## 6. Processor Mode

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



**Figure 6.1 PM0 Register, PM1 Register**

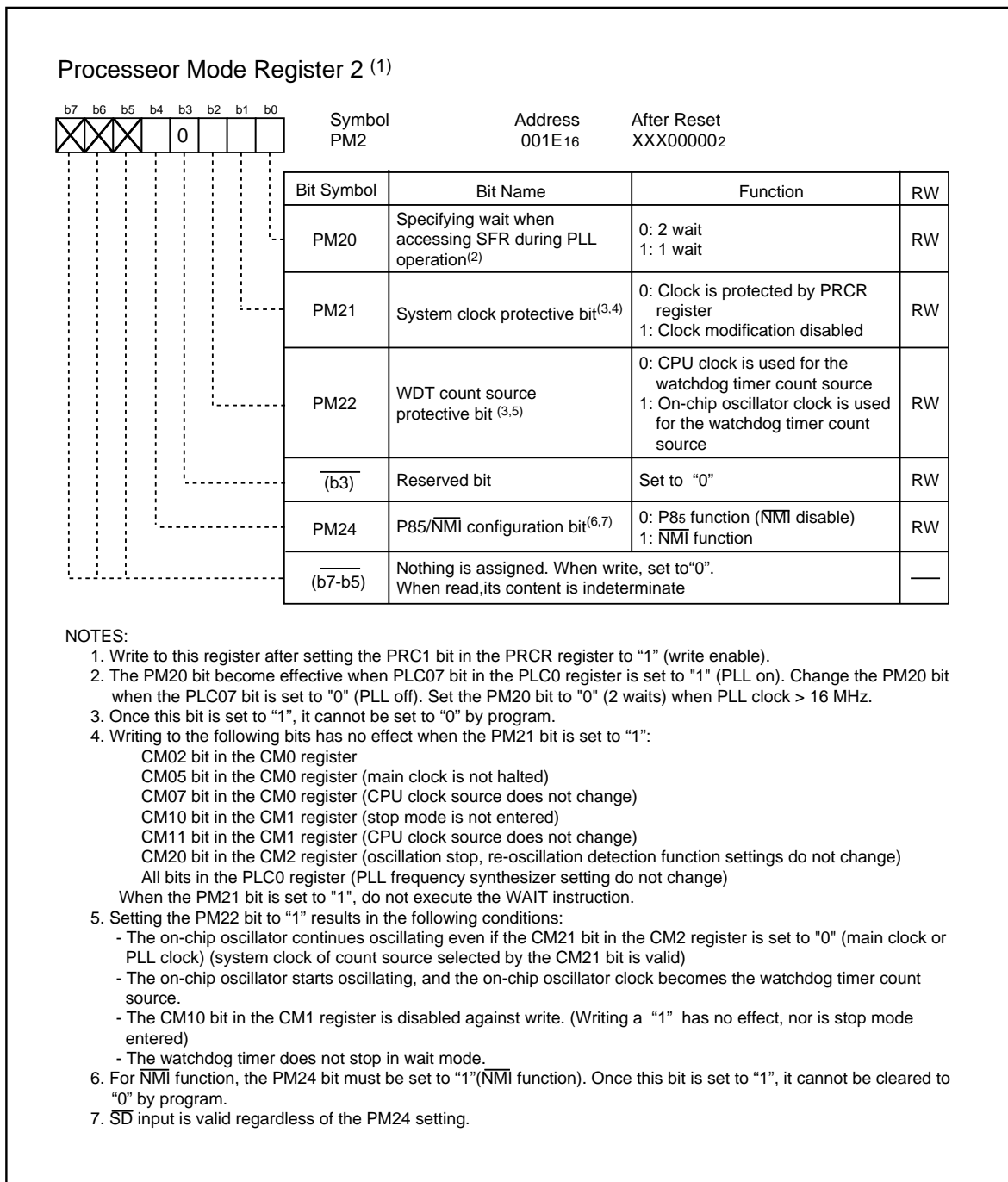
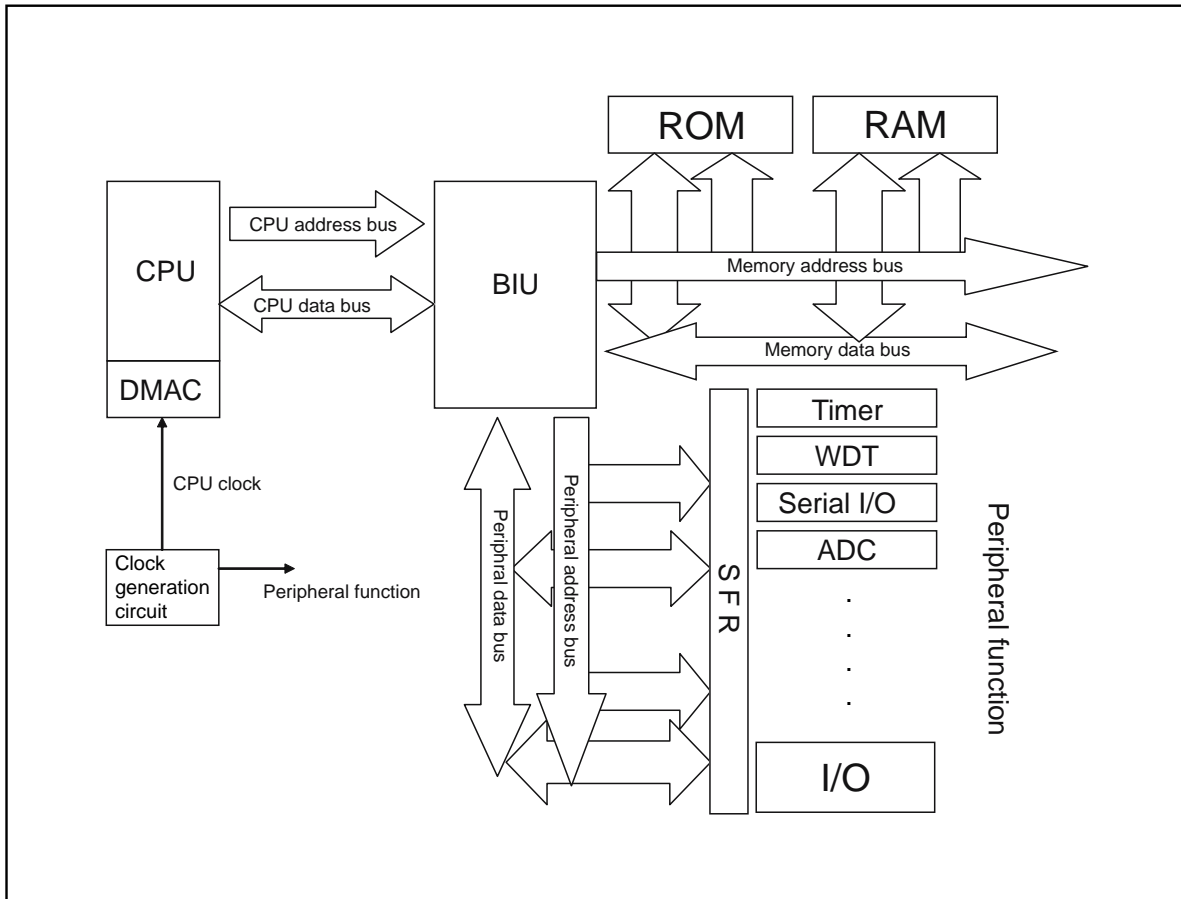


Figure 6.2 PM2 Register

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



**Figure 6.3 Bus Block Diagram**

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

**Table 6.1 Accessible Area and Bus Cycle**

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

## 7. Clock Generation Circuit

The clock generation circuit contains four oscillator circuits as follows:

- (1) Main clock oscillation circuit
- (2) Sub clock oscillation circuit
- (3) On-chip oscillator (available at reset, oscillation stop detect function)
- (4) PLL frequency synthesizer

Table 7.1 lists the clock generation circuit specifications. Figure 7.1 shows the clock generation circuit. Figures 7.2 to 7.6 show the clock-related registers.

**Table 7.1. 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"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Timer A, B's clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> <li>• CPU and peripheral function clock sources when the main clock stops oscillating</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>
Clock frequency	0 to 20 MHz	32.768 kHz	<ul style="list-style-type: none"> <li>• Selectable source frequency: f<sub>1</sub>(ROC), f<sub>2</sub>(ROC), f<sub>3</sub>(ROC)</li> <li>• Selectable divider: by 2, by 4, by 8</li> </ul>	10 to 20 MHz (M16C/26A) 10 to 24 MHz (M16C/26B)
Usable oscillator	<ul style="list-style-type: none"> <li>• Ceramic oscillator</li> <li>• Crystal oscillator</li> </ul>	<ul style="list-style-type: none"> <li>• Crystal oscillator</li> </ul>	—	—
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	—	—
Oscillation stop, restart function	Available	Available	Available	Available
Oscillator status after reset	Oscillating (M16C/26A) Stopped (M16C/26B) Stopped (M16C/26T)	Stopped	Oscillating (CPU clock source)	Stopped
Other	Externally derived clock can be input		—	—

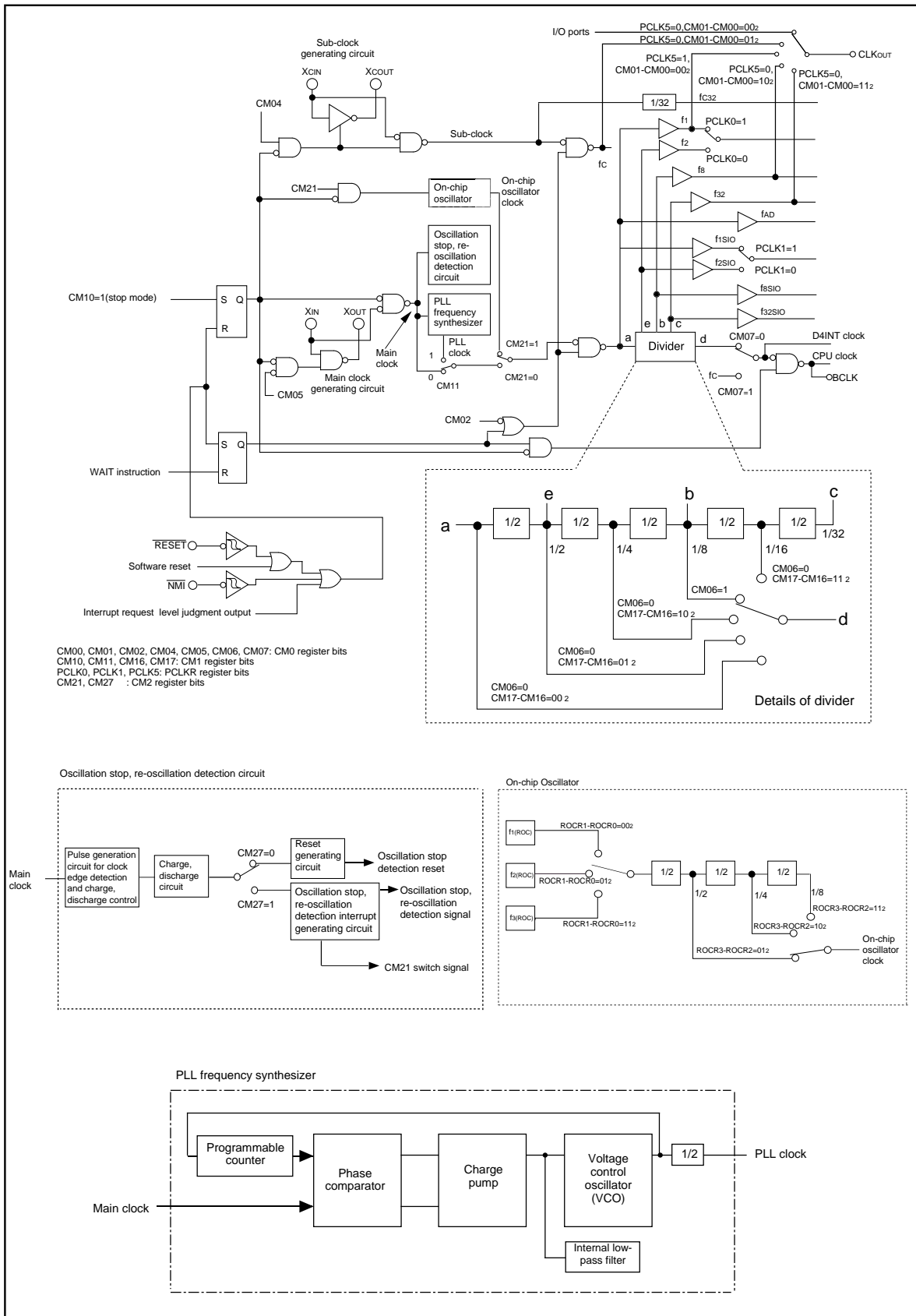


Figure 7.1. Clock Generation Circuit



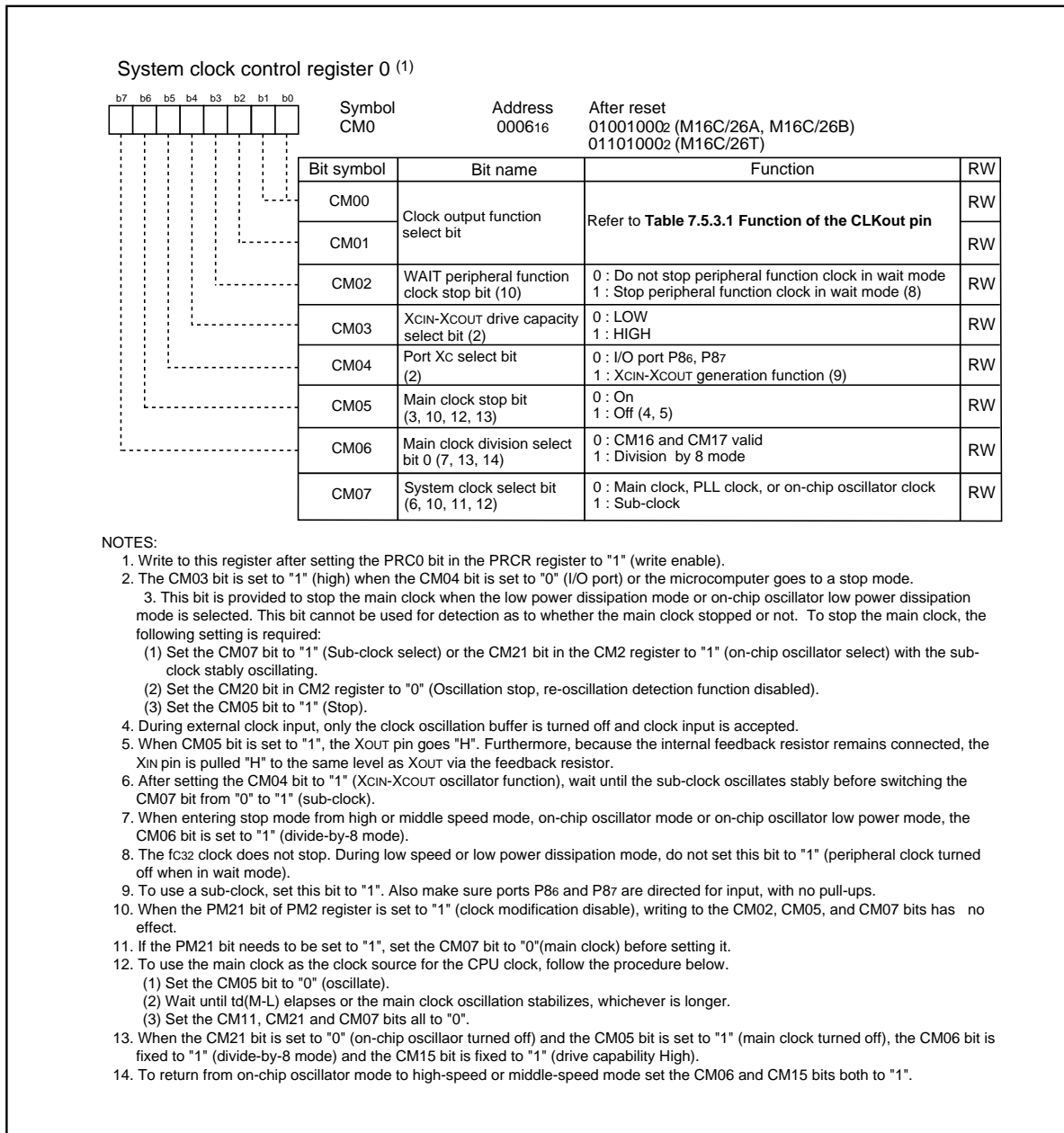


Figure 7.2. CM0 Register

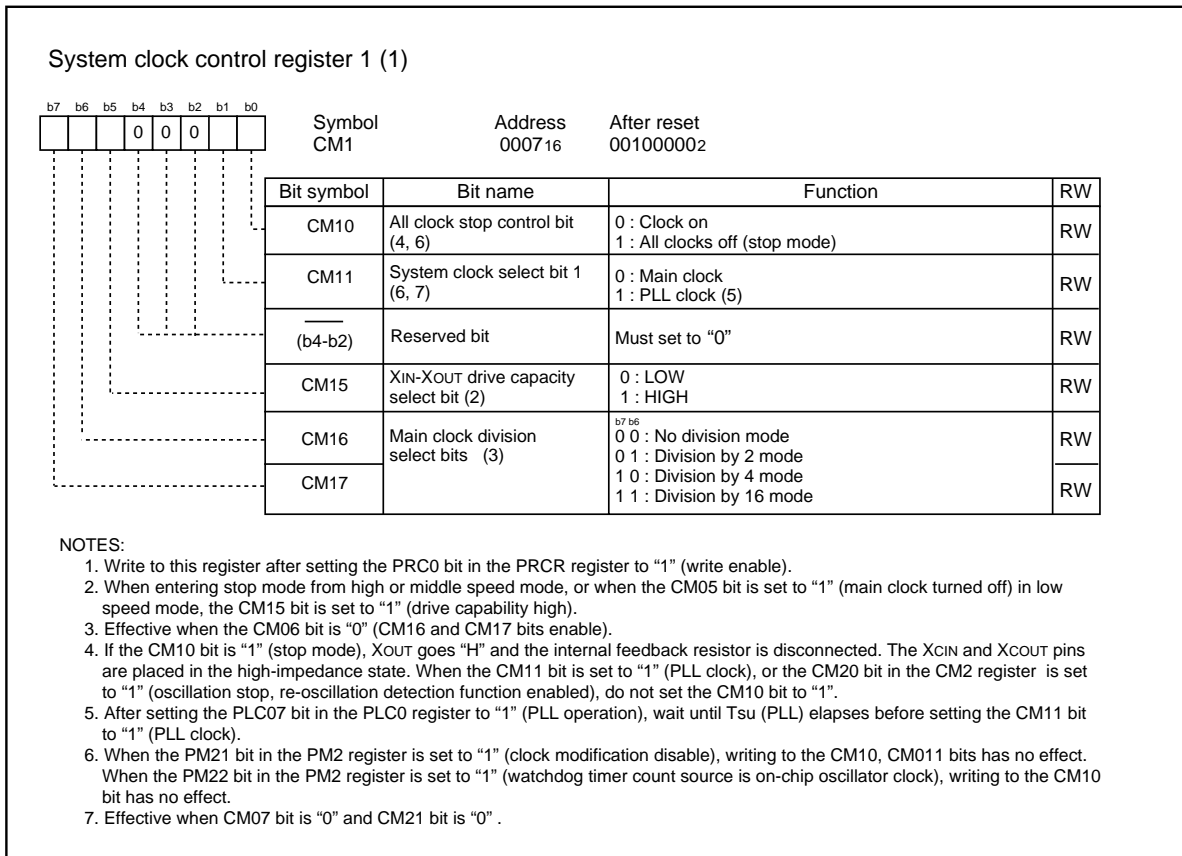


Figure 7.3. CM1 Register

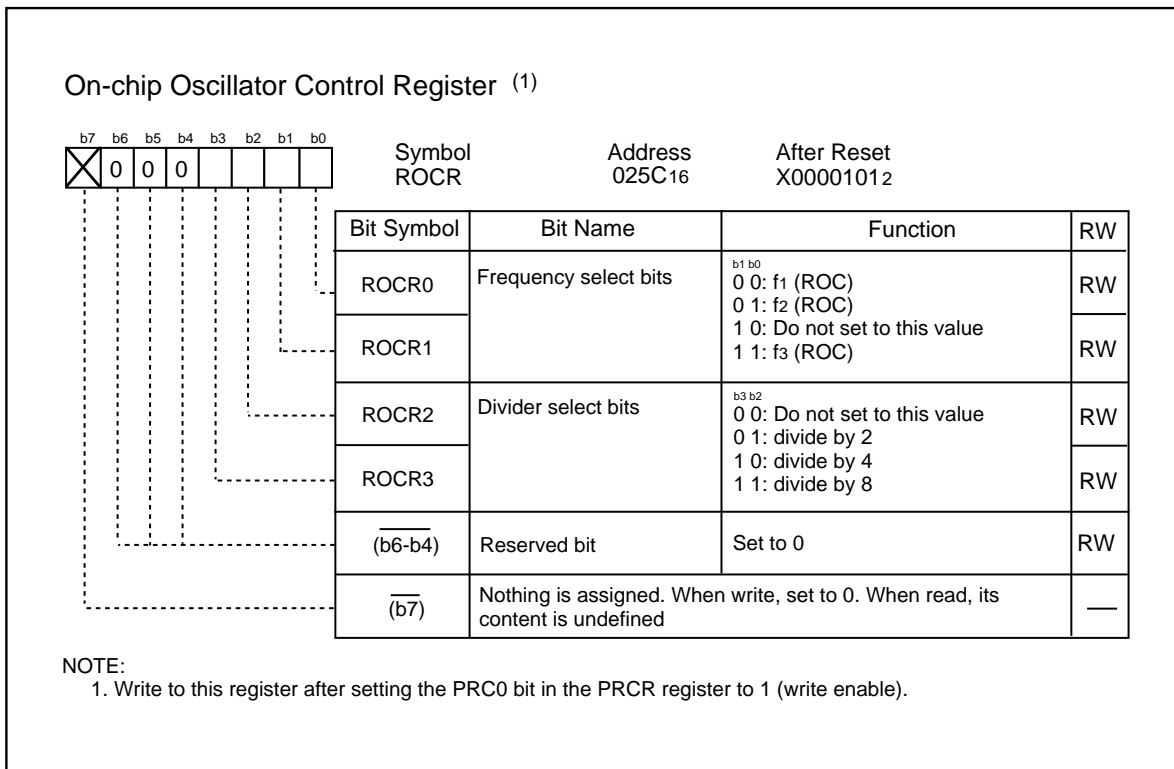


Figure 7.4. ROCR Register

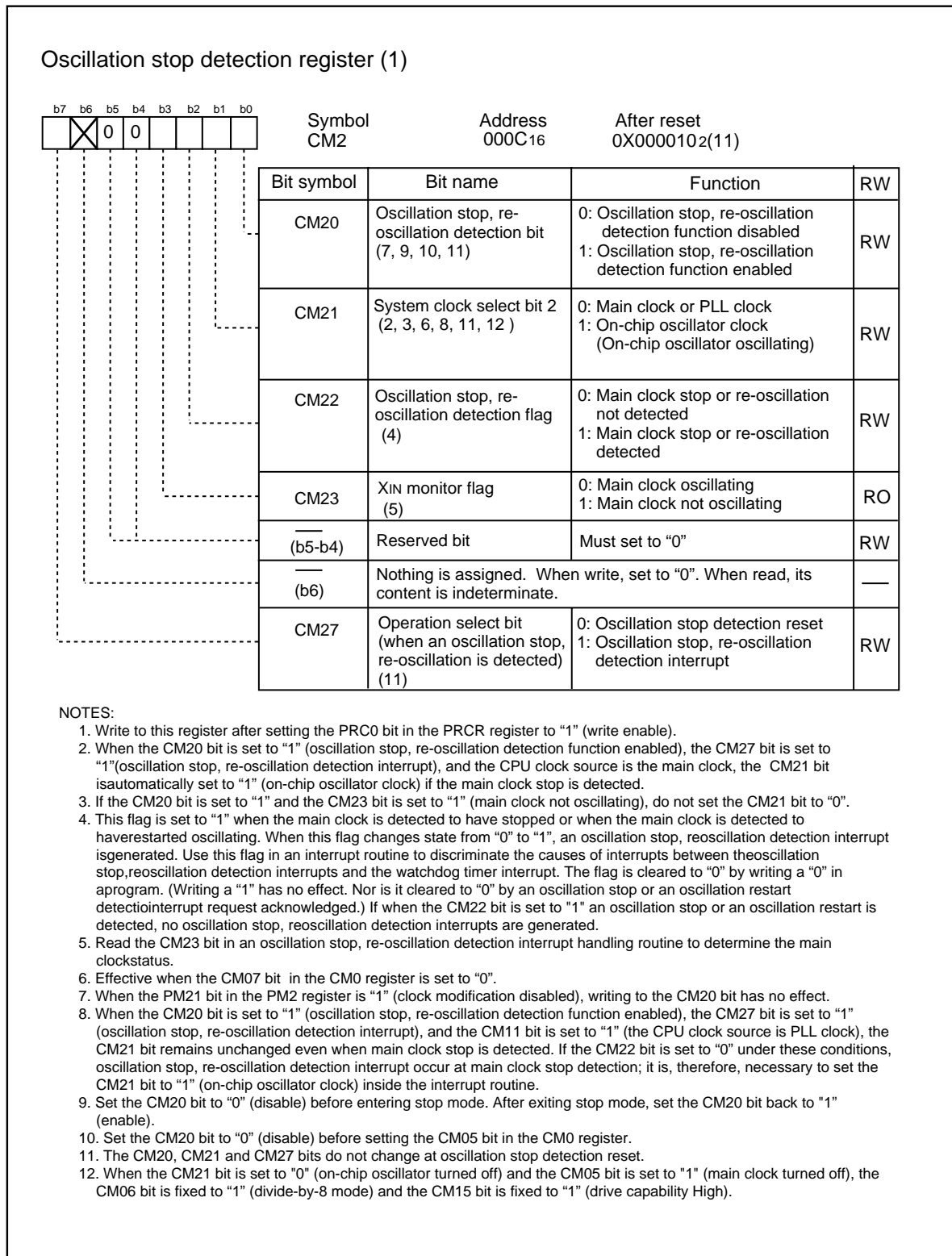


Figure 7.5. CM2 Register

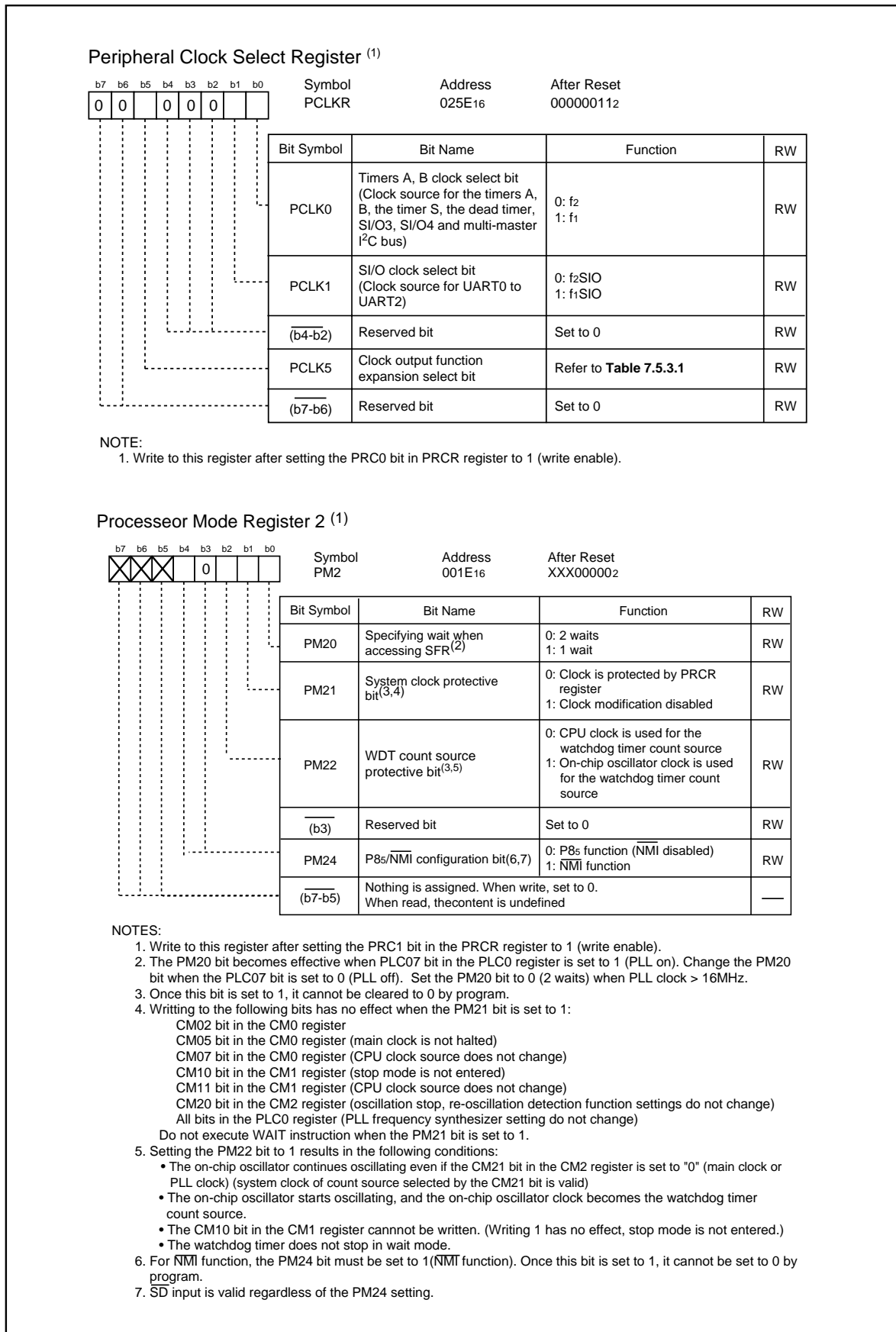


Figure 7.6. PCLKR Register and PM2 Register

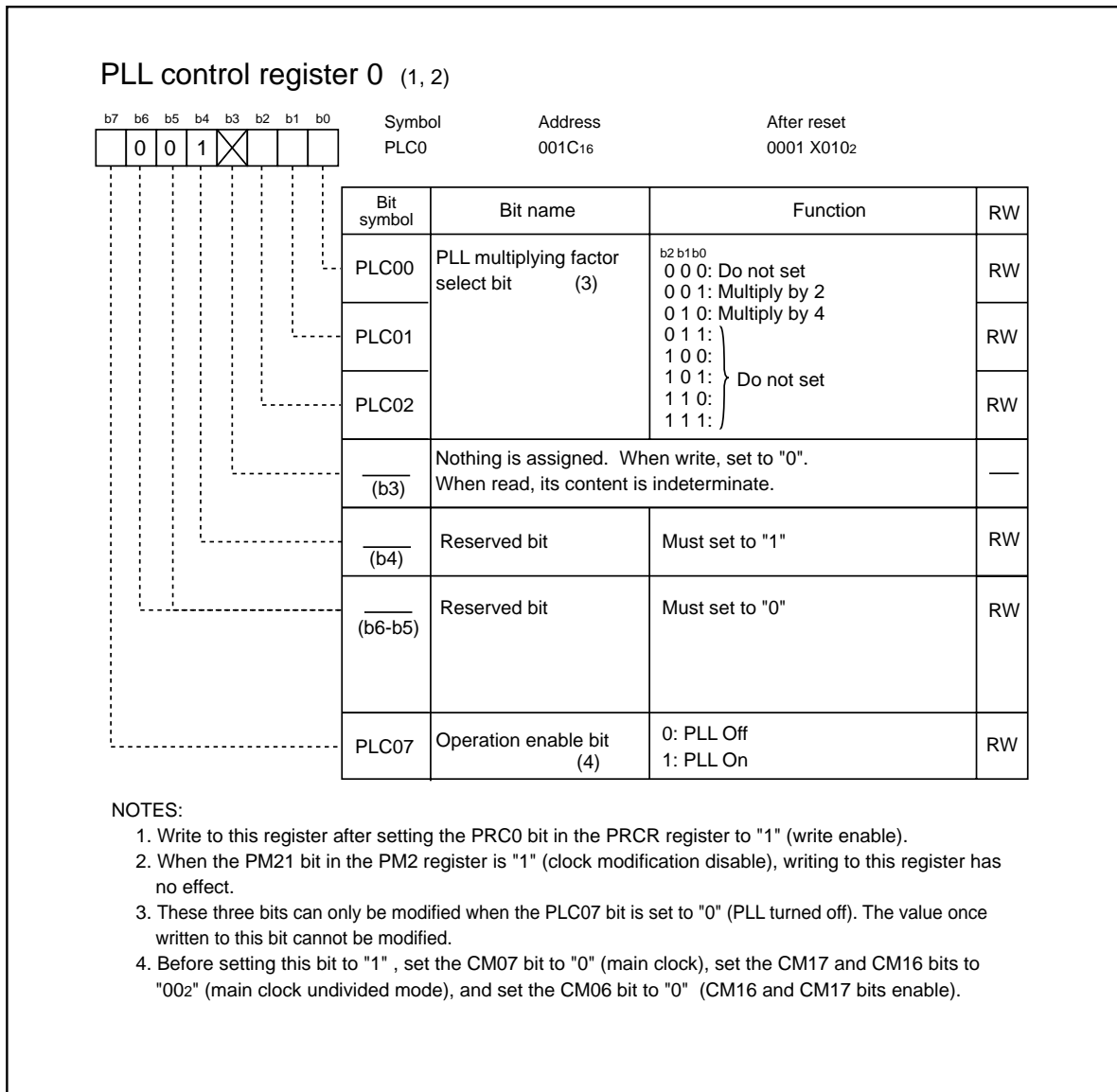


Figure 7.7. PLC0 Register

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

## 7.1 Main Clock

The main clock is generated by the main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillator circuit is configured by connecting a resonator between the XIN and XOUT pins. The main clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillator circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 7.1.1 shows the examples of main clock connection circuit.

The main clock after reset oscillates in the M16C/26A and M16C/26B, but stop in the M16C/26T.

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

During stop mode, all clocks including the main clock are turned off. Refer to **7.6 power control**.

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

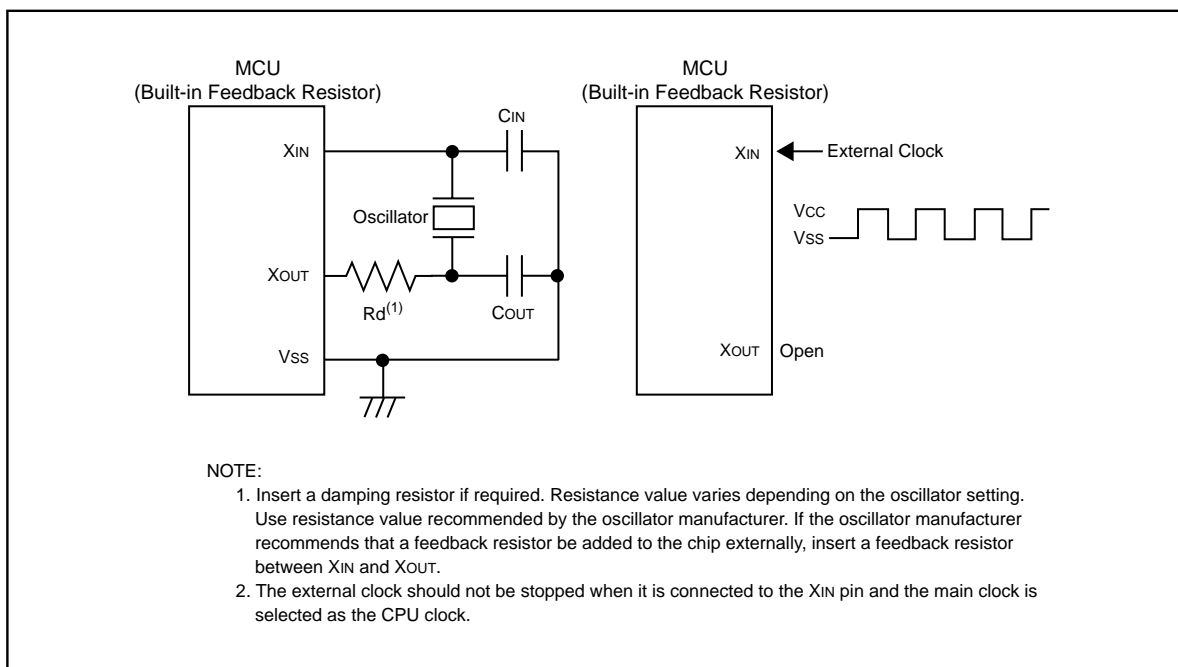


Figure 7.1.1. Examples of Main Clock Connection Circuit

## 7.2 Sub Clock

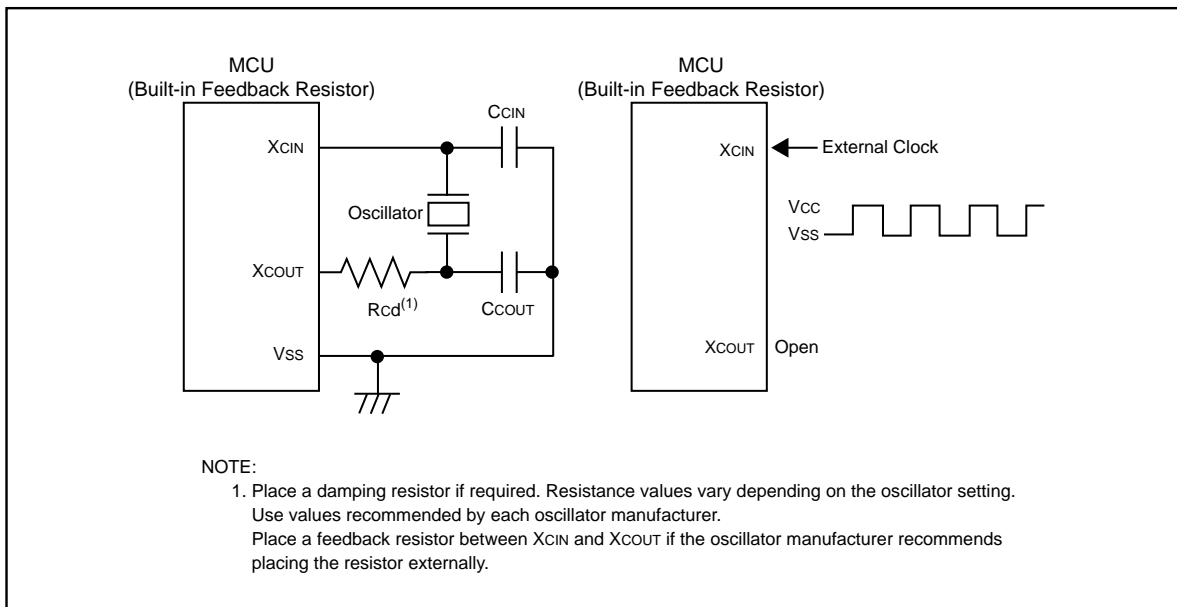
The sub clock is generated by the sub clock oscillation circuit. This clock is used as the clock source for the CPU clock, as well as the timer A and timer B count sources.

The sub clock oscillator circuit is configured by connecting a crystal resonator between the XCIN and XCOUT pins. The sub clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The sub clock oscillator circuit may also be configured by feeding an externally generated clock to the XCIN pin. Figure 7.2.1 shows the examples of sub clock connection circuit.

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

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

During stop mode, all clocks including the sub clock are turned off. Refer to **7.6 Power Control**.



**Figure 7.2.1. Examples of Sub Clock Connection Circuit**

### 7.3 On-chip Oscillator Clock

This clock 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 **10.1 Count source protective mode**).

The on-chip oscillator clock after reset oscillates. The on-chip oscillator clock  $f_2(\text{ROC})$  divided by 16 is used for the CPU clock. It can also be turned off by setting the CM21 bit in the CM2 register to “0” (main clock or PLL 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 micro-computer.

### 7.4 PLL Clock

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

Before entering wait mode or stop mode, be sure to set the CM11 bit to “0” (CPU clock source is the main clock). Furthermore, before entering stop mode, be sure to set the PLC07 bit in the PLC0 register to “0” (PLL stops). Figure 7.4.1 shows the procedure for using the PLL clock as the clock source for the CPU.

The PLL clock frequency is determined by the equation below.

PLL clock frequency =  $f(\text{XIN}) \times$  (multiplying factor set by the PLC02 to PLC00 bits in the PLC0 register)  
 (However,  $10 \text{ MHz} \leq \text{PLL clock frequency} \leq 20 \text{ MHz}$  in M16C/26A and M16C/26T,  $10 \text{ MHz} \leq \text{PLL clock frequency} \leq 24 \text{ MHz}$  in M16C/26B)

The PLC02 to PLC00 bits can be set only once after reset. Table 7.4.1 shows the example for setting PLL clock frequencies.

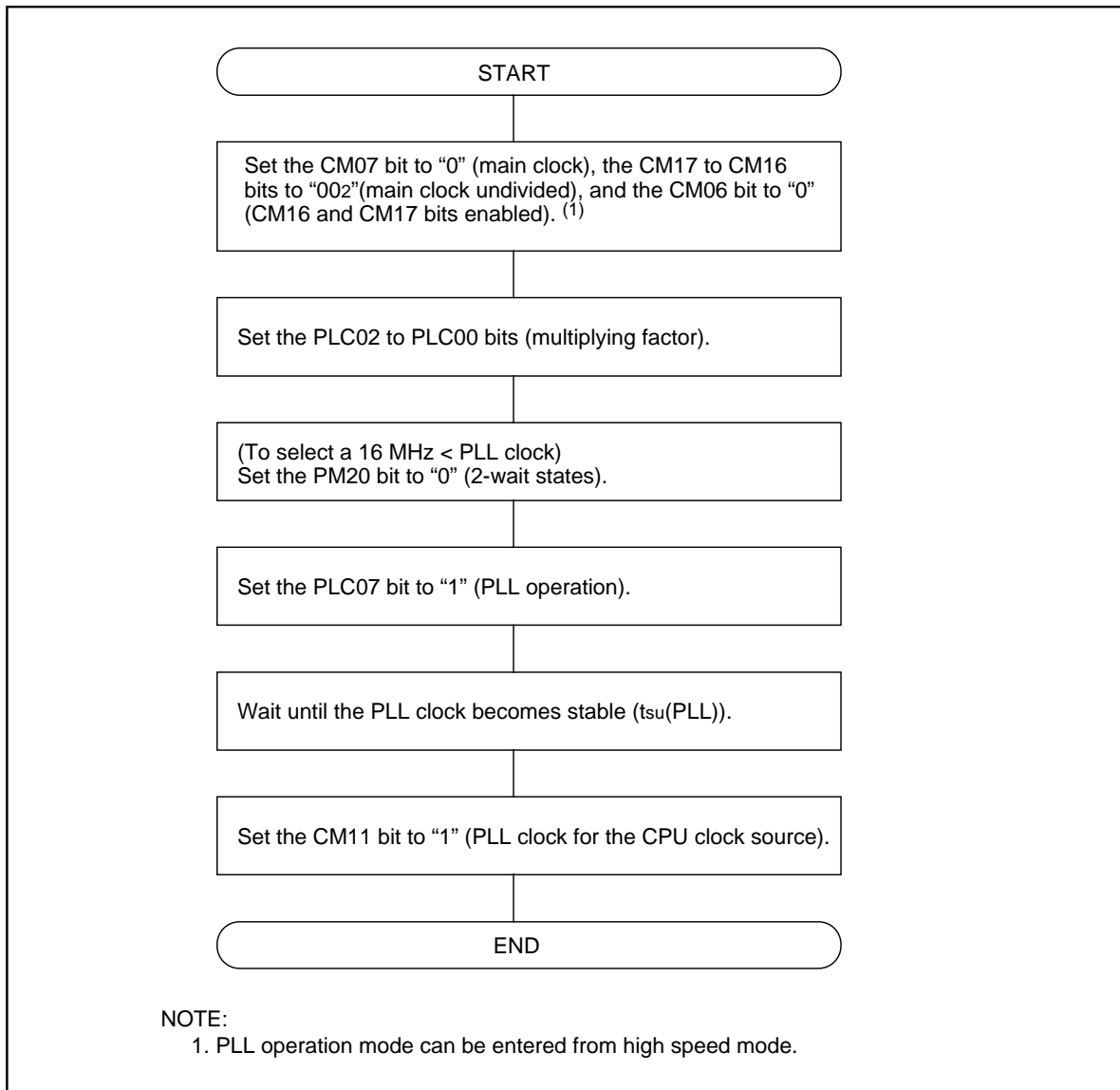
**Table 7.4.1. Example for Setting PLL Clock Frequencies**

XIN (MHz)	PLC02	PLC01	PLC00	Multiplying factor	PLL clock (MHz) <sup>(1)</sup>
10	0	0	1	2	20
5	0	1	0	4	

NOTE:

1.  $10 \text{ MHz} \leq \text{PLL clock frequency} \leq 20 \text{ MHz}$  in M16C/26A and M16C/26T,  $10 \text{ MHz} \leq \text{PLL clock frequency} \leq 24 \text{ MHz}$  in M16C/26B)



**Figure 7.4.1. Procedure to Use PLL Clock as CPU Clock Source**

## 7.5 CPU Clock and Peripheral Function Clock

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

### 7.5.1 CPU Clock

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

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

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

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

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

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

### 7.5.2 Peripheral Function Clock (f1, f2, f8, f32, f1SIO, f2SIO, f8SIO, f32SIO, fAD, fc32)

These are operating clocks for the peripheral functions.

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 divided by  $i$ . The clock  $f_i$  is used for Timer A and Timer B while  $f_{iSIO}$  is used for UART0 to UART2. Additionally, the  $f_1$  and  $f_2$  clocks are also used for dead time timer.

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.

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 microcomputer is in low power dissipation mode, the  $f_i$ ,  $f_{iSIO}$  and  $f_{AD}$  clocks are turned off.

The  $f_{c32}$  clock is produced from the sub clock, and is used for timers A and B. This clock can only be used when the sub clock is on.

### 7.5.3 ClockOutput Function

The  $f_1$ ,  $f_8$ ,  $f_{32}$  or  $f_c$  clock can be output from the CLKOUT pin. Use the PCLK5 bit in the PCLKR register and CM01 to CM00 bits in the CM0 register to select. Table 7.5.3.1 shows the function of the CLKOUT pin.

**Table 7.5.3.1 The function of the CLKOUT pin**

PCLK5	CM01	CM00	The function of the CLKOUT pin
0	0	0	I/O port P9 <sub>0</sub>
0	0	1	$f_c$
0	1	0	$f_8$
0	1	1	$f_{32}$
1	0	0	$f_1$
1	0	1	Do not set
1	1	0	Do not set
1	1	1	Do not set

## 7.6 Power Control

There are three power control modes. For convenience' sake, all modes other than wait and stop modes are referred to as normal operation mode here.

### 7.6.1 Normal Operation Mode

Normal operation mode is further classified into seven modes.

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

Before the clock sources for the CPU clock can be switched over, the new clock source 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 operation modes cannot be changed directly from low power dissipation mode to on-chip oscillator mode or on-chip oscillator low power dissipation mode. Nor can operation modes be changed directly from on-chip oscillator mode or on-chip oscillator low power dissipation mode to low power dissipation mode.

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

#### 7.6.1.1 High-speed Mode

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

#### 7.6.1.2 PLL Operation Mode

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

#### 7.6.1.3 Medium-speed Mode

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

#### 7.6.1.4 Low-speed Mode

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

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

#### 7.6.1.5 Low Power Dissipation Mode

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

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

### 7.6.1.6 On-chip Oscillator Mode

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

### 7.6.1.7 On-chip Oscillator Low Power Dissipation Mode

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

**Table 7.6.1.1. Setting Clock Related Bit and Modes**

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

**NOTES:**

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

## 7.6.2 Wait Mode

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

### 7.6.2.1 Peripheral Function Clock Stop Function

If the CM02 bit is “1” (peripheral function clocks turned off during wait mode), the f1, f2, f8, f32, f1SIO, f8SIO, f32SIO and fAD clocks are turned off when in wait mode, with the power consumption reduced that much. However, fc32 remains on.

### 7.6.2.2 Entering Wait Mode

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

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

### 7.6.2.3 Pin Status During Wait Mode

Table 7.6.2.3.1 lists pin status during wait mode.

**Table 7.6.2.3.1 Pin Status in Wait Mode**

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

### 7.6.2.4 Exiting Wait Mode

The microcomputer is moved out of wait mode by a hardware reset,  $\overline{\text{NMI}}$  interrupt or peripheral function interrupt.

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

The peripheral function interrupts are affected by the CM02 bit. If the CM02 bit is set to "0" (peripheral function clocks not turned off during wait mode), all peripheral function interrupts can be used to exit wait mode. If the CM02 bit is set to "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 7.6.2.4.1 lists the interrupts to exit wait mode.

**Table 7.6.2.4.1. Interrupts to Exit Wait Mode**

Interrupt	CM02=0	CM02=1
NMI interrupt	Can be used	Can be used
Serial I/O 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

If the microcomputer is to be moved out of wait mode by a peripheral function interrupt, set up the following before executing the WAIT instruction.

- In the ILVL2 to ILVL0 bits in the interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit wait mode.  
Also, for all of the peripheral function interrupts not used to exit wait mode, set the ILVL2 to ILVL0 bits to "0002" (interrupt disable).
- Set the I flag to "1".
- Enable the peripheral function whose interrupt is to be used to exit wait mode.

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

The CPU clock turned on when exiting wait mode by a peripheral function interrupt is the same CPU clock that was on when the WAIT instruction was executed.

### 7.6.3 Stop Mode

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

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

- $\overline{NMI}$  interrupt
- Key interrupt
- $\overline{INT}$  interrupt
- Timer A, Timer B interrupt (when counting external pulses in event counter mode)
- Serial I/O interrupt (when external clock is selected)
- Voltage down detection interrupt

(refer to **5.5.1 Voltage Down Detection Interrupt** for an operating condition)

#### 7.6.3.1 Entering Stop Mode

The microcomputer is placed into stop mode by setting the CM10 bit in the CM1 register to “1” (all clocks turned off). At the same time, the CM06 bit in the CM0 register is set to “1” (divide-by-8 mode) and the CM15 bit in the CM10 register is set to “1” (main clock oscillator circuit drive capability high). Before entering stop mode, set the CM20 bit to “0” (oscillation stop, re-oscillation detection function disable).

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

#### 7.6.3.2 Pin Status during Stop Mode

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

#### 7.6.3.3 Exiting Stop Mode

The microcomputer is moved out of stop mode by a hardware reset,  $\overline{NMI}$  interrupt or peripheral function interrupt.

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

If the microcomputer is to be moved out of stop mode by a peripheral function interrupt, set up the following before setting the CM10 bit to “1”.

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

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

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

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

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

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

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

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

Figure 7.6.1 shows the state transition from normal operation mode to stop mode and wait mode. Figure 7.6.1.1 shows the state transition in normal operation mode. Table 7.6.1 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line shows state after transition.

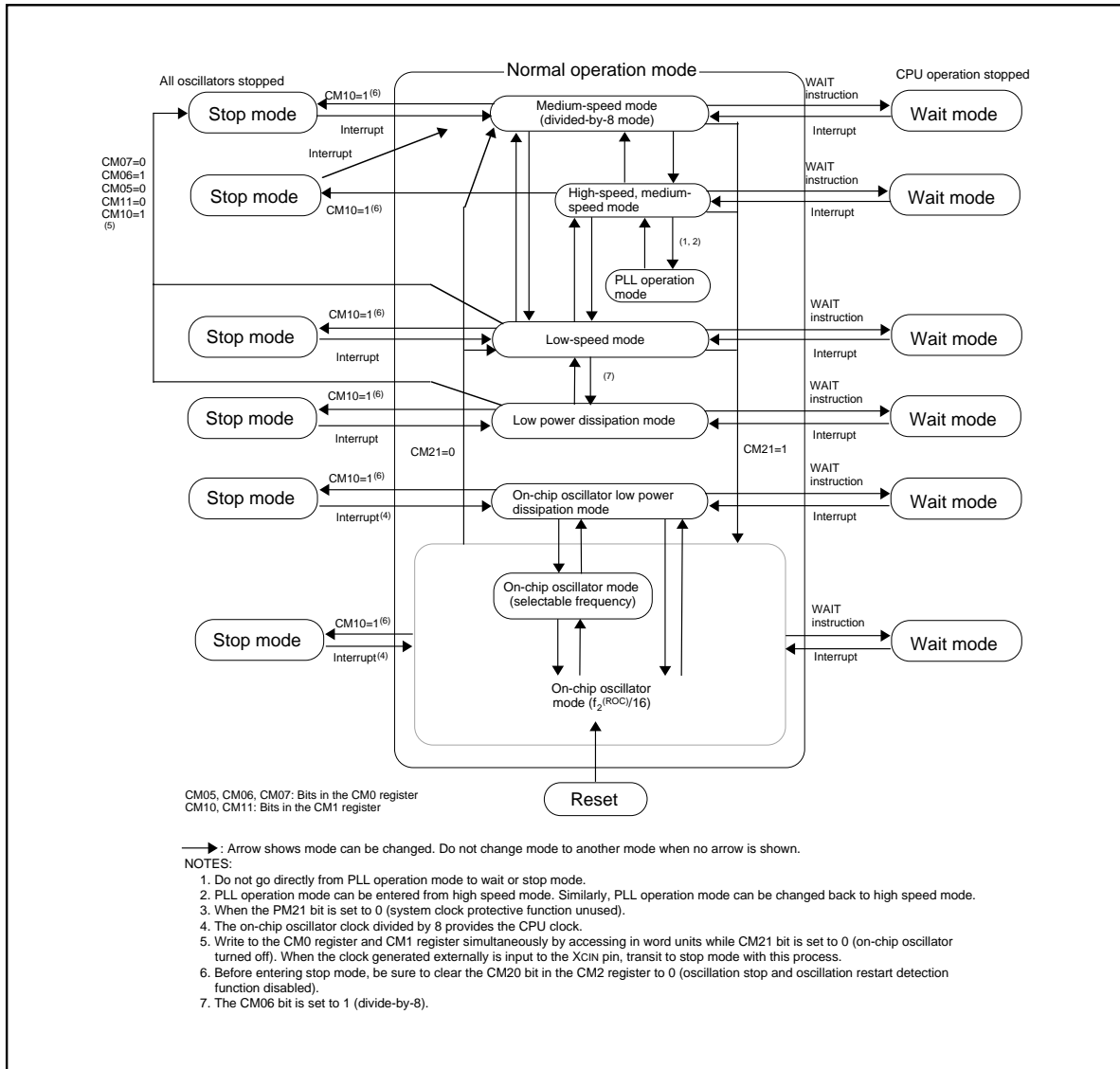


Figure 7.6.1. State Transition to Stop Mode and Wait Mode

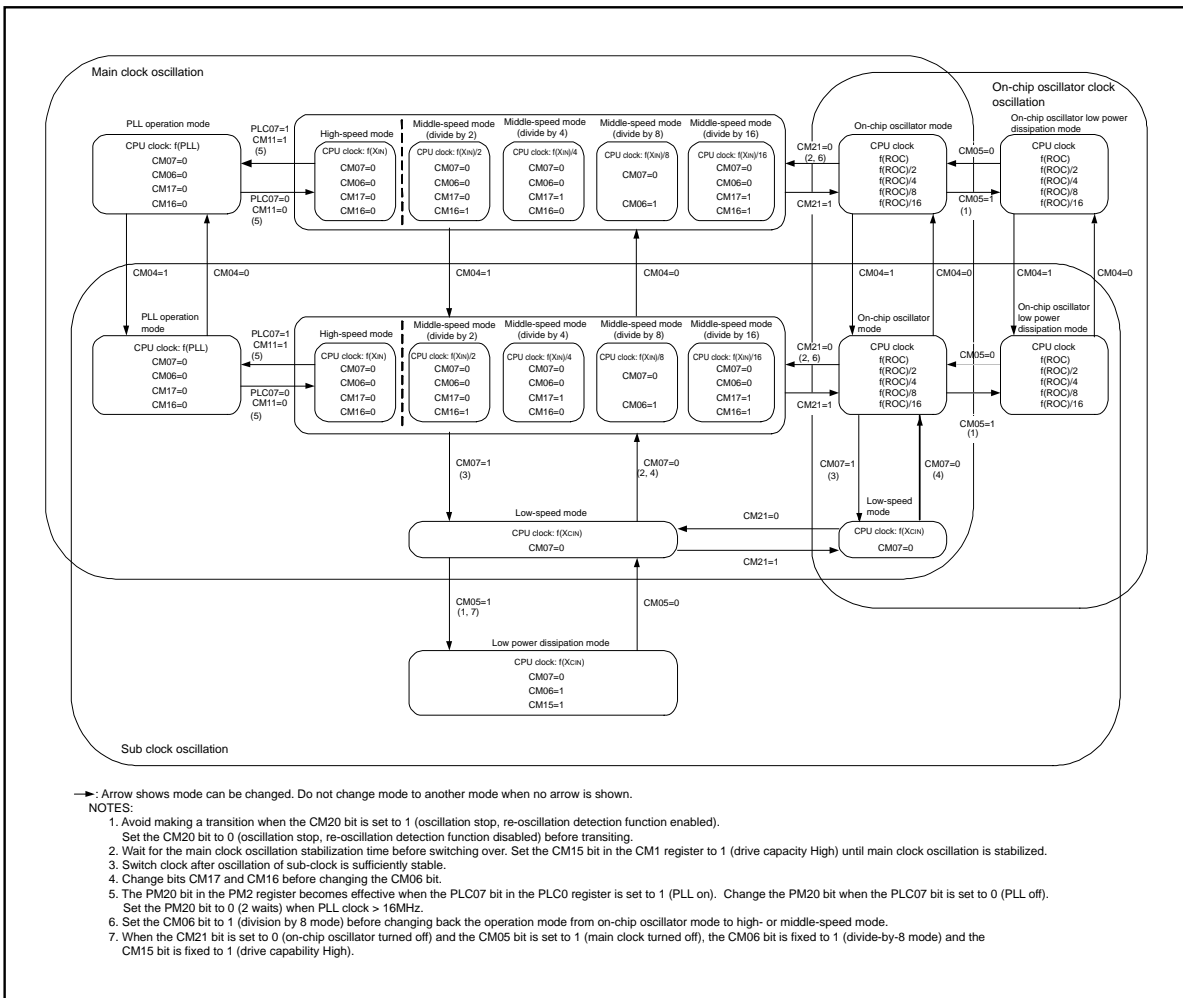


Figure 7.6.1.1. State Transition in Normal Mode



**Table 7.6.1. Allowed Transition and Setting**

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

NOTES:

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

--: Cannot transit

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

9. ( ) : setting method. Refer to following table.

--: Cannot transit

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

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

## 7.7 System Clock Protective Function

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

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

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

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

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

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

## 7.8 Oscillation Stop and Re-oscillation Detect Function

The oscillation stop and re-oscillation detect function allows the detection of main clock oscillation stop and re-oscillation. At oscillation stop or re-oscillation detection, reset or oscillation stop, re-oscillation detection interrupt are generated. Depending on the CM27 bit in the CM2 register. The oscillation stop detection function can be enabled and disabled by the CM20 bit in the CM2 register. Table 7.8.1 lists a specification overview of the oscillation stop and re-oscillation detect function.

**Table 7.8.1. Specification Overview of Oscillation Stop and Re-oscillation Detect Function**

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

### 7.8.1 Operation When the CM27 bit is set to "0" (Oscillation Stop Detection Reset)

When main clock stop is detected when the CM20 bit is "1" (oscillation stop, re-oscillation detection function enabled), the microcomputer is initialized, coming to a halt (oscillation stop reset; refer to 4. SFR, 5. Reset).

This status is reset with hardware reset 1 or hardware reset 2. Also, even when re-oscillation is detected, the microcomputer can be initialized and stopped; it is, however, necessary to avoid such usage. (During main clock stop, do not set the CM20 bit to "1" and the CM27 bit to "0".)

### 7.8.2 Operation When the CM27 bit is set to "1" (Oscillation Stop and Re-oscillation Detect Interrupt)

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

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

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

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

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

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

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

- The oscillation stop and re-oscillation detect interrupt shares the vector with the watchdog timer interrupt. If the oscillation stop, re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, return the main clock to the CPU clock and peripheral function clock source in the program. Figure 7.8.3.1 shows the procedure for switching the clock source from the on-chip oscillator to the main clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt occurrence, the CM22 bit becomes "1". When the CM22 bit is set at "1", oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to "0" 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, re-oscillation detection function, set the CM02 bit to "0" (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop, re-oscillation detection function is provided in preparation for main clock stop due to external factors, set the CM20 bit to "0" (Oscillation stop, re-oscillation detection function disabled) where the main clock is stopped or oscillated 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".

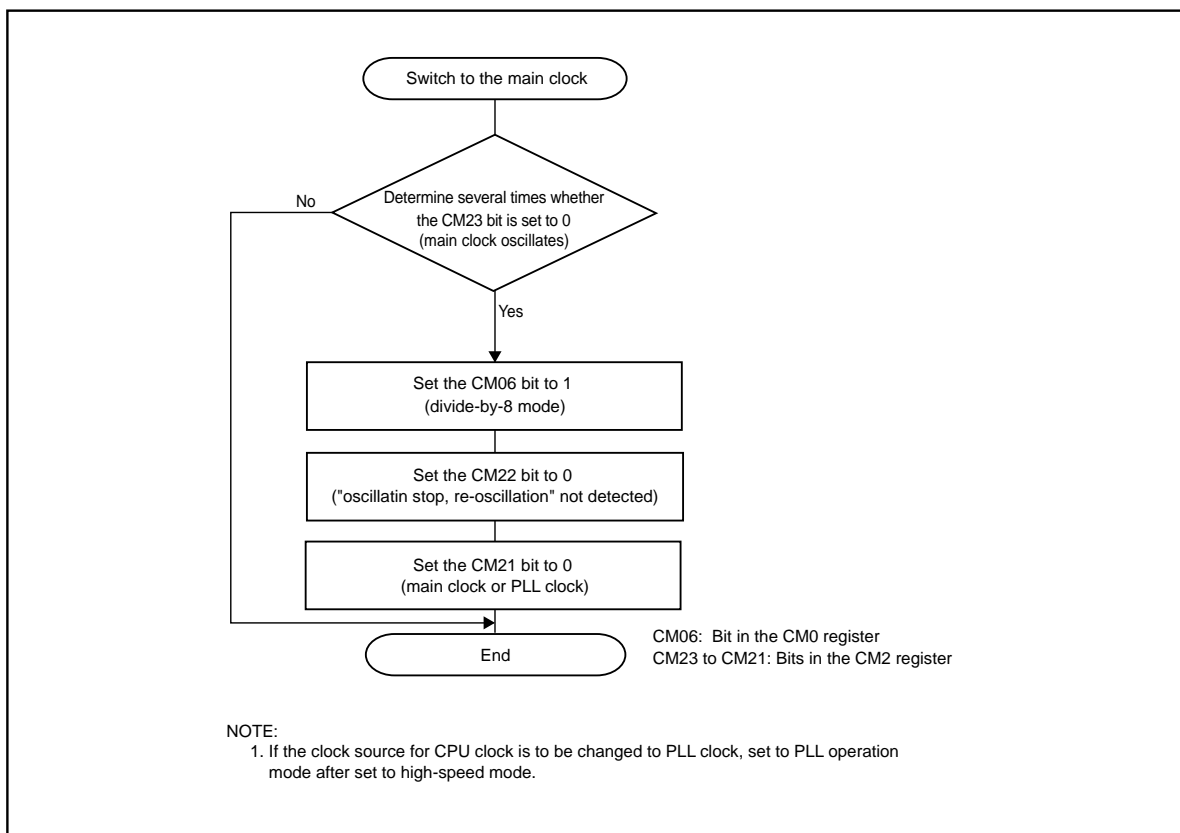


Figure 7.8.3.1. Procedure to Switch Clock Source From On-chip Oscillator to Main Clock

## 8. Protection

### Note

The PRC3 bit in the PRCR register is not available in M16C/26T.

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

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

Set the PRC2 bit to "1" (write enabled) and then write to SFR area, and the PRC2 bit will be cleared to "0" (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to "1". Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to "1" and the next instruction. The PRC0, PRC1 and PRC3 bits are not automatically cleared to "0" by writing to any address. They can only be cleared in a program.

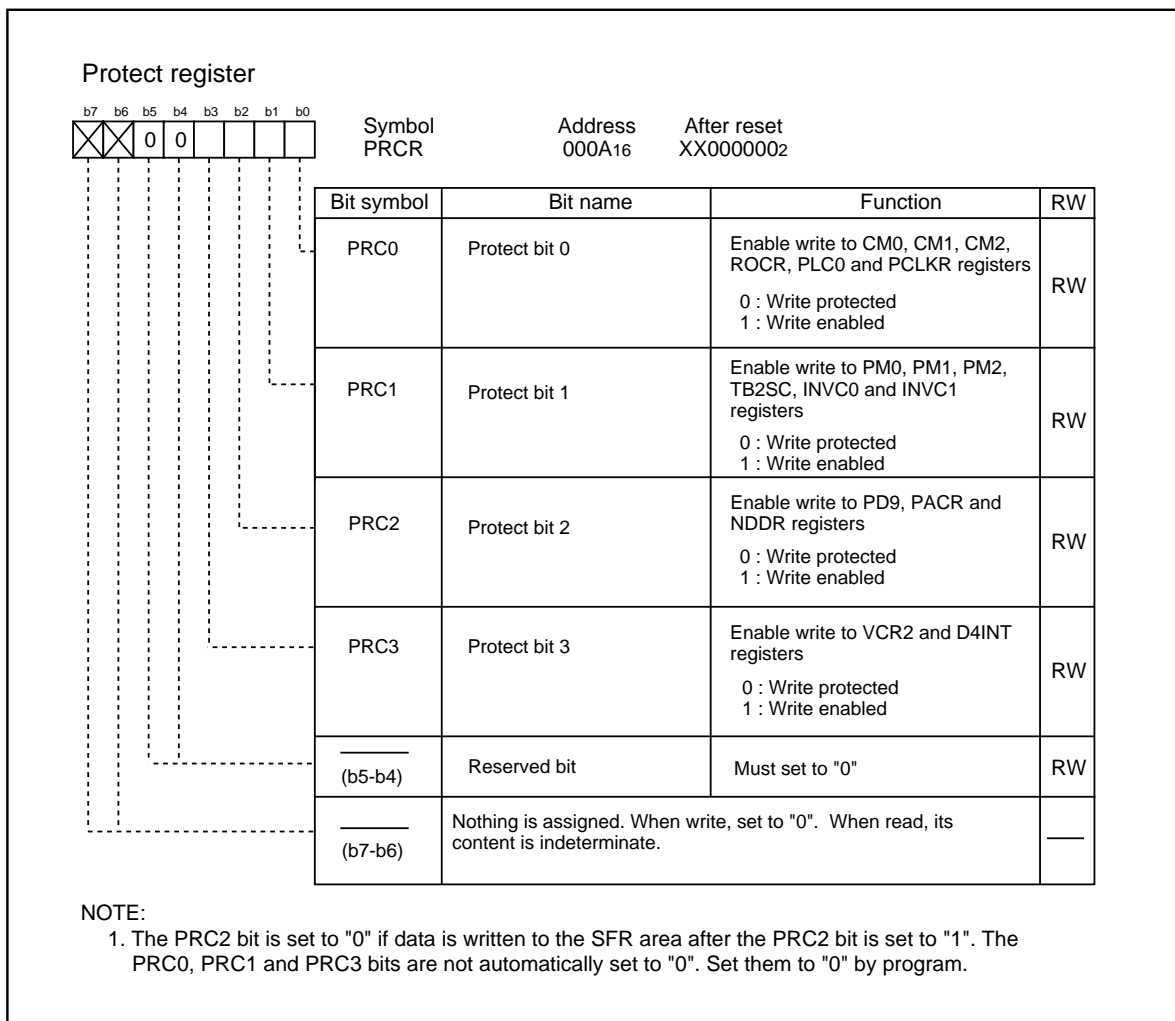


Figure 8.1. PRCR Register

## 9. Interrupt

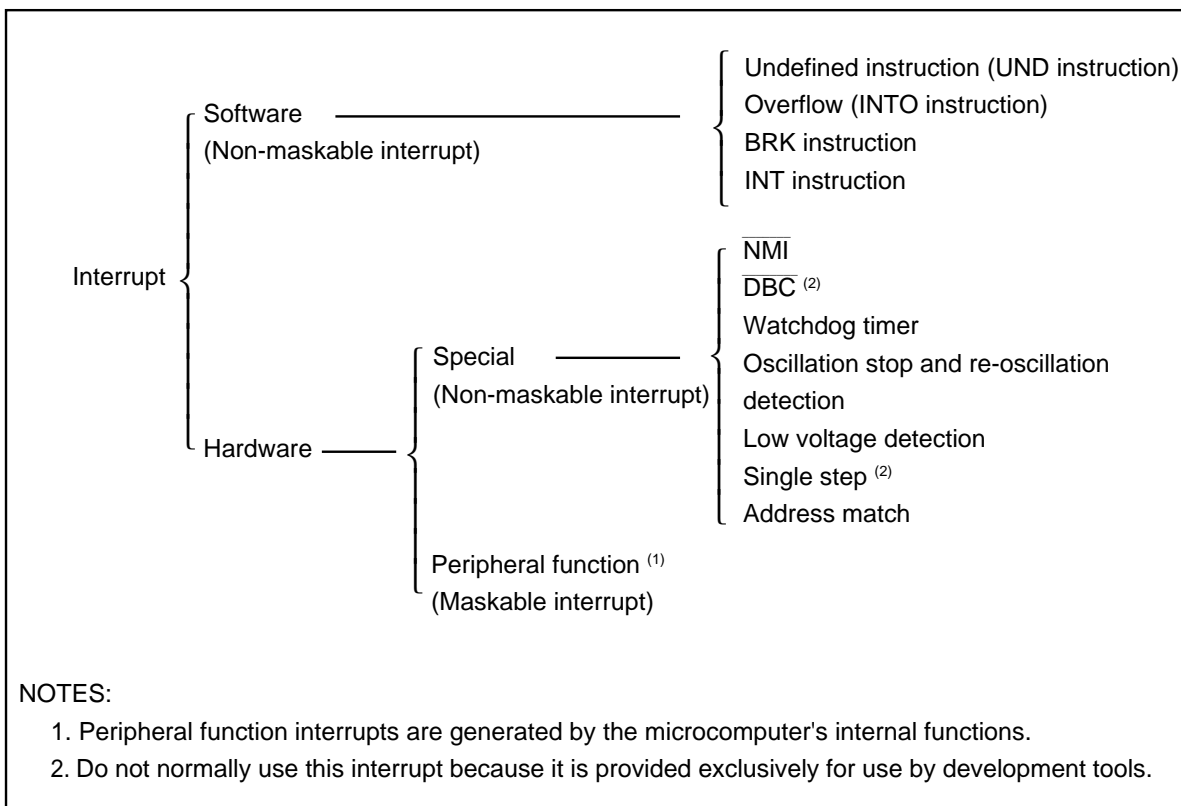
### Note

The 42-pin package does not use UART0 transmission interrupt and UART0 reception interrupt of peripheral function.

M16C/26T does not use voltage down detection interrupt.

### 9.1 Type of Interrupts

Figure 9.1.1 shows types of interrupts.



**Figure 9.1.1. Interrupts**

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

## 9.1.1 Software Interrupts

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

### 9.1.1.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

### 9.1.1.2 Overflow Interrupt

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

### 9.1.1.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

### 9.1.1.4 INT Instruction Interrupt

An INT instruction interrupt occurs when executing the INT instruction. Software interrupt Nos. 0 to 63 can be specified for the INT instruction. Because software interrupt Nos. 4, 8 to 31 are assigned to peripheral function interrupts, the same interrupt routine as for peripheral function interrupts can be executed by executing the INT instruction.

In software interrupt Nos. 0 to 31, the U flag is saved to the stack during instruction execution and is cleared to "0" (ISP selected) before executing an interrupt sequence. The U flag is restored from the stack when returning from the interrupt routine. In software interrupt Nos. 32 to 63, the U flag does not change state during instruction execution, and the SP then selected is used.

## 9.1.2 Hardware Interrupts

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

### 9.1.2.1 Special Interrupts

Special interrupts are non-maskable interrupts.

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

An  $\overline{\text{NMI}}$  interrupt is generated when input on the  $\overline{\text{NMI}}$  pin changes state from high to low. For details about the  $\overline{\text{NMI}}$  interrupt, refer to the section **9.7  $\overline{\text{NMI}}$  Interrupt**.

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

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

#### 9.1.2.1.3 Watchdog Timer Interrupt

Generated by the watchdog timer. Once a watchdog timer interrupt is generated, be sure to initialize the watchdog timer. For details about the watchdog timer, refer to the section **10. Watchdog Timer**.

#### 9.1.2.1.4 Oscillation Stop and Re-oscillation Detection Interrupt

Generated by the oscillation stop and re-oscillation detection function. For details about the oscillation stop and re-oscillation detection function, refer to the section **7. Clock Generating Circuit**.

#### 9.1.2.1.5 Voltage Down Detection Interrupt

Generated by the voltage detection circuit. For details about the voltage detection circuit, refer to the section **5.5 Voltage Detection Circuit**.

#### 9.1.2.1.6 Single-step Interrupt

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

#### 9.1.2.1.7 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMAD0 or RMAD1 register, if the corresponding enable bit (the AIER0 or AIER1 bit in the AIER register) is set to "1". For details about the address match interrupt, refer to the section **9.9 Address Match Interrupt**.

### 9.1.2.2 Peripheral Function Interrupts

Peripheral function interrupts are maskable interrupts and generated by the microcomputer's internal functions. The interrupt sources for peripheral function interrupts are listed in **Table 9.2.2.1 Relocatable Vector Tables**. For details about the peripheral functions, refer to the description of each peripheral function in this manual.



## 9.2 Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. Figure 9.2.1 shows the interrupt vector.

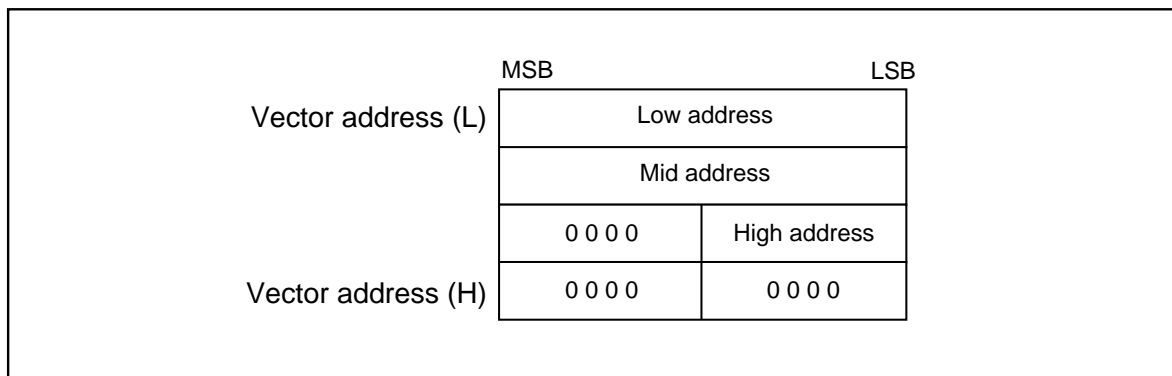


Figure 9.2.1. Interrupt Vector

### 9.2.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDC<sub>16</sub> to FFFFF<sub>16</sub>. Table 9.2.1.1 lists the fixed vector tables. In the flash memory version of microcomputer, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to the section **17.3 Flash Memory Rewrite Disabling Function**.

Table 9.2.1.1. Fixed Vector Tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks	Reference
Undefined instruction	FFFDC <sub>16</sub> to FFFDF <sub>16</sub>	Interrupt on UND instruction	M16C/60, M16C/20 serise software maual
Overflow	FFFE0 <sub>16</sub> to FFFE3 <sub>16</sub>	Interrupt on INTO instruction	
BRK instruction	FFFE4 <sub>16</sub> to FFFE7 <sub>16</sub>	If the contents of address FFFE7 <sub>16</sub> is FF <sub>16</sub> , program execution starts from the address shown by the vector in the relocatable vector table.	
Address match	FFFE8 <sub>16</sub> to FFFEB <sub>16</sub>		Address match interrupt
Single step (1)	FFFE <sub>C16</sub> to FFFE <sub>F16</sub>		
Watchdog timer Oscillation stop and re-oscillation detection Voltage down detection	FFFF0 <sub>16</sub> to FFFF3 <sub>16</sub>		Watchdog timer  Clock generating circuit  Voltage detection circuit
DBC (1)	FFFF4 <sub>16</sub> to FFFF7 <sub>16</sub>		
NMI	FFFF8 <sub>16</sub> to FFFF <sub>B16</sub>		NMI interrupt
Reset (2)	FFFF <sub>C16</sub> to FFFF <sub>F16</sub>		Reset

#### NOTES:

- Do not normally use this interrupt because it is provided exclusively for use by development tools.
- The b3 to b0 in address 0FFFF<sub>16</sub> are reserve bits. Set these bits to "11112".

## 9.2.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. Table 9.2.2.1 lists the relocatable vector tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than in the case of odd addresses.

**Table 9.2.2.1. Relocatable Vector Tables**

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

**NOTES:**

- Address relative to address in INTB.
- Set the IFSR6 and IFSR7 bits in the IFSR register.
- During I<sup>2</sup>C bus mode, NACK and ACK interrupts comprise the interrupt source.
- These interrupts cannot be disabled using the I flag.
- Bus collision detection:  
During IEBus mode, this bus collision detection constitutes the cause of an interrupt.  
During I<sup>2</sup>C bus mode, however, a start condition or a stop condition detection constitutes the cause of an interrupt.

### 9.3 Interrupt Control

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

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

Figure 9.3.1 shows the interrupt control registers.

Figure 9.3.2 shows the IFSR, IFSR2A registers.

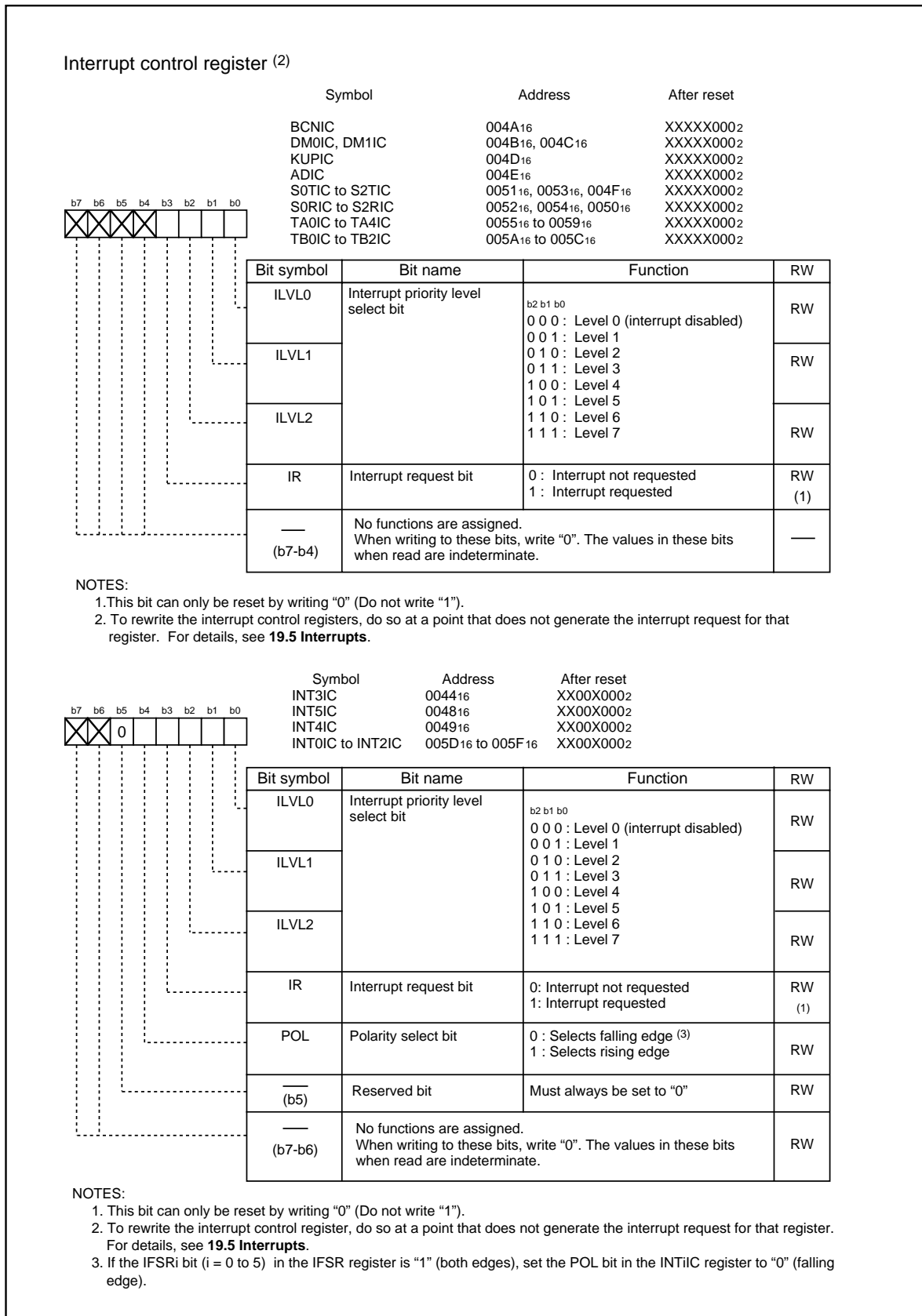


Figure 9.3.1. Interrupt Control Registers

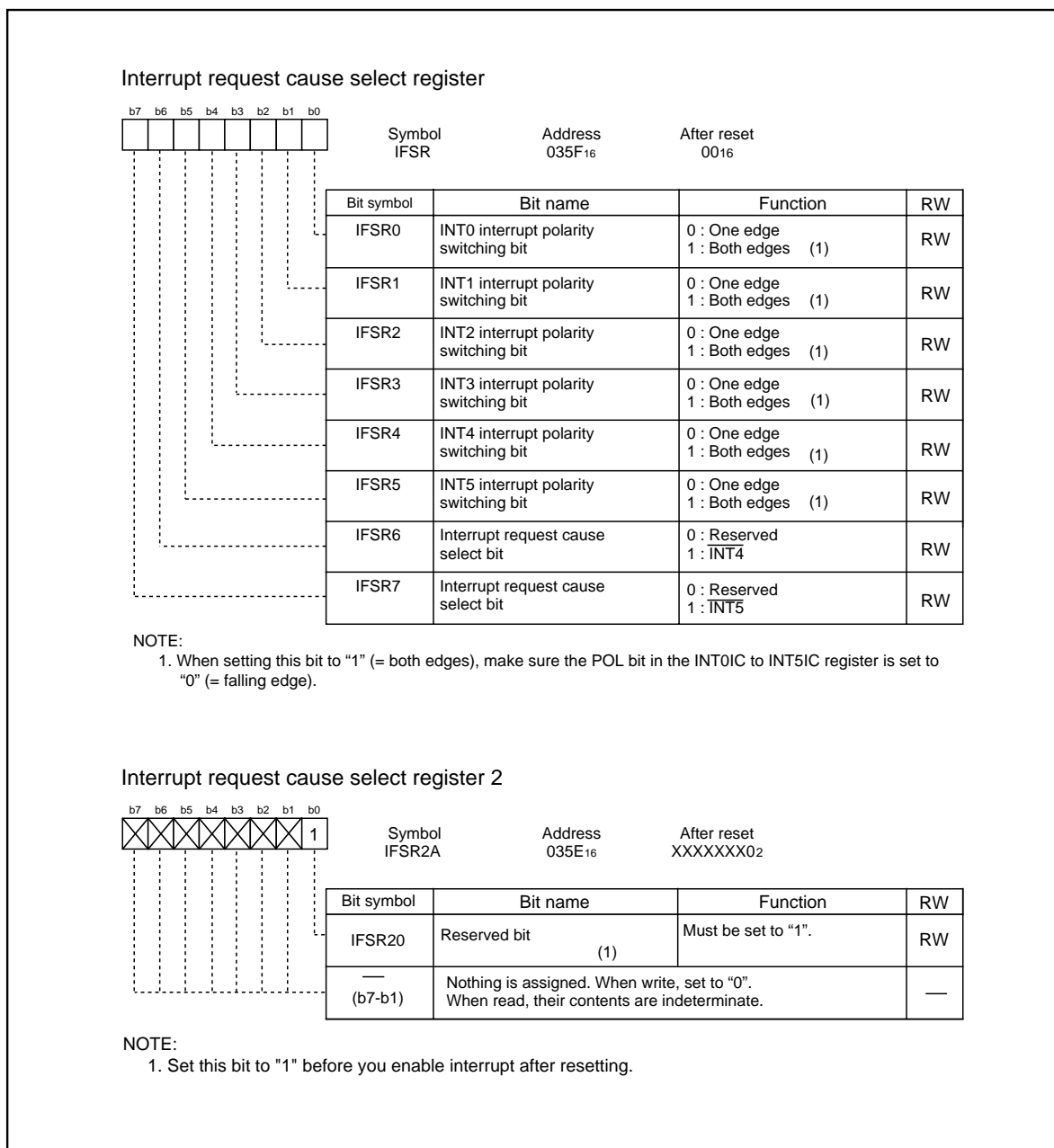


Figure 9.3.2. IFSR Register and IFSR2A Register

### 9.3.1 I Flag

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

### 9.3.2 IR Bit

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

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

### 9.3.3 ILVL2 to ILVL0 Bits and IPL

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

Table 9.3.3.1 shows the settings of interrupt priority levels and Table 9.3.3.2 shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:

- I flag is set to “1”
- IR bit is set to “1”
- interrupt priority level > IPL

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

**Table 9.3.3.1. Settings of Interrupt Priority Levels**

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

**Table 9.3.3.2. Interrupt Priority Levels Enabled by IPL**

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

## 9.4 Interrupt Sequence

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

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

The CPU behavior during the interrupt sequence is described below. Figure 9.4.1 shows time required for executing the interrupt sequence.

- (1) The CPU gets interrupt information (interrupt number and interrupt request priority level) by reading the address 0000<sub>16</sub>. Then it clears the IR bit for the corresponding interrupt to "0" (interrupt not requested).
- (2) The FLG register immediately before entering the interrupt sequence is saved to the CPU's internal temporary register<sup>(1)</sup>.
- (3) The I, D and U flags in the FLG register become as follows:
  - The I flag is cleared to "0" (interrupts disabled).
  - The D flag is cleared to "0" (single-step interrupt disabled).
  - The U flag is cleared to "0" (ISP selected).
 However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.
- (4) The CPU's internal temporary register <sup>(1)</sup> is saved to the stack.
- (5) The PC is saved to the stack.
- (6) The interrupt priority level of the accepted interrupt is set in the IPL.
- (7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

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

### NOTE:

1. This register cannot be used by user.

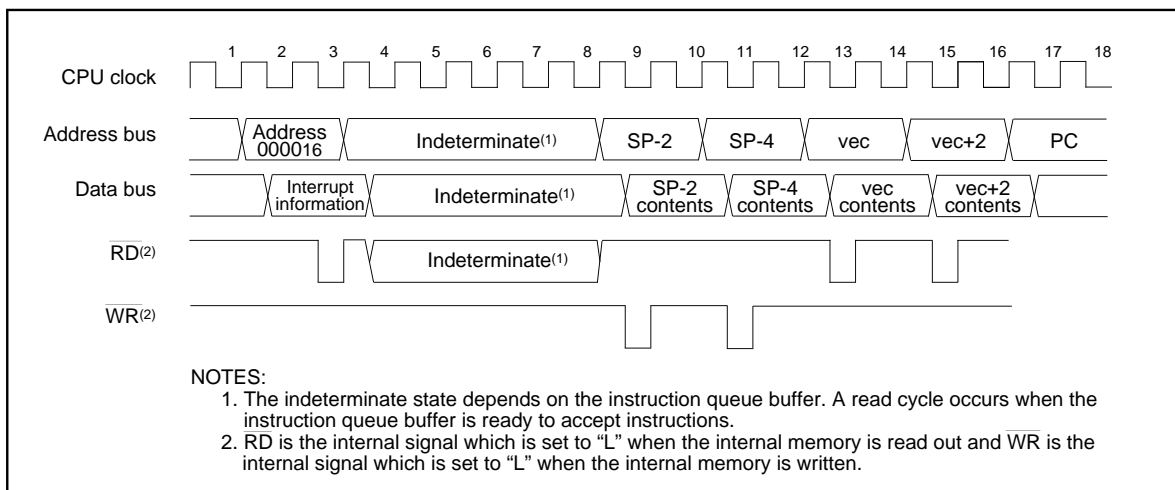


Figure 9.4.1. Time Required for Executing Interrupt Sequence

### 9.4.1 Interrupt Response Time

Figure 9.4.1.1 shows the interrupt response time. The interrupt response or interrupt acknowledge time denotes the time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of the time from when an interrupt request is generated till when the instruction then executing is completed ((a) in Figure 9.4.1.1) and the time during which the interrupt sequence is executed ((b) in Figure 9.4.1.1).

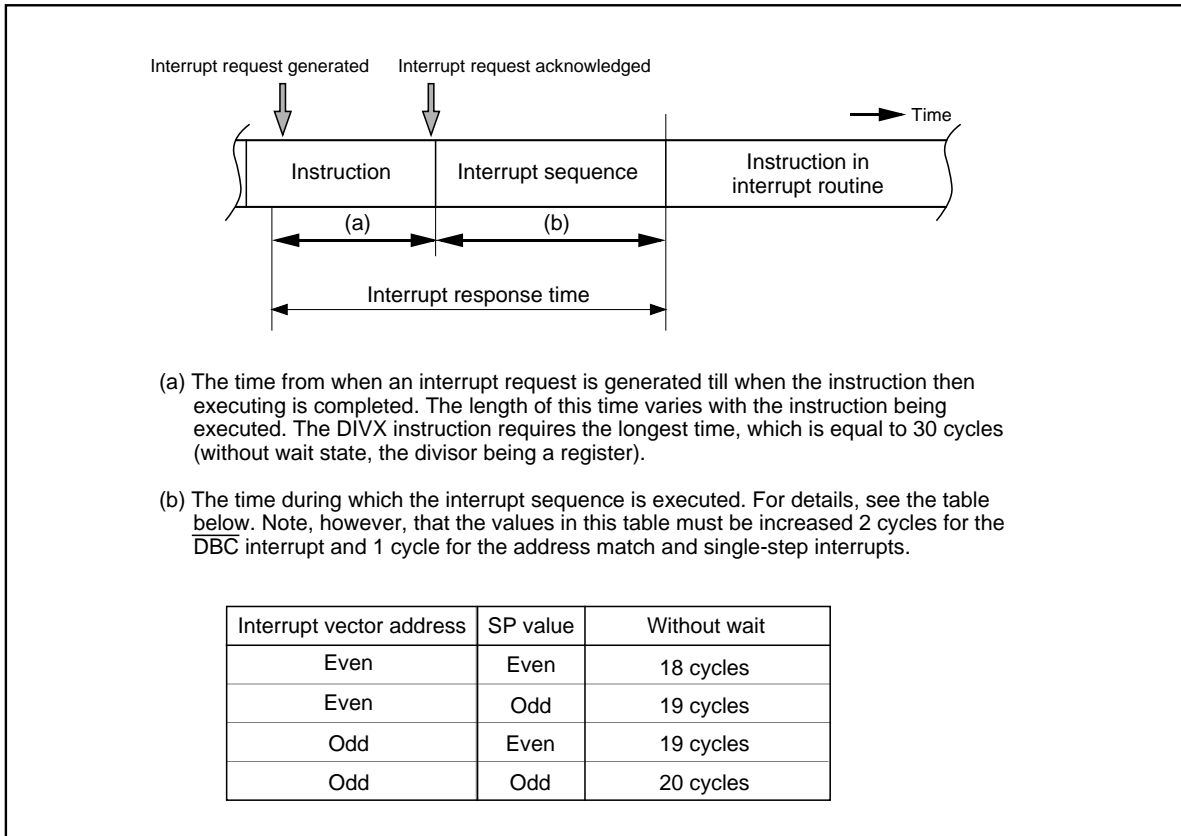


Figure 9.4.1.1. Interrupt response time

### 9.4.2 Variation of IPL when Interrupt Request is Accepted

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

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in Table 9.4.2.1 is set in the IPL. Shown in Table 9.4.2.1 are the IPL values of software and special interrupts when they are accepted.

Table 9.4.2.1. IPL Level That is Set to IPL When A Software or Special Interrupt Is Accepted

Interrupt sources	Level that is set to IPL
Watchdog timer, $\overline{NMI}$ , Oscillation stop and re-oscillation detection, voltage down detection	7
Software, address match, $\overline{DBC}$ , single-step	Not changed



### 9.4.3 Saving Registers

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

At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits 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 9.4.3.1 shows the stack status before and after an interrupt request is accepted.

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

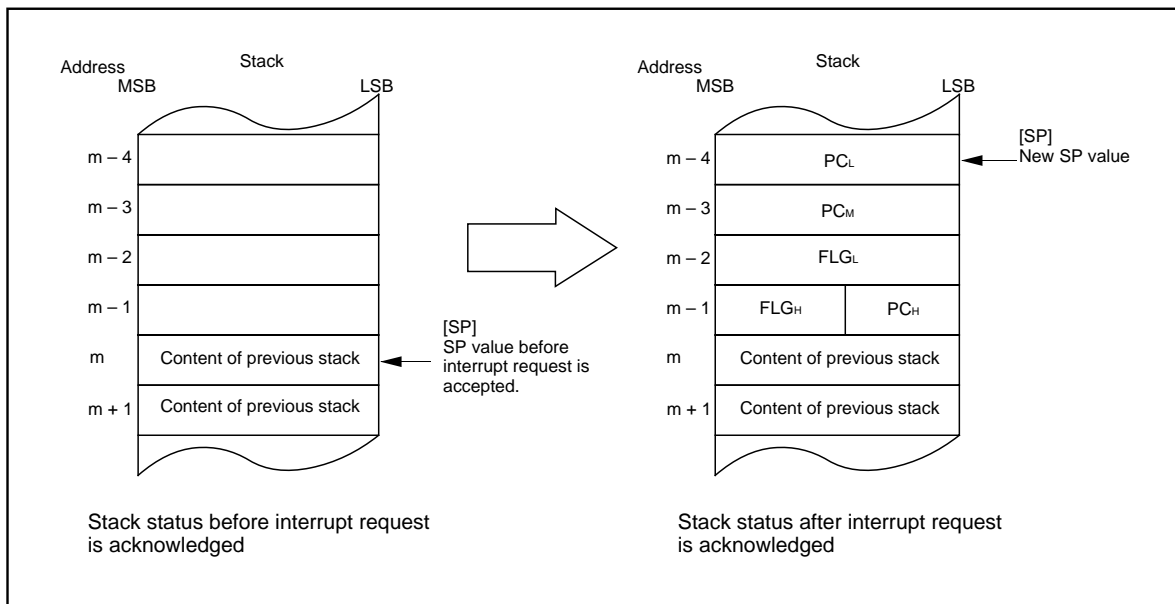
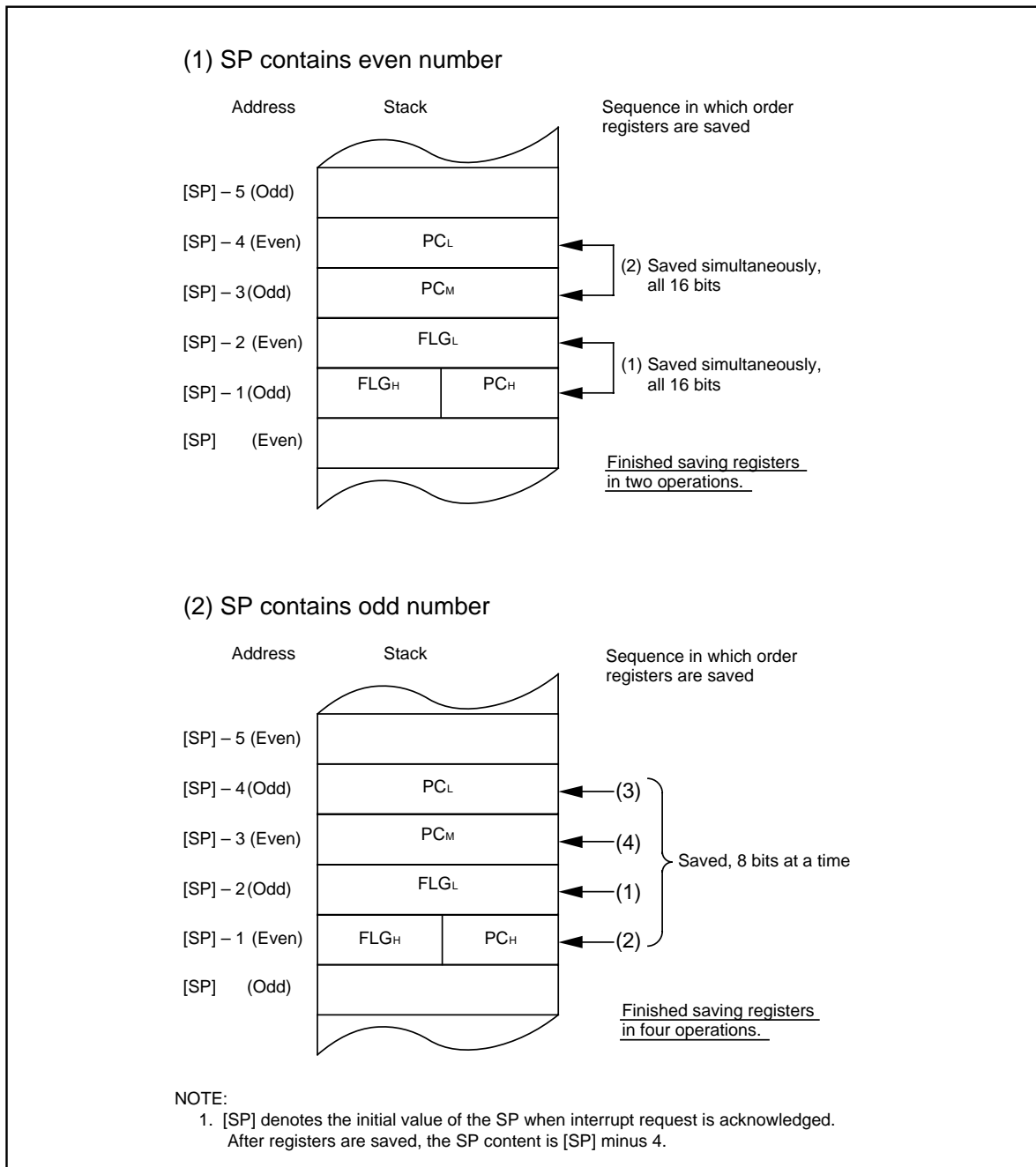


Figure 9.4.3.1. Stack Status Before and After Acceptance of Interrupt Request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP<sup>(1)</sup>, at the time of acceptance of an interrupt request, is even or odd. If the stack pointer <sup>(1)</sup> is even, the FLG register and the PC are saved, 16 bits at a time. If odd, they are saved in two steps, 8 bits at a time. **Figure 9.4.3.2** shows the operation of the saving registers.

## NOTE:

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



**Figure 9.4.3.2. Operation of Saving Register**

### 9.4.4 Returning from an Interrupt Routine

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

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

## 9.5 Interrupt Priority

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

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

The watchdog timer and other special interrupts have their priority levels set in hardware. Figure 9.5.1 shows the priorities of hardware interrupts.

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

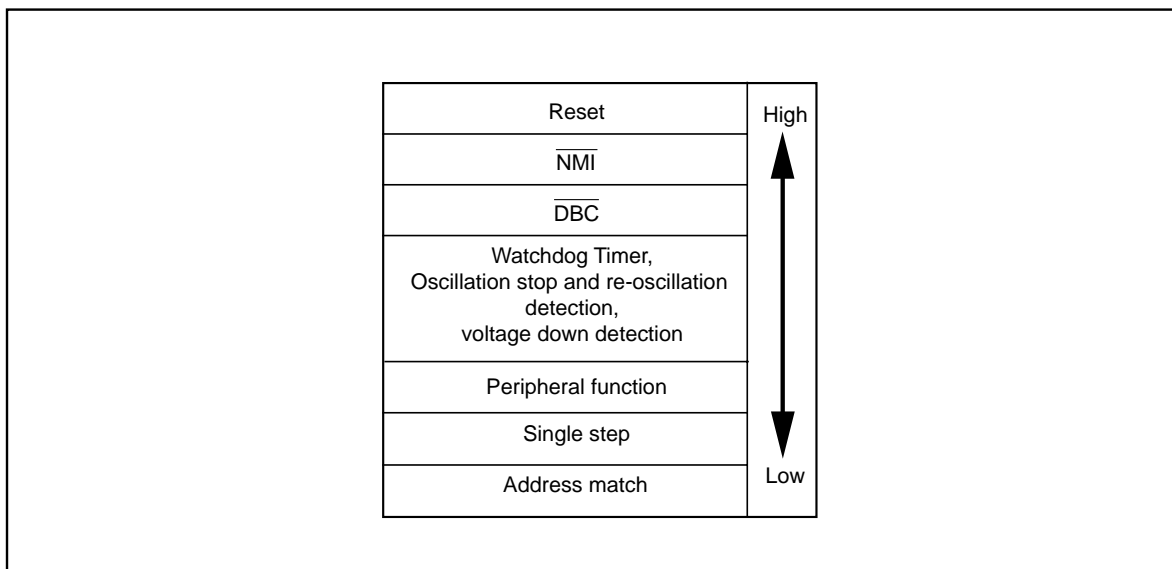


Figure 9.5.1. Hardware Interrupt Priority

### 9.5.1 Interrupt Priority Resolution Circuit

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

Figure 9.5.1.1 shows the circuit that judges the interrupt priority level.

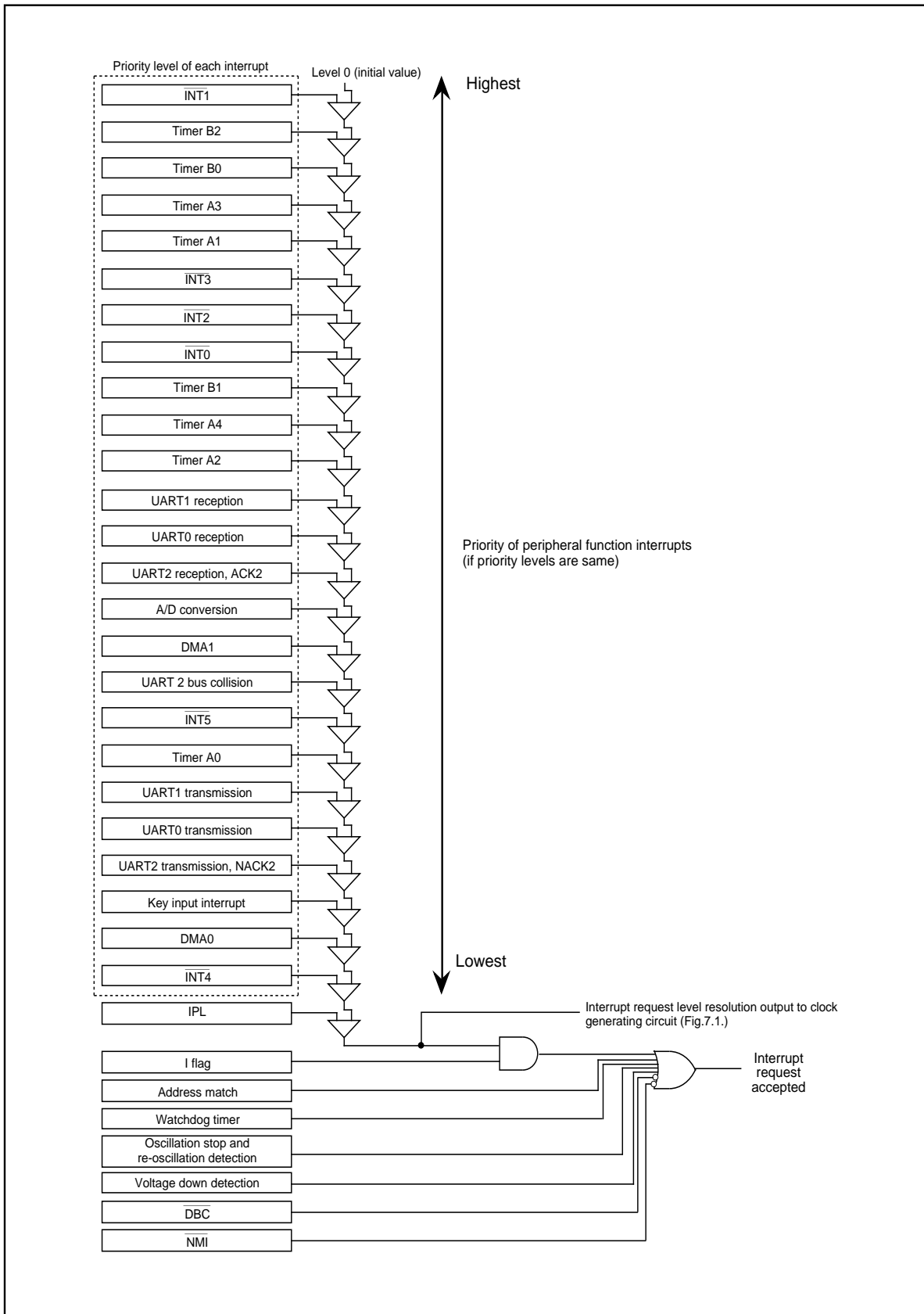


Figure 9.5.1.1. Interrupts Priority Select Circuit

## 9.6 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 the IFSR $_i$  bit in the IFSR register.

To use the  $\overline{\text{INT}}_4$  interrupt, set the IFSR6 bit in the IFSR register to "1" ( $=\overline{\text{INT}}_4$ ). To use the  $\overline{\text{INT}}_5$  interrupt, set the IFSR7 bit in the IFSR register to "1" ( $=\overline{\text{INT}}_5$ ).

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

The  $\overline{\text{INT}}_5$  input has an effective digital debounce function for a noise rejection. Refer to **16.6 Digital Debounce function** for this detail. When using  $\overline{\text{INT}}_5$  interrupt to exit stop mode, set the P17DDR register to "FF16" before entering stop mode.

Figure 9.6.1 shows the IFSR register.

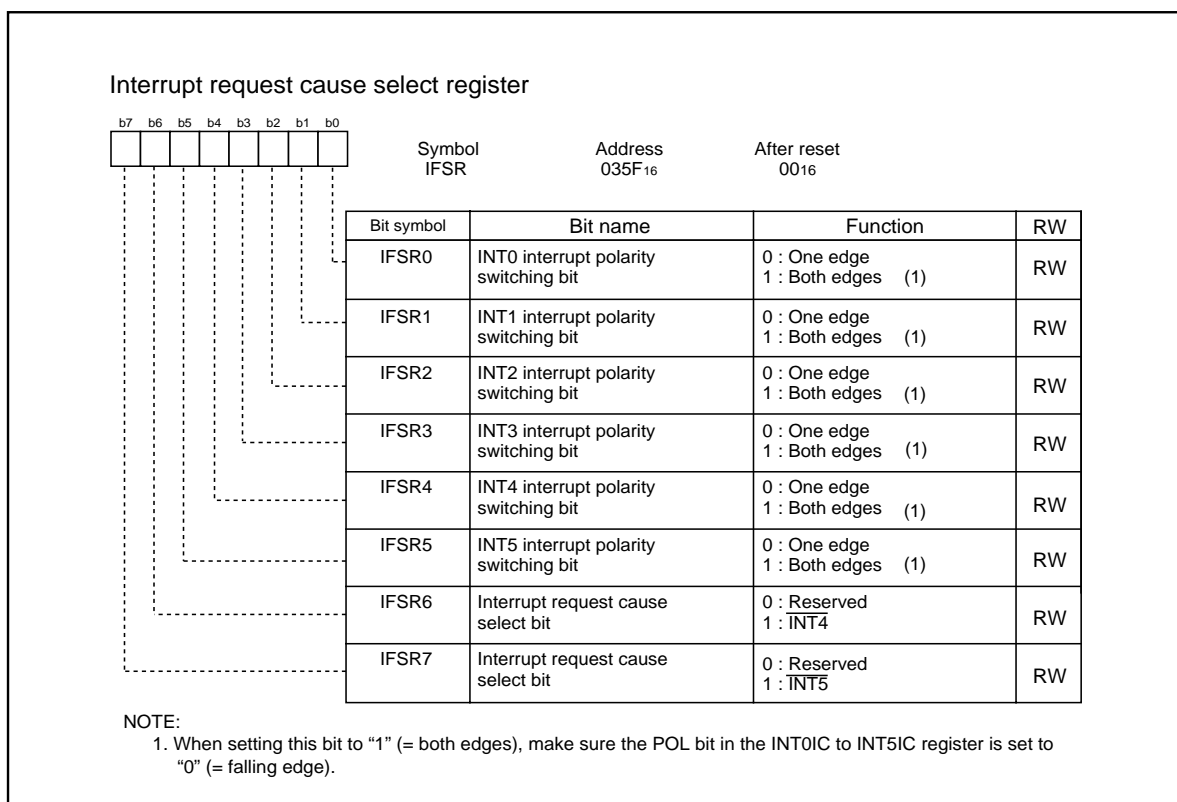


Figure 9.6.1. IFSR Register

## 9.7 NMI Interrupt

An  $\overline{\text{NMI}}$  interrupt request is generated when input on the  $\overline{\text{NMI}}$  pin changes state from high to low, after the  $\overline{\text{NMI}}$  interrupt was enabled by writing a "1" to PM24 bit in the PM2 register. The  $\overline{\text{NMI}}$  interrupt is a non-maskable interrupt, once it is enabled.

The input level of this  $\overline{\text{NMI}}$  interrupt input pin can be read by accessing the P8\_5 bit in the P8 register.

$\overline{\text{NMI}}$  is disabled by default after reset (the pin is a GPIO pin, P85) and can be enabled using PM24 bit in the PM2 register. Once enabled, it can only be disabled by a reset signal.

The  $\overline{\text{NMI}}$  input has an effective digital debounce function for a noise rejection. Refer to **16.6 Digital Debounce Function** for this detail. When using  $\overline{\text{NMI}}$  interrupt to exit stop mode, set the NDDR register to "FF16" before entering stop mode.

## 9.8 Key Input Interrupt

Of P104 to P107, a key input interrupt is generated when input on any of the P104 to P107 pins which has had the PD10\_4 to PD10\_7 bits in the PD10 register set to "0" (= input) goes low. Key input interrupts can be used as a key-on wakeup function, the function which gets the microcomputer out of wait or stop mode. However, if you intend to use the key input interrupt, do not use P104 to P107 as analog input ports. Figure 9.8.1 shows the block diagram of the key input interrupt. Note, however, that while input on any pin which has had the PD10\_4 to PD10\_7 bits set to "0" (= input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

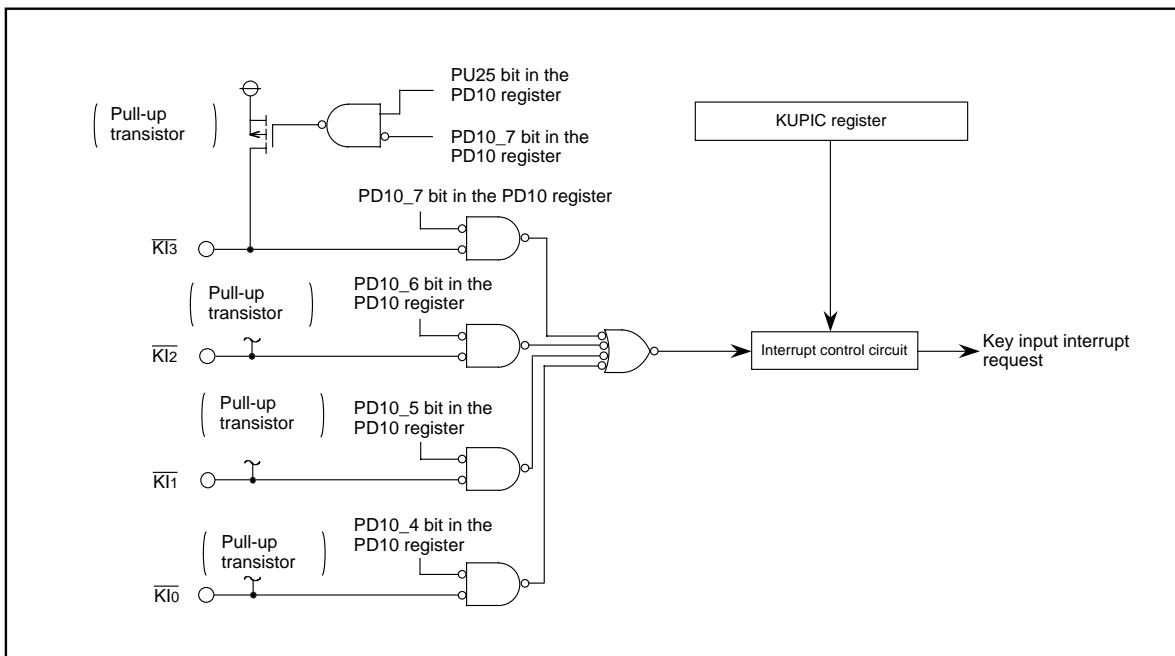


Figure 9.8.1. Key Input Interrupt

## 9.9 Address Match Interrupt

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

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

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

Table 9.9.1 shows the value of the PC that is saved to the stack area when an address match interrupt request is accepted.

Figure 9.9.1 shows the AIER, RMAD0 and RMAD1 registers.

**Table 9.9.1. Value of the PC that is saved to the stack area when an address match interrupt request is accepted.**

Instruction at the address indicated by the RMAD <sub>i</sub> register	Value of the PC that is saved to the stack area																																				
<ul style="list-style-type: none"> <li>• 2-byte op-code instruction</li> <li>• 1-byte op-code instructions which are followed:               <table border="0" style="margin-left: 20px;"> <tr> <td>ADD.B:S</td><td>#IMM8,dest</td><td>SUB.B:S</td><td>#IMM8,dest</td><td>AND.B:S</td><td>#IMM8,dest</td></tr> <tr> <td>OR.B:S</td><td>#IMM8,dest</td><td>MOV.B:S</td><td>#IMM8,dest</td><td>STZ.B</td><td>#IMM8,dest</td></tr> <tr> <td>STNZ.B</td><td>#IMM8,dest</td><td>STZX.B</td><td>#IMM81,#IMM82,dest</td><td></td><td></td></tr> <tr> <td>CMP.B:S</td><td>#IMM8,dest</td><td>PUSHM</td><td>src</td><td>POPM</td><td>dest</td></tr> <tr> <td>JMPS</td><td>#IMM8</td><td>JSRS</td><td>#IMM8</td><td></td><td></td></tr> <tr> <td>MOV.B:S</td><td>#IMM,dest (However, dest=A0 or A1)</td><td></td><td></td><td></td><td></td></tr> </table> </li> </ul>	ADD.B:S	#IMM8,dest	SUB.B:S	#IMM8,dest	AND.B:S	#IMM8,dest	OR.B:S	#IMM8,dest	MOV.B:S	#IMM8,dest	STZ.B	#IMM8,dest	STNZ.B	#IMM8,dest	STZX.B	#IMM81,#IMM82,dest			CMP.B:S	#IMM8,dest	PUSHM	src	POPM	dest	JMPS	#IMM8	JSRS	#IMM8			MOV.B:S	#IMM,dest (However, dest=A0 or A1)					The address indicated by the RMAD <sub>i</sub> register +2
ADD.B:S	#IMM8,dest	SUB.B:S	#IMM8,dest	AND.B:S	#IMM8,dest																																
OR.B:S	#IMM8,dest	MOV.B:S	#IMM8,dest	STZ.B	#IMM8,dest																																
STNZ.B	#IMM8,dest	STZX.B	#IMM81,#IMM82,dest																																		
CMP.B:S	#IMM8,dest	PUSHM	src	POPM	dest																																
JMPS	#IMM8	JSRS	#IMM8																																		
MOV.B:S	#IMM,dest (However, dest=A0 or A1)																																				
Instructions other than the above	The address indicated by the RMAD <sub>i</sub> register +1																																				

Value of the PC that is saved to the stack area : Refer to "Saving Registers".

Op-code is an abbreviation of Operation Code. It is a portion of instruction code.

Refer to Chapter 4 Instruction Code/Number of Cycles in M16C/60, M16C/20 Series Software Manual. Op-code is shown as a bold-framed figure directly below the Syntax.

**Table 9.9.2. Relationship Between Address Match Interrupt Sources and Associated Registers**

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

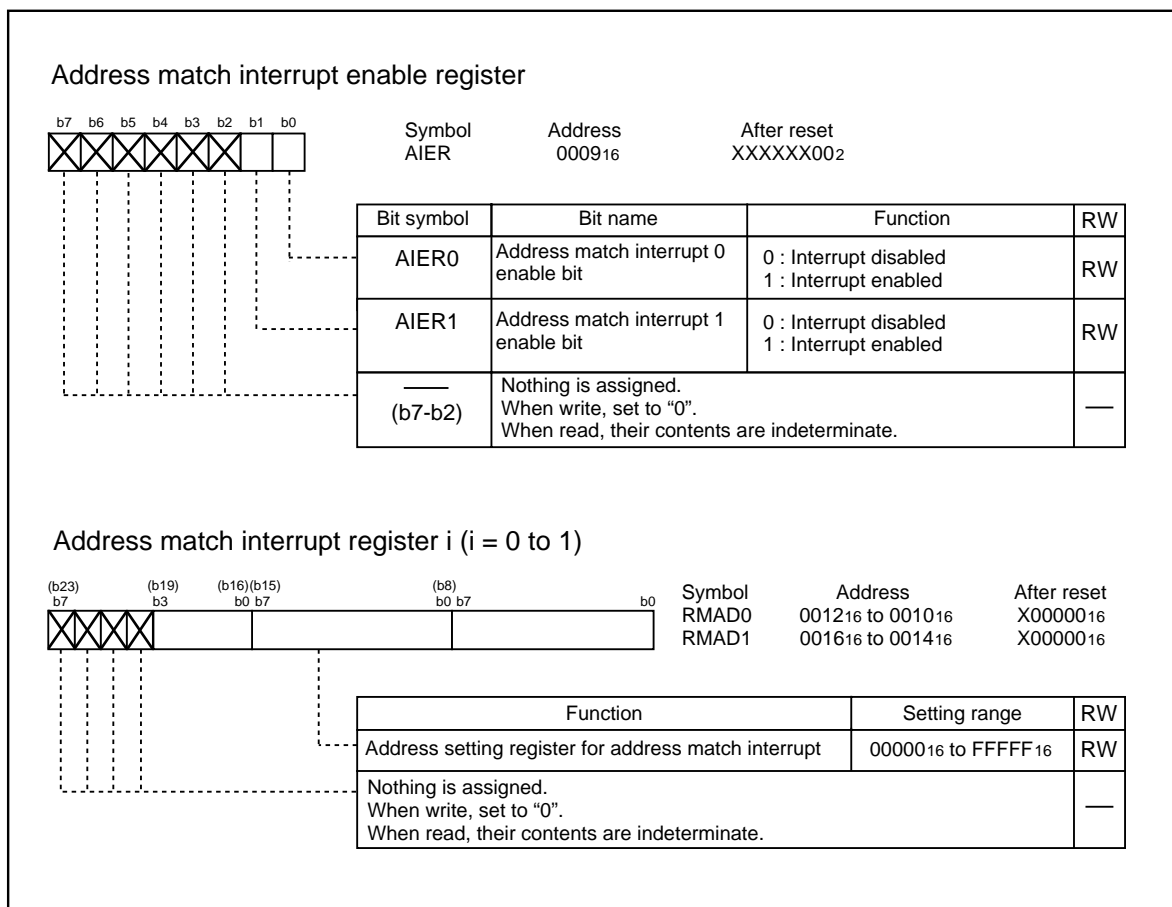


Figure 9.9.1. AIER Register, RMAD0 and RMAD1 Registers



## 10. Watchdog Timer

The watchdog timer is the function that detects when a program is out of control. Use the watchdog timer is recommended to improve reliability of the system. The watchdog timer contains a 15-bit counter which is decremented by the CPU clock that the prescaler divides. The PM12 bit in the PM1 register determines whether to generate a watchdog timer interrupt request or reset the watchdog timer when the watchdog timer underflows. The PM12 bit can only be set to "1" (reset). Once the PM12 bit is set to "1", it cannot be changed to "0" (watchdog timer interrupt) by program. Refer to "5.3 Watchdog Timer Reset" for watchdog timer reset.

When the main clock, on-chip oscillator clock, or PLL clock runs as CPU clock, the WDC7 bit in the WDC register determines whether the prescaler divides the clock by 16 or 128. When the sub clock runs as CPU clock, the prescaler divides the clock by 2 regardless of the WDC7 bit setting. Watchdog timer cycle is calculated as follows. Marginal errors, due to the prescaler, may occur in watchdog timer cycle.

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

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

With sub-clock chosen for CPU clock

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

For example, when CPU clock = 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.

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

In stop mode, wait mode and when erase/program operation is executing in EW1 mode without erase suspend required, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.

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

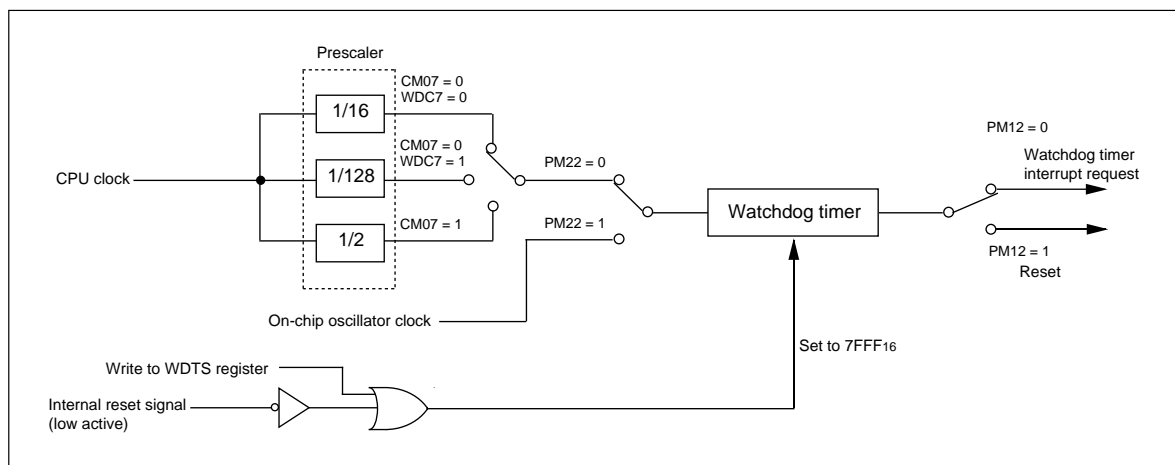


Figure 10.1. Watchdog Timer Block Diagram

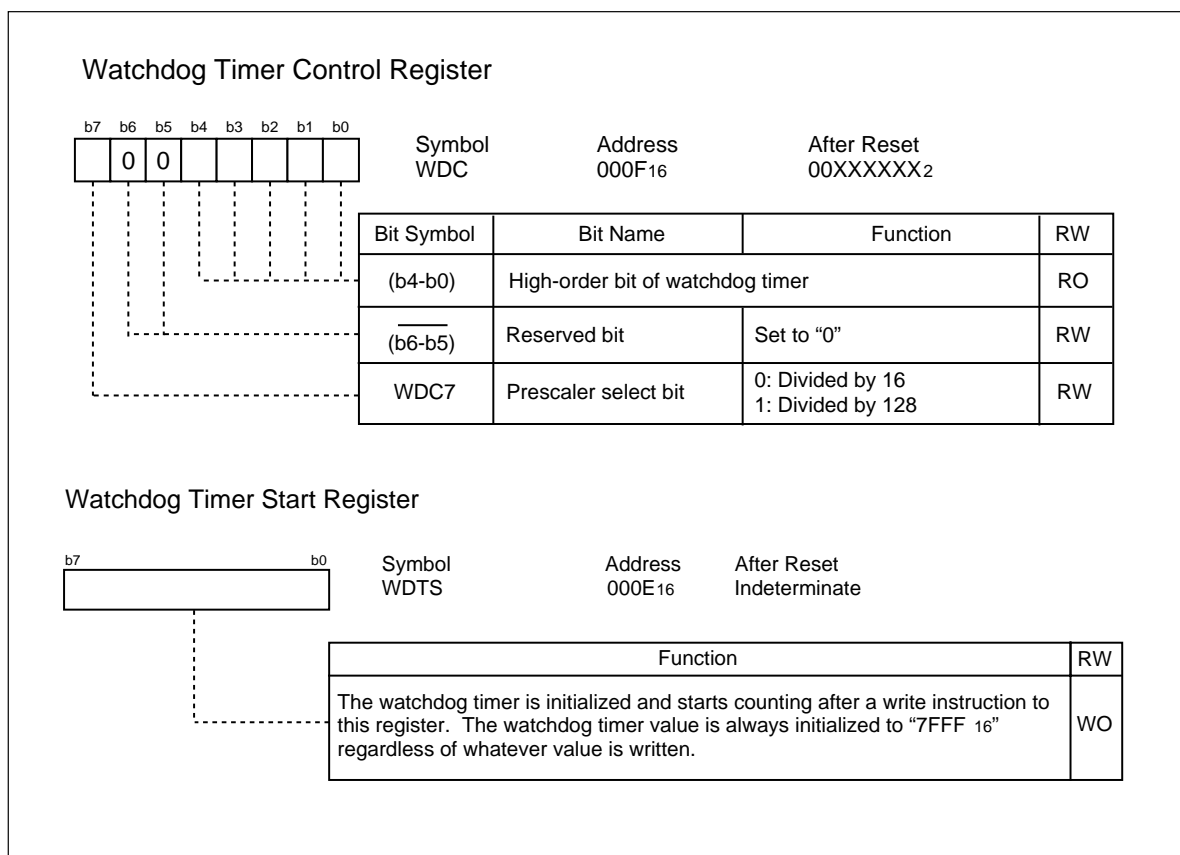


Figure 10.2 WDC Register and WDTS Register

### 10.1 Count Source Protective Mode

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

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

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

Setting the PM22 bit to "1" results in the following conditions

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

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

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

## 11. DMAC

### Note

Do not use UART0 transfer and UART0 reception interrupt request as a DMA request in the 42-pin package.

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

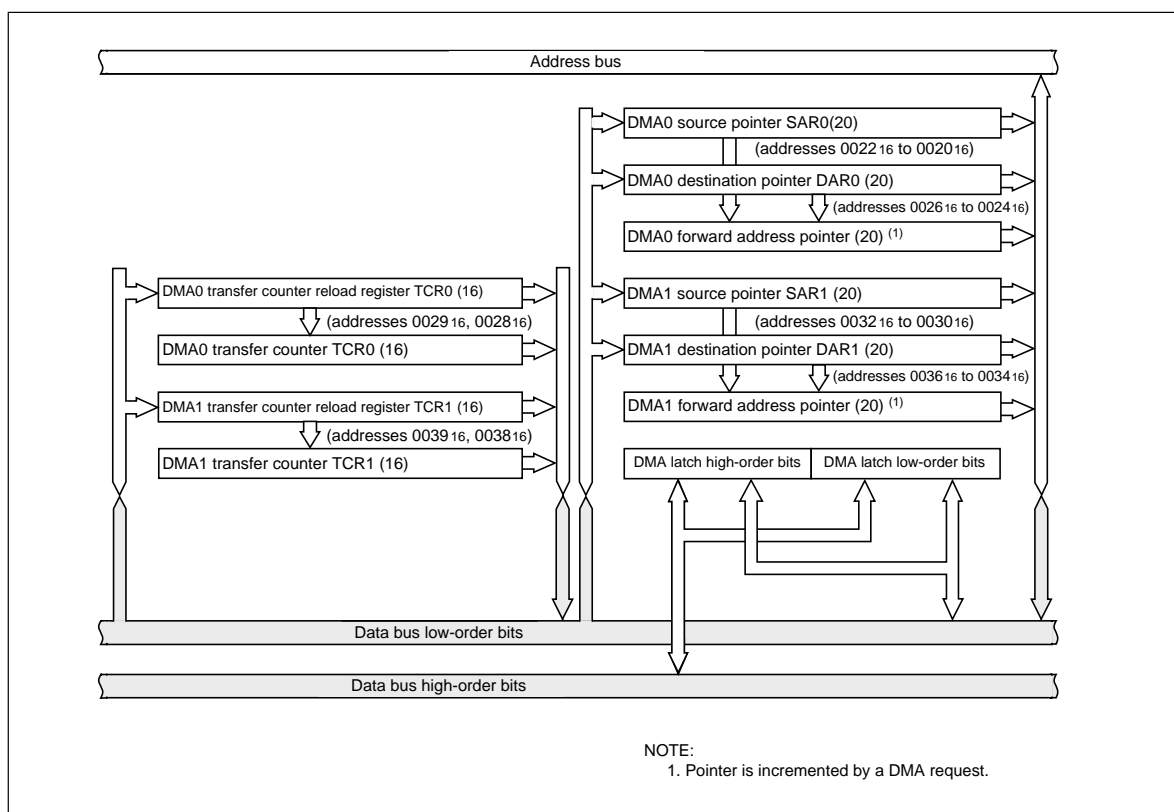


Figure 11.1 DMAC Block Diagram

A DMA request is generated by a write to the DSR bit in the DMiSL register ( $i = 0, 1$ ), as well as by an interrupt request which is generated by any function specified by the DMS and DSEL3 to DSEL0 bits in the DMiSL register. However, unlike in the case of interrupt requests, DMA requests are not affected by the I flag and the interrupt control register, so that even when interrupt requests are disabled and no interrupt request can be accepted, DMA requests are always accepted. Furthermore, because the DMAC does not affect interrupts, the IR bit in the interrupt control register does not change state due to a DMA transfer.

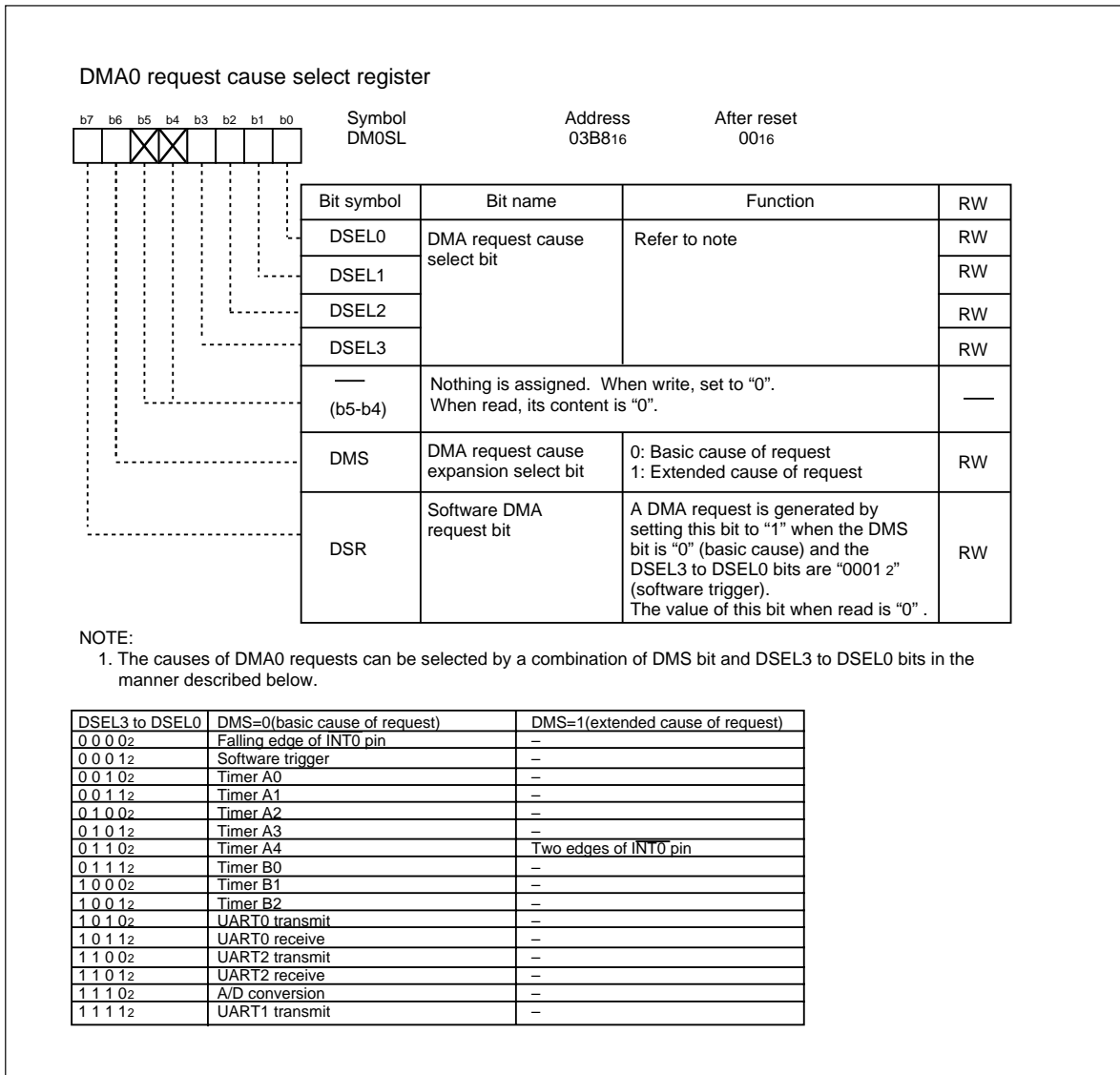
A data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMiCON register is set to "1" (DMA enabled). However, if the cycle in which a DMA request is generated is faster than the DMA transfer cycle, the number of transfer requests generated and the number of times data is transferred may not match. For details, refer to **11.4 DMA Requests**.

**Table 11.1 DMAC Specifications**

Item		Specification
No. of channels		2 (cycle steal method)
Transfer memory space		<ul style="list-style-type: none"> <li>• From any address in the 1M bytes space to a fixed address</li> <li>• From a fixed address to any address in the 1M bytes space</li> <li>• From a fixed address to a fixed address</li> </ul>
Maximum No. of bytes transferred		128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors <sup>(1, 2)</sup>		Falling edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Both edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transfer, UART0 reception interrupt requests UART1 transfer, UART1 reception interrupt requests UART2 transfer, UART2 reception interrupt requests 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 <sub>i</sub> transfer counter (i = 0,1) underflows after reaching the terminal count.
	Repeat transfer	When the DMA <sub>i</sub> transfer counter underflows, it is reloaded with the value of the DMA <sub>i</sub> transfer counter reload register and a DMA transfer is continued with it.
DMA interrupt request generation timing		When the DMA <sub>i</sub> transfer counter underflows
DMA startup		Data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMA <sub>i</sub> CON register is set to "1" (enabled).
DMA shutdown	Single transfer	<ul style="list-style-type: none"> <li>• When the DMAE bit is set to "0" (disabled)</li> <li>• After the DMA<sub>i</sub> transfer counter underflows</li> </ul>
	Repeat transfer	When the DMAE bit is set to "0" (disabled)
		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 <sub>i</sub> or the DAR <sub>i</sub> pointer whichever is specified to be in the forward direction and the DMA <sub>i</sub> transfer counter is reloaded with the value of the DMA <sub>i</sub> transfer counter reload register.

**NOTES:**

1. DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the I flag nor by the interrupt control register.
2. The selectable causes of DMA requests differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0020<sub>16</sub> to 003F<sub>16</sub>) are accessed by the DMAC.



**Figure 11.2 DM0SL Register**

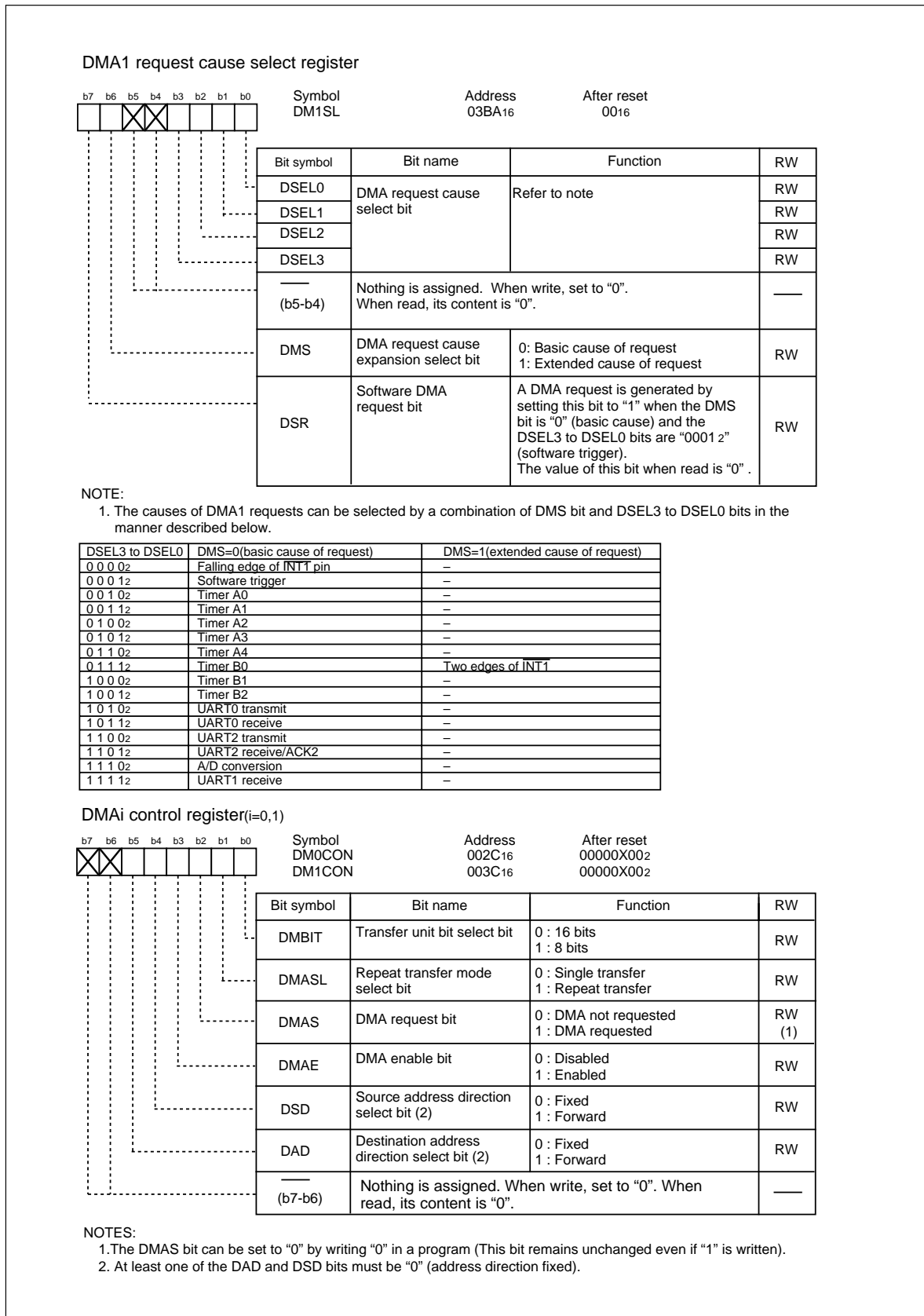


Figure 11.3 DM1SL Register, DM0CON Register, and DM1CON Register

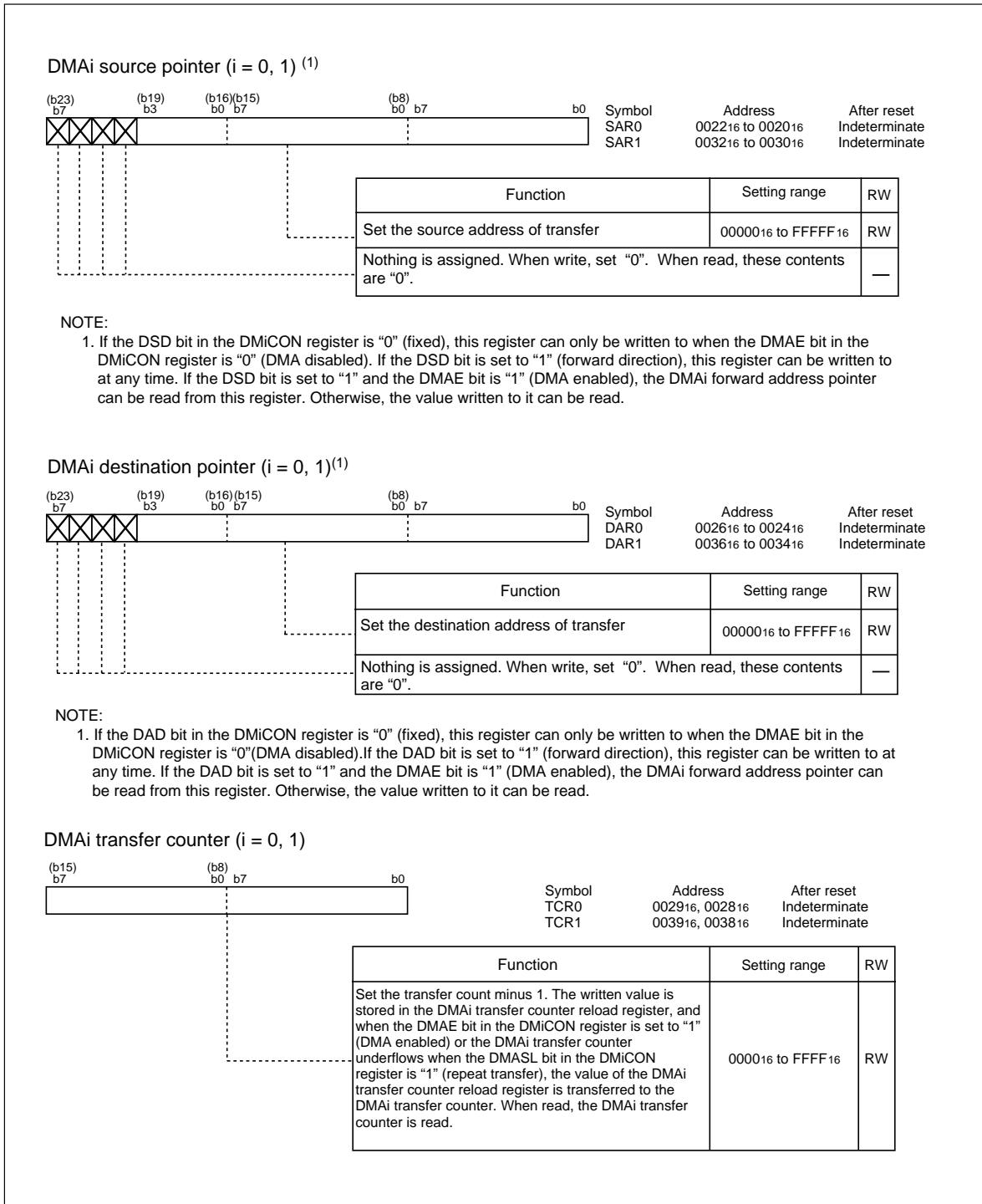


Figure 11.4 SAR0 and SAR1, DAR0 and DAR1, TCR0 and TCR1 Registers

## 11.1 Transfer Cycles

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

### 11.1.1 Effect of Source and Destination Addresses

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

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

### 11.1.2 Effect of Software Wait

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

Figure 11.1.1 shows the example of the cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16 bit units and when both the source address and destination address are an odd address ((2) in Figure 11.1.1), two source read bus cycles and two destination write bus cycles are required.



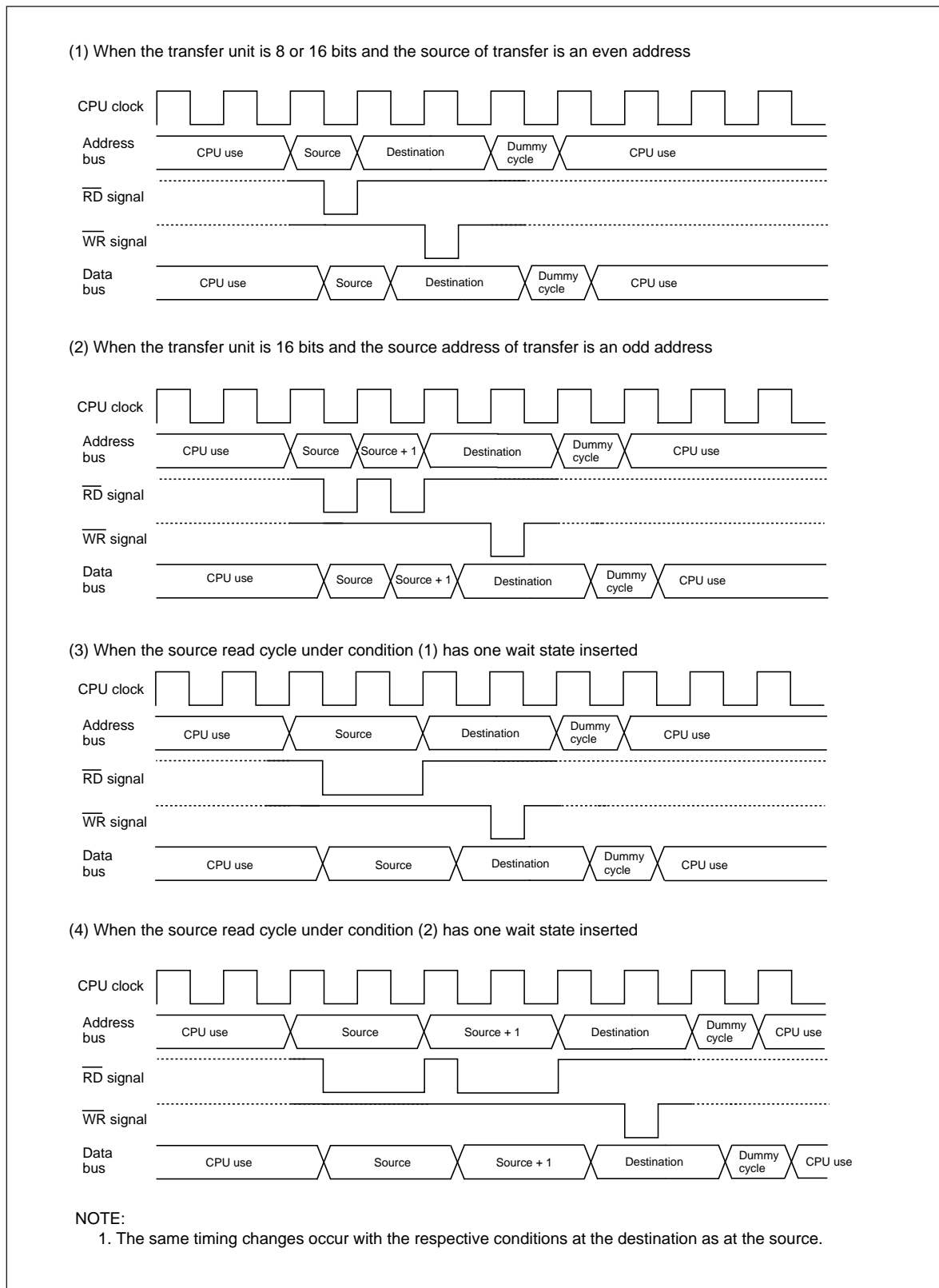


Figure 11.1.1 Transfer Cycles for Source Read

## 11.2. DMA Transfer Cycles

Any combination of even or odd transfer read and write addresses is possible. Table 11.2.1 shows the number of DMA transfer cycles. Table 11.2.2 shows the Coefficient j, k.

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

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

**Table 11.2.1 DMA Transfer Cycles**

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

**Table 11.2.2 Coefficient j, k**

	Internal area			
	Internal ROM, RAM		SFR	
	No wait	With wait	1 wait (1)	2 wait (1)
j	1	2	2	3
k	1	2	2	3

NOTE:

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

### 11.3 DMA Enable

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

- (1) Reload the forward address pointer with the SAR<sub>i</sub> register value when the DSD bit in the DMiCON register is “1” (forward) or the DAR<sub>i</sub> register value when the DAD bit in the DMiCON register is “1” (forward).
- (2) Reload the DMA<sub>i</sub> transfer counter with the DMA<sub>i</sub> 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 DMiCON register simultaneously.

Step 2: Make sure that the DMA<sub>i</sub> is in an initial state as described above (1) and (2) in a program.

If the DMA<sub>i</sub> is not in an initial state, the above steps should be repeated.

### 11.4 DMA Request

The DMAC can generate a DMA request as triggered by the cause of request that is selected with the DMS and DSEL3 to DSEL0 bits in the DMiSL register ( $i = 0, 1$ ) on either channel. Table 11.4.1 shows the timing at which the DMAS bit changes state.

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

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

Because if the DMAE bit is “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 11.4.1 Timing at Which the DMAS Bit Changes State**

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

### 11.5 Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of CPU clock), the DMAS bit on each channel is set to "1" (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1. The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period. Figure 11.5.1 shows an example of DMA transfer effected by external factors.

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

In addition, DMA requests cannot be counted up since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in Figure 11.5.1, occurs more than one time, the DMAS bit is set to "0" as soon as getting the bus arbitration. The bus arbitration is returned to the CPU when one transfer is completed.

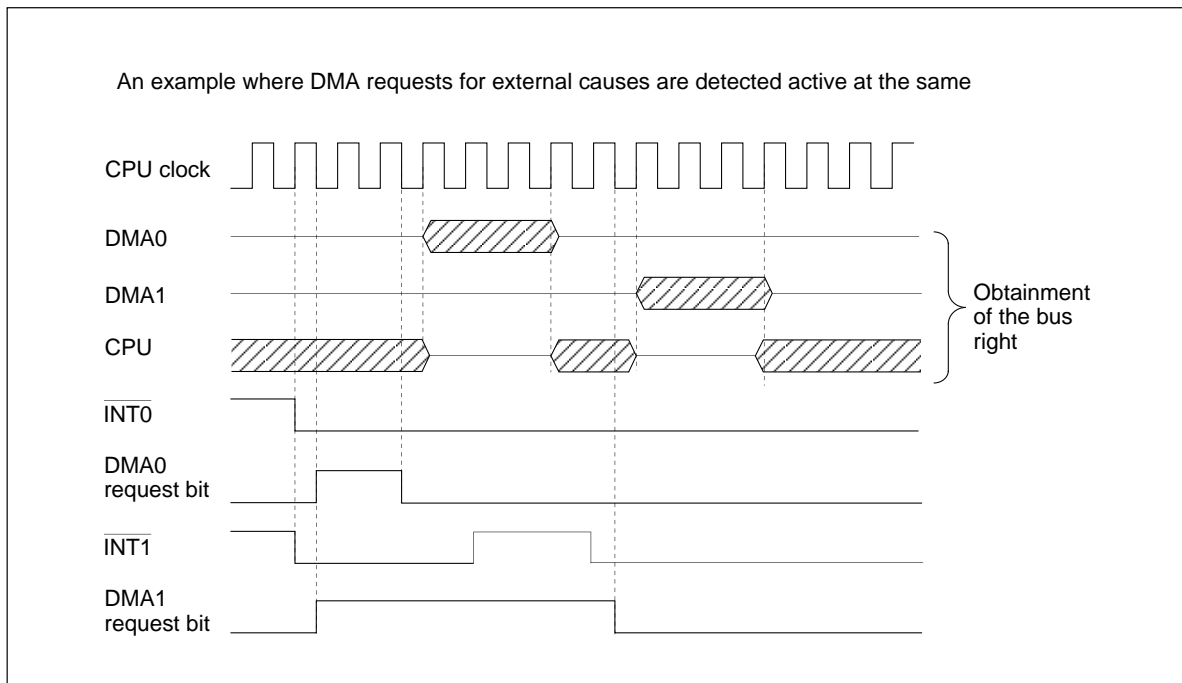


Figure 11.5.1 DMA Transfer by External Factors

## 12. Timer

### Note

The TB2IN pin is not available in the 42-pin package. Do not use functions associated with the TB2IN pin.

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

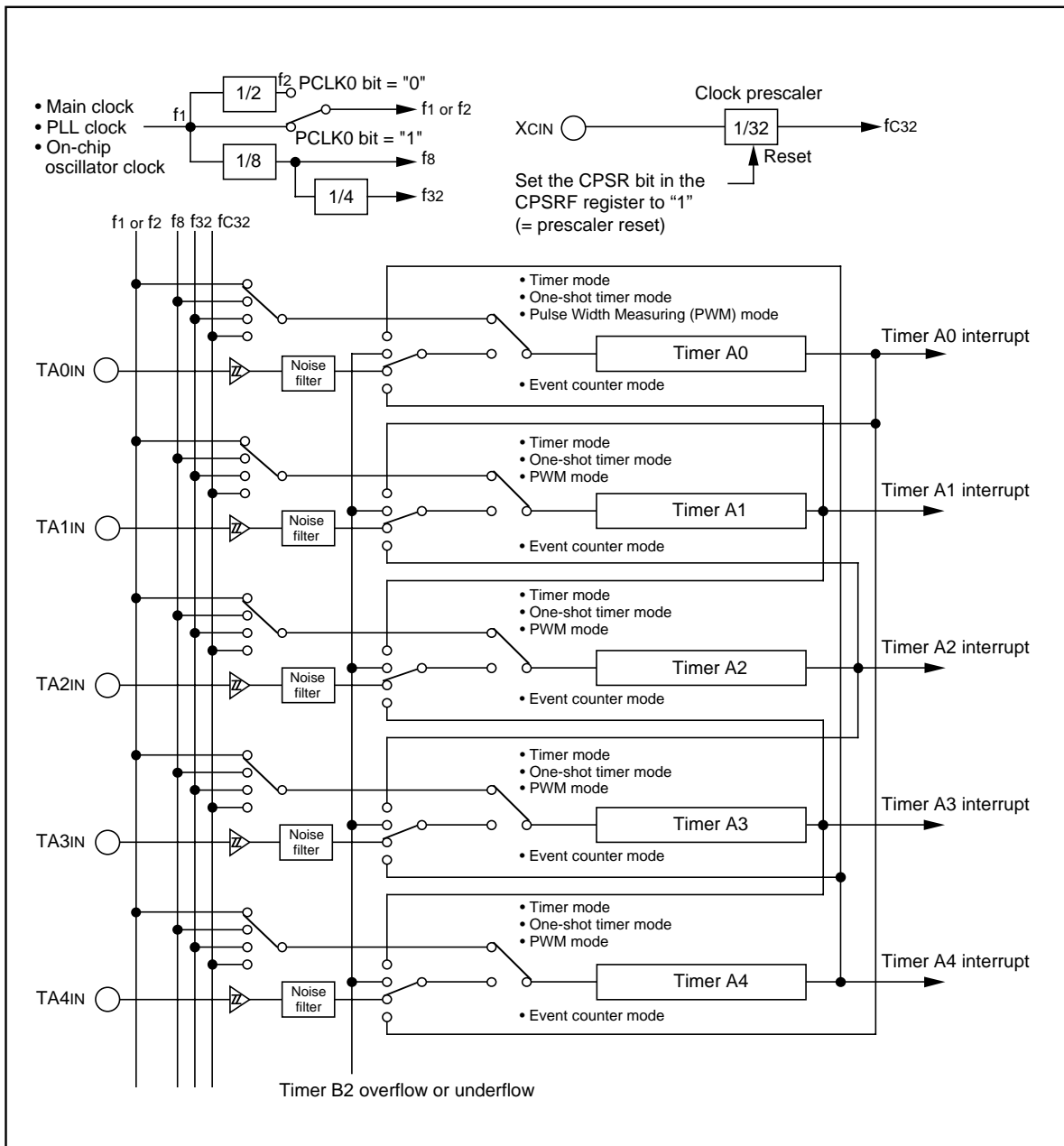


Figure 12.1. Timer A Configuration

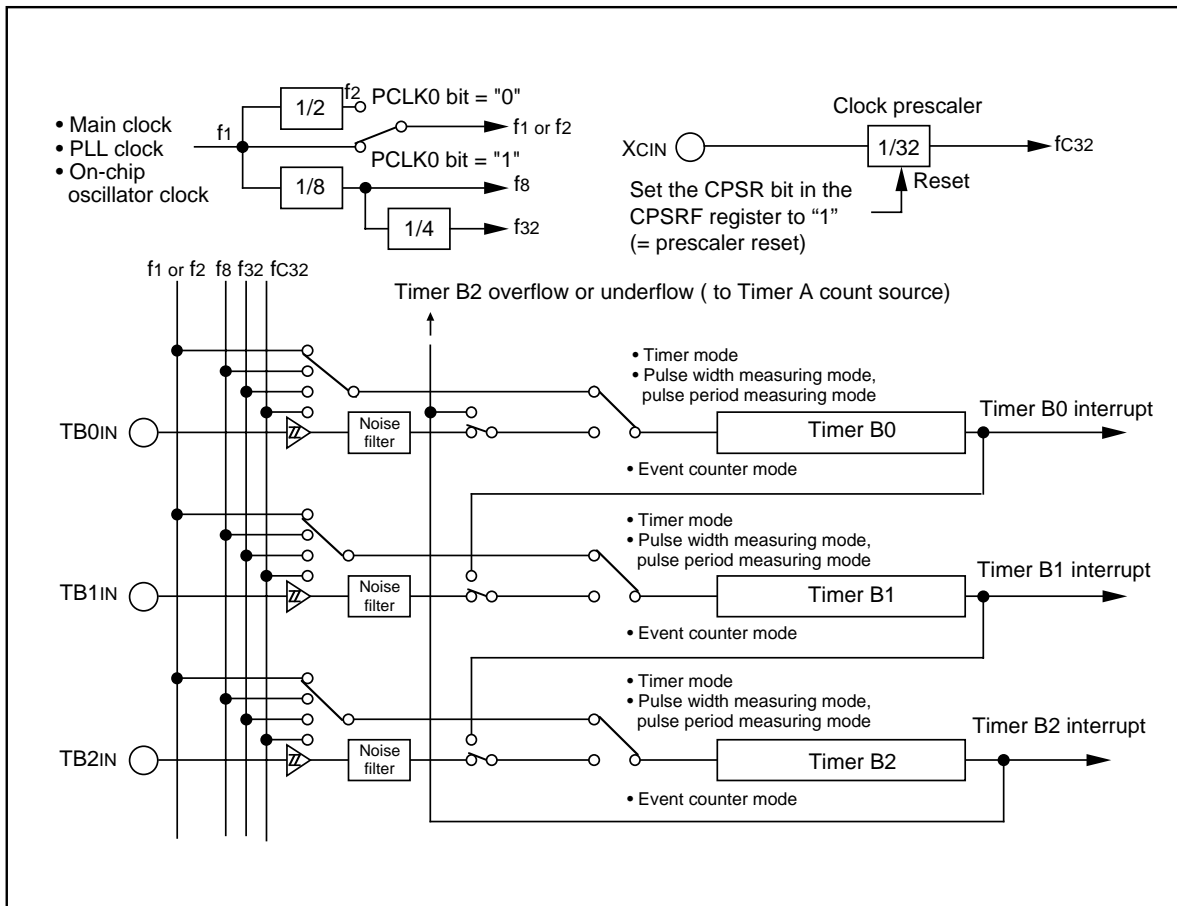


Figure 12.2. Timer B Configuration

### 12.1 Timer A

Figure 12.1.1 shows a block diagram of the timer A. Figures 12.1.2 to 12.1.4 show registers related to the timer A.

The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use the TMOD1 to TMOD0 bits in the TAIiMR 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 “0000<sub>16</sub>.”
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.

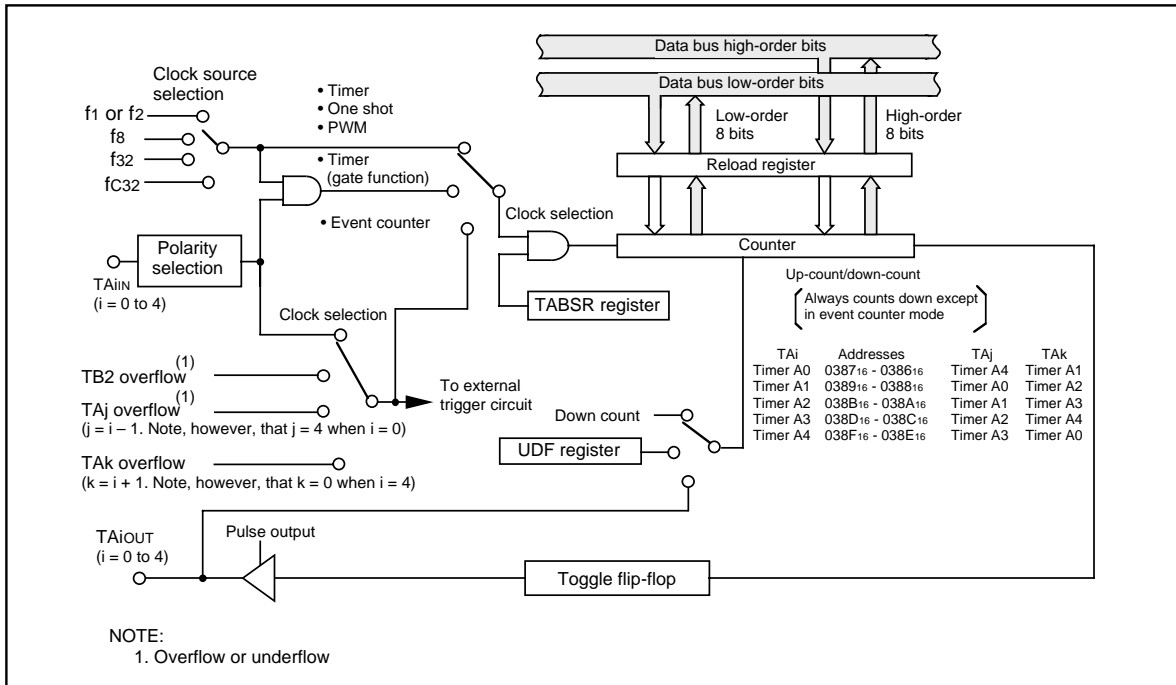


Figure 12.1.1. Timer A Block Diagram

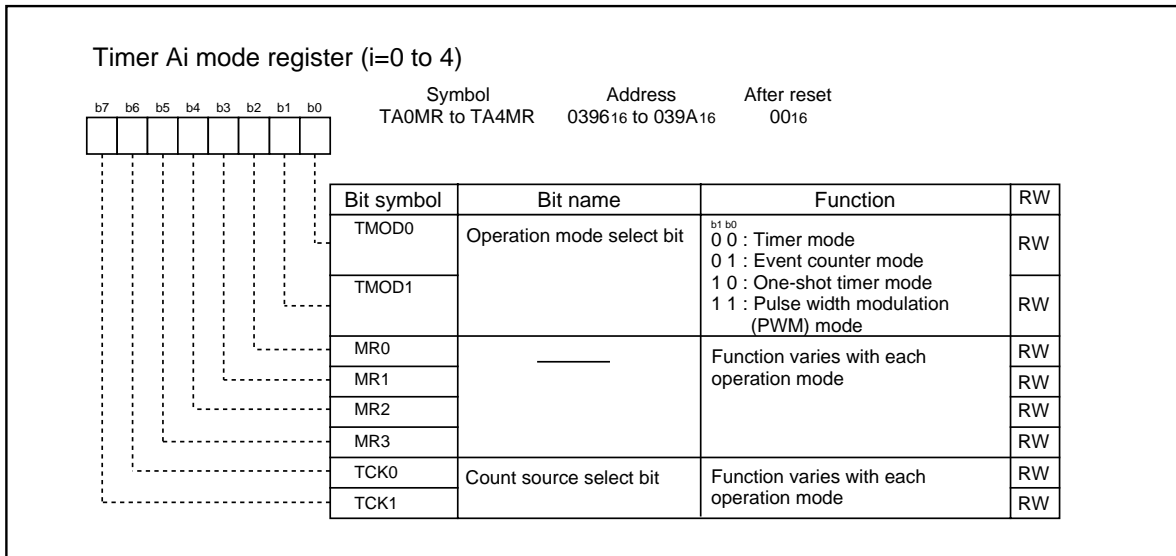


Figure 12.1.2. TA0MR to TA4MR Registers

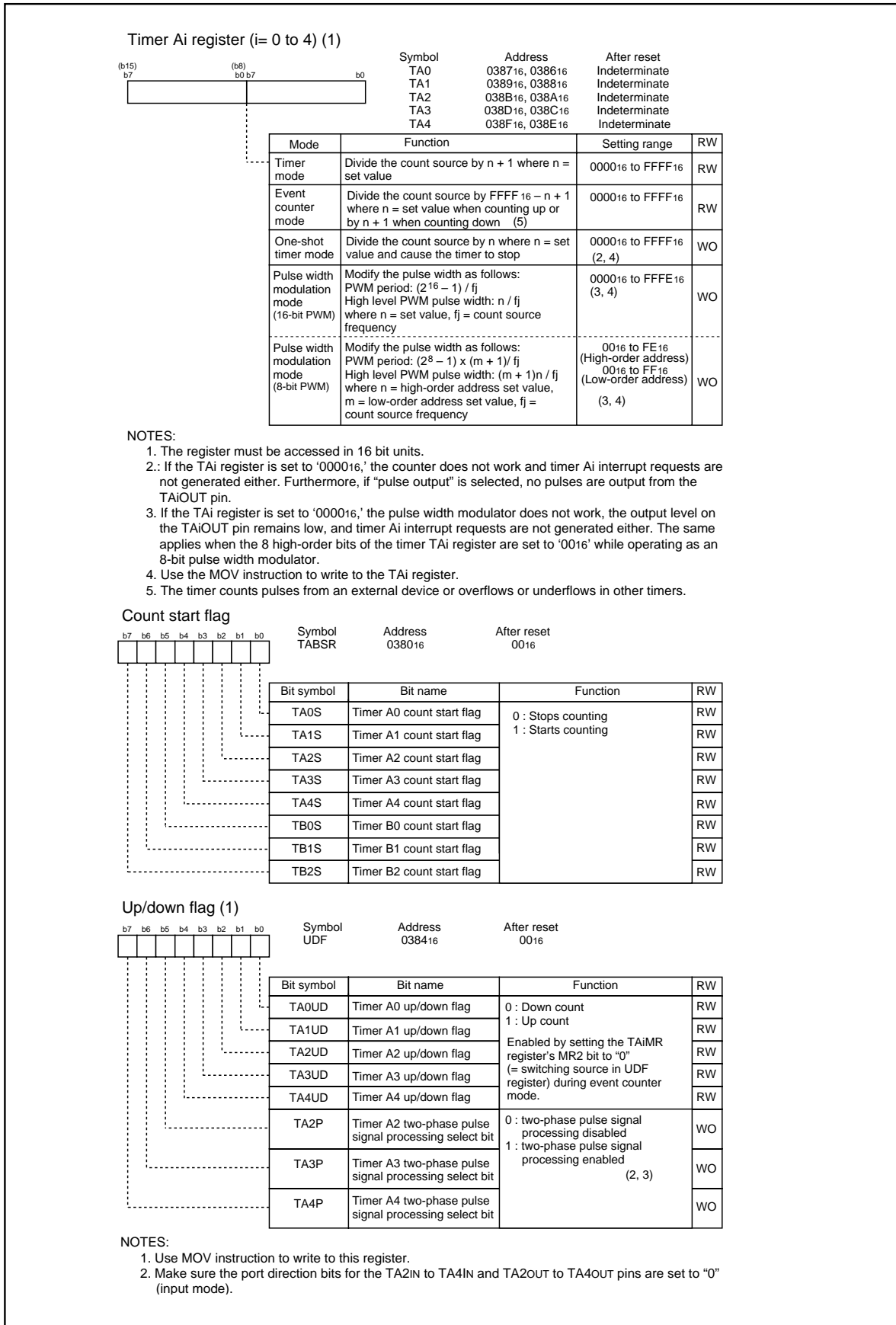


Figure 12.1.3. TA0 to TA4 Registers, TABSR Register, and UDF Register



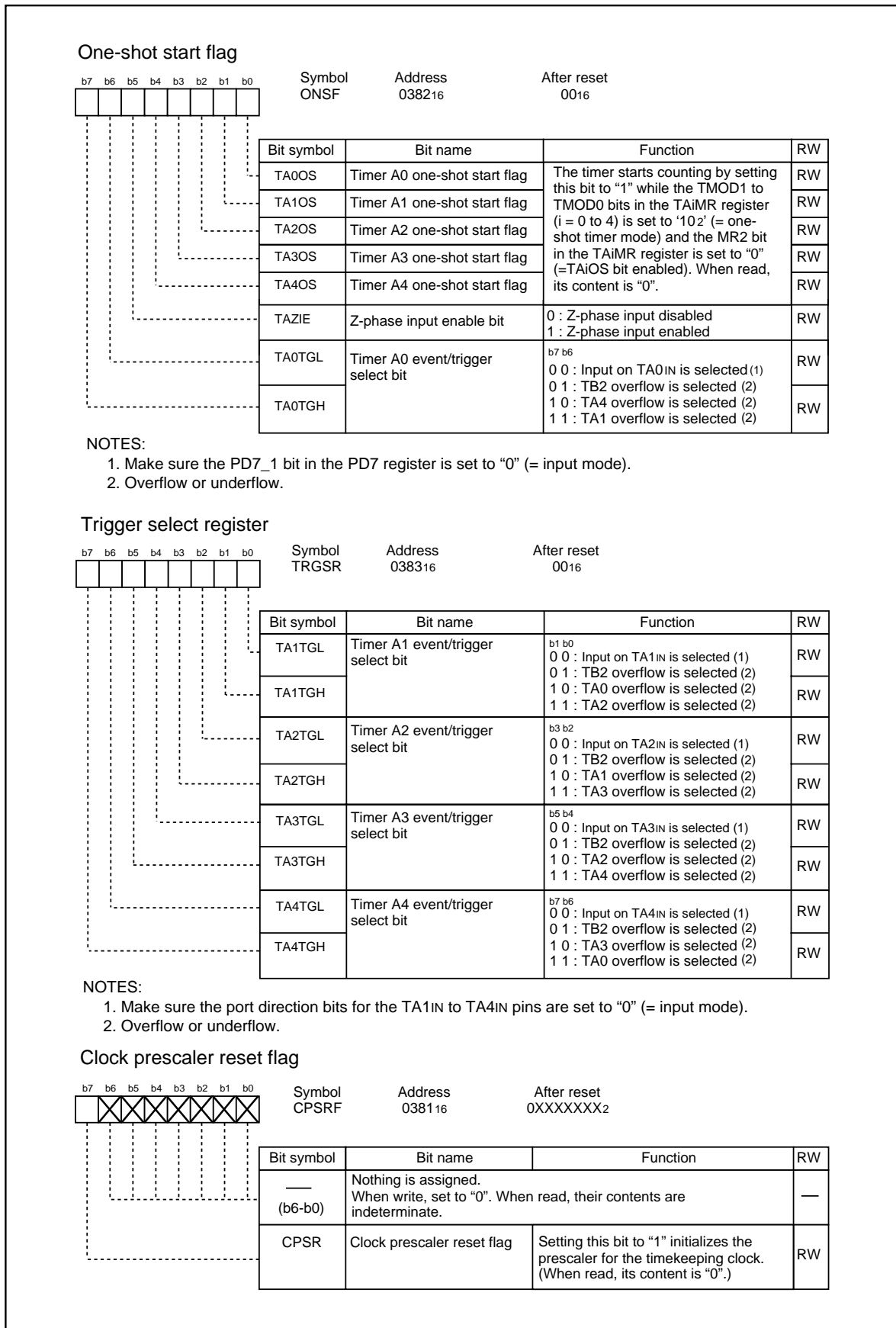


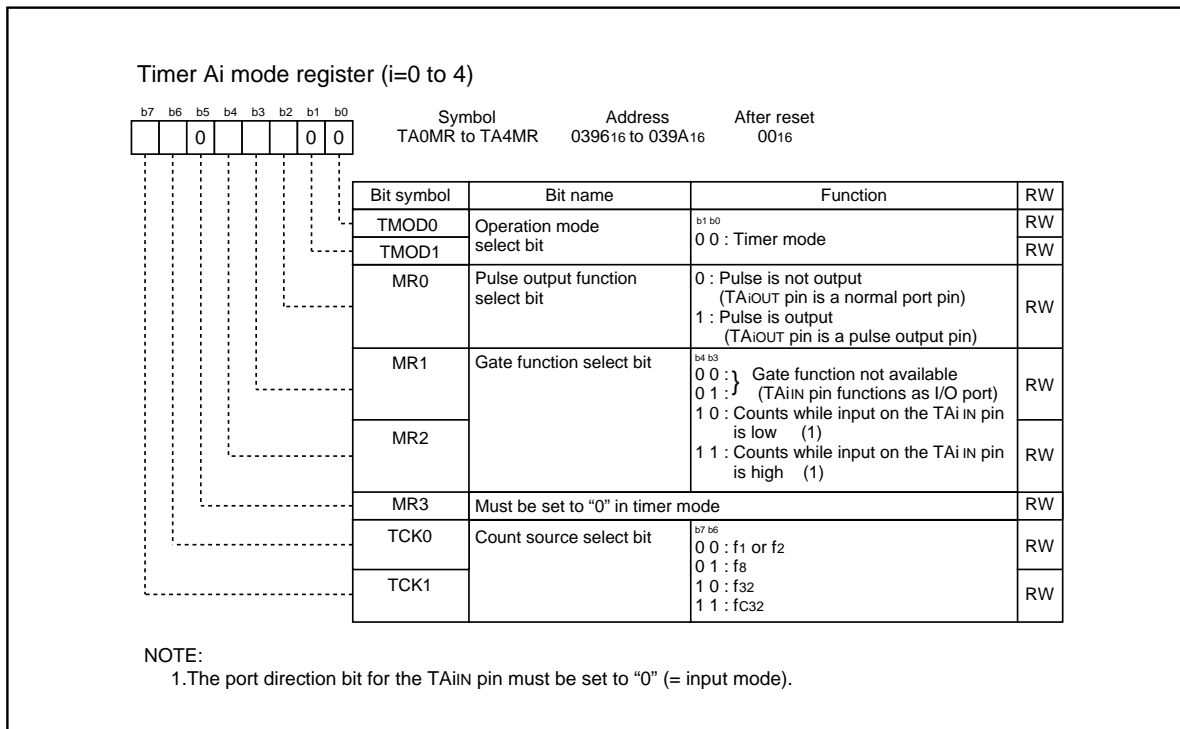
Figure 12.1.4. ONSF Register, TRGSR Register, and CPSRF Register

### 12.1.1. Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 12.1.1.1). Figure 1.2.1.1.1 shows TAI<sub>i</sub>MR register in timer mode.

**Table 12.1.1.1. Specifications in Timer Mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>2</sub> , f <sub>8</sub> , f <sub>32</sub> , f <sub>C32</sub>
Count operation	<ul style="list-style-type: none"> <li>Down-count</li> <li>When the timer underflows, it reloads the reload register contents and continues counting</li> </ul>
Divide ratio	1/(n+1) n: set value of TAI register (i= 0 to 4) 0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	Set TAI <sub>S</sub> bit in the TABSR register to "1" (= start counting)
Count stop condition	Set TAI <sub>S</sub> bit to "0" (= stop counting)
Interrupt request generation timing	Timer underflow
TAI <sub>IN</sub> pin function	I/O port or gate input
TAI <sub>OUT</sub> pin function	I/O port or pulse output
Read from timer	Count value can be read by reading TAI register
Write to timer	<ul style="list-style-type: none"> <li>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</li> <li>When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Gate function Counting can be started and stopped by an input signal to TAI<sub>IN</sub> pin</li> <li>Pulse output function Whenever the timer underflows, the output polarity of TAI<sub>OUT</sub> pin is inverted. When not counting, the pin outputs a low.</li> </ul>



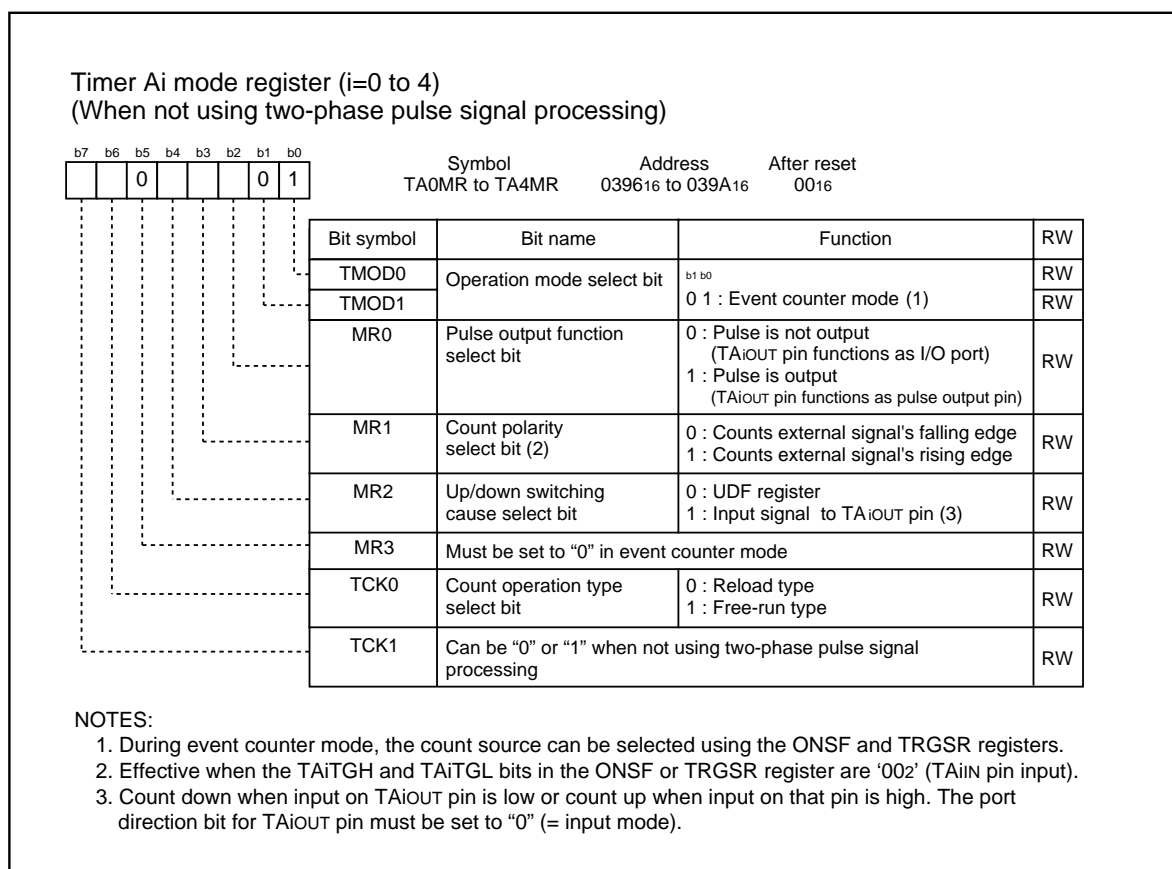
**Figure 12.1.1.1. Timer Ai Mode Register in Timer Mode**

### 12.1.2. Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3 and A4 can count two-phase external signals. Table 12.1.2.1 lists specifications in event counter mode (when not processing two-phase pulse signal). Table 12.1.2.2 lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4). Figure 12.1.2.1 shows TAI<sub>MR</sub> register in event counter mode (when not processing two-phase pulse signal). Figure 12.1.2.2 shows TA2<sub>MR</sub> to TA4<sub>MR</sub> registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

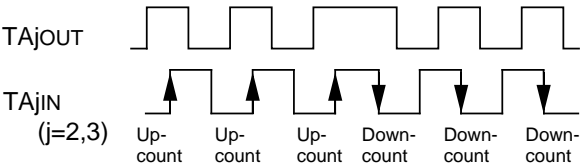
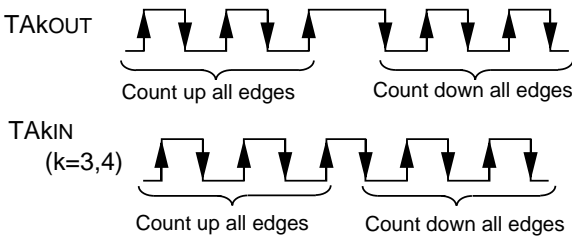
**Table 12.1.2.1. Specifications in Event Counter Mode (when not processing two-phase pulse signal)**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signals input to TAI<sub>IN</sub> pin (i=0 to 4) (effective edge can be selected in program)</li> <li>Timer B2 overflows or underflows,</li> <li>timer A<sub>j</sub> (j=i-1, except j=4 if i=0) overflows or underflows,</li> <li>timer A<sub>k</sub> (k=i+1, except k=0 if i=4) overflows or underflows</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Up-count or down-count can be selected by external signal or program</li> <li>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.</li> </ul>
Divided ratio	1/ (FFFF <sub>16</sub> - n + 1) for up-count 1/ (n + 1) for down-count    n : set value of TAI register    0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	Set TAI <sub>S</sub> bit in the TABSR register to "1" (= start counting)
Count stop condition	Set TAI <sub>S</sub> bit to "0" (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAI <sub>IN</sub> pin function	I/O port or count source input
TAI <sub>OUT</sub> pin function	I/O port, pulse output, or up/down-count select input
Read from timer	Count value can be read by reading TAI register
Write to timer	<ul style="list-style-type: none"> <li>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</li> <li>When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it</li> <li>Pulse output function Whenever the timer underflows or underflows, the output polarity of TAI<sub>OUT</sub> pin is inverted . When not counting, the pin outputs a low.</li> </ul>



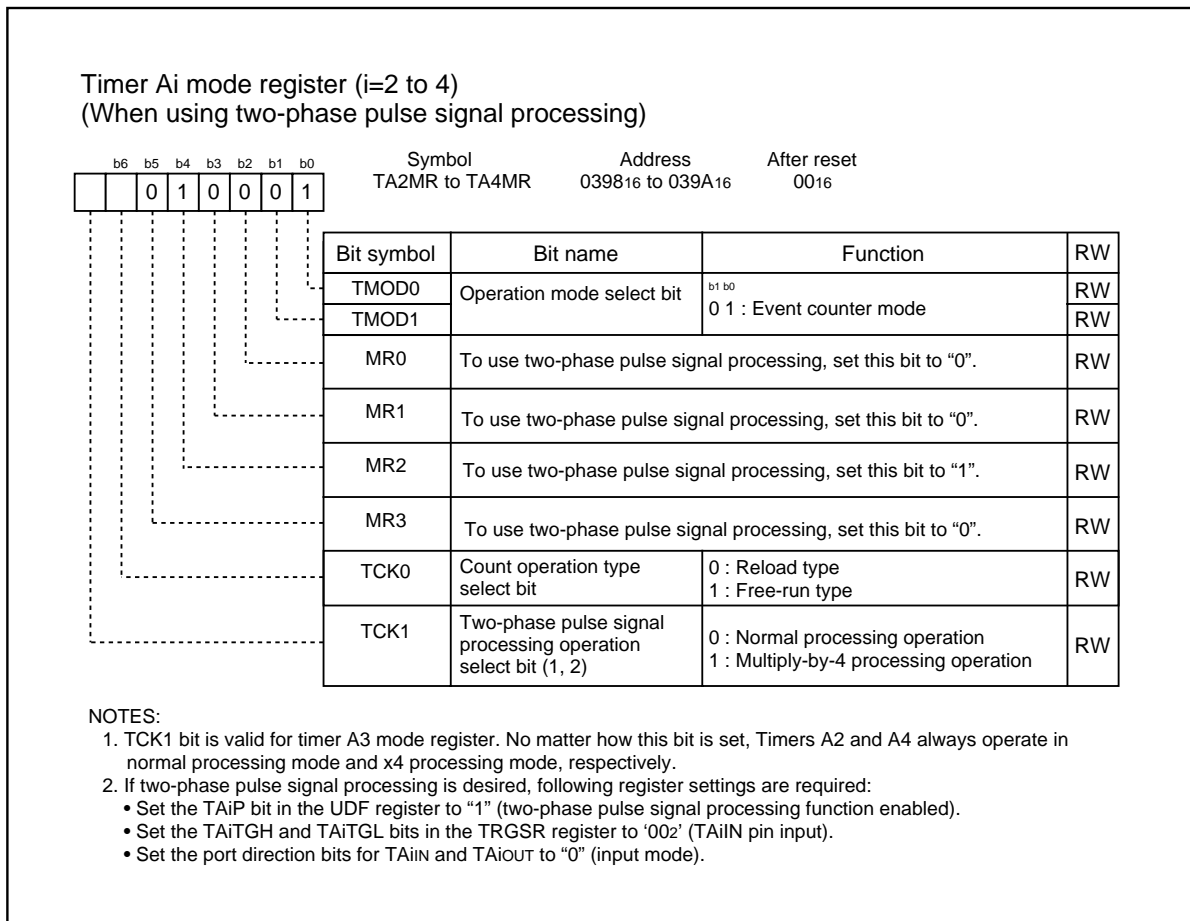
**Figure 12.1.2.1. TA<sub>i</sub>MR Register in Event Counter Mode (when not using two-phase pulse signal processing)**

Table 12.1.2.2. Specifications in Event Counter Mode (when processing two-phase pulse signal with timers A2, A3 and A4)

Item	Specification
Count source	• Two-phase pulse signals input to TAIIN or TAIOUT pins (i = 2 to 4)
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/ (FFFF <sub>16</sub> - n + 1) for up-count 1/ (n + 1) for down-count    n : set value of TAI register    0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	Set TAI <sub>S</sub> bit in the TABSR register to "1" (= start counting)
Count stop condition	Set TAI <sub>S</sub> bit to "0" (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read by reading timer A2, A3 or A4 register
Write to timer	• 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 <sup>(1)</sup>	<ul style="list-style-type: none"> <li>Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on TAJIN (j=2, 3) pin when input signals on TAJOUT pin is "H".</li> </ul>  <ul style="list-style-type: none"> <li>Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAKIN(k=3, 4) pin goes "H" when the input signal on TAKOUT pin is "H", the timer counts up rising and falling edges on TAKOUT and TAKIN pins. If the phase relationship is such that TAKIN pin goes "L" when the input signal on TAKOUT pin is "H", the timer counts down rising and falling edges on TAKOUT and TAKIN pins.</li> </ul>  <ul style="list-style-type: none"> <li>Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input.</li> </ul>

## NOTE:

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



**Figure 12.1.2.2. TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2, A3 or A4)**

### 12.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing

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

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

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

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

The counter is initialized at the next count timing after recognizing Z-phase input. Figure 12.1.2.1.1 shows the relationship between the two-phase pulse (A phase and B phase) and the Z phase.

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

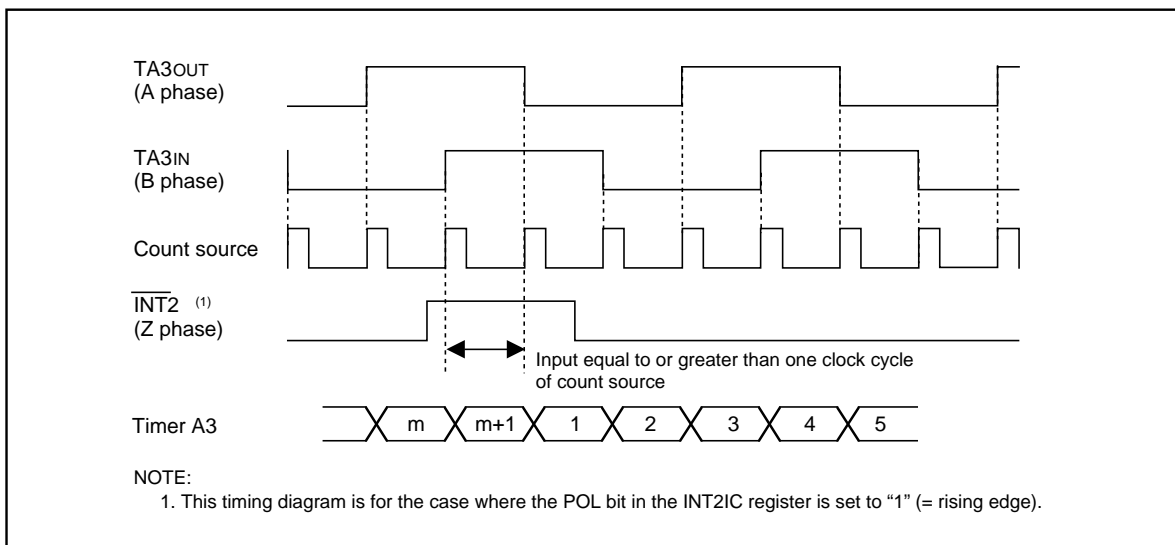


Figure 12.1.2.1.1. Two-phase Pulse (A phase and B phase) and the Z Phase

### 12.1.3. One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. (See Table 12.1.3.1.) When the trigger occurs, the timer starts up and continues operating for a given period. Figure 12.1.3.1 shows the TAI<sub>MR</sub> register in one-shot timer mode.

**Table 12.1.3.1. Specifications in One-shot Timer Mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>2</sub> , f <sub>8</sub> , f <sub>32</sub> , f <sub>C32</sub>
Count operation	<ul style="list-style-type: none"> <li>Down-count</li> <li>When the counter reaches 0000<sub>16</sub>, it stops counting after reloading a new value</li> <li>If a trigger occurs when counting, the timer reloads a new count and restarts counting</li> </ul>
Divide ratio	1/n    n : set value of TAI register 0000 <sub>16</sub> to FFFF <sub>16</sub> However, the counter does not work if the divide-by-n value is set to 0000 <sub>16</sub> .
Count start condition	<p>TAIS bit in the TABSR register is set to "1" (start counting) and one of the following triggers occurs.</p> <ul style="list-style-type: none"> <li>External trigger input from the TAI<sub>IN</sub> pin</li> <li>Timer B2 overflow or underflow, timer A<sub>j</sub> (j=i-1, except j=4 if i=0) overflow or underflow, timer A<sub>k</sub> (k=i+1, except k=0 if i=4) overflow or underflow</li> <li>The TAIOS bit in the ONSF register is set to "1" (= timer starts)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>When the counter is reloaded after reaching "0000<sub>16</sub>"</li> <li>TAIS bit is set to "0" (= stop counting)</li> </ul>
Interrupt request generation timing	When the counter reaches "0000 <sub>16</sub> "
TAI <sub>IN</sub> pin function	I/O port or trigger input
TAI <sub>OUT</sub> pin function	I/O port or pulse output
Read from timer	An indeterminate value is read by reading TAI register
Write to timer	<ul style="list-style-type: none"> <li>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</li> <li>When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Pulse output function The timer outputs a low when not counting and a high when counting.</li> </ul>



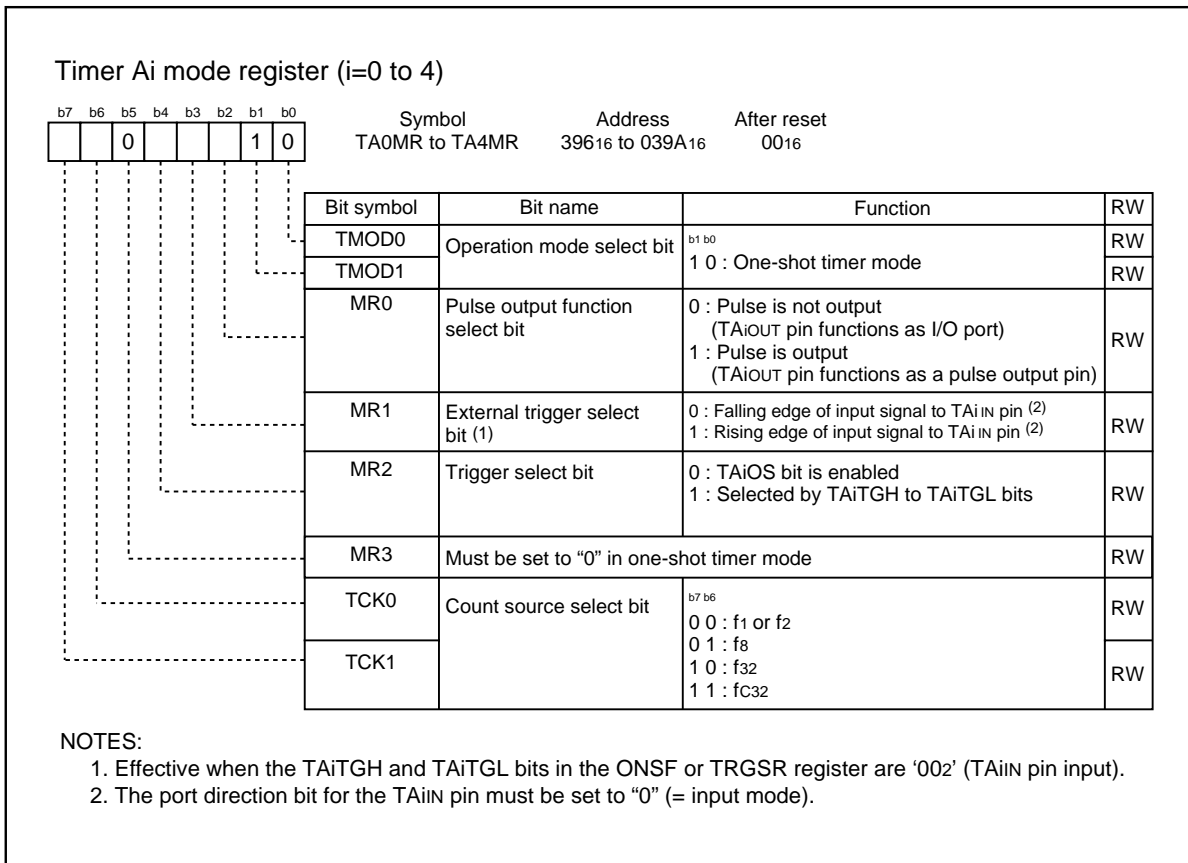


Figure 12.1.3.1. TAIMR Register in One-shot Timer Mode

### 12.1.4. Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see Table 12.1.4.1). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 12.1.4.1 shows TAI<sub>MR</sub> register in pulse width modulation mode. Figures 12.1.4.2 and 12.1.4.3 show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates.

**Table 12.1.4.1. Specifications in Pulse Width Modulation Mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>2</sub> , f <sub>8</sub> , f <sub>32</sub> , f <sub>C32</sub>
Count operation	<ul style="list-style-type: none"> <li>Down-count (operating as an 8-bit or a 16-bit pulse width modulator)</li> <li>The timer reloads a new value at a rising edge of PWM pulse and continues counting</li> <li>The timer is not affected by a trigger that occurs during counting</li> </ul>
16-bit PWM	<ul style="list-style-type: none"> <li>High level width <math>n / f_j</math> <math>n</math>: set value of TAI register (i=0 to 4)</li> <li>Cycle time <math>(2^{16}-1) / f_j</math> fixed <math>f_j</math>: count source frequency (f<sub>1</sub>, f<sub>2</sub>, f<sub>8</sub>, f<sub>32</sub>, f<sub>C32</sub>)</li> </ul>
8-bit PWM	<ul style="list-style-type: none"> <li>High level width <math>n \times (m+1) / f_j</math> <math>n</math>: set value of TAI register high-order address</li> <li>Cycle time <math>(2^8-1) \times (m+1) / f_j</math> <math>m</math>: set value of TAI register low-order address</li> </ul>
Count start condition	<ul style="list-style-type: none"> <li>TAIS bit in the TABSR register is set to "1" (= start counting)</li> <li>The TAIS bit is set to "1" and external trigger input from the TAI<sub>IN</sub> pin</li> <li>The TAIS bit is set to "1" and one of the following external triggers occurs Timer B2 overflow or underflow, timer A<sub>j</sub> (j=i-1, except j=4 if i=0) overflow or underflow, timer A<sub>k</sub> (k=i+1, except k=0 if i=4) overflow or underflow</li> </ul>
Count stop condition	TAIS bit is set to "0" (= stop counting)
Interrupt request generation timing	PWM pulse goes "L"
TAI <sub>IN</sub> pin function	I/O port or trigger input
TAI <sub>OUT</sub> pin function	Pulse output
Read from timer	An indeterminate value is read by reading TAI register
Write to timer	<ul style="list-style-type: none"> <li>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</li> <li>When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>

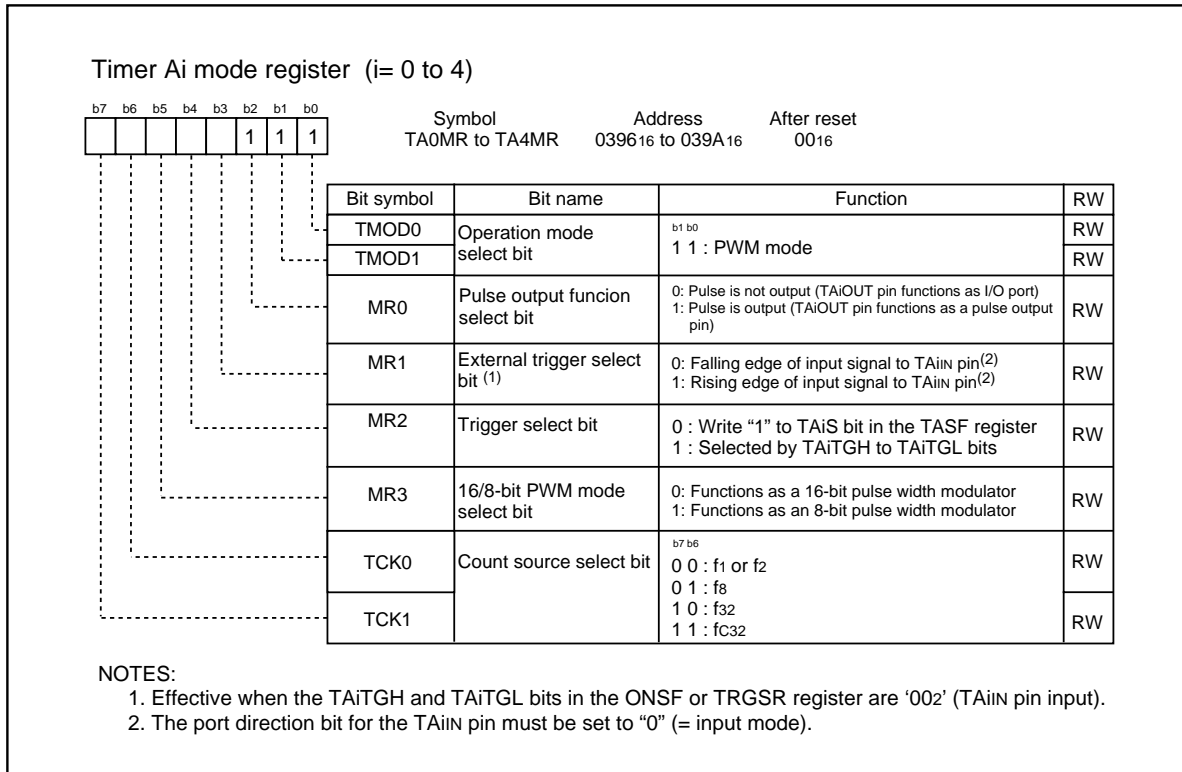


Figure 12.1.4.1. TAIiMR Register in Pulse Width Modulation Mode

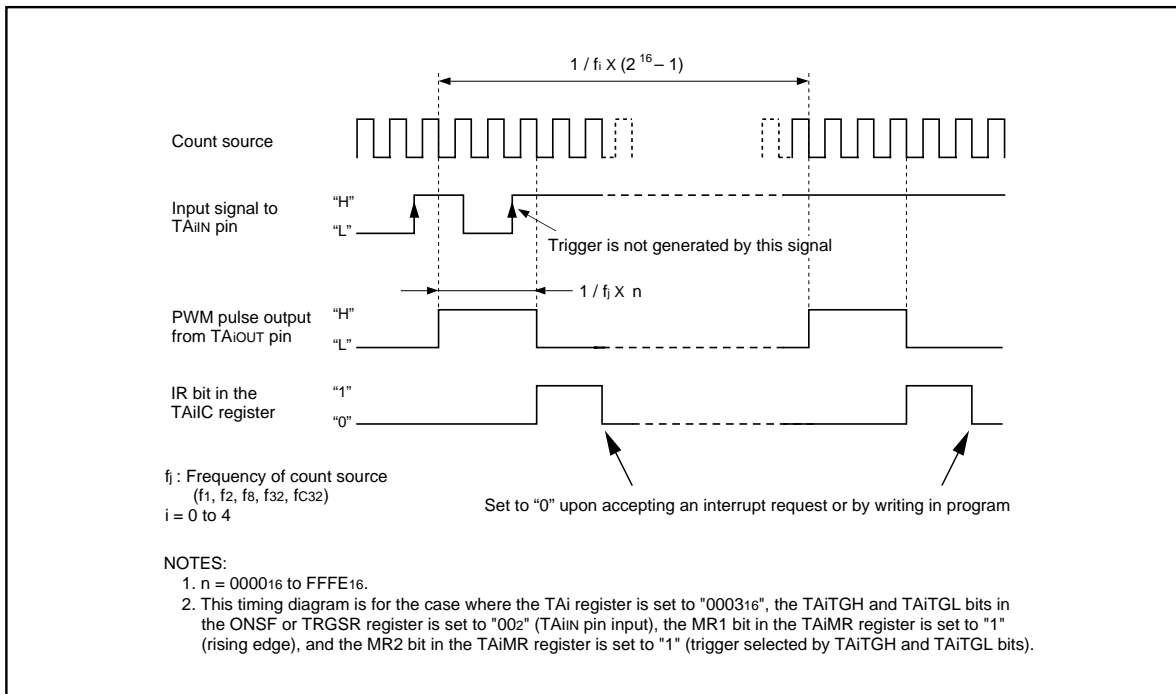


Figure 12.1.4.2. Example of 16-bit Pulse Width Modulator Operation

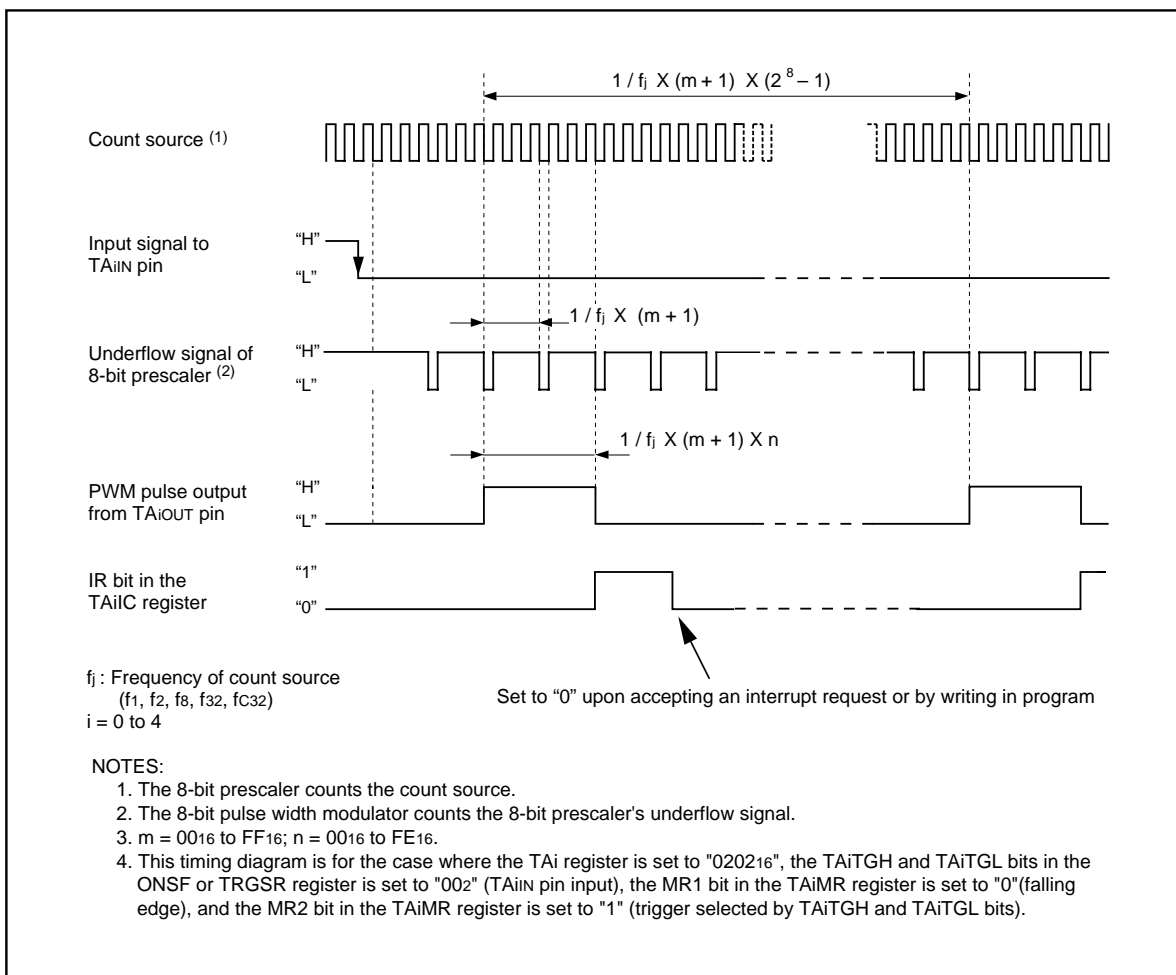


Figure 12.1.4.3. Example of 8-bit Pulse Width Modulator Operation

## 12.2 Timer B

### Note

The TB2IN pin for Timer B2 is not available in 42-pin package.

### [Precautions when using Timer B2]

- Event Counter Mode The external input signals cannot be counted. Set the TCK1 bit in the TB2MR register to "1" when using the Event Count Mode.
- Pulse Period/Pulse Width Measurement Mode This mode cannot be used.

Figure 12.2.1 shows a block diagram of the timer B. Figures 12.2.2 and 12.2.3 show registers related to the timer B.

Timer B supports the following four modes. Use the TMOD1 and TMOD0 bits in the TBiMR register ( $i = 0$  to 2) 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 or underflows of other timers.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.
- A/D trigger mode: The timer starts counting by one trigger until the count value becomes  $0000_{16}$ . This mode is used together with simultaneous sample sweep mode or delayed trigger mode 0 of A/D converter to start A/D conversion.

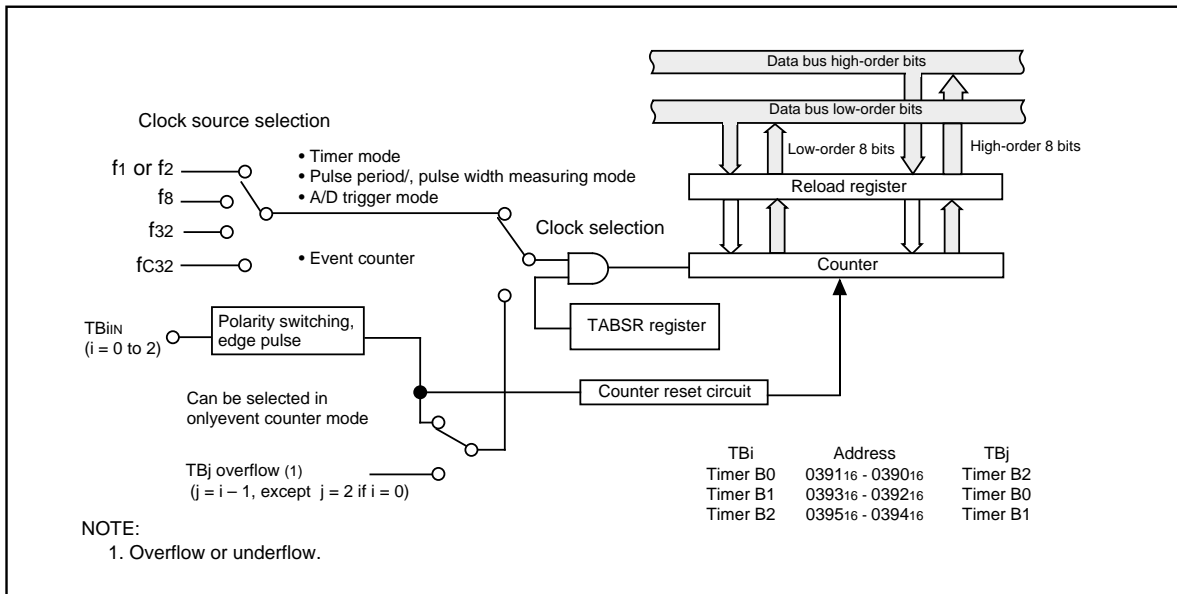


Figure 12.2.1. Timer B Block Diagram

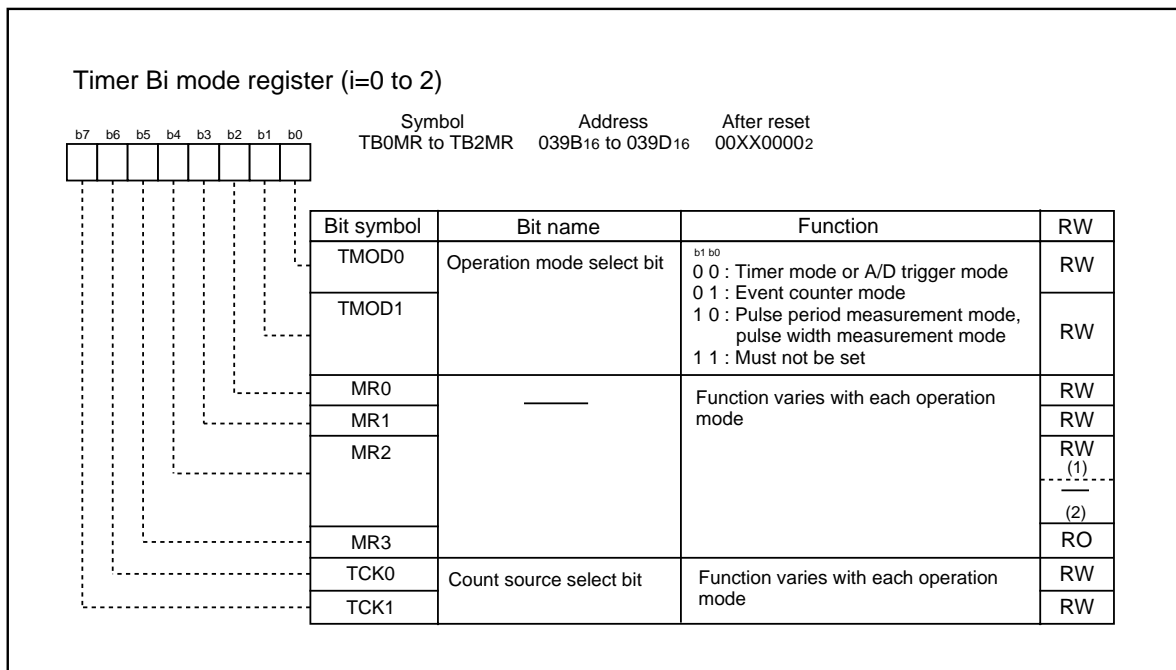


Figure 12.2.2. TB0MR to TB2MR Registers

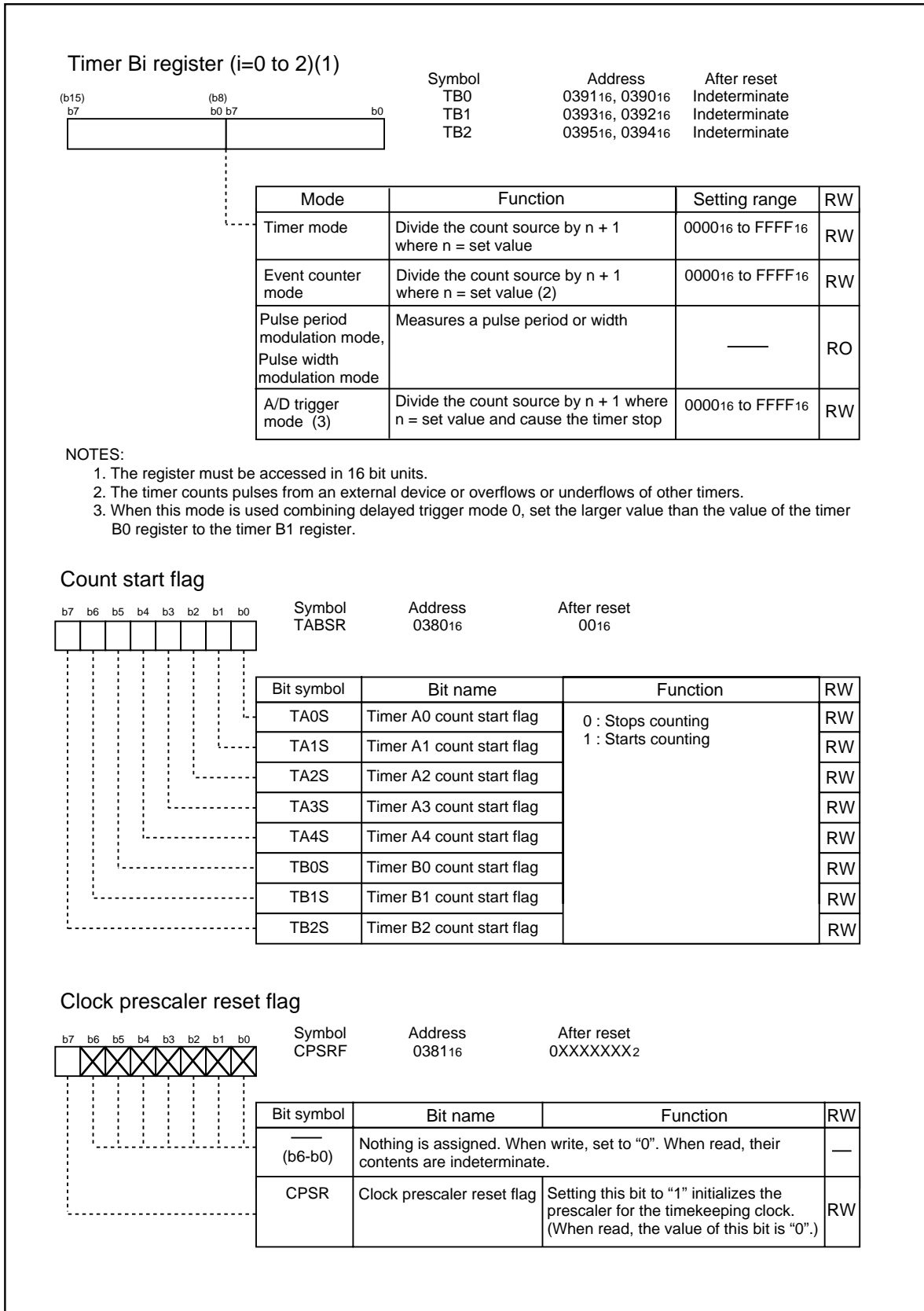


Figure 12.2.3. TB0 to TB2 Registers, TABSR Register, CPSRF Register

### 12.2.1 Timer Mode

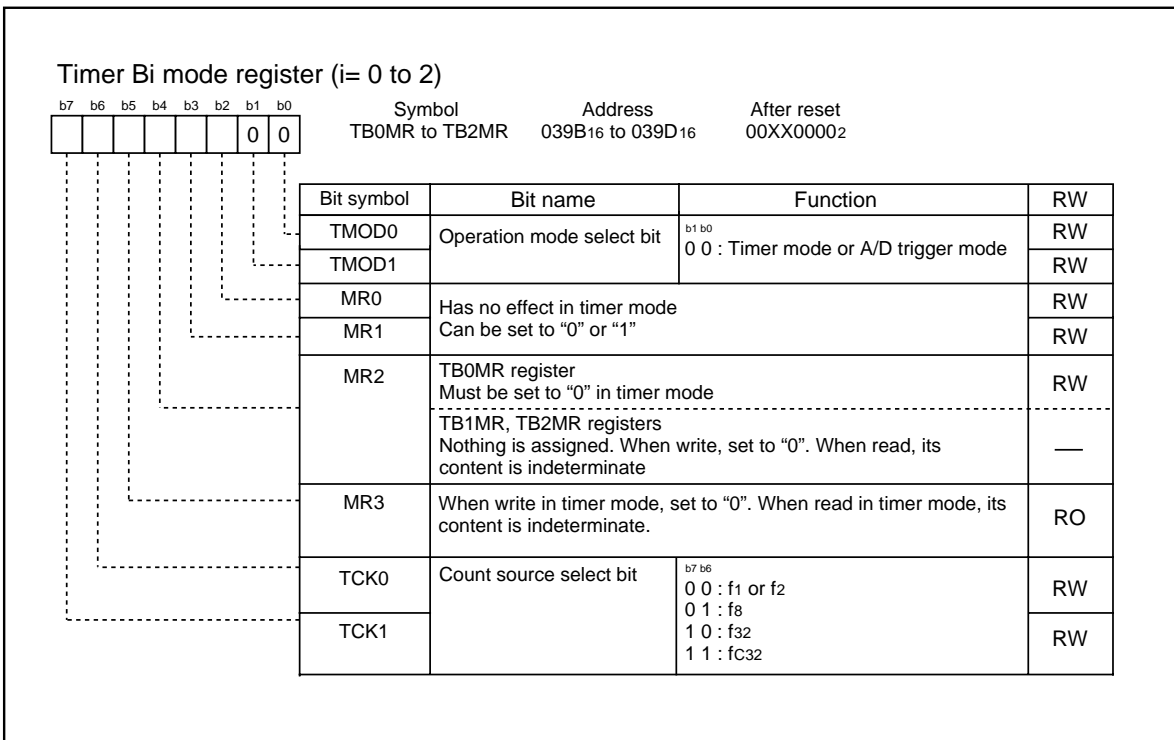
In timer mode, the timer counts a count source generated internally (see Table 12.2.1.1). Figure 12.2.1.1 shows TBiMR register in timer mode.

**Table 12.2.1.1 Specifications in Timer Mode**

Item	Specification
Count source	f1, f2, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>Down-count</li> <li>When the timer underflows, it reloads the reload register contents and continues counting</li> </ul>
Divide ratio	1/(n+1) n: set value of TBi register (i= 0 to 2) 0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	Set TBiS bit <sup>(1)</sup> to "1" (= start counting)
Count stop condition	Set TBiS bit to "0" (= stop counting)
Interrupt request generation timing	Timer underflow
TBiIN pin function	I/O port
Read from timer	Count value can be read by reading TBi register
Write to timer	<ul style="list-style-type: none"> <li>When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter</li> <li>When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>

NOTE:

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



**Figure 12.2.1.1 TBiMR Register in Timer Mode**



## 12.2.2 Event Counter Mode

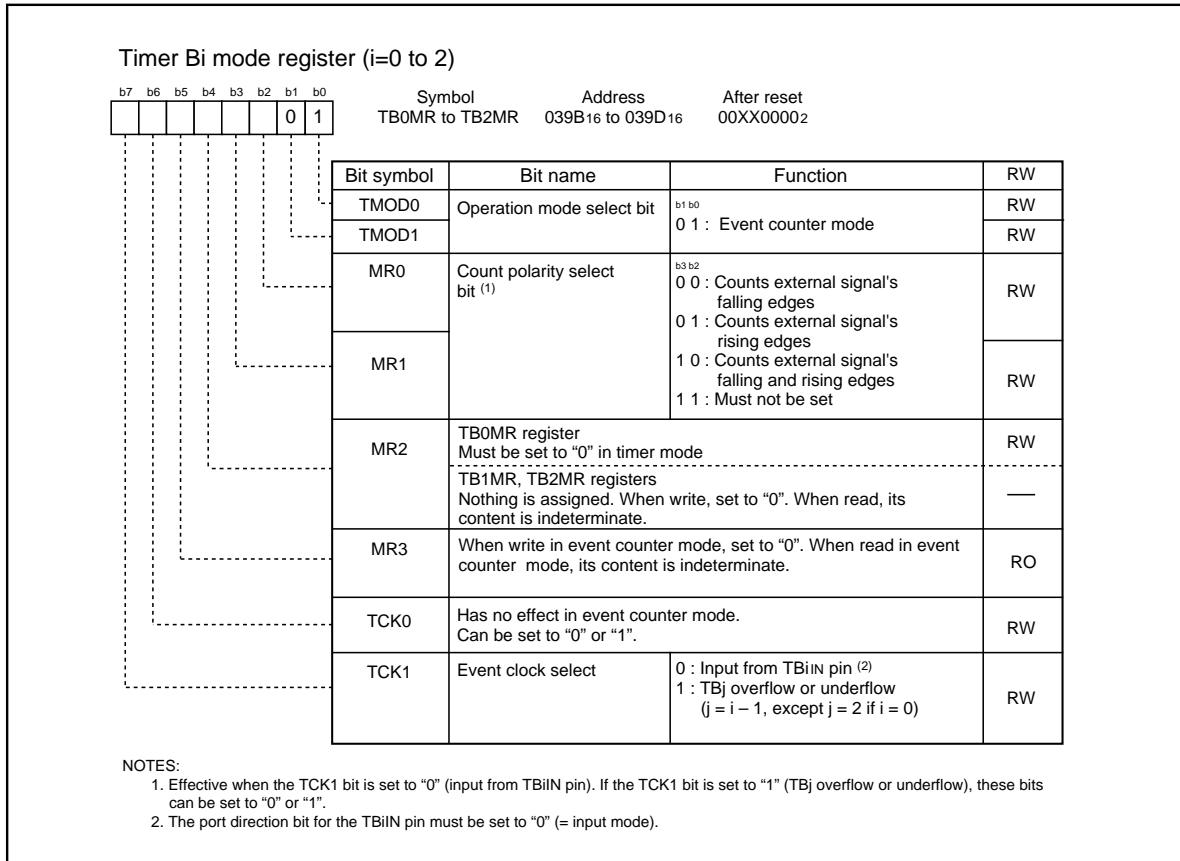
In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers (see Table 12.2.2.1). Figure 12.2.2.1 shows TBiMR register in event counter mode.

**Table 12.2.2.1 Specifications in Event Counter Mode**

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

NOTE:

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



**Figure 12.2.2.1 TBiMR Register in Event Counter Mode**

### 12.2.3 Pulse Period and Pulse Width Measurement Mode

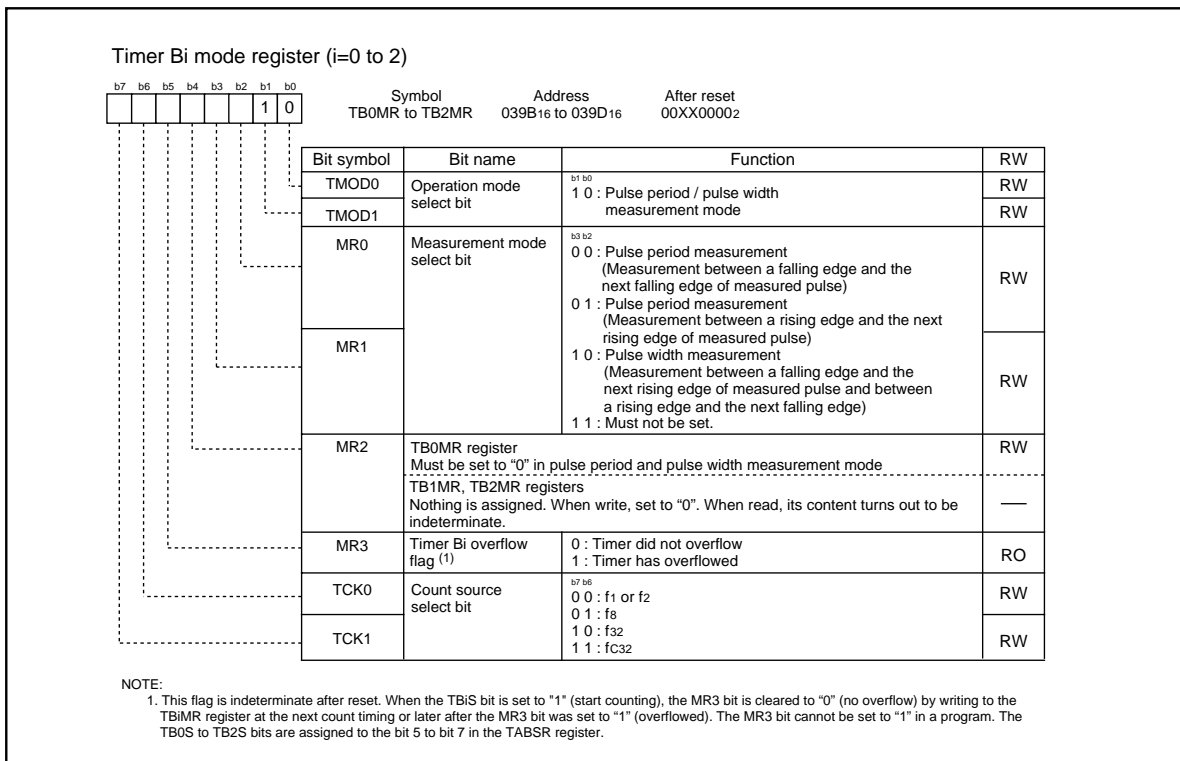
In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal (see Table 12.2.3.1). Figure 12.2.3.1 shows TBiMR register in pulse period and pulse width measurement mode. Figure 12.2.3.2 shows the operation timing when measuring a pulse period. Figure 12.2.3.3 shows the operation timing when measuring a pulse width.

**Table 12.2.3.1 Specifications in Pulse Period and Pulse Width Measurement Mode**

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

**NOTES:**

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



**Figure 12.2.3.1 TBiMR Register in Pulse Period and Pulse Width Measurement Mode**

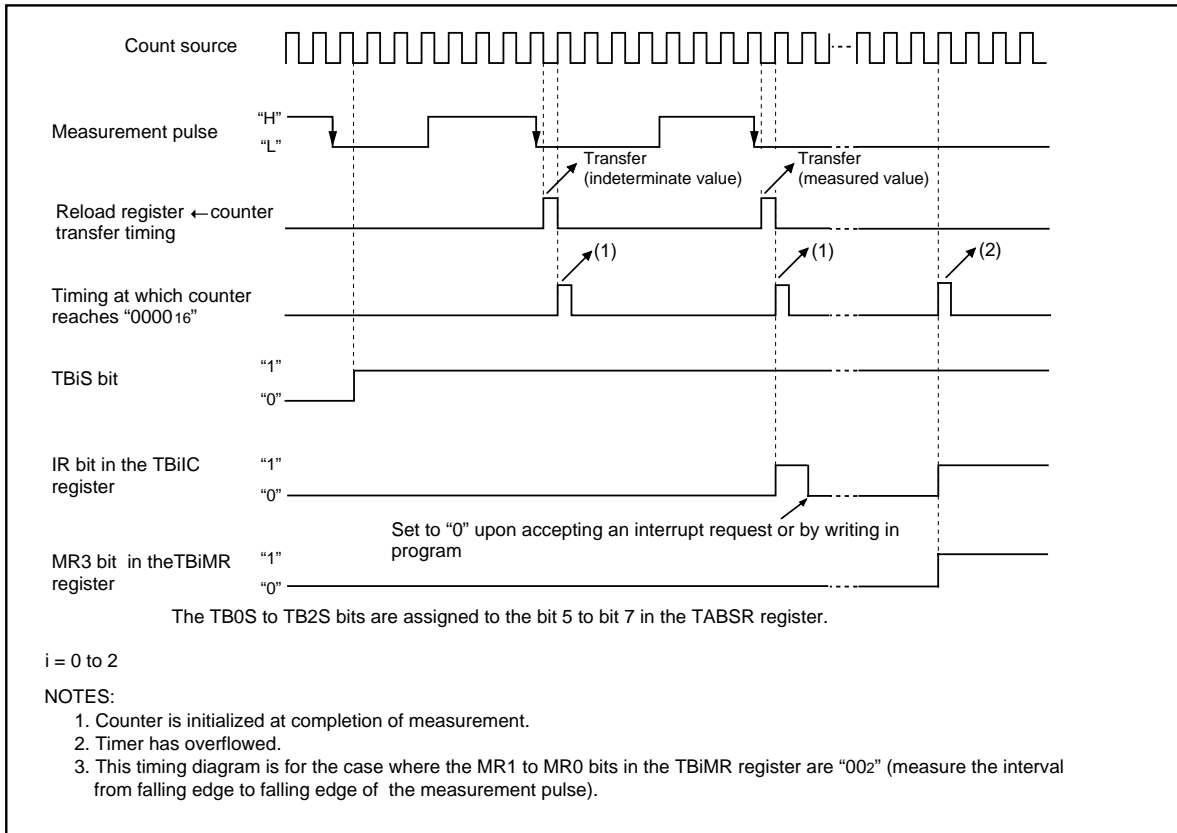


Figure 12.2.3.2 Operation timing when measuring a pulse period

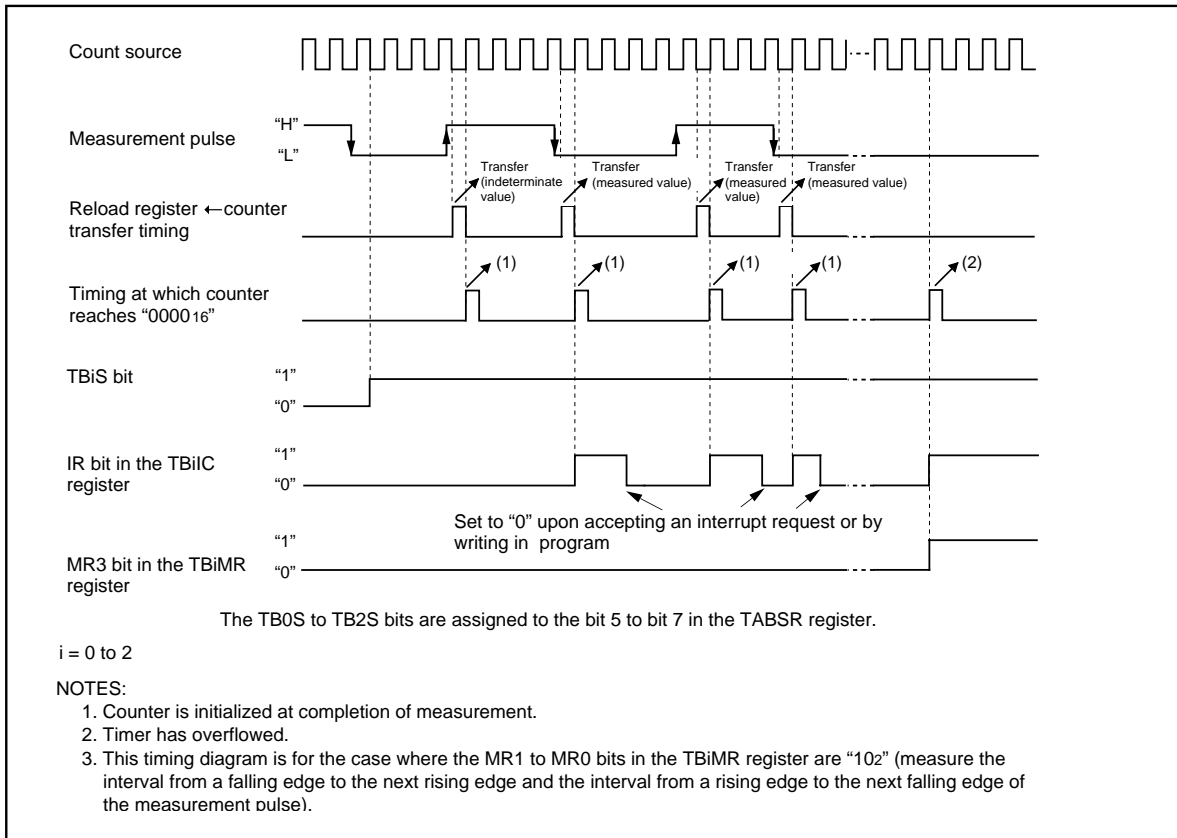


Figure 12.2.3.3 Operation timing when measuring a pulse width

### 12.2.4 A/D Trigger Mode

A/D trigger mode is used as conversion start trigger for A/D converter in simultaneous sample sweep mode of A/D conversion or delayed trigger mode 0. This mode is used as conversion start trigger of A/D converter. A/D trigger mode is used in Timer B0 and Timer B1. In this mode, the timer starts counting by one trigger until the count value becomes 0000<sub>16</sub>. Figure 12.2.4.1 shows the TBiMR register in A/D trigger mode and figure 12.2.4.2 shows the TB2SC register.

**Table 12.2.4.1 A/D Trigger Mode Specifications**

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

**NOTES:**

1. A/D conversion is started by the timer underflow. For details refer to **Section 14. A/D Converter**.
2. When using in delayed trigger mode 0, set the larger value than the value of the timer B0 register to the timer B1 register.

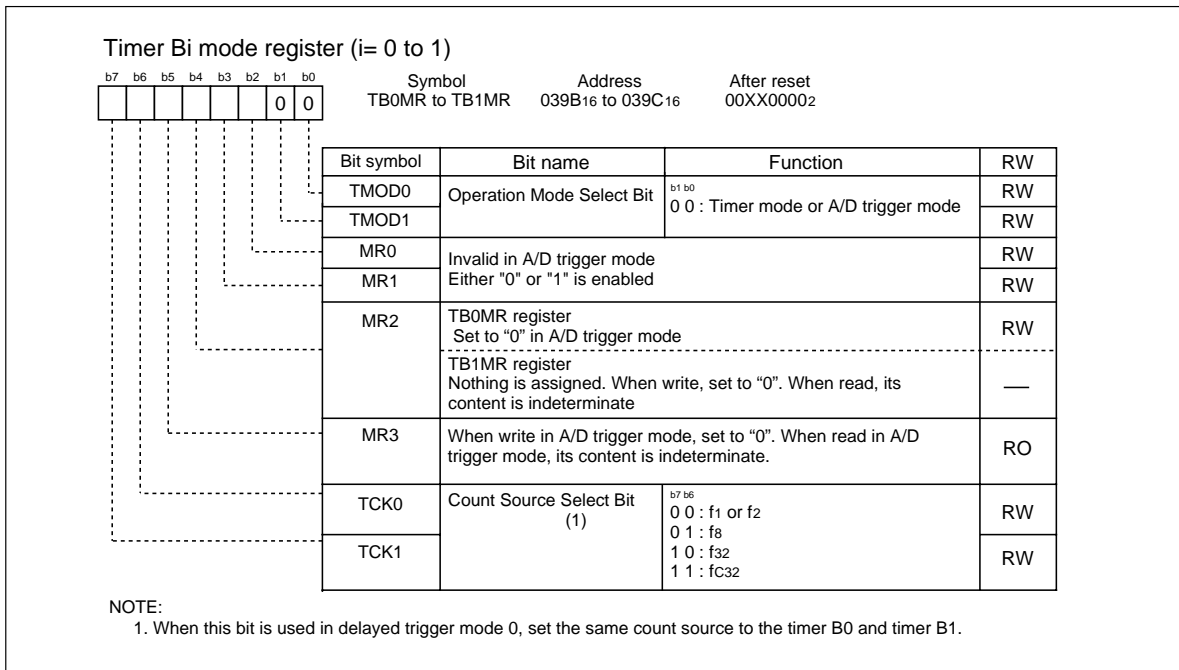


Figure 12.2.4.1 TBiMR Register in A/D Trigger Mode

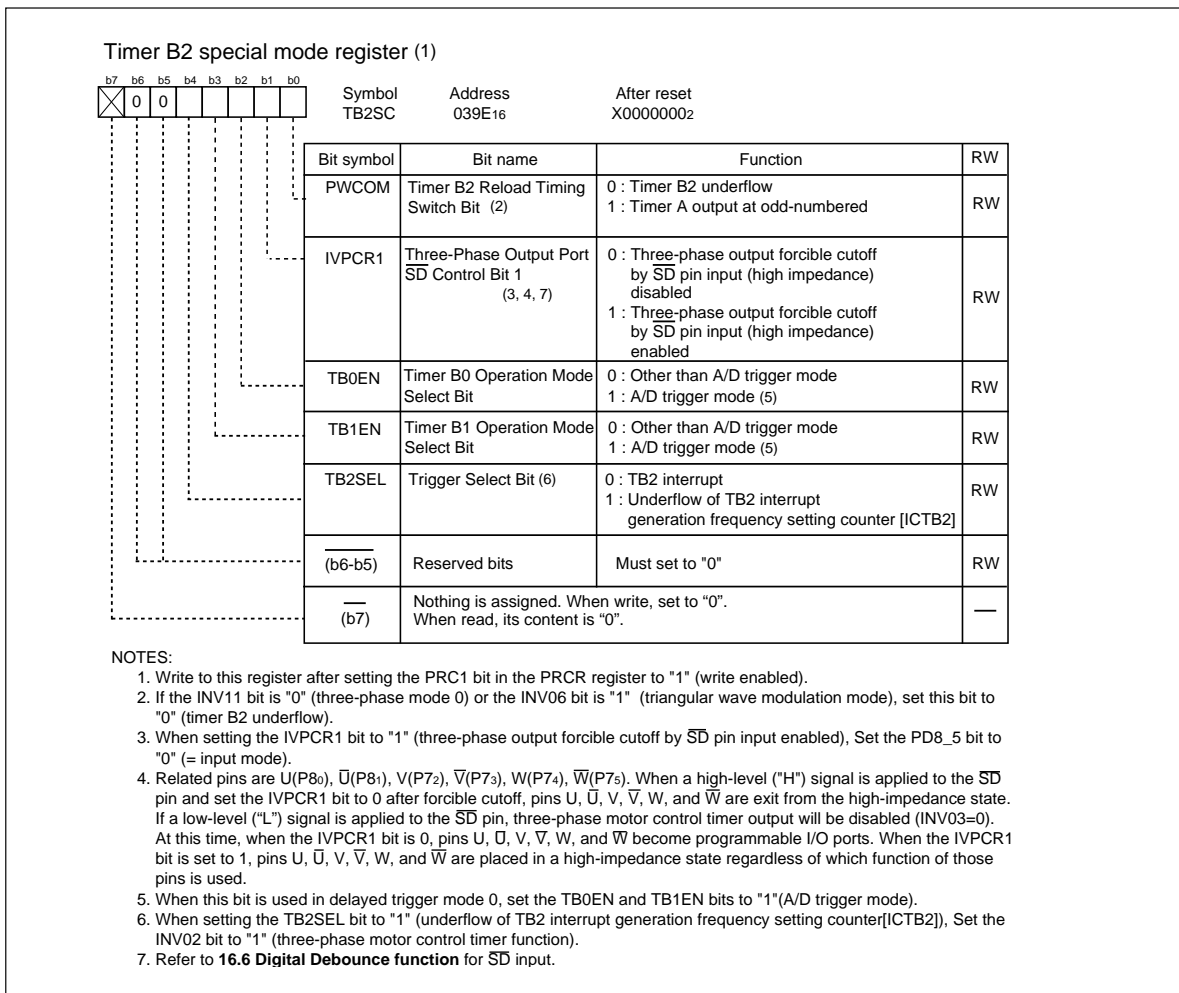


Figure 12.2.4.2 TB2SC Register

### 12.3 Three-phase Motor Control Timer Function

Timers A1, A2, A4 and B2 can be used to output three-phase motor drive waveforms. Table 12.3.1 lists the specifications of the three-phase motor control timer function. Figure 12.3.1 shows the block diagram for three-phase motor control timer function. Also, the related registers are shown on Figure 12.3.2 to Figure 12.3.8.

**Table 12.3.1. 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 <sup>(1)</sup>	Input "L" to $\bar{SD}$ pin
Used Timers	Timer A4, A1, A2 (used in the one-shot timer mode) Timer A4: U- and $\bar{U}$ -phase waveform control Timer A1: V- and $\bar{V}$ -phase waveform control Timer A2: W- and $\bar{W}$ -phase waveform control Timer B2 (used in the timer mode) Carrier wave cycle control Dead timer timer (3 eight-bit timer and shared reload register) Dead time control
Output waveform	Triangular wave modulation, Sawtooth wave modification Enable to output "H" or "L" for one cycle Enable to set positive-phase level and negative-phase level respectively
Carrier wave cycle	Triangular wave modulation: count source x (m+1) x 2 Sawtooth wave modulation: count source x (m+1) m: Setting value of TB2 register, 0 to 65535 Count source: f1, f2, f8, f32, fc32
Three-phase PWM output width	Triangular wave modulation: count source x n x 2 Sawtooth wave modulation: count source x n n: Setting value of TA4, TA1 and TA2 register (of TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11 bit to "1"), 1 to 65535 Count source: f1, f2, f8, f32, fc32
Dead time active disable function	Count source x p, or no dead time p: Setting value of DTT register, 1 to 255 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	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

**NOTES:**

- When the INV02 bit in the INVC0 register is set to "1" (three-phase motor control timer function), the  $\bar{SD}$  function of the P85/ $\bar{SD}$  pin is enabled. At this time, the P85 pin cannot be used as a programmable I/O port. When the  $\bar{SD}$  function is not used, apply "H" to the P85/ $\bar{SD}$  pin.
- When the IVPCR1 bit in the TB2SC register is set to "1" (enable three-phase output forced cutoff by  $\bar{SD}$  pin input), and "L" is applied to the  $\bar{SD}$  pin, the related pins enter high-impedance state regardless of the functions which are used. When the IVPCR1 bit is set to "0" (disabled three-phase output forced cutoff by  $\bar{SD}$  pin input) and "L" is applied to the  $\bar{SD}$  pin, the related pins can be selected as a programmable I/O port and the setting of the port and port direction registers are enable.

Related pins      P72/CLK2/TA1OUT/V/RxD1  
                       P73/CTS2/RTS2/TA1IN/V/TxD1  
                       P74/TA2OUT/W  
                       P75/TA2IN/ $\bar{W}$   
                       P80/TA4OUT/U  
                       P81/TA4IN/ $\bar{U}$

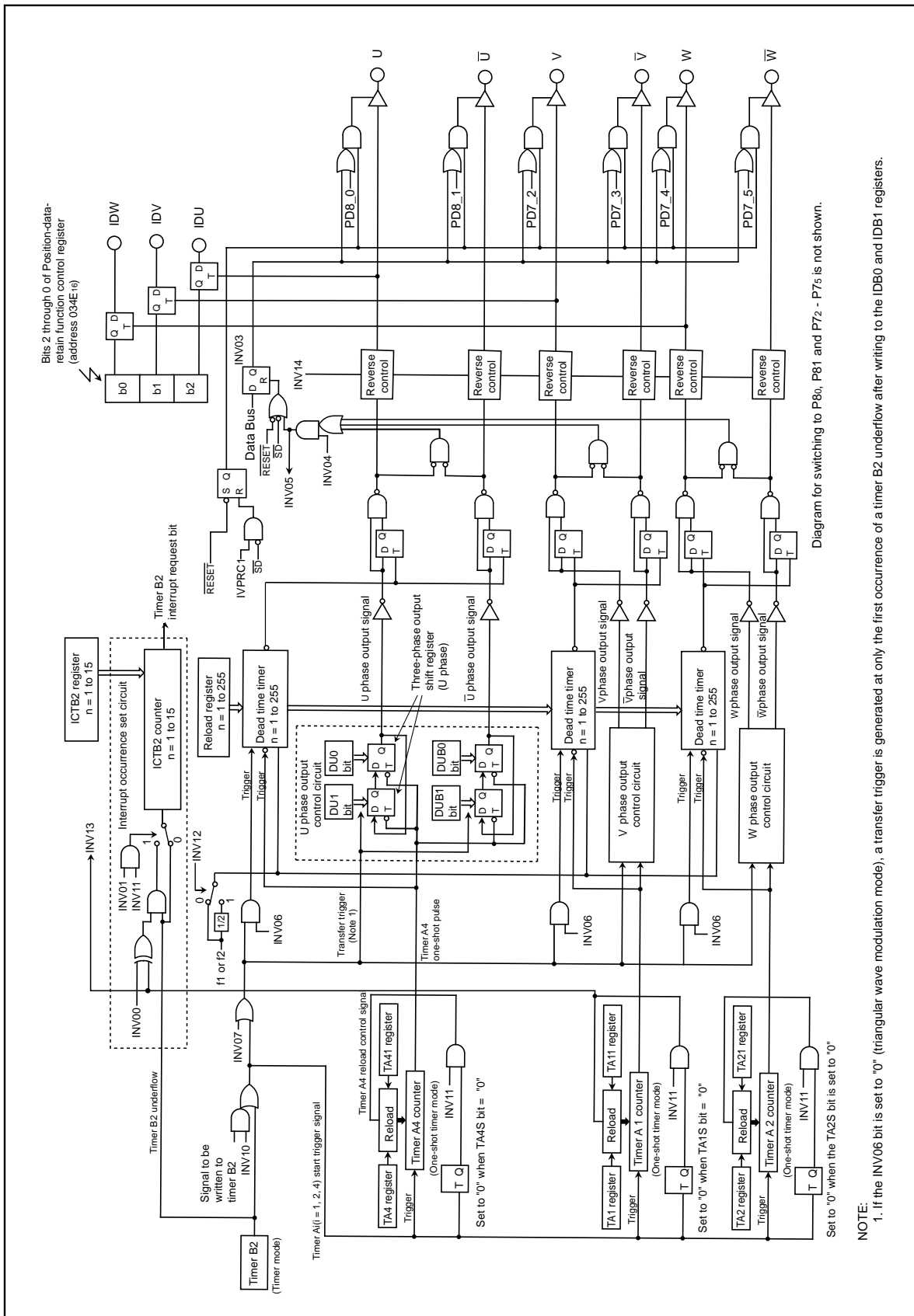


Figure 12.3.1. Three-phase Motor Control Timer Functions Block Diagram

Diagram for switching to P80, P81 and P72 - P75 is not shown.

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

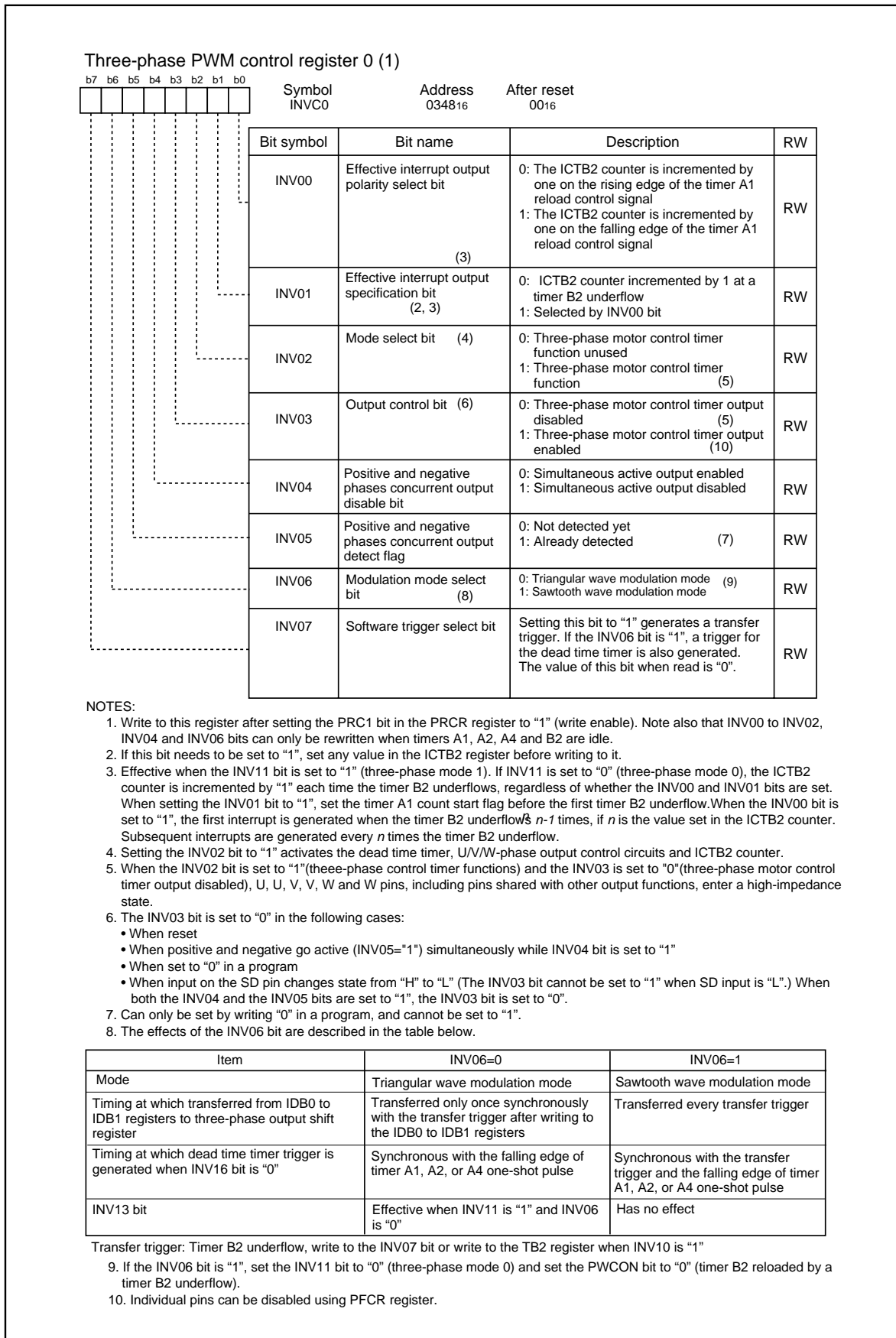


Figure 12.3.2. INVC0 Register



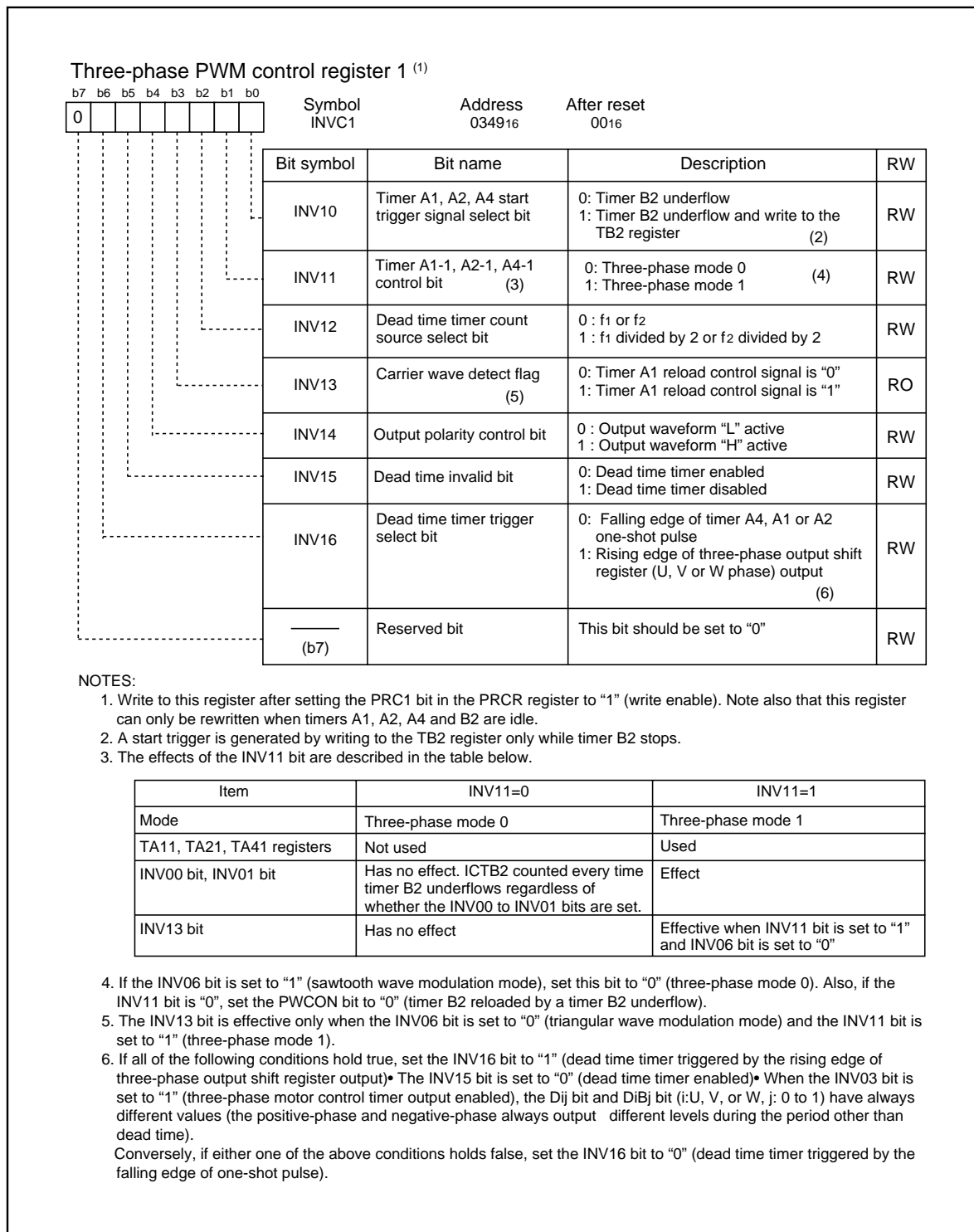


Figure 12.3.3. INVC1 Register

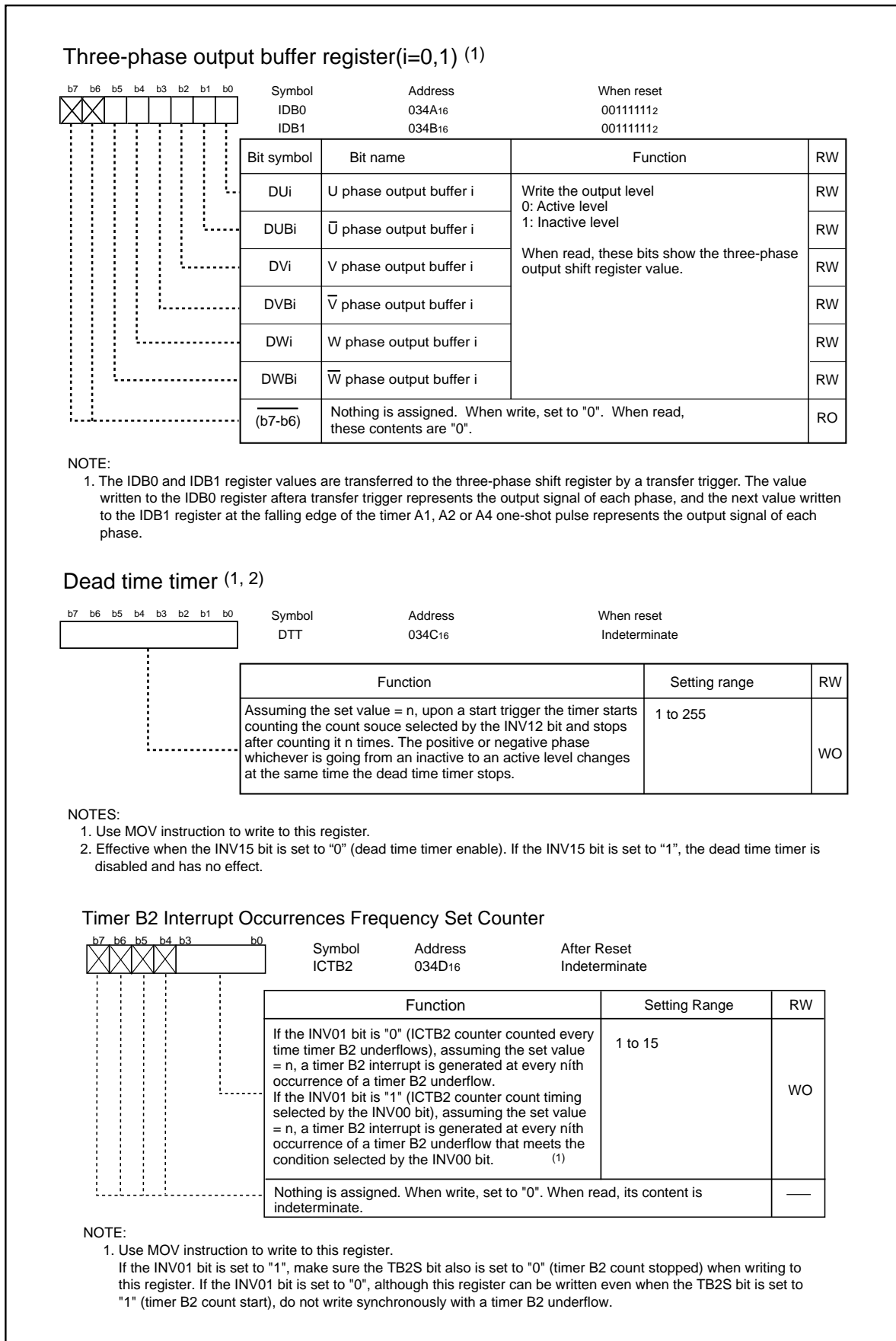


Figure 12.3.4. IDB0 Register, IDB1 Register, DTT Register, and ICTB2 Register

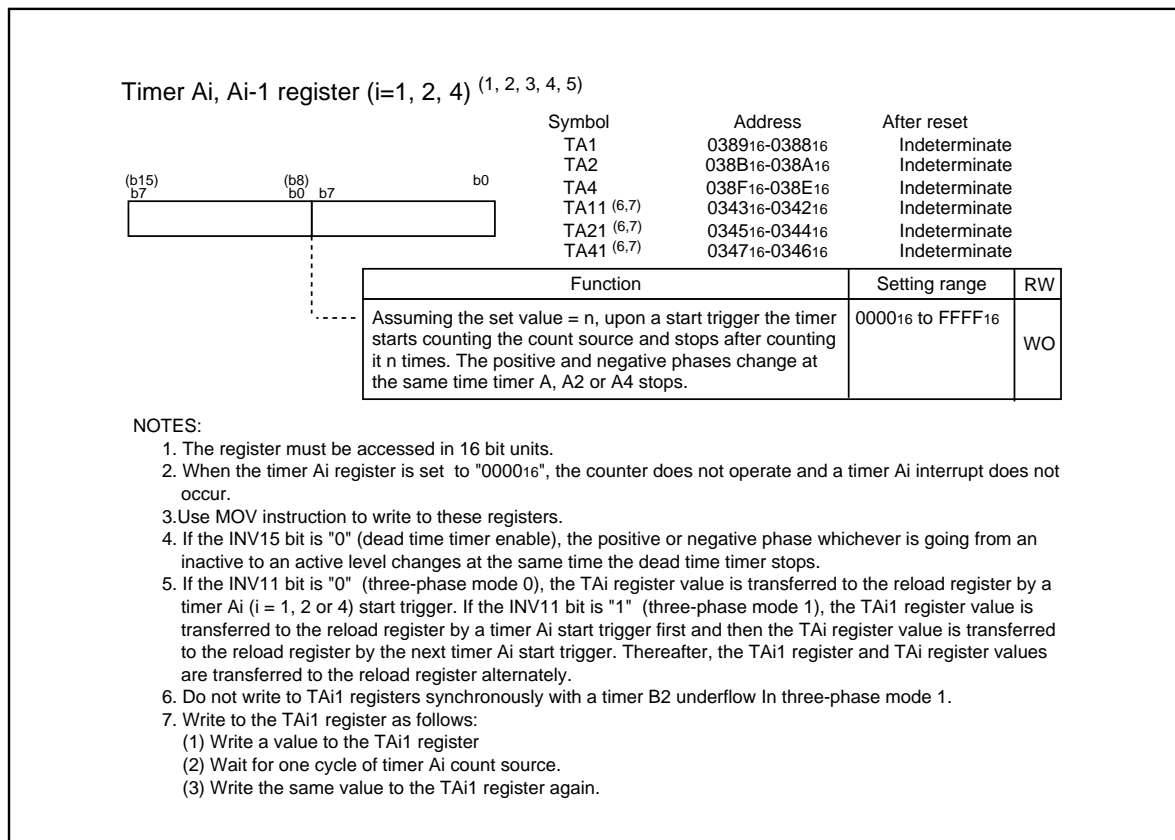


Figure 12.3.5. TA1, TA2, TA4, TA11, TA21 and TA41 Registers

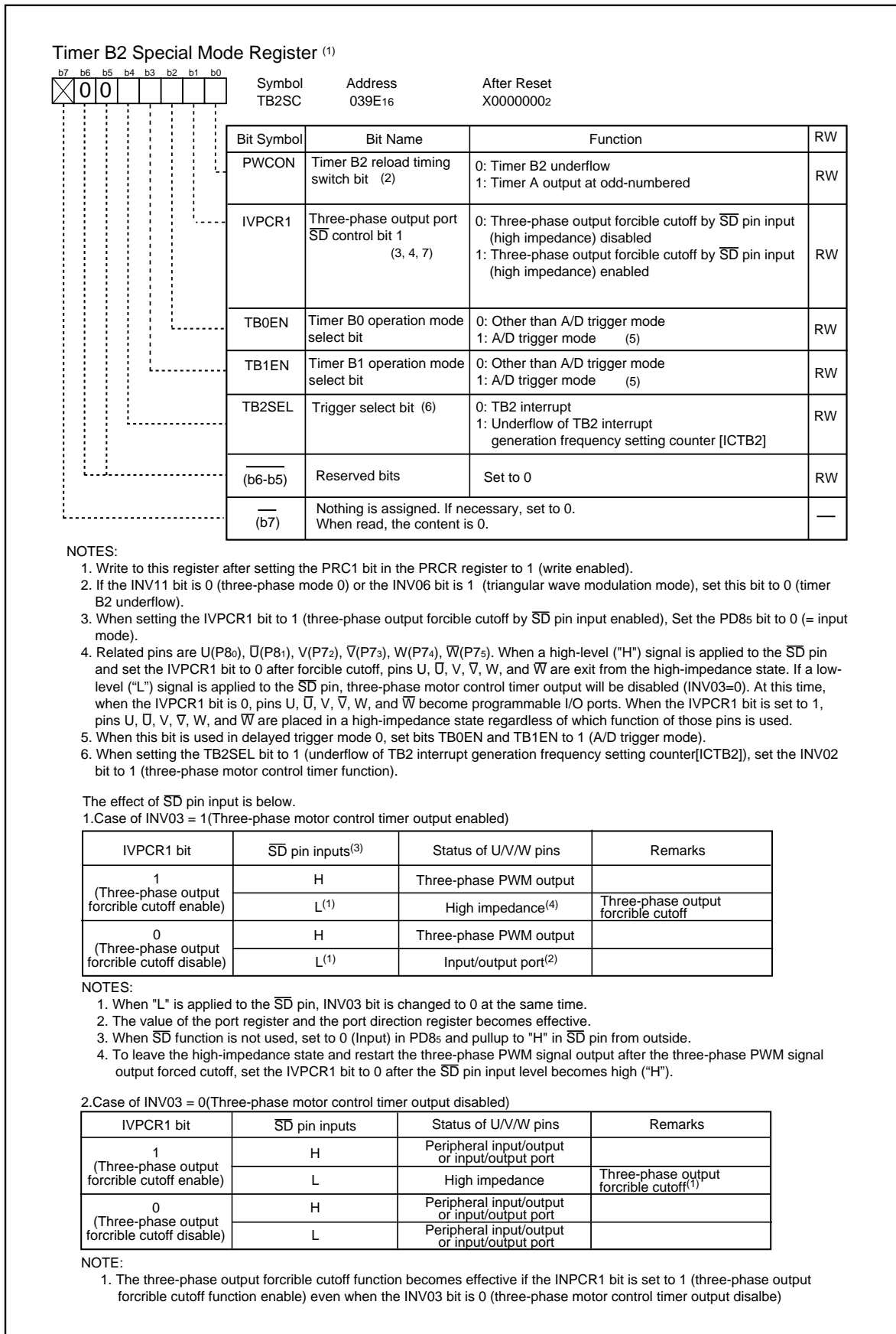


Figure 12.3.6. TB2SC Registers

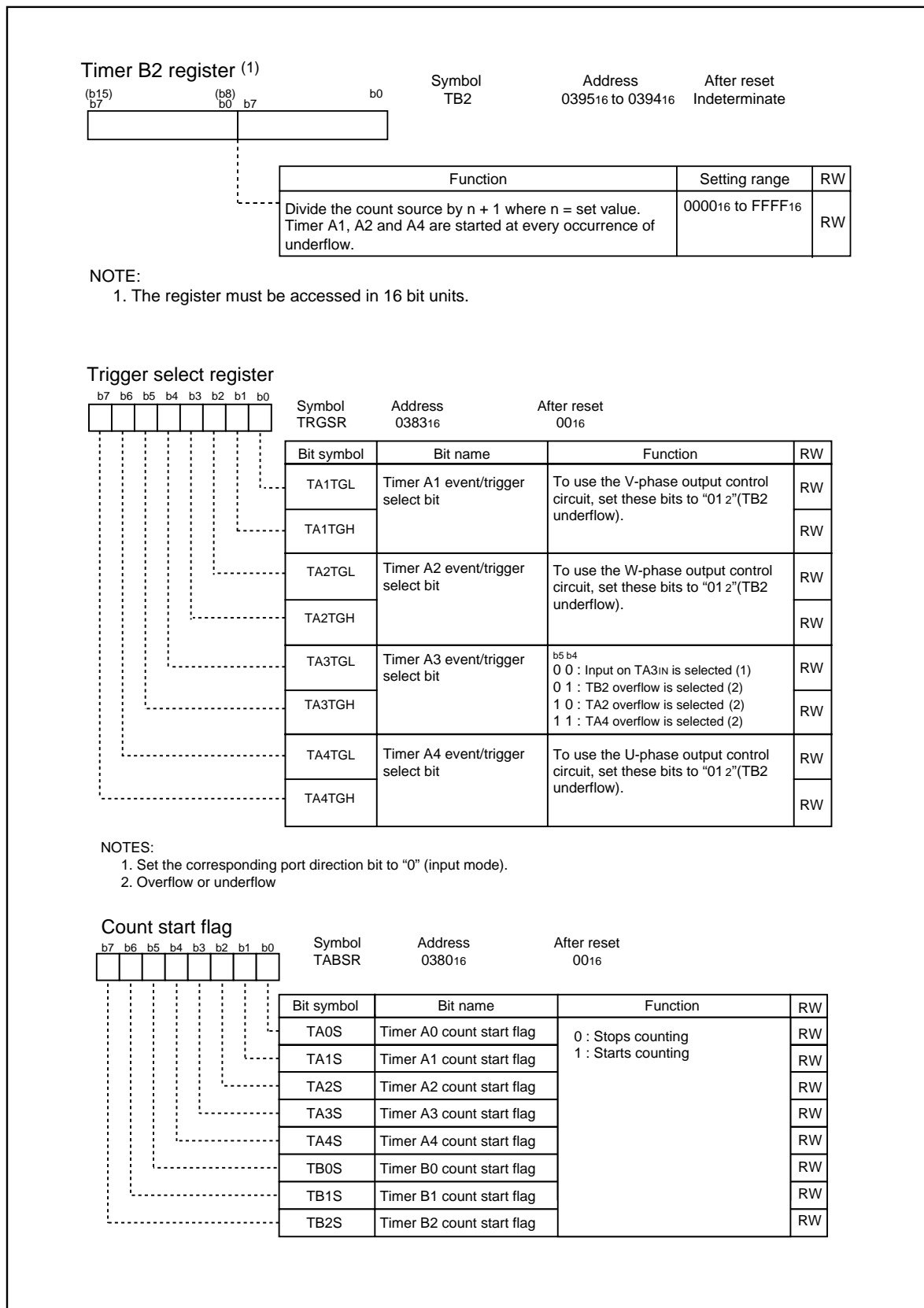


Figure 12.3.7. TB2 Register, TRGSR Register, and TABSR Register

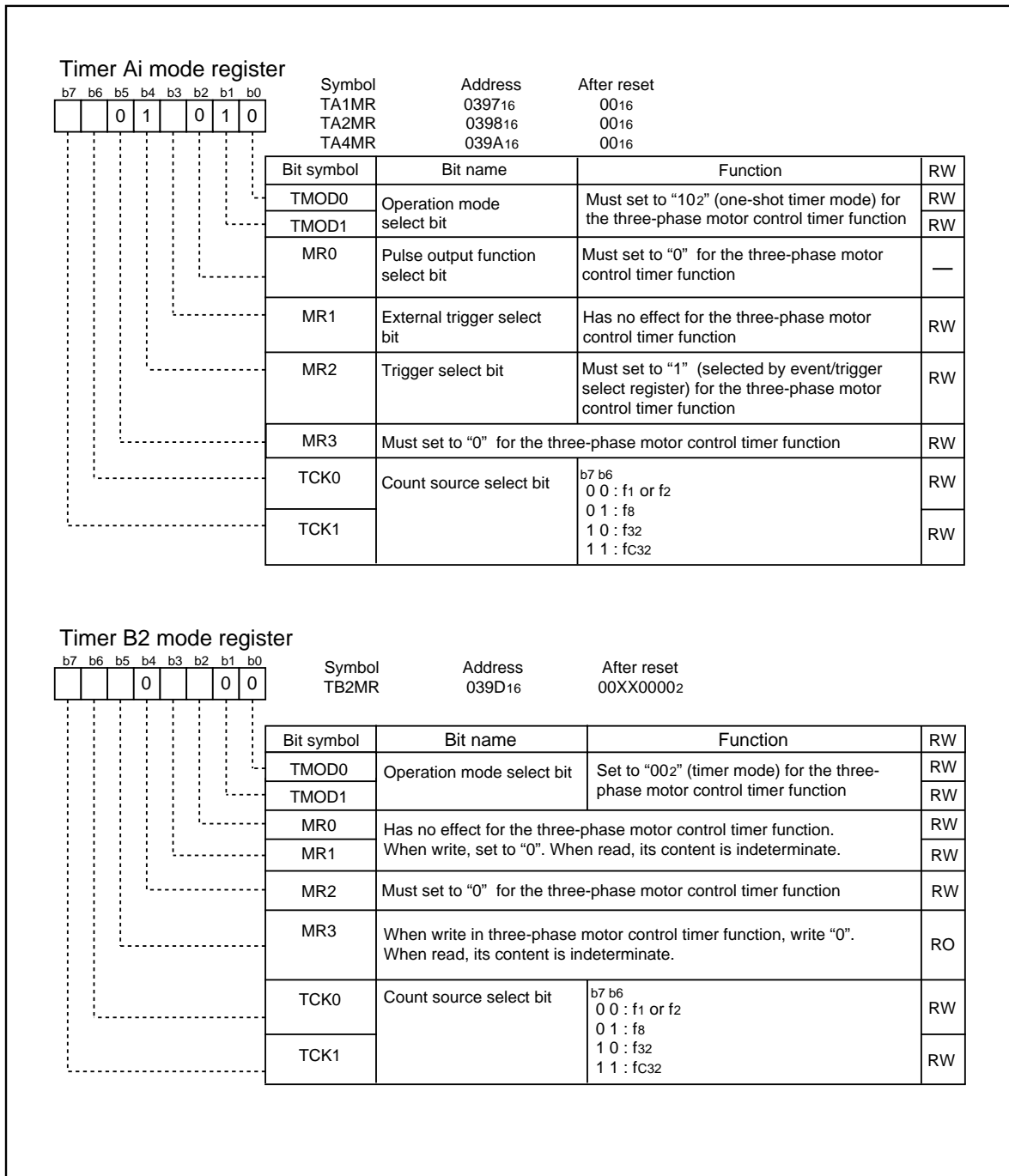
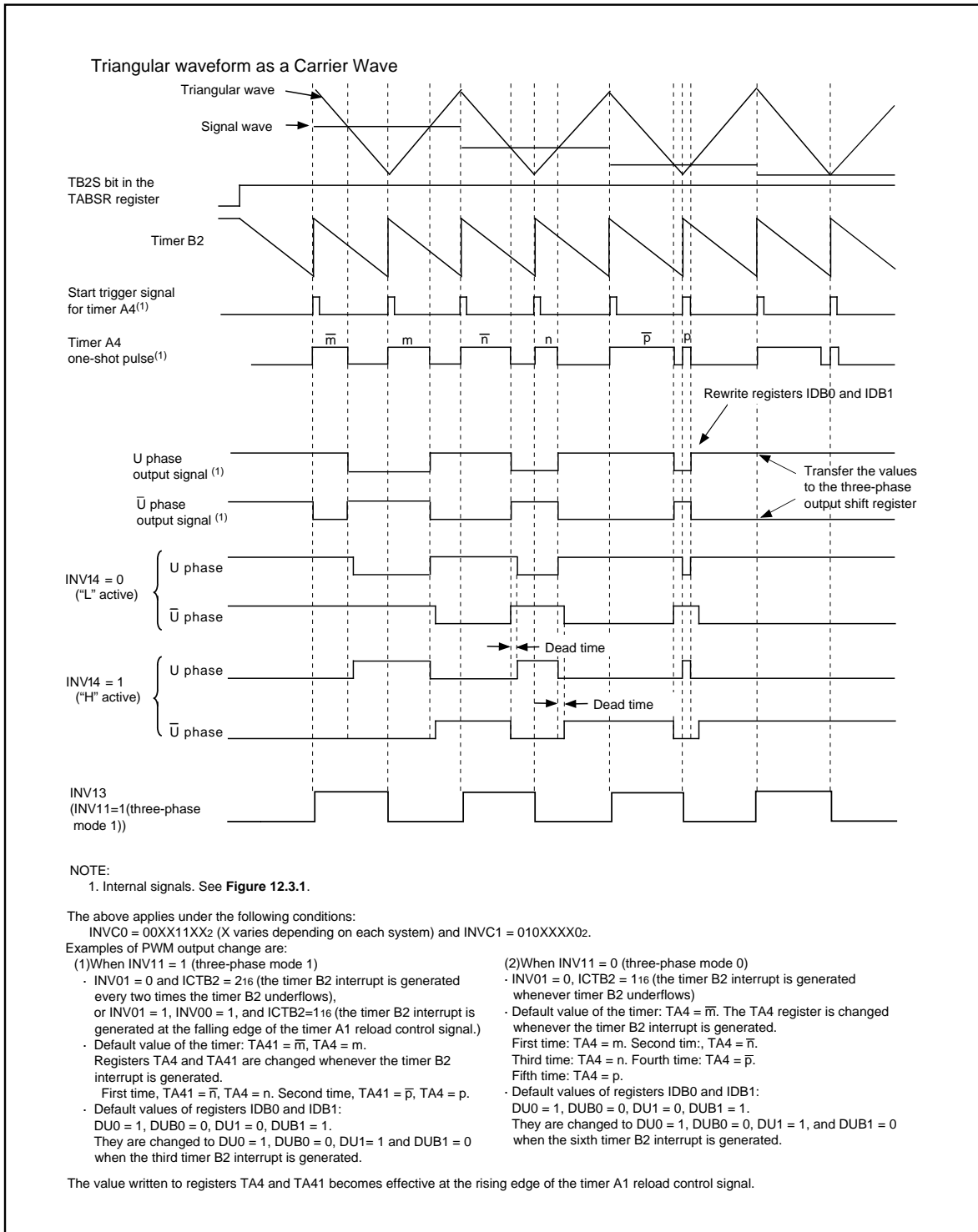


Figure 12.3.8. TA1MR, TA2MR, TA4MR, and TB2MR Registers

The three-phase motor control timer function is enabled by setting the INV02 bit in the VC0 register to "1". When this function is on, timer B2 is used to control the carrier wave, and timers A4, A1 and A2 are used to control three-phase PWM outputs (U,  $\bar{U}$ , V,  $\bar{V}$ , W and  $\bar{W}$ ). The dead time is controlled by a dedicated dead-time timer. Figure 12.3.9 shows the example of triangular modulation waveform, and Figure 12.3.10 shows the example of sawtooth modulation waveform.



**Figure 12.3.9. Triangular Wave Modulation Operation**

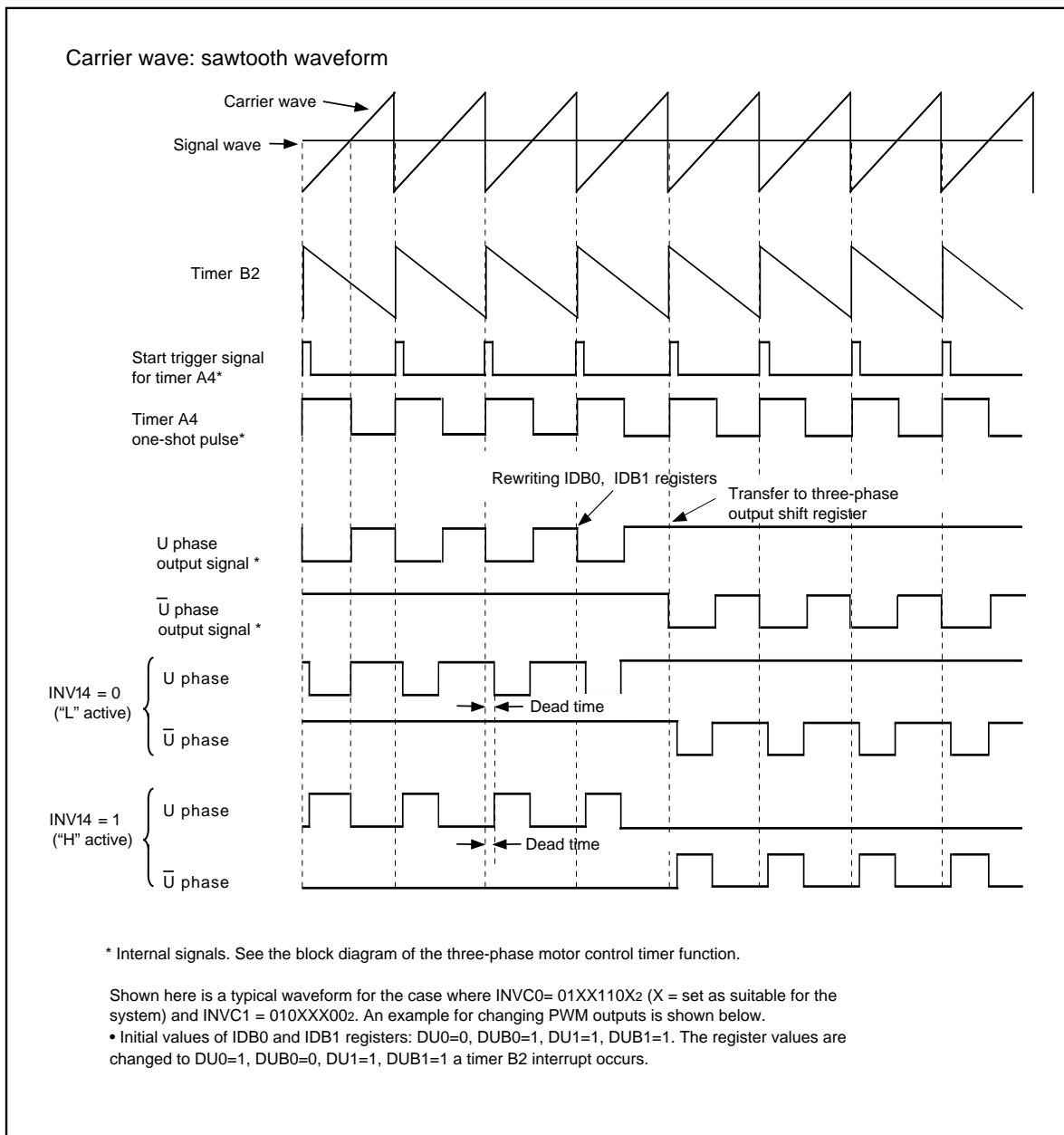


Figure 12.3.10. Sawtooth Wave Modulation Operation



### 12.3.1 Position-data-retain Function

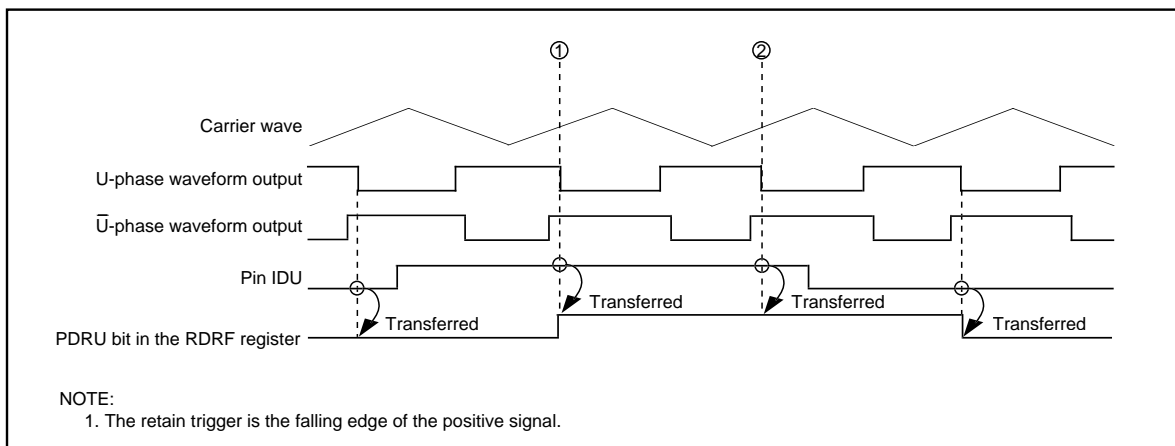
This function is used to retain the position data synchronously with the three-phase waveform output. There are three position-data input pins for U, V, and W phases.

A trigger to retain the position data (hereafter, this trigger is referred to as "retain trigger") can be selected by the retain-trigger polarity select bit (bit 3 of the position-data-retain function control register, at address 034E16). This bit selects the retain trigger to be the falling edge of each positive phase, or the rising edge of each positive phase.

#### 12.3.1.1 Operation of the Position-data-retain Function

Figure 12.3.1.1.1 shows a usage example of the position-data-retain function (U phase) when the retain trigger is selected as the falling edge of the positive signal.

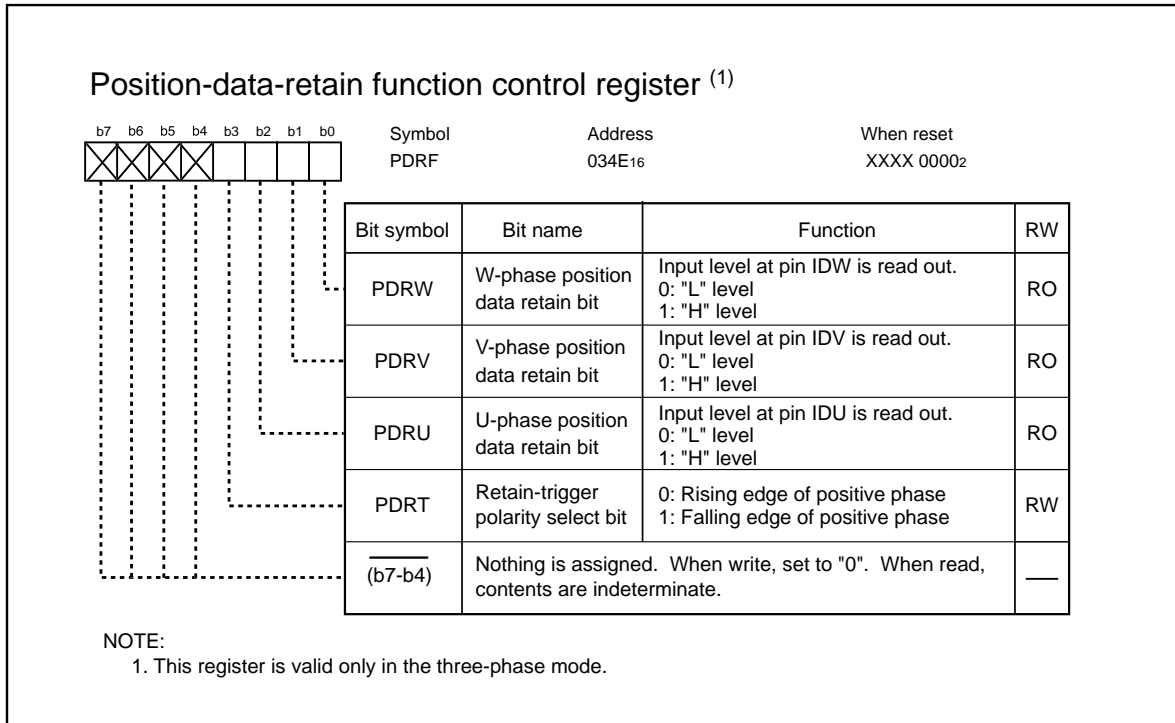
- (1) At the falling edge of the U-phase waveform output, the state at pin IDU is transferred to the U-phase position data retain bit ( bit2 at address 034E16 ).
- (2) Until the next falling edge of the Uphase waveform output, the above value is retained.



**Figure 12.3.1.1.1 Usage Example of Position-data-retain Function ( U phase )**

### 12.3.1.2 Position-data-retain Function Control Register

Figure 12.3.1.2.1 shows the structure of the position-data-retain function control register.



**Figure 12.3.1.2.1. PDRF Register**

#### 12.3.1.2.1 W-phase Position Data Retain Bit (PDRW)

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

#### 12.3.1.2.2 V-phase Position Data Retain Bit (PDRV)

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

#### 12.3.1.2.3 U-phase Position Data Retain Bit (PDRU)

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

#### 12.3.1.2.4 Retain-trigger Polarity Select Bit (PDRT)

This bit is used to select the trigger polarity to retain the position data.

When this bit is set to "0", the rising edge of each positive phase selected.

When this bit is set to "1", the falling edge of each positive phase selected.

### 12.3.2 Three-phase/Port Output Switch Function

When the INVC03 bit in the INVC0 register set to "1" (Timer output enabled for three-phase motor control) and setting the PFCi (i=0 to 5) in the PFCR register to "0" (I/O port), the three-phase PWM output pin (U,  $\bar{U}$ , V,  $\bar{V}$ , W and  $\bar{W}$ ) functions as I/O port. Each bit in the PFCi bits (i=0 to 5) is applicable for each one of three-phase PWM output pins. Figure 12.3.2.1 shows the example of three-phase/port output switch function. Figure 12.3.2.2 shows the PFCR register and the three-phase protect control register.

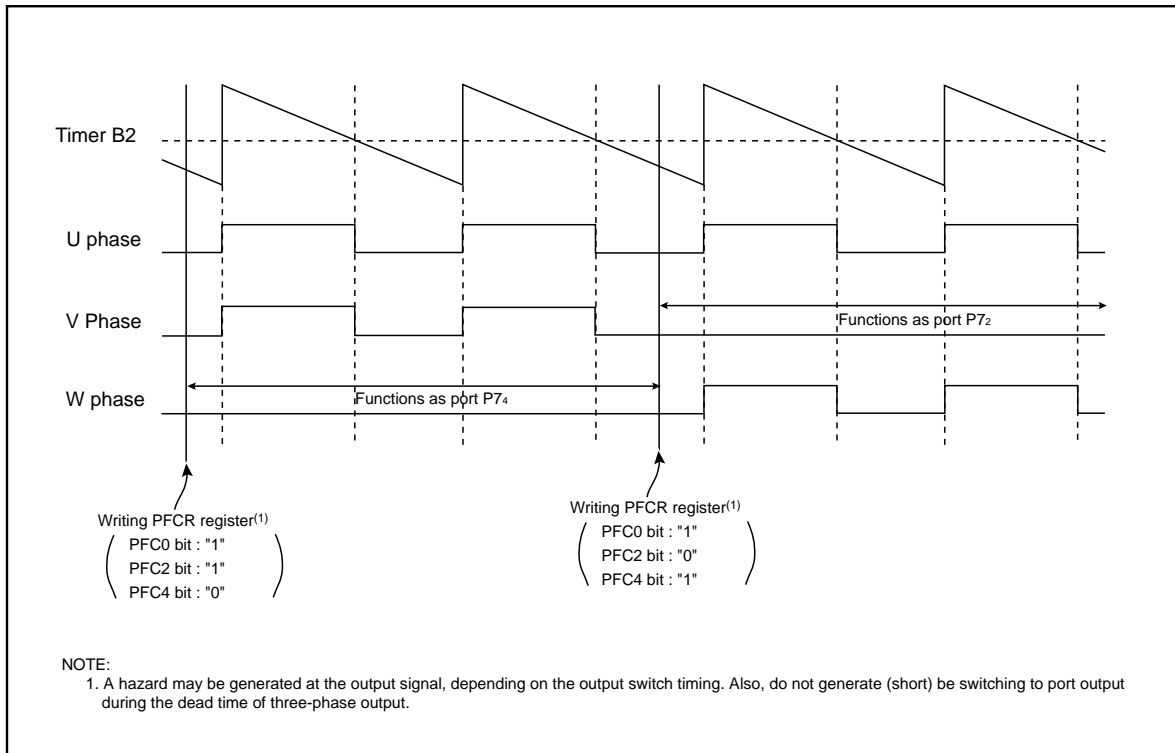


Figure 12.3.2.1. Usage Example of Three-phase/Port output switch function

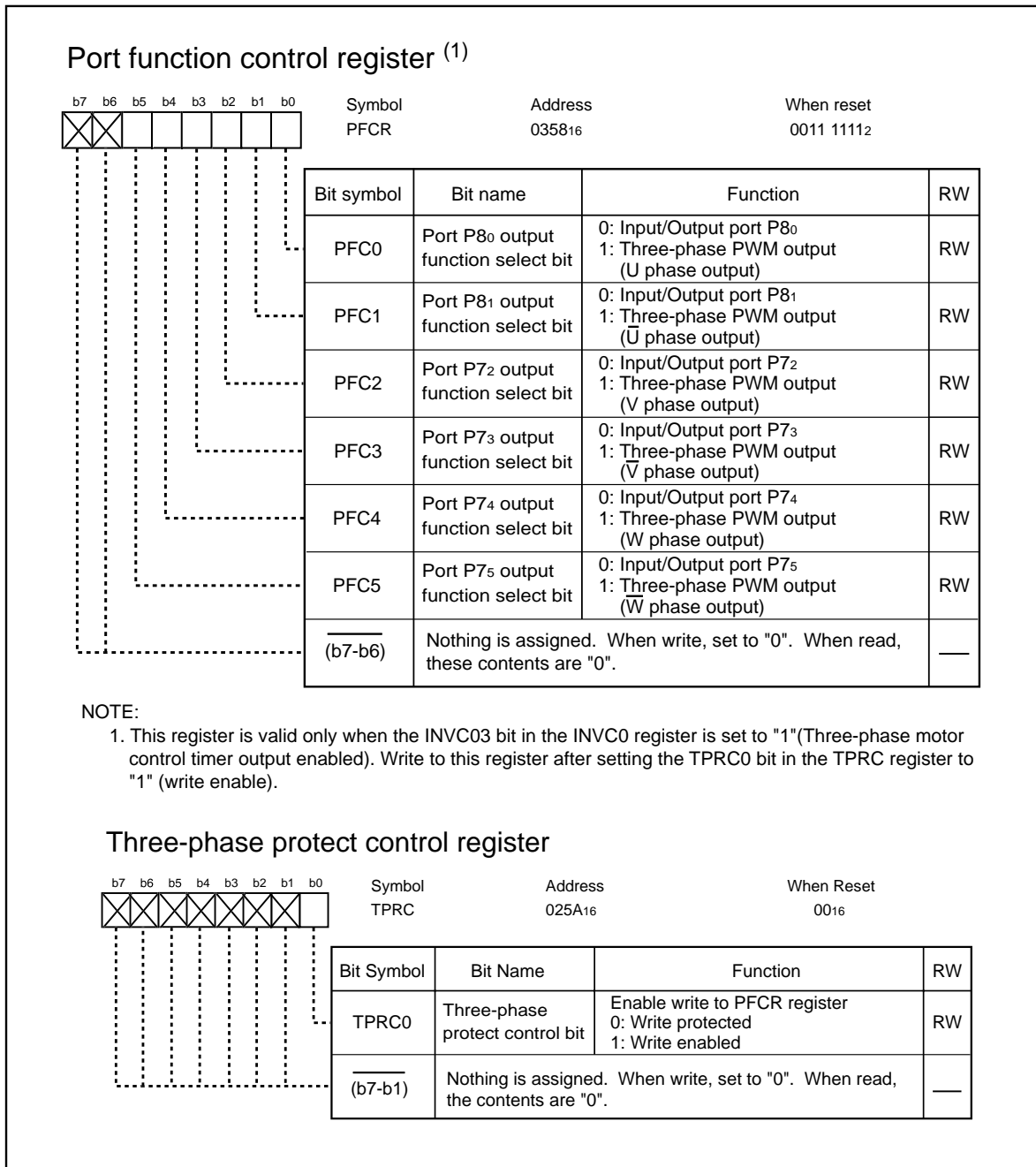


Figure 12.3.2.2. PFCR Register, and TPRC Register

## 13. Serial I/O

**Note**

UART0 is not available in the 42-pin package.

Serial I/O is configured with three channels: UART0 to UART2.

### 13.1. UARTi (i=0 to 2)

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

Figure 13.1.1 shows the block diagram of UARTi. Figures 13.1.2 and 13.1.3 shows the block diagram of the UARTi transmit/receive.

UARTi has the following modes:

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

Figures 13.1.4 to 13.1.9 show the UARTi-related registers.

Refer to tables listing each mode for register setting.

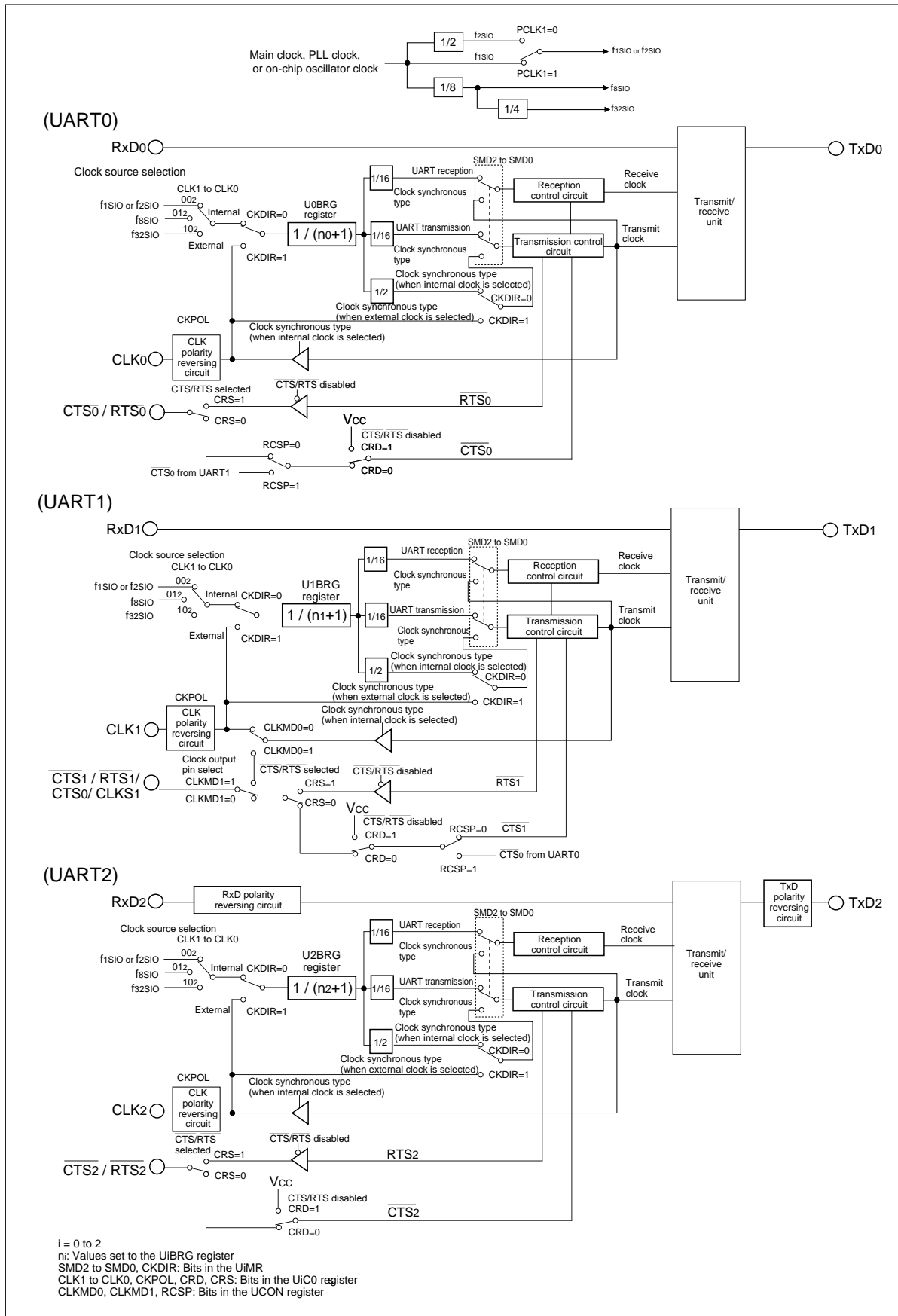


Figure 13.1.1. Block diagram of UART<sub>i</sub> (i = 0 to 2)

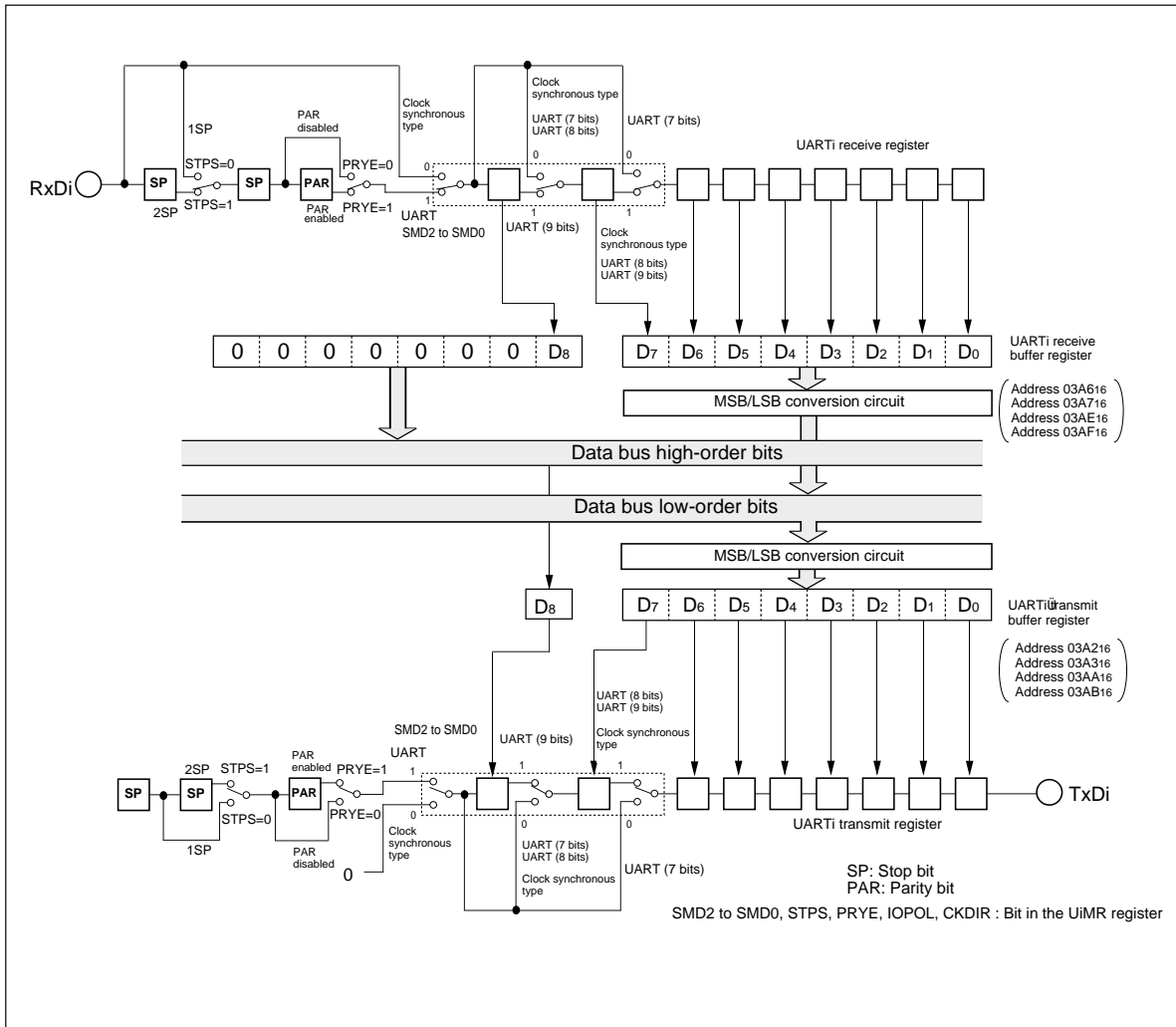


Figure 13.1.2. Block diagram of UARTi (i = 0, 1) transmit/receive unit

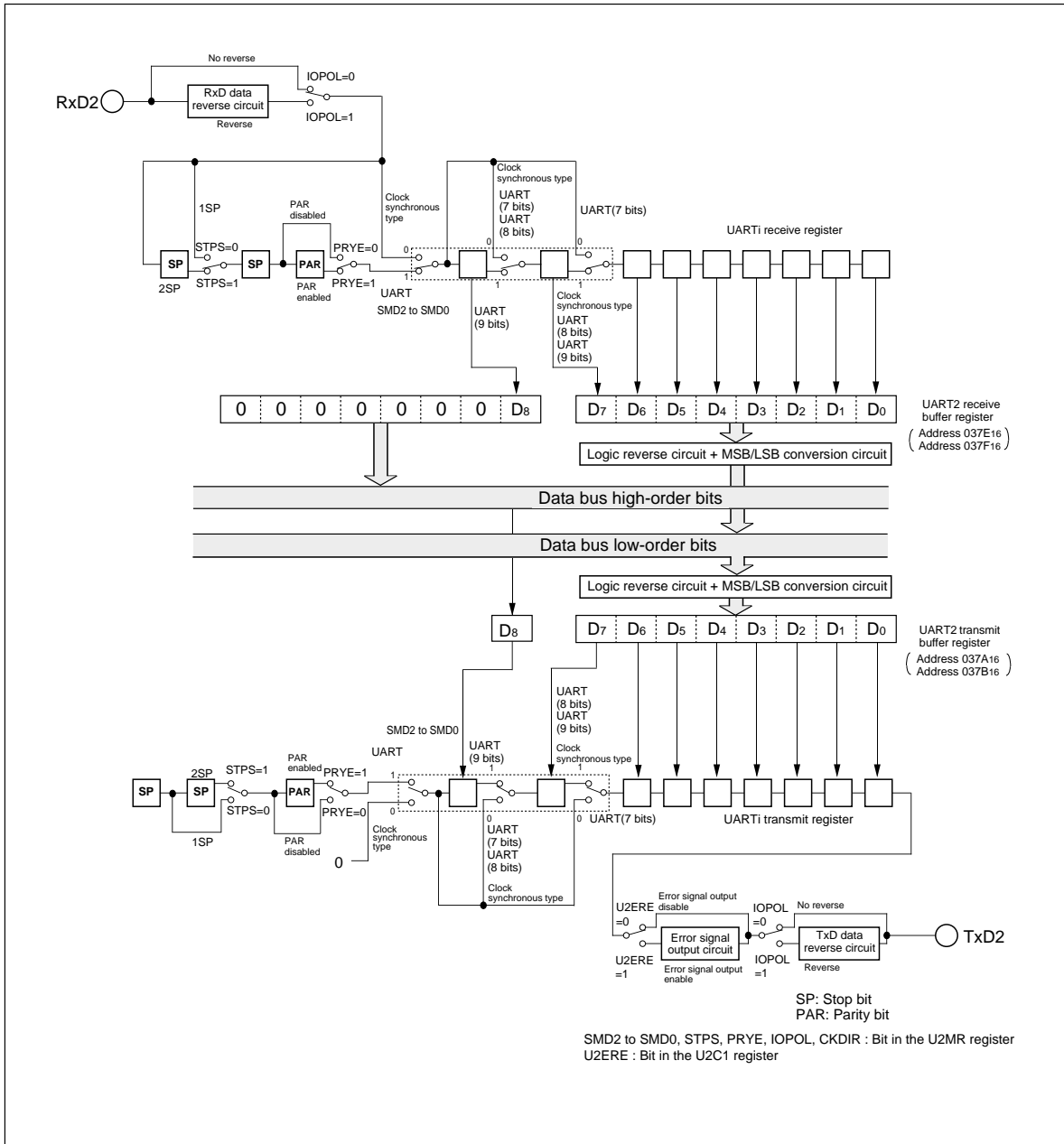


Figure 13.1.3. Block diagram of UART2 transmit/receive unit



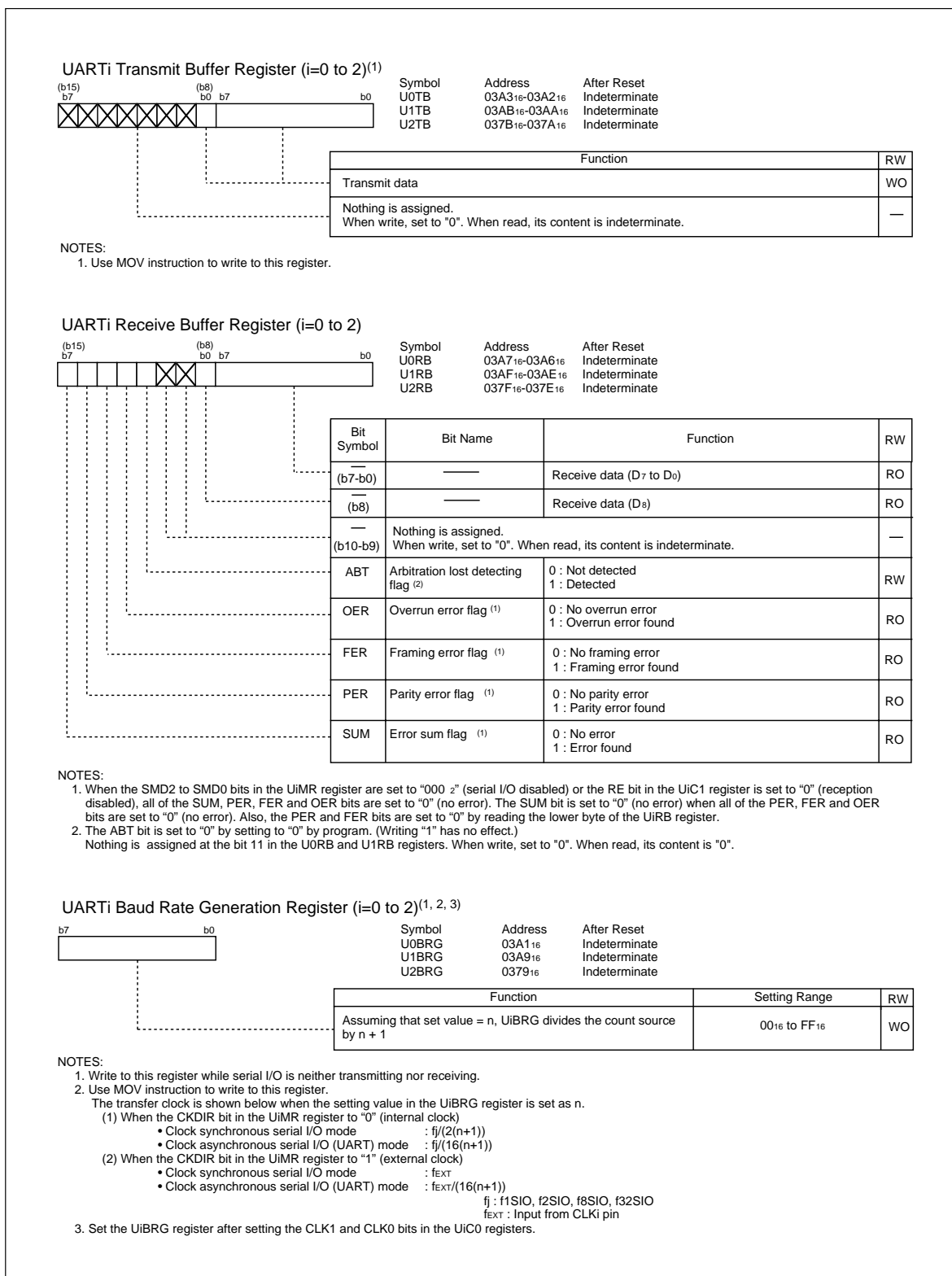


Figure 13.1.4. U0TB to U2TB registers, U0RB to U2RB registers, U0BRG to U2BRG registers

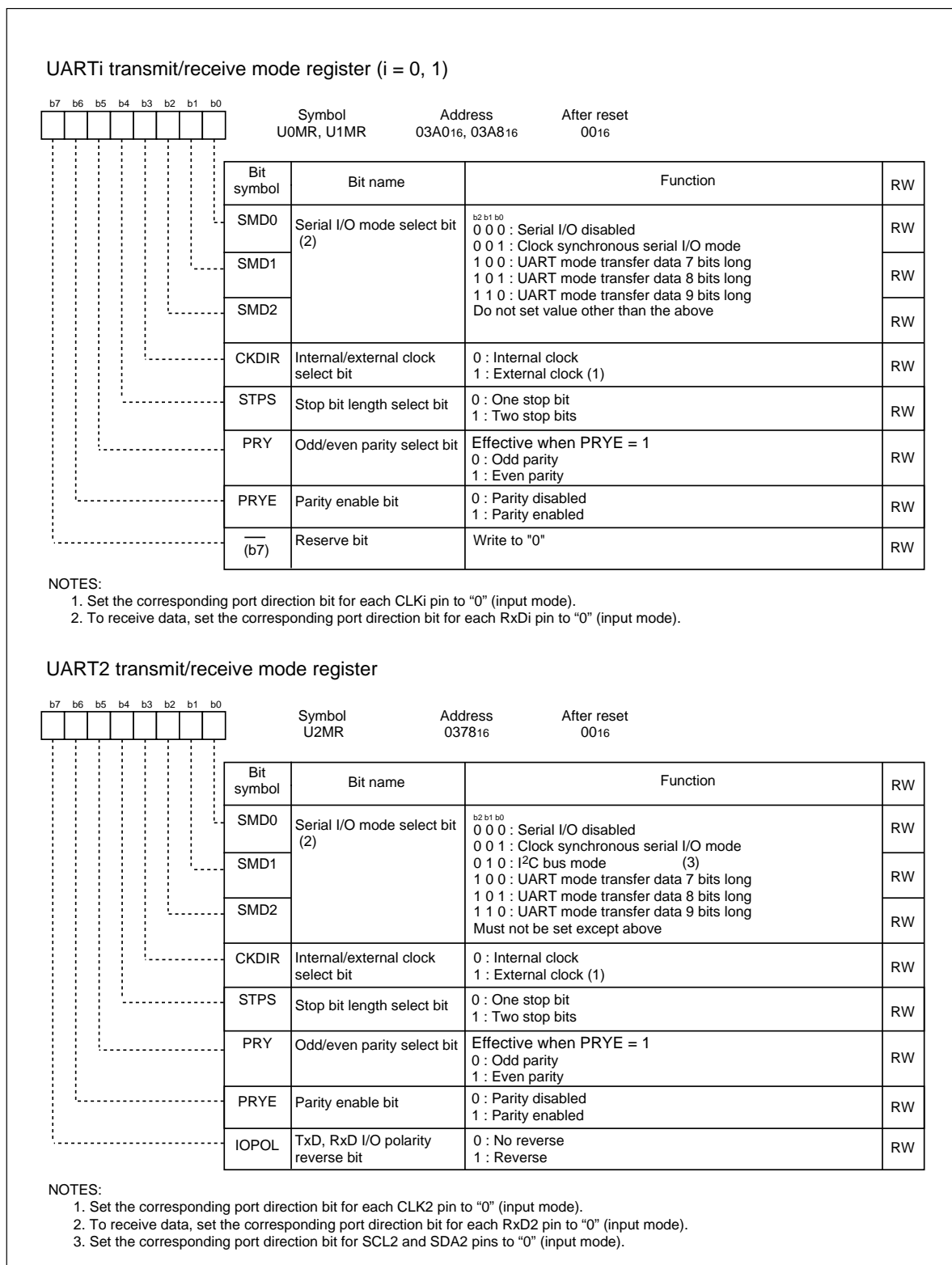


Figure 13.1.5. U0MR to U2MR registers

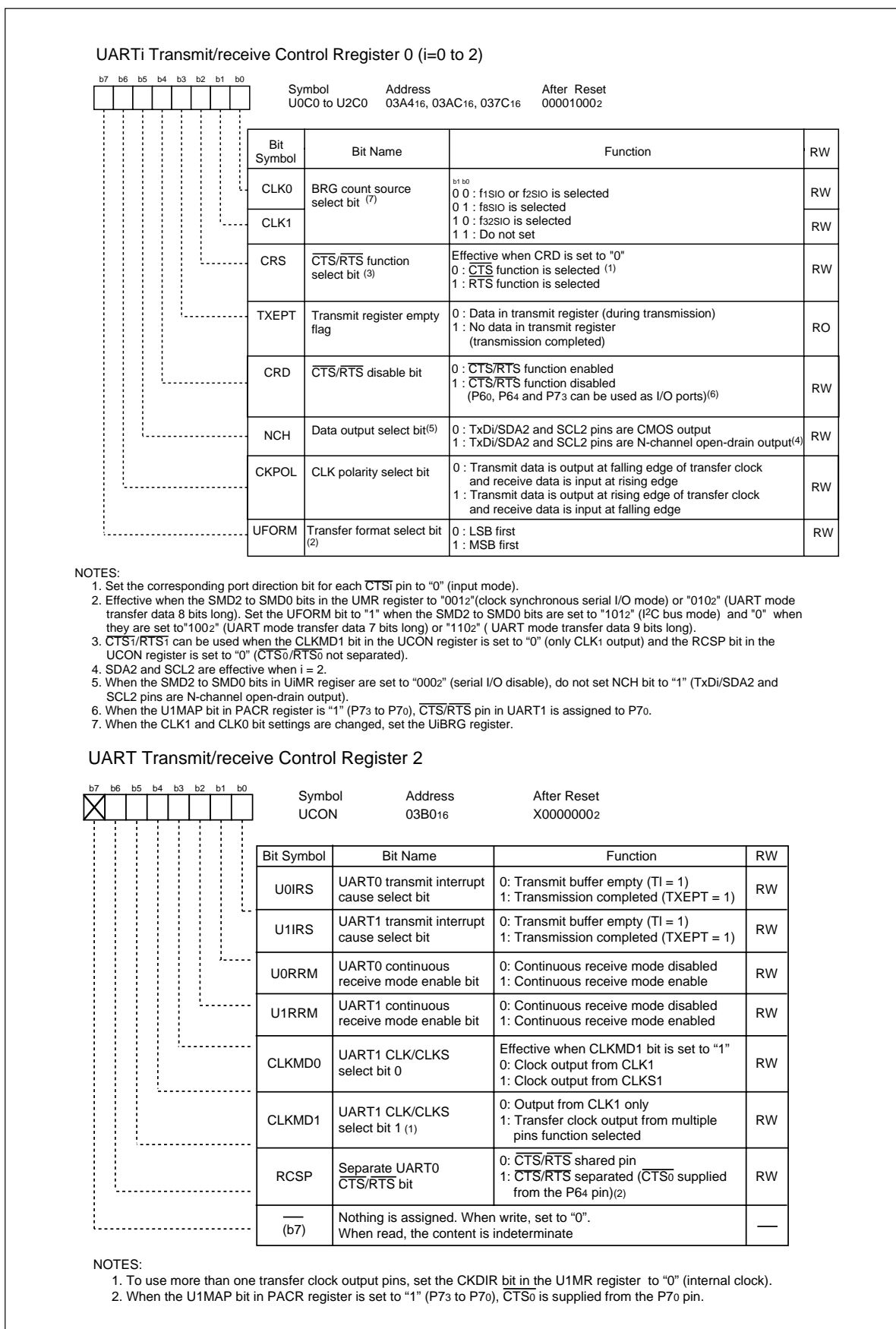


Figure 13.1.6. U0C0 to U2C0 registers and UCON register

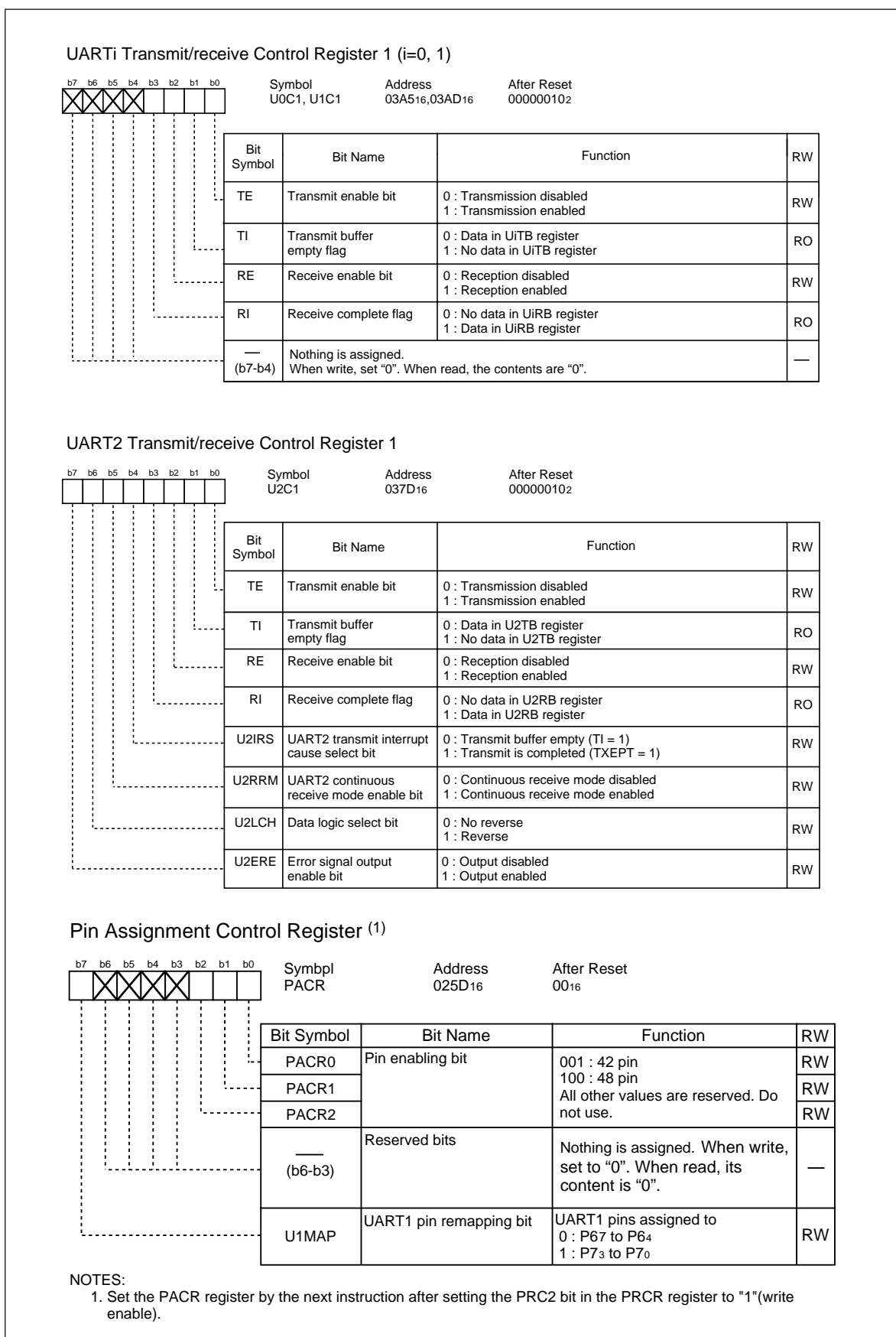


Figure 13.1.7. U0C1 to U2C1 registers, PACR register

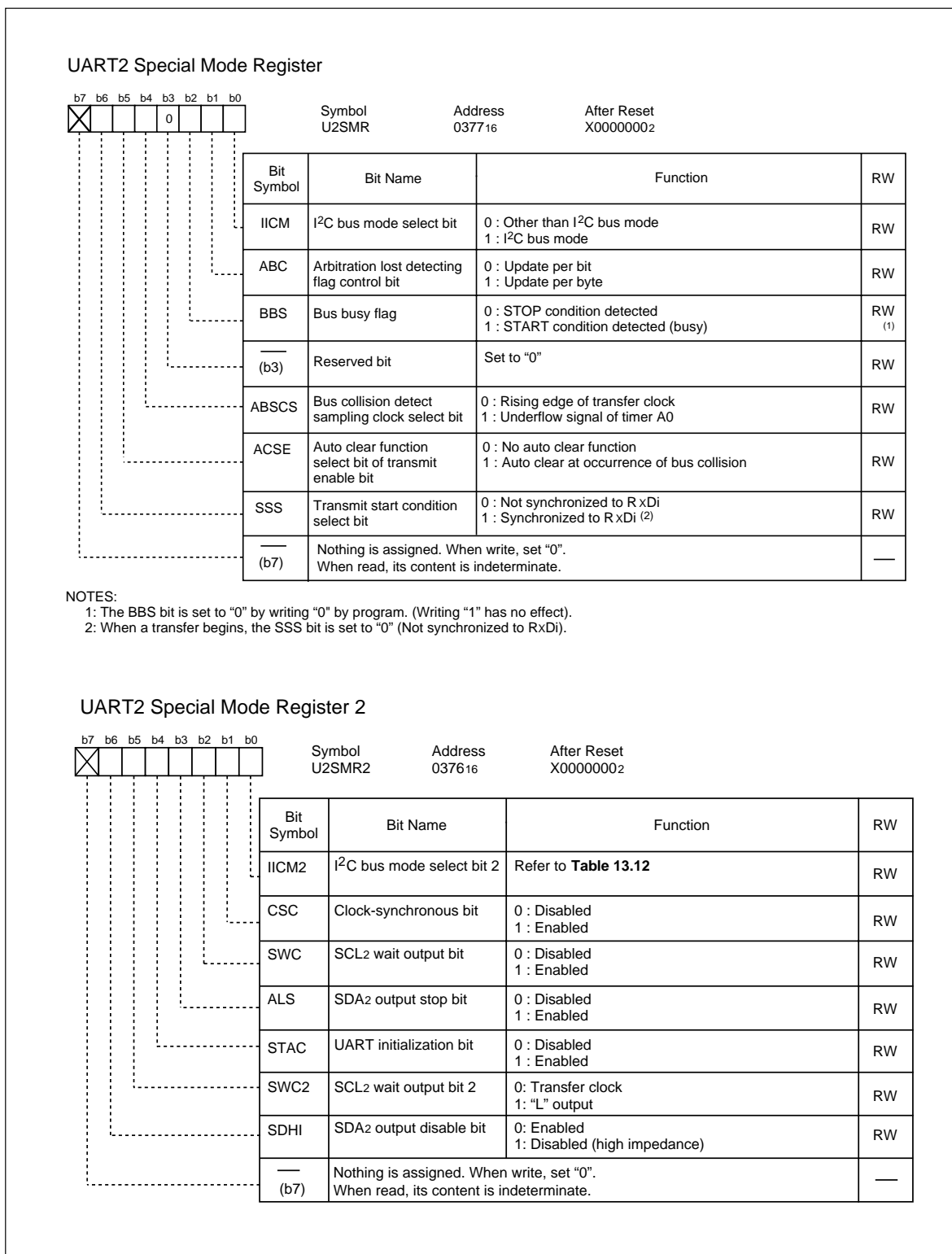


Figure 13.1.8. U2SMR register and U2SMR2 register

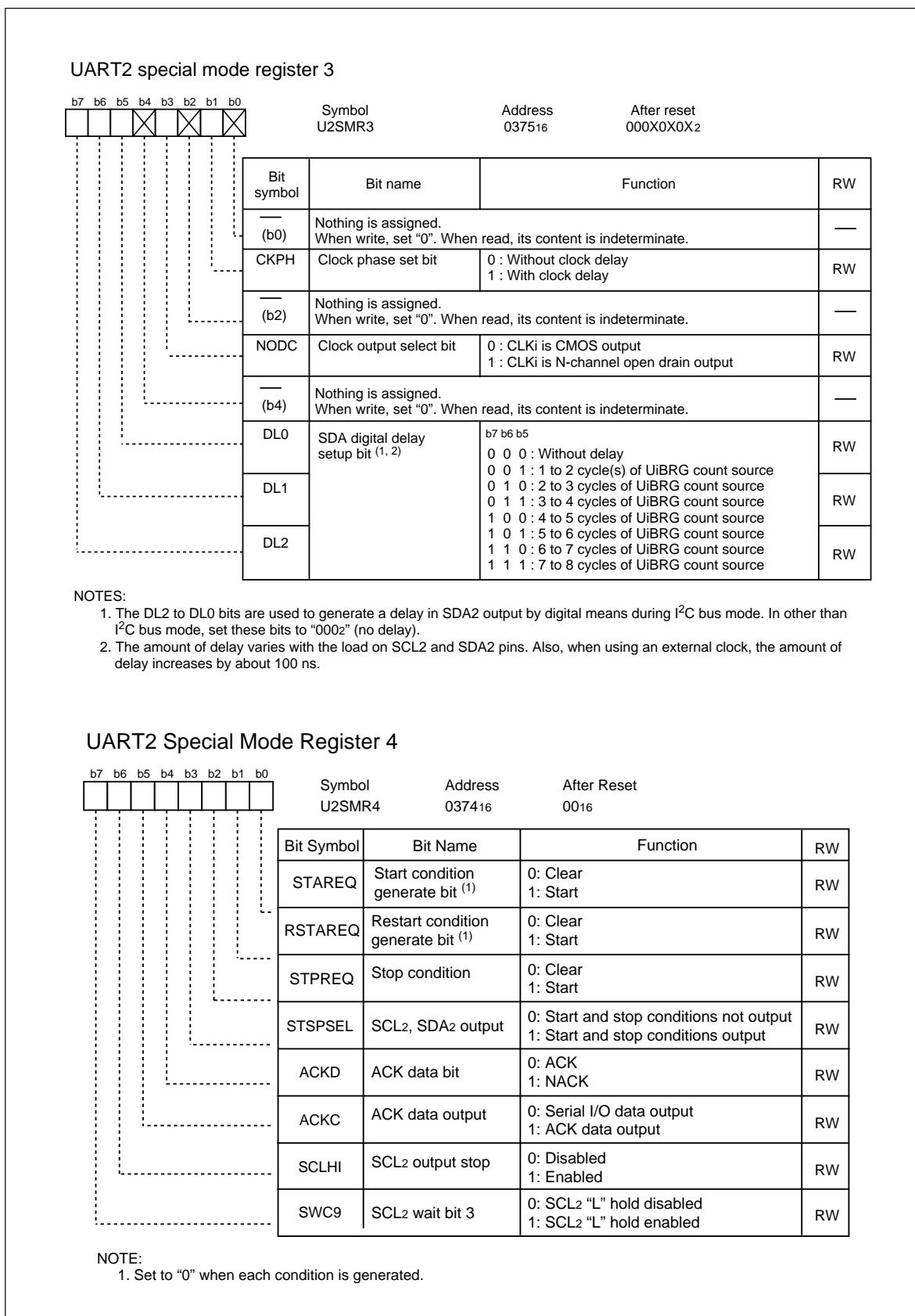


Figure 13.1.9. U2SMR3 register and U2SMR4 register

### 13.1.1. Clock Synchronous serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 13.1.1.1 lists the specifications of the clock synchronous serial I/O mode. Table 13.1.1.2 lists the registers used in clock synchronous serial I/O mode and the register values set.

**Table 13.1.1.1. Clock Synchronous Serial I/O Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>The CKDIR bit in the UiMR(i=0 to 2) register is set to "0" (internal clock) : <math>f_j / (2(n+1))</math>  <math>f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math>. n: Setting value of UiBRG register 00<sub>16</sub> to FF<sub>16</sub></li> <li>The CKDIR bit is set to "1" (external clock) : Input from CLKi pin</li> </ul>
Transmission, reception control	<ul style="list-style-type: none"> <li>Selectable from <math>\overline{CTS}</math> function, RTS function or CTS/RTS function disable</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>Before transmission can start, the following requirements must be met <sup>(1)</sup> <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register is set to "1" (transmission enabled)</li> <li>The TI bit in the UiC1 register is set to "0" (data present in UiTB register)</li> <li>If <math>\overline{CTS}</math> function is selected, input on the <math>\overline{CTS}_i</math> pin is "L"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>Before reception can start, the following requirements must be met <sup>(1)</sup> <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register is set to "1" (reception enabled)</li> <li>The TE bit in the UiC1 register is set to "1" (transmission enabled)</li> <li>The TI bit in the UiC1 register is set to "0" (data present in the UiTB register)</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> <li>The UiIRS bit <sup>(3)</sup> is set to "0" (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)</li> <li>The UiIRS bit is set to "1" (transfer completed): when the serial I/O finished sending data from the UARTi transmit register</li> </ul> </li> <li>For reception <ul style="list-style-type: none"> <li>When transferring data from the UARTi receive register to the UiRB register (at completion of reception)</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error <sup>(2)</sup>  This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 7th bit of the next data</li> </ul>
Select function	<ul style="list-style-type: none"> <li>CLK polarity selection  Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock</li> <li>LSB first, MSB first selection  Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> <li>Continuous receive mode selection  Reception is enabled immediately by reading the UiRB register</li> <li>Switching serial data logic (UART2)  This function reverses the logic value of the transmit/receive data</li> <li>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</li> <li>Separate <math>\overline{CTS}</math>/RTS pins (UART0)  <math>\overline{CTS}_0</math> and <math>\overline{RTS}_0</math> are input/output from separate pins</li> <li>UART1 pin remapping selection  The UART1 pin can be selected from the P67 to P64 or P73 to P70.</li> </ul>

**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.
- If an overrun error occurs, bits 8 to 0 in UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.
- The U0IRS and U1IRS bits respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

**Table 13.1.1. 2. Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode**

Register	Bit	Function
UiTB <sup>(3)</sup>	0 to 7	Set transmission data
UiRB <sup>(3)</sup>	0 to 7	Reception data can be read
	OER	Overflow error flag
UiBRG	0 to 7	Set a transfer rate
UiMR <sup>(3)</sup>	SMD2 to SMD0	Set to "0012"
	CKDIR	Select the internal clock or external clock
	IOPOL(i=2) <sup>(4)</sup>	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	Enable or disable the CTS or RTS function
	NCH	Select TxDi pin output mode
	CKPOL	Select the transfer clock polarity
	UFORM	Select the LSB first or MSB first
UiC1	TE	Set this bit to "1" to enable transmission/reception
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS <sup>(1)</sup>	Select the source of UART2 transmit interrupt
	U2RRM <sup>(1)</sup>	Set this bit to "1" to use UART2 continuous receive mode
	U2LCH <sup>(3)</sup>	Set this bit to "1" to use UART2 inverted data logic
U2ERE <sup>(3)</sup>	Set to "0"	
U2SMR	0 to 7	Set to "0"
U2SMR2	0 to 7	Set to "0"
U2SMR3	0 to 2	Set to "0"
	NODC	Select clock output mode
	4 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set this bit to "1" to use continuous receive mode
	CLKMD0	Select the transfer clock output pin when CLKMD1 = 1
	CLKMD1	Set this bit to "1" to output UART1 transfer clock from two pins
	RCSP	Set this bit to "1" to accept as input the UART0 CTS <sub>0</sub> signal from the P64 pin or P70 pin
	7	Set to "0"

## NOTES:

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

i=0 to 2



Table 13.1.1.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Table 13.3 shows pin functions for the case where the multiple transfer clock output pin select function is deselected. Table 13.1.1.4 lists the P64 pin functions during clock synchronous serial I/O mode.

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

**Table 13.1.1.3. Pin Functions<sup>(1)</sup> (When Not Select Multiple Transfer Clock Output Pin Function)**

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

NOTE:

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

**Table 13.1.1.4. P64 Pin Functions<sup>(1)</sup>**

Pin function	Bit set value					
	U1C0 register		UCON register			PD6 register
	CRD	CRS	RCSP	CLKMD1	CLKMD0	PD6_4
P64	1	—	0	0	—	Input: 0, Output: 1
CTS <sub>1</sub>	0	0	0	0	—	0
RTS <sub>1</sub>	0	1	0	0	—	—
CTS <sub>0</sub> <sup>(2)</sup>	0	0	1	0	—	0
CLKS <sub>1</sub>	—	—	—	1 <sup>(3)</sup>	1	—

NOTES:

1. When the U1MAP bit in PACR register is "1" (P73 to P70), this table lists the P70 functions.
2. In addition to this, set the CRD bit in the U0C0 register to "0" (CT00/RT00 enabled) and the CRS bit in the U0C0 register to "1" (RTS0 selected).
3. When the CLKMD1 bit is set to "1" and the CLKMD0 bit is set to "0", the following logiclevels are output:
  - High if the CLKPOL bit in the U1C0 register is set to "0"
  - Low if the CLKPOL bit in the U1C0 register is set to "1"

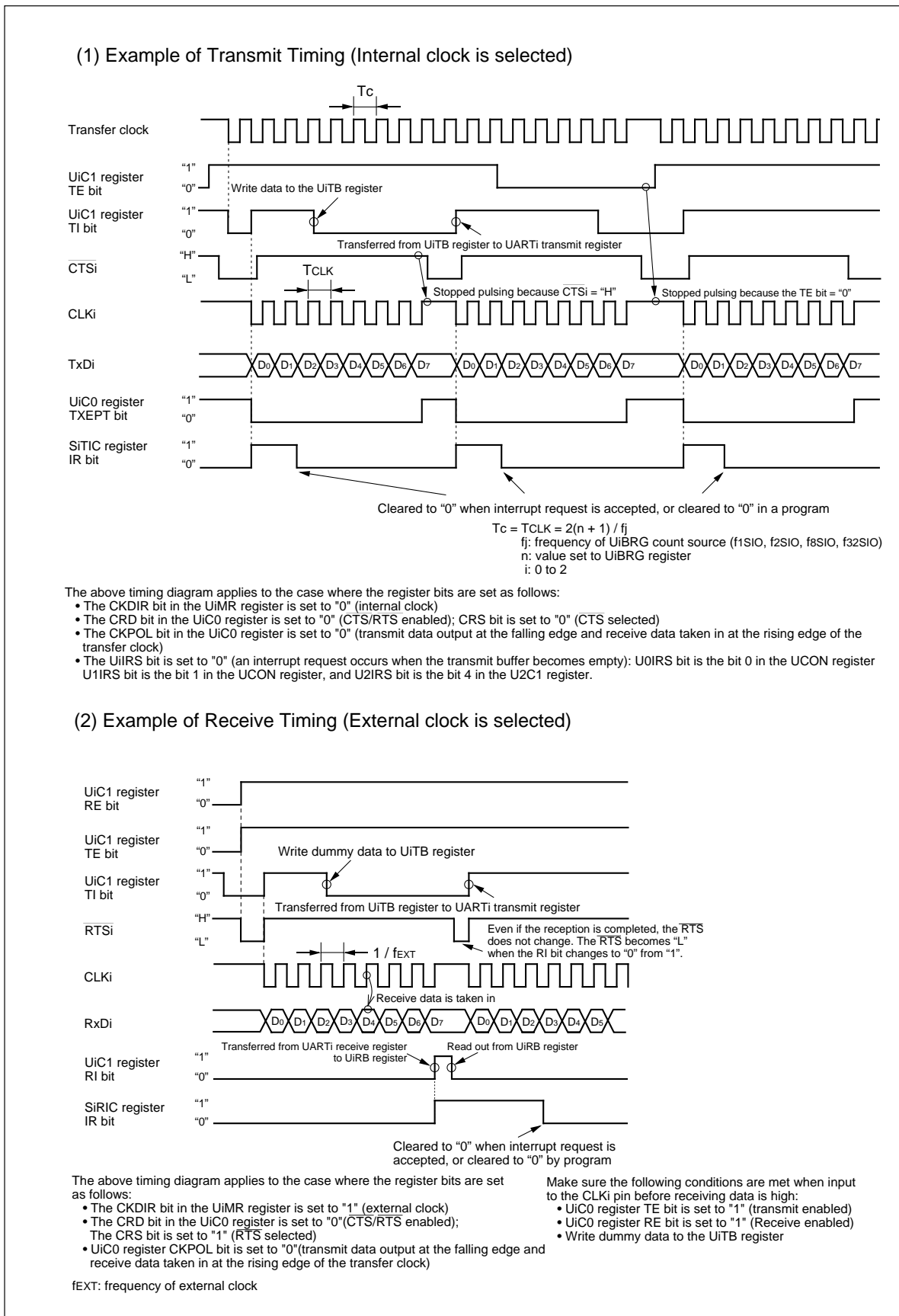


Figure 13.1.1.1. Typical transmit/receive timings in clock synchronous serial I/O mode

### 13.1.1.1 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 the SMD2 to SMD0 bits in the UiMR register to "0002" (Serial I/O disabled)
  - (3) Set the SMD2 to SMD0 bits in the UiMR register to "0012" (Clock synchronous serial I/O mode)
  - (4) Set the RE bit in the UiC1 register to "1" (reception enabled)
- Resetting the UiTB register (i=0 to 2)
  - (1) Set the SMD2 to SMD0 bits in the UiMR register to "0002" (Serial I/O disabled)
  - (2) Set the SMD2 to SMD0 bits in the UiMR register to "0012" (Clock synchronous serial I/O mode)
  - (3) "1" is written to RE bit in the UiC1 register (reception enabled), regardless to the TE bit in the UiC1 register.

### 13.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register (i = 0 to 2) to select the transfer clock polarity. Figure 13.1.1.2.1 shows the polarity of the transfer clock.

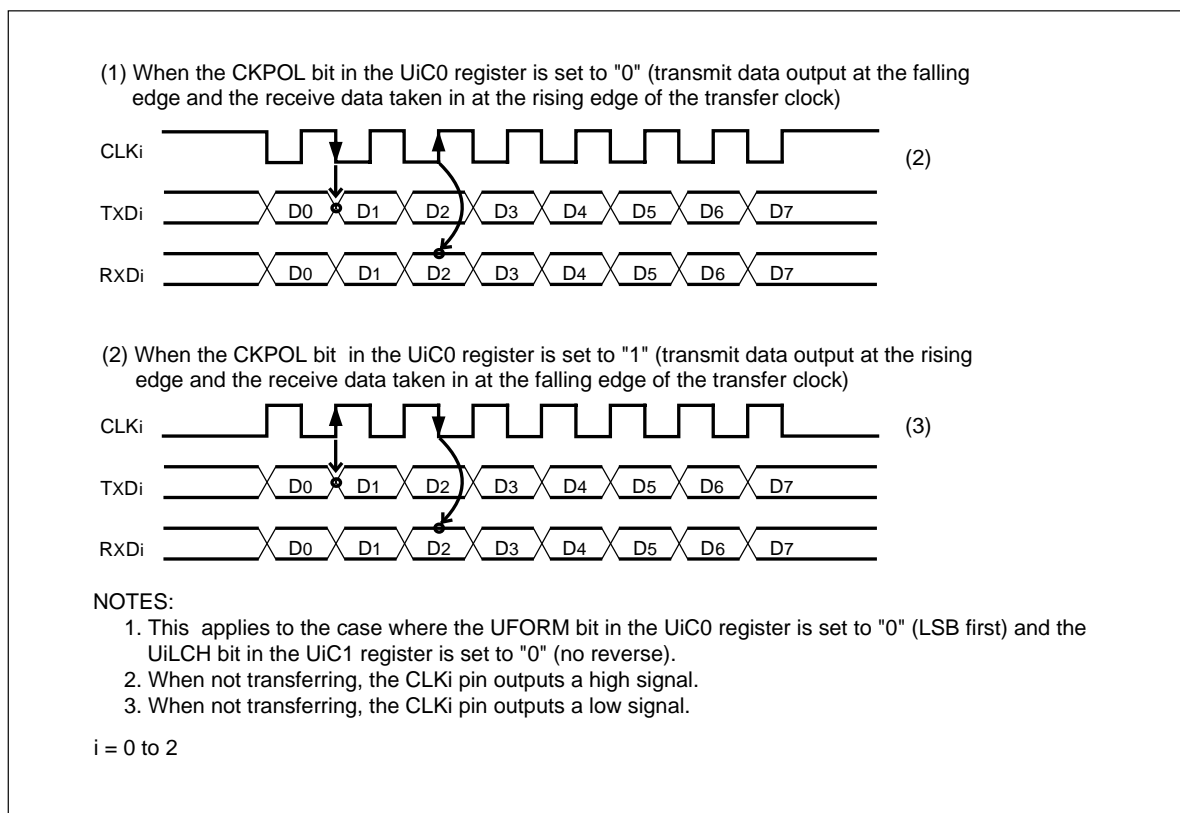


Figure 13.1.1.2.1. Polarity of transfer clock

### 13.1.1.3 LSB First/MSB First Select Function

Use the UFORM bit in the UiC0 register ( $i = 0$  to 2) to select the transfer format. Figure 13.1.1.3.1 shows the transfer format.

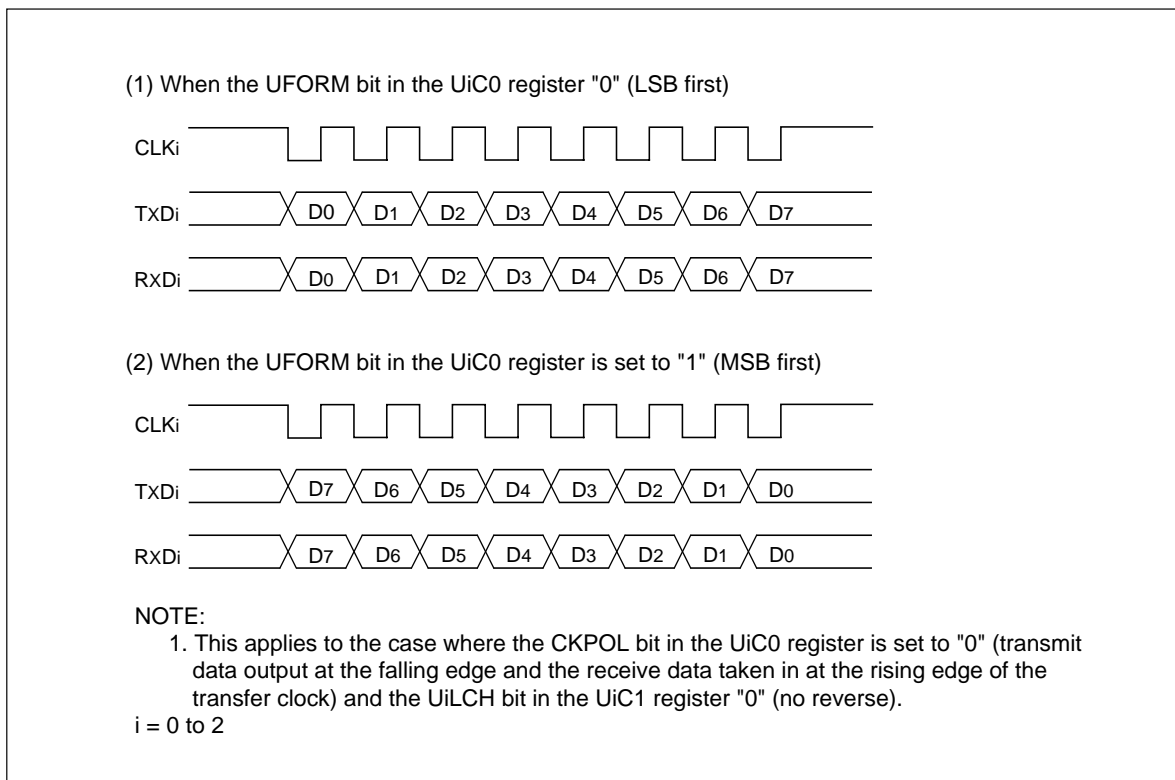


Figure 13.1.1.3.1 Transfer format

### 13.1.1.4 Continuous receive mode

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

### 13.1.1.5 Serial data logic switch function (UART2)

When the U2LCH bit in the U2C1 register is set to "1" (reverse), the data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. Figure 13.1.1.4.1 shows serial data logic.

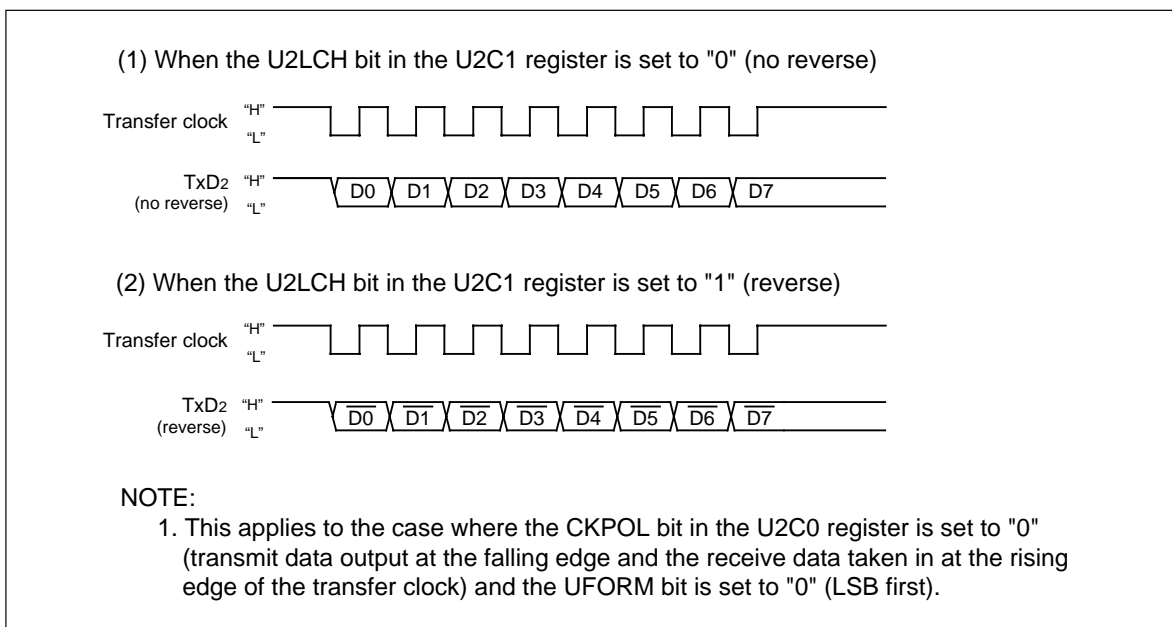


Figure 13.1.1.4.1. Serial data logic switch timing

### 13.1.1.6 Transfer clock output from multiple pins function (UART1)

The CLKMD1 to CLKMD0 bits in the UCON register can choose one from two transfer clock output pins. (See Figure 13.1.1.6.1) This function is valid when the internal clock is selected for UART1.

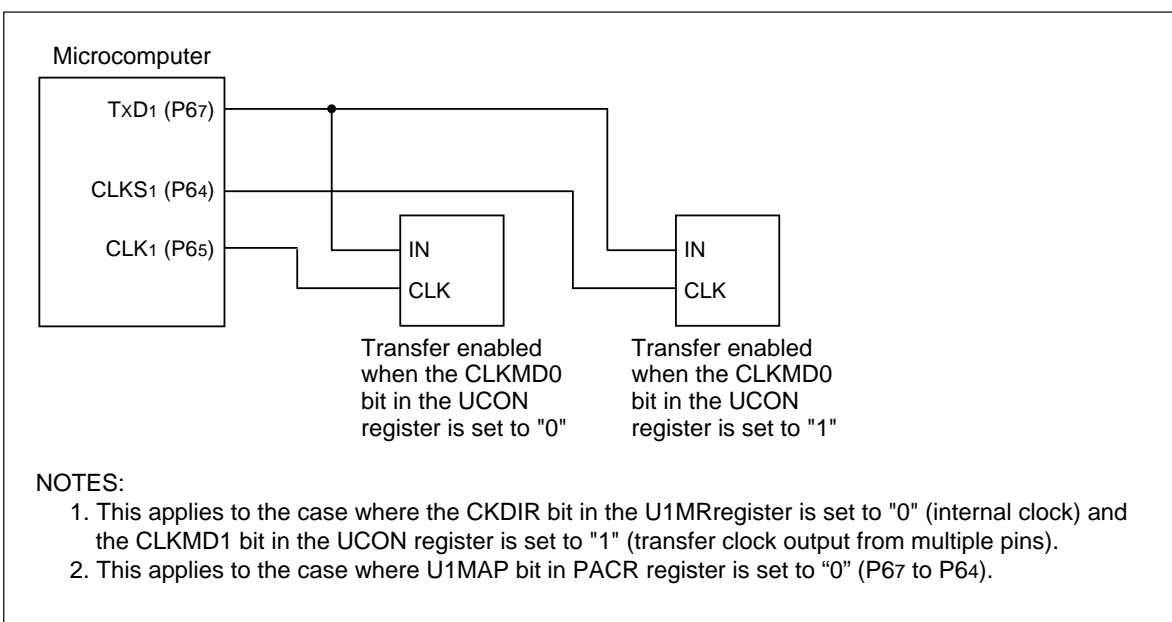


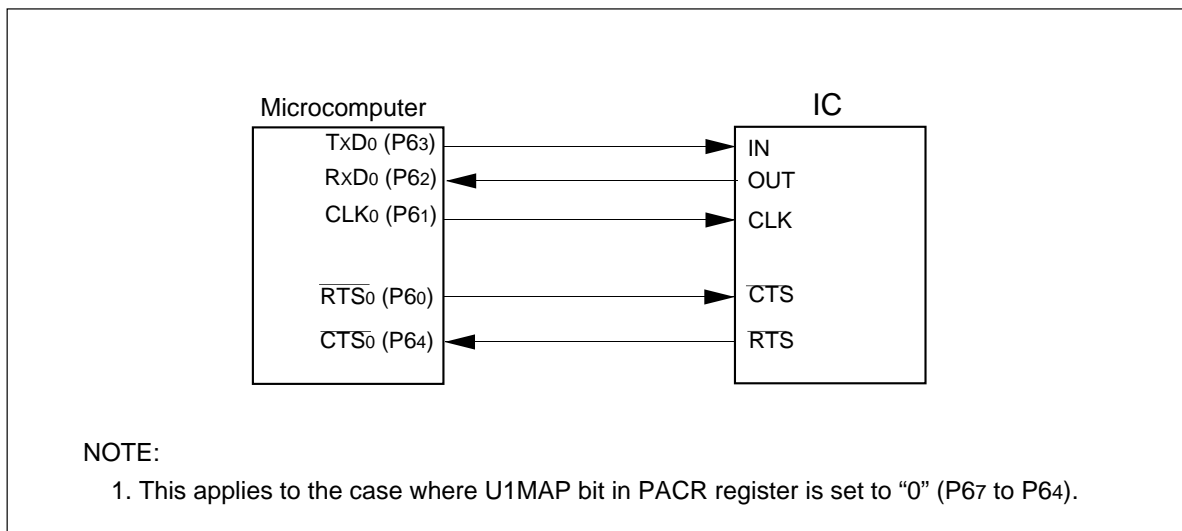
Figure 13.1.1.6.1 Transfer Clock Output From Multiple Pins

### 13.1.1.7 $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function (UART0)

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

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

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



**Figure 13.1.1.7.1. CTS/RTS separate function usage**

### 13.1.2. Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables 13.1.2.1 lists the specifications of the UART mode.

**Table 13.1.2.1. UART Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Character bit (transfer data): Selectable from 7, 8 or 9 bits</li> <li>• Start bit: 1 bit</li> <li>• Parity bit: Selectable from odd, even, or none</li> <li>• Stop bit: Selectable from 1 or 2 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• The CKDIR bit in the UiMR(i=0 to 2) register is set to "0" (internal clock) : <math>f_j/(16(n+1))</math>  <math>f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math>. n: Setting value of UiBRG register 00<sub>16</sub> to FF<sub>16</sub></li> <li>• CKDIR bit is set to "1" (external clock) : <math>f_{EXT}/(16(n+1))</math>  <math>f_{EXT}</math>: Input from CLKi pin. n: Setting value of UiBRG register 00<sub>16</sub> to FF<sub>16</sub></li> </ul>
Transmission, reception control	<ul style="list-style-type: none"> <li>• Selectable from CTS function, RTS function or CTS/RTS function disable</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>• Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> <li>– The TE bit in the UiC1 register is set to "1" (transmission enabled)</li> <li>– The TI bit in the UiC1 register "0" (data present in UiTB register)</li> <li>– If CTS function is selected, input "L" to the CTSi pin</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>• Before reception can start, the following requirements must be met <ul style="list-style-type: none"> <li>– The RE bit in the UiC1 register is set to "1" (reception enabled)</li> <li>– Start bit detection</li> </ul> </li> </ul>
Interrupt request generation timing data from	<ul style="list-style-type: none"> <li>• For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> <li>– The UiIRS bit <sup>(2)</sup> is set to "0" (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)</li> <li>– The UiIRS bit is set to "1" (transfer completed): when the serial I/O finished sending the UARTi transmit register</li> </ul> </li> <li>• For reception  When transferring data from the UARTi receive register to the UiRB register (at completion of reception)</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error <sup>(1)</sup>  This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the bit one before the last stop bit of the next data</li> <li>• Framing error  This error occurs when the number of stop bits set is not detected</li> <li>• Parity error  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</li> <li>• Error sum flag  This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered</li> </ul>
Select function	<ul style="list-style-type: none"> <li>• LSB first, MSB first selection  Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> <li>• Serial data logic switch (UART2)  This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed.</li> <li>• Tx/D, Rx/D I/O polarity switch (UART2)  This function reverses the polarities of the Tx/D pin output and Rx/D pin input. The logic levels of all I/O data is reversed.</li> <li>• Separate CTS/RTS pins (UART0)  <math>\overline{CTS_0}</math> and <math>\overline{RTS_0}</math> are input/output from separate pins</li> <li>• UART1 pin remapping selection  The UART1 pin can be selected from the P67 to P64 or P73 to P70.</li> </ul>

**NOTES:**

1. If an overrun error occurs, bits 8 to 0 in UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.
2. The U0IRS and U1IRS bits respectively are the bits "0" and "1" in the UCON register; the U2IRS bit is the bit 4 in the U2C1 register.

**Table 13.1.2.2. Registers to Be Used and Settings in UART Mode**

Register	Bit	Function
UiTB	0 to 8	Set transmission data <sup>(1)</sup>
UiRB	0 to 8	Reception data can be read <sup>(1)</sup>
	OER,FER,PER,SUM	Error flag
UiBRG	0 to 7	Set a transfer rate
UiMR	SMD2 to SMD0	Set these bits to '1002' when transfer data is 7 bits long Set these bits to '1012' when transfer data is 8 bits long Set these bits to '1102' when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
	IOPOL(i=2) <sup>(4)</sup>	Select the TxD/RxD input/output polarity
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	CRS	Select CTS or RTS to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the CTS or RTS function
	NCH	Select TxDi pin output mode
	CKPOL	Set to "0"
	UFORM	LSB first or MSB first can be selected when transfer data is 8 bits long. Set this bit to "0" when transfer data is 7 or 9 bits long.
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS <sup>(2)</sup>	Select the source of UART2 transmit interrupt
	U2RRM <sup>(2)</sup>	Set to "0"
	U2LCH <sup>(3)</sup>	Set this bit to "1" to use UART2 inverted data logic
	U2ERE <sup>(3)</sup>	Set to "0"
U2SMR	0 to 7	Set to "0"
U2SMR2	0 to 7	Set to "0"
U2SMR3	0 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1	Set to "0"
	RCSP	Set this bit to "1" to accept as input the UART0 $\overline{CTS_0}$ signal from the P64 pin or P70 pin
	7	Set to "0"

## NOTES:

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

i=0 to 2



Table 13.1.2.3 lists the functions of the input/output pins during UART mode. Table 13.1.2.4 lists the P64 pin functions during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an "H". (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

**Table 13.1.2.3. I/O Pin Functions in UART mode<sup>(1)</sup>**

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

NOTE:

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

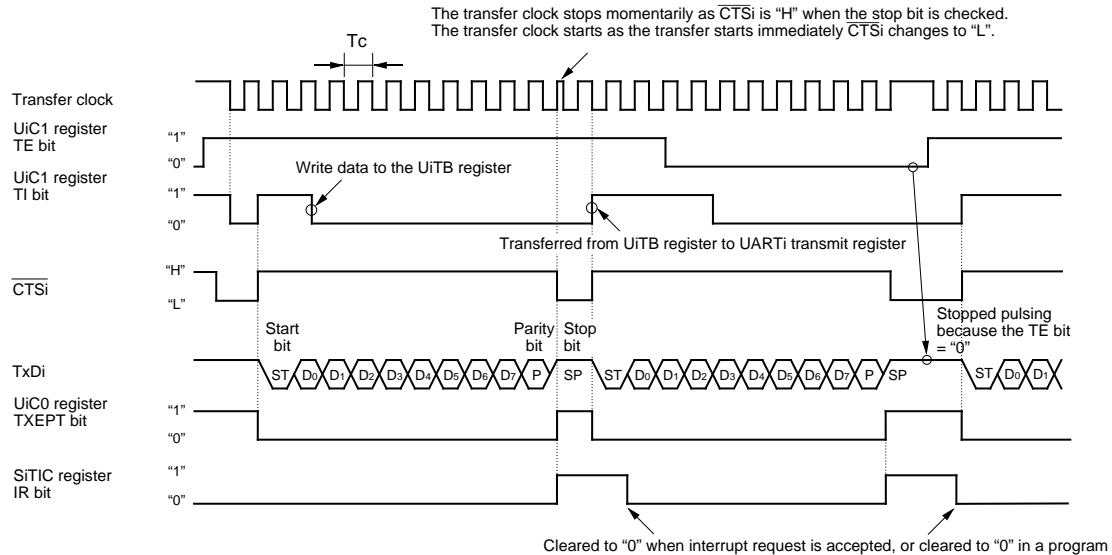
**Table 13.1.2.4. P64 Pin Functions in UART mode<sup>(1)</sup>**

Pin function	Bit set value				
	U1C0 register		U1CON register		PD6 register
	CRD	CRS	RCSP	CLKMD1	PD6_4
P64	1	—	0	0	Input: 0, Output: 1
$\overline{\text{CTS}}_1$	0	0	0	0	0
$\overline{\text{RTS}}_1$	0	1	0	0	—
$\overline{\text{CTS}}_0$ <sup>(2)</sup>	0	0	1	0	0

NOTES:

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

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



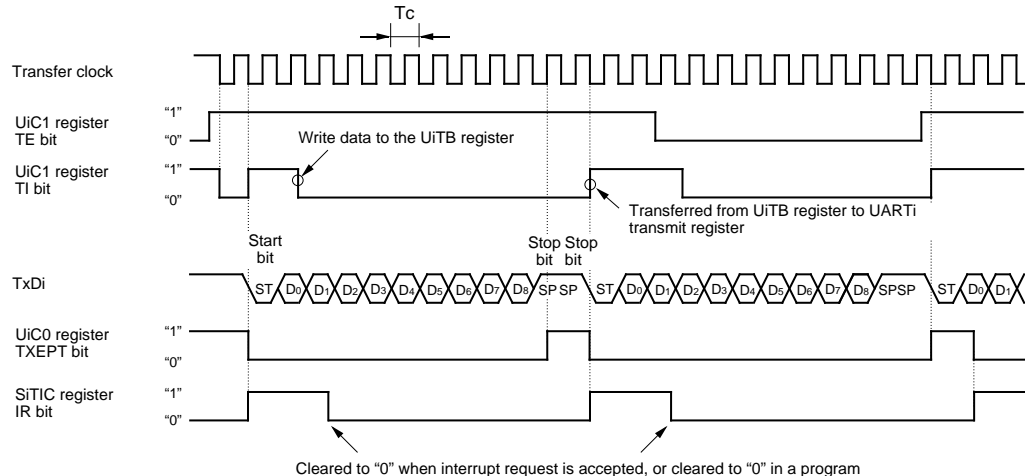
The above timing diagram applies to the case where the register bits are set as follows:

- Set the PRYE bit in the UiMR register to "1" (parity enabled)
- Set the STPS bit in the UiMR register to "0" (1 stop bit)
- Set the CRD bit in the UiC0 register to "0" (CTS/RTS enabled), the CRS bit to "0" (CTS selected)
- Set the UiIRS bit to "1" (an interrupt request occurs when transmit completed):  
 UiIRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$$Tc = 16(n + 1) / fj \text{ or } 16(n + 1) / fEXT$$

$fj$  : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)  
 $fEXT$  : frequency of UiBRG count source (external clock)  
 $n$  : value set to UiBRG  
 $i$  : 0 to 2

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



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

- Set the PRYE bit in the UiMR register to "0" (parity disabled)
- Set the STPS bit in the UiMR register to "1" (2 stop bits)
- Set the CRD bit in the UiC0 register to "1" (CTS/RTS disabled)
- Set the UiIRS bit to "0" (an interrupt request occurs when transmit buffer becomes empty):  
 UiIRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$$Tc = 16(n + 1) / fj \text{ or } 16(n + 1) / fEXT$$

$fj$  : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)  
 $fEXT$  : frequency of UiBRG count source (external clock)  
 $n$  : value set to UiBRG  
 $i$  : 0 to 2

Figure 13.1.2.1. Typical transmit timing in UART mode (UART0, UART1)

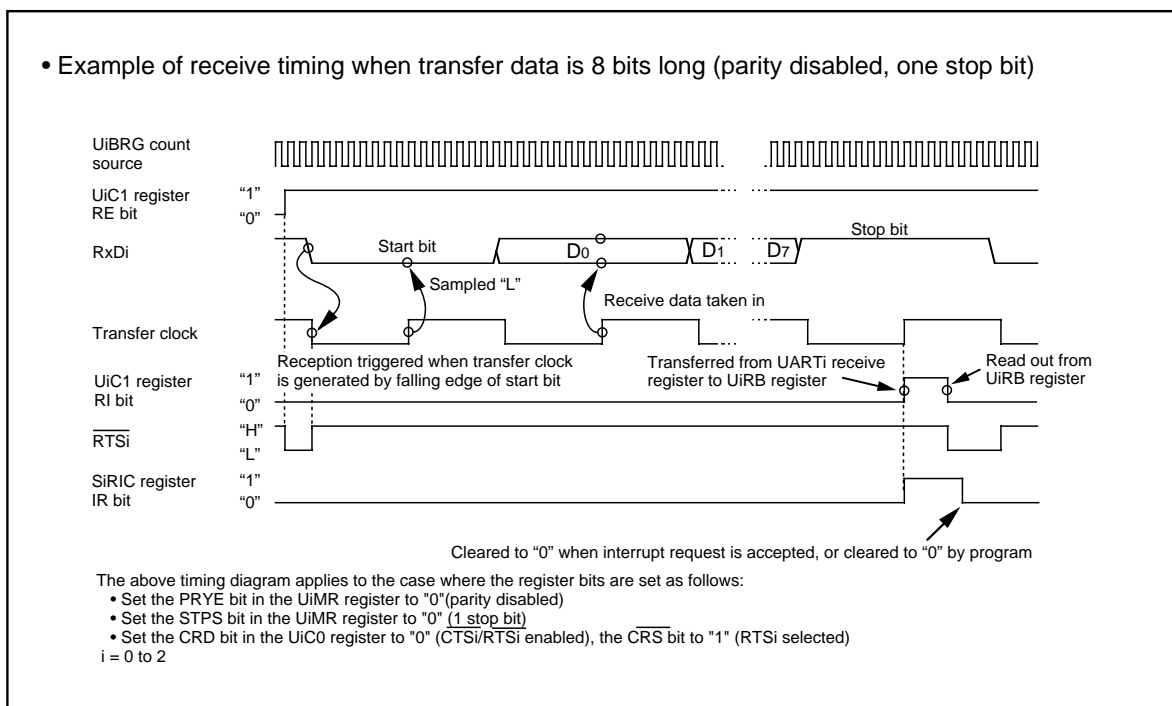


Figure 13.1.2.2. Receive Operation

### 13.1.2.1. Bit Rates

In UART mode, the frequency set by the UIRBG register ( $i=0$  to  $2$ ) divided by 16 become the bit rates. Table 13.1.2.1.1 lists example of bit rate and settings.

Table 13.1.2.1.1 Example of Bit Rates and Settings

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

### 13.1.2.2. Counter Measure for Communication Error

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

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

### 13.1.2.3. LSB First/MSB First Select Function

As shown in Figure 14.1.2.3.1, use the UFORM bit in the UiC0 register to select the transfer format. This function is valid when transfer data is 8 bits long.

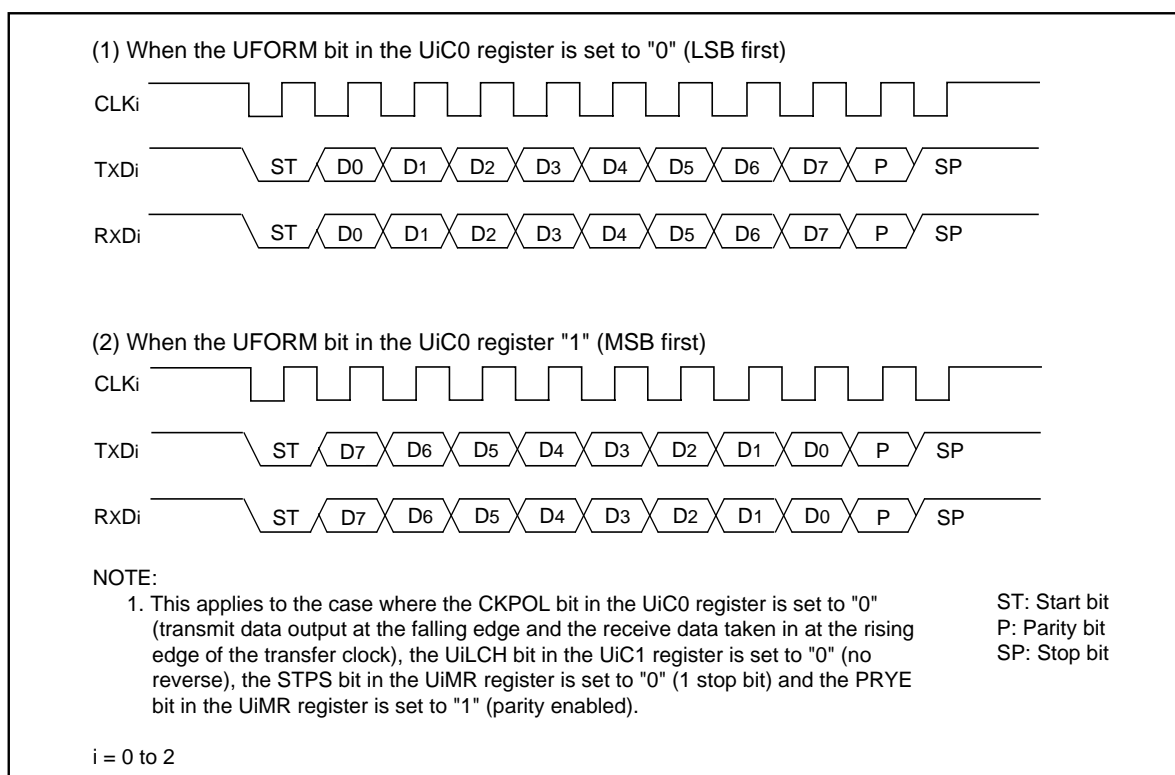
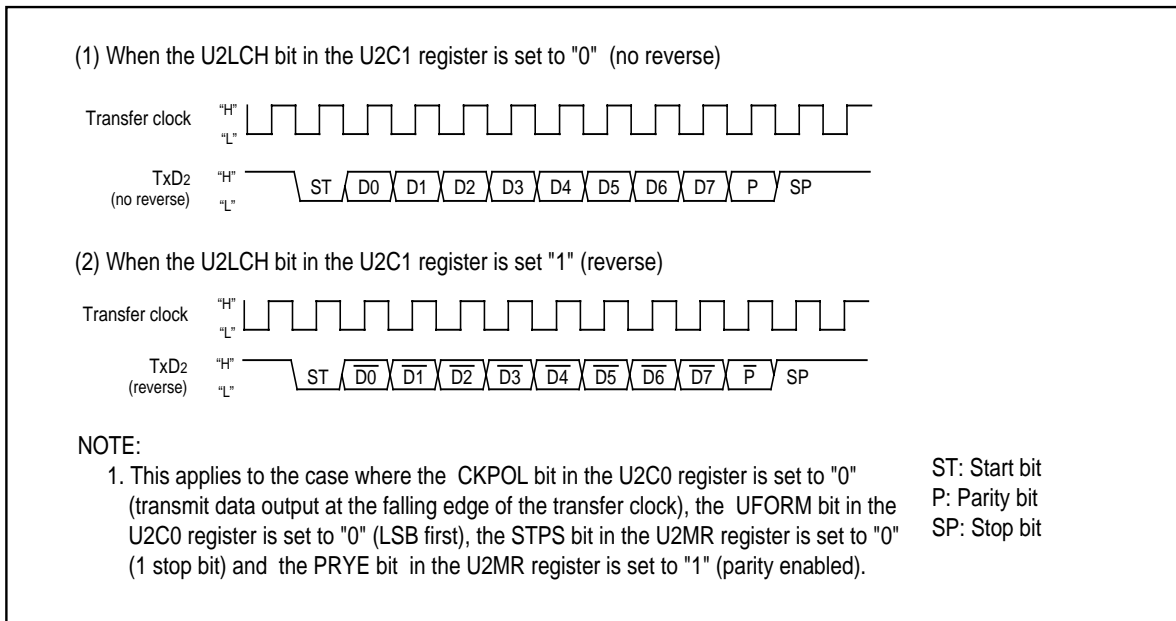


Figure 13.1.2.3.1. Transfer Format

### 13.1.2.4. Serial Data Logic Switching Function (UART2)

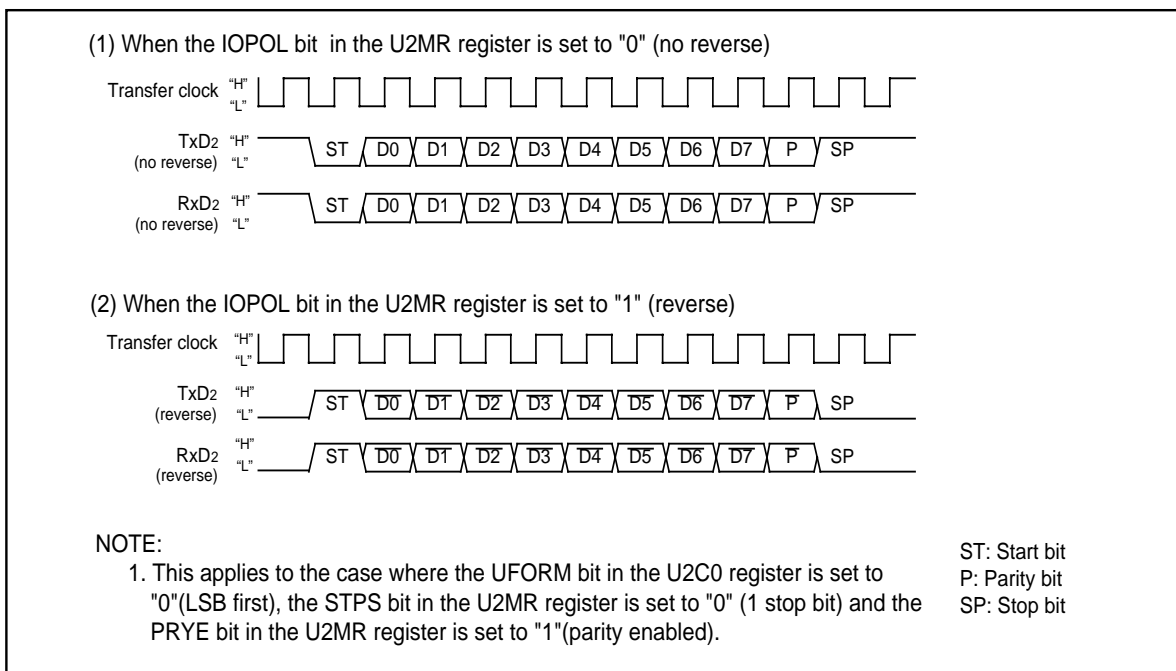
The data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. Figure 13.1.2.4.1 shows serial data logic.



**Figure 13.1.2.4.1. Serial Data Logic Switching**

### 13.1.2.5. Tx D and Rx D I/O Polarity Inverse Function (UART2)

This function inverses the polarities of the Tx D2 pin output and Rx D2 pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inverted. Figure 13.1.2.5.1 shows the Tx D pin output and Rx D pin input polarity inverse.



**Figure 13.1.2.5.1. Tx D and Rx D I/O Polarity Inverse**

### 13.1.2.6. CTS/RTS Separate Function (UART0)

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

- Set the CRD bit in the U0C0 register to "0" (enables UART0  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- Set the CRS bit in the U0C0 register to "1" (outputs UART0  $\overline{\text{RTS}}$ )
- Set the CRD bit in the U1C0 register to "0" (enables UART1  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- Set the CRS bit in the U1C0 register to "0" (inputs UART1  $\overline{\text{CTS}}$ )
- Set the RCSP bit in the UCON register to "1" (inputs  $\overline{\text{CTS}}_0$  from the P64 pin)
- Set the CLKMD1 bit in the UCON register to "0" (CLKS<sub>1</sub> not used)

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

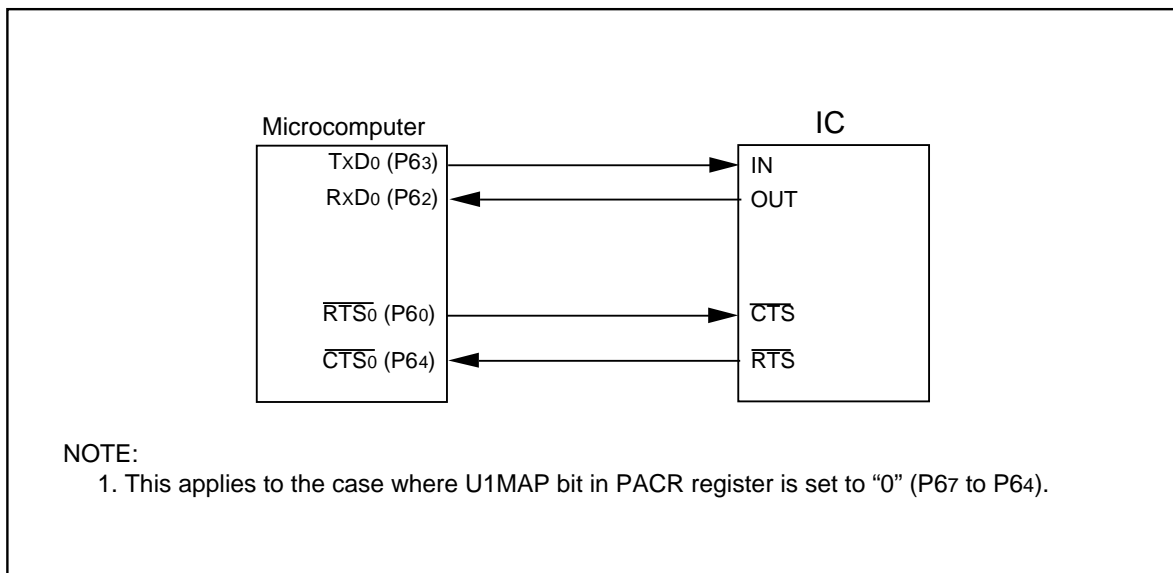


Figure 13.1.2.6.1. CTS/RTS Separate Function

### 13.1.3 Special Mode 1 (I<sup>2</sup>C bus mode)(UART2)

I<sup>2</sup>C bus mode is provided for use as a simplified I<sup>2</sup>C bus interface compatible mode. Table 13.1.3.1 lists the specifications of the I<sup>2</sup>C bus mode. Table 13.1.3.2 and 13.1.3.3 list the registers used in the I<sup>2</sup>C bus mode and the register values set. Table 13.1.3.4 lists the I<sup>2</sup>C bus mode functions. Figure 13.1.3.1 shows the block diagram for I<sup>2</sup>C bus mode. Figure 13.1.3.2 shows SCL2 timing.

As shown in Table 13.1.3.2, the microcomputer is placed in I<sup>2</sup>C bus mode by setting the SMD2 to SMD0 bits to '0102' and the IICM bit to "1". Because SDA2 transmit output has a delay circuit attached, SDA output does not change state until SCL2 goes low and remains stably low.

**Table 13.1.3.1. I<sup>2</sup>C bus Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>During master           <ul style="list-style-type: none"> <li>The CKDIR bit in the U2MR register is set to "0" (internal clock) : <math>f_j / (2(n+1))</math>  <math>f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math>. n: Setting value in the U2BRG register 00<sub>16</sub> to FF<sub>16</sub></li> </ul> </li> <li>During slave           <ul style="list-style-type: none"> <li>The CKDIR bit is set to "1" (external clock) : Input from SCL2 pin</li> </ul> </li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>Before transmission can start, the following requirements must be met <sup>(1)</sup> <ul style="list-style-type: none"> <li>The TE bit in the U2C1 register is set to "1" (transmission enabled)</li> <li>The TI bit in the U2C1 register is set to "0" (data present in U2TB register)</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>Before reception can start, the following requirements must be met <sup>(1)</sup> <ul style="list-style-type: none"> <li>The RE bit in the U2C1 register is set to "1" (reception enabled)</li> <li>The TE bit in the U2C1 register is set to "1" (transmission enabled)</li> <li>The TI bit in the U2C1 register is set to "0" (data present in the UiTB register)</li> </ul> </li> </ul>
Interrupt request generation timing	When start or stop condition is detected, acknowledge undetected, and acknowledge detected
Error detection	<ul style="list-style-type: none"> <li>Overrun error <sup>(2)</sup> <ul style="list-style-type: none"> <li>This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the 8th bit of the next data</li> </ul> </li> </ul>
Select function	<ul style="list-style-type: none"> <li>Arbitration lost           <ul style="list-style-type: none"> <li>Timing at which the ABT bit in the U2RB register is updated can be selected</li> </ul> </li> <li>SDA2 digital delay           <ul style="list-style-type: none"> <li>No digital delay or a delay of 2 to 8 U2BRG count source clock cycles selectable</li> </ul> </li> <li>Clock phase setting           <ul style="list-style-type: none"> <li>With or without clock delay selectable</li> </ul> </li> </ul>

**NOTES:**

- When an external clock is selected, the conditions must be met while the external clock is in the high state.
- If an overrun error occurs, bits 8 to 0 in UiRB register are undefined. The IR bit in the SiRIC register remains unchanged.

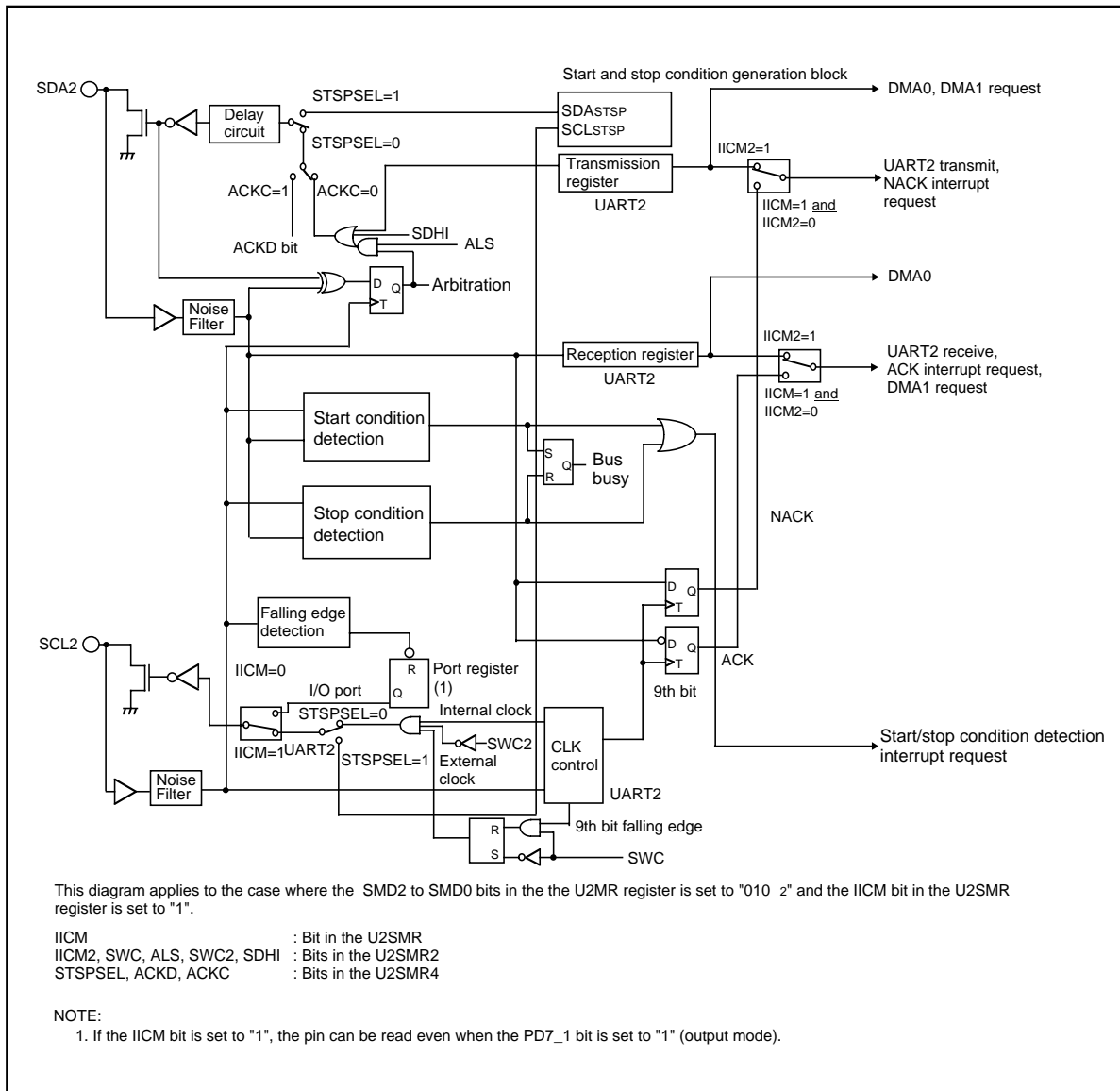


Figure 13.1.3.1. I²C bus Mode Block Diagram



**Table 13.1.3.2. Registers to Be Used and Settings in I<sup>2</sup>C bus Mode (1) (Continued)**

Register	Bit	Function	
		Master	Slave
U2TB (1)	0 to 7	Set transmission data	Set transmission data
U2RB (1)	0 to 7	Reception data can be read	Reception data can be read
	8	ACK or NACK is set in this bit	ACK or NACK is set in this bit
	ABT	Arbitration lost detection flag	Invalid
	OER	Overrun error flag	Overrun error flag
U2BRG	0 to 7	Set a transfer rate	Invalid
U2MR (1)	SMD2 to SMD0	Set to '0102'	Set to '0102'
	CKDIR	Set to "0"	Set to "1"
	IOPOL	Set to "0"	Set to "0"
U2C0	CLK1, CLK0	Select the count source for the U2BRG register	Invalid
	CRS	Invalid because CRD = 1	Invalid because CRD = 1
	TXEPT	Transmit buffer empty flag	Transmit buffer empty flag
	CRD	Set to "1"	Set to "1"
	NCH	Set to "1"	Set to "1"
	CKPOL	Set to "0"	Set to "0"
	UFORM	Set to "1"	Set to "1"
U2C1	TE	Set this bit to "1" to enable transmission	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception	Set this bit to "1" to enable reception
	RI	Reception complete flag	Reception complete flag
	U2IRS	Invalid	Invalid
	U2RRM, U2LCH, U2ERE	Set to "0"	Set to "0"
U2SMR	IICM	Set to "1"	Set to "1"
	ABC	Select the timing at which arbitration-lost is detected	Invalid
	BBS	Bus busy flag	Bus busy flag
	3 to 7	Set to "0"	Set to "0"
U2SMR2	IICM2	Refer to <b>Table 13.1.3.4 I<sup>2</sup>C bus Mode Functions</b>	Refer to <b>Table 13.1.3.4 I<sup>2</sup>C bus Mode Functions</b>
	CSC	Set this bit to "1" to enable clock synchronization	Set to "0"
	SWC	Set this bit to "1" to have SCL2 output fixed to "L" at the falling edge of the 9th bit of clock	Set this bit to "1" to have SCL2 output fixed to "L" at the falling edge of the 9 <sup>th</sup> bit of clock
	ALS	Set this bit to "1" to have SDA2 output stopped when arbitration-lost is detected	Set to "0"
	STAC	Set to "0"	Set this bit to "1" to initialize UART2 at start condition detection
	SWC2	Set this bit to "1" to have SCL2 output forcibly pulled low	Set this bit to "1" to have SCL2 output forcibly pulled low
	SDHI	Set this bit to "1" to disable SDA2 output	Set this bit to "1" to disable SDA2 output
U2SMR3	7	Set to "0"	Set to "0"
	0, 2, 4 and NODC	Set to "0"	Set to "0"
	CKPH	Refer to <b>Table 13.1.3.4 I<sup>2</sup>C bus Mode Functions</b>	Refer to <b>Table 13.1.3.4 I<sup>2</sup>C bus Mode Functions</b>
	DL2 to DL0	Set the amount of SDA2 digital delay	Set the amount of SDA2 digital delay

**NOTE:**

1. Not all register bits are described above. Set those bits to "0" when writing to the registers in I<sup>2</sup>C bus mode.

**Table 13.1.3.3. Registers to Be Used and Settings in I<sup>2</sup>C bus Mode <sup>(2)</sup> (Continued)**

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

## NOTE:

1. Not all bits in the register are described above. Set those bits to "0" when writing to the registers in I<sup>2</sup>C bus mode.

Table 13.1.3.4. I<sup>2</sup>C bus Mode Functions

Function	Clock synchronous serial I/O mode (SMD2 to SMD0 = 0012, IICM = 0)	I <sup>2</sup> C bus mode (SMD2 to SMD0 = 0102, IICM = 1)			
		IICM2 = 0 (NACK/ACK interrupt)		IICM2 = 1 (UART transmit/ receive interrupt)	
		CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)	CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)
Factor of interrupt number 10 <sup>(1)</sup> (Refer to Fig.13.1.3.2.)	—————	Start condition detection or stop condition detection (Refer to Figure 13.1.3.2.1. STSPSEL Bit Function)			
Factor of interrupt number 15 <sup>(1)</sup> (Refer to Fig.13.1.3.2.)	UART2 transmission Transmission started or completed (selected by U2IRS)	No acknowledgment detection (NACK) Rising edge of SCL2 9th bit	UART2 transmission Rising edge of SCL2 9th bit	UART2 transmission Falling edge of SCL2 next to the 9th bit	
Factor of interrupt number 16 <sup>(1)</sup> (Refer to Fig.13.1.3.2.)	UART2 reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) Rising edge of SCL2 9th bit	UART2 transmission Falling edge of SCL2 9th bit		
Timing for transferring data from the UART reception shift register to the U2RB register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Rising edge of SCL2 9th bit	Falling edge of SCL2 9th bit	Falling and rising edges of SCL2 9th bit	
UART2 transmission output delay	Not delayed	Delayed			
Functions of P70 pin	TxD2 output	SDA2 input/output			
Functions of P71 pin	RxD2 input	SCL2 input/output			
Functions of P72 pin	CLK2 input or output selected	————— (Cannot be used in I <sup>2</sup> C mode)			
Noise filter width	15ns	200ns			
Read RxD2 and SCL2 pin levels	Possible when the corresponding port direction bit = 0	Always possible no matter how the corresponding port direction bit is set			
Initial value of TxD2 and SDA2 outputs	CKPOL = 0 (H) CKPOL = 1 (L)	The value set in the port register before setting I <sup>2</sup> C bus mode <sup>(2)</sup>			
Initial and end values of SCL2	—————	H	L	H	L
DMA1 factor (Refer to Fig. 14.1.3.2.)	UART2 reception	Acknowledgment detection (ACK)	UART2 reception Falling edge of SCL2 9th bit		
Store received data	1st to 8th bits are stored in U2RB register bit 0 to bit 7	1st to 8th bits are stored in U2RB register bit 7 to bit 0	1st to 7th bits are stored in U2RB register bit 6 to bit 0, with 8th bit stored in U2RB register bit 8		
			1st to 8th bits are stored in U2RB register bit 7 to bit 0 <sup>(3)</sup>		
Read received data	U2RB register status is read directly as is				Read U2RB register Bit 6 to bit 0 as bit 7 to bit 1, and bit 8 as bit 0 <sup>(4)</sup>

## NOTES:

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

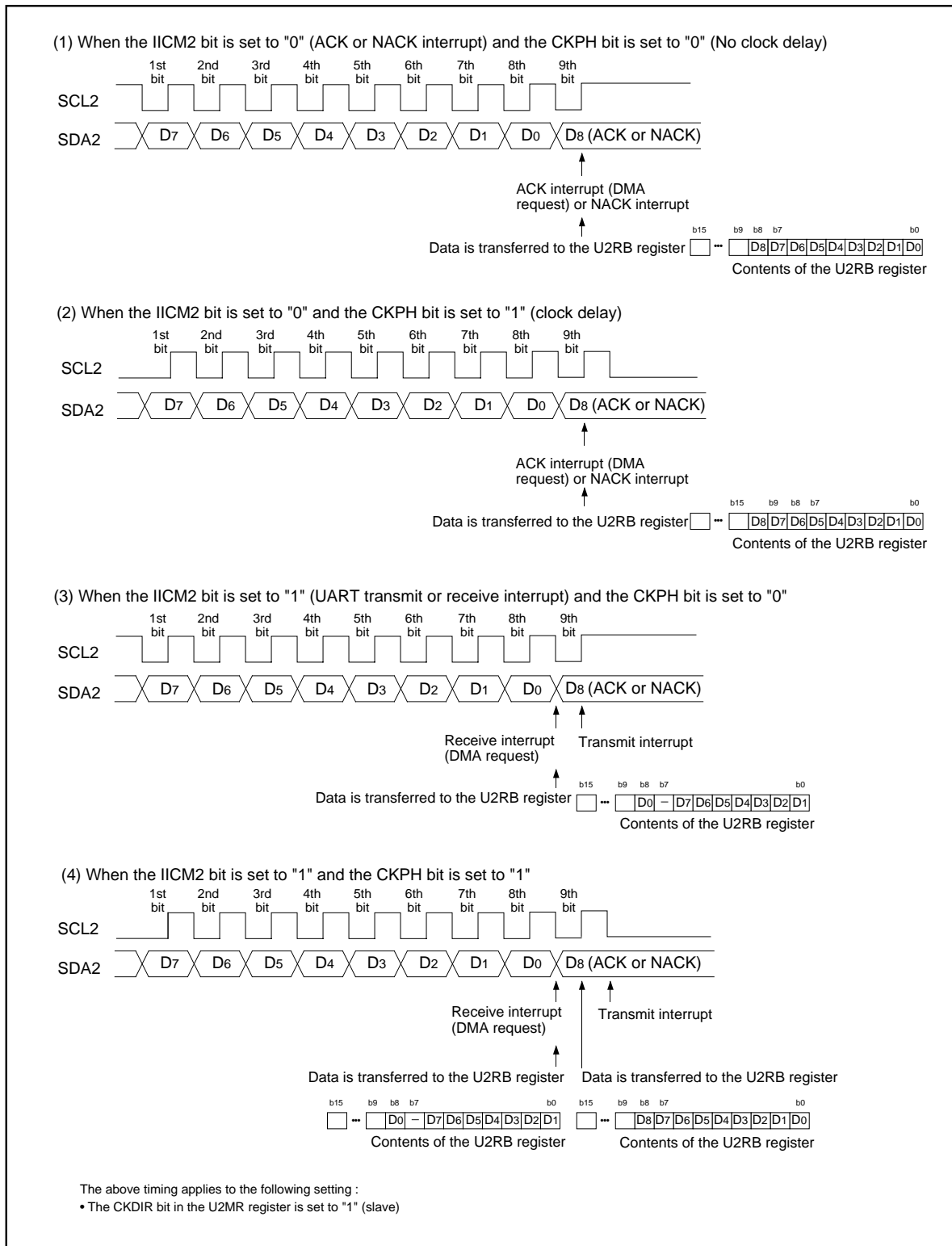


Figure 13.1.3.2. Transfer to U2RB Register and Interrupt Timing

### 13.1.3.1 Detection of Start and Stop Condition

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

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

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

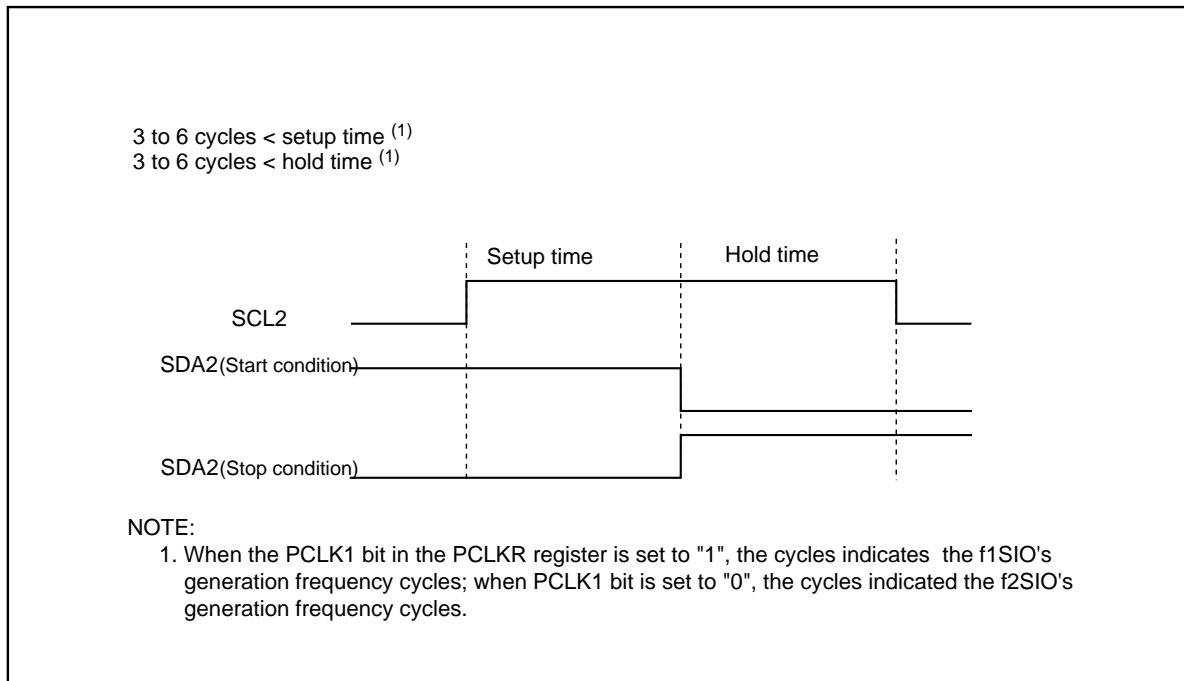


Figure 13.1.3.1.1. Detection of Start and Stop Condition

### 13.1.3.2 Output of Start and Stop Condition

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

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

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

The output procedure is described below.

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

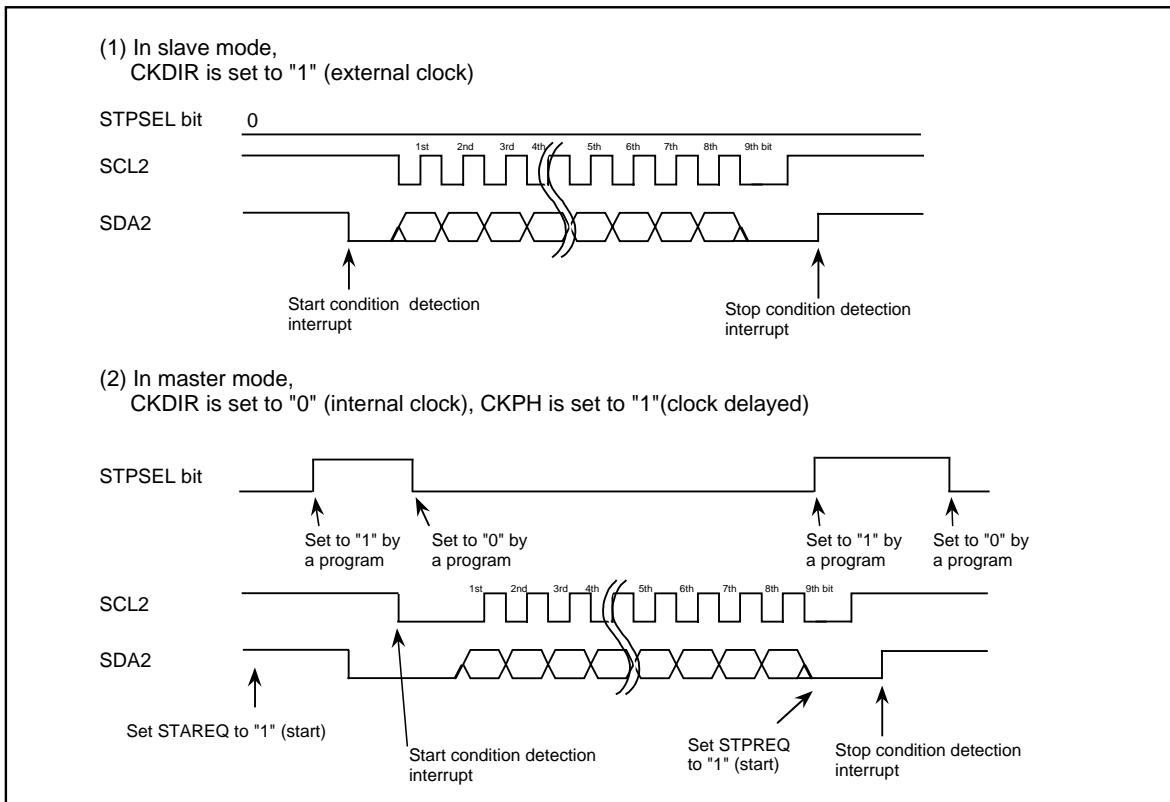
(2) Set the STSPSEL bit in the U2SMR4 register to "1" (output).

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

The function of the STSPSEL bit is shown in Table 13.1.3.2.1 and Figure 13.1.3.2.1.

**Table 13.1.3.2.1. STSPSEL Bit Functions**

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

**Figure 13.1.3.2.1. STSPSEL Bit Functions****13.1.3.3 Arbitration**

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

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

#### 13.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 13.1.3.2.1.

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

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

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

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

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

#### 13.1.3.5 SDA Output

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

The initial value of SDA2 transmit output can only be set when IICM is set to "1" (I<sup>2</sup>C Bus mode) and the SMD2 to SMD0 bits in the U2MR register are set to '0002' (serial I/O disabled).

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

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

#### 13.1.3.6 SDA Input

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

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

### 13.1.3.7 ACK and NACK

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

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

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

### 13.1.3.8 Initialization of Transmission/Reception

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

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

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



### 13.1.4 Special Mode 2 (UART2)

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. Table 13.1.4.1 lists the specifications of Special Mode 2. Table 13.1.4.2 lists the registers used in Special Mode 2 and the register values set. Figure 13.1.4.1 shows communication control example for Special Mode 2.

**Table 13.1.4.1. Special Mode 2 Specifications**

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

**NOTES:**

- When an external clock is selected, the conditions must be met while if the CKPOL bit in the U2C0 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 U2C0 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.
- If an overrun error occurs, bits 8 to 0 in U2RB register are undefined. The IR bit in the SiRIC register remains unchanged.

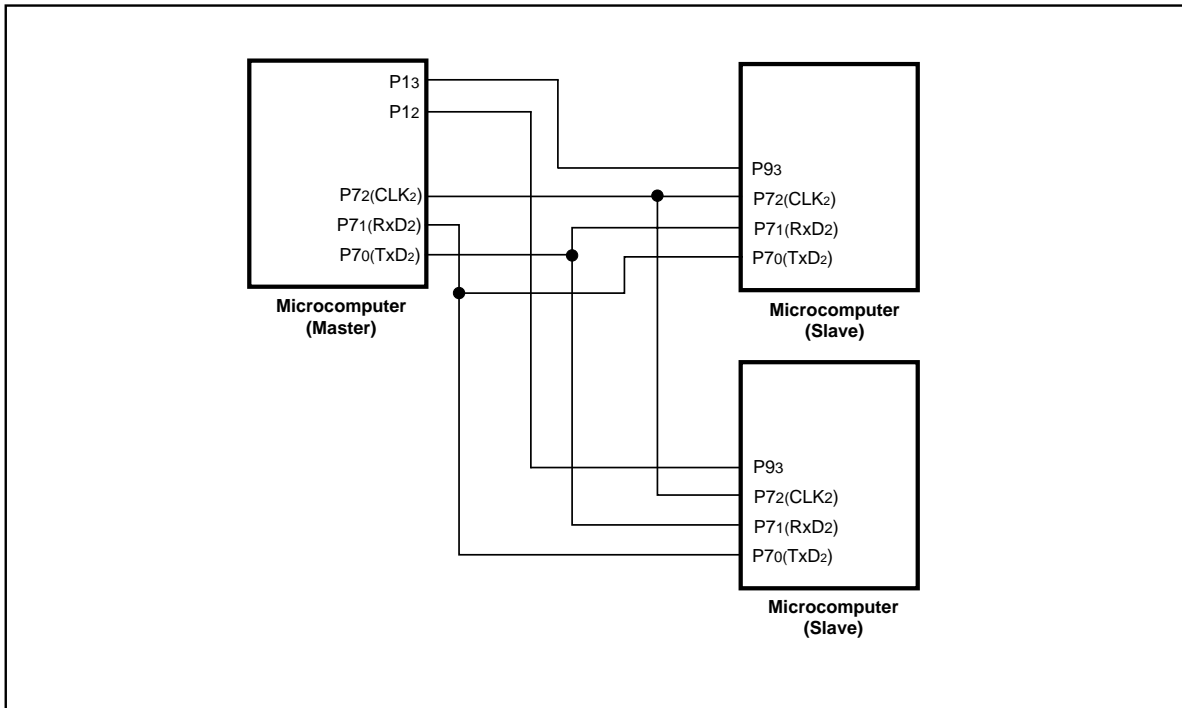


Figure 13.1.4.1. Serial Bus Communication Control Example (UART2)

**Table 13.1.4.2. Registers to Be Used and Settings in Special Mode 2**

Register	Bit	Function
U2TB <sup>(1)</sup>	0 to 7	Set transmission data
U2RB <sup>(1)</sup>	0 to 7	Reception data can be read
	OER	Overrun error flag
U2BRG	0 to 7	Set a transfer rate
U2MR <sup>(1)</sup>	SMD2 to SMD0	Set to '0012'
	CKDIR	Set this bit to "0" for master mode or "1" for slave mode
	IOPOL	Set to "0"
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRD = 1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Select TxD2 pin output format
	CKPOL	Clock phases can be set in combination with the CKPH bit in the U2SMR3 register
	UFORM	Set to "0"
U2C1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS	Select UART2 transmit interrupt cause
	U2RRM, U2LCH, U2ERE	Set to "0"
U2SMR	0 to 7	Set to "0"
U2SMR2	0 to 7	Set to "0"
U2SMR3	CKPH	Clock phases can be set in combination with the CKPOL bit in the U2C0 register
	NODC	Set to "0"
	0, 2, 4 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"

**NOTE:**

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

### 13.1.4.1 Clock Phase Setting Function

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

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

#### 13.1.4.1.1 Master (Internal Clock)

Figure 13.1.4.1.1.1 shows the transmission and reception timing in master (internal clock).

#### 13.1.4.1.2 Slave (External Clock)

Figure 13.1.4.1.2.1 shows the transmission and reception timing (CKPH=0) in slave (external clock) while Figure 13.1.4.1.2.2 shows the transmission and reception timing (CKPH=1) in slave (external clock).

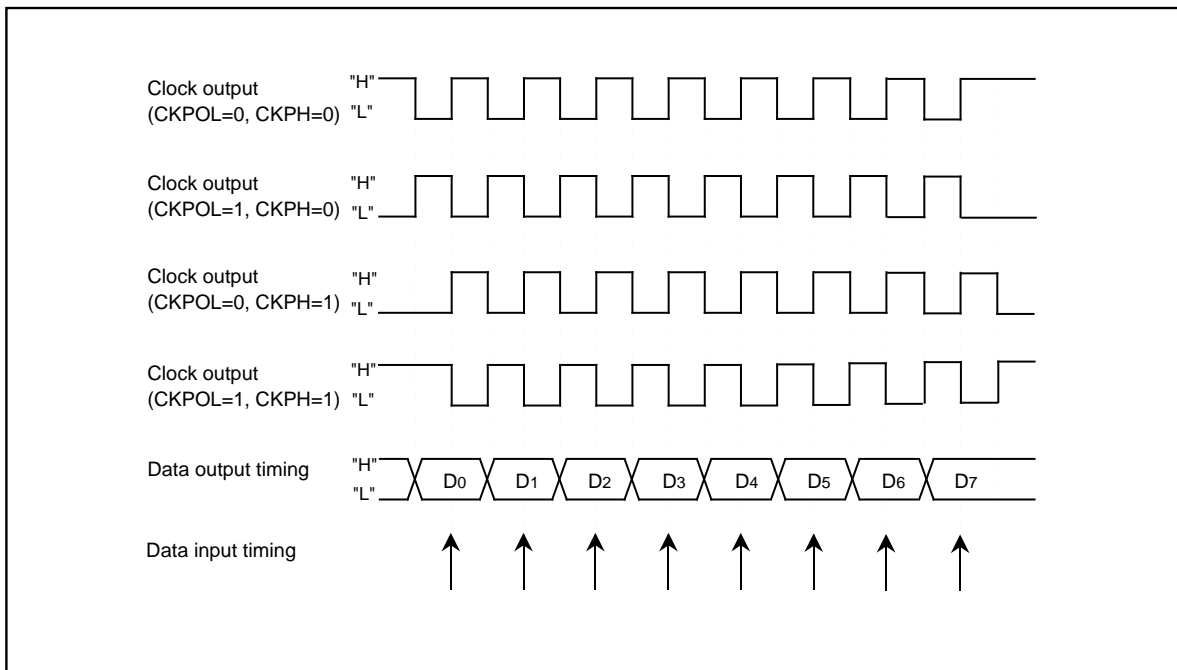


Figure 13.1.4.1.1.1. Transmission and Reception Timing in Master Mode (Internal Clock)

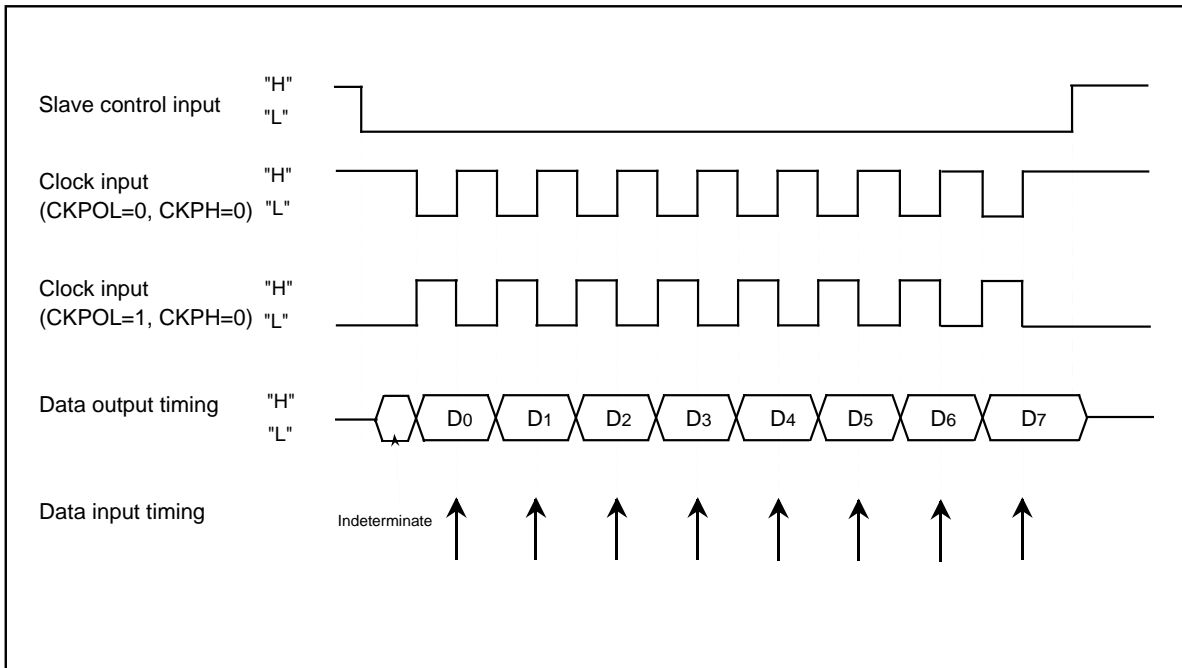


Figure 13.1.4.1.2.1. Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)

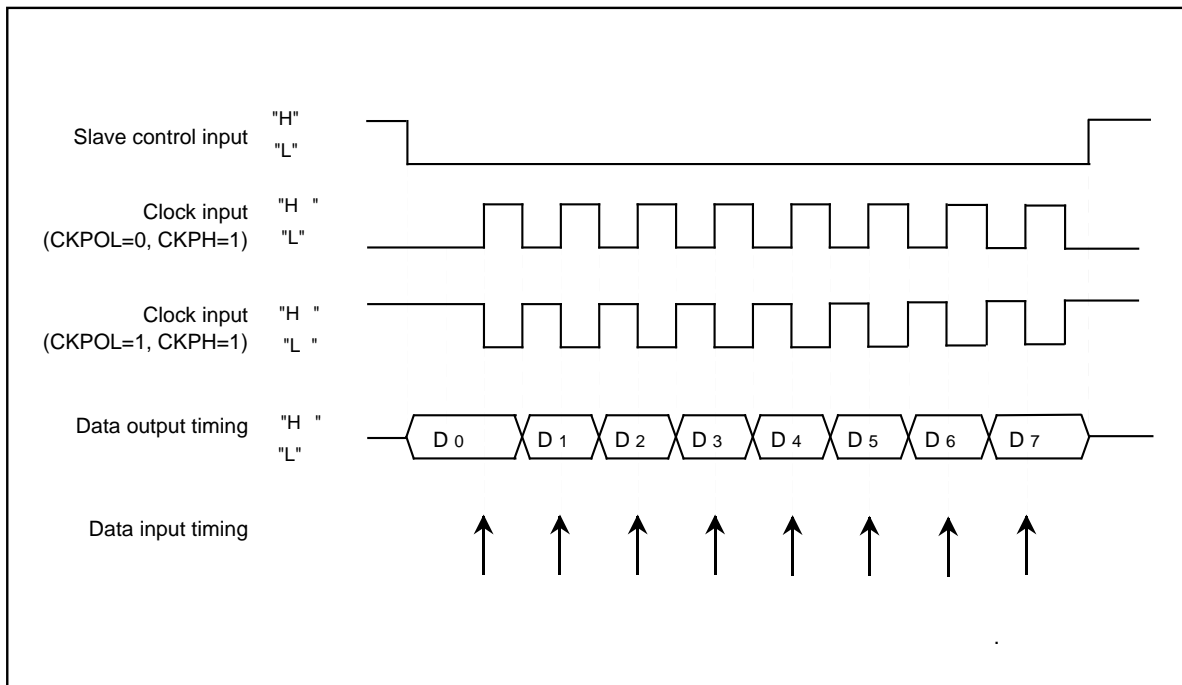


Figure 13.1.4.1.2.2. Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)

### 13.1.5 Special Mode 3 (IE Bus mode )(UART2)

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

Table 13.1.5.1 lists the registers used in IE Bus mode and the register values set. Figure 13.1.5.1 shows the functions of bus collision detect function related bits.

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

**Table 13.1.5.1. Registers to Be Used and Settings in IE Bus Mode**

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

**NOTE:**

1. Not all bits in the registers are described above. Set those bits to "0" when writing to the registers in IE Bus mode.

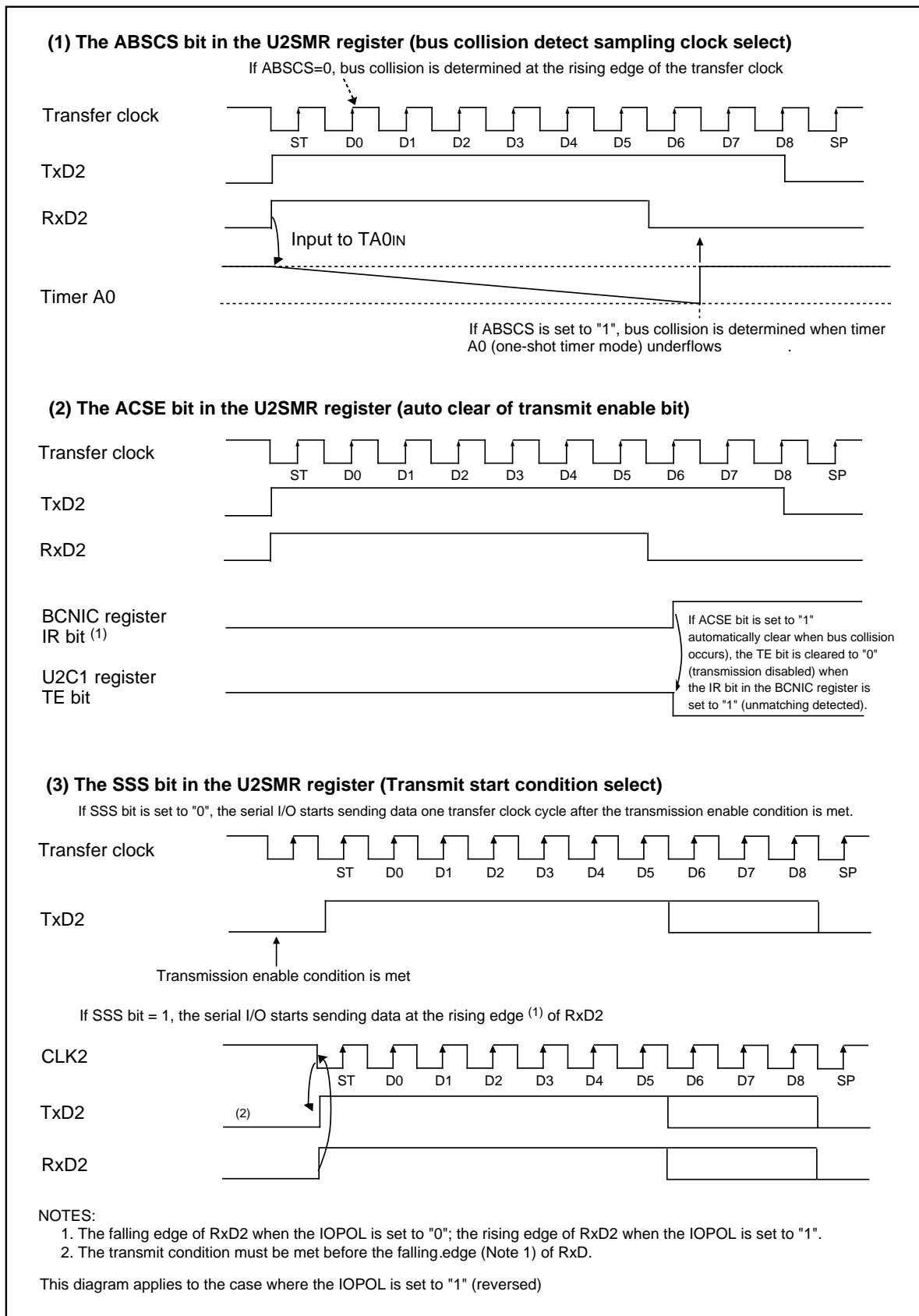


Figure 13.1.5.1. Bus Collision Detect Function-Related Bits

### 13.1.6 Special Mode 4 (SIM Mode) (UART2)

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

Tables 13.1.6.1 lists the specifications of SIM mode. Table 13.1.6.2 lists the registers used in the SIM mode and the register values set.

**Table 13.1.6.1. SIM Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Direct format</li> <li>• Inverse format</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• The CKDIR bit in the U2MR register is set to "0" (internal clock) : <math>f_i/(16(n+1))</math>  <math>f_i = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math>. n: Setting value in U2BRG register 00<sub>16</sub> to FF<sub>16</sub></li> <li>• The CKDIR bit is set to "1" (external clock) : <math>f_{EXT}/(16(n+1))</math>  <math>f_{EXT}</math>: Input from CLK2 pin. n: Setting value in U2BRG register 00<sub>16</sub> to FF<sub>16</sub></li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>• Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> <li>– The TE bit in the U2C1 register is set to "1" (transmission enabled)</li> <li>– The TI bit in the U2C1 register is set to "0" (data present in U2TB register)</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>• Before reception can start, the following requirements must be met <ul style="list-style-type: none"> <li>– The RE bit in the U2C1 register is set to "1" (reception enabled)</li> <li>– Start bit detection</li> </ul> </li> </ul>
Interrupt request generation timing (2)	<ul style="list-style-type: none"> <li>• For transmission When the serial I/O finished sending data from the U2TB transfer register (the U2IRS bit is set to "1")</li> <li>• For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception)</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error <sup>(1)</sup> This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data</li> <li>• Framing error This error occurs when the number of stop bits set is not detected</li> <li>• Parity error During reception, if a parity error is detected, parity error signal is output from the TxD2 pin. During transmission, a parity error is detected by the level of input to the RxD2 pin when a transmission interrupt occurs</li> <li>• Error sum flag This flag is set to "1" when any of the overrun, framing, and parity errors is encountered</li> </ul>

**NOTES:**

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



**Table 13.1.6.2. Registers to Be Used and Settings in SIM Mode**

Register	Bit	Function
U2TB <sup>(1)</sup>	0 to 7	Set transmission data
U2RB <sup>(1)</sup>	0 to 7	Reception data can be read
	OER, FER, PER, SUM	Error flag
U2BRG	0 to 7	Set a transfer rate
U2MR	SMD2 to SMD0	Set to '1012'
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Set this bit to "1" for direct format or "0" for inverse format
	PRYE	Set to "1"
	IOPOL	Set to "0"
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRD=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 <sup>(1)</sup>	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 bits in registers are described above. Set those bits to "0" when writing to the registers in SIM mode.

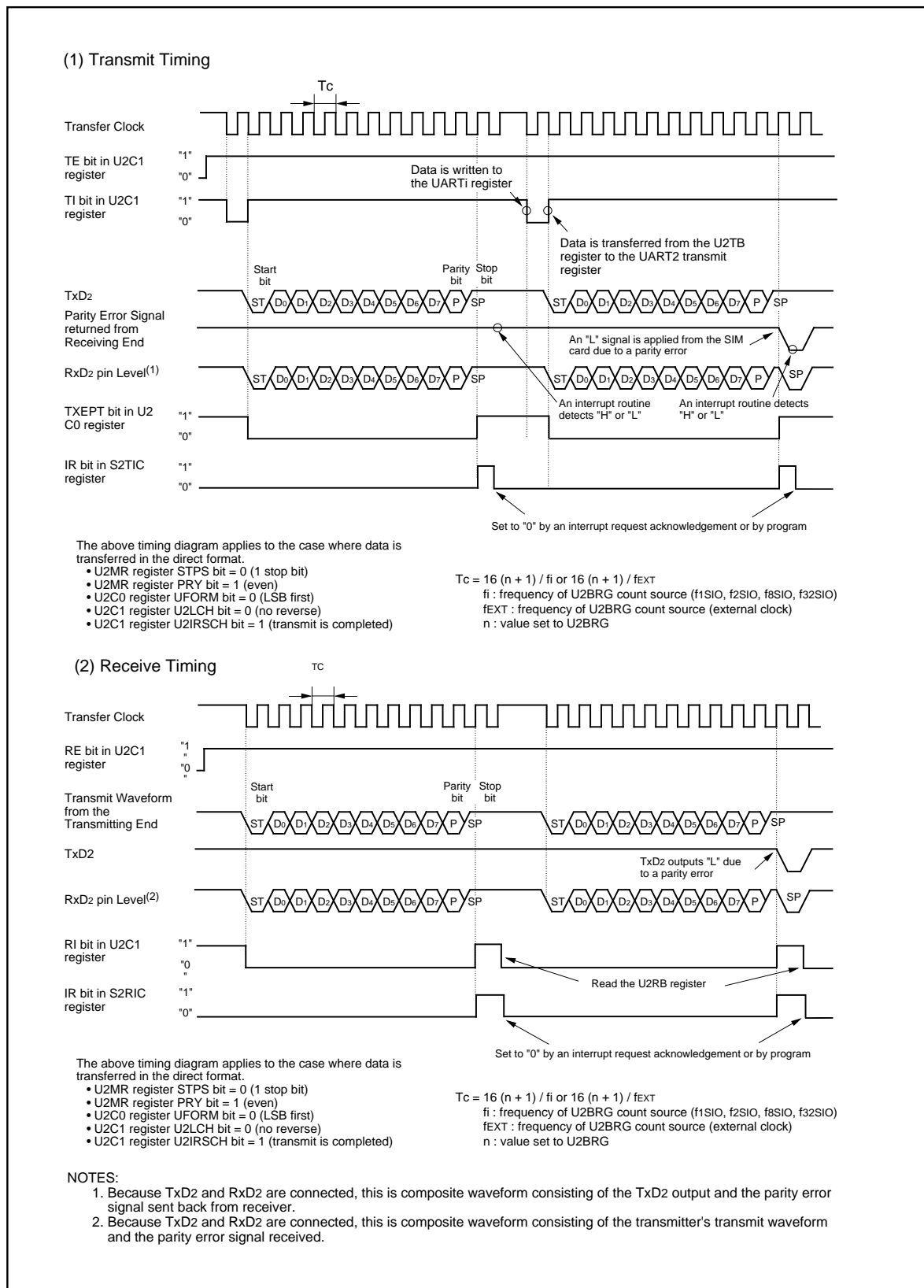


Figure 13.1.6.1. Transmit and Receive Timing in SIM Mode

Figure 13.1.6.2 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.

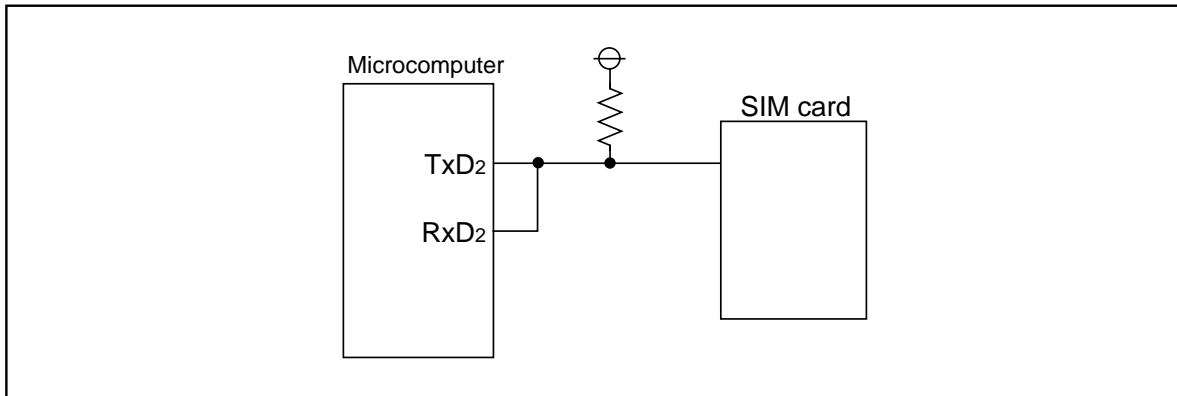


Figure 13.1.6.2. SIM Interface Connection

### 13.1.6.1 Parity Error Signal Output

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

- When receiving

The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TxD2 output low with the timing shown in Figure 13.1.6.1.1. If the R2RB register is read while outputting a parity error signal, the PER bit is cleared to "0" and at the same time the TxD2 output is returned high.

- When transmitting

A transmission-finished interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity signal has been returned can be determined by reading the port that shares the RxD2 pin in a transmission-finished interrupt service routine.

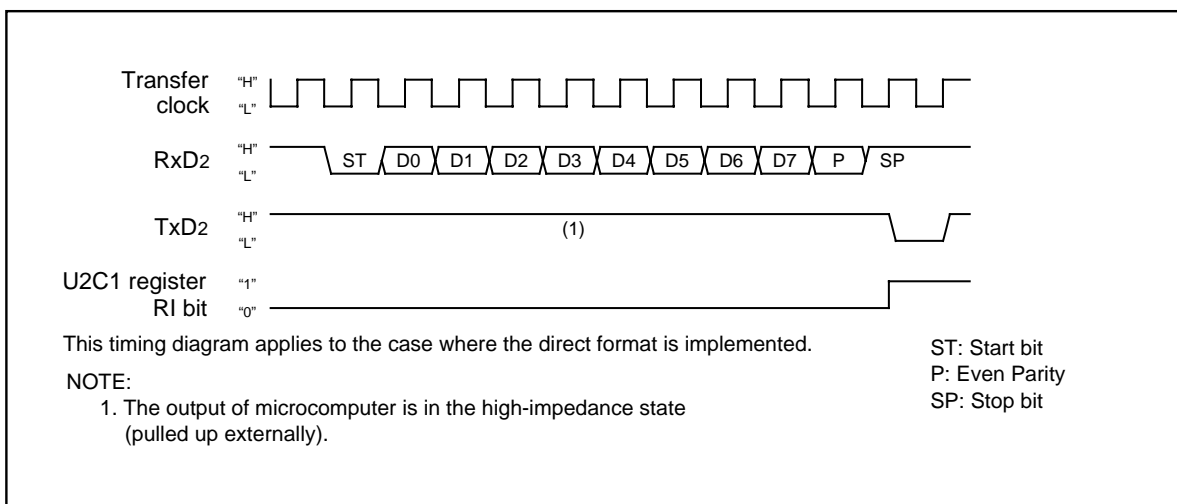


Figure 13.1.6.1.1. Parity Error Signal Output Timing

**13.1.6.2 Format**

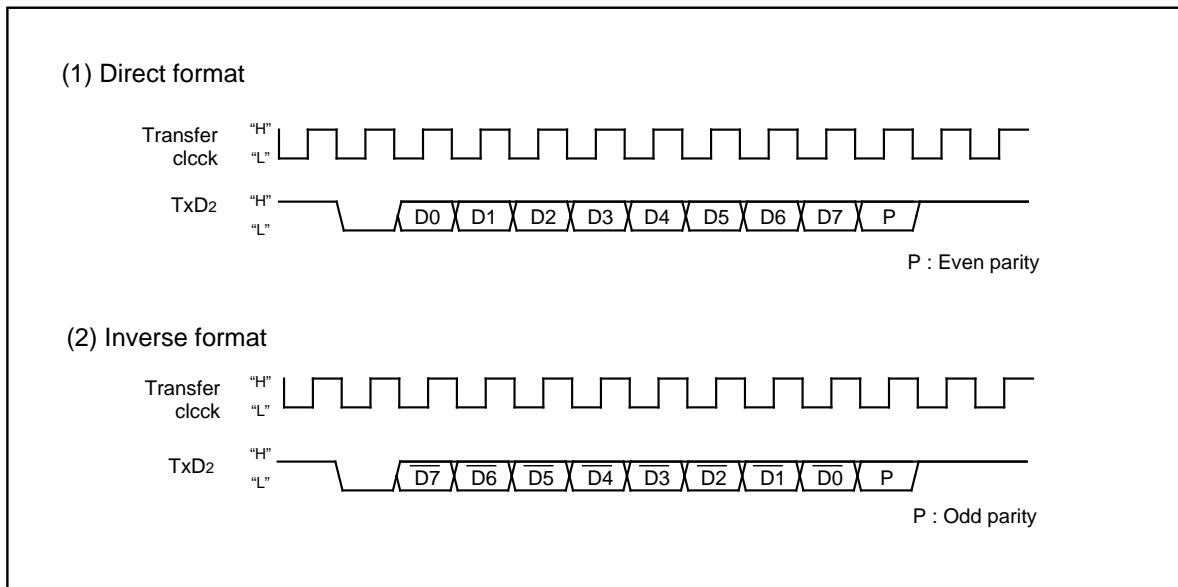
## • Direct Format

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

## • Inverse Format

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

Figure 13.1.6.2.1 shows the SIM interface format.



**Figure 13.1.6.2.1. SIM Interface Format**

## 14. A/D Converter

### Note

P92 and P93 (AN32, AN24) are not available in the 42-pin package.  
Do not use P92 and P93 (AN32, AN24) as analog input pins in the 42-pin package.

The microcomputer contains one A/D converter circuit based on 10-bit successive approximation method configured with a capacitive-coupling amplifier. The analog inputs share the pins with P100 to P107 (AN0 to AN7), P90 to P93 (AN30 to AN32, AN24). Similarly,  $\overline{ADTRG}$  input shares the pin with P15. Therefore, when using these inputs, make sure the corresponding port direction bits are set to "0" (input mode).

When not using the A/D converter, set the VCUT bit to "0" (VREF unconnected), so that no current will flow from the VREF pin into the resistor ladder, helping to reduce the power consumption of the chip.

The A/D conversion result is stored in the  $i$  bits in the A/D register for AN $_i$ , AN $_{3i}$ , and AN $_{2i}$  pins ( $i = 0$  to 7). Table 14.1 shows the A/D converter performance. Figure 14.1 shows the A/D converter block diagram and Figures 14.2 to 14.4 show the A/D converter associated with registers.

**Table 14.1 A/D Converter Performance**

Item	Performance
A/D Conversion Method	Successive approximation (capacitive coupling amplifier)
Analog Input Voltage <sup>(1)</sup>	0V to AVCC (VCC)
Operating Clock fAD <sup>(2)</sup>	fAD/divided-by-2 or fAD/divided-by-3 or fAD/divided-by-4 or fAD/divided-by-6 or fAD/divided-by-12 or fAD
Resolution	8-bit or 10-bit (selectable)
Integral Nonlinearity Error	When AVCC = VREF = 5V <ul style="list-style-type: none"> <li>• With 8-bit resolution: <math>\pm 2</math>LSB</li> <li>• With 10-bit resolution: <math>\pm 3</math>LSB</li> </ul> When AVCC = VREF = 3.3V <ul style="list-style-type: none"> <li>• With 8-bit resolution: <math>\pm 2</math>LSB</li> <li>• With 10-bit resolution: <math>\pm 5</math>LSB</li> </ul>
Operating Modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, repeat sweep mode 1, simultaneous sample sweep mode and delayed trigger mode 0,1
Analog Input Pins <sup>(3)</sup>	8 pins (AN0 to AN7) + 3 pins (AN30 to AN32) + 1 pins (AN24) (48-pin package) 8 pins (AN0 to AN7) + 2 pins (AN30, AN31) (42-pin package)
Conversion Speed Per Pin	<ul style="list-style-type: none"> <li>• Without sample and hold function 8-bit resolution: 49 fAD cycles, 10-bit resolution: 59 fAD cycles</li> <li>• With sample and hold function 8-bit resolution: 28 fAD cycles, 10-bit resolution: 33 fAD cycles</li> </ul>

### NOTES:

1. Not dependent on use of sample and hold function.
2. Set the  $\phi$ AD frequency to 10 MHz or less. For M16C/26B, set it to 12 MHz or less.  
Without sample-and-hold function, set the fAD frequency to 250kHz or more.  
With the sample and hold function, set the fAD frequency to 1MHz or more.

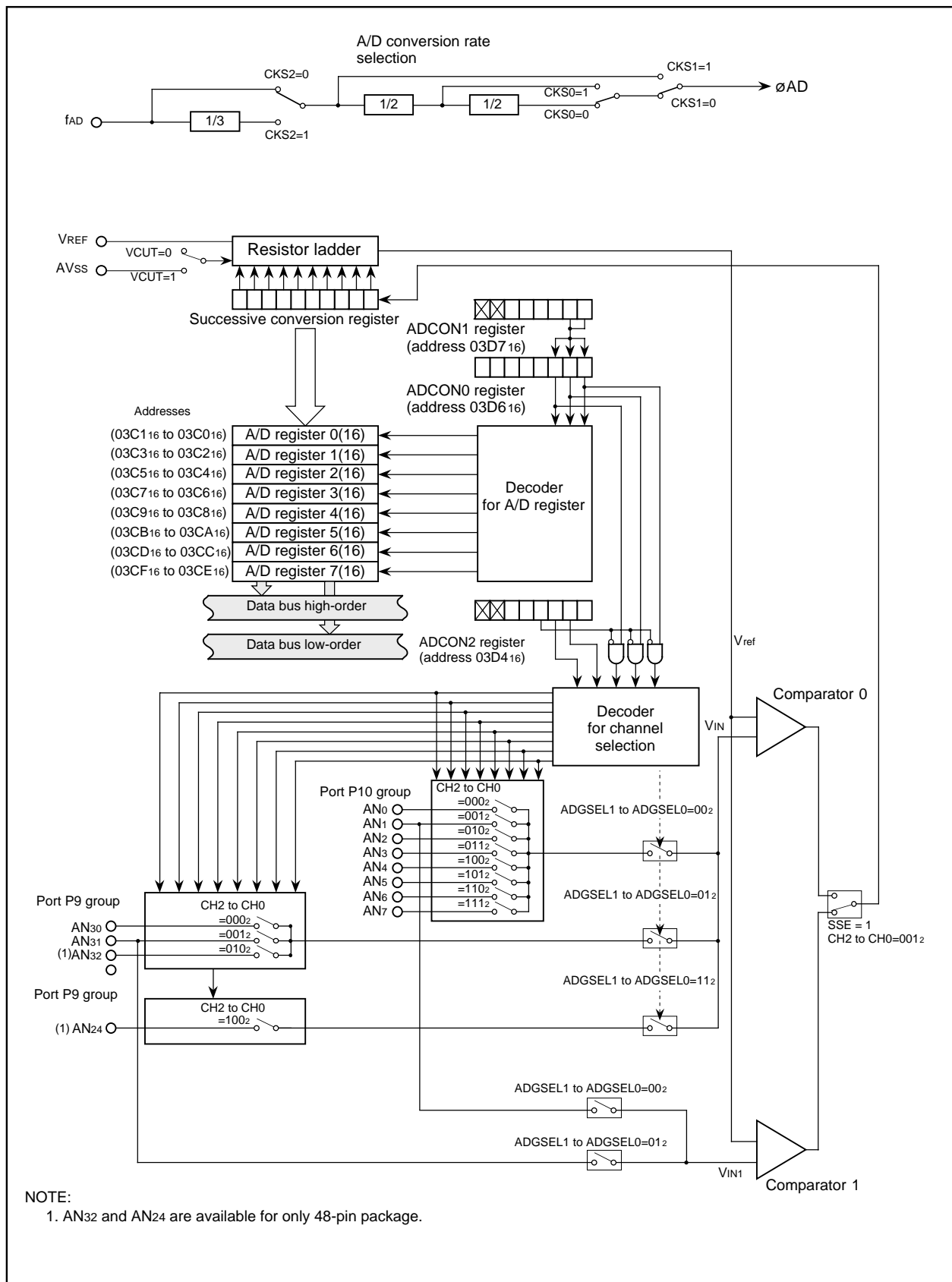


Figure 14.1 A/D Converter Block Diagram

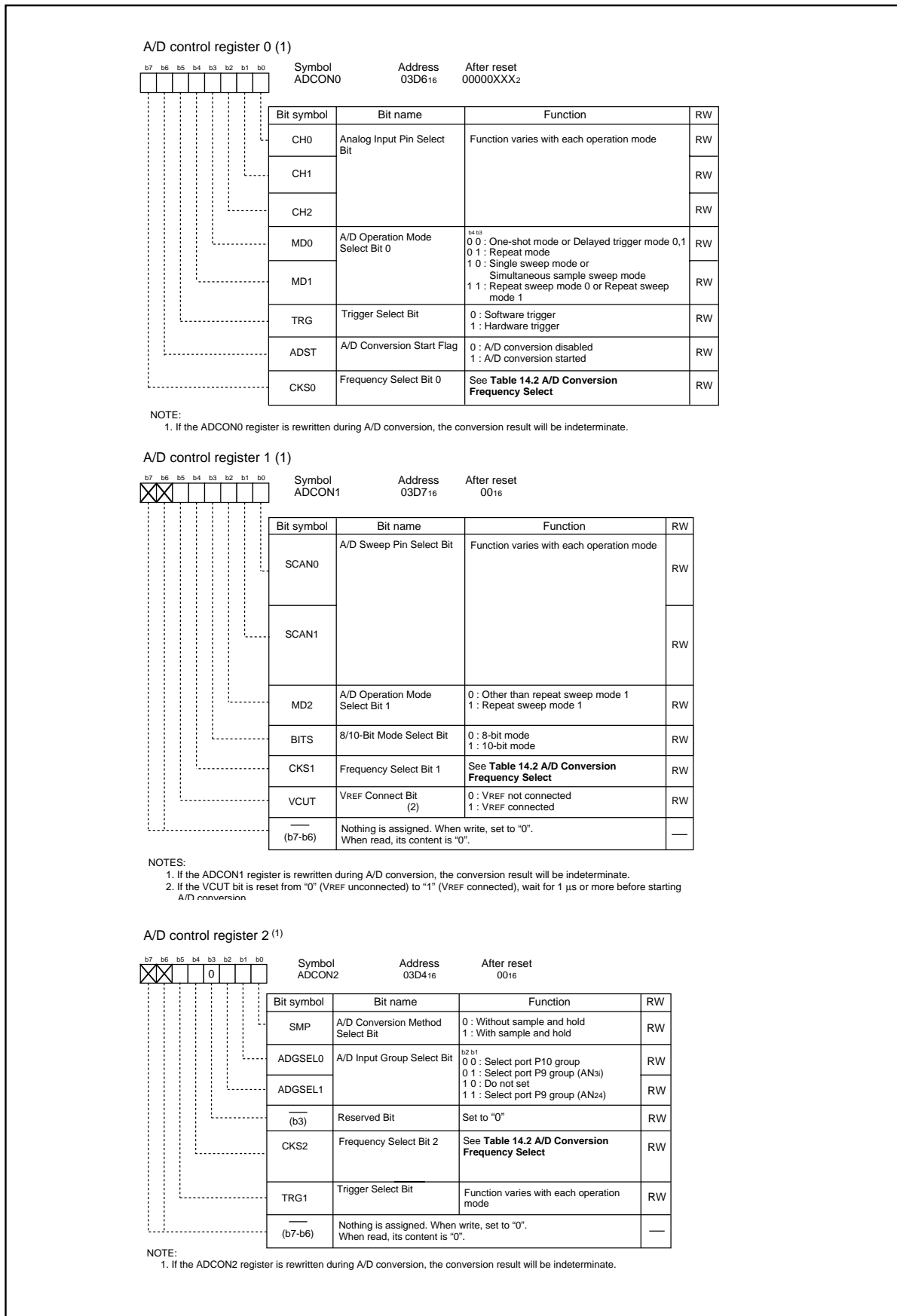


Figure 14.2 ADCON0 to ADCON2 Registers

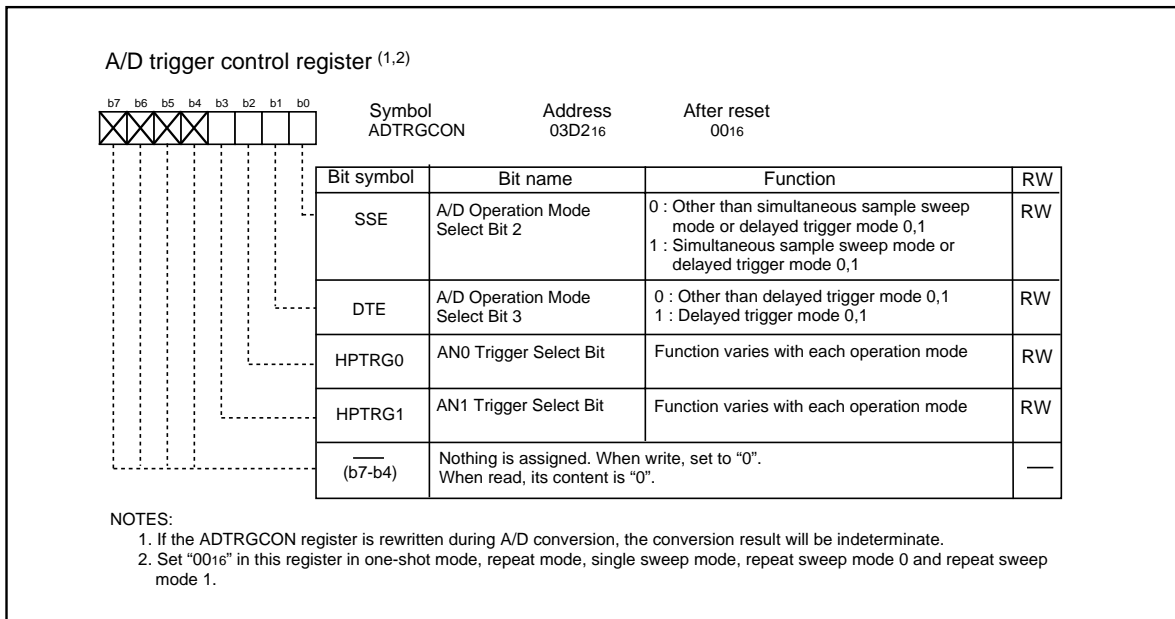


Figure 14.3 ADTRGCON Register

Table 14.2 A/D Conversion Frequency Select

CKS2	CKS1	CKS0	$\phi_{AD}$
0	0	0	Divided-by-4 of $f_{AD}$
0	0	1	Divided-by-2 of $f_{AD}$
0	1	0	$f_{AD}$
0	1	1	
1	0	0	Divided-by-12 of $f_{AD}$
1	0	1	Divided-by-6 of $f_{AD}$
1	1	0	Divided-by-3 of $f_{AD}$
1	1	1	

## NOTE:

- Set the  $\phi_{AD}$  frequency to 10 MHz or less (12 MHz or less in M16C/26B). The  $\phi_{AD}$  is selected with combinations of the CKS0 bit in the ADCON0 register, CKS1 bit in the ADCON1 register, and the CKS2 bit in the ADCON2 register.



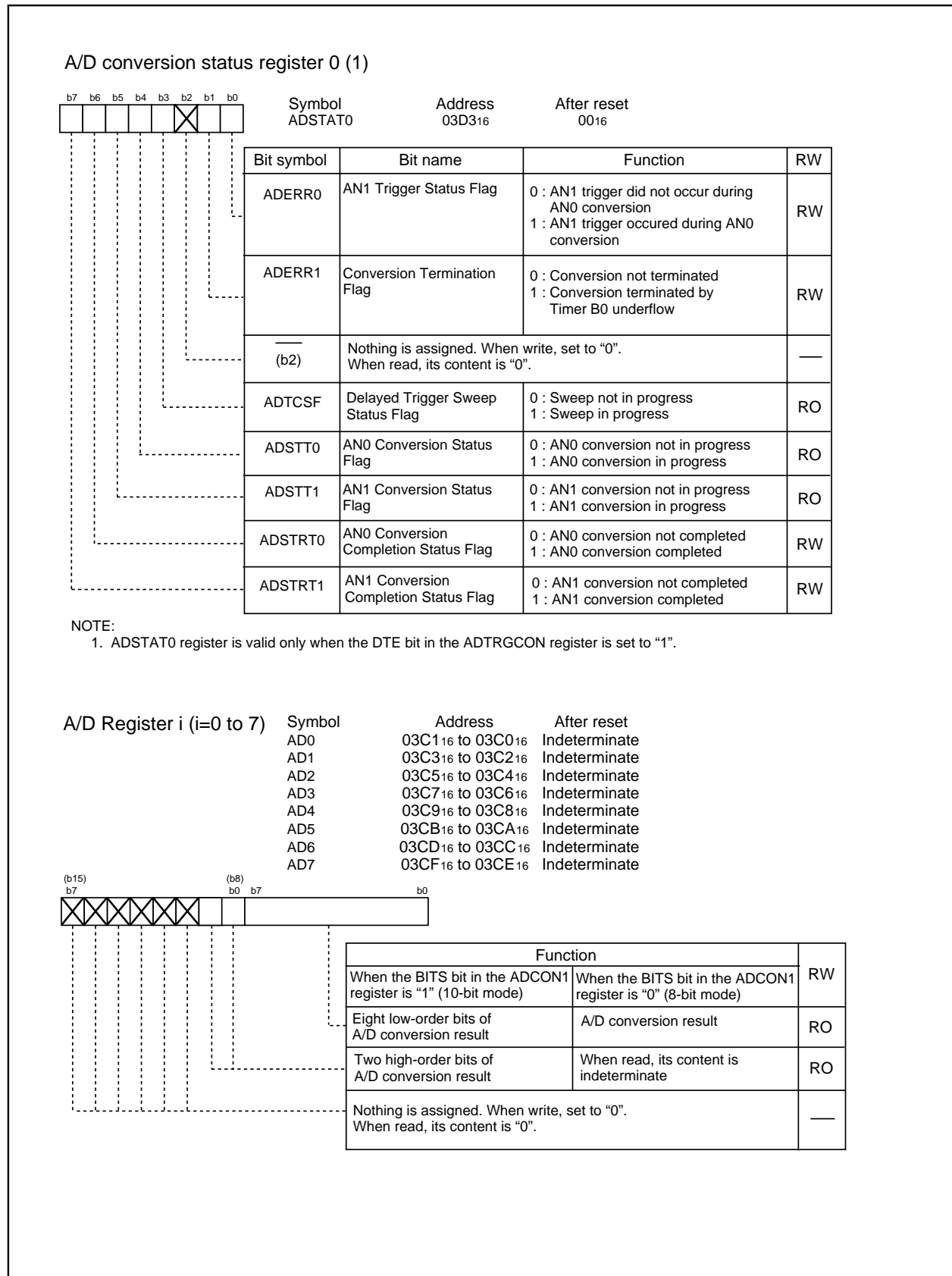


Figure 14.4 ADSTAT0 Register and AD0 to AD7 Registers

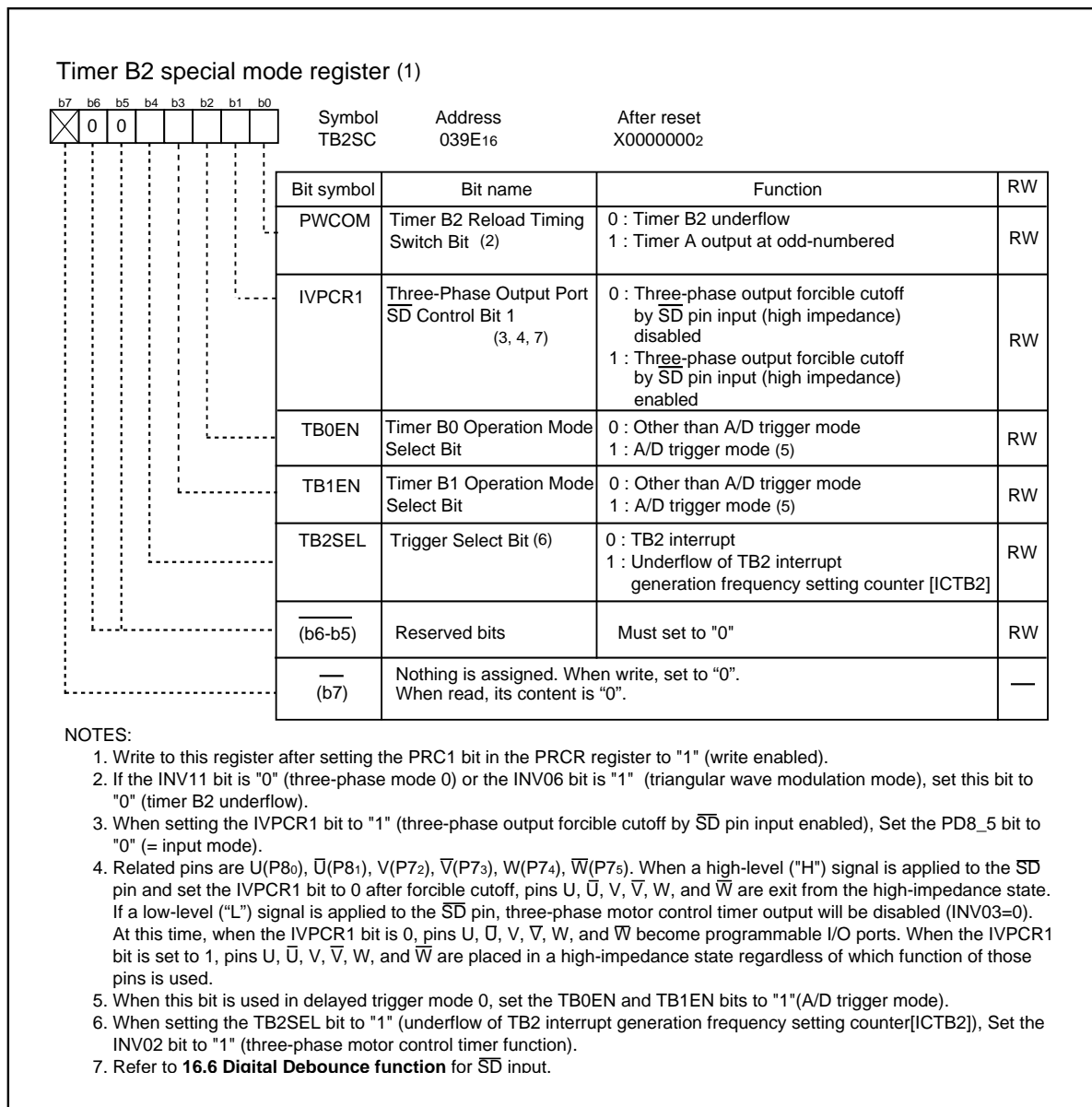


Figure 14.5 TB2SC Register

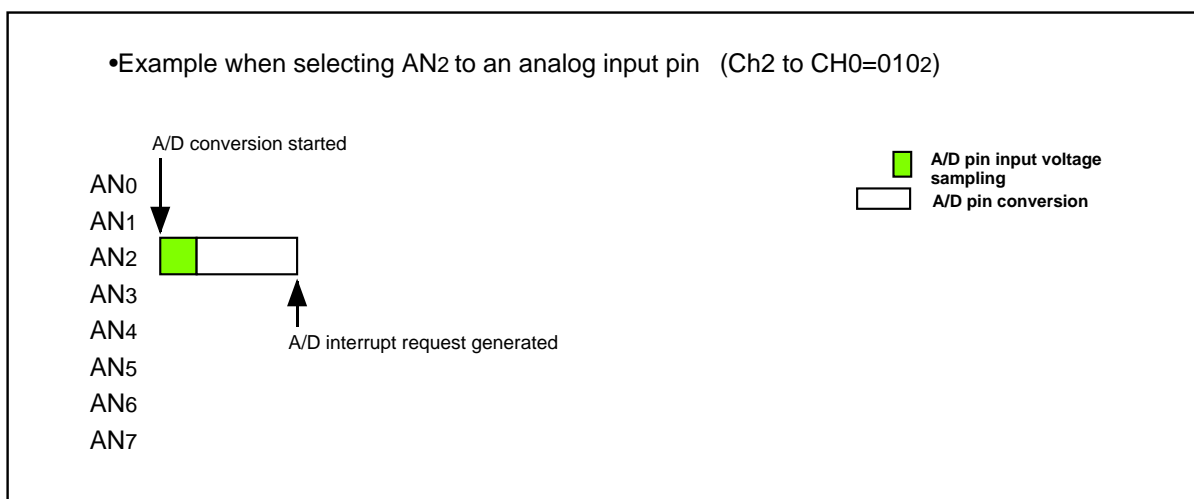
## 14.1 Operation Modes

### 14.1.1 One-Shot Mode

In one-shot mode, analog voltage applied to a selected pin is once converted to a digital code. Table 14.1.1.1 shows the one-shot mode specifications. Figure 14.1.1.1 shows the operation example in one-shot mode. Figure 14.1.1.2 shows the ADCON0 to ADCON2 registers in one-shot mode.

**Table 14.1.1.1 One-shot Mode Specifications**

Item	Specification
Function	The CH2 to CH0 bits in the ADCON0 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to a selected pin is once converted to a digital code
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>When the TRG bit in the ADCON0 register is "0" (software trigger) Set the ADST bit in the ADCON0 register to "1" (A/D conversion started)</li> <li>When the TRG bit in the ADCON0 register is "1" (hardware trigger) The <math>\overline{\text{ADTRG}}</math> pin input changes state from "H" to "L" after setting the ADST bit to "1" (A/D conversion started)</li> </ul>
A/D Conversion Stop Condition	<ul style="list-style-type: none"> <li>A/D conversion completed (If a software trigger is selected, the ADST bit is set to "0" (A/D conversion halted)).</li> <li>Set the ADST bit to "0"</li> </ul>
Interrupt Request Generation Timing	A/D conversion completed
Analog Input Pin	Select one pin from AN0 to AN7, AN30 to AN32, AN24
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin



**Figure 14.1.1.1 Operation Example in One-Shot Mode**

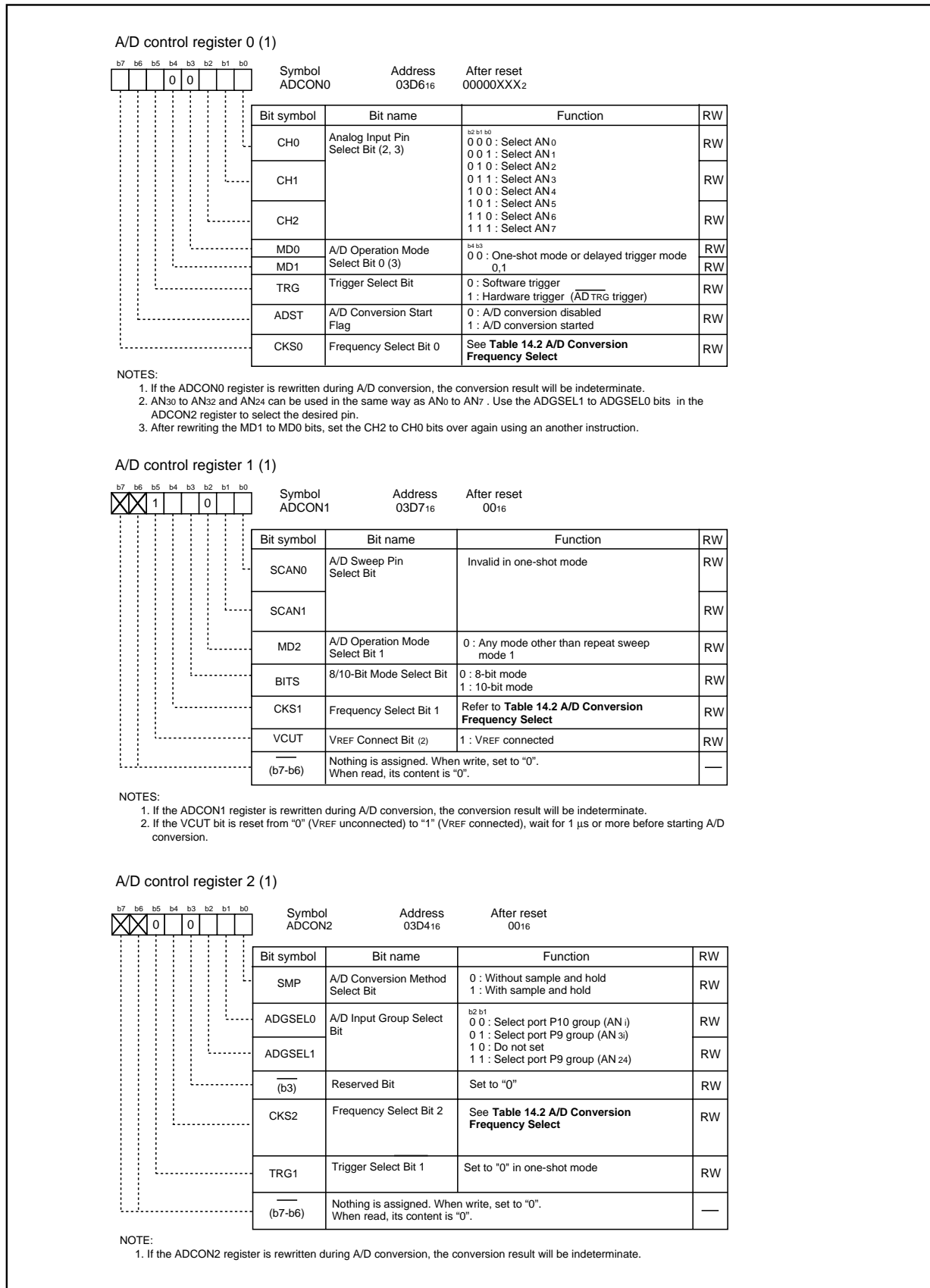


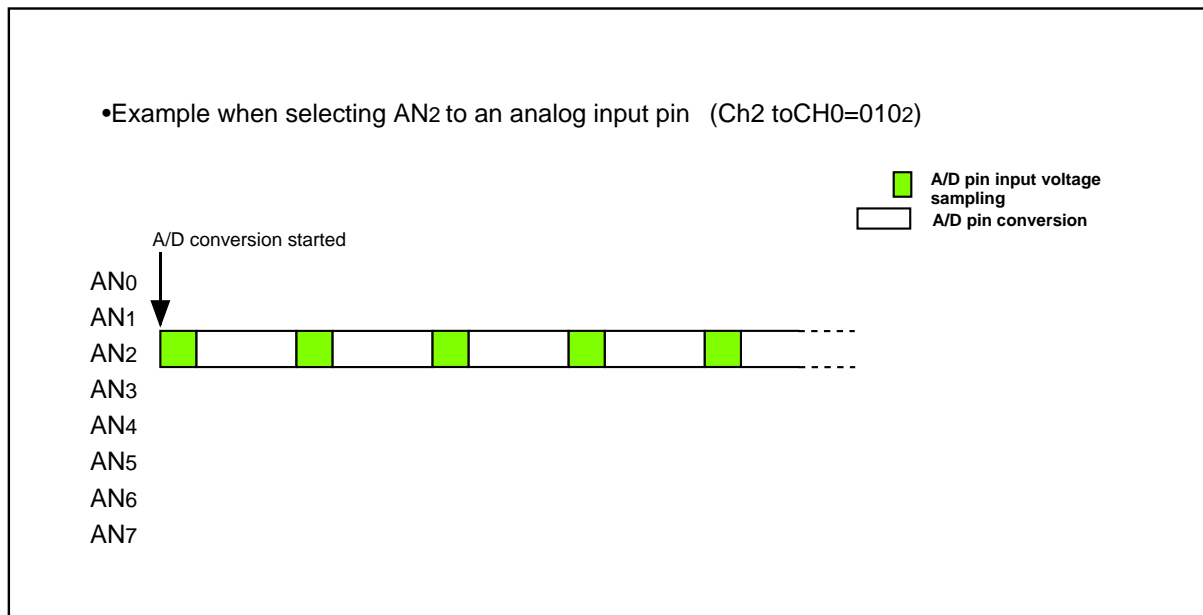
Figure 14.1.1.2 ADCON0 to ADCON2 Registers in One-Shot Mode

### 14.1.2 Repeat mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code. Table 14.1.2.1 shows the repeat mode specifications. Figure 14.1.2.1 shows the operation example in repeat mode. Figure 14.1.2.2 shows the ADCON0 to ADCON2 registers in repeat mode.

**Table 14.1.2.1 Repeat Mode Specifications**

Item	Specification
Function	The CH2 to CH0 bits in the ADCON0 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to a selected pin is repeatedly converted to a digital code
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>When the TRG bit in the ADCON0 register is "0" (software trigger) Set the ADST bit in the ADCON0 register to "1" (A/D conversion started)</li> <li>When the TRG bit in the ADCON0 register is "1" (hardware trigger) The ADTRG pin input changes state from "H" to "L" after setting the ADST bit to "1" (A/D conversion started)</li> </ul>
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, AN30 to AN32 and AN24
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin



**Figure 14.1.2.1 Operation Example in Repeat Mode**

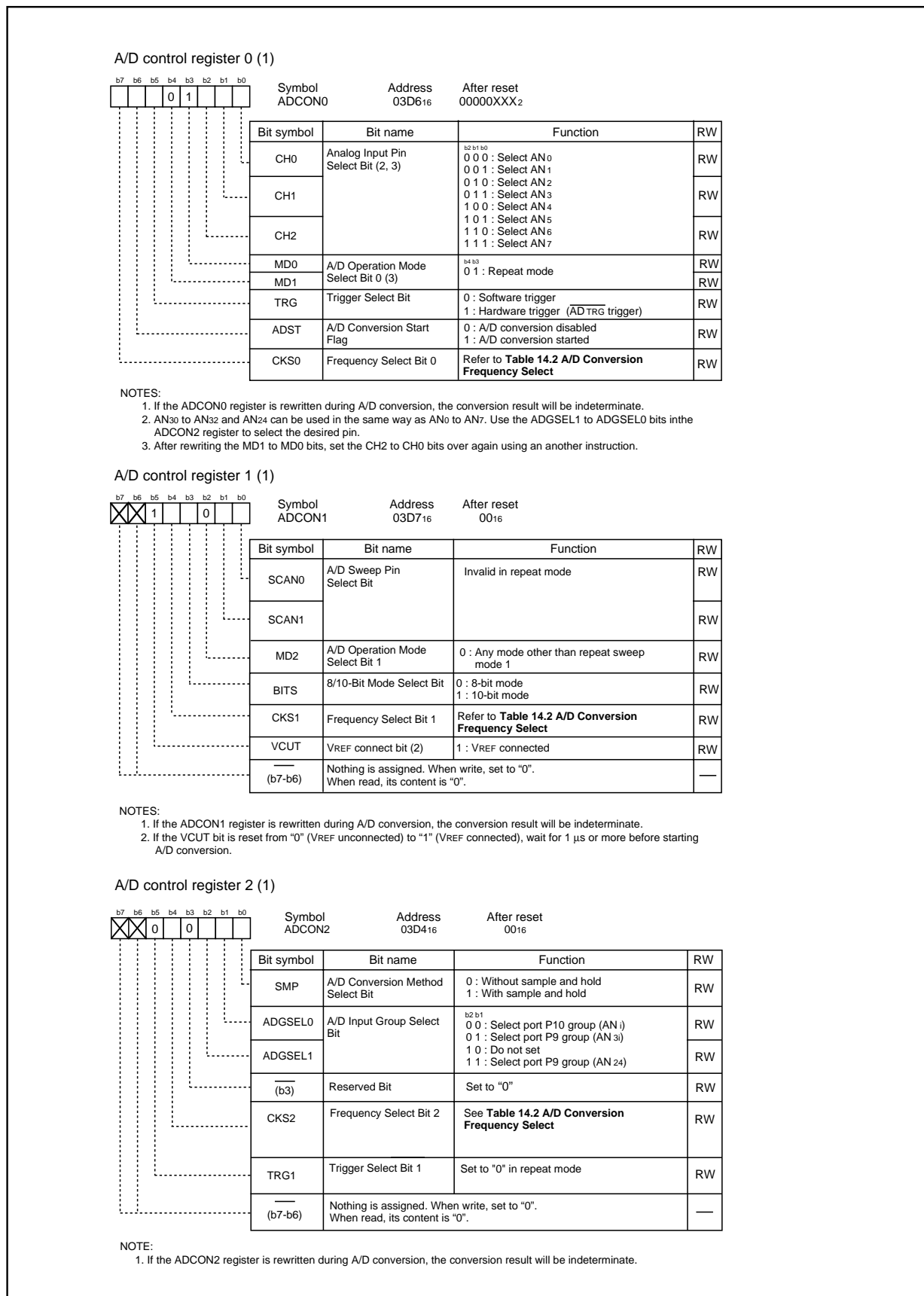


Figure 14.1.2.2 ADCON0 to ADCON2 Registers in Repeat Mode

### 14.1.3 Single Sweep Mode

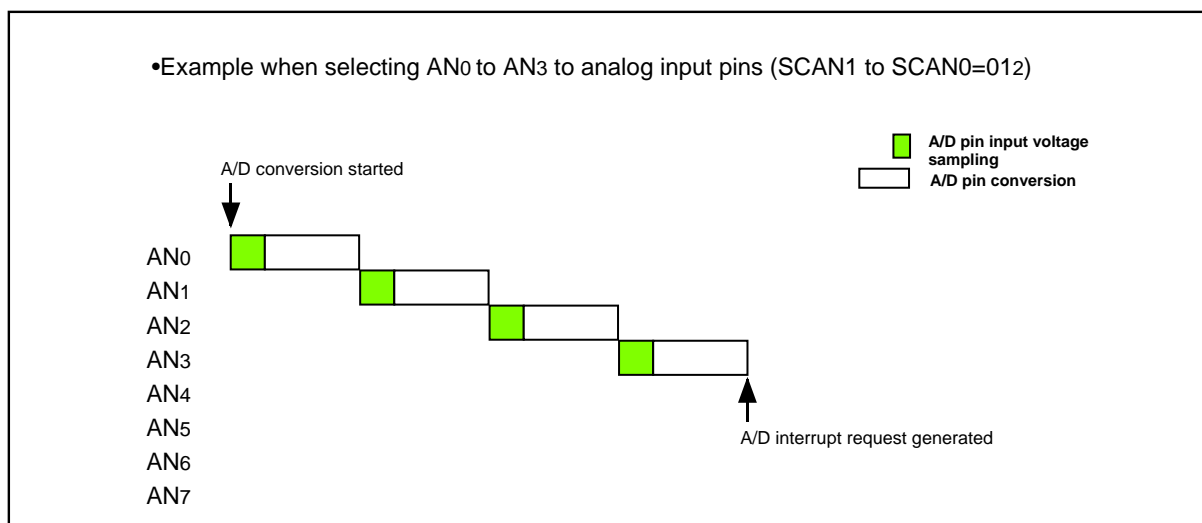
In single sweep mode, analog voltages applied to the selected pins are converted one-by-one to a digital code. Table 14.1.3.1 shows the single sweep mode specifications. Figure 14.1.3.1 shows the operation example in single sweep mode. Figure 14.1.3.2 shows the ADCON0 to ADCON2 registers in single sweep mode.

**Table 14.1.3.1 Single Sweep Mode Specifications**

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

NOTE:

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



**Figure 14.1.3.1 Operation Example in Single Sweep Mode**

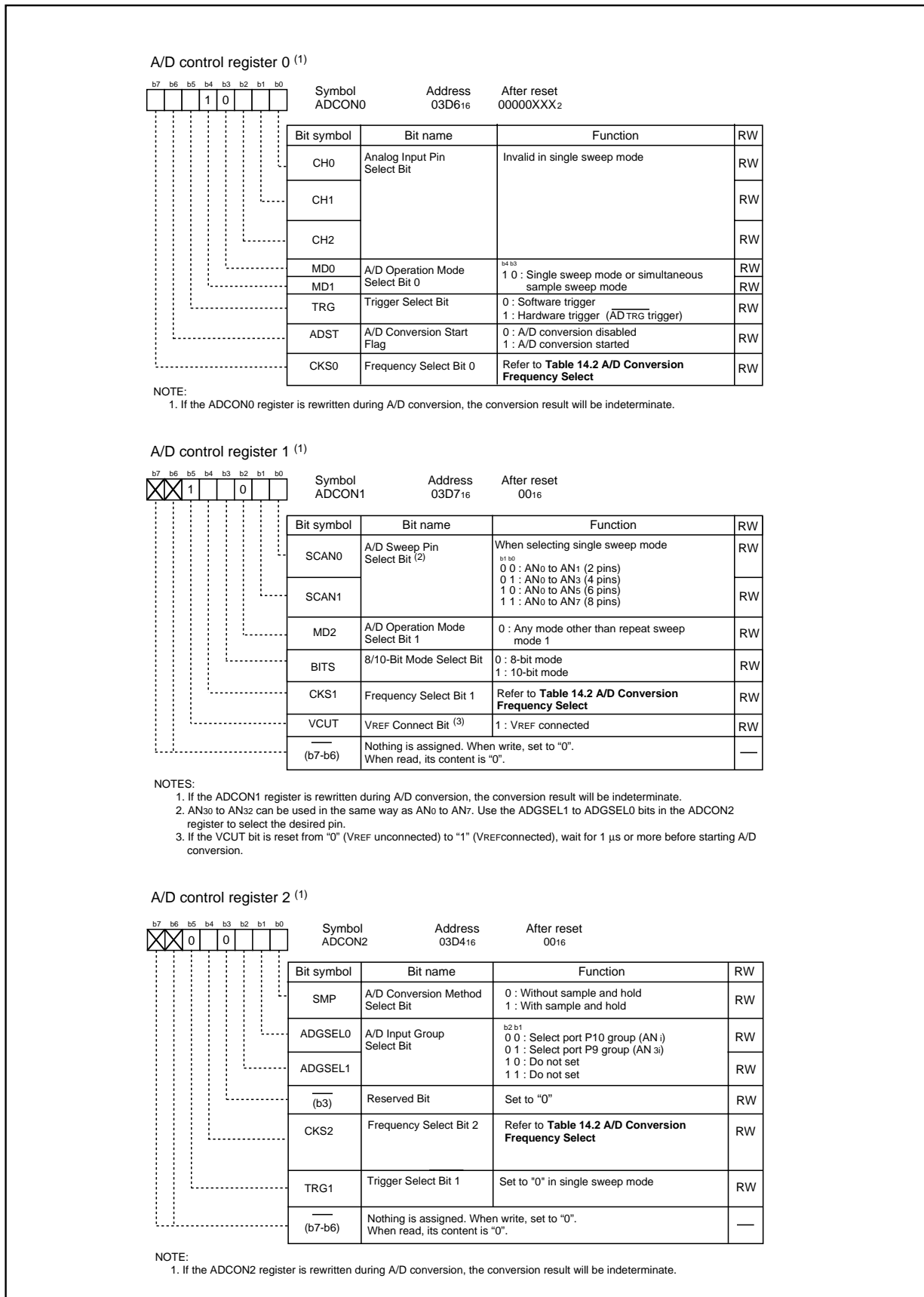


Figure 14.1.3.2 ADCON0 to ADCON2 Registers in Single Sweep Mode



### 14.1.4 Repeat Sweep Mode 0

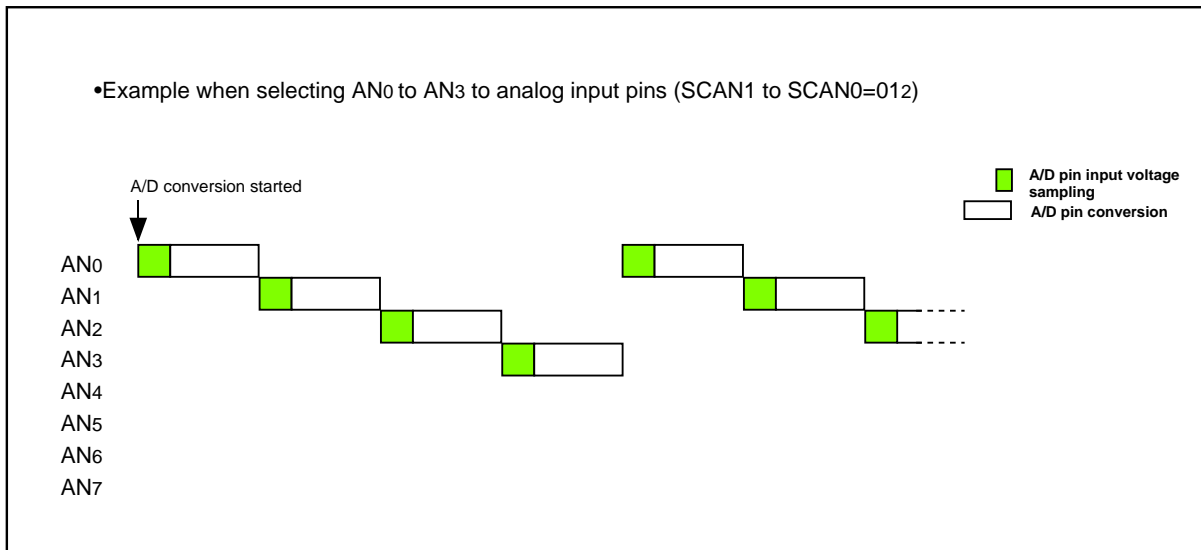
In repeat sweep mode 0, analog voltages applied to the selected pins are repeatedly converted to a digital code. Table 14.1.4.1 shows the repeat sweep mode 0 specifications. Figure 14.1.4.1 shows the operation example in repeat sweep mode 0. Figure 14.1.4.2 shows the ADCON0 to ADCON2 registers in repeat sweep mode 0.

**Table 14.1.4.1 Repeat Sweep Mode 0 Specifications**

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the selected pins is repeatedly converted to a digital code
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>When the TRG bit in the ADCON0 register is "0" (software trigger) Set the ADST bit in the ADCON0 register to "1" (A/D conversion started)</li> <li>When the TRG bit in the ADCON0 register is "1" (Hardware trigger) The ADTRG pin input changes state from "H" to "L" after setting the ADST bit to "1" (A/D conversion started)</li> </ul>
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) <sup>(1)</sup>
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin

**NOTE:**

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



**Figure 14.1.4.1 Operation Example in Repeat Sweep Mode 0**

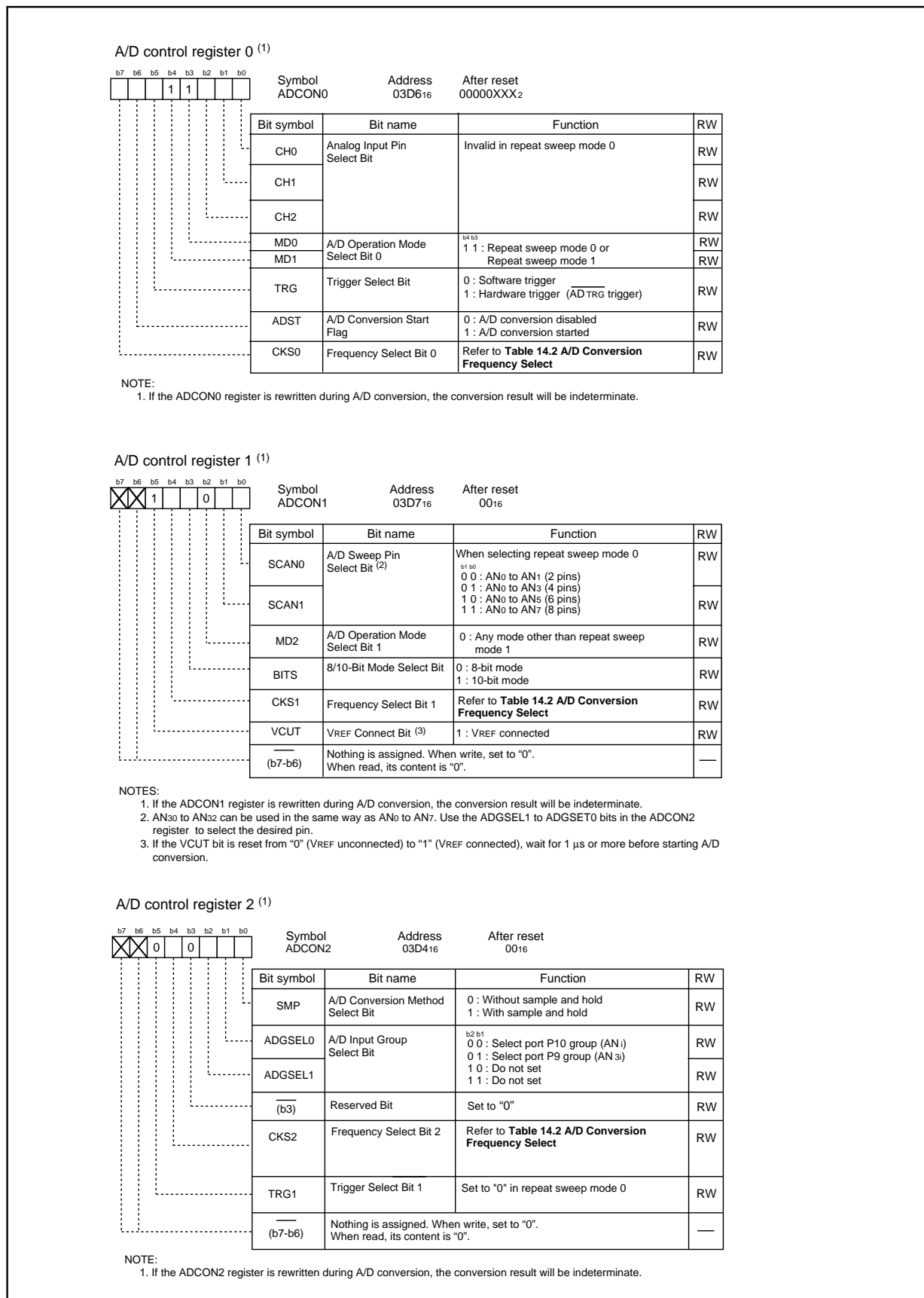


Figure 14.1.4.2 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 0

### 14.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltages applied to the all selected pins are converted to a digital code, with mainly used in the selected pins. Table 14.1.5.1 shows the repeat sweep mode 1 specifications.

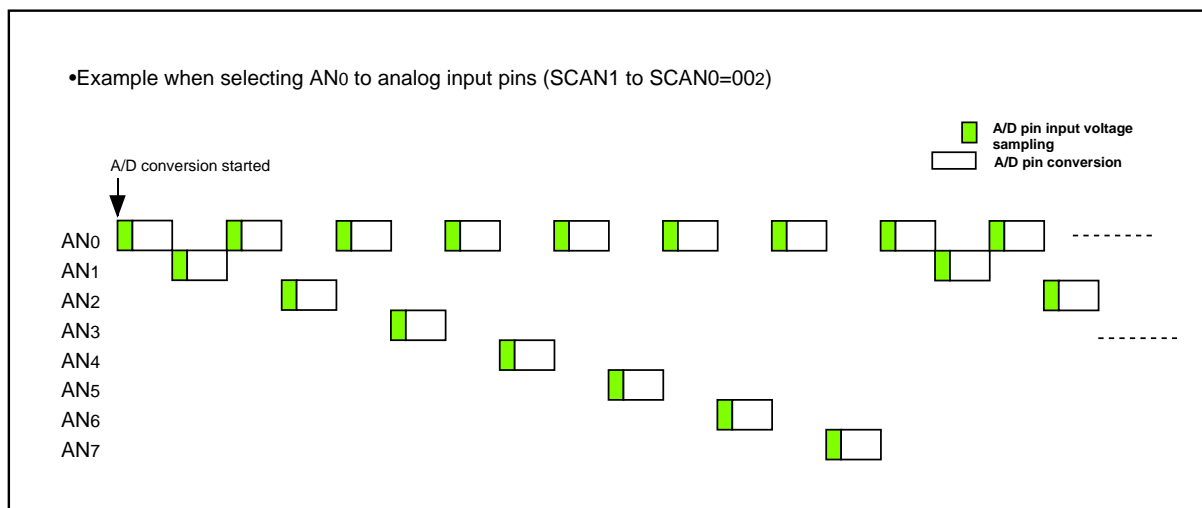
Figure 14.1.5.1 shows the operation example in repeat sweep mode 1. Figure 14.1.5.2 shows the ADCON0 to ADCON2 registers in repeat sweep mode 1.

**Table 14.1.5.1 Repeat Sweep Mode 1 Specifications**

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register mainly select pins. Analog voltage applied to the all selected pins is repeatedly converted to a digital code Example : When selecting AN0 Analog voltage is converted to a digital code in the following order AN0 → AN1 → AN0 → AN2 → AN0 → AN3, and so on.
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>When the TRG bit in the ADCON0 register is "0" (software trigger) Set the ADST bit in the ADCON0 register to "1" (A/D conversion started)</li> <li>When the TRG bit in the ADCON0 register is "1" (hardware trigger) The <math>\overline{\text{ADTRG}}</math> pin input changes state from "H" to "L" after setting the ADST bit to "1" (A/D conversion started)</li> </ul>
A/D Conversion Stop Condition	Set the ADST bit to "0" (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pins Mainly Used in A/D Conversions	Select from AN0 (1 pins), AN0 to AN1 (2 pins), AN0 to AN2 (3 pins), AN0 to AN3 (4 pins) <sup>(1)</sup>
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin

**NOTE:**

1. AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.



**Figure 14.1.5.1 Operation Example in Repeat Sweep Mode 1**

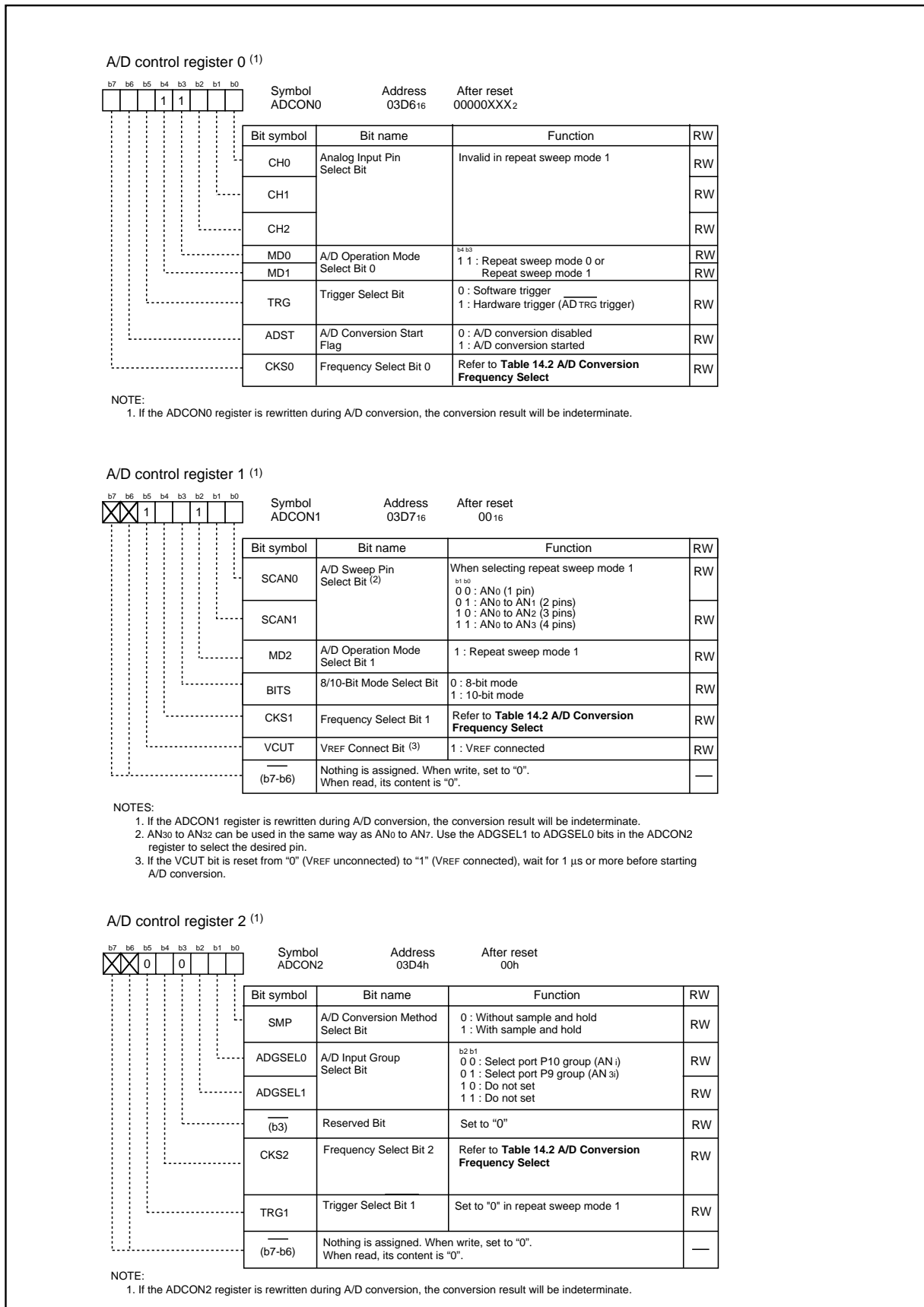


Figure 14.1.5.2 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 1

### 14.1.6 Simultaneous Sample Sweep Mode

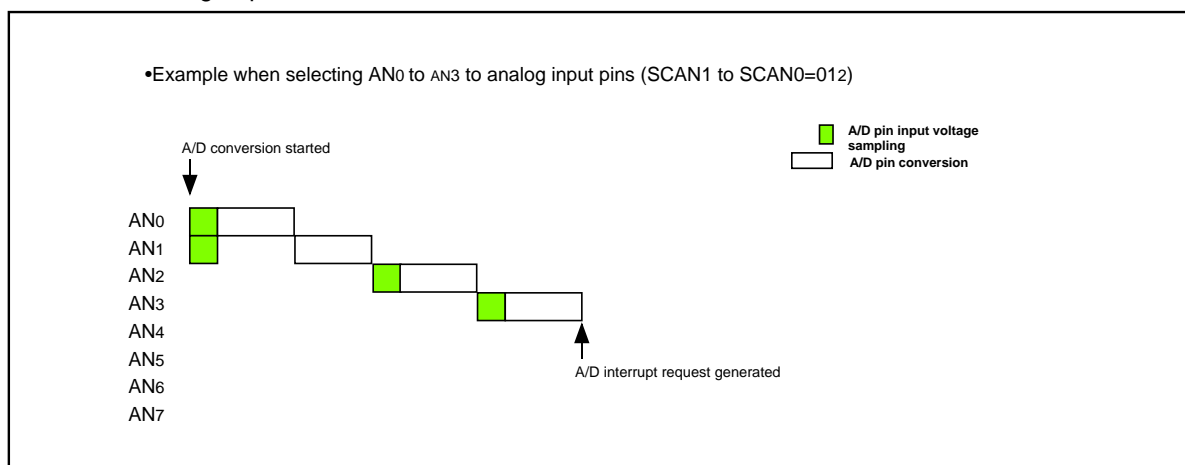
In simultaneous sample sweep mode, analog voltages applied to the selected pins are converted one-by-one to a digital code. At this time, the input voltage of AN0 and AN1 are sampled simultaneously using two circuits of sample and hold circuit. Table 14.1.6.1 shows the simultaneous sample sweep mode specifications. Figure 14.1.6.1 shows the operation example in simultaneous sample sweep mode. Figure 14.1.6.2 shows ADCON0 to ADCON2 registers and Figure 14.1.6.3 shows ADTRGCON registers in simultaneous sample sweep mode. Table 14.1.6.2 shows the trigger select bit setting in simultaneous sample sweep mode. In simultaneous sample sweep mode, Timer B0 underflow can be selected as a trigger by combining software trigger,  $\overline{\text{ADTRG}}$  trigger, Timer B2 underflow, Timer B2 interrupt generation frequency setting counter underflow or A/D trigger mode of Timer B.

**Table 14.1.6.1 Simultaneous Sample Sweep Mode Specifications**

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the selected pins is converted one-by-one to a digital code. At this time, the input voltage of AN0 and AN1 are sampled simultaneously.
A/D Conversion Start Condition	When the TRG bit in the ADCON0 register is "0" (software trigger) Set the ADST bit in the ADCON0 register to "1" (A/D conversion started) When the TRG bit in the ADCON0 register is "1" (hardware trigger) The trigger is selected by TRG1 and HPTRG0 bits (See <b>Table 14.1.6.2</b> ) The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" after setting the ADST bit to "1" (A/D conversion started) Timer B0, B2 or Timer B2 interrupt generation frequency setting counter underflow after setting the ADST bit to "1" (A/D conversion started)
A/D Conversion Stop Condition	A/D conversion completed (If selecting software trigger, the ADST bit is automatically set to "0". Set the ADST bit to "0" (A/D conversion halted)
Interrupt Generation Timing	A/D conversion completed
Analog Input Pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins) <sup>(1)</sup>
Readout of A/D conversion result	Readout one of the AN0 to AN7 registers that corresponds to the selected pin

NOTE:

1. AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.



**Figure 14.1.6.1 Operation Example in Simultaneous Sample Sweep Mode**

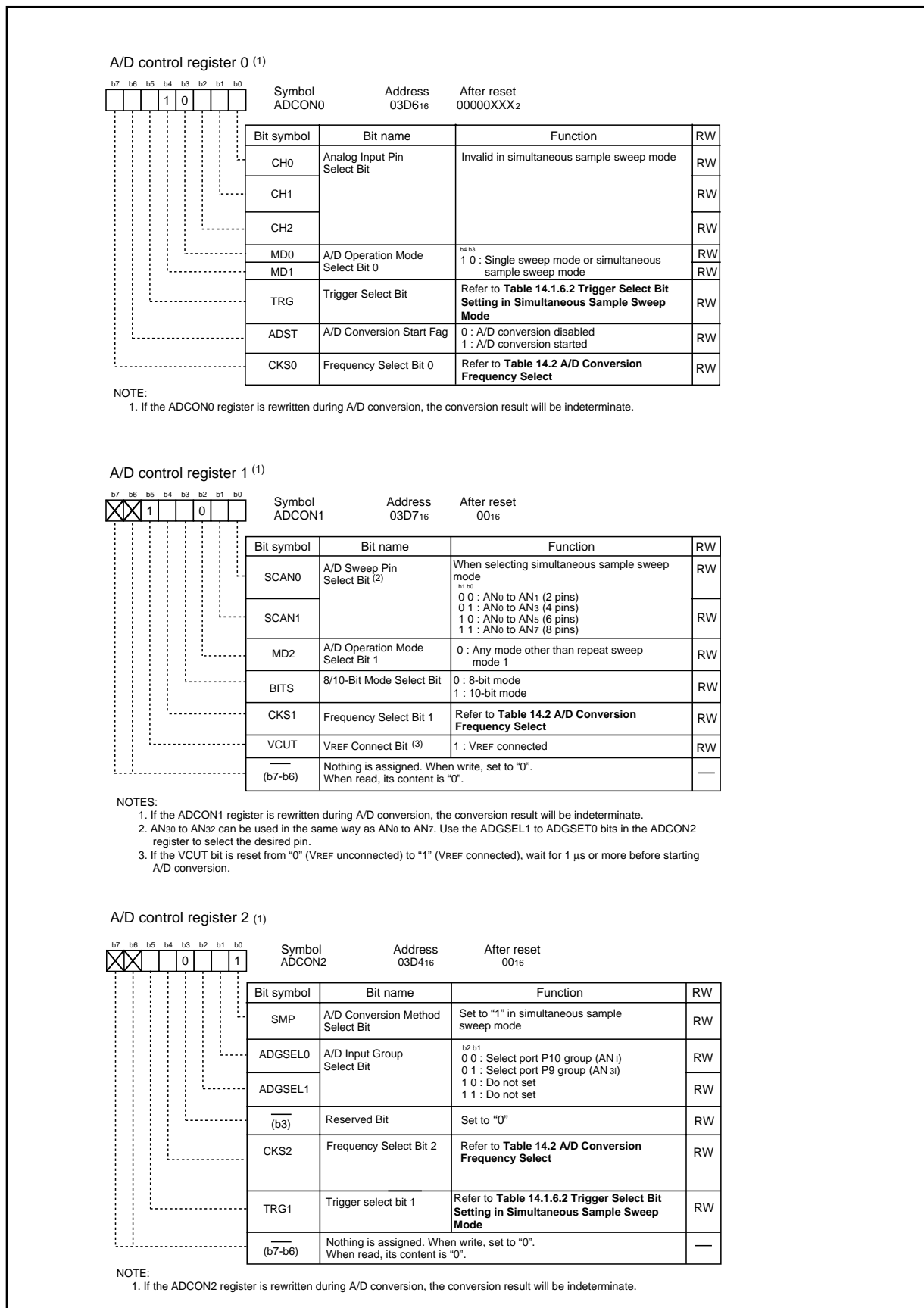


Figure 14.1.6.2 ADCON0 to ADCON2 Registers for Simultaneous Sample Sweep Mode

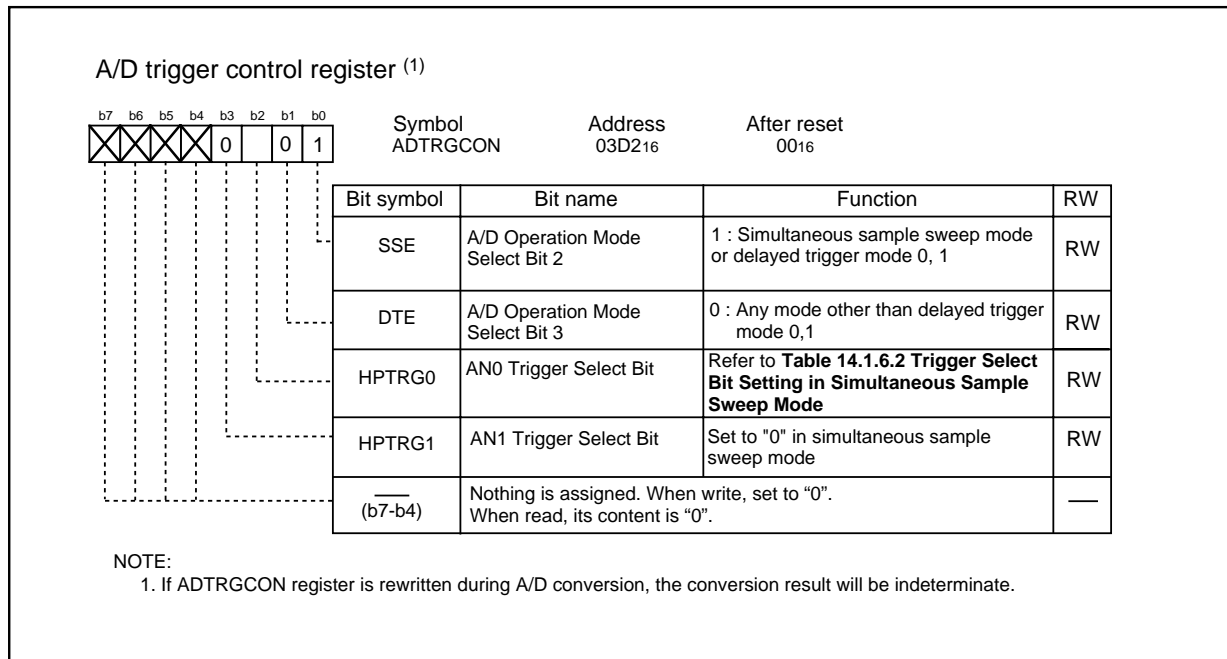


Figure 14.1.6.3 ADTRGCON Register in Simultaneous Sample Sweep Mode

Table 14.1.6.2 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode

TRG	TRG1	HPTRG0	TRIGGER
0	-	-	Software trigger
1	-	1	Timer B0 underflow (1)
1	0	0	$\overline{\text{ADTRG}}$
1	1	0	Timer B2 or Timer B2 interrupt generation frequency setting counter underflow (2)

## NOTE:

1. A count can be started for Timer B2, Timer B2 interrupt generation frequency setting counter underflow or the  $\overline{\text{INT5}}$  pin falling edge as count start conditions of Timer B0.
2. Select Timer B2 or Timer B2 interrupt generation frequency setting counter using the TB2SEL bit in the TB2SC register.

### 14.1.7 Delayed Trigger Mode 0

In delayed trigger mode 0, analog voltages applied to the selected pins are converted one-by-one to a digital code. The delayed trigger mode 0 used in combination with A/D trigger mode of Timer B. The Timer B0 underflow starts a single sweep conversion. After completing the AN0 pin conversion, the AN1 pin is not sampled and converted until the Timer B1 underflow is generated. When the Timer B1 underflow is generated, the single sweep conversion is restarted with the AN1 pin. Table 14.1.7.1 shows the delayed trigger mode 0 specifications. Figure 14.1.7.1 shows the operation example in delayed trigger mode 0. Figure 14.1.7.2 and Figure 14.1.7.3 show each flag operation in the ADSTAT0 register that corresponds to the operation example. Figure 14.1.7.4 shows the ADCON0 to ADCON2 registers in delayed trigger mode 0. Figure 14.1.7.5 shows the ADTRGCON register in delayed trigger mode 0 and Table 14.1.7.2 shows the trigger select bit setting in delayed trigger mode 0.

**Table 14.1.7.1 Delayed Trigger Mode 0 Specifications**

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to the input voltage of the selected pins are converted one-by-one to the digital code. At this time, Timer B0 underflow generation starts AN0 pin conversion. Timer B1 underflow generation starts conversion after the AN1 pin. <sup>(1)</sup>
A/D Conversion Start	AN0 pin conversion start condition <ul style="list-style-type: none"> <li>•When Timer B0 underflow is generated if Timer B0 underflow is generated again before Timer B1 underflow is generated, the conversion is not affected</li> <li>•When Timer B0 underflow is generated during A/D conversion of pins after the AN1 pin, conversion is halted and the sweep is restarted from AN0 pin</li> </ul> AN1 pin conversion start condition <ul style="list-style-type: none"> <li>•When Timer B1 underflow is generated during A/D conversion of the AN0 pin, the input voltage of the AN1 pin is sampled. The AN1 conversion and the rest of the sweep start when AN0 conversion is completed.</li> </ul>
A/D Conversion Stop Condition	<ul style="list-style-type: none"> <li>•When single sweep conversion from the AN0 pin is completed</li> <li>•Set the ADST bit to "0" (A/D conversion halted)<sup>(2)</sup></li> </ul>
Interrupt Request Generation Timing	A/D conversion completed
Analog Input Pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins) and AN0 to AN7 (8 pins) <sup>(3)</sup>
Readout of A/D Conversion Result	Readout one of the AN0 to AN7 registers that corresponds to the selected pins

**NOTES:**

1. Set the larger value than the value of the timer B0 register to the timer B1 register.
2. Do not write "1" (A/D conversion started) to the ADST bit in delayed trigger mode 0. When write "1", unexpected interrupts may be generated.
3. AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.



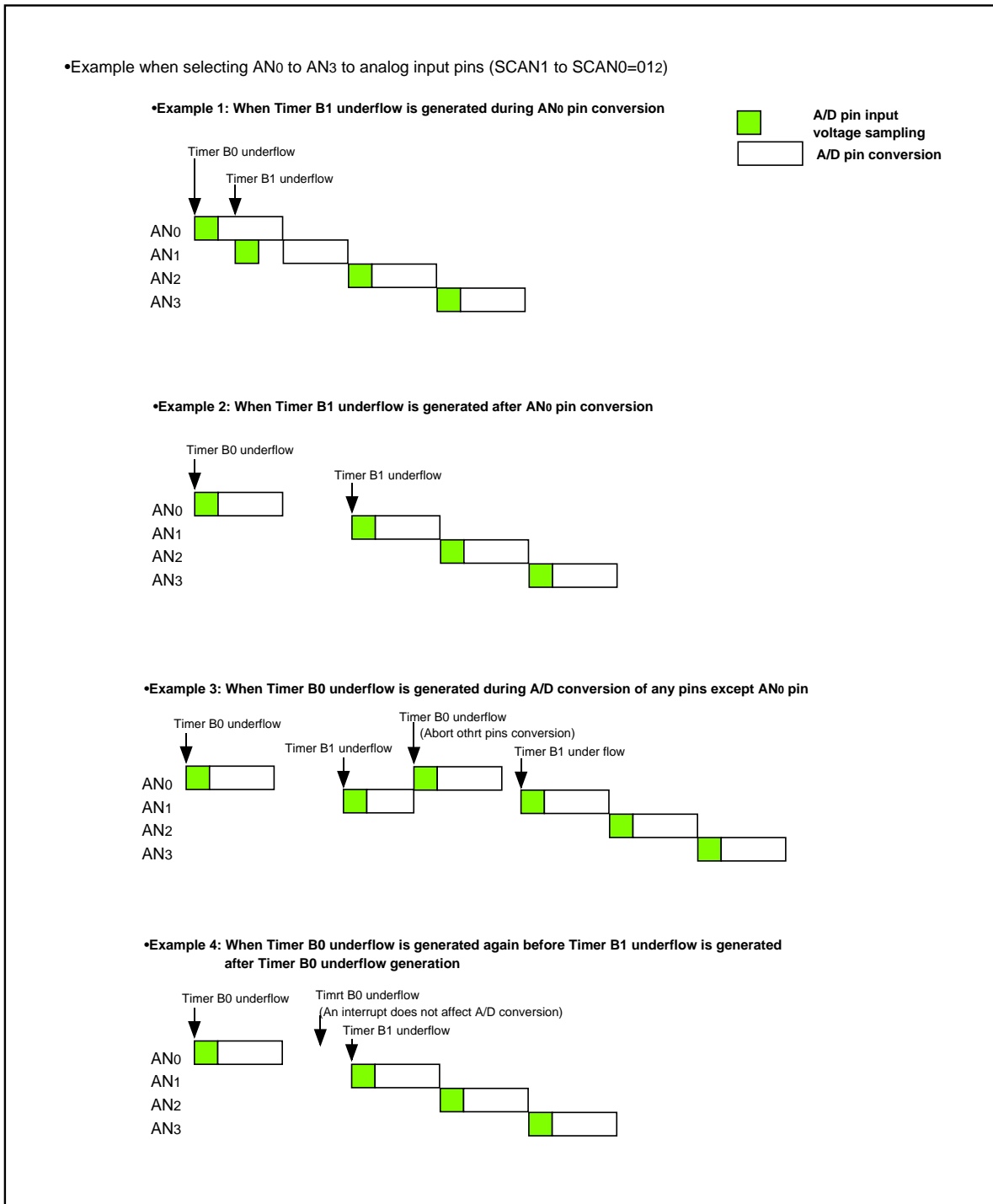


Figure 14.1.7.1 Operation Example in Delayed Trigger Mode 0

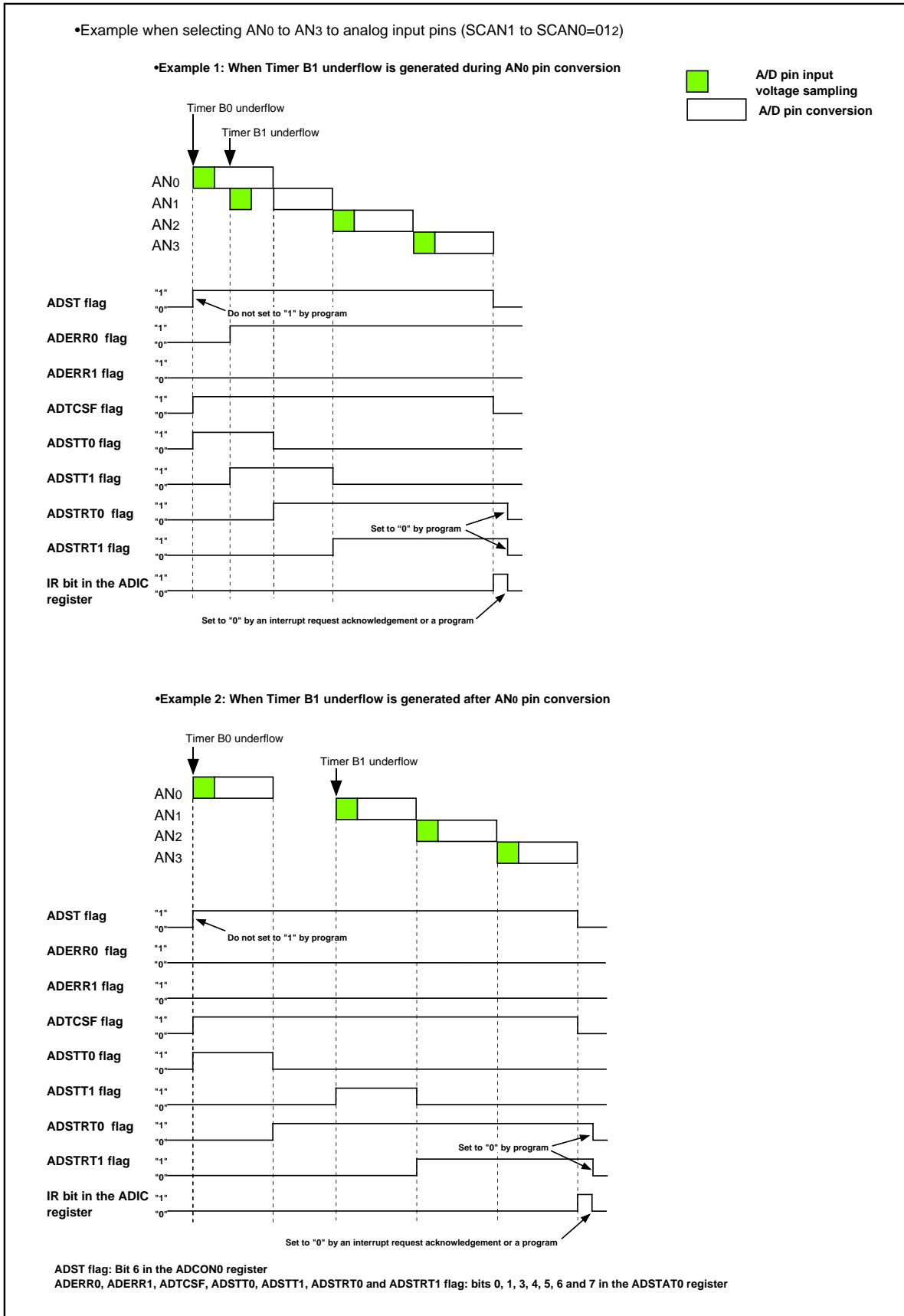


Figure 14.1.7.2 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (1)

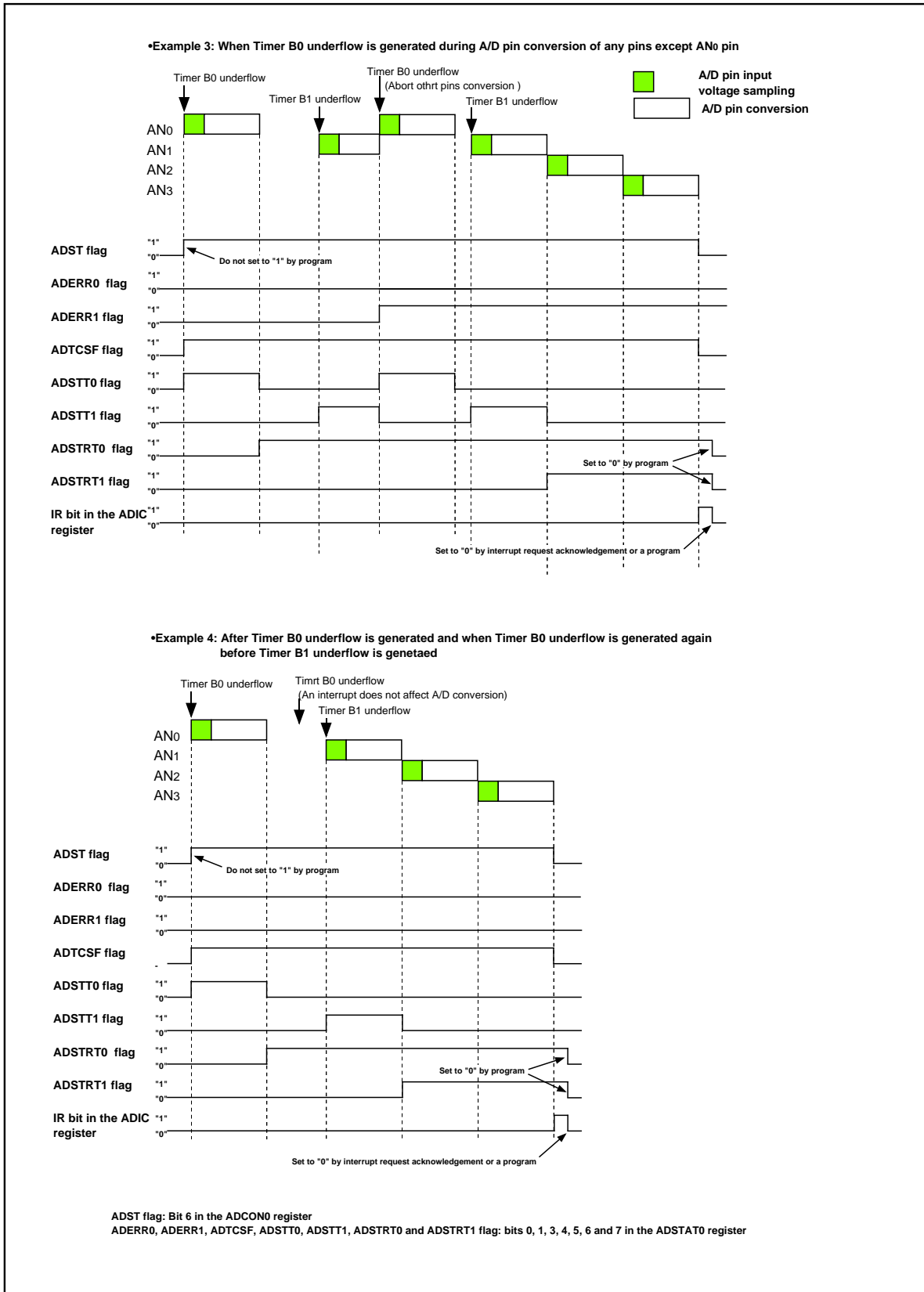


Figure 14.1.7.3 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (2)

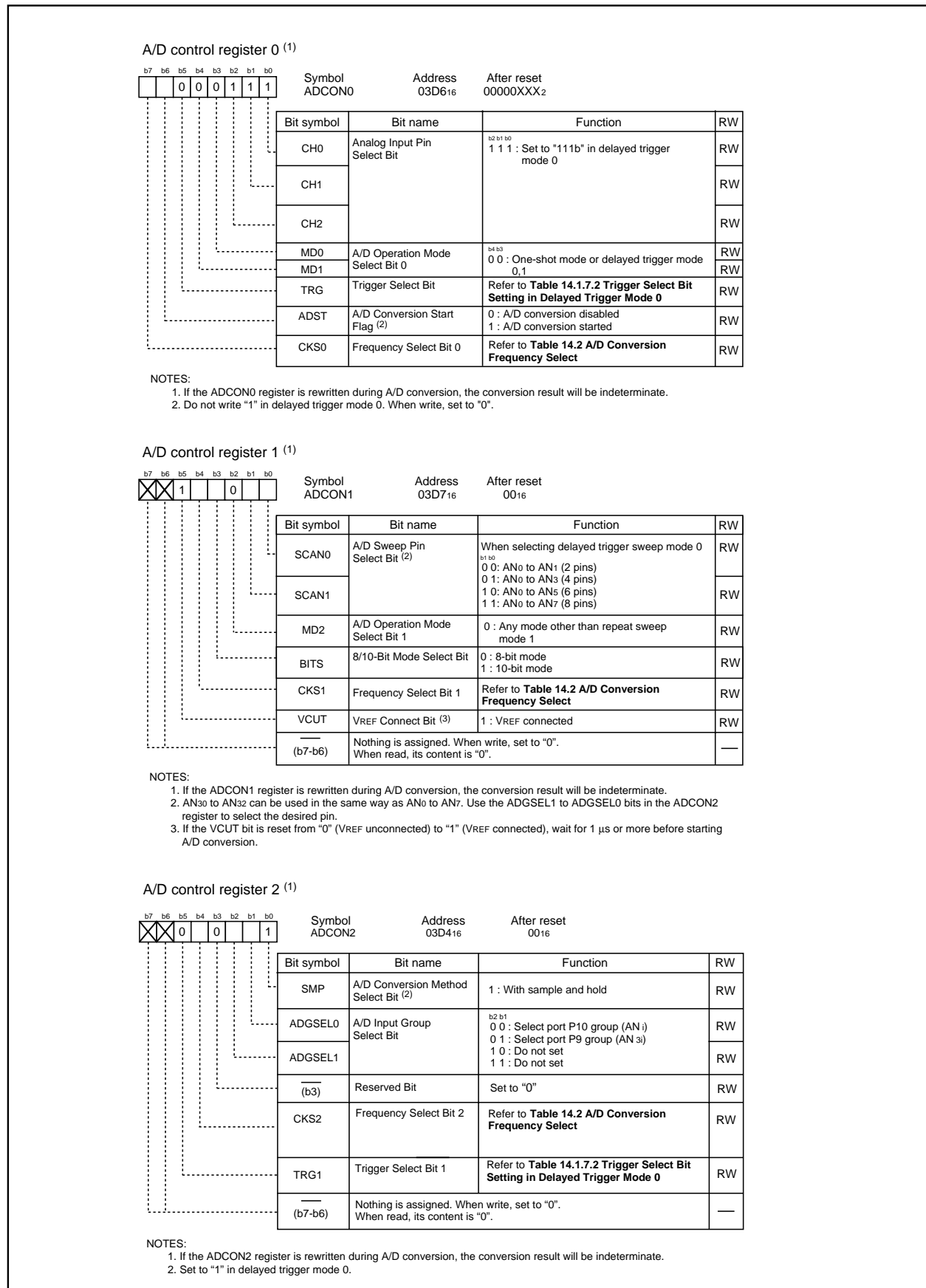


Figure 14.1.7.4 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 0

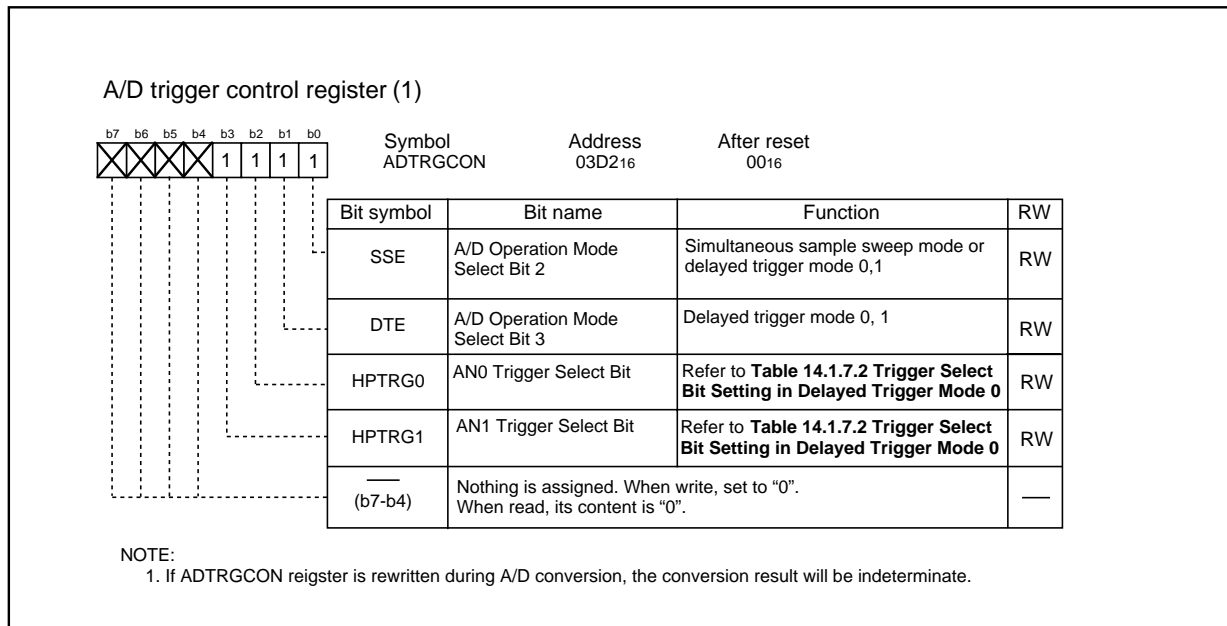


Figure 14.1.7.5 ADTRGCON Register in Delayed Trigger Mode 0

Table 14.1.7.2 Trigger Select Bit Setting in Delayed Trigger Mode 0

TRG	TRG1	HPTRG0	HPTRG1	Trigger
0	0	1	1	Timer B0, B1 underflow

### 14.1.8 Delayed Trigger Mode 1

In delayed trigger mode 1, analog voltages applied to the selected pins are converted one-by-one to a digital code. When the input of the  $\overline{\text{ADTRG}}$  pin (falling edge) changes state from "H" to "L", a single sweep conversion is started. After completing the AN0 pin conversion, the AN1 pin is not sampled and converted until the second  $\overline{\text{ADTRG}}$  pin falling edge is generated. When the second  $\overline{\text{ADTRG}}$  falling edge is generated, the single sweep conversion of the pins after the AN1 pin is restarted. Table 14.1.8.1 shows the delayed trigger mode 1 specifications. Figure 14.1.8.1 shows the operation example of delayed trigger mode 1. Figure 14.1.8.2 to Figure 14.1.8.3 show each flag operation in the ADSTAT0 register that corresponds to the operation example. Figure 14.1.8.4 shows the ADCON0 to ADCON2 registers in delayed trigger mode 1. Figure 14.1.8.5 shows the ADTRGCON register in delayed trigger mode 1 and Table 15.1.8.2 shows the trigger select bit setting in delayed trigger mode 1.

**Table 14.1.8.1 Delayed Trigger Mode 1 Specifications**

Item	Specification
Function	The SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltages applied to the selected pins are converted one-by-one to a digital code. At this time, the $\overline{\text{ADTRG}}$ pin falling edge starts AN0 pin conversion and the second $\overline{\text{ADTRG}}$ pin falling edge starts conversion of the pins after AN1 pin
A/D Conversion Start Condition	AN0 pin conversion start condition The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" (falling edge) <sup>(1)</sup> AN1 pin conversion start condition <sup>(2)</sup> The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" (falling edge) <ul style="list-style-type: none"> <li>When the second <math>\overline{\text{ADTRG}}</math> pin falling edge is generated during or after A/D conversion of the AN0 pin, input voltage of AN1 pin is sampled at the time of <math>\overline{\text{ADTRG}}</math> falling edge. The conversion of AN1 and the rest of the sweep starts when AN0 conversion is completed.</li> <li>When the <math>\overline{\text{ADTRG}}</math> pin falling edge is generated again during single sweep conversion of pins after the AN1 pin, the conversion is not affected</li> </ul>
A/D Conversion Stop Condition	<ul style="list-style-type: none"> <li>A/D conversion completed</li> <li>Set the ADST bit to "0" (A/D conversion halted)<sup>(3)</sup></li> </ul>
Interrupt Request Generation Timing	Single sweep conversion completed
Analog Input Pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins) and AN0 to AN7 (8 pins) <sup>(4)</sup>
Readout of A/D Conversion Result	Readout one of the AN0 to AN7 registers that corresponds to the selected pins

**NOTES:**

- Do not generate the next  $\overline{\text{ADTRG}}$  pin falling edge after the AN1 pin conversion is started until all selected pins complete A/D conversion. When an  $\overline{\text{ADTRG}}$  pin falling edge is generated again during A/D conversion, its trigger is ignored. The falling edge of  $\overline{\text{ADTRG}}$  pin, which was input after all selected pins complete A/D conversion, is considered to be the next AN0 pin conversion start condition.
- The  $\overline{\text{ADTRG}}$  pin falling edge is detected synchronized with the operation clock  $\phi_{\text{AD}}$ . Therefore, when the  $\overline{\text{ADTRG}}$  pin falling edge is generated in shorter periods than  $\phi_{\text{AD}}$ , the second  $\overline{\text{ADTRG}}$  pin falling edge may not be detected. Do not generate the  $\overline{\text{ADTRG}}$  pin falling edge in shorter periods than  $\phi_{\text{AD}}$ .
- Do not write "1" (A/D conversion started) to the ADST bit in delayed trigger mode 1. When write "1", unexpected interrupts may be generated.
- AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

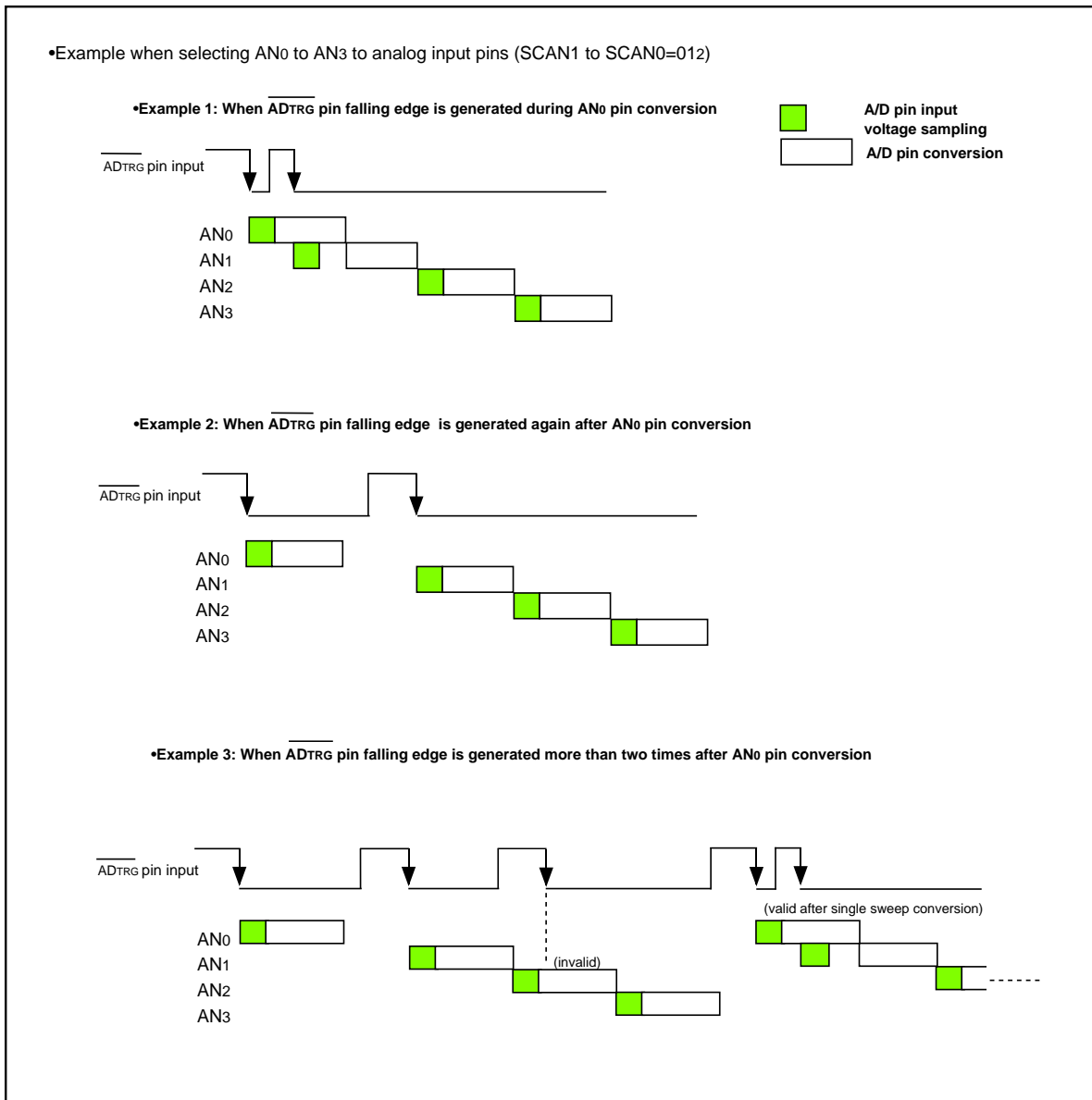


Figure 14.1.8.1 Operation Example in Delayed Trigger Mode1

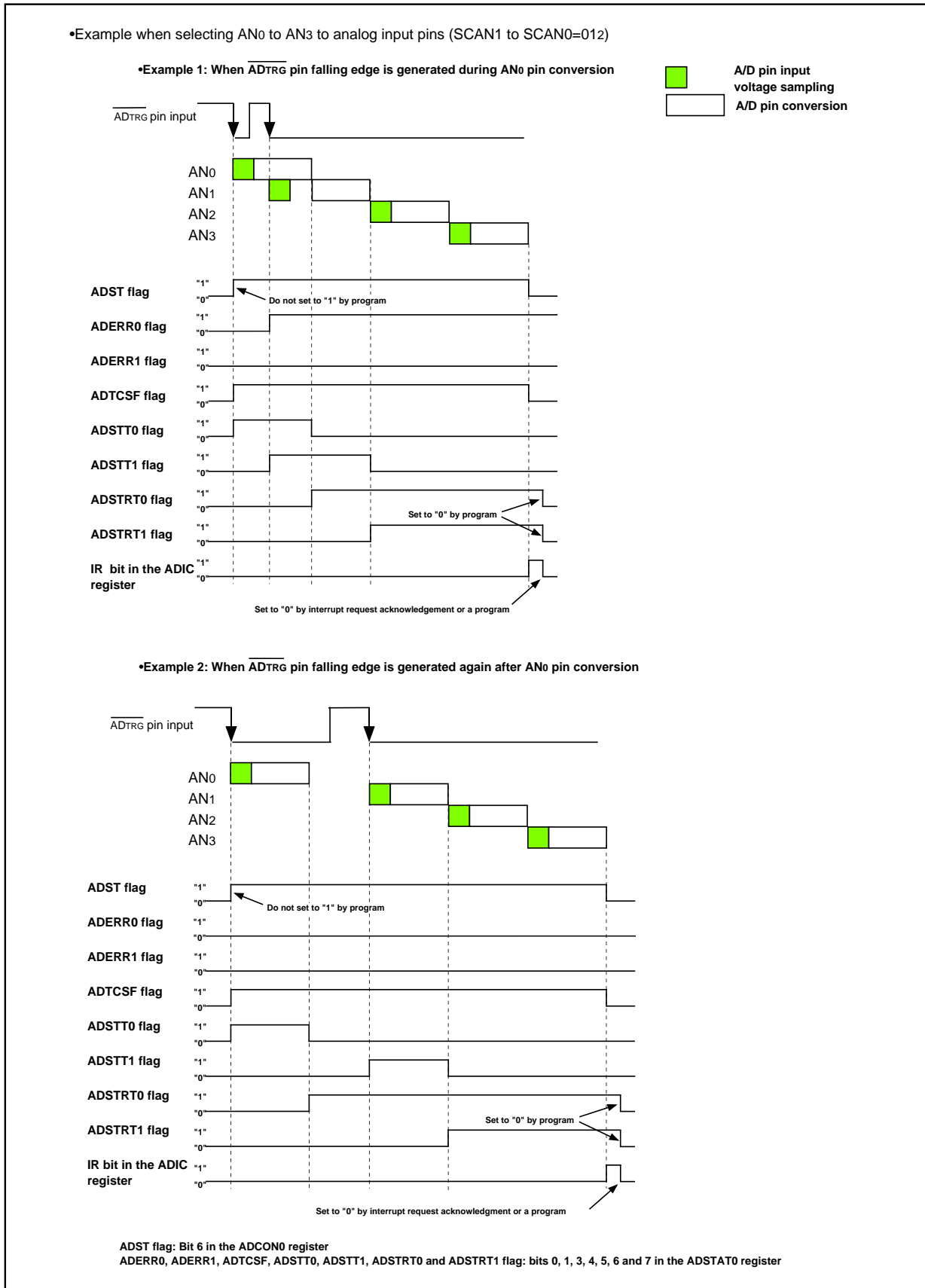


Figure 14.1.8.2 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (1)



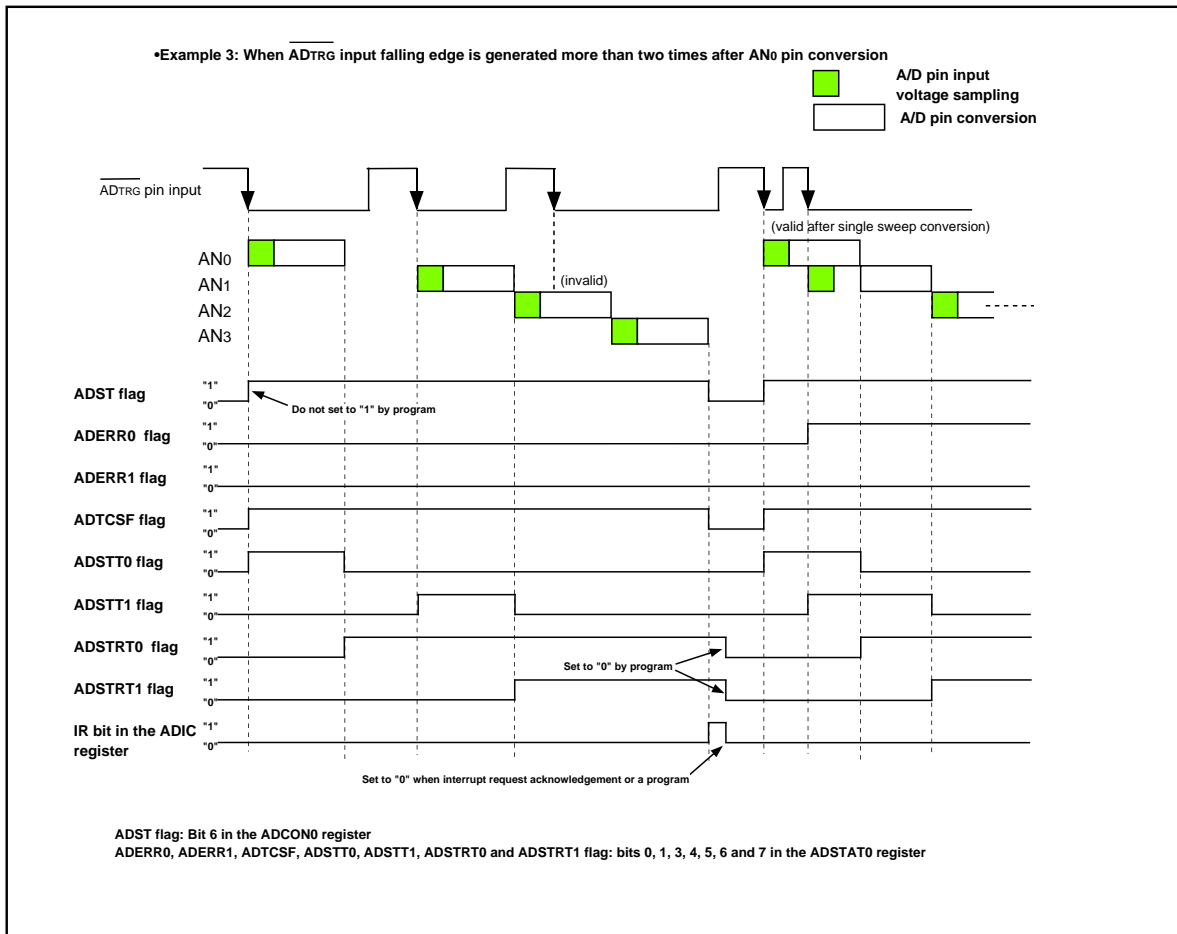


Figure 14.1.8.2 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (2)

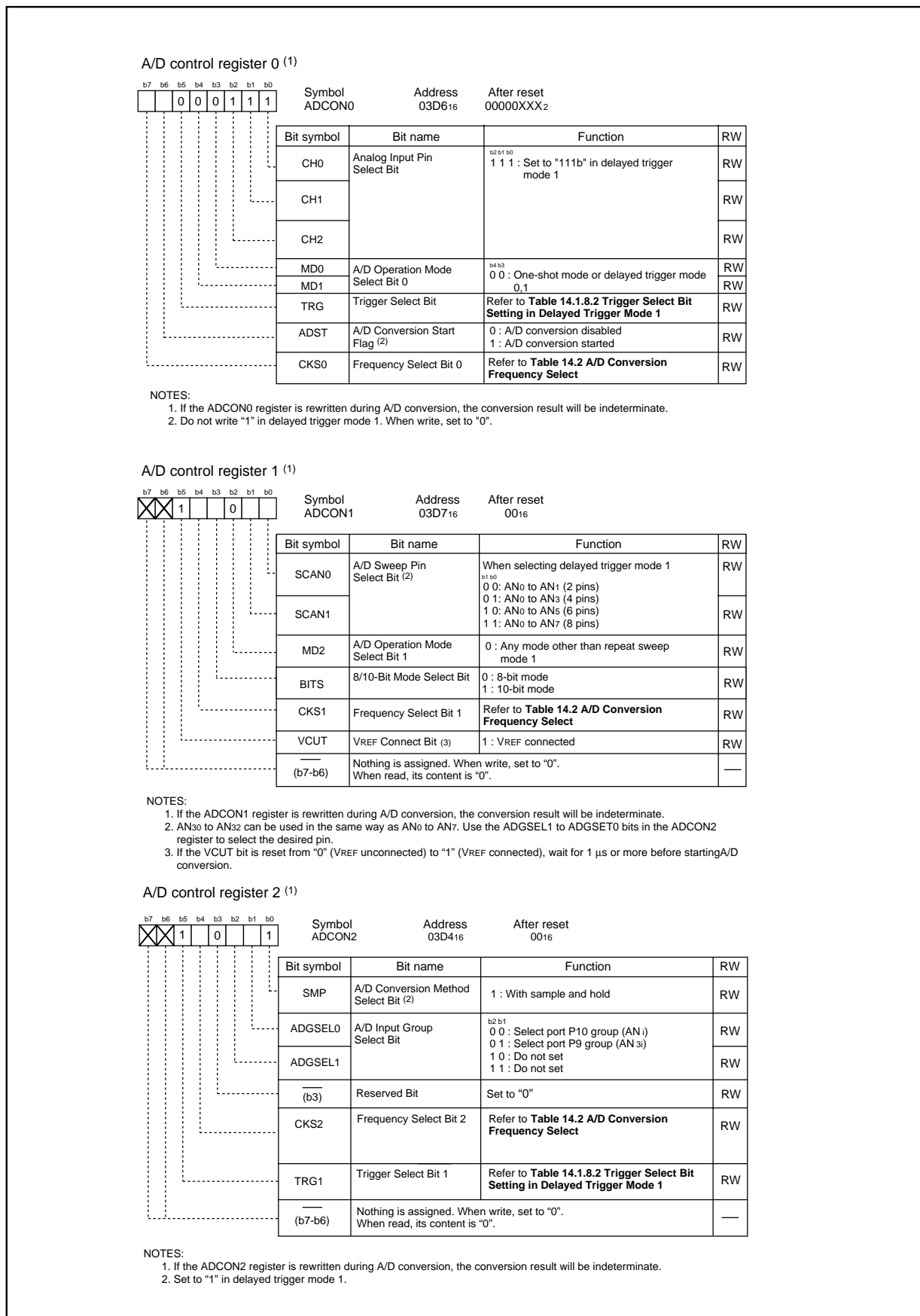
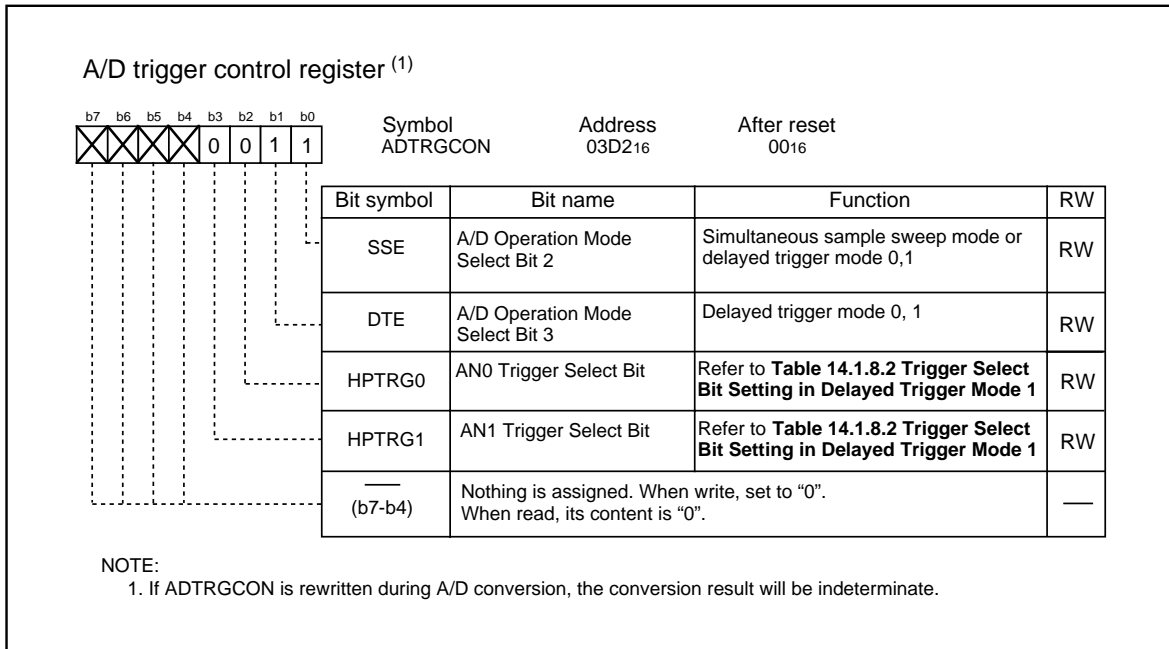


Figure 14.1.8.4 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 1



**Figure 14.1.8.5 ADTRGCON Register in Delayed Trigger Mode 1**

**Table 14.1.8.2 Trigger Select Bit Setting in Delayed Trigger Mode 1**

TRG	TRG1	HPTRG0	HPTRG1	Trigger
0	1	0	0	$\overline{\text{ADTRG}}$

## 14.2 Resolution Select Function

The BITS bit in the ADCON1 register determines the resolution. When the BITS bit is set to “1” (10-bit precision), the A/D conversion result is stored into bits 0 to 9 in the A/D register  $i$  ( $i=0$  to 7). When the BITS bit is set to “0” (8-bit precision), the A/D conversion result is stored into bits 0 to 7 in the AD $i$  register.

## 14.3 Sample and Hold

When the SMP bit in the ADCON 2 register is set to “1” (with the sample and hold function), A/D conversion rate per pin increases to 28  $\phi_{AD}$  cycles for 8-bit resolution or 33  $\phi_{AD}$  cycles for 10-bit resolution. The sample and hold function is available in one-shot mode, repeat mode, single sweep mode, repeat sweep mode 0 and repeat sweep mode 1. In these modes, start A/D conversion after selecting whether the sample and hold circuit is to be used or not. In simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode 1, set to use the Sample and Hold function before starting A/D conversion.

## 14.4 Power Consumption Reducing Function

When the A/D converter is not used, the VCUT bit in the ADCON1 register isolates the resistor ladder of the A/D converter from the reference voltage input pin (VREF). Power consumption is reduced by shutting off any current flow into the resistor ladder from the VREF pin.

When using the A/D converter, set the VCUT bit to “1” (VREF connected) before setting the ADST bit in the ADCON0 register to “1” (A/D conversion started). Do not set the ADST bit and VCUT bit to “1” simultaneously, nor set the VCUT bit to “0” (VREF unconnected) during A/D conversion.

## 14.5 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 14.5.1 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, microcomputer's internal resistance be R, precision (error) of the A/D converter be X, and the A/D converter's resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

$$VC \text{ is generally } VC = VIN \left\{ 1 - e^{-\frac{1}{C(R0+R)} t} \right\}$$

$$\text{And when } t = T, \quad VC = VIN - \frac{X}{Y} \quad VIN = VIN \left( 1 - \frac{X}{Y} \right)$$

$$e^{-\frac{1}{C(R0+R)} T} = \frac{X}{Y}$$

$$-\frac{1}{C(R0+R)} T = \ln \frac{X}{Y}$$

$$\text{Hence, } R0 = -\frac{T}{C \cdot \ln \frac{X}{Y}} - R$$

Figure 14.5.1 shows analog input pin and external sensor equivalent circuit. When the difference between VIN and VC becomes 0.1LSB, we find impedance R0 when voltage between pins. VC changes from 0 to VIN-(0.1/1024) VIN in timer T. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB. When f(XIN) = 10MHz, T=0.3μs in the A/D conversion mode with sample & hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3μs, R = 7.8kΩ, C = 1.5pF, X = 0.1, and Y = 1024. Hence,

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

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A/D converter turns out to be approximately 13.9kΩ.

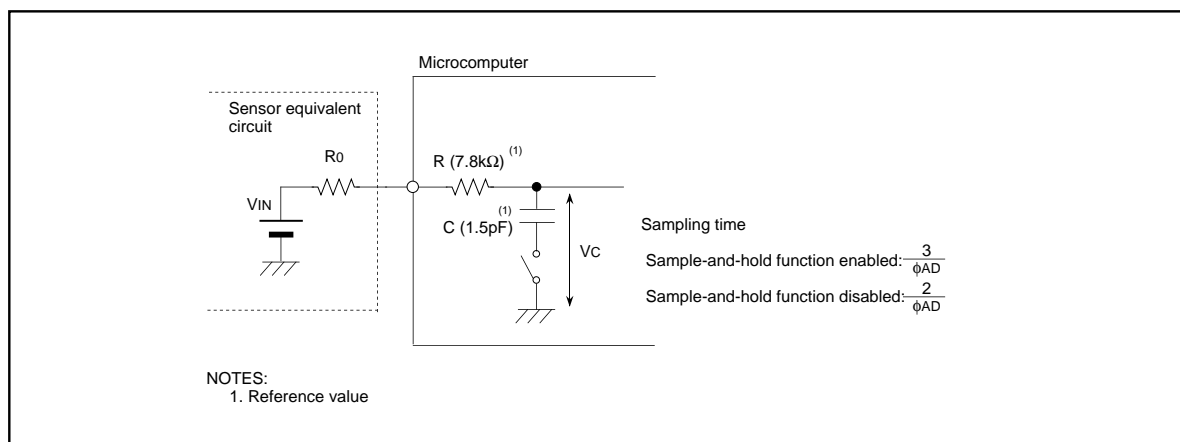


Figure 14.5.1 Analog Input Pin and External Sensor Equivalent Circuit

## 15. CRC Calculation Circuit

The Cyclic Redundancy Check (CRC) operation detects an error in data blocks. The microcomputer uses a generator polynomial of CRC\_CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) or CRC-16 ( $X^{16} + X^{15} + X^2 + 1$ ) to generate CRC code.

The CRC code is a 16-bit code generated for a block of a given data length in multiples of bytes. The code is updated in the CRC data register everytime one byte of data is transferred to a CRC input register. The data register needs to be initialized before use. Generation of CRC code for one byte of data is completed in two machine cycles.

Figure 15.1 shows the block diagram of the CRC circuit. Figure 15.2 shows the CRC-related registers. Figure 15.3 shows the calculation example using the CRC\_CCITT operation.

### 15.1. CRC Snoop

The CRC circuit includes the ability to snoop reads and writes to certain SFR addresses. This can be used to accumulate the CRC value on a stream of data without using extra bandwidth to explicitly write data into the CRCIN register. All SFR addresses after  $0020_{16}$  are subject to the CRC snoop. The CRC snoop is useful to snoop the writes to a UART TX buffer, or the reads from a UART RX buffer.

To snoop an SFR address, the target address is written to the CRC snoop Address Register (bits 9 to 0 in the CRCSAR register). The two most significant bits in this register enable snooping on reads or writes to the target address. If the target SFR is written to by the CPU or DMA, and the CRC snoop write bit is set (the CRCSW bit is set to "1"), the CRC will latch the data into the CRCIN register. The new CRC code will be set in the CRCD register.

Similarly, if the target SFR is read by the CRC or DMA, and the CRC snoop read bit is set (the CRCSR bit is set to "1"), the CRC will latch the data from the target into the CRCIN register and calculate the CRC.

The CRC circuit can only calculate CRC codes on data byte at a time. Therefore, if a target SFR is accessed in a word (16 bit) bus cycle, only the byte of data going to or from the target snooped into CRCIN, the other byte of the word access is ignored.

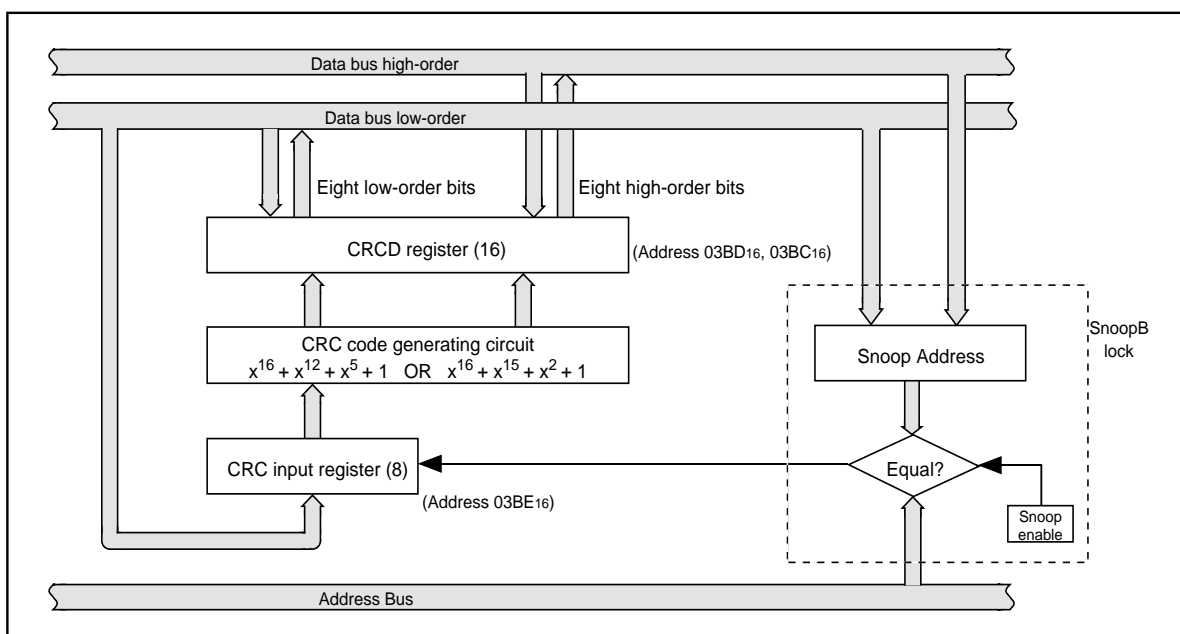


Figure 15.1 CRC circuit block diagram

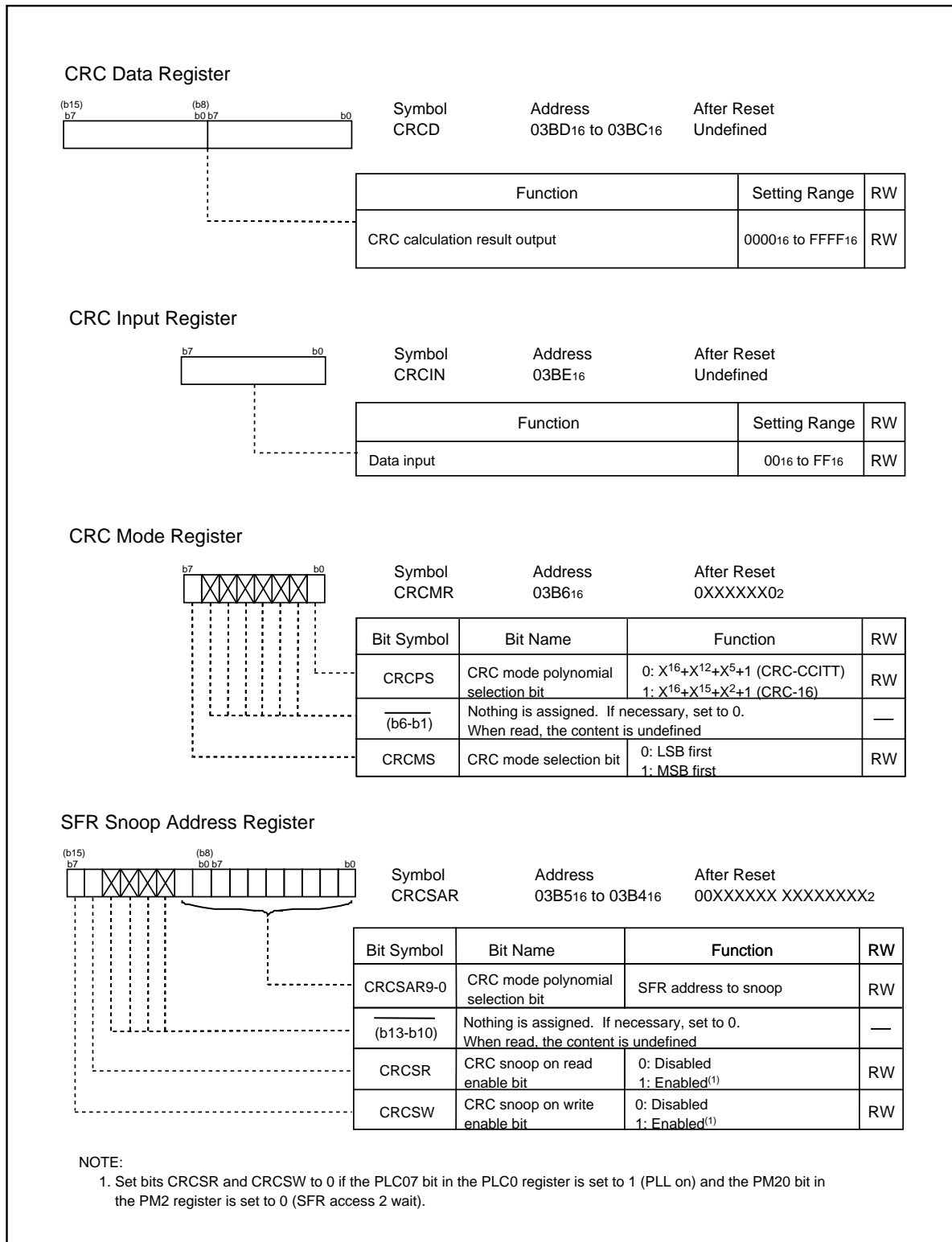


Figure 15.2. CRCD, CRCIN, CRCMR, CRCSAR Register

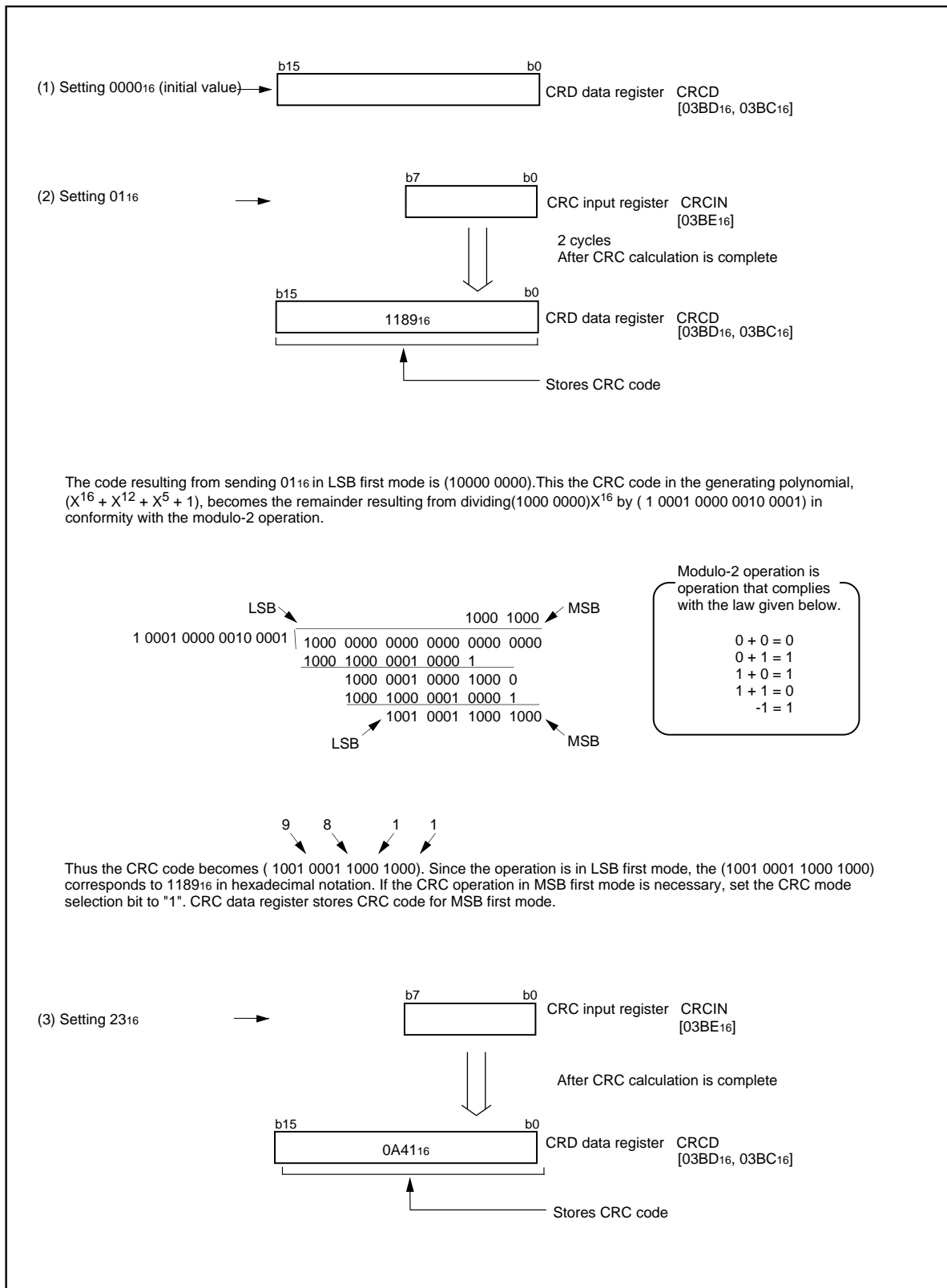


Figure 15.3. CRC Calculation



## 16. Programmable I/O Ports

**Note**

P60 to P63, P92 and P93 are not available in the 42-pin package.

The programmable input/output ports (hereafter referred to simply as “I/O ports”) consist of 39 lines P15 to P17, P6, P7, P8, P90 to P93, P10 for the 48-pin package, or 33 lines P15 to P17, P64 to P67, P7, P8, P90 to P91, P10 for the 42-pin package. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high in sets of 4 lines.

Figures 16.1 to 16.4 show the I/O ports. Figure 16.5 shows the I/O pins.

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

For details on how to set peripheral functions, refer to each functional description in this manual. If any pin is used as a peripheral function input, set the direction bit for that pin to “0” (input mode). Any pin used as an output pin for peripheral functions is directed for output no matter how the corresponding direction bit is set.

### 16.1 Port Pi Direction Register (PDi Register, i = 1, 6 to 10)

Figure 16.1.1 shows the direction registers.

This register selects whether the I/O port is to be used for input or output. The bits in this register correspond one for one to each port.

### 16.2 Port Pi Register (Pi Register, i = 1, 6 to 10)

Figure 16.2.1 shows the Pi registers.

Data input/output to and from external devices are accomplished by reading and writing to the Pi register. The Pi register consists of a port latch to hold the output data and a circuit to read the pin status. For ports set for input mode, the input level of the pin can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register.

For ports set for output mode, the port latch can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register. The data written to the port latch is output from the pin. The bits in the Pi register correspond one for one to each port.

### 16.3 Pull-up Control Register 0 to Pull-up Control Register 2 (PUR0 to PUR2 Registers)

Figure 16.3.1 shows the PUR0 to PUR2 registers.

The PUR0 to PUR2 registers select whether the ports, divided into groups of four ports, are pulled up or not. The ports, selected by setting the bits in registers PUR2 to PUR0 to “1” (pull-up), are pulled up when the direction registers are set to “0” (input mode). The ports are pulled up regardless of their function.

## 16.4 Port Control Register

Figure 16.4.1 shows the port control register.

When the P1 register is read after setting the PCR0 bit in the PCR register to “1”, the corresponding port latch can be read no matter how the PD1 register is set.

## 16.5 Pin Assignment Control register (PACR)

Figure 16.5.1 shows the PACR. After reset set the PACR2 to PACR0 bit before you input and output it to each pin. When the PACR register isn't set up, the input and output function of some of the pins doesn't work.

PACR2 to PACR0 bits: control the pins enabled for use.

At reset, these bits are “000”.

In 48-pin package, set these bits to “100<sub>2</sub>”.

In 42-pin package, set these bits to “001<sub>2</sub>”.

U1MAP: controls the assignment of UART1 pins.

If the U1MAP bit is set to “0” (P67 to P64) the UART1 functions are mapped to P64/CTS1/RTS1, P65/CLK1, P66/RxD1, and P67/TxD1.

If the U1MAP bit is set to “1” (P73 to P70) the UART1 functions are mapped to P70/CTS1/RTS1, P71/CLK1, P72/RxD1, and P73/TxD1.

PACR is write protected by PRC2 bit in the PRCR register. PRC2 bit must be set immediately before the write to PACR.

## 16.6 Digital Debounce function

Two digital debounce function circuits are provided. Level is determined when level is held, after applying either a falling edge or rising edge to the pin, longer than the programmed filter width time. This enables noise reduction.

This function is assigned to  $\overline{\text{INT5}}/\overline{\text{INPC17}}$  and  $\overline{\text{NMI}}/\overline{\text{SD}}$ . Digital filter width is set in the NDDR register and the P17DDR register respectively. Additionally, a digital debounce function is disabled to the port P17 input and port P8s input. Figure 16.6.1 shows the NDDR register and the P17DDR register.

Filter width :  $(n+1) \times 1/f8$  n: count value set in the NDDR register and P17DDR register

The NDDR register and the P17DDR register decrement count value with f8 as the count source. The NDDR register and the P17DDR register indicate count time. Count value is reloaded if a falling edge or a rising edge is applied to the pin.

The NDDR register and the P17DDR register can be set 00<sub>16</sub> to FF<sub>16</sub> when using the digital debounce function. Setting to FF<sub>16</sub> disables the digital filter. See Figure 16.6.2 for details.

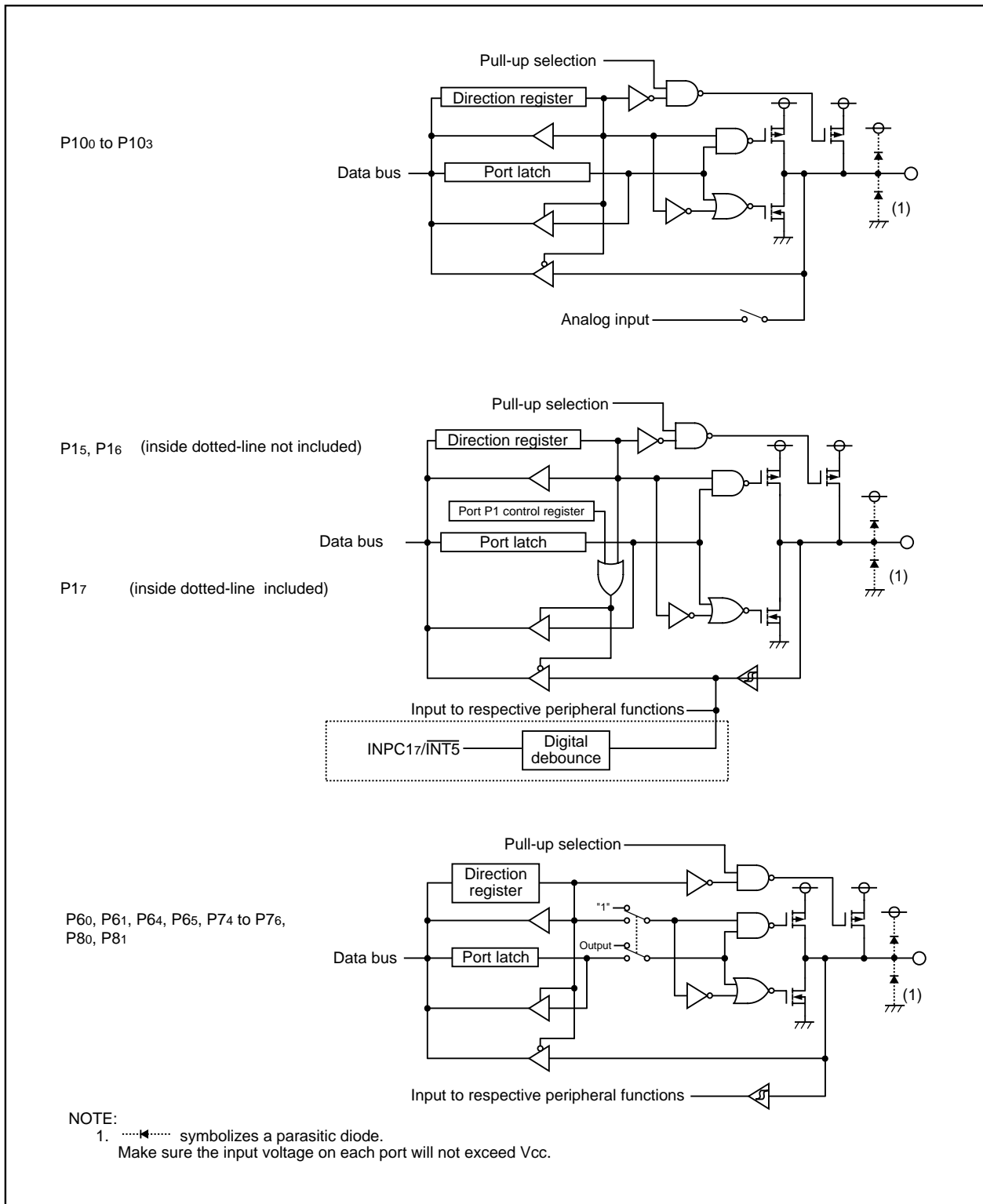


Figure 16.1. I/O Ports (1)

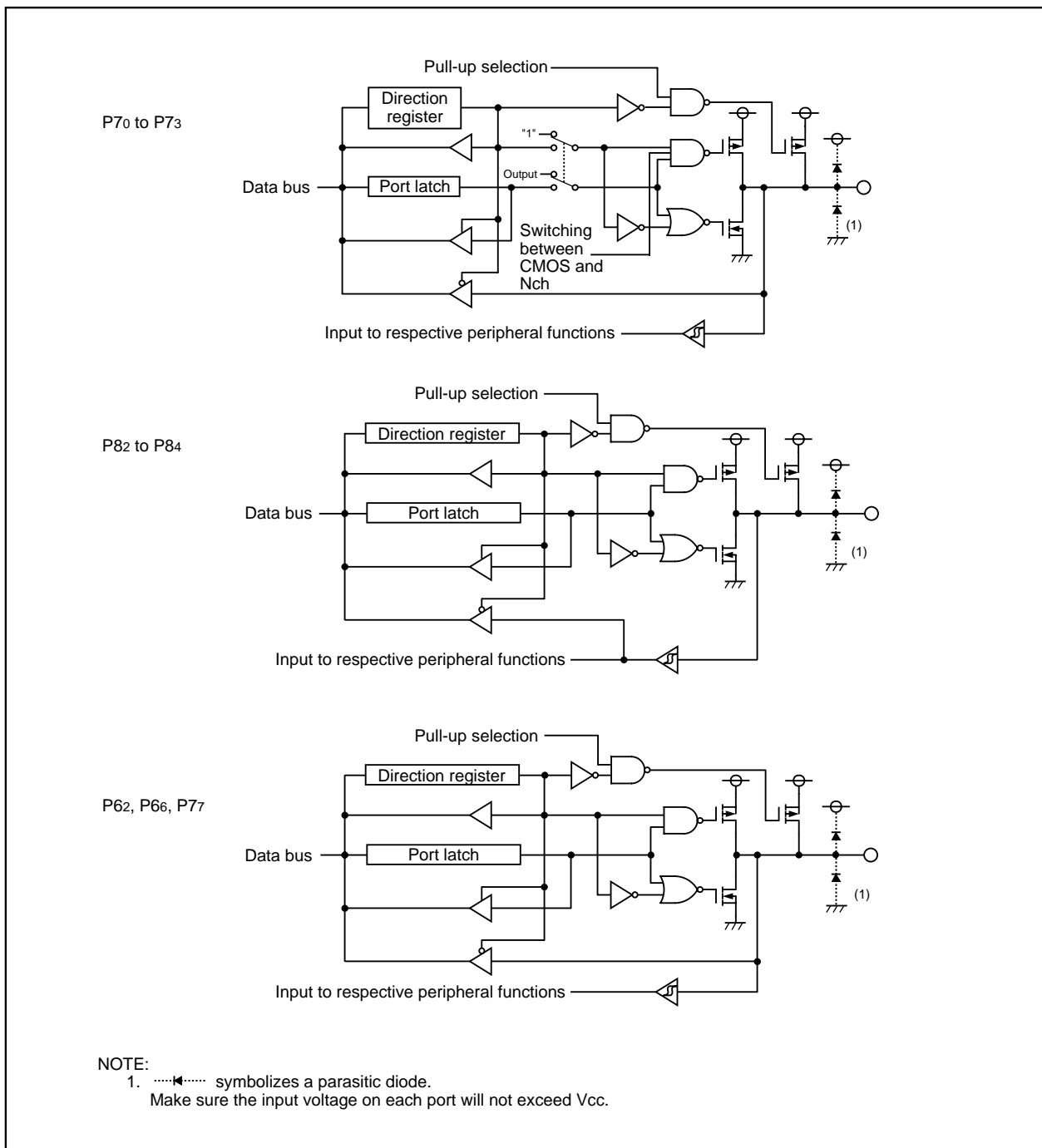


Figure 16.2. I/O Ports (2)

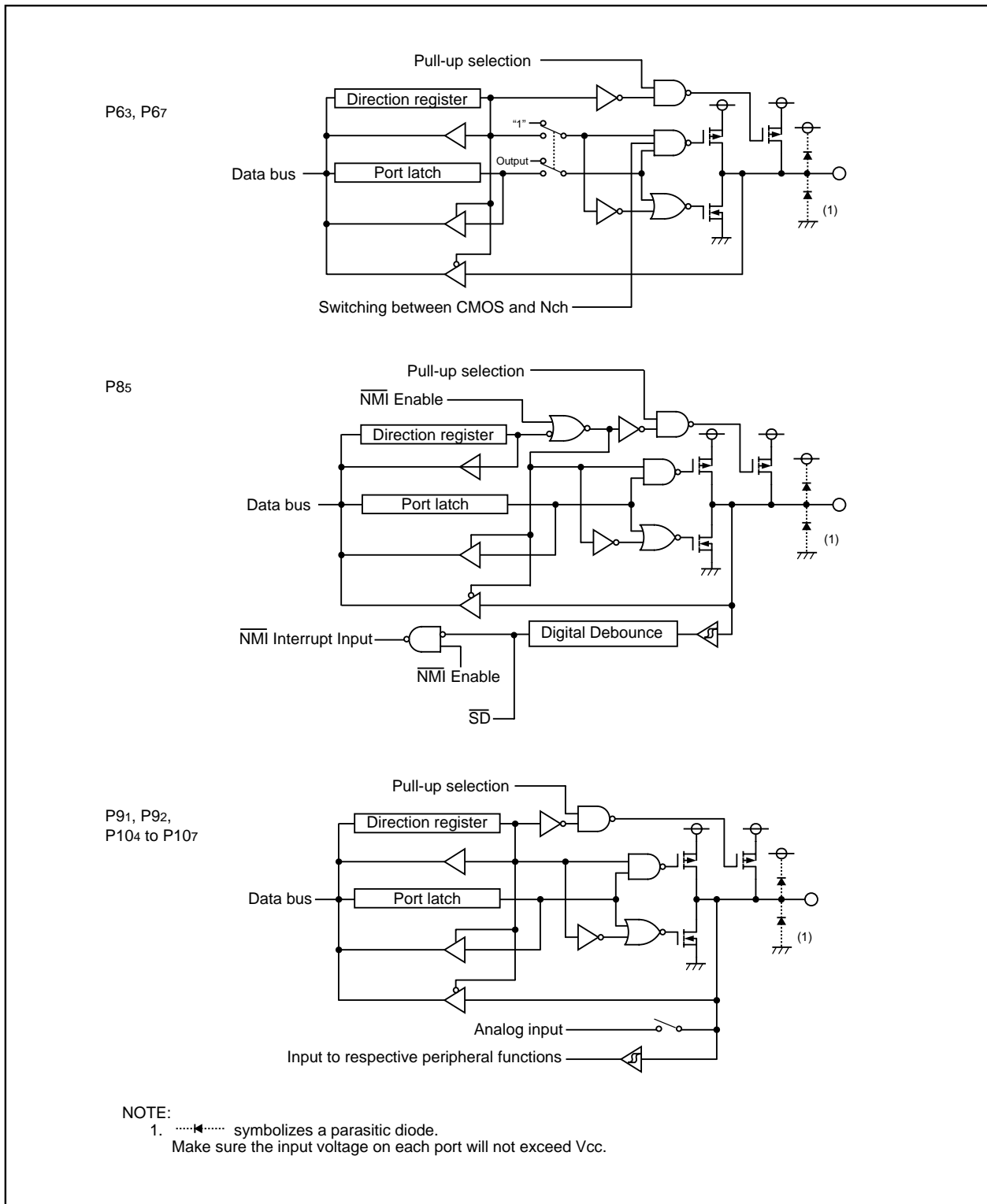


Figure 16.3. I/O Ports (3)

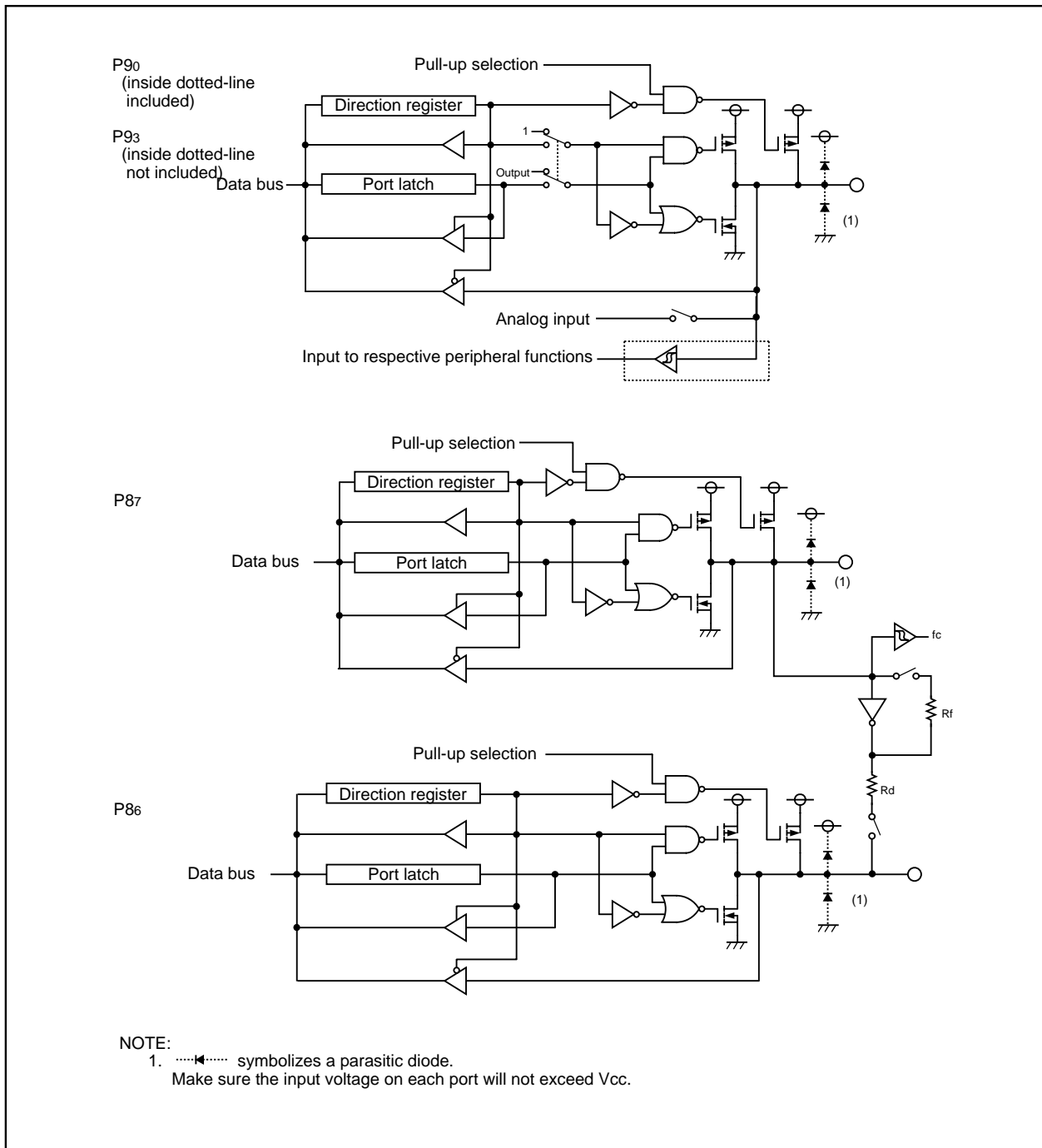
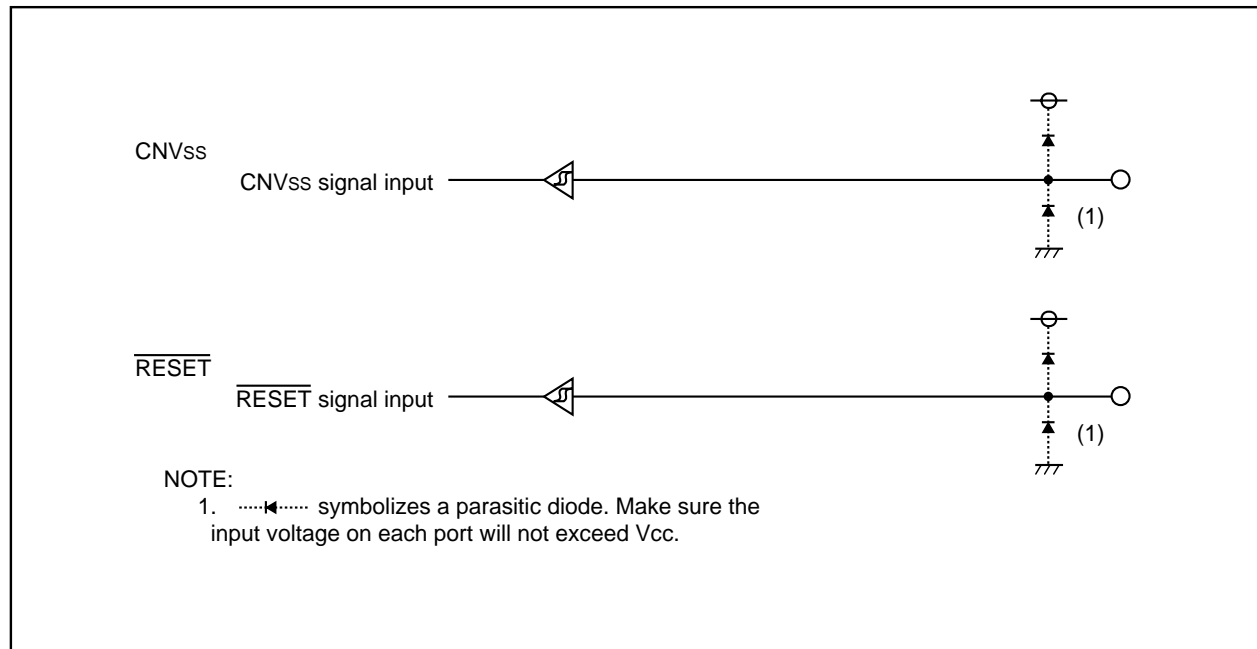


Figure 16.4. I/O Ports (4)

**Figure 16.5. I/O Pins**

## Port Pi direction register (i=6 to 8, and 10) (1)

Symbol	Address	After reset
PD6 to PD8	03EE <sub>16</sub> , 03EF <sub>16</sub> , 03F2 <sub>16</sub>	00 <sub>16</sub>
PD10	03F6 <sub>16</sub>	00 <sub>16</sub>

Bit symbol	Bit name	Function	RW
PD <sub>i</sub> _0	Port Pi0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 6 to 8, and 10)	RW
PD <sub>i</sub> _1	Port Pi1 direction bit		RW
PD <sub>i</sub> _2	Port Pi2 direction bit		RW
PD <sub>i</sub> _3	Port Pi3 direction bit		RW
PD <sub>i</sub> _4	Port Pi4 direction bit		RW
PD <sub>i</sub> _5	Port Pi5 direction bit		RW
PD <sub>i</sub> _6	Port Pi6 direction bit		RW
PD <sub>i</sub> _7	Port Pi7 direction bit		RW

## NOTE:

- Ports must be enabled using the PACR register.  
In 48-pin package, set PACR2, PACR1, PACR0 to "100<sub>2</sub>".  
In 42-pin package, set PACR2, PACR1, PACR0 to "001<sub>2</sub>".

## Port P1 direction register (1)

Symbol	Address	After reset
PD1	03E3 <sub>16</sub>	00 <sub>16</sub>

Bit symbol	Bit name	Function	RW
(b4-b0)	Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		—
PD1_5	Port P15 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD1_6	Port P16 direction bit		RW
PD1_7	Port P17 direction bit		RW

## NOTE:

- Ports must be enabled using the PACR register.  
In 48-pin package, set PACR2, PACR1, PACR0 to "100<sub>2</sub>".  
In 42-pin package, set PACR2, PACR1, PACR0 to "001<sub>2</sub>".

## Port P9 direction register (1,2)

Symbol	Address	After reset
PD9	03F3 <sub>16</sub>	XXXX0000 <sub>2</sub>

Bit symbol	Bit name	Function	RW
PD9_0	Port P90 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD9_1	Port P91 direction bit		RW
PD9_2	Port P92 direction bit		RW
PD9_3	Port P93 direction bit		RW
(B7-b4)	Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		—

## NOTE:

- Make sure the PD9 register is written to by the next instruction after setting the PRC2 bit in the PRCR register to "1" (write enabled).
- Ports must be enabled using the PACR register.

Figure 16.1.1. PD1, PD6, PD7, PD8, PD9, and PD10 Registers



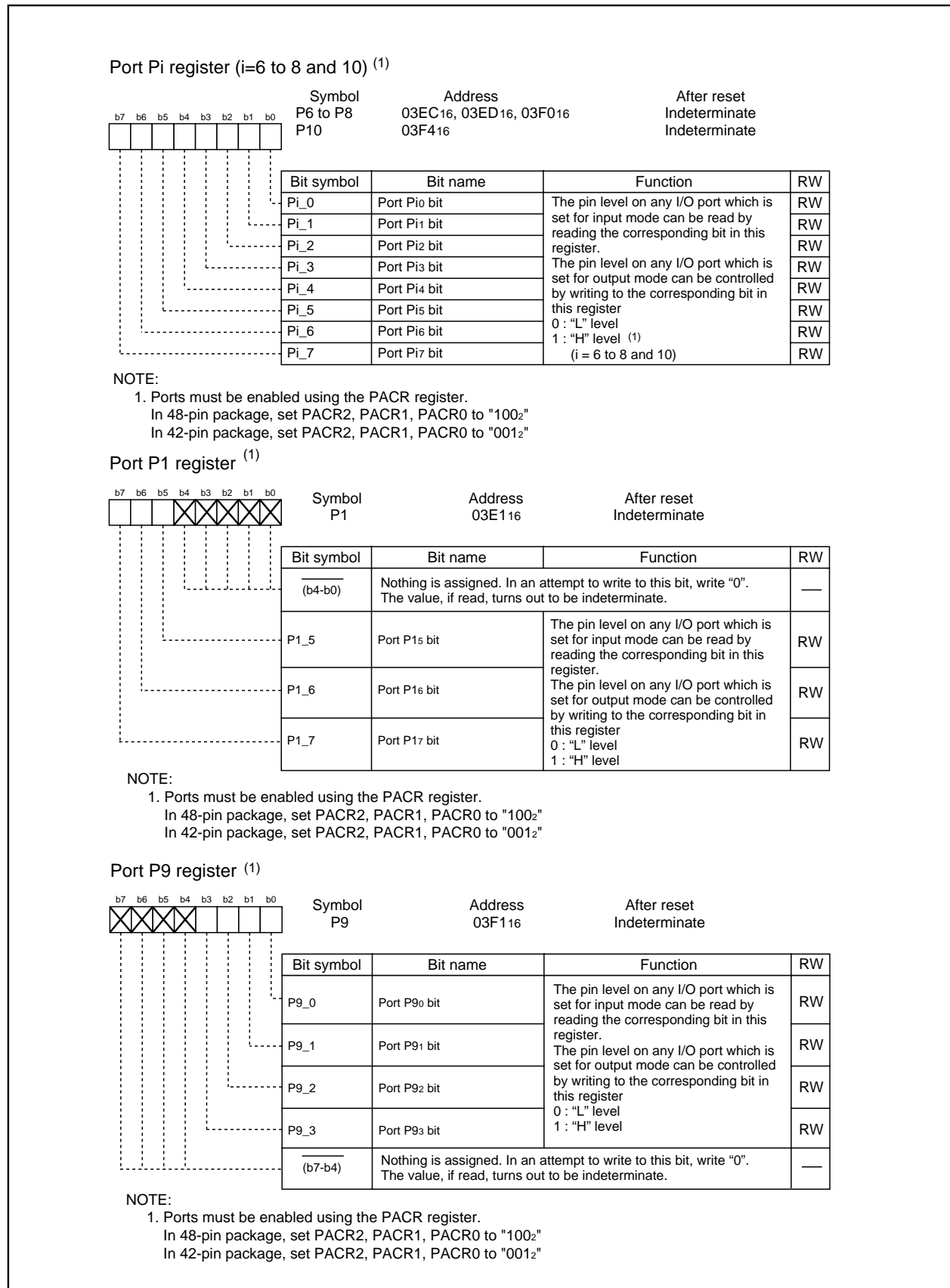


Figure 16.2.1. P1, P6, P7, P8, P9, and P10 Registers

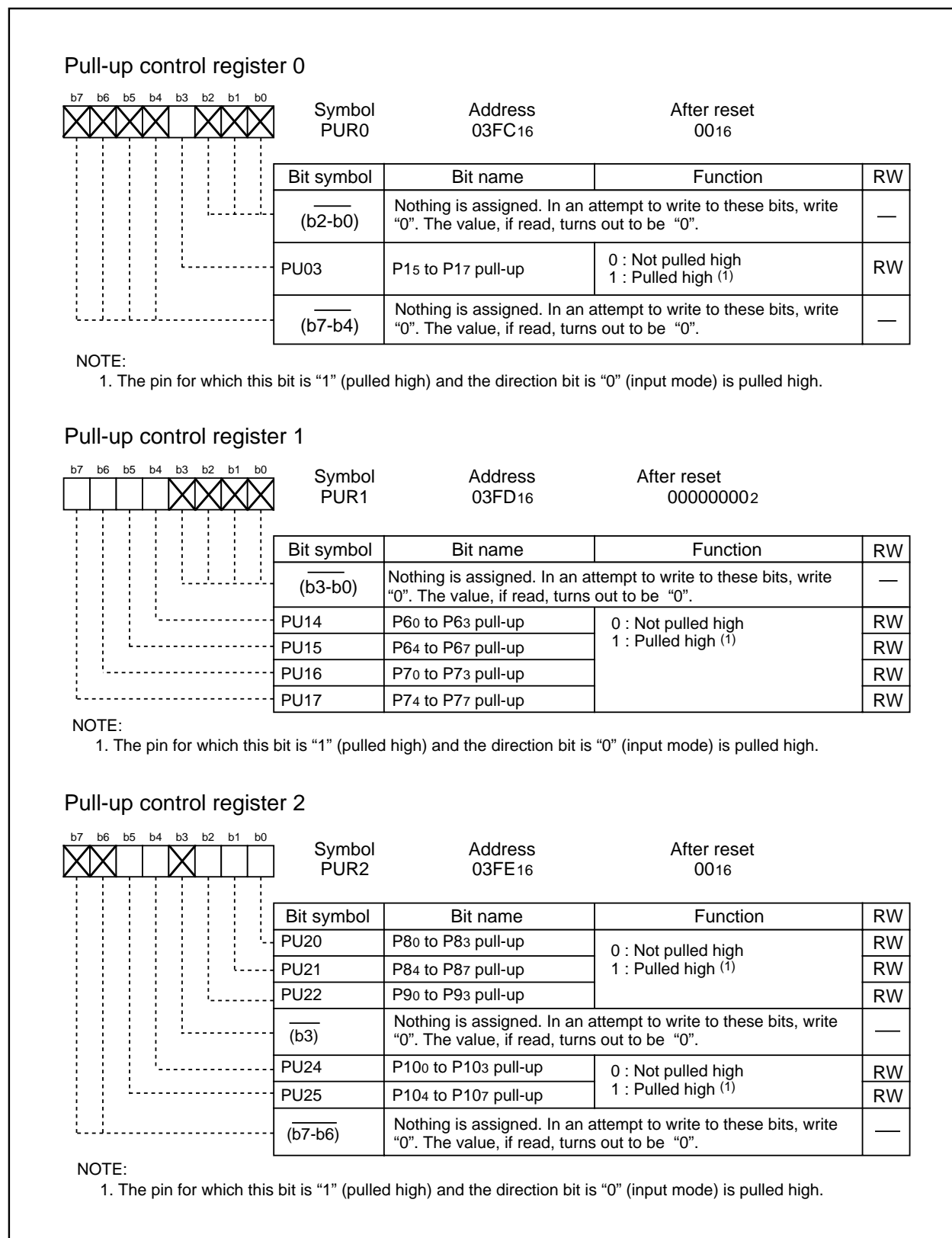


Figure 16.3.1. PUR0 to PUR2 Registers

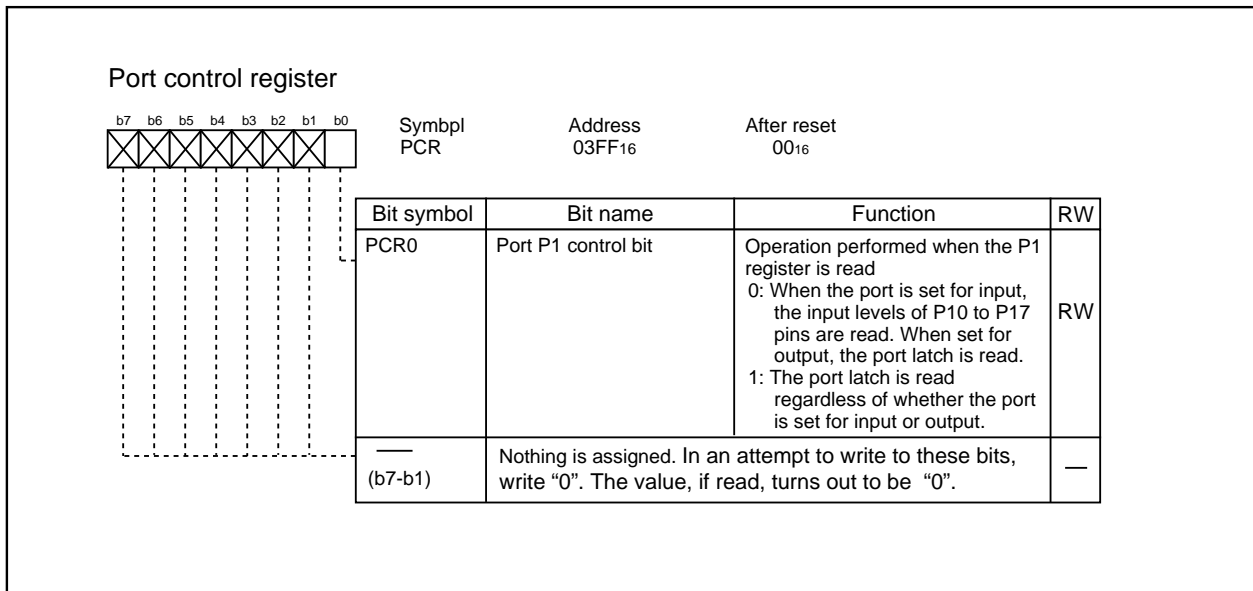


Figure 16.4.1. PCR Register

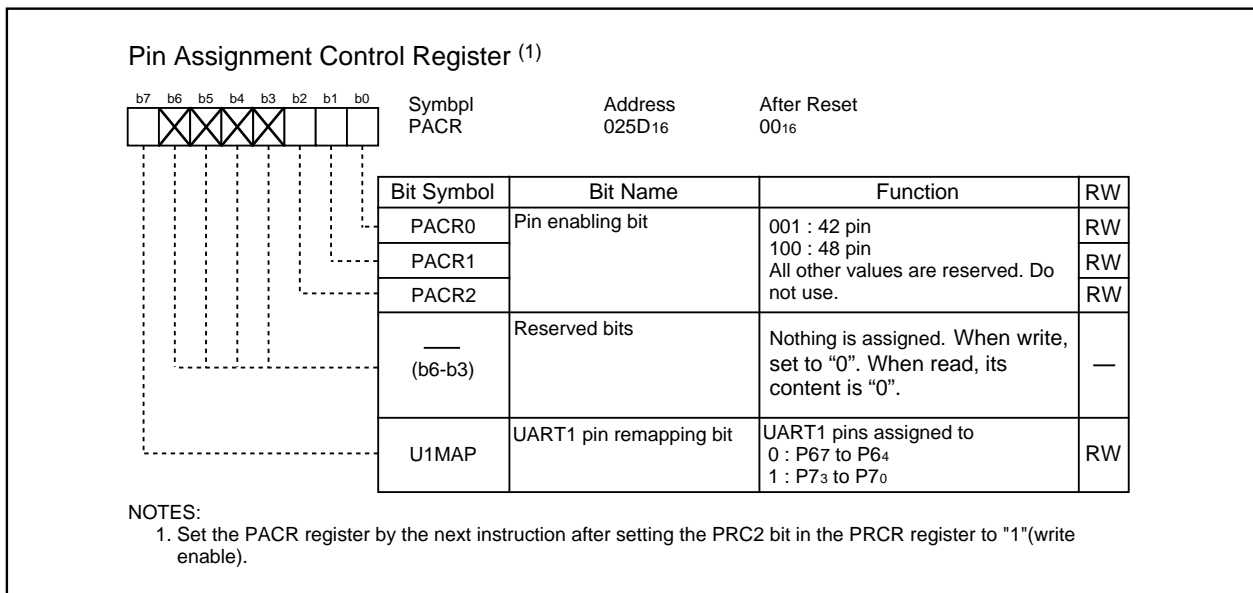


Figure 16.5.1. PACR Register

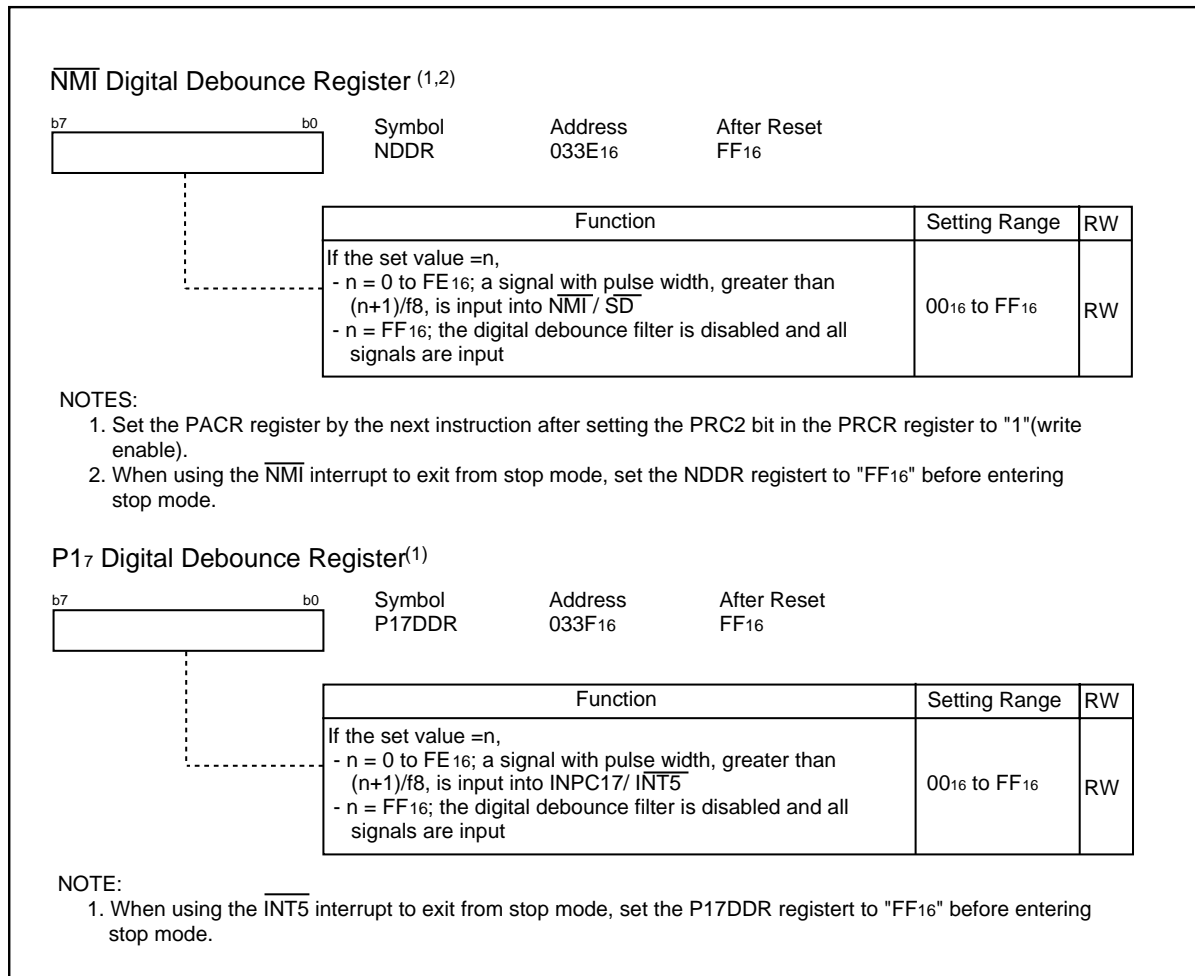
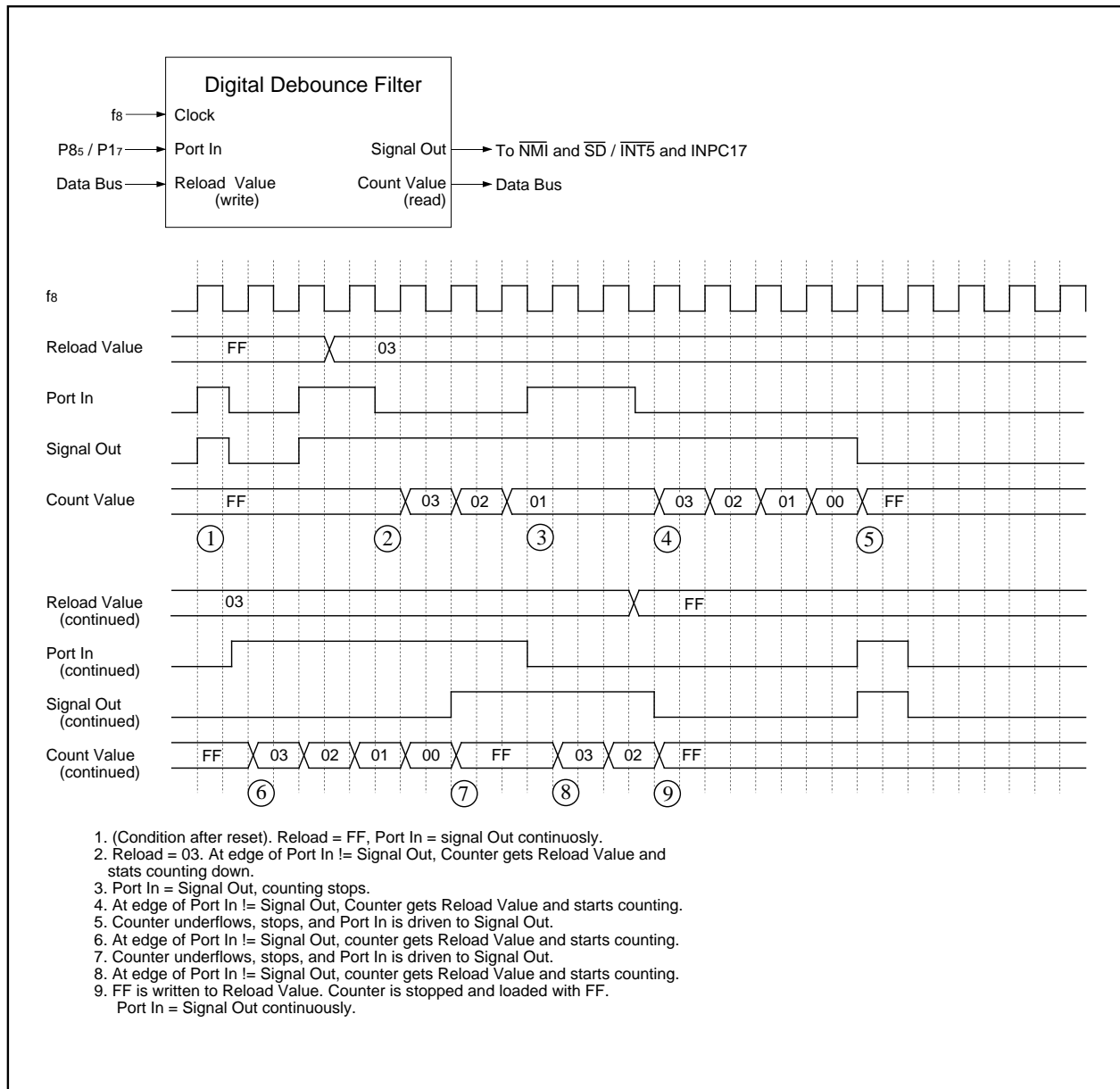


Figure 16.6.1. NDDR and P17DDR Registers



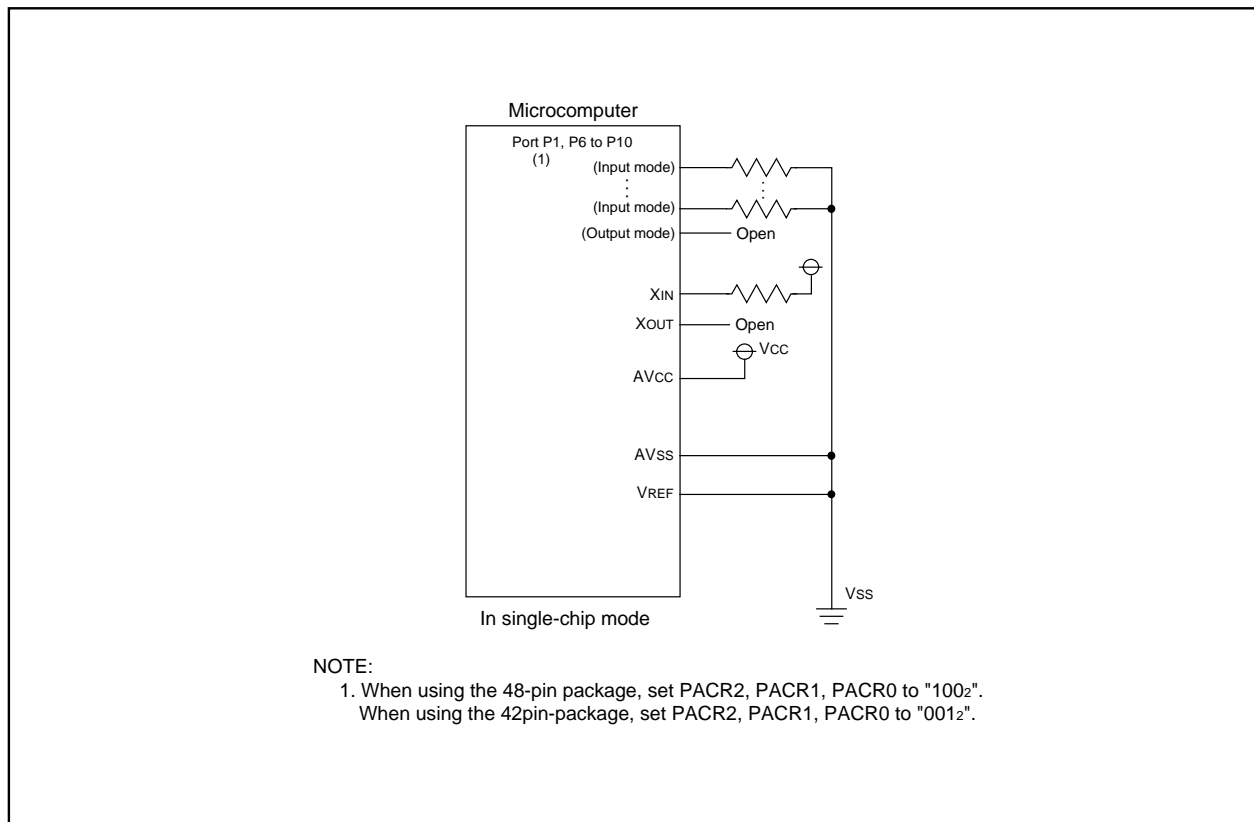
**Figure 16.6.2. Functioning of Digital Debounce Filter**

**Table 16.1. Unassigned Pin Handling in Single-chip Mode**

Pin name	Connection
Ports P1, P6 to P10	After setting for input mode, connect every pin to V <sub>SS</sub> via a resistor(pull-down); or after setting for output mode, leave these pins open. (1, 2, 4)
XOUT (3)	Open
XIN	Connect via resistor to V <sub>CC</sub> (pull-up) (5)
AVCC	Connect to V <sub>CC</sub>
AVSS, VREF	Connect to V <sub>SS</sub>

**NOTES:**

- 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 indeterminate, 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 could be changed by noise or noise-induced runaway, it is recommended that the contents of the direction registers be periodically reset in software, for the increased reliability of the program.
- Make sure the unused pins are processed with the shortest possible wiring from the microcomputer pins (within 2 cm).
- With external clock or VCC input to XIN pin.
- When using the 48-pin package, set PACR2, PACR1, PACR0 to "1002". When using the 42-pin package, set PACR2, PACR1, PACR0 to "0012".
- When the main clock oscillation circuit is not used, set the CM05 bit in the CM0 register to "0" (main clock stops) to reduce power consumption.

**Figure 16.7. Unassigned Pins Handling**

# 17. Flash Memory Version

## 17.1 Flash Memory Performance

The flash memory version is functionally the same as the mask ROM version except that it internally contains flash memory.

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

Table 17.1 shows the flash memory version specifications. (Refer to Table 1.1 or Table 1.2 for the items not listed in Table 17.1.)

**Table 17.1. Flash Memory Version Specifications**

Item		Specification
Flash memory operating mode		3 modes (CPU rewrite, standard serial I/O, parallel I/O)
Erase block		See <b>Figure 17.2.1 to 17.2.3</b> Flash Memory Block Diagram
Program method		In units of word
Erase method		Block erase
Program, erase control method		Program and erase controlled by software command
Protect method		All user blocks are write protected by bit FMR16. In addition, the block 0 and block 1 are write protected by bit FMR02
Number of commands		5 commands
Program/Erase Endurance <sup>(1)</sup>	Block 0 to 3 (program area)	100 times, 1,000 times (See <b>Tables 1.7, 1.9, and 1.10</b> )
	Block A and B (data are) <sup>(2)</sup>	100 times, 10,000 times (See <b>Tables 1.7, 1.9, and 1.10</b> )
Data Retention		20 years (Topr = 55°C)
ROM code protection		Parallel I/O and standard serial I/O modes are supported.

**NOTES:**

1. Program and erase endurance definition Program and erase endurance are the erase endurance of each block. If the program and erase endurance are n times (n=100,1,000,10,000), each block can be erased n times. For example, if a 2-Kbyte block A is erased after writing 1 word data 1024 times, each to different addresses, this is counted as one program and erasure. However, data cannot be written to the same address more than once without erasing the block. (Rewrite disabled)
2. To use the limited number of erasure efficiently, write to unused address within the block instead of rewrite. Erase block only after all possible address are used. For example, an 8-word program can be written 128 times before erase is necessary. Maintaining an equal number of erasure between Block A and B will also improve efficiency. We recommend keeping track of the number of times erasure is used.

**Table 17.2. Flash Memory Rewrite Modes Overview**

Flash memory rewrite mode	CPU rewrite mode	Standard serial I/O mode	Parallel I/O mode
Function	The user ROM area is rewritten when the CPU executes software command EW0 mode: Rewrite in area other than flash memory EW1 mode: Rewrite in flash memory	The user ROM area is rewritten using a dedicated serial programmer. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART	The user ROM area is rewritten using a dedicated parallel programmer
Area which can be rewritten	User ROM area	User ROM area	User ROM area
Operation mode	Single chip mode	Boot mode	Parallel I/O mode
ROM programmer	None	Serial programmer	Parallel programmer

### 17.1.1 Boot Mode

The MCU enters boot mode when a hardware reset is performed while a high-level ("H") signal is applied to pins CNVss and P86 or while an "H" signal is applied to pins CNVss and P16 and a low-level ("L") signal is applied to the P85. A program in the boot ROM area is executed.

The boot ROM area is reserved. The boot ROM area stores the rewrite control program for a standard serial I/O mode before shipping. Do not rewrite the boot ROM area.



## 17.2 Memory Map

The flash memory contains the user ROM area and the boot ROM area (reserved area). Figures 17.2.1 to 17.2.3 show the flash memory block diagram. The user ROM area has space to store the microcomputer operation program in single-chip mode and a separate 2-Kbyte space as the block A and B.

The user ROM area is divided into several blocks. The user ROM area can be rewritten in CPU rewrite, standard serial input/output, and parallel input/output modes. However, if block 0 and 1 are rewritten in CPU rewrite mode, setting the FMR02 bit in the FMR0 register to "1" (block 0, 1 rewrite enabled) and the FMR16 bit in the FMR1 register to "1" (blocks 0 to 3 rewrite enabled) enable rewriting. Also, if blocks 2 to 3 are rewritten in CPU rewrite mode, setting the FMR16 bit in the FMR1 register to "1" (blocks 0 to 3 rewrite enabled) enables writing. Setting the PM10 bit in the PM1 register to "1" (data area access enabled) for block A and B enables to use.

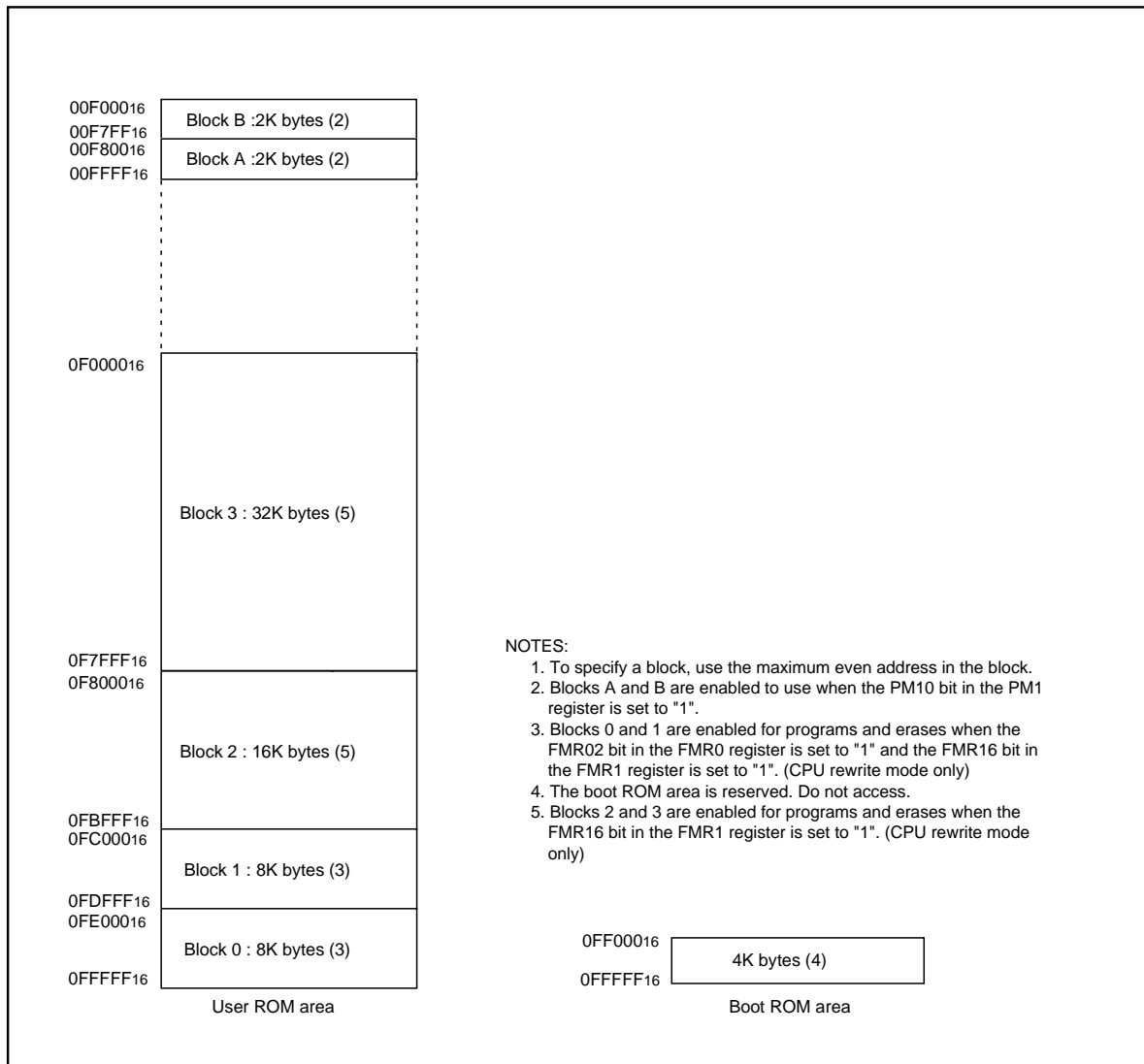
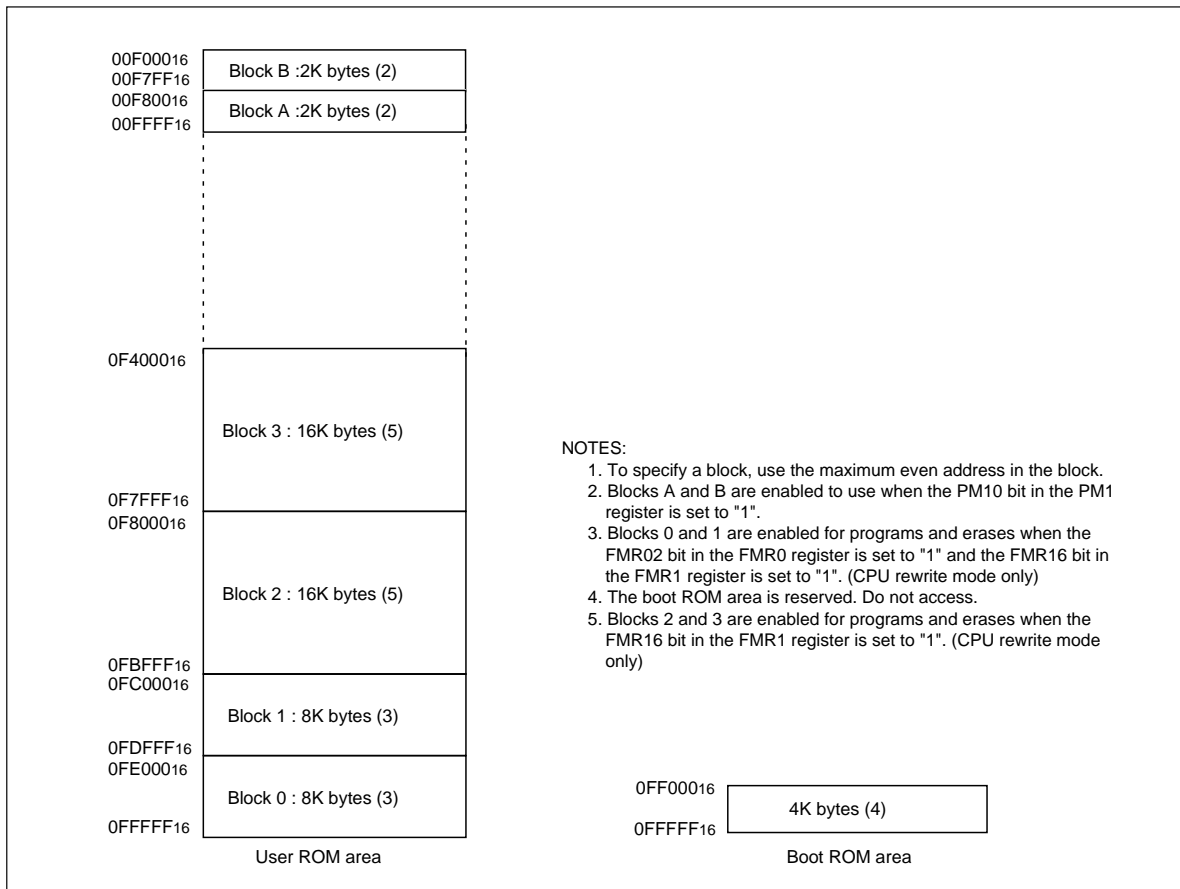
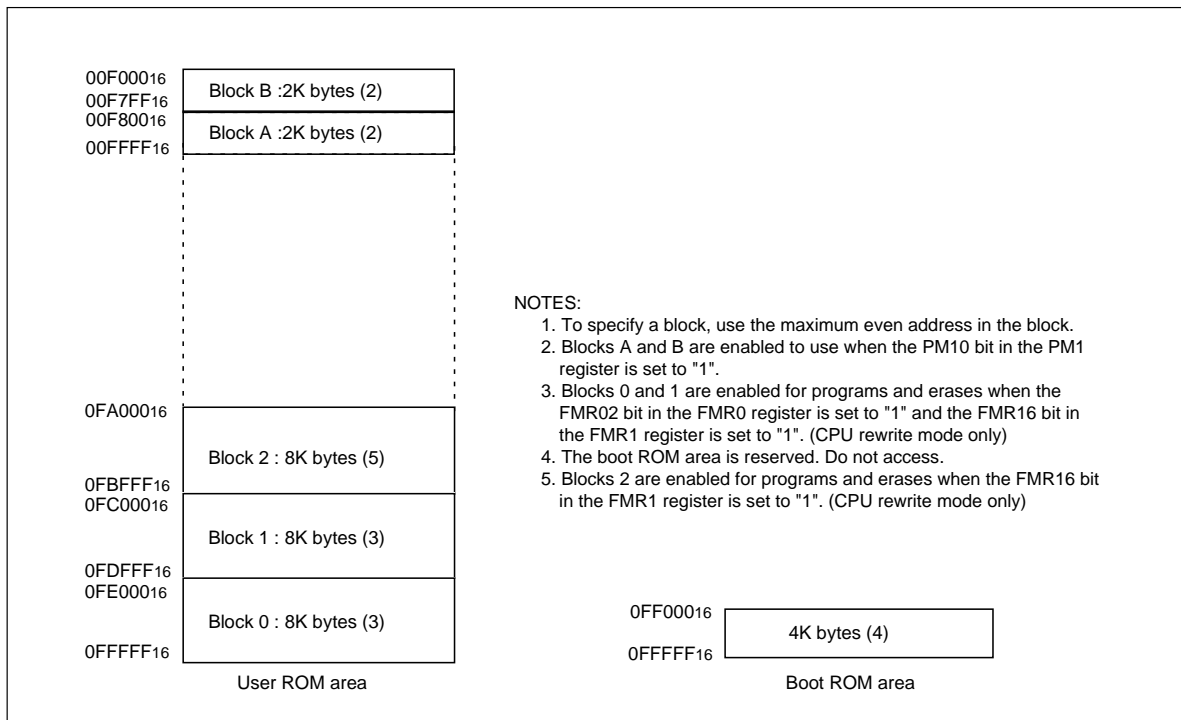


Figure 17.2.1. Flash Memory Block Diagram (ROM capacity 64K byte)



**Figure 17.2.2. Flash Memory Block Diagram (ROM capacity 48K byte)**



**Figure 17.2.3. Flash Memory Block Diagram (ROM capacity 24K byte)**

## 17.3 Functions To Prevent Flash Memory from Rewriting

The flash memory has a built-in ROM code protect function for parallel I/O mode and a built-in ID code check function for standard input/output mode to prevent the flash memory from reading or rewriting.

### 17.3.1 ROM Code Protect Function

The ROM code protect function disables reading or changing the contents of the on-chip flash memory in parallel I/O mode. **Figure 17.3.1.1** shows the ROMCP address. The ROMCP address is located in a user ROM area.

To enable ROM code protect, set the ROMCP1 bit to “002”, “012”, or “102” and set the bit 5 to bit 0 to “1111112”.

To cancel ROM code protect, erase the block including the the ROMCP address in CPU rewrite mode or standard serial I/O mode.

### 17.3.2 ID Code Check Function

Use the ID code check function in standard serial input/output mode. Unless the flash memory is blank, the ID codes sent from the programmer and the seven bytes ID codes written in the flash memory are compared to see if they match. If the ID codes do not match, the commands sent from the programmer are not acknowledged. The ID code consists of 8-bit data, starting with the first byte, into addresses, 0FFFFDF<sub>16</sub>, 0FFFE3<sub>16</sub>, 0FFFE<sub>16</sub>, 0FFFEF<sub>16</sub>, 0FFFF3<sub>16</sub>, 0FFFF7<sub>16</sub>, and 0FFFFB<sub>16</sub>. The flash memory has a program with the ID code set in these addresses.

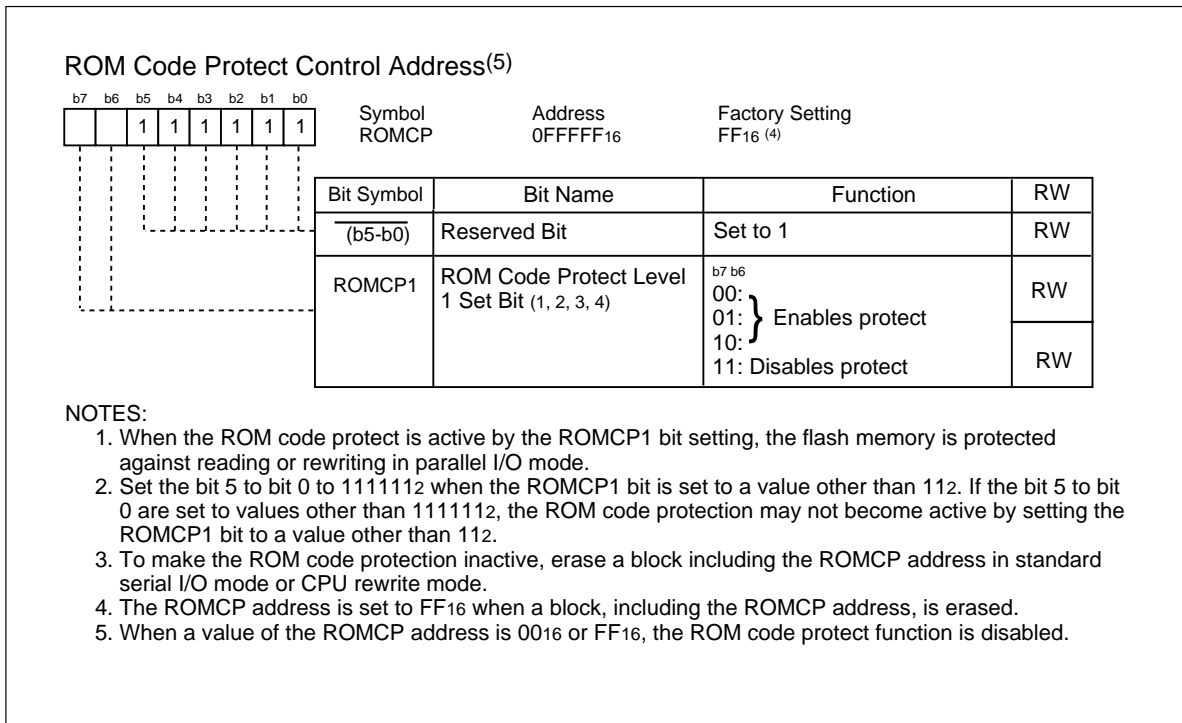


Figure 17.3.1.1. ROMCP Address

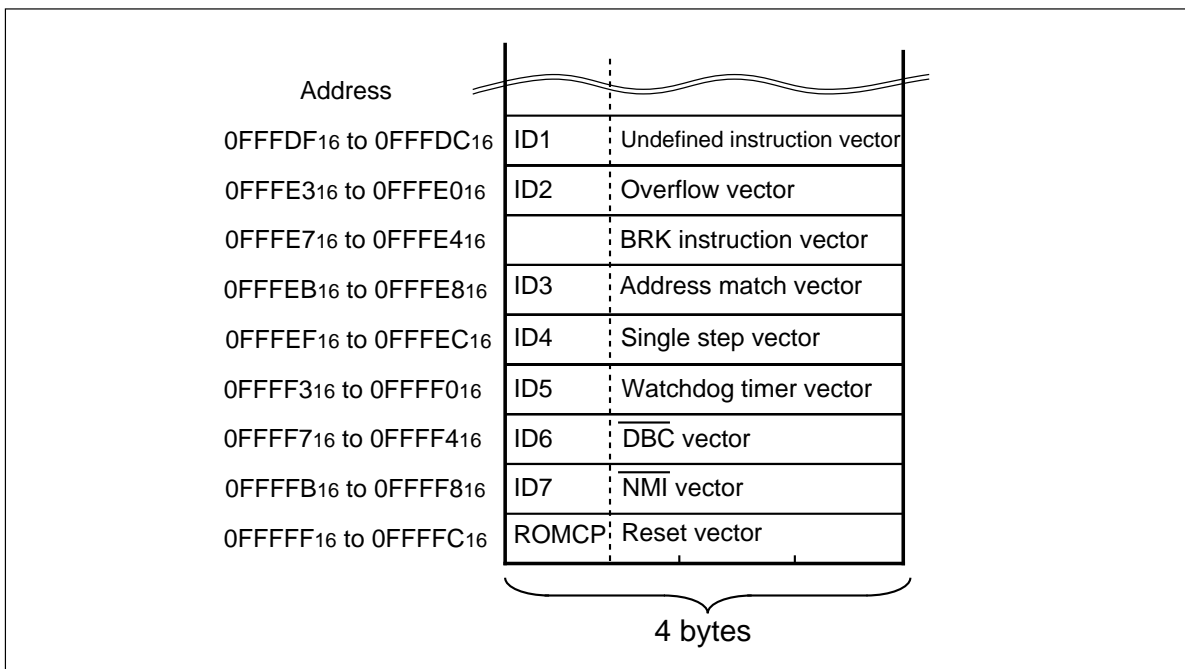


Figure 17.3.2.1. Address for ID Code Stored

## 17.4 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten when the CPU executes software commands. Therefore, the user ROM area can be rewritten directly while the microcomputer is mounted on-board without using a ROM programmer, etc. Verify the Program and the Block Erase commands are executed only on blocks in the user ROM area.

For interrupts requested during an erasing operation in CPU rewrite mode, the M16C/26A Group flash module offers an erase-suspend function which the erasing operation to be suspended, and access made available to the flash. Erase-write 0 (EW0) mode and erase-write 1 (EW1) mode are provided as CPU rewrite mode. Table 17.4.1 shows the differences between erase-write 0 (EW0) and erase-write 1 (EW1) modes. 1 wait is required for the CPU erase-write control.

Table 17.4.1. EW0 Mode and EW1 Mode

Item	EW0 mode	EW1 mode
Operation mode	Single chip mode	Single chip mode
Area where rewrite control program can be placed	User ROM area	User ROM area
Area where rewrite control program can be executed	The rewrite control program must be transferred to any area other than the flash memory (e.g., RAM) before being executed	The rewrite control program can be executed in the user ROM area
Area which can be rewritten	User ROM area	User ROM area However, this excludes blocks with the rewrite control program
Software command Restrictions	None	<ul style="list-style-type: none"> <li>• Program, block erase command Cannot be executed in a block having the rewrite control program</li> <li>• Read status register command Can not be used</li> </ul>
Mode after programming or erasing	Read Status Register mode	Read Array mode
CPU state during auto-write and auto-erase	Operation	Hold state (I/O ports retain the state before the command is executed) (1)
Flash memory status detection (2)	<ul style="list-style-type: none"> <li>• Read the FMR00, FMR06 and FMR07 bits in the FMR0 register by a program</li> <li>• Execute the read status register command and read the SR7, SR5 and SR4 bits</li> </ul>	Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program
Condition for transferring to erase-suspend (3)	Set the FMR40 and FMR41 bits in the FMR4 register to "1" by program.	The FMR40 bit in the FMR4 register is set to "1" and the interrupt request of

### NOTES:

1. Do not generate a DMA transfer.
2. Block 1 and 0 are enabled to rewrite by setting the FMR02 bit in the FMR0 register to "1" and setting the FMR16 bit in the FMR1 register to "1". Block 2 to 3 are enabled to rewrite by setting the FMR16 bit in the FMR1 register to "1".
3. The time, until entering erase suspend and reading flash is enabled, is maximum td (SR-ES) after satisfying the conditions.

### 17.4.1 EW0 Mode

The microcomputer enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to "1" (CPU rewrite mode enabled) and is ready to acknowledge the software commands. EW0 mode is selected by setting the FMR11 bit in the FMR1 register to "0".

When setting the FMR01 bit to "1", set to "1" after first writing "0". The software commands control programming and erasing. The FMR0 register or the status register indicates whether a programming or erasing operations is completed.

When entering the erase-suspend during the auto-erasing, set the FMR40 bit to "1" (erase-suspend enabled) and the FMR41 bit to "1" (suspend request). And wait for td(SR-ES). After verifying the FMR46 bit is set to "1" (auto-erase stop), access to the user ROM area. When setting the FMR41 bit to "0" (erase restart), auto-erasing is restarted.

### 17.4.2 EW1 Mode

EW1 mode is selected by setting the FMR11 bit to "1" after the FMR01 bit is set to "1". (set to "1" after first writing "0"). The FMR0 register indicates whether or not a programming or an erasing operation is completed. Do not execute the software commands of read status register 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.

When enabling an erase suspend function, set the FMR40 bit to "1" (erase suspend enabled) and execute block erase commands. Also, preliminarily set an interrupt to enter the erase-suspend to an interrupt enabled status. After td(SR-ES) from an interrupt request and entering erase suspend, an interrupt can be acknowledged.

When an interrupt request is generated, the FMR41 bit is automatically set to "1" (suspend request) and an auto-erasing is halted. If an auto-erasing is not completed (the FMR00 bit is "0") after an interrupt process completed, set the FMR41 bit to "0" (erase restart) and execute block erase commands again.

## 17.5 Register Description

Figure 17.5.1 shows the flash memory control register 0 and flash memory control register 1. Figure 17.5.2 shows the flash memory control register 4.

### 17.5.1 Flash memory control register 0 (FMR0)

- FMR 00 Bit

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

- FMR01 Bit

The microcomputer enables to acknowledge commands by setting the FMR01 bit to “1” (CPU rewrite mode). To set this bit to “1”, it is necessary to set to “1” after first setting to “0”. Set this bit to “0” by only writing “0”.

- FMR02 Bit

The combined setting of the FMR02 bit and the FMR16 bit enable to program and erase in the user ROM area. See Table 17.5.2.1 for setting details. To set this bit to “1”, it is necessary to set to “1” after first setting to “0”. Set this bit to “0” by only writing “0”. This bit is enabled only when the FMR01 bit is “1” (CPU rewrite mode enable).

- FMSTP Bit

This 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”. Set the FMSTP bit by a program in a space other than the flash memory.

Set the FMSTP bit to “1” if one of the following occurs:

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

Figure 17.5.1.3 shows a flow chart illustrating how to start and stop the flash memory before and after entering low power mode. Follow the procedure on this flow chart.

To enter stop or wait mode when CPU rewrite mode is disabled, do not set the FMR0 register. The flash memory is automatically turned off when entering and turned back on when exiting.

- FMR06 Bit

This is a read-only bit indicating an auto-program operation status. This bit is set to “1” when a program error occurs; otherwise, it is set to “0”. For details, refer to **17.8.4 Full Status Check**.

- FMR07 Bit

This is a read-only bit indicating an auto-erase operation status. The bit is set to “1” when an erase error occurs; otherwise, it is set to “0”. For details, refer to **17.8.4 Full Status Check**.

Figure 17.5.1.1 shows a EW0 mode set/reset flowchart, figure 17.5.1.2 shows a EW1 mode set/reset flowchart.



### 17.5.2 Flash memory control register 1 (FMR1)

- FMR11 Bit

EW1 mode is entered by setting the FMR11 bit to “1” (EW1 mode). This bit is enabled only when the FMR01 bit is “1”.

- FMR16 Bit

The combined setting of the FMR02 bit and the FMR16 bit enables to program and erase in the user ROM area. To set this bit to “1”, it is necessary to set to “1” after first setting to “0”. Set this bit to “0” by only writing “0”. This bit is enabled only when the FMR01 bit is “1”.

- FMR17 Bit

If FMR17 bit is “1” (with wait state), regardless of the content of the PM17 bit, 1 wait is inserted at the access to block A and block B. Regardless of the content of the FMR17 bit, access to other block and the internal RAM is determined by PM17 bit setting.

Set this bit to “1” (with wait state) when rewriting more than 100 times (U7 and U9).

**Table 17.5.2.1. Protection using FMR16 and FMR02**

FMR16	FMR02	Block A, Block B	Block 0, Block 1	other user block
0	0	write enabled	write disabled	write disabled
0	1	write enabled	write disabled	write disabled
1	0	write enabled	write disabled	write enabled
1	1	write enabled	write enabled	write enabled

### 17.5.3 Flash memory control register 4 (FMR4)

- FMR40 Bit

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

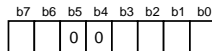
- FMR41 Bit

When setting the FMR41 bit to “1” in a program during auto-erasing in EW0 mode the flash module enters erase suspend mode. In EW1 mode, the FMR41 bit is automatically set to “1” (suspend request) when an interrupt request of an enabled interrupt is generated, the FMR41 bit is automatically set to “1” (suspend request) and when an auto-erasing operation is restarted, set the FMR41 bit to “0” (erase restart).

- FMR46 Bit

The FMR46 bit is set to “0” during auto-erasing execution and set to “1” during erase-suspend mode. Do not access to flash memory while this bit is “0”.

## Flash memory control register 0



Symbol FMR0 Address 01B716 After reset 00000012

Bit symbol	Bit name	Function	RW
FMR00	RY/B $\bar{Y}$ status flag	0: Busy (during writing or erasing) 1: Ready	RO
FMR01	CPU rewrite mode select bit (1)	0: Disables CPU rewrite mode (Disables software command) 1: Enables CPU rewrite mode (Enables software commands)	RW
FMR02	Block 0, 1 rewrite enable bit (2)	Set write protection for user ROM area (see Table 17.5.2.1)	RW
FMSTP	Flash memory stop bit (3, 5)	0: Starts flash memory operation 1: Stops flash memory operation (Enters low-power consumption state and flash memory reset)	RW
(b5-b4)	Reserved bit	Set to "0"	RW
FMR06	Program status flag (4)	0: Terminated normally 1: Terminated in error	RO
FMR07	Erase status flag (4)	0: Terminated normally 1: Terminated in error	RO

## NOTES:

- When setting this bit to "1", set to "1" immediately after setting it first to "0". Do not generate an interrupt or a DMA transfer between setting the bit to "0" and setting it to "1". Set this bit while the P85/NMI/SD pin is "H" when selecting the NMI function. Set by program in a space other than the flash memory in EW0 mode. Set this bit to read alley mode and "0"
- Set this bit to "1" immediately after setting it first to "0" while the FMR01 bit is set to "1". Do not generate an interrupt or a DMA transfer between setting this bit to "0" and setting it to "1".
- Set this bit in a pace other than the flash memory by program. When this bit is set to 1, access to flash memory will be denied. To set this bit to 0 after setting it to 1, wait for 10  $\mu$ sec. or more after setting it to 1. To read data from flash memory after setting this bit to 0, maintain tps wait time before accessing flash memory.
- This bit is set to "0" by executing the clear status command.
- This bit is enabled when the FMR01 bit is set to "1" (CPU rewrite mode). This bit can be set to "1" when the FMR01 bit is set to "0". However, the flash memory does not enter low-power consumption status

## Flash memory control register 1



Symbol FMR1 Address 01B516 After reset 000XX0X2

Bit symbol	Bit name	Function	RW
(b0)	Reserved bit	When read, its content is indeterminate	RO
FMR11	EW1 mode select bit (1)	0: EW0 mode 1: EW1 mode	RW
(b3-b2)	Reserved bit	When read, its content is indeterminate	RO
(b4)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
(b5)	Reserved bit	Set to "0"	RW
FMR16	Block 0 to 3 rewrite enable bit (2)	Set write protection for user ROM area (see Table 17.5.2.1) 0: Disable 1: Enable	RW
FMR17	Block A, B access wait bit (3)	0: PM17 enabled 1: With wait state (1 wait)	RW

## NOTES:

- Set this bit to "1" immediately after setting it first to "0". Do not generate an interrupt or a DMA transfer between setting the bit to "0" and setting it to "1". Set this bit while the P85/NMI/SD pin is "H" when the NMI function is selected. If the FMR01 bit is set to "0", the FMR01 bit and FMR11 bit are both set to "0"
- Set this bit to "1" immediately after setting it first to "0". Do not generate an interrupt or a DMA transfer after setting to "0".
- When rewriting more than 100 times, set this bit to "1" (with wait state). When the FMR17 bit is "1" (with wait state), regardless of the content of the PM17 bit, 1 wait is inserted at the access to the block A and B. Regardless of the content of the FMR17 bit, access to other block and the internal RAM is determined by PM17 bit setting.

Figure 17.5.1. FMR0 and FMR1 register

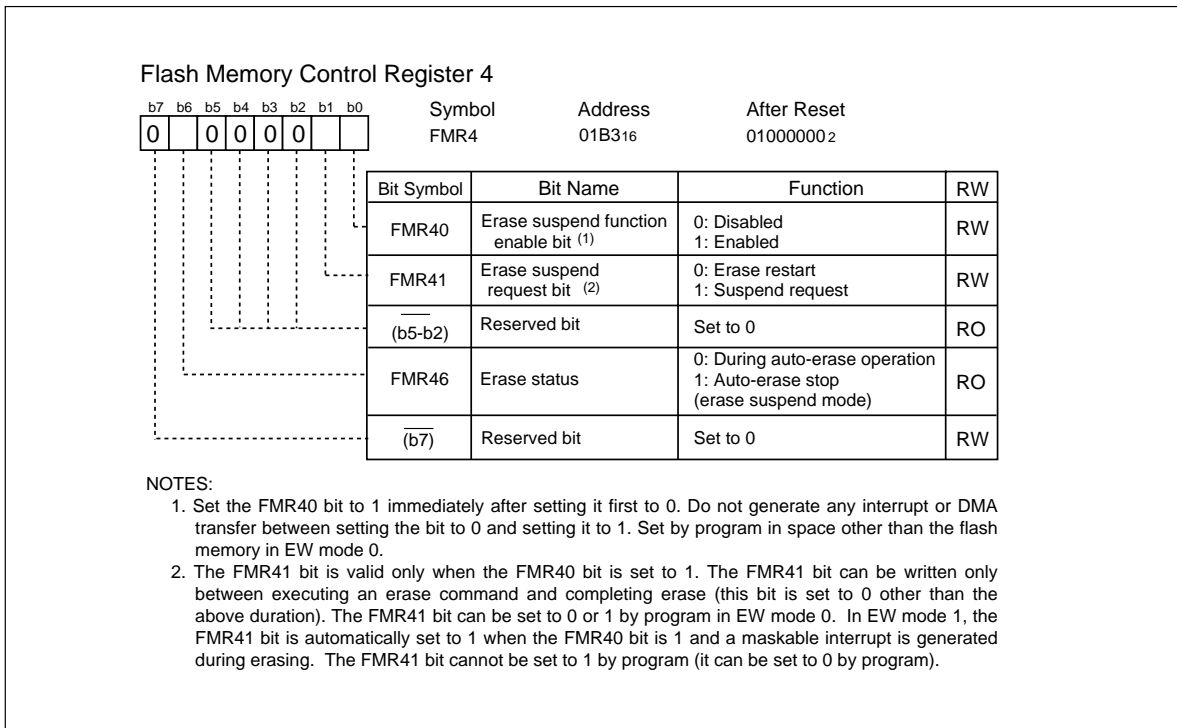


Figure 17.5.2. FMR4 register

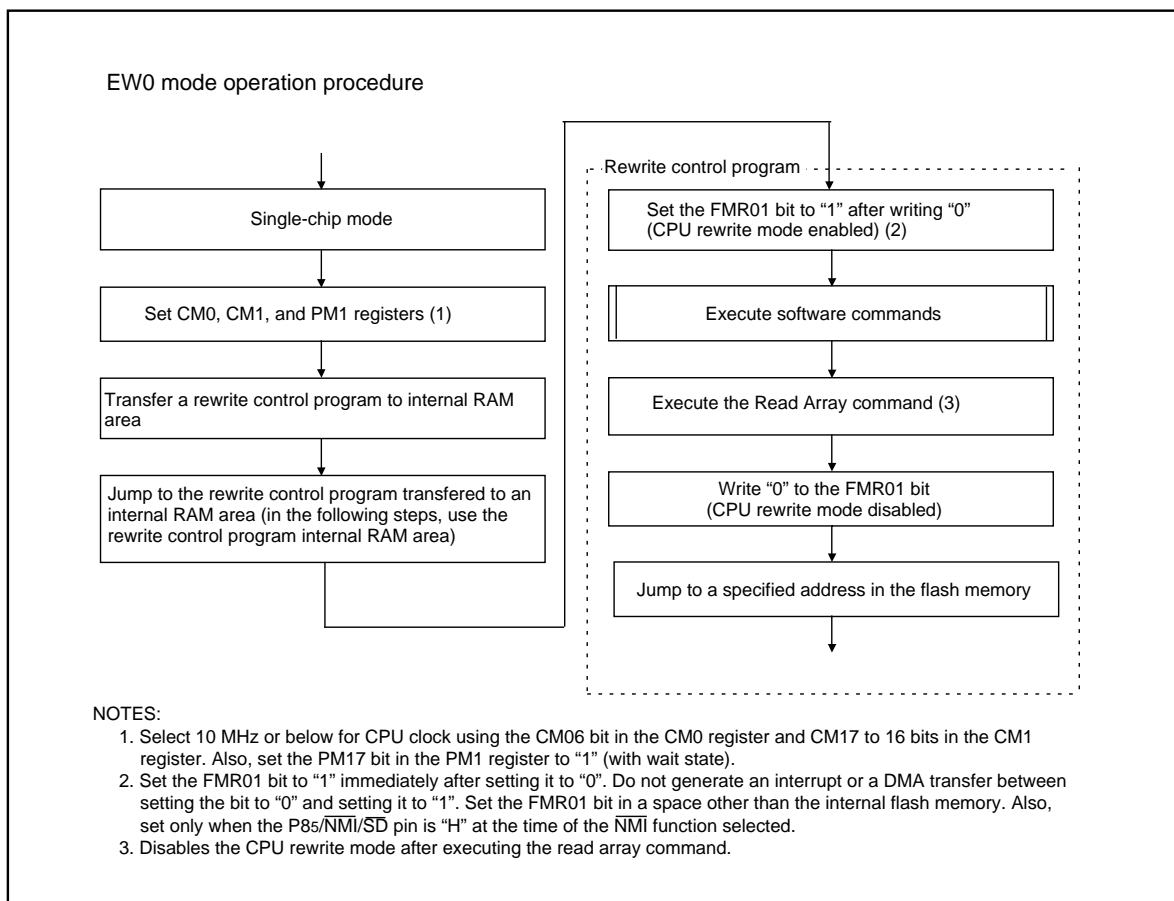


Figure 17.5.1.1. Setting and Resetting of EW0 Mode

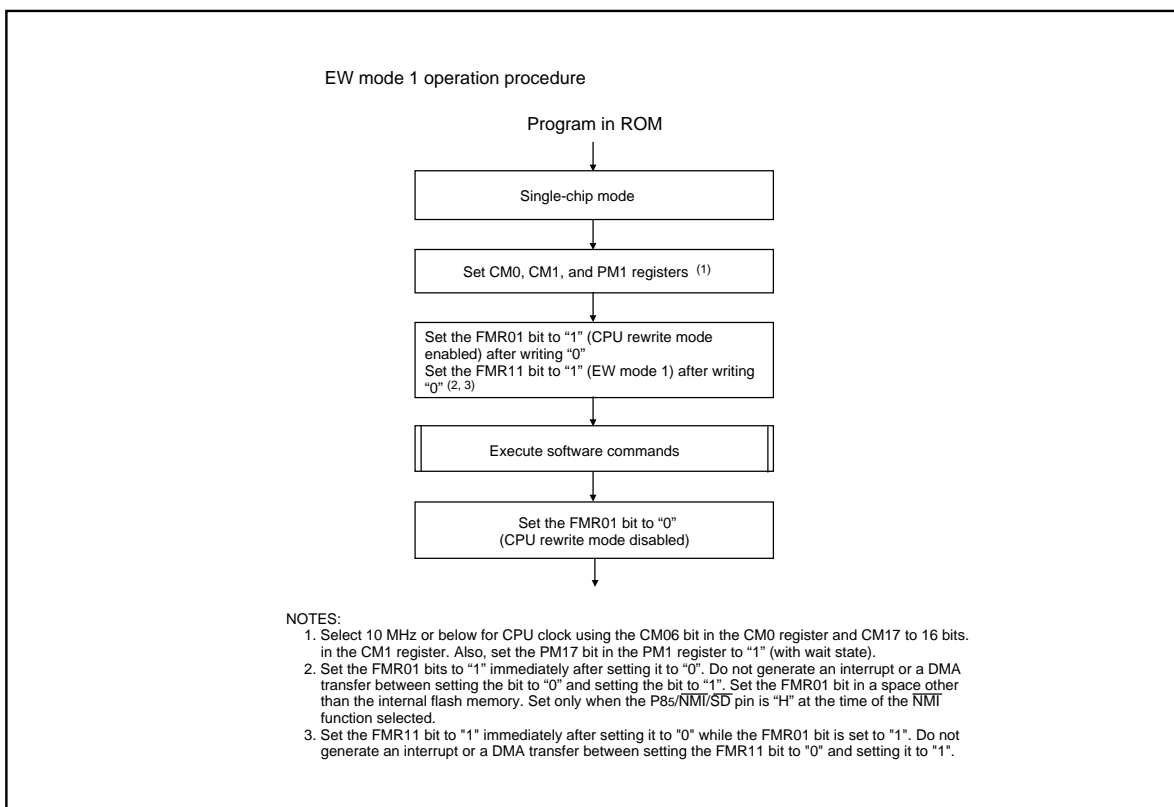


Figure 17.5.1.2. Setting and Resetting of EW1 Mode

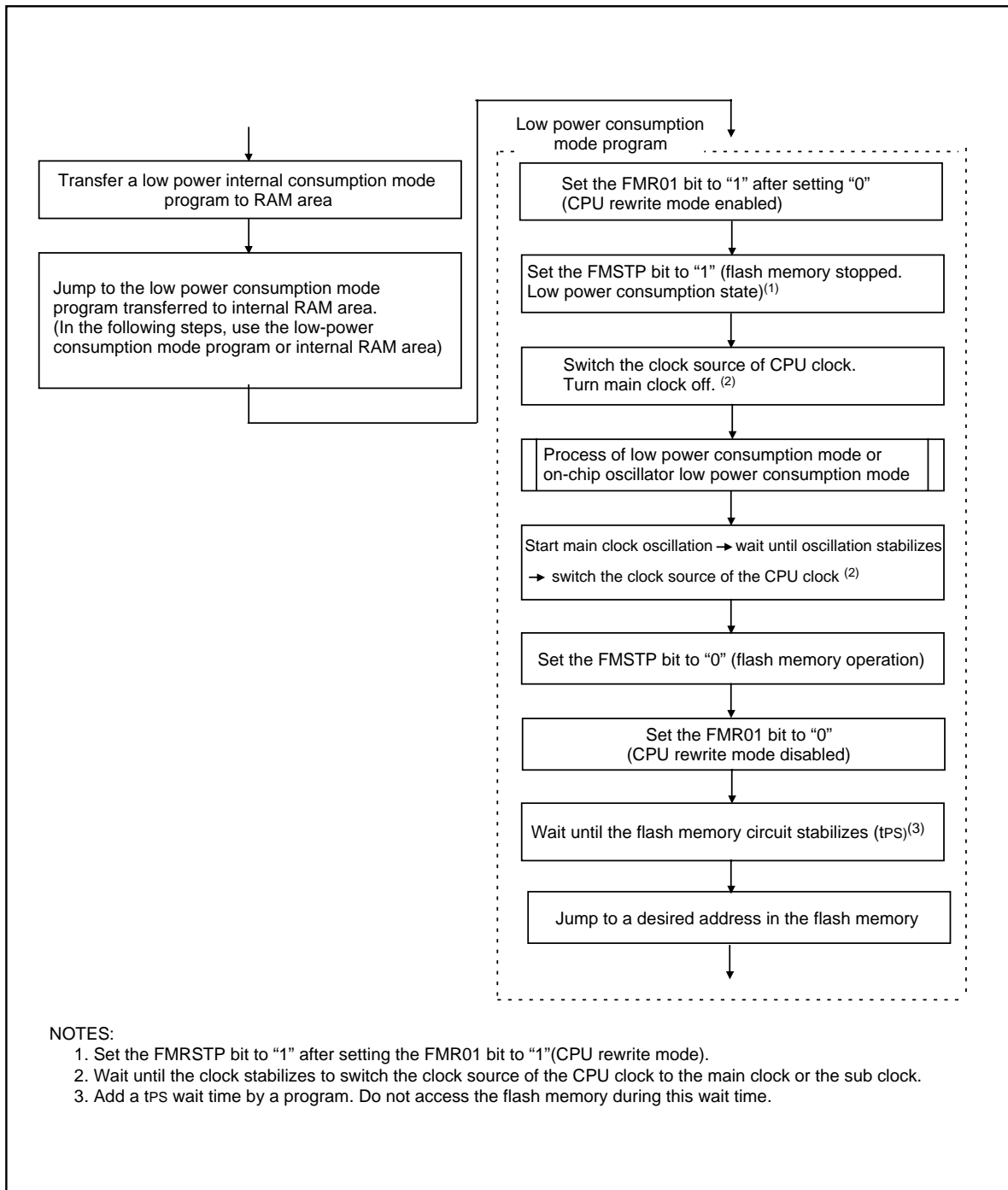


Figure 17.5.1.3. Processing Before and After Low Power Dissipation Mode

## 17.6 Precautions in CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

### 17.6.1 Operation Speed

When CPU clock source is the main clock, before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or below for CPU clock using the CM06 bit in the CM0 register and the CM16 bits in the CM1 register. Also, when selecting f<sub>3</sub>(ROC) of a on-chip oscillator as a CPU clock source, before entering CPU rewrite mode (EW0 or EW1 mode), the ROCR3 to ROCR2 bits in the ROCR register set the CPU clock division rate to “divide-by-4” or “divide-by-8”.

On both cases, set the PM17 bit in the PM1 register to “1” (with wait state).

### 17.6.2 Prohibited Instructions

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

### 17.6.3 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 can be used since the FMR0 and FMR1 registers are forcibly reset when either interrupt is generated. However, the jump addresses for each interrupt service routines to the fixed vector table are set and interrupt programs are required. Flash memory rewrite operation is halted when the NMI or watchdog timer interrupt is generated. Set the FMR01 bit to “1” and execute the rewrite and erase program again after exiting the interrupt routine.
- The address match interrupt can not be used since the CPU tries to read data in the flash memory.

EW1 Mode

- Do not acknowledge any interrupts with vectors in the relocatable vector table or the address match interrupt during the auto-program or erase-suspend function.

### 17.6.4 How to Access

To set the FMR01, FMR02, FMR11 or FMR16 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 it to “1”. When the  $\overline{\text{NMI}}$  function is selected, set the bit while an “H” signal is applied to the P85/ $\overline{\text{NMI/SD}}$  pin.

### 17.6.5 Writing in the User ROM Space

#### 17.6.5.1 EW0 Mode

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

#### 17.6.5.2 EW1 Mode

- Do not rewrite the block where the rewrite control program is stored.

### 17.6.6 DMA Transfer

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

### 17.6.7 Writing Command and Data

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

### 17.6.8 Wait Mode

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

### 17.6.9 Stop Mode

When entering stop mode, set the FMR01 bit to "0" (CPU rewrite mode disabled) and disable the DMA transfer before setting the CM10 bit to "1" (stop mode).

### 17.6.10 Low Power Consumption Mode and On-chip Oscillator-Low Power Consumption Mode

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

- Program
- Block erase

## 17.7 Software Commands

Read or write 16-bit commands and data from or to even addresses in the user ROM area. When writing a command code, 8 high-order bits (D15–D8) are ignored.

**Table 17.7.1. Software Commands**

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read array	Write	X	xxFF <sub>16</sub>			
Read status register	Write	X	xx70 <sub>16</sub>	Read	X	SRD
Clear status register	Write	X	xx50 <sub>16</sub>			
Program	Write	WA	xx40 <sub>16</sub>	Write	WA	WD
Block erase	Write	X	xx20 <sub>16</sub>	Write	BA	xxD0 <sub>16</sub>

SRD: Status register data (D7 to D0)

WA : Write address (However, even address)

WD : Write data (16 bits)

BA : Highest-order block address (However, even address)

X : Any even address in the user ROM area

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

### 17.7.1 Read Array Command (FF<sub>16</sub>)

This command reads the flash memory.

By writing command code 'xxFF<sub>16</sub>' 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 microcomputer remains in read array mode until an another command is written. Therefore, contents of multiple addresses can be read consecutively.

### 17.7.2 Read Status Register Command (70<sub>16</sub>)

This command reads the status register.

By writing command code 'xx70<sub>16</sub>' in the first bus cycle, the status register can be read in the second bus cycle (Refer to **17.8 Status Register**). Read an even address in the user ROM area. Do not execute this command in EW1 mode.



### 17.7.3 Clear Status Register Command (5016)

This command clears the status register to "0".

By writing 'xx5016' in the first bus cycle, and the FMR06 to FMR07 bits in the FMR0 register and SR4 to SR5 bits in the status register are set to "0".

### 17.7.4 Program Command (4016)

The program command writes 2-byte data to the flash memory. By writing 'xx4016' in the first bus cycle and data to the write address specified in the second bus cycle, the auto-programming/erasing (data programming and verify) start. Set the address value specified in the first bus cycle to same and even address as the write address specified in the second bus cycle. The FMR00 bit in the FMR0 register indicates whether an auto-programming operation has been completed. The FMR00 bit is set to "0" during the auto-programming and "1" when the auto-programming operation is completed. After the auto-programming operation is completed, the FMR06 bit in the FMR0 register indicates whether or not the auto-programming operation has been completed as expected. (Refer to **17.8.4 Full Status Check**). Also, each block disables writing (Refer to "Table 17.5.2.1"). Do not write additions to the address which is already programmed. When commands other than a program command are executed immediately after a program command, set the same address as the write address specified in the second bus cycle of the program command, to the specified address value in the first bus cycle of the following command. In EW1 mode, do not execute this command on the blocks where the rewrite control program is allocated. In EW0 mode, the microcomputer enters read status register mode as soon as the auto-programming operation starts and the status register can be read. The SR7 bit in the status register is set to "0" as soon as the auto-programming operation starts. This bit is set to "1" when the auto-programming operation is completed. The microcomputer remains in read status register mode until the read array command is written. After completion of the auto-programming operation, the status register indicates whether or not the auto-programming operation has been completed as expected.

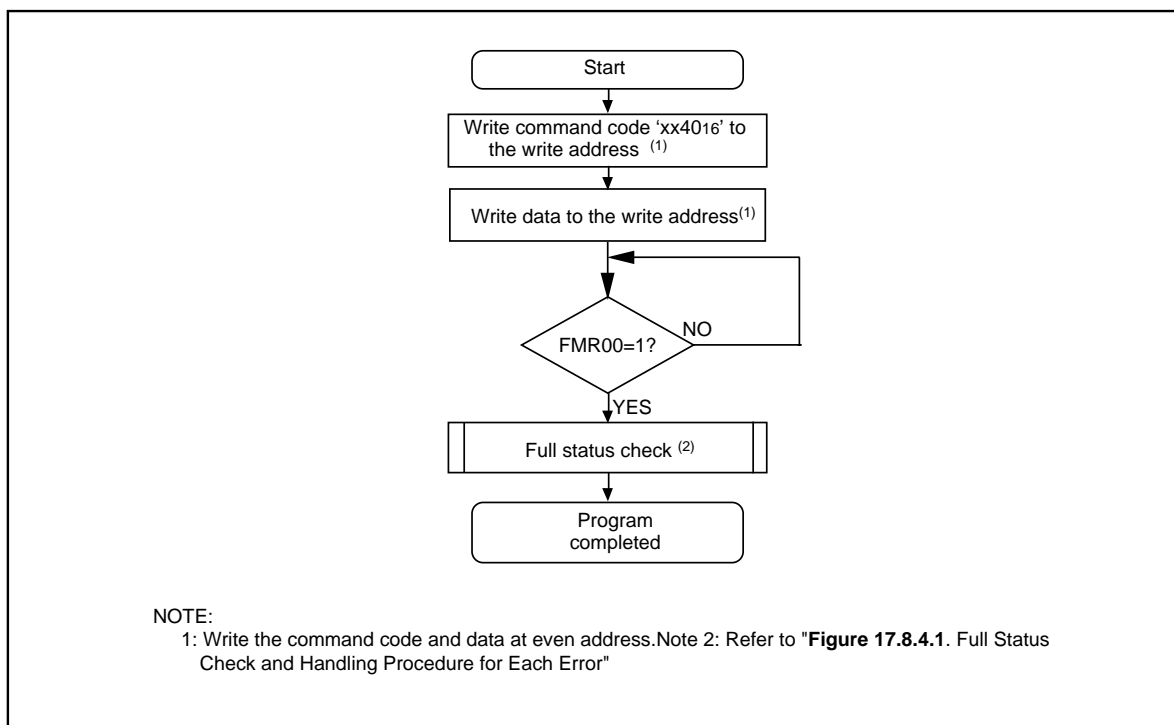
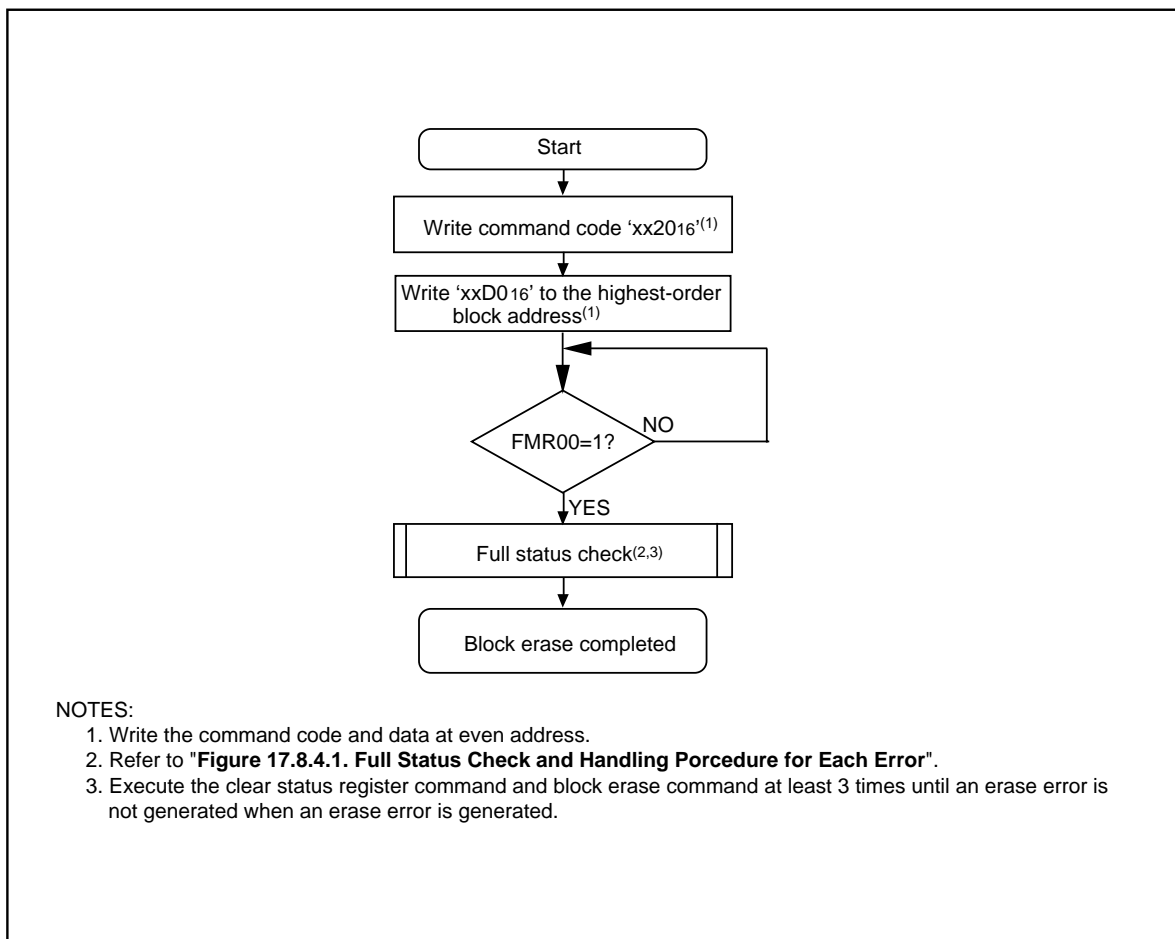


Figure 17.7.4.1. Flow Chart of Program Command

### 17.7.5 Block Erase

By writing 'xx2016' in the first bus cycle and 'xxD016' in the second bus cycle to the highest-order (even address) of a block) and the auto-programming/erasing (erase and erase verify) start. The FMR00 bit in the FMR0 register indicates whether the auto-programming operation has been completed. The FMR00 bit is set to "0" (busy) during the auto-erasing operation and "1" (ready) when the auto-erasing operation is completed. When using the erase-suspend function in EW0 mode, the FMR46 bit in the FMR4 register indicates whether a flash memory has entered erase-suspend mode. The FMR46 bit is set to "0" during auto-erasing operation and "1" when the auto-erasing operation is completed (entering erase-suspend). After the completion of an auto-erasing operation, the FMR07 bit in the FMR0 register indicates whether or not the auto erasing-operation has been completed as expected. (Refer to **17.8.4 Full Status Check**). Also, each block disables erasing. (Refer to "Table 17.5.2.1"). Figure 17.7.5.1 shows a flow chart of the block erase command programming when not using the erase-suspend function. Figure 17.7.5.2 shows a flow chart of the block erase command programming when using an erase-suspend function. In EW1 mode, do not execute this command on the block where the rewrite control program is allocated. In EW0 mode, the microcomputer enters read status register mode as soon as the auto-erasing operation starts and the status register can be read. The SR7 bit in the status register is set to "0" as soon as the auto-erasing operation starts. This bit is set to "1" when the auto-erasing operation is completed. The microcomputer remains in read status register mode until the read array 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 17.7.5.1. Flow Chart of Block Erase Command (when not using erase suspend function)**

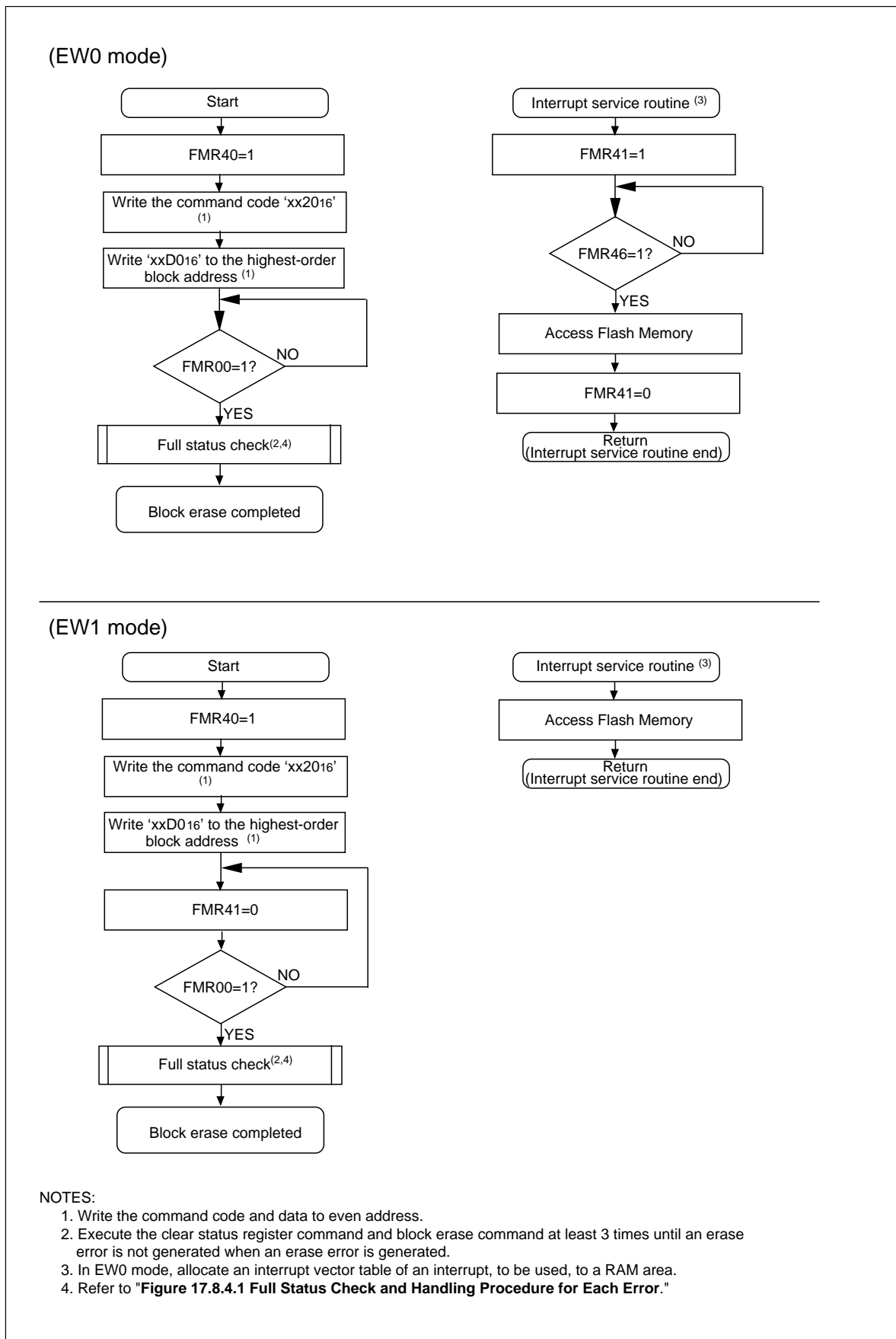


Figure 17.7.5.2. Block Erase Command (at use erase suspend)

## 17.8 Status Register

The status register indicates the operating status of the flash memory and whether an erasing or a programming operates normally and an error ends. The FMR00, FMR06, and FMR07 bits in the FMR0 register indicate the status of the status register.

Table 17.8.1 shows the status register.

In EW0 mode, the status register can be read in the following cases:

- (1) When a given even address in the user ROM area is read after writing the read status register command
- (2) When a given even address in the user ROM area is read after executing the program or block erase command but before executing the read array command.

### 17.8.1 Sequence Status (SR7 and FMR00 Bits)

The sequence status indicates the operating status of the flash memory. This bit is set to "0" (busy) during an auto-programming and auto-erasing and "1" (ready) as soon as these operations are completed. This bit indicates "0" (busy) in erase-suspend mode.

### 17.8.2 Erase Status (SR5 and FMR07 Bits)

Refer to 17.8.4 Full Status Check.

### 17.8.3 Program Status (SR4 and FMR06 Bits)

Refer to 17.8.4 Full Status Check.

**Table 17.8.1. Status Register**

Bits in the SRD register	Bits in the FMR0 register	Status name	Contents		Value after reset
			"0"	"1"	
SR7 (D7)	FMR00	Sequence status	Busy	Ready	1
SR6 (D6)	—	Reserved	-	-	—
SR5 (D5)	FMR07	Erase status	Completed normally	Terminated by error	0
SR4 (D4)	FMR06	Program status	Completed normally	Terminated by error	0
SR3 (D3)	—	Reserved	-	-	—
SR2 (D2)	—	Reserved	-	-	—
SR1 (D1)	—	Reserved	-	-	—
SR0 (D0)	—	Reserved	-	-	—

- D7 to D0: Indicates the data bus which is read out when executing the read status register command.
- The FMR07 bit (SR5) and FMR06 bit (SR4) are set to "0" by executing the clear status register command.
- When the FMR07 bit (SR5) or FMR06 bit (SR4) is 1, the program, and block erase command are not acknowledged.

### 17.8.4 Full Status Check

When an error occurs, the FMR06 to FMR07 bits in the FMR0 register are set to “1”, indicating occurrence of each specific error. Therefore, execution results can be verified by checking these status bits (full status check). Table 17.8.4.1 shows errors and the status of FMR0 register. Figure 17.8.4.1 shows a flow chart of the full status check and handling procedure for each error.

**Table 17.8.4.1. Errors and FMR0 Register Status**

FMR0 register (SRD register) status		Error	Error occurrence condition
FMR07 (SR5)	FMR06 (SR4)		
1	1	Command sequence error	<ul style="list-style-type: none"> <li>• When any commands are not written correctly</li> <li>• A value other than 'xxD016' or 'xxFF16' is written in the second bus cycle of the block erase command <sup>(1)</sup></li> <li>• When the block erase command is executed on protected blocks</li> <li>• When the program command is executed on protected blocks</li> </ul>
1	0	Erase error	<ul style="list-style-type: none"> <li>• When the block erase command is executed on unprotected blocks but the blocks are not automatically erased correctly</li> </ul>
0	1	Program error	<ul style="list-style-type: none"> <li>• When the program command is executed on unprotected blocks but the blocks are not automatically programmed correctly.</li> </ul>

**NOTE:**

1. The flash memory enters read array mode by writing command code 'xxFF16' in the second bus cycle of these commands. The command code written in the first bus cycle becomes invalid.

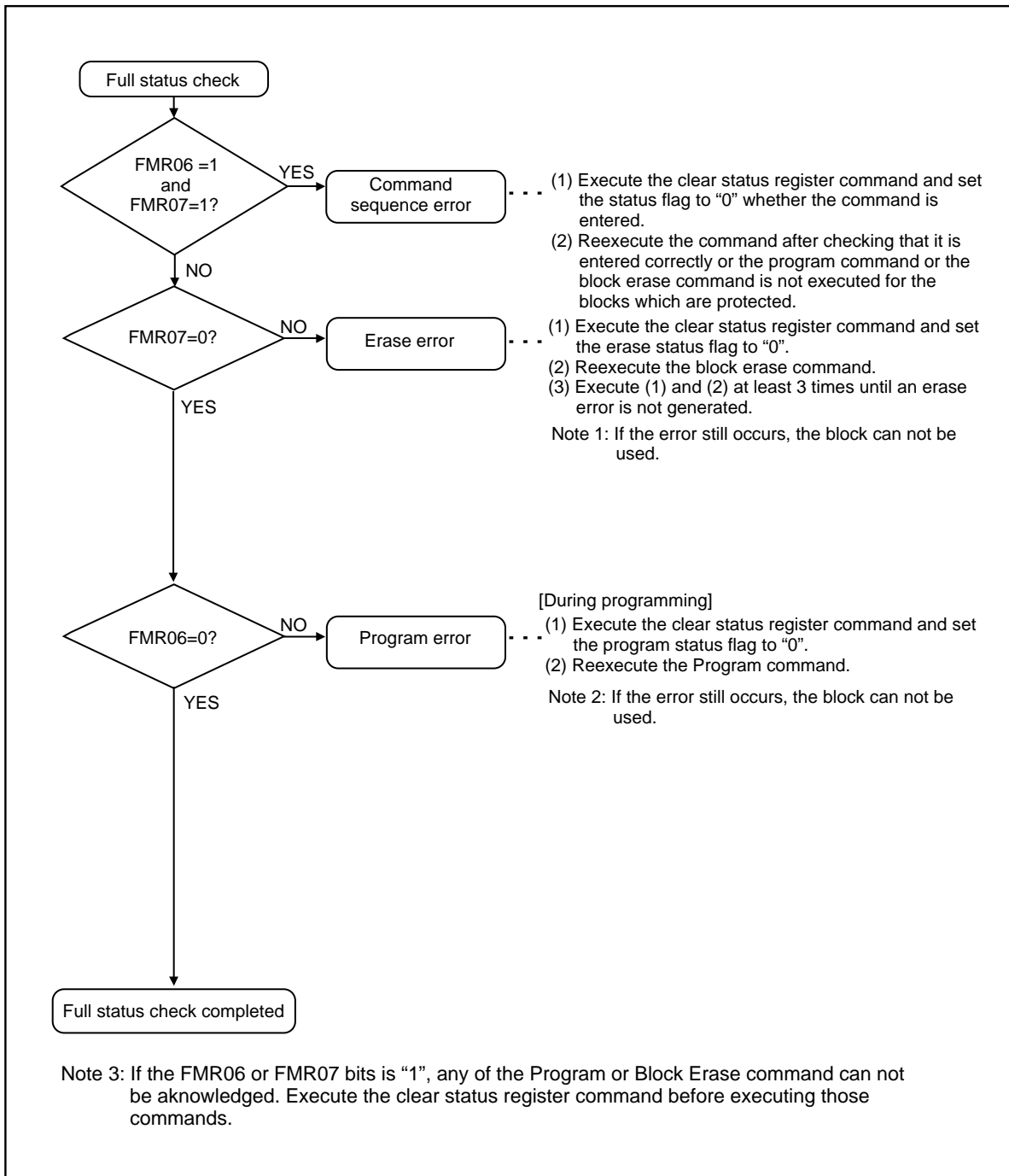


Figure 17.8.4.1. Full Status Check and Handling Procedure for Each Error

## 17.9 Standard Serial I/O Mode

In standard serial input/output mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer which is applicable for the M16C/26A group. For more information about serial programmers, contact the manufacturer of your serial programmer. For details on how to use the serial programmer, refer to the user's manual included with your serial programmer.

Table 17.9.1 shows pin functions (flash memory standard serial input/output mode). Figures 17.9.1 and 17.9.2 show pin connections for standard serial input/output mode.

### 17.9.1 ID Code Check Function

This function determines whether the ID codes sent from the serial programmer and those written in the flash memory match. (Refer to **17.3 Functions To Prevent Flash Memory from Rewriting**.)

**Table 17.9.1. Pin Functions (Flash Memory Standard Serial I/O Mode)**

Pin	Name	I/O	Description
Vcc, Vss	Power input		Apply the voltage guaranteed for Program and Erase to Vcc pin and 0 V to Vss pin.
CNVss	CNVss	I	Connect to Vcc pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is "L" level, wait for td(ROC).
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between X IN and XOUT pins. To input an externally generated clock, input it to X IN pin and open XOUT pin.
XOUT	Clock output	O	
AVcc, AVss	Analog power supply input		Connect AVss to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for AD from this pin.
P15, P17	Input port P1	I	Input "H" or "L" level signal or open.
P16	P16 input	I	Connect this pin to Vcc while $\overline{\text{RESET}}$ is low. (2)
P60 to P63	Input port P6	I	Input "H" or "L" level signal or open.
P64	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitor signal output pin for boot program operation check
P65	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input "L".
P66	RxD input	I	Serial data input pin
P67	TxD output	O	Serial data output pin (1)
P70 to P77	Input port P7	I	Input "H" or "L" level signal or open.
P80 to P84, P87	Input port P8	I	Input "H" or "L" level signal or open.
P85	$\overline{\text{RP}}$ input	I	Connect this pin to Vss while $\overline{\text{RESET}}$ is low. (2)
P86	$\overline{\text{CE}}$ input	I	Connect this pin to Vcc while $\overline{\text{RESET}}$ is low. (2)
P90 to P93,	Input port P9	I	Input "H" or "L" level signal or open.
P100 to P107	Input port P10	I	Input "H" or "L" level signal or open.

**NOTES:**

- When using standard serial input/output mode 1, to input "H" to the TxD pin is necessary while the  $\overline{\text{RESET}}$  pin is "L". Therefore, connect this pin to VCC via a resistor. Adjust the pull-up resistor value on a system not to affect a data transfer after reset, because this pin changes to a data-output pin
- Set following either or both
  - Connect the  $\overline{\text{CE}}$  pin to Vcc.
  - Connect the  $\overline{\text{RP}}$  pin to Vss and the P16 pin to Vcc.



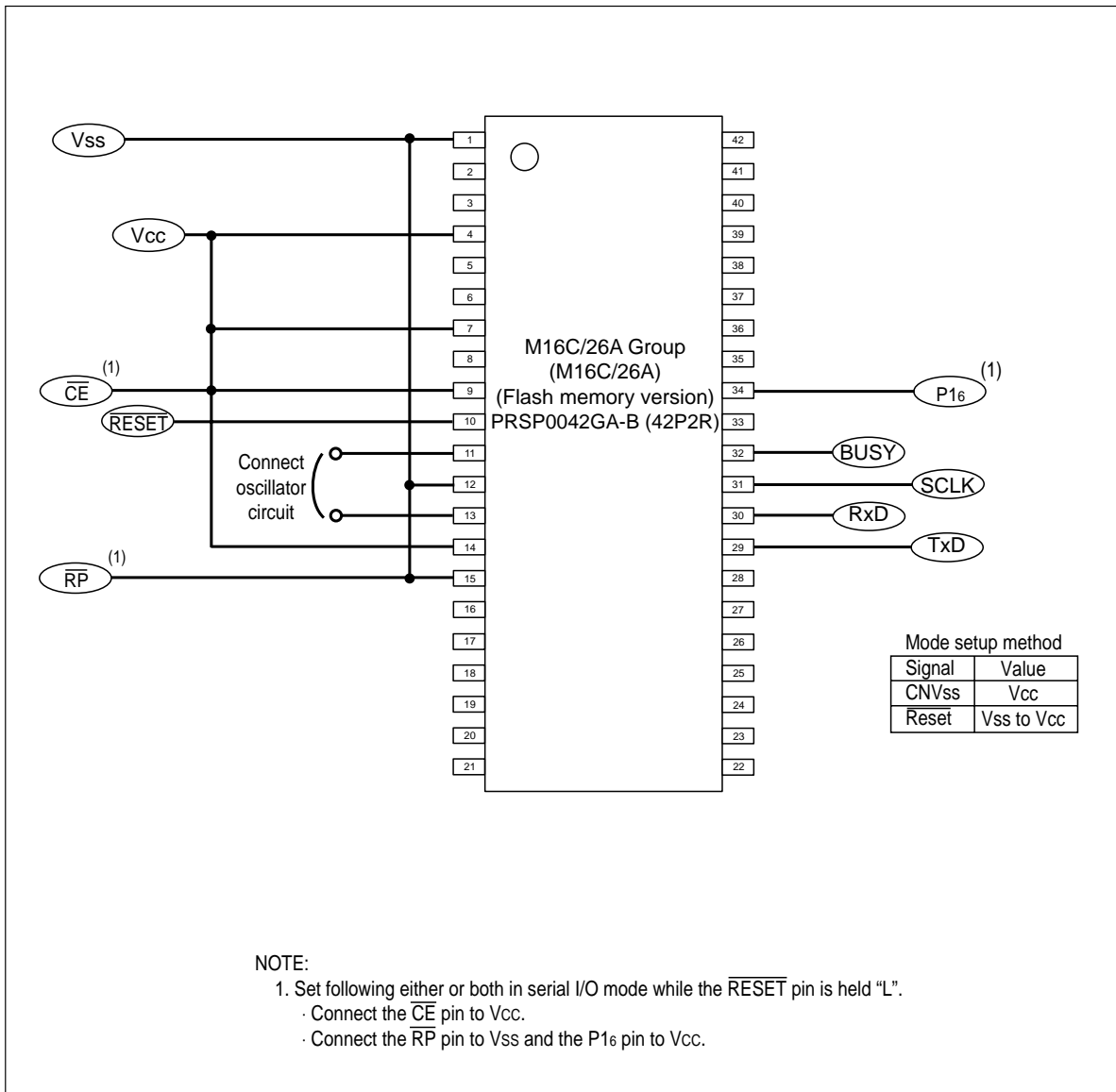


Figure 17.9.1. Pin Connections for Serial I/O Mode (1)

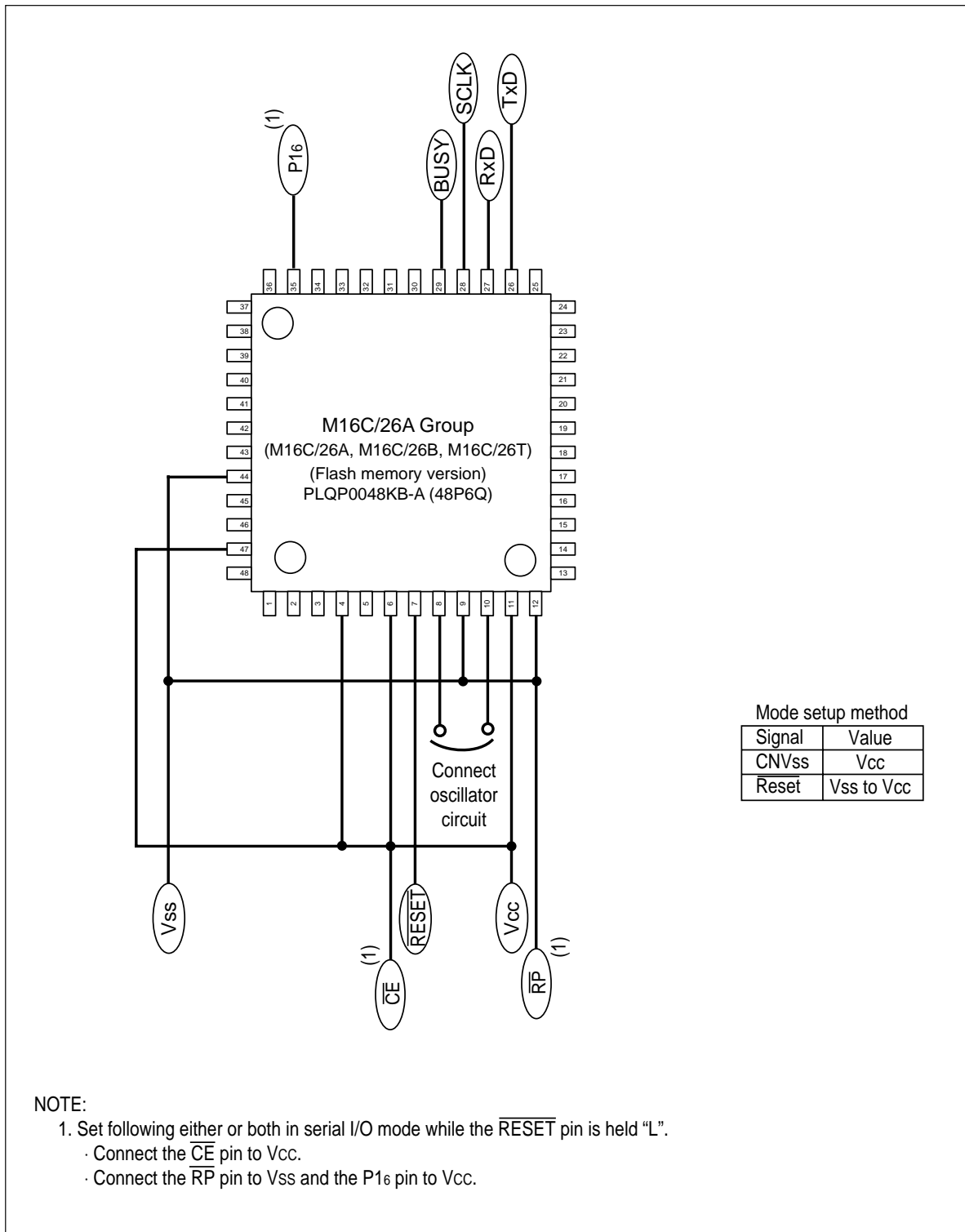


Figure 17.9.2. Pin Connections for Serial I/O Mode (2)

### 17.9.2 Example of Circuit Application in Standard Serial I/O Mode

Figure 17.9.2.1 shows an example of a circuit application in standard serial I/O mode 1 and Figure 17.9.2.2 shows an example of a circuit application in standard serial I/O mode 2. Refer to the user's manual for a serial writer to handle pins controlled by the serial writer.

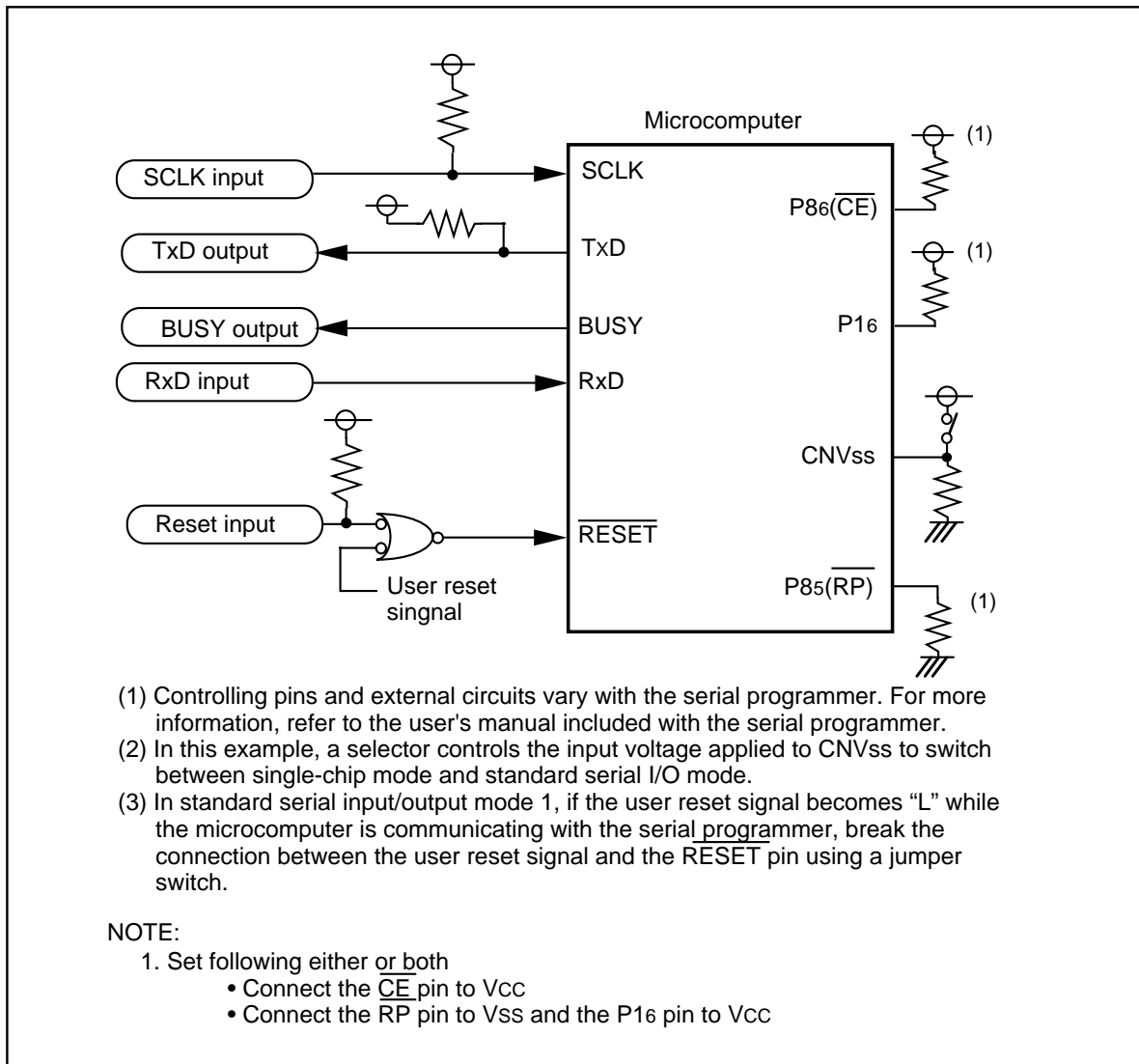
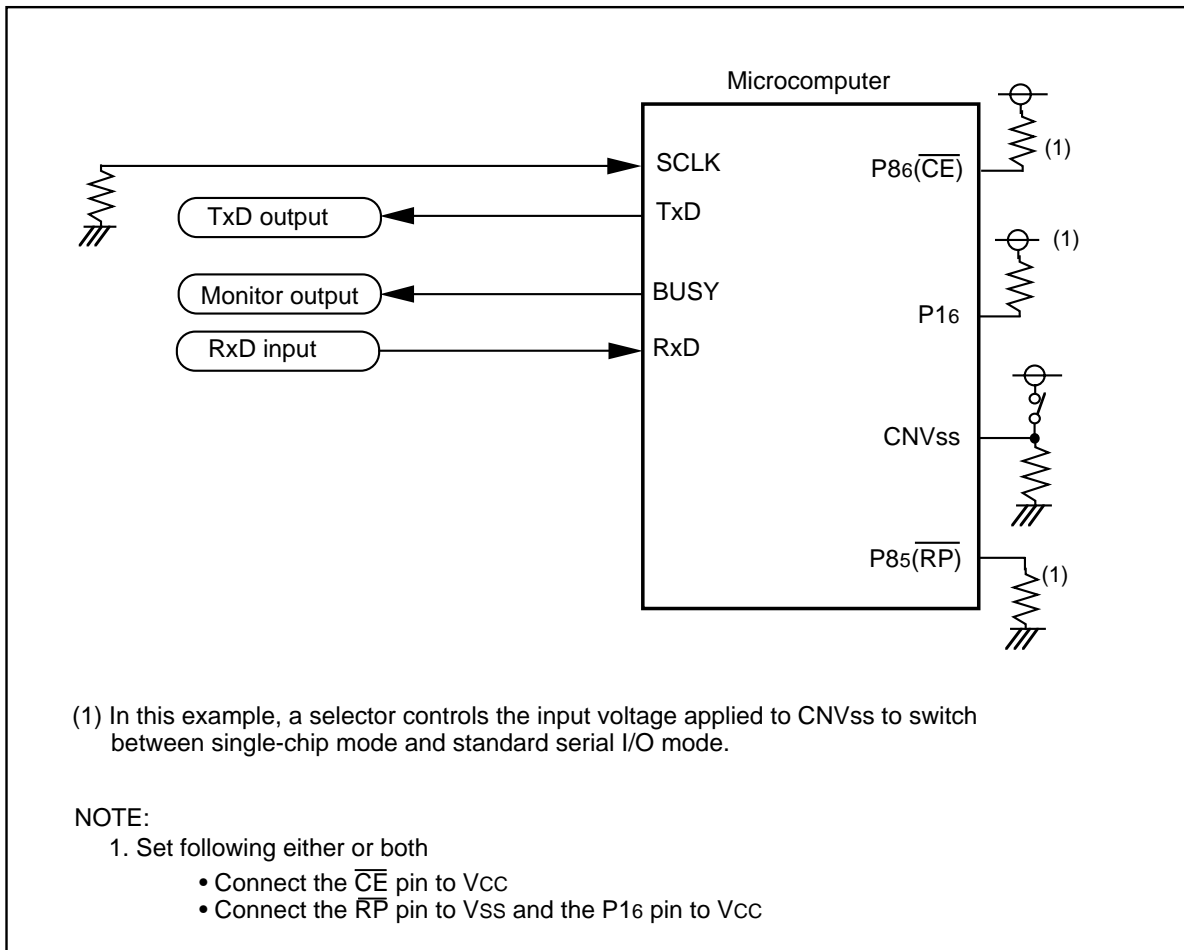


Figure 17.9.2.1. Circuit Application in Standard Serial I/O Mode 1



**Figure 17.9.2.2. Circuit Application in Standard Serial I/O Mode 2**

## 17.10 Parallel I/O Mode

In parallel input/output mode, the user ROM can be rewritten using a parallel programmer which is applicable for the M16C/26A group. For more information about the parallel programmer, contact your parallel programmer manufacturer. For details on how to use the parallel programmer, refer to the user's manual of the parallel programmer.

### 17.10.1 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read or rewritten. (Refer to **17.3 Function to Prevent Flash Memory from Rewriting.**)

## 18. Electrical Characteristics

Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for electrical characteristics of V-ver.

### 18.1. M16C/26A, M16C/26B (Normal version)

**Table 18.1. Absolute Maximum Ratings**

Symbol	Parameter		Condition	Value	Unit
V <sub>CC</sub>	Supply Voltage		V <sub>CC</sub> = AV <sub>CC</sub>	-0.3 to 6.5	V
AV <sub>CC</sub>	Analog Supply Voltage		V <sub>CC</sub> = AV <sub>CC</sub>	-0.3 to 6.5	V
V <sub>I</sub>	Input Voltage	P1 <sub>5</sub> to P1 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>3</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>IN</sub> , V <sub>REF</sub> , RESET, CNV <sub>SS</sub>		-0.3 to V <sub>CC</sub> +0.3	V
V <sub>O</sub>	Output Voltage	P1 <sub>5</sub> to P1 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>3</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>OUT</sub>		-0.3 to V <sub>CC</sub> +0.3	V
P <sub>d</sub>	Power Dissipation		-40 ≤ Topr ≤ 85° C	300	mW
Topr	Operating Ambient Temperature	during CPU operation		-20 to 85 / -40 to 85 <sup>(1)</sup>	° C
		during flash memory program and erase operation	Program Space (Block 0 to Block 3)	0 to 60	° C
			Data Space (Block A, Block B)	0 to 60 / -20 to 85 / -40 to 85 <sup>(1)</sup>	° C
Tstg	Storage Temperature			-65 to 150	° C

NOTE:

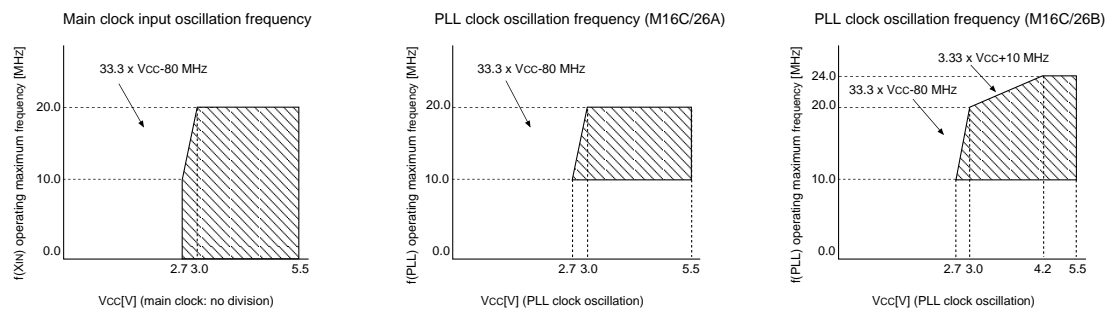
1. Refer to **Tables 1.7 and 1.8.**

**Table 18.2. Recommended Operating Conditions (1)**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V <sub>CC</sub>	Supply Voltage		2.7		5.5	V
AV <sub>CC</sub>	Analog Supply Voltage			V <sub>CC</sub>		V
V <sub>SS</sub>	Supply Voltage			0		V
AV <sub>SS</sub>	Analog Supply Voltage			0		V
V <sub>IH</sub>	Input High ("H") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	0.7 V <sub>CC</sub>		V <sub>CC</sub>	V
		XIN, RESET, CNVSS	0.8 V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	Input Low ("L") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	0		0.3 V <sub>CC</sub>	V
		XIN, RESET, CNVSS	0		0.2 V <sub>CC</sub>	V
I <sub>OH(peak)</sub>	Peak Output High ("H") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107			-10.0	mA
I <sub>OH(avg)</sub>	Average Output High ("H") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107			-5.0	mA
I <sub>OL(peak)</sub>	Peak Output Low ("L") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107			10.0	mA
I <sub>OL(avg)</sub>	Average Output Low ("L") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107			5.0	mA
f(XIN)	Main Clock Input Oscillation Frequency <sup>(4)</sup>	V <sub>CC</sub> = 3.0 to 5.5 V	0		20	MHz
		V <sub>CC</sub> = 2.7 to 3.0 V	0		33 X V <sub>CC</sub> -80	MHz
f(X <sub>CIN</sub> )	Sub Clock Oscillation Frequency			32.768	50	kHz
f <sub>1</sub> (ROC)	On-chip Oscillator Frequency 1		0.5	1	2	MHz
f <sub>2</sub> (ROC)	On-chip Oscillator Frequency 2		1	2	4	MHz
f <sub>3</sub> (ROC)	On-chip Oscillator Frequency 3		8	16	26	MHz
f(PLL)	PLL Clock Oscillation Frequency <sup>(4)</sup>	V <sub>CC</sub> = 4.2 to 5.5 V (M16C/26B)	10		24	MHz
		V <sub>CC</sub> = 3.0 to 4.2 V (M16C/26B)	10		3.33 X V <sub>CC</sub> +10	MHz
		V <sub>CC</sub> = 3.0 to 5.5 V (M16C/26A)	10		20	MHz
		V <sub>CC</sub> = 2.7 to 3.0 V	10		33 X V <sub>CC</sub> -80	MHz
f(BCLK)	CPU Operation Clock Frequency	M16C/26A	0		20	MHz
		M16C/26B	0		24	MHz
t <sub>su</sub> (PLL)	Wait Time to Stabilize PLL Frequency Synthesizer	V <sub>CC</sub> =5.0V			20	ms
		V <sub>CC</sub> =3.0V			50	ms

## NOTES:

1. Referenced to V<sub>CC</sub> = 2.7 to 5.5 V at Topr = -20 to 85 ° C / -40 to 85 ° C unless otherwise specified.
2. The mean output current is the mean value within 100 ms.
3. The total I<sub>OL(peak)</sub> for all ports must be 80 mA or less. The total I<sub>OH(peak)</sub> for all ports must be -80 mA or less.
4. Relationship among main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.



**Table 18.3. A/D Conversion Characteristics<sup>(1)</sup>**

Symbol	Parameter		Measurement Condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution		$V_{REF} = V_{CC}$			10	Bits
INL	Integral Nonlinearity Error	10 bit	$V_{REF} = V_{CC} = 5V$			$\pm 3$	LSB
			$V_{REF} = V_{CC} = 3.3V$			$\pm 5$	LSB
	Absolute Accuracy	8 bit	$V_{REF} = V_{CC} = 3.3V, 5V$			$\pm 2$	LSB
			$V_{REF} = V_{CC} = 5V$			$\pm 3$	LSB
		10 bit	$V_{REF} = V_{CC} = 3.3V$			$\pm 5$	LSB
		8 bit	$V_{REF} = V_{CC} = 3.3V, 5V$			$\pm 2$	LSB
DNL	Differential Nonlinearity Error					$\pm 1$	LSB
-	Offset Error					$\pm 3$	LSB
-	Gain Error					$\pm 3$	LSB
RLADDER	Resistor Ladder		$V_{REF} = V_{CC}$	10		40	k $\Omega$
t <sub>CONV</sub>	10-bit Conversion Time Sample & Hold Function Available		$V_{REF} = V_{CC} = 5V, \phi_{AD} = 10\text{ MHz}$	3.3			$\mu\text{s}$
t <sub>CONV</sub>	8-bit Conversion Time Sample & Hold Function Available		$V_{REF} = V_{CC} = 5V, \phi_{AD} = 10\text{ MHz}$	2.8			$\mu\text{s}$
V <sub>REF</sub>	Reference Voltage			2.0		V <sub>CC</sub>	V
V <sub>IA</sub>	Analog Input Voltage			0		V <sub>REF</sub>	V

**NOTES:**

1. Referenced to  $V_{CC}=AV_{CC}=V_{REF}= 3.3$  to  $5.5V$ ,  $V_{SS}=AV_{SS}=0V$  at  $T_{opr} = -20$  to  $85^\circ\text{C}$  /  $-40$  to  $85^\circ\text{C}$  unless otherwise specified.
2. Keep  $\phi_{AD}$  frequency at 10 MHz or less (12 MHz or less in M16C/26B). Additionally, divide the  $f_{AD}$  if  $V_{CC}$  is less than 4.2V, and make  $\phi_{AD}$  frequency equal to or lower than  $f_{AD}/2$ .
3. When sample & hold function is disabled, keep  $\phi_{AD}$  frequency at 250 kHz or more in addition to the limitation in Note 2. When sample & hold function is enabled, keep  $\phi_{AD}$  frequency at 1 MHz or more in addition to the limitation in Note 2.
4. When sample & hold function is enabled, sampling time is  $3/\phi_{AD}$  frequency. When sample & hold function is disabled, sampling time is  $2/\phi_{AD}$  frequency.



**Table 18.4. Flash Memory Version Electrical Characteristic (1):  
Program Space and Data Space for U3 and U5, Program Space for U7 and U9**

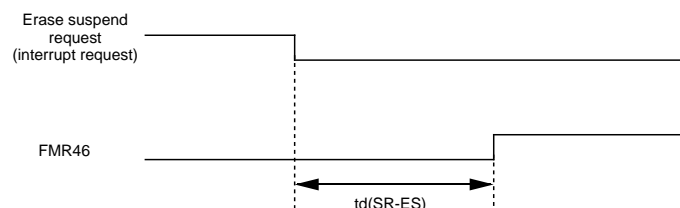
Symbol	Parameter	Standard			Unit
		Min.	Typ. <sup>(2)</sup>	Max.	
-	Program and Erase Endurance <sup>(3)</sup>	100/1000 <sup>(4, 11)</sup>			cycles
-	Word Program Time (V <sub>CC</sub> =5.0V, T <sub>opr</sub> =25° C)		75	600	μs
-	Block Erase Time (V <sub>CC</sub> =5.0V, T <sub>opr</sub> =25° C)	2-Kbyte Block	0.2	9	s
		8-Kbyte Block	0.4	9	s
		16-Kbyte Block	0.7	9	s
		32-Kbyte Block	1.2	9	s
td(SR-ES)	Duration between Suspend Request and Erase Suspend			8	ms
t <sub>PS</sub>	Wait Time to Stabilize Flash Memory Circuit			15	μs
-	Data Hold Time <sup>(5)</sup>	20			years

**Table 18.5. Flash Memory Version Electrical Characteristics (6): Data Space for U7 and U9 (7)**

Symbol	Parameter	Standard			Unit
		Min.	Typ. <sup>(2)</sup>	Max.	
-	Program and Erase Endurance <sup>(3, 8, 9)</sup>	10000 <sup>(4, 10)</sup>			cycles
-	Word Program Time (V <sub>CC</sub> =5.0V, T <sub>opr</sub> =25° C)		100		μs
-	Block Erase Time (V <sub>CC</sub> =5.0V, T <sub>opr</sub> =25° C) (2-Kbyte block)		0.3		s
td(SR-ES)	Duration between Suspend Request and Erase Suspend			8	ms
t <sub>PS</sub>	Wait Time to Stabilize Flash Memory Circuit			15	μs
-	Data Hold Time <sup>(5)</sup>	20			years

NOTES:

1. Referenced to V<sub>CC</sub> = 2.7 to 5.5 V at T<sub>opr</sub> = 0 to 60° C (program space), -40 to 85° C (data space), unless otherwise specified.
2. V<sub>CC</sub> = 5.0 V; T<sub>opr</sub> = 25° C
3. Program and erase endurance is defined as number of program-erase cycles per block.  
If program and erase endurance is *n* cycle (*n* = 100, 1000, 10000), each block can be erased and programmed *n* cycles.  
For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guaranteed (1 to minimum value are guaranteed).
5. T<sub>opr</sub> = 55° C
6. Referenced to V<sub>CC</sub> = 2.7 to 5.5 V at T<sub>opr</sub> = -40 to 85° C (U7) / -20 to 85° C (U9) unless otherwise specified.
7. Table 18.5 applies for data space in U7 and U9 when program and erase endurance is more than 1,000 cycles. Otherwise, use Table 18.4.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. 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.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register 1 to "1" (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3 and U5; 1,000 cycles for program space in U7 and U9.
12. Customers desiring E/W failure rate information should contact their Renesas technical support representative.



**Table 18.6. Voltage Detection Circuit Electrical Characteristics (1, 3)**

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
Vdet4	Low Voltage Detection Voltage <sup>(1)</sup>	V <sub>CC</sub> =0.8 to 5.5V	3.2	3.8	4.45	V
Vdet3	Reset Level Detection Voltage <sup>(1)</sup>		2.3	2.8	3.4	V
Vdet3s	Low Voltage Reset Hold Voltage <sup>(2)</sup>				1.7	V
Vdet3r	Low Voltage Reset Release Voltage		2.35	2.9	3.5	V

## NOTES:

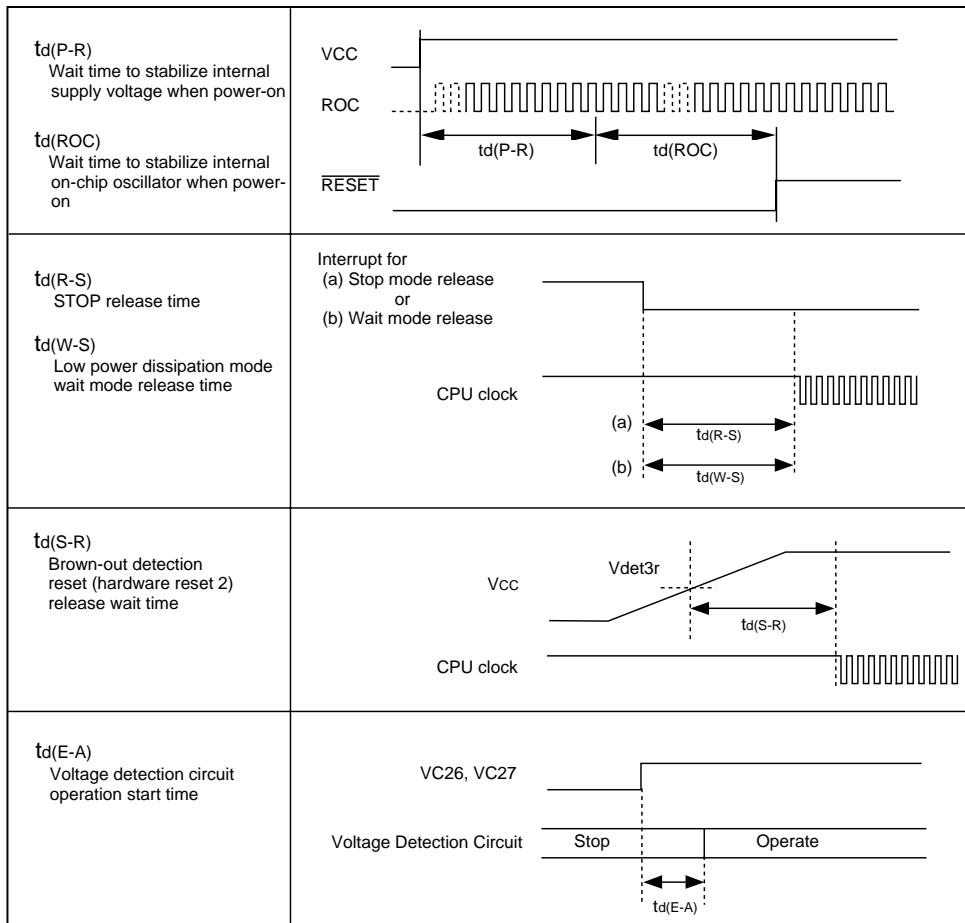
- Vdet4 > Vdet3
- Vdet3s is the minimum voltage to maintain "hardware reset 2".
- The voltage detection circuit is designed to use when V<sub>CC</sub> is set to 5V.
- If the supply power voltage is greater than the reset level detection voltage when the reset level detection voltage is less than 2.7V, the operation at f(BCLK) ≤ 10MHz is guaranteed. However, A/D conversion, serial I/O, flash memory program and erase are excluded.

**Table 18.7. Power Supply Circuit Timing Characteristics**

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Wait Time to Stabilize Internal Supply Voltage when Power-on	V <sub>CC</sub> = 2.7 to 5.5 V			2	ms
td(ROC)	Wait Time to Stabilize Internal On-chip Oscillator when Power-on				40	μs
td(R-S)	STOP Release Time				150	μs
td(W-S)	Low Power Dissipation Mode Wait Mode Release Time				150	μs
td(S-R)	Hardware Reset 2 Release Wait Time	V <sub>CC</sub> = Vdet3r to 5.5 V		6 <sup>(1)</sup>	20	ms
td(E-A)	Voltage Detection Circuit Operation Start Time	V <sub>CC</sub> = 2.7 to 5.5 V			20	μs

## NOTES:

- When V<sub>CC</sub>=5V



$V_{CC} = 5V$ **Table 18.8. Electrical Characteristics (1)**

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
$V_{OH}$	Output High ("H") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OH} = -5 \text{ mA}$	$V_{CC}-2.0$		$V_{CC}$	V
$V_{OH}$	Output High ("H") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OH} = -200 \mu\text{A}$	$V_{CC}-0.3$		$V_{CC}$	V
$V_{OH}$	Output High ("H") Voltage	$X_{OUT}$	High Power	$I_{OH} = -1\text{mA}$	$V_{CC}-2.0$	$V_{CC}$	V
			Low Power	$I_{OH} = -0.5\text{mA}$	$V_{CC}-2.0$	$V_{CC}$	
	Output High ("H") Voltage	$X_{COUT}$	High Power	No load applied		2.5	V
			Low Power	No load applied		1.6	
$V_{OL}$	Output Low ("L") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OL} = 5 \text{ mA}$			2.0	V
$V_{OL}$	Output Low ("L") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OL} = 200 \mu\text{A}$			0.45	V
$V_{OL}$	Output Low ("L") Voltage	$X_{OUT}$	High Power	$I_{OL} = 1 \text{ mA}$		2.0	V
			Low Power	$I_{OL} = 0.5 \text{ mA}$		2.0	
	Output Low ("L") Voltage	$X_{COUT}$	High Power	No load applied		0	V
			Low Power	No load applied		0	
$V_{T+}-V_{T-}$	Hysteresis	TA0IN-TA4IN, TB0IN-TB2IN, INT0-INT5, NMI, ADTRG, CTS0-CTS2, CLK0-CLK2, TA2OUT-TA4OUT, KI0-KI3, RXD0-RXD2		0.2		1.0	V
$V_{T+}-V_{T-}$	Hysteresis	RESET		0.2		2.5	V
$V_{T+}-V_{T-}$	Hysteresis	$X_{IN}$		0.2		0.8	V
$I_{IH}$	Input High ("H") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107, $X_{IN}$ , RESET, CNVSS	$V_I = 5V$			5.0	$\mu\text{A}$
$I_{IL}$	Input Low ("L") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107, $X_{IN}$ , RESET, CNVSS	$V_I = 0V$			-5.0	$\mu\text{A}$
$R_{PULLUP}$	Pull-up Resistance	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$V_I = 0V$	30	50	170	$k\Omega$
$R_{fXIN}$	Feedback Resistance	$X_{IN}$			1.5		$M\Omega$
$R_{fXCIN}$	Feedback Resistance	$X_{CIN}$			15		$M\Omega$
$V_{RAM}$	RAM Standby Voltage		In stop mode	2.0			V

NOTE:

1. Referenced to  $V_{CC}=4.2$  to  $5.5V$ ,  $V_{SS}=0V$  at  $T_{opr}=-20$  to  $85^\circ\text{C}$  /  $-40$  to  $85^\circ\text{C}$ ,  $f(\text{BCLK})=20\text{MHz}$  unless otherwise specified.

$V_{CC} = 5V$ Table 18.9. Electrical Characteristics (2) <sup>(1)</sup>

Symbol	Parameter		Measurement Condition		Standard			Unit
					Min.	Typ.	Max.	
I <sub>CC</sub>	Power Supply Current (V <sub>CC</sub> = 4.0 to 5.5 V)	Output pins are left open and other pins are connected to V <sub>SS</sub>	Mask ROM	f(BCLK) = 20 MHz, main clock, no division		12	17	mA
				On-chip oscillation f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz		1		mA
			Flash memory	f(BCLK) = 24 MHz, PLL operates (M16C/26B)		20	23	mA
				f(BCLK) = 20 MHz, main clock, no division		16	19	mA
				On-chip oscillator operates, f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz		1		mA
			Flash memory program	f(BCLK) = 10 MHz, V <sub>CC</sub> = 5.0 V		11		mA
			Flash memory erase	f(BCLK) = 10 MHz, V <sub>CC</sub> = 5.0 V		12		mA
			Mask ROM	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on ROM <sup>(3)</sup>		25		μA
				On-chip oscillator operates, f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz, In wait mode		30		μA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM <sup>(3)</sup>		25		μA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory <sup>(3)</sup>		450		μA
				On-chip oscillator operates, f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz, In wait mode		50		μA
			Mask ROM, Flash memory	f(BCLK) = 32 kHz, In wait mode <sup>(2)</sup> , Oscillation capacity HIGH		10		μA
f(BCLK) = 32 kHz, In wait mode <sup>(2)</sup> , Oscillation capacity LOW		3			μA			
In stop mode, T <sub>OPR</sub> = 25° C		0.8		3	μA			
I <sub>det4</sub>	Low voltage detection dissipation current <sup>(4)</sup>				0.7	4	μA	
I <sub>det3</sub>	Reset level detection dissipation current <sup>(4)</sup>				1.2	8	μA	

## NOTES:

1. Referenced to V<sub>CC</sub> = 4.2 to 5.5 V, V<sub>SS</sub> = 0 V at T<sub>OPR</sub> = -20 to 85° C / -40 to 85° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using f<sub>CS2</sub>.
3. This indicates the memory in which the program to be executed exists.
4. I<sub>det</sub> is dissipation current when the following bit is set to "1" (detection circuit enabled).  
I<sub>det4</sub>: VC27 bit in the VCR2 register  
I<sub>det3</sub>: VC26 bit in the VCR2 register

$$V_{CC} = 5V$$

**Timing Requirements****( $V_{CC} = 5V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)****Table 18.10. External Clock Input (XIN input)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External Clock Input Cycle Time	50		ns
tw(H)	External Clock Input High ("H") Width	20		ns
tw(L)	External Clock Input Low ("L") Width	20		ns
tr	External Clock Rise Time		9	ns
tf	External Clock Fall Time		9	ns

$$V_{CC} = 5V$$

### Timing Requirements

( $V_{CC} = 5V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)

**Table 18.11. 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 18.12. 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 18.13. 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 18.14. 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 18.15. 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 18.16. Timer A Input (Two-phase Pulse Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	800		ns
$t_{su(TAIN-TAOUT)}$	TAiOUT input setup time	200		ns
$t_{su(TAOUT-TAIN)}$	TAiIN input setup time	200		ns

$V_{CC} = 5V$

**Timing Requirements****( $V_{CC} = 5V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)****Table 18.17. 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 18.18. 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 18.19. 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 18.20. 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 18.21. Serial I/O**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	70		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

**Table 18.22. External Interrupt  $\overline{INT}_i$  Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT}_i$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{INT}_i$ input LOW pulse width	250		ns

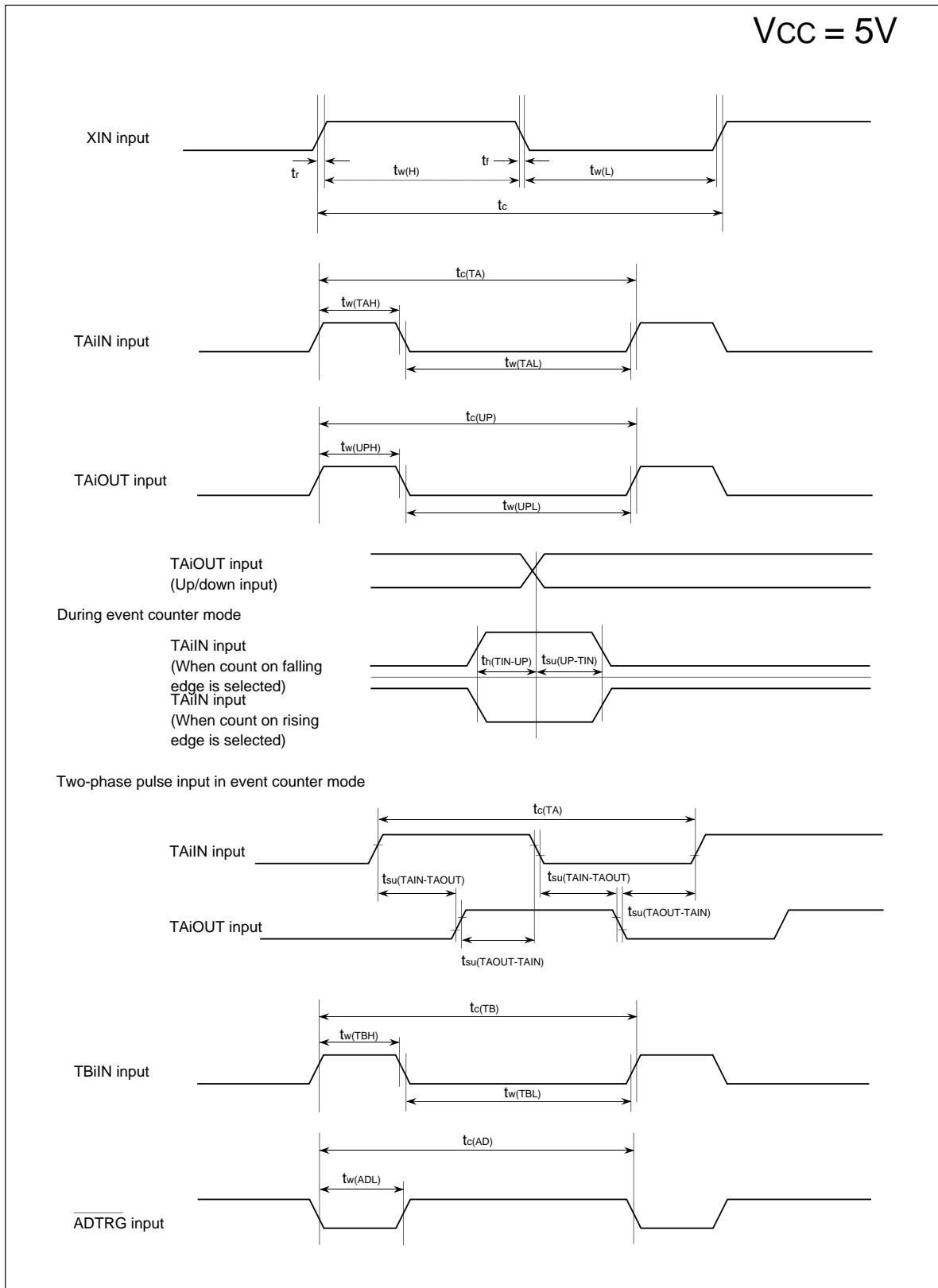


Figure 18.1. Timing Diagram (1)



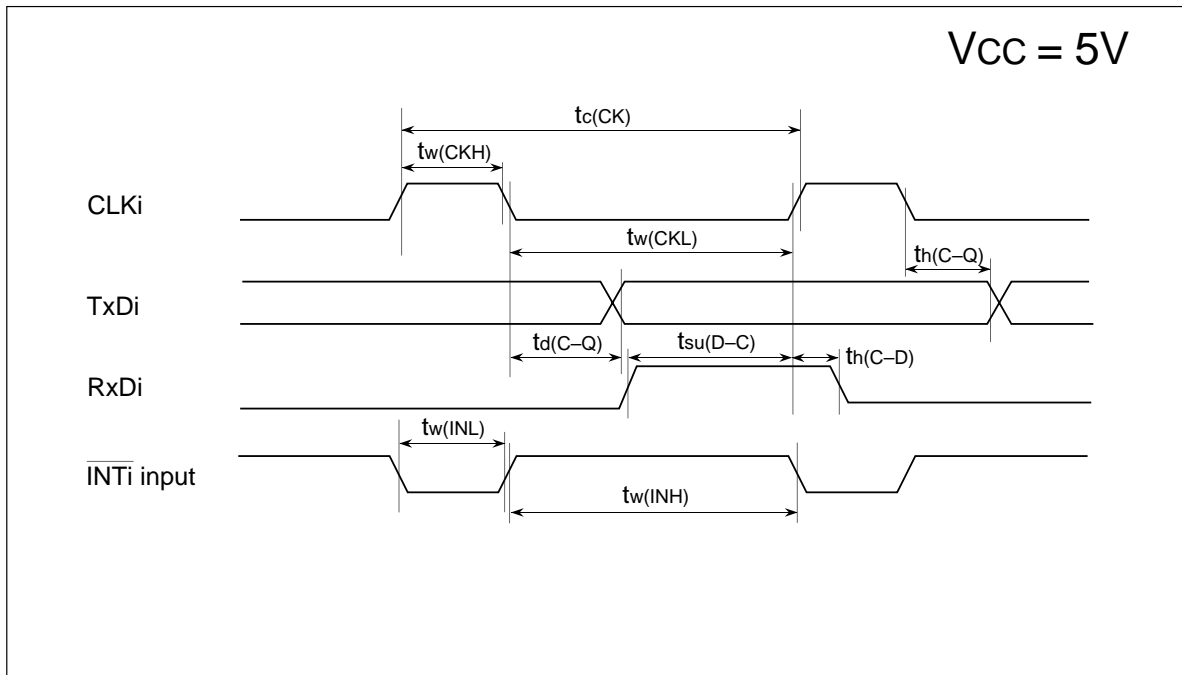


Figure 18.2. Timing Diagram (2)

$V_{CC} = 3V$ 

Table 18.23. Electrical Characteristics (1)

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
$V_{OH}$	Output High ("H") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OH} = -1mA$	$V_{CC} - 0.5$		$V_{CC}$	V
$V_{OH}$	Output High ("H") Voltage	$X_{OUT}$	High Power	$I_{OH} = -0.1mA$	$V_{CC} - 0.5$	$V_{CC}$	V
			Low Power	$I_{OH} = -50\mu A$	$V_{CC} - 0.5$	$V_{CC}$	
	Output High ("H") Voltage	$X_{COUT}$	High Power	No load applied		2.5	V
			Low Power	No load applied		1.6	
$V_{OL}$	Output Low ("L") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OL} = 1mA$			0.5	V
$V_{OL}$	Output Low ("L") Voltage	$X_{OUT}$	High Power	$I_{OL} = 0.1mA$		0.5	V
			Low Power	$I_{OL} = 50\mu A$		0.5	
	Output Low ("L") Voltage	$X_{COUT}$	High Power	No load applied		0	V
			Low Power	No load applied		0	
$V_{TH} - V_{TL}$	Hysteresis	$TA0_{IN} - TA4_{IN}$ , $TB0_{IN} - TB2_{IN}$ , $INT0 - INT5$ , $NMI$ , $ADTRG$ , $CTS0 - CTS2$ , $CLK0 - CLK2$ , $TA2_{OUT} - TA4_{OUT}$ , $KI0 - KI3$ , $RXD0 - RXD2$				0.8	V
$V_{TH} - V_{TL}$	Hysteresis	RESET				1.8	V
$V_{TH} - V_{TL}$	Hysteresis	$X_{IN}$				0.8	V
$I_{IH}$	Input High ("H") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107 $X_{IN}$ , RESET, $CNV_{SS}$	$V_I = 3V$			4.0	$\mu A$
$I_{IL}$	Input Low ("L") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107 $X_{IN}$ , RESET, $CNV_{SS}$	$V_I = 0V$			-4.0	$\mu A$
$R_{PULLUP}$	Pull-up Resistance	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$V_I = 0V$	50	100	500	$k\Omega$
$R_{fXIN}$	Feedback Resistance	$X_{IN}$				3.0	$M\Omega$
$R_{fXCN}$	Feedback Resistance	$X_{CN}$				25	$M\Omega$
$V_{RAM}$	RAM Standby Voltage		In stop mode	2.0			V

## NOTE:

1. Referenced to  $V_{CC} = 2.7$  to  $3.6V$ ,  $V_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^\circ C$  /  $-40$  to  $85^\circ C$ ,  $f(BCLK) = 10MHz$  unless otherwise specified.

$V_{CC} = 3V$ **Table 18.24. Electrical Characteristics (2) (1)**

Symbol	Parameter	Measurement Condition	Standard			Unit		
			Min.	Typ.	Max.			
I <sub>CC</sub>	Power Supply Current (V <sub>CC</sub> = 2.7 to 3.6V)	Output pins are left open and other pins are connected to V <sub>SS</sub>	Mask ROM	f(BCLK) = 10 MHz, Main clock, no division		7	10	mA
				On-chip oscillator operates, f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz		1		mA
			Flash memory	f(BCLK) = 10 MHz, Main clock, no division		7	12	mA
				On-chip oscillator operates, f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz		1		mA
			Flash memory program	f(BCLK) = 10 MHz, V <sub>CC</sub> = 3.0 V		10		mA
			Flash memory erase	f(BCLK) = 10 MHz, V <sub>CC</sub> = 3.0 V		11		mA
			Mask ROM	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on ROM <sup>(3)</sup>		25		μA
				On-chip oscillator operates, f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz, In wait mode		25		μA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM <sup>(3)</sup>		25		μA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory <sup>(3)</sup>		450		μA
				On-chip oscillator operates, f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz, In wait mode		45		μA
			Mask ROM, Flash memory	f(BCLK) = 32 kHz, In wait mode <sup>(2)</sup> , Oscillation capacity HIGH		10		μA
				f(BCLK) = 32 kHz, In wait mode <sup>(2)</sup> , Oscillation capacity LOW		3		μA
		While clock stops, Topr = 25° C		0.7	3	μA		
I <sub>det4</sub>	Low voltage detection dissipation current <sup>(4)</sup>			0.6	4	μA		
I <sub>det3</sub>	Reset level detection dissipation current <sup>(4)</sup>			1.0	5	μA		

## NOTES:

1. Referenced to V<sub>CC</sub> = 2.7 to 3.6 V, V<sub>SS</sub> = 0 V at Topr = -20 to 85 ° C / -40 to 85 ° C, f(BCLK) = 10 MHz unless otherwise specified.
2. With one timer operates, using f<sub>csz</sub>.
3. This indicates the memory in which the program to be executed exists.
4. I<sub>det</sub> is dissipation current when the following bit is set to 1 (detection circuit enabled).  
I<sub>det4</sub>: the VC27 bit of the VCR2 register  
I<sub>det3</sub>: the VC26 bit in the VCR2 register

$$V_{CC} = 3V$$

**Timing Requirements****( $V_{CC} = 3V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)****Table 18.25. External Clock Input (XIN input)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External Clock Input Cycle Time	100		ns
tw(H)	External Clock Input High ("H") Width	40		ns
tw(L)	External Clock Input Low ("L") Width	40		ns
tr	External Clock Rise Time		18	ns
tf	External Clock Fall Time		18	ns

$$V_{CC} = 3V$$

### Timing Requirements

( $V_{CC} = 3V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)

**Table 18.26. 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 18.27. 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 18.28. 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 18.29. 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 18.30. 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 18.31. 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		$\mu s$
$t_{su(TAiN-TAiOUT)}$	TAiOut input setup time	500		ns
$t_{su(TAiOUT-TAiN)}$	TAiIn input setup time	500		ns

$$V_{CC} = 3V$$

### Timing Requirements

( $V_{CC} = 3V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)

**Table 18.32. 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	X.	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 18.33. 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 18.34. Timer B Input (Pulse Width Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIn input cycle time	600	X.	ns
$t_{w(TBH)}$	TBiIn input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width	300		ns

**Table 18.35. A/D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	ADTRG input cycle time (triggerable minimum)	1500	X.	ns
$t_{w(ADL)}$	ADTRG input LOW pulse width	200		ns

**Table 18.36. Serial I/O**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	100		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

**Table 18.37. External Interrupt INTi Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	INTi input HIGH pulse width	380	X.	ns
$t_{w(INL)}$	INTi input LOW pulse width	380		ns

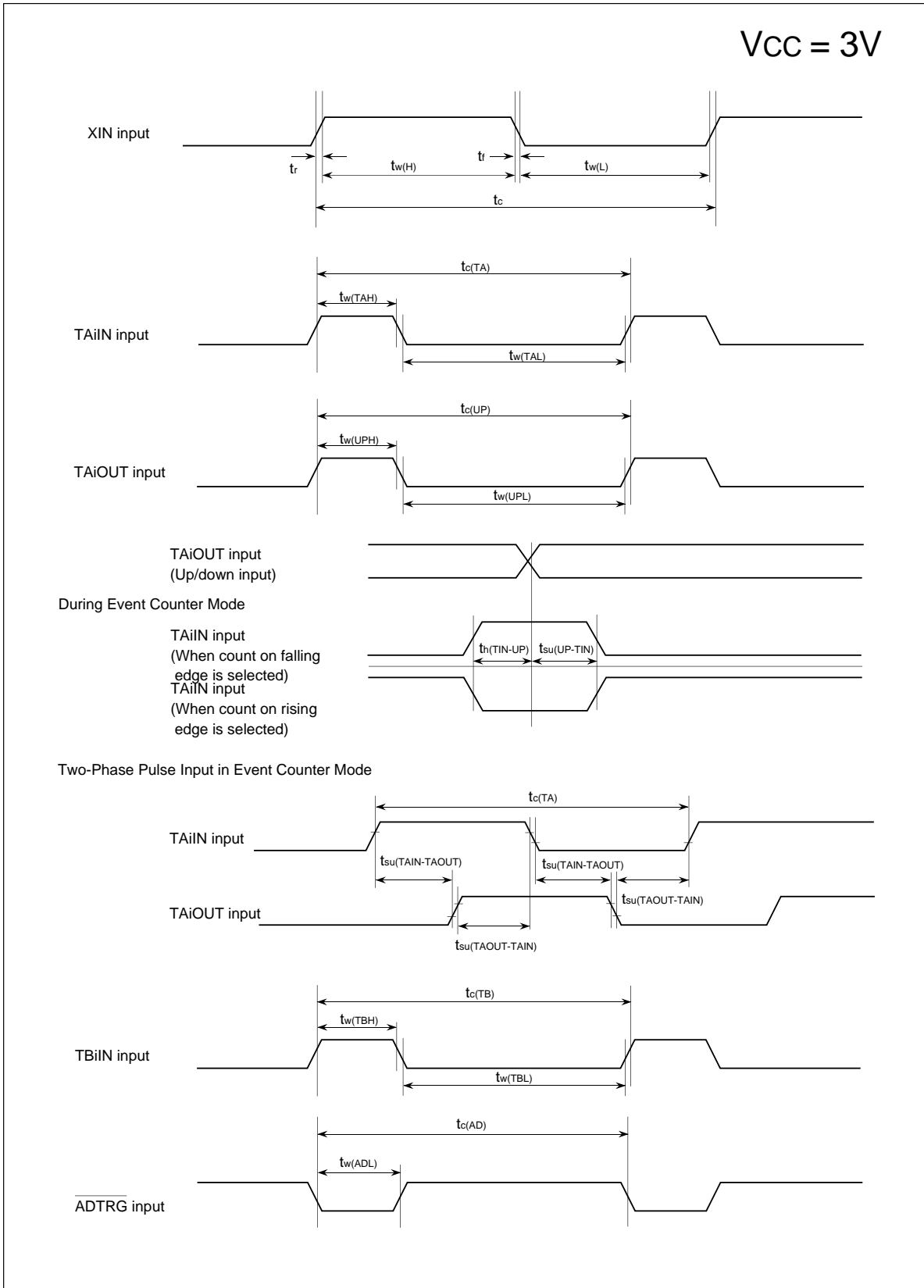


Figure 18.3. Timing Diagram (1)

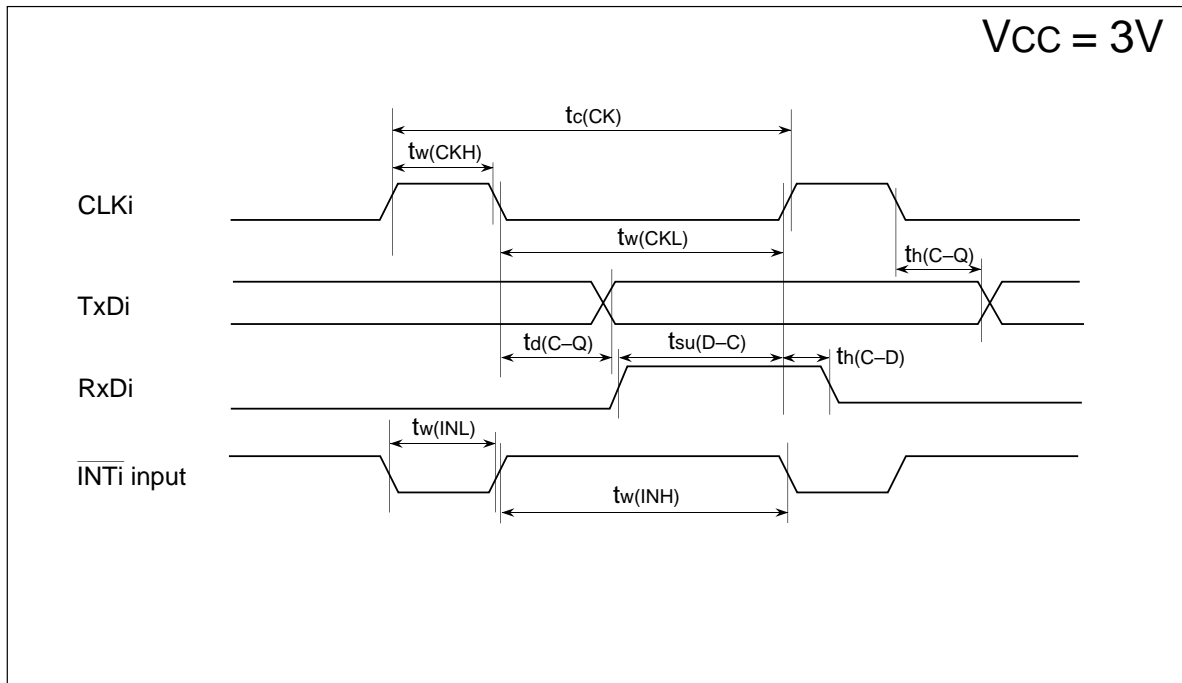


Figure 18.4. Timing Diagram (2)



## 18.2. M16C/26T (T version)

Table 18.38. Absolute Maximum Ratings

Symbol	Parameter		Condition	Value	Unit
V <sub>CC</sub>	Supply Voltage		V <sub>CC</sub> = AV <sub>CC</sub>	-0.3 to 6.5	V
AV <sub>CC</sub>	Analog Supply Voltage		V <sub>CC</sub> = AV <sub>CC</sub>	-0.3 to 6.5	V
V <sub>I</sub>	Input Voltage	P1 <sub>5</sub> to P1 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>3</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>IN</sub> , V <sub>REF</sub> , RESET, CNV <sub>SS</sub>		-0.3 to V <sub>CC</sub> +0.3	V
V <sub>O</sub>	Output Voltage	P1 <sub>5</sub> to P1 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>3</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>OUT</sub>		-0.3 to V <sub>CC</sub> +0.3	V
P <sub>d</sub>	Power Dissipation		-40 ≤ T <sub>opr</sub> ≤ 85° C	300	mW
T <sub>opr</sub>	Operating Ambient Temperature	during CPU operation		-40 to 85	° C
		during flash memory program and erase operation	Program Space (Block 0 to Block 3)	0 to 60	° C
			Data Space (Block A, Block B)	-40 to 85	° C
T <sub>stg</sub>	Storage Temperature			-65 to 150	° C

**Table 18.39. Recommended Operating Conditions (1)**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V <sub>CC</sub>	Supply Voltage		3.0		5.5	V
AV <sub>CC</sub>	Analog Supply Voltage			V <sub>CC</sub>		V
V <sub>SS</sub>	Supply Voltage			0		V
AV <sub>SS</sub>	Analog Supply Voltage			0		V
V <sub>IH</sub>	Input High ("H") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	0.7 V <sub>CC</sub>		V <sub>CC</sub>	V
		XIN, RESET, CNVSS	0.8 V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	Input Low ("L") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	0		0.3 V <sub>CC</sub>	V
		XIN, RESET, CNVSS	0		0.2V <sub>CC</sub>	V
I <sub>OH(peak)</sub>	Peak Output High ("H") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107			-10.0	mA
I <sub>OH(avg)</sub>	Average Output High ("H") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107			-5.0	mA
I <sub>OL(peak)</sub>	Peak Output Low ("L") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107			10.0	mA
I <sub>OL(avg)</sub>	Average Output Low ("L") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107			5.0	mA
f(X <sub>IN</sub> )	Main Clock Input Oscillation Frequency <sup>(4)</sup>		0		20	MHz
f(X <sub>CIN</sub> )	Sub Clock Oscillation Frequency			32.768	50	kHz
f <sub>1</sub> (ROC)	On-chip Oscillator Frequency 1		0.5	1	2	MHz
f <sub>2</sub> (ROC)	On-chip Oscillator Frequency 2		1	2	4	MHz
f <sub>3</sub> (ROC)	On-chip Oscillator Frequency 3		8	16	26	MHz
f(PLL)	PLL Clock Oscillation Frequency <sup>(4)</sup>		10		20	MHz
f(BCLK)	CPU Operation Clock Frequency		0		20	MHz
tsu(PLL)	Wait Time to Stabilize PLL Frequency Synthesizer		V <sub>CC</sub> = 5.0 V		20	ms
			V <sub>CC</sub> = 3.0 V		50	ms

## NOTES:

1. Referenced to V<sub>CC</sub> = 3.0 to 5.5 V at T<sub>opr</sub> = -40 to 85 °C unless otherwise specified.
2. The mean output current is the mean value within 100 ms.
3. The total I<sub>OL(peak)</sub> for all ports must be 80 mA or less. The total I<sub>OH(peak)</sub> for all ports must be -80 mA or less.
4. Relationship among main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.

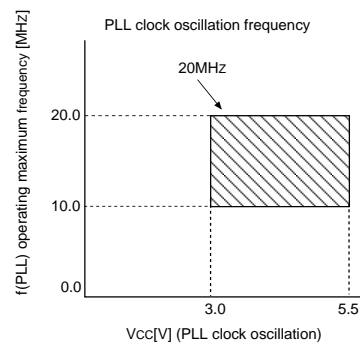
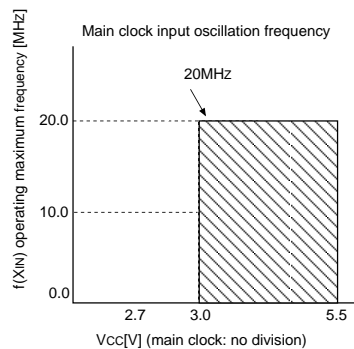


Table 18.40. A/D Conversion Characteristics (1)

Symbol	Parameter		Measurement Condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution		$V_{REF} = V_{CC}$			10	Bits
INL	Integral Nonlinearity Error	10 bit	$V_{REF} = V_{CC} = 5\text{ V}$			$\pm 3$	LSB
			$V_{REF} = V_{CC} = 3.3\text{ V}$			$\pm 5$	LSB
	Absolute Accuracy	8 bit	$V_{REF} = V_{CC} = 3.3\text{ V}, 5\text{ V}$			$\pm 2$	LSB
			$V_{REF} = V_{CC} = 5\text{ V}$			$\pm 3$	LSB
		10 bit	$V_{REF} = V_{CC} = 3.3\text{ V}$			$\pm 5$	LSB
		8 bit	$V_{REF} = V_{CC} = 3.3\text{ V}, 5\text{ V}$			$\pm 2$	LSB
DNL	Differential Nonlinearity Error					$\pm 1$	LSB
-	Offset Error					$\pm 3$	LSB
-	Gain Error					$\pm 3$	LSB
R <sub>LADDER</sub>	Resistor Ladder		$V_{REF} = V_{CC}$	10		40	k $\Omega$
t <sub>CONV</sub>	10-bit Conversion Time Sample & Hold Function Available		$V_{REF} = V_{CC} = 5\text{ V}, \phi_{AD} = 10\text{ MHz}$	3.3			$\mu\text{s}$
t <sub>CONV</sub>	8-bit Conversion Time Sample & Hold Function Available		$V_{REF} = V_{CC} = 5\text{ V}, \phi_{AD} = 10\text{ MHz}$	2.8			$\mu\text{s}$
V <sub>REF</sub>	Reference Voltage			2.0		V <sub>CC</sub>	V
V <sub>IA</sub>	Analog Input Voltage			0		V <sub>REF</sub>	V

## NOTES:

1. Referenced to  $V_{CC} = AV_{CC} = V_{REF} = 3.3$  to  $5.5\text{ V}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$  at  $T_{opr} = -40$  to  $85^\circ\text{ C}$  unless otherwise specified.
2. Keep  $\phi_{AD}$  frequency at 10 MHz or less. Additionally, divide the  $f_{AD}$  if  $V_{CC}$  is less than 4.2 V, and make  $\phi_{AD}$  frequency equal to or lower than  $f_{AD}/2$ .
3. When sample & hold function is disabled, keep  $\phi_{AD}$  frequency at 250 kHz or more in addition to the limitation in Note 2. When sample & hold function is enabled, keep  $\phi_{AD}$  frequency at 1 MHz or more in addition to the limitation in Note 2.
4. When sample & hold function is enabled, sampling time is  $3/\phi_{AD}$  frequency. When sample & hold function is disabled, sampling time is  $2/\phi_{AD}$  frequency.

**Table 18.41. Flash Memory Version Electrical Characteristics (1):  
Program Space and Data Space for U3, Program Space for U7**

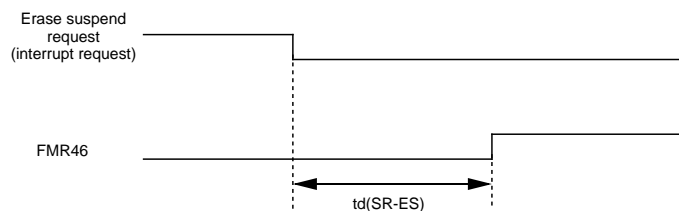
Symbol	Parameter	Standard			Unit
		Min.	Typ. <sup>(2)</sup>	Max.	
-	Program and Erase Endurance <sup>(3)</sup>	100/1000 <sup>(4, 11)</sup>			cycles
-	Word Program Time (V <sub>CC</sub> = 5.0 V, T <sub>opr</sub> = 25° C)		75	600	μs
-	Block Erase Time (V <sub>CC</sub> = 5.0 V, T <sub>opr</sub> = 25° C)	2-Kbyte Block	0.2	9	s
		8-Kbyte Block	0.4	9	s
		16-Kbyte Block	0.7	9	s
		32-Kbyte Block	1.2	9	s
td(SR-ES)	Duration between Suspend Request and Erase Suspend			8	ms
t <sub>ps</sub>	Wait Time to Stabilize Flash Memory Circuit			15	μs
-	Data Hold Time <sup>(5)</sup>	20			years

**Table 18.42. Flash Memory Version Electrical Characteristics (6): Data Space for U7<sup>(7)</sup>**

Symbol	Parameter	Standard			Unit
		Min.	Typ. <sup>(2)</sup>	Max.	
-	Program and Erase Endurance <sup>(3, 8, 9)</sup>	10000 <sup>(4, 10)</sup>			cycles
-	Word Program Time (V <sub>CC</sub> = 5.0 V, T <sub>opr</sub> = 25° C)		100		μs
-	Block Erase Time (V <sub>CC</sub> = 5.0V, T <sub>opr</sub> = 25° C) (2-Kbyte block)		0.3		s
td(SR-ES)	Duration between Suspend Request and Erase Suspend			8	ms
t <sub>ps</sub>	Wait Time to Stabilize Flash Memory Circuit			15	μs
-	Data Hold Time <sup>(5)</sup>	20			years

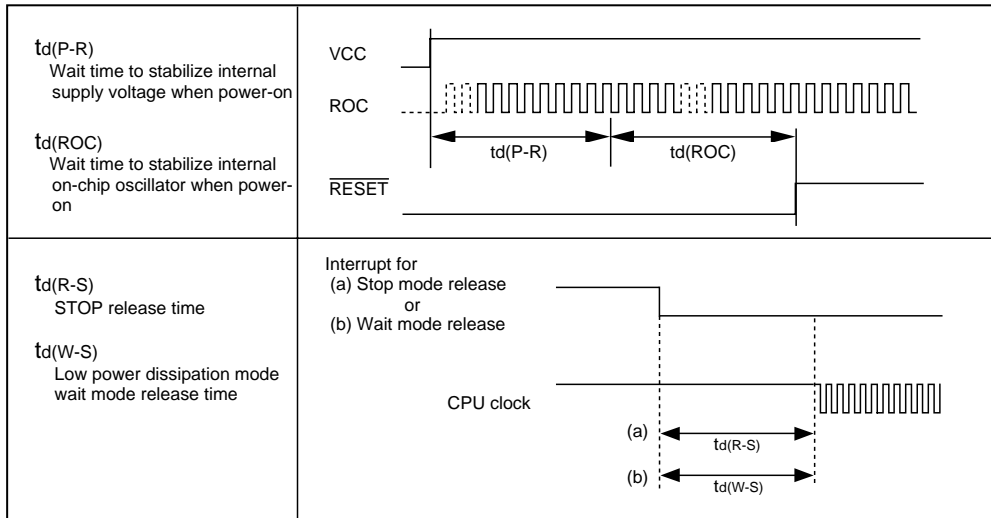
**NOTES:**

1. Referenced to V<sub>CC</sub> = 3.0 to 5.5 V at T<sub>opr</sub> = 0 to 60° C (program space)/ T<sub>opr</sub> = -40 to 85° C (data space), unless otherwise specified.
2. V<sub>CC</sub> = 5.0 V; T<sub>opr</sub> = 25° C
3. Program and erase endurance is defined as number of program-erase cycles per block.  
If program and erase endurance is *n* cycle (*n* = 100, 1000, 10000), each block can be erased and programmed *n* cycles.  
For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guaranteed (1 to minimum value are guaranteed).
5. T<sub>opr</sub> = 55° C
6. Referenced to V<sub>CC</sub> = 3.0 to 5.5 V at T<sub>opr</sub> = -40 to 85° C unless otherwise specified.
7. **Table 18.42** applies for data space in U7 when program and erase endurance is more than 1,000 cycles. Otherwise, use **Table 18.41**.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. If an erase error is generated during block erase, execute the clear status register command and block erase command at least 3 times until an erase error is not generated.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register to 1 (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3; 1,000 cycles for program space in U7.
12. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for further details on the E/W failure rate.



**Table 18.43. Power Supply Circuit Timing Characteristics**

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Wait Time to Stabilize Internal Supply Voltage when Power-on	Vcc = 3.0 to 5.5V			2	ms
td(ROC)	Wait Time to Stabilize Internal On-chip Oscillator when Power-on				40	μs
td(R-S)	STOP Release Time <sup>(1)</sup>				1.5	ms
td(W-S)	Low Power Dissipation Mode Wait Mode Release Time				250	μs



$V_{CC} = 5V$ 

Table 18.44. Electrical Characteristics (1)

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
$V_{OH}$	Output High ("H") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OH} = -5 \text{ mA}$	$V_{CC}-2.0$		$V_{CC}$	V
$V_{OH}$	Output High ("H") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OH} = -200 \mu\text{A}$	$V_{CC}-0.3$		$V_{CC}$	V
$V_{OH}$	Output High ("H") Voltage	$X_{OUT}$	High Power	$I_{OH} = -1 \text{ mA}$	$V_{CC}-2.0$	$V_{CC}$	V
			Low Power	$I_{OH} = -0.5 \text{ mA}$	$V_{CC}-2.0$	$V_{CC}$	
	Output High ("H") Voltage	$X_{COUT}$	High Power	No load applied		2.5	V
			Low Power	No load applied		1.6	
$V_{OL}$	Output Low ("L") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OL} = 5 \text{ mA}$			2.0	V
$V_{OL}$	Output Low ("L") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OL} = 200 \mu\text{A}$			0.45	V
$V_{OL}$	Output Low ("L") Voltage	$X_{OUT}$	High Power	$I_{OL} = 1 \text{ mA}$		2.0	V
			Low Power	$I_{OL} = 0.5 \text{ mA}$		2.0	
	Output Low ("L") Voltage	$X_{COUT}$	High Power	No load applied		0	V
			Low Power	No load applied		0	
$V_{T+}-V_{T-}$	Hysteresis	$\overline{TA0IN}-\overline{TA4IN}$ , $\overline{TB0IN}-\overline{TB2IN}$ , $\overline{INT0}-\overline{INT5}$ , $\overline{NMI}$ , $\overline{ADTRG}$ , $\overline{CTS0}-\overline{CTS2}$ , $\overline{CLK0}-\overline{CLK2}$ , $\overline{TA2OUT}-\overline{TA4OUT}$ , $\overline{Kl0}-\overline{Kl3}$ , $\overline{RXD0}-\overline{RXD2}$		0.2		1.0	V
$V_{T+}-V_{T-}$	Hysteresis	RESET		0.2		2.5	V
$V_{T+}-V_{T-}$	Hysteresis	$X_{IN}$		0.2		0.8	V
$I_{IH}$	Input High ("H") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107 $X_{IN}$ , RESET, $CNV_{SS}$	$V_I = 5 \text{ V}$			5.0	$\mu\text{A}$
$I_{IL}$	Input Low ("L") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107 $X_{IN}$ , RESET, $CNV_{SS}$	$V_I = 0 \text{ V}$			-5.0	$\mu\text{A}$
$R_{PULLUP}$	Pull-up Resistance	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$V_I = 0 \text{ V}$	30	50	170	$k\Omega$
$R_{fXIN}$	Feedback Resistance	$X_{IN}$			1.5		$M\Omega$
$R_{fXCIN}$	Feedback Resistance	$X_{CIN}$			15		$M\Omega$
$V_{RAM}$	RAM Standby Voltage		In stop mode	2.0			V

NOTE:

1. Referenced to  $V_{CC} = 4.2$  to  $5.5 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$  at  $T_{opr} = -40$  to  $85^\circ \text{ C}$ ,  $f(\text{BCLK}) = 20 \text{ MHz}$  unless otherwise specified.

$V_{CC} = 5V$ **Table 18.45. Electrical Characteristics (2) <sup>(1)</sup>**

Symbol	Parameter	Measurement Condition	Standard			Unit		
			Min.	Typ.	Max.			
I <sub>CC</sub>	Power Supply Current (V <sub>CC</sub> =4.0 to 5.5V)	Output pins are left open and other pins are connected to V <sub>SS</sub>	Flash memory	f(BCLK) = 20 MHz, Main clock, no division		16	19	mA
				On-chip oscillator operates, f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz		1		mA
			Flash memory program	f(BCLK) = 10 MHz, V <sub>CC</sub> = 5.0 V		11		mA
			Flash memory erase	f(BCLK) = 10 MHz, V <sub>CC</sub> = 5.0 V		12		mA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM <sup>(3)</sup>		25		μA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory <sup>(3)</sup>		450		μA
			On-chip oscillation, f <sub>2(ROC)</sub> selected, f(BCLK) = 1 MHz, In wait mode			50		μA
				f(BCLK) = 32 kHz, In wait mode <sup>(2)</sup> , Oscillation capacity HIGH		10		μA
			f(BCLK) = 32 kHz, In wait mode <sup>(2)</sup> , Oscillation capacity LOW			3		μA
			While clock stops, T <sub>opr</sub> = 25° C			0.8	3	μA

## NOTES:

1. Referenced to V<sub>CC</sub> = 4.2 to 5.5 V, V<sub>SS</sub> = 0 V at T<sub>opr</sub> = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using f<sub>ca2</sub>.
3. This indicates the memory in which the program to be executed exists.

$$V_{CC} = 5V$$

### Timing Requirements

( $V_{CC} = 5V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -40$  to  $85^{\circ}C$  unless otherwise specified)

**Table 18.46. External Clock Input (XIN input)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_c$	External clock input cycle time	50		ns
$t_{w(H)}$	External clock input HIGH pulse width	20		ns
$t_{w(L)}$	External clock input LOW pulse width	20		ns
$t_r$	External clock rise time		9	ns
$t_f$	External clock fall time		9	ns



$V_{CC} = 5V$

**Timing Requirements****( $V_{CC} = 5V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -40$  to  $85^{\circ}C$  unless otherwise specified)****Table 18.47. 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 18.48. 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 18.49. 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 18.50. 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 18.51. 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 18.52. Timer A Input (Two-phase Pulse Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIn input cycle time	800		ns
$t_{su(TAiN-TAiOUT)}$	TAiOUT input setup time	200		ns
$t_{su(TAiOUT-TAiN)}$	TAiIn input setup time	200		ns

$V_{CC} = 5V$ **Timing Requirements****( $V_{CC} = 5V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -40$  to  $85^{\circ}C$  unless otherwise specified)****Table 18.53. 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 18.54. 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 18.55. 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 18.56. A/D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input LOW pulse width	125		ns

**Table 18.57. Serial I/O**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	70		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

**Table 18.58. 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

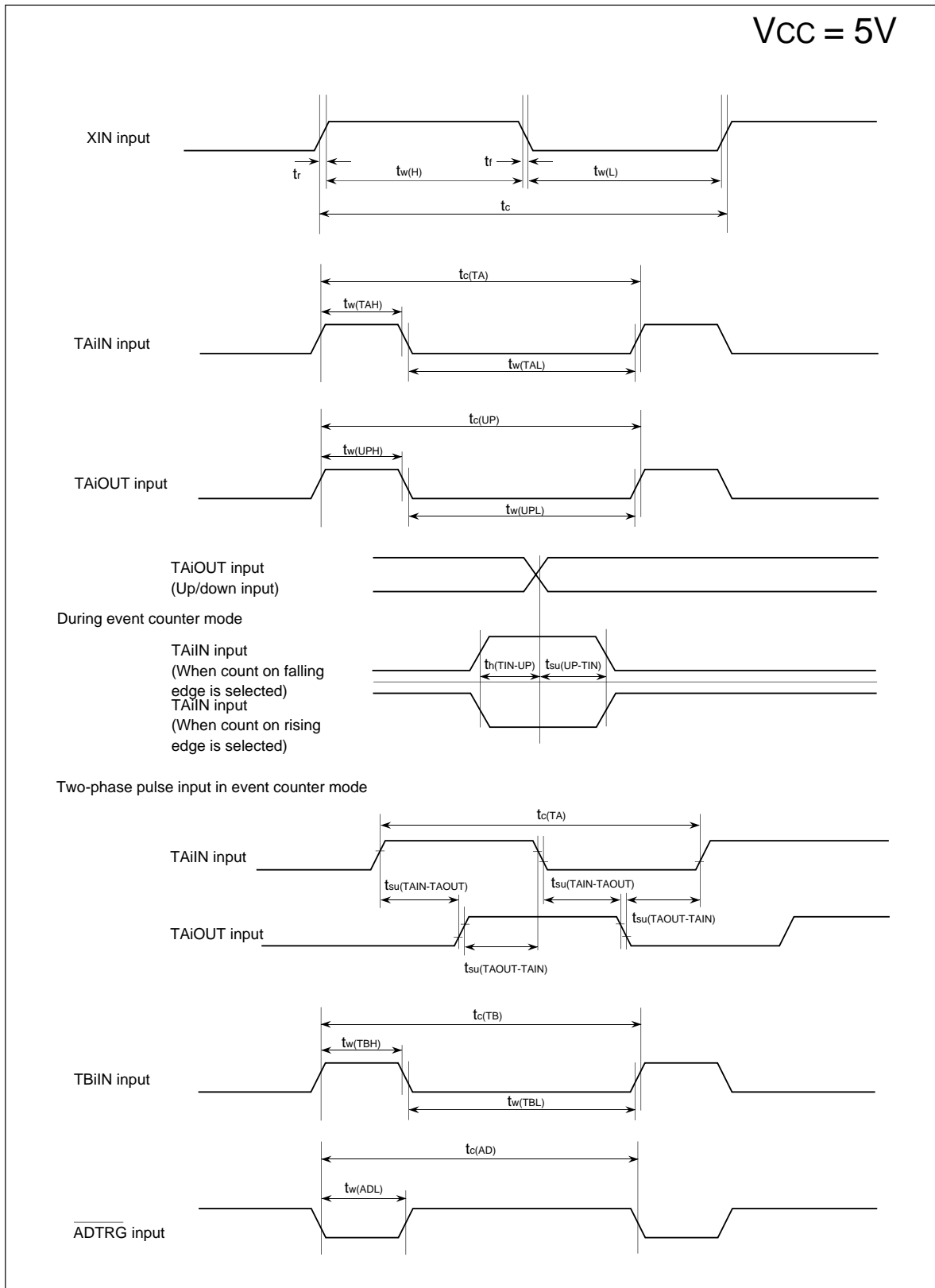


Figure 18.5. Timing Diagram (1)

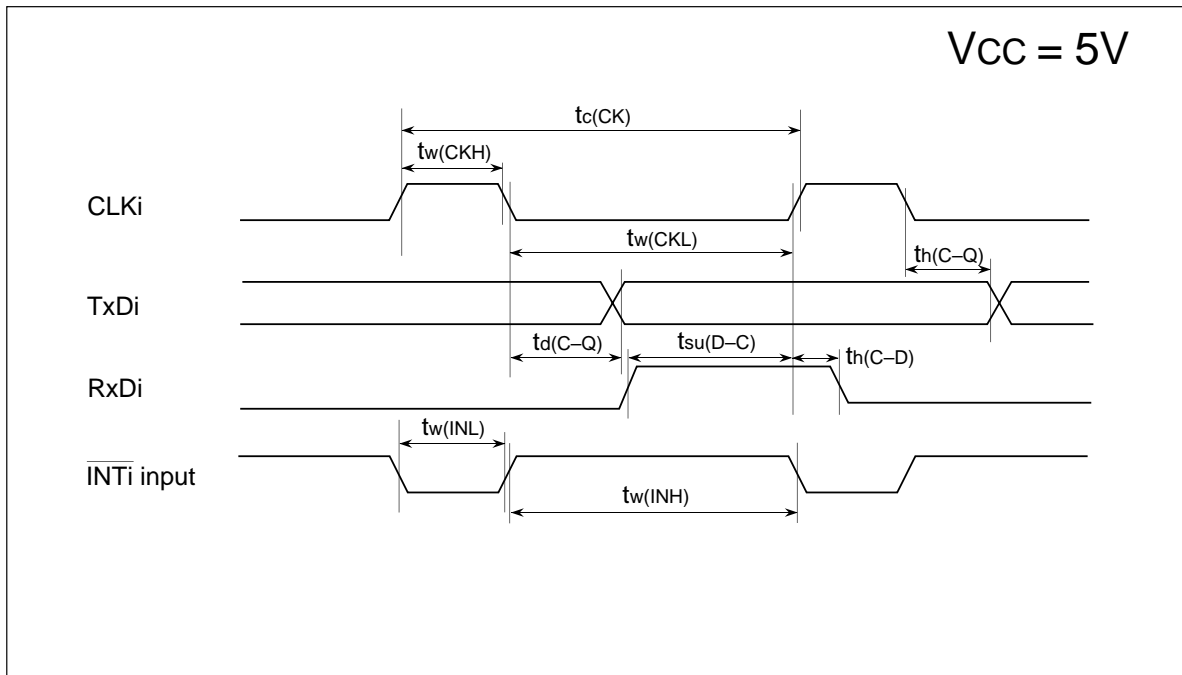


Figure 18.6. Timing Diagram (2)

$V_{CC} = 3V$ **Table 18.59. Electrical Characteristics (1)**

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
$V_{OH}$	Output High ("H") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OH} = -1 \text{ mA}$	$V_{CC}-0.5$		$V_{CC}$	V
$V_{OH}$	Output High ("H") Voltage	$X_{OUT}$	High Power	$I_{OH} = -0.1 \text{ mA}$	$V_{CC}-0.5$	$V_{CC}$	V
			Low Power	$I_{OH} = -50 \mu\text{A}$	$V_{CC}-0.5$	$V_{CC}$	
	Output High ("H") Voltage	$X_{COUT}$	High Power	No load applied		2.5	V
			Low Power	No load applied		1.6	
$V_{OL}$	Output Low ("L") Voltage	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$I_{OL} = 1 \text{ mA}$			0.5	V
$V_{OL}$	Output Low ("L") Voltage	$X_{OUT}$	High Power	$I_{OL} = 0.1 \text{ mA}$		0.5	V
			Low Power	$I_{OL} = 50 \mu\text{A}$		0.5	
	Output Low ("L") Voltage	$X_{COUT}$	High Power	No load applied		0	V
			Low Power	No load applied		0	
$V_{T+}-V_{T-}$	Hysteresis	$\overline{TA0IN-TA4IN}$ , $\overline{TB0IN-TB2IN}$ , $\overline{INT0-INT5}$ , $\overline{NMI}$ , $\overline{ADTRG}$ , $\overline{CTS0-CTS2}$ , $\overline{CLK0-CLK2}$ , $\overline{TA2OUT-TA4OUT}$ , $\overline{KI0-KI3}$ , $\overline{RXD0-RXD2}$				0.8	V
$V_{T+}-V_{T-}$	Hysteresis	$\overline{RESET}$				1.8	V
$V_{T+}-V_{T-}$	Hysteresis	$X_{IN}$				0.8	V
$I_{IH}$	Input High ("H") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107 $X_{IN}$ , $\overline{RESET}$ , $CNV_{SS}$	$V_I = 3 \text{ V}$			4.0	$\mu\text{A}$
$I_{IL}$	Input Low ("L") Current	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107 $X_{IN}$ , $\overline{RESET}$ , $CNV_{SS}$	$V_I = 0 \text{ V}$			-4.0	$\mu\text{A}$
$R_{PULLUP}$	Pull-up Resistance	P15 to P17, P60 to P67, P70 to P77, P80 to P87, P90 to P93, P100 to P107	$V_I = 0 \text{ V}$	50	100	500	$\text{k}\Omega$
$R_{fXIN}$	Feedback Resistance	$X_{IN}$			3.0		$\text{M}\Omega$
$R_{fXCIN}$	Feedback Resistance	$X_{CIN}$			25		$\text{M}\Omega$
$V_{RAM}$	RAM Standby Voltage		In stop mode	2.0			V

NOTE:

1. Referenced to  $V_{CC} = 3.0$  to  $3.6 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$  at  $T_{op} = -40$  to  $85 \text{ }^\circ\text{C}$ ,  $f(\text{BCLK}) = 20 \text{ MHz}$  unless otherwise specified.

$V_{CC} = 3V$ Table 18.60. Electrical Characteristics (2) <sup>(1)</sup>

Symbol	Parameter	Measurement Condition	Standard			Unit		
			Min.	Typ.	Max.			
I <sub>CC</sub>	Power Supply Current (V <sub>CC</sub> =3.0 to 3.6V)	Output pins are left open and other pins are connected to V <sub>SS</sub>	Flash memory	f(BCLK) = 10 MHz, Main clock, no division		7	12	mA
				On-chip oscillator operates, f <sub>2(RCC)</sub> selected, f(BCLK) = 1 MHz		1		mA
			Flash memory program	f(BCLK) = 10 MHz, V <sub>CC</sub> = 3.0 V		10		mA
			Flash memory erase	f(BCLK) = 10MHz, V <sub>CC</sub> = 3.0 V		11		mA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM <sup>(3)</sup>		25		μA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory <sup>(3)</sup>		450		μA
			Flash memory	On-chip oscillator operates, f <sub>2(RCC)</sub> selected, f(BCLK) = 1 MHz, In wait mode		45		μA
				f(BCLK) = 32 kHz, In wait mode <sup>(2)</sup> , Oscillation capacity HIGH		10		μA
			f(BCLK) = 32 kHz, In wait mode <sup>(2)</sup> , Oscillation capacity LOW		3		μA	
			While clock stops, T <sub>opr</sub> = 25° C		0.7	3	μA	

## NOTES:

1. Referenced to V<sub>CC</sub> = 3.0 to 3.6 V, V<sub>SS</sub> = 0 V at T<sub>opr</sub> = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using fc<sub>32</sub>.
3. This indicates the memory in which the program to be executed exists.

$$V_{CC} = 3V$$

### Timing Requirements

( $V_{CC} = 3V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -40$  to  $85^{\circ}C$  unless otherwise specified)

**Table 18.61. External Clock Input (XIN input)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_c$	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
$t_r$	External clock rise time		18	ns
$t_f$	External clock fall time		18	ns

$$V_{CC} = 3V$$

### Timing Requirements

( $V_{CC} = 3V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -40$  to  $85^{\circ}C$  unless otherwise specified)

**Table 18.62. 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 18.63. 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 18.64. 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 18.65. 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 18.66. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiOut input cycle time	3000		ns
$t_{w(TAH)}$	TAiOut input HIGH pulse width	1500		ns
$t_{w(TAL)}$	TAiOut input LOW pulse width	1500		ns
$t_{su(TA)}$	TAiOut input setup time	600		ns
$t_{h(TA)}$	TAiOut input hold time	600		ns

**Table 18.67. 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		$\mu s$
$t_{su(TA)}$	TAiOut input setup time	500		ns
$t_{su(TA)}$	TAiIn input setup time	500		ns



$$V_{CC} = 3V$$

### Timing Requirements

( $V_{CC} = 3V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -40$  to  $85^{\circ}C$  unless otherwise specified)

**Table 18.68. 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 18.69. 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 18.70. 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 18.71. 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 18.72. Serial I/O**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	100		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

**Table 18.73. 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

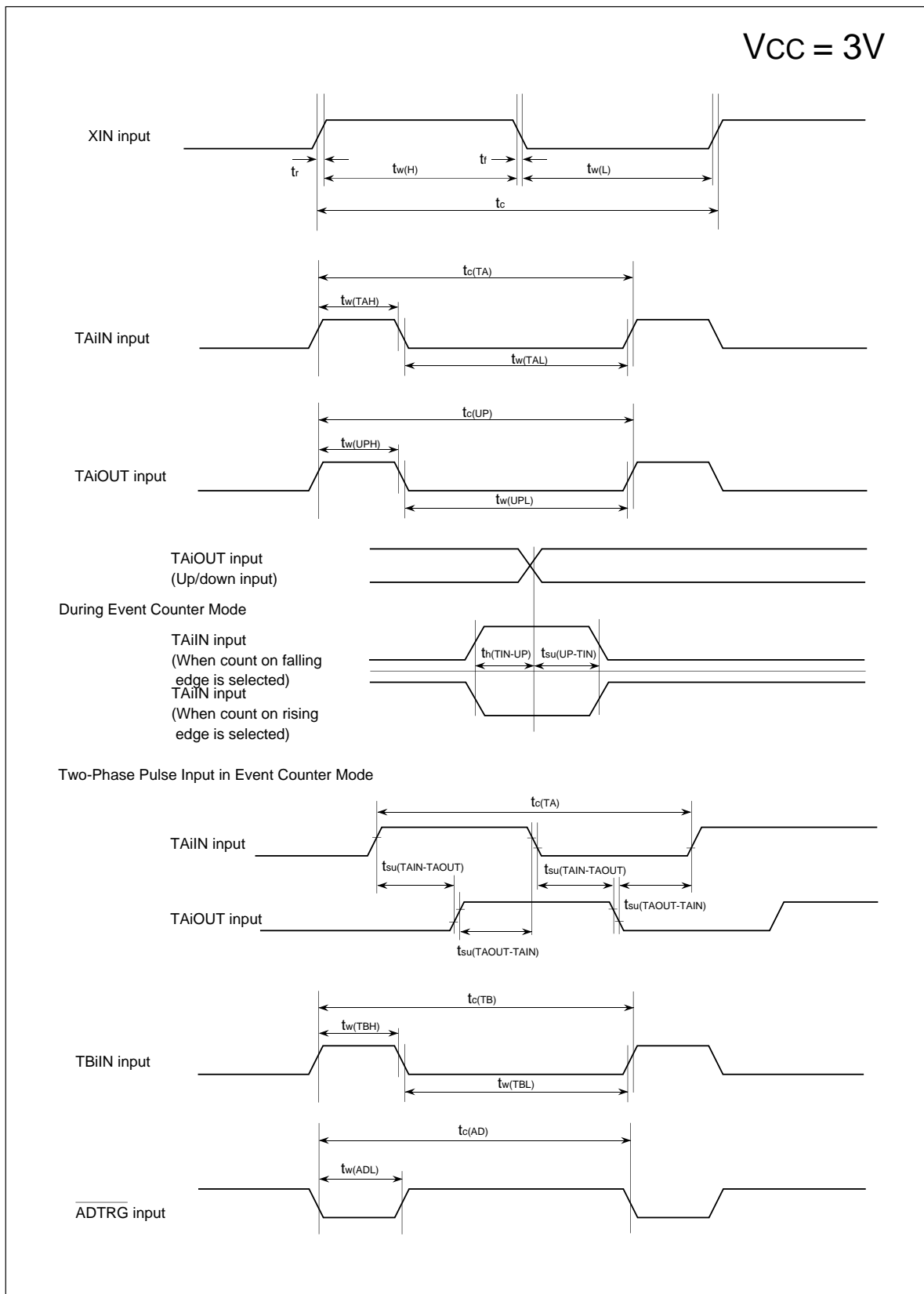


Figure 18.7. Timing Diagram (1)

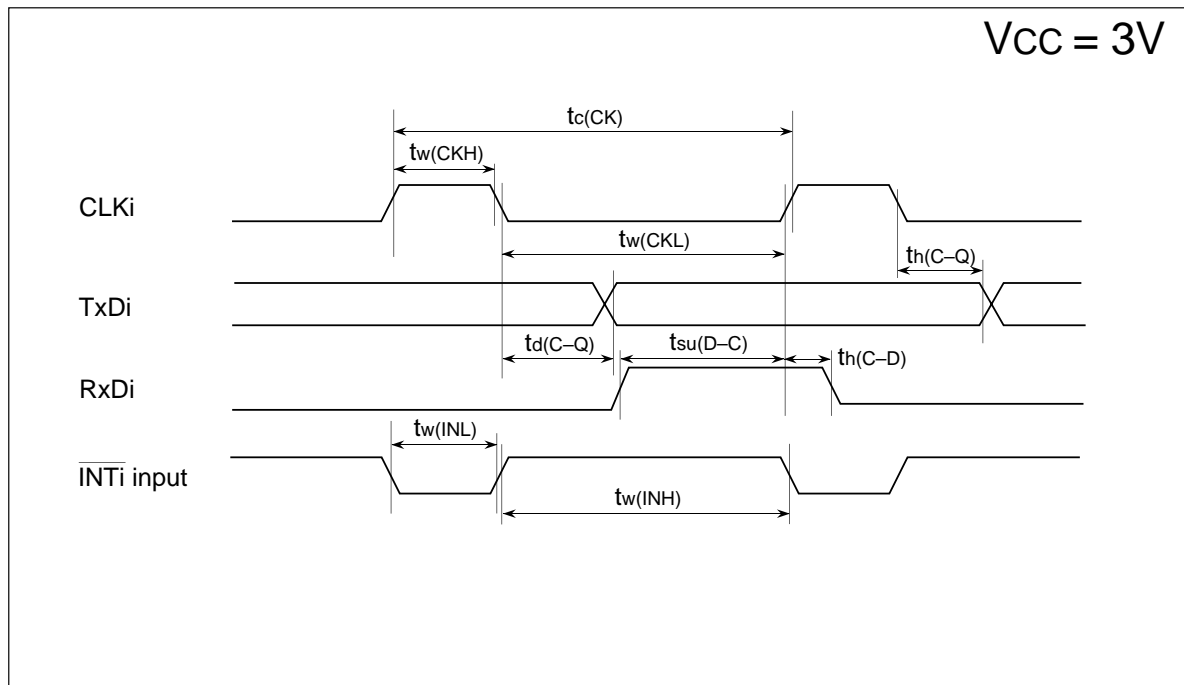


Figure 18.8. Timing Diagram (2)

## 19. Usage Notes

### 19.1 SFR

#### 19.1.1 Precaution for 48-pin package

Set the IFSR20 bit in the IFSR2A register to "1" after reset and set the PACR2 to PACR0 bits in the PACR register to "1002".

#### 19.1.2 Precaution for 42-pin package

Set the IFSR20 bit in the IFSR2A register to "1" after reset and set the PACR2 to PACR0 bits in the PACR register to "0012".

#### 19.1.3 Register Setting

Immediate values should be set in the registers containing write-only bits. When establishing a new value by modifying a previous value, write the previous value into RAM as well as the register. Change the contents of the RAM and then transfer the new value to the register.

## 19.2 PLL Frequency Synthesizer

Stabilize supply voltage so that the standard of the power supply ripple is met.

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
$f(\text{ripple})$	Power supply ripple allowable frequency( $V_{CC}$ )			10	kHz
$V_{p-p}(\text{ripple})$	Power supply ripple allowable amplitude voltage	( $V_{CC}=5V$ )		0.5	V
		( $V_{CC}=3V$ )		0.3	V
$V_{CC}(\Delta V/\Delta T)$	Power supply ripple rising/falling gradient	( $V_{CC}=5V$ )		0.3	V/ms
		( $V_{CC}=3V$ )		0.3	V/ms

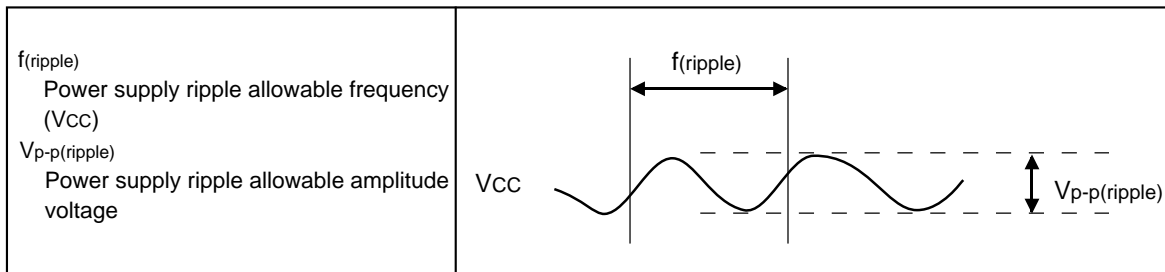


Figure 19.1 Timing of Voltage Fluctuation

### 19.3 Power Control

1. When exiting stop mode by hardware reset, the device will startup using the on-chip oscillator.
2. Set the MR0 bit in the TAI MR register(i=0 to 4) to "0"(pulse is not output) to use the timer A to exit stop mode.
3. When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not excute any instructions which can generate a write to RAM between the JMP.B and WAIT instructions. Disable the DMA transfers, if a DMA transfer may occur between the JMP.B and WAIT instructions. After the WAIT instruction, insert at least 4 NOP instructions. When entering wait mode, the instruction queue reads ahead the instructions following WAIT, and depending on timing, some of these may execute before the microcomputer enters wait mode.

Program example when entering wait mode

```

Program Example:    JMP.B    L1    ; Insert JMP.B instruction before WAIT instruction
                   L1:
                   FSET     I    ;
                   WAIT     ; Enter wait mode
                   NOP      ; More than 4 NOP instructions
                   NOP
                   NOP
                   NOP

```

4. When entering stop mode, insert a JMP.B instruction immediately after executing an instruction which sets the CM10 bit in the CM1 register to "1", and then insert at least 4 NOP instructions. When entering stop mode, the instruction queue reads ahead the instructions following the instruction which sets the CM10 bit to "1" (all clock stops), and, some of these may execute before the microcomputer enters stop mode or before the interrupt routine for returning from stop mode.

Program example when entering stop mode

```

Program Example:    FSET     I
                   BSET     CM10 ; Enter stop mode
                   JMP.B    L1    ; Insert JMP.B instruction
                   L1:
                   NOP      ; More than 4 NOP instructions
                   NOP
                   NOP
                   NOP

```

5. Wait until the main clock oscillation stabilization time, before switching the CPU clock source to the main clock.

Similarly, wait until the sub clock oscillates stably before switching the CPU clock source to the sub clock.

6. Suggestions to reduce power consumption

**(a) Ports**

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A dash current may flow through the input ports in high impedance state, if the input is floating. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

**(b) A/D converter**

When A/D conversion is not performed, set the VCUT bit in the ADCON1 register to "0" (no VREF connection). When A/D conversion is performed, start the A/D conversion at least 1  $\mu$ s or longer after setting the VCUT bit to "1" (VREF connection).

**(c) 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 stop peripheral function clocks in wait mode), before changing wait mode.

**(d) Switching the oscillation-driving capacity**

Set the driving capacity to "LOW" when oscillation is stable.

### 19.4 Protect

Set the PRC2 bit to “1” (write enabled) and then write to any address, and the PRC2 bit will be cleared to “0” (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to “1”. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to “1” and the next instruction.



## 19.5 Interrupts

### 19.5.1 Reading address 00000<sub>16</sub>

Do not read the address 00000<sub>16</sub> 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 00000<sub>16</sub> during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to "0". If the address 00000<sub>16</sub> is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to "0". This causes a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

### 19.5.2 Setting the SP

Set any value in the SP(USP, ISP) before accepting an interrupt. The SP(USP, ISP) is cleared to '0000<sub>16</sub>' after reset. Therefore, if an interrupt is accepted before setting any value in the SP(USP, ISP), the program may go out of control.

### 19.5.3 The $\overline{\text{NMI}}$ Interrupt

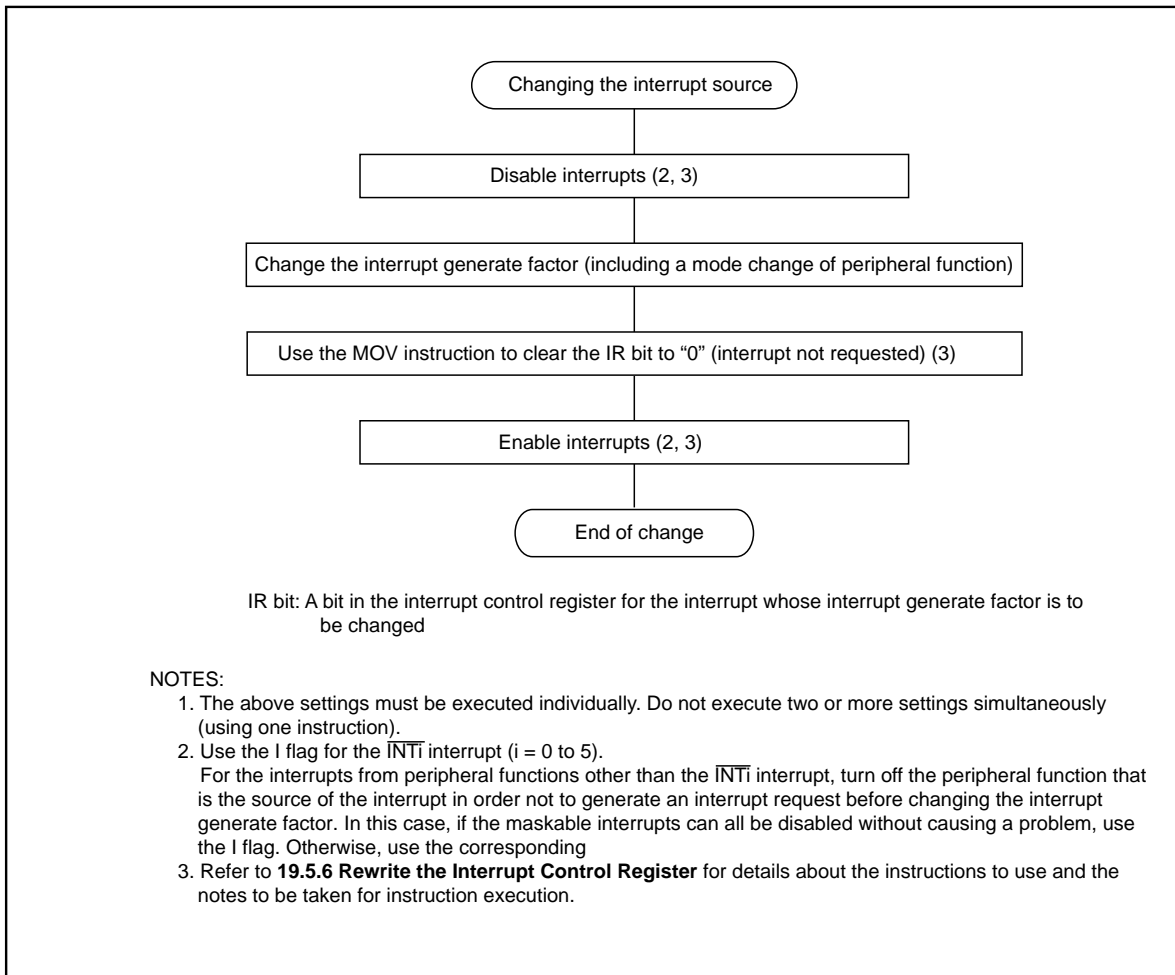
1. The  $\overline{\text{NMI}}$  interrupt is invalid after reset. The  $\overline{\text{NMI}}$  interrupt becomes effective by setting to "1" the PM24 bit in the PM2 register. Set the PM24 bit to "1" when a high-level signal ("H") is applied to the  $\overline{\text{NMI}}$  pin. If the PM24 bit is set to "1" when a low-level signal ("L") is applied,  $\overline{\text{NMI}}$  interrupt is generated. Once  $\overline{\text{NMI}}$  interrupt is enabled, it will not be disabled unless a reset is applied.
2. The input level of the  $\overline{\text{NMI}}$  pin can be read by accessing the P8\_5 bit in the P8 register.
3. When selecting  $\overline{\text{NMI}}$  function, stop mode cannot be entered into while input on the  $\overline{\text{NMI}}$  pin is low. This is because while input on the  $\overline{\text{NMI}}$  pin is low the CM10 bit in the CM1 register is fixed to "0".
4. When selecting  $\overline{\text{NMI}}$  function, do not go to wait mode while input on the  $\overline{\text{NMI}}$  pin is low. This is because when input on the  $\overline{\text{NMI}}$  pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
5. When selecting  $\overline{\text{NMI}}$  function, the low and high level durations of the input signal to the  $\overline{\text{NMI}}$  pin must each be 2 CPU clock cycles + 300 ns or more.
6. When using the  $\overline{\text{NMI}}$  interrupt for exiting stop mode, set the NDDR register to "FF<sub>16</sub>" (disable digital debounce filter) before entering stop mode.

### 19.5.4 Changing the Interrupt Generation Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to "1" (interrupt requested). If you changed the interrupt generate factor for an interrupt that needs to be used, be sure to clear the IR bit for that interrupt to "0" (interrupt not requested).

"Changing the interrupt generate factor" referred to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the generate factor, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to "0" (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.

Figure 19.2 shows the procedure for changing the interrupt generate factor.



**Figure 19.2. Procedure for Changing the Interrupt Generate Factor**

### 19.5.5 $\overline{\text{INT}}$ Interrupt

1. Either an "L" level of at least  $t_{w(\text{INH})}$  or an "H" level of at least  $t_{w(\text{INL})}$  width is necessary for the signal input to pins  $\overline{\text{INT}}_0$  through  $\overline{\text{INT}}_5$  regardless of the CPU operation clock.
2. If the POL bit in the INT0IC to INT5IC registers or the IFSR7 to IFSR0 bits in the IFSR register are changed, the IR bit may inadvertently set to 1 (interrupt requested). Be sure to clear the IR bit to 0 (interrupt not requested) after changing any of those register bits.
3. When using the INT5 interrupt for exiting stop mode, set the P17DDR register to "FF16" (disable digital debounce filter) before entering stop mode.

### 19.5.6 Rewrite the Interrupt Control Register

- (1) The interrupt control register for any interrupt should be modified in places where no requests for that interrupt may occur. Otherwise, disable the interrupt before rewriting the interrupt control register.
- (2) To rewrite the interrupt control register for any interrupt after disabling that interrupt, be careful with the instruction to be used.

#### Changing any bit other than the IR bit

If while executing an instruction, a request for an interrupt controlled by the register being modified occurs, the IR bit in the register may not be set to "1" (interrupt requested), with the result that the interrupt request is ignored. If such a situation presents a problem, use the instructions shown below to modify the register.

Usable instructions: AND, OR, BCLR, BSET

#### Changing the IR bit

Depending on the instruction used, the IR bit may not always be cleared to "0" (interrupt not requested). Therefore, be sure to use the MOV instruction to clear the IR bit.

- (3) When using the I flag to disable an interrupt, refer to the sample program fragments shown below as you set the I flag. (Refer to (2) for details about rewrite the interrupt control registers in the sample program fragments.)

Examples 1 through 3 show how to prevent the I flag from being set to "1" (interrupts enabled) before the interrupt control register is rewritten, due to the internal bus and the instruction queue buffer timing.

#### Example 1: Using the NOP instruction to keep the program waiting until the interrupt control register is modified

```
INT_SWITCH1:
  FCLR   I           ; Disable interrupts
  AND.B  #00h, 0055h ; Set the TA0IC register to 0016
  NOP
  NOP
  FSET   I           ; Enable interrupts
```

The number of NOP instruction is as follows.  
 PM20 = 1 (1 wait) : 2, PM20 = 0 (2 waits): 3

#### Example 2: Using the dummy read to keep the FSET instruction waiting

```
INT_SWITCH2:
  FCLR   I           ; Disable interrupts
  AND.B  #00h, 0055h ; Set the TA0IC register to 0016
  MOV.W  MEM, R0     ; Dummy read
  FSET   I           ; Enable interrupts
```

#### Example 3: Using the POPC instruction to changing the I flag

```
INT_SWITCH3:
  PUSHC  FLG
  FCLR   I           ; Disable interrupts
  AND.B  #00h, 0055h ; Set the TA0IC register to 0016
  POPC   FLG        ; Enable interrupts
```

### 19.5.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

## 19.6 DMAC

### 19.6.1 Write to DMAE Bit in DMiCON Register

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 DMiCON register simultaneously<sup>(\*1)</sup>.

Step 2: Make sure that the DMAi is in an initial state<sup>(\*2)</sup> in a program.

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

#### Notes:

1. The DMAS bit remains unchanged even if “1” is written. However, if “0” is written to this bit, it is set to “0” (DMA not requested). In order to prevent the DMAS bit from being modified to “0”, “1” should be written to the DMAS bit when “1” is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.  
Similarly, when writing to the DMAE bit with a read-modify-write instruction, “1” should be written to the DMAS bit in order to maintain a DMA request which is generated during execution.
2. Read the TCRi register to verify whether the DMAi is in an initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in an initial state. (If a DMA request occurs after writing to the DMAE bit, the value written to the TCRi register is “1”.) If the read value is a value in the middle of transfer, the DMAi is not in an initial state.

## 19.7 Timer

### 19.7.1 Timer A

#### 19.7.1.1 Timer A (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI<sub>MR</sub> (i = 0 to 4) register and the TAI register before setting the TAI<sub>S</sub> bit in the TABSR register to “1” (count starts).  
Always make sure the TAI<sub>MR</sub> register is modified while the TAI<sub>S</sub> bit remains “0” (count stops) regardless whether after reset or not.
2. While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, if the TAI register is read at the same time the counter is reloaded, the read value is always “FFFF<sub>16</sub>”. If the TAI register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.
3. If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on  $\overline{SD}$  pin enabled), the TA1<sub>OUT</sub>, TA2<sub>OUT</sub> and TA4<sub>OUT</sub> pins go to a high-impedance state.

#### 19.7.1.2 Timer A (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI<sub>MR</sub> (i = 0 to 4) register, the TAI register, the UDF register, the TAZIE, TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAI<sub>S</sub> bit in the TABSR register to “1” (count starts).  
Always make sure the TAI<sub>MR</sub> register, the UDF register, the TAZIE, TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAI<sub>S</sub> bit remains “0” (count stops) regardless whether after reset or not.
2. While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, if the TAI register is read at the same time the counter is reloaded, the read value is always “FFFF<sub>16</sub>” when the timer counter underflows and “0000<sub>16</sub>” when the timer counter overflows. If the TAI register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.
3. If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on  $\overline{SD}$  pin enabled), the TA1<sub>OUT</sub>, TA2<sub>OUT</sub> and TA4<sub>OUT</sub> pins go to a high-impedance state.

### 19.7.1.3 Timer A (One-shot Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI<sub>i</sub>MR (i = 0 to 4) register, the TAI<sub>i</sub> register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAI<sub>i</sub>S bit in the TABSR register to “1” (count starts). Always make sure the TAI<sub>i</sub>MR register, the TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAI<sub>i</sub>S bit remains “0” (count stops) regardless whether after reset or not.
2. When setting TAI<sub>i</sub>S bit to “0” (count stop), the following occur:
  - The counter stops counting and the content of reload register is reloaded.
  - TAI<sub>i</sub>OUT pin outputs “L”.
  - After one cycle of the CPU clock, the IR bit in the TAI<sub>i</sub>C register is set to “1” (interrupt request).
3. Output in one-shot timer mode synchronizes with a count source internally generated. When the external trigger has been selected, a maximum delay of one cycle of the count source occurs between the trigger input to TAI<sub>i</sub>N pin and output in one-shot timer mode.
4. The IR bit is set to “1” when timer operation mode is set with any of the following procedures:
  - Select one-shot timer mode after reset.
  - Change the operation mode from timer mode to one-shot timer mode.
  - Change the operation mode from event counter mode to one-shot timer mode.To use the timer A<sub>i</sub> interrupt (the IR bit), set the IR bit to “0” after the changes listed above have been made.
5. When a trigger occurs while the timer is counting, the counter reloads the reload register value, and continues counting after a second trigger is generated and the counter is decremented once. To generate a trigger while counting, space more than one cycle of the timer count source from the first trigger and generate again.
6. When selecting the external trigger for the count start conditions in timer A one-shot timer mode, do generate an external trigger 300ns before the count value of timer A is set to “0000<sub>16</sub>”. The one-shot timer does not continue counting and may stop.
7. If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on  $\overline{SD}$  pin enabled), the TA1<sub>OUT</sub>, TA2<sub>OUT</sub> and TA4<sub>OUT</sub> pins go to a high-impedance state.

#### 19.7.1.4 Timer A (Pulse Width Modulation Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI<sub>i</sub>MR (i = 0 to 4) register, the TAI<sub>i</sub> register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAI<sub>i</sub>S bit in the TABSR register to “1” (count starts).  
Always make sure the TAI<sub>i</sub>MR register, the TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAI<sub>i</sub>S bit remains “0” (count stops) regardless whether after reset or not.
2. The IR bit is set to “1” when setting a timer operation mode with any of the following procedures:
  - Select the PWM mode after reset.
  - Change an operation mode from timer mode to PWM mode.
  - Change an operation mode from event counter mode to PWM mode.To use the timer A<sub>i</sub> interrupt (interrupt request bit), set the IR bit to “0” by program after the above listed changes have been made.
3. When setting TAI<sub>i</sub>S register to “0” (count stop) during PWM pulse output, the following action occurs:
  - Stop counting.
  - When TAI<sub>i</sub>OUT pin is output “H”, output level is set to “L” and the IR bit is set to “1”.
  - When TAI<sub>i</sub>OUT pin is output “L”, both output level and the IR bit remains unchanged.
4. If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register is set to “1” (three-phase output forcible cutoff by input on  $\overline{SD}$  pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

## 19.7.2 Timer B

### 19.7.2.1 Timer B (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR register to "1" (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains "0" (count stops) regardless whether after reset or not.

2. The counter value can be read out at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always "FFFF<sub>16</sub>." If the TBi register is read after setting a value in it but before the counter starts counting, the read value is the one that has been set in the register.

### 19.7.2.2 Timer B (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR register to "1" (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains "0" (count stops) regardless whether after reset or not.

2. The counter value can be read out at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always "FFFF<sub>16</sub>." If the TBi register is read after setting a value in it but before the counter starts counting, the read value is the one that has been set in the register.



### 19.7.2.3 Timer B (Pulse Period/pulse Width Measurement Mode)

1. The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR (i = 0 to 2) register before setting the TBiS bit in the TABSR register to "1" (count starts).  
Always make sure the TBiMR register is modified while the TBiS bit remains "0" (count stops) regardless whether after reset or not. To clear the MR3 bit to "0" by writing to the TBiMR register while the TBiS bit is set to "1" (count starts), be sure to set the TM0D0, TM0D1, MR0, MR1, TCK0 and TCK1 bits to the same value as previously written and the MR2 bit to "0".
2. The IR bit in the TBiIC register (i=0 to 2) goes to "1" (interrupt request), when an effective edge of a measurement pulse is input or timer Bi is overflowed. The factor of interrupt request can be determined by use of the MR3 bit in the TBiMR register within the interrupt routine.
3. If the source of interrupt cannot be identified by the MR3 bit such as when the measurement pulse input and a timer overflow occur at the same time, use another timer to count the number of times timer B has overflowed.
4. To set the MR3 bit to "0" (no overflow), set TBiMR register with setting the TBiS bit to "1" and counting the next count source after setting the MR3 bit to "1" (overflow).
5. Use the IR bit in the TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt factor within the interrupt routine.
6. When the count is started and the first effective edge is input, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.
7. The value of the counter is indeterminate at the beginning of a count. MR3 may be set to "1" and timer Bi interrupt request may be generated between the count start and an effective edge input.
8. For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an "H" level width or an "L" level width.

### 19.7.3 Three-phase Motor Control Timer Function

When the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forced cutoff by SD pin input (high-impedance) enabled), the INV03 bit in the INVC0 register is set to 1 (three-phase motor control timer output enabled), and a low-level ("L") signal is applied to the  $\overline{SD}$  pin while a three-phase PWM signal is output, the MCU is forced to cutoff and pins U,  $\overline{U}$ , V,  $\overline{V}$ , W, and  $\overline{W}$  are placed in a high-impedance state and the INV03 bit is set to 0 (three-phase motor control timer output disabled).

To resume the three-phase PWM signal output from pins U,  $\overline{U}$ , V,  $\overline{V}$ , W, and  $\overline{W}$ , set the INV03 bit to 1 and the IVPCR1 bit to 0 (three-phase output forced cutoff disabled) after the  $\overline{SD}$  pin level becomes "H". Then set the IVPCR1 bit to 1 (three-phase output forced cutoff enabled) in order to enable the three-phase output forced cutoff function by input to the SD pin again.

The INV03 bit cannot be set to 1 while an "L" signal is input to the  $\overline{SD}$  pin. To set the INV03 bit to 1 after forcible cutoff, write 1 to the INV03 bit and read the bit to ensure that it is set to 1 by program. Then set the IVPCR1 bit to 1 after setting it to 0.

## 19.8 Serial I/O

### 19.8.1 Clock-Synchronous Serial I/O

#### 19.8.1.1 Transmission/reception

1. With an external clock selected, and choosing the  $\overline{RTS}$  function, the output level of the  $\overline{RTSi}$  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  $\overline{RTSi}$  pin goes to "H" when reception starts. So if the  $\overline{RTSi}$  pin is connected to the  $\overline{CTS}$  pin on the transmission side, the circuit can transmit and receive data with consistent timing. With the internal clock, the  $\overline{RTS}$  function has no effect.
2. If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register is set to "1" (three-phase output forcible cutoff by input on  $\overline{SD}$  pin enabled), the P73/ $\overline{RTS2}$ /TxDi (when the U1MAP bit in PACR register is "1") and CLK2 pins go to a high-impedance state.

#### 19.8.1.2 Transmission

When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register is set to "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register is set to "1" (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit in the UiC1 register is set to "1" (transmission enabled)
- The TI bit in the UiC1 register is set to "0" (data present in UiTB register)
- If  $\overline{CTS}$  function is selected, input on the  $\overline{CTS}$  pin is "L"

#### 19.8.1.3 Reception

1. In operating the clock-synchronous serial I/O, operating the transmitter generates a clock for the receiver shift register. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TxDi pin when receiving data.
2. When an internal clock is selected, set the TE bit in the UiC1 register ( $i = 0$  to 2) to 1 (transmission enabled) and write dummy data to the UiTB register, and the clock for the receiver shift register 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 clock for the receiver shift register will be generated when the external clock is fed to the CLKi input pin.
3. When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the RE bit in the UiC1 register ( $i = 0$  to 2) is set to "1" (data present in the UiRB register), an overrun error occurs and the OER bit in the UiRB register is set to "1" (overrun error occurred). In this case, because the content of the UiRB register is indeterminate, 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.
4. To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.
5. When an external clock is selected, make sure the external clock is in high state if the CKPOL bit is set to "0", and in low state if the CKPOL bit is set to "1" before the following conditions are met:
  - Set the RE bit in the UiC1 register to "1" (reception enabled)
  - Set the TE bit in the UiC1 register to "1" (transmission enabled)
  - Set the TI bit in the UiC1 register to "0" (data present in the UiTB register)

## 19.8.2 Serial I/O (UART Mode)

### 19.8.1.1 Special Mode 1 (I<sup>2</sup>C bus Mode)

When generating start, stop and restart conditions, set the STSPSEL bit in the U2SMR4 register to "0" and wait for more than half cycle of the transfer clock before setting each condition generate bit (STAREQ, RSTAREQ and STPREQ) from "0" to "1".

### 19.8.1.2 Special Mode 2

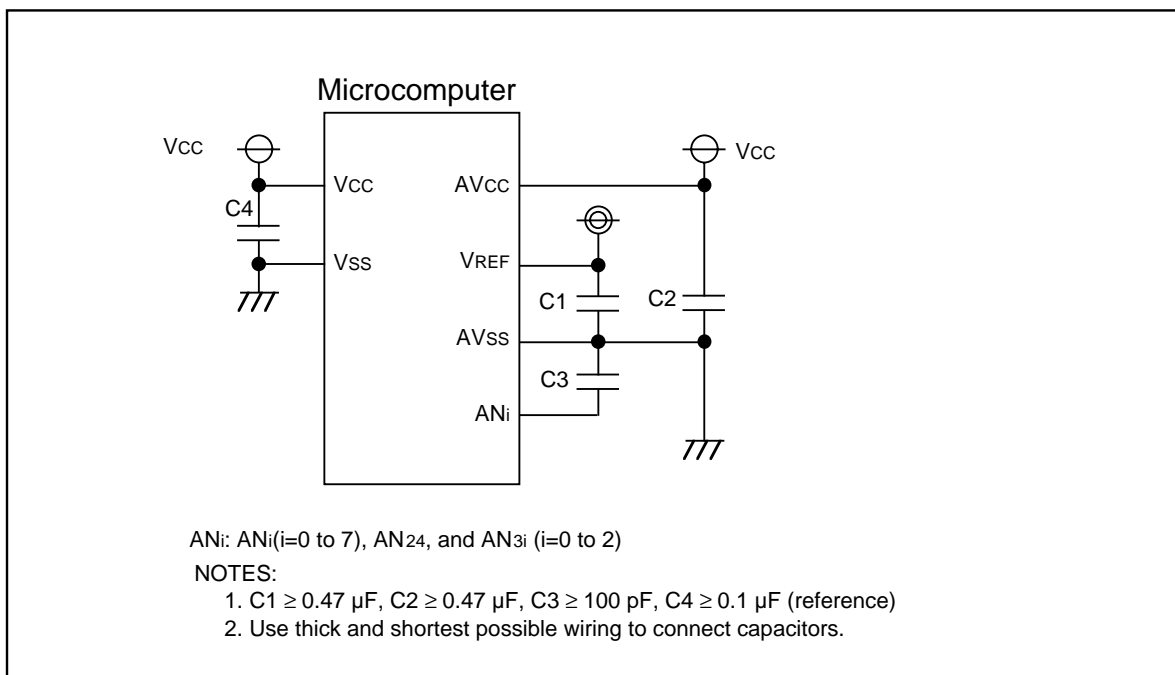
If a low-level signal is applied to the P85/ $\overline{\text{NMI}}/\overline{\text{SD}}$  pin when the IVPCR1 bit in the TB2SC register is set to "1" (three-phase output forcible cutoff by input on  $\overline{\text{SD}}$  pin enabled), the P73/RTS2/TxD1 (when the U1MAP bit in PACR register is "1") and CLK2 pins go to a high-impedance state.

### 19.8.1.3 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to "1" (transmission complete) and U2ERE bit to "1" (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to "0" (no interrupt request) after setting these bits.

## 19.9 A/D Converter

1. Set ADCON0 (except bit 6), ADCON1 and ADCON2 registers when A/D conversion is stopped (before a trigger occurs).
2. When the VCUT bit in the ADCON1 register is changed from "0" (Vref not connected) to "1" (VREF connected), start A/D conversion after waiting 1  $\mu$ s or longer.
3. To prevent noise-induced device malfunction or latchup, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins ( $AN_i$ ( $i=0$  to 7), AN24,  $AN_{3i}$ ( $i=0$  to 2)) each and the AVSS pin. Similarly, insert a capacitor between the VCC pin and the VSS pin. Figure 19.4 is an example connection of each pin.
4. Make sure the port direction bits for those pins that are used as analog inputs are set to "0" (input mode). Also, if the TGR bit in ADCON0 register is set to "1" (external trigger), make sure the port direction bit for the  $\overline{ADTRG}$  pin is set to "0" (input mode).
5. When using key input interrupts, do not use any of the four AN4 to AN7 pins as analog inputs. (A key input interrupt request is generated when the A/D input voltage goes low.)
6. The  $\phi_{AD}$  frequency must be 10 MHz or less (12 MHz or less in M16C/26B). Without sample-and-hold function, limit the  $\phi_{AD}$  frequency to 250kHz or more. With the sample and hold function, limit the  $\phi_{AD}$  frequency to 1MHz or more.
7. When changing an A/D operation mode, select analog input pin again in the CH2 to CH0 bits in the ADCON0 register and the SCAN1 to SCAN0 bits in the ADCON1 register.



**Figure 19.3. Use of capacitors to reduce noise**

8. If the CPU reads the A/D register  $i$  ( $i = 0$  to  $7$ ) at the same time the conversion result is stored in the A/D register  $i$  after completion of A/D conversion, an incorrect value may be stored in the A/D register  $i$ . This problem occurs when a divide-by- $n$  clock derived from the main clock or a subclock is selected for CPU clock.
  - When operating in one-shot mode, single-sweep mode, simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode 1  
Check to see that A/D conversion is completed before reading the target A/D register  $i$ . (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.
  
9. If A/D conversion is forcibly terminated while in progress by setting the ADST bit in the ADCON0 register to "0" (A/D conversion halted), the conversion result of the A/D converter is indeterminate. The contents of A/D register  $i$  irrelevant to A/D conversion may also become indeterminate. If while A/D conversion is underway the ADST bit is cleared to "0" in a program, ignore the values of all A/D register  $i$ .
  
10. When setting the ADST bit in the ADCON register to "0" to terminate a conversion forcefully by the program in single sweep conversion mode, A/D delayed trigger mode 0 and A/D delayed trigger mode 1 during A/D conversion operation, the A/D interrupt request may be generated. If this causes a problem, set the ADST bit to "0" after the interrupt is disabled.

### 19.10 Programmable I/O Ports

1. If a low-level signal is applied to the  $\overline{SD}$  pin when the IVPCR1 bit in the TB2SC register is set to "1" (three-phase output forcible cutoff by input on  $\overline{SD}$  pin enabled), the P72 to P75, P80 and P81 pins go to a high-impedance state.
2. The input threshold voltage of pins differs between programmable input/output ports and peripheral functions.  
Therefore, if any pin is shared by a programmable input/output port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions  $V_{IH}$  and  $V_{IL}$  (neither "high" nor "low"), the input level may be determined differently depending on which side—the programmable input/output port or the peripheral function—is currently selected.
3. When the INV03 bit in the INVC0 register is "1" (three-phase motor control timer output enabled), an "L" input on the P85  $\overline{NMI}/\overline{SD}$  pin, has the following effect:
  - When the IVPCR1 bit in the TB2SC register is set to "1" (three-phase output forcible cutoff by input on the  $\overline{SD}$  pin enabled), the U/  $\overline{U}$ / V/  $\overline{V}$ / W/  $\overline{W}$  pins go to a high-impedance state.
  - When the IVPCR1 bit is set to "0" (three-phase output forcible cutoff by input on  $\overline{SD}$  pin disabled), the U/  $\overline{U}$ / V/  $\overline{V}$ / W/  $\overline{W}$  pins go to a normal port.

Therefore, the P85 pin can not be used as programmable I/O port when the INV03 bit is set to "1".

When the SD function isn't used, set PD85 to "0" (Input) and pull the P85  $\overline{NMI}/\overline{SD}$  pin to "H" externally.

### **19.11 Electric Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers**

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.

## 19.12 Mask ROM Version

### 19.12.1 Internal ROM area

When using the masked ROM version, write nothing to internal ROM area. Writing to the area may increase power consumption.

### 19.12.2 Reserve bit

The b3 to b0 in address 0FFFFFF<sub>16</sub> are reserved bits. Set these bits to "11112".



## 19.13 Flash Memory Version

### 19.13.1 Functions to Inhibit Rewriting Flash Memory

ID codes are stored in addresses 0FFFDF<sub>16</sub>, 0FFFE3<sub>16</sub>, 0FFFE<sub>16</sub>, 0FFFEF<sub>16</sub>, 0FFFF3<sub>16</sub>, 0FFFF7<sub>16</sub>, and 0FFFFB<sub>16</sub>. If wrong data is written to these addresses, the flash memory cannot be read or written in standard serial I/O mode.

The ROMCP register is mapped in address 0FFFFFF<sub>16</sub>. 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 microcomputer, these addresses are allocated to the vector addresses (H) of fixed vectors. The b3 to b0 in address 0FFFFFF<sub>16</sub> are reserved bits. Set these bits to "11112".

### 19.13.2 Stop mode

When the microcomputer enters stop mode, execute the instruction which sets the CM10 bit to "1"(stop mode) after setting the FMR01 bit to "0"(CPU rewrite mode disabled) and disabling the DMA transfer.

### 19.13.3 Wait mode

When the microcomputer enters wait mode, execute the WAIT instruction after setting the FMR01 bit to "0"(CPU rewrite mode disabled).

### 19.13.4 Low power dissipation mode, on-chip oscillator low power dissipation mode

If the CM05 bit is set to "1" (main clock stop), the following commands must not be executed.

- Program
- Block erase

### 19.13.5 Writing command and data

Write the command code and data at even addresses.

### 19.13.6 Program Command

Write 'xx40<sub>16</sub>' in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

### 19.13.7 Operation speed

When CPU clock source is main clock, before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or less for BCLK using the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register. Also, when CPU clock is f<sub>3</sub>(ROC) on-chip oscillator clock, before entering CPU rewrite mode (EW0 or EW1 mode), set the ROCR3 to ROCR2 bits in the ROCR register to "divided by 4" or "divide by 8". On both cases, set the PM17 bit in the PM1 register to "1" (with wait state).

### 19.13.8 Instructions prohibited in EW0 Mode

The following instructions cannot be used in EW0 mode because the flash memory's internal data is referenced: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

### 19.13.9 Interrupts

#### EW0 Mode

- Any interrupt which has a vector in the variable vector table can be used, providing that its vector is transferred into the RAM area.
- The  $\overline{\text{NMI}}$  and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.

Because the rewrite operation is halted when a  $\overline{\text{NMI}}$  or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

- The address match interrupt cannot be used because the flash memory's internal data is referenced.

#### EW1 Mode

- Make sure that any interrupt which has a vector in the relocatable vector table or address match interrupt will not be accepted during the auto program period or auto erase period with erase-suspend function disabled.
- The  $\overline{\text{NMI}}$  interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.

Because the rewrite operation is halted when a  $\overline{\text{NMI}}$  interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

### 19.13.10 How to access

To set the FMR01, FMR02, FMR11 or FMR16 bit to "1", set the subject bit to "1" immediately after setting to "0". Do not generate an interrupt or a DMA transfer between the instruction to set the bit to "0" and the instruction to set the bit to "1". When the PM24 bit is set to "1" ( $\overline{\text{NMI}}$  function), apply a high-level ("H") signal to the  $\overline{\text{NMI}}$  pin to set those bits.

### 19.13.11 Writing in the user ROM area

#### EW0 Mode

- If the power supply voltage drops while rewriting any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, standard serial I/O or parallel I/O mode should be used.

#### EW1 Mode

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

### 19.13.12 DMA transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR00 bit in the FMR0 register is set to "0" (during the auto program or auto erase period).

### 19.13.13 Regarding Programming/Erase Times and Execution Time

As the number of programming/erase times increases, so does the execution time for software commands (Program, and Block Erase).

The software commands are aborted by hardware reset 1, hardware reset 2,  $\overline{\text{NMI}}$  interrupt, and watchdog timer interrupt. If a software command is aborted by such reset or interrupt, the affected block must be erased before reexecuting the aborted command.

### 19.13.14 Definition of Programming/Erase Times

"Number of programs and erasure" refers to the number of erasure per block.

If the number of program and erasure is  $n$  ( $n=100, 1,000, 10,000$ ) each block can be erased  $n$  times.

For example, if a 2K byte block A is erased after writing 1 word data 1024 times, each to a different address, this is counted as one program and erasure. However, data cannot be written to the same address more than once without erasing the block. (Rewrite prohibited)

### 19.13.15 Flash Memory Version Electrical Characteristics 10,000 E/W cycle products (U7, U9)

When Block A or B E/W cycles exceed 100, select one wait state per block access. When FMR17 is set to "1", one wait state is inserted per access to Block A or B - regardless of the value of PM17. Wait state insertion during access to all other blocks, as well as to internal RAM, is controlled by PM17 - regardless of the setting of FMR17.

To use the limited number of erasure efficiently, write to unused address within the block instead of rewrite. Erase block only after all possible address are used. For example, an 8-word program can be written 128 times before erase becomes necessary.

Maintaining an equal number of erasure between Block A and B will also improve efficiency.

We recommend keeping track of the number of times erasure is used.

### 19.13.16 Boot Mode

An indeterminate value is sometimes output in the I/O port until the internal power supply becomes stable when "H" is applied to the CNVss pin and "L" is applied to the  $\overline{\text{RESET}}$  pin.

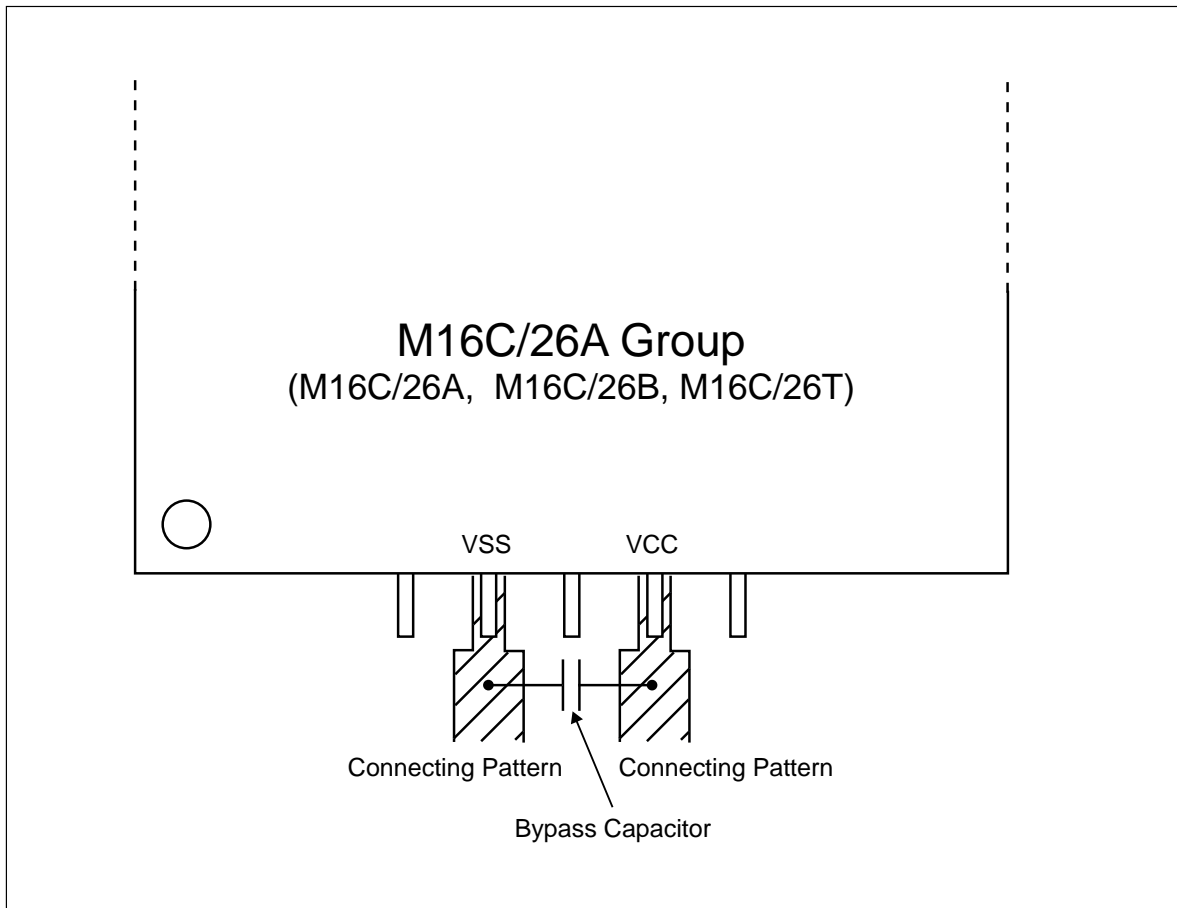
When setting the CNVss pin to "H", the following procedure is required:

- (1) Apply an "L" signal to the  $\overline{\text{RESET}}$  pin and the CNVss pin.
- (2) Bring VCC to more than 2.7V, and wait at least 2msec. (Internal power supply stable waiting time)
- (3) Apply an "H" signal to the CNVss pin.
- (4) Apply an "H" signal to the  $\overline{\text{RESET}}$  pin.

When the CNVss pin is "H" and RESET pin is "L", P67 pin is connected to the pull-up resistor.

### 19.14 Noise

Connect a bypass capacitor (approximately  $0.1\mu\text{F}$ ) across the VCC and VSS pins using the shortest and thicker possible wiring. Figure 19.4 shows the bypass capacitor connection.

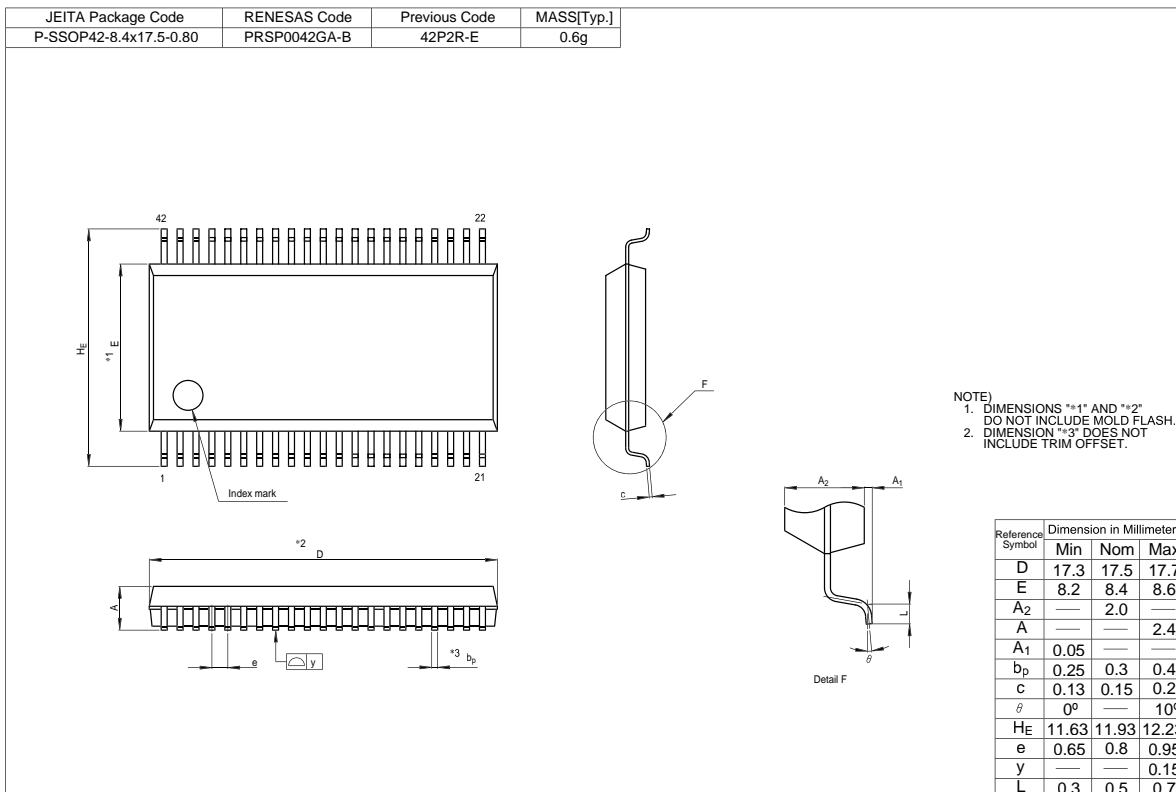
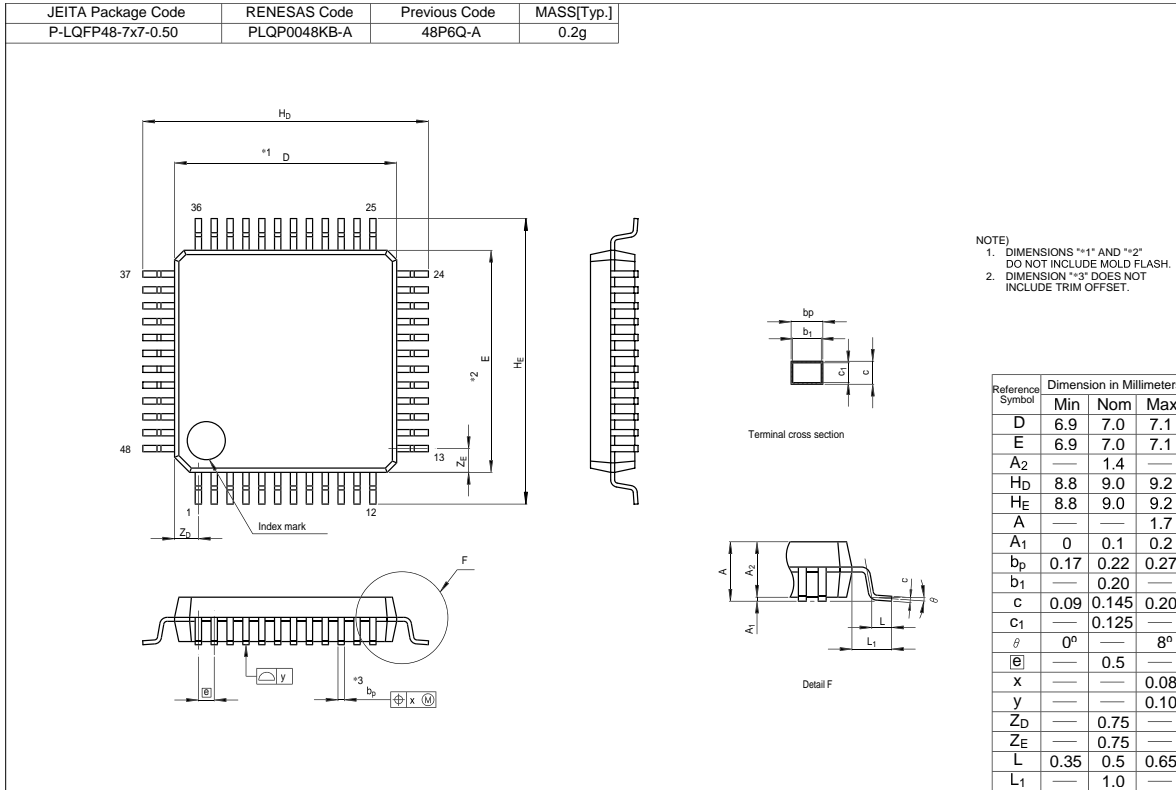


**Figure 19.4 Bypass Capacitor Connection**

**19.15 Instruction for a Device Use**

When handling a device, extra attention is necessary to prevent it from crashing during the electrostatic discharge period.

# Appendix 1. Package Dimensions



## Appendix 2. Functional Difference

### Appendix 2.1 Differences between M16C/26A, M16C/26B, and M16C/26T

Item	M16C/26A, M16C/26B	M16C/26T
Main Clock during and after Reset	Oscillating (Default value "0" while and after the CM05 bit is reset.)	Stoped (Default value "1" while and after the CM05 bit is reset.)
Voltage Detection Circuit (Function of 0019 <sub>16</sub> , 001A <sub>16</sub> , 001F <sub>16</sub> )	Available (VCR1 register, VCR2 register, D4INT register)	Not available (reserved register)
Package	PLQP0048KB-A(48P6Q), PRSP0042GA-B(42P2R)	PLP0048KB-A(48P6Q)

**NOTE:**

1. Since the emulator between the M16C/26A Group and M16C/29 Group are the same, all functions of M16C/29 are built in the emulator. When evaluating M16C/26A Group, do not access to the SFR which is not built in M16C/26A Group. Refer to Hardware Manual about detail and electrical characteristics.

## Appendix 2.2 Differences between M16C/26A Group and M16C/26 Group

Item	M16C/26A Group	M16C/26 Group
Clock Generation Circuit	4 circuits (Main clock oscillation circuit, Sub clock oscillation circuit, on-chip oscillator, PLL frequency synthesizer)	3 circuits (Main clock oscillation circuit, Sub clock oscillation circuit, on-chip oscillator)
System Clock Source After Reset (Initial value of the CM21 bit in the CM2 register)	On-chip oscillator (Initial value "1" of CM21 bit)	Main clock (Initial value "0" of CM21 bit)
On-chip Oscillator Clock	Selectable (8MHz/1MHz/500KHz)	Fixed (1MHz)
PACR2 to PACR0 in the PACR register	Necessary to set after reset 48pin:"1002", 42pin:"0012"	No PACR register
IFSR20 bit in the IFSR2A register	Necessary to set to "1" after reset	No IFSR2A register
External Interrupt	8 causes (INT2 added)	7 causes
13 pin (48-pin version) Function	P84/INT2/ZP	IVCC
P70, P71	N-ch open drain output and CMOS output are selectable by S/W	N-ch open drain output
A/D Input Pin (48-pin version)	12 channels	8 channels
A/D operation Mode	8 modes (single, repeat, single sweep, repeat sweep mode 0, repeat sweep mode 1, simultaneous sampling, delayed trigger mode 0, delayed trigger mode 1) 1 shunt current measurement function is available	5 modes (single, repeat, single sweep, repeat sweep mode 0, repeat sweep mode 1)
Timer B Operation Mode	5 modes (timer, event counter, pulse periods measurement, pulse width measurement, A/D trigger) 1 shunt current measurement function is available	4 modes (timer, event counter, pulse periods measurement, pulse width measurement)
CRC Calculation	Available (compatible to CRC-CCITT and CRC-16 methods)	Not available
Three-phase motor Control	<ul style="list-style-type: none"> <li>Waveform output/Switching port output by software is enabled</li> <li>Position data retention function</li> </ul>	<ul style="list-style-type: none"> <li>Waveform output/Switching port output by software is disabled</li> <li>No position data retention function</li> </ul>
Digital Debounce Function	This function is in the NMI/SD pin and INT5 pin	Not available
3 pin (48-pin version) function	P90/CLKOUT/TB0IN/AN30 (CLKOUT: f1, f8, f32, and fc output)	P90/TB0IN
UART1 Compatible pin	Switching to P64 to P67 or P70 to P73 is enabled	P64 to P67
Flash Memory Protect Function	Protection to blocks 0, 1 by FMR02 bit Protection to the blocks 0 to 3 by FMR16 bit	Protection to blocks 0,1 by FMR02 bit
Package	PLQP0048KB-A(48P6Q), PRSP0042GA-B(42P2R)	PLQP0048KB-A(48P6Q)

### NOTE:

1. Since the emulator between the M16C/26A Group and M16C/29 Group are the same, all functions of M16C/29 are built in the emulator. When evaluating M16C/26A Group, do not access to the SFR which is not built in M16C/26A Group. Refer to Hardware Manual about detail and electrical characteristics.



# Register Index

## A

AD0 to AD7 **184**  
 ADCON0 to ADCON2 **182**  
 ADIC **67**  
 ADSTAT0 **184**  
 ADTRGCON **183**  
 AIER **79**

## B

BCNIC **67**

## C

CM0 **40**  
 CM1 **41**  
 CM2 **42**  
 CPSRF **96, 110**  
 CRCD **214**  
 CRCIN **214**  
 CRCMR **214**  
 CRCSAR **214**

## D

D4INT **30**  
 DAR0 **86**  
 DAR1 **86**  
 DM0CON **85**  
 DM0IC **67**  
 DM0SL **84**  
 DM1CON **85**  
 DM1IC **67**  
 DM1SL **85**  
 DTT **121**

## F

FMR0 **241**  
 FMR1 **241**  
 FMR4 **242**

## I

ICTB2 **121**  
 IDB0 **121**  
 IDB1 **121**  
 IFSR **68, 76**

IFSR2A **68**  
 INT0IC to INT2IC **67**  
 INT3IC **67**  
 INT4IC **67**  
 INT5IC **67**  
 INVC0 **119**  
 INVC1 **120**

## K

KUPIC **67**

## N

NDDR **227**

## O

ONSF **96**

## P

P0 to P13 **224**  
 P17DDR **227**  
 PACR **139, 226**  
 PCLKR **43**  
 PCR **226**  
 PD0 to PD13 **223**  
 PDRF **129**  
 PFCR **131**  
 PLC0 **44**  
 PM0 **35**  
 PM1 **35**  
 PM2 **36, 43**  
 PRCR **60**  
 PUR0 to PUR2 **225**

## R

RMAD0 **79**  
 RMAD1 **79**  
 ROCR **41**  
 ROMCP **236**

## S

S0RIC to S2RIC **67**  
 S0TIC to S2TIC **67**  
 SAR0 **86**  
 SAR1 **86**

**T**

TA0 to TA4 **95**  
TA0IC to TA4IC **67**  
TA0MR to TA4MR **94**  
TA1 **122**  
TA11 **122**  
TA1MR **125**  
TA2 **122**  
TA21 **122**  
TA2MR **125**  
TA2MR to TA4MR **101**  
TA4 **122**  
TA41 **122**  
TA4MR **125**  
TABSR **95, 110, 124**  
TAiMR **99, 106**  
TB0 to TB5 **110**  
TB0IC TO TB2IC **67**  
TB0MR to TB5MR **109**  
TB2 **124**  
TB2MR **125**  
TB2SC **123, 185**  
TCR0 **86**  
TCR1 **86**  
TPRC **131**  
TRGSR **96, 124**

**U**

U0BRG to U2BRG **136**  
U0C0 to U2C0 **138**  
U0C1 to U2C1 **139**  
U0MR to U2MR **137**  
U0RB to U2RB **136**  
U0TB to U2TB **136**  
U2SMR **140**  
U2SMR2 **140**  
U2SMR3 **141**  
U2SMR4 **141**  
UCON **138**  
UDF **95**

**V**

VCR1 **30**  
VCR2 **30**

**W**

WDC **81**  
WDTS **81**

<b>REVISION HISTORY</b>	<b>M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) Hardware Manual</b>
-------------------------	--

Rev.	Date	Description	
		Page	Summary
2.00	Feb.15,07	-	M16C/26B newly added, word standardized: on-chip oscillator, development tool
		1	<b>Overview</b> •Description partially deleted
		2 - 3	• <b>1.2 Performance Outline</b> modified
		4 - 5	• <b>Figure 1.1 and 1.2 Block Diagrams</b> updated
		6	• <b>1.4 Product List</b> updated
		7	• <b>Figure 1.3 Product Numbering System</b> updated
		8	• <b>Tables 1.7 to 1.10 Product Codes</b> updated
		12, 14	• <b>Tables 1.11 and 1.12 Pin Characteristics</b> newly added
		15 - 16	• <b>Tables 1.13 Pin Description</b> newly added
		20	<b>SFRs</b> • <b>Table 4.1 SFR Information(1)</b> Note about WDC register is deleted
22	• <b>Table 4.3 SFR Information(3)</b> Value after reset for ROCR register modified		
23	• <b>Table 4.4 SFR Information(4)</b> Note 2 added to IFSR2A register		
28	<b>Reset</b> • <b>Figure 5.1.1.2. Reset Sequence</b> Vcc line and ROC line are modified		
29	• <b>Figure 5.5.1. Voltage Detection Circuit Block</b> WDC register's block is deleted		
35	<b>Processor Mode</b> • <b>Figure 6.1 PM1 Register</b> Note 2 partially added		
36	• <b>Figure 6.2 PM2 Register</b> newly added		
37	• <b>Figure 6.3 Bus Block Diagram</b> and <b>Table 6.1 Accessible Area and Bus Cycle</b> newly added		
41	<b>Clock Generation Circuit</b> • <b>Figure 7.4 ROCR Register</b> modified		
43	• <b>Figure 7.6 PM2 Register</b> Notes 2, 5, 6 modified		
45 - 46	• <b>Figure 7.1.1 and 7.2.1 Examples of Main Clock Connection Circuit</b> updated		
47	• <b>7.4 PLL Clock</b> Description modified for M16C/26B		
	• <b>Table 7.4.1 Example for Setting PLL Clock Frequencies</b> Note 1 modified		
50	• <b>7.6.1 Normal Operation Mode</b> Description modified		
51	• <b>Table 7.6.1.1 Setting Clock Related Bit and Modes</b> modified		
54	• <b>Figure 7.6.1 State Transition to Stop Mode and Wait Mode</b> modified		
55	• <b>Figure 7.6.1.1. State Transition in Normal Mode</b> modified		
56	• <b>Table 7.6.1 Allowed Transition and Setting</b> modified, Notes 1 and 2 modified		
59	• <b>Figure 7.8.3.1 Procedure to Switch Clock Source From On-chip Oscillator to Main Clock</b> updated		
60	<b>Protection</b> •Description partially modified		
76	<b>Interrupt</b> • <b>9.6 INT Interrupt</b> Description partially added		

<b>REVISION HISTORY</b>	<b>M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) Hardware Manual</b>
-------------------------	--

Rev.	Date	Description	
		Page	Summary
		77	• <b>9.7 NMI Interrupt</b> Description partially added
		78	• <b>Table 9.9.1 Value of the PC that is saved to the stack area when an address match interrupt request is accepted</b> modified, note 1 added
		-	<b>Watchdog Timer</b>
		80	•Section of Cold Start/Warm Start deleted
		80	•Description partially added
		81	• <b>Figure 10.1 Watchdog Timer Block Diagram</b> partially modified
		81	• <b>Figure 10.2 WDC Register and WDTS Register</b> notes deleted, WDC5 bit deleted
			• <b>10.1 Count source protective mode</b> description partially added
		108	<b>Timer</b>
		108	•Description about A/D trigger mode modified
		115	• <b>Figure 12.2.1 Timber B Block Diagram</b> A/D trigger mode added
		115	• <b>12.2.4 A/D Trigger Mode</b> Description modified
		121	• <b>Figure 12.3.4 IDB0 Register, IDB1 Register, DTT Register, and ICTB2 Register</b> modified
		123	• <b>Figure 12.3.6 TB2SC Register</b> modified, note 4 added
		131	• <b>Figure 12.3.2.2 TPRC Register</b> bit map modified
		133	<b>Serial I/O</b>
		133	• <b>Figure 13.1.1 Block Diagram of UARTi (i = 0 to 2)</b> PLL clock added
		136	• <b>Figure 13.1.4 U0TB to U2TB Registers, U0RB to U2RB Registers, U0BRG to U2BRG Registers</b> modified, note 3 for UiBRG added
		138	• <b>Figure 13.1.6 U0C0 to U2C0 Registers</b> Note 2 modified, note 7 added
		139	• <b>Figure 13.1.7 PACR Register</b> note 1 modified
		142	• <b>Table 13.1.1.1 Clock Synchronous Serial I/O Mode Specification</b> note 2 modified
		145	• <b>Figure 13.1.1.1 Typical Transmit/Receive Timings in Clock Synchronous Serial I/O Mode</b> partially modified
		150	• <b>Table 13.1.2.1 UART Mode Specifications</b> Note 1 modified
		154	• <b>Figure 13.1.2.2 Receive Operation</b> Figure modified
		158	• <b>Table 13.1.3.1 I<sup>2</sup>C bus Mode Specifications</b> note 2 modified
		168	• <b>Table 13.1.4.1 Special Mode 2 Specifications</b> note 2 modified
		175	• <b>Table 13.1.6.1 SIM Mode Specifications</b> note 1 modified
		177	• <b>Figure 13.1.6.1 Transmit and Receive Timing in SIM Mode</b> timing modified
			<b>A/D Converter</b>
		180	• <b>Table 14.1 A/D Converter Performance</b> note 2 partially added
		183	• <b>Table 14.2 A/D Conversion Frequency Select</b> note 1 partially added
		205	• <b>Table 14.1.8.1 Delayed Trigger Mode 1 Specifications</b> note 1 modified
		212	• <b>Figure 14.5.1 Analog Input Pin and External Sensor Equivalent Circuit</b> note

<b>REVISION HISTORY</b>	<b>M16C/26A Group (M16C/26A, M16C/26B, M16C/26T) Hardware Manual</b>
-------------------------	--

Rev.	Date	Description	
		Page	Summary
			1 added
			<b>CRC Calculation Circuit</b>
		213	• <b>15.1 CRC Snoop</b> Description partially modified
		214	• <b>Figure 15.2 CRCSAR Register</b> note 1 added
			<b>Programmable I/O Ports</b>
		216	• <b>16.3 Pull-up Control Register 0 to Pull-up Control Register 2</b> description modified
		217	• <b>16.6 Digital Debounce function</b> equation modified
		218 - 221	• <b>Figure 16.1 I/O Ports (1) to 16.4 I/O Ports (4)</b> modified
		227	• <b>Figure 16.6.1 NDDR and P17DDR Registers</b> equation modified, note 2 modified
			<b>Flash Memory Version</b>
		231	• <b>17.1.1 Boot Mode</b> newly added
		232	• <b>17.2 Memory Map</b> partially deleted
		235	• <b>17.3.1 ROM Code Protect Function</b> description modified
		236	• <b>Figure 17.3.1.1 ROMCP Address</b> modified
		237	• <b>Table 17.4.1 EW0 Mode and EW1 Mode</b> note 2 mark deleted
		239	• <b>17.5.1 Flash Memory Control Register 0</b> Description about low power consumption mode or on-chip oscillator low-power consumption mode is entered partially modified
		240	• <b>17.5.2 Flash Memory Control Register 1</b> Description about FMR17 bit modified
		241	• <b>Figure 17.5.1 FMR0 Register</b> note 3 modified, <b>FMR1 Register:</b> bit map modified
		242	• <b>Figure 17.5.2 FMR4 Register</b> note 2 modified
		243	• <b>Figure 17.5.1.2 Setting and Resetting of EW1 Mode</b> note for single-chip mode deleted, note 3 for FMR11 bit added
			<b>Electrical Characteristics</b>
		261	• <b>Table 18.1 Absolute Maximum Ratings</b> Rated value modified, note 1 added
		262	• <b>Table 18.2 Recommended Operating Conditions</b> value partially added, figures in note 4 partially added
		263	• <b>Table 18.3 A/D Conversion Characteristics</b> note 4 modified
		264	• <b>Table 18.4 and Table 18.5 Flash Memory Version Electrical Characteristic</b> note 4 partially added, note 6 and note 7 modified
		265	• <b>Table 18.6 Voltage Detection Circuit Electrical Characteristics</b> conditions modified
			• <b>Figure for td(P-R) and td(ROC)</b> modified
		267	• <b>Table 18.9 eElectrical Characteristics (2)</b> flash memory's value for M16C/26B added, note 5 deleted
		274	• <b>Table 18.24 Electrical Characteristics (2)</b> note 5 deleted



---

**RENESAS 16-BIT CMOS SINGLE-CHIP MICROCOMPUTER  
HARDWARE MANUAL  
M16C/26A Group (M16C/26A, M16C/26B, M16C/26T)**

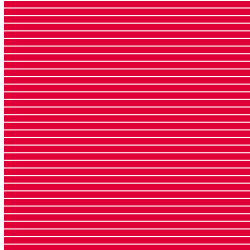
**Publication Date: Rev.1.00 Mar.15, 2005  
Rev.2.00 Feb.15, 2007**

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

---

© 2007. Renesas Technology Corp., All rights reserved. Printed in Japan.

M16C/26A Group (M16C/26A, M16C/26B, M16C/26T)  
Hardware Manual



Renesas Technology Corp.

2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan