REJ09B0007-0100Z

Everywhere you imagine. RENESAS

16

M16C/1N Group

Hardware Manual

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER

M16C FAMILY / M16C/10 SERIES

Hardware Manual

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

Rev. 1.00
Revision date: Oct 20, 2004

RenesasTechnology
www.renesas.com

Keep safety first in your circuit designs!

1. 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 nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. 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.
2. 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.
3. 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).
4. 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.
5. 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.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. 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.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.

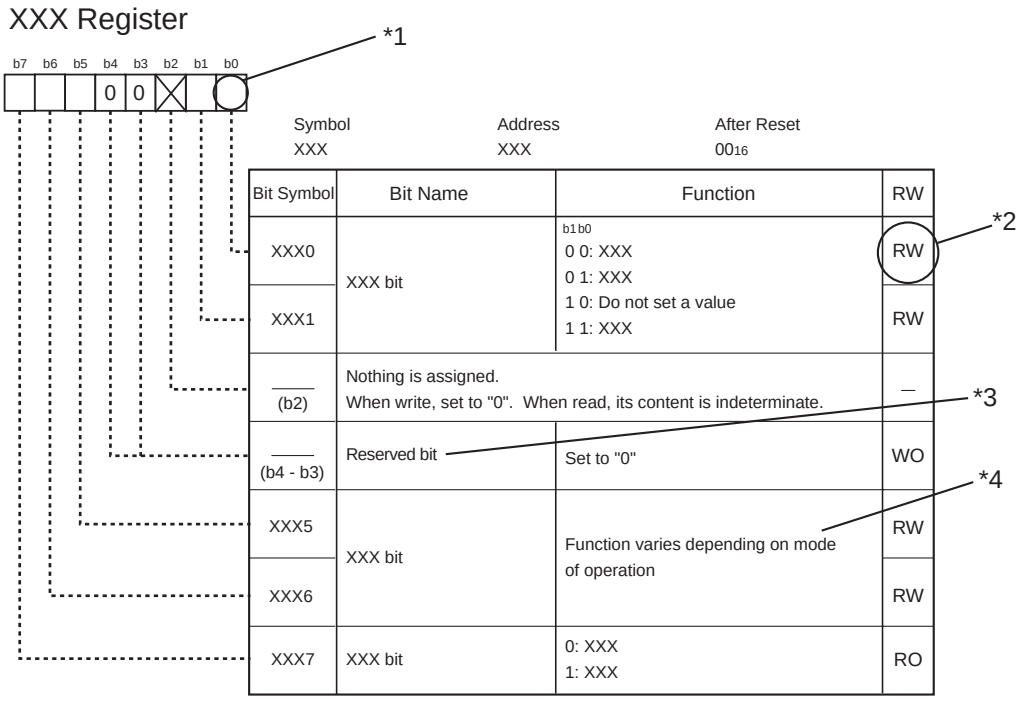
How to Use This Manual

1. Introduction

This hardware manual provides detailed information on the M16C/1N Group 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 a 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.

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	• Application examples of peripheral functions • Sample programs • Introduction to the basic functions in the M16C family • Programming method with Assembly and C languages
Technical Update	Preliminary report about the specification of a product, a document, etc.

NOTES :

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

Table of Contents

M16C/1N Group Usage Note Reference Book

For the most current Usage Note Reference Book, please visit our website.

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.

Quick Reference to Pages Classified by Address

Address	Register	Symbol	Page
0000_{16}			
0001_{16}			
0002_{16}			
0003_{16}			
0004_{16}	Processor mode register 0	PM0	22, 41
0005_{16}	Processor mode register 1	PM1	22, 41, 69
0006_{16}	System clock control register 0	CM0	25
0007_{16}	System clock control register 1	CM1	25
0008_{16}			
0009_{16}	Address match interrupt enable register	AIER	65
$000A_{16}$	Protect register	PRCR	40
$000B_{16}$			
$000C_{16}$	Oscillation stop detection register	CM2	26, 37
$000D_{16}$			
$000E_{16}$	Watchdog timer start register	WDTS	69
$000F_{16}$	Watchdog timer control register	WDC	69
0010_{16}	Address match interrupt register 0	RMAD0	65
0011_{16}			
0012_{16}			
0013_{16}			
0014_{16}	Address match interrupt register 1	RMAD1	65
0015_{16}			
0016_{16}			
0017_{16}			
0018_{16}			
0019_{16}			
$001A_{16}$			
$001B_{16}$			
$001C_{16}$			
$001D_{16}$			
$001E_{16}$	$\overline{\text{INT0}}$ input filter select register	INT0F	60, 94
$001F_{16}$			
0020_{16}			
0021_{16}			
0022_{16}			
0023_{16}			
0024_{16}			
0025_{16}			
0026_{16}			
0027_{16}			
0028_{16}			
0029_{16}			
$002A_{16}$			
$002B_{16}$			
$002C_{16}$			
$002D_{16}$			
$002E_{16}$			
$002F_{16}$			
0030_{16}			
0031_{16}			
0032_{16}			
0033_{16}			
0034_{16}			
0035_{16}			
0036_{16}			
0037_{16}			
0038_{16}			
0039_{16}			
$003A_{16}$			
$003B_{16}$			
$003C_{16}$			
$003D_{16}$			
$003E_{16}$			
$003F_{16}$			

Address	Register	Symbol	Page
0040_{16}			
0041_{16}			
0042_{16}			
0043_{16}			
0044_{16}			
0045_{16}	CAN0 wake up interrupt control register	C01WKIC	51
0046_{16}	CAN0 error interrupt control register	C01ERRIC	51
0047_{16}			
0048_{16}	CAN0 successful reception interrupt control register	C0RECIC	51
0049_{16}	CAN0 successful transmission interrupt control register	C0TRMIC	51
$004A_{16}$			
$004B_{16}$			
$004C_{16}$			
$004D_{16}$	Key input interrupt control register	KUPIC	51
$004E_{16}$	A/D conversion interrupt control register	ADIC	51
$004F_{16}$			
0050_{16}			
0051_{16}	UART0 transmit interrupt control register	S0TIC	51
0052_{16}	UART0 receive interrupt control register	S0RIC	
0053_{16}	UART1 transmit interrupt control register	S1TIC	
0054_{16}	UART1 receive interrupt control register	S1RIC	
0055_{16}	Timer 1 interrupt control register	T1IC	
0056_{16}	Timer X interrupt control register	TXIC	
0057_{16}	Timer Y interrupt control register	TYIC	
0058_{16}	Timer Z interrupt control register	TZIC	
0059_{16}	CNTR0 interrupt control register	CNTR0IC	
$005A_{16}$	TCIN interrupt control register	TCINIC	
$005B_{16}$	Timer C interrupt control register	TCIC	
$005C_{16}$	INT3 interrupt control register	INT3IC	
$005D_{16}$	INT0 interrupt control register	INT0IC	
$005E_{16}$	INT1 interrupt control register	INT1IC	
$005F_{16}$	INT2 interrupt control register	INT2IC	
0060_{16}			
0061_{16}			
0062_{16}			
0063_{16}			
0064_{16}			
0065_{16}			
0066_{16}			
0067_{16}			
0068_{16}			
0069_{16}			
$006A_{16}$			
$006B_{16}$			
$006C_{16}$			
$006D_{16}$			
$006E_{16}$			
$006F_{16}$			
0070_{16}			
0071_{16}			
0072_{16}			
0073_{16}			
0074_{16}			
0075_{16}			
0076_{16}			
0077_{16}			
0078_{16}			
0079_{16}			
$007A_{16}$			
$007B_{16}$			
$007C_{16}$			
$007D_{16}$			
$007E_{16}$			
$007F_{16}$			

Note 1: The blank areas are reserved.

Address	Register	Symbol	Page
0080_{16}	Timer Y, Z mode register	TYZMR	82, 86, 88, 91, 96, 98, 100, 103
0081_{16}	Prescaler Y	PREY	83
0082_{16}	Timer Y secondary	TYSC	
0083_{16}	Timer Y Primary	TYPR	
0084_{16}	Timer Y, Z waveform output control register	PUM	84, 86, 88, 93, 96, 98, 100, 103
0085_{16}	Prescaler Z	PREZ	92
0086_{16}	Timer Z secondary	TZSC	
0087_{16}	Timer Z Primary	TZPR	
0088_{16}	Prescaler 1	PRE1	72
0089_{16}	Timer 1	T1	
$008A_{16}$	Timer Y, Z output control register	TYZOC	83, 94
$008B_{16}$	Timer X mode register	TXMR	62, 73, 75-78, 80
$008C_{16}$	Prescaler X	PREX	74
$008D_{16}$	Timer X	TX	
$008E_{16}$	Timer count source setting register	TCSS	72, 74, 84, 93
$008F_{16}$	Clock prescaler reset flag	CPSRF	26
0090_{16}	Timer C	TC	106
0091_{16}			
0092_{16}			
0093_{16}			
0094_{16}			
0095_{16}			
0096_{16}	External input enable register	INTEN	60, 94
0097_{16}			
0098_{16}	Key input enable register	KIEN	64
0099_{16}			
$009A_{16}$	Timer C control register 0	TCC0	63, 106
$009B_{16}$	Timer C control register 1	TCC1	
$009C_{16}$	Timer measurement register	TM	106
$009D_{16}$			
$009E_{16}$			
$009F_{16}$			
$00A0_{16}$	UART0 transmit/receve mode register	U0MR	111, 114, 119
$00A1_{16}$	UART0 bit rate generator	U0BRG	110
$00A2_{16}$	UART0 transmit buffer register	U0TB	
$00A3_{16}$			
$00A4_{16}$	UART0 transmit/receive control register 0	U0C0	111
$00A5_{16}$	UART0 transmit/receive control register 1	U0C1	112
$00A6_{16}$	UART0 receive buffer register	U0RB	110
$00A7_{16}$			
$00A8_{16}$	UART1 transmit/receive mode register	U1MR	111, 114, 119
$00A9_{16}$	UART1 bit rate generator	U1BRG	110
$00AA_{16}$	UART1 transmit buffer register	U1TB	
$00AB_{16}$			
$00AC_{16}$	UART1 transmit/receive control register 0	U1C0	111
$00AD_{16}$	UART1 transmit/receive control register 1	U1C1	112
$00AE_{16}$	UART1 receive buffer register	U1RB	110
$00AF_{16}$			
$00B0_{16}$	UART transmit/receive control register 2	UCON	112
$00B1_{16}$			
$00B2_{16}$			
$00B3_{16}$			
$00B4_{16}$			
$00B5_{16}$			
$00B6_{16}$			
$00B7_{16}$			
$00B8_{16}$			
$00B9_{16}$			
$00BA_{16}$			
$00BB_{16}$			
$00BC_{16}$			
$00BD_{16}$			
$00BE_{16}$			
$00BF_{16}$			

Address	Register	Symbol	Page
$00C0_{16}$	A/D register	AD	125
$00C1_{16}$			
$00C2_{16}$			
$00C3_{16}$			
$00C4_{16}$			
$00C5_{16}$			
$00C6_{16}$			
$00C7_{16}$			
$00C8_{16}$			
$00C9_{16}$			
$00CA_{16}$			
$00CB_{16}$			
$00CC_{16}$			
$00CD_{16}$			
$00CE_{16}$			
$00CF_{16}$			
$00D0_{16}$			
$00D1_{16}$			
$00D2_{16}$			
$00D3_{16}$			
$00D4_{16}$	A/D control register 2	ADCON2	125
$00D5_{16}$			
$00D6_{16}$	A/D control register 0	ADCON0	124, 126, 127
$00D7_{16}$	A/D control register 1	ADCON1	
$00D8_{16}$	D/A register	DA	130
$00D9_{16}$			
$00DA_{16}$			
$00DB_{16}$			
$00DC_{16}$	D/A control register	DACON	130
$00DD_{16}$			
$00DE_{16}$			
$00DF_{16}$			
$00E0_{16}$	Port P0 register	P0	160
$00E1_{16}$	Port P1 register	P1	
$00E2_{16}$	Port P0 direction register	PD0	
$00E3_{16}$	Port P1 direction register	PD1	
$00E4_{16}$	Port P2 register	P2	
$00E5_{16}$	Port P3 register	P3	
$00E6_{16}$	Port P2 direction register	PD2	
$00E7_{16}$	Port P3 direction register	PD3	
$00E8_{16}$	Port P4 register	P4	
$00E9_{16}$	Port P5 register	P5	
$00EA_{16}$	Port P4 direction register	PD4	
$00EB_{16}$	Port P5 direction register	PD5	
$00EC_{16}$			
$00ED_{16}$			
$00EE_{16}$			
$00EF_{16}$			
$00F0_{16}$			
$00F1_{16}$			
$00F2_{16}$			
$00F3_{16}$			
$00F4_{16}$			
$00F5_{16}$			
$00F6_{16}$			
$00F7_{16}$			
$00F8_{16}$	CAN0 I/O pin select register	CIOSR	162
$00F9_{16}$			
$00FA_{16}$			
$00FB_{16}$			
$00FC_{16}$	Pull-up control register 0	PUR0	161
$00FD_{16}$	Pull-up control register 1	PUR1	
$00FE_{16}$	Port P1 drive capacity control register	DRR	
$00FF_{16}$			

Note 1: The blank areas are reserved.

Address	Register	symbol	Page
0100₁₆			
0101₁₆			
0102₁₆			
0103₁₆			
0104₁₆ ≈ 01B0₁₆			
01B1₁₆			
01B2₁₆			
01B3₁₆	Flash memory control register 4	FMR4	182
01B4₁₆			
01B5₁₆	Flash memory control register 1	FMR1	182
01B6₁₆			
01B7₁₆	Flash memory control register 0	FMR0	181
01B8₁₆			
01B9₁₆			
01BA₁₆			
01BB₁₆			
01BC₁₆			
01BD₁₆			
01BE₁₆			
01BF₁₆ ≈ 0215₁₆			
0216₁₆			
0217₁₆			
0218₁₆			
0219₁₆			
021A₁₆			
021B₁₆			
021C₁₆			
021D₁₆			
021E₁₆			
021F₁₆			
0220₁₆	CAN0 message control register 0	C0MCTL0	136
0221₁₆	CAN0 message control register 1	C0MCTL1	
0222₁₆	CAN0 message control register 2	C0MCTL2	
0223₁₆	CAN0 message control register 3	C0MCTL3	
0224₁₆	CAN0 message control register 4	C0MCTL4	
0225₁₆	CAN0 message control register 5	C0MCTL5	
0226₁₆	CAN0 message control register 6	C0MCTL6	
0227₁₆	CAN0 message control register 7	C0MCTL7	
0228₁₆	CAN0 message control register 8	C0MCTL8	
0229₁₆	CAN0 message control register 9	C0MCTL9	
022A₁₆	CAN0 message control register 10	C0MCTL10	
022B₁₆	CAN0 message control register 11	C0MCTL11	
022C₁₆	CAN0 message control register 12	C0MCTL12	
022D₁₆	CAN0 message control register 13	C0MCTL13	
022E₁₆	CAN0 message control register 14	C0MCTL14	
022F₁₆	CAN0 message control register 15	C0MCTL15	
0230₁₆ 0231₁₆	CAN0 control register	C0CTLR	137
0232₁₆ 0233₁₆	CAN0 status register	C0STR	138
0234₁₆ 0235₁₆	CAN0 slot status register	C0SSTR	139
0236₁₆ 0237₁₆	CAN0 interrupt control register	C0ICR	140
0238₁₆ 0239₁₆	CAN0 extended ID register	C0IDR	140
023A₁₆ 023B₁₆	CAN0 configuration register	C0CONR	141
023C₁₆	CAN0 reception error count register	C0RECR	142
023D₁₆	CAN0 transmission error count register	C0TECR	
023E₁₆			
023F₁₆			

Address	Register	Symbol	Page
0240₁₆			
0241₁₆			
0242₁₆			
0243₁₆			
0244₁₆ 0245₁₆	CAN0 acceptance filer support register	C0AFS	143
0246₁₆			
0247₁₆			
0248₁₆			
0249₁₆			
024A₁₆			
024B₁₆			
024C₁₆			
024D₁₆			
024E₁₆			
024F₁₆			
0250₁₆			
0251₁₆			
0252₁₆			
0253₁₆			
0254₁₆			
0255₁₆			
0256₁₆			
0257₁₆			
0258₁₆			
0259₁₆			
025A₁₆			
025B₁₆			
025C₁₆			
025D₁₆			
025E₁₆			
025F₁₆	CAN0 clock select register	CCLKR	27
0260₁₆ 0261₁₆ 0262₁₆ 0263₁₆ 0264₁₆ 0265₁₆	CAN0 message box 0: Identifier/DLC		133 134
0266₁₆ 0267₁₆ 0268₁₆ 0269₁₆ 026A₁₆ 026B₁₆ 026C₁₆ 026D₁₆	CAN0 message box 0: Data field		
026E₁₆ 026F₁₆	CAN0 message box 0: Time stamp		
0270₁₆ 0271₁₆ 0272₁₆ 0273₁₆ 0274₁₆ 0275₁₆	CAN0 message box 1: Identifier/DLC		
0276₁₆ 0277₁₆ 0278₁₆ 0279₁₆ 027A₁₆ 027B₁₆ 027C₁₆ 027D₁₆	CAN0 message box 1: Data field		
027E₁₆ 027F₁₆	CAN0 message box 1: Time stamp		

Note 1: The blank areas are reserved.

Address	Register	Symbol	Page
0280_{16}	CAN0 message box 2: Identifier/DLC		133 134
0281_{16}			
0282_{16}			
0283_{16}			
0284_{16}			
0285_{16}			
0286_{16}	CAN0 message box 2: Data field		
0287_{16}			
0288_{16}			
0289_{16}			
$028A_{16}$			
$028B_{16}$			
$028C_{16}$			
$028D_{16}$			
$028E_{16}$	CAN0 message box 2: Time stamp		
$028F_{16}$			
0290_{16}	CAN0 message box 3: Identifier/DLC		
0291_{16}			
0292_{16}			
0293_{16}			
0294_{16}			
0295_{16}			
0296_{16}	CAN0 message box 3: Data field		
0297_{16}			
0298_{16}			
0299_{16}			
$029A_{16}$			
$029B_{16}$			
$029C_{16}$			
$029D_{16}$			
$029E_{16}$	CAN0 message box 3: Time stamp		
$029F_{16}$			
$02A0_{16}$	CAN0 message box 4: Identifier/DLC		
$02A1_{16}$			
$02A2_{16}$			
$02A3_{16}$			
$02A4_{16}$			
$02A5_{16}$			
$02A6_{16}$	CAN0 message box 4: Data field		
$02A7_{16}$			
$02A8_{16}$			
$02A9_{16}$			
$02AA_{16}$			
$02AB_{16}$			
$02AC_{16}$			
$02AD_{16}$			
$02AE_{16}$	CAN0 message box 4: Time stamp		
$02AF_{16}$			
$02B0_{16}$	CAN0 message box 5: Identifier/DLC		
$02B1_{16}$			
$02B2_{16}$			
$02B3_{16}$			
$02B4_{16}$			
$02B5_{16}$			
$02B6_{16}$	CAN0 message box 5: Data field		
$02B7_{16}$			
$02B8_{16}$			
$02B9_{16}$			
$02BA_{16}$			
$02BB_{16}$			
$02BC_{16}$			
$02BD_{16}$			
$02BE_{16}$	CAN0 message box 5: Time stamp		
$02BF_{16}$			

Address	Register	Symbol	Page
$02C0_{16}$	CAN0 message box 6: Identifier/DLC		
$02C1_{16}$			
$02C2_{16}$			
$02C3_{16}$			
$02C4_{16}$			
$02C5_{16}$			
$02C6_{16}$	CAN0 message box 6: Data field		
$02C7_{16}$			
$02C8_{16}$			
$02C9_{16}$			
$02CA_{16}$			
$02CB_{16}$			
$02CC_{16}$			
$02CD_{16}$			
$02CE_{16}$	CAN0 message box 6: Time stamp		
$02CF_{16}$			
$02D0_{16}$	CAN0 message box 7: Identifier/DLC		
$02D1_{16}$			
$02D2_{16}$			
$02D3_{16}$			
$02D4_{16}$			
$02D5_{16}$			
$02D6_{16}$	CAN0 message box 7: Data field		
$02D7_{16}$			
$02D8_{16}$			
$02D9_{16}$			
$02DA_{16}$			
$02DB_{16}$			
$02DC_{16}$			
$02DD_{16}$			
$02DE_{16}$	CAN0 message box 7: Time stamp		
$02DF_{16}$			
$02E0_{16}$	CAN0 message box 8: Identifier/DLC		
$02E1_{16}$			
$02E2_{16}$			
$02E3_{16}$			
$02E4_{16}$			
$02E5_{16}$			
$02E6_{16}$	CAN0 message box 8: Data field		
$02E7_{16}$			
$02E8_{16}$			
$02E9_{16}$			
$02EA_{16}$			
$02EB_{16}$			
$02EC_{16}$			
$02ED_{16}$			
$02EE_{16}$	CAN0 message box 8: Time stamp		
$02EF_{16}$			
$02F0_{16}$	CAN0 message box 9: Identifier/DLC		
$02F1_{16}$			
$02F2_{16}$			
$02F3_{16}$			
$02F4_{16}$			
$02F5_{16}$			
$02F6_{16}$	CAN0 message box 9: Data field		
$02F7_{16}$			
$02F8_{16}$			
$02F9_{16}$			
$02FA_{16}$			
$02FB_{16}$			
$02FC_{16}$			
$02FD_{16}$			
$02FE_{16}$	CAN0 message box 9: Time stamp		
$02FF_{16}$			

Note 1: The blank areas are reserved.

Address	Register	Symbol	Page
0300_{16}	CAN0 message box 10: Identifier/DLC		
0301_{16}			
0302_{16}			
0303_{16}			
0304_{16}			
0305_{16}			
0306_{16}	CAN0 message box 10: Data field		
0307_{16}			
0308_{16}			
0309_{16}			
$030A_{16}$			
$030B_{16}$			
$030C_{16}$			
$030D_{16}$			
$030E_{16}$	CAN0 message box 10: Time stamp		
$030F_{16}$			
0310_{16}	CAN0 message box 11: Identifier/DLC		
0311_{16}			
0312_{16}			
0313_{16}			
0314_{16}			
0315_{16}			
0316_{16}	CAN0 message box 11: Data field		
0317_{16}			
0318_{16}			
0319_{16}			
$031A_{16}$			
$031B_{16}$			
$031C_{16}$			
$031D_{16}$			
$031E_{16}$	CAN0 message box 11: Time stamp		133
$031F_{16}$			134
0320_{16}	CAN0 message box 12: Identifier/DLC		
0321_{16}			
0322_{16}			
0323_{16}			
0324_{16}			
0325_{16}			
0326_{16}	CAN0 message box 12: Data field		
0327_{16}			
0328_{16}			
0329_{16}			
$032A_{16}$			
$032B_{16}$			
$032C_{16}$			
$032D_{16}$			
$032E_{16}$	CAN0 message box 12: Time stamp		
$032F_{16}$			
0330_{16}	CAN0 message box 13: Identifier/DLC		
0331_{16}			
0332_{16}			
0333_{16}			
0334_{16}			
0335_{16}			
0336_{16}	CAN0 message box 13: Data field		
0337_{16}			
0338_{16}			
0339_{16}			
$033A_{16}$			
$033B_{16}$			
$033C_{16}$			
$033D_{16}$			
$033E_{16}$	CAN0 message box 13: Time stamp		
$033F_{16}$			

Address	Register	Symbol	Page
0340_{16}	CAN0 message box 14: Identifier/DLC		
0341_{16}			
0342_{16}			
0343_{16}			
0344_{16}			
0345_{16}			
0346_{16}	CAN0 message box 14: Data field		
0347_{16}			
0348_{16}			
0349_{16}			
$034A_{16}$			
$034B_{16}$			
$034C_{16}$			
$034D_{16}$			
$034E_{16}$	CAN0 message box 14: Time stamp		
$034F_{16}$			133
0350_{16}	CAN0 message box 15: Identifier/DLC		134
0351_{16}			
0352_{16}			
0353_{16}			
0354_{16}			
0355_{16}			
0356_{16}	CAN0 message box 15: Data field		
0357_{16}			
0358_{16}			
0359_{16}			
$035A_{16}$			
$035B_{16}$			
$035C_{16}$			
$035D_{16}$			
$035E_{16}$	CAN0 message box 15: Time stamp		
$035F_{16}$			
0360_{16}	CAN0 global mask register		
0361_{16}			
0362_{16}			
0363_{16}			
0364_{16}			
0365_{16}			
0366_{16}	CAN0 local mask A register		135
0367_{16}			
0368_{16}			
0369_{16}			
$036A_{16}$			
$036B_{16}$			
$036C_{16}$	CAN0 local mask B register		
$036D_{16}$			
$036E_{16}$			
$036F_{16}$			
0370_{16}			
0371_{16}			
≈			≈
$03B4_{16}$			
$03B5_{16}$			
$03B6_{16}$			
$03B7_{16}$			
$03B8_{16}$			
$03B9_{16}$			
≈			≈
$03FA_{16}$			
$03FB_{16}$			
$03FC_{16}$			
$03FD_{16}$			
$03FE_{16}$			
$03FF_{16}$			

Note 1: The blank areas are reserved.

1. Overview

The M16C/1N group consists of single-chip microcomputers that use high-performance silicon gate CMOS processes and have a on-chip M16C/60 series CPU core. The microcomputers are housed in 48-pin plastic mold QFP package. These single-chip microcomputers have both high function instructions and high instruction efficiency and feature a one-megabyte address space and the capability to execute instructions at high speed.

1.1 Applications

Automotive and industrial control systems, other automobile, other

1.2 Performance Overview

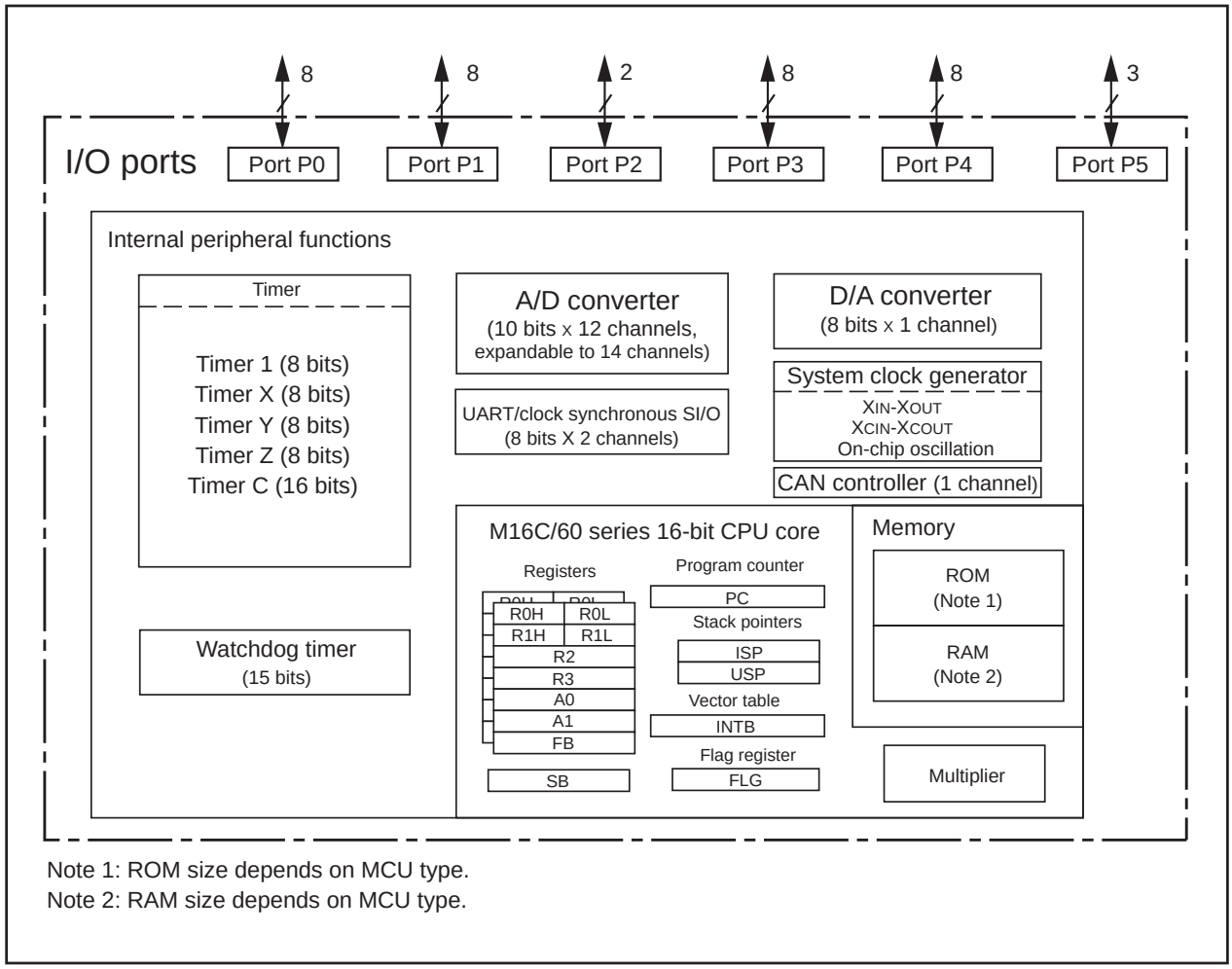
Table 1.1 gives an overview of the M16C/1N group performance specification.

Table 1.1 Performance overview

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		62.5 ns (when f(XIN)=16MHz)
Memory size	ROM	See **Table 1.2 Performance overview**
	RAM	See **Table 1.2 Performance overview**
I/O port		P0 to P5: 37 lines
Multifunction timer	T1	8 bits x 1
	TX, TY, TZ	8 bits x 3
	TC	16 bits x 1
Serial I/O (UART or clock synchronous)		x 2
A/D converter (maximum resolution: 10 bits)		x 12 channels (Expandable up to 14 channels)
D/A converter		8 bits x 1
CAN controller		1 channel, 2.0B active
Watchdog timer		15 bits x 1 (with prescaler)
Interrupts		15 internal causes, 8 external causes, 4 software causes
Clock generating circuits		3 internal circuits
Power supply voltage		4.2 V to 5.5V (when f(XIN)=16MHz)
Power consumption		70mW(VCC=5.0V, f(XIN)=16MHz)
I/O characteristics	I/O withstand voltage	5V
	Output current	5mA (10mA:LED drive port)
Device configuration		CMOS silicon gate
Package		48-pin LQFP

1.3 Block Diagram

Figure 1.1 shows block diagram of the M16C/1N group.



Note 1: ROM size depends on MCU type.
Note 2: RAM size depends on MCU type.

Figure 1.1 Block diagram

1.4 Performance Overview

Table 1.2 shows performance overview.

Table 1.2 Performance overview

As of June 2004

Type No.	ROM	RAM	Package	Remarks
M301N2M4T-XXXFP(D)	32Kbytes	1Kbytes	48P6Q-A	Mask ROM
M301N2M8T-XXXFP(D)	64Kbytes	3Kbytes		
M301N2F8TFP(D)				Flash memory
M301N2F8FP(D)				

(D): Under development

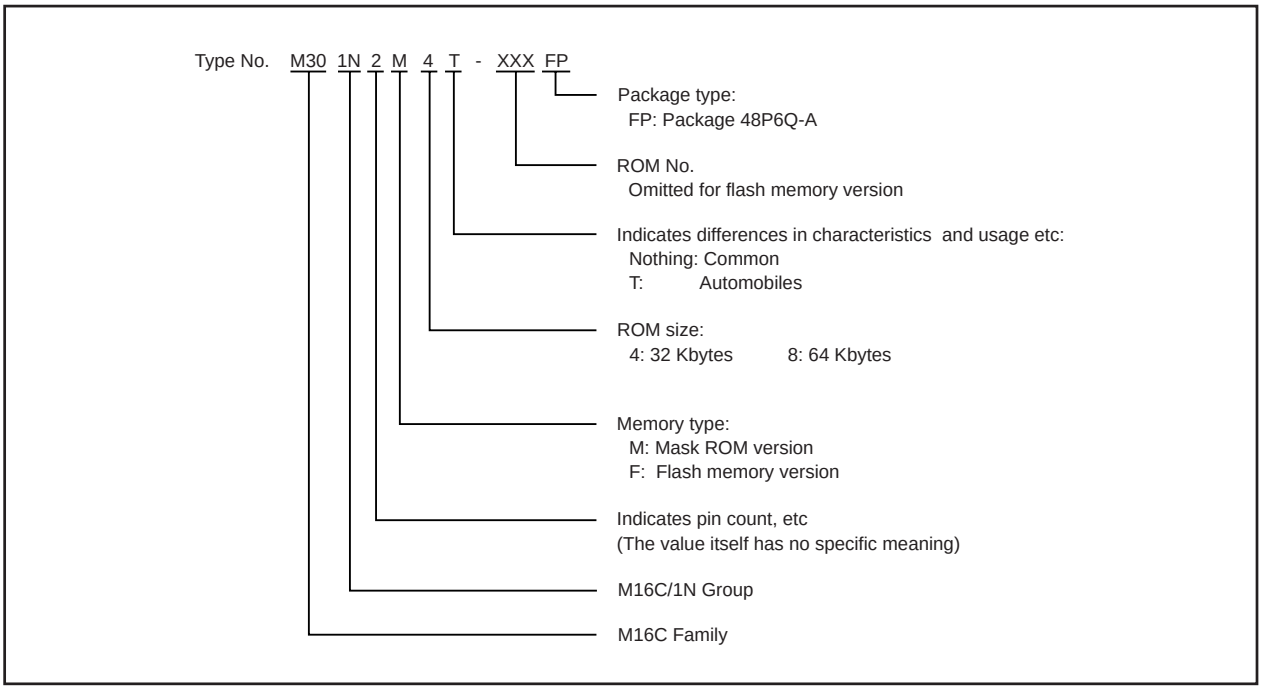
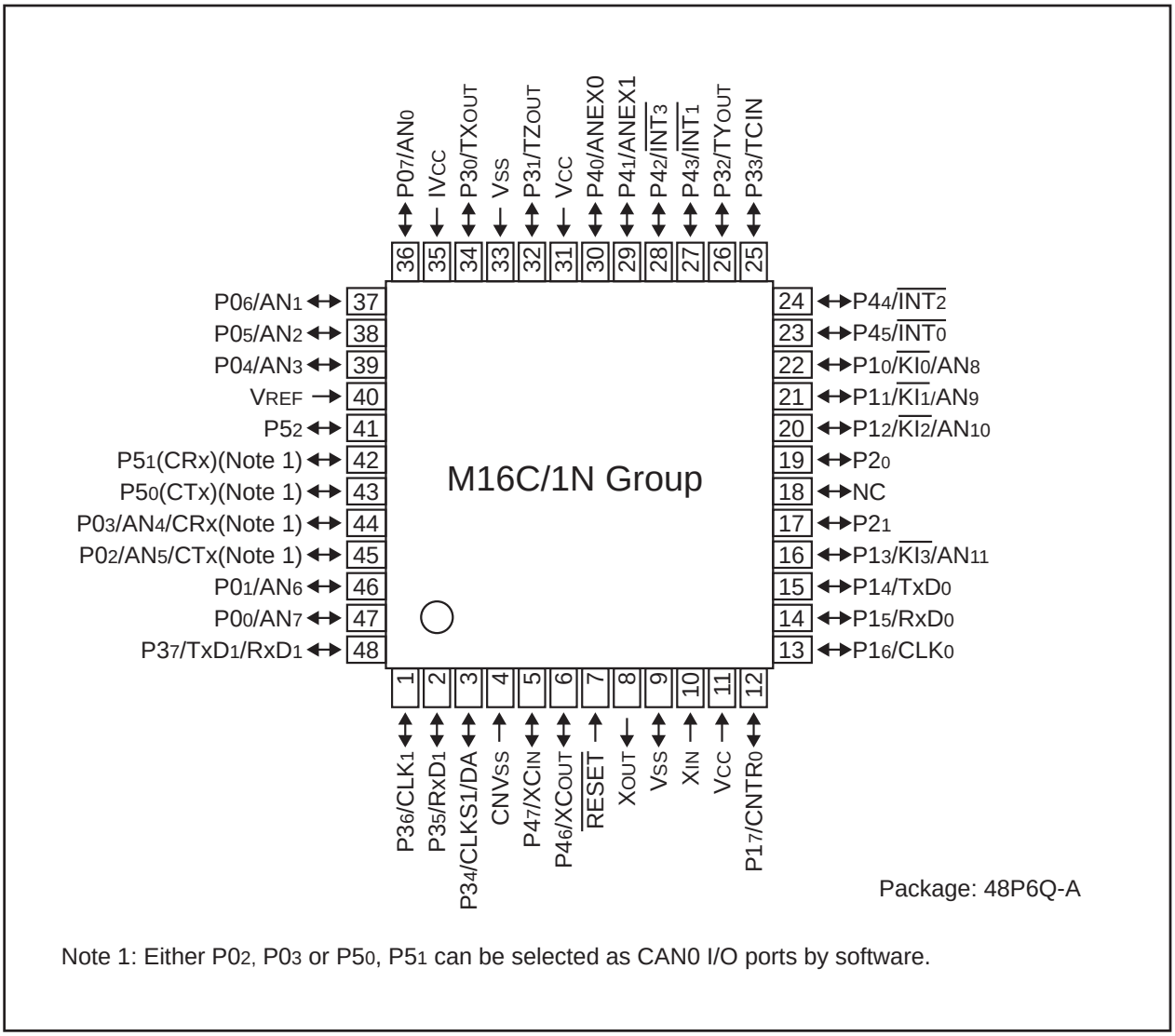


Figure 1.2 Type No., memory size, and package

1.5 Pin Configuration

Figure 1.3 shows pin configurations (top view) of the M16C/1N group.



Note 1: Either P02, P03 or P50, P51 can be selected as CAN0 I/O ports by software.

Figure 1.3 Pin configuration diagram (top view)

1.6 Pin Description

Table 1.3 shows the pin description.

Table 1.3 Pin Description

Pin name	Signal name	I/O type	Function
VCC, VSS	Power supply input	Input	Supply 4.2 to 5.5 V to the VCC pin. Supply 0 V to the VSS pin.
IVCC	IVCC	Input	Connect a capacitor (0.1 µF) between this pin and VSS.
CNVSS	CNVSS	Input	Connect it to the VSS pin via resistance (about 5 kΩ).
$\overline{\text{RESET}}$	Reset input	Input	A "L" on this input resets the microcomputer.
XIN	Clock input	Input	These pins are provided for the main clock oscillation circuit. Connect a ceramic resonator or crystal between the XIN and XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
XOUT	Clock output	Output	
VREF	Reference voltage input	Input	This pin is a reference voltage input for the A/D converter.
P00 to P07	I/O port P0	Input/output	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When set for input, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistor. These pins are shared with analog input pins. P02 and P03 function as CAN0 I/O pins by using software.
P10 to P17	I/O port P1	Input/output	This is an 8-bit I/O port equivalent to P0. P10 to P13 are shared with analog inputs and key input interrupts. P14 to P16 are shared with serial I/O pins. P17 is shared with timer input. Can be used as an LED drive port.
P20 to P21	I/O port P2	Input/output	This is a 2-bit I/O port equivalent to P0.
P30 to P37	I/O port P3	Input/output	This is a 8-bit I/O port equivalent to P0. P30 to P33 are shared with timer input/output. P34 to P37 are shared with serial I/O. P34 is shared with analog outputs.
P40 to P47	I/O port P4	Input/output	This is a 8-bit I/O port equivalent to P0. P40 to 41 are shared with analog inputs. P42 to P45 are shared with interrupt inputs. P46 to P47 are shared with the I/O pin of the clock oscillation circuit for the clock.
P50 to P52	I/O port P5	Input/output	This is a 3-bit I/O port equivalent to P0. P50 and P51 function as CAN0 I/O pins by using software.

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.

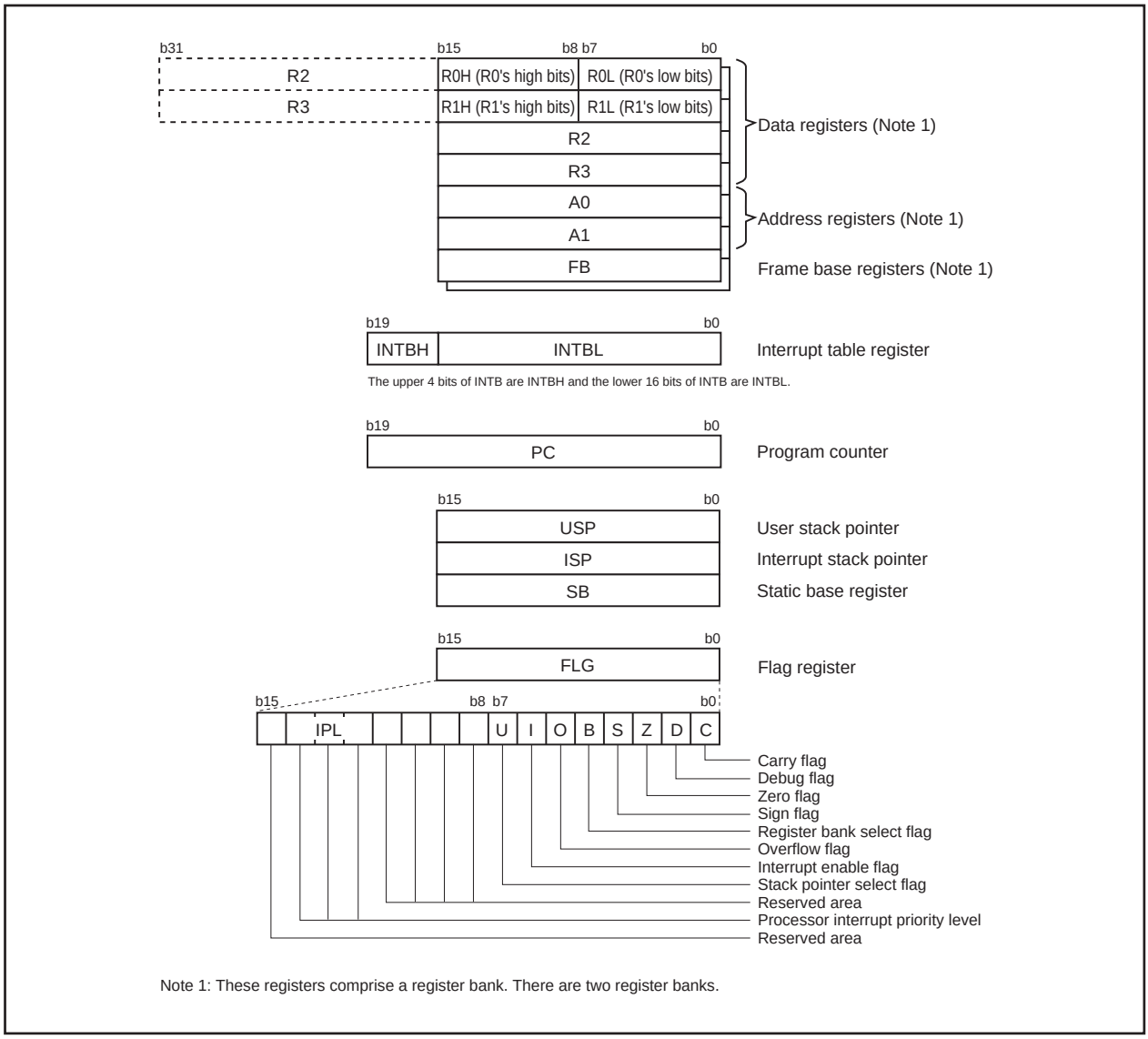


Figure 2.1 CPU Registers

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 white to this bit, write “0”. When read, its content is indeterminate.

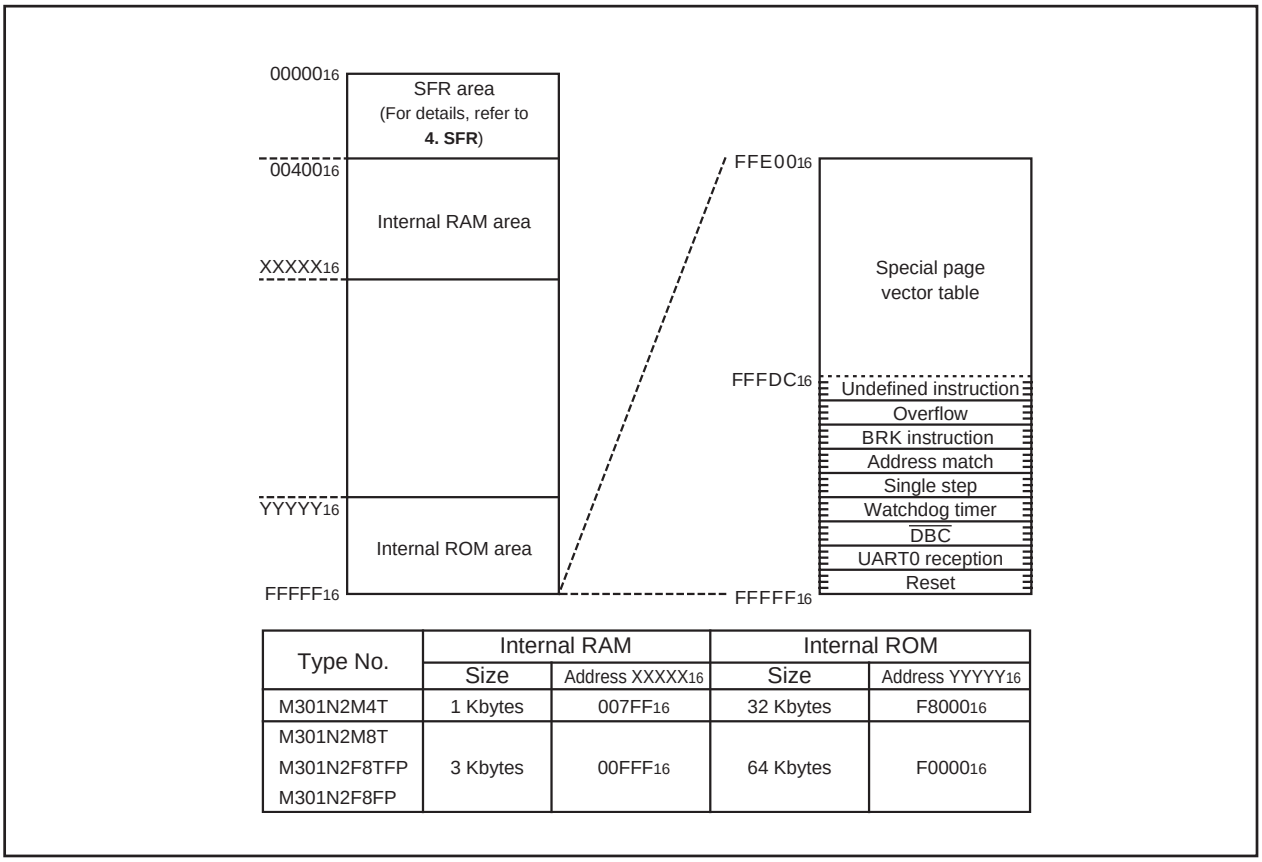
3. Memory

Figure 3.1 is a memory map. The address space extends the 1M bytes from address 00000$_{16}$ to FFFFF$_{16}$. From FFFFF$_{16}$ down is ROM. For example, in the M301N2M4T-XXXFP, there is 32K bytes of internal ROM from F8000$_{16}$ to FFFFF$_{16}$. The vector table for fixed interrupts such as the reset are mapped to FFFDC$_{16}$ to FFFFF$_{16}$. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

From 00400$_{16}$ up is RAM. For example, in the M301N2M4T-XXXFP, there is 1K byte of internal RAM from 00400$_{16}$ to 007FF$_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SFR area is mapped to 00000$_{16}$ to 003FF$_{16}$. This area accommodates the control registers for peripheral devices such as I/O ports, A/D converter, serial I/O, and timers, etc. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.

The special page vector table is mapped to FFE00$_{16}$ to FFFDB$_{16}$. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.



Type No.	Internal RAM		Internal ROM	
	Size	Address XXXXX$_{16}$	Size	Address YYYYY$_{16}$
M301N2M4T	1 Kbytes	007FF$_{16}$	32 Kbytes	F8000$_{16}$
M301N2M8T M301N2F8TFP M301N2F8FP	3 Kbytes	00FFF$_{16}$	64 Kbytes	F0000$_{16}$

Figure 3.1 Memory map

4. Special Function Registers (SFR)

Address	Register	Symbol	After reset
0000$_{16}$			
0001$_{16}$			
0002$_{16}$			
0003$_{16}$			
0004$_{16}$	Processor mode register 0	PM0	XXXX0X00$_2$
0005$_{16}$	Processor mode register 1	PM1	00XXX0X0$_2$
0006$_{16}$	System clock control register 0	CM0	48$_{16}$
0007$_{16}$	System clock control register 1	CM1	20$_{16}$
0008$_{16}$			
0009$_{16}$	Address match interrupt enable register	AIER	XXXXXX00$_2$
000A$_{16}$	Protect register	PRCR	XXXXX000$_2$
000B$_{16}$			
000C$_{16}$	Oscillation stop detection register	CM2	04$_{16}$
000D$_{16}$			
000E$_{16}$	Watchdog timer start register	WDTS	XX$_{16}$
000F$_{16}$	Watchdog timer control register	WDC	000XXXXX$_2$
0010$_{16}$	Address match interrupt register 0	RMAD0	00000000$_2$
0011$_{16}$			00000000$_2$
0012$_{16}$			XXXX0000$_2$
0013$_{16}$			
0014$_{16}$	Address match interrupt register 1	RMAD1	00000000$_2$
0015$_{16}$			00000000$_2$
0016$_{16}$			XXXX0000$_2$
0017$_{16}$			
0018$_{16}$			
0019$_{16}$			
001A$_{16}$			
001B$_{16}$			
001C$_{16}$			
001D$_{16}$			
001E$_{16}$	$\overline{\text{INT0}}$ input filter select register	INT0F	XXXXX000$_2$
001F$_{16}$			
0020$_{16}$			
0021$_{16}$			
0022$_{16}$			
0023$_{16}$			
0024$_{16}$			
0025$_{16}$			
0026$_{16}$			
0027$_{16}$			
0028$_{16}$			
0029$_{16}$			
002A$_{16}$			
002B$_{16}$			
002C$_{16}$			
002D$_{16}$			
002E$_{16}$			
002F$_{16}$			
0030$_{16}$			
0031$_{16}$			
0032$_{16}$			
0033$_{16}$			
0034$_{16}$			
0035$_{16}$			
0036$_{16}$			
0037$_{16}$			
0038$_{16}$			
0039$_{16}$			
003A$_{16}$			
003B$_{16}$			
003C$_{16}$			
003D$_{16}$			
003E$_{16}$			
003F$_{16}$			

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

X : Undefined

Address	Register	Symbol	After reset
0040$_{16}$			
0041$_{16}$			
0042$_{16}$			
0043$_{16}$			
0044$_{16}$			
0045$_{16}$	CAN0 wakeup interrupt control register	C01WKIC	XXXXX000$_2$
0046$_{16}$	CAN0 state/error interrupt control register	C01ERRIC	XXXXX000$_2$
0047$_{16}$			
0048$_{16}$	CAN0 reception successful interrupt control register	C0RECIC	XXXXX000$_2$
0049$_{16}$	CAN0 transmission successful interrupt control register	C0TRMIC	XXXXX000$_2$
004A$_{16}$			
004B$_{16}$			
004C$_{16}$			
004D$_{16}$	Key input interrupt control register	KUPIC	XXXXX000$_2$
004E$_{16}$	A/D conversion interrupt control register	ADIC	XXXXX000$_2$
004F$_{16}$			
0050$_{16}$			
0051$_{16}$	UART0 transmit interrupt control register	S0TIC	XXXXX000$_2$
0052$_{16}$	UART0 receive interrupt control register	S0RIC	XXXXX000$_2$
0053$_{16}$	UART1 transmit interrupt control register	S1TIC	XXXXX000$_2$
0054$_{16}$	UART1 receive interrupt control register	S1RIC	XXXXX000$_2$
0055$_{16}$	Timer 1 interrupt control register	T1IC	XXXXX000$_2$
0056$_{16}$	Timer X interrupt control register	TXIC	XXXXX000$_2$
0057$_{16}$	Timer Y interrupt control register	TYIC	XXXXX000$_2$
0058$_{16}$	Timer Z interrupt control register	TZIC	XXXXX000$_2$
0059$_{16}$	CNTR0 interrupt control register	CNTR0IC	XXXXX000$_2$
005A$_{16}$	TCIN interrupt control register	TCINIC	XXXXX000$_2$
005B$_{16}$	Timer C interrupt control register	TCIC	XXXXX000$_2$
005C$_{16}$	INT3 interrupt control register	INT3IC	XXXXX000$_2$
005D$_{16}$	INT0 interrupt control register	INT0IC	XX00X000$_2$
005E$_{16}$	INT1 interrupt control register	INT1IC	XX00X000$_2$
005F$_{16}$	INT2 interrupt control register	INT2IC	XX00X000$_2$
0060$_{16}$			
0061$_{16}$			
0062$_{16}$			
0063$_{16}$			
0064$_{16}$			
0065$_{16}$			
0066$_{16}$			
0067$_{16}$			
0068$_{16}$			
0069$_{16}$			
006A$_{16}$			
006B$_{16}$			
006C$_{16}$			
006D$_{16}$			
006E$_{16}$			
006F$_{16}$			
0070$_{16}$			
0071$_{16}$			
0072$_{16}$			
0073$_{16}$			
0074$_{16}$			
0075$_{16}$			
0076$_{16}$			
0077$_{16}$			
0078$_{16}$			
0079$_{16}$			
007A$_{16}$			
007B$_{16}$			
007C$_{16}$			
007D$_{16}$			
007E$_{16}$			
007F$_{16}$			

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

X : Undefined

Address	Register	Symbol	After reset
0080_{16}	Timer Y, Z mode register	TYZMR	$000000X0_2$
0081_{16}	Prescaler Y	PREY	FF_{16}
0082_{16}	Timer Y secondary	TYSC	FF_{16}
0083_{16}	Timer Y primary	TYPR	FF_{16}
0084_{16}	Timer Y, Z waveform output control register	PUM	00_{16}
0085_{16}	Prescaler Z	PREZ	FF_{16}
0086_{16}	Timer Z secondary	TZSC	FF_{16}
0087_{16}	Timer Z primary	TZPR	FF_{16}
0088_{16}	Prescaler 1	PRE1	XX_{16}
0089_{16}	Timer 1	T1	XX_{16}
$008A_{16}$	Timer Y, Z output control register	TYZOC	$XXXXX000_2$
$008B_{16}$	Timer X mode register	TXMR	00000000_2
$008C_{16}$	Prescaler X	PREX	FF_{16}
$008D_{16}$	Timer X	TX	FF_{16}
$008E_{16}$	Timer count source set register	TCSS	00_{16}
$008F_{16}$	Clock prescaler reset flag	CPSRF	$0XXXXXXX_2$
0090_{16}	Timer C counter	TC	XX_{16}
0091_{16}			XX_{16}
0092_{16}			
0093_{16}			
0094_{16}			
0095_{16}			
0096_{16}	External input enable register	INTEN	00_{16}
0097_{16}			
0098_{16}	Key input enable register	KIEN	00_{16}
0099_{16}			
$009A_{16}$	Timer C control register 0	TCC0	$0XX00000_2$
$009B_{16}$	Timer C control register 1	TCC1	$XXXXXX11_2$
$009C_{16}$	Time measurement register	TM	XX_{16}
$009D_{16}$			XX_{16}
$009E_{16}$			
$009F_{16}$			
$00A0_{16}$	UART0 transmit/receive mode register	U0MR	00_{16}
$00A1_{16}$	UART0 bit rate generator	U0BRG	XX_{16}
$00A2_{16}$	UART0 transmit buffer register	U0TB	XX_{16}
$00A3_{16}$			XX_{16}
$00A4_{16}$	UART0 transmit/receive control register 0	U0C0	08_{16}
$00A5_{16}$	UART0 transmit/receive control register 1	U0C1	$XXXX0010_2$
$00A6_{16}$	UART0 receive buffer register	U0RB	XX_{16}
$00A7_{16}$			XX_{16}
$00A8_{16}$	UART1 transmit/receive mode register	U1MR	00_{16}
$00A9_{16}$	UART1 bit rate generator	U1BRG	XX_{16}
$00AA_{16}$	UART1 transmit buffer register	U1TB	XX_{16}
$00AB_{16}$			XX_{16}
$00AC_{16}$	UART1 transmit/receive control register 0	U1C0	08_{16}
$00AD_{16}$	UART1 transmit/receive control register 1	U1C1	$XXXX0010_2$
$00AE_{16}$	UART1 receive buffer register	U1RB	XX_{16}
$00AF_{16}$			XX_{16}
$00B0_{16}$	UART transmit/receive control register 2	UCON	$X0000000_2$
$00B1_{16}$			
$00B2_{16}$			
$00B3_{16}$			
$00B4_{16}$			
$00B5_{16}$			
$00B6_{16}$			
$00B7_{16}$			
$00B8_{16}$			
$00B9_{16}$			
$00BA_{16}$			
$00BB_{16}$			
$00BC_{16}$			
$00BD_{16}$			
$00BE_{16}$			
$00BF_{16}$			

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

X : Undefined

Address	Register	Symbol	After reset
$00C0_{16}$	A/D register	AD	XX_{16}
$00C1_{16}$			XX_{16}
$00C2_{16}$			
$00C3_{16}$			
$00C4_{16}$			
$00C5_{16}$			
$00C6_{16}$			
$00C7_{16}$			
$00C8_{16}$			
$00C9_{16}$			
$00CA_{16}$			
$00CB_{16}$			
$00CC_{16}$			
$00CD_{16}$			
$00CE_{16}$			
$00CF_{16}$			
$00D0_{16}$			
$00D1_{16}$			
$00D2_{16}$			
$00D3_{16}$			
$00D4_{16}$	A/D control register 2	ADCON2	$XXXX0000_2$
$00D5_{16}$			
$00D6_{16}$	A/D control register 0	ADCON0	$00000XXX_2$
$00D7_{16}$	A/D control register 1	ADCON1	00_{16}
$00D8_{16}$	D/A register	DA	XX_{16}
$00D9_{16}$			
$00DA_{16}$			
$00DB_{16}$			
$00DC_{16}$	D/A control register	DACON	$XXXXX0X0_2$
$00DD_{16}$			
$00DE_{16}$			
$00DF_{16}$			
$00E0_{16}$	Port P0 register	P0	XX_{16}
$00E1_{16}$	Port P1 register	P1	XX_{16}
$00E2_{16}$	Port P0 direction register	PD0	00_{16}
$00E3_{16}$	Port P1 direction register	PD1	00_{16}
$00E4_{16}$	Port P2 register	P2	XX_{16}
$00E5_{16}$	Port P3 register	P3	XX_{16}
$00E6_{16}$	Port P2 direction register	PD2	$XXXXXX00_2$
$00E7_{16}$	Port P3 direction register	PD3	00_{16}
$00E8_{16}$	Port P4 register	P4	XX_{16}
$00E9_{16}$	Port P5 register	P5	XX_{16}
$00EA_{16}$	Port P4 direction register	PD4	00_{16}
$00EB_{16}$	Port P5 direction register	PD5	$XXXXX000_2$
$00EC_{16}$			
$00ED_{16}$			
$00EE_{16}$			
$00EF_{16}$			
$00F0_{16}$			
$00F1_{16}$			
$00F2_{16}$			
$00F3_{16}$			
$00F4_{16}$			
$00F5_{16}$			
$00F6_{16}$			
$00F7_{16}$			
$00F8_{16}$	CAN0 I/O port select register	CIOSR	$XXXXXXX0_2$
$00F9_{16}$			
$00FA_{16}$			
$00FB_{16}$			
$00FC_{16}$	Pull-up control register 0	PUR0	$00X00000_2$
$00FD_{16}$	Pull-up control register 1	PUR1	$XXXXX000_2$
$00FE_{16}$	Port P1 drive capacity control register	DRR	00_{16}
$00FF_{16}$			

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

X : Undefined

Address	Register	Symbol	After reset
0100₁₆			
0101₁₆			
0102₁₆			
0103₁₆			
0104₁₆ ≈ 01B0₁₆			
01B1₁₆			
01B2₁₆			
01B3₁₆	Flash memory control register 4 (Note 2)	FMR4	01000000₂
01B4₁₆			
01B5₁₆	Flash memory control register 1 (Note 2)	FMR1	0000XX0X₂
01B6₁₆			
01B7₁₆	Flash memory control register 0 (Note 2)	FMR0	XX000001₂
01B8₁₆			
01B9₁₆			
01BA₁₆			
01BB₁₆			
01BC₁₆			
01BD₁₆			
01BE₁₆			
01BF₁₆ ≈ 0215₁₆			
0216₁₆			
0217₁₆			
0218₁₆			
0219₁₆			
021A₁₆			
021B₁₆			
021C₁₆			
021D₁₆			
021E₁₆			
021F₁₆			
0220₁₆	CAN0 message control register 0	C0MCTL0	00₁₆
0221₁₆	CAN0 message control register 1	C0MCTL1	00₁₆
0222₁₆	CAN0 message control register 2	C0MCTL2	00₁₆
0223₁₆	CAN0 message control register 3	C0MCTL3	00₁₆
0224₁₆	CAN0 message control register 4	C0MCTL4	00₁₆
0225₁₆	CAN0 message control register 5	C0MCTL5	00₁₆
0226₁₆	CAN0 message control register 6	C0MCTL6	00₁₆
0227₁₆	CAN0 message control register 7	C0MCTL7	00₁₆
0228₁₆	CAN0 message control register 8	C0MCTL8	00₁₆
0229₁₆	CAN0 message control register 9	C0MCTL9	00₁₆
022A₁₆	CAN0 message control register 10	C0MCTL10	00₁₆
022B₁₆	CAN0 message control register 11	C0MCTL11	00₁₆
022C₁₆	CAN0 message control register 12	C0MCTL12	00₁₆
022D₁₆	CAN0 message control register 13	C0MCTL13	00₁₆
022E₁₆	CAN0 message control register 14	C0MCTL14	00₁₆
022F₁₆	CAN0 message control register 15	C0MCTL15	00₁₆
0230₁₆	CAN0 control register	C0CTLR	X0000001₂
0231₁₆			XX0X0000₂
0232₁₆	CAN0 status register	C0STR	00₁₆
0233₁₆			X0000001₂
0234₁₆	CAN0 slot status register	C0SSTR	0000₁₆
0235₁₆			0000₁₆
0236₁₆	CAN0 interrupt control register	C0ICR	0000₁₆
0237₁₆			0000₁₆
0238₁₆	CAN0 extended ID register	C0IDR	0000₁₆
0239₁₆			0000₁₆
023A₁₆	CAN0 configuration register	C0CONR	XX₁₆
023B₁₆			XX₁₆
023C₁₆	CAN0 receive error count register	C0RECR	00₁₆
023D₁₆	CAN0 transmit error count register	C0TECR	00₁₆
023E₁₆			
023F₁₆			

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.
Note 2: These registers are available on flash memory versions only.

X : Undefined

Address	Register	Symbol	After reset
0240$_{16}$			
0241$_{16}$			
0242$_{16}$			
0243$_{16}$			
0244$_{16}$	CAN0 acceptance filter support register	C0AFS	XX$_{16}$
0245$_{16}$			XX$_{16}$
0246$_{16}$			
0247$_{16}$			
0248$_{16}$			
0249$_{16}$			
024A$_{16}$			
024B$_{16}$			
024C$_{16}$			
024D$_{16}$			
024E$_{16}$			
024F$_{16}$			
0250$_{16}$			
0251$_{16}$			
0252$_{16}$			
0253$_{16}$			
0254$_{16}$			
0255$_{16}$			
0256$_{16}$			
0257$_{16}$			
0258$_{16}$			
0259$_{16}$			
025A$_{16}$			
025B$_{16}$			
025C$_{16}$			
025D$_{16}$			
025E$_{16}$			
025F$_{16}$	CAN0 clock select register	CCLKR	X000XXXX$_2$
0260$_{16}$	CAN0 slot 0: Identifier / DLC		XX$_{16}$
0261$_{16}$			XX$_{16}$
0262$_{16}$			XX$_{16}$
0263$_{16}$			XX$_{16}$
0264$_{16}$			XX$_{16}$
0265$_{16}$			XX$_{16}$
0266$_{16}$	CAN0 slot 0: Data Field		XX$_{16}$
0267$_{16}$			XX$_{16}$
0268$_{16}$			XX$_{16}$
0269$_{16}$			XX$_{16}$
026A$_{16}$			XX$_{16}$
026B$_{16}$			XX$_{16}$
026C$_{16}$			XX$_{16}$
026D$_{16}$			XX$_{16}$
026E$_{16}$	CAN0 slot 0: Time Stamp		XX$_{16}$
026F$_{16}$			XX$_{16}$
0270$_{16}$	CAN0 slot 1: Identifier / DLC		XX$_{16}$
0271$_{16}$			XX$_{16}$
0272$_{16}$			XX$_{16}$
0273$_{16}$			XX$_{16}$
0274$_{16}$			XX$_{16}$
0275$_{16}$			XX$_{16}$
0276$_{16}$	CAN0 slot 1: Data Field		XX$_{16}$
0277$_{16}$			XX$_{16}$
0278$_{16}$			XX$_{16}$
0279$_{16}$			XX$_{16}$
027A$_{16}$			XX$_{16}$
027B$_{16}$			XX$_{16}$
027C$_{16}$			XX$_{16}$
027D$_{16}$			XX$_{16}$
027E$_{16}$	CAN0 slot 1: Time Stamp		XX$_{16}$
027F$_{16}$			XX$_{16}$

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

X : Undefined

Address	Register	Symbol	After reset
0280$_{16}$	CAN0 slot 2: Identifier / DLC		XX$_{16}$
0281$_{16}$			XX$_{16}$
0282$_{16}$			XX$_{16}$
0283$_{16}$			XX$_{16}$
0284$_{16}$			XX$_{16}$
0285$_{16}$			XX$_{16}$
0286$_{16}$	CAN0 slot 2: Data Field		XX$_{16}$
0287$_{16}$			XX$_{16}$
0288$_{16}$			XX$_{16}$
0289$_{16}$			XX$_{16}$
028A$_{16}$			XX$_{16}$
028B$_{16}$			XX$_{16}$
028C$_{16}$			XX$_{16}$
028D$_{16}$			XX$_{16}$
028E$_{16}$	CAN0 slot 2: Time Stamp		XX$_{16}$
028F$_{16}$			XX$_{16}$
0290$_{16}$	CAN0 slot 3: Identifier / DLC		XX$_{16}$
0291$_{16}$			XX$_{16}$
0292$_{16}$			XX$_{16}$
0293$_{16}$			XX$_{16}$
0294$_{16}$			XX$_{16}$
0295$_{16}$			XX$_{16}$
0296$_{16}$	CAN0 slot 3: Data Field		XX$_{16}$
0297$_{16}$			XX$_{16}$
0298$_{16}$			XX$_{16}$
0299$_{16}$			XX$_{16}$
029A$_{16}$			XX$_{16}$
029B$_{16}$			XX$_{16}$
029C$_{16}$			XX$_{16}$
029D$_{16}$			XX$_{16}$
029E$_{16}$	CAN0 slot 3: Time Stamp		XX$_{16}$
029F$_{16}$			XX$_{16}$
02A0$_{16}$	CAN0 slot 4: Identifier / DLC		XX$_{16}$
02A1$_{16}$			XX$_{16}$
02A2$_{16}$			XX$_{16}$
02A3$_{16}$			XX$_{16}$
02A4$_{16}$			XX$_{16}$
02A5$_{16}$			XX$_{16}$
02A6$_{16}$	CAN0 slot 4: Data Field		XX$_{16}$
02A7$_{16}$			XX$_{16}$
02A8$_{16}$			XX$_{16}$
02A9$_{16}$			XX$_{16}$
02AA$_{16}$			XX$_{16}$
02AB$_{16}$			XX$_{16}$
02AC$_{16}$			XX$_{16}$
02AD$_{16}$			XX$_{16}$
02AE$_{16}$	CAN0 slot 4: Time Stamp		XX$_{16}$
02AF$_{16}$			XX$_{16}$
02B0$_{16}$	CAN0 slot 5: Identifier / DLC		XX$_{16}$
02B1$_{16}$			XX$_{16}$
02B2$_{16}$			XX$_{16}$
02B3$_{16}$			XX$_{16}$
02B4$_{16}$			XX$_{16}$
02B5$_{16}$			XX$_{16}$
02B6$_{16}$	CAN0 slot 5: Data Field		XX$_{16}$
02B7$_{16}$			XX$_{16}$
02B8$_{16}$			XX$_{16}$
02B9$_{16}$			XX$_{16}$
02BA$_{16}$			XX$_{16}$
02BB$_{16}$			XX$_{16}$
02BC$_{16}$			XX$_{16}$
02BD$_{16}$			XX$_{16}$
02BE$_{16}$	CAN0 slot 5: Time Stamp		XX$_{16}$
02BF$_{16}$			XX$_{16}$

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

X : Undefined

Address	Register	Symbol	After reset
02C0$_{16}$	CAN0 slot 6: Identifier / DLC		XX$_{16}$
02C1$_{16}$			XX$_{16}$
02C2$_{16}$			XX$_{16}$
02C3$_{16}$			XX$_{16}$
02C4$_{16}$			XX$_{16}$
02C5$_{16}$			XX$_{16}$
02C6$_{16}$	CAN0 slot 6: Data Field		XX$_{16}$
02C7$_{16}$			XX$_{16}$
02C8$_{16}$			XX$_{16}$
02C9$_{16}$			XX$_{16}$
02CA$_{16}$			XX$_{16}$
02CB$_{16}$			XX$_{16}$
02CC$_{16}$			XX$_{16}$
02CD$_{16}$			XX$_{16}$
02CE$_{16}$	CAN0 slot 6: Time Stamp		XX$_{16}$
02CF$_{16}$			XX$_{16}$
02D0$_{16}$	CAN0 slot 7: Identifier / DLC		XX$_{16}$
02D1$_{16}$			XX$_{16}$
02D2$_{16}$			XX$_{16}$
02D3$_{16}$			XX$_{16}$
02D4$_{16}$			XX$_{16}$
02D5$_{16}$			XX$_{16}$
02D6$_{16}$	CAN0 slot 7: Data Field		XX$_{16}$
02D7$_{16}$			XX$_{16}$
02D8$_{16}$			XX$_{16}$
02D9$_{16}$			XX$_{16}$
02DA$_{16}$			XX$_{16}$
02DB$_{16}$			XX$_{16}$
02DC$_{16}$			XX$_{16}$
02DD$_{16}$			XX$_{16}$
02DE$_{16}$	CAN0 slot 7: Time Stamp		XX$_{16}$
02DF$_{16}$			XX$_{16}$
02E0$_{16}$	CAN0 slot 8: Identifier / DLC		XX$_{16}$
02E1$_{16}$			XX$_{16}$
02E2$_{16}$			XX$_{16}$
02E3$_{16}$			XX$_{16}$
02E4$_{16}$			XX$_{16}$
02E5$_{16}$			XX$_{16}$
02E6$_{16}$	CAN0 slot 8: Data Field		XX$_{16}$
02E7$_{16}$			XX$_{16}$
02E8$_{16}$			XX$_{16}$
02E9$_{16}$			XX$_{16}$
02EA$_{16}$			XX$_{16}$
02EB$_{16}$			XX$_{16}$
02EC$_{16}$			XX$_{16}$
02ED$_{16}$			XX$_{16}$
02EE$_{16}$	CAN0 slot 8: Time Stamp		XX$_{16}$
02EF$_{16}$			XX$_{16}$
02F0$_{16}$	CAN0 slot 9: Identifier / DLC		XX$_{16}$
02F1$_{16}$			XX$_{16}$
02F2$_{16}$			XX$_{16}$
02F3$_{16}$			XX$_{16}$
02F4$_{16}$			XX$_{16}$
02F5$_{16}$			XX$_{16}$
02F6$_{16}$	CAN0 slot 9: Data Field		XX$_{16}$
02F7$_{16}$			XX$_{16}$
02F8$_{16}$			XX$_{16}$
02F9$_{16}$			XX$_{16}$
02FA$_{16}$			XX$_{16}$
02FB$_{16}$			XX$_{16}$
02FC$_{16}$			XX$_{16}$
02FD$_{16}$			XX$_{16}$
02FE$_{16}$	CAN0 slot 9: Time Stamp		XX$_{16}$
02FF$_{16}$			XX$_{16}$

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

X : Undefined

Address	Register	Symbol	After reset
0300$_{16}$	CAN0 slot 10: Identifier / DLC		XX$_{16}$
0301$_{16}$			XX$_{16}$
0302$_{16}$			XX$_{16}$
0303$_{16}$			XX$_{16}$
0304$_{16}$			XX$_{16}$
0305$_{16}$			XX$_{16}$
0306$_{16}$	CAN0 slot 10: Data Field		XX$_{16}$
0307$_{16}$			XX$_{16}$
0308$_{16}$			XX$_{16}$
0309$_{16}$			XX$_{16}$
030A$_{16}$			XX$_{16}$
030B$_{16}$			XX$_{16}$
030C$_{16}$			XX$_{16}$
030D$_{16}$			XX$_{16}$
030E$_{16}$	CAN0 slot 10: Time Stamp		XX$_{16}$
030F$_{16}$			XX$_{16}$
0310$_{16}$	CAN0 slot 11: Identifier / DLC		XX$_{16}$
0311$_{16}$			XX$_{16}$
0312$_{16}$			XX$_{16}$
0313$_{16}$			XX$_{16}$
0314$_{16}$			XX$_{16}$
0315$_{16}$			XX$_{16}$
0316$_{16}$	CAN0 slot 11: Data Field		XX$_{16}$
0317$_{16}$			XX$_{16}$
0318$_{16}$			XX$_{16}$
0319$_{16}$			XX$_{16}$
031A$_{16}$			XX$_{16}$
031B$_{16}$			XX$_{16}$
031C$_{16}$			XX$_{16}$
031D$_{16}$			XX$_{16}$
031E$_{16}$	CAN0 slot 11: Time Stamp		XX$_{16}$
031F$_{16}$			XX$_{16}$
0320$_{16}$	CAN0 slot 12: Identifier / DLC		XX$_{16}$
0321$_{16}$			XX$_{16}$
0322$_{16}$			XX$_{16}$
0323$_{16}$			XX$_{16}$
0324$_{16}$			XX$_{16}$
0325$_{16}$			XX$_{16}$
0326$_{16}$	CAN0 slot 12: Data Field		XX$_{16}$
0327$_{16}$			XX$_{16}$
0328$_{16}$			XX$_{16}$
0329$_{16}$			XX$_{16}$
032A$_{16}$			XX$_{16}$
032B$_{16}$			XX$_{16}$
032C$_{16}$			XX$_{16}$
032D$_{16}$			XX$_{16}$
032E$_{16}$	CAN0 slot 12: Time Stamp		XX$_{16}$
032F$_{16}$			XX$_{16}$
0330$_{16}$	CAN0 slot 13: Identifier / DLC		XX$_{16}$
0331$_{16}$			XX$_{16}$
0332$_{16}$			XX$_{16}$
0333$_{16}$			XX$_{16}$
0334$_{16}$			XX$_{16}$
0335$_{16}$			XX$_{16}$
0336$_{16}$	CAN0 slot 13: Data Field		XX$_{16}$
0337$_{16}$			XX$_{16}$
0338$_{16}$			XX$_{16}$
0339$_{16}$			XX$_{16}$
033A$_{16}$			XX$_{16}$
033B$_{16}$			XX$_{16}$
033C$_{16}$			XX$_{16}$
033D$_{16}$			XX$_{16}$
033E$_{16}$	CAN0 slot 13: Time Stamp		XX$_{16}$
033F$_{16}$			XX$_{16}$

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

X : Undefined

Address	Register	Symbol	After reset
0340₁₆	CAN0 slot 14: Identifier / DLC		XX₁₆
0341₁₆			XX₁₆
0342₁₆			XX₁₆
0343₁₆			XX₁₆
0344₁₆			XX₁₆
0345₁₆			XX₁₆
0346₁₆	CAN0 slot 14: Data Field		XX₁₆
0347₁₆			XX₁₆
0348₁₆			XX₁₆
0349₁₆			XX₁₆
034A₁₆			XX₁₆
034B₁₆			XX₁₆
034C₁₆			XX₁₆
034D₁₆			XX₁₆
034E₁₆	CAN0 slot 14: Time Stamp		XX₁₆
034F₁₆			XX₁₆
0350₁₆	CAN0 slot 15: Identifier / DLC		XX₁₆
0351₁₆			XX₁₆
0352₁₆			XX₁₆
0353₁₆			XX₁₆
0354₁₆			XX₁₆
0355₁₆			XX₁₆
0356₁₆	CAN0 slot 15: Data Field		XX₁₆
0357₁₆			XX₁₆
0358₁₆			XX₁₆
0359₁₆			XX₁₆
035A₁₆			XX₁₆
035B₁₆			XX₁₆
035C₁₆			XX₁₆
035D₁₆			XX₁₆
035E₁₆	CAN0 slot 15: Time Stamp		XX₁₆
035F₁₆			XX₁₆
0360₁₆	CAN0 Global mask	C0GMR	XX₁₆
0361₁₆			XX₁₆
0362₁₆			XX₁₆
0363₁₆			XX₁₆
0364₁₆			XX₁₆
0365₁₆			XX₁₆
0366₁₆	CAN0 local mask A	C0LMAR	XX₁₆
0367₁₆			XX₁₆
0368₁₆			XX₁₆
0369₁₆			XX₁₆
036A₁₆			XX₁₆
036B₁₆			XX₁₆
036C₁₆	CAN0 local mask B	C0LMBR	XX₁₆
036D₁₆			XX₁₆
036E₁₆			XX₁₆
036F₁₆			XX₁₆
0370₁₆			XX₁₆
0371₁₆			XX₁₆
≈			≈
03B4₁₆			
03B5₁₆			
03B6₁₆			
03B7₁₆			
03B8₁₆			
03B9₁₆			
≈			≈
03FA₁₆			
03FB₁₆			
03FC₁₆			
03FD₁₆			
03FE₁₆			
03FF₁₆			

Note 1: Location in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

X : Undefined

5. Reset

There are two types of resets; hardware and software. In both cases, operation is the same after the reset.

5.1 Hardware Reset

A reset is applied using the $\overline{\text{RESET}}$ pin.

When the supply voltage is in the range where operation is guaranteed, a reset is effected by holding the reset pin level "L" (0.2V_{CC} max.). When the $\overline{\text{RESET}}$ pin level is then returned to the "H" level, the reset status is cancelled and program execution resumes from the address in the reset vector table. Since the value of RAM is indeterminate when power is applied, the initial values must be set. Also, if a reset signal is input during write to RAM, the access to the RAM will be interrupted. Consequently, the value of the RAM being written may change to an unintended value due to the interruption.

Note 1: M16C/1N group is delayed more than 2ms until the execution of the program after reset clear in comparison with M16C/10 group products.

Figures 5.1 and 5.2 show the example reset circuit. Figure 5.3 shows the reset sequence.

5.1.1 When the power supply is stable

(1)Apply a "L" signal to the $\overline{\text{RESET}}$ pin for at least 200μs.
(2)Apply a "H" signal to the $\overline{\text{RESET}}$ pin.

5.1.2 Power on

(1)Apply a "L" signal to the $\overline{\text{RESET}}$ pin.
(2)Let the power supply voltage increase until it meets the recommended operating condition.
(3)Wait for td(P-R) + 200μs or more until the internal power supply stabilizes.
(4)Apply a "H" signal to the $\overline{\text{RESET}}$ pin.

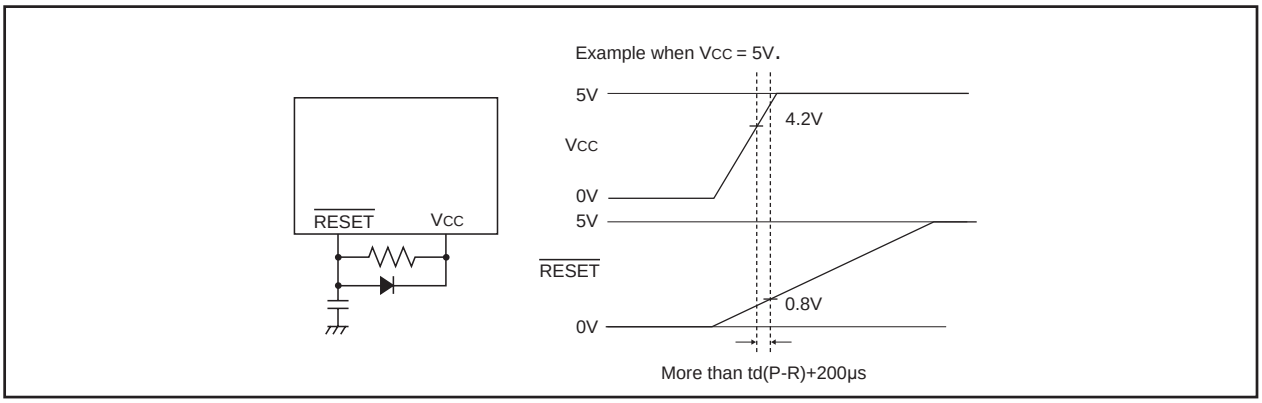


Figure 5.1 Example reset circuit

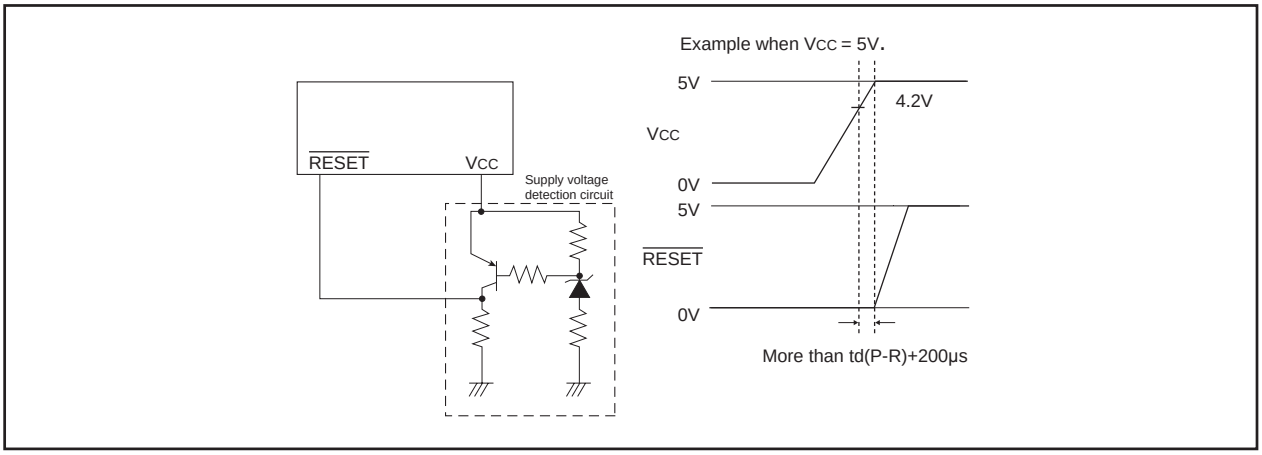


Figure 5.2 Example reset circuit (example voltage check circuit)

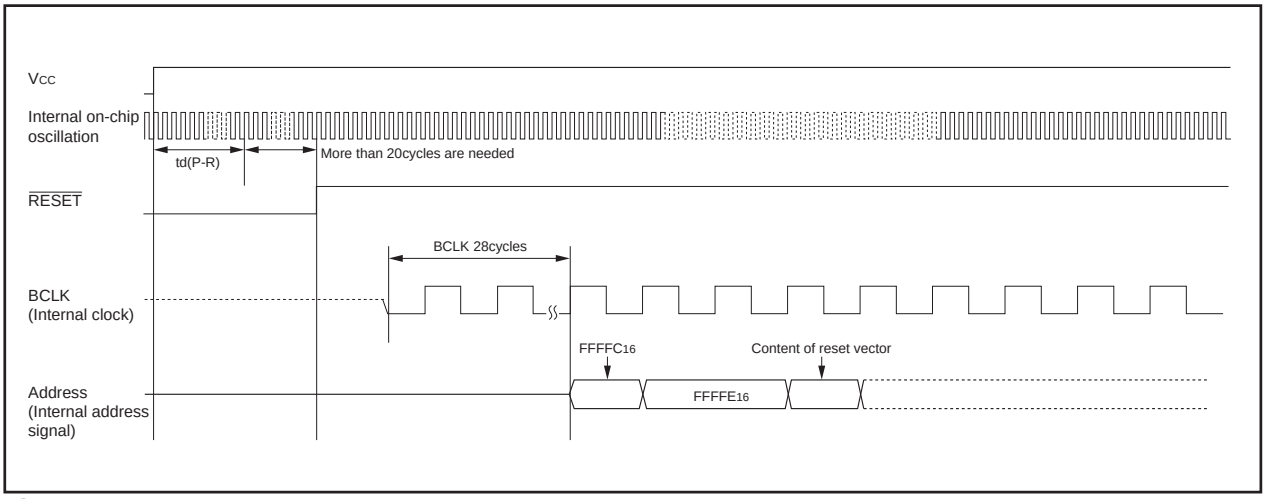


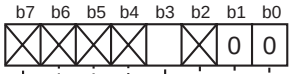
Figure 5.3 Reset sequence

5.2 Software Reset

Writing "1" to bit 3 of the processor mode register 0 (address 0004_{16}) applies a (software) reset to the microcomputer. Set the PM03 bit to "1" after selecting on-chip oscillator for CPU's operating clock source. A software reset has almost the same effect as a hardware reset. The contents of internal RAM are preserved.

Figure 5.4 shows the processor mode register 0 and 1.

Processor mode register 0 (Note 1)

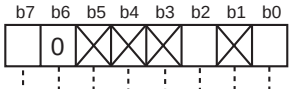


Symbol	Address	When reset
PM0	0004_{16}	$XXXX0X00_2$

Bit symbol	Bit name	Function	RW
Reserved bit		Set to "0"	RW
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			—
PM03	Software reset bit	The device is reset when this bit is set to "1". The value of this bit is "0" when read.	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to "1" when writing new values to this register.

Processor mode register 1 (Note 1)



Symbol	Address	When reset
PM1	0005_{16}	$00XXX0X0_2$

Bit symbol	Bit name	Function	RW
PM10	DATAROM area access bit (Note 2)	0 : Disabled 1 : Enabled	RW
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			—
PM12	WDT interrupt/reset switching bit	0 : Watchdog timer interrupt 1 : Reset (Note 3)	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—
Reserved bit		Set to "0"	RW
PM17	Wait bit	0 : No wait 1 : Wait	RW

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to "1" when writing new values to this register.
Note 2: This bit is valid for the flash memory version. For the mask ROM version, this bit must be set to "0".
Note 3: After setting this bit to "1", can not change to "0" by software.

Figure 5.4 Processor mode register 0 and 1

6. Clock Generation Circuit

The clock regeneration circuit contains three circuits as follows:

- Main clock oscillation circuit
- Sub clock oscillation circuit
- On-chip oscillator

Table 6.1 lists Clock Generation Circuit Specifications. Figure 6.1 shows a Clock Generation Circuit. Figure 6.2, 6.3 and 6.5 show clock-associated registers. Figure 6.4 shows fC32 block diagram.

Table 6.1 Main clock, sub-clock, and on-chip oscillator circuits

	Main clock oscillation circuit	Sub clock oscillation circuit	On-chip oscillator circuit
Use of clock	• CPU's operating clock source • Internal peripheral unit's operating clock source	• CPU's operating clock source • Timer 1/X/Y/Z's count clock source	• CPU's operating clock source • Internal peripheral unit's operating clock source • Timer Y's count clock source
Usable oscillator connectable (Note 1)	• Ceramic oscillator • Crystal oscillator	• Crystal oscillator	–
Oscillator connect pins	XIN, XOUT	XCIN, XCOUT	None (has internal pins)
Oscillation stop/restart function	Available	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped	Oscillating
Other	Externally generated clock can be input		–

Note 1: When not using the main clock generating circuit, pull up the XIN pin and leave the XOUT pin open. Also, set the main clock stop bit (bit 5 at address 000616) to "1" (stop).

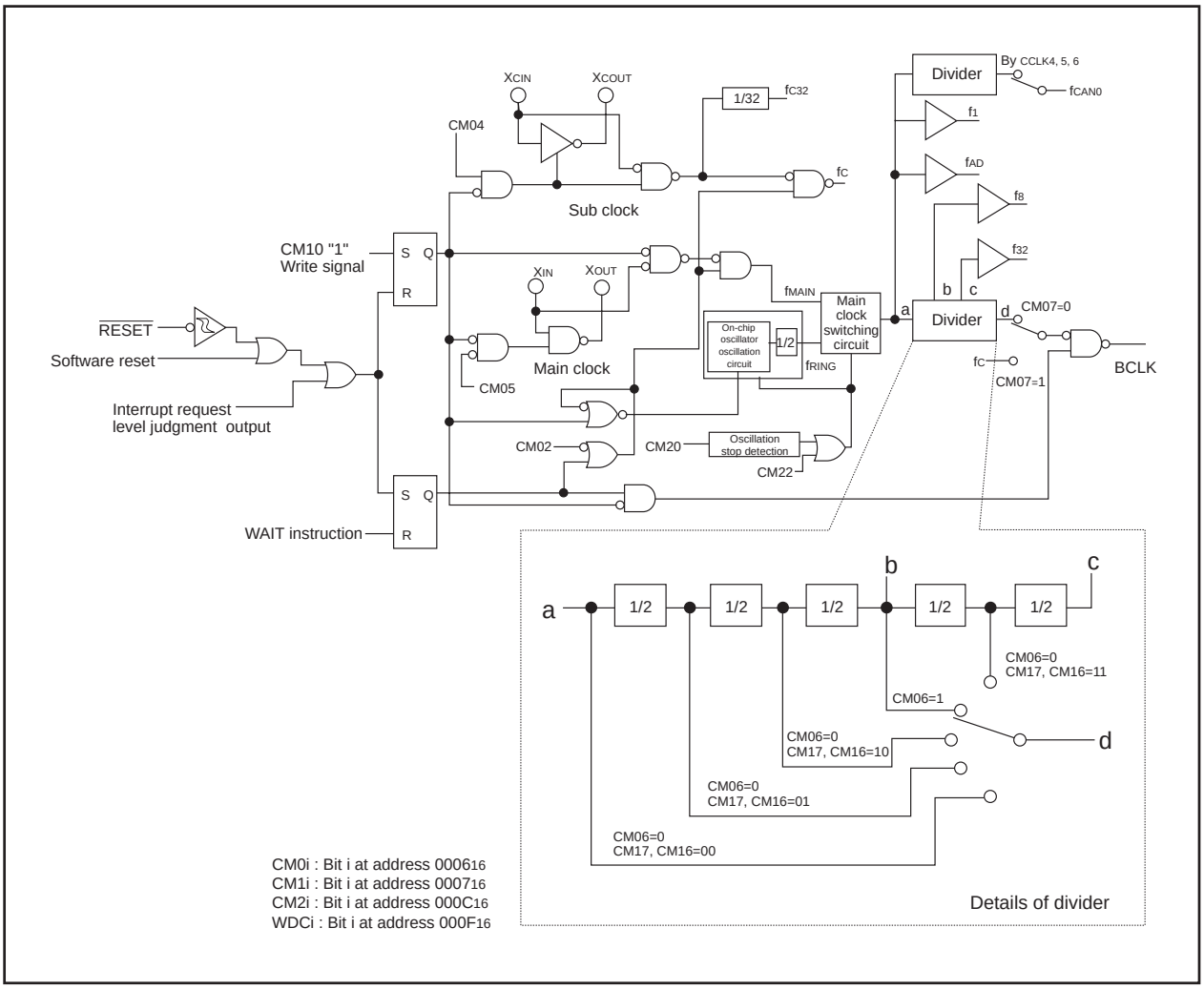


Figure 6.1 Block diagram of clock generating circuit

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol: CM0 | Address: 0006_{16} | When reset: 01001000_2

Bit symbol	Bit name	Function	RW
Reserved bit		Set to "0"	RW
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (Note 2)	RW
CM03	XCIN-XCOUT drive capacity select bit (Note 3)	0 : LOW 1 : HIGH	RW
CM04	Port XC select bit	0 : I/O port 1 : XCIN-XCOUT generation	RW
CM05	Main clock (XIN-XOUT) stop bit (Note 4,5,6)	0 : On 1 : Off	RW
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Division by 8 mode	RW
CM07	System clock select bit (Note 8)	0 : XIN, XOUT 1 : XCIN, XCOUT	RW

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to "1" before writing to this register.
Note 2: f_{C32} is not included. Do not set to "1" when using low-speed, low power dissipation or on-chip oscillator mode.
Note 3: Changes to "1" when shifting to stop mode.
Note 4: This bit is used to stop the main clock when placing the device in a low-power mode. If you want to operate with XIN after exiting from the stop mode, set this bit to "0". When operating with a self-excited oscillator, set the system clock select bit (CM07) to "1" before setting this bit to "1".
Note 5: When inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.
Note 6: If this bit is set to "1", XOUT turns "H". The built-in feedback resistor remains being ON, so XIN turns pulled up to XOUT ("H") via the feedback resistor.
Note 7: This bit changes to "1" when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 8: Set port Xc select bit (CM04) to "1" before setting this bit to "1". Can not write to both bits at the same time.

System clock control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
				0	0	0	

Symbol: CM1 | Address: 0007_{16} | When reset: 00100000_2

Bit symbol	Bit name	Function	RW
CM10	All clock stop control bit (Note 2)	0 : Clock on 1 : All clocks off (stop mode)	RW
Reserved bit		Set to "0"	RW
CM14	On-chip oscillation stop bit	0 : Oscillation enabled 1 : Oscillation stopped (Note 3)	RW
CM15	XIN-XOUT drive capacity select bit (Note 4)	0 : LOW 1 : HIGH	RW
CM16	Main clock division select bit 1 (Note 5)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	RW
CM17			

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to "1" before writing to this register.
Note 2: The mode of power control cannot be shifted on the stop mode directly from the on-chip oscillator. If this bit is set to "1", XOUT turns "H", and the built-in feedback resistor is ineffective.
Note 3: This bit can be set to "1" only when both the main clock switch bit (CM22) and clock monitor bit (CM23) are set to "0". Moreover, this bit is automatically set to "0" if the main clock switch bit (CM22) is set to "1".
Note 4: This bit changes to "1" when shifting from high-speed/middle-speed mode to stop mode or at reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 5: Can be selected when bit 6 of the system clock control register 0 (address 0006_{16}) is "0". If "1", division mode is fixed at 8.

Figure 6.2 System clock control registers 0 and 1

Oscillation stop detection register (Note 1)

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

Symbol: CM2 | Address: $000C_{16}$ | When reset: 00000100_2

Bit symbol	Bit name	Function	RW
CM20	Oscillation stop detection bit	0 : The function is invalid 1 : The function is valid (Note 2)	RW
CM21	Oscillation stop detection interrupt enable bit	0 : Disabled 1 : Enabled (Note 3)	RW
CM22	Main clock switch bit	0 : Select XIN clock 1 : Select on-chip oscillator (Note 4)	RW
CM23	Clock monitor bit (Note 5)	0 : XIN oscillating normally 1 : XIN stopping abnormally	RO
Reserved bit		Set to "0"	RW

Note 1: In case of writing to this register, set bit 0 of the protect register (address $000A_{16}$) to "1".
Note 2: Set to "0" before stopping the oscillation of the main clock (XIN-XOUT).
(stop mode, low power dissipation mode, on-chip oscillation mode)
An oscillation stop is detected if the oscillation of the main clock (XIN-XOUT) is stopped when the following two conditions are satisfied: (1) the oscillation stop detection function is valid and (2) CM21=1.
Note 3: Valid when CM20=1.
Note 4: CM22 bit switches to "1" automatically if an oscillation stop is detected when both CM20 bit and CM 21 bit are "1". CM22 bit cannot be cleared when CM23=1.
Note 5: This bit is valid when both CM20 bit is "1". Use this bit for the purpose of confirming XIN operation for oscillation stop detection interrupt execution.

Clock prescaler reset flag

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CPSRF | Address: $008F_{16}$ | When reset: $0XXXXXXX_2$

Bit symbol	Bit name	Function	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is "0")	RW

Figure 6.3 Oscillation stop detection register and clock prescaler reset flag

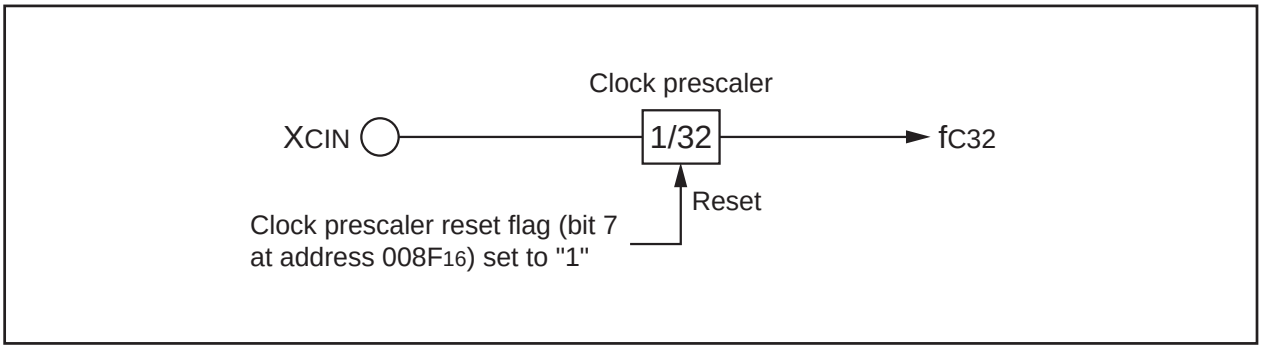


Figure 6.4 f_{C32} block diagram

CAN0 clock select register (Note 1, 2)

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol	Address	When reset
CCLKR	025F16	X000XXXX2

Bit symbol	Bit name	Function	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—
CCLK4	CAN0 clock select bit	b6 b5 b4 0 0 0 : No division mode 0 0 1 : Division by 2 mode 0 1 0 : Division by 4 mode 0 1 1 : Division by 8 mode 1 0 0 : Division by 16 mode 1 0 1 : Inhibited 1 1 0 : Inhibited 1 1 1 : Inhibited	RW
CCLK5			RW
CCLK6			RW
Reserved bit		Set to "0"	RW

Note 1: Set bit 0 of the protect register (address 000A16) to "1" before writing in this register.
Note 2: Change the register value only when the CAN module is in Reset/Initialization mode (the bit 0 of the CAN Control Register (address 023016) is "1").

Figure 6.5 CAN0 clock select register

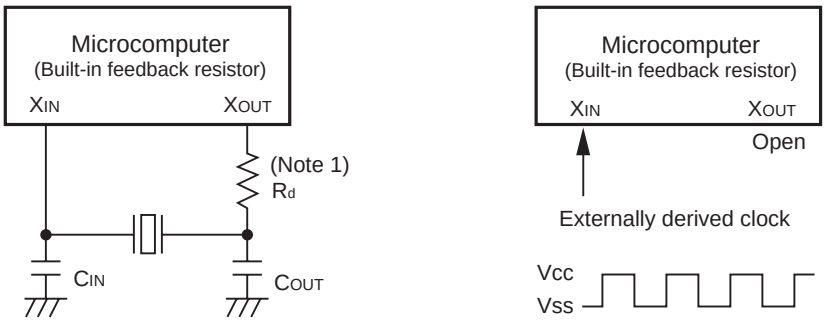
The following describes the clocks generated by the clock generation circuit.

6.1 Main Clock

The main clock is generated by the main clock oscillation circuit. After reset, oscillation starts. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). Stopping the clock reduces the power dissipation.

After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the XOUT pin can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the XOUT pin reduces the power dissipation. This bit changes to "1" when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

Figure 6.5 shows the examples of main clock connection circuit.



Note 1: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XIN and XOUT following the instruction.

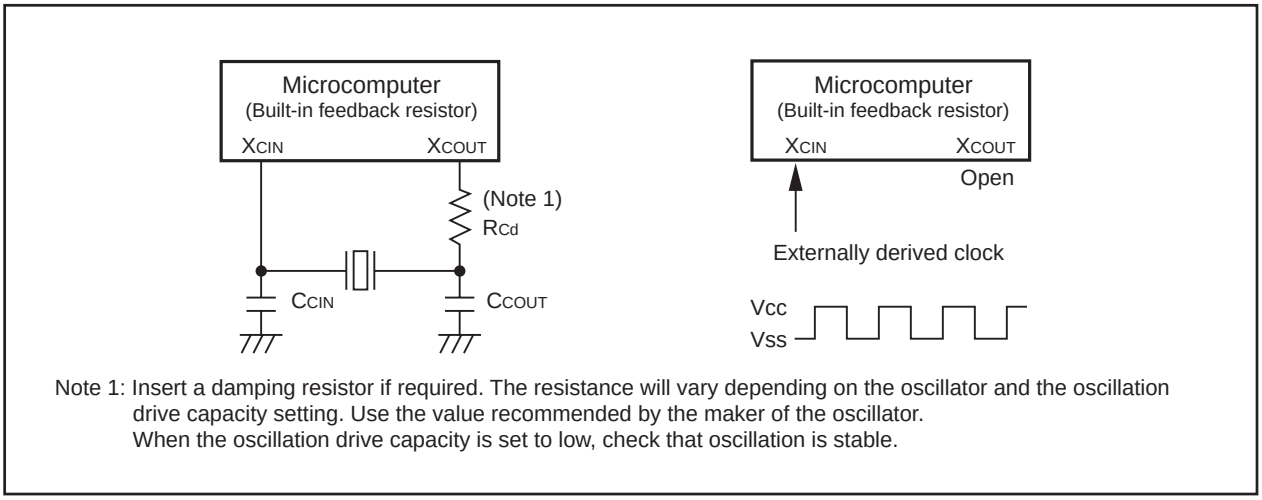
Figure 6.6 Examples of main clock connection circuit

6.2 Sub-clock

The sub-clock is generated by the sub-clock oscillation circuit. No sub-clock is generated after a reset. After oscillation is started using the port Xc select bit (bit 4 at address 000616), the sub-clock can be selected as BCLK by using the system clock select bit (bit 7 at address 000616). However, be sure that the sub-clock oscillation has fully stabilized before switching.

After the oscillation of the sub-clock oscillation circuit has stabilized, the drive capacity of the XCOUT pin can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 000616). Reducing the drive capacity of the XCOUT pin reduces the power dissipation. This bit changes to "1" when shifting to stop mode and at a reset.

Figure 6.7 shows the examples of sub-clock connection circuit.



Note 1: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator. When the oscillation drive capacity is set to low, check that oscillation is stable.

Figure 6.7 Examples of sub-clock connection circuit

6.3 On-chip Oscillator Clock

This clock by supplied by a on-chip oscillator. The oscillation of the on-ship oscillator can be used as BCLK by setting the main clock selected bit (bit 2 at address 000C16. Lower power consumption can be realized because the oscillating frequency of the on-chip oscillator is much lower compared to that of XIN. The frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range. The application products must be designed with sufficient margin to accommodate the frequency range.

6.4 CPU Clock and Peripheral Function Clock

6.4.1 BCLK

The BCLK is the clock that drives the CPU. The clock source for BCLK is as follows: (1) the clock derived by dividing the main clock by 1, 2, 4, 8, or 16, (2) fc, or (3) the clock derived by dividing the clock supplied by the on-chip oscillator circuit (f_{RING}) by 1, 2, 4, 8 or 16. After reset, the BCLK is derived by dividing the f_{RING} by 8.

The main clock division select bit 0 (bit 6 at address 0006_{16}) changes to "1" when shifting from high-speed/medium-speed mode to stop mode and at reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

6.4.2 Peripheral Function Clock

6.4.2.1 f_1, f_8, f_{32}

The clock for the peripheral devices is derived from the main clock or by dividing it by 8 or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at address 0006_{16}) to "1" and then executing a WAIT instruction.

6.4.2.2 f_{AD}

This clock has the same frequency as the main clock and is used in A/D conversion.

6.4.2.3 f_{CAN0}

This clock is derived by dividing the main clock by 1, 2, 4, 8, 16 by setting the CAN0 clock select register.

It is used for the corresponding CAN module.

This clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at address 0006_{16}) to "1" and then executing a WAIT instruction.

6.4.2.4 f_{C32}

This clock is derived by dividing the sub-clock by 32. It is used for the timer 1, timer X, timer Y and timer X counts.

6.4.2.5 f_C

This clock has the same frequency as the sub-clock. It is used for BCLK and for the watchdog timer.

6.4.3 f_{RING}

This clock is supplied by the on-chip oscillator circuit. In the on-chip oscillator mode, the clock divided by the division ratio selected with the main clock division select bit 0 and bit 1 (bit 6 at address 0006_{16}, and bit 6 and bit 7 at address 0007_{16}) is supplied as BCLK. Immediately after reset, 8 divisions of this clock is supplied as BCLK. The on-chip oscillator oscillation can be set to BCLK when oscillation stop is detected or with the main clock switching bit (bit 2 at address $000C_{16}$).

6.5 Power Control

There are three power control modes. All modes other than wait and stop modes are referred to as normal operation mode.

6.5.1 Normal Operating Modes

Normal operation mode is further separated into five modes.

In normal operation mode, the CPU clock and the peripheral function clock are supplied to operate the CPU and the peripheral function.

Power consumption control is enabled by controlling the CPU clock frequency. The higher the CPU clock frequency, the more processing power increases. The lower the CPU clock frequency, the more power consumption decreased. When unnecessary oscillator circuits stop, power consumption is further reduced.

Before the clock sources for the CPU clock can be switched over, the new clock source after switching needs to be stabilized and oscillated. If the new clock source is the main clock, allow sufficient wait time in a program until an oscillation is stabilized. When the clock source for the CPU clock is changed from the one-chip oscillator to the main clock, change the operation mode to the medium-speed mode (divided-by-8 mode) after the clock was divided by 8 in on-chip oscillator mode.

6.5.1.1 High-speed Mode

The main clock divided-by-1 (undivided) provides the CPU clock. The peripheral functions operate on the clocks specified for each respective function.

6.5.1.2 Medium-speed Mode

The main clock divided-by-2, -4, -8 or -16 provides the CPU clock. The peripheral functions operated on the clocks specified for each respective function. The main clock must be oscillating stably before transferring from the main clock divided-by-8 to divided-by-1, -2 or -4 and the sub-clock must be oscillating stably before transferring to low-speed or lower power-dissipation mode.

6.5.1.3 Low-speed Mode

The sub-clock provides the CPU clock. The peripheral functions operate on the clocks specified for each respective function. Note that oscillation of both the main and sub-clocks must have stabilized before transferring from this mode to another or vice versa. At least 2 to 3 seconds are required after the sub-clock status. Therefore, the program must be written to wait until this clock has stabilized immediately after powering up and after stop mode is cancelled.

6.5.1.4 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. Only the peripheral functions for which the sub-clock was selected as the count source continue to operate.

6.5.1.5 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. The higher division and the main clock is turned off after being placed in on-chip oscillator mode, power consumption is reduced further.

Table 6.2 lists the setting and mode of clock associated bit.

Table 6.2 Setting and mode of clock associated bit

Mode		CM2 register	CM1 register	CM0 regiser			
		CM22	CM17, CM16	CM07	CM06	CM05	CM04
High-speed mode		0	00₂	0	0	0	–
Medium-speed mode	Divide-by-2	0	01₂	0	0	0	–
	Divide-by-4	0	10₂	0	0	0	–
	Divide-by-8	0	–	0	1	0	–
	Divide-by-16	0	11₂	0	0	0	–
Low-speed mode		0	–	1	–	0	1
Low power-dissipation mode		0	–	1	–	1	1
On-chip oscillator mode	Undivided	1	00₂	0	0	–	–
	Divide-by-2	1	01₂	0	0	–	–
	Divide-by-4	1	10₂	0	0	–	–
	Divide-by-8	1	–	0	1	–	–
	Divide-by-16	1	11₂	0	0	–	–

6.5.2 Wait Mode

When a WAIT instruction is executed, BCLK stops and the microcomputer enters wait mode. In this mode, oscillation continues but BCLK and watchdog timer stop. Writing "1" to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. Table 6.3 lists the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or an interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts using as BCLK, the clock that had been selected when the WAIT instruction was executed.

Table 6.3 Port status during wait mode

Pin	States
Port	Retains status before wait mode

6.5.3 Stop Mode

Writing "1" to the all-clock stop control bit (bit 0 at address 0007_{16}) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that V_{CC} remains above 2V.

Because the oscillation of BCLK, f_1 to f_{32}, fc, fc_{32}, f_{AD} and f_{CAN0} stop in stop mode, peripheral functions such as the A/D converter and watchdog timer do not function. However, timer X operate provided that the event counter mode is set to an external pulse, and UART0 and UART1 function provided an external clock is selected. Table 6.4 lists the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or an interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled, and the priority level of the interrupt which is not used to cancel must have been changed to 0 before shifting to stop mode. If returning by an interrupt, that interrupt routine is executed. If only a hardware reset is used to cancel stop mode, change the priority level of all interrupt to 0, then shift to stop mode.

When shifting from high-speed/medium-speed mode to stop mode or at a reset, the main clock division select bit 0 (bit 6 at address 0006_{16}) is set to "1". When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

Stop mode must not be use while operating in on-chip oscillator mode.

Table 6.4 Port status during stop mode

Pin	States
Port	Retains status before stop mode

Figure 6.8 shows the state transition to stop and wait modes. Figure 6.9 shows the state transition in normal operation mode.

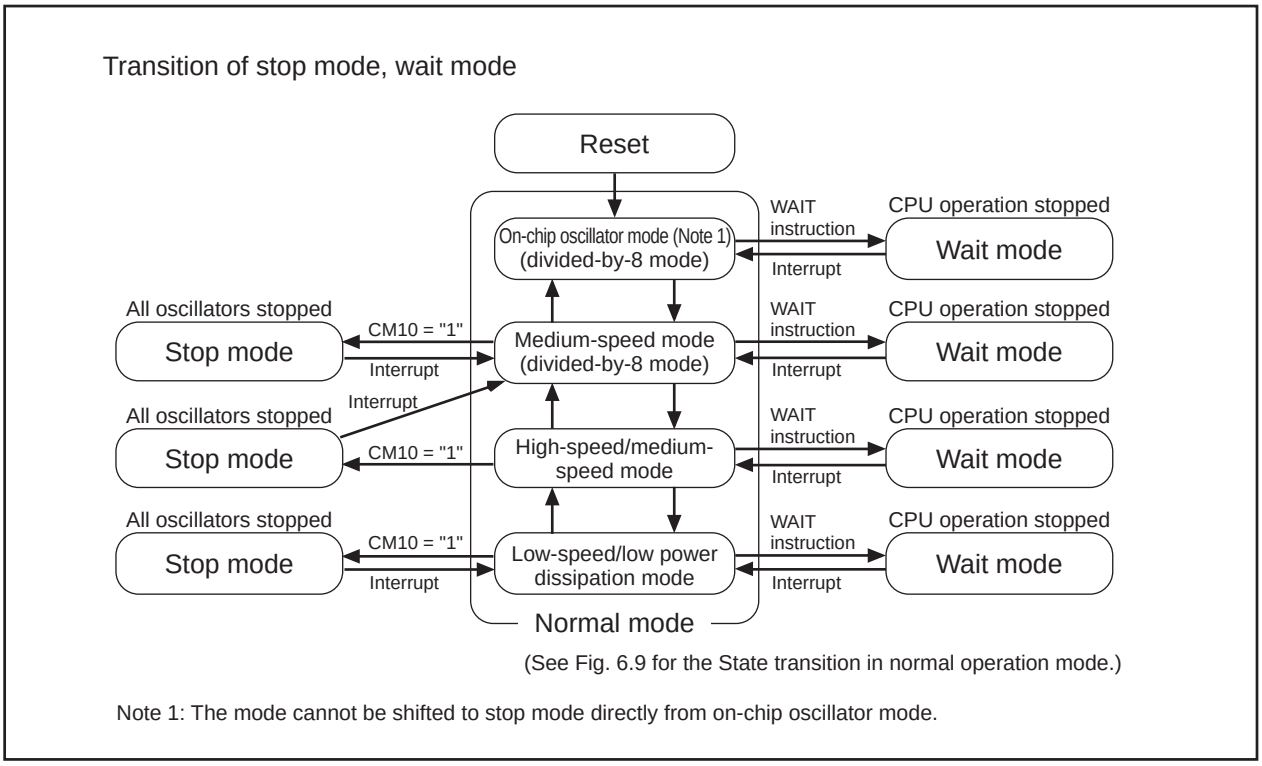


Figure 6.8 State transition of stop and wait modes

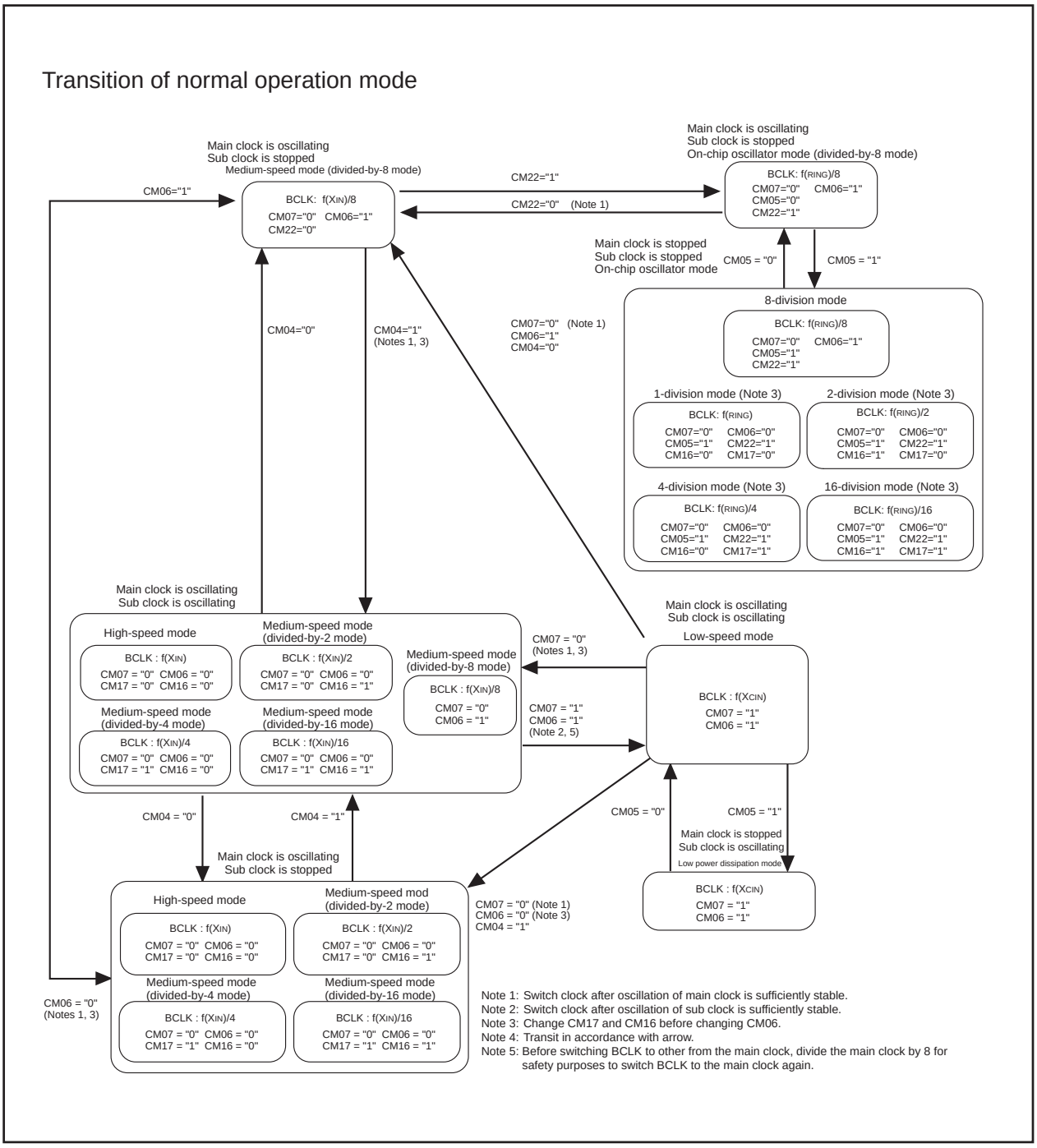


Figure 6.9 State transition in normal operation mode

6.6 Oscillation Stop Detection Function

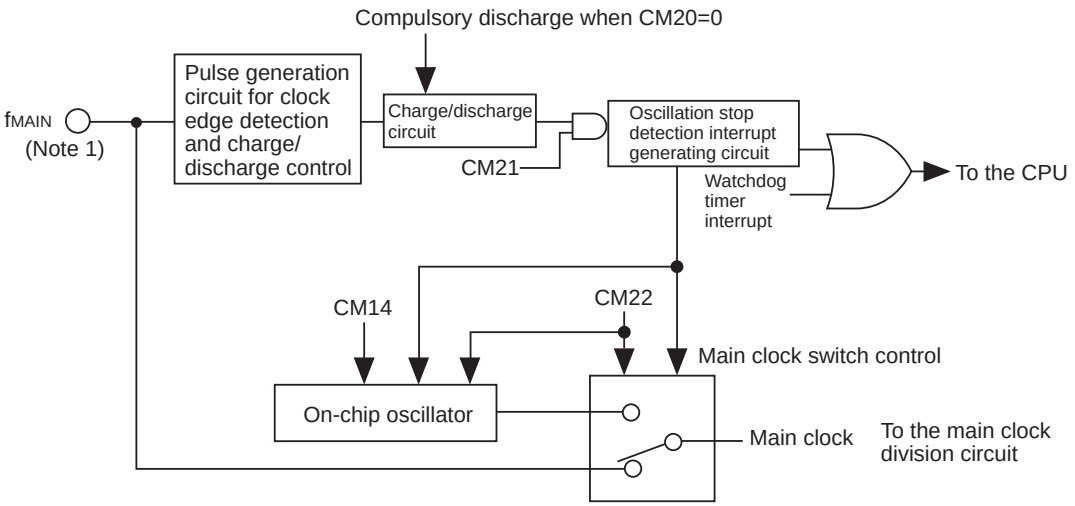
The oscillation stop detection function detects abnormal stopping of the main clock by causes such as opening and shorting of the XIN oscillation circuit. When oscillation stop is detected, an oscillation stop detection interrupt is issued. When an oscillation stop detection interrupt is issued, the on-chip oscillator in the microcomputer operates automatically and is used as the main clock in place of the XIN clock. This allows interrupt processing.

The oscillation stop detection function can be enabled/disabled with bit 0 and bit 1 of the oscillation stop detection register. When this bit is set to "11$_2$", the function is enabled. After the reset is released, the oscillation stop detection function becomes disabled because the bit value is "00$_2$".

Table 6.5 lists the specification of oscillation stop detection function, Figure 6.10 shows a configuration diagram of the oscillation stop detection circuit and Figure 6.11 shows the configuration of the oscillation stop detection register.

Table 6.5 Specification overview of the oscillation stop detection function

Item	Specification
Oscillation stop detectable clock and frequency bandwidth	XIN ≥ 2 MHz
Enabling condition for oscillation stop detection function	When the oscillation stop detection bit (bit 0 at address 000C$_{16}$) and the oscillation stop detection interrupt enable bit (bit 1 at address 000C$_{16}$) are set to "1"
Operation at oscillation stop detection	• Oscillation stop detection interrupt occurs
Notes on STOP mode, low power dissipation mode, and on-chip oscillator mode	Before stopping the main clock (XIN-XOUT), set the oscillation stop detection enable bit to "0" to disable the oscillation stop detection function. Enable main clock (XIN-XOUT) oscillation and after the oscillation stabilizes, set the bit to "1" again.
Notes on WAIT mode	If the peripheral function clock is stopped in WAIT mode with the WAIT mode peripheral function clock stop bit (bit 2 at address 0006$_{16}$), oscillation stop will be detected. Do not stop the peripheral function clock in WAIT mode.



#: When XIN is supplied, this repeats charge and discharge with pulses by XIN edge detection. When XIN is not supplied, this continues charging. When the charge exceeds a certain level, it regards the oscillation as stopped.

Note 1: As for the fMAIN, see Figure 6.1 clock generating circuit.

Figure 6.10 Oscillation stop detection circuit

Oscillation stop detection register (Note 1)

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

Symbol	Address	When reset
CM2	$000C_{16}$	04_{16}

Bit symbol	Bit name	Function	RW
CM20	Oscillation stop detection bit	0 : The function is invalid 1 : The function is valid (Note 2)	RW
CM21	Oscillation stop detection interrupt enable bit	0 : Disabled 1 : Enabled (Note 3)	RW
CM22	Main clock switch bit	0 : Select XIN clock 1 : Select on-chip oscillator (Note 4)	RW
CM23	Clock monitor bit (Note 5)	0 : XIN oscillating normally 1 : XIN stopping abnormally	RO
Reserved bit		Set to "0"	RW

Note 1: In case of writing to this register, set bit 0 of the protect register (address $000A_{16}$) to "1".
Note 2: Set to "0" before stopping the oscillation of the main clock (XIN-XOUT).
(stop mode, low power dissipation mode, on-chip oscillation mode)
An oscillation stop is detected if the oscillation of the main clock (XIN-XOUT) is stopped when the following two conditions are satisfied: (1) the oscillation stop detection function is valid and (2) CM21=1.
Note 3: Valid when CM20=1.
Note 4: CM22 bit switches to "1" automatically if an oscillation stop is detected when both CM20 bit and CM 21 bit are "1". CM22 bit cannot be cleared when CM23=1.
Note 5: This bit is valid when both CM20 bit is "1". Use this bit for the purpose of confirming XIN operation for oscillation stop detection interrupt execution.

Figure 6.11 Oscillation stop detection register

6.6.1 Oscillation Stop Detection Bit (CM20)

You can start the oscillation stop detection by setting this bit to "1" and CM21=1 (oscillation stop detection interrupt enabled). The detection is not executed when this bit is set to "0" or in reset status. Be sure to set this bit to "0" before setting for the stop-mode. Set this bit again to "1" after release from stop-mode. Set this bit to "0" also before setting the main clock stop bit (bit 5 at address 0006_{16}) to "1". Do not set this bit to "1" if the frequency of XIN is lower than 2 MHz.

An oscillation stop is detected if CM02="1" (peripheral function clock has been set for stop in wait mode) and the mode is shifted to wait.

6.6.2 Oscillation Stop Detection Interrupt Enable Bit (CM21)

When CM20=1 and CM21=1, an oscillation stop detection interrupt is generated if an abnormal stop of XIN is detected. The on-chip oscillator starts operation instead of the XIN clock which stopped abnormally. The operation goes further with the main clock supplied from the on-chip oscillator. For the oscillation stop detection interrupt, judgment on the interrupt condition is necessary, because this interrupt shares the vector table with watchdog timer interrupt. Figure 6.12 shows flow of the judgment with oscillation stop detection interrupt processing program.

6.6.3 Main Clock Switch Bit (CM22)

When setting this bit to "1", the on-chip oscillator is selected as main clock. At this time, the on-chip oscillator starts simultaneously if it has been stopped (CM14=1). This bit is cleared only when CM23 is "0" (when XIN is oscillating).

If an oscillation stop is detected while both CM20 and CM21 are "1", this bit automatically switches to "1".

When this bit is set to "1", the on-chip oscillation stop bit (bit 4 at address 0007_{16}) is automatically set to "0".

6.6.4 Clock Monitor Bit (CM23)

You can see the operation status of the XIN clock. When this bit is "0", XIN is operating correctly. You can check the oscillation status of XIN when an oscillation stop detection interrupt is generated or after reset.

When oscillation stop detection is invalid (CM20="0"), the clock monitor bit is "0".

Figure 6.12 shows the flow of interrupt cause determination of oscillation stop detection or watchdog timer interrupt.

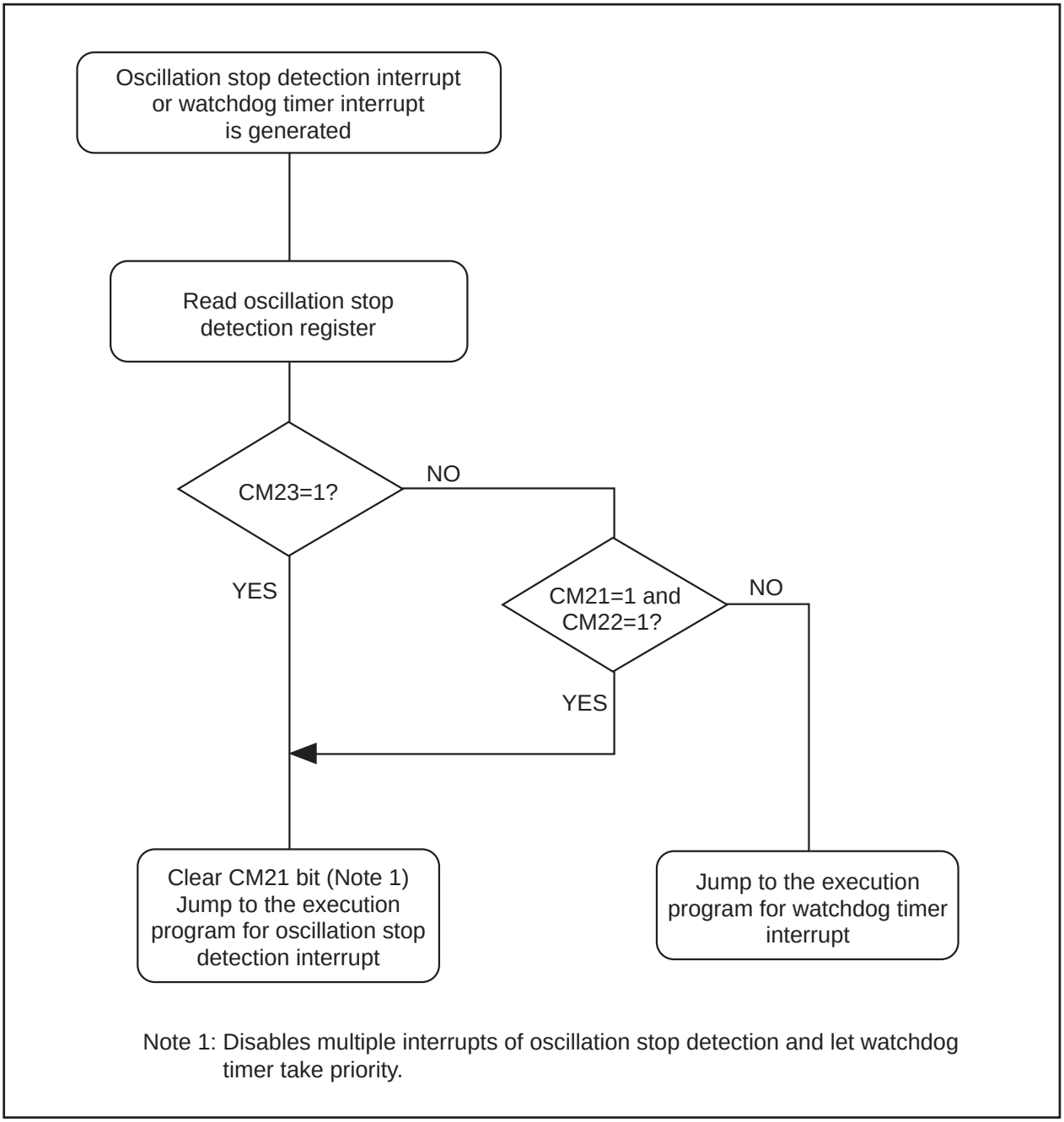


Figure 6.12 Flow of interrupt cause determination

7. Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Figure 7.1 shows the protect register. The values in the processor mode register 0 (address 0004_{16}), processor mode register 1 (address 0005_{16}), system clock control register 0 (address 0006_{16}), system clock control register 1 (address 0007_{16}) and port P0 direction register (address $00E2_{16}$) can only be changed when the respective bit in the protect register is set to "1". Therefore, important outputs can be allocated to port P0.

If, after "1" (write-enabled) has been written to bit "enables writing to port P0 direction register" (bit 2 at address $000A_{16}$), a value is written to any address, the bit automatically reverts to "0" (write-inhibited). Make sure no interrupts will generate between the instruction in which the PRC2 bit to "1" and the next instruction. The system clock control registers 0 and 1 and oscillation stop detection register write-enable bit (bit 0 at address $000A_{16}$) and processor mode register 0 and 1 write-enable bit (bit 1 at address $000A_{16}$) do not automatically return to "0" after a value has been written to an address. The program must therefore be written to return these bits to "0".

Protect register

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

Symbol	Address	When reset
PRCR	$000A_{16}$	$XXXXX000_2$

Bit symbol	Bit name	Function	RW
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16}), oscillation stop detection register (address $000C_{16}$) and CAN0 clock select register (address $025F_{16}$)	0 : Write-inhibited 1 : Write-enabled	RW
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	RW
PRC2	Enables writing to port P0 direction register (address $00E2_{16}$) and CAN0 I/O port select register (address $00F8_{16}$) (Note 1)	0 : Write-inhibited 1 : Write-enabled	RW
Reserved bit		Set to "0"	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—

Note 1: Writing a value to an address after "1" is written to this bit returns the bit to "0".
Other bits do not automatically return to "0" and they must therefore be reset by the program.

Figure 7.1 Protect register

8. Processor Mode

8.1 Types of Processor Mode

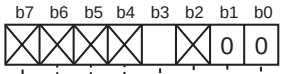
The processor mode is single-chip mode. Table 8.1 lists features of processor mode.

Figure 8.1 shows the processor mode register 0 and 1.

Table 8.1 Features of processor mode

Processor mode	Access space	Pins to which I/O ports are assigned
Single chip mode	SFR, Internal RAM, Internal ROM	All pins are I/O ports or peripheral function I/O pins.

Processor mode register 0 (Note 1)

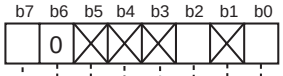


Symbol: PM0 | Address: 0004_{16} | When reset: $XXXX0X00_2$

Bit symbol	Bit name	Function	RW
Reserved bit		Set to "0"	RW
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			—
PM03	Software reset bit	The device is reset when this bit is set to "1". The value of this bit is "0" when read.	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to "1" when writing new values to this register.

Processor mode register 1 (Note 1)



Symbol: PM1 | Address: 0005_{16} | When reset: $00XXX0X0_2$

Bit symbol	Bit name	Function	RW
PM10	DATAROM area access bit (Note 2)	0 : Disabled 1 : Enabled	RW
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			—
PM12	WDT interrupt/reset switching bit	0 : Watchdog timer interrupt 1 : Reset (Note 3)	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—
Reserved bit		Set to "0"	RW
PM17	Wait bit	0 : No wait 1 : Wait	RW

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to "1" when writing new values to this register.
Note 2: This bit is valid for the flash memory version. For the mask ROM version, this bit must be set to "0".
Note 3: After setting this bit to "1", can not change to "0" by software.

Figure 8.1 Processor mode register 0 and 1

9. Bus Control

During access, the memory areas (ROM, RAM, FLASH, etc.) and the SFR area have different bus cycles. The memory areas can be accessed in one cycle of the CPU operation clock BCLK. The SFR area can be accessed in two cycles of BCLK.

Software wait states can be inserted to the memory areas by using the PM17 bit of the processor mode register 1 (bit 7 at address 0005_{16}) (Note 1). When the PM17 bit is set to "0", the memory areas are accessed in one cycle of BCLK. When the PM17 bit is set to "1", the memory areas are accessed in two cycles of BCLK. The PM17 bit is "0" after the reset status is cancelled. The SFR area is not influenced by the PM17 bit and is always accessed in two cycles of BCLK.

The Table 9.1 lists bus cycle for access areas. Figure 9.1 shows SFR area and memory areas.

Note 1: When rewriting the processor mode register 1, set the PRC1 bit of the protect register (bit 1 at address $000A_{16}$) to "1".

Table 9.1 Bus cycle for access areas

Area	PM17	Bus cycle
SFR	——	2 BCLK cycles
Internal ROM/RAM	0	1 BCLK cycle
	1	2 BCLK cycles

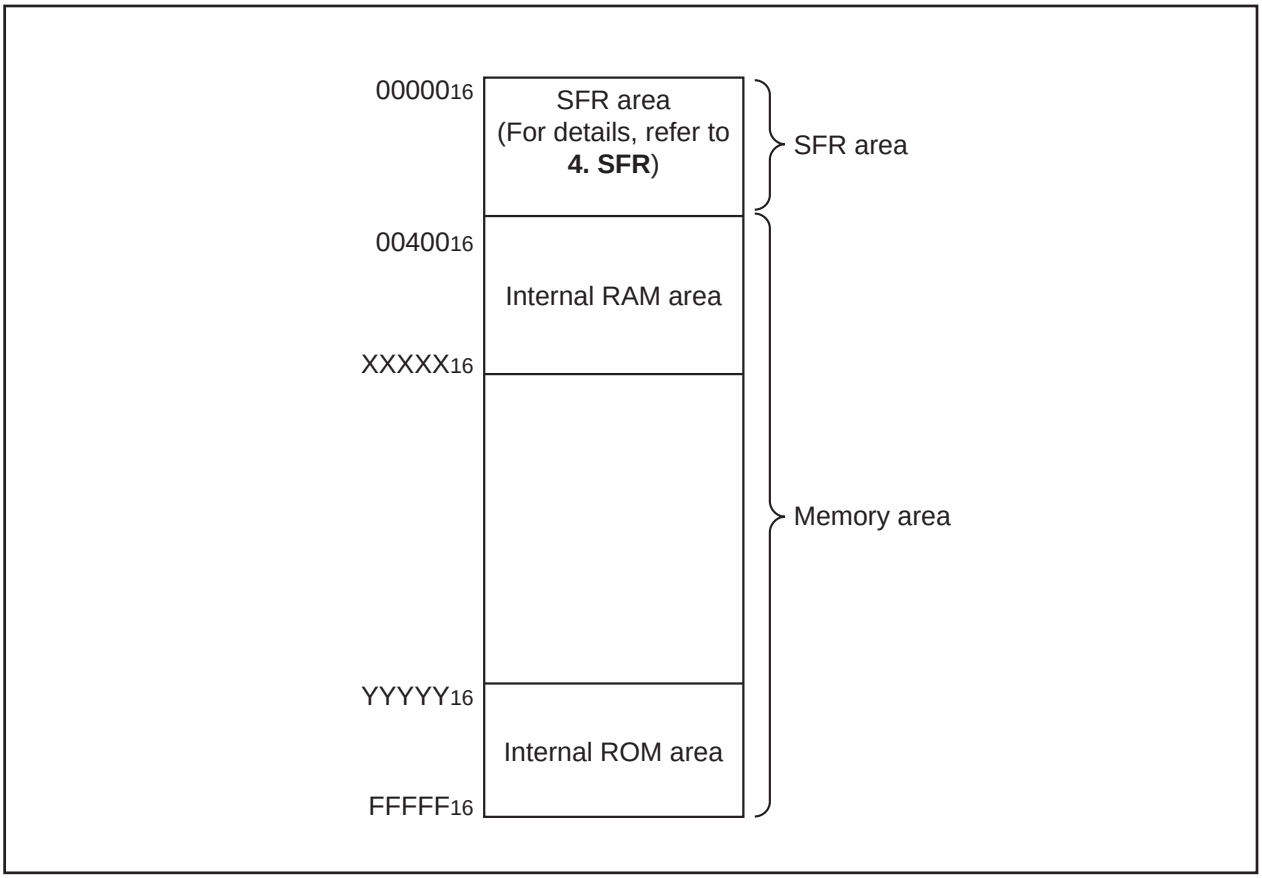
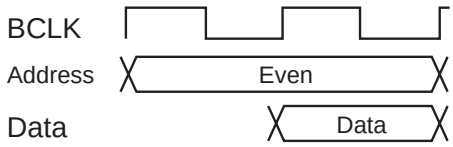
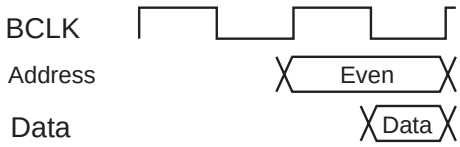
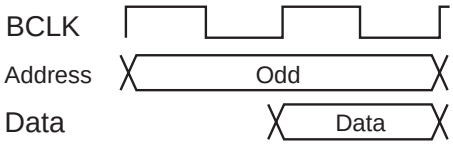
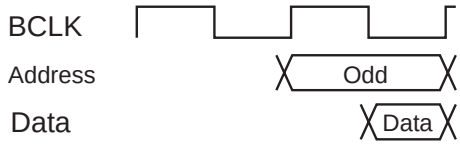
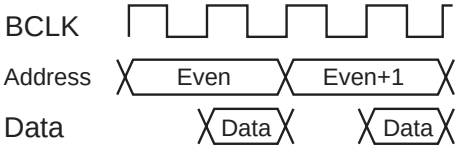
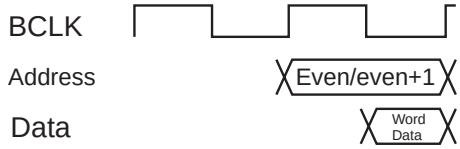
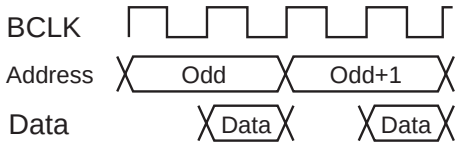
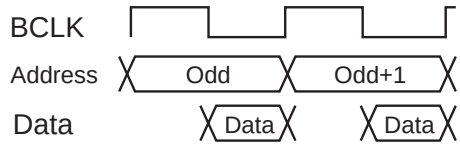


Figure 9.1 SFR area and memory areas

The memory areas and the SFR area also have different bus widths. The memory areas have a 16-bit bus width, while the SFR area has an 8-bit bus width. Consequently, different operations are used when the areas are accessed in word (16 bits) units.
Table 9.2 lists access unit and bus operation.

Table 9.2 Access unit and bus operation

Space	SFR	ROM/RAM (No wait setting)
Even address byte access		
Odd address byte access		
Even address word access		
Odd address word access		

10. Interrupt

10.1 Overview of Interrupt

10.1.1 Type of Interrupts

Figure 10.1 lists the types of interrupts.

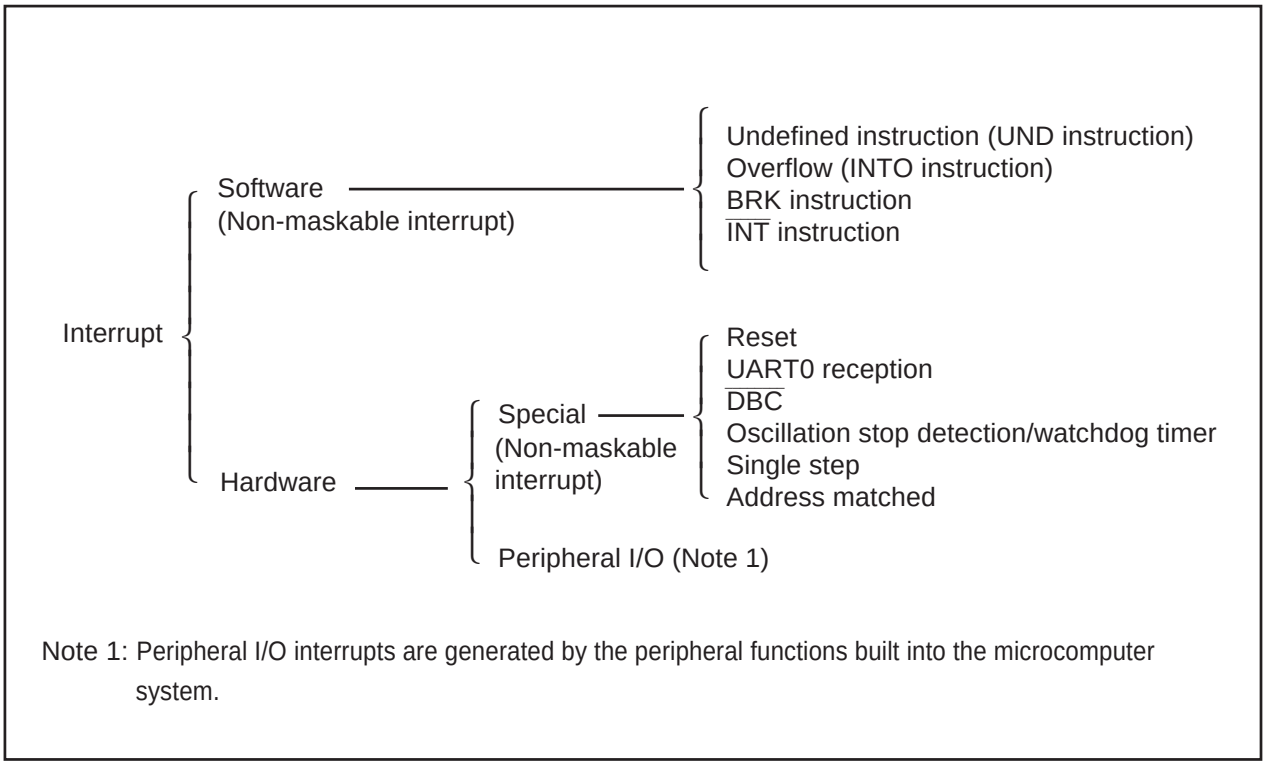


Figure 10.1 Classification of interrupts

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

10.1.2 Software Interrupts

A software interrupt is generated when an instruction is executed. The software interrupts are non-maskable interrupts.

10.1.2.1 Undefined Instruction Interrupt

The undefined instruction interrupt is generated when the UND instruction is executed.

10.1.2.2 Overflow Interrupt

The overflow interrupt is generated when the INTO instruction is executed with the overflow flag (O flag) set to "1". The following are instructions whose O flag changes by arithmetic:

ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

10.1.2.3 BRK Interrupt

A BRK interrupt is generated when the BRK instruction is executed.

10.1.2.4 $\overline{\text{INT}}$ instruction Interrupt

An $\overline{\text{INT}}$ instruction interrupt is generated when the $\overline{\text{INT}}$ instruction is executed. The $\overline{\text{INT}}$ instruction can select the software interrupt numbers 0 to 63. The software interrupt numbers 0 to 31 are assigned to the peripheral function interrupt. Therefore, the microcomputer executes the same interrupt routine when the $\overline{\text{INT}}$ instruction interrupt is executed as when a peripheral function interrupt is generated.

The stack pointer (SP) used for the $\overline{\text{INT}}$ instruction interrupt is dependent on which software interrupt number is involved.

So far as software interrupt numbers 0 to 31 are concerned, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. If change the U flag to "0" and select the interrupt stack pointer (ISP), and then execute an interrupt sequence. When returning from the interrupt routine, the U flag is returned to the state it was before the acceptance of interrupt request. So far as software numbers 32 to 63 are concerned, the stack pointer does not make a shift.

10.1.3 Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral I/O interrupts.

10.1.3.1 Special Interrupts

Special interrupts are non-maskable interrupts.

- **Reset**

  Reset occurs if an "L" is input to the $\overline{\text{RESET}}$ pin.

- **UART0 reception interrupt**

  UART0 reception interrupt occurs when UART0 is received. This interrupt can be enabled with bit 2 of the INT_0 input filter select register (address $001E_{16}$).

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

- **$\overline{\text{DBC}}$ interrupt**

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

- **Oscillation stop detection/watchdog timer interrupt**

  Generated by the oscillation stop detection or watchdog timer.

- **Single-step interrupt**

  This interrupt is exclusively for the debugger, do not use it in other circumstances. With the debug flag (D flag) set to "1", a single-step interrupt occurs after one instruction is executed.

- **Address match interrupt**

  An address match interrupt occurs immediately before the instruction held in the address indicated by the address match interrupt register is executed with the address match interrupt enable bit set to "1".

  If an address other than the first address of the instruction in the address match interrupt register is set, no address match interrupt occurs.

10.1.3.2 Peripheral I/O Interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. The interrupt vector table is the same as the one for software interrupt numbers 0 through 31 the $\overline{\text{INT}}$ instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **CAN0 error interrupt**

  Tis is an interrupt that CAN error generates.

- **CAN0 wake up interrupt**

  CAN0 wake up interrupt occurs if a falling edge is input to the CRx pin.

- **CAN0 successful reception interrupt**

  This is an interrupt that the CAN reception generates.

- **CAN0 successful transmission interrupt**

  This is an interrupt that the CAN transmission generates.

- **Key-input interrupt**

  A key-input interrupt occurs if a falling or rising edge is input to the $\overline{\text{KI}}$ pin.

- **A/D conversion interrupt**

  This is an interrupt that the A/D converter generates.

- **UART0 and UART1 transmission interrupt**

  These are interrupts that the serial I/O transmission generates.

- **UART0 and UART1 reception interrupt**

  These are interrupts that the serial I/O reception generates.

- **Timer 1 interrupt**

  This is an interrupt that timer 1 generates.

- **Timer X interrupt**
  This is an interrupt that timer X generates.
- **Timer Y interrupt**
  This is an interrupt that timer Y generates.
- **Timer Z interrupt**
  This is an interrupt that timer Z generates.
- **Timer C interrupt**
  This is an interrupt that timer C generates.
- **CNTR0 interrupt**
  This interrupt occurs if either a falling edge or a rising edge is input to the CNTR0 pin.
- **TCIN interrupt**
  This interrupt occurs if any one of a falling edge, a rising edge or both edges is input to the TCIN pin. This interrupt also occurs with the $f_{RING256}$.
- **INT0 to INT3 interrupt**
  INT_0 to INT_2 interrupts occur if any one of a falling edge, a rising edge or both edges is input to the $\overline{INT}$ pin.
  INT_3 interrupt occurs if either a falling edge or both edges is input to the $\overline{INT}$ pin.

10.1.4 Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Figure 10.2 shows format for specifying interrupt vector addresses.

Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.

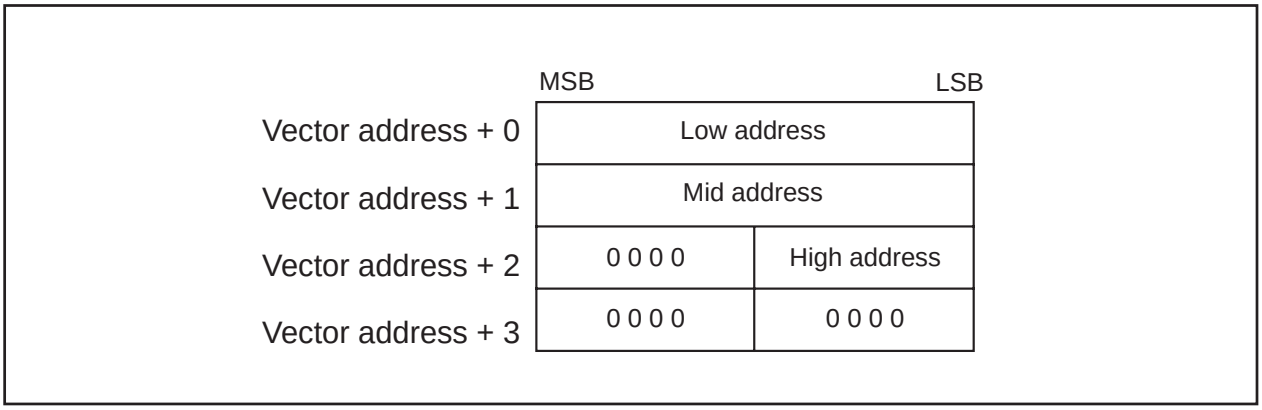


Figure 10.2 Format for specifying interrupt vector addresses

10.1.4.1 Fixed Vector Tables

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFDC$_{16}$ to FFFFF$_{16}$. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 10.1 lists the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table 10.1 Interrupt and fixed vector address

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFDC$_{16}$ to FFFDF$_{16}$	Interrupt on UND instruction
Overflow	FFFE0$_{16}$ to FFFE3$_{16}$	Interrupt on INTO instruction
BRK instruction	FFFE4$_{16}$ to FFFE7$_{16}$	If the vector is filled with FF$_{16}$, program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE8$_{16}$ to FFFEB$_{16}$	There is an address-matching interrupt enable bit
Single step (Note 1)	FFFEC$_{16}$ to FFFEF$_{16}$	Do not use
Oscillation stop detection/ Watchdog timer	FFFF0$_{16}$ to FFFF3$_{16}$	
$\overline{\text{DBC}}$ (Note 1)	FFFF4$_{16}$ to FFFF7$_{16}$	Do not use
UART0 reception (Note 1)	FFFF8$_{16}$ to FFFFB$_{16}$	Do not use
Reset	FFFFC$_{16}$ to FFFFF$_{16}$	

Note 1: Interrupts used for debugging purposes only.

10.1.4.2 Variable Vector Tables

The addresses in the variable vector table can be modified, according to the user's settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table 10.2 lists the interrupts assigned to the variable vector tables and addresses of vector tables.

Table 10.2 Interrupt causes (variable interrupt vector addresses)

Software interrupt number	Vector table address (Note 1) Address (L) to address (H)	Interrupt source	Remarks
0	+0 to +3	BRK instruction	Cannot be masked by I flag
1	+4 to +7	—	
2	+8 to +11	—	
3	+12 to +15	—	
4	+16 to +19	—	
5	+20 to +23	CAN0 wake up	
6	+24 to +27	CAN0 error	
7	+28 to +31	—	
8	+32 to +35	CAN0 successful reception	
9	+36 to +39	CAN0 successful transmission	
10	+40 to +43	—	
11	+44 to +47	—	
12	+48 to +51	—	
13	+52 to +55	Key input	
14	+56 to +59	A/D	
15	+60 to +63	—	
16	+64 to +67	—	
17	+68 to +71	UART0 transmission	
18	+72 to +75	UART0 reception	
19	+76 to +79	UART1 transmission	
20	+80 to +83	UART1 reception	
21	+84 to +87	Timer 1	
22	+88 to +91	Timer X	
23	+92 to +95	Timer Y	
24	+96 to +99	Timer Z	
25	+100 to +103	$CNTR_0$	
26	+104 to +107	TCIN	
27	+108 to +111	Timer C	
28	+112 to +115	$\overline{INT_3}$	
29	+116 to +119	$\overline{INT_0}$	
30	+120 to +123	$\overline{INT_1}$	
31	+124 to +127	$\overline{INT_2}$	
32 to 63	+128 to +131 to +252 to +255	software interrupt	Cannot be masked by I flag

Note 1: Address relative to address in interrupt table register (INTB).

10.1.5 Interrupt Control

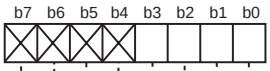
Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

Enable or disable a maskable interrupt using the interrupt enable flag (I flag), interrupt priority level select bit, and processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt priority level selection bit are located in the interrupt control register of each interrupt. Also, the interrupt enable flag (I flag) and the IPL are located in the flag register (FLG).

Figure 10.3 shows the interrupt control registers.

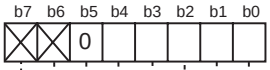
Interrupt control register

Symbol	Address	When reset
C01WKIC	0045_{16}	$XXXXX000_2$
C01ERRIC	0046_{16}	$XXXXX000_2$
C0RECIC	0048_{16}	$XXXXX000_2$
C0TRMIC	0049_{16}	$XXXXX000_2$
KUPIC	$004D_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
SiTIC(i=0,1)	0051_{16}, 0053_{16}	$XXXXX000_2$
SiRIC(i=0,1)	0052_{16}, 0054_{16}	$XXXXX000_2$
T1IC	0055_{16}	$XXXXX000_2$
TXIC	0056_{16}	$XXXXX000_2$
TYIC	0057_{16}	$XXXXX000_2$
TZIC	0058_{16}	$XXXXX000_2$
CNTR0IC	0059_{16}	$XXXXX000_2$
TCINIC	$005A_{16}$	$XXXXX000_2$
TCIC	$005B_{16}$	$XXXXX000_2$
INT3IC	$005C_{16}$	$XXXXX000_2$



Bit symbol	Bit name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	RW (Note 1)
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—

Note 1: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).
Note 2: To rewrite the interrupt control register, do so at a point that dose not generate the interrupt request for that register. For details, see 10.7 the precautions for interrupts.



Symbol	Address	When reset
INTiIC(i=0, 1, 2)	$005D_{16}$, $005E_{16}$, $005F_{16}$	$XX00X000_2$ $XX00X000_2$

Bit symbol	Bit name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	RW (Note 1)
POL	Polarity select bit	0 : Selects falling edge 1 : Selects rising edge	RW
Reserved bit		Set to "0"	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—

Note 1: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).
Note 2: To rewrite the interrupt control register, do so at a point that dose not generate the interrupt request for that register. For details, see the precautions for interrupts.

Figure 10.3 Interrupt control register

10.1.5.1 Interrupt Enable Flag (I flag)

The interrupt enable flag (I flag) controls the enabling and disabling of maskable interrupts. Setting this flag to "1" enables all maskable interrupts; setting it to "0" disables all maskable interrupts. This flag is set to "0" after reset.

10.1.5.2 Interrupt Request Bit

The interrupt request bit is set to "1" by hardware when an interrupt is requested. After the interrupt is accepted and jumps to the corresponding interrupt vector, the request bit is set to "0" by hardware. The interrupt request bit can also be set to "0" by software (Do not set this bit to "1").

10.1.5.3 Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to "0" disables the interrupt. Table 10.3 lists the settings of interrupt priority levels and Table 10.4 lists the interrupt levels enabled, according to the consist of the IPL.

The following are conditions under which an interrupt is accepted:

- interrupt enable flag (I flag) = 1
- interrupt request bit = 1
- interrupt priority level > IPL

The interrupt enable flag (I flag), the interrupt request bit, the interrupt priority select bit, and the IPL are independent, and they are not affected by one another.

Table 10.3 Settings of interrupt priority levels

Interrupt priority level select bit (b2 b1 b0)	Interrupt priority level	Priority order
0 0 0	Level 0 (interrupt disabled)	———
0 0 1	Level 1	Low
0 1 0	Level 2	↓
0 1 1	Level 3	↓
1 0 0	Level 4	↓
1 0 1	Level 5	↓
1 1 0	Level 6	↓
1 1 1	Level 7	High

Table 10.4 Interrupt levels enabled according to the contents of the IPL

IPL (IPL_2 IPL_1 IPL_0)	Enabled interrupt priority levels
0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

10.1.5.4 Rewrite the Interrupt Control Register

To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

Example 1:

```
INT_SWITCH1:
    FCLR    I           ; Disable interrupts.
    AND.B   #00h, 0055h ; Clear T1IC int. priority level and int. request bit.
    NOP                 ;
    NOP
    FSET    I           ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
    FCLR    I           ; Disable interrupts.
    AND.B   #00h, 0055h ; Clear T1IC int. priority level and int. request bit.
    MOV.W   MEM, R0     ; Dummy read.
    FSET    I           ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
    PUSHC   FLG         ; Push Flag register onto stack
    FCLR    I           ; Disable interrupts.
    AND.B   #00h, 0055h ; Clear T1IC int. priority level and int. request bit.
    POPC    FLG         ; Enable interrupts.
```

The reason why two NOP instructions or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.
Instructions : AND, OR, BCLR, BSET

Changing the interrupt request bit
When attempting to clear the interrupt request bit of an interrupt control register, the interrupt request bit is not cleared sometimes. This will depend on the instruction. If this creates problems, use the below instructions to change the register.
Instructions : MOV

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

In the interrupt sequence, the processor carries out the following in sequence given:

(1) CPUgets the interrupt information (the interrupt number and interrupt request level) by reading address 00000_{16}. After this, the corresponding interrupt request bit becomes "0".

(2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note 1) within the CPU.

(3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to "0" (the U flag, however, does not change if the $\overline{\text{INT}}$ instruction, in software interrupt numbers 32 through 63, is executed).

(4) Saves the content of the temporary register (Note 1) within the CPU in the stack area.

(5) Saves the content of the program counter (PC) in the stack area.

(6) Sets the interrupt priority level of the accepted instruction in the IPL.

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

Note 1: This register cannot be utilized by the user.

Figure 10.4 shows the time required for executing interrupt sequence.

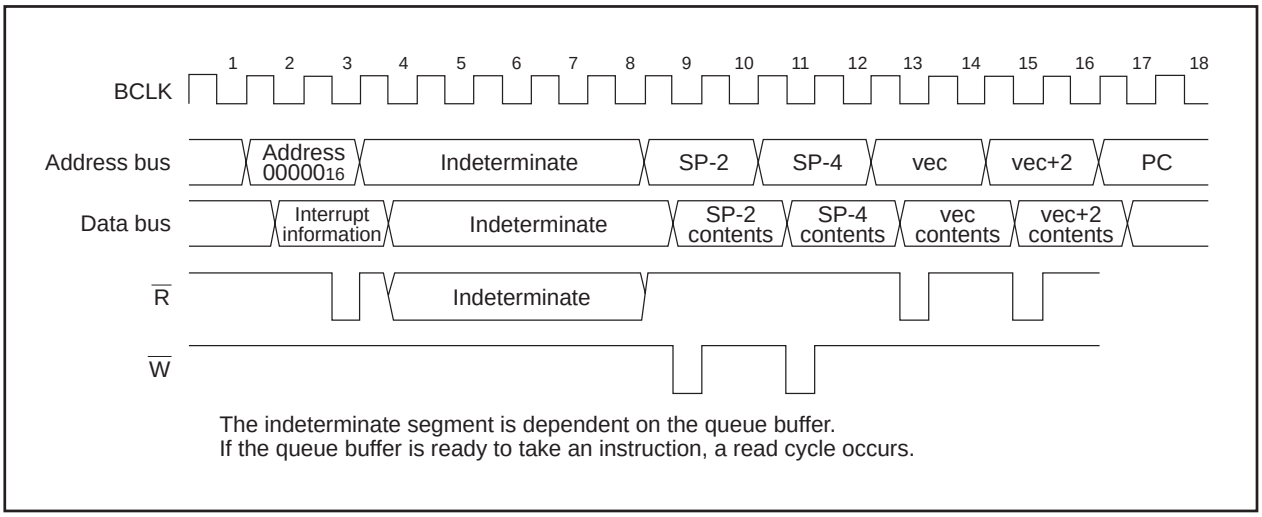
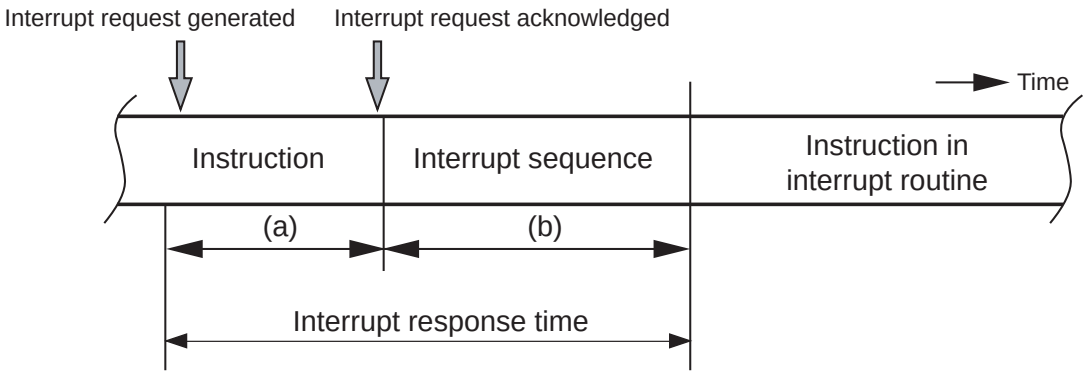


Figure 10.4 Time required for executing interrupt sequence

10.1.5.6 Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure 10.5 shows the interrupt response time.



(a) A time from when an interrupt request is generated till when the instruction then executing is completed. The length of this time varies with the instruction being executed. The DIVX instruction requires the longest time, which is equal to 30 cycles (without wait state, the divisor being a register).

(b) A time during which the interrupt sequence is executed. For details, see the table below. Note, however, that the values in this table must be increased 2 cycles for the $\overline{\text{DBC}}$ interrupt and 1 cycle for the address match and single-step interrupts.
Locate an interrupt vector address in an even address, if possible.

Interrupt vector address	Stack pointer (SP) value	Without wait
Even	Even	18 cycles
Even	Odd	19 cycles
Odd	Even	19 cycles
Odd	Odd	20 cycles

Figure 10.5 Interrupt response time

10.1.5.7 Variation of IPL when Interrupt Request is Accepted

If an interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL. If an interrupt request, that does not have an interrupt priority level, is accepted, one of the values shown in Table 10.6 is set in the IPL.

Table 10.6 Relationship between interrupts without interrupt priority levels and IPL

Interrupt sources without priority levels	Value set in the IPL
Watchdog timer	7
Reset	0
Other	Not changed

10.1.5.8 Saving Registers

In the interrupt sequence, only the contents of the flag register (FLG) and that of the program counter (PC) are saved in the stack area.

First, the processor saves the 4 high-order bits of the program counter, and 4 high-order bits and 8 low-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 low-order bits of the program counter. Figure 10.6 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

Save other necessary registers at the beginning of the interrupt routine using software. Using the PUSHM instruction alone can save all the registers except the stack pointer (SP).

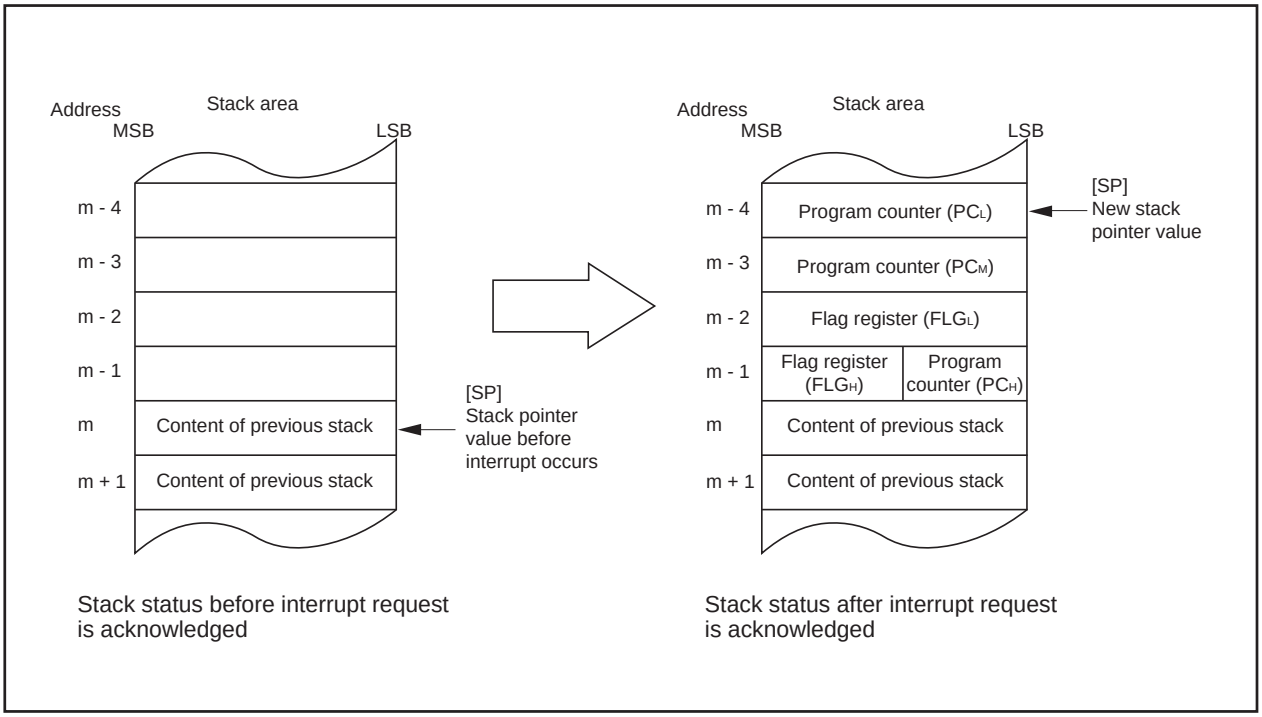
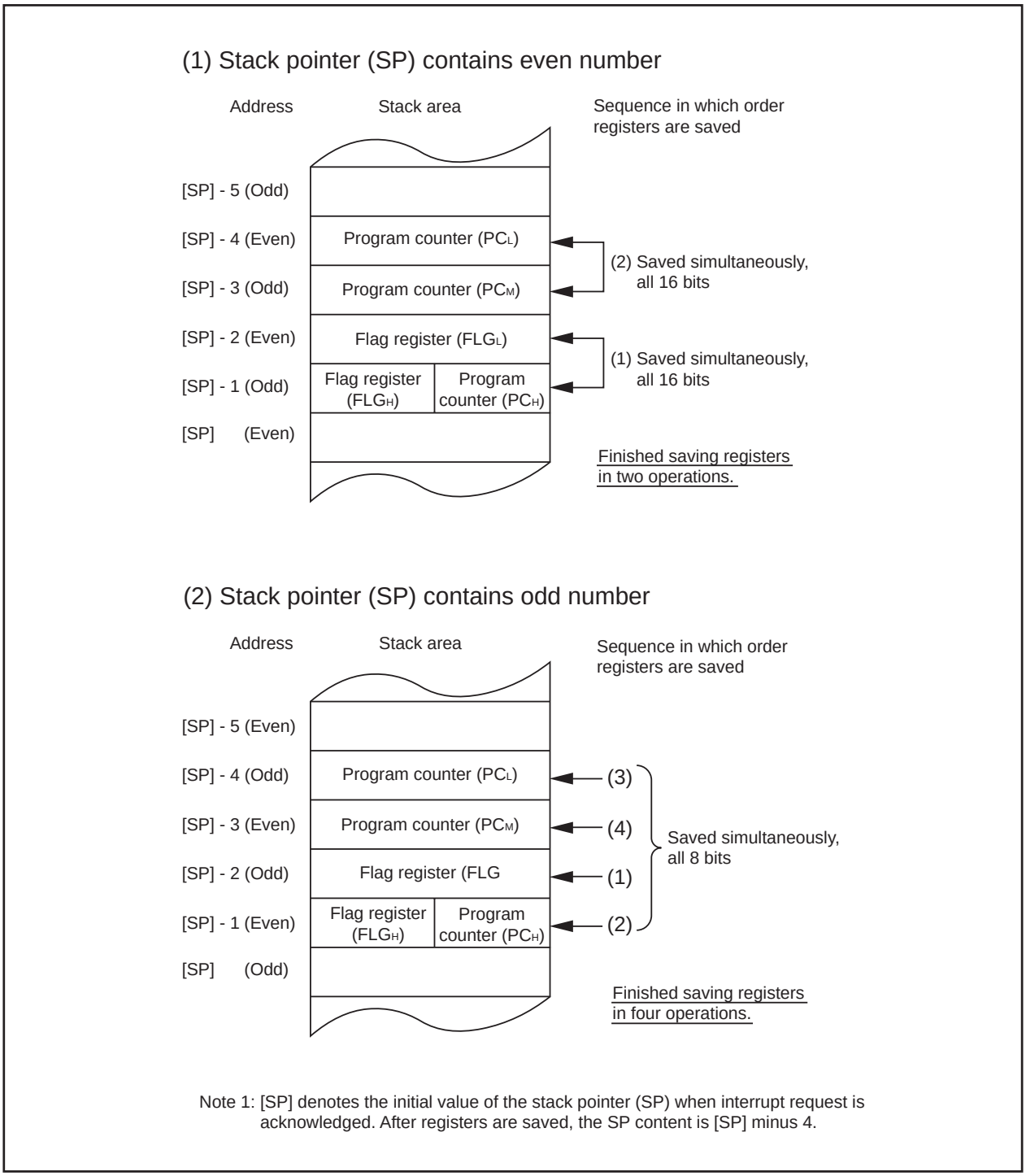


Figure 10.6 State of stack before and after acceptance of interrupt request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer (Note 1), at the time of acceptance of an interrupt request, is even or odd. If the content of the stack pointer (Note 1) is even, the content of the flag register (FLG) and the content of the program counter (PC) are saved, 16 bits at a time. If odd, their contents are saved in two steps, 8 bits at a time. Figure 10.7 shows the operation of the saving registers.
Note 1: This is the stack pointer indicated by the U flag.



Note 1: [SP] denotes the initial value of the stack pointer (SP) when interrupt request is acknowledged. After registers are saved, the SP content is [SP] minus 4.

Figure 10.7 Operation of saving registers

10.1.5.9 Returning from an Interrupt Routine

Executing the REIT instruction at the end of an interrupt routine returns the contents of the flag register (FLG) as it was immediately before the start of interrupt sequence and the contents of the program counter (PC), both of which have been saved in the stack area. Then control returns to the program that was being executed before the acceptance of the interrupt request, so that the suspended process resumes.

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

10.1.5.10 Interrupt Priority

If there are two or more interrupt requests occurring at a point in time within a single sampling (checking whether interrupt requests are made), the interrupt assigned a higher priority is accepted.

Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bit. If the same interrupt priority level is assigned, however, the interrupt assigned a higher hardware priority is accepted.

Priorities of the special interrupts, such as Reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are regulated by hardware.

Figure 10.8 shows the priorities of hardware interrupts.

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

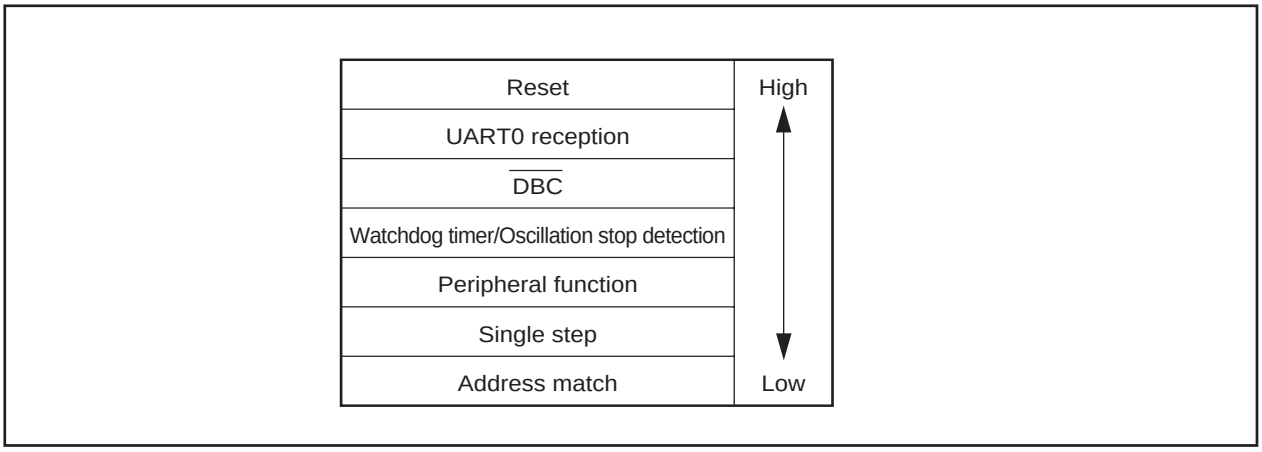


Figure 10.8 Hardware interrupts priorities

10.1.5.11 Interrupt Priority Level Judge Circuit

This circuit selects the interrupt with the highest priority level when two or more interrupts are generated simultaneously.

Figure 10.9 shows the interrupt resolution circuit.

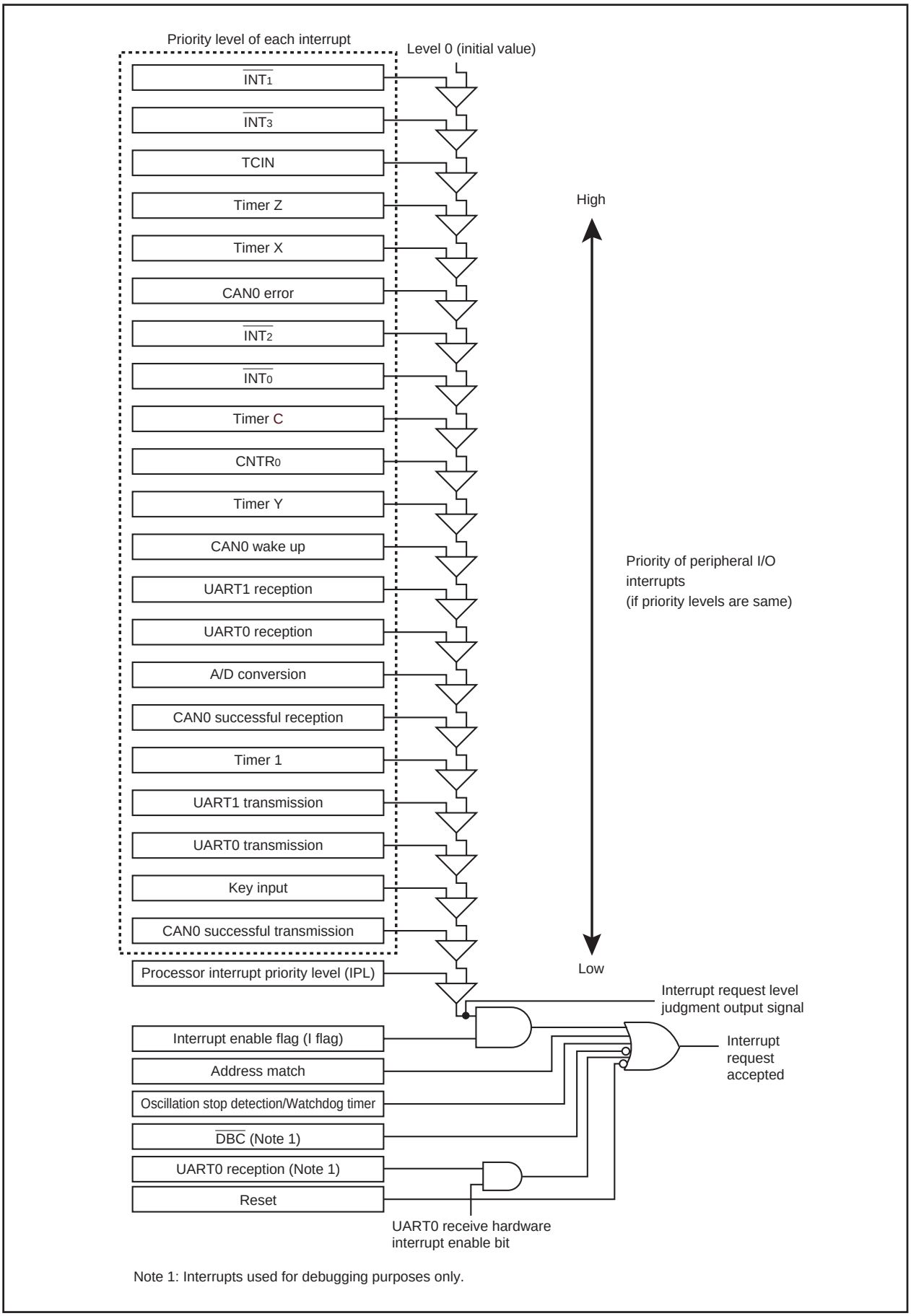


Figure 10.9 Interrupt resolution circuit

10.2 $\overline{\text{INT}}$ Interrupt

10.2.1 $\overline{\text{INT}}_0$ Interrupt

INT_0 to INT_3 are triggered by the edges of external inputs. The edge polarity of INT_0 to INT_2 is selected using the polarity select bit (bit 4 at addresses $005D_{16}$, $005E_{16}$ and $005F_{16}$). Input to INT_0 is available via filter with three different sampling frequencies.

As to external interrupt input, an interrupt can be generated both at the rising edge and at the falling edge by setting the $\overline{\text{INTi}}$ (i=0 to 3) input polarity select bit of the external input enable register (address 0096_{16}) to "1". To select both edges, set the polarity switching bit of the corresponding interrupt control register to "0" (falling edge). To select one edge, set the polarity switching bit of the corresponding interrupt control register to either "1" (raising edge) or "0" (falling edge). Please note that when one edge is selected using INT_3, the polarity will be a falling edge.

After setting the external input enable register, clear the interrupt request bit, and then enable the corresponding input interrupt. Moreover, you should write to the external input enable bit only under conditions where the corresponding input interrupt is disabled.

Figure 10.10 shows the external input related registers.

External input enable register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
INTEN	0096_{16}	00_{16}

Bit symbol	Bit name	Function	RW
INT0EN	$\overline{\text{INT}}_0$ input enable bit (Note 1)	0 : Disabled 1 : Enabled	RW
INT0PL	$\overline{\text{INT}}_0$ input polarity select bit (Note 1)	0 : One edge 1 : Two edges	RW
INT1EN	$\overline{\text{INT}}_1$ input enable bit	0 : Disabled 1 : Enabled	RW
INT1PL	$\overline{\text{INT}}_1$ input polarity select bit	0 : One edge 1 : Two edges	RW
INT2EN	$\overline{\text{INT}}_2$ input enable bit	0 : Disabled 1 : Enabled	RW
INT2PL	$\overline{\text{INT}}_2$ input polarity select bit	0 : One edge 1 : Two edges	RW
INT3EN	$\overline{\text{INT}}_3$ input enable bit	0 : Disabled 1 : Enabled	RW
INT3PL	$\overline{\text{INT}}_3$ input polarity select bit	0 : One edge 1 : Two edges	RW

Note 1: This bit must be set in condition of the $\overline{\text{INT}}_0$ pin one-shot trigger control bit (bit 6 at address 0084_{16})="0" ($\overline{\text{INT}}_0$ pin one-shot trigger invalid).

$\overline{\text{INT0}}$ input filter select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
INT0F	$001E_{16}$	$XXXXX000_2$

Bit symbol	Bit name	Function	RW
INT0F0	$\overline{\text{INT}}_0$ input filter select bit	b1 b0 0 0 : No filter 0 1 : Filter with f_1 sampling 1 0 : Filter with f_8 sampling 1 1 : Filter with f_{32} sampling	RW
INT0F1			RW
INT0F2	UART0 receive hardware interrupt enable bit (Note 1)	0 : Disabled 1 : Enabled	RW
Nothing is assigned. When write, nothing can be written. When read, their contents are indeterminate.			—

Note 1: Interrupts used for debugging purposes only. Be sure to set "0" to this bit.

Figure 10.10 External input related registers

10.2.2 INT0 Input Filter

The INT0 input has a digital filter which can be sampled by one of three sampling clocks. You select the sampling clock using the INT0 Input Filter Select bits, bits 1 and 0.

INT0 interrupt request occurs when the sampled input level matches three times.

When selecting "Sampling with filter", the value of the port P45, if read, will be the value after filtering.

Figure 10.11 shows the INT0 input filter.

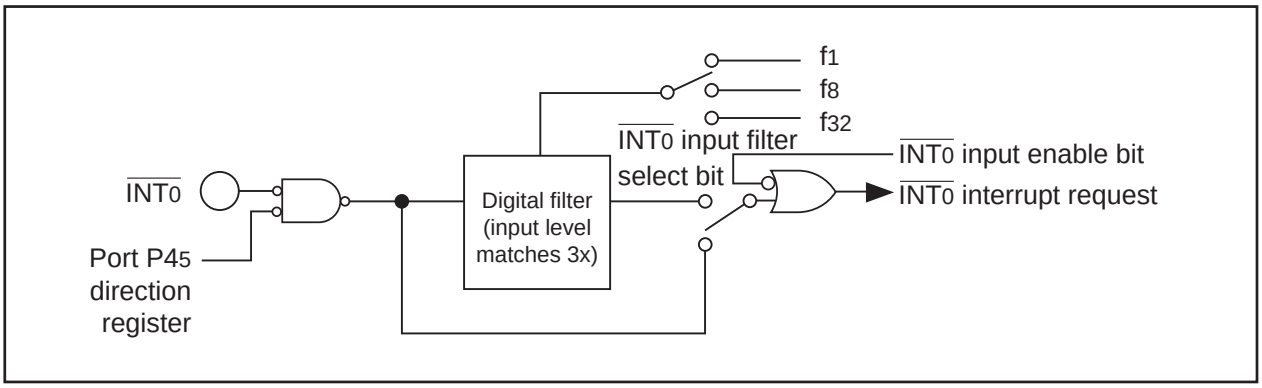


Figure 10.11 INT0 input filter

10.3 CNTR0 Interrupt

A CNTR0 interrupt is generated from the selected edge polarity, a rising or a falling edge, of the CNTR0 input signal. The edge polarity is selected using the CNTR0 polarity select bit (bit 2 at address 008B16).
When using the CNTR0 interrupt, the port P17 direction register should be set to input.
When the pulse output mode of timer X is selected, the CNTR0 pin functions as a pulse output pin. In this case, a CNTR0 interrupt occurs by a falling or rising edge output from the CNTR0 pin. The port P17 direction register should else be set to input at this time.
Figure 10.12 shows the timer X mode register.

Timer X mode register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
TXMR	008B16	0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1 (Note 2)	b1 b0 0 0 : Timer mode or pulse period measurement mode 0 1 : Pulse output mode (Note 1) 1 0 : Event counter mode 1 1 : Pulse width measurement mode	RW
TXMOD1			RW
R0EDG	CNTR0 polarity switching bit (Note 2)	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P30/TXOUT select bit	Function varies with each operation mode	RW
TXMOD2	Operation mode select bit 2	0 : Except in pulse period measurement mode 1 : Pulse period measurement mode	RW
TXEDG	Effectual edge reception flag (Note 3)	Function varies with each operation mode	RW
TXUND	Timer X under flow flag (Note 3)	Function varies with each operation mode	RW

Note 1: In the pulse output mode, the direction register of port P17 should be set to input.
Note 2: This bit should rewrite with inhibiting the CNTR0 interrupt. To use all interrupt, enable an interrupt after the CNTR0 interrupt request bit is cleared with the MOV instruction.
Note 3: Nothing is assigned to the pod probe for M16C/1N group (M301N2T-PRB).

Figure 10.12 Timer X mode register

10.4 TCIN Interrupt

A TCIN interrupt is generated from edges of a TCIN input signal or after 256 divisions of fRING.

To use TCIN input signal, set the time measurement input source switching bit (bit 7 at address 009A16) of timer C control register 0 to "0" (TCIN). The level of input to TCIN pin is sampled by one of three sampling clocks, f1, f8 or f32, selected with the digital filter clock select bit (bits 0 and 1 at address 009B16). The input level is determined when the sampled input level matches three times. (However, if the port P33 is read, the value will be the unfiltered value.) The edge polarity of an interrupt can be a rising edge, a falling edge, or both edges using the time measurement edge trigger select bits (bits 3 and 4 at address 009A16).

When triggered after 256 divisions of fRING, set the time measurement input source switching bit (bit 7 at address 009A16) to "1" (fRING256).

Figure 10.13 shows the timer C control registers 0 and 1.

Timer C control register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TCC0	009A16	0XX000002

Bit symbol	Bit name	Function	RW
TCC00	Time measurement control bit	0 : Time measurement disabled 1 : Time measurement enabled	RW
TCC01	Timer C clock select bit (Note 1)	b2 b1 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : Inhibited	RW
TCC02			RW
TCC03	Time measurement input edge trigger bit (Note 1)	b4 b3 0 0 : Rising edge 0 1 : Falling edge 1 0 : Both edges 1 1 : Inhibited	RW
TCC04			RW
Nothing is assigned. When write, set "0". When read, their contents are "0".			—
TCC07	Time measurement input source switching bit (Note 1 to 3)	0 : TCIN 1 : fRING256	RW

Note 1: Change this bit when time measurement is disabled.
Note 2: Set the ring oscillation stop bit (CM14) to "0" before setting this bit to "1".
Note 3: Inhibit an interrupt when changing the time measurement input source switching bit. The TCIN interrupt may be generated after changing the time measurement input source switching bit. An interrupt should be enabled after the interrupt request bit is cleared.

Timer C control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TCC1	009B16	XXXXXX112

Bit symbol	Bit name	Function	RW
TCC10	Digital filter clock select bit (Note 1)	b1 b0 0 0 : Inhibited 0 1 : f1 1 0 : f8 1 1 : f32	RW
TCC11			RW
Nothing is assigned. When write, set "0". When read, their contents are "0".			—

Note 1: Input edge becomes active when the same value from TCIN pin is sampled three times in succession.

Figure 10.13 Timer C control registers 0 and 1

10.5 Key Input Interrupt

When the direction register of any of P10 to P13 is set for input and the $\overline{KIi}$ (i=0 to 3) input enable bit of this port is set for enabled, if a falling or rising edge is input to that port, a key input interrupt is generated. A key input interrupt can also be used as a key-on wakeup function for cancelling the wait mode or stop mode.

Figure 10.14 shows the block diagram of the key input interrupts. When the appropriate signal ("L" for a pin that has falling edge selected and "H" for a pin that has rising edge selected) is input to a pin for the input inhibit process has not been executed, inputs to the other pins are not detected as interrupts.

You should overwrite the $\overline{KIi}$ (i=0 to 3) input polarity select bit or the $\overline{KIi}$ (i =0 to 3) input enable bit only under conditions where the key input interrupt is disabled. After overwriting the $\overline{KIi}$ (i=0 to 3) input polarity select bit or the $\overline{KIi}$ (i=0 to 3) input enable bit, clear the interrupt request bit, and then enable the key input interrupt.

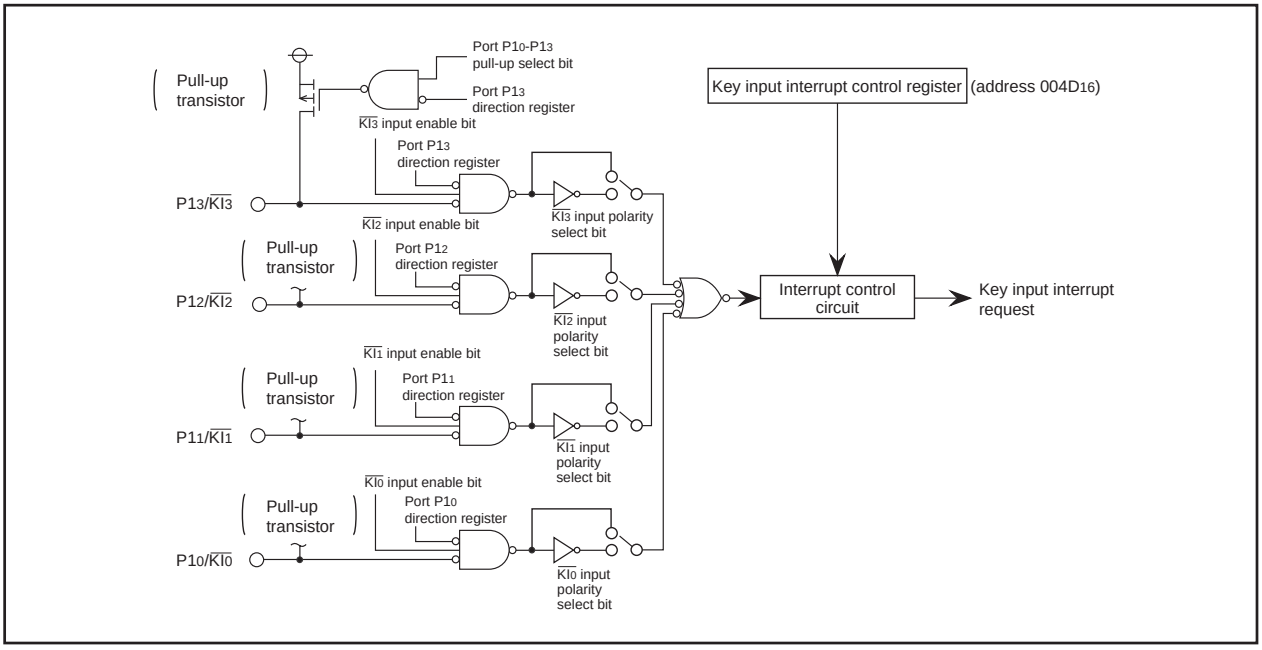


Figure 10.14 Block diagram of key input interrupt

Key input enable register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
KIEN	009816	0016

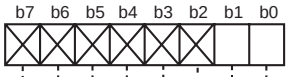
Bit symbol	Bit name	Function	RW
KI0EN	$\overline{KI0}$ input enable bit	0 : Disabled 1 : Enabled	RW
KI0PL	$\overline{KI0}$ input polarity select bit	0 : Falling edge 1 : Rising edges	RW
KI1EN	$\overline{KI1}$ input enable bit	0 : Disabled 1 : Enabled	RW
KI1PL	$\overline{KI1}$ input polarity select bit	0 : Falling edge 1 : Rising edges	RW
KI2EN	$\overline{KI2}$ input enable bit	0 : Disabled 1 : Enabled	RW
KI2PL	$\overline{KI2}$ input polarity select bit	0 : Falling edge 1 : Rising edges	RW
KI3EN	$\overline{KI3}$ input enable bit	0 : Disabled 1 : Enabled	RW
KI3PL	$\overline{KI3}$ input polarity select bit	0 : Falling edge 1 : Rising edges	RW

Figure 10.15 Key input enable register

10.6 Address Match Interrupt

An address match interrupt is generated immediately before the instruction at the address indicated by the address match interrupt register is executed. Two address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL). The value of the program counter (PC) for an address match interrupt varies depending on the instruction being executed. Figure 10.16 shows the address match interrupt-related registers.

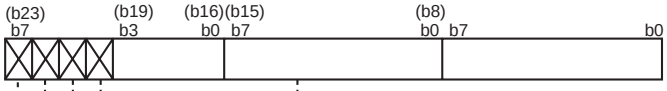
Address match interrupt enable register



Symbol	Address	When reset
AIER	0009_{16}	$XXXXXX00_2$

Bit symbol	Bit name	Function	RW
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—

Address match interrupt register i (i = 0, 1)



Symbol	Address
RMAD0	0012_{16} to 0010_{16}
RMAD1	0016_{16} to 0014_{16}

When reset
$XXXX0000_2$, 00000000_2, 00000000_2

Function	Values that can be set	RW
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.		—

Figure 10.16 Address match interrupt-related registers

10.7 Precautions for Interrupts

10.7.1 Reading Address 00000$_{16}$

When maskable interrupt is occurred, CPU reads the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.

The interrupt request bit of the certain interrupt written in address 00000$_{16}$ will then be set to "0".

Even if the address 00000$_{16}$ is read out by software, "0" is set to the enabled highest priority interrupt source request bit. Therefore, interrupt can be canceled and unexpected interrupt can occur.

Do not read address 00000$_{16}$ by software.

10.7.2 Setting the Stack Pointer

The value of the stack pointer immediately after reset is initialized to address 0000$_{16}$. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt. Concerning the first instruction immediately after reset, generating any interrupts is prohibited.

10.7.3 External Interrupt

Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to pins $\overline{INT_0}$ to $\overline{INT_3}$ regardless of the CPU operation clock.

When changing a polarity of pins $\overline{INT_0}$ to $\overline{INT_3}$ and CNTR$_0$, the interrupt request bit may become "1". Clear the interrupt request bit after changing the polarity. Figure 10.17 shows the switching condition of external interrupt request.

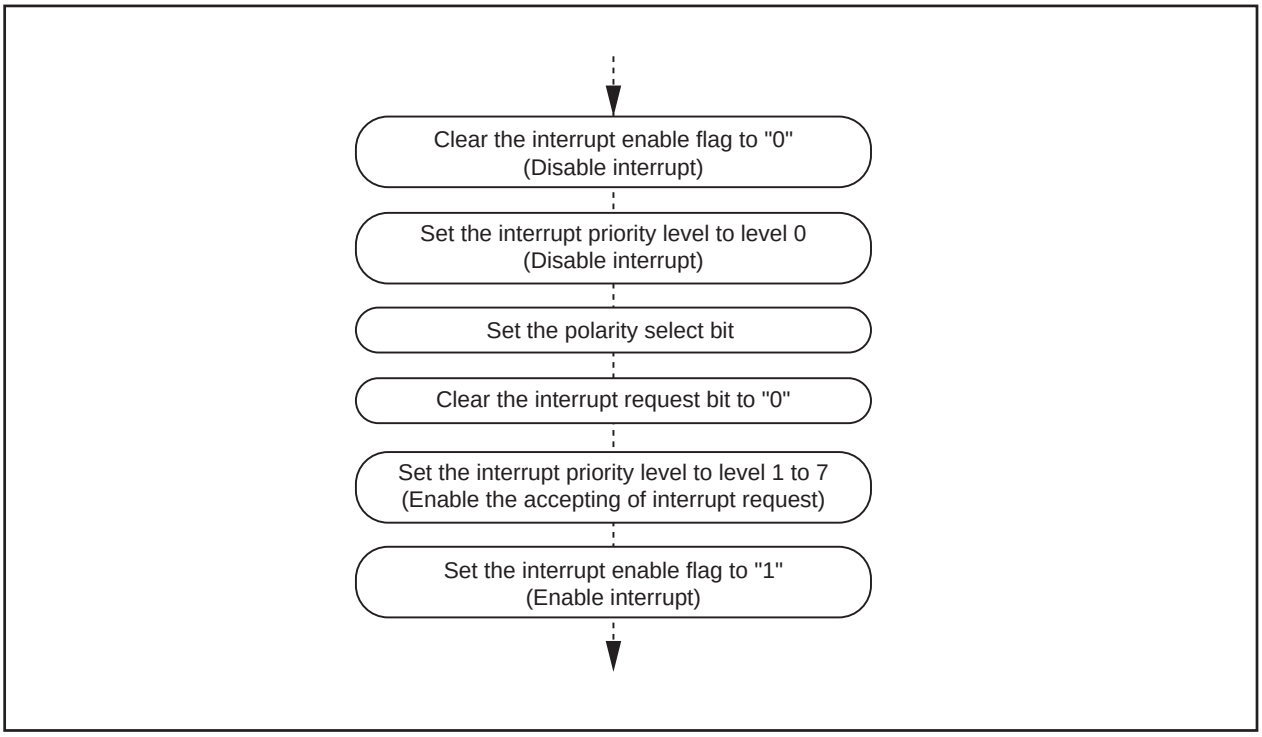


Figure 10.17 Switching condition of external interrupt request

10.7.4 Changing Interrupt Control Register

See "10.1.5.4 Rewrite the Interrupt Control Register".

11. Watchdog Timer

The watchdog timer has 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 is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. A watchdog timer interrupt or reset is generated when an underflow occurs in the watchdog timer. A watchdog timer interrupt or reset is selected by bit 2 of the processor mode register 1. When XIN is selected for the BCLK, bit 7 of the watchdog timer control register (address 000F16) selects the prescaler division ratio (by 16 or by 128). When XCIN is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address 000F16).

When XIN is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (16 or 128) x watchdog timer count (32768)}}{\text{BCLK}}$$

When XCIN is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (2) x watchdog timer count (32768)}}{\text{BCLK}}$$

For example, when BCLK is 10MHz and the prescaler division ratio is set to 16, the watchdog timer cycle is approximately 52.4 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address 000E16) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address 000E16).
Figure 11.1 shows the block diagram of the watchdog timer. Figure 11.2 shows the watchdog timer-related registers.

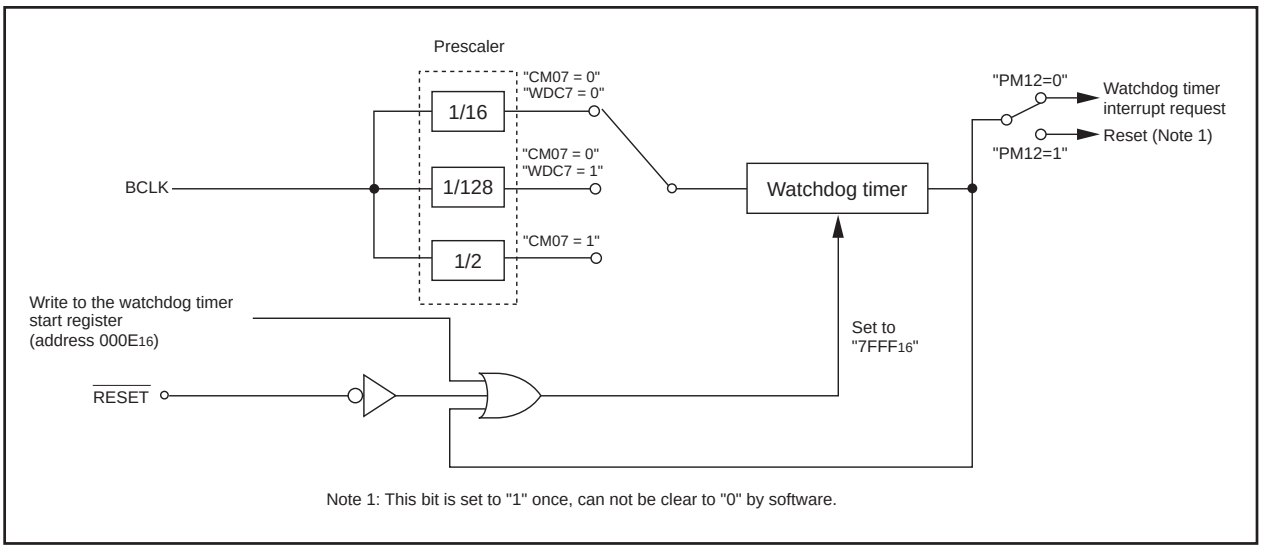


Figure 11.1 Block diagram of watchdog timer

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	When reset
WDC	$000F_{16}$	$000XXXXX_2$

Bit symbol	Bit name	Function	RW
High-order bit of watchdog timer			RO
Reserved bit		Set to "0"	RW
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	RW

Watchdog timer start register

b7	b0

Symbol	Address	When reset
WDTS	$000E_{16}$	Indeterminate

Function	RW
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to "$7FFF_{16}$" regardless of whatever value is written.	WO

Processor mode register 1 (Note 1)

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

Symbol	Address	When reset
PM1	0005_{16}	$00XXX0X0_2$

Bit symbol	Bit name	Function	RW
PM10	DATAROM area access bit (Note 2)	0 : Disabled 1 : Enabled	RW
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			—
PM12	WDT interrupt/reset switching bit	0 : Watchdog timer interrupt 1 : Reset (Note 3)	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—
Reserved bit		Set to "0"	RW
PM17	Wait bit	0 : No wait 1 : Wait	RW

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to "1" when writing new values to this register.
Note 2: This bit is valid for the flash memory version. For the mask ROM version, this bit must be set to "0".
Note 3: After setting this bit to "1", can not change to "0" by software.

Figure 11.2 Watchdog timer control and start registers

12. Timers

The microcomputer has four 8-bit timers and one 16-bit timer. The four 8-bit timers are Timer 1, Timer X, Timer Y, and Timer Z and each one has an 8-bit prescaler. The 16-bit timer is Timer C and has time measurement function. All these timers function independently. The count source for each timer is the operating clock that regulates the timing of timer operations such as counting and reloading.
Table 12.1 shows functional comparison.

Table 12.1 Functional comparison

		Timer1	TimerX	TimerY	TimerZ	TimerC
Configuration		8-bit timer with 8-bit prescaler	8-bit timer with 8-bit prescaler	8-bit timer with 8-bit prescaler	8-bit timer with 8-bit prescaler	16-bit free-run timer
Count		Down	Down	Down	Down	Up
Count source		•f_1 •f_8 •f_{32} •fc_{32}	•f_1 •f_8 •f_{32} •fc_{32}	•f_1 •f_8 •f_{RING} •fc_{32}	•f_1 •f_8 •TmrY underflow •fc_{32}	•f_1 •f_8 •f_{32}
Function	Timer mode	√	√	√	√	–
	Pulse output mode	–	√	–	–	–
	Event counter mode	–	√	–	–	–
	Pulse width measurement mode	–	√	–	–	–
	Pulse period measurement mode	–	√	–	–	–
	Programmable waveform generation mode	–	–	√	√	–
	Programmable one-shot generation mode	–	–	–	√	–
	Programmable wait one-shot generation mode	–	–	–	√	–
	Time measurement	–	–	–	–	√
Input pin		–	$CNTR_0$	–	INT_0	TCIN
Output pin		–	$CNTR_0$ TX_{OUT}	TY_{OUT}	TZ_{OUT}	–
Related interrupt		Tmr1 int	TmrX int $CNTR_0$ int	TmrY int	TmrZ int	TmrC int TCIN int
Timer stop		–	√	√	√	√

12.1 Timer 1

Timer 1 is an 8-bit timer with an 8-bit prescaler. Figure 12.1 shows the block diagram of Timer 1. The timer constantly counts an internally generated count source (clock source). The count source after reset is set to f_1. The timer cannot stop counting. Table 12.2 shows the specifications of Timer 1 and Figure 12.2 shows Timer 1 related registers.

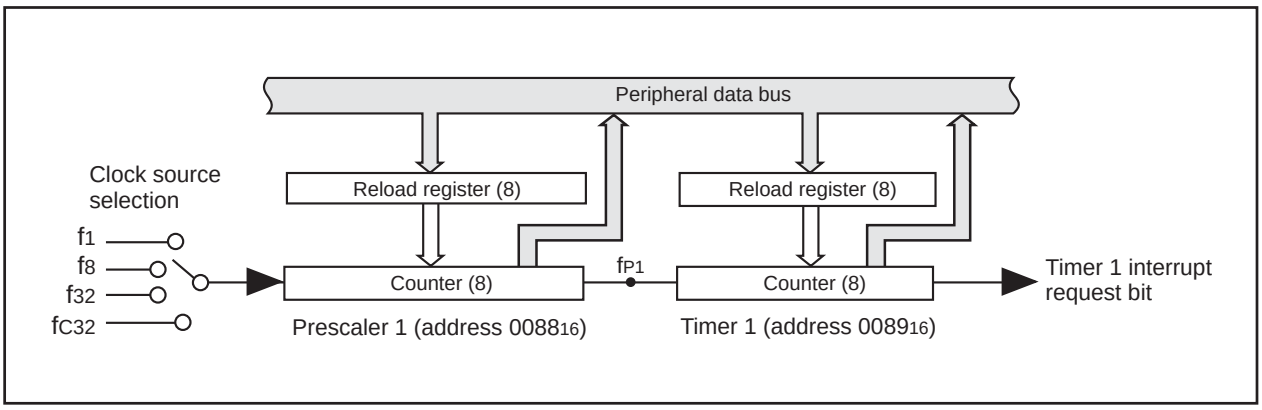


Figure 12.1 Block diagram of Timer 1

Table 12.2 Specifications of Timer 1 (Timer mode)

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	$\frac{1}{(n+1) \times (m+1)}$ n: Set value of Prescaler 1, m: Set value of Timer 1
Count start condition	After reset
Count stop condition	Disable to stop counting
Interrupt request generation timing	When Timer 1 underflows
Read from timer	Count value can be read out by reading Timer 1 register. Same applies to Prescaler 1 register.
Write to timer	When a value is written to Timer 1 register, it is written to both reload register and counter. Same applies to Prescaler 1 register.

Prescaler 1

b7 b0

Symbol	Address	When reset
PRE1	0088₁₆	Indeterminate

Function	Values that can be set	RW
When set value = n, Prescaler 1 divides the internal count source by n+1	00₁₆ to FF₁₆	RW

Timer 1

b7 b0

Symbol	Address	When reset
T1	0089₁₆	Indeterminate

Function	Values that can be set	RW
When set value = m, Timer 1 divides the underflow of Prescaler 1 by m+1	00₁₆ to FF₁₆	RW

Timer count source setting register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TCSS	008E₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TXCK0	Timer X count source select bit (Note 1)	b1 b0 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fc32	RW
TXCK1			RW
TYCK0	Timer Y count source select bit (Note 1, 2)	b3 b2 0 0 : f1 0 1 : f8 1 0 : On-chip oscillator output (Note 3) 1 1 : fc32	RW
TYCK1			RW
TZCK0	Timer Z count source select bit (Note 1, 4)	b5 b4 0 0 : f1 0 1 : f8 1 0 : Timer Y underflow 1 1 : fc32	RW
TZCK1			RW
T1CK0	Timer 1 count source select bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fc32	RW
T1CK1			RW

Note 1: Avoid switching a count source, while a counter is in progress. Timer counter should be stopped before switching a count source.
Note 2: The waveform extend function cannot be used when selecting f1 for count source.
Note 3: When attempting to select on-chip oscillator output, set the on-chip oscillation enable bit (CM14) of the system clock control register (address 0007₁₆) for oscillation enabled.
Note 4: The waveform extend function cannot be used when selecting Timer Y underflow and f1 for count source. Both the Timer Y primary underflow and the Timer Y secondary underflow are counted when selecting the Timer Y underflow for count source.

Figure 12.2 Timer 1-related register

12.2 Timer X

Timer X is an 8-bit timer with an 8-bit prescaler.

Timer X has the five operation modes listed as follows:

- Timer mode: The timer counts an internal count source (clock source).
- Pulse output mode: The timer counts an internal count source and outputs the pulses whose polarity is inverted at the timer the timer underflows.
- Event counter mode: The timer counts pulses from an external source.
- Pulse width measurement mode: The timer measures an external pulse's pulse width.
- Pulse period measurement mode: The timer measures an external pulse's period.

Figure 12.3 shows the block diagram of Timer X. Figures 12.4 and 12.5 shows the Timer X-related registers.

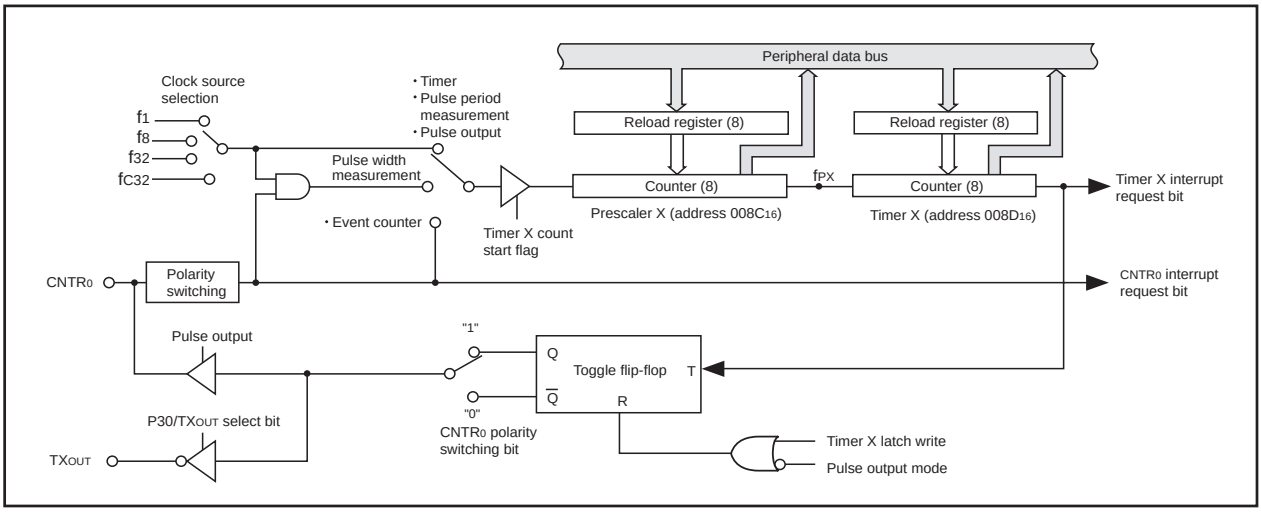


Figure 12.3 Block diagram of Timer X

Timer X mode register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
TXMR	008B16	0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1 (Note 2)	b1 b0 0 0 : Timer mode or pulse period measurement mode 0 1 : Pulse output mode (Note 1) 1 0 : Event counter mode 1 1 : Pulse width measurement mode	RW
TXMOD1			RW
R0EDG	CNTR0 polarity switching bit (Note 2)	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P30/TXOUT select bit	Function varies with each operation mode	RW
TXMOD2	Operation mode select bit 2	0 : Except in pulse period measurement mode 1 : Pulse period measurement mode	RW
TXEDG	Effectual edge reception flag (Note 3)	Function varies with each operation mode	RW
TXUND	Timer X under flow flag (Note 3)	Function varies with each operation mode	RW

Note 1: In the pulse output mode, the direction register of port P17 should be set to input.
Note 2: This bit should rewrite with inhibiting the CNTR0 interrupt. To use all interrupt, enable an interrupt after the CNTR0 interrupt request bit is cleared with the MOV instruction.
Note 3: Nothing is assigned to the pod probe for M16C/1N group (M301N2T-PRB).

Figure 12.4 Timer X-related registers (1)

Prescaler X

b7	b0	Symbol	Address	When reset
		PREX	008C16	FF16

Function	Values that can be set	RW
• Timer mode Internal count source is counted	0016 to FF16	RW
• Pulse output mode Internal count source is counted	0016 to FF16	RW
• Event counter mode Externally input pulses are counted	0016 to FF16	RW
• Pulse width measurement mode Pulse width of externally input pulses is measured (Internal count source is counted)	0016 to FF16	RW
• Pulse period measurement mode Pulse period of externally input pulses is measured (Internal count source is counted)	0016 to FF16	RW

Timer X

b7	b0	Symbol	Address	When reset
		TX	008D16	FF16

Function	Values that can be set	RW
• Timer mode, Pulse output mode, Event counter mode, Pulse width measurement mode Underflow of prescaler X is counted	0016 to FF16	RW
• Pulse period measurement mode Underflow of prescaler X is counted	0116 to FF16	RW

Timer count source setting register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TCSS	008E16	0016

Bit symbol	Bit name	Function	RW
TXCK0	Timer X count source select bit (Note 1)	b1 b0 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fc32	RW
TXCK1			RW
TYCK0	Timer Y count source select bit (Note 1, 2)	b3 b2 0 0 : f1 0 1 : f8 1 0 : On-chip oscillator output (Note 3) 1 1 : fc32	RW
TYCK1			RW
TZCK0	Timer Z count source select bit (Note 1, 4)	b5 b4 0 0 : f1 0 1 : f8 1 0 : Timer Y underflow 1 1 : fc32	RW
TZCK1			RW
T1CK0	Timer 1 count source select bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fc32	RW
T1CK1			RW

Note 1: Avoid switching a count source, while a counter is in progress. Timer counter should be stopped before switching a count source.
Note 2: The waveform extend function cannot be used when selecting f1 for count source.
Note 3: When attempting to select on-chip oscillator output, set the on-chip oscillation enable bit (CM14) of the system clock control register (address 0007₁₆) for oscillation enabled.
Note 4: The waveform extend function cannot be used when selecting Timer Y underflow and f1 for count source. Both the Timer Y primary underflow and the Timer Y secondary underflow are counted when selecting the Timer Y underflow for count source.

Figure 12.5 Timer X-related registers (2)

12.2.1 Timer Mode

In this mode, the timer counts an internally generated count source.

(See Table 12.3) Figure 12.6 shows the Timer X mode register in timer mode.

Table 12.3 Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	$\frac{1}{(n+1) \times (m+1)}$ n: Set value of Prescaler X, m: Set value of Timer X
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0)
Interrupt request generation timing	When Timer X underflows [Timer X interruption]
$CNTR_0$ pin function	Programmable I/O port or $CNTR_0$ interrupt input pin
TX_{OUT} pin function	Programmable I/O port
Read from timer	Count value can be read out by reading Timer X register. Same applies to Prescaler X register.
Write to timer	When a value is written to Timer X register, it is written to both reload register and counter. Same applies to Prescaler X register.

Timer X mode register

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

Symbol	Address	When reset
TXMR	$008B_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1 (Note 1)	b1 b0 0 0 : Timer mode	RW
TXMOD1			RW
R0EDG	$CNTR_0$ polarity switching bit (Note 1)	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	$P3_0/TX_{OUT}$ select bit	0 : In timer mode, set to "0"	RW
TXMOD2	Operation mode select bit 2	0 : In timer mode, set to "0"	RW
TXEDG	Effectual edge reception flag	Invalid in timer mode. When write, set "0". When read, this contents is indeterminate.	RW
TXUND	Timer X under flow flag	Invalid in timer mode. When write, set "0". When read, this contents is indeterminate.	RW

Note 1: This bit should rewrite with inhibiting the $CNTR_0$ interrupt. When using the interrupt, the interrupt must be enabled after clearing the $CNTR_0$ interrupt request bit to "0" using a MOV instruction.

Figure 12.6 Timer X mode register in timer mode

12.2.2 Pulse Output Mode

In this mode, the timer counts an internally generated count source, and outputs from the CNTR0 pin a pulse whose polarity is inverted each time the timer underflows.

(See Table 12.4) Figure 12.7 shows Timer X mode register in pulse output mode.

Table 12.4 Specifications of pulse output mode

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	$\frac{1}{(n+1) \times (m+1)}$ n: Set value of Prescaler X, m: Set value of Timer X
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0)
Interrupt request generation timing	• When Timer X underflows [Timer X interruption] • Rising (R0EDG=0) or falling (R0EDG=1) of CNTR0 output [CNTR0 interruption]
CNTR0 pin function	Pulse output
TXOUT pin function	Programmable I/O port or pulse output (Inverted waveform of the pulse output from the CNTR0 pin)
Read from timer	Count value can be read out by reading Timer X register. Same applies to Prescaler X register.
Write to timer	When a value is written to Timer X register, it is written to both reload register and counter. Same applies to Prescaler X register.
Select function	• Pulse output function Each time the timer underflows, the TXOUT pin's polarity is reversed • CNTR0 polarity switching function The polarity level at starting of pulse output can be selected to be "High" or "Low" with software.

Timer X mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol: TXMR | Address: 008B16 | When reset: 0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1 (Note 2)	b1 b0 0 1 : Pulse output mode (Note 1)	RW
TXMOD1			RW
R0EDG	CNTR0 polarity switching bit (Note 2)	0 : Output starts at "H" (Interrupt at rising edge) 1 : Output starts at "L" (Interrupt at falling edge)	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P30/TXOUT select bit	0 : Port P30 1 : TXOUT output (Note 3)	RW
TXMOD2	Operation mode select bit 2	0 : Set to "0" in pulse output mode	RW
TXEDG	Effectual edge reception flag	Invalid in pulse output mode. When write, set "0". When read, this contents is indeterminate.	RW
TXUND	Timer X under flow flag	Invalid in pulse output mode. When write, set "0". When read, this contents is indeterminate.	RW

Note 1: In the pulse output mode, the direction register of port P17 must be set to input.
Note 2: This bit should rewrite with inhibiting the CNTR0 interrupt. To use this interrupt, enable an interrupt after the CNTR0 interrupt request bit is cleared with the MOV instruction.
Note 3: Output is set regardless of the setting of the direction register of port P30.

Figure 12.7 Timer X mode register in pulse output mode

12.2.3 Event Counter Mode

In this mode, the timer counts an external signal fed to CNTR0 pin. (See Table 12.5) Figure 12.8 shows Timer X mode register in event counter mode.

Table 12.5 Specifications of event counter mode

Item	Specification
Count source	External signals fed to CNTR0 pin (Active edge is selected by software)
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	$\frac{1}{(n+1) \times (m+1)}$ n: Set value of Prescaler X, m: Set value of Timer X
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0)
Interrupt request generation timing	• When Timer X underflows [Timer X interruption] • Rising (R0EDG=0) or falling (R0EDG=1) of CNTR0 input [CNTR0 interruption]
CNTR0 pin function	Count source input
TXOUT pin function	Programmable I/O port
Read from timer	Count value can be read out by reading Timer X register. Same applies to Prescaler X register.
Write to timer	When a value is written to Timer X register, it is written to both reload register and counter. Same applies to Prescaler X register.
Select function	• CNTR0 polarity switching function The active edge of count source can be selected to be the rising or the falling edge with software.

Timer X mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0	0			1	0

Symbol	Address	When reset
TXMR	008B16	0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1 (Note 1)	b1 b0 1 0 : Event counter mode	RW
TXMOD1			RW
R0EDG	CNTR0 polarity switching bit (Note 1)	0 : Counts at rising edge (Interrupt at rising edge) 1 : Counts at falling edge (Interrupt at falling edge)	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P30/TXOUT select bit	0 : Set to "0" in event counter mode	RW
TXMOD2	Operation mode select bit 2	0 : Set to "0" in event counter mode	RW
TXEDG	Effectual edge reception flag	Invalid in event counter mode. When write, set "0". When read, this contents is indeterminate.	RW
TXUND	Timer X under flow flag	Invalid in event counter mode. When write, set "0". When read, this contents is indeterminate.	RW

Note 1: This bit should rewrite with inhibiting the CNTR0 interrupt. When using the interrupt, the interrupt must be enabled after clearing the CNTR0 interrupt request bit to "0" using a MOV instruction.

Figure 12.8 Timer X mode register in event counter mode

12.2.4 Pulse Width Measurement Mode

In this mode, the timer measures the pulse width of an external signal fed to CNTR0 pin.

(See Table 12.6) Figure 12.9 shows the Timer X mode register in pulse width measurement mode.

Figure 12.10 shows an operation example in pulse width measurement mode.

Table 12.6 Specifications of pulse width measurement mode

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	• Down count • Continuously counts the selected signal only when the measurement pulse is "H" level, or conversely only "L" level. • When the timer underflows, it reloads the reload register contents before continuing counting
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0)
Interrupt request generation timing	• When Timer X underflows [Timer X interruption] • Rising (R0EDG=0) or falling (R0EDG=1) of CNTR0 input [CNTR0 interruption]
CNTR0 pin function	Measurement pulse input
TXOUT pin function	Programmable I/O port
Read from timer	Count value can be read out by reading Timer X register. Same applies to Prescaler X register.
Write to timer	When a value is written to Timer X register, it is written to both reload register and counter. Same applies to Prescaler X register.
Select function	• CNTR0 polarity switching function The measurement pulse input can be selected to be "H" level width or "L" level width by software.

Timer X mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0	0			1	1

Symbol	Address	When reset
TXMR	008B16	0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1 (Note 1)	b1 b0 1 1 : Pulse width measurement mode	RW
TXMOD1			RW
R0EDG	CNTR0 polarity switching bit (Note 1)	0 : Measures "L" level width (Interrupt at rising edge) 1 : Measures "H" level width (Interrupt at falling edge)	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P30/TXOUT select bit	0 : Set to "0" in pulse width measurement mode	RW
TXMOD2	Operation mode select bit 2	0 : Set to "0" in pulse width measurement mode	RW
TXEDG	Effectual edge reception flag	Invalid in pulse width measurement mode. When write, set "0". When read, this contents is indeterminate.	RW
TXUND	Timer X under flow flag	Invalid in pulse width measurement mode. When write, set "0". When read, this contents is indeterminate.	RW

Note 1: This bit should rewrite with inhibiting the CNTR0 interrupt. When using the interrupt, the interrupt must be enabled after clearing the CNTR0 interrupt request bit to "0" using a MOV instruction.

Figure 12.9 Timer X mode register in pulse width measurement mode

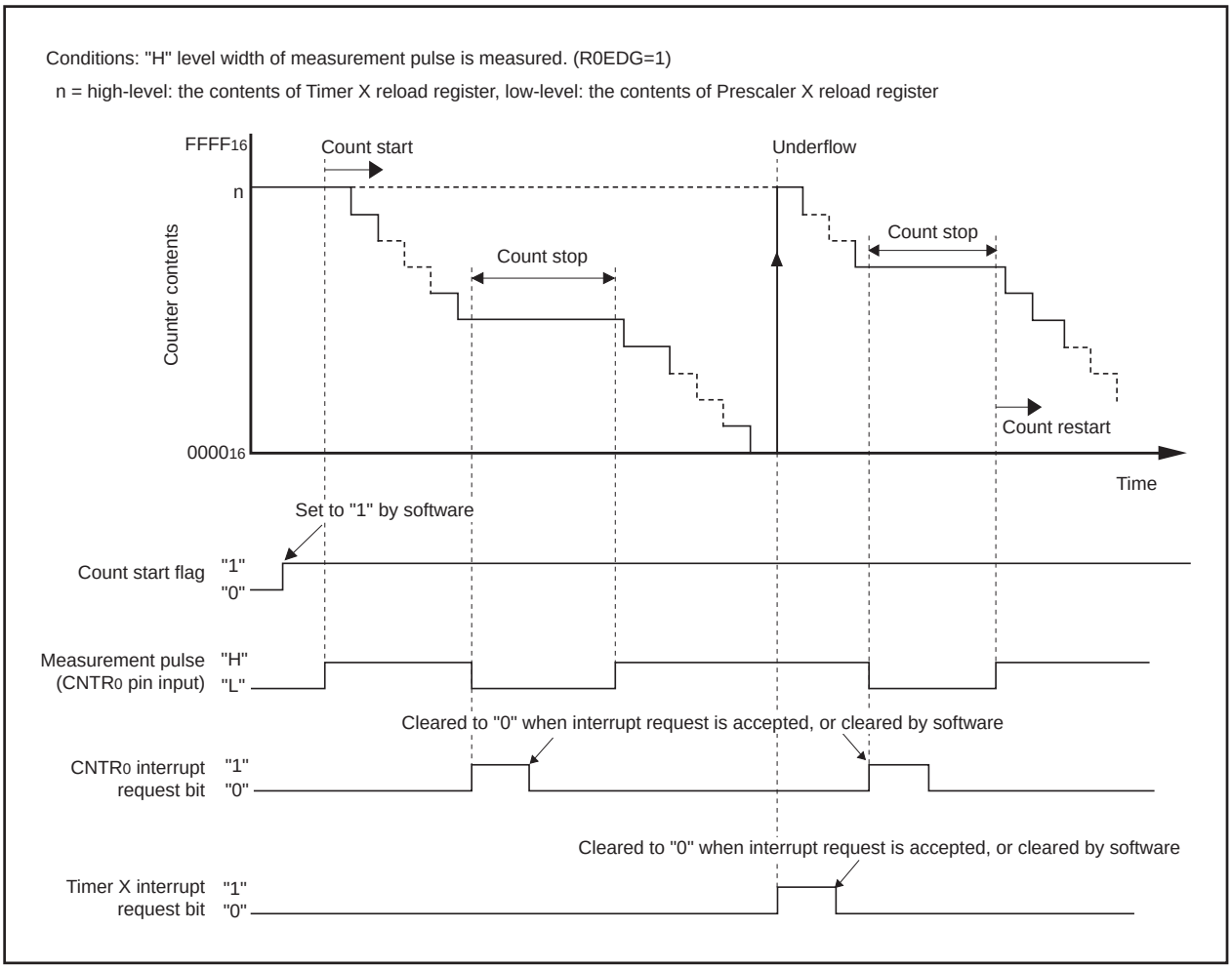


Figure 12.10 Operation example in pulse width measurement mode

12.2.5 Pulse Period Measurement Mode

In this mode, the timer measures the pulse period of an external signal fed to CNTR0 pin.

Table 12.7 lists specifications of pulse period measurement mode. Figure 12.11 shows the Timer X mode register in pulse period measurement mode. Figure 12.12 shows the operation example.

Table 12.7 Specifications of pulse period measurement mode

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	• Down count • After valid edge of measurement pulse is input, the timer X reloads contents in the reload register and continues counting in underflow of the second prescaler X.
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0)
Interrupt request generation timing	• When Timer X underflows [Timer X interruption] • Rising (R0EDG=0) or falling (R0EDG=1) of CNTR0 input [CNTR0 interruption or timer X interrupt]
CNTR0 pin function	Measurement pulse input (Note 1)
TXOUT pin function	Programmable I/O port
Read from timer	When reading Timer X register, the count value of buffer for read purpose can be read out. The buffer of read purpose retains the content of Timer X register upon an active edge of measurement pulse, and starts to read the content of Timer X register by reading Timer X.
Write to timer	When a value is written to Timer X register, it is written to both reload register and counter. Same applies to Prescaler X register.
Select function	• CNTR0 polarity switching function The measurement period of pulse input can be selected to be a period from one rising edge to the next rising edge or from one falling edge to the next falling edge by software.

Note 1: Avoid a shorter period pulse input than double prescaler X period. Longer pulse for H width and L width than the prescaler X period should be input to the CNTR0 pin. If shorter pulse than the period is input to the CNTR0 pin, the input may be disabled.

Timer X mode register

b7	b6	b5	b4	b3	b2	b1	b0
		1	0			0	0

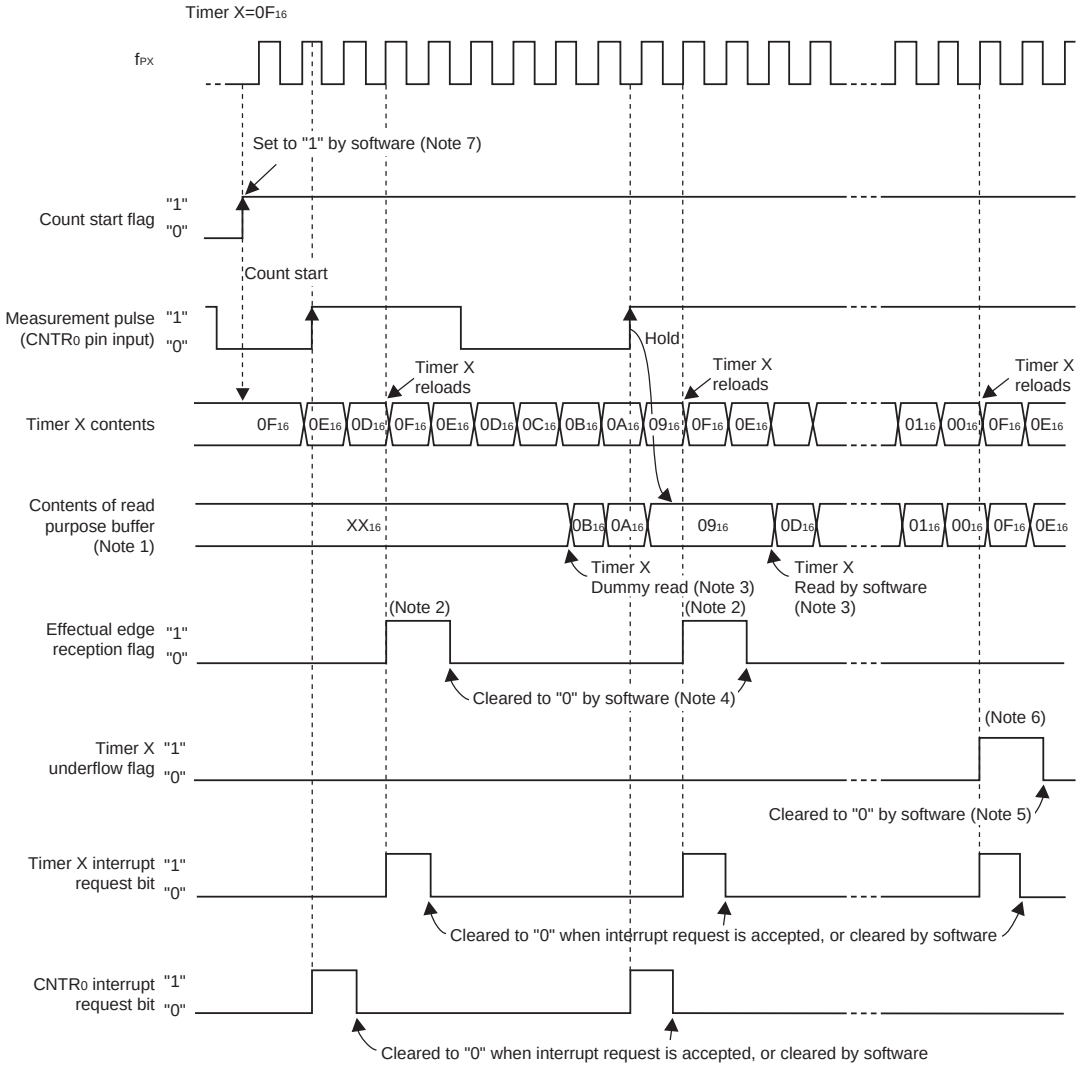
Symbol: TXMR — Address: 008B16 — When reset: 0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1 (Note 1)	b1 b0 0 0 : Pulse period measurement mode	RW
TXMOD1			RW
R0EDG	CNTR0 polarity switching bit (Note 1)	0 : Measures a measurement pulse from one rising edge to the next rising edge (Interrupt at rising edge) 1 : Measures a measurement pulse from one falling edge to the next falling edge (Interrupt at falling edge)	RW
TXS	Timer X count start flag (Note 3)	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P30/TXOUT select bit	0 : In pulse period measurement mode, set to "0"	RW
TXMOD2	Operation mode select bit 2 (Note 1)	1 : Pulse period measurement mode	RW
TXEDG	Effectual edge reception flag (Note 2)	0 : No effectual edge 1 : Effectual edge found	RW
TXUND	Timer X under flow flag (Note 2)	0 : No under flow 1 : Under flow found	RW

Note 1: This bit should rewrite with inhibiting the CNTR0 interrupt. When using the interrupt, the interrupt must be enabled after clearing the CNTR0 interrupt request bit to "0" using a MOV instruction.
Note 2: These bits are set to "0" by writing a "0" in a program. (Writing a "1" has no effect.)
Nothing is assigned to the pod probe for M16C/1N group (M301N2T-PRB).
Note 3: Execute the MOV instruction when stopping the timer X while pulse period measurement mode.

Figure 12.11 Timer X mode register in pulse period measurement mode

Conditions: A period from one rising edge to the next rising edge of measurement pulse is measured. (R0EDG=0)



Note 1: If timer X is read out in pulse period measurement mode, the contents of the read purpose buffer can be read.
Note 2: After an active edge of measurement pulse is input, effectual edge reception flag (TXEDG) is set to "1" when the prescaler X underflows for the second time.
Note 3: The timer X should be read out before the next active edge is input after TXEDG is set to "1". If the timer X is not read before the next active edge is input, the value in the read purpose buffer remains unchanged and therefore is not updated on an active edge.
Note 4: When set to "0" by software, use a MOV instruction to write "0" to the bit 6 (TXEDG) in the timer X mode register (008B16). At the same time, write "1" to the bit 7 (TXUND).
Note 5: When set to "0" by software, use a MOV instruction to write "0" to the bit 7 (TXUND) in the timer X more register (008B16). At the same time, write "1" to the bit 6 (TXEDG).
Note 6: If the timer X underflow flag (TXUND) and TXEDG are both set to "1". In this case, the validity of TXUND should be judged by the contents of the read purpose buffer.
Note 7: When setting the timer X count start flag to "1", the timer X interrupt request bit and the effectual edge reception flag may become "1".
Thus, the timer X interrupt must be enabled after setting the timer X count flag to "1" and clearing the timer X interrupt request bit and the effectual edge reception flag to "0" with inhibiting the timer X interrupt.

Figure 12.12 Operation example in pulse width measurement mode

12.3 Timer Y

Timer Y is an 8-bit timer with an 8-bit prescaler and has two reload registers - Timer Y Primary and Timer Y Secondary.

Timer Y has the two operation modes listed as follows:

- Timer mode: The timer counts an internal count source (clock source).
- Programmable waveform generation mode: The timer outputs pulses of a given width successively.

Figure 12.13 shows the block diagram of Timer Y. Figures 12.14 to 12.16 show the Timer Y-related registers.

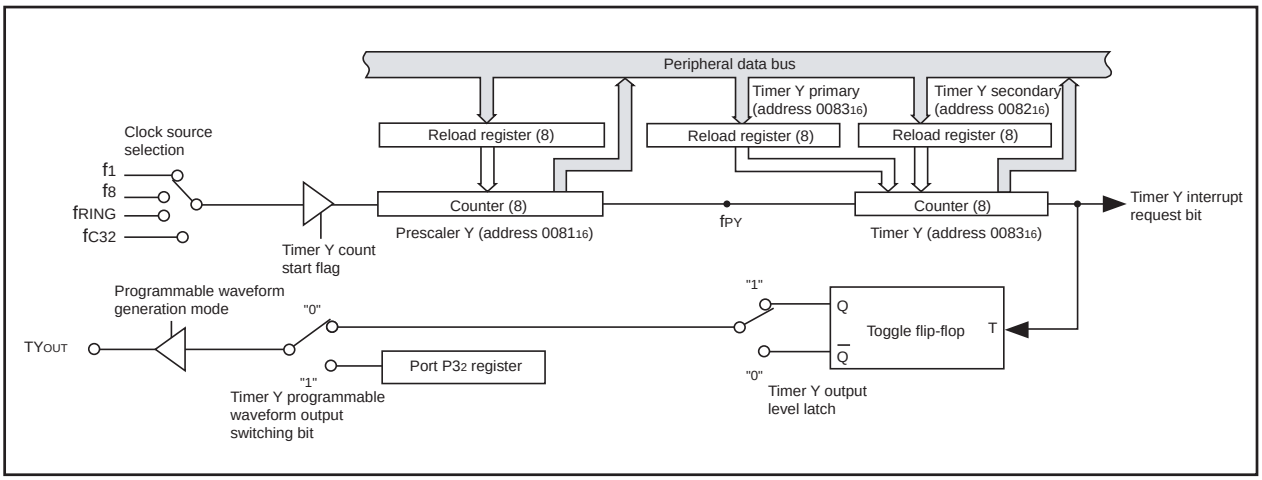


Figure 12.13 Block diagram of Timer Y

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
TYZMR	008016	000000X02

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode (Note 1)	RW
Nothing is assigned. When write, set "0". When read, the content is "0".			—
TYWC	Timer Y write control bit	Function varies depending on the operation mode	RW
TYS	Timer Y count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW
TZMOD0	Timer Z operation mode bit (Note 3)	b5 b4 0 0 : Timer mode 0 1 : Programmable waveform generation mode 1 0 : Programmable one-shot generation mode 1 1 : Programmable wait one-shot generation mode	RW
TZMOD1			RW
TZWC	Timer Z write control bit	Function varies depending on the operation mode	RW
TZS	Timer Z count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW

Note 1: In programmable waveform generation mode, port P32 is set for output regardless of the value of the direction register.
Note 2: When this bit is cleared to "0", the timer reloads the content of the reload register before it stops. Read out the count value before you stop the timer.
Note 3: When timer Z operation mode bit is set for "01", "10" or "11", port P31 is set for output regardless of the value of the direction register.

Figure 12.14 Timer Y-related registers (1)

Prescaler Y

b7 b0

Symbol	Address	When reset
PREY	008116	FF16

Function	Values that can be set	RW
• Timer mode Internal count source is counted	0016 to FF16	RW
• Programmable waveform generation mode Internal count source is counted	0016 to FF16 (Note 1)	RW

Note 1: When using the waveform extend function, set the value "0016" for the Prescaler Y.

Timer Y Secondary

b7 b0

Symbol	Address	When reset
TYSC	008216	FF16

Function	Values that can be set	RW
• Timer mode Invalid	———	—
• Programmable waveform generation mode Underflow of Prescaler Y is counted (Note 1)	0016 to FF16	WO (Note 2)

Note 1: The values of Timer Y Primary and Timer Y Secondary are reloaded to the Timer Y alternately for counting.
Note 2: The count value can be read out by reading the Timer Y Primary even when the secondary period is being counted.

Timer Y Primary

b7 b0

Symbol	Address	When reset
TYPR	008316	FF16

Function	Values that can be set	RW
• Timer mode Underflow of Prescaler Y is counted	0016 to FF16	RW
• Programmable waveform generation mode Underflow of Prescaler Y is counted (Note 1)	0016 to FF16	RW

Note 1: The values of Timer Y Primary and Timer Y Secondary are reloaded to the Timer Y alternately for counting.

Timer Y, Z output control register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TYZOC	008A16	XXXXX0002

Bit symbol	Bit name	Function	RW
TZOS	Timer Z one-shot start bit (Note 1)	0 : Stops one-shot 1 : Starts one-shot	RW
TYOCNT	Timer Y programmable waveform generation output switching bit (Note 2)	0 : Outputs programmable waveform 1 : Outputs the value of P32 port register	RW
TZOCNT	Timer Z programmable waveform generation output switching bit (Note 2)	0 : Outputs programmable waveform 1 : Outputs the value of P31 port register	RW
Nothing is assigned. When write, set "0". When read, their contents are "0".			—

Note 1: This bit is automatically cleared to "0" when the output of one-shot waveform is completed. This bit should be set to "0" by program when the one-shot waveform output is terminated by setting the count start flag to "0" during the waveform output.
Note 2: This bit is valid only when operating in programmable waveform generation mode.

Figure 12.15 Timer Y-related registers (2)

Timer Y, Z waveform output control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: PUM | Address: 0084₁₆ | When reset: 00₁₆

Bit symbol	Bit name	Function	RW
TYPUM0	Timer Y primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1)	RW
TYPUM1	Timer Y secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1)	RW
TZPUM0	Timer Z primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 2)	RW
TZPUM1	Timer Z secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 2)	RW
TYOPL	Timer Y output level latch	Function varies depending on the operation mode	RW
TZOPL	Timer Z output level latch	Function varies depending on the operation mode	RW
INOSTG	$\overline{INT_0}$ pin one-shot trigger control bit (Timer Z)	0 : $\overline{INT_0}$ pin one-shot trigger invalid 1 : $\overline{INT_0}$ pin one-shot trigger valid (Note 3)	RW
INOSEG	$\overline{INT_0}$ pin one-shot trigger polarity select bit (Note 4) (Timer Z)	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	RW

Note 1: When setting this bit to "1", the prescaler Y register must be set to "00₁₆".
Note 2: When setting this bit to "1", the prescaler Z register must be set to "00₁₆".
Note 3: When setting this bit to "1", this bit must be set to "1" after setting the $\overline{INT_0}$ input enable bit (bit 0 at address 0096₁₆), the $\overline{INT_0}$ input polarity select bit (bit 1 at address 0096₁₆), the $\overline{INT_0}$ input filter select bits (bits 0 and 1 at address 001E₁₆) and the $\overline{INT_0}$ pin one-shot trigger polarity select bit.
Note 4: This bit is valid only when the $\overline{INT_0}$ input polarity select bit (bit 1 at address 0096₁₆) is "0" (one-edge).

Timer count source setting register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: TCSS | Address: 008E₁₆ | When reset: 00₁₆

Bit symbol	Bit name	Function	RW
TXCK0	Timer X count source select bit (Note 1)	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	RW
TXCK1			RW
TYCK0	Timer Y count source select bit (Note 1, 2)	b3 b2 0 0 : f_1 0 1 : f_8 1 0 : On-chip oscillator output (Note 3) 1 1 : fc_{32}	RW
TYCK1			RW
TZCK0	Timer Z count source select bit (Note 1, 4)	b5 b4 0 0 : f_1 0 1 : f_8 1 0 : Timer Y underflow 1 1 : fc_{32}	RW
TZCK1			RW
T1CK0	Timer 1 count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	RW
T1CK1			RW

Note 1: Avoid switching a count source, while a counter is in progress. Timer counter should be stopped before switching a counter source.
Note 2: The waveform extend function cannot be used when selecting f_1 for count source.
Note 3: When attempting to select on-chip oscillator output, set the on-chip oscillation enable bit (CM14) of the system clock control register (address 0007₁₆) for oscillation enabled.
Note 4: The waveform extend function cannot be used when selecting Timer Y underflow and f_1 for count source. Both the Timer Y primary underflow and the Timer Y secondary underflow are counted when selecting the Timer Y underflow for count source.

Figure 12.16 Timer Y-related registers (3)

12.3.1 Timer Mode

In this mode, the timer counts an internally generated count source.

(See Table 12.8) The Timer Y secondary is unused in this mode. Figure 12.17 shows the Timer Y, Z mode register and Timer Y, Z waveform output control register in timer mode.

Table 12.8 Specifications of timer mode

Item	Specification
Count source	f_1, f_8, on-chip oscillator output, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting (When the Timer Y underflows, the contents of the Timer Y primary reload register is reloaded.) • When a counting stops, the timer reloads the content of the reload register before it stops.
Divide ratio	$\frac{1}{(n+1) \times (m+1)}$ n: Set value of Prescaler Y, m: Set value of Timer Y primary
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0) (Note 1)
Interrupt request generation timing	When Timer Y underflows
TY_{OUT} pin function	Programmable I/O port
Read from timer	Count value can be read out by reading Timer Y primary register. Same applies to Prescaler Y register.
Write to timer	When a value is written to Timer Y Primary register, it is written to both reload register and counter or written to only reload register. Selected by software. Same applies to Prescaler Y register.
Select function	• Timer Y write control function (Note 2) When a value is written to Timer Y Primary register, it can be selected that the value is written to both reload register and counter or written to only reload register. Same applies to Prescaler Z register.

Note 1: When the count is stopped, the Timer Y interrupt request bit becomes "1" and an interrupt may occur. Thus, interrupts must be disabled before the count is stopped. Furthermore, set the Timer Y interrupt request bit to "0" before starting counting again.

Note 2: If writing to the Timer Y or prescaler Y under the following conditions being filled at the same time the Timer Y interrupt request bit becomes "1" and an interrupt occurs.

<Conditions>

• Timer Y write control bit (bit 2 at address 0080_{16}) is "0" (write to timer and reload register simultaneously)

• Timer Y count start flag (bit 3 at address 0080_{16}) is "1" (count start)

To write to the Timer Y or prescaler Y in the above state, disable interrupts before writing.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
						X	0

Symbol	Address	When reset
TYZMR	0080_{16}	$000000X0_2$

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y operation mode bit	0 : Timer mode	RW
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—
TYWC	Timer Y write control bit	0 : Write to timer and reload register simultaneously (Note 1) 1 : Write to reload register	RW
TYS	Timer Y count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW
TZMOD0	Timer Z-related bit		RW
TZMOD1			RW
TZWC			RW
TZS			RW

Note 1: When this bit is "0", when you write in the prescaler Y while the timer Y is counting, the timer Y reloads the content of the timer Y reload register.
Note 2: When this bit is cleared to "0", the timer reloads the content of the reload register before it stops. Read out the count value before you stop the timer.

Timer Y, Z waveform output control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUM	0084_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TYPUM0	Timer Y primary waveform extension control bit	Invalid in timer mode	RW
TYPUM1	Timer Y secondary waveform extension control bit	Invalid in timer mode	RW
TZPUM0	Timer Z-related bits		RW
TZPUM1			RW
TYOPL	Timer Y output level latch	Invalid in timer mode	RW
TZOPL	Timer Z-related bits		RW
INOSTG			RW
INOSEG			RW

Figure 12.17 Timer Y, Z mode register in timer mode and Timer Y, Z waveform output control register

12.3.2 Programmable Waveform Generation Mode

In this mode, the microcontroller, while counting the set values of Timer Y primary and Timer Y secondary alternately, outputs from the TYOUT pin a waveform whose polarity is inverted each time Timer Y secondary underflows.

(See Table 12.9) A counting starts by counting the set value in the Timer Y primary. Figure 12.18 shows Timer Y, Z mode register in programmable waveform generation mode. Figure 12.19 shows the operation example.

Table 12.9 Specifications of programmable waveform generation mode

Item	Specification
Count source	f_1, f_8, on-chip oscillator output, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the contents of primary reload register and secondary reload register alternately before continuing counting. • When a counting stops, the timer reloads the content of the reload register before it stops.
Divide ratio	$\frac{fi}{(n+1) \times ((m+1)+(l+1))}$ n: Set value of Prescaler Y, m: Set value of Timer Y primary, l: Set value of Timer Y secondary
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0) (Note 1)
Interrupt request generation timing	When Timer Y underflows during secondary period
TYOUT pin function	Pulse output (Note 2)
Read from timer	Count value can be read out by reading Timer Y primary register. Same applies to Prescaler Y register. (Note 3)
Write to timer	When a value is written to Timer Y primary register, it is written to only reload register. Same applies to Timer Y secondary register and Prescaler Y register. (Note 4)
Select function	• Output level latch select function The output level of a waveform being counted during primary and secondary periods is selectable. • Programmable waveform generation output switching function (Note 5) Can select either programmable waveform or the value of port $P3_2$ register for output. • Waveform extend function (Note 6) The waveform output primary period and secondary period can each be extended 0.5 cycles of the count source. Frequency when waveform extended: 2xfi/((2x(m+1))+(2x(l+1))+TYPUM0+TYPUM1) Duty: (2x(m+1)+TYPUM0)/((2x(m+1)+TYPUM0)+(2x(l+1)+TYPUM1)) m: set value of Timer Y primary, l: set value of Timer Y secondary TYPUM0: Timer Y primary waveform extension control bit TYPUM1: Timer Y secondary waveform extension control bit

Note 1: When the count is stopped, the Timer Y interrupt request bit becomes "1" and an interrupt may occur. Thus, interrupts must be disabled before the count is stopped. Furthermore, set the Timer Y interrupt request bit to "0" before starting counting again.

Note 2: When the counting stopped, the pin is the secondary period output level.

Note 3: Even when counting the secondary period, read out the Timer Y primary register.

Note 4: The set value of Timer Y secondary register and waveform extension control bits as well as Timer Y primary register are made effective by writing a value to the Timer Y primary register. The written values are reflected to the waveform output from the next primary period after writing to the Timer Y primary register.

Note 5: The output is switched in sync with Timer Y secondary underflow.

Note 6: When using the waveform extend function, the Prescaler Y register must be set to "00_{16}".

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
					1	X	1

Symbol	Address	When reset
TYZMR	0080_{16}	$000000X0_2$

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y operation mode bit	1 : Programmable waveform generation mode (Note 1)	RW
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—
TYWC	Timer Y write control bit	1 : Set to "1" in programmable waveform generation mode	RW
TYS	Timer Y count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW
TZMOD0	Timer Z-related bit		RW
TZMOD1			RW
TZWC			RW
TZS			RW

Note 1: Output is set for Port $P3_2$ regardless of the value of the direction register.
Note 2: When this bit is cleared to "0", the timer reloads the content of the reload register before it stops. Read out the count value before you stop the timer.

Timer Y, Z waveform output control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUM	0084_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TYPUM0	Timer Y primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1, 2)	RW
TYPUM1	Timer Y secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1, 2)	RW
TZPUM0	Timer Z-related bits		RW
TZPUM1			RW
TYOPL	Timer Y output level latch	0 : Outputs "H" for the period set by Timer Y primary and "L" for the period set by Timer Y secondary. "L" is outputted when the timer is stopped. 1 : Outputs "L" for the period set by Timer Y primary and "H" for the period set by Timer Y secondary. "H" is outputted when the timer is stopped.	RW
TZOPL	Timer Z-related bits		RW
INOSTG			RW
INOSEG			RW

Note 1: When setting this bit to "1", the Prescaler Y Register must be set to "00_{16}".
Note 2: The waveform extend function cannot be used when selecting f_1 for count source.

Figure 12.18 Timer Y, Z mode register and Timer Y, Z waveform output control register in programmable waveform generation mode

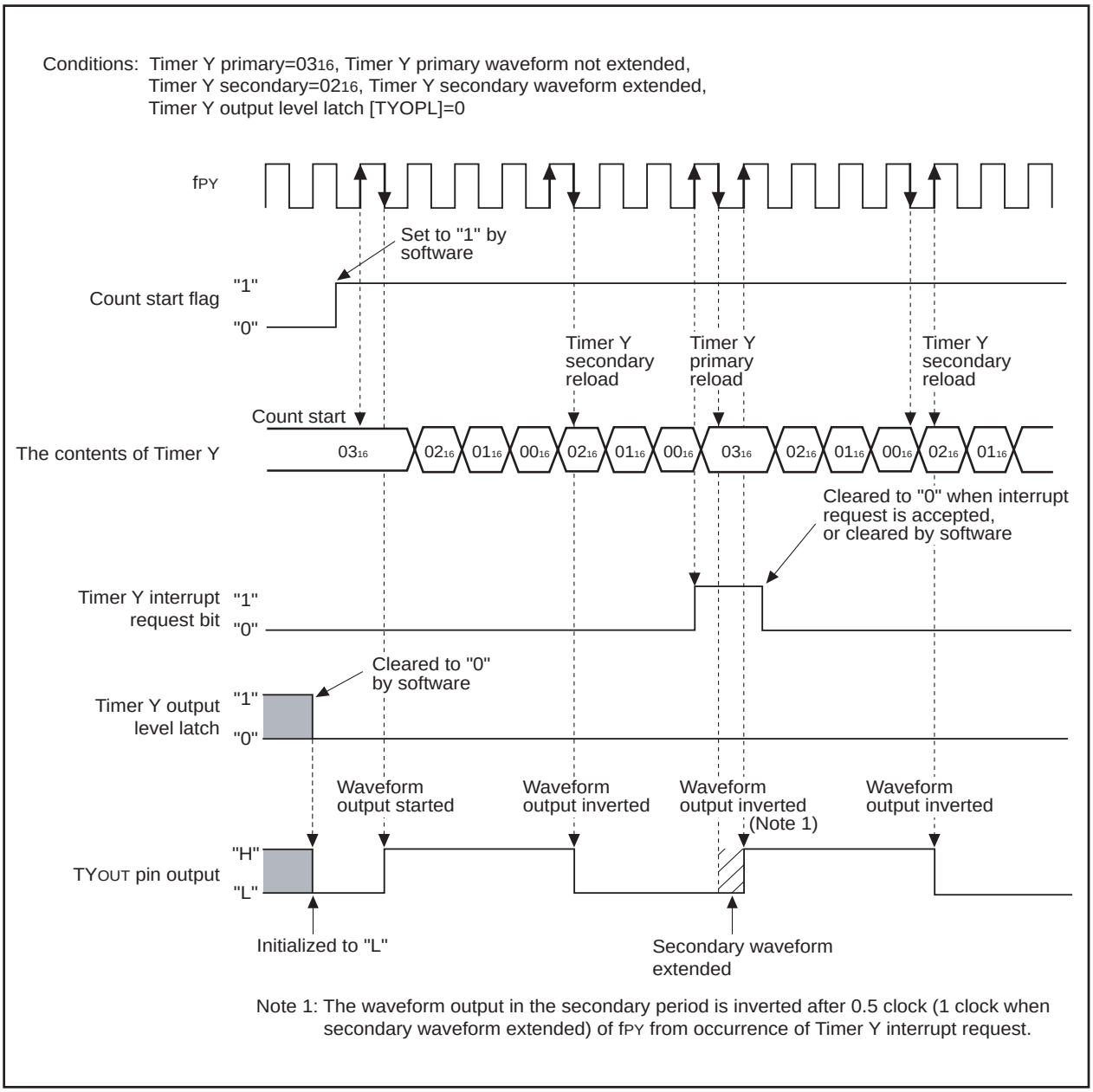


Figure 12.19 Timer Y operation example in programmable waveform generation mode

■Programmable waveform generation output switching function

When the Timer Y programmable waveform generation output switching bit (bit 1 at address 008A16) is set to 0, the output from TYOUT is inverted synchronously when the Timer Y secondary underflows. And when set to 1, the Port P32 register value is output from TYOUT synchronously when the Timer Y secondary underflows.

12.4 Timer Z

Timer Z is an 8-bit timer with an 8-bit prescaler and has two reload registers - Timer Z Primary and Timer Z Secondary.

Timer Z has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source (clock source) or Timer Y underflow.
- Programmable waveform generation mode: The timer outputs pulses of a given width successively.
- Programmable one-shot generation mode: The timer outputs one-shot pulse.
- Programmable wait one-shot generation mode: The timer outputs delayed one-shot pulse.

Figure 12.20 shows the block diagram of Timer Z. Figures 12.21 to 12.24 show the Timer Z-related registers.

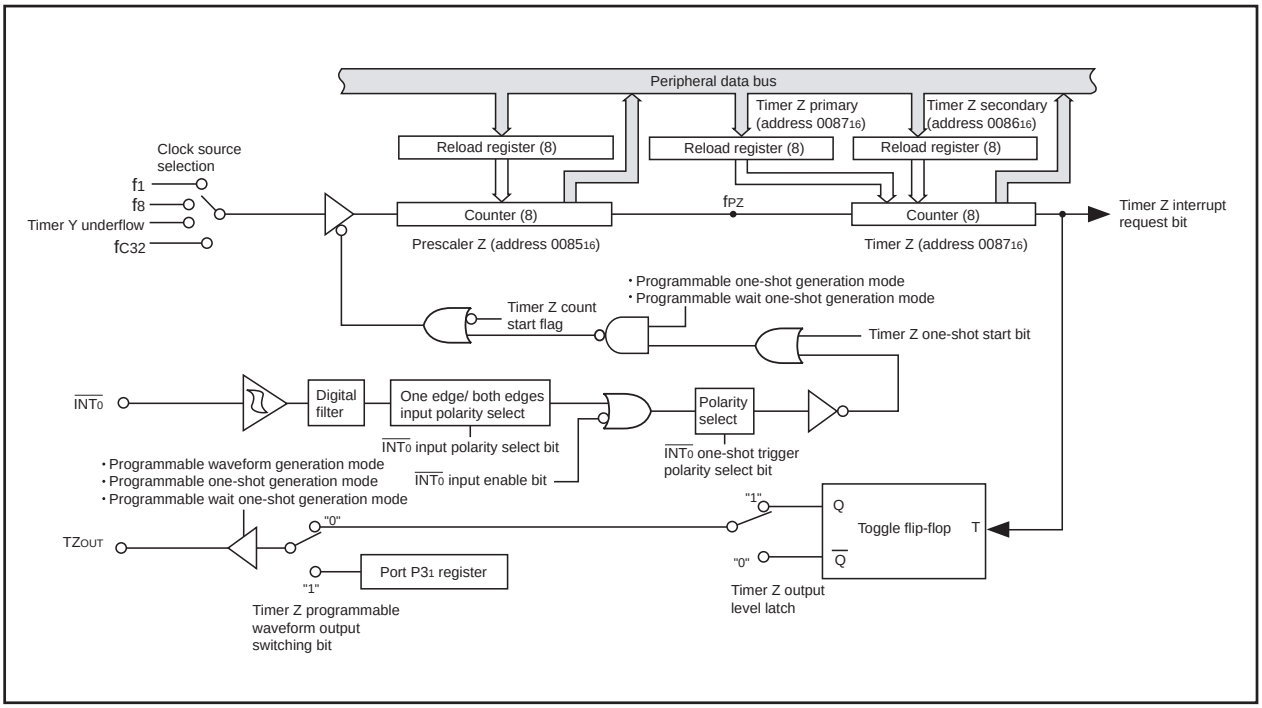
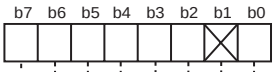


Figure 12.20 Block diagram of Timer Z

Timer Y, Z mode register



Symbol	Address	When reset
TYZMR	0080_{16}	$000000X0_2$

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode (Note 1)	RW
Nothing is assigned. When write, set "0". When read, the content is "0".			—
TYWC	Timer Y write control bit	Function varies depending on the operation mode	RW
TYS	Timer Y count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW
TZMOD0	Timer Z operation mode bit (Note 3)	b5 b4 0 0 : Timer mode 0 1 : Programmable waveform generation mode 1 0 : Programmable one-shot generation mode 1 1 : Programmable wait one-shot generation mode	RW
TZMOD1			RW
TZWC	Timer Z write control bit	Function varies depending on the operation mode	RW
TZS	Timer Z count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW

Note 1: In programmable waveform generation mode, port $P3_2$ is set for output regardless of the value of the direction register.
Note 2: When this bit is cleared to "0", the timer reloads the content of the reload register before it stops. Read out the count value before you stop the timer.
Note 3: When timer Z operation mode bit is set for "01", "10" or "11", port $P3_1$ is set for output regardless of the value of the direction register.

Figure 12.21 Timer Z-related registers (1)

Prescaler Z

b7	b0	Symbol	Address	When reset
		PREZ	0085₁₆	FF₁₆

Function	Values that can be set	RW
• Timer mode Internal count source or Timer Y underflow is counted	00₁₆ to FF₁₆	RW
• Programmable waveform generation mode Internal count source or Timer Y underflow is counted	00₁₆ to FF₁₆ (Note 1)	RW
• Programmable one-shot generation mode Internal count source or Timer Y underflow is counted	00₁₆ to FF₁₆ (Note 1)	RW
• Programmable wait one-shot generation mode Internal count source or Timer Y underflow is counted	00₁₆ to FF₁₆ (Note 1)	RW

Note 1: When using the waveform extend function, set the value "00₁₆" for the Prescaler Z.

Timer Z Secondary

b7	b0	Symbol	Address	When reset
		TZSC	0086₁₆	FF₁₆

Function	Values that can be set	RW
• Timer mode Invalid	————	—
• Programmable waveform generation mode Underflow of Prescaler Z is counted (Note 1)	00₁₆ to FF₁₆	WO (Note 2)
• Programmable one-shot generation mode Invalid	————	—
• Programmable wait one-shot generation mode Underflow of Prescaler Z is counted (One-shot width is counted)	00₁₆ to FF₁₆	WO

Note 1: Each value of Timer Z Primary and Timer Z Secondary is reloaded to the Timer Z alternately for counting.
Note 2: The count value can be read out by reading the Timer Z Primary even when the secondary period is being counted.

Timer Z Primary

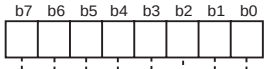
b7	b0	Symbol	Address	When reset
		TZPR	0087₁₆	FF₁₆

Function	Values that can be set	RW
• Timer mode Underflow of Prescaler Z is counted	00₁₆ to FF₁₆	RW
• Programmable waveform generation mode Underflow of Prescaler Z is counted (Note 1)	00₁₆ to FF₁₆	RW
• Programmable one-shot generation mode Underflow of Prescaler Z is counted (One-shot width is counted)	00₁₆ to FF₁₆	RW
• Programmable wait one-shot generation mode Underflow of Prescaler Z is counted (Wait period is counted)	00₁₆ to FF₁₆	RW

Note 1: Each value of Timer Z Primary and Timer Z Secondary is reloaded to the Timer Z alternately for counting.

Figure 12.22 Timer Z-related registers (2)

Timer Y, Z waveform output control register

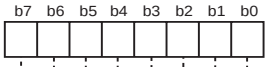


Symbol	Address	When reset
PUM	0084₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TYPUM0	Timer Y primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1)	RW
TYPUM1	Timer Y secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1)	RW
TZPUM0	Timer Z primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 2)	RW
TZPUM1	Timer Z secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 2)	RW
TYOPL	Timer Y output level latch	Function varies depending on the operation mode	RW
TZOPL	Timer Z output level latch	Function varies depending on the operation mode	RW
INOSTG	$\overline{INT_0}$ pin one-shot trigger control bit (Timer Z)	0 : $\overline{INT_0}$ pin one-shot trigger invalid 1 : $\overline{INT_0}$ pin one-shot trigger valid (Note 3)	RW
INOSEG	$\overline{INT_0}$ pin one-shot trigger polarity select bit (Note 4) (Timer Z)	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	RW

Note 1: When setting this bit to "1", the prescaler Y register must be set to "00₁₆".
Note 2: When setting this bit to "1", the prescaler Z register must be set to "00₁₆".
Note 3: When setting this bit to "1", this bit must be set to "1" after setting the $\overline{INT_0}$ input enable bit (bit 0 at address 0096₁₆), the $\overline{INT_0}$ input polarity select bit (bit 1 at address 0096₁₆), the $\overline{INT_0}$ input filter select bits (bits 0 and 1 at address 001E₁₆) and the $\overline{INT_0}$ pin one-shot trigger polarity select bit.
Note 4: This bit is valid only when the $\overline{INT_0}$ input polarity select bit (bit 1 at address 0096₁₆) is "0" (one-edge).

Timer count source setting register



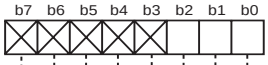
Symbol	Address	When reset
TCSS	008E₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TXCK0	Timer X count source select bit (Note 1)	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TXCK1			RW
TYCK0	Timer Y count source select bit (Note 1, 2)	b3 b2 0 0 : f_1 0 1 : f_8 1 0 : On-chip oscillator output (Note 3) 1 1 : f_{C32}	RW
TYCK1			RW
TZCK0	Timer Z count source select bit (Note 1, 4)	b5 b4 0 0 : f_1 0 1 : f_8 1 0 : Timer Y underflow 1 1 : f_{C32}	RW
TZCK1			RW
T1CK0	Timer 1 count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
T1CK1			RW

Note 1: Avoid switching a count source, while a counter is in progress. Timer counter should be stopped before switching a counter source.
Note 2: The waveform extend function cannot be used when selecting f_1 for count source.
Note 3: When attempting to select on-chip oscillator output, set the on-chip oscillation enable bit (CM14) of the system clock control register (address 0007₁₆) for oscillation enabled.
Note 4: The waveform extend function cannot be used when selecting Timer Y underflow and f_1 for count source. Both the Timer Y primary underflow and the Timer Y secondary underflow are counted when selecting the Timer Y underflow for count source.

Figure 12.23 Timer Z-related registers (3)

Timer Y, Z output control register

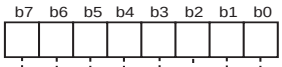


Symbol	Address	When reset
TYZOC	008A16	XXXXX0002

Bit symbol	Bit name	Function	RW
TZOS	Timer Z one-shot start bit (Note 1)	0 : Stops one-shot 1 : Starts one-shot	RW
TYOCNT	Timer Y programmable waveform generation output switching bit (Note 2)	0 : Outputs programmable waveform 1 : Outputs the value of P32 port register	RW
TZOCNT	Timer Z programmable waveform generation output switching bit (Note 2)	0 : Outputs programmable waveform 1 : Outputs the value of P31 port register	RW
Nothing is assigned. When write, set "0". When read, their contents are "0".			—

Note 1: This bit is automatically cleared to "0" when the output of one-shot waveform is completed. This bit should be set to "0" by program when the one-shot waveform output is terminated by setting the count start flag to "0" during the waveform output.
Note 2: This bit is valid only when operating in programmable waveform generation mode.

External input enable register

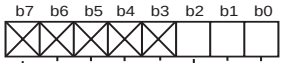


Symbol	Address	When reset
INTEN	009616	0016

Bit symbol	Bit name	Function	RW
INT0EN	$\overline{INT0}$ input enable bit (Note 1)	0 : Disabled 1 : Enabled	RW
INT0PL	$\overline{INT0}$ input polarity select bit (Note 1)	0 : One edge 1 : Two edges	RW
INT1EN	$\overline{INT1}$ input enable bit	0 : Disabled 1 : Enabled	RW
INT1PL	$\overline{INT1}$ input polarity select bit	0 : One edge 1 : Two edges	RW
INT2EN	$\overline{INT2}$ input enable bit	0 : Disabled 1 : Enabled	RW
INT2PL	$\overline{INT2}$ input polarity select bit	0 : One edge 1 : Two edges	RW
INT3EN	$\overline{INT3}$ input enable bit	0 : Disabled 1 : Enabled	RW
INT3PL	$\overline{INT3}$ input polarity select bit	0 : One edge 1 : Two edges	RW

Note 1: This bit must be set in condition of the $\overline{INT0}$ pin one-shot trigger control bit (bit 6 at address 008416)="0" ($\overline{INT0}$ pin one-shot trigger invalid).

$\overline{INT0}$ input filter select register



Symbol	Address	When reset
INT0F	001E16	XXXXX0002

Bit symbol	Bit name	Function	RW
INT0F0	$\overline{INT0}$ input filter select bit	b1 b0 0 0 : No filter 0 1 : Filter with f1 sampling 1 0 : Filter with f8 sampling 1 1 : Filter with f32 sampling	RW
INT0F1			RW
INT0F2	UART0 receive hardware interrupt enable bit (Note 1)	0 : Disabled 1 : Enabled	RW
Nothing is assigned. When write, nothing can be written. When read, their contents are indeterminate.			—

Note 1: Interrupts used for debugging purposes only. Be sure to set "0" to this bit.

Figure 12.24 Timer Z-related registers (4)

12.4.1 Timer Mode

In this mode, the timer counts an internally generated count source or Timer Y underflow. (See Table 12.10) The Timer Z secondary is unused in this mode. Figure 12.25 shows the Timer Y, Z mode register and Timer Y, Z waveform output control register in timer mode.

Table 12.10 Specifications of timer mode

Item	Specification
Count source	f_1, f_8, Timer Y underflow, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting (When the Timer Z underflows, the contents of the Timer Z primary reload register is reloaded.) • When a counting stops, the timer reloads the content of the reload register before stopping counting.
Divide ratio	$\frac{1}{(n+1) \times (m+1)}$ n: Set value of Prescaler Z, m: Set value of Timer Z primary
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0) (Note 1)
Interrupt request generation timing	When Timer Z underflows
TY_{OUT} pin function	Programmable I/O port
INT_0 pin function	Programmable I/O port, or external interrupt input pin
Read from timer	Count value can be read out by reading Timer Z primary register. Same applies to Prescaler Z register.
Write to timer	When a value is written to Timer Z Primary register, it is written to both reload register and counter or written to only reload register. Selected by software. Same applies to Prescaler Z register.
Select function	• Timer Z write control function (Note 2) When a value is written to Timer Z Primary register, it can be selected that the value is written to both reload register and counter or written to only reload register. Same applies to Prescaler Z register.

Note 1: When the count is stopped, the Timer Z interrupt request bit becomes "1" and an interrupt may occur. Thus, interrupts must be disabled before the count is stopped. Furthermore, set the Timer Z interrupt request bit to "0" before starting counting again.

Note 2: If writing to the Timer Z or prescaler Z under the following conditions being filled at the same time the Timer Z interrupt request bit becomes "1" and an interrupt occurs.

<Conditions>

• Timer Z write control bit (bit 6 at address 0080_{16}) is "0" (write to timer and reload register simultaneously)

• Timer Z count start flag (bit 7 at address 0080_{16}) is "1" (count start)

To write to the Timer Z or prescaler Z in the above state, disable interrupts before writing.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0	0			X	

Symbol	Address	When reset
TYZMR	0080_{16}	$000000X0_2$

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y-related bit		RW
Nothing is assigned. When write, set "0". When read, the content is "0".			—
TYWC	Timer Y-related bits		RW
TYS			RW
TZMOD0	Timer Z operation mode bit	b5 b4 0 0 : Timer mode	RW
TZMOD1			RW
TZWC	Timer Z write control bit	0 : Write to timer and reload register simultaneously (Note 1) 1 : Write to reload register	RW
TZS	Timer Z count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW

Note 1: At this bit is "0", when you write in the prescaler Z while the timer Z is counting, the timer Z reloads content of the timer Z reload register.
Note 2: When this bit is cleared to "0", the timer reloads the content of the reload register before it stops. Read out the count value before you stop the timer.

Timer Y, Z waveform output control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUM	0084_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TYPUM0	Timer Y-related bits		RW
TYPUM1			RW
TZPUM0	Timer Z primary waveform extension control bit	Invalid in timer mode	RW
TZPUM1	Timer Z secondary waveform extension control bit	Invalid in timer mode	RW
TYOPL	Timer Y-related bit		RW
TZOPL	Timer Z output level latch	Invalid in timer mode	RW
INOSTG	$\overline{INT_0}$ pin one-shot trigger control bit	Invalid in timer mode	RW
INOSEG	$\overline{INT_0}$ pin one-shot trigger polarity select bit	Invalid in timer mode	RW

Figure 12.25 Timer Y, Z mode register and Timer Y, Z waveform output control register in timer mode

12.4.2 Programmable Waveform Generation Mode

In this mode, the microcontroller, while counting the set values of Timer Z primary and Timer Z secondary alternately, outputs from the TZOUT pin a waveform whose polarity is inverted each time Timer Z secondary underflows. (See Table 12.11) A counting starts by counting the value set in the Timer Z primary. Figure 12.26 shows Timer Y, Z mode register and Timer Y, Z waveform output control register in this mode. The Timer Z operates in the same way as the Timer Y in this mode. See Figure 12.19 shown the Timer Y operating example in programmable waveform generation mode.

Table 12.11 Specifications of programmable waveform generating mode

Item	Specification
Count source	f_1, f_8, Timer Y underflow, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the contents of primary reload register and secondary reload register alternately before continuing counting. • When a counting stops, the timer reloads the content of the reload register before it stops.
Divide ratio	$\frac{fi}{(n+1) \times ((m+1)+(l+1))}$ n: Set value of Prescaler Z, m: Set value of Timer Z primary, l: Set value of Timer Z secondary
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0) (Note 1)
Interrupt request generation timing	When Timer Z underflows during secondary period
TZOUT pin function	Pulse output (Note 2)
INT0 pin function	Programmable I/O port, or external interrupt input pin
Read from timer	Count value can be read out by reading Timer Z primary register. Same applies to Prescaler Z register. (Note 3)
Write to timer	When a value is written to Timer Z primary register, it is written to only reload register. Same applies to Timer Z secondary register and Prescaler Z register. (Note 4)
Select function	• Output level latch select function The output level of an waveform being counted during primary and secondary periods is selectable. • Programmable waveform generation output switching function (Note 5) Can select either programmable waveform or the value of port $P3_1$ register for output. • Waveform extend function (Note 6) The waveform output primary and secondary periods can each be extended 0.5 cycles of the count source. Frequency when waveform extended: 2xfi/((2x(m+1))+(2x(l+1))+TZPUM0+TZPUM1) Duty: (2x(m+1)+TZPUM0)/((2x(m+1)+TZPUM0)+(2x(l+1)+TZPUM1)) m: set value of Timer Z primary, l: set value of Timer Z secondary TZPUM0: Timer Z primary waveform extension control bit TZPUM1: Timer Z secondary waveform extension control bit

Note 1: When the count is stopped, the Timer Z interrupt request bit becomes "1" and an interrupt may occur. Thus, interrupts must be disabled before the count is stopped. Furthermore, set the Timer Z interrupt request bit to "0" before starting counting again.
Note 2: When the counting stopped, the pin is the secondary period output level.
Note 3: Even when counting the secondary period, read out the Timer Z primary register.
Note 4: The set value of Timer Z secondary register and waveform extension control bits as well as Timer Z primary register are made effective by writing a value to the Timer Z primary register. The written values are reflected to the waveform output from the next primary period after writing to the Timer Z primary register.
Note 5: The output is switched in sync with Timer Z secondary underflow.
Note 6: When using the waveform extend function, the Prescaler Z register must be set to "00_{16}".
When selecting Timer Y underflow and f_1 for the count source, the waveform extend function cannot be used.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
	1	0	1			X	

Symbol	Address	When reset
TYZMR	0080_{16}	$000000X0_2$

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y-related bit		RW
Nothing is assigned. When write, set "0". When read, the content is "0".			—
TYWC	Timer Y-related bits		RW
TYS			RW
TZMOD0	Timer Z operation mode bit	b5 b4 0 1 : Programmable waveform generation mode (Note 1)	RW
TZMOD1			RW
TZWC	Timer Z write control bit	1 : Set to "1" in programmable waveform generation mode	RW
TZS	Timer Z count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW

Note 1: When selecting programmable waveform generation mode, output is set for Port $P3_1$ regardless of the value of the direction register.
Note 2: When this bit is cleared to "0", the timer reloads the content of the reload register before it stops. Read out the count value before you stop the timer.

Timer Y, Z waveform output control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUM	0084_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TYPUM0	Timer Y-related bits		RW
TYPUM1			RW
TZPUM0	Timer Z primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1)	RW
TZPUM1	Timer Z secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1)	RW
TYOPL	Timer Y-related bit		RW
TZOPL	Timer Z output level latch	0 : Outputs "H" for the period set by Timer Z primary and "L" for the period set by Timer Z secondary. "L" is outputted when the timer is stopped. 1 : Outputs "L" for the period set by Timer Z primary and "H" for the period set by Timer Z secondary. "H" is outputted when the timer is stopped.	RW
INOSTG	$\overline{INT_0}$ pin one-shot trigger control bit	Invalid in programmable waveform generation mode	RW
INOSEG	$\overline{INT_0}$ pin one-shot trigger polarity select bit	Invalid in programmable waveform generation mode	RW

Note 1: When setting this bit to "1", the Prescaler Z Register must be set to "00_{16}".

Figure 12.26 Timer Y, Z mode register and Timer Y, Z waveform output control register in programmable waveform generation mode

12.4.3 Programmable One-shot Generation Mode

In this mode, upon software command or external trigger input (input to the INT0 pin), the microcomputer outputs the one-shot pulse from the TZOUT pin. (See Table 12.12) When a trigger occurs, the timer starts operating from the point only once for a given period equal to the set value of the Timer Z primary. Timer Z secondary is unused in this mode.

Table 12.12 lists specifications of programmable one-shot generating mode. Figure 12.27 shows the Timer Y, Z mode register and Timer Y, Z waveform output control register in this mode. Figure 12.28 shows the Timer Z operation example in this mode.

Table 12.12 Specifications of programmable one-shot generating mode

Item	Specification
Count source	f1, f8, Timer Y underflow, fC32
Count operation	• Down counts the set value of Timer Z primary • When the timer underflows, it reloads the contents of reload register before stopping counting. • When a counting stops, the timer reloads the contents of the reload register before it stops.
Divide ratio	$\frac{1}{(n+1) \times (m+1)}$ n: Set value of Prescaler Z, m: Set value of Timer Z primary
Count start condition	• Timer Z one-shot start bit is set (=1) (Note 1) • Valid trigger is input to INT0 pin (Note 2)
Count stop condition	• When reloading is completed after count value was set to "0016" • When Count start flag is reset (=0) • Timer Z one-shot start bit is reset (=0) (Note 3)
Interrupt request generation timing	When count value becomes "0016"
TZOUT pin function	Pulse output
INT0 pin function	Programmable I/O port, external interrupt input pin, or external trigger input pin
Read from timer	Count value can be read out by reading Timer Z primary register. Same applies to Prescaler Z register.
Write to timer	When a value is written to Timer Z primary register, it is written to only reload register. Same applies to Prescaler Z register. (Note 4)
Select function	• Output level latch select function The output level of one-shot pulse waveform is selectable. • INT0 pin one-shot trigger control function and polarity select function The trigger input from the INT0 pin can be set to valid or invalid. Also, the valid trigger's polarity can be chosen to be the rising edge, falling edge, or rising and falling both edges. • Waveform extend function (Note 5) The one-shot pulse waveform can be extended 0.5 cycles of the count source. Frequency when waveform extended: 2xfi/(n+1)/(2x(m+1)+TZPUM0) n: set value of Prescaler Z, m: set value of Timer Z primary TZPUM0: Timer Z primary waveform extension control bit

Note 1: Count start flag must have been set to "1".
Note 2: Count start flag must have been set to "1", $\overline{\text{INT0}}$ input enable bit [INT0EN] to "1", and $\overline{\text{INT0}}$ pin one-shot trigger control bit to "1".
Note 3: When the count is stopped by writing "0" to the count start flag or Timer Z one-shot start bit, the Timer Z interrupt request bit becomes "1" and an interrupt may occur. Thus, interrupts must be disabled before the count is stopped. Furthermore, set the Timer Z interrupt request bit to "0" before starting counting again.
Note 4: Each set value becomes effective by writing to the Timer Z primary register. And the set values are reflected collectively beginning with the next one-shot pulse after writing to the Timer Z primary.
Note 5: When using the waveform extend function, the Prescaler Z register must be set to "0016".
When selecting Timer Y underflow and f1 for the count source, the waveform extend function cannot be used.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
	1	1	0			X	

Symbol	Address	When reset
TYZMR	0080_{16}	$000000X0_2$

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y-related bit		RW
Nothing is assigned. When write, set "0". When read, the content is "0".			—
TYWC	Timer Y-related bits		RW
TYS			RW
TZMOD0	Timer Z operation mode bit	b5 b4 1 0 : Programmable one-shot generation mode (Note 1)	RW
TZMOD1			RW
TZWC	Timer Z write control bit	1 : Set to "1" in programmable one-shot generation mode	RW
TZS	Timer Z count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW

Note 1: When selecting programmable one-shot generation mode, output is set for Port $P3_1$ regardless of the value of the direction register.
Note 2: When this bit is cleared to "0", the timer reloads the content of the reload register before it stops. Read out the count value before you stop the timer.

Timer Y, Z waveform output control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUM	0084_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TYPUM0	Timer Y-related bits		RW
TYPUM1			RW
TZPUM0	Timer Z primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1)	RW
TZPUM1	Timer Z secondary waveform extension control bit	Invalid in programmable one-shot generation mode	RW
TYOPL	Timer Y-related bit		RW
TZOPL	Timer Z output level latch	0 : Outputs "H" level one-shot pulse. "L" is outputted when the timer is stopped. 1 : Outputs "L" level one-shot pulse "H" is outputted when the timer is stopped.	RW
INOSTG	$\overline{INT_0}$ pin one-shot trigger control bit	0 : $\overline{INT_0}$ pin one-shot trigger invalid 1 : $\overline{INT_0}$ pin one-shot trigger valid (Note 2)	RW
INOSEG	$\overline{INT_0}$ pin one-shot trigger polarity select bit (Note 3)	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	RW

Note 1: When setting this bit to "1", the Prescaler Z Register must be set to "00_{16}".
Note 2: When changing this bit to "1", set the $\overline{INT_0}$ input filter select bit (bit 0 at address 0096_{16}), the $\overline{INT_0}$ input polarity select bit (bit 1 at address 0096_{16}), the $\overline{INT_0}$ input filter select bit (bits 0 and 1 at address $001E_{16}$) and the $\overline{INT_0}$ pin one-shot trigger polarity select bit.
Note 3: This bit is valid only when the $\overline{INT_0}$ input polarity select bit (bit 1 at address 0096_{16}) is "0" (one-edge).

Figure 12.27 Timer Y, Z mode register and Timer Y, Z waveform output control register in programmable one-shot generation mode

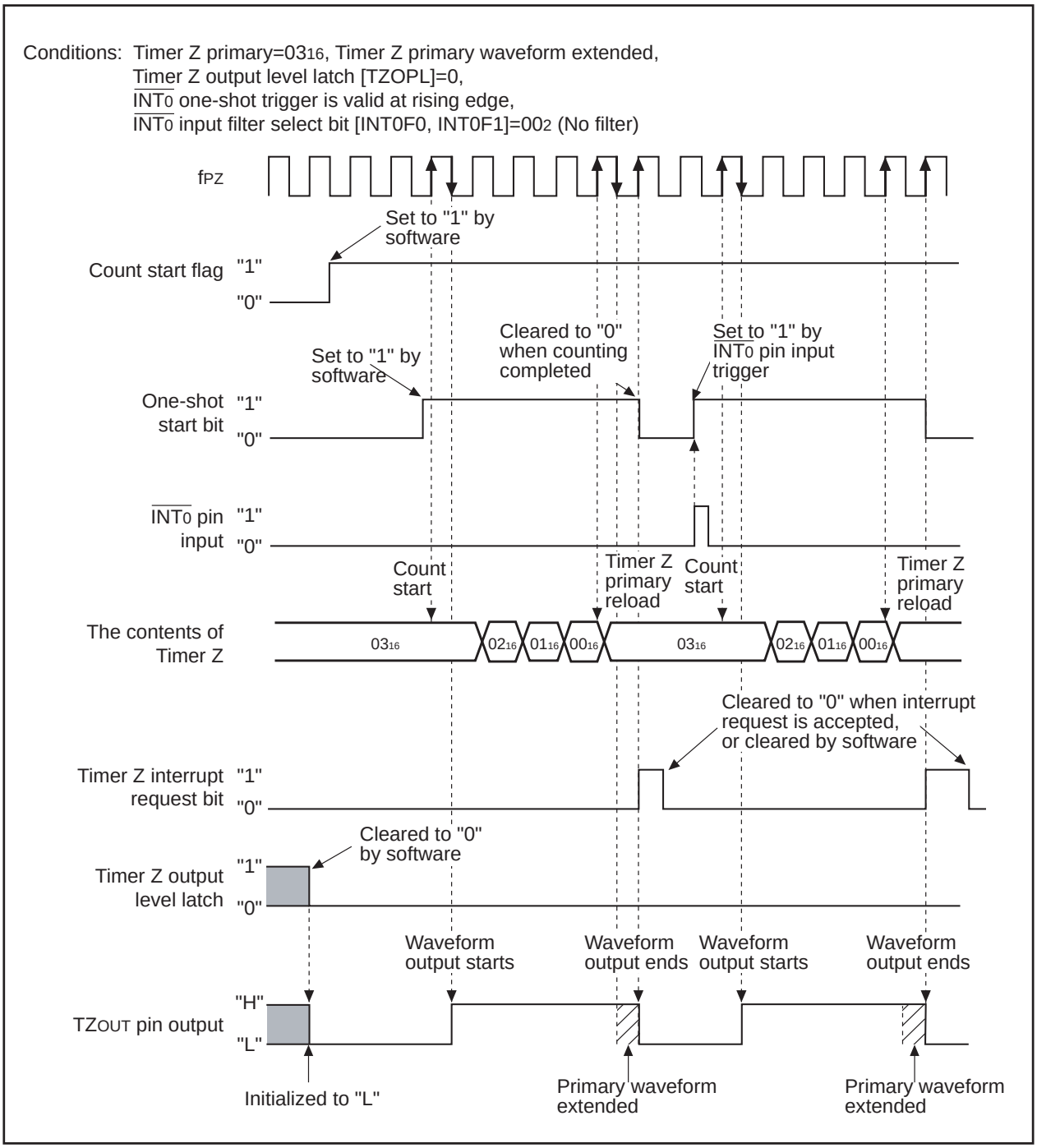


Figure 12.28 Operation example in programmable one-shot generation mode

12.4.4 Programmable Wait One-shot Generation Mode

In this mode, upon software command or external trigger input (input to the INT0 pin), the microcomputer outputs the one-shot pulse from the TZOUT pin after waiting for a given length of time. (See Table 12.13) When a trigger occurs, from this point, the timer starts outputting pulses only once for a given length of time equal to the Timer Z primary set value after waiting for a given length of time equal to the Timer Z primary set value. Figure 16.29 shows the Timer Y, Z mode register and Timer Y, Z waveform output control register in this mode. Figure 12.30 shows the Timer Z operation example in this mode.

Table 12.13 Specifications of programmable wait one-shot generating mode

Item	Specification
Count source	f1, f8, Timer Y underflow, fC32
Count operation	• Down counts the set value of Timer Z primary • When Timer Z primary underflows, the contents of Timer Z secondary is reloaded before continuing counting. • When Timer Z secondary underflows, the contents of Timer Z primary is reloaded before stopping counting. • When a counting stops, the timer reloads the contents of the reload register before it stops.
Wait time	(n+1) x (m+1)/fi, n: Set value of Prescaler Z, m: Set value of Timer Z primary
One-shot pulse output time	(n+1) x (l+1))/fi, n: Set value of Prescaler Z, l: Set value of Timer Z secondary
Count start condition	• Timer Z one-shot start bit is set (=1) (Note 1) • Valid trigger is input to INT0 pin (Note 2)
Count stop condition	• When reloading is completed after count value at counting Timer Z secondary was set to "0016" • When Count start flag is reset (=0) • Timer Z one-shot start bit is reset (=0) (Note 3)
Interrupt request generation timing	When count value at counting Timer Z secondary becomes "0016"
TZOUT pin function	Pulse output
INT0 pin function	Programmable I/O port, external interrupt input pin, or external trigger input pin
Read from timer	Count value can be read out by reading Timer Z primary register. Same applies to Prescaler Z register.
Write to timer	When a value is written to Timer Z primary register, it is written to only reload register. Same applies to Prescaler Z register. (Note 4)
Select function	• Output level latch select function The output level of one-shot pulse waveform is selectable. • INT0 pin one-shot trigger control function and polarity select function The trigger input from the INT0 pin can be set to valid or invalid. Also, the valid trigger's polarity is selectable: rising edge, falling edge, or rising and falling both edges. • Waveform extend function (Note 5) Waiting time and one-shot pulse waveform can each be extended 0.5 cycles of the count source. Waiting time when waveform extended: (n+1) x (2x(m+1)+TZPUM0)/2fi One-shot pulse output time when waveform extended: (n+1) x (2x(l+1)+TZPUM1)/2fi n: set value of Prescaler Z, m: set value of Timer Z primary, l: set value of Timer Z secondary TZPUM0: Timer Z primary waveform extension control bit, TZPUM1: Timer Z secondary waveform extension control bit

Note 1: Count start flag must have been set to "1".
Note 2: Count start flag must have been set to "1", $\overline{\text{INT0}}$ input enable bit [INT0EN] to "1", and $\overline{\text{INT0}}$ pin one-shot trigger control bit to "1".
Note 3: When the count is stopped by writing "0" to the count start flag or Timer Z one-shot start bit, the Timer Z interrupt request bit becomes "1" and an interrupt may occur. Thus, interrupts must be disabled before the count is stopped. Furthermore, set the Timer Z interrupt request bit to "0" before starting counting again.
Note 4: Each set value becomes effective by writing to the Timer Z primary register. And the set values are reflected collectively beginning with the next one-shot pulse after writing to the Timer Z primary.
Note 5: When using the waveform extend function, the Prescaler Z register must be set to "0016".
When selecting Timer Y underflow and f1 for the count source, the waveform extend function cannot be used.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
	1	1	1			X	

Symbol	Address	When reset
TYZMR	0080₁₆	000000X0₂

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y-related bit		RW
Nothing is assigned. When write, set "0". When read, the content is "0".			—
TYWC	Timer Y-related bits		RW
TYS			RW
TZMOD0	Timer Z operation mode bit	b5 b4 1 1 : Programmable wait one-shot generation mode (Note 1)	RW
TZMOD1			RW
TZWC	Timer Z write control bit	1 : Set to "1" in programmable wait one-shot generation mode	RW
TZS	Timer Z count start flag	0 : Stops counting (Note 2) 1 : Starts counting	RW

Note 1: When selecting programmable wait one-shot generation mode, output is set for Port P3₁ regardless of the value of the direction register.
Note 2: When this bit is cleared to "0", the timer reloads the content of the reload register before it stops. Read out the count value before you stop the timer.

Timer Y, Z waveform output control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUM	0084₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TYPUM0	Timer Y-related bits		RW
TYPUM1			RW
TZPUM0	Timer Z primary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1)	RW
TZPUM1	Timer Z secondary waveform extension control bit	0 : No waveform extension 1 : Waveform extension (Note 1)	RW
TYOPL	Timer Y-related bit		RW
TZOPL	Timer Z output level latch	0 : Outputs "H" level one-shot pulse. "L" is outputted when the timer is stopped. 1 : Outputs "L" level one-shot pulse. "H" is outputted when the timer is stopped.	RW
INOSTG	$\overline{INT_0}$ pin one-shot trigger control bit	0 : $\overline{INT_0}$ pin one-shot trigger invalid 1 : $\overline{INT_0}$ pin one-shot trigger valid (Note 2)	RW
INOSEG	$\overline{INT_0}$ pin one-shot trigger polarity select bit (Note 3)	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	RW

Note 1: When setting this bit to "1", the Prescaler Z Register must be set to "00₁₆".
Note 2: When changing this bit to "1", set the $\overline{INT_0}$ input filter select bit (bit 0 at address 0096₁₆), the $\overline{INT_0}$ input polarity select bit (bit 1 at address 0096₁₆), the $\overline{INT_0}$ input filter select bit (bits 0 and 1 at address 001E₁₆) and the $\overline{INT_0}$ pin one-shot trigger polarity select bit.
Note 3: This bit is valid only when the $\overline{INT_0}$ input polarity select bit (bit 1 at address 0096₁₆) is "0" (one-edge).

Figure 12.29 Timer Y, Z mode register and Timer Y, Z waveform output control register in programmable wait one-shot generation mode

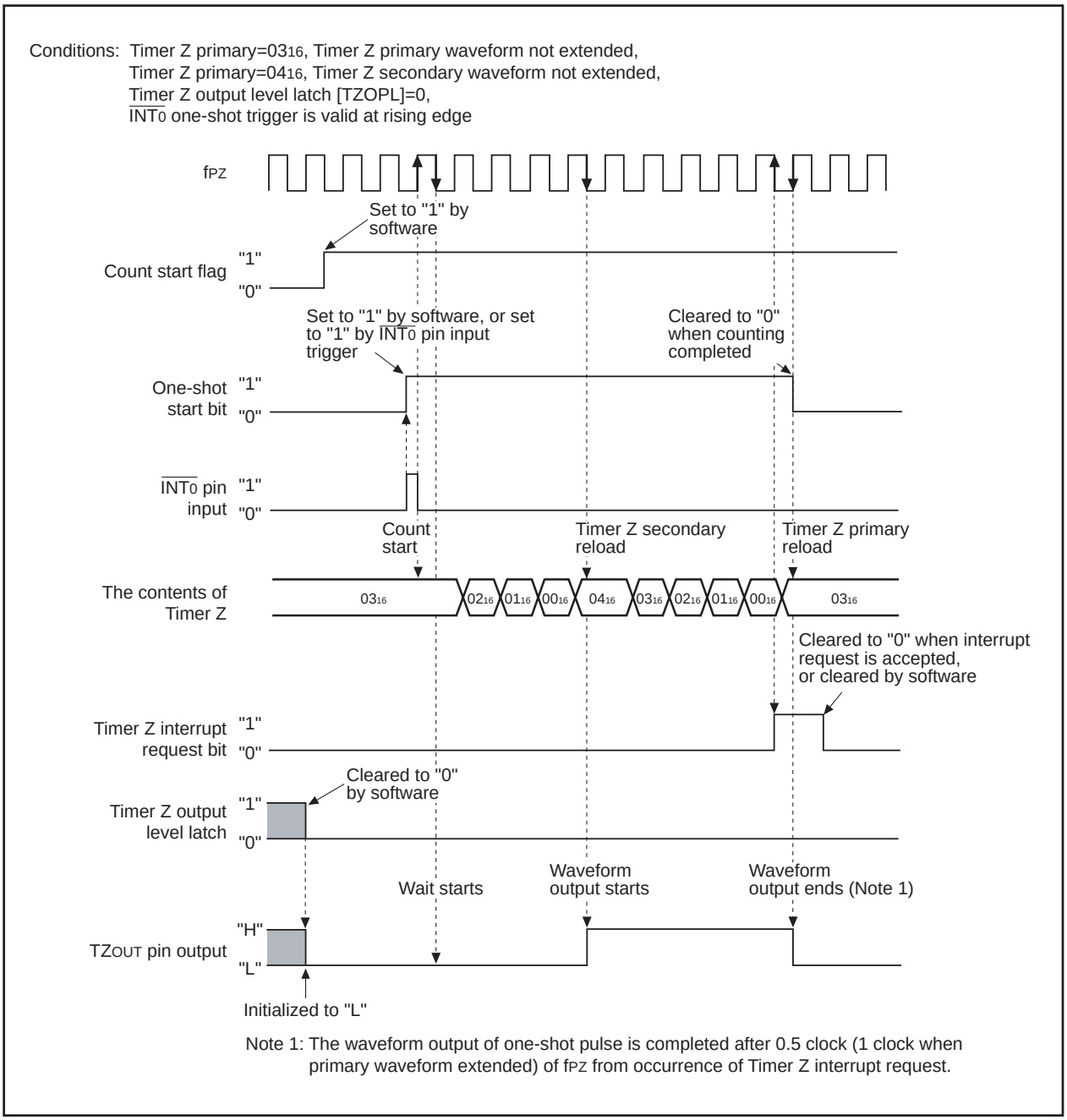


Figure 12.30 Operation example in programmable wait one-shot generation mode

12.5 Timer C

Timer C is a 16-bit free-running timer. The Timer C uses an edge input to TCIN pin or the output of 256 fRING divisions as trigger to latch the timer count value and generates an interrupt request. The TCIN input has a digital filter and this prevents an error caused by noise or so on from occurring.

Figure 12.31 shows the block diagram of Timer C. Table 12.14 shows Timer C specifications. Figure 12.32 shows Timer C-related registers. Figure 12.33 shows an operation example of Timer C and timer measurement register.

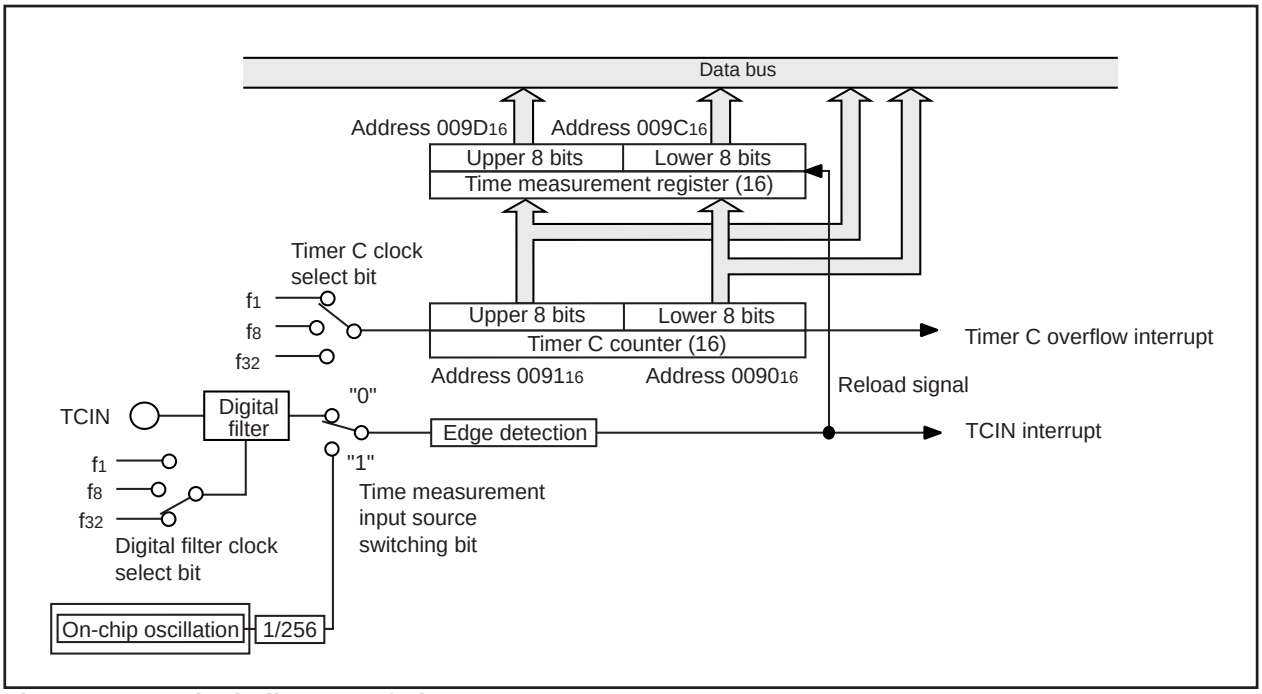


Figure 12.31 Block diagram of Timer C

Table 12.14 Specifications of Timer C

Item	Specification
Count source	f1, f8, f32
Count operation	• Up count • Transfer counter value to time measurement register at active edge of measurement pulse • When timer C stops counting, the value of timer C is reset to "000016".
Count start condition	• Time measurement control bit is set (=1)
Counter stop condition	• Time measurement control bit is reset (=0)
Interrupt request generation timing	• When active edge of measurement pulse is input [TCIN interrupt] • When the time underflows [Timer C interrput]
TCIN pin function	Measurement pulse input
Count value reset timing	When time measurement control bit is reset (=0)
Read from timer (Note 1)	• Count value can be read out by reading Timer C. • Count value at measurement pulse active edge input can be read out by reading time measurement register.
Write to timer	Cannot write to Timer C and time measurement register
Select function	• Measurement pulse active edge: selectable (rising edge/falling edge/both edges) • Measurement pulse: selectable (input from TCIN pin/256 divisions of fRING) • Digital filter sampling frequency: selectable (f1/f8/f32)

Note 1: The Timer C and the timer measurement register must be read in word-size.

Timer C

(b15) b7 (b8) b0 b7 b0

Symbol	Address	When reset
TC	0091₁₆, 0090₁₆	Indeterminate

Function	RW
Internal count source is counted	RO

Time measurement register

(b15) b7 (b8) b0 b7 b0

Symbol	Address	When reset
TM	009C₁₆, 009D₁₆	Indeterminate

Function	RW
When active edge of measurement pulse is input, the count value of Timer C is stored (Note 1)	RO

Note 1: When time measurement is disabled, the value is indeterminate. After enabling time measurement, the value is indeterminate until the first trigger is generated.

Timer C control register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TCC0	009A₁₆	0XX00000₂

Bit symbol	Bit name	Function	RW
TCC00	Time measurement control bit	0 : Time measurement disabled 1 : Time measurement enabled	RW
TCC01	Timer C clock select bit (Note 1)	b2 b1 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : Inhibited	RW
TCC02			RW
TCC03	Time measurement input edge trigger bit (Note 1)	b4 b3 0 0 : Rising edge 0 1 : Falling edge 1 0 : Both edges 1 1 : Inhibited	RW
TCC04			RW
Nothing is assigned. When write, set "0". When read, their contents are "0".			—
TCC07	Time measurement input source switching bit (Note 1 to 3)	0 : TCIN 1 : fRING256	RW

Note 1: Change this bit when time measurement is disabled.
Note 2: Set the on-chip oscillation stop bit (CM14) to "0" before setting this bit to "1".
Note 3: Change this bit when the interrupt is disabled. When switching the timer measurement input source, the TCIN interrupt may be requested. Therefore, enable the interrupt after setting the interrupt request bit to "0".

Timer C control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TCC1	009B₁₆	XXXXXX11₂

Bit symbol	Bit name	Function	RW
TCC10	Digital filter clock select bit (Note 1)	b1 b0 0 0 : Inhibited 0 1 : f1 1 0 : f8 1 1 : f32	RW
TCC11			RW
Nothing is assigned. When write, set "0". When read, their contents are "0".			—

Note 1: Input edge becomes active when the same value from TCIN pin is sampled three times in succession.

Figure 12.32 Timer C-related register

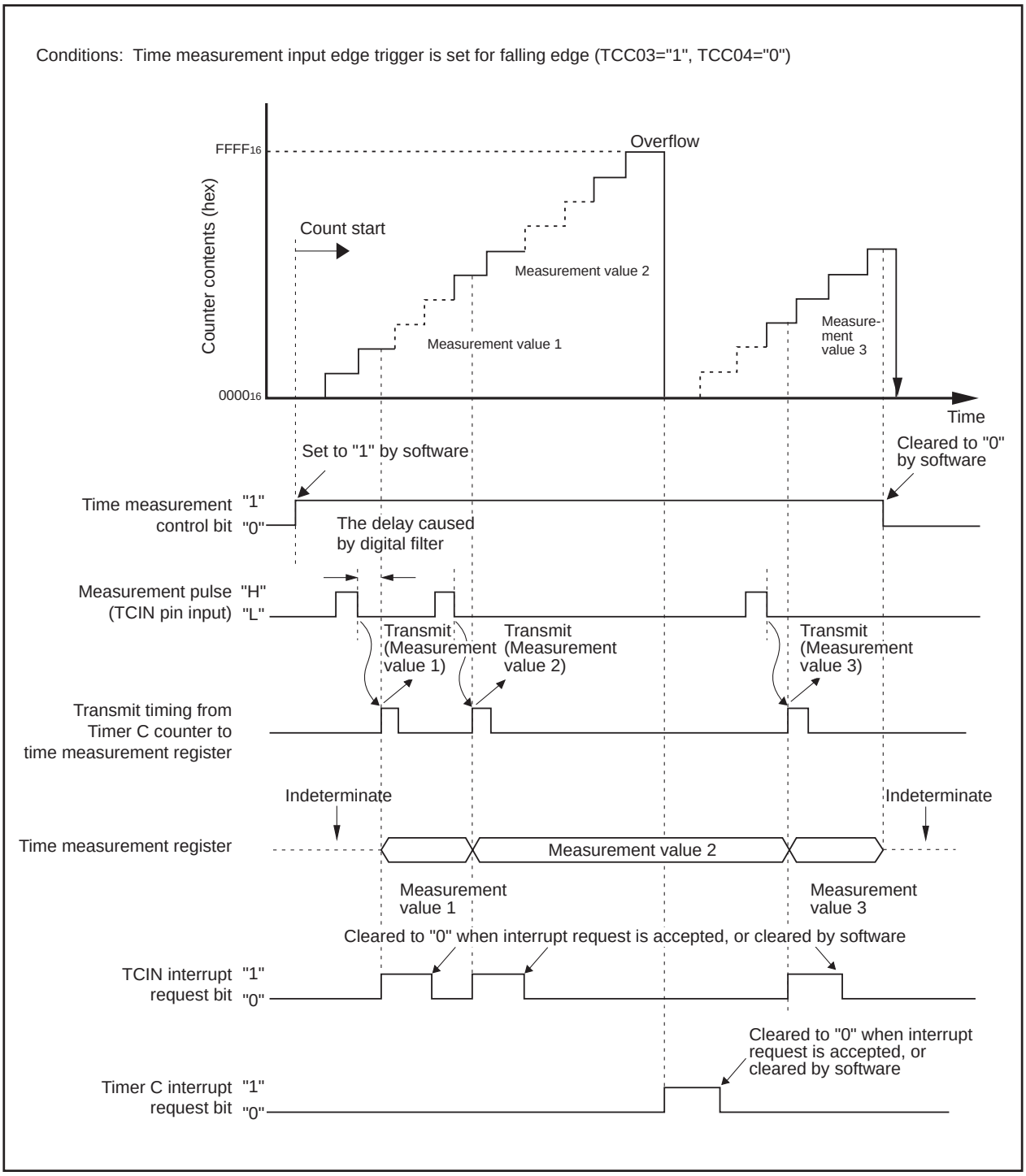


Figure 12.33 Operation example of Timer C and time measurement register

13. Serial I/O

Serial I/O is configured as two channels: UART0 and UART1. UART0 and UART1 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 13.1 shows the block diagram of UARTi (i=0,1). Figure 13.2 shows the block diagram of the transmit/receive unit.

UART0 has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode select bits (bits 0 to 2 at addresses 00A0$_{16}$ and 00A8$_{16}$) determine whether UART0 is used as a clock synchronous serial I/O or as a UART. Although a few functions are different, UART0 and UART1 have almost the same functions.

Figures 13.3 through 13.5 show the registers related to UARTi.

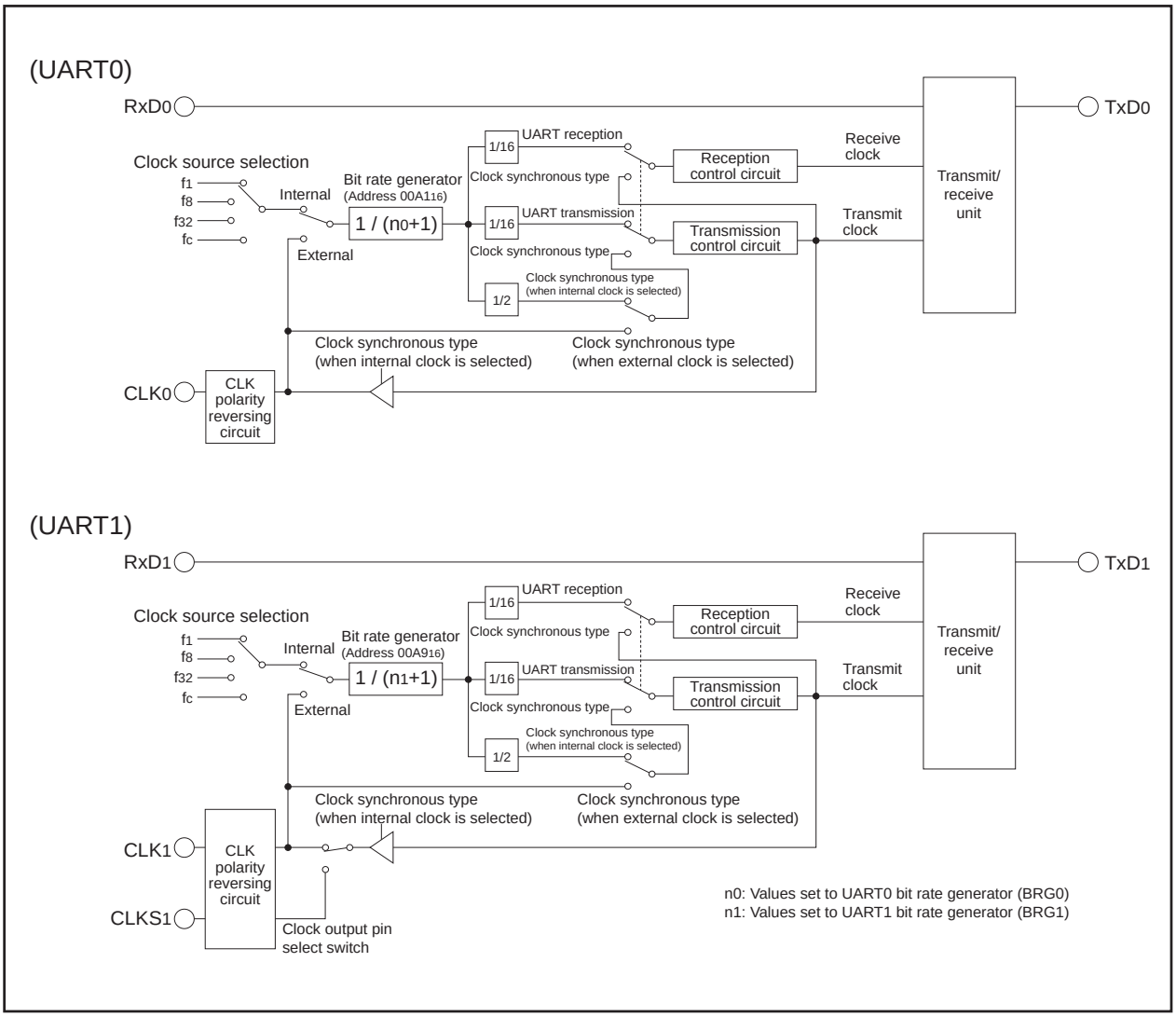


Figure 13.1 Block diagram of UARTi (i= 0, 1)

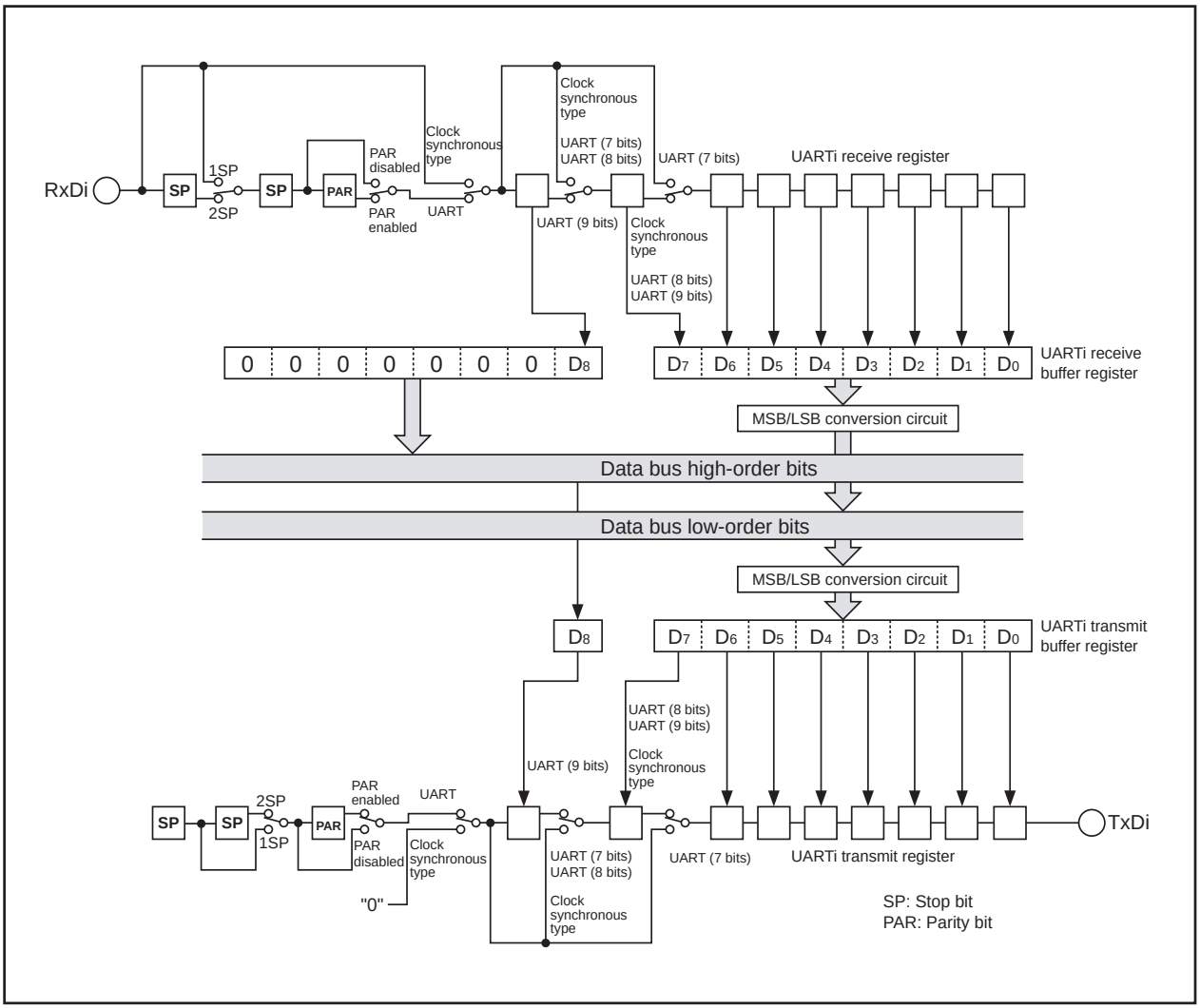
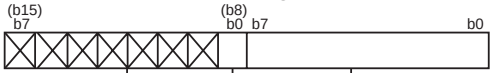


Figure 13.2 Block diagram of transmit/receive unit

UARTi transmit buffer register

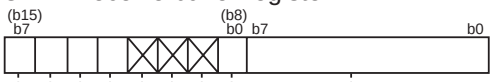


Symbol	Address	When reset
U0TB	$00A3_{16}$, $00A2_{16}$	Indeterminate
U1TB	$00AB_{16}$, $00AA_{16}$	Indeterminate

Function	RW
Transmit data (Note 1)	WO
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.	—

Note 1: When transfer data length is 9-bit long, write high-byte first then low-byte with byte-size.

UARTi receive buffer register

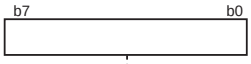


Symbol	Address	When reset
U0RB	$00A7_{16}$, $00A6_{16}$	Indeterminate
U1RB	$00AF_{16}$, $00AE_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	RW
——	———	Receive data	Receive data	RO
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.				—
OER	Overrun error flag (Note 1)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	RO
FER	Framing error flag (Note 1)	Invalid	0 : No framing error 1 : Framing error found	RO
PER	Parity error flag (Note 1)	Invalid	0 : No parity error 1 : Parity error found	RO
SUM	Error sum flag (Note 1)	Invalid	0 : No error 1 : Error found	RO

Note 1: Bits 15 to 12 are set to "0" when the serial I/O mode select bits (bit 2 to 0 at addresses $00A0_{16}$ and $00A8_{16}$) are set to "000_2" or receive enable bit to "0". (Bit 15 is set to "0" when bits 14 to 12 all are set to "0".) Bits 14 and 13 are also set to "0" when the lower byte of the UARTi receive buffer register (addresses $00A6_{16}$, and $00AE_{16}$) is read out or when this register is read out in word-size. When reading data from the UARTi receive buffer, data should be read high-byte first then low-byte using byte-size.

UARTi bit rate generator



Symbol	Address	When reset
U0BRG	$00A1_{16}$	Indeterminate
U1BRG	$00A9_{16}$	Indeterminate

Function	Values that can be set	RW
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	WO

Figure 13.3 Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol: UiMR(i=0,1) — Address: 00A0$_{16}$, 00A8$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	RW
SMD0	Serial I/O mode select bit	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	RW
SMD1				RW
SMD2				RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 1)	0 : Internal clock 1 : External clock	RW
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	RW
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = "1" 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	RW
Reserved bit		Set to "0"		RW

Note 1: Set the corresponding port direction register to "0".

UARTi transmit/receive control register 0

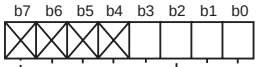
b7	b6	b5	b4	b3	b2	b1	b0
			X		0		

Symbol: UiC0(i=0,1) — Address: 00A4$_{16}$, 00AC$_{16}$ — When reset: 08$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	RW
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	RW
CLK1				RW
Reserved bit		Set to "0"		RW
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	RO
Nothing is assigned		In an attempt to write to this bit, write "0". The value, if read, turn out to be "0".		—
NCH	Data output select bit	0 : TxDi pin is CMOS output 1 : TxDi pin is N-channel open-drain output	0 : TxDi pin is CMOS output 1 : TxDi pin is N-channel open-drain output	RW
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Set to "0"	RW
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	Set to "0"	RW

Figure 13.4 Serial I/O-related registers (2)

UARTi transmit/receive control register 1



Symbol	Address	When reset
UiC1(i=0,1)	$00A5_{16}$, $00AD_{16}$	$XXXX0010_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	RO
RE	Receive enable bit (Note 1)	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	RO
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".				—

Note 1: As for the UART1, set the RxD_1 input port select bit before setting this bit to reception enabled.

UART transmit/receive control register 2

Symbol	Address	When reset
UCON	$00B0_{16}$	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	RW
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U1IRS	UART1 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Invalid	RW
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Invalid	RW
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = "1" 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	RW
CLKMD1	CLK/CLKS select bit 1 (Note 1)	0 : Normal mode (CLK output is CLK0 only) 1 : Transfer clock output from multiple pins function selected	Set to "0"	RW
RXD1EN	RxD_1 input port select bit (Note 2)	0 : $P3_7$ 1 : $P3_5$	0 : $P3_7$ 1 : $P3_5$	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".				—

Note 1: When using multiple pins to output the transfer clock, the following requirements must be met:
- UART1 internal/external clock select bit (bit 3 at address $00A8_{16}$) = "0".

Note 2: For $P3_7$, select "0" for data receive, and "1" for data transfer.
And set the direction register of port $P3_7$ to input ("0") when receiving.

Figure 13.5 Serial I/O-related registers (3)

13.1 Clock Synchronous Serial I/O Mode

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

Table 13.1 lists specifications of clock synchronous sperial I/O mode. Figure 13.6 shows the UARTi transmit/receive mode register.

Table 13.1 Specifications of clock synchronous serial I/O mode

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at address 00A0$_{16}$,00A8$_{16}$ = "0"): fi/ 2(n+1) (Note 1) fi = f$_1$, f$_8$, f$_{32}$, fc • When external clock is selected (bit 3 at address 00A0$_{16}$,00A8$_{16}$ = "1"): Input from CLKi pin
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at address 00A5$_{16}$,00AD$_{16}$) = "1" – Transmit buffer empty flag (bit 1 at addresses 00A5$_{16}$,00AD$_{16}$) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at address 00A4$_{16}$,00AC$_{16}$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at address 00A4$_{16}$,00AC$_{16}$) = "1": CLKi input level = "L"
Reception start conditio	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at address 00A5$_{16}$,00AD$_{16}$) = "1" – Transmit enable bit (bit 0 at address 00A5$_{16}$,00AD$_{16}$) = "1" – Transmit buffer empty flag (bit 1 at address 00A5$_{16}$,00AD$_{16}$) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at address 00A4$_{16}$,00AC$_{16}$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at address 00A4$_{16}$,00AC$_{16}$) = "1": CLKi input level = "L"
Interrupt request generation timing	• When transmitting – Transmit interrupt cause select bit (bit 0 and bit 1 at address 00B0$_{16}$) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed – Transmit interrupt cause select bit (bit 0 and bit 1 at address 00B0$_{16}$) = "1": Interrupts requested when data transmission from UARTi transfer register is completed • When receiving – Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed
Error detection	• Overrun error (Note 2) This error occurs if the serial I/O started receiving the next data before reading UARTi receive buffer register and received the 7th bit of the next data
Select function	• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected • LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected • Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register • Transfer clock output from multiple pins selection UART1 transfer clock can be chosen by software to be output from one of the two pins set • RxD$_1$ input pin selection UART1 RxD$_1$ can be chosen by software to be input to one of the two pins set

Note 1: "n" denotes the value 00$_{16}$ to FF$_{16}$ that is set to the UARTi bit rate generator.

Note 2: If an overrun error occurs, the UARTi receive buffer will be indeterminate. Note also that the UARTi receive interrupt request bit does not change.

UARTi transmit/receive mode registers

b7	b6	b5	b4	b3	b2	b1	b0
0					0	0	1

Symbol: UiMR (i=0,1) | Address: $00A0_{16}$, $00A8_{16}$ | When reset: 00_{16}

Bit symbol	Bit name	Function	RW
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 1)	RW
STPS	Invalid in clock synchronous serial I/O mode		RW
PRY			RW
PRYE			RW
Reserved bit		Set to "0"	RW

Note 1: Set the corresponding port direction register to "0".

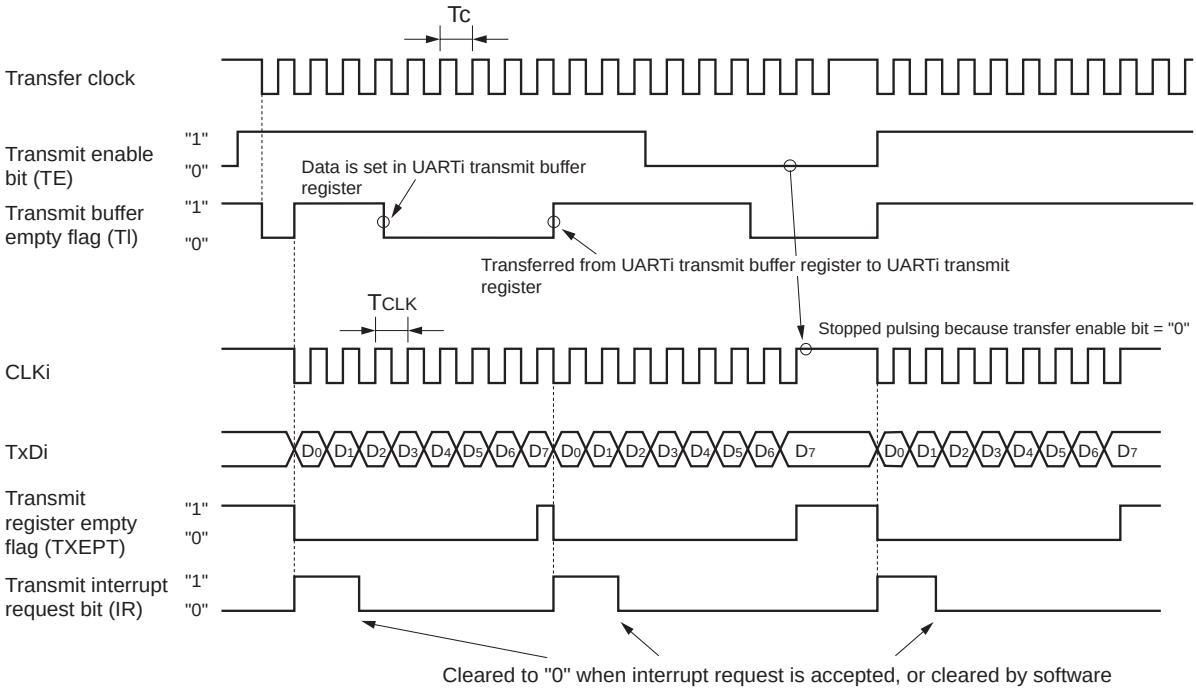
Figure 13.6 UARTi transmit/receive mode register in clock synchronous serial I/O mode

Table 13.2 lists the functions of the input/output pins during clock synchronous serial I/O mode. This table shows the pin functions when the transfer clock output from multiple pins is not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table 13.2 Input/output pin functions in clock synchronous serial I/O mode

Function	Pin name	Method of selection	Remarks
Serial data output	TxD_0 ($P1_4$)	———	Port $P1_4$ cannot be used as an I/O port even when performing only serial data input but not serial data output.
	TxD_1 ($P3_7$)	RxD_1 input pin select bit (bit 6 at address $00B0_{16}$)="1"	Port $P3_7$ cannot be used as an I/O port even when performing only serial data input but not serial data output.
Serial data input	RxD_0 ($P1_5$)	Port $P1_5$ direction register (bit 5 at address $00E3_{16}$)="0"	Port $P1_5$ can be used as an I/O port when performing only serial data output but not serial data input.
	RxD_1 ($P3_5$)	Port $P3_5$ direction register (bit 5 at address $00E7_{16}$)="0" RxD_1 input pin select bit (bit 6 at address $00B0_{16}$)="1"	Port $P3_5$ can be used as an I/O port when performing only serial data output but not serial data input.
	RxD_1 ($P3_7$)	Port $P3_7$ direction register (bit 7 at address $00E7_{16}$)="0" RxD_1 input pin select bit (bit 6 at address $00B0_{16}$)="0"	When setting Port $P3_7$ as RxD_1, serial data output cannot be performed. Port $P3_5$ can be used as an I/O port.
Transfer clock output	CLKi ($P1_6$, $P3_6$)	Internal/external clock select bit (bit 3 at addresses $00A0_{16}$ and $00A8_{16}$)="0"	———
Transfer clock input	CLKi ($P1_6$, $P3_6$)	Internal/external clock select bit (bit 3 at address $00A0_{16}$ and $00A8_{16}$)="1" Ports $P1_6$ and $P3_6$ direction register (bit 6 at address $00E3_{16}$ and $00E7_{16}$)="0"	———

• Example of transmit timing (when internal clock is selected)

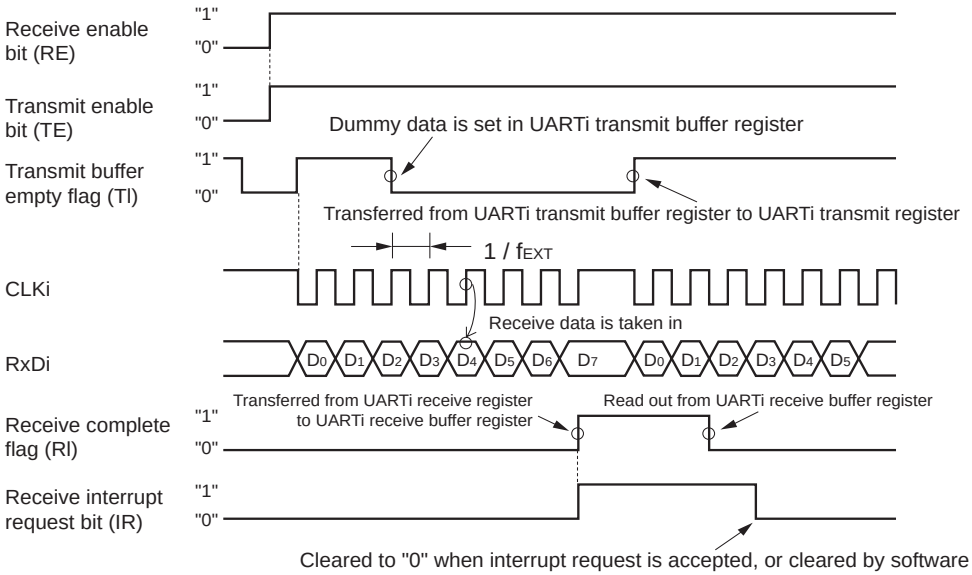


Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- Internal clock is selected.
- CLK polarity select bit = 0.
- Transmit interrupt cause select bit = 0.

$Tc = T_{CLK} = 2(n + 1) / fi$
fi: frequency of BRGi count source (f_1, f_8, f_{32}, fc)
n: value set to BRGi

• Example of receive timing (when external clock is selected)



Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- External clock is selected.
- CLK polarity select bit = 0.

f_{EXT}: frequency of external clock

Meet the following conditions when the CLKi input level before data reception = "H"
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UARTi transmit buffer register

Figure 13.7 Typical transmit/receive timings in clock synchronous serial I/O mode

13.1.1 Polarity Select Function

As shown in Figure 13.8, the CLK polarity select bit (bit 6 at addresses 00A4$_{16}$ and 00AC$_{16}$) allows selection of the polarity of the transfer clock.

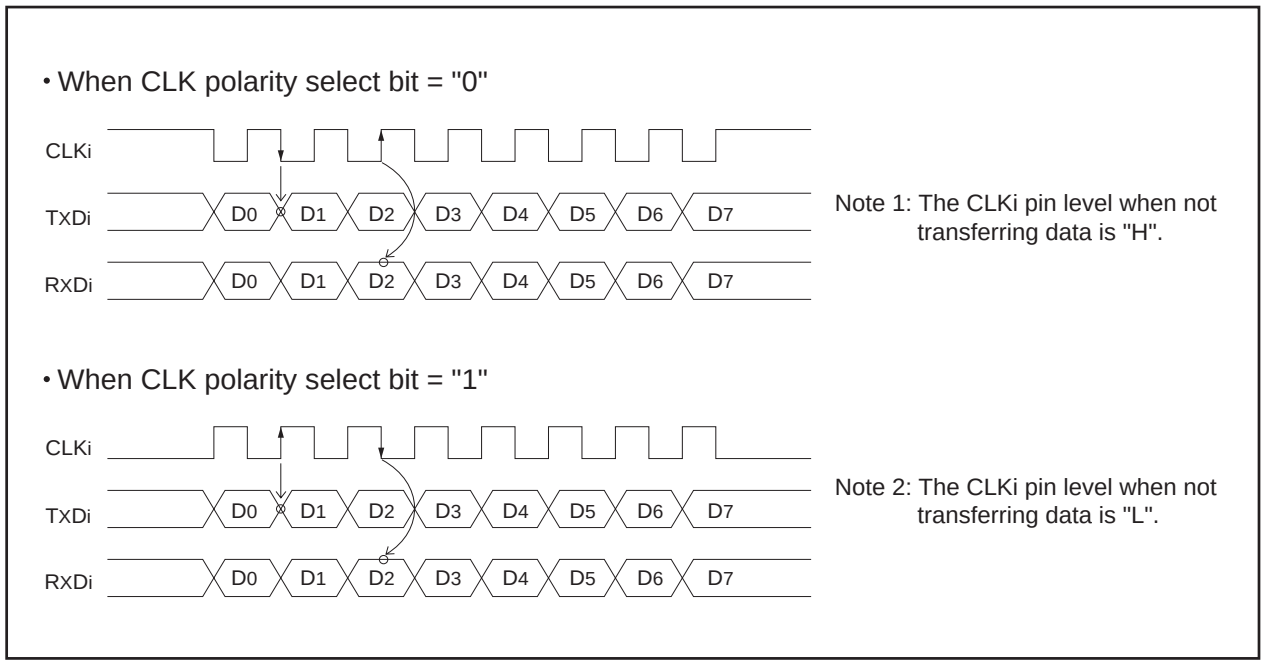


Figure 13.8 Polarity of transfer clock

13.1.2 LSB First/MSB First Select Function

As shown in Figure 13.9, when the transfer format select bit (bit 7 at addresses 00A4$_{16}$ and 00AC$_{16}$) = "0", the transfer format is "LSB first"; when the bit = "1", the transfer format is "MSB first".

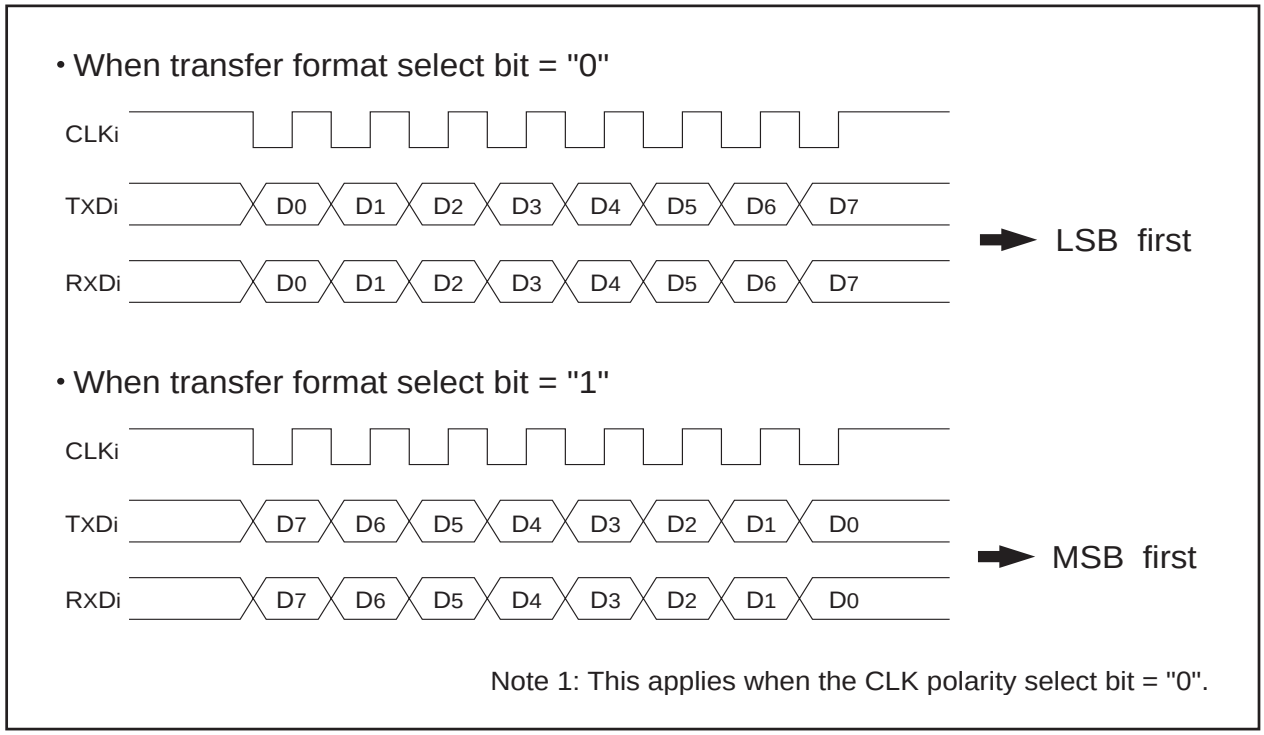
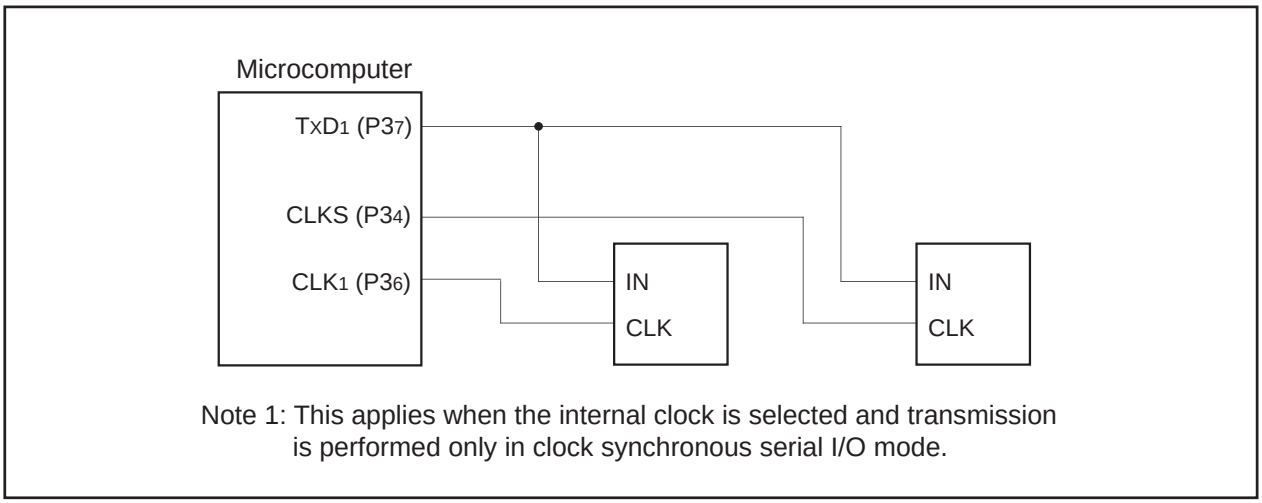


Figure 13.9 Transfer format

13.1.3 Transfer Clock Output from Multiple Pins Function (UART1)

This function allows the setting two transfer clock output pins and choosing one of the two to output a clock by using the CLK and CLKS select bit (bits 4 and 5 at address 00B0$_{16}$). The multiple pins function is valid only when the internal clock is selected for UART1.

Figure 13.10 shows the transfer clock output from the multiple pins function usage.



Note 1: This applies when the internal clock is selected and transmission is performed only in clock synchronous serial I/O mode.

Figure 13.10 The transfer clock output from the multiple pins function usage

13.1.4 Continuous Receive Mode

If the continuous receive mode enable bit (bits 2 and 3 at address 00B0$_{16}$) is set to "1", the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

13.1.5 RxD1 Input Pin Selection Function (UART1)

This function allows the setting two RxD$_1$ input pins and choosing one of the two to input serial data by using the RxD$_1$ input pin select bit (bits 6 at address 00B0$_{16}$).

When selecting "1" (P3$_5$) for RxD$_1$ input pin select bit, P3$_7$ functions as TxD$_1$ output pin. When selecting "0" (P3$_7$), serial data output cannot be performed. However, P3$_5$ can be used as an input/output port.

13.2 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 13.3 lists the specifications of UART mode. Figure 13.11 shows the UARTi transmit/receive mode register.

Table 13.3 Specifications of UART Mode

Item	Specification
Transfer data format	• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected • Start bit: 1 bit • Parity bit: Odd, even, or nothing as selected • Stop bit: 1 bit or 2 bits as selected
Transfer clock	• When internal clock is selected (bit 3 at addresses $00A0_{16}$, $00A8_{16}$ = "0"): $f_i/16(n+1)$ (Note 1) $f_i = f_1, f_8, f_{32}, f_C$ • When external clock is selected (bit 3 at addresses $00A0_{16}$ = "1"): $f_{EXT}/16(n+1)$ (Note 1) (Note 2)
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 at addresses $00A5_{16}$, $00AD_{16}$) = "1" - Transmit buffer empty flag (bit 1 at addresses $00A5_{16}$, $00AD_{16}$) = "0"
Reception start condition	• To start reception, the following requirements must be met: - Receive enable bit (bit 2 at addresses $00A5_{16}$, $00AD_{16}$) = "1" - Start bit detection
Interrupt request generation timing	• When transmitting - Transmit interrupt cause select bits (bits 0,1 at address $00B0_{16}$) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed - Transmit interrupt cause select bits (bits 0, 1 at address $00B0_{16}$) = "1": Interrupts requested when data transmission from UARTi transfer register is completed • When receiving - Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed
Error detection	• Overrun error (Note 3) This error occurs if the serial I/O started receiving the next data before reading the UARTi receive buffer register and the bit one before the last stop bit of the next data • Framing error This error occurs when the number of stop bits set is not detected • Parity error This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered
Select function	• RxD_1 input pin selection UART1 RxD_1 can be chosen by software to be input to one of the two pins set

Note 1: "n" denotes the value 00_{16} to FF_{16} that is set to the UART bit rate generator.

Note 2: f_{EXT} is input from the CLKi pin.

Note 3: If an overrun error occurs, the UARTi receive buffer will be indeterminate. Note also that the UARTi receive interrupt request bit does not change.

UARTi transmit/receive mode registers

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol: UiMR (i=0,1)
Address: 00A0$_{16}$, 00A8$_{16}$
When reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	RW
SMD1			RW
SMD2			RW
CKDIR	Internal / external clock select bit	0 : Internal clock 1 : External clock (Note 1)	RW
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	RW
PRY	Odd / even parity select bit	Valid when bit 6 = "1" 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
Reserved bit		Set to "0"	RW

Note 1: Set the corresponding port direction register to "0".

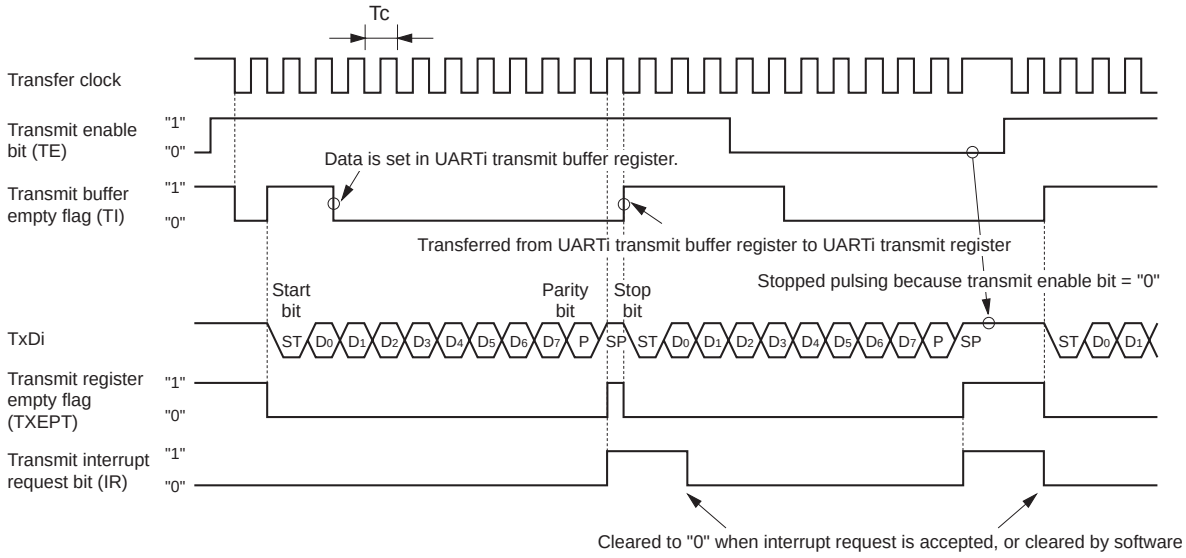
Figure 13.11 UARTi transmit/receive mode register in UART mode

Table 13.4 lists the functions of the input/output pins during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table 13.4 Input/output pin functions in UART mode

Function	Pin name	Method of selection	Remarks
Serial data output	TxD$_0$ (P1$_4$)	———	Port P1$_4$ cannot be used as an I/O port even when performing only serial data input but not serial data output.
	TxD$_1$ (P3$_7$)	RxD$_1$ input pin select bit (bit 6 at address 00B0$_{16}$)="1"	Port P3$_7$ cannot be used as an I/O port even when performing only serial data input but not serial data output.
Serial data input	RxD$_0$ (P1$_5$)	Port P1$_5$ direction register (bit 5 at address 00E3$_{16}$)="0"	Port P1$_5$ can be used as an I/O port when performing only serial data output but not serial data input.
	RxD$_1$ (P3$_5$)	Port P3$_5$ direction register (bit 5 at address 00E7$_{16}$)="0" RxD$_1$ input pin select bit (bit 6 at address 00B0$_{16}$)="1"	Port P3$_5$ can be used as an I/O port when performing only serial data output but not serial data input.
	RxD$_1$ (P3$_7$)	Port P3$_7$ direction register (bit 7 at address 00E7$_{16}$)="0" RxD$_1$ input pin select bit (bit 6 at address 00B0$_{16}$)="0"	When setting Port P3$_7$ as RxD$_1$, serial data output cannot be performed. Port P3$_5$ can be used as an I/O port.
Transfer clock input	CLKi (P1$_6$, P3$_6$)	Internal/external clock select bit (bit 3 at address 00A0$_{16}$ and 00A8$_{16}$)="1" Ports P1$_6$ and P3$_6$ direction register (bit 6 at address 00E3$_{16}$ and 00E7$_{16}$)="0"	Ports P1$_6$ and P3$_6$ can be used as an I/O port when not performing transfer clock input. In this case, set the internal/external clock select bit to "0".

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

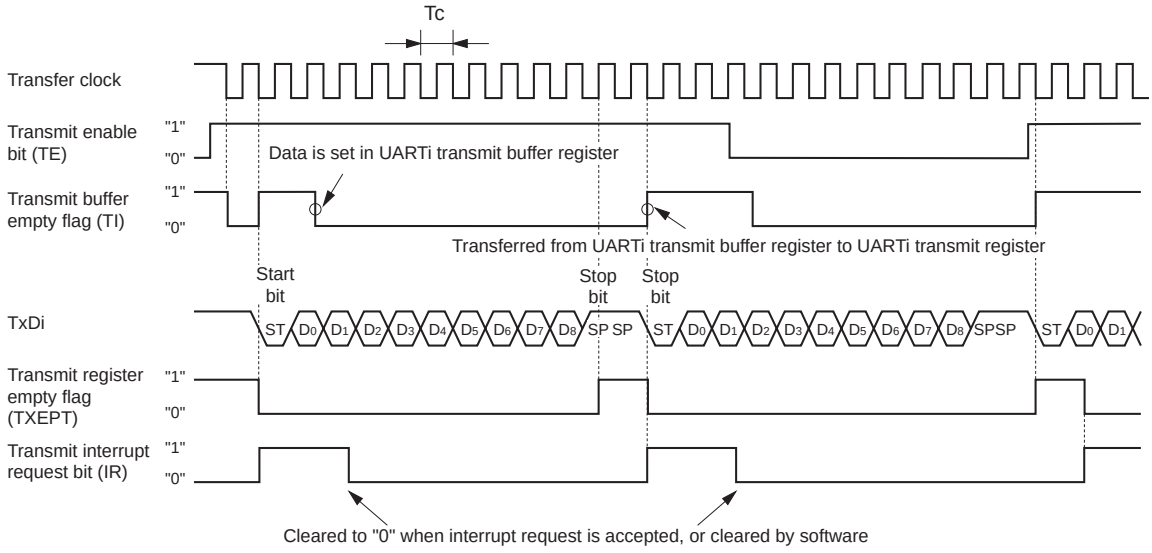


Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = 1.

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32, fc)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

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



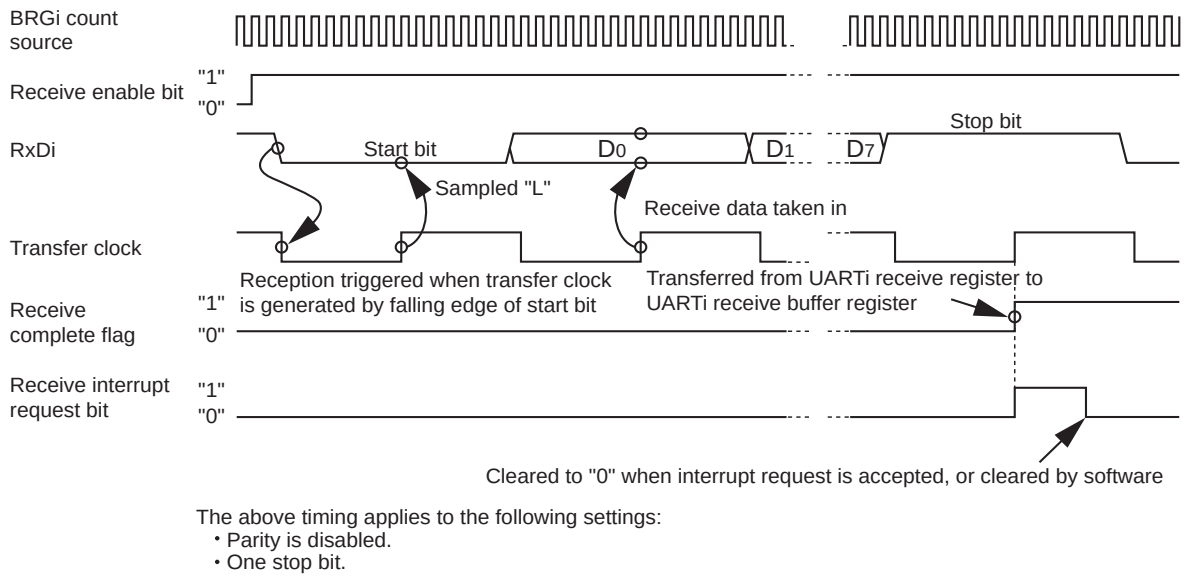
Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is disabled.
- Two stop bits.
- Transmit interrupt cause select bit = 0.

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32, fc)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 13.12 Typical transmit timings in UART mode

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



The above timing applies to the following settings:
- Parity is disabled.
- One stop bit.

Figure 13.13 Typical receive timing in UART mode

13.2.1 RxD_1 Input pin Selection Function (UART1)

This function allows the setting two RxD_1 input pins and choosing one of the two to input serial data by using the RxD_1 input pin select bit (bits 6 at address $00B0_{16}$).

When selecting "1" ($P3_5$) for RxD_1 input pin select bit, $P3_7$ functions as TxD_1 output pin. When selecting "0" ($P3_7$), serial data output cannot be performed. However, $P3_5$ can be used as an input/output port.

14. A/D Converter

The A/D converter consists of one 10-bit successive approximation A/D converter circuit with a capacitive coupling amplifier. Pins $P0_0$ to $P0_7$, $P1_0$ to $P1_3$, $P4_0$ and $P4_1$ also function as the analog signal input pins. The direction registers of these pins for A/D conversion must therefore be set to input. The Vref connect bit (bit 5 at address $00D7_{16}$) can be used to isolate the resistance ladder of the A/D converter from the reference voltage input pin (V_{REF}) when the A/D converter is not used. Doing so stops any current flowing into the resistance ladder from V_{REF}, reducing the power dissipation. When using the A/D converter, start A/D conversion only after connecting to V_{REF}.

The result of A/D conversion is stored in the A/D registers. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses.

Table 14.1 shows the performance of the A/D converter. Figure 14.1 shows the block diagram of the A/D converter, and Figures 14.2 and 14.3 show the A/D converter-related registers.

Table 14.1 Performance of A/D converter

Item	Performance
Method of A/D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to V_{CC}
Operating clock $Ø_{AD}$ (Note 2)	V_{CC} = 5V f_{AD}, divide-by-2 of f_{AD}, divide-by-4 of f_{AD}, $f_{AD}=f(X_{IN})$
Resolution	8-bit or 10-bit (selectable)
Absolute precision	V_{CC} = 5V • Without sample and hold function ±3LSB • With sample and hold function (8-bit resolution) ±2LSB • With sample and hold function (10-bit resolution) AN_0 to AN_{11} input: ±3LSB $ANEX_0$ and $ANEX_1$ input (including mode in which external operation amp is connected): ±7LSB
Operating modes	One-shot mode and repeat mode (Note 3)
Analog input pins	12 pins (AN_0 to AN_{11}) + 2 pins ($ANEX_0$ to $ANEX_1$)
A/D conversion start condition	• Software trigger A/D conversion starts when the A/D conversion start flag changes to "1"
Conversion speed per pin	• Without sample and hold function 8-bit resolution: 49 $Ø_{AD}$ cycles, 10-bit resolution: 59 $Ø_{AD}$ cycles • With sample and hold function 8-bit resolution: 28 $Ø_{AD}$ cycles, 10-bit resolution: 33 $Ø_{AD}$ cycles

Note 1: Does not depend on use of sample and hold function.

Note 2: Divide f_{AD} if (X_{IN}) exceeds 10MHz, and make $Ø_{AD}$ equal to or lower than 10MHz. Also if Vcc is less than 4.2V, divide f_{AD} and make $Ø_{AD}$ equal to or lower than $f_{AD}/2$.
Without sample and hold function, set the $Ø_{AD}$ frequency to 250kHz min.
With the sample and hold function, set the $Ø_{AD}$ frequency to 1MHz min.

Note 3: In repeat mode, only 8-bit mode can be used.

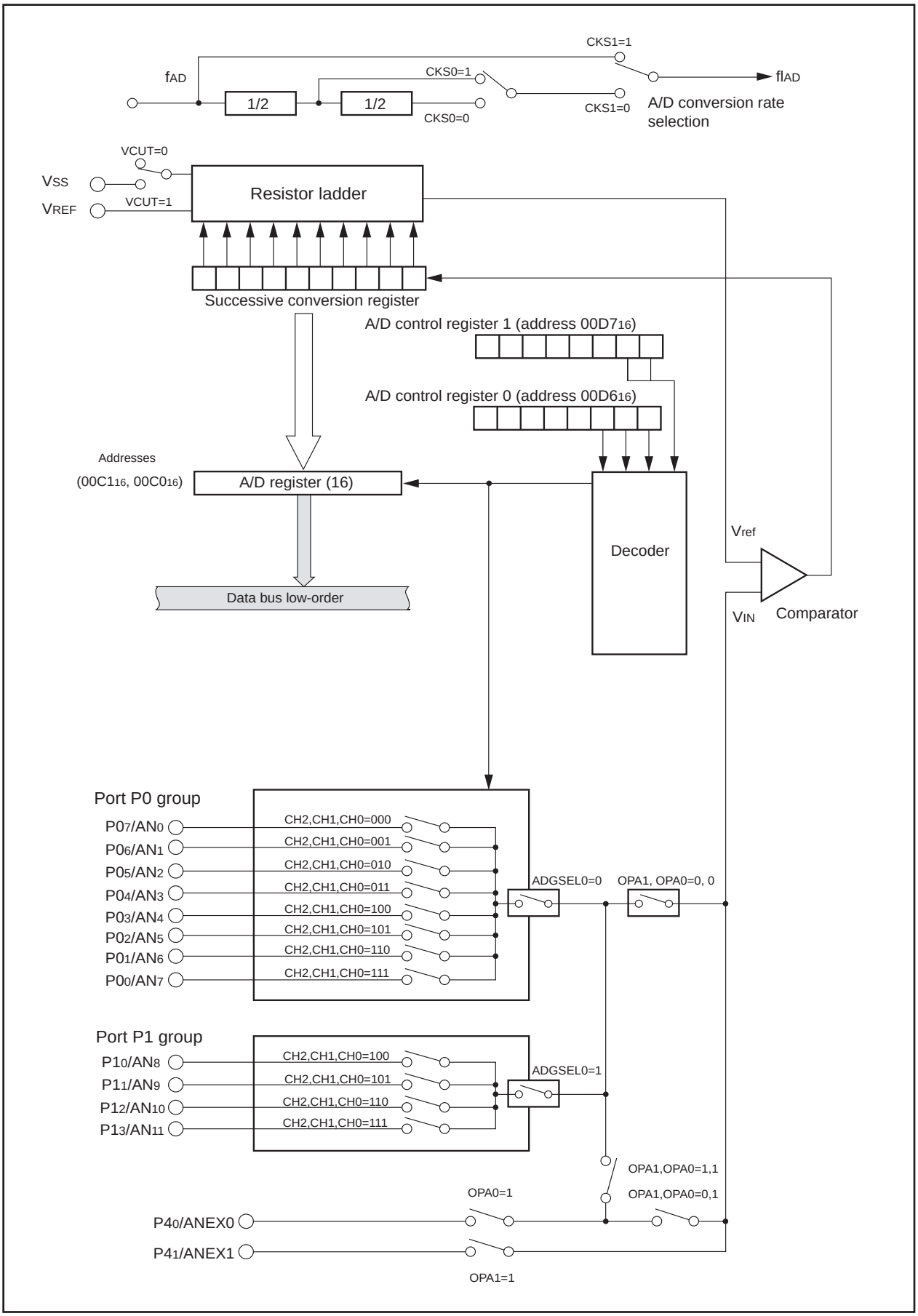


Figure 14.1 Block diagram of A/D converter

A/D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol	Address	When reset
ADCON0	$00D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4, AN_8 are selected 1 0 1 : AN_5, AN_9 are selected 1 1 0 : AN_6, AN_{10} are selected 1 1 1 : AN_7, AN_{11} are selected (Note 2, 3)	RW
CH1			RW
CH2			RW
MD	A/D operation mode select bit 0	0 : One-shot mode 1 : Repeat mode (Note 2)	RW
ADGSEL0	A/D input group select bit	0 : Port P0 group is selected 1 : Port P1 group is selected	RW
Reserved bit		Set to "0"	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	RW

Note 1: If the A/D control register is rewritten during A/D conversion, the conversion result is indeterminate.
Note 2: When changing A/D operation mode, set analog input pin again.
Note 3: AN_4 to AN_7 and AN_8 to AN_{11} are selected by the A/D input group select bit.

A/D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
					0	0	0

Symbol	Address	When reset
ADCON1	$00D7_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
Reserved bit		Set to "0"	RW
BITS	8/10-bit mode select bit (Note 2)	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1 (Note 3)	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	RW
VCUT	V_{REF} connect bit	0 : V_{REF} not connected 1 : V_{REF} connected	RW
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : External op-amp connection mode	RW
OPA1			RW

Note 1: If the A/D control register is rewritten during A/D conversion, the conversion result is indeterminate.
Note 2: In repeat mode, only 8-bit mode can be used.
Note 3: When $f(X_{IN})$ is over 10 MHz, the f_{AD} frequency must be under 10 MHz by dividing.

Figure 14.2 A/D converter-related registers (1)

A/D control register 2 (Note 1)

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

Symbol	Address	When reset
ADCON2	$00D4_{16}$	$XXXX0000_2$

Bit symbol	Bit name	Function	RW
SMP	A/D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	RW
Reserved bit		Set to "0"	RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".			—

Note 1: If the A/D control register is rewritten during A/D conversion, the conversion result is indeterminate.

A/D register

(b15) b7	(b8) b0	b7	b0
×	×	×	×

Symbol	Address	When reset
AD	$00C0_{16}$	Indeterminate
	$00C1_{16}$	Indeterminate

Function	RW
Eight low-order bits of A/D conversion result	RO
• During 10-bit mode Two high-order bits of A/D conversion result	RO
• During 8-bit mode The value, if read, turns out to be indeterminate.	—
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".	—

Figure 14.3 A/D converter-related registers (2)

14.1 One-shot Mode

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A/D conversion.
(See Table 14.2.) Figure 14.4 shows the A/D control register in one-shot mode.

Table 14.2 One-shot mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A/D conversion
Start condition	Writing "1" to A/D conversion start flag
Stop condition	• End of A/D conversion (A/D conversion start flag changes to "0") • Writing "0" to A/D conversion start flag
Interrupt request generation timing	End of A/D conversion
Input pin	One of AN0 to AN11, as selected
Reading of result of A/D converter	Read A/D register

A/D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol: ADCON0 — Address: $00D6_{16}$ — When reset: $00000XXX_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4, AN8 are selected 1 0 1 : AN5, AN9 are selected 1 1 0 : AN6, AN10 are selected 1 1 1 : AN7, AN11 are selected (Note 2, 3)	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	0 : One-shot mode (Note 2)	RW
ADGSEL0	A/D input group select bit	0 : Port P0 group is selected 1 : Port P1 group is selected	RW
Reserved bit		Set to "0"	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	0 : fAD/4 is selected 1 : fAD/2 is selected	RW

Note 1: If the A/D control register is rewritten during A/D conversion, the conversion result is indeterminate.
Note 2: When changing A/D operation mode, set analog input pin again.
Note 3: AN4 to AN7 and AN8 to AN11 are selected by the A/D input group select bit.

A/D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0	0	0

Symbol: ADCON1 — Address: $00D7_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function	RW
Reserved bit		Set to "0"	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1 (Note 2)	0 : fAD/2 or fAD/4 is selected 1 : fAD is selected	RW
VCUT	VREF connect bit	1 : VREF connected	RW
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : External op-amp connection mode	RW
OPA1			RW

Note 1: If the A/D control register is rewritten during A/D conversion, the conversion result is indeterminate.
Note 2: When f(XIN) is over 10 MHz, the fAD frequency must be under 10 MHz by dividing.

Figure 14.4 A/D conversion register in one-shot mode

14.2 Repeat Mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A/D conversion. (See Table 14.3.) Figure 14.5 shows the A/D control register in repeat mode.

Table 14.3 Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A/D conversion
Start condition	Writing "1" to A/D conversion start flag
Stop condition	Writing "0" to A/D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN_0 to AN_{11}, as selected (Note 1)
Reading of result of A/D converter	Read A/D register (at any time)

Note 1: AN_4 to AN_7 can be used in the same way as for AN_8 to AN_{11}.

A/D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0		1			

Symbol: ADCON0 — Address: $00D6_{16}$ — When reset: $00000XXX_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4, AN_8 are selected 1 0 1 : AN_5, AN_9 are selected 1 1 0 : AN_6, AN_{10} are selected 1 1 1 : AN_7, AN_{11} are selected (Note 2, 3)	RW
CH1			RW
CH2			RW
MD	A/D operation mode select bit 0	1 : Repeat mode (Note 2)	RW
ADGSEL0	A/D input group select bit	0 : Port P0 group is selected 1 : Port P1 group is selected	RW
Reserved bit		Set to "0"	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	RW

Note 1: If the A/D control register is rewritten during A/D conversion, the conversion result is indeterminate.
Note 2: When changing A/D operation mode, set analog input pin again.
Note 3: AN_4 to AN_7 and AN_8 to AN_{11} are selected by the A/D input group select bit.

A/D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1		0	0	0	0

Symbol: ADCON1 — Address: $00D7_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function	RW
Reserved bit		Set to "0"	RW
BITS	8/10-bit mode select bit (Note 2)	0 : 8-bit mode	RW
CKS1	Frequency select bit 1 (Note 3)	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	RW
VCUT	V_{REF} connect bit	1 : V_{REF} connected	RW
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : External op-amp connection mode	RW
OPA1			RW

Note 1: If the A/D control register is rewritten during A/D conversion, the conversion result is indeterminate.
Note 2: In repeat mode, only 8-bit mode can be used.
Note 3: When $f(X_{IN})$ is over 10 MHz, the f_{AD} frequency must be under 10 MHz by dividing.

Figure 14.5 A/D conversion register in repeat mode

14.3 Sample and Hold

Sample and hold is selected by setting bit 0 of the A/D control register 2 (address $00D4_{16}$) to "1". When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28 $\emptyset_{AD}$ cycle is achieved with 8-bit resolution and 33 $\emptyset_{AD}$ with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A/D conversion whether sample and hold is to be used.

14.4 Extended Analog Input Pins

In one-shot mode and repeat mode, the input via the extended analog input pins $ANEX_0$ and $ANEX_1$ can also be converted from analog to digital.

When bit 6 of the A/D control register 1 (address $00D7_{16}$) is "1" and bit 7 is "0", input via $ANEX_0$ is converted from analog to digital.

When bit 6 of the A/D control register 1 (address $00D7_{16}$) is "0" and bit 7 is "1", input via $ANEX_1$ is converted from analog to digital.

14.5 External Operation Amp Connection Mode

In this mode, multiple external analog inputs via the extended analog input pins, $ANEX_0$ and $ANEX_1$, can be amplified together by just one operation amp and used as the input for A/D conversion.

When bit 6 of the A/D control register 1 (address $00D7_{16}$) is "1" and bit 7 is "1", input via AN_0 to AN_{11} is output from $ANEX_0$. The input from $ANEX_1$ is converted from analog to digital and the result stored in the A/D register. The speed of A/D conversion depends on the response of the external operation amp. Do not connect the $ANEX_0$ and $ANEX_1$ pins directly.

Figure 14.6 is an example of how to connect the pins in external operation amp mode.

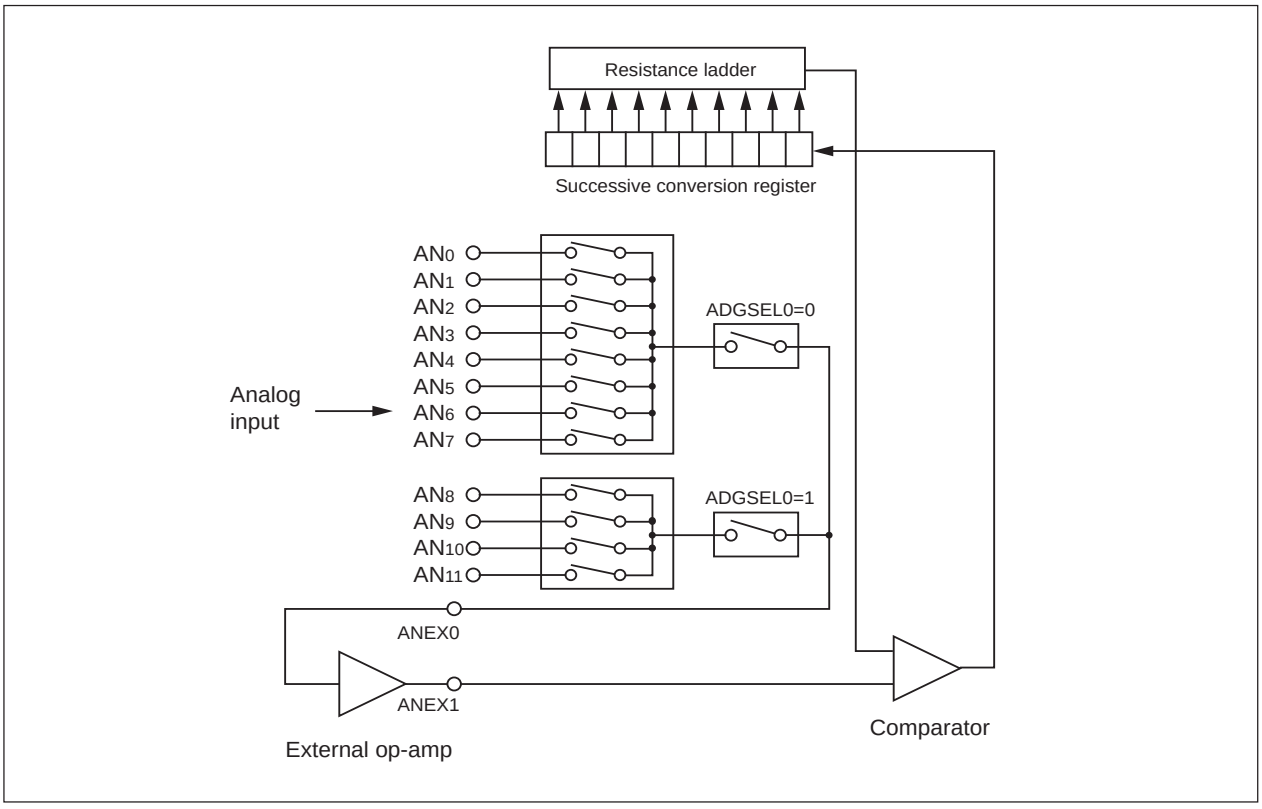


Figure 14.6 Example of external op-amp connection mode

15. D/A Converter

This is an 8-bit, R-2R type D/A converter. The microcomputer contains one independent D/A converter of this type.

D/A conversion is performed when a value is written to the corresponding D/A register. Bit 0 (D/A output enable bit) of the D/A control register decide if the result of conversion is to be output. Do not set the target port to output mode if D/A conversion is to be performed. When D/A output is set for enabled, the corresponding port is inhibited to be pulled up.

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

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

V_{REF}: reference voltage

Table 15.1 lists the performance of the D/A converter. Figure 15.1 shows the block diagram of the D/A converter, Figure 15.2 shows the D/A control register and Figure 15.3 shows D/A converter equivalent circuit.

Table 15.1 Performance of D/A converter

Item	Performance
Conversion method	R-2R method
Resolution	8 bits
Analog output pin	1 channel

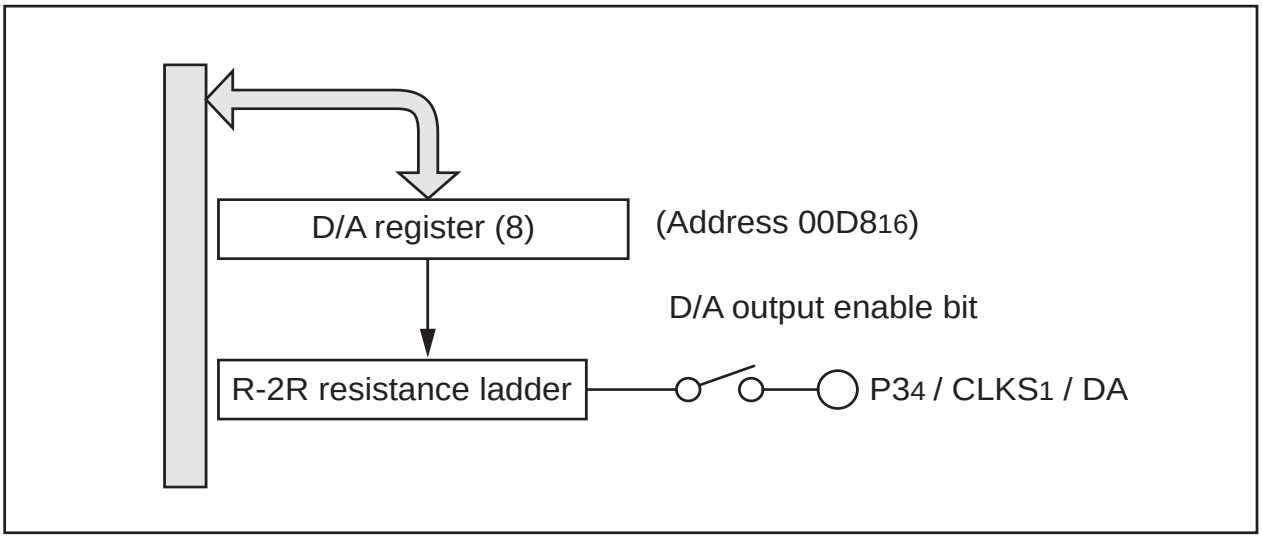


Figure 15.1 Block diagram of D/A converter

D/A control register

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

Symbol	Address	When reset
DACON	$00DC_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
DAE	D/A output enable bit	0 : Output disabled 1 : Output enabled	RW
Nothing is assigned. When write, set "0". When read, the value of this bit is "0".			—
Reserved bit		Set to "0"	RW
Nothing is assigned. When write, set "0". When read, the value of these bits are "0".			—

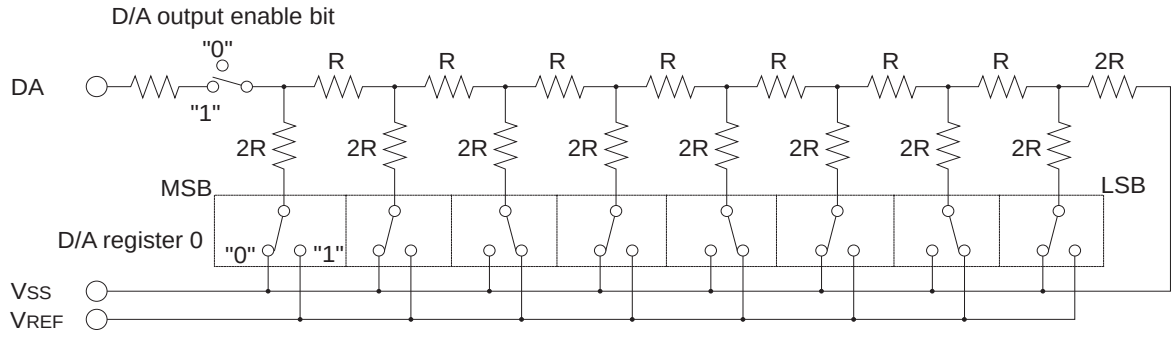
D/A register

b7	b0

Symbol	Address	When reset
DA	$00D8_{16}$	Indeterminate

Function	RW
Output value of D/A conversion	RW

Figure 15.2 D/A control register



Note 1: In the above figure, the D/A register value is "$2A_{16}$".
Note 2: To save power when not using the D/A converter, set the D/A output enable bit to "0" and the D/A register to "00_{16}", and prevent current flowing to the R-2R resistance.

Figure 15.3 D/A converter equivalent circuit

16. CAN Module

The CAN (Controller Area Network) module for the M16C/1N group of microcomputers is a communication controller implementing the CAN 2.0B protocol. The M16C/1N group contains one Full CAN module which can transmit and receive messages in both standard (11-bit) ID and extended (29-bit) ID formats.

Figure 16.1 shows a block diagram of the CAN module.

External CAN bus driver and receiver are required.

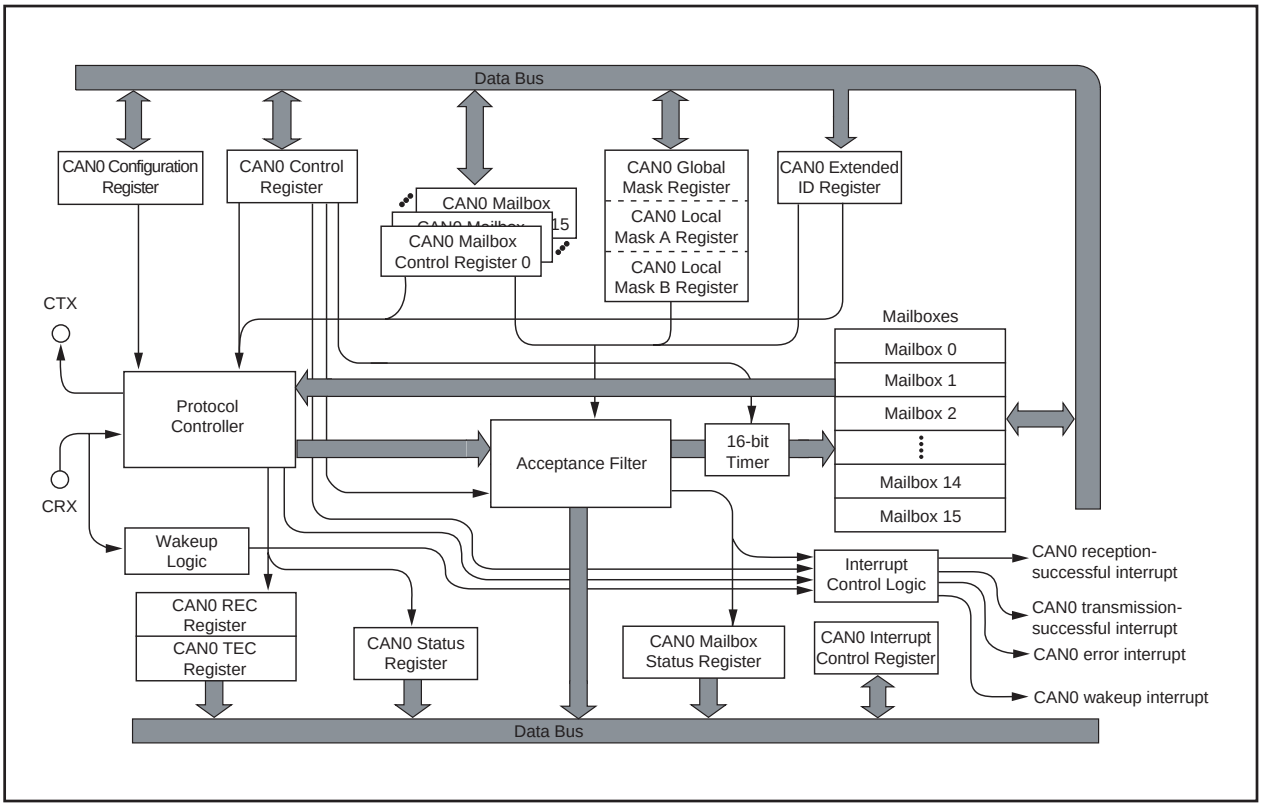


Figure 16.1 Block Diagram of CAN Module

CTx/CRx:	CAN I/O pins. Either P0$_2$, P0$_3$ or P5$_0$, P5$_1$ can be selected as CAN I/O pins by a program.
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 C0GMR register, the C0LMAR register, or the C0LMBR register is used.
16 bit timer:	Used for the time stamp function. When the received message is stored in the message memory, the timer value is stored as a time stamp.
Wake up function:	CAN0 wake up interrupt is generated by a message from the CAN bus.
Interrupt generation function:	The interrupt events are provided by the CAN module. CAN0 successful reception interrupt, CAN0 successful transmission interrupt, CAN0 error interrupt, and CAN0 wake up interrupt.

16.1 CAN Module-Related Registers

The CAN0 module has the following registers.

(1) CAN Message Box

A CAN module is equipped with 16 slots (16 bytes or 8 words each). Slots 14 and 15 can be used as Basic CAN.

- Priority of the slots: The smaller the number of the slot, the higher the priority, in both transmission and reception.
- A program can define whether a slot is defined as transmitter or receiver.

(2) Acceptance Mask Registers

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

- CAN0 global mask register (C0GMR register: 6 bytes)
  Configuration of the masking condition for acceptance filtering processing to slots 0 to 13
- CAN0 local mask A register (C0LMAR register: 6 bytes)
  Configuration of the masking condition for acceptance filtering processing to slot 14
- CAN0 local mask B register (C0LMBR register: 6 bytes)
  Configuration of the masking condition for acceptance filtering processing to slot 15

(3) CAN SFR Registers

- CAN0 message control register i (C0MCTLi register: 8 bits X 16) (i = 0 to 15)
  Control of transmission and reception of a corresponding slot
- CAN0 control register (C0CTLR register: 16 bits)
  Control of the CAN protocol
- CAN0 status register (C0STR register: 16 bits)
  Indication of the protocol status
- CAN0 slot status register (C0SSTR register: 16 bits)
  Indication of the status of contents of each slot
- CAN0 interrupt control register (C0ICR register: 16 bits)
  Selection of interrupt enabled or disabled for each slot
- CAN0 extended ID register (C0IDR register: 16 bits)
  Selection of ID format (standard or extended) for each slot
- CAN0 configuration register (C0CONR register: 16 bits)
  Configuration of the bus timing
- CAN0 receive error count register (C0RECR register: 8 bits)
  Indication of the error status of the CAN module in reception: the counter value is incremented or decremented according to the error occurrence.
- CAN0 transmit error count register (C0TECR register: 8 bits)
  Indication of the error status of the CAN module in transmission: the counter value is incremented or decremented according to the error occurrence.
- CAN0 acceptance filter support register (C0AFS register: 16 bits)
  Decoding the received ID for use by the acceptance filter support unit

Explanation of each register is given below.

16.2 CAN0 Message Box

Table 16.1 shows the memory mapping of the CAN0 message box.

It is possible to access to the message box in byte or word.

Mapping of the message contents differs from byte access to word access. Byte access or word access can be selected by the MsgOrder bit of the C0CTLR register.

Table 16.1 Memory Mapping of CAN0 Message Box (n = 0 to 15: the number of the slot)

Address	Message content (Memory mapping)	
	Byte access (8 bits)	Word access (16 bits)
0260_{16} + n • 16 + 0	SID_{10} to SID_6	SID_5 to SID_0
0260_{16} + n • 16 + 1	SID_5 to SID_0	SID_{10} to SID_6
0260_{16} + n • 16 + 2	EID_{17} to EID_{14}	EID_{13} to EID_6
0260_{16} + n • 16 + 3	EID_{13} to EID_6	EID_{17} to EID_{14}
0260_{16} + n • 16 + 4	EID_5 to EID_0	Data Length Code (DLC)
0260_{16} + n • 16 + 5	Data Length Code (DLC)	EID_5 to EID_0
0260_{16} + n • 16 + 6	Data byte 0	Data byte 1
0260_{16} + n • 16 + 7 ⋮ 0260_{16} + n • 16 + 13	Data byte 1 ⋮ Data byte 7	Data byte 0 ⋮ Data byte 6
0260_{16} + n • 16 + 14	Time stamp high-order byte	Time stamp low-order byte
0260_{16} + n • 16 + 15	Time stamp low-order byte	Time stamp high-order byte

Figures 16.2 and 16.3 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.

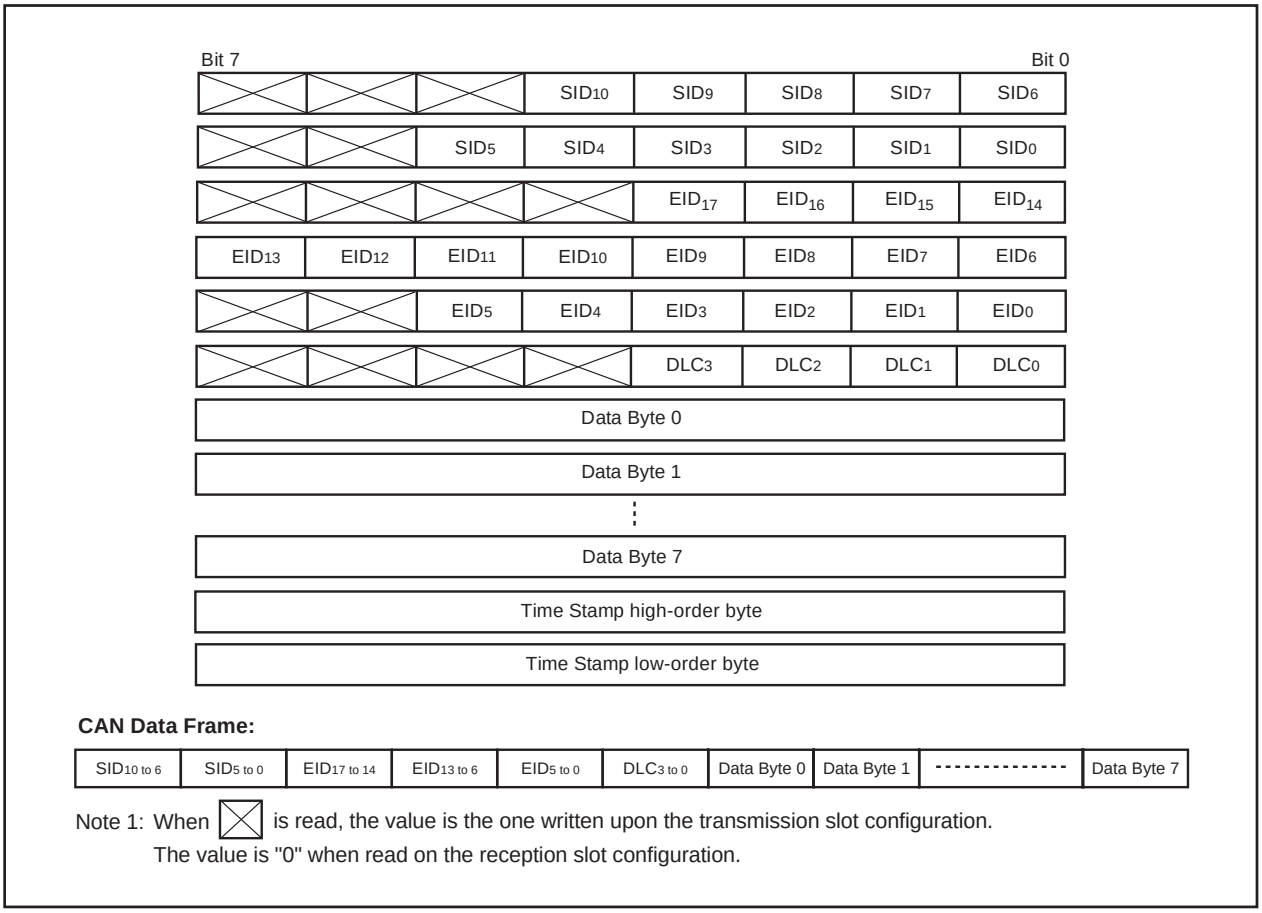


Figure 16.2 Bit Mapping in Byte Access

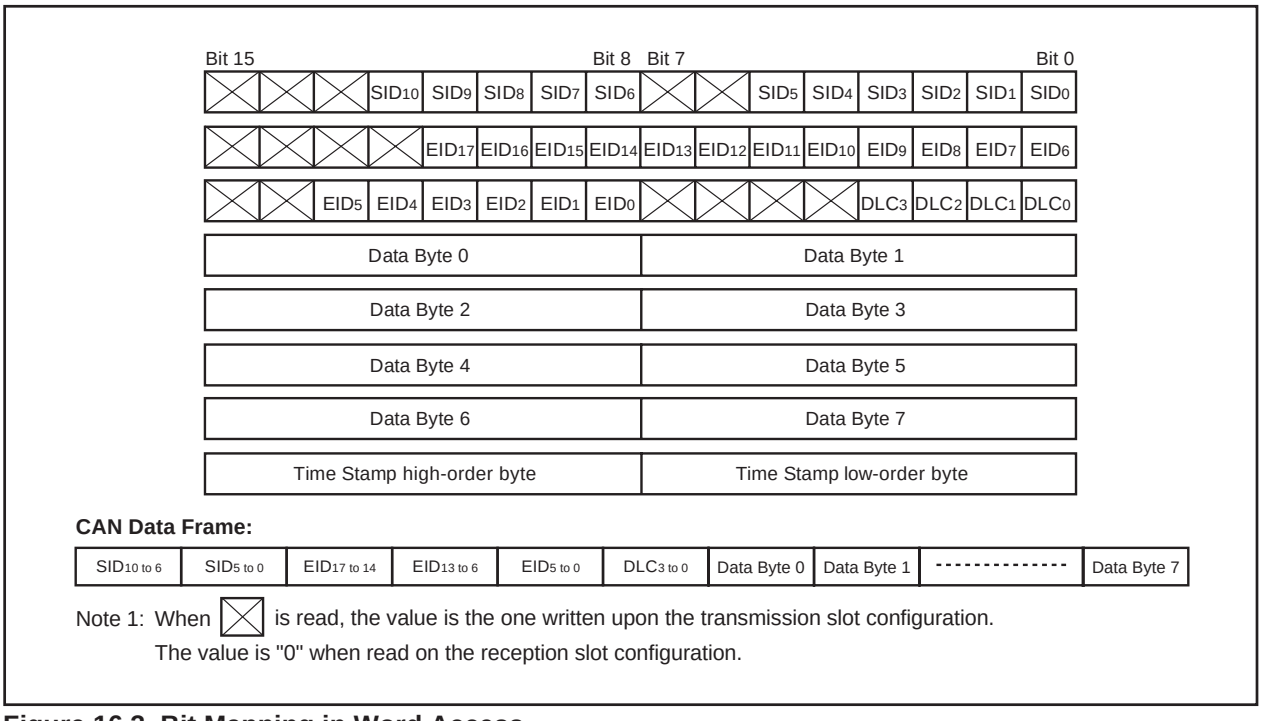
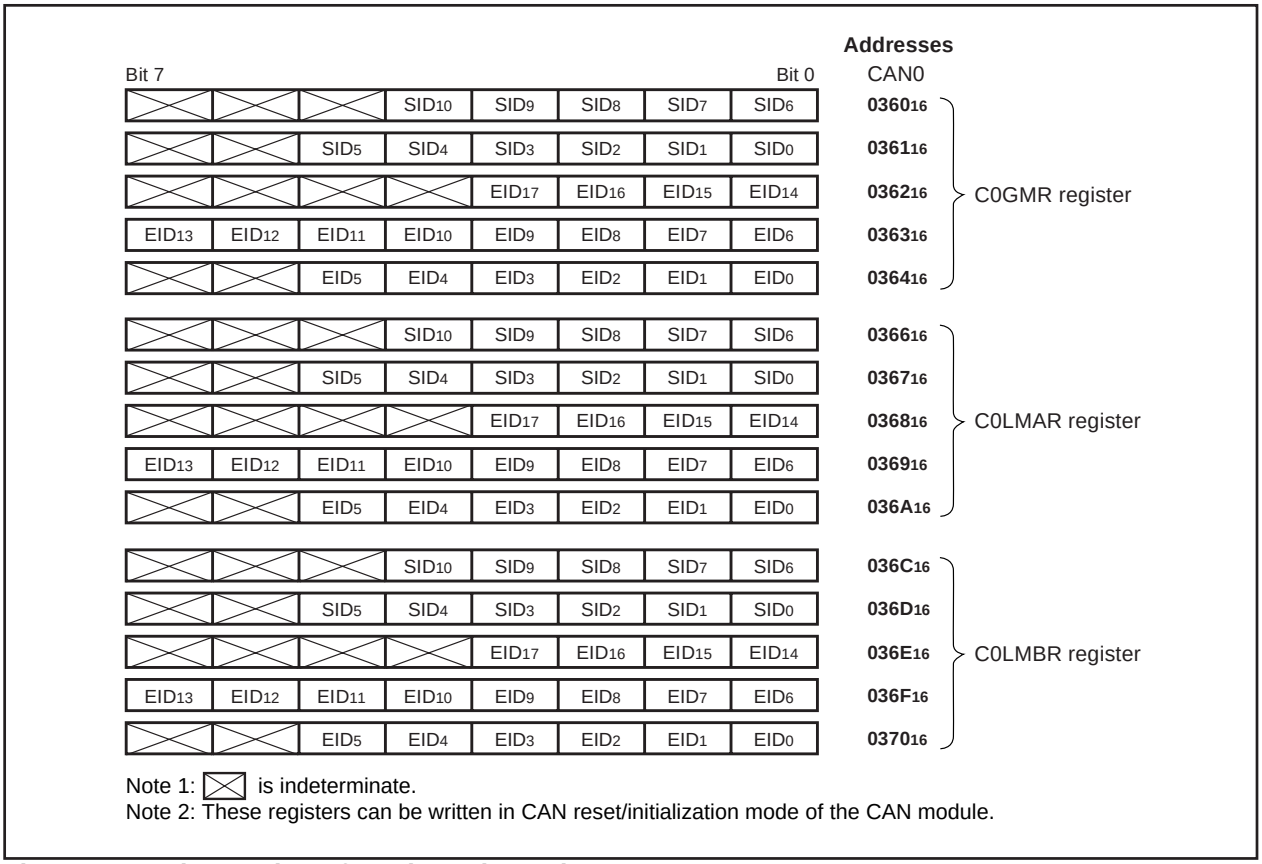


Figure 16.3 Bit Mapping in Word Access

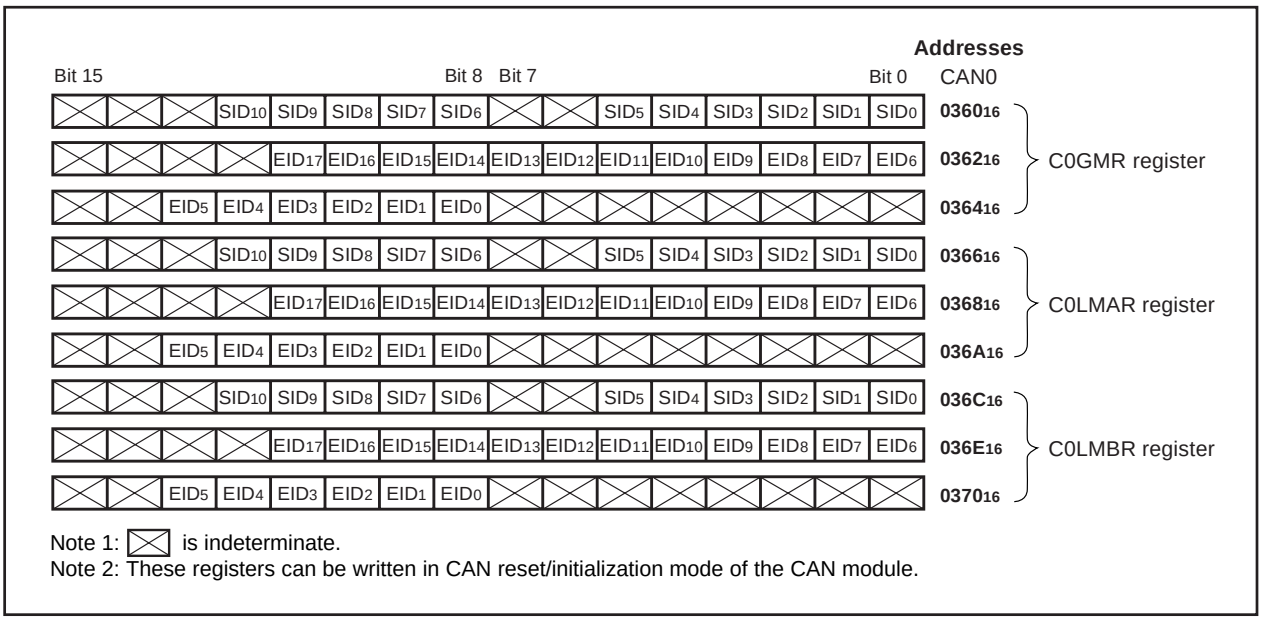
16.3 Acceptance Mask Registers

Figures 16.4 and 16.5 show the C0GMR register, the C0LMAR register, and the C0LMBR register, in which bit mapping in byte access and word access are shown.



Note 1: ⊠ is indeterminate.
Note 2: These registers can be written in CAN reset/initialization mode of the CAN module.

Figure 16.4 Bit Mapping of Mask Registers in Byte Access



Note 1: ⊠ is indeterminate.
Note 2: These registers can be written in CAN reset/initialization mode of the CAN module.

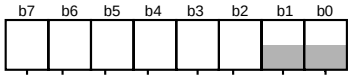
Figure 16.5 Bit Mapping of Mask Registers in Word Access

16.4 CAN SFR Registers

16.4.1 C0MCTLi Register (i = 0 to 15)

Figure 16.6 shows the C0MCTLi register.

CAN0 message control register i (i = 0 to 15) (Note 4)



Symbol	Address	After reset
C0MCTL0 to C0MCTL15	022016 to 022F16	0016

Bit symbol	Bit name	Function	RW
NewData	Successful reception flag	When set to reception slot 0 : The content of the slot is read or still under processing by the CPU. 1 : The CAN module has stored new data in the slot.	RO (Note 1)
SentData	Successful transmission flag	When set to transmission slot 0 : Transmission is not started or completed yet. 1 : Transmission is successfully completed.	RO (Note 1)
InvalData	"Under reception" flag	When set to reception slot 0 : The message is valid. 1 : The message is invalid. (The message is being updated.)	RO
TrmActive	"Under transmission" flag	When set to transmission slot 0 : Waiting for bus idle or completion of arbitration. 1 : Transmitting	RO
MsgLost	Overwrite flag	When set to reception slot 0 : No message has been overwritten in this slot. 1 : This slot already contained a message, but it has been overwritten by a new one.	RO (Note 1)
RemActive	Remote frame transmission/ reception status flag (Note 2)	0 : Data frame transmission/reception status 1 : Remote frame automatic transfer status	RW
RspLock	Transmission/ reception auto response lock mode select bit	When set to reception remote frame slot 0 : After a remote frame is received, it will be answered automatically. 1 : After a remote frame is received, no transmission will be started as long as this bit is set to "1". (Not responding)	RW
Remote	Remote frame corresponding slot select bit	0 : Slot not corresponding to remote frame 1 : Slot corresponding to remote frame	RW
RecReq	Reception slot request bit (Note 3)	0 : Not reception slot 1 : Reception slot	RW
TrmReq	Transmission slot request bit (Note 3)	0 : Not transmission slot 1 : Transmission slot	RW

Note 1: As for write, only writing "0" is possible. The value of each bit is written when the CAN module enters the respective state.
Note 2: In Basic CAN mode, this bit serves as data format identification flag. When receiving a data frame, this bit is set to "0" and when receiving a remote frame, this bit is set to "1".
Note 3: One slot cannot be defined as reception slot and transmission slot at the same time.
Note 4: This register can not be set in CAN reset/initialization mode of the CAN module.

Figure 16.6 C0MCTLi Register

16.4.2 C0CTLR Register

Figures 16.7 shows the C0CTLR register.

CAN0 control register

b7	b6	b5	b4	b3	b2	b1	b0
X							

Symbol	Address	After reset
C0CTLR	023016	X0000001₂

Bit symbol	Bit name	Function	RW
Reset	CAN module reset bit (Note 2)	0 : Operation mode 1 : Reset/initialization mode	RW
LoopBack	Loop back mode select bit (Note 3)	0 : Normal operation mode 1 : Loop back mode	RW
MsgOrder	Message order select bit (Note 3)	0 : Word access 1 : Byte access	RW
BasicCAN	Basic CAN mode select bit (Note 3)	0 : Normal operation mode 1 : Basic CAN mode	RW
BusErrEn	Bus error interrupt enable bit (Note 3)	0 : Bus error interrupt disabled 1 : Bus error interrupt enabled	RW
Sleep	Sleep mode select bit (Note 3, 4)	0 : Sleep mode disabled 1 : Sleep mode enabled; clock supply stopped	RW
PortEn	CAN port enable bit	0 : I/O port function 1 : CTx/CRx function (Note 1)	RW
– (b7)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–

Note 1: Irrespective of setting of PD0 and PD5 registers.
Note 2: When the Reset bit is set to "1" (CAN reset/initialization mode), check that the State_Reset bit of the C0STR register is set to "1" (Reset mode).
Note 3: Set these bits only in CAN reset/initialization mode.
Note 4: When using CAN0 wake up interrupt, set this bit to "1" (Sleep mode disabled).

(b15) b7	b6	b5	b4	b3	b2	b1	(b8) b0
X	X		X				

Symbol	Address	After reset
C0CTLR	023116	XX0X0000₂

Bit symbol	Bit name	Function	RW
Bit1, Bit0	Time stamp prescaler (Note 3)	b1 b0 0 0 : Period of 1 bit time 0 1 : Period of 1/2 bit time 1 0 : Period of 1/4 bit time 1 1 : Period of 1/8 bit time	RW
TSReset	Time stamp counter reset bit (Note 1)	0 : Normal operation mode 1 : Force reset of the time stamp counter	RW
RetBusOff	Return from bus off command bit (Note 2)	0 : Normal operation mode 1 : Force return from bus off	RW
– (b4)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–
RXOnly	Listen-only mode select bit (Note 3)	0 : Normal operation mode 1 : Listen-only mode (Note 4)	RW
– (b7-b6)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–

Note 1: When the TSReset bit = 1, the C0TSR register is set to "0000₁₆". After this, the bit is automatically set to "0".
Note 2: When the RetBusOff bit = 1, the C0RECR register and the C0TECR register are set to "00₁₆". After this, the bit is automatically set to "0".
Note 3: Set these bits only in CAN reset/initialization mode.
Note 4: When listen-only mode is selected, do not request a transmission.

Figure 16.7 C0CTLR Register

16.4.3 C0STR Register

Figure 16.8 shows the C0STR register.

CAN0 status register (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
C0STR	023216	0016

Bit symbol	Bit name	Function	RW
MBox	Active slot bits (Note 2)	b3 b2 b1 b0 0 0 0 0 : Slot 0 0 0 0 1 : Slot 1 0 0 1 0 : Slot 2 ⋮ 1 1 1 0 : Slot 14 1 1 1 1 : Slot 15	RO
TrmSucc	Successful transmission flag	0 : No [successful] transmission 1 : The CAN module has transmitted a message successfully.	RO
RecSucc	Successful reception flag	0 : No [successful] reception 1 : CAN module received a message successfully.	RO
TrmState	Transmission flag (Transmitter)	0 : CAN module is idle or receiver. 1 : CAN module is transmitter.	RO
RecState	Reception flag (Receiver)	0 : CAN module is idle or transmitter. 1 : CAN module is receiver.	RO

Note 1: These bits can not be set in CAN reset/initialization mode of the CAN module.
Note 2: These bits change when a slot enabled interrupt has transmitted or received successfully.

(b15)							(b8)
b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
C0STR	023316	X00000012

Bit symbol	Bit name	Function	RW
State_Reset	Reset state flag	0 : Operation mode 1 : Reset mode	RO
State_ LoopBack	Loop back state flag	0 : Normal operation mode 1 : Loop back mode	RO
State_ MsgOrder	Message order state flag	0 : Word access 1 : Byte access	RO
State_ BasicCAN	Basic CAN mode state flag	0 : Normal operation mode 1 : Basic CAN mode	RO
State_ BusError	Bus error state flag	0 : No error has occurred. 1 : A CAN bus error has occurred.	RO
State_ ErrPass	Error passive state flag	0 : The CAN module is not in error passive state. 1 : The CAN module is in error passive state.	RO
State_ BusOff	Error bus off state flag	0 : The CAN module is not in error bus off state. 1 : The CAN module is in error bus off state.	RO
– (b7)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–

Figure 16.8 C0STR Register

16.4.4 C0SSTR Register

Figure 16.9 shows the C0SSTR register.

CAN0 slot status register

(b15) (b8) b7 b0 b7 b0

Symbol	Address	After reset
C0SSTR	0235_{16}, 0234_{16}	0000_{16}

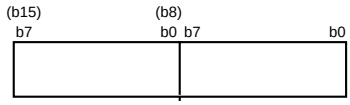
Function	Setting values	RW
Slot status bits Each bit corresponds to the slot with the same number.	0 : Reception slot The message has been read. Transmission slot Transmission is not completed. 1 : Reception slot The message has not been read. Transmission slot Transmission is completed.	RO

Figure 16.9 C0SSTR Register

16.4.5 C0ICR Register

Figure 16.10 shows the C0ICR register.

CAN0 interrupt control register (Note 1)



Symbol	Address	After reset
C0ICR	0237_{16}, 0236_{16}	0000_{16}

Function	Setting values	RW
Interrupt enable bits: Each bit corresponds with a slot with the same number. Enabled/disabled of successful transmission interrupt or successful reception interrupt can be selected.	0 : Interrupt disabled 1 : Interrupt enabled	RW

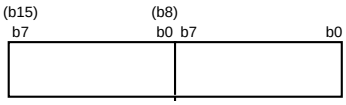
Note 1: These bits can not be set in CAN reset/initialization mode of the CAN module.

Figure 16.10 C0ICR Register

16.4.6 C0IDR Register

Figure 16.11 shows the C0IDR register.

CAN0 extended ID register (Note 1)



Symbol	Address	After reset
C0IDR	0239_{16}, 0238_{16}	0000_{16}

Function	Setting values	RW
Extended ID bits: Each bit corresponds with a slot with the same number. Selection of the ID format that each slot handles.	0 : Standard ID 1 : Extended ID	RW

Note 1: These bits can not be set in CAN reset/initialization mode of the CAN module.

Figure 16.11 C0IDR Register

16.4.7 C0CONR Register

Figure 16.12 shows the C0CONR register.

CAN0 configuration register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
C0CONR	023A16	Indeterminate

Bit symbol	Bit name	Function	RW
BRP	Prescaler division ratio select bits	b3 b2 b1 b0 0 0 0 0 : Divide-by-1 of fCAN 0 0 0 1 : Divide-by-2 of fCAN 0 0 1 0 : Divide-by-3 of fCAN ⋮ (Note 1) 1 1 1 0 : Divide-by-15 of fCAN 1 1 1 1 : Divide-by-16 of fCAN	RW
SAM	Sampling control bit	0 : One time sampling 1 : Three times sampling	RW
PTS	Propagation time segment control bits	b7 b6 b5 0 0 0 : 1Tq 0 0 1 : 2Tq 0 1 0 : 2Tq ⋮ 1 1 0 : 7Tq 1 1 1 : 8Tq	RW

Note 1: fCAN serves for the CAN clock. The period is decided by configuration of the CCLKi bits (i = 4 to 6).

(b15) b7	b6	b5	b4	b3	b2	b1	(b8) b0

Symbol	Address	After reset
C0CONR	023B16	Indeterminate

Bit symbol	Bit name	Function	RW
PBS1	Phase buffer segment 1 control bits	b2 b1b0 0 0 0 : Inhibited 0 0 1 : 2Tq 0 1 0 : 3Tq ⋮ 1 1 0 : 7Tq 1 1 1 : 8Tq	RW
PBS2	Phase buffer segment 2 control bits	b5 b4 b3 0 0 0 : Inhibited 0 0 1 : 2Tq 0 1 0 : 3Tq ⋮ 1 1 0 : 7Tq 1 1 1 : 8Tq	RW
SJW	Re synchronization jump width control bits	b7 b6 0 0 : 1Tq 0 1 : 2Tq 1 0 : 3Tq 1 1 : 4Tq	RW

Figure 16.12 C0CONR Register

16.4.8 C0RECR Register

Figure 16.13 shows the C0RECR register.

CAN0 receive error count register (Note 2)

b7 b0

Symbol	Address	After reset
C0RECR	$023C_{16}$	00_{16}

Function	Counter value	RW
Reception error counting function The value is incremented or decremented according to the CAN module's error status.	00_{16} to FF_{16} (Note 1)	RO

Note 1: The value is indeterminate in bus off state.
Note 2: These bits can not be set in CAN reset/initialization mode of the CAN module.

Figure 16.13 C0RECR Register

16.4.9 C0TECR Register

Figure 16.14 shows the C0TECR register.

CAN0 transmit error count register (Note 1)

b7 b0

Symbol	Address	After reset
C0TECR	$023D_{16}$	00_{16}

Function	Counter value	RW
Transmission error counting function The value is incremented or decremented according to the CAN module's error status.	00_{16} to FF_{16}	RO

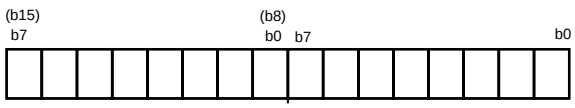
Note 1: These bits can not be set in CAN reset/initialization mode of the CAN module.

Figure 16.14 C0TECR Register

16.4.10 C0AFS Register

Figure 16.15 shows the C0AFS register.

CAN0 acceptance filter support register



Symbol	Address	After reset
C0AFS	0245_{16}, 0244_{16}	Indeterminate

Function	Setting values	RW
Write the content equivalent to the standard frame ID of the received message. The value is "converted standard frame ID" when read.	Standard frame ID	RW

Figure 16.15 C0AFS Register

16.5 Operational Modes

The CAN module has the following three operational modes.

- CAN Reset/Initialization Mode
- CAN Sleep Mode
- CAN Operation Mode

Figure 16.16 shows transition between operational modes.

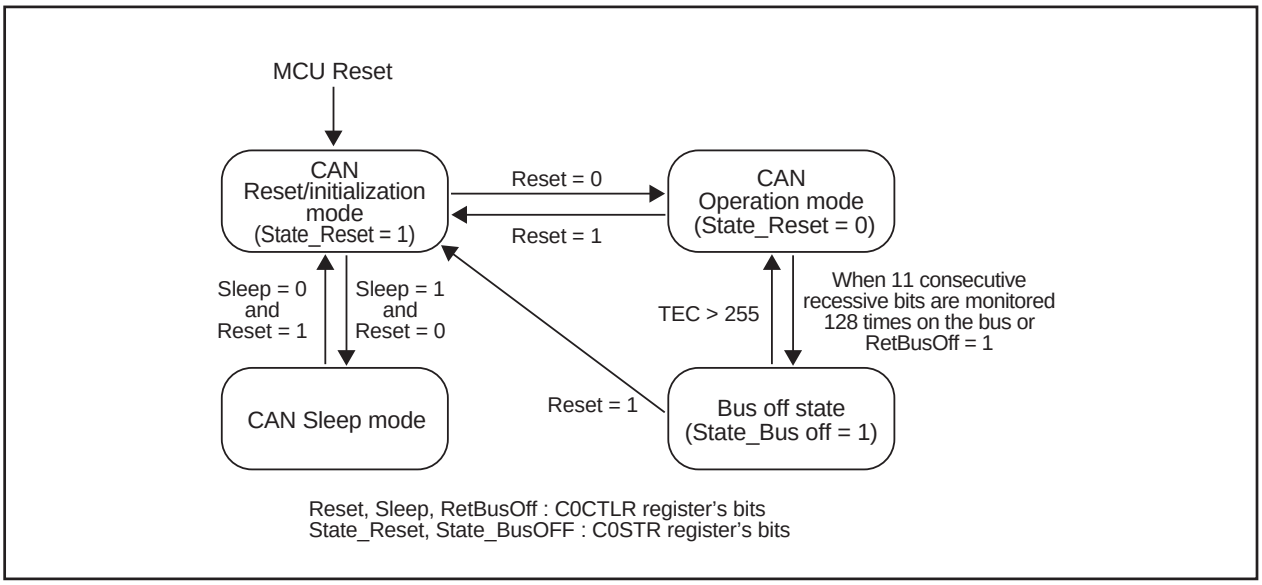


Figure 16.16 Transition Between Operational Modes

16.5.1 CAN Reset/Initialization Mode

CAN reset/initialization mode is activated upon MCU reset or by setting the Reset bit of the C0CTLR register. When setting the Reset bit to "1", check that the State_Reset bit of C0STR register is set to "1". Entering CAN reset/initialization mode, the module initiates the following functions:

- Suspend all communication functions. 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) and then sets the State_Reset bit.
- The C0IDR, C0MCTLi (i = 0 to 15), C0ICR, C0STR, C0RECR and C0TECR registers are initialized. All these registers are locked to prevent CPU modification.
- The C0CTLR, C0CONR, C0GMR, C0LMAR and C0LMBR registers and the CAN0 message box retain their contents and are available for CPU access.

16.5.2 CAN Operation Mode

CAN operation mode is activated by setting the Reset bit of the C0CTLR register to “0”. When setting the Reset bit to "0", check that the State_reset bit of C0STR register is set to "0". In CAN operation mode, the CAN module becomes the following status after having detected 11 consecutive recessive bits on the bus.

- The module's communication functions are released and it becomes an active node on the network and may transmit and receive CAN messages.
- Release the internal fault confinement logic including receive and transmit error counters. The module may leave CAN operation mode depending on the error counts.

Within CAN operation mode, the module may be in three different sub modes, depending on which type of communication functions are performed:

- Module idle: The modules receive and transmit sections are inactive.
- Module receives: The module receives a CAN message sent by another node.
- Module transmits: The module transmits a CAN message. The module may receive its own message simultaneously when the looback function is enabled.

Figure 16.17 shows sub modes of CAN operation mode.

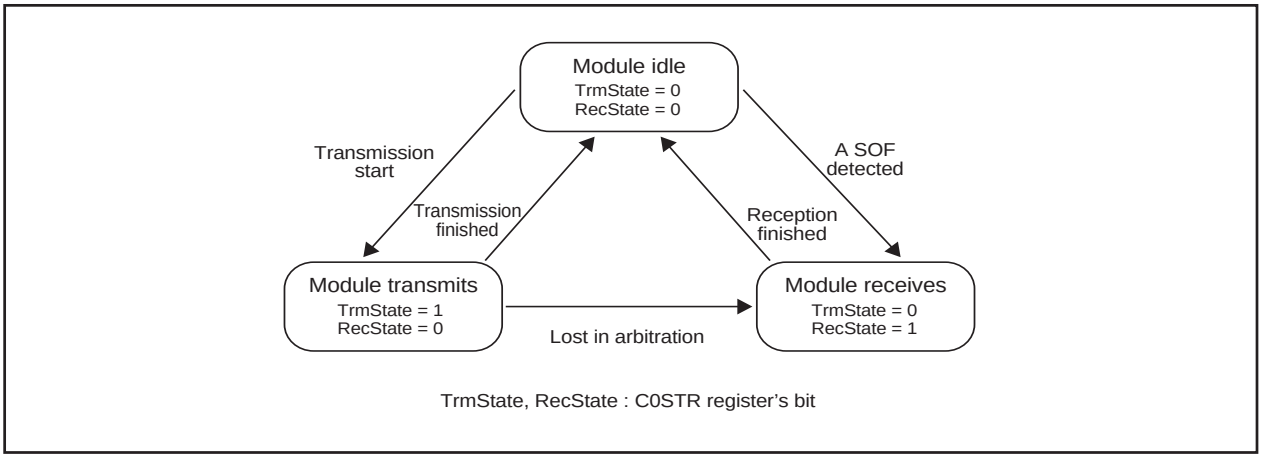


Figure 16.17 Sub Modes of CAN Operation Mode

16.5.3 CAN Sleep Mode

CAN sleep mode is activated by setting the Sleep bit of the C0CTLR register to “1” and Reset bit to “0”. It should never be activated from CAN operation mode but only via CAN reset/initialization mode. Entering CAN sleep mode instantly stops the modules clock supply and thereby reduces power dissipation.

16.5.4 Bus Off State

The bus off sate is entered according to the fault confinement rules of the CAN specification. When returning to CAN operation mode from the bus off state, the module has the following two cases. In this time, the value of any CAN registers, except C0STR, C0RECR and C0TECR registers, does not change.

(1) When 11 consecutive recessive bits are monitored 128 times

The module enters instantly into error active state and the CAN communication becomes possible immediately.

(2) When the RetBus Off bit in the CiCTLR register = 1 (Force return form buss off)

The module enters instantly into error active state, and the CAN communication becomes possible again after 11 consecutive recessive bits are monitored.

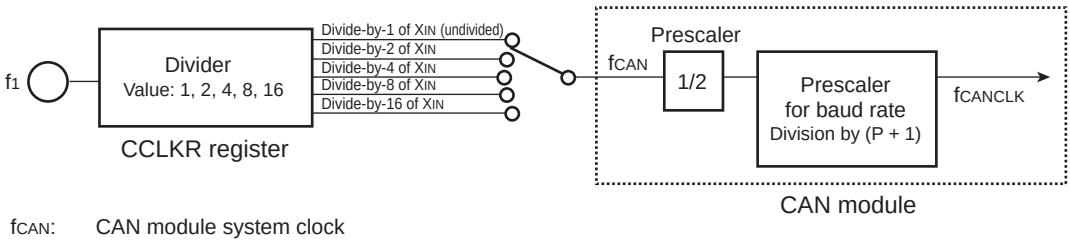
16.6 Configuration of the CAN Module System Clock

The M16C/1N group has a CAN module system clock select circuit.

Configuration of the CAN module system clock can be done through manipulating the CCLKR register and the BRP bit of the C0CONR register.

For the CCLKR register, refer to clock generation circuit.

Figure 16.18 shows a block diagram of the clock generation circuit of the CAN module system.



fCAN: CAN module system clock
P: The value written in the BRP bit of the C0CONR register. P = 0 to 15
fCANCLK: CAN communication clock fCANCLK = fCAN/2(P + 1)

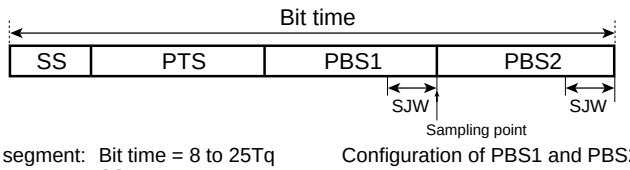
Figure 16.18 Block Diagram of CAN Module System Clock Generation Circuit

16.6.1 Bit Timing Configuration

The bit time consists of the following four segments:

- Synchronization segment (SS)
  This serves for monitoring a falling edge for synchronization.
- Propagation time segment (PTS)
  This segment absorbs physical delay on the CAN network which amounts to double the total sum of delay on the CAN bus, the input comparator delay, and the output driver delay.
- Phase buffer segment 1 (PBS1)
  This serves for compensating the phase error. When the falling edge of the bit falls later than expected, the segment can become longer by the maximum of the value defined in SJW.
- Phase buffer segment 2 (PBS2)
  This segment has the same function as the phase buffer segment 1. When the falling edge of the bit falls earlier than expected, the segment can become shorter by the maximum of the value defined in SJW.

Figure 16.19 shows the bit timing.



The range of each segment: Bit time = 8 to 25Tq
SS = 1Tq
PTS = 1Tq to 8Tq
PBS1 = 2Tq to 8Tq
PBS2 = 2Tq to 8Tq
SJW = 1Tq to 4Tq

Configuration of PBS1 and PBS2: PBS1 ≥ PBS2
PBS1 ≥ SJW
PBS2 ≥ 2 when SJW = 1
PBS2 ≥ SJW when 2 ≤ SJW ≤ 4

Figure 16.19 Bit Timing

16.6.2 Baud Rate

Baud rate depends on X_{IN}, the division value of the CAN module system clock, the division value of the prescaler for baud rate, and the number of Tq of one bit.

Table 16.2 shows the examples of baud rate.

Table 16.2 Examples of Baud Rate

Baud rate	16 MHz	10 MHz	8 MHz
1 Mbps	8Tq (1)	–	–
500 kbps	8Tq (2) 16Tq (1)	10Tq (1) –	8Tq (1) –
125 kbps	8Tq (8) 16Tq (4)	10Tq (4) 20Tq (2)	8Tq (4) 16Tq (2)
83.3 kbps	8Tq (12) 16Tq (6)	10Tq (6) 20Tq (3)	8Tq (6) 16Tq (3)
33.3 kbps	8Tq (30) 16Tq (15)	10Tq (15) –	8Tq (15) –

Note 1: The number in ( ) indicates a value of f_{CAN} division value multiplied by division value of the prescaler for baud rate.

■ **Calculation of Baud Rate**

$$\frac{X_{IN}}{2 \times f_{CAN} \text{ division value (Note 1)} \times \text{division value of prescaler for baud rate (Note 2)} \times \text{number of Tq of one bit}}$$

Note 1: f_{CAN} division value = 1, 2, 4, 8, 16
f_{CAN} division value: a value selected in the CCLKR register

Note 2: Division value of prescaler for baud rate = P + 1 (P: 0 to 15)
P: a value selected in the BRP bit of the C0CONR register

16.7 Acceptance Filtering Function and Masking Function

These functions serve the users to select and receive a facultative message. The C0GMR register, the C0LMAR register, and the C0LMBR register can perform masking to the standard ID and the extended ID of 29 bits. The C0GMR register corresponds to slots 0 to 13, the C0LMAR register corresponds to slot 14, and the C0LMBR register corresponds to slot 15. The masking function becomes valid to 11 bits or 29 bits of a received ID according to the value in the corresponding slot of the C0IDR register upon acceptance filtering operation. When the masking function is employed, it is possible to receive a certain range of IDs. Figure 16.20 shows correspondence of the mask registers and slots, Figure 16.21 shows the acceptance function.

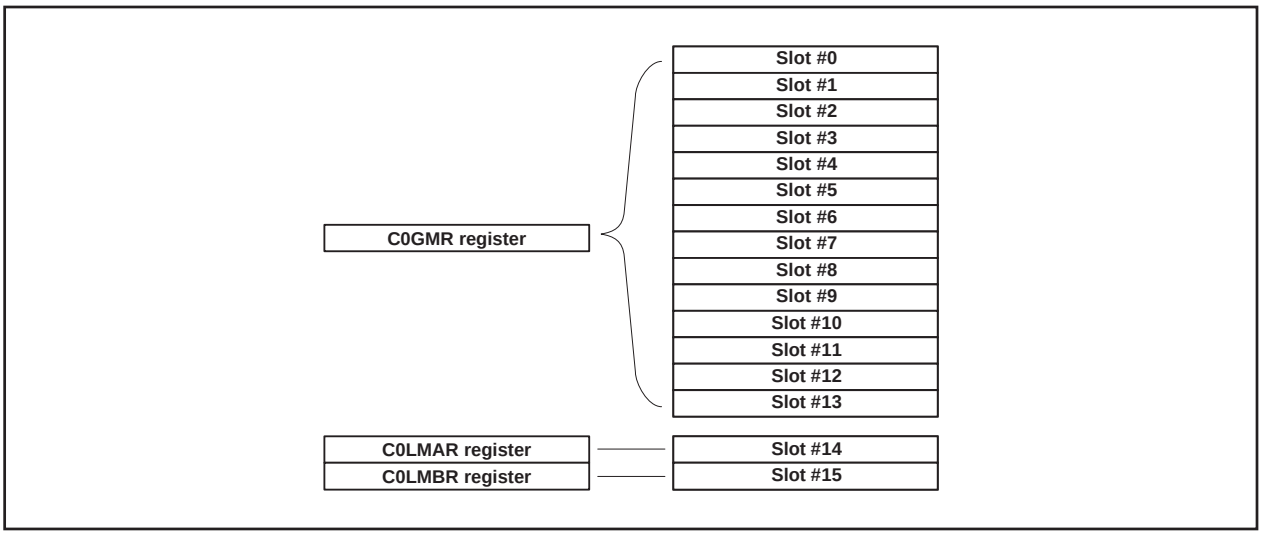


Figure 16.20 Correspondence of Mask Registers to Slots

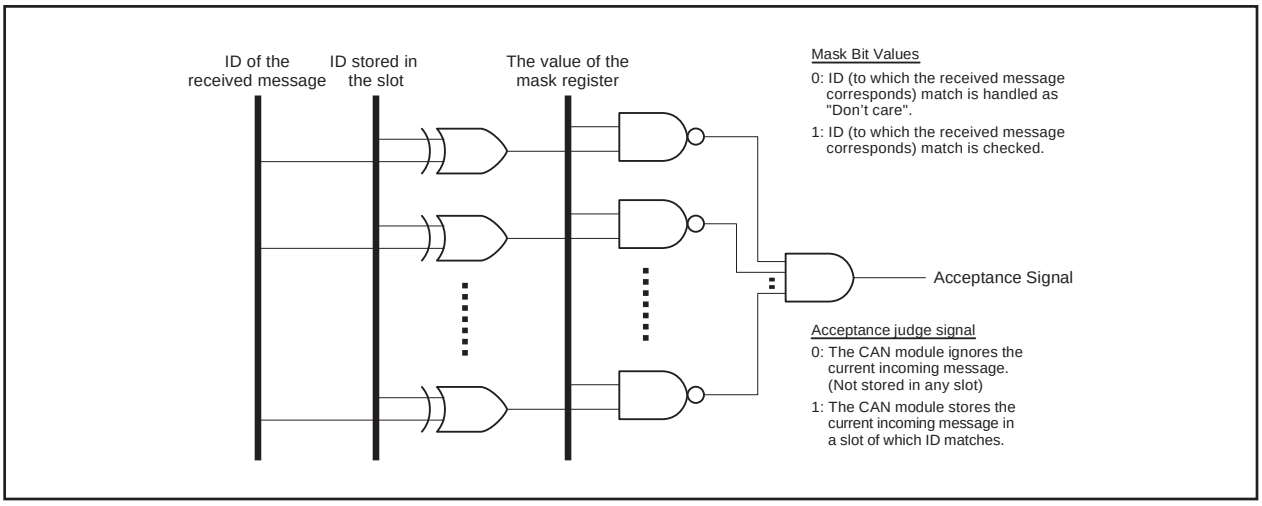


Figure 16.21 Acceptance Function

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.

16.8 Acceptance Filter Support Unit (ASU)

The acceptance filter support unit has a function to judge valid/invalid of a received ID through table search. The IDs to receive are registered in the data table; a received ID is stored in the C0AFS register, and table search is performed with a decoded received ID. The acceptance filter support unit can be used for the IDs of the standard frame only.

The acceptance filter support unit is valid in the following cases.

- When the ID to receive cannot be masked by the acceptance filter.
  (Example) IDs to receive: 078_{16}, 087_{16}, 111_{16}
- When there are too many IDs to receive; it would take too much time to filter them by software.

Figure 16.22 shows the write and read of C0AFS register in word access.

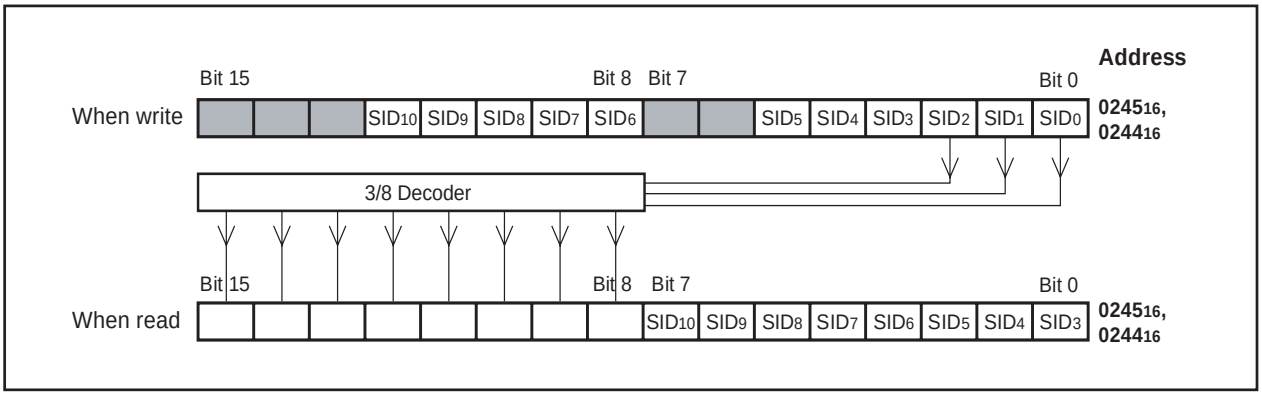


Figure 16.22 Write/read of CiAFS Register in Word Access

16.9 Basic CAN Mode

When the BasicCAN bit in the C0CTLR register is set to "1", 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 C0MCTLi register (i = 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 C0MCTLi register.

Figure 16.23 shows the operation of slots 14 and 15 in Basic CAN mode.

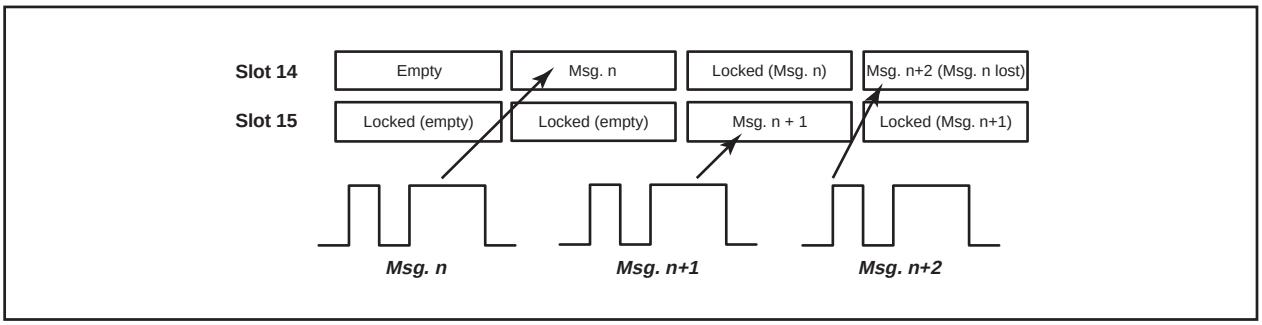


Figure 16.23 Operation of Slots 14 and 15 in Basic CAN Mode

When using Basic CAN mode, note the following points.

(1) Setting of Basic CAN mode has to be done in CAN reset/initialization mode.

(2) Select the same ID for slots 14 and 15. Also, setting of the C0LMAR and C0LMBR registers 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.

16.10 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 the return from bus off function of the C0CTLR register. At this time, the error state changes from bus off state to error active state. Implementation of this function initializes the protocol controller. However, registers of the CAN module such as C0CONR register and the content of each slot are not initialized.

16.11 Listen-Only Mode

When the RXOnly bit of the C0CTLR register is set to "1", the module enters listen-only mode.

Listen-only mode is not allowed to have any influence on the bus. It shall not send any frames nor send acknowledgement, error frames, overload frames. When setting the CAN module to Listen-only mode, do not request a transmission.

16.12 Reception and Transmission

Configuration of CAN Reception and Transmission Mode

Table 16.3 shows configuration of CAN reception and transmission mode.

Table 16.3 Configuration of CAN Reception and Transmission Mode

TrmReq	RecReq	Remote	RspLock	Communication mode of the slot
0	0	——	——	Communication environment configuration mode: configure the communication mode of the slot.
0	1	0	0	Configured as a reception slot for a data frame.
1	0	1	0	Configured as a transmission slot for a remote frame. (At this time the RemActive bit is "1".) After completion of transmission, this functions as a reception slot for a data frame. (At this time the RemActive bit is "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 bit is "1".) After completion of reception, this functions as a transmission slot for a data frame. (At this time the RemActive bit is "0".) However, transmission does not start as long as RspLock bit remains "1"; thus no automatic remote frame response. Response (transmission) starts when RspLock bit is set to "0".

TrmReq, RecReq, Remote, RspLock, RemActive, RspLock: C0MCTLi register's bit

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 C0MCTLi registers (i = 0 to 15) to "00$_{16}$".

(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 C0MCTLi registers to "00$_{16}$".

(2) Set the TrmReq bit to "0" (not transmission slot) before rewriting a transmission slot.

(3) A transmission slot should not be rewritten when the TrmActive bit is "1" (transmitting). If it is rewritten, an indeterminate data will be transmitted.

16.12.1 Reception

Figure 16.24 shows the behavior of the module when receiving two consecutive CAN messages.

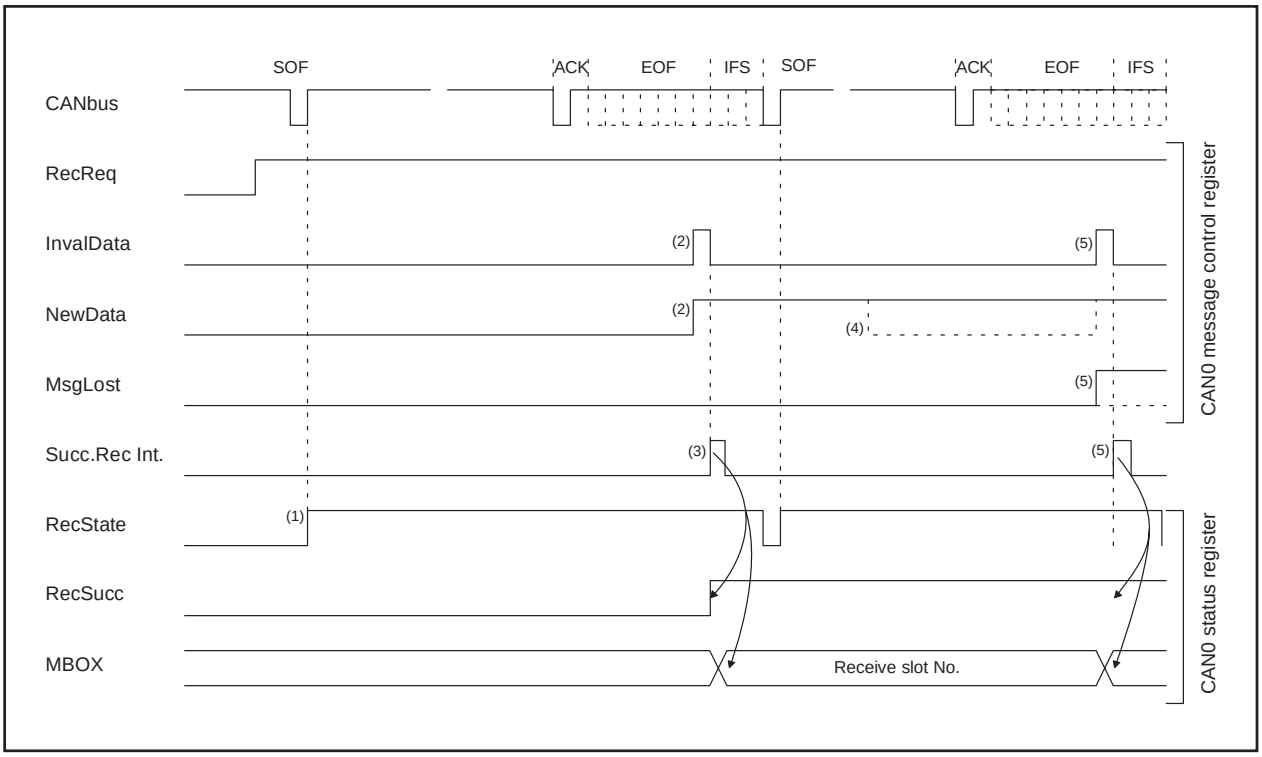


Figure 16.24 Timing of Receive Data Frame Sequence

(1) On monitoring a SOF on the bus the RecState bit becomes active immediately, given the module has no transmission pending (see section "16.12.2 Transmission" below).

(2) After successful reception of the message the NewData bit of the receiving slot becomes active. The InvalData bit becomes active at the same time and becomes inactive again after the complete message was transferred to the slot.

(3) When the bit in the C0ICR register of the receiving slot is active the receive successful interrupt is requested and the C0STR register changes. It shows the slot number where the message was stored and the RecSucc bit is active.

(4) Read the message out of the slot after setting the New Data bit to “0” 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 reception request for the slot is canceled, the MsgLost bit is set to "1". The new received message is transferred to the slot. The interrupt request and the C0STR register change like (3).

16.12.2 Transmission

Figure 16.25 shows the timing of the transmit sequence.

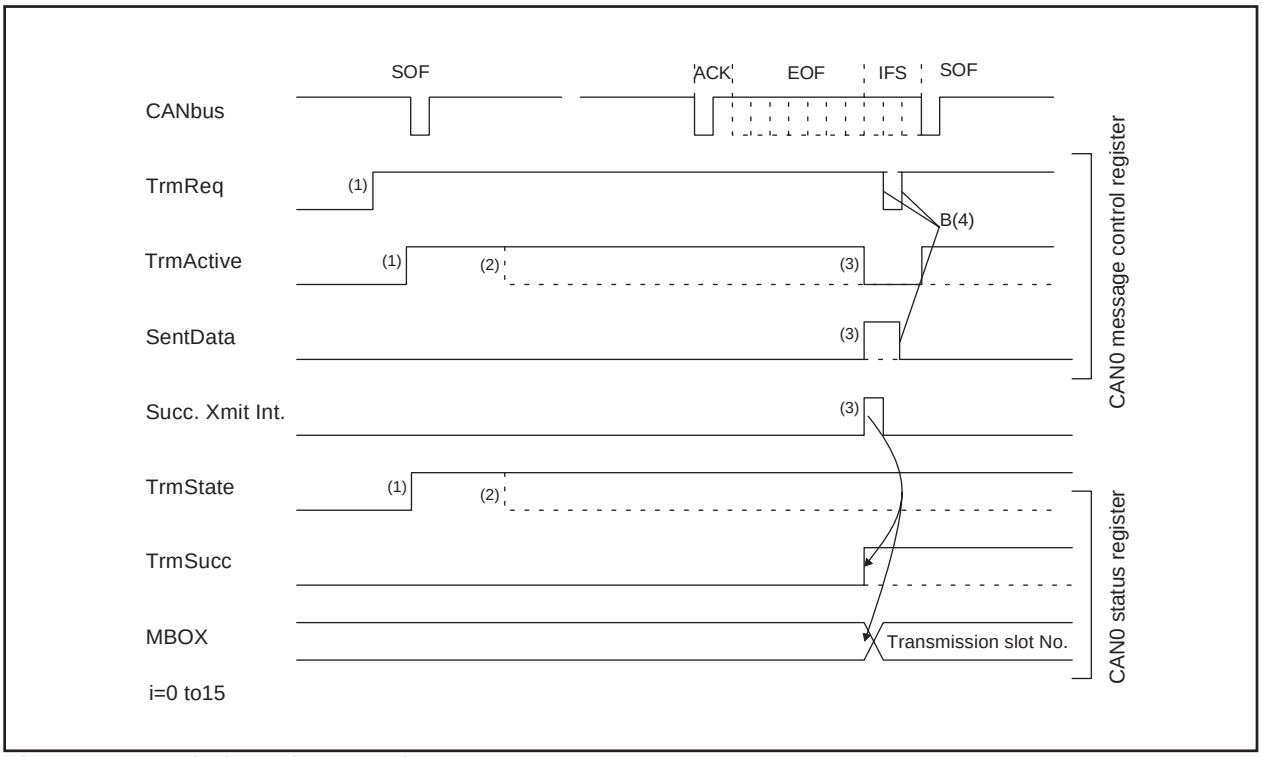


Figure 16.25 Timing of Transmit Sequence

(1) If the TrmReq bit of the C0MCTLi register (i=0 to 15) is set to "1" (Transmission slot) in bus idle state, the TrmActive bit of the C0MCTLi register and the TrmState bit of the C0STR register are set to "1" (Transmitting/Transmitter), and the CAN module starts transmitting.

(2) If the arbitration is lost after the CAN module starts transmitting, the TrmActive and TrmState bits are set to "0".

(3) If the transmission is successful without lost arbitration, the SentData bit of the C0MCTLi register is set to "1" (Transmission is successfully completed) and TrmActive bit of the C0MCTLj register is set to "0" (Waiting for bus idle or completion of arbitration). And when the interrupt enable bits of the C0ICR register = 1 (Interrupt enabled), CAN0 successful transmission interrupt request is generated and the MBOX (the slot number which transmitted the message) and TrmSucc bits of the C0STR register are changed.

(4) When starting the next transmission, set the SentData and TrmReq bits to "0", then set the TrmReq bit to "1" after checking that the SentData and TrmReq bits are set to "0".

16.13 CAN Interrupts

The CAN module provides the following CAN interrupts.

- CAN0 Successful Reception Interrupt
- CAN0 Successful Transmission Interrupt
- CAN0 Error Interrupt
  - Error Passive State
  - Error BusOff State
  - Bus Error (this feature can be disabled separately)
- CAN0 Wake Up Interrupt

When the CPU detects a successful reception/transmission interrupt, the C0STR register must be read to determine which slot has issued the interrupt.

17. Programmable I/O Ports

17.1 Description

There are 37 programmable I/O ports: P0 to P5. Each port can be set independently for input or output using the direction register. A pull-up resistance for each block of 4 ports can be set. The port P1 allows the drive capacity of its N-channel output transistor to be set as necessary. The port P1 can be used as LED drive port if the drive capacity is set to "HIGH".

Figures 17.1 to 17.4 show the programmable I/O ports.

Each pin functions as a programmable I/O port and as the I/O for the built-in peripheral devices.

To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices (other than the D/A converter), they function as outputs regardless of the contents of the direction registers. When a pin is to be used as the output for the D/A converter, do not set the direction register to output mode. See the descriptions of the respective functions for how to set up the built-in peripheral devices.

17.1.1 Direction registers

Figure 17.5 shows the direction registers.

These registers are used to choose the direction of the programmable I/O ports. Each bit in these registers corresponds one for one to each I/O pin.

17.1.2 Port registers

Figure 17.6 shows the port registers.

These registers are used to write and read data for input and output to and from an external device. A port register consists of a port latch to hold output data and a circuit to read the status of a pin. Each bit in port registers corresponds one for one to each I/O pin.

17.1.3 Pull-up control registers

Figure 17.7 shows the pull-up control registers.

The pull-up control register can be set to apply a pull-up resistance to each block of 4 ports. When ports are set to have a pull-up resistance, the pull-up resistance is connected only when the direction register is set for input.

17.1.4 Port P1 drive capacity control register

Figure 17.7 shows a structure of the port P1 drive capacity control register.

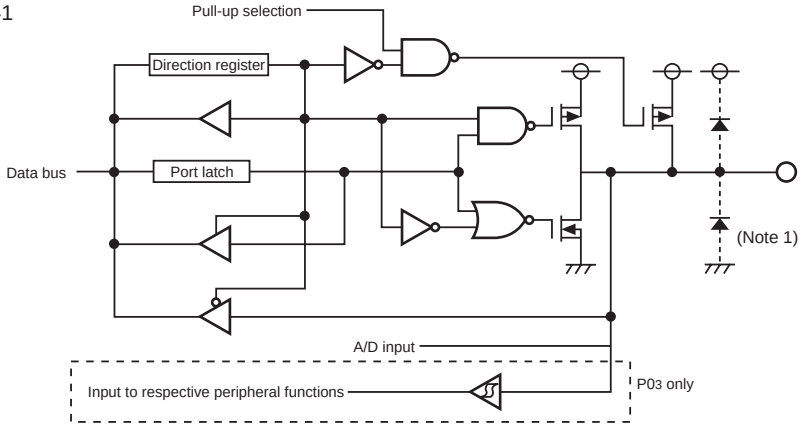
This register is used to control the drive capacity of the port P1's N-channel output transistor. Each bit in this register corresponds one for one to the port pins.

17.1.5 CAN0 I/O port selected register

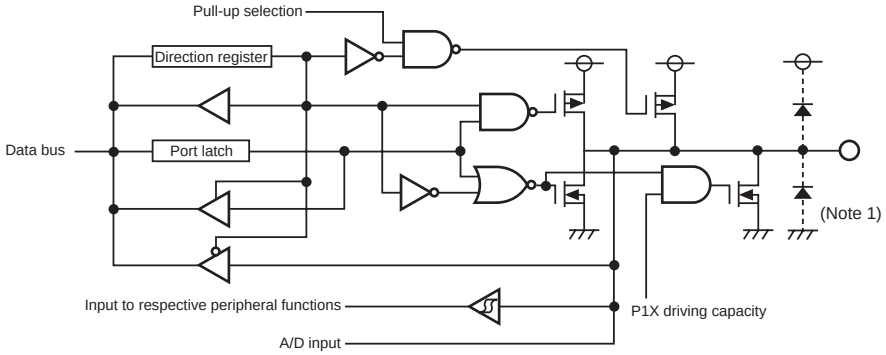
Figure 17.8 shows the CAN0 I/O port selected register.

This register is used to select I/O port for CAN0.

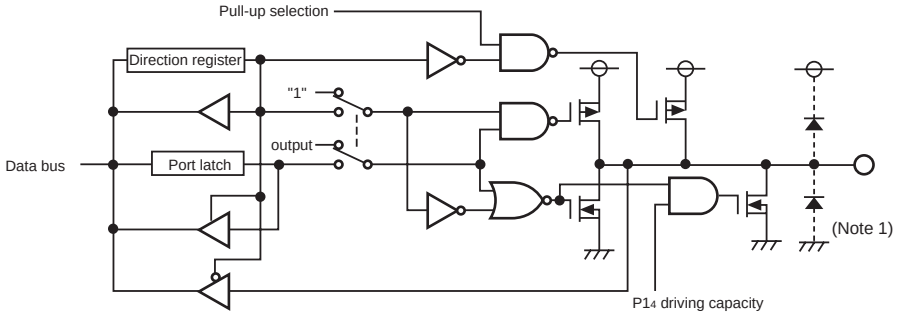
P0$_0$ to P0$_7$, P4$_0$, P4$_1$



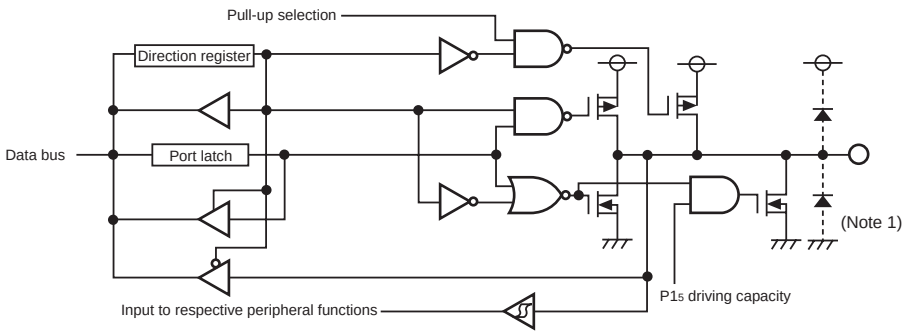
P1$_0$ to P1$_3$



P1$_4$



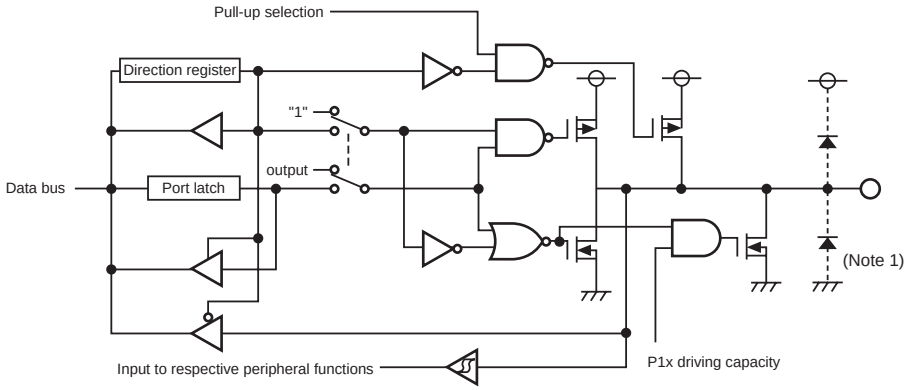
P1$_5$



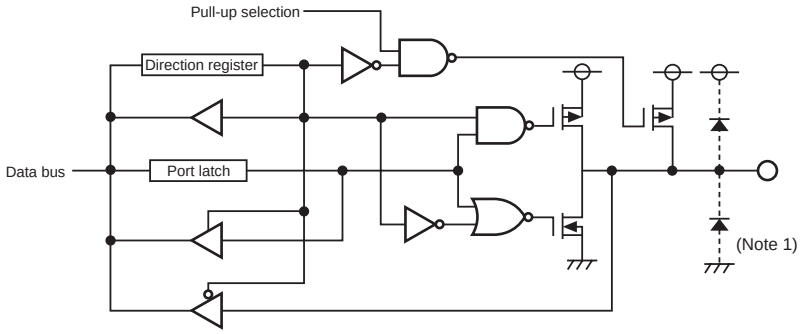
Note 1: Do not apply a voltage higher than Vcc to each port.

Figure 17.1 Programmable I/O ports (1)

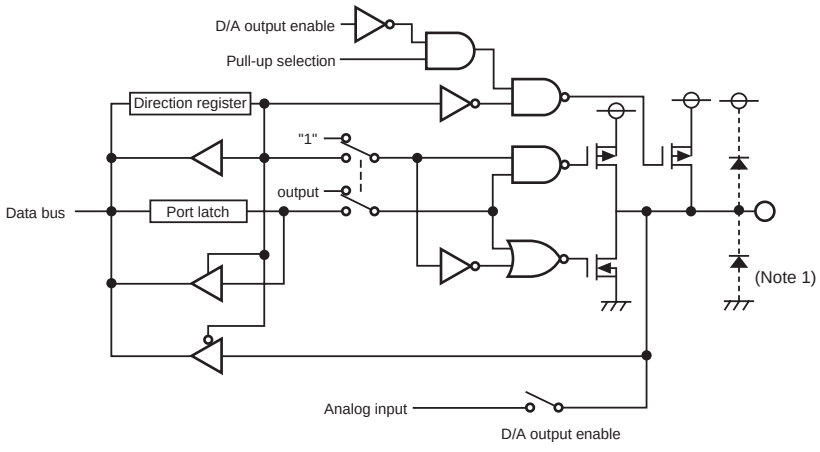
P16, P17



P20, P21
P52



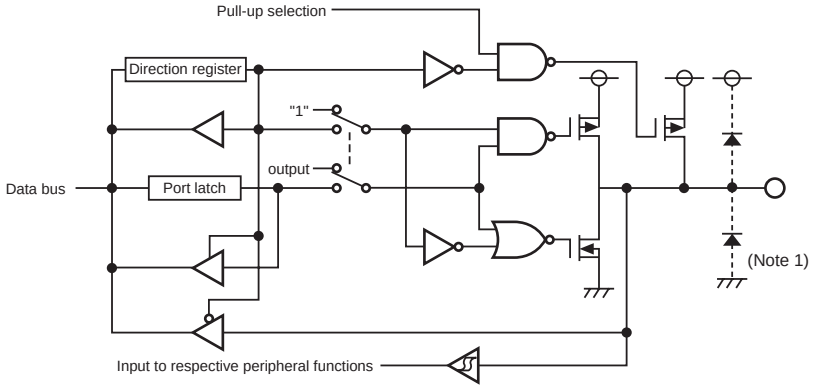
P34



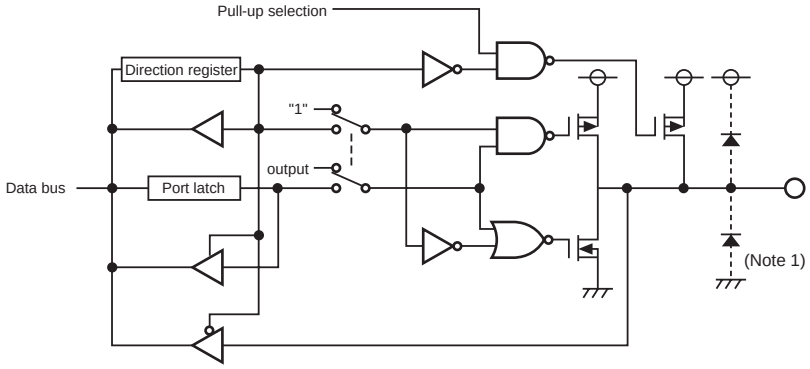
Note 1: Do not apply a voltage higher than Vcc to each port.

Figure 17.2 Programmable I/O ports (2)

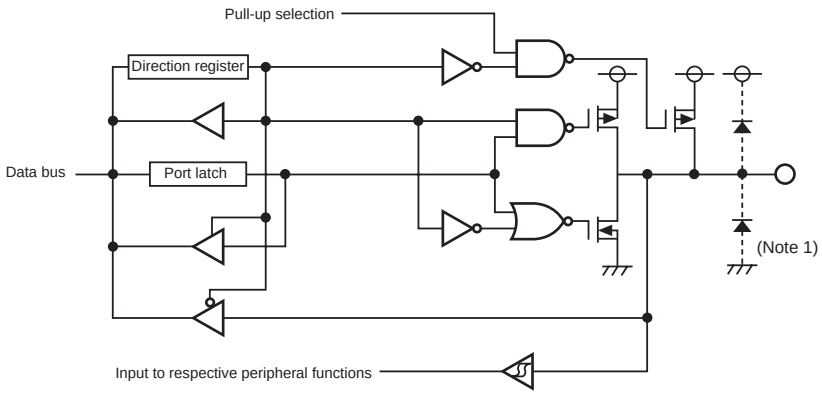
P3$_6$, P3$_7$



P3$_0$, P3$_1$, P3$_2$
P5$_0$



P3$_3$, P3$_5$, P4$_2$ to P4$_4$
P5$_1$



Note 1: Do not apply a voltage higher than Vcc to each port.

Figure 17.3 Programmable I/O ports (3)

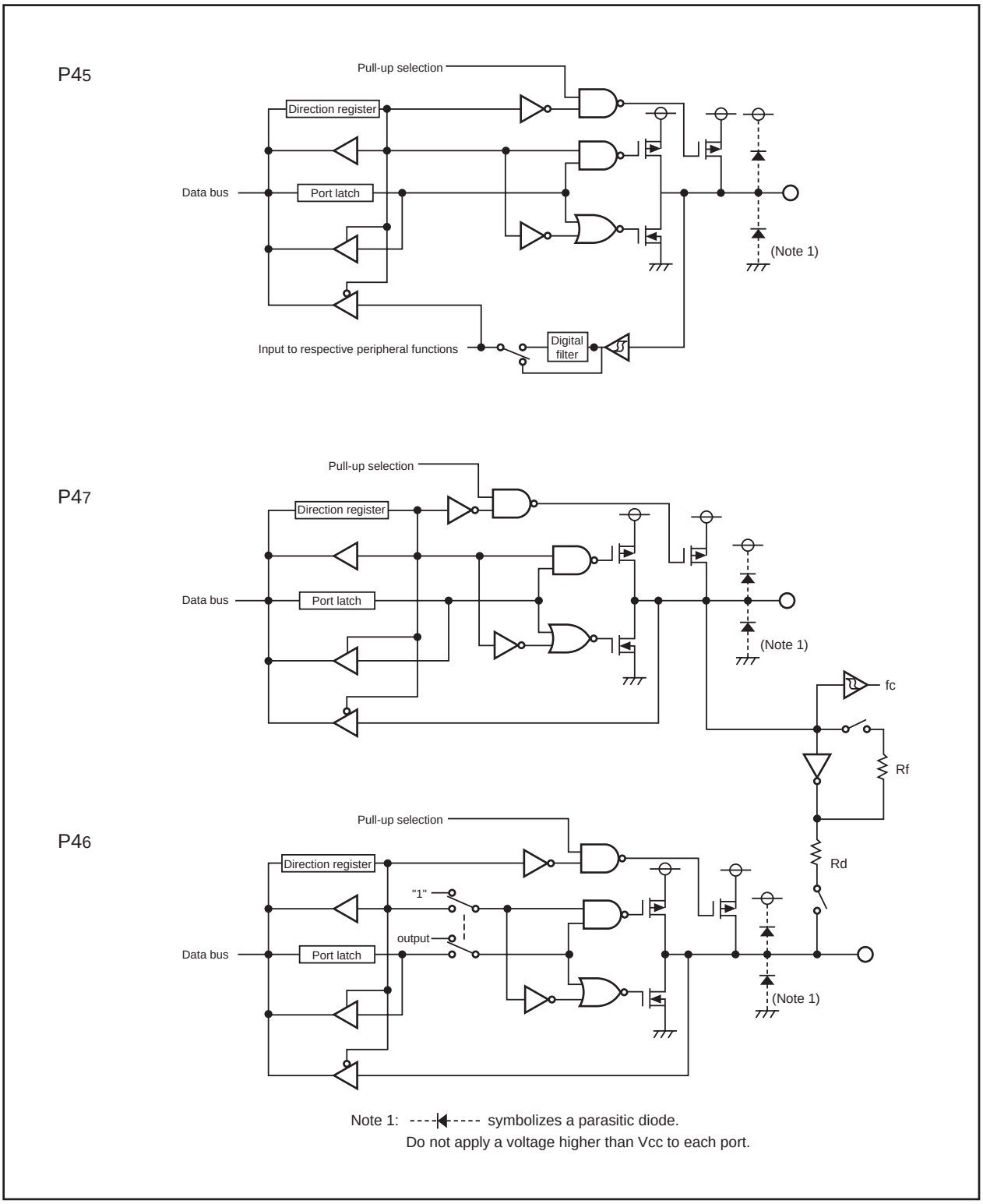
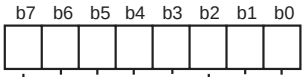


Figure 17.4 Programmable I/O ports (4)

Port Pi direction register (Note 1, 2, 3)



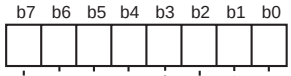
Symbol	Address	When reset
PDi (i = 0 to 5)	$00E2_{16}$, $00E3_{16}$ $00E6_{16}$ $00E7_{16}$, $00EA_{16}$ $00EB_{16}$	00_{16} $XXXXXX00_2$ 00_{16} $XXXXX000_2$

Bit symbol	Bit name	Function	RW
PDi_0	Port Pi_0 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 5)	RW
PDi_1	Port Pi_1 direction register		RW
PDi_2	Port Pi_2 direction register		RW
PDi_3	Port Pi_3 direction register		RW
PDi_4	Port Pi_4 direction register		RW
PDi_5	Port Pi_5 direction register		RW
PDi_6	Port Pi_6 direction register		RW
PDi_7	Port Pi_7 direction register		RW

Note 1: Set bit 2 of protect register (address $000A_{16}$) to "1" before rewriting to the port P0 direction register.
Note 2: Nothing is assigned in bit 2 to bit 7 of port P2 direction register. When write, set "0". When read, their contents are "0".
Note 3: Nothing is assigned in bit 3 to bit 7 of port P5 direction register.
When write, set "0".
When read, their contents are "0".

Figure 17.5 Port Pi direction register

Port Pi register (Note 1, 2)



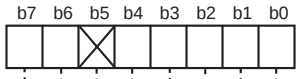
Symbol	Address	When reset
Pi (i = 0 to 5)	$00E0_{16}$, $00E1_{16}$, $00E4_{16}$, $00E5_{16}$, $00E8_{16}$, $00E9_{16}$	Indeterminate Indeterminate

Bit symbol	Bit name	Function	RW
Pi_0	Port Pi_0 register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : "L" level data 1 : "H" level data (i = 0 to 5)	RW
Pi_1	Port Pi_1 register		RW
Pi_2	Port Pi_2 register		RW
Pi_3	Port Pi_3 register		RW
Pi_4	Port Pi_4 register		RW
Pi_5	Port Pi_5 register		RW
Pi_6	Port Pi_6 register		RW
Pi_7	Port Pi_7 register		RW

Note 1: Nothing is assigned in bit 2 to bit 7 of port P2 register.
When write, set "0".
When read, their contents are "0".
Note 2: Nothing is assigned in bit 3 to bit 7 of port P5 register.
When write, set "0".
When read, their contents are "0".

Figure 17.6 Port Pi register

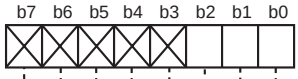
Pull-up control register 0



Symbol	Address	When reset
PUR0	00FC16	00X000002

Bit symbol	Bit name	Function	RW
PU00	P00 to P03 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	RW
PU01	P04 to P07 pull-up		RW
PU02	P10 to P13 pull-up		RW
PU03	P14 to P17 pull-up		RW
PU04	P20 to P21 pull-up		RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—
PU06	P30 to P33 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	RW
PU07	P34 to P37 pull-up		RW

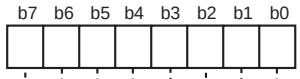
Pull-up control register 1



Symbol	Address	When reset
PUR1	00FD16	XXXXX0002

Bit symbol	Bit name	Function	RW
PU10	P40 to P43 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	RW
PU11	P44 to P47 pull-up		RW
PU12	P50 to P52 pull-up		RW
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—

Port P1 drive capacity control register

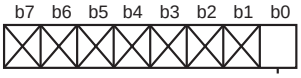


Symbol	Address	When reset
DRR	00FE16	0016

Bit symbol	Bit name	Function	RW
DRR0	Port P10 drive capacity	Set P1 N-channel output transistor drive capacity 0 : LOW 1 : HIGH	RW
DRR1	Port P11 drive capacity		RW
DRR2	Port P12 drive capacity		RW
DRR3	Port P13 drive capacity		RW
DRR4	Port P14 drive capacity		RW
DRR5	Port P15 drive capacity		RW
DRR6	Port P16 drive capacity		RW
DRR7	Port P17 drive capacity		RW

Figure 17.7 Pull-up control register 0, Pull-up control register 1 and Port P1 drive capacity control register

CAN0 I/O pin select register (Note 1)



Symbol	Address	When reset
CIOSR	$00F8_{16}$	$XXXXXXX0_2$

Bit symbol	Bit name	Function	RW
CIO0C	CAN0 I/O pin	0 : CTx = $P0_2$ pin CRx = $P0_3$ pin 1 : CTx = $P5_0$ pin CRx = $P5_1$ pin	RW

Note 1: Set bit 2 of protect register (address $000A_{16}$) to "1" before rewriting to this register.

Figure 17.8 CAN0 I/O pin select register

17.2 Example connection of unused pins

Table 17.1 shows example connection of unused pins.

Table 17.1 Example connection of unused pins

Pin name	Connection
Ports P0 to P5 (Note 1)	After setting for input mode, connect every pin to VSS (pull-down); or after setting for output mode, leave these pins open.
XOUT (Note 2)	Open
VREF	Connect to VSS
XIN (Note 3)	Connect to VCC (pull-up) via a resistor

Note 1: Connect unused pins as described above. If connected otherwise, power supply current may increase due to flow-through current on Schmitt circuit in the port.
Note 2: With external clock input to XIN pin.
Note 3: When the main clock oscillation circuit isn't used, connect XIN pin to VCC (pull-up), leave XOUT pin open and set the main clock stop bit (bit 5 at address 0006_{16}) to "1" (STOP).

18. Electrical Characteristics

Table 18.1 Absolute maximum ratings

Symbol	Parameter		Condition	Rated value	Unit
Vcc	Supply voltage			- 0.3 to 6.5	V
V_I	Input voltage	$\overline{RESET}$, V_{REF}, X_{IN} $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$, $P2_1$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_2$, CNVss (Note 1)		- 0.3 to Vcc + 0.3	V
V_O	Output voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$, $P2_1$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_2$, X_{OUT}		- 0.3 to Vcc + 0.3	V
		IV_{CC}		- 0.3 to 2.8V	V
P_d	Power dissipation		Topr = 25 °C	300	mW
T_{opr}	Operating ambient temperature			- 40 to 85 (Note 2)	°C
T_{stg}	Storage temperature			- 65 to 150	°C

Note 1: CNVSS pin of flash memory version: -0.3 to 6.5 V

Note 2: When flash memory version is program/erase mode: 0 to 60 °C

Table 18.2 Recommended operating conditions
(Unless otherwise noted: Vcc = 4.2V to 5.5V, Topr = -40 to 85°C)

Symbol	Parameter			Standard			Unit
				Min	Typ.	Max.	
Vcc	Supply voltage			4.2	5.0	5.5	V
Vss	Supply voltage				0		V
VIH	HIGH input voltage	P00 to P07, P10 to P17, P20, P21, P30 to P37, P40 to P47, P50 to P52, XIN, RESET, CNVSS		0.8Vcc		Vcc	V
VIL	LOW input voltage	P00 to P07, P10 to P17, P20, P21, P30 to P37, P40 to P47, P50 to P52, XIN, RESET, CNVSS		0		0.2Vcc	V
IOH (peak)	HIGH peak output current	P00 to P07, P10 to P17, P20, P21, P30 to P37, P40 to P47, P50 to P52				- 10.0	mA
IOH (avg)	HIGH average output current	P00 to P07, P10 to P17, P20, P21, P30 to P37, P40 to P47, P50 to P52				- 5.0	mA
IOL (peak)	LOW peak output current	P00 to P07, P20, P21, P30 to P37, P40 to P47, P50 to P52				10.0	mA
		P10 to P17	HIGH POWER			20.0	mA
			LOW POWER			10.0	
IOL (avg)	LOW average output current	P00 to P07, P20, P21, P30 to P37, P40 to P47, P50 to P52				5.0	mA
		P10 to P17	HIGH POWER			10.0	mA
			LOW POWER			5.0	
f (XIN)	Main clock input oscillation frequency (Note 3)		Vcc=4.2V to 5.5V	0		16	MHz
f (XCIN)	Subclock oscillation frequency				32.768	50	kHz

Note 1: The average output current is an average value measured over 100ms.

Note 2: Keep output current as follows:

The sum of port P00 to P03, P13 to P17, P21, P34 to P37, P46, P47, P50 to P52 IOL (peak) is under 60 mA. The sum of port P00 to P03, P13 to P17, P21, P34 to P37, P46, P47, P50 to P52 IOH (peak) is under 60 mA. The sum of port P04 to P07, P10 to P12, P20, P30 to P33, P40 to P45 IOL (peak) is under 60 mA. The sum of port P04 to P07, P10 to P12, P20, P30 to P33, P40 to P45 IOH (peak) is under 60 mA.

Note 3: Relationship between main clock oscillation frequency and supply voltage is shown as below.

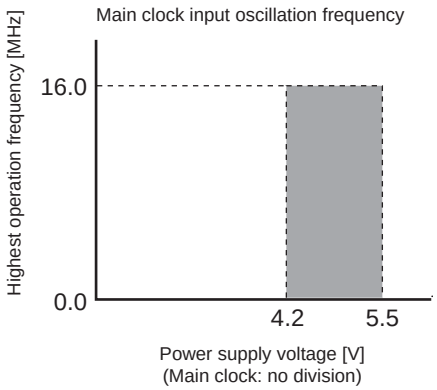


Table 18.3 Electrical characteristics (1)
(Unless otherwise noted: V_{CC} = 5V, V_{SS} = 0V at Topr = -40 to 85°C, f(X_{IN}) = 16MHz)

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 $P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$,$P5_0$ to $P5_2$		I_{OH} = - 5 mA	3.0			V
			I_{OH} = - 200 µA	4.7			
V_{OH}	HIGH output voltage X_{OUT}	HIGH POWER	I_{OH} = - 1 mA	3.0			V
		LOW POWER	I_{OH} = - 0.5 mA	3.0			
V_{OH}	HIGH output voltage X_{COUT}	HIGH POWER	No load		2.5		V
		LOW POWER	No load		1.6		
V_{OL}	LOW output voltage $P0_0$ to $P0_7$,$P2_0$,$P2_1$,$P3_0$ to $P3_7$, $P4_0$ to $P4_7$,$P5_0$ to $P5_2$		I_{OL} = 5 mA			2.0	V
			I_{OL} = 200 µA			0.45	
V_{OL}	LOW output voltage $P1_0$ to $P1_7$	HIGH POWER	I_{OL} = 10 mA			2.0	V
		LOW POWER	I_{OL} = 5 mA			2.0	
V_{OL}	LOW output voltage X_{OUT}	HIGH POWER	I_{OH} = 1 mA			2.0	V
		LOW POWER	I_{OH} = 0.5 mA			2.0	
V_{OL}	LOW output voltage X_{COUT}	HIGH POWER	No load		0		V
		LOW POWER	No load		0		
V_{T+} -V_{T-}	Hysteresis $\overline{CNTR_0}$,$\overline{TCIN}$, $\overline{INT_0}$ to $\overline{INT_3}$,$\overline{CLK_0}$,$\overline{CLK_1}$,$P4_5$ RxD_0,RxD_1,$\overline{KI_0}$ to $\overline{KI_3}$,CRX_0			0.2		0.8	V
V_{T+} -V_{T-}	Hysteresis $\overline{RESET}$			0.2		1.8	V
I_{IH}	HIGH input current $P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$,$P2_1$, $P3_0$ to $\overline{P3_7}$,$P4_0$ to $P4_7$,$P5_0$ to $P5_2$, X_{IN},$\overline{RESET}$,CNVss		V_I = 5V			5.0	µA
I_{IL}	LOW input current $P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$,$P2_1$, $P3_0$ to $\overline{P3_7}$,$P4_0$ to $P4_7$,$P5_0$ to $P5_2$, X_{IN},$\overline{RESET}$,CNVss		V_I = 0V			-5.0	µA
R_{PULLUP}	Pull-up resistor $P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$,$P2_1$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$,$P5_0$ to $P5_2$		V_I = 0V	30.0	50.0	167.0	kΩ
R_{fXIN}	Feedback resistor X_{IN}				1.0		MΩ
R_{fXCIN}	Feedback resistor X_{CIN}				15.0		MΩ
V_{RAM}	RAM retention voltage		When clock is stopped	2.0			V
R_{OSC}	Oscillation frequency of On-chip oscillator	Mask ROM		300	600	1200	kHz
		Flash memory					

Table 18.4 Electrical characteristics (2)
(Unless otherwise noted: V_{CC} = 5V, V_{SS} = 0V at Topr = 25°C, f(X_{IN}) = 16MHz)

Symbol	Parameter	Measuring condition			Standard			Unit
					Min.	Typ.	Max.	
Icc	Power supply current	I/O pin has no load	Mask ROM	f(X_{IN}) = 16 MHz Square wave, no division		12.0	22.0	mA
			Flash memory			14.0	24.0	mA
			Mask ROM	On-chip oscillator mode No division		300		µA
			Flash memory			800		µA
			Mask ROM	On-chip oscillator mode When a WAIT instruction is executed		60		µA
			Flash memory			100		µA
			Mask ROM	f(X_{CIN}) = 32 kHz Square wave		20		µA
			Flash memory			450		µA
			Mask ROM	f(X_{CIN}) = 32 kHz When a WAIT instruction is executed		2		µA
			Flash memory	f(X_{CIN}) = 32 kHz When a WAIT instruction is executed		2		µA
			Mask ROM	Topr = 25 °C when clock is stopped		0.8	3	µA
			Flash memory			0.8	3	µA

Table 18.5 Power supply timing circuit characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Timer for internal power supply stabilization during powering-on	V_{CC} = 4.2 to 5.5 V			2	ms
td(R-S)	Stop release time				150	µs
td(W-S)	Wait release time during low power dissipation mode				150	µs
td(M-L)	Timer for internal power supply stabilization when main clock oscillation starts				150	µs

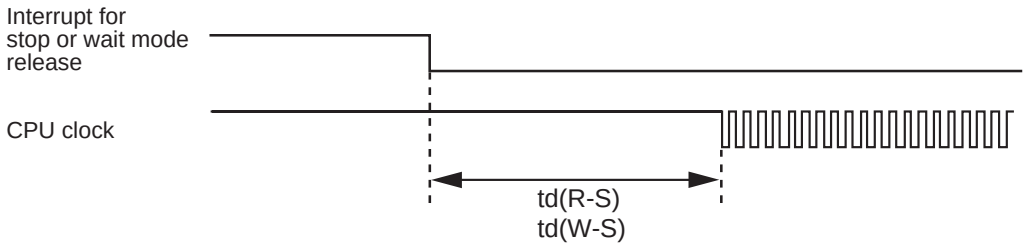


Table 18.6 Flash memory version electrical characteristics (Unless otherwise noted: Vcc = 4.2 to 5.5 V, Topr= 0 to 60°C)

Symbol	Parameter		Standard			Unit
			Min.	Typ. (Note 1)	Max.	
-	Erase/write cycle (Note 2)		100 (Note 3)			cycle
-	Word programming time			75	600	µs
-	Block erasing time	2Kbyte block		0.2	9	s
		8Kbyte block		0.4	9	s
		16Kbyte block		0.7	9	s
		32Kbyte block		1.2	9	s
td(SR-ES)	Transition time from erasure operation to erase-suspend				20	ms
-	Data retention		10			year

Note1: Vcc=5.0V, Topr=25°C

Note2: Definition of Programming and erasure times

The Programming and erasure times are defined to be per-block erasure times. For example a case where a 2K-byte block is programmed in 1,024 operations by writing one word at a time and erased thereafter. Performing multiple programs to the same address before an erase operation is prohibited.

Note 3: Minimum number of programming/erasure for which operation is guaranteed.

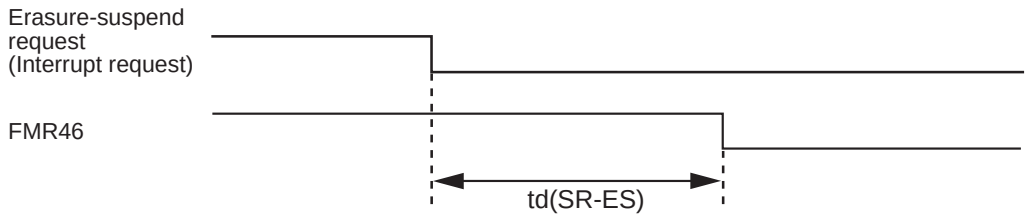


Table 18.7 A/D conversion characteristics
(Unless otherwise noted: V_{CC} = V_{REF} = 5V, V_{SS} = 0V at Topr = 25°C, f(X_{IN}) = 16MHz)

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
–	Resolution		V_{REF} =V_{CC}				10	Bits
–	Absolute accuracy	Sample & hold function not available	V_{REF} =V_{CC} = 5V				±3	LSB
		Sample & hold function available(10bit)	V_{REF} =V_{CC} = 5V	AN_0 to AN_{11} input			±3	LSB
				$ANEX_0$, $ANEX_1$ input, external op-amp connected mode			±7	LSB
		Sample & hold function available(8bit)	V_{REF} =V_{CC} = 5V				±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} =V_{CC}		10		40	kΩ
t_{CONV}	Conversion time(10bit)		f(X_{IN})=10MHz, $Ø_{AD}$=f_{AD}=10MHz		3.3			µs
t_{CONV}	Conversion time(8bit)		f(X_{IN})=10MHz, $Ø_{AD}$=f_{AD}=10MHz		2.8			µs
t_{SAMP}	Sampling time		f(X_{IN})=10MHz, $Ø_{AD}$=f_{AD}=10MHz		0.3			µs
V_{REF}	Reference voltage		f(X_{IN})=10MHz, $Ø_{AD}$=f_{AD}=10MHz		2		V_{CC}	V
V_{IA}	Analog input voltage		f(X_{IN})=10MHz, $Ø_{AD}$=f_{AD}=10MHz		0		V_{REF}	V

Note 1: Divide the f_{AD} if f(X_{IN}) exceeds 10MHz, and make AD operation clock frequency ($Ø_{AD}$) equal to or lower than 10MHz.

Table 18.8 D/A conversion characteristics
(Unless otherwise noted: V_{CC} = V_{REF} = 5V, V_{SS} = 0V at Topr = 25°C, f(X_{IN}) = 16MHz)

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 1)			1.5	mA

Note 1: The A/D converter's ladder resistance is not included.
When D/A register contents are not "00$_{16}$", the current I_{VREF} always flows even though V_{REF} may have been set to be unconnected by the A/D control register.

18.1 Timing requirements

(Unless otherwise noted: Vcc = 5V, Vss = 0V at Topr = -40 to 85°C)

Table 18.9 XIN input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(XIN)}$	XIN input cycle time	62.5		ns
$t_{wH(XIN)}$	XIN input HIGH pulse width	30		ns
$t_{wL(XIN)}$	XIN input LOW pulse width	30		ns

Table 18.10 CNTR0 input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CNTR0)}$	CNTR0 input cycle time	100		ns
$t_{wH(CNTR0)}$	CNTR0 input HIGH pulse width	40		ns
$t_{wL(CNTR0)}$	CNTR0 input LOW pulse width	40		ns

Table 18.11 TCIN input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TCIN)}$	TCIN input cycle time	400(Note 1)		ns
$t_{wH(TCIN)}$	TCIN input HIGH pulse width	200(Note 2)		ns
$t_{wL(TCIN)}$	TCIN input LOW pulse width	200(Note 2)		ns

Note 1: Use the greater value, either (1/digital filter clock frequency X 6) or min. value.

Note 2: Use the greater value, either (1/digital filter clock frequency X 3) or min. value.

Table 18.12 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	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 18.13 External interrupt $\overline{INTi}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	250(Note 1)		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	250(Note 2)		ns

Note 1: When the $\overline{INT0}$ input filter select bit selects the digital filter, use the $\overline{INT0}$ input HIGH pulse width to the greater value, either (1/digital filter clock frequency X 3) or min. value.

Note 2: When the $\overline{INT0}$ input filter select bit selects the digital filter, use the $\overline{INT0}$ input LOW pulse width to the greater value, either (1/digital filter clock frequency X 3) or min. value.

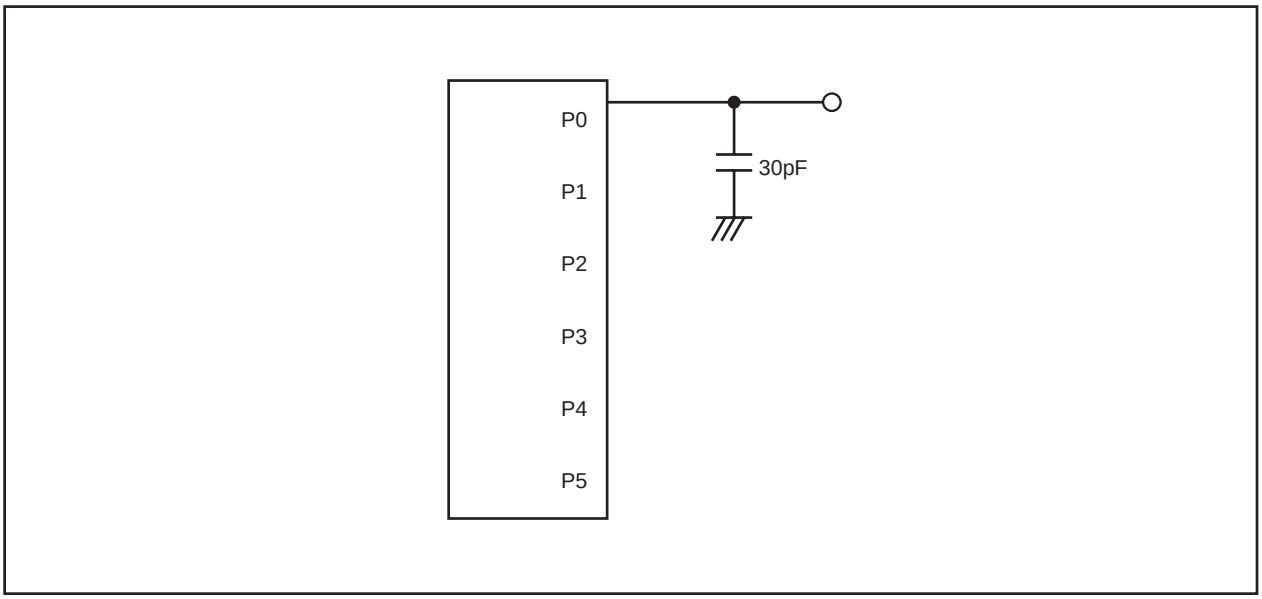


Figure 18.1 Port P0 to P5 measurement circuit

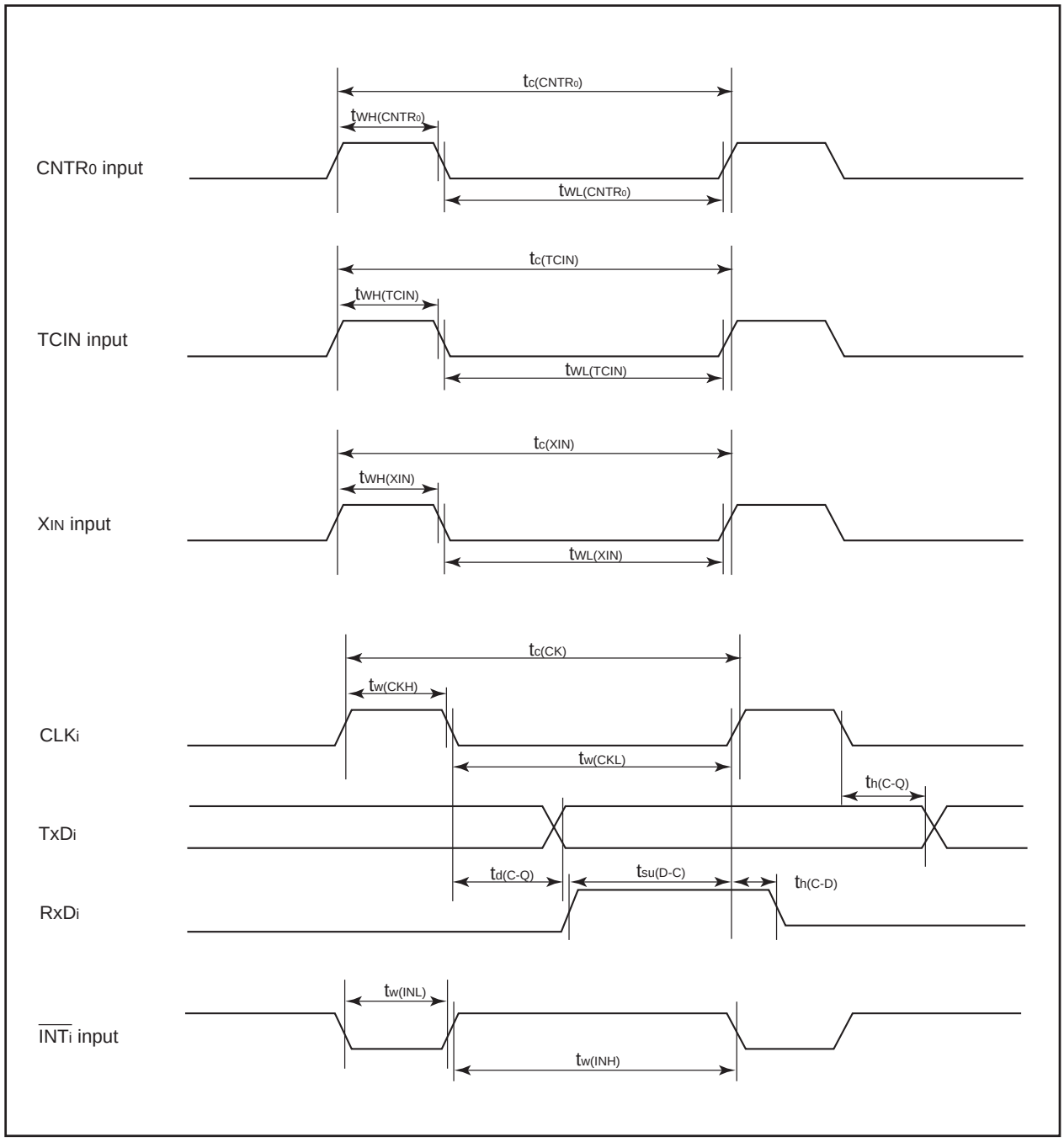


Figure 18.2 Vcc=5V timing diagram

19. Flash Memory Version

19.1 Overview

The flash memory version has four modes—CPU rewrite, standard serial input/output (hereinafter referred to as standard serial I/O), parallel input/output (hereinafter referred to as parallel I/O), and CAN input/output (hereinafter referred to as CAN I/O) modes—in which its internal flash memory can be operated on.

Table 19.1 shows the outline performance of flash memory version and Table 19.2 shows the outline of flash memory rewrite mode. (see **Table 1.1 Performance outline** for the items not listed in Table 19.1).

Table 19.1 Outline performance of flash memory version

Item		Performance
Flash memory operation mode		Four modes (CPU rewrite, parallel I/O, standard serial I/O and CAN I/O)
Erase block division		See **Figure 19.1 Outline performance of flash memory version**
Program method		In units of word, in units of byte (Note 1)
Erase method		Block erase
Program, erase control method		Program and erase controlled by software command
Protect method		Block 0 and 1 are protected by register rewrite (FMR02) Block 0 to 3 are protected by register rewrite enable bit (FMR16)
Number of commands		5 commands
Program, erase count	Block 0 to 3	100 times
	Block A and B (Data area)	100 times
Data retention		10 years
ROM code protect		Parallel I/O, standard serial I/O and CAN I/O modes are supported

Note 1: Can be programmed in byte units in only parallel I/O mode.

Table 19.2 Outline of flash memory rewrite mode

Flash memory rewrite mode	CPU rewrite mode	Parallel I/O mode	Standard serial I/O mode	CAN I/O mode
Function	The user ROM area is rewritten by executing software commands from the CPU. EW0 mode: Can be rewritten in any area other than the flash memory EW1 mode: Can be rewritten in the flash memory	The user ROM area is rewritten by using a dedicated parallel programmer.	The user ROM area is rewritten by using a dedicated serial programmer. Standard serial I/O mode 1: Clock sync. serial I/O Standard serial I/O mode 2 (Note 1): UART	The user ROM area is rewritten by using a dedicated CAN programmer.
Areas which can be rewritten	User ROM area	User ROM area, Boot ROM area	User ROM area	User ROM area
Operation mode	Single chip mode	Parallel I/O mode	Boot mode	Boot mode
ROM programmer	None	Parallel programmer	Serial programmer	CAN programmer

Note 1: When using the standard serial I/O mode 2, make sure a main clock input oscillation frequency is set to 10 or 16 MHz.

19.2 Flash Memory

The ROM in the flash memory version is separated between a user ROM area and a boot ROM area. Figure 19.1 shows the block diagram of flash memory. The user ROM area has 2K-byte block A and B, in addition to the area that stores a program for microcomputer operation during singe-chip mode.

The user ROM area is divided into several blocks. The user ROM area can be rewritten in all of CPU rewrite, standard serial I/O, parallel I/O and CAN I/O modes. Block 0 and 1 can be rewritten by setting FMR0 register's FMR02 bit to "1" and the FMR1 register's FMR16 bit to "1" in CPU rewrite mode only. Block 2 and 3 can be rewriting by setting the FMR 1 register's FMR 16 bit to "1". Block A and B are enabled for use by setting the PM1 register's PM10 bit to "1".

The boot ROM area is reserved area. A rewrite control program for standard serial I/O and CAN I/O modes is written into the boot ROM area when the device is shipped from the factory. The boot ROM area can be rewritten in parallel I/O mode.

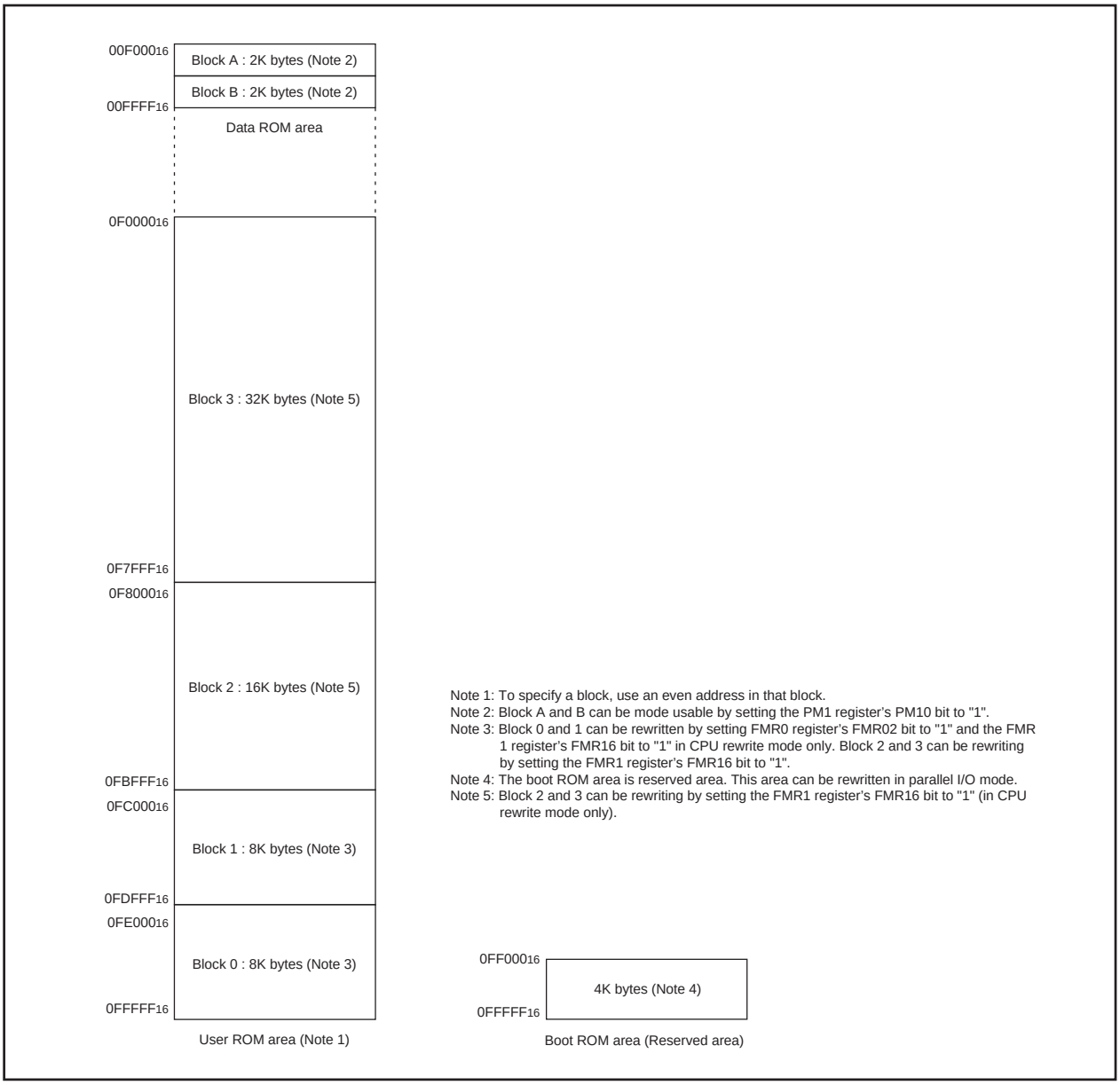


Figure 19.1 Block diagram of flash memory

19.3 Functions to Inhibit Rewriting Flash Memory Version

To prevent the flash memory from being read or rewritten easily, parallel I/O mode has a ROM code protect and standard serial I/O and CAN I/O modes have an ID code check function.

19.3.1 ROM code protect function

The ROM code protect function inhibits the flash memory from being read or rewritten during parallel I/O mode. Figure 19.2 shows the ROMCP register.

The ROMCP register is located in the user ROM area. The ROMCP1 bit consists of two bits. The ROM code protect function is enabled by clearing one or both of two ROMCP1 bits to "0" when the ROMCR bits are not '00_2,' with the flash memory thereby protected against reading or rewriting. Conversely, when the ROMCR bits are '00_2' (ROM code protect removed), the flash memory can be read or rewritten. Once the ROM code protect function is enabled, the ROMCR bits cannot be changed during parallel I/O mode. Therefore, use standard serial I/O or other modes to rewrite the flash memory.

19.3.2 ID code check function

Use this function in standard serial I/O and CAN I/O modes. Unless the flash memory is blank, the ID codes sent from the programmer and the ID codes written in the flash memory are compared to see if they match. If the ID codes do not match, the commands sent from the programmer are not accepted.

The ID code consists of 8-bit data, the areas of which, beginning with the first byte, are $0FFFDF_{16}$, $0FFFE3_{16}$, $0FFFEB_{16}$, $0FFFEF_{16}$, $0FFFF3_{16}$, $0FFFF7_{16}$, and $0FFFFB_{16}$. Prepare a program in which the ID codes are preset at these addresses and write it in the flash memory.

Figure 19.3 shows ID code store addresses.

ROM code protect control address

b7	b6	b5	b4	b3	b2	b1	b0
				1	1	1	1

Symbol: ROMCP
Address: 0FFFFF16
Value when shipped: FF16 (Note 1)

Bit symbol	Bit name	Function	RW
$\overline{\text{(b3-b0)}}$	Reserved bit	Set this bit to "1"	RW
ROMCR	ROM code protect reset bit (Note 1, 2)	b5 b4 0 0 : Removes protect 0 1 : 1 0 : } Enables ROMCP1 bit 1 1 :	RW RW
ROMCP1	ROM code protect level 1 set bit (Note 1, 3, 4)	b7 b6 0 0 : 0 1 : } Protect enabled 1 0 : 1 1 : Protect disabled	RW RW

Note 1: Once any of these bits is cleared to "0", it cannot be set back to "1". If a memory block that contains the ROMCP register is erased, the ROMCP register is set to 'FF16'.
Note 2: If the ROMCR bits are set to '002' when the ROMCR bits are other than '002' and the ROMCP1 bits are other than '112', ROM code protect level 1 is removed. However, because the ROMCR bits cannot be modified during parallel I/O mode, they need to be modified in standard serial I/O or other modes.
Note 3: If the ROMCR bits are set to other than '002' and the ROMCP1 bits are set to other than '112' (ROM code protect enabled), the flash memory is disabled against reading and rewriting in parallel I/O mode.
Note 4: The ROMCP1 bits are effective when the ROMCR bits are '012', '102', or '112'.

Figure 19.2 ROMCP register

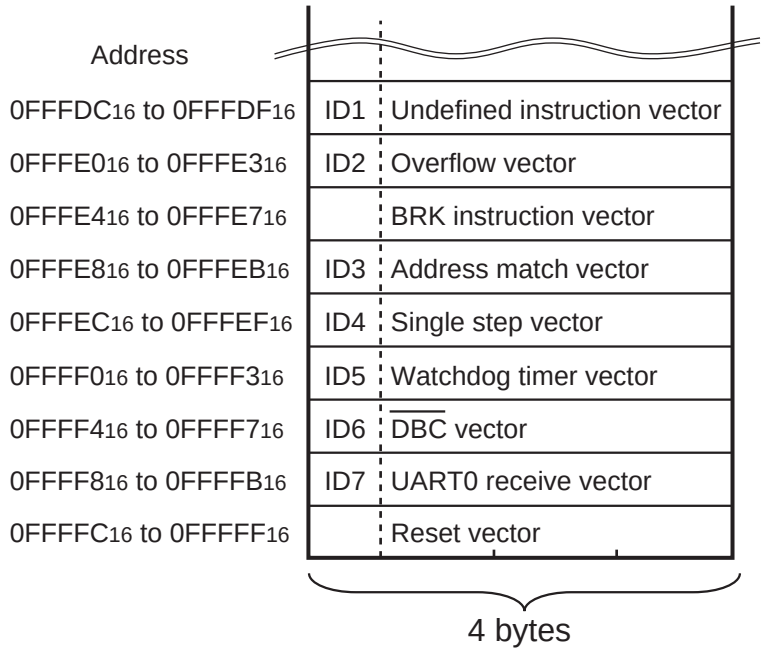


Figure 19.3 ID code store addresses

19.4 Boot Mode

When the microcomputer is reset by applying a high-level signal to the CNVSS and $\overline{CE}$ pins, it is placed in boot mode, thereby executing the program in the boot ROM area.

During boot mode, the boot ROM and user ROM areas are switched over by the FMR0 register's FMR05 bit.

19.5 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten by executing software commands from the CPU. Therefore, the user ROM area can be rewritten directly while the microcomputer is mounted on-board without having to use a ROM programmer, etc.

Make sure the program and the block erase commands are executed only on each block in the user ROM area.

When generating an interrupt request during erasure operation in CPU mode, the M16C/1N Group flash memory can offer the erasure-suspend feature which allows erasure operation to be suspended and to process the interrupt. User ROM area can be read in a program during erasure-suspend.

During CPU rewrite mode, the user ROM area be operated on in either Erase Write 0 (EW0) mode or Erase Write 1 (EW1) mode. Table 19.3 lists the differences between Erase Write 0 (EW0) and Erase Write 1 (EW1) modes.

Table 19.3 Differences between EW0 mode and EW1 mode

Item	EW0 mode	EW1 mode
Operation mode	• Single chip mode	Single chip mode
Areas in which a rewrite control program can be located	• User ROM area	User ROM area
Areas in which a rewrite control program can be executed	Must be transferred to any area other than the flash memory (e.g., RAM) before being executed	Can be executed directly in the user ROM area
Areas which can be rewritten	User ROM area (Note 1)	User ROM area (Note 1) However, this does not include the area in which a rewrite control program exists
Software command limitations	None	• Program, Block Erase command Cannot be executed on any block in which a rewrite control program exists • Read Status Register command Cannot be executed
Modes after Program or Erase	Read Status Register mode	Read Array mode
CPU status during Auto Write and Auto Erase	Operating	Hold state (I/O ports retain the state in which they were before the command was executed)(Note 2)
Flash memory status detection	• Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program • Execute the Read Status Register command to read the status register's SR7, SR5, and SR4 flags.	Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program
The shift conditions to erasure-suspend (Note 3)	Set the FMR4 register's FMR40 and RMR41 bits to "1" by program.	The FMR register's FMR40 bit is "1" and generated the interrupt request of enabled interrupt.

Note 1: Can be rewritten block 0 and 1 when setting the FMR0 register's FMR02 bit to "1" and the FMR1 register's FMR16 bit to "1". Block 2 and 3 can be rewriting by setting the FMR1 register's FMR16 bit to "1".

Note 2: Make sure no interrupts will occur.

Note 3: The conditions are met and it takes a maximum of td(SR-ES) time until a flash memory can be read after shifting to erasure-suspend.

19.5.1 EW0 mode

The microcomputer is placed in CPU rewrite mode by setting the FMR0 register's FMR01 bit to "1" (CPU rewrite mode enabled), ready to accept commands. In this case, because the FMR1 register's FMR11 bit = 0, EW0 mode is selected. The FMR01 bit can be set to "1" by writing "0" and then "1" in succession.

Use software commands to control program and erase operations. Read the FMR0 register or status register to check the status of program or erase operation at completion.

When shifting to erasure-suspend during auto erasing, set the FMR40 bit to "1" (Suspend enable) and the FMR41 bit to "1" (Suspend request).

After waiting for td (SR-ES) time, access user ROM area after confirming that the FMR46bit has been set to "1" (Erase inactive).

Setting the FMR41 bit to "0" (Erase restart), the erasure operation is resumed.

19.5.2 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).

Read the FMR0 register to check the status of program or erase operation at completion. The status register cannot be read during EW1 mode.

When enabling the erasure-suspend feature, execute the block erase command after setting the FMR40 bit to "1" (Suspend enable).

In addition, the interrupt for shifting to erasure suspend has to have been enabled beforehand.

When shifting to erasure-suspend after td (SR-ES) time from interrupt request, the interrupt request is accepted.

When the interrupt request occurs, the FMR41bit is set to "1" (Suspend request) automatically and erasure operation is suspended.

After processing the interrupt, when the FM00 bit is "0" (Busy (being erased)), set the FMR41 bit to "0" and re-execute the block erase command.

Figure 19.4 shows the FMR0 register and Figure 19.5 shows the FMR1 and FMR4 registers.

(1) FMR00 bit

This bit indicates the operating status of the flash memory. The bit is "0" when the program or erase program is running; otherwise, the bit is "1".

(2) FMR01 bit

The microcomputer is made ready to accept commands by setting the FMR01 bit to "1" (CPU rewrite mode). During boot mode, make sure the FMR05 bit also is "1" (user ROM area access).

(3) FMR02 bit

Block 0 and 1 do not accept the program or erase command while the FMR02 bit is set to "0" (inhibit rewriting).

(4) FMSTP bit

This bit is provided for initializing the flash memory control circuits, as well as for reducing the amount of current consumed in the flash memory. The internal flash memory cannot be accessed by setting the FMSTP bit to "1". Therefore, the FMSTP bit must be written to by a program in a memory area other than the flash memory.

In the following cases, set the FMSTP bit to "1":

- When flash memory access resulted in an error while erasing or programming in EW0 mode (FMR00 bit not reset to "1" (ready))
- When entering low power mode or on-chip oscillator low power mode

Figure 19.7 shows a flow chart to be followed before and after entering low power mode.

Note that when going to stop or wait mode, the FMR0 register does not need to be set because the power for the internal flash memory is automatically turned off and is turned back on again after returning from stop or wait mode.

(5) FMR05 bit

This bit switches between the boot ROM and user ROM areas during boot mode. Set this bit to "0" when accessing the boot ROM area (for read) or "1" (user ROM access) when accessing the user ROM area (for read, write or erase).

(6) FMR06 bit

This is a read-only bit indicating the status of auto program operation. The bit is set to "1" when a program error occurs; otherwise, it is cleared to "0". For details, refer to the description of **19.5.6 Full Status Check**.

(7) FMR07 bit

This is a read-only bit indicating the status of auto erase operation. The bit is set to "1" when an erase error occurs; otherwise, it is cleared to "0". For details, refer to the description of **19.5.6 Full Status Check**.

(8) FMR11 bit

Setting this bit to "1" places the microcomputer in EW1 mode. This bit is relevant if the FMR01 bit is set.

(9) FMR16 bit

When the FMR16bit is "0" (Rewrite disable), the block 0-3 don't accept the program and the block erase commands. This bit is relevant if the FMR01 bit is set.

(10) FMR 40bit

When setting the FMR40 bit to "1", the erasure-suspend feature is enabled.

(11) FMR 41bit

In EW0 mode, when setting the FMR41 bit to "1" by software during auto erasing, the microcomputer shifts to the erasure-suspend mode.

In EW1 mode, when occurring the enabled interrupt request, the FMR 41bit changes to "1" (Suspend-request) automatically and the microcomputer shifts to the erasure-suspend mode.

Setting the FMR41 bit to "0" (Erase restart), auto erasing is resumed.

(12) FMR46bit

The FMR46 bit is "0" during auto erasing and is "1" during the erasure-suspend mode.

The internal flash memory is banned to access while the FMR41 bit is "0".

Flash memory control register 0

b7	b6	b5	b4	b3	b2	b1	b0
		0	0				

Symbol	Address	After reset
FMR0	$01B7_{16}$	$XX000001_2$

Bit symbol	Bit name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0 : Busy (being written or erased) 1 : Ready	RO
FMR01	CPU rewrite mode select bit (Note 1)	0 : Disables CPU rewrite mode 1 : Enables CPU rewrite mode	RW
FMR02	Block 0, 1 rewrite-enable bit (Note 2)	0 : Disables rewriting 1 : Enables rewriting	RW
FMSTP	Flash memory stop bit (Note 3, 4)	0 : Enables flash memory operation 1 : Stops flash memory operation (placed in low power mode, flash memory initialized)	RW
$\overline{(b4)}$	Reserved bit	Set to "0"	RW
FMR05	User ROM area select bit (Effective in only boot mode) (Note 3)	0 : Boot ROM area is accessed 1 : User ROM area is accessed	RW
FMR06	Program status flag (Note 5)	0 : Terminated normally 1 : Terminated in error	RO
FMR07	Erase status flag (Note 5)	0 : Terminated normally 1 : Terminated in error	RO

Note 1: When setting this bit to "1", write a "0" and then a "1" in to it in succession. Make sure no interrupts will occur before completion of these two write operations.
In EW0 mode, write this bit by a program placed to an area other than internal flash memory.
Set this bit to "0" after placing in read array mode.

Note 2: When setting this bit to "1", write a "0" and then a "1" in to it in succession while the FMR01 bit is "1". Make sure no interrupts will occur before completion of these two write operations.

Note 3: Write this bit by a program placed to an area other than internal flash memory.

Note 4: This bit is valid when the FMR01 bit is "1" (CPU rewrite mode). When the FMR01 bit is "0", although the FMSTP bit can be set to "1" by writing "1" in a program, the flash memory is not initialized.

Note 5: This bit is cleared to "0" by executing the clear status command.

Figure 19.4 FMR0 Register

Flash memory control register 1

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

Symbol: FMR1 — Address: $01B5_{16}$ — After reset: $0000XX0X_2$

Bit symbol	Bit name	Function	RW
$\overline{(b0)}$	Reserved bit	The value in this bit when read is indeterminate.	RO
FMR11	EW1 mode select bit (Note 1)	0 : EW0 mode 1 : EW1 mode	RW
$\overline{(b3\text{-}b2)}$	Reserved bit	The value in this bit when read is indeterminate.	RO
$\overline{(b5\text{-}b4)}$	Reserved bit	Set to "0"	RW
FMR16	Block 0-3 rewrite enable bit (Note 2)	0 : Disables rewriting 1 : Enables rewriting	RW
$\overline{(b7)}$	Reserved bit	Set to "0"	RW

Note 1: When setting this bit to "1", write a "0" and then a "1" in succession while the FMR01 bit is "1". Make sure no interrupts will occur before completion of these two write operations.
The FMR01 and FMR11 bits both are cleared to "0" by setting the FMR01 bit to "0".
Note 2: When setting this bit to "1", write a "0" and then a "1" to it in succession while the FMR01 bit is "1". Make sure no interrupts will occur before completion of these two write operations.

Flash memory control register 4

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

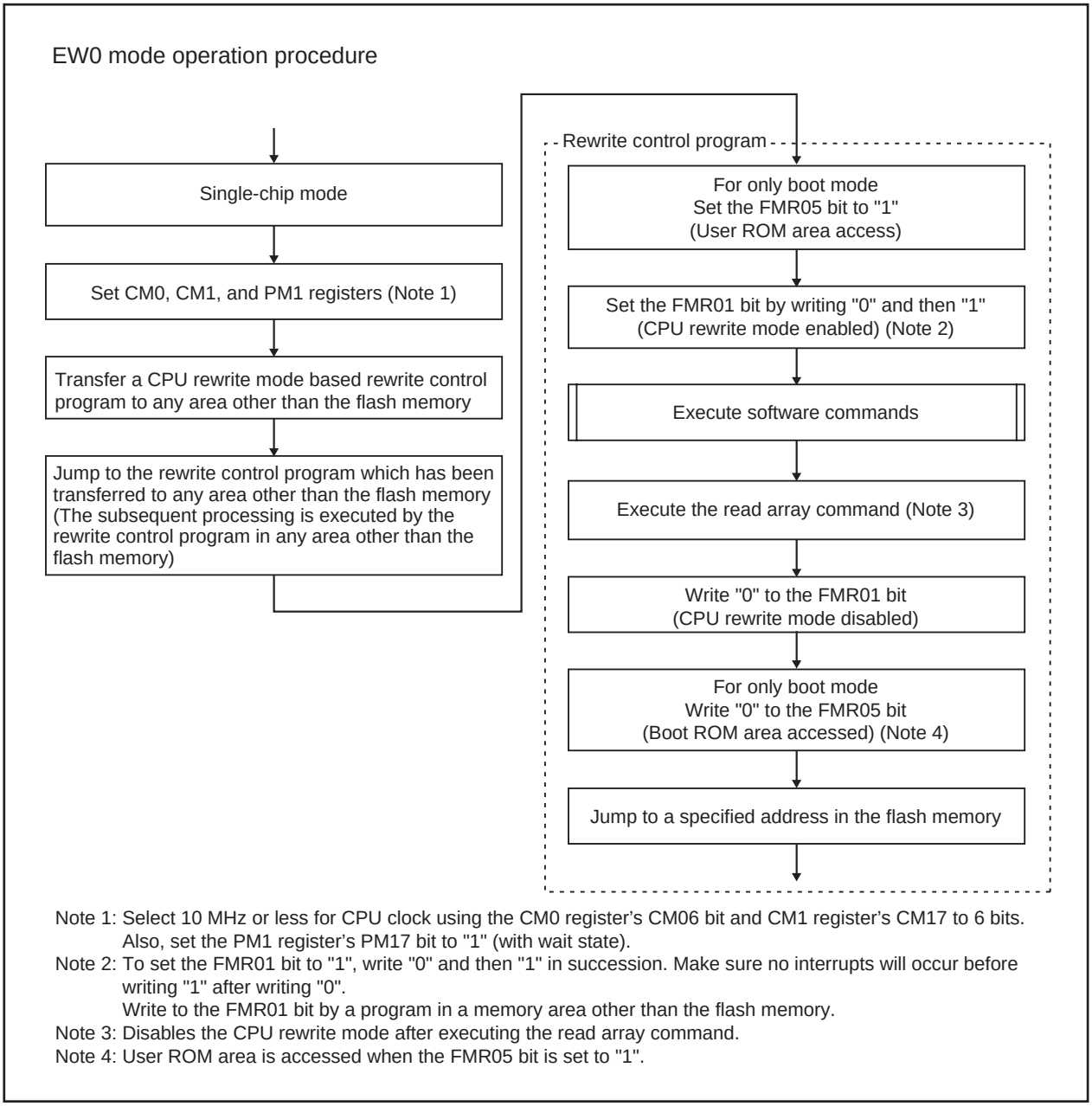
Symbol: FMR4 — Address: $01B3_{16}$ — After reset: 01000000_2

Bit symbol	Bit name	Function	RW
FMR40	Suspend enable bit (Note 1)	0 : Invalid 1 : Valid	RW
FMR41	Suspend request bit (Note 2)	0 : Erase restart 1 : Suspend request	RW
$\overline{(b5\text{-}b2)}$	Reserved bits	Set to "0"	RO
FMR46	Suspend status	0 : Erase active 1 : Erase inactive (erasure-suspend)	RO
$\overline{(b7)}$	Reserved bit	Set to "0"	RW

Note 1: When setting this bit to "1", write a "0" and then a "1" to it in succession. Make sure no interrupts will occur before completion of these two write operations.
Note 2: This bit is valid only when the FMR40 bit is "1" and can be written in only the period from issuing an erase command until completion of erasing.
- In EW0 mode, this bit can be set to "0" or "1" by program.
- In EW1 mode, this bit is automatically set to "1" when a maskable interrupt occurs during erasure execution while the FMR40 bit is "1". It can not be set to "1" by program. (Writing "0" is available.)

Figure 19.5 FMR1 Register, FMR4 Register

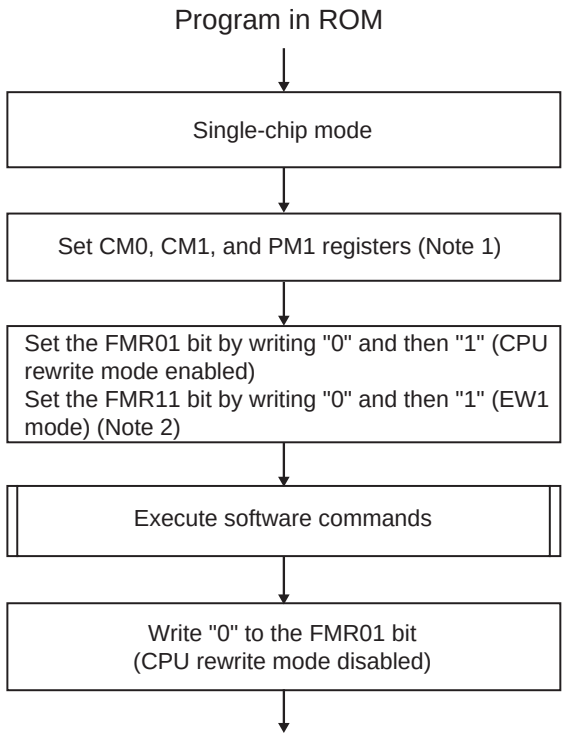
Figure 19.6 and 19.7 show the setting and resetting of EW0 mode and EW1 mode, respectively. Figure 19.8 shows the processing before and after low power dissipation mode.



Note 1: Select 10 MHz or less for CPU clock using the CM0 register's CM06 bit and CM1 register's CM17 to 6 bits. Also, set the PM1 register's PM17 bit to "1" (with wait state).
Note 2: To set the FMR01 bit to "1", write "0" and then "1" in succession. Make sure no interrupts will occur before writing "1" after writing "0".
Write to the FMR01 bit by a program in a memory area other than the flash memory.
Note 3: Disables the CPU rewrite mode after executing the read array command.
Note 4: User ROM area is accessed when the FMR05 bit is set to "1".

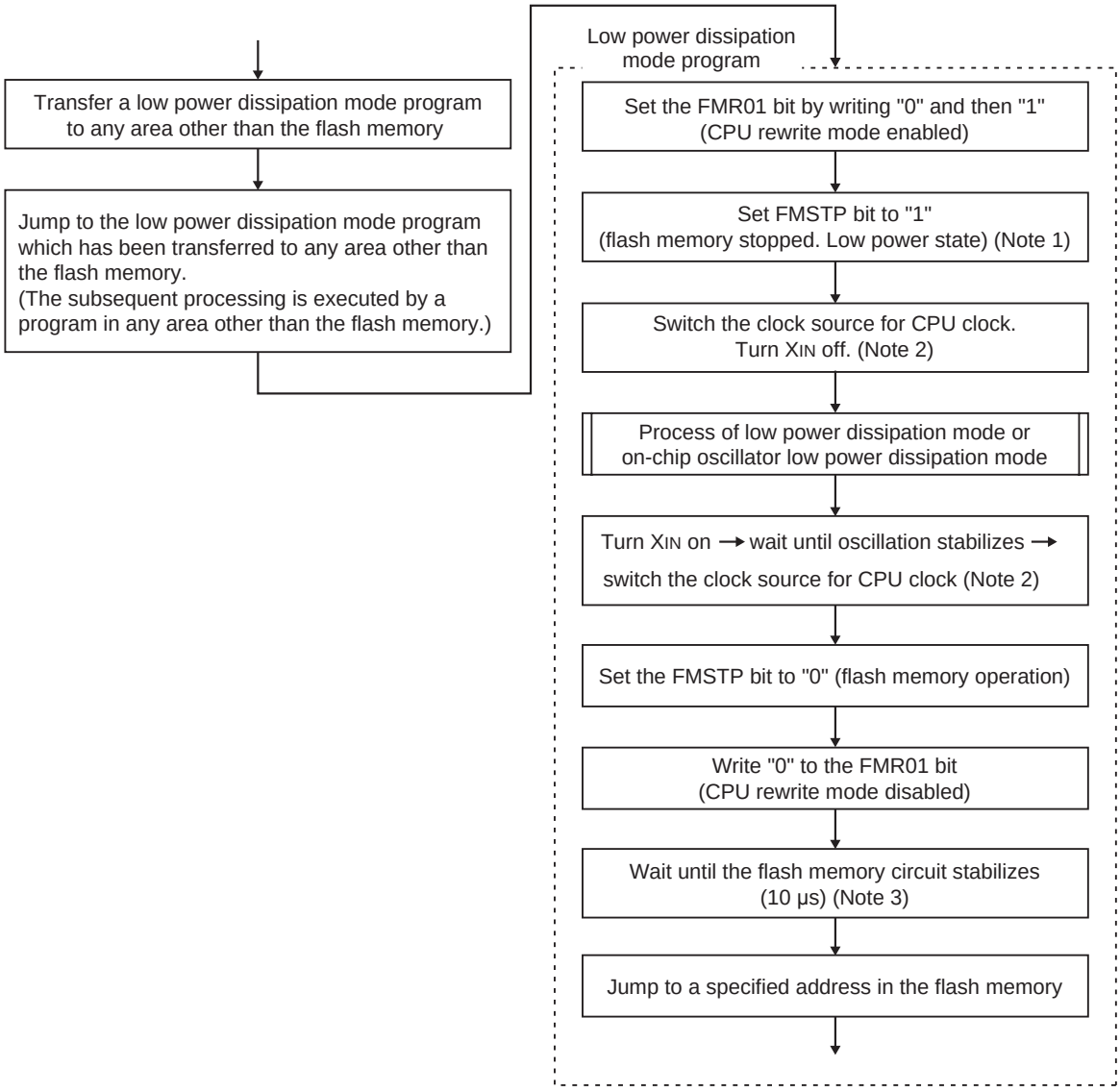
Figure 19.6 Setting and resetting of EW0 mode

EW1 mode operation procedure



Note 1: Select 10 MHz or less for CPU clock using the CM0 register's CM06 bit and CM1 register's CM17 to 6 bits. Also, set the PM1 register's PM17 bit to "1" (with wait state).
Note 2: To set the FMR01 bit to "1", write "0" and then "1" in succession. Make sure no interrupts will occur before writing "1" after writing "0".

Figure 19.7 Setting and resetting of EW1 mode



Note 1: Set the FMSTP bit to "1" after setting the FMR01 bit to "1".
Note 2: Before the clock source for CPU clock can be changed, the clock to which to be changed must be stable.
Note 3: Insert a 10 µs wait time in a program. The flash memory cannot be accessed during this wait time.

Figure 19.8 Processing before and after low power dissipation mode

19.5.3 Precautions on CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

(1) Operation speed

Before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or less for BCLK using the CM0 register's CM06 bit and CM1 register's CM17–6 bits. Also, set the PM1 register's PM17 bit to "1" (with wait state).

(2) Instructions inhibited against use

In EW0 mode, the following instructions cannot be used because the flash memory's internal data is referenced: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

(3) Interrupts

EW0 mode

- Any interrupt which has a vector in the variable vector table can be used providing that its vector is transferred into the RAM area.
- The watchdog timer interrupt can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. However, it is necessary that the jump addresses for those interrupts are set in the fixed vector table, and that interrupt service routines are available for those interrupts.

  Because the rewrite operation is halted when a watchdog timer interrupt occurs, the FMR01 bit must be set back to "1" again in order to enable erase or programming operation after exiting the interrupt service routine.
- The address match interrupt cannot be used because the flash memory's internal data is referenced.

EW1 mode

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program or auto erase period.

(4) How to access

To set the FMR01, FMR02, or FMR11 bit to "1", write "0" and then "1" in succession. This is necessary to ensure that no interrupts will occur before writing "1" after writing "0".

(5) Writing in the user ROM area

If the power supply voltage drops while rewriting in EW0 mode any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. It is recommended that such a block be rewritten using standard serial I/O, CAN I/O or parallel I/O mode.

(6) Writing command and data

Write the command code and data at even addresses.

(7) Wait mode

When shifting to wait mode, set the FMR01 bit to "0" (CPU rewrite mode disabled) before executing the WAIT instruction.

(8) Stop mode

When shifting to stop mode, the following settings are required:

- Set the FMR01 bit to "0" (CPU rewrite mode disabled) and setting the CM10 bit to "1" (stop mode).
- 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 returning from stop mode
```

(9) Low power dissipation mode, on-chip oscillator low power dissipation mode

If the CM05 bit is set to "1" (main clock stop), the following commands must not be executed.

- Program
- Block erase

19.5.4 Software Commands

Software commands are described below. The command code and data must be read and written in 16 bit units, to and from even addresses in the user ROM area. When writing command code, the 8 high-order bits (D_{15}-D_8) are ignored.

Table 19.4 shows the list of software commands.

Table 19.4 List of software commands

Software Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_{15}-D_0)	Mode	Address	Data (D_{15}-D_0)
Read array	Write	X	xxFF16			
Read status register	Write	X	xx7016	Read	X	SRD
Clear status register	Write	X	xx5016			
Program	Write	WA	xx4016	Write	WA	WD
Block erase	Write	X	xx2016	Write	BA	xxD016

SRD: Status register data (D_7-D_0)
WA: Write address (even address, however)
WD: Write data (16 bits)
BA: Uppermost block address (even address, however)
X: Any even address in the user ROM area
x: High-order 8 bits of command code (ignored)

(1) Read array

This command reads the flash memory.

Writing 'xxFF16' in the first bus cycle places the microcomputer in read array mode. Enter the read address in the next or subsequent bus cycles, and the content of the specified address can be read in 16 bit units.

Because the microcomputer remains in read array mode until another command is written, the contents of multiple addresses can be read in succession.

(2) Read status register

This command reads the status register.

Write 'xx7016' in the first bus cycle, and the status register can be read in the second bus cycle (refer to **19.5.5 Status Register**). When reading the status register too, specify an even address in the user ROM area.

Do not execute this command in EW1 mode.

(3) Clear status register

This command clears the status register to "0".

Write 'xx5016' in the first bus cycle, and the FMR0 register's FMR06 to FMR07 bits and the status register's SR4 to SR5 will be cleared to "0".

(4) Program

This command writes data to the flash memory in 1 word (2 byte) units.

Write 'xx40$_{16}$' in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

Check the FMR0 register's FMR00 bit to see if auto programming has finished. The FMR00 bit is "0" during auto programming and set to "1" when auto programming is completed.

Check the FMR0 register's FMR06 bit after auto programming has finished, and the result of auto programming can be known (refer to **19.5.6 Full Status Check**).

Figure 19.9 shows an example of program flowchart.

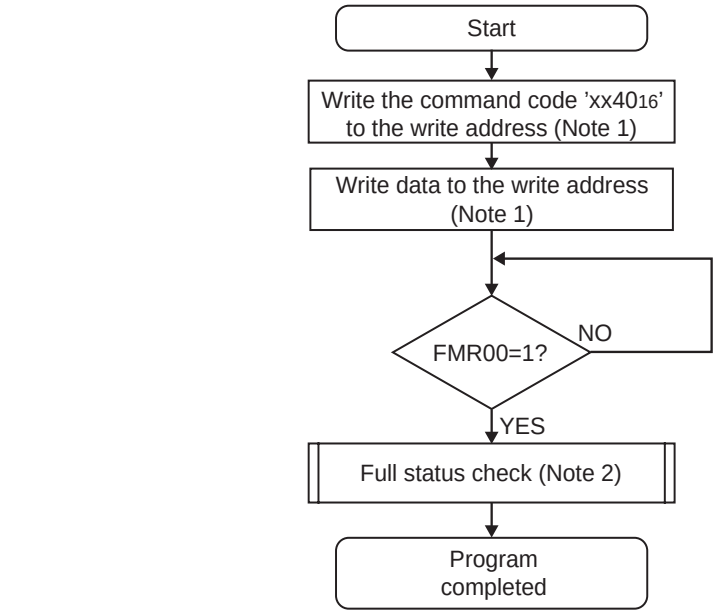
Writing over already programmed addresses is inhibited.

Also, block 0 to 3 do not accept the program command while the FMR 1 register's FMR 16 bit is "0" and the FMR0 register's FMR02 bit is "0" (Inhibit rewriting.)

To execute another command immediately after the program command, use the same write address that was specified in the second bus cycle of the program command for the address value to be specified in the first bus cycle of the next command.

In EW1 mode, do not execute this command on any address at which the rewrite control program is located.

In EW0 mode, the microcomputer goes to read status register mode at the same time auto programming starts, making it possible to read the status register. The status register bit 7 (SR7) is cleared to "0" at the same time auto programming starts, and set back to "1" when auto programming finishes. In this case, the microcomputer remains in read status register mode until a read command is written next. The result of auto programming can be known by reading the status register after auto programming has finished.

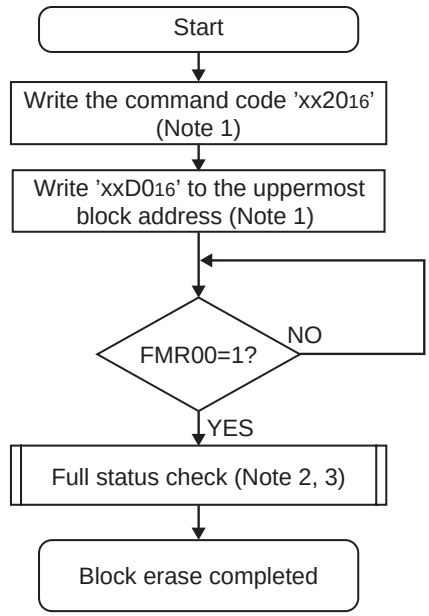


Note 1: Write the command code and data at even number.
Note 2: See **Figure 19.12 Full status check flowchart, handling each error generated**.

Figure 19.9 Program flowchart

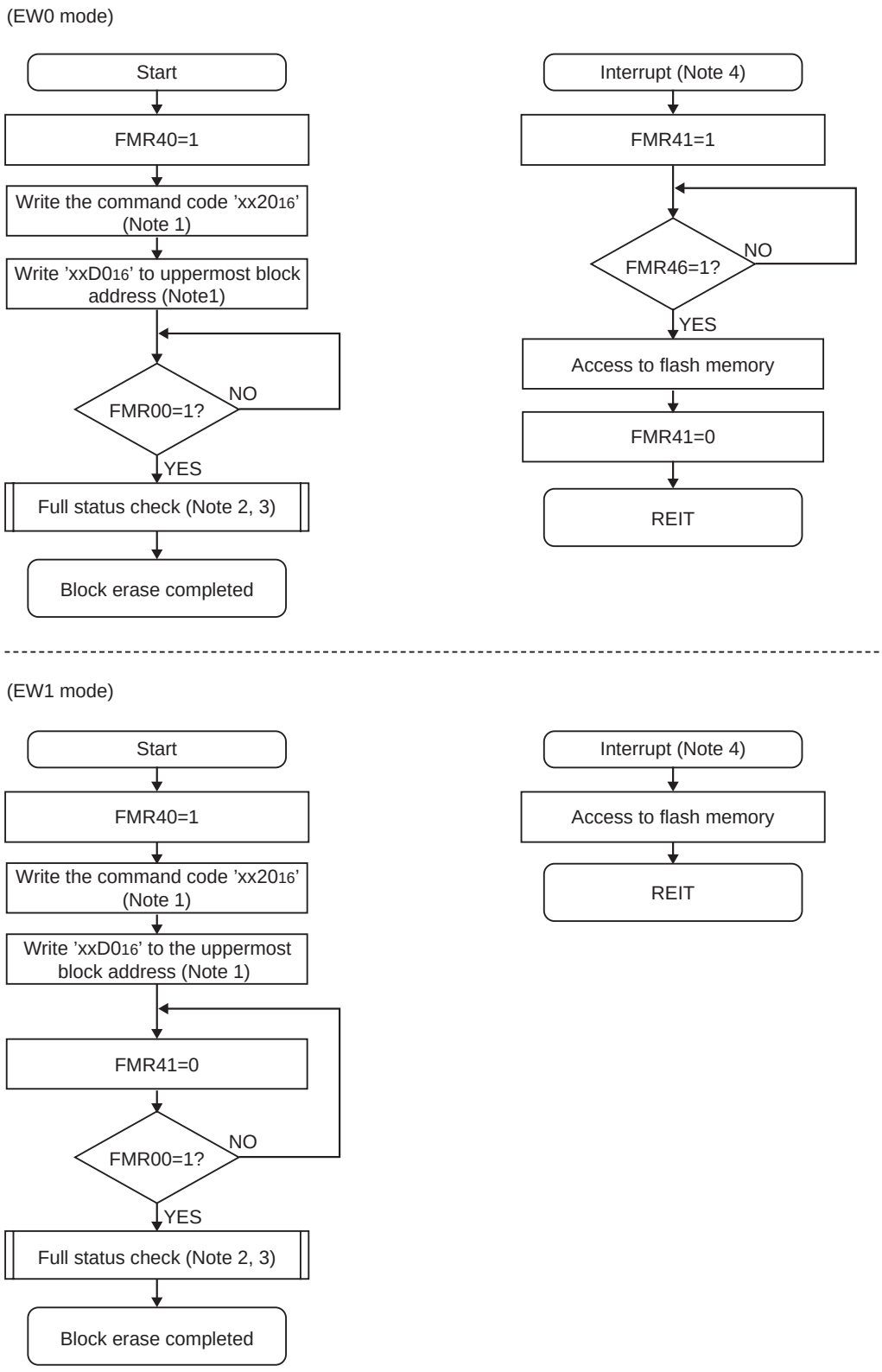
(5) Block erase

Write 'xx20$_{16}$' in the first bus cycle and write 'xxD0$_{16}$' to the uppermost address of a block (even address, however) in the second bus cycle, and an auto erase operation (erase and verify) will start.
Check the FMR0 register's FMR00 bit to see if auto erasing has finished.
The FMR00 bit is "0" during auto erasing and set to "1" when auto erasing is completed.
In EW0 mode, when using the erasure suspend feature, confirm the FMR4 register's FMR46 bit whether it has shifted to erasure suspend.
The FMR46 bit is "0" during auto erasing and is set to "1" when auto erasing is suspended (shift to erasure suspend).
Check the FMR0 register's FMR07 bit after auto erasing has finished, and the result of auto erasing can be known (refer to **19.5.6 Full Status Check**).
Also, block 0 and 1 do not accept the block erase command while the FMR0 register's FMR02 bit is set to "0" (Inhibit rewriting.)
Figure 19.10 shows an example of a block erase flowchart when not using the erasure suspend feature and Figure 19.11 shows an example of a block erase flowchart when using the erasure suspend feature.
In EW1 mode, do not execute this command on any address at which the rewrite control program is located.
In EW0 mode, the microcomputer goes to read status register mode at the same time auto erasing starts, making it possible to read the status register. The status register bit 7 (SR7) is cleared to "0" at the same time auto erasing starts, and set back to "1" when auto erasing finishes. In this case, the microcomputer remains in read status register mode until the read array command is written next. In addition, when the erase error occurred, repeat the operation executing the clear status register command and then the block erase command in succession at least 3 times until the error is eliminated.



Note 1: Write the command code and data at even number.
Note 2: See **Figure 19.12 Full status check flowchart, handling each error generated**.
Note 3: When the erase error occurred, repeat the operation executing the clear status register command and then the block erase command in succession at least 3 times until the error is eliminated.

Figure 19.10 Block erase flowchart (when not using the erasure suspend feature)



Note 1: Write the command code and data at even number.
Note 2: See **Figure 19.12 Full status check flowchart, handling each error generated**.
Note 3: When the erase error occurred, repeat the operation executing the clear status register command and then the block erase command in succession at least 3 times until the error is eliminated.
Note 4: In EW0 mode, allocate the interrupt vector table of used interrupt to the internal RAM.

Figure 19.11 Block erase flowchart (when using the erasure suspend feature)

19.5.5 Status Register

The status register indicates the operating status of the flash memory and whether an erase or programming operation terminated normally or in error. The status of the status register can be known by reading the FMR0 register's FMR00, FMR06, and FMR07 bits.

Table 19.5 shows the status register.

In EW0 mode, the status register can be read in the following cases:

- When a given even address in the user ROM area is read after writing the Read Status Register command
- When a given even address in the user ROM area is read after executing the program, or block erase command but before executing the read array command.

(1) Sequencer status (SR7 and FMR00 bits)

The sequence status indicates the operating status of the flash memory. SR7 = 0 (busy) during auto programming and auto erase is set to "1" (ready) at the same time the operation finishes.

(2) Erase status (SR5 and FMR07 bits)

Refer to **19.5.6 Full status check**.

(3) Program status (SR4 and FMR06 bits)

Refer to **19.5.6 Full status check**.

Table 19.5 Status register

Status register bit	FMR0 register bit	Status name	Contents		Value after reset
			"0"	"1"	
SR7 (D_7)	FMR00	Sequencer status	Busy	Ready	1
SR6 (D_6)	——	Reserved	-	-	——
SR5 (D_5)	FMR07	Erase status	Terminated normally	Terminated in error	0
SR4 (D_4)	FMR06	Program status	Terminated normally	Terminated in error	0
SR3 (D_3)	——	Reserved	-	-	——
SR2 (D_2)	——	Reserved	-	-	——
SR1 (D_1)	——	Reserved	-	-	——
SR0 (D_0)	——	Reserved	-	-	——

- The FMR07 bit (SR5) and FMR06 bit (SR4) are cleared to "0" by executing the clear status register command.
- When the FMR07 bit (SR5) or FMR06 bit (SR4) = 1, the program, block and erase commands are not accepted.
- D_0-D_7: Indicates the data bus which is read out when the read status register command is executed.

19.5.6 Full Status Check

When an error occurs, the FMR0 register's FMR06 to FMR07 bits are set to "1", indicating occurrence of each specific error. Therefore, execution results can be verified by checking these status bits (full status check).

Table 19.6 lists errors and FMR0 register status. Figure 19.12 shows a full status check flowchart and the action to be taken when each error occurs.

Table 19.6 Errors and FMR0 register status

FRM00 register (status register) status		Error	Error occurrence condition
FMR07 (SR5)	FMR06 (SR4)		
1	1	Command sequence error	• When any command is not written correctly • When invalid data was written other than those that can be written in the second bus cycle of the block erase command (i.e., other than 'xxD0$_{16}$' or 'xxFF$_{16}$') (Note 1)
1	0	Erase error	• When the block erase command was executed on locked blocks but the blocks were not automatically erased correctly.
0	1	Program error	• When the program command was executed on unlocked blocks but the blocks were not automatically programmed correctly.

Note 1: Writing 'xxFF$_{16}$' in the first bus cycle places the microcomputer in read array mode. Simultaneously, the command code written in the first cycle becomes invalid.

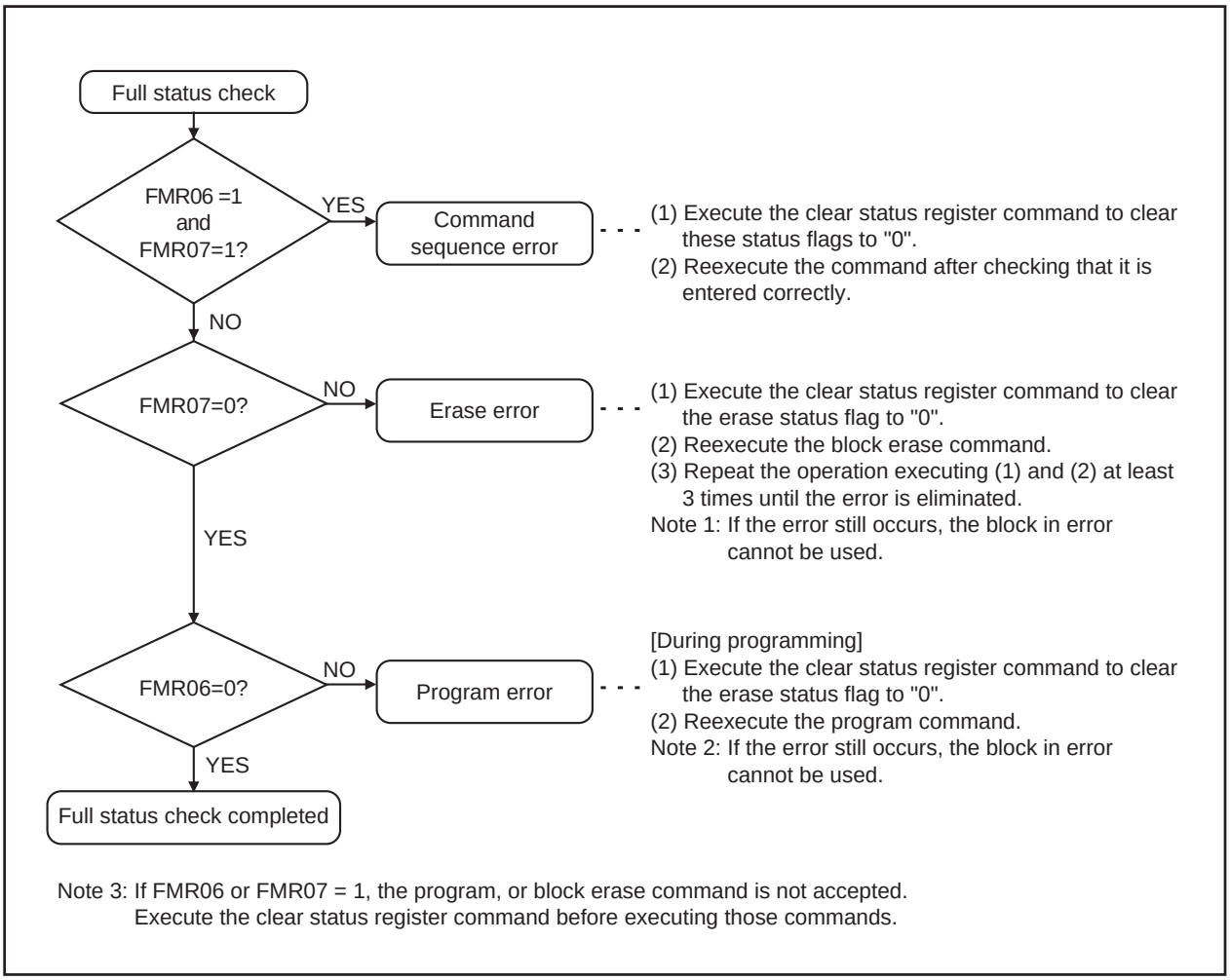


Figure 19.12 Full status check flowchart, handling each error generated

19.6 Parallel Input/Output Mode

In parallel I/O mode, the user ROM area can be rewritten by using a parallel programmer suitable for the M16C/1N group. For more information about parallel programmers, contact the manufacturer of your parallel programmer. For details on how to use, refer to the user's manual included with your parallel programmer.

19.6.1 ROM code protect function

The ROM code protect function inhibits the flash memory from being read or rewritten. (refer to the description of **19.3 Functions to Inhibit Rewriting Flash Memory Version**).

19.7 Standard Serial Input/Output Mode

In standard serial I/O mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer suitable for the M16C/1N group. For more information about serial programmers, contact the manufacturer of your serial programmer. For details on how to use, refer to the user's manual included with your serial programmer.

There are actually two standard serial I/O modes: mode 1, which is clock synchronized, and mode 2, which is asynchronized.

Table 19.7 lists pin functions (flash memory standard serial I/O mode). Figure 19.13 shows pin connections for standard serial I/O mode.

19.7.1 ID code check function

This function determines whether the ID codes sent from the serial programmer and those written in the flash memory match (refer to the description of **19.3 Functions to Inhibit Rewriting Flash Memory Version**).

Table 19.7 Pin functions (Flash memory standard serial I/O mode)

Pin	Name	I/O	Description
VCC, VSS	Power input	I	Apply the voltage guaranteed for program and erase to Vcc pin and 0 V to Vss pin.
IVCC	IVCC input	I	Connect a capacitor (0.1µF) to VSS pin.
CNVSS	CNVSS input	I	Connect to VCC pin
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is "L" level, input a 20 cycle 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	
VREF	Reference voltage input	I	Enter the reference voltage for AD from this pin. Connect to VCC or VSS pin.
P00 to P07	Input port P0	I	Input "H" or "L" level signal or open.
P10 to P13	Input port P1	I	Input "H" or "L" level signal or open.
P14	TxD output	O	Serial data output pin (Note 1)
P15	RxD input	I	Serial data input pin
P16	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin. Standard serial I/O mode 2: Input "L" level signal.
P17	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.
P20, P21	Input port P2	I	Input "H" or "L" level signal or open.
P30	SEL input	I	SEL signal input pin. Input "L" level signal.
P31	CE input	I	CE signal input pin. Input "H" level signal.
P32 to P37	Input port P3	I	Input "H" or "L" level signal or open.
P40 to P47	Input port P4	I	Input "H" or "L" level signal or open.
P50 to P52	Input port P5	I	Input "H" or "L" level signal or open.

Note 1: When using standard serial I/O mode 1, the TxD pin must be held high while the $\overline{\text{RESET}}$ pin is low. Therefore, connect this pin to VCC 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.

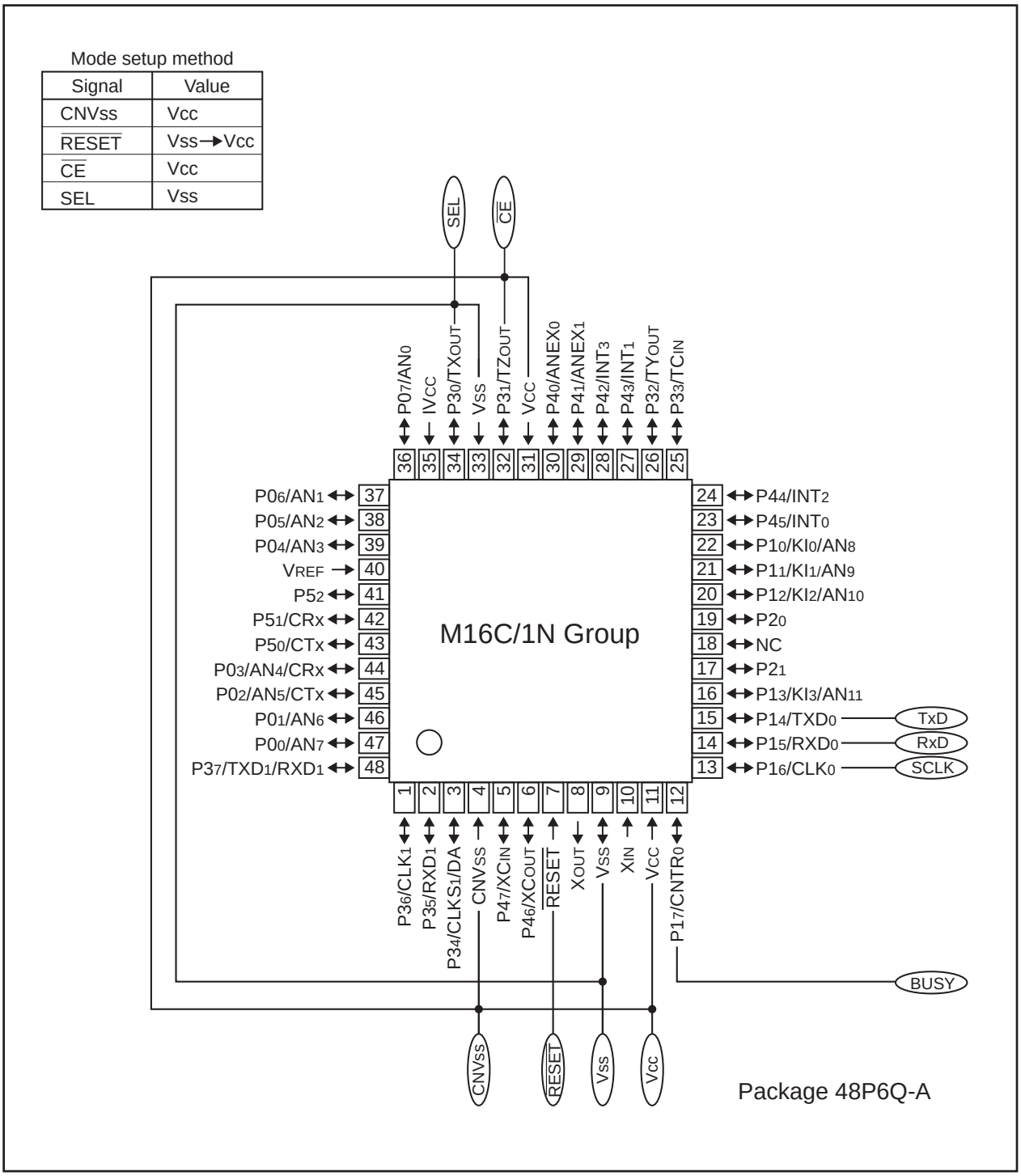
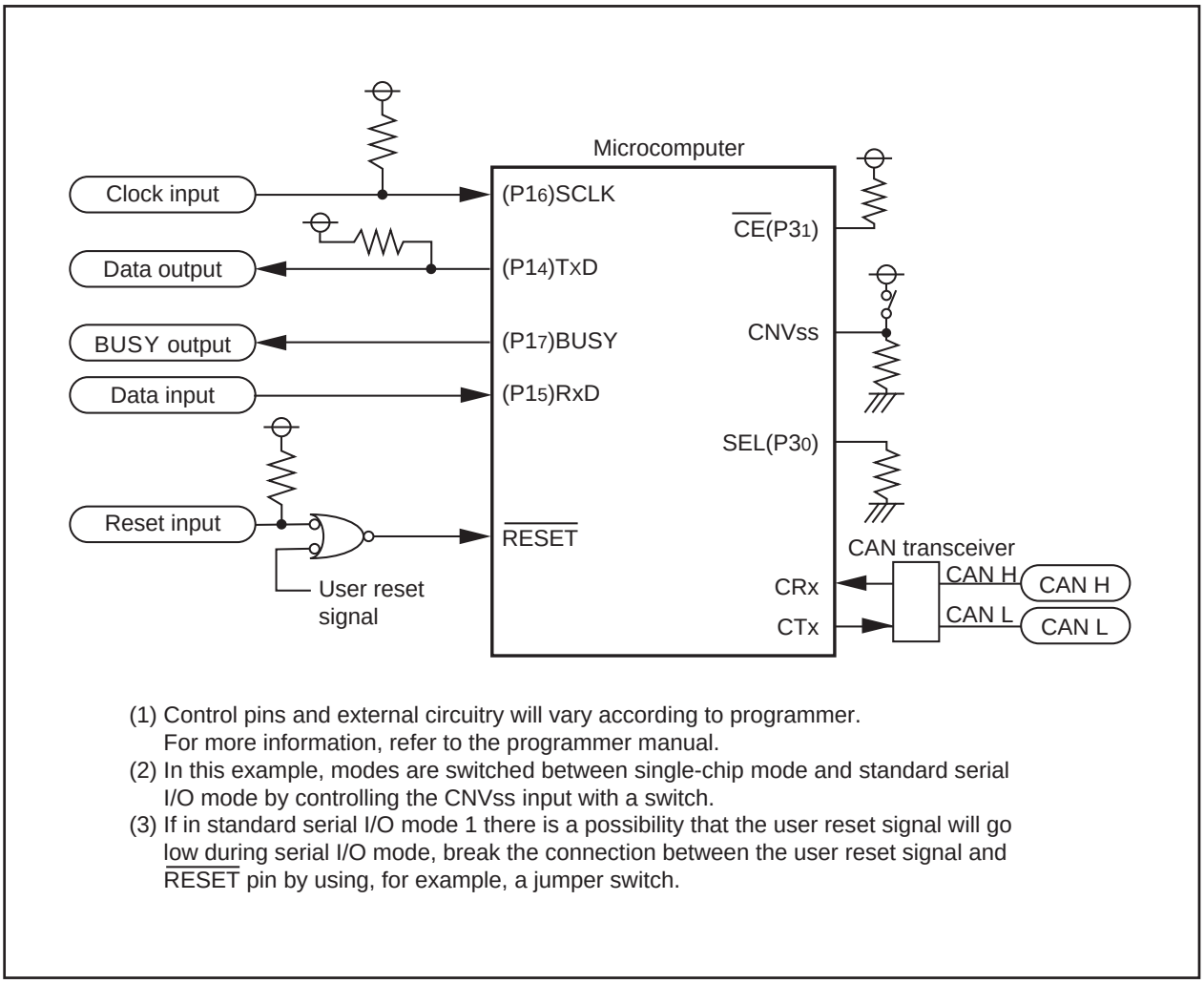


Figure 19.13 Pin connections for standard serial I/O mode

(1) Example for Processing Pins when Using Standard Serial Input/Output Mode

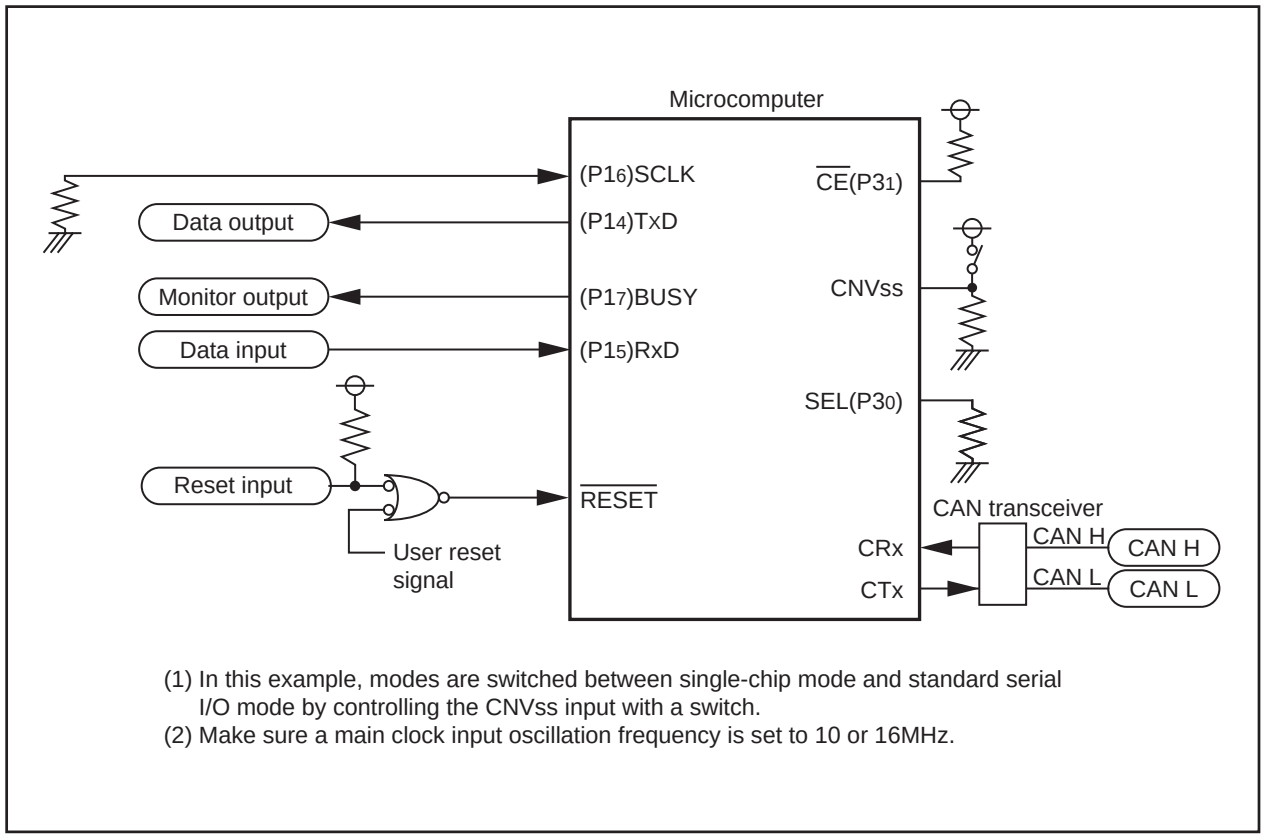
Figure 19.14 and 19.15 show example for processing pins when using standard serial I/O mode 1 and mode 2, respectively. Control pins will vary according to programmer, therefore refer to the programmer manual for more information.

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



(1) Control pins and external circuitry will vary according to programmer. For more information, refer to the programmer manual.
(2) In this example, modes are switched between single-chip mode and standard serial I/O mode by controlling the CNVss input with a switch.
(3) If in standard serial I/O mode 1 there is a possibility that the user reset signal will go low during serial I/O mode, break the connection between the user reset signal and $\overline{\text{RESET}}$ pin by using, for example, a jumper switch.

Figure 19.14 Example for processing pins when using standard serial I/O mode 1



(1) In this example, modes are switched between single-chip mode and standard serial I/O mode by controlling the CNVss input with a switch.
(2) Make sure a main clock input oscillation frequency is set to 10 or 16MHz.

Figure 19.15 Example for processing pins when using standard serial I/O mode 2

19.8 CAN Input/Output Mode

In CAN I/O mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a CAN programmer suitable for the M16C/1N group. For more information about CAN programmers, contact the manufacturer of your CAN programmer. For details on how to use, refer to the user's manual included with your CAN programmer.

Table 19.8 lists pin functions (flash memory CAN I/O mode). Figure 19.16 shows pin connections for CAN I/O mode.

19.8.1 ID code check function

This function determines whether the ID codes sent from the CAN programmer and those written in the flash memory match (refer to the description of **19.3 Functions to Inhibit Rewriting Flash Memory Version**)

Table 19.8 Pin functions (Flash memory CAN I/O mode)

Pin	Name	I/O	Description
V_{CC}, V_{SS}	Power input	I	Apply the voltage guaranteed for program and erase to Vcc pin and 0 V to Vss pin.
IV_{CC}	IV_{CC} input	I	Connect a capacitor (0.1μF) to V_{SS} pin.
CNV_{SS}	CNV_{SS} input	I	Connect to V_{CC} pin
$\overline{RESET}$	Reset input	I	Reset input pin. While $\overline{RESET}$ pin is "L" level, input a 20 cycle or longer clock to X_{IN} pin.
X_{IN}	Clock input	I	Connect a ceramic resonator or crystal oscillator between X_{IN} and X_{OUT} pins. To input an externally generated clock, input it to X_{IN} pin and open X_{OUT} pin.
X_{OUT}	Clock output	O	
V_{REF}	Reference voltage input	I	Enter the reference voltage for AD from this pin. Connect to V_{CC} or V_{SS} pin.
$P0_0$, $P0_1$ $P0_4$ to $P0_7$	Input port P0	I	Input "H" or "L" level signal or open.
$P0_2$	CT_X output	O	CAN output pin. Connect this pin to CAN transceiver.
$P0_3$	CR_X input	I	CAN input pin. Connect this pin to CAN transceiver.
$P1_0$ to $P1_5$, $P1_7$	Input port P1	I	Input "H" or "L" level signal or open.
$P1_6$	SCLK input	I	SCLK signal input pin. Input "L" level signal.
$P2_0$, $P2_1$	Input port P2	I	Input "H" or "L" level signal or open.
$P3_0$	SEL input	I	SEL signal input pin. Input "H" level signal.
$P3_1$	CE input	I	CE signal input pin. Input "H" level signal.
$P3_2$ 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.
$P5_0$ to $P5_2$	Input port P5	I	Input "H" or "L" level signal or open.

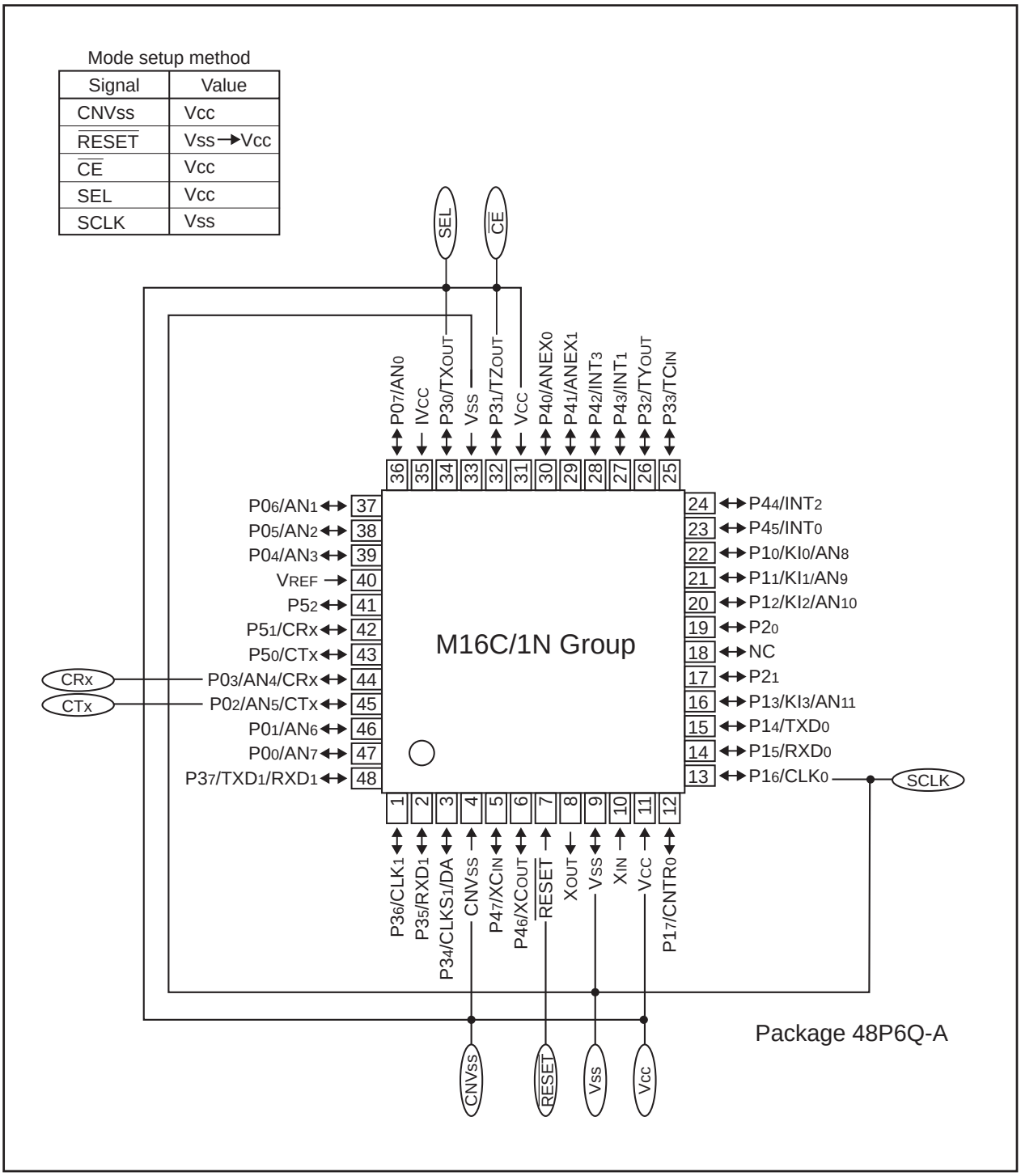
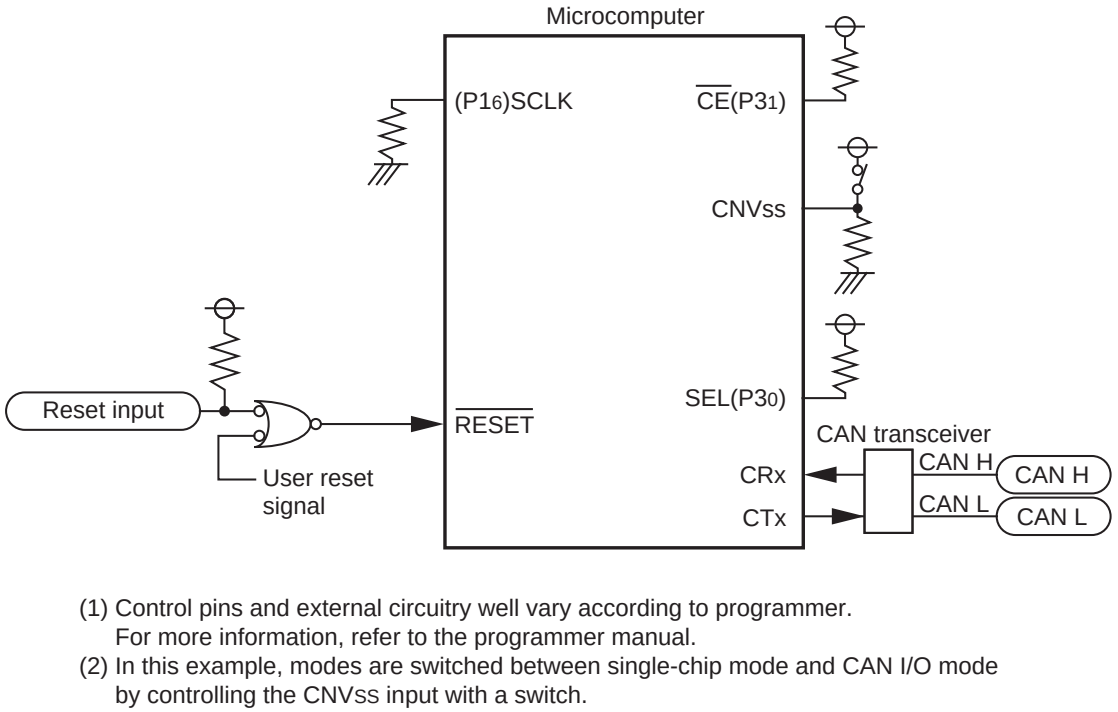


Figure 19.16 Pin connections for CAN I/O mode

(1) Example for Processing Pins when Using CAN Input/Output Mode

Figure 19.17 shows example for processing pins when using CAN I/O. Control pins will vary according to programmer, therefore refer to the programmer manual for more information.



(1) Control pins and external circuitry well vary according to programmer.
For more information, refer to the programmer manual.
(2) In this example, modes are switched between single-chip mode and CAN I/O mode by controlling the CNVss input with a switch.

Figure 19.17 Example for processing pins when using CAN I/O mode

20. Precautionary Notes in Using the Device

20.1 Clock

20.1.1 External Clock

Do not stop the external clock when it is connected to the X_{IN} pin and the main clock is selected as the CPU clock.

20.1.2 Power Control

1. When exiting stop mode by hardware reset, set $\overline{RESET}$ pin to "L" for at least 200 μs or longer.

2. Insert more than four NOP instructions after an WAIT instruction or a instruction to set the CM10 bit of the CM1 register to "1". 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" (all clocks stopped). The next instruction may be executed before entering wait mode or stop mode, depending on a combination of instruction and an execution timing.

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

4. Wait until the t_d(M-L) elapses or main clock oscillation stabilization time, whichever is longer, 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.

5. 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 of the ADCON1 register to "0" (V_{REF} not connection). When A/D conversion is performed, start the A/D conversion at least 1 μs or longer after setting the VCUT bit to "1" (V_{REF} connection).
   - **D/A converter**
     When not performing D/A conversion, set the DAE bit of the DACON register to "0" (input inhibited) and DA register to "00_{16}".
   - **Switching the oscillation-driving capacity**
     Set the driving capacity to "LOW" when oscillation is stable.
   - **External clock**
     When using an external clock input for the CPU clock, set the CM05 bit of the CM0 register to "1" (stop). Setting the CM05 bit to "1" disables the X_{OUT} pin from functioning, which helps to reduce the amount of current drawn in the chip. (When using an external clock input, note that the clock remains fed into the chip regardless of how the CM05 bit is set.)

20.1.3 Stop and Wait Modes

1. When returning from a stop mode by hardware reset, $\overline{\text{RESET}}$ pin must be "L" level until the main-clock has stabilized.

2. When switching to a stop or wait mode, 4 instructions are prefetched after the stop or wait instruction. And so, ensure that at least 4 NOPs follow the stop (the all-clock stop bit to "1") or wait instruction.

3. A Stop or wait mode is cancelled by a hardware reset or an interrupt. If an interrupt is to be used to cancel a stop or wait mode, that interrupt must first have been enabled, and the priority level of the interrupt which is not used to cancel must have been changed to 0 before shifting to either mode. If only a hardware reset is used to cancel a stop or wait mode, change the priority level of all interrupt to 0, then shift to either mode.

Example 1. When an interrupt is used to cancel wait mode
When canceling wait mode by a hardware reset and an $\overline{\text{INT}}_0$ interrupt.

Set the interrupt enable flag (I flag) to "0"	; Disable interrupt
Change the $\overline{\text{INT}}_0$ interrupt priority level to 1 or higher	; Enable $\overline{\text{INT}}_0$ interrupt In case of processor interrupt priority level = 0
Change all other interrupt priority level to 0	; Disable all other interrupts
Insert 4 NOPs instructions	; Prevent irregular interrupt occurring
Set the interrupt enable flag (I flag) to "1"	; Enable interrupts
WAIT instruction	
Insert 4 NOPs instructions	; Put at least 4 NOPs after a wait instruction because when switching to a wait mode, 4 instructions are prefetched after the wait instruction.

Example 2. When only hardware reset is used to cancel wait mode

Set the interrupt enable flag (I flag) to "0"	; Disable interrupt
Change all interrupt priority level to 0	; Disable maskable interrupt
WAIT instruction	
Insert 4 NOPs instructions	; Put at least 4 NOPs after a wait instruction because when switching to a wait mode, 4 instructions are prefetched after the wait instruction.

4. After returning from stop mode, an unexpected operation may occur (for example, undefined instruction interrupt, BRK instruction interrupt, etc.).
   Execute a JMP.B instruction after an instruction to write data to the all clock stop control bit. A program example is described as follows:

   Code examples are shown below.

   Example 1:

```
        BSET 0,     CM1         ; writing to the all clock stop control bit to "1" (stop mode)
        JMP.B       L1
    L1:
        NOP
        NOP
        NOP
        NOP
```

   Example 2:

```
        MOV.B:S     #21h, CM1   ; writing to the all clock stop control bit to "1" (stop mode)
        JMP.B       L1
    L1:
        NOP
        NOP
        NOP
        NOP
```

20.2 Interrupts

20.2.1 Reading Address 00000_{16}

Do not read the address 00000_{16} in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000_{16} during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to "0". If the address 00000_{16} is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to "0". This causes a problem that the interrupt is canceled, or an unexpected interrupt is generated.

20.2.2 Stack Pointer

Set the value of the stack pointer before accepting interrupts. Immediately after a reset, the value of the stack pointer is 0000_{16}. Accepting an interrupt before setting a value of the stack pointer may produce unpredictable results (runaway program, etc.) Make sure that you set the value of the stack pointer before accepting interrupts.

20.2.3 External interrupts

Clear the interrupt request bit to "0" when the $\overline{INT_0}$ to $\overline{INT_3}$ pins and $CNTR_0$ pin polarity are changed. The reason being is that an interrupt request may be generated when the polarity is changed.

20.2.4 Rewriting the Interrupt Control Register

When rewriting the Interrupt Control Register, do it at a point where it does not generate an interrupt request for that register. If there is a possibility that an interrupt may occur, disable the interrupt before rewriting. Examples are shown below.

Example 1:

```
INT_SWITCH1:
    FCLR    I           ; Disable interrupts.
    AND.B   #00H, 0055H ; Clear T1IC int. priority level and int. request bit.
    NOP                 ;
    NOP
    FSET    I           ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
    FCLR    I           ; Disable interrupts.
    AND.B   #00H, 0055H ; Clear T1IC int. priority level and int. request bit.
    MOV.W   MEM, R0     ; Dummy read.
    FSET    I           ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
    PUSHC   FLG         ; Push Flag register onto stack
    FCLR    I           ; Disable interrupts.
    AND.B   #00H, 0055H ; Clear T1IC int. priority level and int. request bit.
    POPC    FLG         ; Enable interrupts.
```

Note 1: The reason why two NOP instructions or dummy read were inserted before the FSET I for ex. 1 & 2 is to prevent interrupt enable flag from being set, due to the effects of instruction queue, before the rewritten value of the interrupt control register takes effect.

When an instruction to rewrite the interrupt control register is executed while the interrupt is disabled, depending on the instruction used for rewriting, there are times the interrupt request bit is not set even if an interrupt request for that register has been generated. If this creates a problem, please use any of the instructions below to rewrite the register.
Instructions : AND, OR, BCLR, BSET

20.2.5 Changing the interrupt request bit

When attempting to clear the interrupt request bit of an interrupt control register, the interrupt request bit is not cleared sometimes. This will depend on the instruction. If this creates problems, use the below instructions to change the register.
Instructions : MOV

20.3 Timer

20.3.1 Timer 1

1. Even if the prescaler 1 and Timer 1 are read out simultaneously in word-size, these registers are read byte-by-byte in the microcomputer. Consequently, the timer value may be updated during the period these two registers are being read.

20.3.2 Timers X, Y and Z

1. These timers stop counting after reset. Therefore, set values to Timer (X, Y, Z) and prescaler (X, Y, Z) before starting counting.
2. Even if prescaler (X, Y, Z) and Timer (X, Y, Z) are read out simultaneously in word-size, these registers are read byte-by-byte in the microcomputer. Consequently, the timer value may be updated during the period these two registers are being read.

20.3.3 Timer X

1. Using in the timer X pulse period measurement mode, the effectual edge reception flag and the timer X under flow flag are set to "0" by writing a "0" in a program. Writing a "1" has no effect. Write "1" in the other flag by using the MOV instruction when you make the flag of either one side "0" by program. (The clearance of the flag which isn't intend can be prevented.)

   Example:

   ```
   MOV.B    #10XXXXXXB,008BH
   ```

2. When changing to the timer X pulse period measurement mode from other mode, the contents of the effectual edge reception flag and the timer X under flow flag are indetermind. Write "0" in the effectual edge reception flag and the timer X under flow flag before starting the timer.
3. In the timer X pulse period measurement mode, use the MOV instruction to stop the timer.

   Example:

   ```
   MOV.B    #1100X00B,008BH
   ```

20.3.4 Timer Y

1. When count is stopped by writing "0" to the timer Y count start flag, the timer reloads the value of reload register and stops. Therefore, the timer count value should be read out before the timer stops.
2. When count is stopped by writing "0" to the timer Y count start flag, the timer Y interrupt request bit becomes "1" and an interrupt may occur. Thus, disable interrupts before the timer stops. Furthermore, set the Timer Y interrupt request bit to "0" before starting counting again.

20.3.5 Timer Z

1. When count is stopped by writing "0" to the timer Z count start flag, the timer reloads the value of reload register and stops. Therefore, the timer count value should be read out before the timer stops.
2. When count is stopped by writing "0" to the timer Z count start flag (all modes) or by writing "0" to the one-shot start bit (programmable one-shot generation mode/programmable wait one-shot generation mode), the timer Z interrupt request flag becomes "1" and an interrupt occurs. Thus, disable interrupts before the timer stops. Furthermore, set the Timer Z interrupt request bit to "0" before starting counting again.

20.3.6 Timer C

1. Read out the timer C or timer measurement register using in word-size.
   Even if the Timer C is read out in word-size, the timer value is not updated during the period the high-byte and low-byte are being read.

   Example:

```
MOV.W    0091H,R0    ;Read out timer C
```

20.4 Serial I/O

1. When reading data from the UARTi receive buffer in the clock asynchronous serial I/O mode, data should be read high-byte first then low-byte using a byte-size. If data is read as low-byte then high-byte or in word-size the framing error and parity error flags are cleared.

   A code example is shown below.

```
MOV.B    00A7H. R0H    ; Read the high-byte of UART0 receive buffer register
MOV.B    00A6H. R0L    ; Read the low-byte of UART0 receive buffer register
```

2. When writing data to the UARTi transmit buffer register in the clock asynchronous serial I/O mode with 9-bit transfer data length, data should be written high-byte first then low-byte using a byte-size.

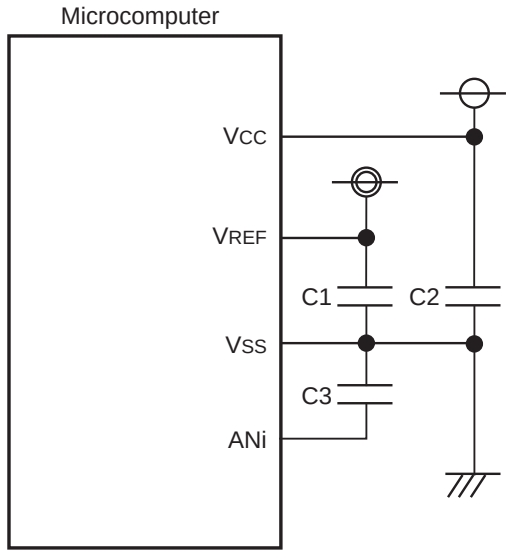
   A code example is shown below.

```
MOV.B    #XXH, 00A3H    ; Write the high-byte of UART0 transmit buffer register
MOV.B    #XXH, 00A2H    ; Write the low-byte of UART0 transmit buffer register
```

20.5 A/D Converter

1. Only write to each bit (except bit 6) of the AD Control Register 0, or each bit of the AD Control Register 1, or bit 0 of the AD Control Register 2 when AD conversion is stopped (before a trigger occurs). When the VREF connection bit is changed from "0" to "1", wait 1 µs or longer before starting AD conversion.

2. To prevent noise-induced device malfunction or latchup, as well as to reduce conversion errors, insert capacitors between the VCC, VREF, and analog input pins (ANi) each and the VSS pin. Figure 20.1 shows an example connection of each pin.



Note 1: C1≥0.47µF, C2≥0.47µF, C3≥100pF (reference).
Note 2: Use thick and shortest possible wiring to connect capacitors.

Figure 20.1 Example connection of each pin

3. Make sure the port direction bits for those pins that are used as analog inputs are set to "0" (input mode).

4. When setting the $\overline{KIi}$ input enable bit to "1" (Enabled) to use key input interrupt and using AN8 to AN11 as analog input pins, be careful about the following points.
   - A key input interrupt request is generated when the A/D input voltage goes "LOW".
   - If the A/D input voltage approaches 1/2 VCC, power supply current may increase due to pass current on schmitt circuit of key input interrupt. (When setting the $\overline{KIi}$ input enable bit to "0" (Disabled), pass current doesn't flow.)

5. The ØAD frequency must be 10 MHz or less. Without sample-and-hold function, limit the ØAD frequency to 250 kHz or more. With the sample and hold function, limit the ØAD frequency to 1 MHz or more.

6. When changing AD operation mode, select an analog pin again.

7. One Shot Mode
   Read the AD register only after confirming AD conversion is completed, which can be determined by using the AD conversion interrupt.

8. Repeat Mode
   Use the undivided main clock as the internal CPU clock when using this mode. The main clock can be divided by an internal divider circuit but make sure that you use main clock when using this mode.

9. If A/D conversion is forcibly terminated while in progress by setting the ADST bit of ADCON0 register to "0" (A/D conversion halted), the conversion result of the A/D converter is indeterminate. If the ADST bit is cleared to "0" in a program, ignore the value of A/D register.

20.6 CAN Module

20.6.1 Reading C0STR Register

The CAN module on the M16C/1N group updates the status of the C0STR register in a certain period. When the CPU and the CAN module access to the C0STR register at the same time, the CPU has the access priority; the access from the CAN module is disabled. Consequently, when the updating period of the CAN module matches the access period from the CPU, the status of the CAN module cannot be updated. (See Figure 20.2)

Accordingly, be careful about the following points so that the access period from the CPU should not match the updating period of the CAN module:

1. There should be a wait time of $3f_{CAN}$ or longer (see Table 20.1) before the CPU reads the C0STR register. (See Figure 20.3)

2. When the CPU polls the C0STR register, the polling period must be $3f_{CAN}$ or longer. (See Figure 20.4)

Table 20.1 CAN Module Status Updating Period

$3f_{CAN}$ period = 3 X X_{IN} (Original oscillation period) X Division value of the CAN clock (CCLK)	
(Example 1) Condition X_{IN} 16MHz CCLK: Divided by 1	$3f_{CAN}$ period = 3 X 62.5 ns X 1= 187.5 ns
(Example 2) Condition X_{IN} 16MHz CCLK: Divided by 2	$3f_{CAN}$ period = 3 X 62.5 ns X 2= 375 ns
(Example 3) Condition X_{IN} 16MHz CCLK: Divided by 4	$3f_{CAN}$ period = 3 X 62.5 ns X 4= 750 ns
(Example 4) Condition X_{IN} 16MHz CCLK: Divided by 8	$3f_{CAN}$ period = 3 X 62.5 ns X 8= 1.5 μs
(Example 5) Condition X_{IN} 16MHz CCLK: Divided by 16	$3f_{CAN}$ period = 3 X 62.5 ns X 16= 3 μs

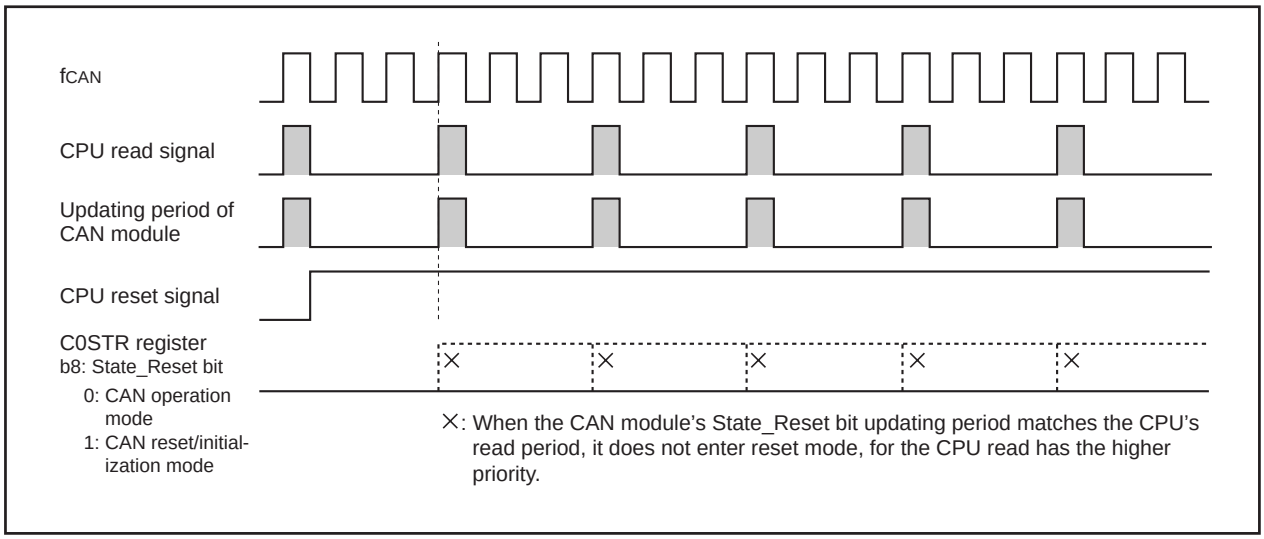


Figure 20.2 When Updating Period of CAN Module Matches Access Period from CPU

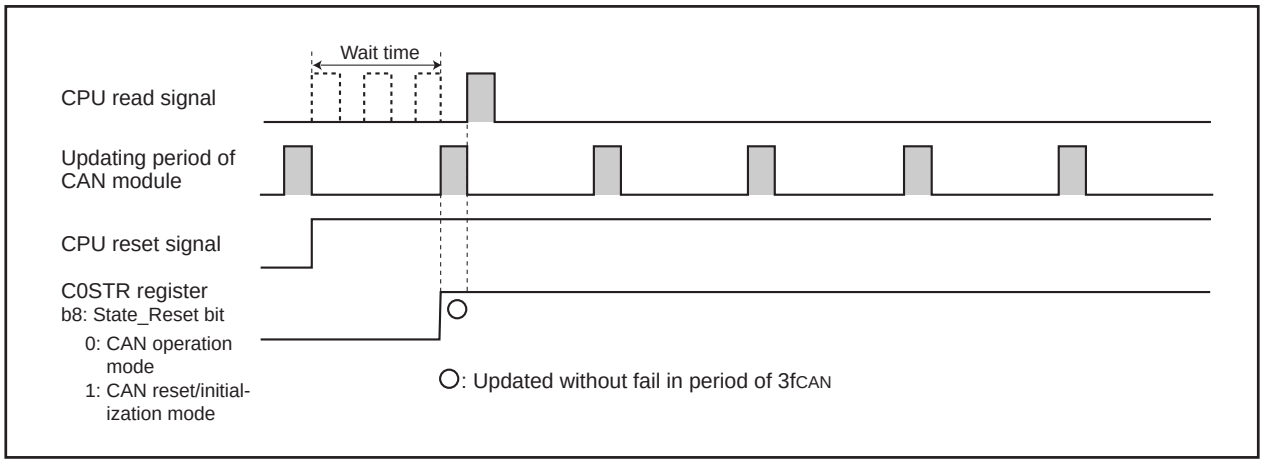


Figure 20.3 With a Wait Time of 3fCAN Before CPU Read

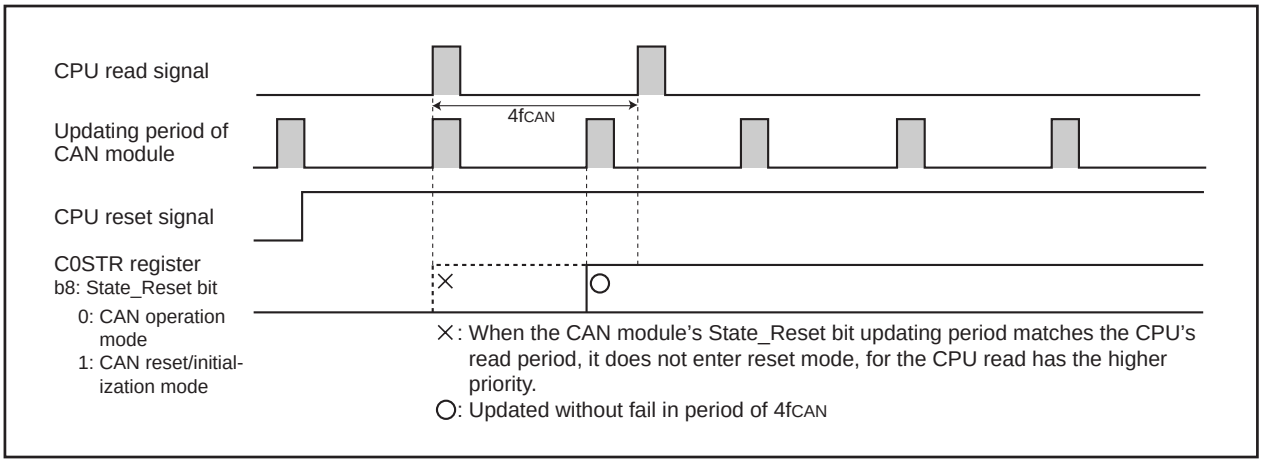
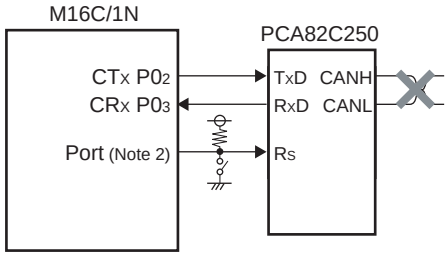
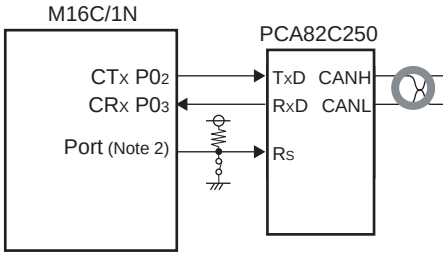


Figure 20.4 When Polling Period of CPU is 3fCAN or Longer

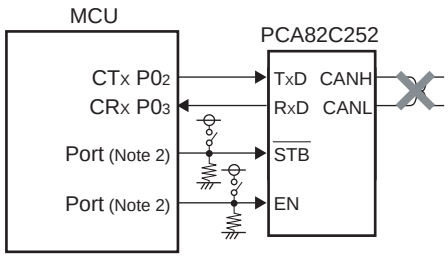
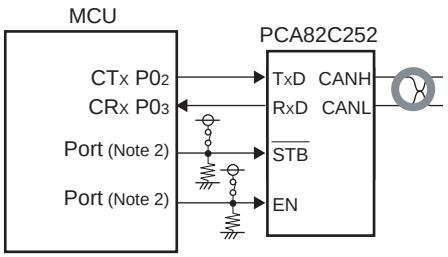
20.6.2 CAN Transceiver in Boot Mode

When programming the flash memory in boot mode via CAN bus, the operation mode of CAN transceiver should be set to "high-speed mode" or "normal operation mode". If the operation mode is controlled by the 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. Figure 20.5 shows pin connections of CAN transceiver.

In case of PCA82C250 (Philips product)

	Standby mode	high-speed mode
Rs pin (Note 1)	"H"	"L"
CAN communication	impossible	possible
connection		

In case of PCA82C252 (Philips product)

	sleep mode	normal operation mode
$\overline{\text{STB}}$ pin (Note 1) EN pin (Note 1)	"L" "L"	"H" "H"
CAN communication	impossible	possible
connection		

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

Figure 20.5 CAN Transceiver Connection

20.7 Noise

1. Bypass Capacitor between VCC and VSS Pins
   Insert a bypass capacitor (at least 0.1 μF) between VCC and VSS pins as noise and latch-up countermeasures. In addition, make sure that connecting lines are the shortest and widest possible.

2. Port Control Registers Data Read Error
   During severe noise testing, mainly power supply system noise, and introduction of external noise, the data of port related registers may changed. As a firmware countermeasure, it is recommended to periodically re-set the port registers, port direction registers and pull-up control registers. However, you should fully examine before introducing the re-set routine as conflicts may be created between this re-set routine and interrupt routines (i. e. ports are switched during interrupts).

3. CNVss pin wiring
   CNVSS pin functions as a pin to change to shipment examination mode or flash memory rewrite mode in the flash memory version.
   In order to improve the pin tolerance to noise, insert a pull down resistance (about 5 kΩ) between CNVss and Vss, and placed as close as possible to the CNVss pin.

20.8 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.

20.9 Flash Memory Version

20.9.1 Functions to Prevent Flash Memory from Rewriting

ID codes are stored in addresses 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716, and 0FFFFB16. If wrong data are written to these addresses, the flash memory cannot be read or written in standard serial I/O mode and CAN I/O mode.

The ROMCP register is mapped in address 0FFFFF16. 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.

20.9.2 Stop Mode

When entering stop mode, the following settings are required:

- Set the CM10 bit to "1" (stop mode) after setting FMR01 bit to "0" (CPU rewrite mode disable).
- 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 from stop mode
```

20.9.3 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.

20.9.4 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

20.9.5 Writing Command and Data

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

20.9.6 Program Command

By writing "xx4016" 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.

20.9.7 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).

20.9.8 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

20.9.9 Interrupt

EW0 Mode

To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.

- The watchdog timer interrupt is 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 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.

20.9.10 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 between the instruction to set the bit to "0" and the instruction to set the bit to "1".

20.9.11 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, parallel I/O mode, or CAN I/O mode.

EW1 Mode

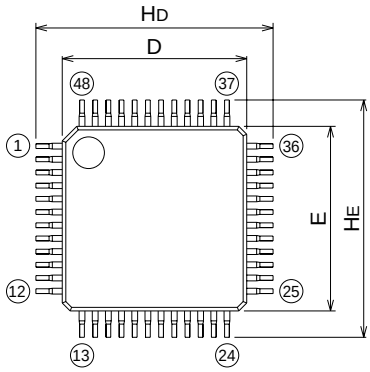
Avoid rewriting any block in which the rewrite control program is stored.

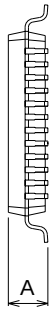
Package Dimension

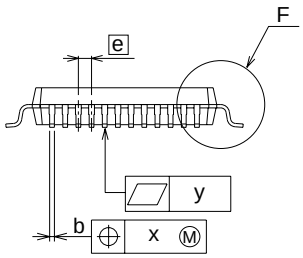
48P6Q-A Recommended

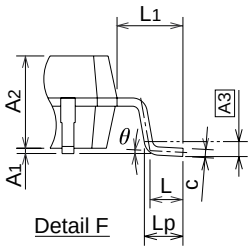
Plastic 48pin 7×7mm body LQFP

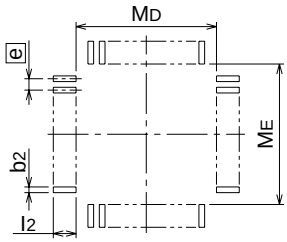
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP48-P-77-0.50	–	–	Cu Alloy











Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A1	0	0.1	0.2
A2	–	1.4	–
b	0.17	0.22	0.27
c	0.105	0.125	0.175
D	6.9	7.0	7.1
E	6.9	7.0	7.1
e	–	0.5	–
HD	8.8	9.0	9.2
HE	8.8	9.0	9.2
L	0.35	0.5	0.65
L1	–	1.0	–
Lp	0.45	0.6	0.75
A3	–	0.25	–
x	–	–	0.08
y	–	–	0.1
θ	0°	–	8°
b2	–	0.225	–
l2	1.0	–	–
MD	–	7.4	–
ME	–	7.4	–

Register Index

REVISION HISTORY	M16C/1N Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Oct 20, 2004	–	First edition issued (Renesas Technology version)

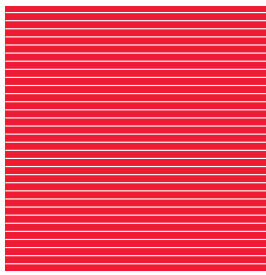
Blank page

M16C/1N Group Hardware Manual

Publication Data : Rev.1.00 Oct 20, 2004

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

M16C/1N Group
Hardware Manual



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