

16

M16C/6N Group
(M16C/6NK, M16C/6NM)
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

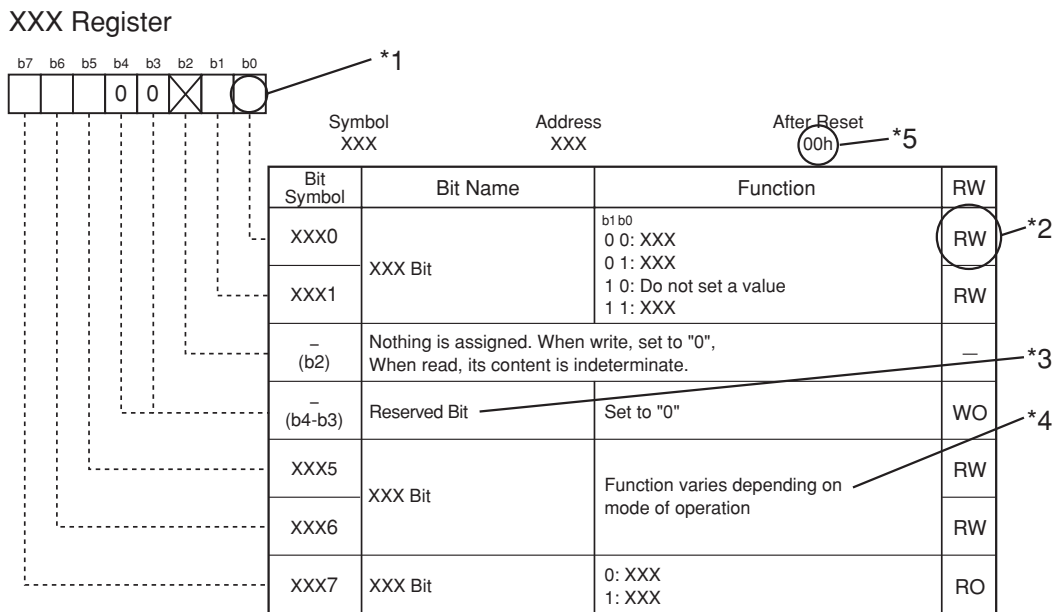
1. Introduction

This hardware manual provides detailed information on the M16C/6N Group (M16C/6NK, M16C/6NM) of microcomputers.

Users are expected to have basic knowledge of electric circuits, logical circuits and microcomputers.

2. Register Diagram

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



*1

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

0 : Set to "0"

1 : Set to "1"

X : Nothing is assigned

*2

RW : Read and write

RO : Read only

WO : Write only

– : Nothing is assigned

*3

- Reserved bit
Reserved bit. Set to specified value.

*4

- Nothing is assigned
Nothing is assigned to the bit concerned. As the bit may be use for future functions, set to "0" when writing to this bit.
- Do not set to this value
The operation is not guaranteed when a value is set.
- Function varies depending on mode of operation
Bit function varies depending on peripheral function mode.
Refer to respective register for each mode.

*5

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

3. M16C Family Documents

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

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

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	4
1.4 Product List	5
1.5 Pin Configuration	6
1.6 Pin Description	8
2. Central Processing Unit (CPU)	10
2.1 Data Registers (R0, R1, R2, and R3)	10
2.2 Address Registers (A0 and A1)	10
2.3 Frame Base Register (FB)	11
2.4 Interrupt Table Register (INTB)	11
2.5 Program Counter (PC)	11
2.6 User Stack Pointer (USP), Interrupt Stack Pointer (ISP)	11
2.7 Static Base Register (SB)	11
2.8 Flag Register (FLG)	11
2.8.1 Carry Flag (C Flag)	11
2.8.2 Debug Flag (D Flag)	11
2.8.3 Zero Flag (Z Flag)	11
2.8.4 Sign Flag (S Flag)	11
2.8.5 Register Bank Select Flag (B Flag)	11
2.8.6 Overflow Flag (O Flag)	11
2.8.7 Interrupt Enable Flag (I Flag)	11
2.8.8 Stack Pointer Select Flag (U Flag)	11
2.8.9 Processor Interrupt Priority Level (IPL)	11
2.8.10 Reserved Area	11
3. Memory	12
4. Special Function Register (SFR)	13
5. Reset	29
5.1 Hardware Reset	29
5.1.1 Reset on a Stable Supply Voltage	29
5.1.2 Power-on Reset	29
5.2 Software Reset	29
5.3 Watchdog Timer Reset	29
5.4 Oscillation Stop Detection Reset	29
6. Processor Mode	32
7. Clock Generating Circuit	35
7.1 Types of Clock Generating Circuit	35
7.1.1 Main Clock	43
7.1.2 Sub Clock	44
7.1.3 On-chip Oscillator Clock	45
7.1.4 PLL Clock	45

7.2 CPU Clock and Peripheral Function Clock	47
7.2.1 CPU Clock and BCLK	47
7.2.2 Peripheral Function Clock	47
7.3 Clock Output Function	47
7.4 Power Control	48
7.4.1 Normal Operation Mode	48
7.4.2 Wait Mode	50
7.4.3 Stop Mode	52
7.5 Oscillation Stop and Re-oscillation Detection Function	57
7.5.1 Operation When CM27 Bit = 0 (Oscillation Stop Detection Reset)	57
7.5.2 Operation When CM27 Bit = 1 (Oscillation Stop, Re-oscillation Detection Interrupt)	57
7.5.3 How to Use Oscillation Stop and Re-oscillation Detection Function	58
8. Protection	59
9. Interrupt	60
9.1 Type of Interrupts	60
9.2 Software Interrupts	61
9.2.1 Undefined Instruction Interrupt	61
9.2.2 Overflow Interrupt	61
9.2.3 BRK Interrupt	61
9.2.4 INT Instruction Interrupt	61
9.3 Hardware Interrupts	62
9.3.1 Special Interrupts	62
9.3.2 Peripheral Function Interrupts	62
9.4 Interrupts and Interrupt Vector	63
9.4.1 Fixed Vector Tables	63
9.4.2 Relocatable Vector Tables	63
9.5 Interrupt Control	65
9.5.1 I Flag	67
9.5.2 IR Bit	67
9.5.3 ILVL2 to ILVL0 Bits and IPL	67
9.5.4 Interrupt Sequence	68
9.5.5 Interrupt Response Time	69
9.5.6 Variation of IPL when Interrupt Request is Accepted	69
9.5.7 Saving Registers	70
9.5.8 Returning from an Interrupt Routine	71
9.5.9 Interrupt Priority	71
9.5.10 Interrupt Priority Resolution Circuit	71
9.6 $\overline{\text{INT}}$ Interrupt	73
9.7 $\overline{\text{NMI}}$ Interrupt	77
9.8 Key Input Interrupt	77
9.9 CAN0/1 Wake-up Interrupt	77
9.10 Address Match Interrupt	78
10. Watchdog Timer	80
10.1 Count Source Protective Mode	81

11. DMAC	82
11.1 Transfer Cycle	87
11.1.1 Effect of Source and Destination Addresses	87
11.1.2 Effect of Software Wait	87
11.2 DMA Transfer Cycles	89
11.3 DMA Enable	90
11.4 DMA Request	90
11.5 Channel Priority and DMA Transfer Timing	91
12. Timers	92
12.1 Timer A	94
12.1.1 Timer Mode	98
12.1.2 Event Counter Mode	99
12.1.3 One-shot Timer Mode	104
12.1.4 Pulse Width Modulation (PWM) Mode	106
12.2 Timer B	109
12.2.1 Timer Mode	112
12.2.2 Event Counter Mode	113
12.2.3 Pulse Period and Pulse Width Measurement Mode	114
13. Three-Phase Motor Control Timer Function	117
14. Serial I/O	128
14.1 UARTi	128
14.1.1 Clock Synchronous Serial I/O Mode	138
14.1.2 Clock Asynchronous Serial I/O (UART) Mode	146
14.1.3 Special Mode 1 (I ² C Mode)	154
14.1.4 Special Mode 2	163
14.1.5 Special Mode 3 (IE Mode)	168
14.1.6 Special Mode 4 (SIM Mode) (UART2)	170
14.2 SI/Oi	175
14.2.1 SI/Oi Operation Timing	179
14.2.2 CLK Polarity Selection	179
14.2.3 Functions for Setting an SOUTi Initial Value	180
15. A/D Converter	181
15.1 Mode Description	185
15.1.1 One-shot Mode	185
15.1.2 Repeat Mode	187
15.1.3 Single Sweep Mode	189
15.1.4 Repeat Sweep Mode 0	191
15.1.5 Repeat Sweep Mode 1	193
15.2 Function	195
15.2.1 Resolution Select Function	195
15.2.2 Sample and Hold	195
15.2.3 Extended Analog Input Pins	195
15.2.4 External Operation Amplifier (Op-Amp) Connection Mode	195
15.2.5 Current Consumption Reducing Function	196
15.2.6 Output Impedance of Sensor under A/D Conversion	196
16. D/A Converter	198
17. CRC Calculation	200

18. CAN Module	202
18.1 CAN Module-Related Registers	203
18.1.1 CAN Message Box	203
18.1.2 Acceptance Mask Registers	203
18.1.3 CAN SFR Registers	203
18.2 CANi Message Box	204
18.3 Acceptance Mask Registers	206
18.4 CAN SFR Registers	207
18.5 Operational Modes	213
18.5.1 CAN Reset/Initialization Mode	213
18.5.2 CAN Operation Mode	214
18.5.3 CAN Sleep Mode	214
18.5.4 CAN Interface Sleep Mode	214
18.5.5 Bus Off State	215
18.6 Configuration CAN Module System Clock	216
18.7 Bit Timing Configuration	216
18.8 Bit-rate	217
18.9 Acceptance Filtering Function and Masking Function	218
18.10 Acceptance Filter Support Unit (ASU)	219
18.11 Basic CAN Mode	220
18.12 Return from Bus off Function	221
18.13 Time Stamp Counter and Time Stamp Function	221
18.14 Listen-Only Mode	221
18.15 Reception and Transmission	222
18.15.1 Reception	223
18.15.2 Transmission	224
18.16 CAN Interrupt	225
19. Programmable I/O Ports	226
19.1 PDi Register	227
19.2 Pi Register, PC14 Register	227
19.3 PURj Register	227
19.4 PCR Register	227
20. Flash Memory Version	238
20.1 Memory Map	239
20.1.1 Boot Mode	240
20.2 Functions to Prevent Flash Memory from Rewriting	240
20.2.1 ROM Code Protect Function	240
20.2.2 ID Code Check Function	240
20.3 CPU Rewrite Mode	242
20.3.1 EW0 Mode	243
20.3.2 EW1 Mode	243
20.3.3 FMR0, FMR1 Registers	244
20.3.4 Precautions on CPU Rewrite Mode	248
20.3.5 Software Commands	250
20.3.6 Data Protect Function	255
20.3.7 Status Register (SRD Register)	255
20.3.8 Full Status Check	257

20.4 Standard Serial I/O Mode	259
20.4.1 ID Code Check Function	259
20.4.2 Example of Circuit Application in Standard Serial I/O Mode	263
20.5 Parallel I/O Mode	264
20.5.1 User ROM and Boot ROM Areas	264
20.5.2 ROM Code Protect Function	264
20.6 CAN I/O Mode	265
20.6.1 ID Code Check Function	265
20.6.2 Example of Circuit Application in CAN I/O Mode	268
21. Electrical Characteristics	269
22. Usage Precaution	279
22.1 SFR	279
22.2 External Clock	280
22.3 PLL Frequency Synthesizer	281
22.4 Power Control	282
22.5 Oscillation Stop, Re-oscillation Detection Function	284
22.6 Protection	285
22.7 Interrupt	286
22.7.1 Reading Address 00000h	286
22.7.2 Setting SP	286
22.7.3 $\overline{\text{NMI}}$ Interrupt	286
22.7.4 Changing Interrupt Generate Factor	287
22.7.5 $\overline{\text{INT}}$ Interrupt	287
22.7.6 Rewrite Interrupt Control Register	288
22.7.7 Watchdog Timer Interrupt	288
22.8 DMAC	289
22.8.1 Write to DMAE Bit in DMiCON Register	289
22.9 Timers	290
22.9.1 Timer A	290
22.9.2 Timer B	294
22.10 Three-Phase Motor Control Timer Function	296
22.11 Serial I/O	297
22.11.1 Clock Synchronous Serial I/O Mode	297
22.11.2 Special Modes	298
22.11.3 SI/Oi	299
22.12 A/D Converter	300
22.13 CAN Module	302
22.13.1 Reading CiSTR Register	302
22.13.2 Performing CAN Configuration	304
22.13.3 Suggestions to Reduce Power Consumption	305
22.13.4 CAN Transceiver in Boot Mode	306
22.14 Programmable I/O Ports	307
22.15 Dedicated Input Pin	308
22.16 Electrical Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers	309
22.17 Mask ROM Version	310

22.18 Flash Memory Version	311
22.18.1 Functions to Prevent Flash Memory from Rewriting	311
22.18.2 Stop Mode	311
22.18.3 Wait Mode	311
22.18.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode	311
22.18.5 Writing Command and Data	311
22.18.6 Program Command	311
22.18.7 Lock Bit Program Command	311
22.18.8 Operation Speed	312
22.18.9 Prohibited Instructions	312
22.18.10 Interrupt	312
22.18.11 How to Access	312
22.18.12 Rewriting in User ROM Area	312
22.18.13 DMA Transfer	312
22.19 Flash Memory Programming Using Boot Program	313
22.19.1 Programming Using Serial I/O Mode	313
22.19.2 Programming Using CAN I/O Mode	313
22.20 Noise	314
 Appendix 1. Package Dimensions	 315
 Register Index	 317

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

SFR Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	32
0005h	Processor Mode Register 1	PM1	33
0006h	System Clock Control Register 0	CM0	37
0007h	System Clock Control Register 1	CM1	38
0008h			
0009h	Address Match Interrupt Enable Register	AIER	79
000Ah	Protect Register	PRCR	59
000Bh			
000Ch	Oscillation Stop Detection Register	CM2	39
000Dh			
000Eh	Watchdog Timer Start Register	WDTS	81
000Fh	Watchdog Timer Control Register	WDC	81
0010h			
0011h	Address Match Interrupt Register 0	RMAD0	79
0012h			
0013h			
0014h			
0015h	Address Match Interrupt Register 1	RMAD1	79
0016h			
0017h			
0018h			
0019h			
001Ah			
001Bh			
001Ch	PLL Control Register 0	PLC0	42
001Dh			
001Eh	Processor Mode Register 2	PM2	41
001Fh			
0020h			
0021h	DMA0 Source Pointer	SAR0	86
0022h			
0023h			
0024h			
0025h	DMA0 Destination Pointer	DAR0	86
0026h			
0027h			
0028h	DMA0 Transfer Counter	TCR0	86
0029h			
002Ah			
002Bh			
002Ch	DMA0 Control Register	DM0CON	85
002Dh			
002Eh			
002Fh			
0030h			
0031h	DMA1 Source Pointer	SAR1	86
0032h			
0033h			
0034h			
0035h	DMA1 Destination Pointer	DAR1	86
0036h			
0037h			
0038h	DMA1 Transfer Counter	TCR1	86
0039h			
003Ah			
003Bh			
003Ch	DMA1 Control Register	DM1CON	85
003Dh			
003Eh			
003Fh			

The blank areas are reserved.

Address	Register	Symbol	Page
0040h			
0041h	CAN0/1 Wake-up Interrupt Control Register	C01WKIC	65
0042h	CAN0 Successful Reception Interrupt Control Register	C0RECIC	65
0043h	CAN0 Successful Transmission Interrupt Control Register	C0TRMIC	65
0044h	INT3 Interrupt Control Register	INT3IC	66
0045h	Timer B5 Interrupt Control Register	TB5IC	65
	SI/O5 Interrupt Control Register	S5IC	65
0046h	Timer B4 Interrupt Control Register	TB4IC	65
	UART1 Bus Collision Detection Interrupt Control Register	U1BCNIC	65
0047h	Timer B3 Interrupt Control Register	TB3IC	65
	UART0 Bus Collision Detection Interrupt Control Register	U0BCNIC	65
0048h	CAN1 Successful Reception Interrupt Control Register	C1RECIC	66
	SI/O4 Interrupt Control Register	S4IC	66
	INT5 Interrupt Control Register	INT5IC	66
0049h	CAN1 Successful Transmission Interrupt Control Register	C1TRMIC	66
	SI/O3 Interrupt Control Register	S3IC	66
	INT4 Interrupt Control Register	INT4IC	66
004Ah	UART2 Bus Collision Detection Interrupt Control Register	U2BCNIC	65
004Bh	DMA0 Interrupt Control Register	DM0IC	65
004Ch	DMA1 Interrupt Control Register	DM1IC	65
004Dh	CAN0/1 Error Interrupt Control Register	C01ERRIC	65
004Eh	A/D Conversion Interrupt Control Register	ADIC	65
	Key Input Interrupt Control Register	KUPIC	65
004Fh	UART2 Transmit Interrupt Control Register	S2TIC	65
0050h	UART2 Receive Interrupt Control Register	S2RIC	65
0051h	UART0 Transmit Interrupt Control Register	S0TIC	65
0052h	UART0 Receive Interrupt Control Register	S0RIC	65
0053h	UART1 Transmit Interrupt Control Register	S1TIC	65
0054h	UART1 Receive Interrupt Control Register	S1RIC	65
0055h	Timer A0 Interrupt Control Register	TA0IC	65
0056h	Timer A1 Interrupt Control Register	TA1IC	65
0057h	Timer A2 Interrupt Control Register	TA2IC	66
	INT7 Interrupt Control Register	INT7IC	66
0058h	Timer A3 Interrupt Control Register	TA3IC	66
	INT6 Interrupt Control Register	INT6IC	66
0059h	Timer A4 Interrupt Control Register	TA4IC	65
005Ah	Timer B0 Interrupt Control Register	TB0IC	65
	SI/O6 Interrupt Control Register	S6IC	65
005Bh	Timer B1 Interrupt Control Register	TB1IC	66
	INT8 Interrupt Control Register	INT8IC	66
005Ch	Timer B2 Interrupt Control Register	TB2IC	65
005Dh	INT0 Interrupt Control Register	INT0IC	66
005Eh	INT1 Interrupt Control Register	INT1IC	66
005Fh	INT2 Interrupt Control Register	INT2IC	66
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h			
0069h			
006Ah			
006Bh			
006Ch			
006Dh			
006Eh			
006Fh			
0070h			
0071h			
0072h			
0073h			
0074h			
0075h			
0076h			
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh			
007Fh			

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

204
205

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

204
205

Address	Register	Symbol	Page
0100h	CAN0 Message Box 10: Identifier / DLC		204 205
0101h			
0102h			
0103h			
0104h			
0105h			
0106h	CAN0 Message Box 10: Data Field		
0107h			
0108h			
0109h			
010Ah			
010Bh			
010Ch			
010Dh			
010Eh	CAN0 Message Box 10: Time Stamp		
010Fh			
0110h	CAN0 Message Box 11: Identifier / DLC		
0111h			
0112h			
0113h			
0114h			
0115h			
0116h	CAN0 Message Box 11: Data Field		
0117h			
0118h			
0119h			
011Ah			
011Bh			
011Ch			
011Dh			
011Eh	CAN0 Message Box 11: Time Stamp		
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			
012Dh			
012Eh	CAN0 Message Box 12: Time Stamp		
012Fh			
0130h	CAN0 Message Box 13: Identifier / DLC		
0131h			
0132h			
0133h			
0134h			
0135h			
0136h	CAN0 Message Box 13: Data Field		
0137h			
0138h			
0139h			
013Ah			
013Bh			
013Ch			
013Dh			
013Eh	CAN0 Message Box 13: Time Stamp		
013Fh			

Address	Register	Symbol	Page
0140h	CAN0 Message Box 14: Identifier /DLC		204 205
0141h			
0142h			
0143h			
0144h			
0145h			
0146h	CAN0 Message Box 14: Data Field		
0147h			
0148h			
0149h			
014Ah			
014Bh			
014Ch			
014Dh			
014Eh	CAN0 Message Box 14: Time Stamp		
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			
015Dh			
015Eh	CAN0 Message Box 15: Time Stamp		
015Fh			
0160h	CAN0 Global Mask Register	C0GMR	206
0161h			
0162h			
0163h			
0164h			
0165h			
0166h	CAN0 Local Mask A Register	C0LMAR	206
0167h			
0168h			
0169h			
016Ah			
016Bh			
016Ch	CAN0 Local Mask B Register	C0LMBR	206
016Dh			
016Eh			
016Fh			
0170h			
0171h			
0172h			
0173h			
0174h			
0175h			
0176h			
0177h			
0178h			
0179h			
017Ah			
017Bh			
017Ch			
017Dh			
017Eh			
017Fh			

The blank areas are reserved.

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	244
01B6h			
01B7h	Flash Memory Control Register 0	FMR0	244
01B8h			
01B9h	Address Match Interrupt Register 2	RMAD2	79
01BAh			
01BBh	Address Match Interrupt Enable Register 2	AIER2	79
01BCh			
01BDh	Address Match Interrupt Register 3	RMAD3	79
01BEh			
01BFh			

Address	Register	Symbol	Page
01C0h	Timer B3, B4, B5 Count Start Flag	TBSR	111
01C1h			
01C2h			
01C3h	Timer A1-1 Register	TA11	122
01C4h			
01C5h	Timer A2-1 Register	TA21	122
01C6h			
01C7h	Timer A4-1 Register	TA41	122
01C8h	Three-Phase PWM Control Register 0	INVC0	119
01C9h	Three-Phase PWM Control Register 1	INVC1	120
01CAh	Three-Phase Output Buffer Register 0	IDB0	121
01CBh	Three-Phase Output Buffer Register 1	IDB1	121
01CCh	Dead Time Timer	DTT	121
01CDh	Timer B2 Interrupt Occurrence Frequency Set Counter	ICTB2	123
01CEh			
01CFh	Interrupt Cause Select Register 2	IFSR2	76
01D0h			
01D1h	Timer B3 Register	TB3	110
01D2h			
01D3h	Timer B4 Register	TB4	110
01D4h			
01D5h	Timer B5 Register	TB5	110
01D6h	SI/O6 Transmit/Receive Register	S6TRR	176
01D7h			
01D8h	SI/O6 Control Register	S6C	176
01D9h	SI/O6 Bit Rate Generator	S6BRG	176
01DAh	SI/O3, 4, 5, 6 Transmit/Receive Register	S3456TRR	177
01DBh	Timer B3 Mode Register	TB3MR	110
01DCh	Timer B4 Mode Register	TB4MR	112
01DDh	Timer B5 Mode Register	TB5MR	113
01DEh	Interrupt Cause Select Register 0	IFSR0	115
01DFh	Interrupt Cause Select Register 1	IFSR1	74
01E0h	SI/O3 Transmit/Receive Register	S3TRR	75
01E1h			
01E2h	SI/O3 Control Register	S3C	176
01E3h	SI/O3 Bit Rate Generator	S3BRG	176
01E4h	SI/O4 Transmit/Receive Register	S4TRR	176
01E5h			
01E6h	SI/O4 Control Register	S4C	176
01E7h	SI/O4 Bit Rate Generator	S4BRG	176
01E8h	SI/O5 Transmit/Receive Register	S5TRR	176
01E9h			
01EAh	SI/O5 Control Register	S5C	176
01EBh	SI/O5 Bit Rate Generator	S5BRG	176
01ECh	UART0 Special Mode Register 4	U0SMR4	137
01EDh	UART0 Special Mode Register 3	U0SMR3	136
01EEh	UART0 Special Mode Register 2	U0SMR2	136
01EFh	UART0 Special Mode Register	U0SMR	135
01F0h	UART1 Special Mode Register 4	U1SMR4	137
01F1h	UART1 Special Mode Register 3	U1SMR3	137
01F2h	UART1 Special Mode Register 2	U1SMR2	136
01F3h	UART1 Special Mode Register	U1SMR	135
01F4h	UART2 Special Mode Register 4	U2SMR4	137
01F5h	UART2 Special Mode Register 3	U2SMR3	136
01F6h	UART2 Special Mode Register 2	U2SMR2	136
01F7h	UART2 Special Mode Register	U2SMR	135
01F8h	UART2 Transmit/Receive Mode Register	U2MR	133
01F9h	UART2 Bit Rate Generator	U2BRG	132
01FAh			
01FBh	UART2 Transmit Buffer Register	U2TB	132
01FCh	UART2 Transmit/Receive Control Register 0	U2C0	133
01FDh	UART2 Transmit/Receive Control Register 1	U2C1	134
01FEh			
01FFh	UART2 Receive Buffer Register	U2RB	132

The blank areas are reserved.

Address	Register	Symbol	Page
0200h	CAN0 Message Control Register 0	C0MCTL0	207
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	208
0211h			
0212h	CAN0 Status Register	C0STR	209
0213h			
0214h	CAN0 Slot Status Register	C0SSTR	210
0215h			
0216h	CAN0 Interrupt Control Register	C0ICR	210
0217h			
0218h	CAN0 Extended ID Register	C0IDR	210
0219h			
021Ah	CAN0 Configuration Register	C0CONR	211
021Bh			
021Ch	CAN0 Receive Error Count Register	C0RECR	212
021Dh	CAN0 Transmit Error Count Register	C0TECR	212
021Eh	CAN0 Time Stamp Register	C0TSR	212
021Fh			
0220h	CAN1 Message Control Register 0	C1MCTL0	207
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	208
0231h			
0232h	CAN1 Status Register	C1STR	209
0233h			
0234h	CAN1 Slot Status Register	C1SSTR	210
0235h			
0236h	CAN1 Interrupt Control Register	C1ICR	210
0237h			
0238h	CAN1 Extended ID Register	C1IDR	210
0239h			
023Ah	CAN1 Configuration Register	C1CONR	211
023Bh			
023Ch	CAN1 Receive Error Count Register	C1RECR	212
023Dh	CAN1 Transmit Error Count Register	C1TECR	212
023Eh	CAN1 Time Stamp Register	C1TSR	212
023Fh			

Address	Register	Symbol	Page
0240h			
0241h			
0242h	CAN0 Acceptance Filter Support Register	C0AFS	212
0243h			
0244h	CAN1 Acceptance Filter Support Register	C1AFS	212
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	40
025Fh	CAN0/1 Clock Select Register	CCLKR	41
0260h	CAN1 Message Box 0: Identifier / DLC		
0261h			
0262h			
0263h			
0264h			
0265h			
0266h	CAN1 Message Box 0: Data Field		
0267h			
0268h			
0269h			
026Ah			
026Bh			
026Ch			
026Dh			
026Eh	CAN1 Message Box 0:Time Stamp		204
026Fh			
0270h	CAN1 Message Box 1: Identifier / DLC		205
0271h			
0272h			
0273h			
0274h			
0275h			
0276h	CAN1 Message Box 1: Data Field		
0277h			
0278h			
0279h			
027Ah			
027Bh			
027Ch			
027Dh			
027Eh	CAN1 Message Box 1:Time Stamp		
027Fh			

The blank areas are reserved.

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

204
205

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

204
205

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

204
205

Address	Register	Symbol	Page
0340h	CAN1 Message Box 14: Identifier / DLC		
0341h			
0342h			
0343h			
0344h			
0345h			
0346h	CAN1 Message Box 14: Data Field		
0347h			
0348h			
0349h			
034Ah			
034Bh			
034Ch			
034Dh			
034Eh	CAN1 Message Box 14: Time Stamp		
034Fh			
0350h	CAN1 Message Box 15: Identifier / DLC		
0351h			
0352h			
0353h			
0354h			
0355h			
0356h	CAN1 Message Box 15: Data Field		
0357h			
0358h			
0359h			
035Ah			
035Bh			
035Ch			
035Dh			
035Eh	CAN1 Message Box 15: Time Stamp		
035Fh			
0360h	CAN1 Global Mask Register	C1GMR	206
0361h			
0362h			
0363h			
0364h			
0365h			
0366h	CAN1 Local Mask A Register	C1LMAR	206
0367h			
0368h			
0369h			
036Ah			
036Bh			
036Ch	CAN1 Local Mask B Register	C1LMBR	206
036Dh			
036Eh			
036Fh			
0370h			
0371h			
0372h			
0373h			
0374h			
0375h			
0376h			
0377h			
0378h			
0379h			
037Ah			
037Bh			
037Ch			
037Dh			
037Eh			
037Fh			

204
205

The blank areas are reserved.

Address	Register	Symbol	Page
0380h	Count Start Flag	TABSR	96,111,124
0381h	Clock Prescaler Reset Flag	CPSRF	97,111
0382h	One-Shot Start Flag	ONSF	97
0383h	Trigger Select Register	TRGSR	97,124
0384h	Up/Down Flag	UDF	96
0385h			
0386h	Timer A0 Register	TA0	95
0387h			
0388h	Timer A1 Register	TA1	95
0389h			122
038Ah	Timer A2 Register	TA2	95
038Bh			122
038Ch	Timer A3 Register	TA3	95
038Dh			
038Eh	Timer A4 Register	TA4	95
038Fh			122
0390h	Timer B0 Register	TB0	110
0391h			
0392h	Timer B1 Register	TB1	110
0393h			
0394h	Timer B2 Register	TB2	110
0395h			122
0396h	Timer A0 Mode Register	TA0MR	95
0397h	Timer A1 Mode Register	TA1MR	98
0398h	Timer A2 Mode Register	TA2MR	100
0399h	Timer A3 Mode Register	TA3MR	105
039Ah	Timer A4 Mode Register	TA4MR	107
039Bh	Timer B0 Mode Register	TB0MR	110,112
039Ch	Timer B1 Mode Register	TB1MR	113,115
039Dh	Timer B2 Mode Register	TB2MR	125
039Eh	Timer B2 Special Mode Register	TB2SC	123
039Fh			
03A0h	UART0 Transmit/Receive Mode Register	U0MR	133
03A1h	UART0 Bit Rate Generator	U0BRG	132
03A2h	UART0 Transmit Buffer Register	U0TB	132
03A3h			
03A4h	UART0 Transmit/Receive Control Register 0	U0C0	133
03A5h	UART0 Transmit/Receive Control Register 1	U0C1	134
03A6h	UART0 Receive Buffer Register	U0RB	132
03A7h			
03A8h	UART1 Transmit/Receive Mode Register	U1MR	133
03A9h	UART1 Bit Rate Generator	U1BRG	132
03AAh	UART1 Transmit Buffer Register	U1TB	132
03ABh			
03ACh	UART1 Transmit/Receive Control Register 0	U1C0	133
03ADh	UART1 Transmit/Receive Control Register 1	U1C1	134
03AEh	UART1 Receive Buffer Register	U1RB	132
03AFh			
03B0h	UART Transmit/Receive Control Register 2	UCON	135
03B1h			
03B2h			
03B3h			
03B4h			
03B5h			
03B6h			
03B7h			
03B8h	DMA0 Request Cause Select Register	DM0SL	84
03B9h			
03BAh	DMA1 Request Cause Select Register	DM1SL	85
03BBh			
03BCh	CRC Data Register	CRCD	200
03BDh			
03BEh	CRC Input Register	CRCIN	200
03BFh			

The blank areas are reserved.

Address	Register	Symbol	Page
03C0h	A/D Register 0	AD0	184
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	184
03D5h			
03D6h	A/D Control Register 0	ADCON0	183,186,188
03D7h	A/D Control Register 1	ADCON1	190,192,194
03D8h	D/A Register 0	DA0	199
03D9h			
03DAh	D/A Register 1	DA1	199
03DBh			
03DCh	D/A Control Register	DACON	199
03DDh			
03DEh	Port P14 Control Register	PC14	234
03DFh	Pull-Up Control Register 3	PUR3	236
03E0h	Port P0 Register	P0	234
03E1h	Port P1 Register	P1	234
03E2h	Port P0 Direction Register	PD0	233
03E3h	Port P1 Direction Register	PD1	233
03E4h	Port P2 Register	P2	234
03E5h	Port P3 Register	P3	234
03E6h	Port P2 Direction Register	PD2	233
03E7h	Port P3 Direction Register	PD3	233
03E8h	Port P4 Register	P4	234
03E9h	Port P5 Register	P5	234
03EAh	Port P4 Direction Register	PD4	233
03EBh	Port P5 Direction Register	PD5	233
03ECh	Port P6 Register	P6	234
03EDh	Port P7 Register	P7	234
03EEh	Port P6 Direction Register	PD6	233
03EFh	Port P7 Direction Register	PD7	233
03F0h	Port P8 Register	P8	234
03F1h	Port P9 Register	P9	234
03F2h	Port P8 Direction Register	PD8	233
03F3h	Port P9 Direction Register	PD9	233
03F4h	Port P10 Register	P10	234
03F5h	Port P11 Register	P11	234
03F6h	Port P10 Direction Register	PD10	233
03F7h	Port P11 Direction Register	PD11	233
03F8h	Port P12 Register	P12	234
03F9h	Port P13 Register	P13	234
03FAh	Port P12 Direction Register	PD12	233
03FBh	Port P13 Direction Register	PD13	233
03FCh	Pull-up Control Register 0	PUR0	235
03FDh	Pull-up Control Register 1	PUR1	235
03FEh	Pull-up Control Register 2	PUR2	235
03FFh	Port Control Register	PCR	236

M16C/6N Group (M16C/6NK, M16C/6NM)

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

Rev.1.10
Jul 01, 2005

1. Overview

The M16C/6N Group (M16C/6NK, M16C/6NM) 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 and 128-pin plastic molded 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/6NK, M16C/6NM), the microcomputer is suited to car audio 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 and communication equipment which requires high-speed arithmetic/logic operations.

1.1 Applications

Car audio and industrial control systems, other

1.2 Performance Outline

Tables 1.1 and 1.2 list a performance outline of M16C/6N Group (M16C/6NK, M16C/6NM).

Table 1.1 Performance Outline of M16C/6N Group (100-pin Version: M16C/6NK)

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 mode		
	Address Space	1 Mbyte		
	Memory Capacity	See Table 1.3 Product List		
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 I/O	3 channels Clock synchronous, UART, I ² C-bus ⁽¹⁾ , IEBus ⁽²⁾ 2 channels 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: 32 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	21mA (f(BCLK) = 24MHz, PLL operation, no division)	-
		Flash Memory	23mA (f(BCLK) = 24MHz, PLL operation, no division)	21mA (f(BCLK) = 20MHz, PLL operation, no division)
		Mask ROM Flash Memory	3μA (f(BCLK) = 32kHz, Wait mode, Oscillation capacity Low) 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 LQFP			

NOTES:

1. I²C-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.

Table 1.2 Performance Outline of M16C/6N Group (128-pin Version: M16C/6NM)

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 mode		
	Address Space	1 Mbyte		
	Memory Capacity	See Table 1.3 Product List		
Peripheral Function	Port	Input/Output: 113 pins, Input: 1 pin		
	Multifunction Timer	Timer A: 16 bits × 5 channels Timer B: 16 bits × 6 channels Three-phase motor control circuit		
	Serial I/O	3 channels Clock synchronous, UART, I ² C-bus ⁽¹⁾ , IEBus ⁽²⁾ 4 channels 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: 34 sources, External: 12 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	21mA (f(BCLK) = 24MHz, PLL operation, no division)	-
		Flash Memory	23mA (f(BCLK) = 24MHz, PLL operation, no division)	21mA (f(BCLK) = 20MHz, PLL operation, no division)
		Mask ROM Flash Memory	3μA (f(BCLK) = 32kHz, Wait mode, Oscillation capacity Low) 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	128-pin plastic mold LQFP			

NOTES:

1. I²C-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/6NK, M16C/6NM).

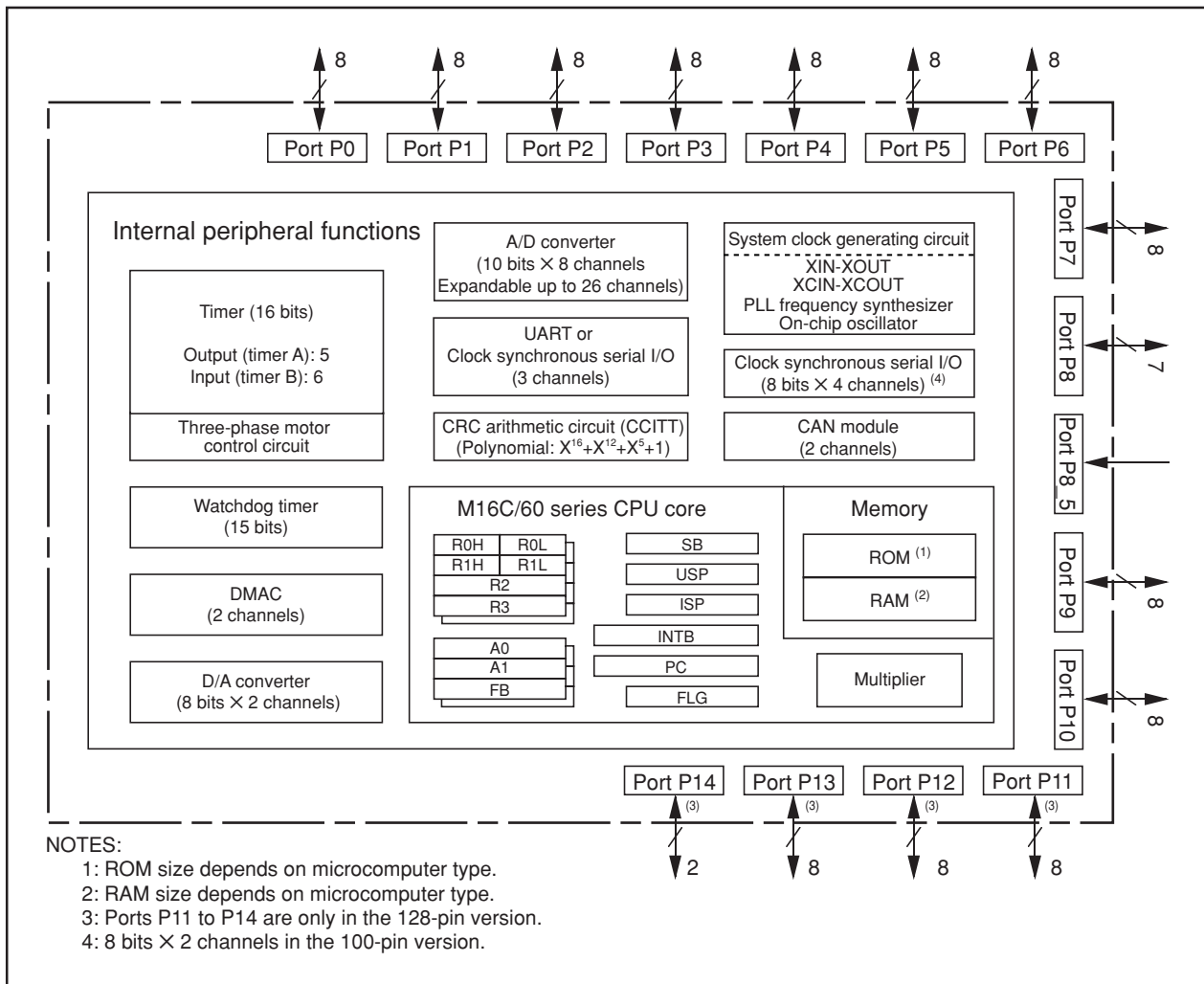


Figure 1.1 Block Diagram

1.4 Product List

Table 1.3 lists the M16C/6N Group (M16C/6NK, M16C/6NM) products and Figure 1.2 shows the type numbers, memory sizes and packages.

Table 1.3 Product List

As of Jul. 2005

Type No.	ROM Capacity	RAM Capacity	Package Type	Remarks	
M306NKFHGP	384 K + 4 Kbytes	31 Kbytes	PLQP0100KB-A	Flash memory version	Normal-ver.
M306NMFHGP			PLQP0128KB-A		
M306NKFJGP (D)	512 K + 4 Kbytes	31 Kbytes	PLQP0100KB-A		
M306NMFJGP			PLQP0128KB-A		
M306NKFHTGP (D)	384 K + 4 Kbytes	31 Kbytes	PLQP0100KB-A	T-ver.	
M306NMFHTGP (D)			PLQP0128KB-A		
M306NKFJTGP (D)	512 K + 4 Kbytes	31 Kbytes	PLQP0100KB-A		
M306NMFJTGP (D)			PLQP0128KB-A		
M306NKFHVGP (D)	384 K + 4 Kbytes	31 Kbytes	PLQP0100KB-A	V-ver.	
M306NMFHVGP (D)			PLQP0128KB-A		
M306NKFJVGP (D)	512 K + 4 Kbytes	31 Kbytes	PLQP0100KB-A		
M306NMFJVGP (D)			PLQP0128KB-A		
M306NKME-XXXGP	192 Kbytes	16 Kbytes	PLQP0100KB-A	Mask ROM version	Normal-ver.
M306NMME-XXXGP			PLQP0128KB-A		
M306NKMJ-XXXGP	256 Kbytes	20 Kbytes	PLQP0100KB-A		
M306NMMJ-XXXGP			PLQP0128KB-A		

(D): Under development

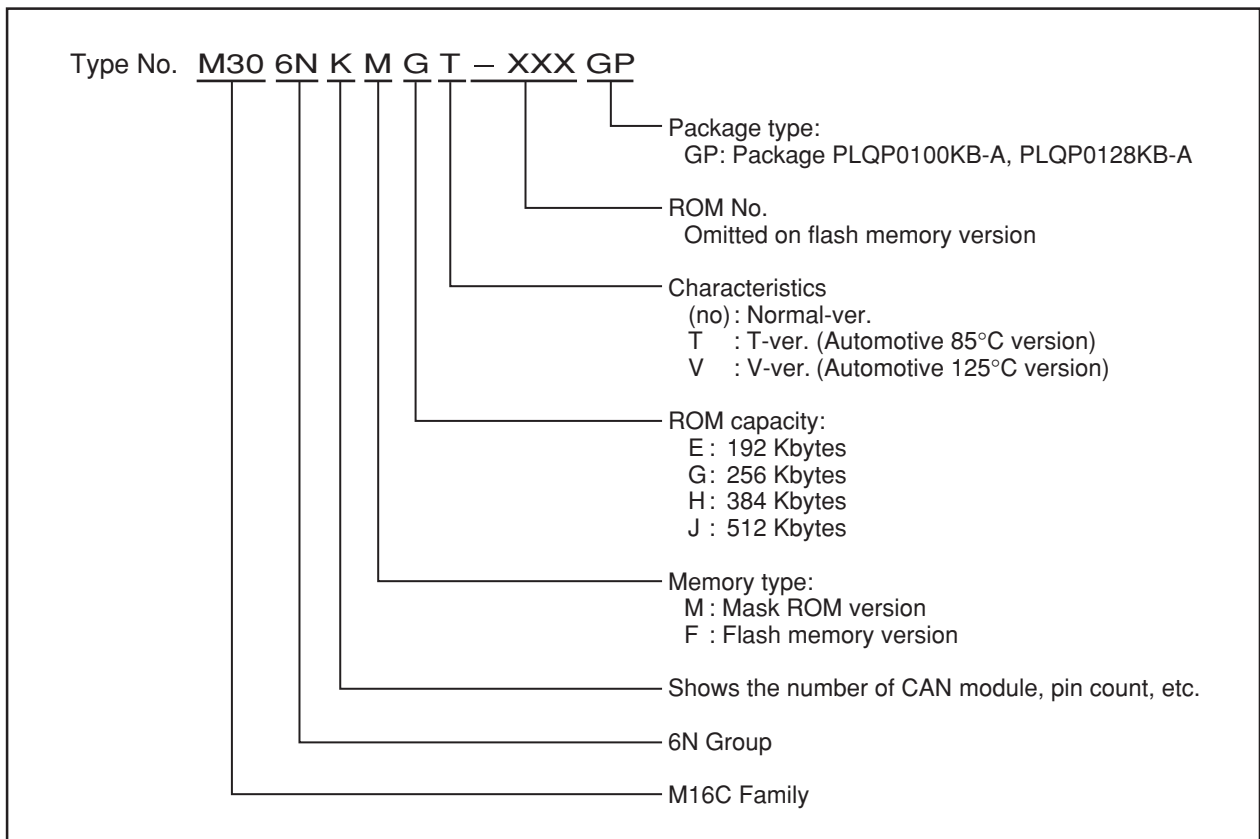


Figure 1.2 Type No., Memory Size, and Package

1.5 Pin Configuration

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

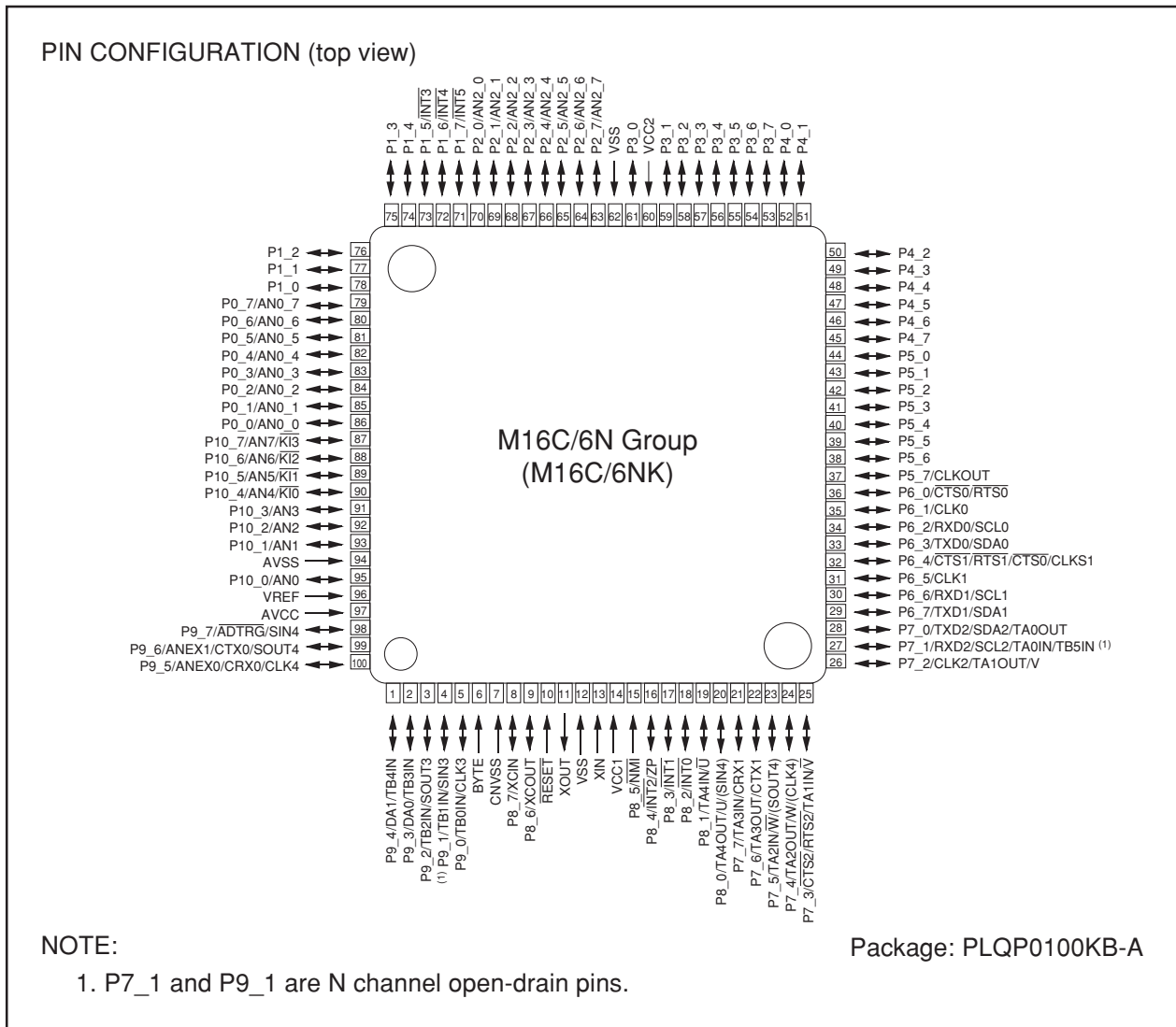


Figure 1.3 Pin Configuration (Top View) (1)

1.6 Pin Description

Tables 1.4 and 1.5 list the pin descriptions.

Table 1.4 Pin Description (100-pin and 128-pin Versions) (1)

Signal Name	Pin Name	I/O Type	Description
Power supply input	VCC1, VCC2, VSS	I	Apply 3.0 to 5.5V to the VCC1 and VCC2 pins and 0V to the VSS pin. The VCC apply condition is that VCC2 = VCC1 ⁽¹⁾ .
Analog power supply input	AVCC, AVSS	I	Applies the power supply for the A/D converter. Connect the AVCC pin to VCC1. Connect the AVSS pin to VSS.
Reset input	RESET	I	The microcomputer is in a reset state when applying "L" to the this pin.
CNVSS	CNVSS	I	Connect this pin to VSS.
External data bus width select input	BYTE	I	Connect this pin to VSS.
Main clock input	XIN	I	I/O pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT ⁽²⁾ .
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 ⁽²⁾ .
Sub clock output	XCOU	O	To use the external clock, input the clock from XCIN and leave XCOU open.
Clock output	CLKOUT	O	The clock of the same cycle as fC, f8, or f32 is output.
INT interrupt input	INT0 to INT8 ⁽³⁾	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, \bar{U} , V, \bar{V} , W, \bar{W}	O	These are Three-phase motor control output pins.
Serial I/O	CTS0 to CTS2	I	These are send control input pins.
	RTS0 to RTS2	O	These are receive control output pins.
	CLK0 to CLK6 ⁽³⁾	I/O	These are transfer clock I/O pins.
	RXD0 to RXD2	I	These are serial data input pins.
	SIN3 to SIN6 ⁽³⁾	I	These are serial data input pins.
	TXD0 to TXD2	O	These are serial data output pins.
	SOUT3 to SOUT6 ⁽³⁾	O	These are serial data output pins.
I ² C mode	CLKS1	O	This is output pin for transfer clock output from multiple pins function.
	SDA0 to SDA2	I/O	These are serial data I/O pins.
	SCL0 to SCL2	I/O	These are transfer clock I/O pins. (except SCL2 for the N-channel open drain output.)

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

NOTES:

1. In this manual, hereafter, VCC refers to VCC1 unless otherwise noted.
2. Ask the oscillator maker the oscillation characteristic.
3. INT6 to INT8, CLK5, CLK6, SIN5, SIN6, SOUT5, SOUT6 are only in the 128-pin version.

Table 1.5 Pin Description (100-pin and 128-pin Versions) (2)

Signal Name	Pin Name	I/O Type	Description
Reference voltage input	VREF	I	Applies the reference voltage for the A/D converter and D/A converter.
A/D converter	AN0 to AN7 AN0_0 to AN0_7 AN2_0 to AN2_7	I	Analog input pins for the A/D converter.
	ADTRG	I	This is an A/D trigger input pin.
	ANEX0	I/O	This is the extended analog input pin for the A/D converter, and is the output in external op-amp connection mode.
	ANEX1	I	This is the extended analog input pin for the A/D converter.
D/A converter	DA0, DA1	O	These are the output pins for the D/A converter.
CAN module	CRX0, CRX1	I	These are the input pins for the CAN module.
	CTX0, CTX1	O	These are the output pins for the CAN module.
I/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 P11_0 to P11_7 ⁽¹⁾ P12_0 to P12_7 ⁽¹⁾ P13_0 to P13_7 ⁽¹⁾ P14_0, P14_1 ⁽¹⁾	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. (except P7_1 and P9_1 for the N-channel open drain output.)
Input port	P8_5	I	Input pin for the $\overline{\text{NMI}}$ interrupt. Pin states can be read by the P8_5 bit in the P8 register.

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

NOTE:

- Ports P11 to P14 are only in the 128-pin version.

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.

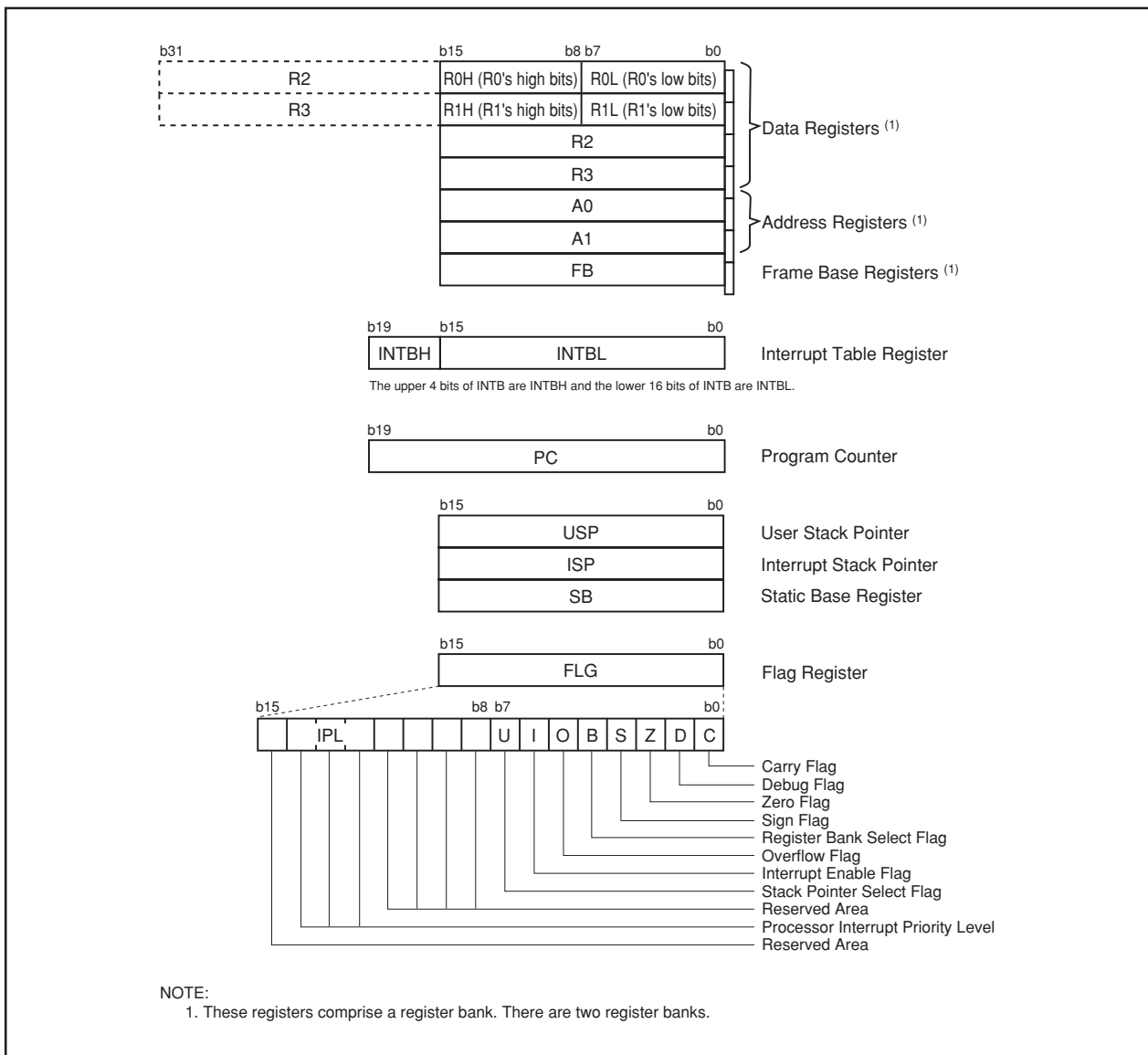


Figure 2.1 CPU Registers

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/6NK, M16C/6NM). 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 512-Kbyte internal ROM is allocated to the addresses from 80000h 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 31-Kbyte internal RAM is allocated to the addresses from 00400h to 07FFFh. 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 and M16C/20 Series Software Manual**.

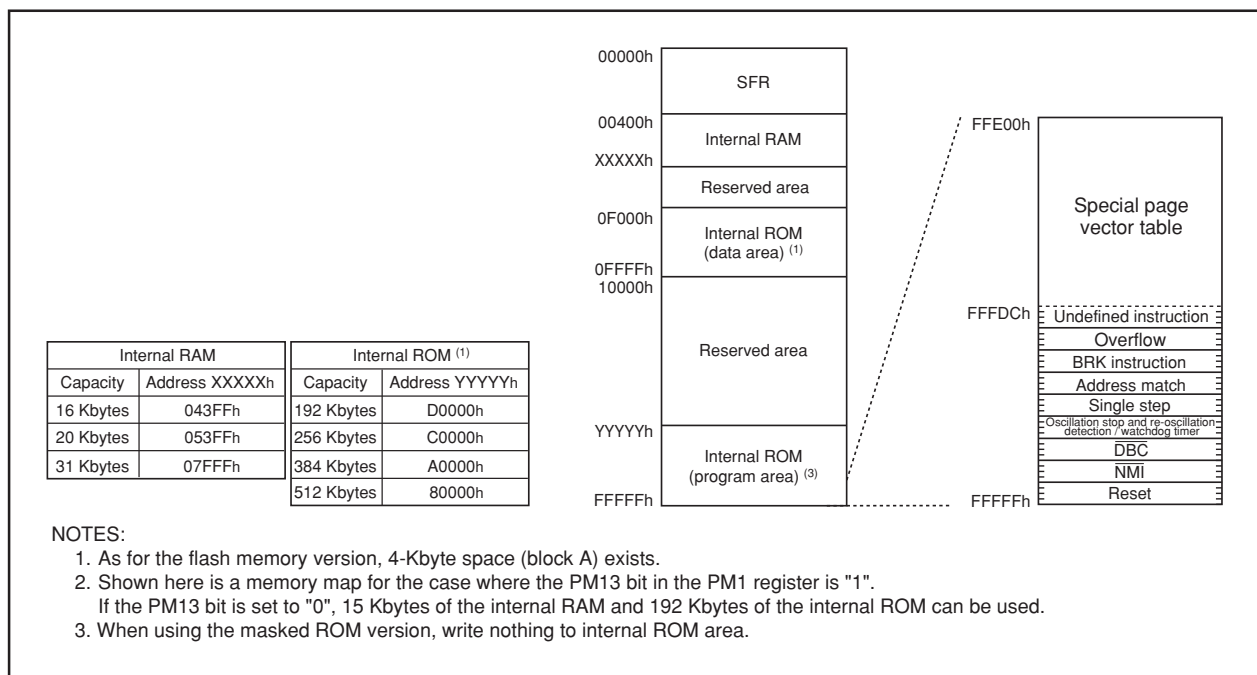


Figure 3.1 Memory Map

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	PM0	00h
0005h	Processor Mode Register 1	PM1	00001000b
0006h	System Clock Control Register 0	CM0	01001000b
0007h	System Clock Control Register 1	CM1	00100000b
0008h			
0009h	Address Match Interrupt Enable Register	AIER	XXXXXX00b
000Ah	Protect Register	PRCR	XX000000b
000Bh			
000Ch	Oscillation Stop Detection Register ⁽¹⁾	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			
001Ch	PLL Control Register 0	PLC0	0001X010b
001Dh			
001Eh	Processor Mode Register 2	PM2	XXX00000b
001Fh			
0020h			XXh
0021h	DMA0 Source Pointer	SAR0	XXh
0022h			XXh
0023h			
0024h			XXh
0025h	DMA0 Destination Pointer	DAR0	XXh
0026h			XXh
0027h			
0028h	DMA0 Transfer Counter	TCR0	XXh
0029h			XXh
002Ah			
002Bh			
002Ch	DMA0 Control Register	DM0CON	00000X00b
002Dh			
002Eh			
002Fh			
0030h			XXh
0031h	DMA1 Source Pointer	SAR1	XXh
0032h			XXh
0033h			
0034h			XXh
0035h	DMA1 Destination Pointer	DAR1	XXh
0036h			XXh
0037h			
0038h	DMA1 Transfer Counter	TCR1	XXh
0039h			XXh
003Ah			
003Bh			
003Ch	DMA1 Control Register	DM1CON	00000X00b
003Dh			
003Eh			
003Fh			

X: Undefined

NOTES:

1. The CM20, CM21, and CM27 bits in the CM2 register do not change at oscillation stop detection reset.
2. 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
	SI/O5 Interrupt Control Register ⁽¹⁾	S5IC	
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
	SI/O4 Interrupt Control Register	S4IC	
	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	KUPIIC	
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	XX00X000b
	INT7 Interrupt Control Register ⁽¹⁾	INT7IC	
0058h	Timer A3 Interrupt Control Register	TA3IC	XX00X000b
	INT6 Interrupt Control Register ⁽¹⁾	INT6IC	
0059h	Timer A4 Interrupt Control Register	TA4IC	XXXXX000b
005Ah	Timer B0 Interrupt Control Register	TB0IC	XXXXX000b
	SI/O6 Interrupt Control Register ⁽¹⁾	S6IC	
005Bh	Timer B1 Interrupt Control Register	TB1IC	XX00X000b
	INT8 Interrupt Control Register ⁽¹⁾	INT8IC	
005Ch	Timer B2 Interrupt Control Register	TB2IC	XXXXX000b
005Dh	INT0 Interrupt Control Register	INT0IC	XX00X000b
005Eh	INT1 Interrupt Control Register	INT1IC	XX00X000b
005Fh	INT2 Interrupt Control Register	INT2IC	XX00X000b
0060h			XXh
0061h			XXh
0062h	CAN0 Message Box 0: Identifier / DLC		XXh
0063h			XXh
0064h			XXh
0065h			XXh
0066h	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			XXh
0071h			XXh
0072h	CAN0 Message Box 1: Identifier / DLC		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

NOTES:

1. These registers exist only in the 128-pin version.
2. 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	CAN0 Message Box 2: Data Field		XXh		
0087h			XXh		
0088h			XXh		
0089h			XXh		
008Ah			XXh		
008Bh			XXh		
008Ch	CAN0 Message Box 2: Time Stamp		XXh		
008Dh			XXh		
008Eh			XXh		
008Fh			XXh		
0090h			CAN0 Message Box 3: Identifier / DLC		XXh
0091h					XXh
0092h	XXh				
0093h	XXh				
0094h	XXh				
0095h	XXh				
0096h	CAN0 Message Box 3: Data Field		XXh		
0097h			XXh		
0098h			XXh		
0099h			XXh		
009Ah			XXh		
009Bh			XXh		
009Ch	CAN0 Message Box 3: Time Stamp		XXh		
009Dh			XXh		
009Eh			XXh		
009Fh			XXh		
00A0h			CAN0 Message Box 4: Identifier / DLC		XXh
00A1h					XXh
00A2h	XXh				
00A3h	XXh				
00A4h	XXh				
00A5h	XXh				
00A6h	CAN0 Message Box 4: Data Field		XXh		
00A7h			XXh		
00A8h			XXh		
00A9h			XXh		
00AAh			XXh		
00ABh			XXh		
00ACh	CAN0 Message Box 4: Time Stamp		XXh		
00ADh			XXh		
00AEh			XXh		
00AFh			XXh		
00B0h			CAN0 Message Box 5: Identifier / DLC		XXh
00B1h					XXh
00B2h	XXh				
00B3h	XXh				
00B4h	XXh				
00B5h	XXh				
00B6h	CAN0 Message Box 5: Data Field		XXh		
00B7h			XXh		
00B8h			XXh		
00B9h			XXh		
00BAh			XXh		
00BBh			XXh		
00BCh	CAN0 Message Box 5: Time Stamp		XXh		
00BDh			XXh		
00BEh			XXh		
00BFh			XXh		

X: Undefined

Table 4.4 SFR Information (4)

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

X: Undefined

Table 4.5 SFR Information (5)

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

X: Undefined

Table 4.6 SFR Information (6)

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

X: Undefined

NOTE:

- 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 ⁽¹⁾	FMR1	0X00XX0Xb
01B6h			
01B7h	Flash Memory Control Register 0 ⁽¹⁾	FMR0	00000001b
01B8h			00h
01B9h	Address Match Interrupt Register 2	RMAD2	00h
01BAh			X0h
01BBh	Address Match Interrupt Enable Register 2	AIER2	XXXXXX00b
01BCh			00h
01BDh	Address Match Interrupt Register 3	RMAD3	00h
01BEh			X0h
01BFh			

X: Undefined

NOTES:

1. These registers are included in the flash memory version. Cannot be accessed by users in the mask ROM version.
2. 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			
01C2h			XXh
01C3h	Timer A1-1 Register	TA11	XXh
01C4h			XXh
01C5h	Timer A2-1 Register	TA21	XXh
01C6h			XXh
01C7h	Timer A4-1 Register	TA41	XXh
01C8h	Three-Phase PWM Control Register 0	INVC0	00h
01C9h	Three-Phase PWM Control Register 1	INVC1	00h
01CAh	Three-Phase Output Buffer Register 0	IDB0	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	Interrupt Cause Select Register 2	IFSR2	X0000000b
01D0h			XXh
01D1h	Timer B3 Register	TB3	XXh
01D2h			XXh
01D3h	Timer B4 Register	TB4	XXh
01D4h			XXh
01D5h	Timer B5 Register	TB5	XXh
01D6h	SI/O6 Transmit/Receive Register ⁽¹⁾	S6TRR	XXh
01D7h			
01D8h	SI/O6 Control Register ⁽¹⁾	S6C	01000000b
01D9h	SI/O6 Bit Rate Generator ⁽¹⁾	S6BRG	XXh
01DAh	SI/O3, 4, 5, 6 Transmit/Receive Register ⁽²⁾	S3456TRR	XXXX0000b
01DBh	Timer B3 Mode Register	TB3MR	00X0000b
01DCh	Timer B4 Mode Register	TB4MR	00X0000b
01DDh	Timer B5 Mode Register	TB5MR	00X0000b
01DEh	Interrupt Cause Select Register 0	IFSR0	00h
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	SI/O4 Transmit/Receive Register	S4TRR	XXh
01E5h			
01E6h	SI/O4 Control Register	S4C	01000000b
01E7h	SI/O4 Bit Rate Generator	S4BRG	XXh
01E8h	SI/O5 Transmit/Receive Register ⁽¹⁾	S5TRR	XXh
01E9h			
01EAh	SI/O5 Control Register ⁽¹⁾	S5C	01000000b
01EBh	SI/O5 Bit Rate Generator ⁽¹⁾	S5BRG	XXh
01ECh	UART0 Special Mode Register 4	U0SMR4	00h
01EDh	UART0 Special Mode Register 3	U0SMR3	00X0X0Xb
01EEh	UART0 Special Mode Register 2	U0SMR2	X0000000b
01EFh	UART0 Special Mode Register	U0SMR	X0000000b
01F0h	UART1 Special Mode Register 4	U1SMR4	00h
01F1h	UART1 Special Mode Register 3	U1SMR3	00X0X0Xb
01F2h	UART1 Special Mode Register 2	U1SMR2	X0000000b
01F3h	UART1 Special Mode Register	U1SMR	X0000000b
01F4h	UART2 Special Mode Register 4	U2SMR4	00h
01F5h	UART2 Special Mode Register 3	U2SMR3	00X0X0Xb
01F6h	UART2 Special Mode Register 2	U2SMR2	X0000000b
01F7h	UART2 Special Mode Register	U2SMR	X0000000b
01F8h	UART2 Transmit/Receive Mode Register	U2MR	00h
01F9h	UART2 Bit Rate Generator	U2BRG	XXh
01FAh			XXh
01FBh	UART2 Transmit Buffer Register	U2TB	XXh
01FCh	UART2 Transmit/Receive Control Register 0	U2C0	00001000b
01FDh	UART2 Transmit/Receive Control Register 1	U2C1	0000010b
01FEh			XXh
01FFh	UART2 Receive Buffer Register	U2RB	XXh

X: Undefined

NOTES:

1. These registers exist only in the 128-pin version.
2. The S5TRF and S6TRF bits in the S3456TRR register are used in the 128-pin version.
3. 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			XXh
0261h	CAN1 Message Box 0: Identifier / DLC		XXh
0262h			XXh
0263h			XXh
0264h			XXh
0265h			XXh
0266h	CAN1 Message Box 0: Data Field		XXh
0267h			XXh
0268h			XXh
0269h			XXh
026Ah			XXh
026Bh	CAN1 Message Box 0:Time Stamp		XXh
026Ch			XXh
026Dh			XXh
026Eh			XXh
026Fh			XXh
0270h	CAN1 Message Box 1: Identifier / DLC		XXh
0271h			XXh
0272h			XXh
0273h			XXh
0274h			XXh
0275h	CAN1 Message Box 1: Data Field		XXh
0276h			XXh
0277h			XXh
0278h			XXh
0279h			XXh
027Ah	CAN1 Message Box 1:Time Stamp		XXh
027Bh			XXh
027Ch			XXh
027Dh			XXh
027Eh			XXh
027Fh			XXh

X: Undefined

NOTE:

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

X: Undefined

Table 4.12 SFR Information (12)

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

X: Undefined

Table 4.13 SFR Information (13)

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

X: Undefined

Table 4.14 SFR Information (14)

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

X: Undefined

NOTE:

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 ⁽¹⁾
0385h			
0386h	Timer A0 Register	TA0	XXh
0387h			XXh
0388h	Timer A1 Register	TA1	XXh
0389h			XXh
038Ah	Timer A2 Register	TA2	XXh
038Bh			XXh
038Ch	Timer A3 Register	TA3	XXh
038Dh			XXh
038Eh	Timer A4 Register	TA4	XXh
038Fh			XXh
0390h	Timer B0 Register	TB0	XXh
0391h			XXh
0392h	Timer B1 Register	TB1	XXh
0393h			XXh
0394h	Timer B2 Register	TB2	XXh
0395h			XXh
0396h	Timer A0 Mode Register	TA0MR	00h
0397h	Timer A1 Mode Register	TA1MR	00h
0398h	Timer A2 Mode Register	TA2MR	00h
0399h	Timer A3 Mode Register	TA3MR	00h
039Ah	Timer A4 Mode Register	TA4MR	00h
039Bh	Timer B0 Mode Register	TB0MR	00XX0000b
039Ch	Timer B1 Mode Register	TB1MR	00XX0000b
039Dh	Timer B2 Mode Register	TB2MR	00XX0000b
039Eh	Timer B2 Special Mode Register	TB2SC	XXXXXX00b
039Fh			
03A0h	UART0 Transmit/Receive Mode Register	U0MR	00h
03A1h	UART0 Bit Rate 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	A/D Register 1	AD1	XXh
03C3h			XXh
03C4h	A/D Register 2	AD2	XXh
03C5h			XXh
03C6h	A/D Register 3	AD3	XXh
03C7h			XXh
03C8h	A/D Register 4	AD4	XXh
03C9h			XXh
03CAh	A/D Register 5	AD5	XXh
03CBh			XXh
03CCh	A/D Register 6	AD6	XXh
03CDh			XXh
03CEh	A/D Register 7	AD7	XXh
03CFh			XXh
03D0h			
03D1h			
03D2h			
03D3h			
03D4h	A/D Control Register 2	ADCON2	00h
03D5h			
03D6h	A/D Control Register 0	ADCON0	00000XXXb
03D7h	A/D Control Register 1	ADCON1	00h
03D8h	D/A Register 0	DA0	00h
03D9h			
03DAh	D/A Register 1	DA1	00h
03DBh			
03DCh	D/A Control Register	DACON	00h
03DDh			
03DEh	Port P14 Control Register ⁽¹⁾	PC14	XX00XXXb
03DFh	Pull-Up Control Register 3 ⁽¹⁾	PUR3	00h
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 P11 Register ⁽¹⁾	P11	XXh
03F6h	Port P10 Direction Register	PD10	00h
03F7h	Port P11 Direction Register ⁽¹⁾	PD11	00h
03F8h	Port P12 Register ⁽¹⁾	P12	XXh
03F9h	Port P13 Register ⁽¹⁾	P13	XXh
03FAh	Port P12 Direction Register ⁽¹⁾	PD12	00h
03FBh	Port P13 Direction Register ⁽¹⁾	PD13	00h
03FCh	Pull-up Control Register 0	PUR0	00h
03FDh	Pull-up Control Register 1	PUR1	00h
03FEh	Pull-up Control Register 2	PUR2	00h
03FFh	Port Control Register	PCR	00h

X: Undefined

NOTES:

1. These registers exist only in the 128-pin version.
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”). Figure 5.3 shows CPU register states after reset. Refer to **4. SFR** for SFR states after reset.

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

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. SFR** for details.

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. SFR** for details.

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 **7.5 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. SFR** for details.

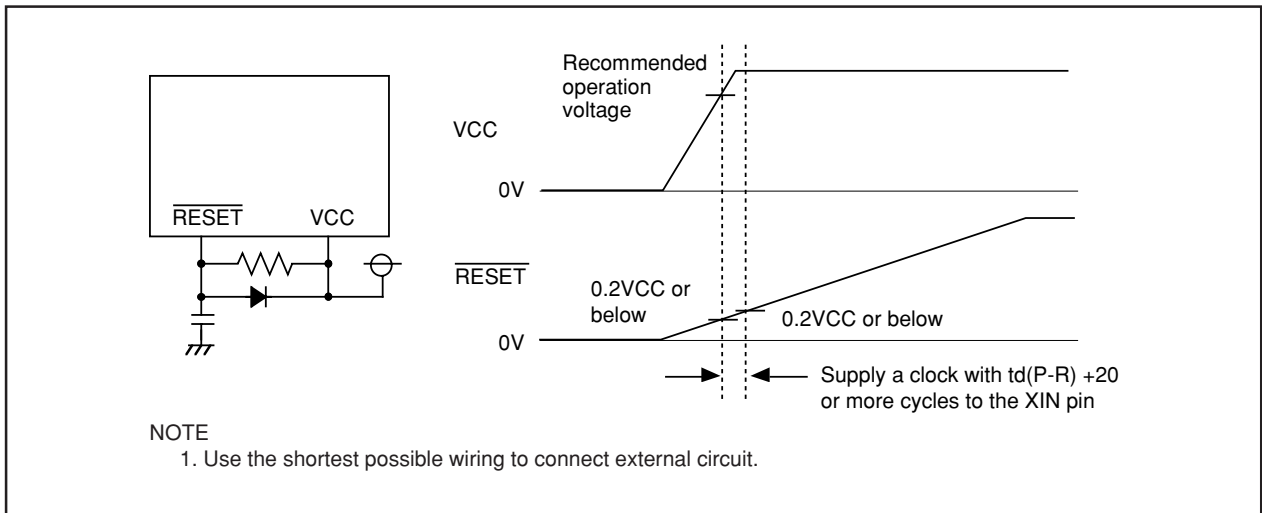


Figure 5.1 Example Reset Circuit

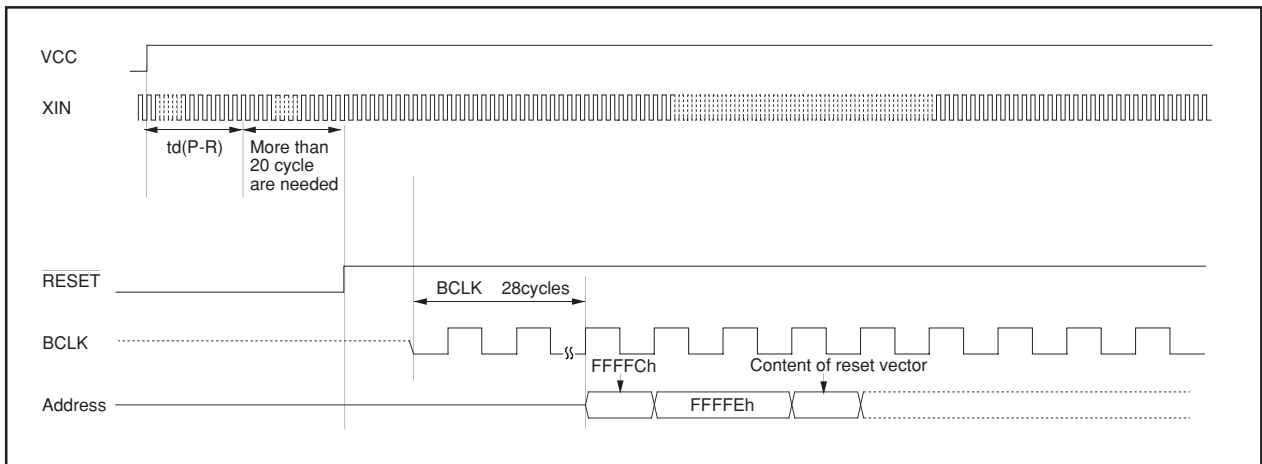


Figure 5.2 Reset Sequence

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

Pin Name	Status (CNVSS = VSS)
P0, P1, P2, P3, P4, P5, P6, P7, P8_0 to P8_4, P8_6, P8_7, P9, P10, P11, P12, P13, P14_0, P14_1 ⁽²⁾	Input port

NOTE:

1. P11, P12, P13, P14_0 and P14_1 pins are only in the 128-pin version.

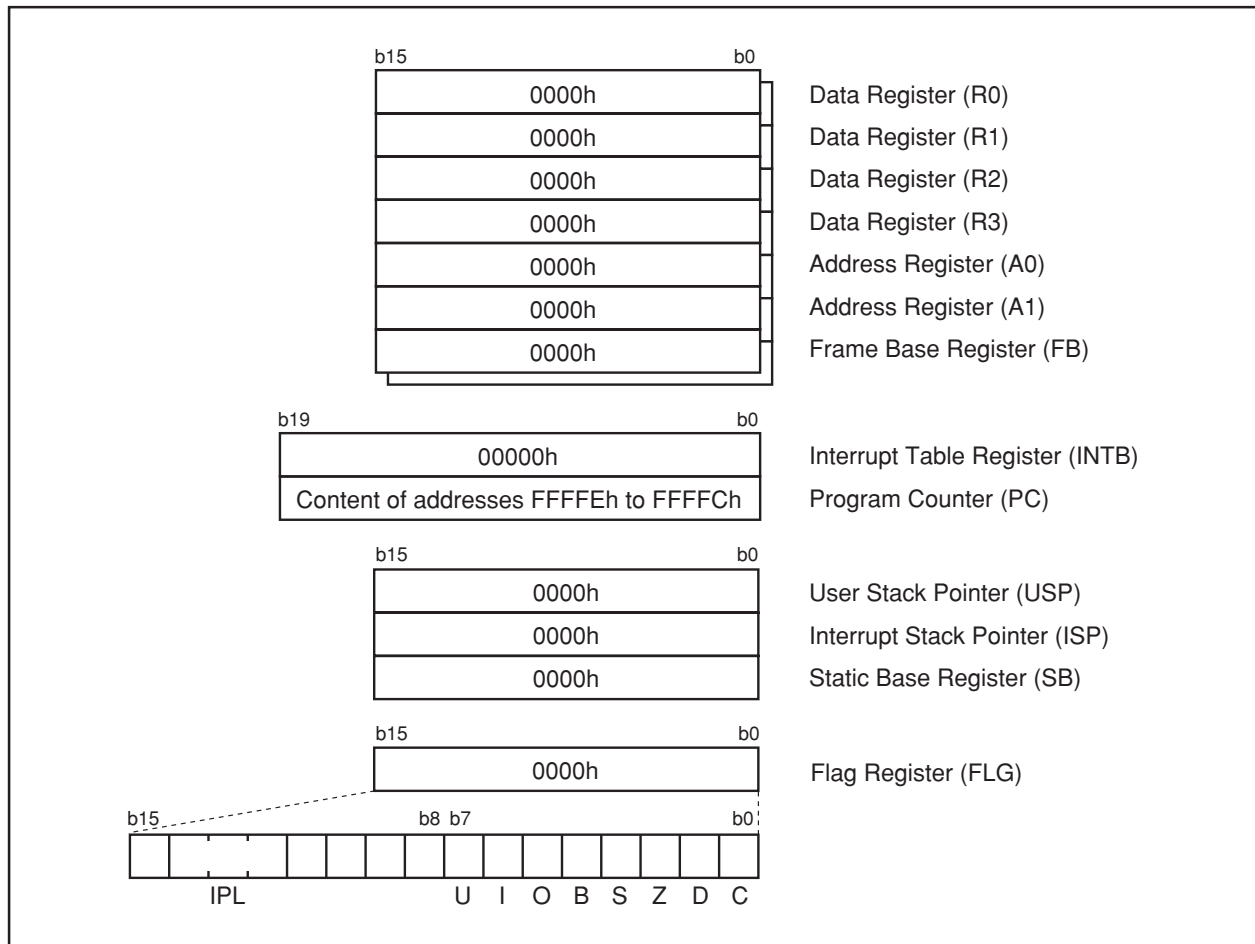


Figure 5.3 CPU Register Status After Reset

6. Processor Mode

Three processor mode is available single-chip mode only.

Figures 6.1 and 6.2 show the processor mode related registers. Figure 6.3 shows the memory map.

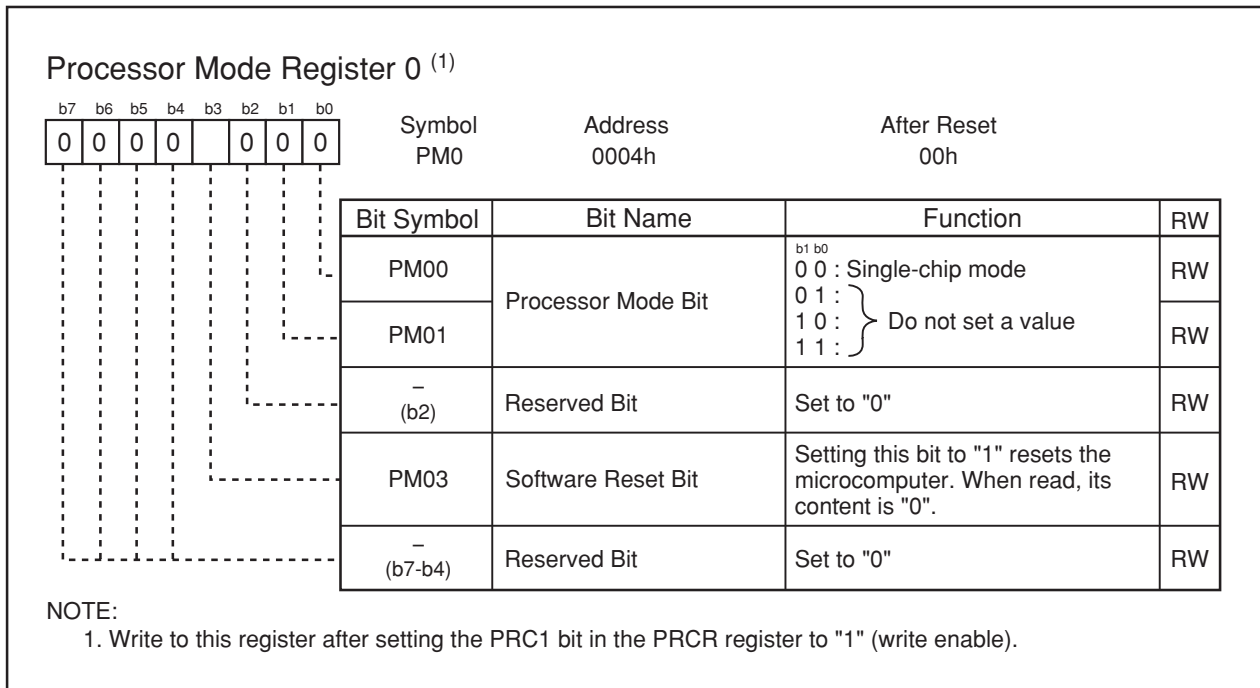


Figure 6.1 PM0 Register

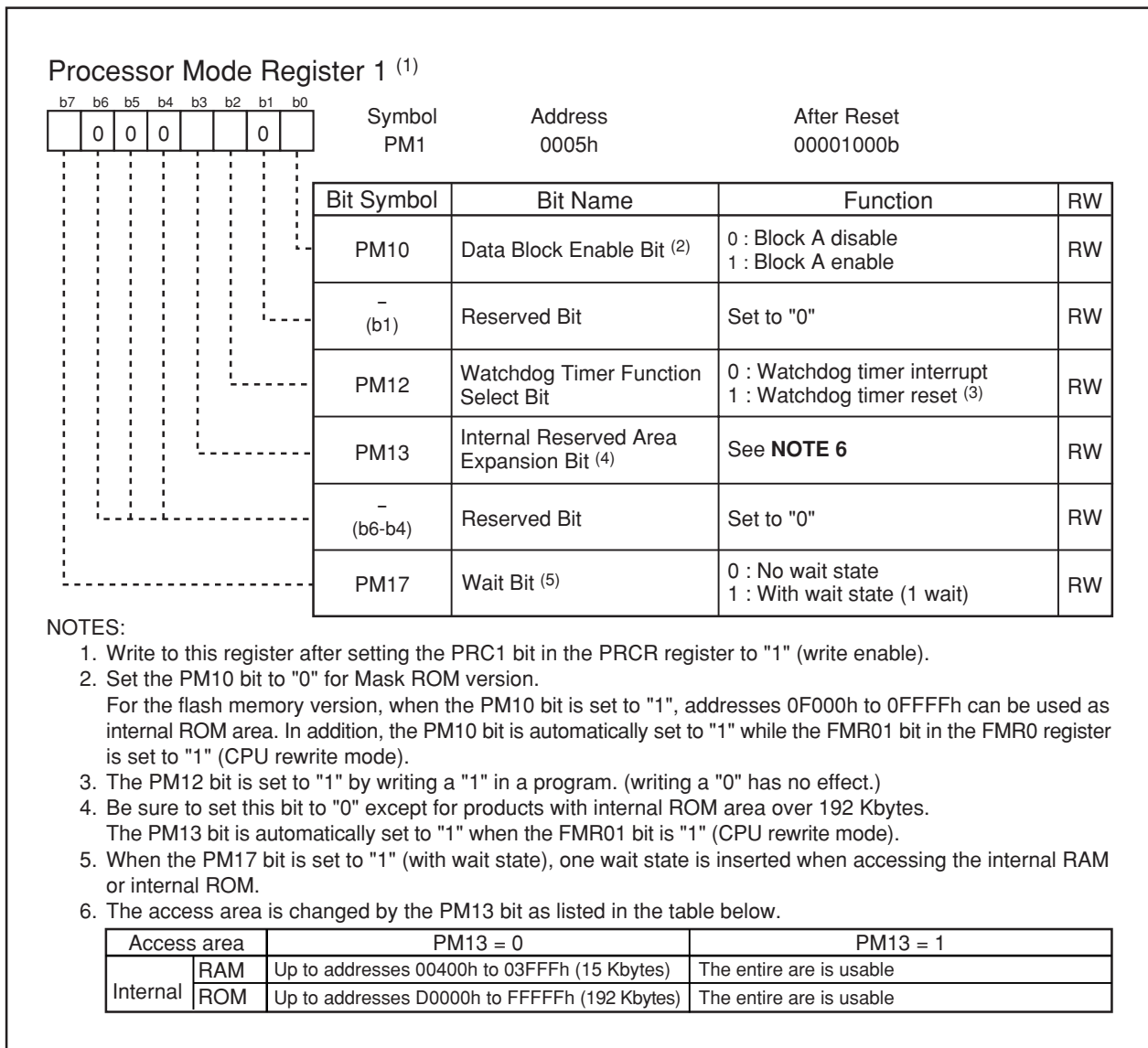


Figure 6.2 PM1 Register

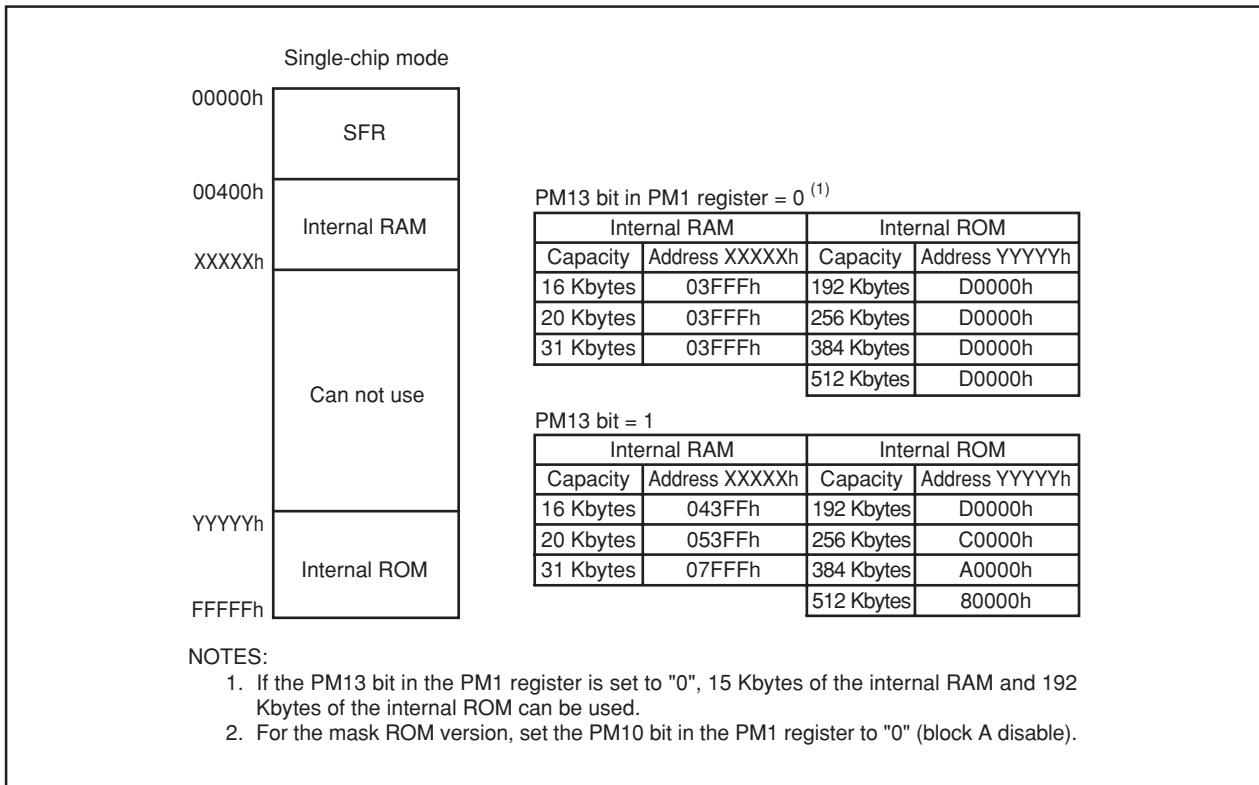


Figure 6.3 Memory Map

7. Clock Generating Circuit

7.1 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 7.1 lists the clock generating circuit specifications. Figure 7.1 shows the clock generating circuit. Figures 7.2 to 7.8 show the clock-related registers.

Table 7.1 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"> • CPU clock source • Peripheral function clock source 	<ul style="list-style-type: none"> • CPU clock source • Clock source of Timer A, B 	<ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when the main clock stops oscillating 	<ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source
Clock Frequency	0 to 16 MHz	32.768 kHz	About 1 MHz	16 MHz, 20 MHz, 24 MHz
Usable Oscillator	<ul style="list-style-type: none"> •Ceramic oscillator •Crystal oscillator 	•Crystal oscillator	-	-
Pins to Connect Oscillator	XIN, XOUT	XCIN, XCOU	-	-
Oscillation Stop and Re-Oscillation Detection Function	Available	Available	Available	Available
Oscillation Status After Reset	Oscillating	Stopped	Stopped	Stopped
Other	Externally derived clock can be input		-	-

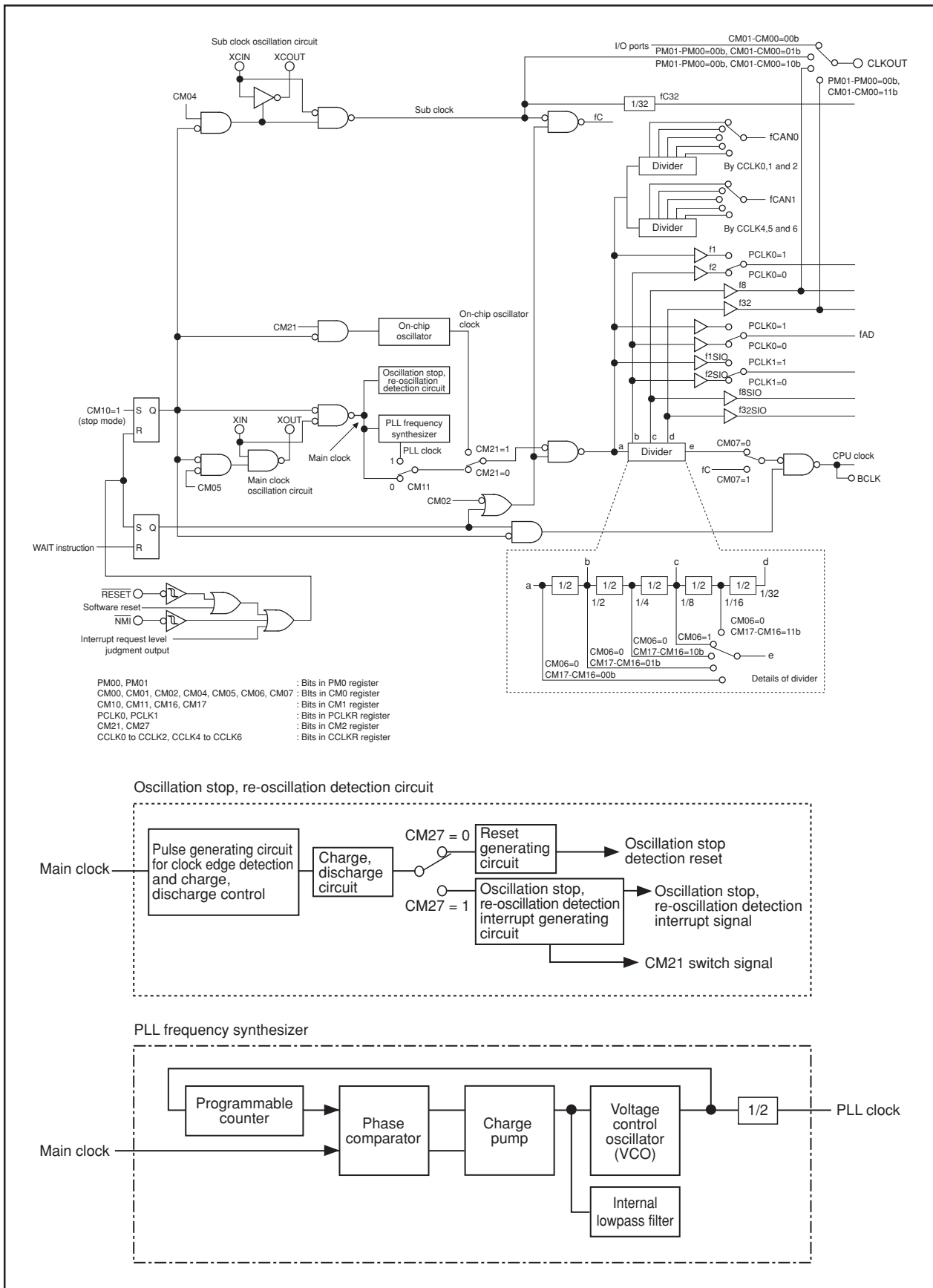


Figure 7.1 Clock Generating Circuit

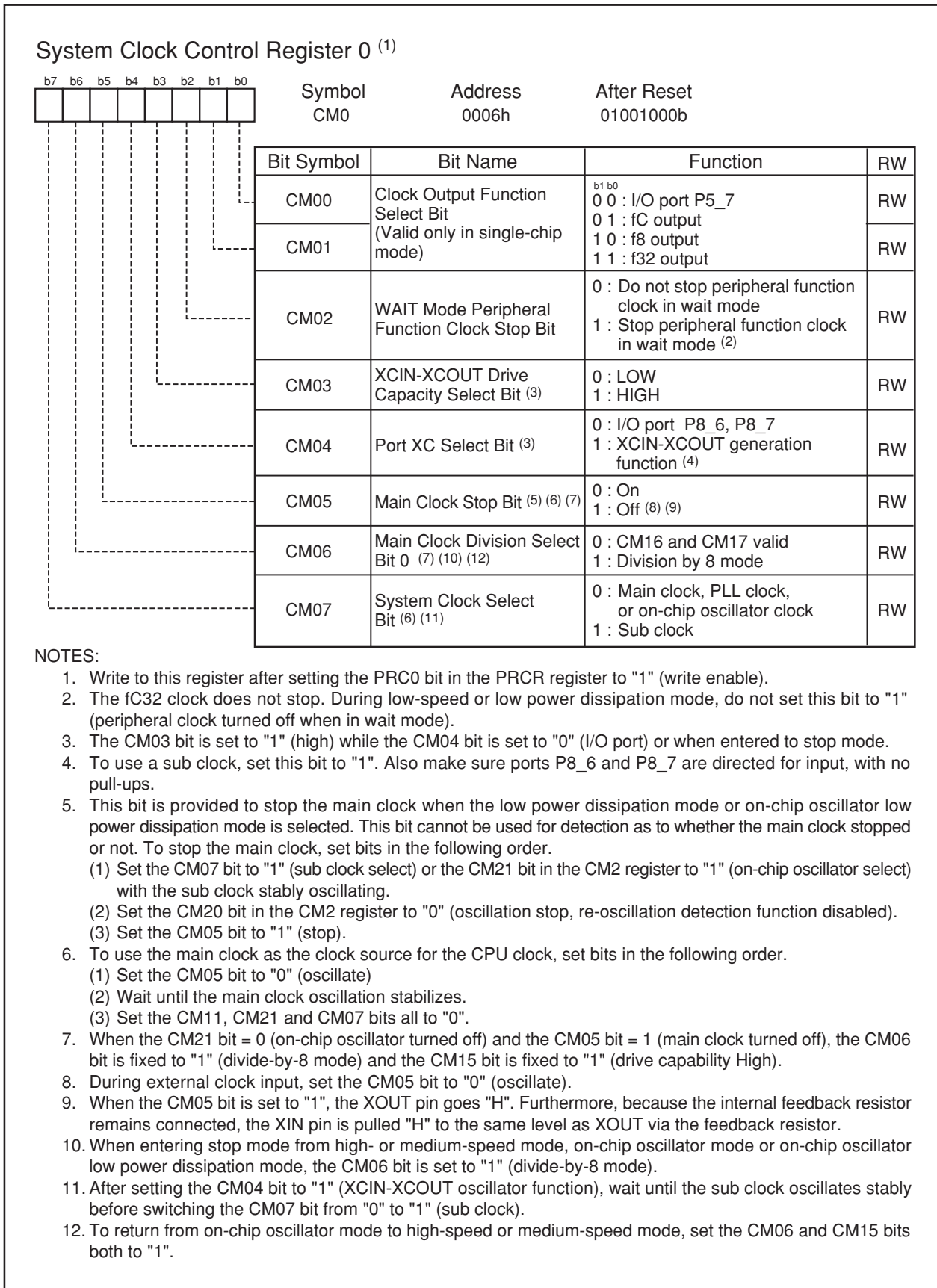


Figure 7.2 CM0 Register

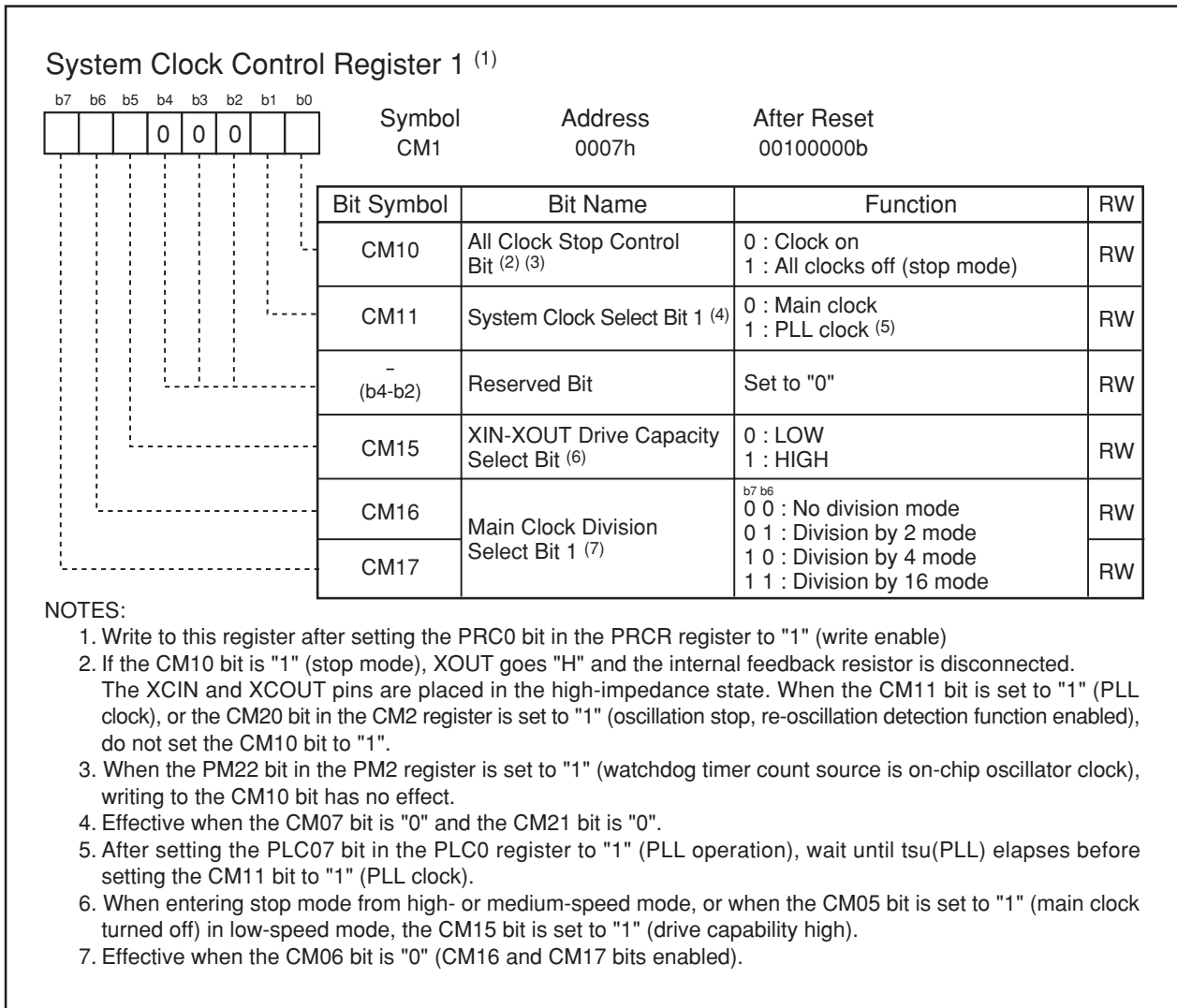


Figure 7.3 CM1 Register

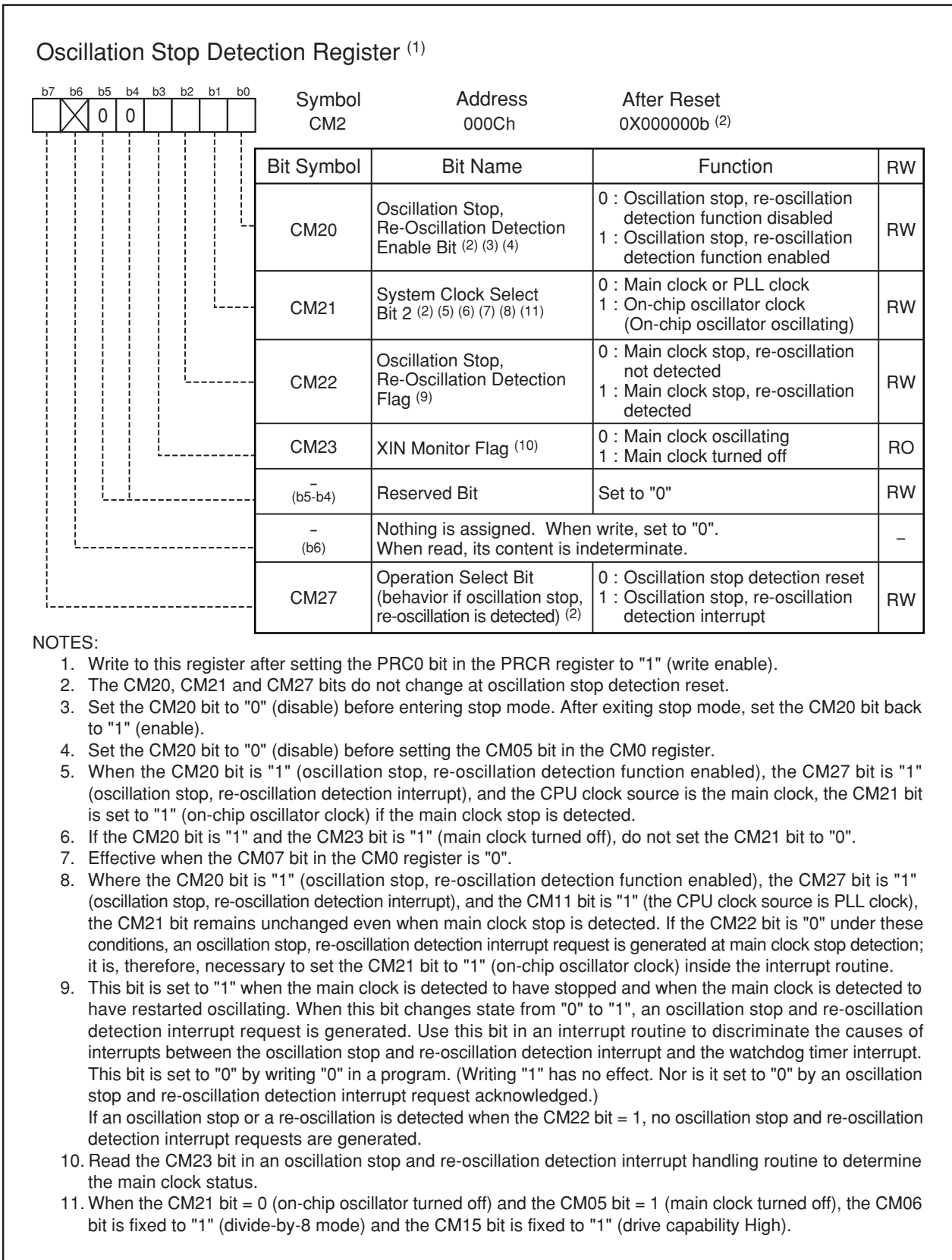


Figure 7.4 CM2 Register

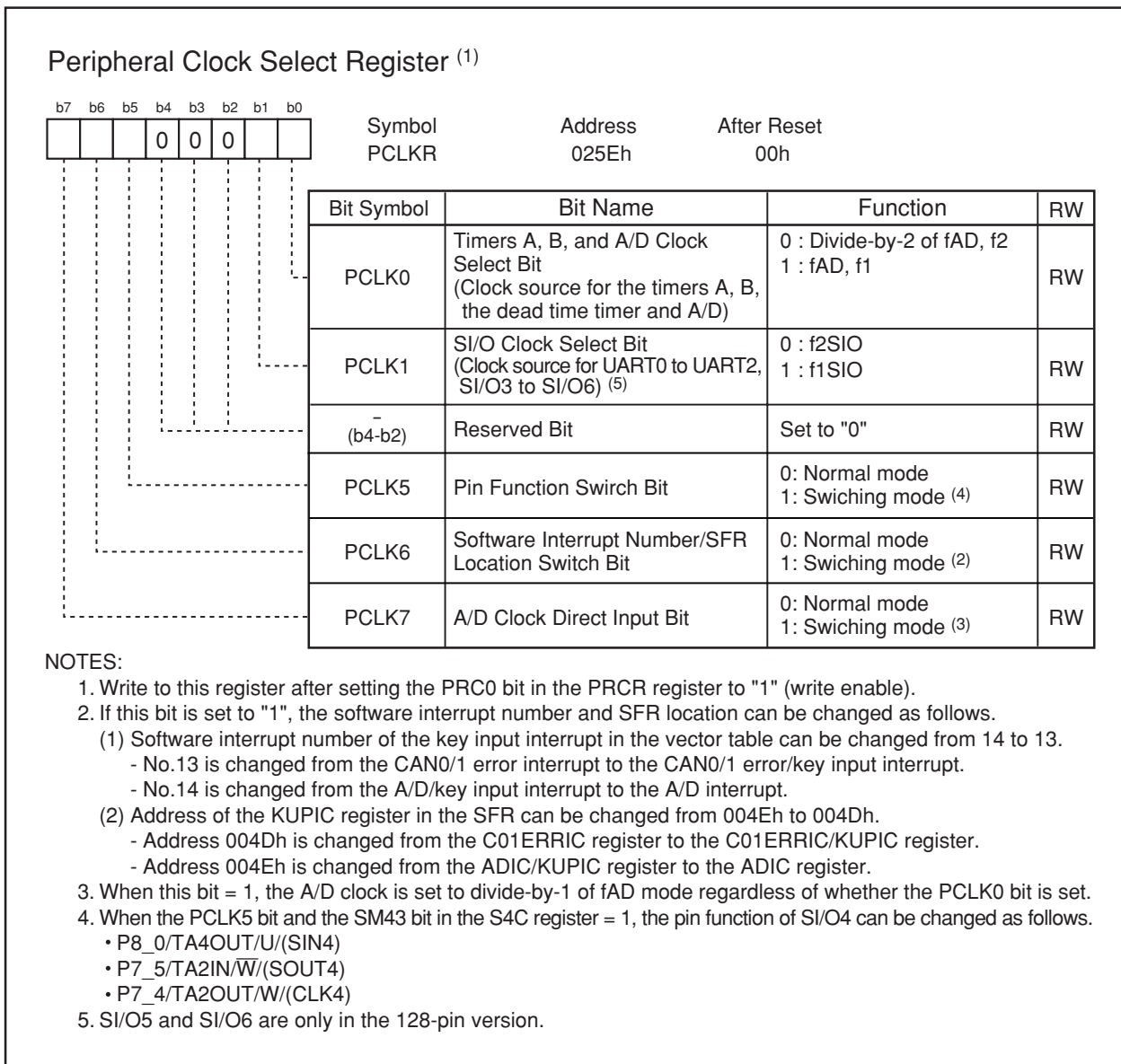


Figure 7.5 PCLKR Register

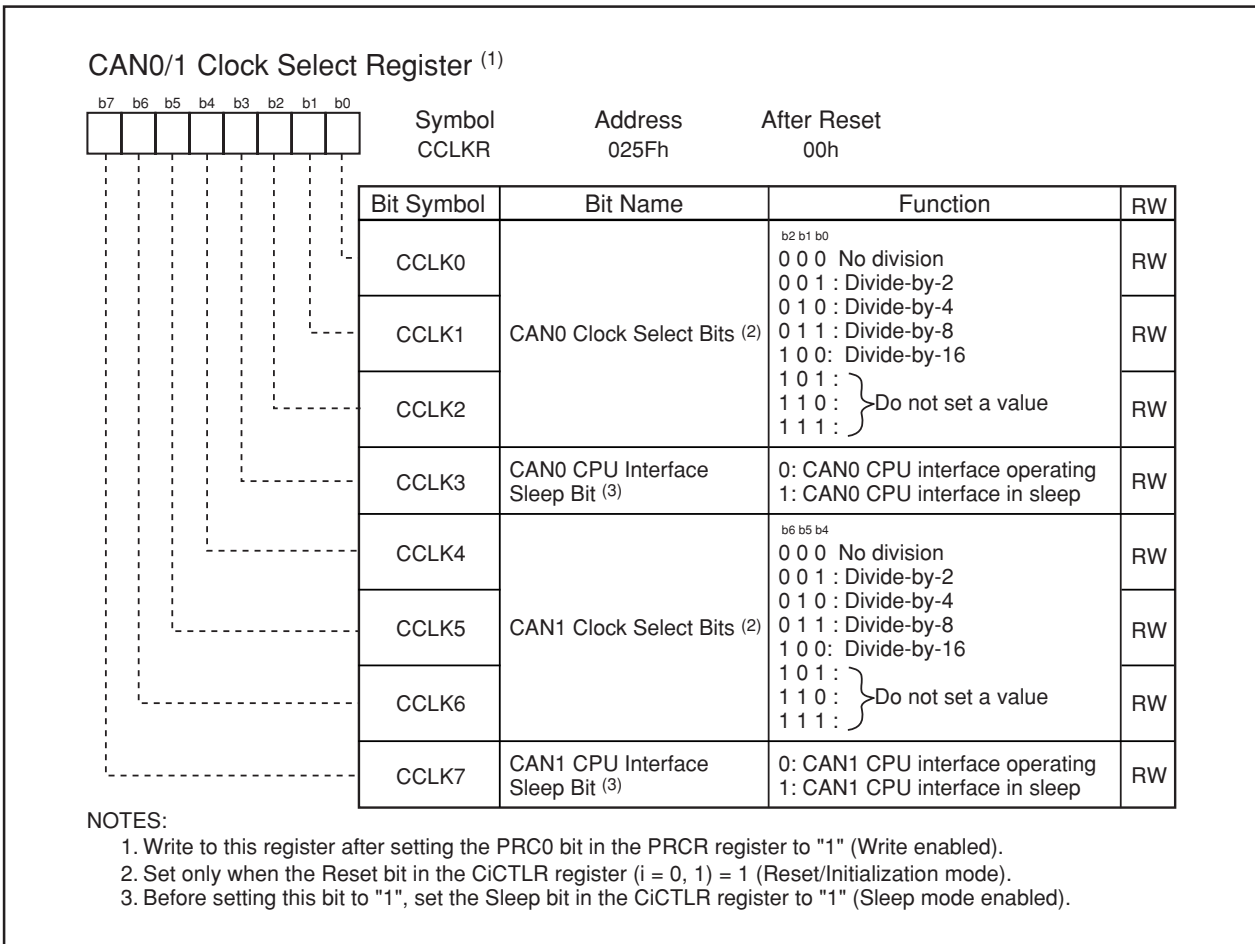


Figure 7.6 CCLKR Register

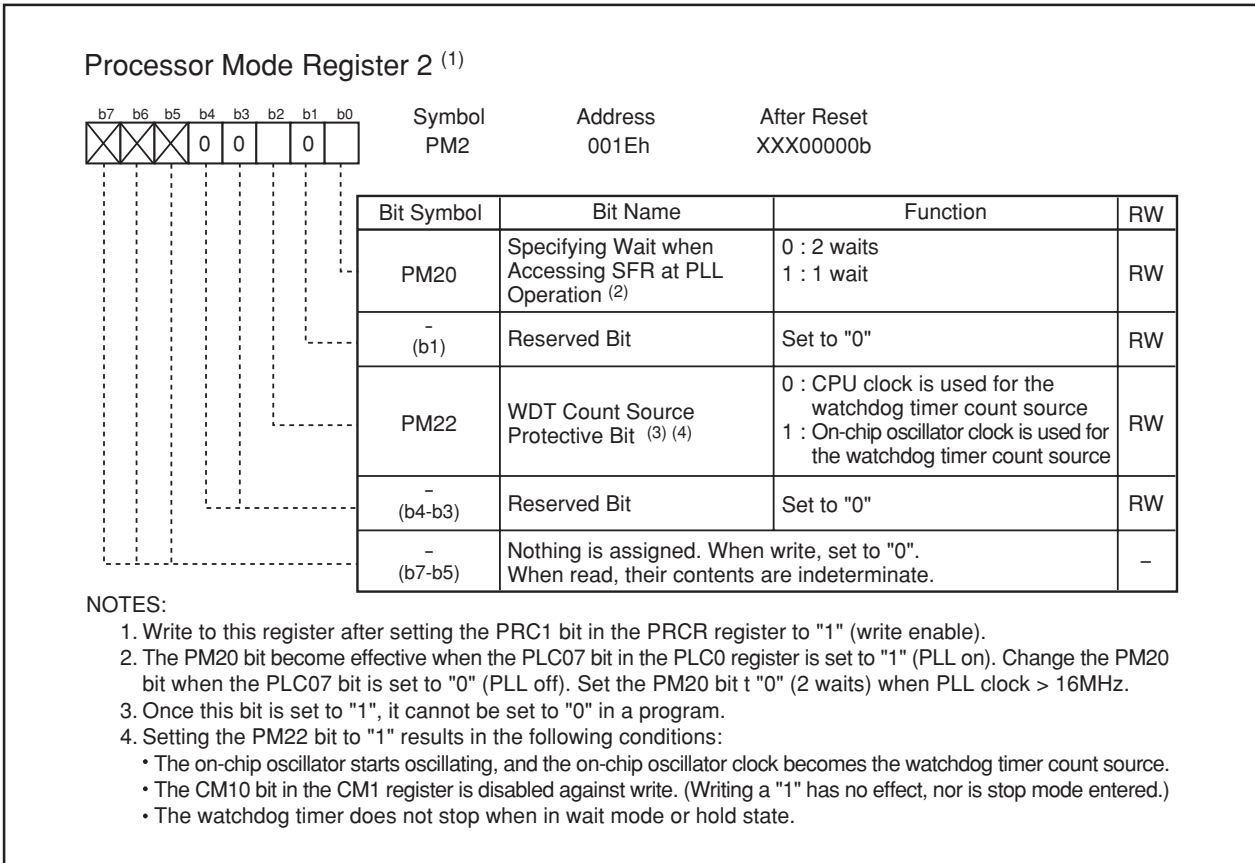


Figure 7.7 PM2 Register

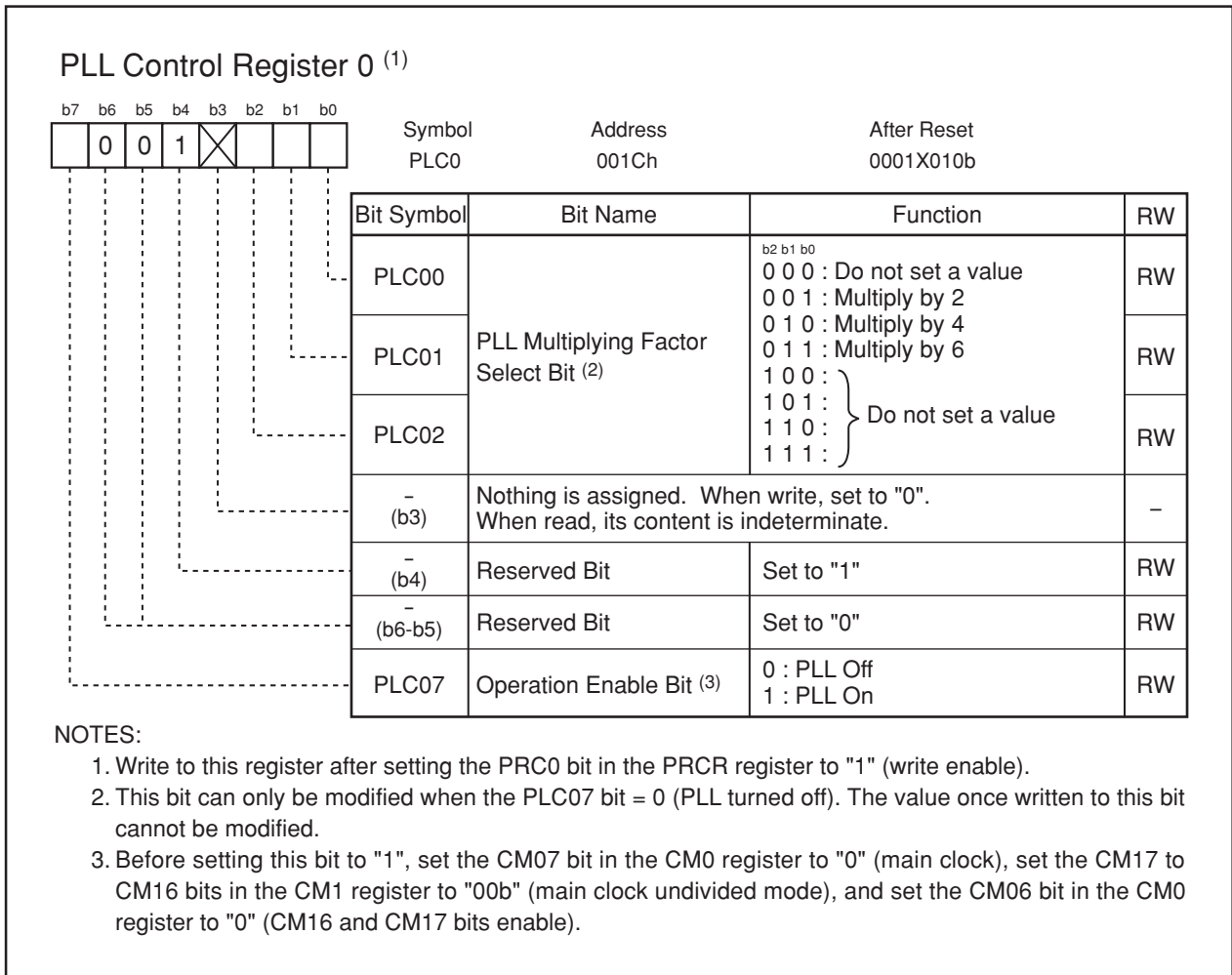


Figure 7.8 PLC0 Register

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

7.1.1 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 7.9 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 **7.4 Power Control**.

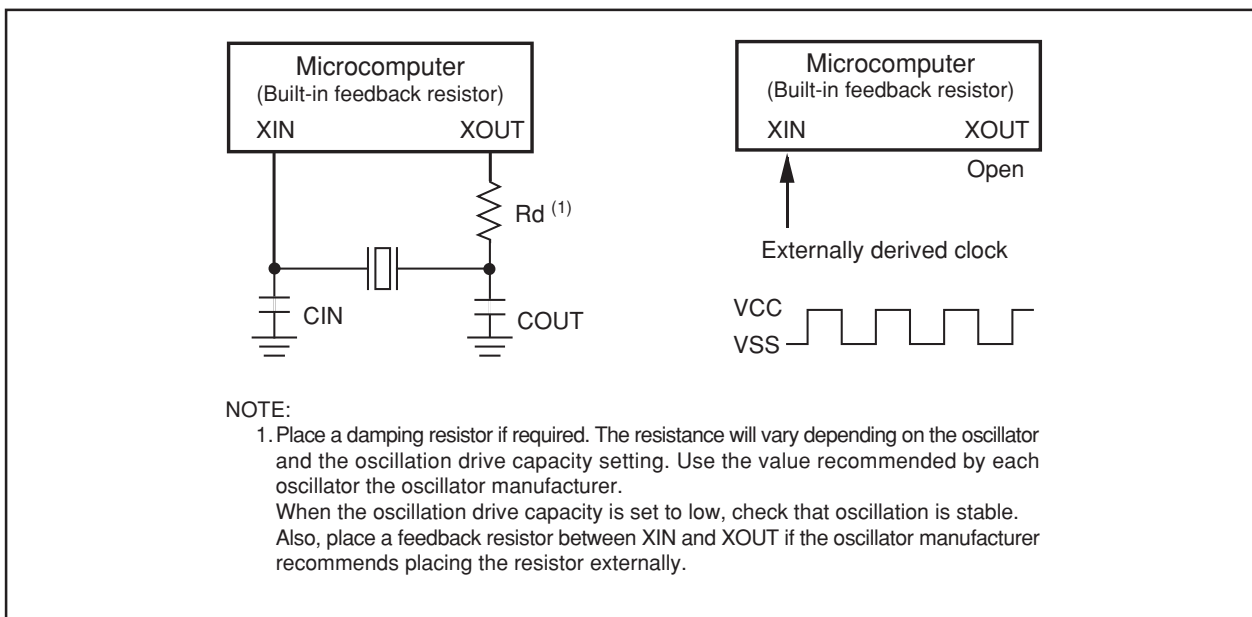


Figure 7.9 Examples of Main Clock Connection Circuit

7.1.2 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 7.10 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 **7.4 Power Control**.

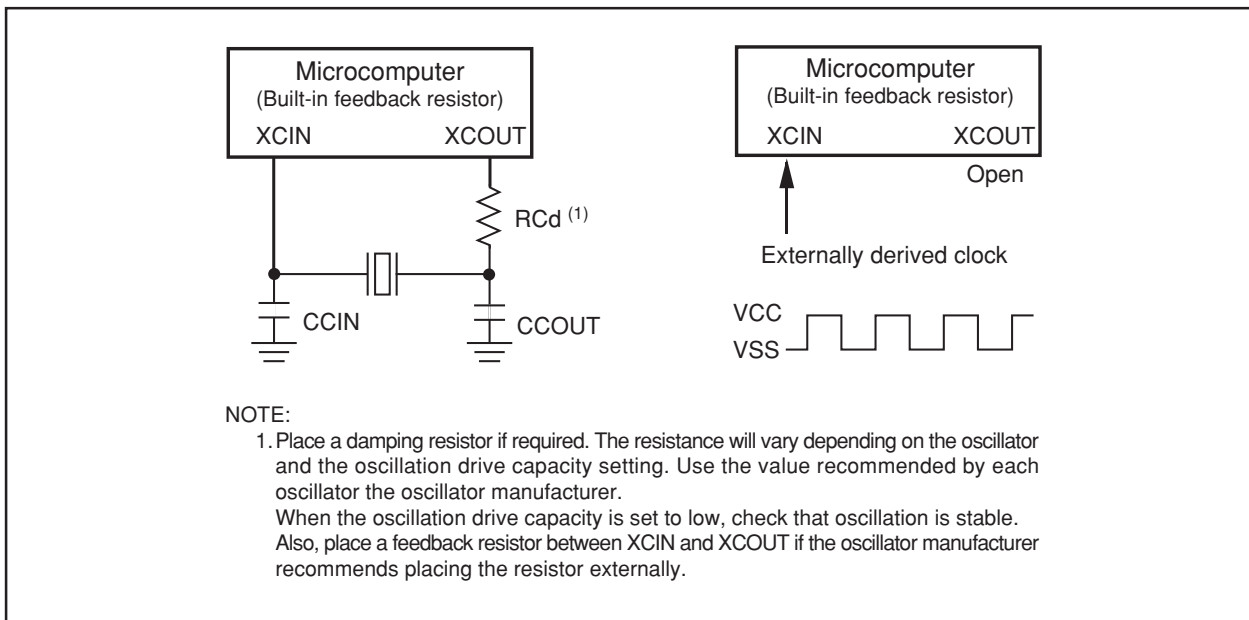


Figure 7.10 Examples of Sub Clock Connection Circuit

7.1.3 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 **10.1 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.

7.1.4 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 $t_{su}(PLL)$ for the PLL clock to be stable, and then set the CM11 bit in the CM1 register to “1”.

Before entering wait mode or stop mode, be sure to set the CM11 bit to “0” (CPU clock source is the main clock). Furthermore, before entering stop mode, be sure to set the PLC07 bit in the PLC0 register to “0” (PLL stops). Figure 7.11 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.

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

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

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

Table 7.2 Example for Setting PLL Clock Frequencies

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

NOTE:

1. PLL clock frequency = 16 MHz , 20 MHz or 24 MHz

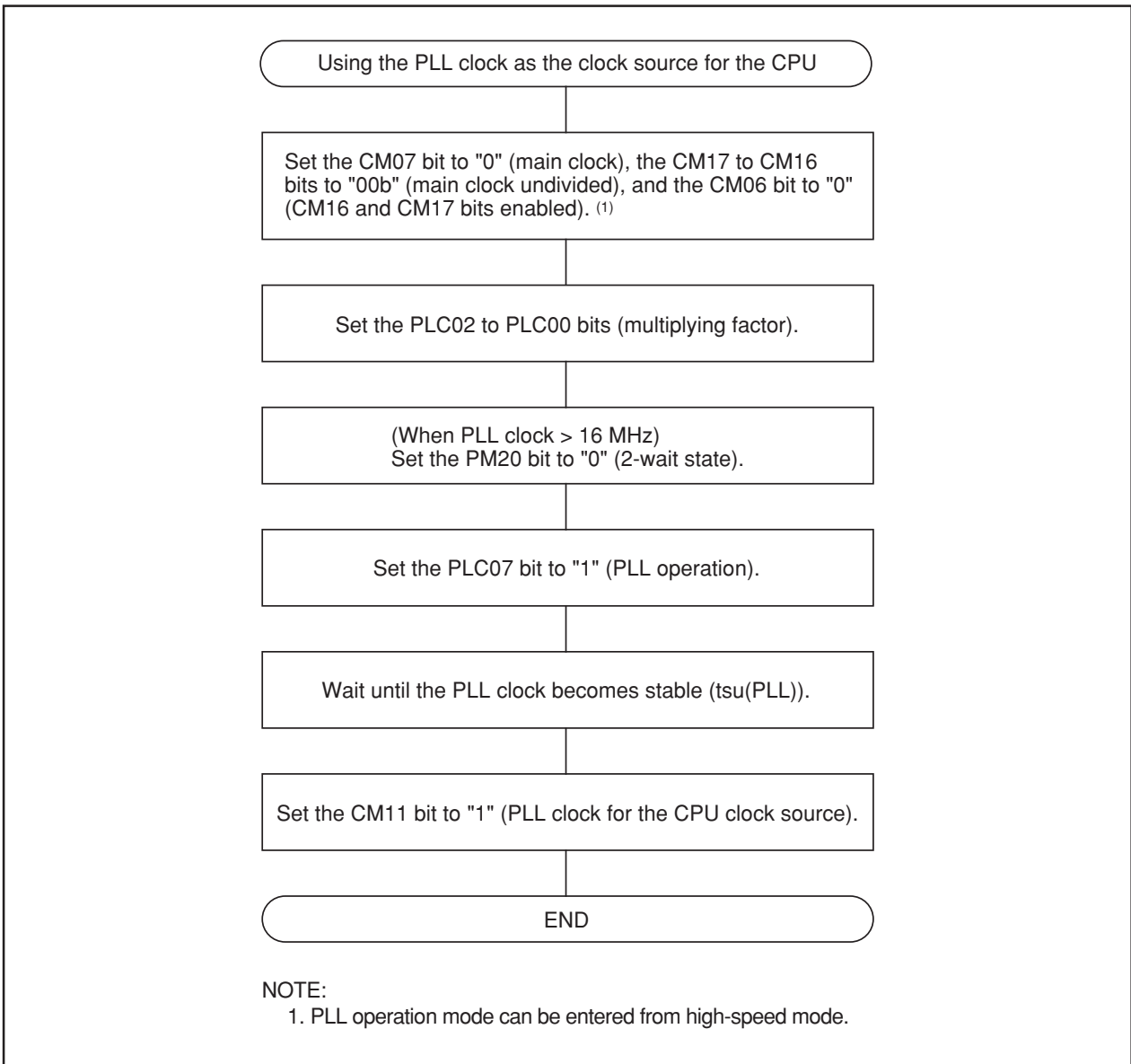


Figure 7.11 Procedure to Use PLL Clock as CPU Clock Source

7.2 CPU Clock and Peripheral Function Clock

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

7.2.1 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.

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).

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

These are operating clocks for the peripheral functions.

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

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

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

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to "1" (peripheral function clock turned off during wait mode), or when the microcomputer is in low power dissipation mode, the f_i , $f_i\text{SIO}$, f_{AD} , f_{CAN0} and f_{CAN1} clocks are turned off ⁽¹⁾.

The f_{C32} 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. f_{CAN0} and f_{CAN1} clocks stop at "H" in CAN0, 1 sleep mode.

7.3 Clock Output Function

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

7.4 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.

7.4.1 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.

7.4.1.1 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.

7.4.1.2 PLL Operation Mode

The main clock multiplied by 2, 4 or 6 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.

7.4.1.3 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.

7.4.1.4 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.

7.4.1.5 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.

7.4.1.6 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.

7.4.1.7 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. 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).

Table 7.3 lists the setting clock related bit and modes.

Table 7.3 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 ⁽¹⁾	1 ⁽¹⁾	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.

7.4.2 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.

7.4.2.1 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.

7.4.2.2 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).

7.4.2.3 Pin Status During Wait Mode

Table 7.4 lists the pin status during wait mode.

Table 7.4 Pin Status During Wait Mode

Pin		Single-Chip Mode
I/O Ports		Retains status before wait mode
CLKOUT	When fC selected	Does not stop
	When f8, f32 selected	<ul style="list-style-type: none"> •CM02 bit = 0: Does not stop •CM02 bit = 1: Retains status before wait mode

7.4.2.4 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 7.5 lists the interrupts to exit wait mode.

Table 7.5 Interrupts to Exit Wait Mode

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

7.4.3 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 V_{RAM} or more, the internal RAM is retained.

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

- $\overline{\text{NMI}}$ interrupt
- Key interrupt
- $\overline{\text{INT}}$ interrupt
- Timer A, Timer B interrupt (when counting external pulses in event counter mode)
- Serial I/O interrupt (when external clock is selected)
- CAN0/1 Wake-up interrupt (when CAN sleep mode is selected)

7.4.3.1 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.

7.4.3.2 Pin Status in Stop Mode

Table 7.6 lists the pin status in stop mode.

Table 7.6 Pin Status in Stop Mode

Pin		Single-Chip Mode
I/O Ports		Retains status before stop mode
CLKOUT	When fC selected	“H”
	When f8, f32 selected	Retains status before stop mode

7.4.3.3 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 7.12 shows the state transition from normal operation mode to stop mode and wait mode. Figure 7.13 shows the state transition in normal operation mode.

Table 7.7 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line show state after transition.

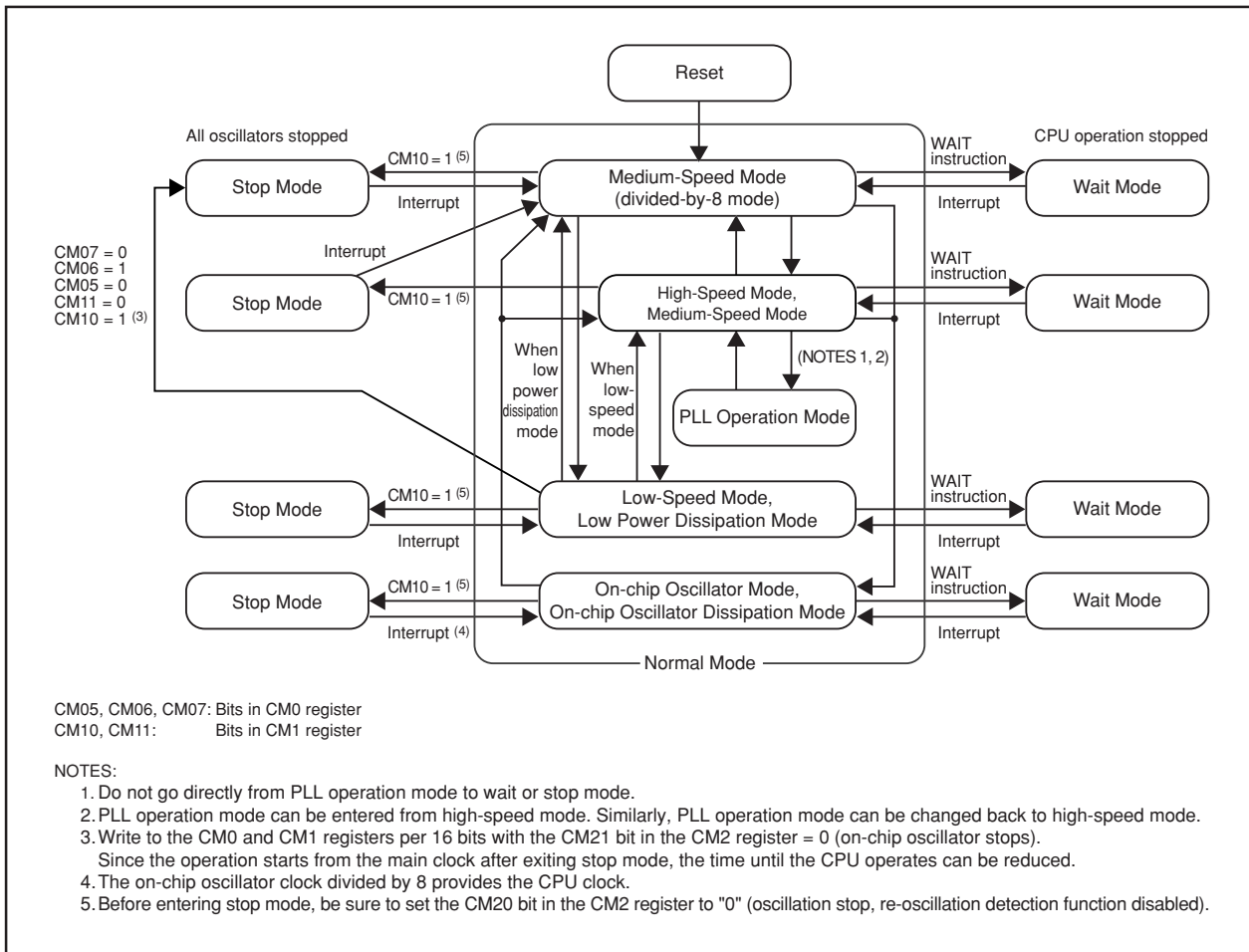


Figure 7.12 State Transition to Stop Mode and Wait Mode

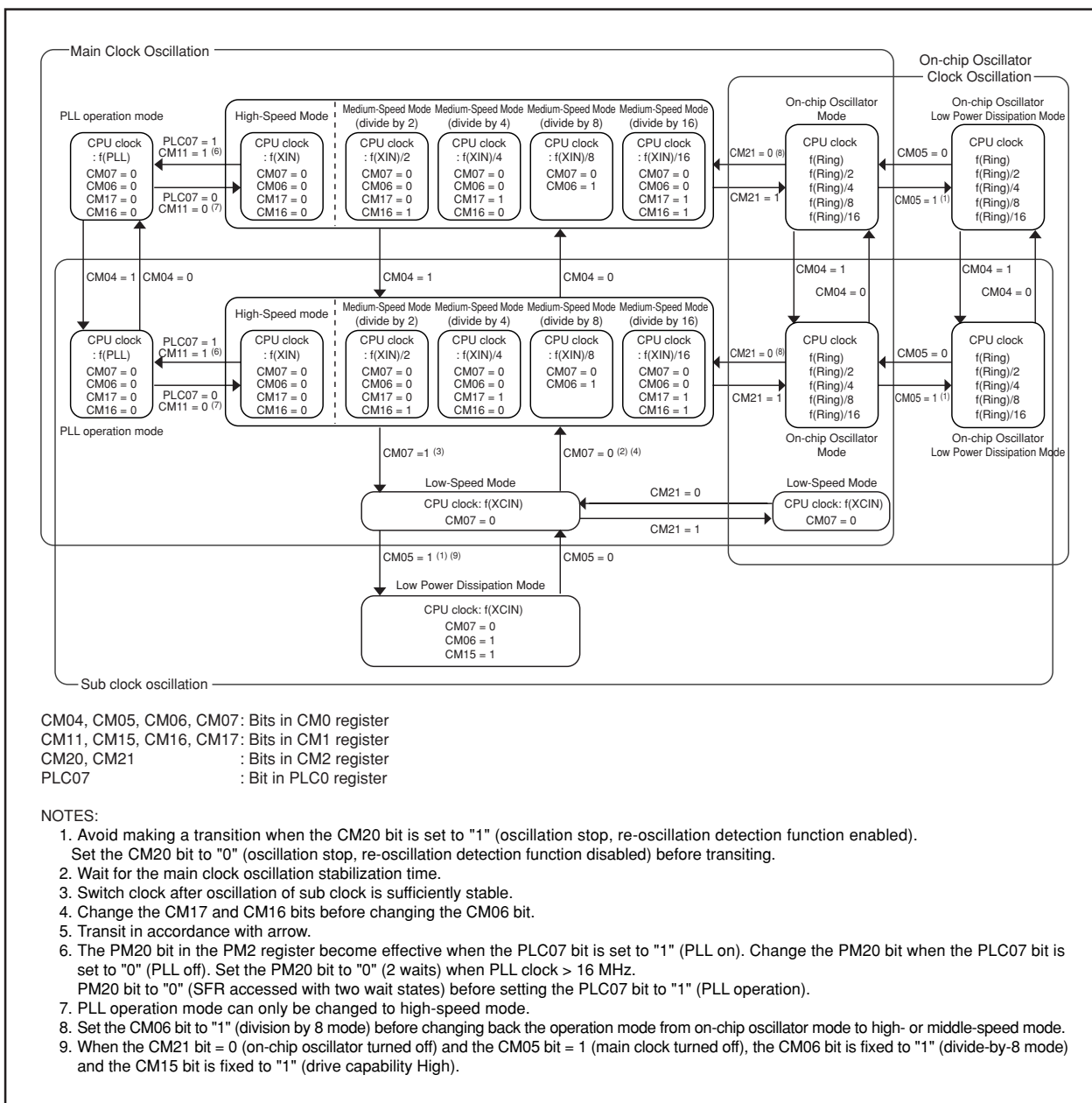


Figure 7.13 State Transition in Normal Operation Mode

Table 7.7 Allowed Transition and Setting

		State after transition							
		High-Speed Mode, Medium-Speed Mode	Low-Speed Mode ⁽²⁾	Low Power Dissipation Mode	PLL Operation Mode (2)	On-chip Oscillator Mode	On-chip Oscillator Low Power Dissipation Mode	Stop Mode	Wait Mode
Current state	High-Speed Mode, Medium-Speed Mode	(NOTE 8)	(9) ⁽⁷⁾	-	(13) ⁽³⁾	(15)	-	(16) ⁽¹⁾	(17)
	Low-Speed Mode ⁽²⁾	(8)	/	(11) ^{(1) (6)}	-	-	-	(16) ⁽¹⁾	(17)
	Low Power Dissipation Mode	-	(10)	/	-	-	-	(16) ⁽¹⁾	(17)
	PLL Operation Mode ⁽²⁾	(12) ⁽³⁾	-	-	/	-	-	-	-
	On-chip Oscillator Mode	(14) ⁽⁴⁾	-	-	-	(NOTE 8)	(11) ⁽¹⁾	(16) ⁽¹⁾	(17)
	On-chip Oscillator Low Power Dissipation Mode	-	-	-	-	(10)	(NOTE 8)	(16) ⁽¹⁾	(17)
	Stop Mode	(18) ⁽⁵⁾	(18)	(18)	-	(18) ⁽⁵⁾	(18) ⁽⁵⁾	/	-
	Wait Mode	(18)	(18)	(18)	-	(18)	(18)	-	/

-: Cannot transit

NOTES:

- 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.
- 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.
- PLL operation mode can only be entered from and changed to high-speed mode.
- Set the CM06 bit to "1" (division by 8 mode) before transiting from on-chip oscillator mode to high- or medium-speed mode.
- When exiting stop mode, the CM06 bit is set to "1" (division by 8 mode).
- If the CM05 bit is set to "1" (main clock stop), then the CM06 bit is set to "1" (division by 8 mode).
- A transition can be made only when sub clock is oscillating.
- State transitions within the same mode (divide-by-n values changed or sub clock oscillation turned on or off) are shown in the table below.

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

9. () :setting method. 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 division by 2 mode
(5) CM06=0 CM17=1 CM16=0	CPU clock division by 4 mode
(6) CM06=0 CM17=1 CM16=1	CPU clock division by 16 mode
(7) CM06=1	CPU clock division by 8 mode
(8) CM07=0	Main clock, PLL clock or on-chip oscillator clock selected
(9) CM07=1	Sub clock selected
(10) CM05=0	Main clock oscillating
(11) CM05=1	Main clock turned off
(12) PLC07=0 CM11=0	Main clock selected
(13) PLC07=1 CM11=1	PLL clock selected
(14) CM21=0	Main clock or PLL clock selected
(15) CM21=1	On-chip oscillator clock selected
(16) CM10=1	Transition to stop mode
(17) WAIT instruction	Transition to wait mode
(18) Hardware interrupt	Exit stop mode or wait mode

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

7.5 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 7.8 lists a specification overview of the oscillation stop and re-oscillation detection function.

Table 7.8 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"> •Reset occurs (when CM27 bit = 0) •Oscillation stop, re-oscillation detection interrupt occurs (when the CM27 bit =1)

7.5.1 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. 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").

7.5.2 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

7.5.3 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 7.14 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".

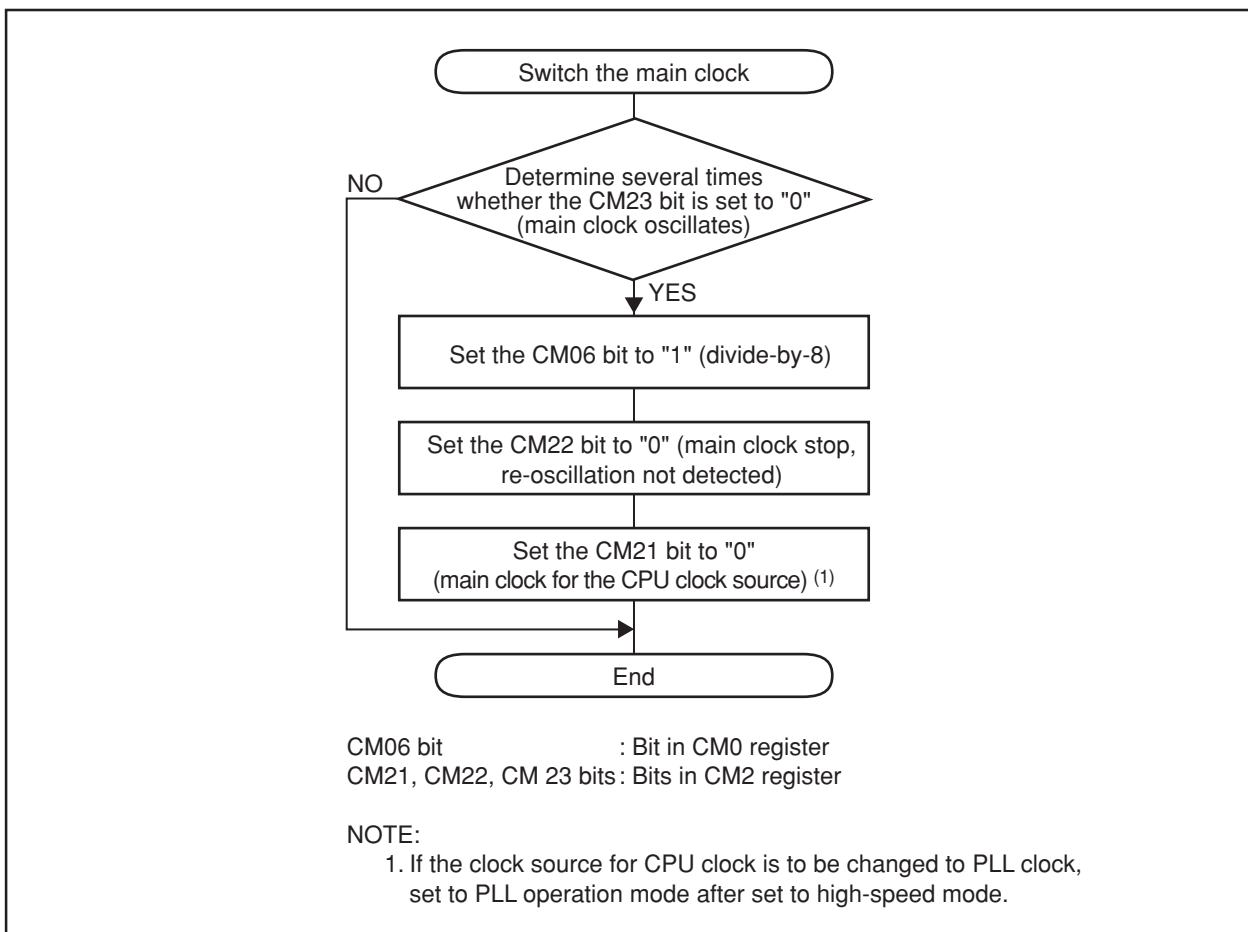


Figure 7.14 Procedure to Switch Clock Source from On-chip Oscillator to Main Clock

8. Protection

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

- 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, S3C, S4C, S5C and S6C registers ⁽¹⁾.

NOTE:

1. The S5C and S6C registers are only in the 128-pin version.

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.

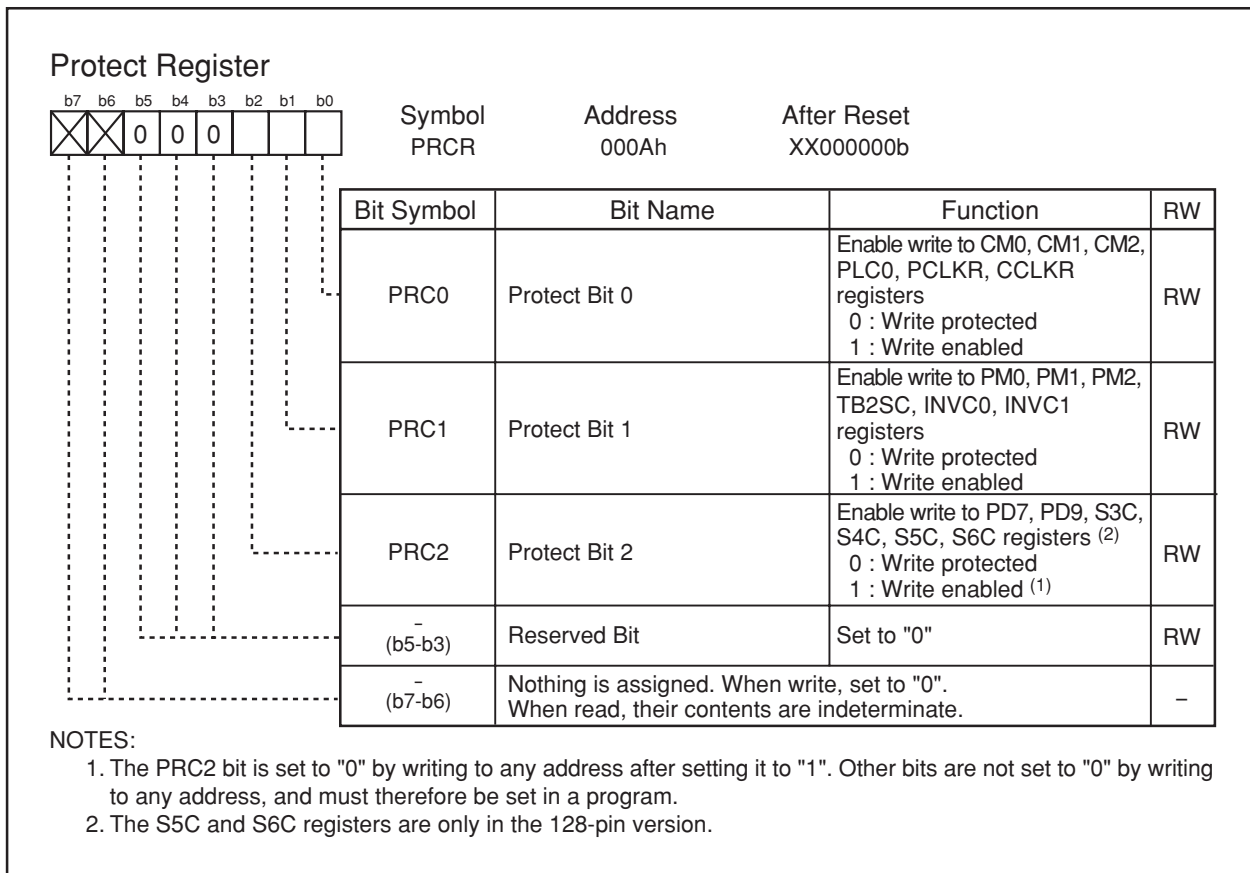


Figure 8.1 PRCR Register

9. Interrupt

9.1 Type of Interrupts

Figure 9.1 shows the types of interrupts.

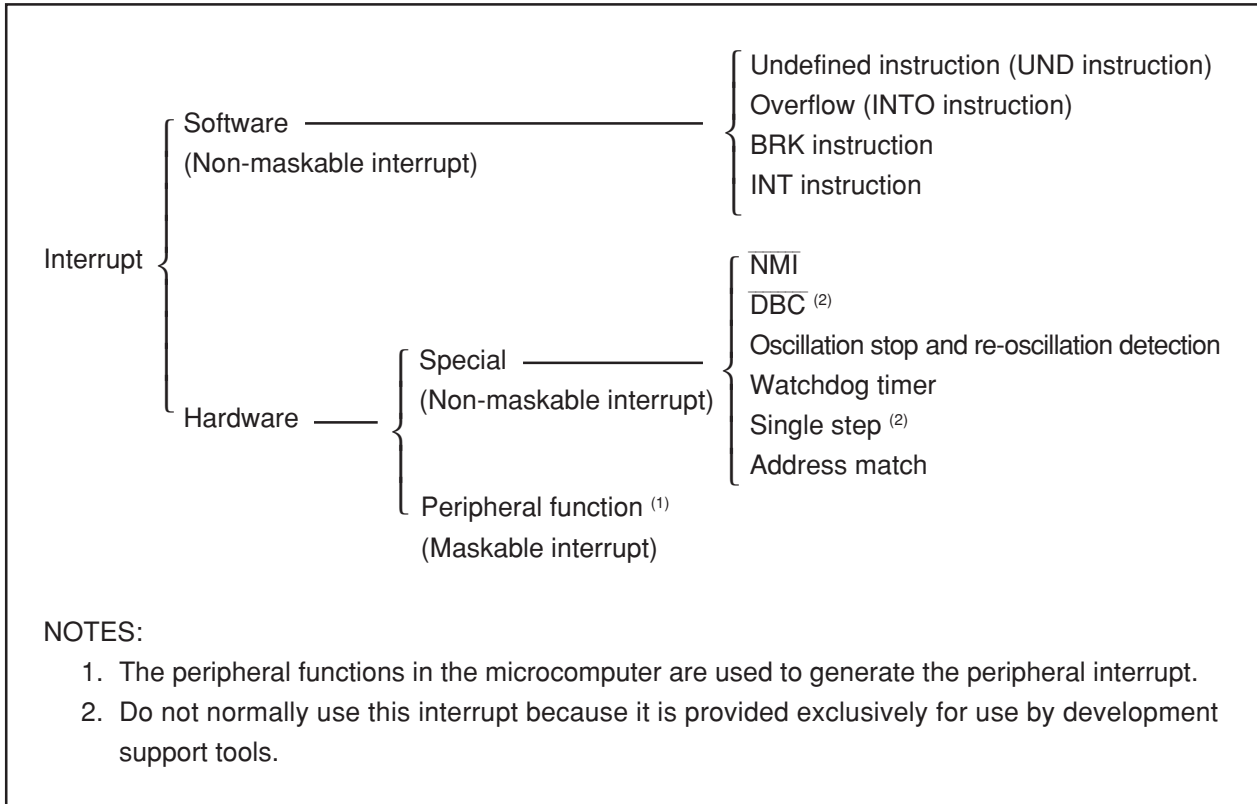


Figure 9.1 Interrupts

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

9.2 Software Interrupts

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

9.2.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

9.2.2 Overflow Interrupt

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

9.2.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

9.2.4 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.

9.3 Hardware Interrupts

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

9.3.1 Special Interrupts

Special interrupts are non-maskable interrupts.

9.3.1.1 $\overline{\text{NMI}}$ 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 **9.7 $\overline{\text{NMI}}$ Interrupt**.

9.3.1.2 $\overline{\text{DBC}}$ Interrupt

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

9.3.1.3 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 **10. Watchdog Timer**.

9.3.1.4 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 **7. Clock Generating Circuit**.

9.3.1.5 Single-Step Interrupt

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

9.3.1.6 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 **9.10 Address Match Interrupt**.

9.3.2 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 9.2 Relocatable Vector Tables** about how the peripheral function interrupt occurs. Refer to the descriptions of each function for details.

9.4 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 9.2 shows the interrupt vector.

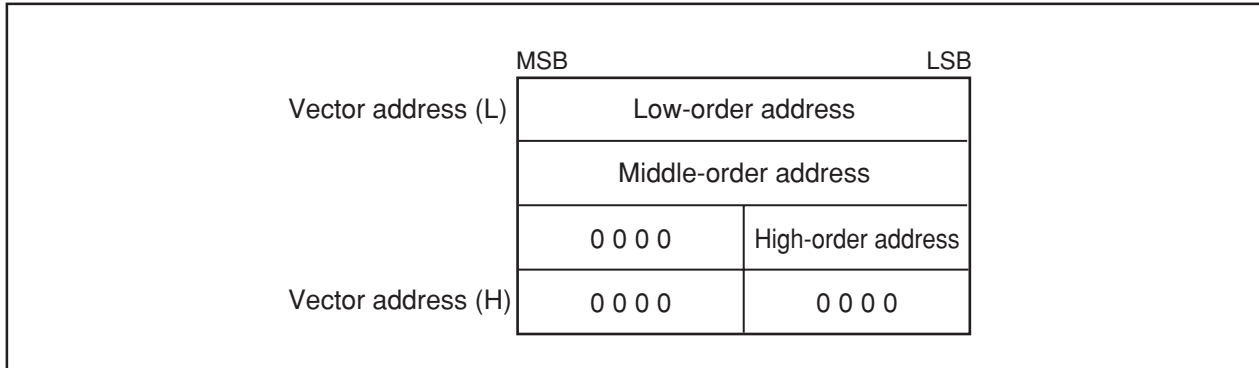


Figure 9.2 Interrupt Vector

9.4.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDCh to FFFFFh. Table 9.1 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 **20.2 Functions to Prevent Flash Memory from Rewriting**.

Table 9.1 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 Series Software Manual
Overflow (INTO instruction)	FFFE0h to FFFE3h	
BRK Instruction ⁽²⁾	FFFE4h to FFFE7h	
Address Match	FFFE8h to FFFEbH	9.10 Address Match Interrupt
Single Step ⁽¹⁾	FFFECh to FFEFh	
Oscillation Stop and Re-oscillation Detection, Watchdog Timer	FFFF0h to FFFF3h	7. Clock Generating Circuit 10. Watchdog Timer
DBC ⁽¹⁾	FFFF4h to FFFF7h	
NMI	FFFF8h to FFFFBh	9.7 NMI Interrupt
Reset	FFFFCh to FFFFFh	5. Reset

NOTES:

- Do not normally use this interrupt because it is provided exclusively for use by development support tools.
- If the contents of address FFE7h is FFh, program execution starts from the address shown by the vector in the relocatable vector table.

9.4.2 Relocatable Vector Tables

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

Table 9.2 Relocatable Vector Tables

Interrupt Source	Vector Address ⁽¹⁾ Address (L) to Address (H)	Software Interrupt Number	Reference	
BRK Instruction ⁽²⁾	+0 to +3 (0000h to 0003h)	0	M16C/60, M16C/20 Series Software Manual	
CAN0/1 Wake-up ⁽¹⁰⁾	+4 to +7 (0004h to 0007h)	1	18. 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	9.6 INT Interrupt	
Timer B5, SI/O5 ⁽¹²⁾	+20 to +23 (0014h to 0017h)	5	12. Timers	
Timer B4, UART1 Bus Collision Detection ^{(3) (9)}	+24 to +27 (0018h to 001Bh)	6	14. Serial I/O	
Timer B3, UART0 Bus Collision Detection ^{(4) (9)}	+28 to +31 (001Ch to 001Fh)	7		
CAN1 Successful Reception, SIO4, INT5 ⁽⁵⁾	+32 to +35 (0020h to 0023h)	8	18. CAN Module, 14. Serial I/O,	
CAN1 Successful Transmission, SIO3, INT4 ⁽⁶⁾	+36 to +39 (0024h to 0027h)	9	9.6 INT Interrupt	
UART2 Bus Collision Detection ⁽⁹⁾	+40 to +43 (0028h to 002Bh)	10	14. Serial I/O	
DMA0	+44 to +47 (002Ch to 002Fh)	11	11. DMAC	
DMA1	+48 to +51 (0030h to 0033h)	12		
CAN0/1 Error ^{(11) (17)}	+52 to +55 (0034h to 0037h)	13	18. CAN Module	
A/D, Key Input ^{(7) (17)}	+56 to +59 (0038h to 003Bh)	14	15. A/D Converter, 9.8 Key Input Interrupt	
UART2 Transmission, NACK2 ⁽⁸⁾	+60 to +63 (003Ch to 003Fh)	15	14. Serial I/O	
UART2 Reception, ACK2 ⁽⁸⁾	+64 to +67 (0040h to 0043h)	16		
UART0 Transmission, NACK0 ⁽⁸⁾	+68 to +71 (0044h to 0047h)	17		
UART0 Reception, ACK0 ⁽⁸⁾	+72 to +75 (0048h to 004Bh)	18		
UART1 Transmission, NACK1 ⁽⁸⁾	+76 to +79 (004Ch to 004Fh)	19		
UART1 Reception, ACK1 ⁽⁸⁾	+80 to +83 (0050h to 0053h)	20		
Timer A0	+84 to +87 (0054h to 0057h)	21		12. Timers
Timer A1	+88 to +91 (0058h to 005Bh)	22		
Timer A2, INT7 ⁽¹³⁾	+92 to +95 (005Ch to 005Fh)	23	12. Timers	
Timer A3, INT6 ⁽¹⁴⁾	+96 to +99 (0060h to 0063h)	24	9.6 INT Interrupt	
Timer A4	+100 to +103 (0064h to 0067h)	25	12. Timers	
Timer B0, SI/O6 ⁽¹⁵⁾	+104 to +107 (0068h to 006Bh)	26	12. Timers, 14. Serial I/O	
Timer B1, INT8 ⁽¹⁶⁾	+108 to +111 (006Ch to 006Fh)	27	12. Timers, 9.6 INT Interrupt	
Timer B2	+112 to +115 (0070h to 0073h)	28	12. Timers	
INT0	+116 to +119 (0074h to 0077h)	29	9.6 INT Interrupt	
INT1	+120 to +123 (0078h to 007Bh)	30		
INT2	+124 to +127 (007Ch to 007Fh)	31		
INT Instruction Interrupt ⁽²⁾	+128 to +131 (0080h to 0083h) to +252 to +255 (00FCh to 00FFh)	32 to 63	M16C/60, M16C/20 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.
Furthermore, use the IFSR03 bit in the IFSR0 register to select, when selecting SI/O4 or CAN1 successful reception.
- Use the IFSR16 bit in the IFSR1 register to select.
Furthermore, use the IFSR00 bit in the IFSR0 register to select, when selecting SI/O3 or CAN1 successful transmission.
- Use the IFSR01 bit in the IFSR0 register to select.
- During I²C mode, NACK and ACK interrupts comprise the interrupt source.
- Bus collision detection: During IE mode, this bus collision detection constitutes the cause of an interrupt.
During I²C 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.
- Use the IFSR04 bit in the IFSR0 register to select.
SI/O5 is only in the 128-pin version. In the 100-pin version, set the IFSR04 bit to "0" (Timer B5).
- Use the IFSR20 bit in the IFSR2 register to select.
INT7 is only in the 128-pin version. In the 100-pin version, set the IFSR20 bit to "0" (Timer A2).
- Use the IFSR21 bit in the IFSR2 register to select.
INT6 is only in the 128-pin version. In the 100-pin version, set the IFSR21 bit to "0" (Timer A3).
- Use the IFSR05 bit in the IFSR0 register to select.
SI/O6 is only in the 128-pin version. In the 100-pin version, set the IFSR05 bit to "0" (Timer B0).
- Use the IFSR22 bit in the IFSR2 register to select.
INT8 is only in the 128-pin version. In the 100-pin version, set the IFSR22 bit to "0" (Timer B1).
- If the PCLK6 bit in the PCLKR register is set to "1", software interrupt number 13 can be changed to CAN0/1 error or key input interrupt, and software interrupt number 14 can be changed to A/D interrupt. (The software interrupt number of key input is changed from 14 to 13.) Use the IFSR26 bit in the IFSR2 register to select when selecting CAN0/1 error or key input.

9.5 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 9.3 and 9.4 show the interrupt control registers.

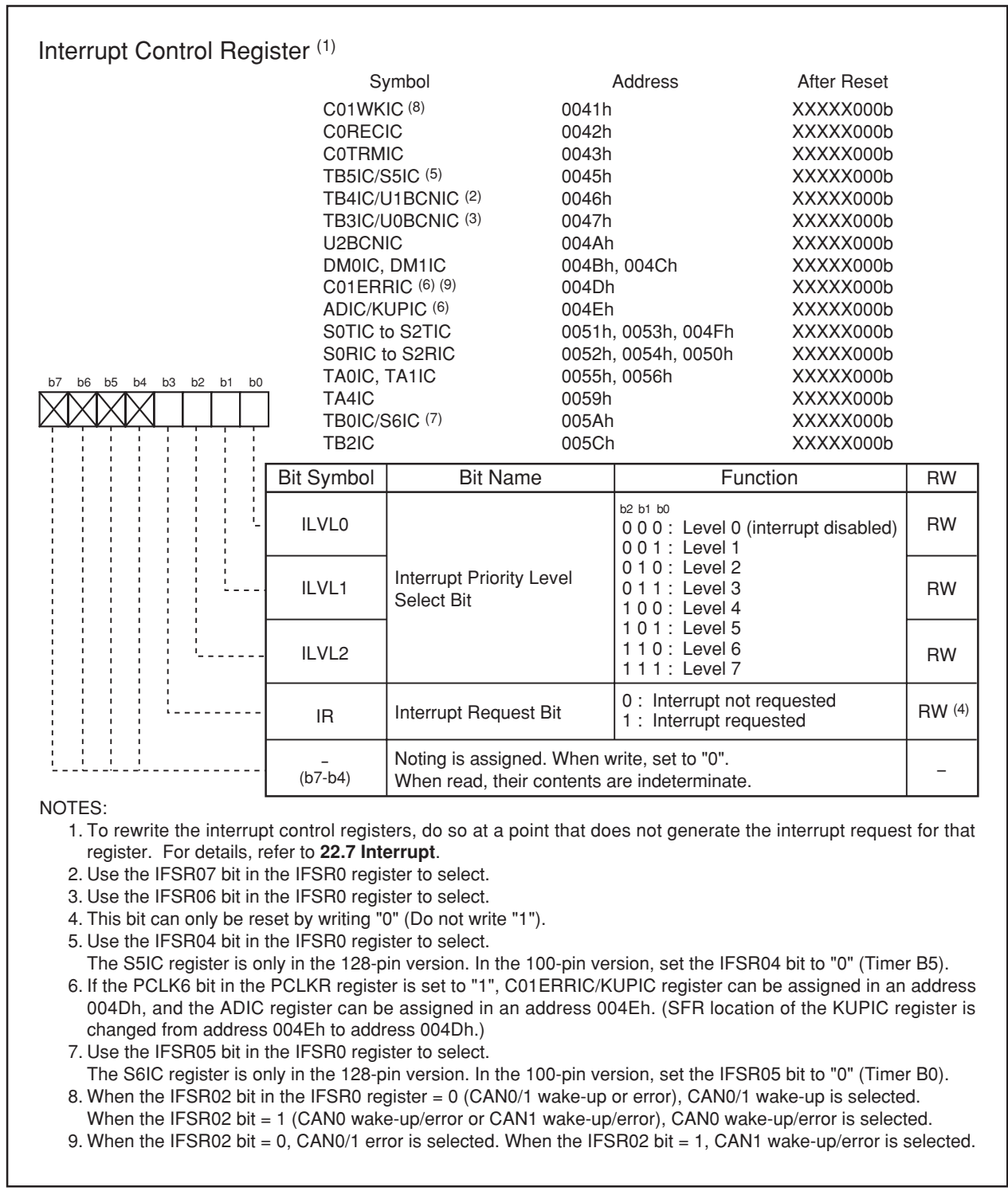


Figure 9.3 Interrupt Control Registers (1)

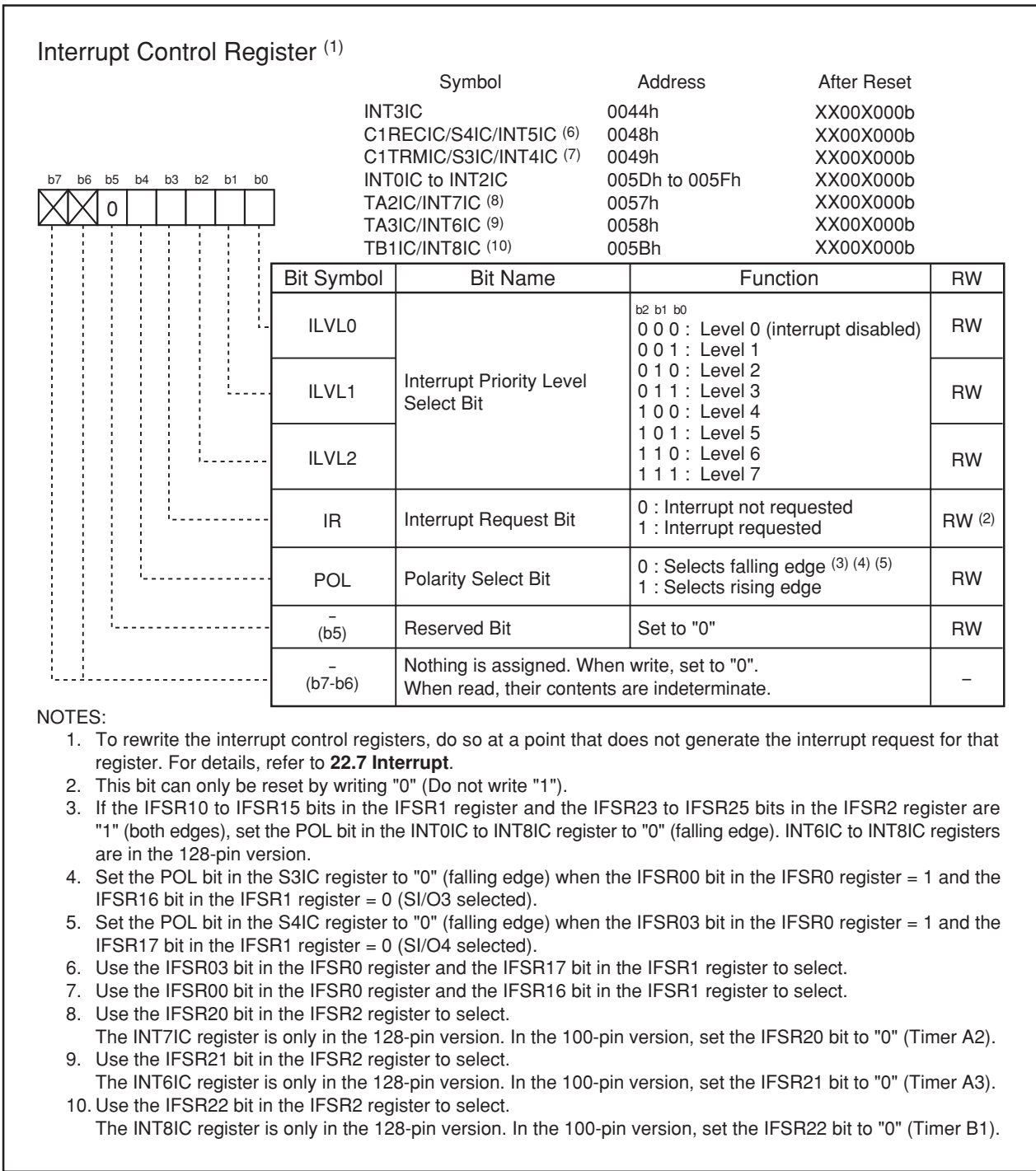


Figure 9.4 Interrupt Control Registers (2)

9.5.1 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.

9.5.2 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.

9.5.3 ILVL2 to ILVL0 Bits and IPL

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

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

The following are conditions under which an interrupt is accepted:

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

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

Table 9.3 Settings of Interrupt Priority Levels


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

Table 9.4 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

9.5.4 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 9.5 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 ⁽¹⁾ 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.

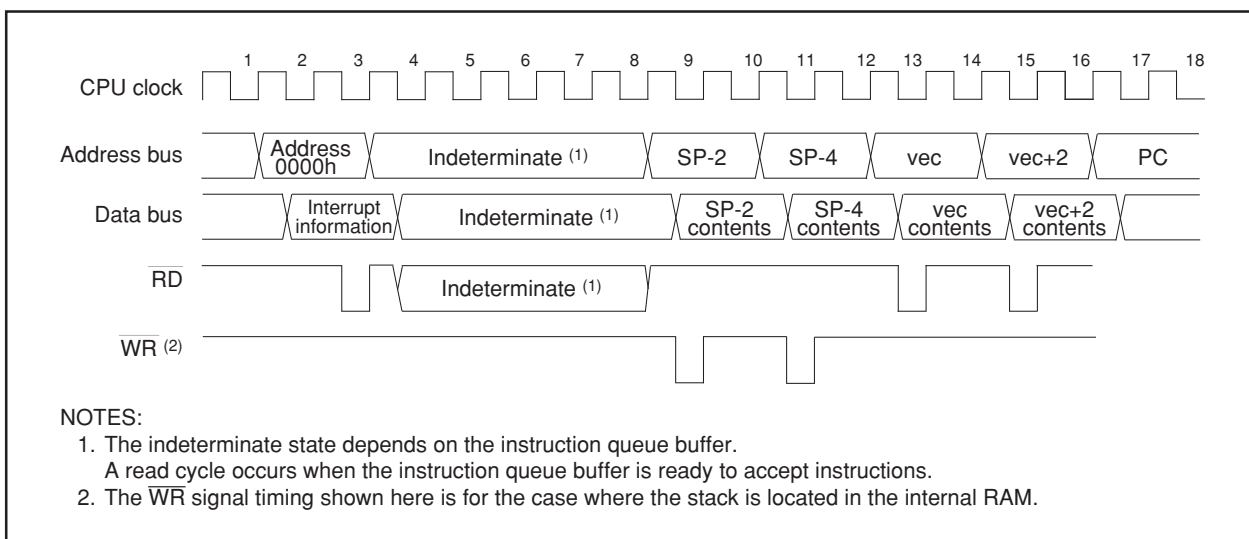


Figure 9.5 Time Required for Executing Interrupt Sequence

9.5.5 Interrupt Response Time

Figure 9.6 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 9.6) and a time during which the interrupt sequence is executed ((b) on Figure 9.6).

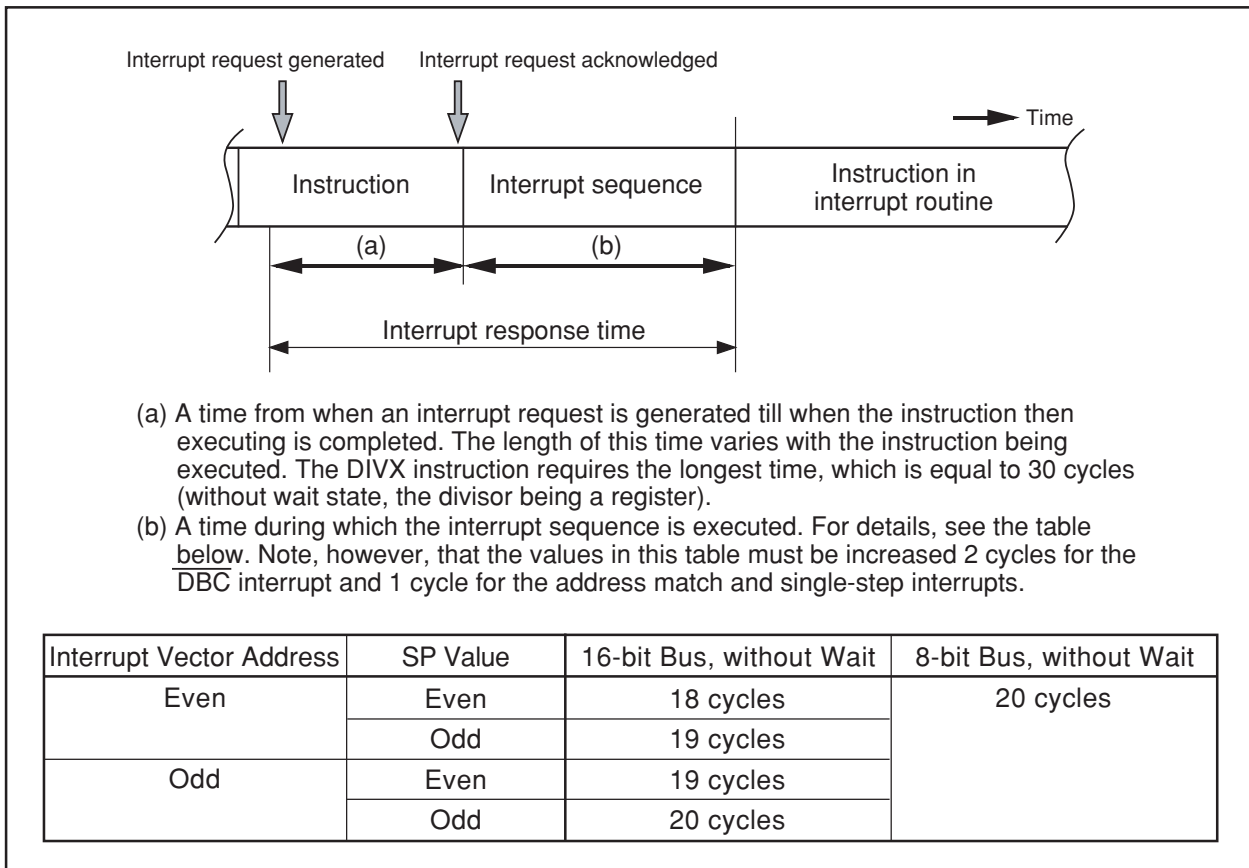


Figure 9.6 Interrupt response time

9.5.6 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 9.5 is set in the IPL. Table 9.5 shows the IPL values of software and special interrupts when they are accepted.

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

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

9.5.7 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 9.7 shows the stack status before and after an interrupt request is accepted.

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

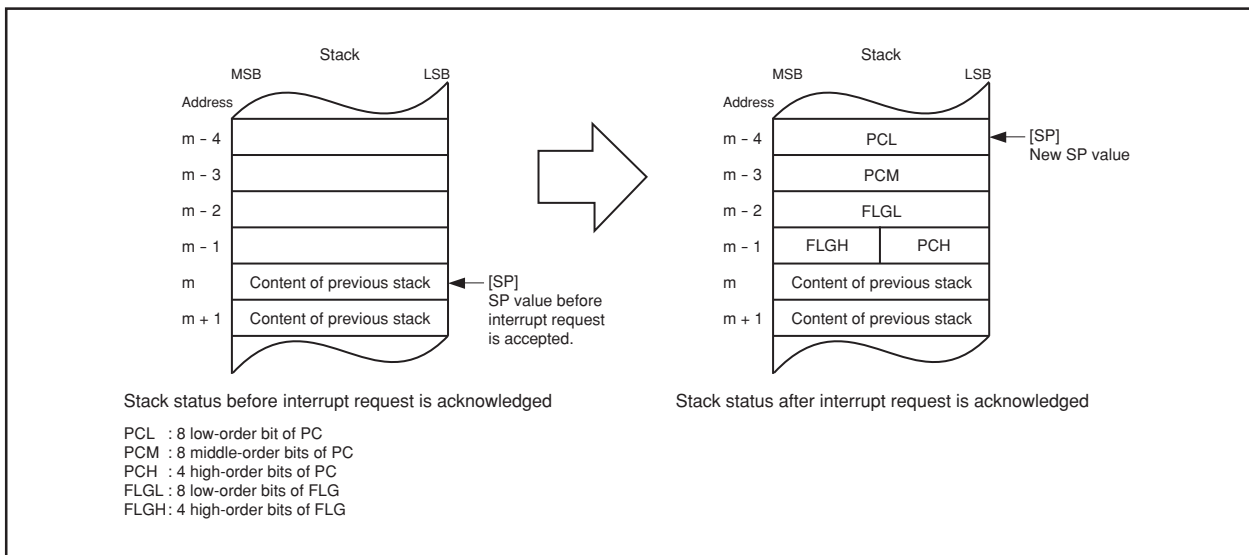


Figure 9.7 Stack Status Before and After Acceptance of Interrupt Request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP⁽¹⁾, at the time of acceptance of an interrupt request, is even or odd. If the 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 9.8 shows the operation of the saving registers.

NOTE:

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

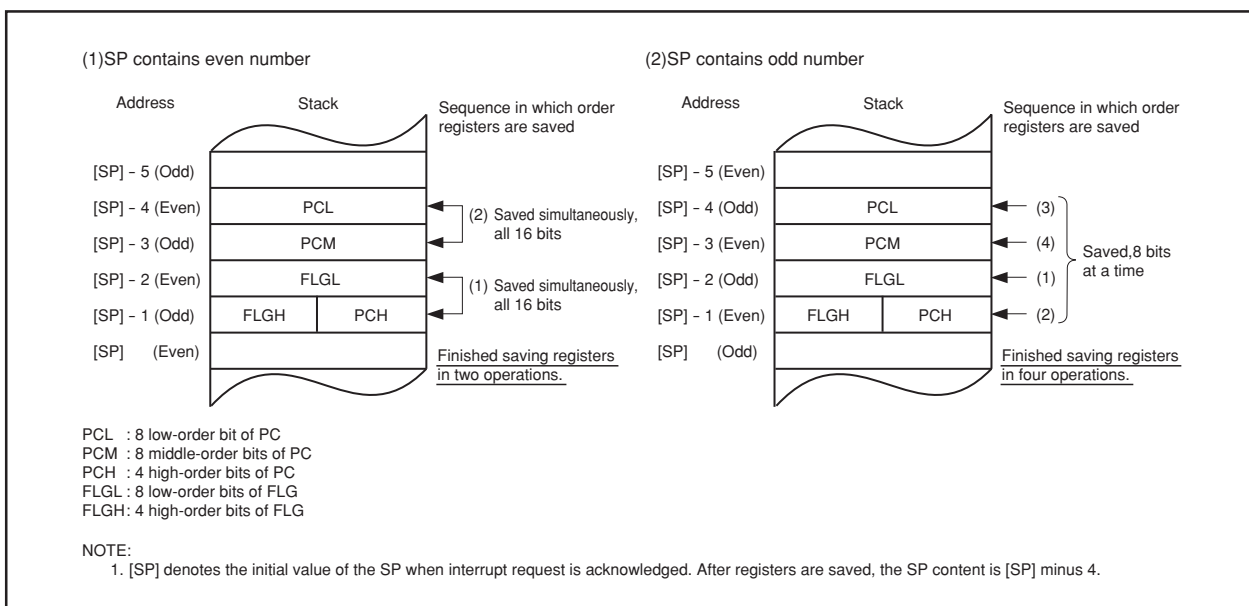


Figure 9.8 Operation of Saving Registers

9.5.8 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.

9.5.9 Interrupt Priority

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

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

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

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

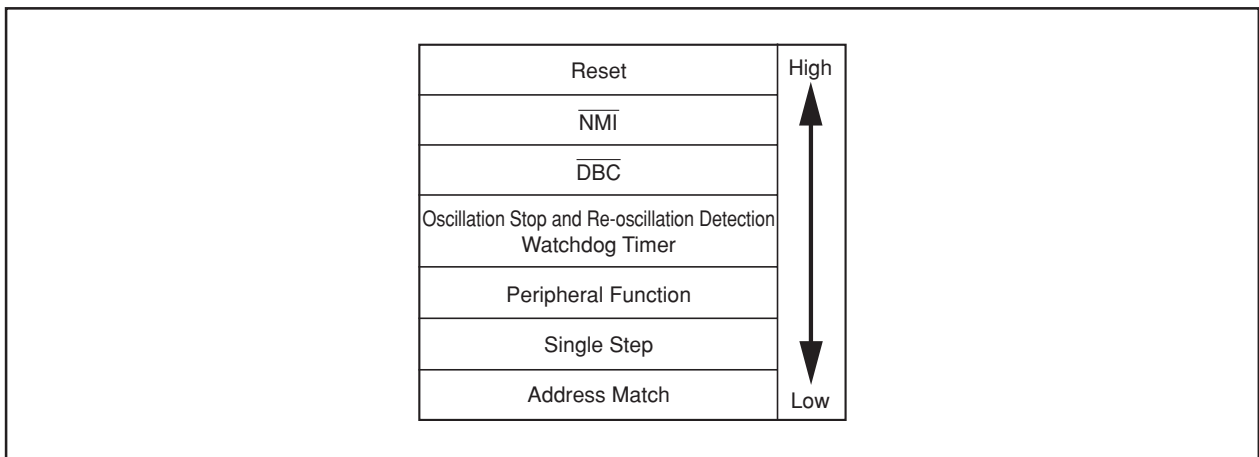


Figure 9.9 Hardware Interrupt Priority

9.5.10 Interrupt Priority Resolution Circuit

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

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

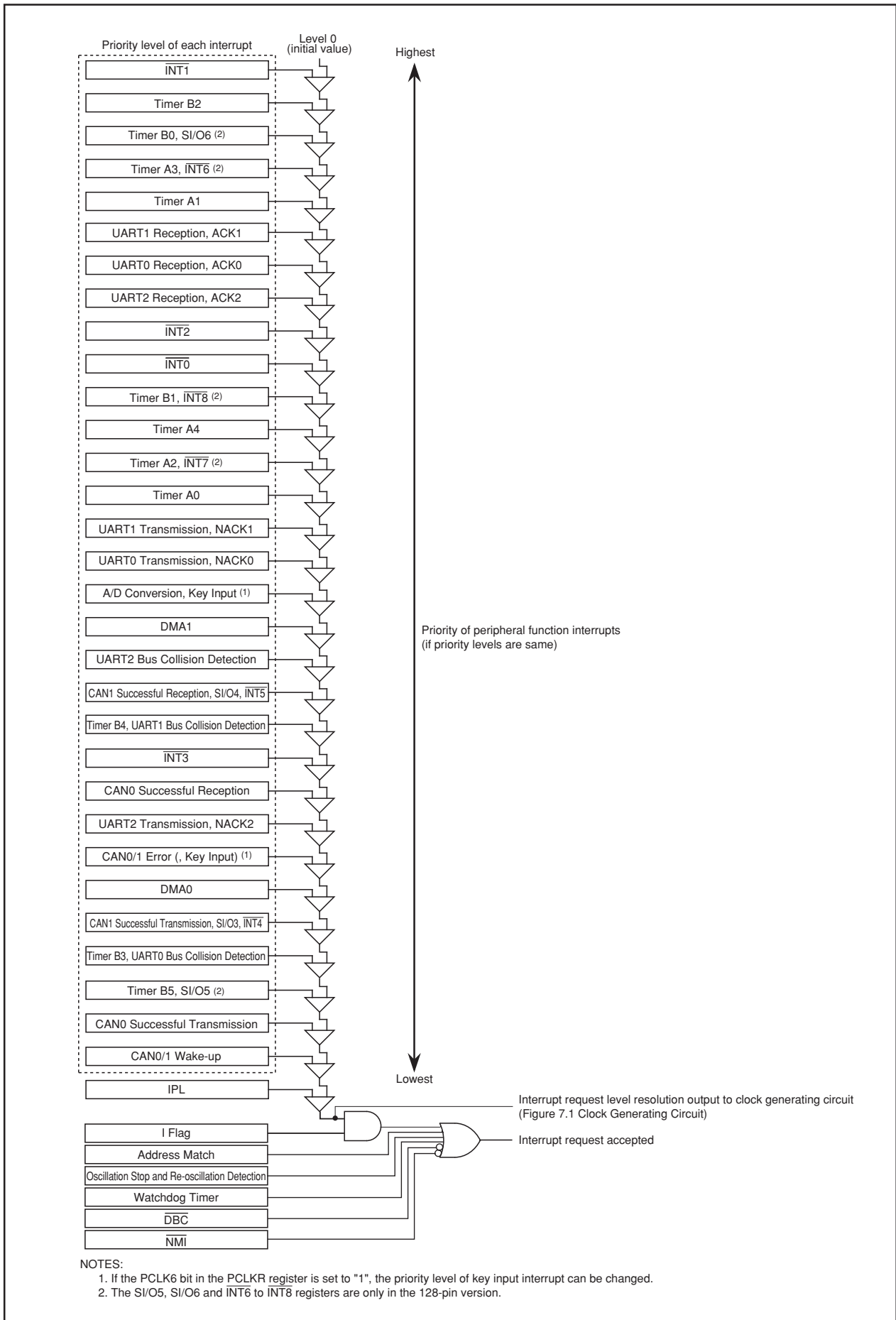


Figure 9.10 Interrupts Priority Select Circuit

9.6 INT Interrupt

$\overline{\text{INT}}_i$ interrupt ($i = 0$ to 8)⁽¹⁾ is triggered by the edges of external inputs. The edge polarity is selected using the IFSR10 to IFSR15 bits in the IFSR1 register and the IFSR23 to IFSR25 bits in the IFSR2 register.

$\overline{\text{INT}}_4$ share the interrupt vector and interrupt control register with CAN1 successful transmission and SI/O3, $\overline{\text{INT}}_5$ share with CAN1 successful reception and SI/O4, $\overline{\text{INT}}_6$ share with Timer A3, $\overline{\text{INT}}_7$ share with Timer A2, $\overline{\text{INT}}_8$ share with Timer B1. To use the $\overline{\text{INT}}_4$ to $\overline{\text{INT}}_8$ interrupts⁽¹⁾, set the each bits as follows.

- To use the $\overline{\text{INT}}_4$ interrupt: Set the IFSR16 bit in the IFSR1 register to “1” ($\overline{\text{INT}}_4$).
- To use the $\overline{\text{INT}}_5$ interrupt: Set the IFSR17 bit in the IFSR1 register to “1” ($\overline{\text{INT}}_5$).
- To use the $\overline{\text{INT}}_6$ interrupt: Set the IFSR21 bit in the IFSR2 register to “1” ($\overline{\text{INT}}_6$).⁽¹⁾
- To use the $\overline{\text{INT}}_7$ interrupt: Set the IFSR20 bit in the IFSR2 register to “1” ($\overline{\text{INT}}_7$).⁽¹⁾
- To use the $\overline{\text{INT}}_8$ interrupt: Set the IFSR22 bit in the IFSR2 register to “1” ($\overline{\text{INT}}_8$).⁽¹⁾

After modifying the IFSR16, IFSR17, IFSR20, IFSR21 and IFSR22 bits, set the corresponding IR bit to “0” (interrupt not requested) before enabling the interrupt.

NOTE:

1. $\overline{\text{INT}}_6$ to $\overline{\text{INT}}_8$ interrupts are only in the 128-pin version.

Figures 9.11 to 9.13 show the IFSR0, IFSR1 and IFSR2 registers.

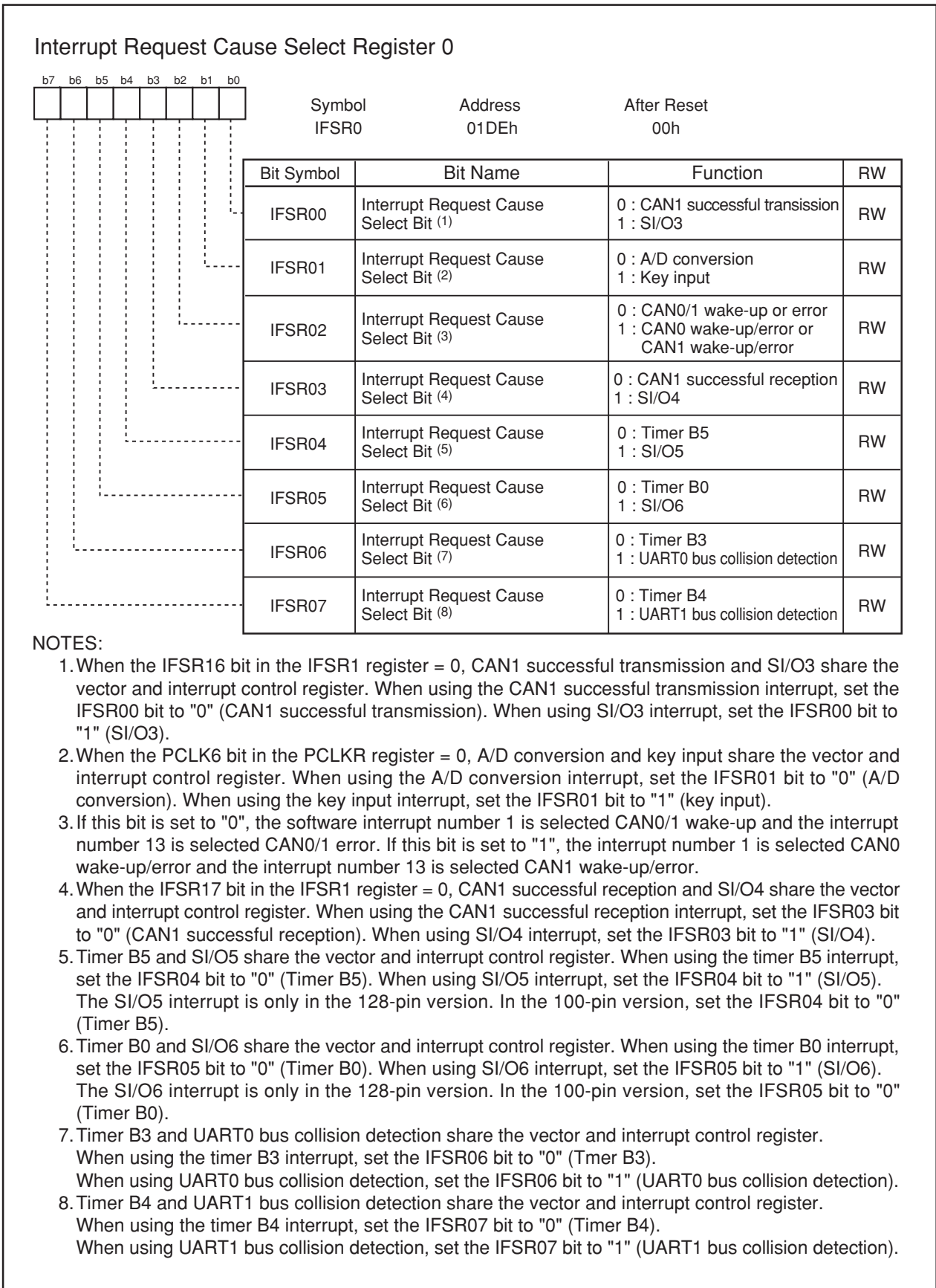


Figure 9.11 IFSR0 Register

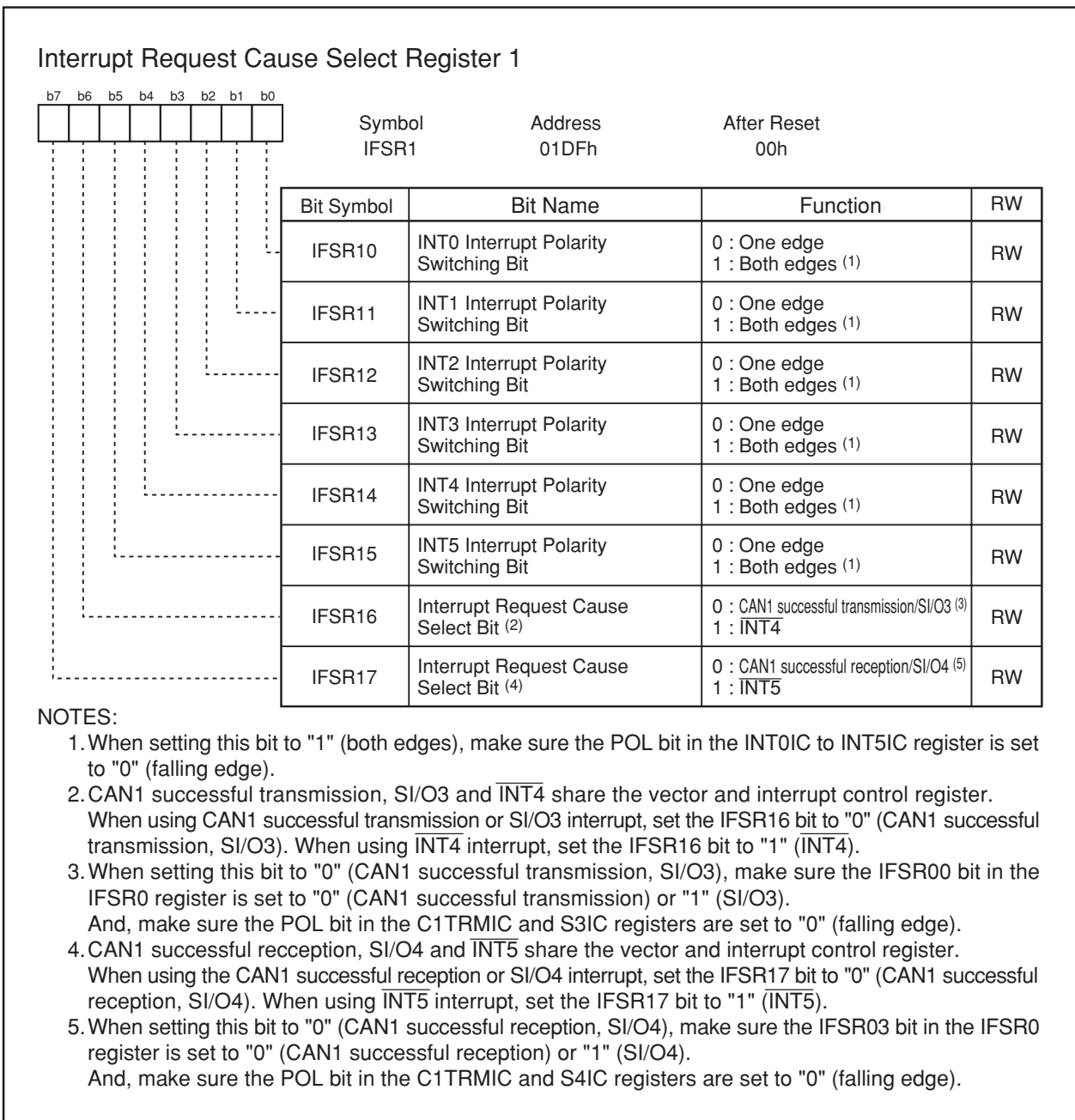


Figure 9.12 IFSR1 Register

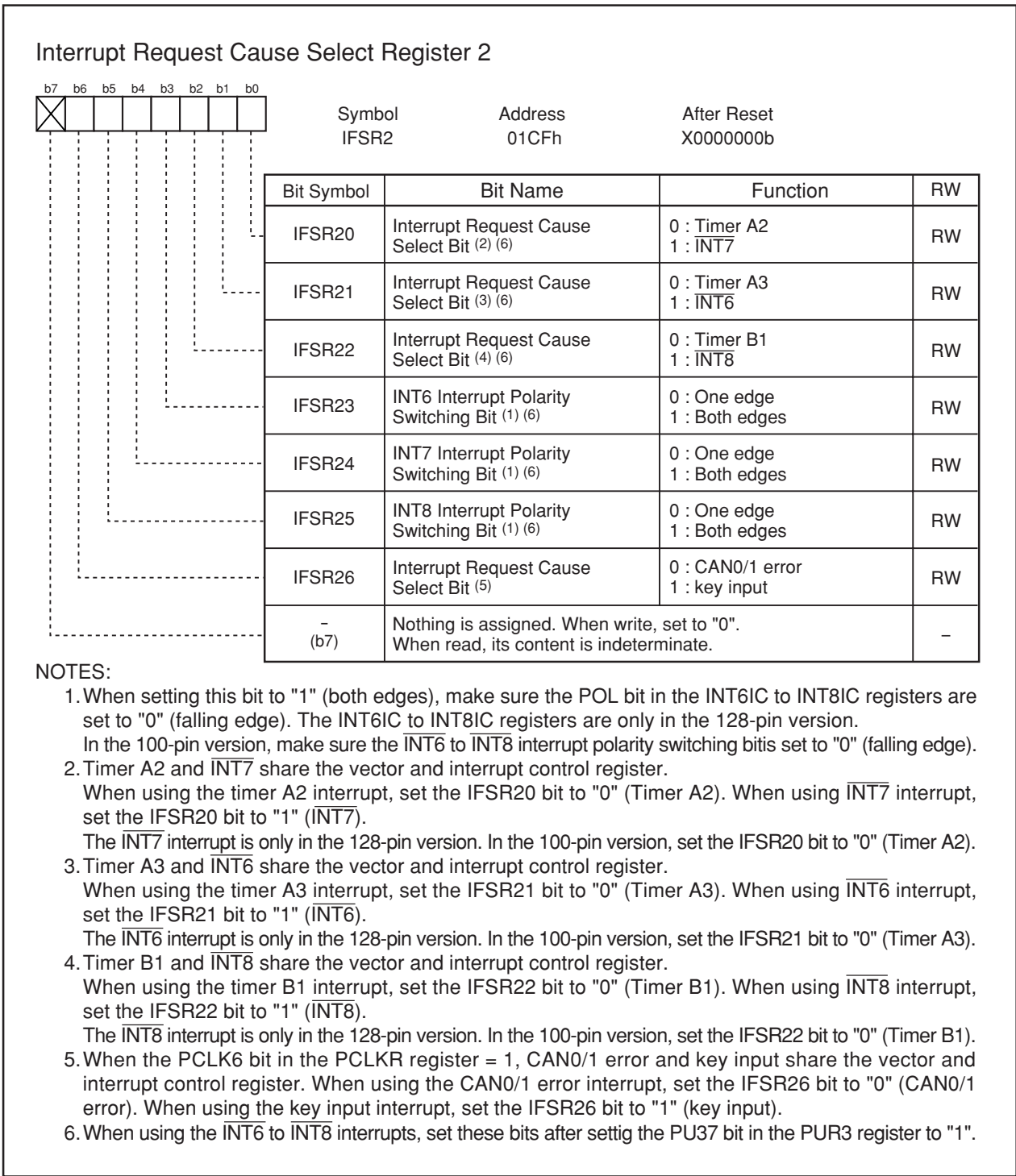


Figure 9.13 IFSR2 Register

9.7 NMI Interrupt

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

9.8 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 9.14 shows the block diagram of the key input interrupt. Note, however, that while input on any pin which has had the PD10_4 to PD10_7 bits set to "0" (input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

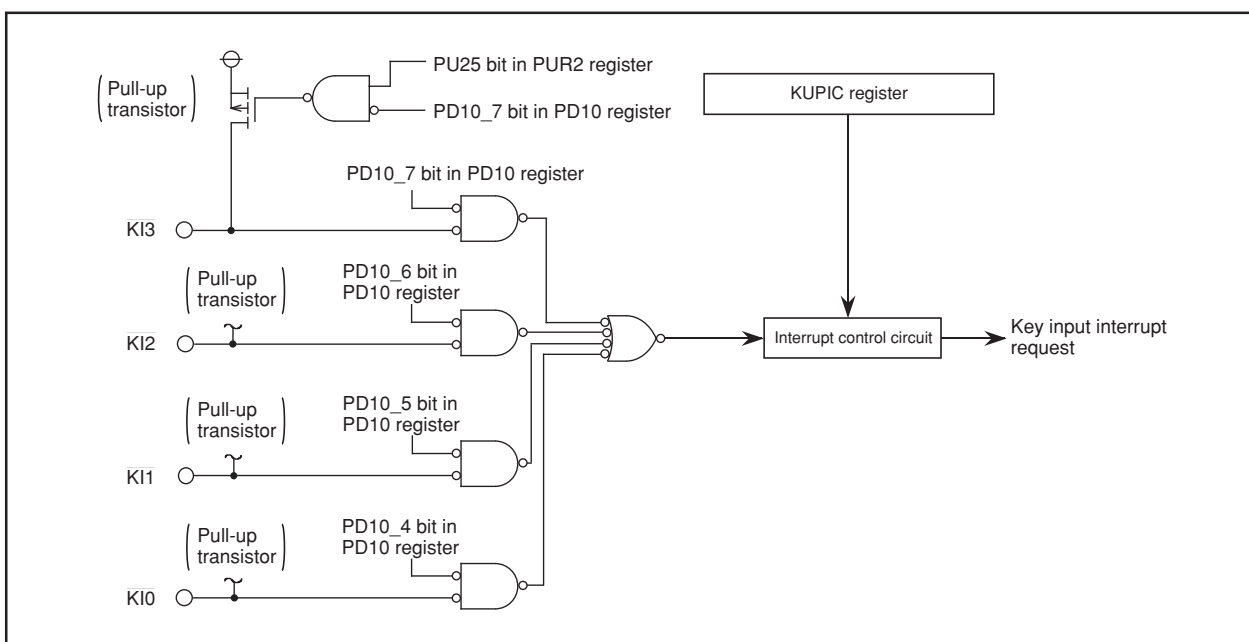


Figure 9.14 Key Input Interrupt Block Diagram

9.9 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 CiCTRL register (i = 0, 1). Figure 9.15 shows the block diagram of the CAN0/1 wake-up interrupt. Please note that the wake-up message will be lost.

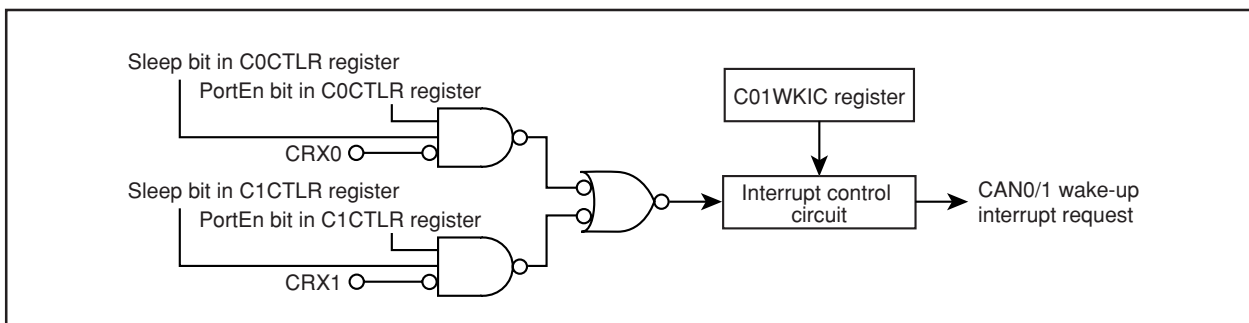


Figure 9.15 CAN0/1 Wake-up Interrupt Block Diagram

9.10 Address Match Interrupt

An address match interrupt request is generated immediately before executing the instruction at the address indicated by the RMAD_i register (i = 0 to 3). Set the start address of any instruction in the RMAD_i 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 **9.5.7 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 9.6 shows the value of the PC that is saved to the stack area when an address match interrupt request is accepted.

Table 9.7 shows the relationship between address match interrupt sources and associated registers.

Figure 9.16 shows the AIER, AIER2, and RMAD0 to RMAD3 registers.

Table 9.6 Value of PC That is Saved to Stack Area When Address Match Interrupt Request is Accepted

Instruction at Address Indicated by RMAD _i Register	Value of PC that is Saved to Stack Area
<ul style="list-style-type: none"> • 16-bit operation code • Instruction shown below among 8-bit operation code instructions <div style="font-family: monospace; font-size: small;"> 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 RMAD _i register + 2
Instructions other than the above	Address indicated by RMAD _i register + 1

Value of PC that is saved to stack area: Refer to **9.5.7 Saving Registers**.

Table 9.7 Relationship Between Address Match Interrupt Sources and Associated Registers

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

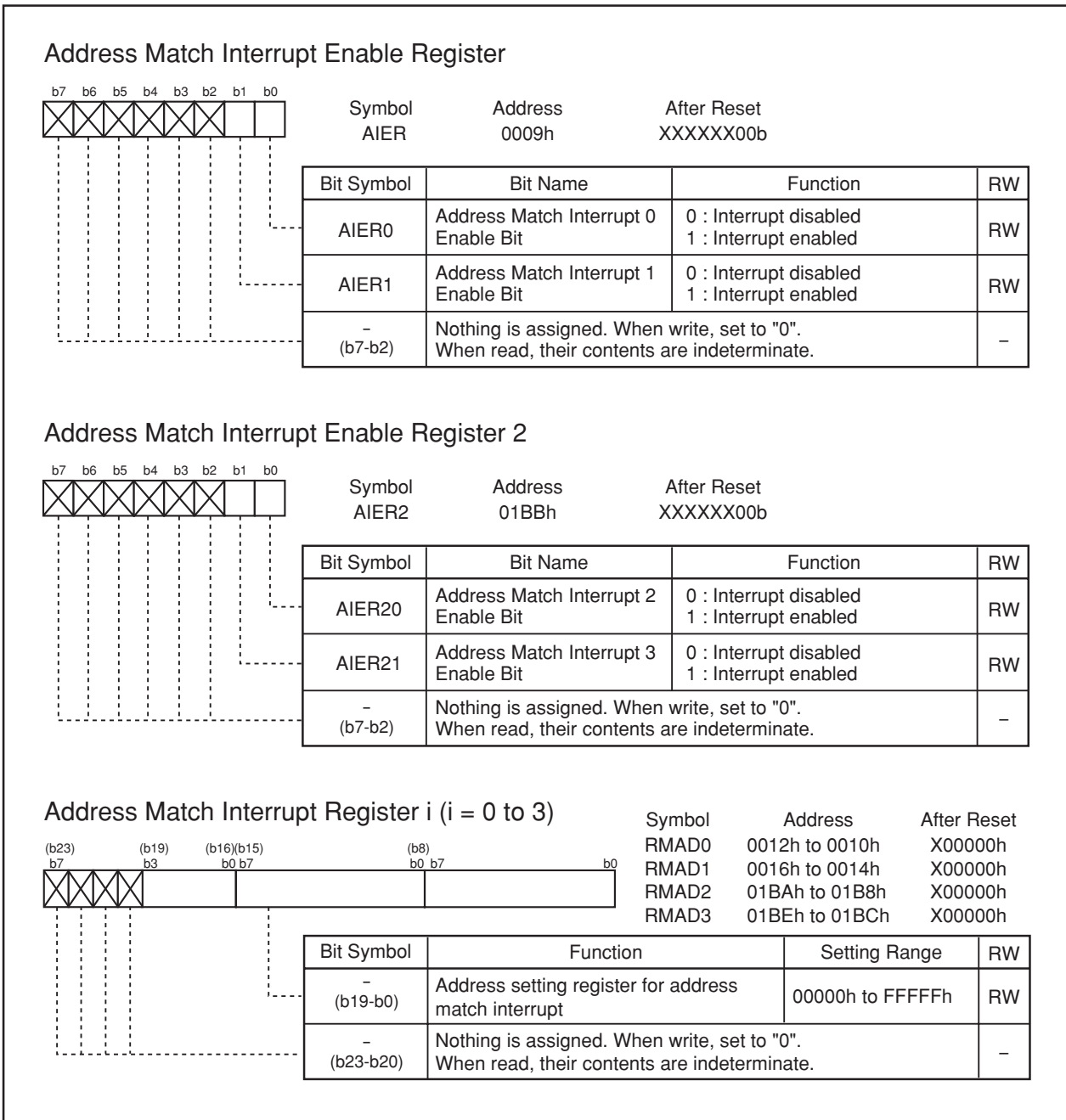


Figure 9.16 AIER Register, AIER2 Register and RMAD0 to RMAD3 Registers

10. Watchdog Timer

The watchdog timer is the function of detecting when the program is out of control. Therefore, we recommend using the watchdog timer to improve reliability of a system. The watchdog timer contains a 15-bit counter which counts down the clock derived by dividing the CPU clock using the prescaler. Whether to generate a watchdog timer interrupt request or apply a watchdog timer reset as an operation to be performed when the watchdog timer underflows after reaching the terminal count can be selected using the PM12 bit in the PM1 register. The PM12 bit can only be set to "1" (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 10.1 shows the block diagram of the watchdog timer. Figure 10.2 shows the watchdog timer-related registers.

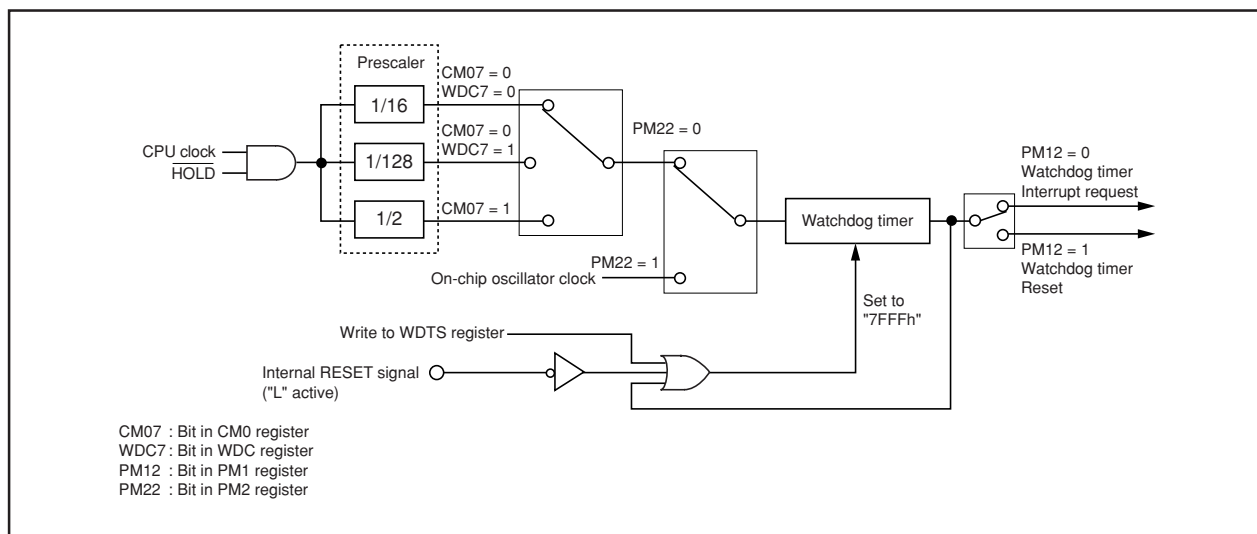


Figure 10.1 Watchdog Timer Block Diagram

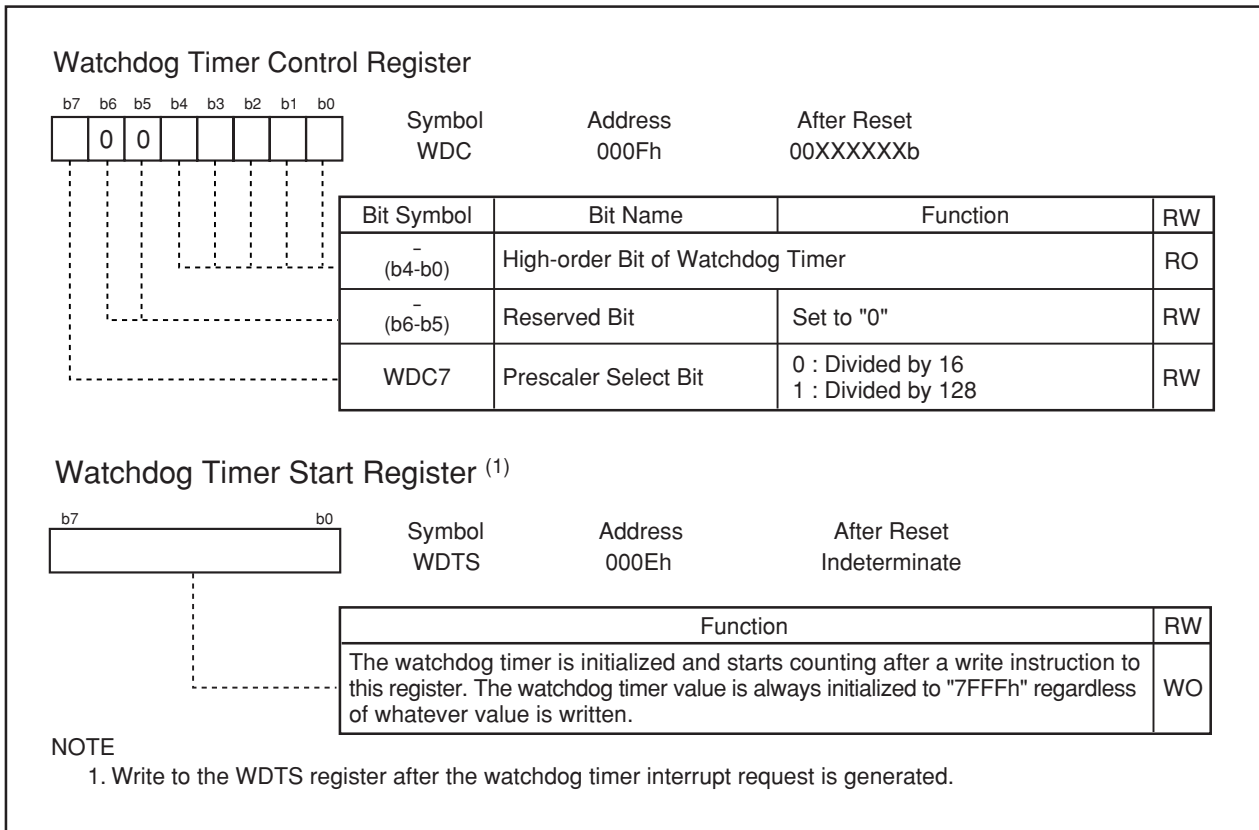


Figure 10.2 WDC Register and WDTS Register

10.1 Count Source Protective Mode

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

11. 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 11.1 shows the block diagram of the DMAC. Table 11.1 shows the DMAC specifications. Figures 11.2 to 11.4 show the DMAC related-registers.

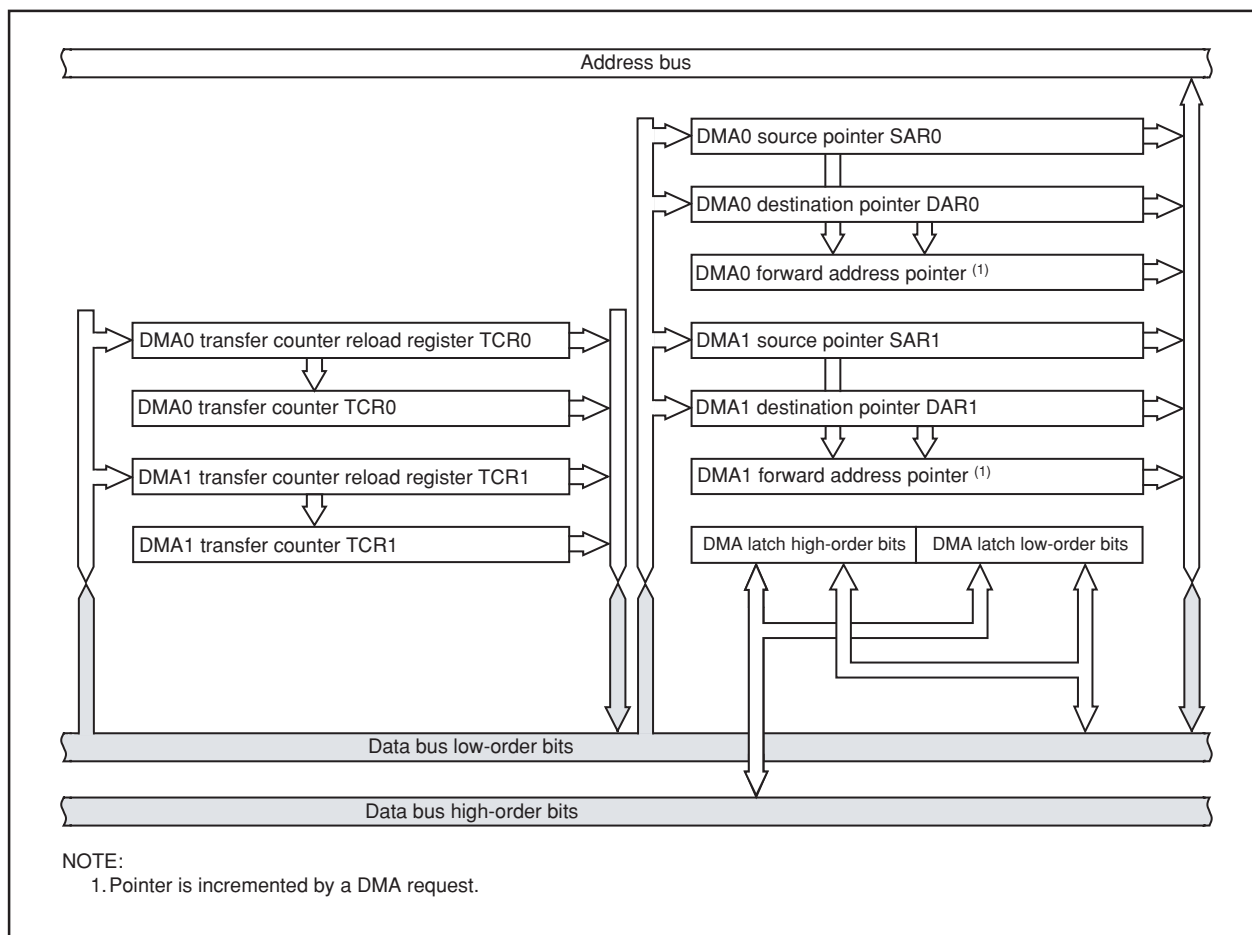


Figure 11.1 DMAC Block Diagram

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

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

Table 11.1 DMAC Specifications

Item		Specification
No. of Channels		2 (cycle steal method)
Transfer Memory Space		<ul style="list-style-type: none"> • From any address in the 1-Mbyte space to a fixed address • From a fixed address to any address in the 1-Mbyte space • From a fixed address to a fixed address
Maximum No. of Bytes Transferred		128 Kbytes (with 16-bit transfer) or 64 Kbytes (with 8-bit transfer)
DMA Request Factors ^{(1) (2)}		Falling edge of INT ₀ or INT ₁ Both edge of INT ₀ or INT ₁ 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, SI/O4 interrupt requests A/D conversion interrupt requests Software triggers
Channel Priority		DMA0 > DMA1 (DMA0 takes precedence)
Transfer Unit		8 bits or 16 bits
Transfer Address Direction		forward or fixed (The source and destination addresses cannot both be in the forward direction.)
Transfer Mode	Single Transfer	Transfer is completed when the DMA _i transfer counter underflows after reaching the terminal count.
	Repeat Transfer	When the DMA _i transfer counter underflows, it is reloaded with the value of the DMA _i transfer counter reload register and a DMA transfer is continued with it.
DMA Interrupt Request Generation Timing		When the DMA _i transfer counter underflowed
DMA Start Up		Data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMA _i CON register = 1 (enabled).
DMA Shutdown	Single Transfer	<ul style="list-style-type: none"> • When the DMAE bit is set to "0" (disabled) • After the DMA_i transfer counter underflows
	Repeat Transfer	When the DMAE bit is set to "0" (disabled)
Reload Timing for Forward Address Pointer and Transfer Counter		When a data transfer is started after setting the DMAE bit to "1" (enabled), the forward address pointer is reloaded with the value of the SAR _i or the DAR _i pointer whichever is specified to be in the forward direction and the DMA _i transfer counter is reloaded with the value of the DMA _i transfer counter reload register.

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.

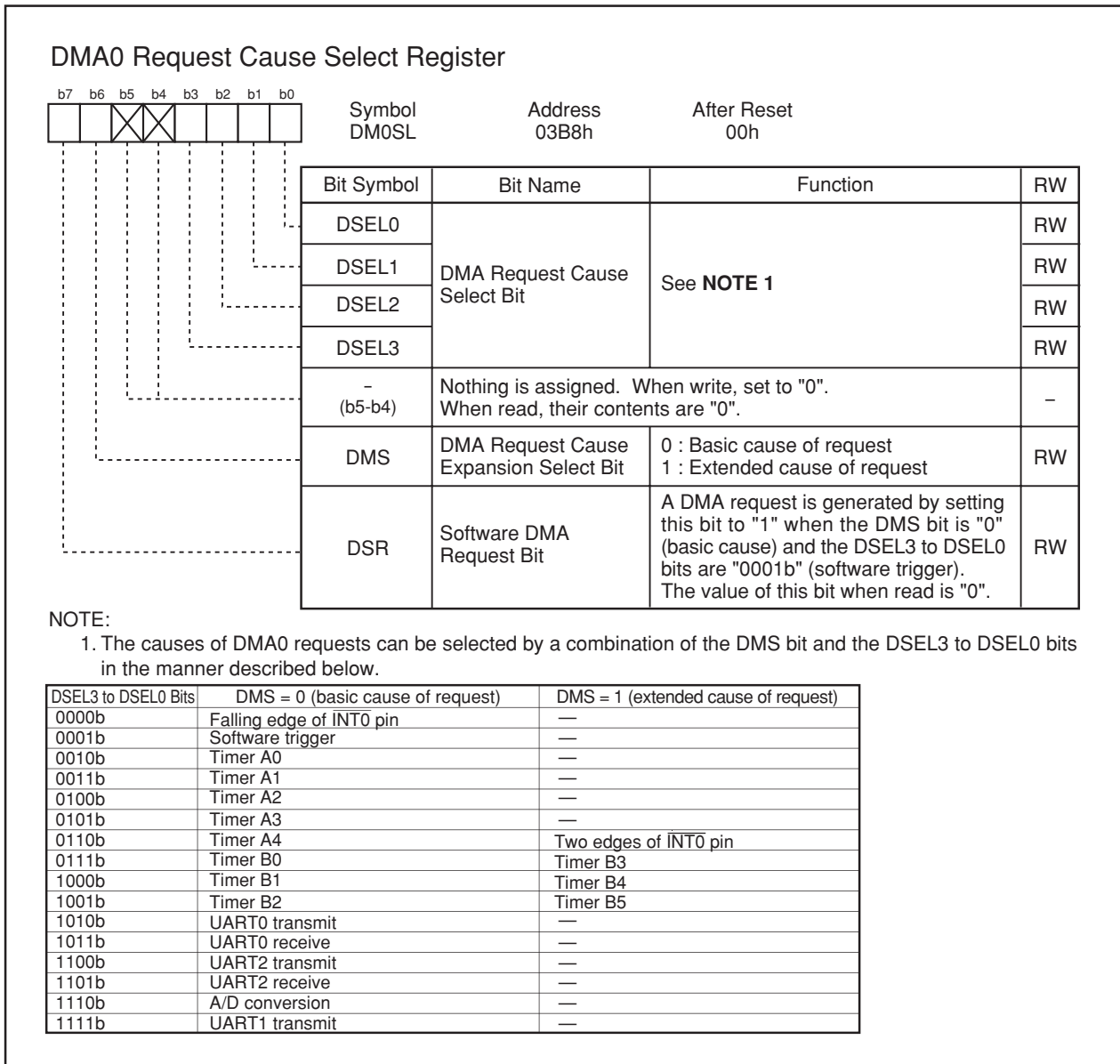


Figure 11.2 DM0SL Register

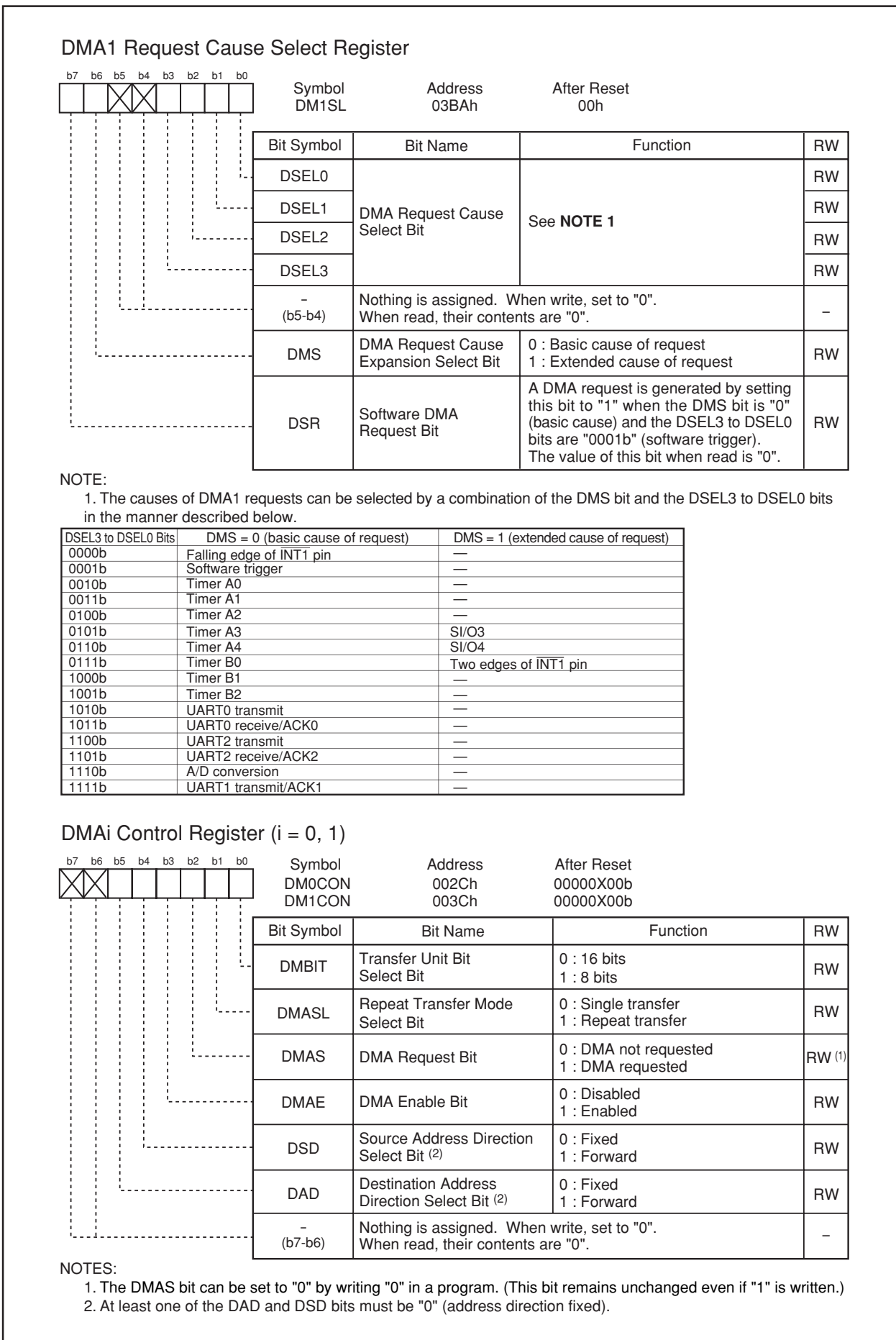


Figure 11.3 DM1SL Register, DM0CON and DM1CON Registers

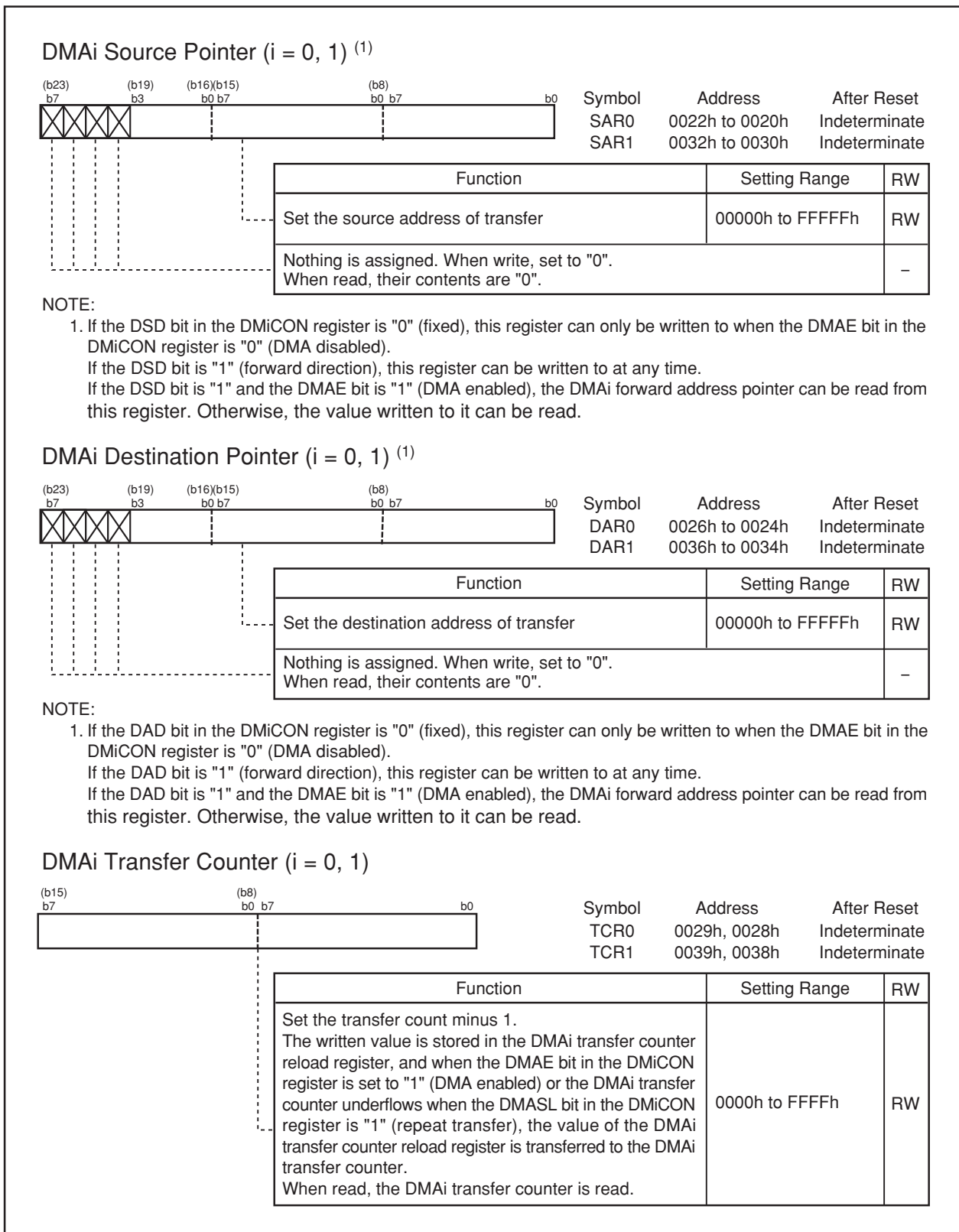


Figure 11.4 SAR0 and SAR1 Registers, DAR0 and DAR1 Registers, TCR0 and TCR1 Registers

11.1 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. The bus cycle itself is extended by a software wait.

11.1.1 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.

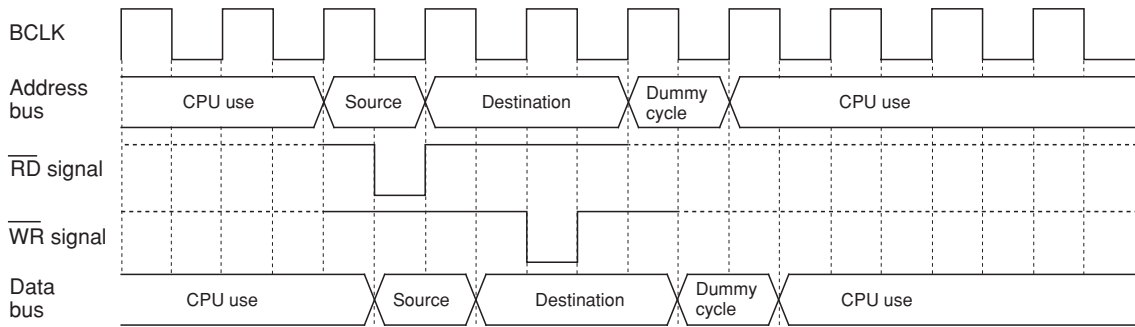
Similarly, if the transfer unit and data bus both are 16 bits and the destination address of transfer begins with an odd address, the destination write cycle consists of one more bus cycle than when the destination address of transfer begins with an even address.

11.1.2 Effect of Software Wait

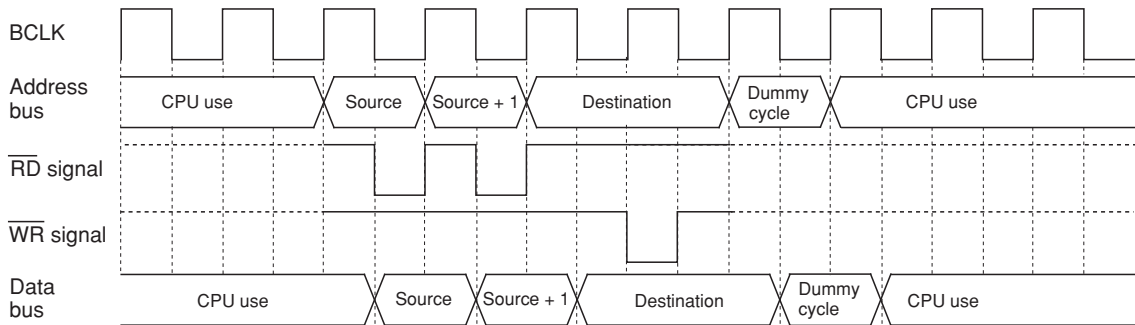
For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

Figure 11.5 shows the example of the cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16-bit unit using an 8-bit bus ((2) on Figure 11.5), two source read bus cycles and two destination write bus cycles are required.

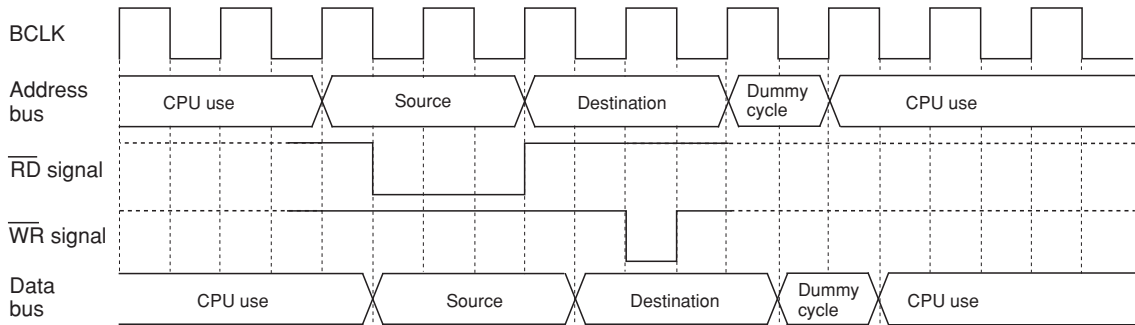
(1) When the transfer unit is 8 or 16 bits and the source of transfer is an even address



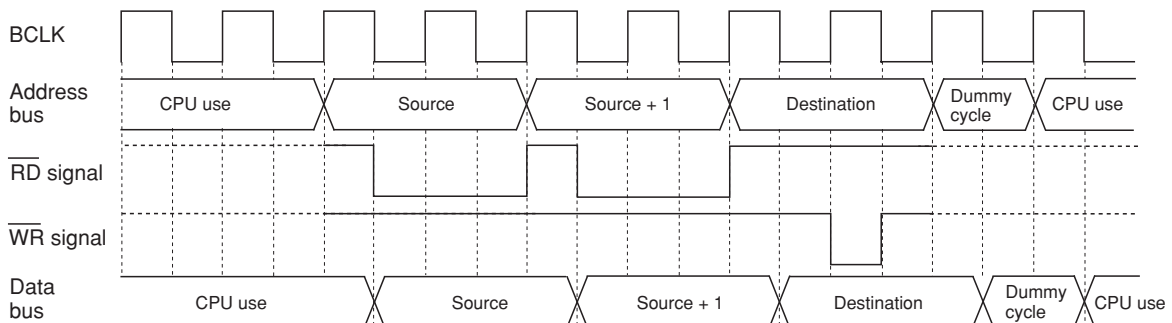
(2) When the transfer unit is 16 bits and the source address of transfer is an odd address, or when the transfer unit is 16 bits and an 8-bit bus is used



(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 11.5 Transfer Cycles for Source Read

11.2 DMA Transfer Cycles

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

Table 11.2 shows the number of DMA transfer cycles. Table 11.3 shows the coefficient j, k.

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

No. of transfer cycles per transfer unit = No. of read cycles \times j + No. of write cycles \times k

Table 11.2 DMA Transfer Cycles

Transfer Unit	Access Address	No. of Read Cycles	No. of Write Cycles
8-bit Transfer (DMBIT = 1)	Even	1	1
	Odd	1	1
16-bit Transfer (DMBIT = 0)	Even	1	1
	Odd	2	2

Table 11.3 Coefficient j, k

	Internal ROM, RAM		SFR	
	No Wait	With Wait	1 Wait ⁽¹⁾	2 Waits ⁽¹⁾
j	1	2	2	3
k	1	2	2	3

NOTE:

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

11.3 DMA Enable

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

- (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.

11.4 DMA Request

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

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

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

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

Table 11.4 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"> • Immediately before a data transfer starts • When set by writing “0” in a program
Peripheral Function	When the interrupt control register for the peripheral function that is selected by the DSEL3 to DSEL0 and DMS bits in the DMiSL register has its IR bit set to “1”.	

i = 0, 1

11.5 Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of 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 11.6 shows an example of DMA transfer effected by external factors.

In Figure 11.6, 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 11.6, 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.

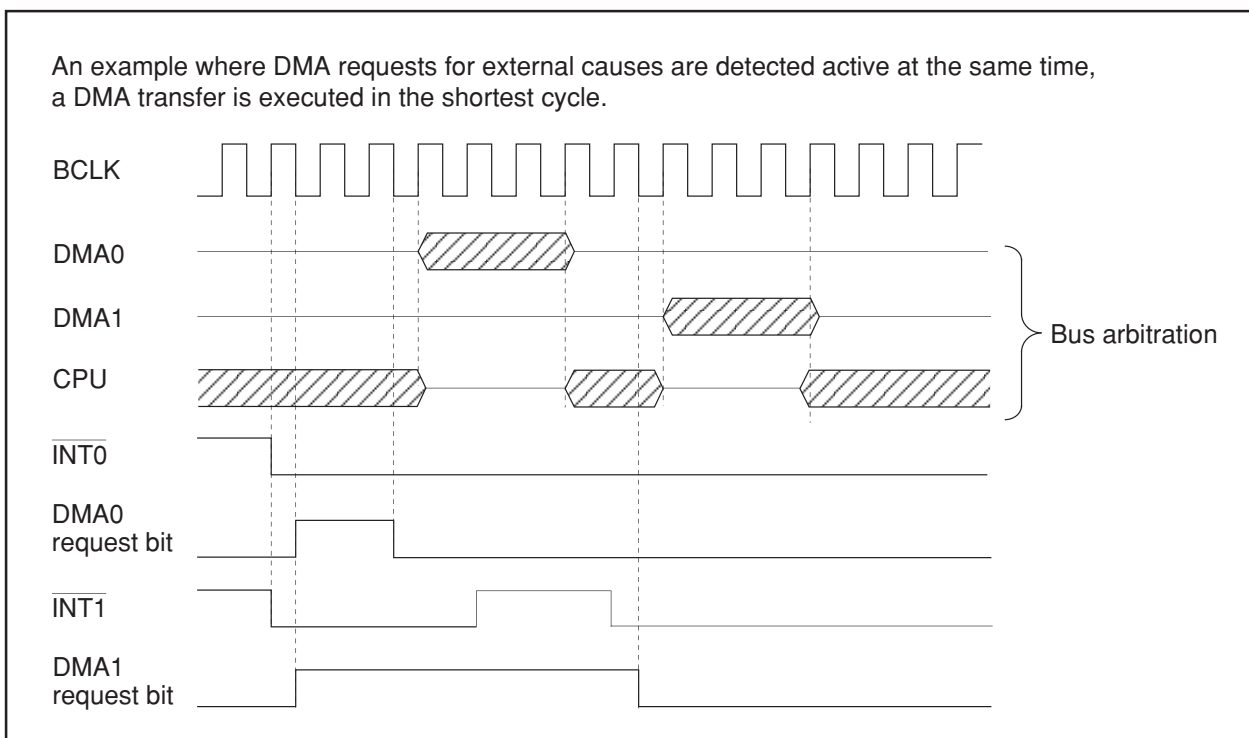


Figure 11.6 DMA Transfer by External Factors

12. 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 12.1 and 12.2 show block diagrams of Timer A and Timer B configuration, respectively.

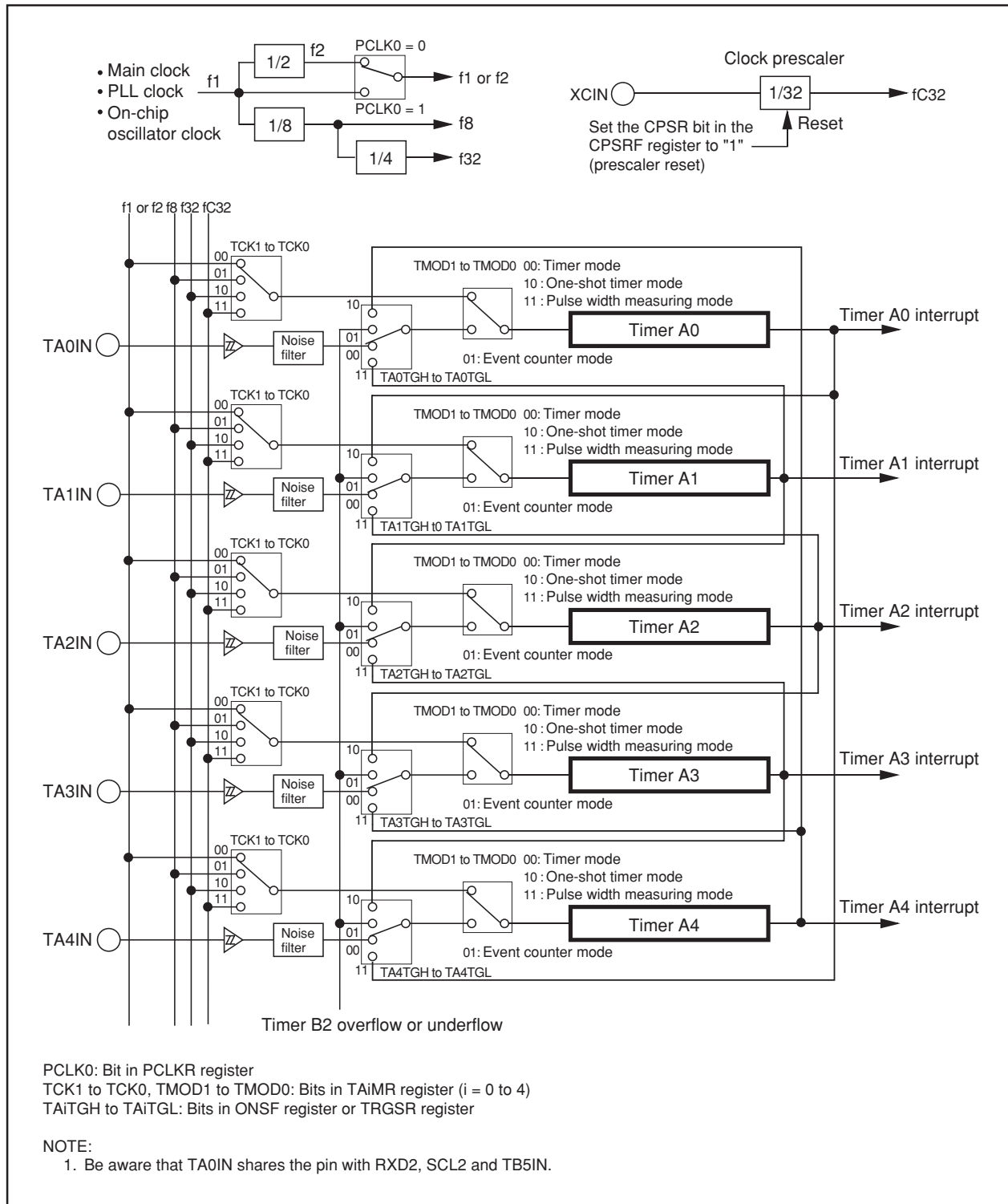


Figure 12.1 Timer A Configuration

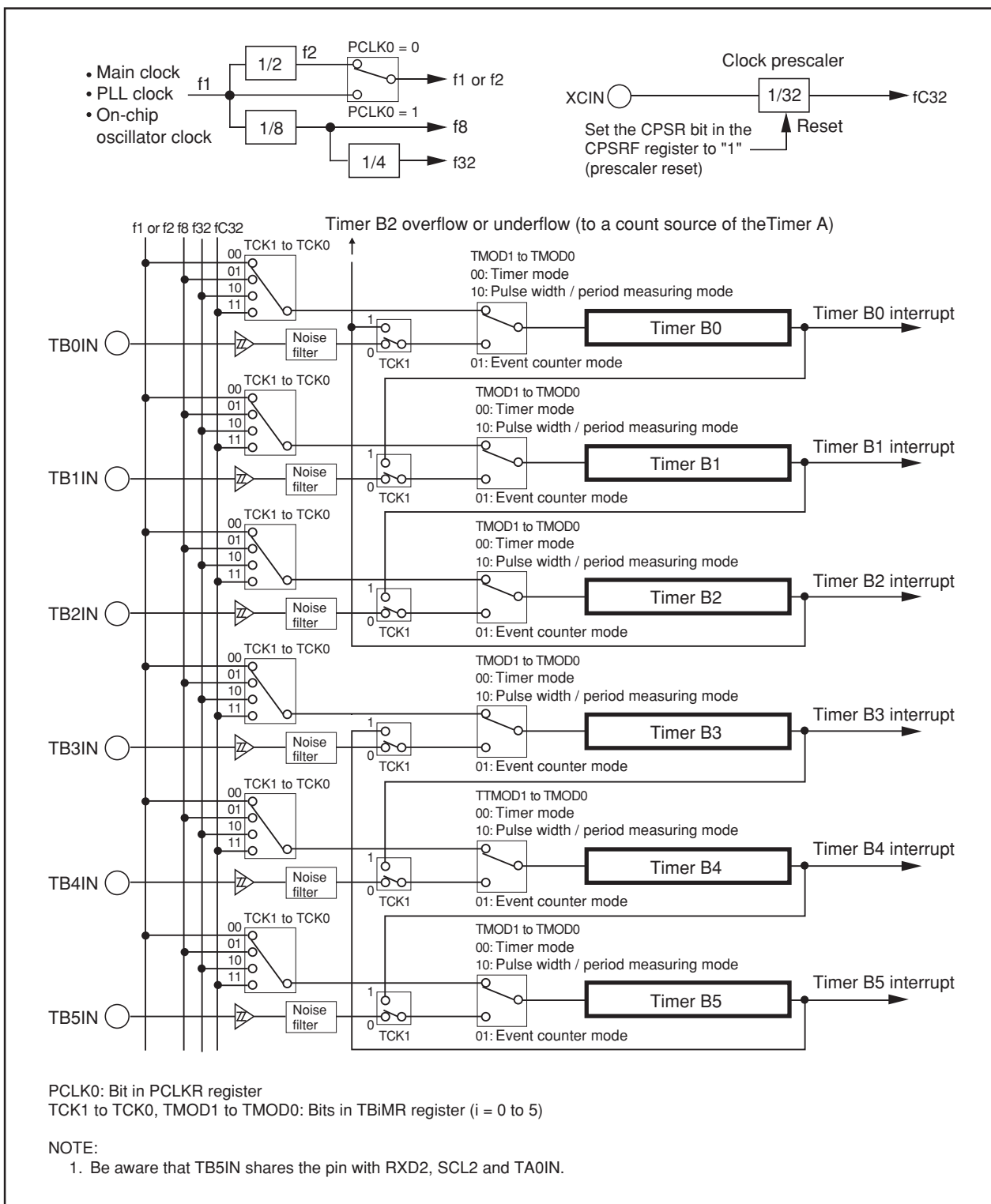


Figure 12.2 Timer B Configuration

12.1 Timer A

Figure 12.3 shows a block diagram of the timer A. Figures 12.4 to 12.6 show 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_{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 mode: The timer outputs pulses in a given width successively.

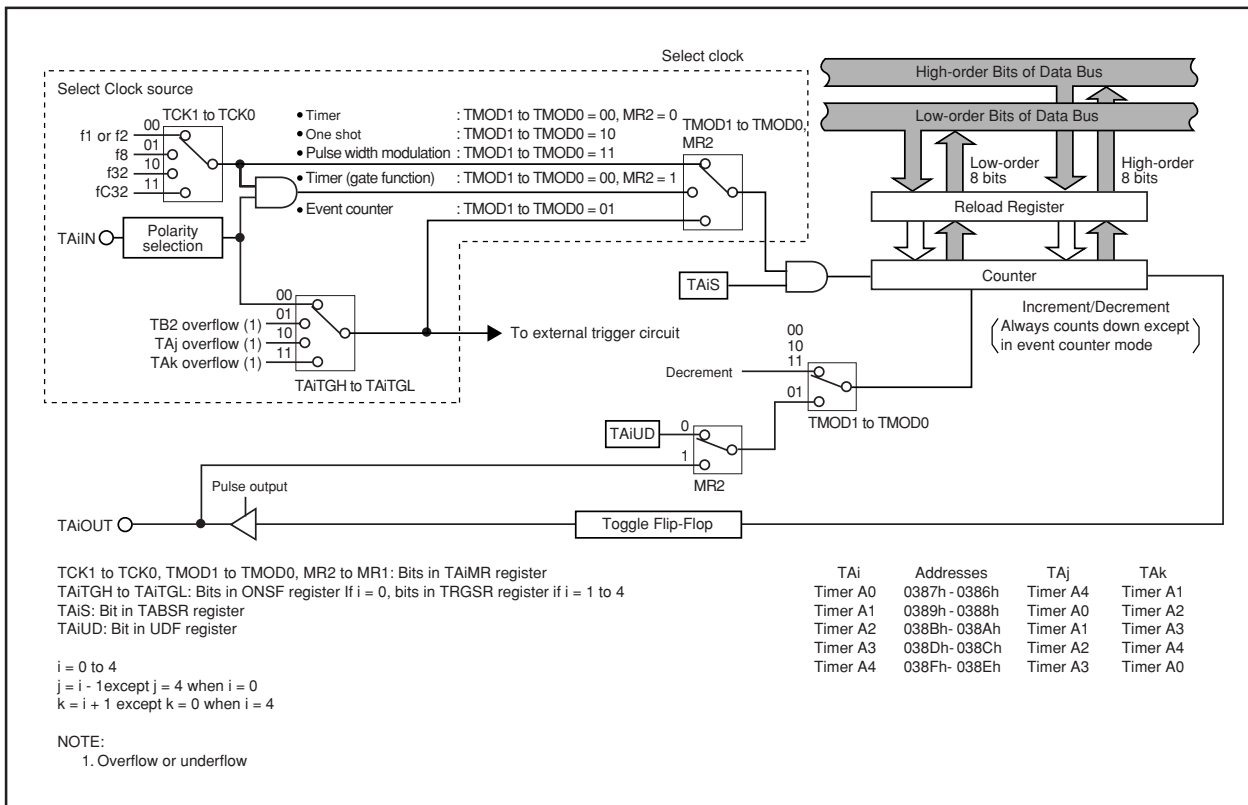


Figure 12.3 Timer A Block Diagram

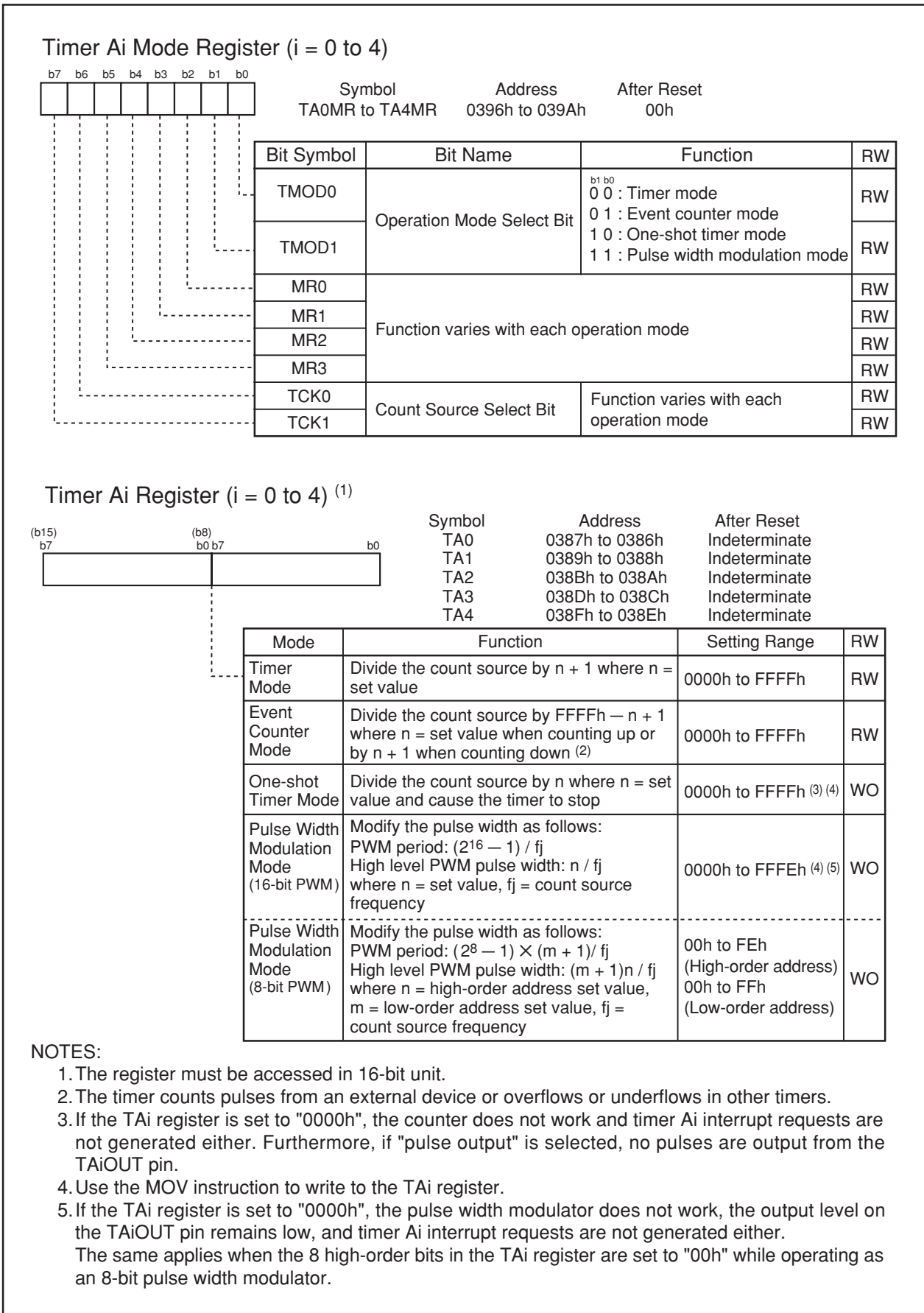


Figure 12.4 TA0MR to TA4MR Registers and TA0 to TA4 Registers

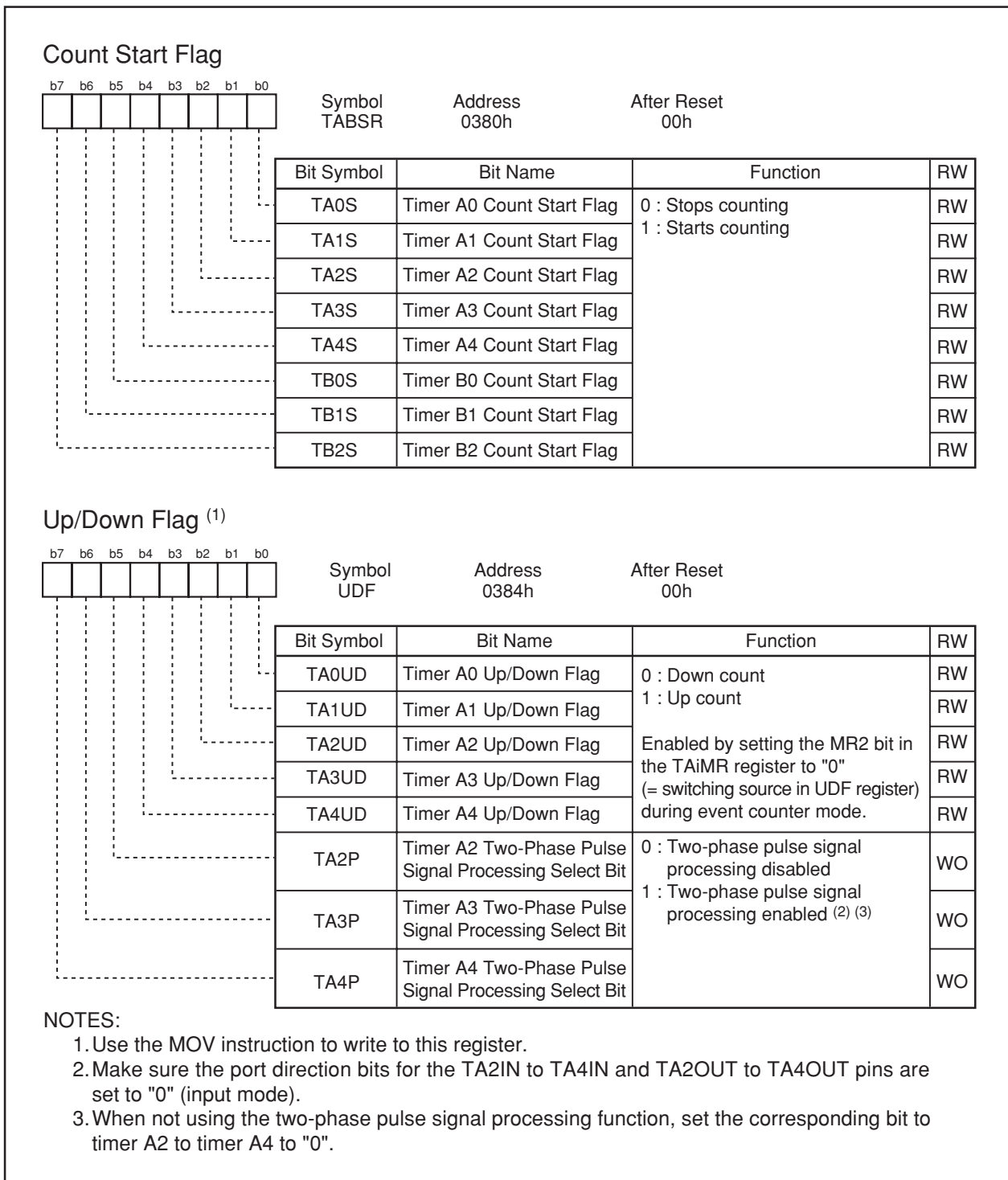


Figure 12.5 TABSR Register and UDF Register

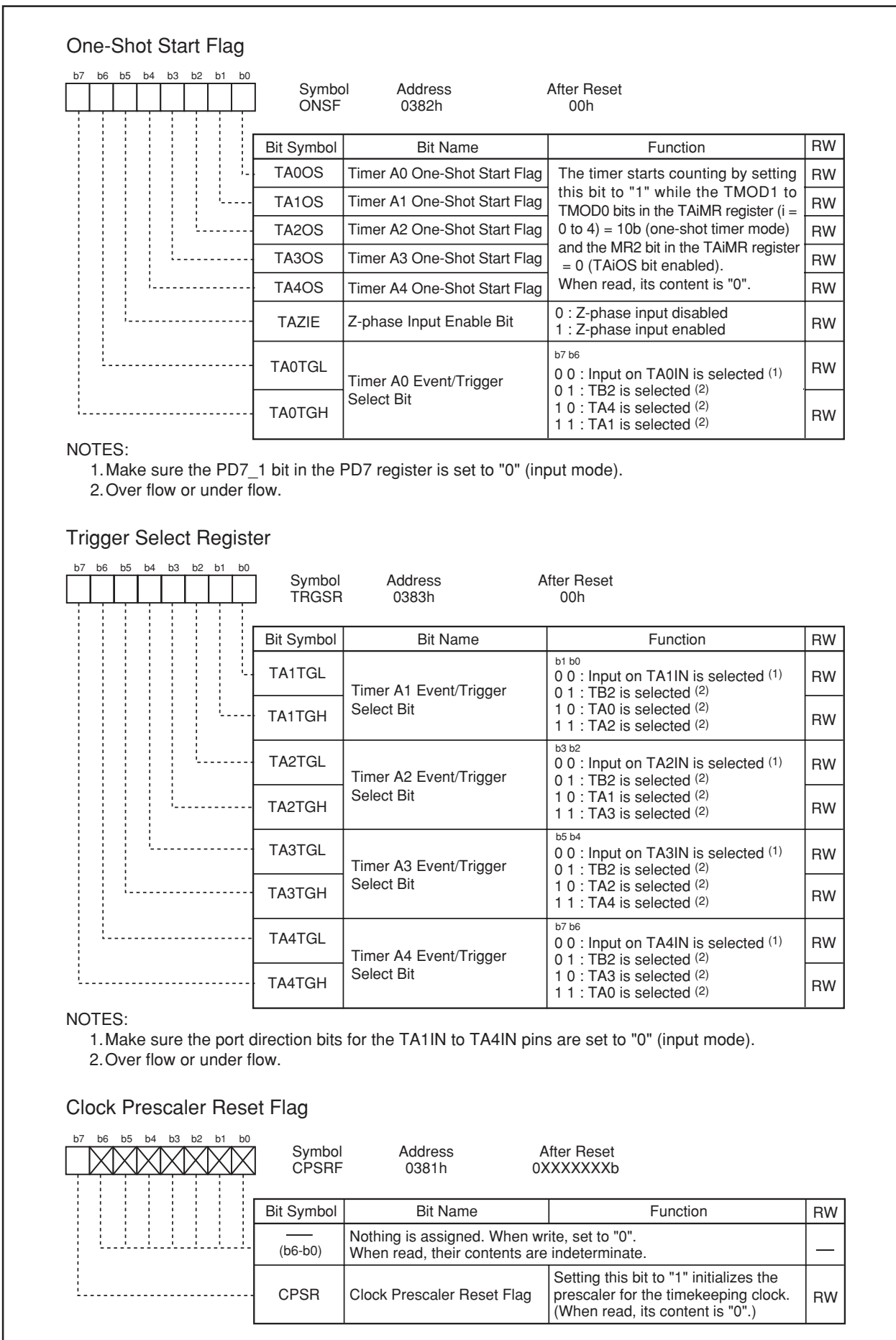


Figure 12.6 ONSF Register, TRGSR Register and CPSRF Register

12.1.1 Timer Mode

In timer mode, the timer counts a count source generated internally. Table 12.1 lists specifications in timer mode. Figure 12.7 shows TAIiMR register in timer mode.

Table 12.1 Specifications in Timer Mode

Item	Specification
Count Source	f1, f2, f8, f32, fC32
Count Operation	<ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the reload register contents and continues counting
Divide Ratio	1/(n+1) n: set value of the TAI register 0000h to FFFFh
Count Start Condition	Set the 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"> When not counting and until the 1st count source is input after counting start Value written to the TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)
Select Function	<ul style="list-style-type: none"> Gate function Counting can be started and stopped by an input signal to TAIiN pin Pulse output function Whenever the timer underflows, the output polarity of TAIiOUT pin is inverted. When TAIiS bit is set to "0" (stop counting), the pin outputs a low.

i = 0 to 4

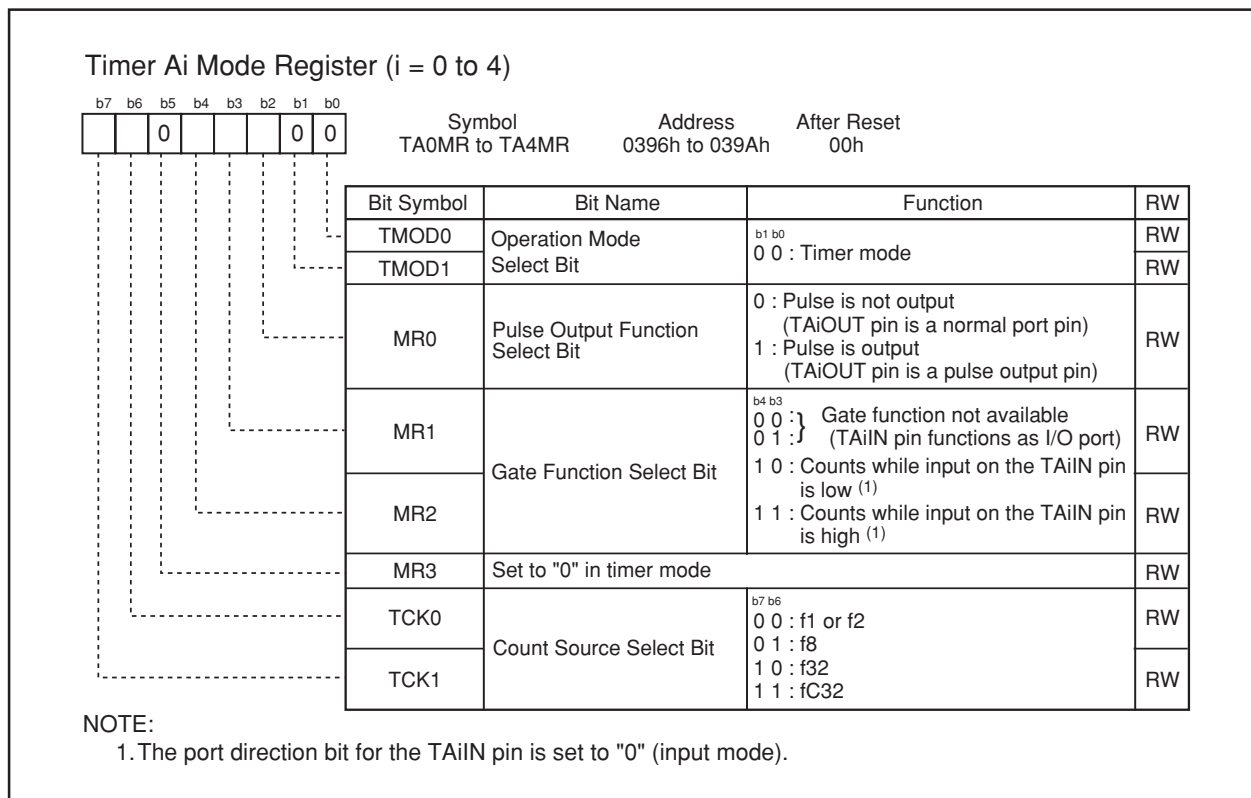


Figure 12.7 TA0MR to TA4MR Registers in Timer Mode

12.1.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3 and A4 can count two-phase external signals. Table 12.2 lists specifications in event counter mode (when not processing two-phase pulse signal). Figure 12.8 shows TAI_{MR} register in event counter mode (when not processing two-phase pulse signal). Table 12.3 lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4). Figure 12.9 shows TA2MR to TA4MR registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

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

Item	Specification
Count Source	<ul style="list-style-type: none"> External signals input to TAI_{IN} pin (effective edge can be selected in program) Timer B2 overflows or underflows, Timer A_j overflows or underflows, Timer A_k overflows or underflows
Count Operation	<ul style="list-style-type: none"> Up-count or down-count can be selected by external signal or program When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divided Ratio	1/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count n : set value of the TAI register 0000h to FFFFh
Count Start Condition	Set the TAI _S bit in the TABSR register to "1" (start counting)
Count Stop Condition	Set the TAI _S bit to "0" (stop counting)
Interrupt Request Generation Timing	Timer overflow or underflow
TAI _{IN} Pin Function	I/O port or count source input
TAI _{OUT} Pin Function	I/O port, pulse output, or up/down-count select input
Read from Timer	Count value can be read by reading the TAI register
Write to Timer	<ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to the TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)
Select Function	<ul style="list-style-type: none"> Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it Pulse output function Whenever the timer underflows or underflows, the output polarity of TAI_{OUT} pin is inverted. When TAI_S bit is set to "0" (stop counting), the pin outputs a low.

i = 0 to 4

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

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

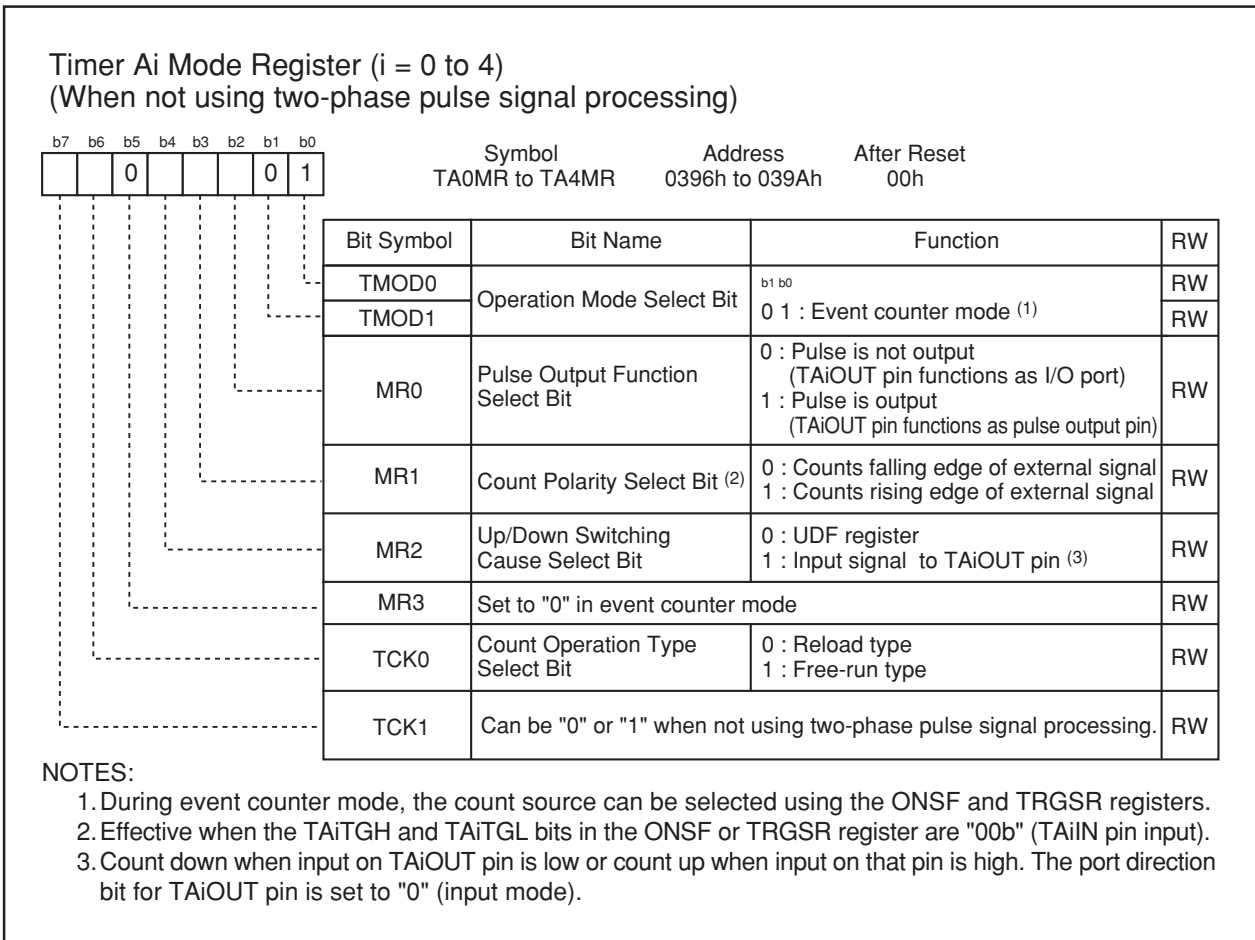


Figure 12.8 TA0MR to TA4MR Registers in Event Counter Mode (when not using two-phase pulse signal processing)

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

Item	Specification
Count Source	• Two-phase pulse signals input to TAIiN or TAIiOUT pins
Count Operation	• Up-count or down-count can be selected by two-phase pulse signal • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divide Ratio	1/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count n : set value of the TAI register 0000h to FFFFh
Count Start Condition	Set the TAIiS bit in the TABSR register to "1" (start counting)
Count Stop Condition	Set the TAIiS bit to "0" (stop counting)
Interrupt Request Generation Timing	Timer overflow or underflow
TAiIN Pin Function	Two-phase pulse input
TAiOUT Pin Function	Two-phase pulse input
Read from Timer	Count value can be read by reading the TAI register
Write to Timer	• When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAI register is written to reload register (Transferred to counter when reloaded next)
Select Function ⁽¹⁾	<ul style="list-style-type: none"> • Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on TAJiN pin when input signals on TAJiOUT pin is "H". <ul style="list-style-type: none"> • Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAKiN pin goes "H" when the input signal on TAKiOUT pin is "H", the timer counts up rising and falling edges on TAKiOUT and TAKiN pins. If the phase relationship is such that TAKiN pin goes "L" when the input signal on TAKiOUT pin is "H", the timer counts down rising and falling edges on TAKiOUT and TAKiN pins. <ul style="list-style-type: none"> • Counter initialization by Z-phase input (timer A3) The timer count value is initialized to "0" by Z-phase input.

i = 2 to 4

j = 2, 3

k = 3, 4

NOTE:

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

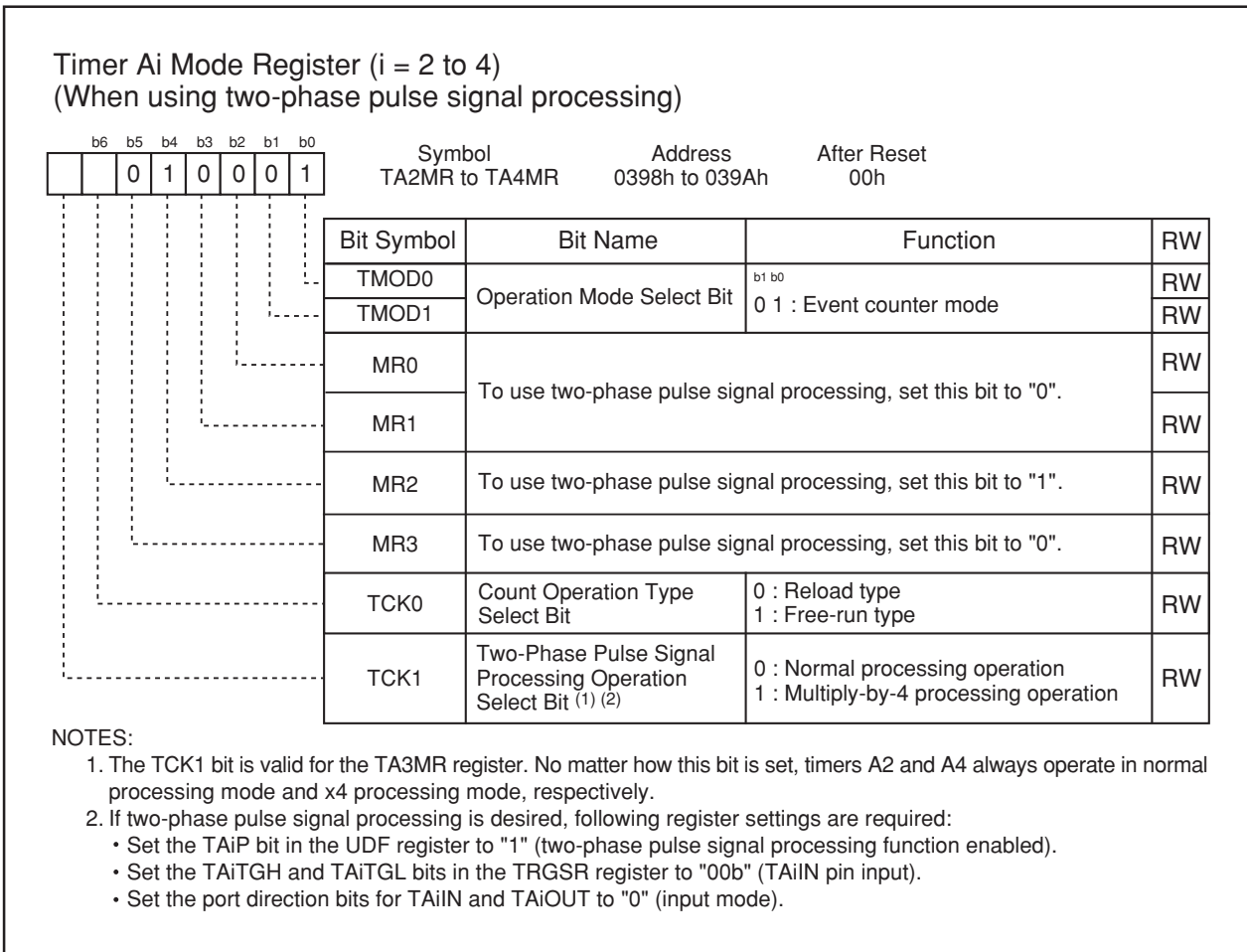


Figure 12.9 TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2, A3 or A4)

12.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing

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

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

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

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

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

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

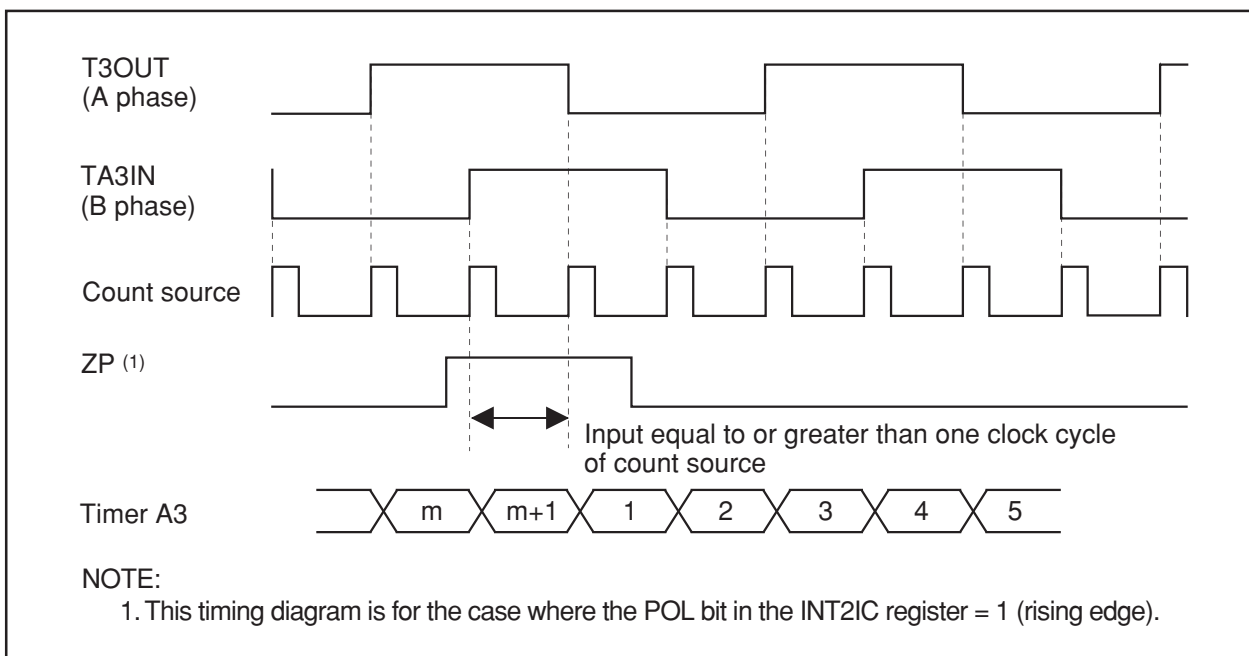


Figure 12.10 Two-phase Pulse (A phase and B phase) and Z Phase

12.1.3 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 12.4 lists specifications in one-shot timer mode. Figure 12.11 shows the TAIMR register in the one-shot timer mode.

Table 12.4 Specifications in One-shot Timer Mode

Item	Specification
Count Source	f1, f2, f8, f32, fC32
Count Operation	<ul style="list-style-type: none"> • Down-count • When the counter reaches 0000h, it stops counting after reloading a new value • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide Ratio	1/n n : set value of the TAI register 0000h to FFFFh However, the counter does not work if the divide-by-n value is set to 0000h.
Count Start Condition	The TAI _S bit in the TABSR register = 1 (start counting) and one of the following triggers occurs. <ul style="list-style-type: none"> • External trigger input from the TAI_{IN} pin • Timer B2 overflow or underflow, Timer A_j overflow or underflow, Timer A_k overflow or underflow • The TAI_{OS} bit in the ONSF register is set to “1” (timer starts)
Count Stop Condition	<ul style="list-style-type: none"> • When the counter is reloaded after reaching “0000h” • TAI_S bit is set to “0” (stop counting)
Interrupt Request Generation Timing	When the counter reaches “0000h”
TAI _{IN} Pin Function	I/O port or trigger input
TAI _{OUT} 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"> • When not counting and until the 1st count source is input after counting start Value written to the TAI register is written to both reload register and counter • When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)
Select Function	<ul style="list-style-type: none"> • Pulse output function The timer outputs a low when not counting and a high when counting.

i = 0 to 4

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

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

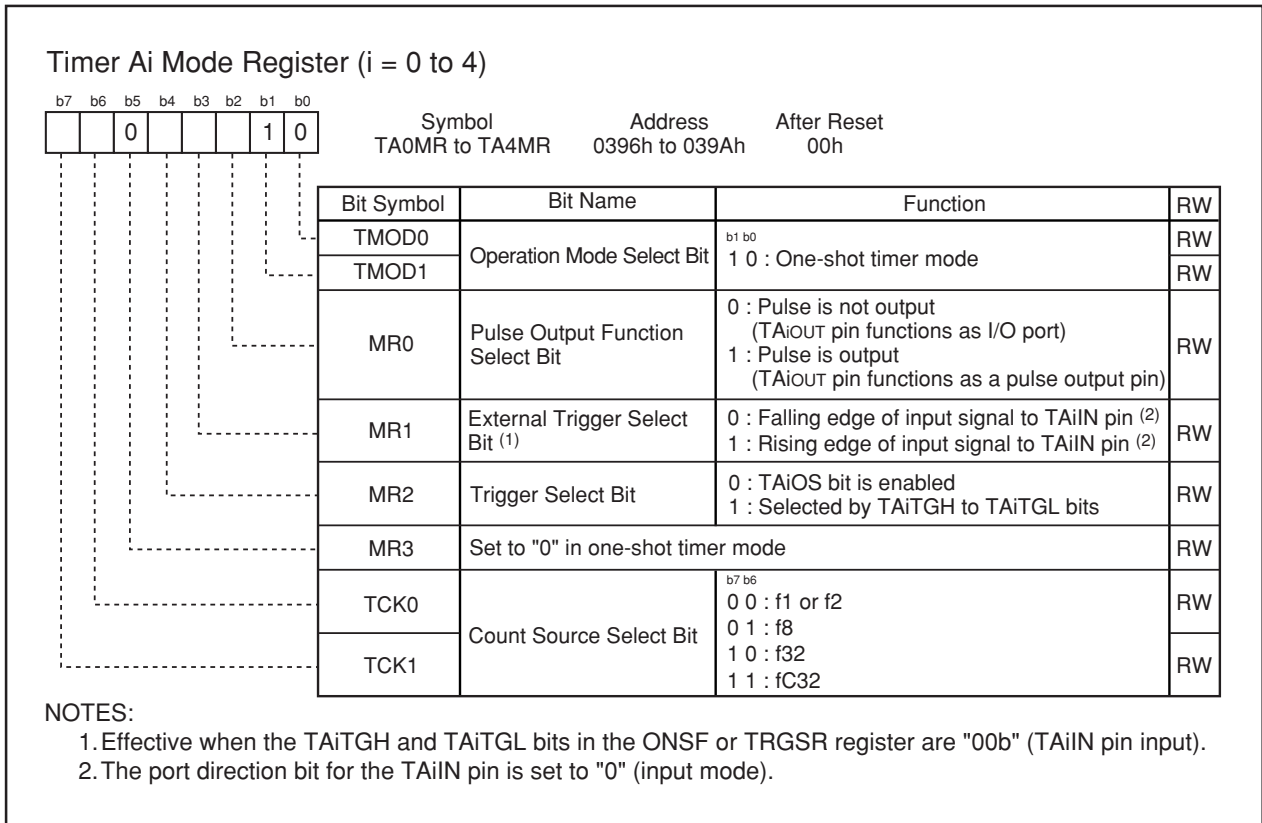


Figure 12.11 TAiMR Register in One-shot Timer Mode

12.1.4 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 12.5 lists specifications in pulse width modulation mode. Figure 12.12 shows TAI_iMR register in pulse width modulation mode.

Figures 12.13 and 12.14 show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates, respectively.

Table 12.5 Specifications in Pulse Width Modulation Mode

Item	Specification
Count Source	f1, f2, f8, f32, fC32
Count Operation	<ul style="list-style-type: none"> Down-count (operating as an 8-bit or a 16-bit pulse width modulator) The timer reloads a new value at a rising edge of PWM pulse and continues counting The timer is not affected by a trigger that occurs during counting
16-bit PWM	<ul style="list-style-type: none"> High level width n / f_j n : set value of the TAI register Cycle time $(2^{16}-1) / f_j$ fixed f_j : count source frequency (f1, f2, f8, f32, fC32)
8-bit PWM	<ul style="list-style-type: none"> High level width $n \times (m+1) / f_j$ n : set value of the TAI register high-order address Cycle time $(2^8-1) \times (m+1) / f_j$ m : set value of the TAI register low-order address
Count Start Condition	<ul style="list-style-type: none"> The TAI_S bit in the TABSR register is set to "1" (start counting) The TAI_S bit = 1 and external trigger input from the TAI_iIN pin The TAI_S bit = 1 and one of the following external triggers occurs Timer B2 overflow or underflow, Timer A_j overflow or underflow, Timer A_k overflow or underflow
Count Stop Condition	The TAI _S bit is set to "0" (stop counting)
Interrupt Request Generation Timing	On the falling edge of the PWM pulse
TAI _i IN Pin Function	I/O port or trigger input
TAI _i OUT 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"> When not counting and until the 1st count source is input after counting start Value written to the TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to the TAI register is written to only reload register (Transferred to counter when reloaded next)

$i = 0$ to 4

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

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

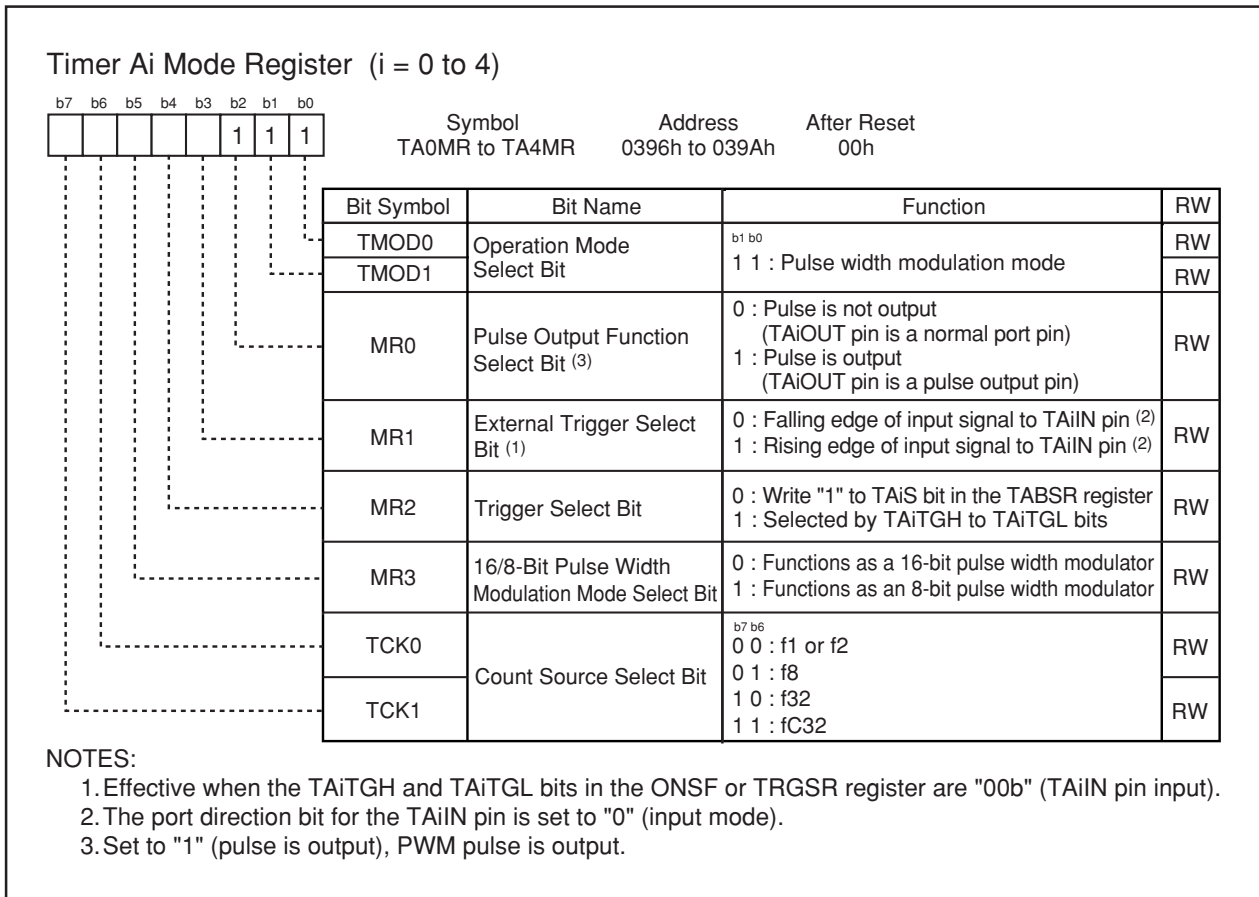


Figure 12.12 TA0MR to TA4MR Registers in Pulse Width Modulation Mode

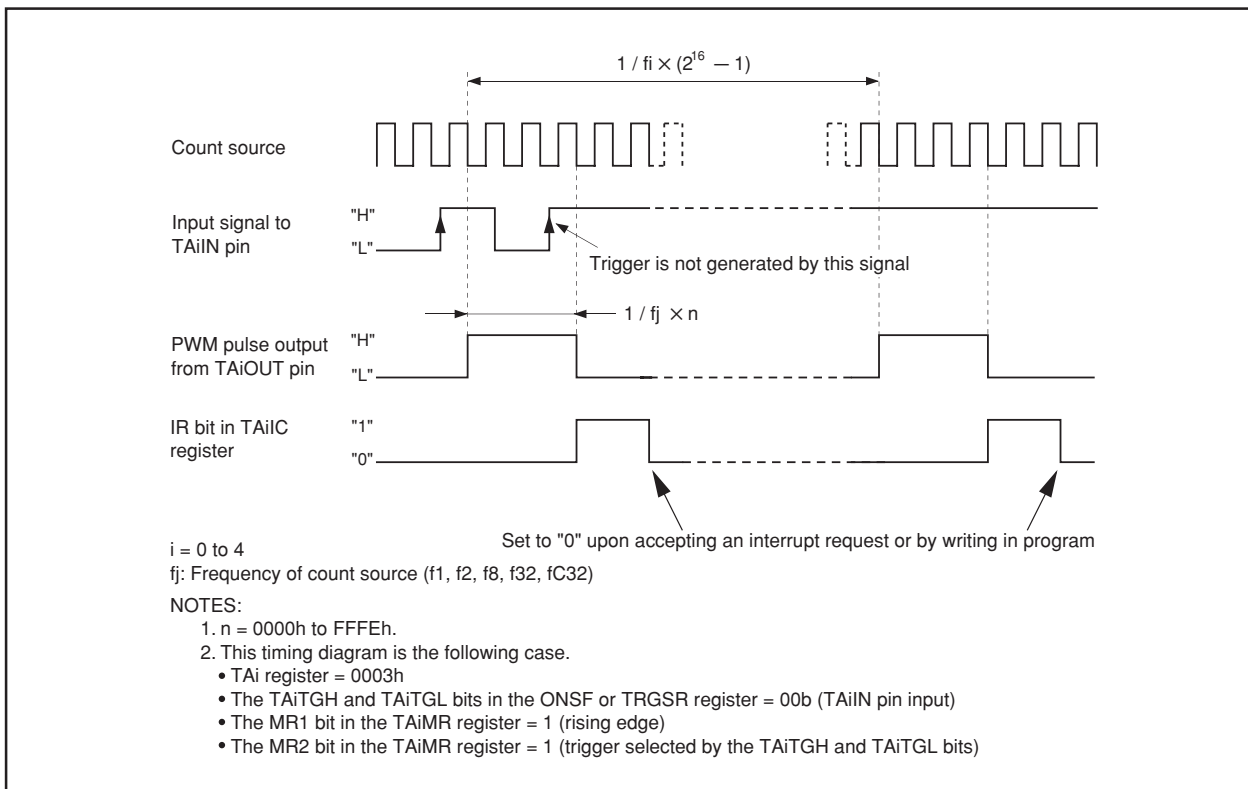


Figure 12.13 Example of 16-bit Pulse Width Modulator Operation

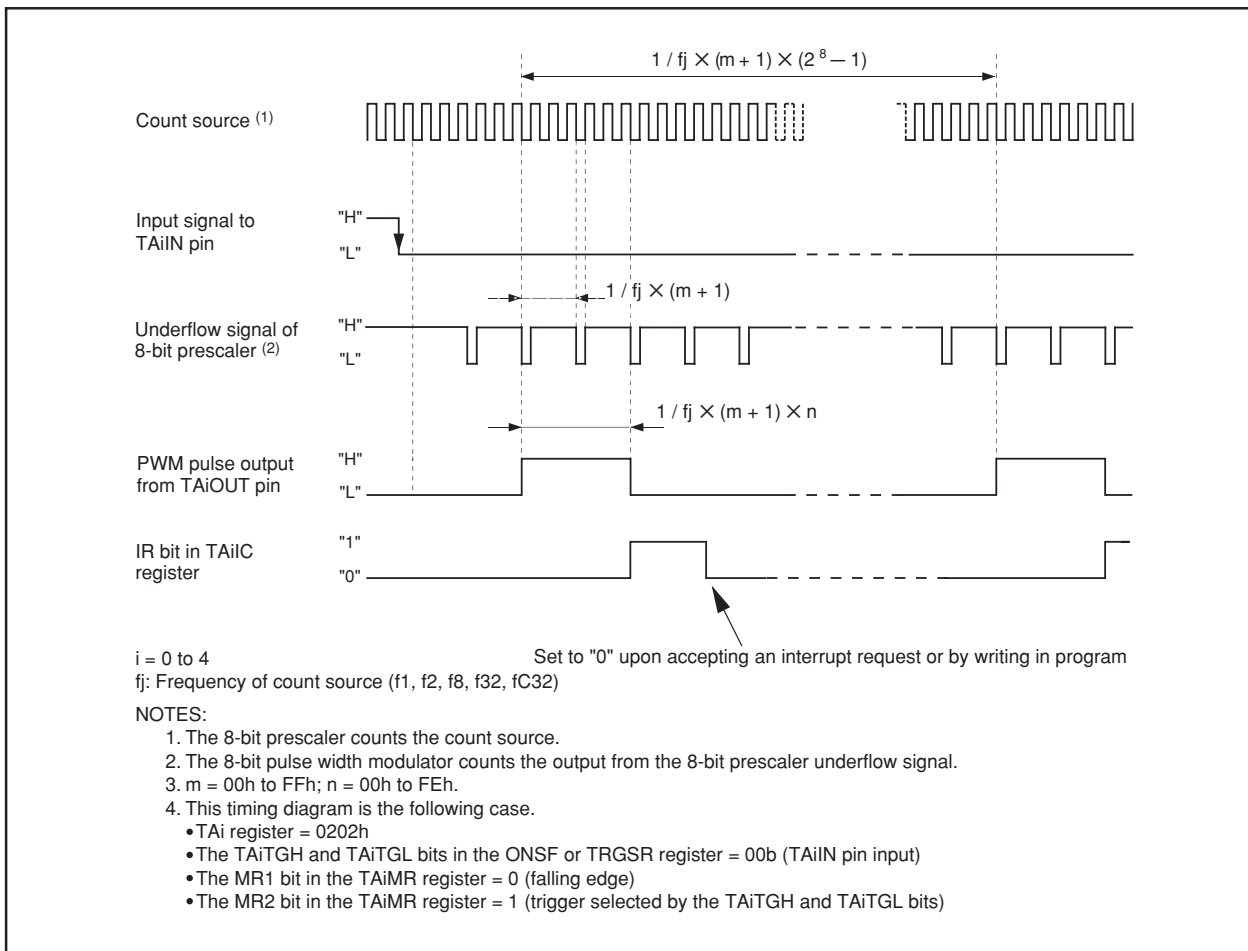


Figure 12.14 Example of 8-bit Pulse Width Modulator Operation

12.2 Timer B

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

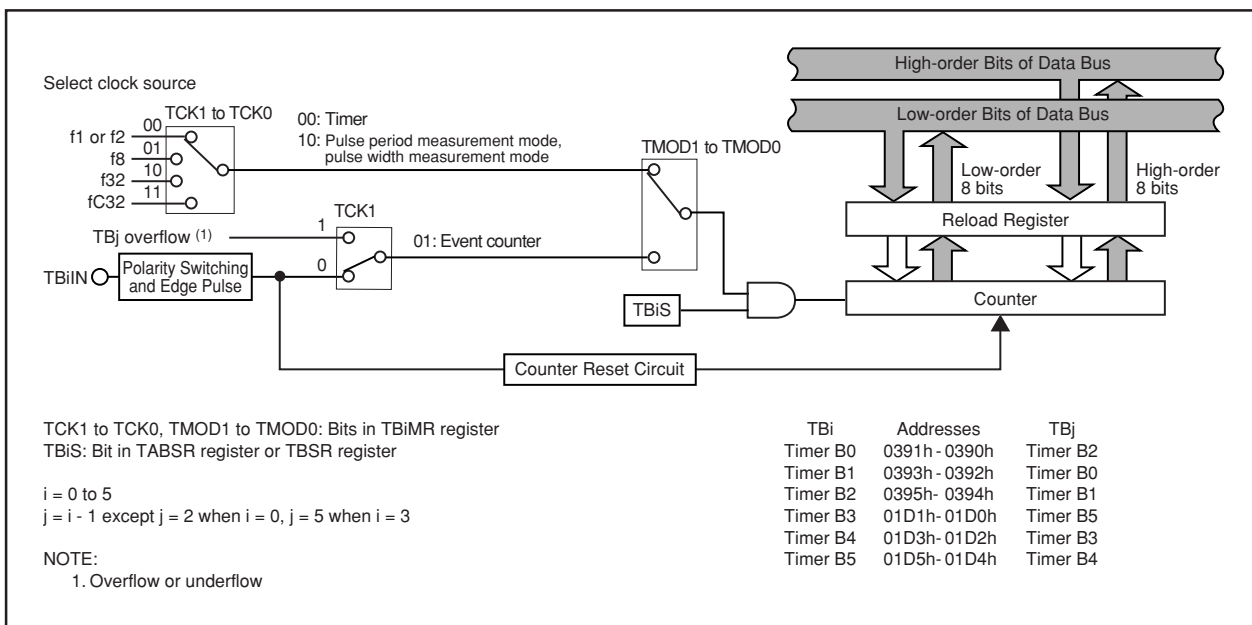


Figure 12.15 Timer B Block Diagram

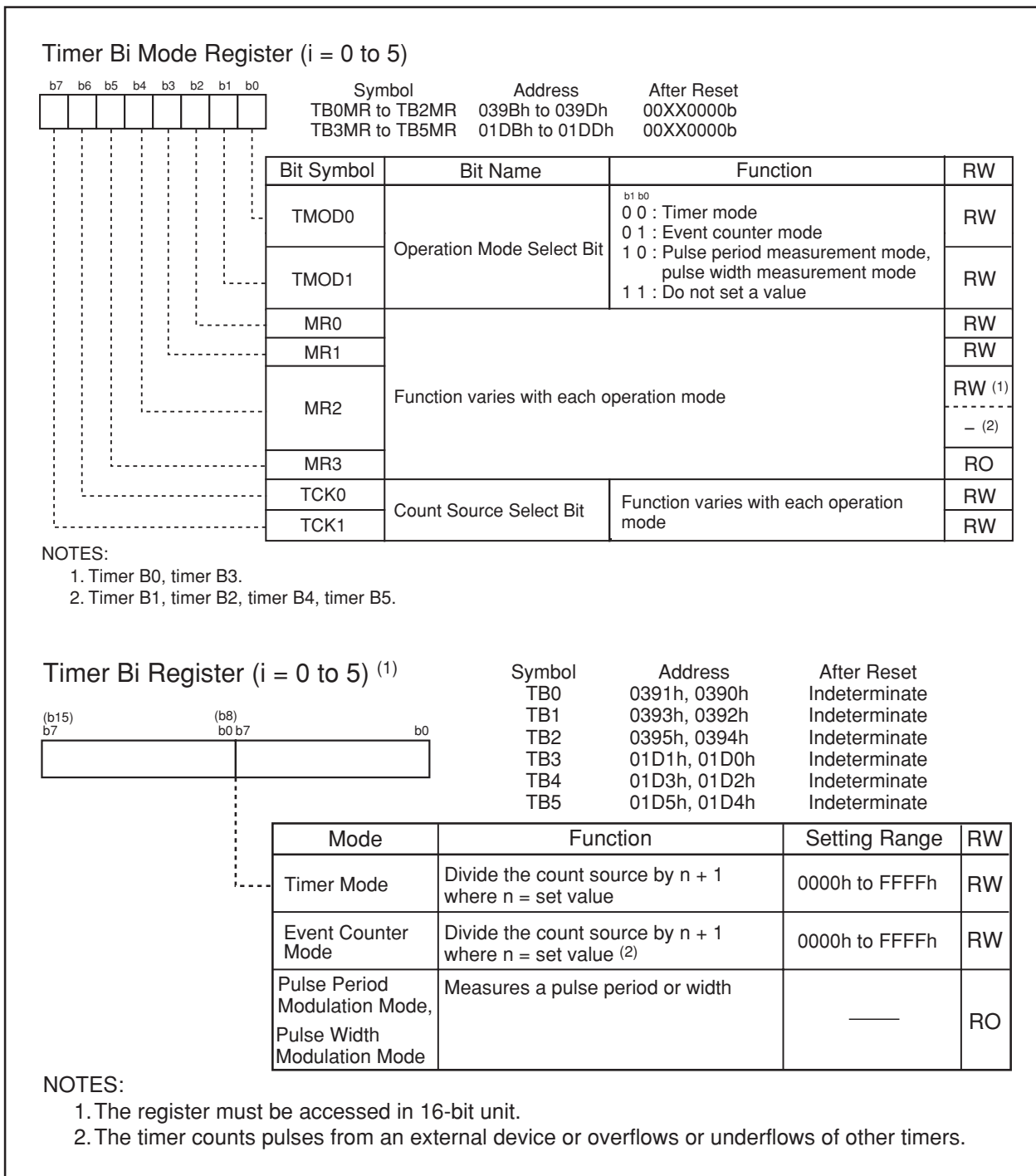


Figure 12.16 TB0MR to TB5MR Registers and TB0 to TB5 Registers

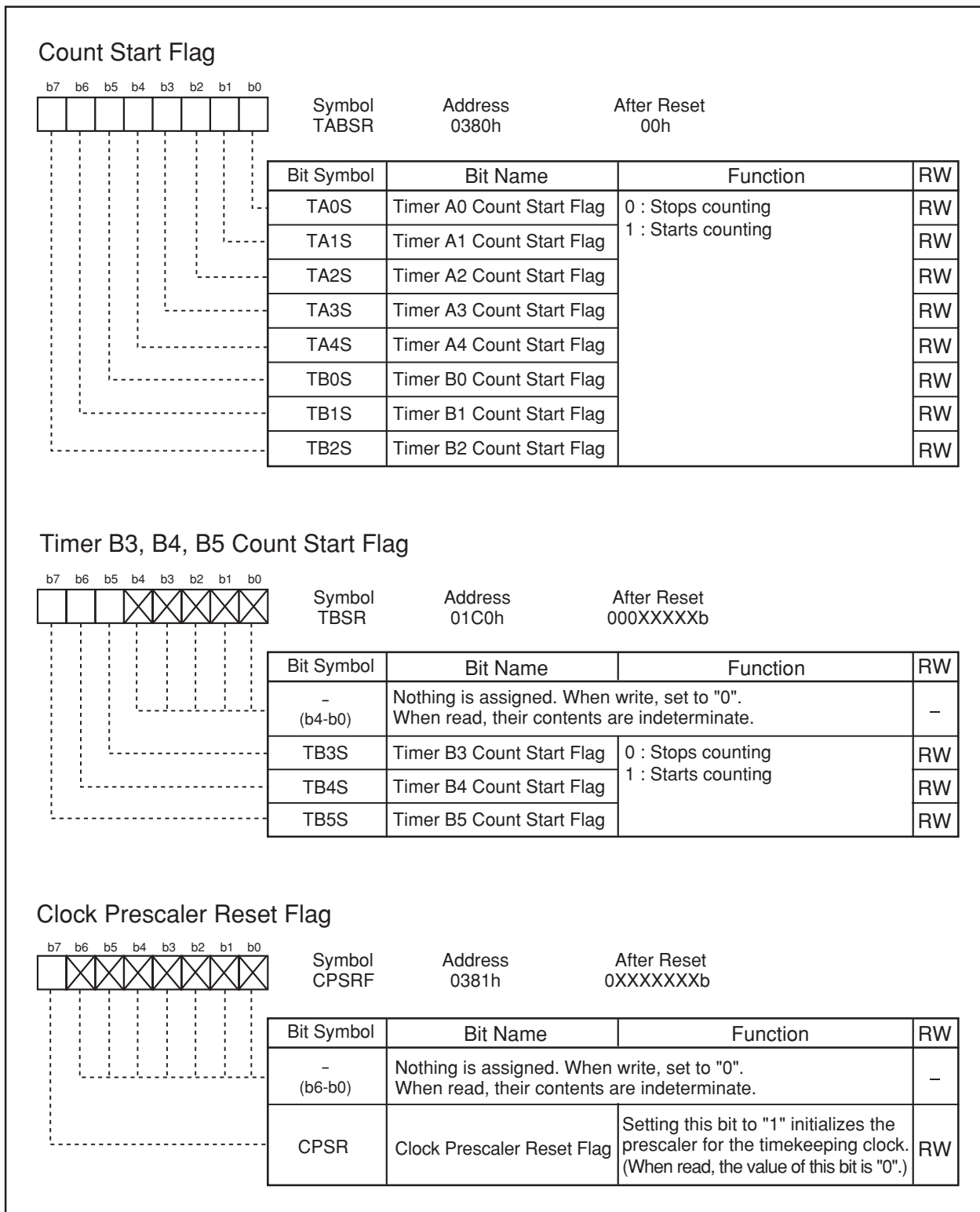


Figure 12.17 TABSR Register, TBSR Register and CPSRF Register

12.2.1 Timer Mode

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

Table 12.6 lists specifications in timer mode. Figure 12.18 shows TBiMR register in timer mode.

Table 12.6 Specifications in Timer Mode

Item	Specification
Count Source	f1, f2, f8, f32, fC32
Count Operation	<ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the reload register contents and continues counting
Divide Ratio	1/(n+1) n: set value of the TBi register 0000h to FFFFh
Count Start Condition	Set the TBiS bit ⁽¹⁾ to "1" (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"> When not counting and until the 1st count source is input after counting start Value written to the TBi register is written to both reload register and counter When counting (after 1st count source input) Value written to the TBi register is written to only reload register (Transferred to counter when reloaded next)

i = 0 to 5

NOTE:

- 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.

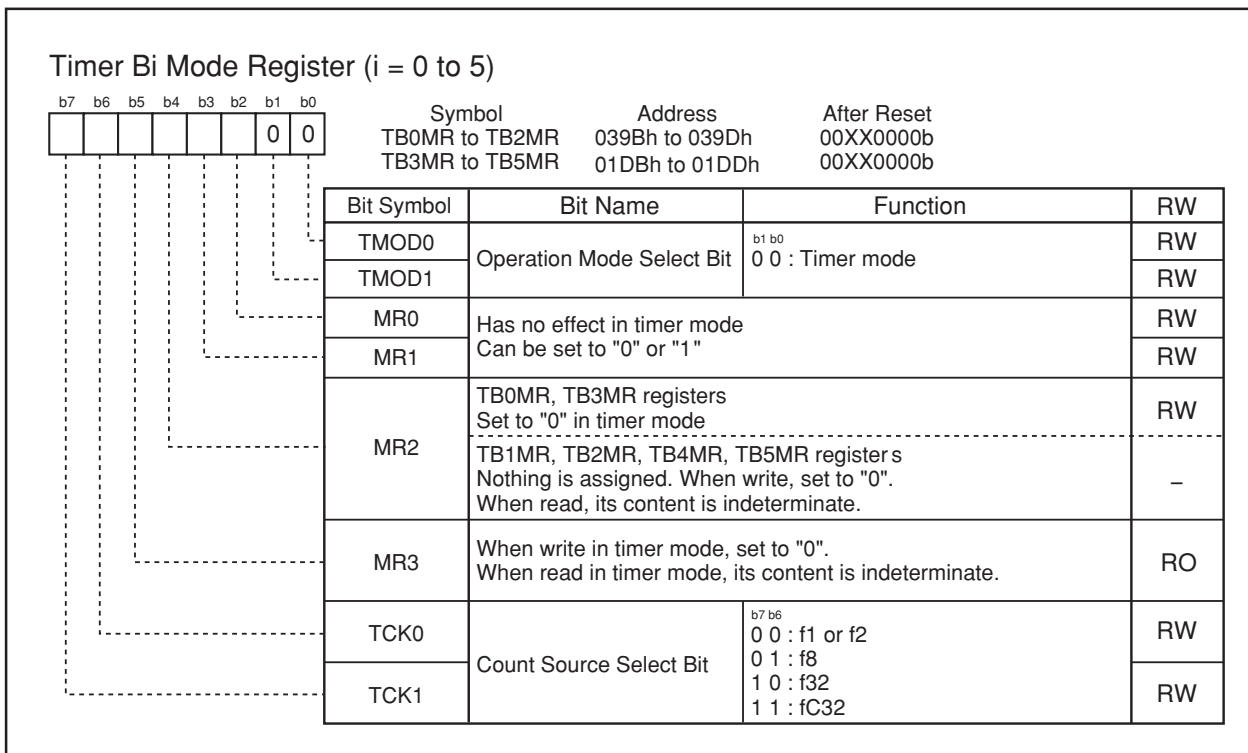


Figure 12.18 TB0MR to TB5MR Registers in Timer Mode

12.2.2 Event Counter Mode

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

Table 12.7 Specifications in Event Counter Mode

Item	Specification
Count Source	<ul style="list-style-type: none"> External signals input to TBiIN pin (effective edge can be selected in program) Timer Bj overflow or underflow
Count Operation	<ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the reload register contents and continues counting
Divide Ratio	1/(n+1) n: set value of the TBi register 0000h to FFFFh
Count Start Condition	Set TBiS bit ⁽¹⁾ to "1" (start counting)
Count Stop Condition	Set TBiS bit to "0" (stop counting)
Interrupt Request Generation Timing	Timer underflow
TBiIN Pin Function	Count source input
Read from Timer	Count value can be read by reading the TBi register
Write to Timer	<ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to the TBi register is written to both reload register and counter When counting (after 1st count source input) Value written to the TBi register is written to only reload register (Transferred to counter when reloaded next)

i = 0 to 5

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

NOTE:

- 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.

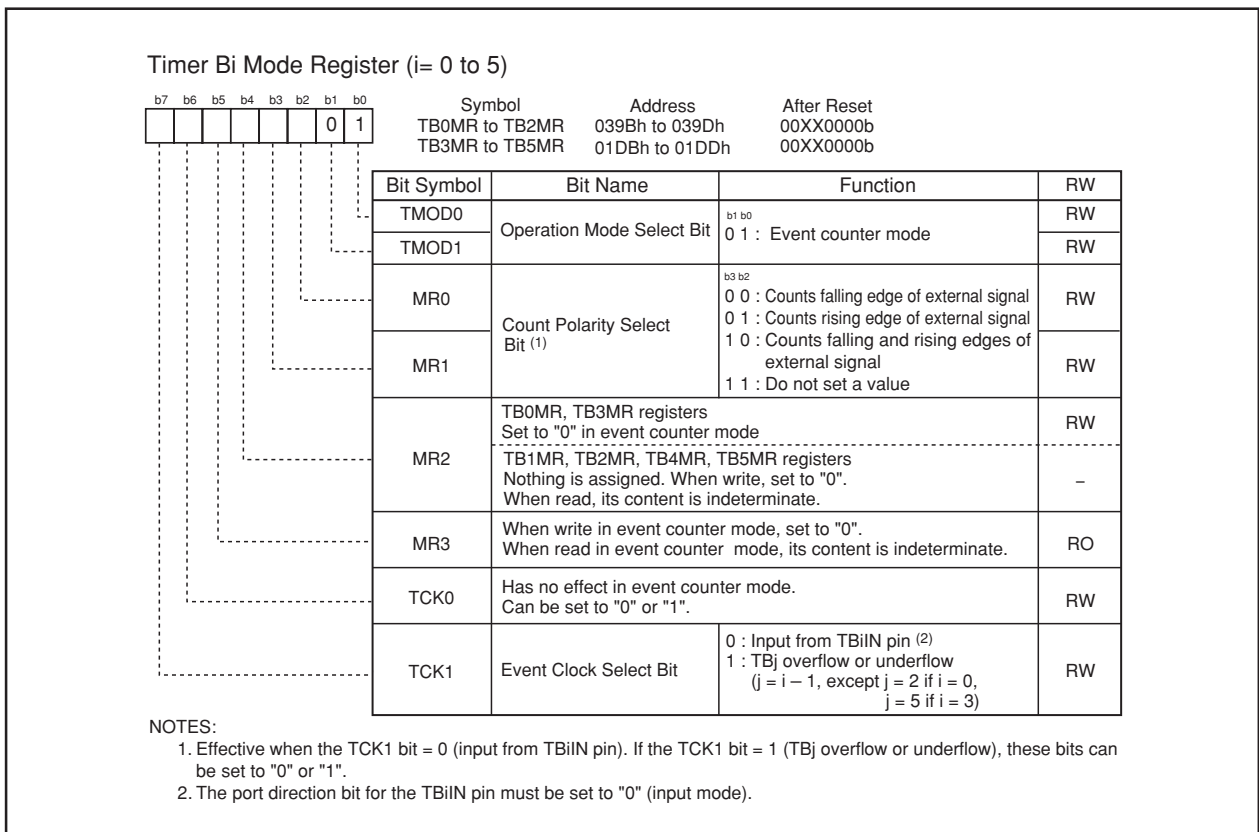


Figure 12.19 TB0MR to TB5MR Registers in Event Counter Mode

12.2.3 Pulse Period and Pulse Width Measurement Mode

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

Table 12.8 Specifications in Pulse Period and Pulse Width Measurement Mode

Item	Specification
Count Source	f1, f2, f8, f32, fC32
Count Operation	<ul style="list-style-type: none"> • Up-count • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to "0000h" to continue counting.
Count Start Condition	Set the TBiS bit ⁽¹⁾ to "1" (start counting)
Count Stop Condition	Set the TBiS bit to "0" (stop counting)
Interrupt Request Generation Timing	<ul style="list-style-type: none"> • When an effective edge of measurement pulse is input ⁽²⁾ • 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).
TBiIN Pin Function	Measurement pulse input
Read from Timer	Contents of the reload register (measurement result) can be read by reading TBi register ⁽³⁾
Write to Timer	Value written to the TBi register is written to neither reload register nor counter

i = 0 to 5

NOTES:

1. 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.

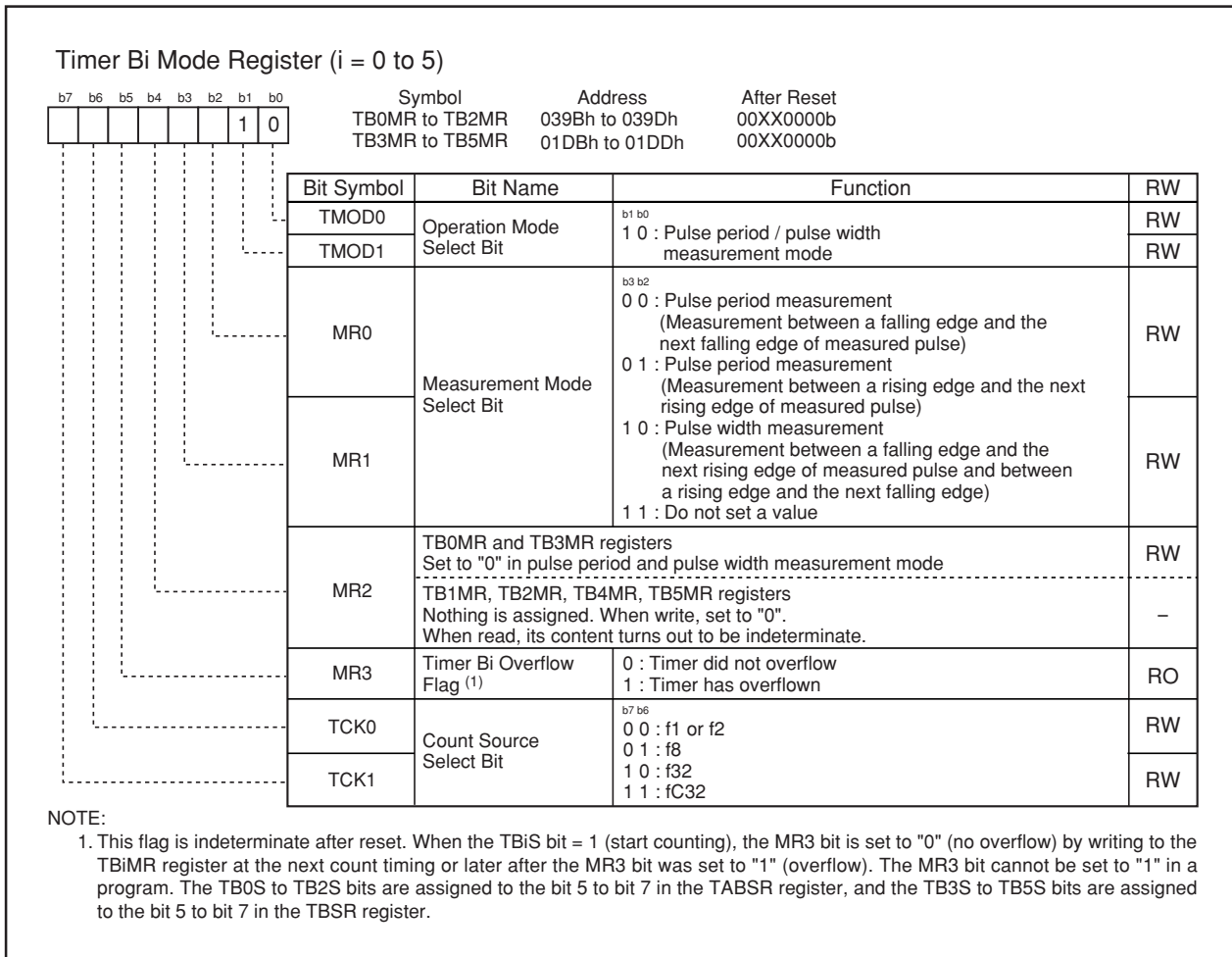


Figure 12.20 TB0MR to TB5MR Registers in Pulse Period and Pulse Width Measurement Mode

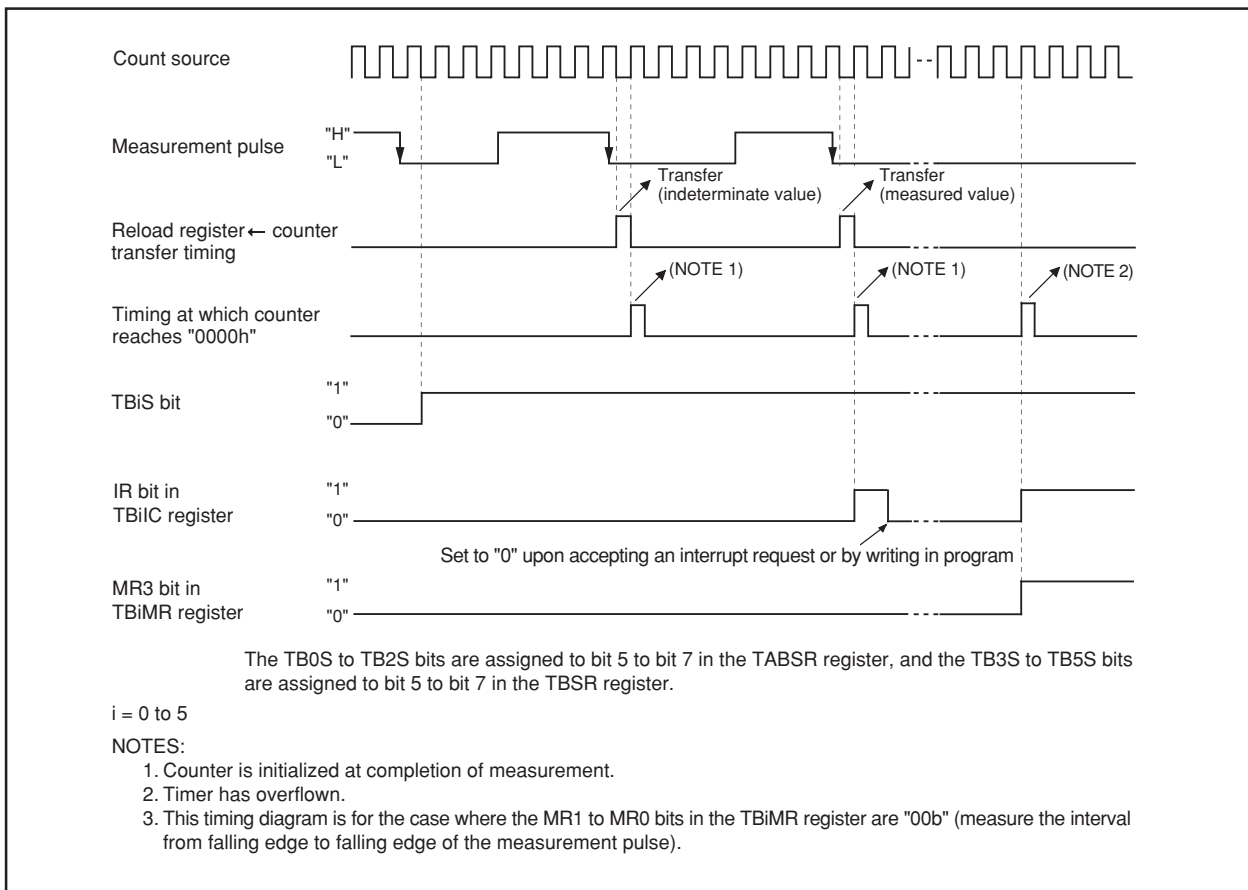


Figure 12.21 Operation Timing When Measuring Pulse Period

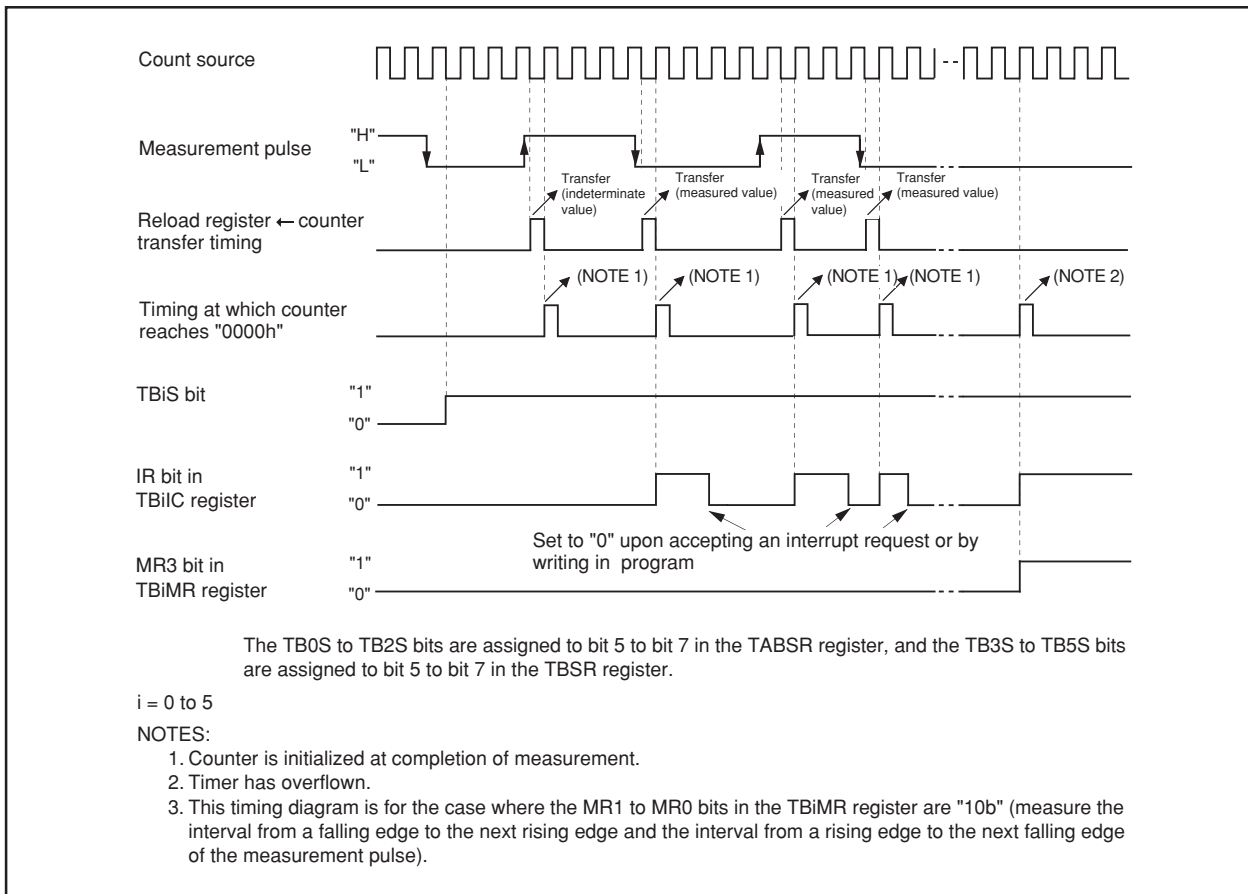


Figure 12.22 Operation Timing When Measuring Pulse Width

13. Three-Phase Motor Control Timer Function

Timers A1, A2, A4 and B2 can be used to output three-phase motor drive waveforms. Table 13.1 lists the specifications of the three-phase motor control timer function. Figure 13.1 shows the block diagram for three-phase motor control timer function. Also, the related registers are shown on Figures 13.2 to 13.8.

Table 13.1 Three-Phase Motor Control Timer Function Specifications

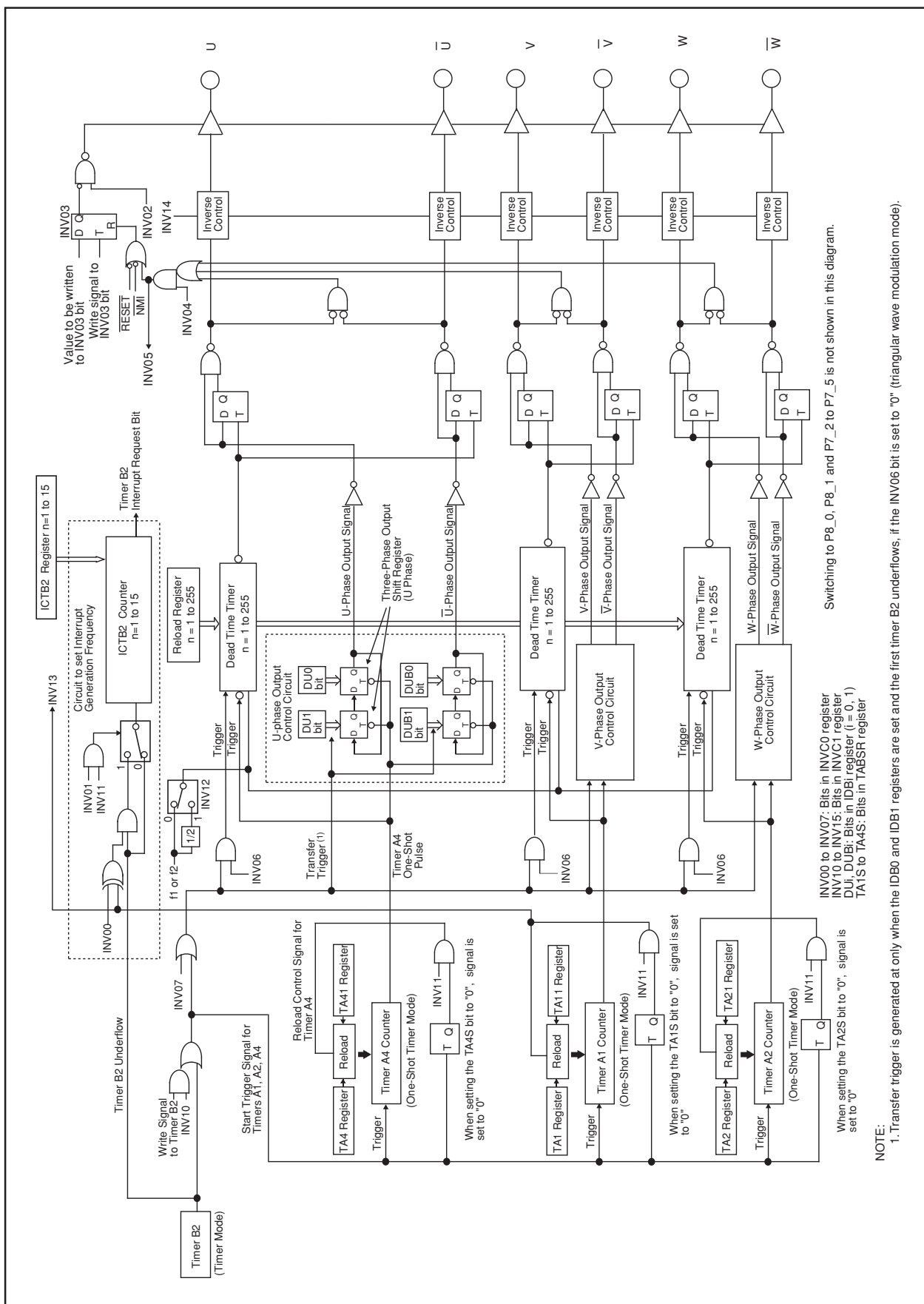
Item	Specification
Three-Phase Waveform Output Pin	Six pins (U, \bar{U} , V, \bar{V} , W, \bar{W})
Forced Cutoff Input ⁽¹⁾	Input "L" to NMI pin
Used Timers	Timer A4, A1, A2 (used in the one-shot timer mode) <ul style="list-style-type: none"> • Timer A4: U- and \bar{U}-phase waveform control • Timer A1: V- and \bar{V}-phase waveform control • Timer A2: W- and \bar{W}-phase waveform control Timer B2 (used in the timer mode) <ul style="list-style-type: none"> • Carrier wave cycle control Dead time timer (3 eight-bit timer and shared reload register) <ul style="list-style-type: none"> • Dead time control
Output Waveform	Triangular wave modulation, Sawtooth wave modification <ul style="list-style-type: none"> • Enable to output "H" or "L" for one cycle • Enable to set positive-phase level and negative-phase level respectively
Carrier Wave Cycle	Triangular wave modulation: count source $\times (m+1) \times 2$ Sawtooth wave modulation: count source $\times (m+1)$ m: Setting value of the TB2 register, 0 to 65535 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"), 1 to 65535 Count source: f1, f2, f8, f32, fC32
Dead Time	Count source $\times p$, or no dead time p: Setting value of the DTT register, 1 to 255 Count source: f1, f2, f1 divided by 2, f2 divided by 2
Active Level	Enable to select "H" or "L"
Positive and Negative-Phase Concurrent Active Disable Function	Positive and negative-phases concurrent active disable function
Interrupt Frequency	For Timer B2 interrupt, select a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis

NOTE:

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

Related pins:

- P7_2/CLK2/TA1OUT/V
- P7_3/CTS2/RTS2/TA1IN/ \bar{V}
- P7_4/TA2OUT/W/(CLK4)
- P7_5/TA2IN/ \bar{W} /(SOUT4)
- P8_0/TA4OUT/U(SIN4)
- P8_1/TA4IN/ \bar{U}



Switching to P8_0, P8_1 and P7_2 to P7_5 is not shown in this diagram.

INV00 to INV07: Bits in INVC0 register
 INV10 to INV15: Bits in INVC1 register
 DU1, DUB1: Bits in IDBI register (I = 0, 1)
 TA1S to TA4S: Bits in TABSR register

NOTE:

1. Transfer trigger is generated at only when the IDB0 and IDB1 registers are set and the first timer B2 underflows, if the INV06 bit is set to "0" (triangular wave modulation mode).

Figure 13.1 Three-Phase Motor Control Timer Function Block Diagram

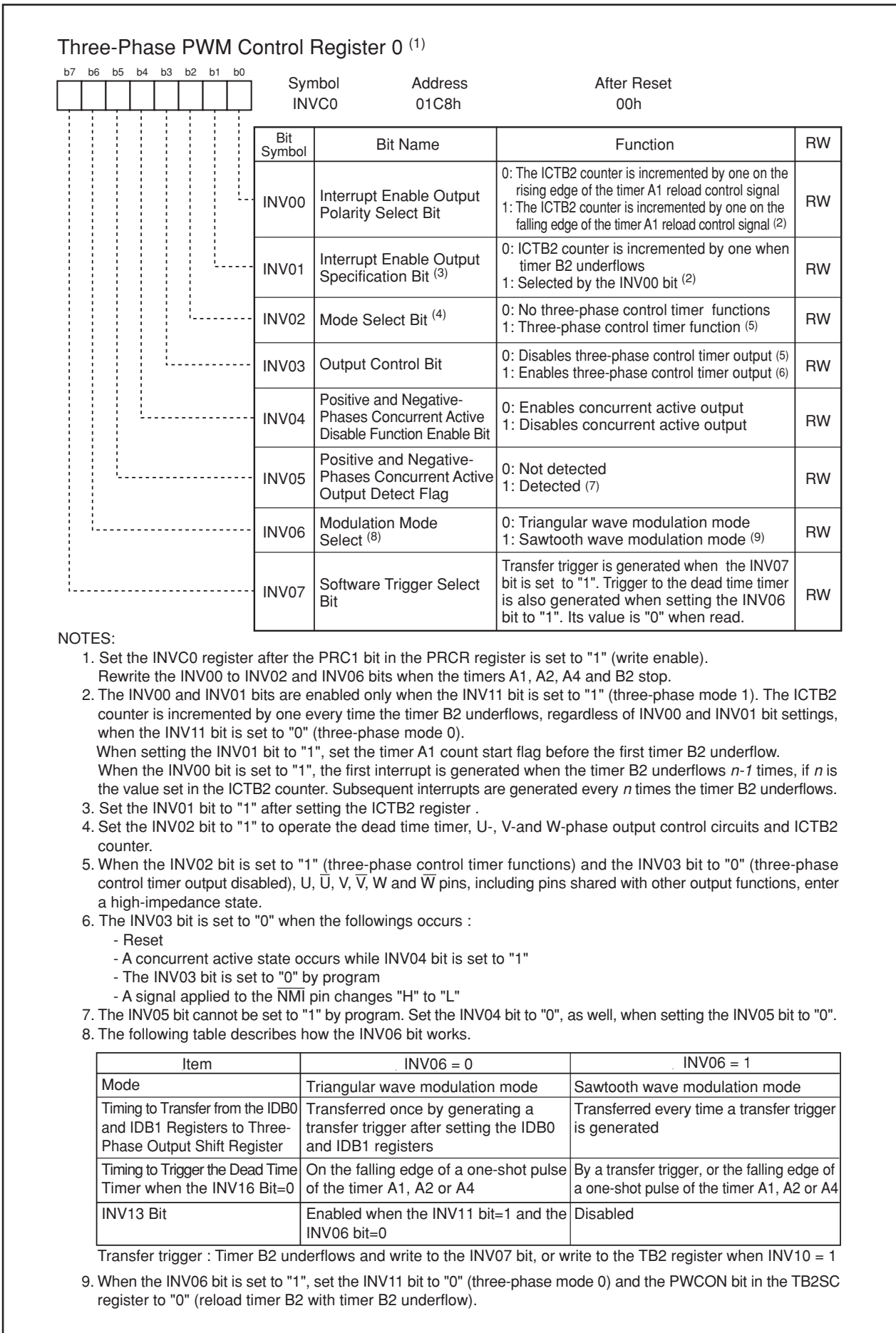


Figure 13.2 INVC0 Register

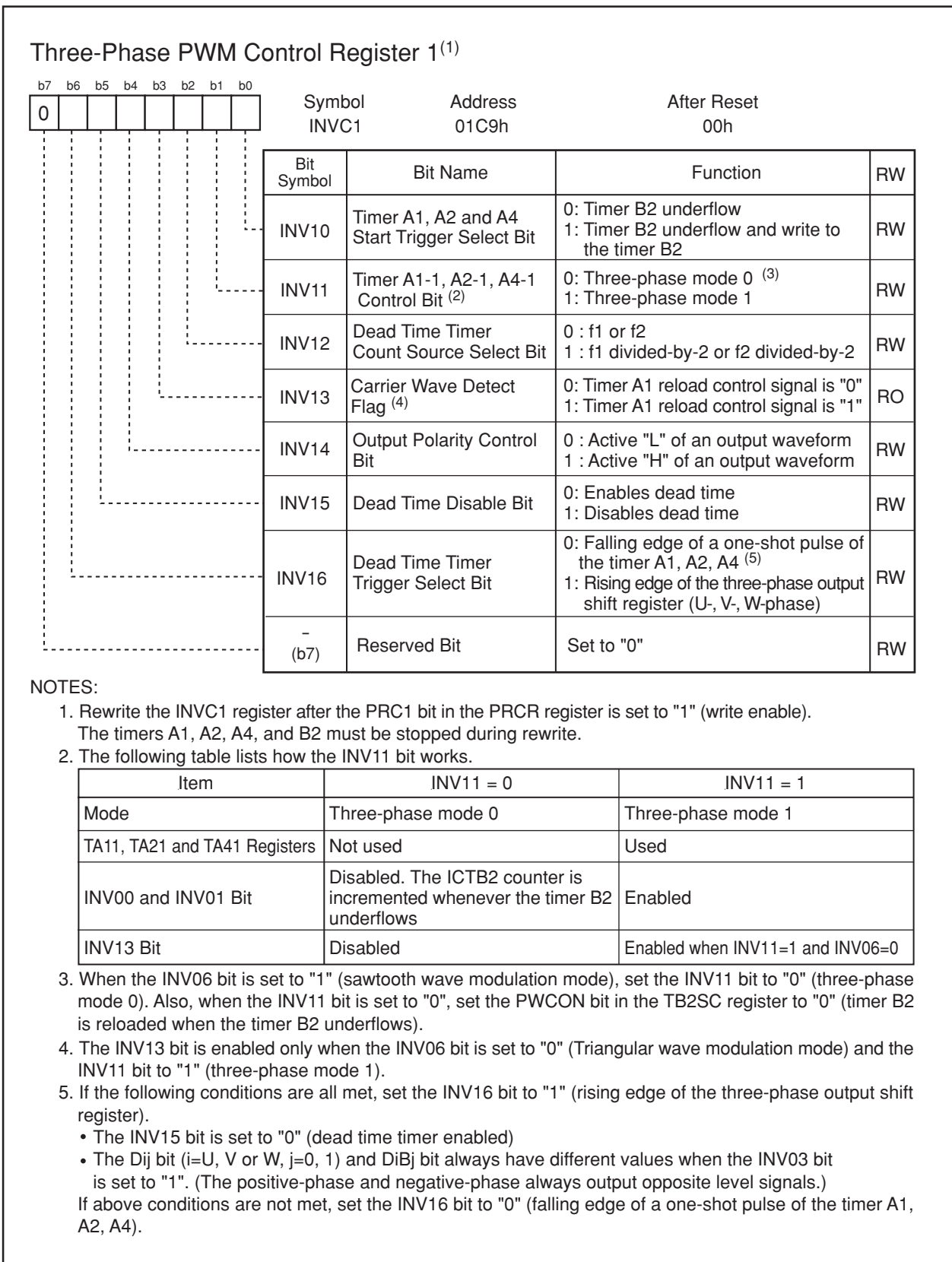


Figure 13.3 INVC1 Register

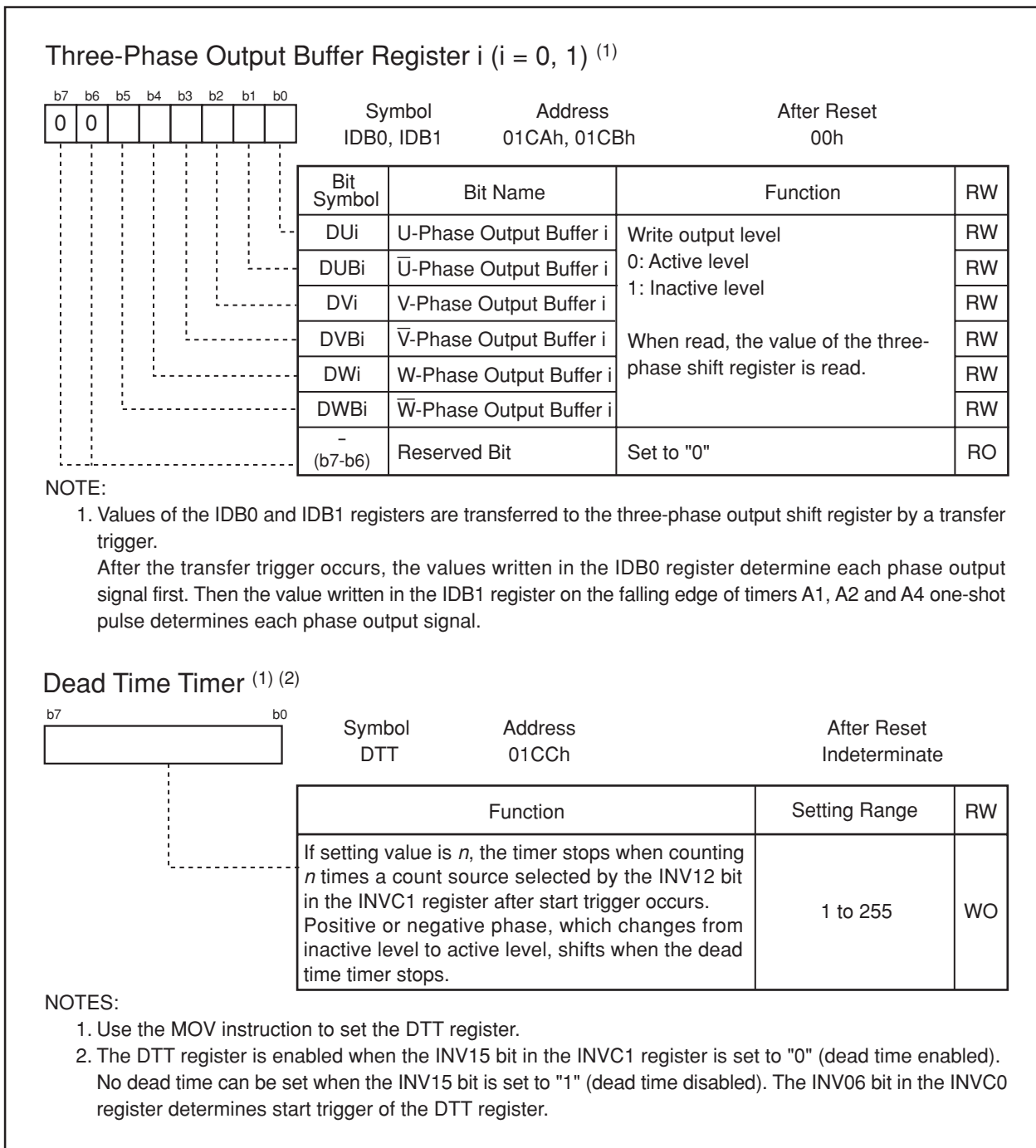


Figure 13.4 IDB0 and IDB1 Registers and DTT Register

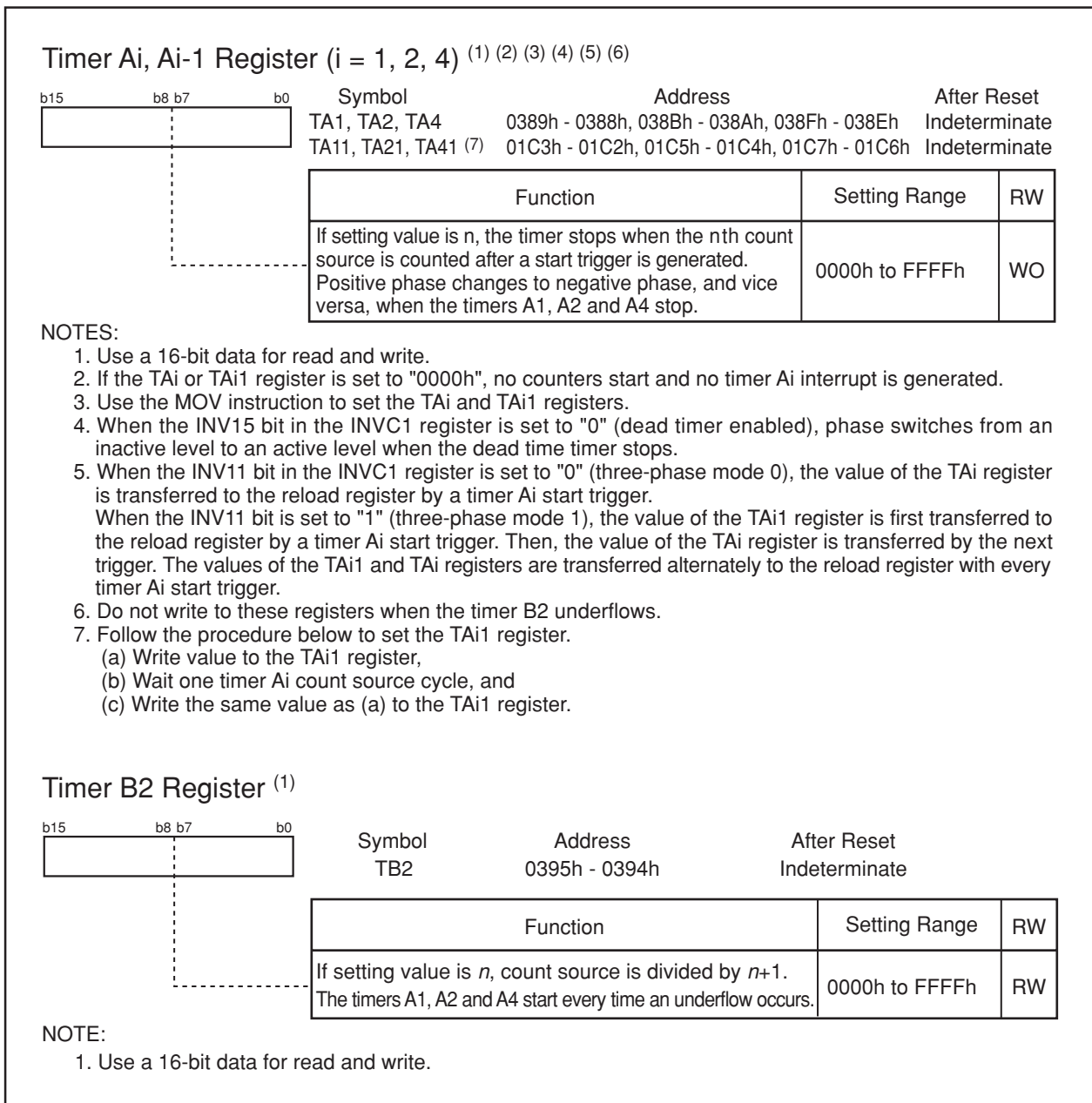


Figure 13.5 TA1, TA2, TA4, TA11, TA21 and TA41 Registers, and TB2 Register

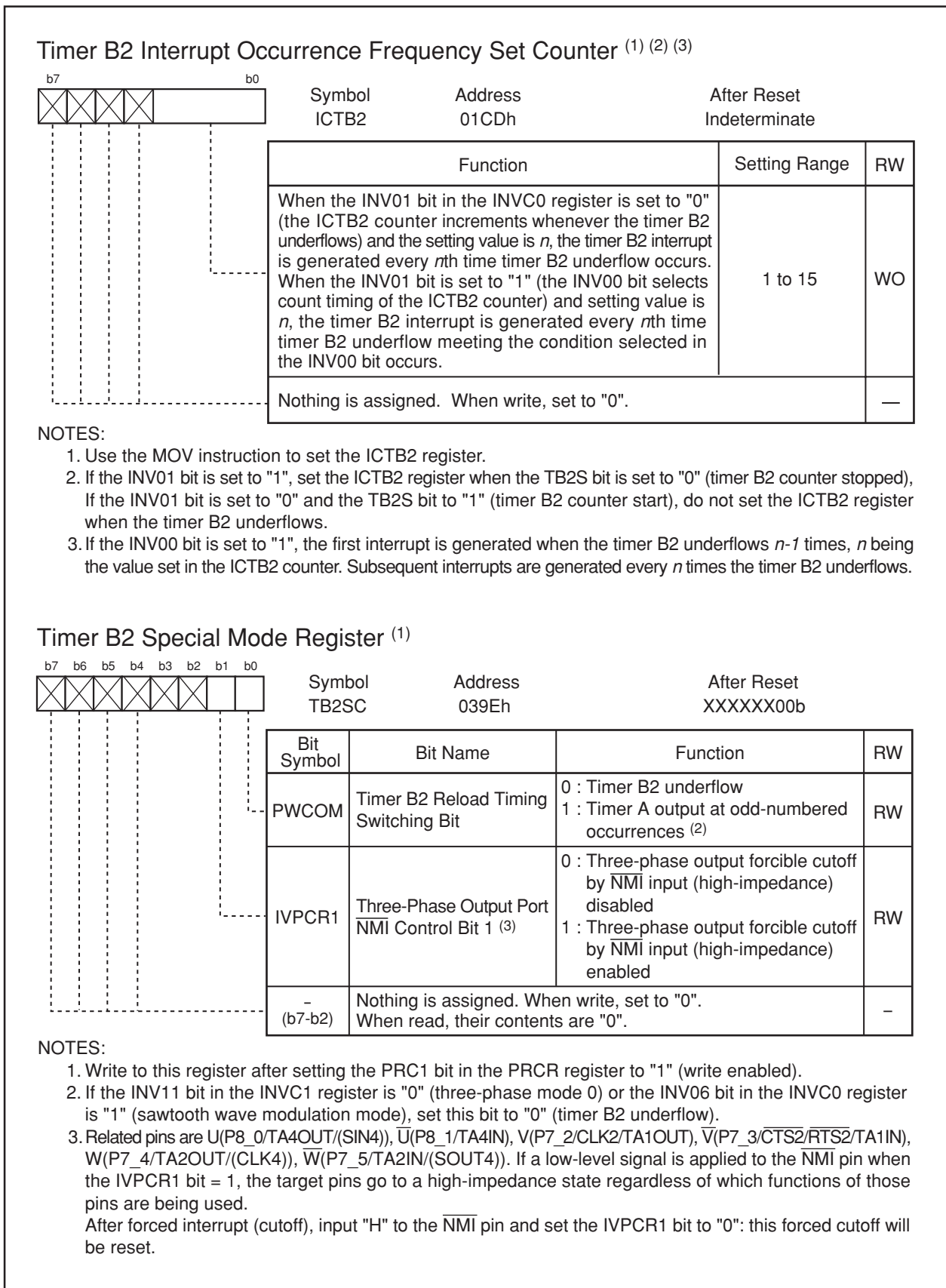


Figure 13.6 ICTB2 Register and TB2SC Register

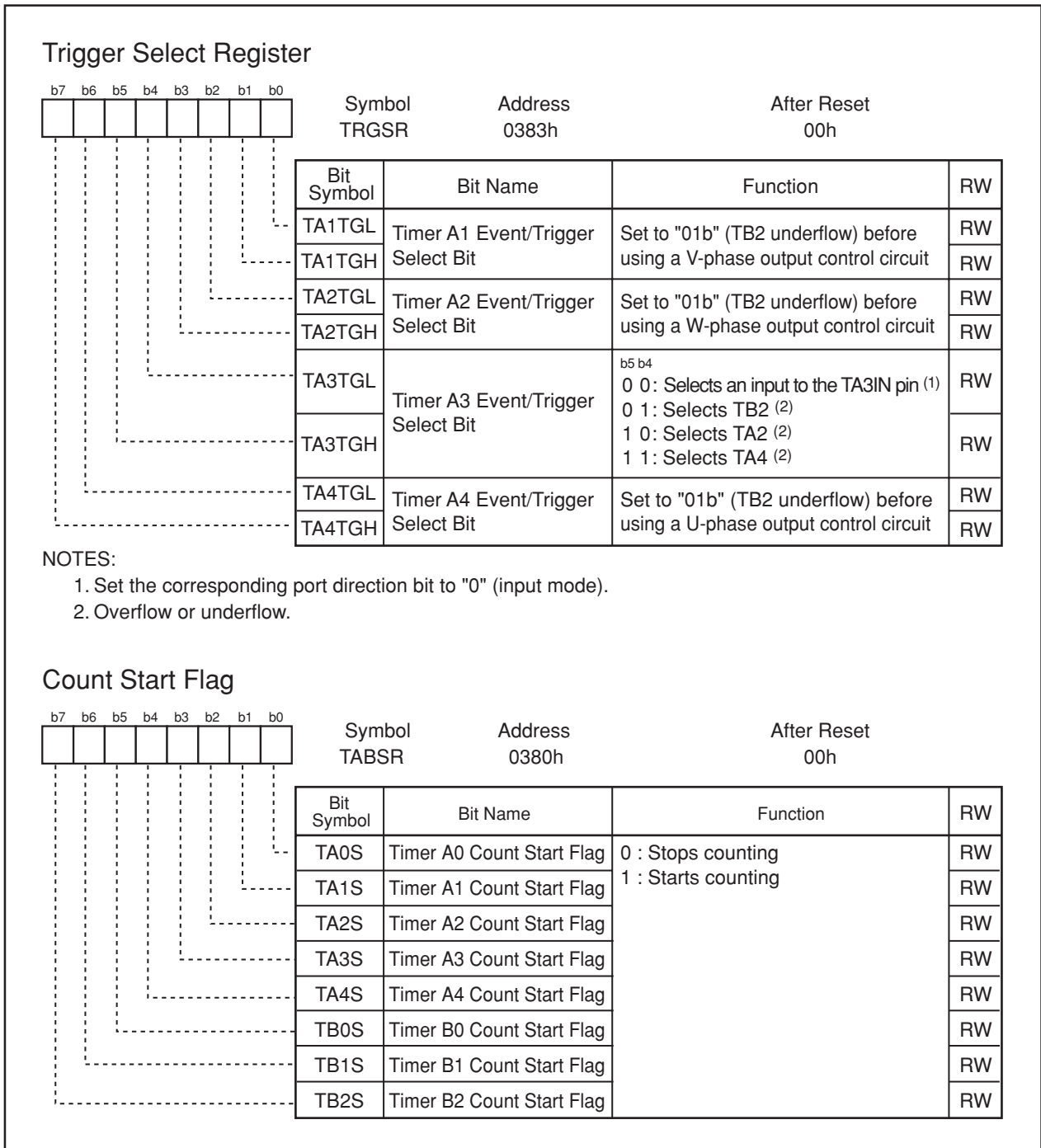


Figure 13.7 TRGSR Register and TRBSR Register

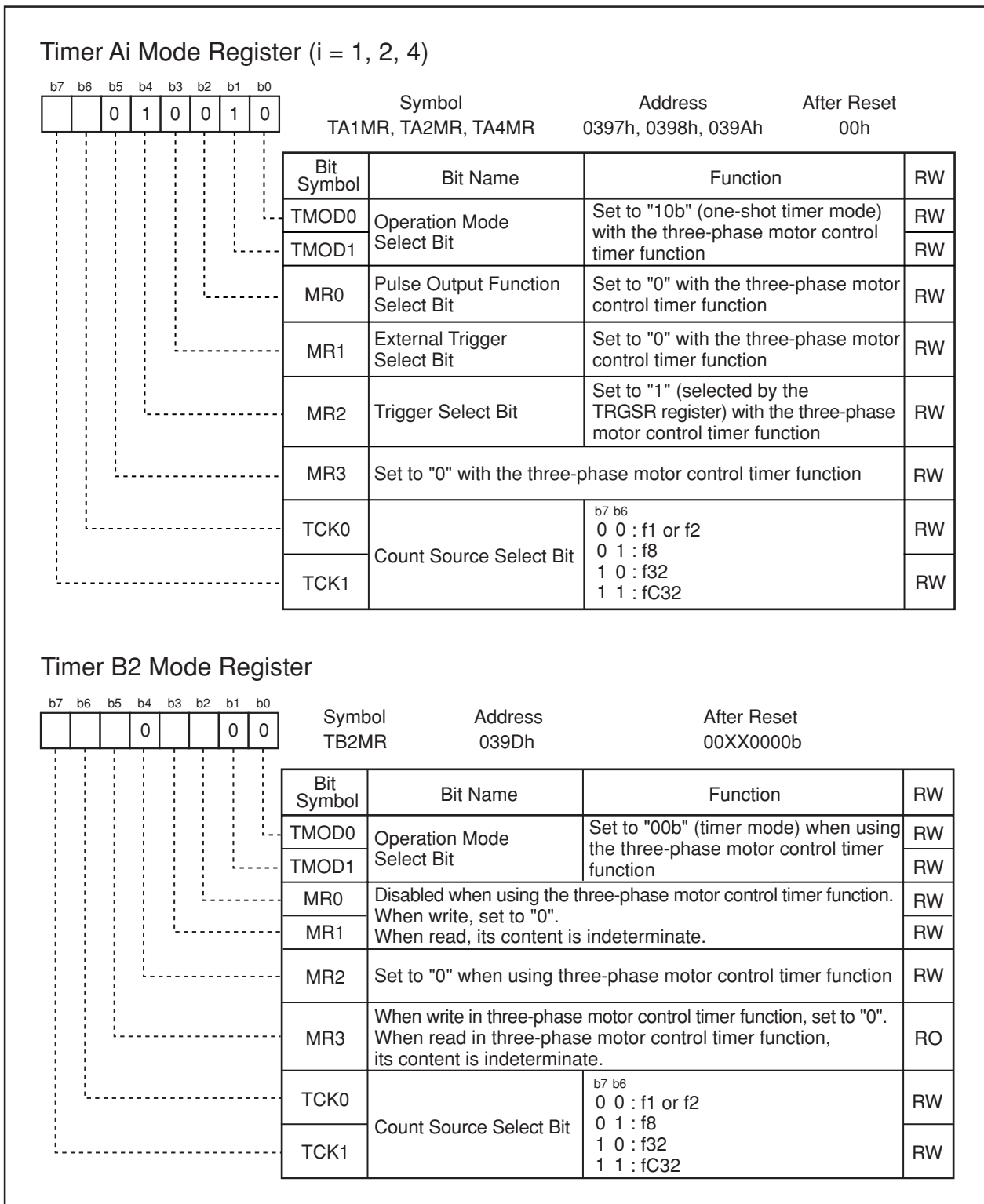


Figure 13.8 TA1MR, TA2MR and TA4MR Registers, and TB2MR Register

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 13.9 shows the example of triangular modulation waveform and Figure 13.10 shows the example of sawtooth modulation waveform.

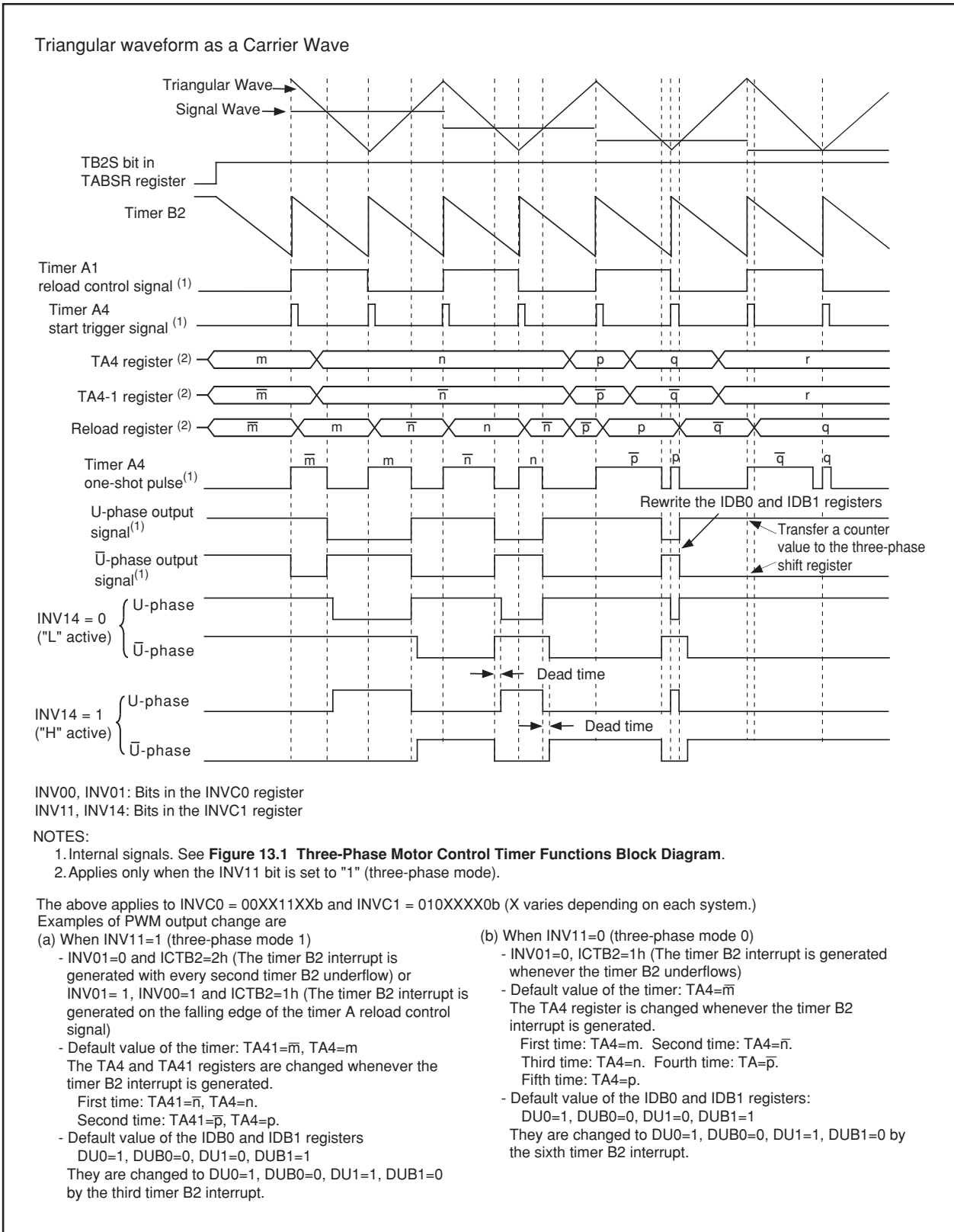


Figure 13.9 Triangular Wave Modulation Operation

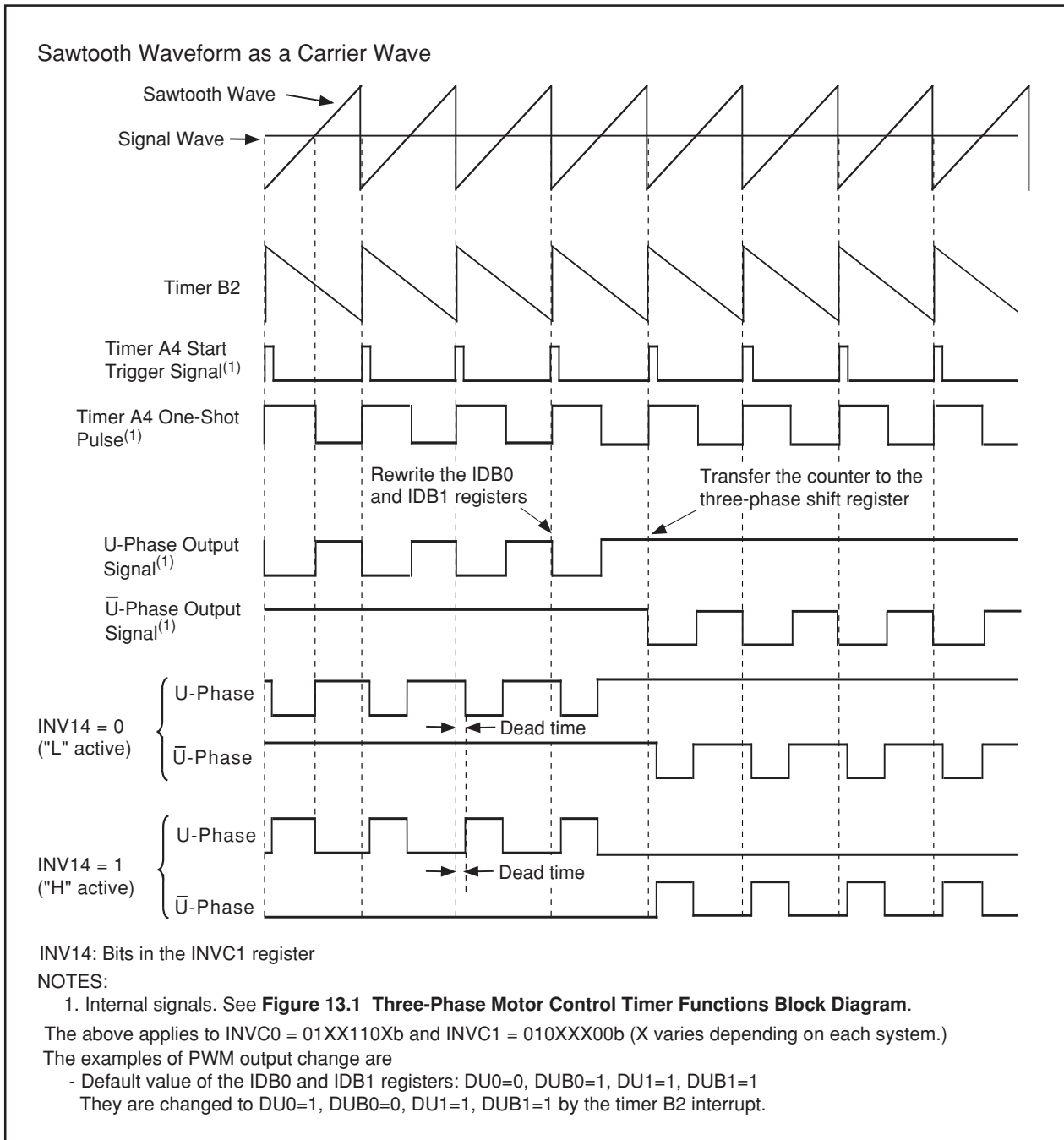


Figure 13.10 Sawtooth Wave Modulation Operation

14. Serial I/O

Serial I/O is configured with 7 channels: UART0 to UART2 and SI/O3 to SI/O6 ⁽¹⁾.

NOTE:

1. 100-pin version supports 5 channels; UART0 to UART2, SI/O3, SI/O4
128-pin version supports 7 channels; UART0 to UART2, SI/O3 to SI/O6

14.1 UARTi (i = 0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other. Figures 14.1 to 14.3 show the block diagram of UARTi. Figure 14.4 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 (I²C mode)
- Special mode 2
- Special mode 3 (Bus collision detection function, IE mode)
- Special mode 4 (SIM mode) : UART2

Figures 14.5 to 14.10 show the UARTi-related registers.

Refer to tables listing each mode for register setting.

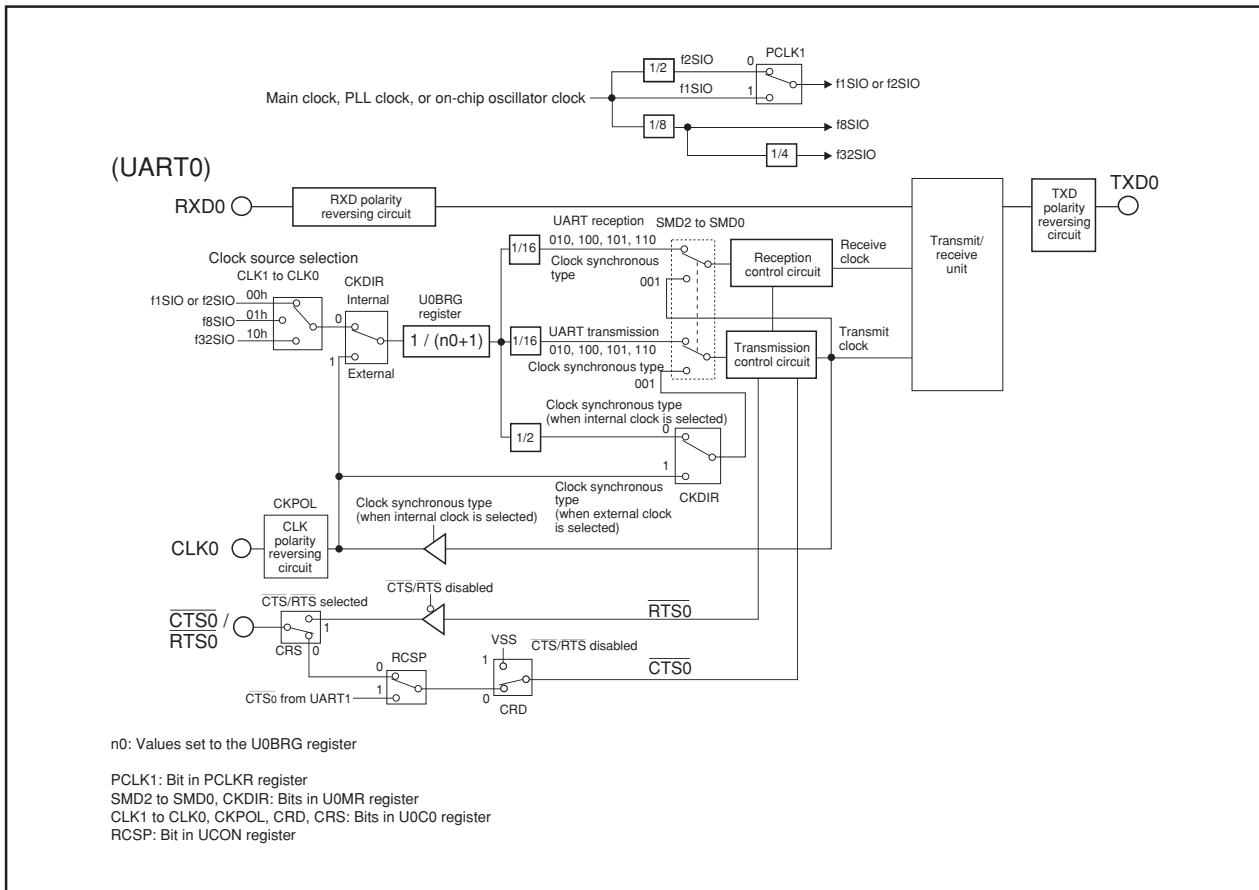


Figure 14.1 UART0 Block Diagram

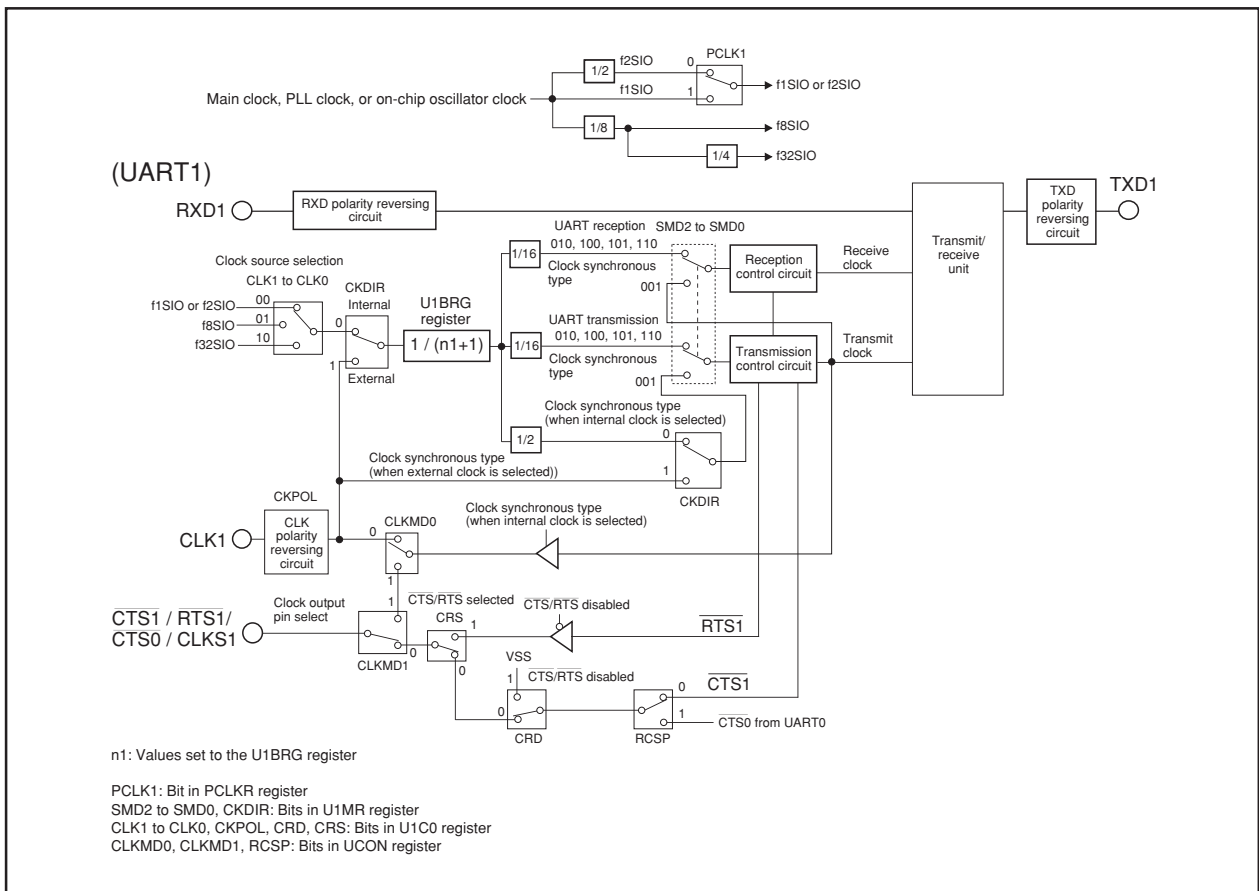


Figure 14.2 UART1 Block Diagram

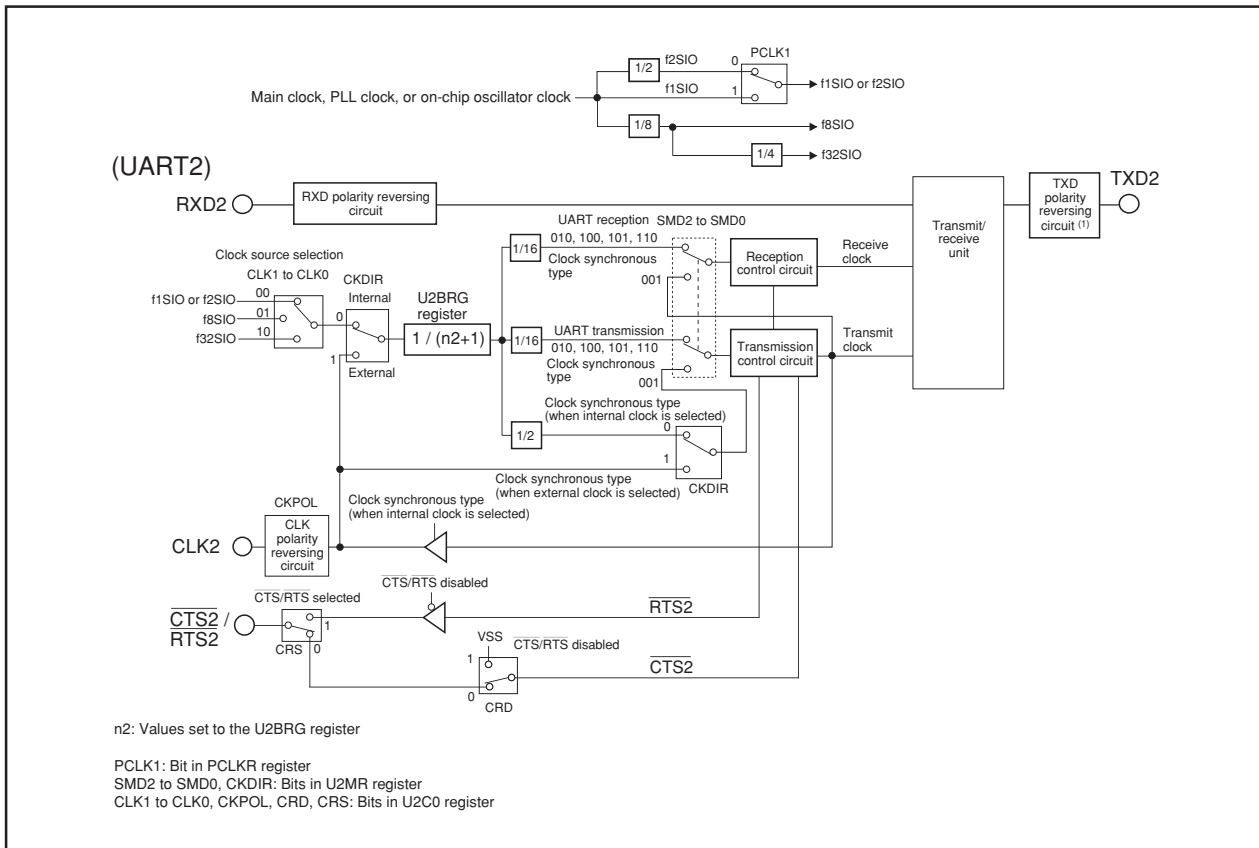


Figure 14.3 UART2 Block Diagram

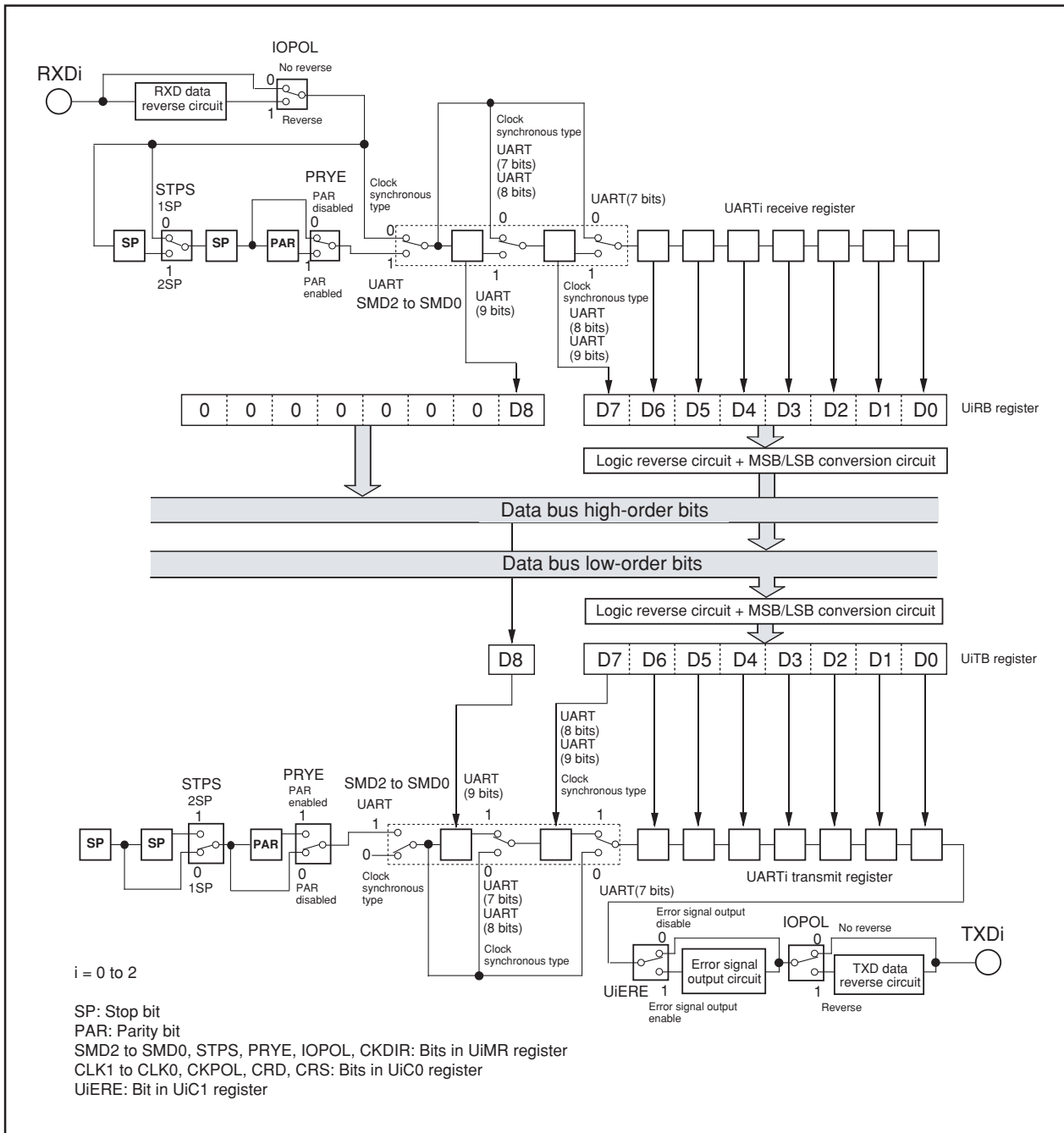


Figure 14.4 UARTi Transmit/Receive Unit

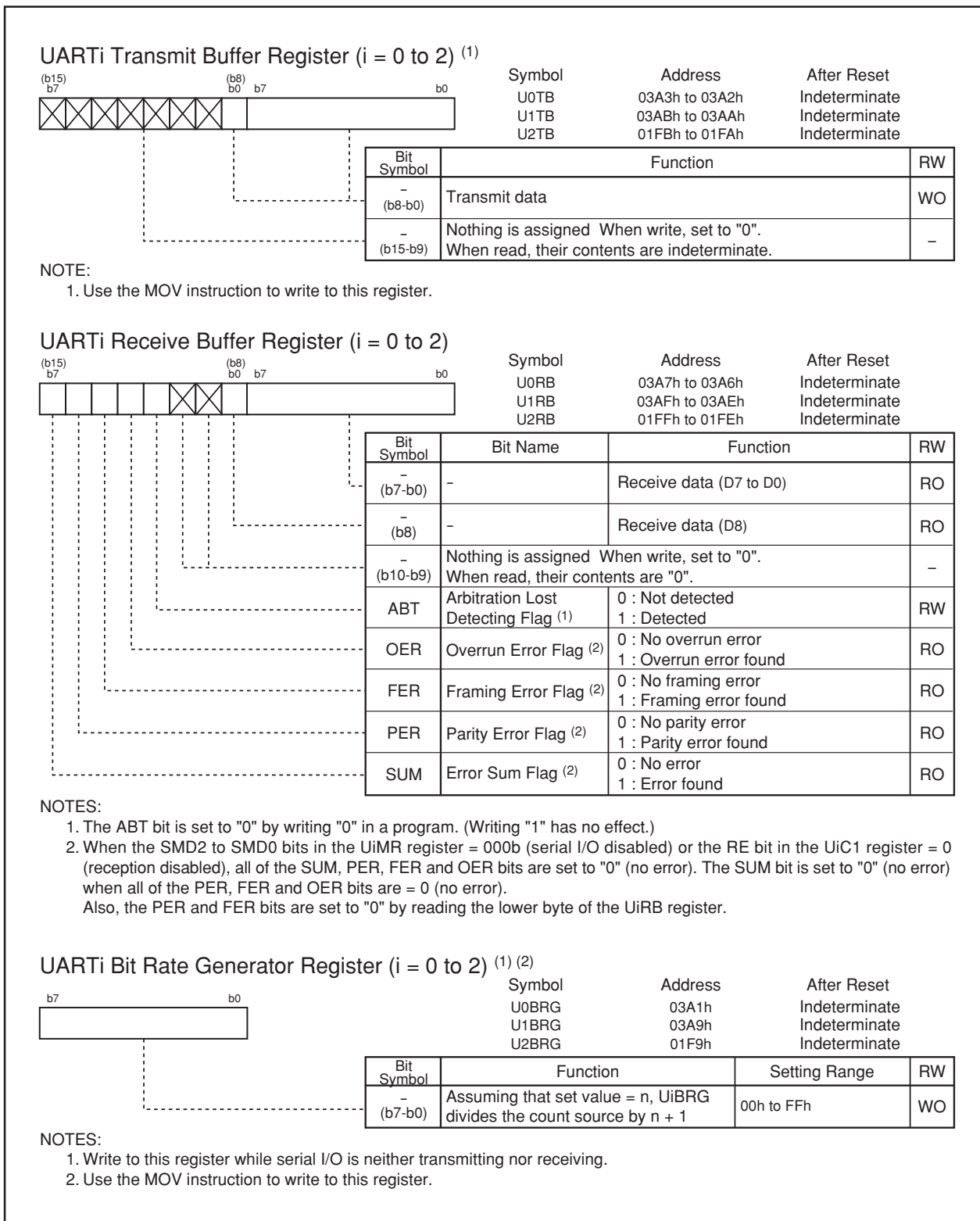


Figure 14.5 U0TB to U2TB Registers, U0RB to U2RB Registers, and U0BRG to U2BRG Registers

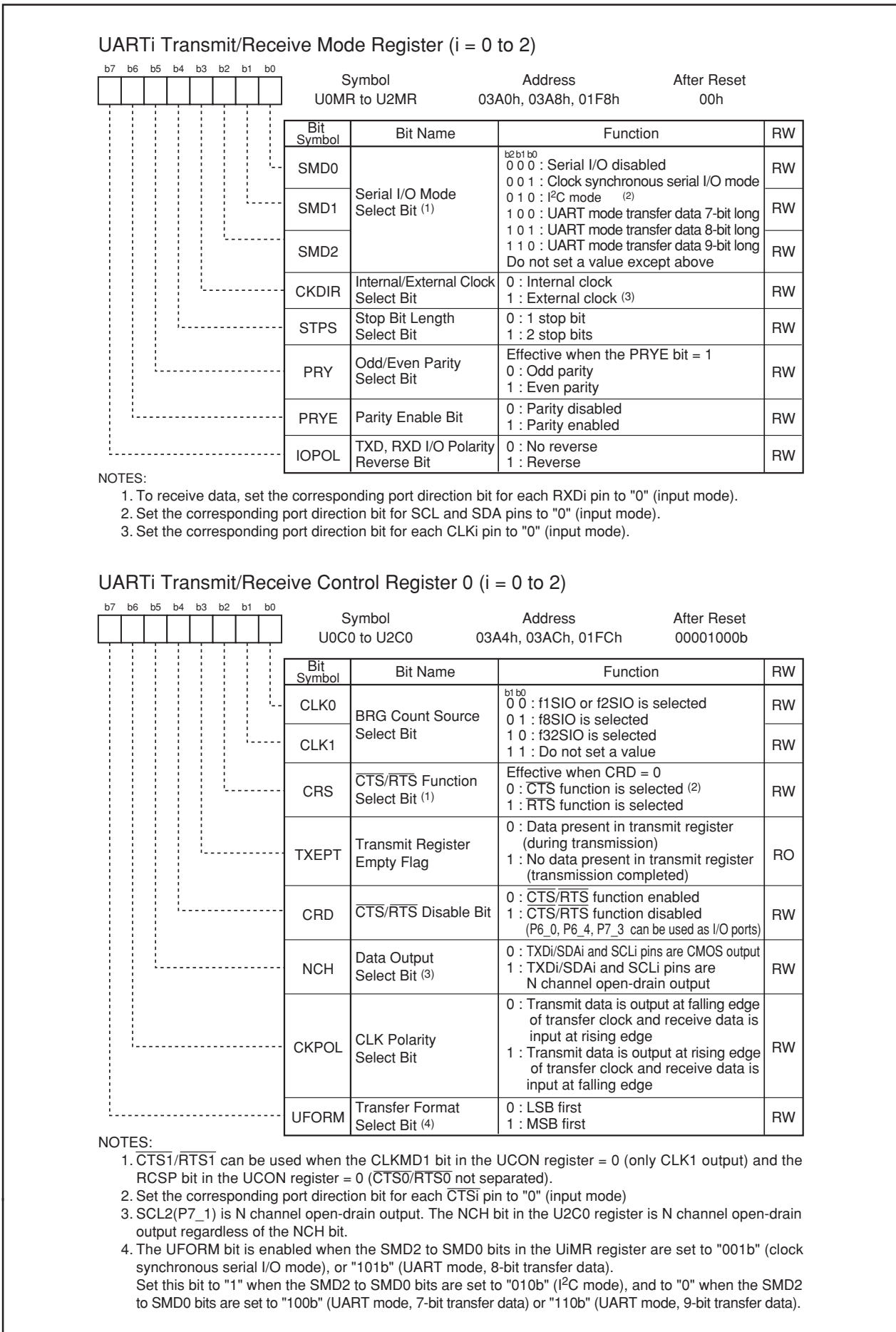


Figure 14.6 U0MR to U2MR Registers and U0C0 to U2C0 Registers

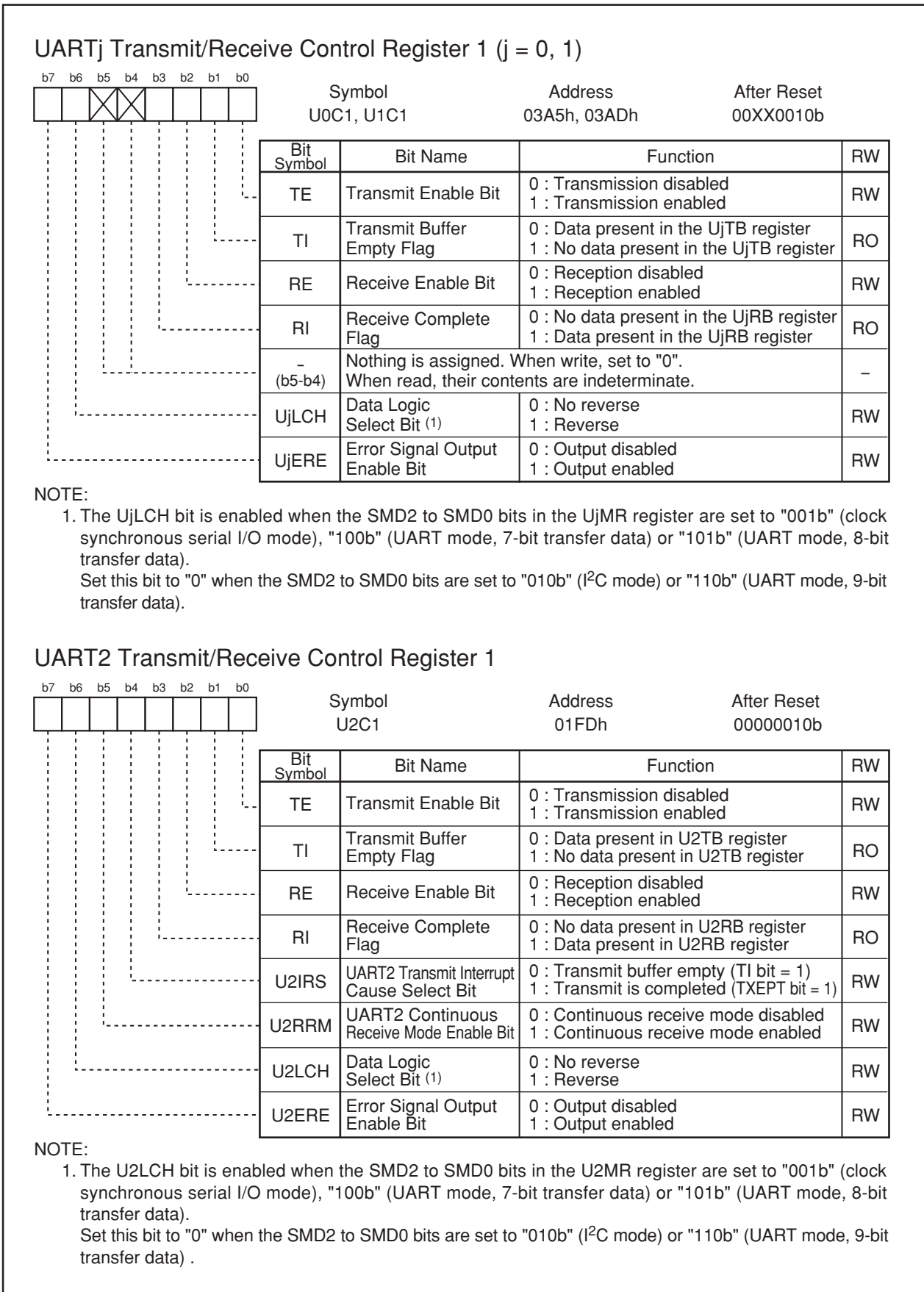


Figure 14.7 U0C1, U1C1 Registers and U2C1 Register

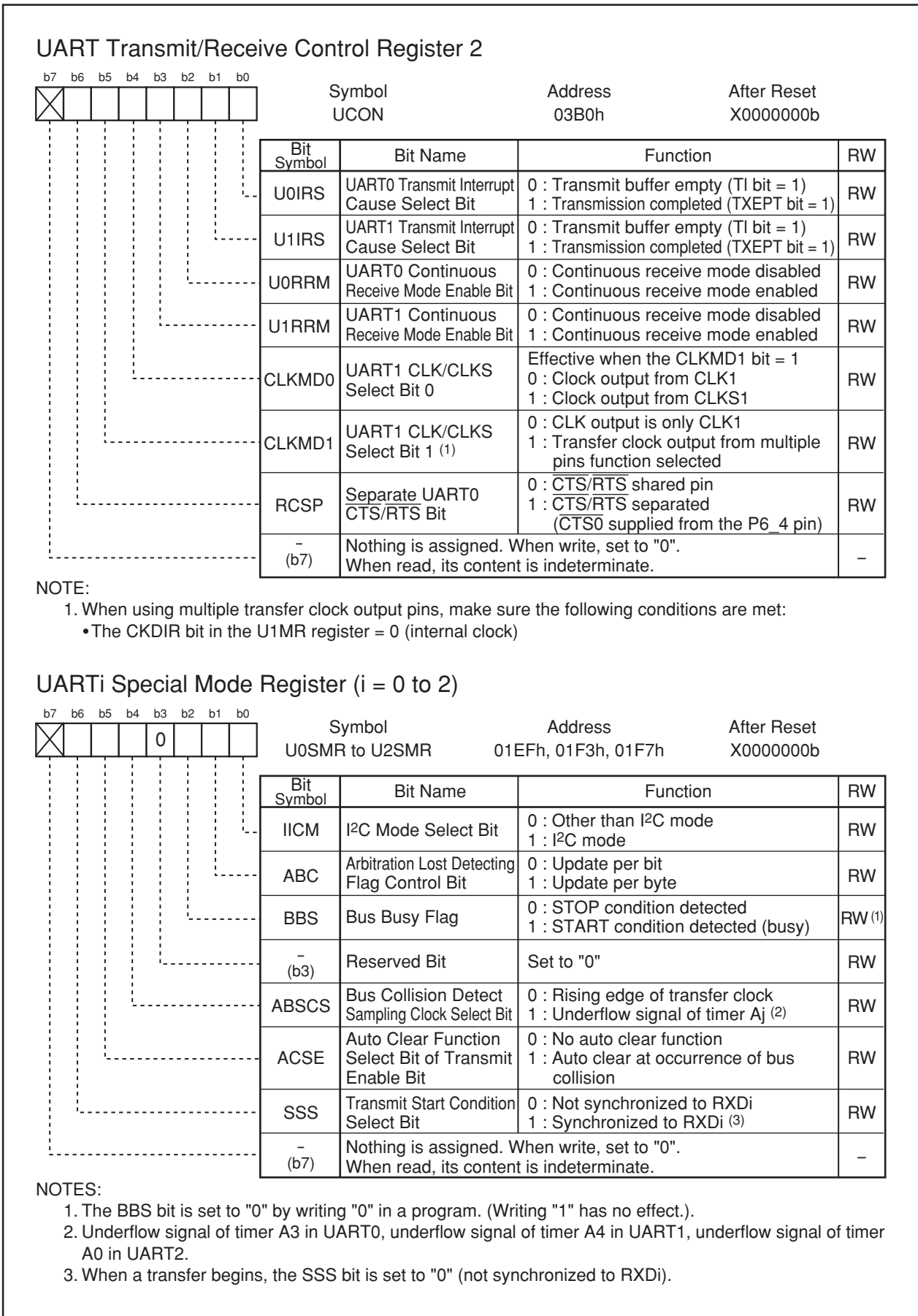


Figure 14.8 UCON Register and U0SMR to U2SMR Registers

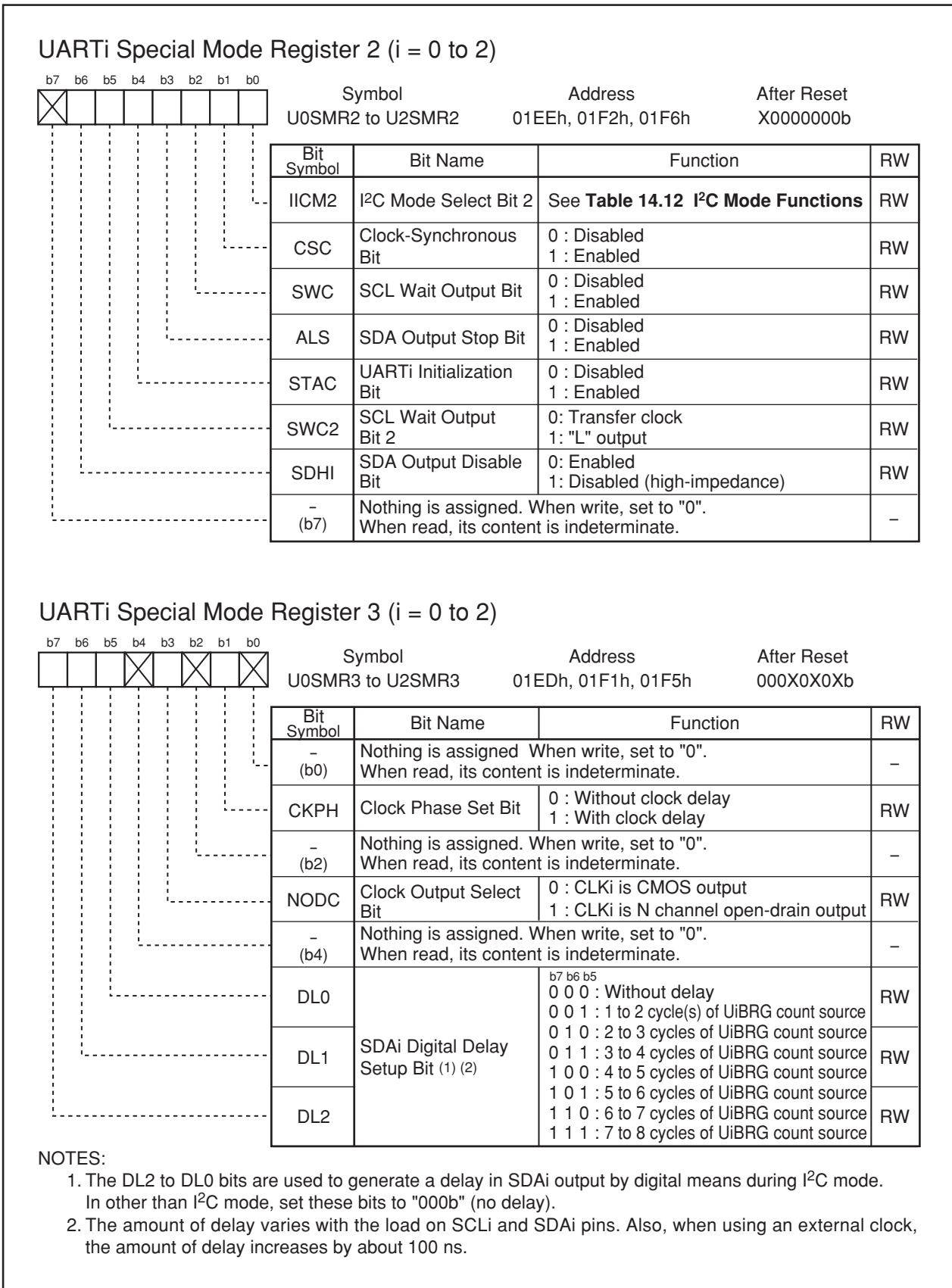


Figure 14.9 U0SMR2 to U2SMR2 Registers and U0SMR3 to U2SMR3 Registers

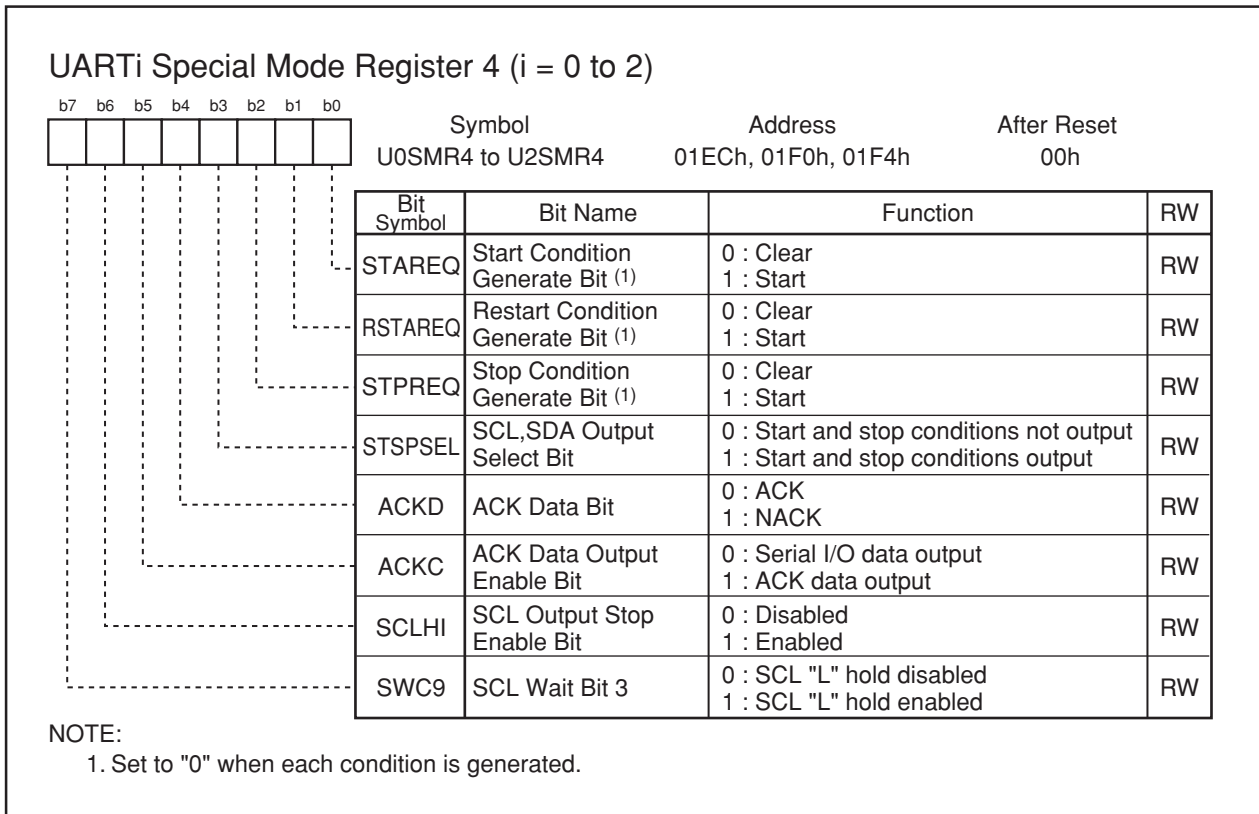


Figure 14.10 U0SMR4 to U2SMR4 Registers

14.1.1 Clock Synchronous Serial I/O Mode

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

Table 14.1 Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	The CKDIR bit in the UiMR register = 0 (internal clock) : $f_j / 2^{(n+1)}$ <ul style="list-style-type: none"> $f_j = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the UiBRG register 00h to FFh The CKDIR bit = 1 (external clock) : Input from CLKi pin
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 ⁽¹⁾ <ul style="list-style-type: none"> The TE bit in the UiC1 register = 1 (transmission enabled) The TI bit in the UiC1 register = 0 (data present in the UiTB register) If CTS function is selected, input on the \overline{CTS}_i pin = L
Reception Start Condition	Before reception can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> The RE bit in the UiC1 register = 1 (reception enabled) The TE bit in the UiC1 register = 1 (transmission enabled) The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> The UiIRS bit ⁽²⁾ = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) The UiIRS bit = 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register For reception <ul style="list-style-type: none"> When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error Detection	Overrun error ⁽³⁾ 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	<ul style="list-style-type: none"> CLK polarity selection Transfer data input/output can be selected to occur synchronously with the rising or the falling edge of the transfer clock LSB first, MSB first selection Whether to start 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:

- 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.
- 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.
- If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit in the SiRIC register does not change.

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

Register	Bit	Function
UiTB ⁽¹⁾	0 to 7	Set transmission data
UiRB ⁽¹⁾	0 to 7	Reception data can be read
	OER	Overflow error flag
UiBRG	0 to 7	Set a transfer rate
UiMR ⁽¹⁾	SMD2 to SMD0	Set to "001b"
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to "0"
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select CTS or RTS to use
	TXEPT	Transmit register empty flag
	CRD	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 ⁽²⁾	Select the source of UART2 transmit interrupt
	U2RRM ⁽²⁾	Set this bit to "1" to use continuous receive mode
	UiLCH	Set this bit to "1" to use inverted data logic
	UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 2	Set to "0"
	NODC	Select clock output mode
	4 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the 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 14.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Table 14.3 shows pin functions for the case where the multiple transfer clock output pin select function is deselected. Table 14.4 lists the P6_4 pin functions during clock synchronous serial I/O mode.

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

Figure 14.11 shows the transmit/receive timings during clock synchronous serial I/O mode.

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

Pin Name	Function	Method of Selection
TXD _i (P6_3, P6_7, P7_0)	Serial Data Output	(Outputs dummy data when performing reception only)
RXD _i (P6_2, P6_6, P7_1)	Serial Data Input	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 _i (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 _i /RTS _i (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
	I/O Port	CRD bit = 1

i = 0 to 2

Table 14.4 P6_4 Pin Functions

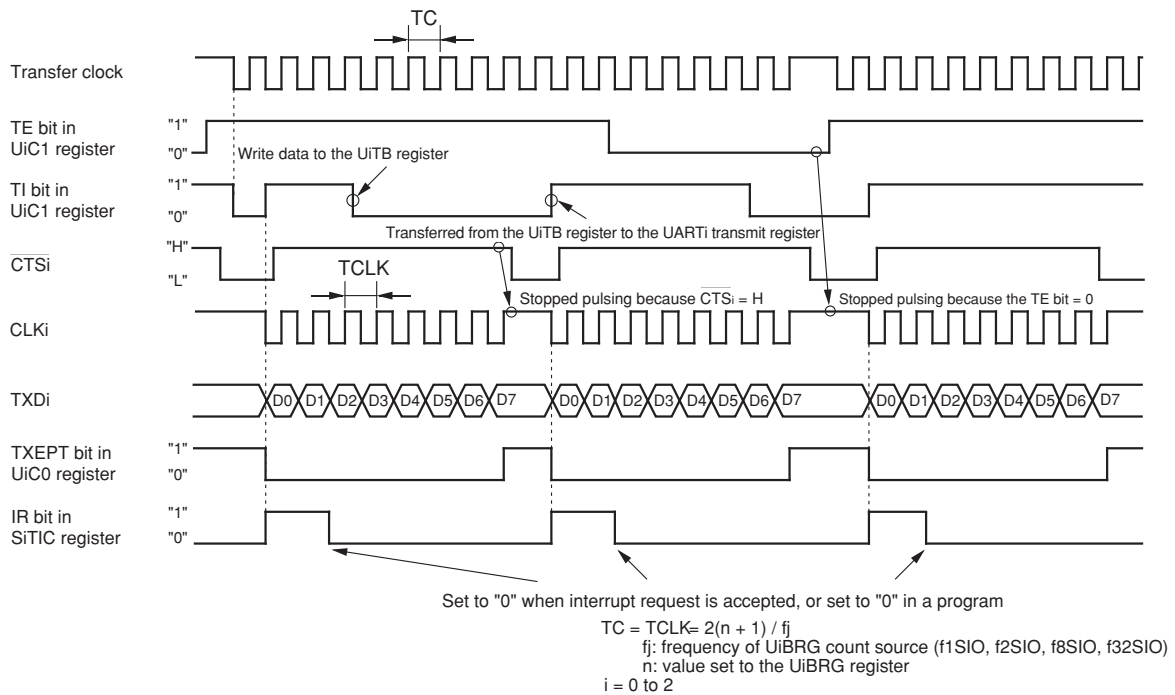
Pin Function	Bit set Value					
	U1C0 Register		UCON Register			PD6 Register
	CRD bit	CRS bit	RCSP bit	CLKMD1 bit	CLKMD0 bit	PD6_4 bit
P6_4	1	-	0	0	-	Input: 0, Output: 1
CTS ₁	0	0	0	0	-	0
RTS ₁	0	1	0	0	-	-
CTS ₀ ⁽¹⁾	0	0	1	0	-	0
CLKS ₁	-	-	-	1 ⁽²⁾	1	-

-: “0” or “1”

NOTES:

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

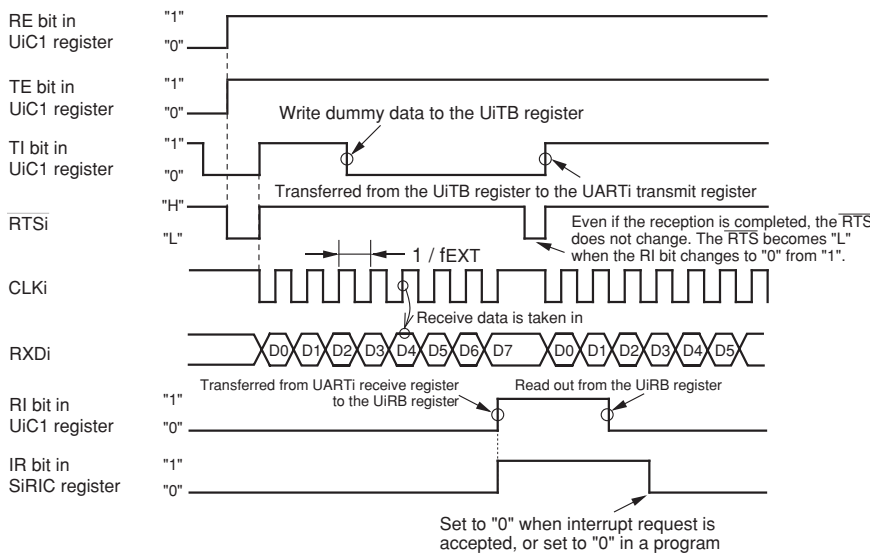
(1) Example of Transmit Timing (when internal clock is selected)



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 (CTS 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 (RTS 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)

- 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

fEXT: frequency of external clock

Figure 14.11 Transmit and Receive Operation

14.1.1.1 Counter Measure for Communication Error Occurs

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

- Resetting the UiRB register (i = 0 to 2)
 - (1) Set the RE bit in the UiC1 register to “0” (reception disabled)
 - (2) Set the SMD2 to SMD0 bits in the UiMR register to “000b” (serial I/O 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 I/O 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

14.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register (i = 0 to 2) to select the transfer clock polarity. Figure 14.12 shows the polarity of the transfer clock.

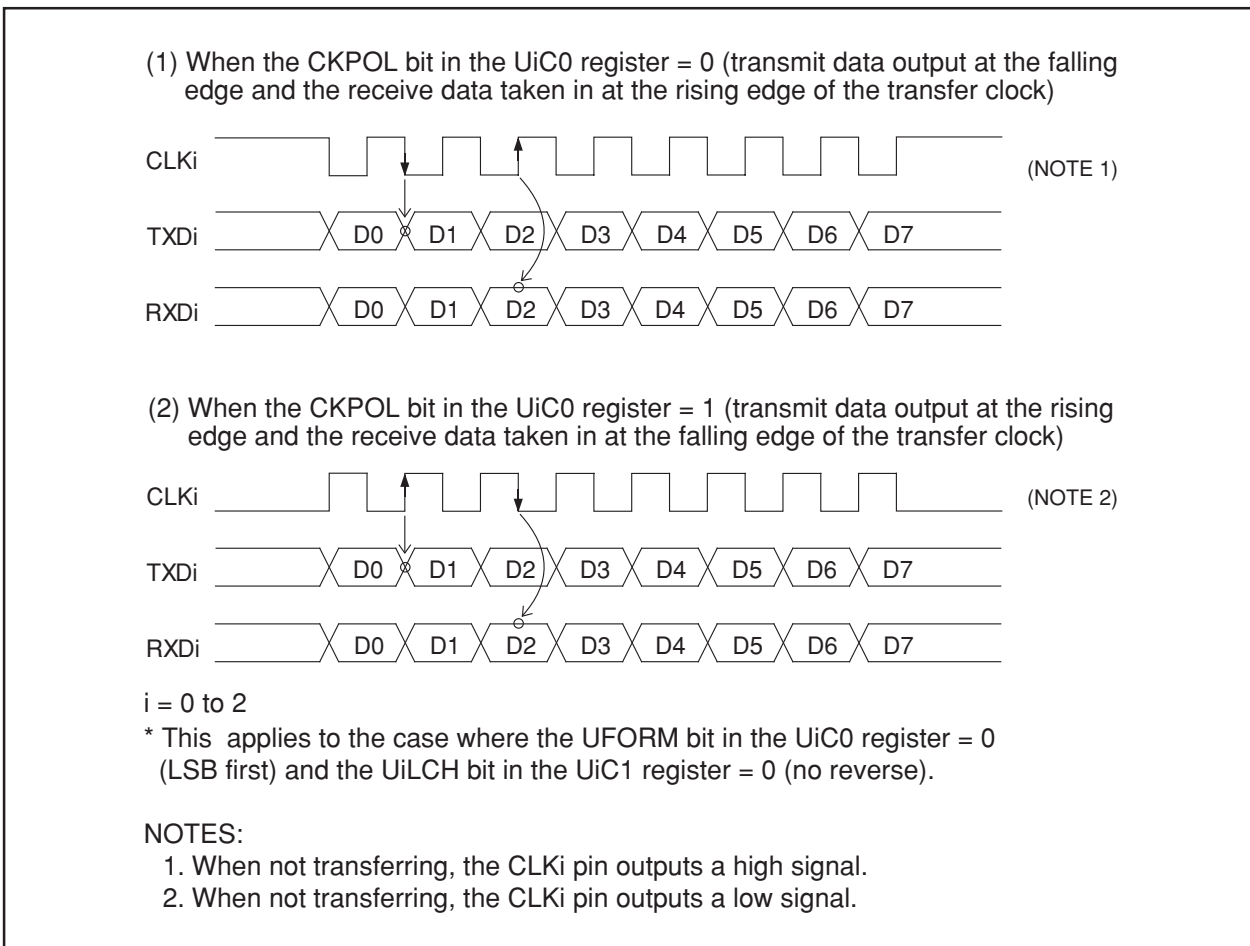


Figure 14.12 Transfer Clock Polarity

14.1.1.3 LSB First/MSB First Select Function

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

Figure 14.13 shows the transfer format.

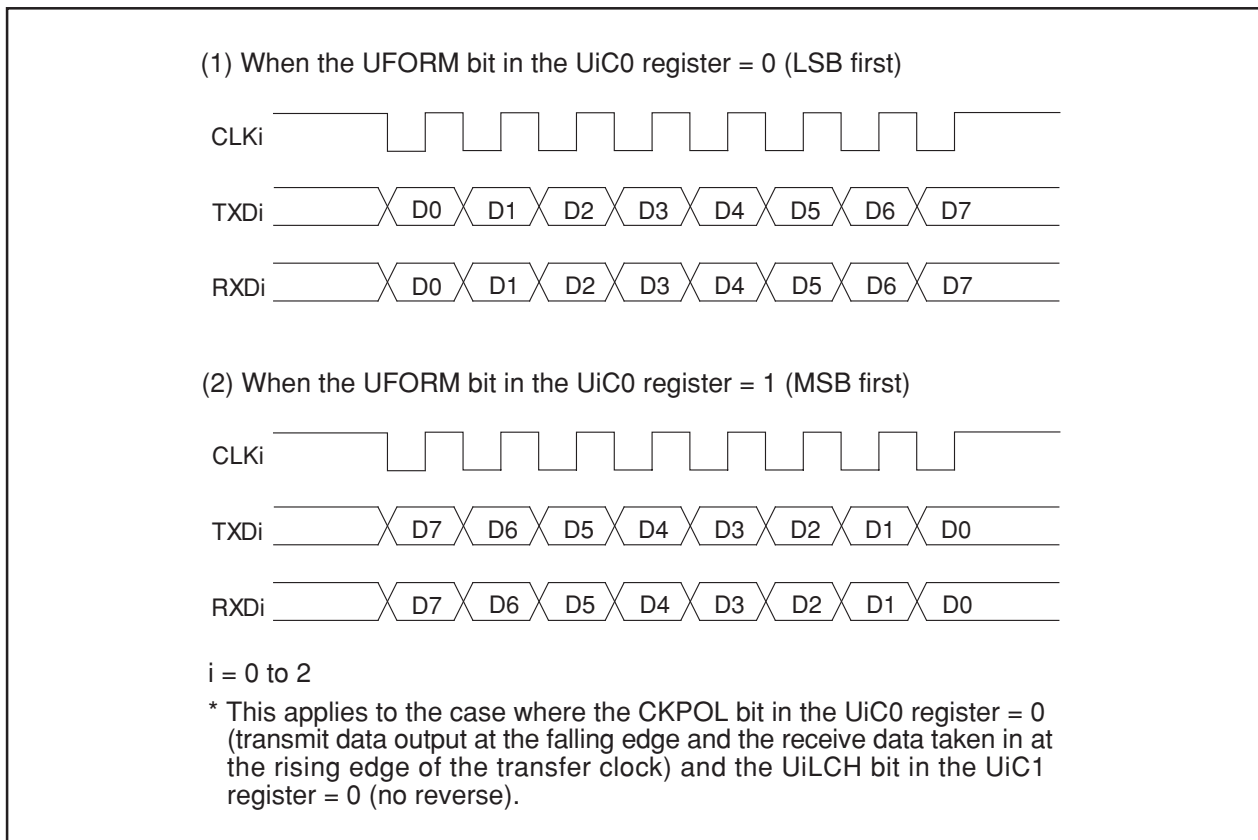


Figure 14.13 Transfer Format

14.1.1.4 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.

14.1.1.5 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 14.14 shows serial data logic.

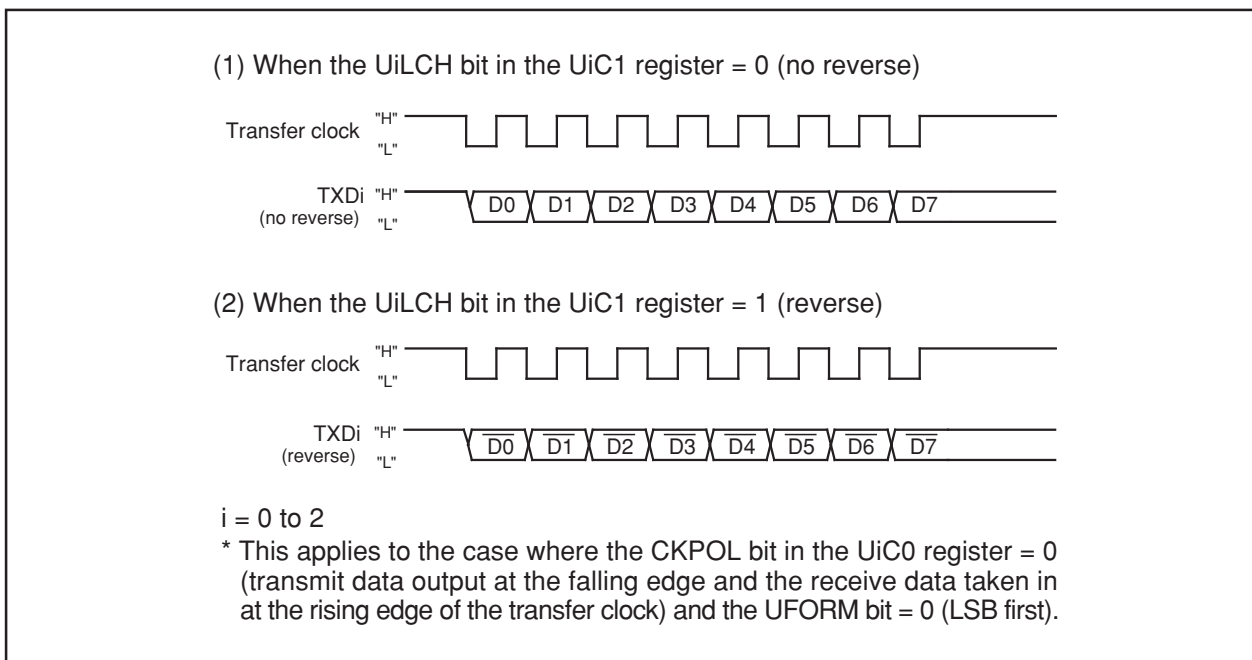


Figure 14.14 Serial Data Logic Switching

14.1.1.6 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 14.15 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.

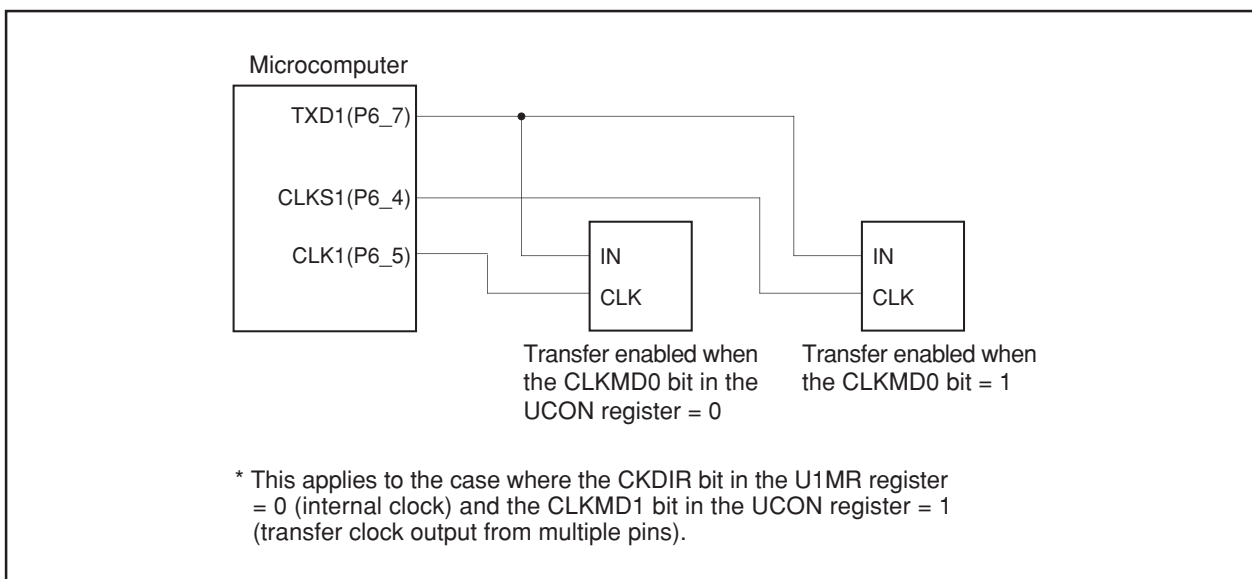


Figure 14.15 Transfer Clock Output From Multiple Pins

14.1.1.7 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Function

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

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

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

14.1.1.8 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function (UART0)

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

- CRD bit in U0C0 register = 0 (enables UART0 $\overline{\text{CTS}}/\overline{\text{RTS}}$)
- CRS bit in U0C0 register = 1 (outputs UART0 $\overline{\text{RTS}}$)
- 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 $\overline{\text{CTS}}_0$ from the P6_4 pin)
- CLKMD1 bit in UCON register = 0 (CLKS1 not used)

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

Figure 14.16 shows $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function usage.

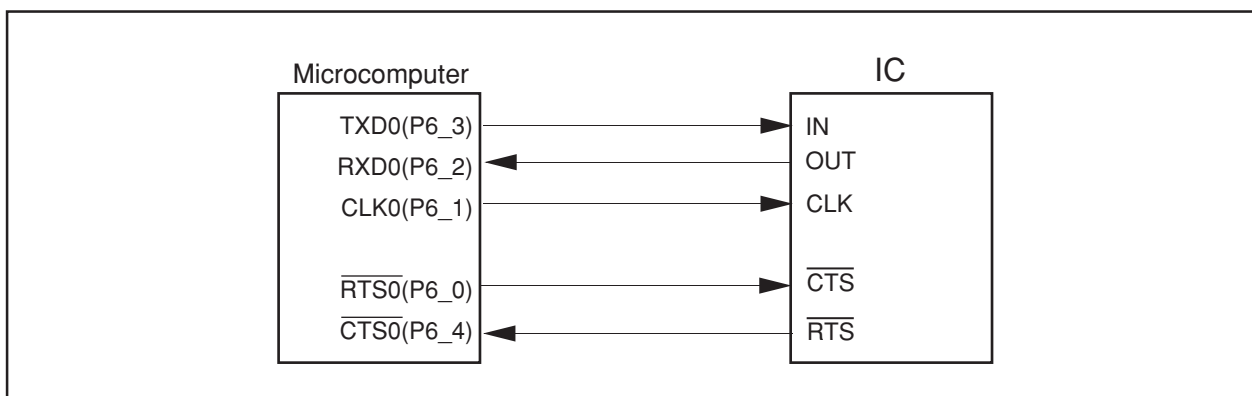


Figure 14.16 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

14.1.2 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 14.5 lists the specifications of the UART mode. Table 14.6 lists the registers used in UART mode and the register values set.

Table 14.5 UART Mode Specifications

Item	Specification
Transfer Data Format	<ul style="list-style-type: none"> Character bit (transfer data): Selectable from 7, 8 or 9 bits Start bit: 1 bit Parity bit: Selectable from odd, even, or none Stop bit: Selectable from 1 or 2 bits
Transfer Clock	<ul style="list-style-type: none"> CKDIR bit in UiMR register = 0 (internal clock) : $f_j / 16(n+1)$ $f_j = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the UiBRG register 00h to FFh The CKDIR bit = 1 (external clock) : $fEXT/16(n+1)$ $fEXT$: Input from CLKi pin. n: Setting value of the UiBRG register 00h to FFh
Transmission, Reception Control	Selectable from \overline{CTS} function, RTS function or \overline{CTS}/RTS function disabled
Transmission Start Condition	<p>Before transmission can start, the following requirements must be met</p> <ul style="list-style-type: none"> The TE bit in the UiC1 register = 1 (transmission enabled) The TI bit in the UiC1 register = 0 (data present in UiTB register) If \overline{CTS} function is selected, input on the \overline{CTS}_i pin = L
Reception Start Condition	<p>Before reception can start, the following requirements must be met</p> <ul style="list-style-type: none"> The RE bit in the UiC1 register = 1 (reception enabled) Start bit detection
Interrupt Request Generation Timing	<p>For transmission, one of the following conditions can be selected</p> <ul style="list-style-type: none"> The UiIRS bit ⁽¹⁾ = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) The UiIRS bit = 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register <p>For reception</p> <ul style="list-style-type: none"> When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error Detection	<ul style="list-style-type: none"> Overrun error ⁽²⁾ This error occurs if the serial 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 Framing error ⁽³⁾ This error occurs when the number of stop bits set is not detected Parity error ⁽³⁾ This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set Error sum flag This flag is set to "1" when any of the overrun, framing, or parity errors occur
Select Function	<ul style="list-style-type: none"> LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected Serial data logic switch This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. TXD, RXD I/O polarity switch This function reverses the polarities of the TXD pin output and RXD pin input. The logic levels of all I/O data is reversed. Separate \overline{CTS}/RTS pins (UART0) \overline{CTS}_0 and RTS_0 are input/output from separate pins

i = 0 to 2

NOTES:

- The U0IRS and U1IRS bits are bits 0 and 1 in the UCON register. The U2IRS bit is bit 4 in the U2C1 register.
- If an overrun error occurs, the value of the UiRB register will be indeterminate. The IR bit in the SiRIC register does not change.
- 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 14.6 Registers to Be Used and Settings in UART Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data ⁽¹⁾
UiRB	0 to 8	Reception data can be read ⁽¹⁾
	OER,FER,PER,SUM	Error flag
UiBRG	0 to 7	Set 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 ⁽²⁾	Select the source of UART2 transmit interrupt
	U2RRM ⁽²⁾	Set to "0"
	UiLCH	Set this bit to "1" to use inverted data logic
	UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the 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:

1. 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.

2. 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 14.7 lists the functions of the input/output pins during UART mode. Table 14.8 lists the P6_4 pin functions during UART mode. Note that for a period from when the UART_i operation mode is selected to when transfer starts, the TXD_i pin outputs an “H”.

Figure 14.17 shows the typical transmit timings in UART mode. Figure 14.18 shows the typical receive timing in UART mode.

Table 14.7 I/O Pin Functions

Pin Name	Function	Method of Selection
TXD _i (P6_3, P6_7, P7_0)	Serial Data Output	(Outputs “H” when performing reception only)
RXD _i (P6_2, P6_6, P7_1)	Serial Data Input	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 _i (P6_1, P6_5, P7_2)	I/O Port	CKDIR bit in UiMR register = 0
	Transfer Clock Input	CKDIR bit in UiMR register = 1 PD6_1 and PD6_5 bits in PD6 register = 0 PD7_2 bit in PD7 register = 0
CTS _i /RTS _i (P6_0, P6_4, P7_3)	CTS Input	CRD bit in UiC0 register = 0 CRS bit in UiC0 register = 0 PD6_0 and PD6_4 bits in PD6 register = 0 PD7_3 bit in PD7 register = 0
		RTS Output
	I/O Port	CRD bit = 1

i = 0 to 2

Table 14.8 P6_4 Pin Functions

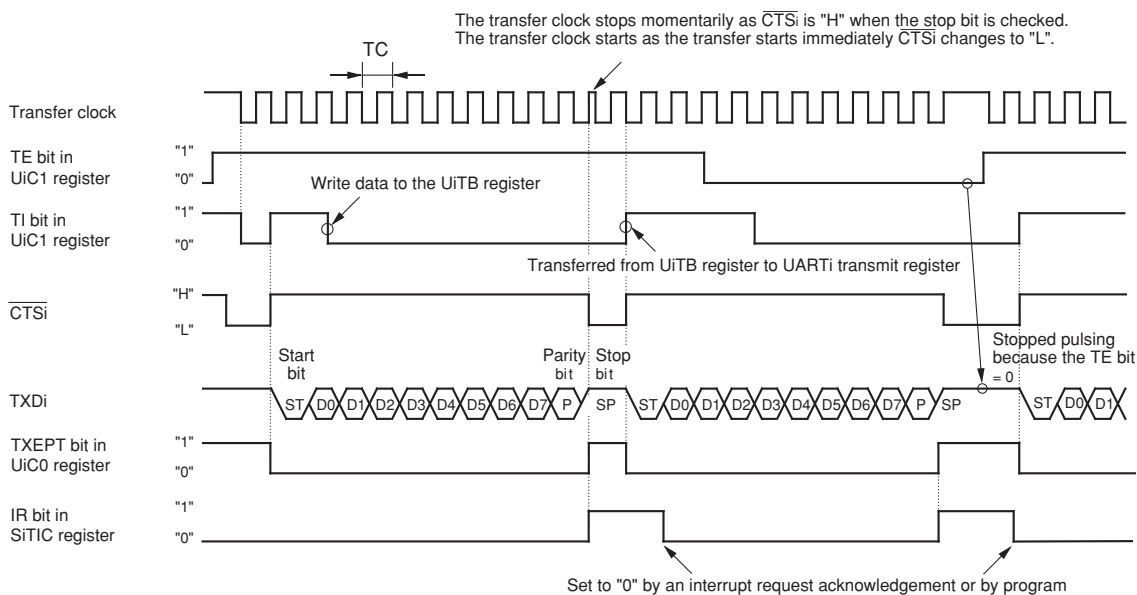
Pin Function	Bit set Value				
	U1C0 Register		UCON Register		PD6 Register
	CRD bit	CRS bit	RCSP bit	CLKMD1 bit	PD6_4 bit
P6_4	1	-	0	0	Input: 0, Output: 1
CTS ₁	0	0	0	0	0
RTS ₁	0	1	0	0	-
CTS ₀ ⁽¹⁾	0	0	1	0	0

-: “0” or “1”

NOTE:

- 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)



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

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

$TC = 16(n + 1) / f_j$ or $16(n + 1) / f_{EXT}$

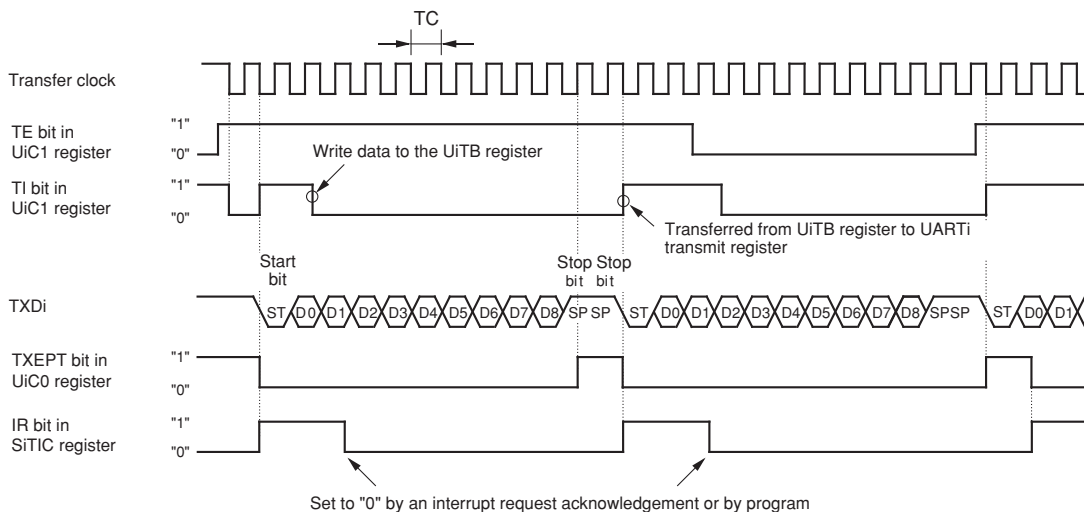
f_j : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)

f_{EXT} : frequency of UiBRG count source (external clock)

n : value set to UiBRG

$i = 0$ to 2

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



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

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

$TC = 16(n + 1) / f_j$ or $16(n + 1) / f_{EXT}$

f_j : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)

f_{EXT} : frequency of UiBRG count source (external clock)

n : value set to UiBRG

$i = 0$ to 2

Figure 14.17 Transmit Operation

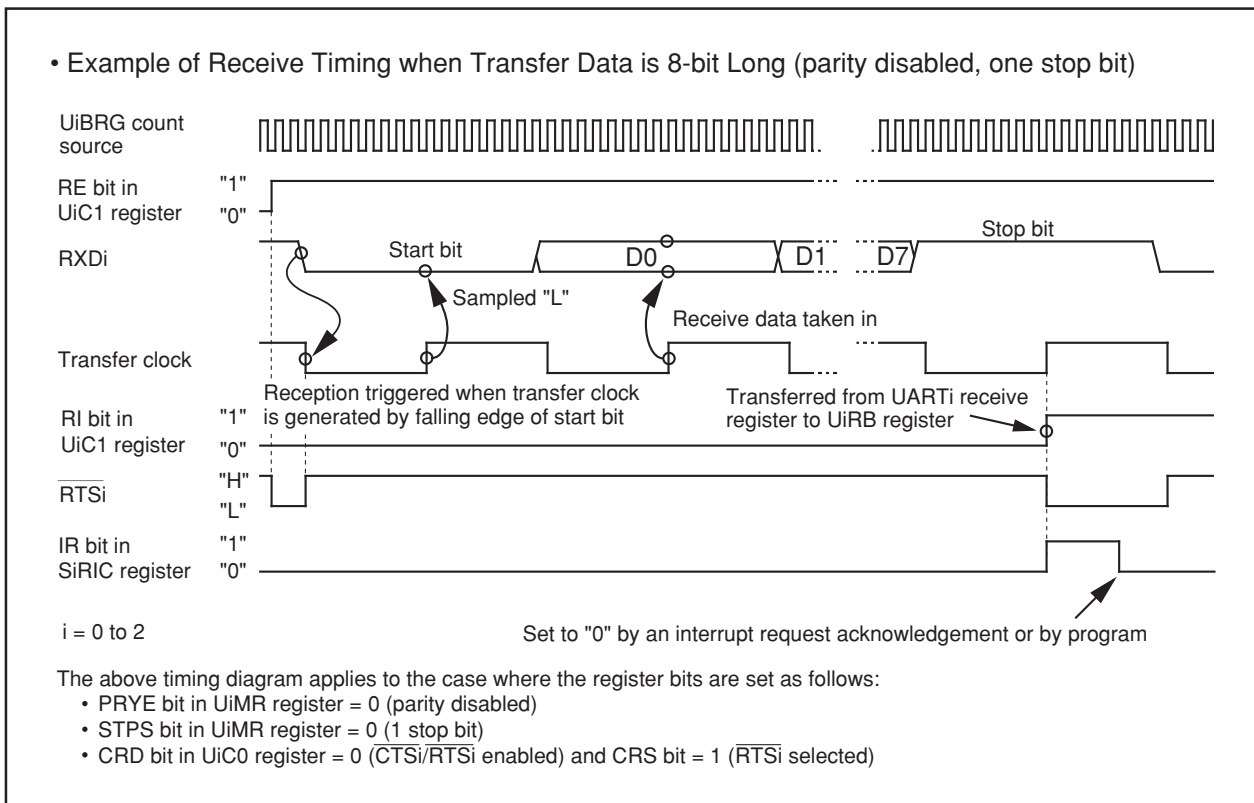


Figure 14.18 Receive Operation

14.1.2.1 Bit Rates

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

Table 14.9 Example of Bit Rates and Settings

Bit-rate (bps)	Count source of BRG	Peripheral function clock: 16MHz		Peripheral function clock: 24MHz	
		Set value of BRG: n	Actual time (bps)	Set value of BRG: n	Actual time (bps)
1200	f8	103 (67h)	1202	155 (9Bh)	1202
2400	f8	51 (33h)	2404	77 (4Dh)	2404
4800	f8	25 (19h)	4808	38 (26h)	4808
9600	f1	103 (67h)	9615	155 (9Bh)	9615
14400	f1	68 (44h)	14493	103 (67h)	14423
19200	f1	51 (33h)	19231	77 (4Dh)	19231
28800	f1	34 (22h)	28571	51 (33h)	28846
31250	f1	31 (1Fh)	31250	47 (2Fh)	31250
38400	f1	25 (19h)	38462	38 (26h)	38462
51200	f1	19 (13h)	50000	28 (1Ch)	51724

14.1.2.2 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 I/O 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

14.1.2.3 LSB First/MSB First Select Function

As shown in Figure 14.19, use the UFORM bit in the UiC0 register to select the transfer format. This function is valid when transfer data is 8-bit long.

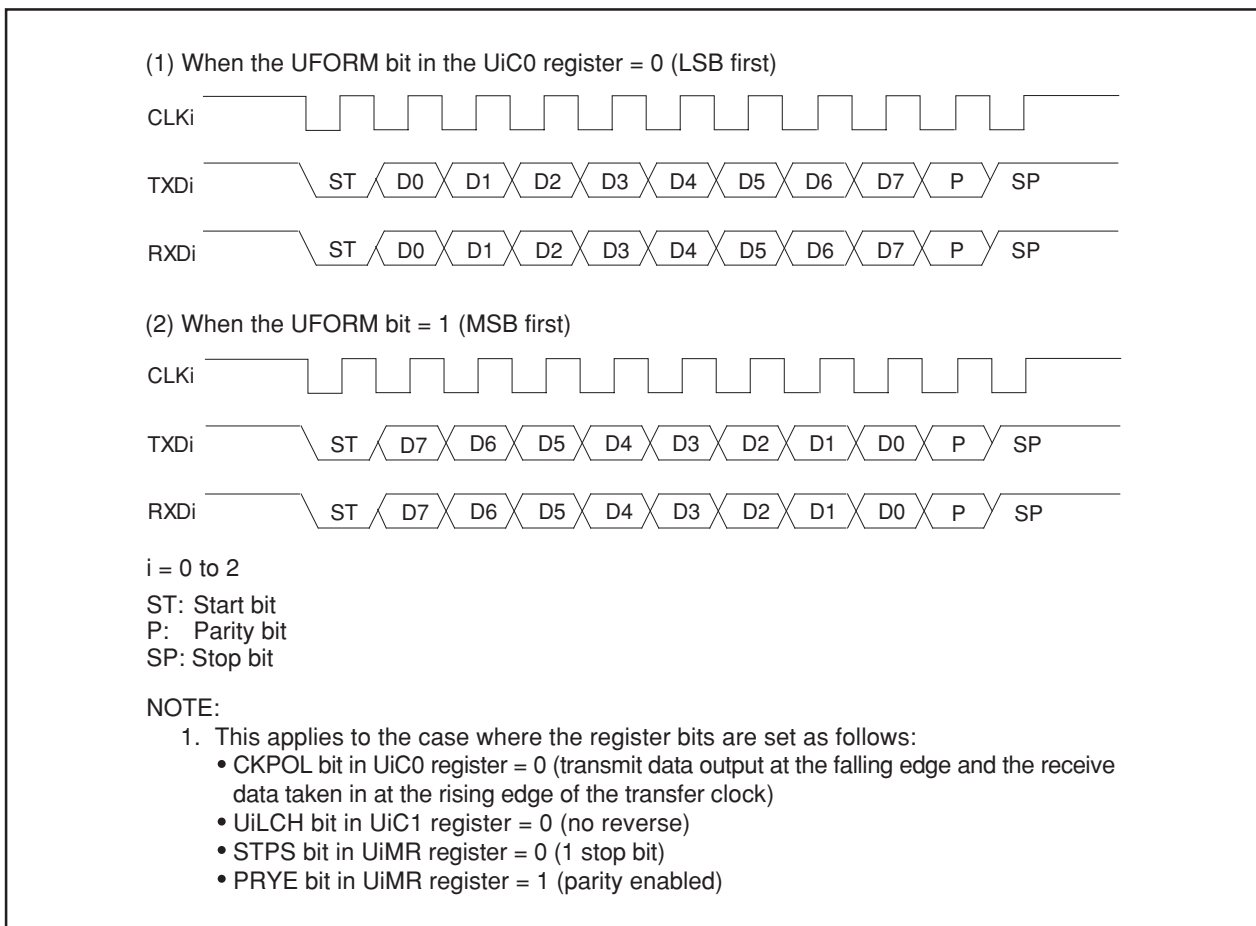


Figure 14.19 Transfer Format

14.1.2.4 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 14.20 shows serial data logic.

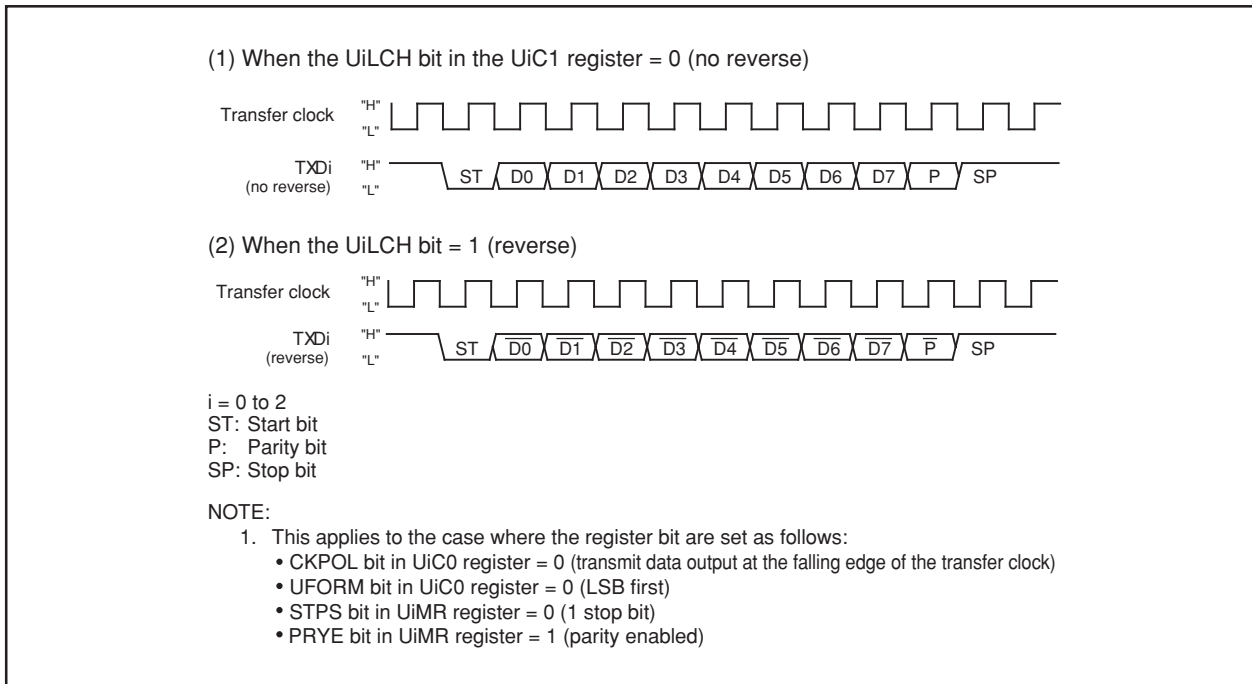


Figure 14.20 Serial Data Logic Switching

14.1.2.5 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 14.21 shows the TXD and RXD input/output polarity inverse.

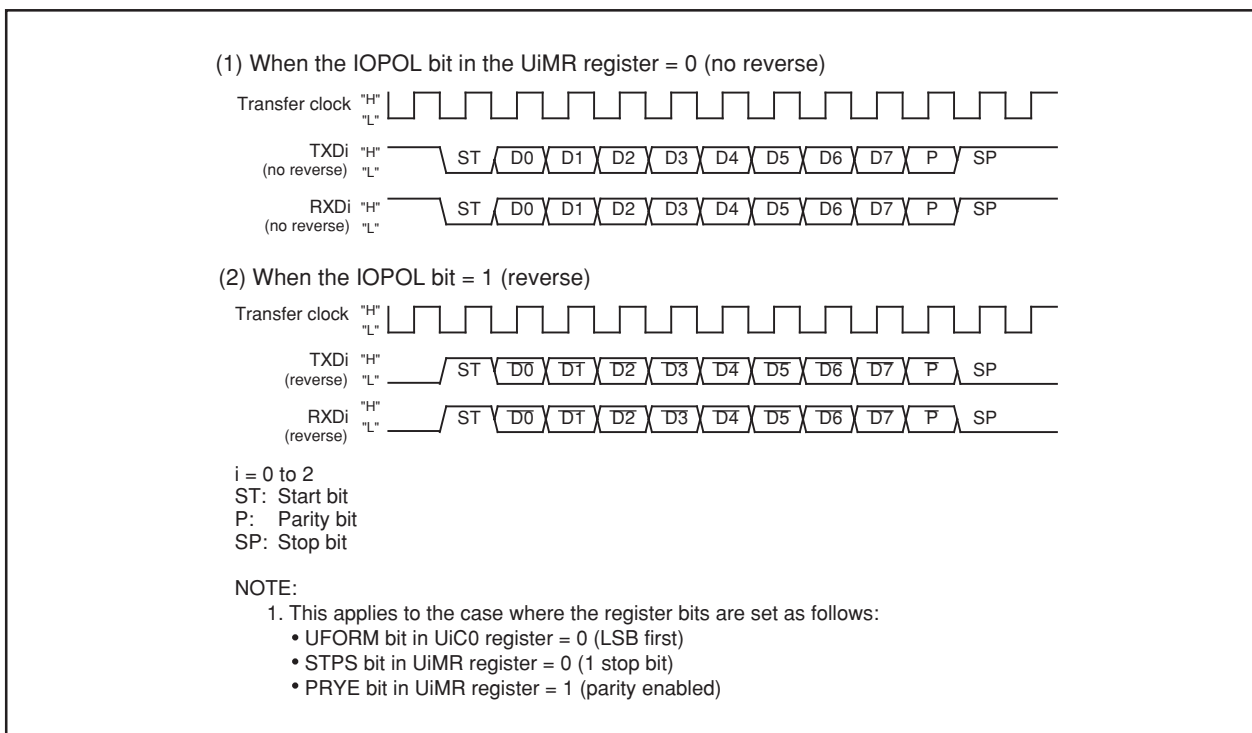


Figure 14.21 TXD and RXD I/O Polarity Inverse

14.1.2.6 CTS/RTS Function

When the CTS function is used transmit operation start when “L” is applied to the CTSi/RTSi (i = 0 to 2) pin. Transmit operation begins when the CTSi/RTSi 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 RTS function is used, the CTSi/RTSi pin outputs on “L” signal when the microcomputer is ready to receive. The output level becomes “H” on the first falling edge of the CLKi pin.

- CRD bit in UiC0 register = 1 (disables UART0 CTS/RTS function) CTSi/RTSi pin is programmable I/O function
- CRD bit = 0, CRS bit in UiC0 register= 0 (CTS function is selected) CTSi/RTSi pin is CTS function
- CRD bit = 0, CRS bit = 1 (RTS function is selected) CTSi/RTSi pin is RTS function

14.1.2.7 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 CTS/RTS)
- CRS bit in U0C0 register = 1 (outputs UART0 RTS)
- CRD bit in U1C0 register = 0 (enables UART1 CTS/RTS)
- CRS bit in U1C0 register = 0 (inputs UART1 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 14.22 shows CTS/RTS separate function usage.

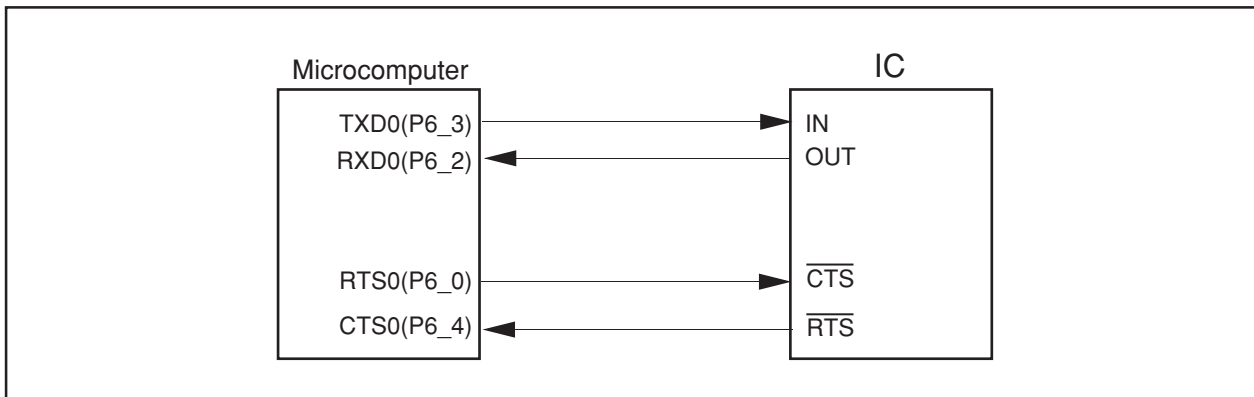


Figure 14.22 CTS/RTS Separate Function

14.1.3 Special Mode 1 (I²C Mode)

I²C mode is provided for use as a simplified I²C interface compatible mode. Table 14.10 lists the specifications of the I²C mode. Figure 14.23 shows the block diagram for I²C mode. Table 14.11 lists the registers used in the I²C mode and the register values set. Table 14.12 lists the functions in I²C mode. Figure 14.24 shows the transfer to the UiRB register and interrupt timing.

As shown in Table 14.12, the microcomputer is placed in I²C 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 14.10 I²C Mode Specifications

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	<ul style="list-style-type: none"> • During master The CKDIR bit in the UiMR register = 0 (internal clock) : $f_j / 2^{(n+1)}$ $f_j = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the UiBRG register 00h to FFh • During slave The CKDIR bit = 1 (external clock) : Input from SCLi pin
Transmission Start Condition	Before transmission can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Reception Start Condition	Before reception can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> • The RE bit in the UiC1 register = 1 (reception enabled) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Interrupt Request Generation Timing	When start or stop condition is detected, acknowledge undetected, and acknowledge detected
Error Detection	Overrun error ⁽²⁾ This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 8th bit of the next data
Select Function	<ul style="list-style-type: none"> • Arbitration lost Timing at which the ABT bit in the UiRB register is updated can be selected • SDAi digital delay No digital delay or a delay of 2 to 8 UiBRG count source clock cycles selectable • Clock phase setting With or without clock delay selectable

i = 0 to 2

NOTES:

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

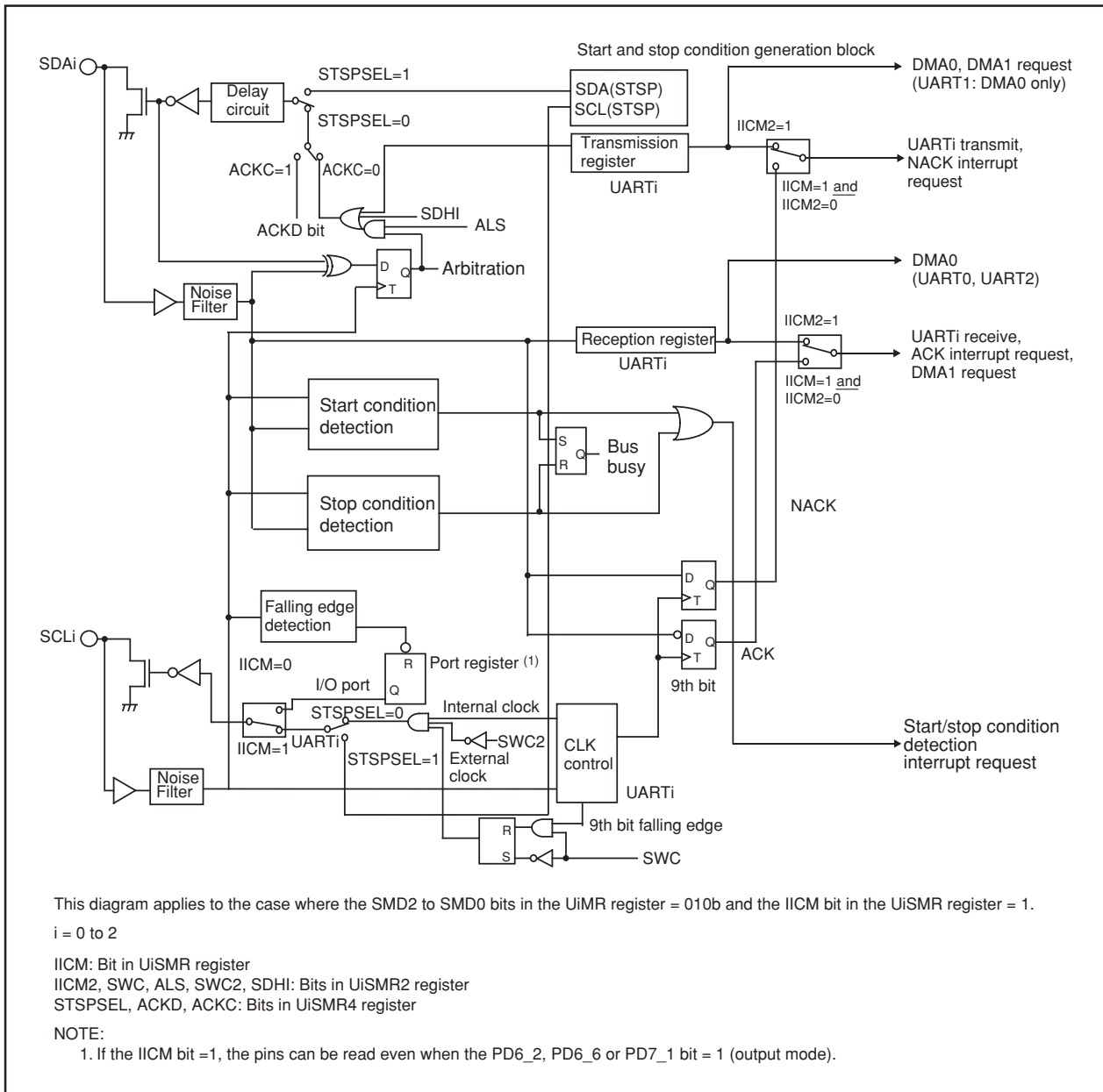


Figure 14.23 I²C Mode Block Diagram

Table 14.11 Registers to Be Used and Settings in I²C 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 Table 14.12 I²C Mode Functions	
	CSC	Set this bit to "1" to enable clock synchronization	Set to "0"
	SWC	Set this bit to "1" to have SCLi output fixed to "L" at the falling edge of the 9th bit of clock	
	ALS	Set this bit to "1" to have SDAi output stopped when arbitration-lost is detected	Set to "0"
	STAC	Set to "0"	Set this bit to "1" to initialize UARTi at start condition detection
	SWC2	Set this bit to "1" to have SCLi output forcibly pulled low	
	SDHI	Set this bit to "1" to disable SDAi output	
	7	Set to "0"	
UiSMR3	0, 2, 4 and NODC	Set to "0"	
	CKPH	See Table 14.12 I²C Mode Functions	
	DL2 to DL0	Set the amount of SDAi digital delay	
UiSMR4	STAREQ	Set this bit to "1" to generate start condition	Set to "0"
	RSTAREQ	Set this bit to "1" to generate restart condition	Set to "0"
	STPREQ	Set this bit to "1" to generate stop condition	Set to "0"
	STPSEL	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 I²C 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 14.12 I²C Mode Functions

Function	Clock Synchronous Serial I/O Mode (SMD2 to SMD0 = 001b, IICM = 0)	I ² C Mode (SMD2 to SMD0 = 010b, IICM = 1)			
		IICM2 = 0 (NACK/ACK interrupt)		IICM2 = 1 (UART transmit/receive interrupt)	
		CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)	CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)
Factor of Interrupt Number 6, 7 and 10 ^{(1) (5) (7)}	-	Start condition detection or stop condition detection (See Table 14.13 STSPSEL Bit Functions)			
Factor of Interrupt Number 15, 17 and 19 ^{(1) (6)}	UARTi transmission Transmission started or completed (selected by UiIRS)	No acknowledgment detection (NACK) Rising edge of SCLi 9th bit	UARTi transmission Rising edge of SCLi 9th bit	UARTi transmission Falling edge of SCLi next to the 9th bit	
Factor of Interrupt Number 16, 18 and 20 ^{(1) (6)}	UARTi reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) Rising edge of SCLi 9th bit	UARTi reception Falling edge of SCLi 9th bit		
Timing for Transferring Data from UART Reception Shift Register to UiRB Register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Rising edge of SCLi 9th bit	Falling edge of SCLi 9th bit	Falling and rising edges of SCLi 9th bit	
UARTi Transmission Output Delay	Not delayed	Delayed			
Functions of 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 ² C mode)			
Noise Filter Width	15 ns	200 ns			
Read RXDi and SCLi Pins Levels	Possible when the corresponding port direction bit = 0	Always possible no matter how the corresponding port direction bit is set			
Initial Value of TXDi and SDAi Outputs	CKPOL = 0 (H) CKPOL = 1 (L)	The value set in the port register before setting I ² C mode ⁽²⁾			
Initial and End Value of SCLi	-	H	L	H	L
DMA1 Factor ⁽⁶⁾	UARTi reception	Acknowledgment detection (ACK)	UARTi reception Falling edge of SCLi 9th bit		
Store Received Data	1st to 8th bits of the received data are stored into 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 ⁽³⁾
Read Received Data	The UiRB register status is read			Bit 6 to bit 0 in the UiRB register ⁽⁴⁾ are read as bit 7 to bit 1. Bit 8 in the UiRB register is read as bit 0.	

i = 0 to 2

NOTES:

- If the 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 22.7 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 I/O 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 14.26 STSPSEL Bit Functions.
- See Figure 14.24 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).

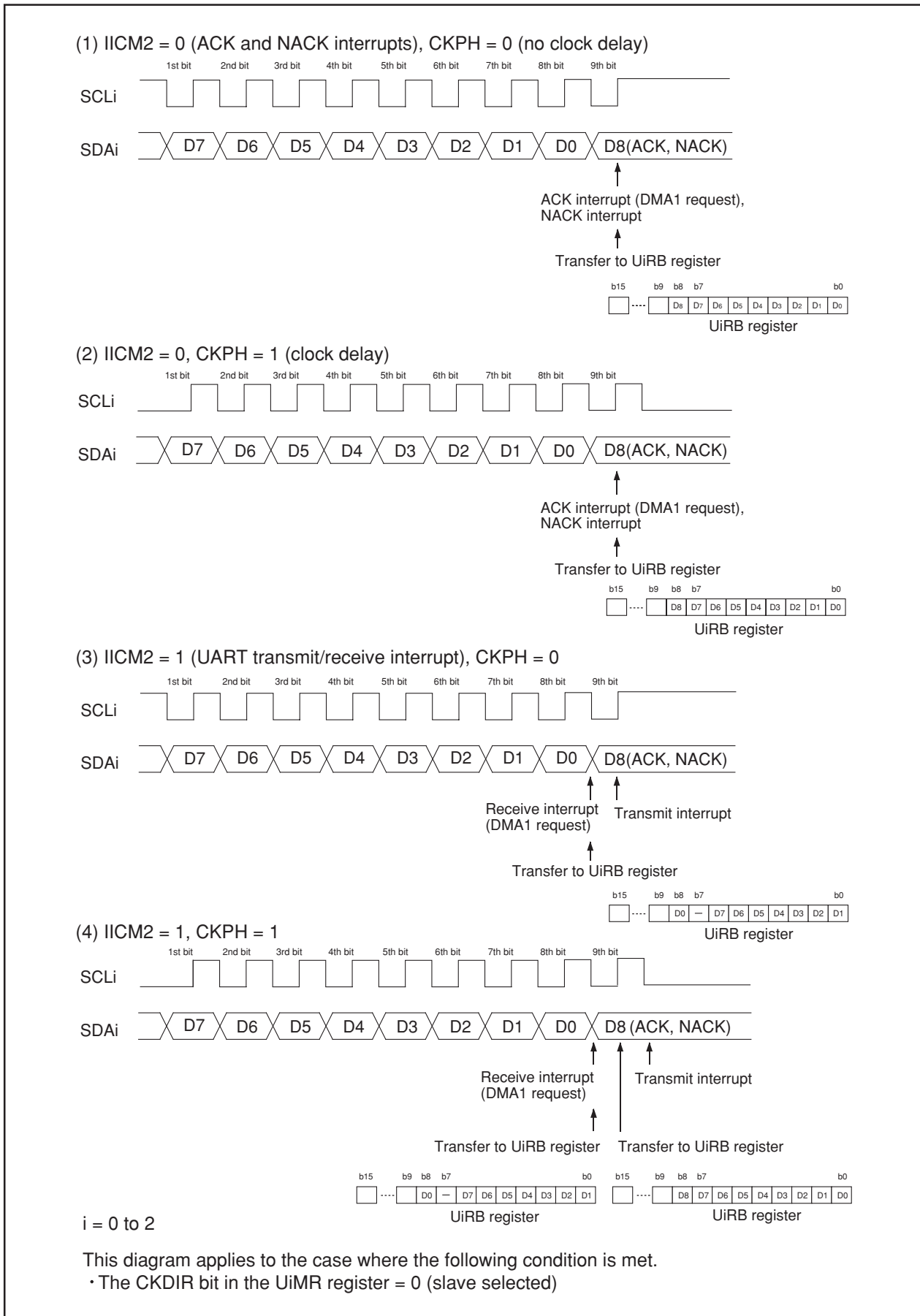


Figure 14.24 Transfer to UIRB Register and Interrupt Timing

14.1.3.1 Detection of Start and Stop Condition

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

A start condition-detected interrupt request is generated when the 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 14.25 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.

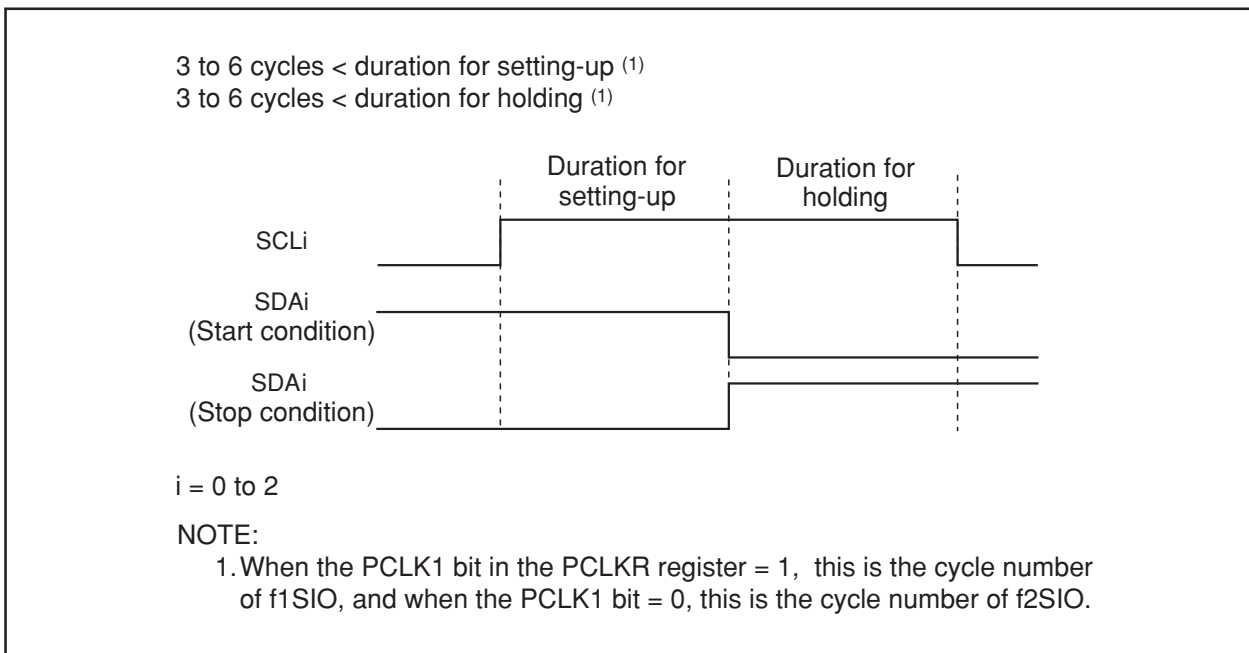


Figure 14.25 Detection of Start and Stop Condition

14.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the STAREQ bit in the 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 14.13 and Figure 14.26 show the functions of the STSPSEL bit.

Table 14.13 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

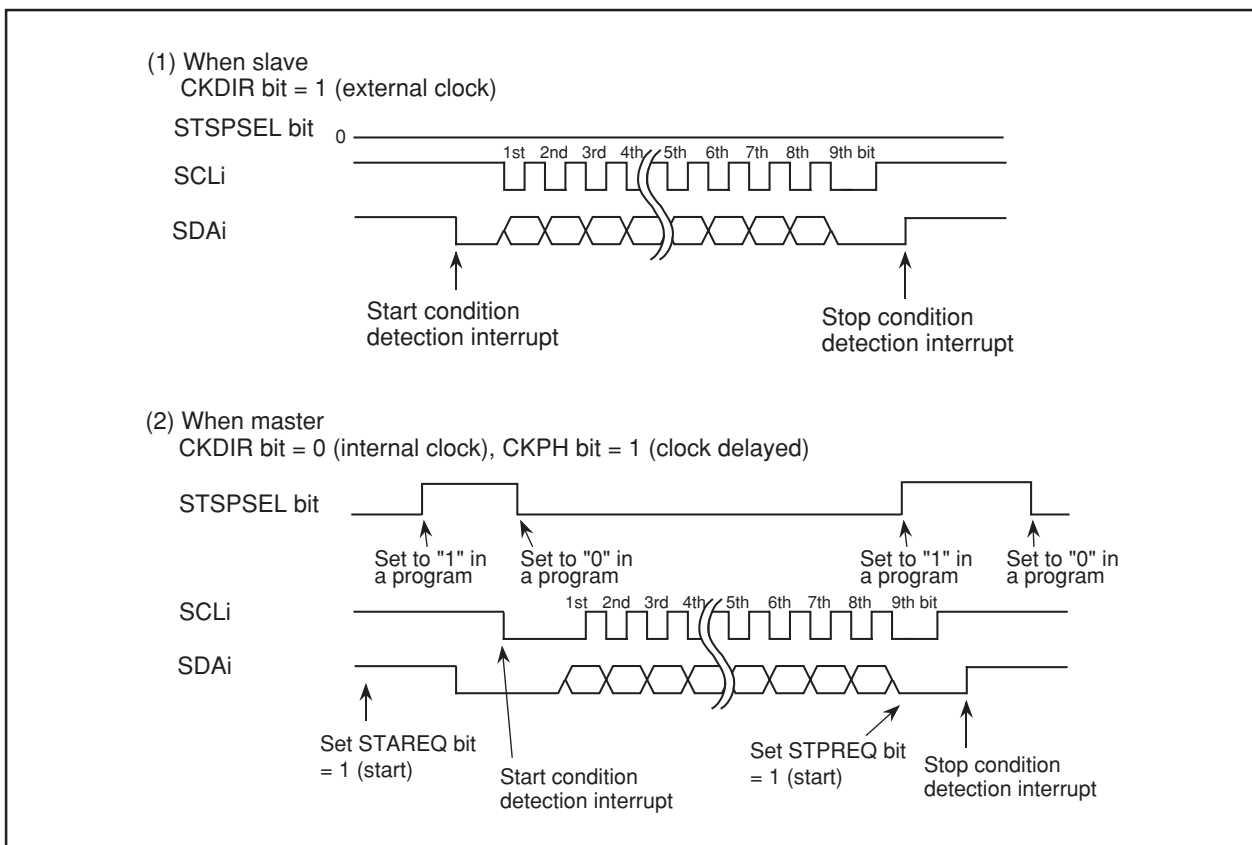


Figure 14.26 STSPSEL Bit Functions

14.1.3.3 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).

14.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 14.24.

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.

14.1.3.5 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 (I²C mode) and the SMD2 to SMD0 bits in the UiMR register = 000b (serial I/O 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).

14.1.3.6 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.

14.1.3.7 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.

14.1.3.8 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.

14.1.4 Special Mode 2

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. Table 14.14 lists the specifications of Special Mode 2. Figure 14.27 shows communication control example for Special Mode 2. Table 14.15 lists the registers used in Special Mode 2 and the register values set.

Table 14.14 Special Mode 2 Specifications

Item	Specification
Transfer data format	Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> • Master mode The CKDIR bit in the UiMR register = 0 (internal clock) : $f_j / 2(n+1)$ $f_j = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the UiBRG register 00h to FFh • Slave mode The CKDIR bit = 1 (external clock selected) : Input from CLKi pin
Transmit/receive control	Controlled by input/output ports
Transmission start condition	Before transmission can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Reception start condition	Before reception can start, the following requirements must be met ⁽¹⁾ <ul style="list-style-type: none"> • The RE bit in the UiC1 register = 1 (reception enabled) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> • The UiIRS bit ⁽²⁾ = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit = 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register For reception <ul style="list-style-type: none"> • When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	Overrun error ⁽³⁾ 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.

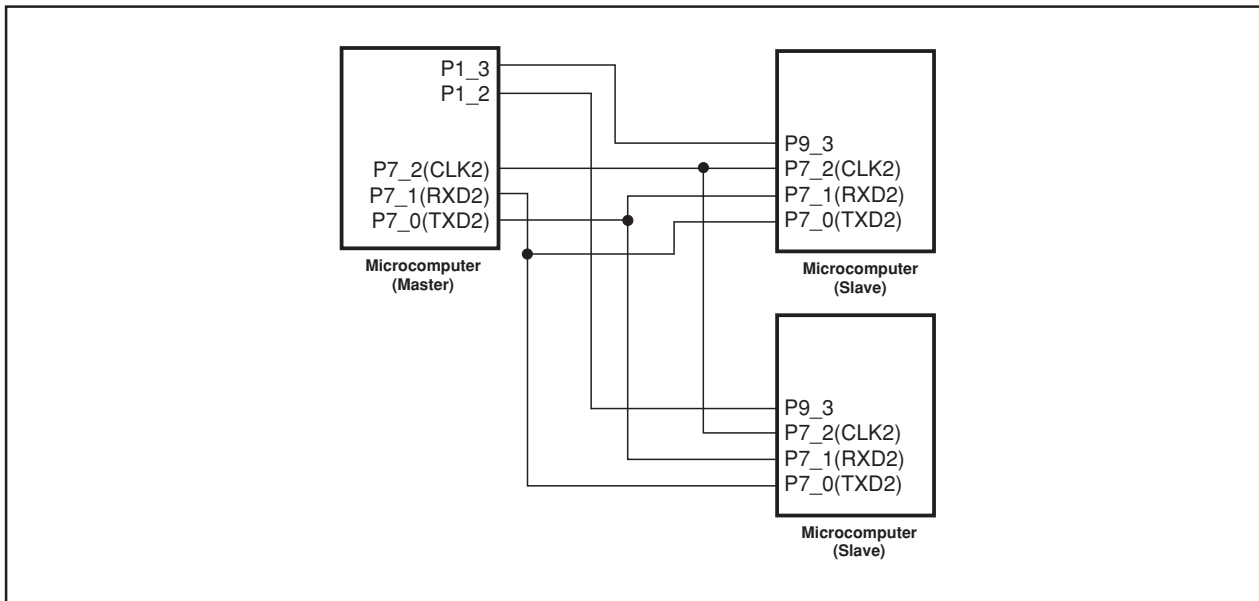


Figure 14.27 Serial Bus Communication Control Example (UART2)

Table 14.15 Registers to Be Used and Settings in Special Mode 2

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

14.1.4.1 Clock Phase Setting Function

One of four combinations of transfer clock phases and polarities can be selected using the CKPH bit in the 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 14.28 shows the transmission and reception timing in master (internal clock).

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

Figure 14.30 shows the transmission and reception timing (CKPH = 1) in slave (external clock).

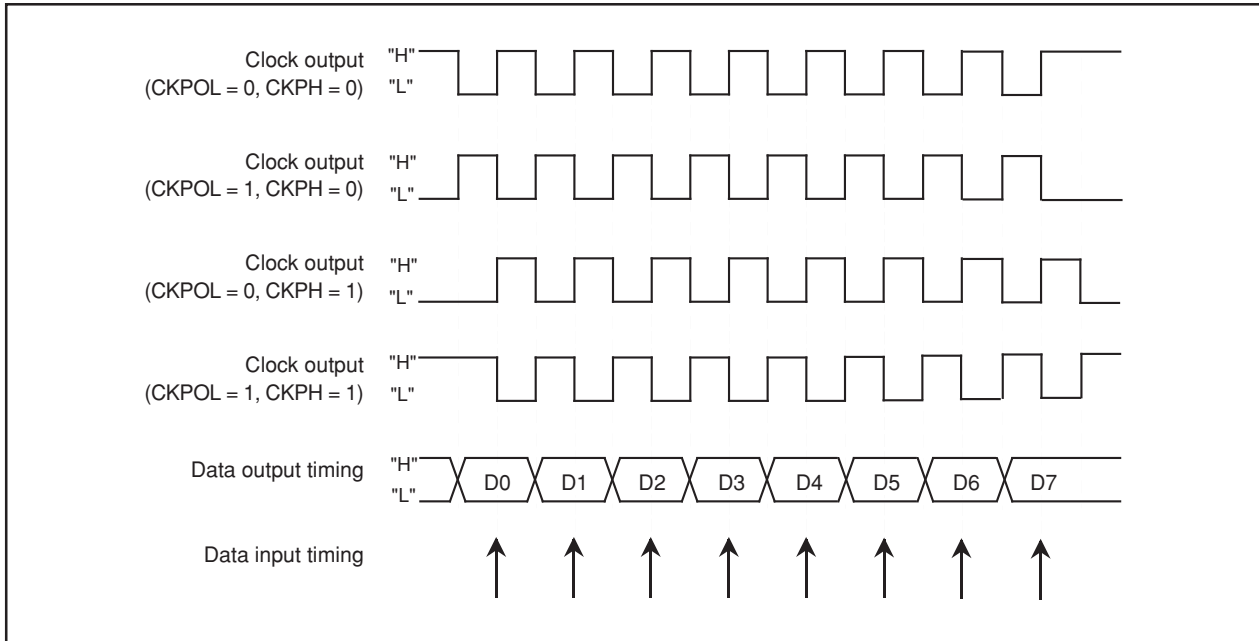


Figure 14.28 Transmission and Reception Timing in Master Mode (Internal Clock)

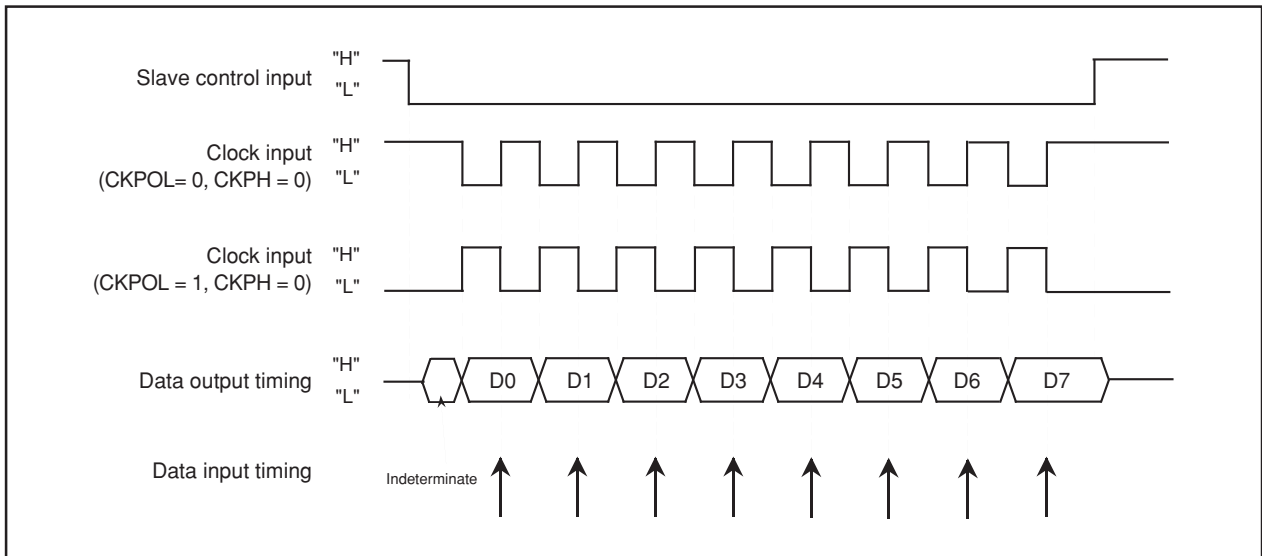


Figure 14.29 Transmission and Reception Timing (CKPH = 0) in Slave Mode (External Clock)

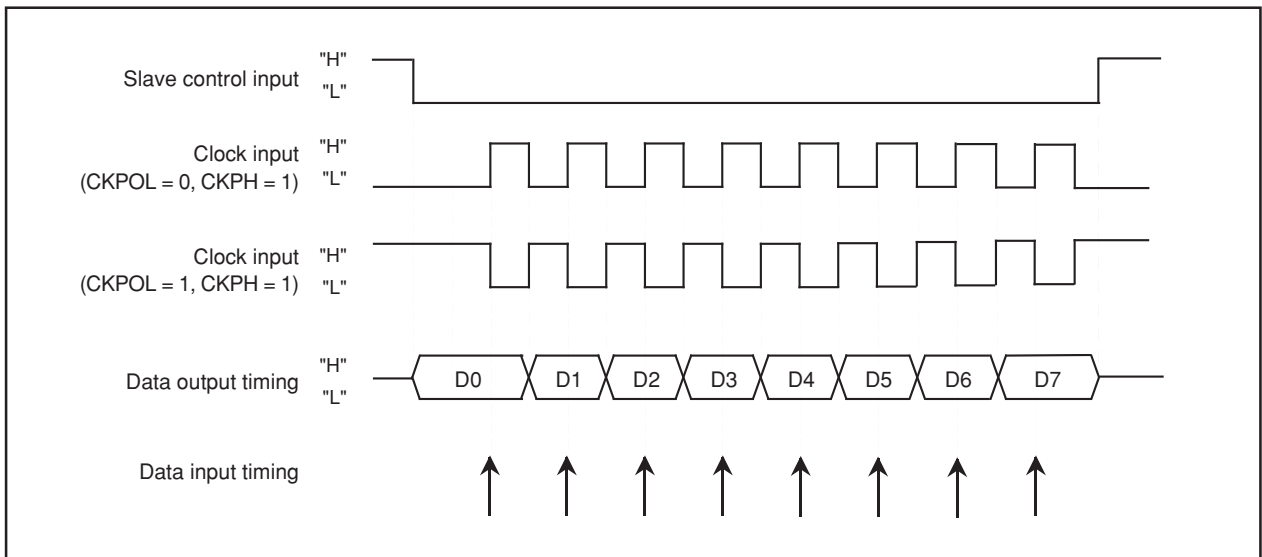


Figure 14.30 Transmission and Reception Timing (CKPH = 1) in Slave Mode (External Clock)

14.1.5 Special Mode 3 (IE Mode)

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

Table 14.16 lists the registers used in IE mode and the register values set. Figure 14.31 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 14.16 Registers to Be Used and Settings in IE Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data
UiRB ⁽¹⁾	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 ⁽²⁾	Select the source of UART2 transmit interrupt
	U2RRM ⁽²⁾ , UiLCH, UiERE	Set to "0"
UiSMR	0 to 3, 7	Set to "0"
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to "1" to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
IFSR0	IFSR06, IFSR07	Set to "1"
UCON	U0IRS, U1IRS	Select the 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.

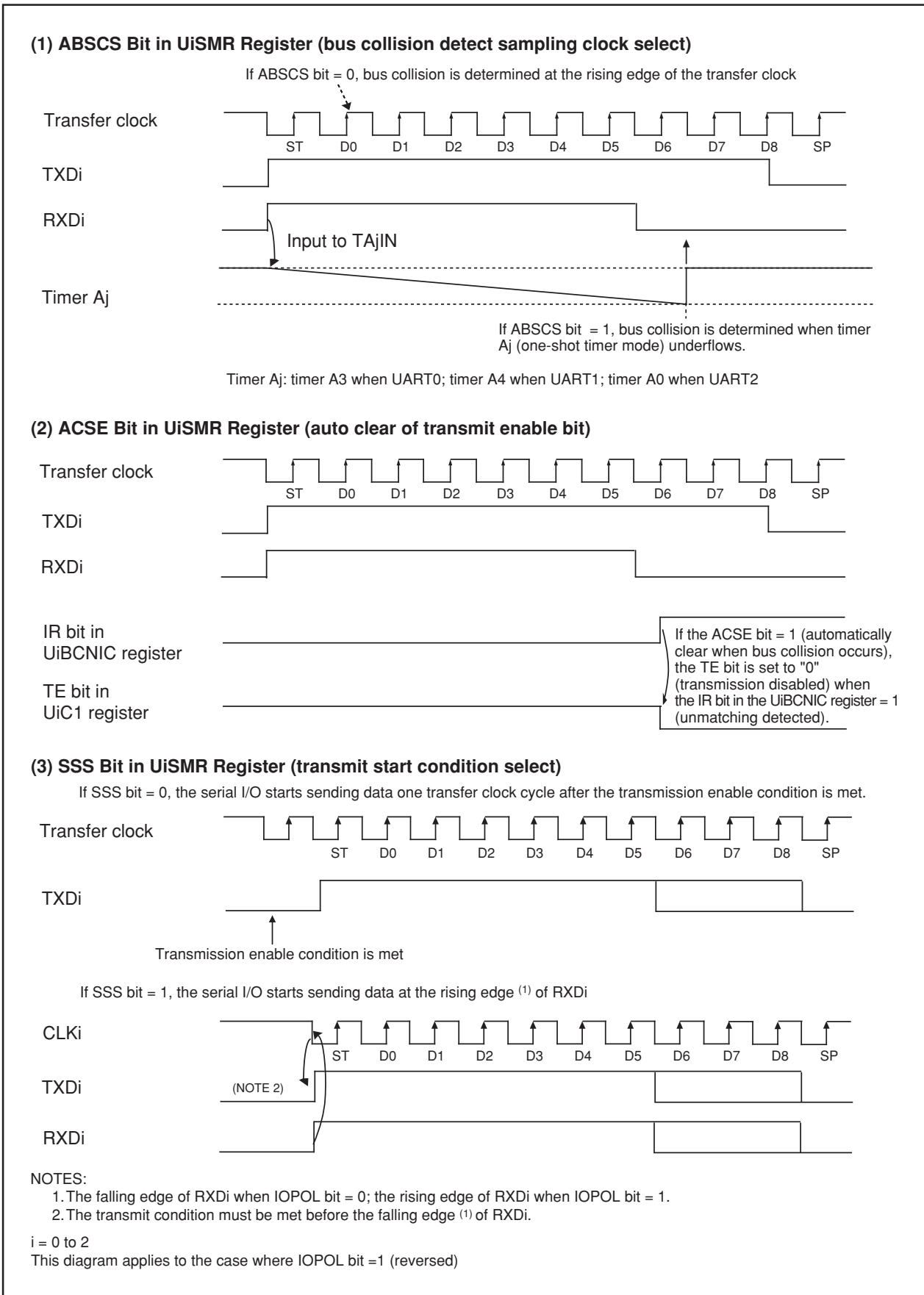


Figure 14.31 Bus Collision Detect Function-Related Bits

14.1.6 Special Mode 4 (SIM Mode) (UART2)

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

Table 14.17 lists the specifications of SIM mode. Table 14.18 lists the registers used in the SIM mode and the register values set. Figure 14.32 shows the typical transmit/receive timing in SIM mode.

Table 14.17 SIM Mode Specifications

Item	Specification
Transfer data format	<ul style="list-style-type: none"> • Direct format • Inverse format
Transfer clock	<ul style="list-style-type: none"> • The CKDIR bit in the U2MR register = 0 (internal clock) : $f_i/16(n+1)$ $f_i = f1SIO, f2SIO, f8SIO, f32SIO$. n: Setting value of the U2BRG register 00h to FFh • The CKDIR bit = 1 (external clock) : $fEXT/16(n+1)$ $fEXT$: Input from CLK2 pin. n: Setting value of the U2BRG register 00h to FFh
Transmission start condition	Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> • The TE bit in the U2C1 register = 1 (transmission enabled) • The TI bit in the U2C1 register = 0 (data present in the U2TB register)
Reception start condition	Before reception can start, the following requirements must be met <ul style="list-style-type: none"> • The RE bit in the U2C1 register = 1 (reception enabled) • Start bit detection
Interrupt request generation timing ⁽²⁾	<ul style="list-style-type: none"> • For transmission When the serial I/O finished sending data from the U2TB transfer register (U2IRS bit = 1) • For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception)
Error detection	<ul style="list-style-type: none"> • Overrun error ⁽¹⁾ This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data • Framing error ⁽³⁾ This error occurs when the number of stop bits set is not detected • Parity error ⁽³⁾ During reception, if a parity error is detected, parity error signal is output from the TXD2 pin. During transmission, a parity error is detected by the level of input to the RXD2 pin when a transmission interrupt occurs • Error sum flag This flag is set to "1" when any of the overrun, framing, and parity errors is encountered

NOTES:

1. If an overrun error occurs, the value of the U2RB register will be 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 14.18 Registers to Be Used and Settings in SIM Mode

Register	Bit	Function
U2TB ⁽¹⁾	0 to 7	Set transmission data
U2RB ⁽¹⁾	0 to 7	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	0 to 7	Set 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 ⁽¹⁾	0 to 3	Set to "0"
U2SMR2	0 to 7	Set to "0"
U2SMR3	0 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"

NOTE:

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

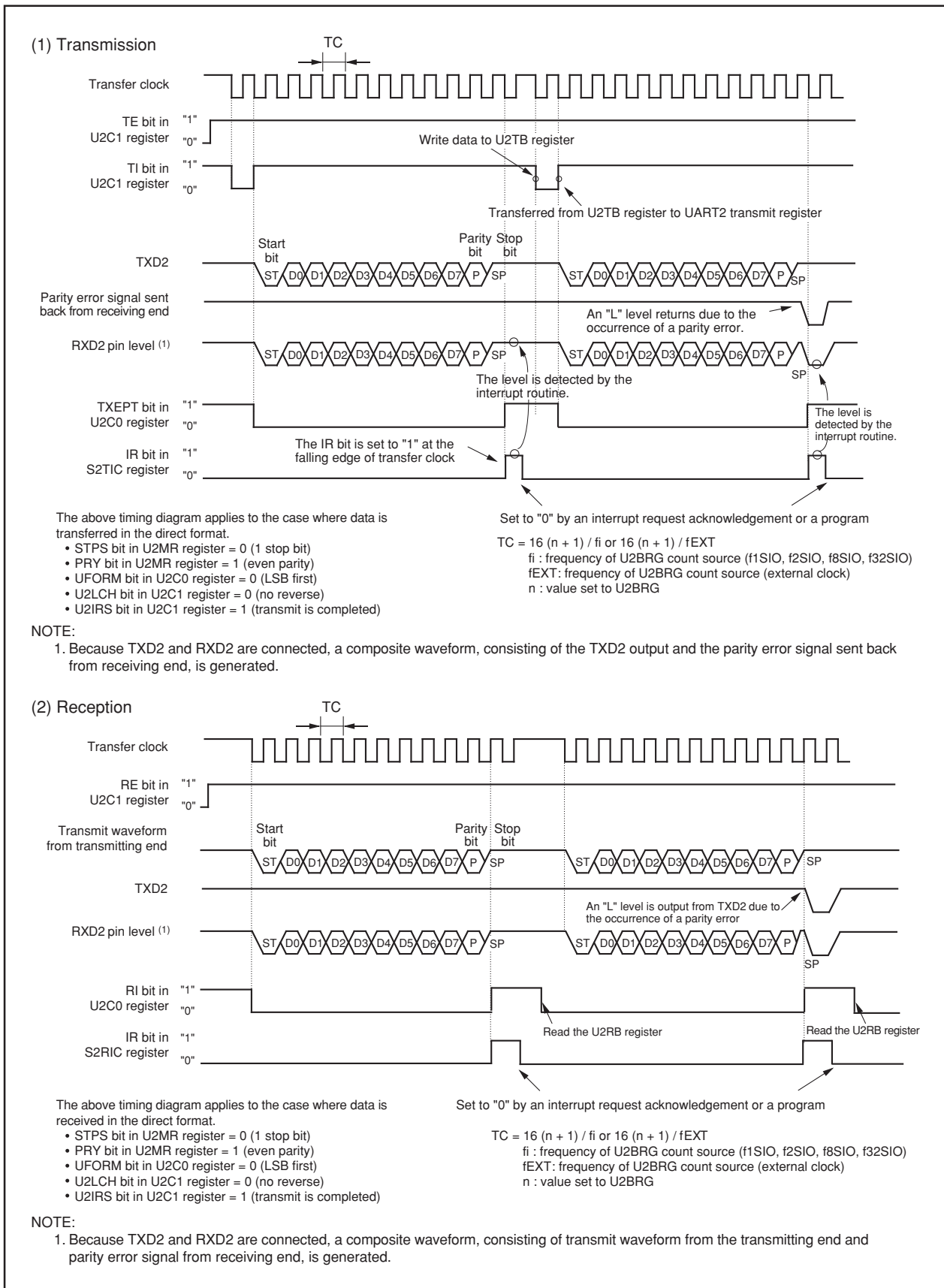


Figure 14.32 Transmit and Receive Timing in SIM Mode

Figure 14.33 shows the example of connecting the SIM interface. Connect TXD2 and RXD2 and apply pull-up.

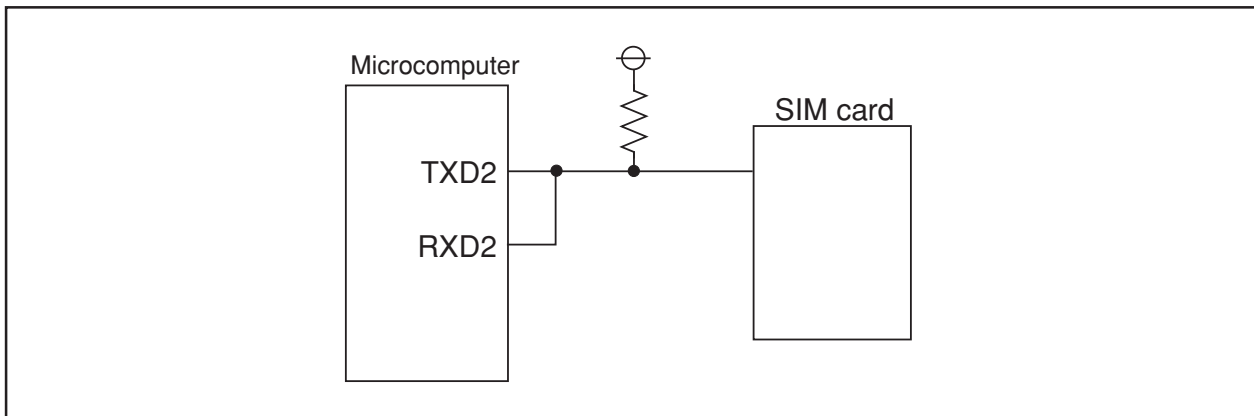


Figure 14.33 SIM Interface Connection

14.1.6.1 Parity Error Signal Output

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

The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TXD2 output low with the timing shown in Figure 14.32. 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 14.34 shows the output timing of the parity error signal

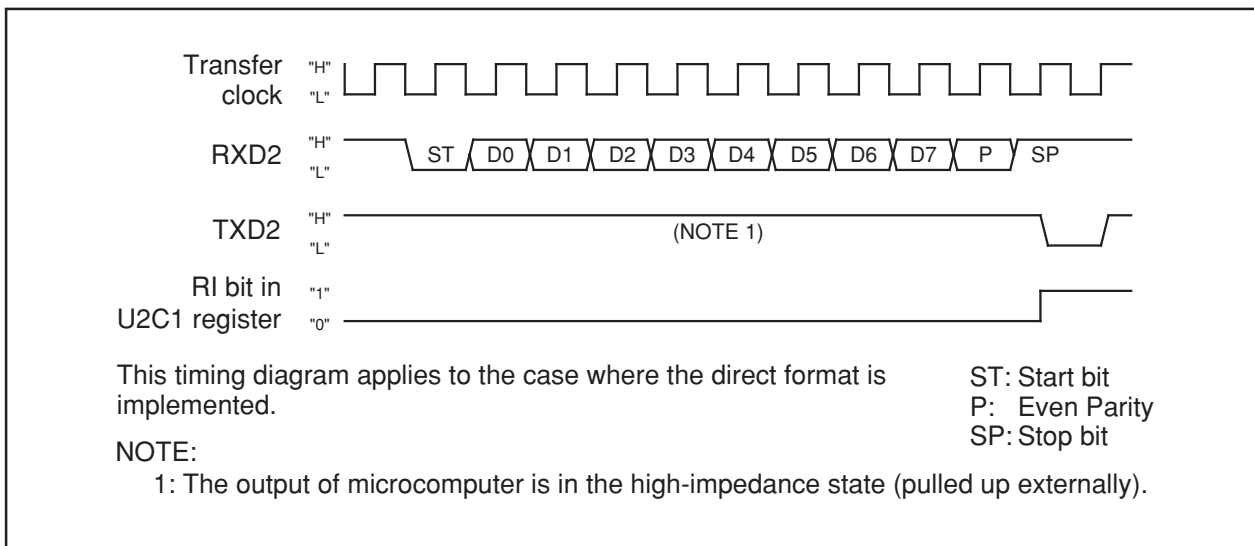


Figure 14.34 Parity Error Signal Output Timing

14.1.6.2 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 14.35 shows the SIM interface format.

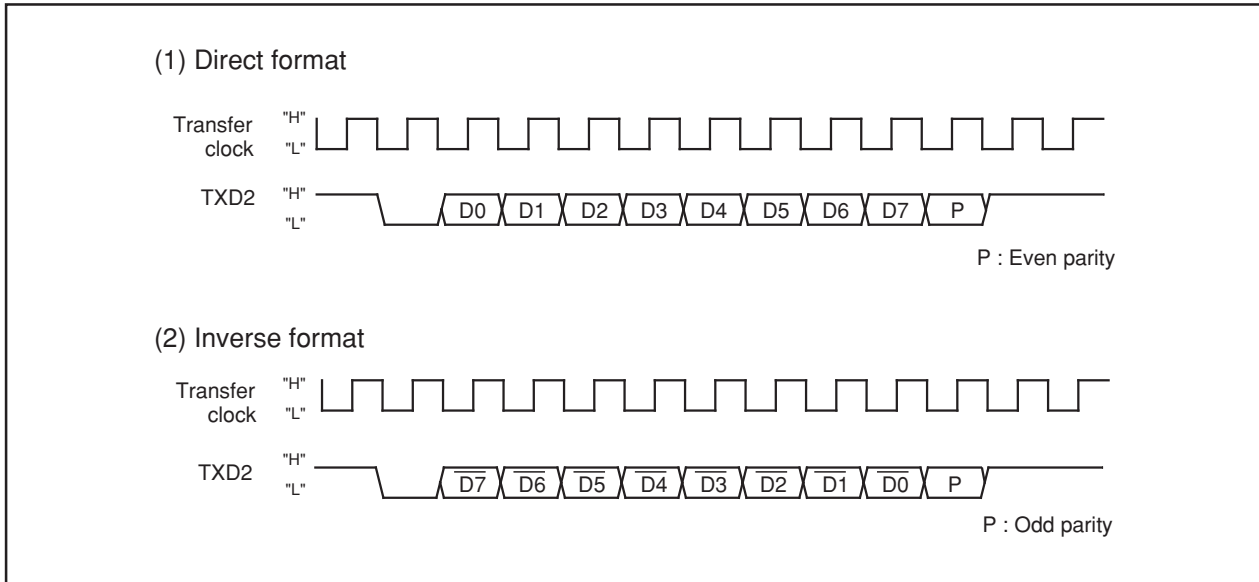


Figure 14.35 SIM Interface Format

14.2 SI/Oi (i = 3 to 6) ⁽¹⁾

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

Figure 14.36 shows the block diagram of SI/Oi, and Figures 14.37 and 14.38 show the SI/Oi-related registers. Table 14.19 lists the specifications of SI/Oi.

NOTE:

1. 100-pin version supports SI/O3 and SI/O4.
- 128-pin version supports SI/O3, SI/O4, SI/O5 and SI/O6.

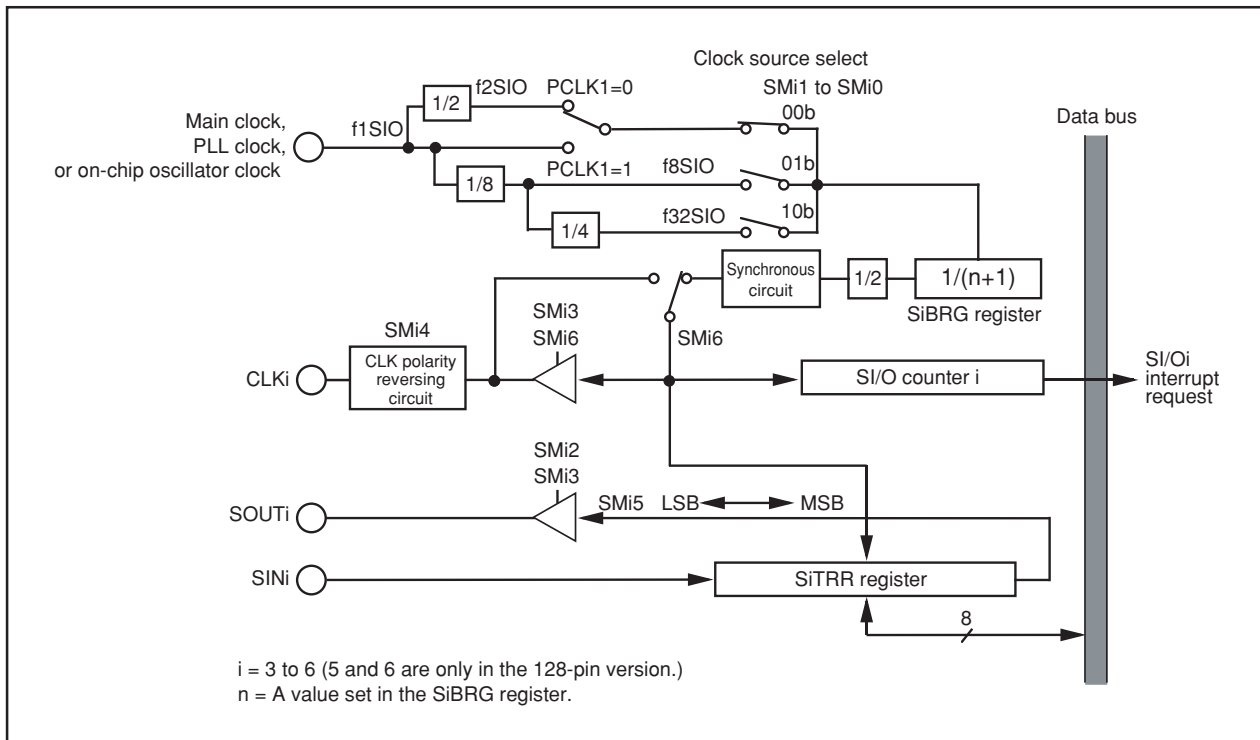


Figure 14.36 SI/Oi Block Diagram

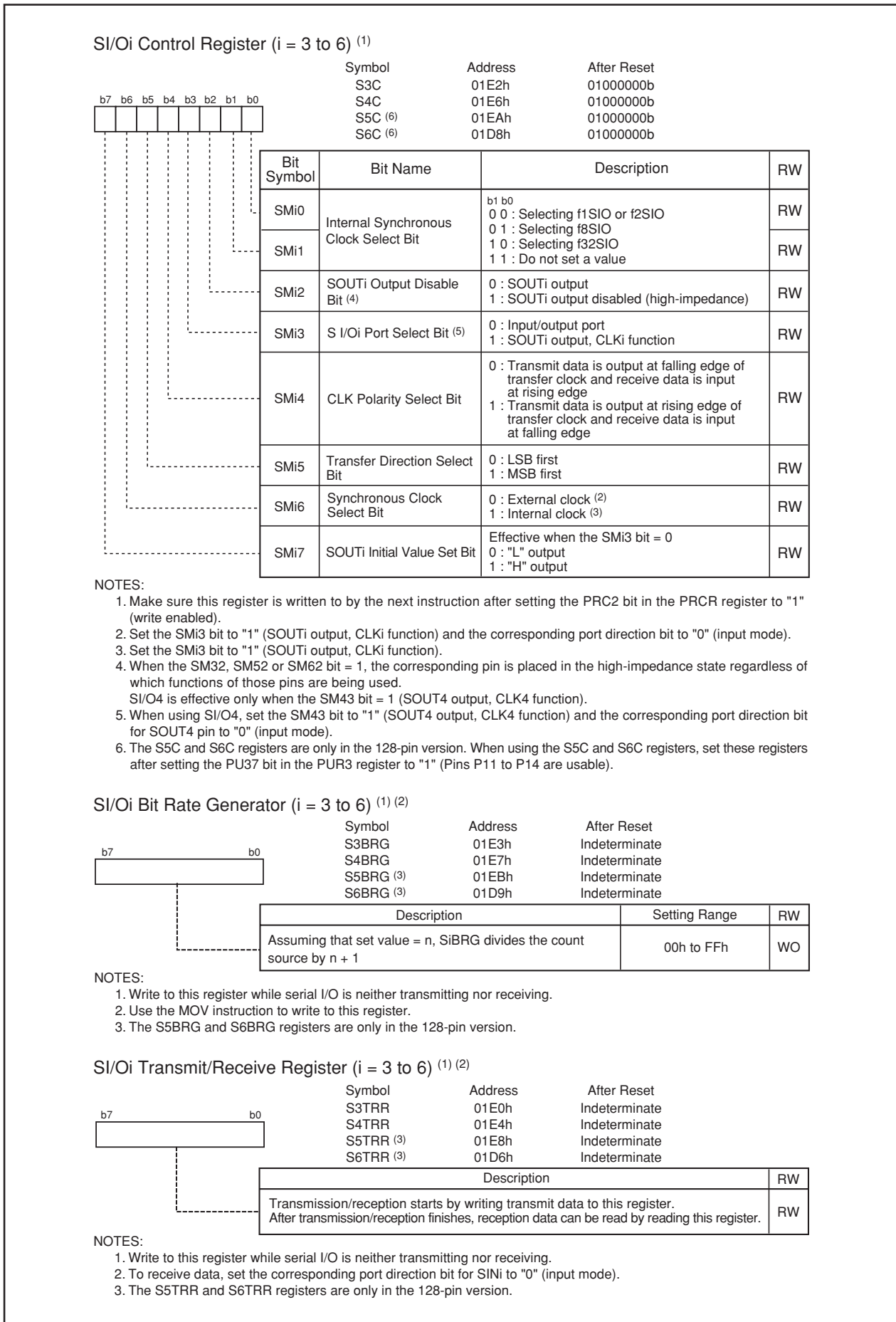


Figure 14.37 S3C to S6C Registers, S3BRG to S6BRG Registers, and S3TRR to S6TRR Registers

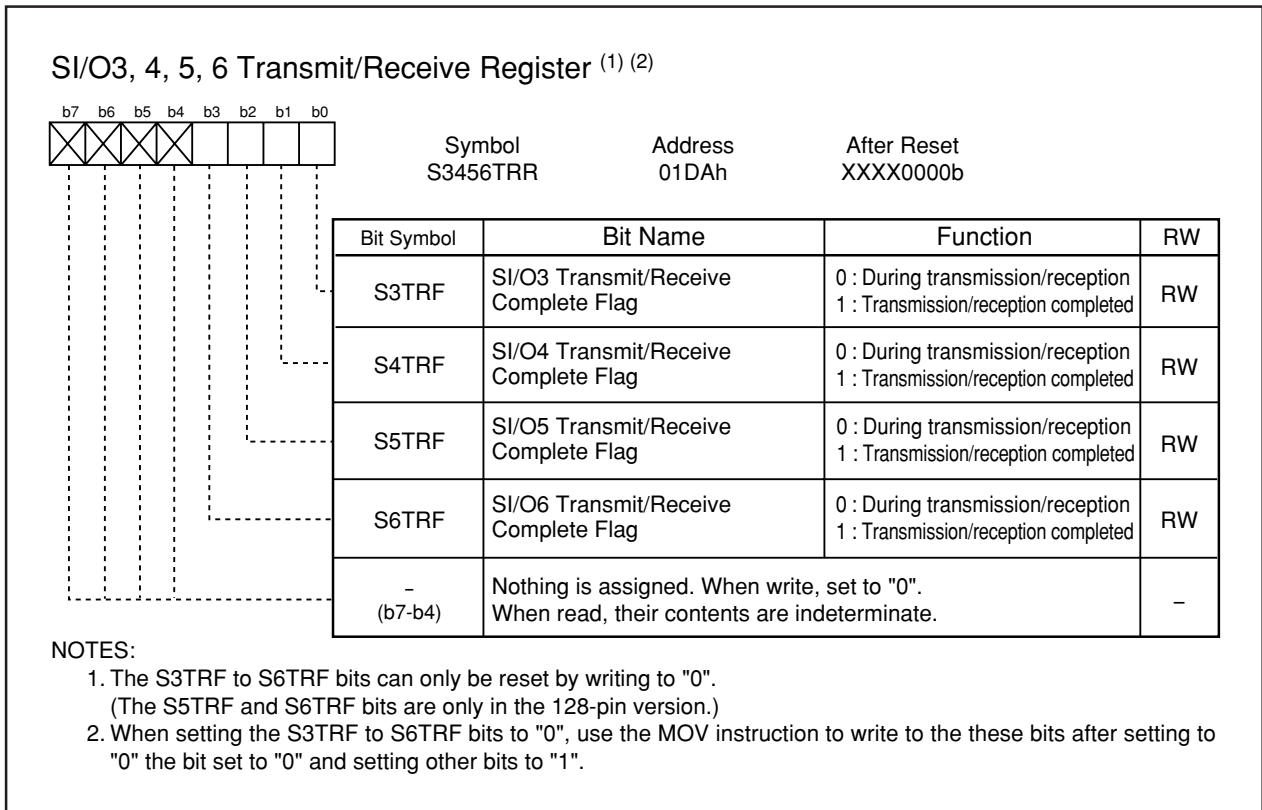


Figure 14.38 S3456TRR Register

Table 14.19 SI/Oi Specifications

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> • SMI6 bit in SiC register = 1 (internal clock) : $f_j / 2(n+1)$ $f_j = f1SIO, f8SIO, f32SIO$. n = Setting value of SiBRG register 00h to FFh • SMI6 bit = 0 (external clock) : Input from CLKi pin ⁽¹⁾
Transmission/Reception Start Condition	Before transmission/reception can start, the following requirements must be met Write transmit data to the SiTRR register ^{(2) (3)}
Interrupt Request Generation Timing	<ul style="list-style-type: none"> • When SMI4 bit in SiC register = 0 The rising edge of the last transfer clock pulse ⁽⁴⁾ • When SMI4 bit = 1 The falling edge of the last transfer clock pulse ⁽⁴⁾
CLKi Pin Function	I/O port, transfer clock input, transfer clock output
SOUTi Pin Function	I/O port, transmit data output, high-impedance
SINi Pin Function	I/O port, receive data input
Select Function	<ul style="list-style-type: none"> • LSB first or MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Function for setting an SOUTi initial value set function When the SMI6 bit in the SiC register = 0 (external clock), the SOUTi pin output level while not transmitting can be selected. • CLK polarity selection Whether transmit data is output/input timing at the rising edge or falling edge of transfer clock can be selected.

i = 3 to 6 (5 and 6 are only in the 128-pin version.)

NOTES:

- To set the SMI6 bit in the SiC register to "0" (external clock), follow the procedure described below.
 - If the SMI4 bit in the SiC register = 0, write transmit data to the SiTRR register while input on the CLKi pin is high. The same applies when rewriting the SMI7 bit in the SiC register.
 - If the SMI4 bit = 1, write transmit data to the SiTRR register while input on the CLKi pin is low. The same applies when rewriting the SMI7 bit.
 - Because shift operation continues as long as the transfer clock is supplied to the SI/Oi circuit, stop the transfer clock after supplying eight pulses. If the SMI6 bit = 1 (internal clock), the transfer clock automatically stops.
- Unlike UART0 to UART2, SI/Oi is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the SiTRR register during transmission.
- When the SMI6 bit = 1 (internal clock), SOUTi retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to a high-impedance state. However, if transmit data is written to the SiTRR register during this period, SOUTi immediately goes to a high-impedance state, with the data hold time thereby reduced.
- When the SMI6 bit = 1 (internal clock), the transfer clock stops in the high state if the SMI4 bit = 0, or stops in the low state if the SMI4 bit = 1.

14.2.1 SI/Oi Operation Timing

Figure 14.39 shows the SI/Oi operation timing.

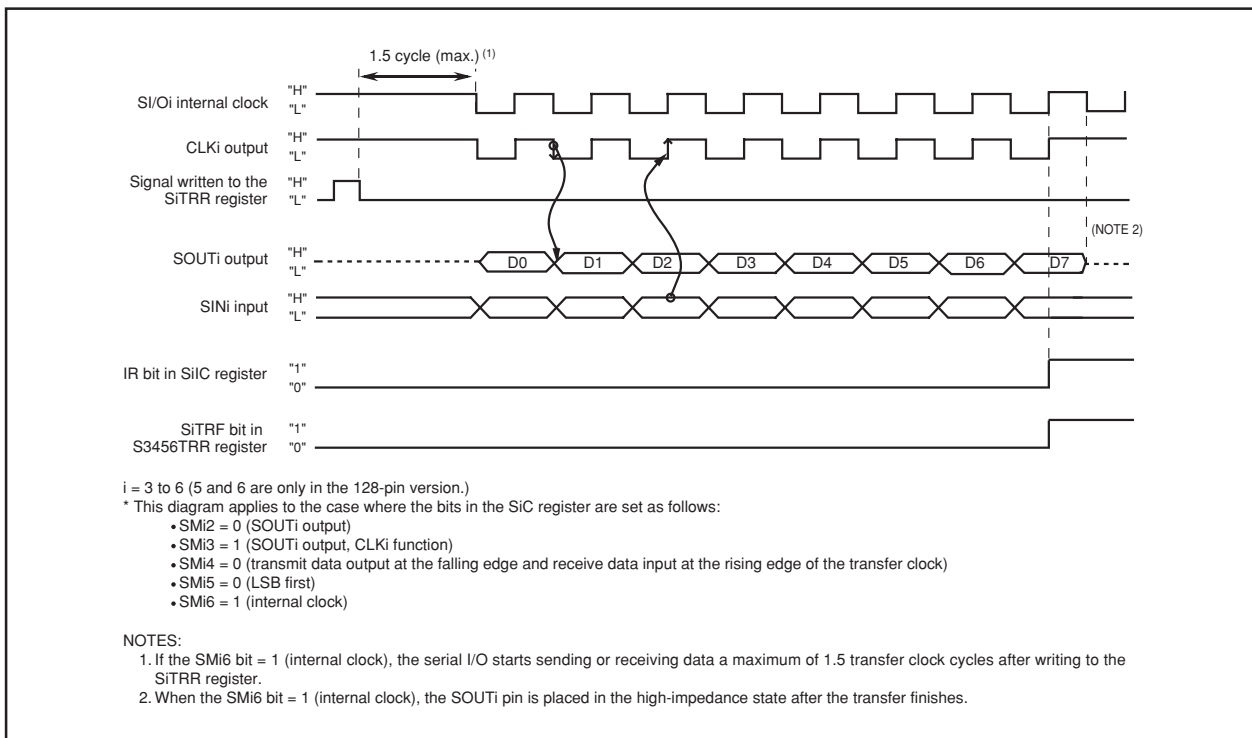


Figure 14.39 SI/Oi Operation Timing

14.2.2 CLK Polarity Selection

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

Figure 14.40 shows the polarity of the transfer clock.

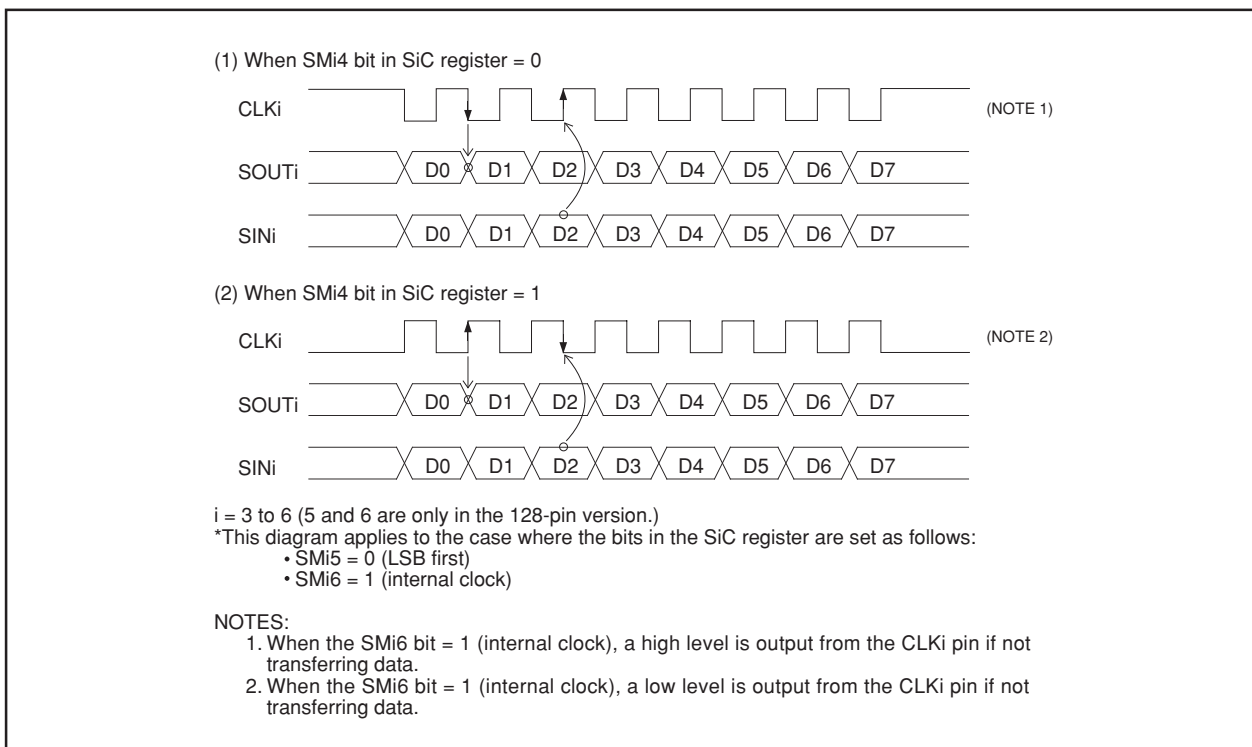


Figure 14.40 Polarity of Transfer Clock

14.2.3 Functions for Setting an SOUTi Initial Value

If the SMi6 bit in the SiC register = 0 (external clock), the SOUTi pin output can be fixed high or low when not transferring ⁽¹⁾. Figure 14.41 shows the timing chart for setting an SOUTi initial value and how to set it.

NOTE:

1. When CAN0 function is selected, P7_4, P7_5 and P8_0 can be used as input/output pins for SI/O4. When CAN0 function is not selected, P9_5, P9_6 and P9_7 can be used as input/output pins for SI/O4.

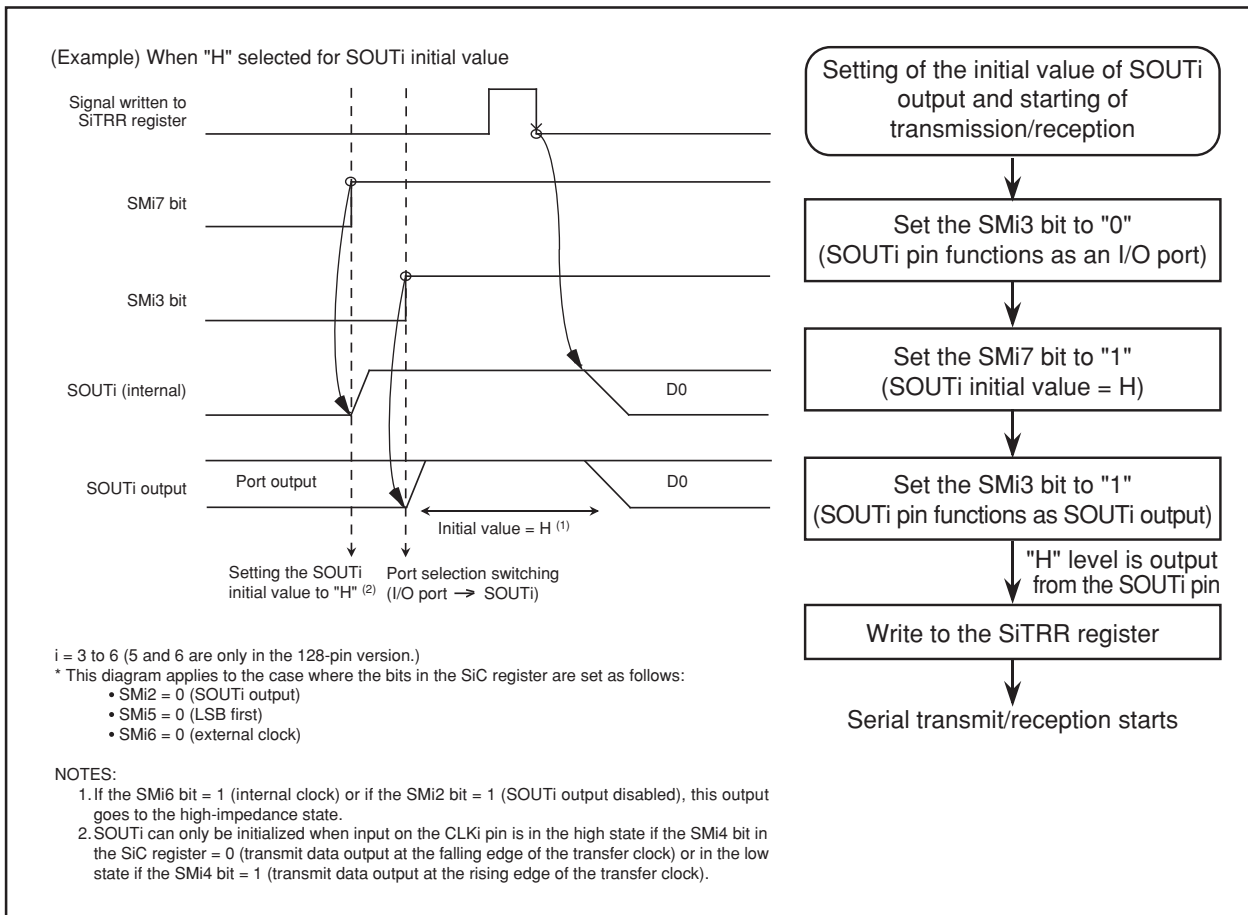


Figure 14.41 SOUTi's Initial Value Setting

15. 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, 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 15.1 shows the performance of the A/D converter. Figure 15.1 shows the block diagram of the A/D converter, and Figures 15.2 and 15.3 show the A/D converter-related registers.

Table 15.1 A/D Converter Performance

Item	Performance
Method of A/D Conversion	Successive approximation (capacitive coupling amplifier)
Analog Input Voltage ⁽¹⁾	0V to AVCC (VCC)
Operating Clock ϕ AD ⁽²⁾	fAD, divide-by-2 of fAD, divide-by-3 of fAD, divide-by-4 of fAD, divide-by-6 of fAD, divide-by-12 of fAD
Resolution	8 bits or 10 bits (selectable)
Integral Nonlinearity Error	When AVCC = VREF = 5 V <ul style="list-style-type: none"> • With 8-bit resolution: ± 2LSB • With 10-bit resolution <ul style="list-style-type: none"> AN0 to AN7 input, AN0_0 to AN0_7 input and AN2_0 to AN2_7 input: ± 3LSB ANEX0 and ANEX1 input (including mode in which external operation amp is selected): ± 7LSB When AVCC = VREF = 3.3 V <ul style="list-style-type: none"> • With 8-bit resolution: ± 2LSB • With 10-bit resolution <ul style="list-style-type: none"> AN0 to AN7 input, AN0_0 to AN0_7 input and AN2_0 to AN2_7 input: ± 5LSB ANEX0 and ANEX1 input (including mode in which external operation amp is selected): ± 7LSB
Operating Modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog Input Pins	8 pins (AN0 to AN7) + 2 pins (ANEX0 and ANEX1) + 8 pins (AN0_0 to AN0_7) + 8 pins (AN2_0 to AN2_7)
A/D Conversion Start Condition	<ul style="list-style-type: none"> • Software trigger <ul style="list-style-type: none"> The ADST bit in the ADCON0 register is set to “1” (A/D conversion starts) • External trigger (retriggerable) <ul style="list-style-type: none"> Input on the ADTRG pin changes state from high to low after the ADST bit is set to “1” (A/D conversion starts)
Conversion Speed Per Pin	<ul style="list-style-type: none"> • Without sample and hold function <ul style="list-style-type: none"> 8-bit resolution: 49 ϕAD cycles, 10-bit resolution: 59 ϕAD cycles • With sample and hold function <ul style="list-style-type: none"> 8-bit resolution: 28 ϕAD cycles, 10-bit resolution: 33 ϕAD cycles

NOTES:

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

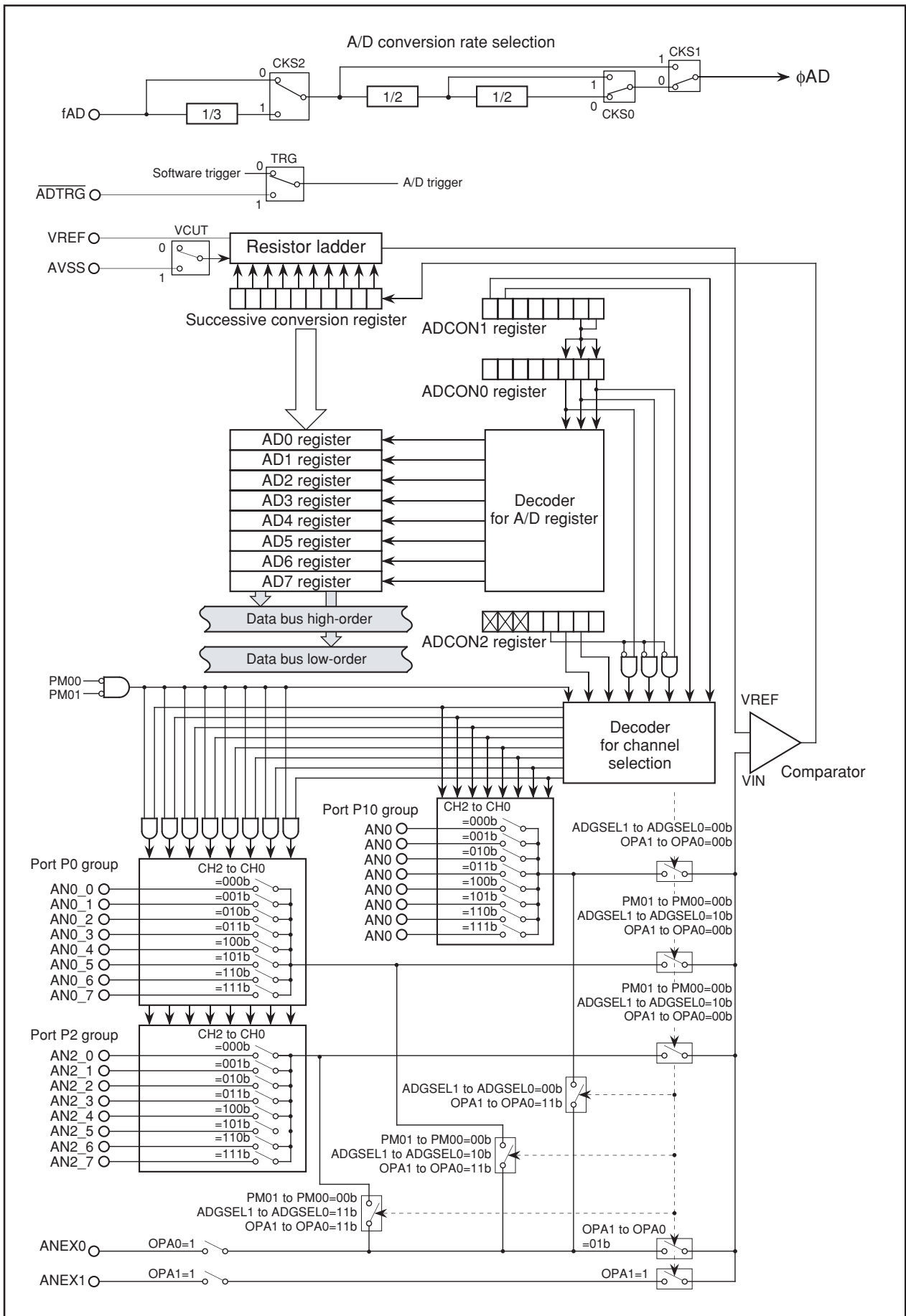


Figure 15.1 A/D Converter Block Diagram

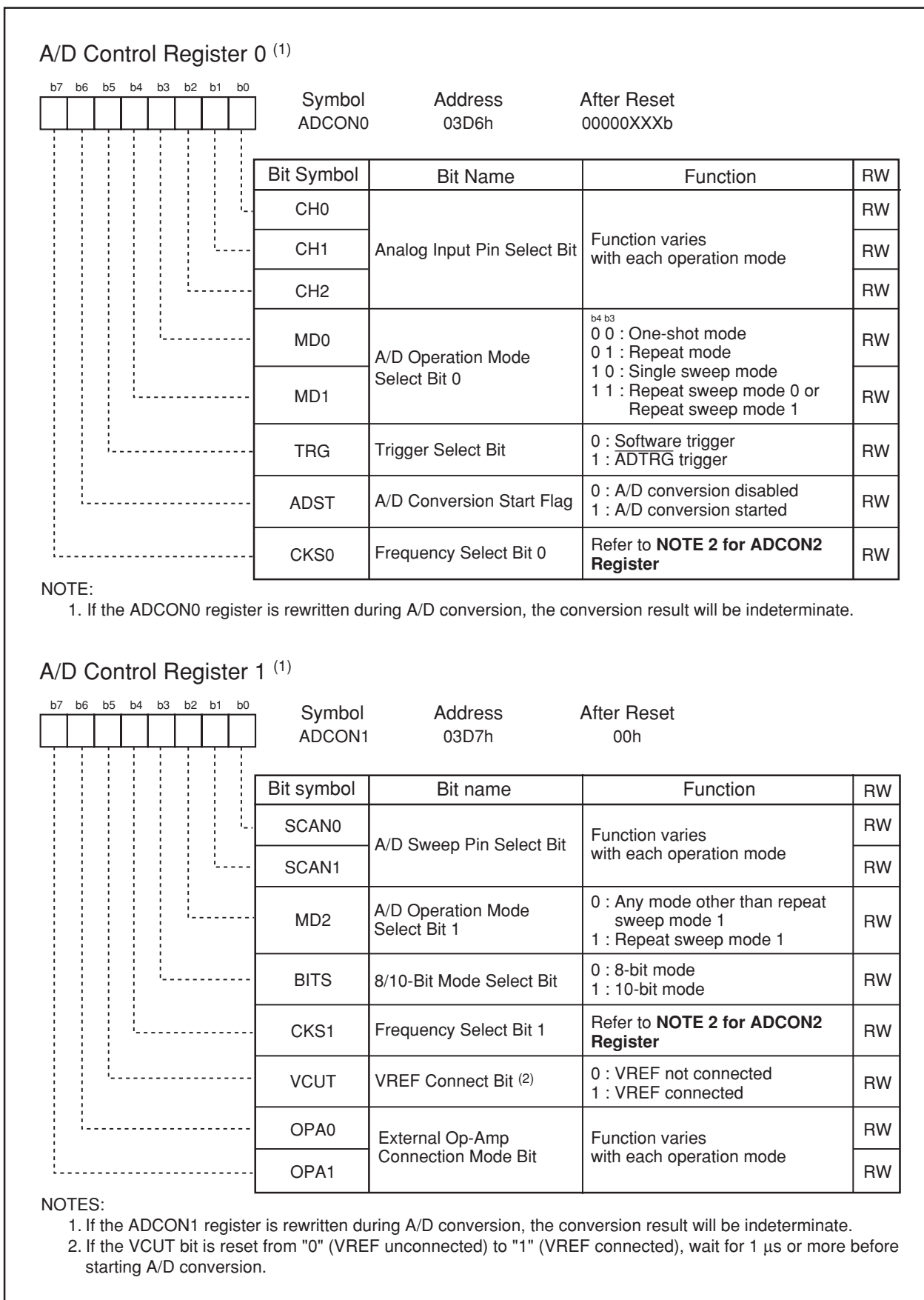


Figure 15.2 ADCON0 Register and ADCON1 Register

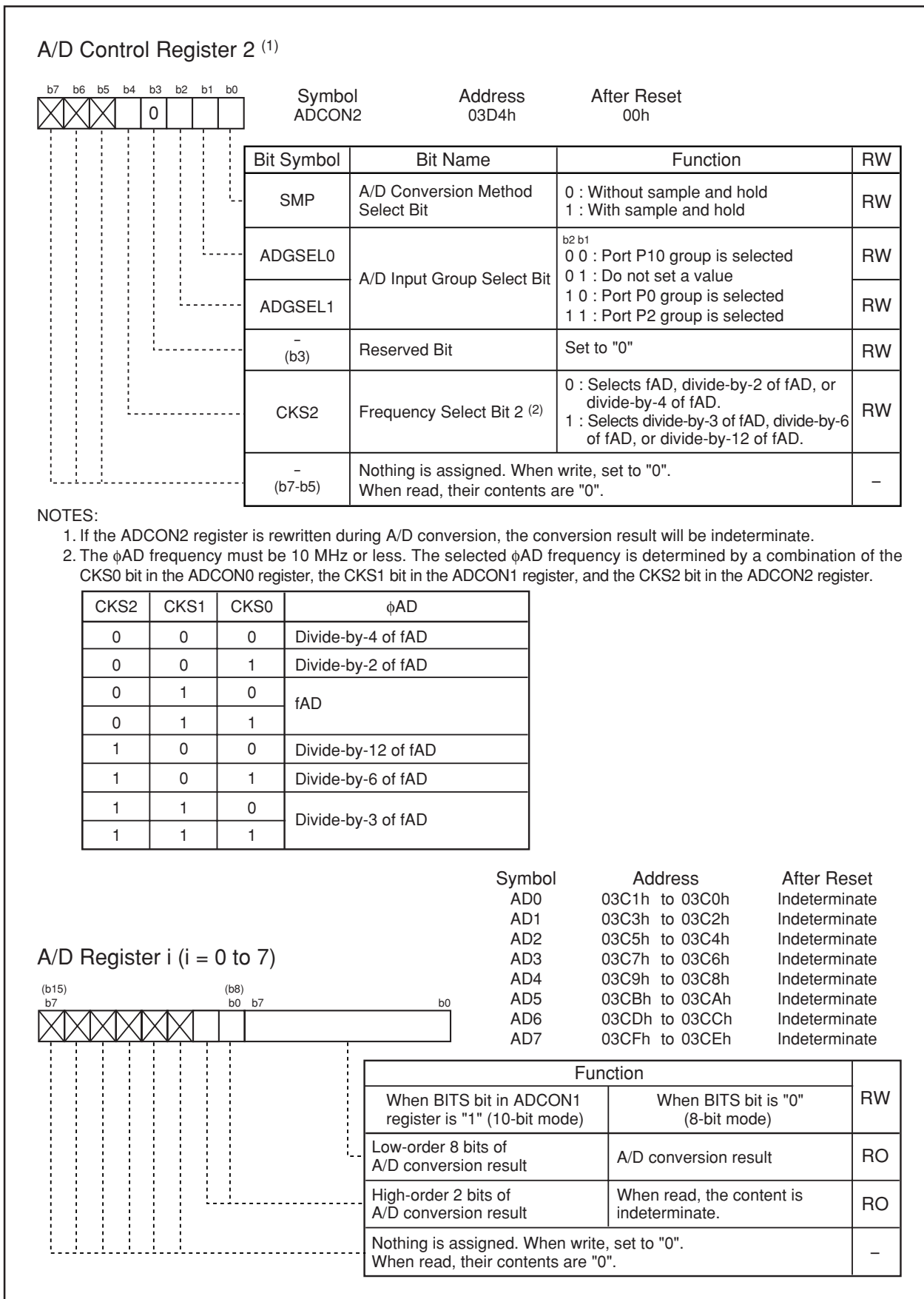


Figure 15.3 ADCON2 Register, and AD0 to AD7 Registers

15.1 Mode Description

15.1.1 One-shot Mode

In one-shot mode, analog voltage applied to a selected pin is A/D converted once. Table 15.2 lists the specifications of one-shot mode. Figure 15.4 shows the ADCON0 and ADCON1 registers in one-shot mode.

Table 15.2 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"> When the TRG bit in the ADCON0 register is "0" (software trigger) The ADST bit in the ADCON0 register is set to "1" (A/D conversion starts) When the TRG bit is "1" ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to "1" (A/D conversion starts)
A/D Conversion Stop Condition	<ul style="list-style-type: none"> Completion of A/D conversion (If a software trigger is selected, the ADST bit is set to "0" (A/D conversion halted).) Set the ADST bit to "0"
Interrupt Request Generation Timing	Completion of A/D conversion
Analog Input Pin	Select one pin from AN0 to AN7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0 to ANEX1
Reading of Result of A/D Converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

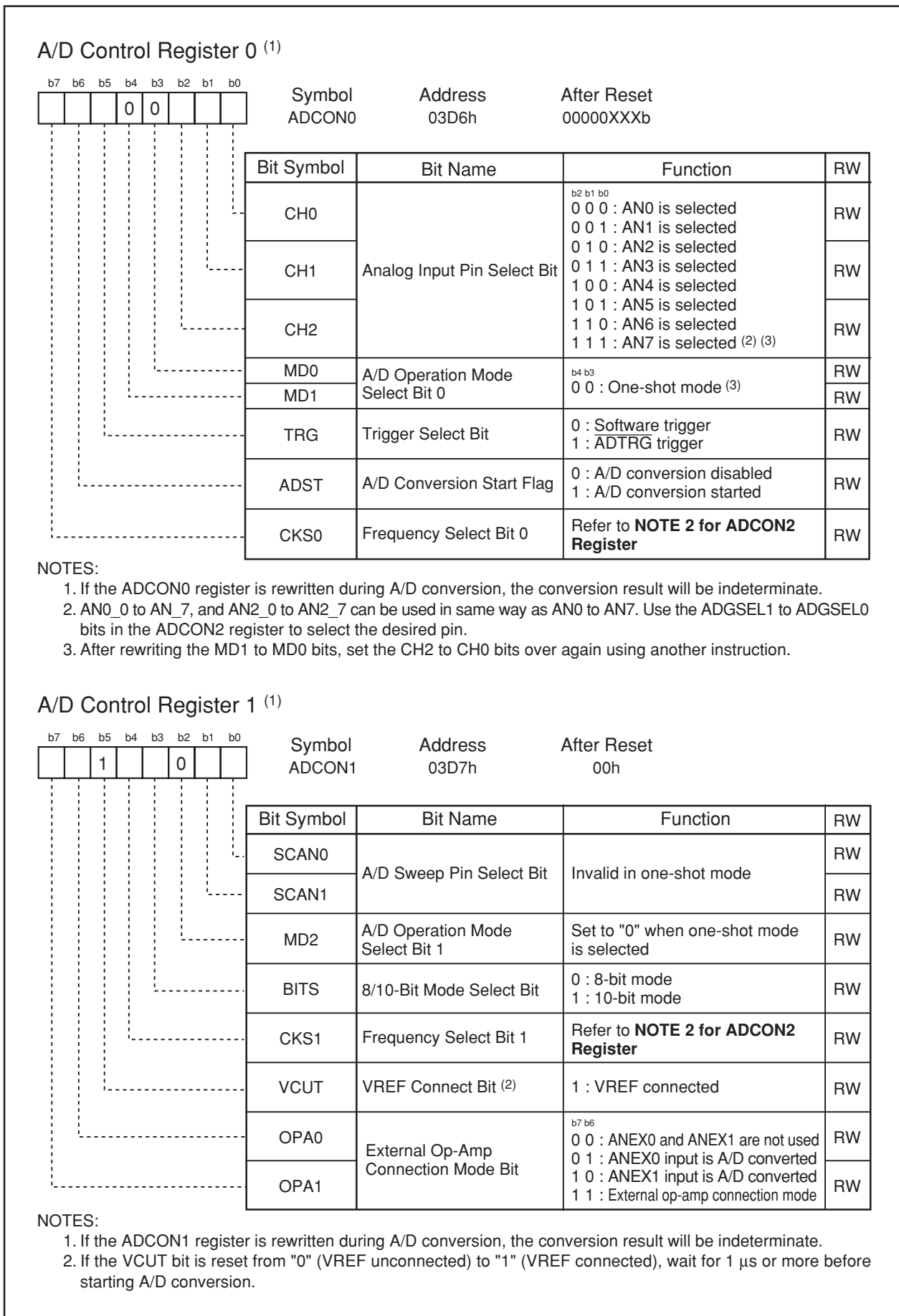


Figure 15.4 ADCON0 Register and ADCON1 Register in One-shot Mode

15.1.2 Repeat Mode

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

Table 15.3 lists the specifications of repeat mode. Figure 15.5 shows the ADCON0 and ADCON1 registers in repeat mode.

Table 15.3 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"> When the TRG bit in the ADCON0 register is "0" (software trigger) The ADST bit in the ADCON0 register is set to "1" (A/D conversion starts) When the TRG bit is "1" ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to "1" (A/D conversion starts)
A/D Conversion Stop Condition	Set the ADST bit to "0" (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input 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

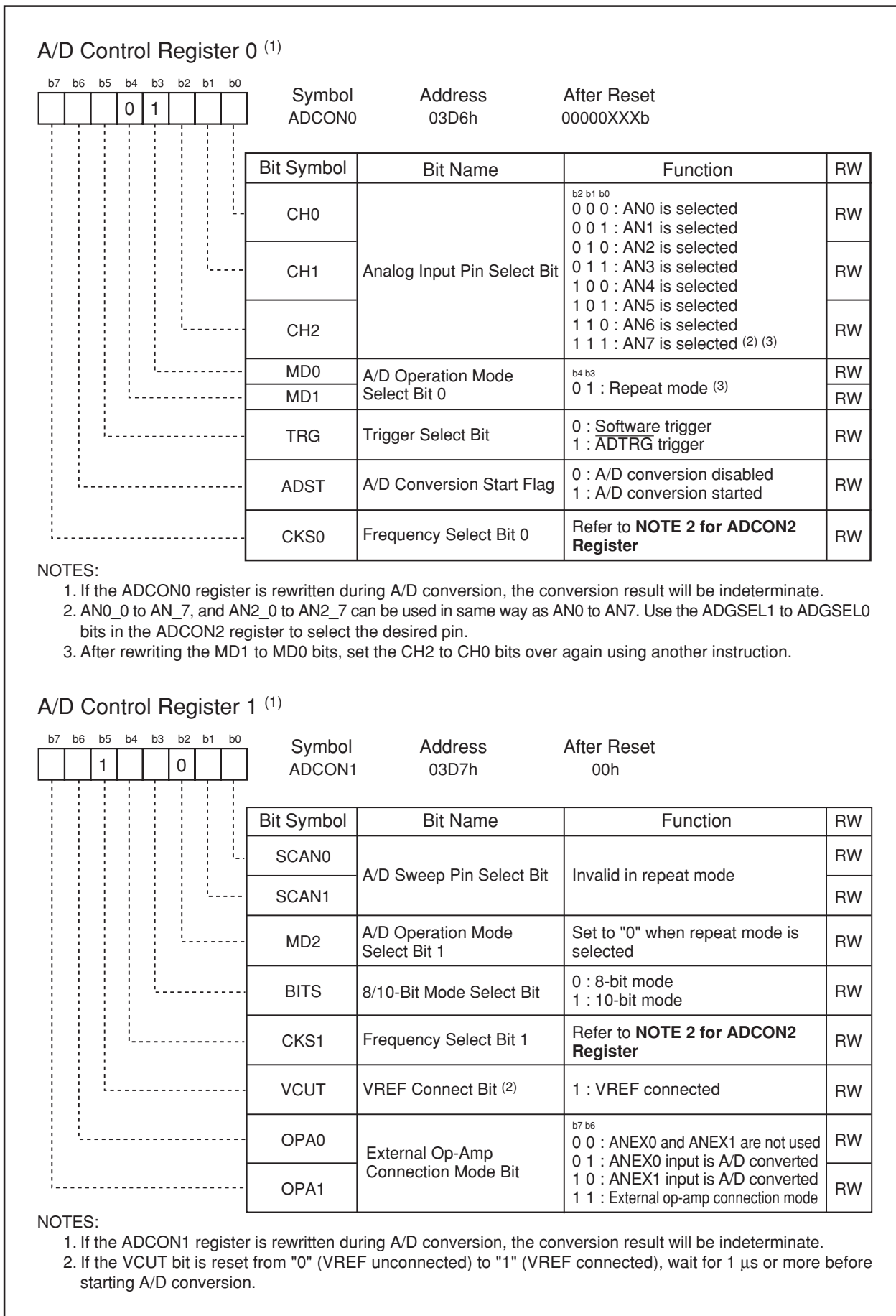


Figure 15.5 ADCON0 Register and ADCON1 Register in Repeat Mode

15.1.3 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 15.4 lists the specifications of single sweep mode. Figure 15.6 shows the ADCON0 and ADCON1 registers in single sweep mode.

Table 15.4 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"> When the TRG bit in the ADCON0 register is "0" (software trigger) The ADST bit in the ADCON0 register is set to "1" (A/D conversion starts) When the TRG bit is "1" ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to "1" (A/D conversion starts)
A/D Conversion Stop Condition	<ul style="list-style-type: none"> Completion of A/D conversion (If a software trigger is selected, the ADST bit is set to "0" (A/D conversion halted).) Set the ADST bit to "0"
Interrupt Request Generation Timing	Completion of A/D conversion
Analog Input Pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), AN0 to AN7 (8 pins) ⁽¹⁾
Reading of Result of A/D Converter	Read one of 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.

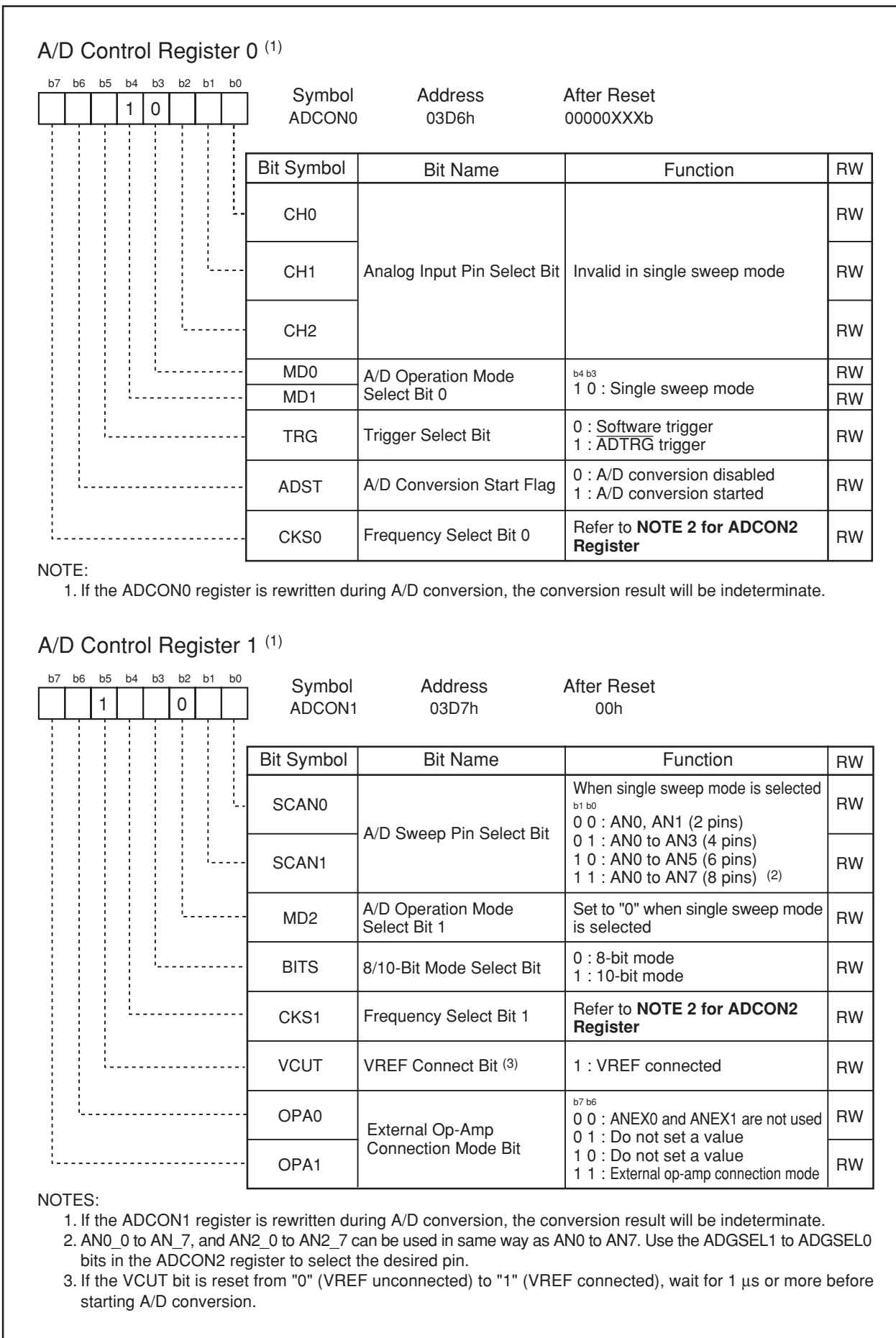


Figure 15.6 ADCON0 Register and ADCON1 Register in Single Sweep Mode

15.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltage applied to selected pins is repeatedly converted to a digital code.

Table 15.5 lists the specifications of repeat sweep mode 0. Figure 15.7 shows the ADCON0 and ADCON1 registers in repeat sweep mode 0.

Table 15.5 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"> • When the TRG bit in the ADCON0 register is “0” (software trigger) The ADST bit in the ADCON0 register is set to “1” (A/D conversion starts) • When the TRG bit is “1” ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to “1” (A/D conversion starts)
A/D Conversion Stop Condition	Set the ADST bit to “0” (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), AN0 to AN7 (8 pins) ⁽¹⁾
Reading of Result of A/D Converter	Read one of 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.

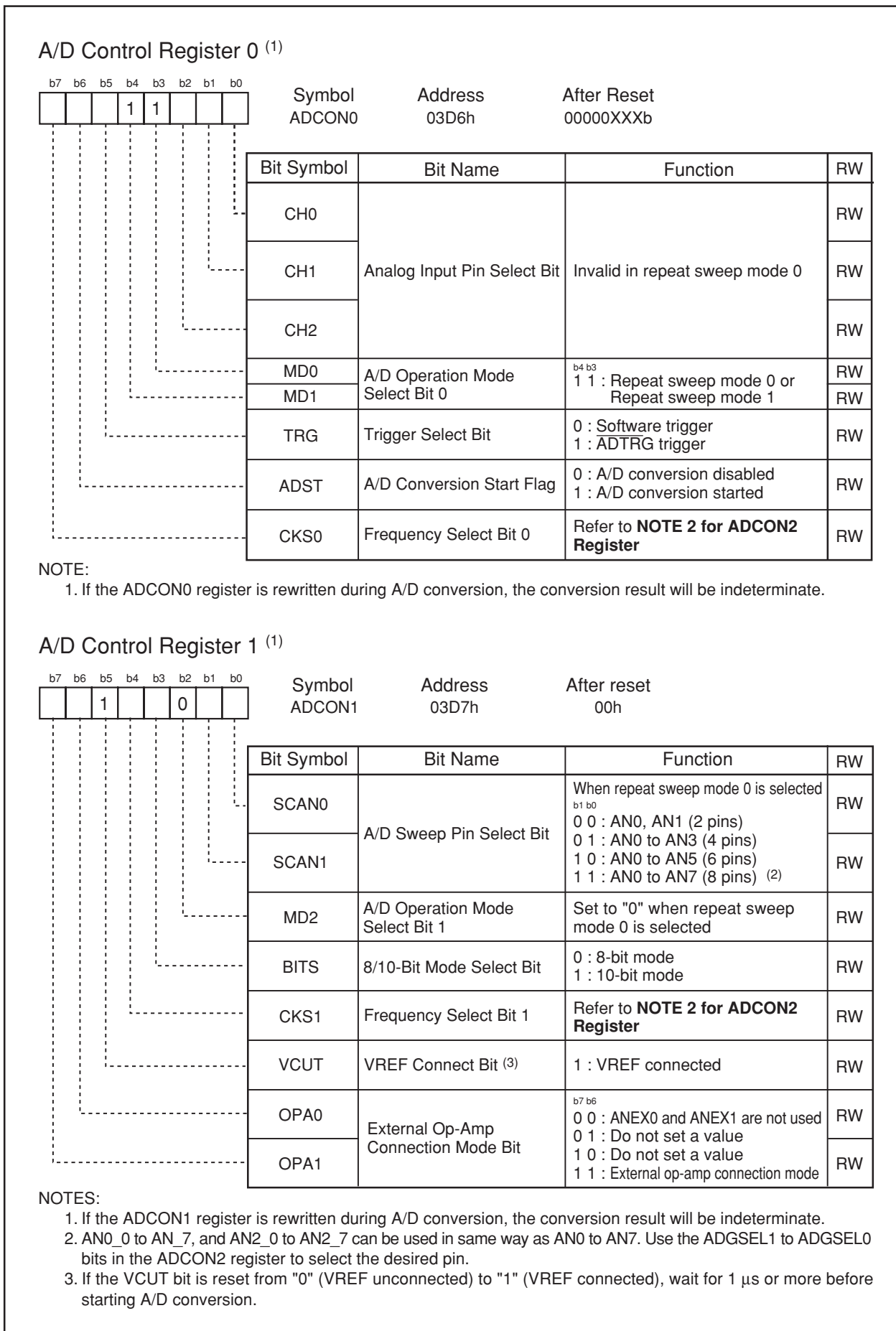


Figure 15.7 ADCON0 Register and ADCON1 Register in Repeat Sweep Mode 0

15.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage selectively applied to all pins is repeatedly converted to a digital code. Table 15.6 lists the specifications of repeat sweep mode 1. Figure 15.8 shows the ADCON0 and ADCON1 registers in repeat sweep mode 1.

Table 15.6 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"> When the TRG bit in the ADCON0 register is "0" (software trigger) The ADST bit in the ADCON0 register is set to "1" (A/D conversion starts) When the TRG bit is "1" ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to "1" (A/D conversion starts)
A/D Conversion Stop Condition	Set the ADST bit to "0" (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pins to be Given Priority when A/D Converted	Select from AN0 (1 pin), AN0 to AN1 (2 pins), AN0 to AN2 (3 pins), AN0 to AN3 (4 pins) ⁽¹⁾
Reading of Result of A/D Converter	Read one of 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.

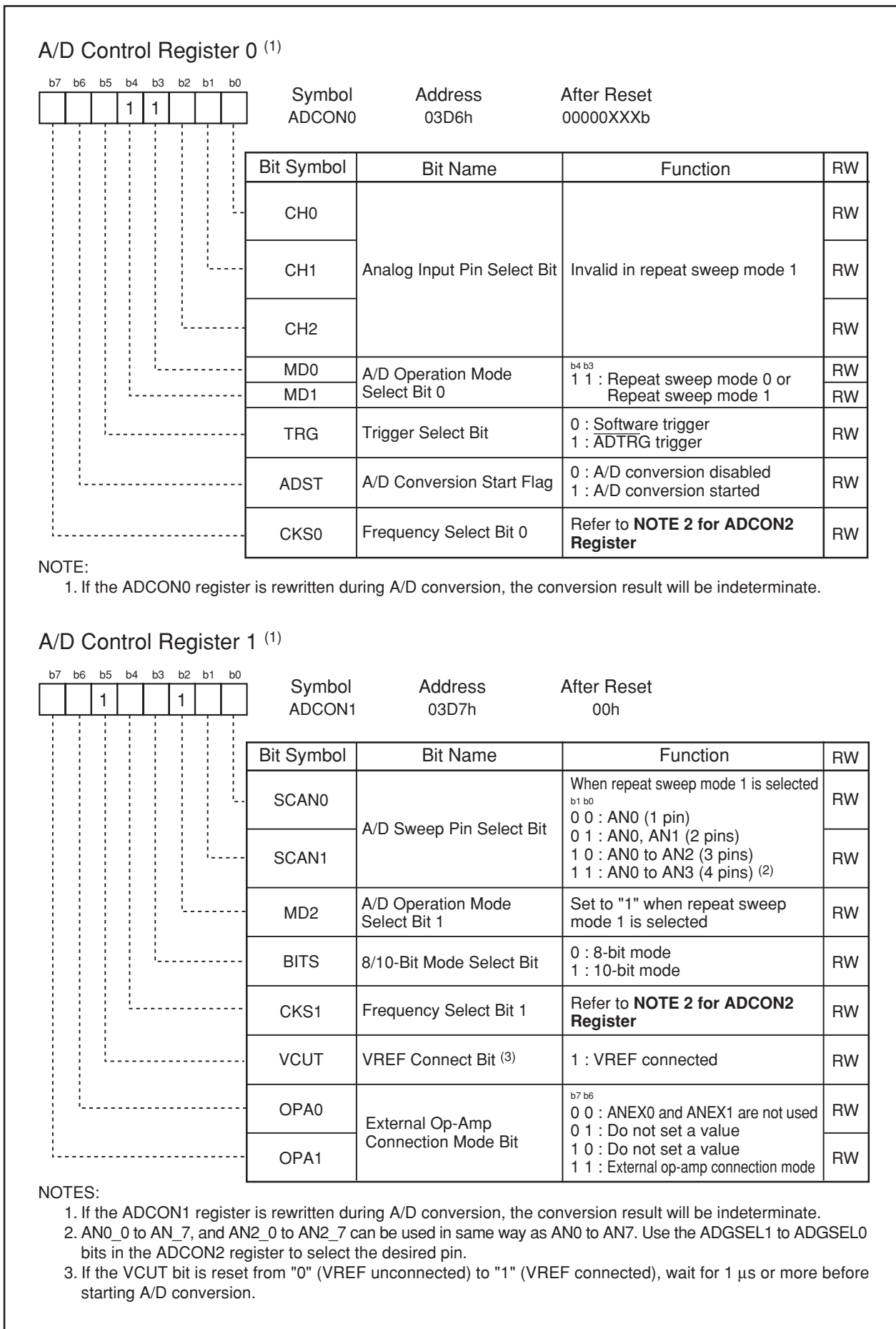


Figure 15.8 ADCON0 Register and ADCON1 Register in Repeat Sweep Mode 1

15.2 Function

15.2.1 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.

15.2.2 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.

15.2.3 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.

15.2.4 External Operation Amplifier (Op-Amp) Connection Mode

Multiple analog inputs can be amplified using a single external op-amp via the ANXE0 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) ⁽¹⁾ are output from the ANEX0 pin. Amplify this output with an external op-amp before sending it back to the ANEX1 pin. The A/D conversion result is stored in the corresponding ADi register. The A/D conversion speed depends on the response characteristics of the external op-amp. Figure 15.9 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.

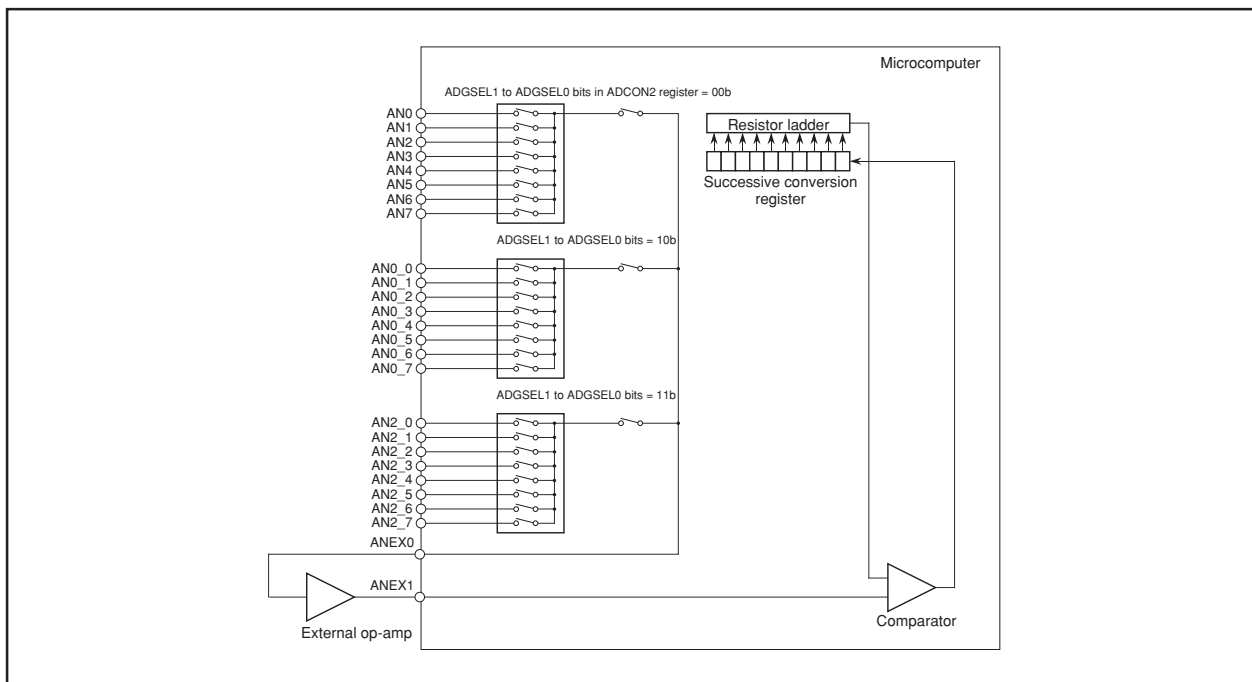


Figure 15.9 External Op-Amp Connection

15.2.5 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).

15.2.6 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 15.10 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 A/D converter's resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

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

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

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

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

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

Figure 15.10 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(XIN) = 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,

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

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

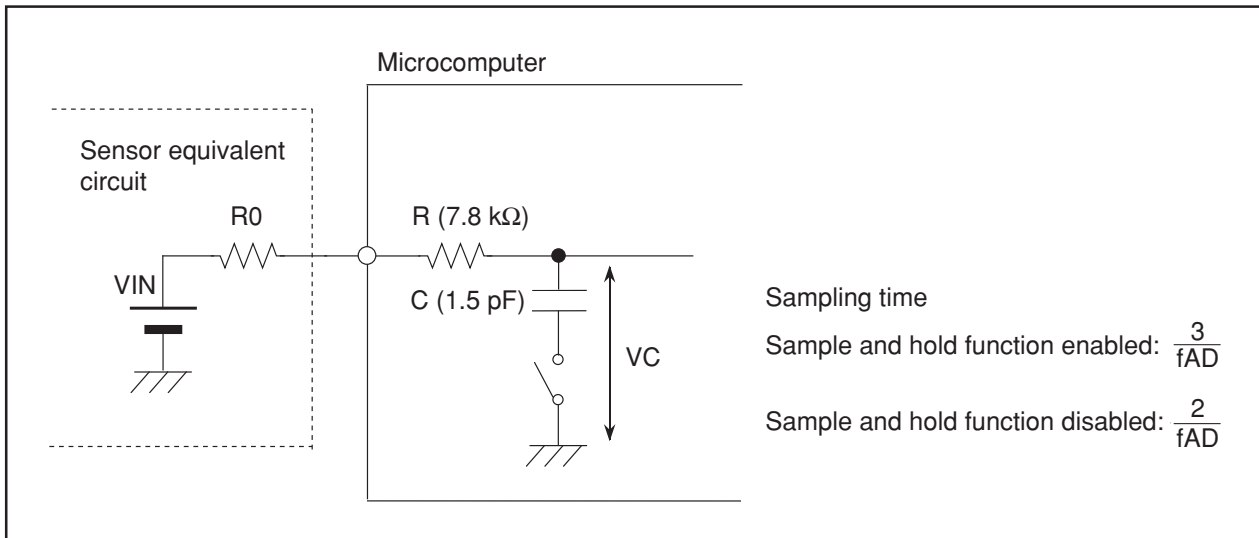


Figure 15.10 Analog Input Pin and External Sensor Equivalent Circuit

16. D/A Converter

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

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

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

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

VREF : reference voltage

Table 16.1 lists the performance of the D/A converter. Figure 16.1 shows the block diagram of the D/A converter. Figure 16.2 shows the D/A converter-related registers. Figure 16.3 shows the D/A converter equivalent circuit.

Table 16.1 D/A Converter Performance

Item	Performance
D/A conversion Method	R-2R method
Resolution	8 bits
Analog Output Pin	2 (DA0 and DA1)

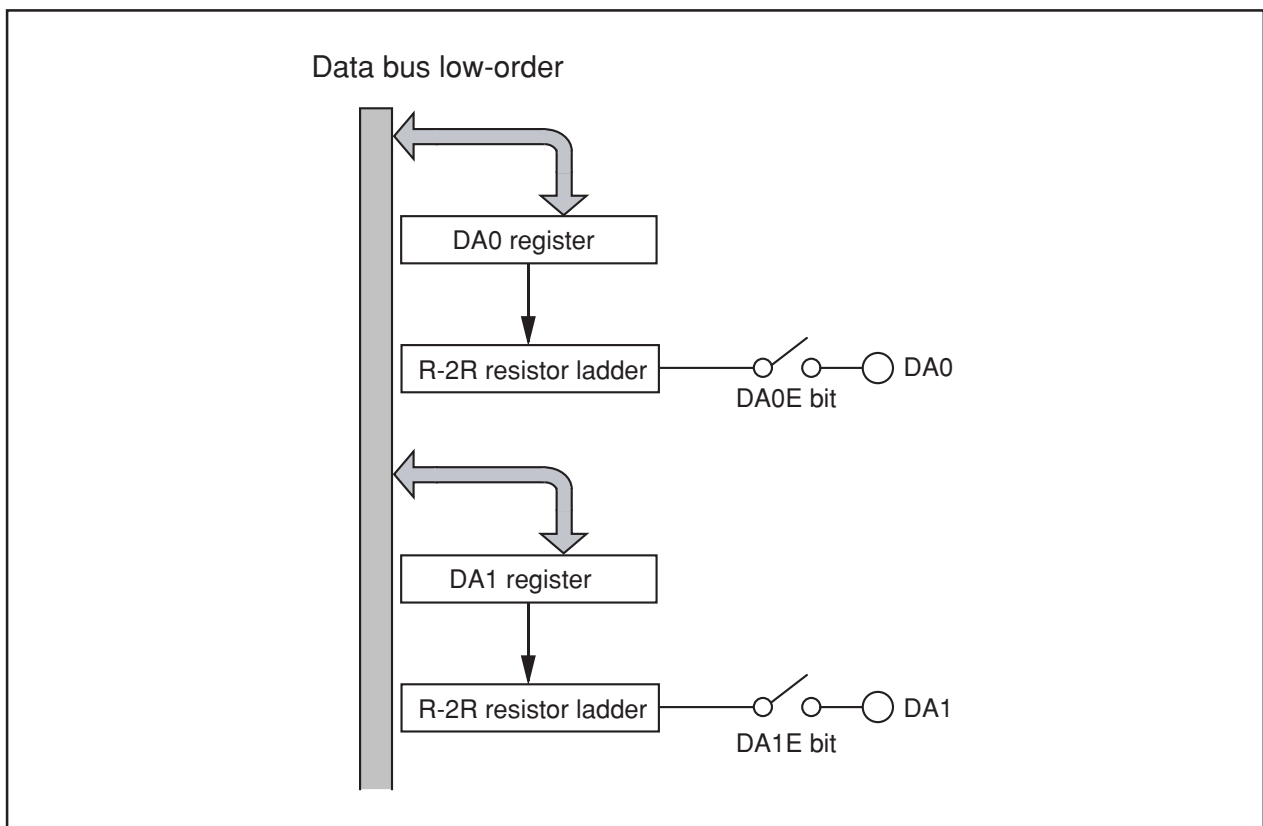


Figure 16.1 D/A Converter Block Diagram

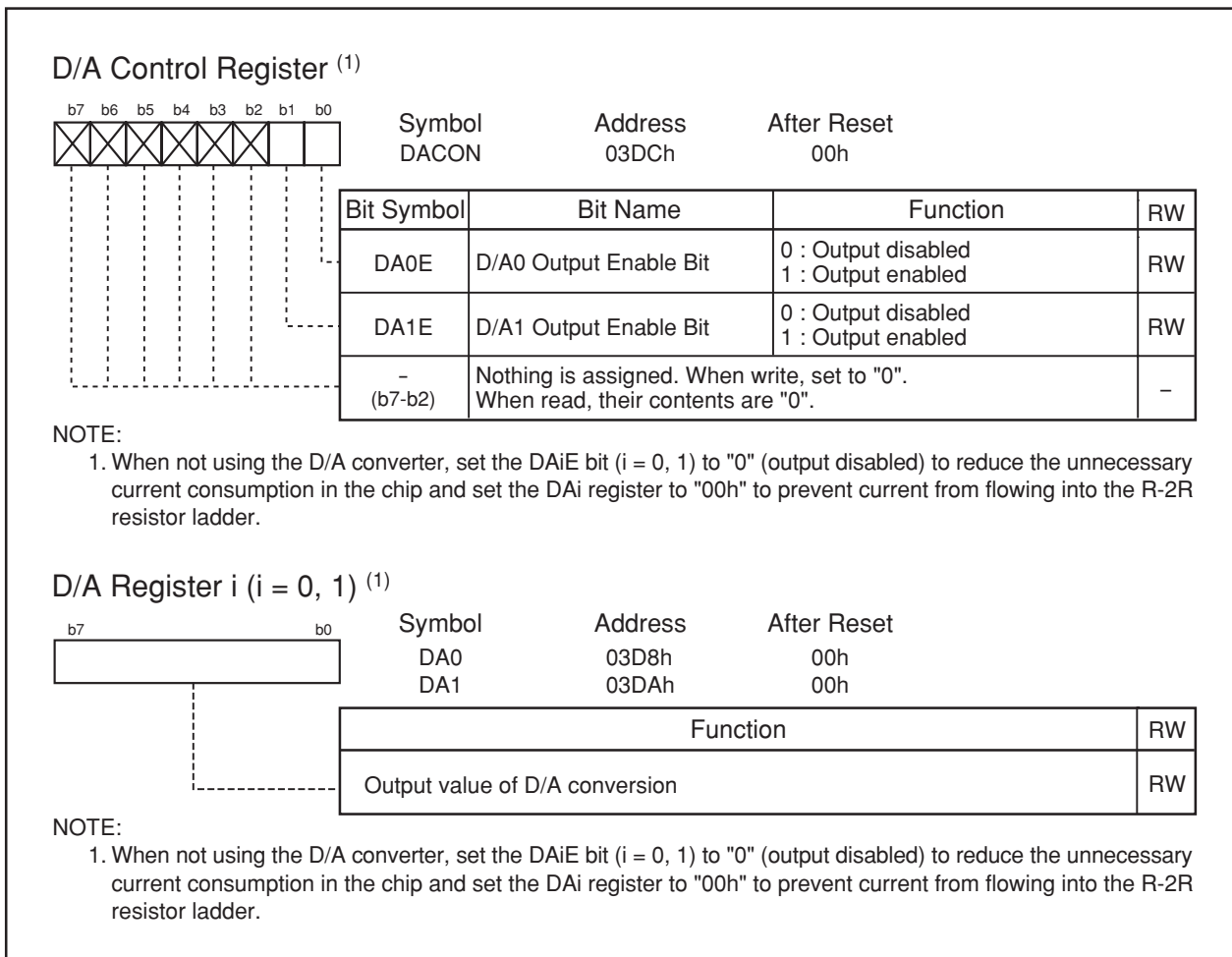


Figure 16.2 DACON Register, DA0 and DA1 Registers

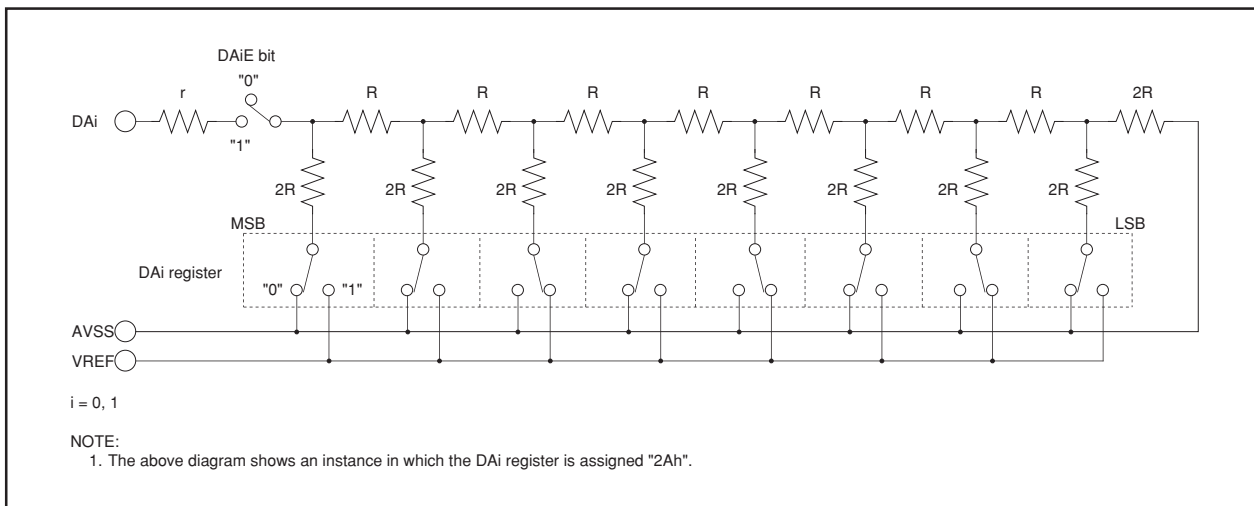


Figure 16.3 D/A Converter Equivalent Circuit

17. 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 17.1 shows the block diagram of the CRC circuit. Figure 17.2 shows the CRC-related registers. Figure 17.3 shows the calculation example using the CRC operation.

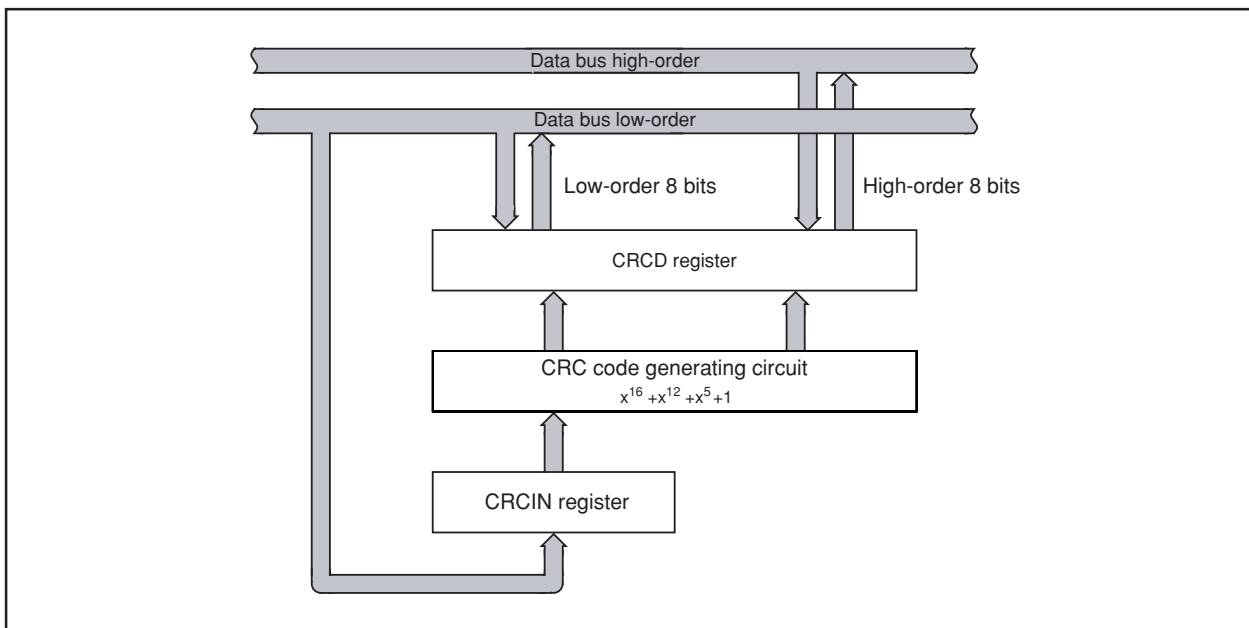


Figure 17.1 CRC Circuit Block Diagram

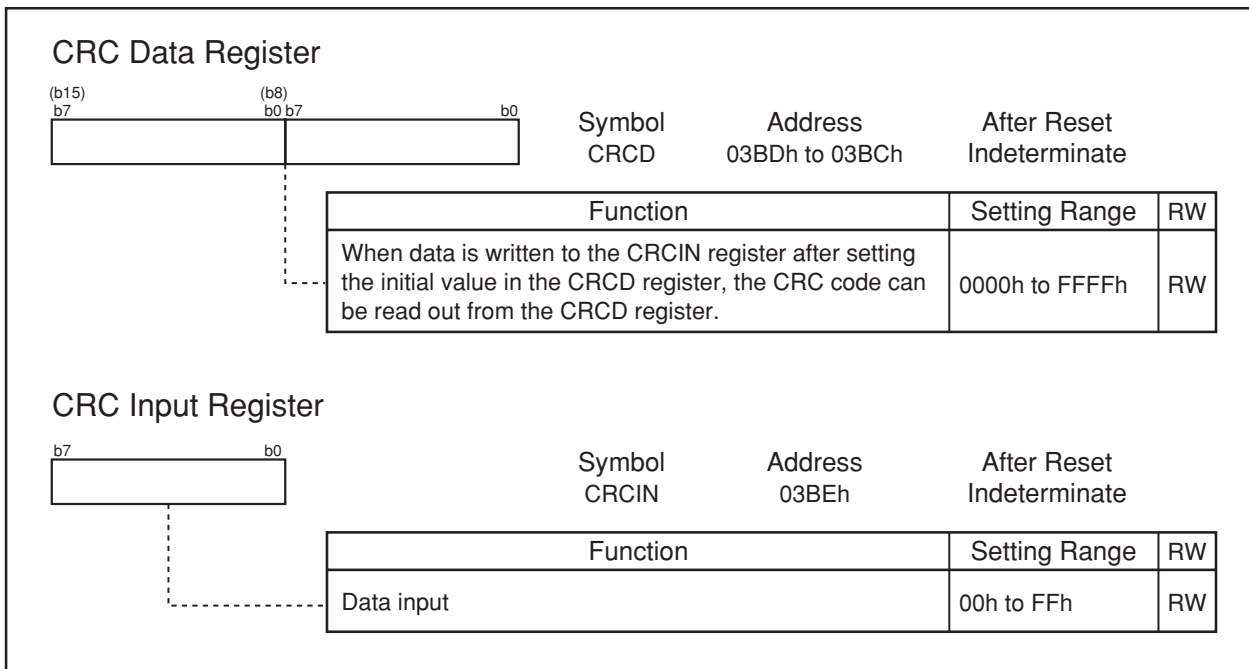


Figure 17.2 CRCD Register and CRCIN Register

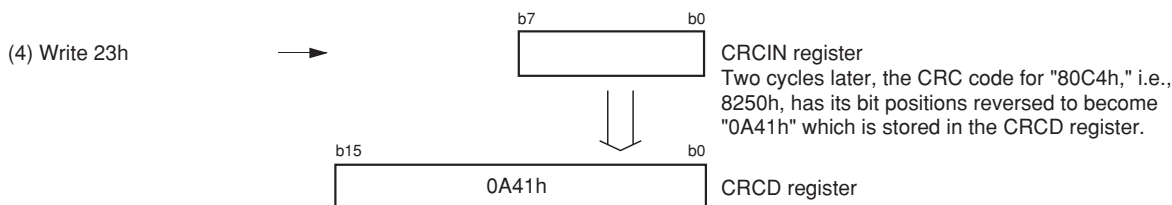
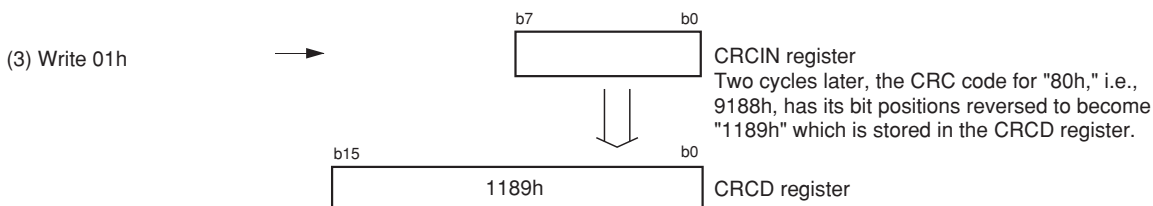
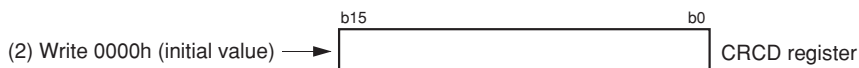
Setup procedure and CRC operation when generating CRC code "80C4h"

- 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)

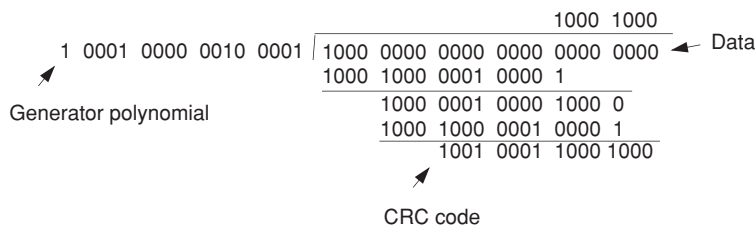
- Setting procedure

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



- Details of CRC operation

In the case of (3) above, the value written to the CRCIN register "01h (00000001b)" has its bit positions reversed to become "10000000b". The value "1000 0000 0000 0000 0000b" derived from that by adding 16 digits and the initial value of the CRCD register, "0000h" are added. The result is divided by the generator polynomial using modulo-2 arithmetic.



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

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

The value "0001 0001 1000 1001b (1189h)" derived from the remainder "1001 0001 1000 1000b (9188h)" by reversing its bit positions may be read from the CRCD register.

If operation (4) above is performed subsequently, the value written to the CRCIN register "23h (00100011b)" has its bit positions reversed to become "11000100b". The value "1100 0100 0000 0000 0000b" derived from that by adding 16 digits and the remainder in (3) "1001 0001 1000 1000b" which is left in the CRCD register are added, the result of which is divided by the generator polynomial using modulo-2 arithmetic. The value "0000 1010 0100 0001b (0A41h)" derived from the remainder by reversing its bit positions may be read from the CRCD register.

Figure 17.3 CRC Calculation

18. CAN Module

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

Figure 18.1 shows a block diagram of the CAN module.

External CAN bus driver and receiver are required.

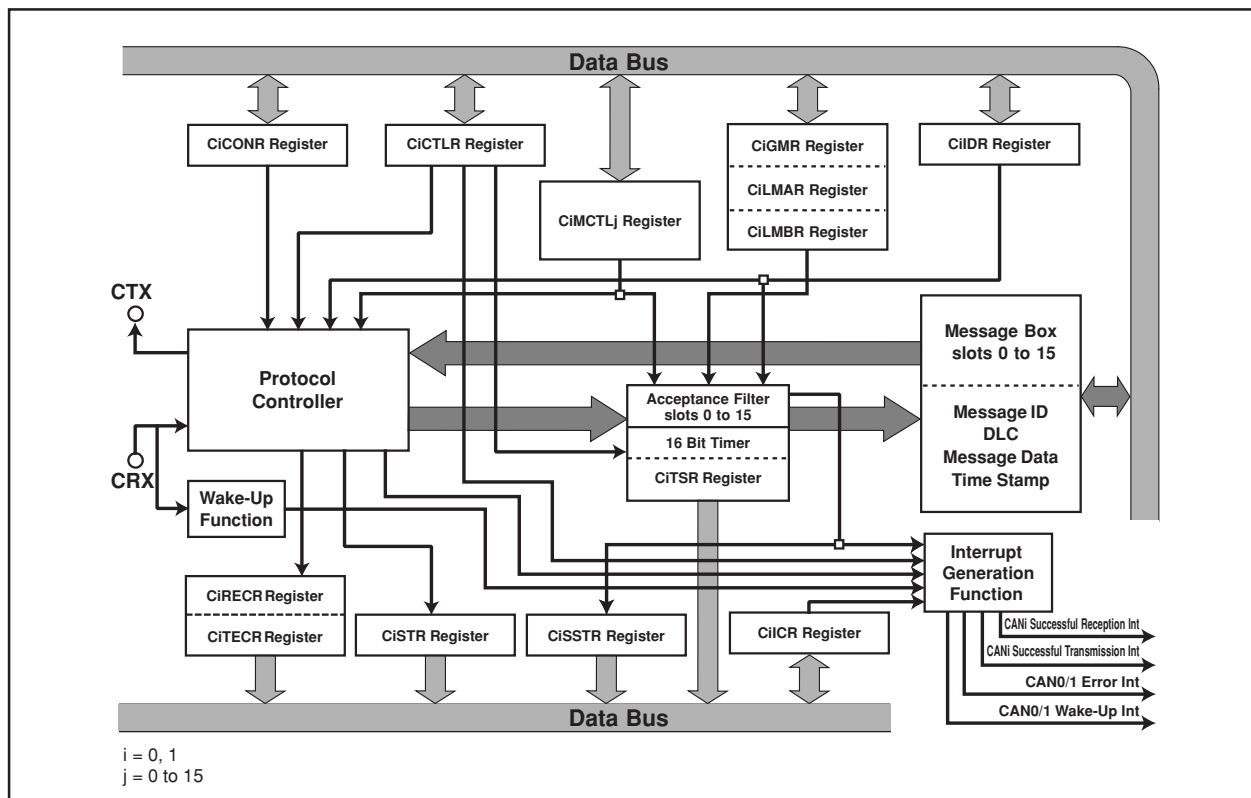


Figure 18.1 CAN Module Block Diagram

- 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.

18.1 CAN Module-Related Registers

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

18.1.1 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.

18.1.2 Acceptance Mask Registers

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

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

18.1.3 CAN SFR Registers

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

Explanation of each register is given below.

18.2 CANi Message Box (i = 0, 1)

Table 18.1 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 CiCTRL register.

Table 18.1 Memory Mapping of CANi Message Box

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

i = 0, 1

n = 0 to 15: the number of the slot

Figures 18.2 and 18.3 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.

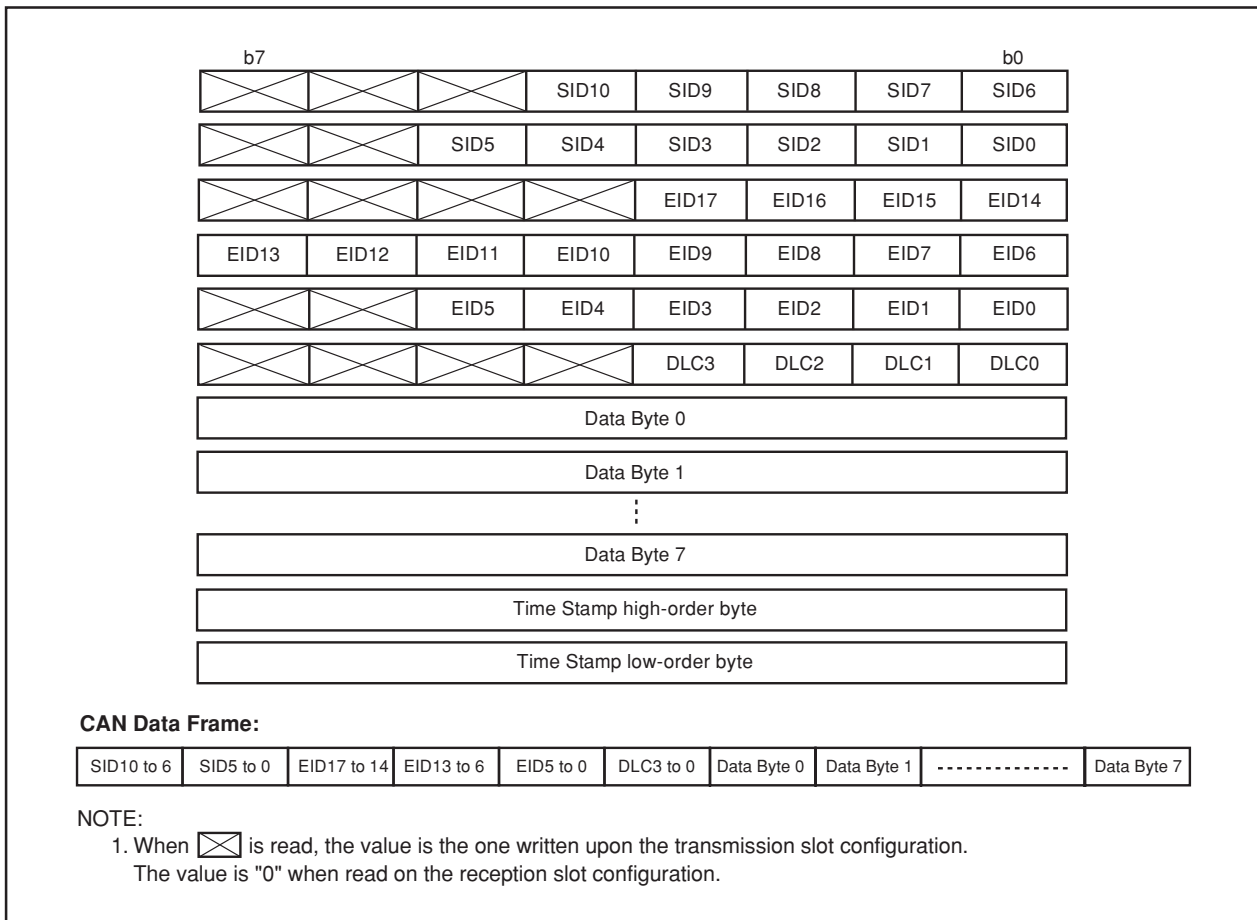


Figure 18.2 Bit Mapping in Byte Access

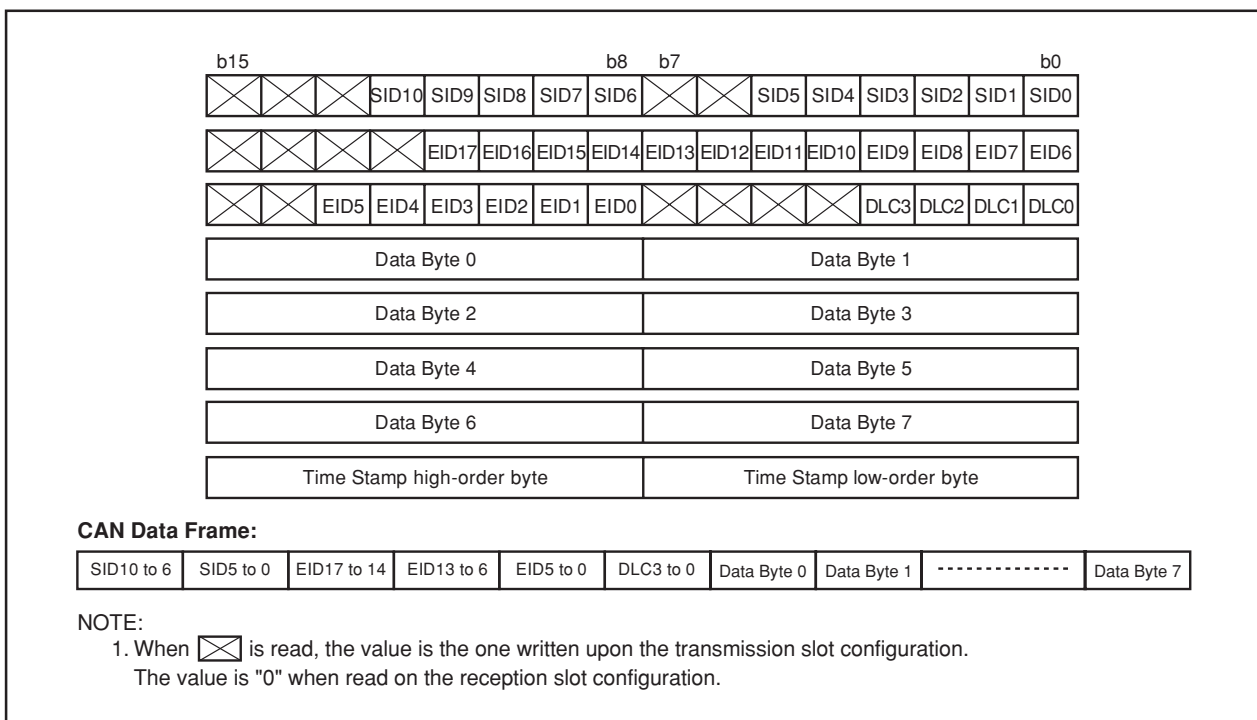


Figure 18.3 Bit Mapping in Word Access

18.3 Acceptance Mask Registers

Figures 18.4 and 18.5 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.

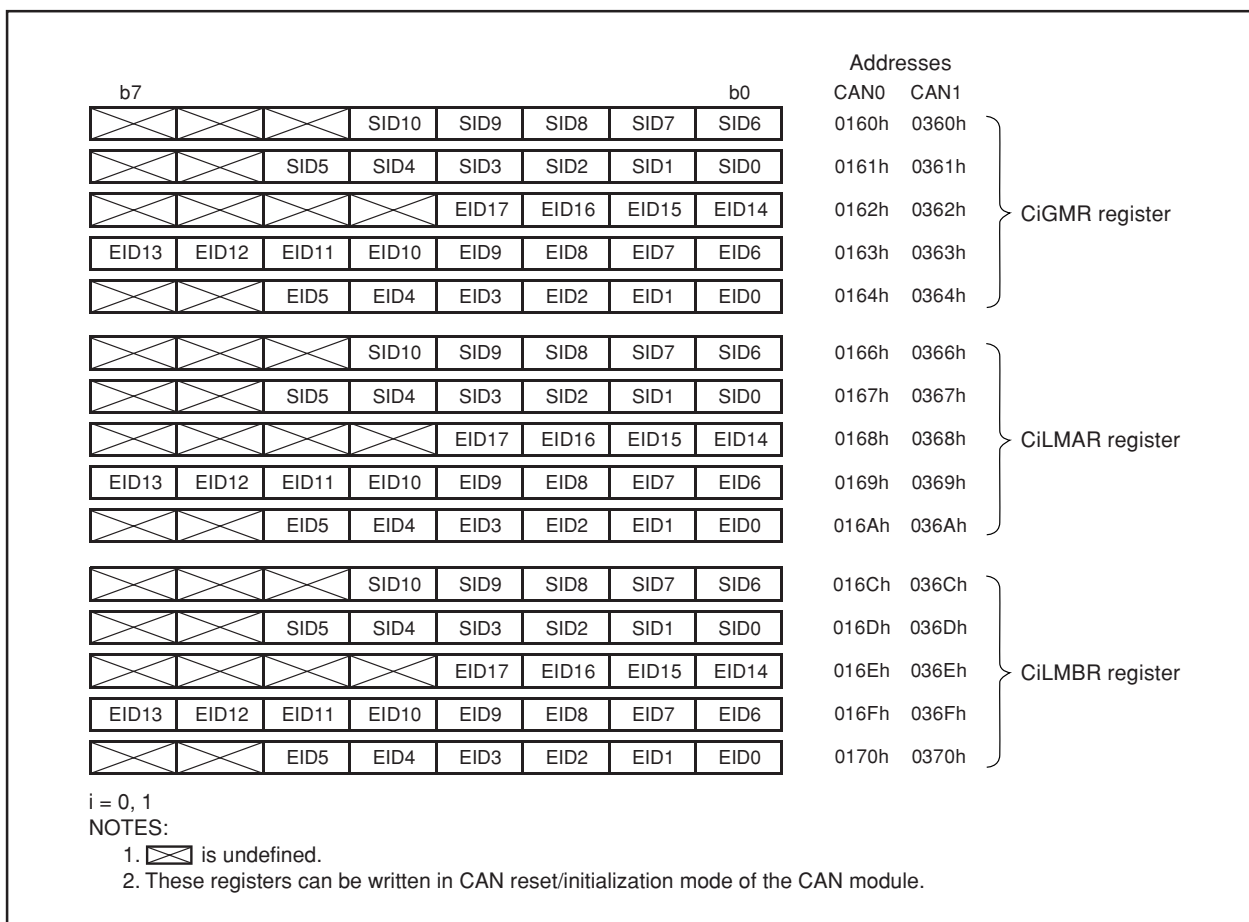


Figure 18.4 Bit Mapping of Mask Registers in Byte Access

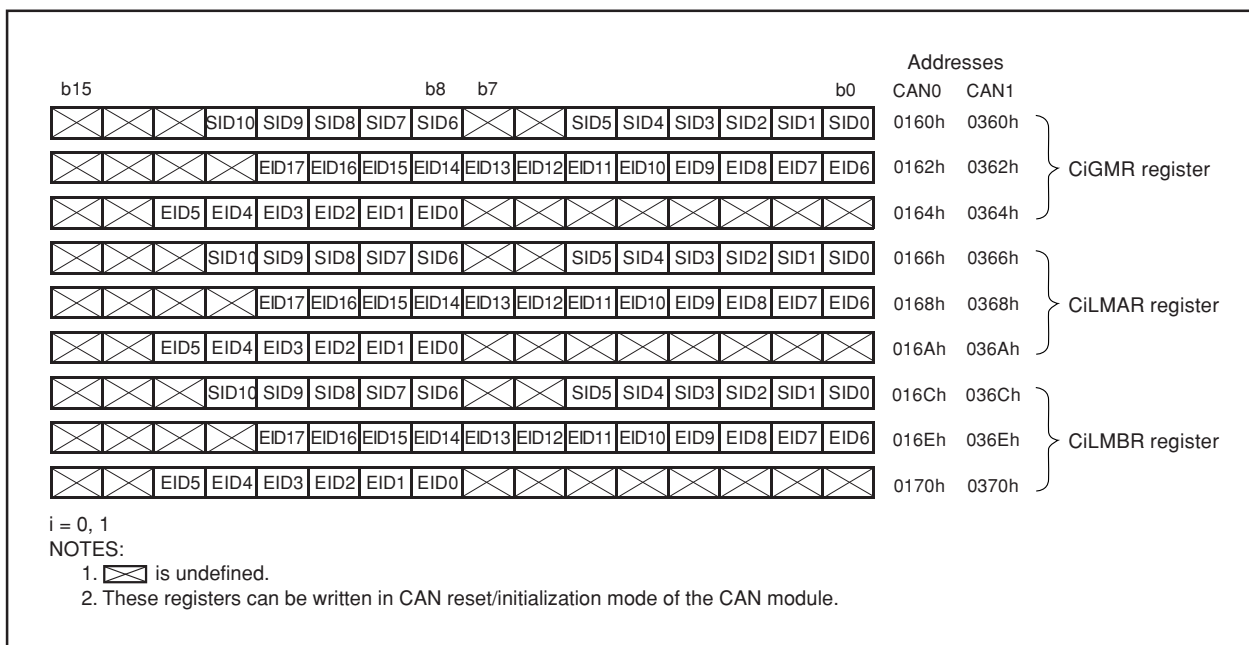


Figure 18.5 Bit Mapping of Mask Registers in Word Access

18.4 CAN SFR Registers

Figures 18.6 to 18.11 show the CAN SFR registers.

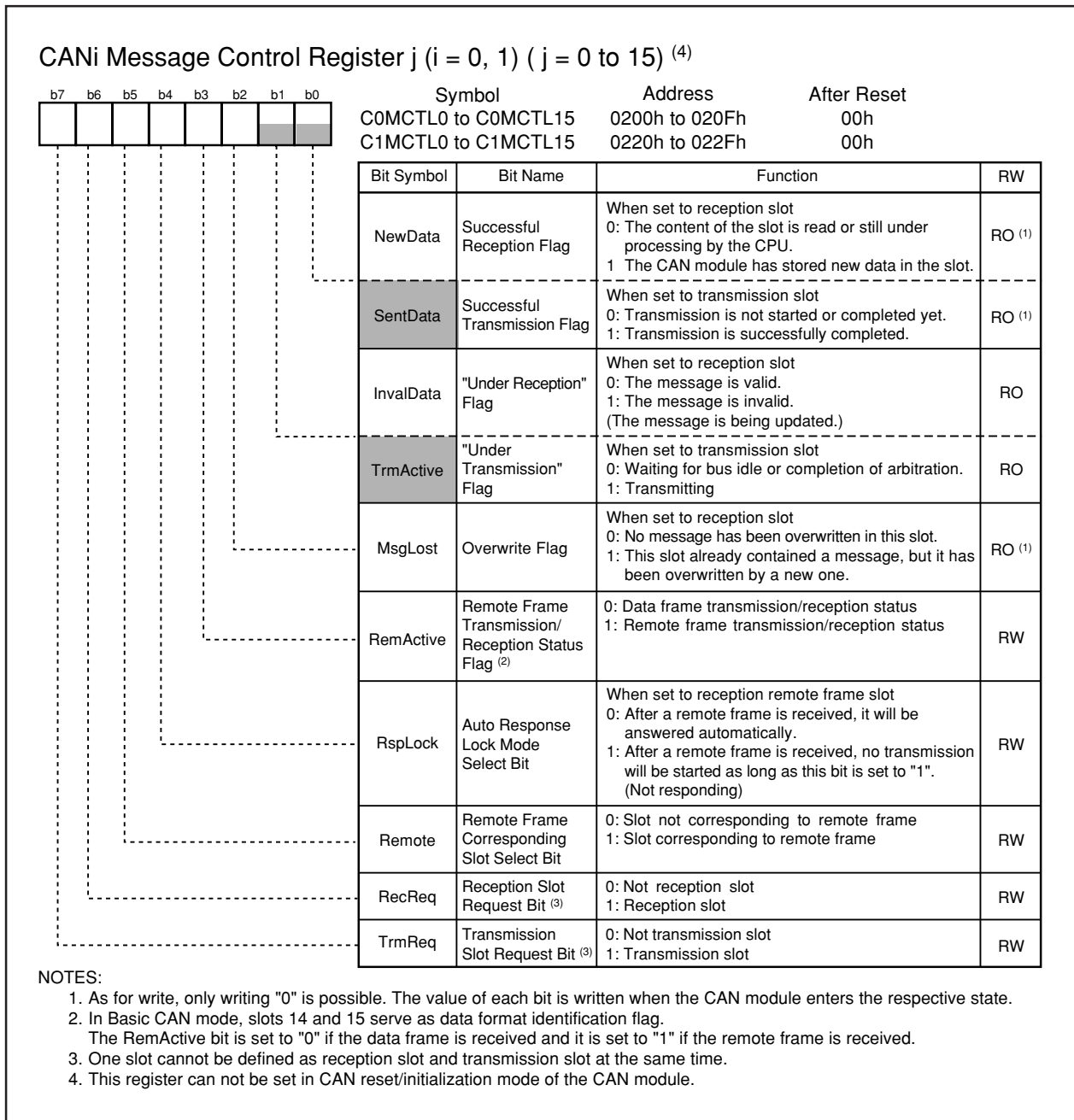


Figure 18.6 C0MCTLj and C1MCTLj Registers

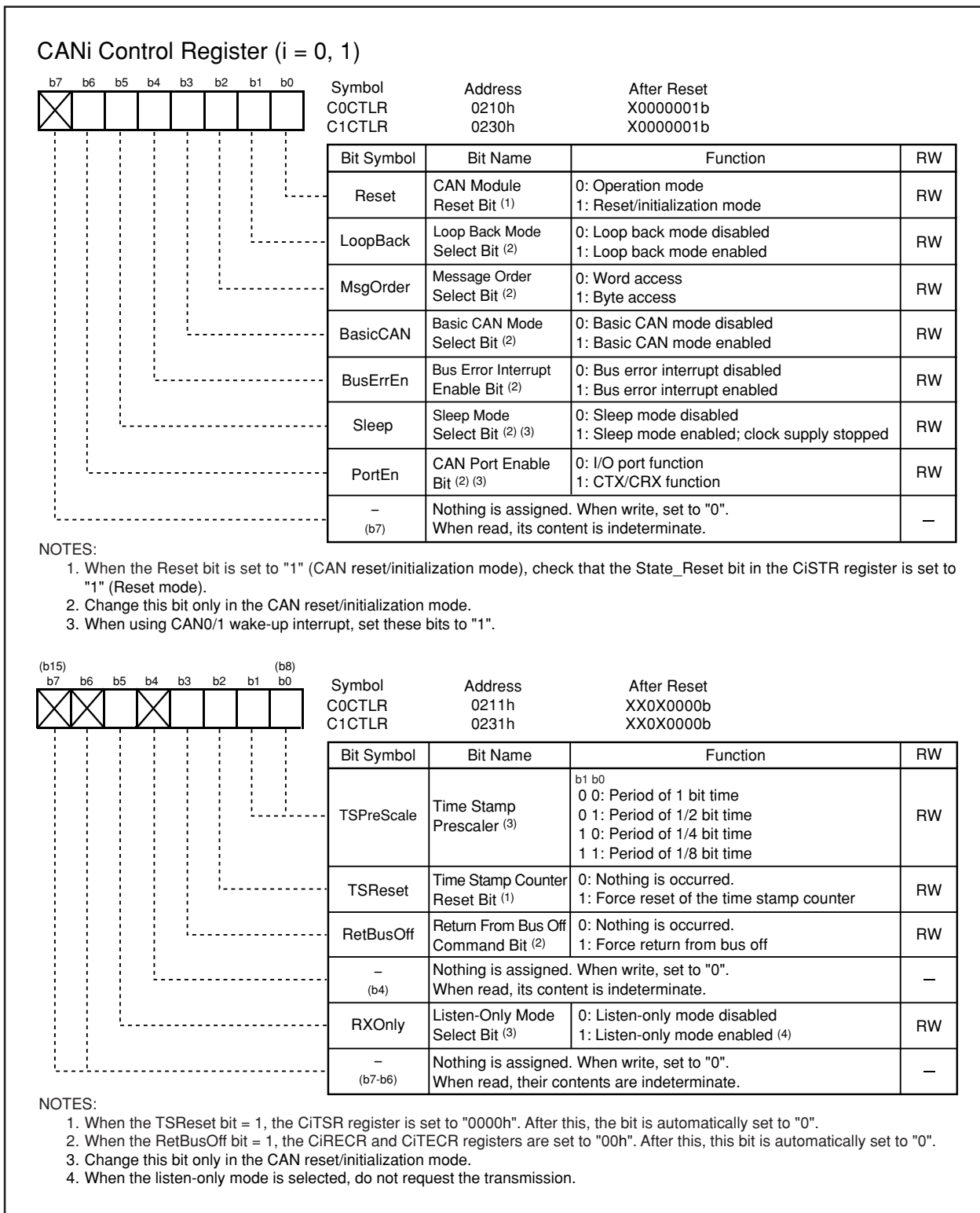


Figure 18.7 C0CTRL and C1CTRL Registers

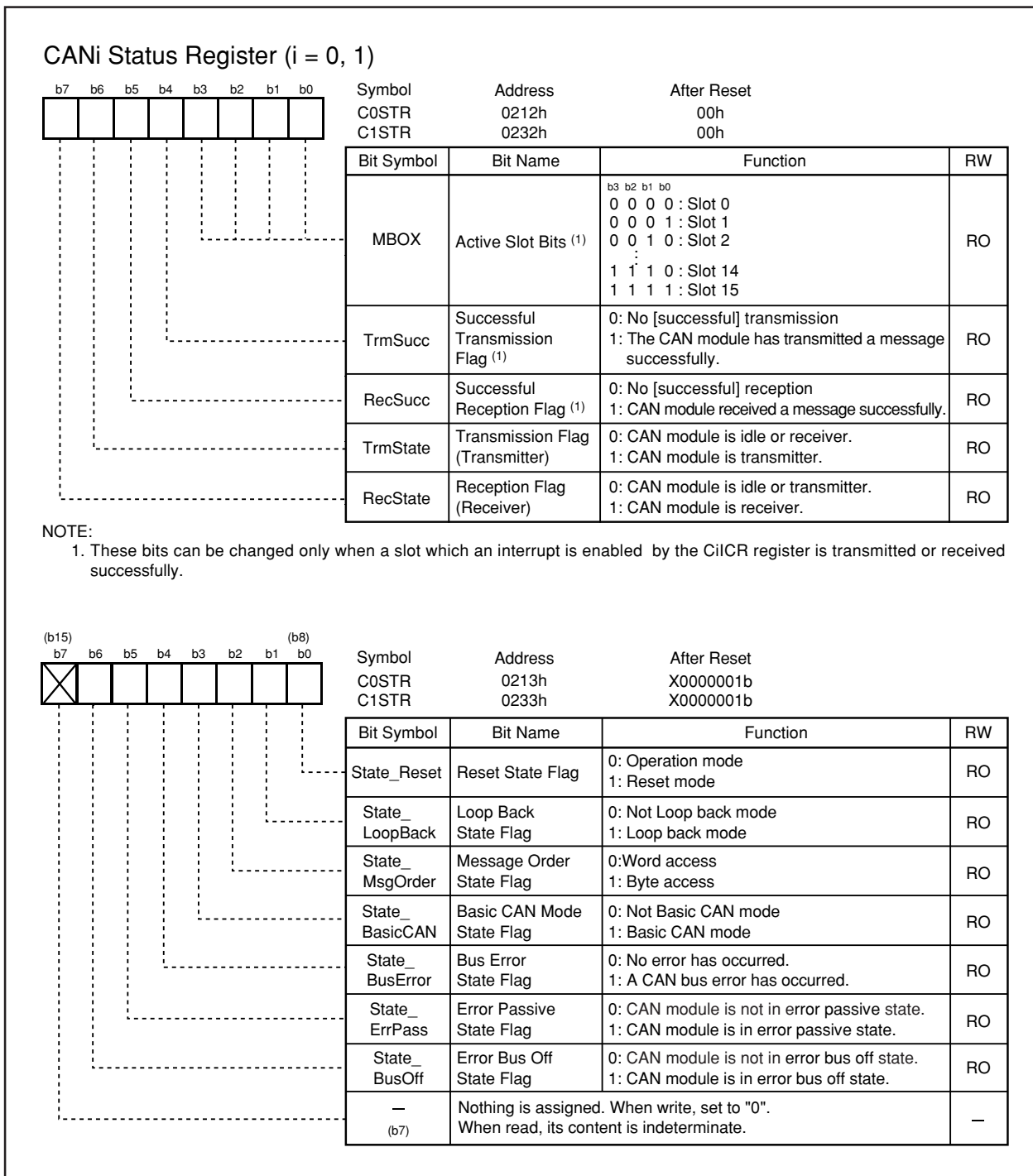


Figure 18.8 C0STR and C1STR Registers

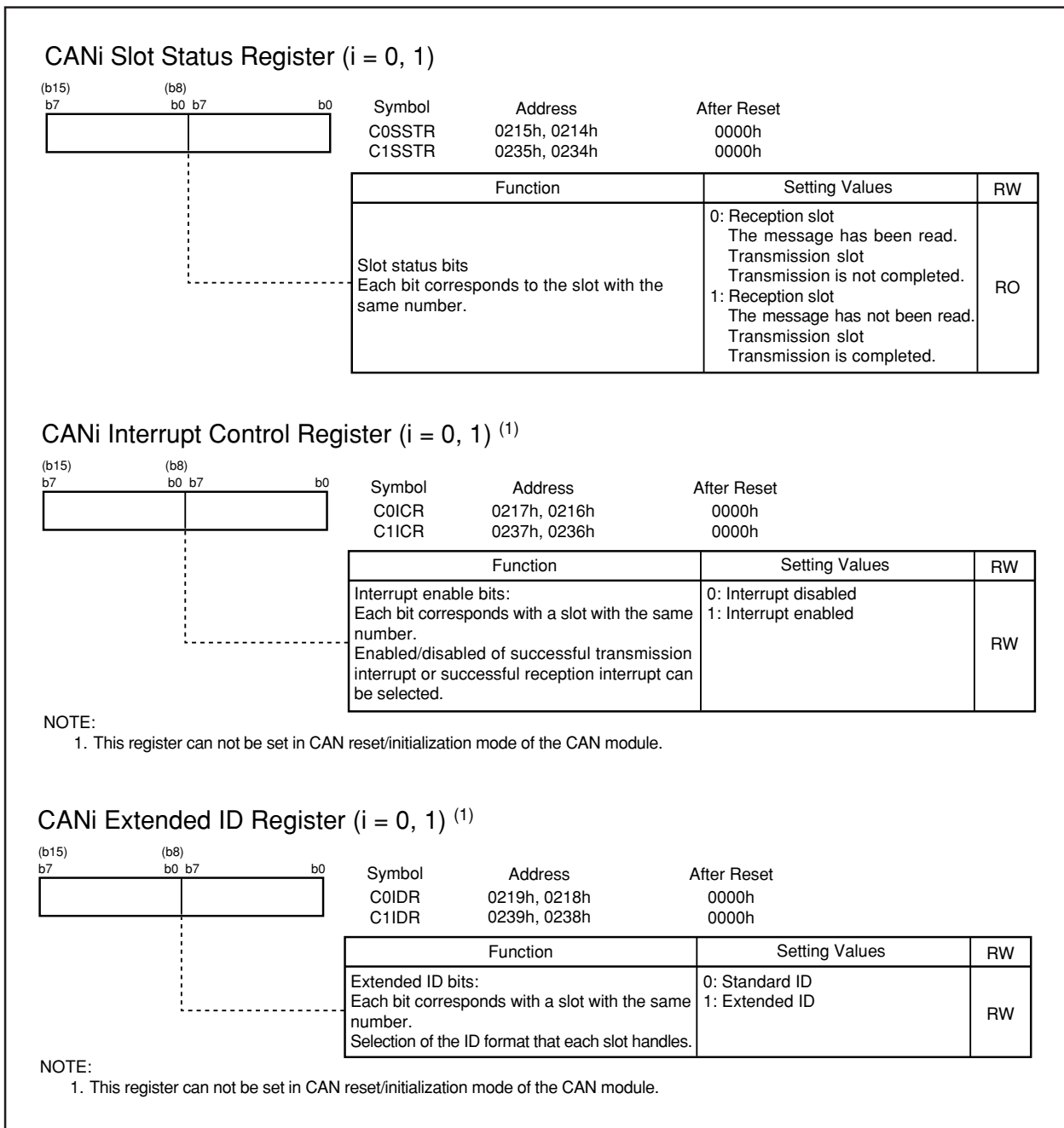


Figure 18.9 C0SSTR, C1SSTR Registers, C0ICR, C1ICR Registers, and C0IDR, C1IDR Registers

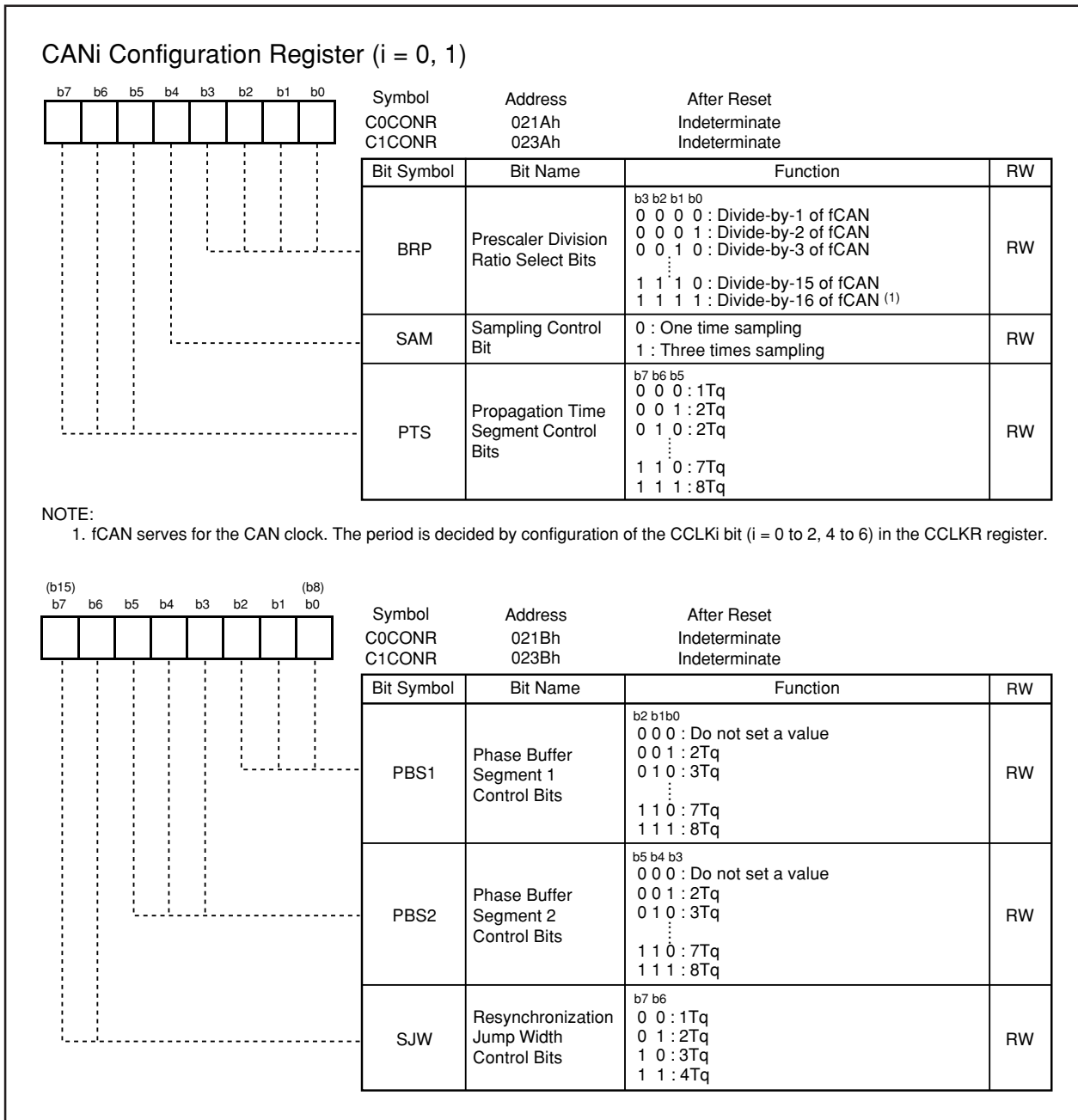


Figure 18.10 C0CONR and C1CONR Registers

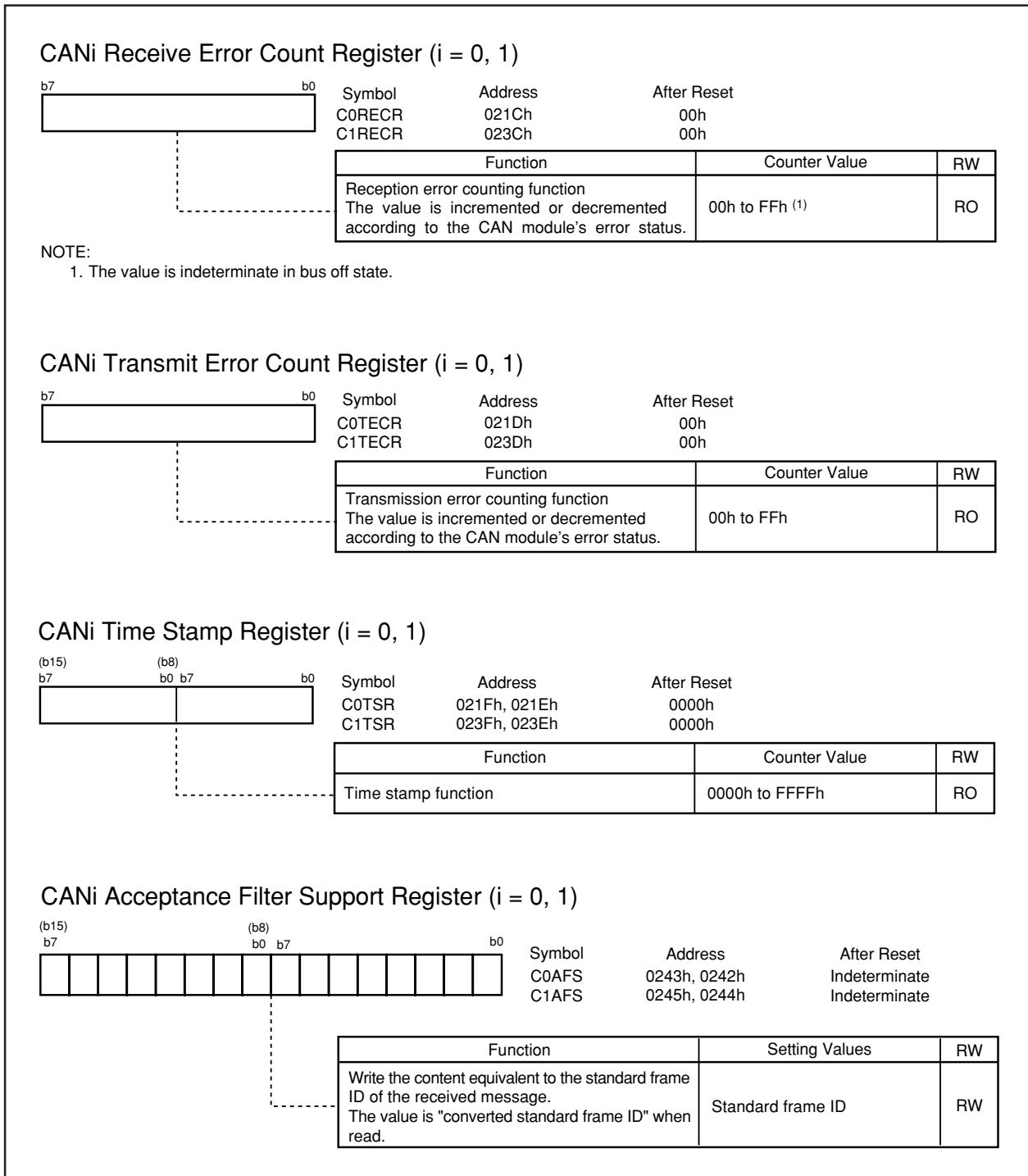


Figure 18.11 C0RECR, C1RECR Registers, C0TECR, C1TECR Registers, C0TSR, C1TSR Registers, and C0AFS, C1AFS Registers

18.5 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 18.12 shows transition between operational modes.

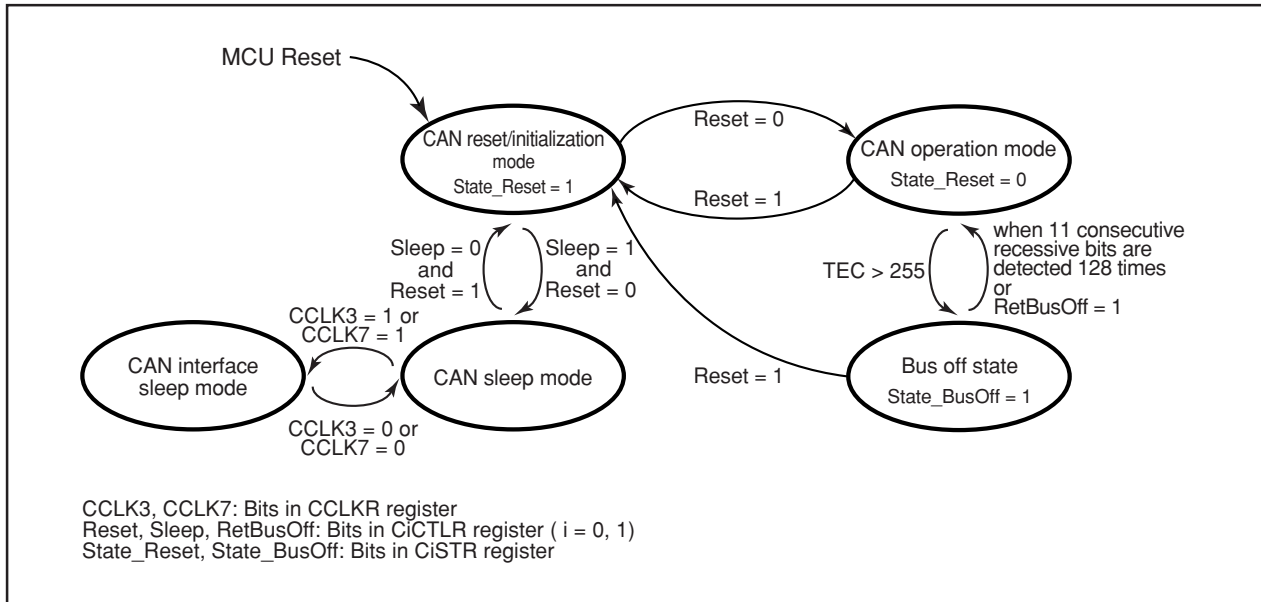


Figure 18.12 Transition Between Operational Modes

18.5.1 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.

18.5.2 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 18.13 shows sub modes of the CAN operation mode.

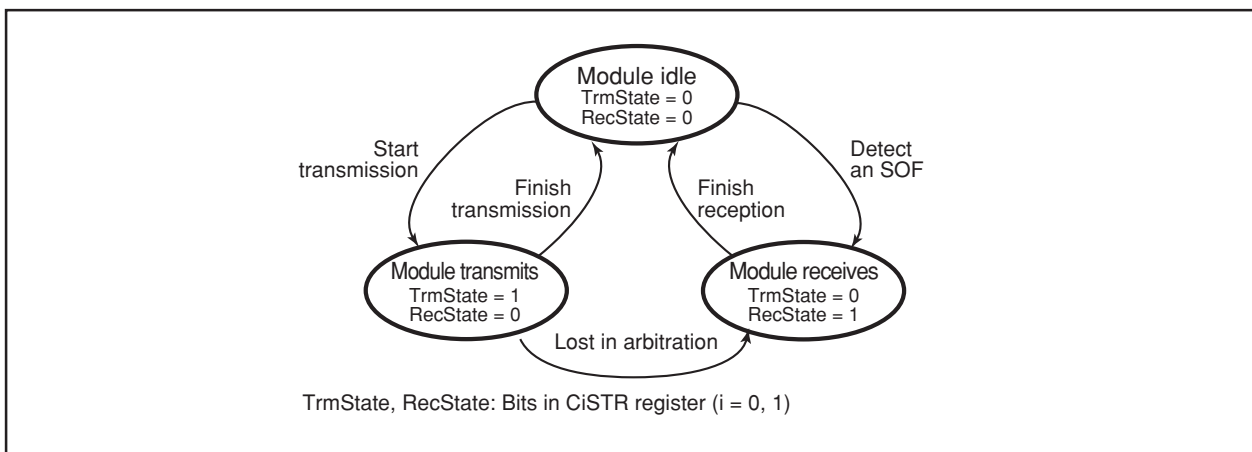


Figure 18.13 Sub Modes of CAN Operation Mode

18.5.3 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.

18.5.4 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.

18.5.5 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 CiCLR 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.

18.6 Configuration CAN Module System Clock

The M16C/6N Group (M16C/6NK, M16C/6NM) 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 7. Clock Generating Circuit.

Figure 18.14 shows a block diagram of the clock generating circuit of the CAN module system.

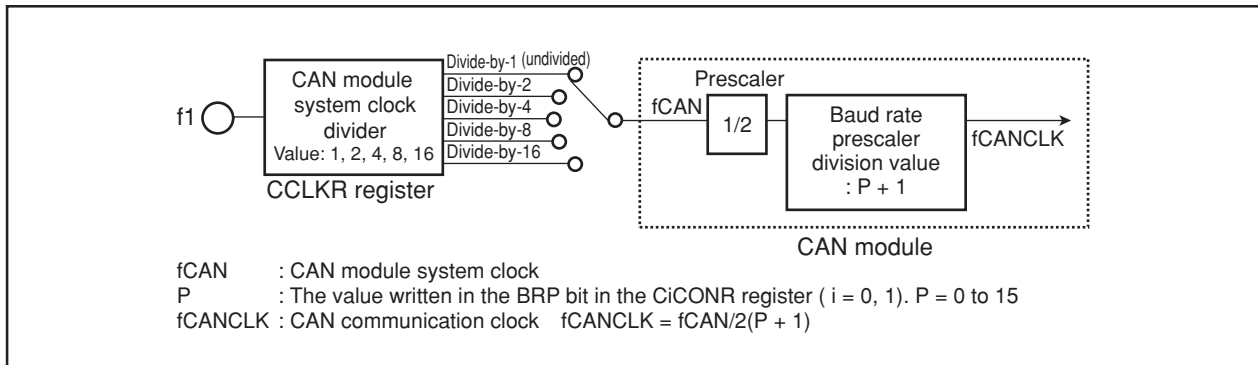


Figure 18.14 Block Diagram of CAN Module System Clock Generating Circuit

18.7 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 18.15 shows the bit timing.

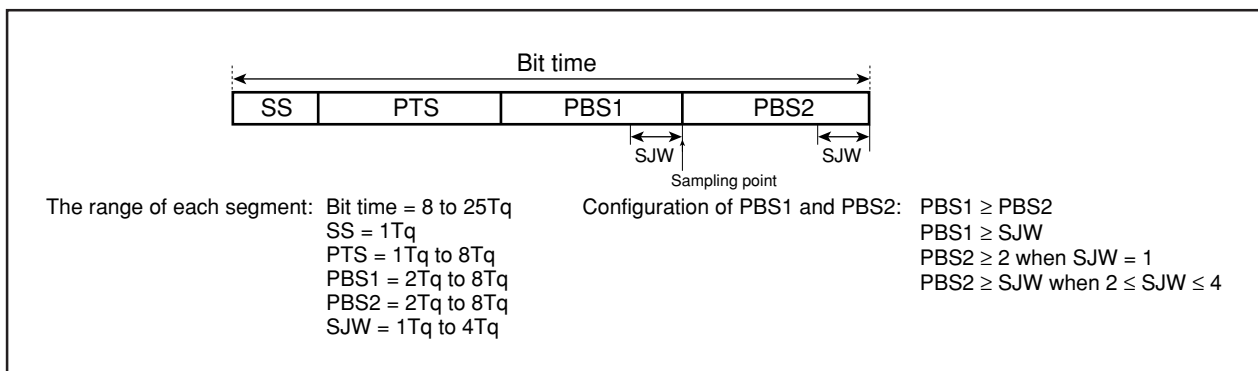


Figure 18.15 Bit Timing

18.8 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 18.2 shows the examples of bit-rate.

Table 18.2 Examples of Bit-rate

Bit-rate	24MHz	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)	–	–

NOTE:

- The number in () indicates a value of “fCAN division value” multiplied by “baud rate prescaler division value”.

■ 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:

- fCAN division value = 1, 2, 4, 8, 16
fCAN division value: a value selected in the CCLKR register
- 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)

18.9 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 18.16 shows correspondence of the mask registers and slots, Figure 18.17 shows the acceptance function.

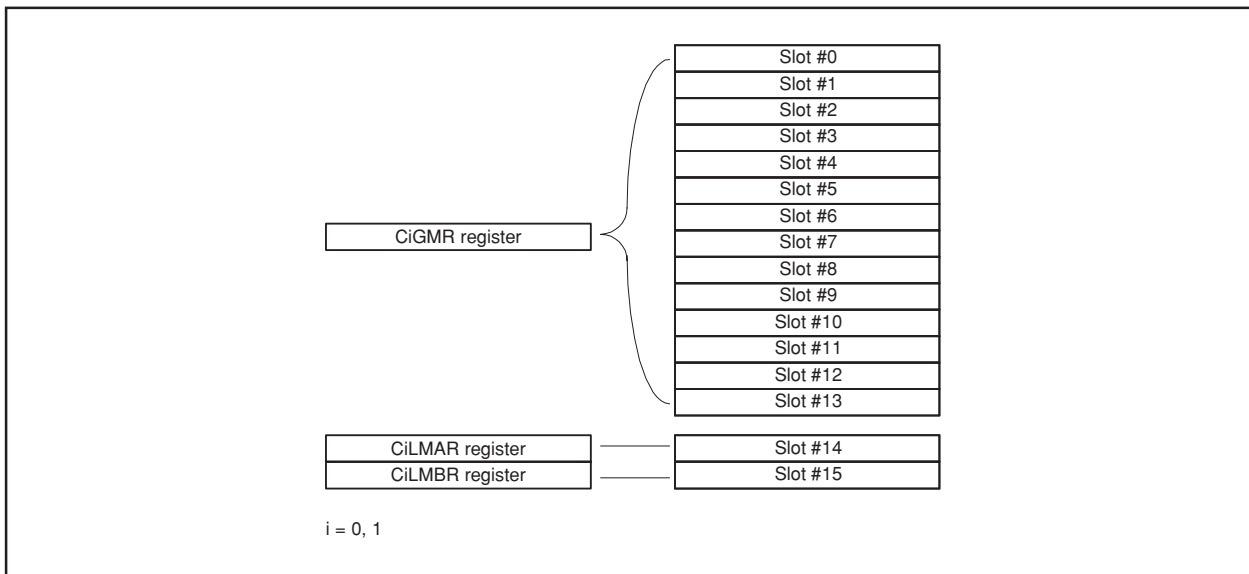


Figure 18.16 Correspondence of Mask Registers to Slots

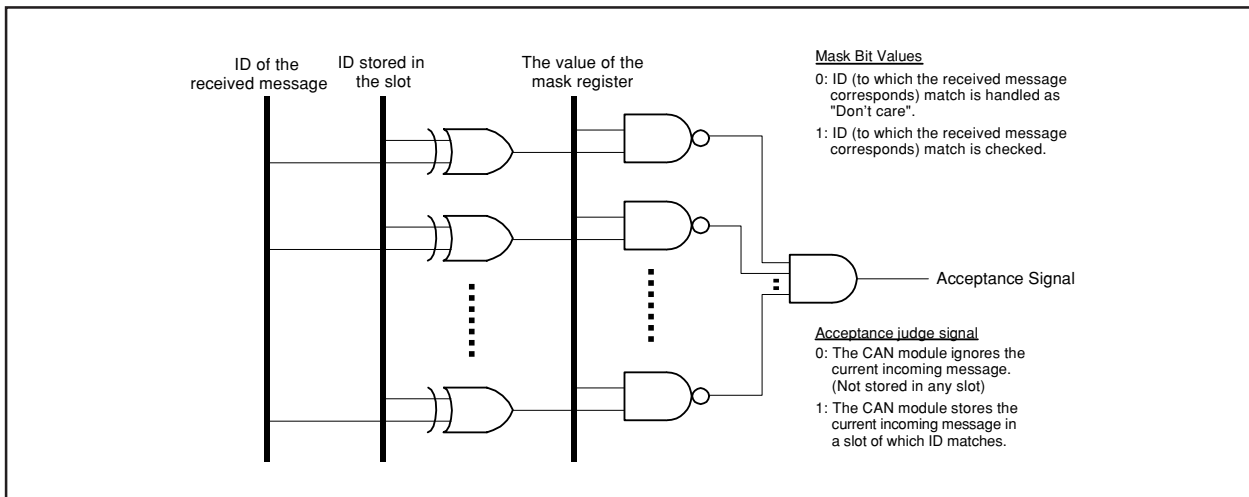


Figure 18.17 Acceptance Function

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.

18.10 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 18.18 shows the write and read of the CiAFS register in word access.

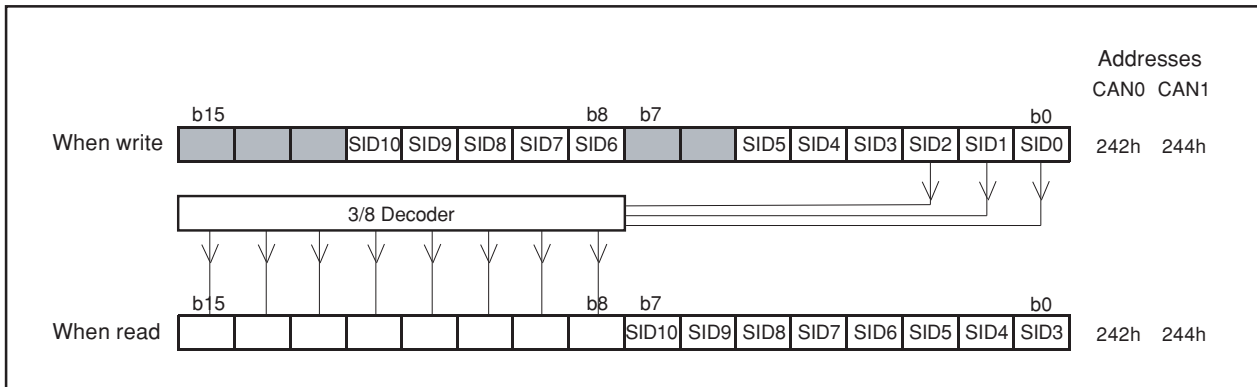


Figure 18.18 Write/read of CiAFS Register in Word Access

18.11 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 18.19 shows the operation of slots 14 and 15 in Basic CAN mode.

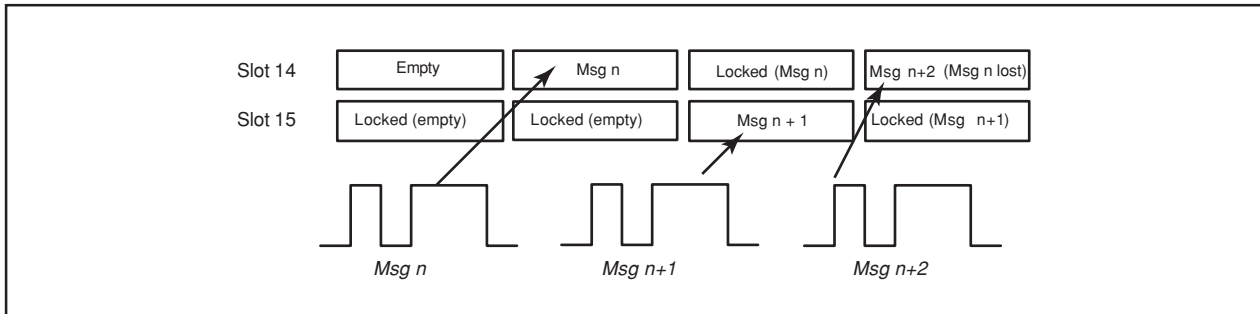


Figure 18.19 Operation of Slots 14 and 15 in Basic CAN Mode

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.

18.12 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.

18.13 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.

18.14 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.

18.15 Reception and Transmission

Table 18.3 shows configuration of CAN reception and transmission mode.

Table 18.3 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.

18.15.1 Reception

Figure 18.20 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.

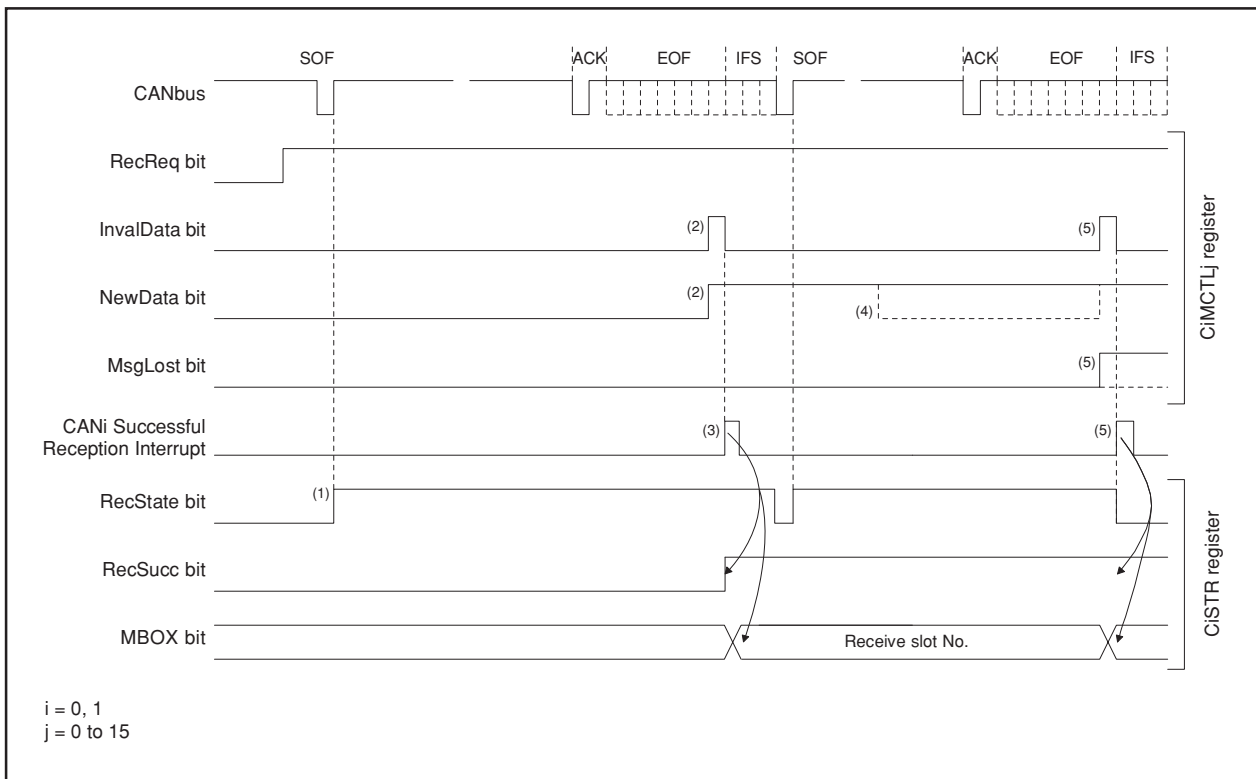


Figure 18.20 Timing of Receive Data Frame Sequence

- (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) If the NewData bit is set to "0" by a program or the next CAN message is received successfully before the receive request for the 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).

18.15.2 Transmission

Figure 18.21 shows the timing of the transmit sequence.

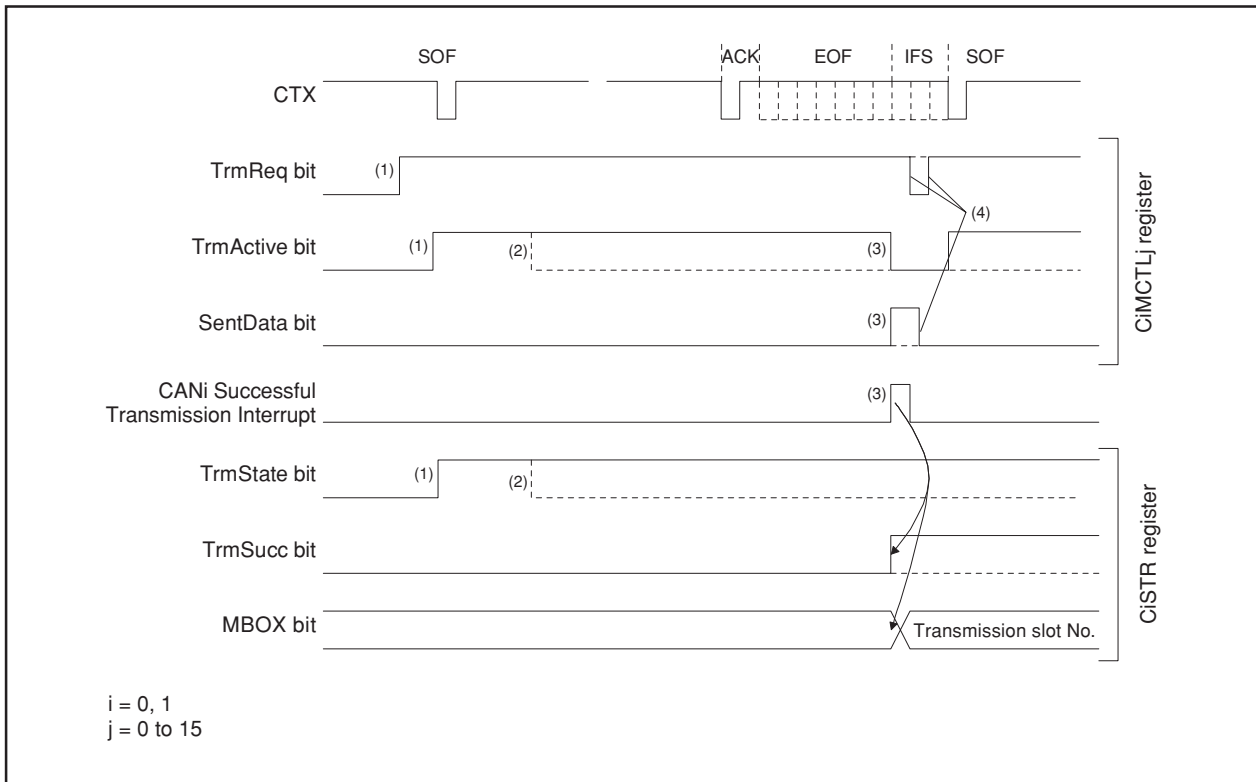


Figure 18.21 Timing of Transmit Sequence

- (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".

18.16 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.

19. Programmable I/O Ports

The programmable input/output ports (hereafter referred to simply as I/O ports) consist of 87 lines P0 to P10 in the 100-pin version and consist of 113 lines P0 to P14 in the 128-pin version. 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.

Table 19.1 lists the number of pins of the I/O ports of each package. Figures 19.1 to 19.5 show the I/O ports. Figure 19.6 shows the I/O pins.

Each pin functions as an I/O port or a peripheral function input/output 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, SI/O4 output 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 SI/O4 and D/A converter is directed for output no matter how the corresponding direction bit is set.

Table 19.1 Number of Pins of I/O Ports of Each Package

	128-pin Version	100-pin Version
I/O Ports	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 (P8_5 is an input port) P9_0 to P9_7 P10_0 to P10_7 P11_0 to P11_7 P12_0 to P12_7 P13_0 to P13_7 P14_0, P14_1	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 (P8_5 is an input port) P9_0 to P9_7 P10_0 to P10_7
Total	113 pins	87 pins

19.1 PDi Register (100-pin Version: i = 0 to 10, 128-pin Version: i = 0 to 13)

Figure 19.7 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.

No direction register bit for P8_5 is available.

19.2 Pi Register (100-pin Version: i = 0 to 10, 128-pin Version: i = 0 to 13), PC14 Register

Figure 19.8 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.

About the port P14 (128-pin version), Figure 19.8 shows the PC14 register.

19.3 PURj Register (100-pin Version: j = 0 to 2, 128-pin Version: j = 0 to 3)

Figures 19.9 and 19.10 show 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.

When using the ports P11 to P14, set the PUR37 bit in the PUR3 register to "1" (P11 to P14 are usable).

19.4 PCR Register

Figure 19.11 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.

Table 19.2 lists an example connection of unused pins. Figure 19.12 shows an example connection of unused pins.

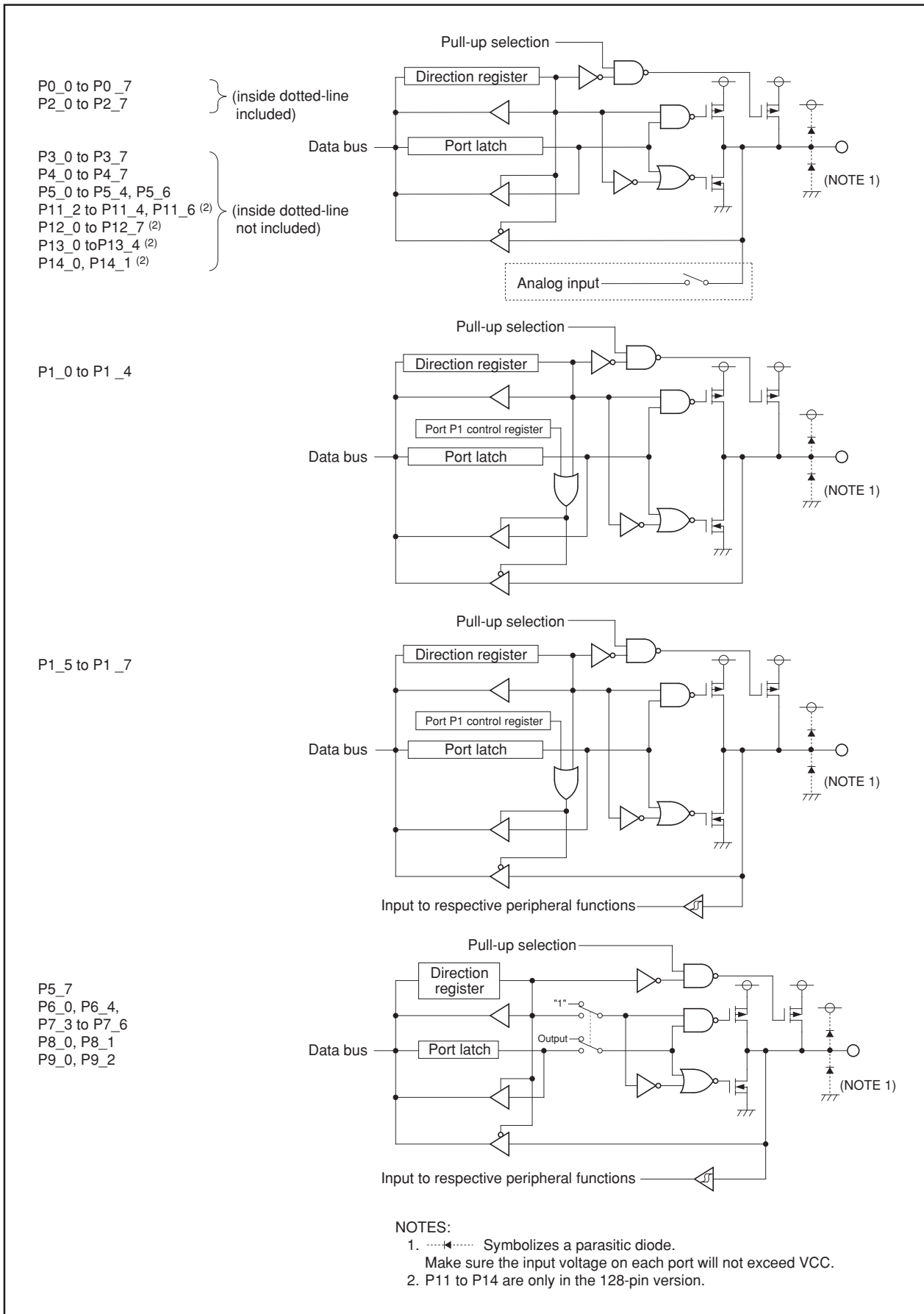


Figure19.1 I/O Ports (1)

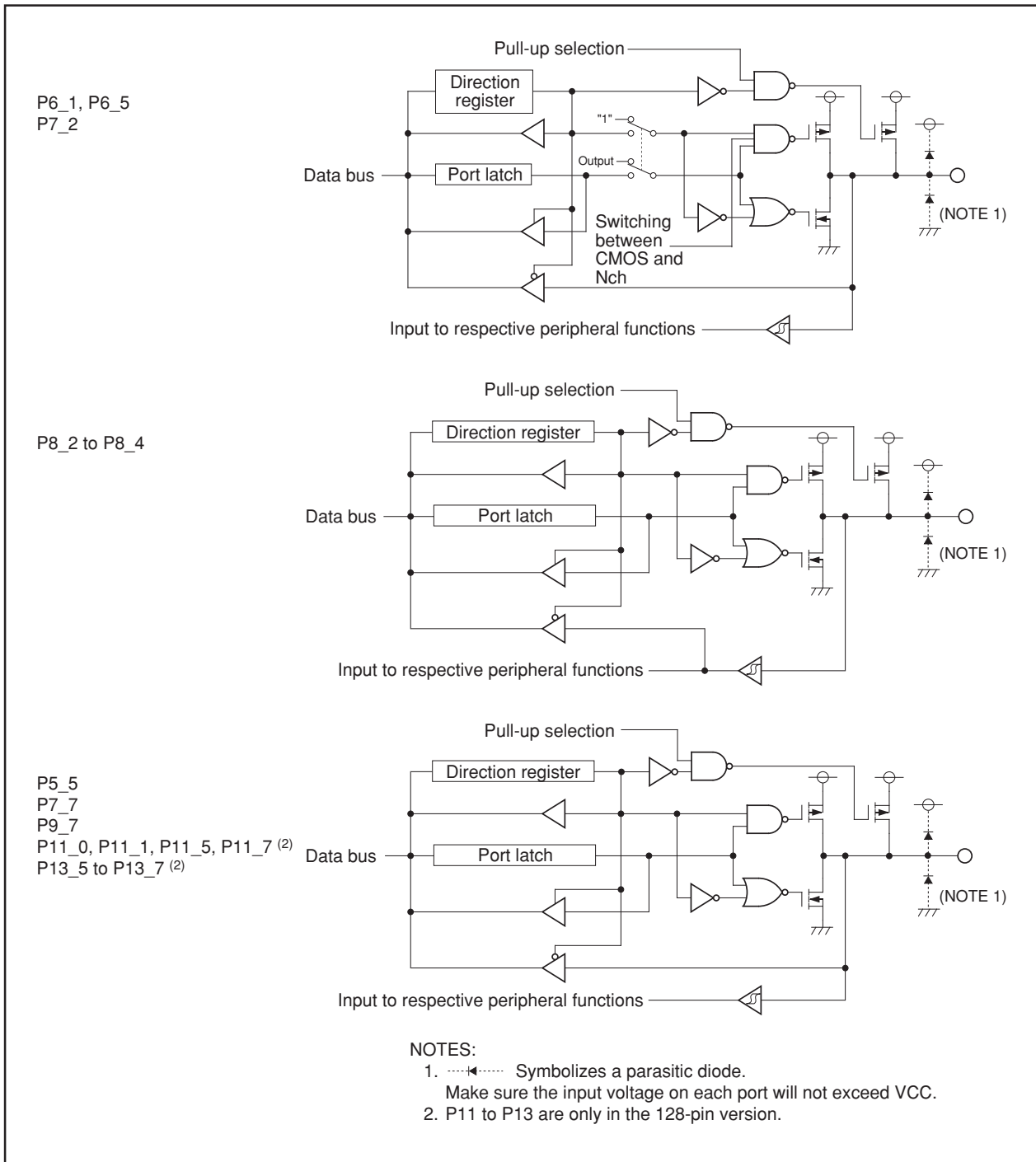


Figure19.2 I/O Ports (2)

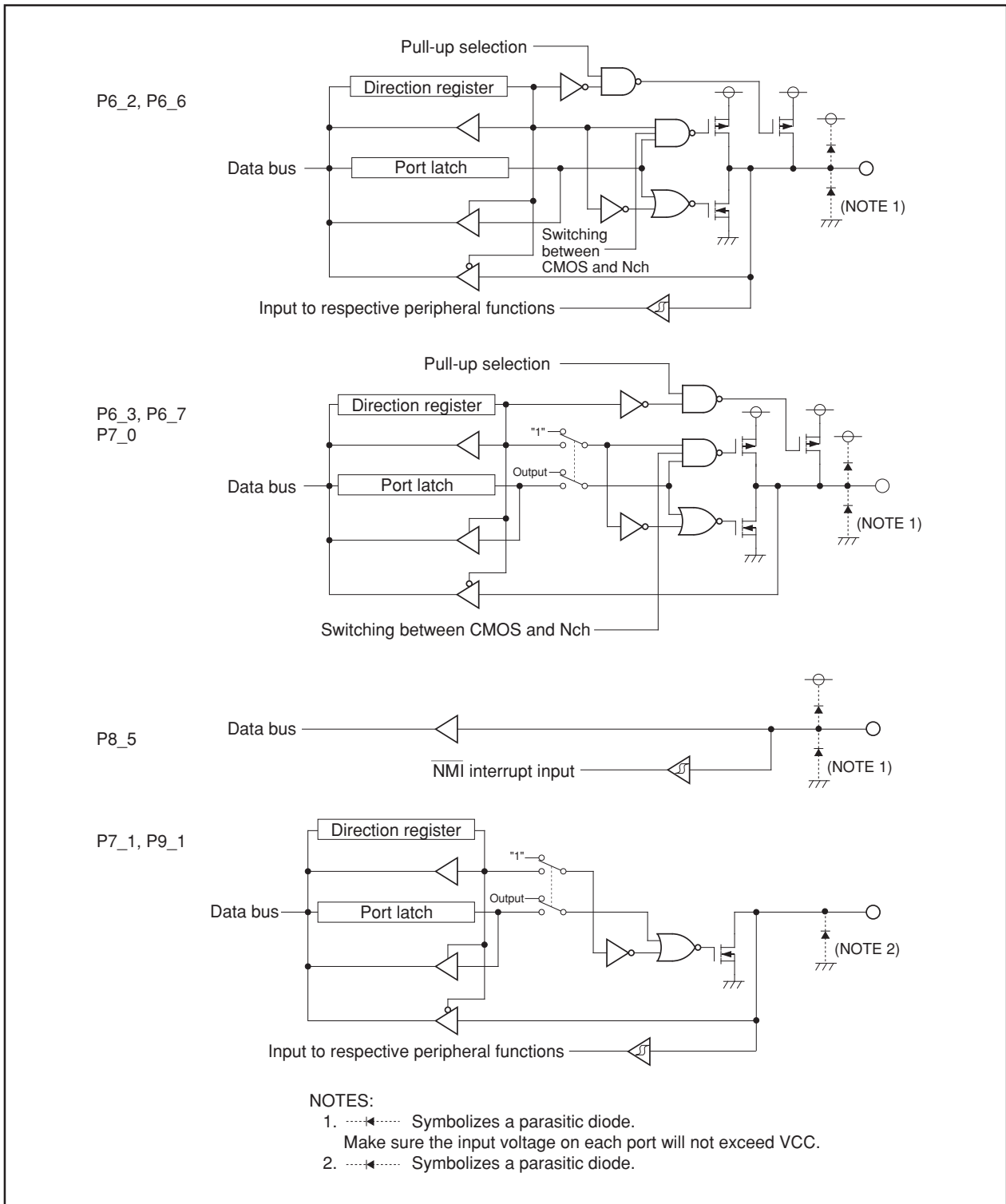


Figure19.3 I/O Ports (3)

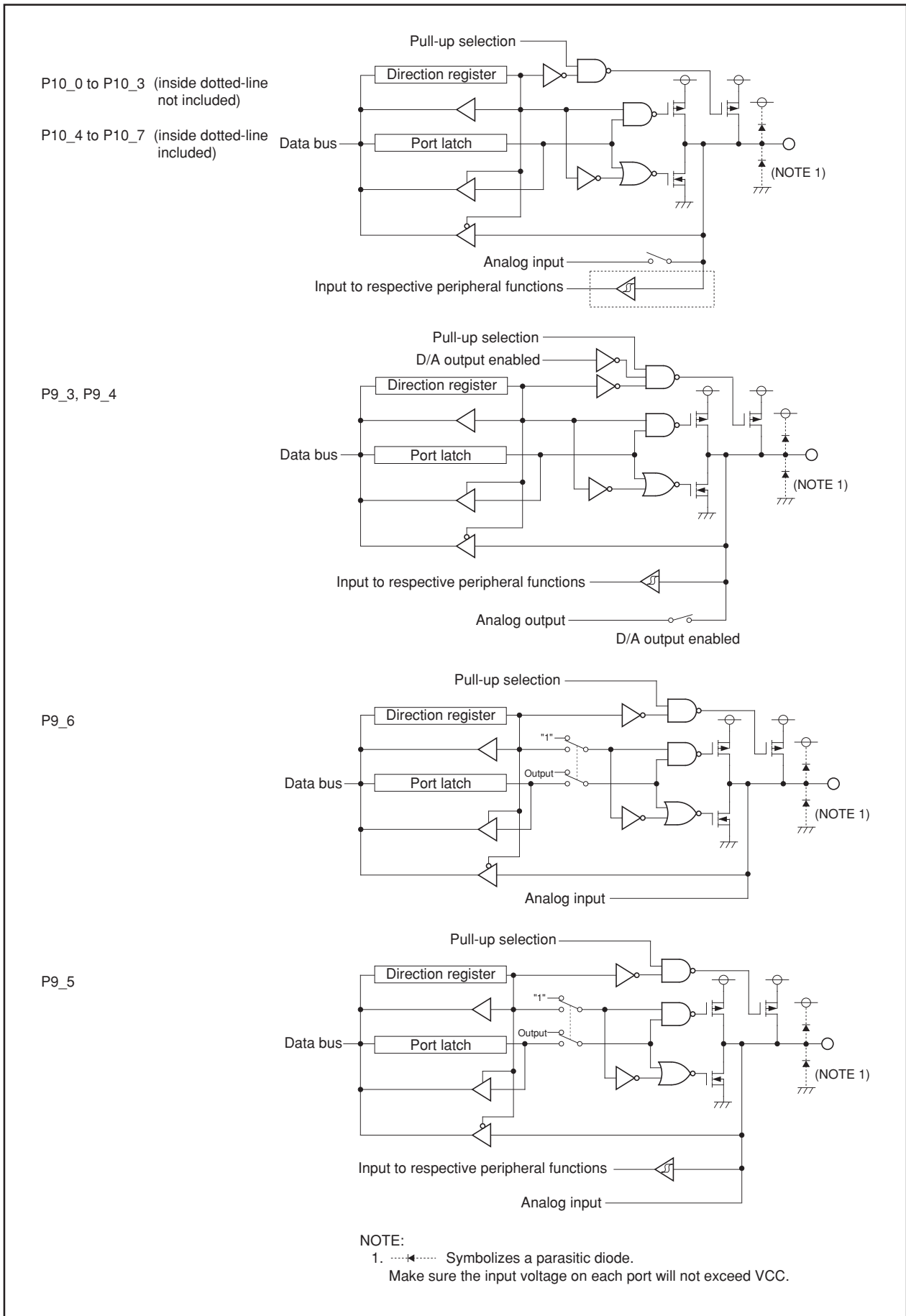


Figure19.4 I/O Ports (4)

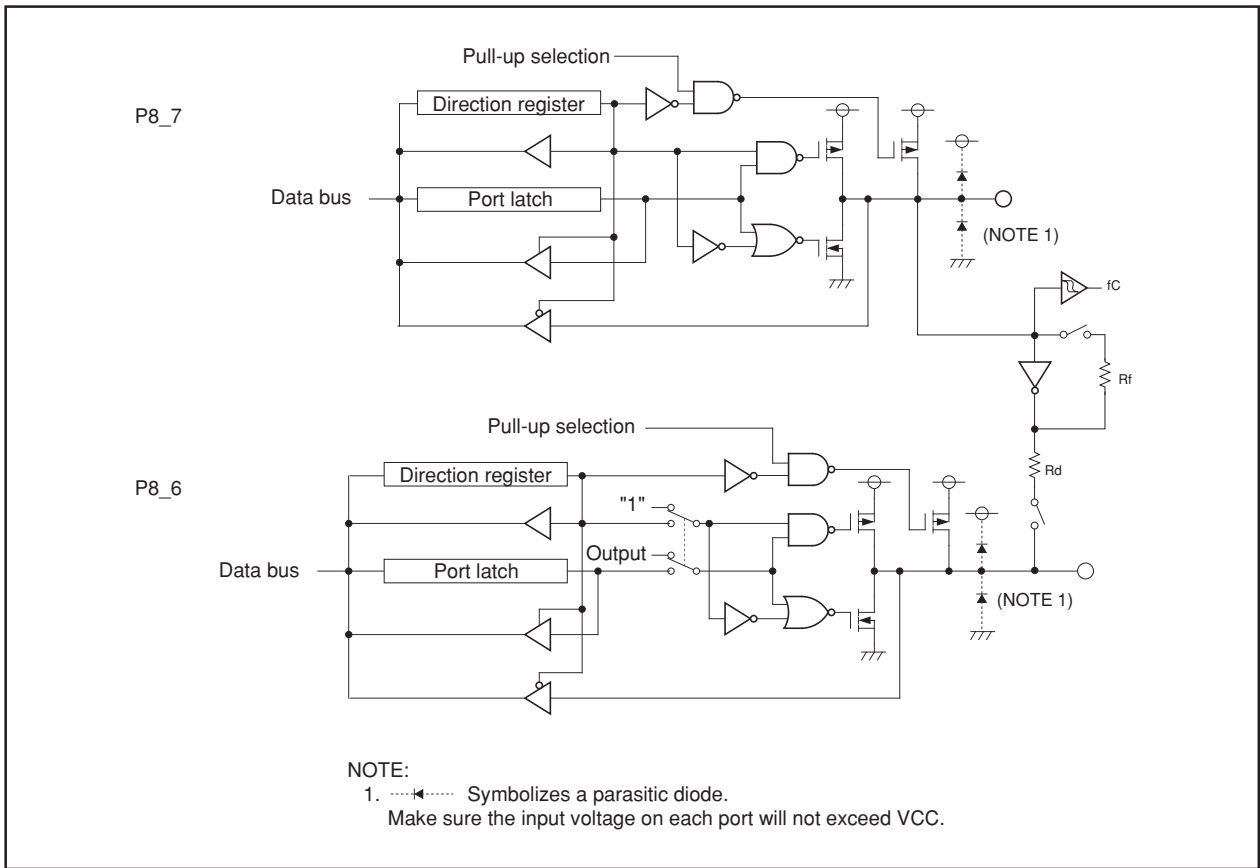


Figure19.5 I/O Ports (5)

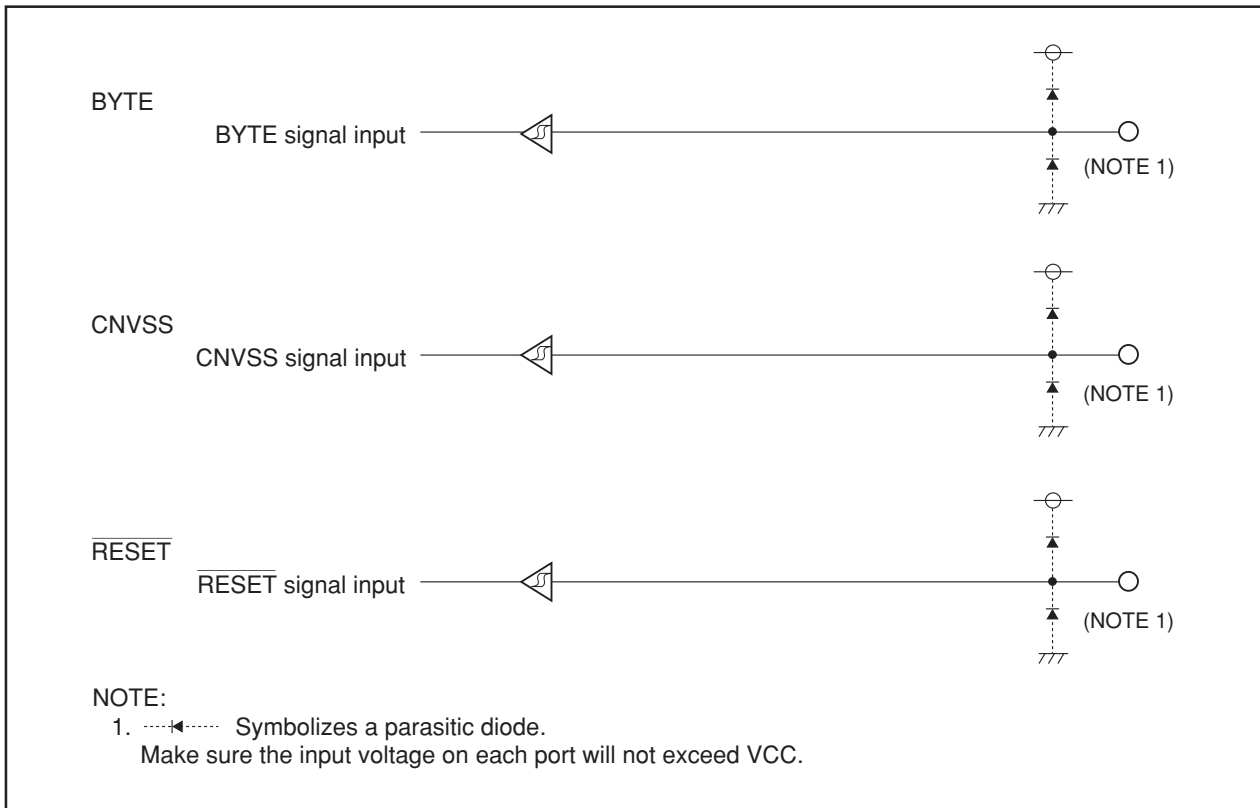


Figure19.6 I/O Pins

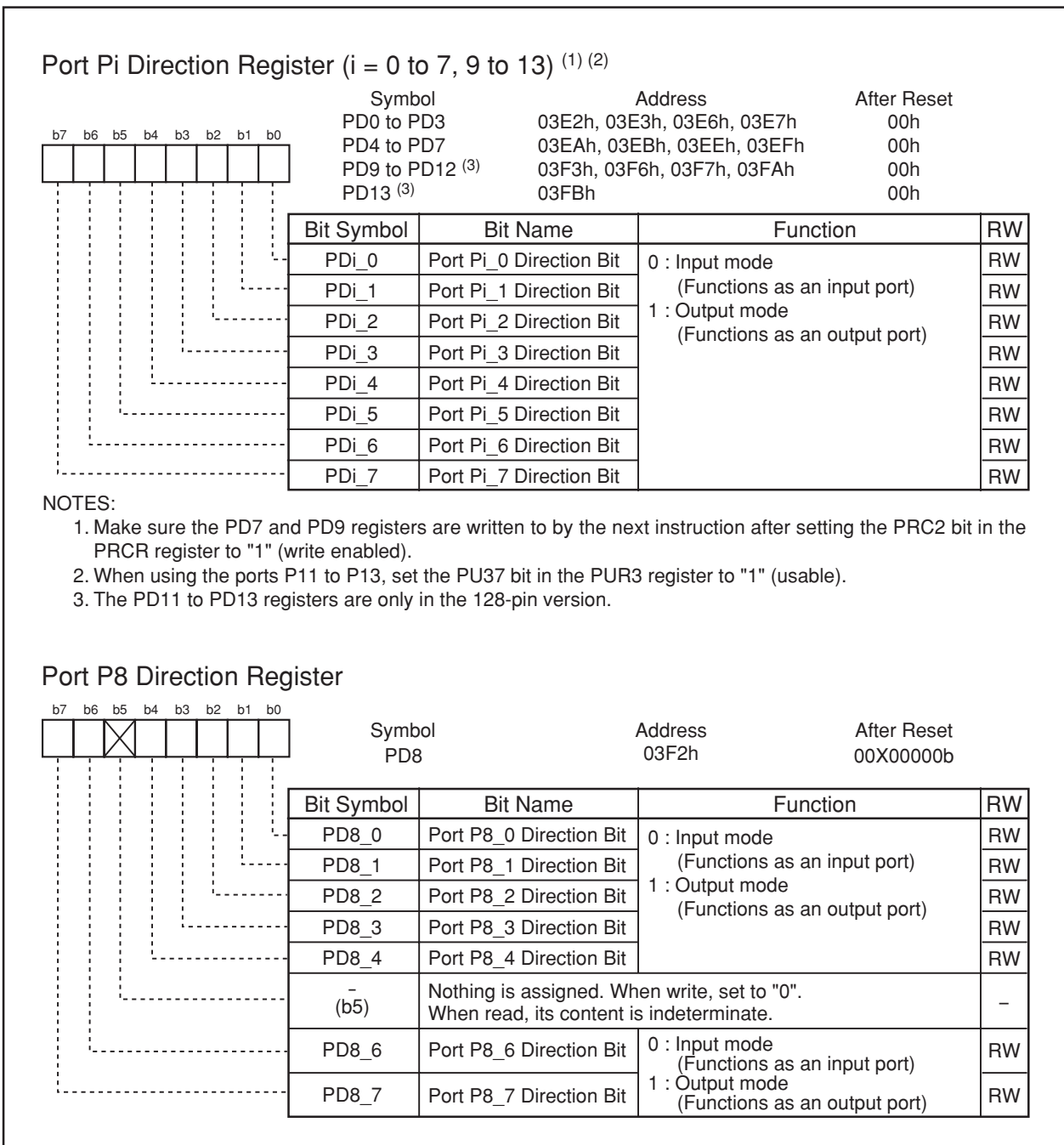


Figure19.7 PD0 to PD13 Registers

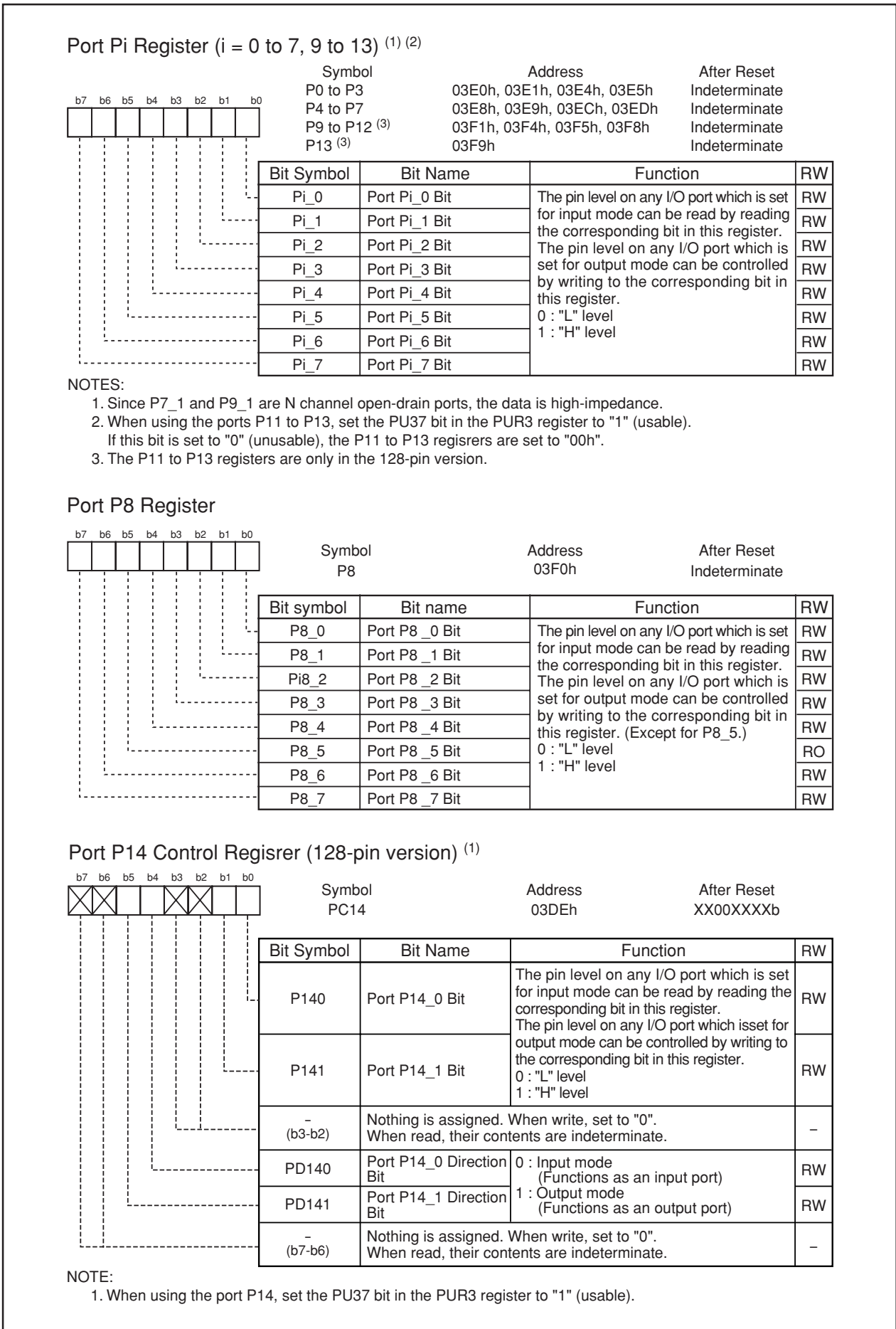


Figure19.8 P0 to P13 Registers and PC14 Register

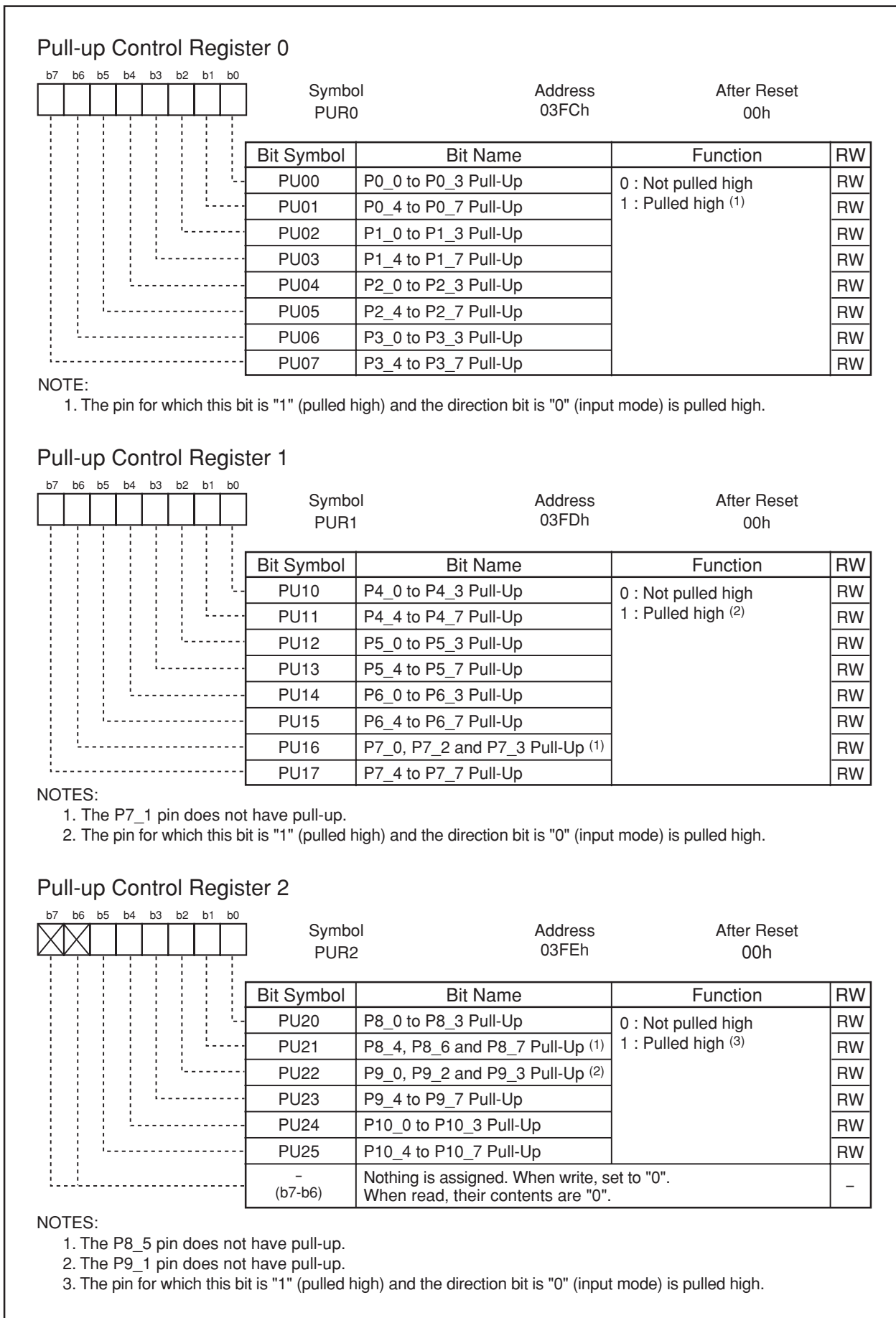


Figure19.9 PUR0, PUR1 and PUR2 Registers

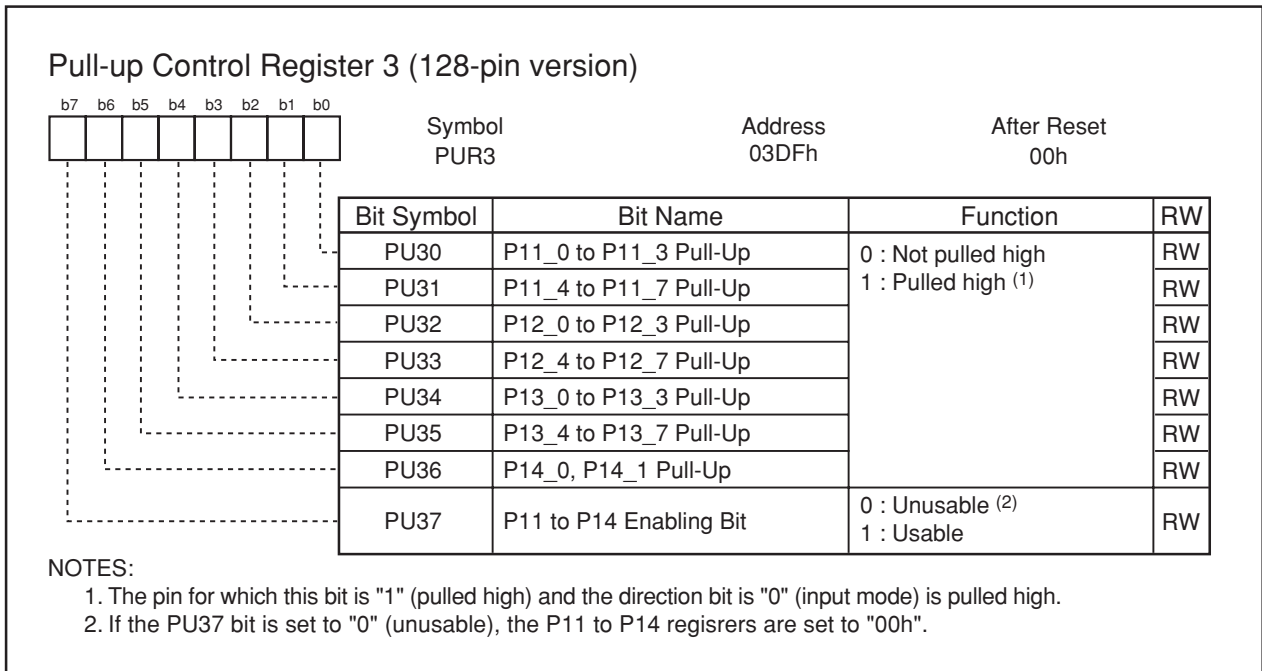


Figure19.10 PUR3 Register

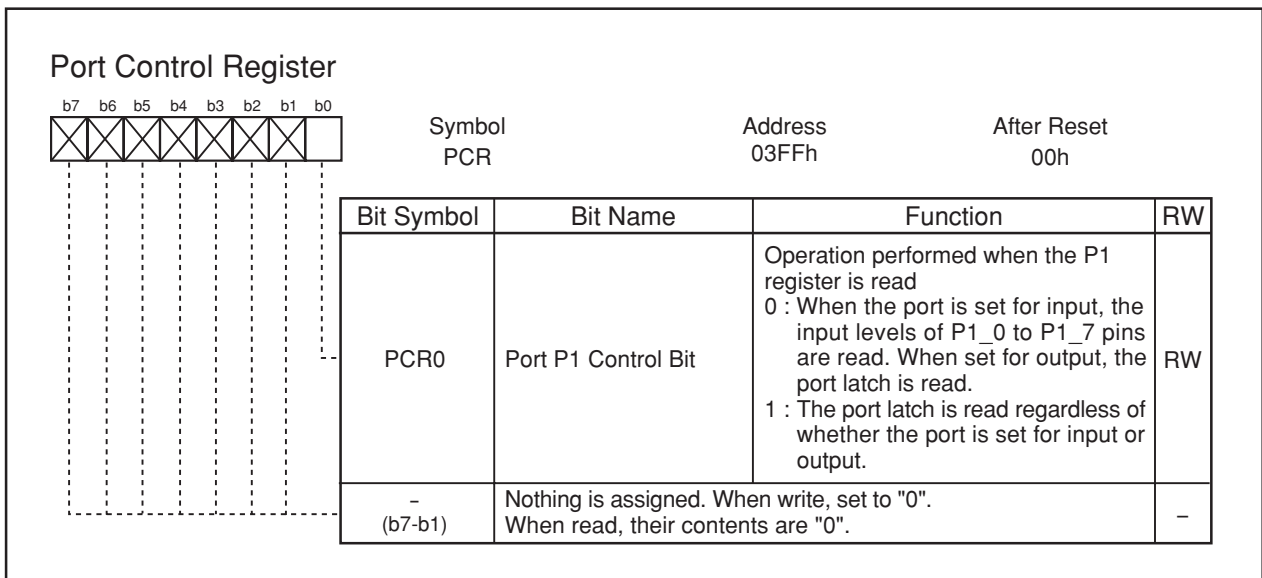


Figure19.11 PCR Register

Table 19.2 Unassigned Pin Handling

Pin Name	Connection
Ports P0 to P7, P8_0 to P8_4, P8_6, P8_7, P9 to P14 ⁽⁵⁾	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open. ^{(1) (2) (3)}
XOUT ⁽⁴⁾	Open
NMI(P8_5)	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF, BYTE	Connect to VSS

NOTES:

1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes 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.
5. The ports P11 to P14 are only in the 128-pin version. When not using all of the P11 to p14 pins may be left open by setting the PU37 bit in the PUR3 register to "0" (P11 to P14 unusable), without causing any problem.

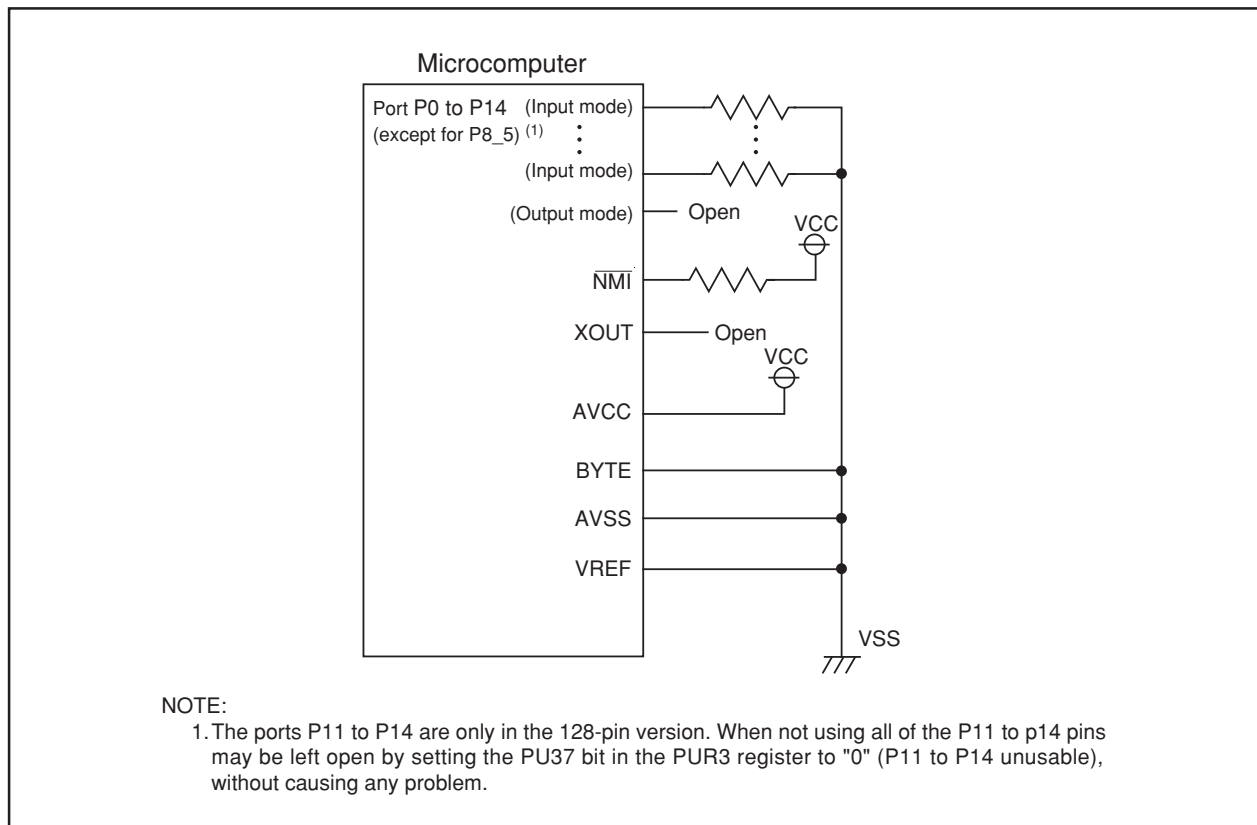


Figure 19.12 Unassigned Pins Handling

20. 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 20.1 lists the specifications of the flash memory version. See **Tables 1.1 and 1.2 Performance outline**, for the items not listed in Table 20.1). Table 20.2 shows the outline of flash memory rewrite mode.

Table 20.1 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
	Boot ROM Area
Program Method	In units of word, in units of byte ⁽²⁾
Erase Method	Collective erase, block erase
Program and Erase Control Method	Program and erase controlled by software command
Protect Method	Lock bit protects each block
Number of Commands	8 commands
Program and Erase Endurance ⁽³⁾	100 times
ROM Code Protection	Parallel I/O, standard serial I/O and CAN I/O modes are supported.

NOTES:

1. The boot ROM area contains 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 guaranteed of 100 times of programming and erasure, each block in it can be erased up to 100 times.

Table 20.2 Flash Memory Rewrite Modes Overview

Flash Memory Rewrite Mode	CPU Rewrite Mode ⁽¹⁾	Standard Serial I/O Mode	Parallel I/O Mode	CAN I/O Mode
Function	The user ROM area is rewritten when the CPU executes software commands. EW0 mode: Rewrite in areas other than flash memory ⁽²⁾ EW1 mode: Can be rewritten in the flash memory	The user ROM area is rewritten using a dedicated serial programmer. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART ⁽³⁾	The boot ROM and user ROM areas are rewritten using a dedicated parallel programmer.	The user ROM area is rewritten using a dedicated CAN programmer.
Areas which can be Rewritten	User ROM area	User ROM area	User ROM area Boot ROM area	User ROM area
Operation Mode	Single-chip 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 area.
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.

20.1 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 a separate 4-Kbyte space as the block A.

Figure 20.1 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).

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 **20.1.1 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 **20.1.1 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.

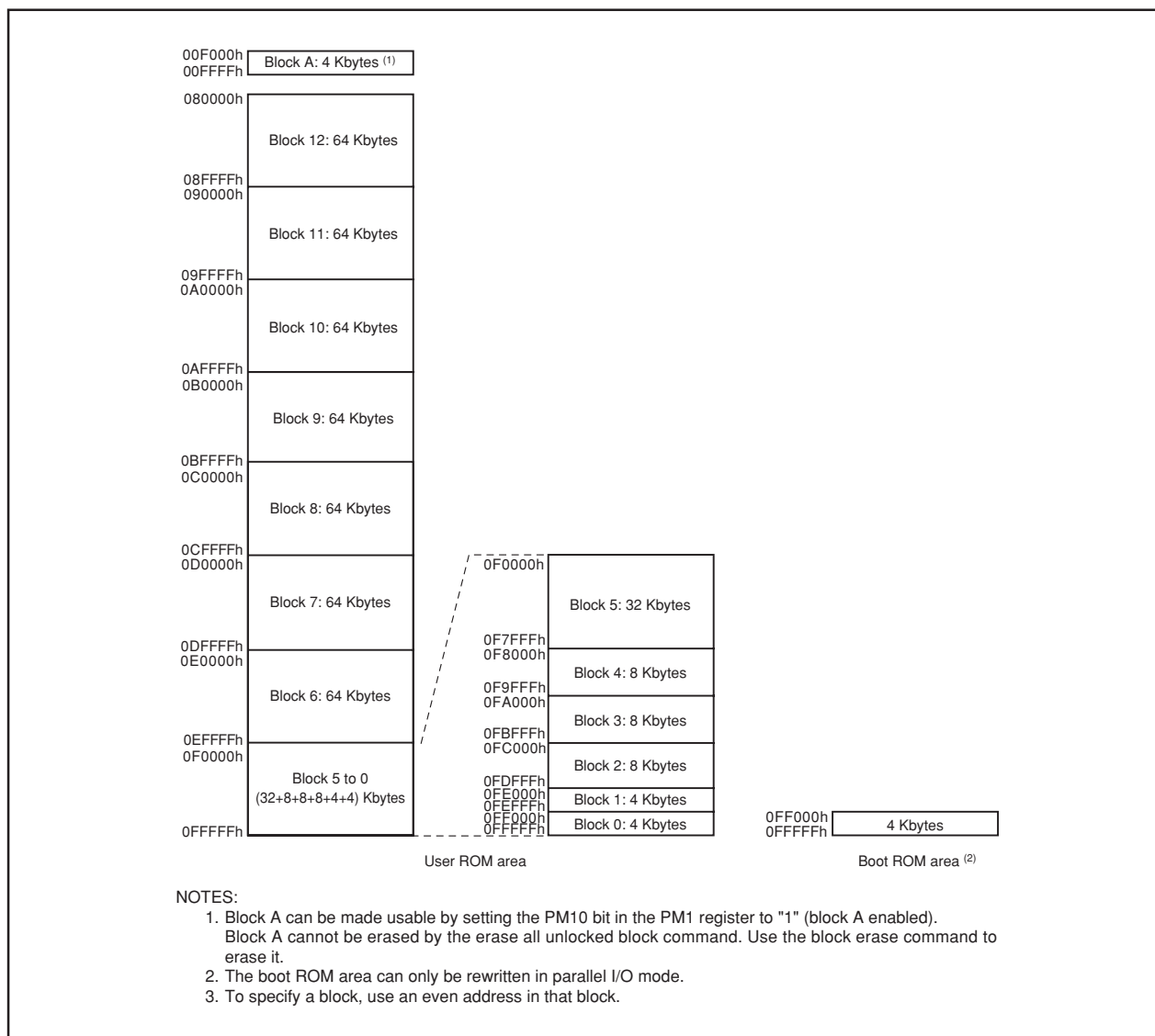


Figure 20.1 Flash Memory Block Diagram

20.1.1 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.

20.2 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.

20.2.1 ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten during parallel I/O mode. Figure 20.2 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.

20.2.2 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 20.3 shows the ID code store addresses.

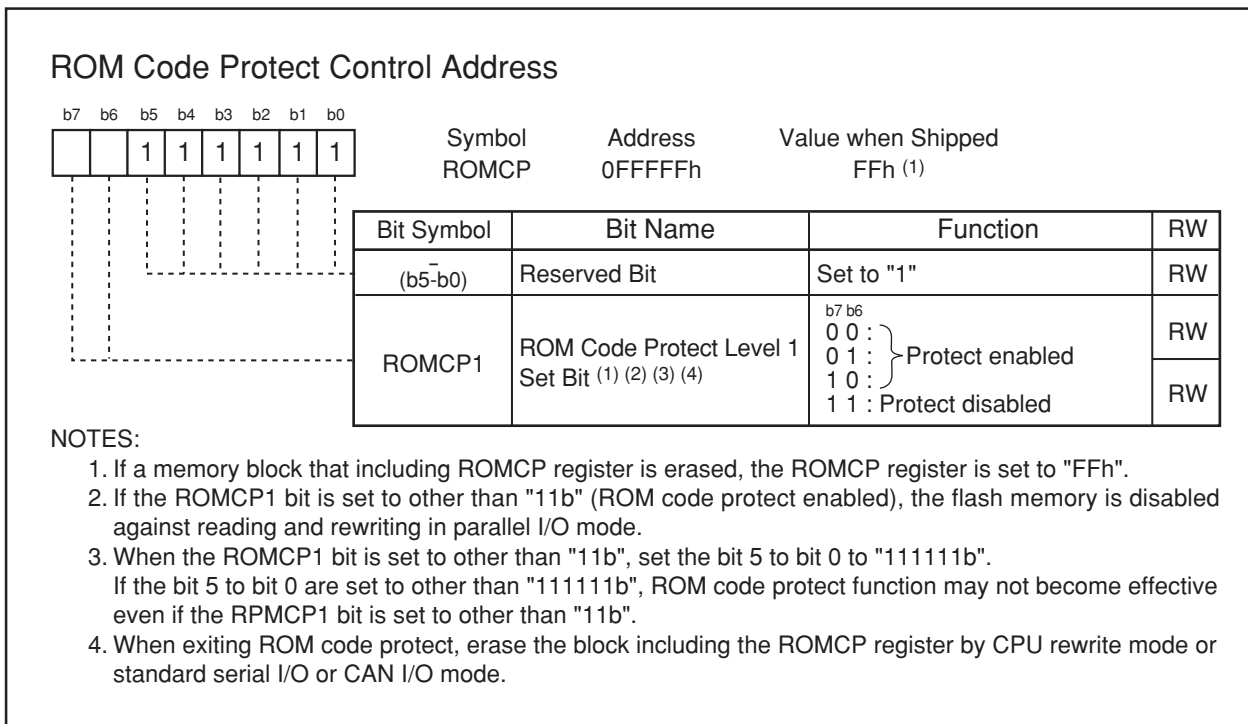


Figure 20.2 ROMCP Register

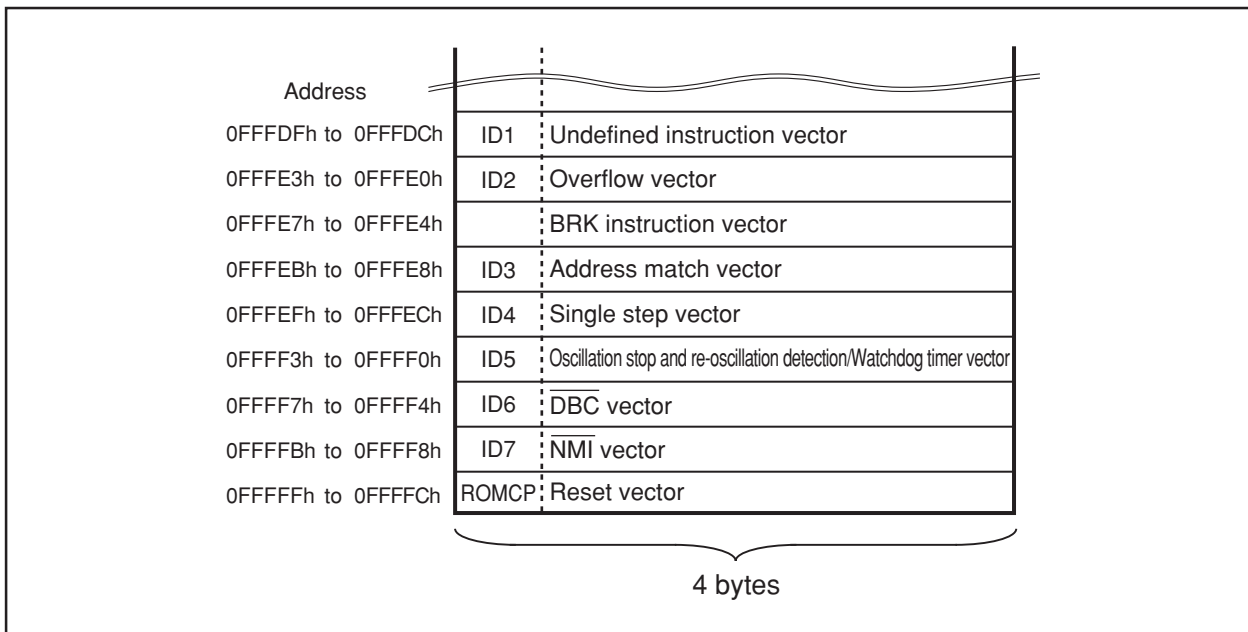


Figure 20.3 Address for ID Code Stored

20.3 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 when 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 20.1 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 20.3 lists the differences between EW0 and EW1 modes.

Table 20.3 EW0 Mode and EW1 Mode

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

NOTES:

1. Do not generate an interrupts (except \overline{NMI} and watchdog timer interrupts) 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 area.

20.3.1 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.

20.3.2 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.

20.3.3 FMR0, FMR1 Registers

Figure 20.4 shows FMR0 and FMR1 registers.

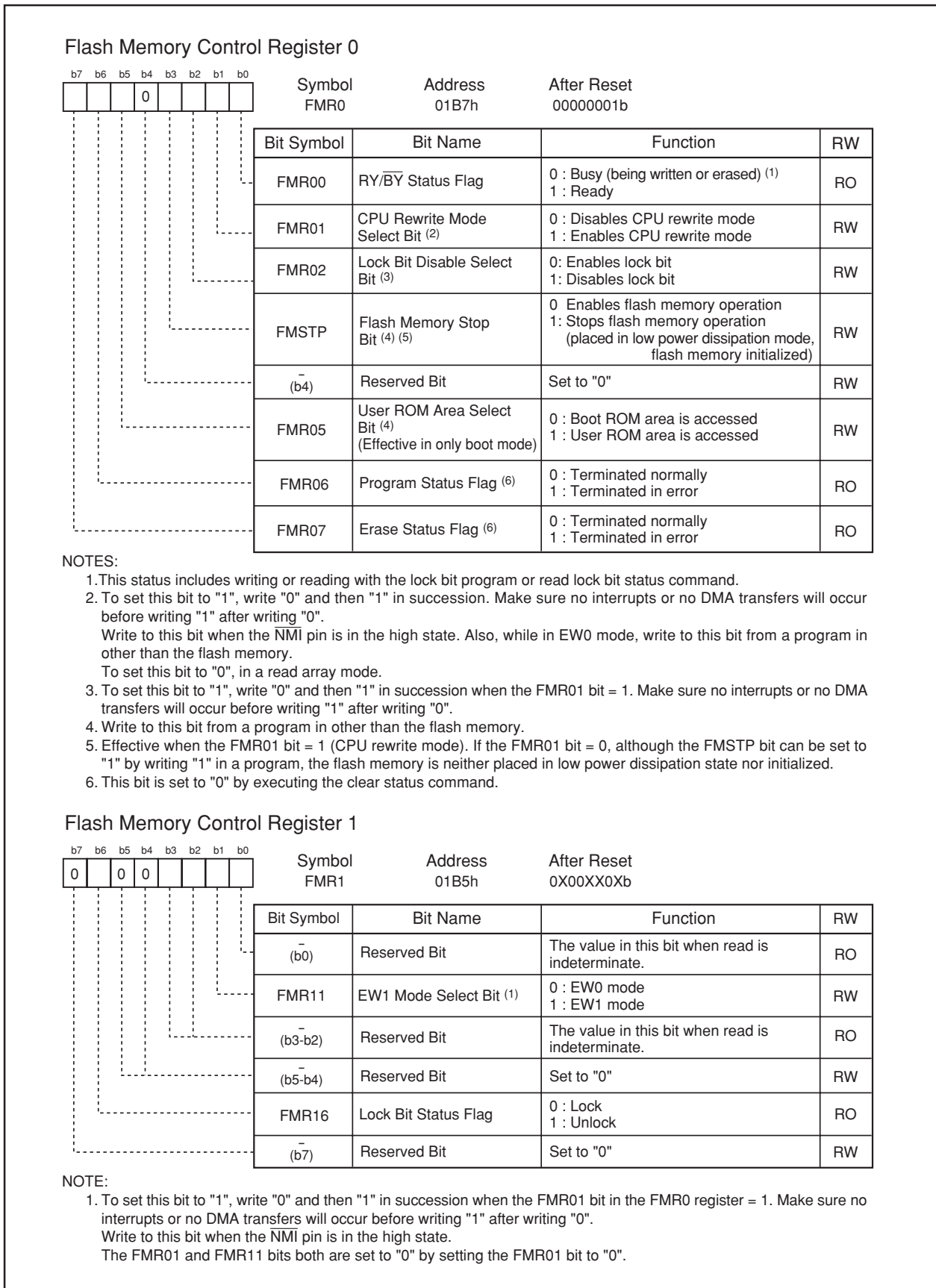


Figure 20.4 FMR0 Register and FMR1 Register

20.3.3.1 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”.

20.3.3.2 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.

20.3.3.3 FMR02 Bit

The lock bit is disabled by setting the FMR02 bit to “1” (lock bit disabled). (Refer to **20.3.6 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.

20.3.3.4 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

Figure 20.7 shows a flow chart illustrating how to start and stop the flash memory before and after entering 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.

20.3.3.5 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.

20.3.3.6 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 **20.3.8 Full Status Check**.

20.3.3.7 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 **20.3.8 Full Status Check**.

20.3.3.8 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).

20.3.3.9 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 20.5 shows how to enter and exit EW0 mode. Figure 20.6 show how to enter and exit EW1 mode.

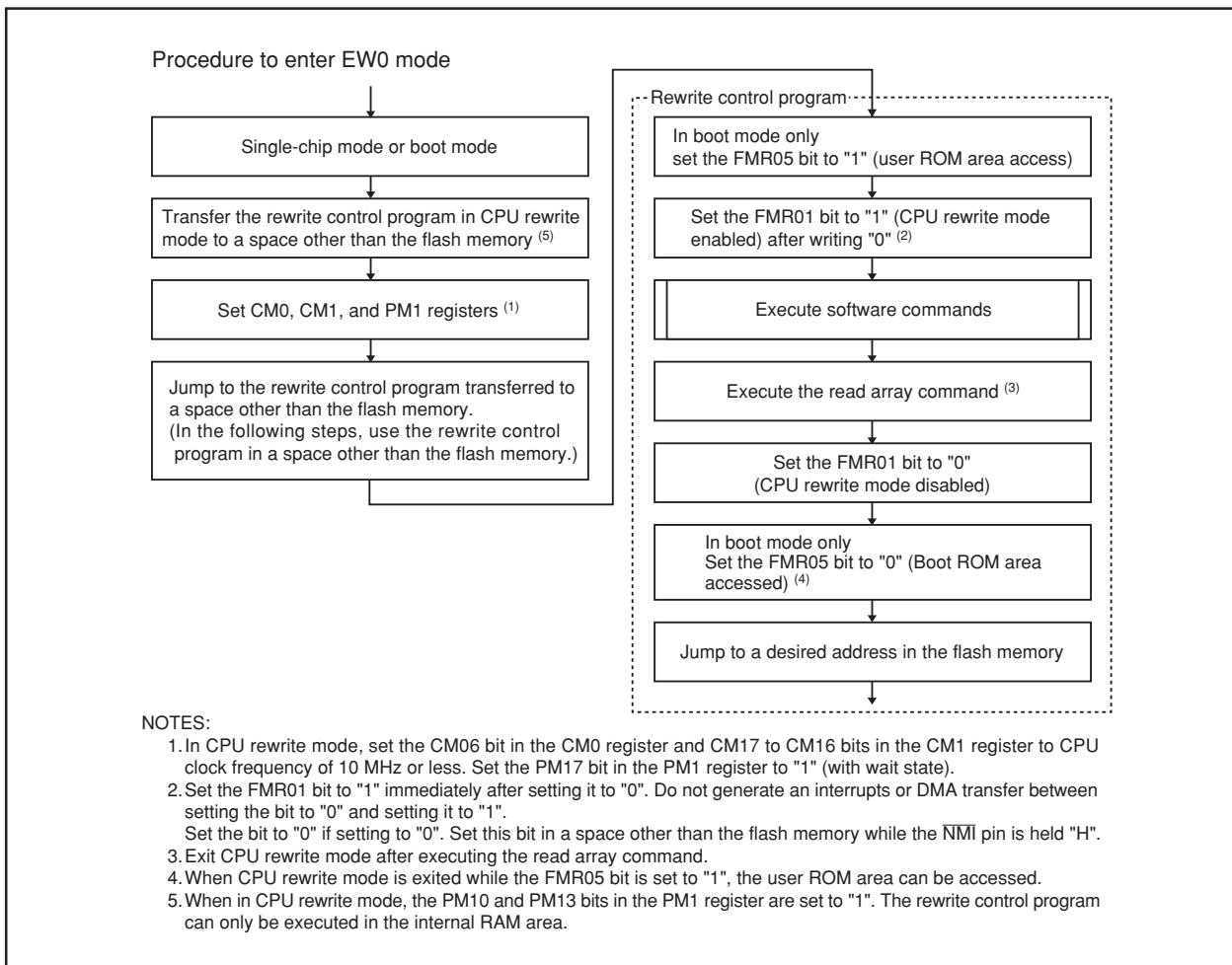


Figure 20.5 Setting and Resetting of EW0 Mode

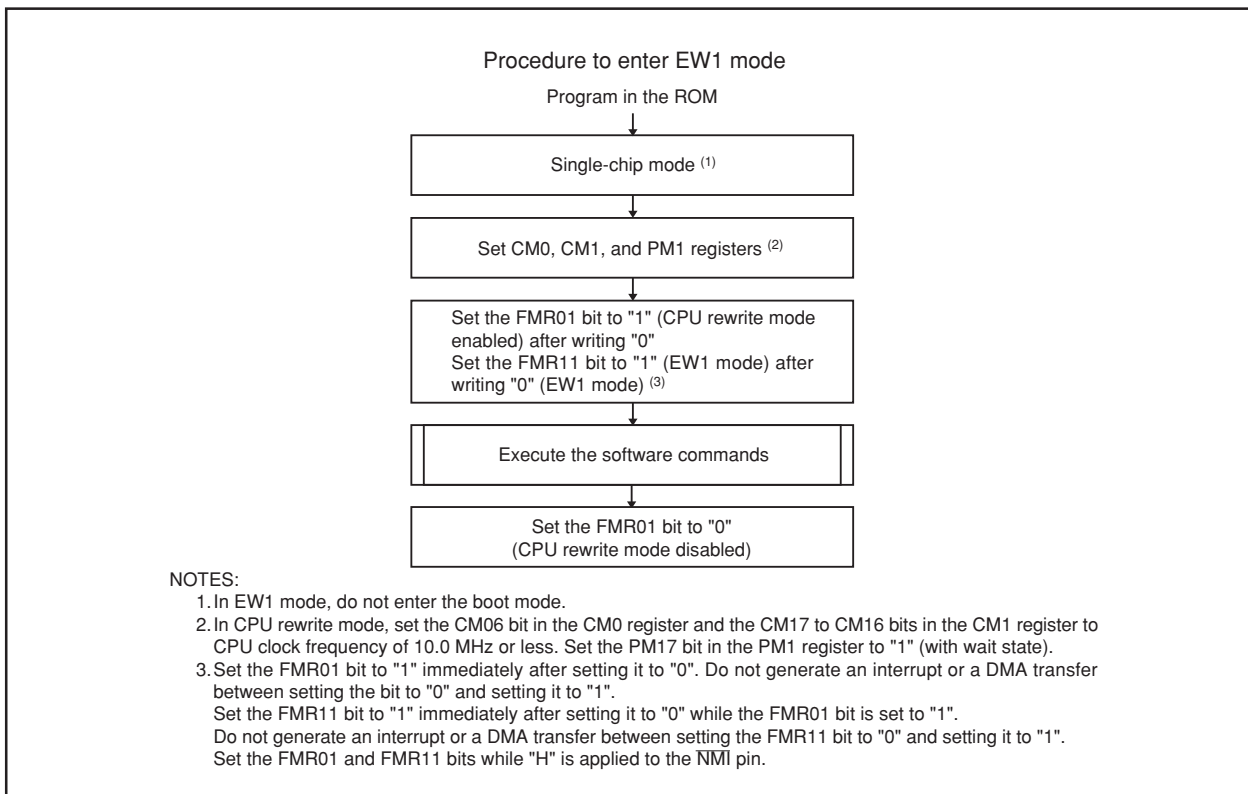


Figure 20.6 Setting and Resetting of EW1 Mode

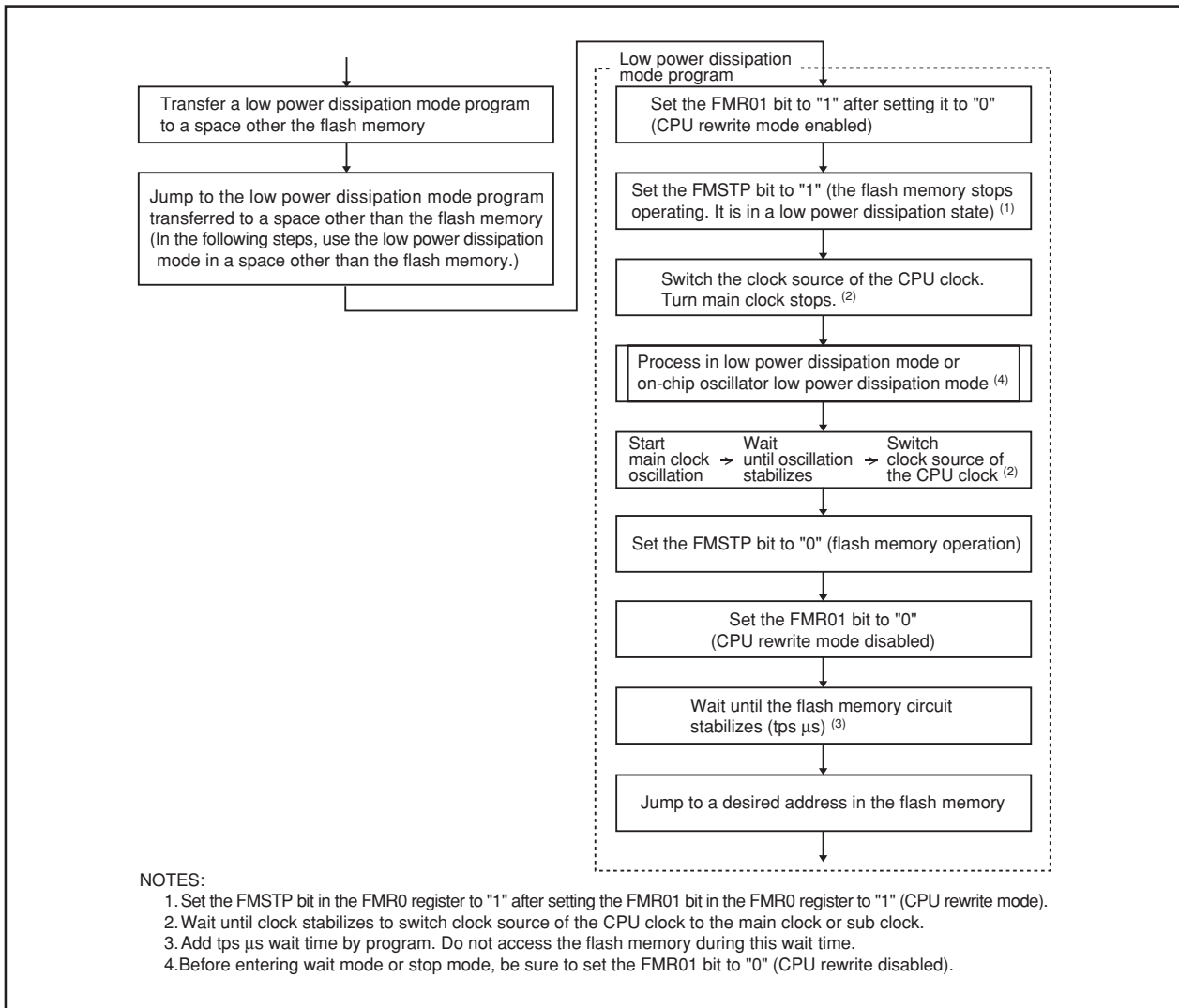


Figure 20.7 Processing Before and After Low Power Dissipation Mode

20.3.4 Precautions on CPU Rewrite Mode

20.3.4.1 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).

20.3.4.2 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

20.3.4.3 Interrupts (EW0 Mode)

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts are available since 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.

20.3.4.4 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 $\overline{\text{NMI}}$ 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 $\overline{\text{NMI}}$ interrupt request is generated. Execute the rewrite program again after exiting the interrupt service routine.

20.3.4.5 How to Access

To set the FMR01, FMR02 or FMR11 bit to "1", write "1" after first setting the bit to "0". Do not generate an interrupt or a DMA transfer between the instruction to set the bit to "0" and the instruction to set the bit to "1". Set the bit while an "H" signal is applied to the $\overline{\text{NMI}}$ pin.

20.3.4.6 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.

20.3.4.7 Rewriting in User ROM Area (EW1 Mode)

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

20.3.4.8 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).

20.3.4.9 Writing Command and Data

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

20.3.4.10 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.

20.3.4.11 Stop Mode

When entering stop mode, the following settings are required:

- Set the FMR01 bit to “0” (CPU rewrite mode disabled). Disable DMA transfer before setting the CM10 bit to “1” (stop mode).
- Execute the instruction to set the CM10 bit to “1” (stop mode) and then the JMP.B instruction.

Example program BSET 0, CM1 ; Stop mode

 JMP.B L1

 L1:

 Program after exiting stop mode

20.3.4.12 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 software command
- Read lock bit status

20.3.5 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 20.4 lists the software commands.

Table 20.4 Software Commands

Software Command	First Bus Cycle			Second Bus Cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read Array	Write	X	xxFFh	-	-	-
Read Status Register	Write	X	xx70h	Read	X	SRD
Clear Status Register	Write	X	xx50h	-	-	-
Program	Write	WA	xx40h	Write	WA	WD
Block Erase	Write	X	xx20h	Write	BA	xxD0h
Erase All Unlocked Block ⁽¹⁾	Write	X	xxA7h	Write	X	xxD0h
Lock Bit Program	Write	BA	xx77h	Write	BA	xxD0h
Read Lock Bit Status	Write	X	xx71h	Write	BA	xxD0h

SRD: data in SRD register (D7 to D0)

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

WD: 16-bit write data

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

X: 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 12 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.

20.3.5.1 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.

20.3.5.2 Read Status Register Command (70h)

The read status register command reads the status register (refer to **20.3.7 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.

20.3.5.3 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”.

20.3.5.4 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 **20.3.8 Full Status Check.**)

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

Figure 20.8 shows a flow chart of the program command programming.

The lock bit protects each block from being programmed inadvertently. (Refer to **20.3.6 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.

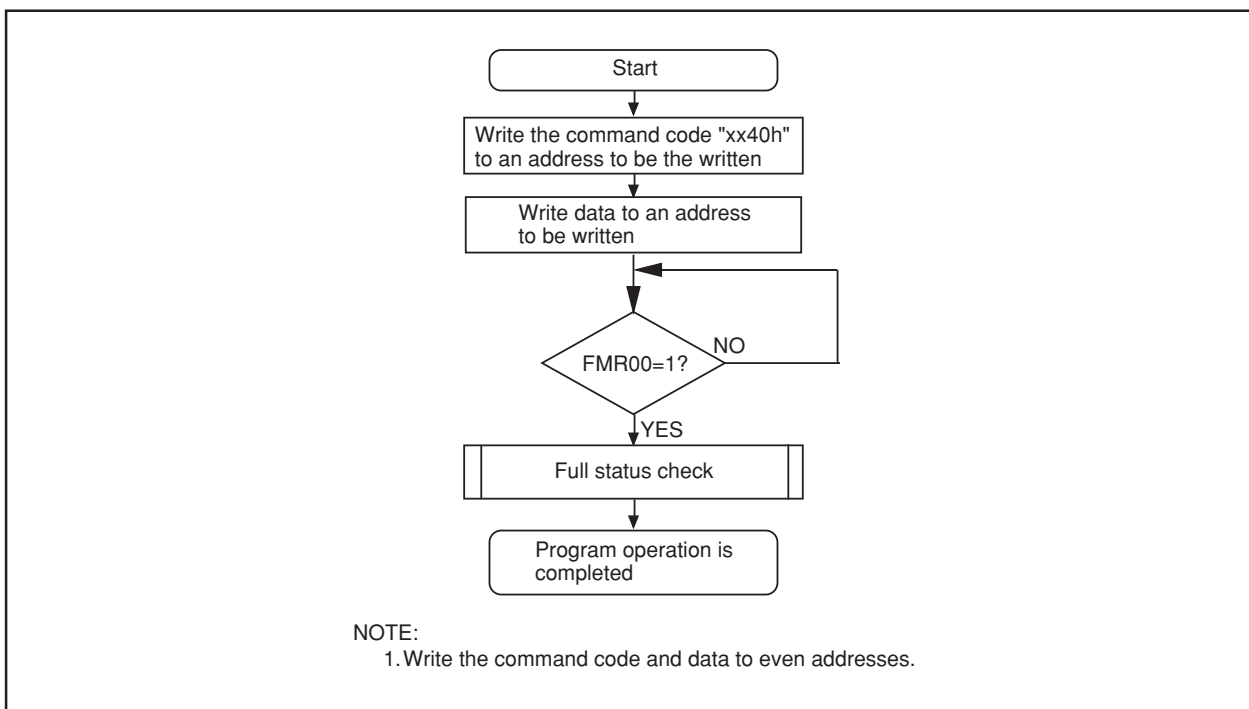


Figure 20.8 Program Command

20.3.5.5 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 **20.3.8 Full Status Check.**)

Figure 20.9 shows a flow chart of the block erase command programming.

The lock bit protects each block from being programmed inadvertently. (Refer to **20.3.6 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.

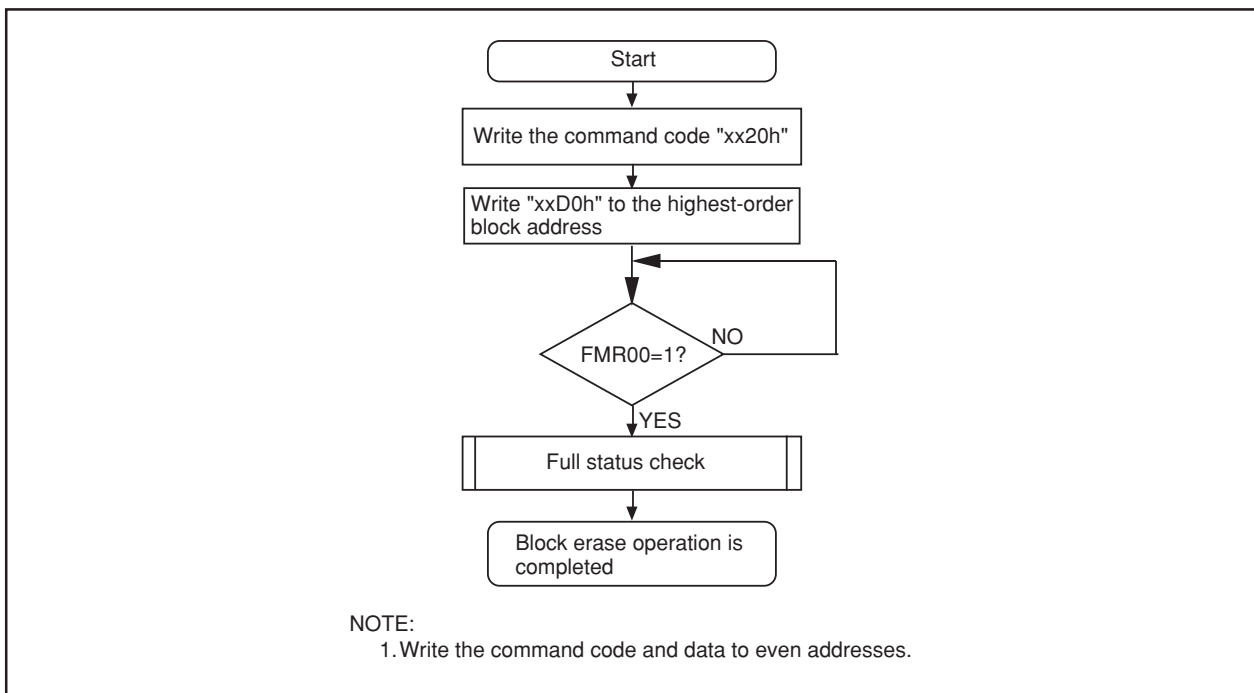


Figure 20.9 Block Erase Command

20.3.5.6 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 **20.3.6 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 12 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.

20.3.5.7 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 20.10 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 **20.3.6 Data Protect Function** for details on lock bit functions and how to set it to "1" (unlocked).

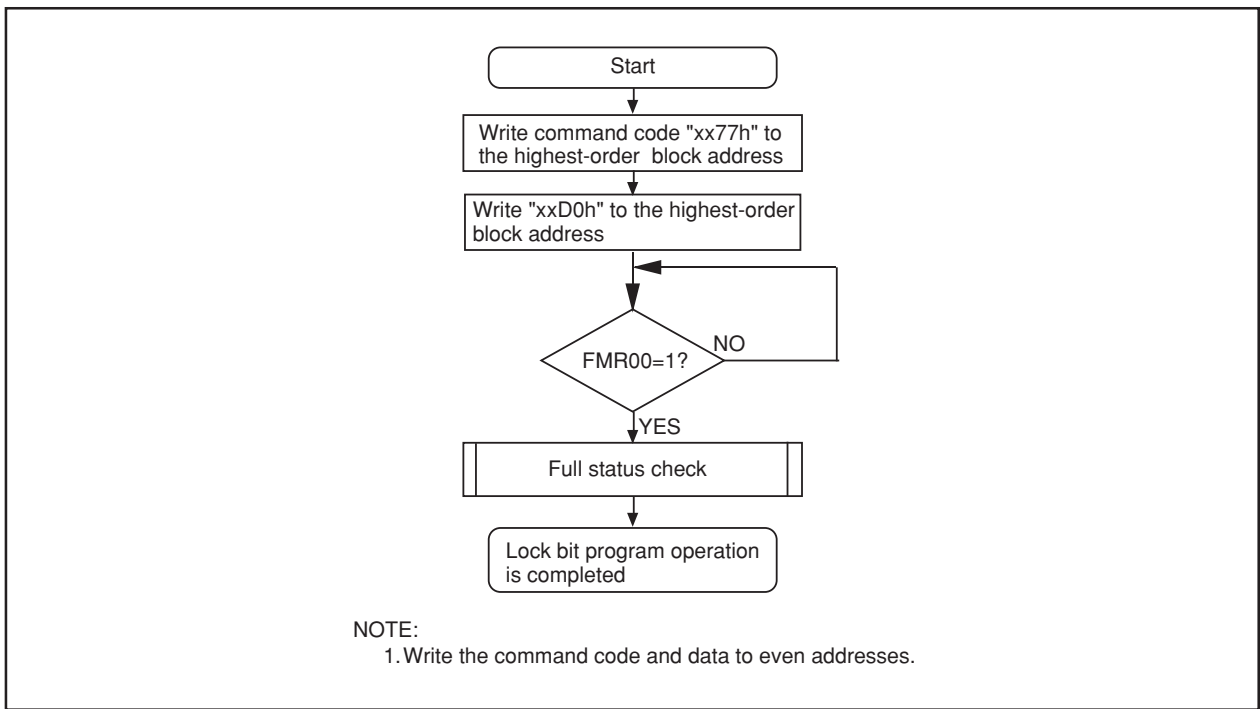


Figure 20.10 Lock Bit Program Command

20.3.5.8 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 20.11 shows a flow chart of the read lock bit status command programming.

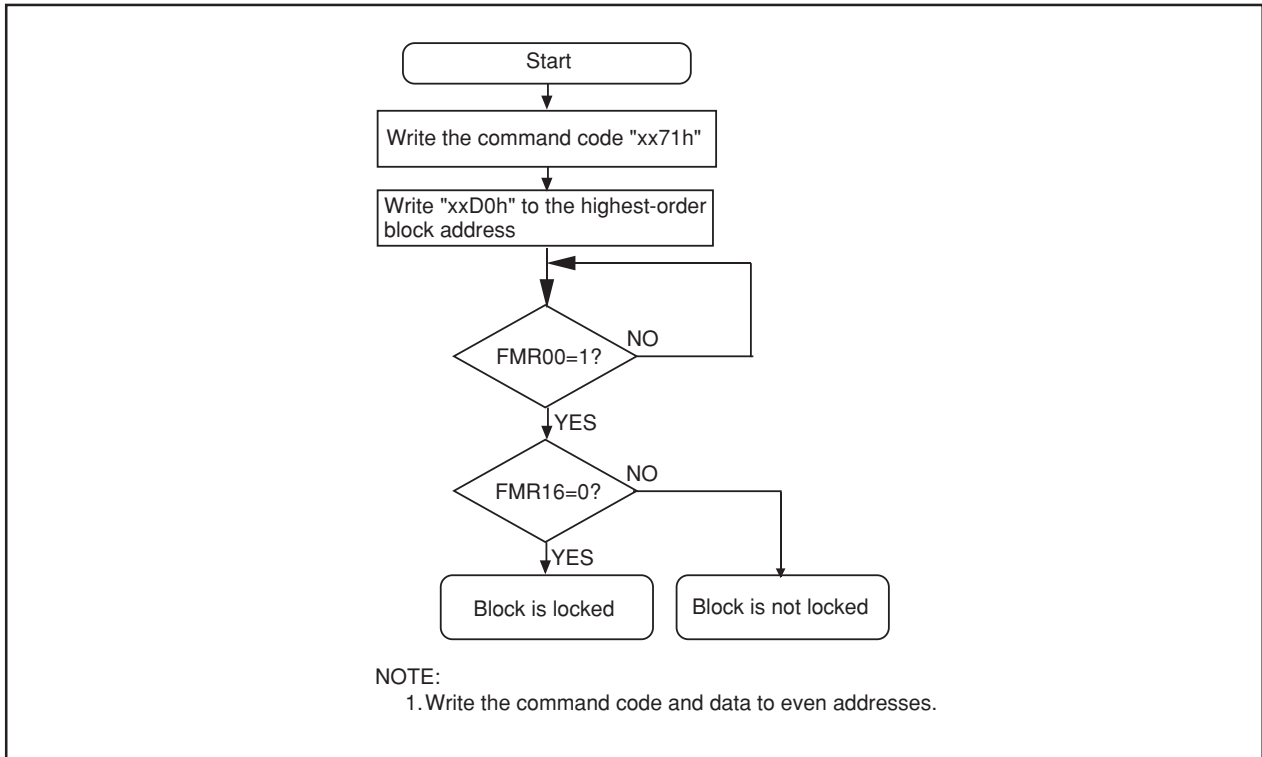


Figure 20.11 Read Lock Bit Status Command

20.3.6 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 **20.3.5 Software Commands** for details on each command.

20.3.7 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 20.5 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.

20.3.7.1 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”.

20.3.7.2 Erase Status (SR5 and FMR07 Bits)

Refer to **20.3.8 Full Status Check**.

20.3.7.3 Program Status (SR4 and FMR06 Bits)

Refer to **20.3.8 Full Status Check**.

Table 20.5 Status Register

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

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

NOTE:

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

20.3.8 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 20.6 lists errors and FMR0 register state. Figure 20.12 shows a flow chart of the full status check and handling procedure for each error.

Table 20.6 Errors and FMR0 Register Status

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

NOTES:

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

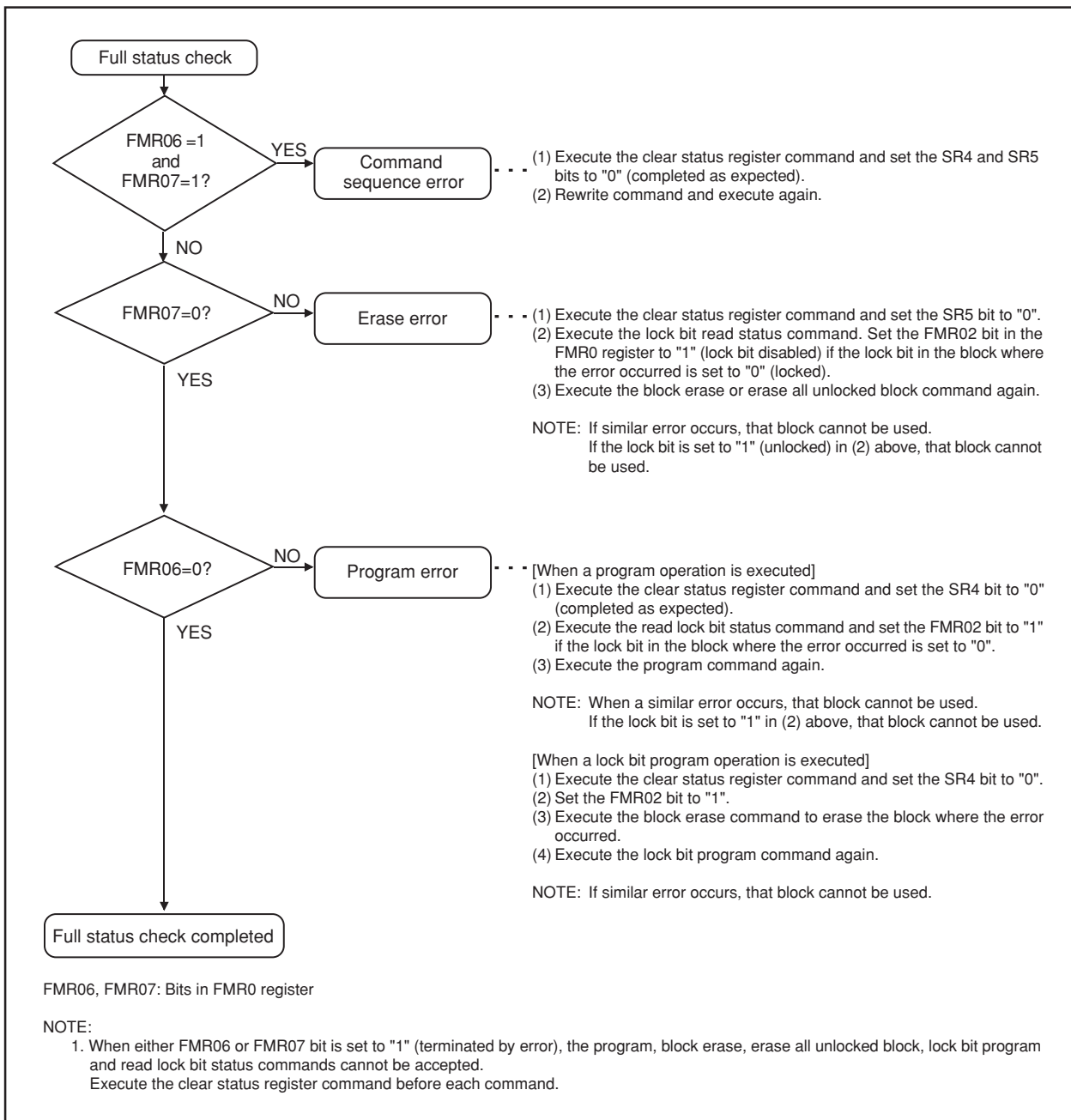


Figure 20.12 Full Status Check and Handling Procedure for Each Error

20.4 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the M16C/6N Group (M16C/6NK, M16C/6NM) 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 20.7 lists pin functions for standard serial I/O mode. Figures 20.13 and 20.14 show pin connections for standard serial I/O mode.

20.4.1 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 **20.2 Functions to Prevent Flash Memory from Rewriting.**)

Table 20.7 Pin Functions for Standard Serial I/O Mode

Pin	Name	I/O	Description
VCC1, VCC2, VSS	Power supply input		Apply the voltage guaranteed for Program and Erase to VCC1 pin and VCC2 to VCC2 pin. The VCC apply condition is that VCC2 = VCC1. Apply 0 V to VSS pin.
CNVSS	CNVSS	I	Connect to VCC1 pin.
RESET	Reset input	I	Reset input pin. While RESET pin is "L" level, input 20 cycles or longer clock to XIN pin.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
BYTE	BYTE	I	Connect this pin to VCC1 or VSS.
AVCC, AVSS	Analog power supply input		Connect AVCC to VCC1 and AVSS to VSS, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for A/D and D/A converters from this pin.
P0_0 to P0_7	Input port P0	I	Input "H" or "L" level signal or open.
P1_0 to P1_7	Input port P1	I	Input "H" or "L" level signal or open.
P2_0 to P2_7	Input port P2	I	Input "H" or "L" level signal or open.
P3_0 to P3_7	Input port P3	I	Input "H" or "L" level signal or open.
P4_0 to P4_7	Input port P4	I	Input "H" or "L" level signal or open.
P5_0	CE input	I	Input "H" level signal.
P5_1 to P5_4, P5_6, P5_7	Input port P5	I	Input "H" or "L" level signal or open.
P5_5	EPM input	I	Input "L" level signal.
P6_0 to P6_3	Input port P6	I	Input "H" or "L" level signal or open.
P6_4/RTS1	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitors the boot program operation check signal output pin.
P6_5/CLK1	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin. Standard serial I/O mode 2: Input "L".
P6_6/RXD1	RXD input	I	Serial data input pin
P6_7/TXD1	TXD output	O	Serial data output pin ⁽¹⁾
P7_0 to P7_7	Input port P7	I	Input "H" or "L" level signal or open.
P8_0 to P8_4, P8_6, P8_7	Input port P8	I	Input "H" or "L" level signal or open.
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.
P11_0 to P11_7 ⁽²⁾	Input port P11	I	Input "H" or "L" level signal or open.
P12_0 to P12_7 ⁽²⁾	Input port P12	I	Input "H" or "L" level signal or open.
P13_0 to P13_7 ⁽²⁾	Input port P13	I	Input "H" or "L" level signal or open.
P14_0, P14_1 ⁽²⁾	Input port P14	I	Input "H" or "L" level signal or open.

NOTES:

1. When using standard serial I/O mode 1, the TXD pin must be held high while the RESET pin is pulled low. Therefore, connect this pin to VCC1 via a resistor. Because this pin is directed for data output after reset, adjust the pull-up resistance value in the system so that data transfers will not be affected.
2. The pins P11 to P14 are only in the 128-pin version.

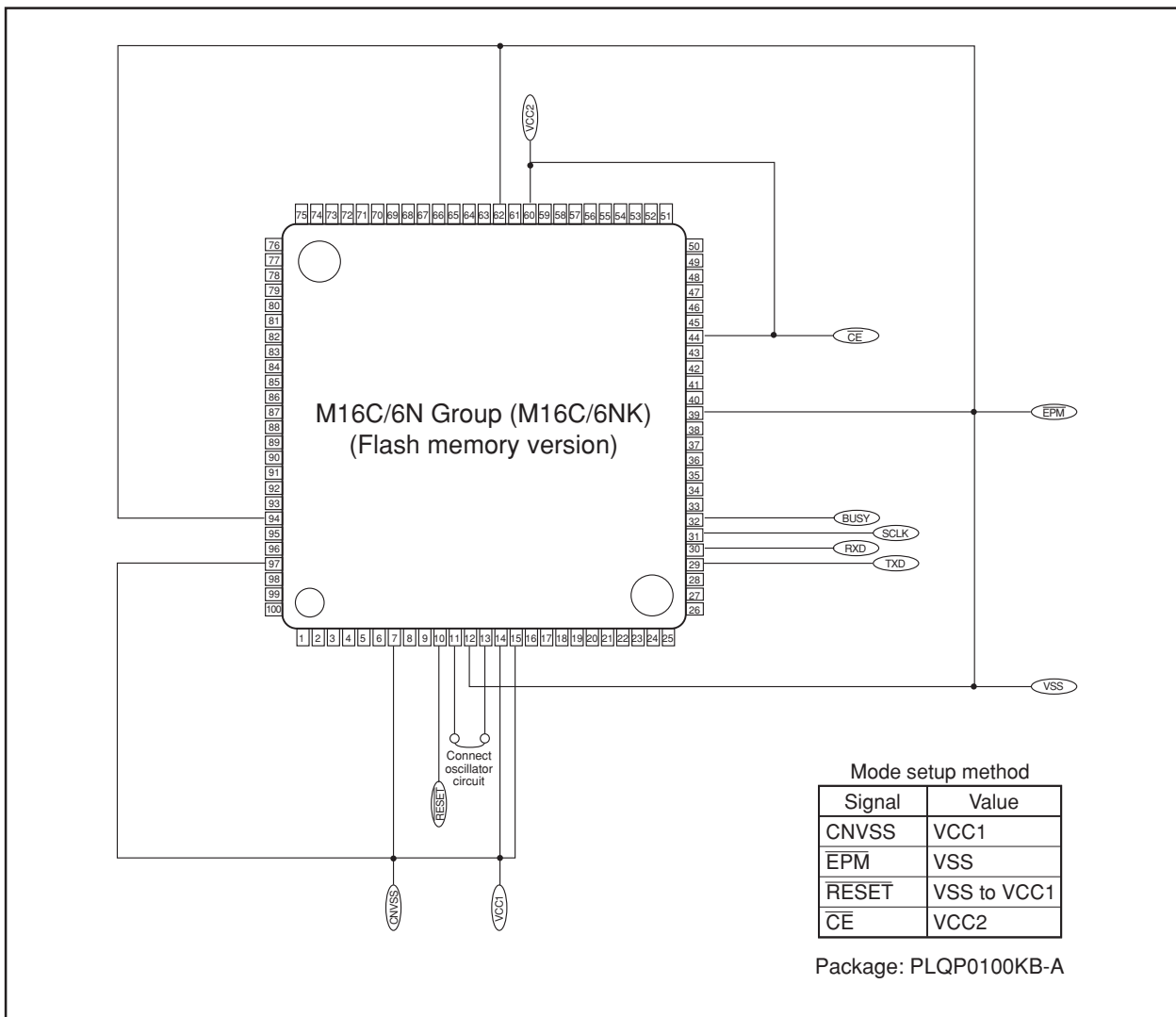


Figure 20.13 Pin Connections for Standard Serial I/O Mode (1)

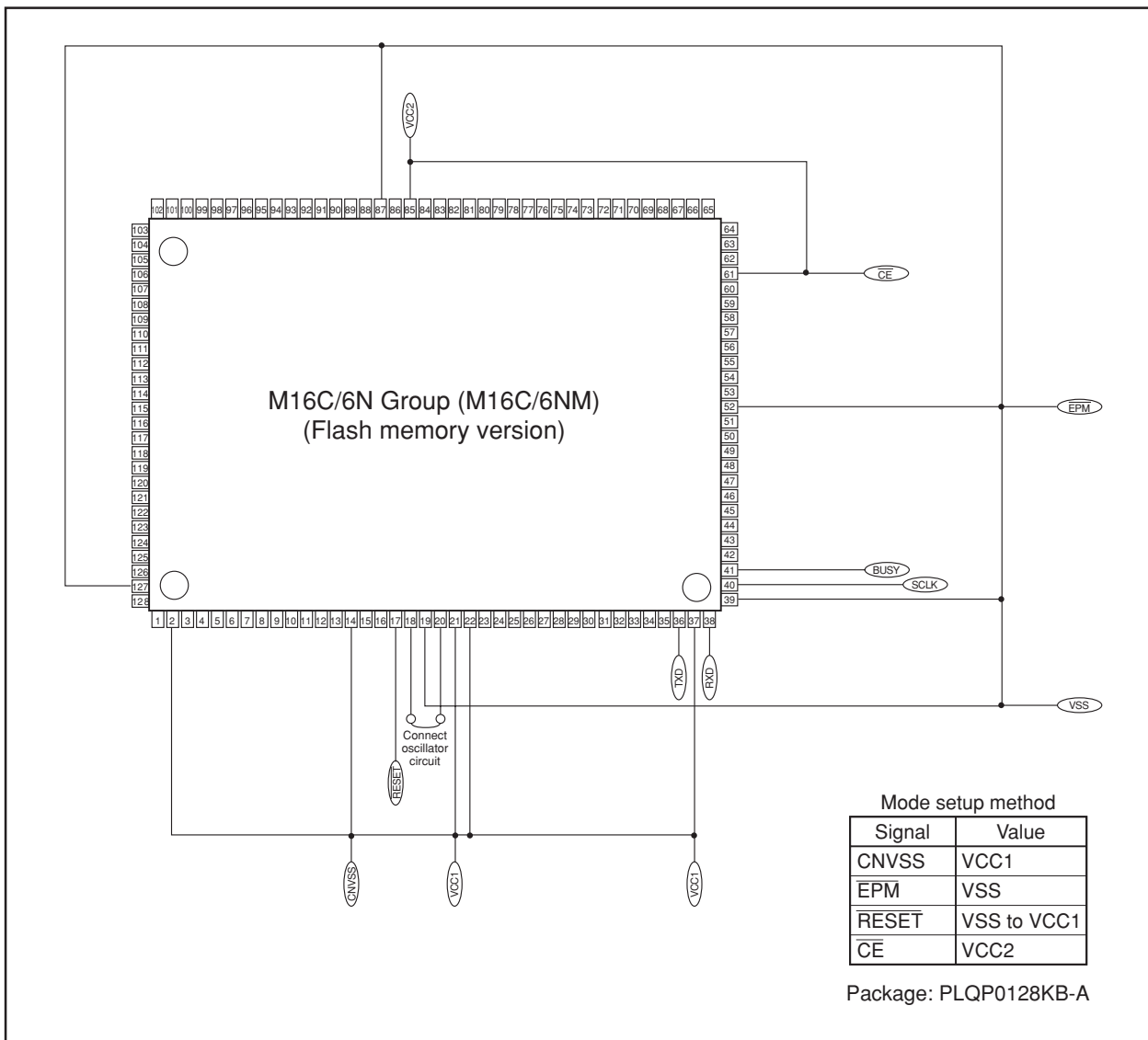


Figure 20.14 Pin Connections for Standard Serial I/O Mode (2)

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

Figures 20.15 and 20.16 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.

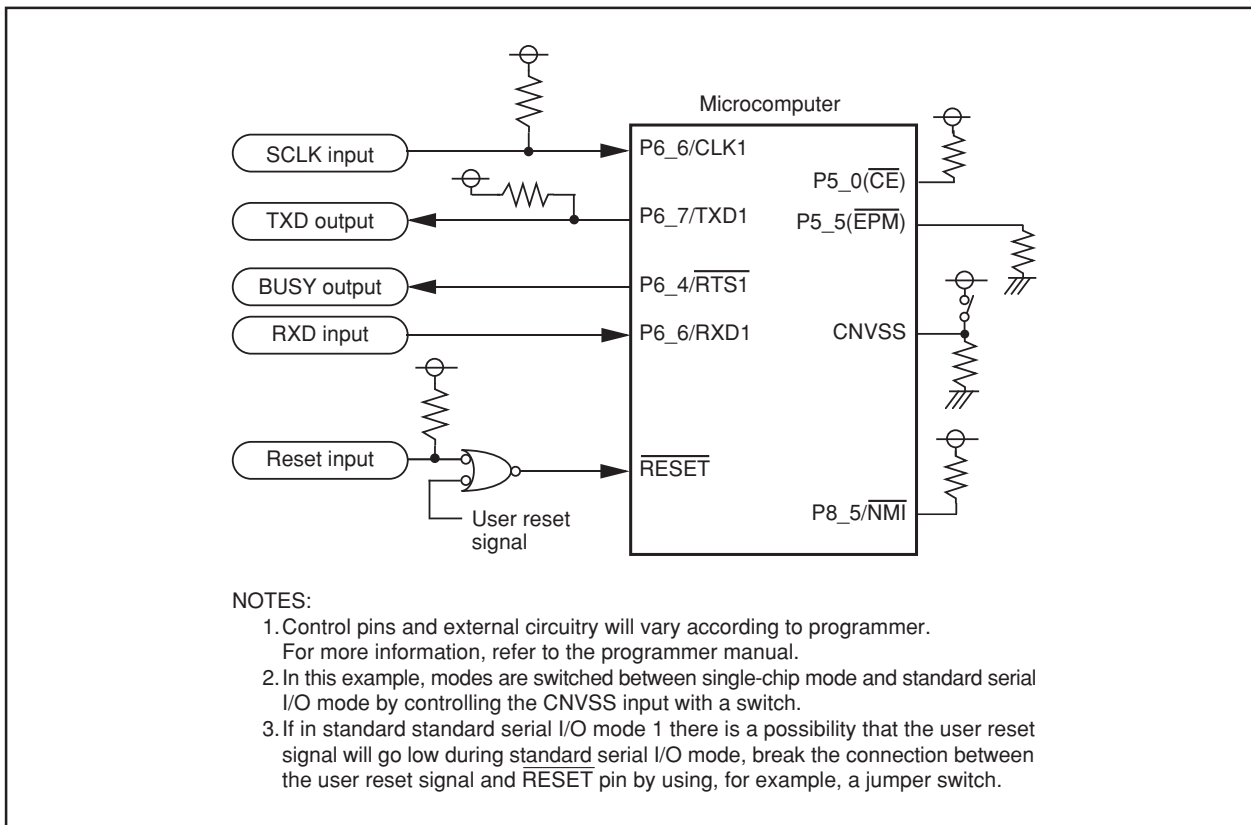


Figure 20.15 Circuit Application in Standard Serial I/O Mode 1

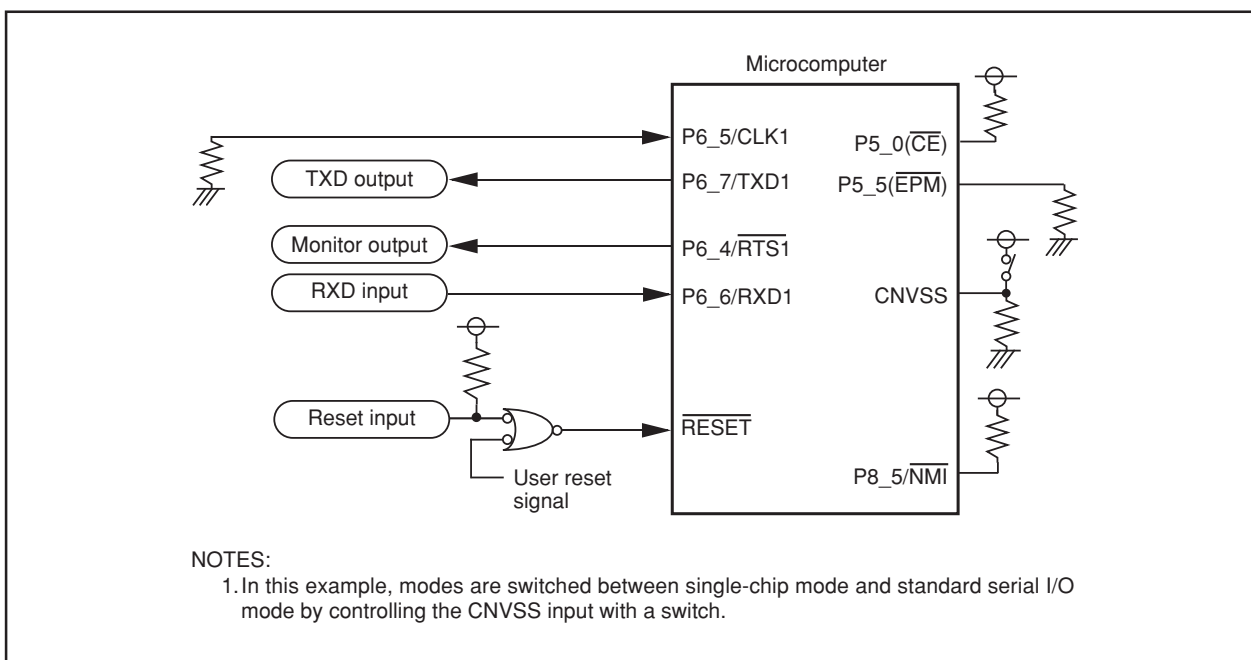


Figure 20.16 Circuit Application in Standard Serial I/O Mode 2

20.5 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/6NK, M16C/6NM). Contact your parallel programmer manufacturer for more information on the parallel programmer. Refer to the user's manual included with your parallel programmer for instructions.

20.5.1 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.)

20.5.2 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 **20.2 Functions to Prevent Flash Memory from Rewriting.**)

20.6 CAN I/O Mode

In CAN I/O mode, the CAN programmer supporting the M16C/6N Group (M16C/6NK, M16C/6NM) 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 20.8 lists pin functions for CAN I/O mode. Figures 20.17 and 20.18 show pin connections for CAN I/O mode.

20.6.1 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 **20.2 Functions to Prevent Flash Memory from Rewriting.**)

Table 20.8 Pin Functions for CAN I/O Mode

Pin	Name	I/O	Description
VCC1, VCC2, VSS	Power supply input		Apply the voltage guaranteed for Program and Erase 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_4, P8_6, P8_7	Input port P8	I	Input "H" or "L" level signal or open.
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.
P11_0 to P11_7 ⁽¹⁾	Input port P11	I	Input "H" or "L" level signal or open.
P12_0 to P12_7 ⁽¹⁾	Input port P12	I	Input "H" or "L" level signal or open.
P13_0 to P13_7 ⁽¹⁾	Input port P13	I	Input "H" or "L" level signal or open.
P14_0, P14_1 ⁽¹⁾	Input port P14	I	Input "H" or "L" level signal or open.

NOTE:

1. The pins P11 to P14 are only in the 128-pin version.

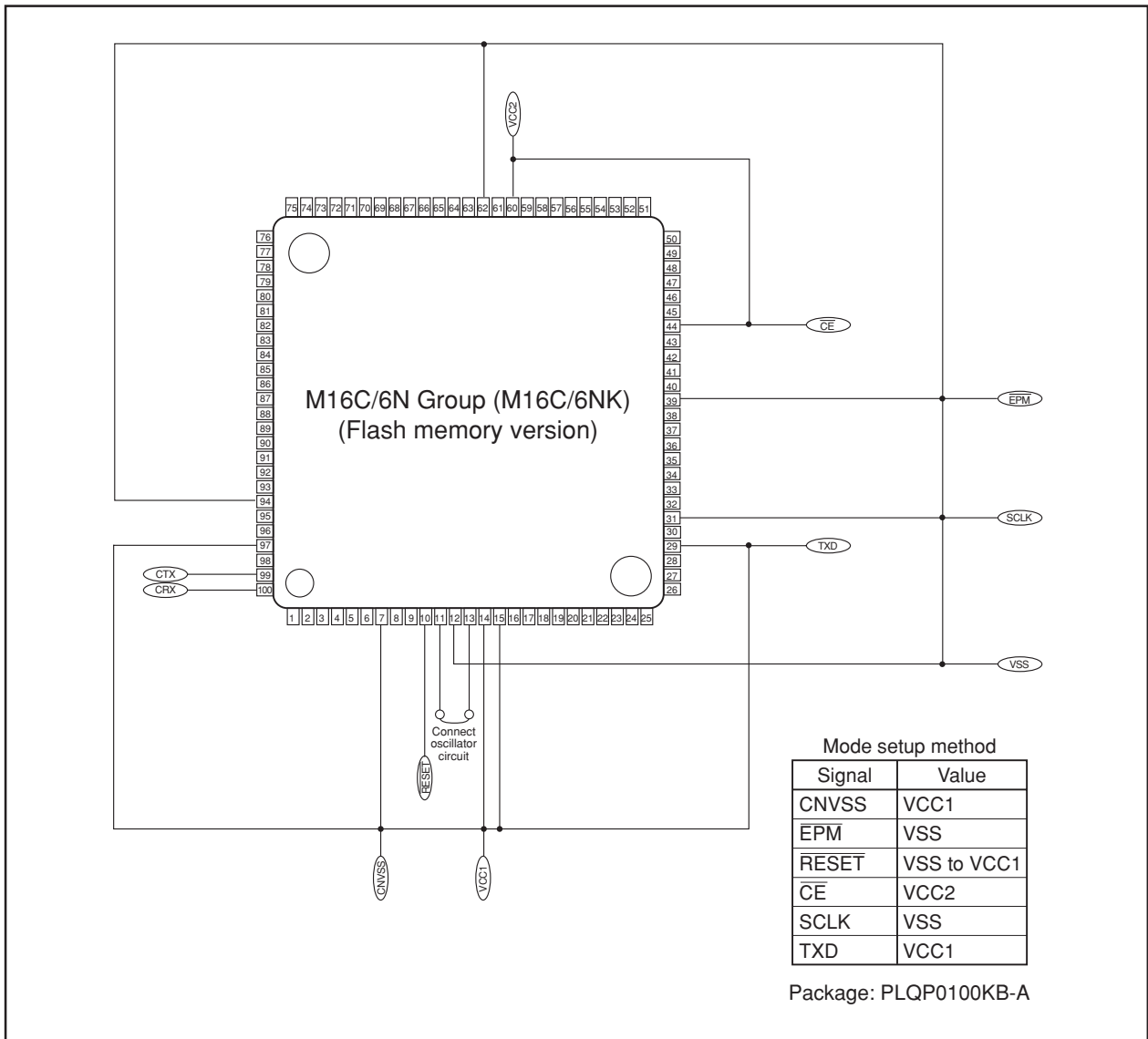


Figure 20.17 Pin Connections for CAN I/O Mode (1)

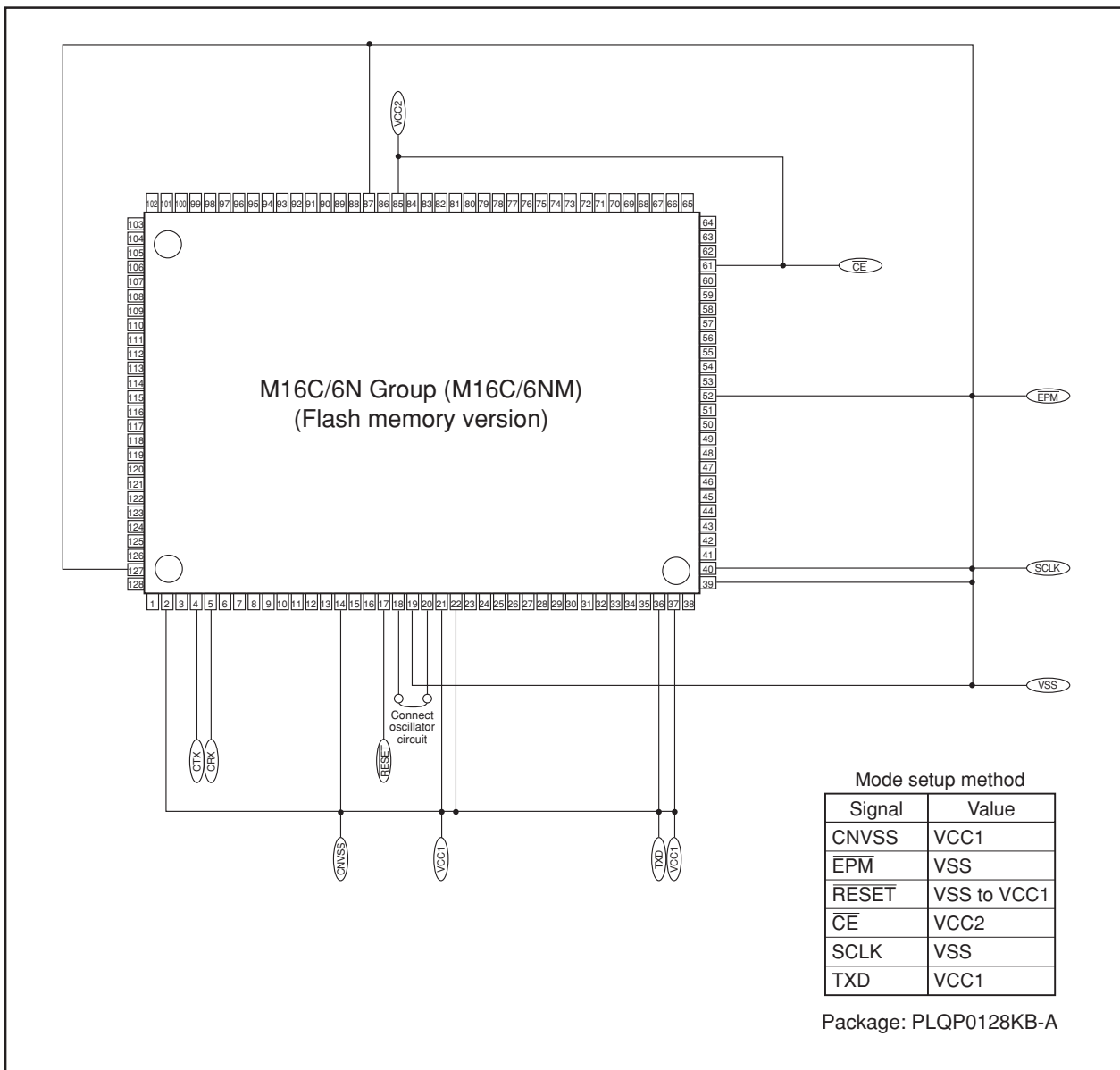


Figure 20.18 Pin Connections for CAN I/O Mode (2)

20.6.2 Example of Circuit Application in CAN I/O Mode

Figure 20.19 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.

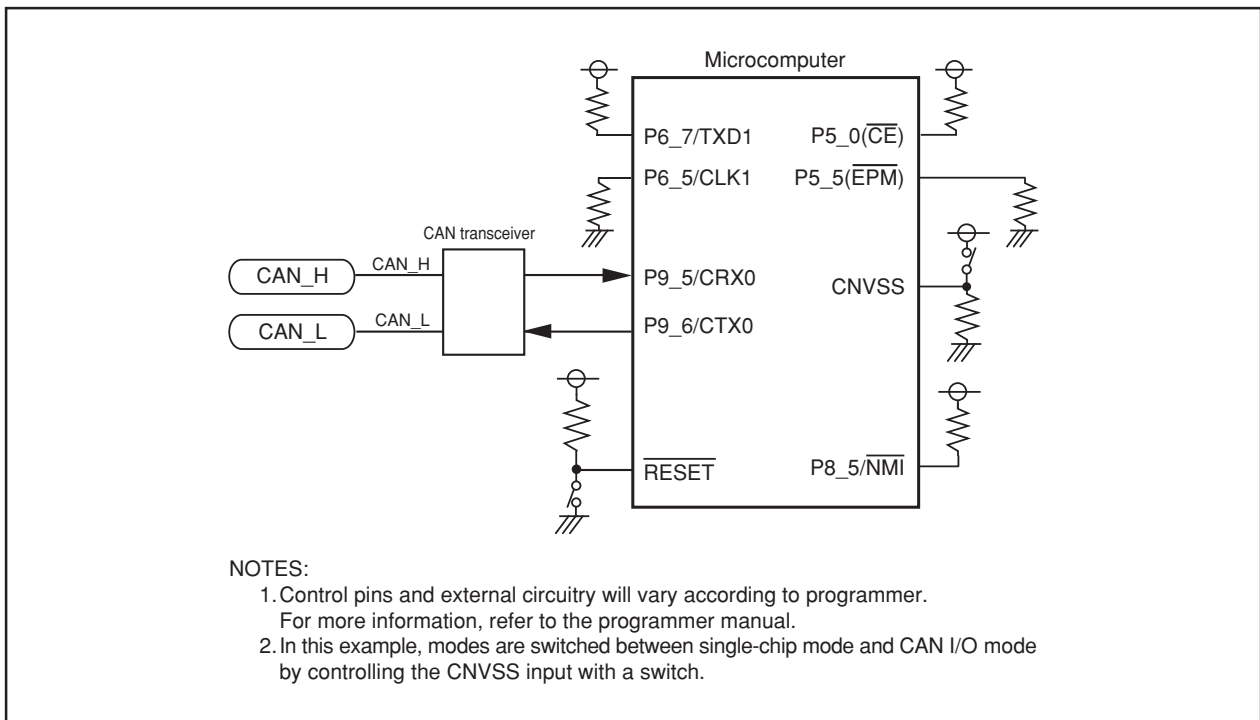


Figure 20.19 Circuit Application in CAN I/O Mode

21. Electrical Characteristics

Table 21.1 Absolute Maximum Ratings

Symbol	Parameter		Condition	Rated Value	Unit
V _{CC}	Supply Voltage (VCC1 = VCC2)		VCC = AVCC	-0.3 to 6.5	V
AV _{CC}	Analog Supply Voltage		VCC = AVCC	-0.3 to 6.5	V
V _i	Input Voltage	RESET, CNVSS, BYTE, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, VREF, XIN		-0.3 to VCC+0.3	V
		P7_1, P9_1		-0.3 to 6.5	V
V _o	Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XOUT		-0.3 to VCC+0.3	V
		P7_1, P9_1		-0.3 to 6.5	V
P _d	Power Dissipation		T _{opr} = 25°C	700	mW
T _{opr}	Operating Ambient Temperature	When the Microcomputer is Operating		-40 to 85	°C
		Flash Program Erase		0 to 60	
T _{stg}	Storage Temperature			-65 to 150	°C

NOTE:

1. Ports P11 to P14 are only in the 128-pin version.

Table 21.2 Recommended Operating Conditions (1) ⁽¹⁾

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V _{CC}	Supply Voltage (V _{CC1} = V _{CC2})		3.0	5.0	5.5	V
AV _{CC}	Analog Supply Voltage			V _{CC}		V
V _{SS}	Supply Voltage			0		V
AV _{SS}	Analog Supply Voltage			0		V
V _{IH}	HIGH Input Voltage	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, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, RESET, CNVSS, BYTE	0.8V _{CC}		V _{CC}	V
		P7_1, P9_1	0.8V _{CC}		6.5	V
V _{IL}	LOW Input 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_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, RESET, CNVSS, BYTE	0		0.2V _{CC}	V
I _{OH(peak)}	HIGH Peak Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1			-10.0	mA
I _{OH(avg)}	HIGH Average Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1			-5.0	mA
I _{OL(peak)}	LOW Peak Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1			10.0	mA
I _{OL(avg)}	LOW Average Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1			5.0	mA

NOTES:

1. Referenced to V_{CC} = 3.0 to 5.5V at T_{opr} = -40 to 85°C unless otherwise specified.
2. The mean output current is the mean value within 100 ms.
3. The total I_{OL(peak)} for ports P0, P1, P2, P8_6, P8_7, P9, P10, P11, P14_0 and P14_1 must be 80mA max.
The total I_{OL(peak)} for ports P3, P4, P5, P6, P7, P8_0 to P8_4, P12 and P13 must be 80mA max.
The total I_{OH(peak)} for ports P0, P1, and P2 must be -40mA max.
The total I_{OH(peak)} for ports P3, P4, P5, P12 and P13 must be -40mA max.
The total I_{OH(peak)} for ports P6, P7 and P8_0 to P8_4 must be -40mA max.
The total I_{OH(peak)} for ports P8_6, P8_7, P9, P10, P11, P14_0 and P14_1 must be -40mA max.
4. P11 to P14 are only in the 128-pin version.

Table 21.3 Recommended Operating Conditions (2) ⁽¹⁾

Symbol	Parameter				Standard			Unit
					Min.	Typ.	Max.	
f(XIN)	Main Clock Input Oscillation Frequency ^{(2) (3) (4)}	No Wait	Mask ROM Version Flash Memory Version	VCC = 3.0 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		24	MHz
f(BCLK)	CPU Operation Clock			VCC = 3.0 to 5.5V	0		24	MHz
t _{su(PLL)}	PLL Frequency Synthesizer Stabilization Wait Time						20	ms
f _(ripple)	Power Supply Ripple Allowable Frequency (VCC)						10	kHz
V _{P-P(ripple)}	Power Supply Ripple Allowable Amplitude Voltage			VCC = 5V			0.5	V
				VCC = 3V			0.3	
V _{CC(ΔV/ΔT)}	Power Supply Ripple Rising/Falling Gradient			VCC = 5V			0.3	V/ms
				VCC = 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 16MHz and over, use PLL clock. PLL clock oscillation frequency which can be used is 16MHz, 20MHz or 24MHz.

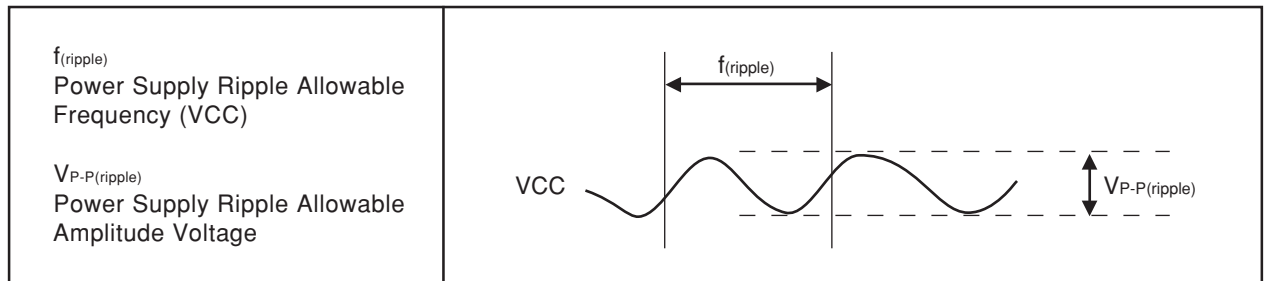
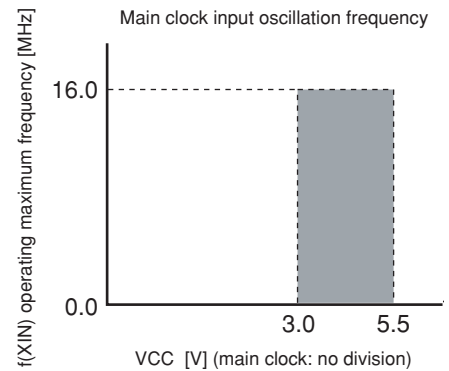


Figure 21.1 Timing of Voltage Fluctuation

Table 21.4 Electrical Characteristics (1) ⁽¹⁾

Symbol	Parameter		Measuring Condition	Standard			Unit	
				Min.	Typ.	Max.		
V _{OH}	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1	I _{OH} = -5mA	V _{CC} -2.0		V _{CC}	V	
V _{OH}	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1	I _{OH} = -200μA	V _{CC} -0.3		V _{CC}	V	
V _{OH}	HIGH Output Voltage	XOUT	HIGHPOWER	I _{OH} = -1mA	3.0	V _{CC}	V	
			LOWPOWER	I _{OH} = -0.5mA	3.0	V _{CC}		
	HIGH Output Voltage	XCOUT	HIGHPOWER	With no load applied		2.5		V
			LOWPOWER	With no load applied		1.6		
V _{OL}	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1	I _{OL} = 5mA			2.0	V	
V _{OL}	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1	I _{OL} = 200μA			0.45	V	
V _{OL}	LOW Output Voltage	XOUT	HIGHPOWER	I _{OL} = 1mA		2.0	V	
			LOWPOWER	I _{OL} = 0.5mA		2.0		
	LOW Output Voltage	XCOUT	HIGHPOWER	With no load applied		0		V
			LOWPOWER	With no load applied		0		
V _{T+} -V _{T-}	Hysteresis	TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT8, NMI, ADTRG, CTS0 to CTS2, SCL0 to SCL2, SDA0 to SDA2, CLK0 to CLK6, TA0OUT to TA4OUT, K10 to K13, RXD0 to RXD2, SIN3 to SIN6		0.2		1.0	V	
V _{T+} -V _{T-}	Hysteresis	RESET		0.2		2.5	V	
V _{T+} -V _{T-}	Hysteresis	XIN		0.2		0.8	V	
I _{IH}	HIGH Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, RESET, CNVSS, BYTE	V _I = 5V			5.0	μA	
I _{IL}	LOW Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, RESET, CNVSS, BYTE	V _I = 0V			-5.0	μA	
R _{PULLUP}	Pull-up Resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1	V _I = 0V	30	50	170	kΩ	
R _{I_{XIN}}	Feedback Resistance	XIN			1.5		MΩ	
R _{I_{XCIN}}	Feedback Resistance	XCIN			15		MΩ	
V _{RAM}	RAM Retention Voltage		At stop mode	2.0			V	

NOTES:

1. Referenced to VCC = 3.0 to 5.5V, VSS = 0V at T_{opr} = -40 to 85°C, f(BCLK) = 24MHz unless otherwise specified.
2. P11 to P14, INT6 to INT8, CLK5, CLK6, SIN5 and SIN6 are only in the 128-pin version.

Table 21.5 Electrical Characteristics (2) ⁽¹⁾

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
I _{cc}	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		21	37	mA
				On-chip oscillation, No division		1		mA
			Flash Memory	f(BCLK) = 24MHz, PLL operation, No division		23	39	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 ⁽²⁾		25		μA
			Flash Memory	f(BCLK) = 32kHz, Low power dissipation mode, RAM ⁽²⁾		25		μA
				f(BCLK) = 32kHz, Low power dissipation mode, Flash memory ⁽²⁾		420		μA
				Mask ROM Flash Memory	On-chip oscillation, Wait mode		50	
			Flash Memory	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, Topr = 25°C		0.8	3.0	μA

NOTES:

1. Referenced to VCC = 3.0 to 5.5V, VSS = 0V at Topr = -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.

Table 21.6 A/D Conversion Characteristics ⁽¹⁾

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
–	Resolution		VREF = VCC				10	Bit
INL	Integral Nonlinearity Error	10 bits	VREF = VCC	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±3	LSB
				External operation amp connection mode			±7	LSB
		8 bits	VREF = AVCC = VCC = 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
–	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
		8 bits	VREF = AVCC = VCC = 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
DNL	Differential Nonlinearity Error						±1	LSB
–	Offset Error						±3	LSB
–	Gain Error						±3	LSB
R _{LADDER}	Resistor Ladder		VREF = VCC		10		40	kΩ
t _{CONV}	10-bit Conversion Time, Sample & Hold function Available		VREF = VCC = 5V, φ _{AD} = 10MHz		3.3			μs
	8-bit Conversion time, Sample & Hold function Available		VREF = VCC = 5V, φ _{AD} = 10MHz		2.8			μs
t _{SAMP}	Sampling Time				0.3			μs
V _{REF}	Reference Voltage				2.0		V _{CC}	V
V _{IA}	Analog Input Voltage				0		V _{REF}	V

NOTES:

1. Referenced to VCC = AVCC = VREF = 3.3 to 5.5V, VSS = AVSS = 0V, –40 to 85°C unless otherwise specified.
2. φ_{AD} frequency must be 10MHz or less.
3. When sample & hold function is disabled, φ_{AD} frequency must be 250kHz or more in addition to a limit of NOTE 2. When sample & hold function is enabled, φ_{AD} frequency must be 1MHz or more in addition to a limit of NOTE 2.

Table 21.7 D/A conversion Characteristics ⁽¹⁾

Symbol	Parameter	Measuring Condition	Standard			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute Accuracy				1.0	%
t _{su}	Setup Time				3	μs
R _O	Output Resistance		4	10	20	kΩ
I _{VREF}	Reference Power Supply Input Current	(NOTE 2)			1.5	mA

NOTES:

1. Referenced to VCC = AVCC = VREF = 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 DA_i register (i = 0, 1) for the unused D/A converter set to “00h”. The resistor ladder of the A/D converter is not included. Also, the current I_{VREF} always flows even though VREF may have been set to be unconnected by the ADCON1 register.

Table 21.8 Flash Memory Version Electrical Characteristics ⁽¹⁾

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
-	Word Program Time		30	200	μs
-	Block Erase Time		1	4	s
-	Erase All Unlocked Blocks Time		1 × n ⁽²⁾	4 × n ⁽²⁾	s
-	Lock Bit Program Time		30	200	μs
t _{ps}	Flash Memory Circuit Stabilization Wait Time			15	μs

NOTES:

1. Referenced to VCC = 4.5 to 5.5V, 3.0 to 3.6V, T_{opr} = 0 to 60°C unless otherwise specified.
2. n denotes the number of blocks to erase.

Table 21.9 Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics (at T_{opr} = 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

Table 21.10 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	μs
t _{d(W-S)}	Low Power Dissipation Mode Wait Mode Release Time				150	μs

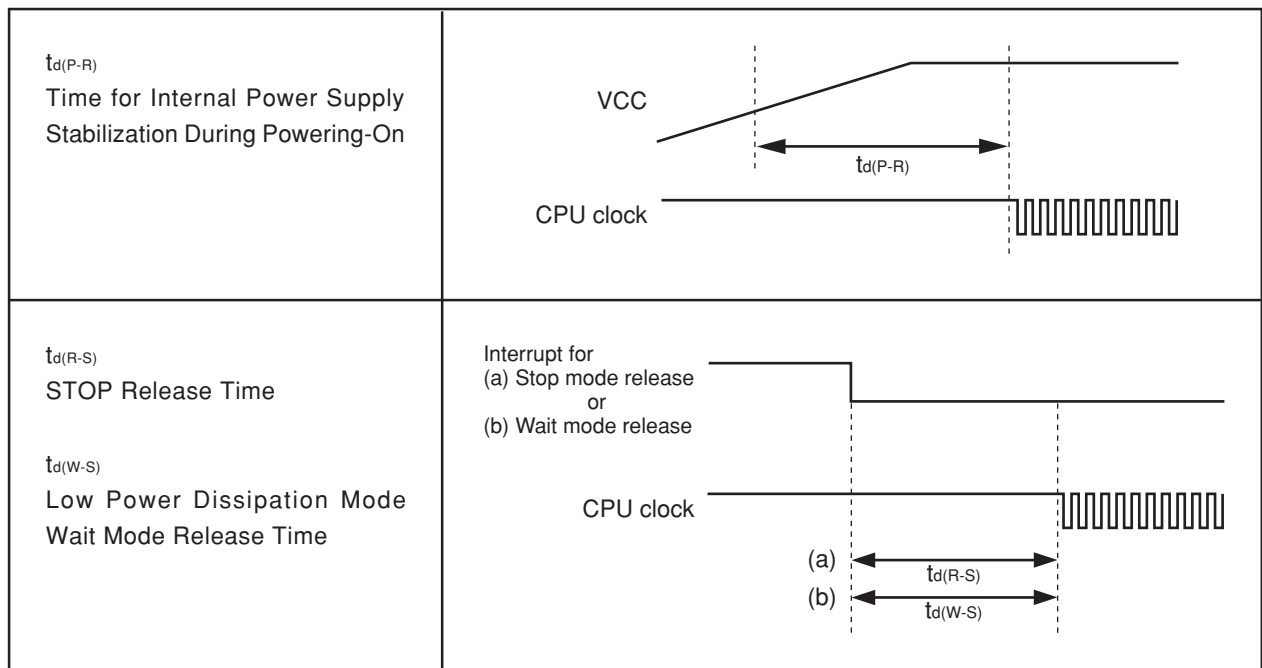


Figure 21.2 Power Supply Circuit Timing Diagram

Timing Requirements

(Referenced to VCC = 5V, VSS = 0V, at Topr = -40 to 85°C unless otherwise specified)

Table 21.11 External Clock Input (XIN Input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c	External Clock Input Cycle Time	62.5		ns
t _{w(H)}	External Clock Input HIGH Pulse Width	25		ns
t _{w(L)}	External Clock Input LOW Pulse Width	25		ns
t _r	External Clock Rise Time		15	ns
t _f	External Clock Fall Time		15	ns

Table 21.12 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(TA)}	TAiN Input Cycle Time	100		ns
t _{w(TAH)}	TAiN Input HIGH Pulse Width	40		ns
t _{w(TAL)}	TAiN Input LOW Pulse Width	40		ns

Table 21.13 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(TA)}	TAiN Input Cycle Time	400		ns
t _{w(TAH)}	TAiN Input HIGH Pulse Width	200		ns
t _{w(TAL)}	TAiN Input LOW Pulse Width	200		ns

Table 21.14 Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(TA)}	TAiN Input Cycle Time	200		ns
t _{w(TAH)}	TAiN Input HIGH Pulse Width	100		ns
t _{w(TAL)}	TAiN Input LOW Pulse Width	100		ns

Table 21.15 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{w(TAH)}	TAiN Input HIGH Pulse Width	100		ns
t _{w(TAL)}	TAiN Input LOW Pulse Width	100		ns

Table 21.16 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(UP)}	TAiOUT Input Cycle Time	2000		ns
t _{w(UPH)}	TAiOUT Input HIGH Pulse Width	1000		ns
t _{w(UPL)}	TAiOUT Input LOW Pulse Width	1000		ns
t _{su(UP-TIN)}	TAiOUT Input Setup Time	400		ns
t _{h(TIN-UP)}	TAiOUT Input Hold Time	400		ns

Table 21.17 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(TA)}	TAiN Input Cycle Time	800		ns
t _{su(TAIN-TAOUT)}	TAiOUT Input Setup Time	200		ns
t _{su(TAOUT-TAIN)}	TAiN Input Setup Time	200		ns

Timing Requirements**(Referenced to VCC = 5V, VSS = 0V, at Topr = –40 to 85°C unless otherwise specified)****Table 21.18 Timer B Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on both edges)	80		ns

Table 21.19 Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	400		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	200		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	200		ns

Table 21.20 Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	400		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	200		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	200		ns

Table 21.21 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	ADTRG Input Cycle Time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	ADTRG Input LOW Pulse Width	125		ns

Table 21.22 Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi Input Cycle Time	200		ns
$t_{w(CKH)}$	CLKi Input HIGH Pulse Width	100		ns
$t_{w(CKL)}$	CLKi Input LOW Pulse Width	100		ns
$t_{d(C-Q)}$	TXDi Output Delay Time		80	ns
$t_{h(C-Q)}$	TXDi Hold Time	0		ns
$t_{su(D-C)}$	RXDi Input Setup Time	70		ns
$t_{h(C-D)}$	RXDi Input Hold Time	90		ns

Table 21.23 External Interrupt INTi Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	INTi Input HIGH Pulse Width	250		ns
$t_{w(INL)}$	INTi Input LOW Pulse Width	250		ns

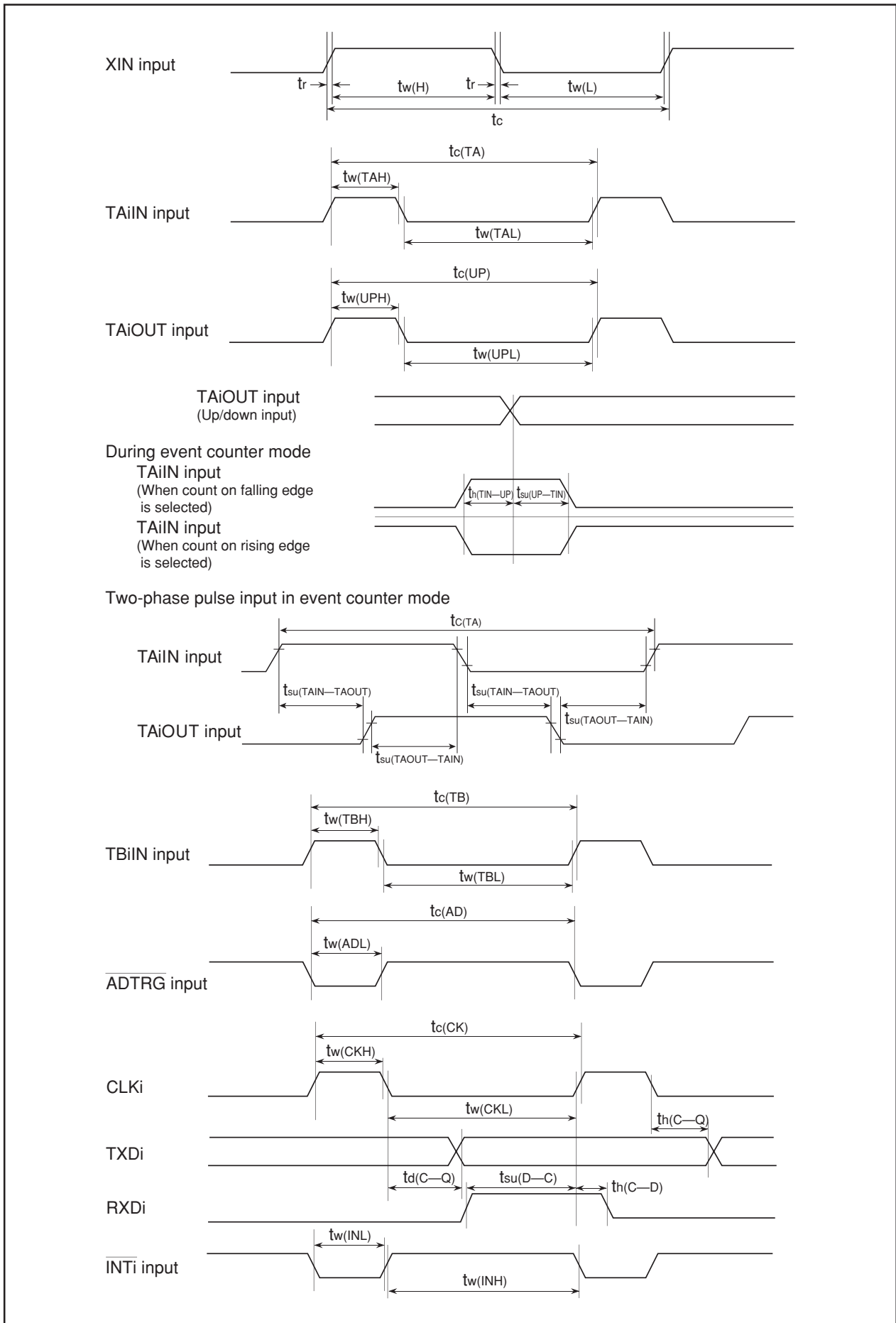


Figure 21.3 Timing Diagram

22. Usage Precaution

22.1 SFR

There is the SFR which can not be read (containing bits that will result in unknown data when read).

Please set these registers to their previous values with the instructions other than the read modify write instructions.

Table 22.1 lists the registers contain bits that will result in unknown data when read and Table 22.2 lists the instruction table for read modify write.

Table 22.1 Registers Contain Bits that Will Result in Unknown Data When Read

Register Name	Symbol	Address
Timer A1-1 Register ⁽¹⁾	TA11	01C3h, 01C2h
Timer A2-1 Register ⁽¹⁾	TA21	01C5h, 01C4h
Timer A4-1 Register ⁽¹⁾	TA41	01C7h, 01C6h
Dead Time Timer	DTT	01CCh
Timer B2 Interrupt Occurrences Frequency Set Counter	ICTB2	01CDh
SI/O6 Bit Rate Generator ⁽²⁾	S6BRG	01D9h
SI/O3 Bit Rate Generator	S3BRG	01E3h
SI/O4 Bit Rate Generator	S4BRG	01E7h
SI/O5 Bit Rate Generator ⁽²⁾	S5BRG	01EBh
UART2 Bit Rate Generator	U2BRG	01F9h
UART2 Transmit Buffer Register	U2TB	01FBh, 01FAh
Up-Down Flag	UDF	0384h
Timer A0 Register ⁽³⁾	TA0	0387h, 0386h
Timer A1 Register ⁽¹⁾⁽³⁾	TA1	0389h, 0388h
Timer A2 Register ⁽¹⁾⁽³⁾	TA2	038Bh, 038Ah
Timer A3 Register ⁽³⁾	TA3	038Dh, 038Ch
Timer A4 Register ⁽¹⁾⁽³⁾	TA4	038Fh, 038Eh
UART0 Bit Rate Generator	U0BRG	03A1h
UART0 Transmit Buffer Register	U0TB	03A3h, 03A2h
UART1 Bit Rate Generator	U1BRG	03A9h
UART1 Transmit Buffer Register	U1TB	03ABh, 03AAh

NOTES:

1. It is affected only in three-phase motor control timer function.
2. These registers are only in the 128-pin version.
3. It is affected only in one-shot timer mode and pulse width modulation mode.

Table 22.2 Instruction Table for Read Modify Write

Function	Mnemonic
Bit Manipulation	BCLR, BNOT, BSET, BTSTC, BTSTS
Shift	RCLC, RORC, ROT, SHA, SHL
Arithmetic	ABS, ADC, ADCF, ADD, DEC, EXTS, INC, MUL, MULU, NEG, SBB, SUB
Logical	AND, NOT, OR, XOR
Jump	ADJNZ, SBJNZ

22.2 External Clock

Do not stop the external clock when it is connected to the XIN pin and the main clock is selected as the CPU clock.

22.3 PLL Frequency Synthesizer

Stabilize supply voltage so that the standard of the power supply ripple is met. (Refer to **21. Electrical characteristics.**)

22.4 Power Control

When exiting stop mode by hardware reset, set $\overline{\text{RESET}}$ pin to “L” until a main clock oscillation is stabilized.

Set the MR0 bit in the TAI MR register (i = 0 to 4) to “0” (pulse is not output) to use the timer A to exit stop mode.

Insert more than four NOP instructions after an WAIT instruction or a instruction to set the CM10 bit in the CM1 register to “1” (all clock stopped). When shifting to wait mode or stop mode, an instruction queue reads ahead to the next instruction to halt a program by an WAIT instruction and an instruction to set the CM10 bit to “1”. The next instruction may be executed before entering wait mode or stop mode, depending on a combination of instruction and an execution timing.

In the main clock oscillation or low power dissipation mode, set the CM02 bit in the CM0 register to “0” (do not stop peripheral function clock in wait mode) before shifting to stop mode.

When entering wait mode by executing the WAIT instruction after writing to addresses 03FDh to 03FFh or internal RAM area, execute the JMP.B instruction between writing to corresponding area and the executing the WAIT instruction.

If DMA transfer may occur between executing the JMP.B instruction and the WAIT instruction, set the DMAE bit (DMA enable bit) in the DMiCOM register (i = 0, 1) to “0” (disabled) before executing the WAIT instruction.

```
Example program MOV.B    #55H, 0601H    ; Write to internal RAM area
                JMP.B    L1
                L1:
                FSET    I                ; Enable interrupt
                WAIT                    ; Enter to wait mode
```

When using the interrupt to exit stop mode, the fifth instruction ⁽¹⁾ from the instruction to enter the stop mode may be executed before executing a program of the interrupt to exit stop mode.

If this execution causes no problem with the system, there are no need for measures to be taken ⁽²⁾.

If such a situation presents a problem, execute the JMP.B instruction subsequent to the instruction which sets the CM10 bit to “1” (stop mode).

```
Example program BSET    0, CM1        ; Stop mode
                JMP.B    L1
                L1:
                Program after exiting stop mode
```

NOTES:

1. Insert more than four NOP instructions after the instruction shifting to wait mode or stop mode.
2. In the flash memory version, be sure to execute the measures. For details, refer to **22.18.2 Stop Mode**.

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 μ s or longer after setting the VCUT bit to “1” (VREF connection).

D/A converter

When not performing D/A conversion, set the DAiE bit (i = 0, 1) in the DACON register to “0” (input disabled) and DAi register to “00h”.

Switching the oscillation-driving capacity

Set the driving capacity to “LOW” when oscillation is stable.

22.5 Oscillation Stop, Re-oscillation Detection Function

If the following conditions are all met, the following restriction occur in operation of oscillation stop, re-oscillation stop detection interrupt.

Conditions

- CM20 bit in CM2 register =1 (oscillation stop, re-oscillation stop detection function enabled)
- CM27 bit in CM2 register =1 (oscillation stop, re-oscillation stop detection interrupt)
- CM02 bit in CM0 register =0 (do not stop peripheral function clock in wait mode)
- Enter wait mode from high-speed or middle-speed mode

Restriction

If the oscillation of XIN stops during wait mode, the oscillation stop, re-oscillation stop detection interrupt request is generated after the microcomputer is moved out of wait mode, without starting immediately.

Figures 22.1 and 22.2 show the operation timing at oscillation stop, re-oscillation stop detection.

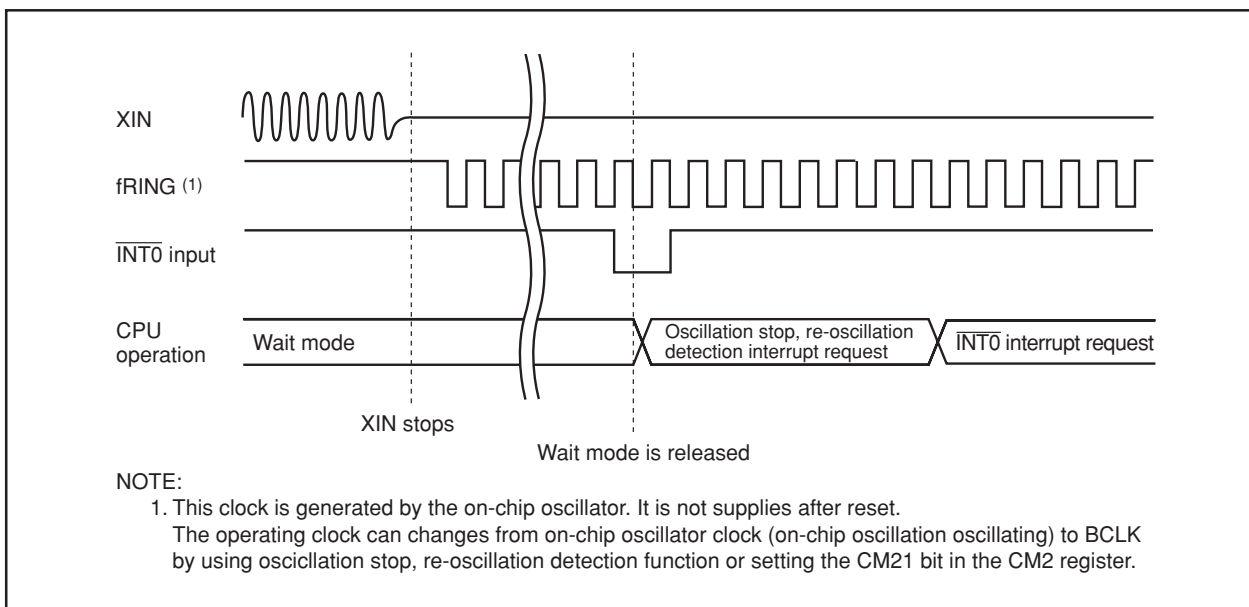


Figure 22.1 Operation Timing at Oscillation Stop, Re-oscillation Stop Detection at Wait Mode (when moving out of wait mode by using INT0 interrupt)

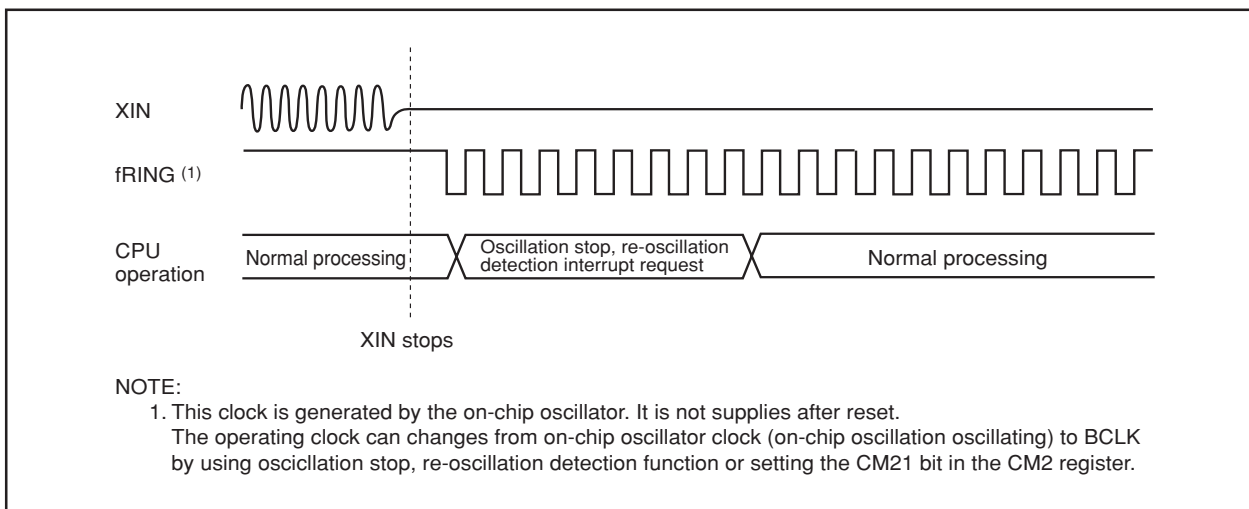


Figure 22.2 Operation Timing at Oscillation Stop, Re-oscillation Stop Detection at Normal Processing

22.6 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.

22.7 Interrupt

22.7.1 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.

22.7.2 Setting SP

Set any value in the SP (USP, ISP) before accepting an interrupt. The SP (USP, ISP) is set to "0000h" after reset. Therefore, if an interrupt is accepted before setting any value in the SP (USP, ISP), the program may go out of control.

Especially when using $\overline{\text{NMI}}$ interrupt, set a value in the ISP at the beginning of the program. For the first and only the first instruction after reset, all interrupts including $\overline{\text{NMI}}$ interrupt are disabled.

22.7.3 $\overline{\text{NMI}}$ 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.

22.7.4 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 22.3 shows the procedure for changing the interrupt generate factor.

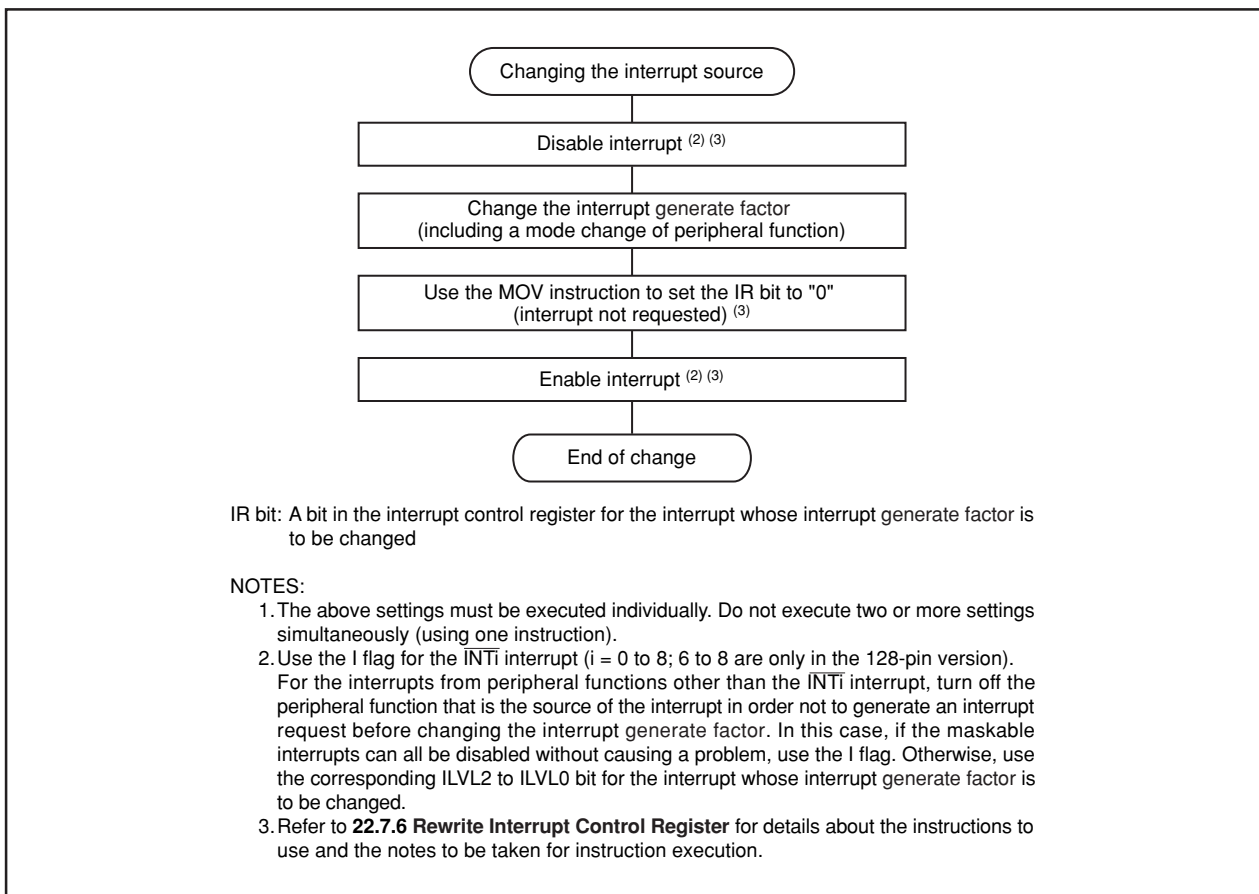


Figure 22.3 Procedure for Changing Interrupt Generate Factor

22.7.5 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 $\overline{\text{INT}}_8$ ⁽¹⁾ regardless of the CPU operation clock.
- If the POL bit in the INT0IC to INT8IC registers ⁽²⁾, the IFSR10 to IFSR15 bits in the IFSR1 register or the IFSR23 to IFSR25 bits ⁽³⁾ in the IFSR2 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.

NOTES:

1. The pins $\overline{\text{INT}}_6$ to $\overline{\text{INT}}_8$ are only in the 128-pin version.
2. The INT6IC to INT8IC registers are only in the 128-pin version.
3. The IFSR23 to IFSR25 bits are effective only in the 128-pin version. In the 100-pin version, these bits are set to “0” (one edge).

22.7.6 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.
```

22.7.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt request is generated.

22.8 DMAC

22.8.1 Write to DMAE Bit in DMiCON Register (i = 0, 1)

When both of the conditions below are met, follow the steps below.

Conditions

- The DMAE bit is set to “1” again while it remains set (DMAi is in an active state).
- A DMA request may occur simultaneously when the DMAE bit is being written.

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

Step 2: Make sure that the DMAi is in an initial state ⁽²⁾ in a program.

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

NOTES:

1. The DMAS bit remains unchanged even if “1” is written. However, if “0” is written to this bit, it is set to “0” (DMA not requested). In order to prevent the DMAS bit from being modified to “0”, “1” should be written to the DMAS bit when “1” is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.
Similarly, when writing to the DMAE bit with a read-modify-write instruction, “1” should be written to the DMAS bit in order to maintain a DMA request which is generated during execution.
2. Read the TCRi register to verify whether the DMAi is in an initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in an initial state. (If a DMA request occurs after writing to the DMAE bit, the value written to the TCRi register is “1”.) If the read value is a value in the middle of transfer, the DMAi is not in an initial state.

22.9 Timers

22.9.1 Timer A

22.9.1.1 Timer A (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_iMR (i = 0 to 4) register and the TAI register before setting the TAI_S bit in the TABSR register to "1" (count starts). Always make sure the TAI_iMR register is modified while the TAI_S bit remains "0" (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, if the counter is read at the same time it is reloaded, the value "FFFFh" is read. Also, if the counter is read before it starts counting after a value is set in the TAI register while not counting, the set value is read.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

22.9.1.2 Timer A (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_iMR (i = 0 to 4) register, the TAI register, the UDF register, the TAZIE, TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAI_S bit in the TABSR register to “1” (count starts). Always make sure the TAI_iMR register, the UDF register, the TAZIE, TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register are modified while the TAI_S bit remains “0” (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, “FFFFh” can be read in underflow, while reloading, and “0000h” in overflow. When setting the TAI register to a value during a counter stop, the setting value can be read before a counter starts counting. Also, if the counter is read before it starts counting after a value is set in the TAI register while not counting, the set value is read.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

22.9.1.3 Timer A (One-shot Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_{MR} (i = 0 to 4) register, the TAI register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAI_S bit in the TABSR register to “1” (count starts).

Always make sure the TAI_{MR} register, the TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAI_S bit remains “0” (count stops) regardless whether after reset or not.

When setting the TAI_S bit to “0” (count stop), the followings occur:

- A counter stops counting and a content of reload register is reloaded.
- TAI_{OUT} pin outputs “L”.
- After one cycle of the CPU clock, the IR bit in the TAI_{IC} register is set to “1” (interrupt request).

Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger has been selected, one-cycle delay of a count source as maximum occurs between a trigger input to TAI_{IN} pin and output in one-shot timer mode.

The IR bit is set to “1” when timer 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.

When the external trigger is selected as count start condition, do not input again the external trigger between 300 ns before the counter reaches “0000h”.

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.

22.9.1.4 Timer A (Pulse Width Modulation Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_iMR (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_iS bit in the TABSR register to “1” (count starts).

Always make sure the TAI_iMR register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register are modified while the TAI_iS 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 A_i interrupt (the IR bit), set the IR bit to “0” by program after the above listed changes have been made.

When setting TAI_iS bit to “0” (count stop) during PWM pulse output, the following action occurs:

- Stop counting.
- When TAI_iOUT pin is output “H”, output level is set to “L” and the IR bit is set to “1”.
- When TAI_iOUT pin is output “L”, both output level and the IR bit remain unchanged.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

22.9.2 Timer B

22.9.2.1 Timer B (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 5) register and TBi register before setting the TBiS bit ⁽¹⁾ in the TABSR or the TBSR register to “1” (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not.

NOTE:

1. 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.

22.9.2.2 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.

22.9.2.3 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 within the interrupt routine.

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.

22.10 Three-Phase Motor Control Timer Function

If there is a possibility that you may write data to TAI-1 register ($i = 1, 2, 4$) near Timer B2 overflow, read the value of TB2 register, verify that there is sufficient time until Timer B2 overflows, before doing an immediate write to TAI-1 register.

In order to shorten the period from reading TB2 register to writing data to TAI-1 register, ensure that no interrupt will be processed during this period.

If there is not enough time till Timer B2 overflows, only write to TAI-1 register after Timer B2 overflowed.

22.11 Serial I/O

22.11.1 Clock Synchronous Serial I/O Mode

22.11.1.1 Transmission/reception

With an external clock selected, and choosing the $\overline{\text{RTS}}$ function, the output level of the $\overline{\text{RTSi}}$ pin goes to “L” when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{\text{RTSi}}$ pin goes to “H” when reception starts. So if the $\overline{\text{RTSi}}$ pin is connected to the $\overline{\text{CTS}}$ pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the $\overline{\text{RTS}}$ function has no effect.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the $\overline{\text{RTS2}}$ and CLK2 pins go to a high-impedance state.

22.11.1.2 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 $\overline{\text{CTS}}$ pin = L

22.11.1.3 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)

22.11.2 Special Modes

22.11.2.1 Special Mode 1 (I²C Mode)

When generating start, stop and restart conditions, set the STSPSEL bit in the UiSMR4 register to “0” (start and stop conditions not output) and wait for more than half cycle of the transfer clock before setting each condition generate bit (STAREQ, RSTAREQ and STPREQ bits) from “0” (clear) to “1” (start).

22.11.2.2 Special Mode 2

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), the $\overline{\text{RTS2}}$ and CLK2 pins go to a high-impedance state.

22.11.2.3 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.

22.11.3 SI/Oi (i = 3 to 6) ⁽¹⁾

The SOUTi default value which is set to the SOUTi pin by the SMi7 in the SiC register bit approximately 10ns may be output when changing the SMi3 bit in the SiC register from “0” (I/O port) to “1” (SOUTi output and CLKi function) while the SMi2 bit in the SiC register to “0” (SOUTi output) and the SMi6 bit is set to “1” (internal clock). And then the SOUTi pin is held high-impedance.

If the level which is output from the SOUTi pin is a problem when changing the SMi3 bit from “0” to “1”, set the default value of the SOUTi pin by the SMi7 bit.

NOTE:

1. SI/O5 and SI/O6 are only in the 128-pin version.

22.12 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 μ s or longer.

To prevent noise-induced device malfunction or latch-up, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins (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 22.4 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 $\overline{\text{ADTRG}}$ 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 ϕ_{AD} frequency must be 10 MHz or less. Without sample-and-hold function, limit the ϕ_{AD} frequency to 250 kHz or more. With the sample and hold function, limit the ϕ_{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.

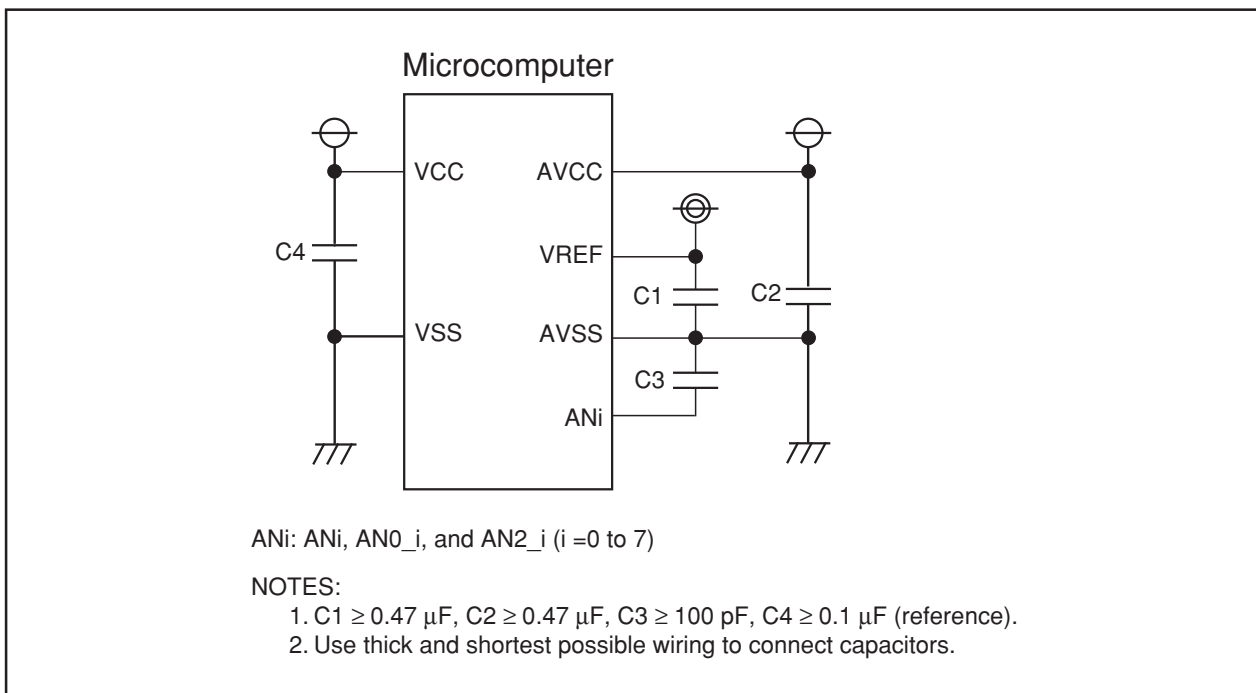


Figure 22.4 Use of Capacitors to Reduce Noise

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”.

22.13 CAN Module

22.13.1 Reading CiSTR Register (i = 0, 1)

The CAN module on the M16C/6N Group (M16C/6NK, M16C/6NM) 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 22.5 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 $3f_{CAN}$ or longer (see **Table 22.3 CAN Module Status Updating Period**) before the CPU reads the CiSTR register. (See **Figure 22.6 With a Wait Time of $3f_{CAN}$ Before CPU Read.**)
- (b) When the CPU polls the CiSTR register, the polling period must be $3f_{CAN}$ or longer. (See **Figure 22.7 When Polling Period of CPU is $3f_{CAN}$ or Longer.**)

Table 22.3 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

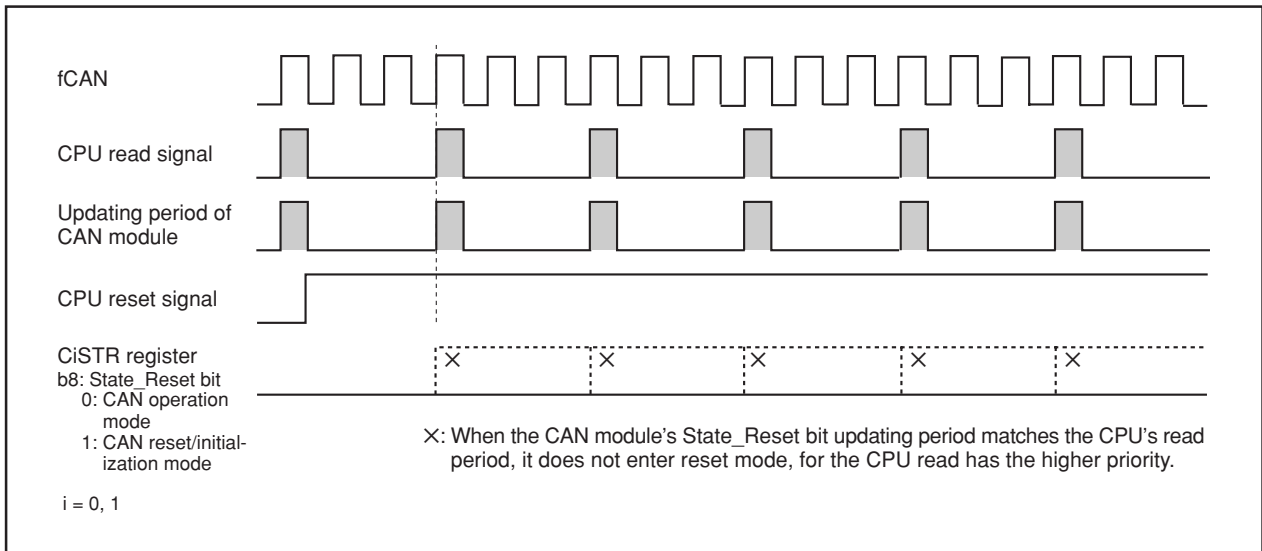


Figure 22.5 When Updating Period of CAN Module Matches Access Period from CPU

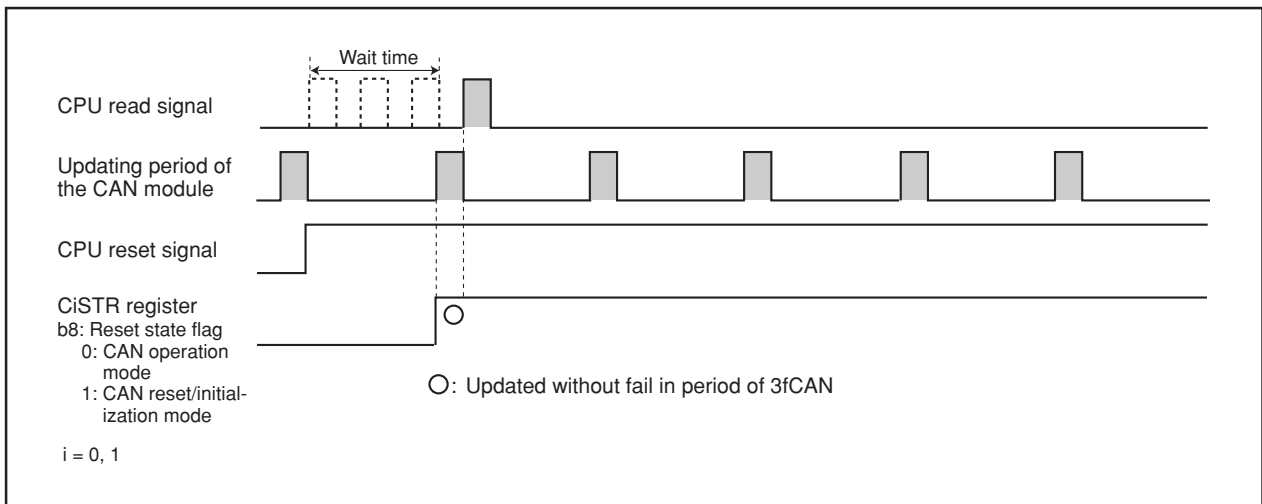


Figure 22.6 With a Wait Time of 3fCAN Before CPU Read

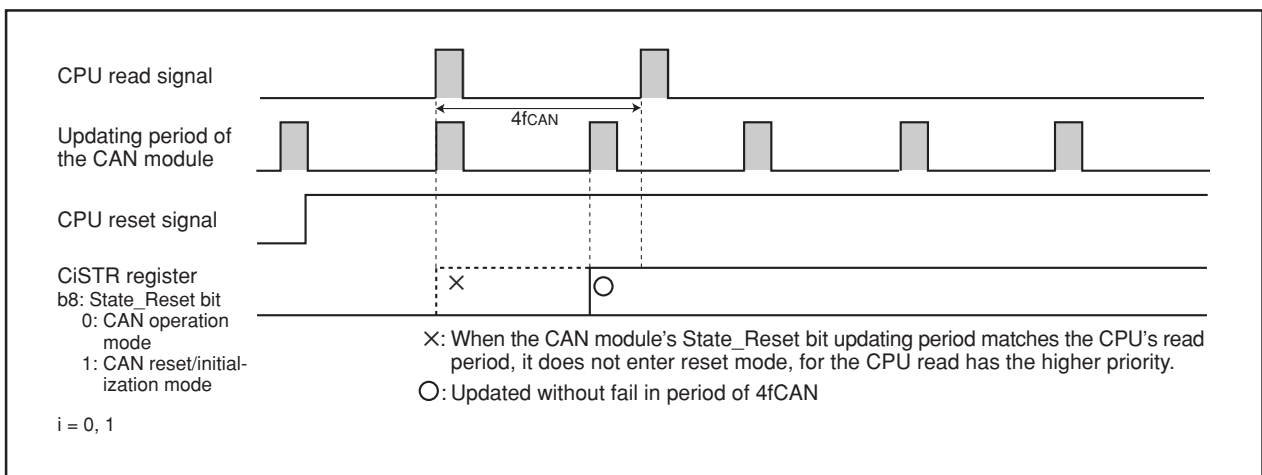


Figure 22.7 When Polling Period of CPU is 3fCAN or Longer

22.13.2 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”.

22.13.3 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 22.4 and 22.5 show recommended pin connections.

Table 22.4 Recommended Pin Connections (In case of PCA82C250: Philips product)

	Standby Mode	High-speed Mode
Rs Pin ⁽¹⁾	“H”	“L”
Power Consumption in CAN Transceiver ⁽²⁾	less than 170 μ A	less than 70 mA
CAN Communication	impossible	possible
Connection		

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. In case of Ta = 25 °C
3. Connect to enabled port to control CAN transceiver.

Table 22.5 Recommended Pin Connections (In case of PCA82C252: Philips product)

	Sleep Mode	Normal Operation Mode
STB Pin ⁽¹⁾	“L”	“H”
EN Pin ⁽¹⁾	“L”	“H”
Power Consumption in CAN Transceiver ⁽²⁾	less than 50 μ A	less than 35 mA
CAN Communication	impossible	possible
Connection		

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. Ta = 25 °C
3. Connect to enabled port to control CAN transceiver.

22.13.4 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 22.6 and 22.7 show pin connections of CAN transceiver.

Table 22.6 Pin Connections of CAN Transceiver (In case of PCA82C250: Philips product)

	Standby Mode	High-speed Mode
Rs Pin ⁽¹⁾	“H”	“L”
CAN Communication	impossible	possible
Connection		

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. Connect to enabled port to control CAN transceiver.

Table 22.7 Pin Connections of CAN Transceiver (In case of PCA82C252: Philips product)

	Sleep Mode	Normal Operation Mode
STB Pin ⁽¹⁾	“L”	“H”
EN Pin ⁽¹⁾	“L”	“H”
CAN Communication	impossible	possible
Connection		

i = 0, 1

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. Connect to enabled port to control CAN transceiver.

22.14 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.

Setting the SM42 bit in the S4C register to “1” causes the P9_6 pin to go to a high-impedance state ⁽¹⁾.

Setting the SM52 bit in the S5C register to “1” causes the P11_2 pin to go to a high-impedance state ⁽²⁾.

Setting the SM62 bit in the S6C register to “1” causes the P11_6 pin to go to a high-impedance state ⁽²⁾.

NOTES:

1. When using SI/O4, set the SM43 bit in the S4C register to “1” (SOUT4 output, CLK4 function) and the port direction bit corresponding for SOUT4 pin to “0” (input mode).
2. The S5C and S6C registers are only in the 128-pin version. When using these registers, set these registers after setting the PU37 bit in the PUR3 register to “1” (Pins P11 to P14 are usable).

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.

When changing the PD14_i bit (i = 0, 1) in the PC14 register from “0” (input port) to “1” (output port), follow the procedures below.

Setting Procedure

- | | | |
|---|---|-------------------------|
| (1) Set P14_i bit | :MOV.B #000000 0 1b, PC14 | ; P14_i bit setting |
| (2) Change PD14_i bit to “1” by MOV instruction | :MOV.B #00 1 100 0 1b, PC14 | ; Change to output port |

22.15 Dedicated Input Pin

When dedicated input pin voltage is larger than VCC pin voltage, latch up occurs.

When different power supplied to the system, and input voltage of unused dedicated input pin is larger than voltage of VCC pin, connect dedicated input pin to VCC via resistor (approximately 1kΩ).

Figure 22.8 shows the circuit connection.

This note is also applicable when VINPUT exceeds VCC during power-up.

The resistor is not necessary when VCC pin voltage is same or larger than dedicated input pin voltage.

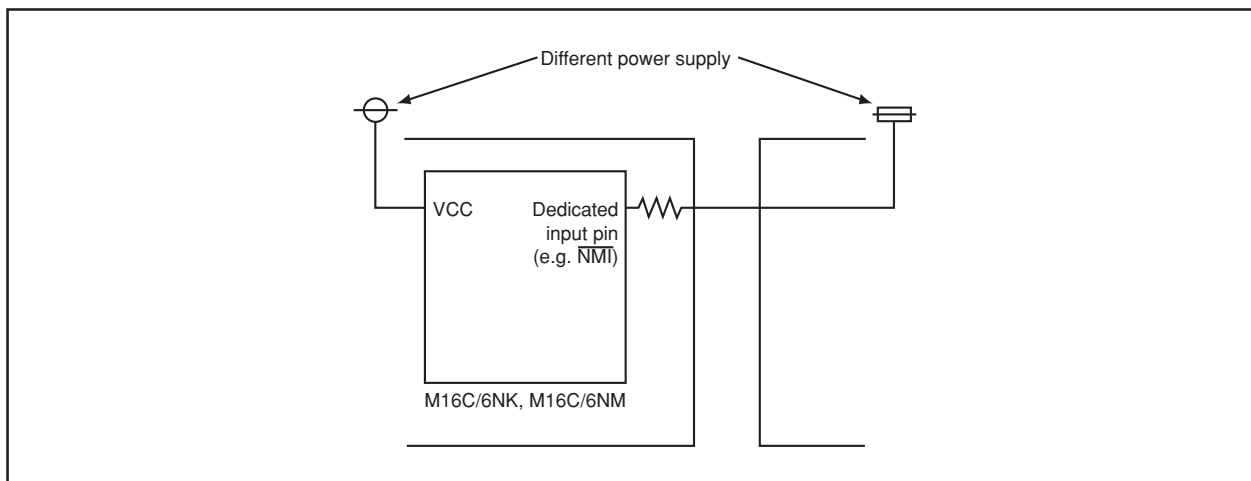


Figure 22.8 Circuit Connection

22.16 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.

22.17 Mask ROM Version

When using the masked ROM version, write nothing to internal ROM area.

22.18 Flash Memory Version

22.18.1 Functions to Prevent Flash Memory from Rewriting

ID codes are stored in addresses 0FFFFDFh, 0FFFFE3h, 0FFFFEBh, 0FFFFEFh, 0FFFFF3h, 0FFFFF7h, and 0FFFFFBh. If wrong data are written to these addresses, the flash memory cannot be read or written in standard serial I/O mode and CAN I/O mode.

The ROMCP register is mapped in address 0FFFFFFh. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.

In the flash memory version of microcomputer, these addresses are allocated to the vector addresses (H) of fixed vectors.

22.18.2 Stop Mode

When entering stop mode, the following settings are required:

- Set the FMR01 bit to “0” (CPU rewrite mode disabled). Disable DMA transfer before setting the CM10 bit to “1” (stop mode).
- Execute the instruction to set the CM10 bit to “1” (stop mode) and then the JMP.B instruction.

```
Example program      BSET      0, CM1      ; Stop mode
                    JMP.B    L1
```

L1:

Program after exiting stop mode

22.18.3 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.

22.18.4 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 software command
- Read lock bit status

22.18.5 Writing Command and Data

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

22.18.6 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.

22.18.7 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.

22.18.8 Operation 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).

22.18.9 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

22.18.10 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 $\overline{\text{NMI}}$ 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 $\overline{\text{NMI}}$ interrupt request is generated. Execute the rewrite program again after exiting the interrupt service routine.

22.18.11 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.

22.18.12 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.

22.18.13 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).

22.19 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.

22.19.1 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 22.9 shows a pin connection example for programming using serial I/O mode.

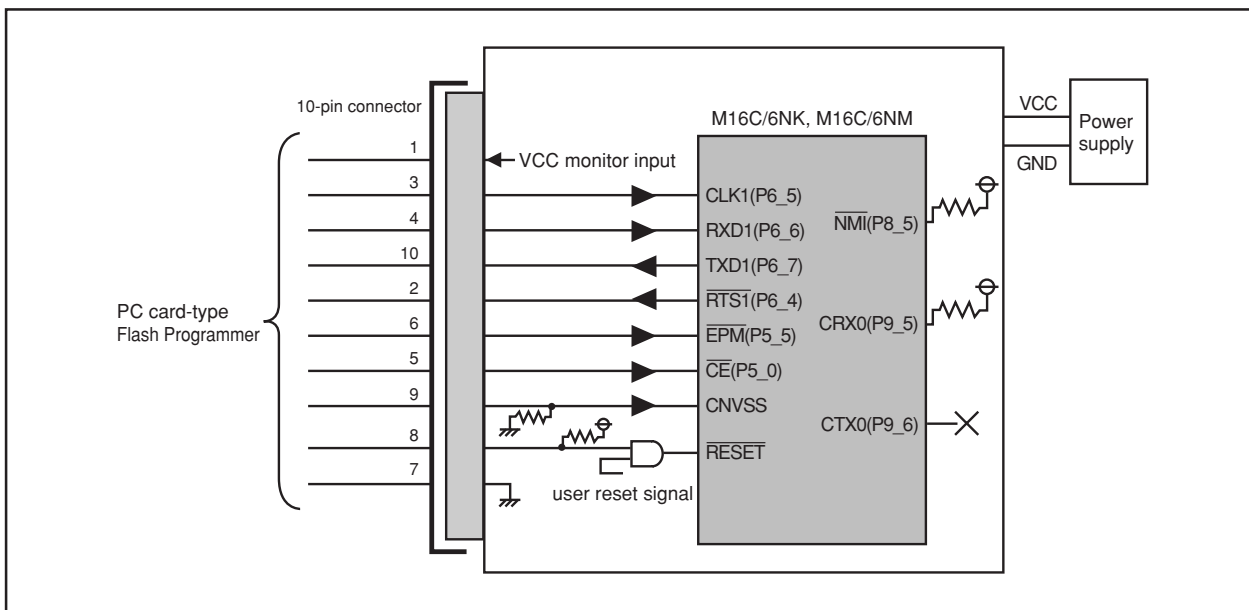


Figure 22.9 Pin Connection for Programming Using Serial I/O Mode

22.19.2 Programming Using CAN I/O Mode

RTS1 pin : This pin automatically outputs “H” and “L” level.

Figure 22.10 shows a pin connection example for programming using CAN I/O mode.

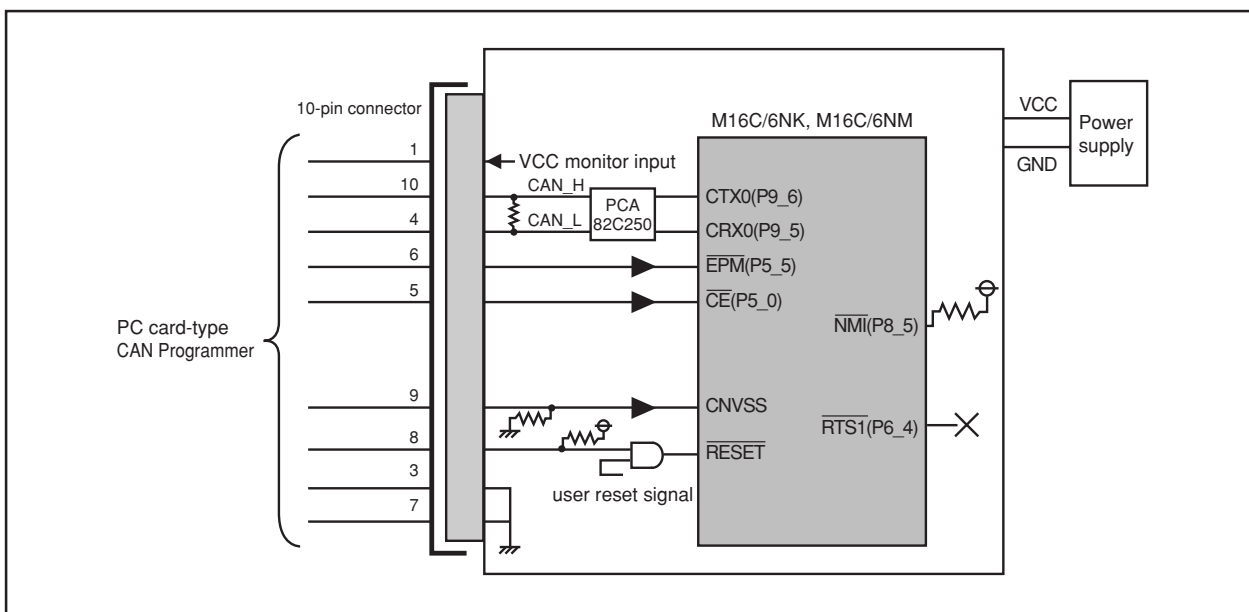


Figure 22.10 Pin Connection for Programming Using CAN I/O Mode

22.20 Noise

Connect a bypass capacitor (approximately 0.1 μF) across the VCC1 and VSS pins, and VCC2 and VSS pins using the shortest and thicker possible wiring. Figure 22.11 shows the bypass capacitor connection.

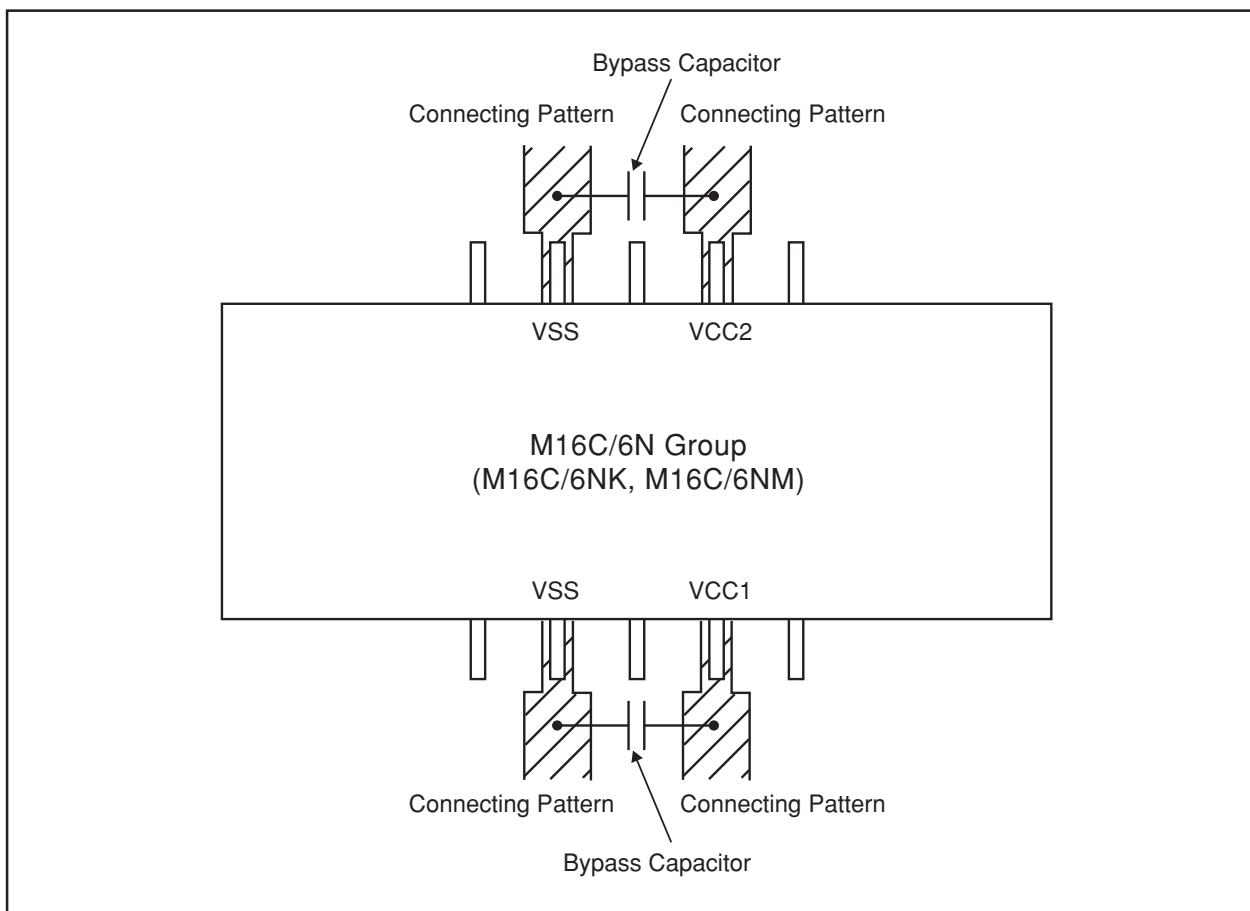
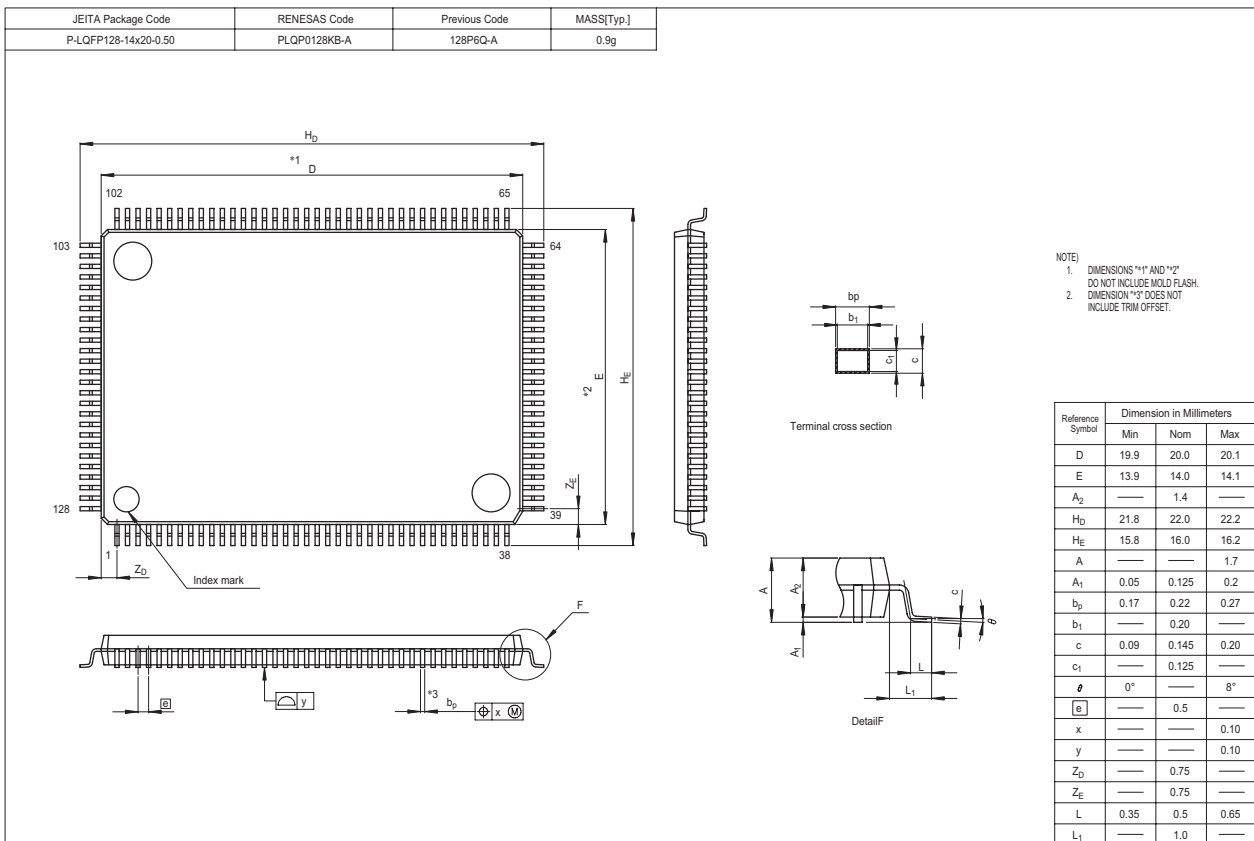
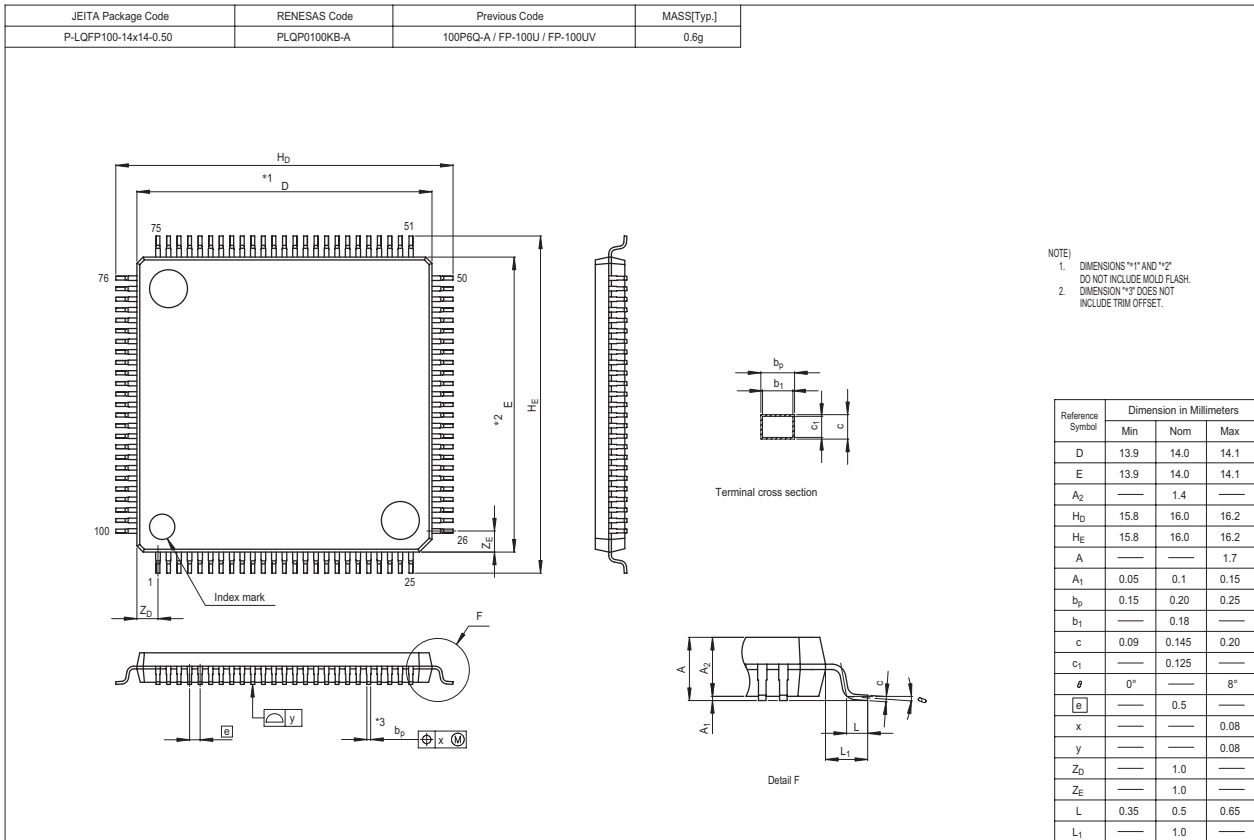


Figure 22.11 Bypass Capacitor Connection

Appendix 1. Package Dimensions



Memo

U0C0 to U2C0	133
U0C1 to U2C1	134
U0MR to U2MR	133
U0RB to U2RB	132
U0SMR to U2SMR	135
U0SMR2 to U2SMR2	136
U0SMR3 to U2SMR3	136
U0SMR4 to U2SMR4	137
U0TB to U2TB	132
UCON	135
UDF	96

W

WDC	81
WDTS	81

REVISION HISTORY

M16C/6N Group (M16C/6NK, M16C/6NM) Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 30, 2004	–	First edition issued
1.01	Nov. 01, 2004	–	Revised edition issued * Revised parts and revised contents are as follows (except for expressional change).
		3	Table 1.2 Performance Outline of M16C/6N Group (128-pin Version: M16C/6NM) • Interrupt: Internal interrupt source is revised from “32 sources” to “34 sources”.
		270	Table 21.2 Recommended Operating Conditions (1) • I _{OH(peak)} : Unit is revised from “V” to “mA”.
		271	Table 21.3 Recommended Operating Conditions (2) • NOTE 3: “VCC = 3.0 ± 0.3 V” is revised to “VCC = 3.3 ± 0.3 V”.
		291	22.9.1.2 Timer A (Event Counter Mode) is revised.
1.10	Jul. 01, 2005	–	Revised edition issued * The contents of product are revised. (T/V-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 (100-pin Version: M16C/6NM) • Performance outline of T/V-ver. is added.
		3	Table 1.2 Performance outline of M16C/6N Group (128-pin Version: M16C/6NN) • Performance outline of T/V-ver. is added.
		5	Table 1.3 Product List is revised. (T/V-ver. is added.) Figure 1.2 Type No., Memory Size, and Package: “Characteristics” is added.
		13	Figure 4.1 SFR Information (1): The value of After Reset in CM2 Register is revised.
		19	Figure 4.7 SFR Information (7): NOTE 1 is revised.
		39	Figure 7.4 CM2 Register: The value of After Reset is revised.
		55	Figure 7.13 State Transition in Normal Operation Mode: NOTE 7 is revised.
		78	9.10 Address Match Interrupt: After of 13th line • “Note that when using the external bus in 8-bit width, no address match interrupts can be used for external areas.” is deleted.
		176	Figure 14.37 (upper) SiC Register: NOTE 4 is revised.
		207	Figure 18.6 C0MCTLj and C1MCTLj Registers • RemActive bit: Function is revised. • RspLock bit: Bit Name is revised. • NOTE 2 is revised.
		208	Figure 18.7 C0CTLR and C1CTLR Registers (upper) • LoopBack bit: The expression of Function is revised. • BasicCAN bit: The expression of Function is revised. Figure 18.7 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.
		209	Figure 18.8 C0STR and C1STR Registers (upper): NOTE 1 is deleted. Figure 18.8 C0STR and C1STR Registers (lower) • State_LoopBack bit: The expression of Function is revised. • State_BasicCAN bit: The expression of Function is revised.

REVISION HISTORY

M16C/6N Group (M16C/6NK, M16C/6NM) Hardware Manual

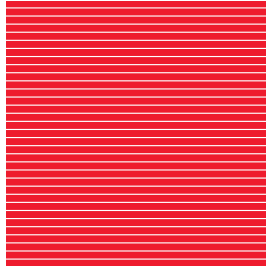
Rev.	Date	Description	
		Page	Summary
1.10	Jul. 01, 2005	212	Figure 18.11 C0RECR, C1RECR Registers, C0TECR, C1TECR Registers, C0TSR, C1TSR Registers, and C0AFS, C1AFS Registers <ul style="list-style-type: none"> • C0RECR, C1RECR Registers: NOTE 2 is deleted. • C0TECR, C1TECR Registers: NOTE 1 is deleted. • C0TSR, C1TSR Registers: NOTE 1 is deleted.
		223	18.15.1 Reception (1): "(refer to 18.15.2 Transmission)" is deleted.
		228	Figure 19.1 I/O Ports (1): "P7_0" in 4th figure is deleted.
		230	Figure 19.3 I/O Ports (3): "P7_0" is added to middle figure.
		232	Figure 19.6 I/O Pins: NOTE 1 is deleted.
		272	Table 21.4 Electrical Characteristics (1) <ul style="list-style-type: none"> • Measuring Condition of V_{OL} is revised from "$L_{OL} = -200\mu A$" to "$L_{OL} = 200\mu A$".
		273	Table 21.5 Electrical Characteristics (2): Mask ROM (5th item) <ul style="list-style-type: none"> • "f(XCIN)" is changed to "f(BCLK)".
		274	Table 21.6 A/D Conversion Characteristics: "Tolerance Level Impedance" is deleted.
		307	22.14 Programmable I/O Ports: last 1 to 2 lines <ul style="list-style-type: none"> • (1) Setting Procedure is revised from "#00010000b" to "#00000001b". • (2) Setting Procedure is revised from "#00010011b" to "#00110001b".

M16C/6N Group (M16C/6NK, M16C/6NM) Hardware Manual

**Publication Data : Rev.1.00 Sep 30, 2004
Rev.1.10 Jul 01, 2005**

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

M16C/6N Group (M16C/6NK, M16C/6NM) Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan