

# 16

## M16C/6N Group (M16C/6N4) Hardware Manual

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER  
M16C FAMILY / M16C/60 SERIES

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

### Keep safety first in your circuit designs!

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

### Notes regarding these materials

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

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

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

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

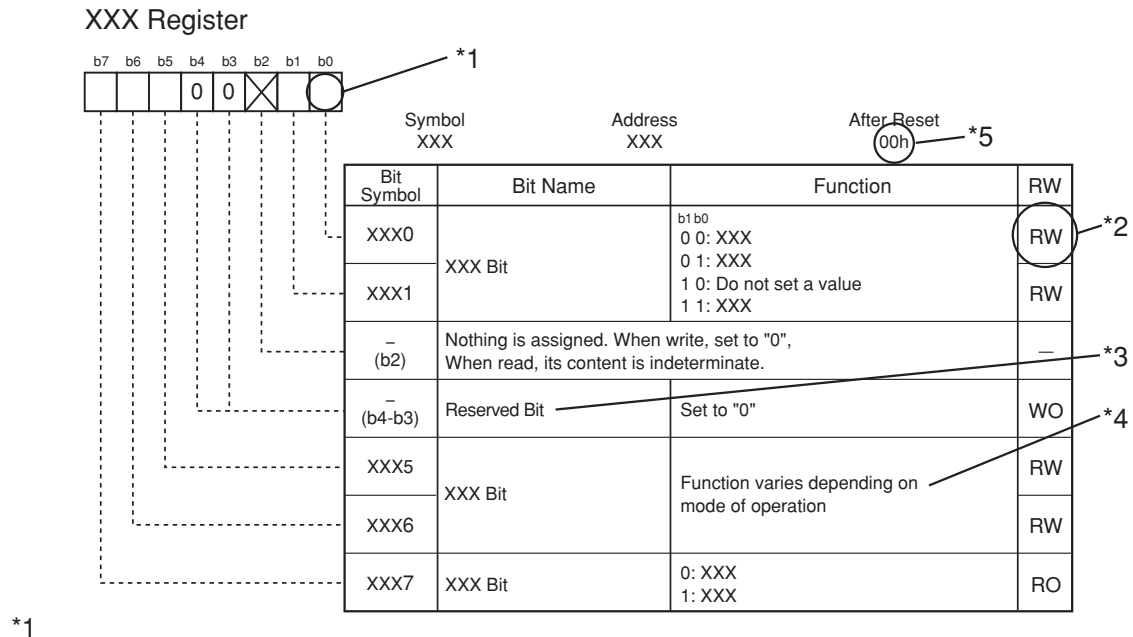
# How to Use This Manual

## 1. Introduction

This hardware manual provides detailed information on the M16C/6N Group (M16C/6N4) of microcomputers. Users are expected to have basic knowledge of electric circuits, logical circuits and microcomputers.

## 2. Register Diagram

The symbols, and descriptions, used for bit function in each register are shown below.



Blank: Set to "0" or "1" according to the application

0 : Set to "0"

1 : Set to "1"

X: Nothing is assigned

RW: Read and write

RO: Read only

WO: Write only

—: Nothing is assigned

• Reserved bit

Reserved bit. Set to specified value.

• Nothing is assigned

Nothing is assigned to the bit concerned. As the bit may be use for future functions, set to "0" when writing to this bit.

• Do not set to this value

The operation is not guaranteed when a value is set.

• Function varies depending on mode of operation

Bit function varies depending on peripheral function mode.

Refer to respective register for each mode.

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

### 3. M16C Family Documents

The following documents were prepared for the M16C family <sup>(1)</sup>.

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

**NOTE:**

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

# Table of Contents

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

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

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

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



20. Programmable I/O Ports .....	238
20.1 PDi Register .....	238
20.2 Pi Register .....	238
20.3 PURj Register .....	238
20.4 PCR Register .....	239
21. Flash Memory Version .....	251
21.1 Memory Map .....	252
21.1.1 Boot Mode .....	252
21.2 Functions to Prevent Flash Memory from Rewriting .....	253
21.2.1 ROM Code Protect Function .....	253
21.2.2 ID Code Check Function .....	253
21.3 CPU Rewrite Mode .....	255
21.3.1 EW0 Mode .....	256
21.3.2 EW1 Mode .....	256
21.3.3 FMR0, FMR1 Registers .....	257
21.3.4 Precautions on CPU Rewrite Mode .....	262
21.3.5 Software Commands .....	264
21.3.6 Data Protect Function .....	269
21.3.7 Status Register (SRD Register) .....	269
21.3.8 Full Status Check .....	271
21.4 Standard Serial I/O Mode .....	273
21.4.1 ID Code Check Function .....	273
21.4.2 Example of Circuit Application in Standard Serial I/O Mode .....	277
21.5 Parallel I/O Mode .....	278
21.5.1 User ROM and Boot ROM Areas .....	278
21.5.2 ROM Code Protect Function .....	278
21.6 CAN I/O Mode .....	279
21.6.1 ID Code Check Function .....	279
21.6.2 Example of Circuit Application in CAN I/O Mode .....	282
21.7 Electrical Characteristics .....	283
21.7.1 Electrical Characteristics (T/V-ver.) .....	283
21.7.2 Electrical Characteristics (Normal-ver.) .....	284
22. Electrical Characteristics .....	285
22.1 Electrical Characteristics (T/V-ver.) .....	285
22.2 Electrical Characteristics (Normal-ver.) .....	306
23. Usage Precaution .....	342
23.1 External Bus .....	342
23.2 PLL Frequency Synthesizer .....	343
23.3 Power Control .....	344
23.4 Protection .....	346
23.5 Interrupt .....	347
23.5.1 Reading Address 00000h .....	347
23.5.2 Setting SP .....	347
23.5.3 $\overline{\text{NMI}}$ Interrupt .....	347
23.5.4 Changing Interrupt Generate Factor .....	348
23.5.5 $\overline{\text{INT}}$ Interrupt .....	348
23.5.6 Rewrite Interrupt Control Register .....	349
23.5.7 Watchdog Timer Interrupt .....	349

23.6 DMAC .....	350
23.6.1 Write to DMAE Bit in DMiCON Register .....	350
23.7 Timers .....	351
23.7.1 Timer A .....	351
23.7.2 Timer B .....	354
23.8 Serial Interface .....	356
23.8.1 Clock Synchronous Serial I/O Mode .....	356
23.8.2 Special Modes .....	357
23.8.3 SI/O3 .....	358
23.9 A/D Converter .....	359
23.10 CAN Module .....	361
23.10.1 Reading CiSTR Register .....	361
23.10.2 Performing CAN Configuration .....	363
23.10.3 Suggestions to Reduce Power Consumption .....	364
23.10.4 CAN Transceiver in Boot Mode .....	365
23.11 Programmable I/O Ports .....	366
23.12 Electrical Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers ....	367
23.13 Mask ROM Version .....	368
23.14 Flash Memory Version .....	369
23.14.1 Functions to Prevent Flash Memory from Rewriting .....	369
23.14.2 Stop Mode .....	369
23.14.3 Wait Mode .....	369
23.14.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode .....	369
23.14.5 Writing Command and Data .....	369
23.14.6 Program Command .....	369
23.14.7 Lock Bit Program Command .....	369
23.14.8 Operation Speed .....	370
23.14.9 Prohibited Instructions .....	370
23.14.10 Interrupt .....	370
23.14.11 How to Access .....	370
23.14.12 Rewriting in User ROM Area .....	370
23.14.13 DMA Transfer .....	370
23.15 Flash Memory Programming Using Boot Program .....	371
23.15.1 Programming Using Serial I/O Mode .....	371
23.15.2 Programming Using CAN I/O Mode .....	371
23.16 Noise .....	372
Appendix 1. Package Dimensions .....	373
Register Index .....	375

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

## SFR Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	35
0005h	Processor Mode Register 1	PM1	36
0006h	System Clock Control Register 0	CM0	53
0007h	System Clock Control Register 1	CM1	54
0008h	Chip Select Control Register	CSR	41
0009h	Address Match Interrupt Enable Register	AIER	92
000Ah	Protect Register	PRCR	74
000Bh			
000Ch	Oscillation Stop Detection Register	CM2	55
000Dh			
000Eh	Watchdog Timer Start Register	WDTS	94
000Fh	Watchdog Timer Control Register	WDC	94
0010h			
0011h	Address Match Interrupt Register 0	RMAD0	92
0012h			
0013h			
0014h			
0015h	Address Match Interrupt Register 1	RMAD1	92
0016h			
0017h			
0018h			
0019h			
001Ah			
001Bh	Chip Select Expansion Control Register	CSE	47
001Ch	PLL Control Register 0	PLC0	57
001Dh			
001Eh	Processor Mode Register 2	PM2	57
001Fh			
0020h			
0021h	DMA0 Source Pointer	SAR0	99
0022h			
0023h			
0024h			
0025h	DMA0 Destination Pointer	DAR0	99
0026h			
0027h			
0028h	DMA0 Transfer Counter	TCR0	99
0029h			
002Ah			
002Bh			
002Ch	DMA0 Control Register	DM0CON	98
002Dh			
002Eh			
002Fh			
0030h			
0031h	DMA1 Source Pointer	SAR1	99
0032h			
0033h			
0034h			
0035h	DMA1 Destination Pointer	DAR1	99
0036h			
0037h			
0038h	DMA1 Transfer Counter	TCR1	99
0039h			
003Ah			
003Bh			
003Ch	DMA1 Control Register	DM1CON	98
003Dh			
003Eh			
003Fh			

The blank areas are reserved.

Address	Register	Symbol	Page
0040h			
0041h	CAN0/1 Wake-up Interrupt Control Register	C01WKIC	80
0042h	CAN0 Successful Reception Interrupt Control Register	C0RECIC	80
0043h	CAN0 Successful Transmission Interrupt Control Register	C0TRMIC	80
0044h	INT3 Interrupt Control Register	INT3IC	81
0045h	Timer B5 Interrupt Control Register	TB5IC	80
0046h	Timer B4 Interrupt Control Register	TB4IC	80
	UART1 Bus Collision Detection Interrupt Control Register	U1BCNIC	80
0047h	Timer B3 Interrupt Control Register	TB3IC	80
	UART0 Bus Collision Detection Interrupt Control Register	U0BCNIC	80
0048h	CAN1 Successful Reception Interrupt Control Register	C1RECIC	81
	INT5 Interrupt Control Register	INT5IC	81
0049h	CAN1 Successful Transmission Interrupt Control Register	C1TRMIC	81
	SI/O3 Interrupt Control Register	S3IC	81
	INT4 Interrupt Control Register	INT4IC	81
004Ah	UART2 Bus Collision Detection Interrupt Control Register	U2BCNIC	80
004Bh	DMA0 Interrupt Control Register	DM0IC	80
004Ch	DMA1 Interrupt Control Register	DM1IC	80
004Dh	CAN0/1 Error Interrupt Control Register	C01ERRIC	80
004Eh	A/D Conversion Interrupt Control Register	ADIC	80
	Key Input Interrupt Control Register	KUPIC	80
004Fh	UART2 Transmit Interrupt Control Register	S2TIC	80
0050h	UART2 Receive Interrupt Control Register	S2RIC	80
0051h	UART0 Transmit Interrupt Control Register	S0TIC	80
0052h	UART0 Receive Interrupt Control Register	S0RIC	80
0053h	UART1 Transmit Interrupt Control Register	S1TIC	80
0054h	UART1 Receive Interrupt Control Register	S1RIC	80
0055h	Timer A0 Interrupt Control Register	TA0IC	80
0056h	Timer A1 Interrupt Control Register	TA1IC	80
0057h	Timer A2 Interrupt Control Register	TA2IC	80
0058h	Timer A3 Interrupt Control Register	TA3IC	80
0059h	Timer A4 Interrupt Control Register	TA4IC	80
005Ah	Timer B0 Interrupt Control Register	TB0IC	80
005Bh	Timer B1 Interrupt Control Register	TB1IC	80
005Ch	Timer B2 Interrupt Control Register	TB2IC	80
005Dh	INT0 Interrupt Control Register	INT0IC	81
005Eh	INT1 Interrupt Control Register	INT1IC	81
005Fh	INT2 Interrupt Control Register	INT2IC	81
0060h	CAN0 Message Box 0: Identifier / DLC		216 217
0061h			
0062h			
0063h			
0064h			
0065h	CAN0 Message Box 0: Data Field		
0066h			
0067h			
0068h			
0069h			
006Ah	CAN0 Message Box 0: Time Stamp		
006Bh			
006Ch			
006Dh			
006Eh			
006Fh	CAN0 Message Box 1: Identifier / DLC		
0070h			
0071h			
0072h			
0073h			
0074h	CAN0 Message Box 1: Data Field		
0075h			
0076h			
0077h			
0078h			
0079h	CAN0 Message Box 1: Time Stamp		
007Ah			
007Bh			
007Ch			
007Dh			
007Eh	CAN0 Message Box 1: Time Stamp		
007Fh			

Address	Register	Symbol	Page
0080h	CAN0 Message Box 2: Identifier / DLC		216 217
0081h			
0082h			
0083h			
0084h			
0085h	CAN0 Message Box 2: Data Field		
0086h			
0087h			
0088h			
0089h			
008Ah			
008Bh	CAN0 Message Box 2: Time Stamp		
008Ch			
008Dh			
008Eh	CAN0 Message Box 3: Identifier / DLC		
008Fh			
0090h			
0091h			
0092h			
0093h	CAN0 Message Box 3: Data Field		
0094h			
0095h			
0096h			
0097h			
0098h			
0099h	CAN0 Message Box 3: Time Stamp		
009Ah			
009Bh			
009Ch	CAN0 Message Box 4: Identifier / DLC		
009Dh			
009Eh			
009Fh			
00A0h			
00A1h	CAN0 Message Box 4: Data Field		
00A2h			
00A3h			
00A4h			
00A5h			
00A6h			
00A7h	CAN0 Message Box 4: Time Stamp		
00A8h			
00A9h			
00AAh	CAN0 Message Box 5: Identifier / DLC		
00ABh			
00ACh			
00ADh			
00AEh			
00AFh	CAN0 Message Box 5: Data Field		
00B0h			
00B1h			
00B2h			
00B3h			
00B4h			
00B5h	CAN0 Message Box 5: Time Stamp		
00B6h			
00B7h			
00B8h	CAN0 Message Box 5: Time Stamp		
00B9h			
00BAh			
00BBh			
00BCh			
00BDh	CAN0 Message Box 5: Time Stamp		
00BEh			
00BFh			

Address	Register	Symbol	Page
00C0h	CAN0 Message Box 6: Identifier / DLC		216 217
00C1h			
00C2h			
00C3h			
00C4h			
00C5h			
00C6h	CAN0 Message Box 6: Data Field		
00C7h			
00C8h			
00C9h			
00CAh			
00CBh			
00CCh			
00CDh			
00CEh	CAN0 Message Box 6: Time Stamp		
00CFh			
00D0h	CAN0 Message Box 7: Identifier / DLC		
00D1h			
00D2h			
00D3h			
00D4h			
00D5h			
00D6h	CAN0 Message Box 7: Data Field		
00D7h			
00D8h			
00D9h			
00DAh			
00DBh			
00DCh			
00DDh			
00DEh	CAN0 Message Box 7: Time Stamp		
00DFh			
00E0h	CAN0 Message Box 8: Identifier / DLC		
00E1h			
00E2h			
00E3h			
00E4h			
00E5h			
00E6h	CAN0 Message Box 8: Data Field		
00E7h			
00E8h			
00E9h			
00EAh			
00EBh			
00ECh			
00EDh			
00EEh	CAN0 Message Box 8: Time Stamp		
00EFh			
00F0h	CAN0 Message Box 9: Identifier / DLC		
00F1h			
00F2h			
00F3h			
00F4h			
00F5h			
00F6h	CAN0 Message Box 9: Data Field		
00F7h			
00F8h			
00F9h			
00FAh			
00FBh			
00FCh			
00FDh			
00FEh	CAN0 Message Box 9: Time Stamp		
00FFh			

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

The blank areas are reserved.

Address	Register	Symbol	Page		
0140h	CAN0 Message Box 14: Identifier /DLC		216 217		
0141h					
0142h					
0143h					
0144h					
0145h					
0146h	CAN0 Message Box 14: Data Field				
0147h					
0148h					
0149h					
014Ah					
014Bh					
014Ch	CAN0 Message Box 14: Time Stamp				
014Dh					
014Eh					
014Fh					
0150h				CAN0 Message Box 15: Identifier /DLC	
0151h					
0152h					
0153h					
0154h					
0155h					
0156h	CAN0 Message Box 15: Data Field				
0157h					
0158h					
0159h					
015Ah					
015Bh					
015Ch	CAN0 Message Box 15: Time Stamp				
015Dh					
015Eh					
015Fh					
0160h			CAN0 Global Mask Register	C0GMR	218
0161h					
0162h					
0163h					
0164h					
0165h					
0166h	CAN0 Local Mask A Register	C0LMAR	218		
0167h					
0168h					
0169h					
016Ah					
016Bh					
016Ch	CAN0 Local Mask B Register	C0LMBR	218		
016Dh					
016Eh					
016Fh					
0170h					
0171h					
0172h					
0173h					
0174h					
0175h					
0176h					
0177h					
0178h					
0179h					
017Ah					
017Bh					
017Ch					
017Dh					
017Eh					
017Fh					

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

The blank areas are reserved.

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

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

The blank areas are reserved.

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



Address	Register	Symbol	Page
0280h	CAN1 Message Box 2: Identifier / DLC		216 217
0281h			
0282h			
0283h			
0284h			
0285h			
0286h	CAN1 Message Box 2: Data Field		
0287h			
0288h			
0289h			
028Ah			
028Bh			
028Ch			
028Dh			
028Eh	CAN1 Message Box 2: Time Stamp		
028Fh			
0290h	CAN1 Message Box 3: Identifier / DLC		
0291h			
0292h			
0293h			
0294h			
0295h			
0296h	CAN1 Message Box 3: Data Field		
0297h			
0298h			
0299h			
029Ah			
029Bh			
029Ch			
029Dh			
029Eh	CAN1 Message Box 3: Time Stamp		
029Fh			
02A0h	CAN1 Message Box 4: Identifier / DLC		
02A1h			
02A2h			
02A3h			
02A4h			
02A5h			
02A6h	CAN1 Message Box 4: Data Field		
02A7h			
02A8h			
02A9h			
02AAh			
02ABh			
02ACh			
02ADh			
02AEh	CAN1 Message Box 4: Time Stamp		
02AFh			
02B0h	CAN1 Message Box 5: Identifier / DLC		
02B1h			
02B2h			
02B3h			
02B4h			
02B5h			
02B6h	CAN1 Message Box 5: Data Field		
02B7h			
02B8h			
02B9h			
02BAh			
02BBh			
02BCh			
02BDh			
02BEh	CAN1 Message Box 5: Time Stamp		
02BFh			

Address	Register	Symbol	Page
02C0h	CAN1 Message Box 6: Identifier / DLC		216 217
02C1h			
02C2h			
02C3h			
02C4h			
02C5h			
02C6h	CAN1 Message Box 6: Data Field		
02C7h			
02C8h			
02C9h			
02CAh			
02CBh			
02CCh			
02CDh			
02CEh	CAN1 Message Box 6: Time Stamp		
02CFh			
02D0h	CAN1 Message Box 7: Identifier / DLC		
02D1h			
02D2h			
02D3h			
02D4h			
02D5h			
02D6h	CAN1 Message Box 7: Data Field		
02D7h			
02D8h			
02D9h			
02DAh			
02DBh			
02DCh			
02DDh			
02DEh			
02DFh	CAN1 Message Box 7: Time Stamp		
02E0h			
02E1h	CAN1 Message Box 8: Identifier / DLC		
02E2h			
02E3h			
02E4h			
02E5h			
02E6h	CAN1 Message Box 8: Data Field		
02E7h			
02E8h			
02E9h			
02EAh			
02EBh			
02ECh			
02EDh			
02EEh	CAN1 Message Box 8: Time Stamp		
02EFh			
02F0h	CAN1 Message Box 9: Identifier / DLC		
02F1h			
02F2h			
02F3h			
02F4h			
02F5h			
02F6h	CAN1 Message Box 9: Data Field		
02F7h			
02F8h			
02F9h			
02FAh			
02FBh			
02FCh			
02FDh			
02FEh	CAN1 Message Box 9: Time Stamp		
02FFh			



Address	Register	Symbol	Page
0300h	CAN1 Message Box 10: Identifier / DLC		216 217
0301h			
0302h			
0303h			
0304h			
0305h	CAN1 Message Box 10: Data Field		
0306h			
0307h			
0308h			
0309h			
030Ah			
030Bh			
030Ch	CAN1 Message Box 10: Time Stamp		
030Dh			
030Eh	CAN1 Message Box 11: Identifier / DLC		
030Fh			
0310h			
0311h			
0312h			
0313h	CAN1 Message Box 11: Data Field		
0314h			
0315h			
0316h			
0317h			
0318h			
0319h			
031Ah	CAN1 Message Box 11: Time Stamp		
031Bh			
031Ch	CAN1 Message Box 12: Identifier / DLC		
031Dh			
031Eh			
031Fh			
0320h			
0321h	CAN1 Message Box 12: Data Field		
0322h			
0323h			
0324h			
0325h			
0326h			
0327h			
0328h	CAN1 Message Box 12: Time Stamp		
0329h			
032Ah			
032Bh			
032Ch			
032Dh	CAN1 Message Box 13: Identifier / DLC		
032Eh			
032Fh			
0330h			
0331h			
0332h	CAN1 Message Box 13: Data Field		
0333h			
0334h			
0335h			
0336h			
0337h			
0338h			
0339h	CAN1 Message Box 13: Time Stamp		
033Ah			
033Bh			
033Ch			
033Dh			
033Eh	CAN1 Message Box 13: Time Stamp		
033Fh			

The blank areas are reserved.

Address	Register	Symbol	Page
0340h	CAN1 Message Box 14: Identifier / DLC		216 217
0341h			
0342h			
0343h			
0344h			
0345h	CAN1 Message Box 14: Data Field		
0346h			
0347h			
0348h			
0349h			
034Ah			
034Bh			
034Ch			
034Dh	CAN1 Message Box 14: Time Stamp		
034Eh			
034Fh	CAN1 Message Box 15: Identifier / DLC		
0350h			
0351h			
0352h			
0353h			
0354h			
0355h			
0356h	CAN1 Message Box 15: Data Field		
0357h			
0358h			
0359h			
035Ah			
035Bh			
035Ch			
035Dh	CAN1 Message Box 15: Time Stamp		
035Eh			
035Fh	CAN1 Global Mask Register	C1GMR	218
0360h			
0361h			
0362h			
0363h			
0364h	CAN1 Local Mask A Register	C1LMAR	218
0365h			
0366h			
0367h			
0368h			
0369h	CAN1 Local Mask B Register	C1LMBR	218
036Ah			
036Bh			
036Ch			
036Dh			
036Eh			
036Fh			
0370h			
0371h			
0372h			
0373h			
0374h			
0375h			
0376h			
0377h			
0378h			
0379h			
037Ah			
037Bh			
037Ch			
037Dh			
037Eh			
037Fh			

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

The blank areas are reserved.

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

# M16C/6N Group (M16C/6N4)

## SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

Rev.2.30  
Oct 24, 2005

### 1. Overview

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

#### 1.1 Applications

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

## 1.2 Performance Outline

Table 1.1 lists a performance outline of M16C/6N Group (M16C/6N4).

**Table 1.1 Performance Outline of M16C/6N Group (M16C/6N4)**

Item			Performance	
			Normal-ver.	T/V-ver.
CPU	Number of Basic Instructions		91 instructions	
	Minimum Instruction Execution Time		41.7ns (f(BCLK) = 24MHz, 1/1 prescaler, without software wait)	50.0ns (f(BCLK) = 20MHz, 1/1 prescaler, without software wait)
	Operation Mode		Single-chip, memory expansion and microprocessor modes	
	Address Space		1 Mbyte	
	Memory Capacity		See <b>Table 1.2 Product List</b>	
Peripheral Function	Port		Input/Output: 87 pins, Input: 1 pin	
	Multifunction Timer		Timer A: 16 bits × 5 channels Timer B: 16 bits × 6 channels Three-phase motor control circuit	
	Serial Interface		3 channels Clock synchronous, UART, I <sup>2</sup> C-bus <sup>(1)</sup> , IEBus <sup>(2)</sup> 1 channel Clock synchronous	
	A/D Converter		10-bit A/D converter: 1 circuit, 26 channels	
	D/A Converter		8 bits × 2 channels	
	DMAC		2 channels	
	CRC Calculation Circuit		CRC-CCITT	
	CAN Module		2 channels with 2.0B specification	
	Watchdog Timer		15 bits × 1 channel (with prescaler)	
	Interrupt		Internal: 31 sources, External: 9 sources Software: 4 sources, Priority level: 7 levels	
	Clock Generating 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 Function		Main clock oscillation stop and re-oscillation detection function	
Electrical Characteristics	Supply Voltage		VCC = 3.0 to 5.5V (f(BCLK) = 24MHz, 1/1 prescaler, without software wait)	VCC = 4.2 to 5.5V (f(BCLK) = 20MHz, 1/1 prescaler, without software wait)
	Power Consumption	Mask ROM	20mA (f(BCLK) = 24MHz, PLL operation, no division)	18mA (f(BCLK) = 20MHz, PLL operation, no division)
		Flash Memory	22mA (f(BCLK) = 24MHz, PLL operation, no division)	20mA (f(BCLK) = 20MHz, PLL operation, no division)
		Mask ROM	3μA (f(BCLK) = 32kHz, Wait mode, Oscillation capacity Low)	
		Flash Memory	0.8μA (Stop mode, Topr = 25°C)	
Flash Memory Version	Program/Erase Supply Voltage		3.0 ± 0.3V or 5.0 ± 0.5V	5.0 ± 0.5V
	Program and Erase Endurance		100 times	
I/O Characteristics	I/O Withstand Voltage		5.0V	
	Output Current		5mA	
Operating Ambient Temperature			-40 to 85°C	T version: -40 to 85°C V version: -40 to 125°C (option)
Device Configuration			CMOS high performance silicon gate	
Package			100-pin plastic mold QFP, LQFP	

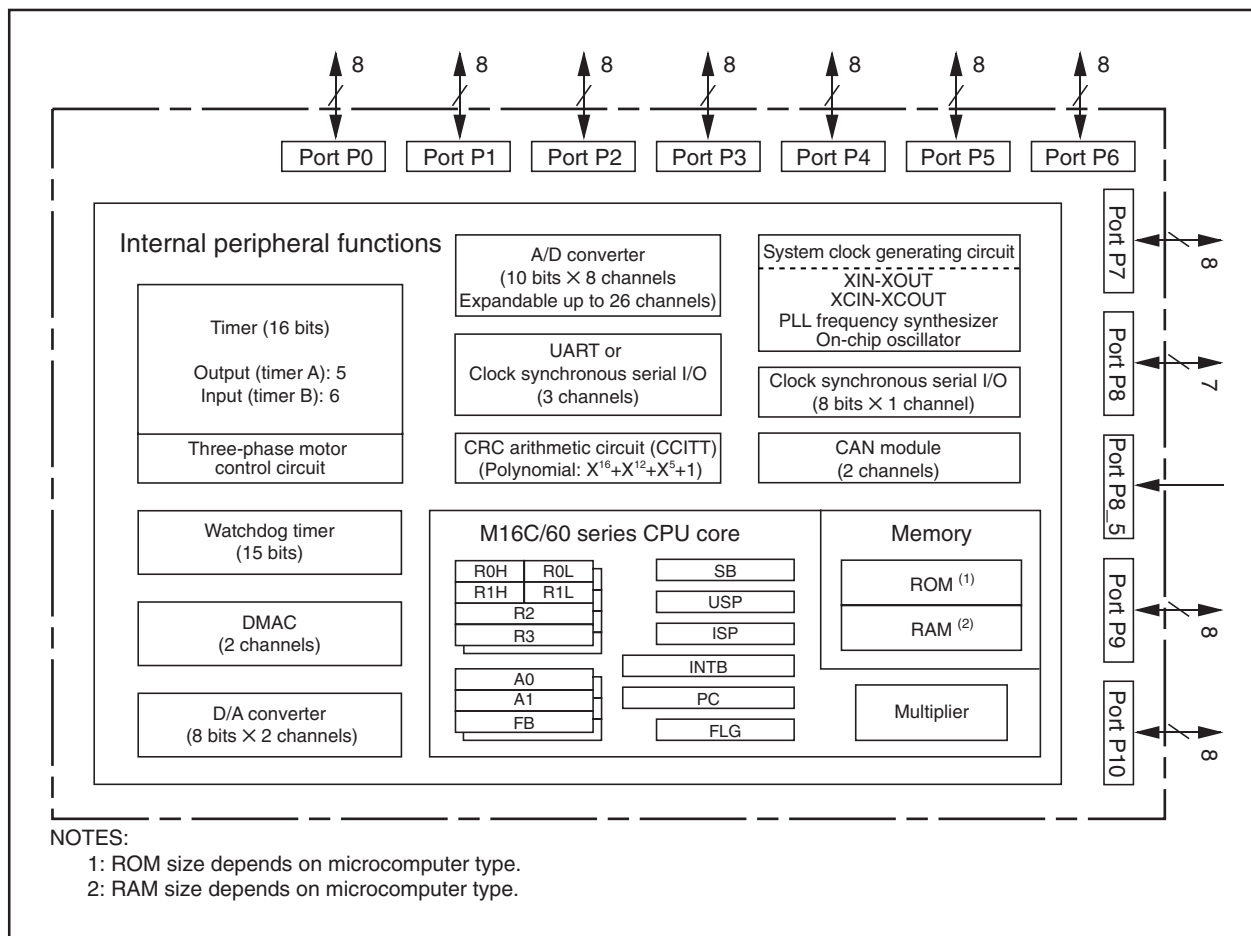
**NOTES:**

1. I<sup>2</sup>C-bus is a registered trademark of Koninklijke Philips Electronics N.V.
2. IEBus is a registered trademark of NEC Electronics Corporation.

option: All options are on request basis.

### 1.3 Block Diagram

Figure 1.1 shows a block diagram of M16C/6N Group (M16C/6N4).



**Figure 1.1 Block Diagram**

## 1.4 Product List

Table 1.2 lists the M16C/6N Group (M16C/6N4) products and Figure 1.2 shows the type numbers, memory sizes and packages.

**Table 1.2 Product List**

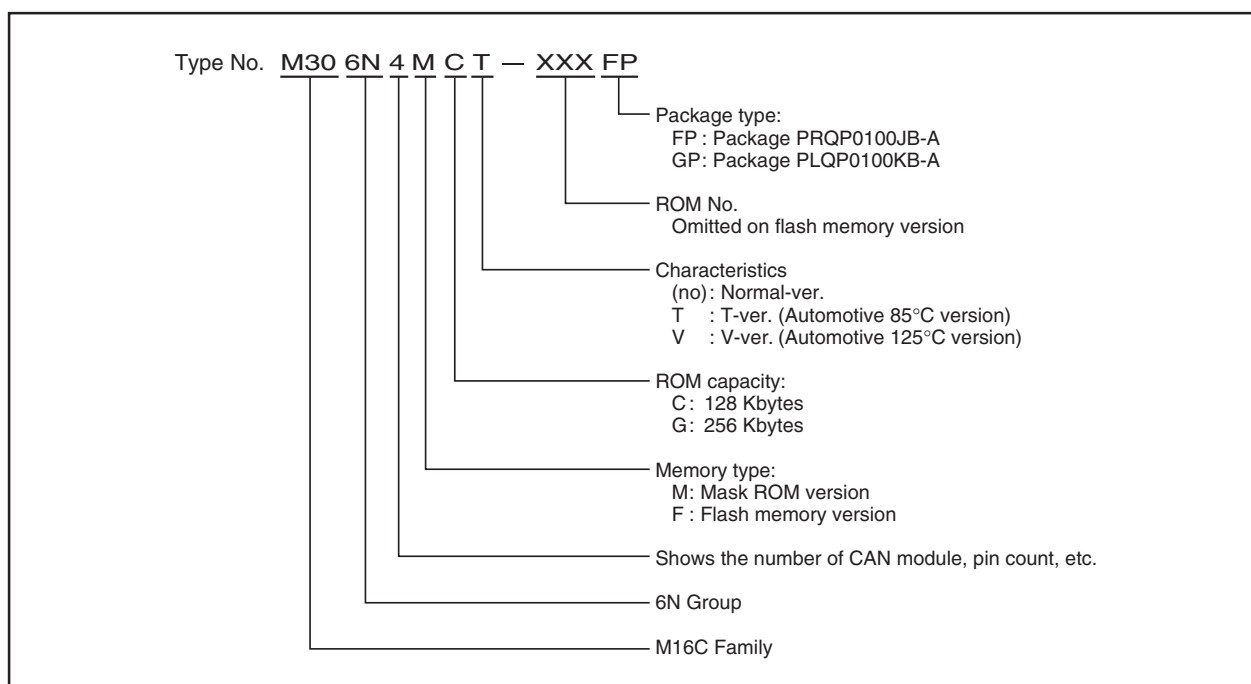
As of Oct. 2005

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

(D): Under development

NOTE:

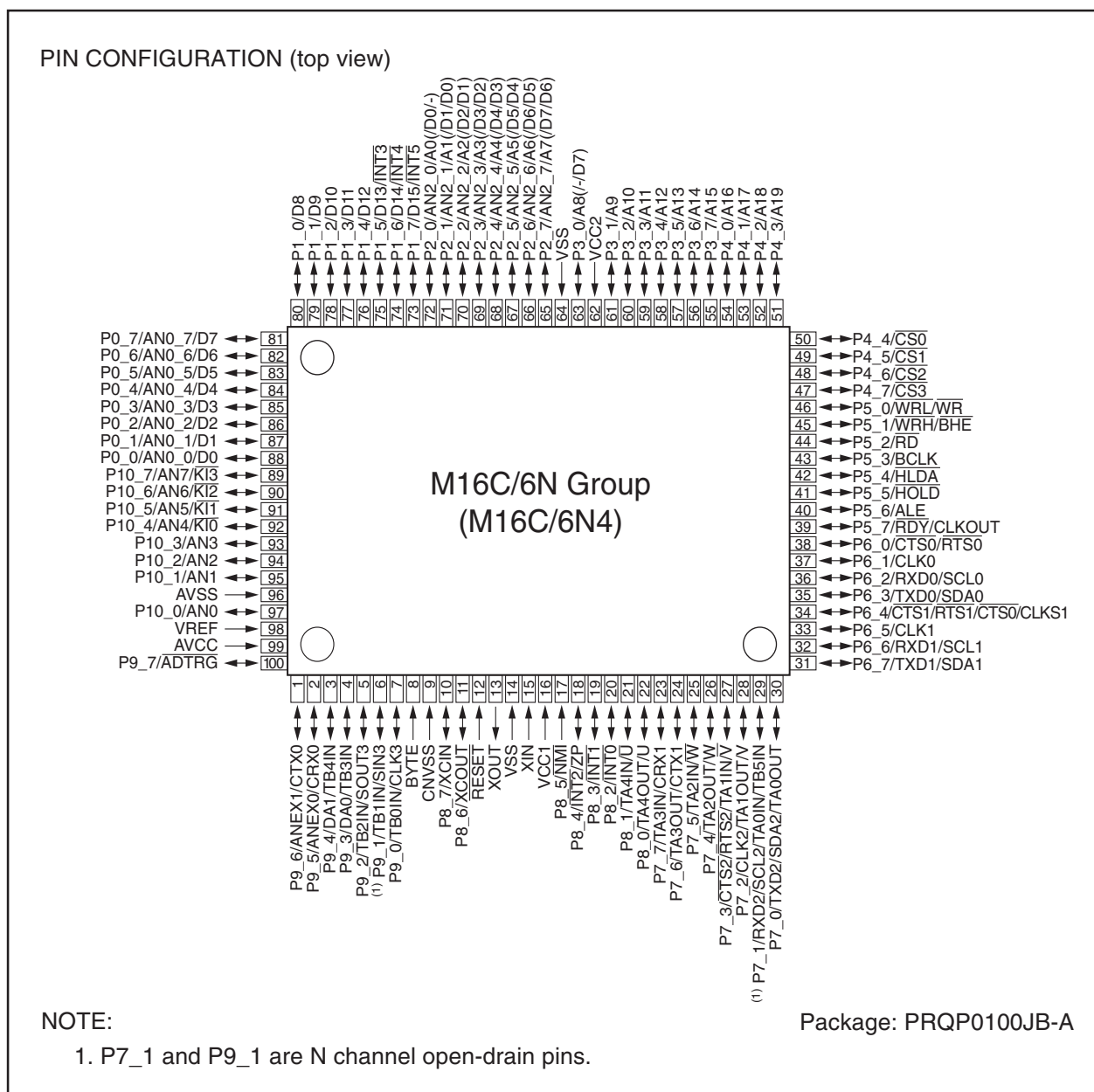
1. In the flash memory version, there is 4-Kbyte space (block A).



**Figure 1.2 Type No., Memory Size, and Package**

## 1.5 Pin Configuration

Figures 1.3 and 1.4 show the pin configuration (top view). Tables 1.3 and 1.4 list the pin characteristics.



**Figure 1.3 Pin Configuration (Top View) (1)**

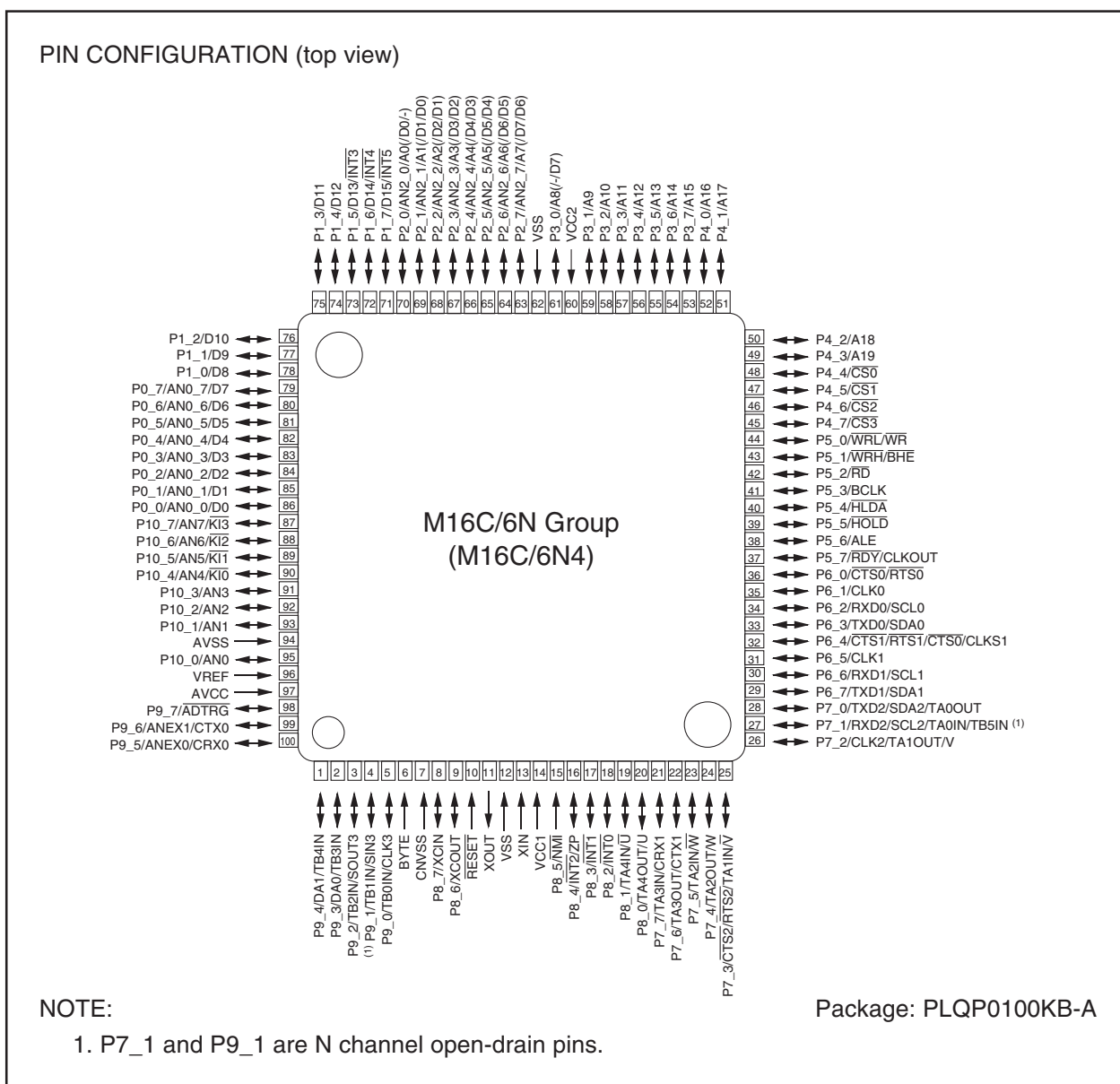


Figure 1.4 Pin Configuration (Top View) (2)



Table 1.3 Pin Characteristics (1)

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

FP: PRQP0100JB-A, GP: PLQP0100KB-A

Table 1.4 Pin Characteristics (2)

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

FP: PRQP0100JB-A, GP: PLQP0100KB-A

## 1.6 Pin Description

Tables 1.5 to 1.7 list the pin descriptions.

**Table 1.5 Pin Description (1)**

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

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

**NOTE:**

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

**Table 1.6 Pin Description (2)**

Signal Name	Pin Name	I/O Type	Description
Main clock input	XIN	I	I/O pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT <sup>(1)</sup> .
Main clock output	XOUT	O	To use the external clock, input the clock from XIN and leave XOUT open.
Sub clock input	XCIN	I	I/O pins for a sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOU <sup>(1)</sup> .
Sub clock output	XCOU	O	To use the external clock, input the clock from XCIN and leave XCOU open.
BCLK output	BCLK	O	Outputs the BCLK signal.
Clock output	CLKOUT	O	The clock of the same cycle as fC, f8, or f32 is output.
INT interrupt input	INT0 to INT5	I	Input pins for the INT interrupt.
NMI interrupt input	NMI	I	Input pin for the NMI interrupt.
Key input interrupt input	KI0 to KI3	I	Input pins for the key input interrupt.
Timer A	TA0OUT to TA4OUT	I/O	These are timer A0 to timer A4 I/O pins.
	TA0IN to TA4IN	I	These are timer A0 to timer A4 input pins.
	ZP	I	Input pin for the Z-phase.
Timer B	TB0IN to TB5IN	I	These are timer B0 to timer B5 input pins.
Three-phase motor control output	U, U, V, V, W, W	O	These are Three-phase motor control output pins.
Serial interface	CTS0 to CTS2	I	These are send control input pins.
	RTS0 to RTS2	O	These are receive control output pins.
	CLK0 to CLK3	I/O	These are transfer clock I/O pins.
	RXD0 to RXD2	I	These are serial data input pins.
	SIN3	I	These are serial data input pins.
	TXD0 to TXD2	O	These are serial data output pins.
	SOUT3	O	These are serial data output pins.
	CLKS1	O	This is output pin for transfer clock output from multiple pins function.
I <sup>2</sup> C mode	SDA0 to SDA2	I/O	These are serial data I/O pins.
	SCL0 to SCL2	I/O	These are transfer clock I/O pins. (however, SCL2 for the N-channel open drain output.)
Reference voltage input	VREF	I	Applies the reference voltage for the A/D converter and D/A converter.
A/D converter	AN0 to AN7	I	Analog input pins for the A/D converter.
	AN0_0 to AN0_7		
	AN2_0 to AN2_7		
	ADTRG	I	This is an A/D trigger input pin.
	ANEX0	I/O	This is the extended analog input pin for the A/D converter, and is the output in external op-amp connection mode.
	ANEX1	I	This is the extended analog input pin for the A/D converter.
D/A converter	DA0, DA1	O	These are the output pins for the D/A converter.
CAN module	CRX0, CRX1	I	These are the input pins for the CAN module.
	CTX0, CTX1	O	These are the output pins for the CAN module.

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

**NOTE:**

1. Ask the oscillator maker the oscillation characteristic.

**Table 1.7 Pin Description (3)**

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

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

## 2. Central Processing Unit (CPU)

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

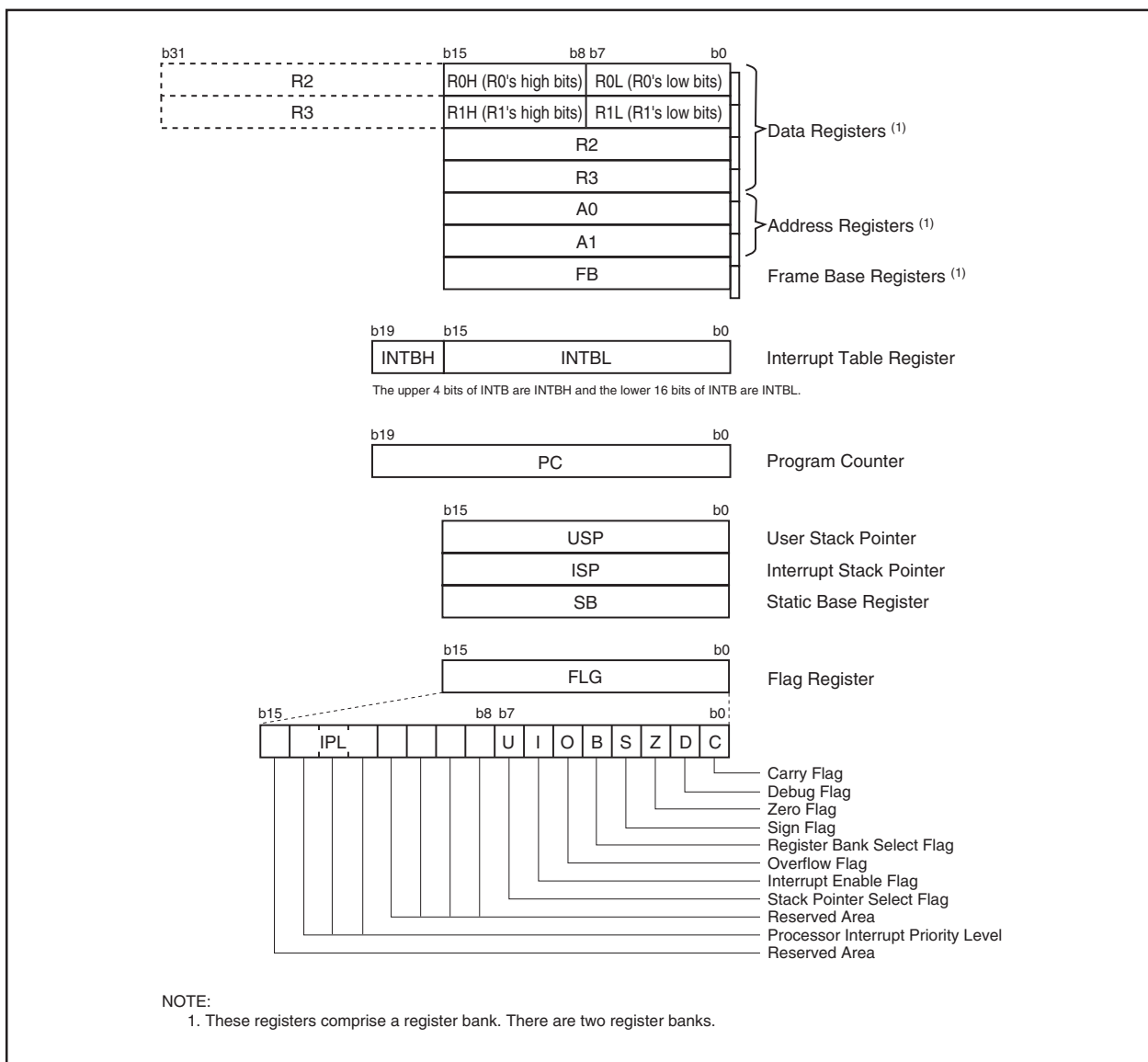


Figure 2.1 CPU Registers

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

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

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

### 2.2 Address Registers (A0 and A1)

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

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

### 2.3 Frame Base Register (FB)

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

### 2.4 Interrupt Table Register (INTB)

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

### 2.5 Program Counter (PC)

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

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

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

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

### 2.7 Static Base Register (SB)

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

### 2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

#### 2.8.1 Carry Flag (C Flag)

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

#### 2.8.2 Debug Flag (D Flag)

This flag is used exclusively for debugging purpose. During normal use, 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 set to "0" when the interrupt request is accepted.

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

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

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

#### 2.8.9 Processor Interrupt Priority Level (IPL)

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

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

#### 2.8.10 Reserved Area

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

### 3. Memory

Figure 3.1 shows a memory map of the M16C/6N Group (M16C/6N4). The address space extends the 1 Mbyte from address 00000h to FFFFFh.

The internal ROM is allocated in a lower address direction beginning with address FFFFFh. For example, a 128-Kbyte internal ROM is allocated to the addresses from E0000h to FFFFFh.

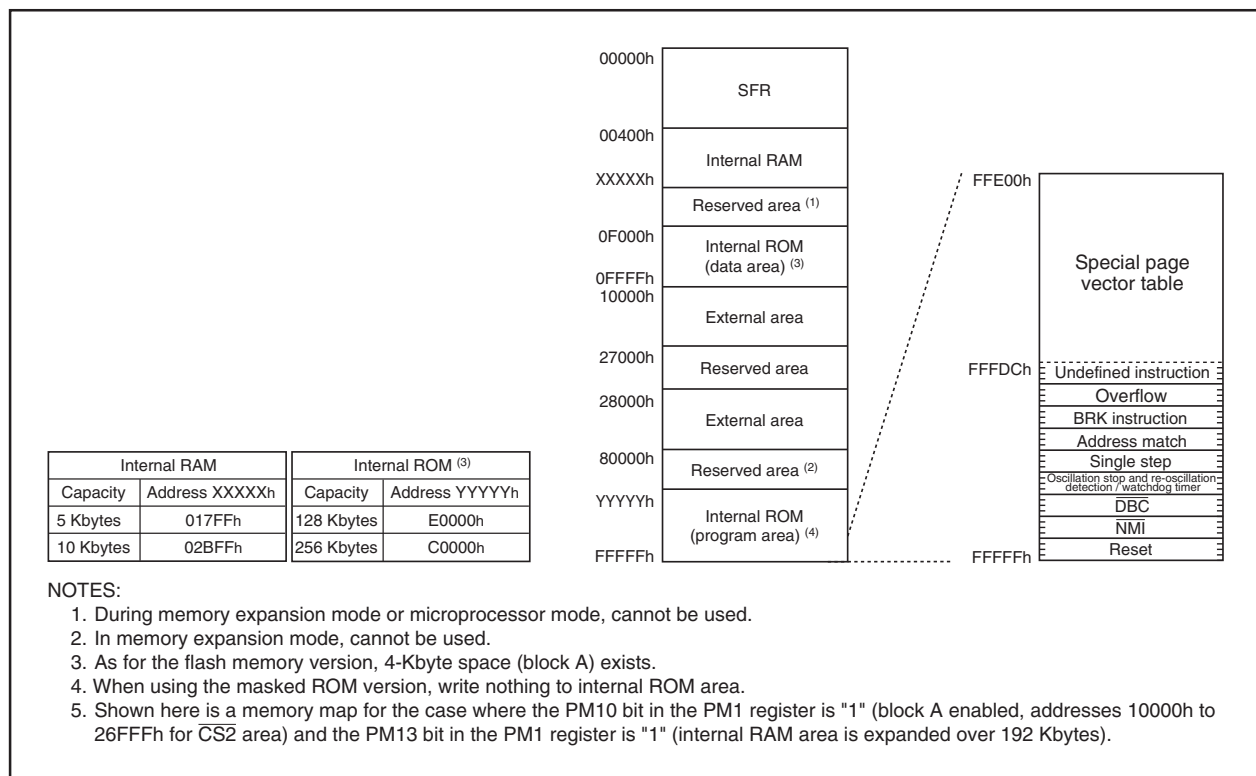
As for the flash memory version, 4-Kbyte space (block A) exists in 0F000h to 0FFFFh. 4-Kbyte space is mainly for storing data. In addition to storing data, 4-Kbyte space also can store programs.

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

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

The SFR is allocated to the addresses from 00000h to 003FFh. Peripheral function control registers are located here. Of the SFR, any area which has no functions allocated is reserved for future use and cannot be used by users.

The special page vector table is allocated to the addresses from FFE00h to FFFDBh. This vector is used by the JMPS or JSRS instruction. For details, refer to **M16C/60, M16C/20, M16C/Tiny Series Software Manual**. In memory expansion and microprocessor modes, some areas are reserved for future use and cannot be used by users.



**Figure 3.1 Memory Map**



## 4. Special Function Register (SFR)

SFR (Special Function Register) is the control register of peripheral functions.

Tables 4.1 to 4.16 list the SFR information.

**Table 4.1 SFR Information (1)**

Address	Register	Symbol	After Reset
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0 <sup>(1)</sup>	PM0	00000000b (CNVSS pin is "L") 00000011b (CNVSS pin is "H")
0005h	Processor Mode Register 1	PM1	00001000b
0006h	System Clock Control Register 0	CM0	01001000b
0007h	System Clock Control Register 1	CM1	00100000b
0008h	Chip Select Control Register	CSR	00000001b
0009h	Address Match Interrupt Enable Register	AIER	XXXXXX00b
000Ah	Protect Register	PRCR	XX000000b
000Bh			
000Ch	Oscillation Stop Detection Register <sup>(2)</sup>	CM2	0X000000b
000Dh			
000Eh	Watchdog Timer Start Register	WDTS	XXh
000Fh	Watchdog Timer Control Register	WDC	00XXXXXXb
0010h			00h
0011h	Address Match Interrupt Register 0	RMAD0	00h
0012h			X0h
0013h			
0014h			00h
0015h	Address Match Interrupt Register 1	RMAD1	00h
0016h			X0h
0017h			
0018h			
0019h			
001Ah			
001Bh	Chip Select Expansion Control Register	CSE	00h
001Ch	PLL Control Register 0	PLC0	0001X010b
001Dh			
001Eh	Processor Mode Register 2	PM2	XXX00000b
001Fh			
0020h	DMA0 Source Pointer	SAR0	XXh
0021h			XXh
0022h			XXh
0023h			
0024h	DMA0 Destination Pointer	DAR0	XXh
0025h			XXh
0026h			XXh
0027h			
0028h	DMA0 Transfer Counter	TCR0	XXh
0029h			XXh
002Ah			
002Bh			
002Ch	DMA0 Control Register	DM0CON	00000X00b
002Dh			
002Eh			
002Fh			
0030h	DMA1 Source Pointer	SAR1	XXh
0031h			XXh
0032h			XXh
0033h			
0034h	DMA1 Destination Pointer	DAR1	XXh
0035h			XXh
0036h			XXh
0037h			
0038h	DMA1 Transfer Counter	TCR1	XXh
0039h			XXh
003Ah			
003Bh			
003Ch	DMA1 Control Register	DM1CON	00000X00b
003Dh			
003Eh			
003Fh			

X: Undefined

### NOTES:

1. The PM00 and PM01 bits in the PM0 register do not change at software reset, watchdog timer reset and oscillation stop detection reset.
2. The CM20, CM21, and CM27 bits in the CM2 register do not change at oscillation stop detection reset.
3. The blank areas are reserved and cannot be accessed by users.

Table 4.2 SFR Information (2)

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

X: Undefined

## NOTE:

1. The blank area is reserved and cannot be accessed by users.

**Table 4.3 SFR Information (3)**

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

X: Undefined

Table 4.4 SFR Information (4)

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

X: Undefined

Table 4.5 SFR Information (5)

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

X: Undefined

Table 4.6 SFR Information (6)

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

X: Undefined

## NOTE:

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

Table 4.7 SFR Information (7)

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

X: Undefined

## NOTES:

1. These registers are included in the flash memory version. Cannot be accessed by users in the mask ROM version.
2. The blank areas are reserved and cannot be accessed by users.

Table 4.8 SFR Information (8)

Address	Register	Symbol	After Reset
01C0h	Timer B3, B4, B5 Count Start Flag	TBSR	000XXXXXb
01C1h	Timer A1-1 Register	TA11	XXh
01C2h			XXh
01C3h			XXh
01C4h	Timer A2-1 Register	TA21	XXh
01C5h			XXh
01C6h			XXh
01C7h	Timer A4-1 Register	TA41	XXh
01C8h			XXh
01C9h			XXh
01CAh	Three-Phase PWM Control Register 0	INVC0	00h
01CBh	Three-Phase PWM Control Register 1	INVC1	00h
01CAh	Three-Phase Output Buffer Register 0	IDB0	00h
01CBh	Three-Phase Output Buffer Register 1	IDB1	00h
01CCh	Dead Time Timer	DTT	XXh
01CDh	Timer B2 Interrupt Occurrence Frequency Set Counter	ICTB2	XXh
01CEh			
01CFh			
01D0h	Timer B3 Register	TB3	XXh
01D1h			XXh
01D2h			XXh
01D3h	Timer B4 Register	TB4	XXh
01D4h			XXh
01D5h			XXh
01D6h	Timer B5 Register	TB5	XXh
01D7h			XXh
01D8h			XXh
01D9h			
01DAh			
01DBh	Timer B3 Mode Register	TB3MR	00XX0000b
01DCh	Timer B4 Mode Register	TB4MR	00XX0000b
01DDh	Timer B5 Mode Register	TB5MR	00XX0000b
01DEh	Interrupt Cause Select Register 0	IFSR0	00XX0000b
01DFh	Interrupt Cause Select Register 1	IFSR1	00h
01E0h	SI/O3 Transmit/Receive Register	S3TRR	XXh
01E1h			
01E2h	SI/O3 Control Register	S3C	01000000b
01E3h	SI/O3 Bit Rate Generator	S3BRG	XXh
01E4h			
01E5h			
01E6h			
01E7h			
01E8h			
01E9h			
01EAh			
01EBh			
01ECh	UART0 Special Mode Register 4	U0SMR4	00h
01EDh	UART0 Special Mode Register 3	U0SMR3	000X0X0Xb
01EEh	UART0 Special Mode Register 2	U0SMR2	X0000000b
01EFh	UART0 Special Mode Register 1	U0SMR1	X0000000b
01F0h	UART1 Special Mode Register 4	U1SMR4	00h
01F1h	UART1 Special Mode Register 3	U1SMR3	000X0X0Xb
01F2h	UART1 Special Mode Register 2	U1SMR2	X0000000b
01F3h	UART1 Special Mode Register 1	U1SMR1	X0000000b
01F4h	UART2 Special Mode Register 4	U2SMR4	00h
01F5h	UART2 Special Mode Register 3	U2SMR3	000X0X0Xb
01F6h	UART2 Special Mode Register 2	U2SMR2	X0000000b
01F7h	UART2 Special Mode Register 1	U2SMR1	X0000000b
01F8h	UART2 Transmit/Receive Mode Register	U2MR	00h
01F9h	UART2 Bit Rate Generator	U2BRG	XXh
01FAh	UART2 Transmit Buffer Register	U2TB	XXh
01FBh			XXh
01FCh	UART2 Transmit/Receive Control Register 0	U2C0	00001000b
01FDh	UART2 Transmit/Receive Control Register 1	U2C1	00000010b
01FEh	UART2 Receive Buffer Register	U2RB	XXh
01FFh			XXh

X: Undefined

## NOTE:

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



Table 4.9 SFR Information (9)

Address	Register	Symbol	After Reset
0200h	CAN0 Message Control Register 0	C0MCTL0	00h
0201h	CAN0 Message Control Register 1	C0MCTL1	00h
0202h	CAN0 Message Control Register 2	C0MCTL2	00h
0203h	CAN0 Message Control Register 3	C0MCTL3	00h
0204h	CAN0 Message Control Register 4	C0MCTL4	00h
0205h	CAN0 Message Control Register 5	C0MCTL5	00h
0206h	CAN0 Message Control Register 6	C0MCTL6	00h
0207h	CAN0 Message Control Register 7	C0MCTL7	00h
0208h	CAN0 Message Control Register 8	C0MCTL8	00h
0209h	CAN0 Message Control Register 9	C0MCTL9	00h
020Ah	CAN0 Message Control Register 10	C0MCTL10	00h
020Bh	CAN0 Message Control Register 11	C0MCTL11	00h
020Ch	CAN0 Message Control Register 12	C0MCTL12	00h
020Dh	CAN0 Message Control Register 13	C0MCTL13	00h
020Eh	CAN0 Message Control Register 14	C0MCTL14	00h
020Fh	CAN0 Message Control Register 15	C0MCTL15	00h
0210h	CAN0 Control Register	C0CTLR	X0000001b
0211h			XX0X0000b
0212h	CAN0 Status Register	C0STR	00h
0213h			X0000001b
0214h	CAN0 Slot Status Register	C0SSTR	00h
0215h			00h
0216h	CAN0 Interrupt Control Register	C0ICR	00h
0217h			00h
0218h	CAN0 Extended ID Register	C0IDR	00h
0219h			00h
021Ah	CAN0 Configuration Register	C0CONR	XXh
021Bh			XXh
021Ch	CAN0 Receive Error Count Register	C0RECR	00h
021Dh	CAN0 Transmit Error Count Register	C0TECR	00h
021Eh	CAN0 Time Stamp Register	C0TSR	00h
021Fh			00h
0220h	CAN1 Message Control Register 0	C1MCTL0	00h
0221h	CAN1 Message Control Register 1	C1MCTL1	00h
0222h	CAN1 Message Control Register 2	C1MCTL2	00h
0223h	CAN1 Message Control Register 3	C1MCTL3	00h
0224h	CAN1 Message Control Register 4	C1MCTL4	00h
0225h	CAN1 Message Control Register 5	C1MCTL5	00h
0226h	CAN1 Message Control Register 6	C1MCTL6	00h
0227h	CAN1 Message Control Register 7	C1MCTL7	00h
0228h	CAN1 Message Control Register 8	C1MCTL8	00h
0229h	CAN1 Message Control Register 9	C1MCTL9	00h
022Ah	CAN1 Message Control Register 10	C1MCTL10	00h
022Bh	CAN1 Message Control Register 11	C1MCTL11	00h
022Ch	CAN1 Message Control Register 12	C1MCTL12	00h
022Dh	CAN1 Message Control Register 13	C1MCTL13	00h
022Eh	CAN1 Message Control Register 14	C1MCTL14	00h
022Fh	CAN1 Message Control Register 15	C1MCTL15	00h
0230h	CAN1 Control Register	C1CTLR	X0000001b
0231h			XX0X0000b
0232h	CAN1 Status Register	C1STR	00h
0233h			X0000001b
0234h	CAN1 Slot Status Register	C1SSTR	00h
0235h			00h
0236h	CAN1 Interrupt Control Register	C1ICR	00h
0237h			00h
0238h	CAN1 Extended ID Register	C1IDR	00h
0239h			00h
023Ah	CAN1 Configuration Register	C1CONR	XXh
023Bh			XXh
023Ch	CAN1 Receive Error Count Register	C1RECR	00h
023Dh	CAN1 Transmit Error Count Register	C1TECR	00h
023Eh	CAN1 Time Stamp Register	C1TSR	00h
023Fh			00h

X: Undefined

Table 4.10 SFR Information (10)

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

X: Undefined

## NOTE:

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

**Table 4.11 SFR Information (11)**

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

X: Undefined

Table 4.12 SFR Information (12)

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

X: Undefined

**Table 4.13 SFR Information (13)**

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

X: Undefined

Table 4.14 SFR Information (14)

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

X: Undefined

## NOTE:

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

Table 4.15 SFR Information (15)

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

X: Undefined

## NOTES:

1. The TA2P to TA4P bits in the UDF register are set to "0" after reset. However, the contents in these bits are indeterminate when read.
2. The blank areas are reserved and cannot be accessed by users.

Table 4.16 SFR Information (16)

Address	Register	Symbol	After Reset
03C0h	A/D Register 0	AD0	XXh
03C1h			XXh
03C2h			XXh
03C3h	A/D Register 1	AD1	XXh
03C4h			XXh
03C5h	A/D Register 2	AD2	XXh
03C6h			XXh
03C7h	A/D Register 3	AD3	XXh
03C8h			XXh
03C9h			XXh
03CAh	A/D Register 4	AD4	XXh
03CBh			XXh
03CCh	A/D Register 5	AD5	XXh
03CDh			XXh
03CEh	A/D Register 6	AD6	XXh
03CFh			XXh
03D0h			XXh
03D1h	A/D Register 7	AD7	XXh
03D2h			XXh
03D3h			XXh
03D4h	A/D Control Register 2	ADCON2	00h
03D5h	A/D Control Register 0	ADCON0	00000XXXb
03D6h			00h
03D7h	A/D Control Register 1	ADCON1	00h
03D8h	D/A Register 0	DA0	00h
03D9h	D/A Register 1	DA1	00h
03DAh			00h
03DBh			00h
03DCh	D/A Control Register	DACON	00h
03DDh			
03DEh			
03DFh			
03E0h	Port P0 Register	P0	XXh
03E1h	Port P1 Register	P1	XXh
03E2h	Port P0 Direction Register	PD0	00h
03E3h	Port P1 Direction Register	PD1	00h
03E4h	Port P2 Register	P2	XXh
03E5h	Port P3 Register	P3	XXh
03E6h	Port P2 Direction Register	PD2	00h
03E7h	Port P3 Direction Register	PD3	00h
03E8h	Port P4 Register	P4	XXh
03E9h	Port P5 Register	P5	XXh
03EAh	Port P4 Direction Register	PD4	00h
03EBh	Port P5 Direction Register	PD5	00h
03ECh	Port P6 Register	P6	XXh
03EDh	Port P7 Register	P7	XXh
03EEh	Port P6 Direction Register	PD6	00h
03EFh	Port P7 Direction Register	PD7	00h
03F0h	Port P8 Register	P8	XXh
03F1h	Port P9 Register	P9	XXh
03F2h	Port P8 Direction Register	PD8	00X00000b
03F3h	Port P9 Direction Register	PD9	00h
03F4h	Port P10 Register	P10	XXh
03F5h	Port P10 Direction Register	PD10	00h
03F6h			
03F7h			
03F8h			
03F9h			
03FAh			
03FBh	Pull-up Control Register 0	PUR0	00h
03FCh			
03FDh			
03FEh	Pull-up Control Register 1	PUR1	00000000b <sup>(1)</sup>
03FFh	Pull-up Control Register 2	PUR2	0000010b
	Port Control Register	PCR	00h

X: Undefined

## NOTES:

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

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

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

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

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



## 5. Reset

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

### 5.1 Hardware Reset

The microcomputer resets pins, the CPU and SFR by setting the  $\overline{\text{RESET}}$  pin. If the supply voltage meets the recommended operating conditions, the microcomputer resets all pins when an “L” signal is applied to the  $\overline{\text{RESET}}$  pin (see **Table 5.1 Pin Status When  $\overline{\text{RESET}}$  Pin Level is “L”**). The oscillation circuit is also reset and the main clock starts oscillation. The microcomputer resets the CPU and SFR when the signal applied to the  $\overline{\text{RESET}}$  pin changes low (“L”) to high (“H”). The microcomputer executes the program in an address indicated by the reset vector. The internal RAM is not reset. When an “L” signal is applied to the  $\overline{\text{RESET}}$  pin while writing data to the internal RAM, the internal RAM is in an indeterminate state.

Figure 5.1 shows an example of the reset circuit. Figure 5.2 shows a reset sequence. Table 5.1 lists pin states while the  $\overline{\text{RESET}}$  pin is held low (“L”).

#### 5.1.1 Reset on a Stable Supply Voltage

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

#### 5.1.2 Power-on Reset

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

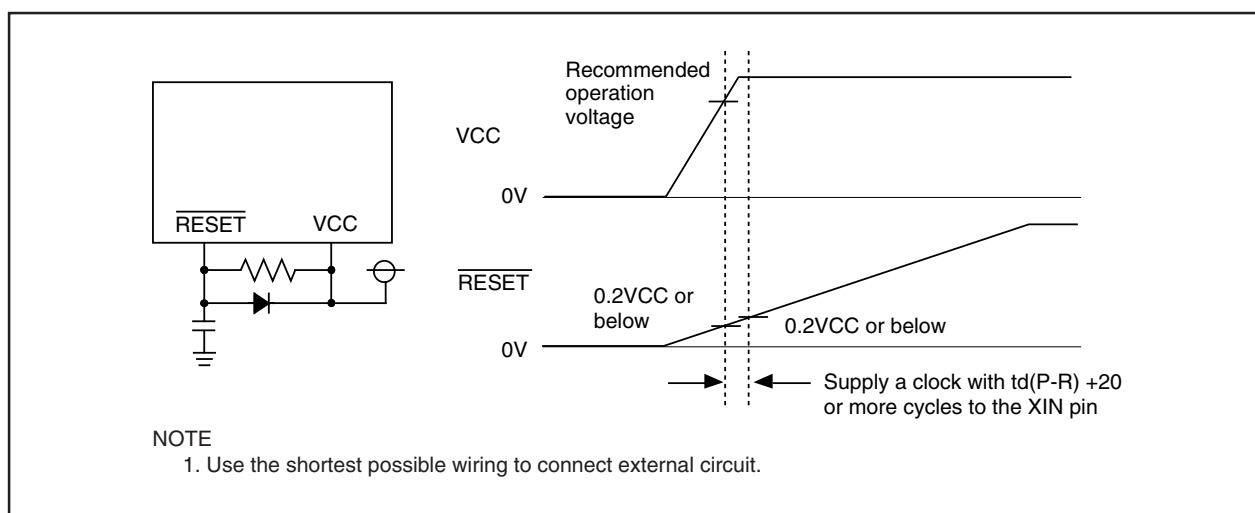


Figure 5.1 Example Reset Circuit

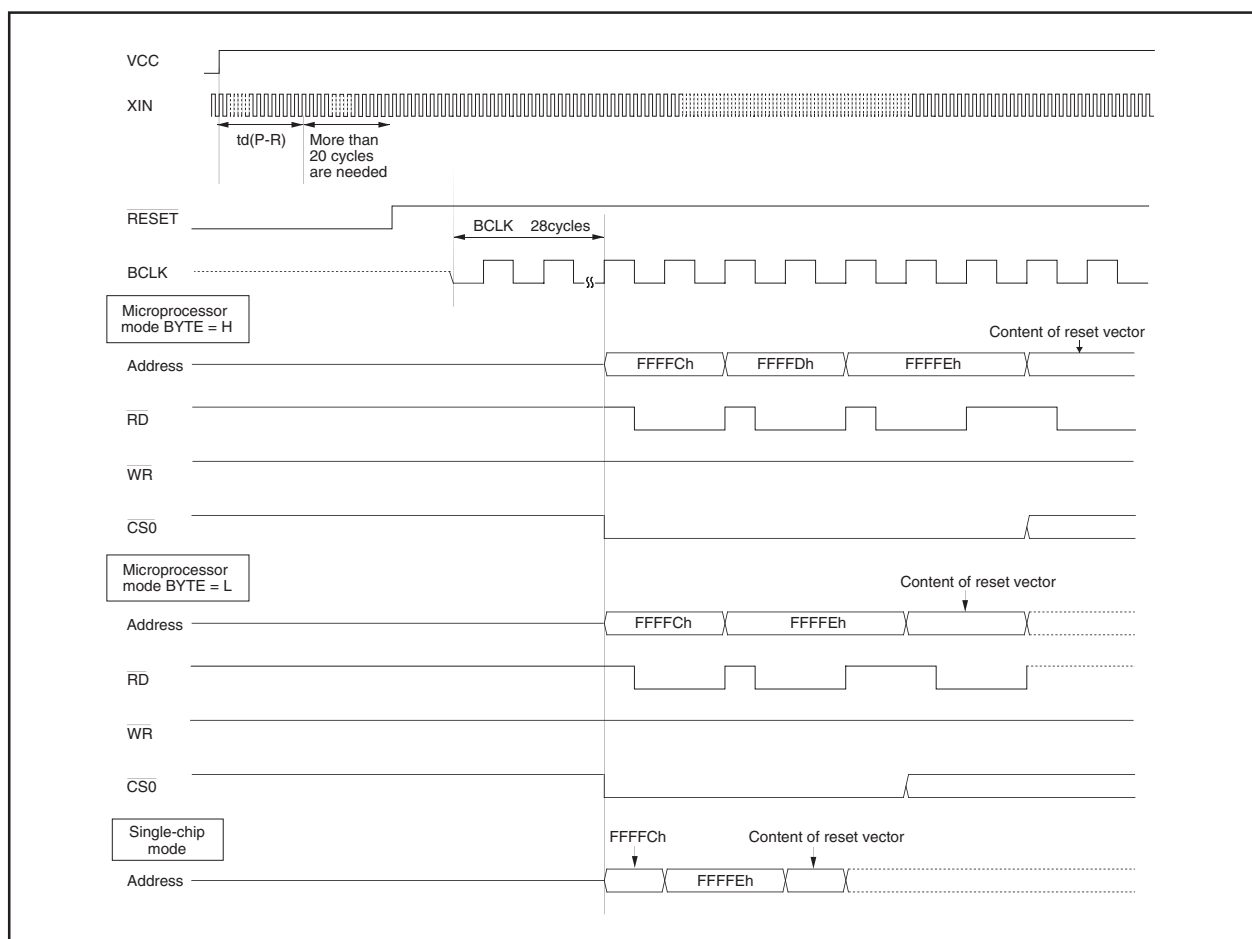


Figure 5.2 Reset Sequence

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

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

## NOTE:

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

## 5.2 Software Reset

The microcomputer resets pins, the CPU and SFR when the PM03 bit in the PM0 register is set to “1” (microcomputer reset). Then the microcomputer executes the program in an address determined by the reset vector. Set the PM03 bit to “1” while the main clock is selected as the CPU clock and the main clock oscillation is stable. In the software reset, the microcomputer does not reset a part of the SFR. Refer to **4. Special Function Register (SFR)** for details.

Processor mode remains unchanged since the PM01 to PM00 bits in the PM0 register are not reset.

## 5.3 Watchdog Timer Reset

The microcomputer resets pins, the CPU and SFR when the PM12 bit in the PM1 register is set to “1” (reset when watchdog timer underflows) and the watchdog timer underflows. Then the microcomputer executes the program in an address determined by the reset vector.

In the watchdog timer reset, the microcomputer does not reset a part of the SFR. Refer to **4. Special Function Register (SFR)** for details.

Processor mode remains unchanged since the PM01 to PM00 bits in the PM0 register are not reset.

## 5.4 Oscillation Stop Detection Reset

The microcomputer resets and stops pins, the CPU and SFR when the CM27 bit in the CM2 register is “0” (reset at oscillation stop, re-oscillation detection), if it detects main clock oscillation circuit stop. Refer to **8.5 Oscillation Stop and Re-Oscillation Detection Function** for details.

In the oscillation stop detection reset, the microcomputer does not reset a part of the SFR. Refer to **4. Special Function Register (SFR)** for details.

Processor mode remains unchanged since the PM01 to PM00 bits in the PM0 register are not reset.

## 5.5 Internal Space

Figure 5.3 shows CPU register status after reset. Refer to **4. Special Function Register (SFR)** for SFR states after reset.

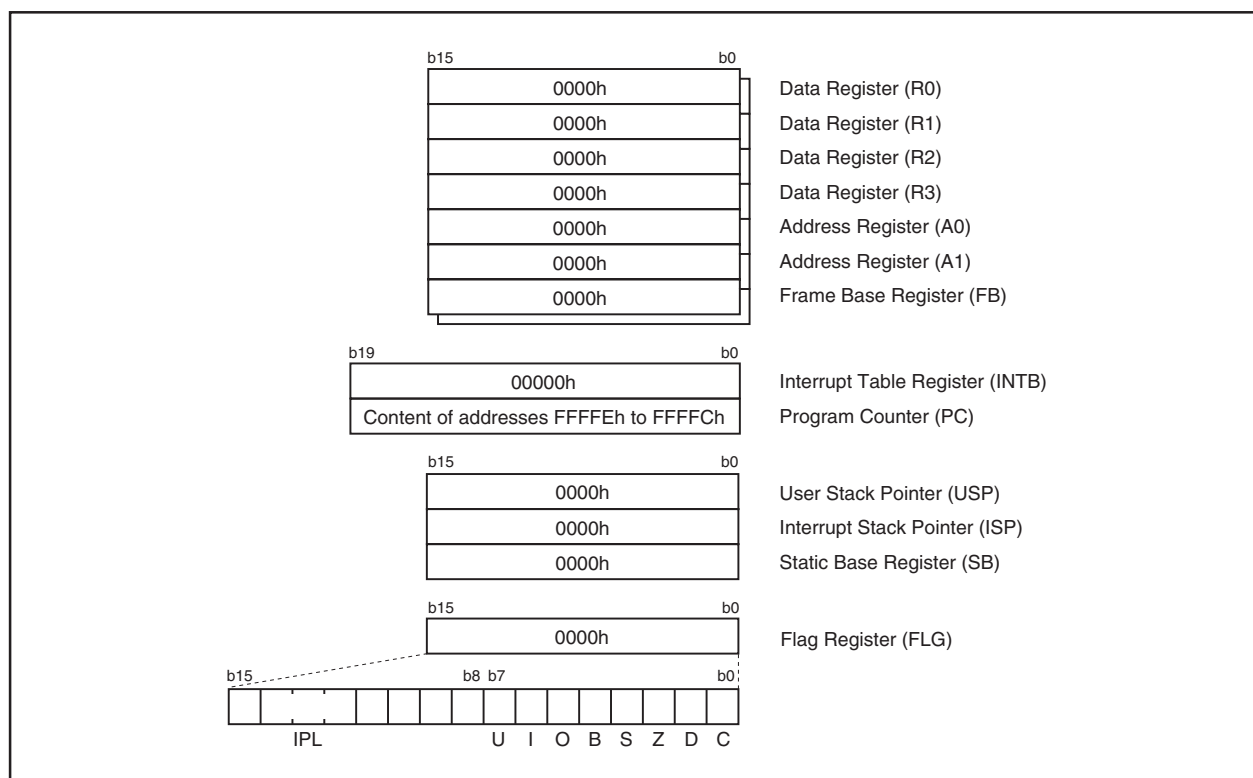


Figure 5.3 CPU Register Status After Reset

## 6. Processor Mode

### 6.1 Types of Processor Mode

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

**Table 6.1 Features of Processor Modes**

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

NOTE:

1. Refer to 7. Bus.

### 6.2 Setting Processor Modes

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

Table 6.2 shows the processor mode after hardware reset. Table 6.3 shows the PM01 to PM00 bits set values and processor modes.

**Table 6.2 Processor Mode After Hardware Reset**

CNVSS Pin Input Level	Processor Mode
VSS	Single-chip mode
VCC <sup>(1) (2)</sup>	Microprocessor mode

NOTES:

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

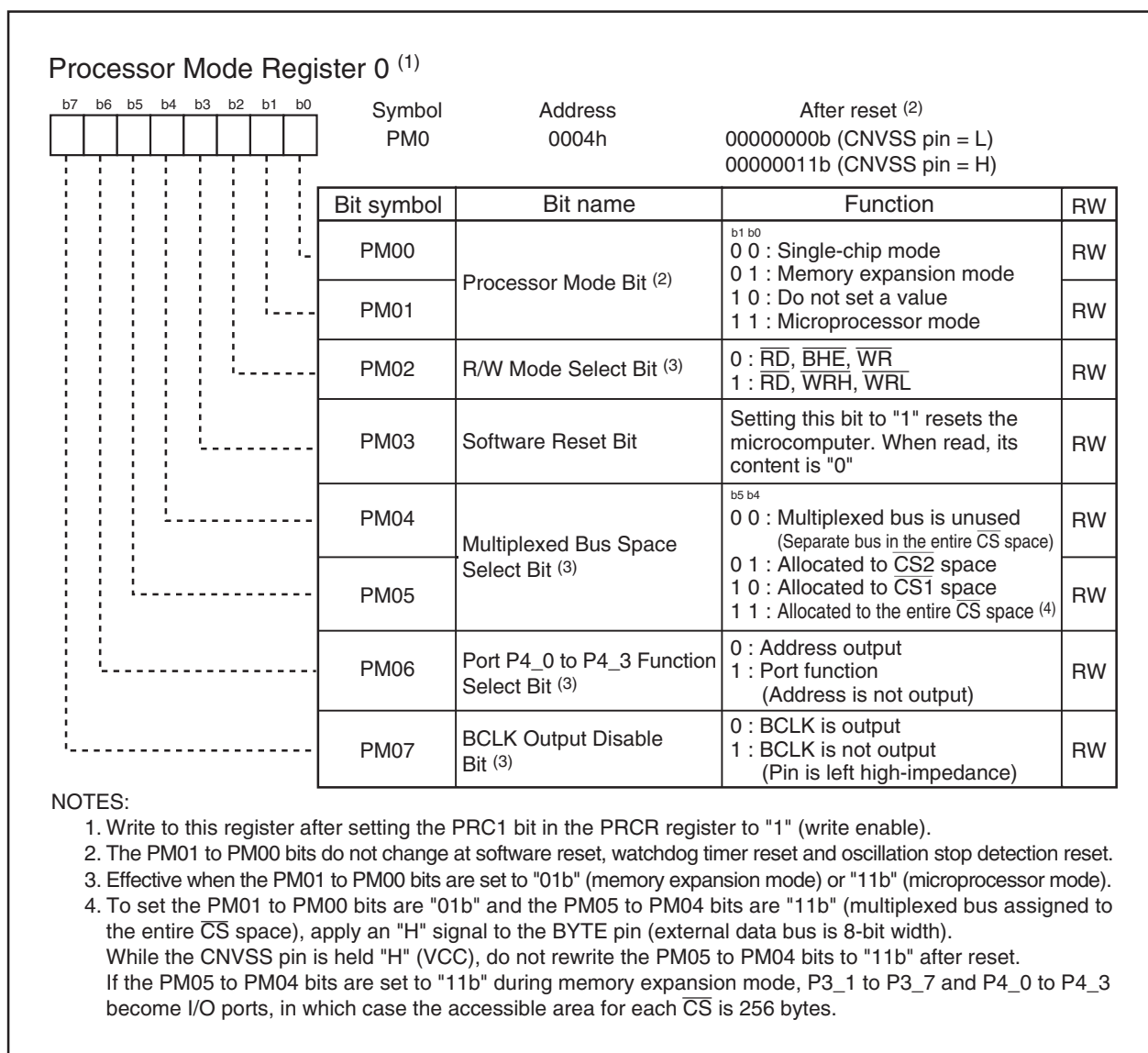
**Table 6.3 PM01 to PM00 Bits Set Values and Processor Modes**

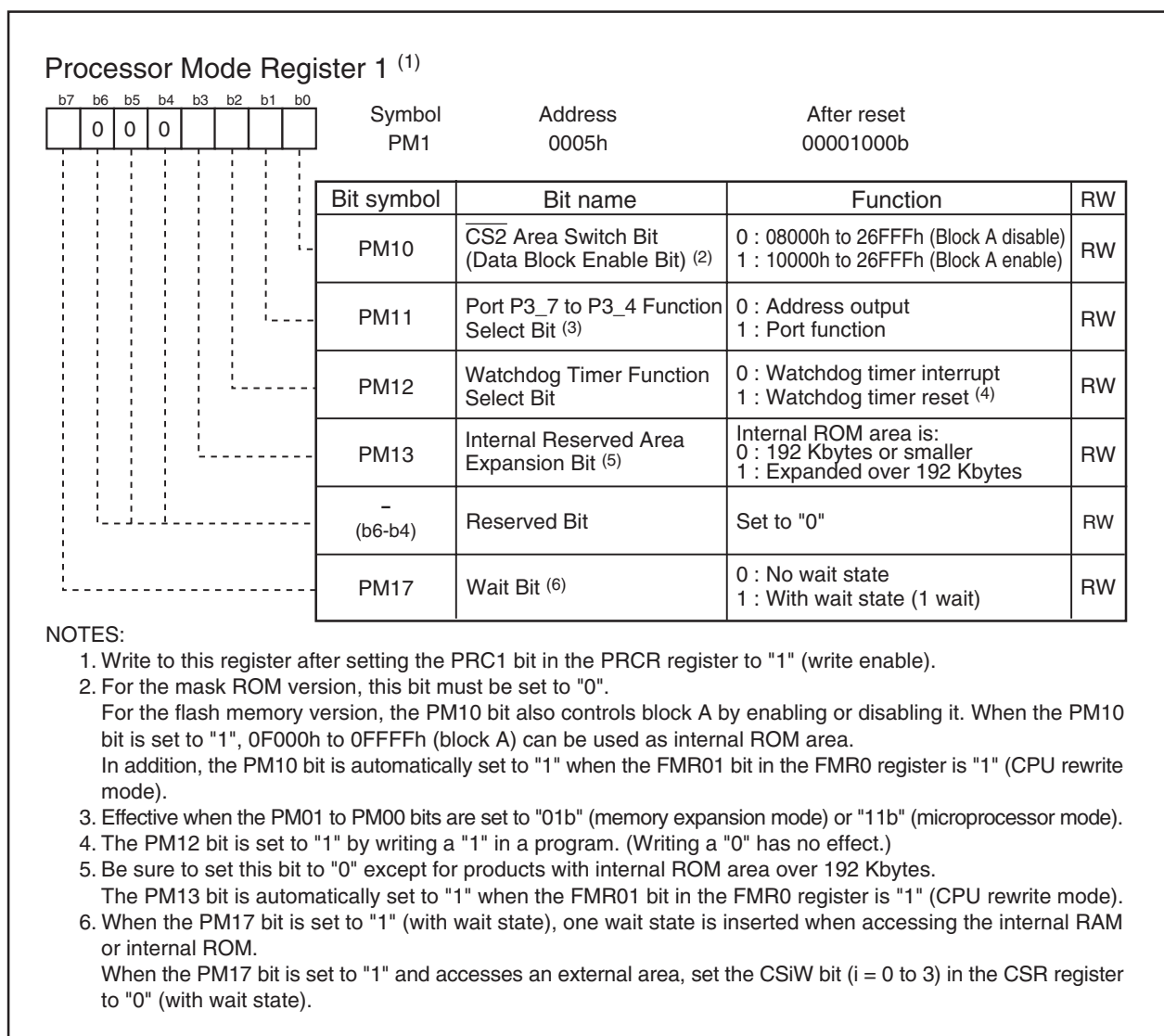
PM01 to PM 00 Bits	Processor Mode
00b	Single-chip mode
01b	Memory expansion mode
10b	Do not set a value
11b	Microprocessor mode

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

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

Figures 6.1 and 6.2 show the processor mode related registers. Figure 6.3 shows the memory map in single-chip mode. Figures 6.4 to 6.7 show the memory map and  $\overline{CS}$  area in memory expansion mode and microprocessor mode.

**Figure 6.1 PM0 Register**

**Figure 6.2 PM1 Register**

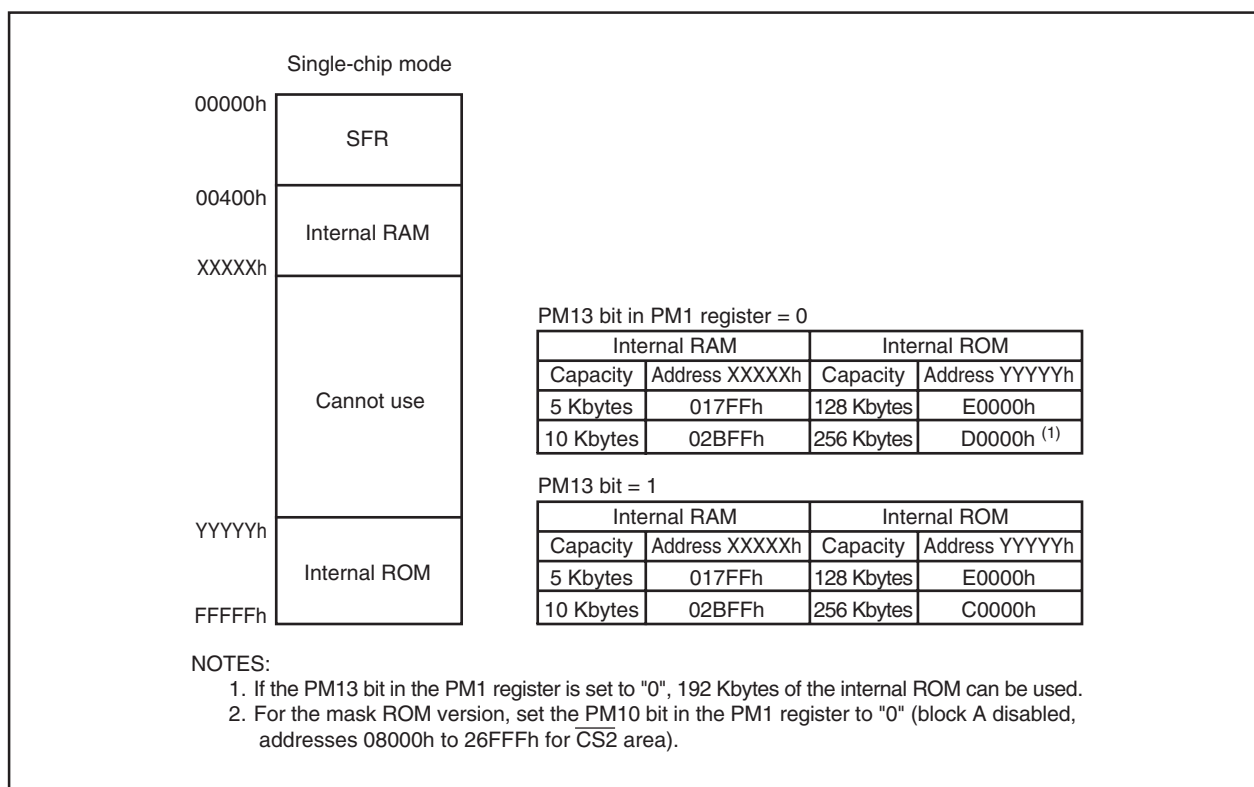
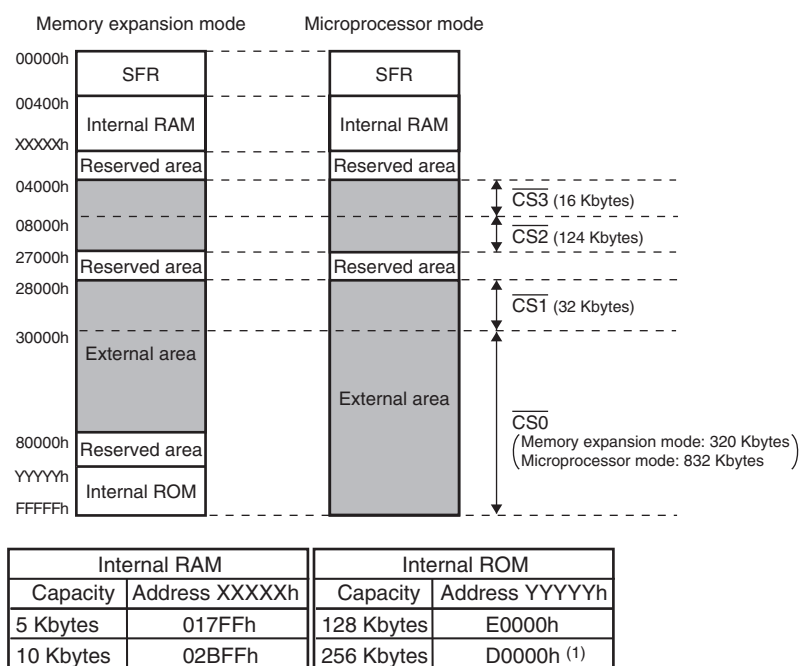


Figure 6.3 Memory Map in Single-chip Mode

■ When PM13 = 0 and PM10 = 0



NOTE:

1. If the PM13 bit in the PM1 register is set to "0", 192 Kbytes of the internal ROM can be used.

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

■ When PM13 = 1 and PM10 = 0

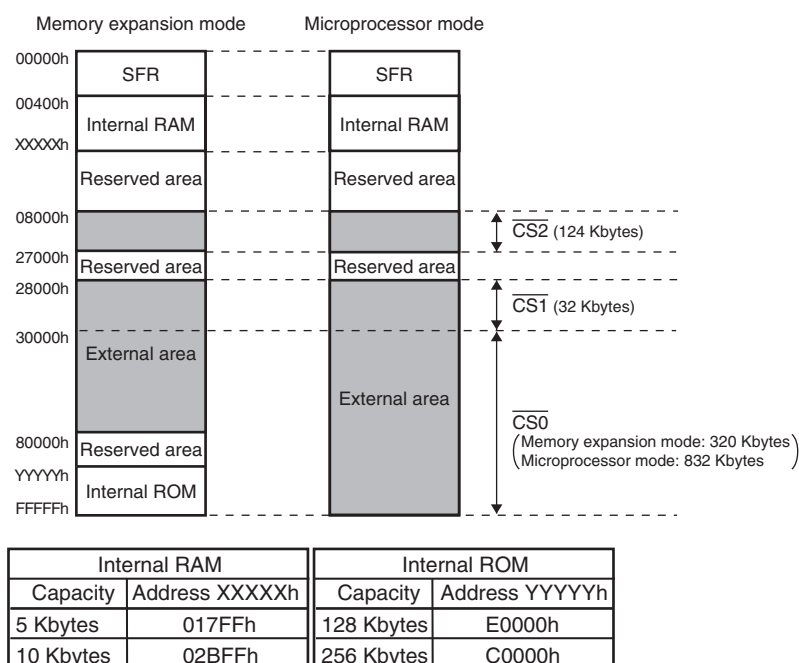
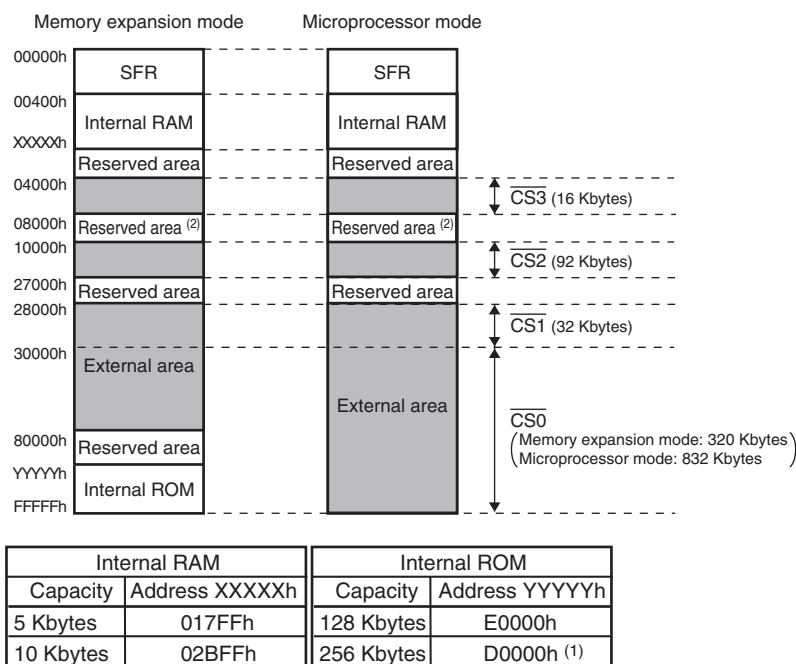


Figure 6.5 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (2)



■ When PM13 = 0 and PM10 = 1

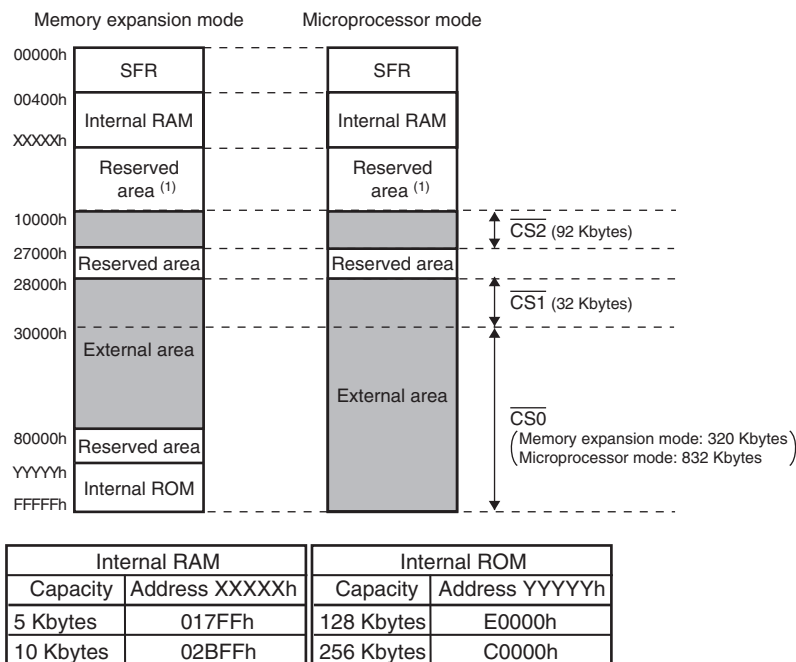


## NOTES:

1. If the PM13 bit in the PM1 register is set to "0", 192 Kbytes of the internal ROM can be used.
2. For the flash memory version, when the PM10 bit is set to "1", 0F000h to 0FFFFh (block A) can be used as internal ROM area.

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

■ When PM13 = 1 and PM10 = 1



## NOTE:

1. For the flash memory version, when the PM10 bit is set to "1", 0F000h to 0FFFFh (block A) can be used as internal ROM area.

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

## 7. Bus

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

### 7.1 Bus Mode

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

#### 7.1.1 Separate Bus

In this bus mode, data and address are separate.

#### 7.1.2 Multiplexed Bus

In this bus mode, data and address are multiplexed.

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

D0 to D7 and A0 to A7 are multiplexed.

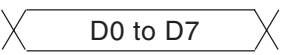

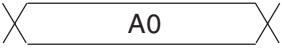

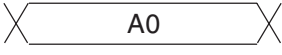
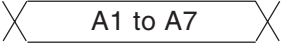


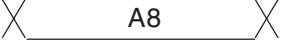
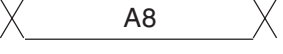

##### 7.1.2.2 When the input level on BYTE pin is low (16-bit data bus)

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

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

Table 7.1 shows the difference between a separate bus and multiplexed bus.

**Table 7.1 Difference between Separate Bus and Multiplexed Bus**

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

NOTES :

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

## 7.2 Bus Control

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

### 7.2.1 Address Bus

The address bus consists of 20 lines, A0 to A19. The address bus width can be chosen to be 12, 16 or 20 bits by using the PM06 bit in the PM0 register and the PM11 bit in the PM1 register. Table 7.2 shows the PM06 and PM11 bits set values and address bus widths.

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

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

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

NOTE:

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

### 7.2.2 Data Bus

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

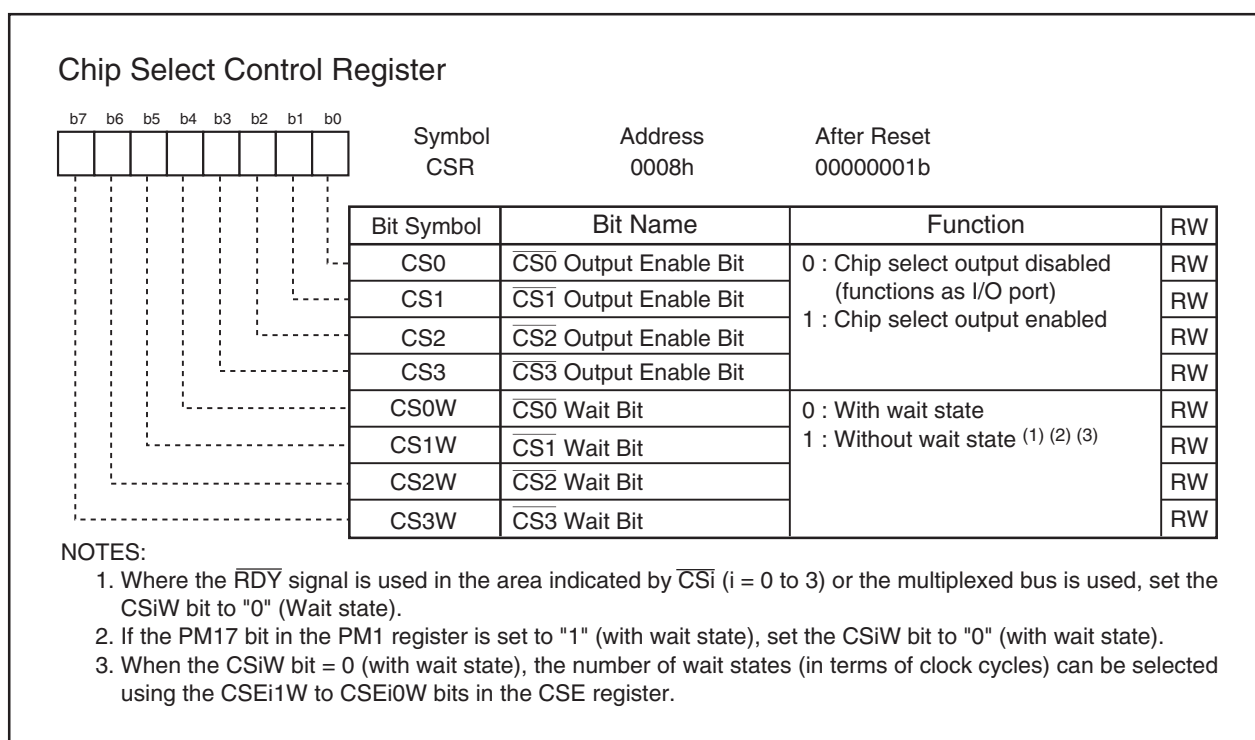
### 7.2.3 Chip Select Signal

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

Figure 7.1 shows the CSR register.

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

Figure 7.2 shows the example of address bus and  $\overline{CS}_i$  signal output.

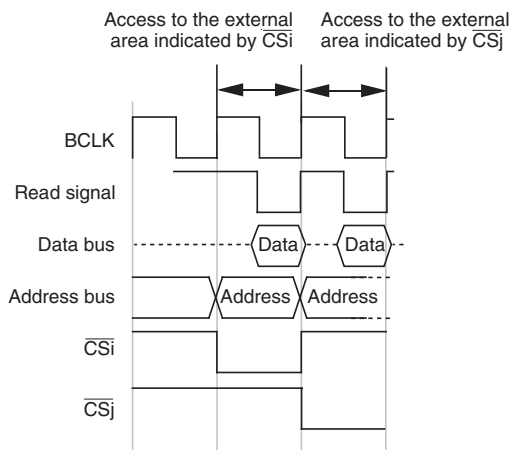


**Figure 7.1 CSR Register**

## Example 1

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

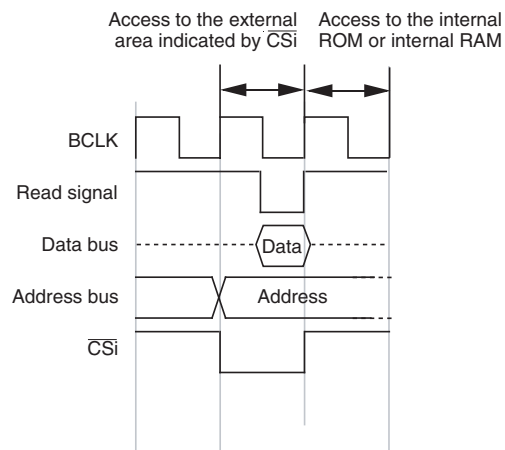
The address bus and the chip select signal both change state between these two cycles.



## Example 2

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

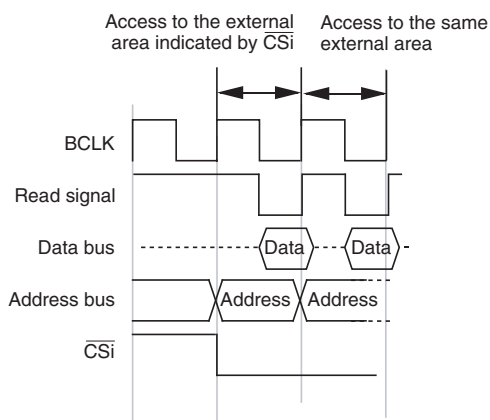
The chip select signal changes state but the address bus does not change state.



## Example 3

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

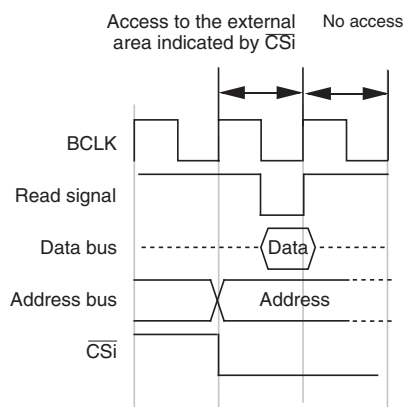
The address bus changes state but the chip select signal does not change state.



## Example 4

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

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



## NOTE:

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

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

Figure 7.2 Example of Address Bus and  $\overline{CSi}$  Signal Output

### 7.2.4 Read and Write Signals

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

Table 7.3 shows the operation of  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  signals. Table 7.4 shows the operation of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$  signals.

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

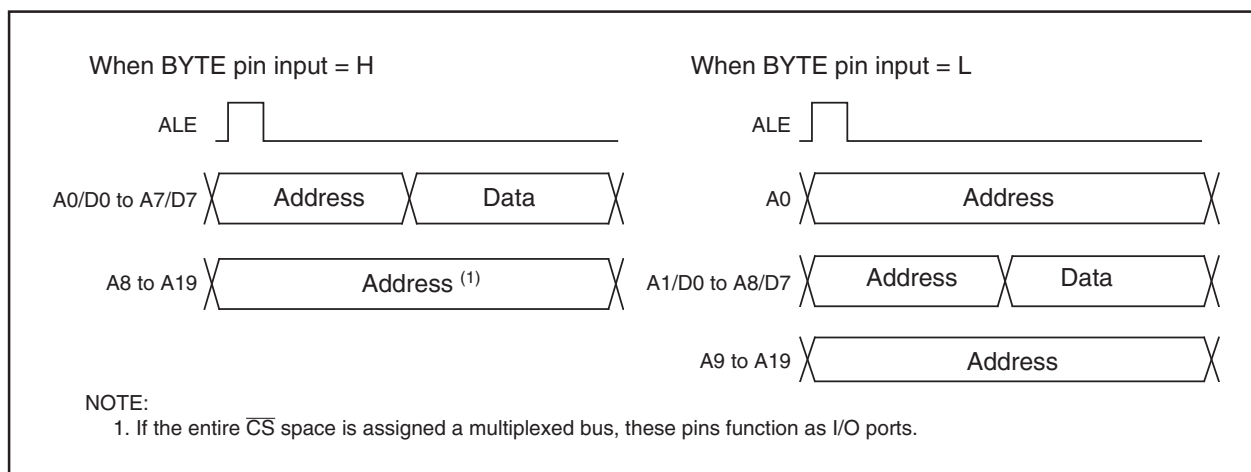
Data Bus Width	$\overline{RD}$	$\overline{WRL}$	$\overline{WRH}$	Status of External Data Bus
16 Bits (BYTE pin input = L)	L	H	H	Read data
	H	L	H	Write 1 byte of data to an even address
	H	H	L	Write 1 byte of data to an odd address
	H	L	L	Write data to both even and odd addresses

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

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

### 7.2.5 ALE Signal

The ALE signal latches the address when accessing the multiplexed bus space. Latch the address when the ALE signal falls. Figure 7.3 shows the ALE signal, address bus and data bus.



**Figure 7.3 ALE Signal, Address Bus, Data Bus**

### 7.2.6 $\overline{\text{RDY}}$ Signal

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

A0 to A19, D0 to D15,  $\overline{\text{CS0}}$  to  $\overline{\text{CS3}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{WRL}}$ ,  $\overline{\text{WRH}}$ ,  $\overline{\text{WR}}$ ,  $\overline{\text{BHE}}$ ,  $\overline{\text{ALE}}$ ,  $\overline{\text{HLDA}}$

Then, when the input on the  $\overline{\text{RDY}}$  pin is detected high at the falling edge of BCLK, the remaining bus cycle is executed. Figure 7.4 shows example in which the wait state was inserted into the read cycle by the  $\overline{\text{RDY}}$  signal. To use the  $\overline{\text{RDY}}$  signal, set the corresponding bit ( $\text{CS3W}$  to  $\text{CS0W}$  bits) in the CSR register to "0" (with wait state). When not using the  $\overline{\text{RDY}}$  signal, the  $\overline{\text{RDY}}$  pin must be pulled-up.

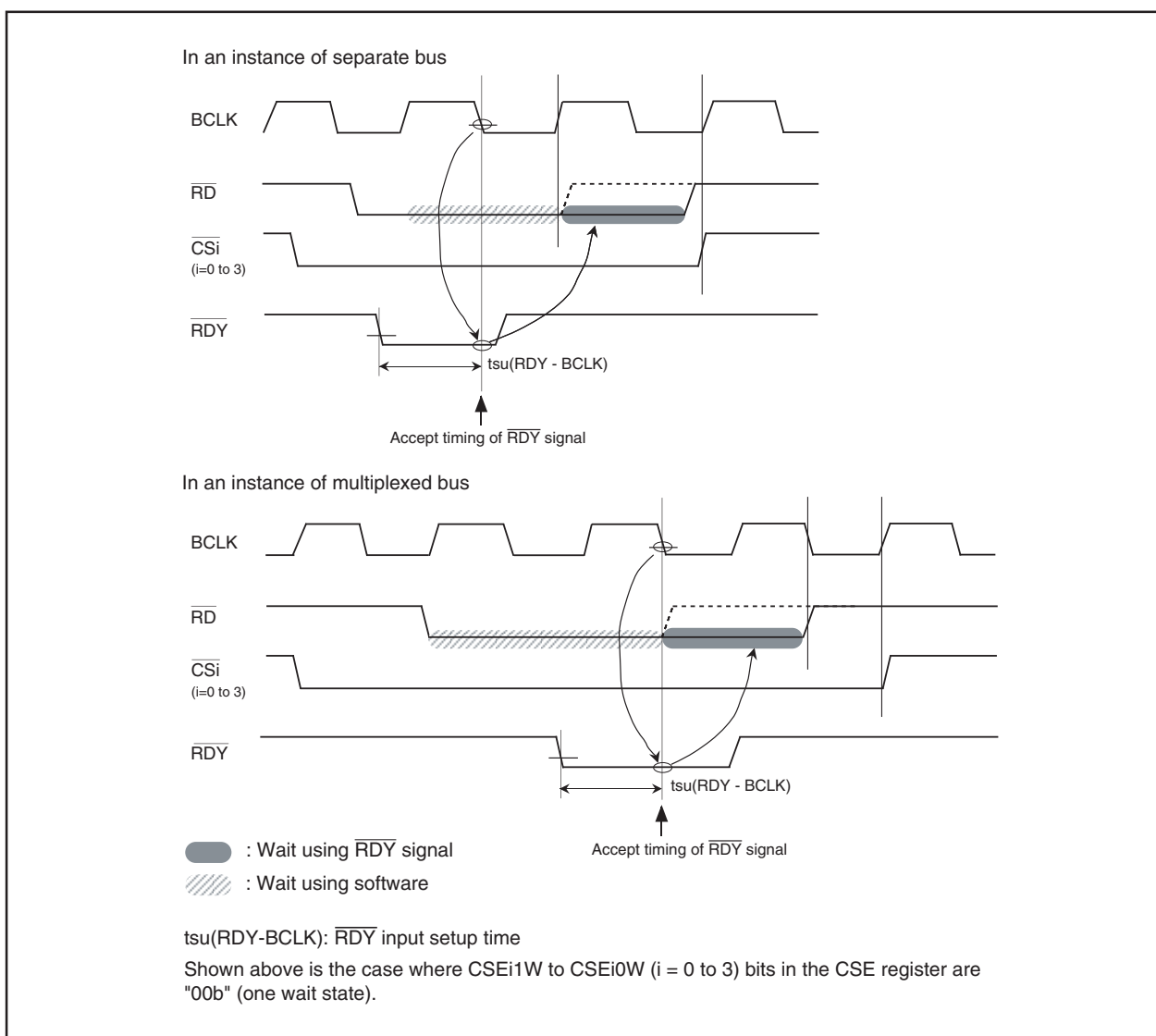


Figure 7.4 Example in which Wait State was Inserted into Read Cycle by  $\overline{\text{RDY}}$  Signal

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

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

Table 7.5 shows the microcomputer status in the hold state.

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

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

**Figure 7.5 Bus-using Priorities**

**Table 7.5 Microcomputer Status in Hold State**

Item		Status
BCLK		Output
A0 to A19, D0 to D15, CS0 to CS3, RD, WRL, WRH, WR, BHE		High-impedance
I/O Ports	P0, P1, P3, P4 <sup>(1)</sup>	High-impedance
	P6 to P10	Maintains status when hold signal is received
HLDA		Output "L"
Internal Peripheral Circuits		ON (but watchdog timer stops <sup>(2)</sup> )
ALE Signal		Undefined

**NOTES:**

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

### 7.2.8 BCLK Output

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

Table 7.6 shows the pin functions for each processor mode.

Table 7.6 Pin Functions for Each Processor Mode

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

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

## NOTES:

1. For setting the PM01 to PM00 bits to "01b" (memory expansion mode) and the PM05 to PM04 bits to "11b" (multiplexed bus assigned to the entire CS space), apply "H" to the BYTE pin (external data bus is an 8-bit width). While the CNVSS pin is held "H" (VCC), do not rewrite the PM05 to PM04 bits to "11b" after reset. If the PM05 to PM04 bits are set to "11b" during memory expansion mode, P3\_1 to P3\_7 and P4\_0 to P4\_3 become I/O ports, in which case the accessible area for each CS is 256 bytes.
2. In separate bus mode, these pins serve as the address bus.
3. If the data bus is 8-bit width, make sure the PM02 bit is set to "0" (RD, BHE, WR).
4. When accessing the area that uses a multiplexed bus, these pins output an indeterminate value during a write.



## 7.2.9 External Bus Status When Internal Area Accessed

Table 7.7 shows the external bus status when the internal area is accessed.

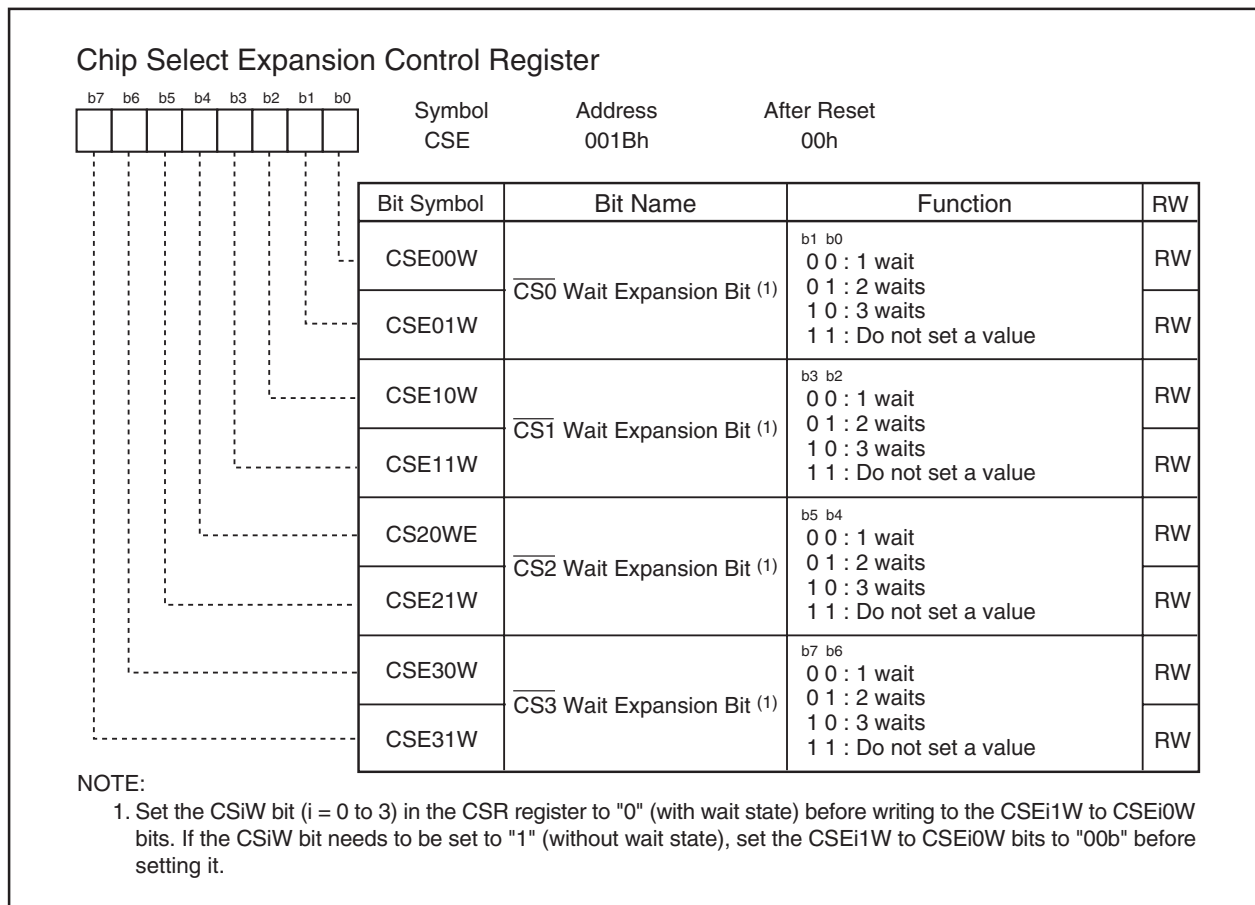
**Table 7.7 External Bus Status When Internal Area Accessed**

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

## 7.2.10 Software Wait

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

To use the  $\overline{\text{RDY}}$  signal, set the corresponding CS3W to CS0W bit to "0" (with wait state). Figure 7.6 shows the CSE register. Table 7.8 shows the software wait related bits and bus cycles. Figures 7.7 and 7.8 show the typical bus timings using software wait.



**Figure 7.6 CSE Register**

**Table 7.8 Software Wait Related Bits and Bus Cycles**

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

**NOTES:**

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

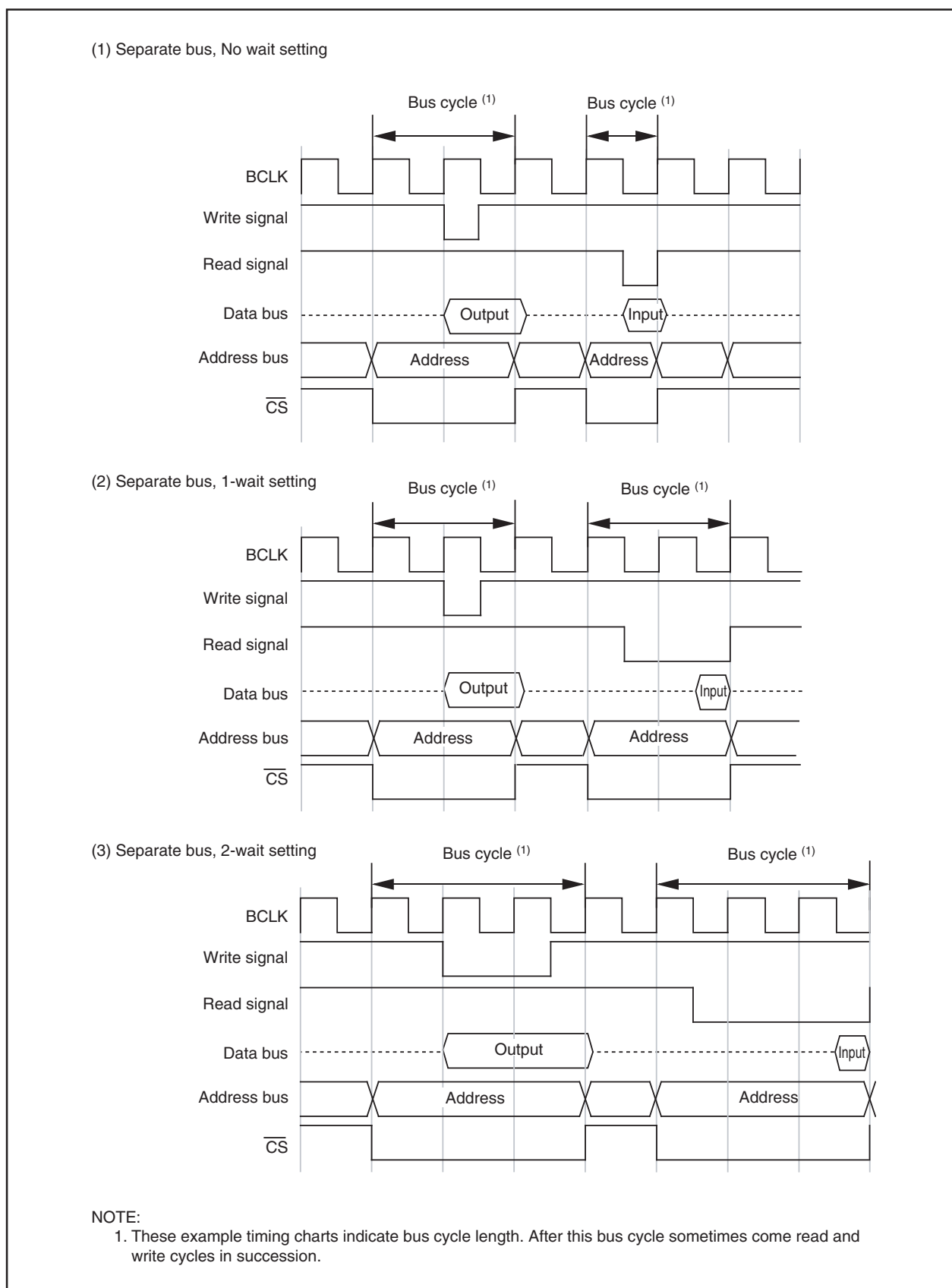
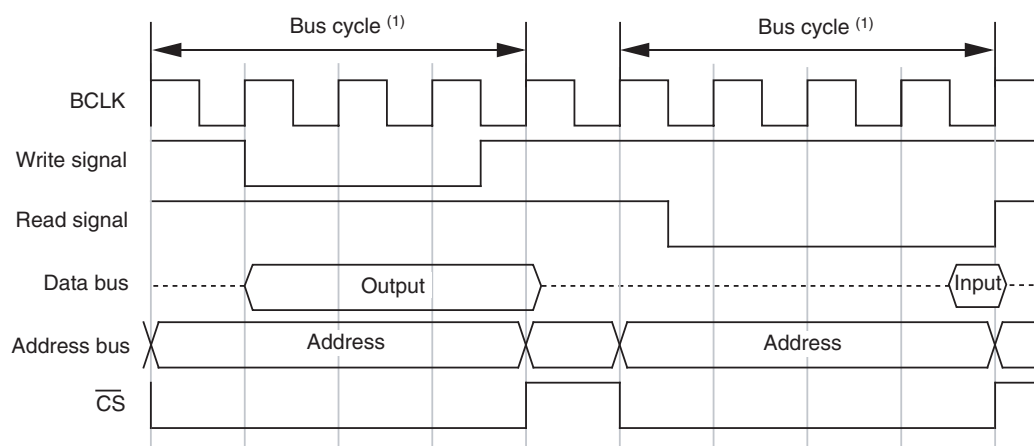
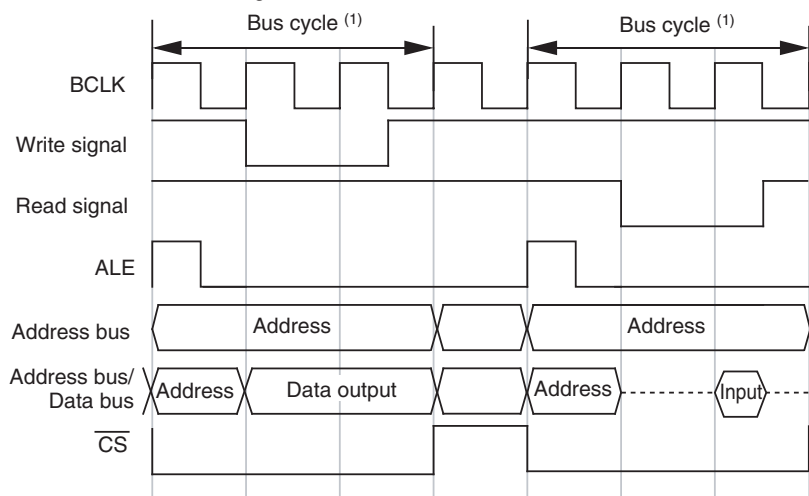


Figure 7.7 Typical Bus Timings Using Software Wait (1)

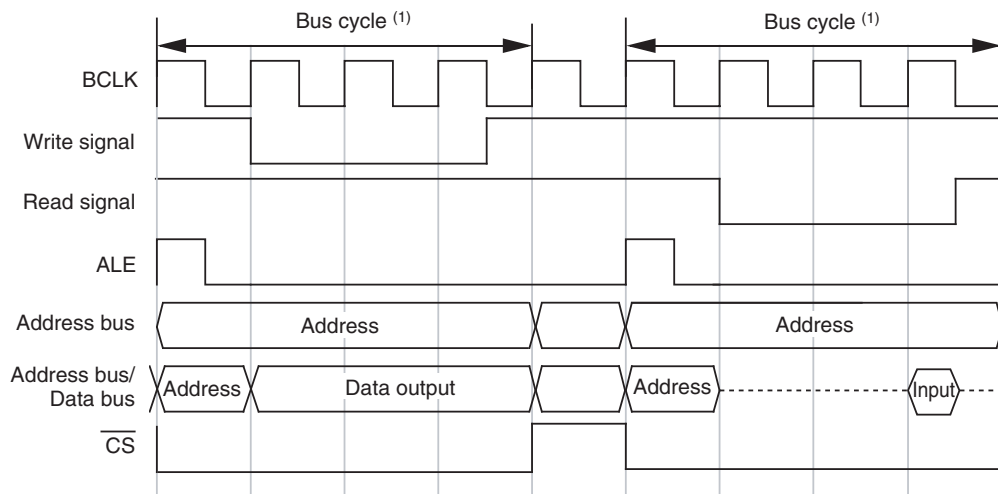
(1) Separate bus, 3-wait setting



(2) Multiplexed bus, 1- or 2-wait setting



(3) Multiplexed bus, 3-wait setting



NOTE:

1. These example timing charts indicate bus cycle length. After this bus cycle sometimes come read and write cycles in succession.

Figure 7.8 Typical Bus Timings Using Software Wait (2)

## 8. Clock Generating Circuit

### 8.1 Types of Clock Generating Circuit

Four circuits are incorporated to generate the system clock signal:

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

Table 8.1 lists the clock generating circuit specifications. Figure 8.1 shows the clock generating circuit. Figures 8.2 to 8.8 show the clock-related registers.

**Table 8.1 Clock Generating 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>• Clock source of Timer A, B</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 16 MHz	32.768 kHz	About 1 MHz	16 MHz, 20 MHz, 24 MHz <sup>(1)</sup>
Usable Oscillator	<ul style="list-style-type: none"> <li>• Ceramic oscillator</li> <li>• Crystal oscillator</li> </ul>	• Crystal oscillator	-	-
Pins to Connect Oscillator	XIN, XOUT	XCIN, XCOU	-	-
Oscillation Stop and Re-Oscillation Detection Function	Available	Available	Available	Available
Oscillation Status After Reset	Oscillating	Stopped	Stopped	Stopped
Other	Externally derived clock can be input		-	-

NOTE:

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



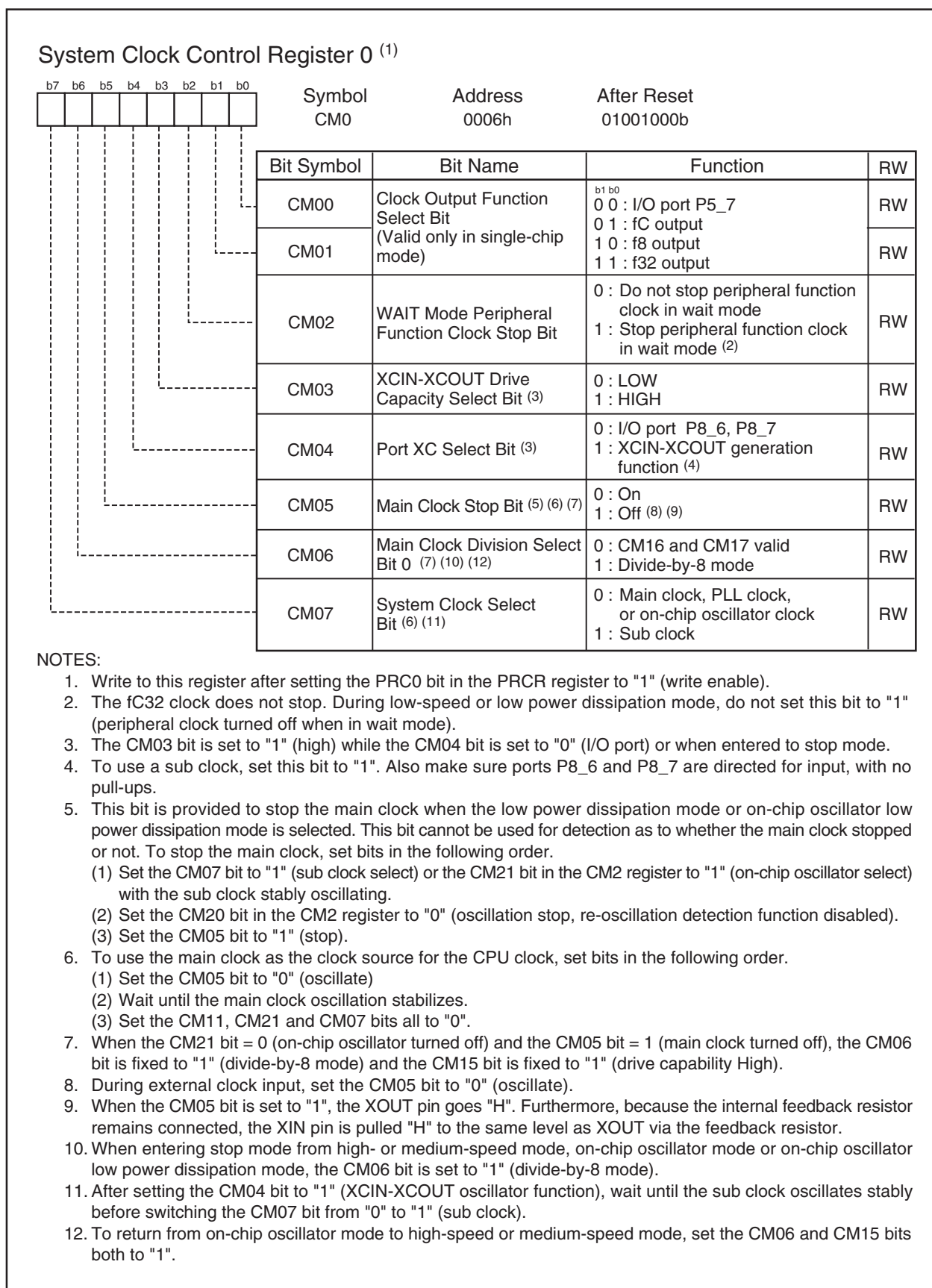
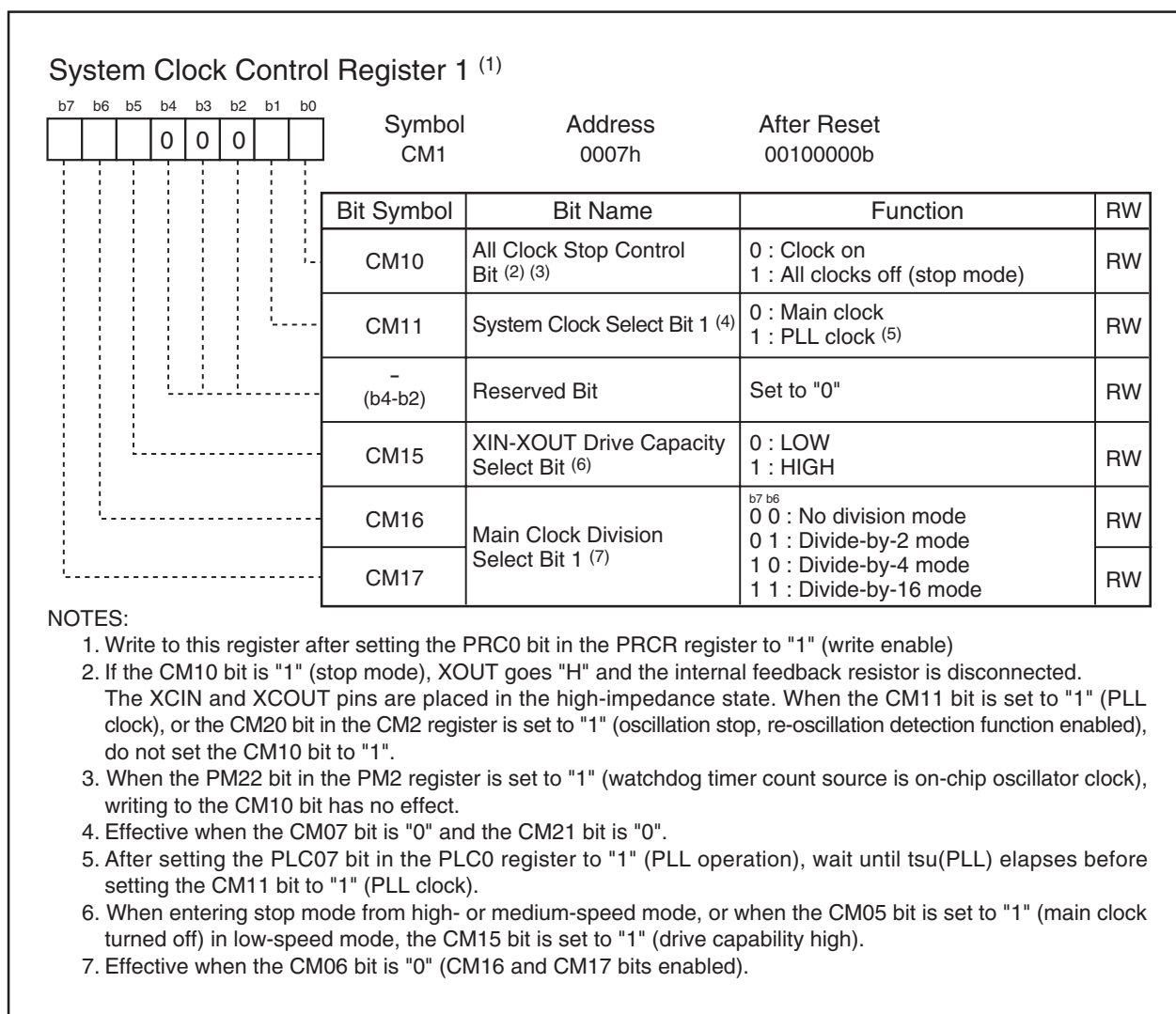
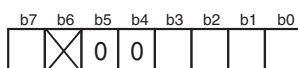


Figure 8.2 CM0 Register

**Figure 8.3 CM1 Register**



Oscillation Stop Detection Register <sup>(1)</sup>

Symbol  
CM2

Address  
000Ch

After Reset  
0X00000b <sup>(2)</sup>

Bit Symbol	Bit Name	Function	RW
CM20	Oscillation Stop, Re-Oscillation Detection Enable Bit <sup>(2) (3) (4)</sup>	0 : Oscillation stop, re-oscillation detection function disabled 1 : Oscillation stop, re-oscillation detection function enabled	RW
CM21	System Clock Select Bit <sup>2 (2) (5) (6) (7) (8) (11)</sup>	0 : Main clock or PLL clock 1 : On-chip oscillator clock (On-chip oscillator oscillating)	RW
CM22	Oscillation Stop, Re-Oscillation Detection Flag <sup>(9)</sup>	0 : Main clock stop, re-oscillation not detected 1 : Main clock stop, re-oscillation detected	RW
CM23	XIN Monitor Flag <sup>(10)</sup>	0 : Main clock oscillating 1 : Main clock turned off	RO
– (b5-b4)	Reserved Bit	Set to "0"	RW
– (b6)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–
CM27	Operation Select Bit (behavior if oscillation stop, re-oscillation is detected) <sup>(2)</sup>	0 : Oscillation stop detection reset 1 : Oscillation stop, re-oscillation detection interrupt	RW

## NOTES:

- Write to this register after setting the PRC0 bit in the PRCR register to "1" (write enable).
- The CM20, CM21 and CM27 bits do not change at oscillation stop detection reset.
- Set the CM20 bit to "0" (disable) before entering stop mode. After exiting stop mode, set the CM20 bit back to "1" (enable).
- Set the CM20 bit to "0" (disable) before setting the CM05 bit in the CM0 register.
- When the CM20 bit is "1" (oscillation stop, re-oscillation detection function enabled), the CM27 bit is "1" (oscillation stop, re-oscillation detection interrupt), and the CPU clock source is the main clock, the CM21 bit is set to "1" (on-chip oscillator clock) if the main clock stop is detected.
- If the CM20 bit is "1" and the CM23 bit is "1" (main clock turned off), do not set the CM21 bit to "0".
- Effective when the CM07 bit in the CM0 register is "0".
- Where the CM20 bit is "1" (oscillation stop, re-oscillation detection function enabled), the CM27 bit is "1" (oscillation stop, re-oscillation detection interrupt), and the CM11 bit is "1" (the CPU clock source is PLL clock), the CM21 bit remains unchanged even when main clock stop is detected. If the CM22 bit is "0" under these conditions, an oscillation stop, re-oscillation detection interrupt request is generated at main clock stop detection; it is, therefore, necessary to set the CM21 bit to "1" (on-chip oscillator clock) inside the interrupt routine.
- This bit is set to "1" when the main clock is detected to have stopped and when the main clock is detected to have restarted oscillating. When this bit changes state from "0" to "1", an oscillation stop and re-oscillation detection interrupt request is generated. Use this bit in an interrupt routine to discriminate the causes of interrupts between the oscillation stop and re-oscillation detection interrupt and the watchdog timer interrupt. This bit is set to "0" by writing "0" in a program. (Writing "1" has no effect. Nor is it set to "0" by an oscillation stop and re-oscillation detection interrupt request acknowledged.)  
If an oscillation stop or a re-oscillation is detected when the CM22 bit = 1, no oscillation stop and re-oscillation detection interrupt requests are generated.
- Read the CM23 bit in an oscillation stop and re-oscillation detection interrupt handling routine to determine the main clock status.
- When the CM21 bit = 0 (on-chip oscillator turned off) and the CM05 bit = 1 (main clock turned off), the CM06 bit is fixed to "1" (divide-by-8 mode) and the CM15 bit is fixed to "1" (drive capability High).

Figure 8.4 CM2 Register

Peripheral Clock Select Register <sup>(1)</sup>

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0		

Symbol  
PCLKR

Address  
025Eh

After Reset  
00h

Bit Symbol	Bit Name	Function	RW
PCLK0	Timers A, B, and A/D Clock Select Bit (Clock source for the timers A, B, the dead time timer and A/D)	0 : Divide-by-2 of fAD, f2 1 : fAD, f1	RW
PCLK1	SI/O Clock Select Bit (Clock source for UART0 to UART2, SI/O3)	0 : f2SIO 1 : f1SIO	RW
(b7-b2)	Reserved Bit	Set to "0"	RW

## NOTE:

1. Write to this register after setting the PRC0 bit in the PRCR register to "1" (write enable).

Figure 8.5 PCLKR Register

CAN0/1 Clock Select Register <sup>(1)</sup>

b7	b6	b5	b4	b3	b2	b1	b0

Symbol  
CCLKR

Address  
025Fh

After Reset  
00h

Bit Symbol	Bit Name	Function	RW
CCLK0	CAN0 Clock Select Bits <sup>(2)</sup>	b2 b1 b0 0 0 0 : No division 0 0 1 : Divide-by-2 0 1 0 : Divide-by-4 0 1 1 : Divide-by-8 1 0 0 : Divide-by-16	RW
CCLK1			RW
CCLK2		1 0 1 : 1 1 0 : } Do not set a value 1 1 1 :	RW
CCLK3	CAN0 CPU Interface Sleep Bit <sup>(3)</sup>	0: CAN0 CPU interface operating 1: CAN0 CPU interface in sleep	RW
CCLK4	CAN1 Clock Select Bits <sup>(2)</sup>	b6 b5 b4 0 0 0 : No division 0 0 1 : Divide-by-2 0 1 0 : Divide-by-4 0 1 1 : Divide-by-8 1 0 0 : Divide-by-16	RW
CCLK5			RW
CCLK6		1 0 1 : 1 1 0 : } Do not set a value 1 1 1 :	RW
CCLK7	CAN1 CPU Interface Sleep Bit <sup>(3)</sup>	0: CAN1 CPU interface operating 1: CAN1 CPU interface in sleep	RW

## NOTES:

1. Write to this register after setting the PRC0 bit in the PRCR register to "1" (Write enabled).
2. Set only when the Reset bit in the CiCTLR register (i = 0, 1) = 1 (Reset/Initialization mode).
3. Before setting this bit to "1", set the Sleep bit in the CiCTLR to "1" (Sleep mode enabled).

Figure 8.6 CCLKR Register

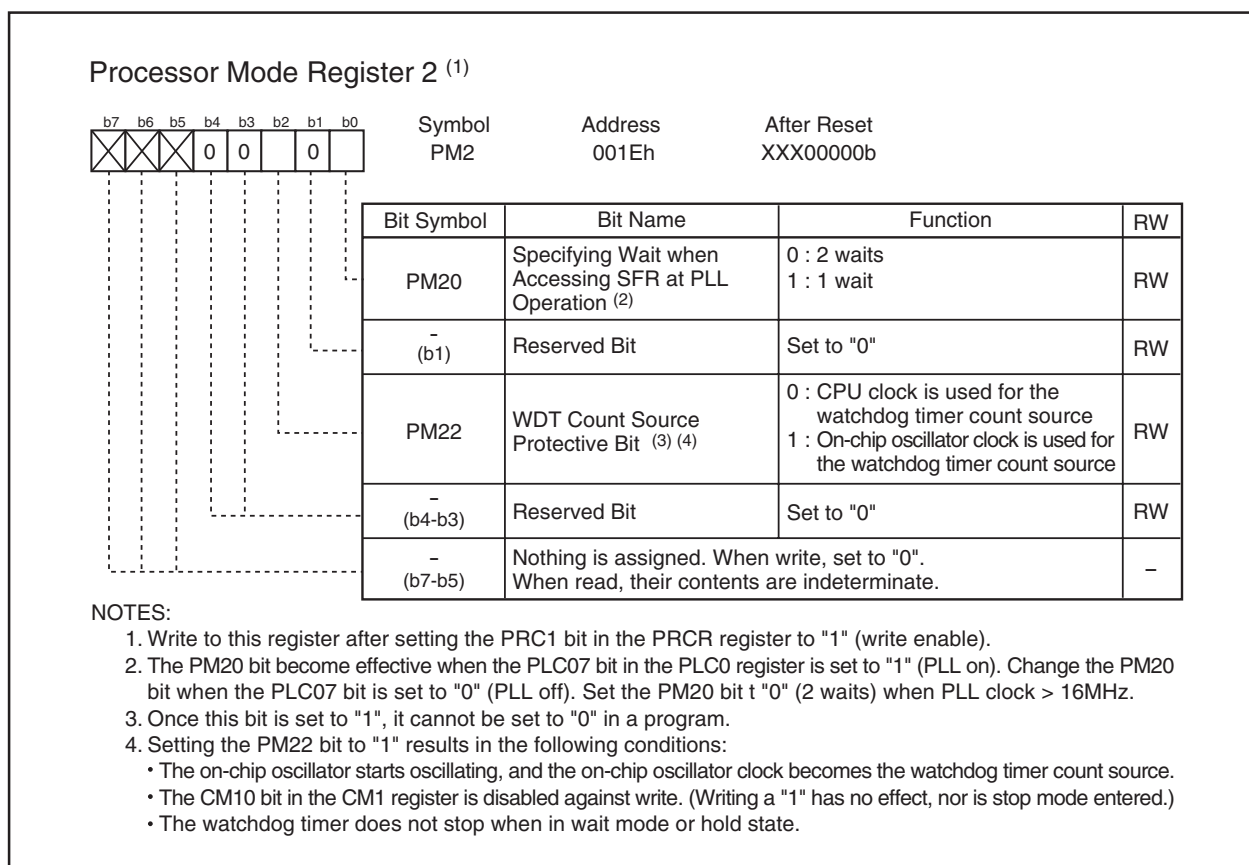


Figure 8.7 PM2 Register

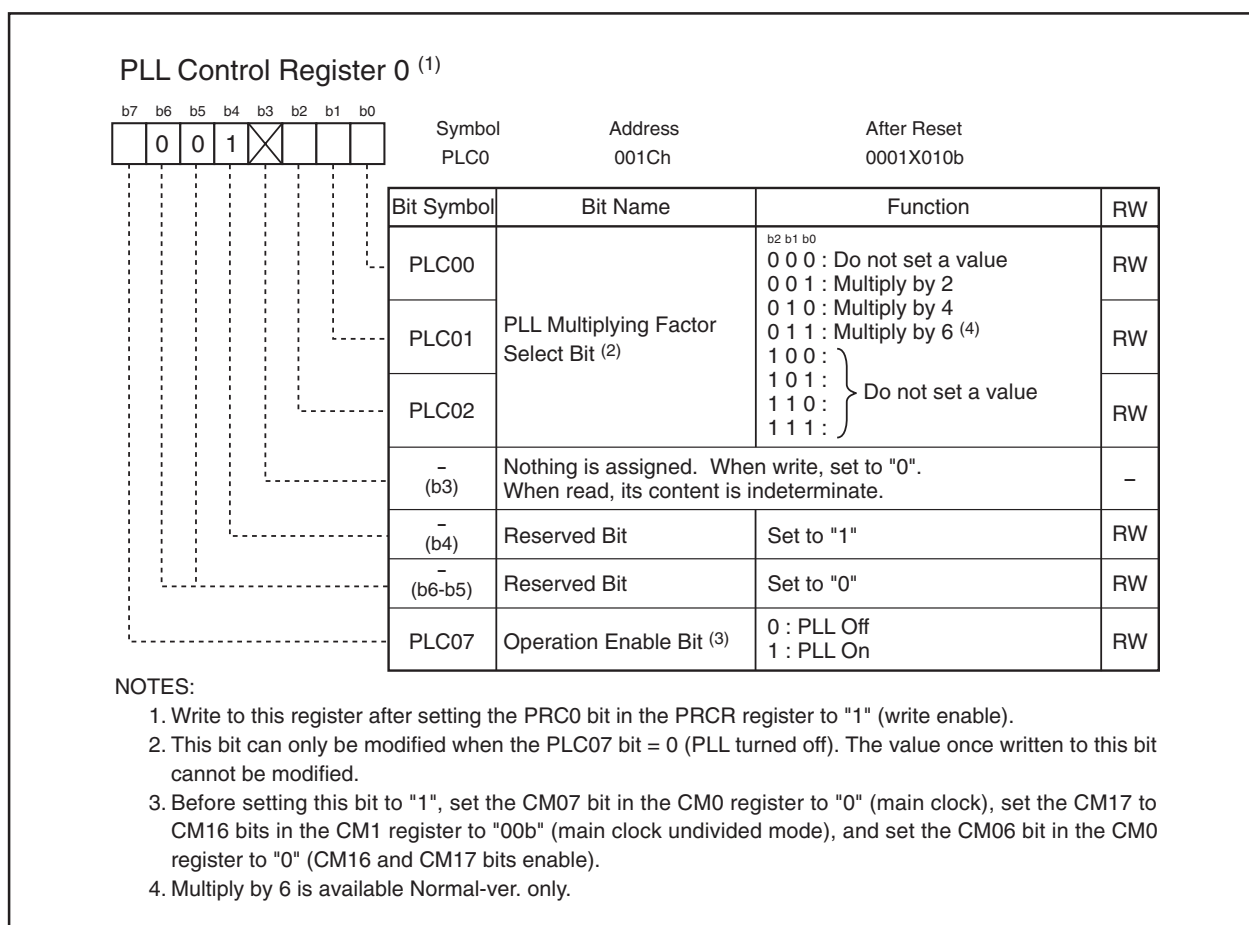


Figure 8.8 PLC0 Register

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

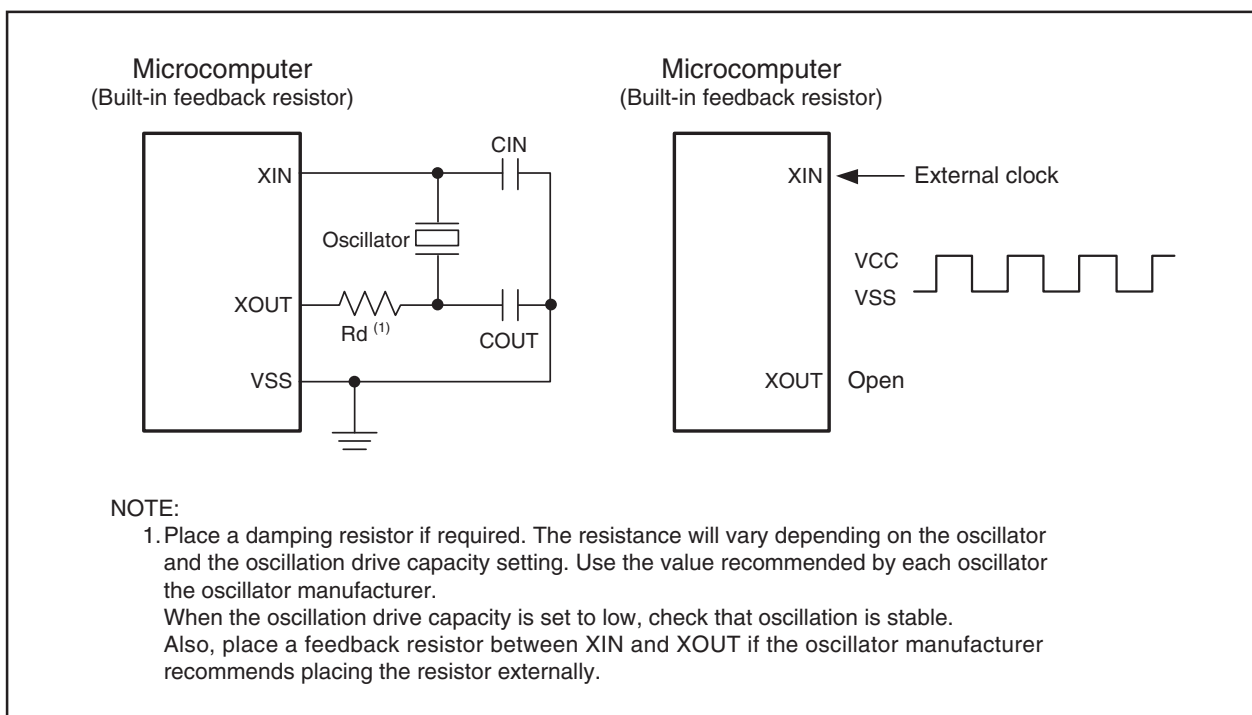
### 8.1.1 Main Clock

The main clock is generated by the main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock 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 8.9 shows the examples of main clock connection circuit.

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

The power consumption in the chip can be reduced by setting the CM05 bit in the CM0 register to “1” (main clock 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. Note, that if an externally generated clock is fed into the XIN pin, the main clock cannot be turned off by setting the CM05 bit to “1” unless the sub clock is selected as a CPU clock. If necessary, use an external circuit to turn off the clock.

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



**Figure 8.9 Examples of Main Clock Connection Circuit**

### 8.1.2 Sub Clock

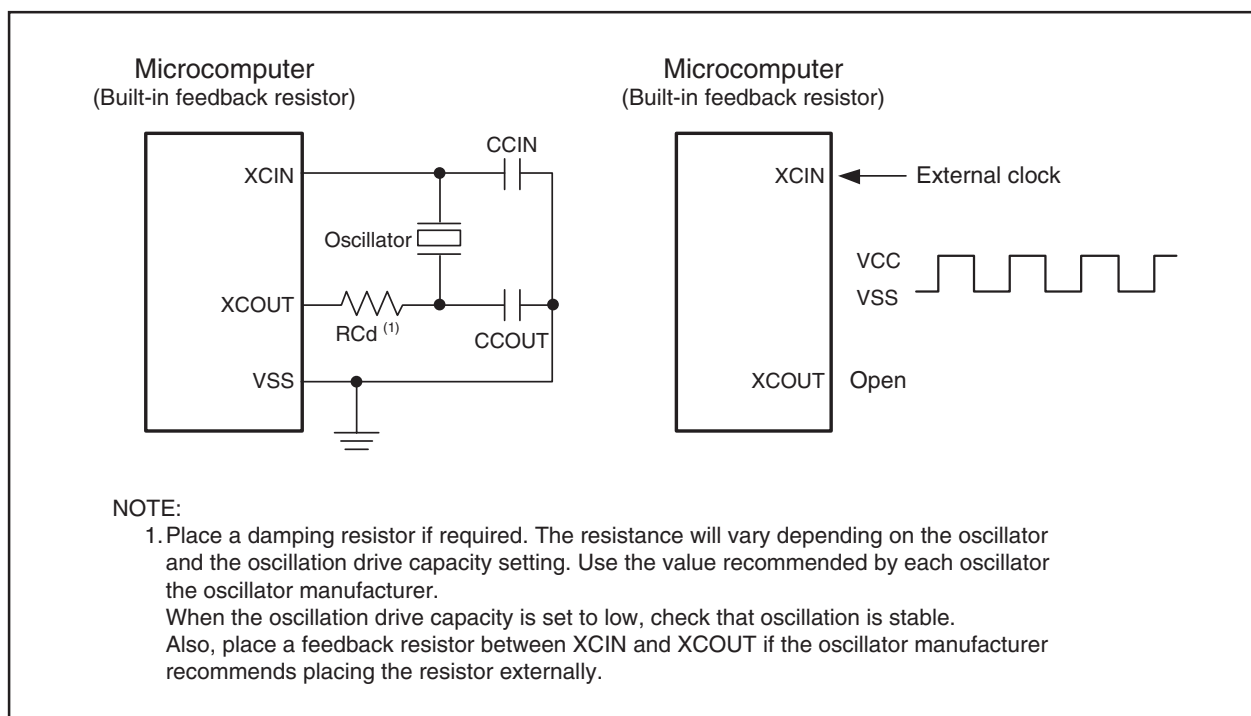
The sub clock is generated by the sub clock oscillation circuit. This clock is used as the clock source for the CPU clock, as well as the timer A and timer B count sources. In addition, an fC clock with the same frequency as that of the sub clock can be output from the CLKOUT pin.

The sub clock oscillator circuit is configured by connecting a crystal resonator between the XCIN and XCOU 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 8.10 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 **8.4 Power Control**.



**Figure 8.10 Examples of Sub Clock Connection Circuit**

### 8.1.3 On-chip Oscillator Clock

This clock, approximately 1 MHz, is supplied by a on-chip oscillator. This clock is used as the clock source for the CPU and peripheral function clocks. In addition, if the PM22 bit in the PM2 register is “1” (on-chip oscillator clock for the watchdog timer count source), this clock is used as the count source for the watchdog timer (refer to **11.1 Count Source Protective Mode**).

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

### 8.1.4 PLL Clock

The PLL clock is generated 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 a fixed period of tsu(PLL) for the PLL clock to be stable, and then set the CM11 bit in the CM1 register to “1”.

Before entering wait mode or stop mode, be sure to set the CM11 bit to “0” (CPU clock source is the main clock). Furthermore, before entering stop mode, be sure to set the PLC07 bit in the PLC0 register to “0” (PLL stops). Figure 8.11 shows the procedure for using the PLL clock as the clock source for the CPU. The PLL clock frequency is determined by the equation below. When the PLL clock frequency is 16 MHz or more, set the PM20 bit in the PM2 register to “0” (2 waits).

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

(However, PLL clock frequency = 16 MHz, 20 MHz or 24 MHz <sup>(1)</sup> )

#### NOTE:

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

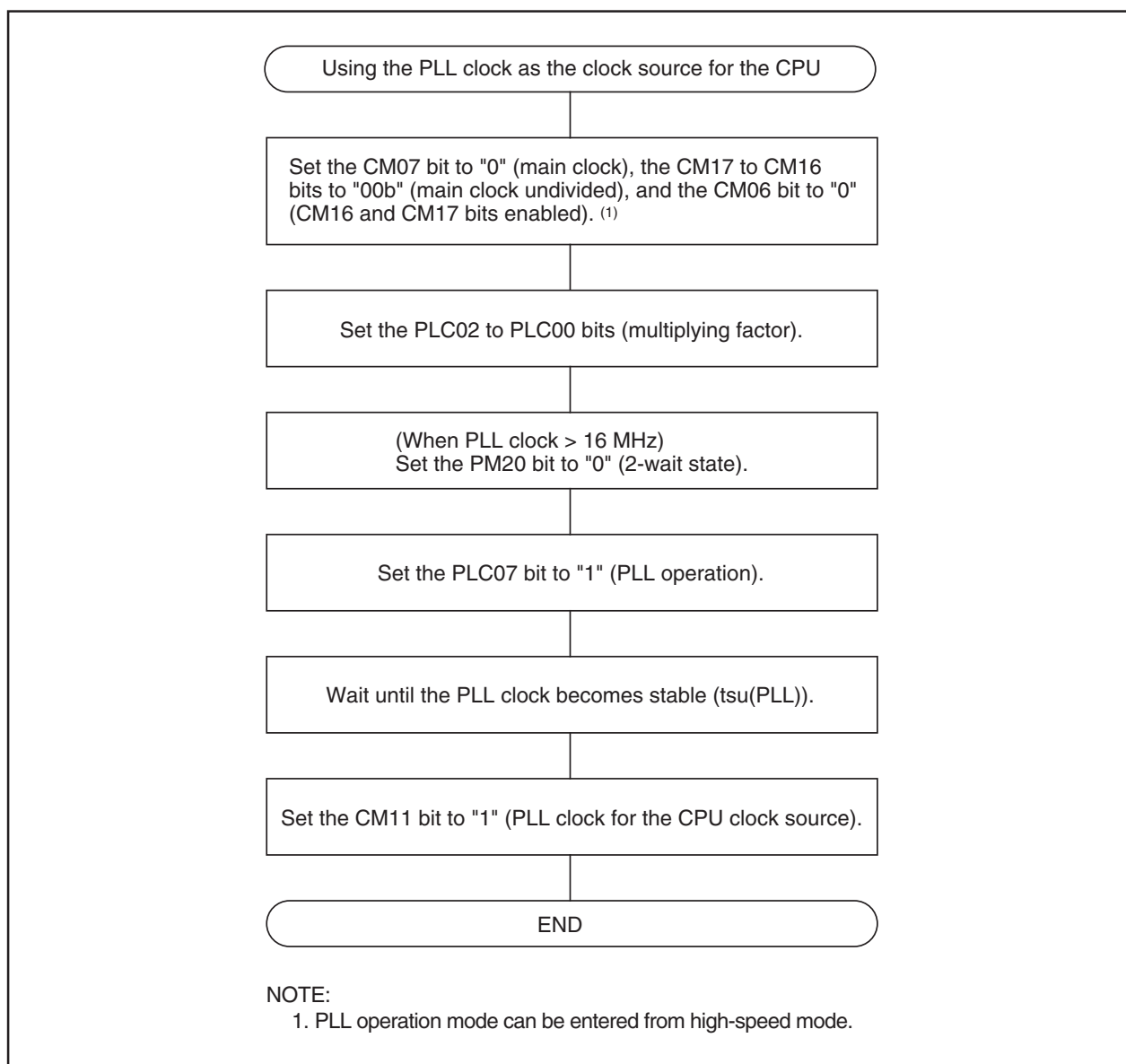
The PLC02 to PLC00 bits can be set only once after reset. Table 8.2 shows the example for setting PLL clock frequencies.

**Table 8.2 Example for Setting PLL Clock Frequencies**

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

#### NOTES:

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

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

## 8.2 CPU Clock and Peripheral Function Clock

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

### 8.2.1 CPU Clock and BCLK

These are operating clocks for the CPU and watchdog timer.

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

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

When the PLL clock is selected as the clock source for the CPU clock, the CM06 bit should be set to "0" and the CM17 to CM16 bits to "00b" (undivided).

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

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

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

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

These are operating clocks for the peripheral functions.

Two of these, fi (i = 1, 2, 8, 32) and fiSIO are derived from the main clock, PLL clock or on-chip oscillator clock by dividing them by i. The clock fi is used for timers A and B, and fiSIO is used for serial interface. The f8 and f32 clocks can be output from the CLKOUT pin.

The fAD clock is produced from the main clock, PLL clock or on-chip oscillator clock, and is used for the A/D converter.

The fCANi (i = 0, 1) clock is derived from the main clock, PLL clock or on-chip oscillator clock by dividing them by 1 (undivided), 2, 4, 8 or 16, and is used for the CAN module.

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to "1" (peripheral function clock turned off during wait mode), or when the microcomputer is in low power dissipation mode, the fi, fiSIO, fAD, fCAN0 and fCAN1 clocks are turned off <sup>(1)</sup>.

The fC32 clock is derived from the sub clock, and is used for timers A and B. This clock can be used when the sub clock is activated.

#### NOTE:

1. fCAN0 and fCAN1 clocks stop at "H" in CAN0, 1 sleep mode.

## 8.3 Clock Output Function

During single-chip mode, the f8, f32 or fC clock can be output from the CLKOUT pin. Use the CM01 to CM00 bits in the CM0 register to select.



## 8.4 Power Control

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

### 8.4.1 Normal Operation Mode

Normal operation mode is further classified into seven sub 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 speed or low power dissipation mode to on-chip oscillator or on-chip oscillator low power dissipation mode. Nor can operation modes be changed directly from on-chip oscillator or on-chip oscillator low power dissipation mode to low-speed or low power dissipation mode. Where the CPU clock source is changed from the on-chip oscillator to the main clock, change the operation mode to the medium-speed mode (divide-by-8 mode) after the clock was divided by 8 (the CM06 bit in the CM0 register was set to "1") in the on-chip oscillator mode.

#### 8.4.1.1 High-speed Mode

The main clock divided by 1 provides the CPU clock. If the sub clock is activated, fC32 can be used as the count source for timers A and B.

#### 8.4.1.2 PLL Operation Mode

The main clock multiplied by 2, 4 or 6 <sup>(1)</sup> provides the PLL clock, and this PLL clock serves as the CPU clock. If the sub clock is activated, 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.

NOTE:

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

#### 8.4.1.3 Medium-speed Mode

The main clock divided by 2, 4, 8 or 16 provides the CPU clock. If the sub clock is activated, fC32 can be used as the count source for timers A and B.

#### 8.4.1.4 Low-speed Mode

The sub clock provides the CPU clock. The main clock is used as the clock source for the peripheral function clock when the CM21 bit in the CM2 register is set to "0" (on-chip oscillator turned off), and the on-chip oscillator clock is used when the CM21 bit is set to "1" (on-chip oscillator oscillating).

The fC32 clock can be used as the count source for timers A and B.

#### 8.4.1.5 Low Power Dissipation Mode

In this mode, the main clock is turned off after being placed in low speed mode. The sub clock provides the CPU clock. The fC32 clock can be used as the count source for timers A and B.

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

**8.4.1.6 On-chip Oscillator Mode**

The on-chip oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The on-chip oscillator clock is also the clock source for the peripheral function clocks. If the sub clock is activated, fC32 can be used as the count source for timers A and B. When the operation mode is returned to the high- and medium-speed modes, set the CM06 bit in the CM0 register to “1” (divide-by-8 mode).

**8.4.1.7 On-chip Oscillator Low Power Dissipation Mode**

The main clock is turned off after being placed in on-chip oscillator mode. The CPU clock can be selected like 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 activated, fC32 can be used as the count source for timers A and B.

Table 8.3 lists the setting clock related bit and modes.

**Table 8.3 Setting Clock Related Bit and Modes**

Modes		CM2 Register	CM1 Register		CM0 Register			
		CM21	CM11	CM17, CM16	CM07	CM06	CM05	CM04
PLL Operation Mode		0	1	00b	0	0	0	-
High-Speed Mode		0	0	00b	0	0	0	-
Medium-Speed Mode	divided by 2	0	0	01b	0	0	0	-
	divided by 4	0	0	10b	0	0	0	-
	divided by 8	0	0	-	0	1	0	-
	divided by 16	0	0	11b	0	0	0	-
Low-Speed Mode		-	0	-	1	-	0	1
Low Power Dissipation Mode		0	0	-	1	1 <sup>(1)</sup>	1 <sup>(1)</sup>	1
On-chip Oscillator Mode	divided by 1	1	0	00b	0	0	0	-
	divided by 2	1	0	01b	0	0	0	-
	divided by 4	1	0	10b	0	0	0	-
	divided by 8	1	0	-	0	1	0	-
	divided by 16	1	0	11b	0	0	0	-
On-chip Oscillator Low power Dissipation Mode		1	0	(NOTE 2)	0	(NOTE 2)	1	-

-. “0” or “1”

**NOTES:**

1. When the CM05 bit is set to “1” (main clock turned off) in low-speed mode, the mode goes to low power dissipation mode and the CM06 bit is set to “1” (divide-by-8 mode) simultaneously.
2. The divide-by-n value can be selected the same way as in on-chip oscillator mode.

## 8.4.2 Wait Mode

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

### 8.4.2.1 Peripheral Function Clock Stop Function

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

### 8.4.2.2 Entering Wait Mode

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

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

### 8.4.2.3 Pin Status During Wait Mode

Table 8.4 lists the pin status during wait mode.

**Table 8.4 Pin Status During Wait Mode**

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

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

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

Table 8.5 lists the interrupts to exit wait mode and use conditions.

**Table 8.5 Interrupts to Exit Wait Mode and Use Conditions**

Interrupt	CM02 Bit = 0	CM02 Bit = 1
NMI Interrupt	Can be used	Can be used
Serial Interface Interrupt	Can be used when operating with internal or external clock	Can be used when operating with external clock
Key Input Interrupt	Can be used	Can be used
A/D Conversion Interrupt	Can be used in one-shot mode or single sweep mode	- (Do not use)
Timer A Interrupt Timer B interrupt	Can be used in all modes	Can be used in event counter mode or when the count source is fc32
INT Interrupt	Can be used	Can be used
CAN0/1 Wake-up Interrupt	Can be used in CAN sleep mode	Can be used in CAN sleep mode

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

- (1) Set the ILVL2 to ILVL0 bits in the interrupt control register, for peripheral function interrupts used to exit wait mode.

The ILVL2 to ILVL0 bits in all other interrupt control registers, for peripheral function interrupts not used to exit wait mode, are set to "000b" (interrupt disable).

- (2) Set the I flag to "1".

- (3) Start operating the peripheral functions used to exit wait mode.

When the peripheral function interrupt is used, an interrupt routine is performed as soon as an interrupt request is acknowledged and the CPU clock is supplied again.

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

### 8.4.3 Stop Mode

In stop mode, all oscillator circuits are turned off, so are the CPU clock and the peripheral function clocks. Therefore, the CPU and the peripheral functions clocked by these clocks stop operating. The least amount of power is consumed in this mode. If the voltage applied to VCC is VRAM or more, the internal RAM is retained.

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

Table 8.6 lists the interrupts to stop mode and use conditions.

**Table 8.6 Interrupts to Stop Mode and Use Conditions**

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

#### 8.4.3.1 Entering Stop Mode

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

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

Also, if the CM11 bit in the CM1 register is “1” (PLL clock for the CPU clock source), set the CM11 bit to “0” (main clock for the CPU clock source) and the PLC07 bit in the PLC0 register to “0” (PLL turned off) before entering stop mode.

#### 8.4.3.2 Pin Status in Stop Mode

Table 8.7 lists the pin status in stop mode.

**Table 8.7 Pin Status in Stop Mode**

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

#### 8.4.3.3 Exiting Stop Mode

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

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

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

(1) The ILVL2 to ILVL0 bits in the interrupt control registers, for the peripheral function interrupt used to exit stop mode, must have larger value than that of the RLVL2 to RLVL0 bits.

The ILVL2 to ILVL0 bits in all other interrupt control registers, for the peripheral function interrupts which are not used to exit stop mode, must be set to "000b" (interrupt disabled).

(2) Set the I flag to "1".

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

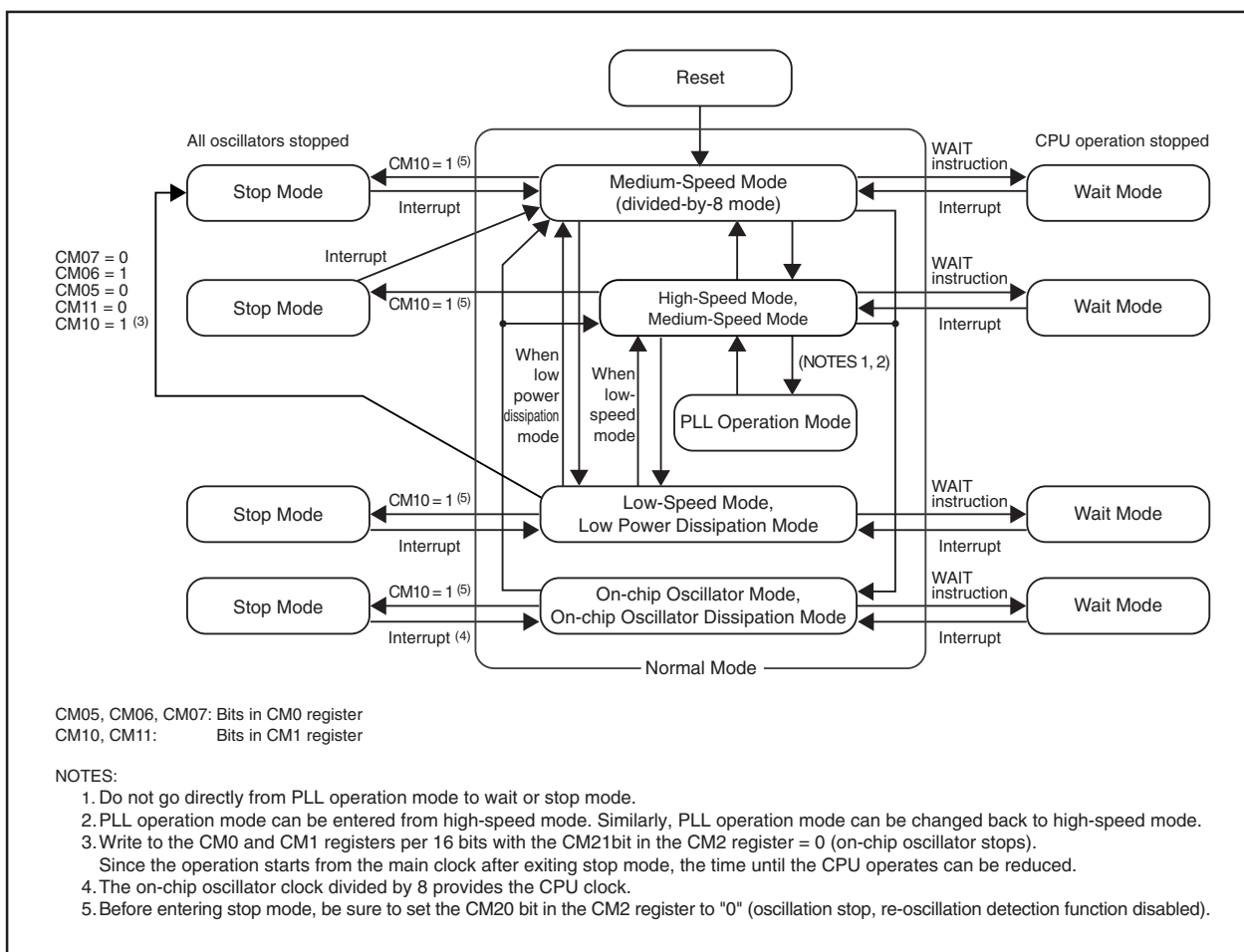
When exiting stop mode by the peripheral function interrupt, the interrupt routine is performed when an interrupt request is generated and the CPU clock is supplied again.

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

- When the sub clock is the CPU clock before entering stop mode: Sub clock
- When the main clock is the CPU clock source before entering stop mode: Main clock divided by 8
- When the on-chip oscillator clock is the CPU clock source before entering stop mode: On-chip oscillator clock divided by 8

Figure 8.12 shows the state transition from normal operation mode to stop mode and wait mode. Figure 8.13 shows the state transition in normal operation mode.

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



**Figure 8.12 State Transition to Stop Mode and Wait Mode**

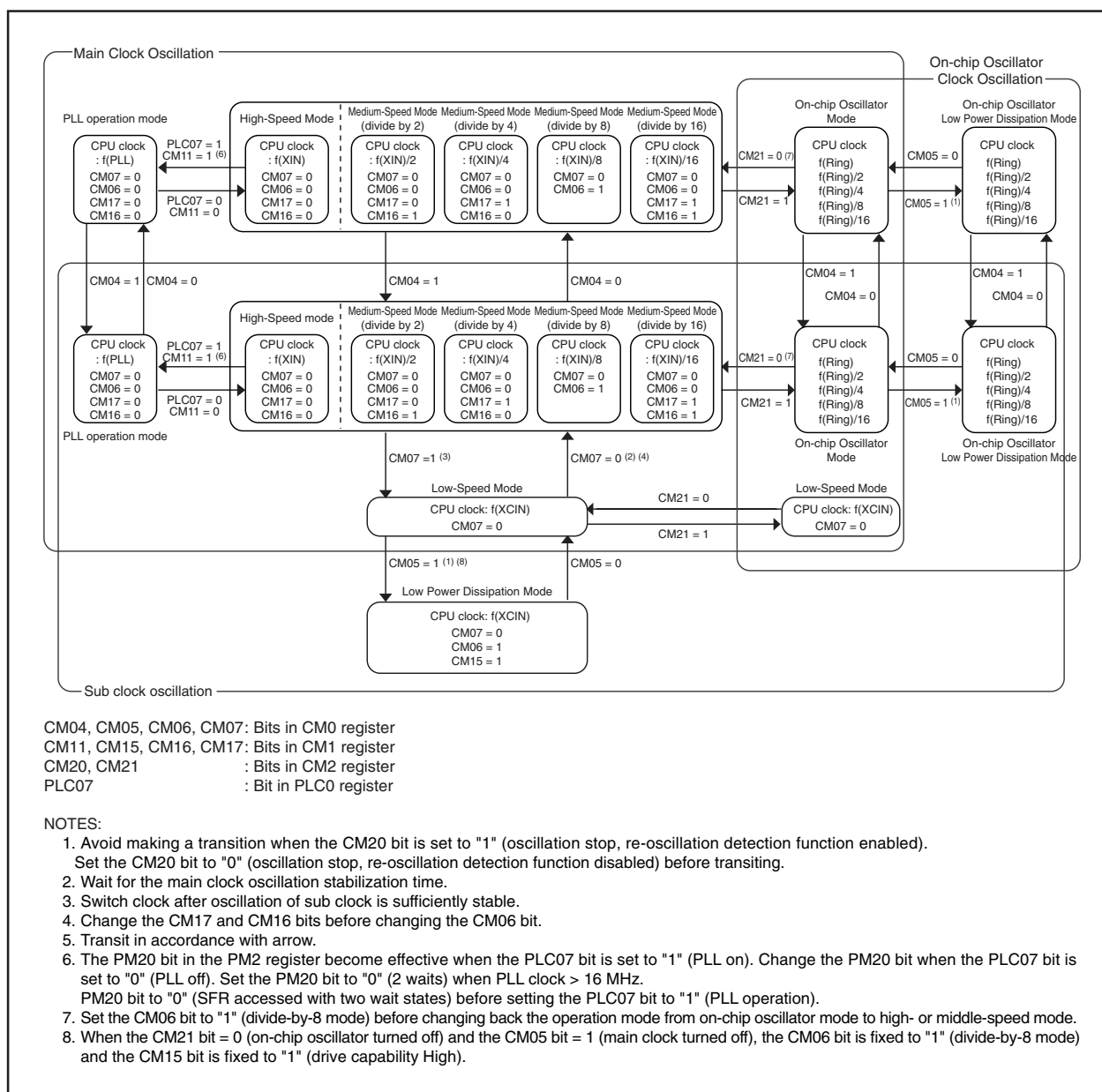


Figure 8.13 State Transition in Normal Operation Mode



Table 8.8 Allowed Transition and Setting <sup>(9)</sup>

		State after transition							
		High-Speed Mode, Medium-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, Medium-Speed Mode	(NOTE 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>	–	–	–	(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>		–	–	(NOTE 8)	(11) <sup>(1)</sup>	(16) <sup>(1)</sup>	(17)
	On-chip Oscillator Low Power Dissipation Mode	–	–	–	–	(10)	(NOTE 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)		

-: Cannot transit

## NOTES:

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

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

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

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

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

## 8.5 Oscillation Stop and Re-oscillation Detection Function

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

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

Table 8.9 lists a specification overview of the oscillation stop and re-oscillation detection function.

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

Item	Specification
Oscillation Stop Detectable Clock and Frequency Bandwidth	$f(XIN) \geq 2 \text{ MHz}$
Enabling Condition for Oscillation Stop and Re-oscillation Detection Function	Set CM20 bit to "1" (enable)
Operation at Oscillation Stop, Re-oscillation Detection	<ul style="list-style-type: none"> <li>•Reset occurs (when CM27 bit = 0)</li> <li>•Oscillation stop, re-oscillation detection interrupt occurs (when the CM27 bit = 1)</li> </ul>

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

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

This status is reset with hardware reset. 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").

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

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

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

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

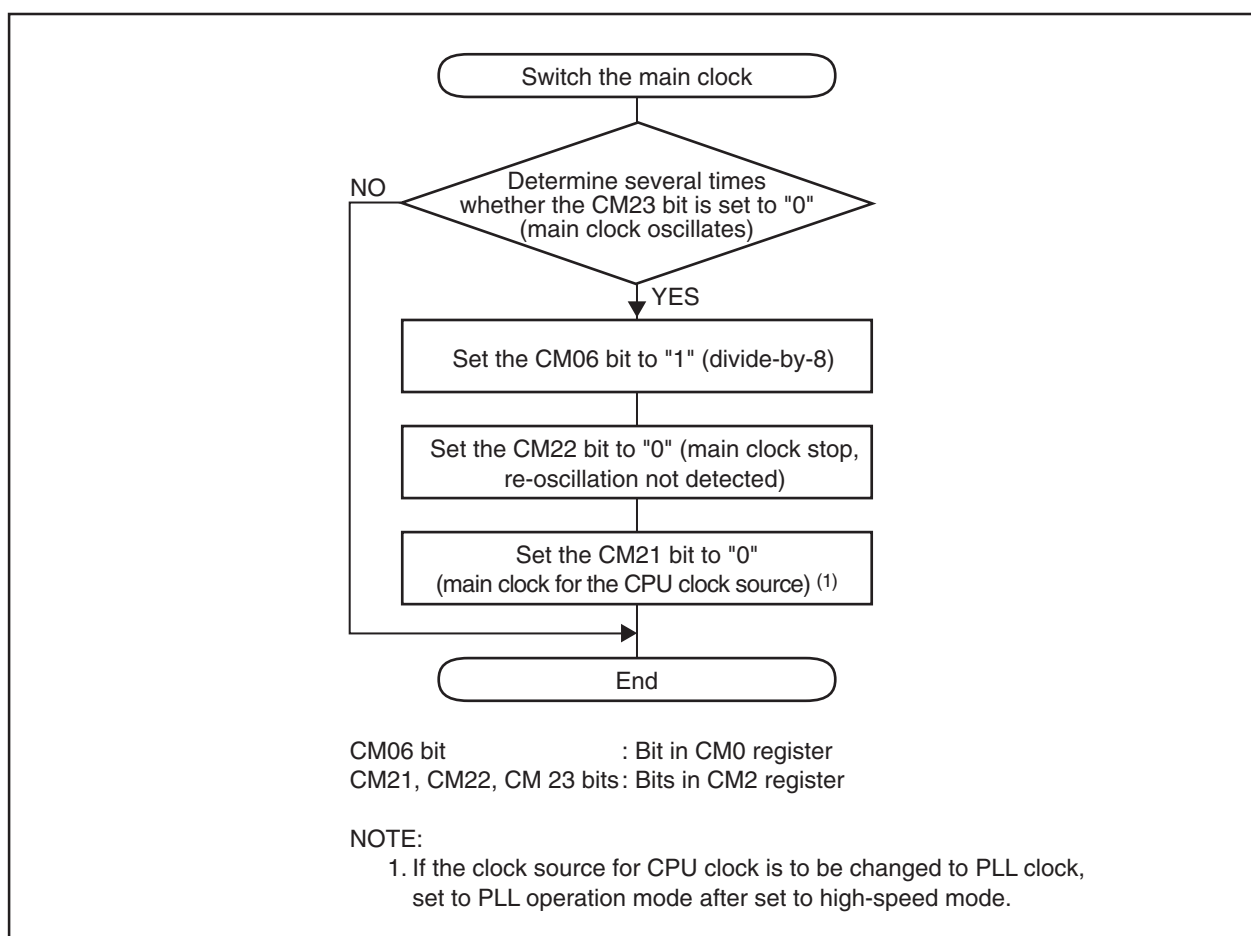
- Oscillation stop, re-oscillation detection interrupt request is generated.
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)
- CM21 bit remains unchanged

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

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

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

- The oscillation stop, re-oscillation detection interrupt shares the vector with the watchdog timer interrupt. If the oscillation stop, re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, the clock source for CPU clock and peripheral function must be switched to the main clock in the program. Figure 8.14 shows the procedure to switch the clock source from the on-chip oscillator to the main clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt request occurrence, the CM22 bit becomes "1". When the CM22 bit is set at "1", oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to "0" in the program, oscillation stop, re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is "1", an oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the on-chip oscillator starts oscillating. In this case, although the CPU clock is derived from the sub clock as it was before the interrupt occurred, the peripheral function clocks now are derived from the on-chip oscillator clock.
- To enter wait mode while using the oscillation stop and re-oscillation detection function, set the CM02 bit to "0" (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop and re-oscillation detection function is provided in preparation for main clock stop due to external 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".



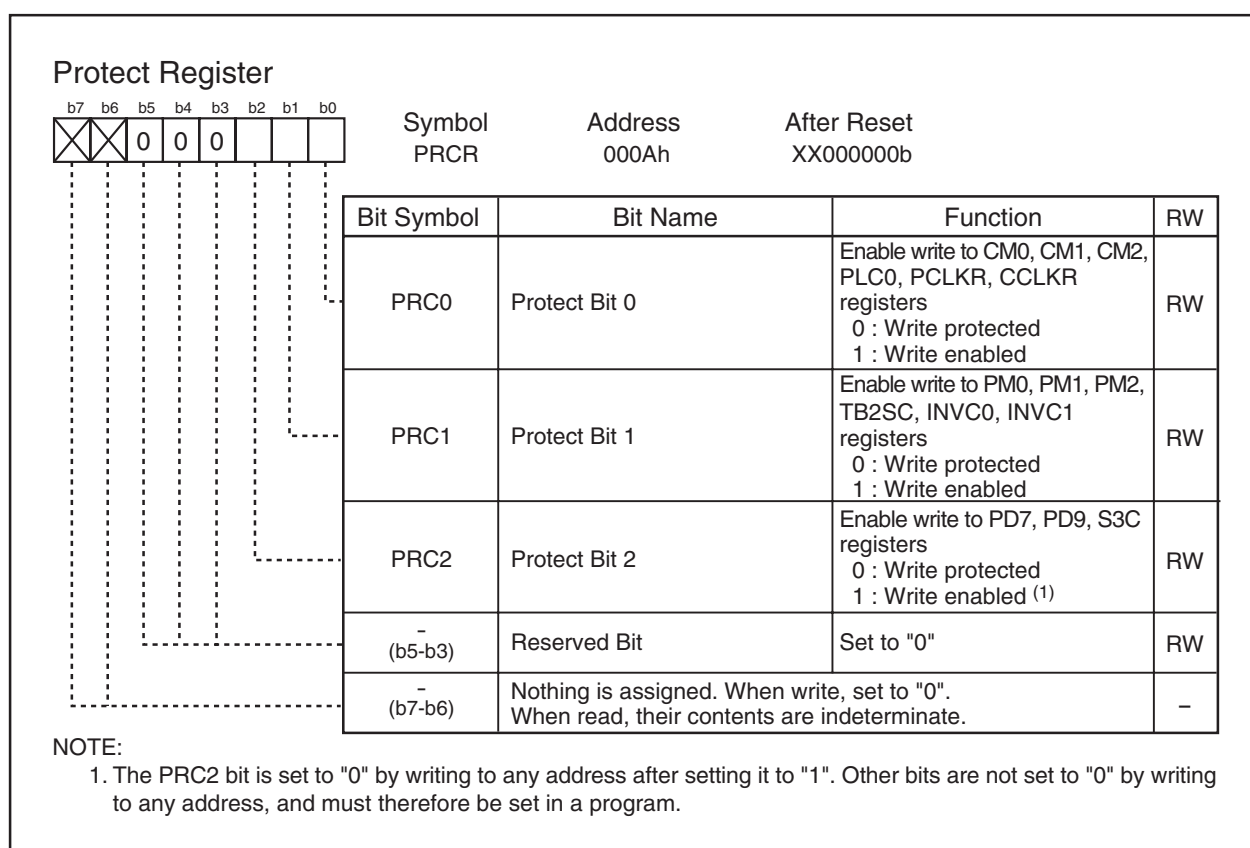
**Figure 8.14 Procedure to Switch Clock Source from On-chip Oscillator to Main Clock**

## 9. Protection

In the event that a program runs out of control, this function protects the important registers so that they will not be rewritten easily. Figure 9.1 shows the PRCR register. The following lists the registers protected by the PRCR register.

- The PRC0 bit protects the CM0, CM1, CM2, PLC0, PCLKR and CCLKR registers;
- The PRC1 bit protects the PM0, PM1, PM2, TB2SC, INVC0 and INVC1 registers;
- The PRC2 bit protects the PD7, PD9 and S3C registers.

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



**Figure 9.1 PRCR Register**

## 10. Interrupt

### 10.1 Type of Interrupts

Figure 10.1 shows the types of interrupts.

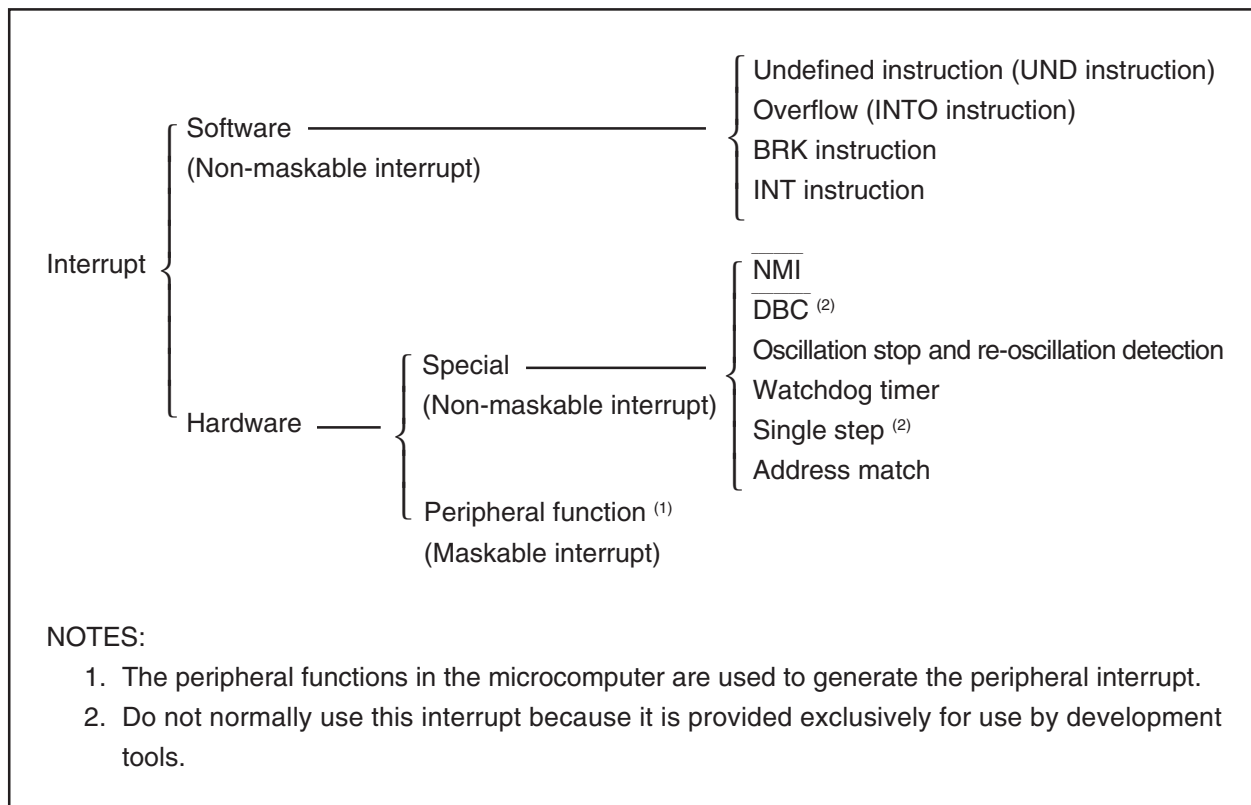


Figure 10.1 Interrupts

- 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.

## 10.2 Software Interrupts

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

### 10.2.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

### 10.2.2 Overflow Interrupt

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

### 10.2.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

### 10.2.4 INT Instruction Interrupt

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

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

## 10.3 Hardware Interrupts

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

### 10.3.1 Special Interrupts

Special interrupts are non-maskable interrupts.

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

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

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

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

#### 10.3.1.3 Watchdog Timer Interrupt

Generated by the watchdog timer. Once a watchdog timer interrupt is generated, be sure to initialize the watchdog timer. For details about the watchdog timer, refer to **11. Watchdog Timer**.

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

Generated by the oscillation stop and re-oscillation detection function. For details about the oscillation stop and re-oscillation detection function, refer to **8. Clock Generating Circuit**.

#### 10.3.1.5 Single-Step Interrupt

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

#### 10.3.1.6 Address Match Interrupt

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

### 10.3.2 Peripheral Function Interrupts

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

## 10.4 Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. Figure 10.2 shows the interrupt vector.

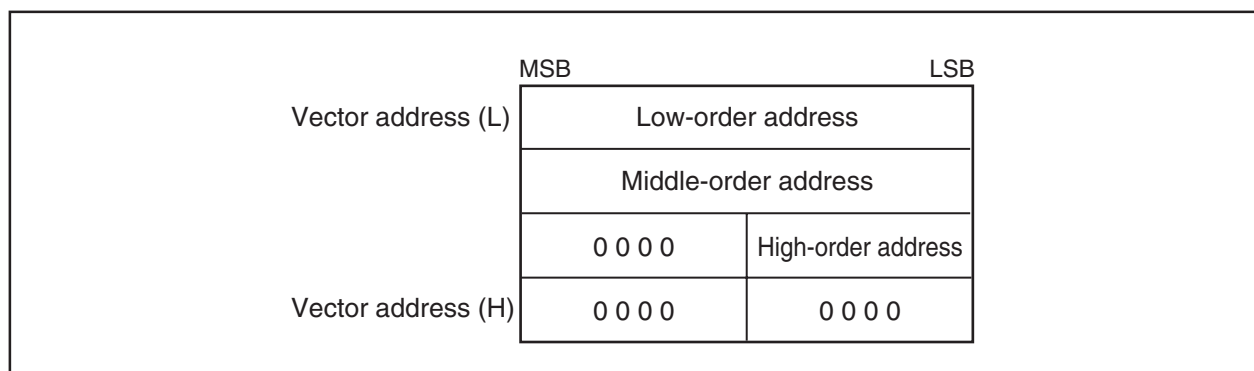


Figure 10.2 Interrupt Vector

### 10.4.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDCh to FFFFFh. Table 10.1 lists the fixed vector tables. In the flash memory version of microcomputer, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to **21.2 Functions to Prevent Flash Memory from Rewriting**.

Table 10.1 Fixed Vector Tables

Interrupt Source	Vector table Addresses Address (L) to Address (H)	Reference
Undefined Instruction (UND instruction)	FFFDCh to FFFDFh	M16C/60, M16C/20, M16C/Tiny Series Software Manual
Overflow (INTO instruction)	FFFE0h to FFFE3h	
BRK Instruction <sup>(2)</sup>	FFFE4h to FFFE7h	
Address Match	FFFE8h to FFFEBh	10.10 Address Match Interrupt
Single Step <sup>(1)</sup>	FFFECh to FFFEFh	
Oscillation Stop and Re-oscillation Detection, Watchdog Timer	FFFF0h to FFFF3h	8. Clock Generating Circuit 11. Watchdog Timer
DBC <sup>(1)</sup>	FFFF4h to FFFF7h	
NMI	FFFF8h to FFFFBh	10.7 NMI Interrupt
Reset	FFFFCh to FFFFFh	5. Reset

NOTES:

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



### 10.4.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. Table 10.2 lists the relocatable vector tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than in the case of odd addresses.

**Table 10.2 Relocatable Vector Tables**

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

**NOTES:**

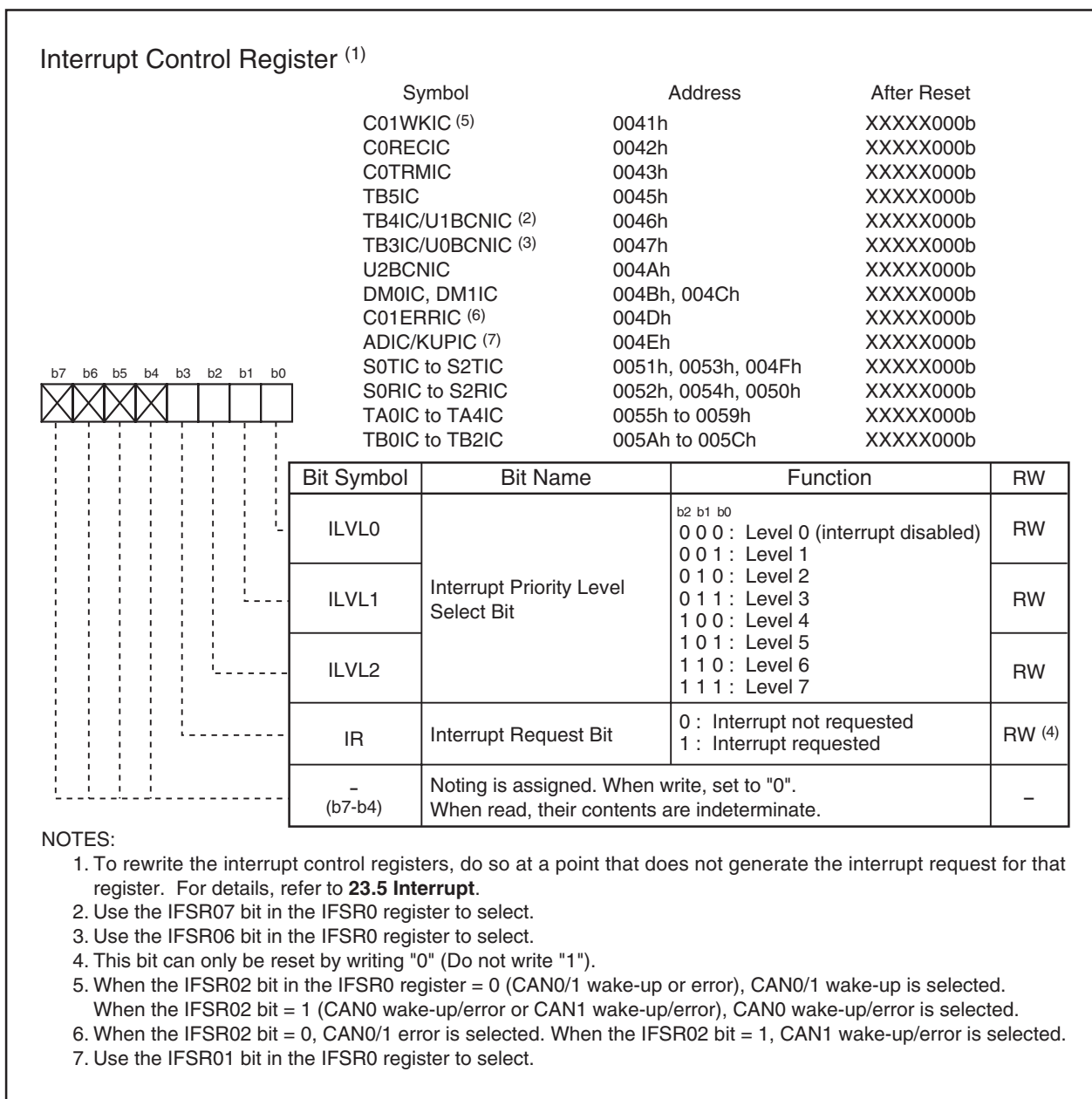
- Address relative to address in INTB.
- These interrupts cannot be disabled using the I flag.
- Use the IFSR07 bit in the IFSR0 register to select.
- Use the IFSR06 bit in the IFSR0 register to select.
- Use the IFSR17 bit in the IFSR1 register to select.
- Use the IFSR16 bit in the IFSR1 register to select.  
Furthermore, use the IFSR00 bit in the IFSR0 register to select, when selecting SI/O3 or CAN1 successful transmission.
- Use the IFSR01 bit in the IFSR0 register to select.
- During I<sup>2</sup>C mode, NACK and ACK interrupts comprise the interrupt source.
- Bus collision detection: During IE mode, this bus collision detection constitutes the cause of an interrupt.  
During I<sup>2</sup>C mode, a start condition or a stop condition detection constitutes the cause of an interrupt.
- Use the IFSR02 bit in the IFSR0 register to select. When the IFSR02 bit = 0, CAN0/1 wake-up is selected. When the IFSR02 bit = 1, CAN0 wake-up/error is selected.
- Use the IFSR02 bit in the IFSR0 register to select. When the IFSR02 bit = 0, CAN0/1 error is selected. When the IFSR02 bit = 1, CAN1 wake-up/error is selected.

## 10.5 Interrupt Control

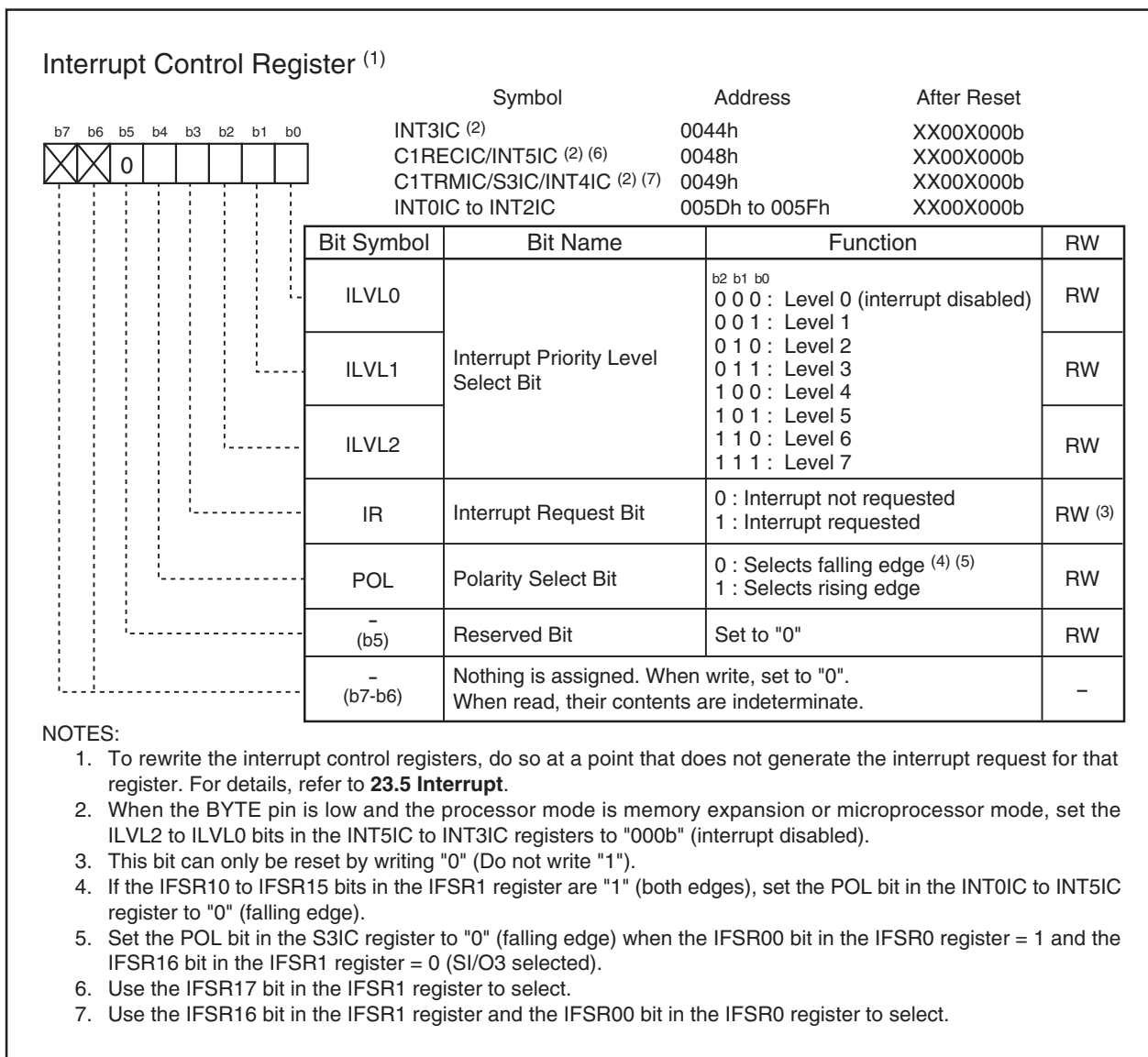
The following describes how to enable/disable the maskable interrupts, and how to set the priority in which order they are accepted. What is explained here does not apply to non-maskable interrupts.

Use the I flag in the FLG register, IPL, and 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 the each interrupt control register.

Figures 10.3 and 10.4 show the interrupt control registers.



**Figure 10.3 Interrupt Control Registers (1)**

**Figure 10.4 Interrupt Control Registers (2)**

### 10.5.1 I Flag

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

### 10.5.2 IR Bit

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

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

### 10.5.3 ILVL2 to ILVL0 Bits and IPL

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

Table 10.3 shows the settings of interrupt priority levels and Table 10.4 shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:

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

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

**Table 10.3 Settings of Interrupt Priority Levels**

ILVL2 to ILVL0 Bits	Interrupt Priority Level	Priority Order
000b	Level 0 (Interrupt disabled)	-
001b	Level 1	<div style="text-align: center;"> Low  ↓  High </div>
010b	Level 2	
011b	Level 3	
100b	Level 4	
101b	Level 5	
110b	Level 6	
111b	Level 7	

**Table 10.4 Interrupt Priority Levels Enabled by IPL**

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

### 10.5.4 Interrupt Sequence

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

If an interrupt request is generated 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 request is generated 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.

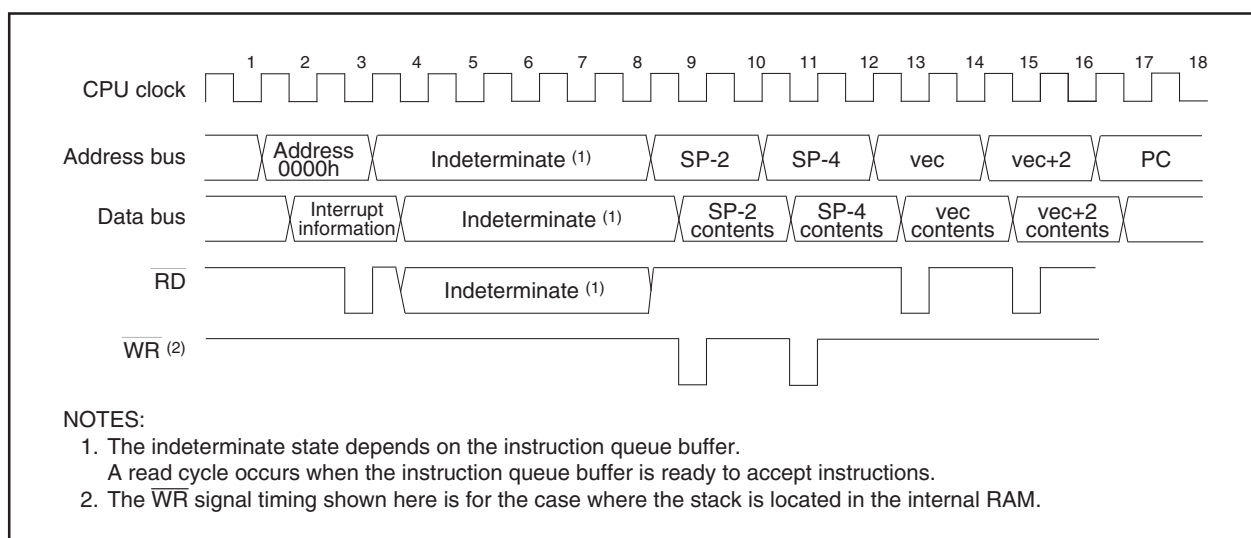
The CPU behavior during the interrupt sequence is described below. Figure 10.5 shows time required for executing the interrupt sequence.

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

After the interrupt sequence is completed, an instruction is executed from the starting address of the interrupt routine.

**NOTE:**

1. Temporary register cannot be modified by users.



**Figure 10.5 Time Required for Executing Interrupt Sequence**

### 10.5.5 Interrupt Response Time

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

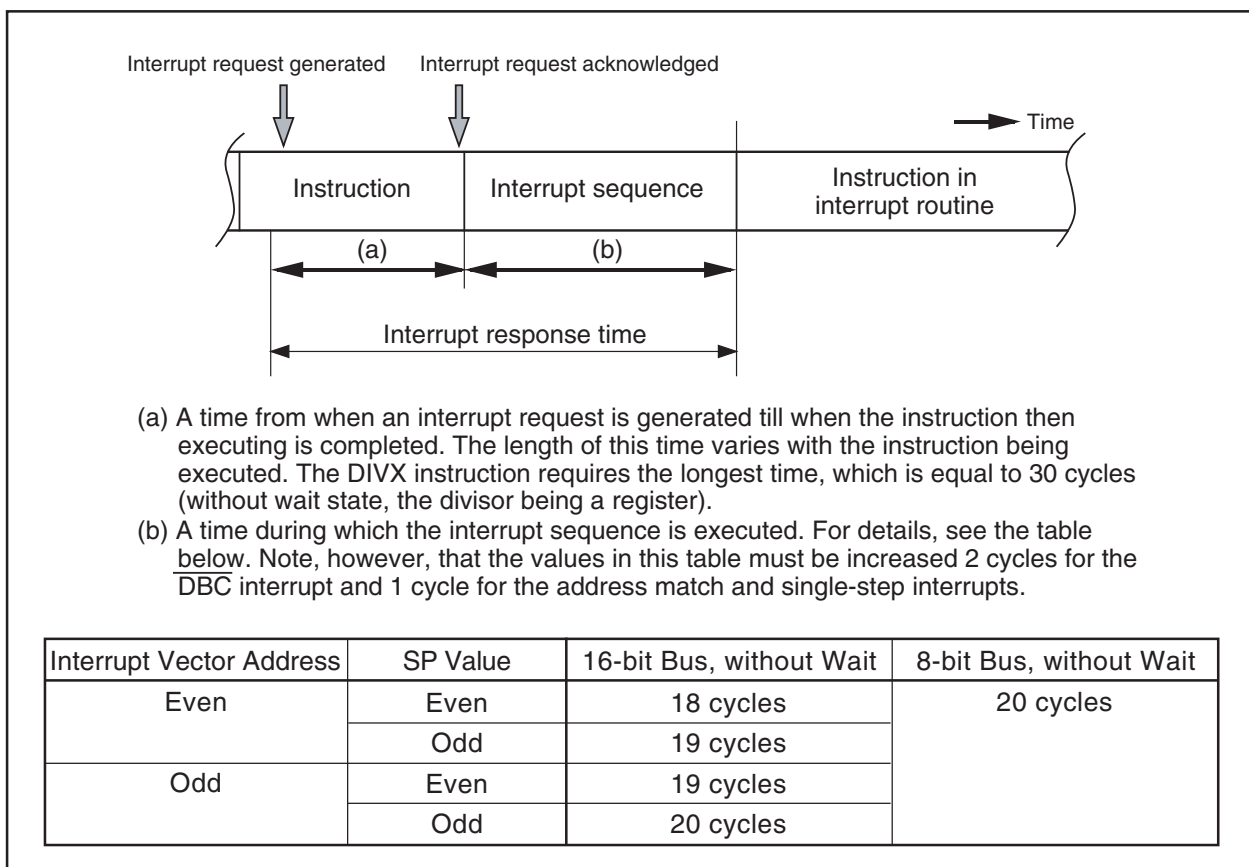


Figure 10.6 Interrupt response time

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

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

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in Table 10.5 is set in the IPL. Table 10.5 shows the IPL values of software and special interrupts when they are accepted.

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

Interrupt Sources	Value that is Set to IPL
Oscillation Stop and Re-oscillation Detection, Watchdog Timer, NMI	7
Software, Address Match, DBC, Single-Step	Not changed

### 10.5.7 Saving Registers

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

At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits in the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits of the PC are saved. Figure 10.7 shows the stack status before and after 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.

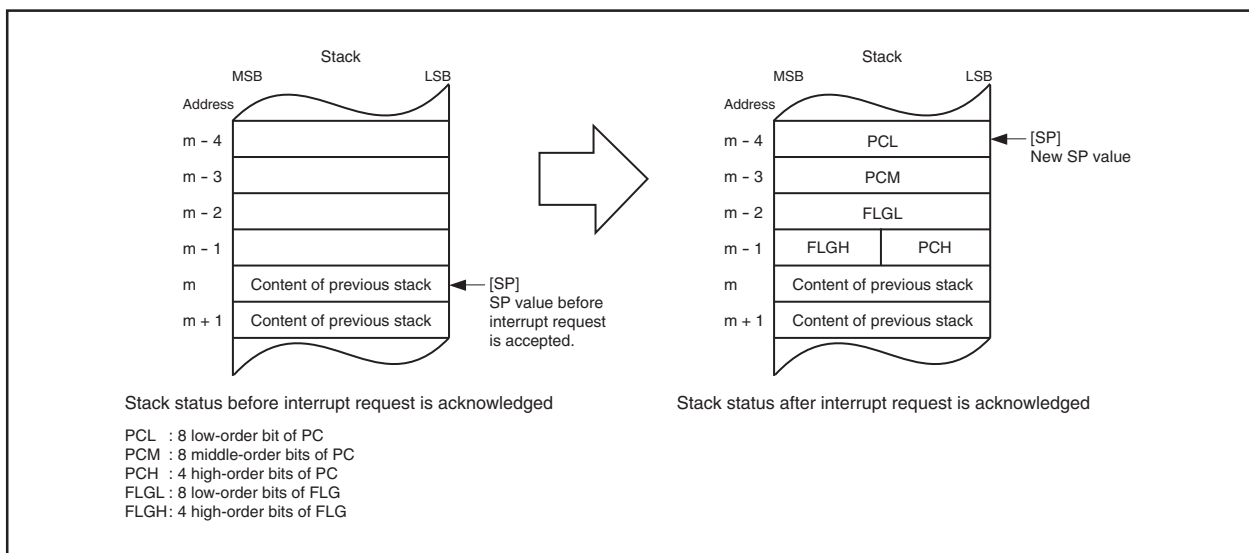


Figure 10.7 Stack Status Before and After Acceptance of Interrupt Request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP <sup>(1)</sup>, at the time of acceptance of an interrupt request, is even or odd. If the SP (Note) is even, the FLG register and the PC are saved, 16 bits at a time. If odd, they are saved in two steps, 8 bits at a time. Figure 10.8 shows the 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.

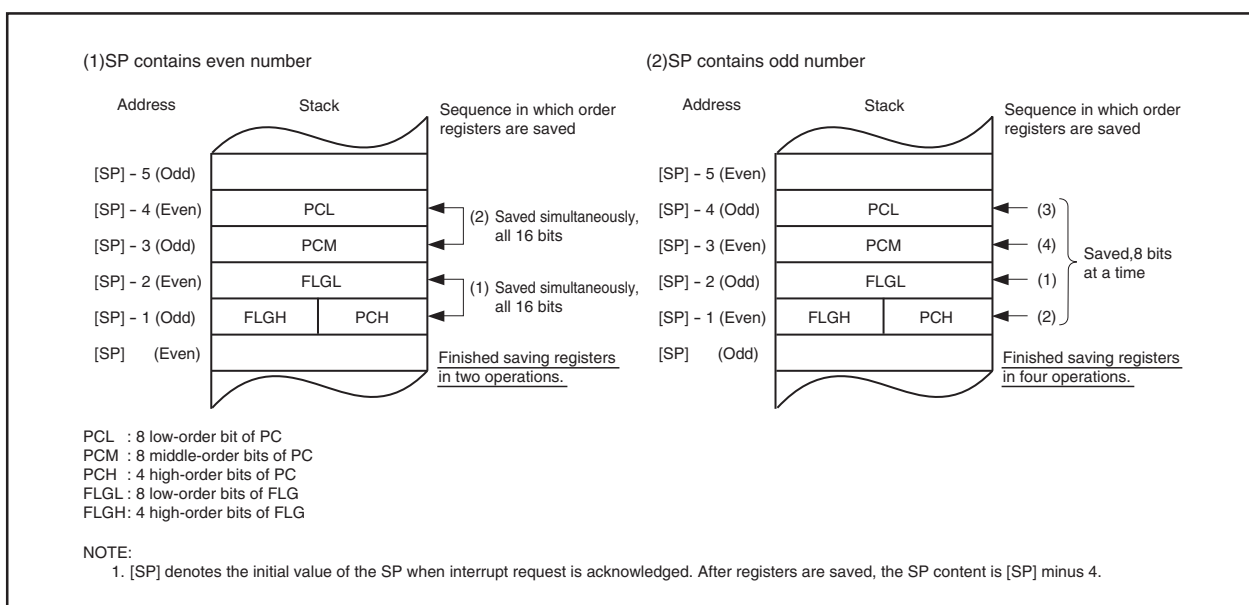


Figure 10.8 Operation of Saving Registers

### 10.5.8 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.

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

### 10.5.9 Interrupt Priority

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

For maskable interrupts (peripheral functions interrupt), 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 10.9 shows the priorities of hardware interrupts.

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

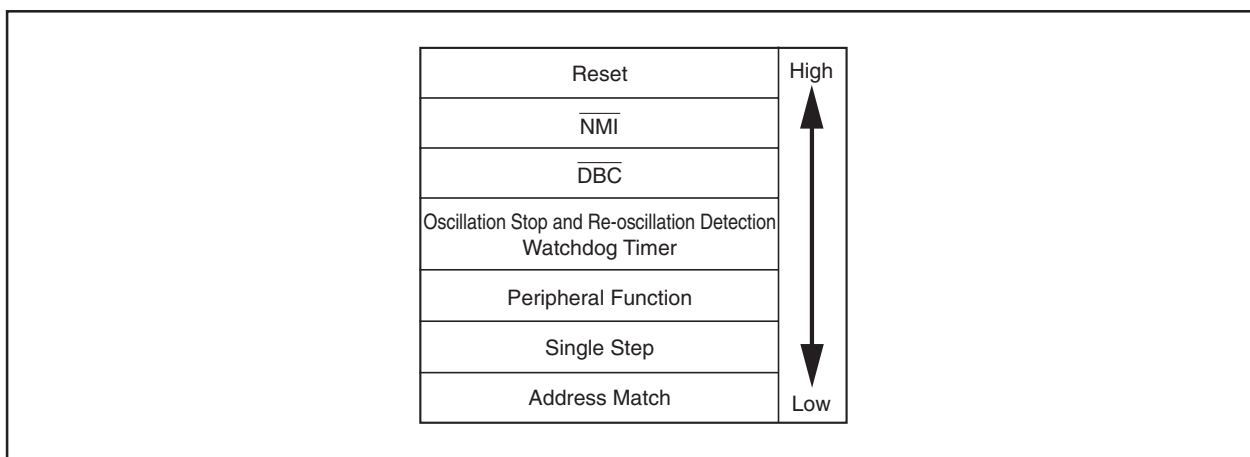


Figure 10.9 Hardware Interrupt Priority

### 10.5.10 Interrupt Priority Resolution Circuit

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

Figure 10.10 shows the circuit that judges the interrupt priority level.



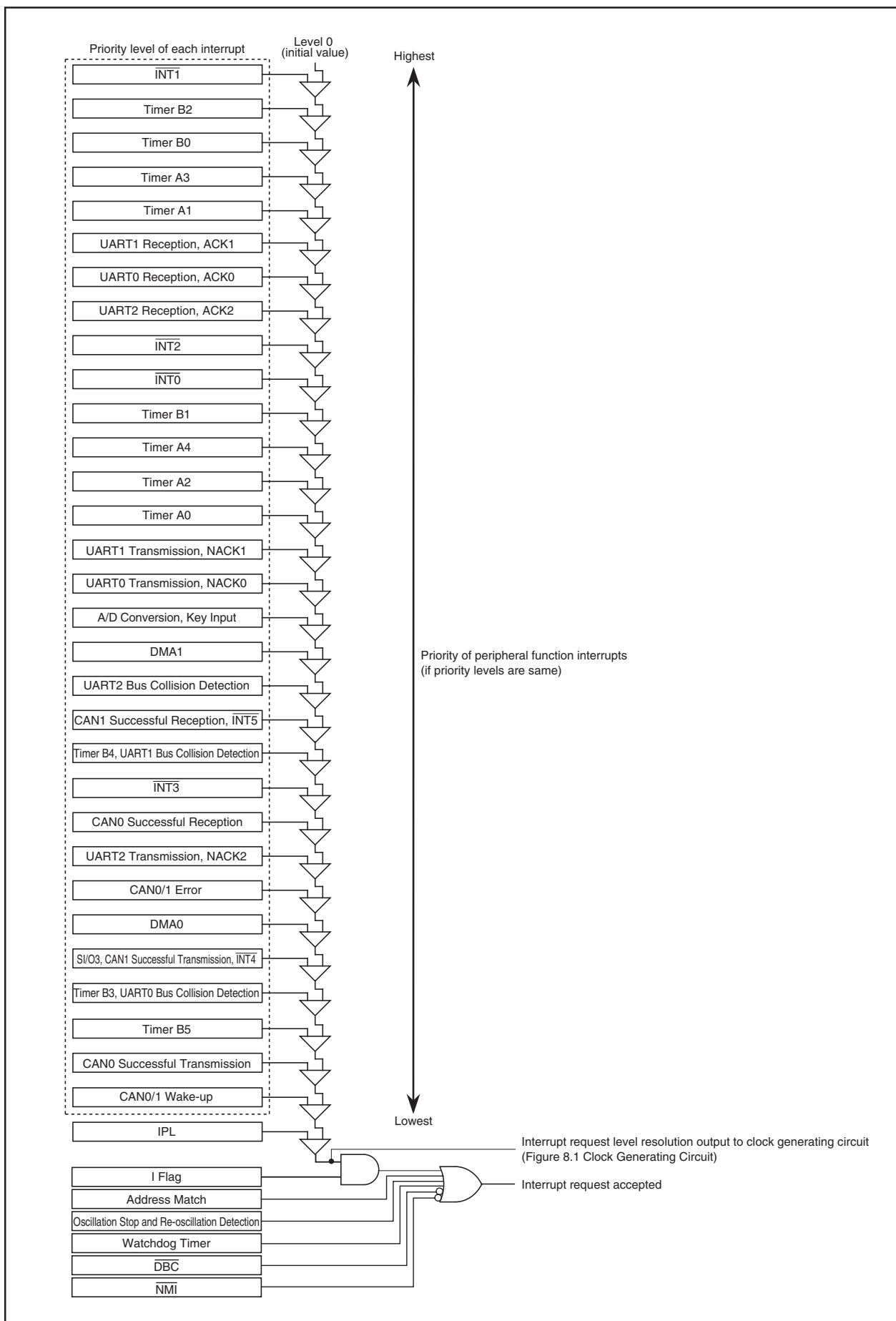


Figure 10.10 Interrupts Priority Select Circuit

## 10.6 INT Interrupt

INT<sub>i</sub> interrupt (i = 0 to 5) is triggered by the edges of external inputs. The edge polarity is selected using the IFSR10 to IFSR15 bits in the IFSR1 register.

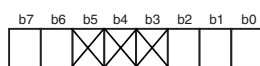
INT4 share the interrupt vector and interrupt control register with CAN1 successful transmission and SI/O3.

INT5 share the interrupt vector and interrupt control register with CAN1 successful reception. To use the INT4 interrupt, set the IFSR16 bit of the IFSR1 register to “1” (INT4). To use the INT5 interrupt, set the IFSR17 bit of the IFSR1 register to “1” (INT5).

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

Figure 10.11 shows the IFSR0 and IFSR1 registers.

## Interrupt Request Cause Select Register 0



Symbol  
IFSR0

Address  
01DEh

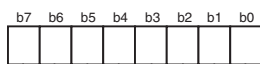
After Reset  
00XX000h

Bit Symbol	Bit Name	Function	RW
IFSR00	Interrupt Request Cause Select Bit	0 : CAN1 successful transission 1 : SI/O3	RW
IFSR01	Interrupt Request Cause Select Bit	0 : A/D conversion 1 : Key input	RW
IFSR02	Interrupt Request Cause Select Bit <sup>(3)</sup>	0 : CAN0/1 wake-up or error 1 : CAN0 wake-up/error or CAN1 wake-up/error	RW
- (b5-b3)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		RW
IFSR06	Interrupt Request Cause Select Bit <sup>(1)</sup>	0 : Timer B3 1 : UART0 bus collision detection	RW
IFSR07	Interrupt Request Cause Select Bit <sup>(2)</sup>	0 : Timer B4 1 : UART1 bus collision detection	RW

## NOTES:

1. Timer B3 and UART0 bus collision detection share the vector and interrupt control register.  
When using the timer B3 interrupt, set the IFSR06 bit to "0" (Tmer B3).  
When using UART0 bus collision detection, set the IFSR06 bit to "1" (UART0 bus collision detection).
2. Timer B4 and UART1 bus collision detection share the vector and interrupt control register.  
When using the timer B4 interrupt, set the IFSR07 bit to "0" (Timer B4).  
When using UART1 bus collision detection, set the IFSR07 bit to "1" (UART1 bus collision detection).
3. If this bit is set to "0", the software interrupt number 1 is selected CAN0/1 wake-up and the interrupt number 13 is selected CAN0/1 error. If this bit is set to "1", the interrupt number 1 is selected CAN0 wake-up/error and the interrupt number 13 is selected CAN1 wake-up/error.

## Interrupt Request Cause Select Register 1



Symbol  
IFSR1

Address  
01DFh

After Reset  
00h

Bit Symbol	Bit Name	Function	RW
IFSR10	INT0 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges <sup>(1)</sup>	RW
IFSR11	INT1 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges <sup>(1)</sup>	RW
IFSR12	INT2 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges <sup>(1)</sup>	RW
IFSR13	INT3 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges <sup>(1)</sup>	RW
IFSR14	INT4 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges <sup>(1)</sup>	RW
IFSR15	INT5 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges <sup>(1)</sup>	RW
IFSR16	Interrupt Request Cause Select Bit <sup>(2)</sup>	0 : SI/O3/CAN1 successful transmission <sup>(3)</sup> 1 : INT4	RW
IFSR17	Interrupt Request Cause Select Bit <sup>(2)</sup>	0 : CAN1 successful reception 1 : INT5	RW

## NOTES:

1. When setting this bit to "1" (both edges), make sure the POL bit in the INT0IC to INT5IC register is set to "0" (falling edge).
2. During memory expansion and microprocessor modes, when the data bus is 16-bit width (BYTE pin is "L"), set this bit to "0".
3. When setting this bit to "0" (SI/O3, CAN1 successful transmission), make sure the IFSR00 bit in the IFSR0 register is set to "0" (CAN1 successful transmission) or "1" (SI/O3).  
And, make sure the POL bit in the S3IC and C1TRMIC registers are set to "0" (falling edge).

Figure 10.11 IFSR0, IFSR1 Registers

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

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

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

This pin cannot be used as an input port.

## 10.8 Key Input Interrupt

Of P10\_4 to P10\_7, a key input interrupt request is generated when input on any of the P10\_4 to P10\_7 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 wake up 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 P10\_4 to P10\_7 as analog input ports. Figure 10.12 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.

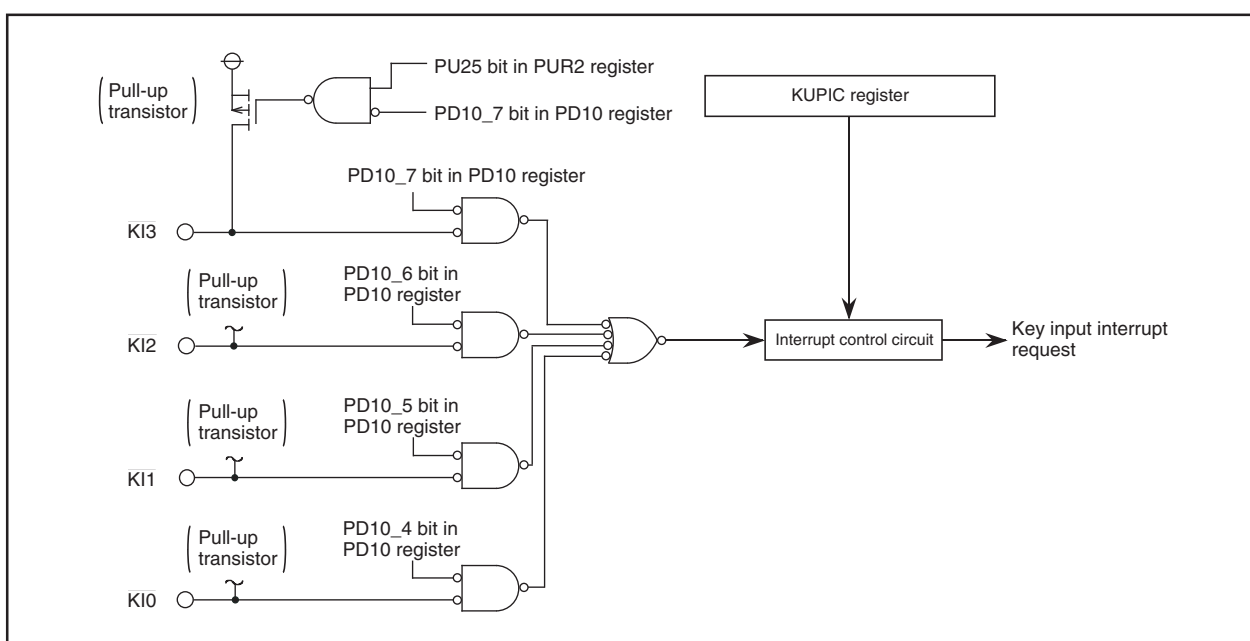


Figure 10.12 Key Input Interrupt Block Diagram

## 10.9 CAN0/1 Wake-up Interrupt

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

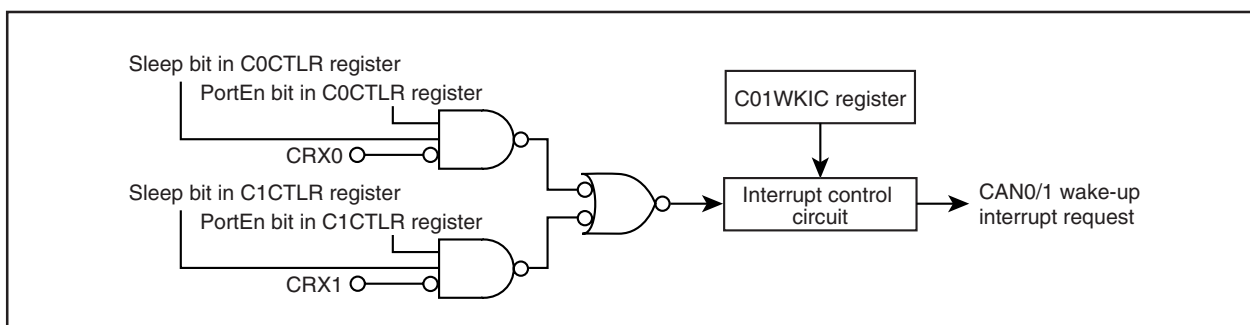


Figure 10.13 CAN0/1 Wake-up Interrupt Block Diagram

## 10.10 Address Match Interrupt

An address match interrupt request is generated immediately before executing the instruction at the address indicated by the RMADi register (i = 0 to 3). Set the start address of any instruction in the RMADi register. Use the AIER0 and AIER1 bits in the AIER register and the AIER20 and AIER21 bits in the AIER2 register to enable or disable the interrupt. Note that the address match interrupt is unaffected by the I flag and IPL. For address match interrupts, the value of the PC that is saved to the stack area varies depending on the instruction being executed (refer to **10.5.7 Saving Registers**). (The value of the PC that is saved to the stack area is not the correct return address.) Therefore, follow one of the methods described below to return from the address match interrupt.

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

Table 10.6 shows the value of the PC that is saved to the stack area when an address match interrupt request is accepted. Table 10.7 shows the relationship between address match interrupt sources and associated registers.

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

Figure 10.14 shows the AIER, AIER2, and RMAD0 to RMAD3 registers.

**Table 10.6 Value of PC That is Saved to Stack Area When Address Match Interrupt Request is Accepted**

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

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

**Table 10.7 Relationship Between Address Match Interrupt Sources and Associated Registers**

Address Match Interrupt Sources	Address Match Interrupt Enable Bit	Address Match Interrupt Register
Address Match Interrupt 0	AIER0	RMAD0
Address Match Interrupt 1	AIER1	RMAD1
Address Match Interrupt 2	AIER20	RMAD2
Address Match Interrupt 3	AIER21	RMAD3

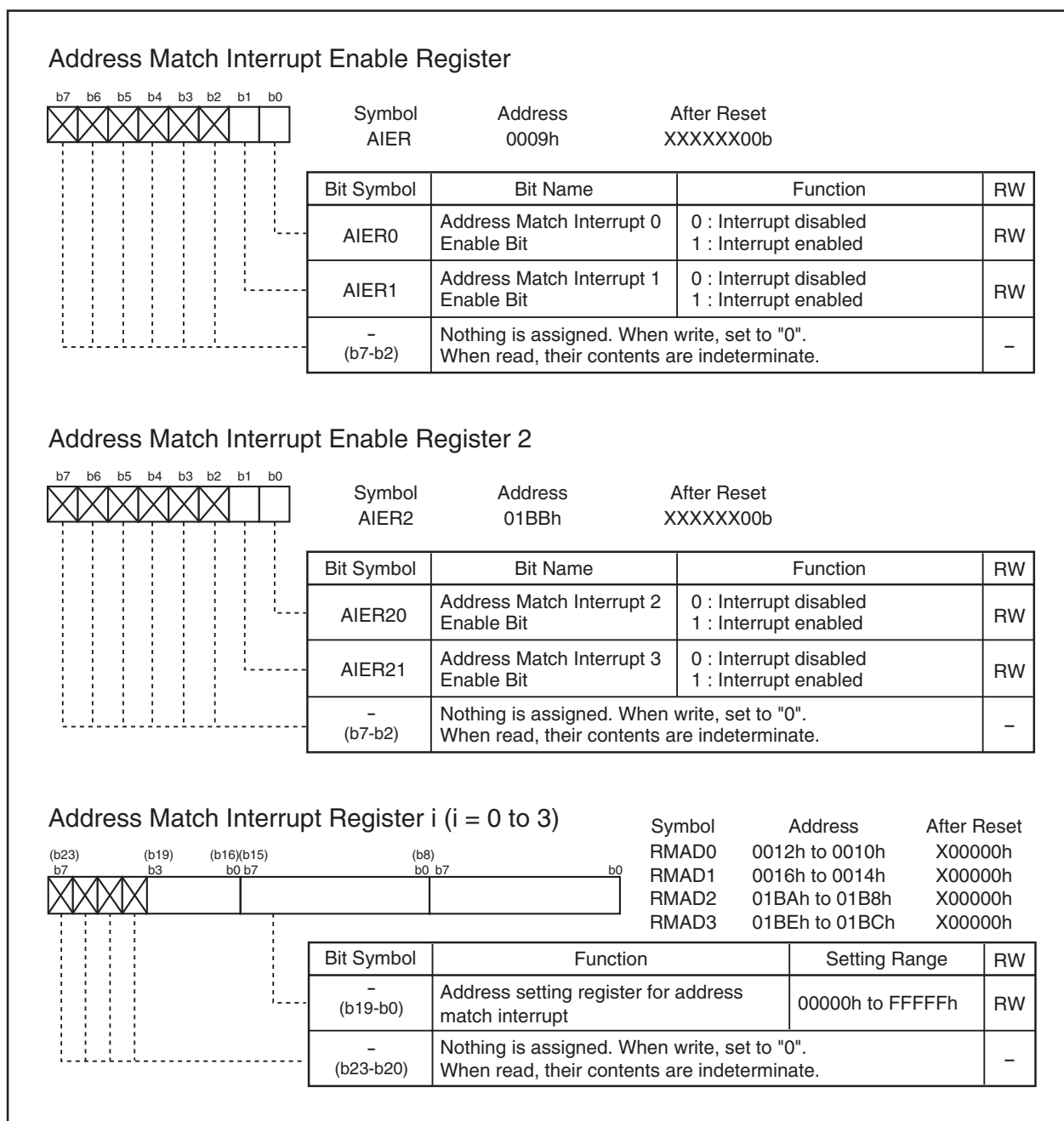


Figure 10.14 AIER Register, AIER2 Register and RMAD0 to RMAD3 Registers

## 11. Watchdog Timer

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

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

With main clock, on-chip oscillator clock or PLL clock selected for CPU clock

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

With sub clock selected for CPU clock

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

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

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

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

Figure 11.1 shows the block diagram of the watchdog timer. Figure 11.2 shows the watchdog timer-related registers.

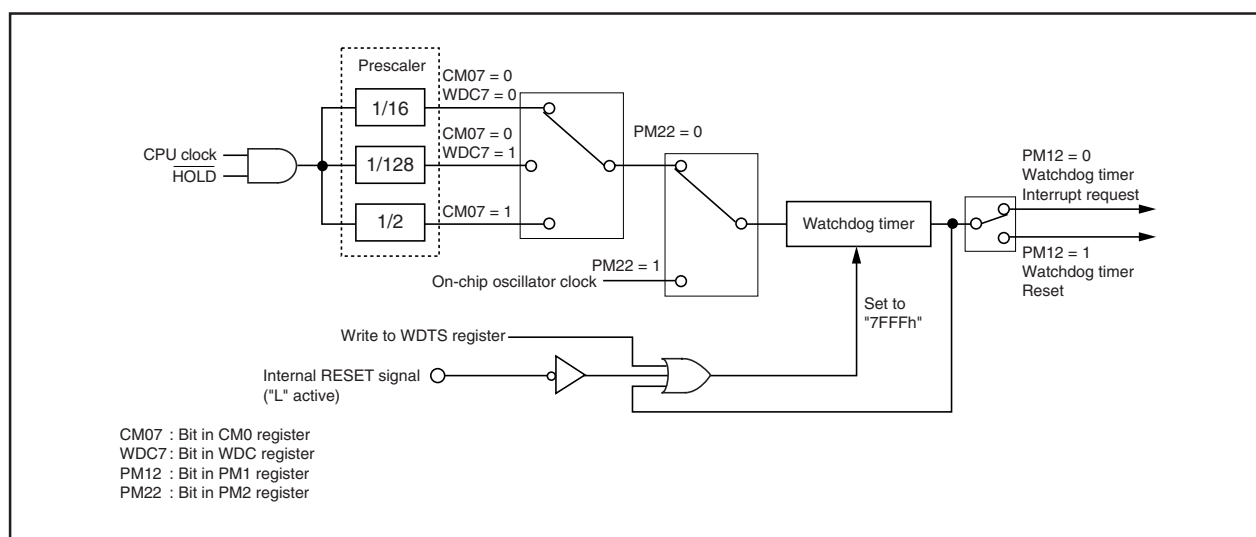


Figure 11.1 Watchdog Timer Block Diagram

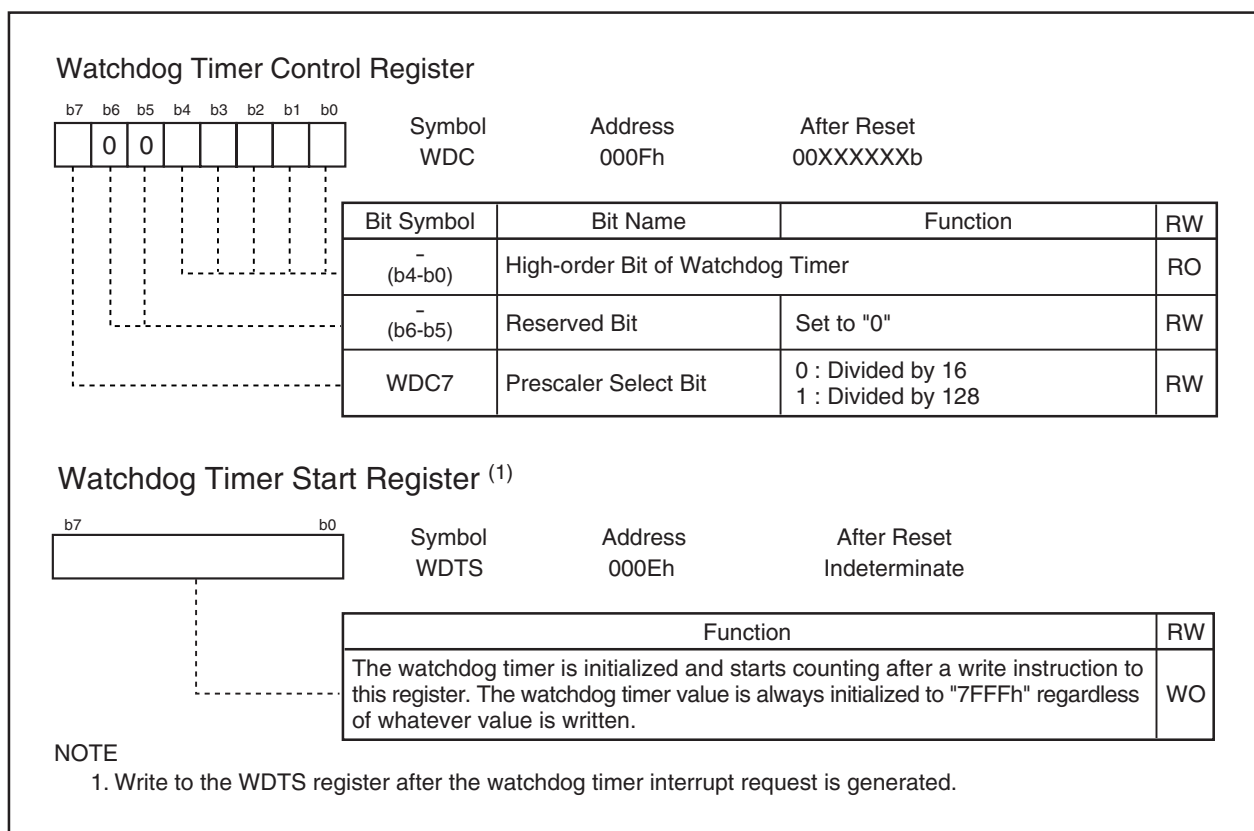


Figure 11.2 WDC Register and WDTS Register

### 11.1 Count Source Protective Mode

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

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

- (1) Set the PRC1 bit in the PRCR register to "1" (enable writes to the 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 the 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 starts oscillating, and the on-chip oscillator clock becomes the watchdog timer count source.

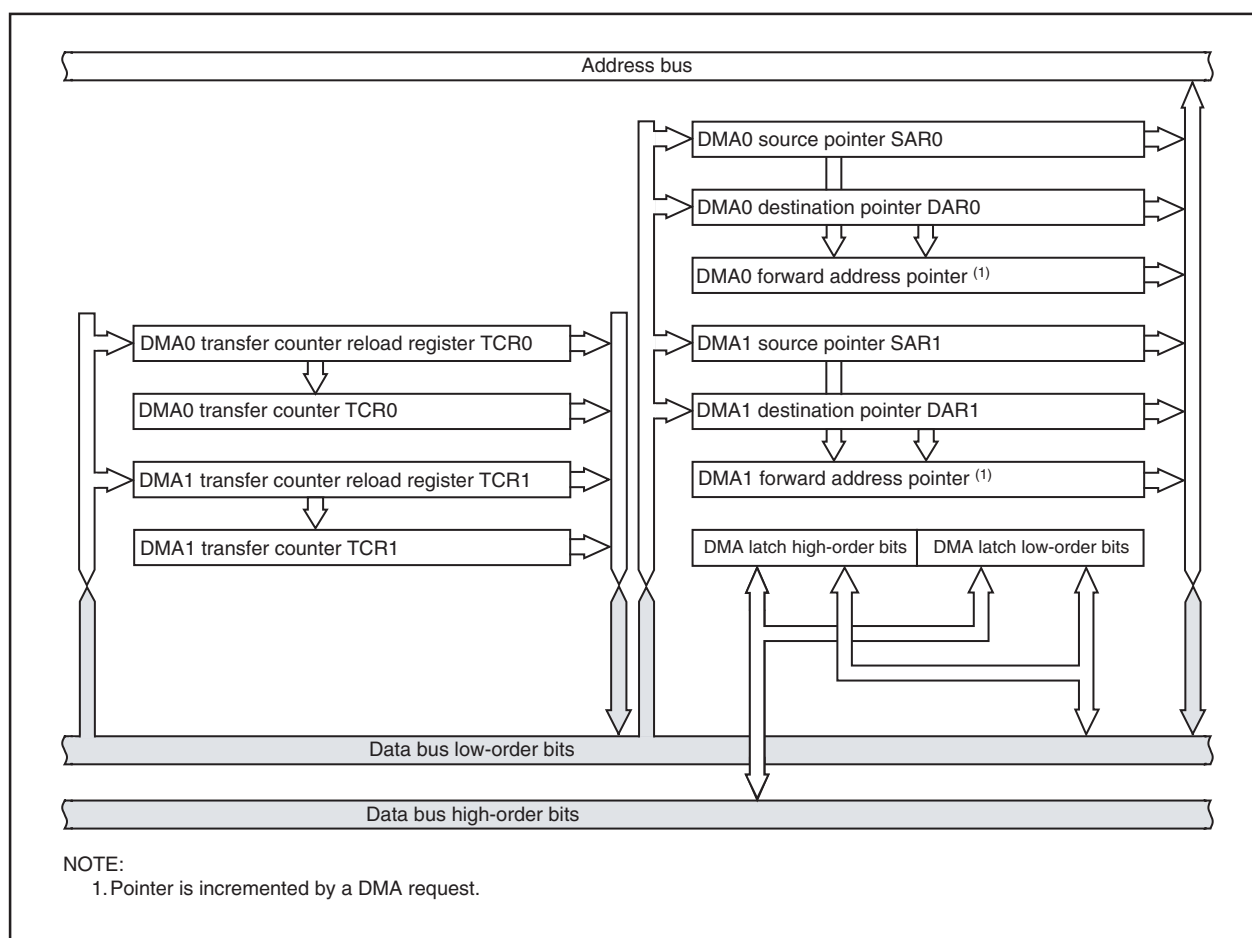
$$\text{Watchdog timer period} = \frac{\text{Watchdog timer count (32768)}}{\text{on-chip oscillator clock}}$$

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



## 12. DMAC

The DMAC (Direct Memory Access Controller) allows data to be transferred without the CPU intervention. Two DMAC channels are included. Each time a DMA request occurs, the DMAC transfers one (8- or 16-bit) data from the source address to the destination address. The DMAC uses the same data bus as used by the CPU. Because the DMAC has higher priority of bus control than the CPU and because it makes use of a cycle steal method, it can transfer one word (16 bits) or one byte (8 bits) of data within a very short time after a DMA request is generated. Figure 12.1 shows the block diagram of the DMAC. Table 12.1 shows the DMAC specifications. Figures 12.2 to 12.4 show the DMAC related-registers.



**Figure 12.1 DMAC Block Diagram**

A DMA request is generated by a write to the DSR bit in the DMiSL register ( $i = 0, 1$ ), as well as by an interrupt request which is generated by any function specified by 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 = 1 (DMA enabled). However, if the cycle in which a DMA request is generated is faster than the DMA transfer cycle, the number of transfer requests generated and the number of times data is transferred may not match. For details, refer to **12.4 DMA Request**.

**Table 12.1 DMAC Specifications**

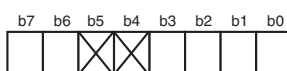
Item		Specification
No. of Channels		2 (cycle steal method)
Transfer Memory Space		<ul style="list-style-type: none"> <li>• From any address in the 1-Mbyte space to a fixed address</li> <li>• From a fixed address to any address in the 1-Mbyte space</li> <li>• From a fixed address to a fixed address</li> </ul>
Maximum No. of Bytes Transferred		128 Kbytes (with 16-bit transfer) or 64 Kbytes (with 8-bit transfer)
DMA Request Factors <sup>(1)</sup> <sup>(2)</sup>		Falling edge of INT0 or INT1 Both edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 transfer, UART0 reception interrupt requests UART1 transfer, UART1 reception interrupt requests UART2 transfer, UART2 reception interrupt requests SI/O3 interrupt request A/D conversion interrupt requests Software triggers
Channel Priority		DMA0 > DMA1 (DMA0 takes precedence)
Transfer Unit		8 bits or 16 bits
Transfer Address Direction		forward or fixed (The source and destination addresses cannot both be in the forward direction.)
Transfer Mode	Single Transfer	Transfer is completed when the DMA <sub>i</sub> transfer counter 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 underflowed
DMA Start Up		Data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMA <sub>i</sub> CON register = 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)
Reload Timing for Forward Address Pointer and Transfer Counter		When a data transfer is started after setting the DMAE bit to "1" (enabled), the forward address pointer is reloaded with the value of the SAR <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.
DMA Transfer Cycles		Minimum 3 cycles between SFR and internal RAM

i = 0, 1

**NOTES:**

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

## DMA0 Request Cause Select Register

Symbol  
DM0SLAddress  
03B8hAfter Reset  
00h

Bit Symbol	Bit Name	Function	RW
DSEL0	DMA Request Cause Select Bit	See <b>NOTE 1</b>	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
– (b5-b4)	Nothing is assigned. When write, set to "0". When read, their contents are "0".		–
DMS	DMA Request Cause Expansion Select Bit	0 : Basic cause of request 1 : Extended cause of request	RW
DSR	Software DMA Request Bit	A DMA request is generated by setting this bit to "1" when the DMS bit is "0" (basic cause) and the DSEL3 to DSEL0 bits are "0001b" (software trigger). The value of this bit when read is "0".	RW

## NOTE:

1. The causes of DMA0 requests can be selected by a combination of the DMS bit and the DSEL3 to DSEL0 bits in the manner described below.

DSEL3 to DSEL0 Bits	DMS = 0 (basic cause of request)	DMS = 1 (extended cause of request)
0000b	Falling edge of INT0 pin	—
0001b	Software trigger	—
0010b	Timer A0	—
0011b	Timer A1	—
0100b	Timer A2	—
0101b	Timer A3	—
0110b	Timer A4	Two edges of INT0 pin
0111b	Timer B0	Timer B3
1000b	Timer B1	Timer B4
1001b	Timer B2	Timer B5
1010b	UART0 transmit	—
1011b	UART0 receive	—
1100b	UART2 transmit	—
1101b	UART2 receive	—
1110b	A/D conversion	—
1111b	UART1 transmit	—

Figure 12.2 DM0SL Register

## DMA1 Request Cause Select Register

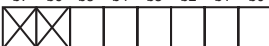
<div><div><div>b7</div><div>b6</div><div>b5</div><div>b4</div><div>b3</div><div>b2</div><div>b1</div><div>b0</div></div><div><div><div></div><div></div><div><div></div><div></div></div><div><div></div><div></div></div><div></div><div></div><div></div><div></div></div></div></div>								Symbol DM1SL	Address 03BAh	After Reset 00h	
								Bit Symbol	Bit Name	Function	RW
								DSEL0	DMA Request Cause Select Bit	See <b>NOTE 1</b>	RW
								DSEL1			RW
								DSEL2			RW
								DSEL3			RW
								- (b5-b4)	Nothing is assigned. When write, set to "0". When read, their contents are "0".		-
								DMS	DMA Request Cause Expansion Select Bit	0 : Basic cause of request 1 : Extended cause of request	RW
								DSR	Software DMA Request Bit	A DMA request is generated by setting this bit to "1" when the DMS bit is "0" (basic cause) and the DSEL3 to DSEL0 bits are "0001b" (software trigger). The value of this bit when read is "0".	RW

## NOTE:

1. The causes of DMA1 requests can be selected by a combination of the DMS bit and the DSEL3 to DSEL0 bits in the manner described below.

DSEL3 to DSEL0 Bits	DMS = 0 (basic cause of request)	DMS = 1 (extended cause of request)
0000b	Falling edge of INT1 pin	—
0001b	Software trigger	—
0010b	Timer A0	—
0011b	Timer A1	—
0100b	Timer A2	—
0101b	Timer A3	SI/O3
0110b	Timer A4	—
0111b	Timer B0	Two edges of INT1 pin
1000b	Timer B1	—
1001b	Timer B2	—
1010b	UART0 transmit	—
1011b	UART0 receive/ACK0	—
1100b	UART2 transmit	—
1101b	UART2 receive/ACK2	—
1110b	A/D conversion	—
1111b	UART1 transmit/ACK1	—

## DMAi Control Register (i = 0, 1)

								Symbol DM0CON DM1CON	Address 002Ch 003Ch	After Reset 00000X00b 00000X00b
Bit Symbol	Bit Name	Function	RW							
DMBIT	Transfer Unit Bit Select Bit	0 : 16 bits 1 : 8 bits	RW							
DMASL	Repeat Transfer Mode Select Bit	0 : Single transfer 1 : Repeat transfer	RW							
DMAS	DMA Request Bit	0 : DMA not requested 1 : DMA requested	RW <sup>(1)</sup>							
DMAE	DMA Enable Bit	0 : Disabled 1 : Enabled	RW							
DSD	Source Address Direction Select Bit <sup>(2)</sup>	0 : Fixed 1 : Forward	RW							
DAD	Destination Address Direction Select Bit <sup>(2)</sup>	0 : Fixed 1 : Forward	RW							
- (b7-b6)	Nothing is assigned. When write, set to "0". When read, their contents are "0".		-							

## NOTES:

1. The DMAS bit can be set to "0" by writing "0" in a program. (This bit remains unchanged even if "1" is written.)  
 2. At least one of the DAD and DSD bits must be "0" (address direction fixed).

Figure 12.3 DM1SL Register, DM0CON and DM1CON Registers

DMAi Source Pointer (i = 0, 1) <sup>(1)</sup>

(b23) b7	(b19) b3	(b16)(b15) b0 b7	(b8) b0 b7			
					Symbol	Address
					SAR0	0022h to 0020h
					SAR1	0032h to 0030h
						After Reset
						Indeterminate
						Indeterminate
					Function	Setting Range
					Set the source address of transfer	00000h to FFFFFh
					Nothing is assigned. When write, set to "0". When read, their contents are "0".	-

## NOTE:

1. If the DSD bit in the DMiCON register is "0" (fixed), this register can only be written to when the DMAE bit in the DMiCON register is "0" (DMA disabled).  
If the DSD bit is "1" (forward direction), this register can be written to at any time.  
If the DSD bit is "1" and the DMAE bit is "1" (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

DMAi Destination Pointer (i = 0, 1) <sup>(1)</sup>

(b23) b7	(b19) b3	(b16)(b15) b0 b7	(b8) b0 b7			
					Symbol	Address
					DAR0	0026h to 0024h
					DAR1	0036h to 0034h
						After Reset
						Indeterminate
						Indeterminate
					Function	Setting Range
					Set the destination address of transfer	00000h to FFFFFh
					Nothing is assigned. When write, set to "0". When read, their contents are "0".	-

## NOTE:

1. If the DAD bit in the DMiCON register is "0" (fixed), this register can only be written to when the DMAE bit in the DMiCON register is "0" (DMA disabled).  
If the DAD bit is "1" (forward direction), this register can be written to at any time.  
If the DAD bit is "1" and the DMAE bit is "1" (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

## DMAi Transfer Counter (i = 0, 1)

(b15) b7	(b8) b0 b7					
					Symbol	Address
					TCR0	0029h, 0028h
					TCR1	0039h, 0038h
						After Reset
						Indeterminate
						Indeterminate
					Function	Setting Range
					Set the transfer count minus 1. The written value is stored in the DMAi transfer counter reload register, and when the DMAE bit in the DMiCON register is set to "1" (DMA enabled) or the DMAi transfer counter underflows when the DMASL bit in the DMiCON register is "1" (repeat transfer), the value of the DMAi transfer counter reload register is transferred to the DMAi transfer counter. When read, the DMAi transfer counter is read.	0000h to FFFFh
						RW

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

## 12.1 Transfer Cycle

The transfer cycle consists of a memory or SFR read (source read) bus cycle and a write (destination write) bus cycle. The number of read and write bus cycles is affected by the source and destination addresses of transfer. During memory expansion and microprocessor modes, it is also affected by the BYTE pin level. Furthermore, the bus cycle itself is extended by a software wait or RDY signal.

### 12.1.1 Effect of Source and Destination Addresses

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

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

### 12.1.2 Effect of BYTE Pin Level

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

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

### 12.1.3 Effect of Software Wait

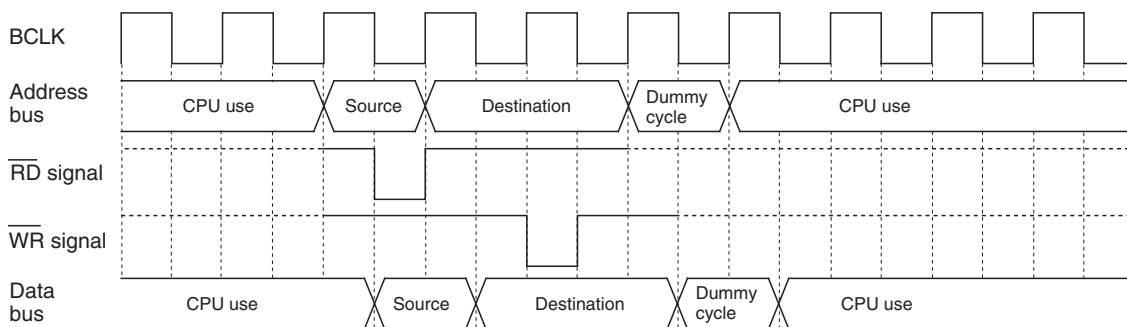
For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

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

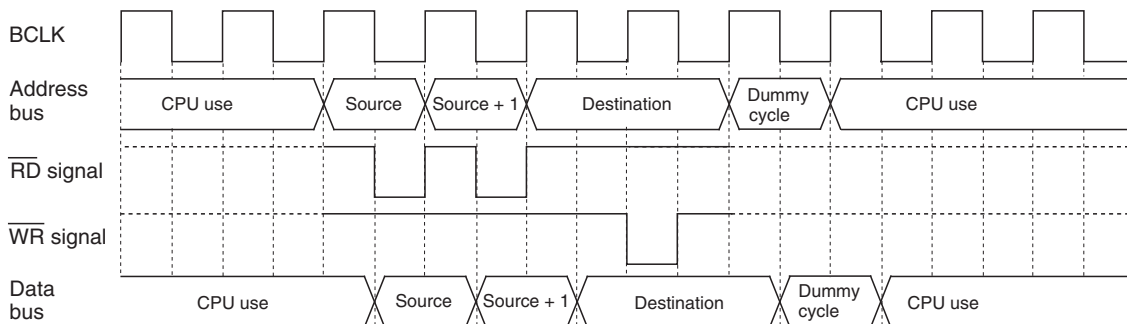
### 12.1.4 Effect of $\overline{\text{RDY}}$ Signal

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

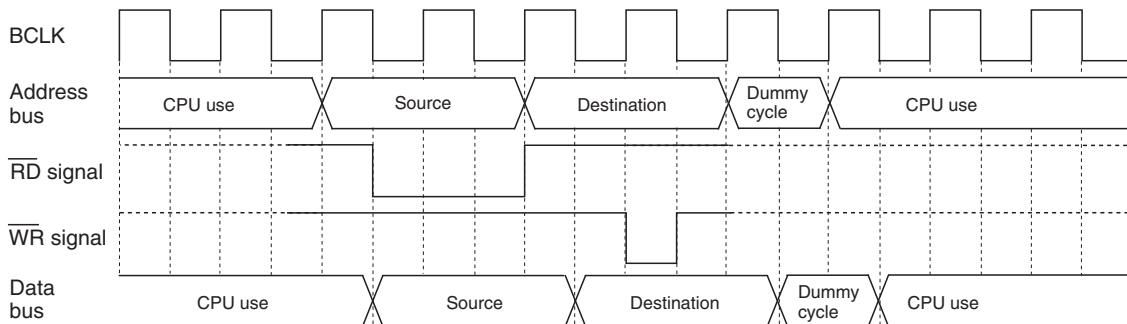
(1) When the transfer unit is 8 or 16 bits and the source of transfer is an even address



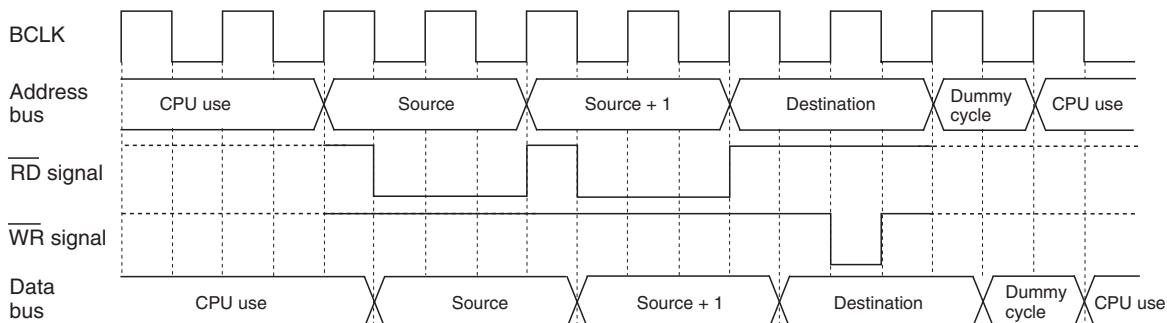
(2) When the transfer unit is 16 bits and the source address of transfer is an odd address, or when the transfer unit is 16 bits and an 8-bit bus is used



(3) When the source read cycle under condition (1) has one wait state inserted



(4) When the source read cycle under condition (2) has one wait state inserted



NOTE:

1. The same timing changes occur with the respective conditions at the destination as at the source.

Figure 12.5 Transfer Cycles for Source Read

## 12.2 DMA Transfer Cycles

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

Table 12.2 shows the number of DMA transfer cycles. Table 12.3 shows the coefficient j, k.

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

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

**Table 12.2 DMA Transfer Cycles**

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

-: This condition does not exist.

**Table 12.3 Coefficient j, k**

	Internal Area				External Area						
	Internal ROM, RAM		SFR		Separate Bus				Multiplexed Bus		
	No Wait	With Wait	1 Wait <sup>(1)</sup>	2 Waits <sup>(1)</sup>	No Wait	With Wait <sup>(2)</sup>			With Wait <sup>(2)</sup>		
						1 Wait	2 Waits	3 Waits	1 Wait	2 Waits	3 Waits
j	1	2	2	3	1	2	3	4	3	3	4
k	1	2	2	3	2	2	3	4	3	3	4

NOTES:

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



### 12.3 DMA Enable

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

- (1) Reload the forward address pointer with the SARi register value when the DSD bit in the DMiCON register is “1” (forward) or the DARI register value when the DAD bit in the DMiCON register is “1” (forward).
- (2) Reload the DMAi transfer counter with the DMAi transfer counter reload register value.

If the DMAE bit is set to “1” again while it remains set, the DMAC performs the above operation.

However, if a DMA request may occur simultaneously when the DMAE bit is being written, follow the steps below.

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

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

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

### 12.4 DMA Request

The DMAC can generate a DMA request as triggered by the 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 12.4 shows the timing at which the DMAS bit changes state.

Whenever a DMA request is generated, the DMAS bit is set to “1” (DMA requested) regardless of whether or not the DMAE bit is set. If the DMAE bit was set to “1” (enabled) when this occurred, the DMAS bit is set to “0” (DMA not requested) immediately before a data transfer starts. This bit cannot be set to “1” 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 12.4 Timing at Which DMAS bit Changes State**

DMA Factor	DMAS Bit in DMiCON Register	
	Timing at which the bit is set to “1”	Timing at which the bit is set to “0”
Software Trigger	When the DSR bit in the DMiSL register is set to “1”	<ul style="list-style-type: none"> <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”.	

$i = 0, 1$

## 12.5 Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of BCLK), the DMAS bit on each channel is set to “1” (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1.

The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period.

Figure 12.6 shows an example of DMA transfer effected by external factors.

In Figure 12.6, DMA0 request having priority is received first to start a transfer when a DMA0 request and DMA1 request are generated simultaneously. After one DMA0 transfer is completed, a bus arbitration is returned to the CPU. When the CPU has completed one bus access, a DMA1 transfer starts. After one DMA1 transfer is completed, the bus arbitration is again returned to the CPU.

In addition, DMA requests cannot be counted up since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in Figure 12.6, occurs more than one time, the DMAS bit is set to “0” as soon as getting the bus arbitration. The bus arbitration is returned to the CPU when one transfer is completed.

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

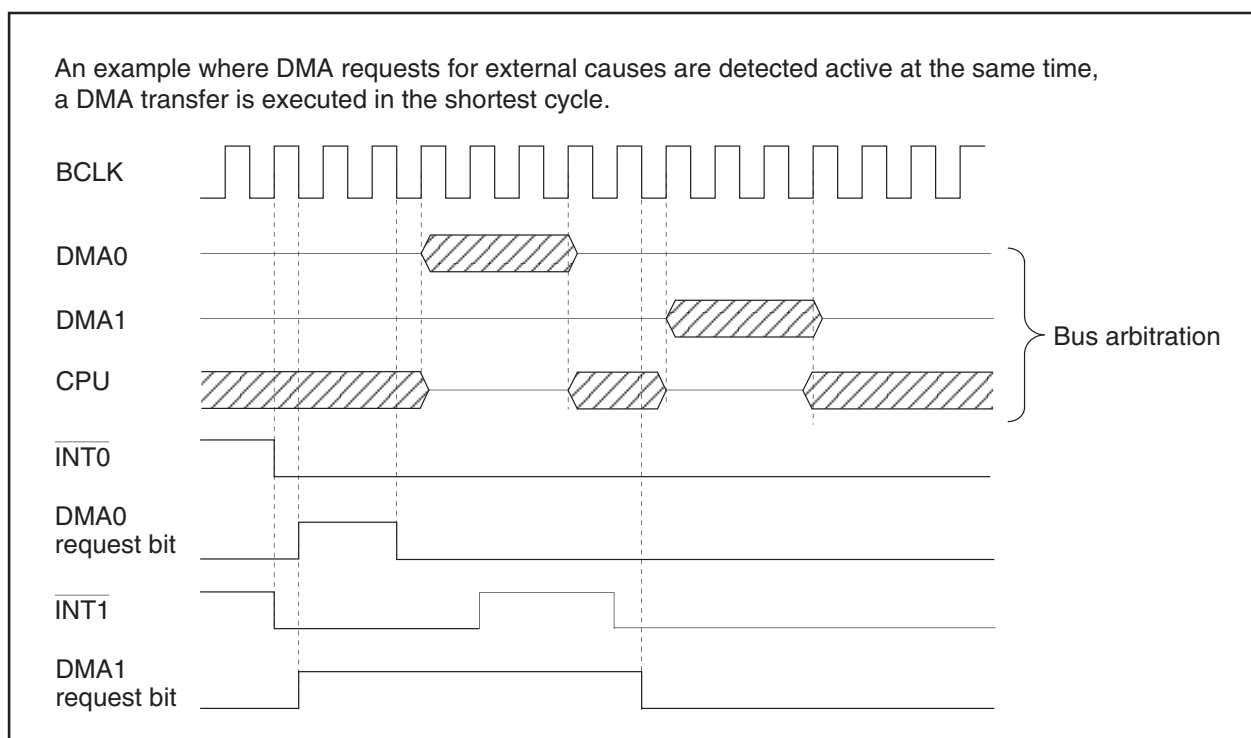
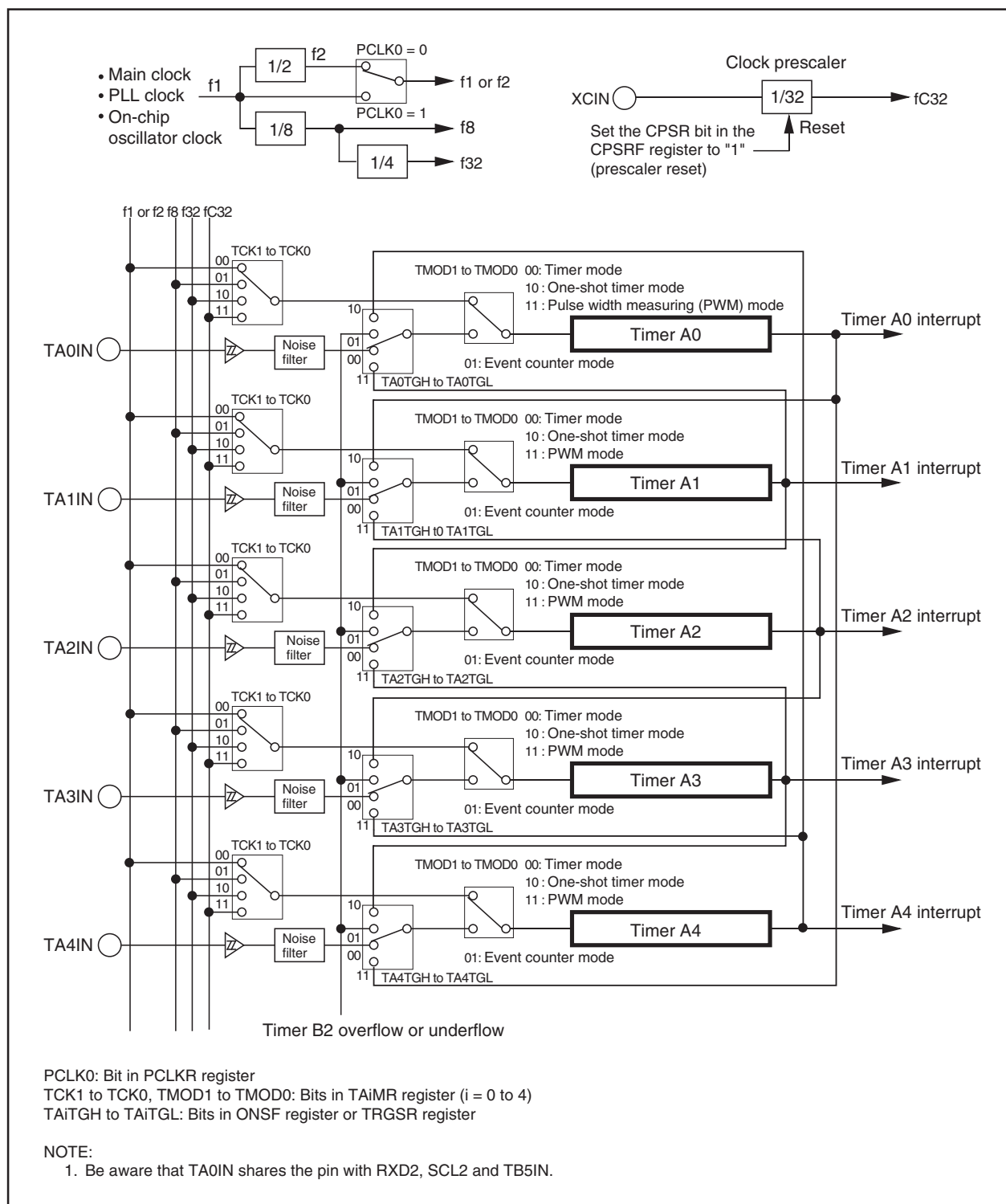


Figure 12.6 DMA Transfer by External Factors

## 13. Timers

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

Figures 13.1 and 13.2 show block diagrams of Timer A and Timer B configuration, respectively.



**Figure 13.1 Timer A Configuration**

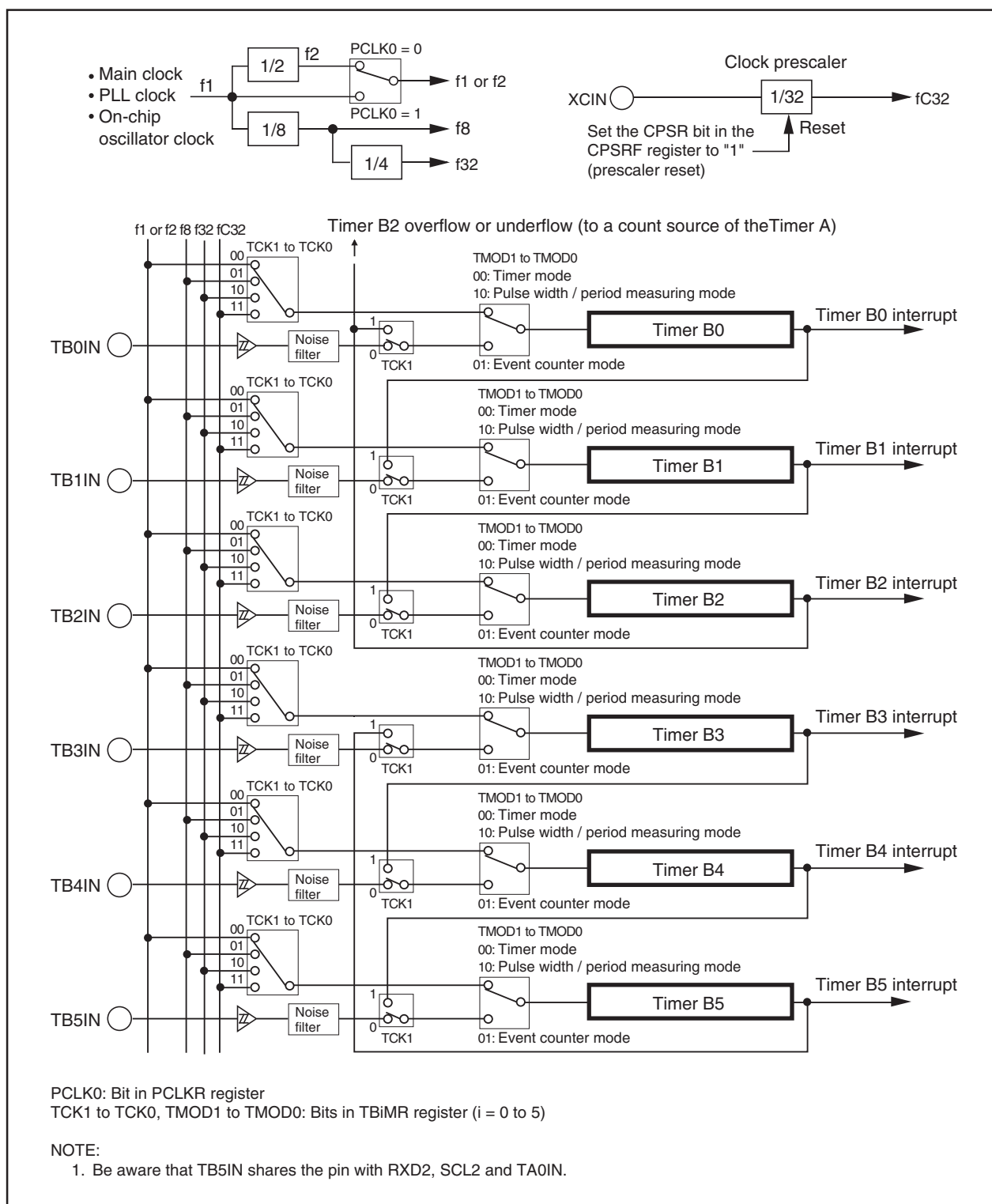
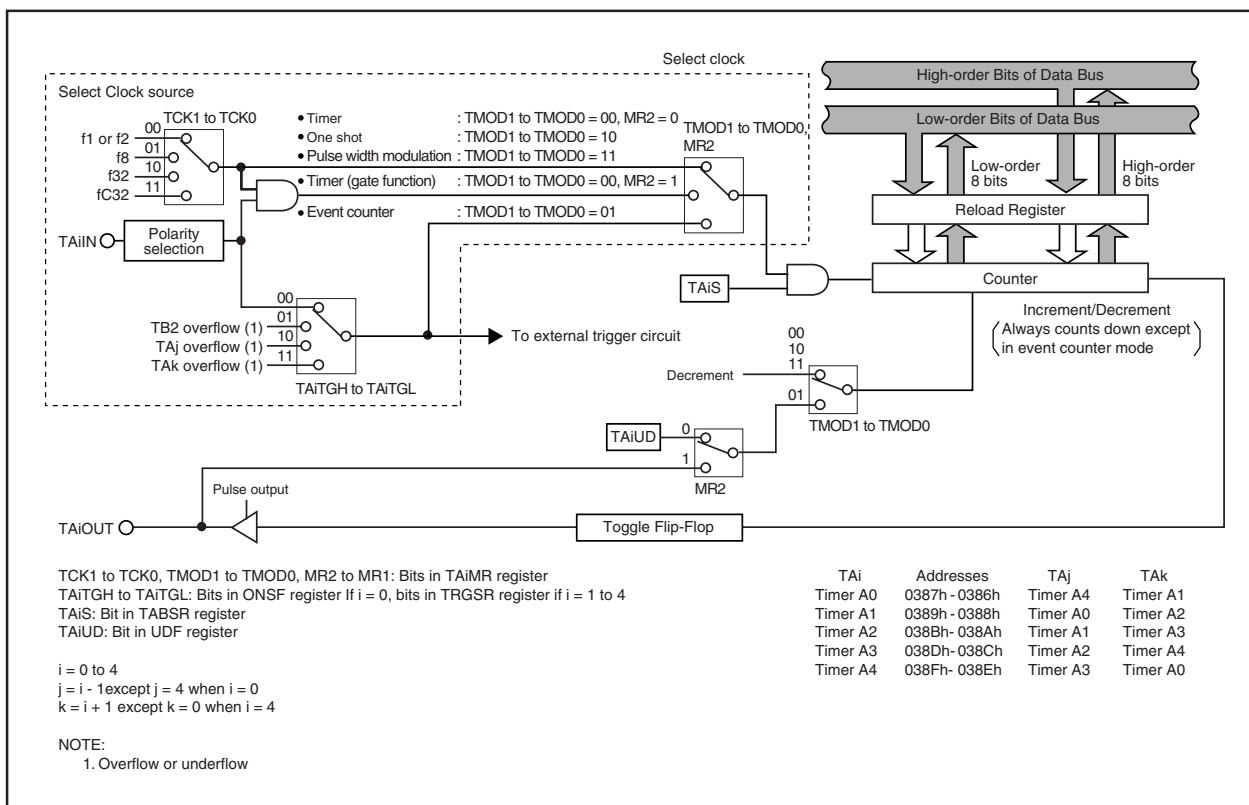


Figure 13.2 Timer B Configuration

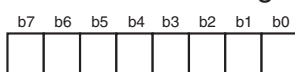
Figure 13.3 shows a block diagram of the timer A. Figures 13.4 to 13.6 show the timer A-related registers. The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use the TMOD1 to TMOD0 bits in the TAI<sub>i</sub>MR register (i = 0 to 4) to select the desired mode.

- **Timer mode:** The timer counts an internal count source.
- **Event counter mode:** The timer counts pulses from an external device or overflows and underflows of other timers.
- **One-shot timer mode:** The timer outputs a pulse only once before it reaches the minimum count “0000h.”
- **Pulse width modulation (PWM) mode:** The timer outputs pulses in a given width successively.



**Figure 13.3 Timer A Block Diagram**

## Timer Ai Mode Register (i = 0 to 4)

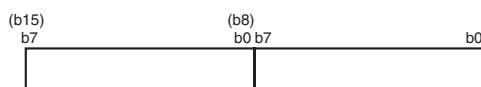


Symbol  
TA0MR to TA4MR

Address  
0396h to 039Ah

After Reset  
00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	<sup>b1 b0</sup> 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation mode	RW
TMOD1			RW
MR0	Function varies with each operation mode		RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count Source Select Bit	Function varies with each operation mode	RW
TCK1			RW

Timer Ai Register (i = 0 to 4) <sup>(1)</sup>

Symbol  
TA0  
TA1  
TA2  
TA3  
TA4

Address  
0387h to 0386h  
0389h to 0388h  
038Bh to 038Ah  
038Dh to 038Ch  
038Fh to 038Eh

After Reset  
Indeterminate  
Indeterminate  
Indeterminate  
Indeterminate  
Indeterminate

Mode	Function	Setting Range	RW
Timer Mode	Divide the count source by $n + 1$ where $n =$ set value	0000h to FFFFh	RW
Event Counter Mode	Divide the count source by $FFFFh - n + 1$ where $n =$ set value when counting up or by $n + 1$ when counting down <sup>(2)</sup>	0000h to FFFFh	RW
One-shot Timer Mode	Divide the count source by $n$ where $n =$ set value and cause the timer to stop	0000h to FFFFh <sup>(3) (4)</sup>	WO
Pulse Width Modulation Mode (16-bit PWM)	Modify the pulse width as follows: PWM period: $(2^{16} - 1) / f_j$ High level PWM pulse width: $n / f_j$ where $n =$ set value, $f_j =$ count source frequency	0000h to FFFEh <sup>(4) (5)</sup>	WO
Pulse Width Modulation Mode (8-bit PWM)	Modify the pulse width as follows: PWM period: $(2^8 - 1) \times (m + 1) / f_j$ High level PWM pulse width: $(m + 1)n / f_j$ where $n =$ high-order address set value, $m =$ low-order address set value, $f_j =$ count source frequency	00h to FEh (High-order address) 00h to FFh (Low-order address)	WO

## NOTES:

1. The register must be accessed in 16-bit unit.
2. The timer counts pulses from an external device or overflows or underflows in other timers.
3. If the TAI register is set to "0000h", the counter does not work and timer Ai interrupt requests are not generated either. Furthermore, if "pulse output" is selected, no pulses are output from the TAIOUT pin.
4. Use the MOV instruction to write to the TAI register.
5. If the TAI register is set to "0000h", the pulse width modulator does not work, the output level on the TAIOUT pin remains low, and timer Ai interrupt requests are not generated either.  
The same applies when the 8 high-order bits in the TAI register are set to "00h" while operating as an 8-bit pulse width modulator.

Figure 13.4 TA0MR to TA4MR Registers and TA0 to TA4 Registers

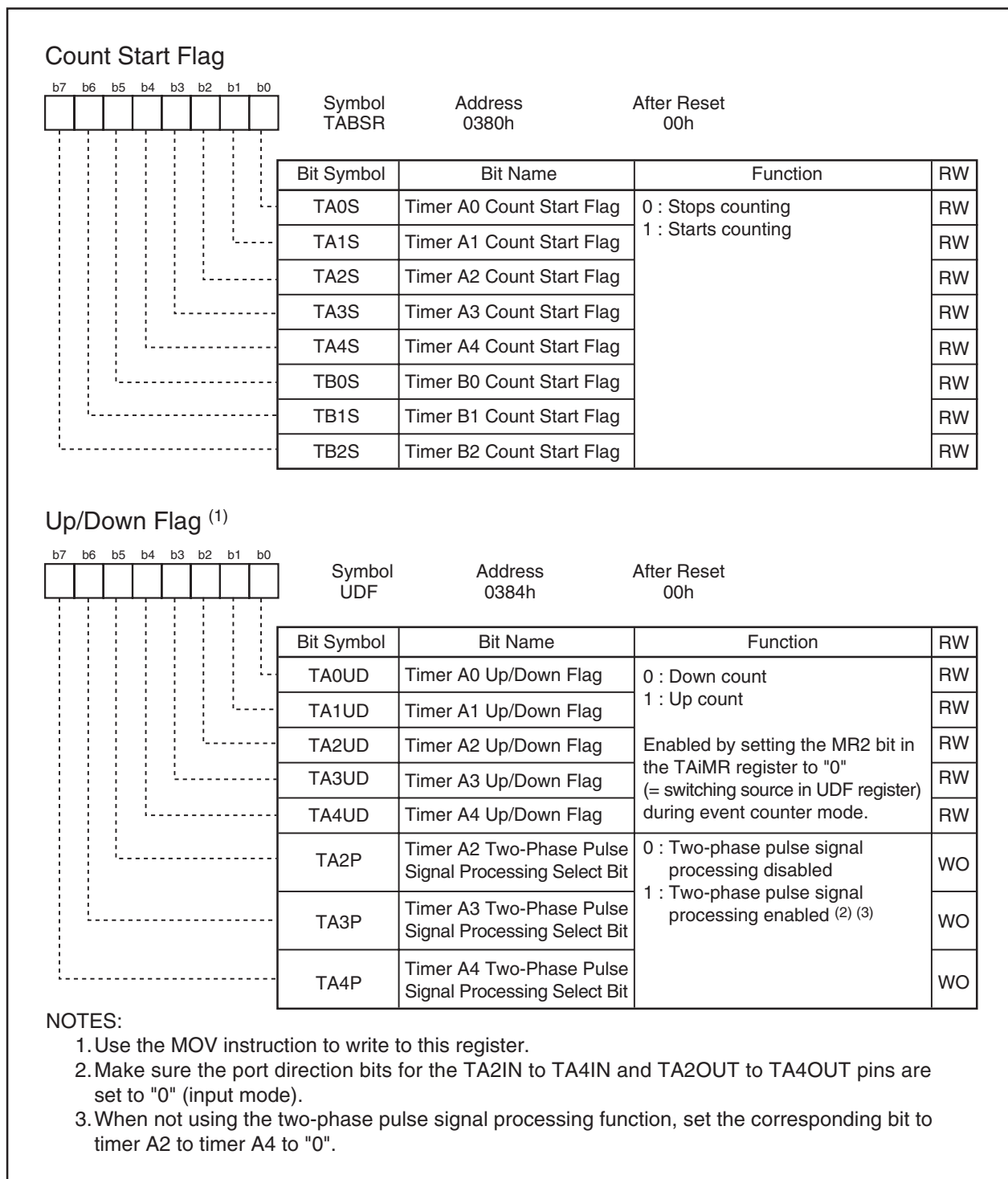


Figure 13.5 TABSR Register and UDF Register

## One-Shot Start Flag

Symbol  
ONSFAddress  
0382hAfter Reset  
00h

Bit Symbol	Bit Name	Function	RW
TA0OS	Timer A0 One-Shot Start Flag	The timer starts counting by setting this bit to "1" while the TMOD1 to TMOD0 bits in the TAIMR register (i = 0 to 4) = 10b (one-shot timer mode) and the MR2 bit in the TAIMR register = 0 (TAiOS bit enabled). When read, its content is "0".	RW
TA1OS	Timer A1 One-Shot Start Flag		RW
TA2OS	Timer A2 One-Shot Start Flag		RW
TA3OS	Timer A3 One-Shot Start Flag		RW
TA4OS	Timer A4 One-Shot Start Flag		RW
TAZIE	Z-phase Input Enable Bit	0 : Z-phase input disabled 1 : Z-phase input enabled	RW
TA0TGL	Timer A0 Event/Trigger Select Bit	b7 b6 0 0 : Input on TA0IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA4 is selected (2) 1 1 : TA1 is selected (2)	RW
TA0TGH			RW

## NOTES:

1. Make sure the PD7\_1 bit in the PD7 register is set to "0" (input mode).
2. Over flow or under flow.

## Trigger Select Register

Symbol  
TRGSRAddress  
0383hAfter Reset  
00h

Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 Event/Trigger Select Bit	b1 b0 0 0 : Input on TA1IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA0 is selected (2) 1 1 : TA2 is selected (2)	RW
TA1TGH			RW
TA2TGL	Timer A2 Event/Trigger Select Bit	b3 b2 0 0 : Input on TA2IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA1 is selected (2) 1 1 : TA3 is selected (2)	RW
TA2TGH			RW
TA3TGL	Timer A3 Event/Trigger Select Bit	b5 b4 0 0 : Input on TA3IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA2 is selected (2) 1 1 : TA4 is selected (2)	RW
TA3TGH			RW
TA4TGL	Timer A4 Event/Trigger Select Bit	b7 b6 0 0 : Input on TA4IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA3 is selected (2) 1 1 : TA0 is selected (2)	RW
TA4TGH			RW

## NOTES:

1. Make sure the port direction bits for the TA1IN to TA4IN pins are set to "0" (input mode).
2. Over flow or under flow.

## Clock Prescaler Reset Flag

Symbol  
CPSRFAddress  
0381hAfter Reset  
0XXXXXXXb

Bit Symbol	Bit Name	Function	RW
(b6-b0)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—
CPSR	Clock Prescaler Reset Flag	Setting this bit to "1" initializes the prescaler for the timekeeping clock. (When read, its content is "0".)	RW

Figure 13.6 ONSF Register, TRGSR Register and CPSRF Register



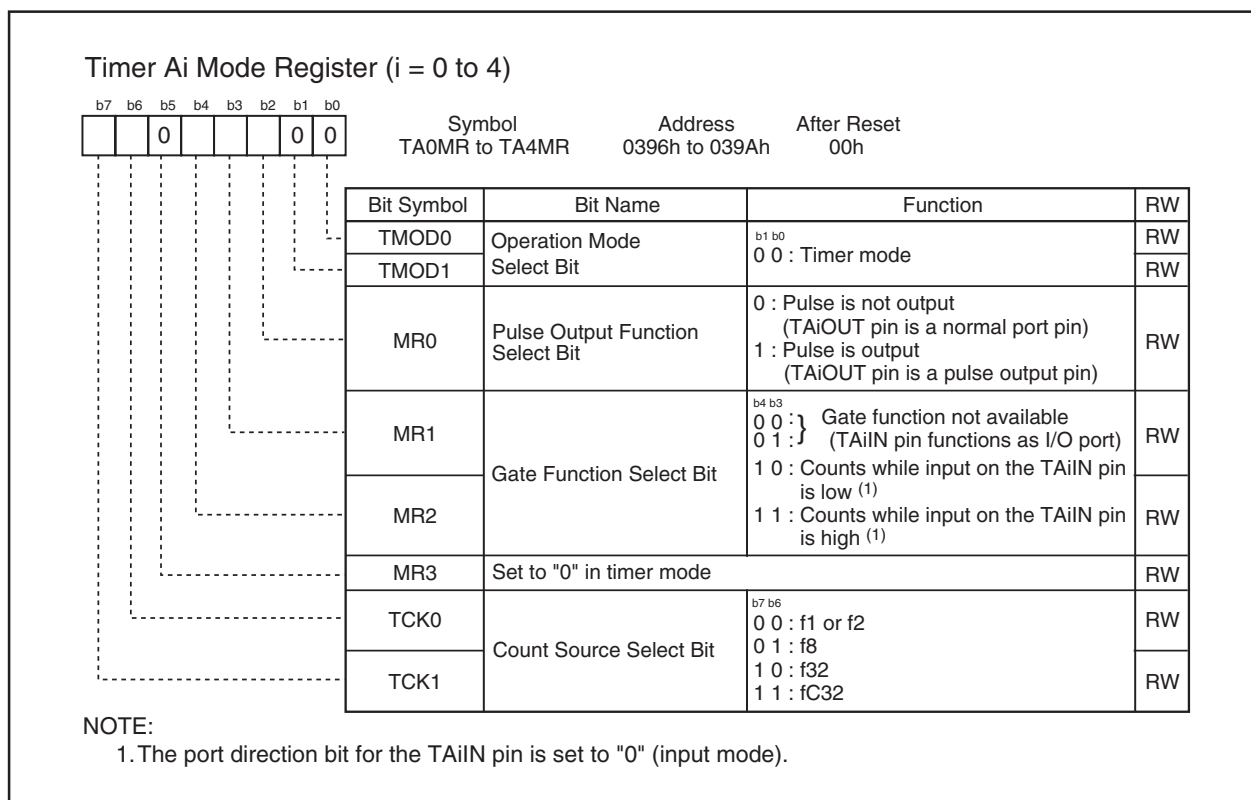
### 13.1.1 Timer Mode

In timer mode, the timer counts a count source generated internally. Table 13.1 lists specifications in timer mode. Figure 13.7 shows TAIiMR register in timer mode.

**Table 13.1 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 the TAI register 0000h to FFFFh
Count Start Condition	Set the TAIiS bit in the TABSR register to "1" (start counting)
Count Stop Condition	Set the TAIiS bit to "0" (stop counting)
Interrupt Request Generation Timing	Timer underflow
TAiIN Pin Function	I/O port or gate input
TAiOUT Pin Function	I/O port or pulse output
Read from Timer	Count value can be read by reading the TAI register
Write to Timer	<ul style="list-style-type: none"> <li>When not counting and until the 1st count source is input after counting start Value written to the TAI register is written to both reload register and counter</li> <li>When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select Function	<ul style="list-style-type: none"> <li>Gate function Counting can be started and stopped by an input signal to TAIiN pin</li> <li>Pulse output function Whenever the timer underflows, the output polarity of TAIOUT pin is inverted. When TAIiS bit is set to "0" (stop counting), the pin outputs a low.</li> </ul>

i = 0 to 4



**Figure 13.7 TA0MR to TA4MR Registers in Timer Mode**

### 13.1.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3 and A4 can count two-phase external signals. Table 13.2 lists specifications in event counter mode (when not processing two-phase pulse signal). Figure 13.8 shows TAI<sub>i</sub>MR register in event counter mode (when not processing two-phase pulse signal). Table 13.3 lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4). Figure 13.9 shows TA2MR to TA4MR registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

**Table 13.2 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>i</sub>N pin (effective edge can be selected in program)</li> <li>Timer B2 overflows or underflows,</li> <li>Timer A<sub>j</sub> overflows or underflows,</li> <li>Timer A<sub>k</sub> 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/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count    n : set value of the TAI register    0000h to FFFFh
Count Start Condition	Set the TAI <sub>i</sub> S bit in the TABSR register to "1" (start counting)
Count Stop Condition	Set the TAI <sub>i</sub> S bit to "0" (stop counting)
Interrupt Request Generation Timing	Timer overflow or underflow
TAI <sub>i</sub> N Pin Function	I/O port or count source input
TAI <sub>i</sub> OUT Pin Function	I/O port, pulse output, or up/down-count select input
Read from Timer	Count value can be read by reading the TAI register
Write to Timer	<ul style="list-style-type: none"> <li>When not counting and until the 1st count source is input after counting start Value written to the TAI register is written to both reload register and counter</li> <li>When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)</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>i</sub>OUT pin is inverted. When TAI<sub>i</sub>S bit is set to "0" (stop counting), the pin outputs a low.</li> </ul>

i = 0 to 4

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

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

Timer Ai Mode Register (i = 0 to 4)  
(When not using two-phase pulse signal processing)

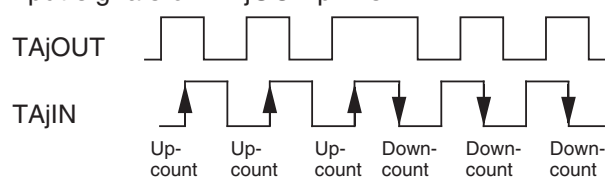
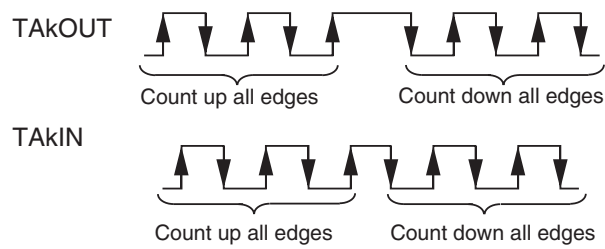
<div><div>b7b6b5b4b3b2b1b0</div><div><div></div><div></div><div>0</div><div></div><div></div><div></div><div>0</div><div>1</div></div></div>								Symbol TA0MR to TA4MR	Address 0396h to 039Ah	After Reset 00h	
								Bit Symbol	Bit Name	Function	RW
								TMOD0	Operation Mode Select Bit	b1 b0 0 1 : Event counter mode (1)	RW
								TMOD1			RW
								MR0	Pulse Output Function Select Bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (TAiOUT pin functions as pulse output pin)	RW
								MR1	Count Polarity Select Bit (2)	0 : Counts falling edge of external signal 1 : Counts rising edge of external signal	RW
								MR2	Up/Down Switching Cause Select Bit	0 : UDF register 1 : Input signal to TAiOUT pin (3)	RW
								MR3	Set to "0" in event counter mode		RW
								TCK0	Count Operation Type Select Bit	0 : Reload type 1 : Free-run type	RW
								TCK1	Can be "0" or "1" when not using two-phase pulse signal processing.		RW

## NOTES:

1. During event counter mode, the count source can be selected using the ONSF and TRGSR registers.
2. Effective when the TAIIGH and TAIIGL bits in the ONSF or TRGSR register are "00b" (TAiIN pin input).
3. Count down when input on TAiOUT pin is low or count up when input on that pin is high. The port direction bit for TAiOUT pin is set to "0" (input mode).

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

**Table 13.3 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 TAI <sub>i</sub> IN or TAI <sub>i</sub> OUT pins
Count Operation	• Up-count or down-count can be selected by two-phase pulse signal • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divide Ratio	1/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count    n : set value of the TAI register 0000h to FFFFh
Count Start Condition	Set the TAI <sub>i</sub> S bit in the TABSR register to "1" (start counting)
Count Stop Condition	Set the TAI <sub>i</sub> S bit to "0" (stop counting)
Interrupt Request Generation Timing	Timer overflow or underflow
TAI <sub>i</sub> IN Pin Function	Two-phase pulse input
TAI <sub>i</sub> OUT Pin Function	Two-phase pulse input
Read from Timer	Count value can be read by reading the TAI register
Write to Timer	• When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAI register is written to reload register (Transferred to counter when reloaded next)
Select Function <sup>(1)</sup>	<p>• Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on TAJ<sub>i</sub>IN pin when input signals on TAJ<sub>i</sub>OUT pin is "H".</p>  <p>• Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAK<sub>i</sub>IN pin goes "H" when the input signal on TAK<sub>i</sub>OUT pin is "H", the timer counts up rising and falling edges on TAK<sub>i</sub>OUT and TAK<sub>i</sub>IN pins. If the phase relationship is such that TAK<sub>i</sub>IN pin goes "L" when the input signal on TAK<sub>i</sub>OUT pin is "H", the timer counts down rising and falling edges on TAK<sub>i</sub>OUT and TAK<sub>i</sub>IN pins.</p>  <p>• Counter initialization by Z-phase input (timer A3) The timer count value is initialized to "0" by Z-phase input.</p>

i = 2 to 4

j = 2, 3

k = 3, 4

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.

Timer Ai Mode Register (i = 2 to 4)  
(When using two-phase pulse signal processing)

b6	b5	b4	b3	b2	b1	b0	Symbol TA2MR to TA4MR	Address 0398h to 039Ah	After Reset 00h
		0	1	0	0	0			
							Bit Symbol	Bit Name	Function
							TMOD0	Operation Mode Select Bit	b1 b0 0 1 : Event counter mode
							TMOD1		
							MR0	To use two-phase pulse signal processing, set this bit to "0".	RW
							MR1		
							MR2	To use two-phase pulse signal processing, set this bit to "1".	RW
							MR3	To use two-phase pulse signal processing, set this bit to "0".	RW
							TCK0	Count Operation Type Select Bit	0 : Reload type 1 : Free-run type
							TCK1	Two-Phase Pulse Signal Processing Operation Select Bit <sup>(1) (2)</sup>	0 : Normal processing operation 1 : Multiply-by-4 processing operation

## NOTES:

1. The TCK1 bit is valid for the TA3MR register. No matter how this bit is set, timers A2 and A4 always operate in normal processing mode and x4 processing mode, respectively.
2. If two-phase pulse signal processing is desired, following register settings are required:
  - Set the TAiP bit in the UDF register to "1" (two-phase pulse signal processing function enabled).
  - Set the TAiTGH and TAiTGL bits in the TRGSR register to "00b" (TAiIN pin input).
  - Set the port direction bits for TAiIN and TAiOUT to "0" (input mode).

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

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

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

This function can only be used in timer A3 event counter mode during two-phase pulse signal processing, free-running type, x4 processing, with Z-phase entered from the ZP pin.

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

Counter initialization is accomplished by detecting Z-phase input edge. The active edge can be selected to be the rising or falling edge by using the POL bit in the INT2IC register. The Z-phase pulse width applied to the INT2 pin must be equal to or greater than one clock cycle of the timer A3 count source. The counter is initialized at the next count timing after recognizing Z-phase input. Figure 13.10 shows the relationship between the two-phase pulse (A phase and B phase) and the Z-phase.

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

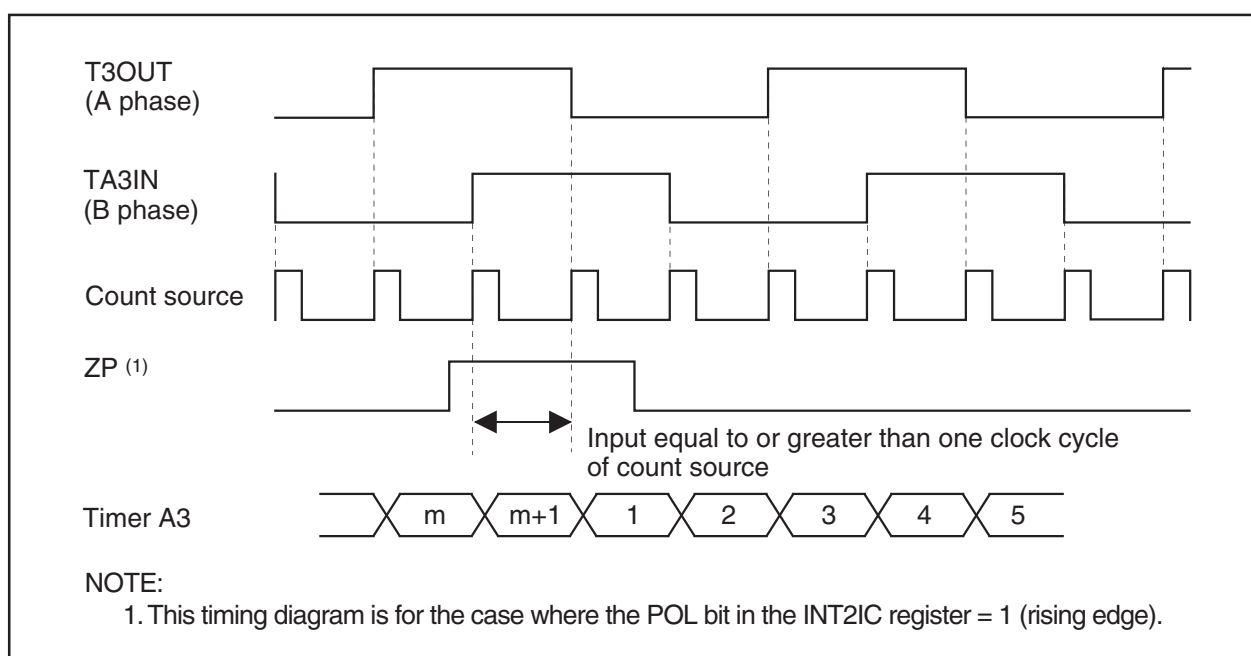


Figure 13.10 Two-phase Pulse (A phase and B phase) and Z Phase

### 13.1.3 One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. When the trigger occurs, the timer starts up and continues operating for a given period. Table 13.4 lists specifications in one-shot timer mode. Figure 13.11 shows the TAI<sub>MR</sub> register in the one-shot timer mode.

**Table 13.4 Specifications in One-shot 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 counter reaches 0000h, 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 the TAI register    0000h to FFFFh However, the counter does not work if the divide-by-n value is set to 0000h.
Count Start Condition	<p>The TAI<sub>S</sub> bit in the TABSR register = 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> overflow or underflow, Timer A<sub>k</sub> overflow or underflow</li> <li>• The TAI<sub>OS</sub> 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 “0000h”</li> <li>• TAI<sub>S</sub> bit is set to “0” (stop counting)</li> </ul>
Interrupt Request Generation Timing	When the counter reaches “0000h”
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 the 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 the TAI register is written to both reload register and counter</li> <li>• When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select Function	<ul style="list-style-type: none"> <li>• Pulse output function</li> </ul> <p>The timer outputs a low when not counting and a high when counting.</p>

i = 0 to 4

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

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

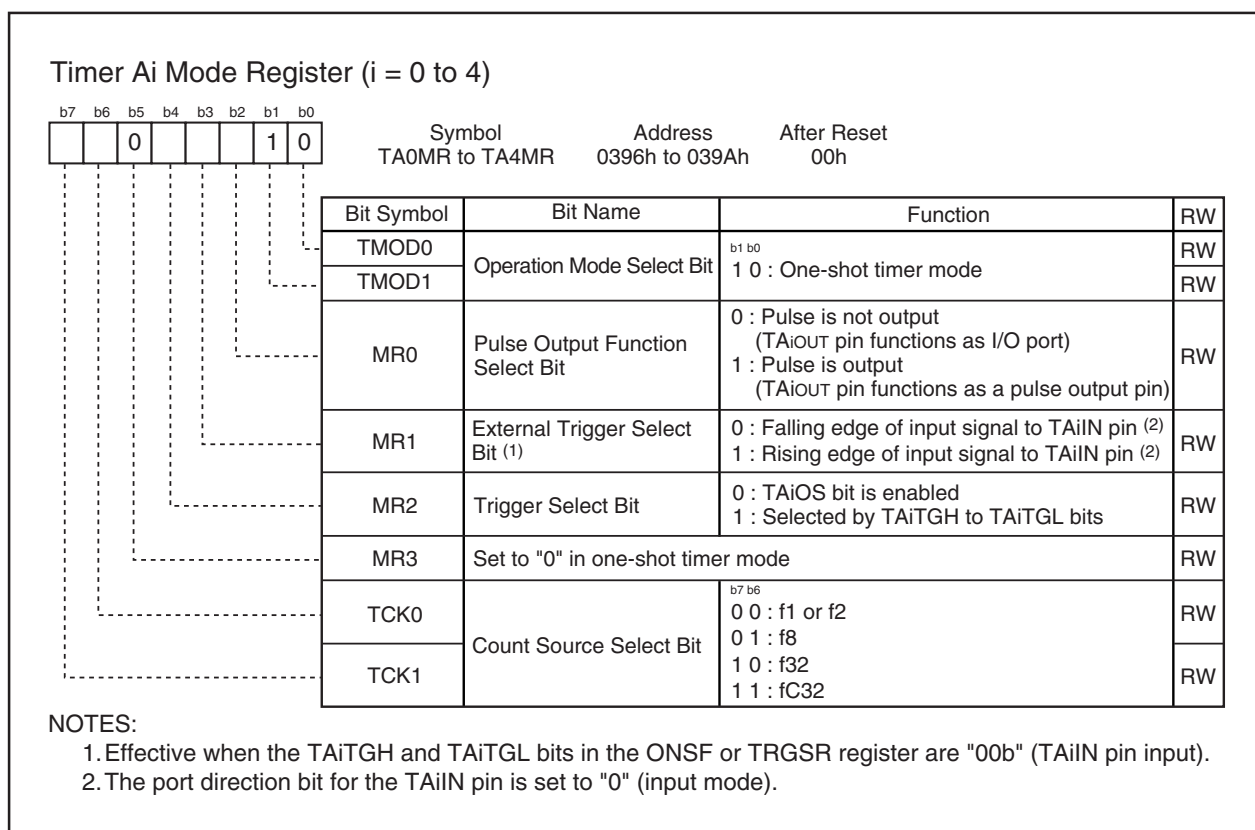


Figure 13.11 TAIiMR Register in One-shot Timer Mode



### 13.1.4 Pulse Width Modulation (PWM) Mode

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

Table 13.5 lists specifications in PWM mode. Figure 13.12 shows TAI<sub>MR</sub> register in PWM mode.

Figures 13.13 and 13.14 show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates, respectively.

**Table 13.5 Specifications in PWM Mode**

Item	Specification
Count Source	f1, f2, f8, f32, fC32
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 the TAI register</li> <li>• Cycle time <math>(2^{16}-1) / f_j</math> fixed <math>f_j</math> : count source frequency (f1, f2, f8, f32, fC32)</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 the 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 the TAI register low-order address</li> </ul>
Count Start Condition	<ul style="list-style-type: none"> <li>• The TAI<sub>S</sub> bit in the TABSR register is set to "1" (start counting)</li> <li>• The TAI<sub>S</sub> bit = 1 and external trigger input from the TAI<sub>IN</sub> pin</li> <li>• The TAI<sub>S</sub> bit = 1 and one of the following external triggers occurs Timer B2 overflow or underflow, Timer A<sub>j</sub> overflow or underflow, Timer A<sub>k</sub> overflow or underflow</li> </ul>
Count Stop Condition	The TAI <sub>S</sub> bit is set to "0" (stop counting)
Interrupt Request Generation Timing	On the falling edge of the PWM pulse
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 the 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 the TAI register is written to both reload register and counter</li> <li>• When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>

$i = 0$  to 4

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

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

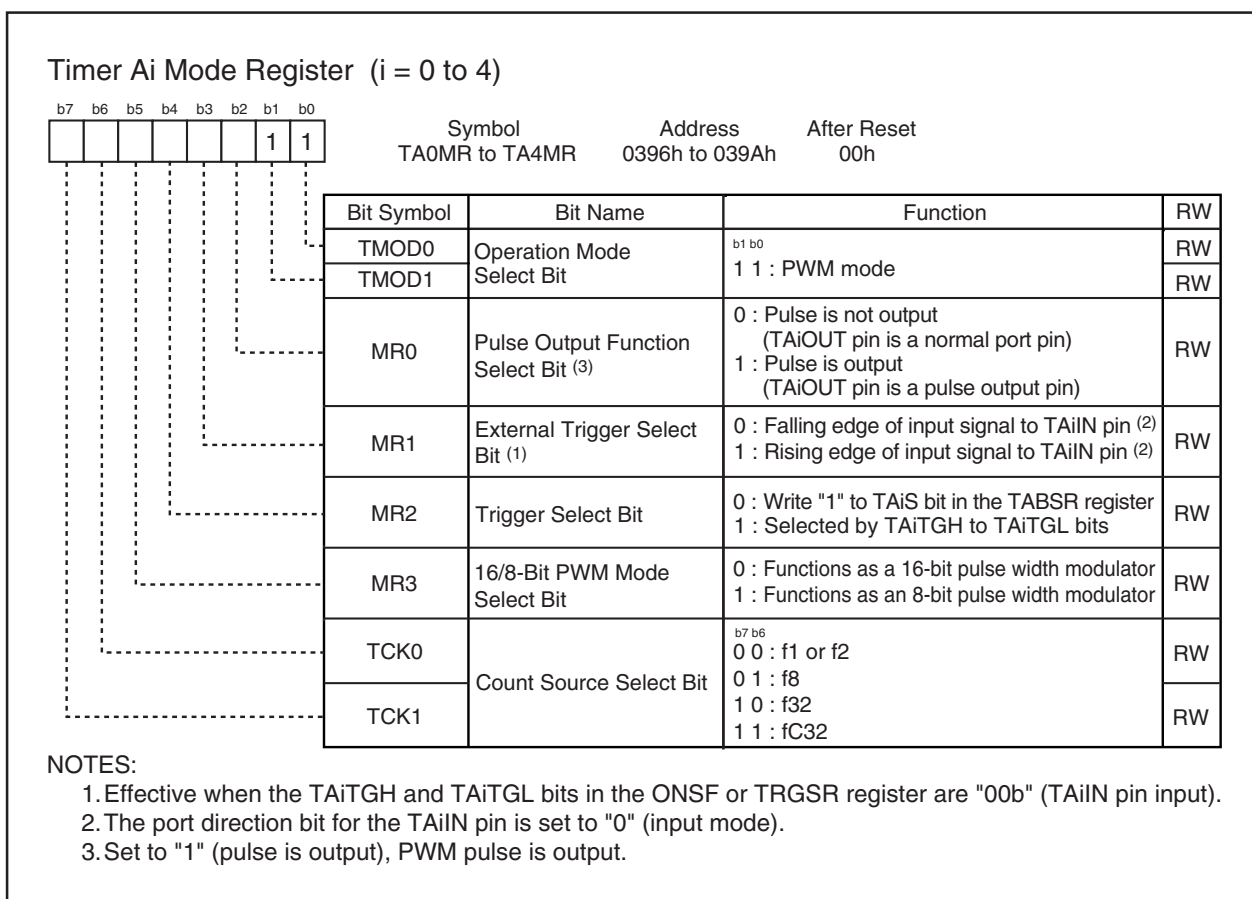


Figure 13.12 TA0MR to TA4MR Registers in PWM Mode

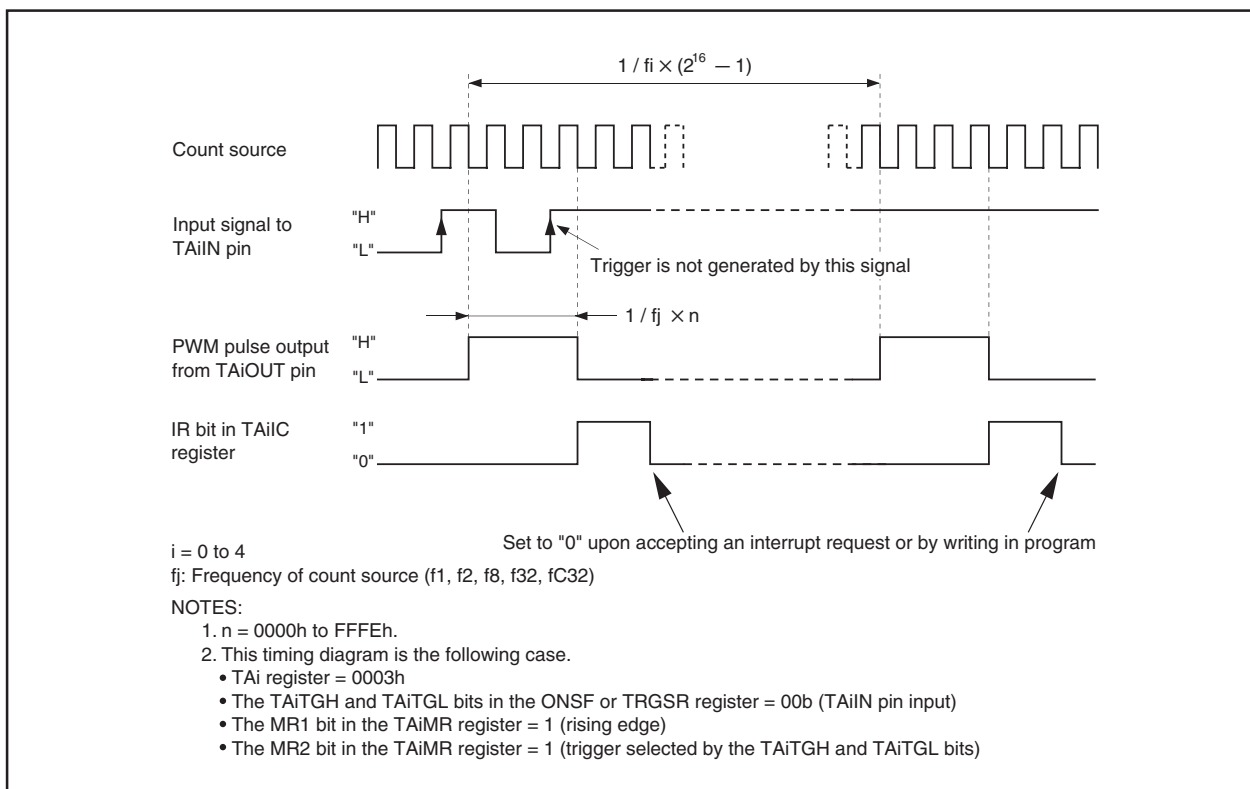


Figure 13.13 Example of 16-bit Pulse Width Modulator Operation

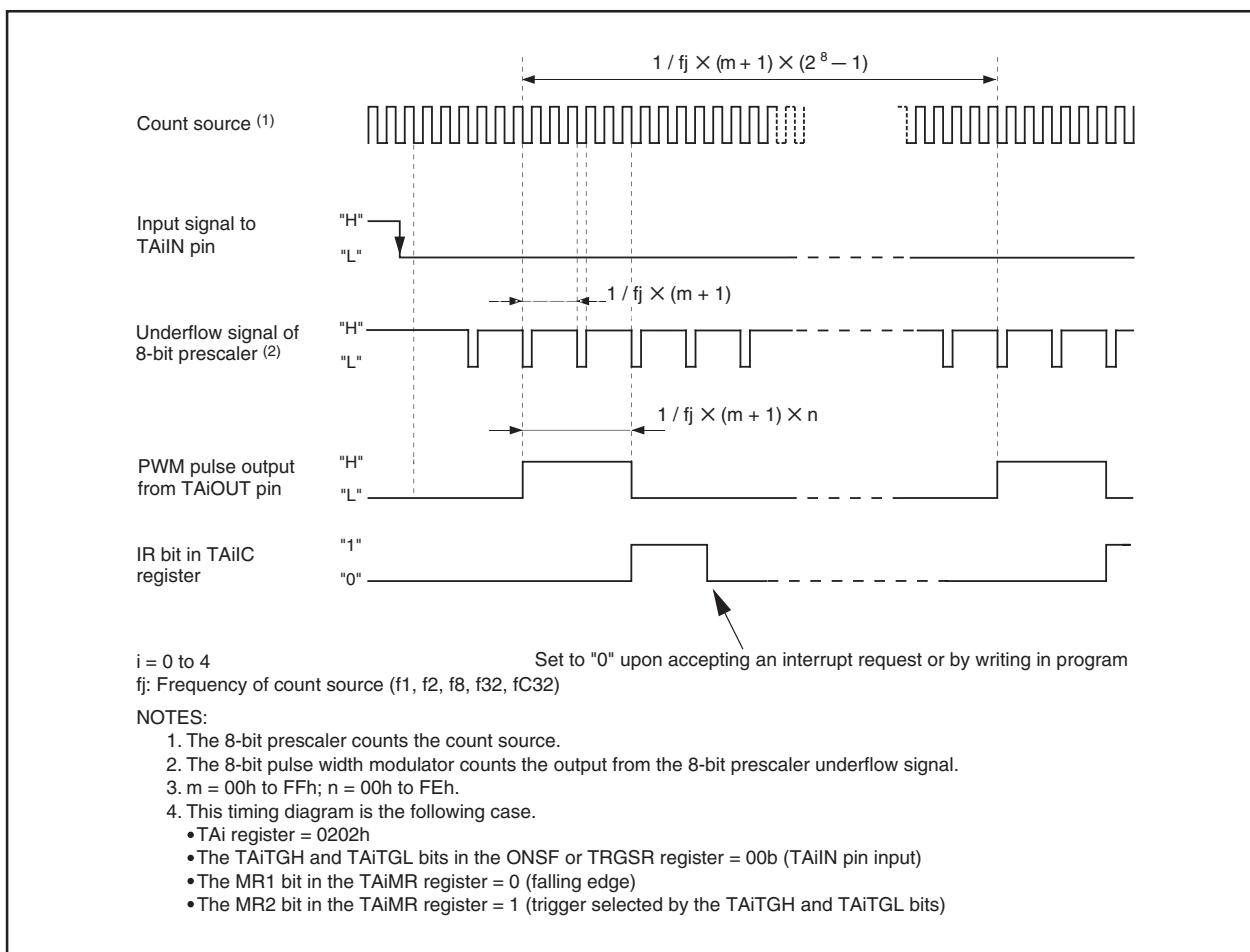


Figure 13.14 Example of 8-bit Pulse Width Modulator Operation

## 13.2 Timer B

Figure 13.15 shows a block diagram of the timer B. Figures 13.16 and 13.17 show the timer B-related registers.

Timer B supports the following three modes. Use the TMOD1 and TMOD0 bits in the TBiMR register (i = 0 to 5) 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 pulse period or pulse width of an external signal.

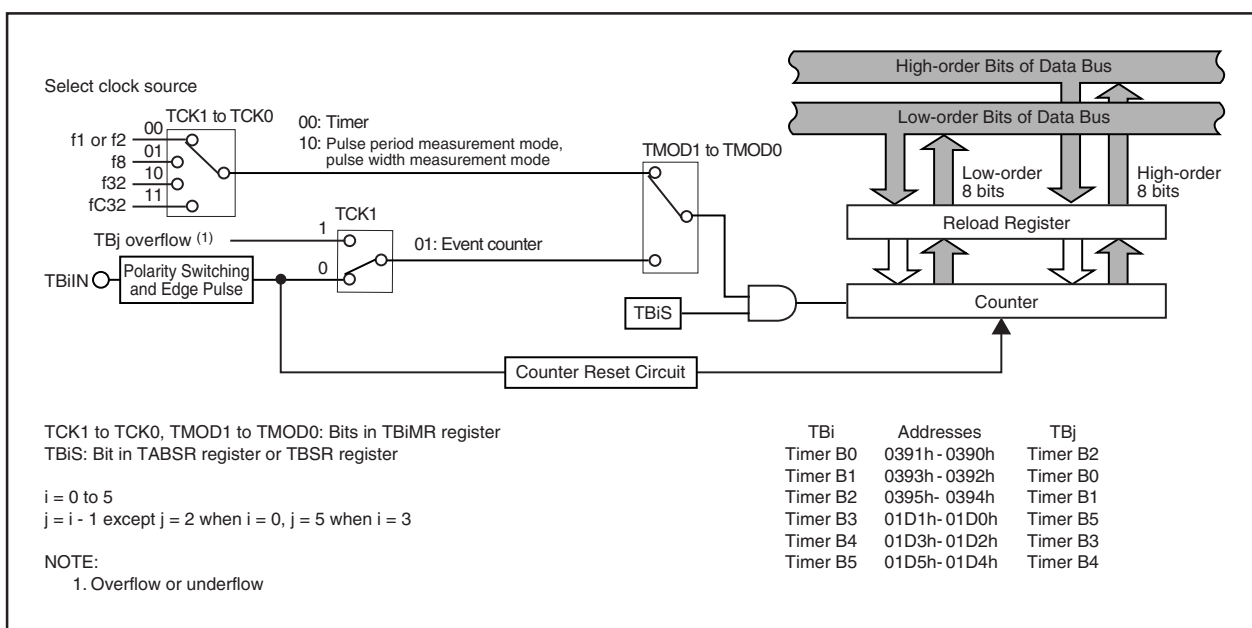


Figure 13.15 Timer B Block Diagram

## Timer Bi Mode Register (i = 0 to 5)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
								TB0MR to TB2MR	039Bh to 039Dh	00XX0000b
								TB3MR to TB5MR	01DBh to 01DDh	00XX0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 0 0 : Timer mode	RW
TMOD1		0 1 : Event counter mode	RW
		1 0 : Pulse period measurement mode, pulse width measurement mode	
		1 1 : Do not set a value	
MR0	Function varies with each operation mode		RW
MR1			RW
MR2			RW (1)
MR3			RO
TCK0	Count Source Select Bit	Function varies with each operation mode	RW
TCK1			RW

## NOTES:

1. Timer B0, timer B3.
2. Timer B1, timer B2, timer B4, timer B5.

## Timer Bi Register (i = 0 to 5) (1)

(b15) b7	(b8) b0 b7	b0	Symbol	Address	After Reset
			TB0	0391h, 0390h	Indeterminate
			TB1	0393h, 0392h	Indeterminate
			TB2	0395h, 0394h	Indeterminate
			TB3	01D1h, 01D0h	Indeterminate
			TB4	01D3h, 01D2h	Indeterminate
			TB5	01D5h, 01D4h	Indeterminate

Mode	Function	Setting Range	RW
Timer Mode	Divide the count source by n + 1 where n = set value	0000h to FFFFh	RW
Event Counter Mode	Divide the count source by n + 1 where n = set value (2)	0000h to FFFFh	RW
Pulse Period Modulation Mode, Pulse Width Modulation Mode	Measures a pulse period or width	—	RO

## NOTES:

1. The register must be accessed in 16-bit unit.
2. The timer counts pulses from an external device or overflows or underflows of other timers.

Figure 13.16 TB0MR to TB5MR Registers and TB0 to TB5 Registers

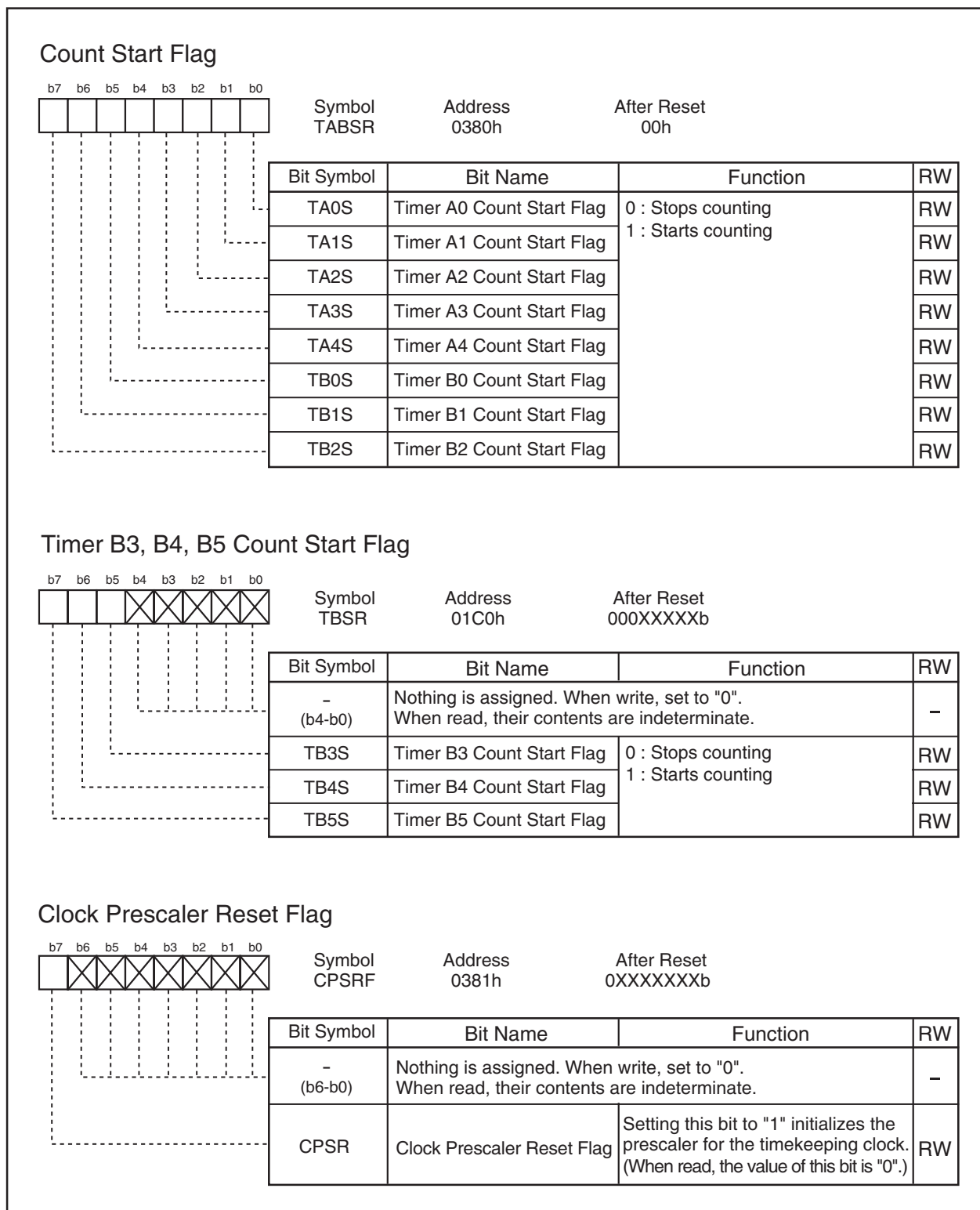


Figure 13.17 TABSR Register, TBSR Register and CPSRF Register

### 13.2.1 Timer Mode

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

Table 13.6 lists specifications in timer mode. Figure 13.18 shows TBiMR register in timer mode.

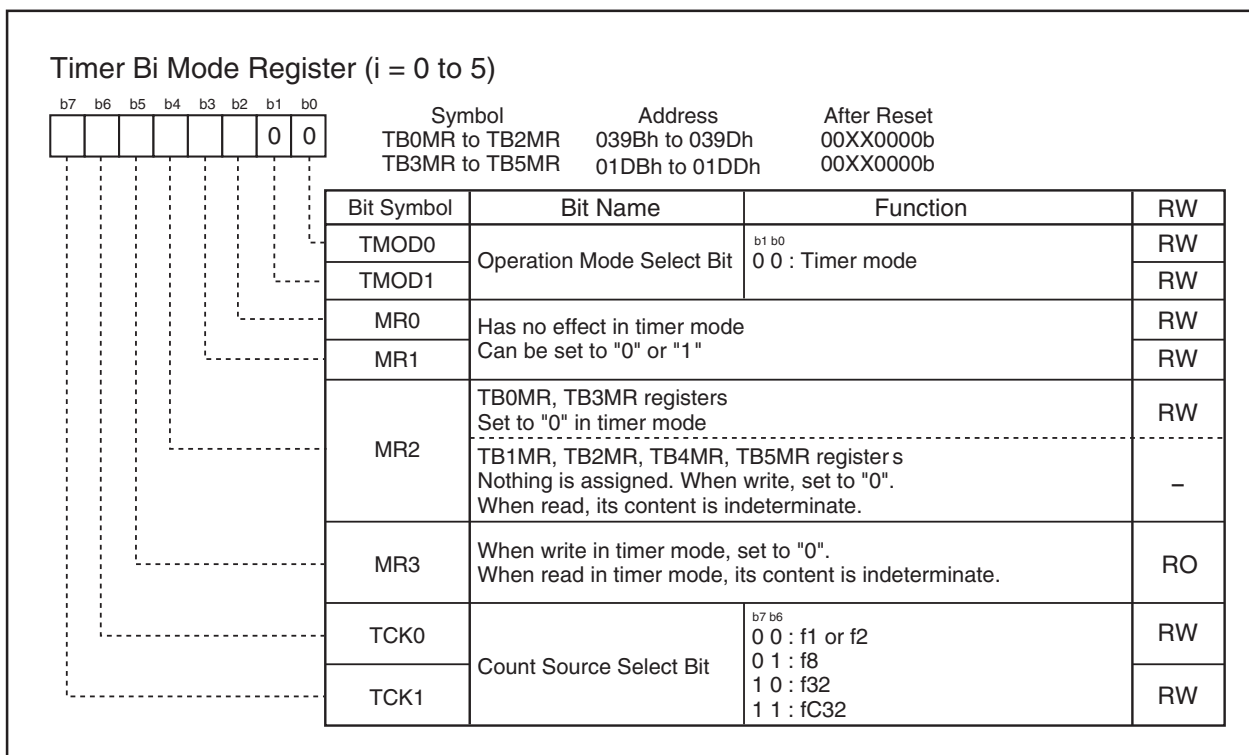
**Table 13.6 Specifications in Timer Mode**

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 the TBi register 0000h to FFFFh
Count Start Condition	Set the TBiS bit <sup>(1)</sup> to "1" (start counting)
Count Stop Condition	Set the 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 the 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 the TBi register is written to both reload register and counter</li> <li>When counting (after 1st count source input) Value written to the TBi register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>

i = 0 to 5

NOTE:

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



**Figure 13.18 TB0MR to TB5MR Registers in Timer Mode**

### 13.2.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Table 13.7 lists specifications in event counter mode. Figure 13.19 shows TBiMR register in event counter mode.

**Table 13.7 Specifications in Event Counter Mode**

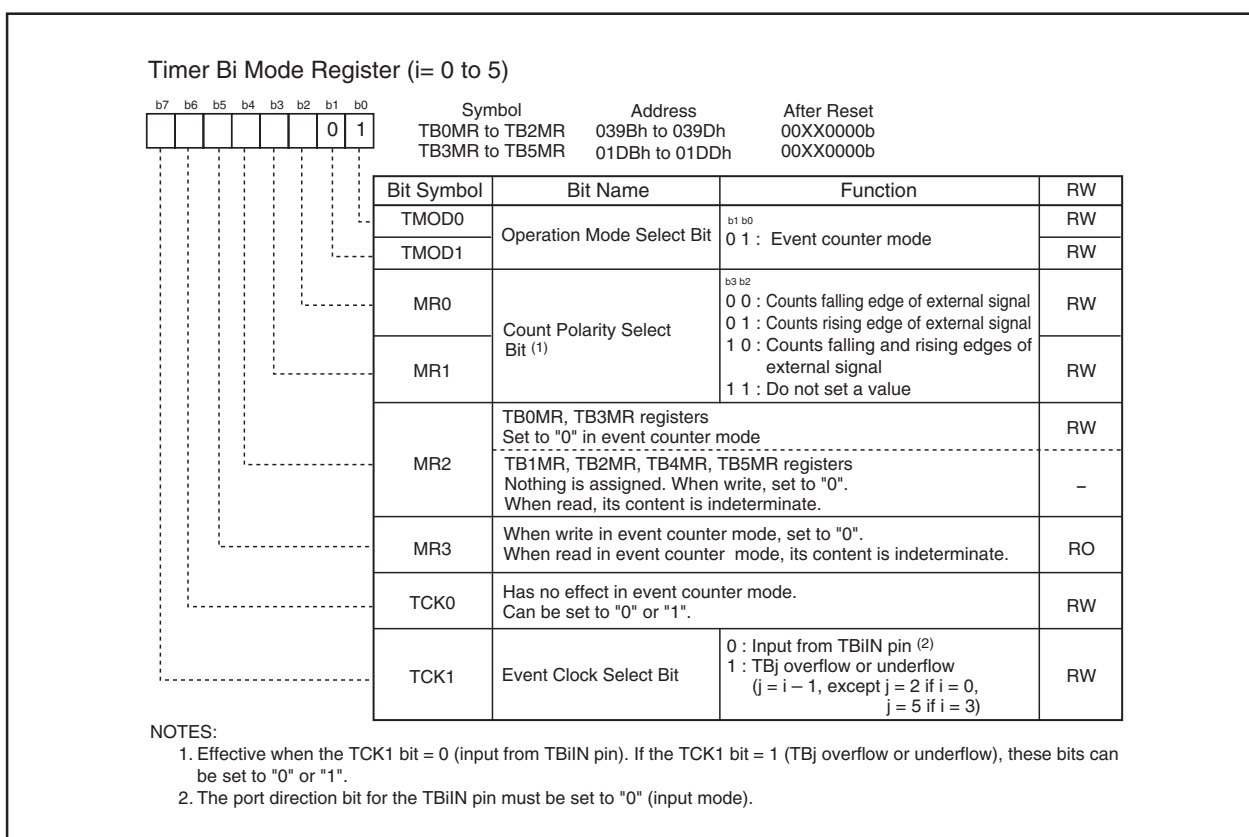
Item	Specification
Count Source	<ul style="list-style-type: none"> <li>External signals input to TBiIN pin (effective edge can be selected in program)</li> <li>Timer Bj overflow or underflow</li> </ul>
Count Operation	<ul style="list-style-type: none"> <li>Down-count</li> <li>When the timer underflows, it reloads the reload register contents and continues counting</li> </ul>
Divide Ratio	$1/(n+1)$ n: set value of the TBi register 0000h to FFFFh
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 the 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 the TBi register is written to both reload register and counter</li> <li>When counting (after 1st count source input) Value written to the TBi register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>

i = 0 to 5

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

NOTE:

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



**Figure 13.19 TB0MR to TB5MR Registers in Event Counter Mode**



### 13.2.3 Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal. Table 13.8 lists specifications in pulse period and pulse width measurement mode. Figure 13.20 shows TBiMR register in pulse period and pulse width measurement mode. Figure 13.21 shows the operation timing when measuring a pulse period. Figure 13.22 shows the operation timing when measuring a pulse width.

**Table 13.8 Specifications in Pulse Period and Pulse Width Measurement Mode**

Item	Specification
Count Source Count Operation	f1, f2, f8, f32, fC32 <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 "0000h" to continue counting.</li> </ul>
Count Start Condition	Set the TBiS bit <sup>(1)</sup> to "1" (start counting)
Count Stop Condition	Set the 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>(2)</sup></li> <li>• Timer overflow. When an overflow occurs, the MR3 bit in the TBiMR register is set to "1" (overflow) simultaneously. The MR3 bit is set to "0" (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to "1". At this time, make sure the TBiS bit is set to "1" (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>(3)</sup>
Write to Timer	Value written to the TBi register is written to neither reload register nor counter

i = 0 to 5

**NOTES:**

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

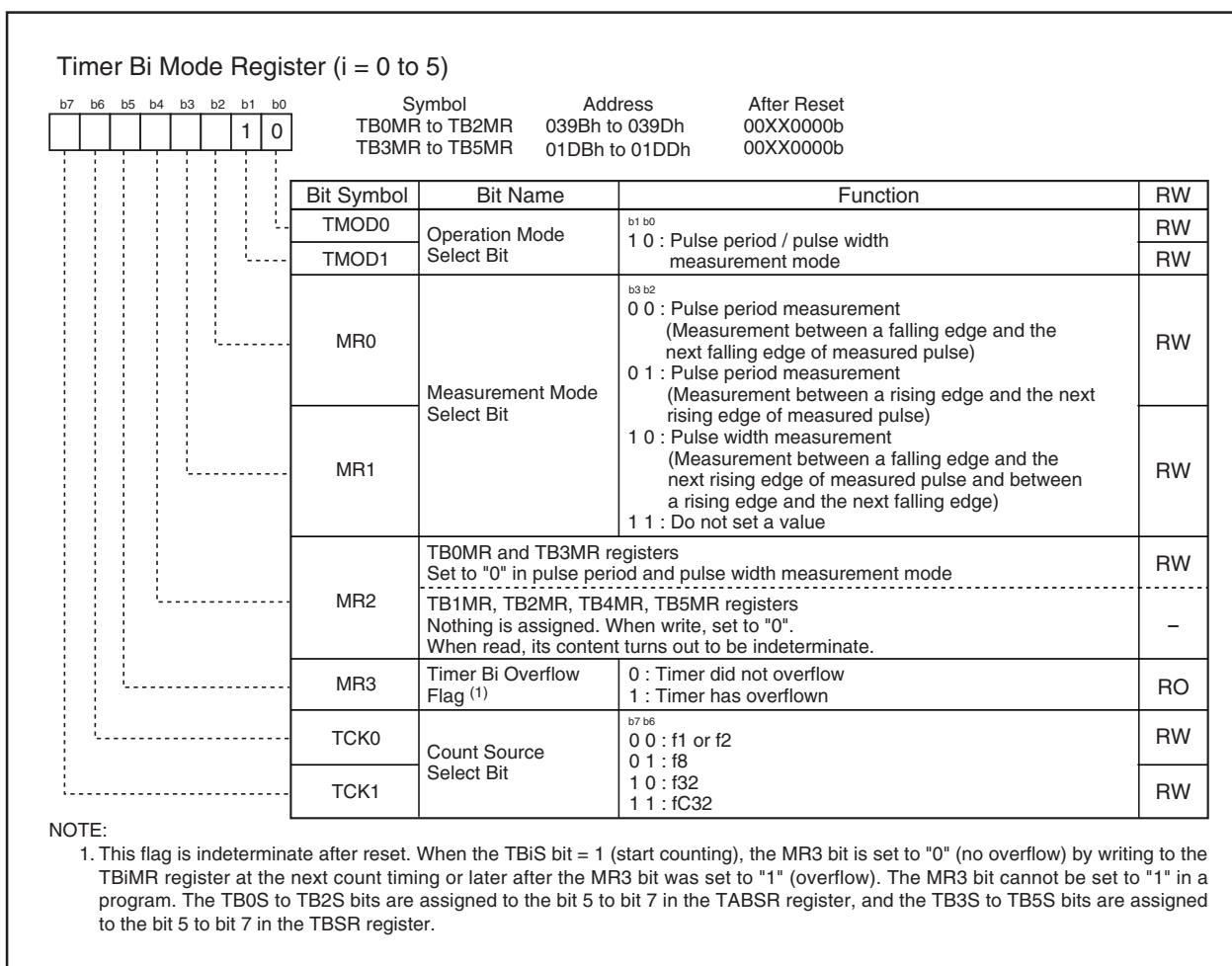


Figure 13.20 TB0MR to TB5MR Registers in Pulse Period and Pulse Width Measurement Mode

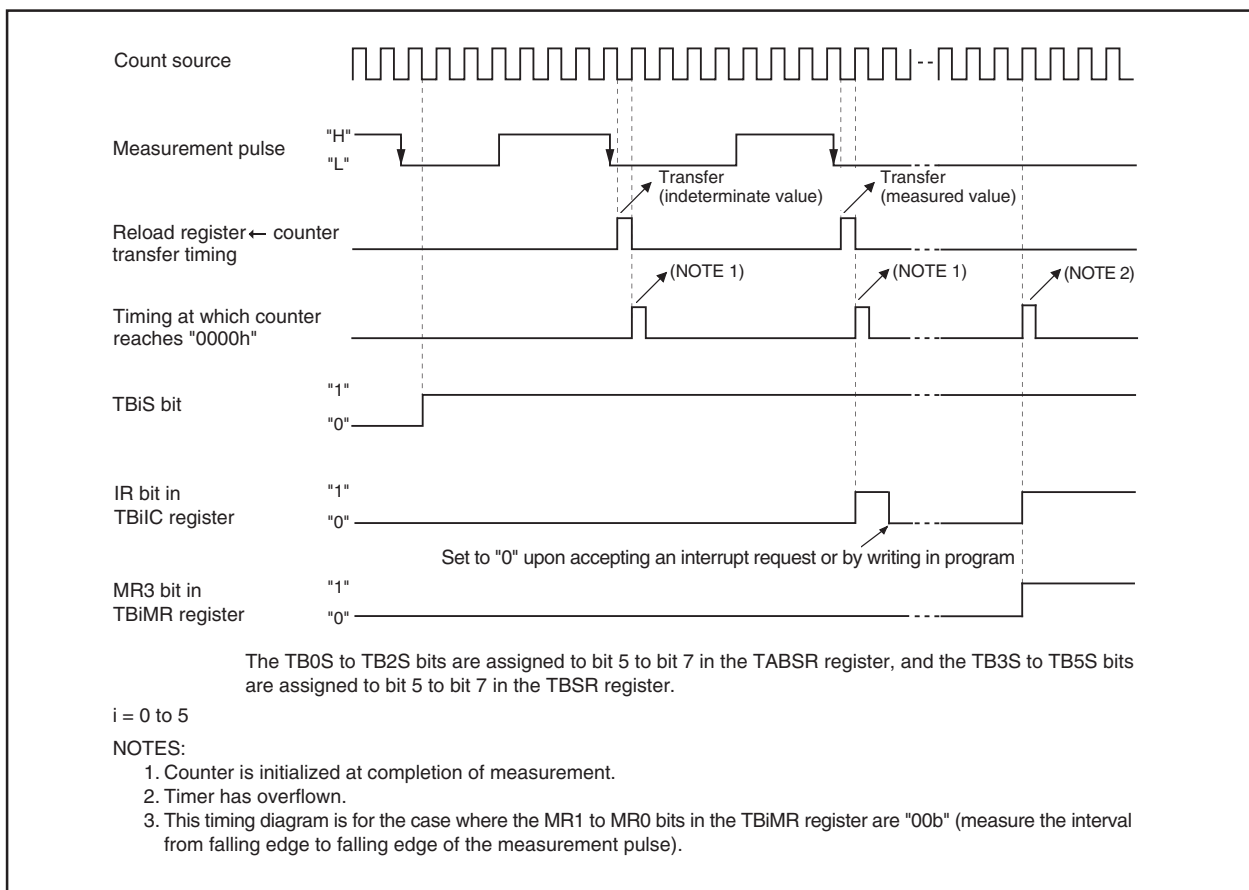


Figure 13.21 Operation Timing When Measuring Pulse Period

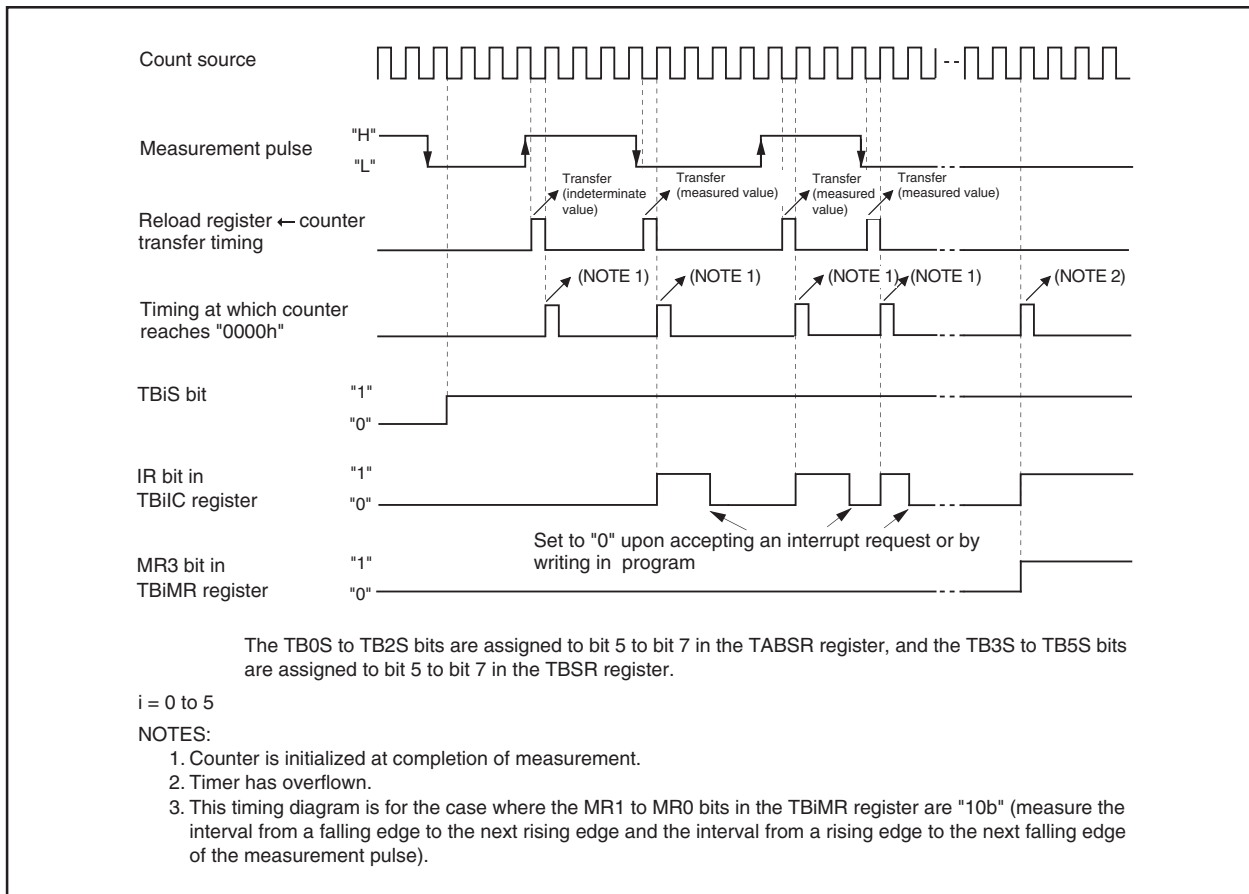


Figure 13.22 Operation Timing When Measuring Pulse Width

## 14. Three-Phase Motor Control Timer Function

Timers A1, A2, A4 and B2 can be used to output three-phase motor drive waveforms. Table 14.1 lists the specifications of the three-phase motor control timer function. Figure 14.1 shows the block diagram for three-phase motor control timer function. Also, the related registers are shown on Figures 14.2 to 14.8.

**Table 14.1 Three-Phase Motor Control Timer Function Specifications**

Item	Specification
Three-Phase Waveform Output Pin	Six pins (U, $\overline{U}$ , V, $\overline{V}$ , W, $\overline{W}$ )
Forced Cutoff Input <sup>(1)</sup>	Input “L” to NMI pin
Used Timers	Timer A4, A1, A2 (used in the one-shot timer mode) <ul style="list-style-type: none"> <li>• Timer A4: U- and <math>\overline{U}</math>-phase waveform control</li> <li>• Timer A1: V- and <math>\overline{V}</math>-phase waveform control</li> <li>• Timer A2: W- and <math>\overline{W}</math>-phase waveform control</li> </ul> Timer B2 (used in the timer mode) <ul style="list-style-type: none"> <li>• Carrier wave cycle control</li> </ul> Dead time timer (3 eight-bit timer and shared reload register) <ul style="list-style-type: none"> <li>• Dead time control</li> </ul>
Output Waveform	Triangular wave modulation, Sawtooth wave modification <ul style="list-style-type: none"> <li>• Enable to output “H” or “L” for one cycle</li> <li>• Enable to set positive-phase level and negative-phase level respectively</li> </ul>
Carrier Wave Cycle	Triangular wave modulation: count source $\times (m+1) \times 2$ Sawtooth wave modulation: count source $\times (m+1)$ m: Setting value of the TB2 register, 0000h to FFFFh Count source: f1, f2, f8, f32, fC32
Three-Phase PWM Output Width	Triangular wave modulation: count source $\times n \times 2$ Sawtooth wave modulation: count source $\times n$ n: Setting value of the TA4, TA1 and TA2 registers (of the TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11 bit to “1”), 0001h to FFFFh Count source: f1, f2, f8, f32, fC32
Dead Time	Count source $\times p$ , or no dead time p: Setting value of the DTT register, 01h to FFh Count source: f1, f2, f1 divided by 2, f2 divided by 2
Active Level	Enable to select “H” or “L”
Positive and Negative-Phase Concurrent Active Disable Function	Positive and negative-phases concurrent active disable function Positive and negative-phases concurrent active detect function
Interrupt Frequency	For Timer B2 interrupt, select a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis

**NOTE:**

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

Related pins: • P7\_2/CLK2/TA1OUT/V  
 • P7\_3/CTS2/RTS2/TA1IN/ $\overline{V}$   
 • P7\_4/TA2OUT/W  
 • P7\_5/TA2IN/ $\overline{W}$   
 • P8\_0/TA4OUT/U  
 • P8\_1/TA4IN/ $\overline{U}$

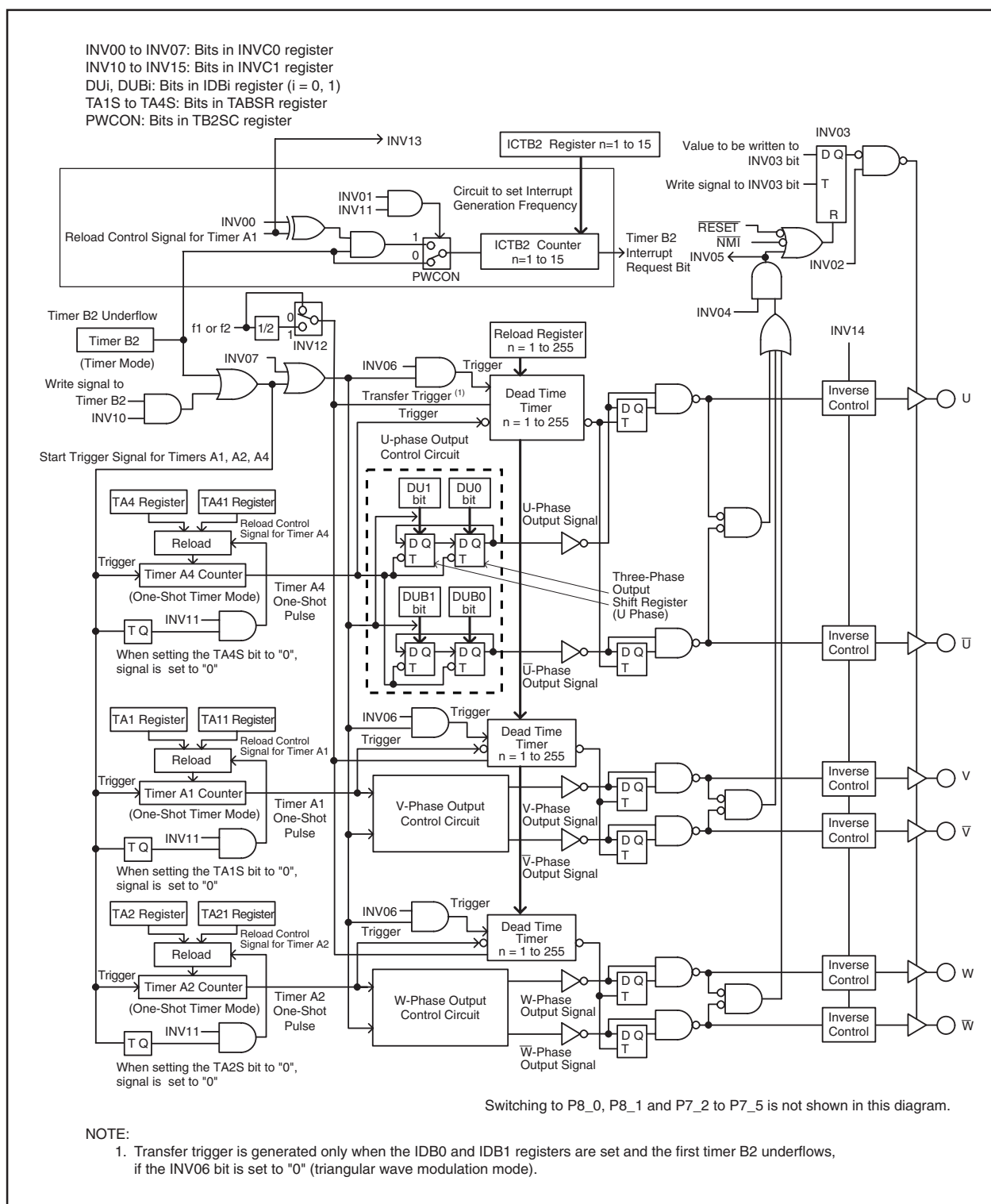


Figure 14.1 Three-Phase Motor Control Timer Function Block Diagram

Three-Phase PWM Control Register 0 <sup>(1)</sup>

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset	
								INVC0	01C8h	00h	
								Bit Symbol	Bit Name	Function	RW
								INV00	Interrupt Enable Output Polarity Select Bit	0: The ICTB2 counter is incremented by one on the rising edge of the timer A1 reload control signal 1: The ICTB2 counter is incremented by one on the falling edge of the timer A1 reload control signal <sup>(2)</sup>	RW
								INV01	Interrupt Enable Output Specification Bit <sup>(3)</sup>	0: ICTB2 counter is incremented by one when timer B2 underflows 1: Selected by the INV00 bit <sup>(2)</sup>	RW
								INV02	Mode Select Bit <sup>(4)</sup>	0: No three-phase control timer functions 1: Three-phase control timer function <sup>(5)</sup>	RW
								INV03	Output Control Bit	0: Disables three-phase control timer output <sup>(5)</sup> 1: Enables three-phase control timer output <sup>(6)</sup>	RW
								INV04	Positive and Negative-Phases Concurrent Active Disable Function Enable Bit	0: Enables concurrent active output 1: Disables concurrent active output	RW
								INV05	Positive and Negative-Phases Concurrent Active Output Detect Flag	0: Not detected 1: Detected <sup>(7)</sup>	RW
								INV06	Modulation Mode Select <sup>(8)</sup>	0: Triangular wave modulation mode 1: Sawtooth wave modulation mode <sup>(9)</sup>	RW
								INV07	Software Trigger Select Bit	Transfer trigger is generated when the INV07 bit is set to "1". Trigger to the dead time timer is also generated when setting the INV06 bit to "1". Its value is "0" when read.	RW

## NOTES:

- Set the INVC0 register after the PRC1 bit in the PRCR register is set to "1" (write enable). Rewrite the INV00 to INV02 and INV06 bits when the timers A1, A2, A4 and B2 stop.
- The INV00 and INV01 bits are enabled only when the INV11 bit is set to "1" (three-phase mode 1). The ICTB2 counter is incremented by one every time the timer B2 underflows, regardless of INV00 and INV01 bit settings, when the INV11 bit is set to "0" (three-phase mode 0).  
When setting the INV01 bit to "1", set the timer A1 count start flag before the first timer B2 underflow.  
When the INV00 bit is set to "1", the first interrupt is generated when the timer B2 underflows  $n-1$  times, if  $n$  is the value set in the ICTB2 counter. Subsequent interrupts are generated every  $n$  times the timer B2 underflows.
- Set the INV01 bit to "1" after setting the ICTB2 register.
- Set the INV02 bit to "1" to operate the dead time timer, U-, V- and W-phase output control circuits and ICTB2 counter.
- When the INV03 bit is set to "1", the pins applied to U/V/W output three-phase PWM.  
The U,  $\bar{U}$ , V,  $\bar{V}$ , W and  $\bar{W}$  pins, including pins shared with other output functions, are all placed in high-impedance states when the following conditions are all met.
  - The INV02 bit is set to "1" (three-phase control timer function)
  - The INV03 bit to "0" (three-phase control timer output disabled)
  - Direction registers of each port are set to "0" (input mode)
- The INV03 bit is set to "0" when the following conditions are all met.
  - Reset
  - A concurrent active state occurs while INV04 bit is set to "1"
  - The INV03 bit is set to "0" by program
  - A signal applied to the  $\overline{\text{NMI}}$  pin changes "H" to "L"
 When both the INV04 and INV05 bits are set to "1", the INV03 bit is set to "0".
- The INV05 bit cannot be set to "1" by program. Set the INV04 bit to "0", as well, when setting the INV05 bit to "0".
- The following table describes how the INV06 bit works.

Item	INV06 = 0	INV06 = 1
Mode	Triangular wave modulation mode	Sawtooth wave modulation mode
Timing to Transfer from the IDB0 and IDB1 Registers to Three-Phase Output Shift Register	Transferred once by generating a transfer trigger after setting the IDB0 and IDB1 registers	Transferred every time a transfer trigger is generated
Timing to Trigger the Dead Time Timer when the INV16 Bit=0	On the falling edge of a one-shot pulse of the timer A1, A2 or A4	By a transfer trigger, or the falling edge of a one-shot pulse of the timer A1, A2 or A4
INV13 Bit	Enabled when the INV11 bit=1 and the INV06 bit=0	Disabled

Transfer trigger : Timer B2 underflows and write to the INV07 bit, or write to the TB2 register when INV10 = 1

- When the INV06 bit is set to "1", set the INV11 bit to "0" (three-phase mode 0) and the PWCON bit in the TB2SC register to "0" (reload timer B2 with timer B2 underflow).

Figure 14.2 INVC0 Register

Three-Phase PWM Control Register 1<sup>(1)</sup>

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset	
0								INVC1	01C9h	00h	
								Bit Symbol	Bit Name	Function	RW
								INV10	Timer A1, A2 and A4 Start Trigger Select Bit	0: Timer B2 underflow 1: Timer B2 underflow and write to the timer B2	RW
								INV11	Timer A1-1, A2-1, A4-1 Control Bit <sup>(2)</sup>	0: Three-phase mode 0 <sup>(3)</sup> 1: Three-phase mode 1	RW
								INV12	Dead Time Timer Count Source Select Bit	0 : f1 or f2 1 : f1 divided-by-2 or f2 divided-by-2	RW
								INV13	Carrier Wave Detect Flag <sup>(4)</sup>	0: Timer A1 reload control signal is "0" 1: Timer A1 reload control signal is "1"	RO
								INV14	Output Polarity Control Bit	0 : Active "L" of an output waveform 1 : Active "H" of an output waveform	RW
								INV15	Dead Time Disable Bit	0: Enables dead time 1: Disables dead time	RW
								INV16	Dead Time Timer Trigger Select Bit	0: Falling edge of a one-shot pulse of the timer A1, A2, A4 <sup>(5)</sup> 1: Rising edge of the three-phase output shift register (U-, V-, W-phase)	RW
								- (b7)	Reserved Bit	Set to "0"	RW

## NOTES:

1. Rewrite the INVC1 register after the PRC1 bit in the PRCR register is set to "1" (write enable). The timers A1, A2, A4, and B2 must be stopped during rewrite.
2. The following table lists how the INV11 bit works.

Item	INV11 = 0	INV11 = 1
Mode	Three-phase mode 0	Three-phase mode 1
TA11, TA21 and TA41 Registers	Not used	Used
INV00 and INV01 Bit	Disabled. The ICTB2 counter is incremented whenever the timer B2 underflows	Enabled
INV13 Bit	Disabled	Enabled when INV11=1 and INV06=0

3. When the INV06 bit is set to "1" (sawtooth wave modulation mode), set the INV11 bit to "0" (three-phase mode 0). Also, when the INV11 bit is set to "0", set the PWCON bit in the TB2SC register to "0" (timer B2 is reloaded when the timer B2 underflows).
4. The INV13 bit is enabled only when the INV06 bit is set to "0" (Triangular wave modulation mode) and the INV11 bit to "1" (three-phase mode 1).
5. If the following conditions are all met, set the INV16 bit to "1" (rising edge of the three-phase output shift register).
  - The INV15 bit is set to "0" (dead time timer enabled)
  - The Dij bit (i=U, V or W, j=0, 1) and DiBj bit always have different values when the INV03 bit is set to "1". (The positive-phase and negative-phase always output opposite level signals.)
 If above conditions are not met, set the INV16 bit to "0" (falling edge of a one-shot pulse of the timer A1, A2, A4).

Figure 14.3 INVC1 Register

Three-Phase Output Buffer Register i (i = 0, 1) <sup>(1)</sup>

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset	
0	0							IDB0, IDB1	01CAh, 01CBh	00h	
								Bit Symbol	Bit Name	Function	RW
								DUi	U-Phase Output Buffer i	Write output level 0: Active level 1: Inactive level	RW
								DUBi	$\bar{U}$ -Phase Output Buffer i		RW
								DVi	V-Phase Output Buffer i	When read, the value of the three-phase shift register is read.	RW
								DVBi	$\bar{V}$ -Phase Output Buffer i		RW
								DWi	W-Phase Output Buffer i		RW
								DWBi	$\bar{W}$ -Phase Output Buffer i		RW
								- (b7-b6)	Reserved Bit	Set to "0"	RO

## NOTE:

1. Values of the IDB0 and IDB1 registers are transferred to the three-phase output shift register by a transfer trigger.

After the transfer trigger occurs, the values written in the IDB0 register determine each phase output signal first. Then the value written in the IDB1 register on the falling edge of timers A1, A2 and A4 one-shot pulse determines each phase output signal.

Dead Time Timer <sup>(1) (2)</sup>

b7	b0	Symbol	Address	After Reset
		DTT	01CCh	Indeterminate
		Function	Setting Range	RW
		If setting value is $n$ , the timer stops when counting $n$ times a count source selected by the INV12 bit in the INVC1 register after start trigger occurs. Positive or negative phase, which changes from inactive level to active level, shifts when the dead time timer stops.	1 to 255	WO

## NOTES:

1. Use the MOV instruction to set the DTT register.
2. The DTT register is enabled when the INV15 bit in the INVC1 register is set to "0" (dead time enabled). No dead time can be set when the INV15 bit is set to "1" (dead time disabled). The INV06 bit in the INVC0 register determines start trigger of the DTT register.

Figure 14.4 IDB0 and IDB1 Registers and DTT Register



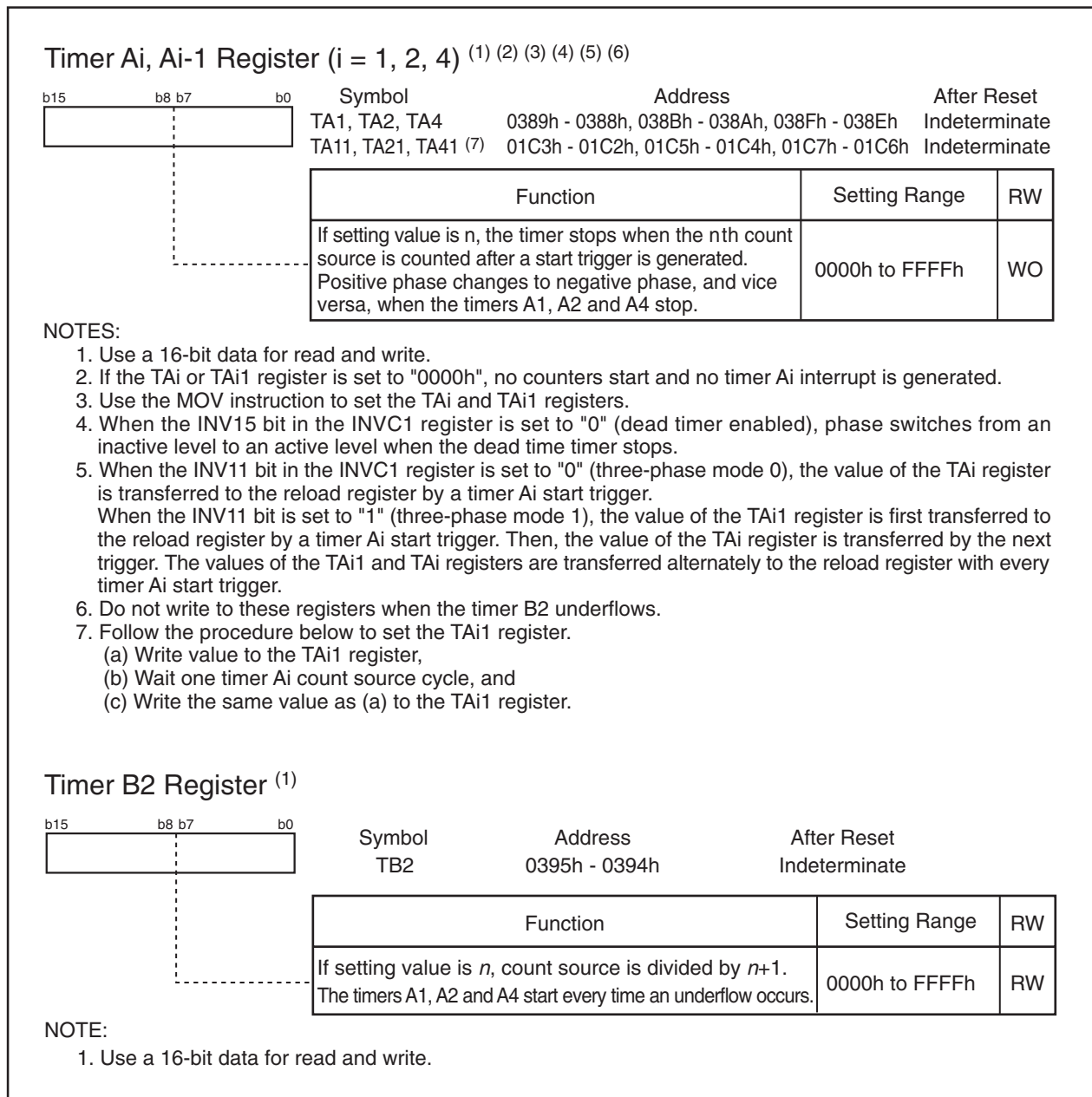


Figure 14.5 TA1, TA2, TA4, TA11, TA21 and TA41 Registers, and TB2 Register


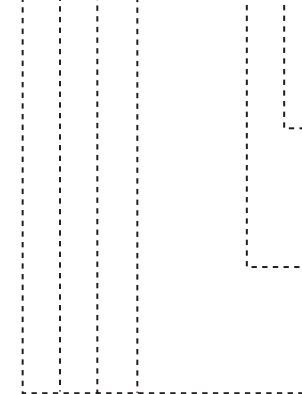
Timer B2 Interrupt Occurrence Frequency Set Counter <sup>(1)</sup> <sup>(2)</sup> <sup>(3)</sup>

<div><div><div><div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div></div></div><div></div></div><div>b7</div></div>	Symbol ICTB2	Address 01CDh	After Reset Indeterminate									
<div><div><div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div></div></div><div></div></div> <div>b0</div>	<table><thead><tr><th>Function</th><th>Setting Range</th><th>RW</th></tr></thead><tbody><tr><td>When the INV01 bit in the INVC0 register is set to "0" (the ICTB2 counter increments whenever the timer B2 underflows) and the setting value is <math>n</math>, the timer B2 interrupt is generated every <math>n</math>th time timer B2 underflow occurs. When the INV01 bit is set to "1" (the INV00 bit selects count timing of the ICTB2 counter) and setting value is <math>n</math>, the timer B2 interrupt is generated every <math>n</math>th time timer B2 underflow meeting the condition selected in the INV00 bit occurs.</td><td>1 to 15</td><td>WO</td></tr><tr><td colspan="2">Nothing is assigned. When write, set to "0".</td><td>—</td></tr></tbody></table>			Function	Setting Range	RW	When the INV01 bit in the INVC0 register is set to "0" (the ICTB2 counter increments whenever the timer B2 underflows) and the setting value is $n$ , the timer B2 interrupt is generated every $n$ th time timer B2 underflow occurs. When the INV01 bit is set to "1" (the INV00 bit selects count timing of the ICTB2 counter) and setting value is $n$ , the timer B2 interrupt is generated every $n$ th time timer B2 underflow meeting the condition selected in the INV00 bit occurs.	1 to 15	WO	Nothing is assigned. When write, set to "0".		—
Function	Setting Range	RW										
When the INV01 bit in the INVC0 register is set to "0" (the ICTB2 counter increments whenever the timer B2 underflows) and the setting value is $n$ , the timer B2 interrupt is generated every $n$ th time timer B2 underflow occurs. When the INV01 bit is set to "1" (the INV00 bit selects count timing of the ICTB2 counter) and setting value is $n$ , the timer B2 interrupt is generated every $n$ th time timer B2 underflow meeting the condition selected in the INV00 bit occurs.	1 to 15	WO										
Nothing is assigned. When write, set to "0".		—										

## NOTES:

1. Use the MOV instruction to set the ICTB2 register.
2. If the INV01 bit is set to "1", set the ICTB2 register when the TB2S bit is set to "0" (timer B2 counter stopped), If the INV01 bit is set to "0" and the TB2S bit to "1" (timer B2 counter start), do not set the ICTB2 register when the timer B2 underflows.
3. If the INV00 bit is set to "1", the first interrupt is generated when the timer B2 underflows  $n-1$  times,  $n$  being the value set in the ICTB2 counter. Subsequent interrupts are generated every  $n$  times the timer B2 underflows.

Timer B2 Special Mode Register <sup>(1)</sup>

								Symbol TB2SC	Address 039Eh	After Reset XXXXXX00b	
								Bit Symbol	Bit Name	Function	RW
								PWCON	Timer B2 Reload Timing Switching Bit	0 : Timer B2 underflow 1 : Timer A output at odd-numbered occurrences (2)	RW
								IVPCR1	Three-Phase Output Port NMI Control Bit 1 (3)	0 : Three-phase output forcible cutoff by NMI input (high-impedance) disabled 1 : Three-phase output forcible cutoff by NMI input (high-impedance) enabled	RW
								- (b7-b2)	Nothing is assigned. When write, set to "0". When read, their contents are "0".	-	

## NOTES:

1. Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enabled).
2. If the INV11 bit in the INVC1 register is "0" (three-phase mode 0) or the INV06 bit in the INVC0 register is "1" (sawtooth wave modulation mode), set this bit to "0" (timer B2 underflow).
3. Related pins are U(P8\_0/TA4OUT),  $\bar{U}$ (P8\_1/TA4IN), V(P7\_2/CLK2/TA1OUT),  $\bar{V}$ (P7\_3/CTS2/RTS2/TA1IN), W(P7\_4/TA2OUT),  $\bar{W}$ (P7\_5/TA2IN). If a low-level signal is applied to the NMI pin when the IVPCR1 bit = 1, the target pins go to a high-impedance state regardless of which functions of those pins are being used.  
After forced interrupt (cutoff), input "H" to the  $\bar{NMI}$  pin and set the IVPCR1 bit to "0": this forced cutoff will be reset.

Figure 14.6 ICTB2 Register and TB2SC Register

## Trigger Select Register

<div><div>b7b6b5b4b3b2b1b0</div><div></div></div>								Symbol TRGSR	Address 0383h	After Reset 00h	
								Bit Symbol	Bit Name	Function	RW
								TA1TGL	Timer A1 Event/Trigger Select Bit	Set to "01b" (TB2 underflow) before using a V-phase output control circuit	RW
								TA1TGH			RW
								TA2TGL	Timer A2 Event/Trigger Select Bit	Set to "01b" (TB2 underflow) before using a W-phase output control circuit	RW
								TA2TGH			RW
								TA3TGL	Timer A3 Event/Trigger Select Bit	b5 b4 0 0: Selects an input to the TA3IN pin <sup>(1)</sup> 0 1: Selects TB2 <sup>(2)</sup> 1 0: Selects TA2 <sup>(2)</sup> 1 1: Selects TA4 <sup>(2)</sup>	RW
								TA3TGH			RW
								TA4TGL	Timer A4 Event/Trigger Select Bit	Set to "01b" (TB2 underflow) before using a U-phase output control circuit	RW
								TA4TGH			RW

## NOTES:

1. Set the corresponding port direction bit to "0" (input mode).
2. Overflow or underflow.

## Count Start Flag

<div><div>b7b6b5b4b3b2b1b0</div><div></div></div>								Symbol TABSR	Address 0380h	After Reset 00h	
								Bit Symbol	Bit Name	Function	RW
								TA0S	Timer A0 Count Start Flag	0 : Stops counting 1 : Starts counting	RW
								TA1S	Timer A1 Count Start Flag		RW
								TA2S	Timer A2 Count Start Flag		RW
								TA3S	Timer A3 Count Start Flag		RW
								TA4S	Timer A4 Count Start Flag		RW
								TB0S	Timer B0 Count Start Flag		RW
								TB1S	Timer B1 Count Start Flag		RW
								TB2S	Timer B2 Count Start Flag		RW

Figure 14.7 TRGSR Register and TRBSR Register

## Timer Ai Mode Register (i = 1, 2, 4)

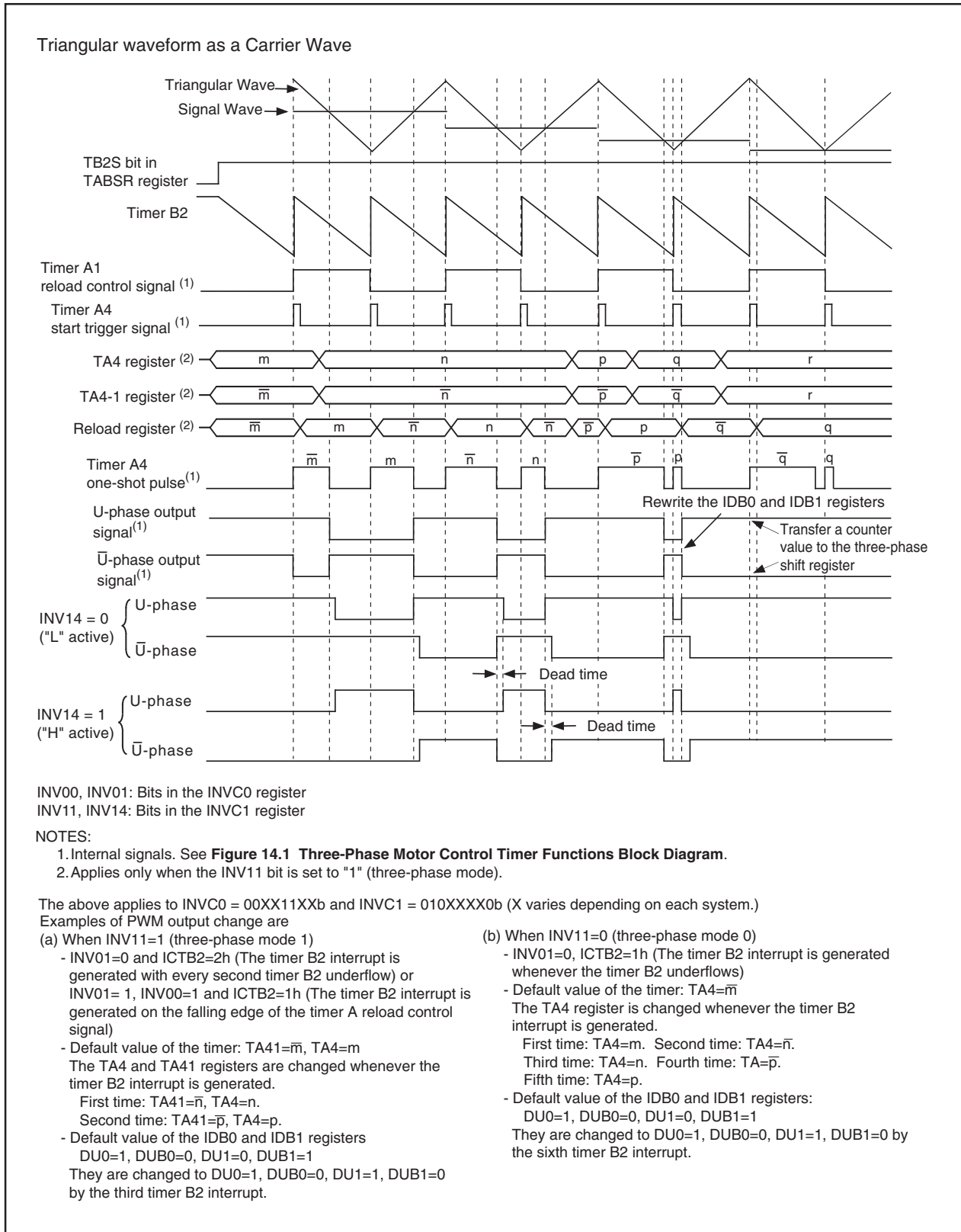
<div><div>b7b6b5b4b3b2b1b0</div><div><div></div><div></div><div>0</div><div>1</div><div>0</div><div>0</div><div>1</div><div>0</div></div></div>								Symbol	Address	After Reset	
								TA1MR, TA2MR, TA4MR	0397h, 0398h, 039Ah	00h	
								Bit Symbol	Bit Name	Function	RW
								TMOD0	Operation Mode Select Bit	Set to "10b" (one-shot timer mode) with the three-phase motor control timer function	RW
								TMOD1			RW
								MR0	Pulse Output Function Select Bit	Set to "0" with the three-phase motor control timer function	RW
								MR1	External Trigger Select Bit	Set to "0" with the three-phase motor control timer function	RW
								MR2	Trigger Select Bit	Set to "1" (selected by the TRGSR register) with the three-phase motor control timer function	RW
								MR3	Set to "0" with the three-phase motor control timer function		RW
								TCK0	Count Source Select Bit	b7 b6 0 0 : f1 or f2 0 1 : f8 1 0 : f32 1 1 : fC32	RW
								TCK1			RW

## Timer B2 Mode Register

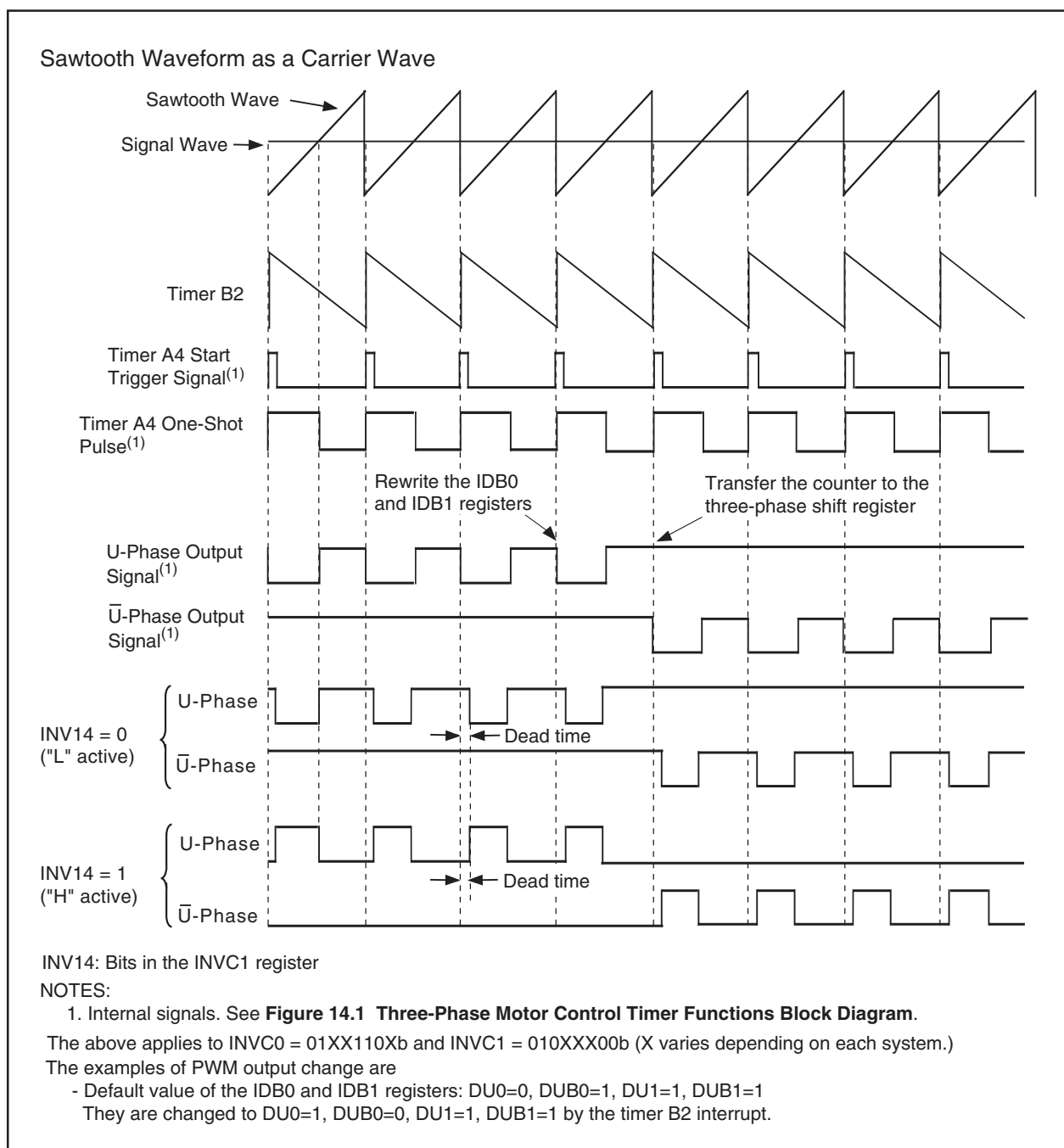
<div><div>b7b6b5b4b3b2b1b0</div><div><div></div><div></div><div></div><div>0</div><div></div><div></div><div>0</div><div>0</div></div></div>								Symbol TB2MR	Address 039Dh	After Reset 00XX0000b		
								Bit Symbol	Bit Name	Function	RW	
								TMOD0	Operation Mode Select Bit	Set to "00b" (timer mode) when using the three-phase motor control timer function	RW	
								TMOD1			RW	
								MR0	Disabled when using the three-phase motor control timer function. When write, set to "0". When read, its content is indeterminate.		RW	
								MR1			RW	
								MR2	Set to "0" when using three-phase motor control timer function		RW	
								MR3	When write in three-phase motor control timer function, set to "0". When read in three-phase motor control timer function, its content is indeterminate.		RO	
								TCK0	Count Source Select Bit	b7 b6 0 0 : f1 or f2 0 1 : f8 1 0 : f32 1 1 : fC32		RW
								TCK1				RW

Figure 14.8 TA1MR, TA2MR and TA4MR Registers, and TB2MR Register

The three-phase motor control timer function is enabled by setting the INV02 bit in the INVC0 register to "1". When this function is selected, timer B2 is used to control the carrier wave, and timers A4, A1 and A2 are used to control three-phase PWM outputs (U,  $\bar{U}$ , V,  $\bar{V}$ , W and  $\bar{W}$ ). The dead time is controlled by a dedicated dead-time timer. Figure 14.9 shows the example of triangular modulation waveform and Figure 14.10 shows the example of sawtooth modulation waveform.



**Figure 14.9 Triangular Wave Modulation Operation**

**Figure 14.10 Sawtooth Wave Modulation Operation**

## 15. Serial Interface

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

### 15.1 UARTi (i = 0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other. Figures 15.1 to 15.3 show the block diagram of UARTi. Figure 15.4 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 mode)
- Special mode 2
- Special mode 3 (Bus collision detection function, IE mode)
- Special mode 4 (SIM mode) : UART2

Figures 15.5 to 15.10 show the UARTi-related registers.

Refer to tables listing each mode for register setting.





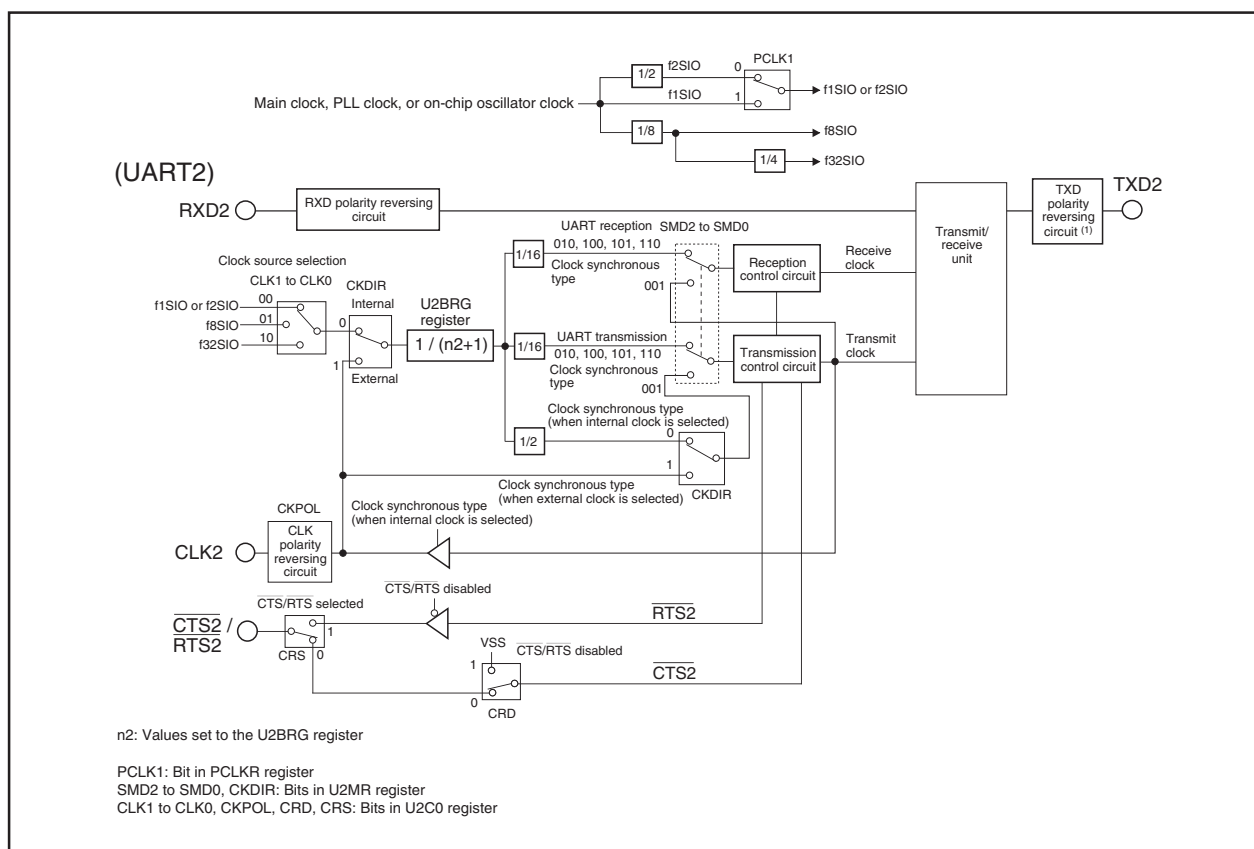


Figure 15.3 UART2 Block Diagram

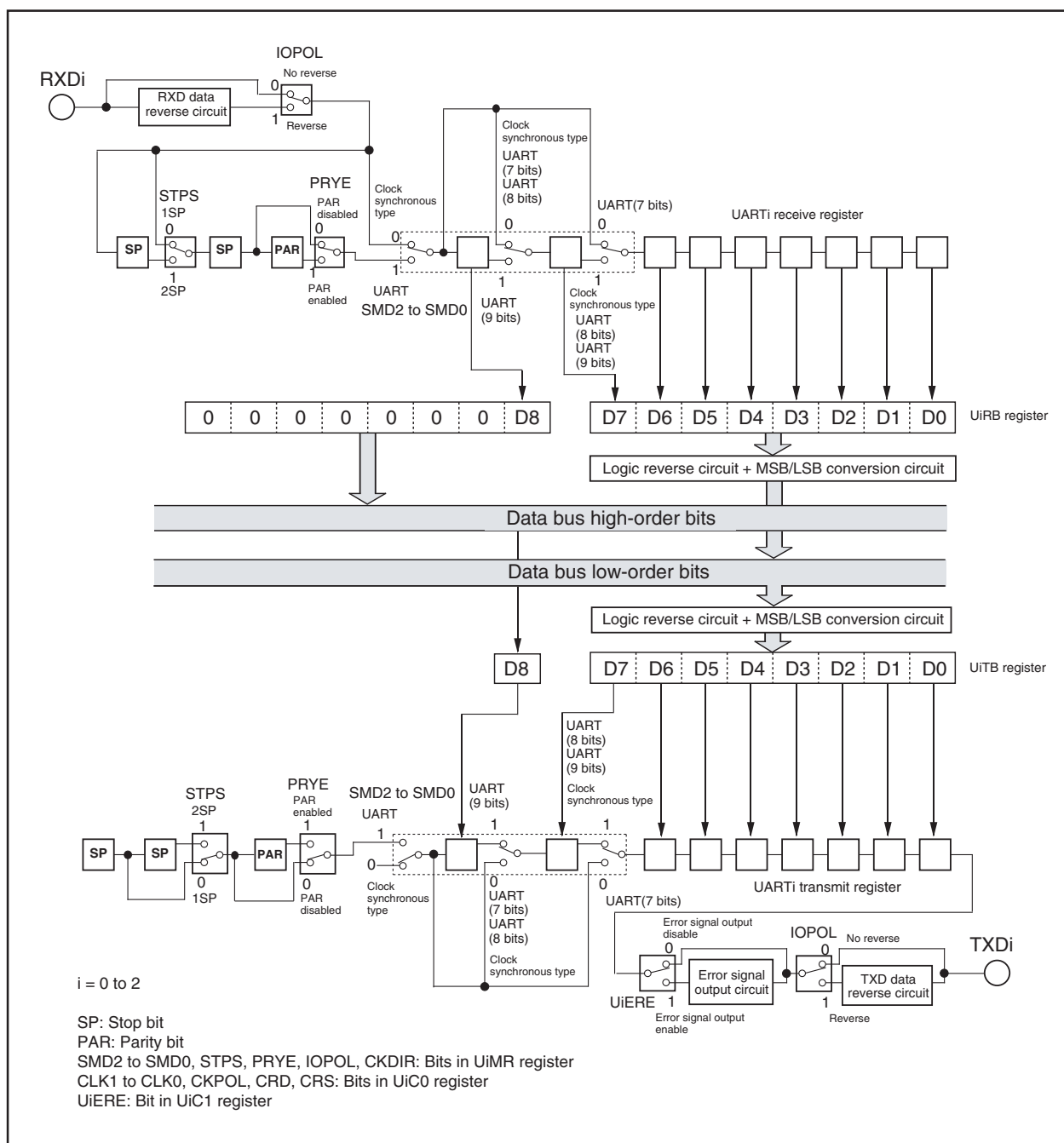


Figure 15.4 UARTi Transmit/Receive Unit

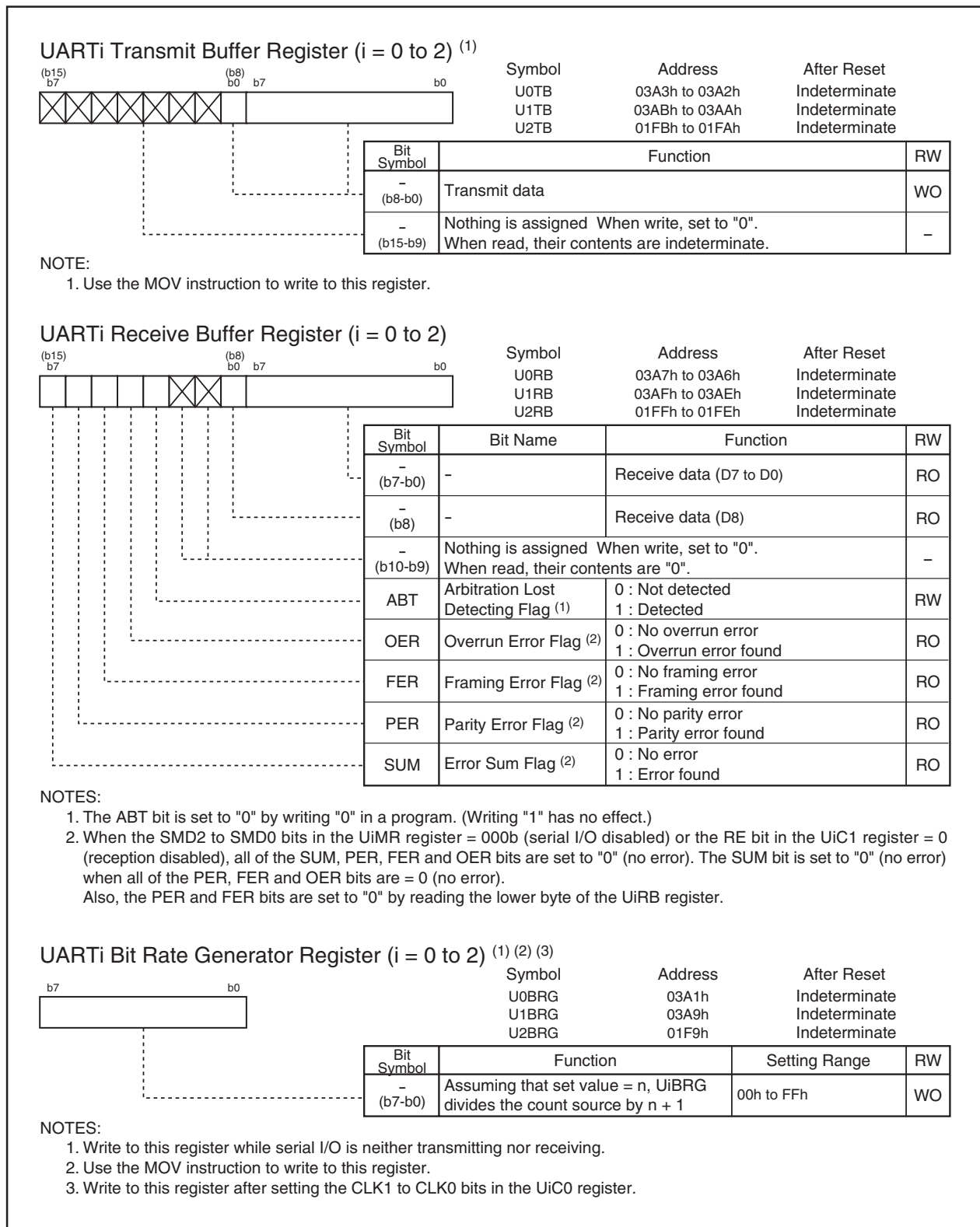


Figure 15.5 U0TB to U2TB Registers, U0RB to U2RB Registers, and U0BRG to U2BRG Registers

## UARTi Transmit/Receive Mode Register (i = 0 to 2)

Symbol								Address		After Reset	
U0MR to U2MR								03A0h, 03A8h, 01F8h		00h	
b7	b6	b5	b4	b3	b2	b1	b0				

## NOTES:

1. To receive data, set the corresponding port direction bit for each RXDi pin to "0" (input mode).
2. Set the corresponding port direction bit for SCL and SDA pins to "0" (input mode).
3. Set the corresponding port direction bit for each CLKi pin to "0" (input mode).

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

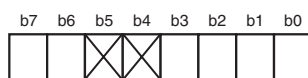
b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
								U0C0 to U2C0	03A4h, 03ACh, 01FCh	00001000b

## NOTES:

1. CTS1/RTS1 can be used when the CLKMD1 bit in the UCON register = 0 (only CLK1 output) and the RCSP bit in the UCON register = 0 (CTS0/RTS0 not separated).
2. Set the corresponding port direction bit for each CTSi pin to "0" (input mode).
3. SCL2/P7\_1 is N channel open-drain output. The NCH bit in the U2C0 register is N channel open-drain output regardless of the NCH bit.
4. The UFORM bit is enabled when the SMD2 to SMD0 bits in the UiMR register are set to "001b" (clock synchronous serial I/O mode), or "101b" (UART mode, 8-bit transfer data). Set this bit to "1" when the SMD2 to SMD0 bits are set to "010b" (I<sup>2</sup>C mode), and to "0" when the SMD2 to SMD0 bits are set to "100b" (UART mode, 7-bit transfer data) or "110b" (UART mode, 9-bit transfer data).
5. When changing the CLK1 to CLK0 bits, set the UIBRG register.

Figure 15.6 U0MR to U2MR Registers and U0C0 to U2C0 Registers

## UARTj Transmit/Receive Control Register 1 (j = 0, 1)



Symbol  
U0C1, U1C1

Address  
03A5h, 03ADh

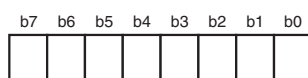
After Reset  
00XX0010b

Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit Buffer Empty Flag	0 : Data present in the UjTB register 1 : No data present in the UjTB register	RO
RE	Receive Enable Bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive Complete Flag	0 : No data present in the UjRB register 1 : Data present in the UjRB register	RO
– (b5-b4)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		–
UjLCH	Data Logic Select Bit (1)	0 : No reverse 1 : Reverse	RW
UjERE	Error Signal Output Enable Bit	0 : Output disabled 1 : Output enabled	RW

## NOTE:

- The UjLCH bit is enabled when the SMD2 to SMD0 bits in the UjMR register are set to "001b" (clock synchronous serial I/O mode), "100b" (UART mode, 7-bit transfer data) or "101b" (UART mode, 8-bit transfer data).  
Set this bit to "0" when the SMD2 to SMD0 bits are set to "010b" (I<sup>2</sup>C mode) or "110b" (UART mode, 9-bit transfer data).

## UART2 Transmit/Receive Control Register 1



Symbol  
U2C1

Address  
01FDh

After Reset  
00000010b

Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit Buffer Empty Flag	0 : Data present in U2TB register 1 : No data present in U2TB register	RO
RE	Receive Enable Bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive Complete Flag	0 : No data present in U2RB register 1 : Data present in U2RB register	RO
U2IRS	UART2 Transmit Interrupt Cause Select Bit	0 : Transmit buffer empty (TI bit = 1) 1 : Transmit is completed (TXEPT bit = 1)	RW
U2RRM	UART2 Continuous Receive Mode Enable Bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
U2LCH	Data Logic Select Bit (1)	0 : No reverse 1 : Reverse	RW
U2ERE	Error Signal Output Enable Bit	0 : Output disabled 1 : Output enabled	RW

## NOTE:

- The U2LCH bit is enabled when the SMD2 to SMD0 bits in the U2MR register are set to "001b" (clock synchronous serial I/O mode), "100b" (UART mode, 7-bit transfer data) or "101b" (UART mode, 8-bit transfer data).  
Set this bit to "0" when the SMD2 to SMD0 bits are set to "010b" (I<sup>2</sup>C mode) or "110b" (UART mode, 9-bit transfer data).

Figure 15.7 U0C1, U1C1 Registers and U2C1 Register

## UART Transmit/Receive Control Register 2

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
<div><div></div><div></div></div>								UCON	03B0h	X0000000b
		</								

## NOTE:

- When using multiple transfer clock output pins, make sure the following conditions are met:
  - The CKDIR bit in the U1MR register = 0 (internal clock)

## UARTi Special Mode Register (i = 0 to 2)

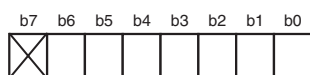
b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
<div><div></div><div></div><div></div><div>0</div><div></div><div></div><div></div><div></div></div>								U0SMR to U2SMR	01EFh, 01F3h, 01F7h	X0000000b

## NOTES:

- The BBS bit is set to "0" by writing "0" in a program. (Writing "1" has no effect.).
- Underflow signal of timer A3 in UART0, underflow signal of timer A4 in UART1, underflow signal of timer A0 in UART2.
- When a transfer begins, the SSS bit is set to "0" (not synchronized to RXDi).

Figure 15.8 UCON Register and U0SMR to U2SMR Registers

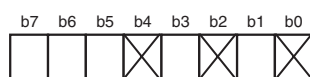
## UARTi Special Mode Register 2 (i = 0 to 2)



Symbol: U0SMR2 to U2SMR2      Address: 01EEh, 01F2h, 01F6h      After Reset: X0000000b

Bit Symbol	Bit Name	Function	RW
IICM2	I <sup>2</sup> C Mode Select Bit 2	See <b>Table 15.12 I<sup>2</sup>C Mode Functions</b>	RW
CSC	Clock-Synchronous Bit	0 : Disabled 1 : Enabled	RW
SWC	SCL Wait Output Bit	0 : Disabled 1 : Enabled	RW
ALS	SDA Output Stop Bit	0 : Disabled 1 : Enabled	RW
STAC	UARTi Initialization Bit	0 : Disabled 1 : Enabled	RW
SWC2	SCL Wait Output Bit 2	0: Transfer clock 1: "L" output	RW
SDHI	SDA Output Disable Bit	0: Enabled 1: Disabled (high-impedance)	RW
– (b7)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–

## UARTi Special Mode Register 3 (i = 0 to 2)



Symbol: U0SMR3 to U2SMR3      Address: 01EDh, 01F1h, 01F5h      After Reset: 000X0X0Xb

Bit Symbol	Bit Name	Function	RW
– (b0)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–
CKPH	Clock Phase Set Bit	0 : Without clock delay 1 : With clock delay	RW
– (b2)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–
NODC	Clock Output Select Bit	0 : CLKi is CMOS output 1 : CLKi is N channel open-drain output	RW
– (b4)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–
DL0	SDAi Digital Delay Setup Bit (1) (2)	b7 b6 b5 0 0 0 : Without delay	RW
DL1		0 0 1 : 1 to 2 cycle(s) of UiBRG count source	RW
DL2		0 1 0 : 2 to 3 cycles of UiBRG count source	RW
		0 1 1 : 3 to 4 cycles of UiBRG count source	
		1 0 0 : 4 to 5 cycles of UiBRG count source	
		1 0 1 : 5 to 6 cycles of UiBRG count source	
		1 1 0 : 6 to 7 cycles of UiBRG count source	
		1 1 1 : 7 to 8 cycles of UiBRG count source	

## NOTES:

- The DL2 to DL0 bits are used to generate a delay in SDAi output by digital means during I<sup>2</sup>C mode. In other than I<sup>2</sup>C mode, set these bits to "000b" (no delay).
- The amount of delay varies with the load on SCLi and SDAi pins. Also, when using an external clock, the amount of delay increases by about 100 ns.

Figure 15.9 U0SMR2 to U2SMR2 Registers and U0SMR3 to U2SMR3 Registers

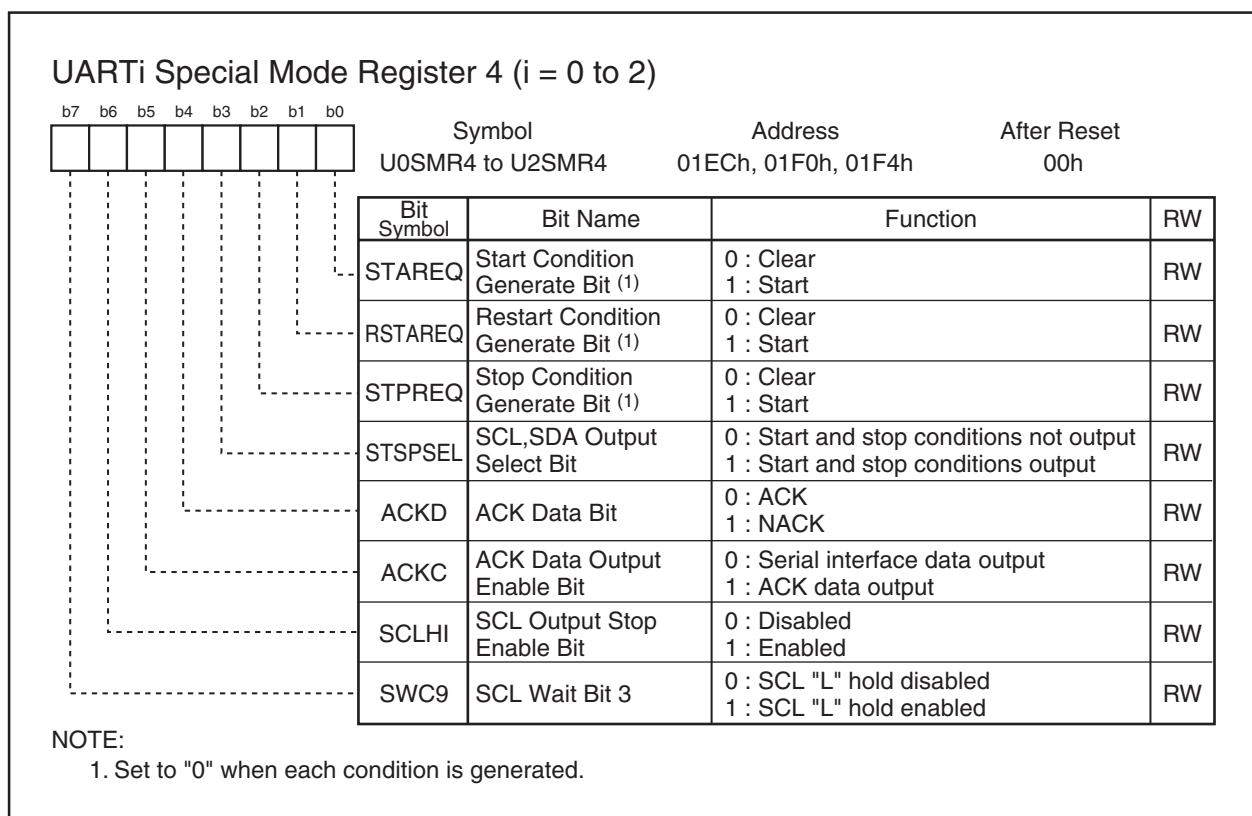


Figure 15.10 U0SMR4 to U2SMR4 Registers



### 15.1.1 Clock Synchronous Serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 15.1 lists the specifications of the clock synchronous serial I/O mode. Table 15.2 lists the registers used in clock synchronous serial I/O mode and the register values set.

**Table 15.1 Clock Synchronous Serial I/O Mode Specifications**

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	The CKDIR bit in the UiMR register = 0 (internal clock) : $f_j / 2(n+1)$ • $f_j = f1SIO, f2SIO, f8SIO, f32SIO$ . n: Setting value of the UiBRG register 00h to FFh The CKDIR bit = 1 (external clock) : Input from CLKi pin
Transmission, Reception Control	Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disabled
Transmission Start Condition	Before transmission can start, the following requirements must be met <sup>(1)</sup> • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register) • If CTS function is selected, input on the $\overline{CTS}_i$ pin = L
Reception Start Condition	Before reception can start, the following requirements must be met <sup>(1)</sup> • The RE bit in the UiC1 register = 1 (reception enabled) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected • The UiIRS bit <sup>(2)</sup> = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit = 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register For reception • When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error Detection	Overrun error <sup>(3)</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
Select Function	• CLK polarity selection Transfer data input/output can be selected to occur synchronously with the rising or the falling edge of the transfer clock • LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Continuous receive mode selection Reception is enabled immediately by reading the UiRB register • Switching serial data logic This function reverses the logic value of the transmit/receive data • Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set • Separate $\overline{CTS}/\overline{RTS}$ pins (UART0) $\overline{CTS}_0$ and $\overline{RTS}_0$ are input/output from separate pins

i = 0 to 2

**NOTES:**

1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit 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.
2. The U0IRS and U1IRS bits respectively are bits 0 and 1 in the UCON register; the U2IRS bit is bit 4 in the U2C1 register.
3. If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit in the SiRIC register does not change.

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

Register	Bit	Function
UiTB <sup>(1)</sup>	0 to 7	Set transmission data
UiRB <sup>(1)</sup>	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a transfer rate
UiMR <sup>(1)</sup>	SMD2 to SMD0	Set to "001b"
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to "0"
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select CTS or RTS to use
	TXEPT	Transmit register empty flag
	CRD	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>(2)</sup>	Select the source of UART2 transmit interrupt
	U2RRM <sup>(2)</sup>	Set this bit to "1" to use continuous receive mode
	UiLCH	Set this bit to "1" to use inverted data logic
	UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 2	Set to "0"
	NODC	Select clock output mode
	4 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the 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 the CLKMD1 bit = 1
	CLKMD1	Set this bit to "1" to output UART1 transfer clock from two pins
	RCSP	Set this bit to "1" to accept as input the UART0 CTS0 signal from the P6_4 pin
	7	Set to "0"

i = 0 to 2

**NOTES:**

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

Table 15.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Table 15.3 shows pin functions for the case where the multiple transfer clock output pin select function is deselected. Table 15.4 lists the P6\_4 pin functions during clock synchronous serial I/O mode.

Note that for a period from when the UART<sub>i</sub> operation mode is selected to when transfer starts, the TXD<sub>i</sub> pin outputs an “H”.

Figure 15.11 shows the transmit/receive timings during clock synchronous serial I/O mode.

**Table 15.3 Pin Functions (When Not Select Multiple Transfer Clock Output Pin Function)**

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

i = 0 to 2

**Table 15.4 P6\_4 Pin Functions**

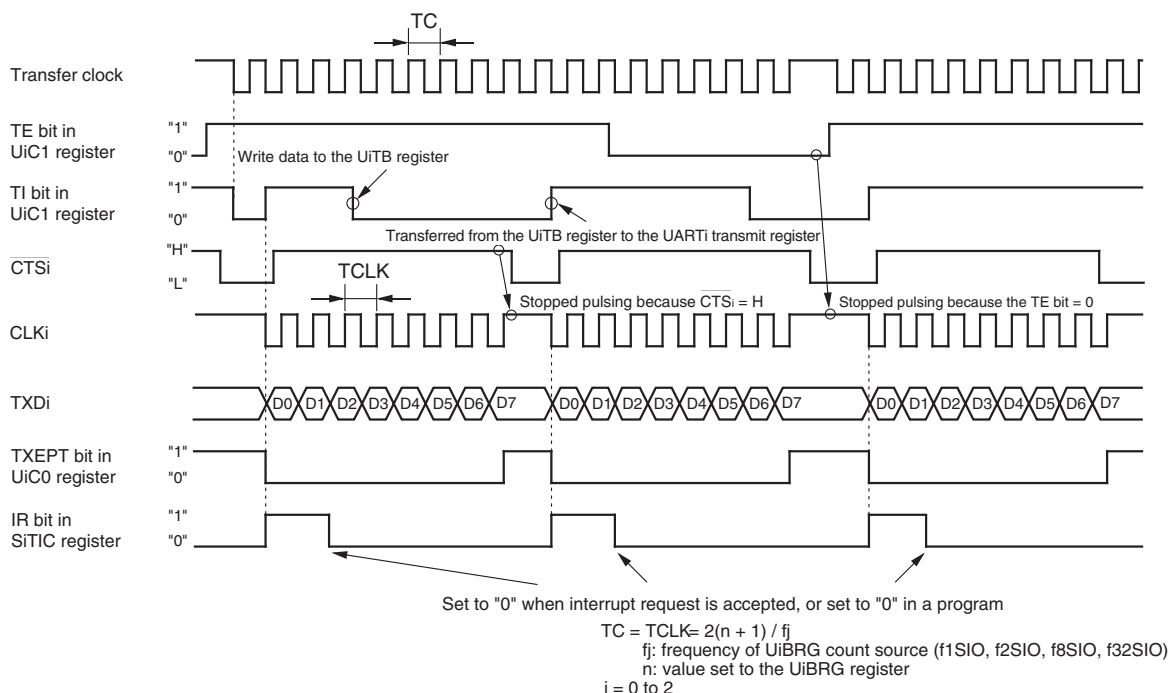
Pin Function	Bit set Value					
	U1C0 Register		UCON Register			PD6 Register
	CRD bit	CRS bit	RCSP bit	CLKMD1 bit	CLKMD0 bit	PD6_4 bit
P6_4	1	-	0	0	-	Input: 0, Output: 1
CTS1	0	0	0	0	-	0
RTS1	0	1	0	0	-	-
CTS0 <sup>(1)</sup>	0	0	1	0	-	0
CLKS1	-	-	-	1 <sup>(2)</sup>	1	-

∴ “0” or “1”

**NOTES:**

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

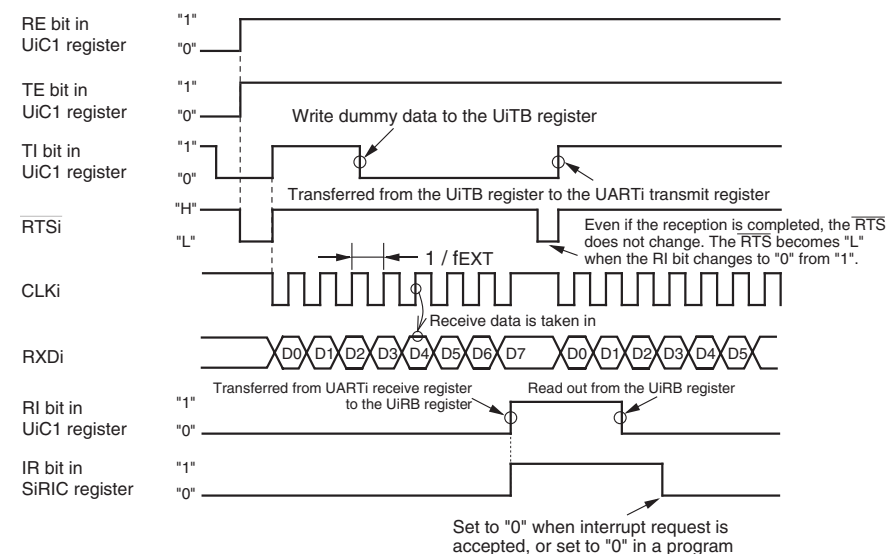
## (1) Example of Transmit Timing (when internal clock is selected)



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

- CKDIR bit in UIMR register = 0 (internal clock)
- CRD bit in UiC0 register = 0 (CTS/RTS enabled), CRS bit in UiC0 register = 0 (CT $\bar{S}$  selected)
- CKPOL bit in UiC0 register = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- UiIRS bit = 0 (an interrupt request occurs when the transmit buffer becomes empty):  
 U0IRS bit is bit 0 in UCON register  
 U1IRS bit is bit 1 in UCON register  
 U2IRS bit is bit 4 in U2C1 register

## (2) Example of Receive Timing (when external clock is selected)



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

- CKDIR bit in UIMR register = 1 (external clock)
- CRD bit in UiC0 register = 0 (CTS/RTS enabled), CRS bit = 1 (RT $\bar{S}$  selected)
- CKPOL bit in UiC0 register = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

fEXT: frequency of external clock

Make sure the following conditions are met when input to the CLKi pin before receiving data is high:

- TE bit in UiC1 register = 1 (transmission enabled)
- RE bit in UiC1 register = 1 (reception enabled)
- Write dummy data to the UiTB register

Figure 15.11 Transmit and Receive Operation

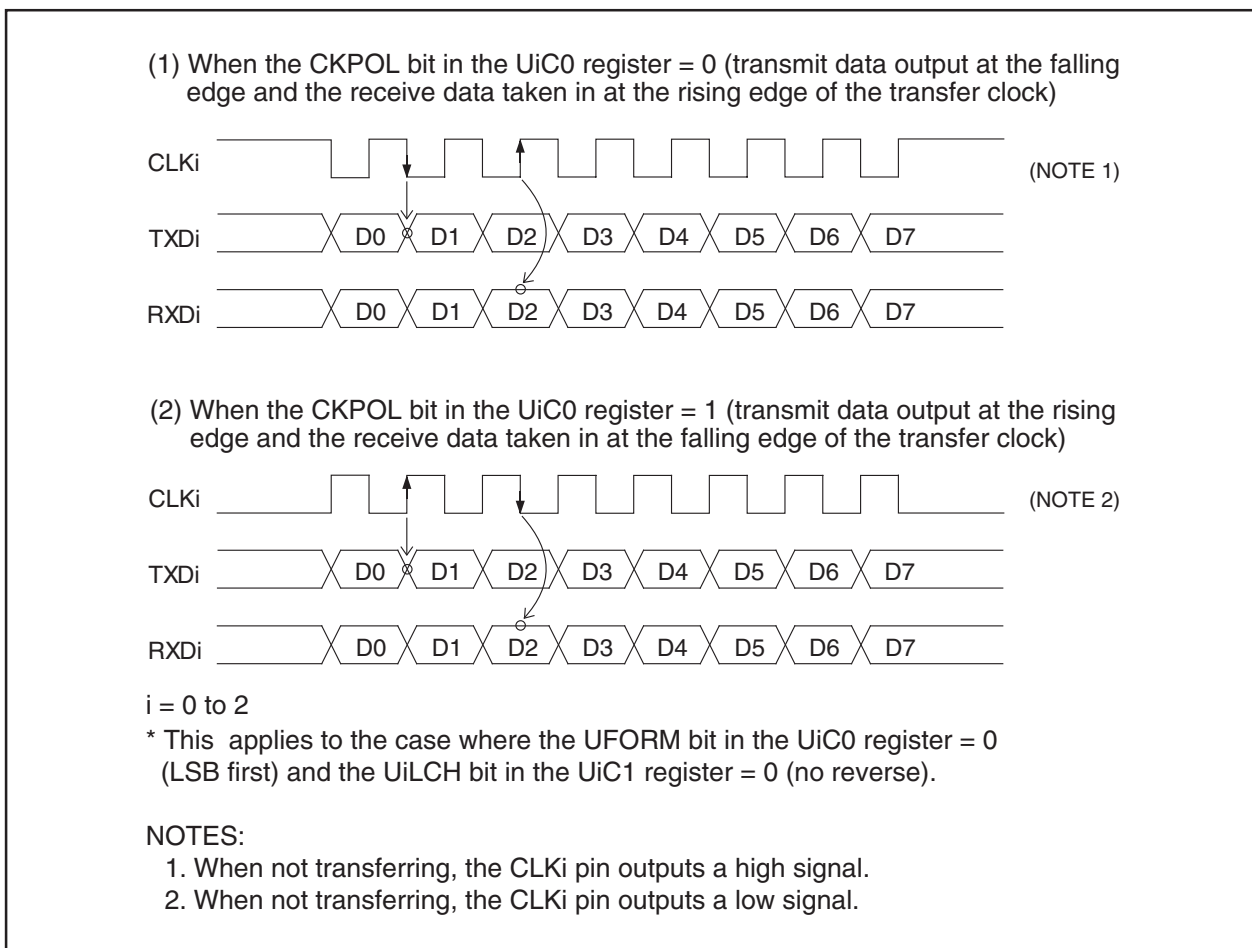
**15.1.1.1 Counter Measure for Communication Error Occurs**

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

- Resetting the UiRB register ( $i = 0$  to 2)
  - (1) Set the RE bit in the UiC1 register to “0” (reception disabled)
  - (2) Set the SMD2 to SMD0 bits in the UiMR register to “000b” (serial interface disabled)
  - (3) Set the SMD2 to SMD0 bits in the UiMR register to “001b” (clock synchronous serial I/O mode)
  - (4) Set the RE bit in the UiC1 register to “1” (reception enabled)
- Resetting the UiTB register ( $i = 0$  to 2)
  - (1) Set the SMD2 to SMD0 bits in the UiMR register to “000b” (serial interface disabled)
  - (2) Set the SMD2 to SMD0 bits in the UiMR register to “001b” (clock synchronous serial I/O mode)
  - (3) “1” (transmission enabled) is written to the TE bit in the UiC1 register, regardless of the TE bit

**15.1.1.2 CLK Polarity Select Function**

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



**Figure 15.12 Transfer Clock Polarity**

### 15.1.1.3 LSB First/MSB First Select Function

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

Figure 15.13 shows the transfer format.

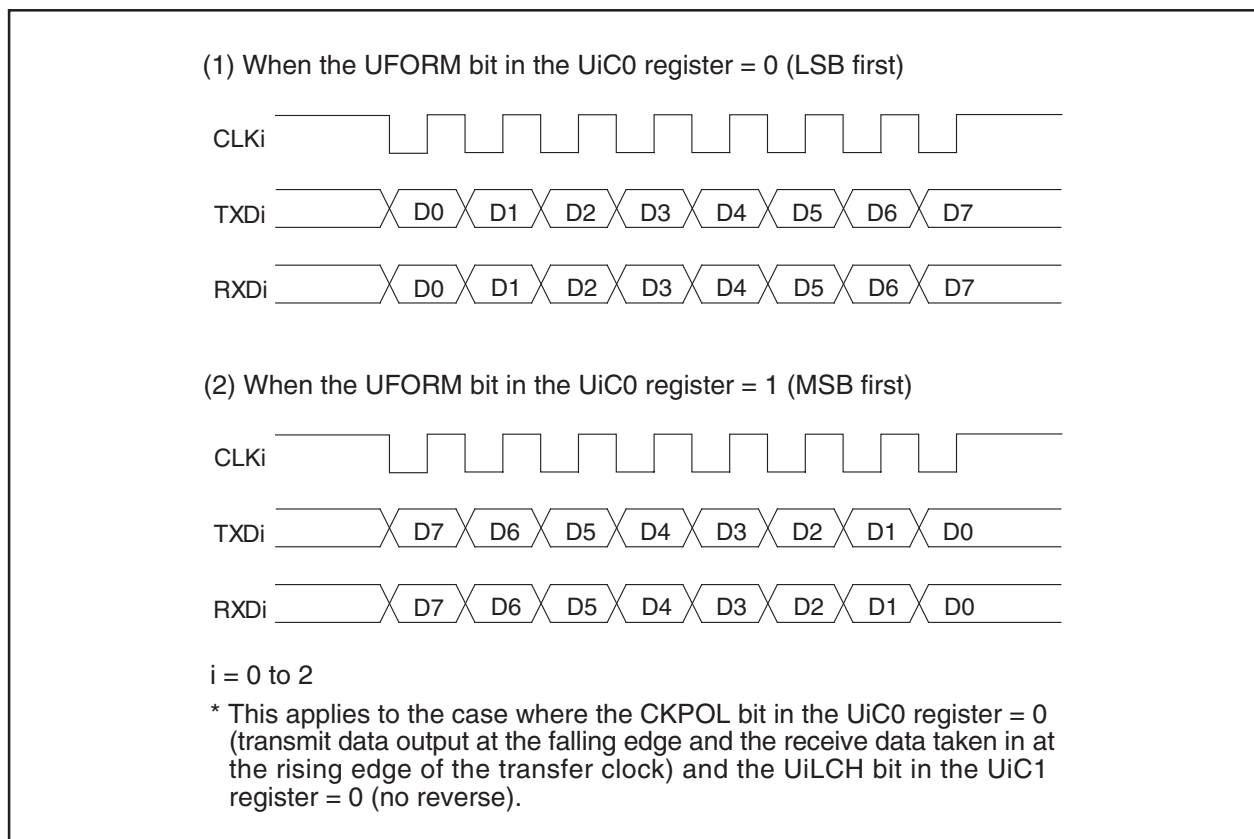


Figure 15.13 Transfer Format

### 15.1.1.4 Continuous Receive Mode

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

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

### 15.1.1.5 Serial Data Logic Switching Function

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

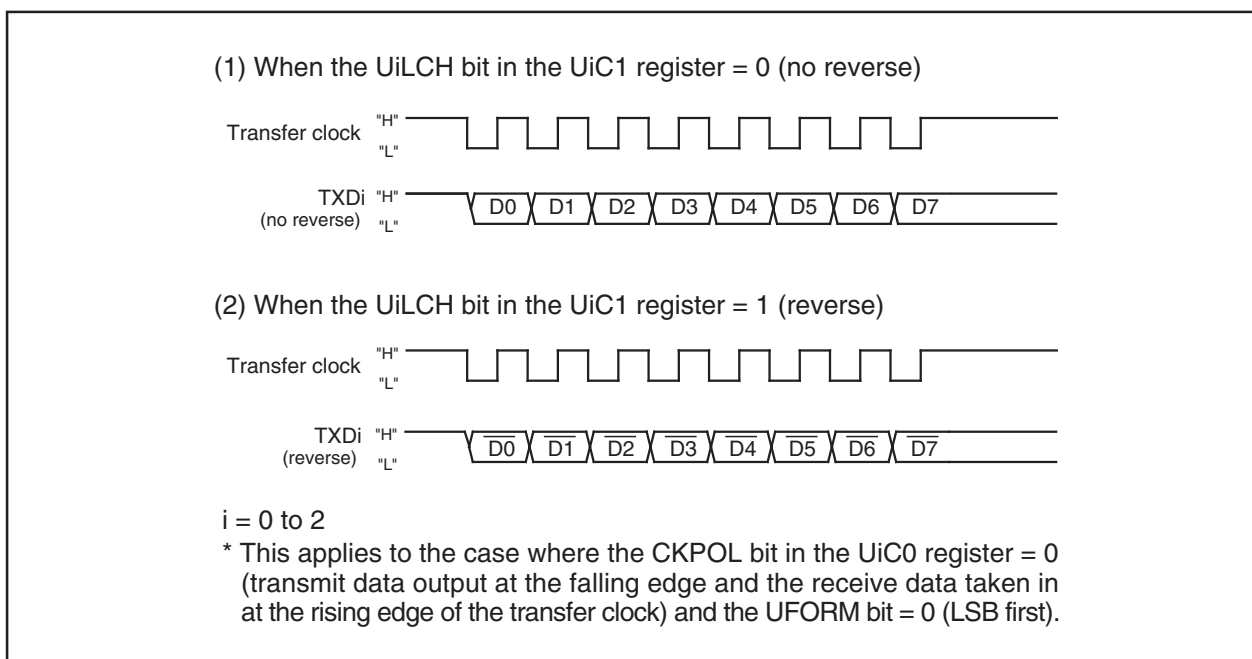


Figure 15.14 Serial Data Logic Switching

### 15.1.1.6 Transfer Clock Output From Multiple Pins (UART1)

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

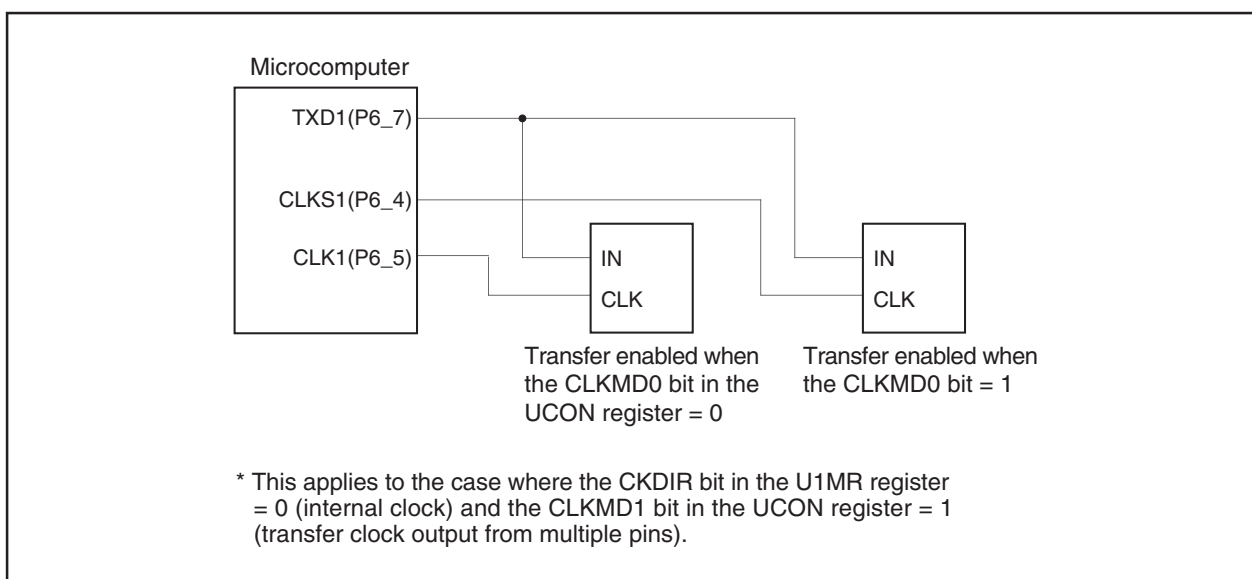


Figure 15.15 Transfer Clock Output From Multiple Pins

**15.1.1.7 CTS/RTS Function**

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

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

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

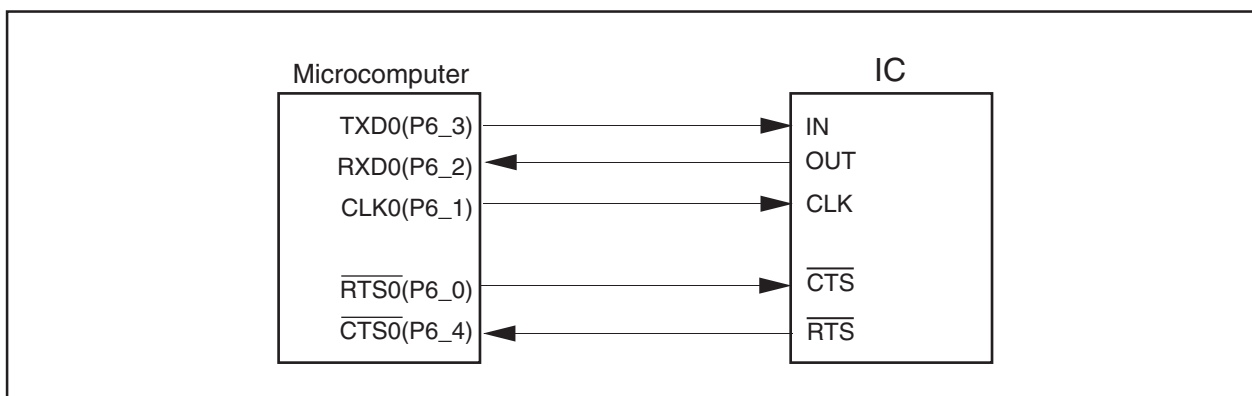
**15.1.1.8 CTS/RTS Separate Function (UART0)**

This function separates CTS0/RTS0, outputs RTS0 from the P6\_0 pin, and accepts as input the CTS0 from the P6\_4 pin. To use this function, set the register bits as shown below.

- CRD bit in U0C0 register = 0 (enables UART0  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- CRS bit in U0C0 register = 1 (outputs UART0 RTS)
- CRD bit in U1C0 register = 0 (enables UART1  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- CRS bit in U1C0 register = 0 (inputs UART1  $\overline{\text{CTS}}$ )
- RCSP bit in UCON register = 1 (inputs CTS0 from the P6\_4 pin)
- CLKMD1 bit in UCON register = 0 (CLKS1 not used)

Note that when using the CTS/RTS separate function, UART1 CTS/RTS separate function cannot be used.

Figure 15.16 shows CTS/RTS separate function usage.



**Figure 15.16 CTS/RTS Separate Function**



### 15.1.2 Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Table 15.5 lists the specifications of the UART mode. Table 15.6 lists the registers used in UART mode and the register values set.

**Table 15.5 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>CKDIR bit in UiMR register = 0 (internal clock) : <math>f_j / 16(n+1)</math>  <math>f_j = f1SIO, f2SIO, f8SIO, f32SIO</math>. n: Setting value of the UiBRG register 00h to FFh</li> <li>The CKDIR bit = 1 (external clock) : <math>fEXT/16(n+1)</math>  <math>fEXT</math>: Input from CLKi pin. n: Setting value of the UiBRG register 00h to FFh</li> </ul>
Transmission, Reception Control	Selectable from CTS function, RTS function or CTS/RTS function disabled
Transmission Start Condition	Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register = 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 on the CTSi pin = L</li> </ul>
Reception Start Condition	Before reception can start, the following requirements must be met <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register = 1 (reception enabled)</li> <li>Start bit detection</li> </ul>
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> <li>The UiIRS bit <sup>(1)</sup> = 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 = 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register</li> </ul> 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>
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 bit one before the last stop bit of the next data</li> <li>Framing error <sup>(3)</sup> This error occurs when the number of stop bits set is not detected</li> <li>Parity error <sup>(3)</sup> 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 to "1" when any of the overrun, framing, or parity errors occur</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 This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed.</li> <li>TXD, RXD I/O polarity switch This function reverses the polarities of the TXD pin output and RXD pin input. The logic levels of all I/O data is reversed.</li> <li>Separate CTS/RTS pins (UART0) CTS0 and RTS0 are input/output from separate pins</li> </ul>

i = 0 to 2

**NOTES:**

1. The U0IRS and U1IRS bits are bits 0 and 1 in the UCON register. The U2IRS bit is bit 4 in the U2C1 register.
2. If an overrun error occurs, the value of the UiRB register will be indeterminate. The IR bit in the SiRIC register does not change.
3. The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UARTi receive register to the UiRB register.

**Table 15.6 Registers to Be Used and Settings in UART Mode**

Register	Bit	Function
UiTB	0 to 8	Set 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 "100b" when transfer data is 7-bit long Set these bits to "101b" when transfer data is 8-bit long Set these bits to "110b" when transfer data is 9-bit long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
	IOPOL	Select the TXD/RXD input/output polarity
UiC0	CLK0, 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-bit long. Set this bit to "0" when transfer data is 7- or 9-bit long.
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS <sup>(2)</sup>	Select the source of UART2 transmit interrupt
	U2RRM <sup>(2)</sup>	Set to "0"
	UiLCH	Set this bit to "1" to use inverted data logic
	UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because the CLKMD1 bit = 0
	CLKMD1	Set to "0"
	RCSP	Set this bit to "1" to accept as input the UART0 CTS0 signal from the P6_4 pin
	7	Set to "0"

i = 0 to 2

**NOTES:**

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

Table 15.7 lists the functions of the input/output pins during UART mode. Table 15.8 lists the P6\_4 pin functions during UART mode. Note that for a period from when the UART<sub>i</sub> operation mode is selected to when transfer starts, the TXD<sub>i</sub> pin outputs an “H”.

Figure 15.17 shows the typical transmit timings in UART mode. Figure 15.18 shows the typical receive timing in UART mode.

**Table 15.7 I/O Pin Functions**

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

i = 0 to 2

**Table 15.8 P6\_4 Pin Functions**

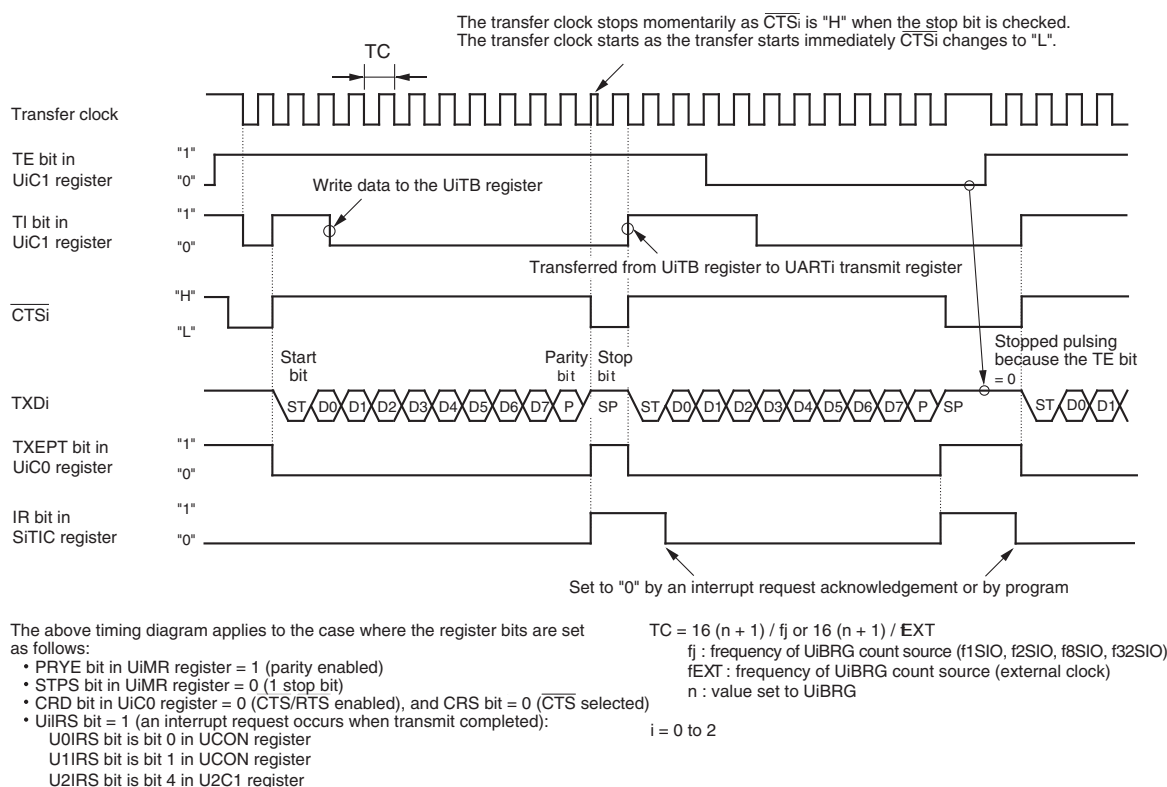
Pin Function	Bit set Value				
	U1C0 Register		UCON Register		PD6 Register
	CRD bit	CRS bit	RCSP bit	CLKMD1 bit	PD6_4 bit
P6_4	1	-	0	0	Input: 0, Output: 1
CTS1	0	0	0	0	0
RTS1	0	1	0	0	-
CTS0 <sup>(1)</sup>	0	0	1	0	0

-: “0” or “1”

**NOTE:**

1. In addition to this, set the CRD bit in the U0C0 register to “0” (CTS0/RTS0 enabled) and the CRS bit in the U0C0 register to “1” (RTS0 selected).

## (1) Example of Transmit Timing when Transfer Data is 8-bit Long (parity enabled, one stop bit)



## (2) Example of Transmit Timing when Transfer Data is 9-bit Long (parity disabled, two stop bits)

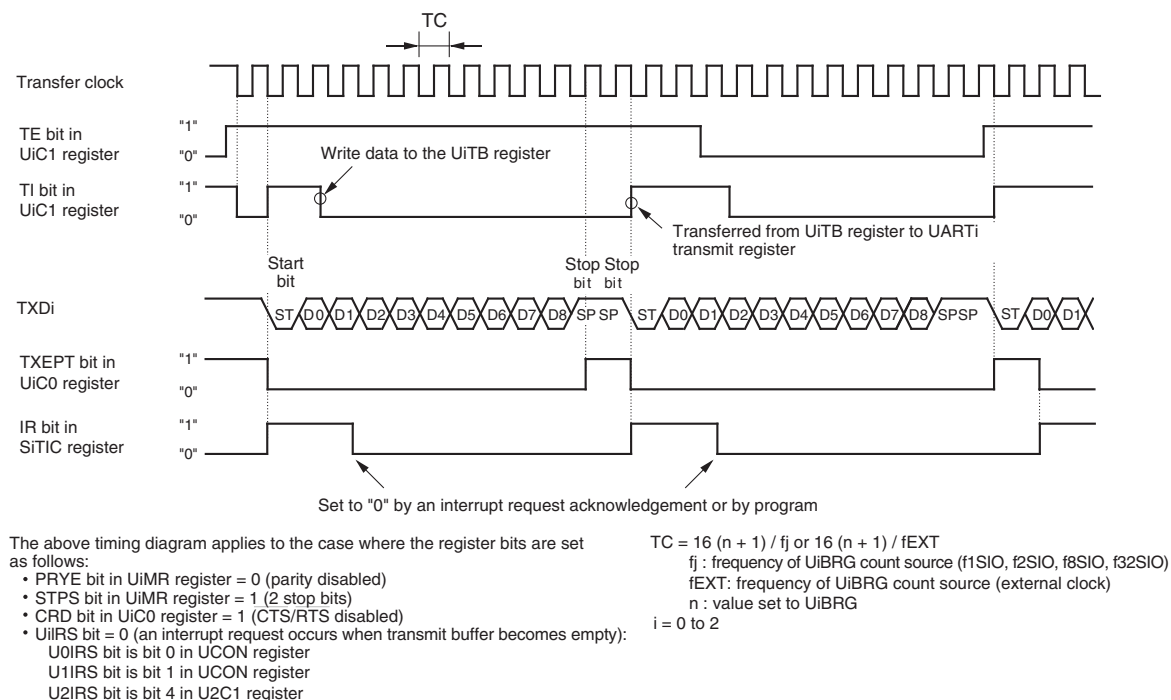


Figure 15.17 Transmit Operation

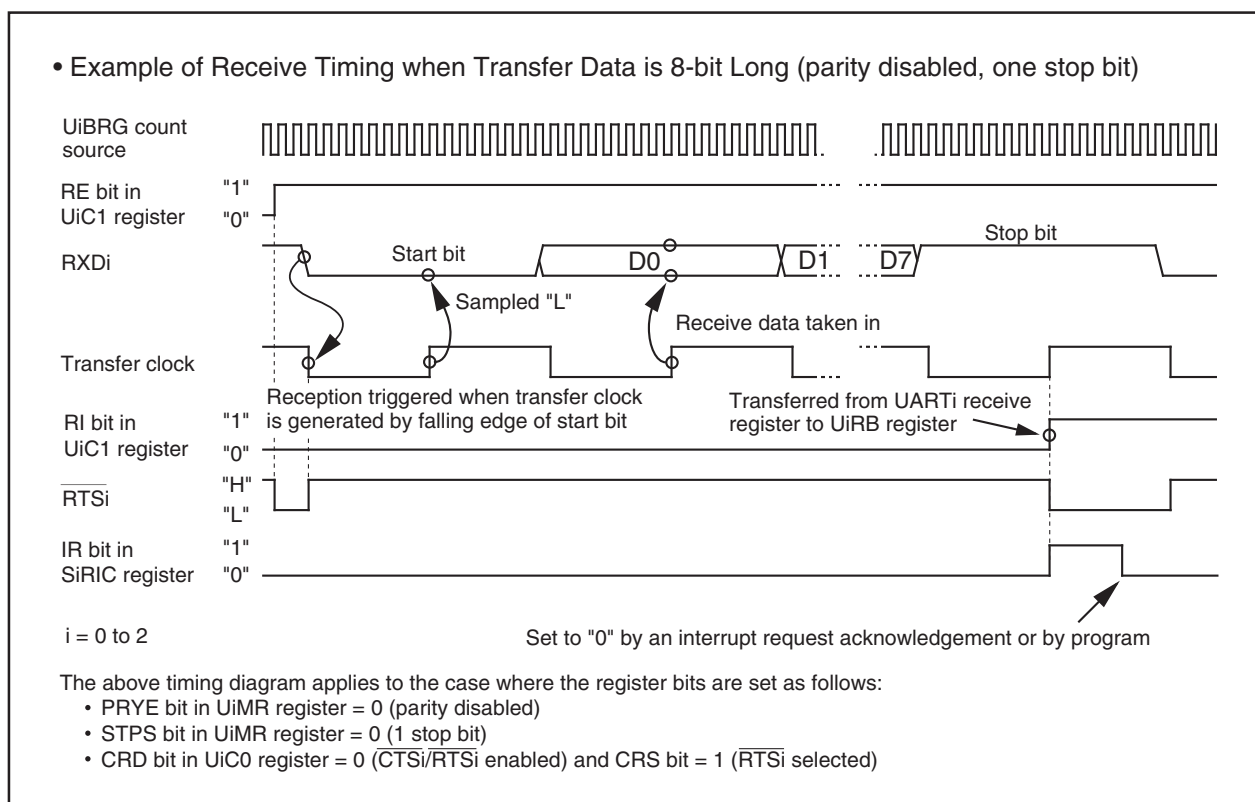


Figure 15.18 Receive Operation

### 15.1.2.1 Bit Rates

In UART mode, the frequency set by the UiBRG register (i = 0 to 2) divided by 16 become the bit rates.

Table 15.9 lists example of bit rates and settings.

Table 15.9 Example of Bit Rates and Settings

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

NOTE:

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

### 15.1.2.2 Counter Measure for Communication Error Occurs

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

- Resetting the UiRB register ( $i = 0$  to 2)
  - (1) Set the RE bit in the UiC1 register to “0” (reception disabled)
  - (2) Set the RE bit in the UiC1 register to “1” (reception enabled)
- Resetting the UiTB register ( $i = 0$  to 2)
  - (1) Set the SMD2 to SMD0 bits in the UiMR register to “000b” (serial interface disabled)
  - (2) Set the SMD2 to SMD0 bits in the UiMR register to “001b”, “101b”, “110b”
  - (3) “1” (transmission enabled) is written to the TE bit in the UiC1 register, regardless of the TE bit

### 15.1.2.3 LSB First/MSB First Select Function

As shown in Figure 15.19, use the UFORM bit in the UiC0 register to select the transfer format. This function is valid when transfer data is 8-bit long.

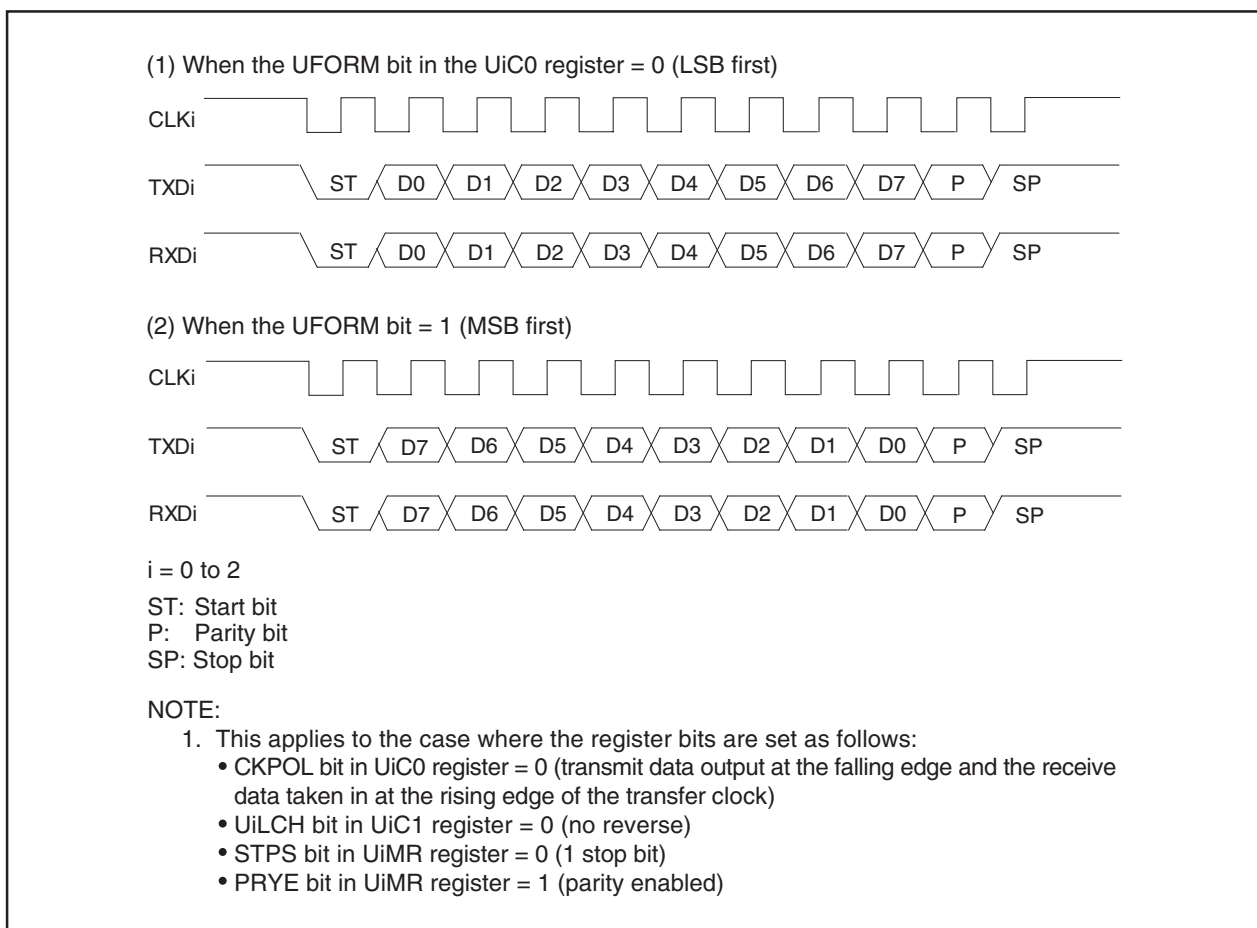


Figure 15.19 Transfer Format

### 15.1.2.4 Serial Data Logic Switching Function

The data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 15.20 shows serial data logic.

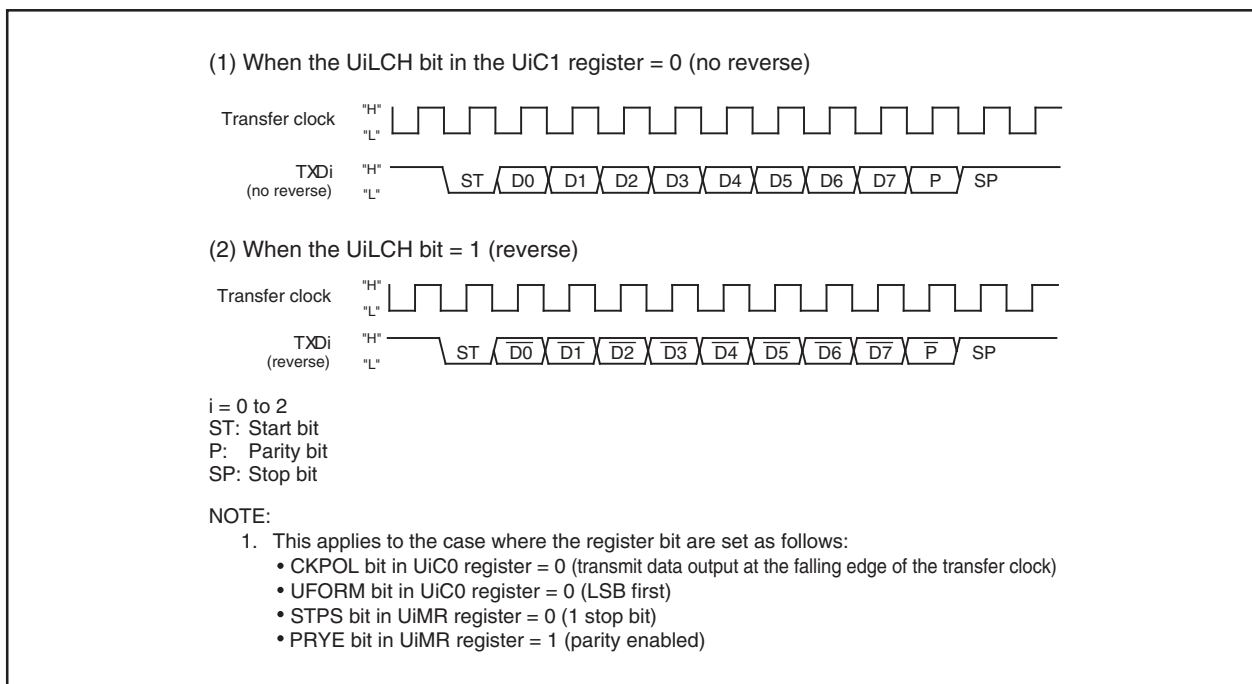


Figure 15.20 Serial Data Logic Switching

### 15.1.2.5 TXD and RXD I/O Polarity Inverse Function

This function inverses the polarities of the TXDi pin output and RXDi pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inverted. Figure 15.21 shows the TXD and RXD input/output polarity inverse.

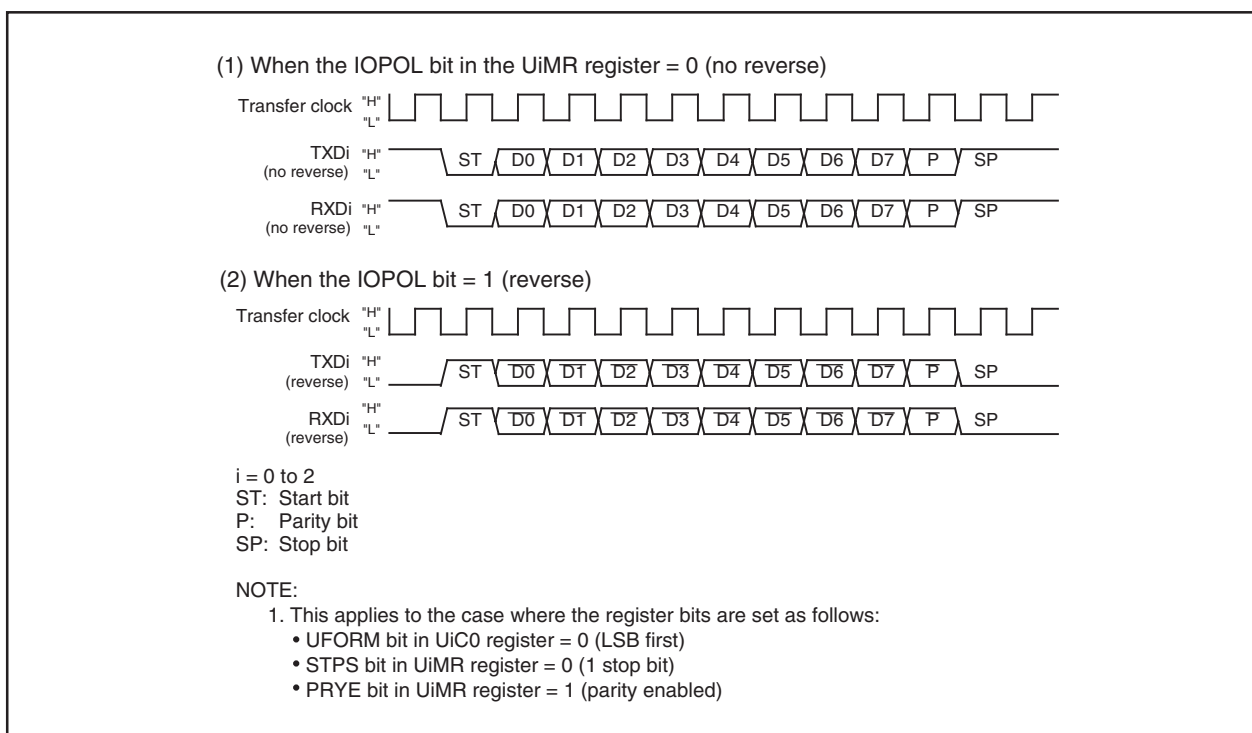


Figure 15.21 TXD and RXD I/O Polarity Inverse

**15.1.2.6 CTS/RTS Function**

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

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

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

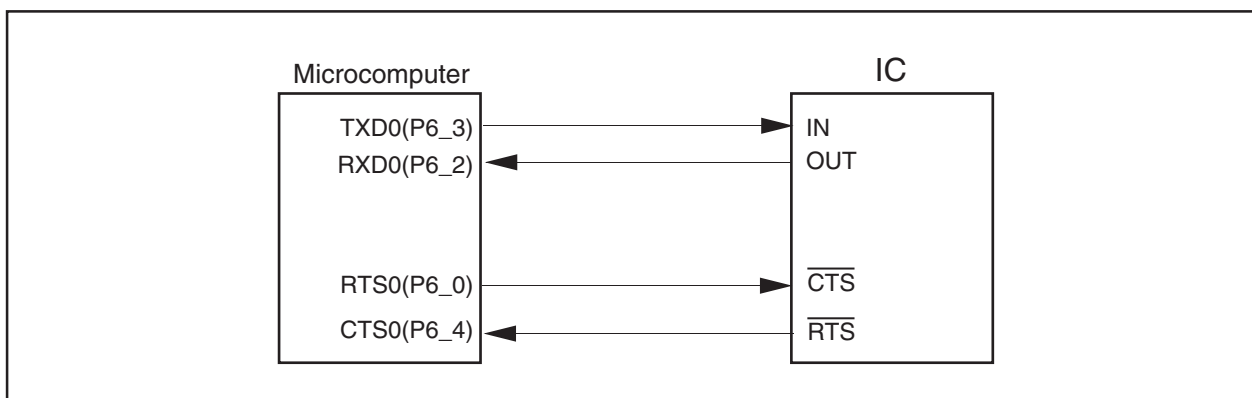
**15.1.2.7 CTS/RTS Separate Function (UART0)**

This function separates  $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$ , outputs  $\overline{\text{RTS}}_0$  from the P6\_0 pin, and accepts as input the  $\overline{\text{CTS}}_0$  from the P6\_4 pin. To use this function, set the register bits as shown below.

- CRD bit in  $\text{U0C0}$  register = 0 (enables UART0  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- CRS bit in  $\text{U0C0}$  register = 1 (outputs UART0  $\overline{\text{RTS}}$ )
- CRD bit in  $\text{U1C0}$  register = 0 (enables UART1  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- CRS bit in  $\text{U1C0}$  register = 0 (inputs UART1  $\overline{\text{CTS}}$ )
- RCSP bit in  $\text{UCON}$  register = 1 (inputs  $\overline{\text{CTS}}_0$  from the P6\_4 pin)
- $\text{CLKMD1}$  bit in  $\text{UCON}$  register = 0 ( $\text{CLKS1}$  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 15.22 shows  $\overline{\text{CTS}}/\overline{\text{RTS}}$  separate function usage.



**Figure 15.22 CTS/RTS Separate Function**



### 15.1.3 Special Mode 1 (I<sup>2</sup>C Mode)

I<sup>2</sup>C mode is provided for use as a simplified I<sup>2</sup>C interface compatible mode. Table 15.10 lists the specifications of the I<sup>2</sup>C mode. Figure 15.23 shows the block diagram for I<sup>2</sup>C mode. Table 15.11 lists the registers used in the I<sup>2</sup>C mode and the register values set. Table 15.12 lists the functions in I<sup>2</sup>C mode. Figure 15.24 shows the transfer to the UiRB register and interrupt timing.

As shown in Table 15.12, the microcomputer is placed in I<sup>2</sup>C mode by setting the SMD2 to SMD0 bits to “010b” and the IICM bit to “1”. Because SDAi transmit output has a delay circuit attached, SDAi output does not change state until SCLi goes low and remains stably low.

**Table 15.10 I<sup>2</sup>C Mode Specifications**

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	<ul style="list-style-type: none"> <li>During master           <ul style="list-style-type: none"> <li>The CKDIR bit in the UiMR register = 0 (internal clock) : <math>f_j / 2(n+1)</math>  <math>f_j = f1SIO, f2SIO, f8SIO, f32SIO</math>. n: Setting value of the UiBRG register 00h to FFh</li> </ul> </li> <li>During slave           <ul style="list-style-type: none"> <li>The CKDIR bit = 1 (external clock) : Input from SCLi pin</li> </ul> </li> </ul>
Transmission Start Condition	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 = 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register = 0 (data present in the UiTB register)</li> </ul>
Reception Start Condition	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 = 1 (reception enabled)</li> <li>The TE bit in the UiC1 register = 1 (transmission enabled)</li> <li>The TI bit in the UiC1 register = 0 (data present in the UiTB register)</li> </ul>
Interrupt Request Generation Timing	When start or stop condition is detected, acknowledge undetected, and acknowledge detected
Error Detection	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 8th bit of the next data
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 UiRB register is updated can be selected</li> </ul> </li> <li>SDAi digital delay           <ul style="list-style-type: none"> <li>No digital delay or a delay of 2 to 8 UiBRG 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>

i = 0 to 2

**NOTES:**

1. When an external clock is selected, the conditions must be met while the external clock is in the high state.
2. If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit in the SiRIC register does not change.

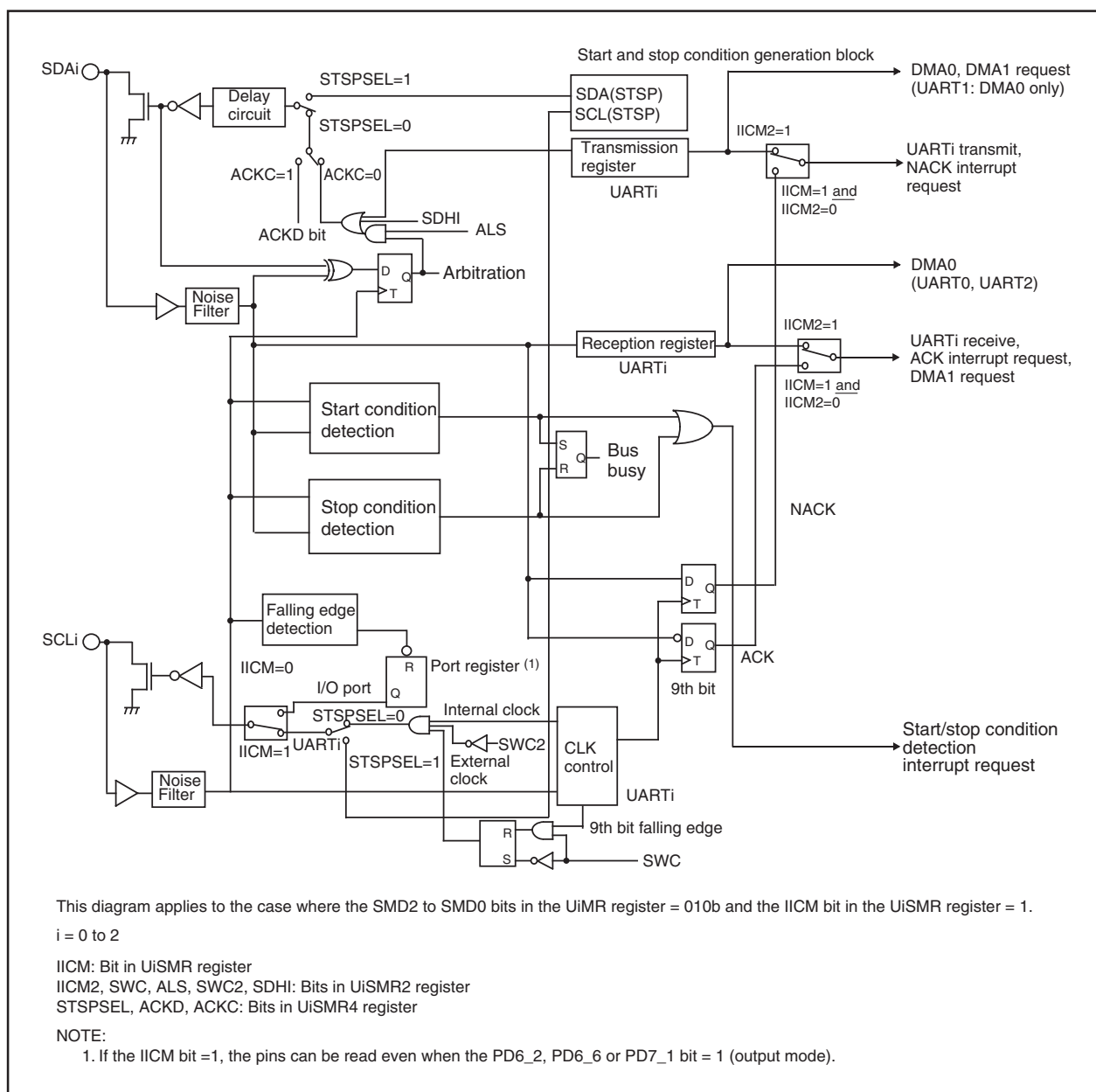


Figure 15.23 I²C Mode Block Diagram

**Table 15.11 Registers to Be Used and Settings in I<sup>2</sup>C Mode**

Register	Bit	Function	
		Master	Slave
UiTB (1)	0 to 7	Set transmission data	
UiRB (1)	0 to 7	Reception data can be read	
	8	ACK or NACK is set in this bit	
	ABT	Arbitration lost detection flag	Invalid
	OER	Overrun error flag	
UiBRG	0 to 7	Set a transfer rate	Invalid
UiMR (1)	SMD2 to SMD0	Set to "010b"	
	CKDIR	Set to "0"	Set to "1"
	IOPOL	Set to "0"	
UiC0	CLK1, CLK0	Select the count source for the UiBRG register	Invalid
	CRS	Invalid because the CRD bit = 1	
	TXEPT	Transmit register empty flag	
	CRD	Set to "1"	
	NCH	Set to "1"	
	CKPOL	Set to "0"	
	UFORM	Set to "1"	
UiC1	TE	Set this bit to "1" to enable transmission	
	TI	Transmit buffer empty flag	
	RE	Set this bit to "1" to enable reception	
	RI	Reception complete flag	
	U2IRS (2)	Invalid	
	U2RRM (2), UiLCH, UiERE	Set to "0"	
UiSMR	IICM	Set to "1"	
	ABC	Select the timing at which arbitration-lost is detected	Invalid
	BBS	Bus busy flag	
	3 to 7	Set to "0"	
UiSMR2	IICM2	See <b>Table 15.12 I<sup>2</sup>C Mode Functions</b>	
	CSC	Set this bit to "1" to enable clock synchronization	Set to "0"
	SWC	Set this bit to "1" to have SCLi output fixed to "L" at the falling edge of the 9th bit of clock	
	ALS	Set this bit to "1" to have SDAi output stopped when arbitration-lost is detected	Set to "0"
	STAC	Set to "0"	Set this bit to "1" to initialize UARTi at start condition detection
	SWC2	Set this bit to "1" to have SCLi output forcibly pulled low	
	SDHI	Set this bit to "1" to disable SDAi output	
	7	Set to "0"	
UiSMR3	0, 2, 4 and NODC	Set to "0"	
	CKPH	See <b>Table 15.12 I<sup>2</sup>C Mode Functions</b>	
	DL2 to DL0	Set the amount of SDAi digital delay	
UiSMR4	STAREQ	Set this bit to "1" to generate start condition	Set to "0"
	RSTAREQ	Set this bit to "1" to generate restart condition	Set to "0"
	STPREQ	Set this bit to "1" to generate stop condition	Set to "0"
	STSPSEL	Set this bit to "1" to output each condition	Set to "0"
	ACKD	Select ACK or NACK	
	ACKC	Set this bit to "1" to output ACK data	
	SCLHI	Set this bit to "1" to have SCLi output stopped when stop condition is detected	Set to "0"
	SWC9	Set to "0"	Set this bit to "1" to set the SCLi to "L" hold at the falling edge of the 9th bit of clock
IFSR0	IFSR06, ISFR07	Set to "1"	
UCON	U0IRS, U1IRS	Invalid	
	2 to 7	Set to "0"	

i = 0 to 2

## NOTES:

1. Not all register bits are described above. Set those bits to "0" when writing to the registers in I<sup>2</sup>C mode.
2. Set the bit 4 and bit 5 in the U0C1 and U1C1 registers to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.

**Table 15.12 I<sup>2</sup>C Mode Functions**

Function	Clock Synchronous Serial I/O Mode (SMD2 to SMD0 = 001b, IICM = 0)	I <sup>2</sup> C Mode (SMD2 to SMD0 = 010b, IICM = 1)			
		IICM2 = 0 (NACK/ACK interrupt)		IICM2 = 1 (UART transmit/receive interrupt)	
		CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)	CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)
Factor of Interrupt Number 6, 7 and 10 <sup>(1) (5) (7)</sup>	-	Start condition detection or stop condition detection (See Table 15.13 <b>STSPSEL Bit Functions</b> )			
Factor of Interrupt Number 15, 17 and 19 <sup>(1) (6)</sup>	UARTi transmission Transmission started or completed (selected by UiIRS)	No acknowledgment detection (NACK) Rising edge of SCLi 9th bit		UARTi transmission Rising edge of SCLi 9th bit	UARTi transmission Falling edge of SCLi next to the 9th bit
Factor of Interrupt Number 16, 18 and 20 <sup>(1) (6)</sup>	UARTi reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) Rising edge of SCLi 9th bit		UARTi reception Falling edge of SCLi 9th bit	
Timing for Transferring Data from UART Reception Shift Register to UiRB Register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Rising edge of SCLi 9th bit		Falling edge of SCLi 9th bit	Falling and rising edges of SCLi 9th bit
UARTi Transmission Output Delay	Not delayed	Delayed			
Functions of P6_3, P6_7 and P7_0 Pins	TXDi output	SDAi input/output			
Functions of P6_2, P6_6 and P7_1 Pins	RXDi input	SCLi input/output			
Functions of P6_1, P6_5 and P7_2 Pins	CLKi input or output selected	- (Cannot be used in I <sup>2</sup> C mode)			
Noise Filter Width	15 ns	200 ns			
Read RXDi and SCLi Pins Levels	Possible when the corresponding port direction bit = 0	Always possible no matter how the corresponding port direction bit is set			
Initial Value of TXDi and SDAi Outputs	CKPOL = 0 (H) CKPOL = 1 (L)	The value set in the port register before setting I <sup>2</sup> C mode <sup>(2)</sup>			
Initial and End Value of SCLi	-	H	L	H	L
DMA1 Factor <sup>(6)</sup>	UARTi reception	Acknowledgment detection (ACK)		UARTi reception Falling edge of SCLi 9th bit	
Store Received Data	1st to 8th bits of the received data are stored into bit 7 to bit 0 in the UiRB register			1st to 7th bits of the received data are stored into bit 6 to bit 0 in the UiRB register, 8th bit is stored into bit 8 in the UiRB register	1st to 8th bits are stored into bit 7 to bit 0 in UiRB register <sup>(3)</sup>
Read Received Data	The UiRB register status is read				Bit 6 to bit 0 in the UiRB register <sup>(4)</sup> are read as bit 7 to bit 1. Bit 8 in the UiRB register is read as bit 0.

i = 0 to 2

## NOTES:

- If the 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 23.5 Interrupts.)  
If one of the bits shown below is changed, the interrupt source, the interrupt timing, etc. change. Therefore, always be sure to set the IR bit to "0" (interrupt not requested) after changing those bits.
  - SMD2 to SMD0 bits in UiMR register
  - IICM2 bit in UiSMR2 register
  - IICM bit in UiSMR register
  - CKPH bit in UiSMR3 register
- Set the initial value of SDAi output while the SMD2 to SMD0 bits in the UiMR register = 000b (serial interface disabled).
- Second data transfer to the UiRB register (rising edge of SCLi 9th bit)
- First data transfer to the UiRB register (falling edge of SCLi 9th bit)
- See Figure 15.26 STSPSEL Bit Functions.
- See Figure 15.24 Transfer to UiRB Register and Interrupt Timing.
- When using UART0, be sure to set the IFSR06 bit in the IFSR0 register to "1" (cause of interrupt: UART0 bus collision detection).  
When using UART1, be sure to set the IFSR07 bit in the IFSR0 register to "1" (cause of interrupt: UART1 bus collision detection).

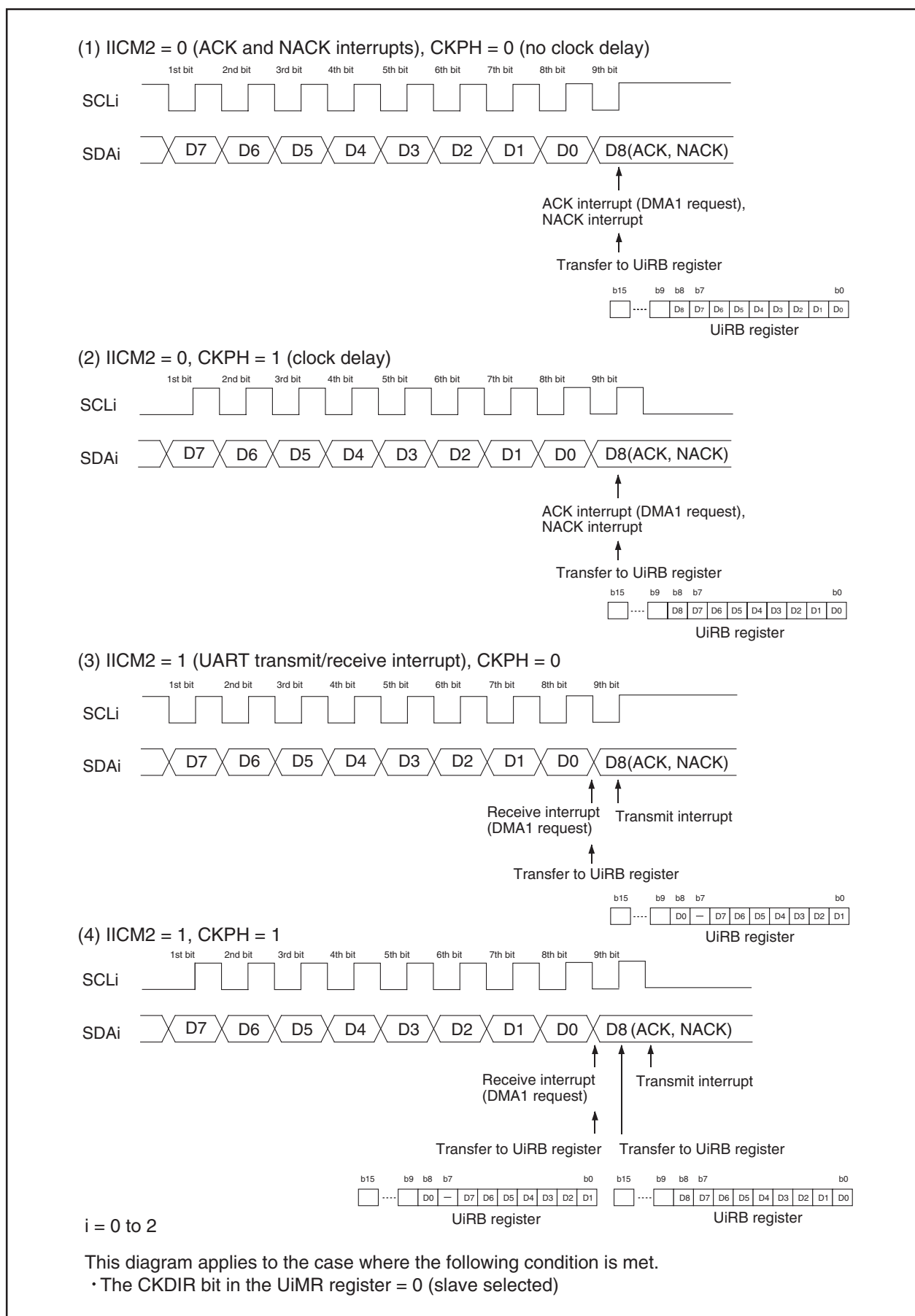


Figure 15.24 Transfer to UiRB Register and Interrupt Timing

### 15.1.3.1 Detection of Start and Stop Condition

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

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

Figure 15.25 shows the detection of start and stop condition.

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

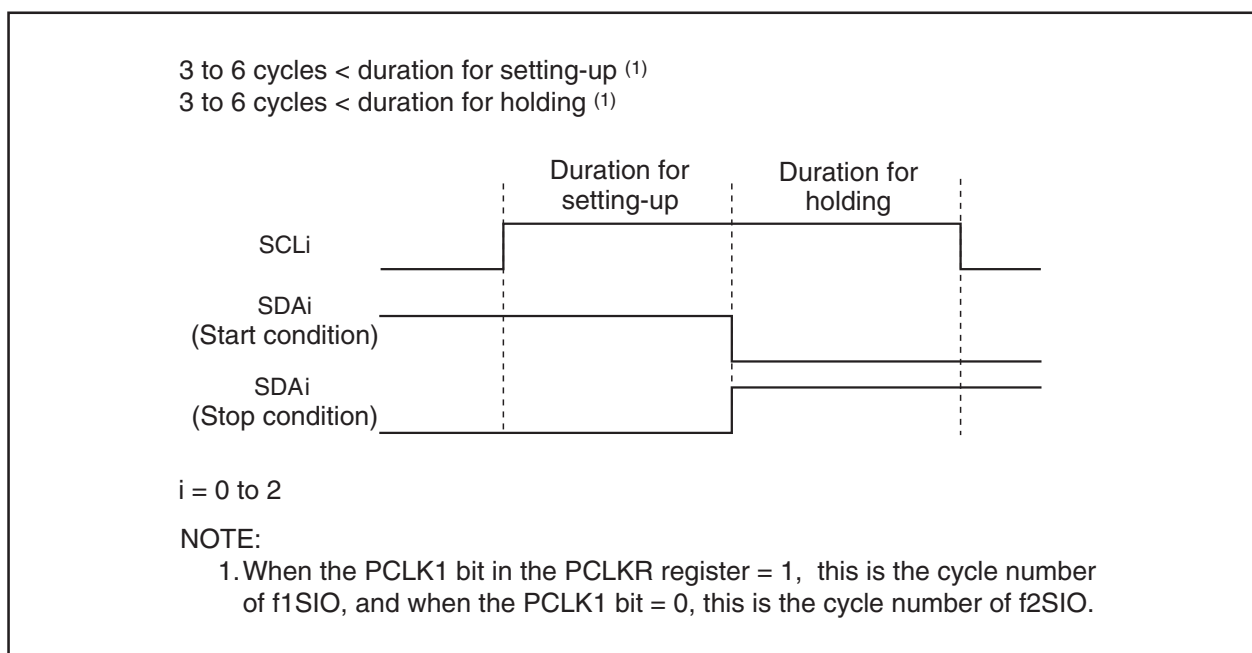


Figure 15.25 Detection of Start and Stop Condition

### 15.1.3.2 Output of Start and Stop Condition

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

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

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

The output procedure is described below.

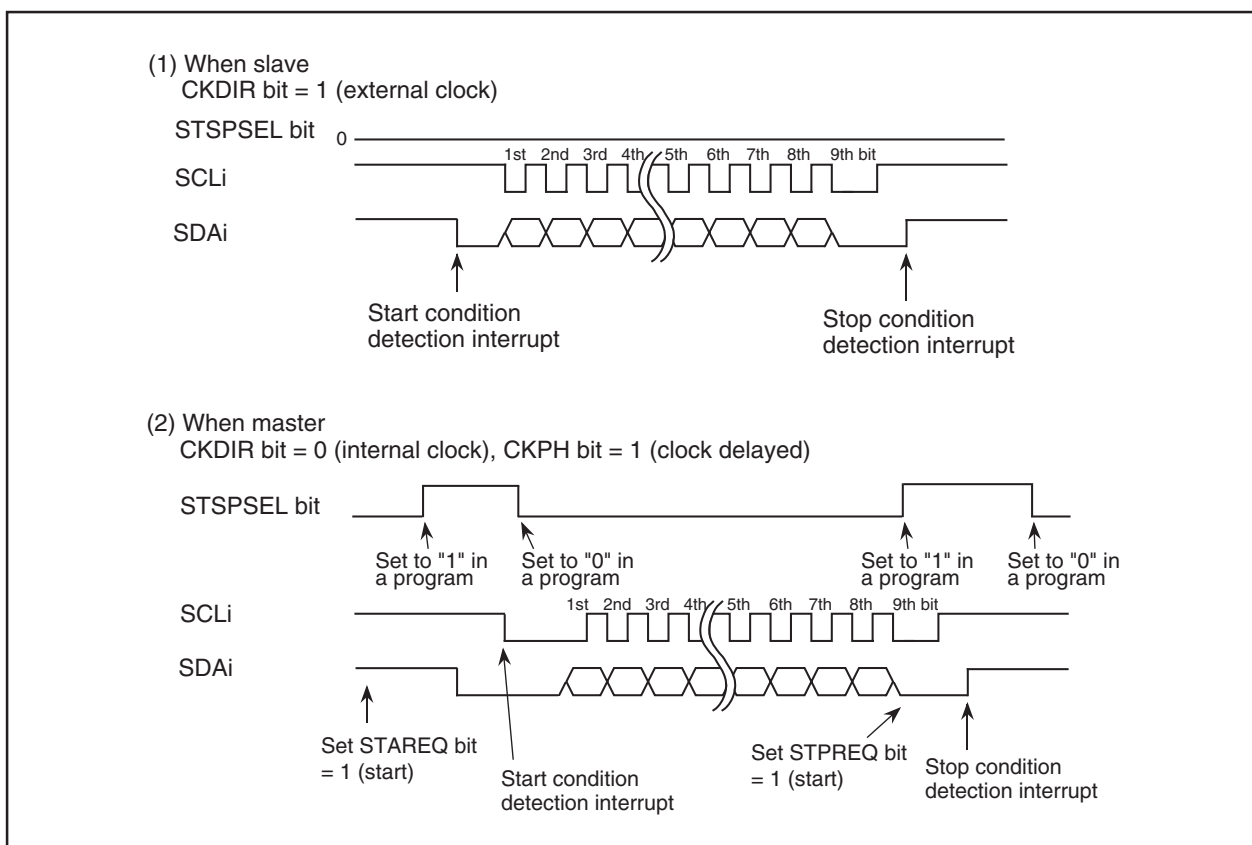
(1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to "1" (start).

(2) Set the STSPSEL bit in the UiSMR4 register to "1" (output).

Table 15.13 and Figure 15.26 show the functions of the STSPSEL bit.

**Table 15.13 STSPSEL Bit Functions**

Function	STSPSEL Bit = 0	STSPSEL Bit = 1
Output of SCLi and SDAi Pins	Output of transfer clock and data Output of start/stop condition is accomplished by a program using ports (not automatically generated in hardware)	Output of a start/stop condition according to the STAREQ, RSTAREQ and STPREQ bits
Start/Stop Condition Interrupt Request Generation Timing	Start/stop condition detection	Finish generating start/stop condition



**Figure 15.26 STSPSEL Bit Functions**

### 15.1.3.3 Arbitration

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

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

#### 15.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 15.24.

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

In this way, the UARTi transfer clock is comprised of the logical product of the internal SCLi and SCLi pin signal. The transfer clock works from a half period before the falling edge of the internal SCLi 1st bit to the rising edge of the 9th bit. To use this function, select an internal clock for the transfer clock.

The SWC bit in the UiSMR2 register allows to select whether the SCLi pin should be fixed to or freed from low-level output at the falling edge of the 9th clock pulse.

If the SCLHI bit in the UiSMR4 register is set to "1" (enabled), SCLi output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the SWC2 bit in the UiSMR2 register = 1 (0 output) makes it possible to forcibly output a low-level signal from the SCLi pin even while sending or receiving data. Setting the SWC2 bit to "0" (transfer clock) allows the transfer clock to be output from or supplied to the SCLi pin, instead of outputting a low-level signal.

If the SWC9 bit in the UiSMR4 register is set to "1" (SCL hold low enabled) when the CKPH bit in the UiSMR3 register = 1, the SCLi pin is fixed to low-level output at the falling edge of the clock pulse next to the ninth. Setting the SWC9 bit = 0 (SCL hold low disabled) frees the SCLi pin from low-level output.

#### 15.1.3.5 SDA Output

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

The initial value of SDAi transmit output can only be set when IICM = 1 (I<sup>2</sup>C mode) and the SMD2 to SMD0 bits in the UiMR register = 000b (serial interface disabled).

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

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

#### 15.1.3.6 SDA Input

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

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



### 15.1.3.7 ACK and NACK

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

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

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

### 15.1.3.8 Initialization of Transmission/Reception

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

- The transmit shift register is initialized, and the content of the UiTB register is transferred to the transmit shift register. In this way, the serial I/O starts sending data synchronously with the next clock pulse applied. However, the UARTi output value does not change state and remains the same as when a start condition was detected until the first bit of data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial 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 SCLi pin is pulled low at the falling edge of the ninth clock pulse.

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

### 15.1.4 Special Mode 2

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. Table 15.14 lists the specifications of Special Mode 2. Figure 15.27 shows communication control example for Special Mode 2. Table 15.15 lists the registers used in Special Mode 2 and the register values set.

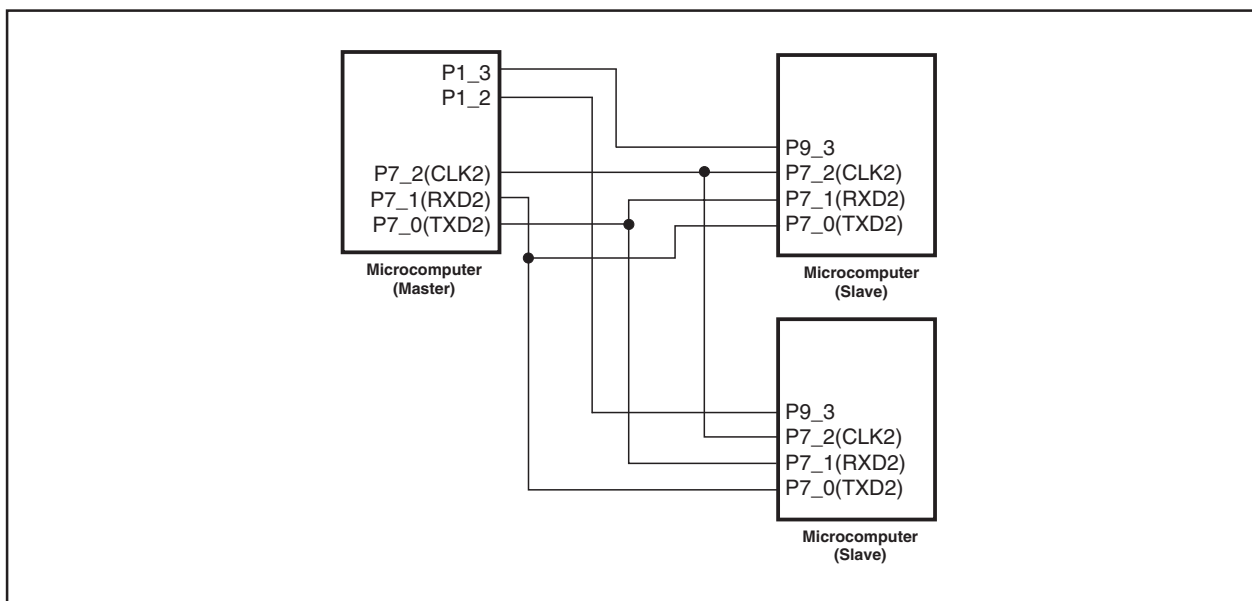
**Table 15.14 Special Mode 2 Specifications**

Item	Specification
Transfer data format	Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> <li>• Master mode The CKDIR bit in the UiMR register = 0 (internal clock) : <math>f_j / 2^{(n+1)}</math> <math>f_j = f1SIO, f2SIO, f8SIO, f32SIO</math>. n: Setting value of the UiBRG register 00h to FFh</li> <li>• Slave mode The CKDIR bit = 1 (external clock selected) : Input from CLKi pin</li> </ul>
Transmit/receive control	Controlled by input/output ports
Transmission start condition	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 = 1 (transmission enabled)</li> <li>• The TI bit in the UiC1 register = 0 (data present in the UiTB register)</li> </ul>
Reception start condition	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 = 1 (reception enabled)</li> <li>• The TE bit in the UiC1 register = 1 (transmission enabled)</li> <li>• The TI bit in the UiC1 register = 0 (data present in the UiTB register)</li> </ul>
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> <li>• The UiIRS bit <sup>(2)</sup> = 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 = 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register</li> </ul> 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>
Error detection	Overrun error <sup>(3)</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
Select function	Clock phase setting Selectable from four combinations of transfer clock polarities and phases

i = 0 to 2

**NOTES:**

1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. The U0IRS and U1IRS bits respectively are bits 0 and 1 in the UCON register ; the U2IRS bit is bit 4 in the U2C1 register.
3. If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit in SiRIC register does not change.

**Figure 15.27 Serial Bus Communication Control Example (UART2)**

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

Register	Bit	Function
UiTB <sup>(1)</sup>	0 to 7	Set transmission data
UiRB <sup>(1)</sup>	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a transfer rate
UiMR <sup>(1)</sup>	SMD2 to SMD0	Set to "001b"
	CKDIR	Set this bit to "0" for master mode or "1" for slave mode
	IOPOL	Set to "0"
UiC0	CLK1, CLK0	Select the count source for the UiBRG register
	CRS	Invalid because the CRD bit = 1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Select TXDi pin output format
	CKPOL	Clock phases can be set in combination with the CKPH bit in the UiSMR3 register
	UFORM	Set to "0"
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS <sup>(2)</sup>	Select UART2 transmit interrupt cause
	U2RRM <sup>(2)</sup> , UiLCH, UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	CKPH	Clock phases can be set in combination with the CKPOL bit in the UiC0 register
	NODC	Set to "0"
	0, 2, 4 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select UART0 and UART1 transmit interrupt cause
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because the CLKMD1 bit = 0
	CLKMD1, RCSP, 7	Set to "0"

i = 0 to 2

**NOTES:**

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

### 15.1.4.1 Clock Phase Setting Function

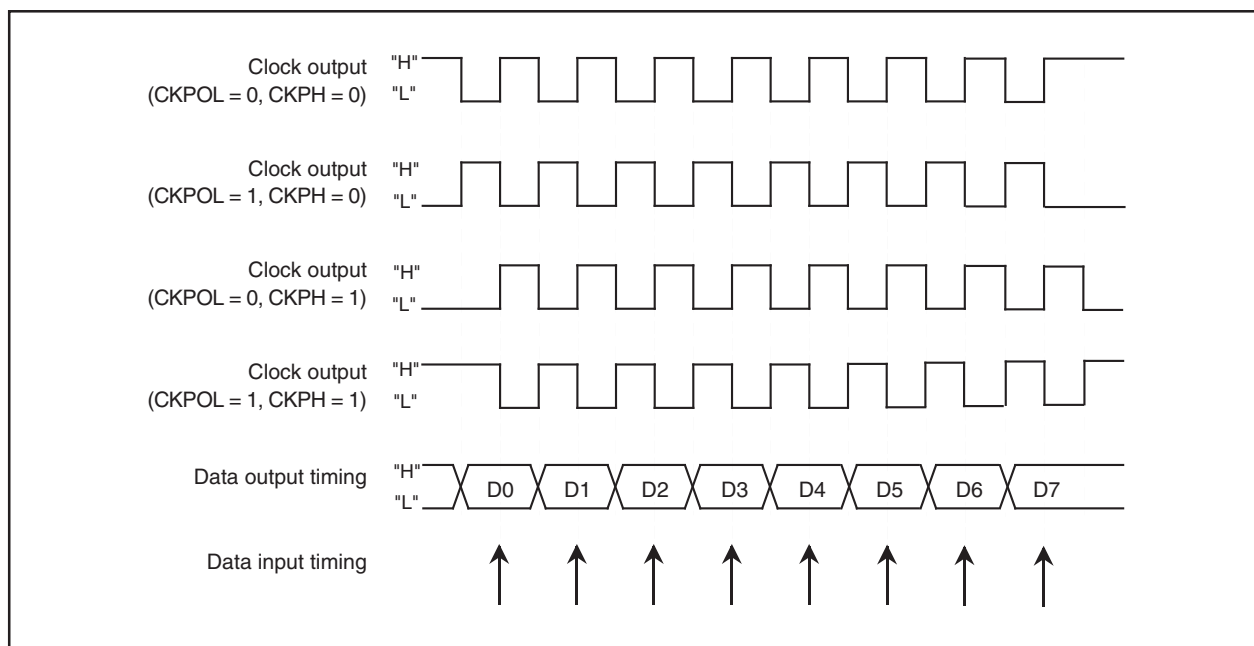
One of four combinations of transfer clock phases and polarities can be selected using the CKPH bit in the UiSMR3 register and the CKPOL bit in the UiC0 register.

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

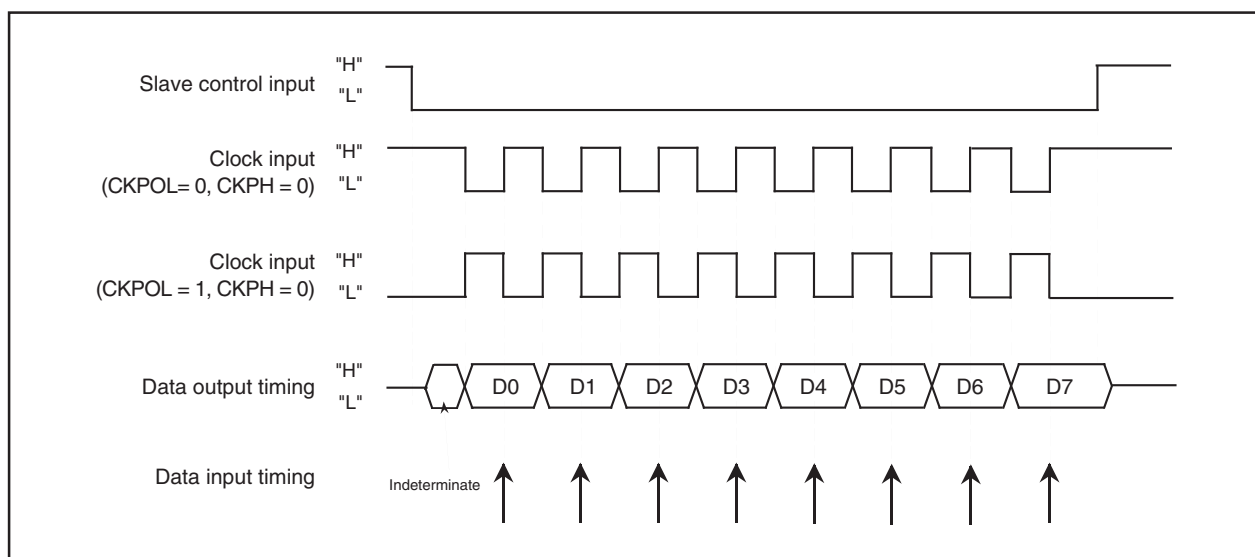
Figure 15.28 shows the transmission and reception timing in master (internal clock).

Figure 15.29 shows the transmission and reception timing (CKPH = 0) in slave (external clock).

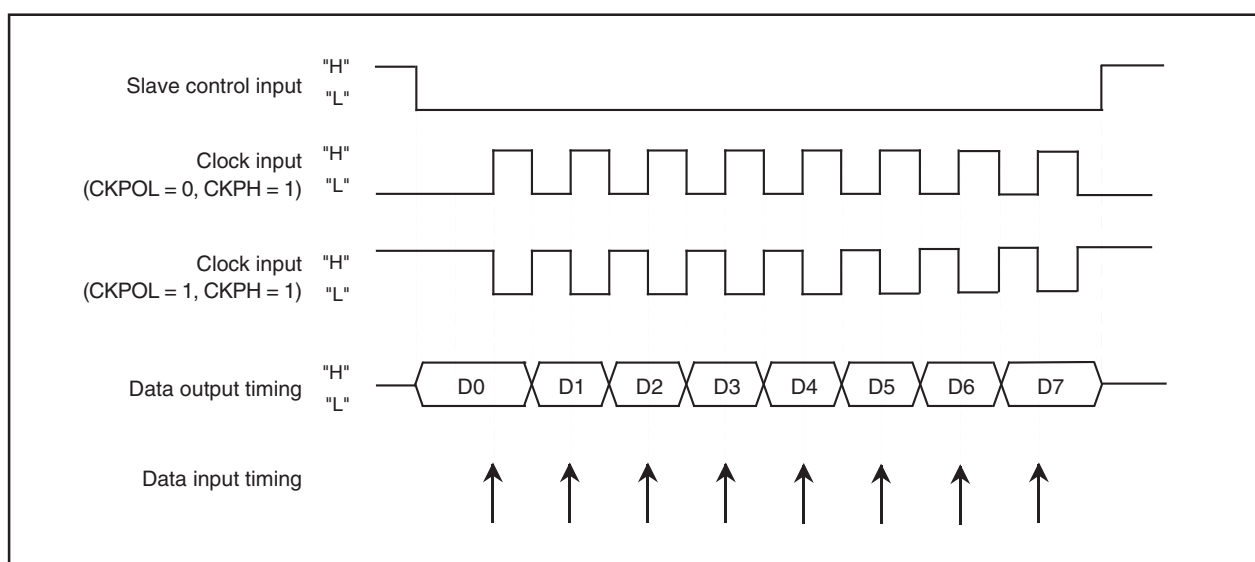
Figure 15.30 shows the transmission and reception timing (CKPH = 1) in slave (external clock).



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



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



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

### 15.1.5 Special Mode 3 (IE Mode)

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

Table 15.16 lists the registers used in IE mode and the register values set. Figure 15.31 shows the functions of bus collision detect function related bits.

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

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

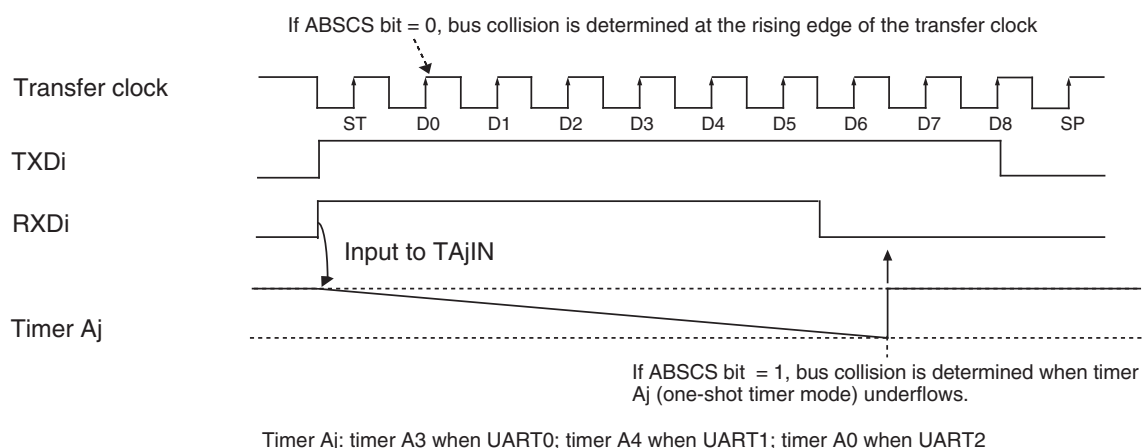
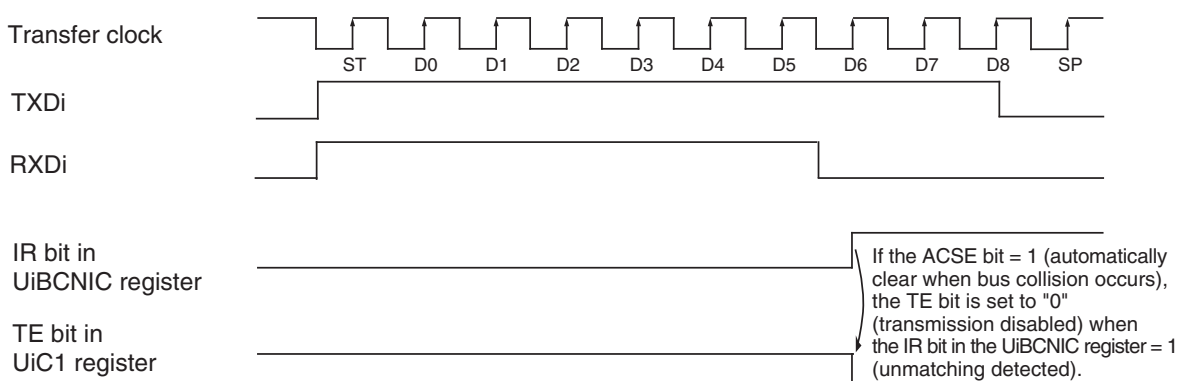
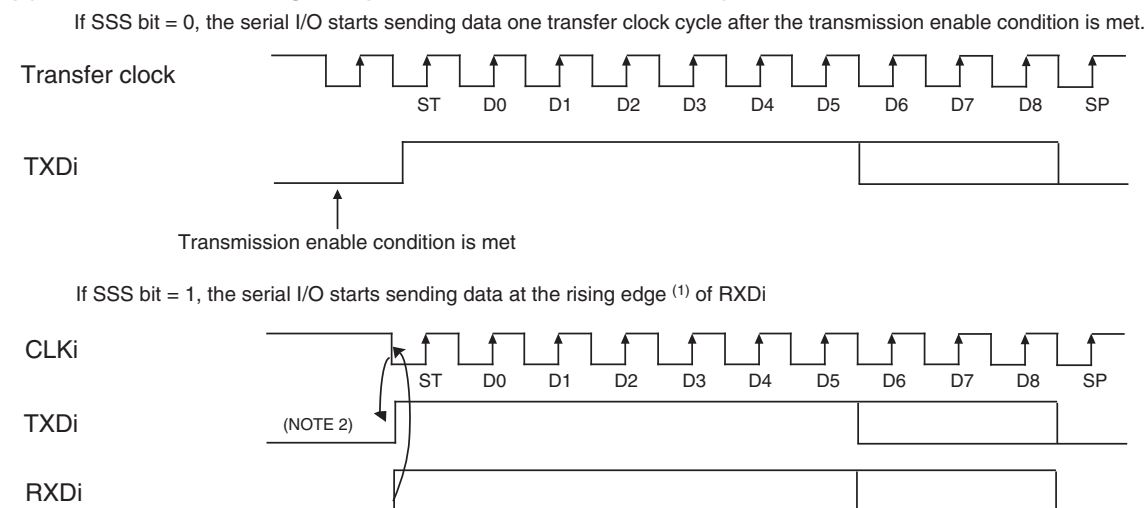
**Table 15.16 Registers to Be Used and Settings in IE Mode**

Register	Bit	Function
UiTB	0 to 8	Set transmission data
UiRB <sup>(1)</sup>	0 to 8	Reception data can be read
	OER, FER, PER, SUM	Error flag
UiBRG	0 to 7	Set a transfer rate
UiMR	SMD2 to SMD0	Set to "110b"
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Invalid because the PRYE bit = 0
	PRYE	Set to "0"
	IOPOL	Select the TXD/RXD input/output polarity
UiC0	CLK1, CLK0	Select the count source for the UiBRG register
	CRS	Invalid because the CRD bit = 1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Select TXDi pin output mode
	CKPOL	Set to "0"
	UFORM	Set to "0"
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS <sup>(2)</sup>	Select the source of UART2 transmit interrupt
	U2RRM <sup>(2)</sup> , UiLCH, UiERE	Set to "0"
UiSMR	0 to 3, 7	Set to "0"
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to "1" to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
IFSR0	IFSR06, IFSR07	Set to "1"
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because the CLKMD1 bit = 0
	CLKMD1, RCSP, 7	Set to "0"

i = 0 to 2

#### NOTES:

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

**(1) ABSCS Bit in UiSMR Register (bus collision detect sampling clock select)****(2) ACSE Bit in UiSMR Register (auto clear of transmit enable bit)****(3) SSS Bit in UiSMR Register (transmit start condition select)****NOTES:**

1. The falling edge of RXDi when IOPOL bit = 0; the rising edge of RXDi when IOPOL bit = 1.
2. The transmit condition must be met before the falling edge <sup>(1)</sup> of RXDi.

i = 0 to 2

This diagram applies to the case where IOPOL bit = 1 (reversed)

**Figure 15.31 Bus Collision Detect Function-Related Bits**



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

Based on UART mode, this is an SIM interface compatible mode. Direct and inverse formats can be implemented, and this mode allows to output a low from the TXD2 pin when a parity error is detected.

Table 15.17 lists the specifications of SIM mode. Table 15.18 lists the registers used in the SIM mode and the register values set. Figure 15.32 shows the typical transmit/receive timing in SIM mode.

**Table 15.17 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 = 0 (internal clock) : <math>f_i / 16(n+1)</math>  <math>f_i = f1SIO, f2SIO, f8SIO, f32SIO</math>. n: Setting value of the U2BRG register 00h to FFh</li> <li>• The CKDIR bit = 1 (external clock) : <math>fEXT/16(n+1)</math>  <math>fEXT</math>: Input from CLK2 pin. n: Setting value of the U2BRG register 00h to FFh</li> </ul>
Transmission start condition	Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> <li>• The TE bit in the U2C1 register = 1 (transmission enabled)</li> <li>• The TI bit in the U2C1 register = 0 (data present in the U2TB register)</li> </ul>
Reception start condition	Before reception can start, the following requirements must be met <ul style="list-style-type: none"> <li>• The RE bit in the U2C1 register = 1 (reception enabled)</li> <li>• Start bit detection</li> </ul>
Interrupt request generation timing <sup>(2)</sup>	<ul style="list-style-type: none"> <li>• For transmission                When the serial I/O finished sending data from the U2TB transfer register (U2IRS bit = 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 <sup>(3)</sup>                This error occurs when the number of stop bits set is not detected</li> <li>• Parity error <sup>(3)</sup>                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, the value of the U2RB register will be indeterminate. The IR bit in the S2RIC register does not change.
2. A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to "1" (transmit is completed) and U2ERE bit to "1" (error signal output) after reset. Therefore, when using SIM mode, set the IR bit to "0" (interrupt not requested) after setting these bits.
3. The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UARTi receive register to the UiRB register.

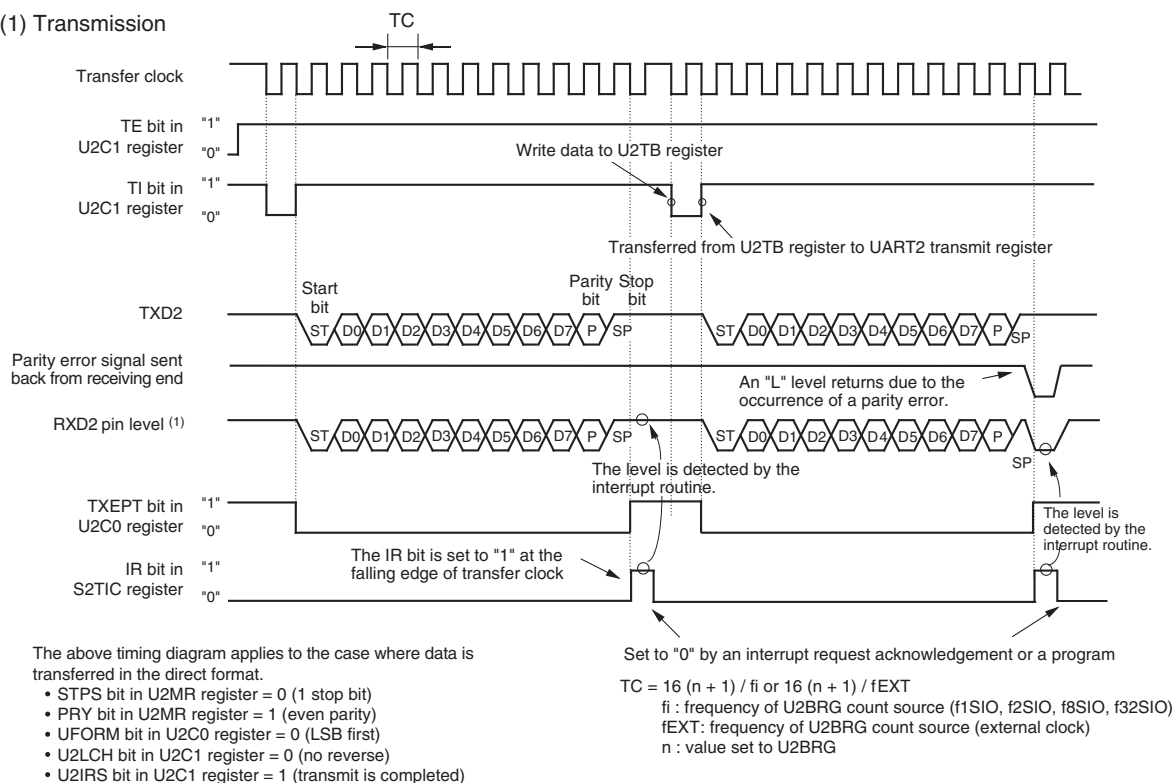
**Table 15.18 Registers to Be Used and Settings in SIM Mode**

Register	Bit	Function
U2TB <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 "101b"
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Set this bit to "1" for direct format or "0" for inverse format
	PRYE	Set to "1"
	IOPOL	Set to "0"
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because the CRD bit = 1
	TXEPT	Transmit register empty flag
	CRD	Set to "1"
	NCH	Set to "0"
	CKPOL	Set to "0"
	UFORM	Set this bit to "0" for direct format or "1" for inverse format
U2C1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS	Set to "1"
	U2RRM	Set to "0"
	U2LCH	Set this bit to "0" for direct format or "1" for inverse format
	U2ERE	Set to "1"
U2SMR <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 register bits are described above. Set those bits to "0" when writing to the registers in SIM mode.

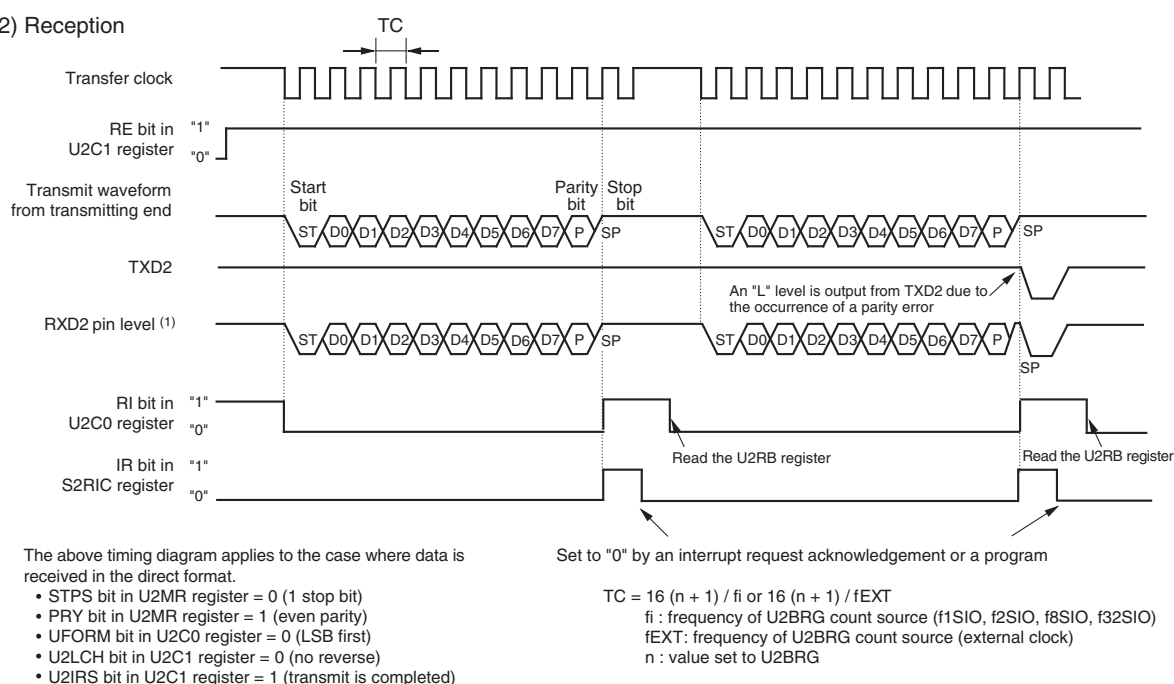
## (1) Transmission



## NOTE:

1. Because TXD2 and RXD2 are connected, a composite waveform, consisting of the TXD2 output and the parity error signal sent back from receiving end, is generated.

## (2) Reception



## NOTE:

1. Because TXD2 and RXD2 are connected, a composite waveform, consisting of transmit waveform from the transmitting end and parity error signal from receiving end, is generated.

Figure 15.32 Transmit and Receive Timing in SIM Mode

Figure 15.33 shows the example of connecting the SIM interface. Connect TXD2 and RXD2 and apply pull-up.

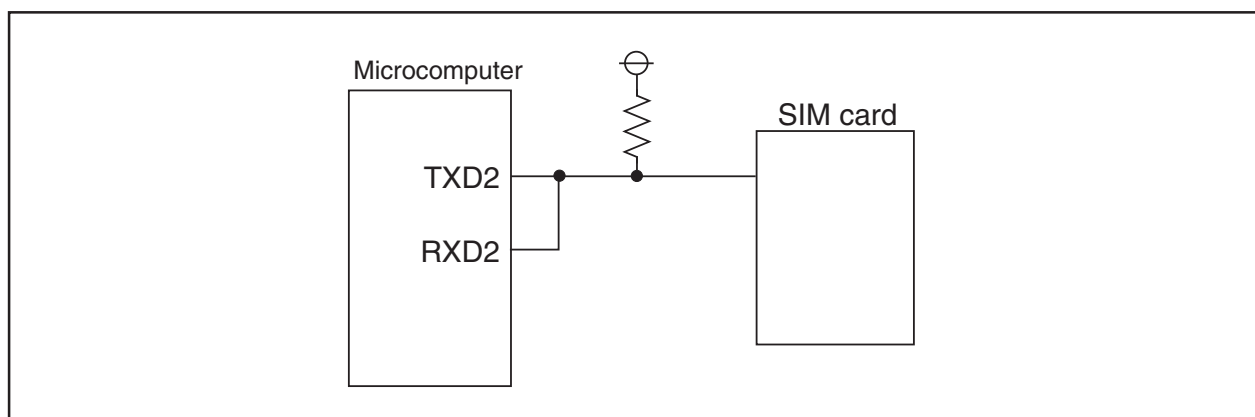


Figure 15.33 SIM Interface Connection

#### 15.1.6.1 Parity Error Signal Output

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

The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TXD2 output low with the timing shown in Figure 15.32. If the R2RB register is read while outputting a parity error signal, the PER bit is set 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 15.34 shows the output timing of the parity error signal

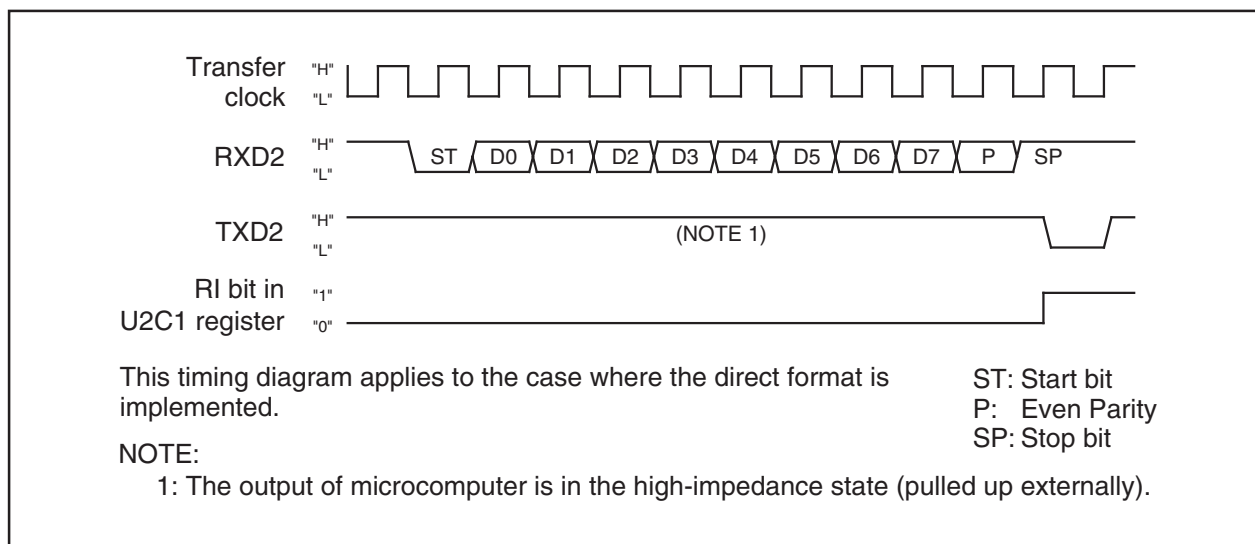


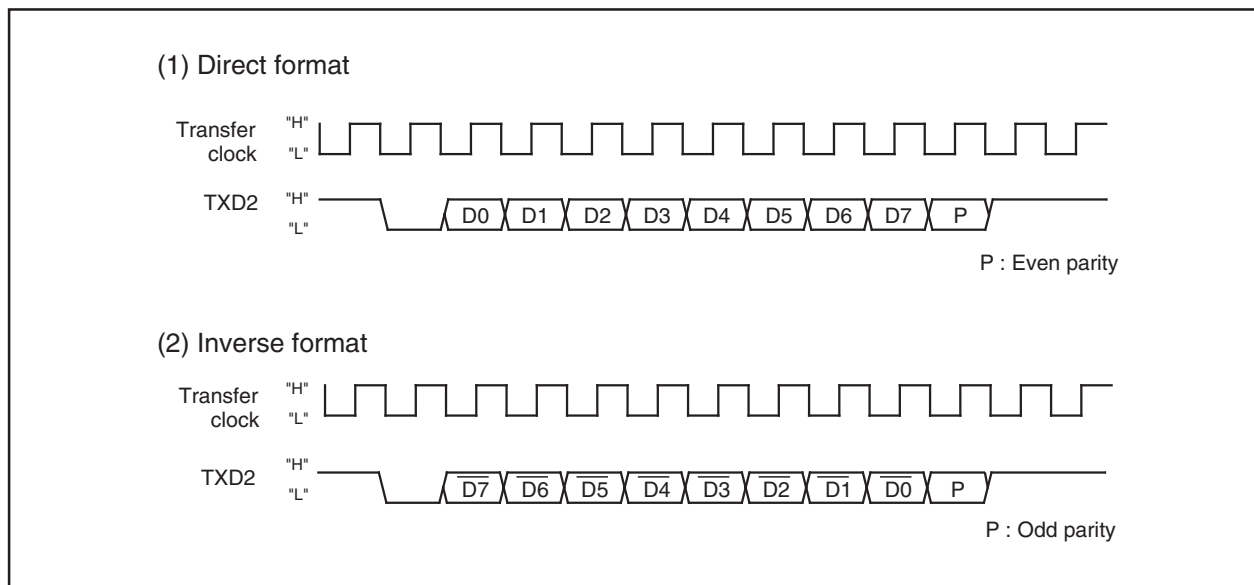
Figure 15.34 Parity Error Signal Output Timing

**15.1.6.2 Format**

When direct format, set the 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”.

When inverse format, set the PRY bit to “0”, UFORM bit to “1” and U2LCH bit to “1”.

Figure 15.35 shows the SIM interface format.



**Figure 15.35 SIM Interface Format**

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

Figure 10: S3 clock and data bus connections. The diagram illustrates the internal clock and data paths of the S3 module. It includes a clock source select circuit that can choose between f1SIO, f2SIO, f8SIO, or f32SIO, with dividers (1/2, 1/8, 1/4) and a PLL clock input. The selected clock is connected to the S33 and S36 registers. The S33 register is connected to the S36 register and the S3TRR register. The S36 register is connected to the S3TRR register and the S3BRG register. The S3TRR register is connected to the S3BRG register and the S3O3 interrupt request. The S3BRG register is connected to the S3O3 interrupt request. The S3O3 interrupt request is connected to the Data bus. The S3TRR register is connected to the Data bus via an 8-bit bus.

### Figure 15.36 SI/O3 Block Diagram

SI/O3 Control Register <sup>(1)</sup>

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
								S3C	01E2h	01000000b

## NOTES:

1. Make sure this register is written to by the next instruction after setting the PRC2 bit in the PRCR register to "1" (write enabled).
2. When the SM32 bit = 1, the corresponding pin is placed in the high-impedance state regardless of which functions of those pins are being used.
3. Set the SM33 bit to "1" (SOUT3 output, CLK3 function) and the corresponding port direction bit to "0" (input mode).
4. Set the SM33 bit to "1" (SOUT3 output, CLK3 function).
5. When changing the SM31 to SM30 bits, set the S3BRG register.

SI/O3 Bit Rate Generator <sup>(1) (2) (3)</sup>

b7	b0	Symbol	Address	After Reset
		S3BRG	01E3h	Indeterminate
		Description	Setting Range	RW
		Assuming that set value = n, S3BRG divides the count source by n + 1	00h to FFh	WO

## NOTES:

1. Write to this register while serial I/O is neither transmitting nor receiving.
2. Use the MOV instruction to write to this register.
3. Write to this register after setting the SM31 to SM30 bits in the S3C register.

SI/O3 Transmit/Receive Register <sup>(1) (2)</sup>

b7	b0	Symbol	Address	After Reset
<div></div>		S3TRR	01E0h	Indeterminate
<div></div>		Description		RW
		Transmission/reception starts by writing transmit data to this register. After transmission/reception finishes, reception data can be read by reading this register.		RW

## NOTES:

1. Write to this register while serial I/O is neither transmitting nor receiving.
2. To receive data, set the corresponding port direction bit for SIN3 to "0" (input mode).

Figure 15.37 S3C Register, S3BRG Register, and S3TRR Register

**Table 15.19 SI/O3 Specifications**

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> <li>• SM36 bit in S3C register = 1 (internal clock) : <math>f_j / 2(n+1)</math>  <math>f_j = f1SIO, f8SIO, f32SIO</math>. n = Setting value of S3BRG register 00h to FFh</li> <li>• SM36 bit = 0 (external clock) : Input from CLK3 pin <sup>(1)</sup></li> </ul>
Transmission/Reception Start Condition	Before transmission/reception can start, the following requirements must be met Write transmit data to the S3TRR register <sup>(2) (3)</sup>
Interrupt Request Generation Timing	<ul style="list-style-type: none"> <li>• When SM34 bit in S3C register = 0 The rising edge of the last transfer clock pulse <sup>(4)</sup></li> <li>• When SM34 bit = 1 The falling edge of the last transfer clock pulse <sup>(4)</sup></li> </ul>
CLK3 Pin Function	I/O port, transfer clock input, transfer clock output
SOUT3 Pin Function	I/O port, transmit data output, high-impedance
SIN3 Pin Function	I/O port, receive data input
Select Function	<ul style="list-style-type: none"> <li>• LSB first or MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> <li>• Function for setting an SOUT3 initial value set function When the SM36 bit in the S3C register = 0 (external clock), the SOUT3 pin output level while not transmitting can be selected.</li> <li>• CLK polarity selection Whether transmit data is output/input timing at the rising edge or falling edge of transfer clock can be selected.</li> </ul>

**NOTES:**

1. To set the SM36 bit in the S3C register to "0" (external clock), follow the procedure described below.
  - If the SM34 bit in the S3C register = 0, write transmit data to the S3TRR register while input on the CLK3 pin is high. The same applies when rewriting the SM37 bit in the S3C register.
  - If the SM34 bit = 1, write transmit data to the S3TRR register while input on the CLK3 pin is low. The same applies when rewriting the SM37 bit.
  - Because shift operation continues as long as the transfer clock is supplied to the SI/O3 circuit, stop the transfer clock after supplying eight pulses. If the SM36 bit = 1 (internal clock), the transfer clock automatically stops.
2. Unlike UART0 to UART2, SI/O3 is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the S3TRR register during transmission.
3. When the SM36 bit = 1 (internal clock), SOUT3 retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to a high-impedance state. However, if transmit data is written to the S3TRR register during this period, SOUT3 immediately goes to a high-impedance state, with the data hold time thereby reduced.
4. When the SM36 bit = 1 (internal clock), the transfer clock stops in the high state if the SM34 bit = 0, or stops in the low state if the SM34 bit = 1.



### 15.2.1 SI/O3 Operation Timing

Figure 15.38 shows the SI/O3 operation timing.

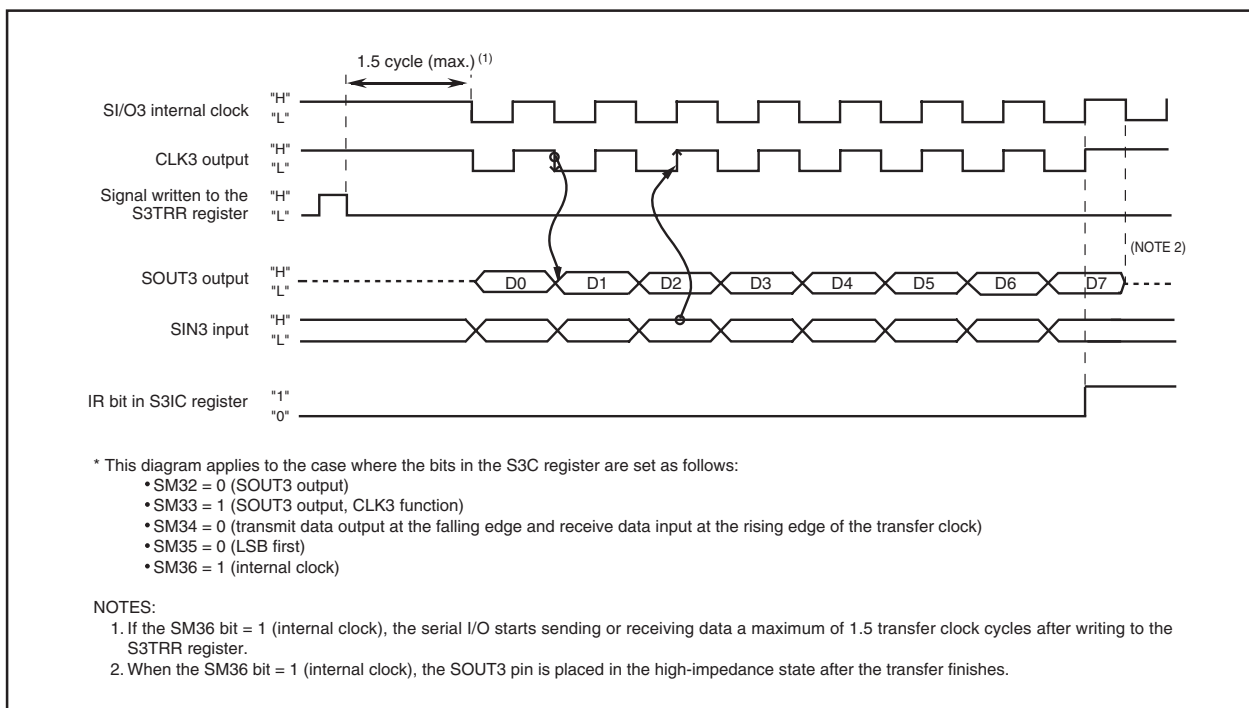


Figure 15.38 SI/O3 Operation Timing

### 15.2.2 CLK Polarity Selection

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

Figure 15.39 shows the polarity of the transfer clock.

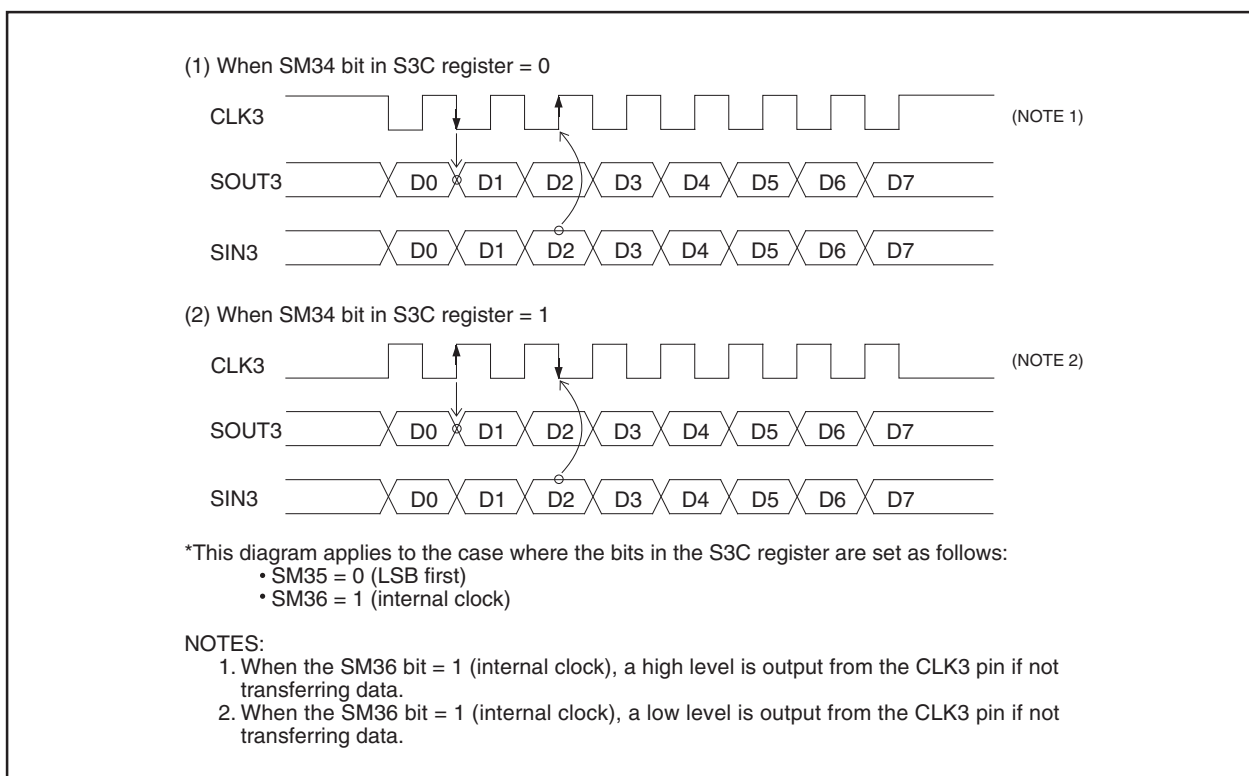


Figure 15.39 Polarity of Transfer Clock

### 15.2.3 Functions for Setting an SOUT3 Initial Value

If the SM36 bit in the S3C register = 0 (external clock), the SOUT3 pin output can be fixed high or low when not transferring. Figure 15.40 shows the timing chart for setting an SOUT3 initial value and how to set it.

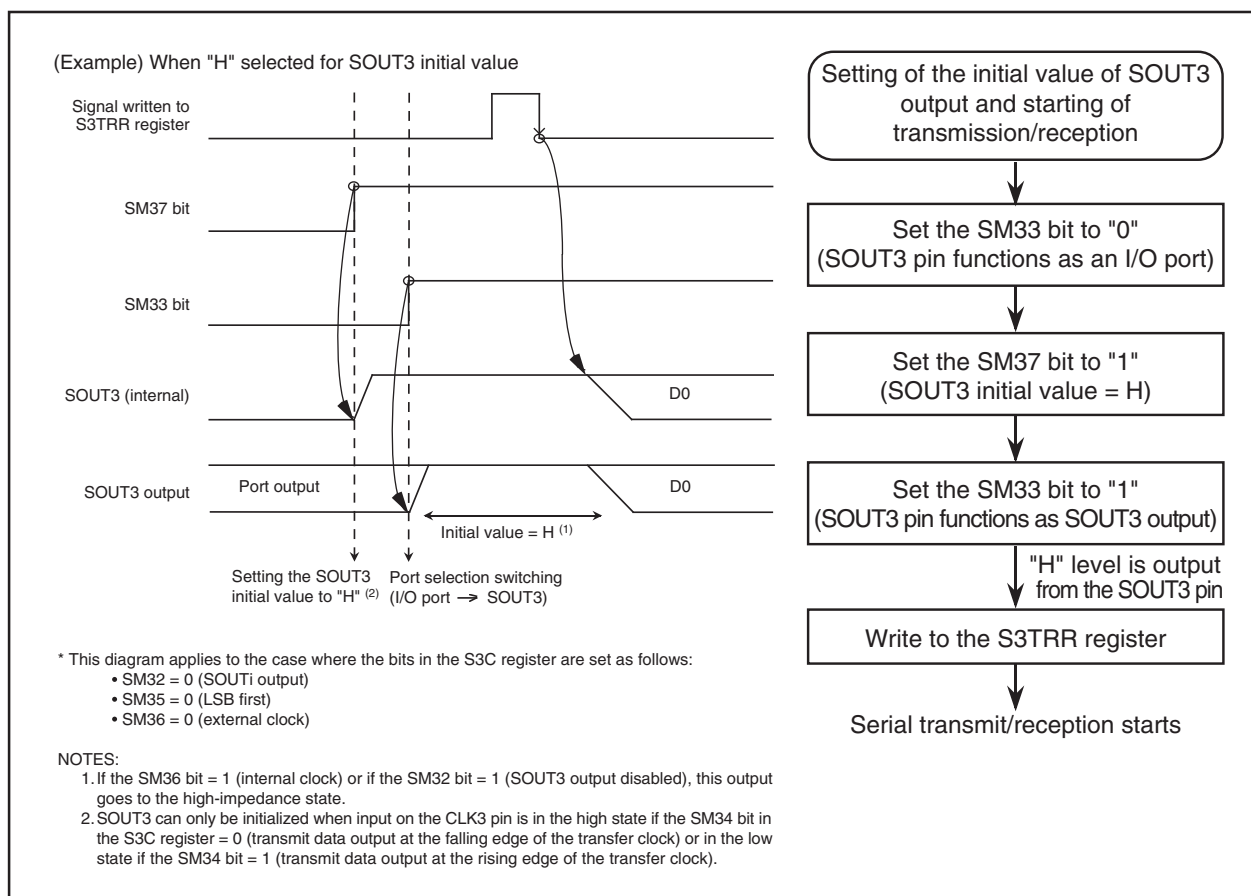


Figure 15.40 SOUT3's Initial Value Setting

## 16. A/D Converter

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 P10\_0 to P10\_7, P9\_5, P9\_6, P0\_0 to P0\_7, and P2\_0 to P2\_7. Similarly,  $\overline{\text{ADTRG}}$  input shares the pin with P9\_7. Therefore, when using these inputs, make sure the corresponding port direction bits are set to “0” (input mode).

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

The A/D conversion result is stored in the ADi register's bits for ANi, AN0\_i, and AN2\_i pins (i = 0 to 7).

Table 16.1 shows the performance of the A/D converter. Figure 16.1 shows the block diagram of the A/D converter, and Figures 16.2 and 16.3 show the A/D converter-related registers.

**Table 16.1 A/D Converter Performance**

Item	Performance
Method of A/D Conversion	Successive approximation (capacitive coupling amplifier)
Analog Input Voltage <sup>(1)</sup>	0V to AVCC (VCC)
Operating Clock $\phi_{\text{AD}}$ <sup>(2)</sup>	$f_{\text{AD}}$ , divide-by-2 of $f_{\text{AD}}$ , divide-by-3 of $f_{\text{AD}}$ , divide-by-4 of $f_{\text{AD}}$ , divide-by-6 of $f_{\text{AD}}$ , divide-by-12 of $f_{\text{AD}}$
Resolution	8 bits or 10 bits (selectable)
Integral Nonlinearity Error	When AVCC = VREF = 5 V <ul style="list-style-type: none"> <li>• With 8-bit resolution: <math>\pm 2\text{LSB}</math></li> <li>• With 10-bit resolution: <math>\pm 3\text{LSB}</math></li> </ul> When external operation amp connection mode is selected: $\pm 7\text{LSB}$ When AVCC = VREF = 3.3 V <ul style="list-style-type: none"> <li>• With 8-bit resolution: <math>\pm 2\text{LSB}</math></li> <li>• With 10-bit resolution: <math>\pm 5\text{LSB}</math></li> </ul> When external operation amp connection mode is selected: $\pm 7\text{LSB}$
Operating Modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog Input Pins	8 pins (AN0 to AN7) + 2 pins (ANEX0 and ANEX1) + 8 pins (AN0_0 to AN0_7) + 8 pins (AN2_0 to AN2_7)
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>• Software trigger The ADST bit in the ADCON0 register is set to “1” (A/D conversion starts)</li> <li>• External trigger (retriggerable) Input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to “1” (A/D conversion starts)</li> </ul>
Conversion Speed Per Pin	<ul style="list-style-type: none"> <li>• Without sample and hold 8-bit resolution: 49 <math>\phi_{\text{AD}}</math> cycles, 10-bit resolution: 59 <math>\phi_{\text{AD}}</math> cycles</li> <li>• With sample and hold 8-bit resolution: 28 <math>\phi_{\text{AD}}</math> cycles, 10-bit resolution: 33 <math>\phi_{\text{AD}}</math> cycles</li> </ul>

**NOTES:**

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

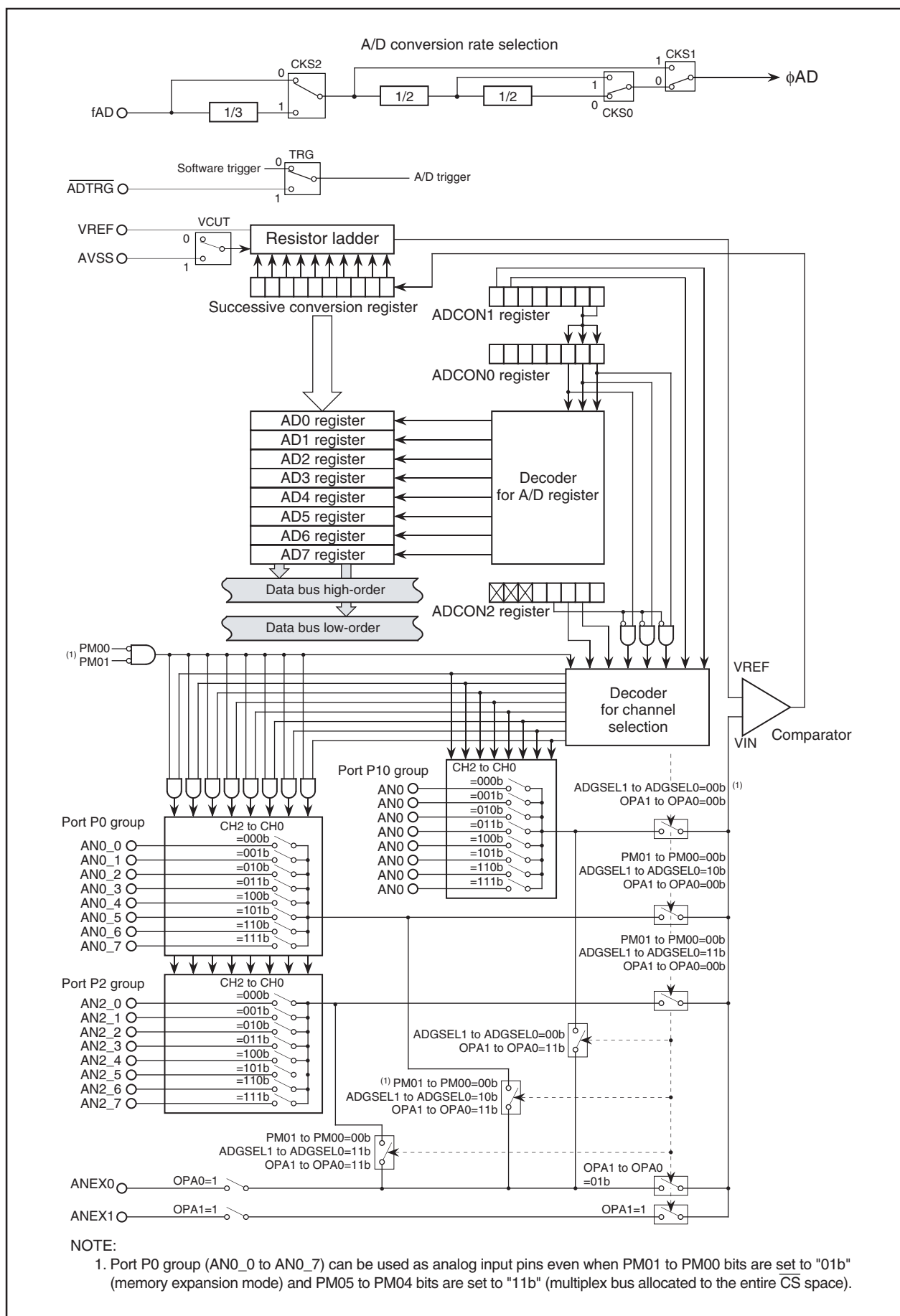


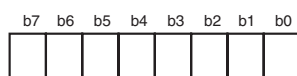
Figure 16.1 A/D Converter Block Diagram

A/D Control Register 0 <sup>(1)</sup>Symbol  
ADCON0Address  
03D6hAfter Reset  
00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit	Function varies with each operation mode	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW

## NOTE:

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

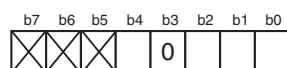
A/D Control Register 1 <sup>(1)</sup>Symbol  
ADCON1Address  
03D7hAfter Reset  
00h

Bit symbol	Bit name	Function	RW
SCAN0	A/D Sweep Pin Select Bit	Function varies with each operation mode	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW
VCUT	VREF Connect Bit <sup>(2)</sup>	0 : VREF not connected 1 : VREF connected	RW
OPA0	External Op-Amp Connection Mode Bit	Function varies with each operation mode	RW
OPA1			RW

## NOTES:

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

Figure 16.2 ADCON0 Register and ADCON1 Register

A/D Control Register 2 <sup>(1)</sup>Symbol  
ADCON2Address  
03D4hAfter Reset  
00h

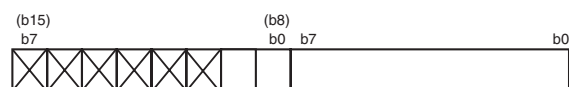
Bit Symbol	Bit Name	Function	RW
SMP	A/D Conversion Method Select Bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A/D Input Group Select Bit	b2 b1 0 0 : Port P10 group is selected 0 1 : Do not set a value 1 0 : Port P0 group is selected 1 1 : Port P2 group is selected	RW
ADGSEL1			RW
— (b3)	Reserved Bit	Set to "0"	RW
CKS2	Frequency Select Bit 2 <sup>(2)</sup>	0 : Selects fAD, divide-by-2 of fAD, or divide-by-4 of fAD. 1 : Selects divide-by-3 of fAD, divide-by-6 of fAD, or divide-by-12 of fAD.	RW
— (b7-b5)	Nothing is assigned. When write, set to "0". When read, their contents are "0".		—

## NOTES:

1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. The  $\phi$ AD frequency must be 10 MHz or less. The selected  $\phi$ AD frequency is determined by a combination of the CKS0 bit in the ADCON0 register, the CKS1 bit in the ADCON1 register, and the CKS2 bit in the ADCON2 register.

CKS2	CKS1	CKS0	$\phi$ AD
0	0	0	Divide-by-4 of fAD
0	0	1	Divide-by-2 of fAD
0	1	0	fAD
0	1	1	
1	0	0	Divide-by-12 of fAD
1	0	1	Divide-by-6 of fAD
1	1	0	Divide-by-3 of fAD
1	1	1	

## A/D Register i (i = 0 to 7)



Symbol	Address	After Reset
AD0	03C1h to 03C0h	Indeterminate
AD1	03C3h to 03C2h	Indeterminate
AD2	03C5h to 03C4h	Indeterminate
AD3	03C7h to 03C6h	Indeterminate
AD4	03C9h to 03C8h	Indeterminate
AD5	03CBh to 03CAh	Indeterminate
AD6	03CDh to 03CCh	Indeterminate
AD7	03CFh to 03CEh	Indeterminate

Function		RW
When BITS bit in ADCON1 register is "1" (10-bit mode)	When BITS bit is "0" (8-bit mode)	RW
Low-order 8 bits of A/D conversion result	A/D conversion result	RO
High-order 2 bits of A/D conversion result	When read, the content is indeterminate.	RO
Nothing is assigned. When write, set to "0". When read, their contents are "0".		—

Figure 16.3 ADCON2 Register, and AD0 to AD7 Registers

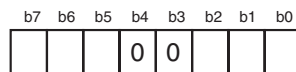
## 16.1 Mode Description

### 16.1.1 One-shot Mode

In one-shot mode, analog voltage applied to a selected pin is A/D converted once. Table 16.2 lists the specifications of one-shot mode. Figure 16.4 shows the ADCON0 and ADCON1 registers in one-shot mode.

**Table 16.2 One-shot Mode Specifications**

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

A/D Control Register 0 <sup>(1)</sup>

Symbol  
ADCON0

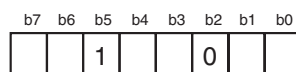
Address  
03D6h

After Reset  
00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected <sup>(2) (3)</sup>	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 0 0 : One-shot mode <sup>(3)</sup>	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW

## NOTES:

1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN0\_0 to AN\_7, and AN2\_0 to AN2\_7 can be used in same way as AN0 to AN7. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. After rewriting the MD1 to MD0 bits, set the CH2 to CH0 bits over again using another instruction.

A/D Control Register 1 <sup>(1)</sup>

Symbol  
ADCON1

Address  
03D7h

After Reset  
00h

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D Sweep Pin Select Bit	Invalid in one-shot mode	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	Set to "0" when one-shot mode is selected	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW
VCUT	VREF Connect Bit <sup>(2)</sup>	1 : VREF connected	RW
OPA0	External Op-Amp Connection Mode Bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : External op-amp connection mode	RW
OPA1			RW

## NOTES:

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

Figure 16.4 ADCON0 Register and ADCON1 Register in One-shot Mode



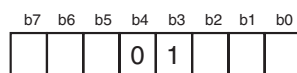
### 16.1.2 Repeat Mode

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

Table 16.3 lists the specifications of repeat mode. Figure 16.5 shows the ADCON0 and ADCON1 registers in repeat mode.

**Table 16.3 Repeat Mode Specifications**

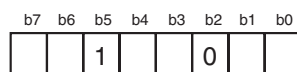
Item	Specification
Function	The CH2 to CH0 bits in the ADCON0 register, the ADGSEL1 to ADGSEL0 bits in the ADCON2 register and the OPA1 to OPA0 bits in the ADCON1 register select a pin. Analog voltage applied to this pin is repeatedly converted to a digital code.
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>When the TRG bit in the ADCON0 register is “0” (software trigger) The ADST bit in the ADCON0 register is set to “1” (A/D conversion starts)</li> <li>When the TRG bit is “1” (ADTRG trigger) Input on the ADTRG pin changes state from high to low after the ADST bit is set to “1” (A/D conversion starts)</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, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0 to ANEX1
Reading of Result of A/D Converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

A/D Control Register 0 <sup>(1)</sup>Symbol  
ADCON0Address  
03D6hAfter Reset  
00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected <sup>(2) (3)</sup>	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 0 1 : Repeat mode <sup>(3)</sup>	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW

## NOTES:

1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN0\_0 to AN\_7, and AN2\_0 to AN2\_7 can be used in same way as AN0 to AN7. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. After rewriting the MD1 to MD0 bits, set the CH2 to CH0 bits over again using another instruction.

A/D Control Register 1 <sup>(1)</sup>Symbol  
ADCON1Address  
03D7hAfter Reset  
00h

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D Sweep Pin Select Bit	Invalid in repeat mode	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	Set to "0" when repeat mode is selected	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW
VCUT	VREF Connect Bit <sup>(2)</sup>	1 : VREF connected	RW
OPA0	External Op-Amp Connection Mode Bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : External op-amp connection mode	RW
OPA1			RW

## NOTES:

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

Figure 16.5 ADCON0 Register and ADCON1 Register in Repeat Mode

### 16.1.3 Single Sweep Mode

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

**Table 16.4 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 this 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) The ADST bit in the ADCON0 register is set to "1" (A/D conversion starts)</li> <li>When the TRG bit is "1" (ADTRG trigger) Input on the ADTRG pin changes state from high to low after the ADST bit is set to "1" (A/D conversion starts)</li> </ul>
A/D Conversion Stop Condition	<ul style="list-style-type: none"> <li>Completion of A/D conversion (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	Completion of A/D conversion
Analog Input Pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), AN0 to AN7 (8 pins) <sup>(1)</sup>
Reading of Result of A/D Converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

NOTE:

1. AN0\_0 to AN0\_7, and AN2\_0 to AN2\_7 can be used in the same way as AN0 to AN7.

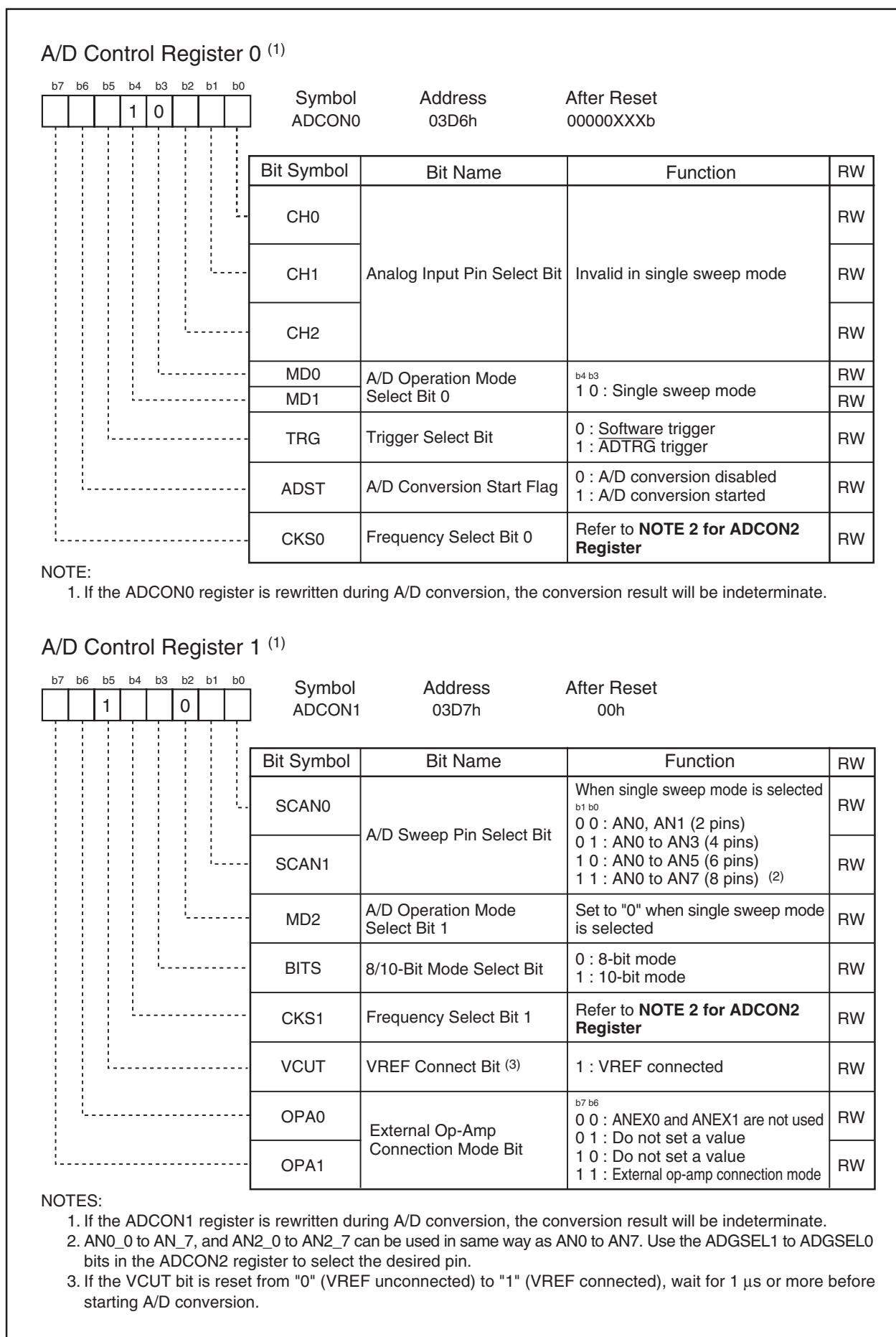


Figure 16.6 ADCON0 Register and ADCON1 Register in Single Sweep Mode

### 16.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltage applied to selected pins is repeatedly converted to a digital code.

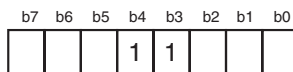
Table 16.5 lists the specifications of repeat sweep mode 0. Figure 16.7 shows the ADCON0 and ADCON1 registers in repeat sweep mode 0.

**Table 16.5 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 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) The ADST bit in the ADCON0 register is set to "1" (A/D conversion starts)</li> <li>When the TRG bit is "1" (ADTRG trigger) Input on the ADTRG pin changes state from high to low after the ADST bit is set to "1" (A/D conversion starts)</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>
Reading of Result of A/D Converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

**NOTE:**

1. AN0\_0 to AN0\_7, and AN2\_0 to AN2\_7 can be used in the same way as AN0 to AN7.

A/D Control Register 0 <sup>(1)</sup>Symbol  
ADCON0Address  
03D6hAfter Reset  
00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit	Invalid in repeat sweep mode 0	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW

## NOTE:

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

A/D Control Register 1 <sup>(1)</sup>Symbol  
ADCON1Address  
03D7hAfter reset  
00h

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D Sweep Pin Select Bit	When repeat sweep mode 0 is selected b1 b0 0 0 : AN0, AN1 (2 pins) 0 1 : AN0 to AN3 (4 pins) 1 0 : AN0 to AN5 (6 pins) 1 1 : AN0 to AN7 (8 pins) <sup>(2)</sup>	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	Set to "0" when repeat sweep mode 0 is selected	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW
VCUT	VREF Connect Bit <sup>(3)</sup>	1 : VREF connected	RW
OPA0	External Op-Amp Connection Mode Bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : Do not set a value 1 0 : Do not set a value 1 1 : External op-amp connection mode	RW
OPA1			RW

## NOTES:

1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN0\_0 to AN\_7, and AN2\_0 to AN2\_7 can be used in same way as AN0 to AN7. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from "0" (VREF unconnected) to "1" (VREF connected), wait for 1 μs or more before starting A/D conversion.

Figure 16.7 ADCON0 Register and ADCON1 Register in Repeat Sweep Mode 0

### 16.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage selectively applied to all pins is repeatedly converted to a digital code.

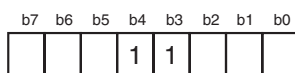
Table 16.6 lists the specifications of repeat sweep mode 1. Figure 16.8 shows the ADCON0 and ADCON1 registers in repeat sweep mode 1.

**Table 16.6 Repeat Sweep Mode 1 Specifications**

Item	Specification
Function	The input voltages on all pins selected by the ADGSEL1 to ADGSEL0 bits in the ADCON2 register are A/D converted repeatedly, with priority given to pins selected by the SCAN1 to SCAN0 bits in the ADCON1 register and ADGSEL1 to ADGSEL0 bits. Example : If AN0 selected, input voltages are A/D converted in order of AN0 → AN1 → AN0 → AN2 → AN0 → AN3, and so on.
A/D Conversion Start Condition	<ul style="list-style-type: none"> <li>When the TRG bit in the ADCON0 register is “0” (software trigger) The ADST bit in the ADCON0 register is set to “1” (A/D conversion starts)</li> <li>When the TRG bit is “1” (ADTRG trigger) Input on the ADTRG pin changes state from high to low after the ADST bit is set to “1” (A/D conversion starts)</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 to be Given Priority when A/D Converted	Select from AN0 (1 pin), AN0 to AN1 (2 pins), AN0 to AN2 (3 pins), AN0 to AN3 (4 pins) <sup>(1)</sup>
Reading of Result of A/D Converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

NOTE:

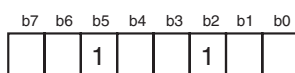
1. AN0\_0 to AN0\_7, and AN2\_0 to AN2\_7 can be used in the same way as AN0 to AN7.

A/D Control Register 0 <sup>(1)</sup>Symbol  
ADCON0Address  
03D6hAfter Reset  
00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit	Invalid in repeat sweep mode 1	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger Select Bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW

## NOTE:

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

A/D Control Register 1 <sup>(1)</sup>Symbol  
ADCON1Address  
03D7hAfter Reset  
00h

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D Sweep Pin Select Bit	When repeat sweep mode 1 is selected b1 b0 0 0 : AN0 (1 pin) 0 1 : AN0, AN1 (2 pins) 1 0 : AN0 to AN2 (3 pins) 1 1 : AN0 to AN3 (4 pins) <sup>(2)</sup>	RW
SCAN1			RW
MD2	A/D Operation Mode Select Bit 1	Set to "1" when repeat sweep mode 1 is selected	RW
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to <b>NOTE 2 for ADCON2 Register</b>	RW
VCUT	VREF Connect Bit <sup>(3)</sup>	1 : VREF connected	RW
OPA0	External Op-Amp Connection Mode Bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : Do not set a value 1 0 : Do not set a value 1 1 : External op-amp connection mode	RW
OPA1			RW

## NOTES:

1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN0\_0 to AN\_7, and AN2\_0 to AN2\_7 can be used in same way as AN0 to AN7. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from "0" (VREF unconnected) to "1" (VREF connected), wait for 1 μs or more before starting A/D conversion.

Figure 16.8 ADCON0 Register and ADCON1 Register in Repeat Sweep Mode 1



## 16.2 Function

### 16.2.1 Resolution Select Function

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

### 16.2.2 Sample and Hold

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

### 16.2.3 Extended Analog Input Pins

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

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

### 16.2.4 External Operation Amplifier (Op-Amp) Connection Mode

Multiple analog inputs can be amplified using a single external op-amp via the ANEX0 and ANEX1 pins. Set the OPA1 to OPA0 bits in the ADCON1 register to "11b" (external op-amp connection mode). The inputs from ANi ( $i = 0$  to 7)<sup>(1)</sup> are output from the ANEX0 pin. Amplify this output with an external op-amp before sending it back to the ANEX1 pin. The A/D conversion result is stored in the corresponding ADi register. The A/D conversion speed depends on the response characteristics of the external op-amp. Figure 16.9 shows an example of how to connect the pins in external operation amp.

#### NOTE:

1. AN0\_i and AN2\_i can be used the same as ANi.

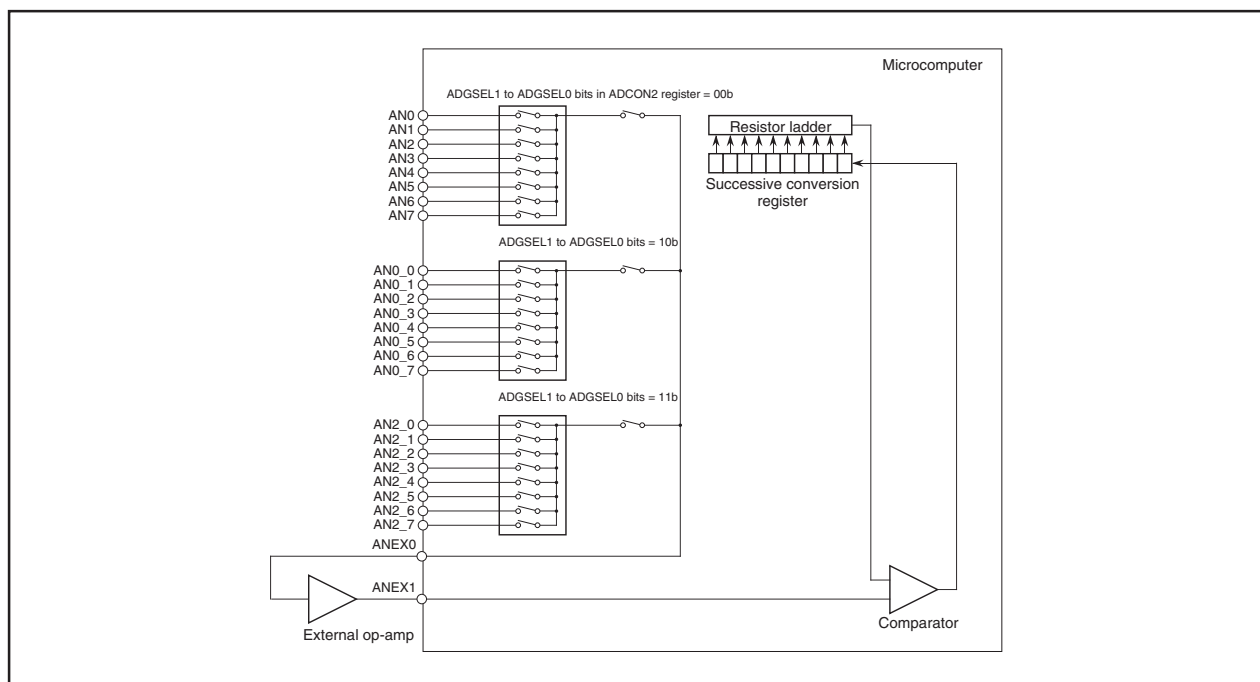


Figure 16.9 External Op-Amp Connection

### 16.2.5 Current Consumption Reducing Function

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

To use the A/D converter, set the VCUT bit to "1" (VREF connected) and then set the ADST bit in the ADCON0 register to "1" (A/D conversion start). The VCUT and ADST bits cannot be set to "1" at the same time. Nor can the VCUT bit be set to "0" (VREF unconnected) during A/D conversion.

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

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

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 16.10 has to be completed within a specified period of time. T (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be R0, microcomputer's internal resistance be R, precision (error) of the A/D converter be X, and the resolution of A/D converter be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

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

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

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

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

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

Figure 16.10 shows analog input pin and external sensor equivalent circuit.

When the difference between VIN and VC becomes 0.1LSB, we find impedance R0 when voltage between pins VC changes from 0 to VIN-(0.1/1024) VIN in time T. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB.

When f(φAD) = 10 MHz, T = 0.3 μs in the A/D conversion mode with sample & hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

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

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

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

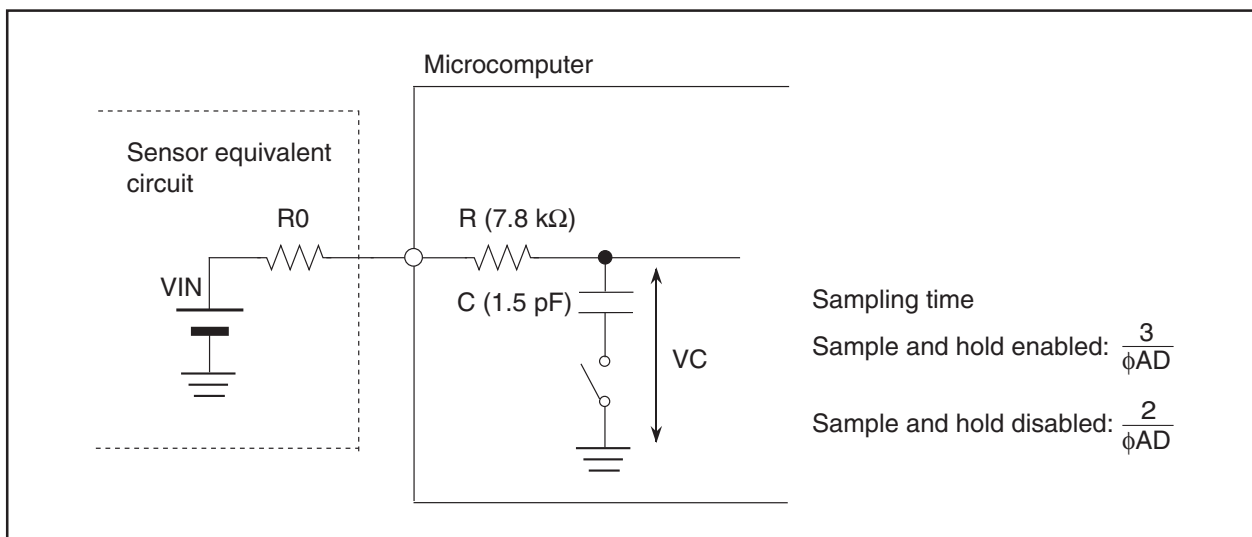


Figure 16.10 Analog Input Pin and External Sensor Equivalent Circuit

## 17. D/A Converter

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

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

Output analog voltage (V) is determined by a set value (n : decimal) in the DA<sub>i</sub> register.

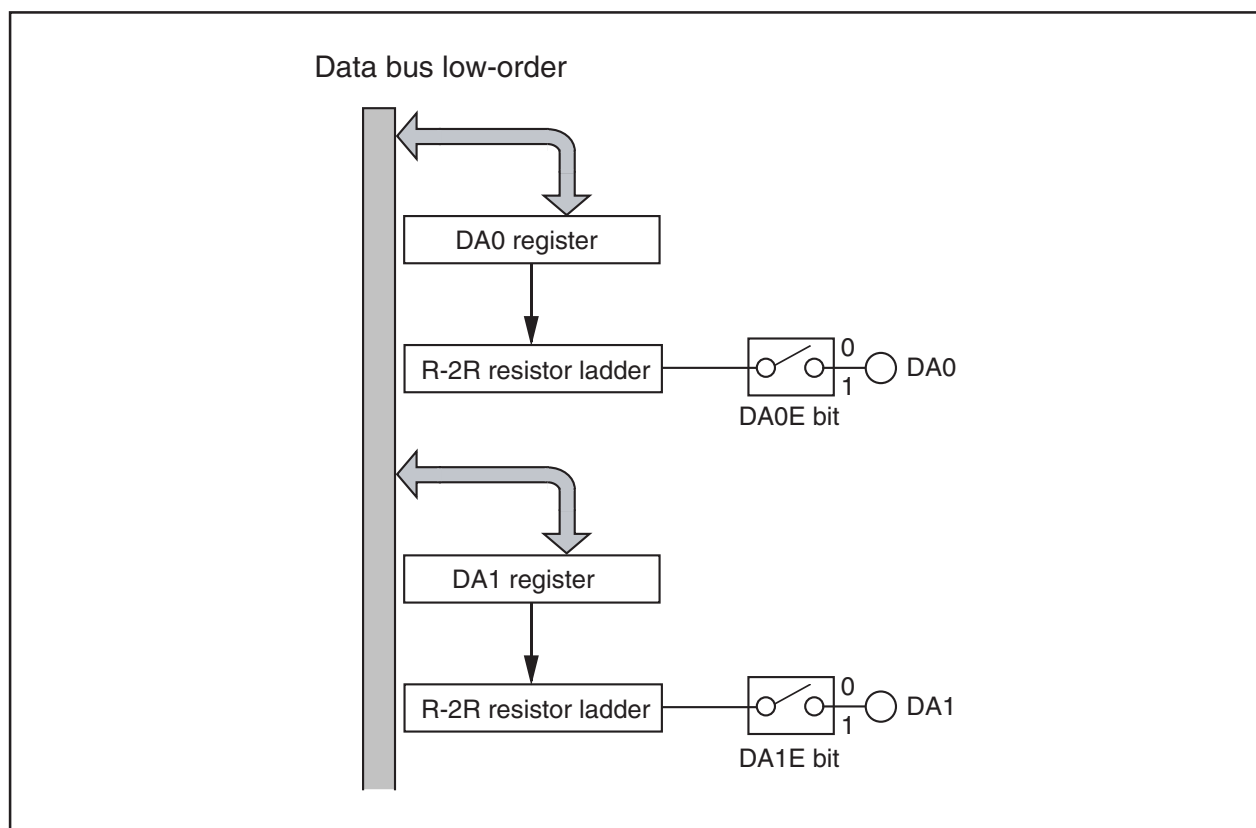
$$V = V_{REF} \times n / 256 \quad (n = 0 \text{ to } 255)$$

V<sub>REF</sub> : reference voltage

Table 17.1 lists the performance of the D/A converter. Figure 17.1 shows the block diagram of the D/A converter. Figure 17.2 shows the D/A converter-related registers. Figure 17.3 shows the D/A converter equivalent circuit.

**Table 17.1 D/A Converter Performance**

Item	Performance
D/A conversion Method	R-2R method
Resolution	8 bits
Analog Output Pin	2 channels (DA0 and DA1)



**Figure 17.1 D/A Converter Block Diagram**

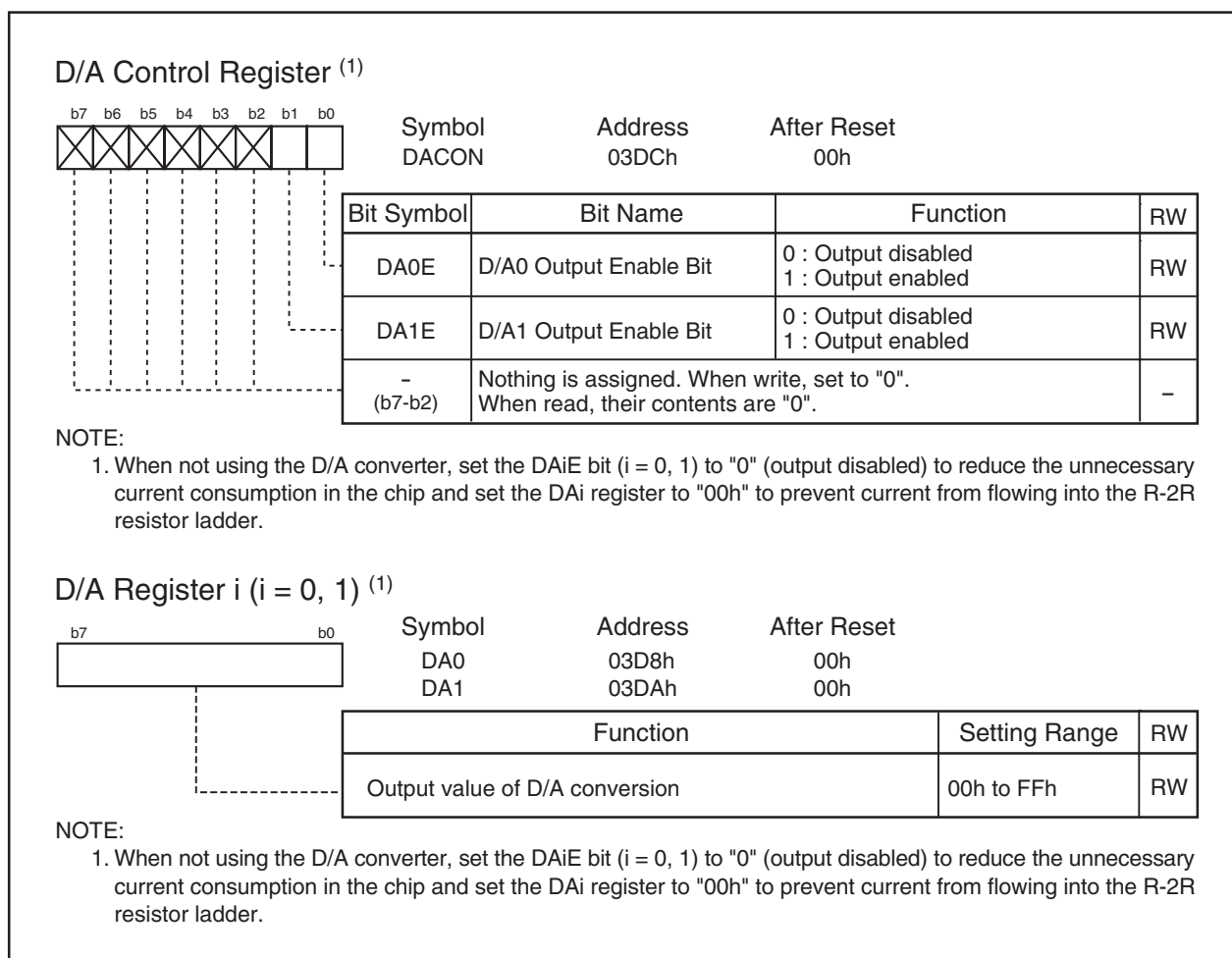


Figure 17.2 DACon Register, DA0 and DA1 Registers

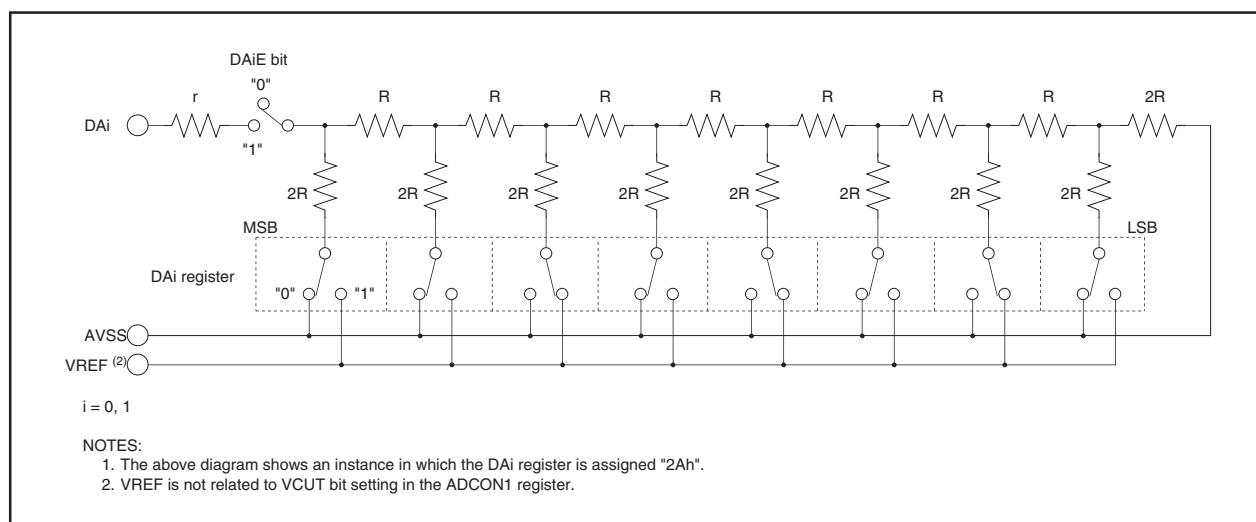


Figure 17.3 D/A Converter Equivalent Circuit

## 18. CRC Calculation

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

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

Figure 18.1 shows the block diagram of the CRC circuit. Figure 18.2 shows the CRC-related registers. Figure 18.3 shows the calculation example using the CRC operation.

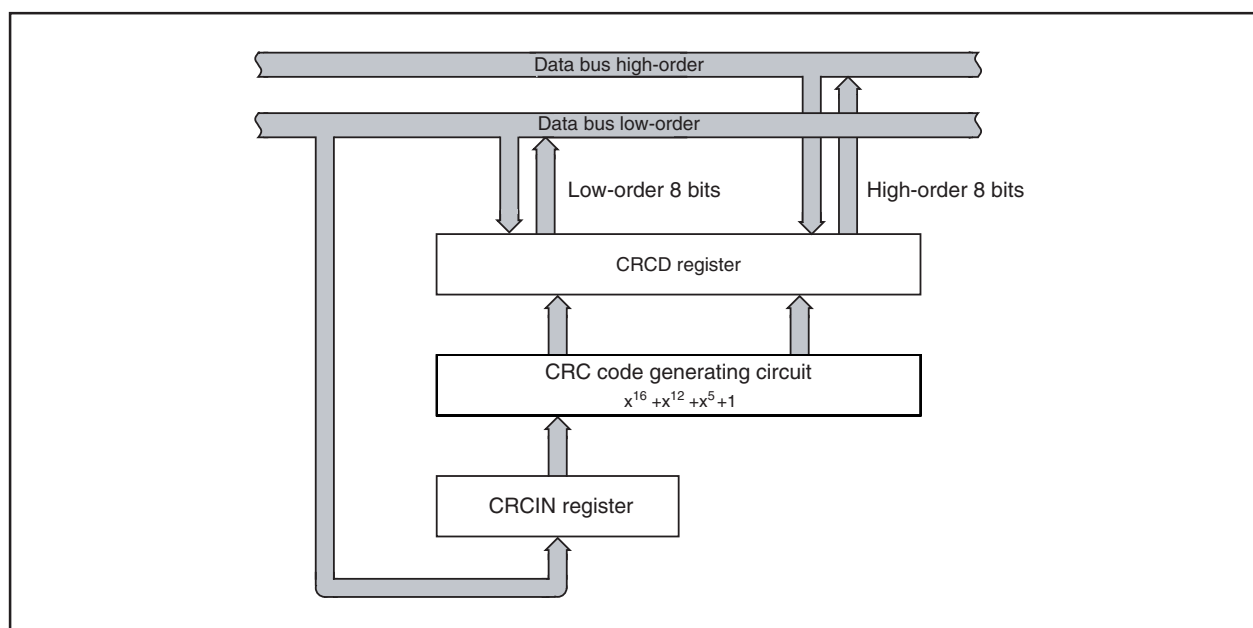


Figure 18.1 CRC Circuit Block Diagram

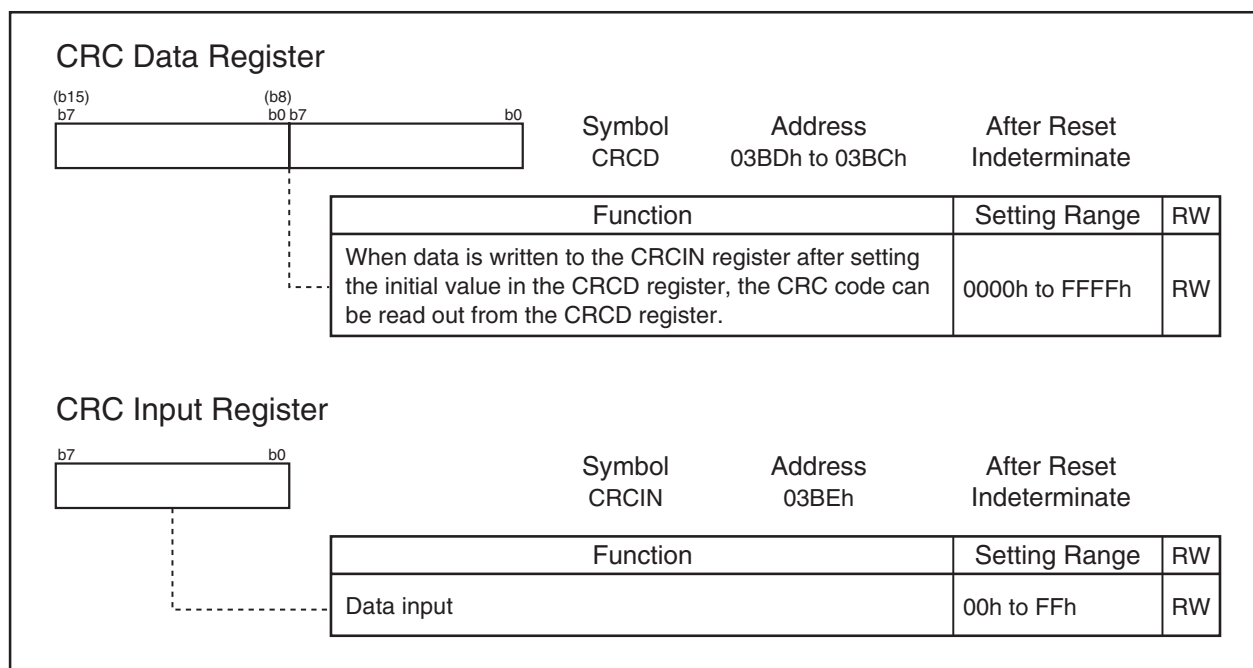


Figure 18.2 CRCD Register and CRCIN Register

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

- CRC operation performed by the M16C

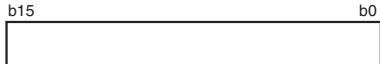
CRC code: Remainder of a division in which the value written to the CRCIN register with its bit positions reversed is divided by the generator polynomial

Generator polynomial:  $X^6 + X^{12} + X^5 + 1$  (1 0001 0000 0010 0001b)

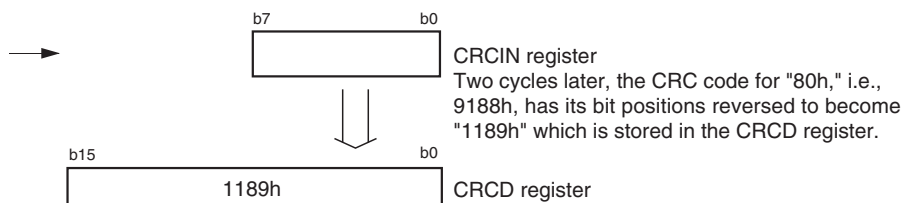
- Setting procedure

- Reverse the bit positions of the value "80C4h" by program in 1-byte unit.

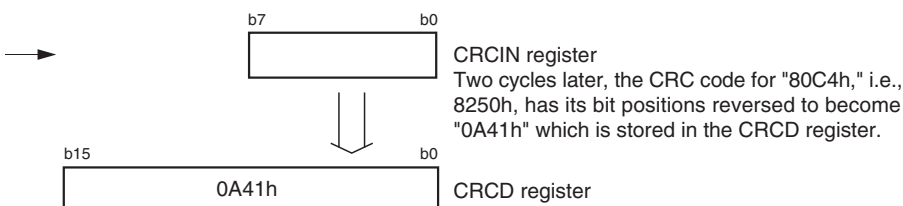
"80h" → "01h", "C4h" → "23h"

(2) Write 0000h (initial value) →  CRCD register

- Write 01h

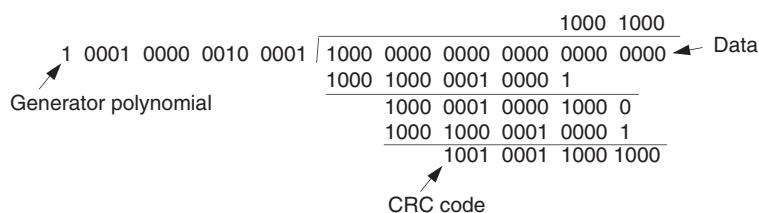


- Write 23h



- Details of CRC operation

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



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

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

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

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

Figure 18.3 CRC Calculation

## 19. CAN Module

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

Figure 19.1 shows a block diagram of the CAN module.

External CAN bus driver and receiver are required.

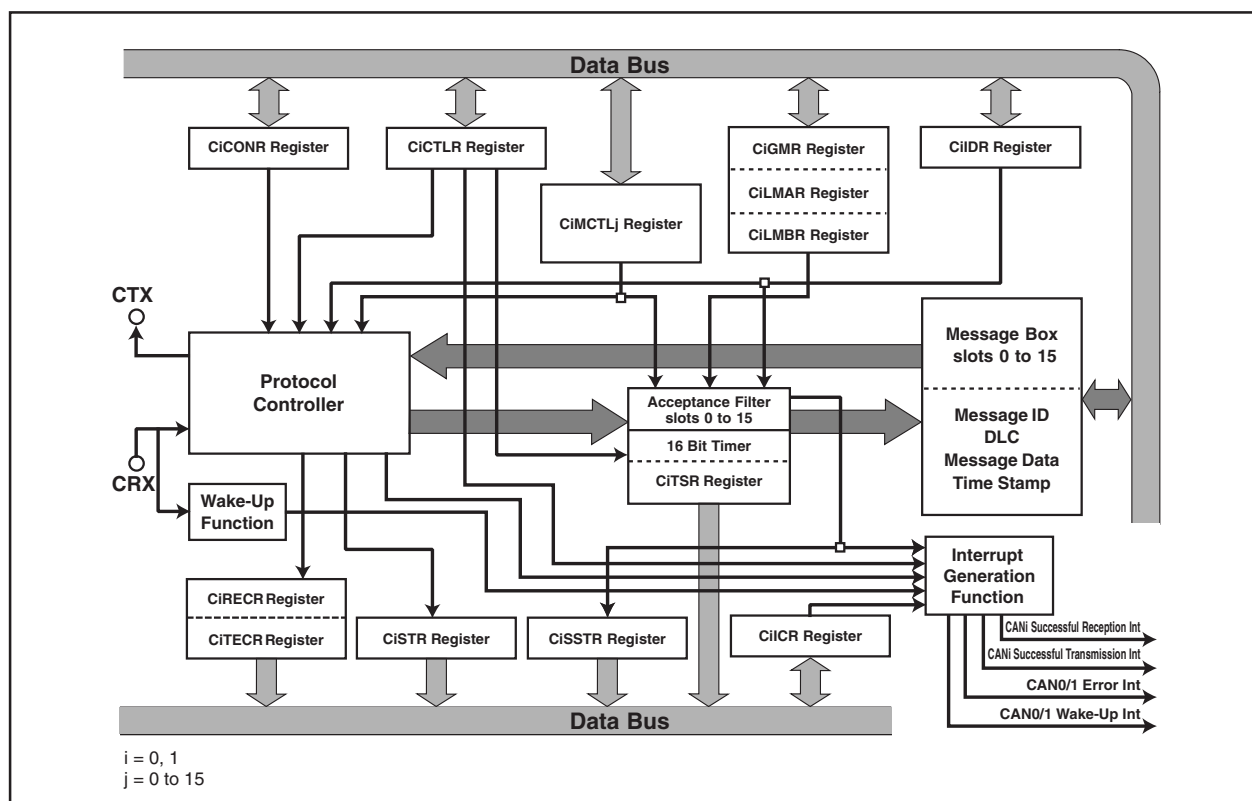


Figure 19.1 CAN Module Block Diagram

CTX/CRX:	CAN I/O pins.
Protocol controller:	This controller handles the bus arbitration and the CAN protocol services, i.e. bit timing, stuffing, error status etc.
Message box:	This memory block consists of 16 slots that can be configured either as transmitter or receiver. Each slot contains an individual ID, data length code, a data field (8 bytes) and a time stamp.
Acceptance filter:	This block performs filtering operation for received messages. For the filtering operation, the CiGMR register ( $i = 0, 1$ ), the CiLMAR register, or the CiLMBR register is used.
16 bit timer:	Used for the time stamp function. When the received message is stored in the message memory, the timer value is stored as a time stamp.
Wake-up function:	CAN0/1 wake-up interrupt request is generated by a message from the CAN bus.
Interrupt generation function:	The interrupt requests are generated by the CAN module. CANi successful reception interrupt, CANi successful transmission interrupt, CAN0/1 error interrupt and CAN0/1 wake-up interrupt.



## 19.1 CAN Module-Related Registers

The CAN<sub>i</sub> (i = 0, 1) module has the following registers.

### 19.1.1 CAN Message Box

A CAN module is equipped with 16 slots (16 bytes or 8 words each). Slots 14 and 15 can be used as Basic CAN.

- Priority of the slots: The smaller the number of the slot, the higher the priority, in both transmission and reception.
- A program can define whether a slot is defined as transmitter or receiver.

### 19.1.2 Acceptance Mask Registers

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

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

### 19.1.3 CAN SFR Registers

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

Explanation of each register is given below.

## 19.2 CANi Message Box (i = 0, 1)

Table 19.1 shows the memory mapping of the CANi message box.

It is possible to access to the message box in byte or word.

Mapping of the message contents differs from byte access to word access. Byte access or word access can be selected by the MsgOrder bit of the CiCTLR register.

**Table 19.1 Memory Mapping of CANi Message Box**

Address		Message Content (Memory mapping)	
CAN0	CAN1	Byte access (8 bits)	Word access (16 bits)
$0060h + n \cdot 16 + 0$	$0260h + n \cdot 16 + 0$	SID10 to SID6	SID5 to SID0
$0060h + n \cdot 16 + 1$	$0260h + n \cdot 16 + 1$	SID5 to SID0	SID10 to SID6
$0060h + n \cdot 16 + 2$	$0260h + n \cdot 16 + 2$	EID17 to EID14	EID13 to EID6
$0060h + n \cdot 16 + 3$	$0260h + n \cdot 16 + 3$	EID13 to EID6	EID17 to EID14
$0060h + n \cdot 16 + 4$	$0260h + n \cdot 16 + 4$	EID5 to EID0	Data Length Code (DLC)
$0060h + n \cdot 16 + 5$	$0260h + n \cdot 16 + 5$	Data Length Code (DLC)	EID5 to EID0
$0060h + n \cdot 16 + 6$	$0260h + n \cdot 16 + 6$	Data byte 0	Data byte 1
$0060h + n \cdot 16 + 7$	$0260h + n \cdot 16 + 7$	Data byte 1	Data byte 0
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$0060h + n \cdot 16 + 13$	$0260h + n \cdot 16 + 13$	Data byte 7	Data byte 6
$0060h + n \cdot 16 + 14$	$0260h + n \cdot 16 + 14$	Time stamp high-order byte	Time stamp low-order byte
$0060h + n \cdot 16 + 15$	$0260h + n \cdot 16 + 15$	Time stamp low-order byte	Time stamp high-order byte

i = 0, 1

n = 0 to 15: the number of the slot

Figures 19.2 and 19.3 show the bit mapping in each slot in byte access and word access. The content of each slot remains unchanged unless transmission or reception of a new message is performed.

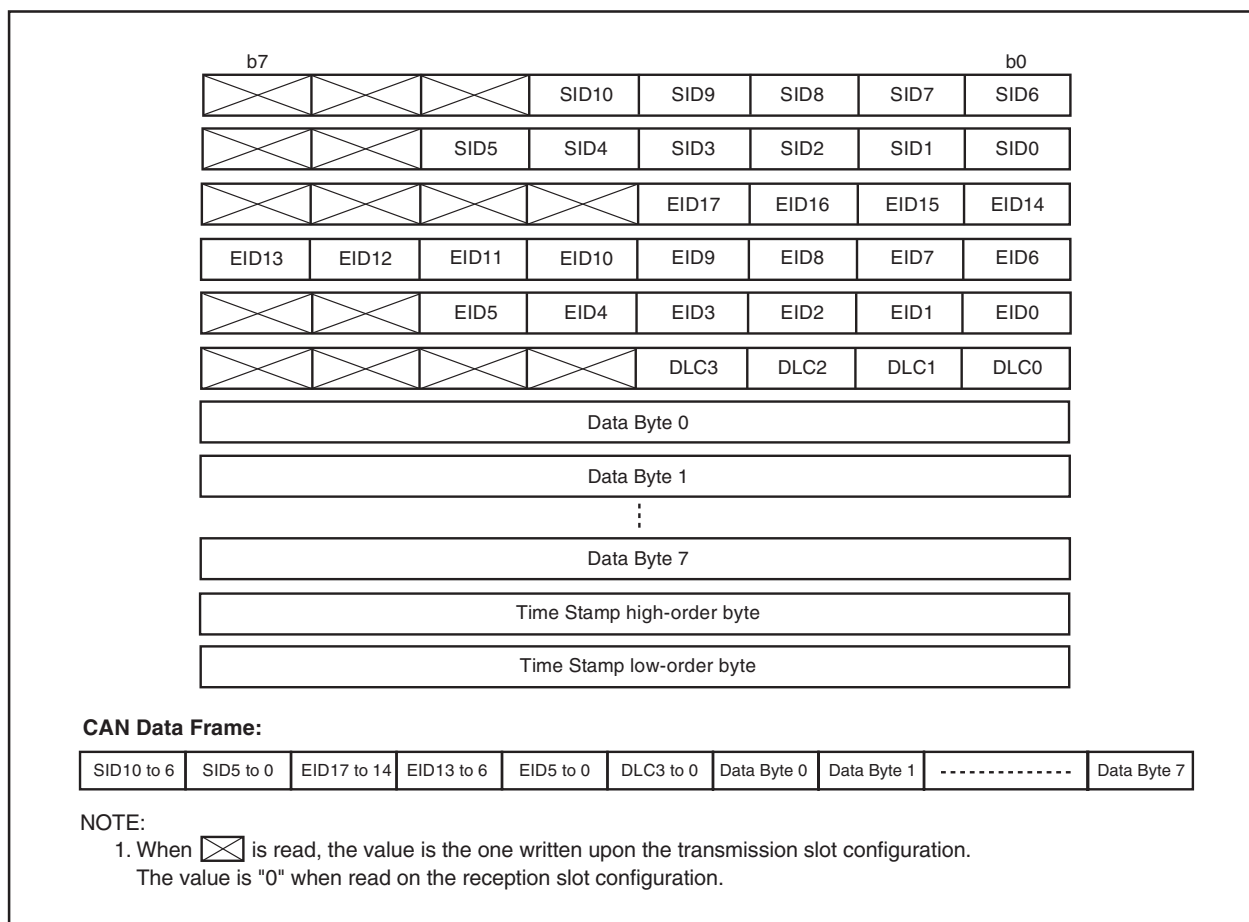


Figure 19.2 Bit Mapping in Byte Access

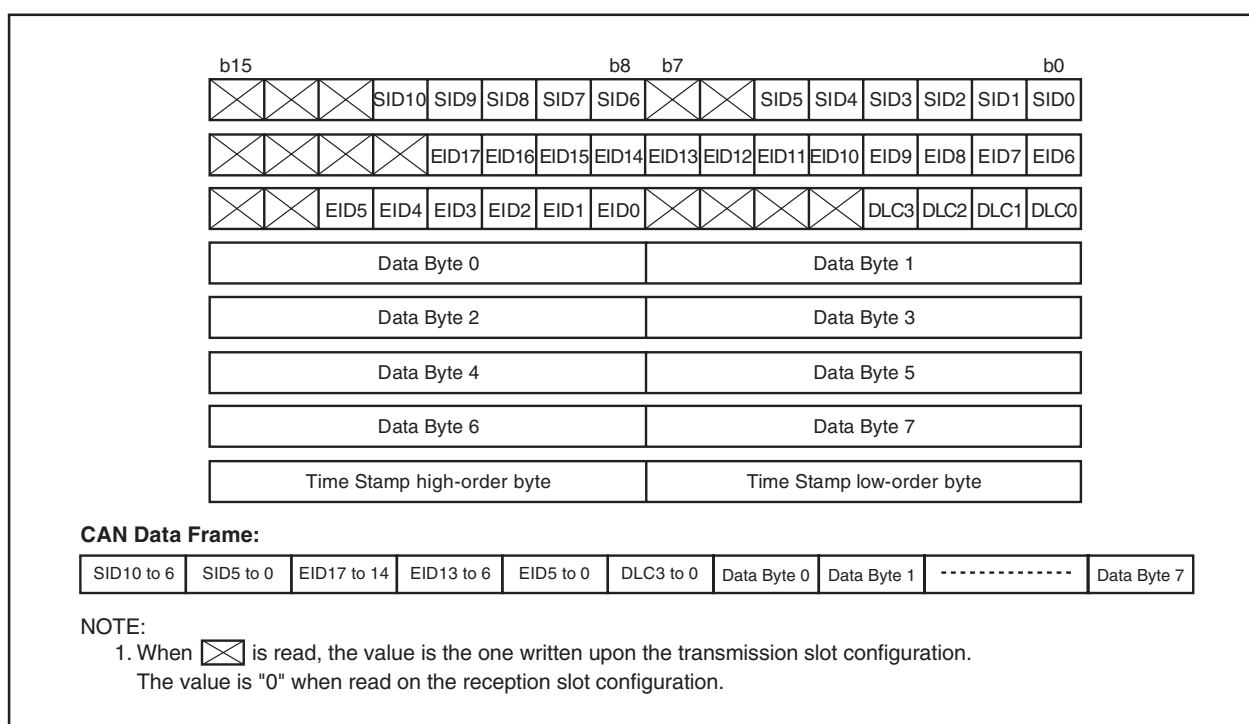
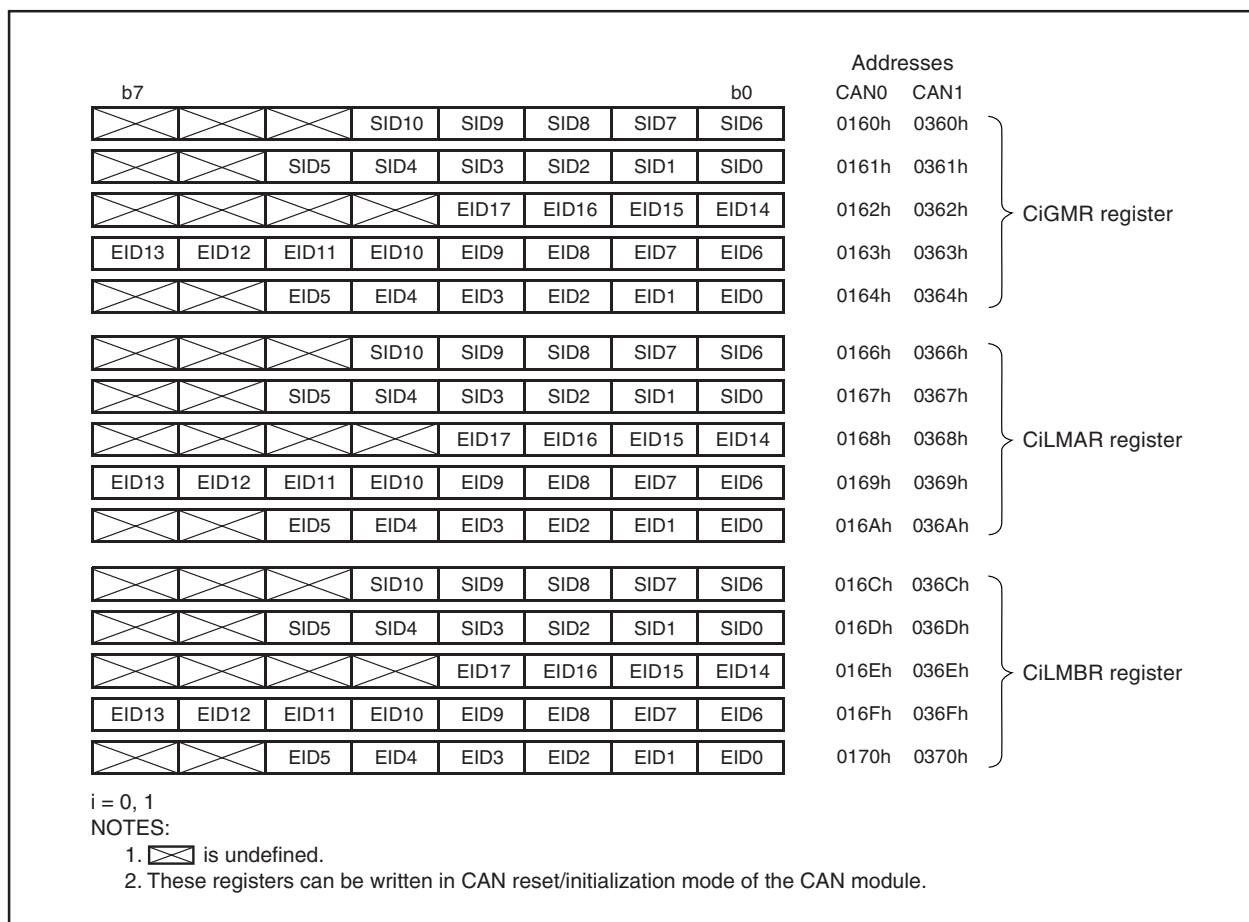


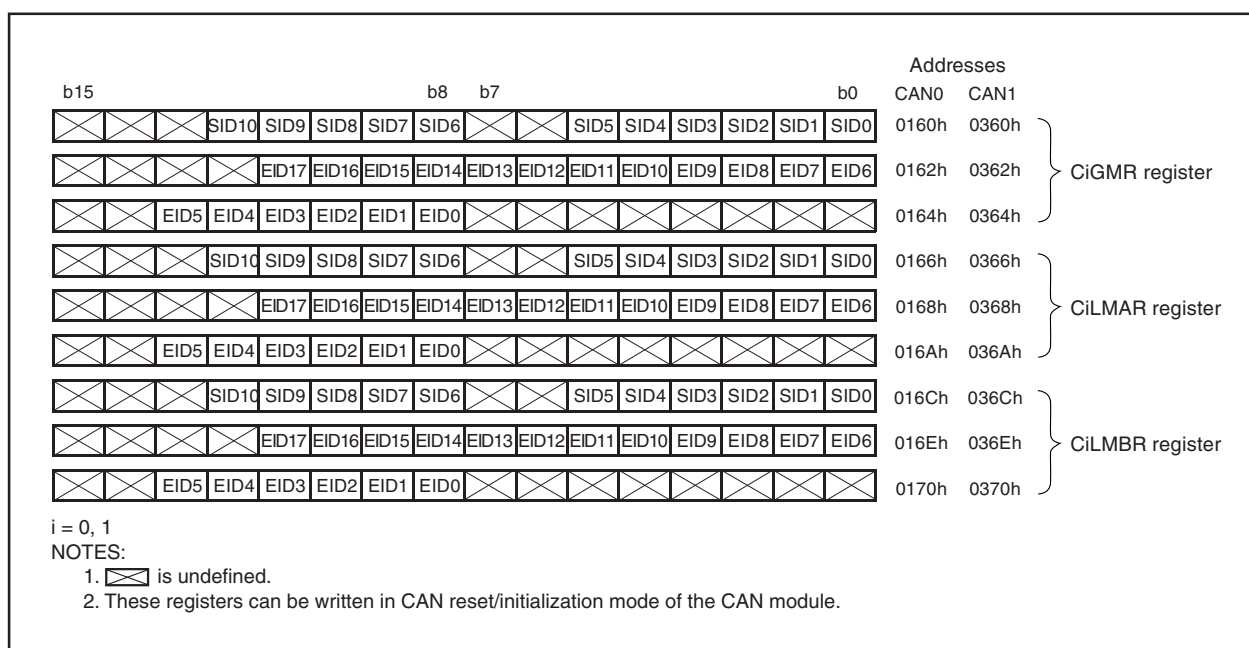
Figure 19.3 Bit Mapping in Word Access

### 19.3 Acceptance Mask Registers

Figures 19.4 and 19.5 show the CiGMR register ( $i = 0, 1$ ), the CiLMAR register, and the CiLMBR register, in which bit mapping in byte access and word access are shown.



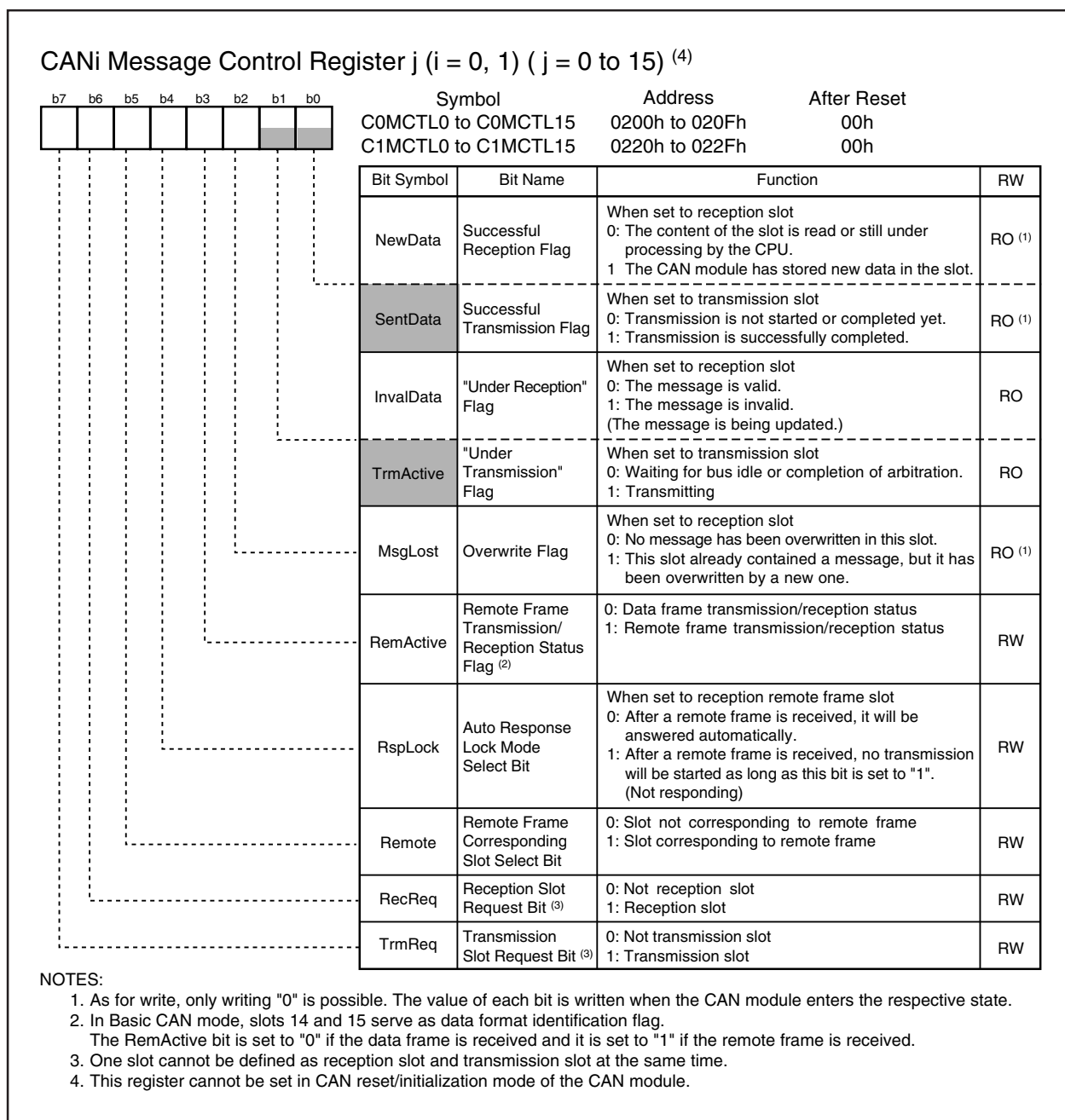
**Figure 19.4 Bit Mapping of Mask Registers in Byte Access**



**Figure 19.5 Bit Mapping of Mask Registers in Word Access**

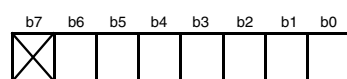
## 19.4 CAN SFR Registers

Figures 19.6 to 19.11 show the CAN SFR registers.



**Figure 19.6 C0MCTLj and C1MCTLj Registers**

## CANi Control Register (i = 0, 1)

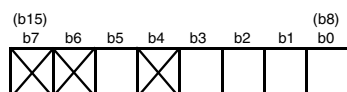


Symbol	Address	After Reset
C0CTLR	0210h	X0000001b
C1CTLR	0230h	X0000001b

Bit Symbol	Bit Name	Function	RW
Reset	CAN Module Reset Bit (1)	0: Operation mode 1: Reset/initialization mode	RW
LoopBack	Loop Back Mode Select Bit (2)	0: Loop back mode disabled 1: Loop back mode enabled	RW
MsgOrder	Message Order Select Bit (2)	0: Word access 1: Byte access	RW
BasicCAN	Basic CAN Mode Select Bit (2)	0: Basic CAN mode disabled 1: Basic CAN mode enabled	RW
BusErrEn	Bus Error Interrupt Enable Bit (2)	0: Bus error interrupt disabled 1: Bus error interrupt enabled	RW
Sleep	Sleep Mode Select Bit (2)(3)	0: Sleep mode disabled 1: Sleep mode enabled; clock supply stopped	RW
PortEn	CAN Port Enable Bit (2)(3)	0: I/O port function 1: CTX/CRX function	RW
— (b7)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

## NOTES:

1. When the Reset bit is set to "1" (CAN reset/initialization mode), check that the State\_Reset bit in the CiSTR register is set to "1" (Reset mode).
2. Change this bit only in the CAN reset/initialization mode.
3. When using CAN0/1 wake-up interrupt, set these bits to "1".



Symbol	Address	After Reset
C0CTLR	0211h	XX0X0000b
C1CTLR	0231h	XX0X0000b

Bit Symbol	Bit Name	Function	RW
TSPreScale	Time Stamp Prescaler (3)	b1 b0 0 0: Period of 1 bit time 0 1: Period of 1/2 bit time 1 0: Period of 1/4 bit time 1 1: Period of 1/8 bit time	RW
TSReset	Time Stamp Counter Reset Bit (1)	0: Nothing is occurred. 1: Force reset of the time stamp counter	RW
RetBusOff	Return From Bus Off Command Bit (2)	0: Nothing is occurred. 1: Force return from bus off	RW
— (b4)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
RXOnly	Listen-Only Mode Select Bit (3)	0: Listen-only mode disabled 1: Listen-only mode enabled (4)	RW
— (b7-b6)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—

## NOTES:

1. When the TSReset bit = 1, the CiTSR register is set to "0000h". After this, the bit is automatically set to "0".
2. When the RetBusOff bit = 1, the CiRECR and CiTECR registers are set to "00h". After this, this bit is automatically set to "0".
3. Change this bit only in the CAN reset/initialization mode.
4. When the listen-only mode is selected, do not request the transmission.

Figure 19.7 C0CTLR and C1CTLR Registers

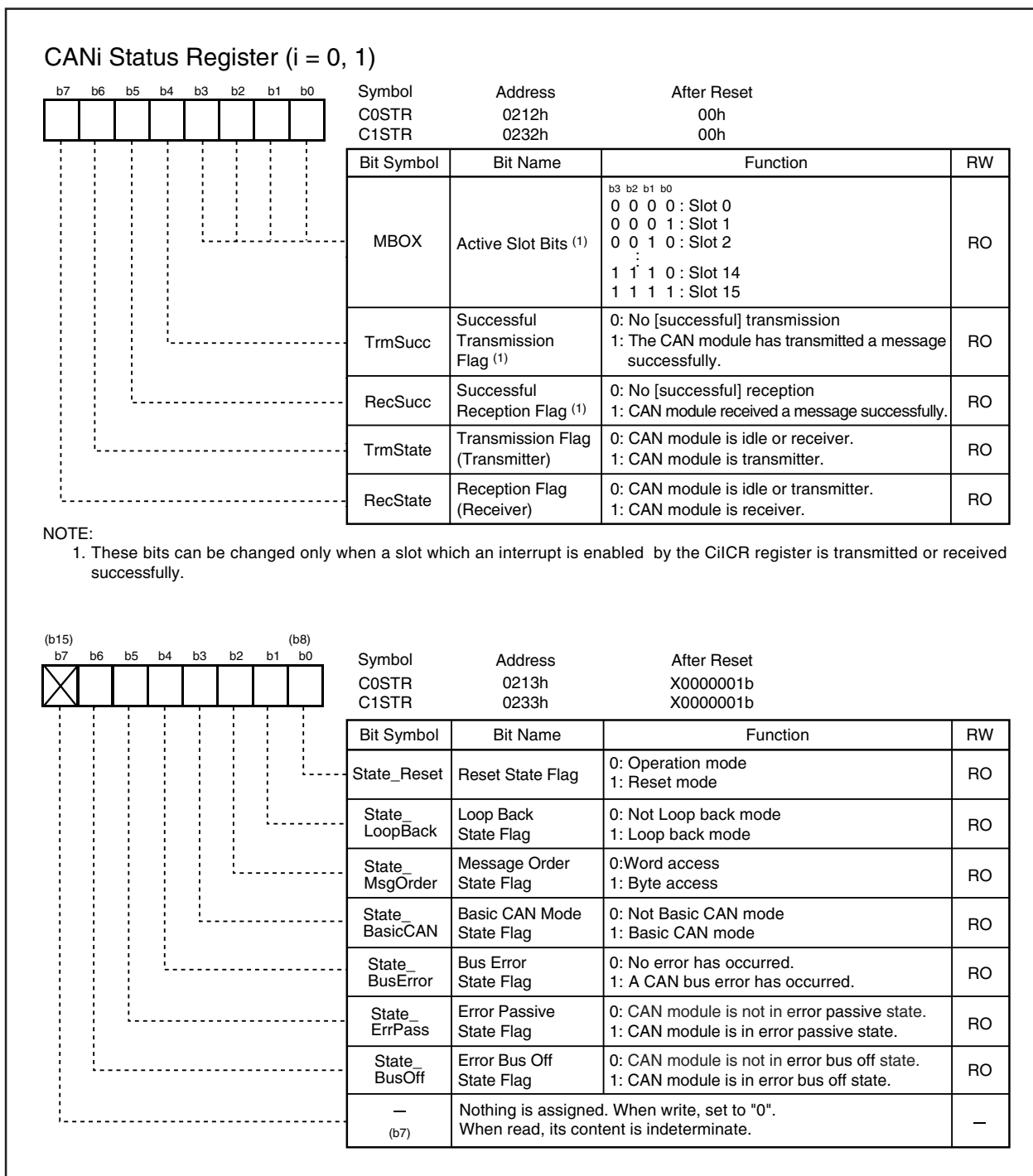


Figure 19.8 C0STR and C1STR Registers

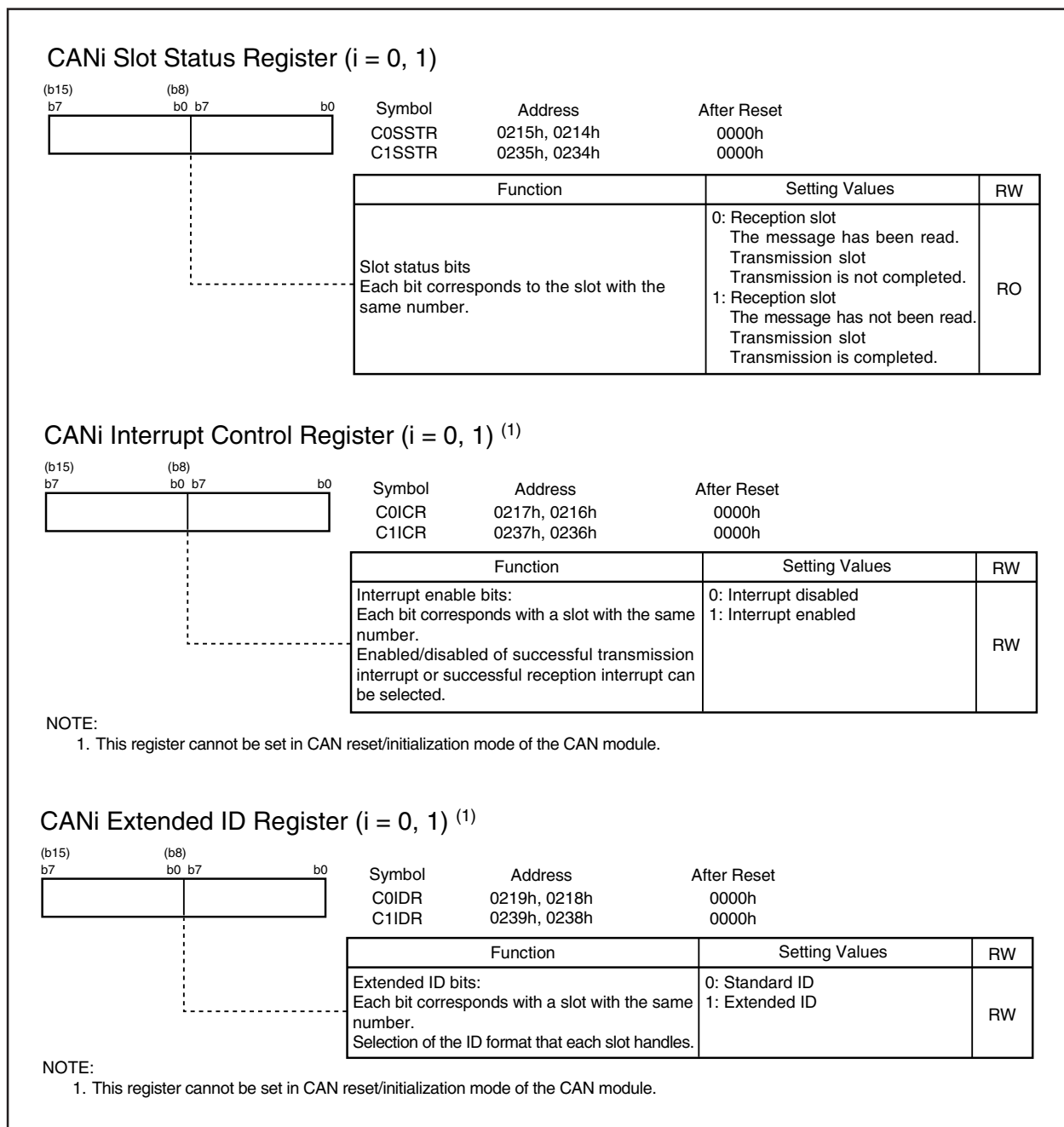


Figure 19.9 C0SSTR, C1SSTR Registers, C0ICR, C1ICR Registers, and C0IDR, C1IDR Registers



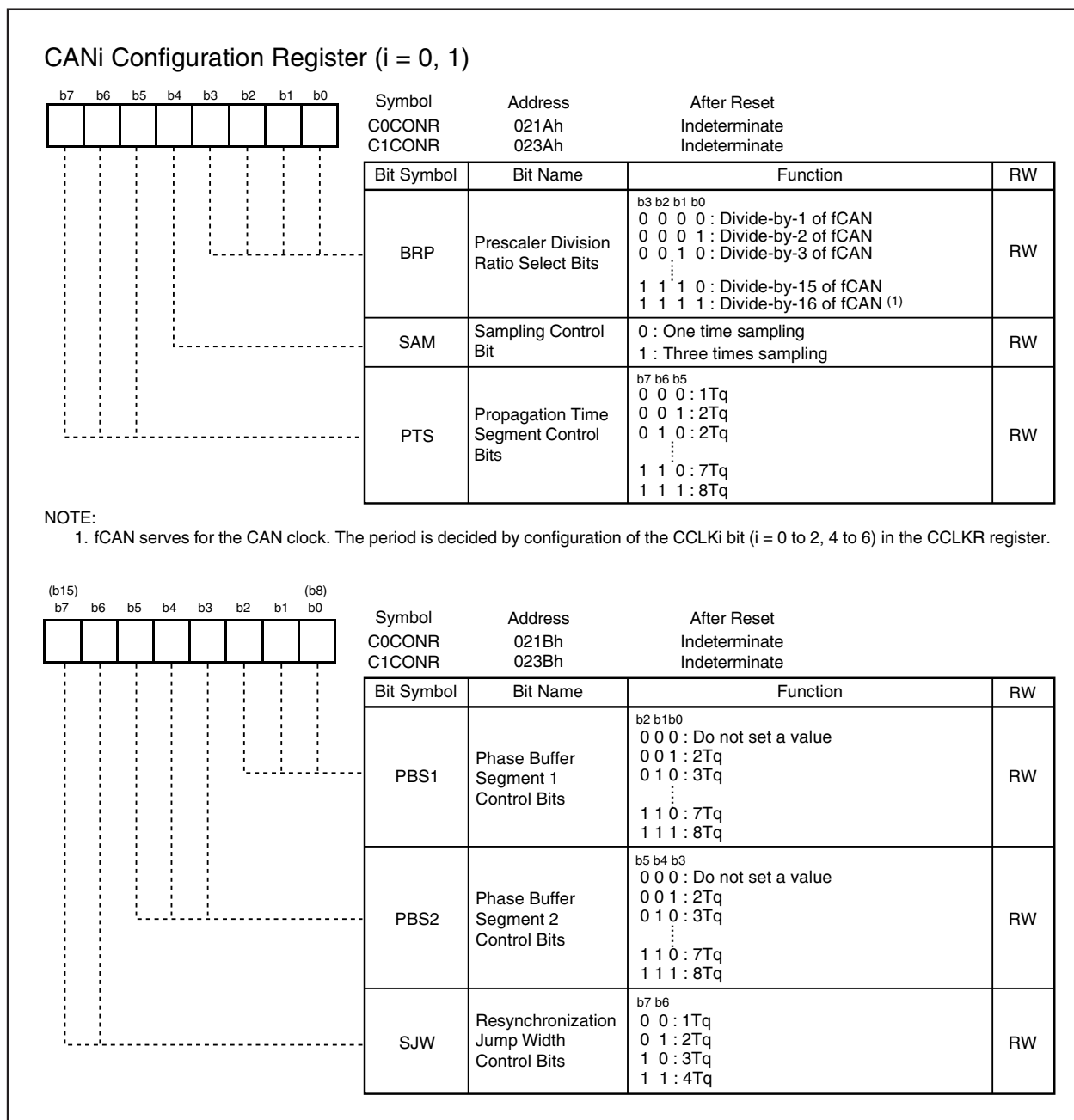
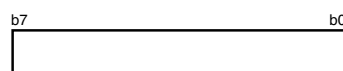


Figure 19.10 C0CONR and C1CONR Registers

## CANi Receive Error Count Register (i = 0, 1)



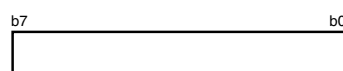
Symbol	Address	After Reset
C0RECR	021Ch	00h
C1RECR	023Ch	00h

Function	Counter Value	RW
Reception error counting function The value is incremented or decremented according to the CAN module's error status.	00h to FFh <sup>(1)</sup>	RO

## NOTE:

1. The value is indeterminate in bus off state.

## CANi Transmit Error Count Register (i = 0, 1)



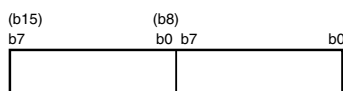
Symbol	Address	After Reset
C0TECR	021Dh	00h
C1TECR	023Dh	00h

Function	Counter Value	RW
Transmission error counting function The value is incremented or decremented according to the CAN module's error status.	00h to FFh <sup>(1)</sup>	RO

## NOTE:

1. The value is indeterminate in bus off state.

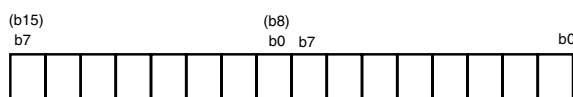
## CANi Time Stamp Register (i = 0, 1)



Symbol	Address	After Reset
C0TSR	021Fh, 021Eh	0000h
C1TSR	023Fh, 023Eh	0000h

Function	Counter Value	RW
Time stamp function	0000h to FFFFh	RO

## CANi Acceptance Filter Support Register (i = 0, 1)



Symbol	Address	After Reset
C0AFS	0243h, 0242h	Indeterminate
C1AFS	0245h, 0244h	Indeterminate

Function	Setting Values	RW
Write the content equivalent to the standard frame ID of the received message. The value is "converted standard frame ID" when read.	Standard frame ID	RW

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

## 19.5 Operational Modes

The CAN module has the following four operational modes.

- CAN Reset/Initialization Mode
- CAN Operation Mode
- CAN Sleep Mode
- CAN Interface Sleep Mode

Figure 19.12 shows transition between operational modes.

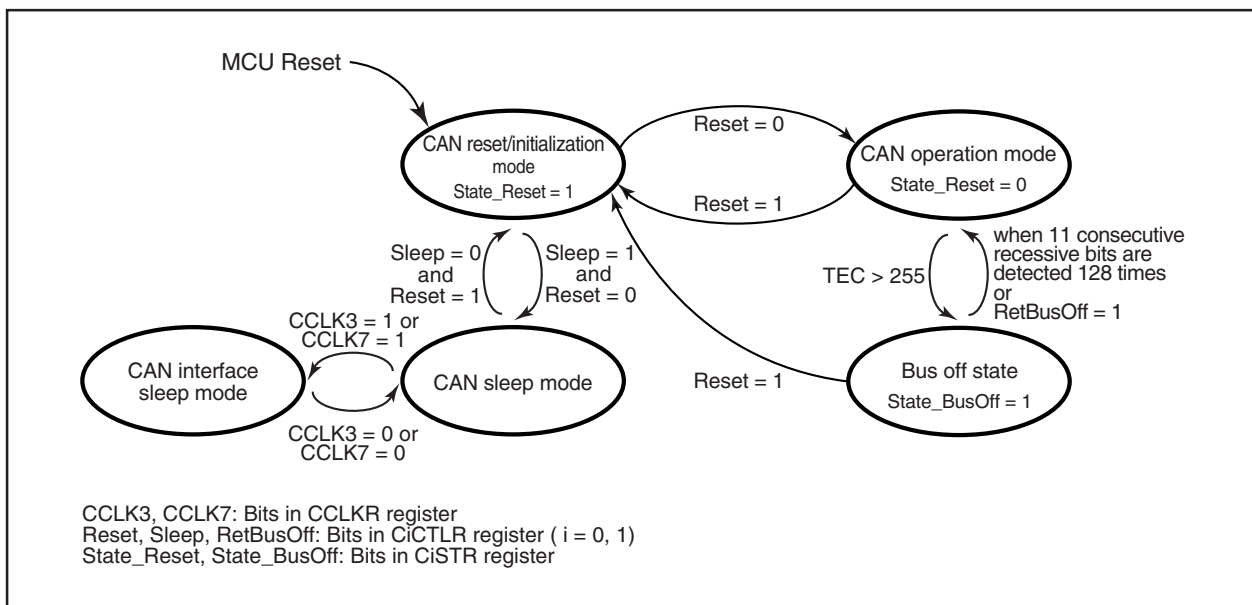


Figure 19.12 Transition Between Operational Modes

### 19.5.1 CAN Reset/Initialization Mode

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

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

- CAN communication is impossible.
- When the CAN reset/initialization mode is activated during an ongoing transmission in operation mode, the module suspends the mode transition until completion of the transmission (successful, arbitration loss, or error detection). Then, the State\_Reset bit is set to “1”, and the CAN reset/initialization mode is activated.
- The CiMCTLj (j = 0 to 15), CiSTR, CiICR, CiIDR, CiRECR, CiTECR and CiTSR registers are initialized. All these registers are locked to prevent CPU modification.
- The CiCTLR, CiCONR, CiGMR, CiLMAR and CiLMBR registers and the CANi message box retain their contents and are available for CPU access.

### 19.5.2 CAN Operation Mode

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

If 11 consecutive recessive bits are detected after entering the CAN operation mode, the module initiates the following functions:

- The module's communication functions are released and it becomes an active node on the network and may transmit and receive CAN messages.
- Release the internal fault confinement logic including receive and transmit error counters. The module may leave the CAN operation mode depending on the error counts.

Within the CAN operation mode, the module may be in three different sub modes, depending on which type of communication functions are performed:

- Module idle : The modules receive and transmit sections are inactive.
- Module receives : The module receives a CAN message sent by another node.
- Module transmits : The module transmits a CAN message. The module may receive its own message simultaneously when the LoopBack bit in the CiCTLR register = 1 (Loop back mode enabled).

Figure 19.13 shows sub modes of the CAN operation mode.

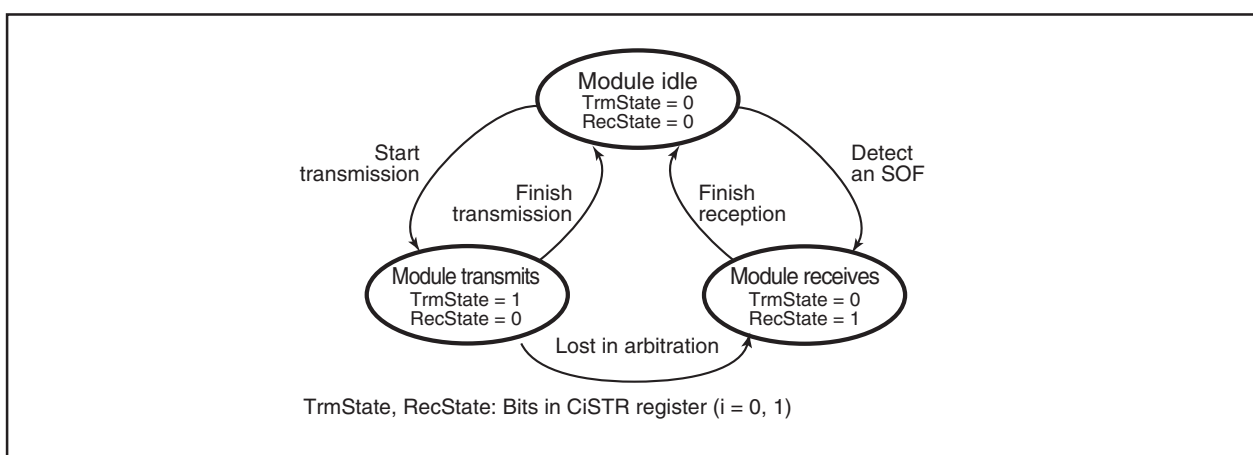


Figure 19.13 Sub Modes of CAN Operation Mode

### 19.5.3 CAN Sleep Mode

The CAN sleep mode is activated by setting the Sleep bit to “1” and the Reset bit to “0” in the CiCTLR register. It should never be activated from the CAN operation mode but only via the CAN reset/initialization mode.

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

### 19.5.4 CAN Interface Sleep Mode

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

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

### 19.5.5 Bus Off State

The bus off state is entered according to the fault confinement rules of the CAN specification. When returning to the CAN operation mode from the bus off state, the module has the following two cases. In this time, the value of any CAN registers, except CiSTR, CiRECR and CiTECR registers, does not change.

- (1) When 11 consecutive recessive bits are detected 128 times

The module enters instantly into error active state and the CAN communication becomes possible immediately.

- (2) When the RetBusOff bit in the CiCTLR register = 1 (Force return from buss off)

The module enters instantly into error active state, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected.

## 19.6 Configuration CAN Module System Clock

The M16C/6N Group (M16C/6N4) has a CAN module system clock select circuit.

Configuration of the CAN module system clock can be done through manipulating the CCLKR register and the BRP bit in the CiCONR register ( $i = 0, 1$ ).

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

Figure 19.14 shows a block diagram of the clock generating circuit of the CAN module system.

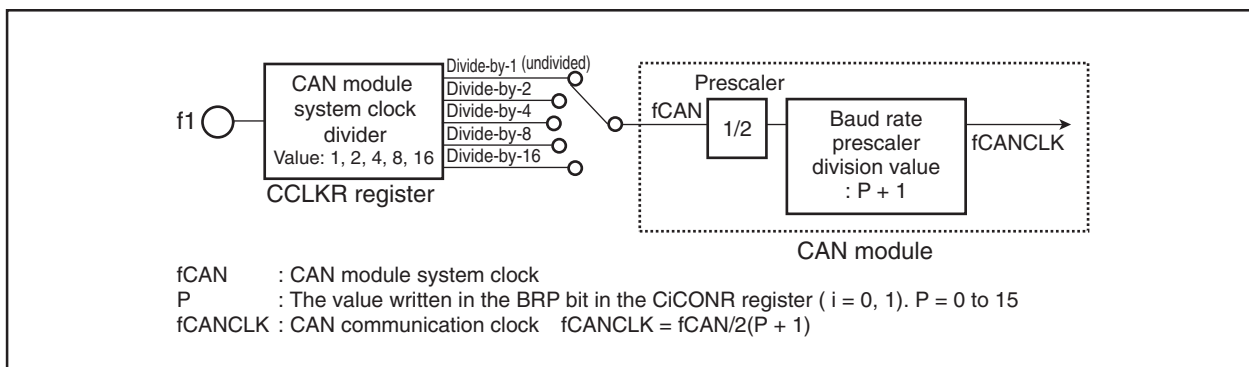


Figure 19.14 Block Diagram of CAN Module System Clock Generating Circuit

## 19.7 Bit Timing Configuration

The bit time consists of the following four segments:

- Synchronization segment (SS)  
This serves for monitoring a falling edge for synchronization.
- Propagation time segment (PTS)  
This segment absorbs physical delay on the CAN network which amounts to double the total sum of delay on the CAN bus, the input comparator delay, and the output driver delay.
- Phase buffer segment 1 (PBS1)  
This serves for compensating the phase error. When the falling edge of the bit falls later than expected, the segment can become longer by the maximum of the value defined in SJW.
- Phase buffer segment 2 (PBS2)  
This segment has the same function as the phase buffer segment 1. When the falling edge of the bit falls earlier than expected, the segment can become shorter by the maximum of the value defined in SJW.

Figure 19.15 shows the bit timing.

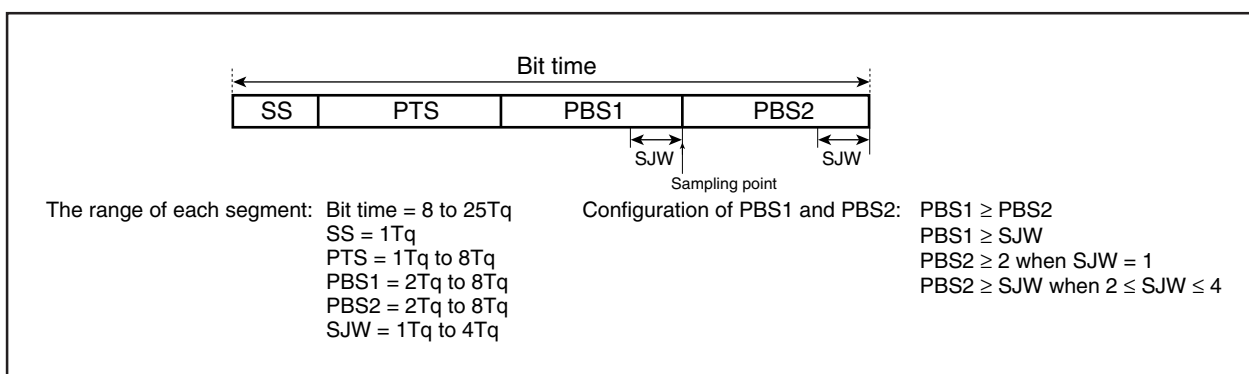


Figure 19.15 Bit Timing

## 19.8 Bit-rate

Bit-rate depends on  $f_1$ , the division value of the CAN module system clock, the division value of the baud rate prescaler, and the number of  $T_q$  of one bit.

Table 19.2 shows the examples of bit-rate.

**Table 19.2 Examples of Bit-rate**

Bit-rate	24MHz <sup>(2)</sup>	20MHz	16MHz	10MHz	8MHz
1Mbps	12Tq (1)	10Tq (1)	8Tq (1)	–	–
500kbps	12Tq (2)	10Tq (2)	8Tq (2)	10Tq (1)	8Tq (1)
	24Tq (1)	20Tq (1)	16Tq (1)	–	–
125kbps	12Tq (8)	10Tq (8)	8Tq (8)	10Tq (4)	8Tq (4)
	16Tq (6)	20Tq (4)	16Tq (4)	20Tq (2)	16Tq (2)
	24Tq (4)	–	–	–	–
83.3kbps	12Tq (12)	10Tq (12)	8Tq (12)	10Tq (6)	8Tq (6)
	16Tq (9)	20Tq (6)	16Tq (6)	20Tq (3)	16Tq (3)
	24Tq (6)	–	–	–	–
33.3kbps	12Tq (30)	10Tq (30)	8Tq (30)	10Tq (15)	8Tq (15)
	24Tq (15)	20Tq (15)	16Tq (15)	–	–

**NOTES:**

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

### 19.8.1 Calculation of Bit-rate

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

**NOTES:**

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

## 19.9 Acceptance Filtering Function and Masking Function

These functions serve the users to select and receive a facultative message. The CiGMR register ( $i = 0, 1$ ), the CiLMAR register, and the CiLMBR register can perform masking to the standard ID and the extended ID of 29 bits. The CiGMR register corresponds to slots 0 to 13, the CiLMAR register corresponds to slot 14, and the CiLMBR register corresponds to slot 15. The masking function becomes valid to 11 bits or 29 bits of a received ID according to the value in the corresponding slot of the CiIDR register upon acceptance filtering operation. When the masking function is employed, it is possible to receive a certain range of IDs. Figure 19.16 shows correspondence of the mask registers and slots, Figure 19.17 shows the acceptance function.

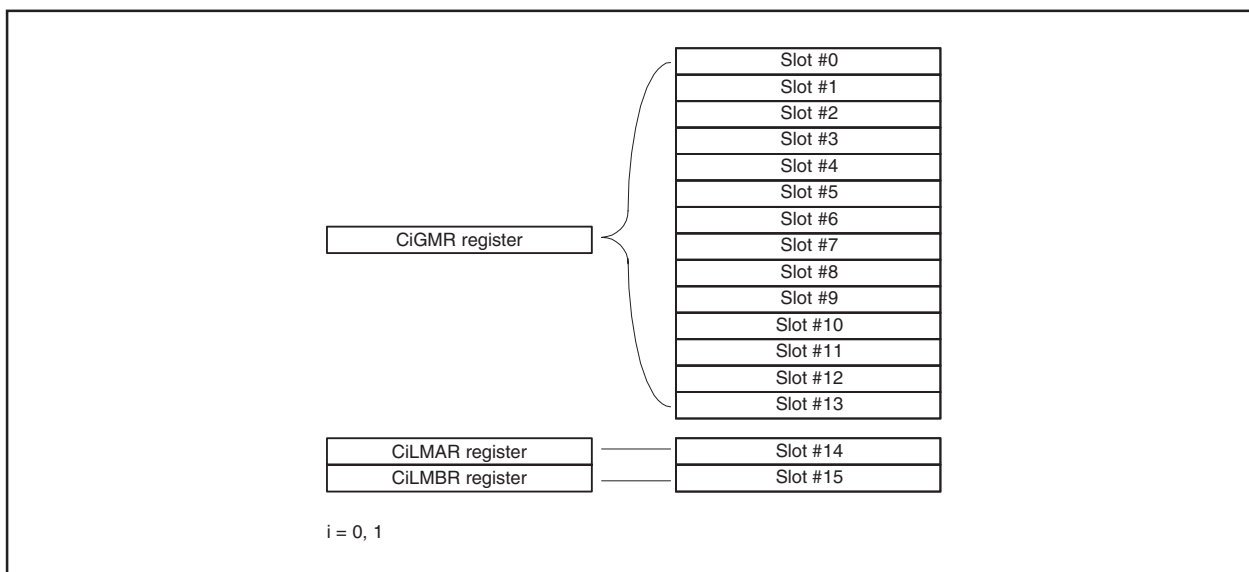


Figure 19.16 Correspondence of Mask Registers to Slots

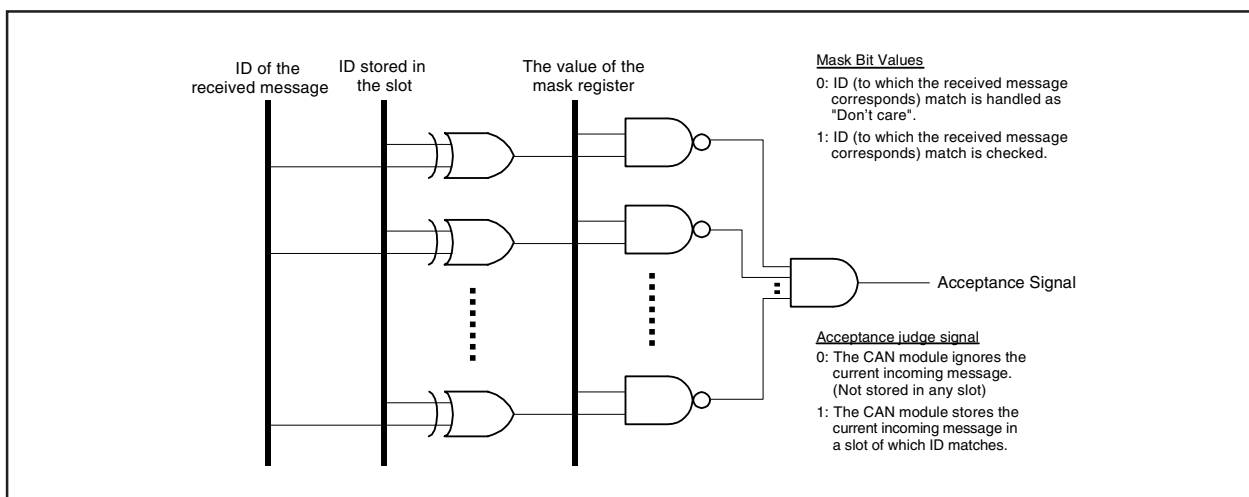


Figure 19.17 Acceptance Function

When using the acceptance function, note the following points.

- (1) When one ID is defined in two slots, the one with a smaller number alone is valid.
- (2) When it is configured that slots 14 and 15 receive all IDs with Basic CAN mode, slots 14 and 15 receive all IDs which are not stored into slots 0 to 13.



## 19.10 Acceptance Filter Support Unit (ASU)

The acceptance filter support unit has a function to judge valid/invalid of a received ID through table search. The IDs to receive are registered in the data table; a received ID is stored in the CiAFS register (  $i = 0, 1$  ), and table search is performed with a decoded received ID. The acceptance filter support unit can be used for the IDs of the standard frame only.

The acceptance filter support unit is valid in the following cases.

- When the ID to receive cannot be masked by the acceptance filter.  
(Example) IDs to receive: 078h, 087h, 111h
- When there are too many IDs to receive; it would take too much time to filter them by software.

Figure 19.18 shows the write and read of the CiAFS register in word access.

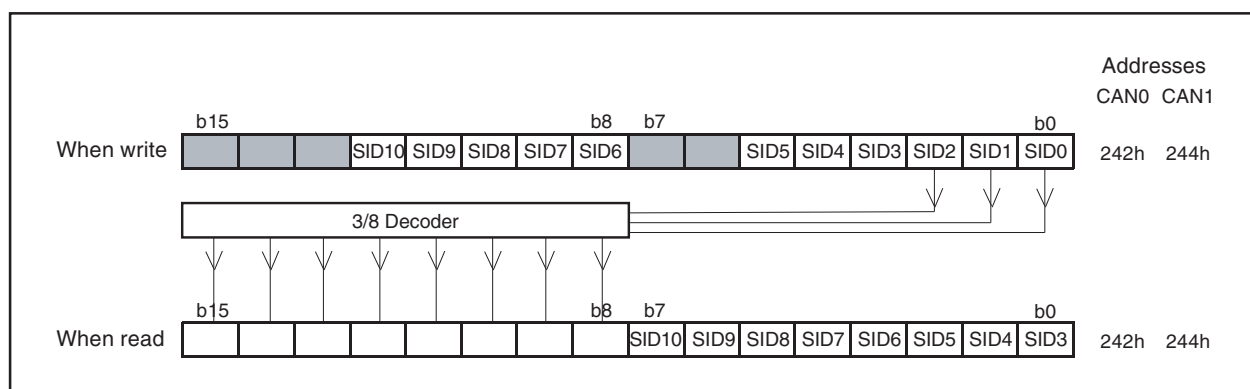


Figure 19.18 Write/read of CiAFS Register in Word Access

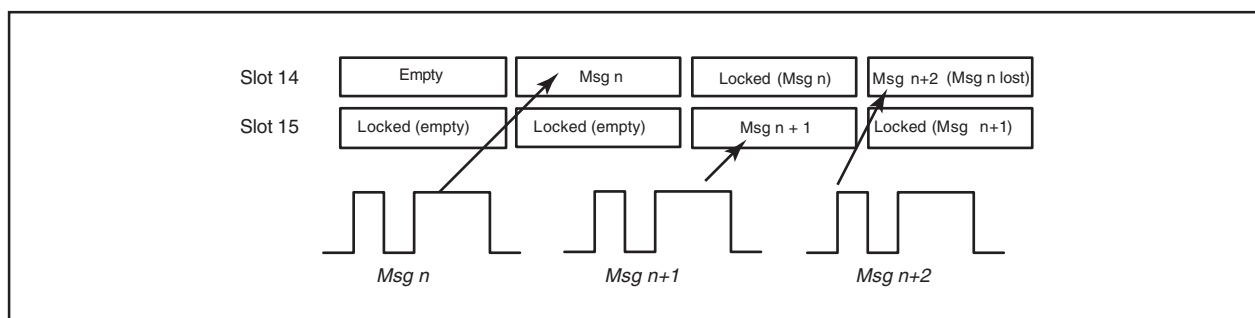
### 19.11 Basic CAN Mode

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

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

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

Figure 19.19 shows the operation of slots 14 and 15 in Basic CAN mode.



**Figure 19.19 Operation of Slots 14 and 15 in Basic CAN Mode**

When using Basic CAN mode, note the following points.

- (1) Setting of Basic CAN mode has to be done in CAN reset/initialization mode.
- (2) Select the same ID for slots 14 and 15. Also, setting of the CiLMAR and CiLMBR register has to be the same.
- (3) Define slots 14 and 15 as reception slot only.
- (4) There is no protection available against message overwrite. A message can be overwritten by a new message.
- (5) Slots 0 to 13 can be used in the same way as in normal CAN operation mode.

### 19.12 Return from Bus Off Function

When the protocol controller enters bus off state, it is possible to make it forced return from bus off state by setting the RetBusOff bit in the CiCTLR register ( $i = 0, 1$ ) to "1" (Force return from bus off). At this time, the error state changes from bus off state to error active state. If the RetBusOff bit is set to "1", the CiRECR and CiTECR registers are initialized and the State\_BusOff bit in the CiSTR register is set to "0" (CAN module is not in error bus off state). However, registers of the CAN module such as CiCONR register and the content of each slot are not initialized.

### 19.13 Time Stamp Counter and Time Stamp Function

When the CiTSR register ( $i = 0, 1$ ) is read, the value of the time stamp counter at the moment is read. The period of the time stamp counter reference clock is the same as that of 1 bit time that is configured by the CiCONR register. The time stamp counter functions as a free run counter.

The 1 bit time period can be divided by 1 (undivided), 2, 4 or 8 to produce the time stamp counter reference clock. Use the TSPreScale bit in the CiCTLR register to select the divide-by-n value.

The time stamp counter is equipped with a register that captures the counter value when the protocol controller regards it as a successful reception. The captured value is stored when a time stamp value is stored in a reception slot.

### 19.14 Listen-Only Mode

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

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

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

## 19.15 Reception and Transmission

Table 19.3 shows configuration of CAN reception and transmission mode.

**Table 19.3 Configuration of CAN Reception and Transmission Mode**

TrmReq	RecReq	Remote	RspLock	Communication Mode of Slot
0	0	-	-	Communication environment configuration mode: configure the communication mode of the slot.
0	1	0	0	Configured as a reception slot for a data frame.
1	0	1	0	Configured as a transmission slot for a remote frame. (At this time the RemActive = 1.) After completion of transmission, this functions as a reception slot for a data frame. (At this time the RemActive = 0.) However, when an ID that matches on the CAN bus is detected before remote frame transmission, this immediately functions as a reception slot for a data frame.
1	0	0	0	Configured as a transmission slot for a data frame.
0	1	1	1/0	Configured as a reception slot for a remote frame. (At this time the RemActive = 1.) After completion of reception, this functions as a transmission slot for a data frame. (At this time the RemActive = 0.) However, transmission does not start as long as RspLock bit remains "1"; thus no automatic response. Response (transmission) starts when the RspLock bit is set to "0".

TrmReq, RecReq, Remote, RspLock, RemActive, RspLock: Bits in CiMCTLj register (i = 0, 1, j = 0 to 15)

When configuring a slot as a reception slot, note the following points.

- (1) Before configuring a slot as a reception slot, be sure to set the CiMCTLj register to "00h".
- (2) A received message is stored in a slot that matches the condition first according to the result of reception mode configuration and acceptance filtering operation. Upon deciding in which slot to store, the smaller the number of the slot is, the higher priority it has.
- (3) In normal CAN operation mode, when a CAN module transmits a message of which ID matches, the CAN module never receives the transmitted data. In loop back mode, however, the CAN module receives back the transmitted data. In this case, the module does not return ACK.

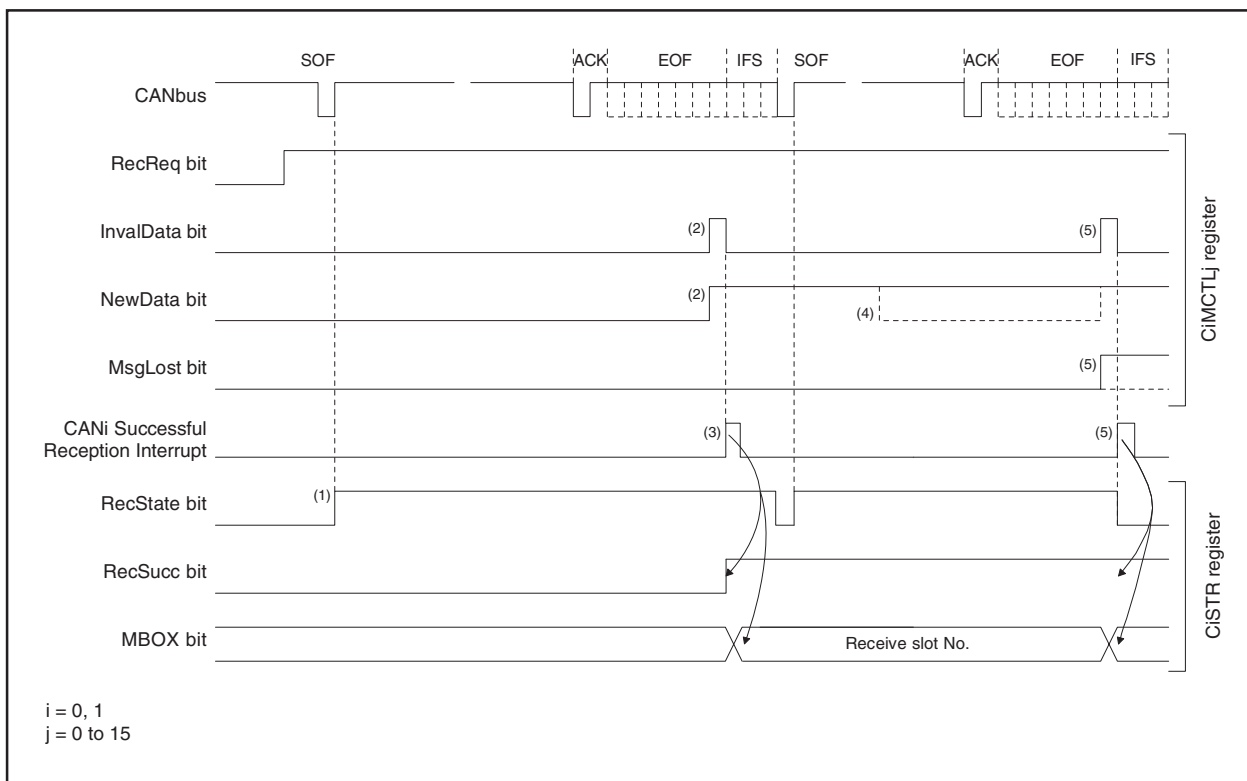
When configuring a slot as a transmission slot, note the following points.

- (1) Before configuring a slot as a transmission slot, be sure to set the CiMCTLj registers to "00h".
- (2) Set the TrmReq bit in the CiMCTLj register to "0" (not transmission slot) before rewriting a transmission slot.
- (3) A transmission slot should not be rewritten when the TrmActive bit in the CiMCTLj register is "1" (transmitting).

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

### 19.15.1 Reception

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

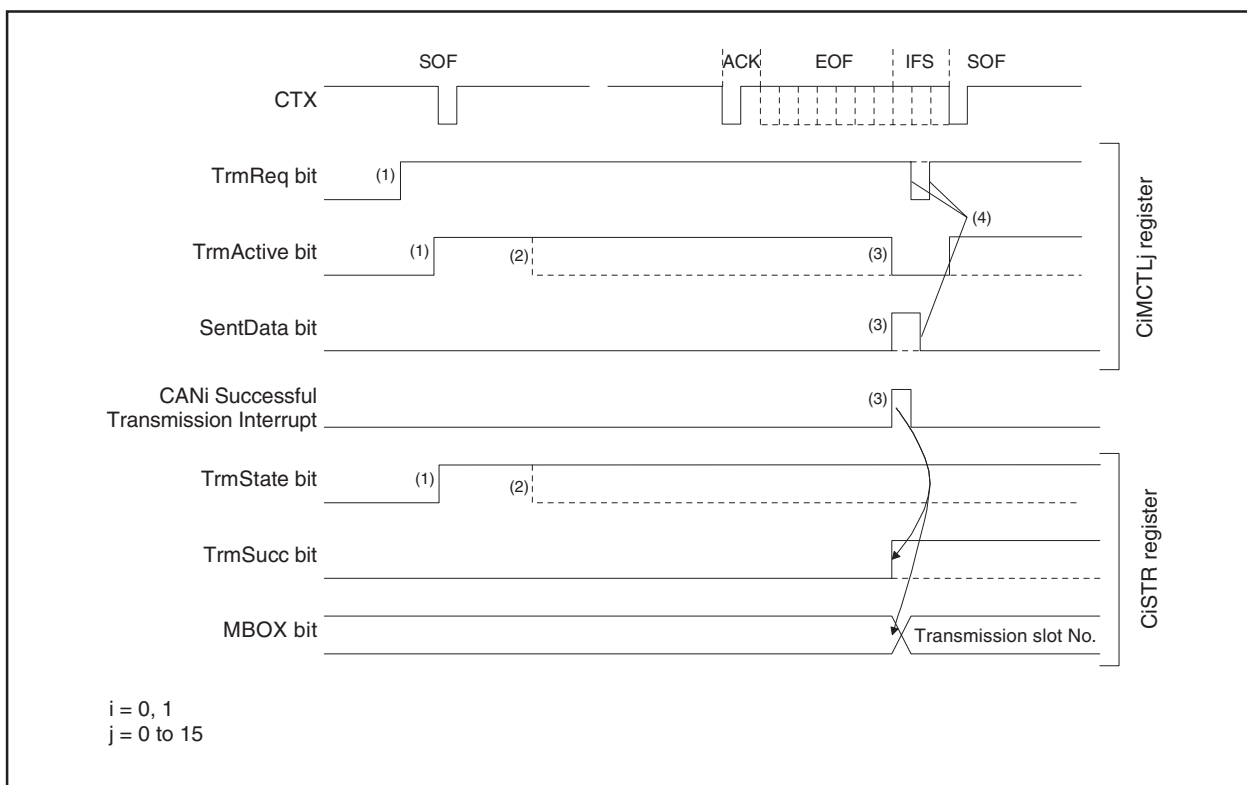


**Figure 19.20 Timing of Receive Data Frame Sequence**

- (1) On monitoring a SOF on the CAN bus the RecState bit in the CiSTR register becomes "1" (CAN module is receiver) immediately, given the module has no transmission pending.
- (2) After successful reception of the message, the NewData bit in the CiMCTLj register of the receiving slot becomes "1" (stored new data in slot). The InvalData bit in the CiMCTLj register becomes "1" (message is being updated) at the same time and the InvalData bit becomes "0" (message is valid) again after the complete message was transferred to the slot.
- (3) When the interrupt enable bit in the CiICR register of the receiving slot = 1 (interrupt enabled), the CANi successful reception interrupt request is generated and the MBOX bit in the CiSTR register is changed. It shows the slot number where the message was stored and the RecSucc bit in the CiSTR register is active.
- (4) Read the message out of the slot after setting the New Data bit to "0" (the content of the slot is read or still under processing by the CPU) by a program.
- (5) When next CAN message is received before the the NewData bit is set to "0" by a program or a receive request to a slot is canceled, the MsgLost bit in the CiMCTLj register is set to "1" (message has been overwritten). The new received message is transferred to the slot. Generating of an interrupt request and change of the CiSTR register are same as in 3).

### 19.15.2 Transmission

Figure 19.21 shows the timing of the transmit sequence.



**Figure 19.21 Timing of Transmit Sequence**

- (1) If the TrmReq bit in the CiMCTLj register ( $i = 0, 1, j = 0 \text{ to } 15$ ) is set to "1" (Transmission slot) in the bus idle state, the TrmActive bit in the CiMCTLj register and the TrmState bit in the CiSTR register are set to "1" (Transmitting/Transmitter), and CAN module starts the transmission.
- (2) If the arbitration is lost after the CAN module starts the transmission, the TrmActive and TrmState bits are set to "0".
- (3) If the transmission has been successful without lost in arbitration, the SentData bit in the CiMCTLj register is set to "1" (Transmission is successfully completed) and TrmActive bit is set to "0" (Waiting for bus idle or completion of arbitration). And when the interrupt enable bits in the CiICR register = 1 (Interrupt enabled), CANi successful transmission interrupt request is generated and the MBOX (the slot number which transmitted the message) and TrmSucc bit in the CiSTR register are changed.
- (4) When starting the next transmission, set the SentData and TrmReq bits to "0". And set the TrmReq bit to "1" after checking that the SentData and TrmReq bits are set to "0".

### 19.16 CAN Interrupt

The CAN module provides the following CAN interrupts.

- CANi Successful Reception Interrupt ( i = 0, 1)
- CANi Successful Transmission Interrupt
- CAN0/1 Error Interrupt: Error Passive State  
Error BusOff State  
Bus Error (this feature can be disabled separately)
- CAN0/1 Wake-up Interrupt

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

## 20. Programmable I/O Ports

The programmable input/output ports (hereafter referred to simply as I/O ports) consist of 87 lines P0 to P10 (except P8\_5). Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high every 4 lines. P8\_5 is an input-only port and does not have a pull-up resistor. Port P8\_5 shares the pin with  $\overline{\text{NMI}}$ , so that the  $\overline{\text{NMI}}$  input level can be read from the P8\_5 bit in the P8 register.

Figures 20.1 to 20.5 show the I/O ports. Figure 20.6 shows the I/O pins.

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

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

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

### 20.1 PDi Register (i = 0 to 10)

Figure 20.7 shows the PDi register.

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

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

No direction register bit for P8\_5 is available.

### 20.2 Pi Register (i = 0 to 10)

Figure 20.8 shows the Pi register.

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

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

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

### 20.3 PURj Register (j = 0 to 2)

Figure 20.9 shows the PURj register.

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

However, the pull-up control register has no effect on P0 to P3, P4\_0 to P4\_3, and P5 during memory expansion and microprocessor modes. Although the register contents can be modified, no pull-up resistors are connected.



## 20.4 PCR Register

Figure 20.10 shows the PCR register.

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

Tables 20.1 and 20.2 list an example connection of unused pins. Figure 20.11 shows an example connection of unused pins.

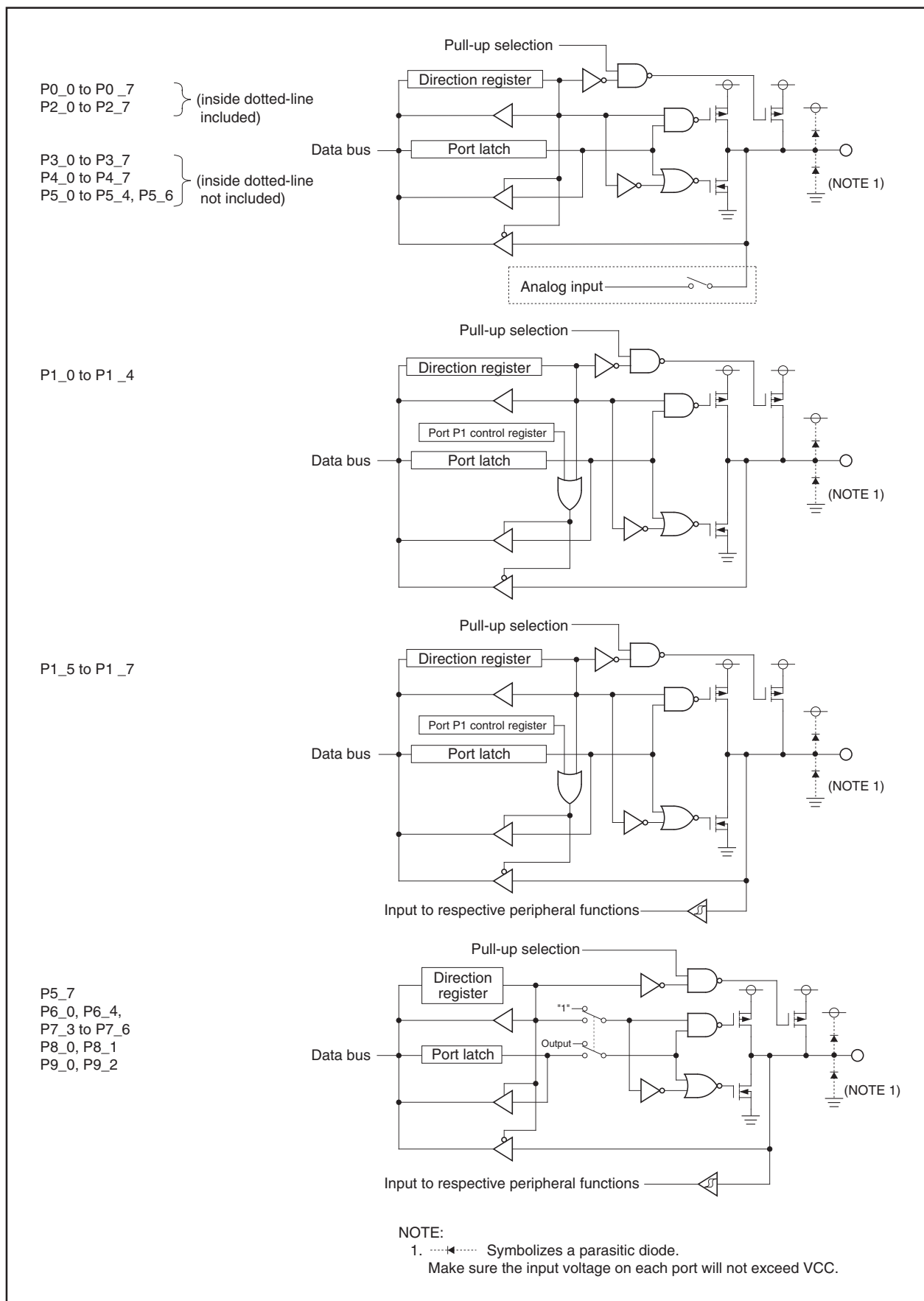


Figure 20.1 I/O Ports (1)

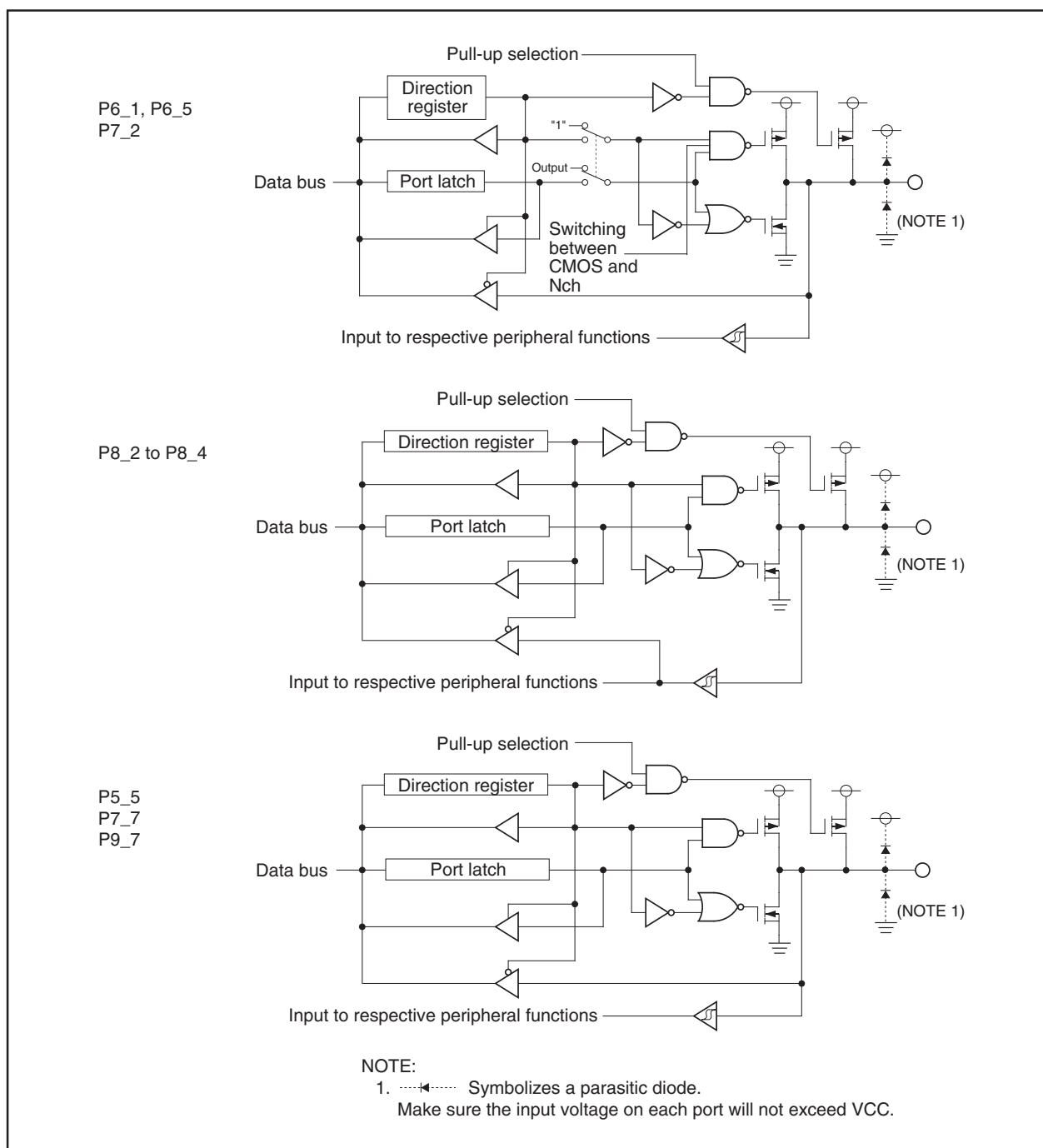


Figure 20.2 I/O Ports (2)

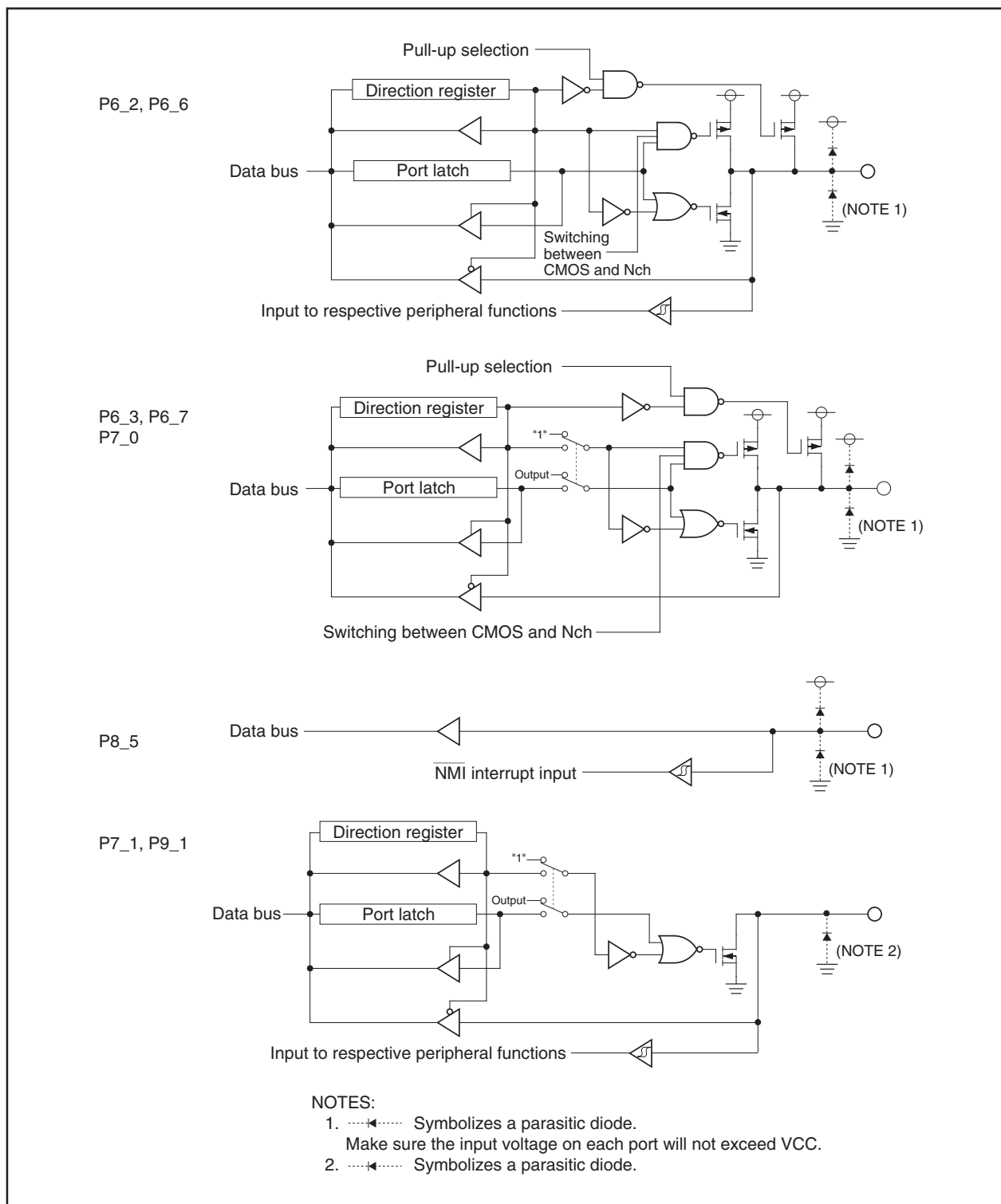


Figure 20.3 I/O Ports (3)

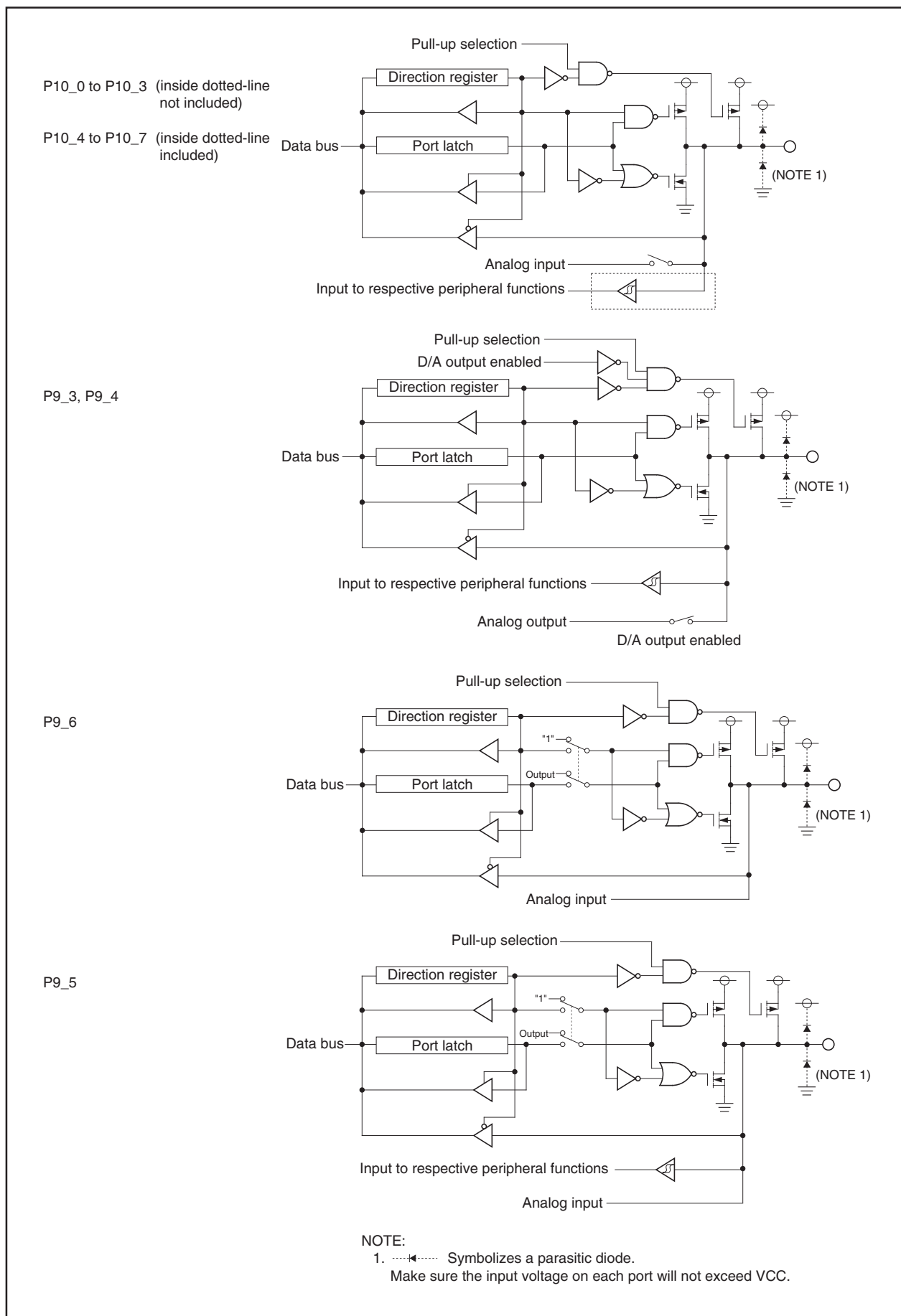


Figure 20.4 I/O Ports (4)

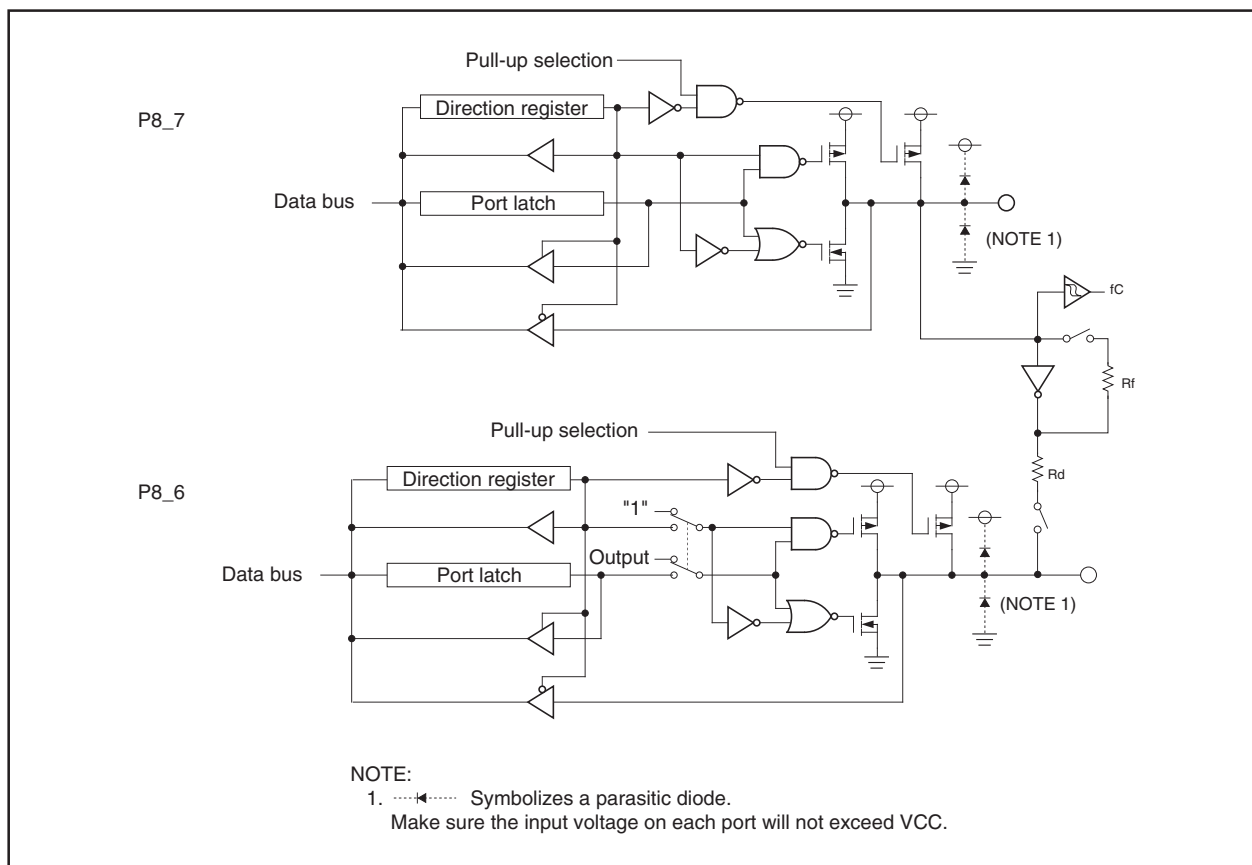


Figure 20.5 I/O Ports (5)

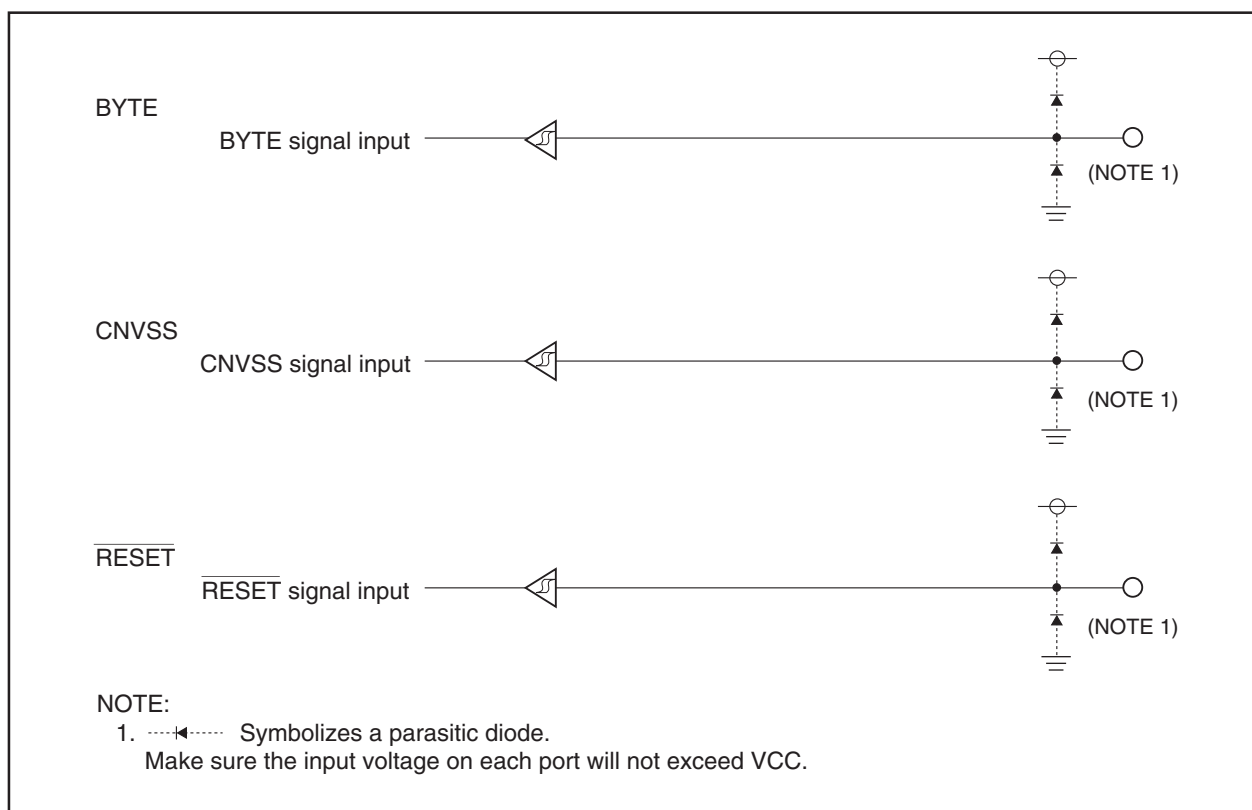


Figure 20.6 I/O Pins

Port Pi Direction Register (i = 0 to 7, 9, 10) <sup>(1) (2)</sup>

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
								PD0 to PD3	03E2h, 03E3h, 03E6h, 03E7h	00h
								PD4 to PD7	03EAh, 03EBh, 03EEh, 03EFh	00h
								PD9, PD10	03F3h, 03F6h	00h

Bit Symbol	Bit Name	Function	RW
PDi_0	Port Pi_0 Direction Bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PDi_1	Port Pi_1 Direction Bit		RW
PDi_2	Port Pi_2 Direction Bit		RW
PDi_3	Port Pi_3 Direction Bit		RW
PDi_4	Port Pi_4 Direction Bit		RW
PDi_5	Port Pi_5 Direction Bit		RW
PDi_6	Port Pi_6 Direction Bit		RW
PDi_7	Port Pi_7 Direction Bit		RW

## NOTES:

1. Make sure the PD7 and PD9 registers are written to by the next instruction after setting the PRC2 bit in the PRCR register to "1" (write enabled).
2. During memory expansion and microprocessor modes, the PD register for the pins functioning as bus control pins (A0 to A19, D0 to D15,  $\overline{CS}0$  to  $\overline{CS}3$ , RD, WRL/WR, WRH/BHE, ALE, RDY, HOLD, HLDA and BCLK) cannot be modified.

## Port P8 Direction Register


b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
		X						PD8	03F2h	00X00000b

Bit Symbol	Bit Name	Function	RW
PD8_0	Port P8_0 Direction Bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_1	Port P8_1 Direction Bit		RW
PD8_2	Port P8_2 Direction Bit		RW
PD8_3	Port P8_3 Direction Bit		RW
PD8_4	Port P8_4 Direction Bit		RW
- (b5)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		-
PD8_6	Port P8_6 Direction Bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_7	Port P8_7 Direction Bit		RW

Figure 20.7 PD0 to PD10 Registers

Port Pi Register (i = 0 to 7, 9, 10) <sup>(1)</sup>

								Symbol	Address	After Reset	
								P0 to P3	03E0h, 03E1h, 03E4h, 03E5h	Indeterminate	
								P4 to P7	03E8h, 03E9h, 03ECh, 03EDh	Indeterminate	
								P9, P10	03F1h, 03F4h	Indeterminate	
								Bit Symbol	Bit Name	Function	RW
								Pi_0	Port Pi_0 Bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register.	RW
								Pi_1	Port Pi_1 Bit		RW
								Pi_2	Port Pi_2 Bit		RW
								Pi_3	Port Pi_3 Bit	The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register.	RW
								Pi_4	Port Pi_4 Bit		RW
								Pi_5	Port Pi_5 Bit		RW
								Pi_6	Port Pi_6 Bit	0 : "L" level 1 : "H" level <sup>(2)</sup>	RW
								Pi_7	Port Pi_7 Bit		RW

## NOTES:

- During memory expansion and microprocessor modes, the Pi register for the pins functioning as bus control pins (A0 to A19, D0 to D15,  $\overline{CS0}$  to  $\overline{CS3}$ ,  $\overline{RD}$ ,  $\overline{WRL/WR}$ ,  $\overline{WRH/BHE}$ , ALE,  $\overline{RDY}$ ,  $\overline{HOLD}$ ,  $\overline{HLDA}$  and  $\overline{BCLK}$ ) cannot be modified.
- Since P7\_1 and P9\_1 are N channel open-drain ports, the data is high-impedance.

## Port P8 Register

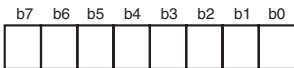
								Symbol P8	Address 03F0h	After Reset Indeterminate					
b7		b6		b5		b4		b3		b2		b1		b0	
P8_0		P8_1		P8_2		P8_3		P8_4		P8_5		P8_6		P8_7	
Port P8 _0 Bit		Port P8 _1 Bit		Port P8 _2 Bit		Port P8 _3 Bit		Port P8 _4 Bit		Port P8 _5 Bit		Port P8 _6 Bit		Port P8 _7 Bit	
<p>The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register.</p> <p>The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register. (Except for P8_5.)</p> <p>0 : "L" level</p> <p>1 : "H" level</p>										RW					
										RW					
										RW					
										RW					
										RW					
										RO					
										RW					
										RW					

Figure 20.8 P0 to P10 Registers



Pull-up Control Register 0 <sup>(1)</sup>Symbol  
PUR0Address  
03FChAfter Reset  
00h

Bit Symbol	Bit Name	Function	RW
PU00	P0_0 to P0_3 Pull-Up	0 : Not pulled high 1 : Pulled high <sup>(2)</sup>	RW
PU01	P0_4 to P0_7 Pull-Up		RW
PU02	P1_0 to P1_3 Pull-Up		RW
PU03	P1_4 to P1_7 Pull-Up		RW
PU04	P2_0 to P2_3 Pull-Up		RW
PU05	P2_4 to P2_7 Pull-Up		RW
PU06	P3_0 to P3_3 Pull-Up		RW
PU07	P3_4 to P3_7 Pull-Up		RW

## NOTES:

- During memory expansion and microprocessor modes, the pins are not pulled high although their corresponding register contents can be modified.
- The pin for which this bit is "1" (pulled high) and the direction bit is "0" (input mode) is pulled high.

## Pull-up Control Register 1

Symbol  
PUR1Address  
03FDhAfter Reset <sup>(1)</sup>  
0000000b  
00000010b

Bit Symbol	Bit Name	Function	RW
PU10	P4_0 to P4_3 Pull-Up <sup>(2)</sup>	0 : Not pulled high 1 : Pulled high <sup>(5)</sup>	RW
PU11	P4_4 to P4_7 Pull-Up <sup>(3)</sup>		RW
PU12	P5_0 to P5_3 Pull-Up <sup>(2)</sup>		RW
PU13	P5_4 to P5_7 Pull-Up <sup>(2)</sup>		RW
PU14	P6_0 to P6_3 Pull-Up		RW
PU15	P6_4 to P6_7 Pull-Up		RW
PU16	P7_0, P7_2 and P7_3 Pull-Up <sup>(4)</sup>		RW
PU17	P7_4 to P7_7 Pull-Up		RW

## NOTES:

- The values after hardware reset is as follows:
  - 00000000b when input on CNVSS pin is "L".
  - 00000010b when input on CNVSS pin is "H".
 The values after software reset, watchdog timer reset and oscillation stop detection reset are as follows:
  - 00000000b when the PM 01 to PM00 bits in the PM0 register are "00b" (single-chip mode).
  - 00000010b when the PM 01 to PM00 bits are "01b" (memory expansion mode) or "11b" (microprocessor mode).
- During memory expansion and microprocessor modes, the pins are not pulled high although their corresponding register contents can be modified.
- If the PM01 to PM00 bits are set to "01b" (memory expansion mode) or "11b" (microprocessor mode) in a program during single-chip mode, the PU11 bit becomes "1".
- The P7\_1 pin does not have pull-up.
- The pin for which this bit is "1" (pulled high) and the direction bit is "0" (input mode) is pulled high.

## Pull-up Control Register 2

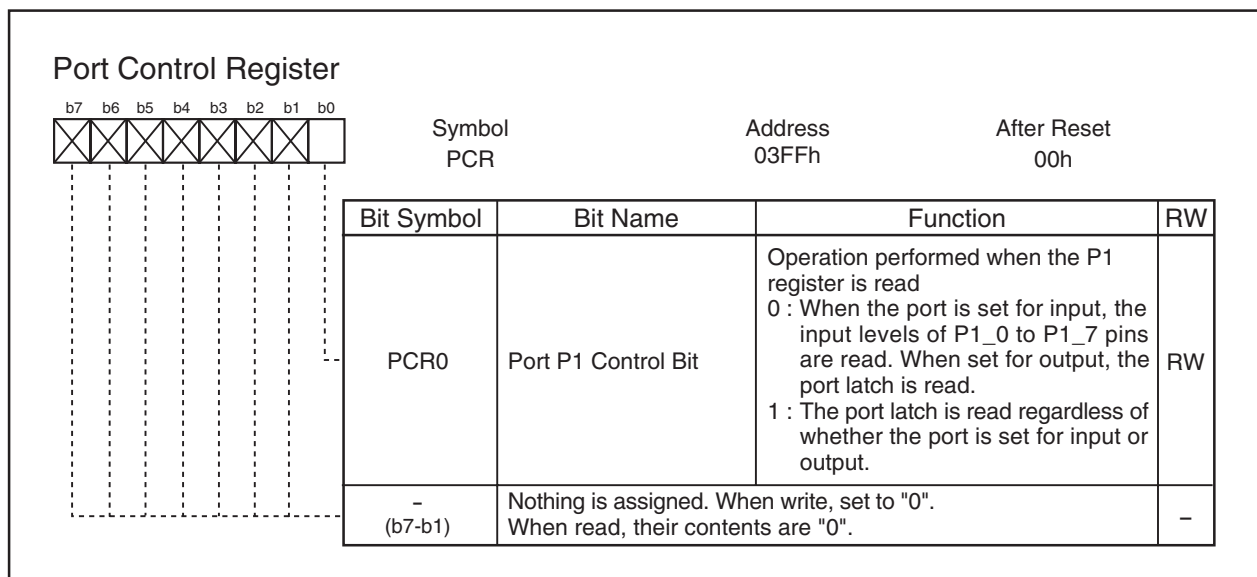
Symbol  
PUR2Address  
03FEhAfter Reset  
00h

Bit Symbol	Bit Name	Function	RW
PU20	P8_0 to P8_3 Pull-Up	0 : Not pulled high 1 : Pulled high <sup>(3)</sup>	RW
PU21	P8_4, P8_6 and P8_7 Pull-Up <sup>(1)</sup>		RW
PU22	P9_0, P9_2 and P9_3 Pull-Up <sup>(2)</sup>		RW
PU23	P9_4 to P9_7 Pull-Up		RW
PU24	P10_0 to P10_3 Pull-Up		RW
PU25	P10_4 to P10_7 Pull-Up		RW
- (b7-b6)	Nothing is assigned. When write, set to "0". When read, their contents are "0".		-

## NOTES:

- The P8\_5 pin does not have pull-up.
- The P9\_1 pin does not have pull-up.
- The pin for which this bit is "1" (pulled high) and the direction bit is "0" (input mode) is pulled high.

Figure 20.9 PUR0, PUR1 and PUR2 Registers



**Figure 20.10 PCR Register**

**Table 20.2 Unassigned Pin Handling in Single-chip Mode**

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

**NOTES:**

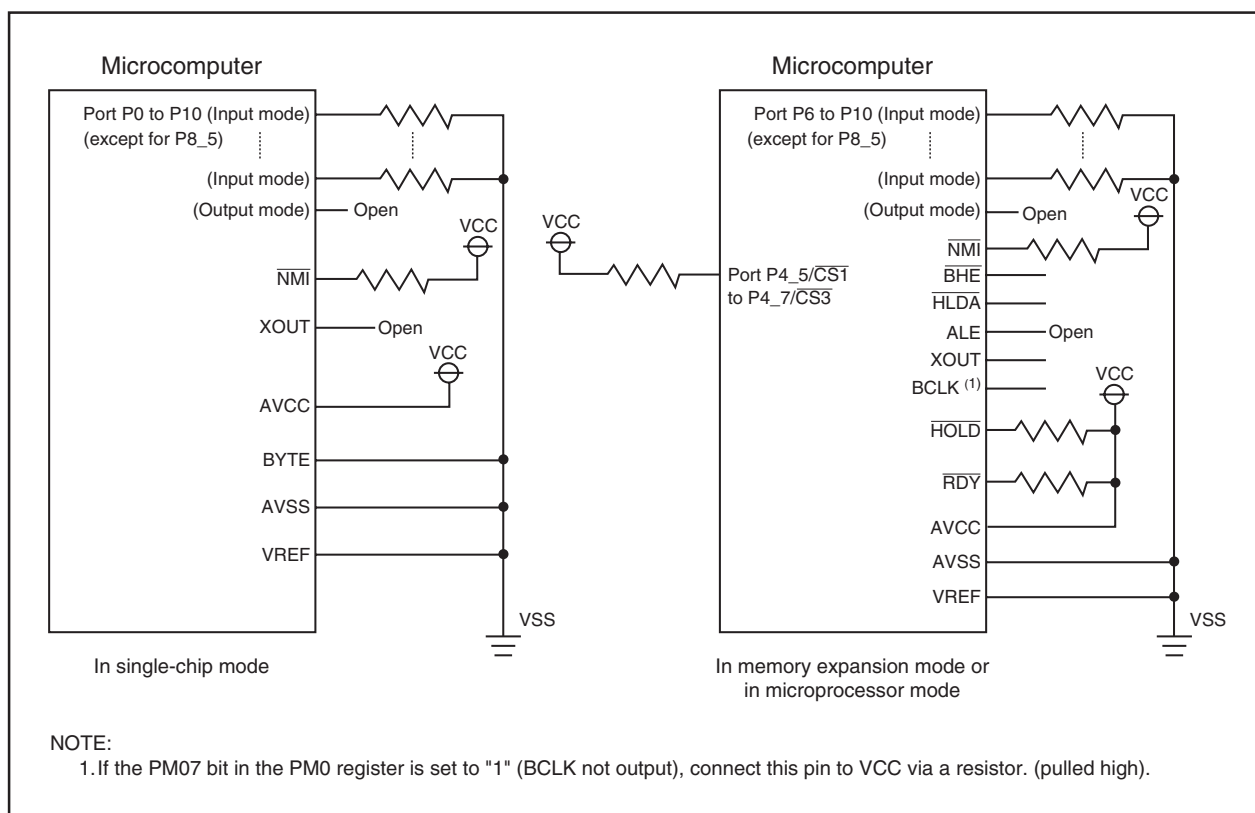
1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes indeterminate, causing the power supply current to increase while the port remains 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.
2. Make sure the unused pins are processed with the shortest possible wiring from the microcomputer pins (within 2 cm).
3. When the ports P7\_1 and P9\_1 are set for output mode, make sure a low-level signal is output from the pins. The ports P7\_1 and P9\_1 are N-channel open-drain outputs.
4. With external clock input to XIN pin.

**Table 20.3 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode**

Pin Name	Connection
Ports P0 to P7, P8_0 to P8_4, P8_6, P8_7, P9, P10	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open. <sup>(1)</sup> <sup>(2)</sup> <sup>(3)</sup> <sup>(4)</sup>
P4_5/CS1 to P4_7/CS3	Connect to VCC via a resistor (pulled high) by setting the PD4 register's corresponding direction bit for $\overline{CS}_i$ ( $i = 1$ to $3$ ) to "0" (input mode) and the $\overline{CS}_i$ bit in the CSR register to "0" (chip select disabled).
BHE, ALE, HLDA, XOUT <sup>(5)</sup> , BCLK <sup>(6)</sup>	Open
HOLD, RDY, NMI(P8_5)	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF	Connect to VSS

**NOTES:**

1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes 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.
2. Make sure the unused pins are processed with the shortest possible wiring from the microcomputer pins (within 2 cm).
3. If the CNVSS pin has the VSS level applied to it, these pins are set for input ports until the processor mode is switched over in a program after reset. For this reason, the voltage levels on these pins become indeterminate, causing the power supply current to increase while they remain set for input ports.
4. When the ports P7\_1 and P9\_1 are set for output mode, make sure a low-level signal is output from the pins. The ports P7\_1 and P9\_1 are N-channel open-drain outputs.
5. With external clock input to XIN pin.
6. If the PM07 bit in the PM0 register is set to "1" (BCLK not output), connect this pin to VCC via a resistor (pulled high).



**Figure 20.11 Unassigned Pins Handling**

## 21. Flash Memory Version

Aside from the built-in flash memory, the flash memory version microcomputer has the same functions as the masked ROM version.

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

Table 21.1 lists the specifications of the flash memory version. See **Table 1.1 Performance outline**, for the items not listed in Table 21.1). Table 21.2 shows the outline of flash memory rewrite mode.

**Table 21.1 Flash Memory Version Specifications**

Item		Specifications
Flash Memory Operating Mode		4 modes (CPU rewrite, standard serial I/O, parallel I/O, CAN I/O)
Erase Block	User ROM Area	See <b>Figure 21.1 Flash Memory Block Diagram</b>
	Boot ROM Area	1 block (4 Kbytes) <sup>(1)</sup>
Program Method		In units of word, in units of byte <sup>(2)</sup>
Erase Method		Collective erase, block erase
Program and Erase Control Method		Program and erase controlled by software command
Protect Method		Lock bit protects each block
Number of Commands		8 commands
Program and Erase Endurance <sup>(3)</sup>		100 times
ROM Code Protection		Parallel I/O , standard serial I/O and CAN I/O modes are supported.

**NOTES:**

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

**Table 21.2 Flash Memory Rewrite Modes Overview**

Flash Memory Rewrite Mode	CPU Rewrite Mode <sup>(1)</sup>	Standard Serial I/O Mode	Parallel I/O Mode	CAN I/O Mode
Function	The user ROM area is rewritten when the CPU executes software commands. EW0 mode: Rewrite in areas other than flash memory <sup>(2)</sup> EW1 mode: Can be rewritten in the flash memory	The user ROM area is rewritten using a dedicated serial programmer. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART <sup>(3)</sup>	The boot ROM and user ROM areas are rewritten using a dedicated parallel programmer.	The user ROM area is rewritten using a dedicated CAN programmer.
Areas which can be Rewritten	User ROM area	User ROM area	User ROM area Boot ROM area	User ROM area
Operation Mode	Single-chip mode Memory expansion mode (EW0 mode) Boot mode (EW0 mode)	Boot mode	Parallel I/O mode	Boot mode
ROM Programmer	None	Serial programmer	Parallel programmer	CAN programmer

**NOTES:**

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

## 21.1 Memory Map

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

Figure 21.1 shows the block diagram of flash memory.

The user ROM area is divided into several blocks, each of which can individually be protected (locked) against programming or erasure. The user ROM area can be rewritten in all of CPU rewrite, standard serial I/O mode, parallel I/O mode and CAN I/O mode. Block A is enabled for use by setting the PM10 bit in the PM1 register to "1" (block A enabled.  $\overline{CS}2$  area at addresses 10000h to 26FFFh).

The boot ROM area is located at the same addresses as the user ROM area. It can only be rewritten in parallel I/O mode (refer to **21.1.1 Boot Mode**). A program in the boot ROM area is executed after a hardware reset occurs while an "H" signal is applied to the CNVSS and P5\_0 pins and an "L" signal is applied to the P5\_5 pin (refer to **21.1.1 Boot Mode**). A program in the user ROM area is executed after a hardware reset occurs while an "L" signal is applied to the CNVSS pin. However, the boot ROM area cannot be read.

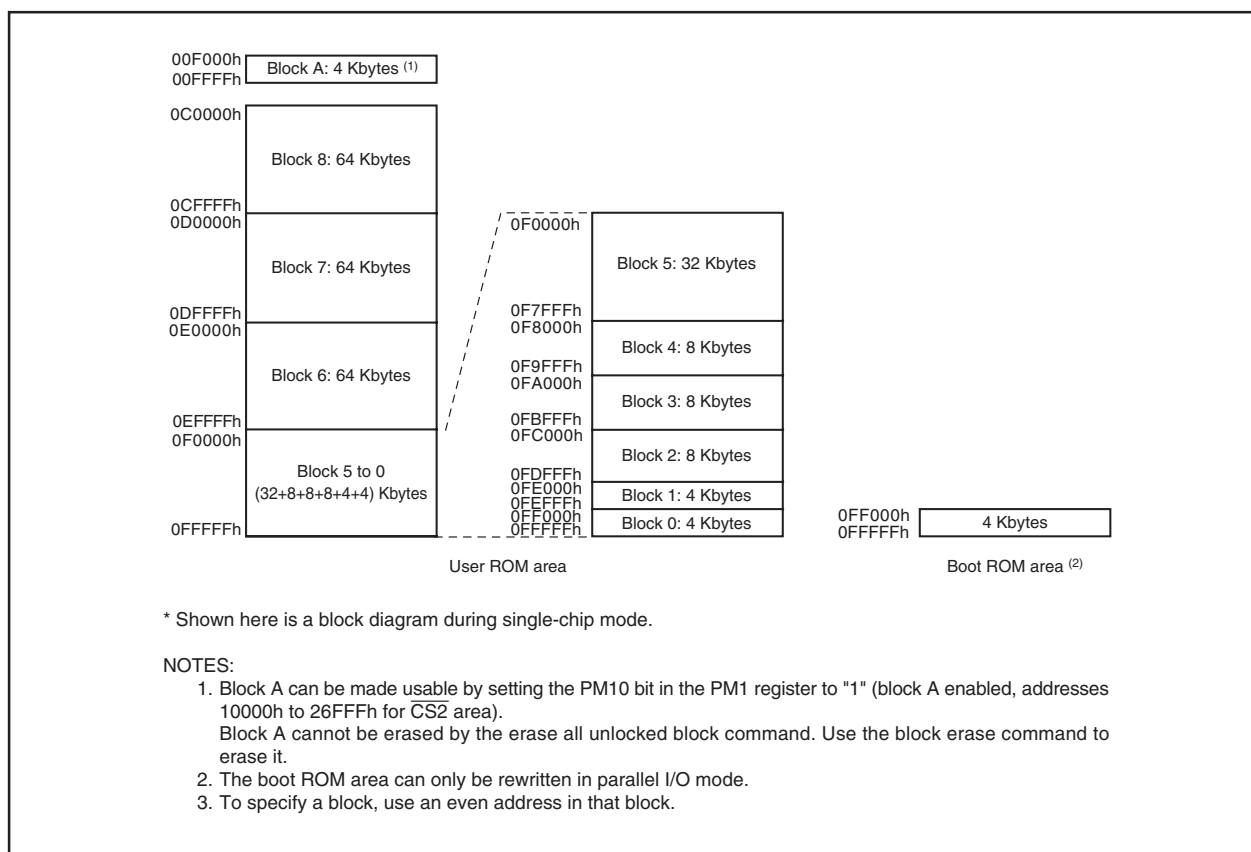


Figure 21.1 Flash Memory Block Diagram

### 21.1.1 Boot Mode

The microcomputer enters boot mode when a hardware reset occurs while an "H" signal is applied to the CNVSS and P5\_0 pins and an "L" signal is applied to the P5\_5 pin. A program in the boot ROM area is executed.

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

## 21.2 Functions to Prevent Flash Memory from Rewriting

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

### 21.2.1 ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten during parallel I/O mode. Figure 21.2 shows the ROMCP register. The ROMCP register is located in the user ROM area.

The ROM code protect function is enabled when the ROMCR bits are set to other than “11b”. In this case, set the bit 5 to bit 0 to “111111b”.

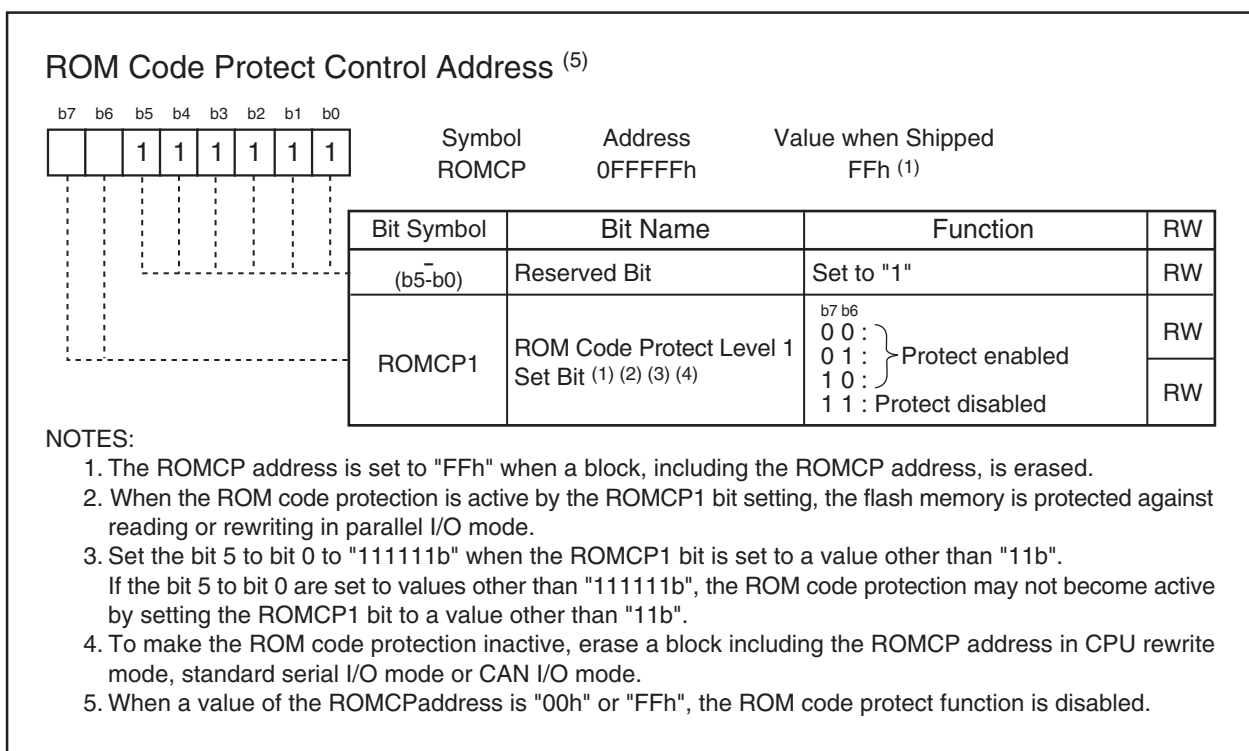
When exiting ROM code protect, erase the block including the ROMCP register by the CPU rewrite mode or the standard serial I/O mode or CAN I/O mode.

### 21.2.2 ID Code Check Function

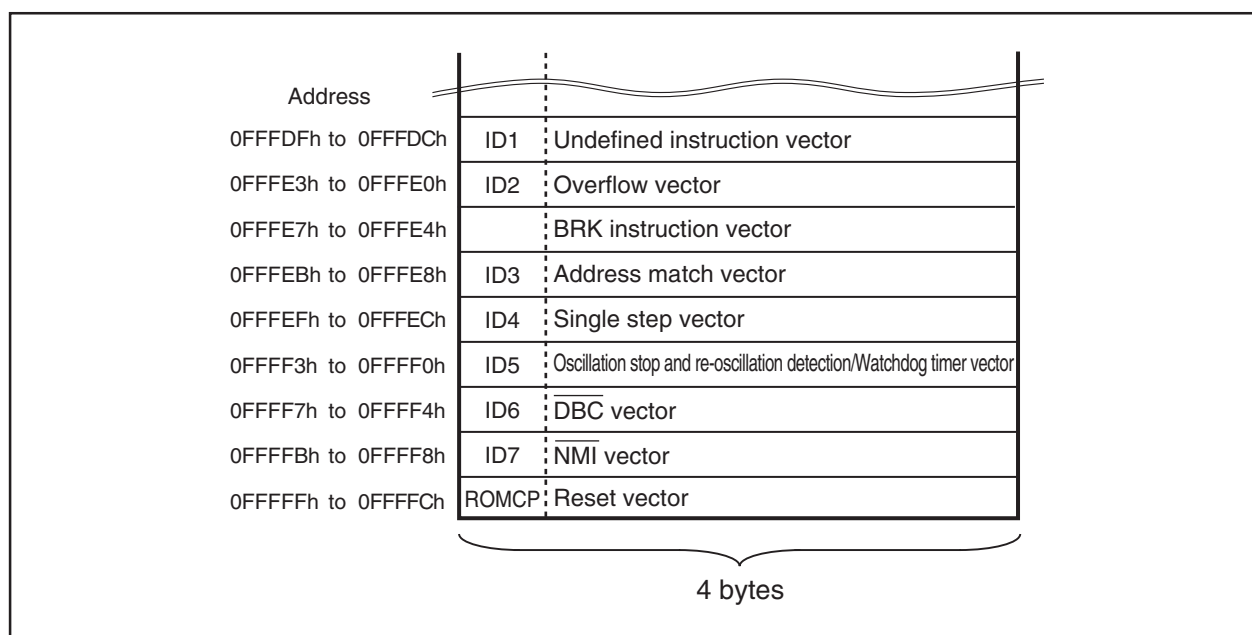
Use the ID code check function in standard serial I/O mode and CAN I/O mode. The ID code sent from the serial programmer is compared with the ID code written in the flash memory for a match. If the ID codes do not match, commands sent from the serial programmer are not accepted. However, if the four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

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

Figure 21.3 shows the ID code store addresses.



**Figure 21.2 ROMCP Register**



**Figure 21.3 Address for ID Code Stored**



### 21.3 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten when the CPU executes software commands. The user ROM area can be rewritten with the microcomputer is mounted on a board without using a parallel, serial or CAN programmer.

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

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

Table 21.3 lists the differences between EW0 and EW1 modes.

**Table 21.3 EW0 Mode and EW1 Mode**

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

**NOTES:**

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

### 21.3.1 EW0 Mode

The microcomputer enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to “1” (CPU rewrite mode enabled) and is ready to accept commands. EW0 mode is selected by setting the FMR11 bit in the FMR1 register to “0”. To set the FMR01 bit to “1”, set to “1” after first writing “0”.

The software commands control programming and erasing. The FMR0 register or the status register indicates whether a program or erase operation is completed as expected or not.

### 21.3.2 EW1 Mode

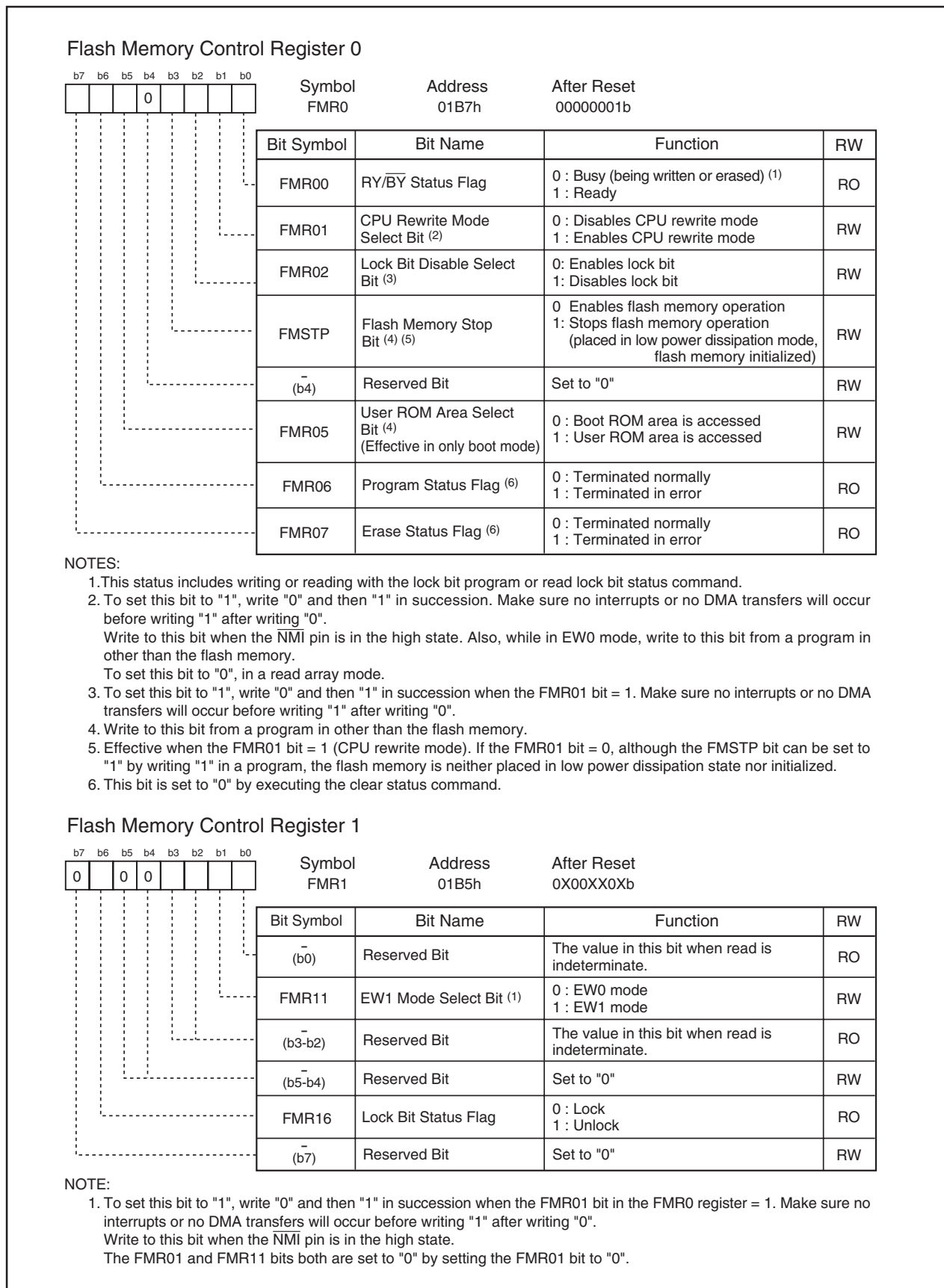
EW1 mode is selected by setting FMR11 bit to “1” (by writing “0” and then “1” in succession) after setting the FMR01 bit to “1” (by writing “0” and then “1” in succession). (Both bits must be set to “0” first before setting to “1”.)

The FMR0 register indicates whether or not a program or erase operation has been completed as expected. The status register cannot be read in EW1 mode.

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

### 21.3.3 FMR0, FMR1 Registers

Figure 21.4 shows FMR0 and FMR1 registers.



**Figure 21.4 FMR0 Register and FMR1 Register**

### 21.3.3.1 FMR00 Bit

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

### 21.3.3.2 FMR01 Bit

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

### 21.3.3.3 FMR02 Bit

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

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

### 21.3.3.4 FMSTP Bit

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 program in a space other than the flash memory.

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

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

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

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

Figure 21.7 shows a flow chart illustrating how to start and stop the flash memory processing before and after low power dissipation mode or on-chip oscillator low power dissipation mode. Follow the procedure on this flow chart.

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

### 21.3.3.5 FMR05 Bit

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

### 21.3.3.6 FMR06 Bit

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

**21.3.3.7 FMR07 Bit**

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

**21.3.3.8 FMR11 Bit**

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

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

**21.3.3.9 FMR16 Bit**

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

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

Figure 21.5 shows setting and resetting of EW0 mode. Figure 21.6 show setting and resetting of EW1 mode.

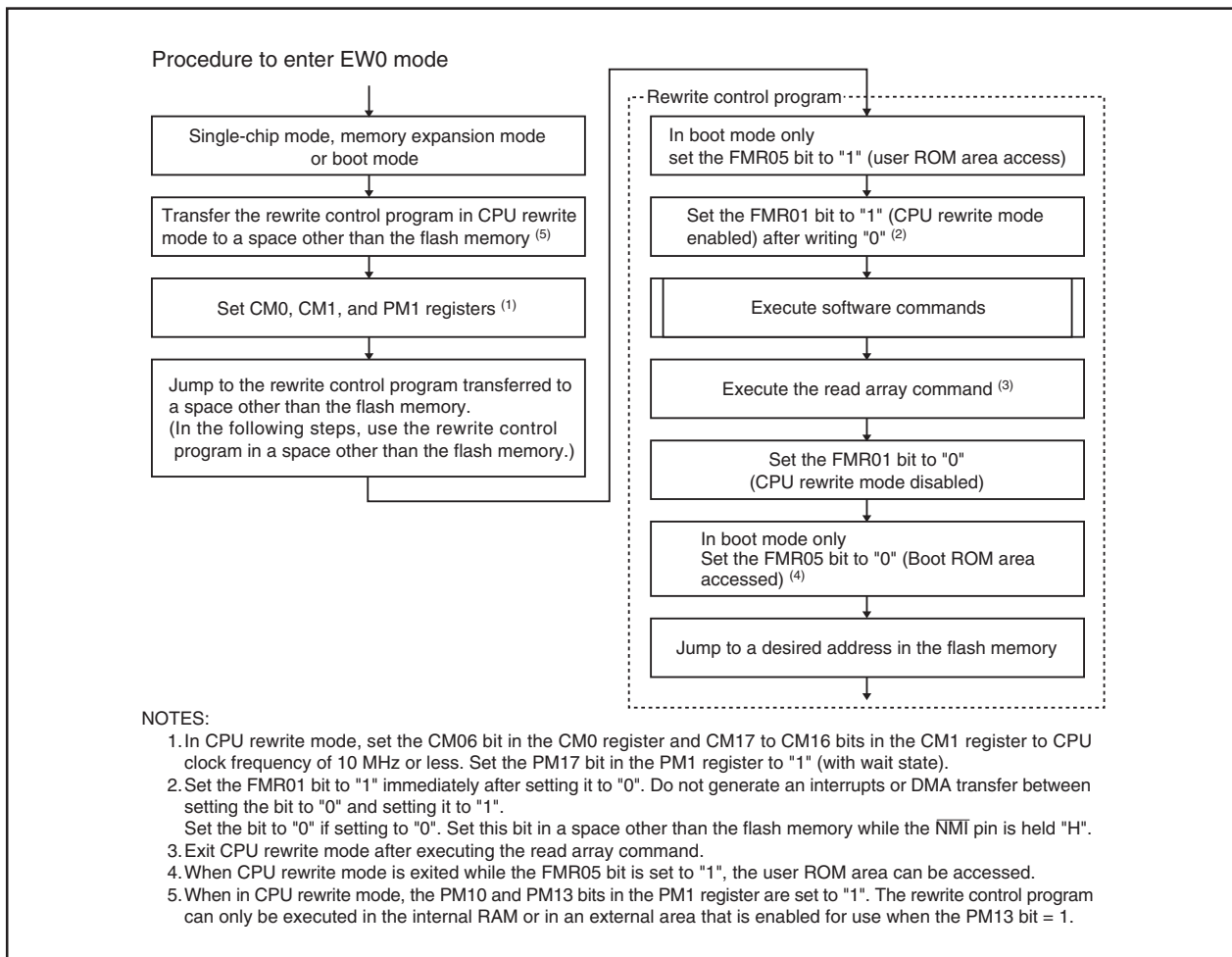


Figure 21.5 Setting and Resetting of EW0 Mode

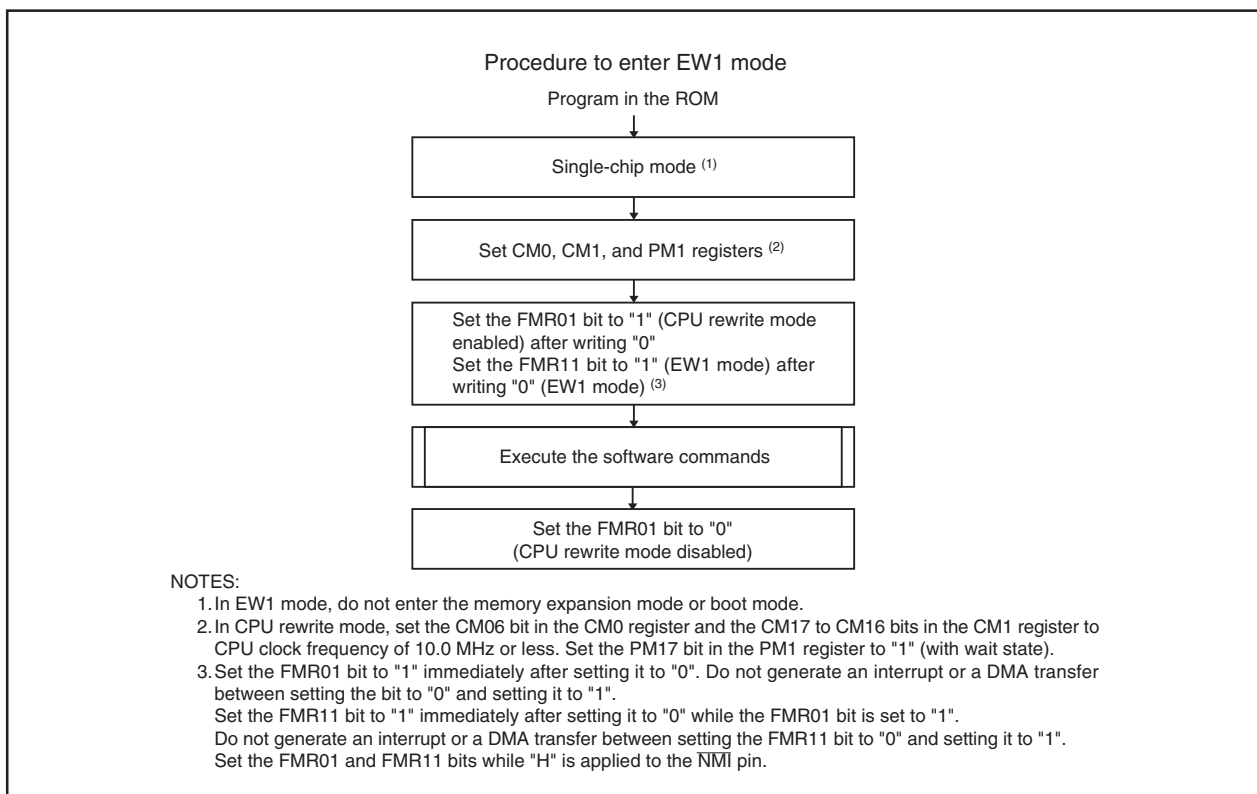
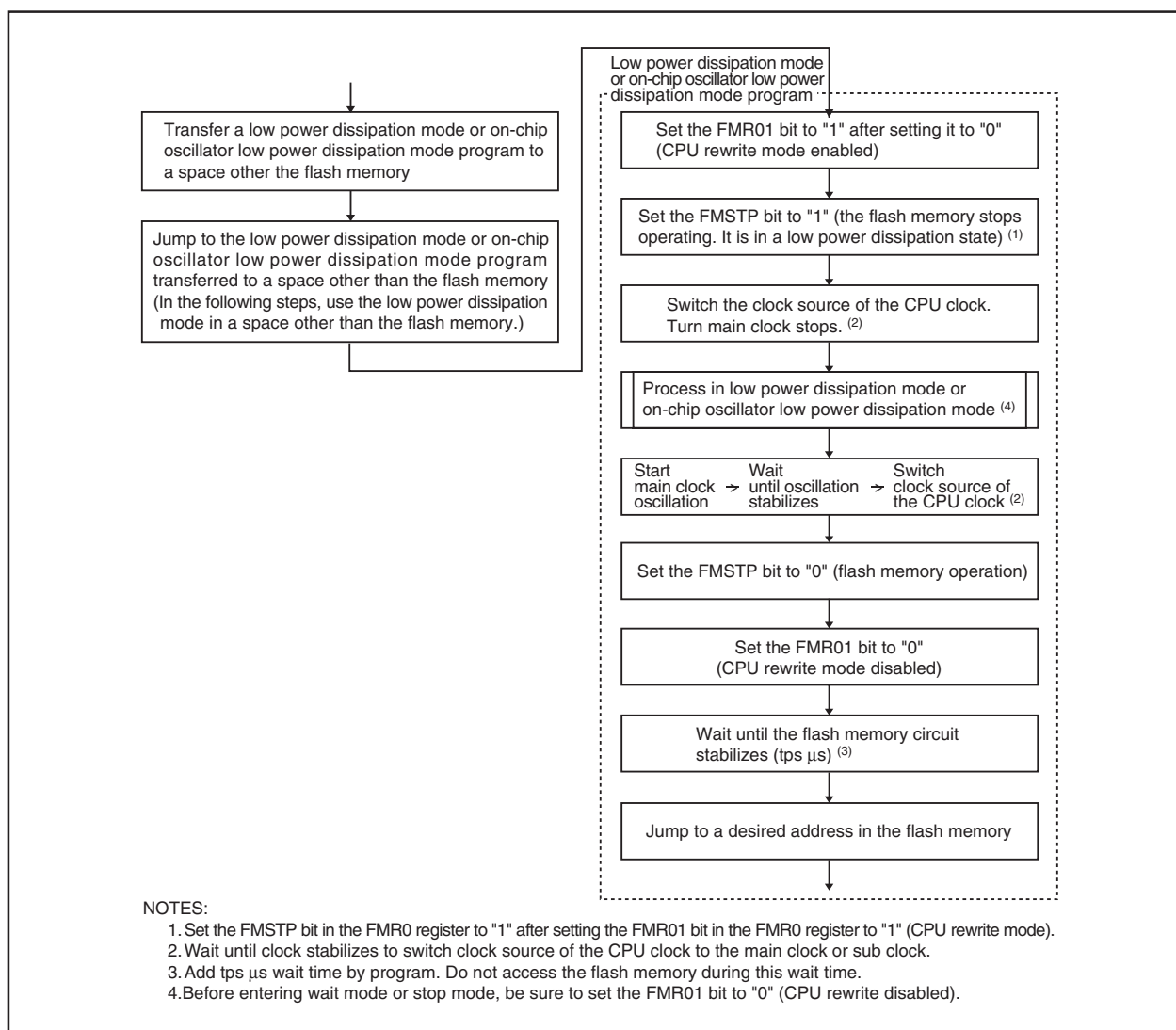


Figure 21.6 Setting and Resetting of EW1 Mode



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

## 21.3.4 Precautions on CPU Rewrite Mode

### 21.3.4.1 Operating Speed

Set the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register to clock frequency of 10 MHz or less before entering CPU rewrite mode (EW0 or EW1 mode). Also, set the PM17 bit in the PM1 register to "1" (with wait state).

### 21.3.4.2 Prohibited Instructions

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

### 21.3.4.3 Interrupts (EW0 Mode)

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.
- The NMI and watchdog timer interrupts are available since the FMR0 and FMR1 registers are forcibly reset when either interrupt request is generated. Allocate the jump addresses for each interrupt service routines to the fixed vector table. Flash memory rewrite operation is aborted when the NMI or watchdog timer interrupt request is generated. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

### 21.3.4.4 Interrupts (EW1 Mode)

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

### 21.3.4.5 How to Access

To set the FMR01, FMR02 or FMR11 bit to "1", write "1" after first setting the bit to "0". Do not generate an interrupt or a DMA transfer between the instruction to set the bit to "0" and the instruction to set the bit to "1". Set the bit while an "H" signal is applied to the NMI pin.

### 21.3.4.6 Rewriting in User ROM Area (EW0 Mode)

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

### 21.3.4.7 Rewriting in User ROM Area (EW1 Mode)

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

### 21.3.4.8 DMA Transfer

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



**21.3.4.9 Writing Command and Data**

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

**21.3.4.10 Wait Mode**

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

**21.3.4.11 Stop Mode**

When 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.

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

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

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

### 21.3.5 Software Commands

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

Table 21.4 lists the software commands.

**Table 21.4 Software Commands**

Software Command	First Bus Cycle			Second Bus Cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read Array	Write	×	xxFFh	-	-	-
Read Status Register	Write	×	xx70h	Read	×	SRD
Clear Status Register	Write	×	xx50h	-	-	-
Program	Write	WA	xx40h	Write	WA	WD
Block Erase	Write	×	xx20h	Write	BA	xxD0h
Erase All Unlocked Block <sup>(1)</sup>	Write	×	xxA7h	Write	×	xxD0h
Lock Bit Program	Write	BA	xx77h	Write	BA	xxD0h
Read Lock Bit Status	Write	×	xx71h	Write	BA	xxD0h

SRD: data in SRD register (D7 to D0)

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

WD: 16-bit write data

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

×: Any even address in the user ROM area

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

**NOTE**

1. It is only blocks 0 to 8 that can be erased by the erase all unlocked block command.  
Block A cannot be erased. The block erase command must be used to erase the block A.

#### 21.3.5.1 Read Array Command (FFh)

The read array command reads the flash memory.

By writing command code “xxFFh” in the first bus cycle, read array mode is entered. Content of a specified address can be read in 16-bit unit after the next bus cycle.

The microcomputer remains in read array mode until another command is written. Therefore, contents from multiple addresses can be read consecutively.

#### 21.3.5.2 Read Status Register Command (70h)

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

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

Do not execute this command in EW1 mode.

#### 21.3.5.3 Clear Status Register Command (50h)

The clear status register command clears the status register.

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

#### 21.3.5.4 Program Command (40h)

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

By writing "xx40h" in the first bus cycle and data to the write address in the second bus cycle, an auto program operation (data program and verify) will start. The address value specified in the first bus cycle must be the same even address as the write address specified in the second bus cycle.

The FMR00 bit in the FMR0 register indicates whether an auto program operation has been completed. The FMR00 bit is set to "0" (busy) during auto program and to "1" (ready) when an auto program operation is completed.

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

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

Figure 21.8 shows a flow chart of the program command programming.

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

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

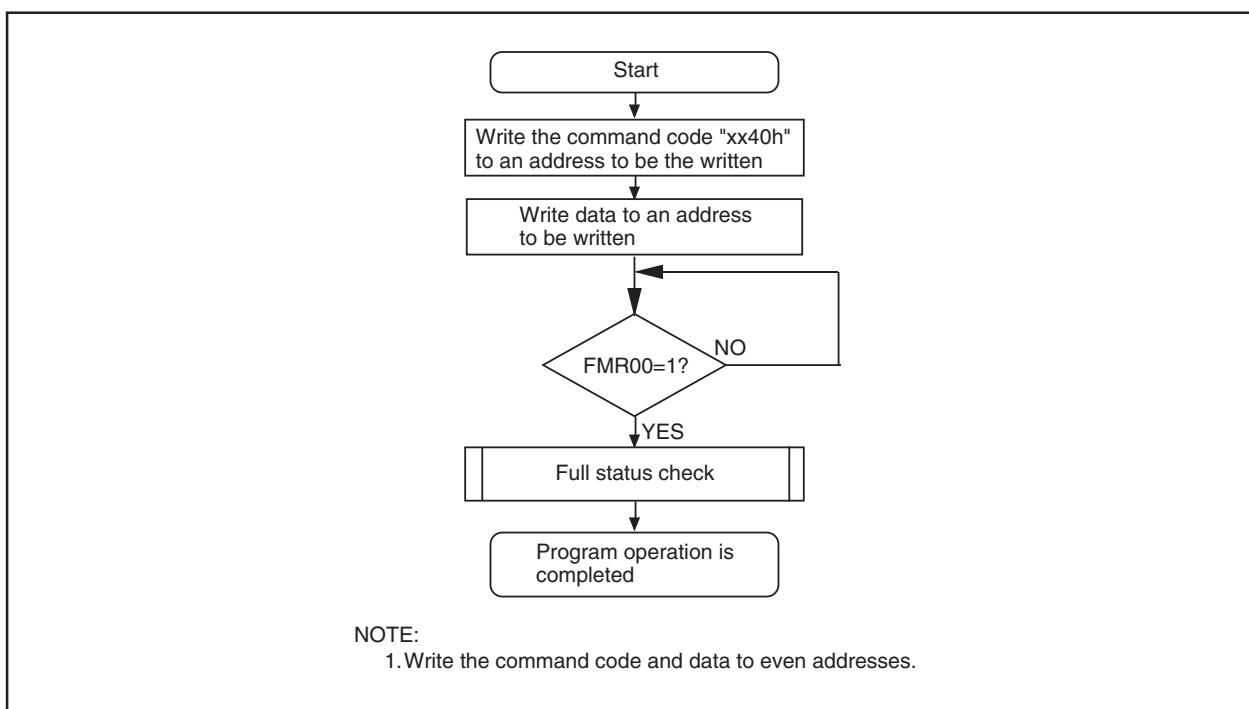


Figure 21.8 Program Command

### 21.3.5.5 Block Erase Command

The block erase command erases each block.

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

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

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

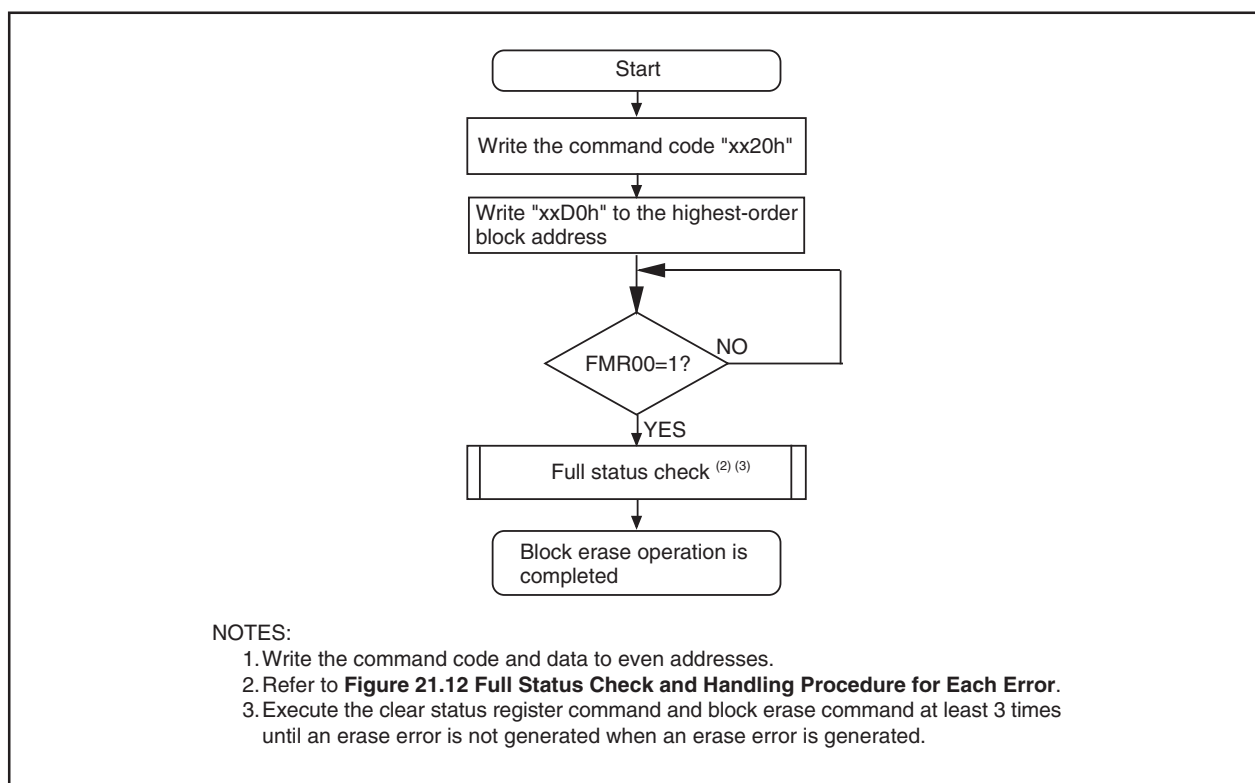
After the completion of an auto erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto erase operation has been completed as expected. (Refer to **21.3.8 Full Status Check**.)

Figure 21.9 shows a flow chart of the block erase command programming.

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

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

In EW0 mode, the microcomputer enters read status register mode as soon as an auto erase operation starts. The status register can be read. The SR7 bit in the status register is set to “0” at the same time an auto erase operation starts. It is set to “1” when an auto erase operation is completed. The microcomputer remains in read status register mode until the read array command or read lock bit status command is written. Also execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.



**Figure 21.9 Block Erase Command**

### 21.3.5.6 Erase All Unlocked Block

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

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

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

After the completion of an auto erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto erase operation has been completed as expected.

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

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

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

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

### 21.3.5.7 Lock Bit Program Command

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

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

Figure 21.10 shows a flow chart of the lock bit program command programming. Execute read lock bit status command to read lock bit state (lock bit data).

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

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

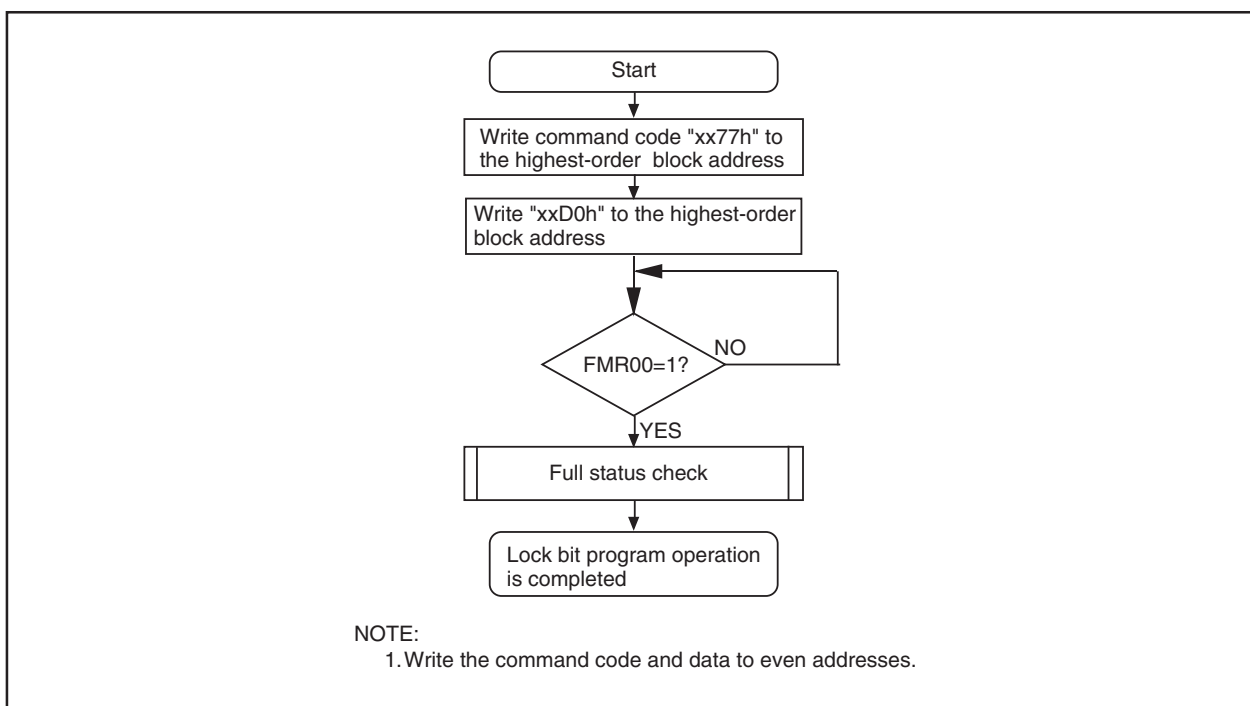


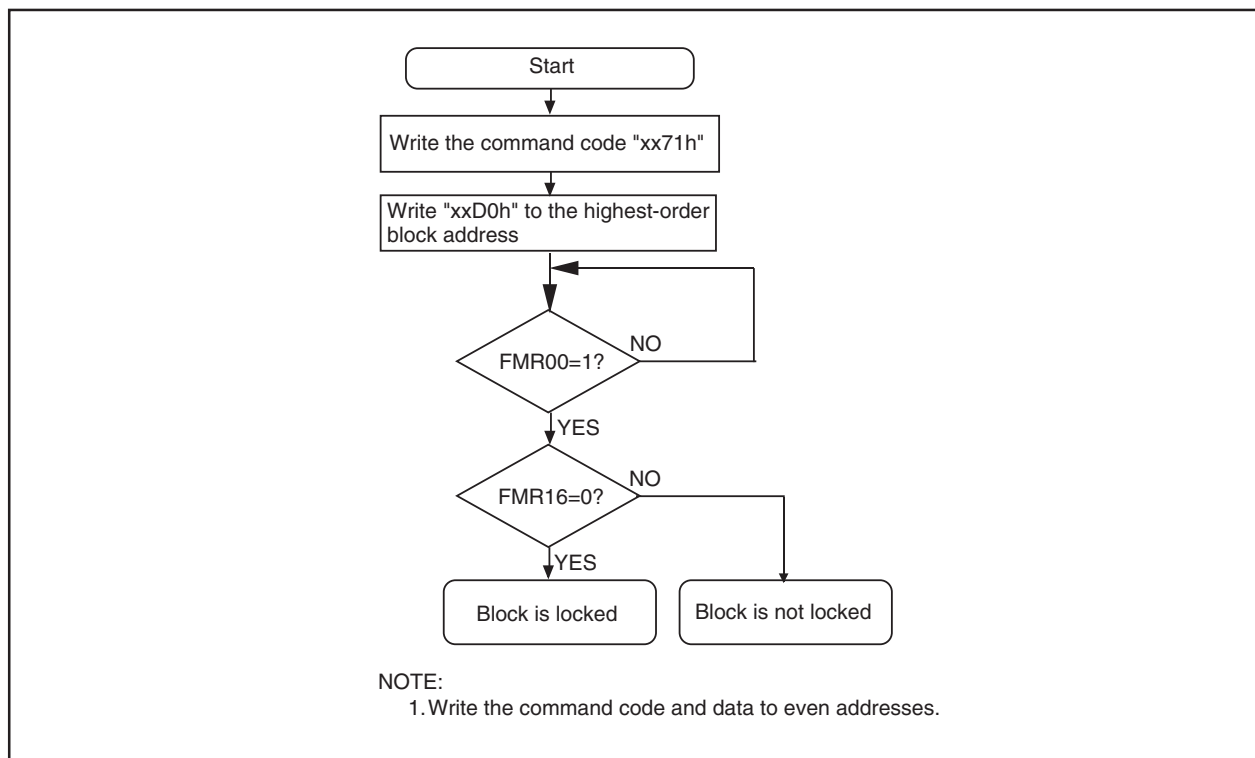
Figure 21.10 Lock Bit Program Command

### 21.3.5.8 Read Lock Bit Status Command (71h)

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

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

Figure 21.11 shows a flow chart of the read lock bit status command programming.



**Figure 21.11 Read Lock Bit Status Command**

### 21.3.6 Data Protect Function

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

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

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

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

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

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

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

### 21.3.7 Status Register (SRD Register)

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

Table 21.5 shows the status register.

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

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

#### 21.3.7.1 Sequencer Status (SR7 and FMR00 Bits)

The sequence status indicates the flash memory operation state. It is set to “0” while the program, block erase, erase all unlocked block, lock bit program, or read lock bit status command is being executed; otherwise, it is set to “1”.

#### 21.3.7.2 Erase Status (SR5 and FMR07 Bits)

Refer to **21.3.8 Full Status Check**.

#### 21.3.7.3 Program Status (SR4 and FMR06 Bits)

Refer to **21.3.8 Full Status Check**.

**Table 21.5 Status Register**

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

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

**NOTE:**

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



### 21.3.8 Full Status Check

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

Table 21.6 lists errors and FMR0 register state. Figure 21.12 shows a flow chart of the full status check and handling procedure for each error.

**Table 21.6 Errors and FMR0 Register Status**

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

**NOTES:**

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

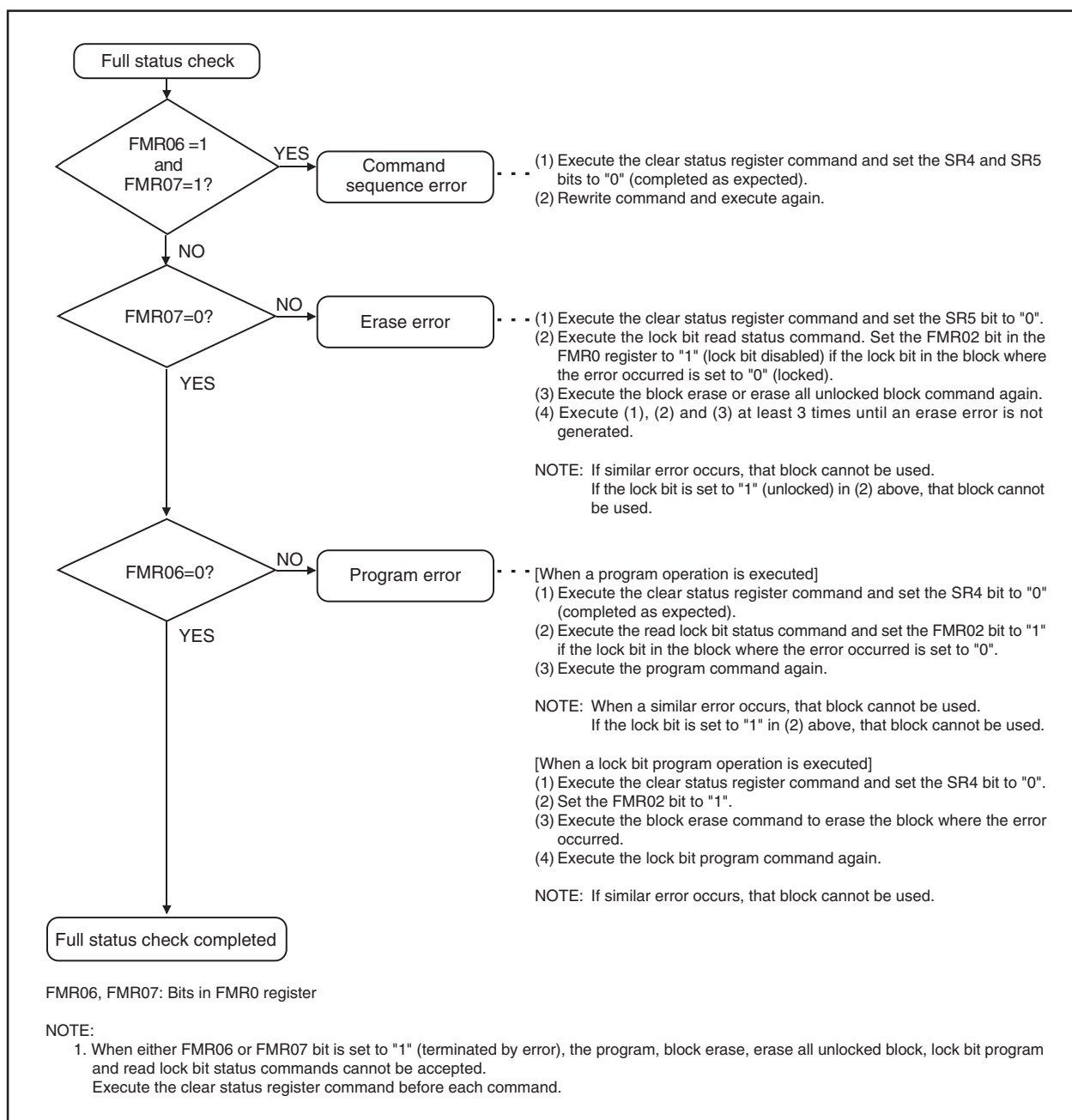


Figure 21.12 Full Status Check and Handling Procedure for Each Error

## 21.4 Standard Serial I/O Mode

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

Table 21.7 lists pin functions for standard serial I/O mode. Figures 21.13 and 21.14 show pin connections for standard serial I/O mode.

### 21.4.1 ID Code Check Function

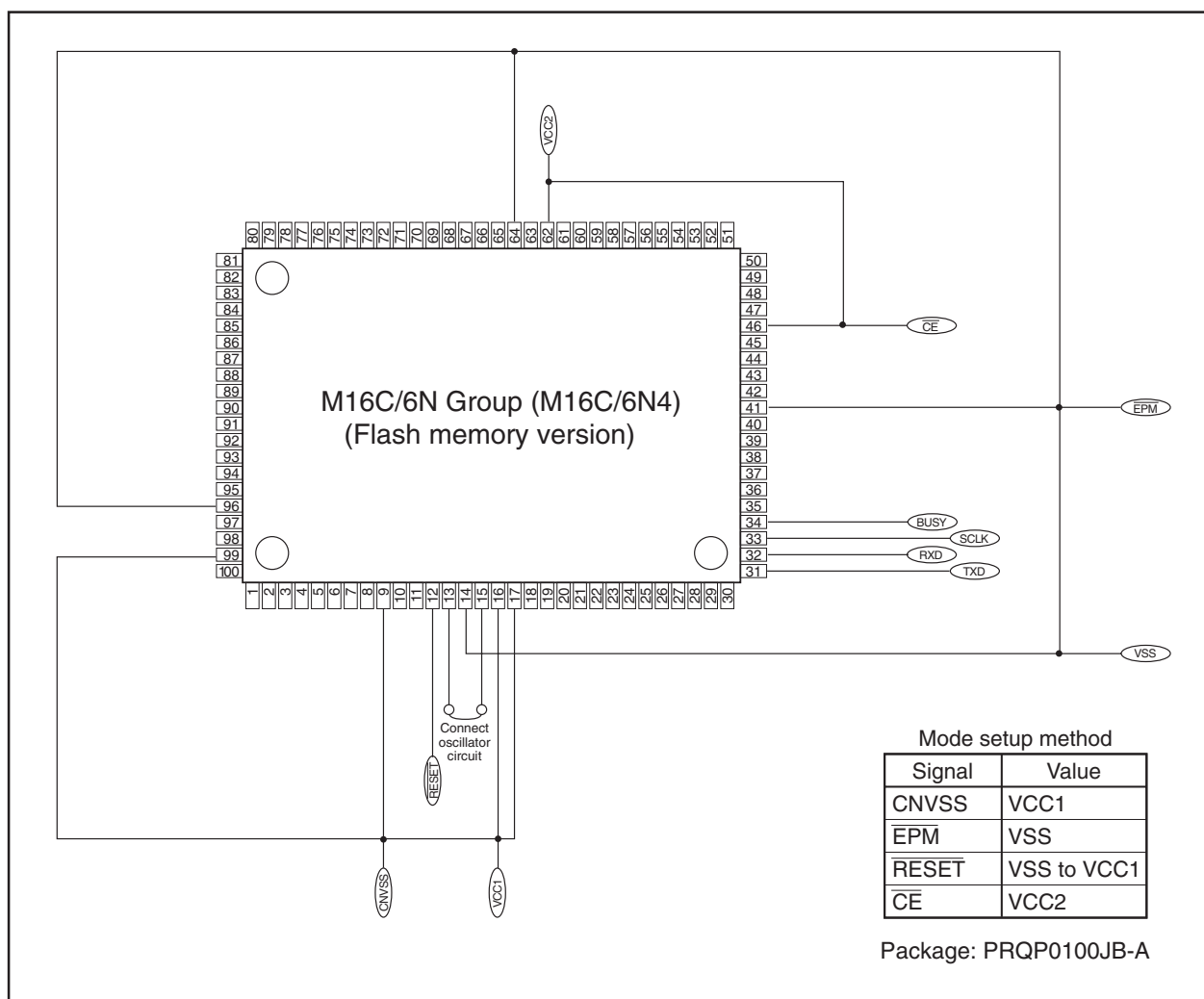
The ID code check function determines whether the ID codes sent from the serial programmer matches those written in the flash memory. (Refer to **21.2 Functions to Prevent Flash Memory from Rewriting.**)

**Table 21.7 Pin Functions for Standard Serial I/O Mode**

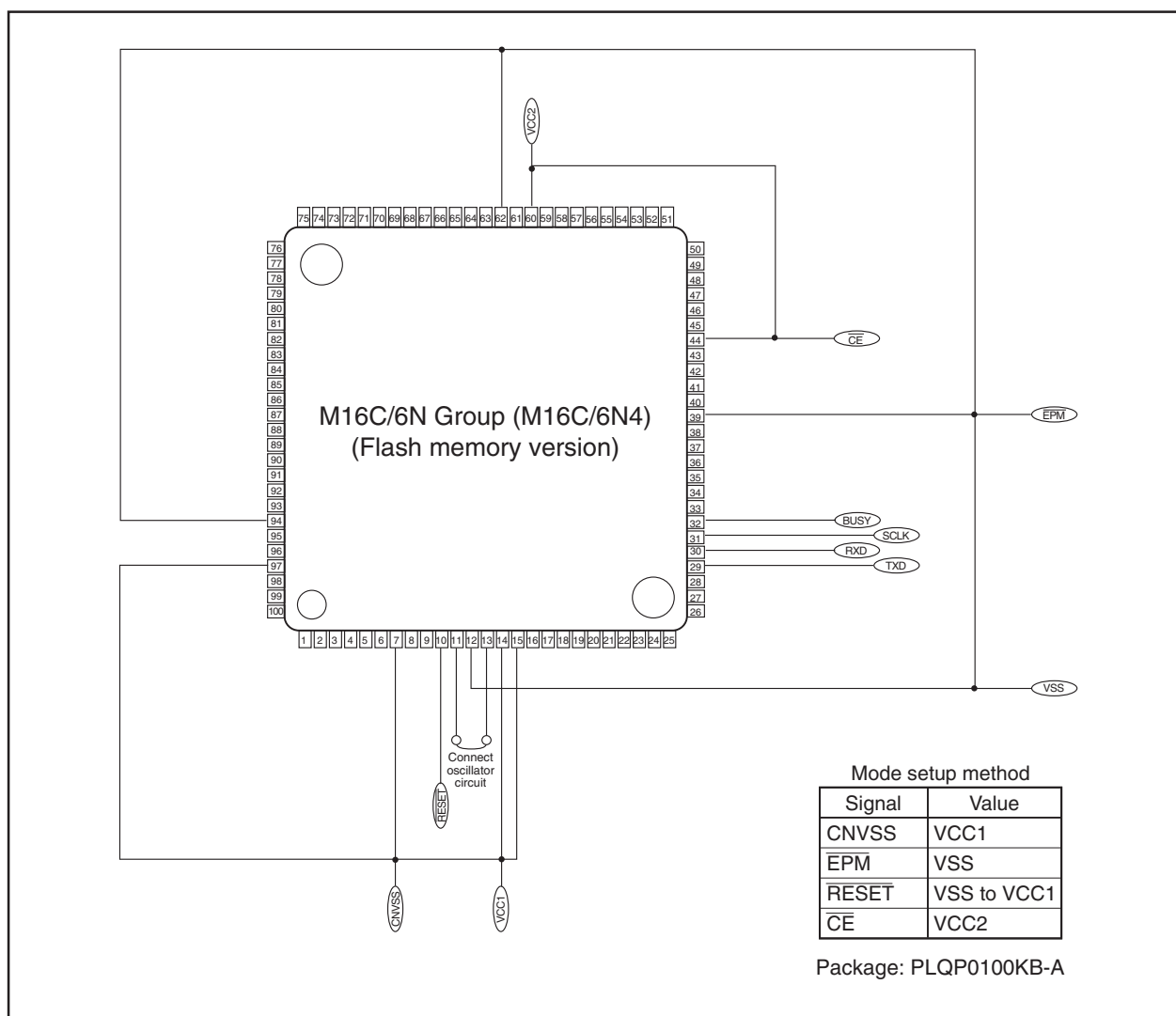
Pin	Name	I/O	Description
VCC1, VCC2, VSS	Power supply input		Apply the Flash Program, Erase Voltage to VCC1 pin and VCC2 to VCC2 pin. The VCC apply condition is that VCC2 = VCC1. Apply 0 V to VSS pin.
CNVSS	CNVSS	I	Connect to VCC1 pin.
RESET	Reset input	I	Reset input pin. While RESET pin is "L" level,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

**NOTES:**

1. When using the standard serial I/O mode, It is necessary to input "H" to the TXD1(P6\_7) pin while the RESET pin is "L". Therefore, the internal pull-up is enabled for the TXD1(P6\_7) pin while the RESET pin is "L".
2. When using the standard serial I/O mode, the P0\_0 to P0\_7, P1\_0 to P1\_7 pins may become indeterminate while the P8\_4 pin is "H" and the RESET pin is "L". If this causes a problem, apply "L" to the P8\_4 pin.



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



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

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

Figures 21.15 and 21.16 show example of circuit application in standard serial I/O mode 1 and mode 2, respectively. Refer to the user's manual of your serial programmer to handle pins controlled by a serial programmer.

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

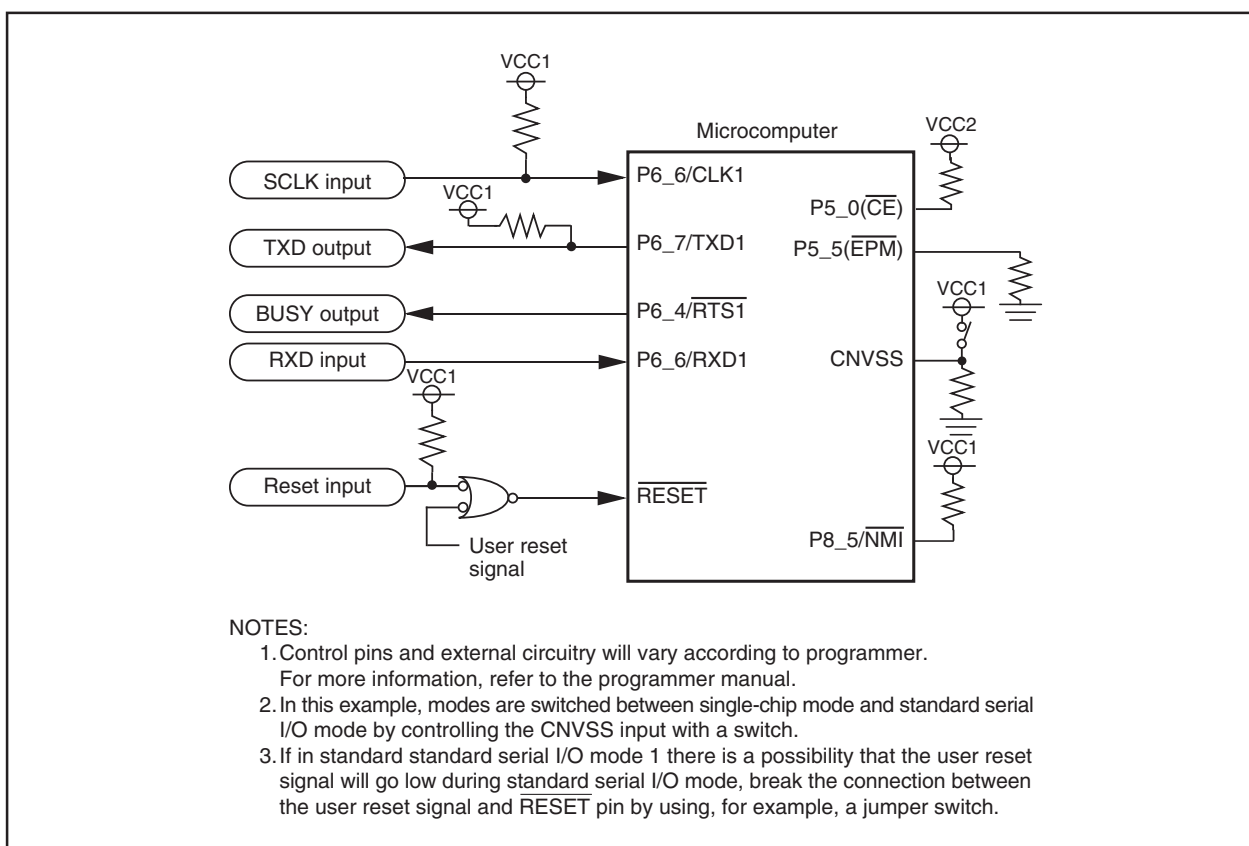


Figure 21.15 Circuit Application in Standard Serial I/O Mode 1

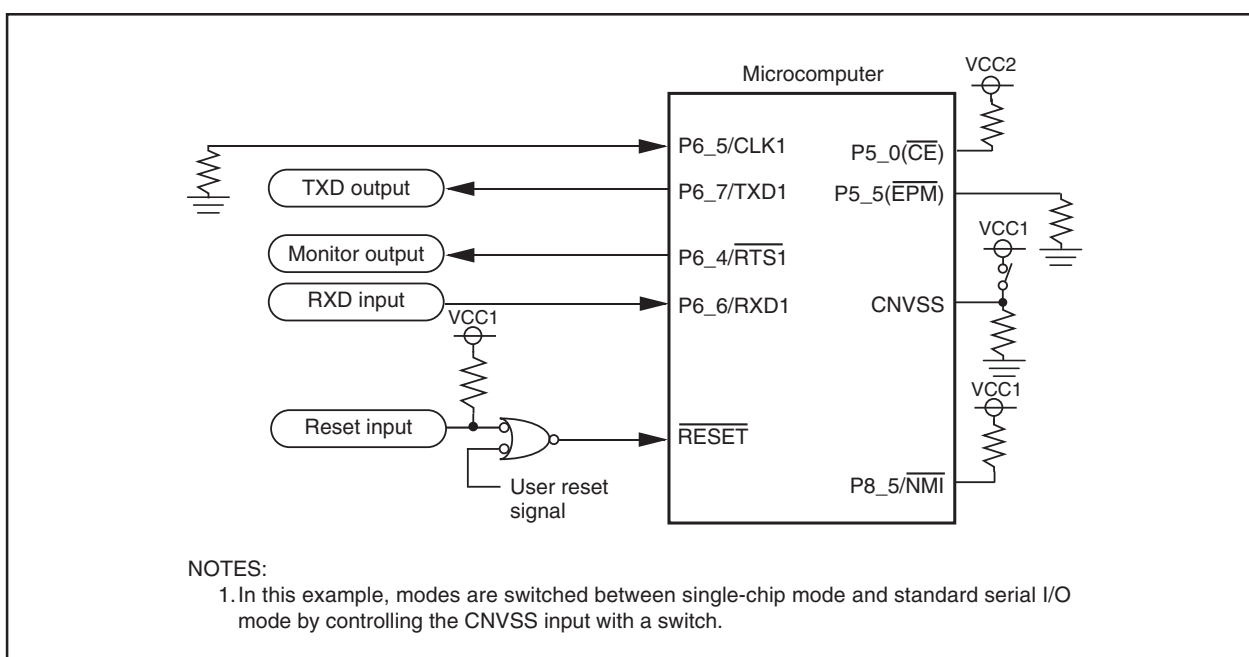


Figure 21.16 Circuit Application in Standard Serial I/O Mode 2

## 21.5 Parallel I/O Mode

In parallel I/O mode, the user ROM area and the boot ROM area can be rewritten by a parallel programmer supporting the M16C/6N Group (M16C/6N4). Contact your parallel programmer manufacturer for more information on the parallel programmer. Refer to the user's manual included with your parallel programmer for instructions.

### 21.5.1 User ROM and Boot ROM Areas

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

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

### 21.5.2 ROM Code Protect Function

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



## 21.6 CAN I/O Mode

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

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

### 21.6.1 ID Code Check Function

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

**Table 21.8 Pin Functions for CAN I/O Mode**

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

NOTE:

1. When using CAN I/O mode, the P0\_0 to P0\_7, P1\_0 to P1\_7 pins may become indeterminate while the P8\_4 pin is "H" and the RESET pin is "L". If this causes a problem, apply "L" to the P8\_4 pin.

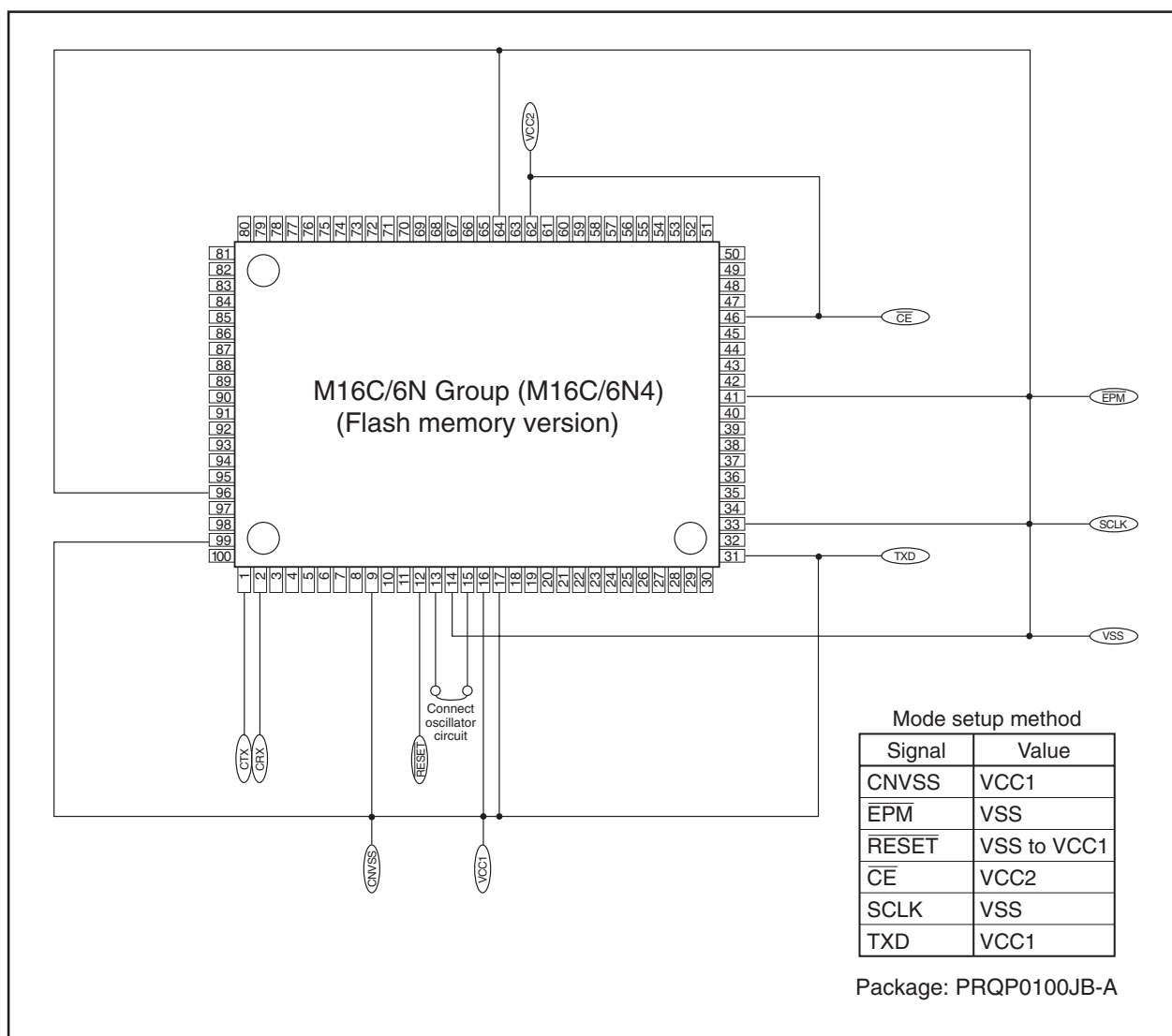


Figure 21.17 Pin Connections for CAN I/O Mode (1)

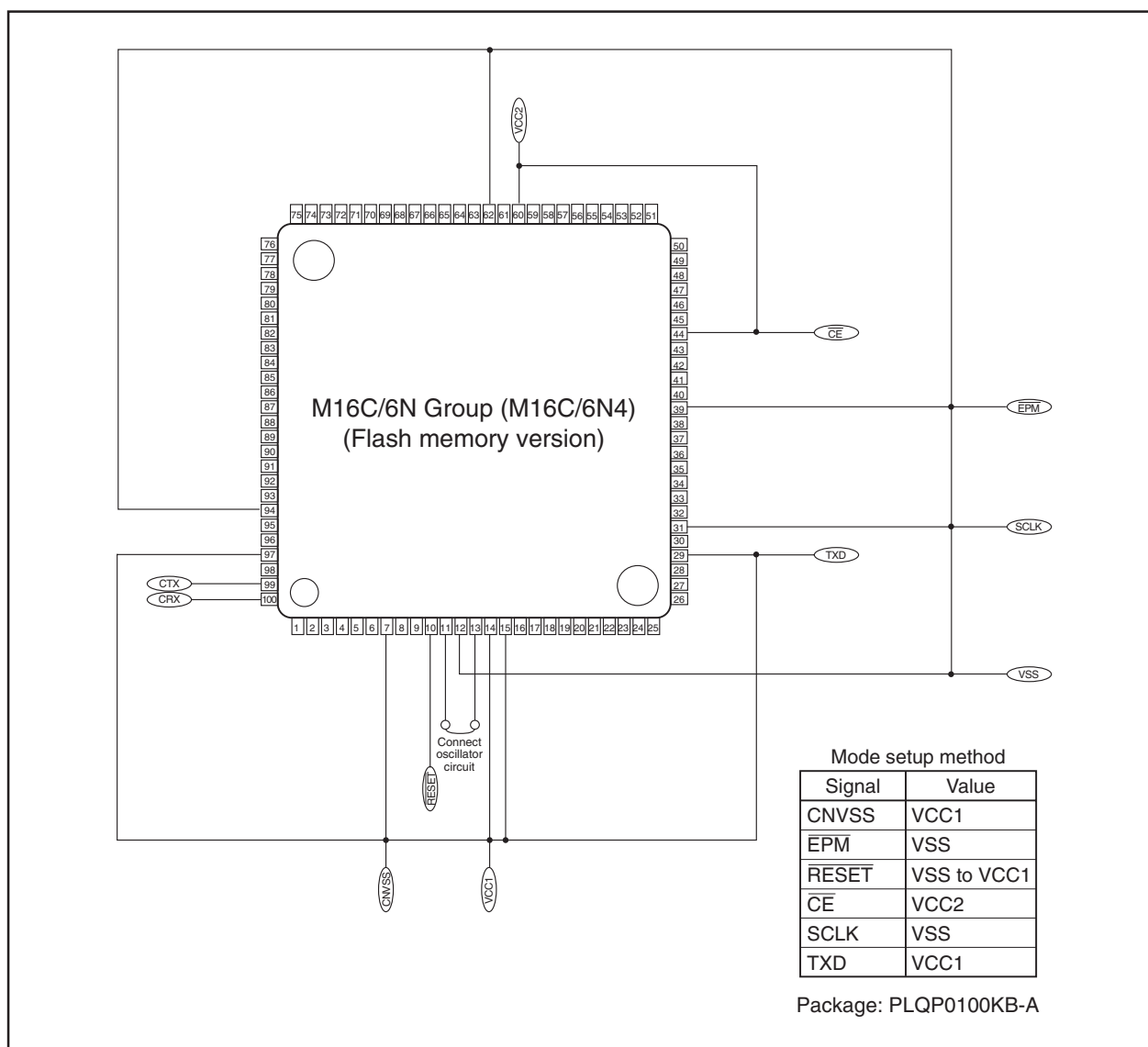
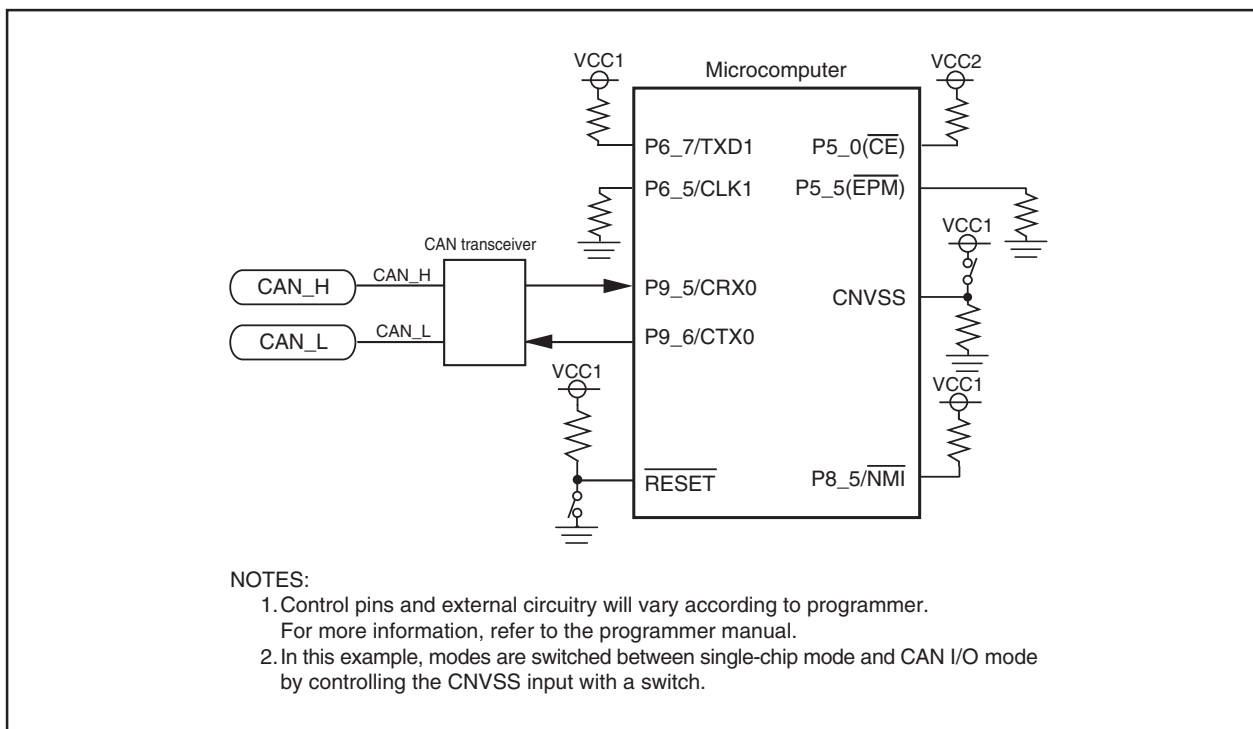


Figure 21.18 Pin Connections for CAN I/O Mode (2)

### 21.6.2 Example of Circuit Application in CAN I/O Mode

Figure 21.19 shows example of circuit application in CAN I/O mode. Refer to the user's manual of your CAN programmer to handle pins controlled by a CAN programmer.



**Figure 21.19 Circuit Application in CAN I/O Mode**

## 21.7 Electrical Characteristics

### 21.7.1 Electrical Characteristics (T/V-ver.)

Table 21.9 lists the flash memory electrical characteristics. Table 21.10 lists the flash memory version program/erase voltage and read operation voltage characteristics.

**Table 21.9 Flash Memory Version Electrical Characteristics <sup>(1)</sup>**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
-	Program and Erase Endurance <sup>(2)</sup>		100			cycle
-	Word Program Time (VCC = 5.0V)			25	200	μs
-	Lock Bit Program Time			25	200	μs
-	Block Erase Time (VCC = 5.0V)	4-Kbyte block		0.3	4	s
		8-Kbyte block		0.3	4	s
		32-Kbyte block		0.5	4	s
		64-Kbyte block		0.8	4	s
-	Erase All Unlocked Blocks Time				4 × n <sup>(3)</sup>	s
tps	Flash Memory Circuit Stabilization Wait Time				15	μs

**NOTES:**

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

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

Flash Program, Erase Voltage	Flash Read Operation Voltage
VCC = 5.0 ± 0.5V	VCC = 4.2 to 5.5V

### 21.7.2 Electrical Characteristics (Normal-ver.)

Table 21.11 lists the flash memory electrical characteristics. Table 21.12 lists the flash memory version program/erase voltage and read operation voltage characteristics.

**Table 21.11 Flash Memory Version Electrical Characteristics <sup>(1)</sup>**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
-	Program and Erase Endurance <sup>(2)</sup>		100			cycle
-	Word Program Time (VCC = 5.0V)			25	200	μs
-	Lock Bit Program Time			25	200	μs
-	Block Erase Time (VCC = 5.0V)	4-Kbyte block		0.3	4	s
		8-Kbyte block		0.3	4	s
		32-Kbyte block		0.5	4	s
		64-Kbyte block		0.8	4	s
-	Erase All Unlocked Blocks Time				4 × n <sup>(3)</sup>	s
tps	Flash Memory Circuit Stabilization Wait Time				15	μs

**NOTES:**

1. Referenced to VCC = 4.5 to 5.5V, 3.0 to 3.6V, Topr = 0 to 60°C unless otherwise specified.
2. Program and Erase Endurance refers to the number of times a block erase can be performed.  
If the program and erase endurance is n (n = 100), each block can be erased n times.  
For example, if a 4-Kbyte block A is erased after writing 1 word data 2,048 times, each to a different address, this counts as one program and erase endurance. Data cannot be written to the same address more than once without erasing the block. (Rewrite prohibited)
3. n denotes the number of blocks to erase.

**Table 21.12 Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics  
(at Topr = 0 to 60 °C)**

Flash Program, Erase Voltage	Flash Read Operation Voltage
VCC = 3.3 ± 0.3V or 5.0 ± 0.5V	VCC = 3.0 to 5.5V

## 22. Electrical Characteristics

### 22.1 Electrical Characteristics (T/V-ver.)

**Table 22.1 Absolute Maximum Ratings**

Symbol	Parameter		Condition	Rated Value	Unit
V <sub>CC</sub>	Supply Voltage (VCC1 = VCC2)		VCC = AVCC	−0.3 to 6.5	V
AV <sub>CC</sub>	Analog Supply Voltage		VCC = AVCC	−0.3 to 6.5	V
V <sub>I</sub>	Input Voltage	RESET, CNVSS, BYTE, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, VREF, XIN		−0.3 to VCC+0.3	V
		P7_1, P9_1		−0.3 to 6.5	V
V <sub>O</sub>	Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, XOUT		−0.3 to VCC+0.3	V
		P7_1, P9_1		−0.3 to 6.5	V
P <sub>d</sub>	Power Dissipation		T <sub>opr</sub> = 25°C	700	mW
T <sub>opr</sub>	Operating Ambient Temperature	When the Microcomputer is Operating		T version: −40 to 85 V version: −40 to 125 (option)	°C
		Flash Program Erase		0 to 60	
T <sub>stg</sub>	Storage Temperature			−65 to 150	°C

option: All options are on request basis.

**Table 22.2 Recommended Operating Conditions (1) <sup>(1)</sup>**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V <sub>CC</sub>	Supply Voltage (VCC1 = VCC2)		4.2	5.0	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>	HIGH Input Voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
		P7_1, P9_1	0.8V <sub>CC</sub>		6.5	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode)	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes)	0.5V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	LOW Input Voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	0		0.2V <sub>CC</sub>	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode)	0		0.2V <sub>CC</sub>	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes)	0		0.16V <sub>CC</sub>	V
I <sub>OH(peak)</sub>	HIGH Peak Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7			−10.0	mA
I <sub>OH(avg)</sub>	HIGH Average Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7			−5.0	mA
I <sub>OL(peak)</sub>	LOW Peak Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			10.0	mA
I <sub>OL(avg)</sub>	LOW Average Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			5.0	mA

**NOTES:**

1. Referenced to VCC = 4.2 to 5.5V at Topr = −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 ports P0, P1, P2, P8\_6, P8\_7, P9 and P10 must be 80mA max.  
The total I<sub>OL(peak)</sub> for ports P3, P4, P5, P6, P7 and P8\_0 to P8\_4 must be 80mA max.  
The total I<sub>OH(peak)</sub> for ports P0, P1, and P2 must be −40mA max.  
The total I<sub>OH(peak)</sub> for ports P3, P4 and P5 must be −40mA max.  
The total I<sub>OH(peak)</sub> for ports P6, P7 and P8\_0 to P8\_4 must be −40mA max.  
The total I<sub>OH(peak)</sub> for ports P8\_6, P8\_7, P9 and P10 must be −40mA max.

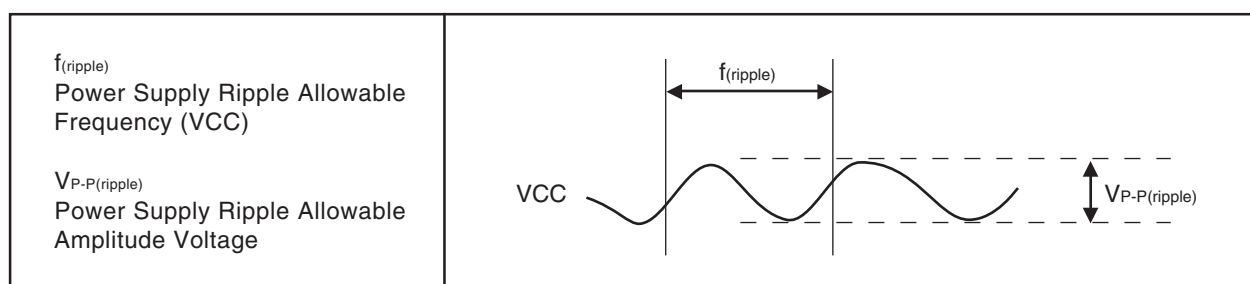
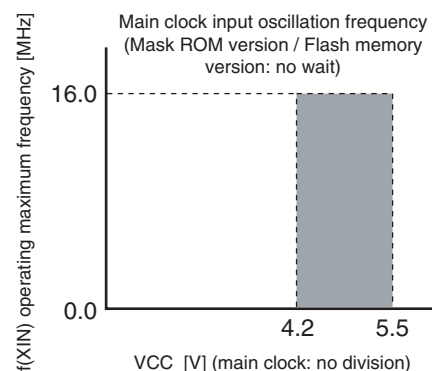


**Table 22.3 Recommended Operating Conditions (2) <sup>(1)</sup>**

Symbol	Parameter				Standard			Unit
					Min.	Typ.	Max.	
f(XIN)	Main Clock Input Oscillation Frequency <sup>(2) (3) (4)</sup>	No Wait	Mask ROM Version Flash Memory Version	VCC = 4.2 to 5.5V	0		16	MHz
f(XCIN)	Sub Clock Oscillation Frequency					32.768	50	kHz
f(Ring)	On-chip Oscillation Frequency					1		MHz
f(PLL)	PLL Clock Oscillation Frequency				16		20	MHz
f(BCLK)	CPU Operation Clock			VCC = 4.2 to 5.5V	0		20	MHz
t <sub>su</sub> (PLL)	PLL Frequency Synthesizer Stabilization Wait Time						20	ms
f(ripple)	Power Supply Ripple Allowable Frequency (VCC)						10	kHz
V <sub>P-P</sub> (ripple)	Power Supply Ripple Allowable Amplitude Voltage			VCC = 5V			0.5	V
V <sub>CC</sub> (ΔV/ΔT)	Power Supply Ripple Rising/Falling Gradient			VCC = 5V			0.3	V/ms

**NOTES:**

1. Referenced to VCC = 4.2 to 5.5V at Topr = -40 to 85°C unless otherwise specified.
2. Relationship between main clock oscillation frequency and supply voltage is shown right.
3. Execute program/erase of flash memory by VCC = 5.0 ± 0.5 V.
4. When using over 16MHz, use PLL clock. PLL clock oscillation frequency which can be used is 16MHz or 20MHz.

**Figure 22.1 Timing of Voltage Fluctuation**

**Table 22.4 Electrical Characteristics (1)** <sup>(1)</sup>

Symbol	Parameter		Measuring Condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7	I <sub>OH</sub> = -5mA	V <sub>CC</sub> -2.0		V <sub>CC</sub>	V
V <sub>OH</sub>	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7	I <sub>OH</sub> = -200μA	V <sub>CC</sub> -0.3		V <sub>CC</sub>	V
V <sub>OH</sub>	HIGH Output Voltage	XOUT	HIGHPOWER	I <sub>OH</sub> = -1mA	3.0	V <sub>CC</sub>	V
			LOWPOWER	I <sub>OH</sub> = -0.5mA	3.0	V <sub>CC</sub>	V
	HIGH Output Voltage	XCOUT	HIGHPOWER	With no load applied	2.5		V
			LOWPOWER	With no load applied	1.6		V
V <sub>OL</sub>	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	I <sub>OL</sub> = 5mA			2.0	V
V <sub>OL</sub>	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	I <sub>OL</sub> = 200μA			0.45	V
V <sub>OL</sub>	LOW Output Voltage	XOUT	HIGHPOWER	I <sub>OL</sub> = 1mA		2.0	V
			LOWPOWER	I <sub>OL</sub> = 0.5mA		2.0	V
	LOW Output Voltage	XCOUT	HIGHPOWER	With no load applied	0		V
			LOWPOWER	With no load applied	0		V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT5, NMI, ADTRG, CTS0 to CTS2, SCL0 to SCL2, SDA0 to SDA2, CLK0 to CLK3, TA0OUT to TA4OUT, KI0 to KI3, RXD0 to RXD2, SIN3		0.2		1.0	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	RESET		0.2		2.5	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	XIN		0.2		0.8	V
I <sub>IH</sub>	HIGH Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	V <sub>I</sub> = 5V			5.0	μA
I <sub>IL</sub>	LOW Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	V <sub>I</sub> = 0V			-5.0	μA
R <sub>PULLUP</sub>	Pull-up Resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7	V <sub>I</sub> = 0V	30	50	170	kΩ
R <sub>IXIN</sub>	Feedback Resistance	XIN			1.5		MΩ
R <sub>IXCIN</sub>	Feedback Resistance	XCIN			15		MΩ
V <sub>RAM</sub>	RAM Retention Voltage		At stop mode	2.0			V

## NOTES:

1. Referenced to V<sub>CC</sub> = 4.2 to 5.5V, V<sub>SS</sub> = 0V at T<sub>opr</sub> = -40 to 85°C, f(BCLK) = 20MHz unless otherwise specified.

**Table 22.5 Electrical Characteristics (2) <sup>(1)</sup>**

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
I <sub>cc</sub>	Power Supply Current (VCC = 4.2 to 5.5V)	Output pins are open and other pins are VSS.	Mask ROM	f(BCLK) = 20MHz, PLL operation, No division		18	32	mA
				On-chip oscillation, No division		1		mA
			Flash Memory	f(BCLK) = 20MHz, PLL operation, No division		20	34	mA
				On-chip oscillation, No division		1.8		mA
			Flash Memory Program	f(BCLK) = 10MHz, VCC = 5V		15		mA
			Flash Memory Erase	f(BCLK) = 10MHz, VCC = 5V		25		mA
			Mask ROM	f(BCLK) = 32kHz, Low power dissipation mode, ROM <sup>(2)</sup>		25		μA
			Flash Memory	f(BCLK) = 32kHz, Low power dissipation mode, RAM <sup>(2)</sup>		25		μA
				f(BCLK) = 32kHz, Low power dissipation mode, Flash memory <sup>(2)</sup>		420		μA
			Mask ROM Flash Memory	On-chip oscillation, Wait mode		50		μA
				f(BCLK) = 32kHz, Wait mode (3), Oscillation capacity High		8.5		μA
				f(BCLK) = 32kHz, Wait mode (3), Oscillation capacity Low		3.0		μA
				Stop mode, T <sub>opr</sub> = 25°C		0.8	3.0	μA

## NOTES:

1. Referenced to VCC = 4.2 to 5.5V, VSS = 0V at T<sub>opr</sub> = -40 to 85°C, f(BCLK) = 20MHz unless otherwise specified.
2. This indicates the memory in which the program to be executed exists.
3. With one timer operated using fC32.

**Table 22.6 A/D Conversion Characteristics** <sup>(1)</sup>

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
—	Resolution		VREF = VCC				10	Bit
INL	Integral Nonlinearity Error	10 bits	VREF = VCC	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±3	LSB
			= VCC					
			= 5V	External operation amp connection mode			±7	LSB
		8 bits	VREF = AVCC = VCC = 5V				±2	LSB
—	Absolute Accuracy	10 bits	VREF = VCC	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±3	LSB
			= VCC					
			= 5V	External operation amp connection mode			±7	LSB
		8 bits	VREF = AVCC = VCC = 5V				±2	LSB
DNL	Differential Nonlinearity Error						±1	LSB
—	Offset Error						±3	LSB
—	Gain Error						±3	LSB
R <sub>LADDER</sub>	Resistor Ladder		VREF = VCC		10		40	kΩ
t <sub>CONV</sub>	10-bit Conversion Time, Sample & Hold Available		VREF = VCC = 5V, φAD = 10MHz		3.3			μs
	8-bit Conversion time, Sample & Hold Available		VREF = VCC = 5V, φAD = 10MHz		2.8			μs
t <sub>SAMP</sub>	Sampling Time				0.3			μs
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 VCC = AVCC = VREF = 4.2 to 5.5V, VSS = AVSS = 0V, -40 to 85°C unless otherwise specified.
2. φAD frequency must be 10MHz or less.
3. When sample & hold is disabled, φAD frequency must be 250kHz or more in addition to a limit of NOTE 2.  
When sample & hold is enabled, φAD frequency must be 1MHz or more in addition to a limit of NOTE 2.

**Table 22.7 D/A conversion Characteristics** <sup>(1)</sup>

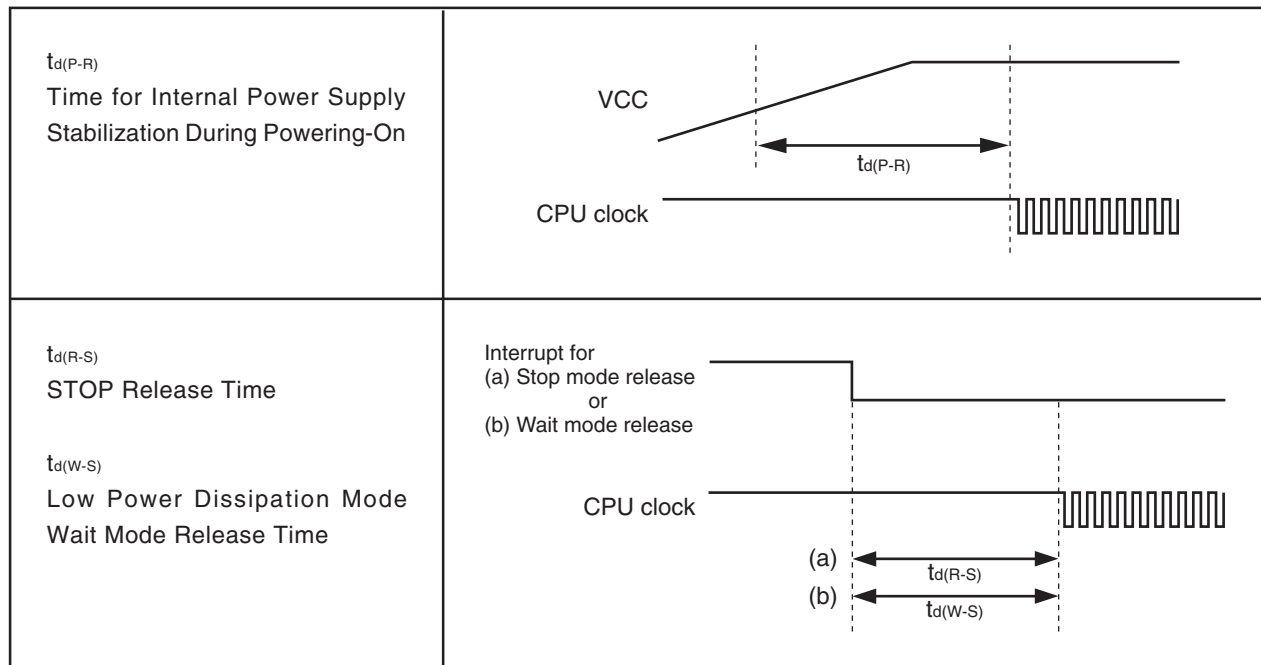
Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute Accuracy				1.0	%
t <sub>su</sub>	Setup Time				3	μs
R <sub>o</sub>	Output Resistance		4	10	20	kΩ
I <sub>VREF</sub>	Reference Power Supply Input Current	(NOTE 2)			1.5	mA

## NOTES:

1. Referenced to VCC = AVCC = VREF = 4.2 to 5.5V, VSS = AVSS = 0V, -40 to 85°C unless otherwise specified.
2. This applies when using one D/A converter, with the DAi register (i = 0, 1) for the unused D/A converter set to "00h".  
The resistor ladder of the A/D converter is not included. Also, the current I<sub>VREF</sub> always flows even though VREF may have been set to be unconnected by the ADCON1 register.

**Table 22.8 Power Supply Circuit Timing Characteristics**

Symbol	Parameter	Measuring Condition	Standard			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Time for Internal Power Supply Stabilization During Powering-On	VCC = 4.2 to 5.5V			2	ms
$t_{d(R-S)}$	STOP Release Time				150	$\mu$ s
$t_{d(W-S)}$	Low Power Dissipation Mode Wait Mode Release Time				150	$\mu$ s



**Figure 22.2 Power Supply Circuit Timing Diagram**

**Timing Requirements****VCC = 5V****(Referenced to VCC = 5V, VSS = 0V, at Topr = –40 to 85°C unless otherwise specified)****Table 22.9 External Clock Input (XIN Input)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c</sub>	External Clock Input Cycle Time	62.5		ns
t <sub>w(H)</sub>	External Clock Input HIGH Pulse Width	25		ns
t <sub>w(L)</sub>	External Clock Input LOW Pulse Width	25		ns
t <sub>r</sub>	External Clock Rise Time		15	ns
t <sub>f</sub>	External Clock Fall Time		15	ns

**Table 22.10 Memory Expansion Mode and Microprocessor Mode**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>ac1</sub> (RD-DB)	Data input access time (for setting with no wait)		(NOTE 1)	ns
t <sub>ac2</sub> (RD-DB)	Data input access time (for setting with wait)		(NOTE 2)	ns
t <sub>ac3</sub> (RD-DB)	Data input access time (when accessing multiplexed bus area)		(NOTE 3)	ns
t <sub>su</sub> (DB-RD)	Data input setup time	40		ns
t <sub>su</sub> (RDY-BCLK)	RDY input setup time	30		ns
t <sub>su</sub> (HOLD-BCLK)	HOLD input setup time	40		ns
t <sub>h</sub> (RD-DB)	Data input hold time	0		ns
t <sub>h</sub> (BCLK-RDY)	RDY input hold time	0		ns
t <sub>h</sub> (BCLK-HOLD)	HOLD input hold time	0		ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]} \quad n \text{ is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

**Timing Requirements****VCC = 5V****(Referenced to VCC = 5V, VSS = 0V, at Topr = –40 to 85°C unless otherwise specified)****Table 22.11 Timer A Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	100		ns
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	40		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	40		ns

**Table 22.12 Timer A Input (Gating Input in Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	400		ns
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	200		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	200		ns

**Table 22.13 Timer A Input (External Trigger Input in One-shot Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	200		ns
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	100		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	100		ns

**Table 22.14 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	100		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	100		ns

**Table 22.15 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(UP)</sub>	TAiOUT Input Cycle Time	2000		ns
t <sub>w(UPH)</sub>	TAiOUT Input HIGH Pulse Width	1000		ns
t <sub>w(UPL)</sub>	TAiOUT Input LOW Pulse Width	1000		ns
t <sub>su(UP-TIN)</sub>	TAiOUT Input Setup Time	400		ns
t <sub>h(TIN-UP)</sub>	TAiOUT Input Hold Time	400		ns

**Table 22.16 Timer A Input (Two-phase Pulse Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	800		ns
t <sub>su(TAIN-TAOUT)</sub>	TAiOUT Input Setup Time	200		ns
t <sub>su(TAOUT-TAIN)</sub>	TAiIN Input Setup Time	200		ns

**Timing Requirements****VCC = 5V****(Referenced to VCC = 5V, VSS = 0V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.17 Timer B Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on both edges)	80		ns

**Table 22.18 Timer B Input (Pulse Period Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	400		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	200		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	200		ns

**Table 22.19 Timer B Input (Pulse Width Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	400		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	200		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	200		ns

**Table 22.20 A/D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	ADTRG $\bar{}$ Input Cycle Time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	ADTRG $\bar{}$ Input LOW Pulse Width	125		ns

**Table 22.21 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi Input Cycle Time	200		ns
$t_{w(CKH)}$	CLKi Input HIGH Pulse Width	100		ns
$t_{w(CKL)}$	CLKi Input LOW Pulse Width	100		ns
$t_{d(C-Q)}$	TXDi Output Delay Time		80	ns
$t_{h(C-Q)}$	TXDi Hold Time	0		ns
$t_{su(D-C)}$	RXDi Input Setup Time	70		ns
$t_{h(C-D)}$	RXDi Input Hold Time	90		ns

**Table 22.22 External Interrupt INTi Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	INTi Input HIGH Pulse Width	250		ns
$t_{w(INL)}$	INTi Input LOW Pulse Width	250		ns



**Switching Characteristics****VCC = 5V****(Referenced to VCC = 5V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.23 Memory Expansion Mode and Microprocessor Mode (for setting with no wait)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 22.3		25	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (refers to RD)		0		ns
t <sub>h</sub> (WR-AD)	Address output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			25	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time			15	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time		–4		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			25	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			25	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (refers to BCLK)			40	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (refers to BCLK) <sup>(3)</sup>		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (refers to WR)		(NOTE 2)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (refers to WR) <sup>(3)</sup>		(NOTE 1)		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time			40	ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]} \quad f(\text{BCLK}) \text{ is 12.5 MHz or less.}$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

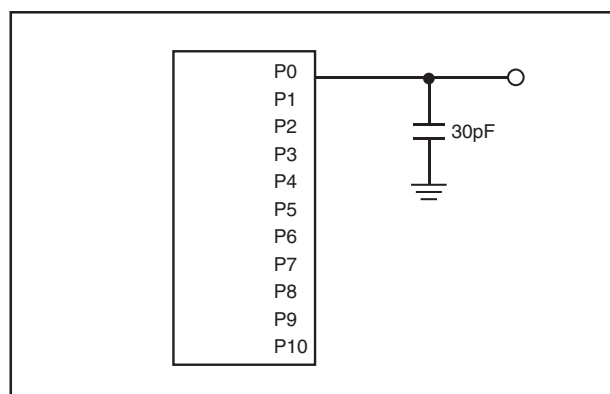
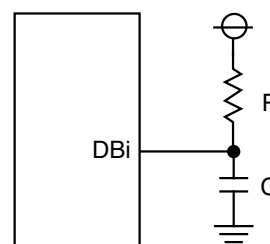
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when  $V_{OL} = 0.2 V_{CC}$ ,  $C = 30 \text{ pF}$ ,

$R = 1 \text{ k}\Omega$ , hold time of output “L” level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$

**Figure 22.3 Port P0 to P10 Measurement Circuit**

**Switching Characteristics****VCC = 5V****(Referenced to VCC = 5V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.24 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 22.3		25	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (refers to RD)		0		ns
t <sub>h</sub> (WR-AD)	Address output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			25	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time			15	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time		–4		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			25	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			25	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (refers to BCLK)			40	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (refers to BCLK) <sup>(3)</sup>		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (refers to WR)		(NOTE 2)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (refers to WR) <sup>(3)</sup>		(NOTE 1)		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time			40	ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]}$$

n is “1” for 1-wait setting, “2” for 2-wait setting and “3” for 3-wait setting.  
When n = 1, f(BCLK) is 12.5 MHz or less.

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

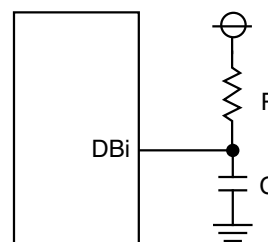
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when  $V_{OL} = 0.2 V_{CC}$ ,  $C = 30 \text{ pF}$ ,

$R = 1 \text{ k}\Omega$ , hold time of output “L” level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$



**Switching Characteristics****VCC = 5V****(Referenced to VCC = 5V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.25 Memory Expansion Mode and Microprocessor Mode  
(for 2- to 3-wait setting, external area access and multiplexed bus selection)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 22.3		25	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (refers to RD)		(NOTE 1)		ns
t <sub>h</sub> (WR-AD)	Address output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			25	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-CS)	Chip select output hold time (refers to RD)		(NOTE 1)		ns
t <sub>h</sub> (WR-CS)	Chip select output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			25	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			25	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (refers to BCLK)			40	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (refers to WR)		(NOTE 2)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time			40	ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time (refers to BCLK)			15	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time (refers to BCLK)		–4		ns
t <sub>d</sub> (AD-ALE)	ALE signal output delay time (refers to Address)		(NOTE 3)		ns
t <sub>h</sub> (ALE-AD)	ALE signal output hold time (refers to Address)		(NOTE 4)		ns
t <sub>d</sub> (AD-RD)	RD signal output delay from the end of Address		0		ns
t <sub>d</sub> (AD-WR)	WR signal output delay from the end of Address		0		ns
t <sub>dZ</sub> (RD-AD)	Address output floating start time			8	ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 25 \text{ [ns]}$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 15 \text{ [ns]}$$

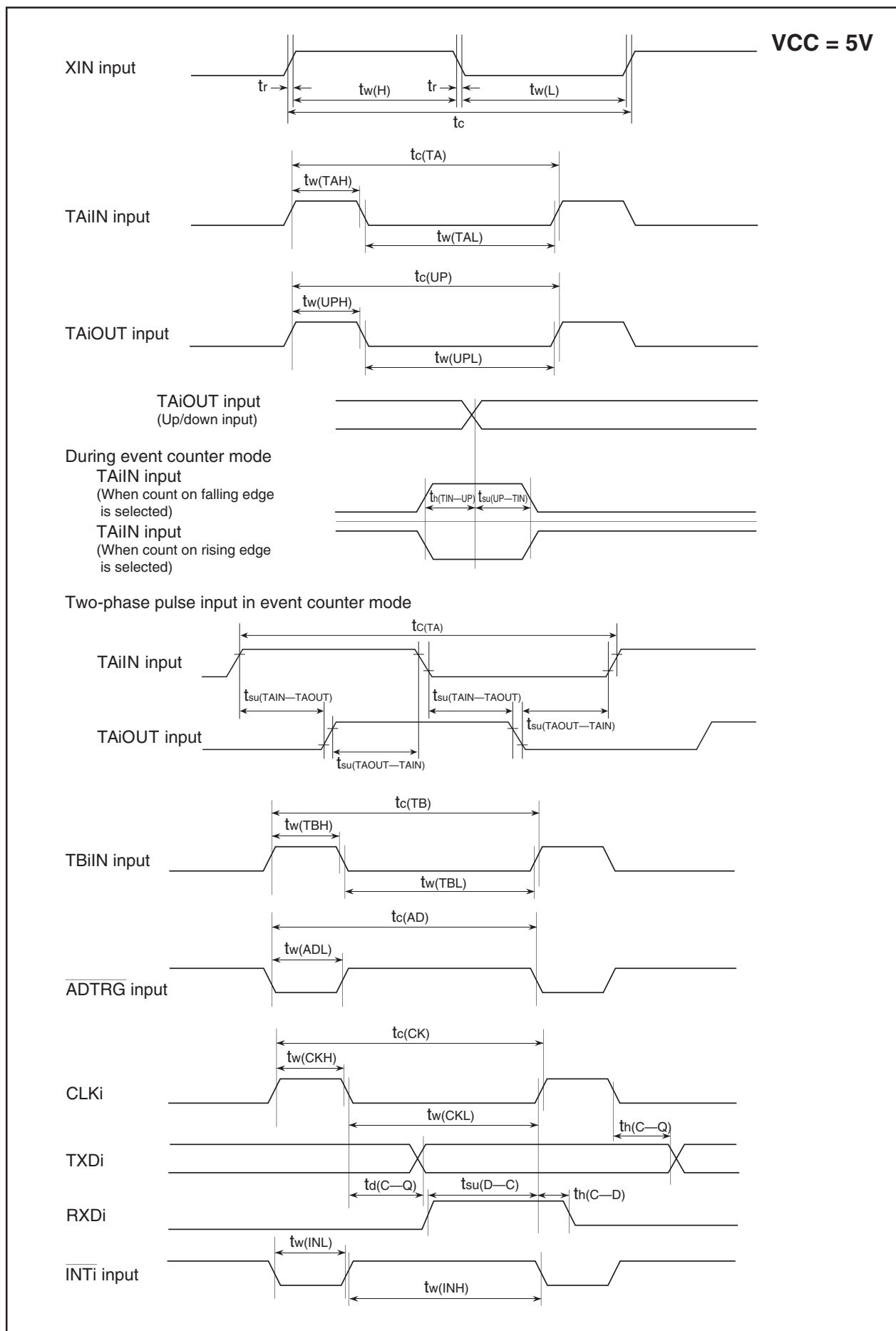
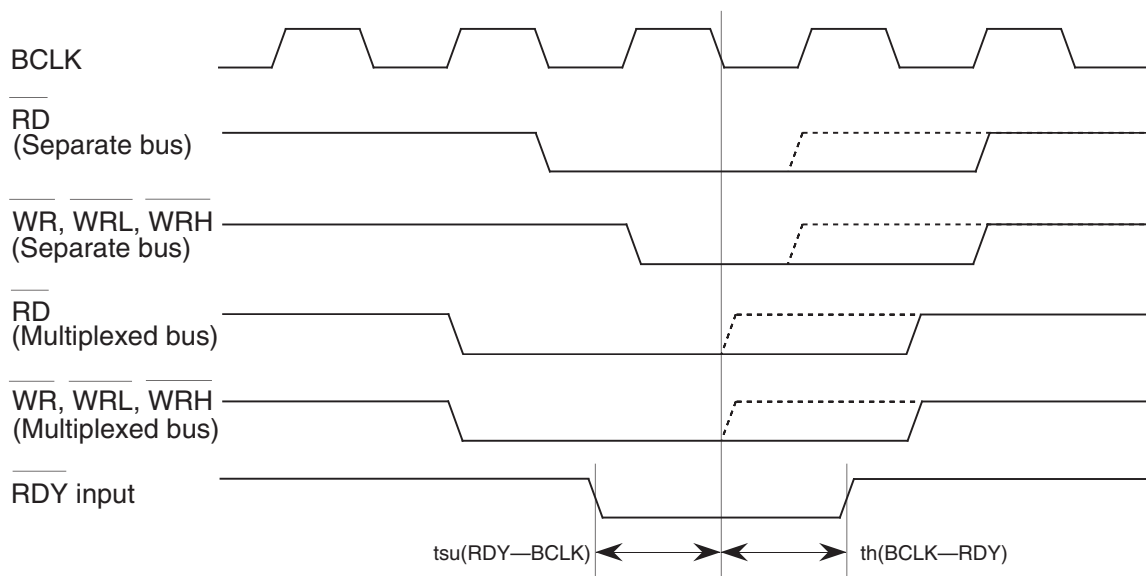


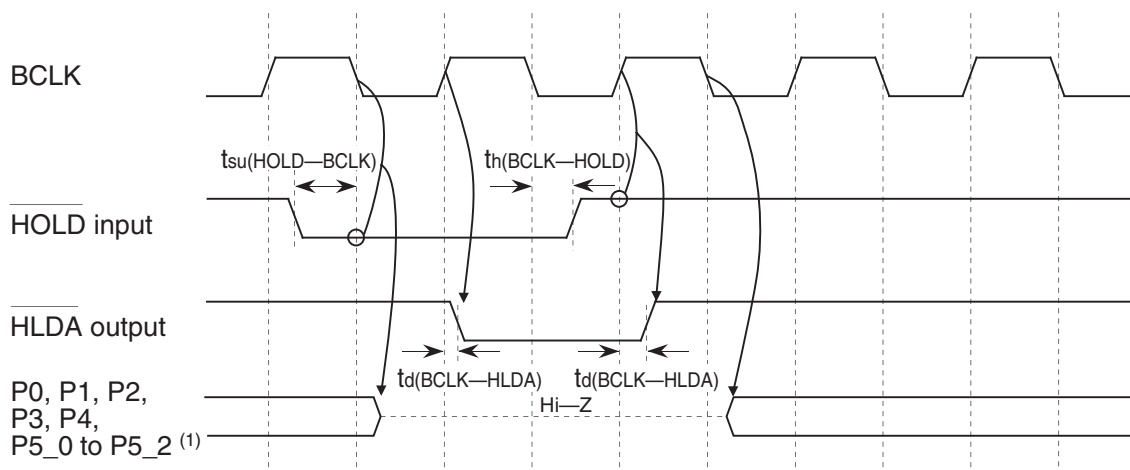
Figure 22.4 Timing Diagram (1)

**Memory Expansion Mode and Microprocessor Mode****VCC = 5V**

(Effective for setting with wait)



(Common to setting with wait and setting without wait)

**NOTE:**

1. The above pins are set to high-impedance regardless of the input level of the BYTE pin, the PM06 bit in the PM0 register and the PM11 bit in the PM1 register.

**Measuring conditions :**

- VCC = 5 V
- Input timing voltage : Determined with  $V_{IL} = 1.0 \text{ V}$ ,  $V_{IH} = 4.0 \text{ V}$
- Output timing voltage: Determined with  $V_{OL} = 2.5 \text{ V}$ ,  $V_{OH} = 2.5 \text{ V}$

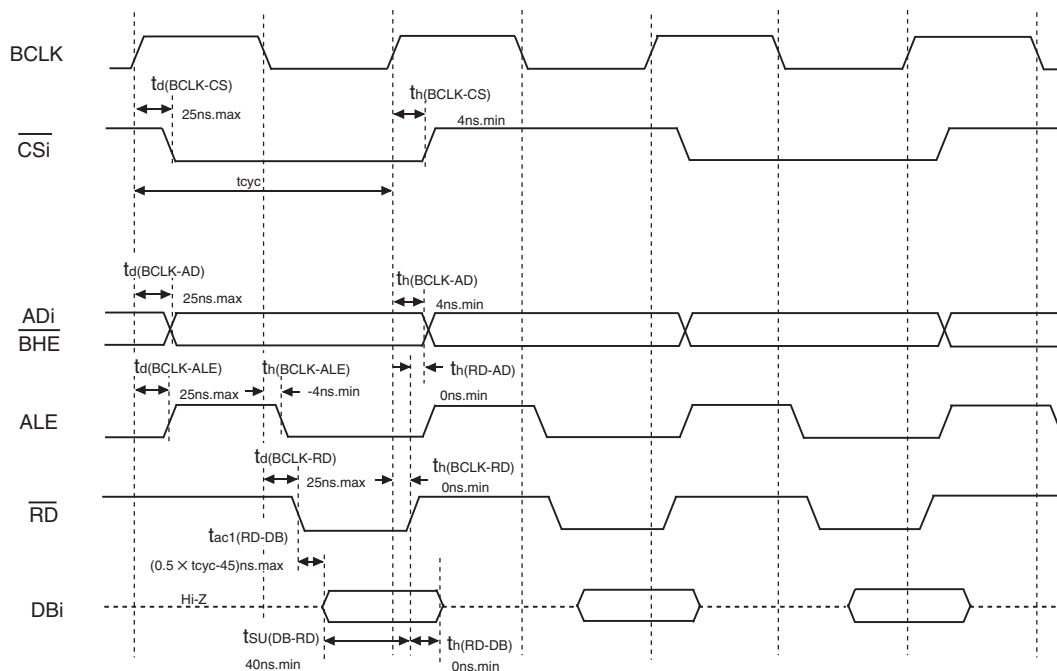
**Figure 22.5 Timing Diagram (2)**

## Memory Expansion Mode and Microprocessor Mode

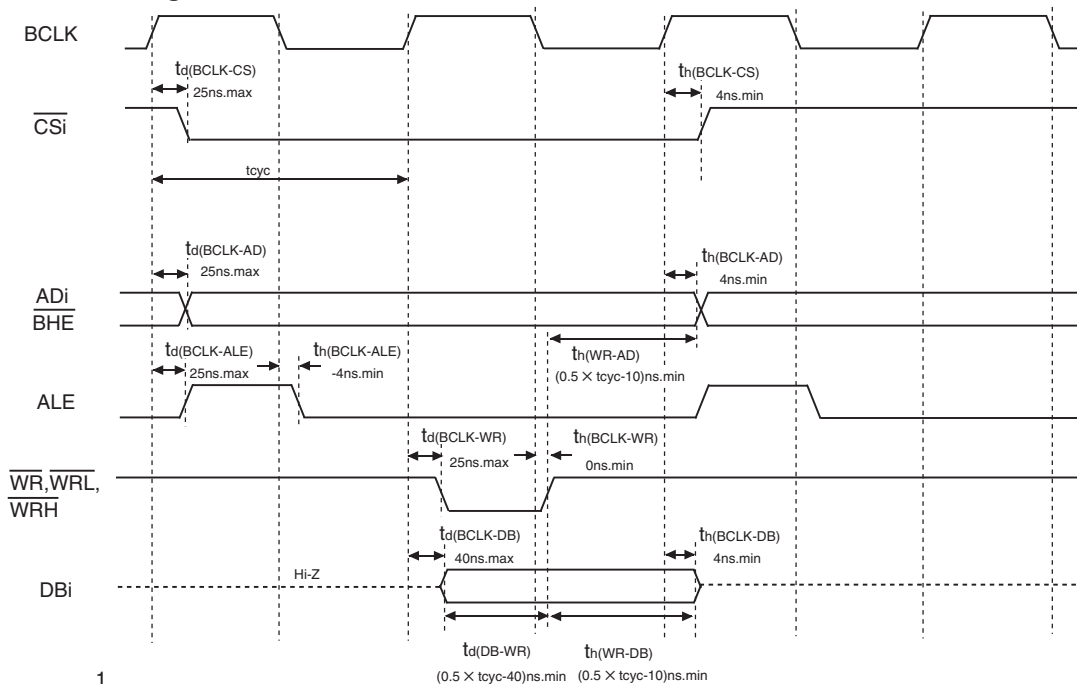
(For setting with no wait)

VCC = 5V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions :

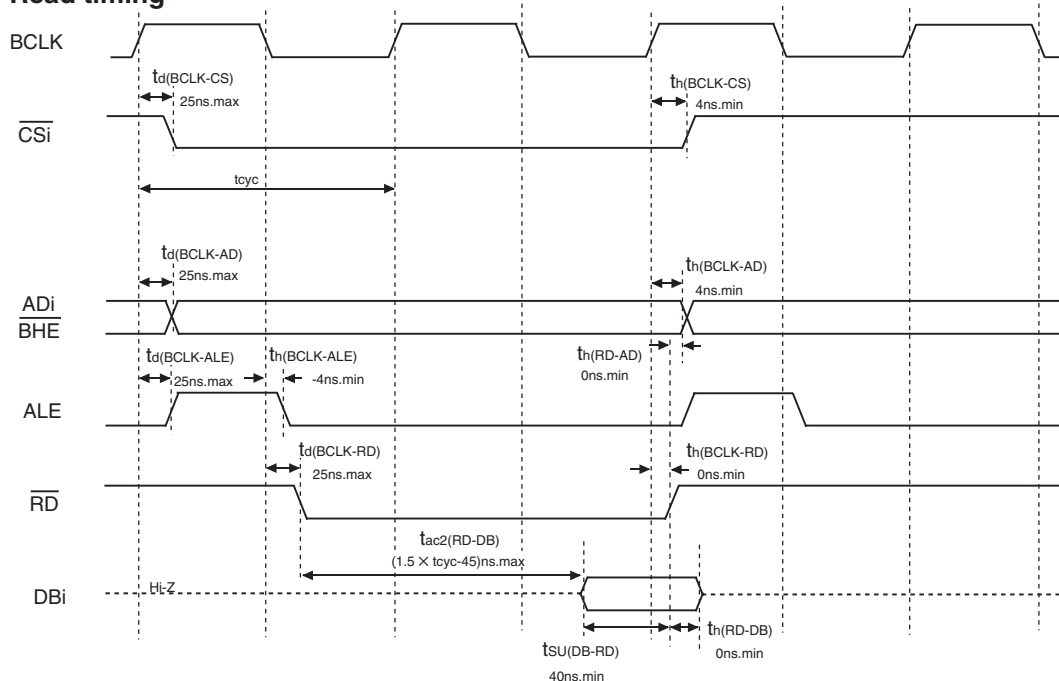
- VCC = 5 V
- Input timing voltage :  $V_{IL} = 0.8 \text{ V}$ ,  $V_{IH} = 2.0 \text{ V}$
- Output timing voltage :  $V_{OL} = 0.4 \text{ V}$ ,  $V_{OH} = 2.4 \text{ V}$

Figure 22.6 Timing Diagram (3)

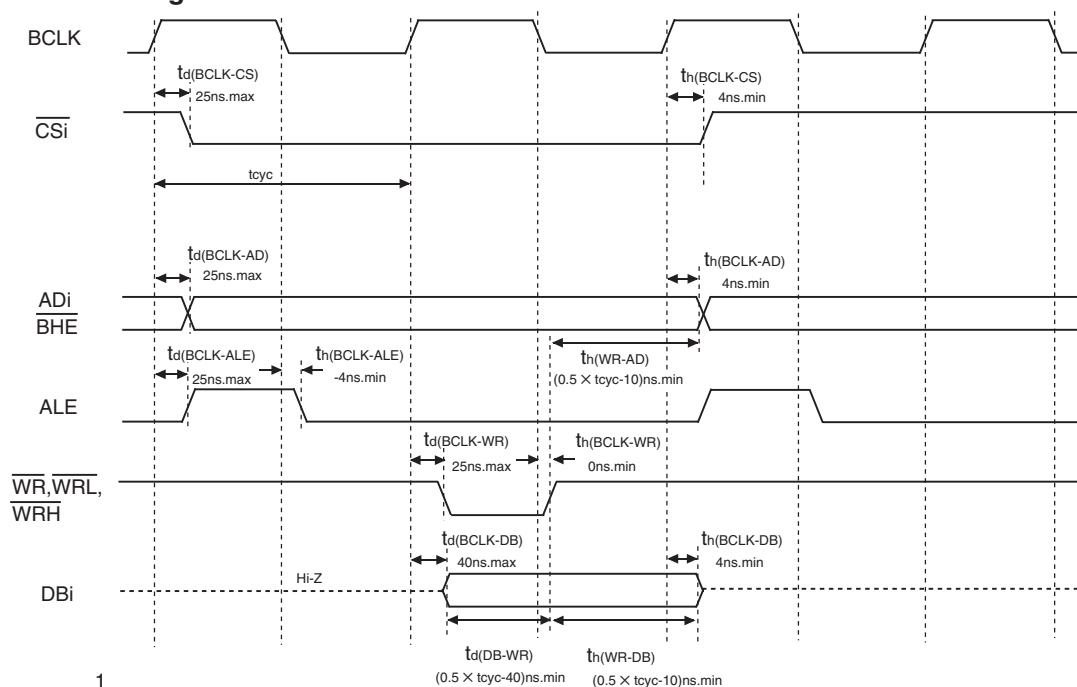
## Memory Expansion Mode and Microprocessor Mode (For 1-wait setting and external area access)

VCC = 5V

### Read timing



### Write timing



$$t_{\text{cyc}} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions :

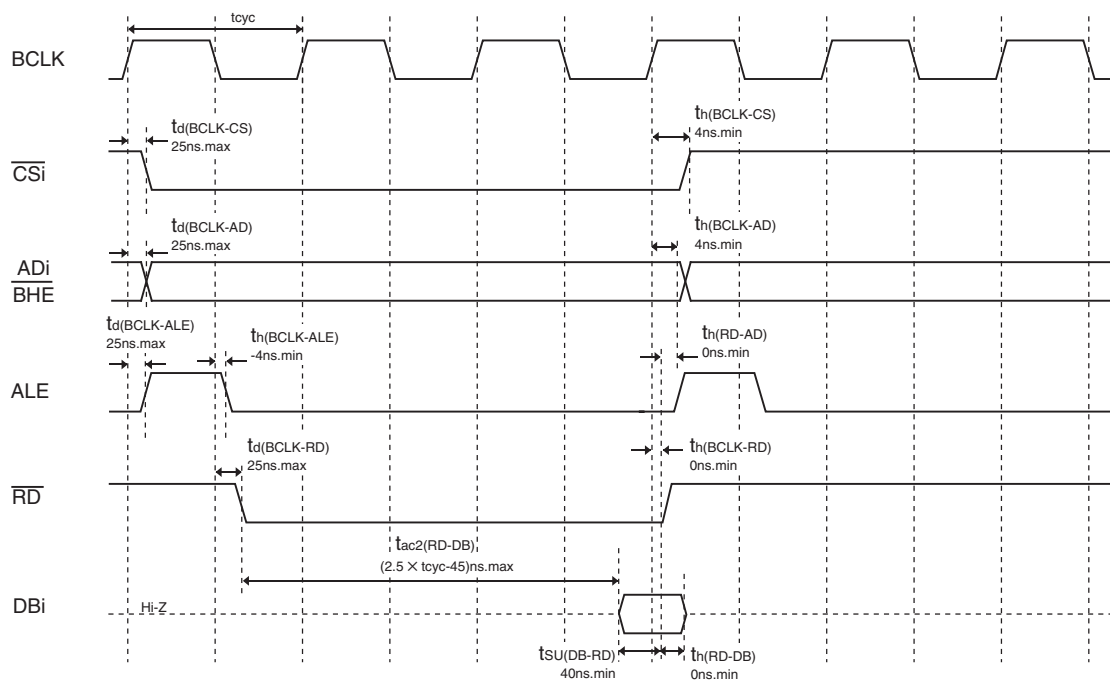
- VCC = 5 V
- Input timing voltage :  $V_{\text{IL}} = 0.8 \text{ V}$ ,  $V_{\text{IH}} = 2.0 \text{ V}$
- Output timing voltage :  $V_{\text{OL}} = 0.4 \text{ V}$ ,  $V_{\text{OH}} = 2.4 \text{ V}$

Figure 22.7 Timing Diagram (4)

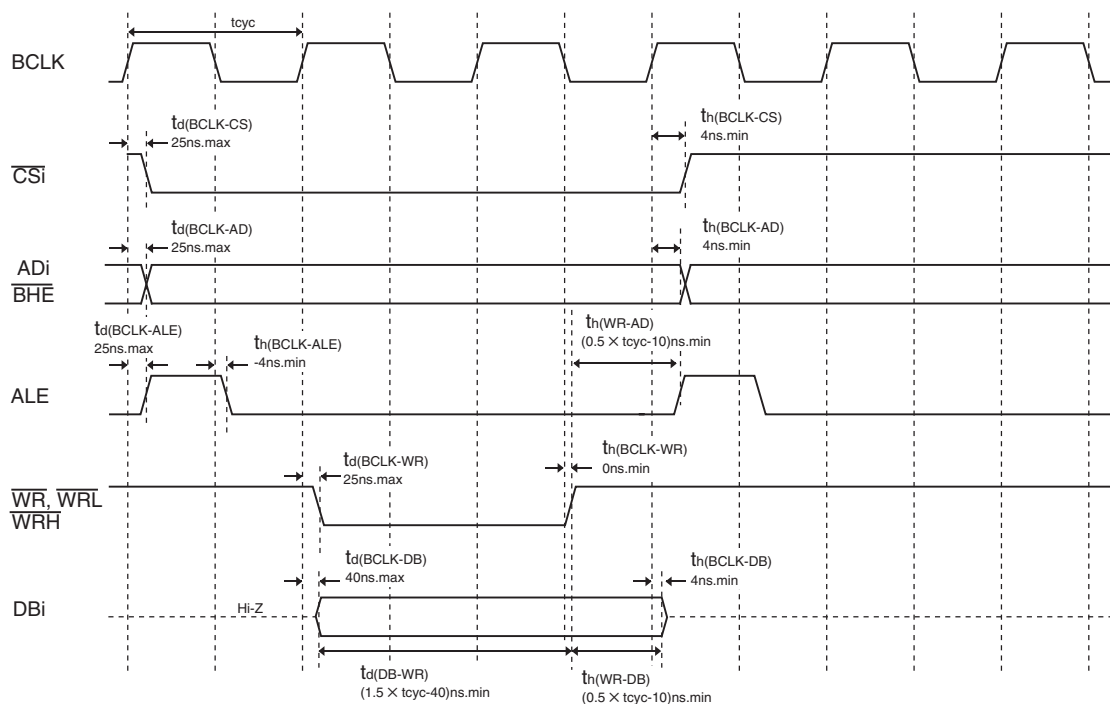
## Memory Expansion Mode and Microprocessor Mode (For 2-wait setting and external area access)

VCC = 5V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

Measuring conditions :

- VCC = 5 V
- Input timing voltage :  $V_{IL} = 0.8 \text{ V}$ ,  $V_{IH} = 2.0 \text{ V}$
- Output timing voltage :  $V_{OL} = 0.4 \text{ V}$ ,  $V_{OH} = 2.4 \text{ V}$

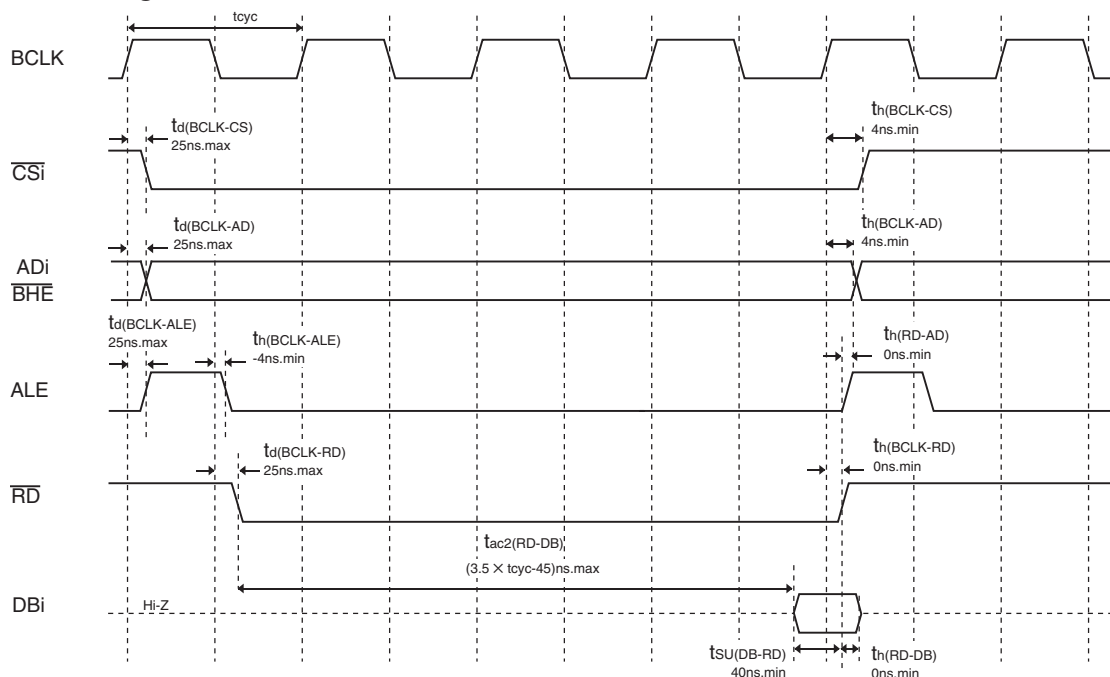
Figure 22.8 Timing Diagram (5)



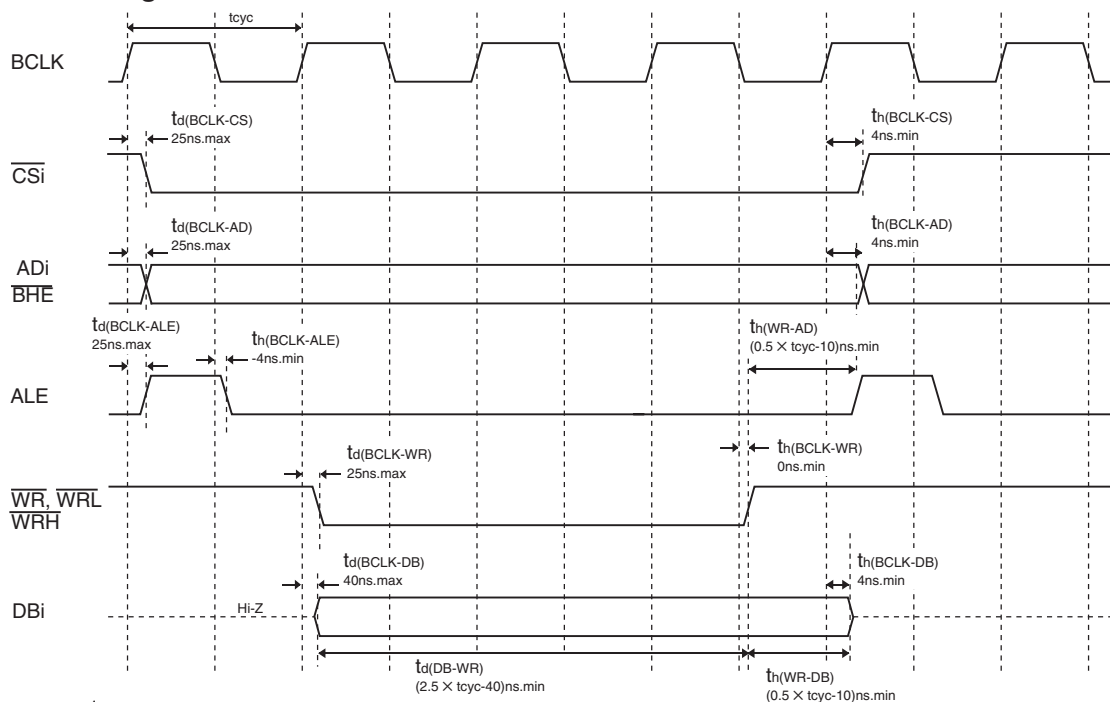
## Memory Expansion Mode and Microprocessor Mode (For 3-wait setting and external area access)

VCC = 5V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

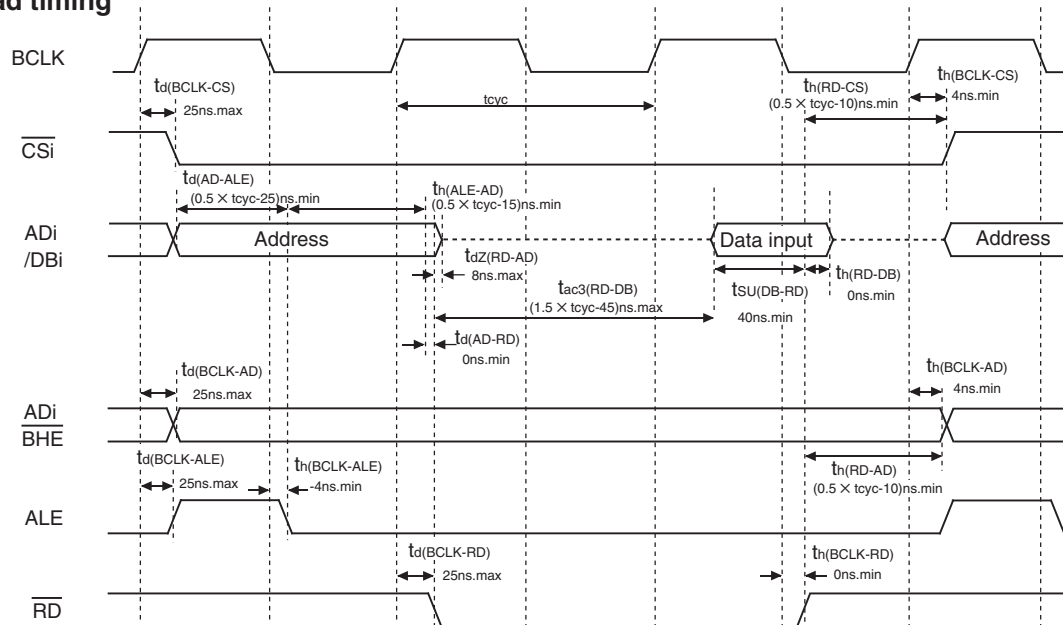
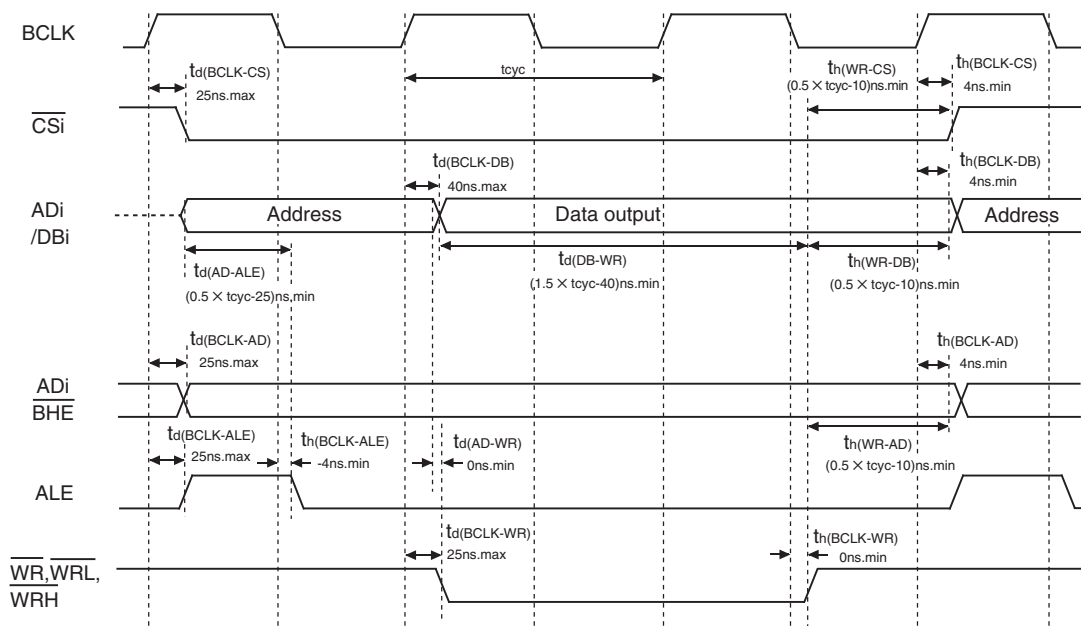
Measuring conditions :

- VCC = 5 V
- Input timing voltage :  $V_{IL} = 0.8 V$ ,  $V_{IH} = 2.0 V$
- Output timing voltage :  $V_{OL} = 0.4 V$ ,  $V_{OH} = 2.4 V$

Figure 22.9 Timing Diagram (6)

**Memory Expansion Mode and Microprocessor Mode****VCC = 5V**

(For 1- or 2-wait setting, external area access and multiplexed bus selection)

**Read timing****Write timing**

$$t_{\text{cyc}} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions :

- VCC = 5 V
- Input timing voltage :  $V_{\text{IL}} = 0.8 \text{ V}$ ,  $V_{\text{IH}} = 2.0 \text{ V}$
- Output timing voltage :  $V_{\text{OL}} = 0.4 \text{ V}$ ,  $V_{\text{OH}} = 2.4 \text{ V}$

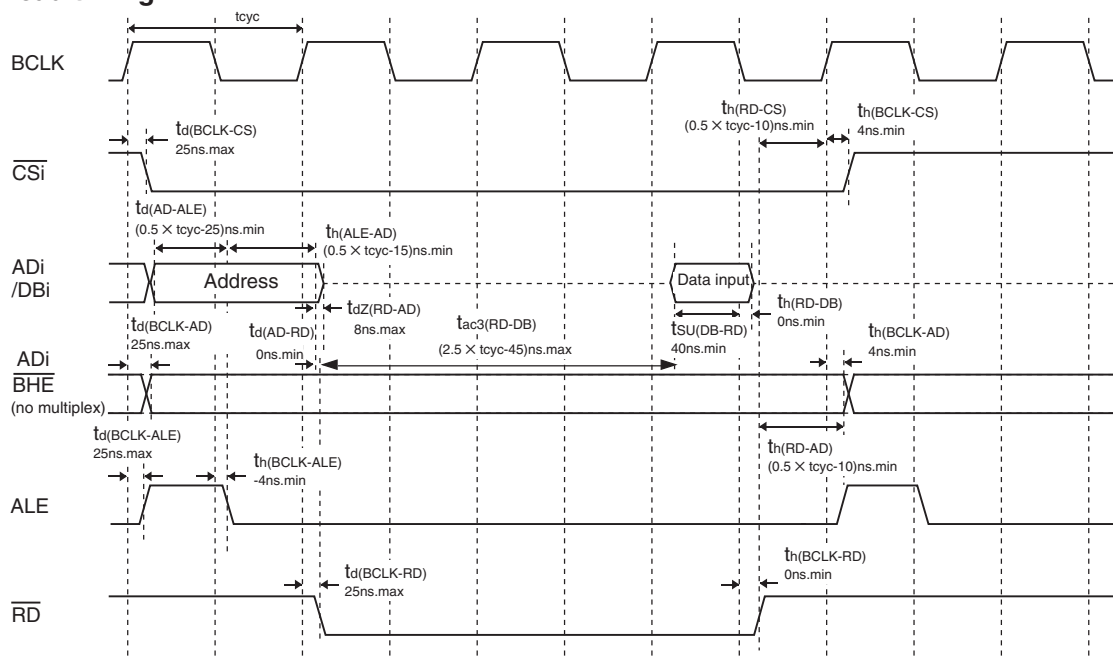
**Figure 22.10 Timing Diagram (7)**

## Memory Expansion Mode and Microprocessor Mode

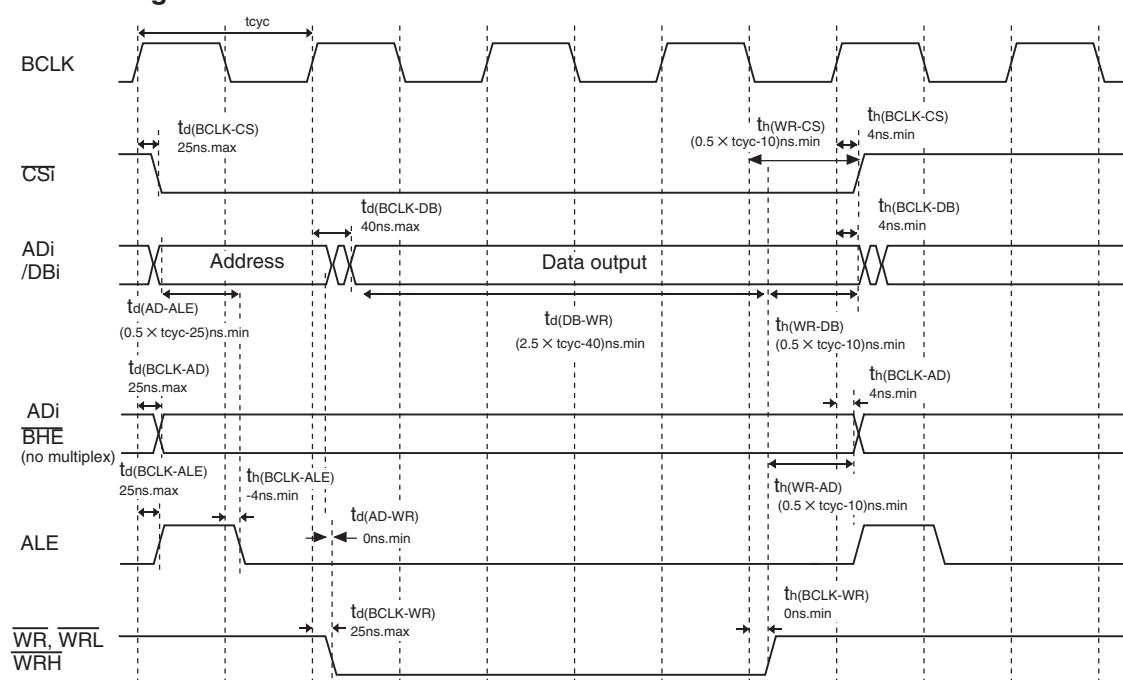
(For 3-wait setting, external area access and multiplexed bus selection)

VCC = 5V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

Measuring conditions :

- VCC = 5 V
- Input timing voltage :  $V_{IL} = 0.8$  V,  $V_{IH} = 2.0$  V
- Output timing voltage :  $V_{OL} = 0.4$  V,  $V_{OH} = 2.4$  V

Figure 22.11 Timing Diagram (8)

## 22.2 Electrical Characteristics (Normal-ver.)

**Table 22.26 Absolute Maximum Ratings**

Symbol	Parameter		Condition	Rated Value	Unit
V <sub>CC</sub>	Supply Voltage (VCC1 = VCC2)		VCC = AVCC	−0.3 to 6.5	V
AV <sub>CC</sub>	Analog Supply Voltage		VCC = AVCC	−0.3 to 6.5	V
V <sub>I</sub>	Input Voltage	RESET, CNVSS, BYTE, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, VREF, XIN		−0.3 to VCC+0.3	V
		P7_1, P9_1		−0.3 to 6.5	V
V <sub>O</sub>	Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, XOUT		−0.3 to VCC+0.3	V
		P7_1, P9_1		−0.3 to 6.5	V
P <sub>d</sub>	Power Dissipation		T <sub>opr</sub> = 25°C	700	mW
T <sub>opr</sub>	Operating Ambient Temperature	When the Microcomputer is Operating		−40 to 85	°C
		Flash Program Erase		0 to 60	
T <sub>stg</sub>	Storage Temperature			−65 to 150	°C

**Table 22.27 Recommended Operating Conditions (1) <sup>(1)</sup>**

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V <sub>CC</sub>	Supply Voltage (VCC1 = VCC2)		3.0	5.0	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>	HIGH Input Voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
		P7_1, P9_1	0.8V <sub>CC</sub>		6.5	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode)	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes)	0.5V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	LOW Input Voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	0		0.2V <sub>CC</sub>	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode)	0		0.2V <sub>CC</sub>	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes)	0		0.16V <sub>CC</sub>	V
I <sub>OH(peak)</sub>	HIGH Peak Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7			−10.0	mA
I <sub>OH(avg)</sub>	HIGH Average Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7			−5.0	mA
I <sub>OL(peak)</sub>	LOW Peak Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			10.0	mA
I <sub>OL(avg)</sub>	LOW Average Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			5.0	mA

**NOTES:**

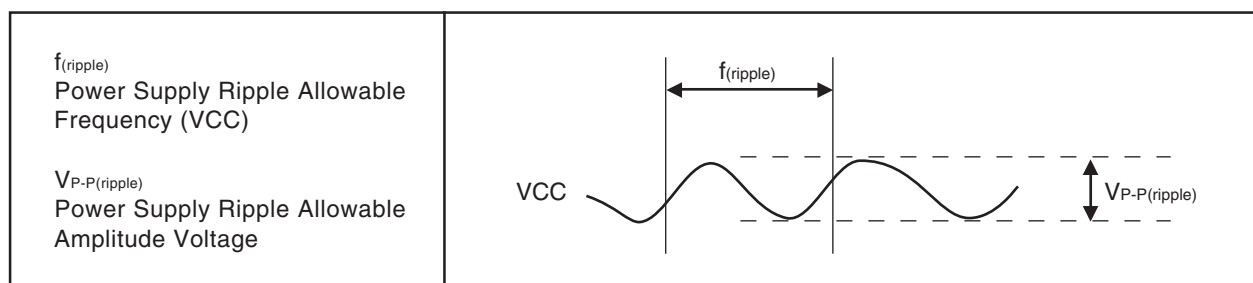
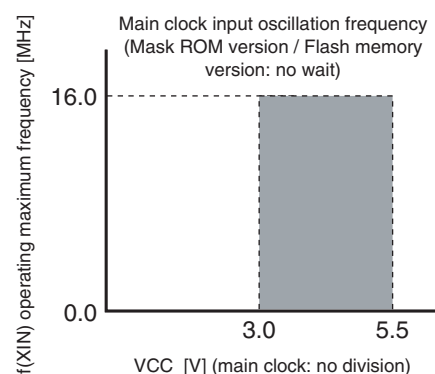
1. Referenced to VCC = 3.0 to 5.5V at Topr = −40 to 85°C unless otherwise specified.
2. The mean output current is the mean value within 100 ms.
3. The total I<sub>OL(peak)</sub> for ports P0, P1, P2, P8\_6, P8\_7, P9 and P10 must be 80mA max.  
The total I<sub>OL(peak)</sub> for ports P3, P4, P5, P6, P7 and P8\_0 to P8\_4 must be 80mA max.  
The total I<sub>OH(peak)</sub> for ports P0, P1, and P2 must be −40mA max.  
The total I<sub>OH(peak)</sub> for ports P3, P4 and P5 must be −40mA max.  
The total I<sub>OH(peak)</sub> for ports P6, P7 and P8\_0 to P8\_4 must be −40mA max.  
The total I<sub>OH(peak)</sub> for ports P8\_6, P8\_7, P9 and P10 must be −40mA max.

**Table 22.28 Recommended Operating Conditions (2) <sup>(1)</sup>**

Symbol	Parameter				Standard			Unit
					Min.	Typ.	Max.	
f(XIN)	Main Clock Input Oscillation Frequency <sup>(2) (3) (4)</sup>	No Wait	Mask ROM Version	VCC = 3.0 to 5.5V	0		16	MHz
			Flash Memory Version					
f(XCIN)	Sub Clock Oscillation Frequency					32.768	50	kHz
f(Ring)	On-chip Oscillation Frequency					1		MHz
f(PLL)	PLL Clock Oscillation Frequency				16		24	MHz
f(BCLK)	CPU Operation Clock			VCC = 3.0 to 5.5V	0		24	MHz
t <sub>su</sub> (PLL)	PLL Frequency Synthesizer Stabilization Wait Time						20	ms
f <sub>(ripple)</sub>	Power Supply Ripple Allowable Frequency (VCC)						10	kHz
V <sub>P-P</sub> (ripple)	Power Supply Ripple Allowable Amplitude Voltage			VCC = 5V			0.5	V
				VCC = 3.3V			0.3	
V <sub>CC</sub> (ΔV/ΔT)	Power Supply Ripple Rising/Falling Gradient			VCC = 5V			0.3	V/ms
				VCC = 3.3V			0.3	

## NOTES:

1. Referenced to VCC = 3.0 to 5.5V at Topr = -40 to 85°C unless otherwise specified.
2. Relationship between main clock oscillation frequency and supply voltage is shown right.
3. Execute program/erase of flash memory by VCC = 3.3 ± 0.3 V or VCC = 5.0 ± 0.5 V.
4. When using over 16MHz, use PLL clock. PLL clock oscillation frequency which can be used is 16MHz, 20MHz or 24MHz.

**Figure 22.12 Timing of Voltage Fluctuation**

**Table 22.29 A/D Conversion Characteristics <sup>(1)</sup>**

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
—	Resolution		VREF = VCC				10	Bit
INL	Integral Nonlinearity Error	10 bits	VREF = VCC	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±3	LSB
				External operation amp connection mode			±7	LSB
			VREF = 3.3V	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±5	LSB
				External operation amp connection mode			±7	LSB
		8 bits	VREF = AVCC = VCC = 5.0V, 3.3V				±2	LSB
—	Absolute Accuracy	10 bits	VREF = VCC	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±3	LSB
				External operation amp connection mode			±7	LSB
			VREF = 3.3V	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±5	LSB
				External operation amp connection mode			±7	LSB
		8 bits	VREF = AVCC = VCC = 5.0V, 3.3V				±2	LSB
DNL	Differential Nonlinearity Error						±1	LSB
—	Offset Error						±3	LSB
—	Gain Error						±3	LSB
RLADDER	Resistor Ladder		VREF = VCC		10		40	kΩ
tCONV	10-bit Conversion Time, Sample & Hold Available		VREF = VCC = 5V, ϕAD = 10MHz		3.3			μs
	8-bit Conversion time, Sample & Hold Available		VREF = VCC = 5V, ϕAD = 10MHz		2.8			μs
tsAMP	Sampling Time				0.3			μs
VREF	Reference Voltage				2.0		VCC	V
VIA	Analog Input Voltage				0		VREF	V

## NOTES:

1. Referenced to VCC = AVCC = VREF = 3.3 to 5.5V, VSS = AVSS = 0V, -40 to 85°C unless otherwise specified.
2. φAD frequency must be 10MHz or less.
3. When sample & hold is disabled, φAD frequency must be 250kHz or more in addition to a limit of NOTE 2.  
When sample & hold is enabled, φAD frequency must be 1MHz or more in addition to a limit of NOTE 2.

**Table 22.30 D/A conversion Characteristics <sup>(1)</sup>**

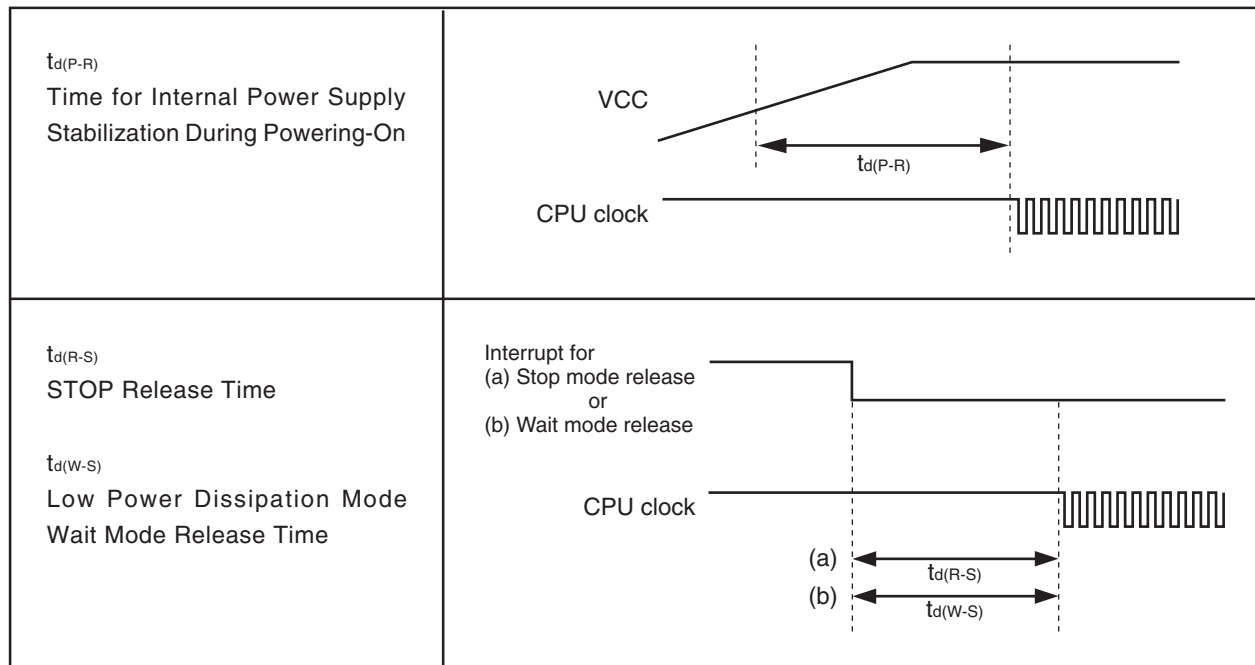
Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute Accuracy				1.0	%
t <sub>su</sub>	Setup Time				3	μs
R <sub>O</sub>	Output Resistance		4	10	20	kΩ
I <sub>VREF</sub>	Reference Power Supply Input Current	(NOTE 2)			1.5	mA

## NOTES:

1. Referenced to VCC = AVCC = VREF = 3.3 to 5.5V, VSS = AVSS = 0V, -40 to 85°C unless otherwise specified.
2. This applies when using one D/A converter, with the DAi register (i = 0, 1) for the unused D/A converter set to "00h".  
The resistor ladder of the A/D converter is not included. Also, the current I<sub>VREF</sub> always flows even though VREF may have been set to be unconnected by the ADCON1 register.

**Table 22.31 Power Supply Circuit Timing Characteristics**

Symbol	Parameter	Measuring Condition	Standard			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Time for Internal Power Supply Stabilization During Powering-On	VCC = 3.0 to 5.5V			2	ms
$t_{d(R-S)}$	STOP Release Time				150	$\mu$ s
$t_{d(W-S)}$	Low Power Dissipation Mode Wait Mode Release Time				150	$\mu$ s



**Figure 22.13 Power Supply Circuit Timing Diagram**



Table 22.32 Electrical Characteristics (1) <sup>(1)</sup>

VCC = 5V

Symbol	Parameter		Measuring Condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7	I <sub>OH</sub> = -5mA	V <sub>CC</sub> -2.0		V <sub>CC</sub>	V
V <sub>OH</sub>	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7	I <sub>OH</sub> = -200μA	V <sub>CC</sub> -0.3		V <sub>CC</sub>	V
V <sub>OH</sub>	HIGH Output Voltage	XOUT	HIGHPOWER	I <sub>OH</sub> = -1mA	3.0	V <sub>CC</sub>	V
			LOWPOWER	I <sub>OH</sub> = -0.5mA	3.0	V <sub>CC</sub>	V
	HIGH Output Voltage	XCOUT	HIGHPOWER	With no load applied	2.5		V
			LOWPOWER	With no load applied	1.6		V
V <sub>OL</sub>	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	I <sub>OL</sub> = 5mA			2.0	V
V <sub>OL</sub>	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	I <sub>OL</sub> = 200μA			0.45	V
V <sub>OL</sub>	LOW Output Voltage	XOUT	HIGHPOWER	I <sub>OL</sub> = 1mA		2.0	V
			LOWPOWER	I <sub>OL</sub> = 0.5mA		2.0	V
	LOW Output Voltage	XCOUT	HIGHPOWER	With no load applied	0		V
			LOWPOWER	With no load applied	0		V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT5, NMI, ADTRG, CTS0 to CTS2, SCL0 to SCL2, SDA0 to SDA2, CLK0 to CLK3, TA0OUT to TA4OUT, $\overline{\text{KI}}$ 0 to $\overline{\text{KI}}$ 3, RXD0 to RXD2, SIN3		0.2		1.0	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	RESET		0.2		2.5	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	XIN		0.2		0.8	V
I <sub>IH</sub>	HIGH Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	V <sub>I</sub> = 5V			5.0	μA
I <sub>IL</sub>	LOW Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	V <sub>I</sub> = 0V			-5.0	μA
R <sub>PULLUP</sub>	Pull-up Resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7	V <sub>I</sub> = 0V	30	50	170	kΩ
R <sub>FXIN</sub>	Feedback Resistance	XIN			1.5		MΩ
R <sub>FXCIN</sub>	Feedback Resistance	XCIN			15		MΩ
V <sub>RAM</sub>	RAM Retention Voltage		At stop mode	2.0			V

## NOTES:

1. Referenced to VCC = 4.2 to 5.5V, VSS = 0V at Topr = -40 to 85°C, f(BCLK) = 24MHz unless otherwise specified.

**Table 22.33 Electrical Characteristics (2) <sup>(1)</sup>**

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
I <sub>cc</sub>	Power Supply Current (VCC = 3.0 to 5.5V)	Output pins are open and other pins are VSS.	Mask ROM	f(BCLK) = 24MHz, PLL operation, No division		20	36	mA
				On-chip oscillation, No division		1		mA
			Flash Memory	f(BCLK) = 24MHz, PLL operation, No division		22	38	mA
				On-chip oscillation, No division		1.8		mA
			Flash Memory Program	f(BCLK) = 10MHz, VCC = 5V		15		mA
			Flash Memory Erase	f(BCLK) = 10MHz, VCC = 5V		25		mA
			Mask ROM	f(BCLK) = 32kHz, Low power dissipation mode, ROM <sup>(2)</sup>		25		μA
			Flash Memory	f(BCLK) = 32kHz, Low power dissipation mode, RAM <sup>(2)</sup>		25		μA
				f(BCLK) = 32kHz, Low power dissipation mode, Flash memory <sup>(2)</sup>		420		μA
			Mask ROM Flash Memory	On-chip oscillation, Wait mode		50		μA
				f(BCLK) = 32kHz, Wait mode (3), Oscillation capacity High		8.5		μA
				f(BCLK) = 32kHz, Wait mode (3), Oscillation capacity Low		3.0		μA
				Stop mode, T <sub>opr</sub> = 25°C		0.8	3.0	μA

## NOTES:

1. Referenced to VCC = 3.0 to 5.5V, VSS = 0V at T<sub>opr</sub> = -40 to 85°C, f(BCLK) = 24MHz unless otherwise specified.
2. This indicates the memory in which the program to be executed exists.
3. With one timer operated using fC32.

**Timing Requirements****VCC = 5V****(Referenced to VCC = 5V, VSS = 0V, at Topr = –40 to 85°C unless otherwise specified)****Table 22.34 External Clock Input (XIN Input)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c</sub>	External Clock Input Cycle Time	62.5		ns
t <sub>w(H)</sub>	External Clock Input HIGH Pulse Width	25		ns
t <sub>w(L)</sub>	External Clock Input LOW Pulse Width	25		ns
t <sub>r</sub>	External Clock Rise Time		15	ns
t <sub>f</sub>	External Clock Fall Time		15	ns

**Table 22.35 Memory Expansion Mode and Microprocessor Mode**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>ac1</sub> (RD-DB)	Data input access time (for setting with no wait)		(NOTE 1)	ns
t <sub>ac2</sub> (RD-DB)	Data input access time (for setting with wait)		(NOTE 2)	ns
t <sub>ac3</sub> (RD-DB)	Data input access time (when accessing multiplexed bus area)		(NOTE 3)	ns
t <sub>su</sub> (DB-RD)	Data input setup time	40		ns
t <sub>su</sub> (RDY-BCLK)	RDY input setup time	30		ns
t <sub>su</sub> (HOLD-BCLK)	HOLD input setup time	40		ns
t <sub>h</sub> (RD-DB)	Data input hold time	0		ns
t <sub>h</sub> (BCLK-RDY)	RDY input hold time	0		ns
t <sub>h</sub> (BCLK-HOLD)	HOLD input hold time	0		ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]} \quad n \text{ is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 45 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

**Timing Requirements****VCC = 5V****(Referenced to VCC = 5V, VSS = 0V, at Topr = –40 to 85°C unless otherwise specified)****Table 22.36 Timer A Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	100		ns
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	40		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	40		ns

**Table 22.37 Timer A Input (Gating Input in Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	400		ns
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	200		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	200		ns

**Table 22.38 Timer A Input (External Trigger Input in One-shot Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	200		ns
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	100		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	100		ns

**Table 22.39 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	100		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	100		ns

**Table 22.40 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(UP)</sub>	TAiOUT Input Cycle Time	2000		ns
t <sub>w(UPH)</sub>	TAiOUT Input HIGH Pulse Width	1000		ns
t <sub>w(UPL)</sub>	TAiOUT Input LOW Pulse Width	1000		ns
t <sub>su(UP-TIN)</sub>	TAiOUT Input Setup Time	400		ns
t <sub>h(TIN-UP)</sub>	TAiOUT Input Hold Time	400		ns

**Table 22.41 Timer A Input (Two-phase Pulse Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	800		ns
t <sub>su(TAIN-TAOUT)</sub>	TAiOUT Input Setup Time	200		ns
t <sub>su(TAOUT-TAIN)</sub>	TAiIN Input Setup Time	200		ns

**Timing Requirements****VCC = 5V****(Referenced to VCC = 5V, VSS = 0V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.42 Timer B Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on both edges)	80		ns

**Table 22.43 Timer B Input (Pulse Period Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	400		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	200		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	200		ns

**Table 22.44 Timer B Input (Pulse Width Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	400		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	200		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	200		ns

**Table 22.45 A/D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	ADTRG $\bar{}$ Input Cycle Time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	ADTRG $\bar{}$ Input LOW Pulse Width	125		ns

**Table 22.46 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi Input Cycle Time	200		ns
$t_{w(CKH)}$	CLKi Input HIGH Pulse Width	100		ns
$t_{w(CKL)}$	CLKi Input LOW Pulse Width	100		ns
$t_{d(C-Q)}$	TXDi Output Delay Time		80	ns
$t_{h(C-Q)}$	TXDi Hold Time	0		ns
$t_{su(D-C)}$	RXDi Input Setup Time	70		ns
$t_{h(C-D)}$	RXDi Input Hold Time	90		ns

**Table 22.47 External Interrupt INTi Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	INTi Input HIGH Pulse Width	250		ns
$t_{w(INL)}$	INTi Input LOW Pulse Width	250		ns

**Switching Characteristics****VCC = 5V****(Referenced to VCC = 5V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.48 Memory Expansion Mode and Microprocessor Mode (for setting with no wait)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_d(\text{BCLK-AD})$	Address output delay time	Figure 22.14		25	ns
$t_h(\text{BCLK-AD})$	Address output hold time (refers to BCLK)		4		ns
$t_h(\text{RD-AD})$	Address output hold time (refers to RD)		0		ns
$t_h(\text{WR-AD})$	Address output hold time (refers to WR)		(NOTE 1)		ns
$t_d(\text{BCLK-CS})$	Chip select output delay time			25	ns
$t_h(\text{BCLK-CS})$	Chip select output hold time (refers to BCLK)		4		ns
$t_d(\text{BCLK-ALE})$	ALE signal output delay time			15	ns
$t_h(\text{BCLK-ALE})$	ALE signal output hold time		–4		ns
$t_d(\text{BCLK-RD})$	RD signal output delay time			25	ns
$t_h(\text{BCLK-RD})$	RD signal output hold time		0		ns
$t_d(\text{BCLK-WR})$	WR signal output delay time			25	ns
$t_h(\text{BCLK-WR})$	WR signal output hold time		0		ns
$t_d(\text{BCLK-DB})$	Data output delay time (refers to BCLK)			40	ns
$t_h(\text{BCLK-DB})$	Data output hold time (refers to BCLK) <sup>(3)</sup>		4		ns
$t_d(\text{DB-WR})$	Data output delay time (refers to WR)		(NOTE 2)		ns
$t_h(\text{WR-DB})$	Data output hold time (refers to WR) <sup>(3)</sup>		(NOTE 1)		ns
$t_d(\text{BCLK-HLDA})$	HLDA output delay time			40	ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]} \quad f(\text{BCLK}) \text{ is 12.5 MHz or less.}$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

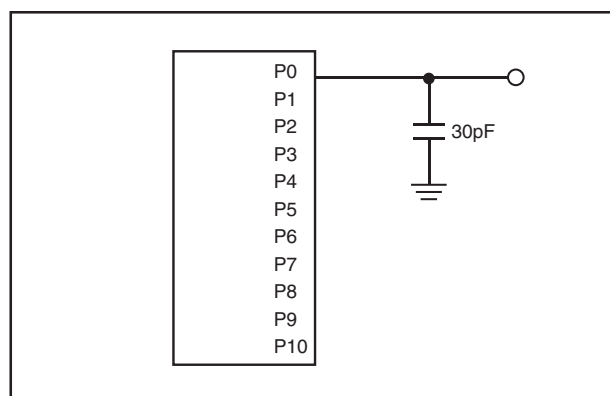
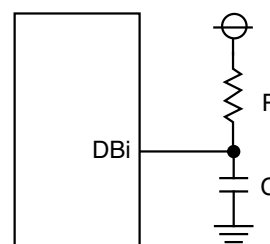
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when  $V_{OL} = 0.2 V_{CC}$ ,  $C = 30 \text{ pF}$ ,

$R = 1 \text{ k}\Omega$ , hold time of output “L” level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$

**Figure 22.14 Port P0 to P10 Measurement Circuit**

**Switching Characteristics****VCC = 5V****(Referenced to VCC = 5V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.49 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 22.14		25	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (refers to RD)		0		ns
t <sub>h</sub> (WR-AD)	Address output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			25	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time			15	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time		–4		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			25	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			25	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (refers to BCLK)			40	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (refers to BCLK) <sup>(3)</sup>		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (refers to WR)		(NOTE 2)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (refers to WR) <sup>(3)</sup>		(NOTE 1)		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time			40	ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]}$$

n is “1” for 1-wait setting, “2” for 2-wait setting and “3” for 3-wait setting.  
When n = 1, f(BCLK) is 12.5 MHz or less.

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

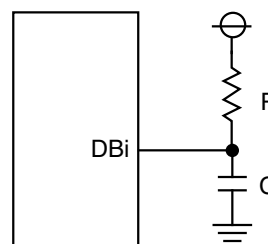
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when  $V_{OL} = 0.2 V_{CC}$ ,  $C = 30 \text{ pF}$ ,

$R = 1 \text{ k}\Omega$ , hold time of output “L” level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$



**Switching Characteristics****VCC = 5V****(Referenced to VCC = 5V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.50 Memory Expansion Mode and Microprocessor Mode  
(for 2- to 3-wait setting, external area access and multiplexed bus selection)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 22.14		25	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (refers to RD)		(NOTE 1)		ns
t <sub>h</sub> (WR-AD)	Address output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			25	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-CS)	Chip select output hold time (refers to RD)		(NOTE 1)		ns
t <sub>h</sub> (WR-CS)	Chip select output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			25	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			25	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (refers to BCLK)			40	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (refers to WR)		(NOTE 2)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time			40	ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time (refers to BCLK)			15	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time (refers to BCLK)		–4		ns
t <sub>d</sub> (AD-ALE)	ALE signal output delay time (refers to Address)		(NOTE 3)		ns
t <sub>h</sub> (ALE-AD)	ALE signal output hold time (refers to Address)		(NOTE 4)		ns
t <sub>d</sub> (AD-RD)	RD signal output delay from the end of Address		0		ns
t <sub>d</sub> (AD-WR)	WR signal output delay from the end of Address		0		ns
t <sub>dZ</sub> (RD-AD)	Address output floating start time			8	ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 25 \text{ [ns]}$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 15 \text{ [ns]}$$



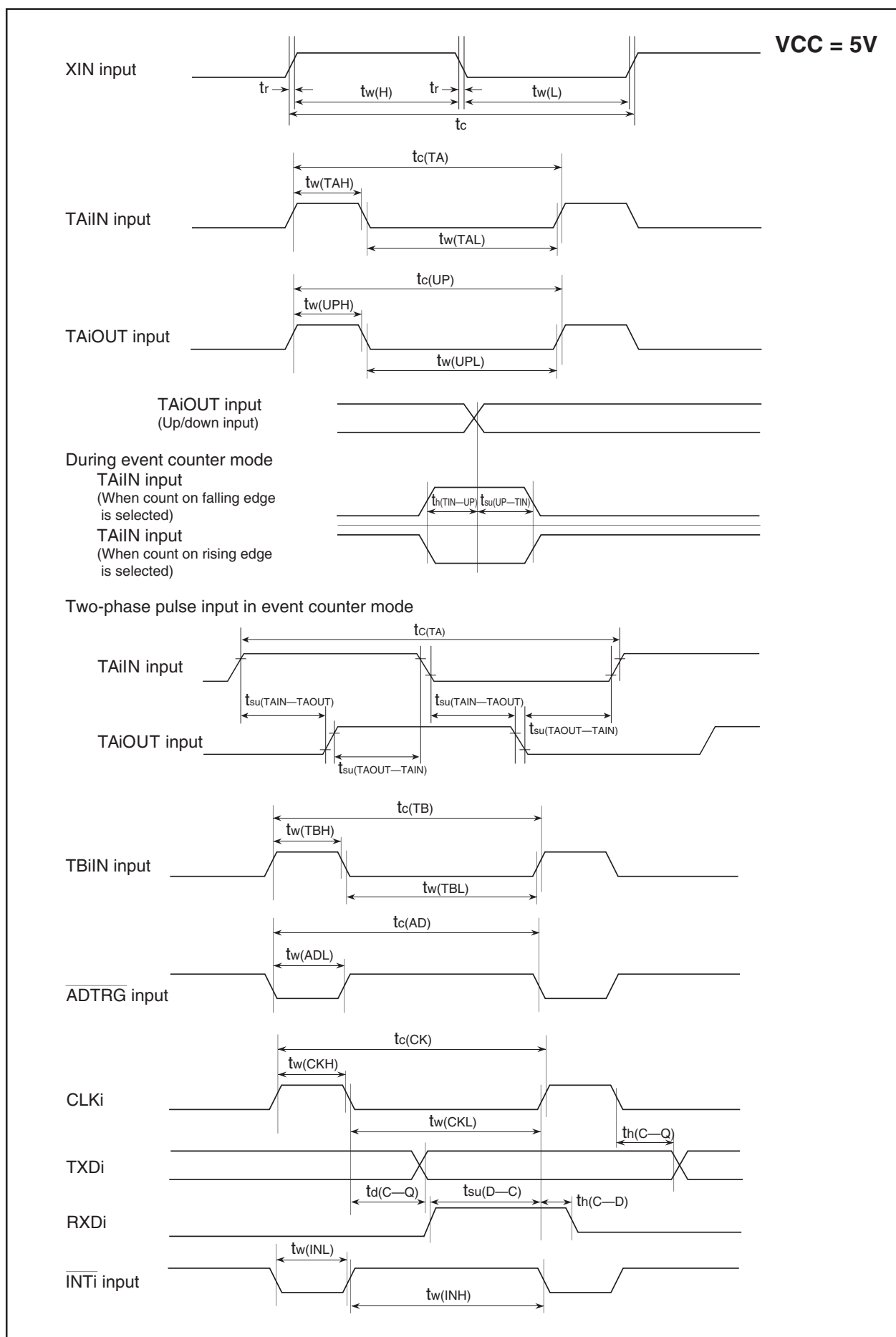
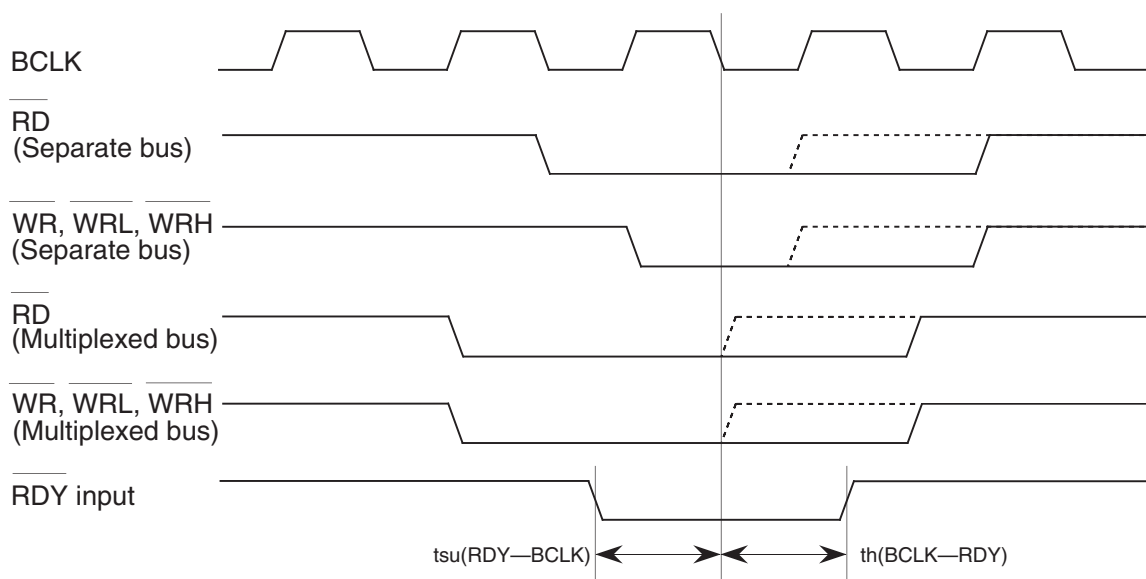


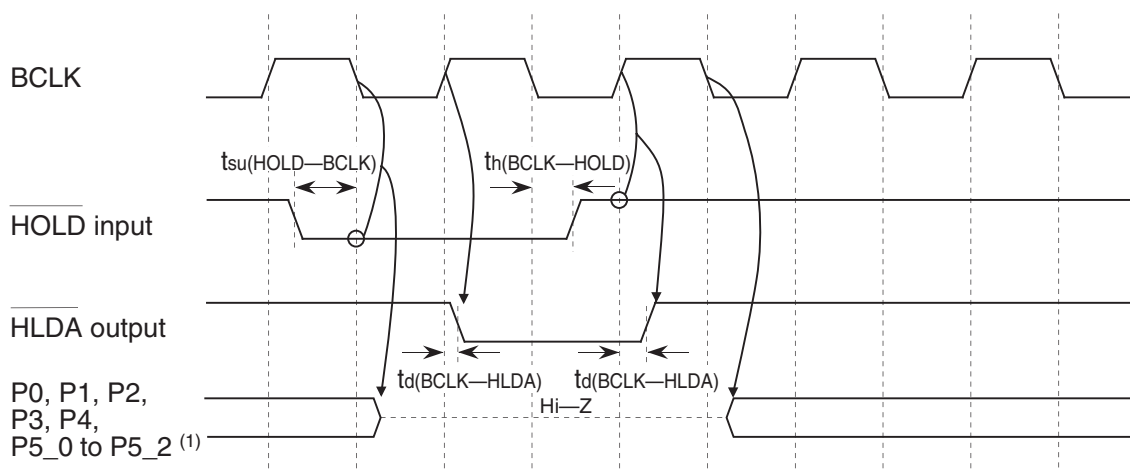
Figure 22.15 Timing Diagram (1)

**Memory Expansion Mode and Microprocessor Mode****VCC = 5V**

(Effective for setting with wait)



(Common to setting with wait and setting without wait)

**NOTE:**

1. The above pins are set to high-impedance regardless of the input level of the BYTE pin, the PM06 bit in the PM0 register and the PM11 bit in the PM1 register.

**Measuring conditions :**

- VCC = 5 V
- Input timing voltage : Determined with  $V_{IL} = 1.0 \text{ V}$ ,  $V_{IH} = 4.0 \text{ V}$
- Output timing voltage: Determined with  $V_{OL} = 2.5 \text{ V}$ ,  $V_{OH} = 2.5 \text{ V}$

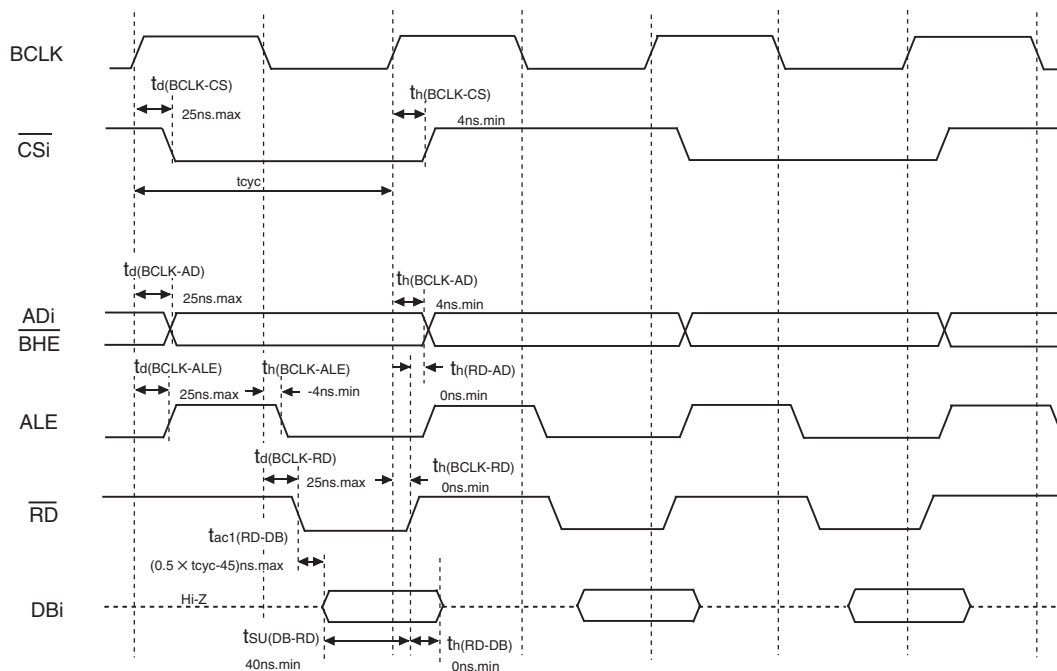
**Figure 22.16 Timing Diagram (2)**

## Memory Expansion Mode and Microprocessor Mode

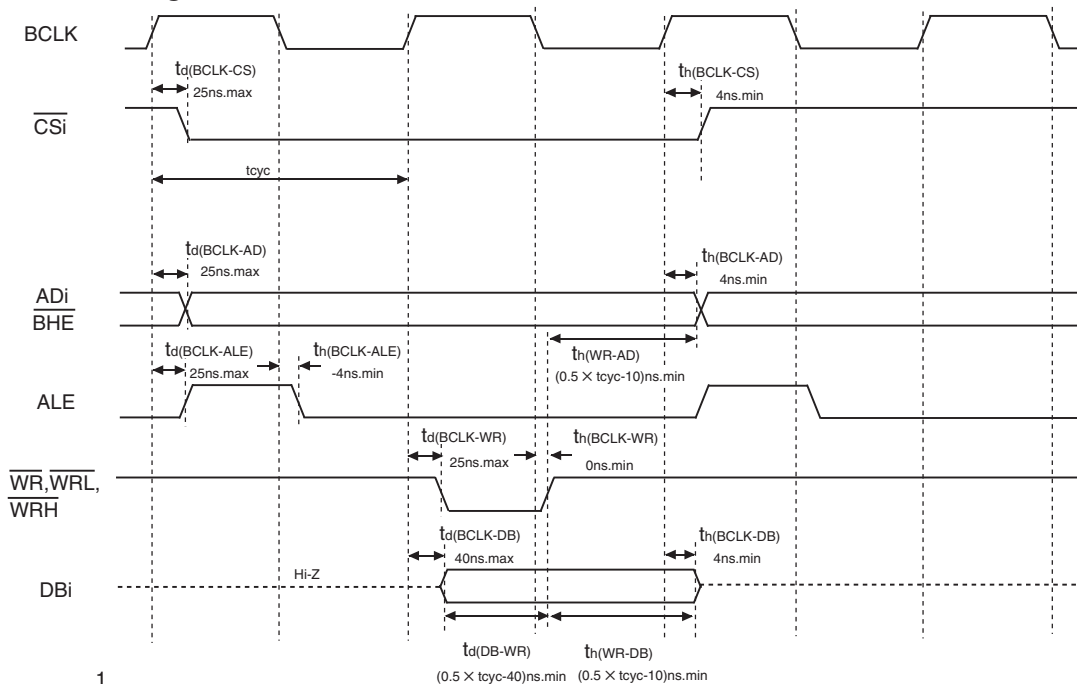
(For setting with no wait)

VCC = 5V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions :

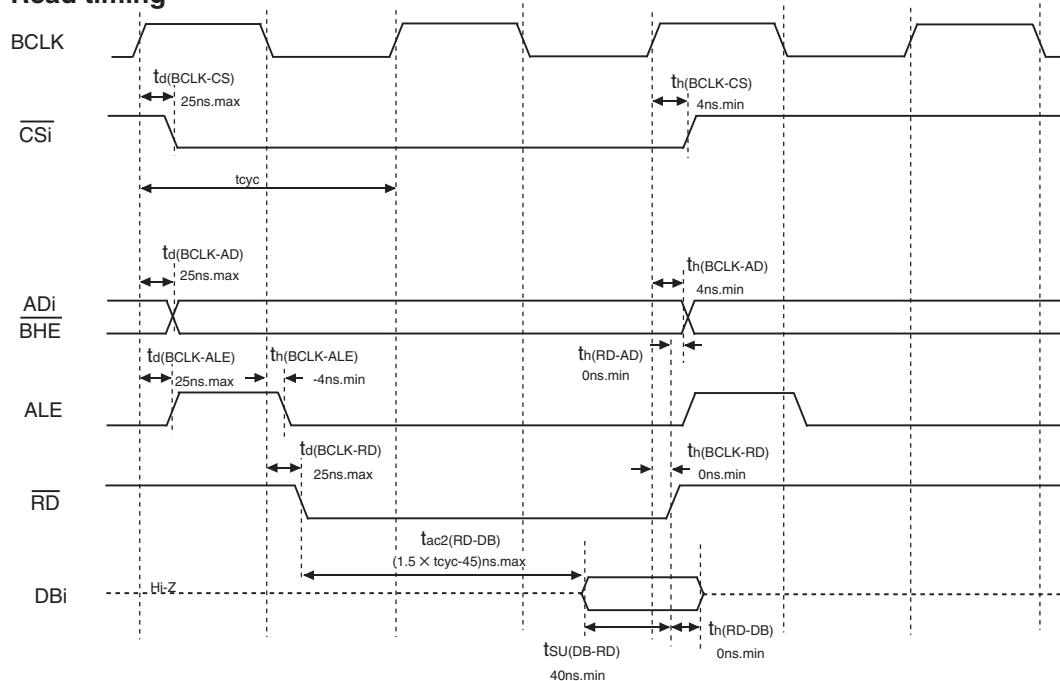
- VCC = 5 V
- Input timing voltage :  $V_{IL} = 0.8 \text{ V}$ ,  $V_{IH} = 2.0 \text{ V}$
- Output timing voltage :  $V_{OL} = 0.4 \text{ V}$ ,  $V_{OH} = 2.4 \text{ V}$

Figure 22.17 Timing Diagram (3)

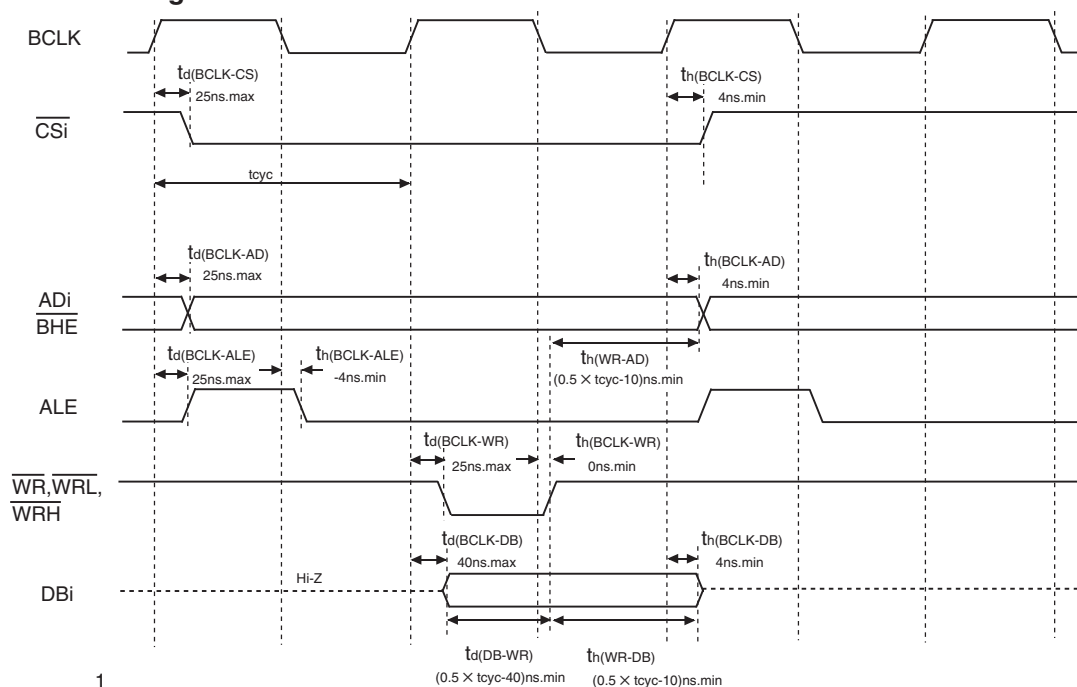
### Memory Expansion Mode and Microprocessor Mode (For 1-wait setting and external area access)

VCC = 5V

#### Read timing



#### Write timing



$$t_{\text{cyc}} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions :

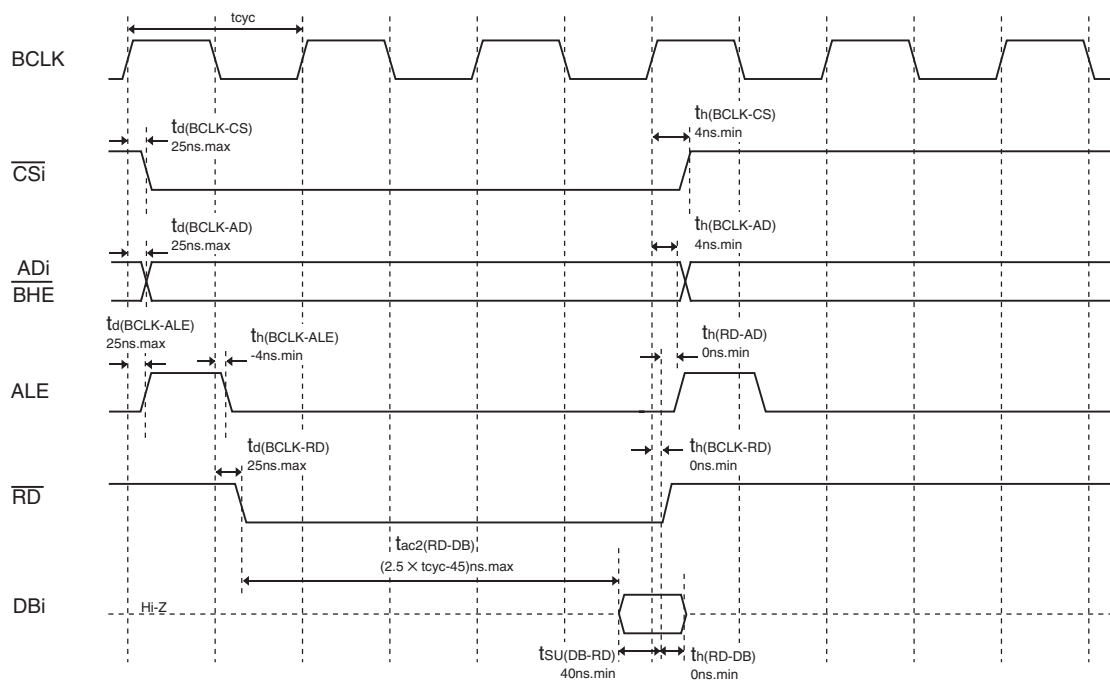
- VCC = 5 V
- Input timing voltage :  $V_{\text{IL}} = 0.8 \text{ V}$ ,  $V_{\text{IH}} = 2.0 \text{ V}$
- Output timing voltage :  $V_{\text{OL}} = 0.4 \text{ V}$ ,  $V_{\text{OH}} = 2.4 \text{ V}$

Figure 22.18 Timing Diagram (4)

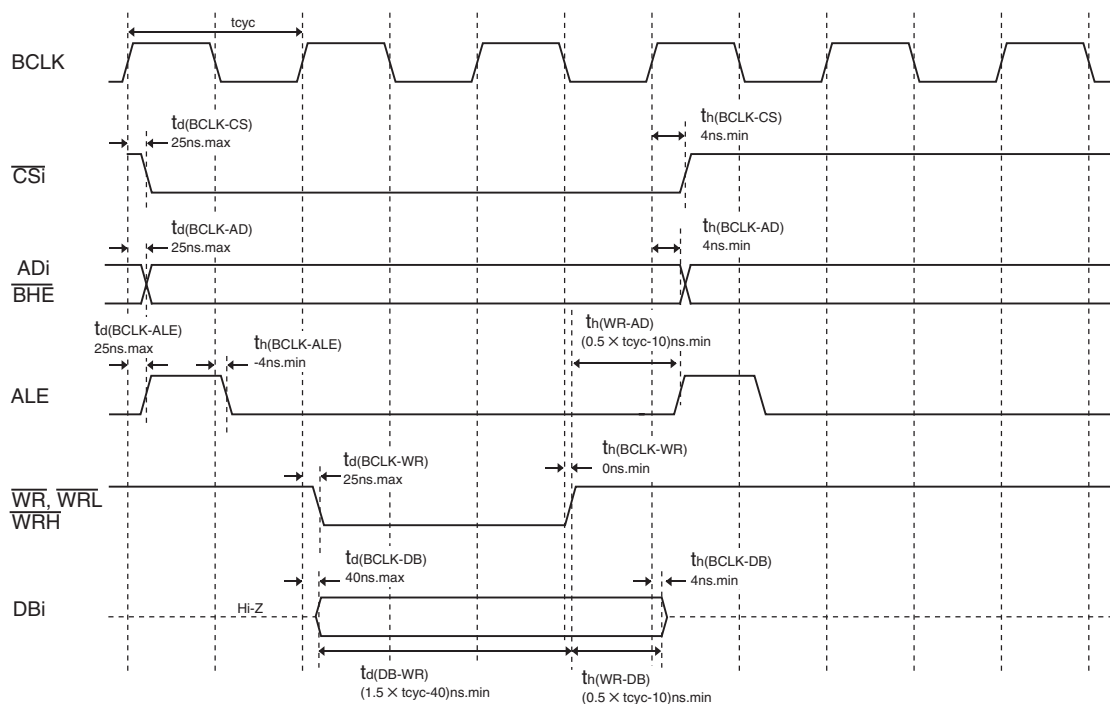
## Memory Expansion Mode and Microprocessor Mode (For 2-wait setting and external area access)

VCC = 5V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

Measuring conditions :

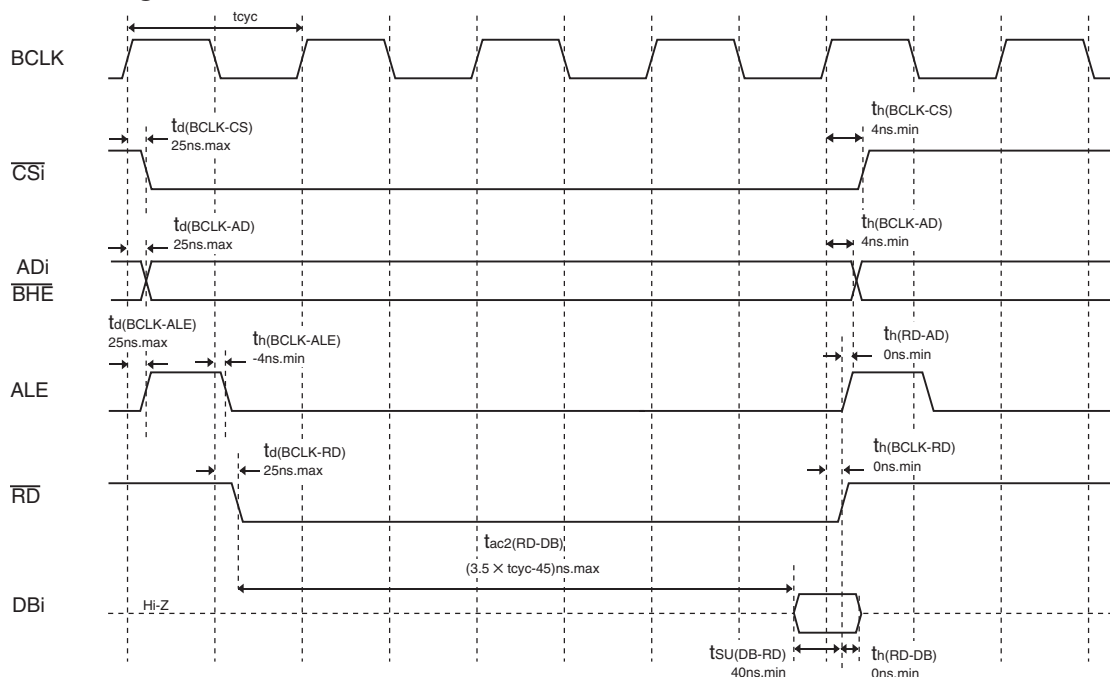
- VCC = 5 V
- Input timing voltage :  $V_{IL} = 0.8$  V,  $V_{IH} = 2.0$  V
- Output timing voltage :  $V_{OL} = 0.4$  V,  $V_{OH} = 2.4$  V

Figure 22.19 Timing Diagram (5)

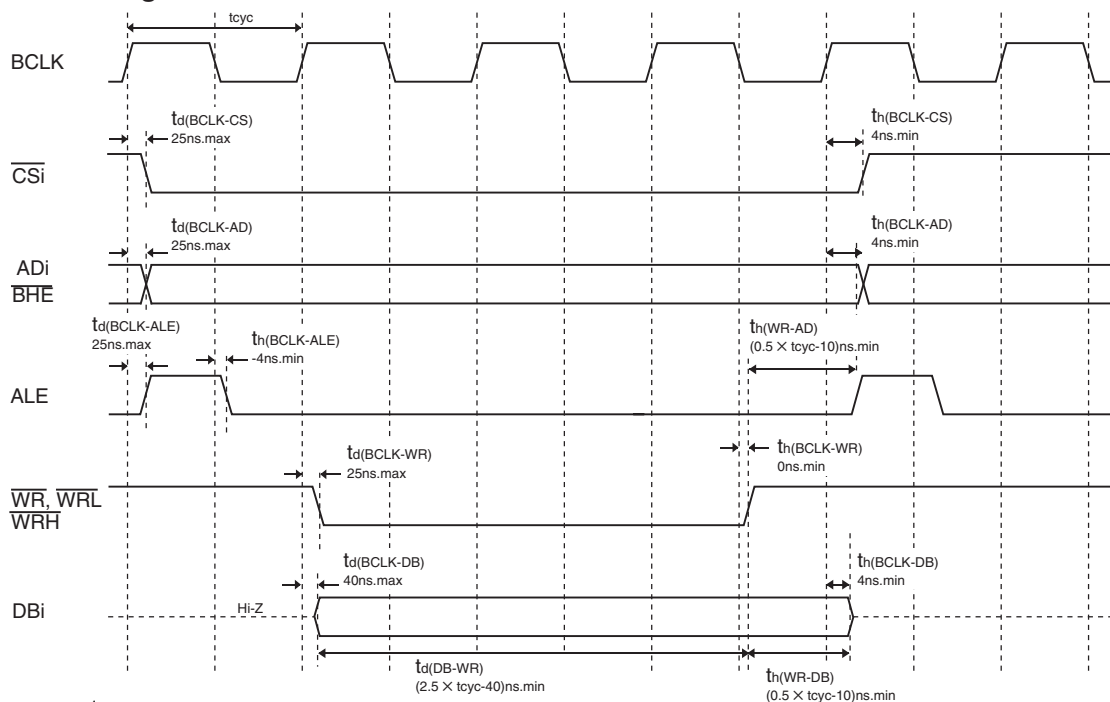
## Memory Expansion Mode and Microprocessor Mode (For 3-wait setting and external area access)

VCC = 5V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

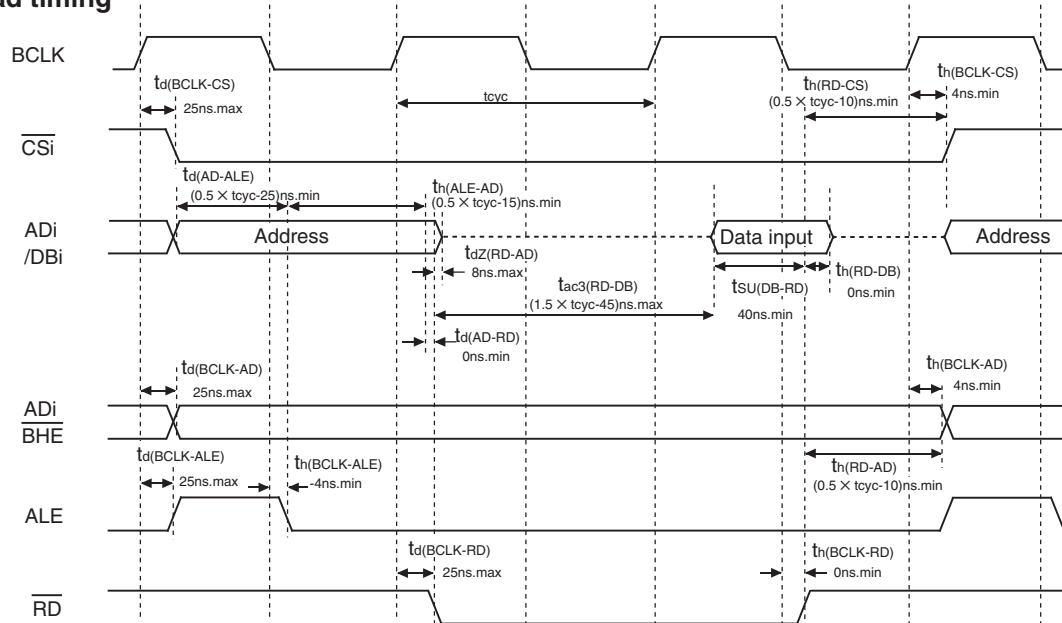
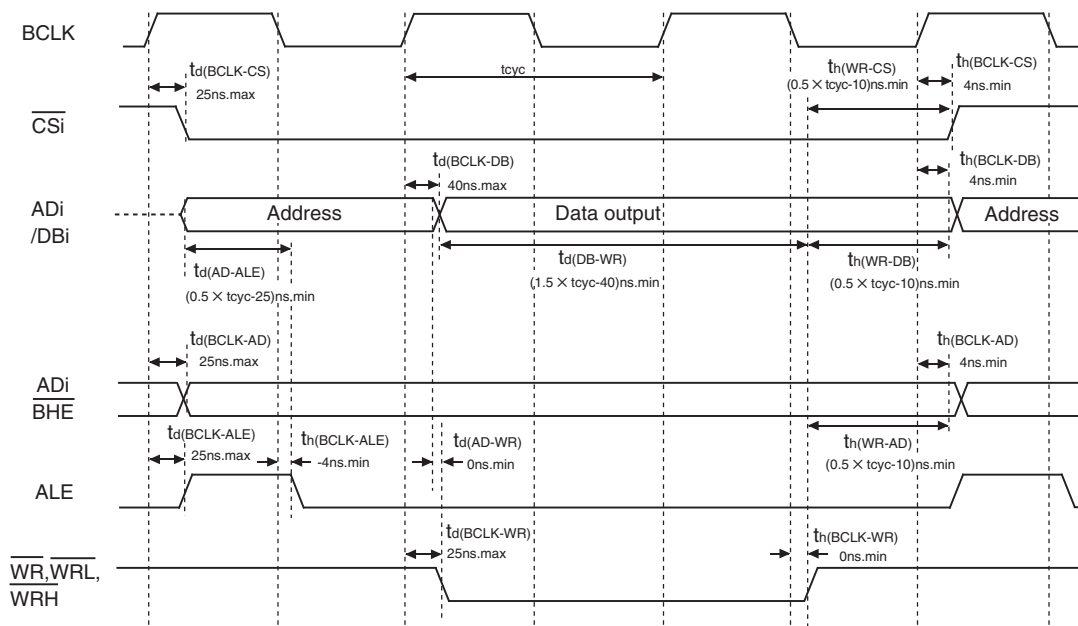
Measuring conditions :

- VCC = 5 V
- Input timing voltage :  $V_{IL} = 0.8 V$ ,  $V_{IH} = 2.0 V$
- Output timing voltage :  $V_{OL} = 0.4 V$ ,  $V_{OH} = 2.4 V$

Figure 22.20 Timing Diagram (6)

**Memory Expansion Mode and Microprocessor Mode****VCC = 5V**

(For 1- or 2-wait setting, external area access and multiplexed bus selection)

**Read timing****Write timing**

$$t_{\text{cyc}} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions :

- VCC = 5 V
- Input timing voltage :  $V_{\text{IL}} = 0.8 \text{ V}$ ,  $V_{\text{IH}} = 2.0 \text{ V}$
- Output timing voltage :  $V_{\text{OL}} = 0.4 \text{ V}$ ,  $V_{\text{OH}} = 2.4 \text{ V}$

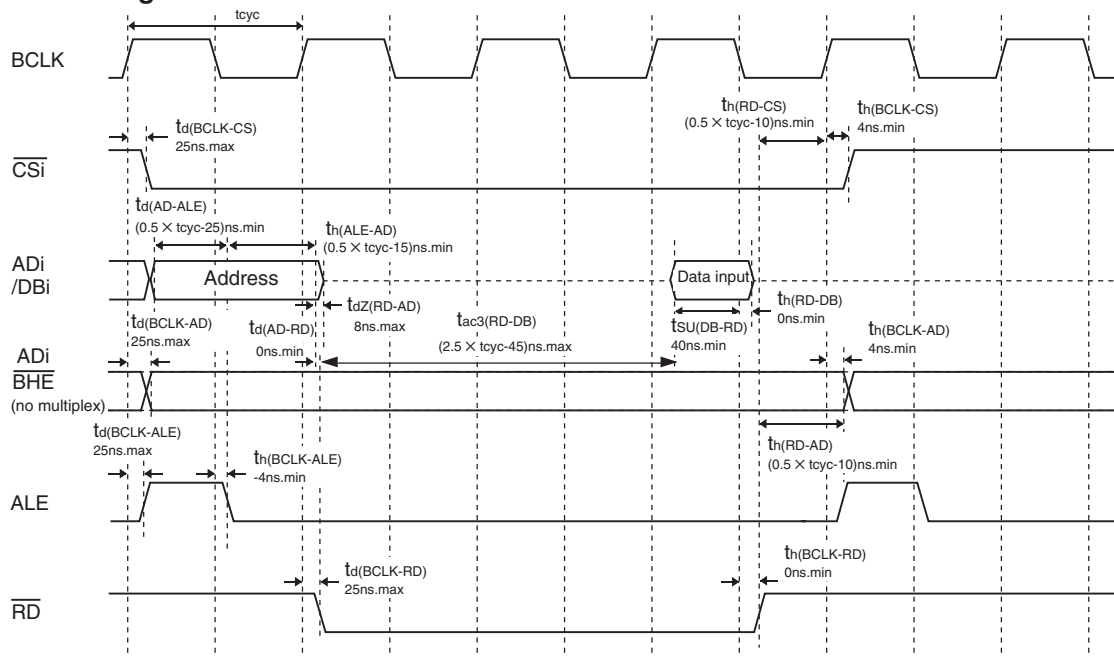
**Figure 22.21 Timing Diagram (7)**

## Memory Expansion Mode and Microprocessor Mode

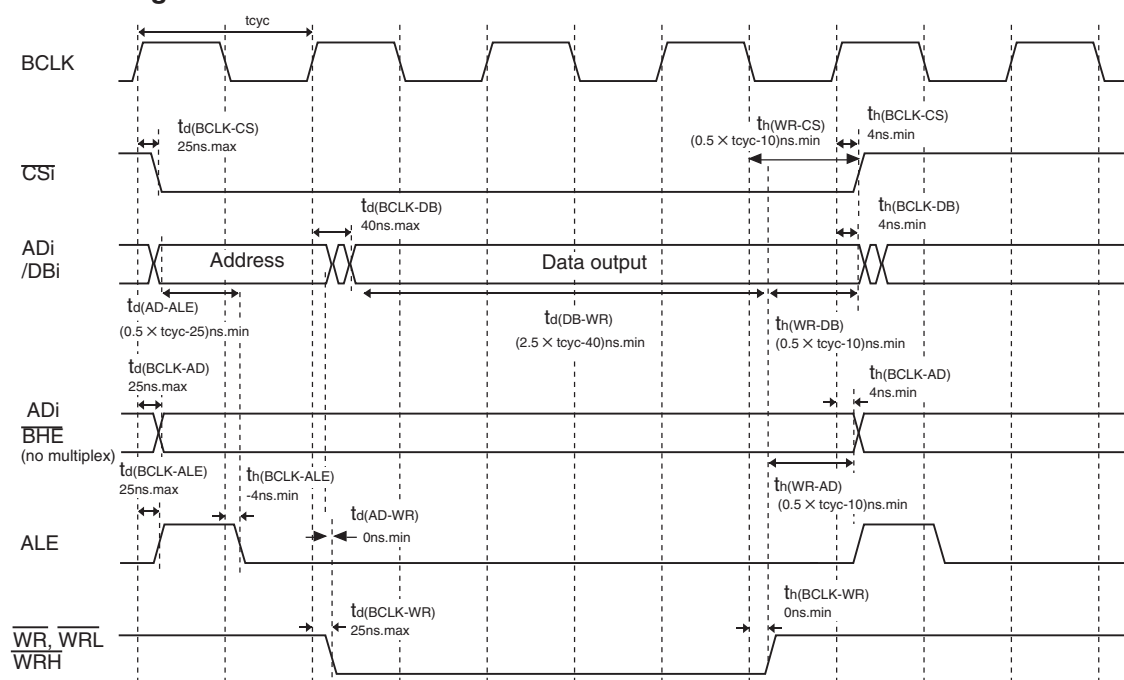
(For 3-wait setting, external area access and multiplexed bus selection)

VCC = 5V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

Measuring conditions :

- VCC = 5 V
- Input timing voltage :  $V_{IL} = 0.8 \text{ V}$ ,  $V_{IH} = 2.0 \text{ V}$
- Output timing voltage :  $V_{OL} = 0.4 \text{ V}$ ,  $V_{OH} = 2.4 \text{ V}$

Figure 22.22 Timing Diagram (8)



Table 22.51 Electrical Characteristics <sup>(1)</sup>

VCC = 3.3V

Symbol	Parameter		Measuring Condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7	I <sub>OH</sub> = -1mA	V <sub>CC</sub> -0.5		V <sub>CC</sub>	V
V <sub>OH</sub>	HIGH Output Voltage	XOUT	HIGHPOWER	I <sub>OH</sub> = -0.1mA	V <sub>CC</sub> -0.5	V <sub>CC</sub>	V
			LOWPOWER	I <sub>OH</sub> = -50μA	V <sub>CC</sub> -0.5	V <sub>CC</sub>	
	HIGH Output Voltage	XCOUT	HIGHPOWER	With no load applied	2.5		V
			LOWPOWER	With no load applied	1.6		
V <sub>OL</sub>	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7	I <sub>OL</sub> = 1mA			0.5	V
V <sub>OL</sub>	LOW Output Voltage	XOUT	HIGHPOWER	I <sub>OL</sub> = 0.1mA		0.5	V
			LOWPOWER	I <sub>OL</sub> = 50μA		0.5	
	LOW Output Voltage	XCOUT	HIGHPOWER	With no load applied	0		V
			LOWPOWER	With no load applied	0		
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT5, NMI, ADTRG, CTS0 to CTS2, SCL0 to SCL2, SDA0 to SDA2, CLK0 to CLK3, TA0OUT to TA4OUT, KI0 to KI3, RXD0 to RXD2, SIN3		0.2		0.8	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	RESET		0.2		1.8	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	XIN		0.2		0.8	V
I <sub>IH</sub>	HIGH Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	V <sub>I</sub> = 3.3V			4.0	μA
I <sub>IL</sub>	LOW Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, RESET, CNVSS, BYTE	V <sub>I</sub> = 0V			-4.0	μA
R <sub>PULLUP</sub>	Pull-up Resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7	V <sub>I</sub> = 0V	50	100	500	kΩ
R <sub>IXIN</sub>	Feedback Resistance	XIN			3.0		MΩ
R <sub>IXCIN</sub>	Feedback Resistance	XCIN			25		MΩ
V <sub>RAM</sub>	RAM Retention Voltage		At stop mode	2.0			V

## NOTES:

1. Referenced to VCC = 3.0 to 3.6V, VSS = 0V at Topr = -40 to 85°C, f(BCLK) = 24MHz unless otherwise specified.

**Timing Requirements****VCC = 3.3V****(Referenced to VCC = 3.3V, VSS = 0V, at Topr = –40 to 85°C unless otherwise specified)****Table 22.52 External Clock Input (XIN Input)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c</sub>	External Clock Input Cycle Time	62.5		ns
t <sub>w(H)</sub>	External Clock Input HIGH Pulse Width	25		ns
t <sub>w(L)</sub>	External Clock Input LOW Pulse Width	25		ns
t <sub>r</sub>	External Clock Rise Time		15	ns
t <sub>f</sub>	External Clock Fall Time		15	ns

**Table 22.53 Memory Expansion Mode and Microprocessor Mode**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>ac1</sub> (RD-DB)	Data input access time (for setting with no wait)		(NOTE 1)	ns
t <sub>ac2</sub> (RD-DB)	Data input access time (for setting with wait)		(NOTE 2)	ns
t <sub>ac3</sub> (RD-DB)	Data input access time (when accessing multiplexed bus area)		(NOTE 3)	ns
t <sub>su</sub> (DB-RD)	Data input setup time	50		ns
t <sub>su</sub> (RDY-BCLK)	RDY input setup time	40		ns
t <sub>su</sub> (HOLD-BCLK)	HOLD input setup time	50		ns
t <sub>h</sub> (RD-DB)	Data input hold time	0		ns
t <sub>h</sub> (BCLK-RDY)	RDY input hold time	0		ns
t <sub>h</sub> (BCLK-HOLD)	HOLD input hold time	0		ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 60 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 60 \text{ [ns]} \quad n \text{ is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 60 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

**Timing Requirements****VCC = 3.3V****(Referenced to VCC = 3.3V, VSS = 0V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.54 Timer A Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	150		ns
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	60		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	60		ns

**Table 22.55 Timer A Input (Gating Input in Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	600		ns
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	300		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	300		ns

**Table 22.56 Timer A Input (External Trigger Input in One-shot Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	300		ns
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	150		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	150		ns

**Table 22.57 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>w(TAH)</sub>	TAiIN Input HIGH Pulse Width	150		ns
t <sub>w(TAL)</sub>	TAiIN Input LOW Pulse Width	150		ns

**Table 22.58 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(UP)</sub>	TAiOUT Input Cycle Time	3000		ns
t <sub>w(UPH)</sub>	TAiOUT Input HIGH Pulse Width	1500		ns
t <sub>w(UPL)</sub>	TAiOUT Input LOW Pulse Width	1500		ns
t <sub>su(UP-TIN)</sub>	TAiOUT Input Setup Time	600		ns
t <sub>h(TIN-UP)</sub>	TAiOUT Input Hold Time	600		ns

**Table 22.59 Timer A Input (Two-phase Pulse Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TA)</sub>	TAiIN Input Cycle Time	2		μs
t <sub>su(TAIN-TAOUT)</sub>	TAiOUT Input Setup Time	500		ns
t <sub>su(TAOUT-TAIN)</sub>	TAiIN Input Setup Time	500		ns

**Timing Requirements****VCC = 3.3V****(Referenced to VCC = 3.3V, VSS = 0V, at Topr = -40 to 85°C unless otherwise specified)****Table 22.60 Timer B Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on one edge)	60		ns
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on both edges)	120		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on both edges)	120		ns

**Table 22.61 Timer B Input (Pulse Period Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	600		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	300		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	300		ns

**Table 22.62 Timer B Input (Pulse Width Measurement Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	600		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	300		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	300		ns

**Table 22.63 A/D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	ADTRG $\bar{}$ Input Cycle Time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	ADTRG $\bar{}$ Input LOW Pulse Width	200		ns

**Table 22.64 Serial Interface**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi Input Cycle Time	300		ns
$t_{w(CKH)}$	CLKi Input HIGH Pulse Width	150		ns
$t_{w(CKL)}$	CLKi Input LOW Pulse Width	150		ns
$t_{d(C-Q)}$	TXDi Output Delay Time		160	ns
$t_{h(C-Q)}$	TXDi Hold Time	0		ns
$t_{su(D-C)}$	RXDi Input Setup Time	100		ns
$t_{h(C-D)}$	RXDi Input Hold Time	90		ns

**Table 22.65 External Interrupt INTi Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	INTi Input HIGH Pulse Width	380		ns
$t_{w(INL)}$	INTi Input LOW Pulse Width	380		ns

**Switching Characteristics****VCC = 3.3V****(Referenced to VCC = 3.3V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.66 Memory Expansion Mode and Microprocessor Mode (for setting with no wait)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 22.23		30	ns
$t_{h(BCLK-AD)}$	Address output hold time (refers to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (refers to RD)		0		ns
$t_{h(WR-AD)}$	Address output hold time (refers to WR)		(NOTE 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			30	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (refers to BCLK)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			30	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			30	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (refers to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (refers to BCLK) <sup>(3)</sup>		4		ns
$t_{d(DB-WR)}$	Data output delay time (refers to WR)		(NOTE 2)		ns
$t_{h(WR-DB)}$	Data output hold time (refers to WR) <sup>(3)</sup>		(NOTE 1)		ns
$t_{d(BCLK-HLDA)}$	HLDA output delay time			40	ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 40 \text{ [ns]} \quad f(BCLK) \text{ is 12.5 MHz or less.}$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

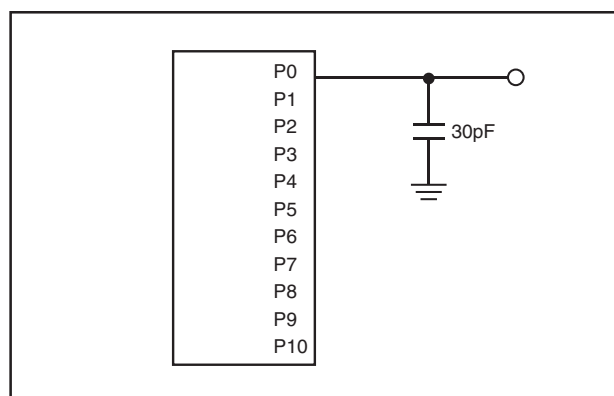
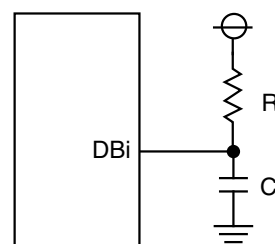
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when  $V_{OL} = 0.2 V_{CC}$ ,  $C = 30 \text{ pF}$ ,

$R = 1 \text{ k}\Omega$ , hold time of output “L” level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$

**Figure 22.23 Port P0 to P10 Measurement Circuit**

**Switching Characteristics****VCC = 3.3V****(Referenced to VCC = 3.3V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.67 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 22.23		30	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (refers to RD)		0		ns
t <sub>h</sub> (WR-AD)	Address output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			30	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time			25	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time		–4		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			30	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			30	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (refers to BCLK)			40	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (refers to BCLK) <sup>(3)</sup>		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (refers to WR)		(NOTE 2)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (refers to WR) <sup>(3)</sup>		(NOTE 1)		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time			40	ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]}$$

n is “1” for 1-wait setting, “2” for 2-wait setting and “3” for 3-wait setting.  
When n = 1, f(BCLK) is 12.5 MHz or less.

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.

Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in

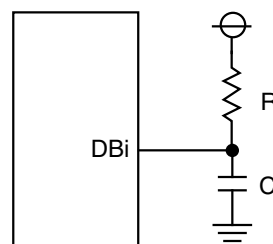
$$t = -CR \times \ln(1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.

For example, when  $V_{OL} = 0.2 V_{CC}$ ,  $C = 30 \text{ pF}$ ,

$R = 1 \text{ k}\Omega$ , hold time of output “L” level is

$$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns.}$$



**Switching Characteristics****VCC = 3.3V****(Referenced to VCC = 3.3V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)****Table 22.68 Memory Expansion Mode and Microprocessor Mode  
(for 2- to 3-wait setting, external area access and multiplexed bus selection)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 22.23		50	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (refers to RD)		(NOTE 1)		ns
t <sub>h</sub> (WR-AD)	Address output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			50	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-CS)	Chip select output hold time (refers to RD)		(NOTE 1)		ns
t <sub>h</sub> (WR-CS)	Chip select output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			40	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			40	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (refers to BCLK)			50	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (refers to WR)		(NOTE 2)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (refers to WR)		(NOTE 1)		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time			40	ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time (refers to BCLK)			25	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time (refers to BCLK)		–4		ns
t <sub>d</sub> (AD-ALE)	ALE signal output delay time (refers to Address)		(NOTE 3)		ns
t <sub>h</sub> (ALE-AD)	ALE signal output hold time (refers to Address)		(NOTE 4)		ns
t <sub>d</sub> (AD-RD)	RD signal output delay from the end of Address		0		ns
t <sub>d</sub> (AD-WR)	WR signal output delay from the end of Address		0		ns
t <sub>dZ</sub> (RD-AD)	Address output floating start time			8	ns

**NOTES:**

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(\text{BCLK})} - 50 \text{ [ns]} \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \text{ [ns]}$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 15 \text{ [ns]}$$

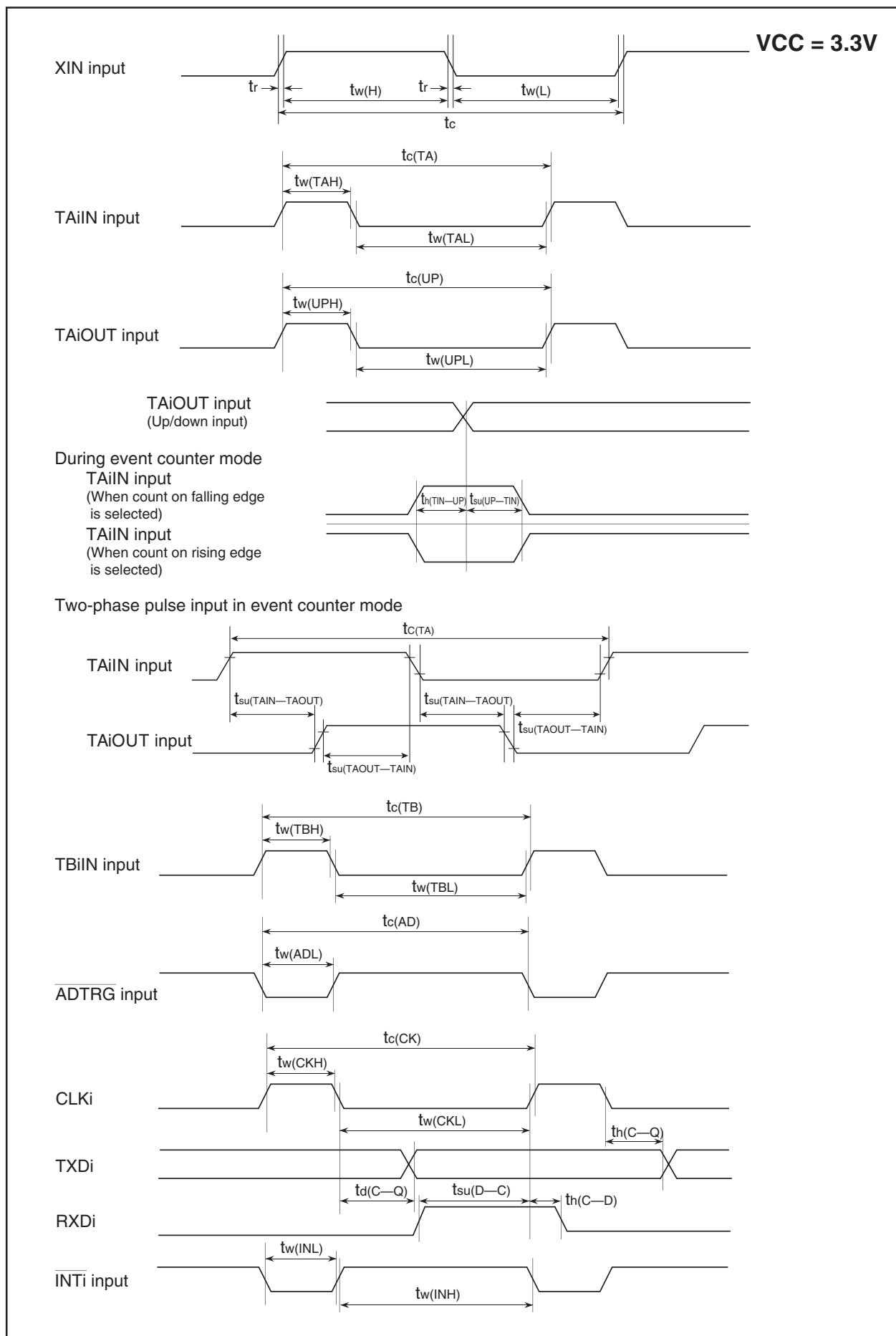
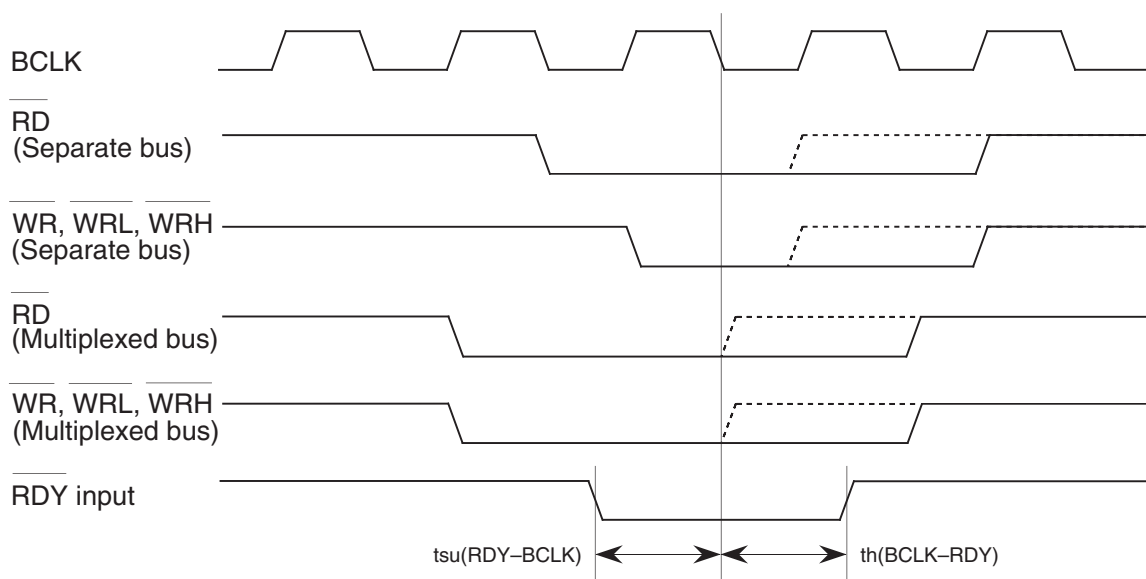


Figure 22.24 Timing Diagram (1)

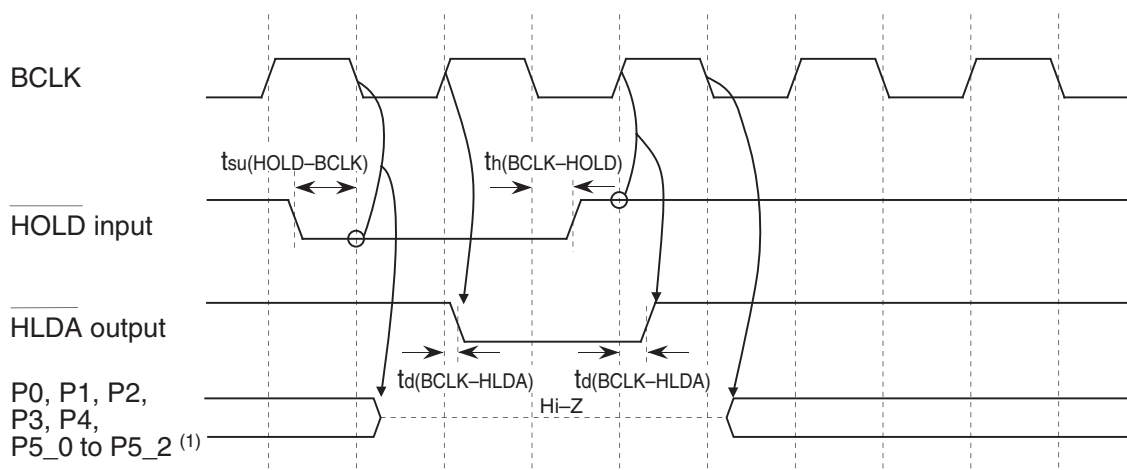


**Memory Expansion Mode and Microprocessor Mode****VCC = 3.3V**

(Effective for setting with wait)



(Common to setting with wait and setting without wait)

**NOTE:**

1. The above pins are set to high-impedance regardless of the input level of the BYTE pin, the PM06 bit in the PM0 register and the PM11 bit in the PM1 register.

**Measuring conditions :**

- VCC = 3.3 V
- Input timing voltage : Determined with  $V_{IL} = 0.6\text{ V}$ ,  $V_{IH} = 2.7\text{ V}$
- Output timing voltage: Determined with  $V_{OL} = 1.65\text{ V}$ ,  $V_{OH} = 1.65\text{ V}$

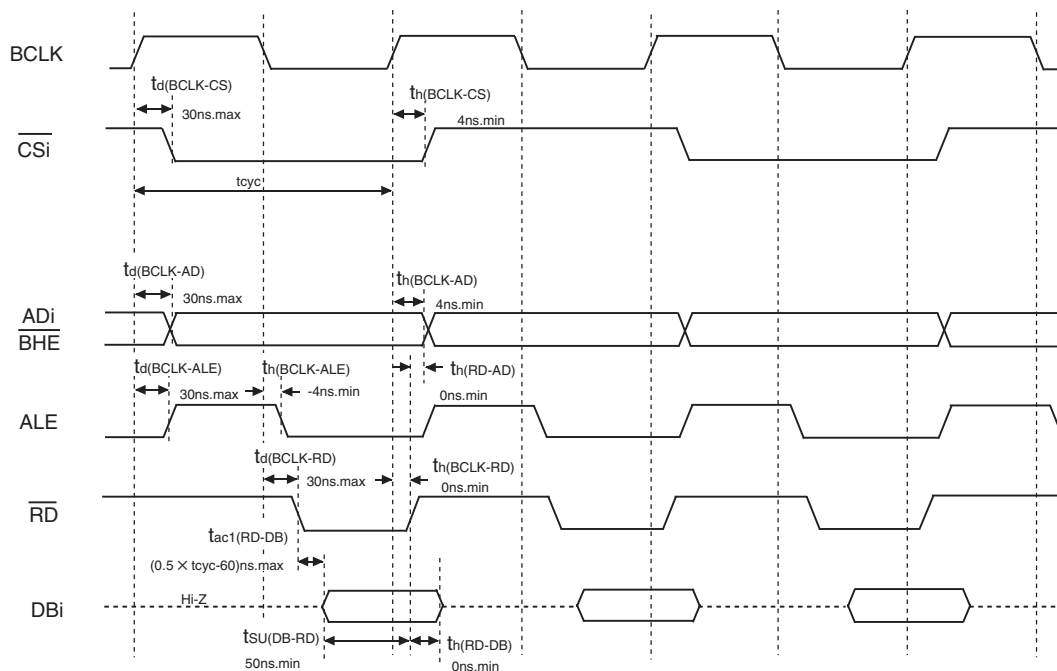
**Figure 22.25 Timing Diagram (2)**

## Memory Expansion Mode and Microprocessor Mode

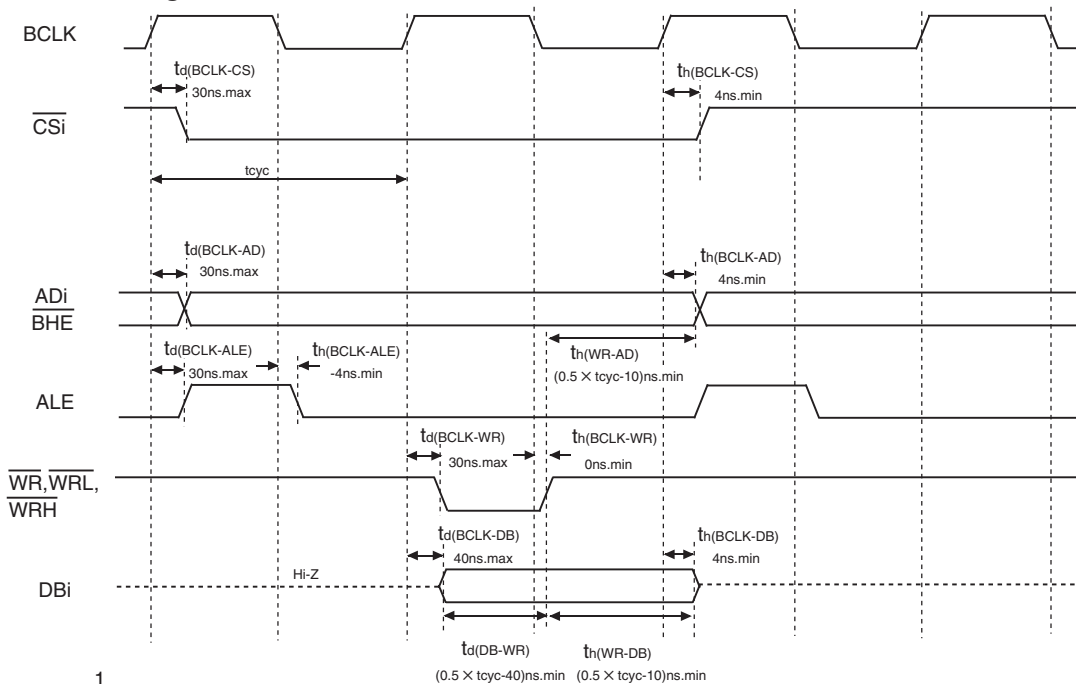
(For setting with no wait)

VCC = 3.3V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions :

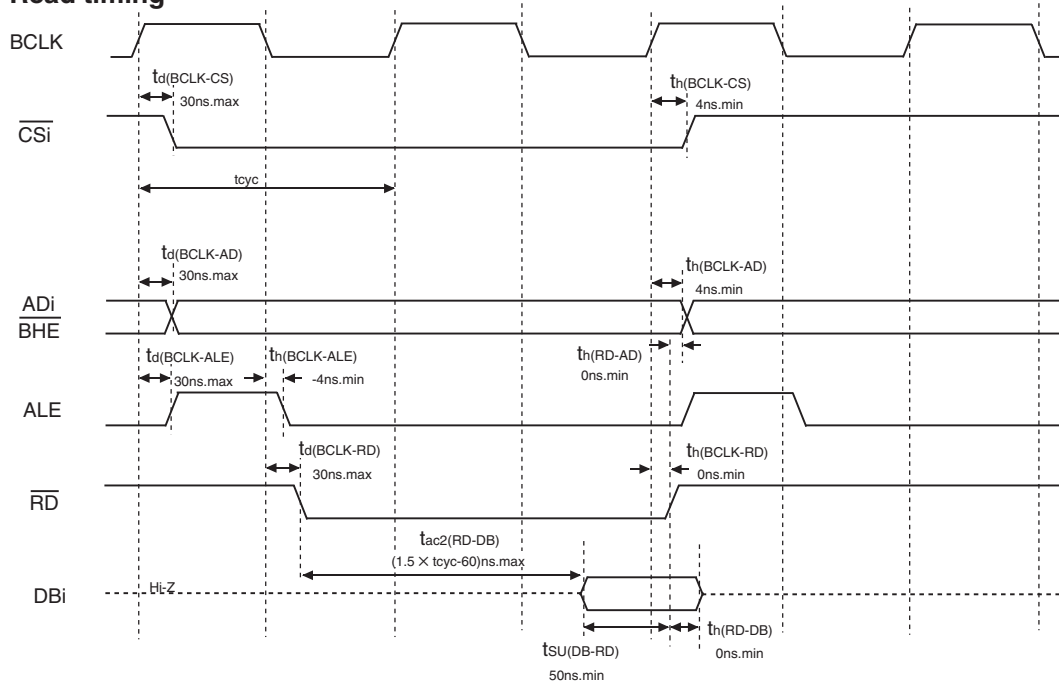
- VCC = 3.3 V
- Input timing voltage :  $V_{IL} = 0.6 \text{ V}$ ,  $V_{IH} = 2.7 \text{ V}$
- Output timing voltage :  $V_{OL} = 1.65 \text{ V}$ ,  $V_{OH} = 1.65 \text{ V}$

Figure 22.26 Timing Diagram (3)

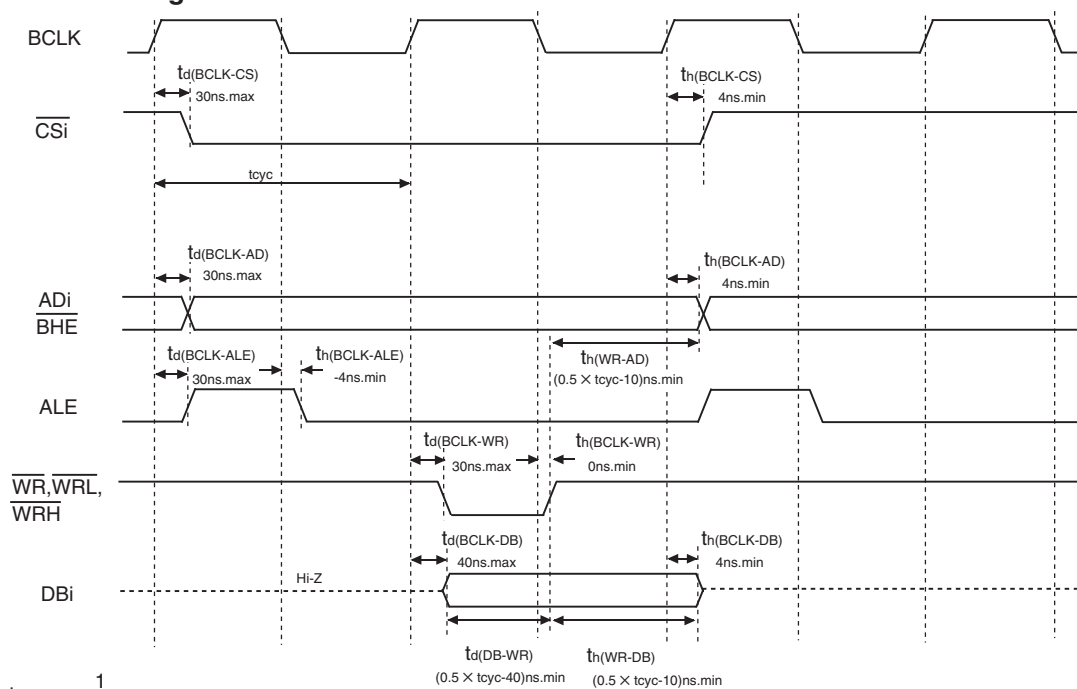
### Memory Expansion Mode and Microprocessor Mode (For 1-wait setting and external area access)

VCC = 3.3V

#### Read timing



#### Write timing



$$t_{\text{cyc}} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions :

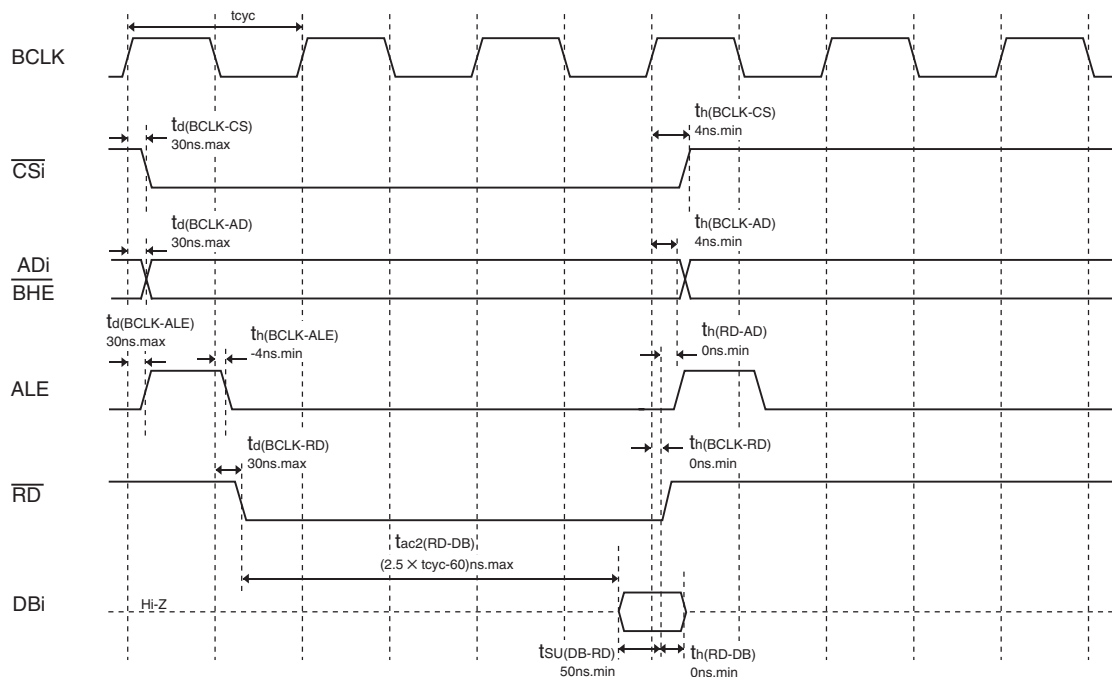
- VCC = 3.3 V
- Input timing voltage :  $V_{\text{IL}} = 0.6 \text{ V}$ ,  $V_{\text{IH}} = 2.7 \text{ V}$
- Output timing voltage :  $V_{\text{OL}} = 1.65 \text{ V}$ ,  $V_{\text{OH}} = 1.65 \text{ V}$

Figure 22.27 Timing Diagram (4)

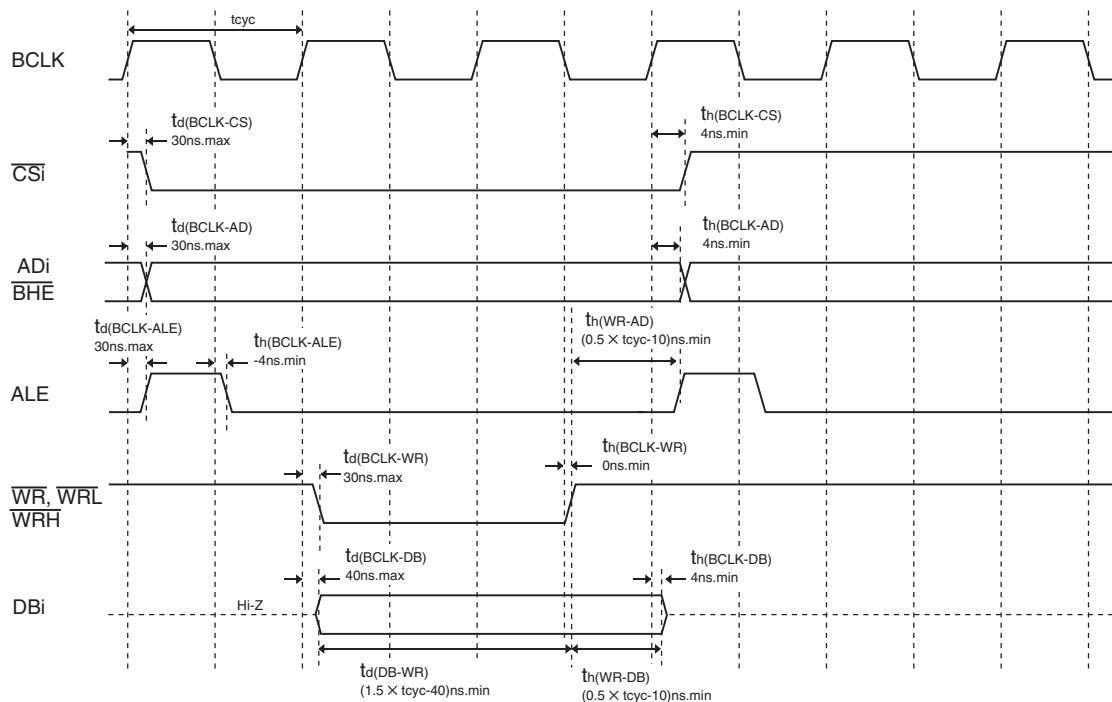
## Memory Expansion Mode and Microprocessor Mode (For 2-wait setting and external area access)

VCC = 3.3V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

Measuring conditions :

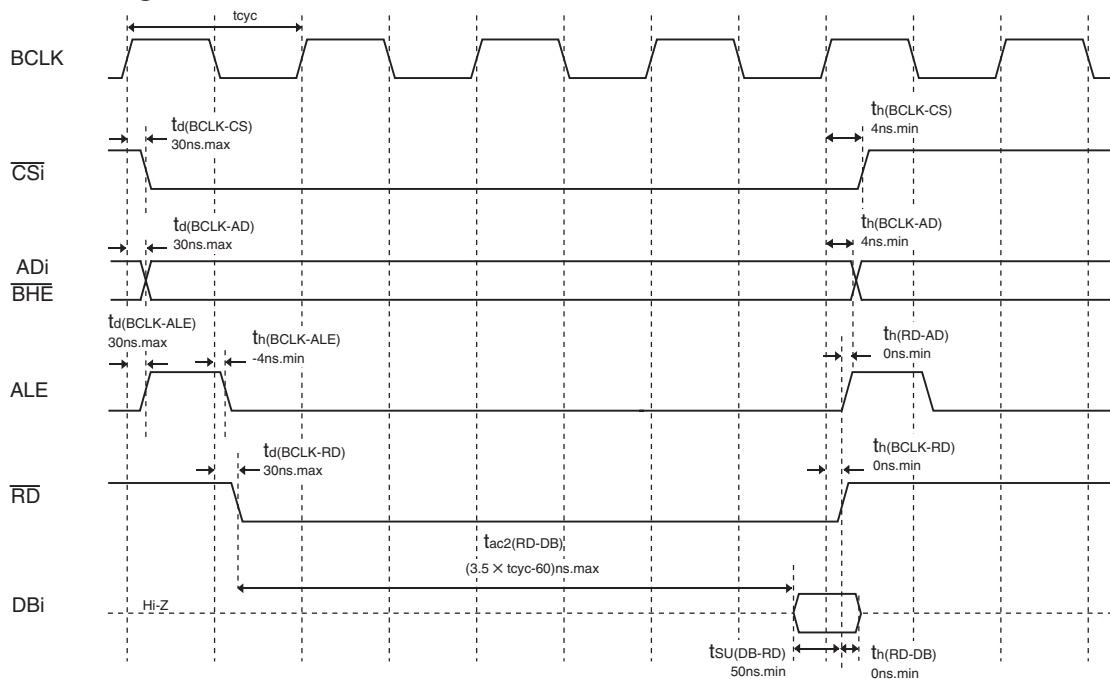
- VCC = 3.3 V
- Input timing voltage :  $V_{IL} = 0.6$  V,  $V_{IH} = 2.7$  V
- Output timing voltage :  $V_{OL} = 1.65$  V,  $V_{OH} = 1.65$  V

Figure 22.28 Timing Diagram (5)

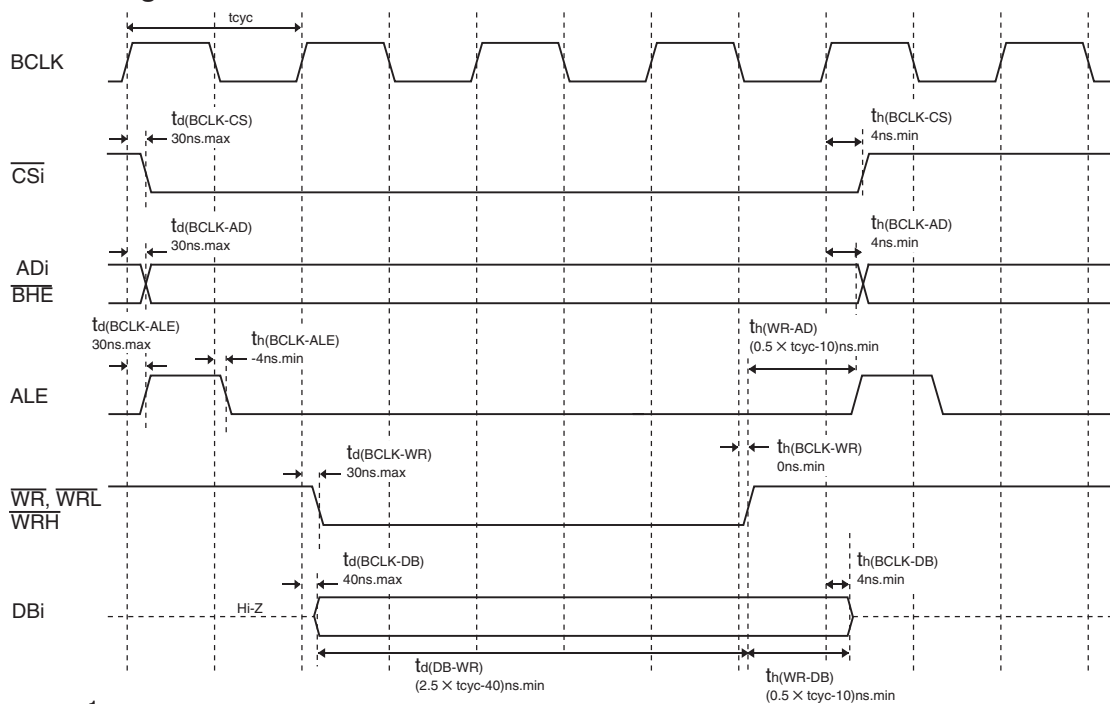
## Memory Expansion Mode and Microprocessor Mode (For 3-wait setting and external area access)

VCC = 3.3V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

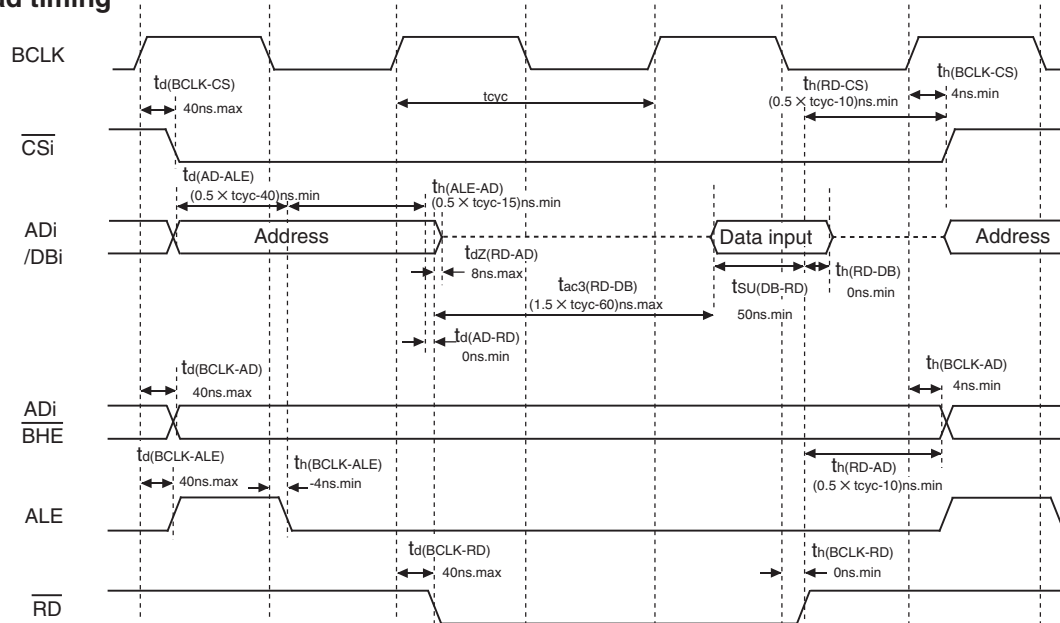
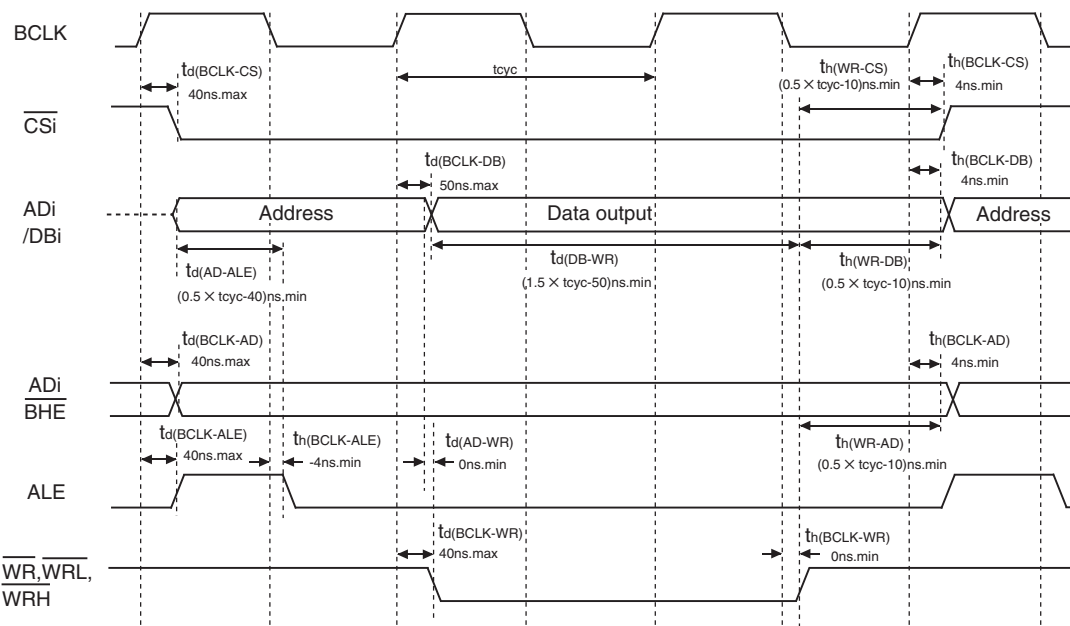
Measuring conditions :

- VCC = 3.3 V
- Input timing voltage :  $V_{IL} = 0.6$  V,  $V_{IH} = 2.7$  V
- Output timing voltage :  $V_{OL} = 1.65$  V,  $V_{OH} = 1.65$  V

Figure 22.29 Timing Diagram (6)

**Memory Expansion Mode and Microprocessor Mode****VCC = 3.3V**

(For 2-wait setting, external area access and multiplexed bus selection)

**Read timing****Write timing**

$$t_{cy} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions :

- VCC = 3.3 V
- Input timing voltage :  $V_{IL} = 0.6 \text{ V}$ ,  $V_{IH} = 2.7 \text{ V}$
- Output timing voltage :  $V_{OL} = 1.65 \text{ V}$ ,  $V_{OH} = 1.65 \text{ V}$

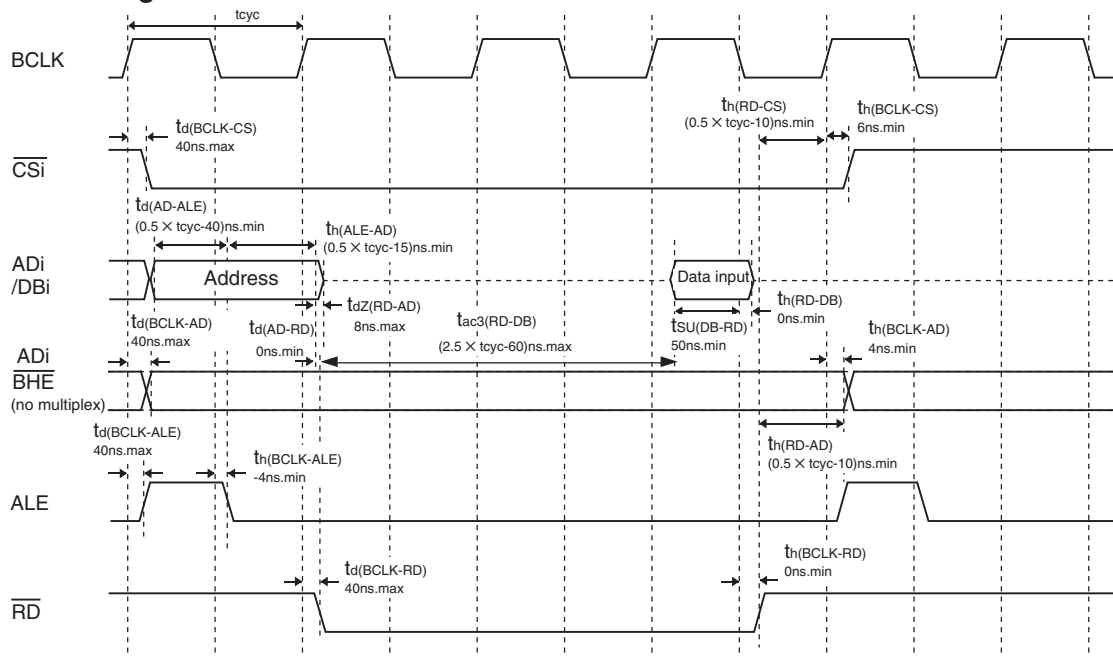
**Figure 22.30 Timing Diagram (7)**

## Memory Expansion Mode and Microprocessor Mode

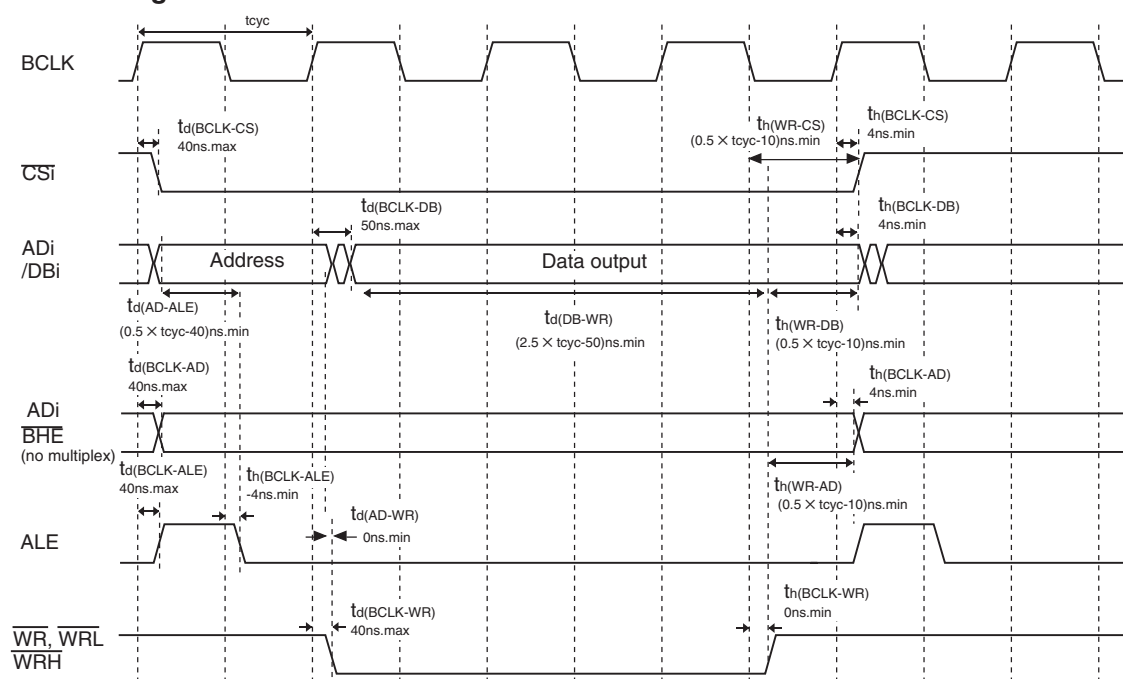
(For 3-wait setting, external area access and multiplexed bus selection)

VCC = 3.3V

### Read timing



### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

Measuring conditions :

- VCC = 3.3 V
- Input timing voltage :  $V_{IL} = 0.6$  V,  $V_{IH} = 2.7$  V
- Output timing voltage :  $V_{OL} = 1.65$  V,  $V_{OH} = 1.65$  V

Figure 22.31 Timing Diagram (8)

## 23. Usage Precaution

### 23.1 External Bus

When resetting CNVSS pin with “H” input, contents of internal ROM cannot be read out.



## 23.2 PLL Frequency Synthesizer

Stabilize supply voltage so that the standard of the power supply ripple is met. (Refer to **22. Electrical characteristics.**)

## 23.3 Power Control

- When exiting stop mode by hardware reset, set RESET pin to “L” until a main clock oscillation is stabilized.
- Set the MR0 bit in the TAI<sub>MR</sub> register (i = 0 to 4) to “0” (pulse is not output) to use the timer A to exit stop mode.
- When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not execute any instructions which can generate a write to RAM between the JMP.B and WAIT instructions. Disable the DMA transfers, if a DMA transfer may occur between the JMP.B and WAIT instructions. After the WAIT instruction, insert at least 4 NOP instructions. When entering wait mode, the instruction queue roadstead the instructions following WAIT, and depending on timing, some of these may execute before the 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

```

- 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      L2    ; Insert JMP.B instruction
                      L2:
                      NOP        ; More than 4 NOP instructions
                      NOP
                      NOP
                      NOP

```

- Wait for main clock oscillation stabilization time, before switching the clock source for CPU clock to the main clock.  
Similarly, wait until the sub clock oscillates stably before switching the clock source for CPU clock to the sub clock.

- Suggestions to reduce power consumption.

## Ports

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A pass current flows in input ports that high-impedance state. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

## A/D converter

When A/D conversion is not performed, set the VCUT bit in the ADCON1 register to “0” (VREF not connection). When A/D conversion is performed, start the A/D conversion at least 1  $\mu$ s or longer after setting the VCUT bit to “1” (VREF connection).

## D/A converter

When not performing D/A conversion, set the DAiE bit ( $i = 0, 1$ ) in the DACON register to “0” (input disabled) and DAi register to “00h”.

## Stopping peripheral functions

Use the CM02 bit in the CM0 register to stop the unnecessary peripheral functions during wait mode. However, because the peripheral function clock (fC32) generated from the sub-clock does not stop, this measure is not conducive to reducing the power consumption of the chip. If low speed mode or low power dissipation mode is to be changed to wait mode, set the CM02 bit to “0” (do not peripheral function clock stopped when in wait mode), before changing wait mode.

## Switching the oscillation-driving capacity

Set the driving capacity to “LOW” when oscillation is stable.

### 23.4 Protection

Set the PRC2 bit to “1” (write enabled) and then write to any address, and the PRC2 bit will be set to “0” (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to “1”. Make sure no interrupts or no DMA transfers will occur between the instruction in which the PRC2 bit is set to “1” and the next instruction.

## 23.5 Interrupt

### 23.5.1 Reading Address 00000h

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

If the address 00000h is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to "0". This causes a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

### 23.5.2 Setting SP

Set any value in the SP (USP, ISP) before accepting an interrupt. The SP (USP, ISP) is set to "0000h" after reset. Therefore, if an interrupt is accepted before setting any value in the SP (USP, ISP), the program may go out of control.

Especially when using  $\overline{\text{NMI}}$  interrupt, set a value in the ISP at the beginning of the program. For the first and only the first instruction after reset, all interrupts including NMI interrupt are disabled.

### 23.5.3 $\overline{\text{NMI}}$ Interrupt

- The  $\overline{\text{NMI}}$  interrupt cannot be disabled. If this interrupt is unused, connect the  $\overline{\text{NMI}}$  pin to VCC via a resistor (pull-up).
- The input level of the  $\overline{\text{NMI}}$  pin can be read by accessing the P8\_5 bit in the P8 register. Note that the P8\_5 bit can only be read when determining the pin level in  $\overline{\text{NMI}}$  interrupt routine.
- Stop mode cannot be entered into while input on the  $\overline{\text{NMI}}$  pin is low. This is because while input on the  $\overline{\text{NMI}}$  pin is low the CM10 bit in the CM1 register is fixed to "0".
- Do not go to wait mode while input on the  $\overline{\text{NMI}}$  pin is low. This is because when input on the  $\overline{\text{NMI}}$  pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
- The low and high level durations of the input signal to the  $\overline{\text{NMI}}$  pin must each be 2 CPU clock cycles + 300 ns or more.

### 23.5.4 Changing Interrupt Generate Factor

If the interrupt generate factor is changed, the IR bit of 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 set 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 set the IR bit for that interrupt to “0” (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.

Figure 23.1 shows the procedure for changing the interrupt generate factor.

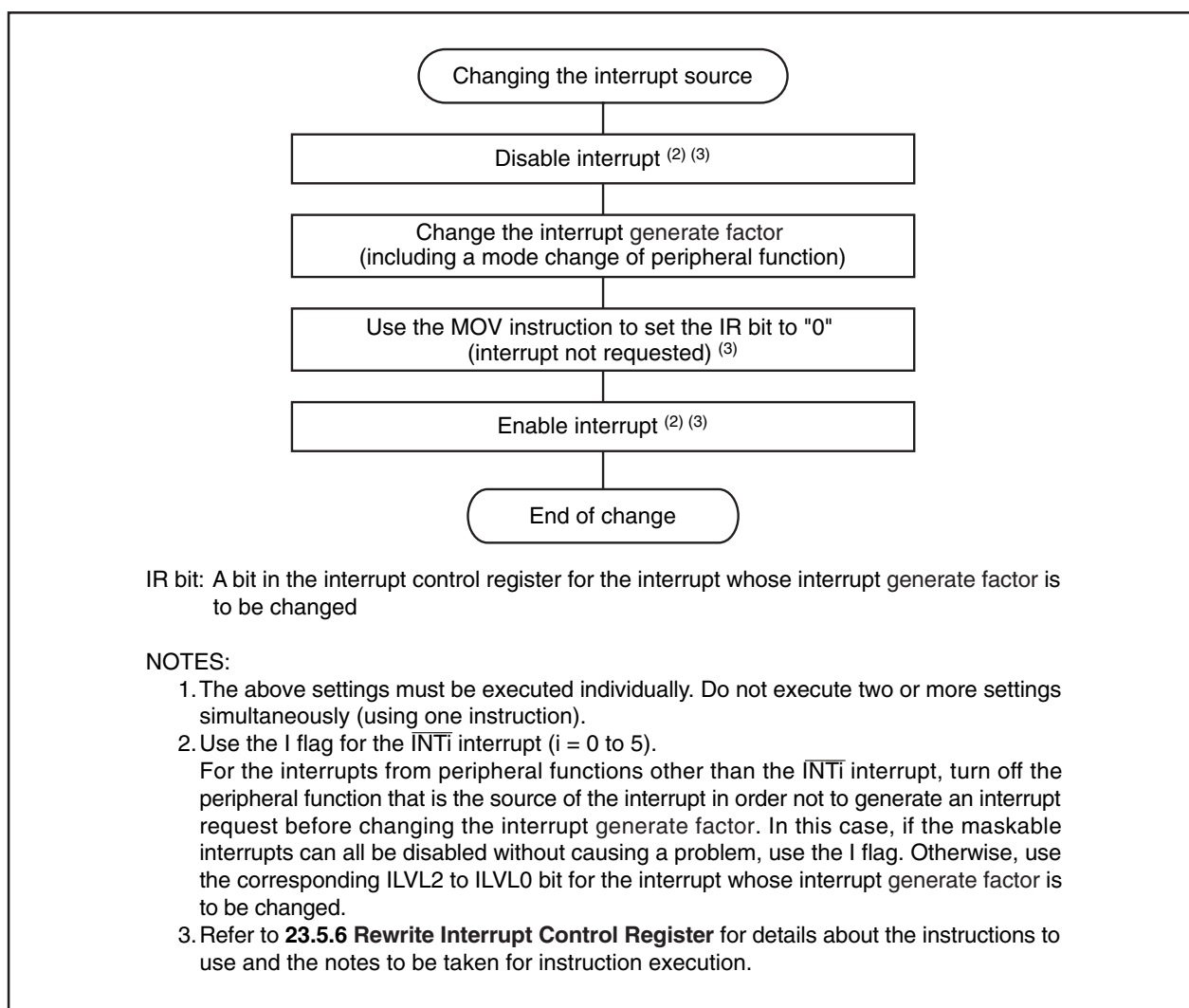


Figure 23.1 Procedure for Changing Interrupt Generate Factor

### 23.5.5 $\overline{\text{INT}}$ Interrupt

- Either an “L” level of at least  $tW(\text{INH})$  or an “H” level of at least  $tW(\text{INL})$  width is necessary for the signal input to pins  $\overline{\text{INT}}0$  to  $\overline{\text{INT}}5$  regardless of the CPU operation clock.
- If the POL bit in the INT0IC to INT5IC registers or the IFSR10 to IFSR17 bits in the IFSR1 register are changed, the IR bit may inadvertently set to “1” (interrupt requested). Be sure to set the IR bit to “0” (interrupt not requested) after changing any of those register bits.

### 23.5.6 Rewrite Interrupt Control Register

- (a) The interrupt control register for any interrupt should be modified in places where no interrupt requests may be generated. Otherwise, disable the interrupt before rewriting the interrupt control register.
- (b) To rewrite the interrupt control register for any interrupt after disabling that interrupt, be careful with the instruction to be used.

#### Changing any bit other than IR bit

If while executing an instruction, an interrupt request controlled by the register being modified is generated, the IR bit of the register may not be set to "1" (interrupt requested), with the result that the interrupt request is ignored. If such a situation presents a problem, use the instructions shown below to modify the register.

Usable instructions: AND, OR, BCLR, BSET

#### Changing IR bit

Depending on the instruction used, the IR bit may not always be set to "0" (interrupt not requested). Therefore, be sure to use the MOV instruction to set the IR bit to "0".

- (c) When using the I flag to disable an interrupt, refer to the sample program fragments shown below as you set the I flag. (Refer to (b) for details about rewrite the interrupt control registers in the sample program fragments.)

Examples 1 through 3 show how to prevent the I flag from being set to "1" (interrupt enabled) before the interrupt control register is rewritten, owing to the effects of the internal bus and the instruction queue buffer.

Example 1: Using the NOP instruction to keep the program waiting until the interrupt control register is modified

```
INT_SWITCH1:
    FCLR    I                ; Disable interrupt.
    AND.B   #00h, 0055h     ; Set the TA0IC register to "00h".
    NOP
    NOP
    FSET    I                ; Enable interrupt.
```

The number of the NOP instruction is as follows.

- The PM20 bit in the PM2 register = 1 (1 wait) : 2
- The PM20 bit = 0 (2 waits) : 3
- When using HOLD function : 4

Example 2: Using the dummy read to keep the FSET instruction waiting

```
INT_SWITCH2:
    FCLR    I                ; Disable interrupt.
    AND.B   #00h, 0055h     ; Set the TA0IC register to "00h".
    MOV.W   MEM, R0         ; Dummy read.
    FSET    I                ; Enable interrupt.
```

Example 3: Using the POPC instruction to changing the I flag

```
INT_SWITCH3:
    PUSHC   FLG
    FCLR    I                ; Disable interrupt.
    AND.B   #00h, 0055h     ; Set the TA0IC register to "00h".
    POPC    FLG             ; Enable interrupt.
```

### 23.5.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt request is generated.

## 23.6 DMAC

### 23.6.1 Write to DMAE Bit in DMiCON Register (i = 0, 1)

When both of the conditions below are met, follow the steps below.

#### Conditions

- The DMAE bit is set to “1” again while it remains set (DMAi is in an active state).
- A DMA request may occur simultaneously when the DMAE bit is being written.

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



## 23.7 Timers

### 23.7.1 Timer A

#### 23.7.1.1 Timer A (Timer Mode)

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.

While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, if the counter is read at the same time it is reloaded, the value “FFFFh” is read. Also, if the counter is read before it starts counting after a value is set in the TAI register while not counting, the set value is read.

If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

#### 23.7.1.2 Timer A (Event Counter Mode)

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 in the ONSF register and the TRGSR register are modified while the TAI<sub>S</sub> bit remains “0” (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, “FFFFh” can be read in underflow, while reloading, and “0000h” in overflow. When setting the TAI register to a value during a counter stop, the setting value can be read before a counter starts counting. Also, if the counter is read before it starts counting after a value is set in the TAI register while not counting, the set value is read.

If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

### 23.7.1.3 Timer A (One-shot Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI<sub>MR</sub> (i = 0 to 4) register, the TAI register, the 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 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.

When setting the TAI<sub>S</sub> bit to “0” (count stop), the followings occur:

- A counter stops counting and a content of reload register is reloaded.
- TAI<sub>OUT</sub> pin outputs “L”.
- After one cycle of the CPU clock, the IR bit in the TAI<sub>IC</sub> register is set to “1” (interrupt request).

Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger has been selected, one-cycle delay of a count source as maximum occurs between a trigger input to TAI<sub>IN</sub> pin and output in one-shot timer mode.

The IR bit is set to “1” when timer operation mode is set with any of the following procedures:

- Select one-shot timer mode after reset.
- Change an operation mode from timer mode to one-shot timer mode.
- Change an operation mode from event counter mode to one-shot timer mode.

To use the Timer Ai interrupt (the IR bit), set the IR bit to “0” after the changes listed above have been made.

When a trigger occurs, while counting, a counter reloads the reload register to continue counting after generating a re-trigger and counting down once. To generate a trigger while counting, generate a second trigger between occurring the previous trigger and operating longer than one cycle of a timer count source.

If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the TA1<sub>OUT</sub>, TA2<sub>OUT</sub> and TA4<sub>OUT</sub> pins go to a high-impedance state.

#### 23.7.1.4 Timer A (Pulse Width Modulation Mode)

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 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 TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register are modified while the TAI<sub>S</sub> bit remains “0” (count stops) regardless whether after reset or not.

The IR bit is set to “1” when setting a timer operation mode with any of the following procedures:

- Select the pulse width modulation mode after reset.
- Change an operation mode from timer mode to pulse width modulation mode.
- Change an operation mode from event counter mode to pulse width modulation mode.

To use the Timer Ai interrupt (the IR bit), set the IR bit to “0” by program after the above listed changes have been made.

When setting TAI<sub>S</sub> bit to “0” (count stop) during PWM pulse output, the following action occurs:

- Stop counting.
- When TAI<sub>OUT</sub> pin is output “H”, output level is set to “L” and the IR bit is set to “1”.
- When TAI<sub>OUT</sub> pin is output “L”, both output level and the IR bit remain unchanged.

If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on NMI pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

## 23.7.2 Timer B

### 23.7.2.1 Timer B (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 5) register and TBi register before setting the TBiS bit <sup>(1)</sup> in the TABSR or the TBSR register to “1” (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not.

NOTE:

1. The TB0S to TB2S bits are the bits 5 to 7 in the TABSR register, the TB3S to TB5S bits are the bits 5 to 7 in the TBSR register.

A value of a counter, while counting, can be read in the TBi register at any time. “FFFFh” is read while reloading. Setting value is read between setting values in the TBi register at count stop and starting a counter.

### 23.7.2.2 Timer B (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 5) register and TBi register before setting the TBiS bit in the TABSR or the TBSR register to “1” (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not.

The counter value can be read out on-the-fly at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always “FFFFh.” If the TBi register is read after setting a value in it while not counting but before the counter starts counting, the read value is the one that has been set in the register.

### 23.7.2.3 Timer B (Pulse Period/pulse Width Measurement Mode)

The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR (i = 0 to 5) register before setting the TBiS bit in the TABSR or TBSR register to "1" (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains "0" (count stops) regardless whether after reset or not. To set the MR3 bit to "0" by writing to the TBiMR register while the TBiS bit = 1 (count starts), be sure to write the same value as previously written to the TM0D0, TM0D1, MR0, MR1, TCK0 and TCK1 bits and a 0 to the MR2 bit.

The IR bit in the TBiIC register goes to "1" (interrupt request), when an effective edge of a measurement pulse is input or timer Bi is overflowed. The factor of interrupt request can be determined by use of the MR3 bit in the TBiMR register within the interrupt routine.

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.

To set the MR3 bit to "0" (no overflow), set the TBiMR register with setting the TBiS bit to "1" and counting the next count source after setting the MR3 bit to "1" (overflow).

Use the IR bit in the TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt factor.

When a 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.

A value of the counter is indeterminate at the beginning of a count. The MR3 bit may be set to "1" and Timer Bi interrupt request may be generated between a count start and an effective edge input.

For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an "H" level width or an "L" level width.

## 23.8 Serial Interface

### 23.8.1 Clock Synchronous Serial I/O Mode

#### 23.8.1.1 Transmission/reception

With an external clock selected, and choosing the  $\overline{\text{RTS}}$  function, the output level of the  $\overline{\text{RTSi}}$  pin goes to “L” when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the  $\overline{\text{RTSi}}$  pin goes to “H” when reception starts. So if the  $\overline{\text{RTSi}}$  pin is connected to the  $\overline{\text{CTSi}}$  pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the  $\overline{\text{RTS}}$  function has no effect.

If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the  $\overline{\text{RTS2}}$  and CLK2 pins go to a high-impedance state.

#### 23.8.1.2 Transmission

When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit in the UiC1 register = 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in UiTB register)
- If  $\overline{\text{CTS}}$  function is selected, input on the  $\overline{\text{CTSi}}$  pin = L

#### 23.8.1.3 Reception

In operating the clock synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TXDi (i = 0 to 2) pin when receiving data.

When an internal clock is selected, set the TE bit in the UiC1 register to “1” (transmission enabled) and write dummy data to the UiTB register, and the shift clock will thereby be generated. When an external clock is selected, set the TE bit to “1” and write dummy data to the UiTB register, and the shift clock will be generated when the external clock is fed to the CLKi input pin.

When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the RI bit in the UiC1 register = 1 (data present in the UiRB register), an overrun error occurs and the OER bit in the UiRB register is set to “1” (overrun error occurred). In this case, because the content of the UiRB register is 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.

To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.

When an external clock is selected, the conditions must be met while if the CKPOL bit = 0, the external clock is in the high state; if the CKPOL bit = 1, the external clock is in the low state.

- The RE bit in the UiC1 register = 1 (reception enabled)
- The TE bit in the UiC1 register = 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in the UiTB register)

## 23.8.2 Special Modes

### 23.8.2.1 Special Mode 1 (I<sup>2</sup>C Mode)

When generating start, stop and restart conditions, set the STSPSEL bit in the UiSMR4 register to “0” (start and stop conditions not output) and wait for more than half cycle of the transfer clock before setting each condition generate bit (STAREQ, RSTAREQ and STPREQ bits) from “0” (clear) to “1” (start).

### 23.8.2.2 Special Mode 2

If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the  $\overline{\text{IVPCR1}}$  bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the  $\overline{\text{RTS2}}$  and  $\overline{\text{CLK2}}$  pins go to a high-impedance state.

### 23.8.2.3 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to “1” (transmission complete) and U2ERE bit in the U2C1 register to “1” (error signal output) after reset. Therefore, when using SIM mode, be sure to set the IR bit to “0” (no interrupt request) after setting these bits.

### 23.8.3 SI/O3

The SOUT3 default value which is set to the SOUT3 pin by the SM37 in the S3C register bit approximately 10ns may be output when changing the SM33 bit in the S3C register from “0” (I/O port) to “1” (SOUT3 output and CLK3 function) while the SM32 bit in the S3C register to “0” (SOUT3 output) and the SM36 bit is set to “1” (internal clock). And then the SOUT3 pin is held high-impedance.

If the level which is output from the SOUT3 pin is a problem when changing the SM33 bit from “0” to “1”, set the default value of the SOUT3 pin by the SM37 bit.



## 23.9 A/D Converter

Set the ADCON0 (except bit 6), ADCON1 and ADCON2 registers when A/D conversion is stopped (before a trigger occurs).

When the VCUT bit in the ADCON1 register is changed from “0” (VREF not connected) to “1” (VREF connected), start A/D conversion after passing 1  $\mu$ s or longer.

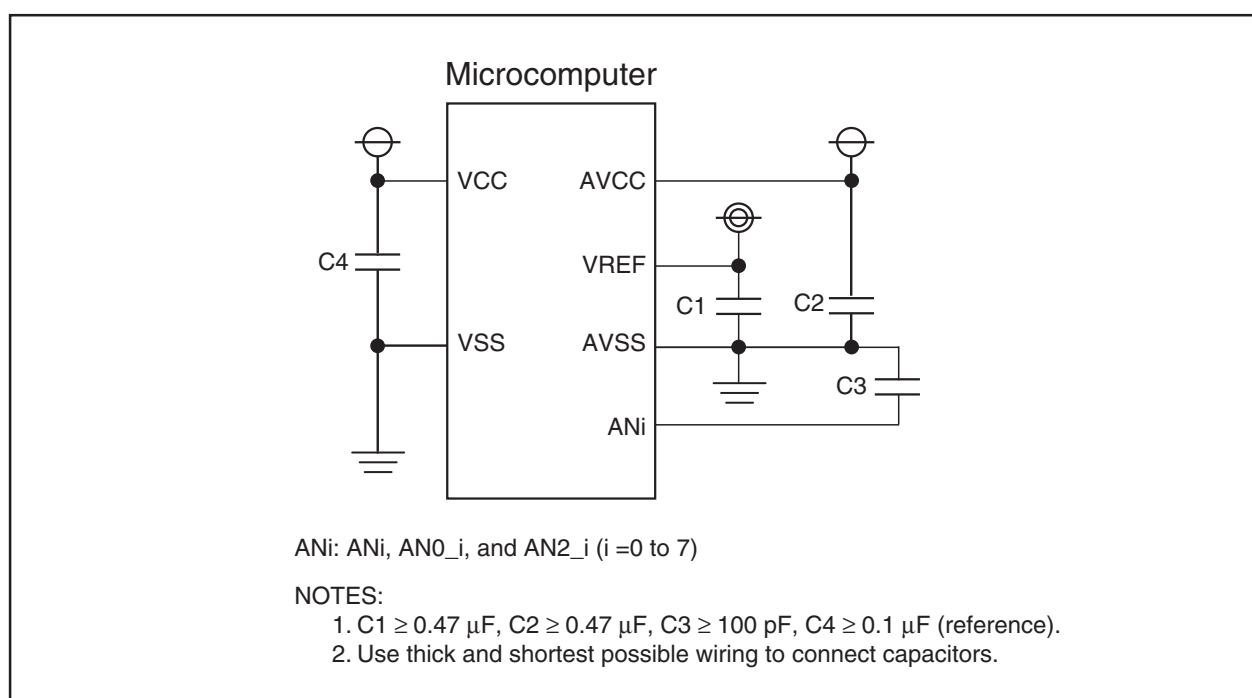
To prevent noise-induced device malfunction or latch-up, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins (ANi (i = 0 to 7), AN0\_i, and AN2\_i) each and the AVSS pin. Similarly, insert a capacitor between the VCC pin and the VSS pin. Figure 23.2 shows an example connection of each pin.

Make sure the port direction bits for those pins that are used as analog inputs are set to “0” (input mode). Also, if the TGR bit in the ADCON0 register = 1 (external trigger), make sure the port direction bit for the ADTRG pin is set to “0” (input mode).

When using key input interrupt, do not use any of the four AN4 to AN7 pins as analog inputs. (A key input interrupt request is generated when the A/D input voltage goes low.)

The  $\phi$ AD frequency must be 10 MHz or less. Without sample and hold, limit the  $\phi$ AD frequency to 250 kHz or more. With the sample and hold, limit the  $\phi$ AD frequency to 1 MHz or more.

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 23.2 Use of Capacitors to Reduce Noise**

If the CPU reads the ADi register at the same time the conversion result is stored in the ADi register after completion of A/D conversion, an incorrect value may be stored in the ADi register. This problem occurs when a divide-by-n clock derived from the main clock or a sub clock is selected for CPU clock.

- When operating in one-shot or single-sweep mode  
Check to see that A/D conversion is completed before reading the target ADi register. (Check the IR bit in the ADIC register to see if A/D conversion is completed.)
- When operating in repeat mode or repeat sweep mode 0 or 1  
Use the main clock for CPU clock directly without dividing it.

If A/D conversion is forcibly terminated while in progress by setting the ADST bit in the ADCON0 register to “0” (A/D conversion halted), the conversion result of the A/D converter is indeterminate. The contents of ADi registers irrelevant to A/D conversion may also become indeterminate. If while A/D conversion is underway the ADST bit is set to “0” in a program, ignore the values of all ADi registers.

When setting the ADST bit to “0” in single sweep mode during A/D conversion and A/D conversion is aborted, disable the interrupt before setting the ADST bit to “0”.

The applied intermediate potential may cause more increase in power consumption than other analog input pins (AN0 to AN3, AN0\_0 to AN0\_7 and AN2\_0 to AN2\_7), since the AN4 to AN7 are used with the KI0 to KI3.

## 23.10 CAN Module

### 23.10.1 Reading CiSTR Register (i = 0, 1)

The CAN module on the M16C/6N Group (M16C/6N4) updates the status of the CiSTR register in a certain period. When the CPU and the CAN module access to the CiSTR register at the same time, the CPU has the access priority; the access from the CAN module is disabled. Consequently, when the updating period of the CAN module matches the access period from the CPU, the status of the CAN module cannot be updated. (See **Figure 23.3 When Updating Period of CAN Module Matches Access Period from CPU.**)

Accordingly, be careful about the following points so that the access period from the CPU should not match the updating period of the CAN module:

- (a) There should be a wait time of 3fCAN or longer (see **Table 23.1 CAN Module Status Updating Period**) before the CPU reads the CiSTR register. (See **Figure 23.4 With a Wait Time of 3fCAN Before CPU Read.**)
- (b) When the CPU polls the CiSTR register, the polling period must be 3fCAN or longer. (See **Figure 23.5 When Polling Period of CPU is 3fCAN or Longer.**)

**Table 23.1 CAN Module Status Updating Period**

3fCAN Period = 3 × XIN (Original Oscillation Period) × Division Value of CAN Clock (CCLK)	
(Example 1) Condition XIN 16 MHz CCLK: Divided by 1	3fCAN period = 3 × 62.5 ns × 1 = 187.5 ns
(Example 2) Condition XIN 16 MHz CCLK: Divided by 2	3fCAN period = 3 × 62.5 ns × 2 = 375 ns
(Example 3) Condition XIN 16 MHz CCLK: Divided by 4	3fCAN period = 3 × 62.5 ns × 4 = 750 ns
(Example 4) Condition XIN 16 MHz CCLK: Divided by 8	3fCAN period = 3 × 62.5 ns × 8 = 1.5 μs
(Example 5) Condition XIN 16 MHz CCLK: Divided by 16	3fCAN period = 3 × 62.5 ns × 16 = 3 μs

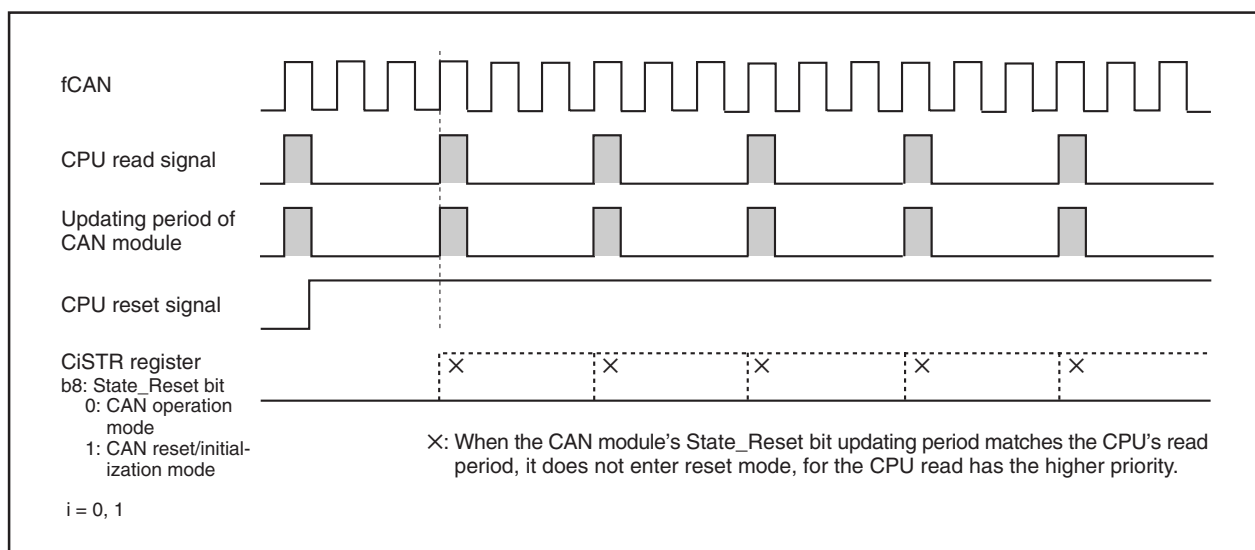
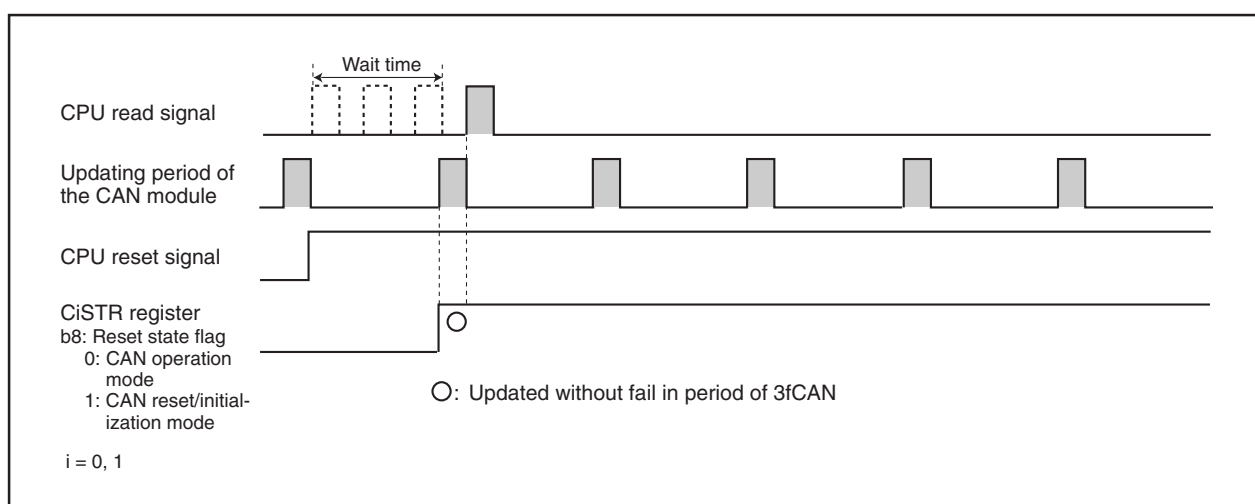
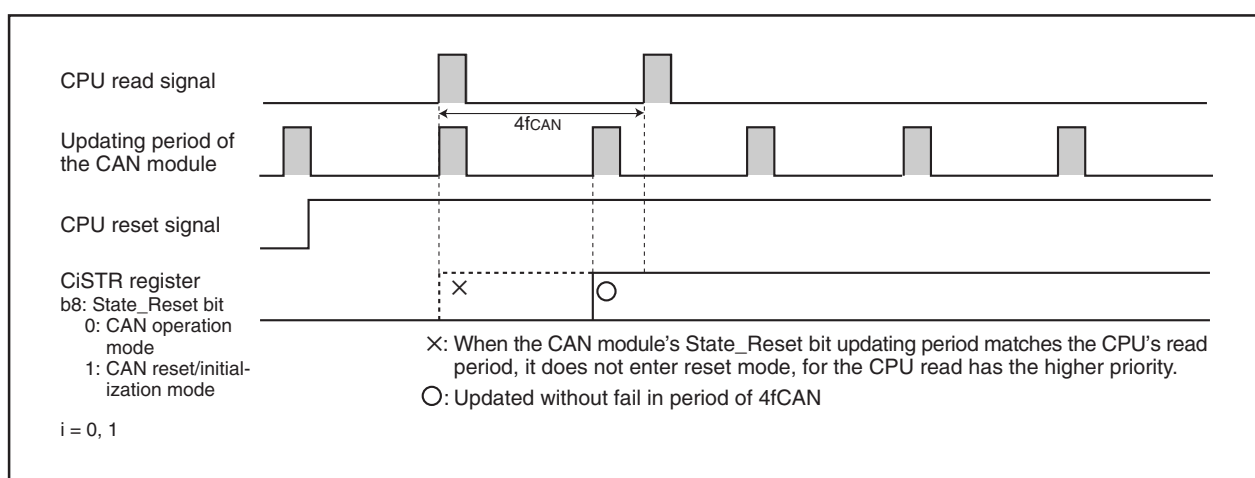


Figure 23.3 When Updating Period of CAN Module Matches Access Period from CPU

Figure 23.4 With a Wait Time of  $3f_{CAN}$  Before CPU ReadFigure 23.5 When Polling Period of CPU is  $3f_{CAN}$  or Longer

### 23.10.2 Performing CAN Configuration

If the Reset bit in the CiCTLR register ( $i = 0, 1$ ) is changed from “0” (operation mode) to “1” (reset/initialization mode) in order to place the CAN module from CAN operation mode into CAN reset/initialization mode, always be sure to check that the State\_Reset bit in the CiSTR register is set to “1” (reset mode). Similarly, if the Reset bit is changed from “1” to “0” in order to place the CAN module from CAN reset/initialization mode into CAN operation mode, always be sure to check that the State\_Reset bit is set to “0” (operation mode).

The procedure is described below.

#### **To place CAN Module from CAN Operation Mode into CAN Reset/Initialization Mode**

- Change the Reset bit from “0” to “1”.
- Check that the State\_Reset bit is set to “1”.

#### **To place CAN Module from CAN Reset/Initialization Mode into CAN Operation Mode**

- Change the Reset bit from “1” to “0”.
- Check that the State\_Reset bit is set to “0”.

### 23.10.3 Suggestions to Reduce Power Consumption

When not performing CAN communication, the operation mode of CAN transceiver should be set to “standby mode” or “sleep mode”.

When performing CAN communication, the power consumption in CAN transceiver in not performing CAN communication can be substantially reduced by controlling the operation mode pins of CAN transceiver.

Tables 23.2 and 23.3 show recommended pin connections.

**Table 23.2 Recommended Pin Connections (In case of PCA82C250: Philips product)**

	Standby Mode	High-speed Mode
Rs Pin <sup>(1)</sup>	“H”	“L”
Power Consumption in CAN Transceiver <sup>(2)</sup>	less than 170 $\mu$ A	less than 70 mA
CAN Communication	impossible	possible
Connection		

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. In case of Ta = 25 °C
3. Connect to enabled port to control CAN transceiver.

**Table 23.3 Recommended Pin Connections (In case of PCA82C252: Philips product)**

	Sleep Mode	Normal Operation Mode
STB Pin <sup>(1)</sup>	“L”	“H”
EN Pin <sup>(1)</sup>	“L”	“H”
Power Consumption in CAN Transceiver <sup>(2)</sup>	less than 50 $\mu$ A	less than 35 mA
CAN Communication	impossible	possible
Connection		

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. Ta = 25 °C
3. Connect to enabled port to control CAN transceiver.

### 23.10.4 CAN Transceiver in Boot Mode

When programming the flash memory in boot mode via CAN bus, the operation mode of CAN transceiver should be set to “high-speed mode” or “normal operation mode”. If the operation mode is controlled by the microcomputer, CAN transceiver must be set the operation mode to “high-speed mode” or “normal operation mode” before programming the flash memory by changing the switch etc. Tables 23.4 and 23.5 show pin connections of CAN transceiver.

**Table 23.4 Pin Connections of CAN Transceiver (In case of PCA82C250: Philips product)**

	Standby Mode	High-speed Mode
Rs Pin <sup>(1)</sup>	“H”	“L”
CAN Communication	impossible	possible
Connection		

i = 0, 1

**NOTES:**

1. The pin which controls the operation mode of CAN transceiver.
2. Connect to enabled port to control CAN transceiver.

**Table 23.5 Pin Connections of CAN Transceiver (In case of PCA82C252: Philips product)**

	Sleep Mode	Normal Operation Mode
STB Pin <sup>(1)</sup>	“L”	“H”
EN Pin <sup>(1)</sup>	“L”	“H”
CAN Communication	impossible	possible
Connection		

i = 0, 1

**NOTES:**

1. The pin which controls the operation mode of CAN transceiver.
2. Connect to enabled port to control CAN transceiver.

### 23.11 Programmable I/O Ports

If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the P7\_2 to P7\_5, P8\_0 and P8\_1 pins go to a high-impedance state.

Setting the SM32 bit in the S3C register to “1” causes the P9\_2 pin to go to a high-impedance state.

The input threshold voltage of pins differs between programmable I/O ports and peripheral functions. Therefore, if any pin is shared by a programmable I/O port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions VIH and VIL (neither “high” nor “low”), the input level may be determined differently depending on which side—the programmable I/O port or the peripheral function—is currently selected.

Indeterminate values are read from the P3\_7 to P3\_4, PD3\_7 to PD3\_4 bits by reading the P3 and PD3 registers when the PM01 to PM00 bits in the PM0 register are set to “01b” (memory expansion mode) or “11b” (microprocessor mode) and setting the PM11 bit to “1”.

Use the MOV instruction when rewriting the P3 and PD3 registers (including the case that the size specifier is “.W” and the P2 and PD2 registers are rewritten).

When the PM01 to PM00 bits are rewritten, “L” is output from the P3\_7 to P3\_4 pins during 0.5 cycles of the BCLK by setting the PM01 to PM00 bits in the PM0 register to “01b” (memory expansion mode) or “11b” (microprocessor mode) from “00b” (single-chip mode) after setting the PM11 bit to “1”.



## **23.12 Electrical 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.

### **23.13 Mask ROM Version**

When using the masked ROM version, write nothing to internal ROM area.

## 23.14 Flash Memory Version

### 23.14.1 Functions to Prevent Flash Memory from Rewriting

ID codes are stored in addresses 0FFFDf, 0FFFE3h, 0FFFEb, 0FFFEf, 0FFFF3h, 0FFFF7h, and 0FFFFb. If wrong data are written to these addresses, the flash memory cannot be read or written in standard serial I/O mode and CAN I/O mode.

The ROMCP register is mapped in address 0FFFFf. 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.

### 23.14.2 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.

### 23.14.3 Wait Mode

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

### 23.14.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode

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

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

### 23.14.5 Writing Command and Data

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

### 23.14.6 Program Command

By writing “xx40h” in the first bus cycle and data to the write address in the second bus cycle, an auto program operation (data program and verify) will start. The address value specified in the first bus cycle must be the same even address as the write address specified in the second bus cycle.

### 23.14.7 Lock Bit Program Command

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

### 23.14.8 Operation Speed

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

### 23.14.9 Prohibited Instructions

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

### 23.14.10 Interrupt

#### EW0 Mode

To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.

- The  $\overline{\text{NMI}}$  and watchdog timer interrupts are available since the FMR0 and FMR1 registers are forcibly reset when either interrupt request is generated. Allocate the jump addresses for each interrupt service routines to the fixed vector table. Flash memory rewrite operation is aborted when the  $\overline{\text{NMI}}$  or watchdog timer interrupt request is generated. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

#### EW1 Mode

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

### 23.14.11 How to Access

To set the FMR01, FMR02 or FMR11 bit to “1”, write “1” after first setting the bit to “0”. Do not generate an interrupt or a DMA transfer between the instruction to set the bit to “0” and the instruction to set the bit to “1”. Set the bit while an “H” signal is applied to the  $\overline{\text{NMI}}$  pin.

### 23.14.12 Rewriting in User ROM Area

#### EW0 Mode

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

#### EW1 Mode

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

### 23.14.13 DMA Transfer

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

## 23.15 Flash Memory Programming Using Boot Program

When programming the internal flash memory using boot program, be careful about the pins state and connection as follows.

### 23.15.1 Programming Using Serial I/O Mode

CTX0 pin : This pin automatically outputs “H” level.

CRX0 pin : Connect to CAN transceiver or connect via resistor to VCC (pull-up)

Figure 23.6 shows a pin connection example for programming using serial I/O mode.

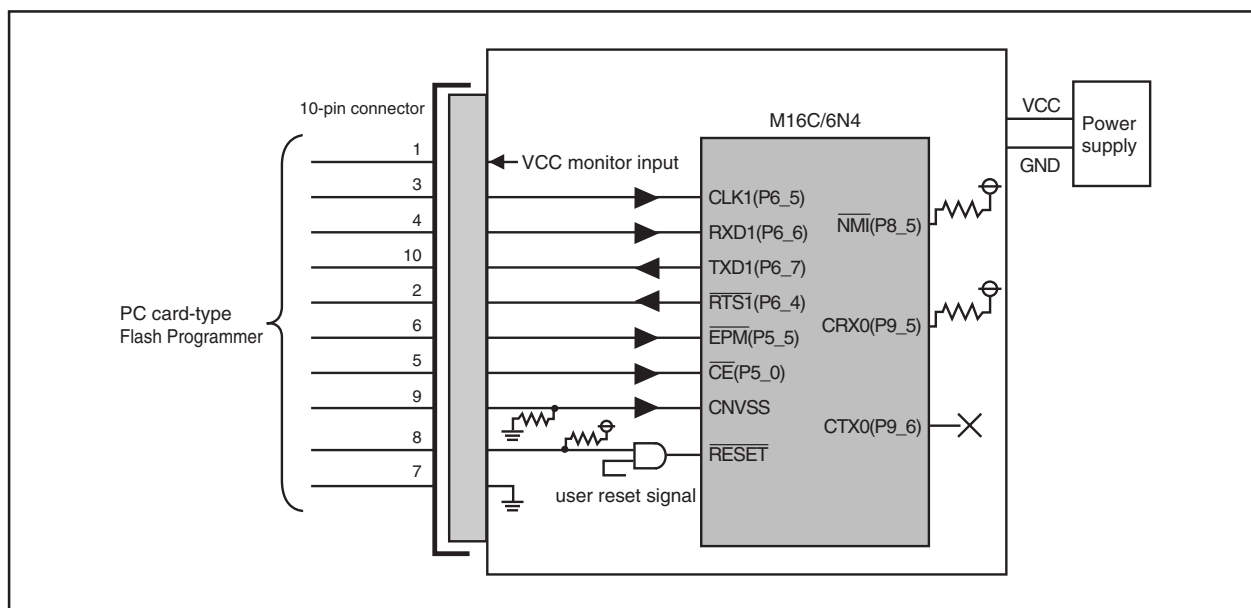


Figure 23.6 Pin Connection for Programming Using Serial I/O Mode

### 23.15.2 Programming Using CAN I/O Mode

RTS1 pin : This pin automatically outputs “H” and “L” level.

Figure 23.7 shows a pin connection example for programming using CAN I/O mode.

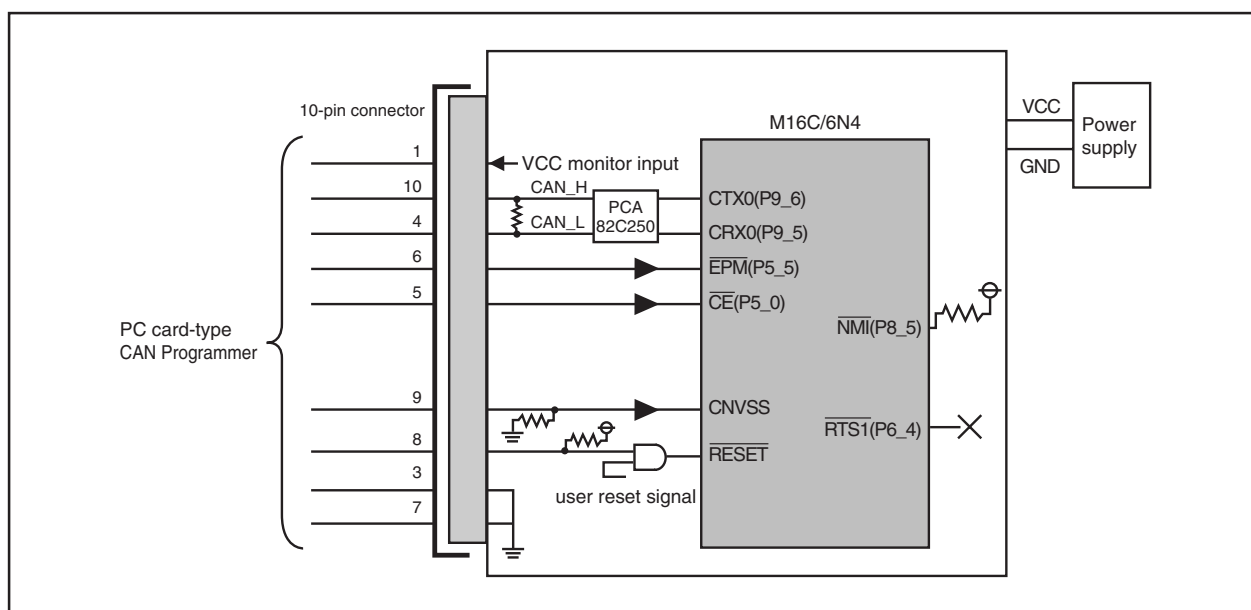
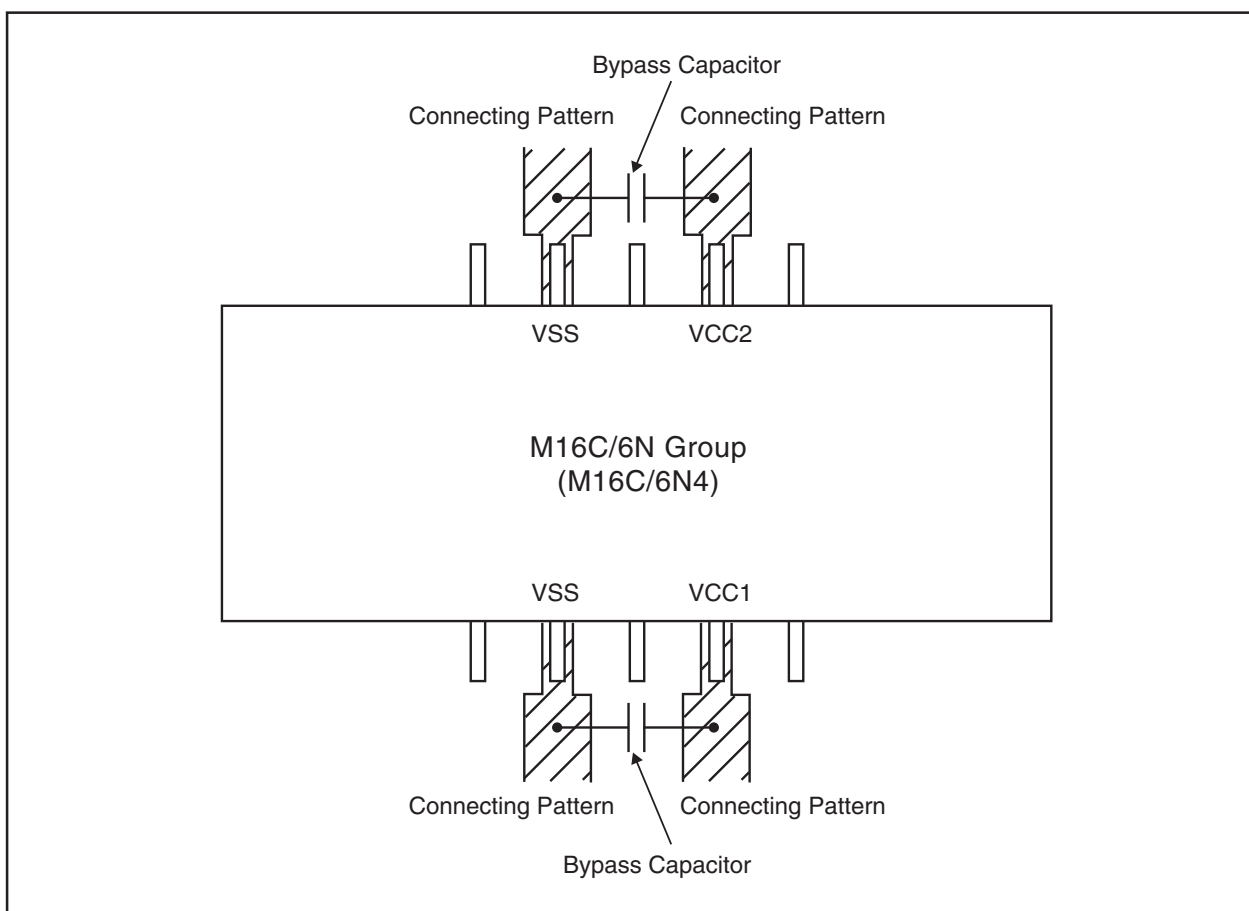


Figure 23.7 Pin Connection for Programming Using CAN I/O Mode

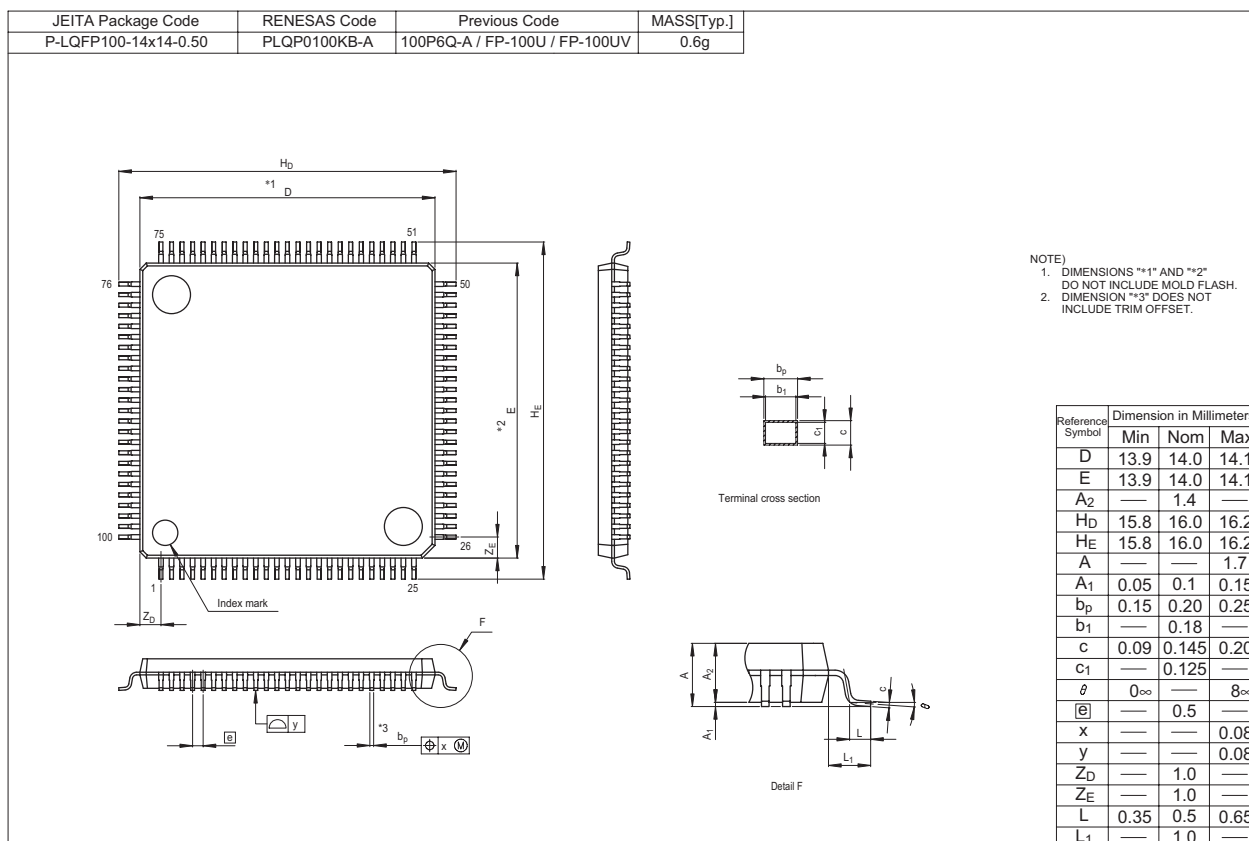
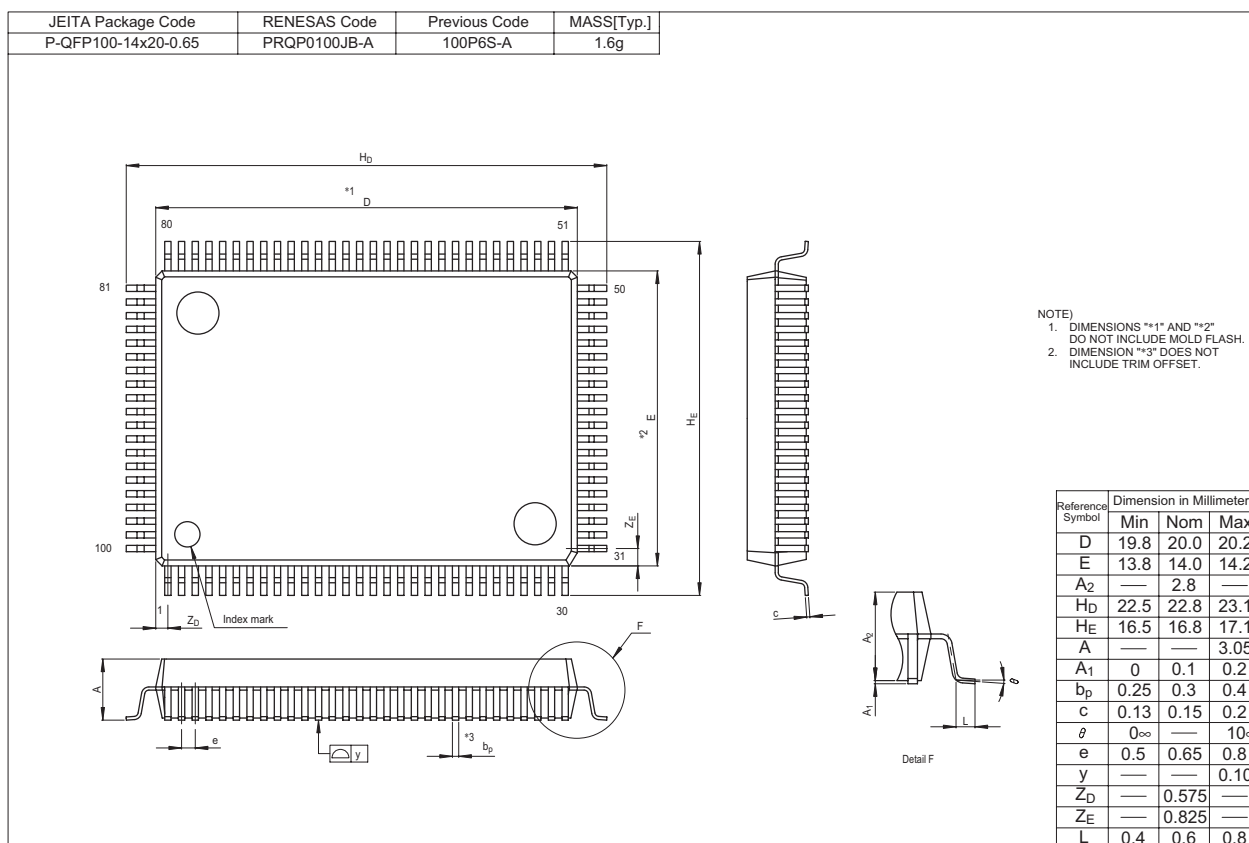
### 23.16 Noise

Connect a bypass capacitor (approximately 0.1  $\mu\text{F}$ ) across the VCC1 and VSS pins, and VCC2 and VSS pins using the shortest and thicker possible wiring. Figure 23.8 shows the bypass capacitor connection.



**Figure 23.8 Bypass Capacitor Connection**

## Appendix 1. Package Dimensions



## *Memo*



# Register Index

<b>A</b>		<b>T</b>	
AD0 to AD7 .....	196	TA0 .....	108
ADCON0 ....	195,198,200,202,204,206	TA0IC .....	80
ADCON1 ....	195,198,200,202,204,206	TA0MR .....	108,111,113,118,120
ADCON2 .....	196	TA1 .....	108,135
ADIC .....	80	TA11 .....	135
AIER .....	92	TA1IC .....	80
AIER2 .....	92	TA1MR .....	108,111,113,118,120,138
<b>C</b>		TA2 .....	108,135
C01ERRIC .....	80	TA21 .....	135
C01WKIC .....	80	TA2IC .....	80
C0AFS, C1AFS .....	224	TA2MR ....	108,111,113,115,118,120,138
C0CONR, C1CONR .....	223	TA3 .....	108
C0CTLR, C1CTLR .....	220	TA3IC .....	80
C0GMR, C1GMR .....	218	TA3MR .....	108,111,113,115,118,120
C0ICR, C1ICR .....	222	TA4 .....	108,135
C0IDR, C1IDR .....	222	TA41 .....	135
C0LMAR, C1LMAR .....	218	TA4IC .....	80
C0LMBR, C1LMBR .....	218	TA4MR ....	108,111,113,115,118,120,138
C0MCTL0 to C0MCTL15 .....	219	TABSR .....	109,124,137
C0RECIC .....	80	TB0 .....	123
C0RECR, C1RECR .....	224	TB0IC .....	80
C0SSTR, C1SSTR .....	225	TB0MR .....	123,125,126,128
C0STR, C1STR .....	221	TB1 .....	123
C0TECR, C1TECR .....	224	TB1IC .....	80
C0TRMIC .....	80	TB1MR .....	123,125,126,128
C0TSR, C1TSR .....	224	TB2 .....	123,135
C1MCTL0 to C1MCTL15 .....	219	TB2IC .....	80
C1RECIC .....	81	TB2MR .....	123,125,126,128,138
C1TRMIC .....	81	TB2SC .....	136
CAN0/1 Slot 0 to 15		TB3 .....	123
: Time Stamp .....	216,217	TB3IC .....	80
: Data Field .....	216,217	TB3MR .....	123,125,126,128
: Message Box .....	216,217	TB4 .....	123
CCLKR .....	56	TB4IC .....	80
CM0 .....	53	TB4MR .....	123,125,126,128
CM1 .....	54	TB5 .....	123
CM2 .....	55	TB5IC .....	80
CPSRF .....	110,124	TB5MR .....	123,125,126,128
CRCD .....	212	TBSR .....	124
CRCIN .....	212	TCR0, TCR1 .....	99
CSE .....	47	TRGSR .....	110,137
CSR .....	41	<b>U</b>	
<b>D</b>		U0BCNIC to U2BCNIC .....	80
DA0, DA1 .....	211	U0BRG to U2BRG .....	145
DACON .....	211	U0C0 to U2C0 .....	146
		U0C1 to U2C1 .....	147
DAR0, DAR1 .....	99		
DM0CON, DM1CON .....	98		
DM0IC, DM1IC .....	80		
DM0SL .....	97		
DM1SL .....	98		
DTT .....	134		
<b>F</b>			
FMR0 .....	257		
FMR1 .....	257		
<b>I</b>			
ICTB2 .....	136		
IDB0, IDB1 .....	134		
IFSR0 .....	89		
IFSR1 .....	89		
INT0IC to INT5IC .....	81		
INVC0 .....	132		
INVC1 .....	133		
<b>K</b>			
KUPIC .....	80		
<b>O</b>			
ONSF .....	110		
<b>P</b>			
P0 to P10 .....	246		
PCLKR .....	56		
PCR .....	248		
PD0 to PD10 .....	245		
PLC0 .....	57		
PM0 .....	35		
PM1 .....	36		
PM2 .....	57		
PRCR .....	74		
PUR0 to PUR2 .....	247		
<b>R</b>			
RMAD0 to RMAD3 .....	92		
ROMCP .....	254		
<b>S</b>			
S0RIC to S2RIC .....	80		
S0TIC to S2TIC .....	80		
S3BRG .....	189		
S3C .....	189		
S3IC .....	81		
S3TRR .....	189		
SAR0, SAR1 .....	99		

U0MR to U2MR .....	146
U0RB to U2RB .....	145
U0SMR to U2SMR .....	148
U0SMR2 to U2SMR2 .....	149
U0SMR3 to U2SMR3 .....	149
U0SMR4 to U2SMR4 .....	150
U0TB to U2TB .....	145
UCON .....	148
UDF .....	109

## **W**

WDC .....	94
WDTS .....	94

# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	May. 30, 2003	—	First edition issued
2.00	Nov. 10, 2004	—	<p>Revised edition issued</p> <ul style="list-style-type: none"> <li>* Words standardizes (on-chip oscillator)</li> <li>* 100P6Q-A (100-pin version) is added.</li> <li>* Usage Notes Reference Book is added to Chapter 23 Usage Precaution.</li> <li>* Revised parts and revised contents are as follows (except for change of chapter composition, change of a layout, and an expressional change).</li> </ul>
		1	1. Overview 3rd line: "and LQFP" is added.
		2	Table 1.1 Performance outline of M16C/6N Group (M16C/6N4) <ul style="list-style-type: none"> <li>• Operation Mode is added.</li> <li>• Address Space is added.</li> <li>• Power Consumption is revised.</li> <li>• "LQFP" is added to Package.</li> </ul>
		4	Table 1.2 Product List is revised. Figure 1.2 Type No., Memory Size, and Package: <ul style="list-style-type: none"> <li>• "GP: Package 100P6Q-A" is added to Package type.</li> </ul>
		5	Figure 1.3 Pin Configuration (Top View) (1): "ZP" is added.
		6	Figure 1.4 Pin Configuration (Top View) (2) is added. (100P6Q-A)
		8	Table 1.4 Pin Description (2): "ZP" is added to Timer A.
		12	<p>3. Memory</p> <ul style="list-style-type: none"> <li>• 5th to 6th lines: The description about the flash memory version (block A) is added.</li> </ul> <p>Figure 3.1 Memory Map:</p> <ul style="list-style-type: none"> <li>• Internal ROM (data area) is added.</li> <li>• NOTES 3, 4 are added and NOTE 5 is revised.</li> </ul>
		13	Table 4.1 SFR Information (1) <ul style="list-style-type: none"> <li>• The value of After Reset in PM1 register is revised.</li> <li>• The value of After Reset in CM2 register is revised.</li> </ul>
		19	Table 4.7 SFR Information (7) <ul style="list-style-type: none"> <li>• The value of After Reset in FMR0 register is revised.</li> </ul>
		27	Table 4.15 SFR Information (15) <ul style="list-style-type: none"> <li>• The value of After Reset in U0C1 register is revised.</li> <li>• The value of After Reset in U1C1 register is revised.</li> <li>• NOTE 1 is added.</li> </ul>
		28	Table 4.16 SFR Information (16) <ul style="list-style-type: none"> <li>• The value of After Reset in DA0, DA1 registers are revised.</li> </ul>
		30	Figure 5.1 Example Reset Circuit: NOTE 1 is added.
		34	Figure 6.2 PM1 Register <ul style="list-style-type: none"> <li>• The value of After Reset is revised.</li> <li>• NOTES 2, 6 are revised.</li> </ul>
		37	Figure 6.6 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (3) <ul style="list-style-type: none"> <li>• NOTE 2 is added.</li> </ul> <p>Figure 6.7 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (4)</p> <ul style="list-style-type: none"> <li>• NOTE 1 is added.</li> </ul>

# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.00	Nov. 10, 2004	38	Table 7.1 Difference between Separate Bus and Multiplexed Bus is added.
		39	Figure 7.1 CSR Register: NOTE 2 is revised.
		46	Table 7.8 Software Wait Related Bits and Bus Cycles <ul style="list-style-type: none"> <li>• Bus Cycle of SFR (PM20 = 0) is revised from "2 BCLK cycles" to "3 BCLK cycles".</li> <li>• Bus Cycle of SFR (PM20 = 1) is revised from "3 BCLK cycles" to "2 BCLK cycles".</li> <li>• From bottom to 5th item in CSR Register: The value is revised from "1" to "0".</li> <li>• NOTE 5 is added.</li> </ul>
		49	Table 8.1 Clock Generating Circuit Specifications <ul style="list-style-type: none"> <li>• Clock Frequency in PLL Frequency Synthesizer: 16 MHz is added.</li> </ul>
		50	Figure 8.1 Clock Generating Circuit: Block diagram (upper) is revised.
		51	Figure 8.2 CM0 Register <ul style="list-style-type: none"> <li>• Bit name of CM02 is revised.</li> <li>• NOTE 6 (2) and NOTE 8 are revised.</li> </ul>
		52	Figure 8.3 CM1 Register: NOTE 3 of CM11 bit is deleted.
		54	Figure 8.6 CCLKR Register: Location of NOTE 2 is changed and NOTE 3 is added.
		55	Figure 8.7 PM2 Register: NOTE 2 is revised.
			Figure 8.8 PLC0 Register: Function of 011b and 100b in PLC02 to PLC00 bits are revised from "Multiply by 6 and Multiply by 8" to "Do not set a value".
		58	8.1.4 PLL Clock 11th line: 16 MHz is added to PLL clock frequency.
			Table 8.2 Example for Setting PLL Clock Frequencies <ul style="list-style-type: none"> <li>• PLL clock = 16 MHz is added. (8×2, 4×4)</li> <li>• 16 MHz is added to NOTE 1.</li> </ul>
		59	Figure 8.11 Procedure to Use PLL Clock as CPU Clock Source <ul style="list-style-type: none"> <li>• 4th frame: "(To select a 16 MHz or higher PLL clock)" is revised to "(When PLL clock &gt;16 MHz)".</li> </ul>
		61	8.4.1.2 PLL Operation Mode: 1st line <ul style="list-style-type: none"> <li>• The main clock multiplied is revised from "by 2, 4, 6 or 8" to "by 2 or 4".</li> </ul>
		62	Table 8.3 Setting Clock Related Bit and Modes <ul style="list-style-type: none"> <li>• CM21 bit in Low Power Dissipation Mode: Value is revised from "-" to "0".</li> <li>• CM11 bit in Low-Speed Mode, Low Power Dissipation Mode, On-chip Oscillator Mode and On-chip Oscillator Low Power Dissipation Mode: Value is revised from "-" to "0".</li> </ul>
		63	8.4.2 Wait Mode 4th line: "PLL clock" is deleted.
			Table 8.4 Pin Status During Wait Mode <ul style="list-style-type: none"> <li>• Memory Expansion Mode, Microprocessor Mode in ALE: Value is revised from "H" to "L".</li> </ul>
		64	Table 8.5 Interrupts to Exit Wait Mode <ul style="list-style-type: none"> <li>• CAN0/1 Wake-up Interrupt: "in CAN sleep mode" is added.</li> </ul>
		65	8.4.3 Stop Mode <ul style="list-style-type: none"> <li>• CAN0/1 Wake-up interrupt: "(when CAN sleep mode is selected)" is added.</li> </ul> Table 8.6 Pin Status in Stop Mode <ul style="list-style-type: none"> <li>• Memory Expansion Mode, Microprocessor Mode in ALE: Value is revised from "H" to "indeterminate".</li> </ul>

# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.00	Nov. 10, 2004	67	Figure 8.12 State Transition to Stop Mode and Wait Mode <ul style="list-style-type: none"> <li>• Figure is revised.</li> <li>• NOTE 3 is revised.</li> </ul>
		68	Figure 8.13 State Transition in Normal Operation Mode <ul style="list-style-type: none"> <li>• Low-Speed and Low Power Dissipation Mode: "CM7 = 1" is revised to "CM7 = 0" (3 places).</li> <li>• NOTES 2, 6 are revised.</li> </ul>
		71	Figure 8.14 Procedure to Switch Clock Source from On-chip Oscillator to Main Clock is revised.
		77	Table 10.2 Relocatable Vector Tables <ul style="list-style-type: none"> <li>• Interrupt Source: "Software interrupt" is revised to "INT Instruction Interrupt"</li> <li>• NOTES 10, 11 are added.</li> </ul>
		78	Figure 10.3 Interrupt Control Registers (1): NOTES 5, 6, 7 are added.
		79	Figure 10.4 Interrupt Control Registers (2) <ul style="list-style-type: none"> <li>• NOTE 2 is added to C1RECIC/INT5IC, C1TRMIC/S3IC/INT4IC</li> <li>• NOTES 6, 7 are added.</li> </ul>
		87	Figure 10.11 (upper) IFSR0 Register: NOTE 3 is added.
		88	10.9 CAN0/1 Wake-up Interrupt is revised. Figure 10.13 CAN0/1 Wake-up Interrupt Block Diagram is revised.
		91	Figure 11.1 Watchdog Timer Block Diagram: "RESET" is revised to "Internal RESET signal".
		108	Figure 13.6 (upper and middle) ONSF Register, TRGSR Register: NOTE 2 is added.
		109	Table 13.1 Specifications in Timer Mode <ul style="list-style-type: none"> <li>• Specification of Divide Ratio: "TAiMR register" is revised to "TAi register".</li> <li>• Specification of Select Function: "When not counting, the pin outputs a low" is revised to "When TAiS bit is set to "0" (stop counting), the pin outputs a low".</li> </ul>
		110	Table 13.2 Specifications in Event Counter Mode (when not processing two-phase pulse signal) <ul style="list-style-type: none"> <li>• Specification in Select Function: "When not counting, the pin outputs a low" is revised to "When TAiS bit is set to "0" (stop counting), the pin outputs a low".</li> </ul>
		114	13.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing 4th line <ul style="list-style-type: none"> <li>• "the INT2 pin" is revised to "the ZP pin".</li> </ul> Figure 13.10 Two-phase Pulse (A phase and B phase) and Z Phase <ul style="list-style-type: none"> <li>• "INT2 (Z phase)" is revised to "ZP".</li> </ul>
		118	Figure 13.12 TA0MR to TA4MR Registers in PWM Mode <ul style="list-style-type: none"> <li>• Bit name and Function in MR0 bit is revised from "Set to "1" in PWM mode" to "Pulse Output Function Select Bit <sup>(3)</sup>".</li> <li>• NOTE 3 is added.</li> </ul>
		123	Table 13.6 Specifications in Timer Mode <ul style="list-style-type: none"> <li>• Specification in Divide Ratio: "TBiMR register" is revised to "TBi register".</li> </ul>
		129	Figure 14.1 Three-Phase Motor Control Timer Function Block Diagram is revised.
		130	Figure 14.2 INVC0 Register is revised.
		131	Figure 14.3 INVC1 Register: Function of INV13 bit is revised.
		132	Figure 14.4 (upper) IDB0 and IDB1 Registers: (b7-b6) is revised. Figure 14.4 (lower) DTT Register: NOTE 2 is revised.

# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.00	Nov. 10, 2004	134	Figure 14.6 (upper) ICTB2 Register <ul style="list-style-type: none"> <li>•(b7-b4) is revised.</li> <li>•NOTE 3 is added.</li> </ul>
		135	Figure 14.7 (upper) TRGSR Register: NOTE 2 is added.
		136	Figure 14.8 (upper) TA1MR, TA2MR and TA4MR Registers <ul style="list-style-type: none"> <li>•Function of MR1 bit: "Has no effect" is revised to "Set to "0" ".</li> </ul>
		137	Figure 14.9 Triangular Wave Modulation Operation is revised.
		139	15.1 UARTi: "UART0, UART1" in Special mode 3 is deleted.
		140, 141	Figures 15.1 to 15.3 UART0 to 2 Block Diagram are revised.
		142	Figure 15.4 UARTi Transmit/Receive Unit is revised.
		144	Figure 15.6 (lower) U0C0 to U2C0 Registers: NOTES 3, 4 are revised.
		145	Figure 15.7 (upper) U0C1, U1C1 Registers <ul style="list-style-type: none"> <li>•The value of After Reset is revised.</li> <li>•(b5-b4) is revised from "When read, their contents are "0" " to "When read, their contents are indeterminate".</li> <li>•NOTE 1 is added.</li> </ul>
			Figure 15.7 (lower) U2C1 Register: NOTE 1 is added.
		153	15.1.1.1 Counter Measure for Communication Error Occurs is added.
		154	15.1.1.4 Continuous Receive Mode: first to 4th lines are added.
		156	15.1.1.7 CTS/RTS Function is added.
		157	Table 15.5 UART Mode Specifications: NOTE 3 is added.
		159	Table 15.7 I/O Pin Functions <ul style="list-style-type: none"> <li>•Method of Selection in TXDi: "Output dummy data" is revised to "Output "H" ".</li> </ul>
		161	15.1.2.1 Bit Rates and Table 15.9 Example of Bit Rates and Settings are added.
		162	15.1.2.2 Counter Measure for Communication Error Occurs is added.
		164	15.1.2.6 CTS/RTS Function is added.
		176	Table 15.15 Registers to Be Used and Settings in Special Mode 2 <ul style="list-style-type: none"> <li>•"U2LCH" in UiC1 register is revised to "UiLCH".</li> </ul>
		179	Table 15.16 Registers to Be Used and Settings in IE Mode <ul style="list-style-type: none"> <li>•"UiRRM" in UiC1 register is revised to "U2RRM".</li> </ul>
		181	Table 15.17 SIM Mode Specifications: NOTE 3 is added.
		189	Figure 15.39 Polarity of Transfer Clock is revised.
		205	16.2.4 External Operation Amplifier (Op-Amp) Connection Mode: 6th line <ul style="list-style-type: none"> <li>•"Note that the ANEX0 and ANEX1 pins cannot be directly connected to each other." is deleted.</li> </ul>
		206	16.2.6 Output Impedance of Sensor under A/D Conversion is added.
		209	Figure 17.2 (lower) DA0 and DA1 Registers: The value of After Reset are revised.
		216	Figure 19.4 Bit Mapping of Mask Registers in Byte Access: NOTES 1, 2 are added. Figure 19.5 Bit Mapping of Mask Registers in Word Access: NOTES 1, 2 are added.
		217	Figure 19.6 C0MCTLj and C1MCTLj Registers: NOTE 2 is revised.
		218	Figure 19.7 C0CTLR and C1CTLR Registers (upper) <ul style="list-style-type: none"> <li>•NOTE 1 (Rev.1.00) is deleted and NOTES 1, 2, 3 are added.</li> </ul>
			Figure 19.7 C0CTLR and C1CTLR Registers (lower): NOTES 3, 4 are added.

# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.00	Nov. 10, 2004	219	Figure 19.8 C0STR and C1STR Registers (upper): NOTE 2 is added.
		223	19.5 Operational Modes <ul style="list-style-type: none"> <li>• 1st line: "three operational modes" is revised to "four operational modes".</li> <li>• 5th line: "CAN Interface Sleep Mode" is added.</li> </ul> Figure 19.12 Transition Between Operational Modes is revised.
		224	19.5.1 CAN Reset/Initialization Mode is revised.
		224	19.5.2 CAN Operation Mode is revised.
		224	19.5.3 CAN Sleep Mode is revised.
		224	19.5.4 CAN Interface Sleep Mode is added.
		225	19.5.5 Bus Off State is revised.
		231	19.12 Return from Bus Off Function is revised.
		231	19.14 Listen-Only Mode <ul style="list-style-type: none"> <li>• last line: "When listen-only mode is selected, do not request the transmission." is added.</li> </ul>
		233	Figure 19.20 Timing of Receive Data Frame Sequence: Waveform of RecState bit is revised.
		233	19.15.1 Reception: (4) (5) are revised.
		234	Figure 19.21 Timing of Transmit Sequence <ul style="list-style-type: none"> <li>• The position of the number corresponding to the text is revised.</li> </ul>
		234	19.15.2 Transmission: (1) to (4) are revised.
		251	21.2.1 ROM Code Protect Function is revised.
		251	21.2.2 ID Code Check Function is revised.
		252	Figure 21.2 ROMCP Register is revised.
		255	Figure 21.4 (upper) FMR0 Register: The value of After Reset is revised.
		256	21.3.3.1 FMR00 Bit is revised.
		256	21.3.3.8 FMR11 Bit is revised.
		256	21.3.3.9 FMR16 Bit is revised.
		257	Figure 21.5 Setting and Resetting of EW0 Mode is revised.
		257	Figure 21.6 Setting and Resetting of EW1 Mode: NOTE 3 is revised.
		258	Figure 21.7 Processing Before and After Low Power Dissipation Mode: NOTE 4 is added.
		260	21.3.4.12 Low Power Dissipation Mode and On-chip Oscillator Low Power Dissipation Mode is revised.
		261	Table 21.4 Software Commands: NOTE 2 is deleted.
		262	21.3.5.4 Program Command (40h) <ul style="list-style-type: none"> <li>• From bottom to 3rd line: "read command" is revised to "read array command".</li> </ul>
		265	Figure 21.11 Read Lock Bit Status Command <ul style="list-style-type: none"> <li>• "Locked", "Not locked" are revised to "Block is locked", "Block is not locked".</li> </ul>
		266	21.3.7.1 Sequencer Status (SR7 and FMR00 Bits) is revised.
		271	Table 21.7 Pin Functions for Standard Serial I/O Mode <ul style="list-style-type: none"> <li>• "VCC" is revised to "VCC1", and "VCC2" is added.</li> <li>• VCC1, VCC2, VSS: VCC apply condition is added.</li> </ul>
		273	Figure 21.14 Pin Connections for Standard Serial I/O Mode (2) is added.
		274	Figure 21.16 Circuit Application in Standard Serial I/O Mode 2: "RESET" is added.
		276	Table 21.8 Pin Functions for CAN I/O Mode <ul style="list-style-type: none"> <li>• "VCC" is revised to "VCC1", and "VCC2" is added.</li> <li>• VCC1, VCC2, VSS: VCC apply condition is added.</li> </ul>

# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.00	Nov. 10, 2004	278	Figure 21.18 Pin Connections for CAN I/O Mode (2) is added.
		280	Table 21.9 Flash Memory Version Electrical Characteristics •Parameter is added and the value of some item is revised.
		281	Table 22.1 Absolute Maximum Ratings •"Flash Program Erase" in Operating Ambient Temperature is added.
		283	Table 22.3 Recommended Operating Conditions (2) •Parameters of Power Supply Ripple are added. •NOTE 4 is revised.
			Figure 22.1 Timing of Voltage Fluctuation is added.
		284	Table 22.4 Electrical Characteristics (1): Hysteresis •"CLK4" is revised to "CLK3", and "TA2OUT" is revised to "TA0OUT". •Max. of Standard in RESET is revised from "2.2" to "2.5". •XIN is added.
		286	Table 22.6 A/D Conversion Characteristics: "Tolerance Level Impedance" is added.
		287	Table 22.8 Power Supply Circuit Timing Characteristics: " $t_{d(M-L)}$ " is deleted. Figure 22.2 Power Supply Circuit Timing Diagram is added.
		288	Table 22.10 Memory Expansion Mode and Microprocessor Mode: " $t_{d(BCLK-HLDA)}$ " is deleted.
		290	Table 22.21 Serial I/O: Min. of standard in $t_{su(D-C)}$ is revised from "30" to "70".
		291	Table 22.23 Memory Expansion Mode and Microprocessor Mode (for setting with no wait) •Max. of Standard in $t_{d(BCLK-ALE)}$ is revised from "25" to "15". • $t_{d(BCLK-HLDA)}$ is added.
		292	Table 22.24 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access) •Max. of Standard in $t_{d(BCLK-ALE)}$ is revised from "25" to "15". • $t_{d(BCLK-HLDA)}$ is added.
		293	Table 22.25 Memory Expansion Mode and Microprocessor Mode (for 2- to 3-wait setting, external area access and multiplexed bus selection) • $t_{d(BCLK-HLDA)}$ is added. •Max. of Standard in $t_{d(BCLK-ALE)}$ is revised from "25" to "15".
		294	Figure 22.4 Timing Diagram (1): "XIN input" is added.
		296, 297	Figures 22.6 and 22.7 Timing Diagram (3) (4): "DB" in Read timing is revised to "DBi".
		298, 299	Figures 22.8 and 22.9 Timing Diagram (5) (6): "DB" in Write timing is revised to "DBi".
		301	Figure 22.11 Timing Diagram (8) •"ADi/DB" in Read/Write timing is revised to "ADi/DBi".
		302	23.1 External Bus: The description of the external ROM version is deleted.
		303	23.2 PLL Frequency Synthesizer is revised.
		304	23.3 Power Control •2nd item is added. (Set the MR0 bit in the TAI <sub>MR</sub> register to ●●●) •4th item is revised. (Wait for main clock oscillation ●●●) •Section of "External clock" is deleted.
		316	23.8.2.1 Special Mode 1 (I <sup>2</sup> C Mode) is added.
		317	23.8.3 SI/O3 is added.



# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.00	Nov. 10, 2004	319	23.9 A/D Converter: last item is added. (When setting the ADST bit to ●●)
		322	23.10.2 Performing CAN Configuration is added.
		323	23.10.3 Suggestions to Reduce Power Consumption is added.
		327	23.13 Mask ROM Version is added.
		328	23.14.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode is revised.
		330	23.15 Flash Memory Programming Using Boot Program is added.
		331	23.16 Noise is added.
		332	Appendix 1. Package Dimensions: 100P6Q-A is added.
2.10	Jun. 24, 2005	—	Revised edition issued * The contents of product are revised. (Normal-ver. is added.) * Revised parts and revised contents are as follows (except for expressional change).
		2	Table 1.1 Performance outline of M16C/6N Group (M16C/6N4) • Performance outline of Normal-ver. is added.
		4	Table 1.2 Product List is revised. (Normal-ver. is added.) Figure 1.2 Type No., Memory Size, and Package: • "(no): Normal-ver." is added to Characteristics.
		19	Figure 4.7 SFR Information (7): NOTE 1 is revised.
		53	Figure 8.4 CM2 Register: The value of After Reset is revised.
		68	Figure 8.13 State Transition in Normal Operation Mode: NOTE 7 is revised.
		217	Figure 19.6 C0MCTLj and C1MCTLj Registers • RemActive bit: Function is revised. • RspLock bit: Bit Name is revised. • NOTE 2 is revised.
		218	Figure 19.7 C0CTLR and C1CTLR Registers (upper) • LoopBack bit: The expression of Function is revised. • BasicCAN bit: The expression of Function is revised. Figure 19.7 C0CTLR and C1CTLR Registers (lower) • TSPreScale bit: Bit Symbol is revised. ("Bit1, Bit0" is deleted.) • TSReset bit: The expression of Function is revised. • RetBusOff bit: The expression of Function is revised. • RXOnly bit: The expression of Function is revised.
		219	Figure 19.8 C0STR and C1STR Registers (upper): NOTE 1 is deleted. Figure 19.8 C0STR and C1STR Registers (lower) • State_LoopBack bit: The expression of Function is revised. • State_BasicCAN bit: The expression of Function is revised.
		222	Figure 19.11 C0RECR, C1RECR Registers, C0TECR, C1TECR Registers, C0TSR, C1TSR Registers, and C0AFS, C1AFS Registers • C0RECR, C1RECR Registers: NOTE 2 is deleted. • C0TECR, C1TECR Registers: NOTE 1 is deleted. • C0TSR, C1TSR Registers: NOTE 1 is deleted.
		233	19.15.1 Reception (1): "(refer to 19.15.2 Transmission)" is deleted.

# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.10	Jun. 24, 2005	238	Figure 20.1 I/O Ports (1): "P7_0" in 4th figure is deleted.
		240	Figure 20.3 I/O Ports (3): "P7_0" is added to middle figure.
		242	Figure 20.6 I/O Pins: NOTE 1 is deleted.
		284	Table 22.4 Electrical Characteristics (1) <ul style="list-style-type: none"> <li>Measuring Condition of <math>V_{OL}</math> is revised from "<math>L_{OL} = -200\mu A</math>" to "<math>L_{OL} = 200\mu A</math>".</li> </ul>
		285	Table 22.5 Electrical Characteristics (2): Mask ROM (5th item) <ul style="list-style-type: none"> <li>"f(XCIN)" is changed to "f(BCLK)".</li> </ul>
		286	Table 22.6 A/D Conversion Characteristics: "Tolerance Level Impedance" is deleted.
2.30	Oct. 24, 2005	–	Revised edition issued * Electric Characteristics of Normal-ver. is added. * Revised parts and revised contents are as follows (except for expressional change).
		1	1.1 Applications: Comment of Normal-ver. is added.
		4	Table 1.2 Product List: NOTE 1 is added.
		7, 8	Tables 1.3 and 1.4 Pin Characteristics (1)(2) are added.
		9	Table 1.5 Pin Description (1) <ul style="list-style-type: none"> <li>3.0 to 3.6 V (Normal-ver.) is added to Description of Power supply input.</li> </ul>
		31 to 33	5. Reset: Layout is changed.
		33	5.5 Internal Space is added.
		44	7.2.6 RDY Signal: Last sentence is revised.
		51	Table 8.1 Clock Generating Circuit Specifications <ul style="list-style-type: none"> <li>Clock Frequency in PLL Frequency Synthesizer: 24 MHz <sup>(1)</sup> is added.</li> <li>NOTE 1 is added.</li> </ul>
		57	Figure 8.8 PLC0 Register <ul style="list-style-type: none"> <li>PLC02 to PLC00 bits: Function of 011b is revised.</li> <li>NOTE 4 is added.</li> </ul>
		58	Figure 8.9 Examples of Main Clock Connection Circuit is revised.
		59	Figure 8.10 Examples of Sub Clock Connection Circuit is revised.
		60	8.1.4 PLL Clock <ul style="list-style-type: none"> <li>9th line: The sentence (When the PLL ... to) is added.</li> <li>12th line: 24 MHz and NOTE 1 is added to PLL clock frequency.</li> <li>NOTE 1 is added.</li> </ul> Figure 8.2 Example for Setting PLL Clock Frequencies <ul style="list-style-type: none"> <li>24 MHz is added to PLL clock.</li> <li>24 MHz is added to NOTE 1.</li> <li>NOTES 2 and 3 are added.</li> </ul>
		63	8.4.1.2 PLL Operation Mode <ul style="list-style-type: none"> <li>1st line: The main clock multiplied by "6" and NOTE 1 is added.</li> </ul>
		64	8.4.1.6 On-chip Oscillator Mode: Last sentence (When the operation mode is ...) is added. 8.4.1.7 On-chip Oscillator Low Power Dissipation Mode: Last sentence (When the operation mode is ...) is deleted.
		67	Table 8.6 Interrupts to Stop Mode and Use Conditions is added.
		70	Figure 8.13 State Transition in Normal Operation Mode: NOTE 7 is deleted.

# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.30	Oct. 24, 2005	86	10.5.8 Returning from an Interrupt Routine: Last sentence (Register bank ...) is added. 10.5.9 Interrupt Priority: First sentence (If two or more...) is revised. 10.5.10 Interrupt Priority Resolution Circuit: First sentence (The interrupt priority level ...) is revised.
		89	Figure 10.11 IFSR1 Register (upper) • IFSR17: NOTE 2 is added to Bit Name. • NOTE 2 is revised.
		96	Table 12.1 DMAC Specifications: DMA transfer Cycles is added.
		100	12.1.3 Effect of Software Wait: 3rd to 9th lines is moved from next section of 12.1.4.
		120	Figure 13.12 TA0MR to TA4MR Registers in PWM Mode: b2 is revised from "1" to "(blank)".
		131	Figure 14.1 Three-Phase Motor Control Timer Function Block Diagram is revised.
		132	Figure 14.2 INVC0 Register: NOTES 5 and 6 are revised.
		145	Figure 15.5 U0BRG to U2BRG Registers (lower): NOTE 3 is added.
		146	Figure 15.6 U0C0 to U2C0 Registers (lower): NOTE 5 is added.
		163	Table 15.9 Example of Bit Rates and Settings: 24 MHz and NOTE 1 is added.
		189	Figure 15.37 S3C Register (upper): NOTE 5 is added. Figure 15.37 S3BRG Register (middle): NOTE 3 is added.
		193	Table 16.1 A/D Converter Performance • Performance of Integral Nonlinearity Error: "When AVCC = VREF = 3.3 V" is added.
		194	Figure 16.1 A/D Converter Block Diagram • ADGSEL1 to ADGSEL0 (right/lower) is revised from "10b" to "11b".
		208	16.2.6 Output Impedance of Sensor under A/D Conversion • 10th line: f(XIN) is revised to f( $\phi$ AD).
		209	Figure 16.10 Analog Input Pin and External Sensor Equivalent Circuit • fAD is revised to $\phi$ AD.
		210	Figure 17.1 D/A Converter Block Diagram is revised.
		211	Figure 17.2 DA0 and DA1 Registers: Setting Range is added. Figure 17.3 D/A Converter Equivalent Circuit: NOTE 2 is added.
		213	Figure 18.3 CRC Calculation: Details of CRC operation is revised.
		224	Figure 19.11 C0TECR, C1TECR Registers (2nd register): NOTE 1 is added.
		229	Table 19.2 Examples of Bit-rate: 24 MHz and NOTE 2 is added.
		247	Figure 20.9 PUR1 Register (middle): Value of After Reset is revised.
		252	Figure 21.1 Flash Memory Block Diagram is revised.
		254	Figure 21.2 ROMCP Register is revised.
		255	Table 21.3 EW0 Mode and EW1 Mode: NOTE 1 is revised.
		256	21.3.2 EW1 Mode: Last sentence (When an erase/program ...) is added.
		258	21.3.3.4 FMSTP Bit • 8th line: Procedure to change the FMSTP bit setting (1) to (4) are added.
		261	Figure 21.7 Processing Before and After Low Power Dissipation Mode or On-chip Oscillator Low Power Dissipation Mode • Title, First and second frames (left) and top of right: "on-chip oscillator low power dissipation mode" is added.

# REVISION HISTORY

# M16C/6N Group (M16C/6N4) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.30	Oct. 24, 2005	263	21.3.4.11 Stop Mode is revised.
			21.3.4.12 Low Power Dissipation Mode and On-chip Oscillator Low Power Dissipation Mode is partly revised.
		266	21.3.5.5 Block Erase Command: Last sentence (Also execute ...) is added.
			Figure 21.9 Block Erase Command: NOTES 2 and 3 are added.
		272	Figure 21.12 Full Status Check and Handling Procedure for Each Error
			• Erase error: (4) is added.
		274	Table 21.7 Pin Functions for Standard Serial I/O Mode
			• Description of VCC1, VCC2, VSS is revised.
			• Description of P8_4 is revised.
			• NOTE 1 is revised.
			• NOTE 2 is added.
		277	Figures 21.15 and 21.16 Circuit Application in Serial I/O Mode 1/2
			• “VCC1” and “VCC2” are added.
		279	Table 21.8 Pin Functions for CAN I/O Mode
			• Description of VCC1, VCC2, VSS is revised.
			• Description of P8_4 is revised.
			• NOTE 1 is added.
		282	Figure 21.19 Circuit Application in CAN I/O Mode: “VCC1” and “VCC2” are added.
		283	Table 21.9 Flash Memory Version Electrical Characteristics
			• Measuring condition is revised in word program time and block erase time.
		284	21.7.2 Electrical Characteristics (Normal-ver.) is added.
		306 to 341	22.2 Electrical Characteristics (Normal-ver.) is added.
		344	23.3 Power Control: 3rd and 4th items (When entering wait mode ... and When entering stop mode ...) are revised.
		359	Figure 23.2 Use of Capacitors to Reduce Noise is partly revised.
		360	23.9 A/D Converter: Last item (The applied intermediate ...) is added.
		366	23.11 Programmable I/O Ports: 4th and 5th items (Indeterminate values ... and When the PM01 ...) are added.
		369	23.14.2 Stop Mode is revised.
			23.14.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode is partly revised.
			23.14.8 Operation Speed is revised.

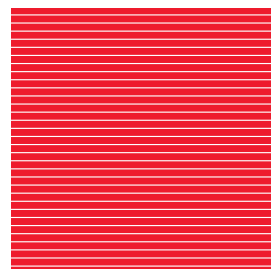
---

**M16C/6N Group (M16C/6N4) Hardware Manual**

**Publication Data :   Rev.1.00   May 30, 2003  
                             Rev.2.30   Oct 24, 2005**

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

# M16C/6N Group (M16C/6N4) Hardware Manual



Renesas Technology Corp.  
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan