REJ09B0019-0091Z

RENESAS

16

R8C/10 Group

Hardware Manual

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER

M16C FAMILY / R8C /Tiny SERIES

Preliminary

Hardware Manual

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

Rev. 0.91
Revision date: Sep 8, 2003

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

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

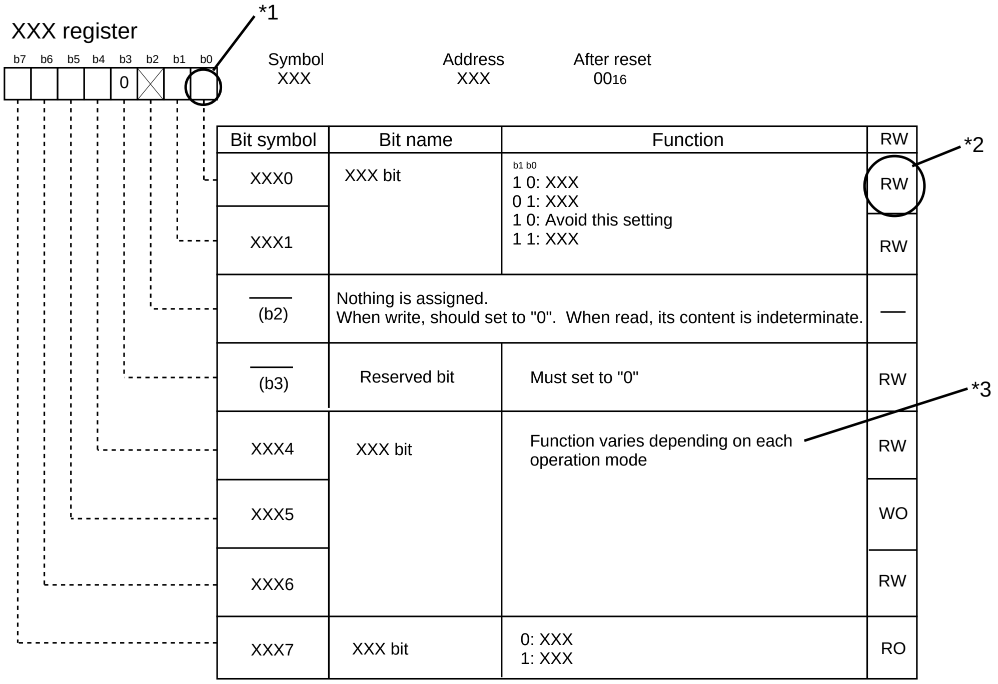
  Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
- When using any or all of the information contained in these materials, including product data, diagrams,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

  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.

How to Use This Manual

This hardware manual provides detailed information on features in the R8C/10 Group microcomputer.
Users are expected to have basic knowledge of electric circuits, logical circuits and microcomputer.

Each register diagram contains bit functions with the following symbols and descriptions.



Bit symbol	Bit name	Function	RW
XXX0	XXX bit	b1 b0 1 0: XXX 0 1: XXX 1 0: Avoid this setting 1 1: XXX	RW
XXX1			RW
— (b2)	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
— (b3)	Reserved bit	Must set to "0"	RW
XXX4	XXX bit	Function varies depending on each operation mode	RW
XXX5			WO
XXX6			RW
XXX7	XXX bit	0: XXX 1: XXX	RO

*1

Blank: Set to "0" or "1" according to your intended use
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

Terms to use here are explained as follows.

- Nothing is assigned
  Nothing is assigned to the bit concerned. When write, set to "0" for new function in future plan.
- Reserved bit
  Reserved bit. Set the specified value.
- Avoid this setting
  The operation at having selected is not guaranteed.
- Function varies depending on each operation mode
  Bit function varies depending on peripheral function mode.
  Refer to register diagrams in each mode.

M16C Family Documents

Document	Contents
Short Sheet	Hardware overview
Data Sheet	Hardware overview and electrical characteristics
Hardware Manual	Hardware specifications (pin assignments, memory maps, specifications of peripheral functions, electrical characteristics, timing charts)
Software Manual	Detailed description about instructions and microcomputer performance by each instruction
Application Note	• Application examples of peripheral functions • Sample programs • Introductory description about basic functions in M16C family • Programming method with the assembly and C languages

Table of Contents

SFR Page Reference

Chapter 1. Overview 1

1.1 Applications 1

1.2 Performance Outline 2

1.3 Block Diagram 3

1.4 Product Information 4

1.5 Pin Assignments 5

1.6 Pin Description 6

Chapter 2. Central Processing Unit (CPU) 7

2.1 Data Registers (R0, R1, R2 and R3) 7

2.2 AddressRegisters (A0 and A1) 7

2.3 Frame Base Register(FB) 8

2.4 Interrupt Table Register (INTB) 8

2.5 Program Counter (PC) 8

2.6 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) 8

2.7 Static Base Register (SB) 8

2.8 Flag Register (FLG) 8

2.8.1 Carry Flag (C Flag) 8

2.8.2 Debug Flag (D Flag) 8

2.8.3 Zero Flag (Z Flag) 8

2.8.4 Sign Flag (S Flag) 8

2.8.5 Register Bank Select Flag (B Flag) 8

2.8.6 Overflow Flag (O Flag) 8

2.8.7 Interrupt Enable Flag (I Flag) 8

2.8.8 Stack Pointer Select Flag (U Flag) 8

2.8.9 Processor Interrupt Priority Level (IPL) 8

2.8.10 Reserved Area 8

Chapter 3. Memory 9

Chapter 4. Special Function Registers (SFR) 10

Chapter 5. Reset 14

5.1 Hardware Reset 14

5.2 Software Reset 14

5.3 Watchdog Timer Reset 14

Chapter 6. Clock Generating Circuit 17

6.1 Main Clock 21
6.2 Ring Oscillator Clock 22
6.3 CPU Clock and Peripheral Function Clock 23
6.3.1 CPU Clock 21
6.3.2 Peripheral Function Clock (f1, f2, f8, f32, fAD, f1SIO, f8SIO, f32SIO) 23
6.3.3 f_{RING} and $f_{RING128}$ 23
6.4 Power Control 24
6.4.1 Normal Operation Mode 24
6.4.2 Wait Mode 25
6.4.3 Stop Mode 26
6.5 Oscillation Stop Detection Function 29
6.5.1 How to Use Oscillation Stop Detection Function 29
Chapter 7. Protection 31
Chapter 8. Processor Mode 32
8.1 Types of Processor Mode 32
Chapter 9. Bus 33
Chapter 10. Interrupt 34
10.1 Interrupt Overview 34
10.1.1 Type of Interrupts 34
10.1.2 Software Interrupts 35
10.1.3 Hardware Interrupts 36
10.1.4 Interrupts and Interrupt Vector 37
10.1.5 Interrupt Control 39
10.2 $\overline{INT}$ Interrupt 47
10.2.1 $\overline{INT0}$ Interrupt 47
10.2.2 $\overline{INT0}$ Input Filter 48
10.2.3 $\overline{INT1}$ Interrupt and $\overline{INT2}$ Interrupt 49
10.2.4 $\overline{INT3}$ Interrupt 50
10.3 Key Input Interrupt 51
10.4 Address Match Interrupt 52
Chapter 11. Watchdog Timer 54
Chapter 12. Timers 56
12.1 Timer X 57
12.1.1 Timer Mode 59
12.1.2 Pulse Output Mode 60
12.1.3 Event Counter Mode 61
12.1.4 Pulse Width Measurement Mode 62
12.1.5 Pulse Period Measurement Mode 64
12.2 Timer Y 66

12.2.1 Timer Mode 69
12.2.2 Programmable Waveform Generation Mode 71
12.3 Timer Z 74
12.3.1Timer Mode 77
12.3.2 Programmable Waveform Generation Mode 79
12.3.3 Programmable One-shot Generation Mode 81
12.3.4 Programmable Wait One-shot Generation Mode 84
12.4 Timer C 87
Chapter 13. Serial I/O 90
13.1 Clock Synchronous Serial I/O Mode 95
13.1.1 Polarity Select Function 98
13.1.2 LSB First/MSB First Select Function 98
13.1.3 Continuous Receive Mode 99
13.2 Clock Asynchronous Serial I/O (UART) Mode 100
13.2.1 TxD10/RxD1 Select Function (UART1) 103
13.2.2 TxD11 Select Function (UART1) 103
Chapter 14. A-D Converter 104
14.1 One-shot Mode 108
14.2 Repeat Mode 109
14.3 Sample & Hold 110
Chapter 15. Programmable I/O Ports 111
15. 1 Description 111
15.1.1 Port Pi Direction Register (PDi Register, i = 0, 1, 3, 4) 111
15.1.2 Port Pi Register (Pi Register, i = 0 to 4) 111
15.1.3 Pull-up Control Register 0, Pull-up Control Register 1 (PUR0 and PUR1 Registers) 111
15.1.4 Port P1 Drive Capacity Control Register (DRR Register) 111
15.2 Unassigned Pin Handling 118
Chapter 16. Electrical Characteristics 119
Chapter 17. Flash Memory Version 125
17.1 Overview 125
17.2 Memory Map 126
17.3 Functions To Prevent Flash Memory from Rewriting 127
17.3.1 ID Code Check Function 127
17.4 CPU Rewrite Mode 128
17.4.1 EW0 Mode 129
17.4.2 EW1 Mode 129
17.4.3 Software Commands 135
17.4.4 Status Register 139
17.4.5 Full Status Check 140

17.5 Standard Serial I/O Mode 142

17.5.1 ID Code Check Function 142

Chapter 18. On-chip Debugger 146

18.1 Address Match Interrupt 146

18.2 Single Step Interrupt 146

18.3 UART1 146

18.4 BRK Instrucstion 146

Chapter 19. Usage Notes 147

19.1 Stop Mode and Wait Mode 147

19.2 Interrupts 147

19.2.1 Reading Address 0000016 147

19.2.2 SP Setting 147

19.2.3 External Interrupt and Key Input Interrupt 147

19.2.4 Watchdog Timer Interrupt 147

19.2.5 Changing Interrupt Source 147

19.2.6 Changing Interrupt Control Register 149

19.3 Timers 150

19.3.1 Timers X, Y and Z 150

19.3.2 Timer X 150

19.3.3 Timer Z 150

19.3.4 Timer C 150

19.4 Serial I/O 151

19.5 A-D Converter 152

19.6 Flash Memory Version 153

19.6.1 CPU Rewrite Mode 153

19.7 Noise 155

Chapter 20. Usage Notes for On-chip Debugger 156

Appendix 1 Package Dimensions 157

Appendix 2 Connecting Examples for Serial Writer and On-chip Debugging Emulator 158

Register Index 160

R8C/10 Group Usage Note Reference Book

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

SFR Page Reference

Address	Register	Symbol	Page
0000_{16}			
0001_{16}			
0002_{16}			
0003_{16}			
0004_{16}	Processor mode register 0	PM0	32
0005_{16}	Processor mode register 1	PM1	32
0006_{16}	System clock control register 0	CM0	19
0007_{16}	System clock control register 1	CM1	19
0008_{16}			
0009_{16}	Address match interrupt enable register	AIER	53
$000A_{16}$	Protect register	PRCR	31
$000B_{16}$			
$000C_{16}$	Oscillation stop detection register	OCD	20
$000D_{16}$	Watchdog timer reset register	WDTR	55
$000E_{16}$	Watchdog timer start register	WDTS	55
$000F_{16}$	Watchdog timer control register	WDC	55
0010_{16}	Address match interrupt register 0	RMAD0	53
0011_{16}			
0012_{16}			
0013_{16}			
0014_{16}	Address match interrupt register 1	RMAD1	53
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	47
$001F_{16}$			
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}			
0046_{16}			
0047_{16}			
0048_{16}			
0049_{16}			
$004A_{16}$			
$004B_{16}$			
$004C_{16}$			
$004D_{16}$	Key input interrupt control register	KUPIC	40
$004E_{16}$	A-D conversion interrupt control register	ADIC	40
$004F_{16}$			
0050_{16}			
0051_{16}	UART0 transmit interrupt control register	S0TIC	40
0052_{16}	UART0 receive interrupt control register	S0RIC	40
0053_{16}	UART1 transmit interrupt control register	S1TIC	40
0054_{16}	UART1 receive interrupt control register	S1RIC	40
0055_{16}	$\overline{\text{INT2}}$ interrupt control register	INT2IC	40
0056_{16}	Timer X interrupt control register	TXIC	40
0057_{16}	Timer Y interrupt control register	TYIC	40
0058_{16}	Timer Z interrupt control register	TZIC	40
0059_{16}	$\overline{\text{INT1}}$ interrupt control register	INT1IC	40
$005A_{16}$	INT3 interrupt control register	INT3IC	40
$005B_{16}$	Timer C interrupt control register	TCIC	40
$005C_{16}$			
$005D_{16}$	$\overline{\text{INT0}}$ interrupt control register	INT0IC	40
$005E_{16}$			
$005F_{16}$			
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}			
$007A_{16}$			
$007B_{16}$			
$007C_{16}$			
$007D_{16}$			
$007E_{16}$			
$007F_{16}$			

Blank columns are all reserved space. No use is allowed.

SFR Page Reference

Address	Register	Symbol	Page
0080_{16}	Timer Y, Z mode register	TYZMR	66/74
0081_{16}	Prescaler Y register	PREY	67
0082_{16}	Timer Y secondary register	TYSC	67
0083_{16}	Timer Y primary register	TYPR	67
0084_{16}	Timer Y, Z waveform output control register	PUM	68/76
0085_{16}	Prescaler Z register	PREZ	75
0086_{16}	Timer Z secondary register	TZSC	75
0087_{16}	Timer Z primary register	TZPR	75
0088_{16}			
0089_{16}			
$008A_{16}$	Timer Y, Z output control register	TYZOC	67/75
$008B_{16}$	Timer X mode register	TXMR	57
$008C_{16}$	Prescaler X register	PREX	58
$008D_{16}$	Timer X register register	TX	58
$008E_{16}$	Count source set register	TCSS	58
$008F_{16}$			
0090_{16}	Timer C register	TC	88
0091_{16}			
0092_{16}			
0093_{16}			
0094_{16}			
0095_{16}			
0096_{16}	External input enable register	INTEN	47
0097_{16}			
0098_{16}	Key input enable register	KIEN	51
0099_{16}			
$009A_{16}$	Timer C control register 0	TCC0	88
$009B_{16}$	Timer C control register 1	TCC1	88
$009C_{16}$	Capture register	TM0	88
$009D_{16}$			
$009E_{16}$			
$009F_{16}$			
$00A0_{16}$	UART0 transmit/receive mode register	U0MR	93
$00A1_{16}$	UART0 bit rate generator	U0BRG	92
$00A2_{16}$	UART0 transmit buffer register	U0TB	92
$00A3_{16}$			
$00A4_{16}$	UART0 transmit/receive control register 0	U0C0	93
$00A5_{16}$	UART0 transmit/receive control register 1	U0C1	94
$00A6_{16}$	UART0 receive buffer register	U0RB	92
$00A7_{16}$			
$00A8_{16}$	UART1 transmit/receive mode register	U1MR	93
$00A9_{16}$	UART1 bit rate generator	U1BRG	92
$00AA_{16}$	UART1 transmit buffer register	U1TB	92
$00AB_{16}$			
$00AC_{16}$	UART1 transmit/receive control register 0	U1C0	93
$00AD_{16}$	UART1 transmit/receive control register 1	U1C1	94
$00AE_{16}$	UART1 receive buffer register	U1RB	92
$00AF_{16}$			
$00B0_{16}$	UART transmit/receive control register 2	UCON	94
$00B1_{16}$			
$00B2_{16}$			
$00B3_{16}$			
$00B4_{16}$			
$00B5_{16}$			
$00B6_{16}$			
$00B7_{16}$			
$00B8_{16}$			
$00B9_{16}$			
$00BA_{16}$			
$00BB_{16}$			
$00BC_{16}$			
$00BD_{16}$			
$00BE_{16}$			
$00BF_{16}$			

Blank columns are all reserved space. No use is allowed.

Address	Register	Symbol	Page
$00C0_{16}$	A-D register	AD	107
$00C1_{16}$			
$00C2_{16}$			
$00C3_{16}$			
$00C4_{16}$			
$00C5_{16}$			
$00C6_{16}$			
$00C7_{16}$			
$00C8_{16}$			
$00C9_{16}$			
$00CA_{16}$			
$00CB_{16}$			
$00CC_{16}$			
$00CD_{16}$			
$00CE_{16}$			
$00CF_{16}$			
$00D0_{16}$			
$00D1_{16}$			
$00D2_{16}$			
$00D3_{16}$			
$00D4_{16}$	A-D control register 2	ADCON2	107
$00D5_{16}$			
$00D6_{16}$	A-D control register 0	ADCON0	106
$00D7_{16}$	A-D control register 1	ADCON1	106
$00D8_{16}$			
$00D9_{16}$			
$00DA_{16}$			
$00DB_{16}$			
$00DC_{16}$			
$00DD_{16}$			
$00DE_{16}$			
$00DF_{16}$			
$00E0_{16}$	Port P0 register	P0	116
$00E1_{16}$	Port P1 register	P1	116
$00E2_{16}$	Port P0 direction register	PD0	116
$00E3_{16}$	Port P1 direction register	PD1	116
$00E4_{16}$			
$00E5_{16}$	Port P3 register	P3	116
$00E6_{16}$			
$00E7_{16}$	Port P3 direction register	PD3	116
$00E8_{16}$	Port P4 register	P4	116
$00E9_{16}$			
$00EA_{16}$	Port P4 direction register	PD4	116
$00EB_{16}$			
$00EC_{16}$			
$00ED_{16}$			
$00EE_{16}$			
$00EF_{16}$			
$00F0_{16}$			
$00F1_{16}$			
$00F2_{16}$			
$00F3_{16}$			
$00F4_{16}$			
$00F5_{16}$			
$00F6_{16}$			
$00F7_{16}$			
$00F8_{16}$			
$00F9_{16}$			
$03FA_{16}$			
$00FB_{16}$			
$00FC_{16}$	Pull-up control register 0	PUR0	117
$00FD_{16}$	Pull-up control register 1	PUR1	117
$00FE_{16}$	Port P1 drivability control register	DRR	117
$00FF_{16}$			
≈			≈
$01B3_{16}$	Flash memory control register 4	FMR4	133
$01B4_{16}$			
$01B5_{16}$	Flash memory control register 1	FMR1	133
$01B6_{16}$			
$01B7_{16}$	Flash memory control register 0	FMR0	132

1. Overview

This MCU is built using the high-performance silicon gate CMOS process using a R8C Tiny Series CPU core and is packaged in a 32-pin plastic molded LQFP. This MCU operates using sophisticated instructions featuring a high level of instruction efficiency. With 1M bytes of address space, it is capable of executing instructions at high speed.

1.1 Applications

Electric household appliance, office equipment, housing equipment (sensor, security), general industrial equipment, audio, etc.

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.

1.2 Performance Outline

Table 1.1. lists the performance outline of this MCU.

Table 1.1 Performance outline

Item		Performance
CPU	Number of basic instructions	89 instructions
	Shortest instruction execution time	62.5 ns (f(XIN) = 16 MHz, VCC = 3.0 to 5.5 V) 100 ns (f(XIN) = 10 MHz, VCC = 2.7 to 5.5 V)
	Operating mode	Single-chip
	Address space	1M bytes
	Memory capacity	See Table 1.2 “Product List”
Peripheral function	Interrupt	Internal: 10 sources, External: 5 sources, Software: 4 sources, Priority level: 7 levels
	Watchdog timer	15 bits x 1 (with prescaler)
	Timer	Timer X: 8 bits x 1 channel, Timer Y: 8 bits x 1 channel, Timer Z: 8 bits x 1 channel (Each timer equipped with 8-bit prescaler) Timer C: 16 bits x 1 channel Input capture circuit
	Serial I/O	•1 channel Clock synchronous, UART •1 channel UART
	A-D converter	10-bit A-D converter: 1 circuit, 8 channels
	Clock generation circuit	2 circuits •Main clock generation circuit (Equipped with a built-in feedback resistor) •Ring oscillator
	Oscillation stop detection function	Stop detection of main clock oscillation
	Port	Input/Output: 22 (including LED drive port), Input: 2 (LED drive I/O port: 8, max. 20 mA)

Electrical characteristics	Power supply voltage	VCC = 3.0 to 5.5 V (f(XIN) = 16 MHz)
		VCC = 2.7 to 5.5 V (f(XIN) = 10 MHz)

	Power consumption	TBD (VCC = 5.0 V, (f(XIN) = 16 MHz) TBD (VCC = 3.0 V, (f(XIN) = 10 MHz) TBD (VCC = 3.0 V, Wait mode) TBD (VCC = 3.0 V, Stop mode)
Flash memory	Program/erase voltage	VCC = 2.7 to 5.5 V
	Number of program/erase	100 times
Operating ambient temperature		-20 to 85 °C -40 to 85 °C (option)
Package		32-pin plastic mold LQFP

If you require this option, please specify so.

1.3 Block Diagram

Figure 1.1. shows this MCU block diagram.

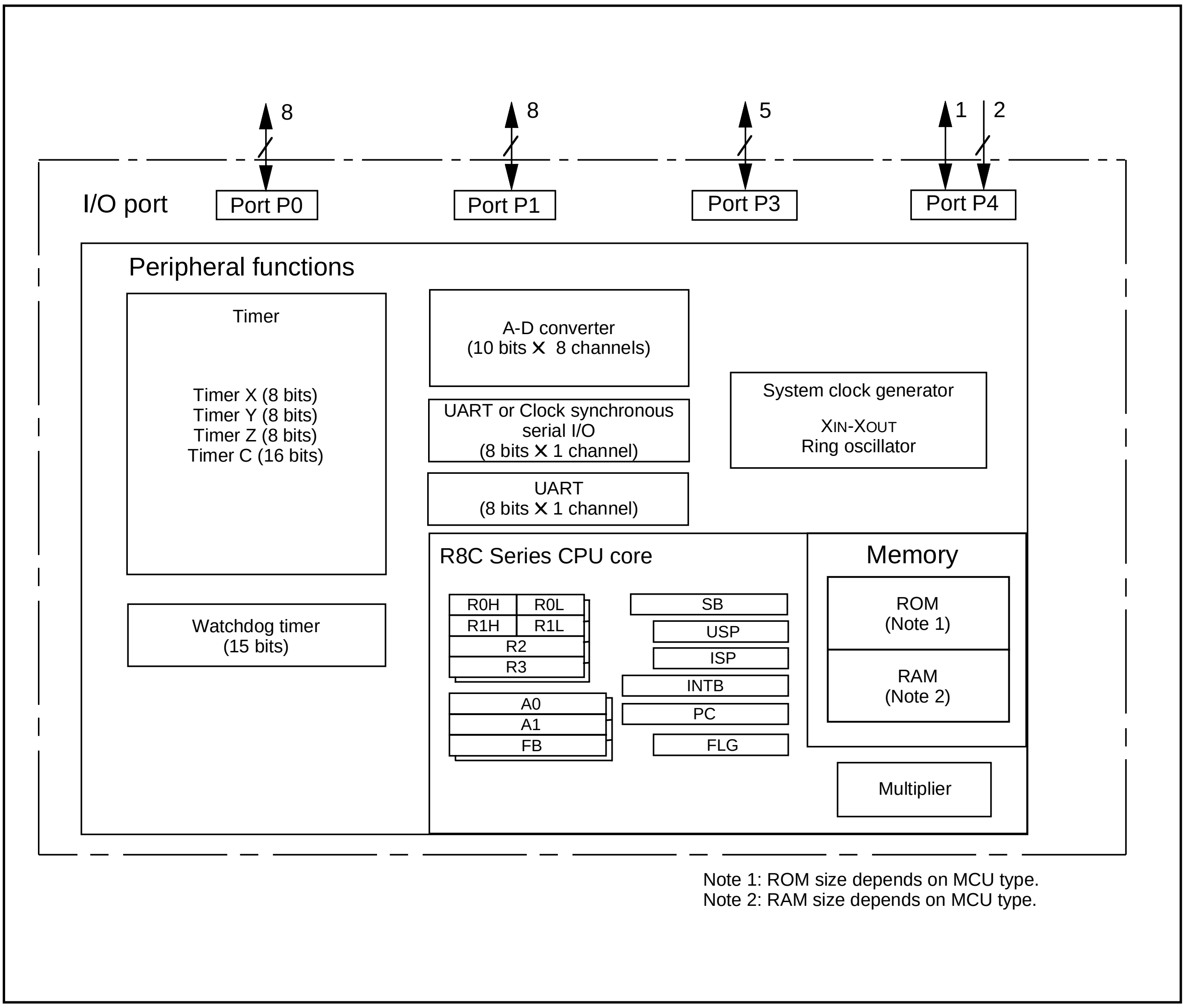


Figure 1.1 Block Diagram

1.4 Product Information

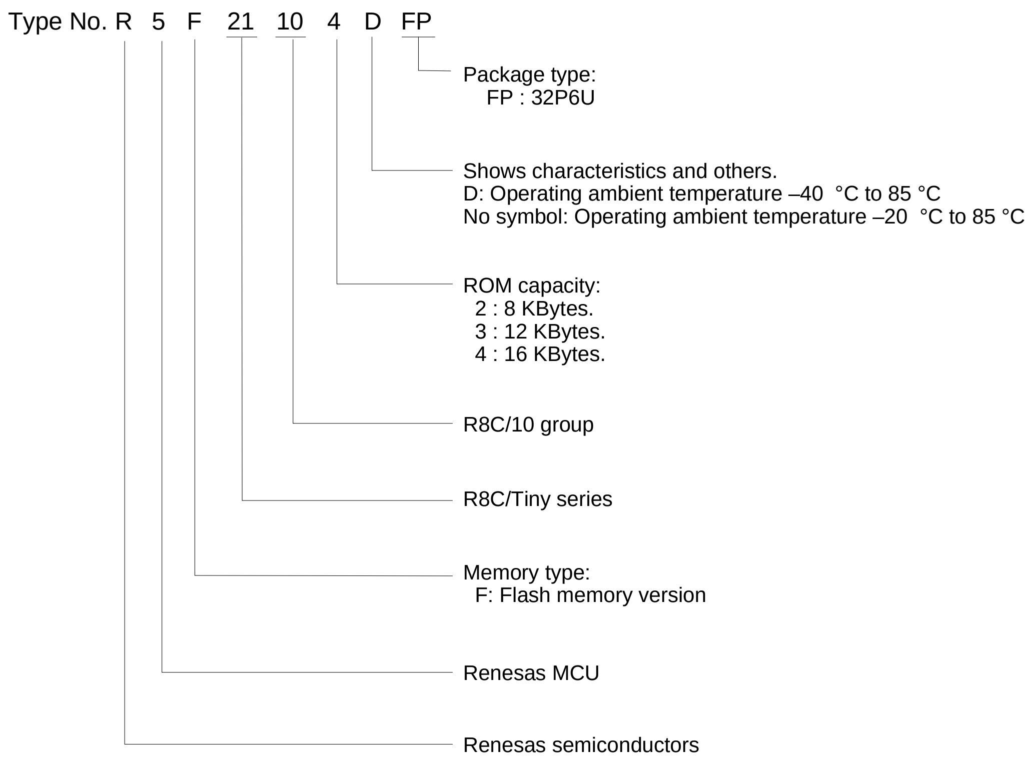
Table 1.2 lists the products.

Table 1.2 Product List

As of September 2003

Type No.		ROM capacity	RAM capacity	Package type	Remarks
R5F21102FP	**	8K bytes	512 bytes	32P6U-A	Flash memory version
R5F21103FP	**	12K bytes	768 bytes	32P6U-A	
R5F21104FP	**	16K bytes	1K bytes	32P6U-A	
R5F21102DFP	**	8K bytes	512 bytes	32P6U-A	
R5F21103DFP	**	12K bytes	768 bytes	32P6U-A	
R5F21104DFP	**	16K bytes	1K bytes	32P6U-A	

** : Under development



Type No. R 5 F 21 10 4 D FP

Package type:
FP : 32P6U

Shows characteristics and others.
D: Operating ambient temperature –40 °C to 85 °C
No symbol: Operating ambient temperature –20 °C to 85 °C

ROM capacity:
2 : 8 KBytes.
3 : 12 KBytes.
4 : 16 KBytes.

R8C/10 group

R8C/Tiny series

Memory type:
F: Flash memory version

Renesas MCU

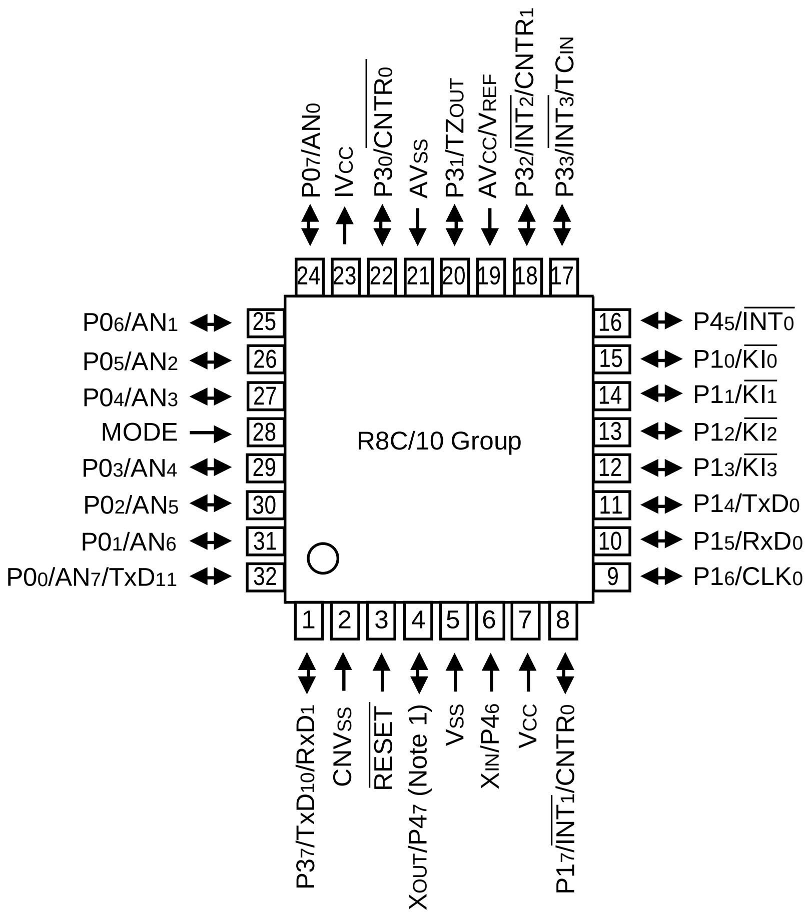
Renesas semiconductors

Figure 1.2 Type No., Memory Size, and Package

1.5 Pin Configuration

Figure 1.3 shows the pin configuration (top view).

PIN CONFIGURATION (top view)



Note 1: $P4_7$ functions only as an input port.

Package: 32P6U-A

Figure 1.3 Pin Configuration (Top View)

1.6 Pin Description

Table 1.3 shows the pin description

Table 1.3 Pin description

Signal name	Pin name	I/O type	Function
Power supply input	Vcc, Vss	Input	Apply 2.7 V to 5.5 V to the Vcc pin. Apply 0 V to the Vss pin.
IVcc	IVcc	Output	Connect this pin to Vss via a capacitor (0.1 µF).
Analog power supply input	AVcc, AVss	Input	These are power supply input pins for A-D converter. Connect the AVcc pin to Vcc. Connect the AVss pin to Vss.
Reset input	$\overline{\mathrm{RESET}}$	Input	“L” on this input resets the MCU.
CNVss	CNVss	Input	Connect this pin to Vss via a resistor (approximately 5 kΩ).
MODE	MODE	Input	Connect this pin to Vcc via a resistor (approximately 5 kΩ).
Main clock input	XIN	Input	These pins are provided for the main clock generating circuit input/output. Connect a ceramic resonator or a crystal oscillator between the XIN and XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
Main clock output	XOUT	Output	
$\overline{\mathrm{INT}}$ interrupt input	$\overline{\mathrm{INT0}}$ to $\overline{\mathrm{INT3}}$	Input	These are $\overline{\mathrm{INT}}$ interrupt input pins.
Key input interrupt input	$\overline{\mathrm{KI0}}$ to $\overline{\mathrm{KI3}}$	Input	These are key input interrupt input pins.
Timer X	CNTR0	Input/Output	This is the timer X I/O pin.
	$\overline{\mathrm{CNTR0}}$	Output	This is the timer X output pin.
Timer Y	CNTR1	Input/Output	This is the timer Y I/O pin.
Timer Z	TZOUT	Output	This is the timer Z output pin.
Timer C	TCIN	Input	This is the timer C input pin.
Serial interface	CLK0	Input/Output	This is a transfer clock I/O pin.
	RxD0, RxD1	Input	These are serial data input pins.
	TxD0, TxD10, TxD11	Output	These are serial data output pins.
Reference voltage input	VREF	Input	This is a reference voltage input pin for A-D converter. Connect the VREF pin to Vcc.
A-D converter	AN0 to AN7	Input	These are analog input pins for A-D converter.
I/O port	P00 to P07, P10 to P17, P30 to P33, P37, P45	Input/Output	These are 8-bit CMOS I/O ports. Each port has an input/output select direction register, allowing each pin in that port to be directed for input or output individually. Any port set to input can select whether to use a pull-up resistor or not by program. P10 to P17 also function as LED drive ports.
Input port	P46, P47	Input	These are input only pins.

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.

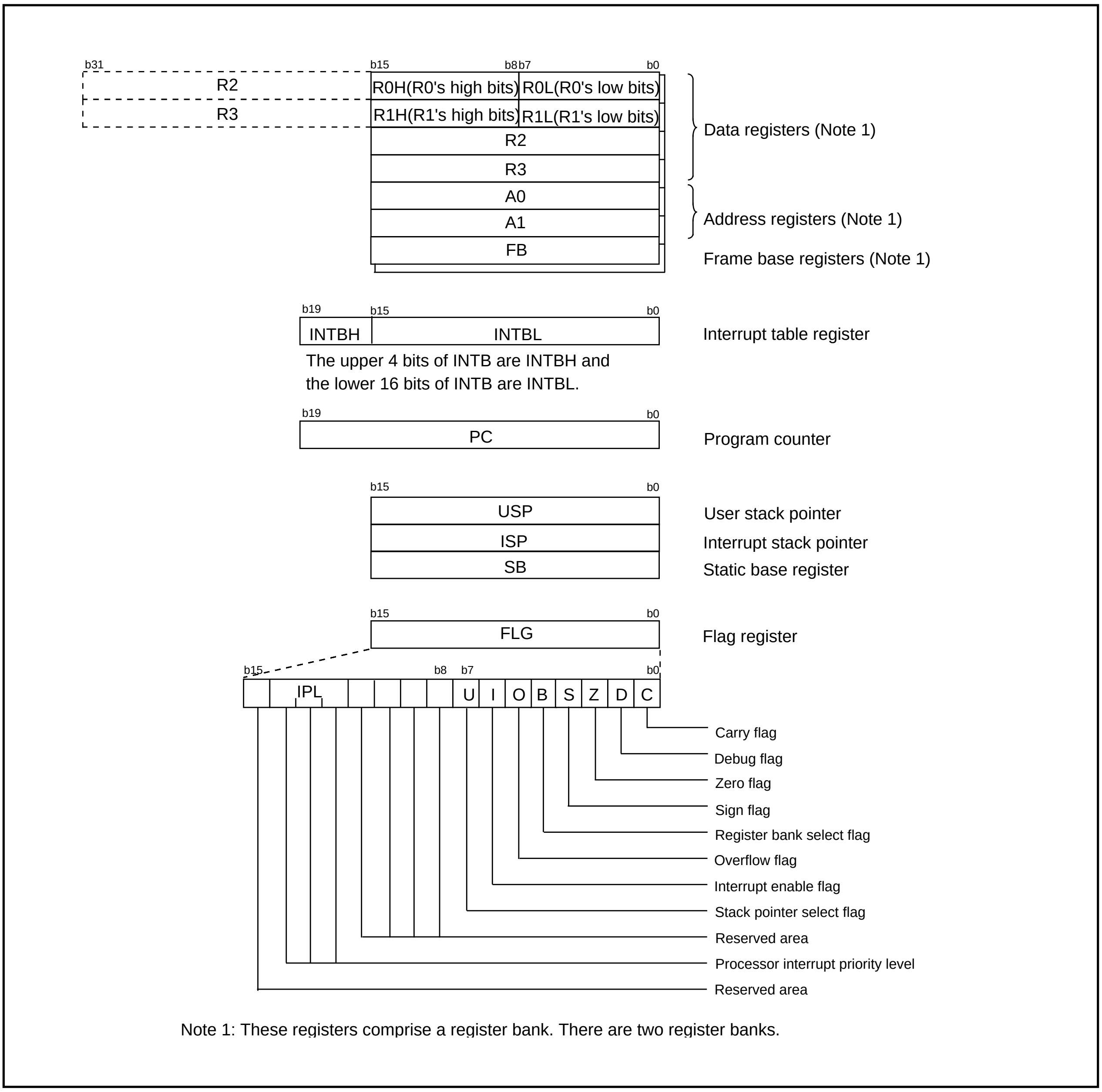


Figure 2.1 Central Processing Unit Register

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

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

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

2.2 Address Registers (A0 and A1)

The register A0 consists of 16 bits, and is used for address register indirect addressing and address register relative addressing. They also are used for transfers and logic/logic operations. A1 is the same as A0. In some instructions, registers A1 and A0 can be combined for use as a 32-bit address register (A1A0).

2.3 Frame Base Register (FB)

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

2.4 Interrupt Table Register (INTB)

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

2.5 Program Counter (PC)

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

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

Stack pointer (SP) comes in two types: USP and ISP, each configured with 16 bits.
Your desired type of stack pointer (USP or ISP) can be selected by the U flag of FLG.

2.7 Static Base Register (SB)

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

2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

2.8.1 Carry Flag (C Flag)

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

2.8.2 Debug Flag (D Flag)

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

2.8.3 Zero Flag (Z Flag)

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

2.8.4 Sign Flag (S Flag)

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

2.8.5 Register Bank Select Flag (B Flag)

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

2.8.6 Overflow Flag (O Flag)

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

2.8.7 Interrupt Enable Flag (I Flag)

This flag enables a maskable interrupt.

Maskable interrupts are disabled when the I flag is “0”, and are enabled when the I flag is “1”. The I flag is cleared to “0” when the interrupt request is accepted.

2.8.8 Stack Pointer Select Flag (U Flag)

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

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

2.8.9 Processor Interrupt Priority Level (IPL)

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

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

2.8.10 Reserved Area

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

3. Memory

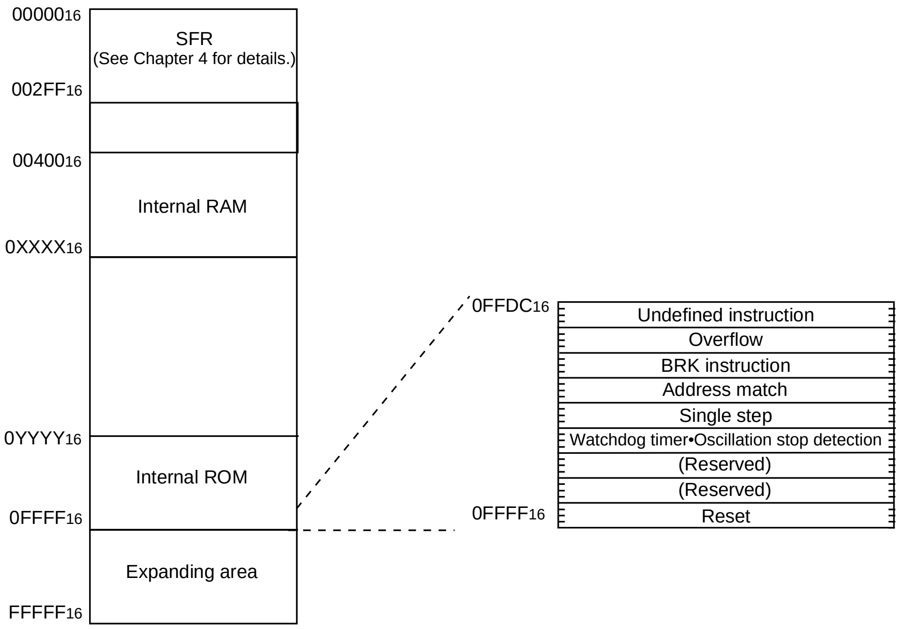
Figure 3.1 is a memory map of this MCU. The address space extends the 1M bytes from address 00000_{16} to $FFFFF_{16}$.

The internal ROM is allocated in a lower address direction beginning with address $0FFFF_{16}$. For example, a 16-Kbyte internal ROM is allocated to the addresses from $0C000_{16}$ to $0FFFF_{16}$.

The fixed interrupt vector table is allocated to the addresses from $0FFDC_{16}$ to $0FFFF_{16}$. Therefore, store the start address of each interrupt routine here.

The internal RAM is allocated in an upper address direction beginning with address 00400_{16}. For example, a 1-Kbyte internal RAM is allocated to the addresses from 00400_{16} to $007FF_{16}$. In addition to storing data, the internal RAM also stores the stack used when calling subroutines and when interrupts are generated.

Special function registers (SFR) are allocated to the addresses from 00000_{16} to $002FF_{16}$. Peripheral function control registers are located here. Of the SFR, any space which has no functions allocated is reserved for future use and cannot be used by users.



Type name	Internal ROM		Internal RAM	
	Size	Address $0YYYY_{16}$	Size	Address $0XXXX_{16}$
R5F21104FP, R5F21104DFP	16K bytes	$0C000_{16}$	1K bytes	$007FF_{16}$
R5F21103FP, R5F21103DFP	12K bytes	$0D000_{16}$	768 bytes	$006FF_{16}$
R5F21102FP, R5F21102DFP	8K bytes	$0E000_{16}$	512 bytes	$005FF_{16}$

Figure 3.1 Memory Map

4. Special Function Register (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	01101000_2
0007_{16}	System clock control register 1	CM1	00100000_2
0008_{16}			
0009_{16}	Address match interrupt enable register	AIER	$XXXXXX00_2$
$000A_{16}$	Protect register	PRCR	$00XXX000_2$
$000B_{16}$			
$000C_{16}$	Oscillation stop detection register	OCD	00000100_2
$000D_{16}$	Watchdog timer reset register	WDTR	XX_{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	00_{16}
0011_{16}			00_{16}
0012_{16}			$X0_{16}$
0013_{16}			
0014_{16}	Address match interrupt register 1	RMAD1	00_{16}
0015_{16}			00_{16}
0016_{16}			$X0_{16}$
0017_{16}			
0018_{16}			
0019_{16}			
$001A_{16}$			
$001B_{16}$			
$001C_{16}$			
$001D_{16}$			
$001E_{16}$	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: The blank areas are reserved and cannot be used by users.

X : Undefined

Address	Register	Symbol	After reset
0040$_{16}$			
0041$_{16}$			
0042$_{16}$			
0043$_{16}$			
0044$_{16}$			
0045$_{16}$			
0046$_{16}$			
0047$_{16}$			
0048$_{16}$			
0049$_{16}$			
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$_{16}$	INT2 interrupt control register	INT2IC	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$_{16}$	INT1 interrupt control register	INT1IC	XXXXX000$_2$
005A$_{16}$	INT3 interrupt control register	INT3IC	XXXXX000$_2$
005B$_{16}$	Timer C interrupt control register	TCIC	XXXXX000$_2$
005C$_{16}$			
005D$_{16}$	INT0 interrupt control register	INT0IC	XX00X000$_2$
005E$_{16}$			
005F$_{16}$			
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 and cannot be used by users.

X : Undefined

Address	Register	Symbol	After reset
0080$_{16}$	Timer Y, Z mode register	TYZMR	00$_{16}$
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$_{16}$			
0089$_{16}$			
008A$_{16}$	Timer Y, Z output control register	TYZOC	00$_{16}$
008B$_{16}$	Timer X mode register	TXMR	00$_{16}$
008C$_{16}$	Prescaler X	PREX	FF$_{16}$
008D$_{16}$	Timer X register	TX	FF$_{16}$
008E$_{16}$	Count source set register	TCSS	00$_{16}$
008F$_{16}$			
0090$_{16}$	Timer C register	TC	00$_{16}$
0091$_{16}$			00$_{16}$
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	00$_{16}$
009B$_{16}$	Timer C control register 1	TCC1	00$_{16}$
009C$_{16}$	Capture register	TM0	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	00001000$_2$
00A5$_{16}$	UART0 transmit/receive control register 1	U0C1	00000010$_2$
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	00001000$_2$
00AD$_{16}$	UART1 transmit/receive control register 1	U1C1	00000010$_2$
00AE$_{16}$	UART1 receive buffer register	U1RB	XX$_{16}$
00AF$_{16}$			XX$_{16}$
00B0$_{16}$	UART transmit/receive control register 2	UCON	00$_{16}$
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 : The blank areas are reserved and cannot be used by users.

X : Undefined

Address	Register	Symbol	After reset
00C0$_{16}$	A-D register	AD	XXXXXXXX$_2$
00C1$_{16}$			XXXXXXXX$_2$
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	00$_{16}$
00D5$_{16}$			
00D6$_{16}$	A-D control register 0	ADCON0	00000XXX$_2$
00D7$_{16}$	A-D control register 1	ADCON1	00$_{16}$
00D8$_{16}$			
00D9$_{16}$			
00DA$_{16}$			
00DB$_{16}$			
00DC$_{16}$			
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$_{16}$			
00E5$_{16}$	Port P3 register	P3	XX$_{16}$
00E6$_{16}$			
00E7$_{16}$	Port P3 direction register	PD3	00$_{16}$
00E8$_{16}$	Port P4 register	P4	XX$_{16}$
00E9$_{16}$			
00EA$_{16}$	Port P4 direction register	PD4	00$_{16}$
00EB$_{16}$			
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$_{16}$			
00F9$_{16}$			
03FA$_{16}$			
00FB$_{16}$			
00FC$_{16}$	Pull-up control register 0	PUR0	00XX0000$_2$
00FD$_{16}$	Pull-up control register 1	PUR1	XXXXXX0X$_2$
00FE$_{16}$	Port P1 drivability control register	DRR	00$_{16}$
00FF$_{16}$			
01B3$_{16}$	Flash memory control register 4	FMR4	0100000X$_2$
01B4$_{16}$			
01B5$_{16}$	Flash memory control register 1	FMR1	0100XX0X$_2$
01B6$_{16}$			
01B7$_{16}$	Flash memory control register 0	FMR0	XX000001$_2$

Note 1: The blank areas are reserved and cannot be used by users.

X : Undefined

5. Reset

There are three types of resets: a hardware reset, a software reset, and an watchdog timer reset.

5.1 Hardware Reset

A reset is applied using the $\overline{\text{RESET}}$ pin. When an “L” signal is applied to the $\overline{\text{RESET}}$ pin while the power supply voltage is within the recommended operating condition, the pins are initialized (see Table 5.1 “Pin Status When $\overline{\text{RESET}}$ Pin Level is 'L'”). When the input level at the $\overline{\text{RESET}}$ pin is released from “L” to “H”, the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. Figure 5.1 shows the CPU register status after reset and figure 5.2 shows the reset sequence. The internal RAM is not initialized. If the $\overline{\text{RESET}}$ pin is pulled “L” while writing to the internal RAM, the internal RAM becomes indeterminate. Figures 5.3 to 5.4 show the reset circuit example. Refer to Chapter 4, “Special Function Register (SFR)” for the status of SFR after reset.

- When the power supply is stable

(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin.
(2) Wait 500 µs.
(3) Apply an “H” signal to the $\overline{\text{RESET}}$ pin.

- Power on

(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin.
(2) Let the power supply voltage increase until it meets the recommended operating condition.
(3) Wait td(P-R) or more until the internal power supply stabilizes.
(4) Wait 500 µs.
(5) Apply an “H” signal to the $\overline{\text{RESET}}$ pin.

Table 5.1 Pin Status When $\overline{\text{RESET}}$ Pin Level is “L”

Pin name	Status
	CNVss = Vss
P0	Input port
P1	Input port
P30 to P33, P37	Input port
P45 to P47	Input port

5.2 Software Reset

When the PM03 bit in the PM0 register is set to “1” (microcomputer reset), the microcomputer has its pins, CPU, and SFR initialized. Then the program is executed starting from the address indicated by the reset vector. Some SFRs are not initialized by the software reset. Refer to Chapter 4, “SFR.”

5.3 Watchdog Timer Reset

Where the PM12 bit in the PM1 register is “1” (reset when watchdog timer underflows), the microcomputer initializes its pins, CPU and SFR if the watchdog timer underflows. Then the program is executed starting from the address indicated by the reset vector.
Some SFRs are not initialized by the watchdog timer reset. Refer to Chapter 4, “SFR.”

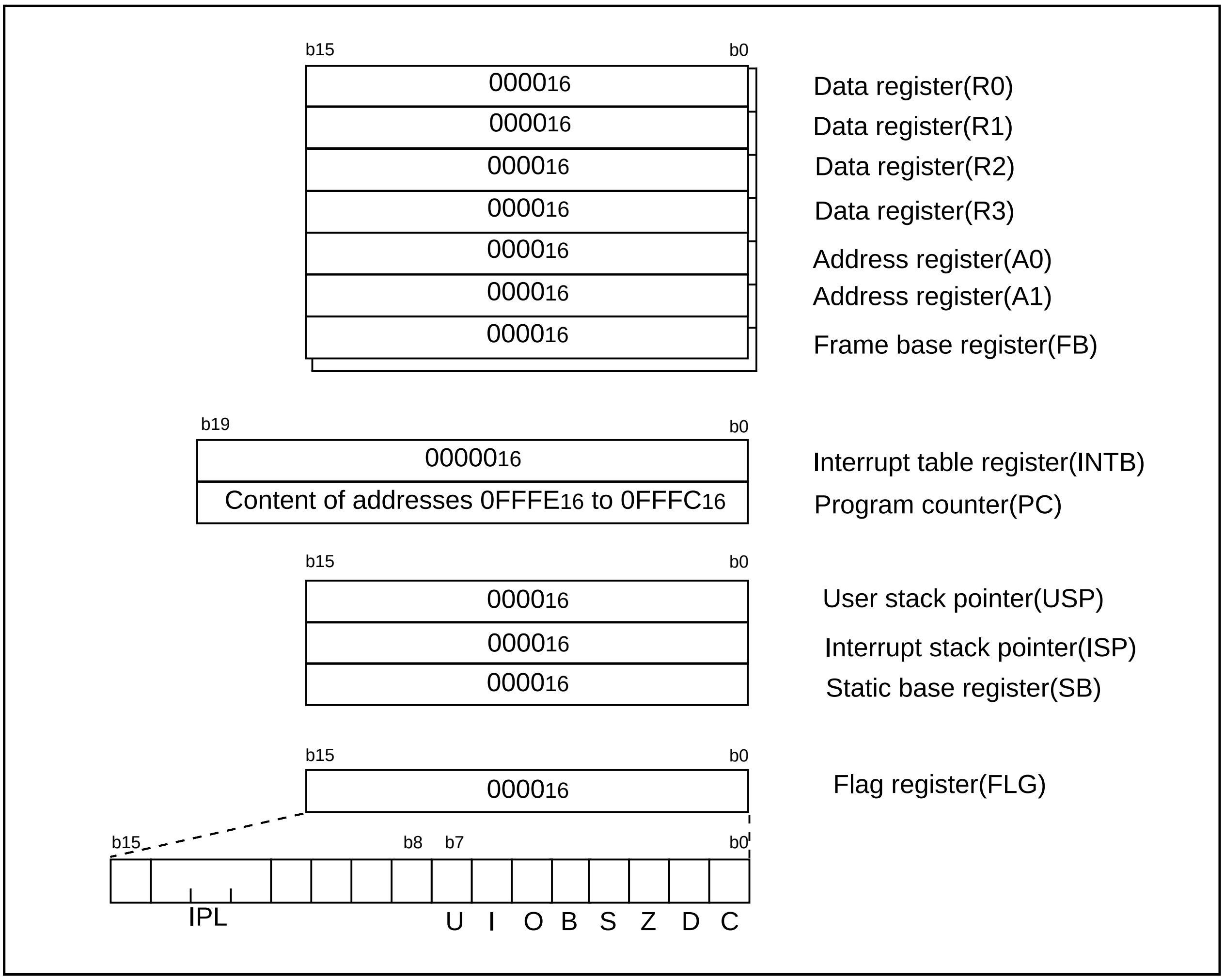


Figure 5.1 CPU Register Status After Reset

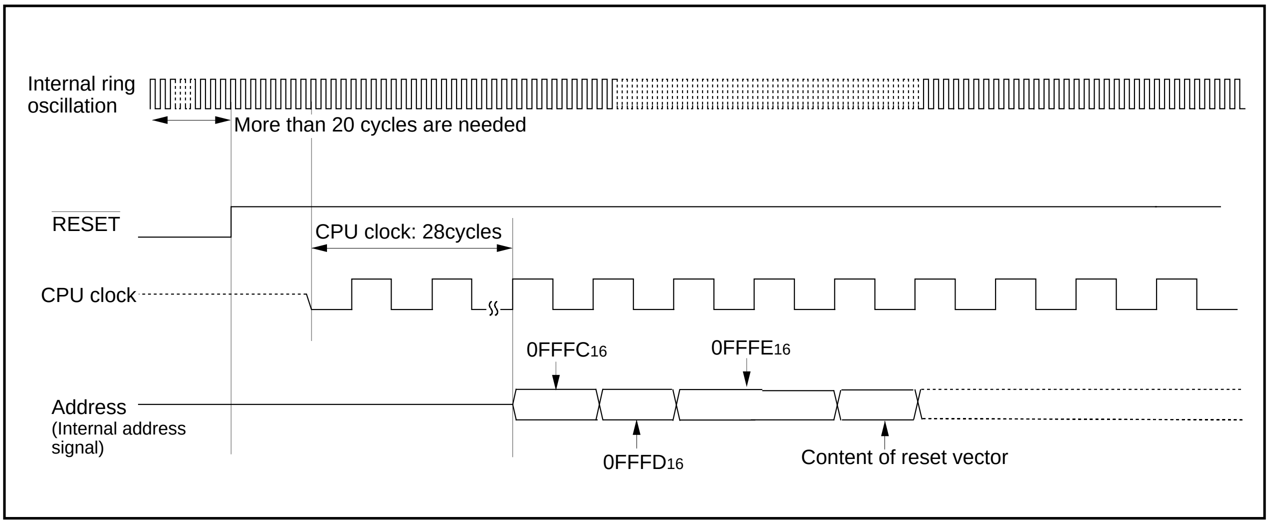


Figure 5.2 Reset Sequence

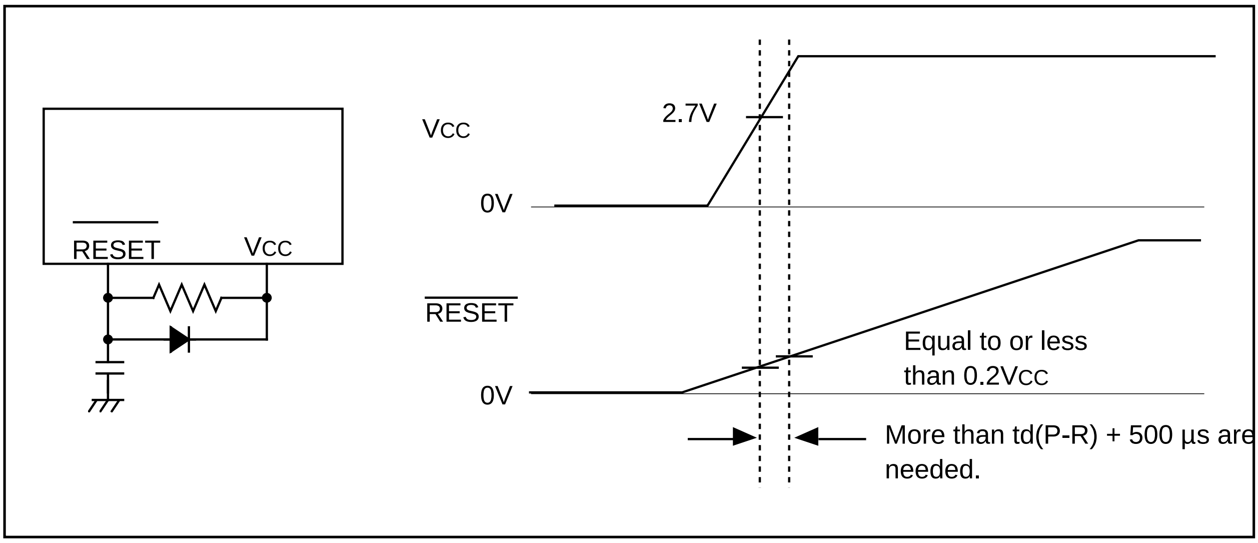

RESET
Vcc
Vcc
2.7V
0V
RESET
0V
Equal to or less than 0.2Vcc
More than td(P-R) + 500 µs are needed.

Figure 5.3 Example Reset Circuit

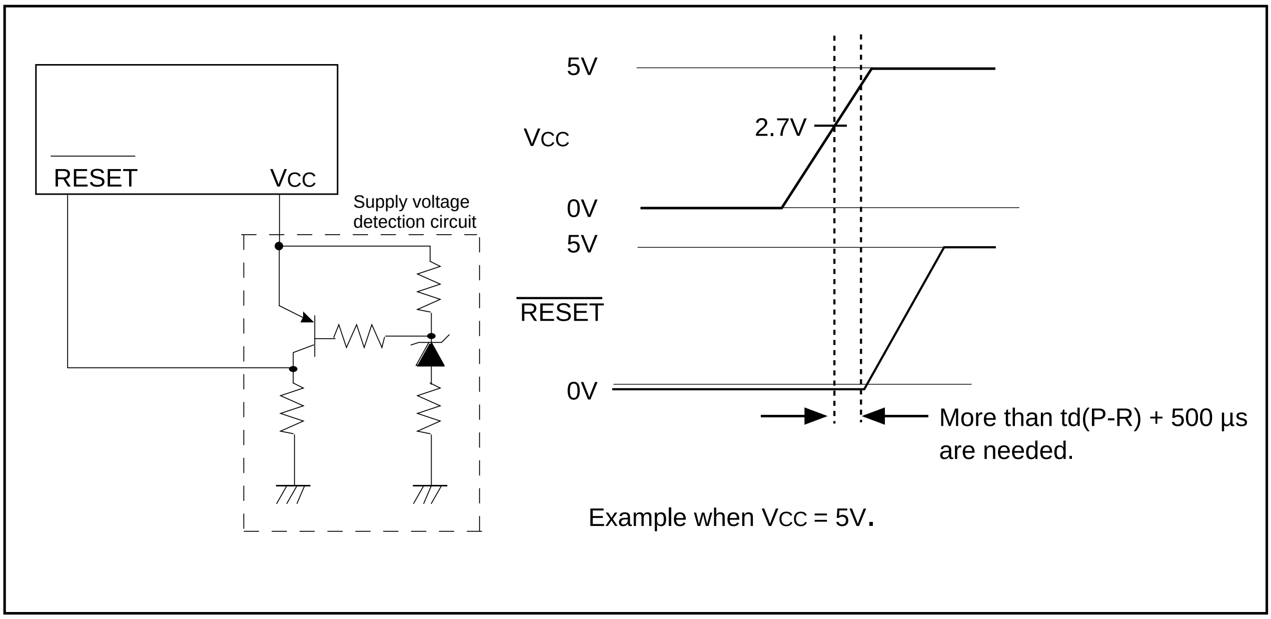

RESET
Vcc
Supply voltage detection circuit
5V
Vcc
2.7V
0V
5V
RESET
0V
More than td(P-R) + 500 µs are needed.
Example when Vcc = 5V.

Figure 5.4 Example Reset Circuit (Voltage Check Circuit)

6. Clock Generation Circuit

The clock generation circuit contains two oscillator circuits as follows:

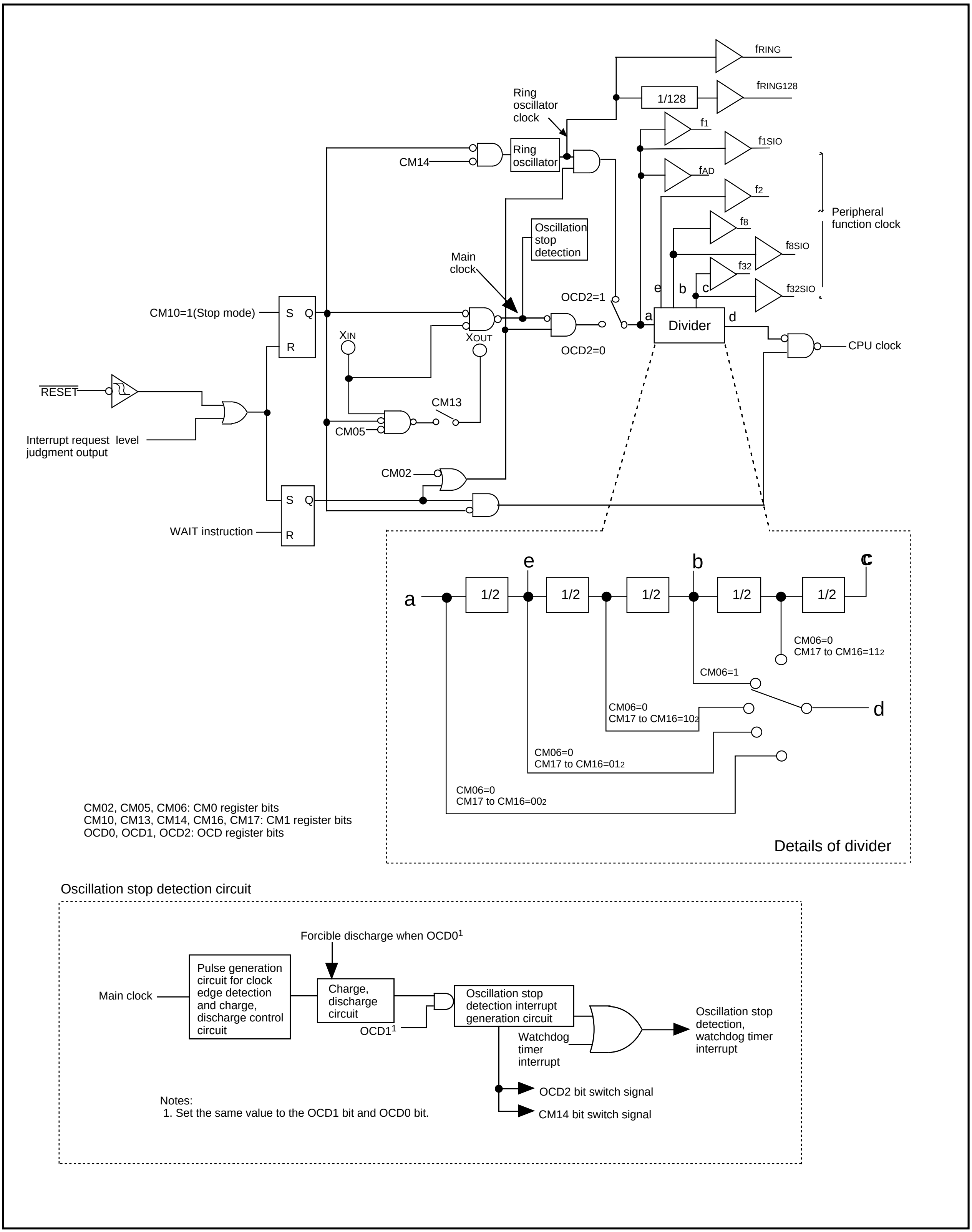
- Main clock oscillation circuit
- Ring oscillator (oscillation stop detect function)

Table 6.1 lists the clock generation circuit specifications. Figure 6.1 shows the clock generation circuit. Figures 6.2 and 6.3 show the clock-related registers.

Table 6.1 Clock Generation Circuit Specifications

Item	Main clock oscillation circuit	Ring oscillator
Use of clock	• CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when the main clock stops oscillating	• CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when the main clock stops oscillating
Clock frequency	0 to 16 MHz	About 125 kHz
Usable oscillator	• Ceramic oscillator • Crystal oscillator	—
Pins to connect oscillator	XIN, XOUT[1]	Note [1]
Oscillation stop, restart function	Present	Present
Oscillator status after reset	Stopped	Oscillating
Other	Externally derived clock can be input	—

Notes:
1. Can be used as $P4_6$ and $P4_7$ when the ring oscillator clock is used for CPU clock while the main clock oscillation circuit is not used.

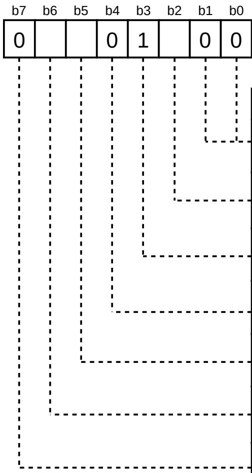


CM02, CM05, CM06: CM0 register bits
CM10, CM13, CM14, CM16, CM17: CM1 register bits
OCD0, OCD1, OCD2: OCD register bits

Notes:
1. Set the same value to the OCD1 bit and OCD0 bit.

Figure 6.1 Clock Generation Circuit

System clock control register 0[1]



Symbol
CM0

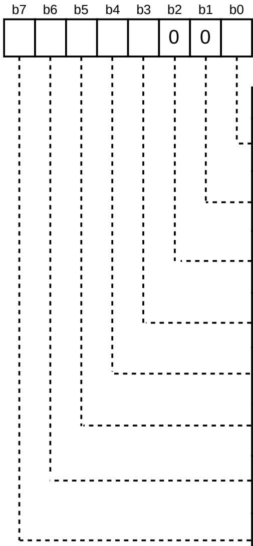
Address
0006₁₆

After reset
68₁₆

Bit symbol	Bit name	Function	RW
— (b1-b0)	Reserved bit	Must set to “0”	RW
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral function clock in wait mode [6] 1 : Stop peripheral function clock in wait mode	RW
— (b3)	Reserved bit	Must set to “1”	RW
— (b4)	Reserved bit	Must set to “0”	RW
CM05	Main clock (XCIN-XCOUT) stop bit[2, 4]	0 : On 1 : Off[3]	RW
CM06	Main clock division select bit 0[5]	0 : CM16 and CM17 valid 1 : Divide-by-8 mode	RW
— (b7)	Reserved bit	Must set to “0”	RW

Notes
1: Set the PRC0 bit of PRCR register to “1” (write enable) before writing to this register.
2: The CM05 bit is provided to stop the main clock when the ring oscillator mode is selected. This bit cannot be used for detection as to whether the main clock stopped or not. To stop the main clock, the following setting is required:
(1) Set the CM06 bit to “1” (divide-by-8 mode)
(2) Set the OCD0 and OCD1 bits in the OCD register to “00 $_2$” (disabling oscillation stop detection function).
(3) Set the OCD2 bit to “1” (selecting ring oscillator).
3: During external clock input, only the clock oscillation buffer is turned off and clock input is accepted.
4: When the CM05 bit is set to “1” (main clock stop), P4 $_6$ and P4$_7$ can be used as input ports.
5: When entering stop mode from high or middle speed mode, the CM06 bit is set to “1” (divide-by-8 mode).
6: During ring oscillator mode, this bit must be set to “0” (peripheral clock turned on when in wait mode).

System clock control register 1[1]



Symbol
CM1

Address
0007$_{16}$

After reset
20$_{16}$

Bit symbol	Bit name	Function	RW
CM10	All clock stop control bit[4]	0 : Clock on 1 : All clocks off (stop mode)	RW
— (b1)	Reserved bit	Must set to “0”	RW
— (b2)	Reserved bit	Must set to “0”	RW
CM13	Port XIN-XOUT switch bit	0 : Input port P4$_6$, P4$_7$ 1 : XIN-XOUT pin	RW
CM14	Ring oscillation stop bit	0 : Ring oscillator on 1 : Ring oscillator off[5]	RW
CM15	XIN-XOUT drive capability select bit[2]	0 : LOW 1 : HIGH	RW
CM16	Main clock division select bit 1[3]	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			RW

Notes:
1: Write to this register after setting the PRC0 bit of PRCR register to “1” (write enable).
2: When entering stop mode from high or middle speed mode, the CM15 bit is set to “1” (drive capability high).
3: Effective when the CM06 bit is “0” (CM16 and CM17 bits enable).
4: If the CM10 bit is “1” (stop mode), the internal feedback resistor becomes ineffective.
5: The CM14 bit can be set to “1” (ring oscillator off) if the OCD2 bit=0 (selecting main clock). When the OCD2 bit is set to “1” (selecting ring oscillator clock), the CM14 bit is set to “0” (ring oscillator on). This bit remains unchanged when “1” is written.

Figure 6.2 CM0 Register and CM1 Register

Oscillation stop detection register[1]

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

Symbol	Address	After reset
OCD	$000C_{16}$	04_{16}

Bit symbol	Bit name	Function	RW
OCD0	Oscillation stop detection enable bit	b1 b0 0 0: The function is disabled[4] 0 1: Avoid this setting 1 0: Avoid this setting 1 1: The function is enabled[7]	RW
OCD1			
OCD2	System clock select bit[6]	0: Select main clock[7] 1: Select ring oscillator clock[2]	RW
OCD3	Clock monitor bit[3, 5]	0: Main clock on 1: Main clock off	RO
— (b7-b4)	Reserved bit	Must set to "0"	RW

Notes:

1. Set the PRC0 bit in the PRCR register to “1” (write enable) before rewriting this register.
2. The OCD2 bit is set to “1” (selecting ring oscillator clock) automatically if a main clock oscillation stop is detected while the OCD1 to OCD0 bits are set to “11_2” (oscillation stop detection function disabled). If the OCD3 bit is set to “1” (main clock stop), the OCD2 bit remains unchanged when trying to write “0” (selecting main clock).
3. The OCD3 bit is enabled when the OCD1 to OCD0 bits are set to “11_2” (oscillation stop detection function enabled). Read the OCD3 bit several times with the oscillation stop detection interrupt processing program to determine the main clock state.
4. The OCD1 to OCD0 bits must be set to “00_2” (oscillation stop detection function disabled)before entering stop mode and ring oscillator mode (main clock stops).
5. The OCD3 bit remains set to “0” (main clock on) if the OCD1 to OCD0 bits are set to “00_2”.
6. The CM14 bit goes to “0” (ring oscillator on) if the OCD2 bit is set to “1” (selecting ring oscillator clock).
7. Refer to Figure 6.8 “switching clock source from ring oscillator to main clock” for the switching procedure when the main clock re-oscillates after detecting an oscillation stop.

Figure 6.3 OCD Register

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

6.1 Main Clock

This clock is supplied by a main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillator circuit is configured by connecting a resonator between the XIN and XOUT pins. The main clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillator circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 6.4 shows examples of main clock connection circuit. After reset, the main clock is turned off.

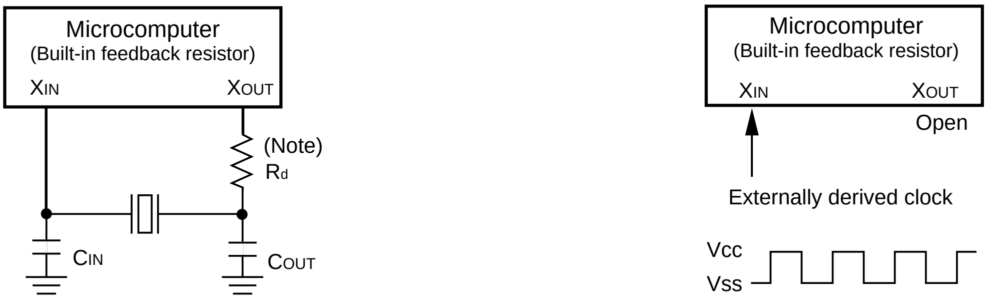
The main clock starts oscillating when the CM05 bit in the CM0 register is set to “0” (main clock on) after setting the CM13 bit in the CM1 register to “1” (XIN- XOUT pin).

To use the main clock for the CPU clock, set the OCD2 bit in the OCD register to “0” (selecting main clock) after the main clock becomes oscillating stably.

The power consumption can be reduced by setting the CM05 bit in the CM0 register to “1” (main clock off) if the OCD2 bit is set to “1” (selecting ring oscillator clock).

Note that if an externally generated clock is fed into the XIN pin, the main clock cannot be turned off by setting the CM05 bit to “1”. If necessary, use an external circuit to turn off the clock.

During stop mode, all clocks including the main clock are turned off. Refer to Section 6.3, “Power Control.”



Note: 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 6.4 Examples of Main Clock Connection Circuit

6.2 Ring Oscillator Clock

This clock, approximately 100 kHz, is supplied by a ring oscillator. This clock is used as the clock source for the CPU clock, peripheral function clock, f_{RING}, and $f_{RING128}$.
After reset, the ring oscillator clock divided by 8 is selected for the CPU clock.
To use the main clock for the CPU clock, set the OCD2 in the OCD register to “0” (selecting main clock) after the main clock becomes oscillating stably. If the main clock stops oscillating when the OCD1 to OCD0 bits in the OCD register is “11_2” (oscillation stop detection function enabled), the ring oscillator automatically starts operating, supplying the necessary clock for the microcomputer.
The frequency of the ring oscillator varies depending on the supply voltage and the operation ambient temperature. The application products must be designed with sufficient margin to accommodate the frequency range.

6.3 CPU Clock and Peripheral Function Clock

There are two type clocks: CPU clock to operate the CPU and peripheral function clock to operate the peripheral functions. Also refer to “Figure 6.1 Clock Generating Circuit”.

6.3.1 CPU Clock

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

The clock source for the CPU clock can be chosen to be the main clock or ring oscillator clock.

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

After reset, the ring oscillator clock divided by 8 provides the CPU clock. When the clock source for the CPU clock is switched over, set the CM06 bit to “1” (divide-by-8 mode) before changing the OCD2 bit. Note that when entering stop mode from high or middle speed mode, the CM06 bit is set to “1” (divide-by-8 mode).

6.3.2 Peripheral Function Clock (f_1, f_2, f_8, f_{32}, f_{AD}, f_{1SIO}, f_{8SIO}, f_{32SIO}, f_{RING}, $f_{RING128}$)

These are operating clocks for the peripheral functions.

Of these, fi (i=1, 2, 8, 32) is derived from the main clock or ring oscillator clock by dividing them by i. The clock fi is used for timers X, Y, Z and C.

The clock fj$_{SIO}$ (j=1, 8, 32) is derived from the main clock or ring oscillator clock by dividing them by j. The clock fj$_{SIO}$ is used for serial I/O.

The f_{AD} clock is produced from the main clock is used for the A-D converter.

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to “1” (peripheral function clock turned off during wait mode), the clocks fi, fj$_{SIO}$, and f_{AD} are turned off.

6.3.3 f_{RING} and $f_{RING128}$

These are operating clocks for the peripheral functions.

The f_{RING} runs at the same frequency as the ring oscillator, and can be used as the source for the timer Y. The $f_{RING128}$ is derived from the f_{RING} by dividing it by 128, and can used for the timer C.

When the WAIT instruction is executed, the clocks f_{RING} and $f_{RING128}$ are not turned off.

6.4 Power Control

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

6.4.1 Normal Operation Mode

Normal operation mode is further classified into three modes.

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

Before the clock sources for the CPU clock can be switched over, the new clock source to which switched must be oscillating stably. If the new clock source is the main clock, allow a sufficient wait time in a program until it becomes oscillating stably.

- **High-speed Mode**

  The main clock divided by 1 undivided provides the CPU clock. If the CM14 bit is set to “0” (ring oscillator on), the f_{RING} is used as the count source for timer Y.

- **Medium-speed Mode**

  The main clock divided by 2, 4, 8 or 16 provides the CPU clock. If the CM14 bit is set to “0” (ring oscillator on), the f_{RING} is used as the count source for timer Y.

- **Ring Oscillator Mode**

  The ring oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The ring oscillator clock is also the clock source for the peripheral function clocks. Set CM06 bit to “1” (divided by 8 mode) when returning to high-speed and medium-speed mode.

Table 6.2 Setting Clock Related Bit and Modes

Modes		OCD register	CM1 register	CM0 register	
		OCD2	CM17, CM16	CM06	CM05
High-speed mode		0	00_2	0	0
Medium-speed mode	divided by 2	0	01_2	0	0
	divided by 4	0	10_2	0	0
	divided by 8	0	—	1	0
	divided by 16	0	11_2	0	0
Ring oscillator mode	no division	1	00_2	0	0 or 1
	divided by 2	1	01_2	0	0 or 1
	divided by 4	1	10_2	0	0 or 1
	divided by 8	1	—	1	0 or 1
	divided by 16	1	11_2	0	0 or 1

6.4.2 Wait Mode

In wait mode, the CPU clock is turned off, so are the CPU and the watchdog timer because both are operated by the CPU clock. Because the main clock and ring oscillator clock both are on, the peripheral functions using these clocks keep operating.

• Peripheral Function Clock Stop Function

If the CM02 bit is “1” (peripheral function clocks turned off during wait mode), the f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{8SIO}, f_{32SIO}, and f_{AD} clocks are turned off when in wait mode, with the power consumption reduced that much.

• Entering Wait Mode

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

• Pin Status During Wait Mode

The status before wait mode is retained.

• Exiting Wait Mode

The microcomputer is moved out of wait mode by a hardware reset or peripheral function interrupt. When using a hardware reset to exit wait mode, set the ILVL2 to ILVL0 bits for the peripheral function interrupts to “000_2” (interrupts disabled) before executing the WAIT instruction.

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

Table 6. 3 lists the interrupts to exit wait mode and the usage conditions.

When using a peripheral function interrupt to exit wait mode, set up the following before executing the WAIT instruction.

1. In the ILVL2 to ILVL0 bits in the interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit wait mode.
   Also, for all of the peripheral function interrupts not used to exit wait mode, set the ILVL2 to ILVL0 bits to “000_2” (interrupt disable).
2. Set the I flag to “1”.
3. Enable the peripheral function whose interrupt is to be used to exit wait mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt sequence is executed.

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

Table 6.3 Interrupts to Exit Wait Mode and Usage Conditions

Interrupt	CM02=0	CM02=1
Serial I/O interrupt	Can be used when operating with internal or external clock	Can be used when operating with external clock
Key input interrupt	Can be used	Can be used
A-D conversion interrupt	Can be used in one-shot mode	— (Do not use)
Timer X interrupt	Can be used in all modes	Can be used in event counter mode
Timer Y interrupt	Can be used in all modes	Can be used when counting inputs from CNTR1 pin in timer mode
$\overline{INT}$ interrupt	Can be used	Can be used ($\overline{INT0}$ and $\overline{INT3}$ can be used if there is no filter.

6.4.3 Stop Mode

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

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

- Key interrupt
- $\overline{\text{INT}}$ interrupt ($\overline{\text{INT}}$0 and $\overline{\text{INT}}$3 can be used only when there is no filter.)
- Timer X interrupt (when counting external pulses in event counter mode)
- Timer Y interrupt (when counting inputs from CNTR1 pin in timer mode)
- Serial I/O interrupt (when external clock is selected)

• Entering Stop Mode

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

Before entering stop mode, set the OCD1 to OCD0 bits to “00_2” (oscillation stop detection function disable).

• Pin Status in Stop Mode

The status before wait mode is retained.

• Exiting Stop Mode

The microcomputer is moved out of stop mode by a hardware reset or peripheral function interrupt.

When using a hardware reset to exit stop mode, set the ILVL2 to ILVL0 bits for the peripheral function interrupts to “000_2” (interrupts disabled) before setting the CM10 bit to “1”.

When using a peripheral function interrupt to exit stop mode, set up the following before setting the CM10 bit to “1”.

1. In the ILVL2 to ILVL0 bits in the interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit stop mode.
   Also, for all of the peripheral function interrupts not used to exit stop mode, set the ILVL2 to ILVL0 bits to “000_2”.
2. Set the I flag to “1”.
3. Enable the peripheral function whose interrupt is to be used to exit stop mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt sequence is executed.

The main clock divided by 8 of the clock which is used right before stop mode is used for the CPU clock when exiting stop mode by a peripheral function interrupt.

Figure 6.5 shows the state transition from normal operation mode to stop mode and wait mode. Figure 6.6 shows the state transition in normal operation mode.

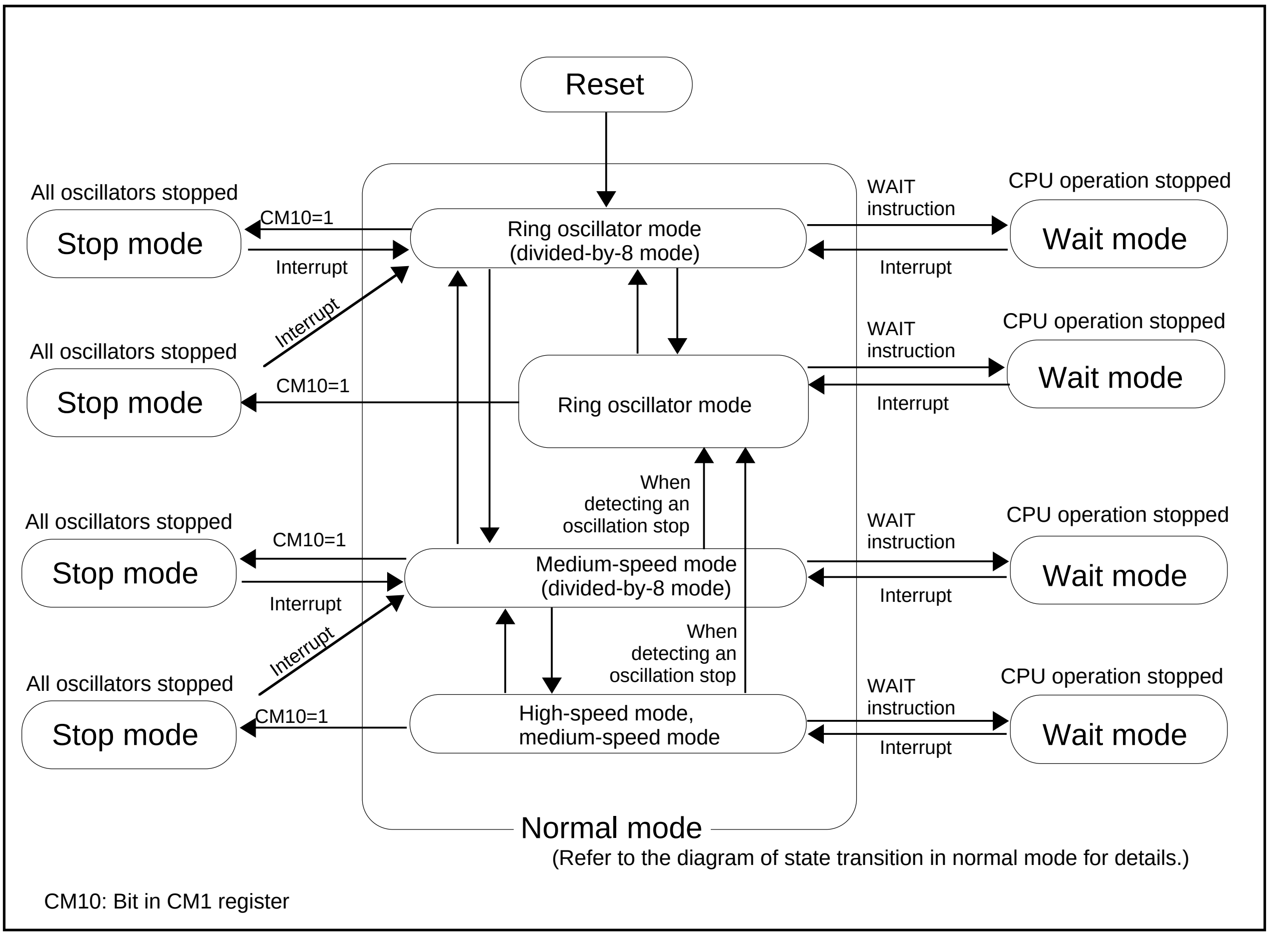
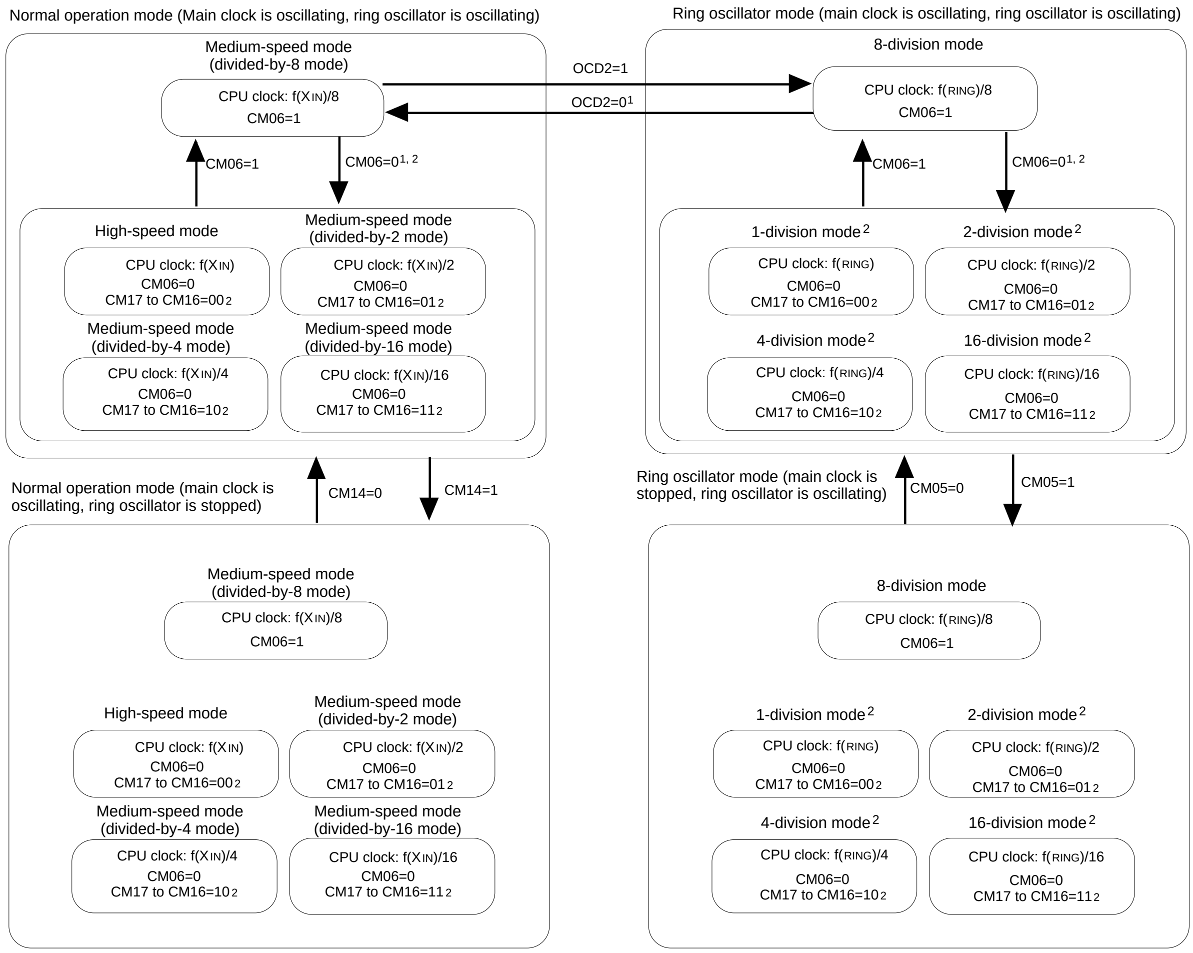


Figure 6.5 State Transition to Stop Mode and Wait Mode



Notes:
1. Switch clock after oscillation of main clock is sufficiently stable.
2. Change the CM17 to CM16 bits before changing CM06 bit.

CM05, CM06: Bits in CM0 register
CM14, CM16, CM17: Bits in CM1 register
OCD2: Bit in OCD register

Figure 6.6 State Transition in Normal Operation Mode

6.5 Oscillation Stop Detection Function

The oscillation stop detection function is such that main clock oscillation circuit stop is detected. The oscillation stop detection function can be enabled and disabled by the OCD1 to OCD0 bits in the OCD register.
Table 6.4 lists the specifications of the oscillation stop detection function.

Where the main clock corresponds to the CPU clock source and the OCD1 to OCD0 bits are “11_2” (oscillation stop detection function enabled), the system is placed in the following state if the main clock comes to a halt:

- The ring oscillator starts oscillation, and the ring oscillator clock becomes the clock source for CPU clock and peripheral functions in place of the main clock
- OCD register OCD2 bit = 1 (selecting ring oscillator clock)
- OCD register OCD3 bit = 1 (main clock stopped)
- CM1 register CM14 bit = 0 (ring oscillator oscillating)
- Oscillation stop detection interrupt request occurs

Table 6.4 Oscillation Stop Detection Function Specifications

Item	Specification
Oscillation stop detectable clock and frequency bandwidth	$f(X_{IN}) \geq 2$ MHz
Enabling condition for oscillation stop detection function	Set OCD1 to OCD0 bits to “11_2” (oscillation stop detection function enabled)
Operation at oscillation stop detection	Oscillation stop detection interrupt occurs

6.5.1 How to Use Oscillation Stop Detection Function

- The oscillation stop detection interrupt shares the vector with the watchdog timer interrupt. If the oscillation stop detection and watchdog timer interrupts both are used, the interrupt source must be determined. Figure 6.7 shows to determine the interrupt source with the oscillation stop detection interrupt processing program.
- Where the main clock re-oscillated after oscillation stop, the clock source for the CPU clock and peripheral functions must be switched to the main clock in the program.
  Figure 6.8 shows the procedure for switching the clock source from the ring oscillator to the main clock.
- To enter wait mode while using the oscillation stop detection function, set the CM02 bit to “0” (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop detection function is provided in preparation for main clock stop due to external factors, set the OCD1 to OCD0 bits to “00_2” (oscillation stop detection function disabled) where the main clock is stopped or oscillated in the program, that is where the stop mode is selected or the CM05 bit is altered.
- This function cannot be used if the main clock frequency is 2 MHz or less. In that case, set the OCD1 to OCD0 bits to “00_2” (oscillation stop detection function disabled).

Table 6.5 Determination of Interrupt Source (Oscillation Stop Detection or Watchdog Timer Interrupt)

Generated Interrupt Source	Bit showing interrupt source
Oscillation stop detection ((a) or (b))	(a) The OCD3 bit in the OCD register = 1 (b) The OCD1 to OCD0 bits in the OCD register = 11_2 and the OCD2 bit = 1
Watchdog timer	The D43 bit in the D4INT register = 1

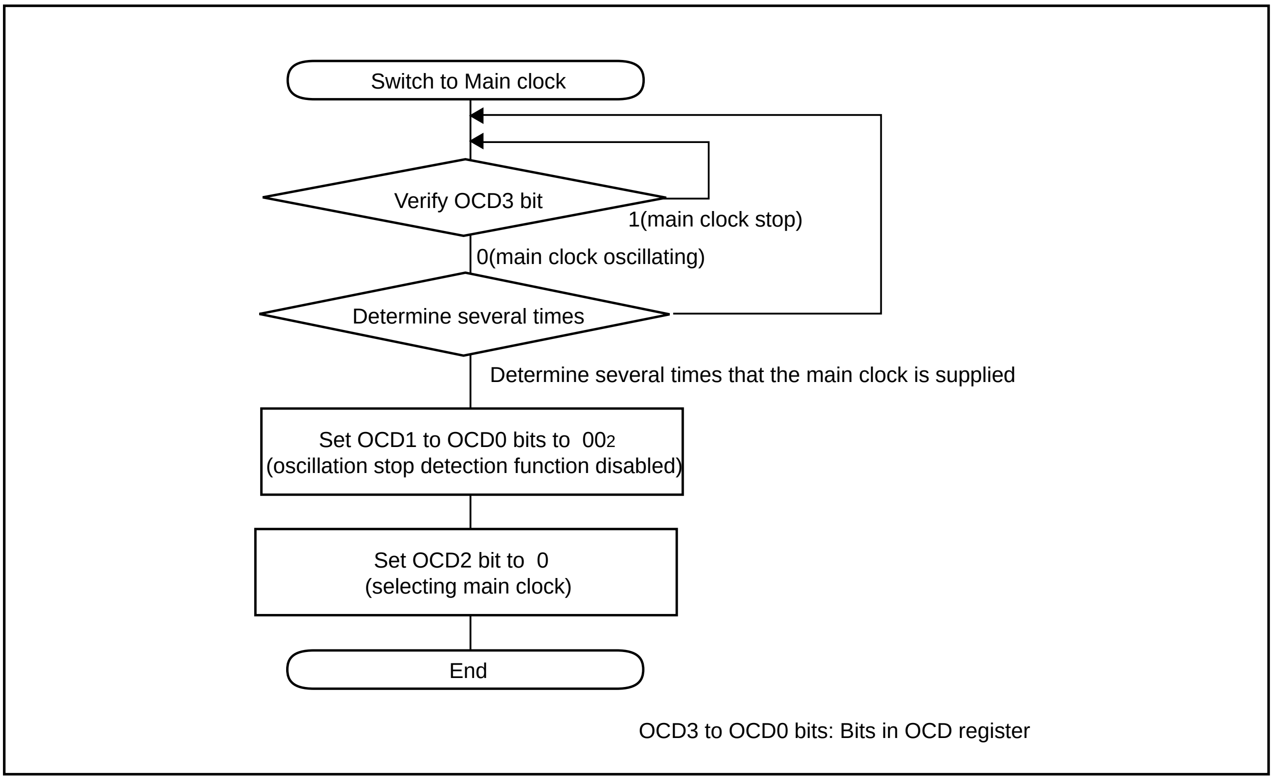

Switch to Main clock
Verify OCD3 bit
1(main clock stop)
0(main clock oscillating)
Determine several times
Determine several times that the main clock is supplied
Set OCD1 to OCD0 bits to 00_2
(oscillation stop detection function disabled)
Set OCD2 bit to 0
(selecting main clock)
End
OCD3 to OCD0 bits: Bits in OCD register

Figure 6.7 Switching Clock Source From Ring Oscillator to Main Clock

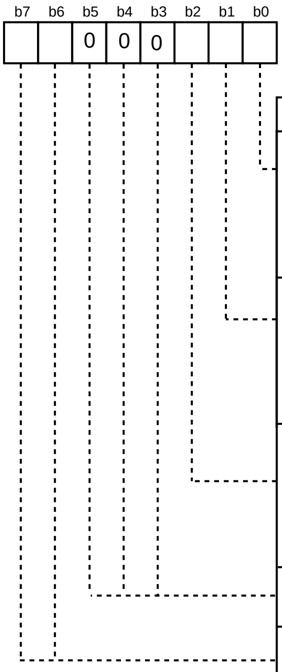
7. Protection

In the event that a program runs out of control, this function protects the important registers so that they will not be rewritten easily. Figure 7.1 shows the PRCR register. The following lists the registers protected by the PRCR register.

- Registers protected by PRC0 bit: CM0, CM1, and OCD registers
- Registers protected by PRC1 bit: PM0 and PM1 registers
- Registers protected by PRC2 bit: PD0 register

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

Protect register



Symbol	Address	After reset
PRCR	$000A_{16}$	$00XXX000_2$

Bit symbol	Bit name	Function	RW
PRC0	Protect bit 0	Enable write to CM0, CM1, OCD registers 0 : Write protected 1 : Write enabled	RW
PRC1	Protect bit 1	Enable write to PM0, PM1 registers 0 : Write protected 1 : Write enabled	RW
PRC2	Protect bit 2	Enable write to PD0 register 0 : Write protected 1 : Write enabled[1]	RW
— (b5-b3)	Reserved bit	When write, must set to "0"	RW
— (b7-b6)	Reserved bit	When read, its content is “0”.	RO

Notes:
1. The PRC2 bit is set to “0” by writing to any address after setting it to “1”. Other bits are not set to “0” by writing to any address, and must therefore be set to “0” in a program.

Figure 7.1 PRCR Register

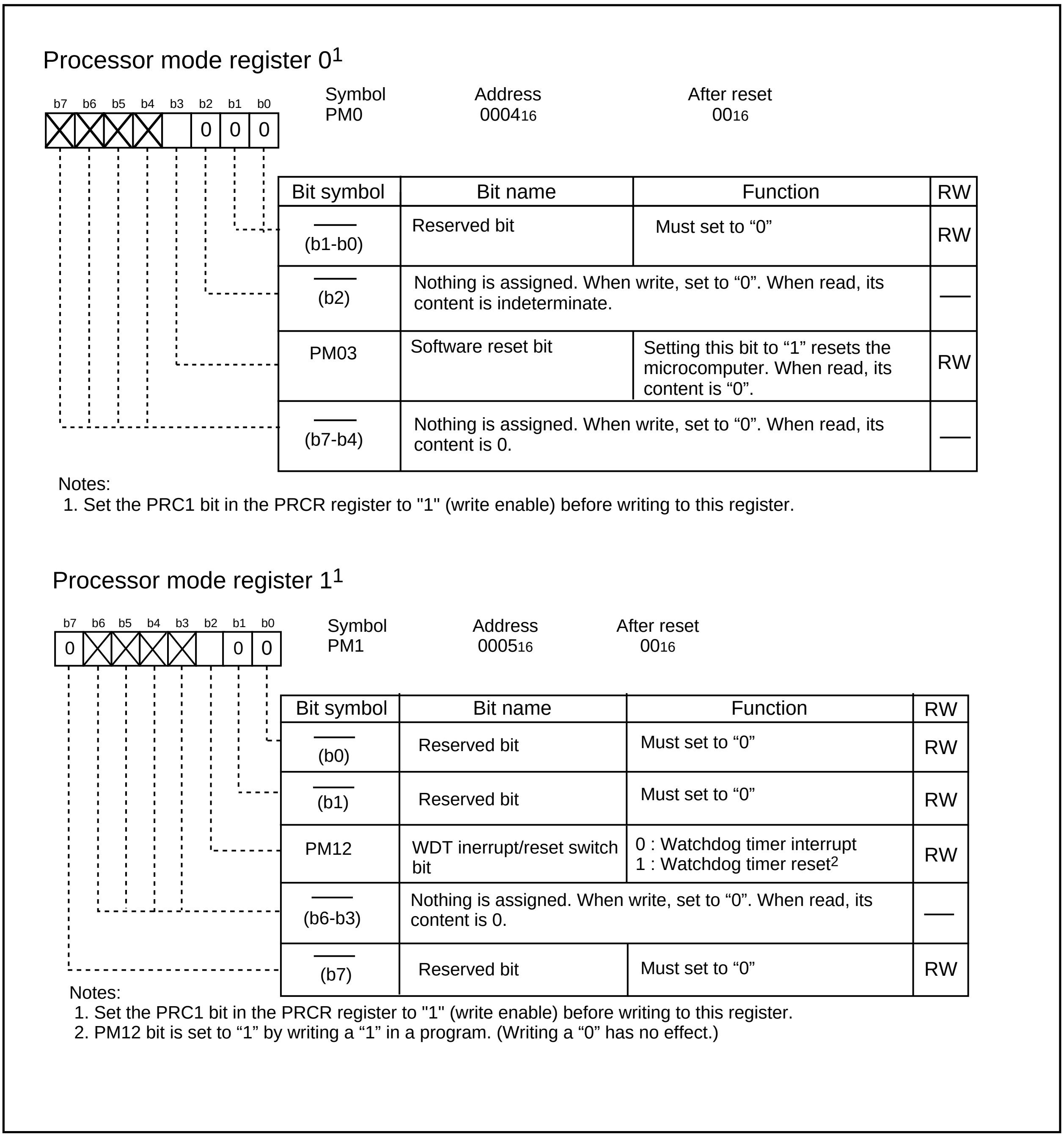
8. Processor Mode

8.1 Types of Processor Mode

The processor mode is single-chip mode. Table 8.1 shows the features of the processor mode. Figure 8.1 shows the PM0 and PM1 register.

Table 8.1 Features of Processor Mode

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



Processor mode register 0[1]

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

Symbol PM0 Address 0004$_{16}$ After reset 00$_{16}$

Bit symbol	Bit name	Function	RW
(b1-b0)	Reserved bit	Must set to “0”	RW
(b2)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
PM03	Software reset bit	Setting this bit to “1” resets the microcomputer. When read, its content is “0”.	RW
(b7-b4)	Nothing is assigned. When write, set to “0”. When read, its content is 0.		—

Notes:
1. Set the PRC1 bit in the PRCR register to "1" (write enable) before writing to this register.

Processor mode register 1[1]

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

Symbol PM1 Address 0005$_{16}$ After reset 00$_{16}$

Bit symbol	Bit name	Function	RW
(b0)	Reserved bit	Must set to “0”	RW
(b1)	Reserved bit	Must set to “0”	RW
PM12	WDT inerrupt/reset switch bit	0 : Watchdog timer interrupt 1 : Watchdog timer reset[2]	RW
(b6-b3)	Nothing is assigned. When write, set to “0”. When read, its content is 0.		—
(b7)	Reserved bit	Must set to “0”	RW

Notes:
1. Set the PRC1 bit in the PRCR register to "1" (write enable) before writing to this register.
2. PM12 bit is set to “1” by writing a “1” in a program. (Writing a “0” has no effect.)

Figure 8.1 PM0 Register and PM1 Register

9. Bus

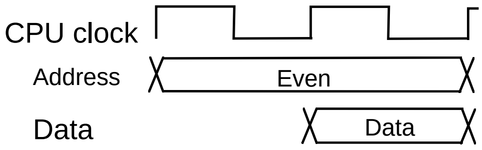
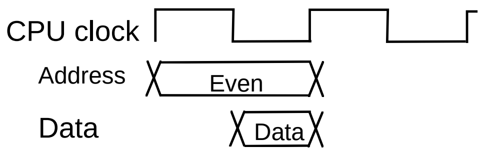
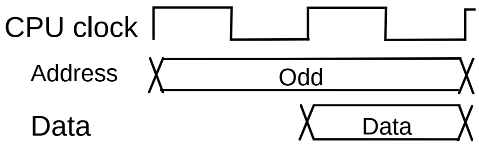
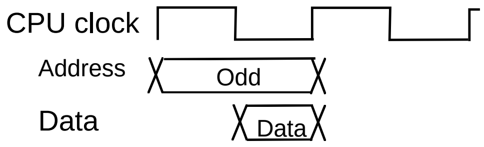
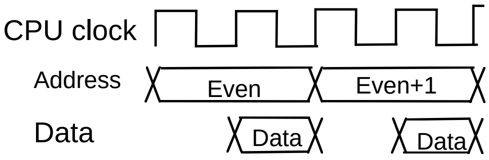
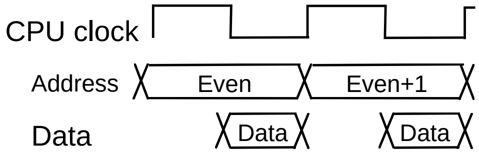
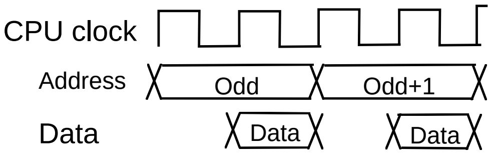
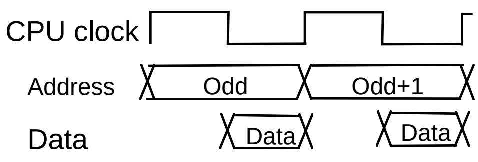
During access, the ROM/RAM and the SFR have different bus cycles. Table 9.1 shows bus cycles for access space.

The ROM/RAM and SFR are connected to the CPU through an 8-bit bus. When accessing in word (16 bits) units, these spaces are accessed twice in 8-bit units. Table 9.2 shows bus cycles in each access space.

Table 9.1 Bus Cycles for Access Space

Access space	Bus cycle
SFR	2 CPU clock cycles
ROM/RAM	1 CPU clock cycles

Table 9.2 Access Unit and Bus Operation

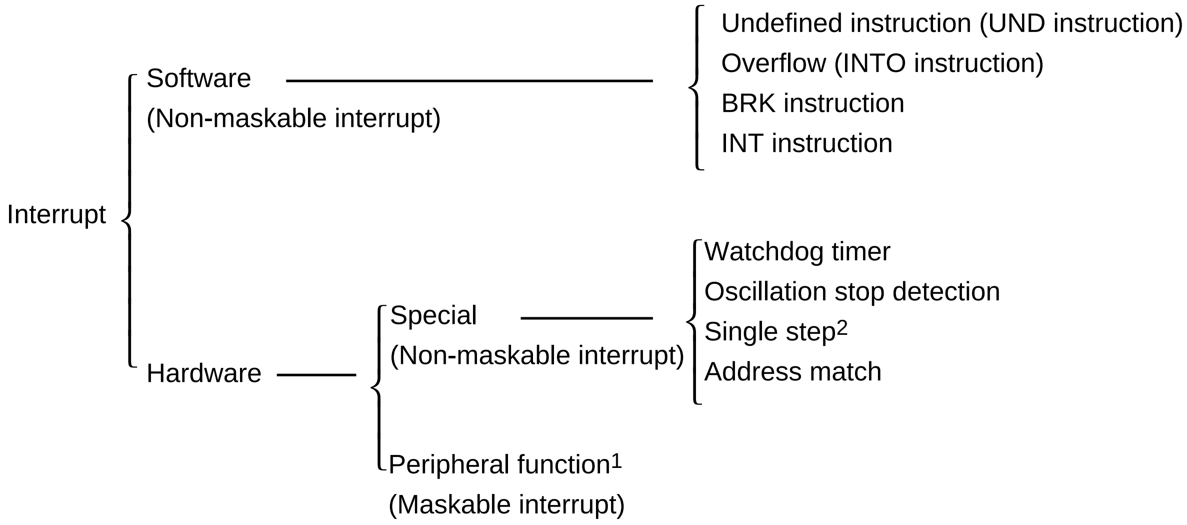
Space	SFR	ROM/RAM
Even address byte access		
Add address byte access		
Even address word access	 CPU clock Address Even Even+1 Data Data Data	 CPU clock Address Even Even+1 Data Data Data
Add address word access	 CPU clock Address Odd Odd+1 Data Data Data	 CPU clock Address Odd Odd+1 Data Data Data

10. Interrupt

10.1 Interrupt Overview

10.1.1 Type of Interrupts

Figure 10.1 shows types of interrupts.



Notes:

1. Peripheral function interrupts are generated by the peripheral functions built in the microcomputer system.
2. Avoid using this interrupt because this is a dedicated interrupt for development support tools only.

Figure 10.1 Interrupts

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

10.1.2 Software Interrupts

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

- **Undefined Instruction Interrupt**

  An undefined instruction interrupt occurs when executing the UND instruction.

- **Overflow Interrupt**

  An overflow interrupt occurs when executing the INTO instruction with the O flag set to “1” (the operation resulted in an overflow). The following are instructions whose O flag changes by arithmetic:

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

- **BRK Interrupt**

  A BRK interrupt occurs when executing the BRK instruction.

- **INT Instruction Interrupt**

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

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

10.1.3 Hardware Interrupts

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

(1) Special Interrupts

Special interrupts are non-maskable interrupts.

• Watchdog Timer Interrupt

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

• Oscillation Stop Detection Interrupt

Generated by the oscillation stop detection function. For details about the oscillation stop detection function, refer to Chapter 6, “Clock Generation Circuit.”

• Single-step Interrupt

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

• Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMAD0 to RMAD1 register that corresponds to one of the AIER register's AIER0 or AIER1 bit which is "1" (address match interrupt enabled). For details about the address match interrupt, refer to Section 10.4, “Address Match Interrupt.”

(2) Peripheral Function Interrupts

Peripheral function interrupts are maskable interrupts and generated by the microcomputer's internal functions. The interrupt sources for peripheral function interrupts are listed in Table 10.2. “Relocatable Vector Tables”. For details about the peripheral functions, refer to the description of each peripheral function in this manual.

10.1.4 Interrupts and Interrupt Vector

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

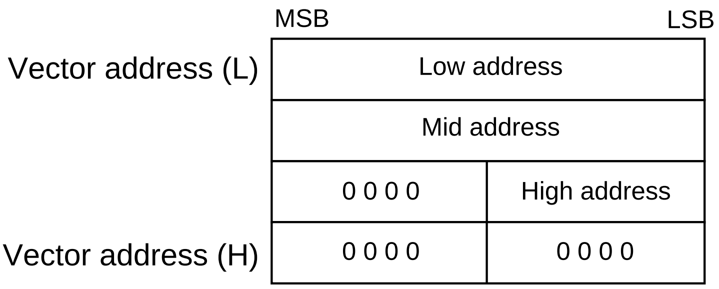


Figure 10.2 Interrupt Vector

• Fixed Vector Tables

The fixed vector tables are allocated to the addresses from 0FFDC$_{16}$ to 0FFFF$_{16}$. Table 10.1 lists the fixed vector tables. In the flash memory version of microcomputer, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to Section 17.3, "Functions to Prevent Flash Memory from Rewriting."

Table 10.1 Fixed Vector Tables

Interrupt source	Vector addresses	Remarks	Reference
	Address (L) to address (H)		
Undefined instruction	0FFDC$_{16}$ to 0FFDF$_{16}$	Interrupt on UND instruction	R8C series software manual
Overflow	0FFE0$_{16}$ to 0FFE3$_{16}$	Interrupt on INTO instruction	
BRK instruction	0FFE4$_{16}$ to 0FFE7$_{16}$	If the contents of address 0FFE7$_{16}$ is FF$_{16}$, program execution starts from the address shown by the vector in the relocatable vector table.	
Address match	0FFE8$_{16}$ to 0FFEB$_{16}$		Address match interrupt
Single step[1]	0FFEC$_{16}$ to 0FFEF$_{16}$		
Watchdog timer, oscillation stop detection	0FFF0$_{16}$ to 0FFF3$_{16}$		Watchdog timer, clock generation circuit
(Reserved)	0FFF4$_{16}$ to 0FFF7$_{16}$		
(Reserved)	0FFF8$_{16}$ to 0FFFB$_{16}$		
Reset	0FFFC$_{16}$ to 0FFFF$_{16}$		Reset

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

- **Relocatable Vector Tables**

  The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. Table 10.2 lists interrupts and vector tables located in the relocatable vector table.

Table 10.2 Interrupt and Vector Tables in Relocatable Vector Tables

Interrupt source	Vector address[1] Address (L) to address (H)	Software interrupt number	Reference
BRK instruction[2]	+0 to +3 (0000_{16} to 0003_{16})	0	R8C/Tiny Series software manual
—— (Reserved)		1 to 12	
Key input interrupt	+52 to +55 (0034_{16} to 0037_{16})	13	Key input interrupt
A-D	+56 to +59 (0038_{16} to $003B_{16}$)	14	A-D converter
—— (Reserved)		15, 16	
UART0 transmit	+68 to +71 (0044_{16} to 0047_{16})	17	Serial I/O
UART0 receive	+72 to +75 (0048_{16} to $004B_{16}$)	18	
UART1 transmit	+76 to +79 ($004C_{16}$ to $004F_{16}$)	19	
UART1 receive	+80 to +83 (0050_{16} to 0053_{16})	20	
$\overline{\text{INT2}}$	+84 to +87 (0054_{16} to 0057_{16})	21	$\overline{\text{INT}}$ interrupt
Timer X	+88 to +91 (0058_{16} to $005B_{16}$)	22	Timer X
Timer Y	+92 to +95 ($005C_{16}$ to $005F_{16}$)	23	Timer Y
Timer Z	+96 to +99 (0060_{16} to 0063_{16})	24	Timer Z
$\overline{\text{INT1}}$	+100 to +103 (0064_{16} to 0067_{16})	25	$\overline{\text{INT}}$ interrupt
$\overline{\text{INT3}}$	+104 to +107 (0068_{16} to $006B_{16}$)	26	
Timer C	+108 to +111 ($006C_{16}$ to $006F_{16}$)	27	Timer C
—— (Reserved)		28	
$\overline{\text{INT0}}$	+116 to +119 (0074_{16} to 0077_{16})	29	$\overline{\text{INT}}$ interrupt
—— (Reserved)		30	
—— (Reserved)		31	
Software interrupt[2]	+128 to +131 (0080_{16} to 0083_{16}) to +252 to +255 ($00FC_{16}$ to $00FF_{16}$)	32 to 63	R8C/Tiny Series software manual

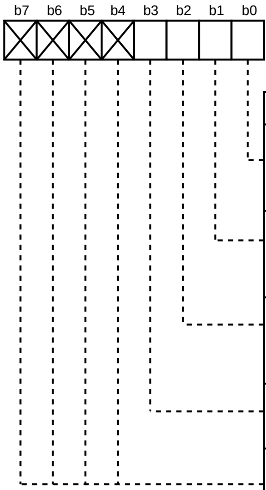
Notes:
1. Address relative to address in INTB.
2. These interrupts cannot be disabled using the I flag.

10.1.5 Interrupt Control

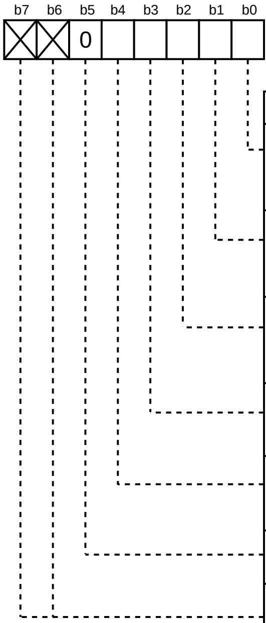
The following describes how to enable/disable the maskable interrupts, and how to set the priority in which order they are accepted. What is explained here does not apply to nonmaskable interrupts.
Use the FLG register’s I flag, IPL, and each interrupt control register's ILVL2 to ILVL0 bits to enable/disable the maskable interrupts. Whether an interrupt is requested is indicated by the IR bit in each interrupt control register.
Figure 10.3 shows the interrupt control registers.

Interrupt control register[2]

Symbol	Address	After reset
KUPIC	004D$_{16}$	XXXXX000$_2$
ADIC	004E$_{16}$	XXXXX000$_2$
S0TIC, S1TIC	0051$_{16}$, 0053$_{16}$	XXXXX000$_2$
S0RIC, S1RIC	0052$_{16}$, 0054$_{16}$	XXXXX000$_2$
INT2IC	0055$_{16}$	XXXXX000$_2$
TXIC	0056$_{16}$	XXXXX000$_2$
TYIC	0057$_{16}$	XXXXX000$_2$
TZIC	0058$_{16}$	XXXXX000$_2$
INT1IC	0059$_{16}$	XXXXX000$_2$
INT3IC	005A$_{16}$	XXXXX000$_2$
TCIC	005B$_{16}$	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[1]
— (b7-b4)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—



Symbol	Address	After reset
INT0IC	005D$_{16}$	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[1]
POL	Polarity select bit[3, 4]	0 : Selects falling edge 1 : Selects rising edge	RW
—	Reserved bit	Must always be set to “0”	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—

Notes:
1. Only "0" can be written to the IR bit. (Do not write "1").
2. To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. Refer to the paragraph 1.2.6 “Changing Interrupt Control Registers” in the Usage Notes Reference Book.
3. If the INT0PL bit in the INTEN register is set to “1” (both edges), set the POL bit to "0 " (selecting falling edge).
4. The IR bit may be set to “1” (interrupt requested) when the POL bit is rewritten. Refer to the paragraph 1.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

Figure 10.3 Interrupt Control Registers

• I Flag

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

• IR Bit

The IR bit is set to “1” (interrupt requested) when an interrupt request is generated. Then, when the interrupt request is accepted and the CPU branches to the corresponding interrupt vector, the IR bit is cleared to “0” (= interrupt not requested).
The IR bit can be cleared to “0” in a program. Note that do not write “1” to this bit.

• ILVL2 to ILVL0 Bits and IPL

Interrupt priority levels can be set using the ILVL2 to ILVL0 bits.
Table 10.3 shows the settings of interrupt priority levels and Table 10.4 shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:

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

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

Table 10.3 Settings of Interrupt Priority Levels

ILVL2 to ILVL0 bits	Interrupt priority level	Priority order
000_2	Level 0 (interrupt disabled)	—
001_2	Level 1	Lowest
010_2	Level 2	
011_2	Level 3	
100_2	Level 4	
101_2	Level 5	
110_2	Level 6	
111_2	Level 7	Highest

Table 10.4 Interrupt Priority Levels Enabled by IPL

IPL	Enabled interrupt priority levels
000_2	Interrupt levels 1 and above are enabled
001_2	Interrupt levels 2 and above are enabled
010_2	Interrupt levels 3 and above are enabled
011_2	Interrupt levels 4 and above are enabled
100_2	Interrupt levels 5 and above are enabled
101_2	Interrupt levels 6 and above are enabled
110_2	Interrupt levels 7 and above are enabled
111_2	All maskable interrupts are disabled

• 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.

The CPU behavior during the interrupt sequence is described below. Figure 10.4 shows time required for executing the interrupt sequence.

(1) The CPU gets interrupt information (interrupt number and interrupt request priority level) by reading the address 00000_{16}. Then it clears the IR bit for the corresponding interrupt to “0” (interrupt not requested).

(2) The FLG register immediately before entering the interrupt sequence is saved to the CPU’s internal temporary register(Note).

(3) The I, D and U flags in the FLG register become as follows:
The I flag is cleared to “0” (interrupts disabled).
The D flag is cleared to “0” (single-step interrupt disabled).
The U flag is cleared to “0” (ISP selected).
However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.

(4) The CPU’s internal temporary register (Note) is saved to the stack.

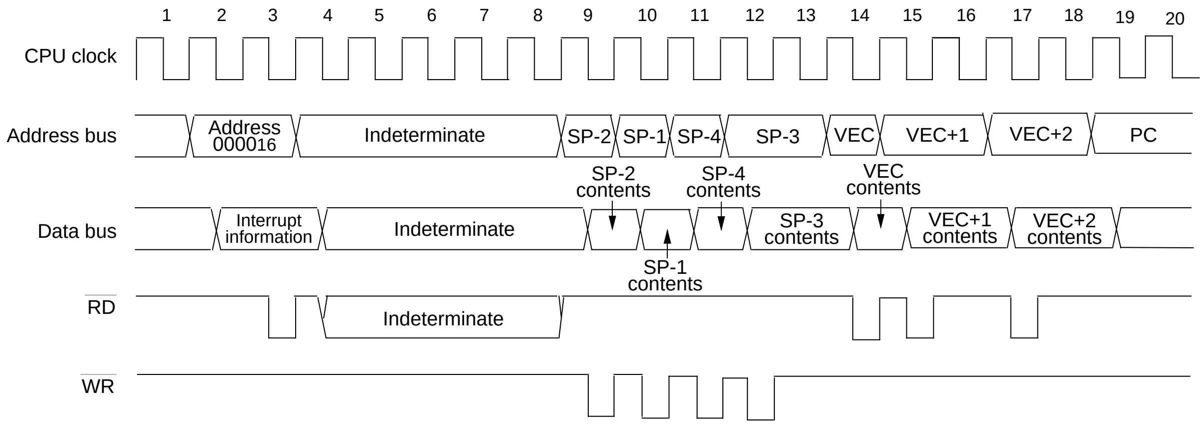
(5) The PC is saved to the stack.

(6) The interrupt priority level of the accepted interrupt is set in the IPL.

(7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

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

Note: This register cannot be used by user.

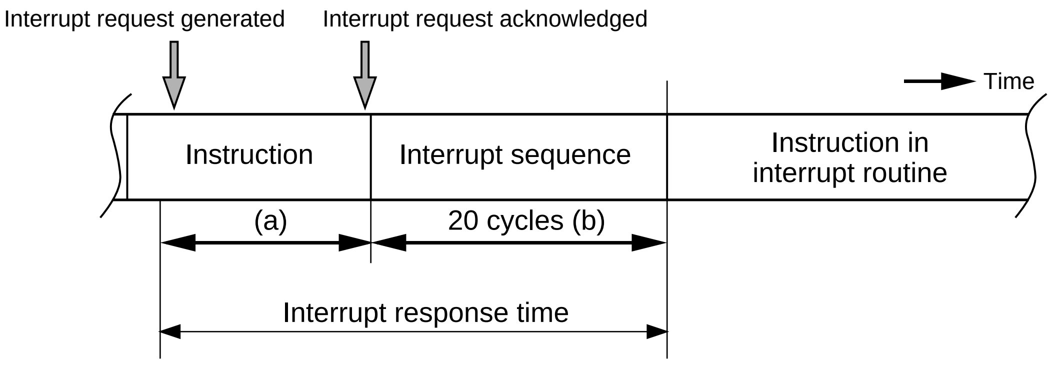


The indeterminate state depends on the instruction queue buffer. A read cycle occurs when the instruction queue buffer is ready to accept instructions.

Figure 10.4 Time Required for Executing Interrupt Sequence

- **Interrupt Response Time**

Figure 10.5 shows the interrupt response time. The interrupt response or interrupt acknowledge time denotes a time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of a time from when an interrupt request is generated till when the instruction then executing is completed (see #a in Figure 10.5) and a time during which the interrupt sequence is executed (20 cycles, see #b in Figure 10.5).


Interrupt request generated
Interrupt request acknowledged
Time
Instruction
Interrupt sequence
Instruction in interrupt routine
(a)
20 cycles (b)
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) 21 cycles for address match and single-step interrupts.

Figure 10.5 Interrupt Response Time

- **Variation of IPL when Interrupt Request is Accepted**

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

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

Table 10.5 IPL Level That Is Set to IPL When A Software or Special Interrupt Is Accepted

Interrupt sources	Level that is set to IPL
Watchdog timer, oscillation stop detection	7
Software, address match, single-step	Not changed

- **Saving Registers**

In the interrupt sequence, the FLG register and PC are saved to the stack.
At this time, the 4 high-order bits in the PC and the 4 high-order (IPL) and 8 low-order bits in the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits in the PC are saved.
Figure 10.6 shows the stack status before and after an interrupt request is accepted.
The other necessary registers must be saved in a program at the beginning of the interrupt routine. Use the PUSHM instruction, and all registers except SP can be saved with a single instruction.

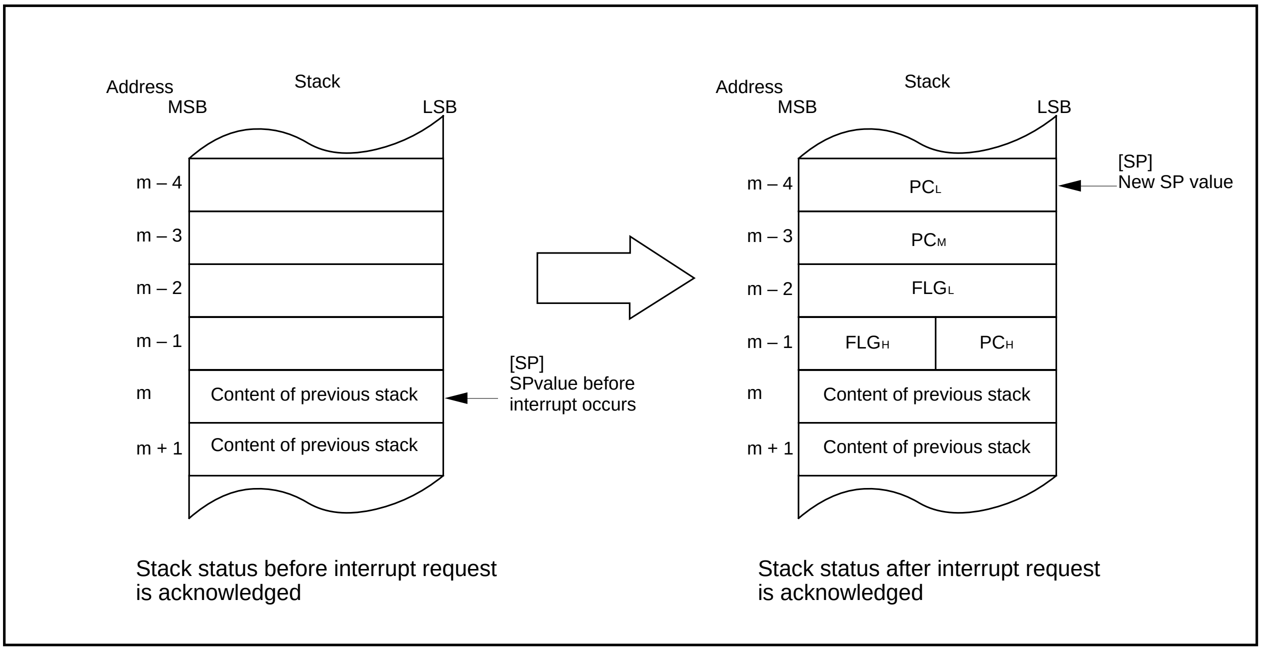
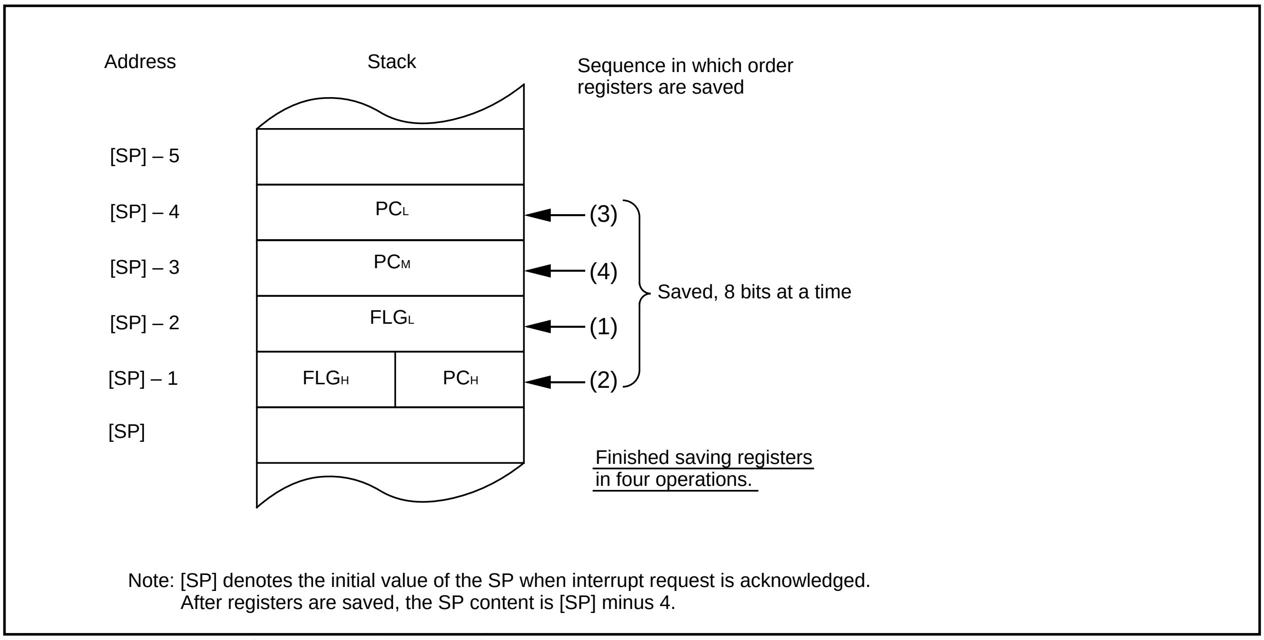

Address
Stack
MSB
LSB
m – 4
m – 3
m – 2
m – 1
m
m + 1
Content of previous stack
Content of previous stack
[SP]
SPvalue before
interrupt occurs
Stack status before interrupt request
is acknowledged
Address
Stack
MSB
LSB
PCL
PCM
FLGL
FLGH
PCH
Content of previous stack
Content of previous stack
[SP]
New SP value
Stack status after interrupt request
is acknowledged

Figure 10.6 Stack Status Before and After Acceptance of Interrupt Request

The registers are saved in four steps, 8 bits at a time. Figure 10.7 shows the operation of the saving registers.

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



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

Figure 10.7 Operation of Saving Register

- **Returning from an Interrupt Routine**

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

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

- **Interrupt Priority**

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

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

The watchdog timer and other special interrupts have their priority levels set in hardware. Figure 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.

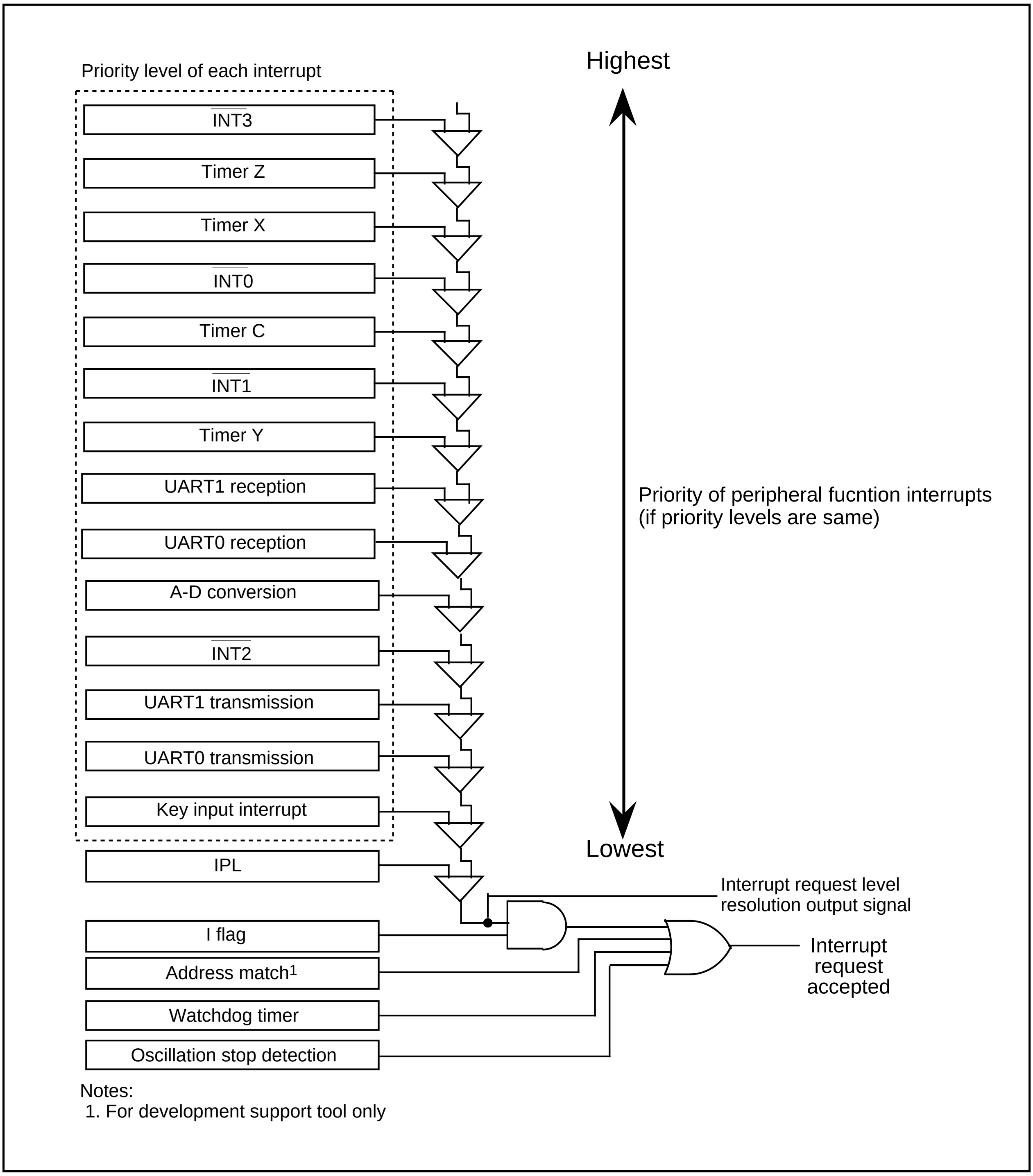
Reset > WDT/Oscillation stop detection > Peripheral function > Single step > Address match

Figure 10.8 Hardware Interrupt Priority

- **Interrupt Priority Resolution Circuit**

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

  Figure 10.9 shows the circuit that judges the interrupt priority level.


Priority level of each interrupt
INT3
Timer Z
Timer X
INT0
Timer C
INT1
Timer Y
UART1 reception
UART0 reception
A-D conversion
INT2
UART1 transmission
UART0 transmission
Key input interrupt
IPL
I flag
Address match1
Watchdog timer
Oscillation stop detection
Highest
Priority of peripheral fucntion interrupts (if priority levels are same)
Lowest
Interrupt request level resolution output signal
Interrupt request accepted

Notes:
1. For development support tool only

Figure 10.9 Interrupts Priority Select Circuit

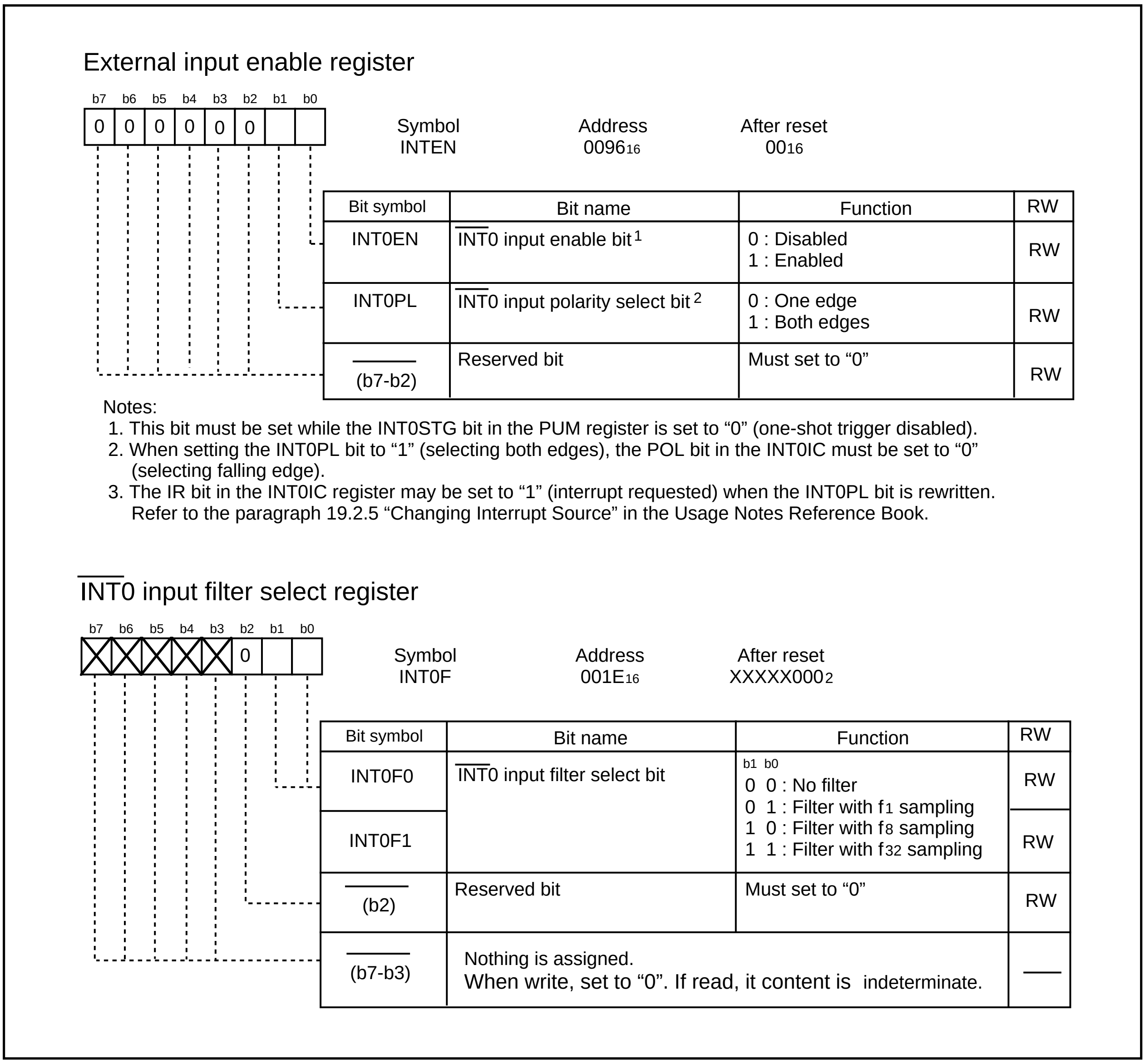
10.2 $\overline{\text{INT}}$ Interrupt

10.2.1 $\overline{\text{INT0}}$ Interrupt

$\overline{\text{INT0}}$ interrupt is triggered by an INT0 input. When using INT0 interrupts, the INT0EN bit in the INTEN register must be set to “1” (enabling). The edge polarity is selected using the INT0PL bit in the INTEN register and the POL bit in the INT0IC register. The IR bit may be set to “1” (interrupt requested) after changing the INT0PL or POL bit. The IR bit must be set to “0” (interrupt not requested) after changing the INT0PL and POL bits.

Inputs can be passed through a digital filter with three different sampling clocks.

Figure 10.10 shows the INTEN and INT0F registers.



External input enable register

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

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

Bit symbol	Bit name	Function	RW
INT0EN	$\overline{\text{INT0}}$ input enable bit [1]	0 : Disabled 1 : Enabled	RW
INT0PL	$\overline{\text{INT0}}$ input polarity select bit [2]	0 : One edge 1 : Both edges	RW
—— (b7-b2)	Reserved bit	Must set to “0”	RW

Notes:
1. This bit must be set while the INT0STG bit in the PUM register is set to “0” (one-shot trigger disabled).
2. When setting the INT0PL bit to “1” (selecting both edges), the POL bit in the INT0IC must be set to “0” (selecting falling edge).
3. The IR bit in the INT0IC register may be set to “1” (interrupt requested) when the INT0PL bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

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

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒	☒	0		

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

Bit symbol	Bit name	Function	RW
INT0F0	$\overline{\text{INT0}}$ 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
—— (b2)	Reserved bit	Must set to “0”	RW
—— (b7-b3)	Nothing is assigned. When write, set to “0”. If read, it content is indeterminate.		——

Figure 10.10 INTEN Register and INT0F Register

10.2.2 $\overline{\text{INT0}}$ Input Filter

The $\overline{\text{INT0}}$ input has a digital filter which can be sampled by one of three sampling clocks. The sampling clock is selected using the INT0F1 to INT0F0 bits in the INT0F register. The IR bit in the INT0IC register is set to “1” (interrupt requested) when the sampled input level matches three times. When the INT0F1 to INT0F0 bits are set to “01_2”, “10_2”, or “11_2”, the P4_5 bit in the P4 register indicates the filtered value.

Figure 10.11 shows the $\overline{\text{INT0}}$ input filter configuration. Figure 10.12 shows an operation example of $\overline{\text{INT0}}$ input filter.

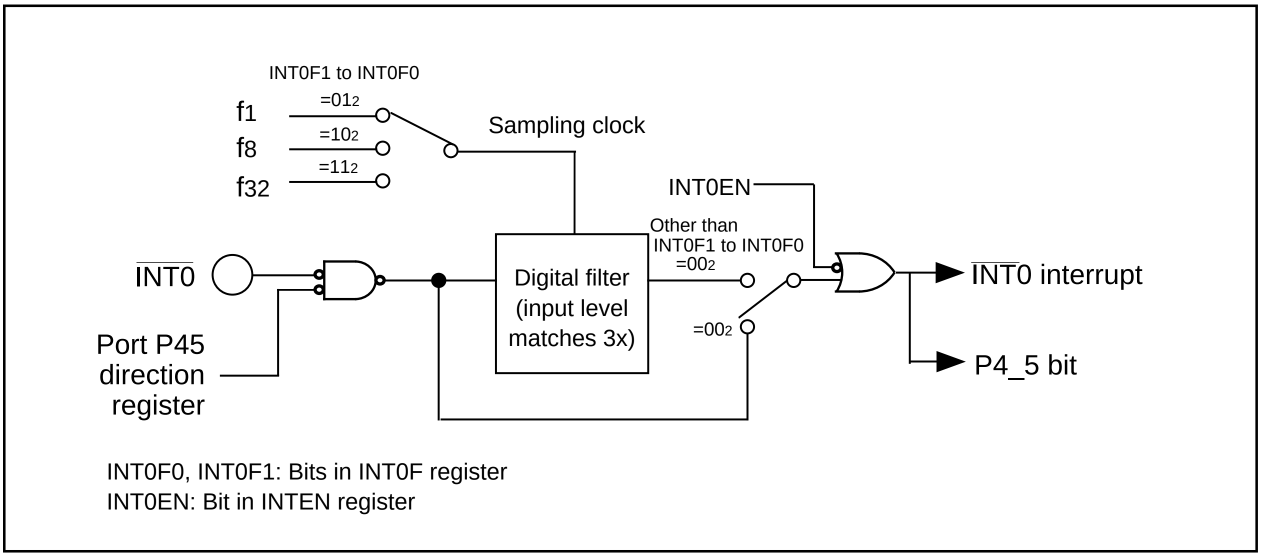

INT0F1 to INT0F0
f1
=01₂
f8
=10₂
f32
=11₂
Sampling clock
INT0EN
Other than
INT0F1 to INT0F0
=00₂
=00₂
$\overline{\text{INT0}}$
Digital filter (input level matches 3x)
$\overline{\text{INT0}}$ interrupt
P4_5 bit
Port P45 direction register
INT0F0, INT0F1: Bits in INT0F register
INT0EN: Bit in INTEN register

Figure 10.11 $\overline{\text{INT0}}$ Input Filter

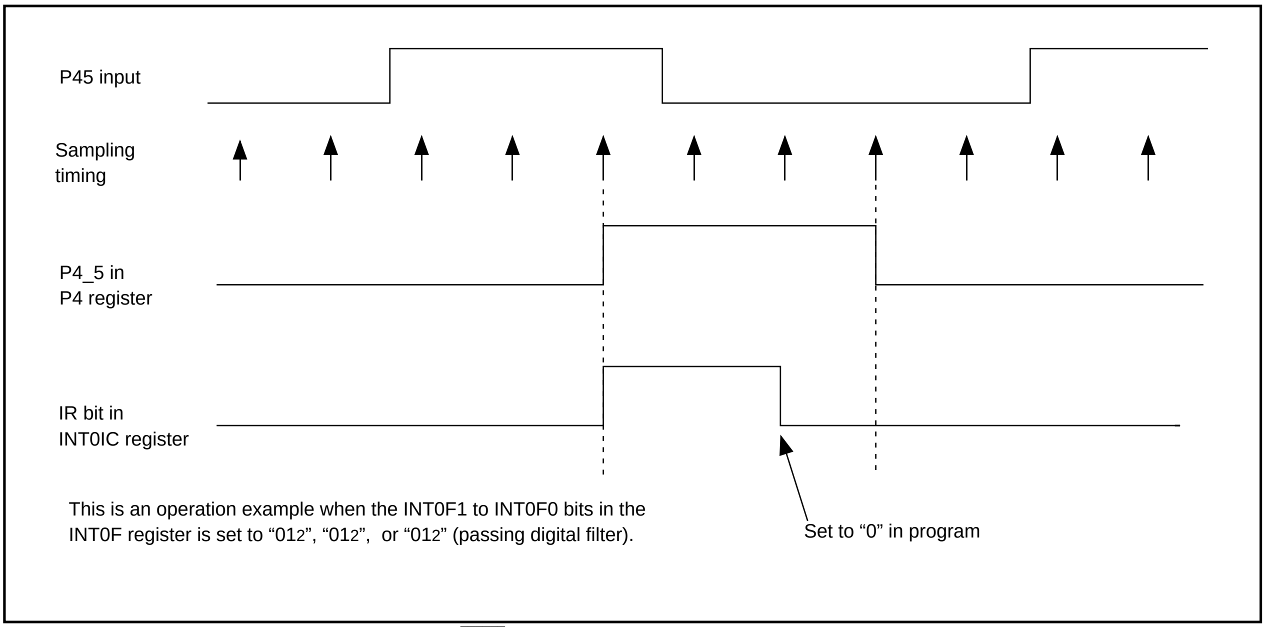

P45 input
Sampling timing
P4_5 in P4 register
IR bit in INT0IC register
This is an operation example when the INT0F1 to INT0F0 bits in the INT0F register is set to “01₂”, “01₂”, or “01₂” (passing digital filter).
Set to “0” in program

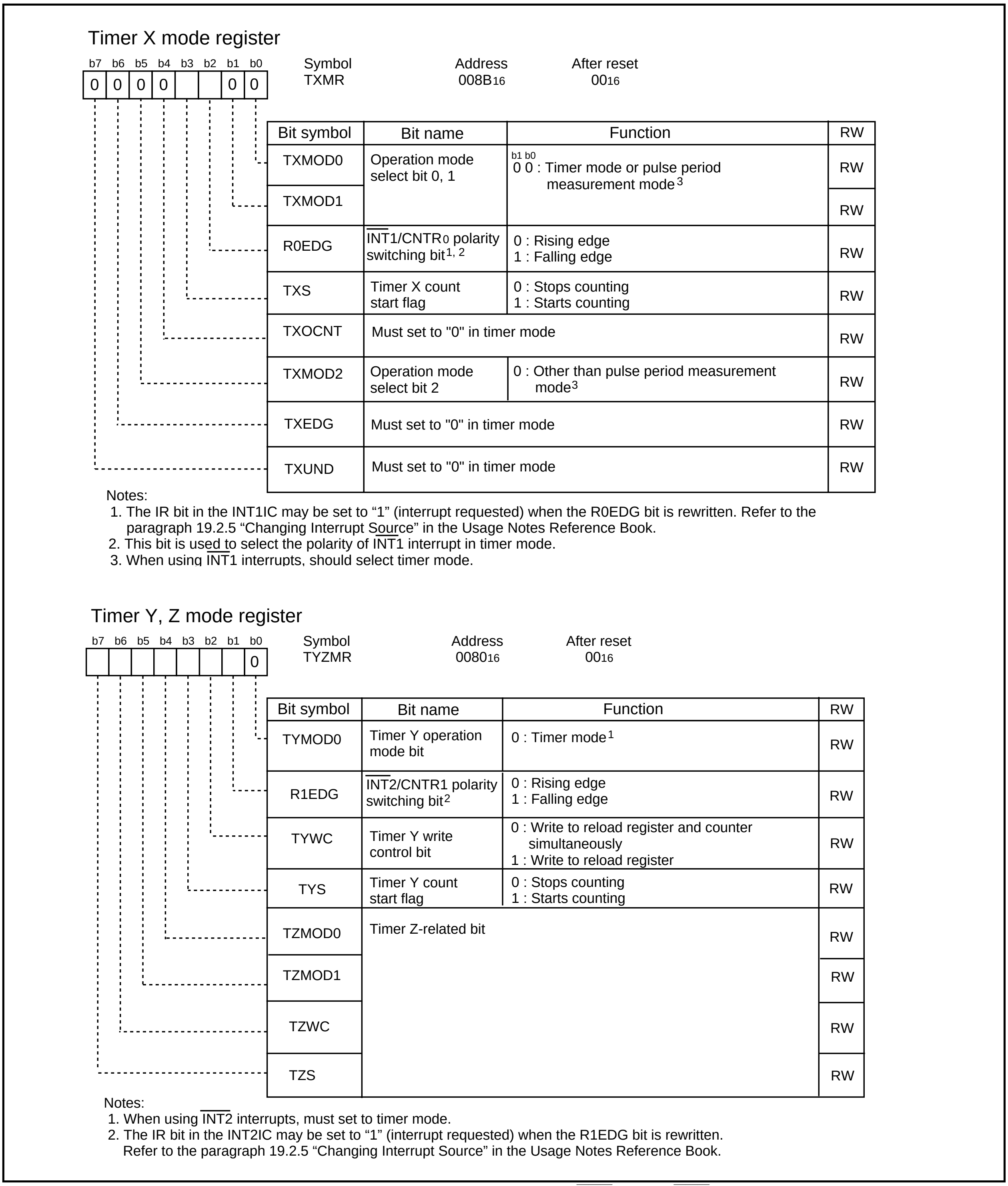
Figure 10.12 Operation Example of $\overline{\text{INT0}}$ Input Filter

10.2.3 $\overline{\mathrm{INT}}$1 Interrupt and $\overline{\mathrm{INT}}$2 Interrupt

$\overline{\mathrm{INT}}$1 interrupts are triggered by $\overline{\mathrm{INT}}$1 inputs. The edge polarity is selected with the R0EDG bit in the TXMR register. The $\overline{\mathrm{INT}}$1 pin can be used only when the Timer X is in timer mode because the $\overline{\mathrm{INT}}$1 pin shares the same pin with the CNTR0 pin.

$\overline{\mathrm{INT}}$2 interrupts are triggered by $\overline{\mathrm{INT}}$2 inputs. The edge polarity is selected with the R1EDG bit in the TYZMR register. The $\overline{\mathrm{INT}}$2 pin can be used only when the Timer Y is in timer mode because the $\overline{\mathrm{INT}}$2 pin shares the same pin with the CNTR1 pin.

Figure 10.13 shows the TXMR and TYZMR registers when using $\overline{\mathrm{INT}}$1 and $\overline{\mathrm{INT}}$2 interrupts.



Timer X mode register

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

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

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1	b1 b0 0 0 : Timer mode or pulse period measurement mode[3]	RW
TXMOD1			RW
R0EDG	$\overline{\mathrm{INT}}$1/CNTR0 polarity switching bit[1, 2]	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	Must set to "0" in timer mode		RW
TXMOD2	Operation mode select bit 2	0 : Other than pulse period measurement mode[3]	RW
TXEDG	Must set to "0" in timer mode		RW
TXUND	Must set to "0" in timer mode		RW

Notes:
1. The IR bit in the INT1IC may be set to “1” (interrupt requested) when the R0EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.
2. This bit is used to select the polarity of $\overline{\mathrm{INT}}$1 interrupt in timer mode.
3. When using $\overline{\mathrm{INT}}$1 interrupts, should select timer mode.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
							0

Symbol TYZMR Address 0080_{16} After reset 00_{16}

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y operation mode bit	0 : Timer mode[1]	RW
R1EDG	$\overline{\mathrm{INT}}$2/CNTR1 polarity switching bit[2]	0 : Rising edge 1 : Falling edge	RW
TYWC	Timer Y write control bit	0 : Write to reload register and counter simultaneously 1 : Write to reload register	RW
TYS	Timer Y count start flag	0 : Stops counting 1 : Starts counting	RW
TZMOD0	Timer Z-related bit		RW
TZMOD1			RW
TZWC			RW
TZS			RW

Notes:
1. When using $\overline{\mathrm{INT}}$2 interrupts, must set to timer mode.
2. The IR bit in the INT2IC may be set to “1” (interrupt requested) when the R1EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

Figure 10.13 TXMR Register and TYZMR Register when $\overline{\mathrm{INT}}$1 and $\overline{\mathrm{INT}}$2 Interrupt Used

10.2.4 $\overline{\text{INT3}}$ Interrupt

$\overline{\text{INT3}}$ interrupts are triggered by $\overline{\text{INT3}}$ inputs. The TCC07 bit in the TCC0 register should be se to “0” ($\overline{\text{INT3}}$). The $\overline{\text{INT3}}$ input has a digital filter which can be sampled by one of three sampling clocks. The sampling clock is selected using the TCC11 to TCC10 bits in the TCC1 register. The IR bit in the INT3IC register is set to “1” (interrupt requested) when the sampled input level matches three times. The P3_3 bit in the P3 register indicates the previous value before filtering regardless of values set in the TCC11 to TCC10 bits.

When setting the TCC07 bit to “1” (fRING128), $\overline{\text{INT3}}$ interrupts are triggered by fRING128 clock. The IR bit in the INT3IC register is set to “1” (interrupt requested) every fRING128 clock cycle or every half fRING128 clock cycle.

Figure 10.14 shows the TCC0 and TCC1 registers.

Timer C control register 0

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	After reset
TCC0	$009A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TCC00	Capture control bit	0 : Capture disabled 1 : Capture enabled	RW
TCC01	Timer C count source select bit[1]	b2 b1 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : Avoid this setting	RW
TCC02			RW
TCC03	$\overline{\text{INT3}}$ interrupt/capture input polarity select bit[1, 2]	b4 b3 0 0 : Rising edge 0 1 : Falling edge 1 0 : Both edges 1 1 : Avoid this setting	RW
TCC04			RW
$\overline{}$ (b6-b5)	reserved bit	Must set to "0"	RW
TCC07	$\overline{\text{INT3}}$ interrupt/capture input switching bit[1, 2]	0 : $\overline{\text{INT3}}$ 1 : fRING128	RW

Notes:
1. Change this bit when TCC00 bit is set to “0” (count stop).
2. The IR bit in the INT3IC may be set to “1” (interrupt requested) when the TCC03, TCC04, or TCC07 bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

Timer C control register 1

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

Symbol	Address	After reset
TCC1	$009B_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TCC10	$\overline{\text{INT3}}$ input filter select bit[1]	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
TCC11			RW
$\overline{}$ (b7-b2)	Reserved bit	Must set to "0"	RW

Notes:
1. Input is recognized only when the same value from $\overline{\text{INT3}}$ pin is sampled three times in succession.

Figure 10.14 TCC0 Register and TCC1 Register

10.3 Key Input Interrupt

A key input interrupt is generated on an input edge of any of the $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$ pins. Key input interrupts can be used as a key-on wakeup function to exit wait or stop mode. $\overline{\text{KIi}}$ input can be enabled or disabled selecting with the KIiEN (i=0 to 3) bit in the KIEN register. The edge polarity can be rising edge or falling edge selecting with the KIiPL bit in the KIEN register. Note, however, that while input on any $\overline{\text{KIi}}$ pin which has had the KIiPL bit set to “0” (falling edge) is pulled low, inputs on all other pins of the port are not detected as interrupts. Similarly, while input on any $\overline{\text{KIi}}$ pin which has had the KIiPL bit set to “1” (rising edge) is pulled high, inputs on all other pins of the port are not detected as interrupts.

Figure 10.15 shows a block diagram of the key input interrupt.

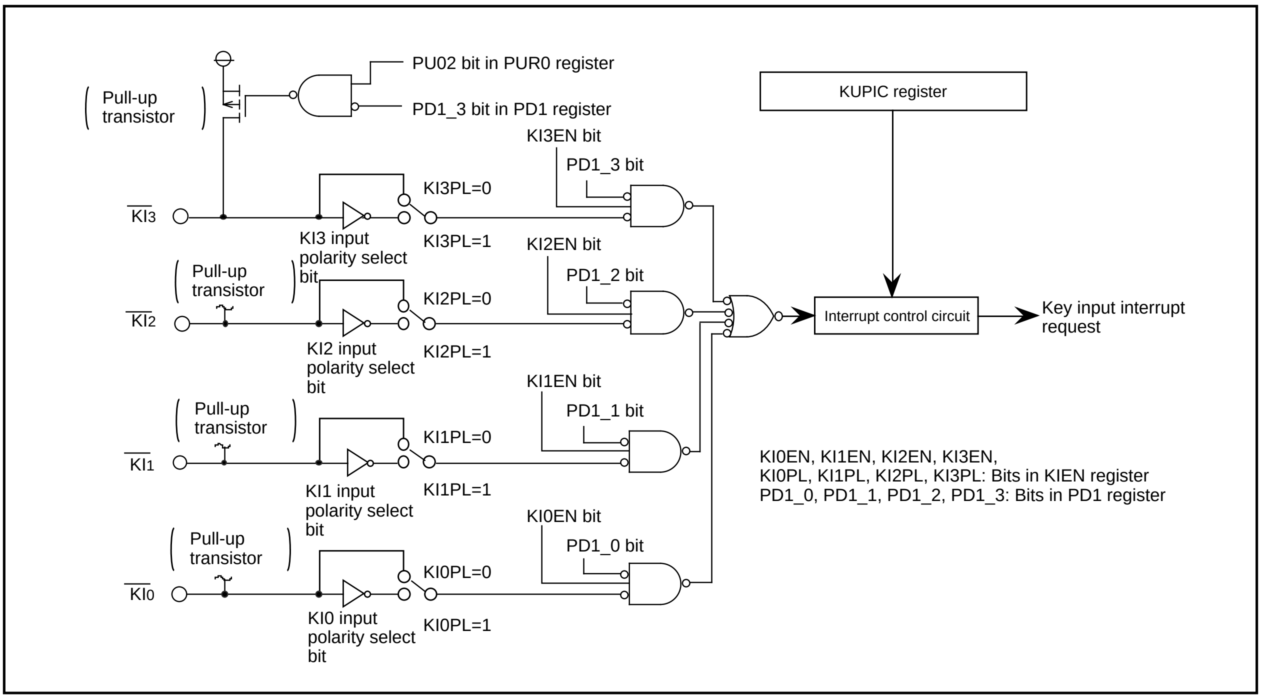

PU02 bit in PUR0 register
PD1_3 bit in PD1 register
Pull-up transistor
KUPIC register
KI3EN bit
PD1_3 bit
KI3PL=0
$\overline{\text{KI3}}$
KI3 input polarity select bit
KI3PL=1
KI2EN bit
Pull-up transistor
PD1_2 bit
KI2PL=0
$\overline{\text{KI2}}$
Interrupt control circuit
Key input interrupt request
KI2 input polarity select bit
KI2PL=1
KI1EN bit
Pull-up transistor
PD1_1 bit
KI1PL=0
$\overline{\text{KI1}}$
KI0EN, KI1EN, KI2EN, KI3EN,
KI0PL, KI1PL, KI2PL, KI3PL: Bits in KIEN register
PD1_0, PD1_1, PD1_2, PD1_3: Bits in PD1 register
KI1 input polarity select bit
KI1PL=1
KI0EN bit
Pull-up transistor
PD1_0 bit
KI0PL=0
$\overline{\text{KI0}}$
KI0 input polarity select bit
KI0PL=1

Figure 10.15 Key Input Interrupt

Key input enable register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After reset
KIEN	0098_{16}	00_{16}

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

Notes:
1. The IR bit in the KUPIC register may be set to “1” (interrupt requested) when the KIEN register is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

Figure 10.16 KIEN Register

10.4 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMADi register (i=0, 1). Set the start address of any instruction in the RMADi register. Use the AIER0 and AIER1 bits in the AIER register to enable or disable the interrupt. Note that the address match interrupt is unaffected by the I flag and IPL.

The value of the PC that is saved to the stack when an address match interrupt is acknowledged varies depending on the instruction at the address indicated by the RMAD i register (see the paragraph “register saving” for the value of the PC). Not appropriate return address is pushed on the stack. There are two ways to return from the address match interrupt as follows:

- Change the content of the stack and use a REIT instruction.
- Use an instruction such as POP to restore the stack as it was before an interrupt request was acknowledged. And then use a jump instruction.

Table 10.6 lists the value of the PC that is saved to the stack when an address match interrupt is acknowledged.

Figure 10.17 shows the AIER, and RMAD1 to RMAD0 registers.

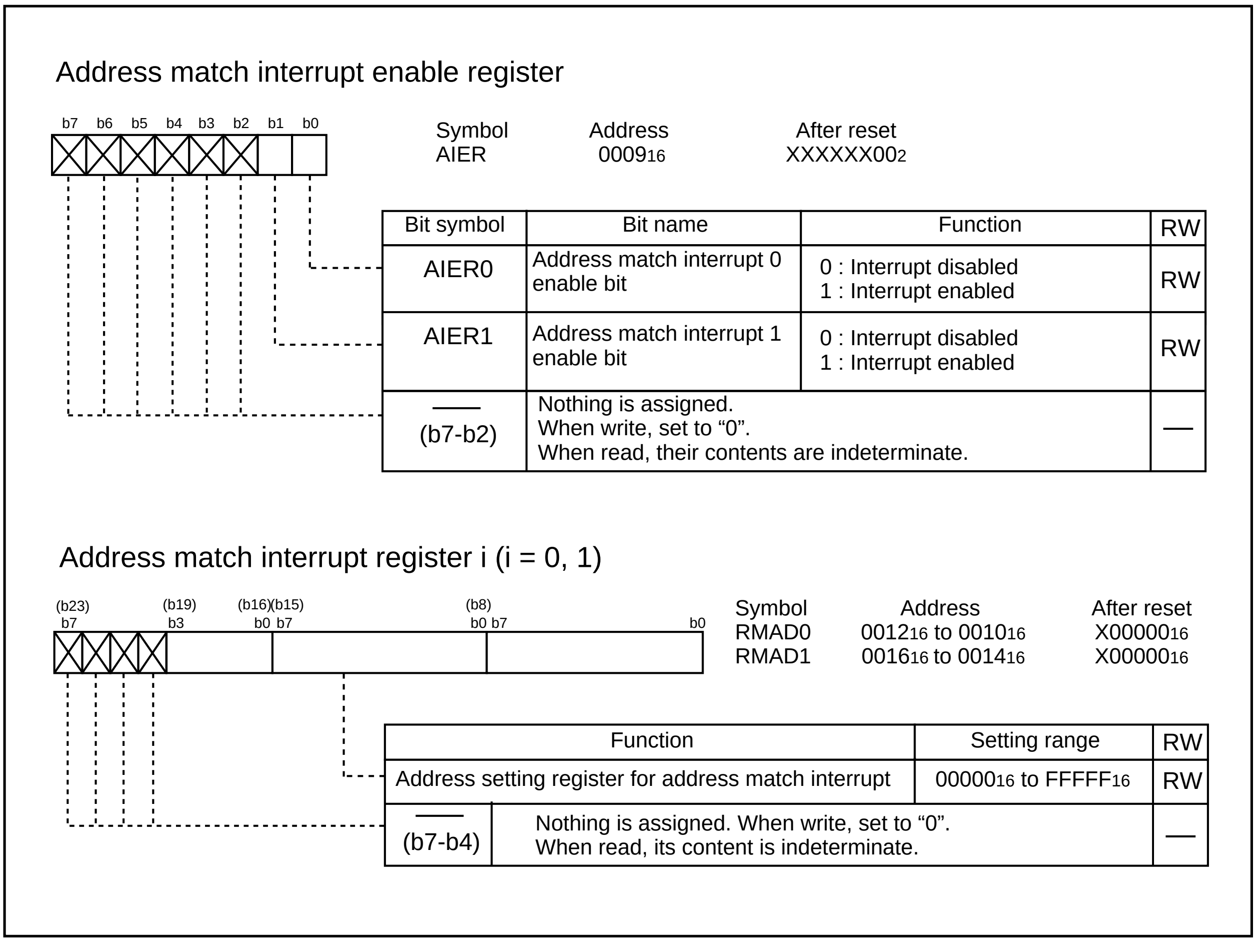
Table 10.6 Value of PC Saved to Stack when Address Match Interrupt Acknowledged

Address indicated by RMADi register (i=0,1)	PC value saved[Note]
• 16-bit operation code instruction • Instruction shown below among 8-bit operation code instructions ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ.B:S #IMM8,dest STNZ.B:S #IMM8,dest STZX.B:S #IMM81,#IMM82,dest CMP.B:S #IMM8,dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM,dest (However, dest = A0 or A1)	Address indicated by RMADi register + 2
• Instructions other than the above	Address indicated by RMADi register + 1

Note: See the paragraph “saving registers” for the PC value saved.

Table 10.7 Relationship Between Address Match Interrupt Sources and Associated Registers

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



Address match interrupt enable register

Symbol	Address	After 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
—— (b7-b2)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

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

Symbol	Address	After reset
RMAD0	0012_{16} to 0010_{16}	X00000$_{16}$
RMAD1	0016_{16} to 0014_{16}	X00000$_{16}$

Function	Setting range	RW
Address setting register for address match interrupt	00000_{16} to FFFFF$_{16}$	RW
—— (b7-b4) Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—

Figure 10.17 AIER Register and RMAD0 to RMAD1 Registers

11. Watchdog Timer

The watchdog timer is the function of detecting when the program is out of control. Therefore, we recommend using the watchdog timer to improve reliability of a system. The watchdog timer contains a 15-bit counter which counts down the clock derived by dividing the CPU clock using the prescaler. Whether to generate a watchdog timer interrupt request or apply a watchdog timer reset as an operation to be performed when the watchdog timer underflows after reaching the terminal count can be selected using the PM12 bit in the PM1 register. The PM12 bit can only be set to “1” (reset). Once this bit is set to “1”, it cannot be set to “0” (watchdog timer interrupt) in a program. Refer to Section 5.1.3, “Watchdog Timer Reset” for details.
The divide-by-N value for the prescaler can be chosen to be 16 or 128 with the WDC7 bit in the WDC register. The period of watchdog timer can be calculated as given below. The period of watchdog timer is, however, subject to an error due to the prescaler.

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

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

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

In stop mode and wait mode, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.
Figure 11.1 shows the block diagram of the watchdog timer. Figure 11.2 shows the watchdog timer-related registers.

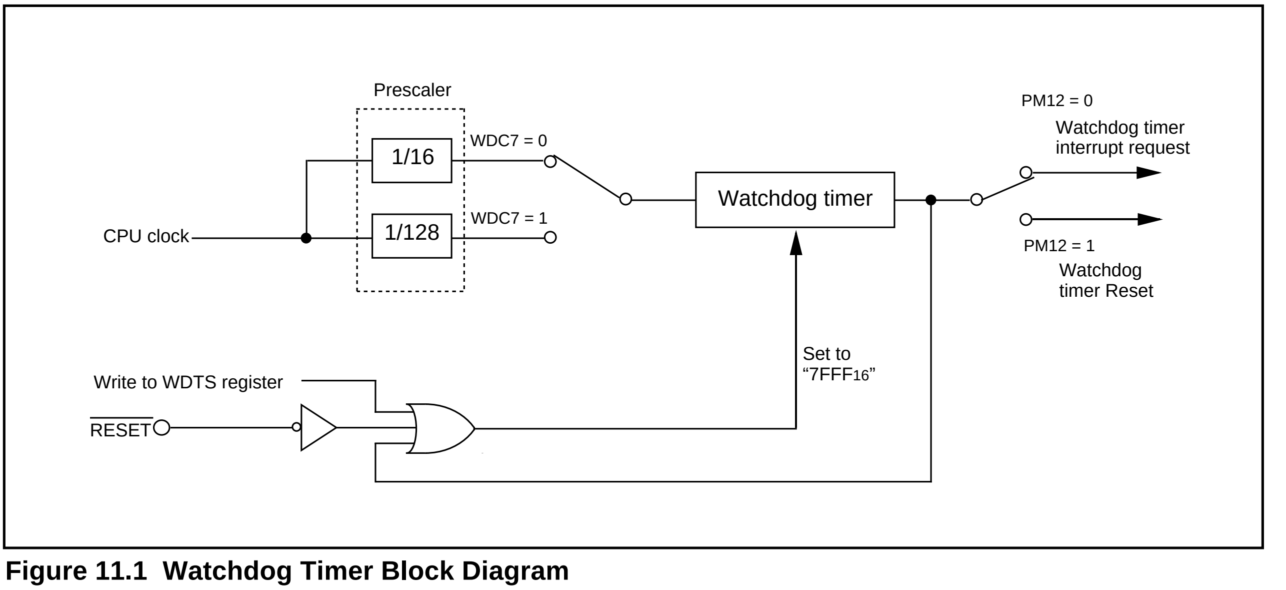
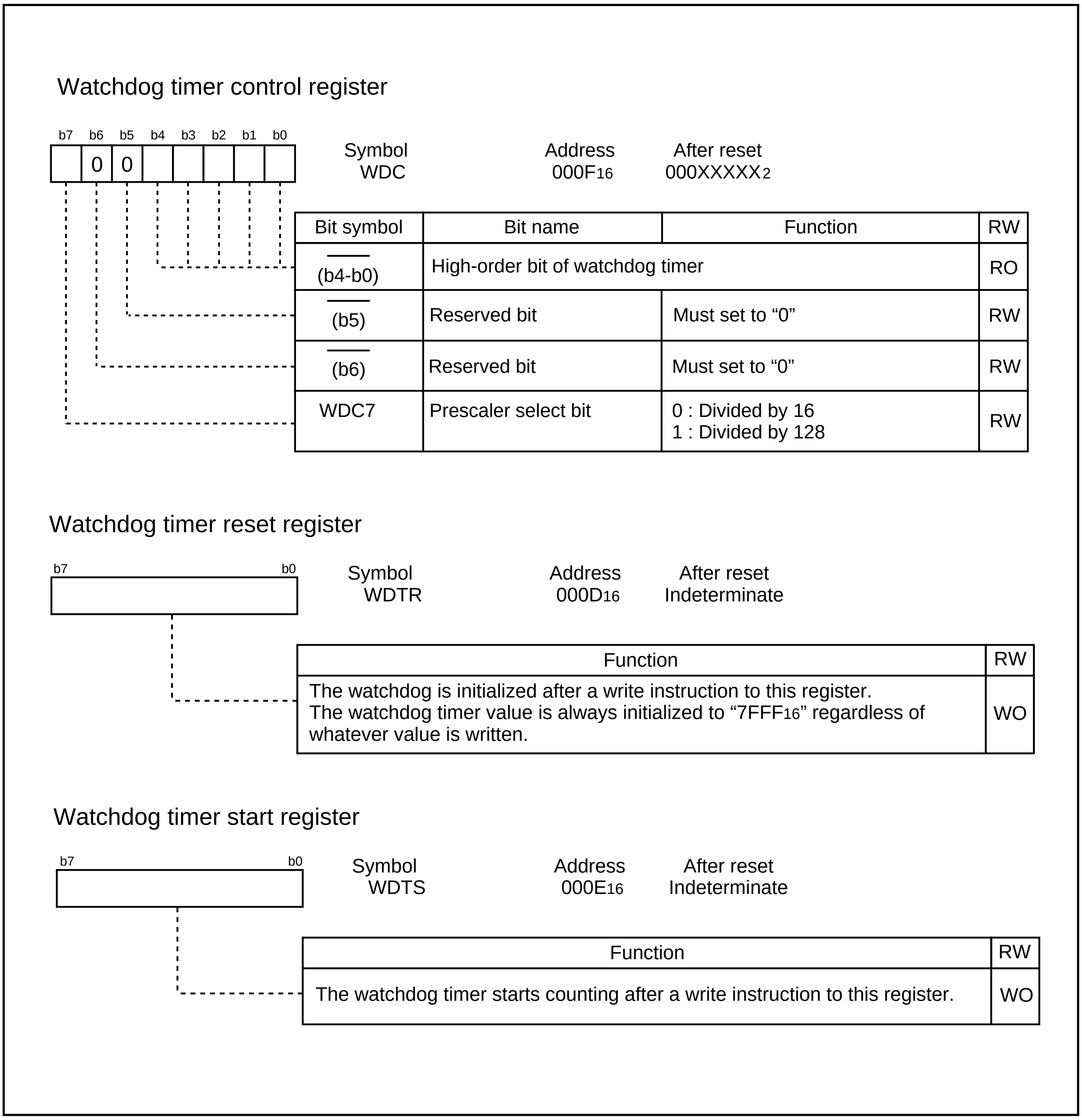


Figure 11.1 Watchdog Timer Block Diagram



Watchdog timer control register

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

Bit symbol	Bit name	Function	RW
— (b4-b0)	High-order bit of watchdog timer		RO
— (b5)	Reserved bit	Must set to “0”	RW
— (b6)	Reserved bit	Must set to “0”	RW
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	RW

Watchdog timer reset register

Symbol	Address	After reset
WDTR	$000D_{16}$	Indeterminate

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

Watchdog timer start register

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

Function	RW
The watchdog timer starts counting after a write instruction to this register.	WO

Figure 11.2 WDC Register, WDTR Register, and WDTS Register

12. Timers

The microcomputer has three 8-bit timers and one 16-bit timer. The three 8-bit timers are Timer X, Timer Y, and Timer Z and each one has an 8-bit prescaler. The 16-bit timer is Timer C and has a capture. 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 lists functional comparison.

Table 12.1 Functional Comparison

Item		Timer X	Timer Y	Timer Z	Timer C
Configuration		8-bit timer with 8-bit prescaler	8-bit timer with 8-bit prescaler	8-bit timer with 8-bit prescaler	16-bit timer
Count		Down	Down	Down	Up
Count source		•f1 •f2 •f8 •f32	•f1 •f8 •fRING •Input from CNTR1 pin	•f1 •f2 •f8 •Timer Y underflow	•f1 •f8 •f32
Function	Timer mode	provided	provided	provided	not provided
	Pulse output mode	provided	not provided	not provided	not provided
	Event counter mode	provided	provided[1]	not provided	not provided
	Pulse width measurement mode	provided	not provided	not provided	not provided
	Pulse period measurement mode	provided	not provided	not provided	not provided
	Programmable waveform generation mode	not provided	provided	provided	not provided
	Programmable one-shot generation mode	not provided	not provided	provided	not provided
	Programmable wait one-shot generation mode	not provided	not provided	provided	not provided
	Capture	not provided	not provided	not provided	provided
Input pin		CNTR0	CNTR1	$\overline{\text{INT0}}$	TCIN
Output pin		CNTR0 $\overline{\text{CNTR0}}$	CNTR1	TZOUT	not provided
Related interrupt		Timer X int $\overline{\text{INT1}}$ int	Timer Y int $\overline{\text{INT2}}$ int	Timer Z int $\overline{\text{INT0}}$ int	Timer C int $\overline{\text{INT3}}$ int
Timer stop		provided	provided	provided	provided

Note: Select the input from the CNTR1 pin as a count source of timer mode.

12.1 Timer X

The Timer X is an 8-bit timer with an 8-bit prescaler. Figure 12.1 shows the block diagram of Timer X. Figures 12.2 and 12.3 show the Timer X-related registers.

The Timer X has 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 external pulses.
• 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.

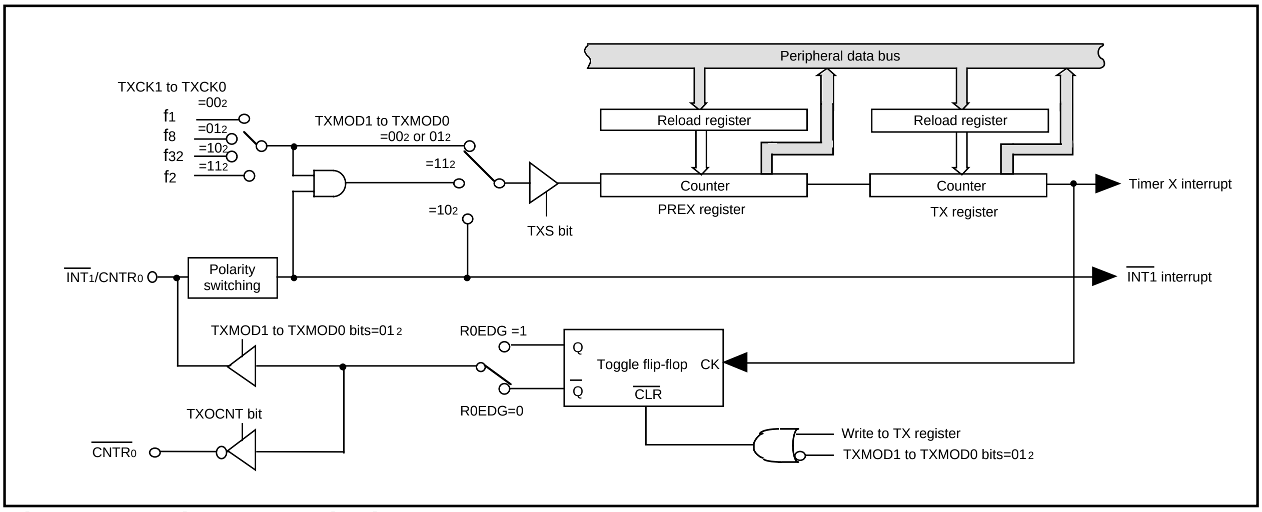


Figure 12.1 Timer X Block Diagram

Timer X mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After reset
TXMR	008B₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1	b1 b0 0 0 : Timer mode or pulse period measurement mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	RW
TXMOD1			RW
R0EDG	INT1/CNTR0 polarity switching bit[1]	Function varies with each operation mode	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P30/CNTR0 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	Active edge reception flag	Function varies with each operation mode.	RW
TXUND	Timer X under flow flag	Function varies with each operation mode.	RW

Notes:
1. The IR bit in the INT1IC register may be set to “1” (interrupt requested) when the R0EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

Figure 12.2 TXMR Register

Prescaler X Register

b7 b0

Symbol	Address	After reset
PREX	008C$_{16}$	FF$_{16}$

Mode	Function	Setting range	RW
Timer mode	Internal count source is counted	00$_{16}$ to FF$_{16}$	RW
Pulse output mode	Internal count source is counted	00$_{16}$ to FF$_{16}$	RW
Event counter mode	Externally input pulses are counted	00$_{16}$ to FF$_{16}$	RW
Pulse width measurement mode	Pulse width of externally input pulses is measured (Internal count source is counted)	00$_{16}$ to FF$_{16}$	RW
Pulse period measurement mode	Pulse period of externally input pulses is measured (Internal count source is counted)	00$_{16}$ to FF$_{16}$	RW

Timer X Register

b7 b0

Symbol	Address	After reset
TX	008D$_{16}$	FF$_{16}$

Function	Setting range	RW
Underflow of Prescaler X is counted	00$_{16}$ to FF$_{16}$	RW

Timer count source setting register

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Symbol	Address	After reset
TCSS	008E$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
TXCK0	Timer X count source select bit[1]	b1 b0 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : f2	RW
TXCK1			RW
TYCK0	Timer Y count source select bit[1]	b3 b2 0 0 : f1 0 1 : f8 1 0 : fRING 1 1 : Selects input from CNTR1 pin	RW
TYCK1			RW
TZCK0	Timer Z count source select bit[1]	b5 b4 0 0 : f1 0 1 : f8 1 0 : Selects Timer Y underflow 1 1 : f2	RW
TZCK1			RW
—— (b7-b6)	Reserved bit	Must be set to “0”	RW

Notes:
1. Avoid switching a count source, while a counter is in progress. Timer counter must be stopped before switching a count source.

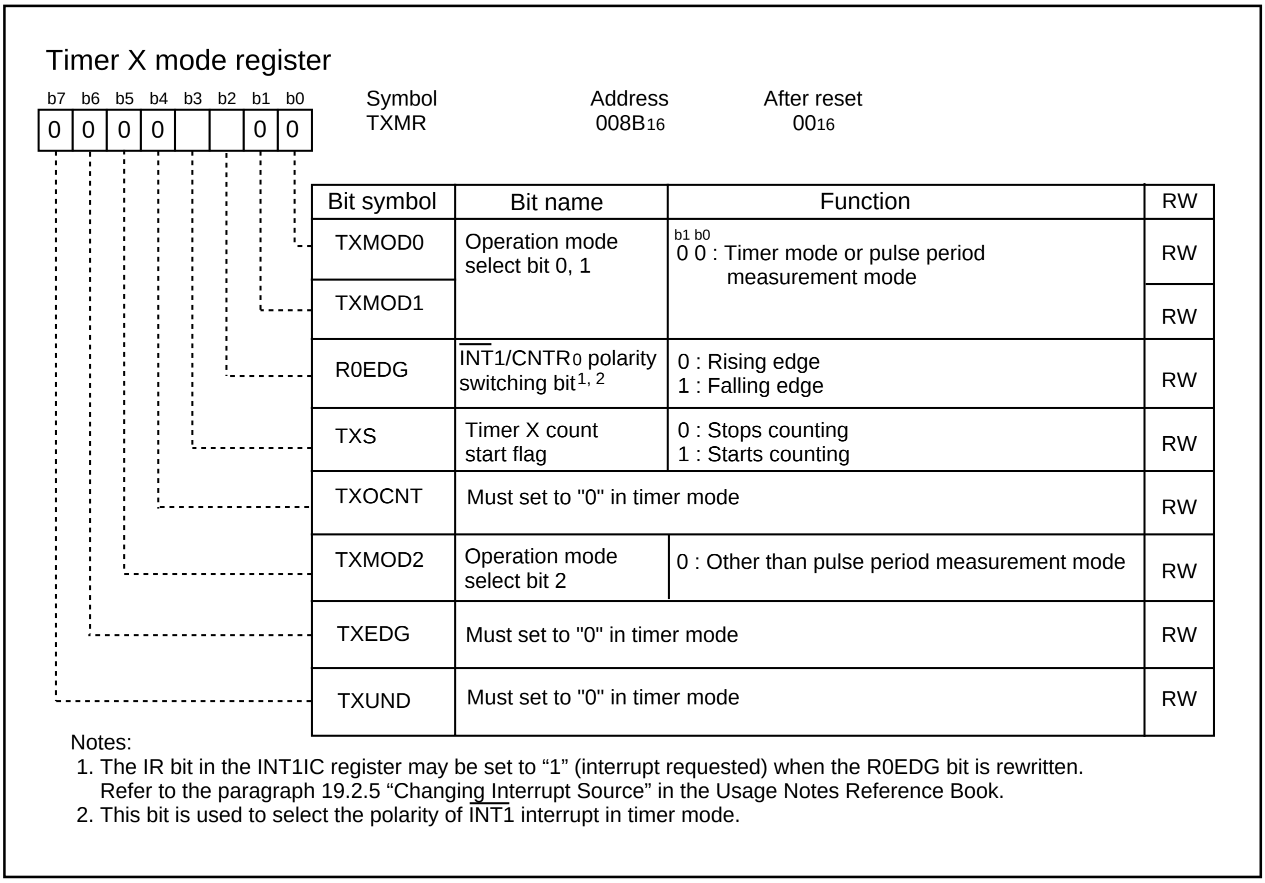
Figure 12.3 PREX Register, TX Register, and TCSS Register

12.1.1 Timer Mode

In this mode, the timer counts an internally generated count source (See “Table 12.2 Timer Mode Specifications”). Figure 12.4 shows the TXMR register in timer mode.

Table 12.2 Timer Mode Specifications

Item	Specification
Count source	f1, f2, f8, f32
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1)(m+1) n: set value of PREX register, m: set value of TX register
Count start condition	Write “1” (count start) to TXS bit in TXMR register
Count stop condition	Write “0” (count stop) to TXS bit in TXMR register
Interrupt request generation timing	When Timer X underflows [Timer X interruption]
$\overline{\mathrm{INT1}}$/CNTR0 pin function	Programmable I/O port, or $\overline{\mathrm{INT1}}$ interrupt input
$\overline{\mathrm{CNTR0}}$ pin function	Programmable I/O port
Read from timer	Count value can be read by reading TX register Same applies to PREX register.
Write to timer	Value written to TX register is written to both reload register and counter. Same applies to PREX register.



Symbol	Address	After reset
TXMR	008B16	0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1	b1 b0 0 0 : Timer mode or pulse period measurement mode	RW
TXMOD1			RW
R0EDG	$\overline{\mathrm{INT1}}$/CNTR0 polarity switching bit[1, 2]	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	Must set to "0" in timer mode		RW
TXMOD2	Operation mode select bit 2	0 : Other than pulse period measurement mode	RW
TXEDG	Must set to "0" in timer mode		RW
TXUND	Must set to "0" in timer mode		RW

Notes:
1. The IR bit in the INT1IC register may be set to “1” (interrupt requested) when the R0EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.
2. This bit is used to select the polarity of $\overline{\mathrm{INT1}}$ interrupt in timer mode.

Figure 12.4 TXMR Register in Timer Mode

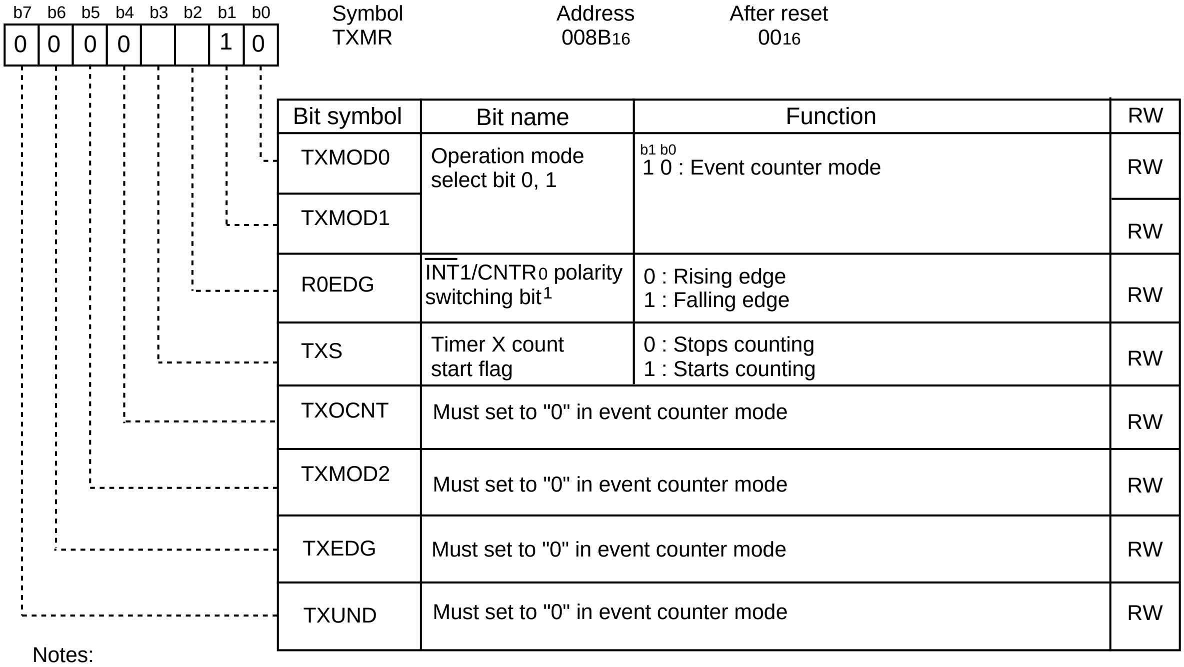
12.1.2 Pulse Output Mode

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 12.3 Pulse Output mode Specifications”). Figure 12.5 shows TXMR register in pulse output mode.

Table 12.3 Pulse Output Mode Specifications

Item	Specification
Count source	f_1, f_2, f_8, f_{32}
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1)(m+1) n: set value of PREX register, m: set value of TX register
Count start condition	Write “1” (count start) to TXS bit in TXMR register
Count stop condition	Write “0” (count stop) to TXS bit in TXMR register
Interrupt request generation timing	• When Timer X underflows [Timer X interruption] • Rising (R0EDG=0) or falling (R0EDG=1) of CNTR$_0$ output [$\overline{\text{INT1}}$ interrupt]
$\overline{\text{INT1}}$/CNTR$_0$ pin function	Pulse output
$\overline{\text{CNTR}_0}$ pin function	Programmable I/O port or inverted output of CNTR$_0$
Read from timer	Count value can be read by reading TX register. Same applies to PREX register.
Write to timer	Value written to TX register is written to both reload register and counter. Same applies to PREX register.
Select function	• Inverted pulse output function The polarity of CNTR0 output pulse can be reversed with TXOCNT bit • $\overline{\text{INT1}}$/CNTR$_0$ polarity switching function Polarity level at starting of pulse output can be selected with R0EDG bit

Timer X mode register



Symbol TXMR Address 008B$_{16}$ After reset 00$_{16}$

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1	b1 b0 1 0 : Event counter mode	RW
TXMOD1			RW
R0EDG	$\overline{\text{INT1}}$/CNTR$_0$ polarity switching bit[1]	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	Must set to "0" in event counter mode		RW
TXMOD2	Must set to "0" in event counter mode		RW
TXEDG	Must set to "0" in event counter mode		RW
TXUND	Must set to "0" in event counter mode		RW

Notes:
1. The IR bit in the INT1IC register may be set to “1” (interrupt requested) when the R0EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

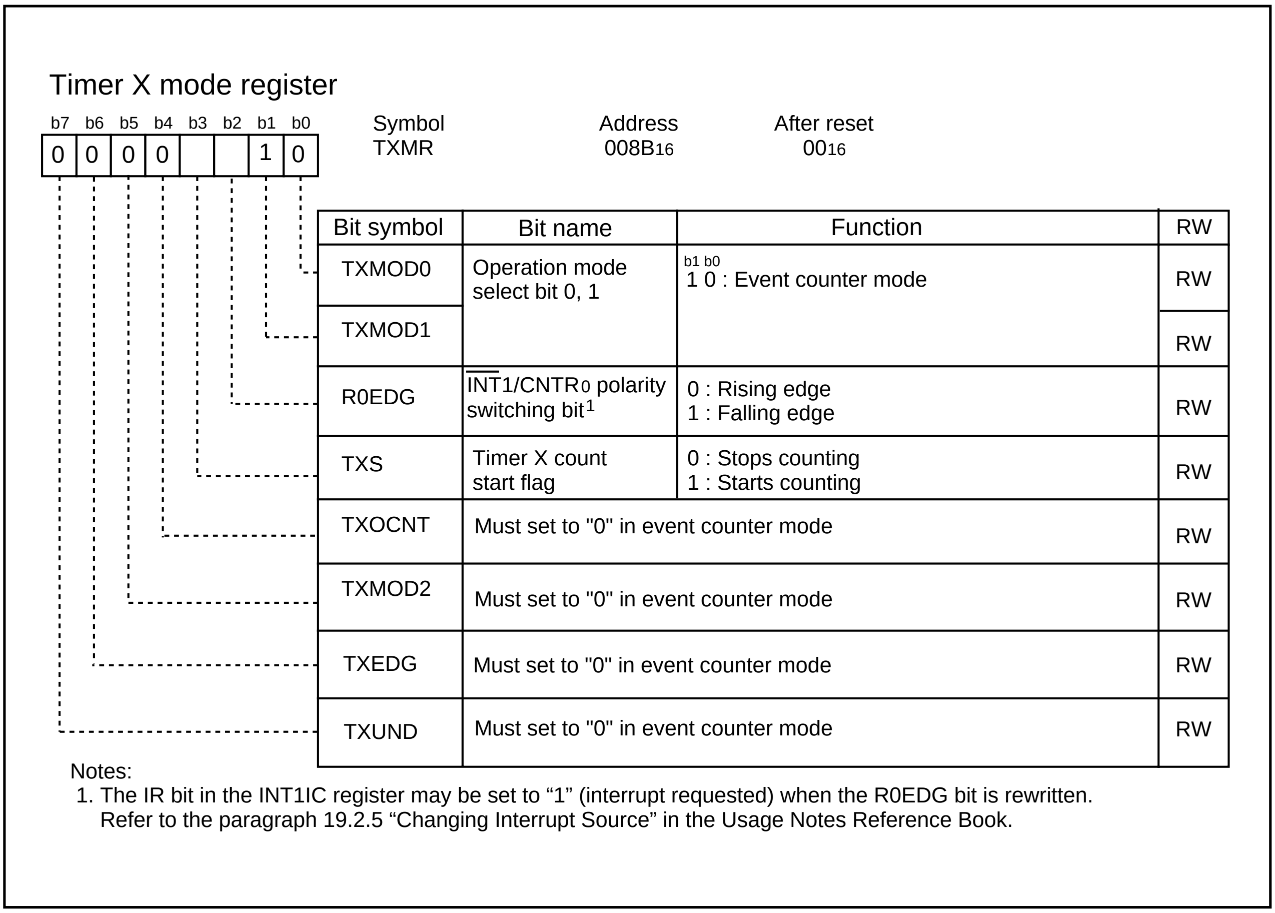
Figure 12.5 TXMR Register in Pulse Output Mode

12.1.3 Event Counter Mode

In this mode, the timer counts an external signal fed to INT1/CNTR0 pin (See “Table 12.4 Event Counter Mode Specifications”). Figure 12.6 shows TXMR register in event counter mode.

Table 12.4 Event Counter Mode Specifications

Item	Specification
Count source	External signals fed to CNTR0 pin (Active edge is selected by program)
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1)(m+1) n: set value of PREX register, m: set value of TX register
Count start condition	Write “1” (count start) to TXS bit in TXMR register
Count stop condition	Write “0” (count stop) to TXS bit in TXMR register
Interrupt request generation timing	• When Timer X underflows [Timer X interrupt] • CNTR0 input count edges [INT1 interrupt]
INT1/CNTR0 pin function	Count source input
CNTR0 pin function	Programmable I/O port
Read from timer	Count value can be read by reading TX register Same applies to PREX register.
Write to timer	Value written to TX register is written to both reload register and counter. Same applies to PREX register.
Select function	• INT1/CNTR0 polarity switching function Active edge of count source can be selected with R0EDG.



Timer X mode register

Symbol TXMR   Address 008B16   After reset 0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1	b1 b0 1 0 : Event counter mode	RW
TXMOD1			RW
R0EDG	INT1/CNTR0 polarity switching bit[1]	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	Must set to "0" in event counter mode		RW
TXMOD2	Must set to "0" in event counter mode		RW
TXEDG	Must set to "0" in event counter mode		RW
TXUND	Must set to "0" in event counter mode		RW

Notes:
1. The IR bit in the INT1IC register may be set to “1” (interrupt requested) when the R0EDG bit is rewritten.
Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

Figure 12.6 TXMR Register in Event Counter Mode

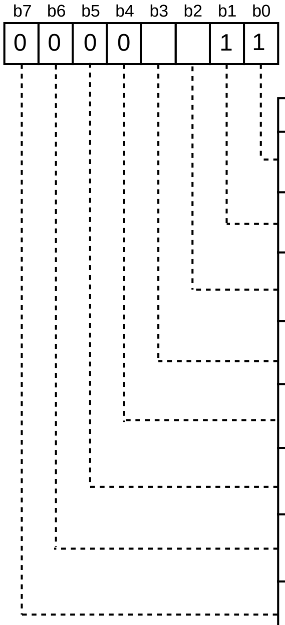
12.1.4 Pulse Width Measurement Mode

In this mode, the timer measures the pulse width of an external signal fed to $\overline{\text{INT1}}$/CNTR0 pin (See “Table 12.5 Pulse Width Measurement Mode Specifications”). Figure 12.7 shows the TXMR register in pulse width measurement mode. Figure 12.8 shows an operation example in pulse width measurement mode.

Table 12.5 Pulse Width Measurement Mode Specifications

Item	Specification
Count source	f_1, f_2, f_8, f_{32}
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	Write “1” (count start) to TXS bit in TXMR register
Count stop condition	Write “0” (count stop) to TXS bit in TXMR register
Interrupt request generation timing	• When Timer X underflows [Timer X interruption] • Rising or falling of CNTR0 input (end of measurement period) [$\overline{\text{INT1}}$ interrupt]
$\overline{\text{INT1}}$/CNTR0 pin function	Measurement pulse input
$\overline{\text{CNTR}_0}$ pin function	Programmable I/O port
Read from timer	Count value can be read by reading TX register Same applies to PREX register.
Write to timer	Value written to TX register is written to both reload register and counter. Same applies to PREX register.
Select function	• $\overline{\text{INT1}}$/CNTR$_0$ polarity switching function Active edge of count source can be selected with R0EDG.

Timer X mode register



Symbol	Address	After reset
TXMR	008B$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1	b1 b0 1 1 : Pulse width measurement mode	RW
TXMOD1			RW
R0EDG	$\overline{\text{INT1}}$/CNTR$_0$ polarity switching bit[1]	0 : Measures “H” level width 1 : Measures “L” level width	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	Must set to "0" in pulse width measurement mode		RW
TXMOD2	Must set to "0" in pulse width measurement mode		RW
TXEDG	Must set to "0" in pulse width measurement mode		RW
TXUND	Must set to "0" in pulse width measurement mode		RW

Notes:
1. IThe IR bit in the INT1IC register may be set to “1” (interrupt requested) when the R0EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

Figure 12.7 TXMR Register in Pulse Width Measurement Mode

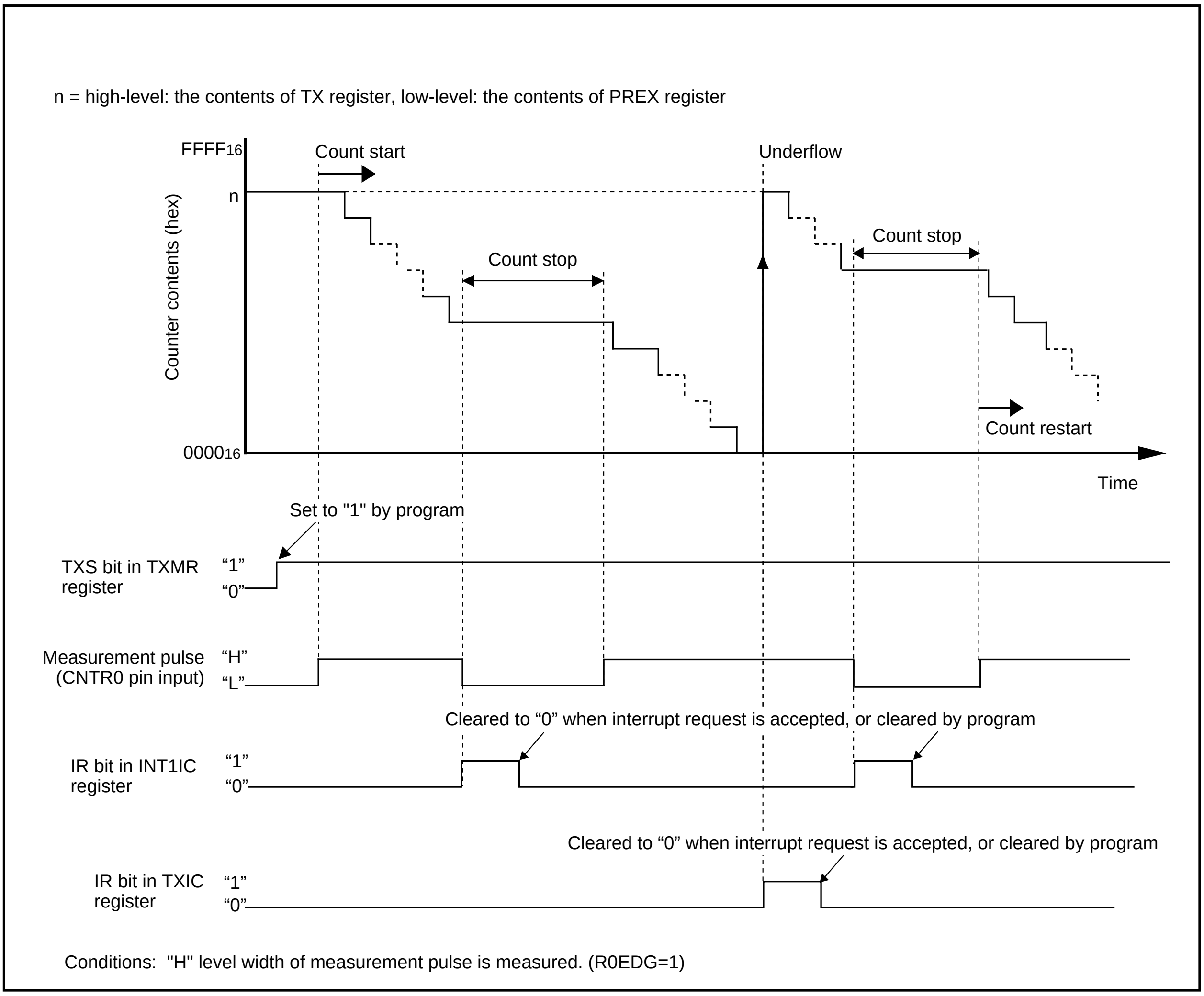

n = high-level: the contents of TX register, low-level: the contents of PREX register
FFFF16
n
000016
Counter contents (hex)
Count start
Underflow
Count stop
Count stop
Count restart
Time
Set to "1" by program
TXS bit in TXMR register
"1"
"0"
Measurement pulse (CNTR0 pin input)
"H"
"L"
Cleared to "0" when interrupt request is accepted, or cleared by program
IR bit in INT1IC register
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by program
IR bit in TXIC register
"1"
"0"
Conditions: "H" level width of measurement pulse is measured. (R0EDG=1)

Figure 12.8 Operation Example in Pulse Width Measurement Mode

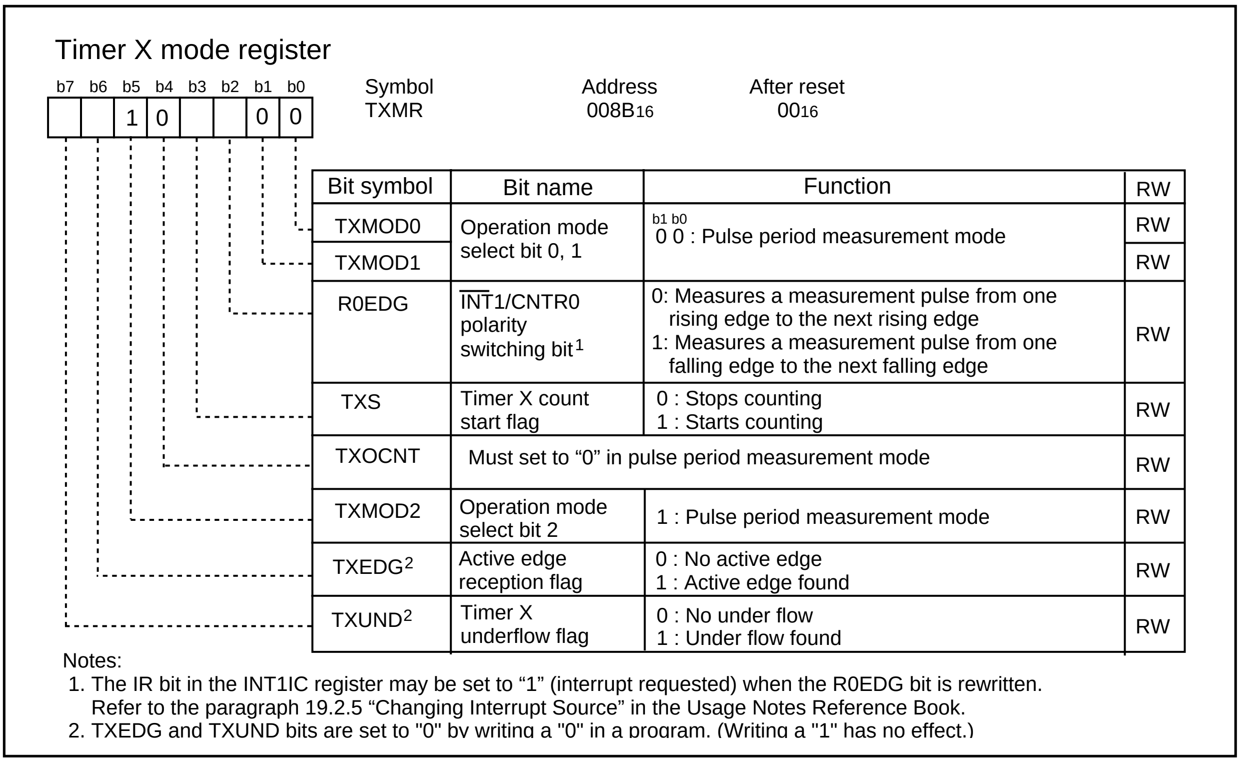
12.1.5 Pulse Period Measurement Mode

In this mode, the timer measures the pulse period of an external signal fed to $\overline{\text{INT1}}$/CNTR0 pin (See “Table 12.6 Pulse Period Measurement Mode Specifications”). Figure 12.9 shows the TXMR register in pulse period measurement mode. Figure 12.10 shows an operation example in pulse period measurement mode.

Table 12.6 Pulse Period Measurement Mode Specifications

Item	Specification
Count source	f_1, f_2, f_8, f_{32}
Count operation	• Down-count • After an active edge of measurement pulse is input, contents in the read-out buffer is retained in the first underflow of prescaler X. Then the timer X reloads contents in the reload register in the second underflow of prescaler X and continues counting.
Count start condition	Write “1” (count start) to TXS bit in TXMR register
Count stop condition	Write “0” (count stop) to TXS bit in TXMR register
Interrupt request generation timing	• When Timer X underflows or reloads [Timer X interrupt] • Rising or falling of CNTR0 input (end of measurement period) [$\overline{\text{INT1}}$ interrupt]
$\overline{\text{INT1}}$/CNTR0 pin function	Measurement pulse input[1]
$\overline{\text{CNTR0}}$ pin function	Programmable I/O port
Read from timer	Contents in the read-out buffer can be read by reading TX register. The value retained in the read-out buffer is released by reading TX register.
Write to timer	Value written to TX register is written to both reload register and counter. Same applies to PREX register.
Select function	• $\overline{\text{INT1}}$/CNTR0 polarity switching function Measurement period of input pulse can be selected with R0EDG bit.

Note: The period of input pulse must be longer than twice the period of prescaler X. Longer pulse for H width and L width than the prescaler X period must be input. If shorter pulse than the period is input to the CNTR0 pin, the input may be disabled.



Timer X mode register

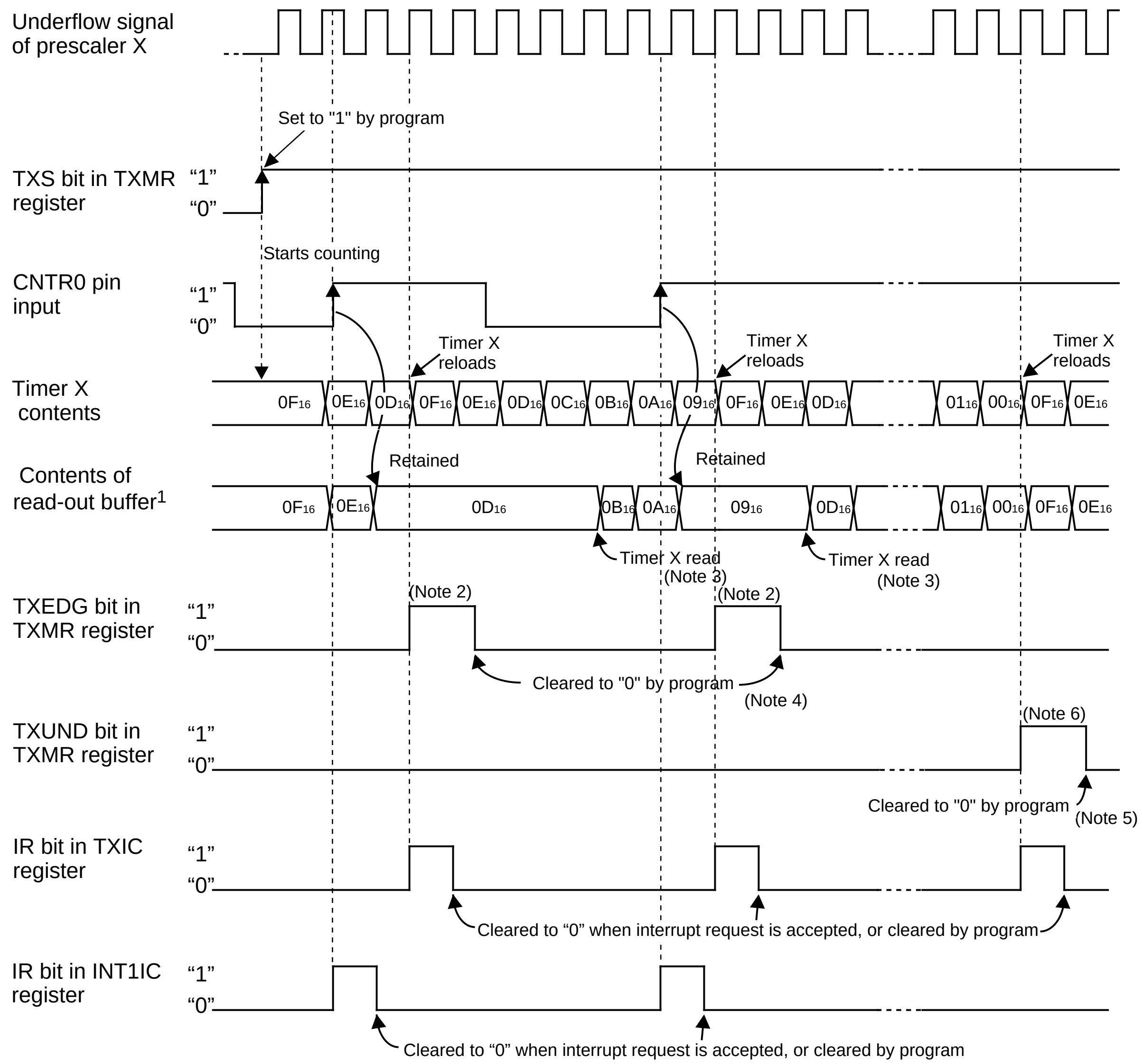
b7	b6	b5	b4	b3	b2	b1	b0
		1	0			0	0

Symbol TXMR   Address 008B16   After reset 0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1	b1 b0 0 0 : Pulse period measurement mode	RW
TXMOD1			RW
R0EDG	$\overline{\text{INT1}}$/CNTR0 polarity switching bit[1]	0: Measures a measurement pulse from one rising edge to the next rising edge 1: Measures a measurement pulse from one falling edge to the next falling edge	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	Must set to “0” in pulse period measurement mode		RW
TXMOD2	Operation mode select bit 2	1 : Pulse period measurement mode	RW
TXEDG[2]	Active edge reception flag	0 : No active edge 1 : Active edge found	RW
TXUND[2]	Timer X underflow flag	0 : No under flow 1 : Under flow found	RW

Notes:
1. The IR bit in the INT1IC register may be set to “1” (interrupt requested) when the R0EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.
2. TXEDG and TXUND bits are set to "0" by writing a "0" in a program. (Writing a "1" has no effect.)

Figure 12.9 TXMR 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) with TX register initial value=0F16.

Notes:

1. The contents of the read-out buffer can be read when the TX register is read in pulse period measurement mode.
2. After an active edge of measurement pulse is input, the TXEDG bit in the TXMR register is set to "1" (active edge found) when the prescaler X underflows for the second time.
3. The TX register should be read before the next active edge is input after the TXEDG bit is set to "1" (active edge found). The contents in the read-out buffer is retained until the TX register is read. If the TX register is not read before the next active edge is input, the measured result of the previous period is retained.
4. When set to "0" by program, use a MOV instruction to write "0" to the TXEDG in the TXMR register. At the same time, write "1" to the TXUND bit.
5. When set to "0" by program, use a MOV instruction to write "0" to the TXUND in the TXMR register. At the same time, write "1" to the TXEDG bit.
6. The TXUND and TXEDG bits are both set to "1" if the timer underflows and reloads on an active edge simultaneously. In this case, the validity of the TXUND bit should be determined by the contents of the read-out buffer.

Figure 12.10 Operation Example in Pulse Period Measurement Mode

12.2 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. Figure 12.11 shows a block diagram of Timer Y. Figures 12.12 to 12.14 show the TYZMR, PREY, TYSC, TYPR, TYZOC, PUM, and YCSS registers.

The Timer Y has two operation modes 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.

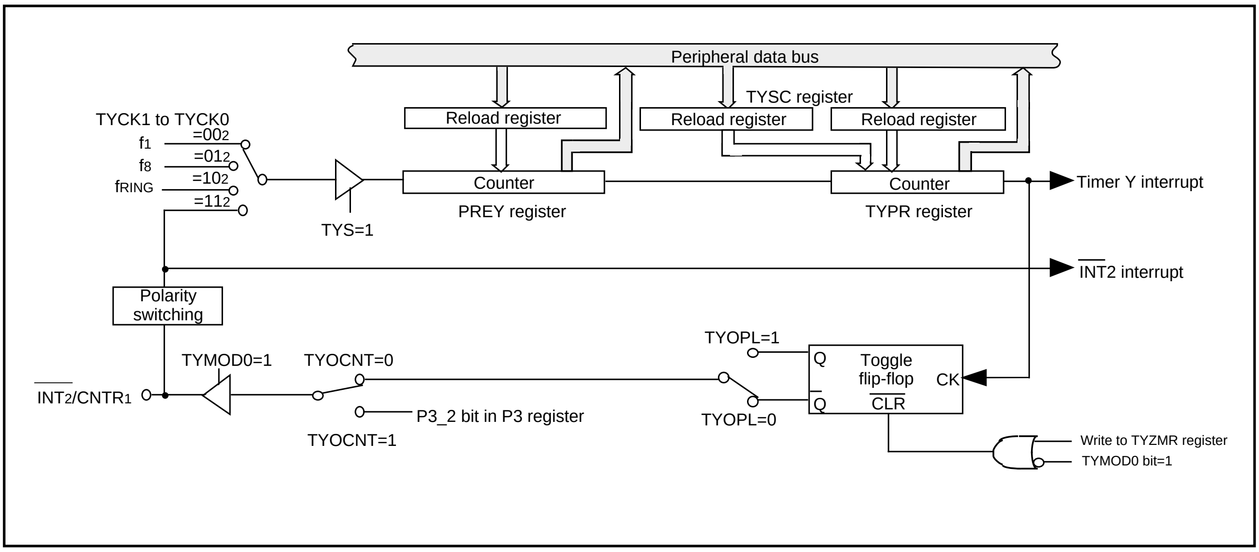


Figure 12.11 Timer Y Block Diagram

Timer Y, Z mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After reset
TYZMR	0080_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode	RW
R1EDG	$\overline{\text{INT2}}$/CNTR1 polarity switching bit[1]	0 : Rising edge 1 : Falling edge	RW
TYWC	Timer Y write control bit	Function varies depending on the operation mode	RW
TYS	Timer Y count start flag	0 : Stops counting 1 : Starts counting	RW
TZMOD0	Timer Z operation mode bit	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 1 : Starts counting	RW

Notes:
1. The IR bit in the INT2IC register may be set to "1" (interrupt requested) when the R1EDG bit is rewritten.
Refer to the paragraph 19.2.5 "Changing Interrupt Source" in the Usage Notes Reference Book.

Figure 12.12 TYZMR Register

Prescaler Y register

b7 b0

Symbol	Address	After reset
PREY	0081_{16}	FF_{16}

Mode	Function	Setting range	RW
Timer mode	Internal count source or CNTR1 input is counted	00_{16} to FF_{16}	RW
Programmable waveform generation mode	Internal count source is counted	00_{16} to FF_{16}	RW

Timer Y secondary register

b7 b0

Symbol	Address	After reset
TYSC	0082_{16}	FF_{16}

Mode	Function	Setting range	RW
Timer mode	Disabled	———	——
Programmable waveform generation mode	Underflow of Prescaler Y is counted[1]	00_{16} to FF_{16}	WO[2]

Notes:

1. The values of TYPR register and TYSC register are reloaded to the counter alternately for counting.
2. The count value can be read out by reading the TYPR register even when the secondary period is being counted.

Timer Y primary register

b7 b0

Symbol	Address	After reset
TYPR	0083_{16}	FF_{16}

Mode	Function	Setting range	RW
Timer mode	Underflow of Prescaler Y is counted	00_{16} to FF_{16}	RW
Programmable waveform generation mode	Underflow of Prescaler Y is counted[1]	00_{16} to FF_{16}	RW

Notes:

1. The values of TYPR register and PYSC register are reloaded to the counter alternately for counting.

Timer Y, Z output control register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After reset
TYZOC	$008A_{16}$	00_{16}

Bit symbol	Bit	Function	RW
TZOS	Timer Z one-shot start bit[1]	0 : Stops one-shot 1 : Starts one-shot	RW
TYOCNT	Timer Y programmable waveform generation output switching bit[2]	0 : Outputs programmable waveform 1 : Outputs the value of $P3_2$ port register	RW
TZOCNT	Timer Z programmable waveform generation output switching bit[2]	0 : Outputs programmable waveform 1 : Outputs the value of $P3_1$ port register	RW
—— (b7-b3)	Nothing is assigned. When write, set to "0". When read, its content is "0".		——

Notes:

1. This bit is set to "0" when the output of one-shot waveform is completed. The TZOS bit should be set to "0" if the one-shot waveform output is terminated by setting the TYS bit in the TYZMR to "0" during the waveform output.
2. This bit is enabled only when operating in programmable waveform generation mode.

Figure 12.13 PREY Register, TYSC Register, TYPR Register, and TYZOC Register

Timer Y, Z waveform output control register

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

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

Bit symbol	Bit name	Function	RW
— (b3-b0)	Reserved bit	Must set to “0”	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	$\overline{\text{INT0}}$ pin one-shot trigger control bit[2] (Timer Z)	0 : $\overline{\text{INT0}}$ pin one-shot trigger invalid 1 : $\overline{\text{INT0}}$ pin one-shot trigger valid	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit[1] (Timer Z)	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	RW

Notes:
1. The INOSEG bit is valid only when the INT0PL bit in the INTEN register is "0" (one-edge).
2. The INOSGT bit must be set to "1" after the INT0EN bit in the INTEN register and the INOSEG bit in the PUM register are set.

Timer count source setting register

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Symbol	Address	After reset
TCSS	$008E_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TXCK0	Timer X count source select bit[1]	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_2	RW
TXCK1			RW
TYCK0	Timer Y count source select bit[1]	b3 b2 0 0 : f_1 0 1 : f_8 1 0 : f_{RING} 1 1 : Selects input from $CNTR_1$ pin	RW
TYCK1			RW
TZCK0	Timer Z count source select bit[1]	b5 b4 0 0 : f_1 0 1 : f_8 1 0 : Selects Timer Y underflow 1 1 : f_2	RW
TZCK1			RW
— (b7-b6)	Reserved bit	Must be set to “0”	RW

Notes:
1. Avoid switching a count source, while a counter is in progress. Timer counter must be stopped before switching a count source.

Figure 12.14 PUM Register and TCSS Register

12.2.1 Timer Mode

In this mode, the timer counts an internally generated count source (see “Table 12.7 Timer Mode Specifications”). An external signal input to the CNTR1 pin can be counted. The TYSC register is unused in timer mode. Figure 12.15 shows the TYZMR and PUM registers in timer mode.

Table 12.7 Timer Mode Specifications

Item	Specification
Count source	f_1, f_8, f_{RING}, external signal fed to CNTR1 pin
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.)
Divide ratio	fi/(n+1)(m+1) n: set value in PREY register, m: set value in TYPR register
Count start condition	Write “1” (count start) to TYS bit in TYZMR register
Count stop condition	Write “0” (count stop) to TYS bit in TYZMR register
Interrupt request generation timing	• When Timer Y underflows [Timer Y interrupt] • Rising or falling of $\overline{\text{INT2}}$/CNTR$_1$ input [$\overline{\text{INT2}}$ interrupt]
$\overline{\text{INT2}}$/CNTR$_1$ pin function	Programmable I/O port, count source input or $\overline{\text{INT2}}$ interrupt
Read from timer	Count value can be read out by reading TYPR register. Same applies to PREY register.
Write to timer[1]	Value written to TYPR register is written to both reload register and counter or written to only reload register. Selected by program. Same applies to PREY register.
Select function	• Event counter function When setting TYCK1 to TYCK0 bits to “11$_2$”, an external signal fed to CNTR1 pin is counted. • $\overline{\text{INT2}}$/CNTR1 switching bit Active edge of count source is selected by R1EDG bit.

Notes:

1. The IR bit in the TYIC register is set to "1" (interrupt requested) if you write to the TYPR or PREY register while both of the following conditions are met.

   Conditions:

   - TYWC bit in TYZMR register is "0" (write to reload register and counter simultaneously)
   - TYS bit is "1" (count start)

   To write to the TYPR or PREY register in the above state, disable interrupts before writing.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
							0

Symbol	Address	After reset
TYZMR	0080_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y operation mode bit	0 : Timer mode	RW
R1EDG	$\overline{\text{INT2}}$/CNTR1 polarity switching bit[1]	0 : Rising edge 1 : Falling edge	RW
TYWC	Timer Y write control bit[2]	0 : Write to reload register and counter simultaneously 1 : Write to reload register	RW
TYS	Timer Y count start flag	0 : Stops counting 1 : Starts counting	RW
TZMOD0	Timer Z-related bit		RW
TZMOD1			RW
TZWC			RW
TZS			RW

Notes:

1. The IR bit in the INT2IC register may be set to “1” (interrupt requested) when the R1EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.
2. When TYS bit=1 (starts counting), the value set in the TYWC bit is valid. If TYWC bit=0, the timer Y count value is written to both reload register and counter. If TYWC bit=1, the timer Y count value is written to the reload register only.
   When TYS bit=0 (stops counting), the timer Y count value is written to both reload register and counter regardless of how the TYWC bit is set.

Timer Y, Z waveform output control register

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

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

Bit symbol	Bit name	Function	RW
— (b3-b0)	Reserved bit	Must set to “0”	RW
TYOPL	Timer Y output level latch	Invalid in timer mode	RW
TZOPL	Timer Z-related bits		RW
INOSTG			RW
INOSEG			RW

Figure 12.15 TYZMR Register and PUM Register in Timer Mode

12.2.2 Programmable Waveform Generation Mode

In this mode, an signal output from the TYOUT pin is inverted each time the counter underflows, while the values in the TYPR register and TYSC register are counted alternately (see "Table 12.8 Programmable Waveform Generation Mode Specifications"). A counting starts by counting the set value in the TYPR register. Figure 12.16 shows the TYZMR register in programmable waveform generation mode. Figure 12.17 shows the operation example.

Table 12.8 Programmable Waveform Generation Mode Specifications

Item	Specification
Count source	f_1, f_8, f_{RING}
Count operation	• Down count • When the timer underflows, it reloads the contents of primary reload register and secondary reload register alternately before continuing counting.
Period	fi/(n+1)((m+1)+(p+1)) n: set value in PREY register, m: set value in TYPR register, p: set value in TYSC register
Count start condition	Write "1" (count start) to TYS bit in TYZMR register
Count stop condition	Write "0" (count stop) to TYS bit in TYZMR register
Interrupt request generation timing	In half of count source, after timer Y underflows during secondary period (at the same time as the CNTR, output change) [Timer Y interrupt]
$\overline{\text{INT}}$2/CNTR1 pin functions	Pulse output[1]
Read from timer	Count value can be read out by reading TYPR register. Same applies to PREY register[2].
Write to timer	Value written to TYPR register is written to only reload register. Same applies to TYSC register and PREY register[3].
Select function	• Output level latch select function The output level during primary and secondary periods is selected by the TYOPL bit. • Programmable waveform generation output switching function When the TYOCNT bit in the TYZOC register is set to "0", the output from TYOUT is inverted synchronously when Timer Y underflows during the secondary period. And when set to "1", a value in the P3_2 bit is output from TYOUT synchronously when Timer Y underflows during the secondary period[4].

Notes:

1. When the counting stopped, the output level is that in the secondary period.
2. Even when counting the secondary period, read out the TYPR register.
3. The set value in the TYPR register and TYSC register are made effective by writing a value to the TYPR register. The written values are reflected to the waveform output from the next primary period after writing to the TYPR register.
4. The output is switched in sync with timer Y underflow in the secondary period.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
					1		1

Symbol TYZMR Address 0080_{16} After reset 00_{16}

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y operation mode bit	1 : Programmable waveform generation mode	RW
R1EDG	$\overline{\mathrm{INT2}}$/CNTR1 polarity switching bit[1]	0 : Rising edge 1 : Falling edge	RW
TYWC	Timer Y write control bit	Must set to “1” in programmable waveform generation mode[2].	RW
TYS	Timer Y count start flag	0 : Stops counting 1 : Starts counting	RW
TZMOD0	Timer Z-related bit		RW
TZMOD1			RW
TZWC			RW
TZS			RW

Notes:
1. The IR bit in the INT2IC register may be set to “1” (interrupt requested) when the R1EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.
2. When TYS bit= 1 (starts counting), the timer Y count value is written to the reload register only. When TYS bit=0 (stops counting), the timer Y count value is written to both reload register and counter.

Timer Y, Z waveform output control register

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

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

Bit symbol	Bit name	Function	RW
— (b3-b0)	Reserved bit	Must set to “0”	RW
TYOPL	Timer Y output level latch	0 : Outputs "H" for primary period Outputs "L" for secondary period Outputs "L" when the timer is stopped 1 : Outputs "L" for primary period Outputs "H" for secondary period Outputs "H" when the timer is stopped	RW
TZOPL	Timer Z-related bits		RW
INOSTG			RW
INOSEG			RW

Figure 12.16 TYZMR Register and PUM Register in Programmable Waveform Generation Mode

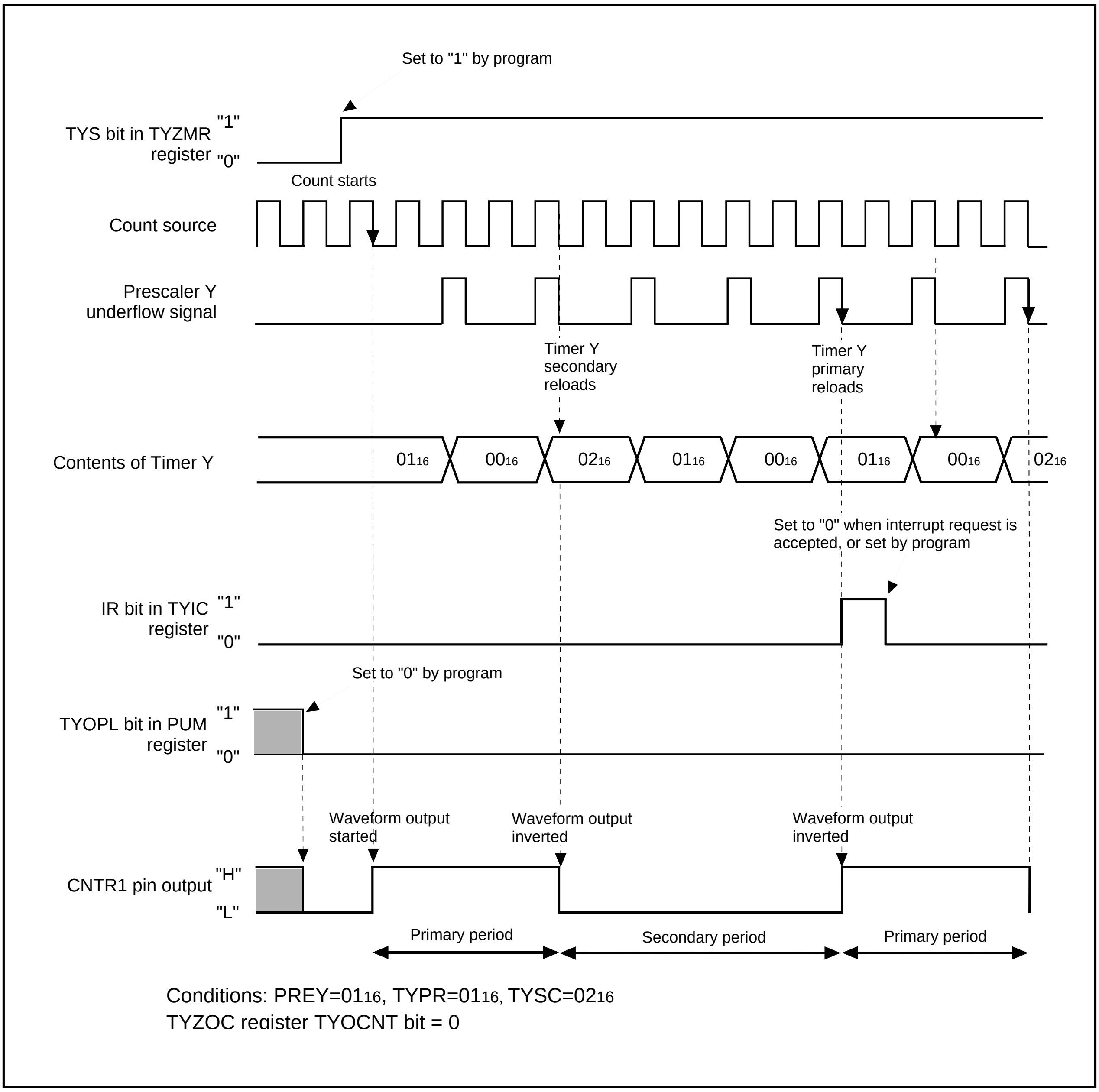


Figure 12.17 Timer Y Operation Example in Programmable Waveform Generation Mode

12.3 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. Figure 12.18 shows a block diagram of Timer Z. Figures 12.19 to 12.21 show the TYZMR, PREZ, TZSC, TZPR, TYZOC, PUM, and TCSS registers.

Timer Z has the following four operation modes.

- 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.

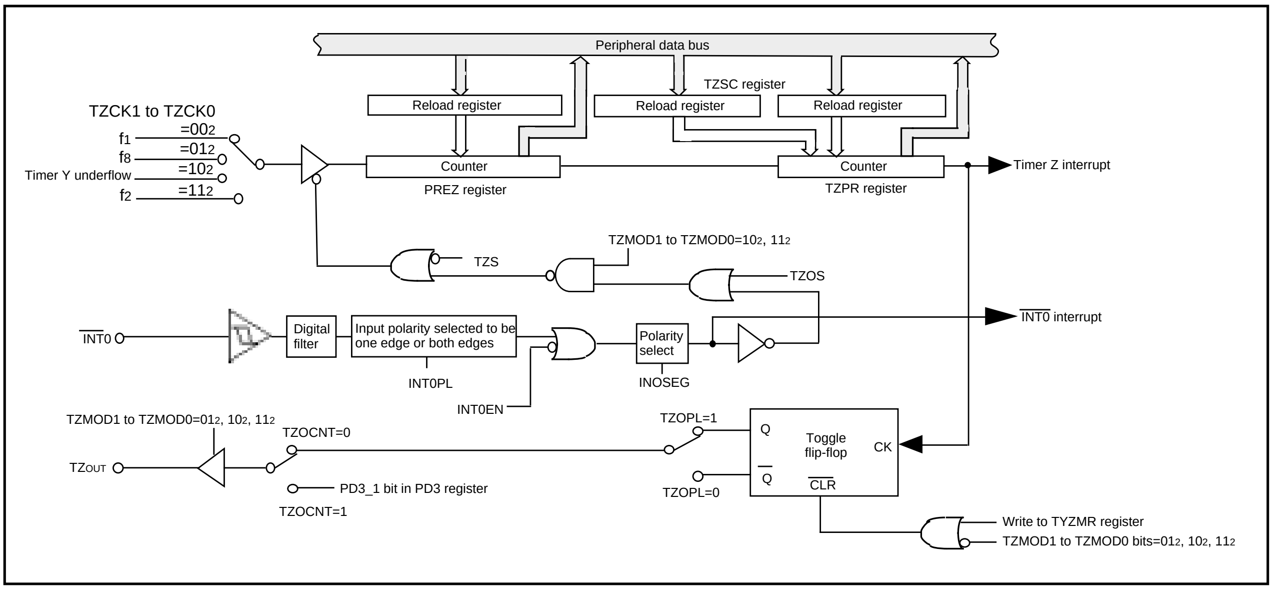


Figure 12.18 Timer Z Block Diagram

Timer Y, Z mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol TYZMR Address 0080_{16} After reset 00_{16}

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y operation mode bit	0 : Timer mode 1 : Programmable waveform generation mode	RW
R1EDG	INT2/CNTR1 polarity switching bit[1]	0 : Rising edge 1 : Falling edge	RW
TYWC	Timer Y write control bit	Function varies depending on the operation mode	RW
TYS	Timer Y count start flag	0 : Stops counting 1 : Starts counting	RW
TZMOD0	Timer Z operation mode bit	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 1 : Starts counting	RW

Notes:

1. The IR bit in the INT2IC register may be set to “1” (interrupt requested) when the R1EDG bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

Figure 12.19 TYZMR Register

Prescaler Z register

b7 b0

Symbol	Address	After reset
PREZ	0085_{16}	FF_{16}

Mode	Function	Setting range	RW
Timer mode	Internal count source or Timer Y underflow is counted	00_{16} to FF_{16}	RW
Programmable waveform generation mode	Internal count source or Timer Y underflow is counted	00_{16} to FF_{16}	RW
Programmable one-shot generation mode	Internal count source or Timer Y underflow is counted	00_{16} to FF_{16}	RW
Programmable wait one-shot generation mode	Internal count source or Timer Y underflow is counted	00_{16} to FF_{16}	RW

Timer Z Secondary register

b7 b0

Symbol	Address	After reset
TZSC	0086_{16}	FF_{16}

Mode	Function	Setting range	RW
Timer mode	Invalid	———	—
Programmable waveform generation mode	Underflow of Prescaler Z is counted[1]	00_{16} to FF_{16}	WO[2]
Programmable one-shot generation mode	Invalid	———	—
Programmable wait one-shot generation mode	Underflow of Prescaler Z is counted (One-shot width is counted)	00_{16} to FF_{16}	WO

Notes:
1. Each value in the TZPR register and TZSC register is reloaded to the counter alternately for counting.
2. The count value can be read out by reading the TZSC register even when the secondary period is being counted.

Timer Z Primary register

b7 b0

Symbol	Address	After reset
TZPR	0087_{16}	FF_{16}

Mode	Function	Setting range	RW
Timer mode	Underflow of Prescaler Z is counted	00_{16} to FF_{16}	RW
Programmable waveform generation mode	Underflow of Prescaler Z is counted[1]	00_{16} to FF_{16}	RW
Programmable one-shot generation mode	Underflow of Prescaler Z is counted (One-shot width is counted)	00_{16} to FF_{16}	RW
Programmable wait one-shot generation mode	Underflow of Prescaler Z is counted (Wait period is counted)	00_{16} to FF_{16}	RW

Notes:
1. Each value in the TZPR register and TZSC register is reloaded to the counter alternately for counting.

Timer Y, Z output control register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After reset
TYZOC	$008A_{16}$	00_{16}

Bit symbol	Bit	Function	RW
TZOS	Timer Z one-shot start bit[1]	0 : Stops one-shot 1 : Starts one-shot	RW
TYOCNT	Timer Y programmable waveform generation output switching bit[2]	0 : Outputs programmable waveform 1 : Outputs the value of $P3_2$ port register	RW
TZOCNT	Timer Z programmable waveform generation output switching bit[2]	0 : Outputs programmable waveform 1 : Outputs the value of $P3_1$ port register	RW
—— (b7-b3)	Nothing is assigned. When write, set to "0". When read, its content is "0".		—

Notes:
1. This bit is set to "0" when the output of one-shot waveform is completed. The TZOS bit should be set to "0" if the one-shot waveform output is terminated by setting the TYS bit in the TYZMR to "0" during the waveform output.
2. This bit is enabled only when operating in programmable waveform generation mode.

Figure 12.20 PREZ Register, TZSC Register, TZPR Register, and TYZOC Register

Timer Y, Z waveform output control register

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

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

Bit symbol	Bit name	Function	RW
— (b3-b0)	Reserved bit	Must set to “0”	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	$\overline{\text{INT0}}$ pin one-shot trigger control bit[2] (Timer Z)	0 : $\overline{\text{INT0}}$ pin one-shot trigger invalid 1 : $\overline{\text{INT0}}$ pin one-shot trigger valid	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit[1] (Timer Z)	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	RW

Notes:

1. The INOSEG bit is valid only when the INT0PL bit in the INTEN register is "0" (one-edge).
2. The INOSGT bit must be set to "1" after the INT0EN bit in the INTEN register and the INOSEG bit in the PUM register are set.

Timer count source setting register

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Symbol	Address	After reset
TCSS	$008E_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TXCK0	Timer X count source select bit[1]	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_2	RW
TXCK1			RW
TYCK0	Timer Y count source select bit[1]	b3 b2 0 0 : f_1 0 1 : f_8 1 0 : f_{RING} 1 1 : Selects input from $CNTR_1$ pin	RW
TYCK1			RW
TZCK0	Timer Z count source select bit[1]	b5 b4 0 0 : f_1 0 1 : f_8 1 0 : Selects Timer Y underflow 1 1 : f_2	RW
TZCK1			RW
— (b7-b6)	Reserved bit	Must be set to “0”	RW

Notes:

1. Avoid switching a count source, while a counter is in progress. Timer counter must be stopped before switching a count source.

Figure 12.21 PUM Register and TCSS Register

12.3.1 Timer Mode

In this mode, the timer counts an internally generated count source or Timer Y underflow (see “Table 12.9 Timer Mode Specifications”). The Timer Z secondary is unused in timer mode. Figure 12.22 shows the TYZMR register and PUM register in timer mode.

Table 12.9 Timer Mode Specifications

Item	Specification
Count source	f_1, f_2, f_8, Timer Y underflow
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.)
Divide ratio	fi/(n+1)(m+1) n: set value in PREZ register, m: set value in TZPR register
Count start condition	Write “1” (count start) to TZS bit in TYZMR register
Count stop condition	Write “0” (count stop) to TZS bit in TYZMR register
Interrupt request generation timing	• When Timer Z underflows [Timer Z interrupt] • Rising, falling, or both edges of $\overline{\text{INT0}}$ pin input [$\overline{\text{INT0}}$ interrupt]
TZOUT pin function	Programmable I/O port
$\overline{\text{INT0}}$ pin function	Programmable I/O port, or external interrupt input pin
Read from timer	Count value can be read out by reading TZPR register. Same applies to PREZ register.
Write to timer[1]	Value written to TZPR register is written to both reload register and counter or written to reload register only. Selected by program. Same applies to PREZ register.

Notes:

1. The IR bit in the TZIC register is set to "1" (interrupt requested) if you write to the TZPR or PREZ register while both of the following conditions are met.

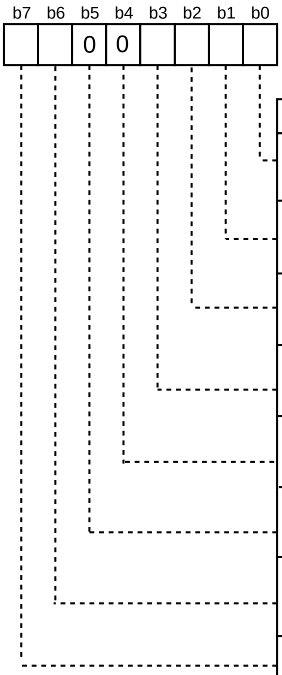
   <Conditions>

   • TZWC bit in TYZMR register is set to "0" (write to reload register and counter simultaneously)

   • TZS bit in TYZMR register is set to "1" (count start)

   To write to the TZPR or PREZ register in the above state, disable interrupts before the writing.

Timer Y, Z mode register



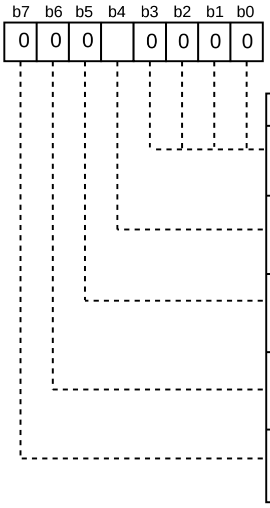
Symbol TYZMR | Address 0080_{16} | After reset 00_{16}

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y-related bit		RW
R1EDG			RW
TYWC			RW
TYS			RW
TZMOD0	Timer Z operation mode bit	b5 b4 0 0 : Timer mode	RW
TZMOD1			RW
TZWC	Timer Z write control bit[1]	0 : Write to reload register and counter 1 : Write to reload register only	RW
TZS	Timer Z count start flag	0 : Stops counting 1 : Starts counting	RW

Notes:

1. When TZS bit=1 (starts counting), the value set in the TZWC bit is valid. If TZWC bit=0, the timer Z count value is written to both reload register and counter. If TZWC bit=1, the timer Z count value is written to the reload register only.
When TZS bit=0 (stops counting), the timer Z count value is written to both reload register and counter regardless of how the TZWC bit is set.

Timer Y, Z waveform output control register



Symbol PUM | Address 0084_{16} | After reset 00_{16}

Bit symbol	Bit name	Function	RW
(b3-b0)	Reserved bit	Must set to “0”	RW
TYOPL	Timer Y-related bit		RW
TZOPL	Timer Z output level latch	Must set to “0” in timer mode	RW
INOSTG	$\overline{\text{INT0}}$ pin one-shot trigger control bit	Must set to “0” in timer mode	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit	Must set to “0” in timer mode	RW

Figure 12.22 TYZMR Register and PUM Register in Timer Mode

12.3.2 Programmable Waveform Generation Mode

In this mode, an signal output from the TZOUT pin is inverted each time the counter underflows, while the values in the TZPR register and TZSC register are counted alternately (see "Table 12.10 Programmable Waveform Generation Mode Specifications"). A counting starts by counting the value set in the TZPR register. Figure 12.23 shows TYZMR and PUM registers in this mode. The Timer Z operates in the same way as the Timer Y in this mode. See Figure 12.17 (Timer Y operation example in programmable waveform generation mode).

Table 12.10 Programmable Waveform Generation Mode Specifications

Item	Specification
Count source	f_1, f_2, f_8, Timer Y underflow
Count operation	• Down-count • When the timer underflows, it reloads the contents of primary reload register and secondary reload register alternately before continuing counting.
Period	fi/(n+1)((m+1)+(p+1)) n: set value in PREZ register, m: set value in TZPR register, p: set value in TZSC register
Count start condition	Write "1" (count start) to the TZS bit in the TYZMR register
Count stop condition	Write "0" (count stop) to the TZS bit in the TYZMR register
Interrupt request generation timing	In half of count source, after timer Z underflows during secondary period (at the same time as the TZout output change) [Timer Z interrupt]
TZOUT pin function	Pulse output[1]
$\overline{\text{INT0}}$ pin functions	Programmable I/O port, or external interrupt input pin
Read from timer	Count value can be read out by reading TZPR register. Same applies to PREZ register[2].
Write to timer	Value written to TZPR register is written to reload register only. Same applies to TZSC register and PREZ register[3].
Select function	• Output level latch select function The output level during primary and secondary periods is selected by the TZOPL bit. • Programmable waveform generation output switching function When the TZOCNT bit in the TYZOC register is set to "0", the output from TZOUT is inverted synchronously when the Timer Z underflows during the secondary period. And when set to "1", a value in the P3_1 bit is output from TZOUT synchronously when the Timer Z underflows during the secondary period[4].

Notes:

1. When the counting stopped, the output level is that in the secondary period.
2. Even when counting the secondary period, read out the TZPR register.
3. The set value in the TZPR register and TZSC register are made effective by writing a value to the TZPR register. The set values are reflected to the waveform output beginning with the next primary period after writing to the Timer Z primary register.
4. The output is switched in sync with timer Z underflow in the secondary period.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
	1	0	1				

Symbol	Address	After reset
TYZMR	0080_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y-related bit		RW
R1EDG			RW
TYWC			RW
TYS			RW
TZMOD0	Timer Z operation mode bit	b5 b4 0 1 : Programmable waveform generation mode	RW
TZMOD1			RW
TZWC	Timer Z write control bit	Set to "1" in programmable waveform generation mode[1]	RW
TZS	Timer Z count start flag	0 : Stops counting 1 : Starts counting	RW

Notes:
1. When TZS bit=1(starts counting), the timer Y count vaue is written to the reload register only.
When TZS bit=0(stops counting), the timer Y count value is written to both reload register and counter.

Timer Y, Z waveform output control register

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

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

Bit symbol	Bit name	Function	RW
(b3-b0)	Reserved bit	Must set to “0”	RW
TYOPL	Timer Y-related bit		RW
TZOPL	Timer Z output level latch	0 : Outputs "H" for primary period Outputs "L" for secondary period Outputs "L" when the timer is stopped 1 : Outputs "L" for primary period Outputs "H" for secondary period Outputs "H" when the timer is stopped	RW
INOSTG	$\overline{\text{INT0}}$ pin one-shot trigger control bit	Must set to “0” in programmable waveform generation mode	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit	Must set to “0” in programmable waveform generation mode	RW

Figure 12.23 TYZMR Register and PUM Register in Programmable Waveform Generation Mode

12.3.3 Programmable One-shot Generation Mode

In this mode, upon program command or external trigger input (input to the INT0 pin), the microcomputer outputs the one-shot pulse from the TZOUT pin (see “Table 12.11 Programmable One-shot Generation Mode Specifications”). When a trigger occurs, the timer starts operating from the point only once for a given period equal to the set value in the TZPR register. The TZSC is unused in this mode. Figure 12.24 shows the TYZMR register and PUM register in this mode. Figure 12.25 shows an operation example in this mode.

Table 12.11 Programmable One-shot Generation Mode Specifications

Item	Specification
Count source	f_1, f_2, f_8, Timer Y underflow
Count operation	• Downcounts set value in TZPR register • 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	fi/(n+1)(m+1) n: set value in PREZ register, m: set value in TZPR register
Count start condition	• Set TZOS bit in TYZOC register to “1” (start one-shot) [1] • Input active trigger to INT0 pin[2]
Count stop condition	• When reloading is completed after count value was set to "00_{16}" • When TZS bit in TYZMR register is set to “0” (stop counting) • When TZOS bit in TYZOC register is set to “0” (stop one-shot)
Interrupt request generation timing	In half cycles of count source, after the timer underflows (at the same time as the TZout output ends) [Timer Z interrupt]
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 TZPR register. Same applies to PREZ register.
Write to timer	Value written to TZPR register is written to reload register only[3]. Same applies to PREZ register.
Select function	• Output level latch select function Output level for one-shot pulse waveform is selected by TZOPL bit. • INT0 pin one-shot trigger control function and polarity select function Trigger input from INT0 pin can be set to active or inactive by INOSTG bit. Also, an active trigger's polarity can be selected by INOSEG bit.

Notes:

1. The TZS bit in the TYZMR register must be set to "1" (start counting).
2. The TZS bit must be set to "1" (start counting), the INT0EN bit in the INTEN register to "1" (enabling INT0 input), and the INOSTG bit in the PUM register to "1" (enabling INT0 one-shot trigger).
3. The set values are reflected beginning with the next one-shot pulse after writing to the TZPR register.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
	1	1	0				

Symbol	Address	After reset
TYZMR	0080$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y-related bit		RW
R1EDG			RW
TYWC			RW
TYS			RW
TZMOD0	Timer Z operation mode bit	b5 b4 1 0 : Programmable one-shot generation mode	RW
TZMOD1			RW
TZWC	Timer Z write control bit	Set to "1" in programmable one-shot generation mode[1]	RW
TZS	Timer Z count start flag	0 : Stops counting 1 : Starts counting	RW

Notes:
1. When TZS bit=1(starts counting), the timer Y count value is written to the reload register only.
When TZS bit=0(stops counting), the timer Y count value is written to both reload and counter.

Timer Y, Z waveform output control register

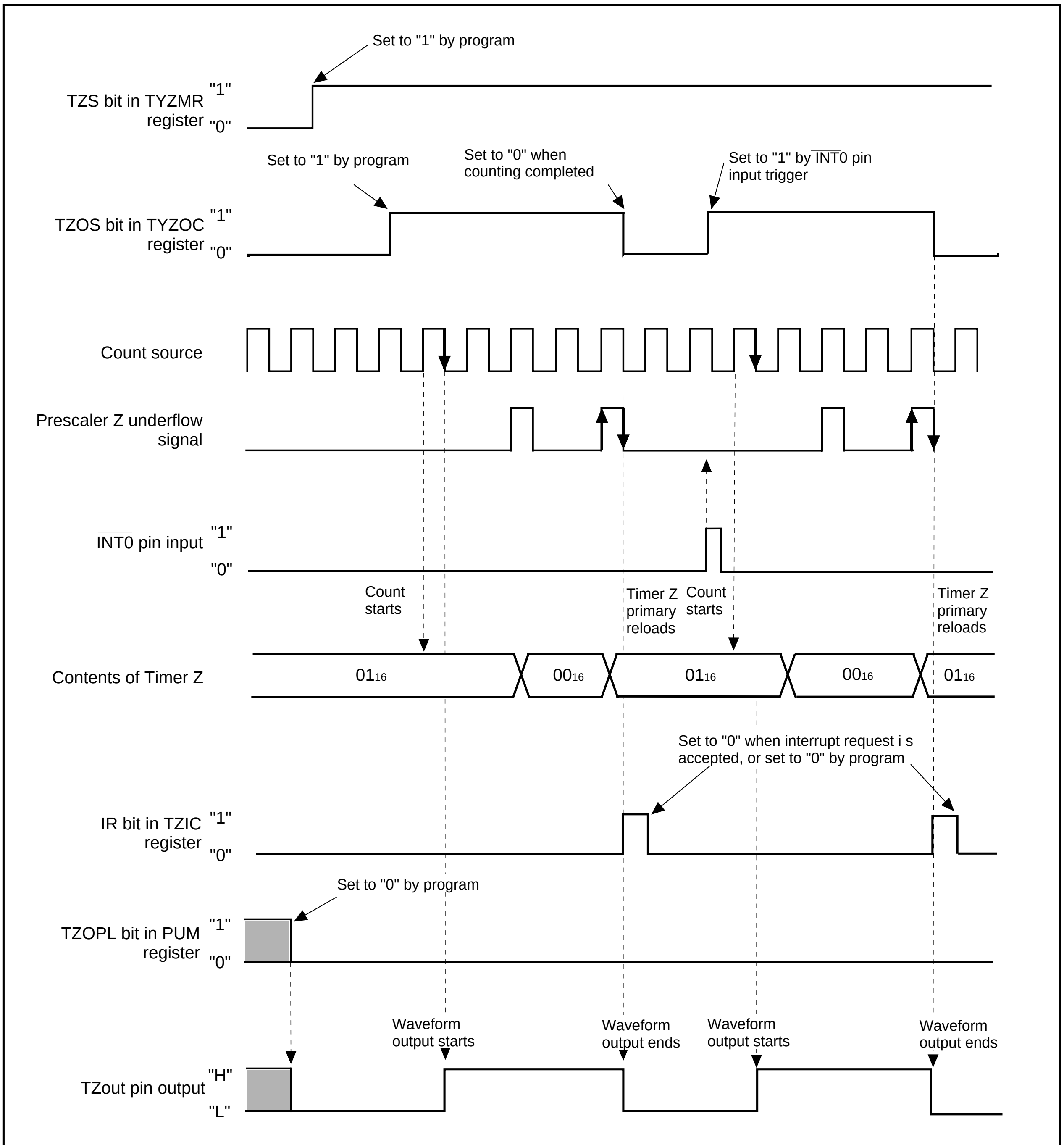
b7	b6	b5	b4	b3	b2	b1	b0
				0	0	0	0

Symbol	Address	After reset
PUM	0084$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
— (b3-b0)	Reserved bit	Must set to “0”	RW
TYOPL	Timer Y-related bit		RW
TZOPL	Timer Z output level latch	0 : Outputs "H" level one-shot pulse. Outputs "L" when the timer is stopped. 1 : Outputs "L" level one-shot pulse Outputs "H" when the timer is stopped.	RW
INOSTG	$\overline{\text{INT0}}$ pin one-shot trigger control bit[2]	0 : $\overline{\text{INT0}}$ pin one-shot trigger disabled 1 : $\overline{\text{INT0}}$ pin one-shot trigger enabled[2]	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit[1]	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	RW

Notes:
1. The INOSEG bit is valid only when the INT0PL bit in the INTEN register is set to "0" (one-edge).
2. The INOSGT bit must be set to “1” after the INT0EN bit in the INTEN register and the INOSEG bit in the PUM register are set.
When setting the INOSTG bit to "1" ($\overline{\text{INT0}}$ pin one-shot trigger enabled), the INT0F0 and INT0F1 bits in the INT0F register must be set.
The INOSTG bit must be set to “0” ($\overline{\text{INT0}}$ pin one-shot trigger disabled) after the TZS bit in the TYZMR register is set to “0” (count stop).

Figure 12.24 TYZMR Register and PUM Register in Programmable One-shot Generation Mode



Conditions: PREZ=01$_{16}$, TZPR=03$_{16}$
PUM register TZOPL bit=0, INOSTG bit=1 (INT0 one-shot trigger enabled)
INOSEG bit=1 (edge trigger at rising edge)

Figure 12.25 Operation Example in Programmable One-shot Generation Mode

12.3.4 Programmable Wait One-shot Generation Mode

In this mode, upon program or external trigger input (input to the $\overline{\text{INT0}}$ pin), the microcomputer outputs the one-shot pulse from the TZOUT pin after waiting for a given length of time (see “Table 12.12 Programmable Wait One-shot Generation Mode Specifications”). When a trigger occurs, from this point, the timer starts outputting pulses only once for a given length of time equal to the set value in the TZSC register after waiting for a given length of time equal to the set value in the TZPR register. Figure 12.26 shows the TYZMR and PUM registers in this mode. Figure 12.27 shows an operation example in this mode.

Table 12.12 Programmable Wait One-shot Generation Mode Specifications

Item	Specification
Count source	f_1, f_2, f_8, Timer Y underflow
Count operation	• Downcounts set value in Timer Z primary • When a counting of TZPR register underflows, the timer reloads the contents of TZSC register before continuing counting. • When a counting of TZSC register underflows, the timer reloads the contents of TZPR register before stopping counting. • When a counting stops, the timer reloads the contents of the reload register before it stops.
Wait time	fi/(n+1)(m+1) n: set value in PREZ register, m: set value in TZPR register
One-shot pulse output time	fi/(n+1)(p+1) n : set value in PREZ, p: set value in TZSC register
Count start condition	• Set TZOS bit in TYZOC register to “1” (start one-shot)[1] • Input active trigger to $\overline{\text{INT0}}$ pin[2]
Count stop condition	• When reloading is completed after count value at counting TZSC register was set to "00_{16}" • When TZS bit in TYZMR register is set to “0” (stop counting) • When TZOS bit in TYZOC register is set to “0” (stop one-shot)
Interrupt request generation timing	In half cycles of count source, after count value at counting TZSC register is set "00_{16}" (at the same time as the TZout output change) [Timer Z interrupt]
TZOUT pin function	Pulse output
$\overline{\text{INT0}}$ 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 TZPR register. Same applies to PREZ register.
Write to timer	Value written to TZPR register and PREZ register are written to reload register only[3]. Same applies to TZSC register.
Select function	• Output level latch select function Output level for one-shot pulse waveform is selected by TZOPL bit. • $\overline{\text{INT0}}$ pin one-shot trigger control function and polarity select function Trigger input from $\overline{\text{INT0}}$ pin can be set to active or inactive by INOSTG bit. Also, an active trigger's polarity can be selected by INOSEG bit.

Notes:

1. The TZS bit in the TYZMR register must be set to "1" (start counting).
2. The TZS bit must be set to "1" (start counting), the INT0EN bit in the INTEN register to "1" (enabling $\overline{\text{INT0}}$ input), and the INOSTG bit in the PUM register to "1" (enabling $\overline{\text{INT0}}$ one-shot trigger)
3. The set values are reflected beginning with the next one-shot pulse after writing to the TZPR register.

Timer Y, Z mode register

b7	b6	b5	b4	b3	b2	b1	b0
	1	1	1				

Symbol	Address	After reset
TYZMR	0080_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TYMOD0	Timer Y-related bit		RW
R1EDG			RW
TYWC			RW
TYS			RW
TZMOD0	Timer Z operation mode bit	b5 b4 1 1 : Programmable wait one-shot generation mode	RW
TZMOD1			RW
TZWC	Timer Z write control bit	Must set to "1" in programmable wait one-shot generation mode[1]	RW
TZS	Timer Z count start flag	0 : Stops counting 1 : Starts counting	RW

Notes:
1. When TZS bit=1(starts counting), the timer Z count value is written to the reload register only.
When TZS bit=0(stops counting), the timer Z count value is written to both reload register and counter.

Timer Y, Z waveform output control register

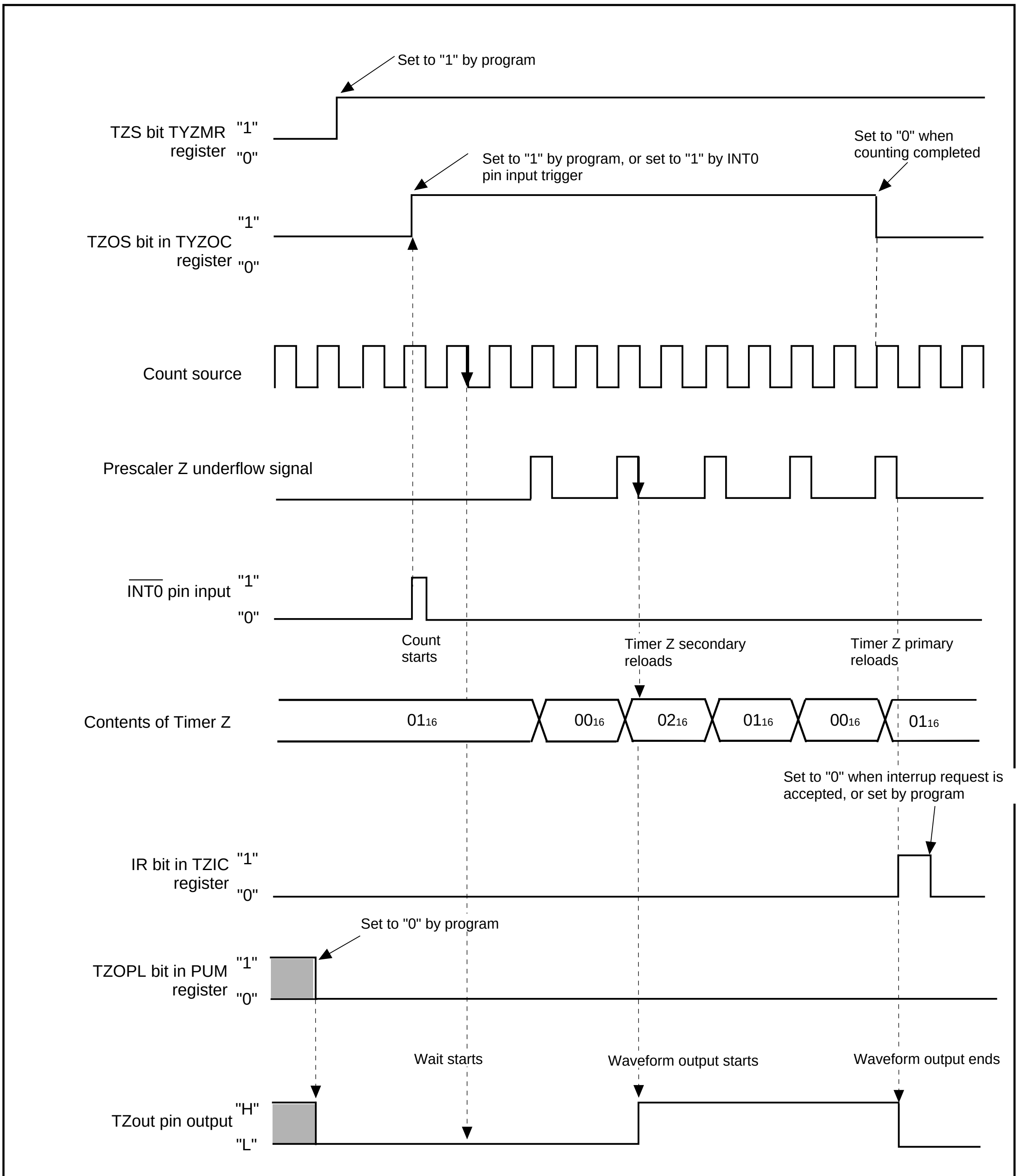
b7	b6	b5	b4	b3	b2	b1	b0
				0	0	0	0

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

Bit symbol	Bit name	Function	RW
(b3-b0)	Reserved bit	Must set to “0”	RW
TYOPL	Timer Y-related bit		RW
TZOPL	Timer Z output level latch	0 : Outputs "H" level one-shot pulse. Outputs "L" when the timer is stopped. 1 : Outputs "L" level one-shot pulse Outputs "H" when the timer is stopped.	RW
INOSTG	$\overline{\text{INT0}}$ pin one-shot trigger control bit[2]	0 : $\overline{\text{INT0}}$ pin one-shot trigger disabled 1 : $\overline{\text{INT0}}$ pin one-shot trigger enabled [2]	RW
INOSEG	$\overline{\text{INT0}}$ pin one-shot trigger polarity select bit[1]	0 : Edge trigger at falling edge 1 : Edge trigger at rising edge	RW

Notes:
1. The INOSEG bit is valid only when the INT0PL bit in the INTEN register is set to "0" (one-edge).
2. The INOSGT bit must be set to “1” after the INT0EN bit in the INTEN register and the INOSEG bit in the PUM register are set.
When setting the INOSTG bit to "1" ($\overline{\text{INT0}}$ pin one-shot trigger enabled), the INT0F0 and INT0F1 bits in the INT0F register must be set.
The INOSTG bit must be set to “0” ($\overline{\text{INT0}}$ pin one-shot trigger disabled) after the TZS bit in the TYZMR register is set to “0” (count stop).

Figure 12.26 TYZMR Register and PUM Register in Programmable Wait One-shot Generation Mode



Conditions: PREZ=01₁₆, TZSC=02₁₆

PUM register TZOPL bit=0, INOSTG bit=1 (INT0 one-shot trigger enabled)
INOSEG bit= 1 (edge trigger at rising edge)

Figure 12.27 Operation Example in Programmable Wait One-shot Generation Mode

12.4 Timer C

Timer C is a 16-bit free-running timer. Figure 12.28 shows a block diagram of Timer C. The Timer C uses an edge input to TCIN pin or the fRING128 clock 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. Table 12.13 shows Timer C specifications. Figure 12.29 shows TC, TM0, TCC0, and TCC1 registers. Figure 12.30 shows an operation example of Timer C.

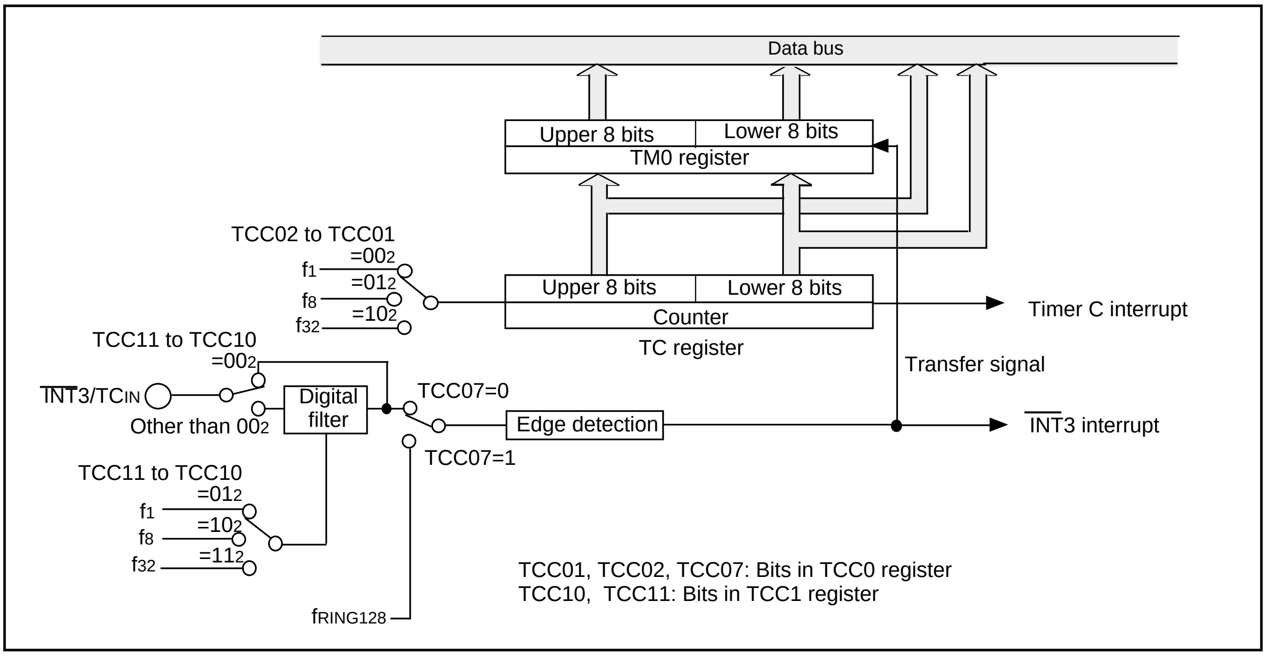


Figure 12.28 Timer C Block Diagram

Table 12.13 Timer C Specifications

Item	Specification
Count source	f1, f8, f32
Count operation	• Count up • Transfer value in TC register to TM0 register at active edge of measurement pulse • Value in TC register is set to "0000$_{16}$" when a counting stops
Count start condition	TCC00 bit in TCC0 register is set to "1" (capture enabled)
Counter stop condition	TCC00 bit in TCC0 register is set to "0" (capture disabled)
Interrupt request generation timing	• When active edge of measurement pulse is input [INT3 interrupt] • When Time C underflows [Timer C interrupt]
INT3/TCIN pin function	Programmable I/O or measurement pulse input
Counter value reset timing	When TCC00 bit in TCC0 register is set to "0" (capture disabled)
Read from timer[1]	• Counter value can be read out by reading TC register. • Counter value at measurement pulse active edge input can be read out by reading TM0 register.
Write to timer	Write to TC register and TM0 register is disabled
Select function	• INT3/TCIN switching function Measurement pulse active edge is selected by TCC03 to TCC04 bits • Digital filter function Digital filter sampling frequency is selected by TCC11 to TCC10 bits • Trigger select function TCIN input or fRING128 is selected by TCC07 bit.

Note: TC register and TM0 register must be read in 16-bit units.

Timer C register

(b15) b7	(b8) b0	b7	b0

Symbol TC — Address 0091_{16}-0090_{16} — After reset 0000_{16}

Function	RW
Internal count source is counted	RO

Capture register

(b15) b7	(b8) b0	b7	b0

Symbol TM0 — Address $009D_{16}$-$009C_{16}$ — After reset Indeterminate

Function	RW
When active edge of measurement pulse is input, the counter value of Timer C is stored	RO

Timer C control register 0

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol TCC0 — Address $009A_{16}$ — After reset 00_{16}

Bit symbol	Bit name	Function	RW
TCC00	Capture control bit	0 : Capture disabled 1 : Capture enabled	RW
TCC01	Timer C count source select bit[1]	b2 b1 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : Avoid this setting	RW
TCC02			RW
TCC03	$\overline{\text{INT3}}$ interrupt/capture input polarity select bit[1, 2]	b4 b3 0 0 : Rising edge 0 1 : Falling edge 1 0 : Both edges 1 1 : Avoid this setting	RW
TCC04			RW
(b6-b5)	reserved bit	Must set to "0"	RW
TCC07	$\overline{\text{INT3}}$ interrupt/capture input switching bit[1, 2]	0 : $\overline{\text{INT3}}$ 1 : $f_{RING128}$	RW

Notes:
1. Change this bit when TCC00 bit is set to “0” (count stop).
2. The IR bit in the INT3IC may be set to “1” (interrupt requested) when the TCC03, TCC04, or TCC07 bit is rewritten. Refer to the paragraph 19.2.5 “Changing Interrupt Source” in the Usage Notes Reference Book.

Timer C control register 1

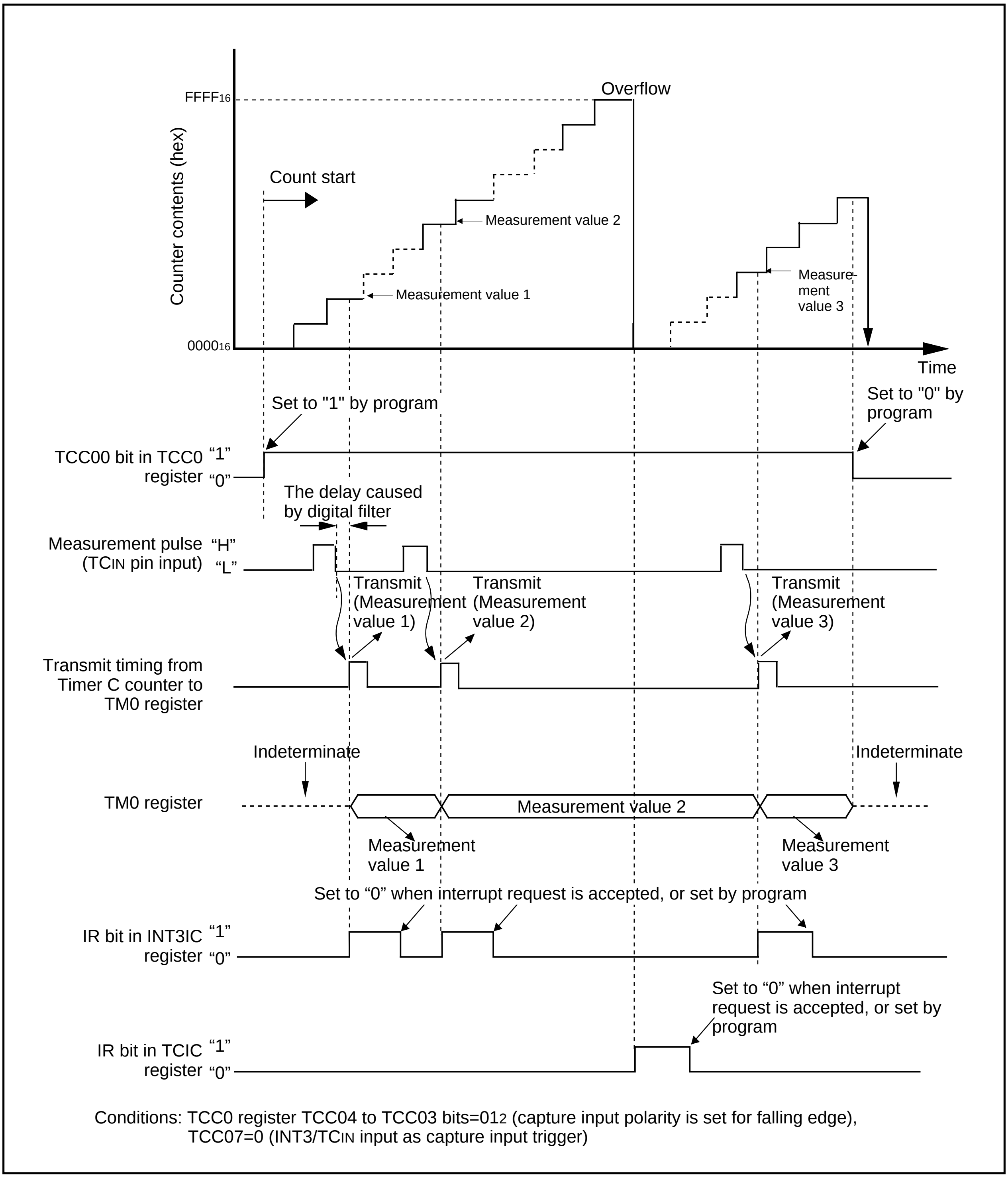
b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0		

Symbol TCC1 — Address $009B_{16}$ — After reset 00_{16}

Bit symbol	Bit name	Function	RW
TCC10	$\overline{\text{INT3}}$ input filter select bit[1]	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
TCC11			RW
(b7-b2)	Reserved bit	Must set to "0"	RW

Notes:
1. Input is recognized only when the same value from $\overline{\text{INT3}}$ pin is sampled three times in succession.

Figure 12.29 TC Register, TM0 Register, TCC0 Register, and TCC1 Register



Conditions: TCC0 register TCC04 to TCC03 bits=01$_2$ (capture input polarity is set for falling edge),
TCC07=0 (INT3/TCIN input as capture input trigger)

Figure 12.30 Operation Example of Timer C

13. Serial I/O

Serial I/O is configured with two channels: UART0 to 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 a block diagram of UARTi (i=0, 1). Figure 13.2 shows a block diagram of the UARTi transmit/receive.
UART0 has two modes: clock synchronous serial I/O mode, and clock asynchronous serial I/O mode (UART mode).
UART1 has only one mode, clock asynchronous serial I/O mode (UART mode).
Figures 13.3 to 13.5 show the UARTi-related registers.

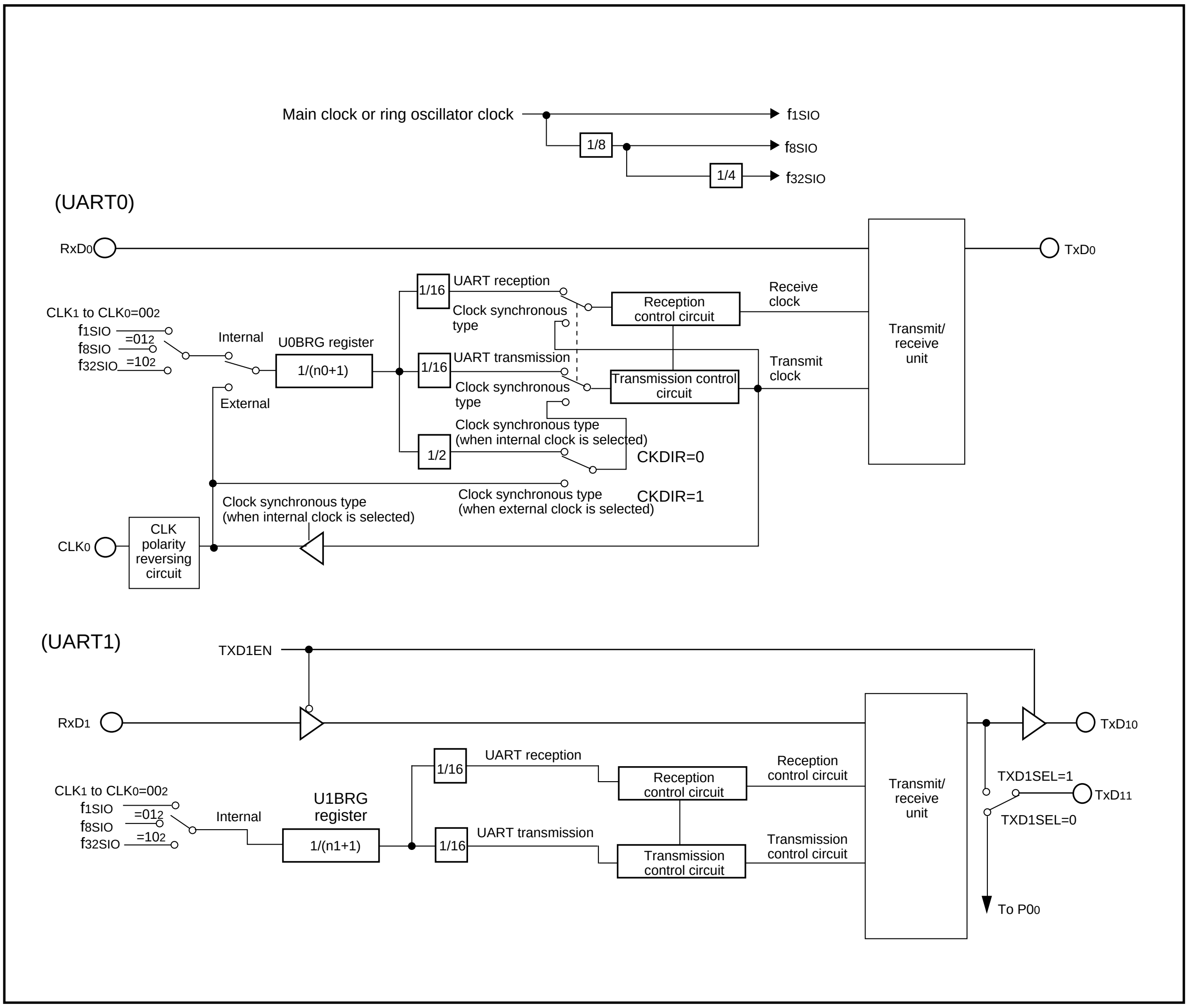


Figure 13.1 UARTi (i=0, 1) Block Diagram

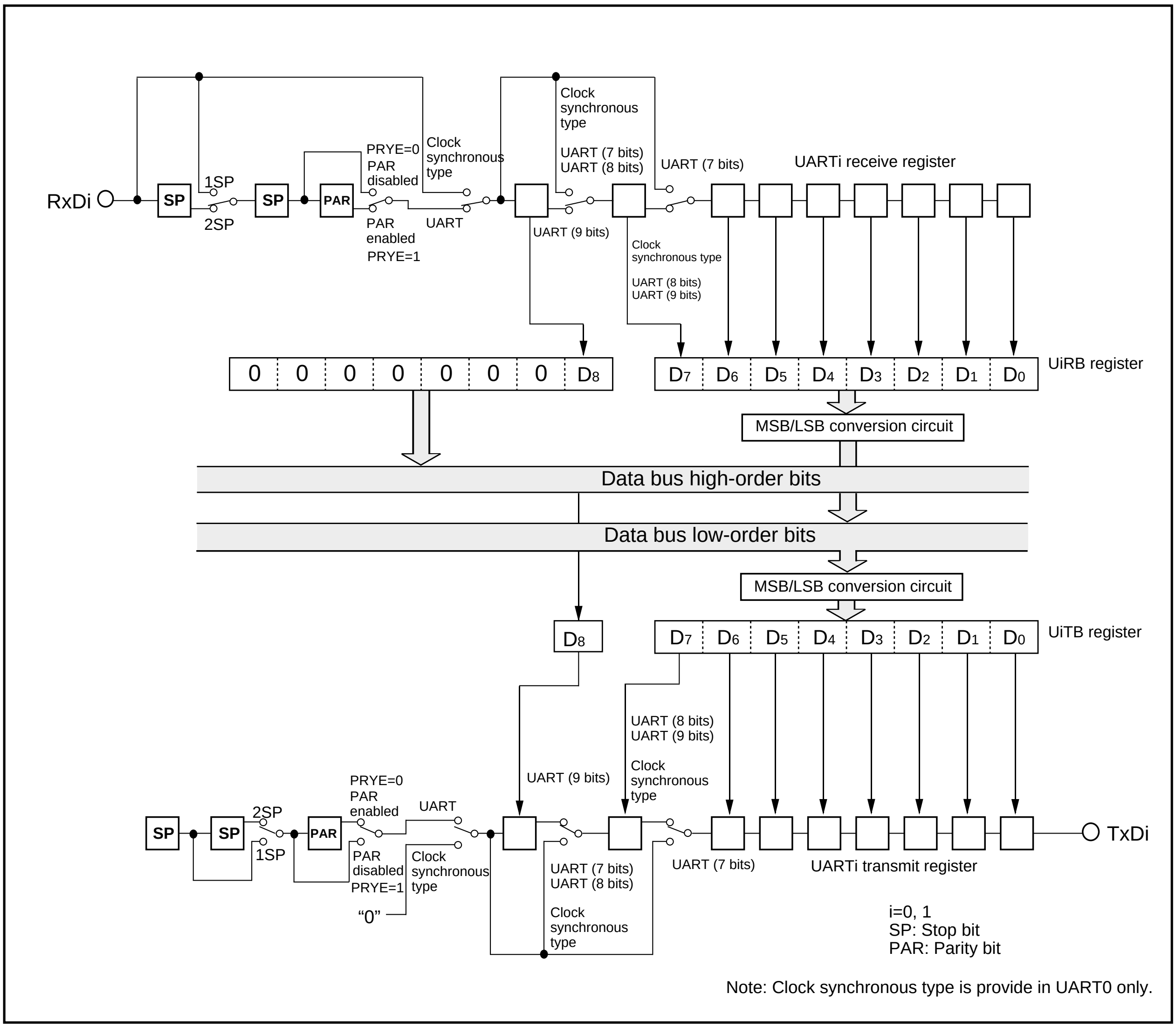
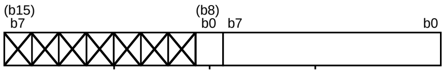


Figure 13.2 UARTi Transmit/Receive Unit

UARTi transmit buffer register[1, 2] (i=0, 1)

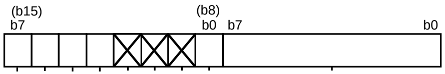


Symbol	Address	After reset
U0TB	$00A3_{16}$-$00A2_{16}$	Indeterminate
U1TB	$00AB_{16}$-$00AA_{16}$	Indeterminate

Function	RW
Transmit data	WO
Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.	—

Notes:
1. When transfer data length is 9-bit long, write high-byte first then low-byte.
2. Use MOV instruction to write to this register.

UARTi receive buffer register[1] (i=0, 1)

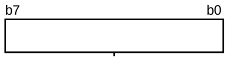


Symbol	Address	After reset
U0RB	$00A7_{16}$-$00A6_{16}$	Indeterminate
U1RB	$00AF_{16}$-$00AE_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
— (b7-b0)	——	Receive data (D_7 to D_0)	RO
— (b8)	——	Receive data (D_8)	RO
— (b11-b9)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
OER	Overrun error flag[2]	0 : No overrun error 1 : Overrun error found	RO
FER	Framing error flag[2]	0 : No framing error 1 : Framing error found	RO
PER	Parity error flag[2]	0 : No parity error 1 : Parity error found	RO
SUM	Error sum flag[2]	0 : No error 1 : Error found	RO

Notes:
1. Be sure to read UiRB register in 16-bit unit.
2. All of the SUM, PER, FER and OER bits are set to “0” (no error) when the SMD2 to SMD0 bits in the UiMR register are set to “000_2” (serial I/O disabled) or the RE bit in the UiC1 register is set to “0” (reception disabled). The SUM bit is set to “0” (no error) when all of the PER, FER and OER bits are set to “0” (no error).
The PER and FER bits are set to “0” even when the higher byte of the UiRB register is read.

UARTi baud rate generation register[1, 2] (i=0, 1)



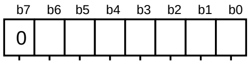
Symbol	Address	After reset
U0BRG	$00A1_{16}$	Indeterminate
U1BRG	$00A9_{16}$	Indeterminate

Function	Setting range	RW
Assuming that set value = n, UiBRG divides the count source by n + 1	00_{16} to FF_{16}	WO

Notes:
1. Write to this register while serial I/O is neither transmitting nor receiving.
2. Use MOV instruction to write to this register.

Figure 13.3 U0TB and U1TB Registers, U0RB and U1RB Registers, and U0BRG and U1BRG Registers

UARTi transmit/receive mode register (i=0, 1)

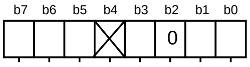


Symbol	Address	After reset
U0MR	$00A0_{16}$	00_{16}
U1MR	$00A8_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
SMD0	Serial I/O mode select bit[2]	b2 b1 b0 0 0 0 : Serial I/O disabled 0 0 1 : Clock synchronous serial I/O mode 1 0 0 : UART mode transfer data 7 bits long 1 0 1 : UART mode transfer data 8 bits long 1 1 0 : UART mode transfer data 9 bits long Must not be set except above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit[3]	0 : Internal clock 1 : External clock[1]	RW
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	RW
PRY	Odd/even parity select bit	Effective when PRYE = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
—— (b7)	Reserved bit	Must set to “0”	RW

Notes:
1. Must set the P1_6 bit in the PD1 register to “0” (input).
2. For the U1MR register, the SMD2 to SMD0 bits must not be set except the followings: “000_2”, “100_2”, “101_2”, or “110_2”.
3. Must set the CKDIR bit to “0” (internal clock) in UART1.

UARTi transmit/receive control register 0 (i=0, 1)

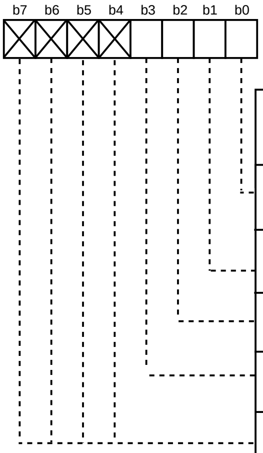


Symbol	Address	After reset
U0C0	$00A4_{16}$	08_{16}
U1C0	$00AC_{16}$	08_{16}

Bit symbol	Bit name	Function	RW
CLK0	BRG count source select bit	b1 b0 0 0 : f_{1SIO} is selected 0 1 : f_{8SIO} is selected 1 0 : f_{32SIO} is selected 1 1 : Avoid this setting	RW
CLK1			RW
—— (b2)	Reserved bit	Must 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)	RO
—— (b4)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		——
NCH	Data output select bit	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	RW
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	RW

Figure 13.4 U0MR and U1MR Registers and U0C0 and U1C0 Registers

UARTi transmit/receive control register 1 (i=0, 1)

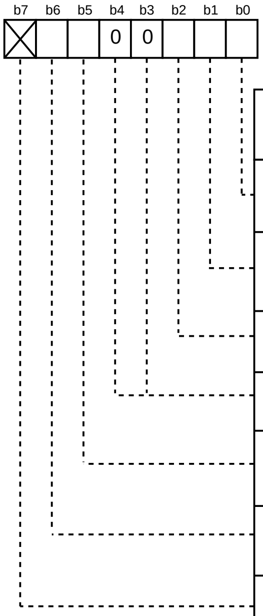


Symbol	Address	After reset
U0C1	00A5$_{16}$	02$_{16}$
U1C1	00AD$_{16}$	02$_{16}$

Bit symbol	Bit name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in UiTB register 1 : No data present in UiTB register	RO
RE	Receive enable bit[1]	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag[2]	0 : No data present in UiRB register 1 : Data present in UiRB register	RO
— (b7-b4)	Nothing is assigned. When write, set “0”. When read, its content is “0”.		—

Notes:
1. As for the UART1, set the TXD1EN bit in the UCON register before setting this bit to reception enabled.
2. The RI bit is set to "0" when the higher byte of the UiRB register is read.

UART transmit/receive control register 2



Symbol	Address	After reset
UCON	00B0$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
U0IRS	UART0 transmit interrupt cause select bit	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)	RW
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	RW
— (b4-b3)	Reserved bit	Must set to “0”	RW
TXD1SEL	Port TxD$_{11}$ switching bit	0 : I/O port P0$_0$ 1 : TxD$_{11}$	RW
TXD1EN	TxD$_{10}$/RxD$_1$ select bit	0 : RxD$_1$ 1 : TxD$_{10}$	RW
— (b7)	Nothing is assigned. When write, set “0”. When read, its content is “0”.		—

Notes
1. For P3$_7$, select “0” (RxD$_1$) for data receive, and “1” (TxD$_{10}$) for data transfer.
Set the PD3_7 bit in the PD3 register to “0” (input mode) when receiving.

Figure 13.5 U0C1 and U1C1 Registers and UCON Register

13.1 Clock Synchronous Serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. This mode can be selected with UART0. Table 13.1 lists the specifications of the clock synchronous serial I/O mode. Table 13.2 lists the registers used in clock synchronous serial I/O mode and the register values set.

Table 13.1 Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• CKDIR bit in U0MR register is set to “0” (internal clock): fi/ 2(n+1) fi=f1SIO, f8SIO, f32SIO n=setting value in UiBRG register: 00_{16} to FF_{16} • CKDIR bit is set to “1” (external clock): input from CLK0 pin
Transmission start condition	• Before transmission can start, the following requirements must be met[1] – TE bit in U0C1 register is set to “1” (transmission enabled) – TI bit in U0C1 register is set to “0” (data present in U0TB register)
Reception start condition	• Before reception can start, the following requirements must be met[1] – RE bit in U0C1 register is set to “1” (reception enabled) – TE bit in U0C1 register is set to “1” (transmission enabled) – TI bit in U0C1 register is set to “0” (data present in the U0TB register)
Interrupt request generation timing	• For transmission, one of the following conditions can be selected – U0IRS bit is set to “0” (transmit buffer empty): when transferring data from U0TB register to UART0 transmit register (at start of transmission) – U0IRS bit is set to “1” (transfer completed): when serial I/O finished sending data from UARTi transmit register • For reception When transferring data from the UART0 receive register to the U0RB register (at completion of reception)
Error detection	• Overrun error[2] This error occurs if serial I/O started receiving the next data before reading the U0RB register and received the 7th bit of the next data
Select function	• CLK polarity selection Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock • LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Continuous receive mode selection Reception is enabled immediately by reading the U0RB register

Notes:

1. When an external clock is selected, the conditions must be met while if the U0C0 register 0 CKPOL bit = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the U0C0 register is set to “1” (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. If an overrun error occurs, the value of U0RB register will be indeterminate. The IR bit of S0RIC register does not change.

Table 13. 2 Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode

Register	Bit	Function
U0TB	0 to 7	Set transmission data
U0RB	0 to 7	Reception data can be read
	OER	Overrun error flag
U0BRG	0 to 7	Set a transfer rate
U0MR	SMD2 to SMD0	Set to “001_2”
	CKDIR	Select the internal clock or external clock
U0C0	CLK1 to CLK0	Select the count source for the U0BRG register
	TXEPT	Transmit register empty flag
	NCH	Select TxD0 pin output mode
	CKPOL	Select the transfer clock polarity
	UFORM	Select the LSB first or MSB first
U0C1	TE	Set this bit to “1” to enable transmission/reception
	TI	Transmit buffer empty flag
	RE	Set this bit to “1” to enable reception
	RI	Reception complete flag
UCON	U0IRS	Select the source of UART0 transmit interrupt
	U0RRM	Set this bit to “1” to use continuous receive mode
	TXDISEL	Set to “0”
	TXDIEN	Set to “0”

Notes:

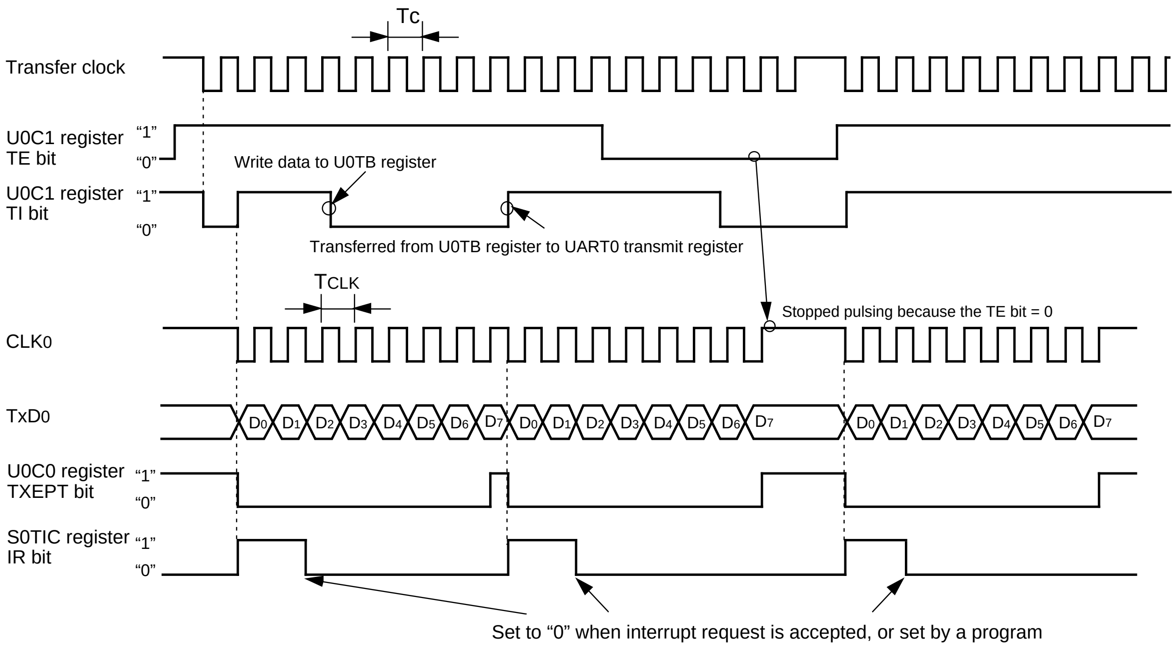
1. Not all register bits are described above. Set those bits to “0” when writing to the registers in clock synchronous serial I/O mode.

Table 13.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Note that for a period from when the UART0 operation mode is selected to when transfer starts, the TxD0 pin outputs an “H”. (If the Nch bit is set to “1”, this pin is in a high-impedance state.)

Table 13.3 Pin Functions

Pin name	Function	Method of selection
TxD0 (P14)	Serial data output	(Outputs dummy data when performing reception only)
RxD0 (P15)	Serial data input	PD1 register PD1_5 bit=0 (P15 can be used as an input port when performing transmission only)
CLK0 (P16)	Transfer clock output	U0MR register CKDIR bit=0
	Transfer clock input	U0MR register CKDIR bit=1 PD1 register PD1_6 bit=0

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

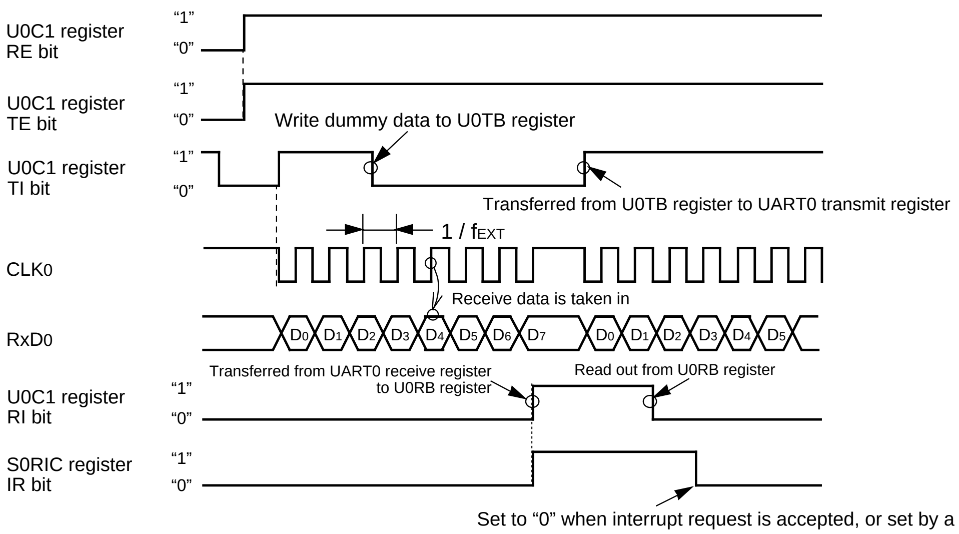

Tc
Transfer clock
U0C1 register TE bit
"1"
"0"
Write data to U0TB register
U0C1 register TI bit
"1"
"0"
Transferred from U0TB register to UART0 transmit register
TCLK
Stopped pulsing because the TE bit = 0
CLK0
TxD0
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7
U0C0 register TXEPT bit
"1"
"0"
S0TIC register IR bit
"1"
"0"
Set to "0" when interrupt request is accepted, or set by a program

Tc = TCLK = 2(n + 1) / fi
fi: frequency of U0BRG count source (f1SIO, f8SIO, f32SIO)
n: value set to U0BRG register

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

- U0MR register CKDIR bit = 0 (internal clock)
- U0C0 register CKPOL bit = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- U0IRS bit = 0 (an interrupt request occurs when the transmit buffer becomes empty):

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


U0C1 register RE bit
"1"
"0"
U0C1 register TE bit
"1"
"0"
Write dummy data to U0TB register
U0C1 register TI bit
"1"
"0"
Transferred from U0TB register to UART0 transmit register
1 / fEXT
CLK0
Receive data is taken in
RxD0
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5
Transferred from UART0 receive register to U0RB register
Read out from U0RB register
U0C1 register RI bit
"1"
"0"
S0RIC register IR bit
"1"
"0"
Set to "0" when interrupt request is accepted, or set by a program

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

- U0MR register CKDIR bit = 1 (external clock)
- U0C0 register CKPOL bit = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

Make sure the following conditions are met when input to the CLK0 pin before receiving data is high:

- U0C1 register TE bit = 1 (transmit enabled)
- U0C1 register RE bit = 1 (receive enabled)
- Write dummy data to the U0TB register

fEXT: frequency of external clock

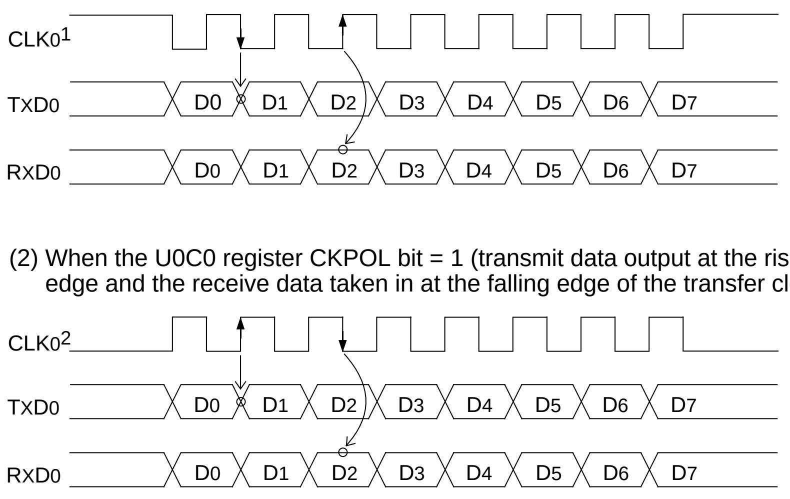
Figure 13.6 Transmit and Receive Operation

13.1.1 Polarity Select Function

Figure 13.7 shows the polarity of the transfer clock. Use the CKPOL bit in the U0C0 register to select the transfer clock polarity.

(1) When the U0C0 register CKPOL bit = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock)

(2) When the U0C0 register CKPOL bit = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock)



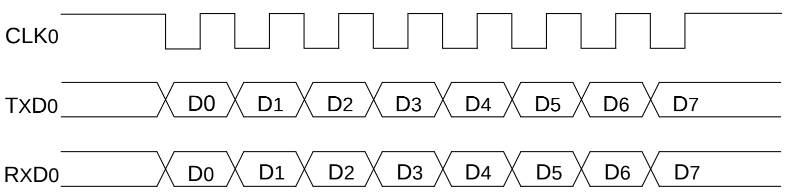
Notes:
1. When not transferring, the CLK0 pin outputs a high signal.
2. When not transferring, the CLK0 pin outputs a low signal.

Figure 13.7 Transfer Clock Polarity

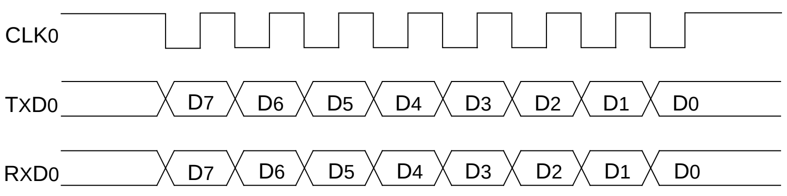
13.1.2 LSB First/MSB First Select Function

Figure 13.8 shows the transfer format. Use the UFORM bit in the U0C0 register to select the transfer format.

(1) When U0C0 register UFORM bit = 0 (LSB first)



(2) When U0C0 register UFORM bit = 1 (MSB first)



Note: This applies to the case where the CKPOL bit in the U0C0 register is set to “0” (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock).

Figure 13.8 Transfer Format

13.1.3 Continuous Receive Mode

The unit is configured to continuous receive mode by setting the U0RRM bit in the UCON register to “1” (enabling continuous receive mode). In this mode, reading the U0RB register enables data reception without resetting dummy data to the U0TB register. When the U0RRM bit is set to “1”, avoid writing dummy data to U0TB register in a program.

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. Tables 13.4 lists the specifications of the UART mode. Table 13.5 lists the registers and settings for UART mode.

Table 13.4 UART Mode Specifications

Item	Specification
Transfer data format	• Character bit (transfer data): selectable from 7, 8 or 9 bits • Start bit: 1 bit • Parity bit: selectable from odd, even, or none • Stop bit: selectable from 1 or 2 bits
Transfer clock	• UiMR(i=0, 1) register CKDIR bit = 0 (internal clock) : fj/ 16(n+1) fj=f_{1SIO}, f_{8SIO}, f_{32SIO} n=setting value in UiBRG register: 00_{16} to FF_{16} • CKDIR bit = “1” (external clock) : f_{EXT}/16(n+1) f_{EXT}: input from CLKi pin n=setting value in UiBRG register: 00_{16} to FF_{16}
Transmission start condition	• Before transmission can start, the following requirements must be met – TE bit in UiC1 register= 1 (transmission enabled) – TI bit in UiC1 register = 0 (data present in UiTB register)
Reception start condition	• Before reception can start, the following requirements must be met – RE bit in UiC1 register= 1 (reception enabled) – Start bit detection
Interrupt request generation timing	• For transmission, one of the following conditions can be selected – UiIRS bit = 0 (transmit buffer empty): when transferring data from UiTB register to UARTi transmit register (at start of transmission) – UiIRS bit =1 (transfer completed): when serial I/O finished sending data from UARTi transmit register • For reception When transferring data from UARTi receive register to UiRB register (at completion of reception)
Error detection	• Overrun error[1] This error occurs if serial I/O started receiving the next data before reading UiRB register and received 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	• TxD_{10}, RxD_1 selection (UART) $P3_7$ pin can be used as RxD_1 pin or TxD_{10} pin in UART1. Select by a program. • TxD_{11} pin selection (UART1) $P0_0$ pin can be used as TxD_{11} pin in UART1 or port $P0_0$. Select by a program.

Notes:

1. If an overrun error occurs, the value of U0RB register will be indeterminate. The IR bit in the S0RIC register does not change.

Table 13.5 Registers to Be Used and Settings in UART Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data[1]
UiRB	0 to 8	Reception data can be read[1]
	OER,FER,PER,SUM	Error flag
UiBRG	---	Set a transfer rate
UiMR	SMD2 to SMD0	Set these bits to ‘100_2’ when transfer data is 7 bits long Set these bits to ‘101_2’ when transfer data is 8 bits long Set these bits to ‘110_2’ when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock[2]
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	TXEPT	Transmit register empty flag
	NCH	Select TxDi pin output mode
	UFORM	LSB first or MSB first can be selected when transfer data is 8 bits long. Set this bit to “0” when transfer data is 7 or 9 bits long.
UiC1	TE	Set this bit to “1” to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to “1” to enable reception
	RI	Reception complete flag
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM	Set to “0”
	TXD1SEL	Select output pin for UART1 transfer data
	TXD1EN	Select TxD_{10} or RxD_1 to be used

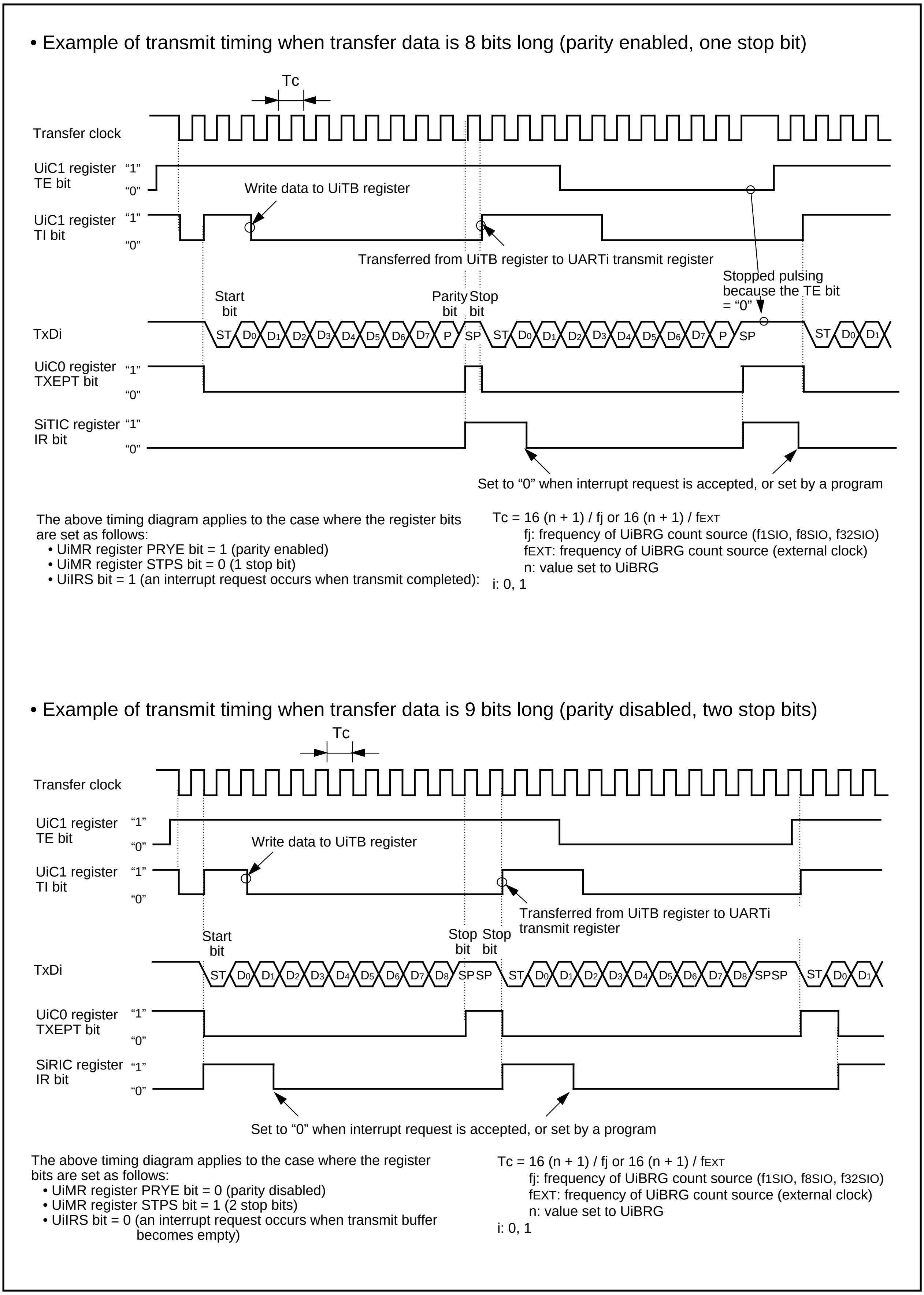
Notes:

1. The bits used for transmit/receive data are as follows: Bit 0 to bit 6 when transfer data is 7 bits long; bit 0 to bit 7 when transfer data is 8 bits long; bit 0 to bit 8 when transfer data is 9 bits long.
2. An external clock can be selected in UART0 only.

Table 13.6 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 an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 13.6 I/O Pin Functions in UART Mode

Pin name	Function	Method of selection
TxD_0 ($P1_4$)	Serial data output	(Cannot be used as a port when performing reception only)
RxD_0 ($P1_5$)	Serial data input	PD1 register PD1_5 bit=0 (Can be used as an input port when performing transmission only)
CLK_0 ($P1_6$)	Transfer clock output	U0MR register CKDIR bit=0
	Transfer clock input	U0MR register CKDIR bit=1 PD1 register PD1_6 bit=0
TxD_{10}/RxD_1 ($P3_7$)	Serial data output	TXD1EN=1
	Serial data input	TXD1EN=0, PD3 register PD3_7 bit=0
TxD_{11} ($P0_0$)	Serial data output	Serial data output, TXD1SEL=1



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

Tc
Transfer clock
UiC1 register TE bit
"1"
"0"
Write data to UiTB register
UiC1 register TI bit
"1"
"0"
Transferred from UiTB register to UARTi transmit register
Stopped pulsing because the TE bit = "0"
Start bit
Parity bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1
UiC0 register TXEPT bit
"1"
"0"
SiTIC register IR bit
"1"
"0"
Set to "0" when interrupt request is accepted, or set by a program

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

- UiMR register PRYE bit = 1 (parity enabled)
- UiMR register STPS bit = 0 (1 stop bit)
- UiIRS bit = 1 (an interrupt request occurs when transmit completed):

Tc = 16 (n + 1) / fj or 16 (n + 1) / fEXT
fj: frequency of UiBRG count source (f1SIO, f8SIO, f32SIO)
fEXT: frequency of UiBRG count source (external clock)
n: value set to UiBRG
i: 0, 1

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

Tc
Transfer clock
UiC1 register TE bit
"1"
"0"
Write data to UiTB register
UiC1 register TI bit
"1"
"0"
Transferred from UiTB register to UARTi transmit register
Start bit
Stop bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1
UiC0 register TXEPT bit
"1"
"0"
SiRIC register IR bit
"1"
"0"
Set to "0" when interrupt request is accepted, or set by a program

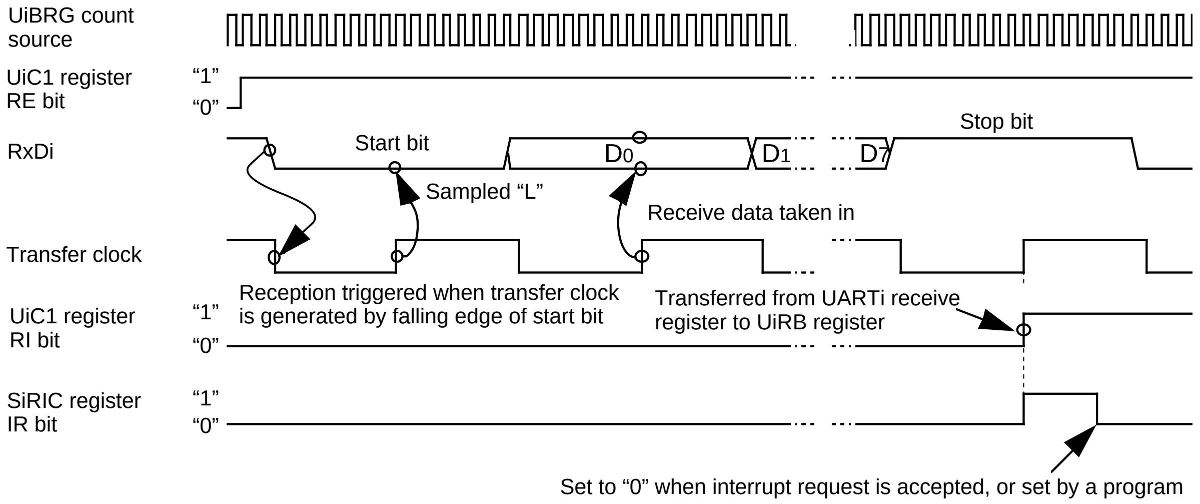
The above timing diagram applies to the case where the register bits are set as follows:

- UiMR register PRYE bit = 0 (parity disabled)
- UiMR register STPS bit = 1 (2 stop bits)
- UiIRS bit = 0 (an interrupt request occurs when transmit buffer becomes empty)

Tc = 16 (n + 1) / fj or 16 (n + 1) / fEXT
fj: frequency of UiBRG count source (f1SIO, f8SIO, f32SIO)
fEXT: frequency of UiBRG count source (external clock)
n: value set to UiBRG
i: 0, 1

Figure 13.9 Transmit Operation

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


UiBRG count source
UiC1 register RE bit
"1"
"0"
Stop bit
RxDi
Start bit
D_0
D_1
D_7
Sampled "L"
Receive data taken in
Transfer clock
Reception triggered when transfer clock is generated by falling edge of start bit
Transferred from UARTi receive register to UiRB register
UiC1 register RI bit
"1"
"0"
SiRIC register IR bit
"1"
"0"
Set to "0" when interrupt request is accepted, or set by a program

The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register PRYE bit = 0 (parity disabled)
- UiMR register STPS bit = 0 (1 stop bit)

i = 0, 1

Figure 13.10 Receive Operation

13.2.1 TxD_{10}/RxD_1 Select Function (UART1)

$P3_7$ can be used as TxD_{10} output pin or RxD_1 input pin by selecting with the TXD1EN bit in the UCON register. $P3_7$ is used as TxD_{10} output pin if the TXD1EN bit is set to "1" (TxD_{10}) and used as RxD_1 input pin if set to "0" (RxD_1).

13.2.2 TxD_{11} Select Function (UART1)

$P0_0$ can be used as TxD_{11} output pin or a port by selecting with the TXD1SEL bit in the UCON register. $P0_0$ is used as TxD_{11} output pin if the TXD1SEL bit is set to "1" (TxD_{11}) and used as an I/O port if set to "0" ($P0_0$).

14. A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. The analog inputs share the pins with P0$_0$ to P0$_7$. Therefore, when using these pins, make sure the corresponding port direction bits are set to “0” (input mode).

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

The result of A-D conversion is stored in the AD register.

Table 14.1 shows the performance of the A-D converter. Figure 14.1 shows a 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[1]	0V to Vref
Operating clock ϕ_{AD}[2]	AV$_{CC}$ = 5V f$_{AD}$, divide-by-2 of f$_{AD}$, divide-by-4 of f$_{AD}$ AV$_{CC}$ = 3V divide-by-2 of f$_{AD}$, divide-by-4 of f$_{AD}$
Resolution	8-bit or 10-bit (selectable)
Integral nonlinearity error	AV$_{CC}$ = Vref = 5V • 8-bit resolution ±2LSB • 10-bit resolution ±3LSB AVcc = Vref = 3.3V • 8-bit resolution ±2LSB • 10-bit resolution ±5LSB
Operating modes	One-shot mode and repeat mode[3]
Analog input pins	8 pins (AN$_0$ to AN$_7$)
A-D conversion start condition	ADST bit in ADCON0 register is set to “1” (A-D conversion starts)
Conversion speed per pin	• Without sample and hold function 8-bit resolution: 49 ϕ_{AD} cycles, 10-bit resolution: 59 ϕ_{AD} cycles • With sample and hold function 8-bit resolution: 28 ϕ_{AD} cycles, 10-bit resolution: 33 ϕ_{AD} cycles

Notes:

1. Does not depend on use of sample and hold function.
2. The frequency of ϕ_{AD} must be 10 MHz or less.
   When Vcc is less than 4.2V, ϕ_{AD} must be f$_{AD}$/2 or less by dividing f$_{AD}$.
   Without sample and hold function, the ϕ_{AD} frequency should be 250 kHz or more.
   With the sample and hold function, the ϕ_{AD} frequency should be 1 MHz or more.
3. In repeat mode, only 8-bit mode can be used.

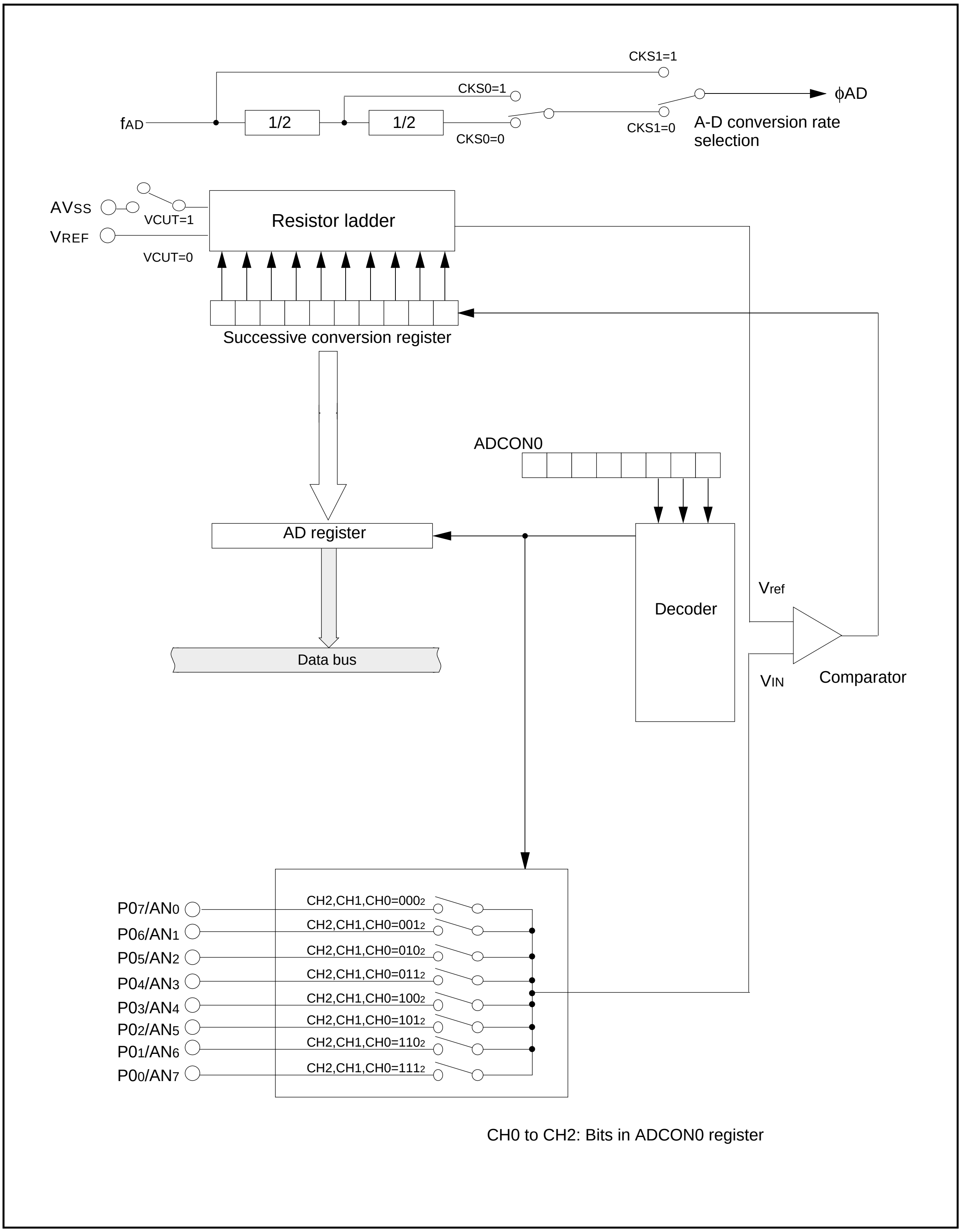


Figure 14.1 A-D Converter Block Diagram

A-D control register 0[1]

b7	b6	b5	b4	b3	b2	b1	b0
		0	0				

Symbol	Address	After reset
ADCON0	00D6$_{16}$	00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit[2]	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$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_7$ is selected	RW
CH1			RW
CH2			RW
MD0	A-D operation mode select bit 0[2]	0 : One-shot mode 1 : Repeat mode	RW
(b4)	Reserved bit	Must set to “0”	RW
(b5)	Reserved bit	Must 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[3]	0 : f_{AD}/4 is selected 1 : f_{AD}/2 is selected	RW

Notes:
1. If the ADCON register is rewritten during A-D conversion, the conversion result is indeterminate.
2. When changing A-D operation mode, set analog input pin again.
3. This bit is valid when the CKS1 bit in the ADCON1 register is set to “0”.

A-D control register 1[1]

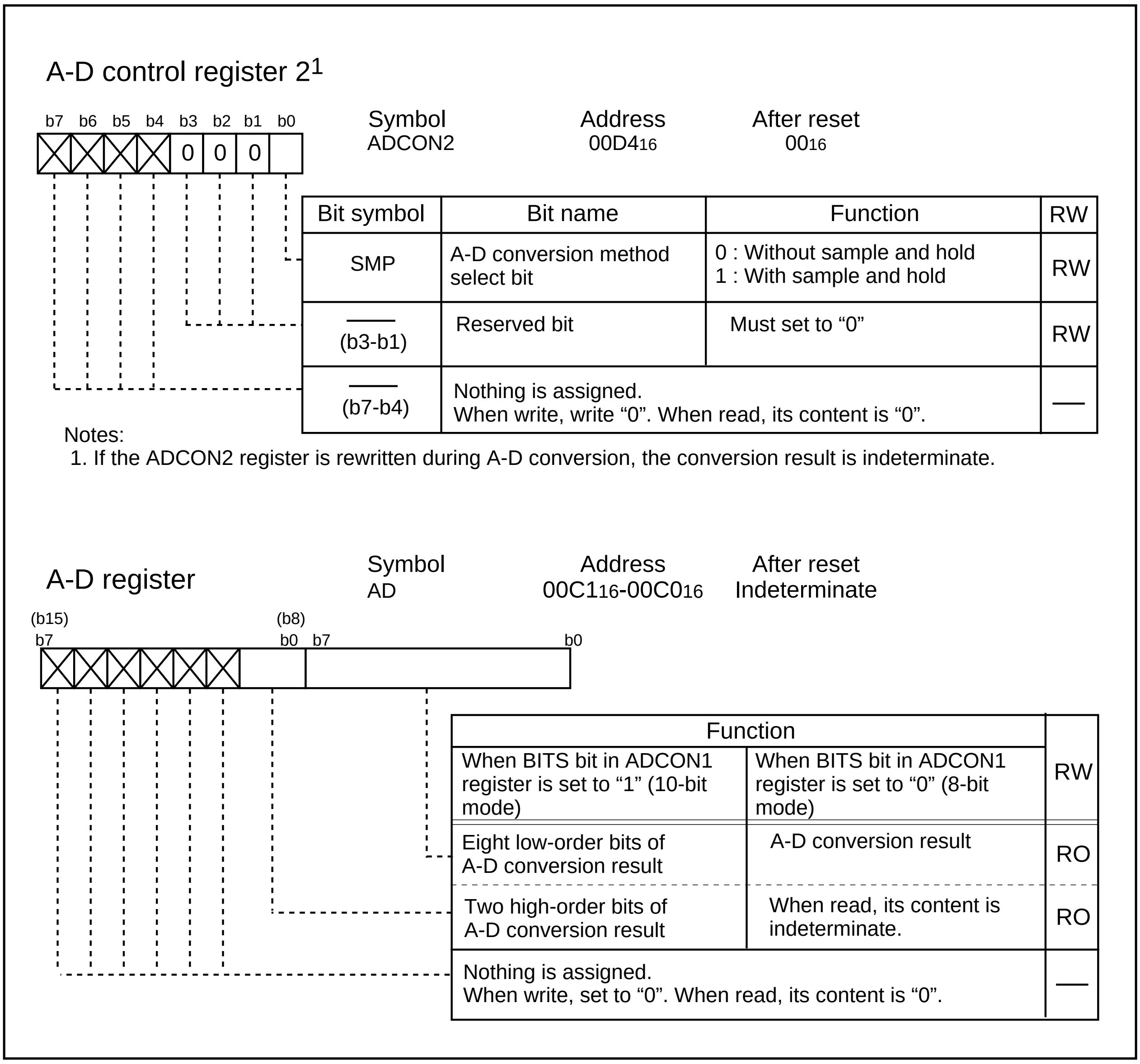
b7	b6	b5	b4	b3	b2	b1	b0
0	0				0	0	0

Symbol	Address	After reset
ADCON1	00D7$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
(b2-b0)	Reserved bit	Must set to “0”	RW
BITS	8/10-bit mode select bit[2]	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1[3]	0 : CKS0 bit in ADCON0 register is valid 1 : f_{AD} is selected	RW
VCUT	Vref connect bit[4]	0 : Vref not connected 1 : Vref connected	RW
(b6-b7)	Reserved bit	Must set to “0”	RW

Notes:
1. If the ADCON1 register is rewritten during A-D conversion, the conversion result is indeterminate.
2. In repeat mode, the BITS bit must be set to “0” (8-bit mode).
3. The ϕ_{AD} frequency must be 10 MHz or less.
4. If the VCUT bit is reset from “0” (Vref unconnected) to “1” (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 14.2 ADCON0 Register and ADCON1 Register



A-D control register 2[1]

Symbol	Address	After reset
ADCON2	00D4$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	RW
—— (b3-b1)	Reserved bit	Must set to “0”	RW
—— (b7-b4)	Nothing is assigned. When write, write “0”. When read, its content is “0”.		——

Notes:
1. If the ADCON2 register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D register

Symbol	Address	After reset
AD	00C1$_{16}$-00C0$_{16}$	Indeterminate

Function		RW
When BITS bit in ADCON1 register is set to “1” (10-bit mode)	When BITS bit in ADCON1 register is set to “0” (8-bit mode)	RW
Eight low-order bits of A-D conversion result	A-D conversion result	RO
Two high-order bits of A-D conversion result	When read, its content is indeterminate.	RO
Nothing is assigned. When write, set to “0”. When read, its content is “0”.		——

Figure 14.3 ADCON2 Register and AD Register

14.1 One-shot Mode

In one-shot mode, the input voltage on one selected pin is A-D converted once. Table 14.2 lists the specifications of one-shot mode. Figure 14.4 shows the ADCON0 and ADCON1 registers in one-shot mode.

Table 14.2 One-shot Mode Specifications

Item	Specification
Function	Input voltage on one pin selected by CH2 to CH0 bits is A-D converted once.
Start condition	Set ADST bit to “1”
Stop condition	• Completion of A-D conversion (ADST bit is set to “0”) • Set ADST bit to “0”
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN$_0$ to AN$_7$, as selected
Reading of result of A-D converter	Read AD register

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0			

Symbol ADCON0 Address 00D6$_{16}$ After reset 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit[2]	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$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_{71}$ is selected	RW
CH1			RW
CH2			RW
MD	A-D operation mode select bit[2]	0 : One-shot mode	RW
(b4)	Reserved bit	Must set to “0”	RW
(b5)	Reserved bit	Must 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[3]	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	RW

Notes:
1. If the ADCON0 register is rewritten during A-D conversion, the conversion result is indeterminate.
2. When changing A-D operation mode, set analog input pin again.
3. This bit is valid when the CKS1 bit in the ADCON1 register is set to “0”.

A-D control register 1[1]

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

Symbol ADCON1 Address 00D7$_{16}$ After reset 00$_{16}$

Bit symbol	Bit name	Function	RW
(b2-b0)	Reserved bit	Must 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[2]	0 : CKS0 bit in ADCON0 register is valid 1 : f$_{AD}$ is selected	RW
VCUT	Vref connect bit[3]	1 : Vref connected	RW
(b6-b7)	Reserved bit	Must set to “0”	RW

Notes:
1. If the ADCON1 register is rewritten during A-D conversion, the conversion result is indeterminate.
2. The ϕ_{AD} frequency must be 10 MHz or less.
3. If the VCUT bit is reset from “0” (Vref unconnected) to “1” (Vref connected), wait for 1 µs or more before starting A-D conversion.

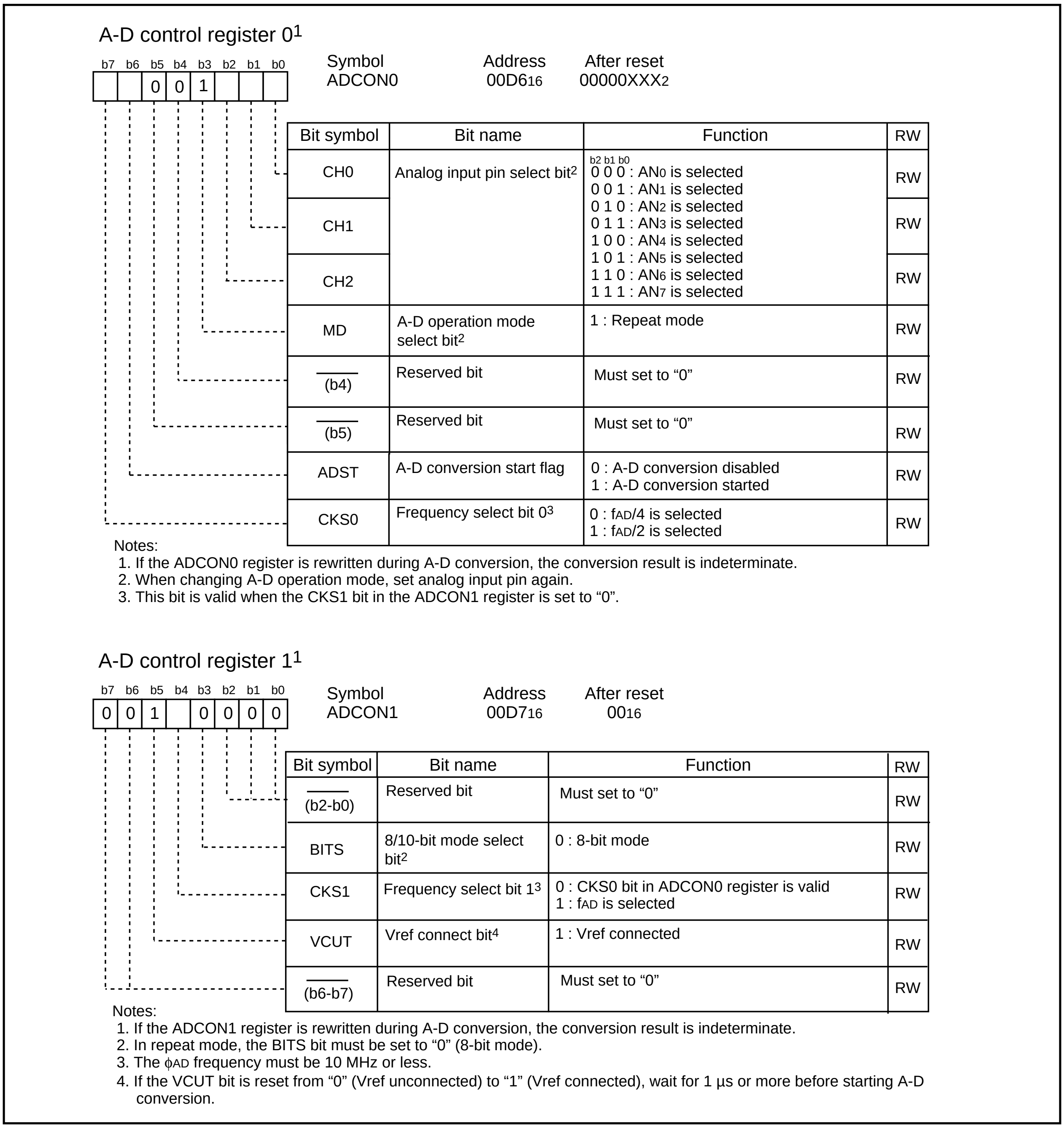
Figure 14.4 ADCON0 Register and ADCON1 Registers in One-shot Mode

14.2 Repeat Mode

In repeat mode, the input on one selected pin is A-D converted repeatedly. Table 14.3 lists the specifications of repeat mode. Figure 14.5 shows the ADCON0 and ADCON1 registers in repeat mode.

Table 14.3 Repeat Mode Specifications

Item	Specification
Function	Input voltage on one pin selected by CH2 to CH0 bits is A-D converted repeatedly
Start condition	Set ADST bit to “1”
Stop condition	Set ADST bit to “0”
Interrupt request generation timing	None generated
Input pin	One of AN_0 to AN_7, as selected
Reading of result of A-D converter	Read AD register



A-D control register 0[1]

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	1			

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

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit[2]	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 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected	RW
CH1			RW
CH2			RW
MD	A-D operation mode select bit[2]	1 : Repeat mode	RW
— (b4)	Reserved bit	Must set to “0”	RW
— (b5)	Reserved bit	Must 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[3]	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	RW

Notes:
1. If the ADCON0 register is rewritten during A-D conversion, the conversion result is indeterminate.
2. When changing A-D operation mode, set analog input pin again.
3. This bit is valid when the CKS1 bit in the ADCON1 register is set to “0”.

A-D control register 1[1]

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

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

Bit symbol	Bit name	Function	RW
— (b2-b0)	Reserved bit	Must set to “0”	RW
BITS	8/10-bit mode select bit[2]	0 : 8-bit mode	RW
CKS1	Frequency select bit 1[3]	0 : CKS0 bit in ADCON0 register is valid 1 : f_{AD} is selected	RW
VCUT	Vref connect bit[4]	1 : Vref connected	RW
— (b6-b7)	Reserved bit	Must set to “0”	RW

Notes:
1. If the ADCON1 register is rewritten during A-D conversion, the conversion result is indeterminate.
2. In repeat mode, the BITS bit must be set to “0” (8-bit mode).
3. The ϕ_{AD} frequency must be 10 MHz or less.
4. If the VCUT bit is reset from “0” (Vref unconnected) to “1” (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 14.5 ADCON0 Register and ADCON1 Register in Repeat Mode

14.3 Sample and Hold

If the SMP bit in the ADCON2 register is set to “1” (with sample-and-hold), the conversion speed per pin is increased to 28 ØAD cycles for 8-bit resolution or 33 ØAD cycles for 10-bit resolution. Sample-and-hold is effective in all operation modes. Select whether or not to use the sample-and-hold function before starting A-D conversion.

15. Programmable I/O Ports

15. 1 Description

The programmable input/output ports (hereafter referred to as “I/O ports”) consist of 22 lines P0, P1, P30 to P33, P37, and P45. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high every 4 lines. 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”.

P46 and P47 can be used as an input only port if the main clock oscillation circuit is not used.

Figures 15.1 to 15.4 show the I/O ports. Figure 15.5 shows the I/O pins.

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

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

15.1.1 Port Pi Direction Register (PDi Register, i = 0, 1, 3, 4)

Figure 15.6 shows the PDi register.

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

15.1.2 Port Pi Register (Pi Register, i = 0 to 4)

Figure 15.7 shows the Pi register.

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

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

15.1.3 Pull-up Control Register 0, Pull-up Control Register 1 (PUR0 and PUR1 Registers)

Figure 15.8 shows the PUR0 and PUR1 registers.

The PUR0 and PUR1 register bits can be used to select whether or not to pull the corresponding port high in 4 bit units. The port chosen to be pulled high has a pull-up resistor connected to it when the direction bit is set for input mode.

15.1.4 Port P1 Drive Capacity Control Register (DRR Register)

Figure 15.8 shows the DRR register.

The DRR register is used to control the drive capacity of the port P1 N-channel output transistor. The bits in this register correspond one for one to each port.

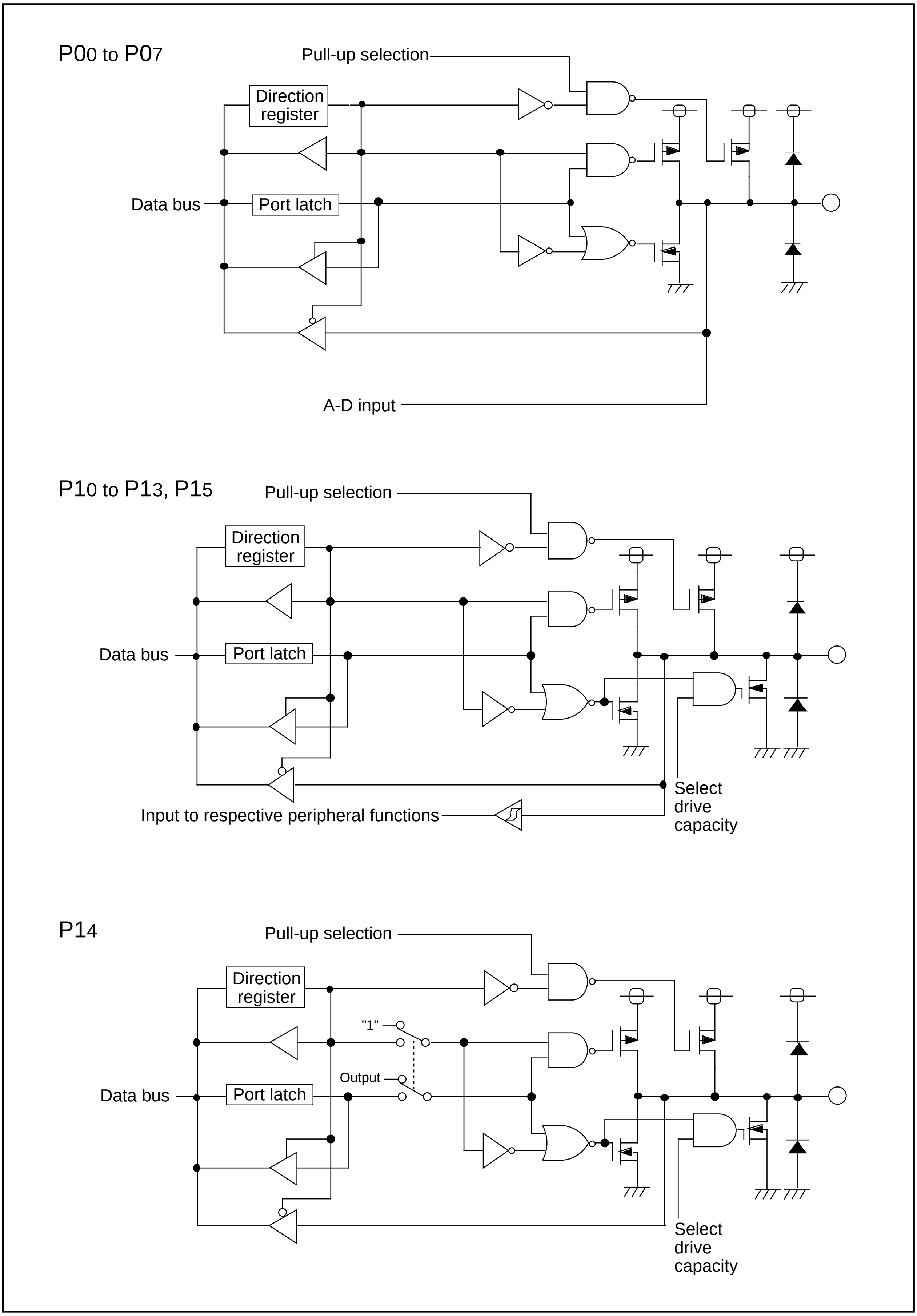


Figure 15.1 Programmable I/O Ports (1)

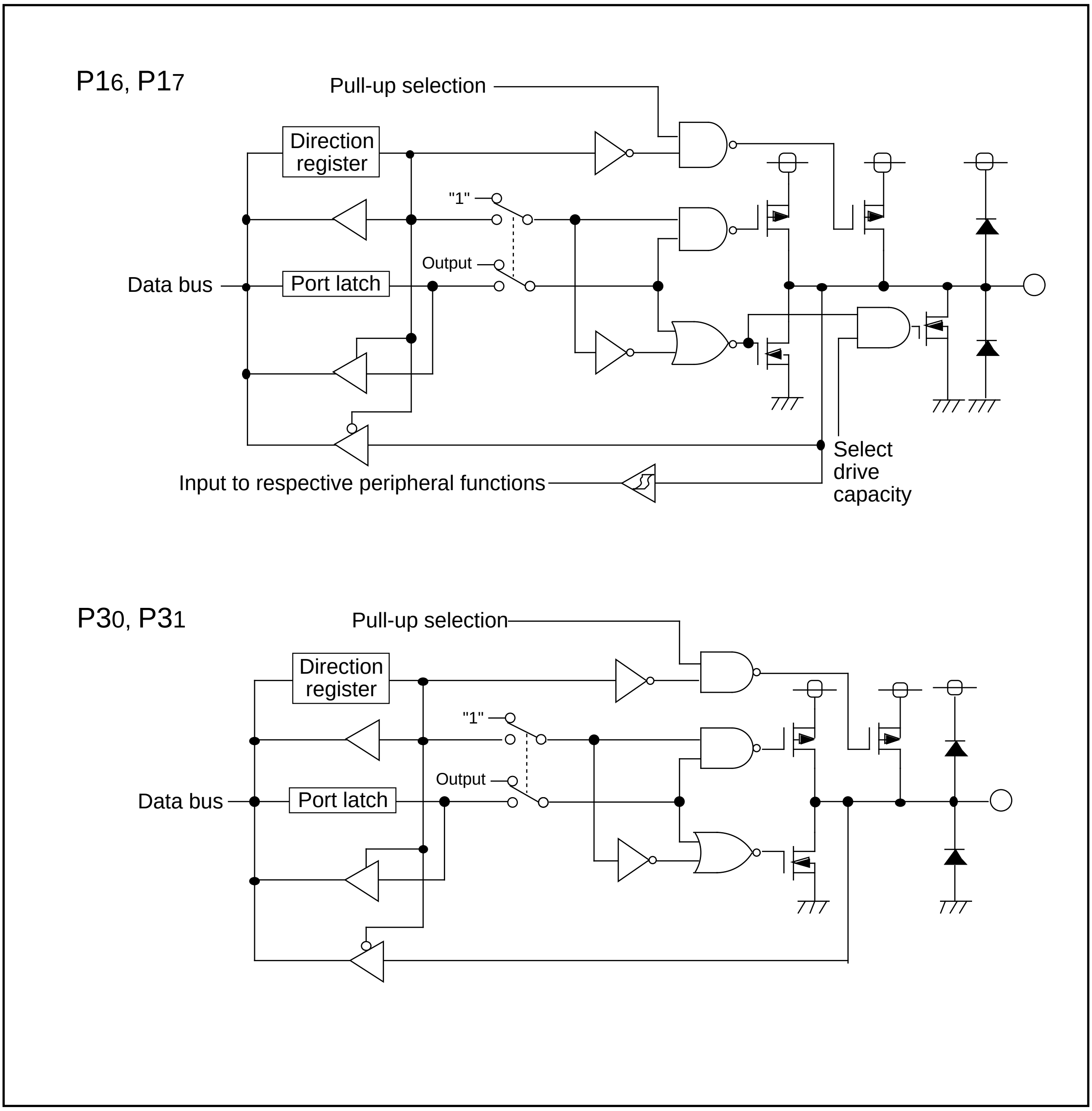


Figure 15.2 Programmable I/O Ports (2)

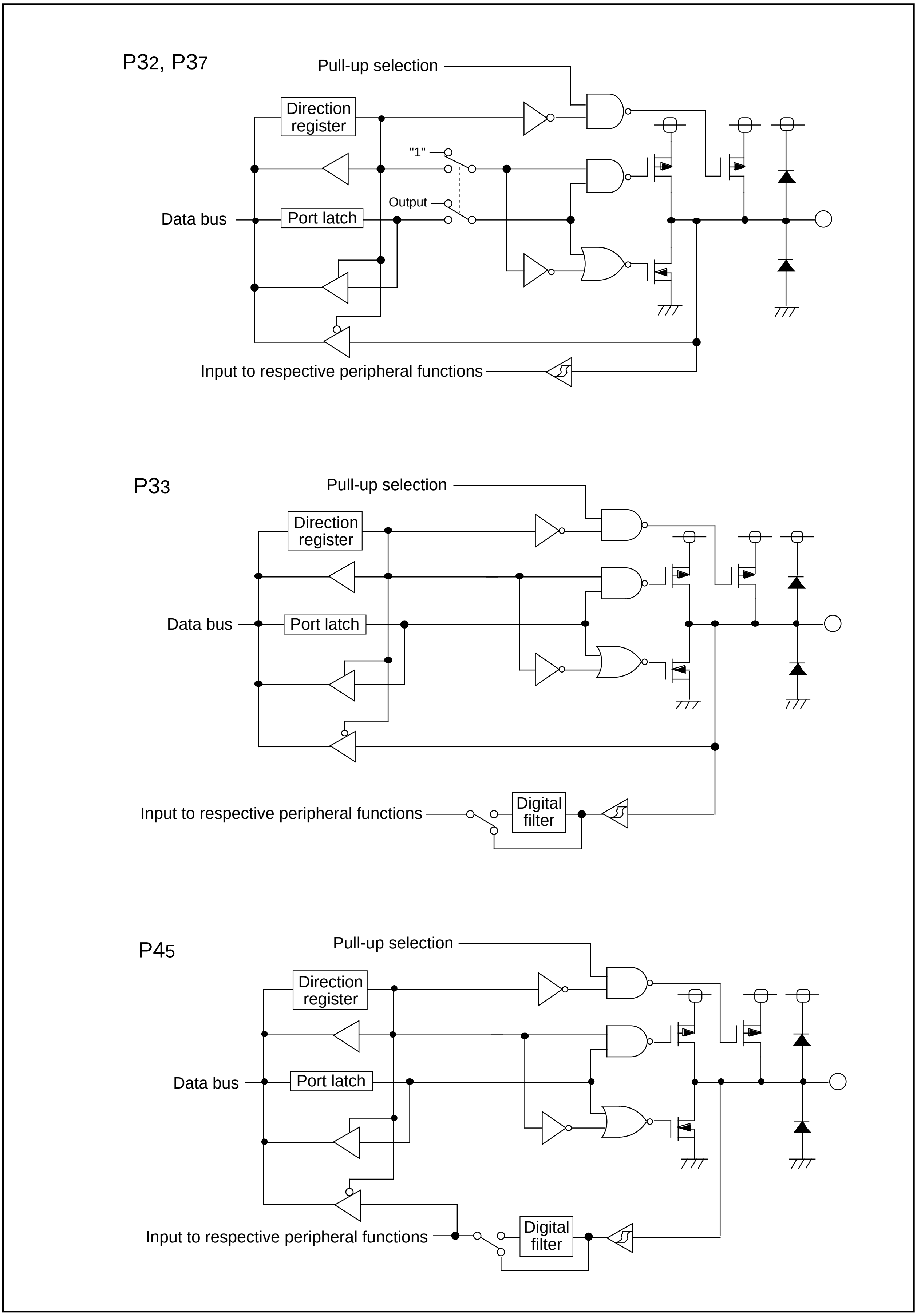
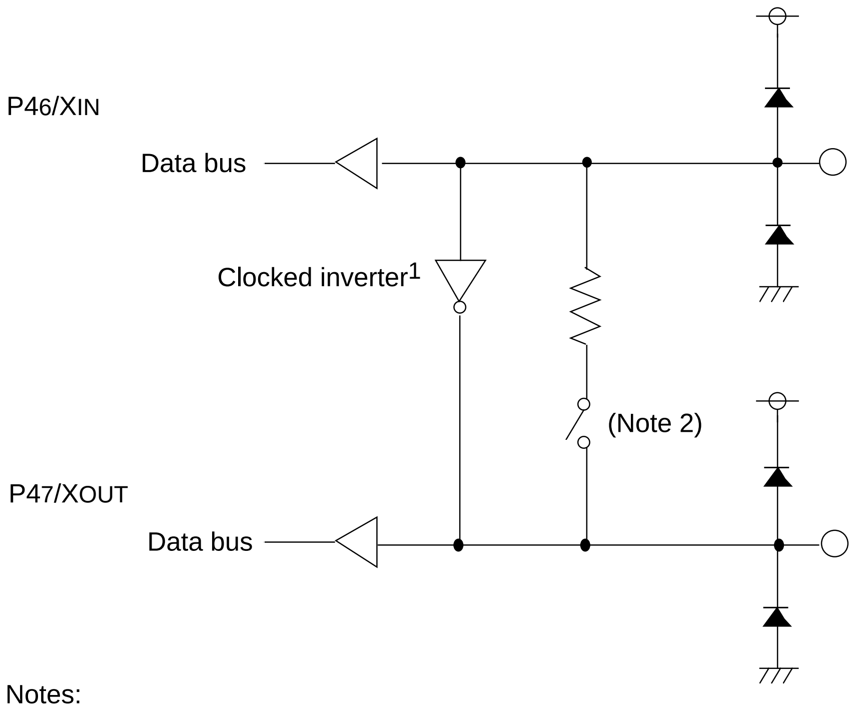
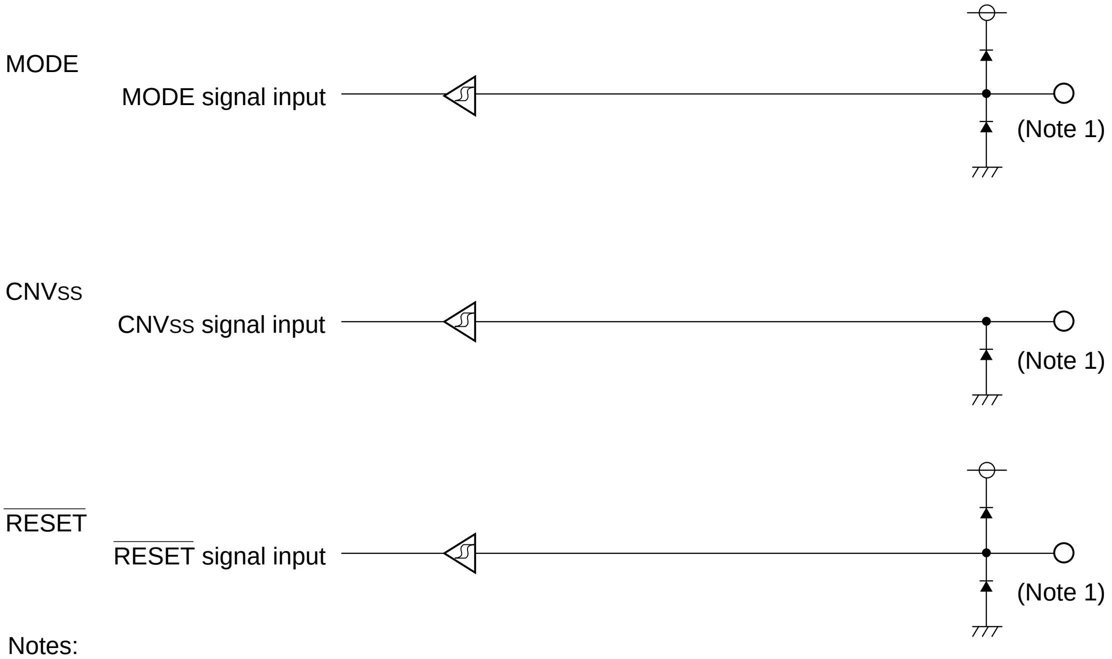


Figure 15.3 Programmable I/O Ports (3)



Notes:
1. When CM05=1, CM10=1, or CM13=0, the clocked inverter is cutoff.
2. When CM10 or CM13=0, the feedback resistor is unconnected.

Figure 15.4 Programmable I/O Port (4)

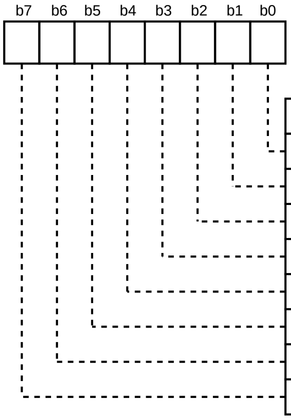


Notes:
1. ····◀···· symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 15.5 I/O Pins

Port Pi direction register (i=0, 1, 3, 4)[1, 2, 3]

Symbol	Address	After reset
PD0	00E2$_{16}$	00$_{16}$
PD1	00E3$_{16}$	00$_{16}$
PD3	00E7$_{16}$	00$_{16}$
PD4	00EA$_{16}$	00$_{16}$



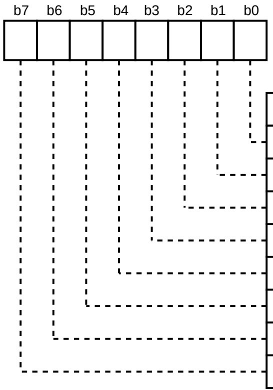
Bit symbol	Bit name	Function	RW
PDi_0	Port Pi$_0$ direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PDi_1	Port Pi$_1$ direction bit		RW
PDi_2	Port Pi$_2$ direction bit		RW
PDi_3	Port Pi$_3$ direction bit		RW
PDi_4	Port Pi$_4$ direction bit		RW
PDi_5	Port Pi$_5$ direction bit		RW
PDi_6	Port Pi$_6$ direction bit		RW
PDi_7	Port Pi$_7$ direction bit		RW

Notes:
1. The PD0 register must be written to by the next instruction after setting the PRC2 bit in the PRCR register to “1” (write enabled).
2. Nothing is assigned to the PD3_4 to PD3_6 bits in the PD3 register.
When writing to the PD3_4 to PD3_6 bits, write “0” (input mode). When read, its content is “0”.
3. Nothing is assigned to the PD4_0 to PD4_4, PD4_6 and PD4_7 bits in the PD4 register.
When writing to the PD4_0 to PD4_4, PD4_6 and PD4_7 bits, write “0” (input mode). When read, its content is “0”.

Figure 15.6 PD0 Register, PD1 Register, PD3 Register, and PD4 Register

Port Pi register (i=0, 1, 3, 4)[1, 2]

Symbol	Address	After reset
P0	00E0$_{16}$	Indeterminate
P1	00E1$_{16}$	Indeterminate
P3	00E5$_{16}$	Indeterminate
P4	00E8$_{16}$	Indeterminate



Bit symbol	Bit name	Function	RW
Pi_0	Port Pi$_0$ bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : “L” level 1 : “H” level (i = 0, 1, 3, 4)	RW
Pi_1	Port Pi$_1$ bit		RW
Pi_2	Port Pi$_2$ bit		RW
Pi_3	Port Pi$_3$ bit		RW
Pi_4	Port Pi$_4$ bit		RW
Pi_5	Port Pi$_5$ bit		RW
Pi_6	Port Pi$_6$ bit		RW
Pi_7	Port Pi$_7$ bit		RW

Notes:
1. Nothing is assigned to the P3_4 to P3_6 bits in the P3 register.
When writing to the P3_4 to P3_6 bits, write “0” (“L” level). When read, its content is “0”.
2. Nothing is assigned to the P4_0 to P4_4 bits in the P4 register.
When writing to the P4_0 to P4_4 bits, write “0” (“L” level). When read, its content is “0”.

Figure 15.7 P0 Register, P1 Register, P3 Register, and P4 Register

Pull-up control register 0

b7	b6	b5	b4	b3	b2	b1	b0
		☒	☒				

Symbol	Address	After reset
PUR0	00FC$_{16}$	00XX0000$_2$

Bit	Bit	Function	RW
PU00	P0$_0$ to P0$_3$ pull-up[1]	0 : Not pulled high 1 : Pulled high[1]	RW
PU01	P0$_4$ to P0$_7$ pull-up[1]		RW
PU02	P1$_0$ to P1$_3$ pull-up[1]		RW
PU03	P1$_4$ to P1$_7$ pull-up[1]		RW
——— (b5-b4)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
PU06	P3$_0$ to P3$_3$ pull-up[1]	0 : Not pulled high 1 : Pulled high[1]	RW
PU07	P3$_7$ pull-up[1]		RW

Notes:
1. The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.

Pull-up control register 1

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

Symbol	Address	After reset
PUR1	00FD$_{16}$	XXXXXX0X$_2$

Bit symbol	Bit name	Function	RW
——— (b0)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
PU11	P4$_5$ pull-up[1]	0 : Not pulled high 1 : Pulled high[1]	RW
——— (b7-b2)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—

Notes:
1. The P4$_5$ pin for which the PU11 bit is “1” (pulled high) and the PD4_5 bit is “0” (input mode) is pulled high.

Port P1 drive capacity control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
DRR	00FE$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
DRR0	P1$_0$ drive capacity	Set P1 N-channel output transistor drive capacity 0 : LOW 1 : HIGH	RW
DRR1	P1$_1$ drive capacity		RW
DRR2	P1$_2$ drive capacity		RW
DRR3	P1$_3$ drive capacity		RW
DRR4	P1$_4$ drive capacity		RW
DRR5	P1$_5$ drive capacity		RW
DRR6	P1$_6$ drive capacity		RW
DRR7	P1$_7$ drive capacity		RW

Figure 15.8 PUR0 Register, PUR1 Register, and DRR Register

15.2 Unassigned Pin Handling

Table 15.1 lists the handling of unassigned pins.

Table 15.1 Unassigned Pin Handling

Pin name	Connection
Ports P0, P1, P3$_0$ to P3$_3$, P4$_5$	After setting for input mode, connect every pin to V$_{SS}$ via a resistor(pull-down); or after setting for output mode, leave these pins open[1, 2].
Ports P4$_6$, P4$_7$	Connect to V$_{CC}$ via resistor (pull-up)[2]
AV$_{CC}$, V$_{REF}$	Connect to V$_{CC}$
AV$_{SS}$	Connect to V$_{SS}$

Notes:

1. When these ports are set for output mode and left open, they remain input mode until they are set for output mode by a program. The voltage level of these pins may be unstable and the power supply voltage may increase for the time the ports remain input mode.
   The content of the direction registers may change due to noise or runaway caused by noise. In order to enhance program reliability, set the direction registers periodically by a program.
2. Connect these unassigned pins to the microcomputer using the shortest wire length (within 2 cm) possible.

16. Electrical Characteristics

Table 16.1 Absolute Maximum Ratings

Symbol	Parameter	Condition	Rated value	Unit
V_{CC}	Supply voltage	V_{CC}=AV$_{CC}$	-0.3 to 6.5	V
AV$_{CC}$	Analog supply voltage	V_{CC}=AV$_{CC}$	-0.3 to 6.5	V
V_I	Input voltage		-0.3 to V_{CC}+0.3	V
V_O	Output voltage		-0.3 to V_{CC}+0.3	V
P_d	Power dissipation	Topr=25 °C	300	mW
T_{opr}	Operating ambient temperature		-20 to 85 / -40 to 85 (D version)	°C
T_{stg}	Storage temperature		-65 to 150	°C

Table 16.2 Recommended Operating Conditions

Symbol	Parameter		Conditions	Standard			Unit
				Min.	Typ.	Max.	
V_{CC}	Supply voltage			2.7	5.0	5.5	V
AVcc	Analog supply voltage			(NOTE3, 4)	V_{CC}	—	V
Vss	Supply voltage			—	0	—	V
AVss	Analog supply voltage			—	0	—	V
V_{IH}	"H" input voltage			0.8V_{CC}	—	V_{CC}	V
V_{IL}	"L" input voltage			0	—	0.2V_{CC}	V
$I_{OH\ (sum)}$	"H" peak all output currents	Sum of all pins' IOH (peak)		—	—	-60.0	mA
$I_{OH\ (peak)}$	"H" peak output current			—	—	-10.0	mA
$I_{OH\ (avg)}$	"H" average output current			—	—	-5.0	mA
$I_{OL\ (sum)}$	"L" peak all output currents	Sum of all pins' IOL (peak)		—	—	60	mA
$I_{OL\ (peak)}$	"L" peak output current	Except P1$_0$ to P1$_7$		—	—	10	mA
		P1$_0$ to P1$_7$	Drive ability HIGH	—	—	30	mA
			Drive ability LOW	—	—	10	mA
$I_{OL\ (avg)}$	"L" average output current	Except P1$_0$ to P1$_7$		—	—	5	mA
		P1$_0$ to P1$_7$	Drive ability HIGH	—	—	15	mA
			Drive ability LOW	—	—	5	mA
f (X$_{IN}$)	Main clock input oscillation frequency		3.0V ≤ Vcc ≤ 5.5V	0	—	16	MHz
			2.7V ≤ Vcc < 3.0V	0	—	10	MHz

Note
1: Referenced to V_{CC} = AV$_{CC}$ = 2.7 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.
2: The mean output current is the mean value within 100ms.
3: When using 10 bit resolution mode of A-D converter, set AVcc ≥ 4.2V.
4: When using sample & hold function of A-D converter, set AVcc ≥ 4.2V.

Table 16.3 A-D Conversion Characteristics

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
–	Resolution		V_{ref} =V_{CC}			10	Bit
–	Absolute accuracy	10 bit mode	f(XIN)=øAD=10 MHz, Vref=Vcc=5.0V			±3	LSB
		8 bit mode	f(XIN)=øAD=10 MHz, Vref=Vcc=5.0V			±2	LSB
		10 bit mode	f(XIN)=øAD=10 MHz, Vref=Vcc=3.3V			±5	LSB
		8 bit mode	f(XIN)=øAD=10 MHz, Vref=Vcc=3.3V			±2	LSB
R_{LADDER}	Ladder resistance		V_{REF}=V_{CC}	10		40	kΩ
t_{CONV}	Conversion time	10 bit mode	f(XIN)=øAD=10 MHz, Vref=Vcc=5.0V	3.3			µs
		8 bit mode	f(XIN)=øAD=10 MHz, Vref=Vcc=5.0V	2.8			µs
t_{SAMP}	Sampling time			TBD			µs
V_{REF}	Reference voltage			2.0		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{ref}	V
–	A-D operation clock frequency[2]	Without sample & hold		0.25		10	MHz
		With sample & hold		1.0		10	MHz

Note

1: Referenced to V_{CC}=AV_{CC}=2.7 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.

2: When f_{AD} is 10 MHz more, divide the f_{AD} and make A-D operation clock frequency ($Ø_{AD}$) lower than 10 MHz.

3: When the Vcc is less than 4.2V, divide the f_{AD} and make A-D operation clock frequency ($Ø_{AD}$) lower than f_{AD}/2.

Table 16.4 Flash Memory Version Electrical Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max	
–	Byte program time		—	75	TBD	µs
–	Block erase time		—	400	TBD	ms
–	Program, Erase Voltage		2.7	—	5.5	V
–	Read Voltage		2.7	—	5.5	V
–	Program, Erase Temperature		0	—	60	°C

Note

1: Referenced to V_{CC1}=AVcc=2.7 to 5.5V at Topr = 0 to 60 °C unless otherwise specified.

Table 16.5 Power Circuit Timing Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Time for internal power supply stabilization during powering-on[2]				2	ms
td(R-S)	STOP release time[3]				150	µs

Note

1: The measureing condition is Vcc=AVcc=2.7 to 5.5 V and Topr=25 °C.

2: This shows the waiting time till the internal power supply generating circuit is stabilized during powering-on.

3: This shows the time till BCLK starts from the interrupt acknowledgement to cancel stop mode.

Table 16.6 Electrical Characteristics (1) [Vcc=5V]

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	"H" output voltage	Except XOUT	I_{OH}=-5mA		V_{CC}-2.0		V_{CC}	V
			I_{OH}=-200µA		V_{CC}-0.3		V_{CC}	V
		X_{OUT}	Drive ability HIGH	I_{OH}=-1 mA	V_{CC}-2.0		V_{CC}	V
			Drive ability LOW	I_{OH}=-500µA	V_{CC}-2.0		V_{CC}	V
V_{OL}	"L" output voltage	P10 to P17 Except X_{OUT}	I_{OH}= 5 mA				2.0	V
			I_{OH}= 200 µA				0.45	V
		P10 to P17	Drive ability HIGH	I_{OH}= 10 mA			2.0	V
			Drive ability LOW	I_{OH}= 5 mA			2.0	V
		X_{OUT}	Drive ability HIGH	I_{OH}= 1 mA			2.0	V
			Drive ability LOW	I_{OH}=500µA			2.0	V
V_{T+}-V_{T-}	Hysteresis	INT_0, INT_1, INT_2, INT_3, KI_0, KI_1, KI_2, KI_3, $CNTR_0$, $CNTR_1$, TC_{IN}, RxD_0, RxD_1			0.2		1.0	V
		RESET			0.2		2.2	V
I_{IH}	"H" input current		V_I=5V				5.0	µA
I_{IL}	"L" input current		V_I=0V				-5.0	µA
R_{PULLUP}	Pull-up resistance		V_I=0V		30	50	167	kΩ
R_{fXIN}	Feedback resistance	X_{IN}				1.0		MΩ
fRING	Ring oscillator frequency				40	125	250	kHz
V_{RAM}	RAM retention voltage		At stop mode		2.0			V

Note

1 : Referenced to V_{CC}=AV_{CC}=4.2 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=20MHz unless otherwise specified.

Table 16.7 Electrical Characteristics (2) [Vcc=5V]

Symbol	Parameter	Measuring condition		Standard			Unit
				Min.	Typ.	Max.	
I_{CC}	Power supply current (V_{CC}=4.2 to 5.5V) In single-chip mode, the output pins are open and other pins are Vss	High-speed mode	X_{IN}=16 MHz (square wave) Ring oscillator on=100 kHz No division		9.0	TBD	mA
			X_{IN}=5 MHz (square wave) Ring oscillator on=100 kHz No division		3.5		mA
		Medium-speed mode	X_{IN}=16 MHz (square wave) Ring oscillator on=100 kHz Division by 8		TBD		mA
			X_{IN}=5 MHz (square wave) Ring oscillator on=100 kHz Division by 8		TBD		mA
		Ring oscillator mode	Main clock off Ring oscillator on=100 kHz Division by 8		0.8		mA
		Wait mode	Main clock off Ring oscillator on=100 kHz At wait mode[2] Peripheral clock operation		43		µA
		Wait mode	Main clock off Ring oscillator on=100 kHz At wait mode[2] Peripheral clock off		33		µA
		Stop mode	Main clock off Ring oscillator off CM10="1" Peripheral clock off		1.0	TBD	µA

Note

1: The power supply current measuring is executed using the measuring program on frash memory.

2: Timer Y is operated with timer mode.

Table 16.8 Electrical Characteristics (3) [Vcc=3V]

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	"H" output voltage	Except XOUT	I_{OH}=-1mA		V_{CC}-0.5		V_{CC}	V
		X_{OUT}	Drive ability HIGH	I_{OH}=-0.1 mA	V_{CC}-0.5		V_{CC}	V
			Drive ability LOW	I_{OH}=-50 µA	V_{CC}-0.5		V_{CC}	V
V_{OL}	"L" output voltage	P1$_0$ to P1$_7$ Except X_{OUT}	I_{OH}= 1 mA				0.5	V
		P1$_0$ to P1$_7$	Drive ability HIGH	I_{OH}= 2 mA			0.5	V
			Drive ability LOW	I_{OH}= 1 mA			0.5	V
		X_{OUT}	Drive ability HIGH	I_{OH}= 0.1 mA			0.5	V
			Drive ability LOW	I_{OH}=50 µA			0.5	V
V_{T+}-V_{T-}	Hysteresis	INT$_0$, INT$_1$, INT$_2$, INT$_3$, KI$_0$, KI$_1$, KI$_2$, KI$_3$, CNTR$_0$, CNTR$_1$, TC$_{IN}$, RxD$_0$, RxD$_1$			0.2		0.8	V
		RESET			0.2		1.8	V
I_{IH}	"H" input current		V_I=3V				4.0	µA
I_{IL}	"L" input current		V_I=0V				-4.0	µA
R_{PULLUP}	Pull-up resistance		V_I=0V		66	160	500	kΩ
R_{fXIN}	Feedback resistance	X_{IN}				3.0		MΩ
fRING-S	Low-speed ring oscillator frequency				40	125	250	kHz
V_{RAM}	RAM retention voltage		At stop mode		2.0			V

Note
1 : Referenced to V_{CC}=AV_{CC}=2.7 to 3.3V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=5MHz unless otherwise specified.

Table 16.9 Electrical Characteristics (4) [Vcc=3V]

Symbol	Parameter	Measuring condition		Standard			Unit
				Min.	Typ.	Max.	
I$_{CC}$	Power supply current (V$_{CC1}$=2.7 to 3.3V) In single-chip mode, the output pins are open and other pins are V$_{SS}$	High-speed mode	X$_{IN}$=16 MHz (square wave) Ring oscillator on=100 kHz No division		8.0	TBD	mA
			X$_{IN}$=5 MHz (square wave) Ring oscillator on=100 kHz No division		3.0		mA
		Medium-speed mode	X$_{IN}$=16 MHz (square wave) Ring oscillator on=100 kHz Division by 8		TBD		mA
			X$_{IN}$=5 MHz (square wave) Low-speed ring oscillator on=100 kHz Division by 8		TBD		mA
		Ring oscillator mode	Main clock off Ring oscillator on=100 kHz Division by 8		0.8		mA
		Wait mode	Main clock off Ring oscillator on=100 kHz At wait mode^2 Peripheral clock operation		TBD		µA
		Wait mode	Main clock off Ring oscillator on=100 kHz At wait mode^2 Peripheral clock off		TBD		µA
		Stop mode	Main clock off Ring oscillator off CM10="1" Peripheral clock off		1.0	TBD	µA

Note
1: The power supply current measuring is executed using the measuring program on frash memory.
2: Timer Y is operated with timer mode.

17. Flash Memory Version

17.1 Overview

The flash memory version has two modes—CPU rewrite and standard serial input/output—in which its flash memory can be operated on.

Table 17.1 outlines the performance of flash memory version (see “Table 1.1 Performance” for the items not listed on Table 17.1).

Table 17.1 Flash Memory Version Performance

Item	Specification
Flash memory operating mode	2 modes (CPU rewrite and standard serial I/O)
Erase block	See “Figure 17.1. Flash Memory Block Diagram”
Method for program	In units of byte
Method for erasure	Block erase
Program, erase control method	Program and erase controlled by software command
Protect method	Blocks 0 and 1 protected by block 0, 1 program enable bit
Number of commands	5 commands
Number of program and erasure	100 times
Data Retention	10 years
ROM code protection	Standard serial I/O mode is supported.

Table 17.2 Flash Memory Rewrite Modes

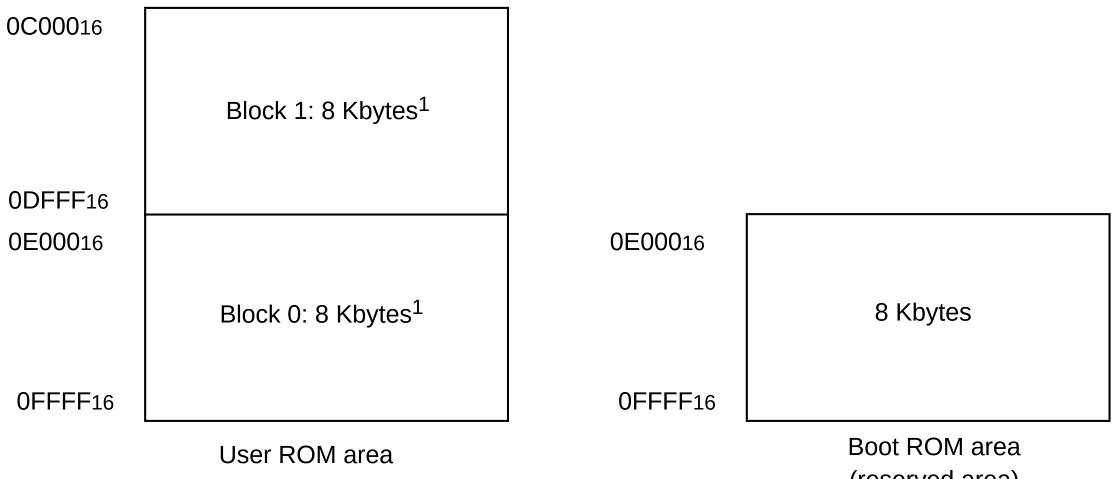
Flash memory rewrite mode	CPU rewrite mode	Standard serial I/O mode
Function	User ROM area is rewritten by executing software commands from the CPU. EW0 mode: Can be rewritten in any area other than the flash memory EW1 mode: Can be rewritten in the flash memory	User ROM area is rewritten by using a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

17.2 Memory Map

The ROM in the flash memory version is separated between a user ROM area and a boot ROM area (reserved area). Figure 17.1 shows the block diagram of flash memory.

The user ROM area is divided into several blocks. The user ROM area can be rewritten in CPU rewrite and standard serial input/output modes. Block 1 and Block 0 are enabled for rewrite in CPU rewrite mode by setting the FMR02 bit in the FMR0 register to “1” (rewrite enabled).

The rewrite program for standard serial I/O mode is stored in the boot ROM area before shipment.



Notes:
1. Block 1 and Block 0 are enabled for rewrite in CPU rewrite mode by setting the FMR02 bit in the FMR0 register to “1” (rewrite enabled).

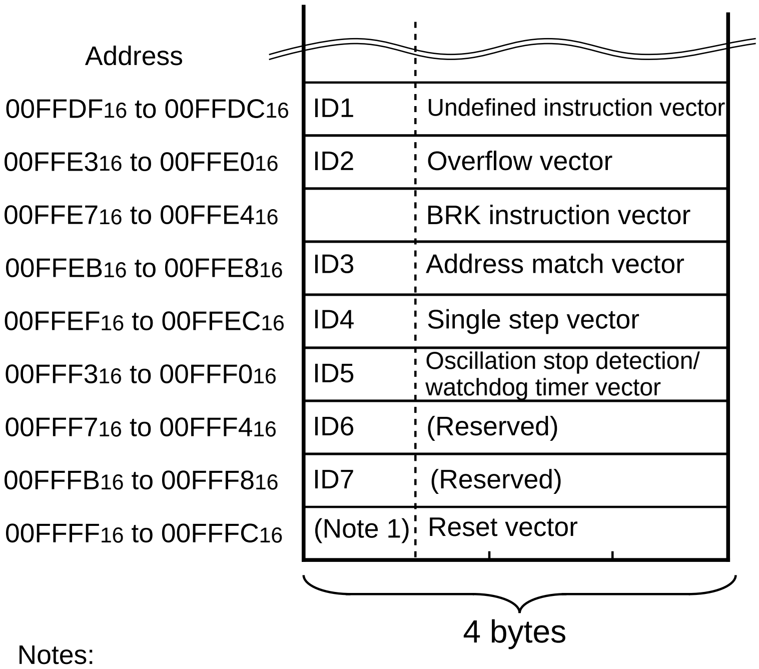
Figure 17.1 Flash Memory Block Diagram

17.3 Functions To Prevent Flash Memory from Rewriting

To prevent the flash memory from being read or rewritten easily, standard serial input/output mode has an ID code check function.

17.3.1 ID Code Check Function

Use this function in standard serial input/output mode. 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 00FFDF$_{16}$, 00FFE3$_{16}$, 00FFEB$_{16}$, 00FFEF$_{16}$, 00FFF3$_{16}$, 00FFF7$_{16}$, and 00FFFB$_{16}$. Prepare a program in which the ID codes are preset at these addresses and write it in the flash memory.



Address		
00FFDF$_{16}$ to 00FFDC$_{16}$	ID1	Undefined instruction vector
00FFE3$_{16}$ to 00FFE0$_{16}$	ID2	Overflow vector
00FFE7$_{16}$ to 00FFE4$_{16}$		BRK instruction vector
00FFEB$_{16}$ to 00FFE8$_{16}$	ID3	Address match vector
00FFEF$_{16}$ to 00FFEC$_{16}$	ID4	Single step vector
00FFF3$_{16}$ to 00FFF0$_{16}$	ID5	Oscillation stop detection/ watchdog timer vector
00FFF7$_{16}$ to 00FFF4$_{16}$	ID6	(Reserved)
00FFFB$_{16}$ to 00FFF8$_{16}$	ID7	(Reserved)
00FFFF$_{16}$ to 00FFFC$_{16}$	(Note 1)	Reset vector

Notes:
1. When write to address 00FFFF$_{16}$, write “FF$_{16}$”.

Figure 17.2 Address for ID Code Stored

17.4 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.

For interrupts requested during an erase operation in CPU rewrite mode, the R8C/10 flash module offers an `erase-suspend` feature which allow the erase operation to be suspended, and access made available to the flash.

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 17.3 lists the differences between Erase Write 0 (EW0) and Erase Write 1 (EW1) modes.

Table 17.3 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	User ROM area However, this does not include the block in which a rewrite control program exists[1]
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)
Flash memory status detection	• Read the FMR0 register FMR00, FMR06, and FMR07 bits in a program • Execute the Read Status Register command to read the status register SR7, SR5, and SR4.	Read the FMR0 register FMR00, FMR06, and FMR07 bits in a program
Conditions for transferring to erase-suspend	Set the FMR40 and FMR41 bits in the FMR4 register to “1” by program.	When an interrupt which is set for enabled occurs while the FMR40 bit in the FMR4 register is set to “1”.

Notes:

1. Block 1 and Block 0 are enabled for rewrite by setting the FMR02 bit in the FMR0 register to “1” (rewrite enabled).

17.4.1 EW0 Mode

The microcomputer is placed in CPU rewrite mode by setting the FMR01 bit in the FMR0 register 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.

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 moving to an erase-suspend, set the FMR40 bit to “1” (erase-suspend) and the FMR41 bit to “1” (suspend requested). Make sure that the FMR46 bit is set to “1” (auto-erase inactive) before accessing the user ROM space. The erase operation resumes by setting the FMR41 bit to “0” (erase restart).

17.4.2 EW1 Mode

EW1 mode is selected by setting FMR11 bit to “1” (EW1 mode) after setting the FMR01 bit to “1” (CPU rewrite mode enabled).

Read the FMR0 register to check the status of program or erase operation at completion. Avoid executing software commands of Read Status register in EW1 mode.

To enable the erase-suspend function, the Block Erase command should be executed after setting the FMR40 bit to “1” (erase-suspend enabled). An interrupt to request an erase-suspend must be in enabled state. Once being placed in an erase-suspend upon the interrupt request, the user ROM space can be accessed and the CPU starts operating.

The FMR41 bit is automatically set to “1” (suspend requested) if the auto-erase operation is halted by an interrupt request. If the erase operation is not completed (FMR00 bit is “0”) when the interrupt routine is ended, the Block Erase command should be executed again by setting the FMR41 bit to “0” (erase restart).

Figure 17.3 shows the FMR0 and FMR1 registers. Figure 17.4 shows the FMR4 register.

- **FMR00 Bit**

  This bit indicates the operating status of the flash memory. The bit is "0" during programming, erasing, or erase-suspend mode; otherwise, the bit is "1".
- **FMR01 Bit**

  The microcomputer is made ready to accept commands by setting the FMR01 bit to "1" (CPU rewrite mode).
- **FMR02 Bit**

  The Block1 and Block0 do not accept the Program and Block Erase commands if the FMR02 bit is set to "0" (rewrite disabled).
- **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 flash memory is disabled against access by setting the FMSTP bit to "1". Therefore, the FMSTP bit must be written to by a program in 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 ring oscillator mode (main clock stop).

  Figure 17.6 shows a flow chart to be followed before and after entering ring oscillator mode (main clock stop).

  Note that when going to stop or wait mode while the CPU rewrite mode is disabled, the FMR0 register does not need to be set because the power for the flash memory is automatically turned off and is turned back on again after returning from stop or wait mode.
- **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 the full status check.
- **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 set to "0". For details, refer to the description of "17.4.5 Full status check".
- **FMR11 Bit**

  Setting this bit to "1" (EW1 mode) places the microcomputer in EW1 mode.
- **FMR40 bit**

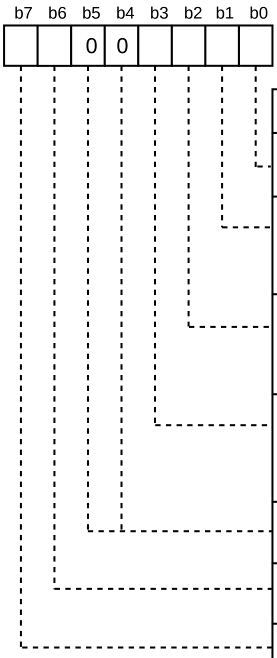
  The erase-suspend function is enabled by setting the FMR40 bit to "1" (valid).
- **FMR41 bit**

  In EW0 mode, the flash module goes to erase-suspend mode when the FMR41 bit is set to "1". In EW1 mode, the FMR41 bit is automatically set to "1" (suspend requested) when an enabled interrupt occurred, and then the flash module goes to erase-suspend mode.

  The auto-erase operation restarts when the FMR41 bit is set to "0" (erase restart).
- **FMR46 bit**

  The FMR46 bit is set to "0" during auto-erase execution and set to "1" during erase-suspend mode. Avoid accessing to the flash memory when this bit is set to "0".

Flash memory control register 0



Symbol	Address	After reset
FMR0	$01B7_{16}$	$XX000001_2$

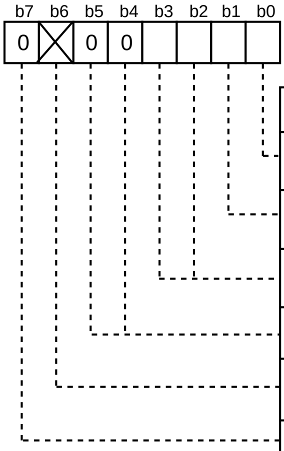
Bit symbol	Bit name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0: Busy (being written or erased) 1: Ready	RO
FMR01	CPU rewrite mode select bit[1]	0: Disables CPU rewrite mode 1: Enables CPU rewrite mode	RW
FMR02	Block1, 0 rewrite enable bit[2]	0: Enables lock bit 1: Disables lock bit	RW
FMSTP	Flash memory stop bit[3, 5]	0: Enables flash memory operation 1: Stops flash memory operation (placed in low power mode, flash memory initialized)	RW
— (b5-b4)	Reserved bit	Must set to “0”	RW
FMR06	Program status flag[4]	0: Terminated normally 1: Terminated in error	RO
FMR07	Erase status flag[4]	0: Terminated normally 1: Terminated in error	RO

Notes:

1. To set this bit to “1”, write “0” and then “1” in succession. Make sure no interrupts will occur before writing “1” after writing “0”.
   Set the microcomputer in read array mode before writing to this bit.
2. To set this bit to “1”, write “0” and then “1” in succession when the FMR01 bit = 1. Make sure no interrupts will occur before writing “1” after writing “0”.
3. Write to this bit from a program in other than the flash memory.
4. This flag is set to “0” by executing the Clear Status command.
5. Effective when the FMR01 bit = 1 (CPU rewrite mode). If the FMR01 bit = 0, although the FMSTP bit can be set to “1” by writing “1”, the flash memory is neither placed in low power mode nor initialized.

Figure 17.3 FMR0 Register

Flash memory control register 1

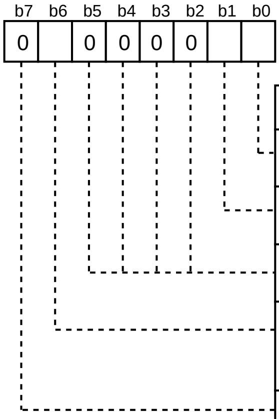


Symbol	Address	After reset
FMR1	$01B5_{16}$	$0100XX0X_2$

Bit symbol	Bit name	Function	RW
(b0)	Reserved bit	When read, its content is indeterminate.	RO
FMR11	EW1 mode select bit[1]	0: EW0 mode 1: EW1 mode	RW
(b3-b2)	Reserved bit	When read, its content is indeterminate.	RO
(b5-b4)	Reserved bit	Must set to “0”	RW
(b6)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		RO
(b7)	Reserved bit	Must set to “0”	RW

Notes:
1. To set this bit to “1”, write “0” and then “1” in succession when the FMR01 bit = 1. Make sure no interrupts will occur before writing “1” after writing “0”.
The FMR01 and FMR11 bits both are set to “0” by setting the FMR01 bit to “0”.

Flash memory control register 4



Symbol	Address	After reset
FMR4	$01B3_{16}$	$0100000X_2$

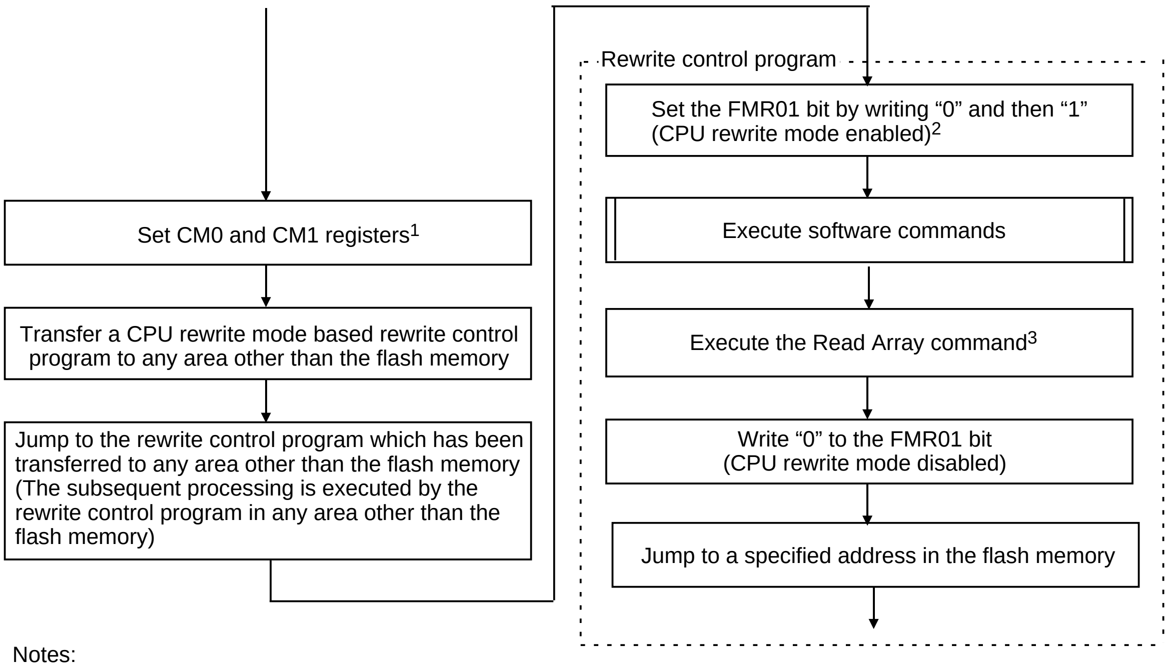
Bit symbol	Bit name	Function	RW
FMR40	Erase-suspend function enable bit[1]	0: Invalid 1: Valid	RW
FMR41	Erase-suspend request bit[2]	0: Erase restart 1: Suspend request	RW
(b5-b2)	Reserved bit	Must set to “0”	RO
FMR46	Erase status	0: Auto-erase active 1: Auto-erase inactive (erase-suspend mode)	RO
(b7)	Reserved bit	Must set to “0”	RW

Notes:
1. To set this bit to “1”, write “0” and then “1” in succession. Make sure no interrupts will occur before writing “1” after writing “0”.
2. This bit is valid only when the FMR40 bit is set to “1” (valid) and can only be written before ending an erase after issuing an erase command. Other than this period, this bit is set to “1”.
In EW0 mode, this bit can be set to “0” and “1” by program.
In EW1 mode, this bit is automatically set to “1” if a maskable interrupt occurs during an erase operation while the FMR40 bit is set to “1”. This bit can not be set to “1” by program. (Can be set to “0”.)

Figure 17.3-2 FMR1 and FMR4

Figures 17.5 and 17.6 show the setting and resetting of EW0 mode and EW1 mode, respectively.

EW0 mode operation procedure

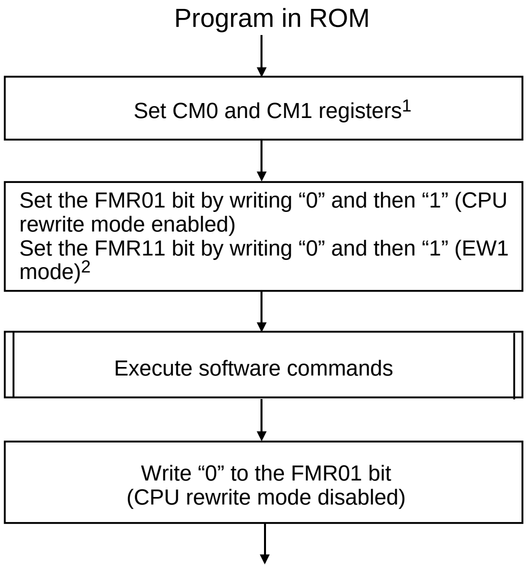

Rewrite control program
Set CM0 and CM1 registers[1]
Transfer a CPU rewrite mode based rewrite control program to any area other than the flash memory
Jump to the rewrite control program which has been transferred to any area other than the flash memory (The subsequent processing is executed by the rewrite control program in any area other than the flash memory)
Set the FMR01 bit by writing "0" and then "1" (CPU rewrite mode enabled)[2]
Execute software commands
Execute the Read Array command[3]
Write "0" to the FMR01 bit (CPU rewrite mode disabled)
Jump to a specified address in the flash memory

Notes:

1. Select 5 MHz or less for CPU clock using the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register.
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 from a program in other than the flash memory.
3. Disables the CPU rewrite mode after executing the Read Array command.

Figure 17.4 Setting and Resetting of EW0 Mode

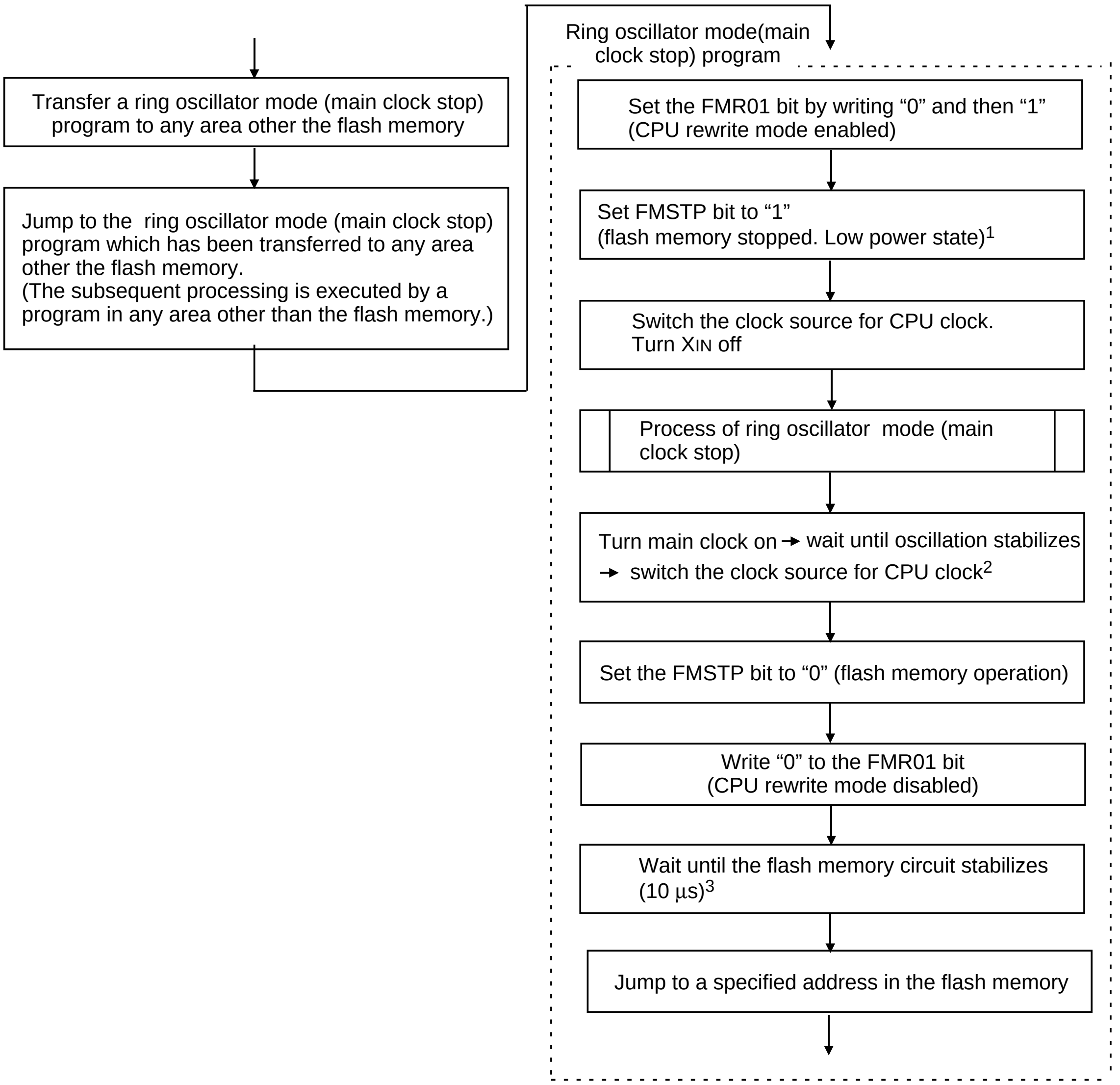
EW1 mode operation procedure


Program in ROM
Set CM0 and CM1 registers[1]
Set the FMR01 bit by writing "0" and then "1" (CPU rewrite mode enabled)
Set the FMR11 bit by writing "0" and then "1" (EW1 mode)[2]
Execute software commands
Write "0" to the FMR01 bit (CPU rewrite mode disabled)

Notes:

1. Select 5 MHz or less for CPU clock using the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register.
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 17.5 Setting and Resetting of EW1 Mode


Transfer a ring oscillator mode (main clock stop) program to any area other the flash memory
Jump to the ring oscillator mode (main clock stop) program which has been transferred to any area other the flash memory.
(The subsequent processing is executed by a program in any area other than the flash memory.)
Ring oscillator mode(main clock stop) program
Set the FMR01 bit by writing "0" and then "1" (CPU rewrite mode enabled)
Set FMSTP bit to "1" (flash memory stopped. Low power state)1
Switch the clock source for CPU clock. Turn XIN off
Process of ring oscillator mode (main clock stop)
Turn main clock on → wait until oscillation stabilizes → switch the clock source for CPU clock2
Set the FMSTP bit to "0" (flash memory operation)
Write "0" to the FMR01 bit (CPU rewrite mode disabled)
Wait until the flash memory circuit stabilizes (10 µs)3
Jump to a specified address in the flash memory

Notes:
1. Set the FMR01 bit to "1" (CPU rewrite mode) before setting the FMSTP bit to "1" .
2. Before the clock source for CPU clock can be changed, the clock to which to be changed must be stable.
3. Insert a 10 µs wait time in a program. Avoid accessing to the flash memory during this wait time.

Figure 17.6 Processing Before and After Ring Oscillator Mode (main clock stop)

17.4.3 Software Commands

Software commands are described below. The command code and data must be read and written in 8-bit units.

Table 17.4 Software Commands

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_7 to D_0)	Mode	Address	Data (D_7 to D_0)
Read array	Write	X	FF_{16}			
Read status register	Write	X	70_{16}	Read	X	SRD
Clear status register	Write	X	50_{16}			
Program	Write	WA	40_{16}	Write	WA	WD
Block erase	Write	X	20_{16}	Write	BA	$D0_{16}$

SRD: Status register data (D_7 to D_0)
WA: Write address (Make sure the address value specified in the the first bus cycle is the same address as the write address specified in the second bus cycle.)
WD: Write data (8 bits)
BA: Uppermost block address
X: Any address in the user ROM area

- **Read Array Command**

This command reads the flash memory.

Writing ‘FF_{16}’ 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 8-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.

- **Read Status Register Command**

This command reads the status register.

Write ‘70_{16}’ in the first bus cycle, and the status register can be read in the second bus cycle. (Refer to Section 17.4.4, “Status Register.”) When reading the status register too, specify an address in the user ROM area.

Avoid executing this command in EW1 mode.

- **Clear Status Register Command**

This command sets the status register to “0”.

Write ‘50_{16}’ in the first bus cycle, and the FMR06 to FMR07 bits in the FMR0 register and SR4 to SR5 in the status register will be set to “0”.

- **Program**

This command writes data to the flash memory in one byte units.

Write ‘40_{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 address as the write address specified in the second bus cycle.

Check the FMR00 bit in the FMR0 register 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 FMR06 bit in the FMR0 register after auto programming has finished, and the result of auto programming can be known. (Refer to Section 17.4.6, “Full Status Check.”)

Writing over already programmed addresses is inhibited.

When the FMR02 bit in the FMR0 register is set to “0” (rewrite disabled), the Program command on the Block0 and Block1 is not accepted.

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 set 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.

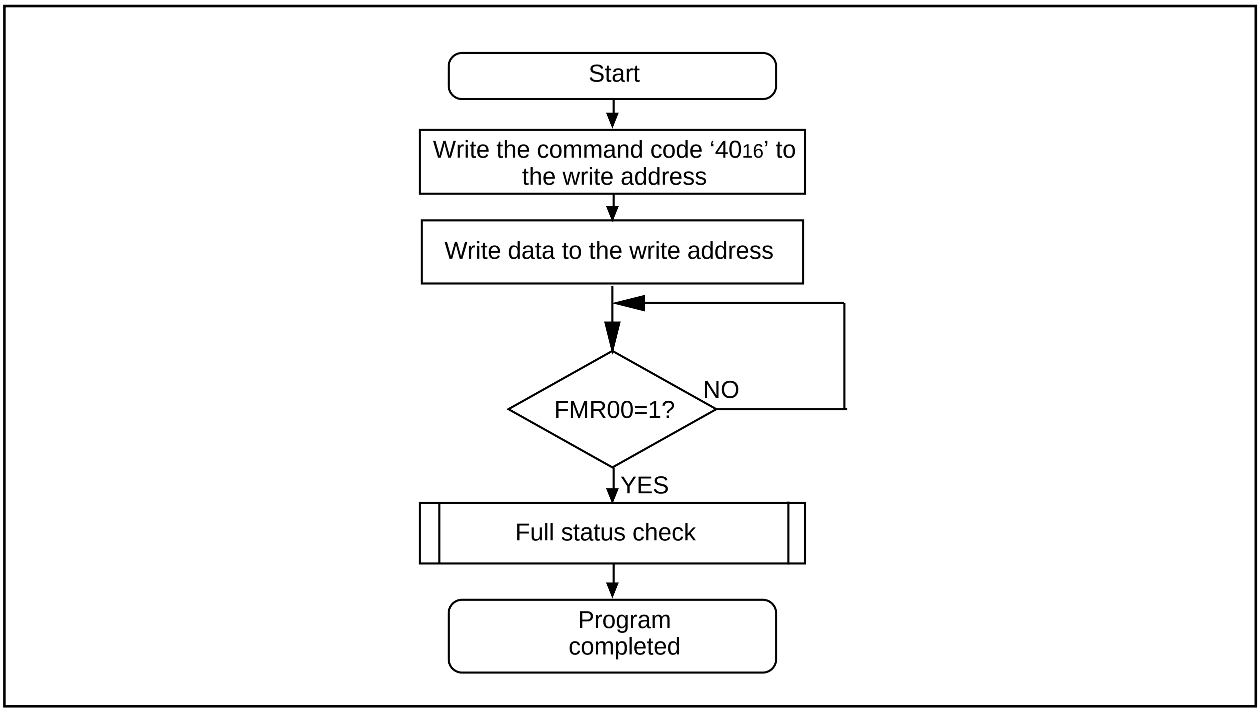

Start
Write the command code ‘4016’ to the write address
Write data to the write address
FMR00=1?
NO
YES
Full status check
Program completed

Figure 17.7 Program Flow Chart

- **Block Erase**

Write ‘20_{16}’ in the first bus cycle and write ‘$D0_{16}$’ to the uppermost address of a block in the second bus cycle, and an auto erase operation (erase and verify) will start.
Check the FMR00 bit in the FMR0 register to see if auto erasing has finished.
The FMR00 bit is “0” during auto erasing and set to “1” when auto erasing is completed.
When using the erase-suspend function in EW0 mode, the FMR46 bit in the FMR4 register should be checked to see if the flash memory is placed in a erase-suspend. The FMR46 bit is set to “0” when auto-erase operation is active and set to “0” auto-erase operation is inactive.
Check the FMR07 bit in the FMR0 register after auto erasing has finished, and the result of auto erasing can be known. (Refer to Section 17.4.6, “Full Status Check.”)
When the FMR02 bit in the FMR0 register is set to “0” (rewrite disabled), the Block Erase command on the Block0 and Block1 is not accepted.
Figure 17.9 shows an example of a block erase flowchart when the erase-suspend function is not used. Figure 17.10 shows an example of a block erase flowchart when the erase-suspend function is used.
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.

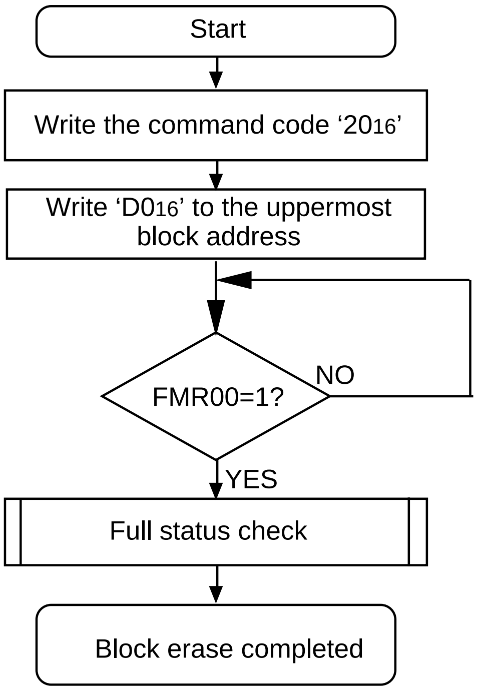
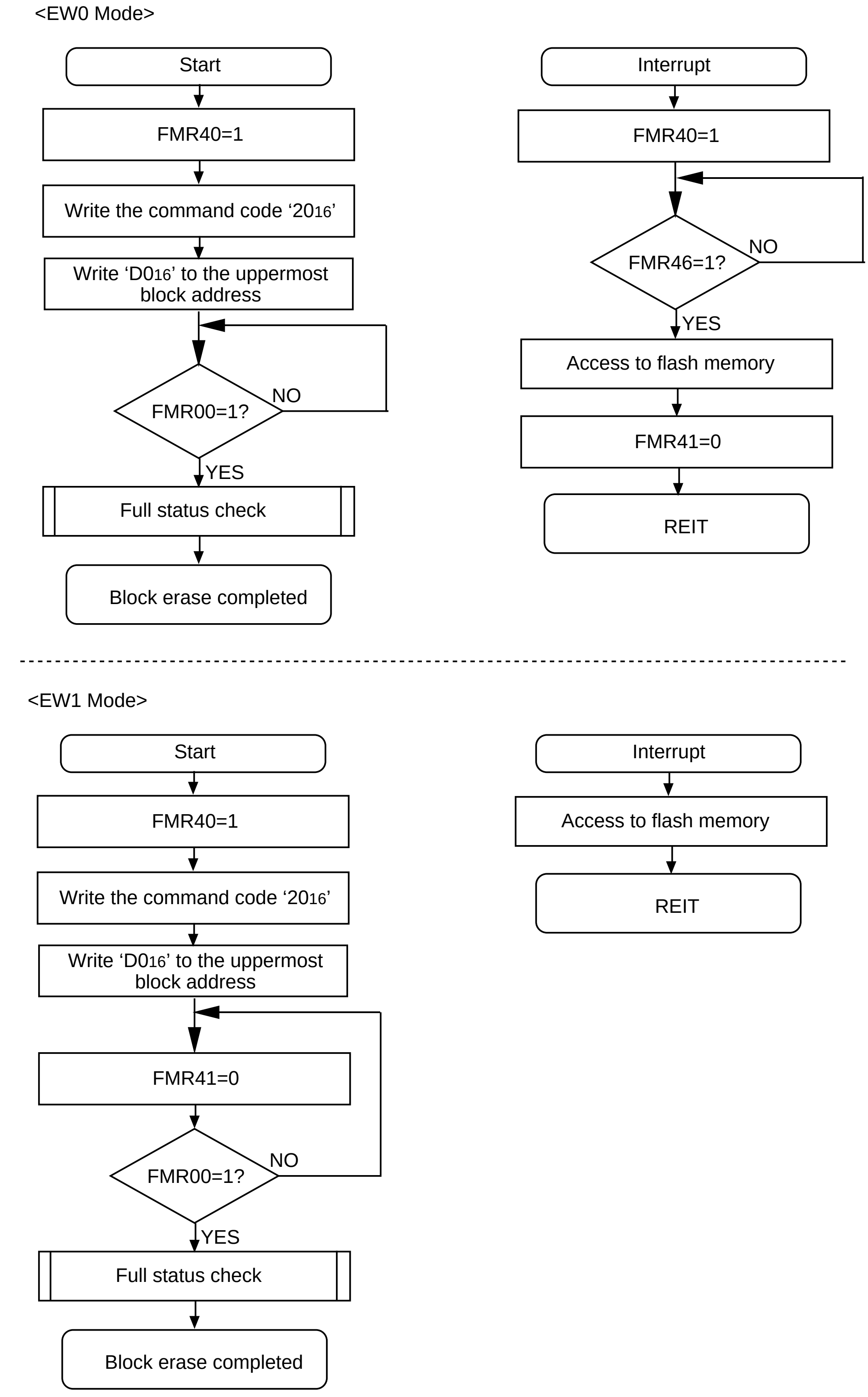

Start
Write the command code ‘2016’
Write ‘D016’ to the uppermost block address
FMR00=1?
NO
YES
Full status check
Block erase completed

Figure 17.8 Block Erase Flow Chart (When Not Using Erase-suspend Function)



Notes:
1. In EW0 mode, interrupt vector table for an interrupt used should be located in the RAM space.

Figure 17.9 Block Erase Command (When Using Erase-suspend Function)

17.4.4 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 FMR00, FMR06, and FMR07 bits in the FMR0 register.

Table 17.5 lists the status register.

In EW0 mode, the status register can be read in the following cases:

(1) When a given address in the user ROM area is read after writing the Read Status Register command

(2) When a given address in the user ROM area is read after executing the Program or Block Erase command but before executing the Read Array command.

- **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, and is set to “1” (ready) at the same time the operation finishes. SR7 = 0 (busy) during erase suspend mode.

- **Erase Status (SR5 and FMR07 Bits)**

Refer to Section 17.4.6, “Full Status Check.”

- **Program Status (SR4 and FMR06 Bits)**

Refer to Section 17.4.6, “Full Status Check.”

Table 17.5 Status Register

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

- D7 to D0: Indicates the data bus which is read out when the Read Status Register command is executed.
- The FMR07 bit (SR5) and FMR06 bit (SR4) are set to “0” by executing the Clear Status Register command.
- When the FMR07 bit (SR5) or FMR06 bit (SR4) = 1, the Program and Block Erase commands are not accepted.

17.4.5 Full Status Check

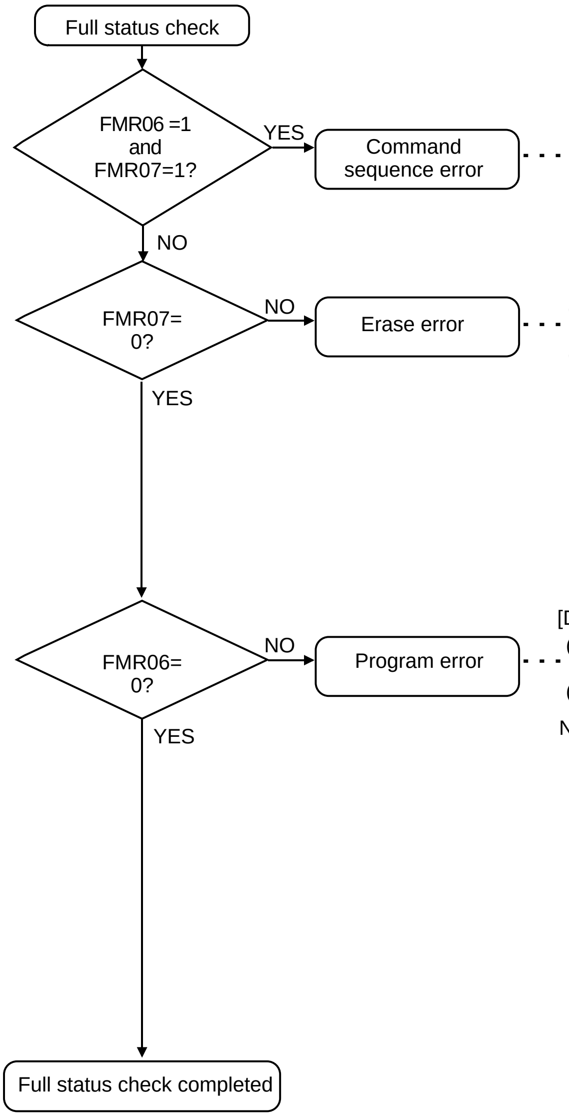
When an error occurs, the FMR06 to FMR07 bits in the FMR0 register are set to “1”, indicating occurrence of each specific error. Therefore, execution results can be verified by checking these status bits (full status check). Table 17.6 lists errors and FMR0 register status. Figure 17.11 shows a full status check flowchart and the action to be taken when each error occurs.

Table 17.6 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 ‘D0$_{16}$’ or ‘FF$_{16}$’)[1]
1	0	Erase error	• When the Block Erase command was executed but not automatically erased correctly
0	1	Program error	• When the Program command was executed but not automatically programmed correctly.

Notes:

1. Writing ‘FF$_{16}$’ in the second bus cycle of these commands places the microcomputer in read array mode, and the command code written in the first bus cycle is nullified.



(1) Execute the Clear Status Register command to set these status flags to “0”.
(2) Re-execute the command after checking that it is entered correctly.

(1) Execute the Clear Status Register command to set the erase status flag to “0”.
(2) Re-execute the Block Erase command.

Note 1: If the error still occurs, the block in error cannot be used.

[During programming]
(1) Execute the Clear Status Register command to set the program status flag to “0”.
(2) Re-execute the Program command.

Note 2: If the error still occurs, the block in error cannot be used.

Note 3: If FMR06 or FMR07 = 1, neither the Program nor Block Erase command is accepted. Execute the Clear Status Register command before executing those commands.

Figure 17.10 Full Status Check and Handling Procedure for Each Error

17.5 Standard Serial I/O Mode

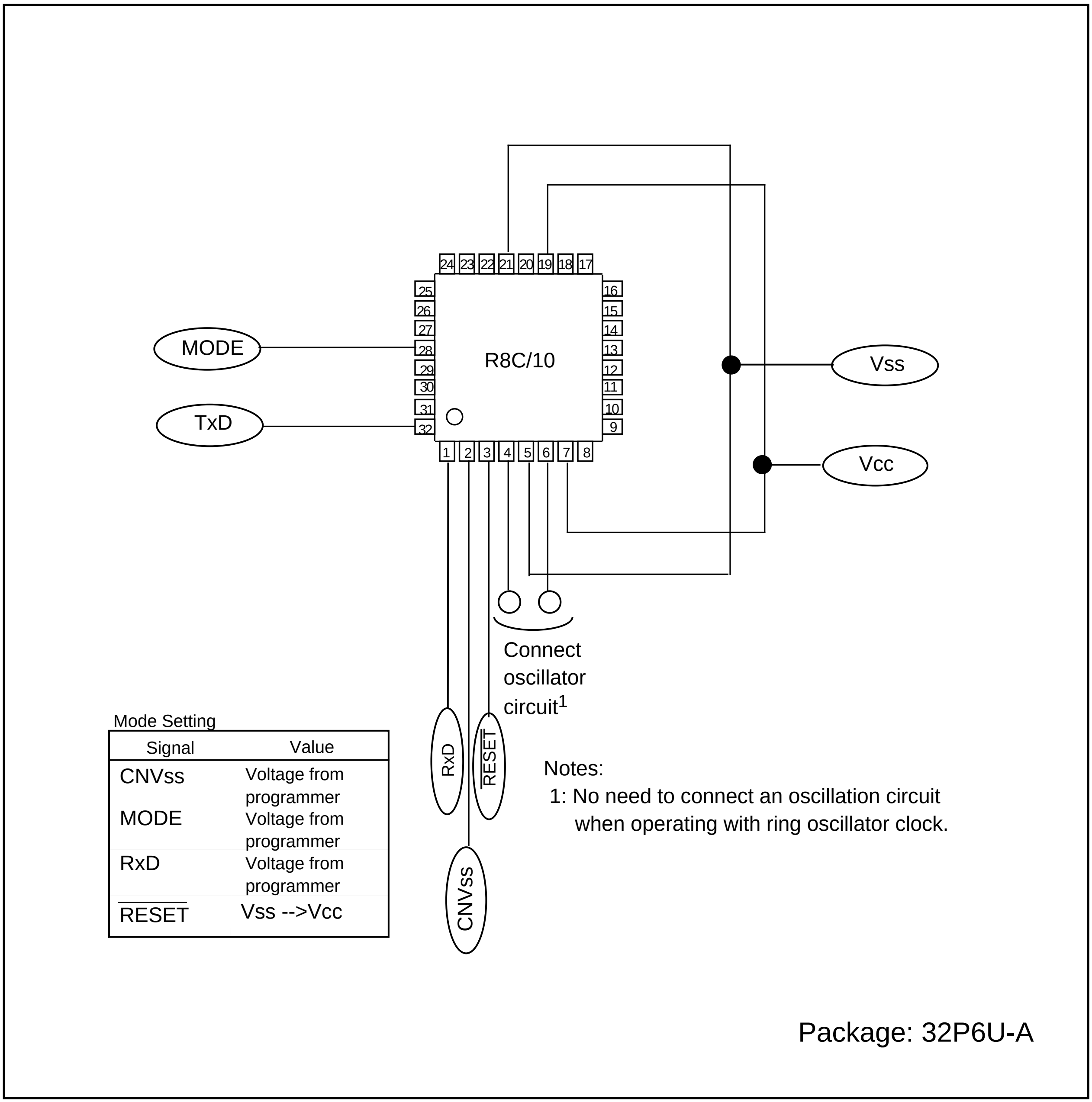
In standard serial input/output mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer suitable for this microcomputer. For more information about serial programmers, contact the manufacturer of your serial programmer. For details on how to use, refer to the user’s manual included with your serial programmer.
Table 17.7 lists pin functions (flash memory standard serial input/output mode). Figures 17.12 to 17.14 show pin connections for standard serial input/output mode.

17.5.1 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 Section 17.3, “Functions to Prevent Flash Memory from Rewriting”).

Table 17.7 Pin Functions (Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply the voltage guaranteed for Program and Erase to Vcc pin and 0V to Vss pin.
IVCC	IVCC		Connect capacitor (0.1 µF) to Vss.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is "L" level, input a 20 cycle or longer clock to XIN pin.
P46/XIN	P46 input/Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins in standard serial I/O mode 2. In standard serial I/O mode 1, connect a ceramic resonator or crystal oscillator between XIN and XOUT pins, or input "H" or "L" level signal, or open.
P47/XOUT	P47 input/Clock output	I/O	
AVCC, AVSS	Analog power supply input	I	Connect AVss to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for AD from this pin.
P01 to P07	Input port P0	I	Input "H" or "L" level signal or open.
P10 to P17	Input port P1	I	Input "H" or "L" level signal or open.
P30 to P33	Input port P3	I	Input "H" or "L" level signal or open.
P45	Input port P4	I	Input "H" or "L" level signal or open.
P00	TxD output	O	Serial data output pin
MODE	MODE	I/O	Standard serial I/O mode 1: connect to flash programmer Standard serial I/O mode 2: Input "L".
CNVSS	CNVSS	I/O	Standard serial I/O mode 1: connect to flash programmer Standard serial I/O mode 2: Input "L".
P37	RxD input	O	Serial data input pin



Mode Setting

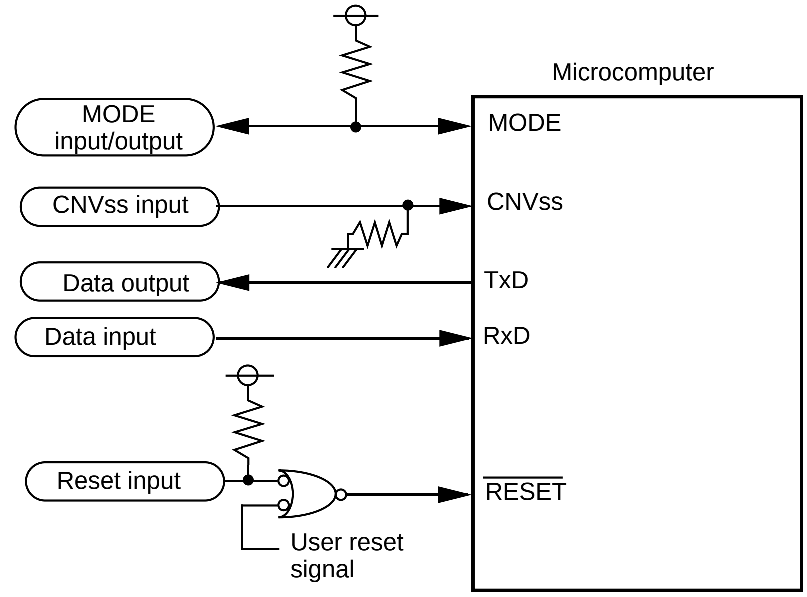
Signal	Value
CNVss	Voltage from programmer
MODE	Voltage from programmer
RxD	Voltage from programmer
$\overline{\text{RESET}}$	Vss -->Vcc

Notes:
1: No need to connect an oscillation circuit when operating with ring oscillator clock.

Figure 17.11 Pin Connections for Standard Serial I/O Mode

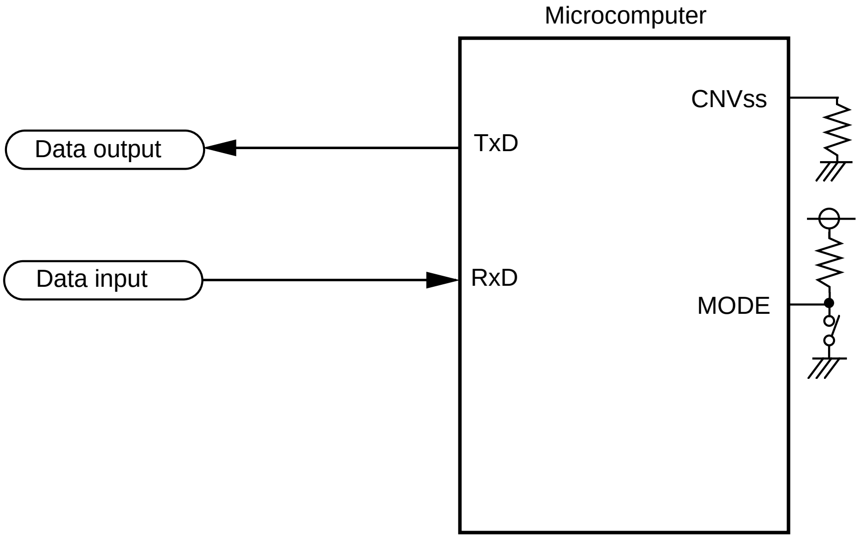
• Example of Circuit Application in the Standard Serial I/O Mode

Figures 17.12 and 17.13 show examples of circuit application in standard serial I/O mode 1 and mode 2, respectively. Refer to the serial programmer manual of your programmer to handle pins controlled by the programmer.



(1) Control pins and external circuitry will vary according to programmer. For more information, see the programmer manual.
(2) In this example, modes are switched between single-chip mode and standard serial input/output mode by connecting a programmer.

Figure 17.12 Circuit Application in Standard Serial I/O Mode 1



(1) In this example, modes are switched between single-chip mode and standard serial input/output mode by controlling the MODE input with a switch.

Figure 17.13 Circuit Application in Standard Serial I/O Mode 2

18. On-chip debugger

The microcomputer has functions to execute the on-chip debugger. Refer to "Appendix 2 Connecting examples for serial writer and on-chip debugging emulator". Refer to the respective on-chip debugger manual for the details of the on-chip debugger. Next, here are some explanations for the respective functions. Debugging the user system which uses these functions is not available. When using the on-chip debugger, design the system without using these functions in advance. Additionally, the on-chip debugger uses the address "$0C000_{16}$ to $0C7FF_{16}$ of the flash memory, thus avoid using for the user system.

18.1 Address match interrupt

The interrupt request is generated right before the arbitrary address instruction is executed. The debugger break function uses the address match interrupt. Refer to "10.4 Address match interrupt" for the details of the address match interrupt. Also, avoid using the address match interrupt with using the user system when using the on-chip debugger.

18.2 Single step interrupt

The interrupt request is generated every time one instruction is executed. The debugger single step function uses the single step interrupt. The other interrupt is not generated when using the single step interrupt. The single step interrupt is only for the developed support tool.

18.3 UART1

The UART1 is used for the communication with the debugger (or the personal computer). Refer to "13. Serial I/O" for the details of UART1. Also, avoid using the UART1 and the functions ($P0_0/AN_7$ and $P3_7$) which share the UART1 pins.

18.4 BRK instruction

The BRK interrupt request is generated. Refer to "10.1 Interrupt overview" and "R8C/Tiny series software manual". Also, avoid using the BRK instruction with using the user system when using the on-chip debugger.

19. Usage Notes

19.1 Stop Mode and Wait Mode

When entering stop mode or wait mode, an instruction queue pre-reads 4 bytes from the WAIT instruction or an instruction that sets the CM10 bit in the CM1 register to “1” (all clocks stopped) before the program stops. Therefore, insert at least four NOPs after the WAIT instruction or an instruction that sets the CM10 bit to “1”.

19.2 Interrupts

19.2.1 Reading Address 00000_{16}

Avoid reading 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 set 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 set to “0”. This may cause a problem that the interrupt is canceled, or an unexpected interrupt is generated.

19.2.2 SP Setting

Set any value in the SP before accepting an interrupt. The SP is set to ‘0000_{16}’ after reset. Therefore, if an interrupt is accepted before setting any value in the SP, the program may go out of control.

19.2.3 External Interrupt and Key Input Interrupt

Either an “L” level or an “H” level of at least 250 ns width is necessary for the signal input to the $\overline{INT}_0$ to $\overline{INT}_3$ pins and KI_0 to KI_3 pins regardless of the CPU clock.

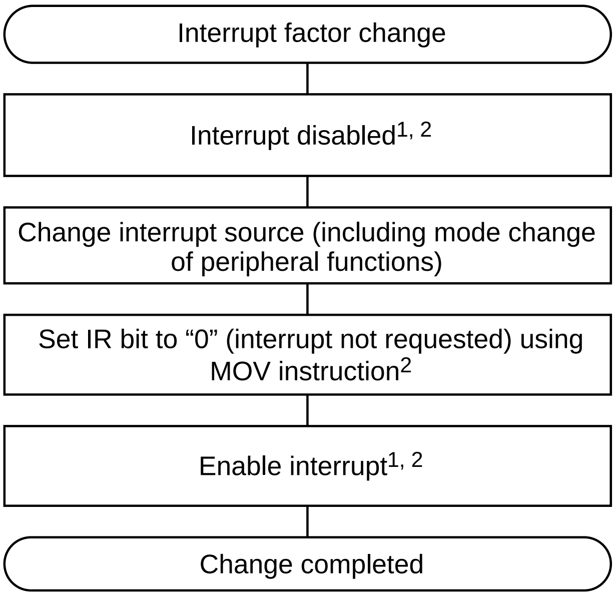
19.2.4 Watchdog Timer Interrupt

Initialize the watchdog timer after a watchdog timer interrupt occurs.

19.2.5 Changing Interrupt Source

The IR bit in the corresponding interrupt control register may be set to “1” (interrupt requested) when the interrupt source changes. When using an interrupt, the corresponding IR bit should be set to “0” (no interrupt requested) after changing the interrupt source.

In addition, the changes of interrupt sources said here include all factors that change the interrupt sources assigned to individual software interrupt numbers, polarities, and timing. Therefore, when a mode change in the peripheral functions etc. involves interrupt sources, edge polarities, and timing, the corresponding IR bit should be set to “0” (no interrupt requested) after the change. Refer to the description of each peripheral function for the interrupts caused by the peripheral functions. Figure 1.1 shows an example of the procedure for changing interrupt sources.


Interrupt factor change
Interrupt disabled[1, 2]
Change interrupt source (including mode change of peripheral functions)
Set IR bit to “0” (interrupt not requested) using MOV instruction[2]
Enable interrupt[1, 2]
Change completed

- IR bit: The corresponding interrupt control register bit of an interrupt by which an interrupt source is changed.
- The setting above should be performed individually. Avoid executing two or more settings at once (by one instruction).

Notes:

1. The I flag should be used for the INTi (i=0 to 3) interrupt. To prevent interrupt requests from being generated when using a peripheral function interrupt other than the INTi interrupt, the corresponding peripheral function should be disabled before changing the interrupt factor. In this case, the I flag should be used when all maskable interrupts can be disabled. When not all maskable interrupts can be disabled, the ILVL2 to ILVL0 bits of the corresponding interrupt should be used.
2. Refer the paragraph 1.2.6 “Changing Interrupt Control Register” for the instructions to be used and their usage notes.

Figure 19.1 Example of Procedure for Changing Interrupt Source

19.2.6 Changing Interrupt Control Register

(1) Each interrupt control register can only be modified while no interrupt requests corresponding to that register are generated. If interrupt requests managed by any interrupt control register are likely to occur, disable the interrupts before changing the interrupt control register.

(2) To modify any interrupt control register after disabling interrupts, be careful with the instructions used.

When Changing Other Than IR Bit

If an interrupt request corresponding to that register is generated while executing the instruction, the IR bit may not be set to “1” (interrupt requested), with the result that the interrupt request is ignored. If this presents a problem, use the following instructions to modify the register.
Instructions to use: AND, OR, BCLR, BSET

When Changing IR Bit

Even when the IR bit is cleared to “0” (interrupt not requested), it may not actually be cleared to “0” depending on the instruction used. Therefore, use the MOV instruction to set the IR bit to “0”.

(3) When disabling interrupts using the I flag, set the I flag according to the following sample programs. Refer to #2 for the change of interrupt control registers in the sample programs.

Sample programs 1 to 3 are to prevent the I flag from being set to “1” (interrupt enabled) before writing to the interrupt control registers for reasons of the internal bus or the instruction queue buffer.

Example 1: Use NOP instructions to prevent I flag being set to “1” before interrupt control register is changed

```
INT_SWITCH1:
    FCLR     I             ; Disable interrupts
    AND.B    #00H, 0056H   ; Set TXIC register to “0016”
    NOP
    NOP
    FSET     I             ; Enable interrupts
```

Example 2: Use dummy read to have FSET instruction wait

```
INT_SWITCH2:
    FCLR     I             ; Disable interrupts
    AND.B    #00H, 0056H   ; Set TXIC register to “0016”
    MOV.W    MEM, R0       ; Dummy read
    FSET     I             ; Enable interrupts
```

Example 3: Use POPC instruction to change I flag

```
INT_SWITCH3:
    PUSHC    FLG
    FCLR     I             ; Disable interrupts
    AND.B    #00H, 0056H   ; Set TXIC register to “0016”
    POPC     FLG           ; Enable interrupts
```

19.3 Timers

19.3.1 Timers X, Y and Z

(1) Timers X, Y and Z stop counting after reset. Therefore, a value must be set to these timers and prescalers before starting counting.

(2) Even if the prescalers and timers are read out simultaneously in 16-bit units, 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.

19.3.2 Timer X

(1) In pulse period measurement mode, the TXEDG bit and TXUND bit in the TXMR register can be set to “0” by writing “0” to these bits in a program. However, these bits remain unchanged when “1” is written. To set one flag to “0” in a program, write "1" to the other flag by using the MOV instruction. (This prevents any unintended changes of flag.)

Example (when setting TXEDG bit to “0”):

```
MOV.B       #10XXXXXXB,008BH
```

(2) When changing to pulse period measurement mode from other mode, the contents of the TXEDG bit and TXUND bit are indeterminate. Write "0" to the TXEDG bit and TXUND bit before starting counting.

19.3.3 Timer Z

In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TZS bit in the TC register to "0" and the timer reloads the value of reload register and stops. Therefore, the timer count value should be read out in programmable one-shot generation mode and programmable wait one-shot generation mode before the timer stops.

19.3.4 Timer C

(1) The TC register and TM0 register must be read in 16-bit units. This prevents the timer value from being updated during the period the high-byte and low-byte are being read.

Example (when Timer C is read):

```
MOV.W      0090H,R0    ; Read out timer C
```

19.4 Serial I/O

(1) When reading data from the UiRB (i=0,1) register even in the clock asynchronous serial I/O mode or in the clock synchronous serial I/O mode. Be sure to read data in 16-bit unit. When the high-byte of the UiRB register is read, the PER and FER bits of the UiRB register and the RI bit of the UiC1 register are set to "0".

Example (when reading receive buffer register):

```
    MOV.W      00A6H, R0           ; Read the U0RB register
```

(2) When writing data to the UiTB 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 in 8-bit unit.

Example (when reading transmit buffer register):

```
    MOV.B      #XXH, 00A3H     ; Write the high-byte of U0TB  register
    MOV.B      #XXH, 00A2H     ; Write the low-byte of U0TB register
```

19.5 A-D Converter

(1) When writing to each bit but except bit 6 in the ADON00 register, each bit in the ADCON1 register, or the SMP bit in the ADCON2 register, A/D conversion must be stopped (before a trigger occurs). When the VCUT bit in the ADCON1 register is changed from “0” (VREF not connected) to “1” (VREF connected), wait at least 1 µs before starting A/D conversion.

(2) When changing AD operation mode, select an analog input pin again.

(3) In one-shot mode, A/D conversion must be completed before reading the AD register. The IR bit in the ADIC register can indicates whether the A/D conversion is completed or not.

(4) In repeat mode, the undivided main clock must be used for the CPU.

(5) If A/D conversion is forcibly terminated while in progress by setting the ADST bit in the ADCON0 register to “0” (A/D conversion halted), the conversion result of the A/D converter is indeterminate. If the ADST bit is cleared to “0” in a program, ignore the value of AD register.

(6) A 0.1 µF capacitor should be connected between the AVcc/VREF pin and AVss pin.

19.6 Flash Memory Version

19.6.1 CPU Rewrite Mode

(1) Operation Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), select 5 MHz or less for CPU clock using the CM06 bit in the CM0 register and the CM16 to CM17 bits in the CM1 register.

(2) Instructions Inhibited Against Use

The following instructions cannot be used in EW0 mode because the flash memory's internal data is referenced: UND instruction, INTO instruction, and BRK instruction

(3) Interrupts

EW0 Mode

- Any interrupt which has a vector in the relocatable vector table can be used providing that its vector is transferred into the RAM space.
- The watchdog timer and oscillation stop detection interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.
  Because the rewrite operation is halted when a watchdog timer, oscillation stop detection or voltage detection interrupt occur, the rewrite program should be executed again after exiting the interrupt service routine.
- The address match interrupt cannot be used because the flash memory's internal data is referenced.

EW1 Mode

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program period or the auto erase period with erase-suspend function disabled.
- Avoid using watchdog timer interrupts.

(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 User ROM Space

In EW0 Mode, if the power supply voltage drops while rewriting any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, standard serial I/O or parallel I/O mode should be used.

(6) Wait Mode

When shifting to wait mode, set the FMR01 bit to "0" (CPU rewrite mode disabled) before executing the WAIT instruction.

(7) Stop Mode

When shifting to stop mode, the following settings are required:

- Set the FMR01 bit to “0” (CPU rewrite mode disabled) and disable DMA transfers before 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
```

(8) Ring Oscillator Low Power Dissipation Mode

If the CM05 bit is set to “1”, the following commands must not be executed.

- Program
- Block erase

19.7 Noise

(1) Bypass Capacitor between VCC and VSS Pins

Insert a bypass capacitor (at least 0.1 µF) between VCC and VSS pins as the countermeasures against noise and latch-up. The connecting wires must be 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 reset the port registers, port direction registers and pull-up control registers. However, you should fully examine before introducing the reset routine as conflicts may be created between this reset routine and interrupt routines (i. e. ports are switched during interrupts).

(3) CNVss Pin Wiring

In order to improve the pin tolerance to noise, insert a pull down resistance (about 5 k) between CNVss and Vss, and placed as close as possible to the CNVss pin.

20. Usage notes for on-chip debugger

When using the on-chip debugger to develop the R8C/10 group program and debug, pay the following attention.

(1) Avoid using $P0_0/AN_7/TxD_{11}$ pin and $P3_7/TxD_{10}/RxD_1$ pin.

(2) When write in the PD3 register ($00E7_{16}$ address), set bit 7 to "0".

(3) Avoid accessing the related serial I/O1 register.

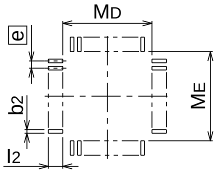
(4) Avoid using from $OC000_{16}$ address to $OC7FF_{16}$ address because the on-chip debugger uses these addresses.

Appendix 1. Package Dimensions

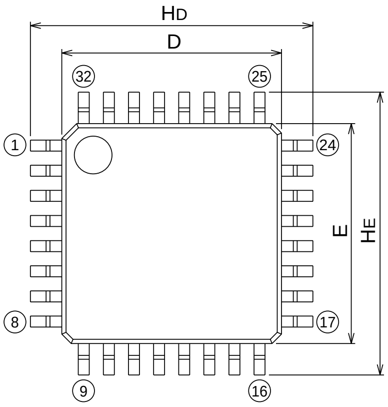
32P6U-A (MMP)

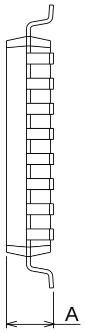
Plastic 32pin 7X7mm body LQFP

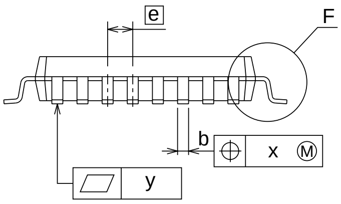
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP32-P-0707-0.80	–		Cu Alloy

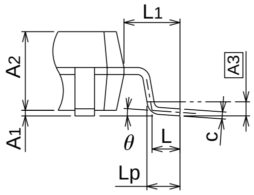


Recommended Mount Pad







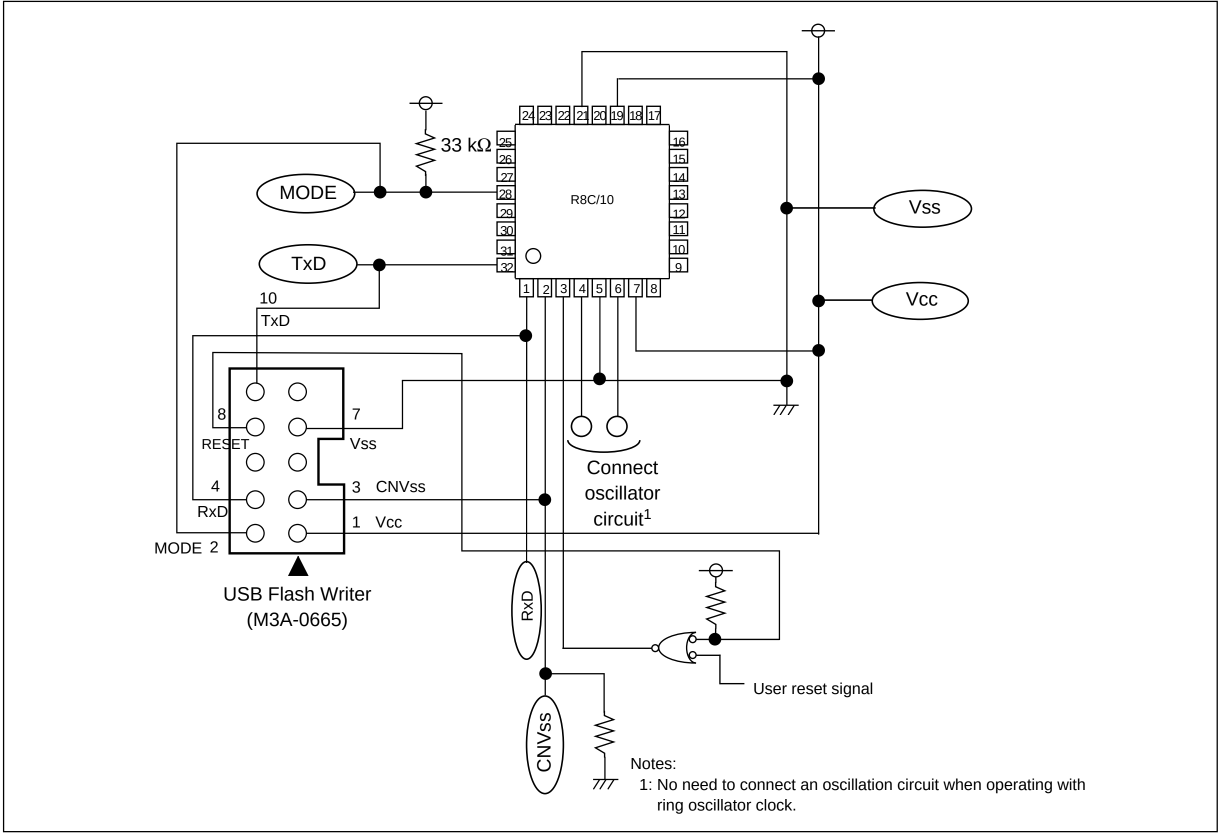


Detail F

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A1	0	0.1	0.2
A2	–	1.4	–
b	0.32	0.37	0.45
c	0.105	0.125	0.175
D	6.9	7.0	7.1
E	6.9	7.0	7.1
e	–	0.8	–
HD	8.8	9.0	9.2
HE	8.8	9.0	9.2
L	0.3	0.5	0.7
L1	–	1.0	–
Lp	0.45	0.6	0.75
A3	–	0.25	–
x	–	–	0.2
y	–	–	0.1
θ	0°	–	10°
b2	–	0.5	–
I2	1.0	–	–
MD	–	7.4	–
ME	–	7.4	–

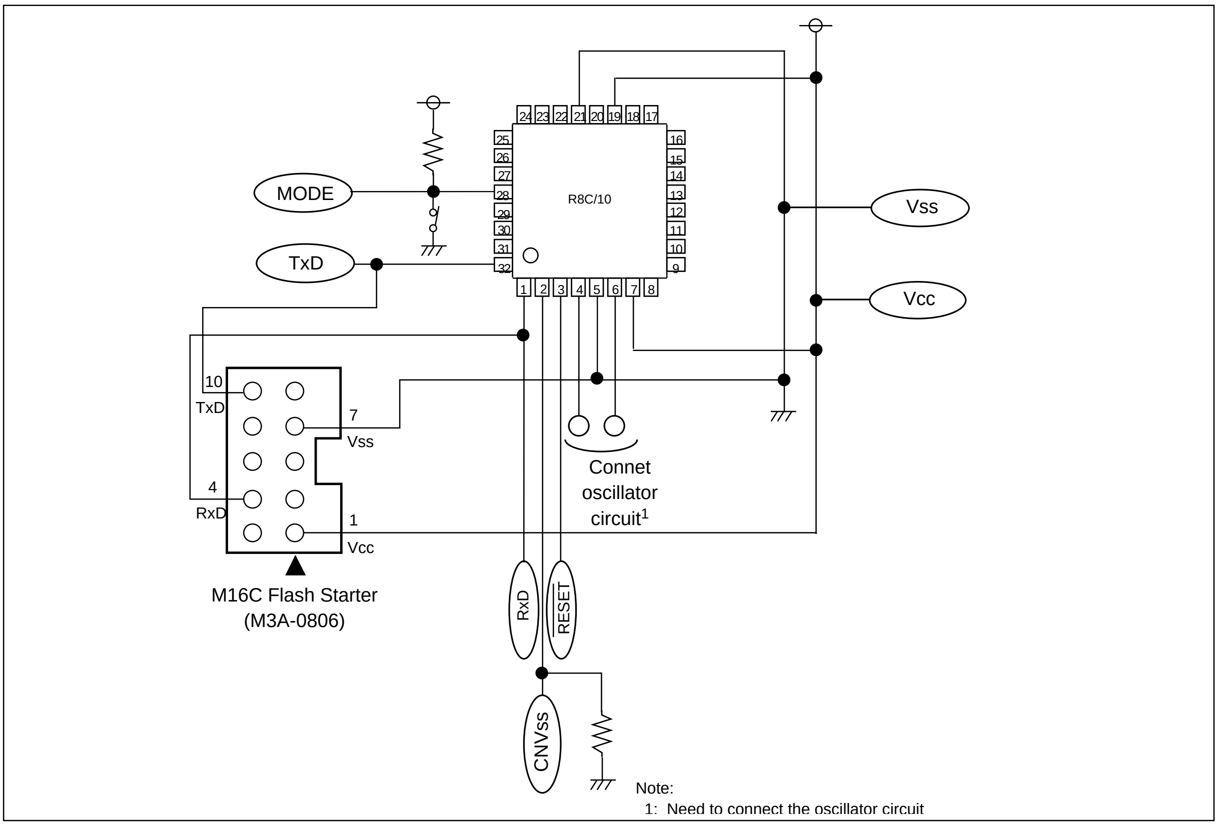
Appendix 2. Connecting examples for serial writer and on-chip debugging emulator

Appendix figure. 2.1 shows connecting examples with USB Flash Writer and appendix figure 2.2 shows connecting examples with M16C Flash Starter.



Notes:
1: No need to connect an oscillation circuit when operating with ring oscillator clock.

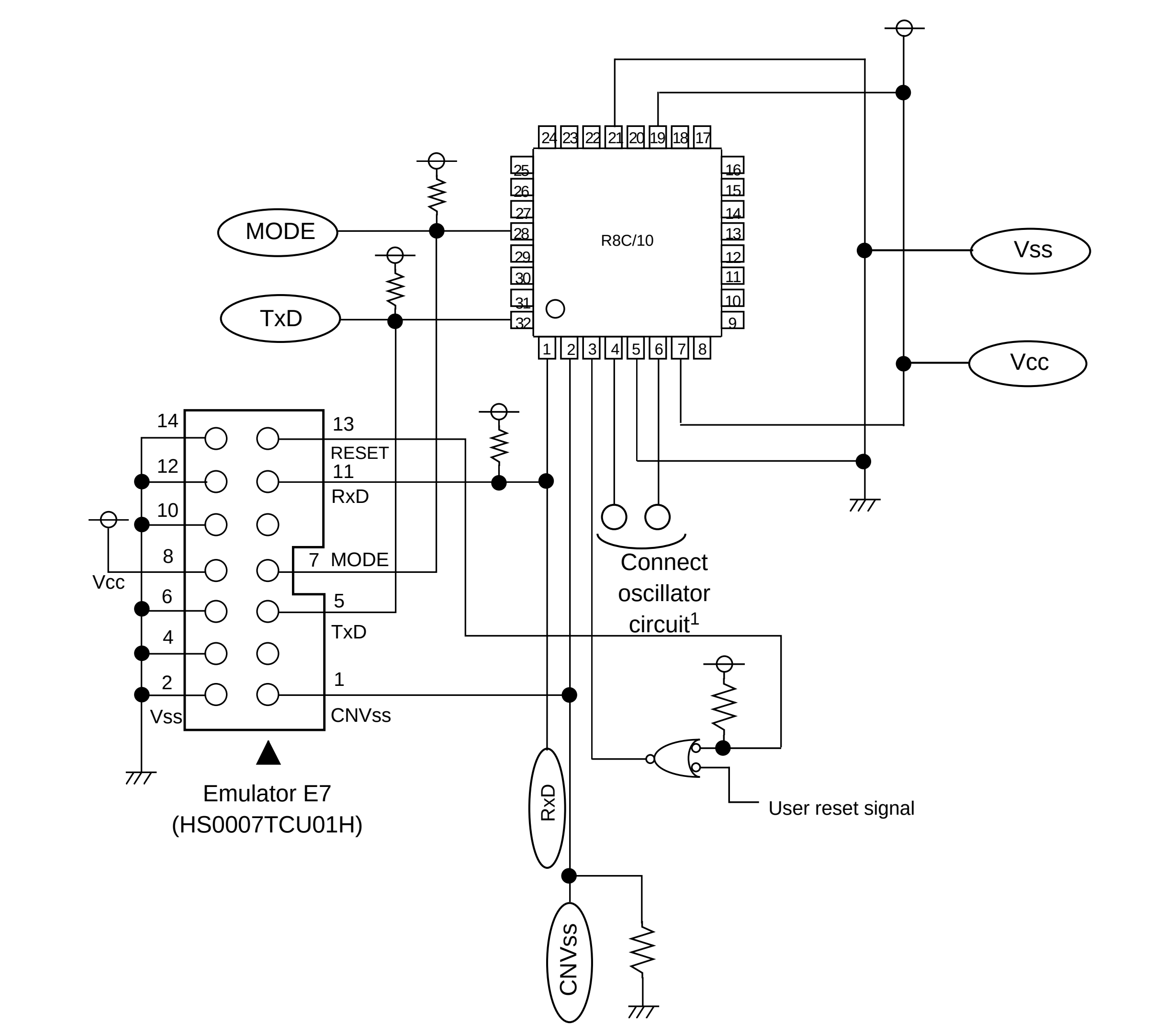
Appendix figure 2.1 Connecting examples with USB Flash Writer (M3A-0665)



Note:
1: Need to connect the oscillator circuit

Appendix figure 2.2 Connecting examples with M16C Flash Starter (M3A-0806)

Appendix figure 2.3 shows connecting examples with emulator E7.



Note:
1: No need to connect an oscillation circuit when operating with ring oscillator clock.

Figure 2.3 Connecting examples with emulator E7 (HS0007TCU01H)

Register Index

A

AD **107**
ADCON0 **106**, 108, 109
ADCON1 **106**, 108, 109
ADCON2 **107**
ADIC **40**
AIER **53**

C

CM0 **19**
CM1 **19**

D

DRR **117**

F

FMR0 **132**
FMR1 **133**
FMR4 **133**

I

INT0F **47**
INT0IC **40**
INT1IC **40**
INT2IC **40**
INT3IC **40**
INTEN **47**

K

KIEN **51**
KUPIC **40**

O

OCD **20**

P

P0 **116**
P1 **116**
P3 **116**
P4 **116**
PD0 **116**
PD1 **116**
PD3 **116**
PD4 **116**
PM0 **32**
PM1 **32**
PRCR **31**
PREX **58**
PREY **67**
PREZ **75**
PUM **68**, 70, 72, **76**, 78, 80, 82, 85
PUR0 **117**
PUR1 **117**

R

RMAD0 to RMAD1 **53**

S

S0RIC **40**
S0TIC **40**
S1RIC **40**
S1TIC **40**

T

TC **88**
TCC0 50, **88**
TCC1 50, **88**
TCIC **40**
TCSS **58**, 68, 76
TM0 **88**
TX **58**
TXIC **40**
TXMR 49, **57**, 59, 60, 61, 62, 64
TYIC **40**
TYPR **67**
TYSC **67**
TYZMR 49, **66**, 70, 72, **74**, 78, 80, 82, 85
TYZOC **67**, **75**
TZIC **40**
TZPR **75**
TZSC **75**

U

U0BRG **92**

U0C0 **93**

U0C1 **94**

U0MR **93**

U0RB **92**

U0TB **92**

U1BRG **92**

U1C0 **93**

U1C1 **94**

U1MR **93**

U1RB **92**

U1TB **92**

UCON **94**

W

WDC **55**

WDTR **55**

WDTS **55**

REVISION HISTORY	R8C/10 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.91	Sep 8, 2003	–	First edition issued

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER
HARDWARE MANUAL
R8C/10 Group

Publication Data : Rev.0.91 Sep 08, 2003
Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

R8C/10 Group
Hardware Manual

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