

16**R8C/10 Group**
Hardware Manual

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER
M16C FAMILY / R8C /Tiny SERIES

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

Keep safety first in your circuit designs!

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

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. 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 Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. 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 Corp. 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 Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. 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 Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination. Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

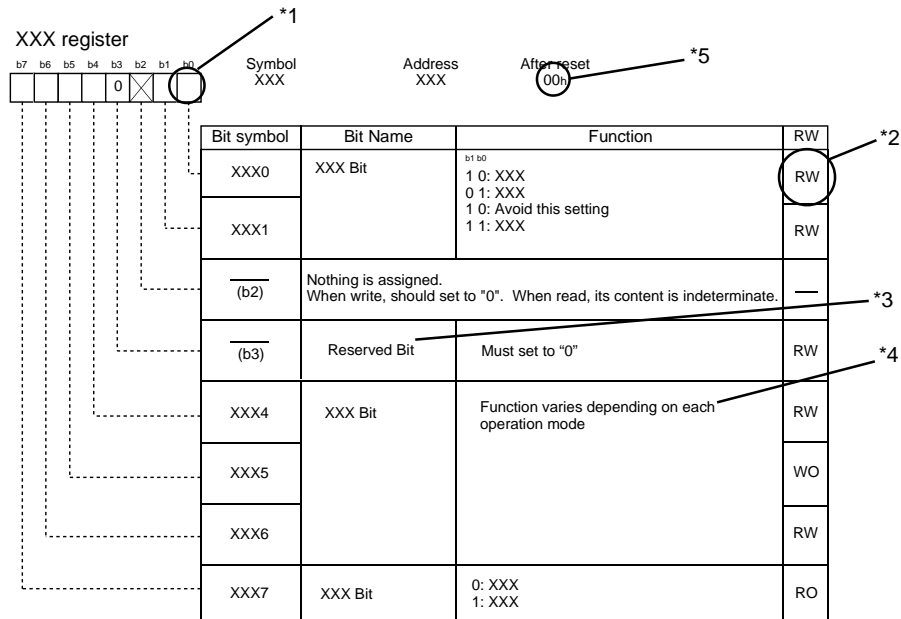
How to Use This Manual

1. Introduction

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

2. Register Diagram

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



- *1
 Blank: Set to "0" or "1" according to the application
 0: Set to "0"
 1: Set to "1"
 X: Nothing is assigned

- *2
 RW: Read and write
 RO: Read only
 WO: Write only
 -: Nothing is assigned

- *3
 •Reserved bit
 Reserved bit. Set to specified value.

- *4
 •Nothing is assigned
 Nothing is assigned to the bit concerned. As the bit may be use for future functions, set to "0" when writing to this bit.
 •Do not set to this value
 The operation is not guaranteed when a value is set.
 •Function varies depending on mode of operation
 Bit function varies depending on peripheral function mode.
 Refer to respective register for each mode.

- *5
 Follow the text in each manual for binary and hexadecimal notations.

3. M16C Family Documents

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

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

NOTES:

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

Table of Contents

SFR Page Reference

Chapter 1. Overview	1
1.1 Applications	1
1.2 Performance Overview	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)	8
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 Generation Circuit	17
6.1 Main Clock	21
6.2 On-Chip Oscillator Clock	22

6.3 CPU Clock and Peripheral Function Clock	23
6.3.1 CPU Clock	23
6.3.2 Peripheral Function Clock (f1, f2, f8, f32, fAD, f1SIO, f8SIO, f32SIO)	23
6.3.3 fRING and fRING128	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	28
6.5.1 How to Use Oscillation Stop Detection Function	28
Chapter 7. Protection	30
Chapter 8. Processor Mode	31
8.1 Types of Processor Mode	31
Chapter 9. Bus	32
Chapter 10. Interrupt	33
10.1 Interrupt Overview	33
10.1.1 Type of Interrupts	33
10.1.2 Software Interrupts	34
10.1.3 Hardware Interrupts	35
10.1.4 Interrupts and Interrupt Vector	36
10.1.5 Interrupt Control	38
10.2 INT Interrupt	46
10.2.1 INT0 Interrupt	46
10.2.2 INT0 Input Filter	47
10.2.3 INT1 Interrupt and INT2 Interrupt	48
10.2.4 INT3 Interrupt	49
10.3 Key Input Interrupt	50
10.4 Address Match Interrupt	51
Chapter 11. Watchdog Timer	53
Chapter 12. Timers	55
12.1 Timer X	56
12.1.1 Timer Mode	58
12.1.2 Pulse Output Mode	59
12.1.3 Event Counter Mode	60
12.1.4 Pulse Width Measurement Mode	61
12.1.5 Pulse Period Measurement Mode	63
12.2 Timer Y	65
12.2.1 Timer Mode	68
12.2.2 Programmable Waveform Generation Mode	70
12.3 Timer Z	73
12.3.1 Timer Mode	76
12.3.2 Programmable Waveform Generation Mode	78

12.3.3 Programmable One-shot Generation Mode	80
12.3.4 Programmable Wait One-shot Generation Mode	83
12.4 Timer C	86
Chapter 13. Serial Interface	89
13.1 Clock Synchronous Serial I/O Mode	94
13.1.1 Polarity Select Function	97
13.1.2 LSB First/MSB First Select Function	97
13.1.3 Continuous Receive Mode	98
13.2 Clock Asynchronous Serial I/O (UART) Mode	99
13.2.1 TxD10/RxD1 Select Function (UART1)	102
13.2.2 TxD11 Select Function (UART1)	102
13.2.3 Bit Rate	103
Chapter 14. A/D Converter	104
14.1 One-shot Mode	108
14.2 Repeat Mode	109
14.3 Sample & Hold	110
14.4 A/D conversion cycles	110
14.5 Internal Equivalent Circuit of Analog Input	111
14.6 Inflow Current Bypass Circuit	112
14.7 Output Impedance of Sensor under A/D Conversion	113
Chapter 15. Programmable I/O Ports	115
15.1 Description	115
15.1.1 Port Pi Direction Register (PDi Register, i = 0, 1, 3, 4)	115
15.1.2 Port Pi Register (Pi Register, i = 0 to 4)	115
15.1.3 Pull-up Control Register 0, Pull-up Control Register 1 (PUR0 and PUR1 Registers)	115
15.1.4 Port P1 Drive Capacity Control Register (DRR Register)	115
15.2 Port setting	122
15.3 Unassigned Pin Handling	128
Chapter 16. Electrical Characteristics	129
Chapter 17. Flash Memory Version	140
17.1 Overview	140
17.2 Memory Map	141
17.3 Functions To Prevent Flash Memory from Rewriting	142
17.3.1 ID Code Check Function	142
17.4 CPU Rewrite Mode	143
17.4.1 EW0 Mode	144
17.4.2 EW1 Mode	144
17.4.3 Software Commands	150
17.4.4 Status Register	154
17.4.5 Full Status Check	155
17.5 Standard Serial I/O Mode	157
17.5.1 ID Code Check Function	157

Chapter 18. On-chip Debugger	161
18.1 Address Match Interrupt	161
18.2 Single Step Interrupt	161
18.3 UART1	161
18.4 BRK Instruction	161
Chapter 19. Usage Notes	162
19.1 Stop Mode and Wait Mode	162
19.1.1 Stop Mode	162
19.1.2 Wait Mode	162
19.2 Interrupts	163
19.2.1 Reading Address 00000 ₁₆	163
19.2.2 SP Setting	163
19.2.3 External Interrupt and Key Input Interrupt	163
19.2.4 Watchdog Timer Interrupt	163
19.2.5 Changing Interrupt Factor	164
19.2.6 Changing Interrupt Control Register	165
19.3 Clock Generation Circuit	166
19.3.1 Oscillation Stop Detection Function	166
19.3.2 Oscillation Circuit Constants	166
19.4 Timers	167
19.3.1 Timers X, Y and Z	167
19.3.2 Timer X	167
19.3.3 Timer Y	167
19.3.4 Timer Z	167
19.3.5 Timer C	167
19.5 Serial Interface	168
19.6 A/D Converter	169
19.7 Flash Memory Version	170
19.7.1 CPU Rewrite Mode	170
19.8 Noise	173
Chapter 20. Usage Notes for On-chip Debugger	174
Appendix 1 Package Dimensions	175
Appendix 2 Connecting Examples for Serial Writer and On-chip Debugging Emulator	176
Appendix 3 Example of Oscillation Evaluation Circuit..	178
Register Index	179

SFR Page Reference

Address	Register	Symbol	Page
0000 ₁₆			
0001 ₁₆			
0002 ₁₆			
0003 ₁₆			
0004 ₁₆	Processor mode register 0	PM0	31
0005 ₁₆	Processor mode register 1	PM1	31
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	52
000A ₁₆	Protect register	PRCR	30
000B ₁₆			
000C ₁₆	Oscillation stop detection register	OCD	20
000D ₁₆	Watchdog timer reset register	WDTR	54
000E ₁₆	Watchdog timer start register	WDTS	54
000F ₁₆	Watchdog timer control register	WDC	54
0010 ₁₆	Address match interrupt register 0	RMAD0	52
0011 ₁₆			
0012 ₁₆			
0013 ₁₆			
0014 ₁₆	Address match interrupt register 1	RMAD1	52
0015 ₁₆			
0016 ₁₆			
0017 ₁₆			
0018 ₁₆			
0019 ₁₆			
001A ₁₆			
001B ₁₆			
001C ₁₆			
001D ₁₆			
001E ₁₆	INT0 input filter select register	INTOF	46
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	39
004E ₁₆	AD conversion interrupt control register	ADIC	39
004F ₁₆			
0050 ₁₆			
0051 ₁₆	UART0 transmit interrupt control register	S0TIC	39
0052 ₁₆	UART0 receive interrupt control register	S0RIC	39
0053 ₁₆	UART1 transmit interrupt control register	S1TIC	39
0054 ₁₆	UART1 receive interrupt control register	S1RIC	39
0055 ₁₆	INT2 interrupt control register	INT2IC	39
0056 ₁₆	Timer X interrupt control register	TXIC	39
0057 ₁₆	Timer Y interrupt control register	TYIC	39
0058 ₁₆	Timer Z interrupt control register	TZIC	39
0059 ₁₆	INT1 interrupt control register	INT1IC	39
005A ₁₆	INT3 interrupt control register	INT3IC	39
005B ₁₆	Timer C interrupt control register	TCIC	39
005C ₁₆			
005D ₁₆	INT0 interrupt control register	INT0IC	39
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.

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.

1.2 Performance Overview

Table 1.1. lists the performance outline of this MCU.

Table 1.1 Performance outline

Item		Performance
CPU	Number of basic instructions	89 instructions
	Minimum 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	Port	Input/Output: 22 (including LED drive port), Input: 2
	LED drive port	I/O port: 8
	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 interface	•1 channel Clock synchronous, UART •1 channel UART
	A/D converter	10-bit A/D converter: 1 circuit, 8 channels
	Watchdog timer	15 bits x 1 (with prescaler)
	Interrupt	Internal: 9 factors, External: 5 factors, Software: 4 factors, Priority level: 7 levels
	Clock generation circuit	2 circuits •Main clock generation circuit (Equipped with a built-in feedback resistor) •On-chip oscillator
	Oscillation stop detection function	Main clock oscillation stop detection function
	Electrical characteristics	Supply voltage
Power consumption		Typ. 8mA ($VCC = 5.0$ V, ($f(XIN) = 16$ MHz) Typ. 5mA ($VCC = 3.0$ V, ($f(XIN) = 10$ MHz) Typ. 35 μ A ($VCC = 3.0$ V, Wait mode, Peripheral clock off) Typ. 0.7 μ A ($VCC = 3.0$ V, Stop mode)
Flash memory	Program/erase supply voltage	$VCC = 2.7$ to 5.5 V
	Program/erase endurance	100 times
Operating ambient temperature		-20 to 85 °C -40 to 85 °C (D-version)
Package		32-pin plastic mold LQFP

1.3 Block Diagram

Figure 1.1 shows this MCU block diagram.

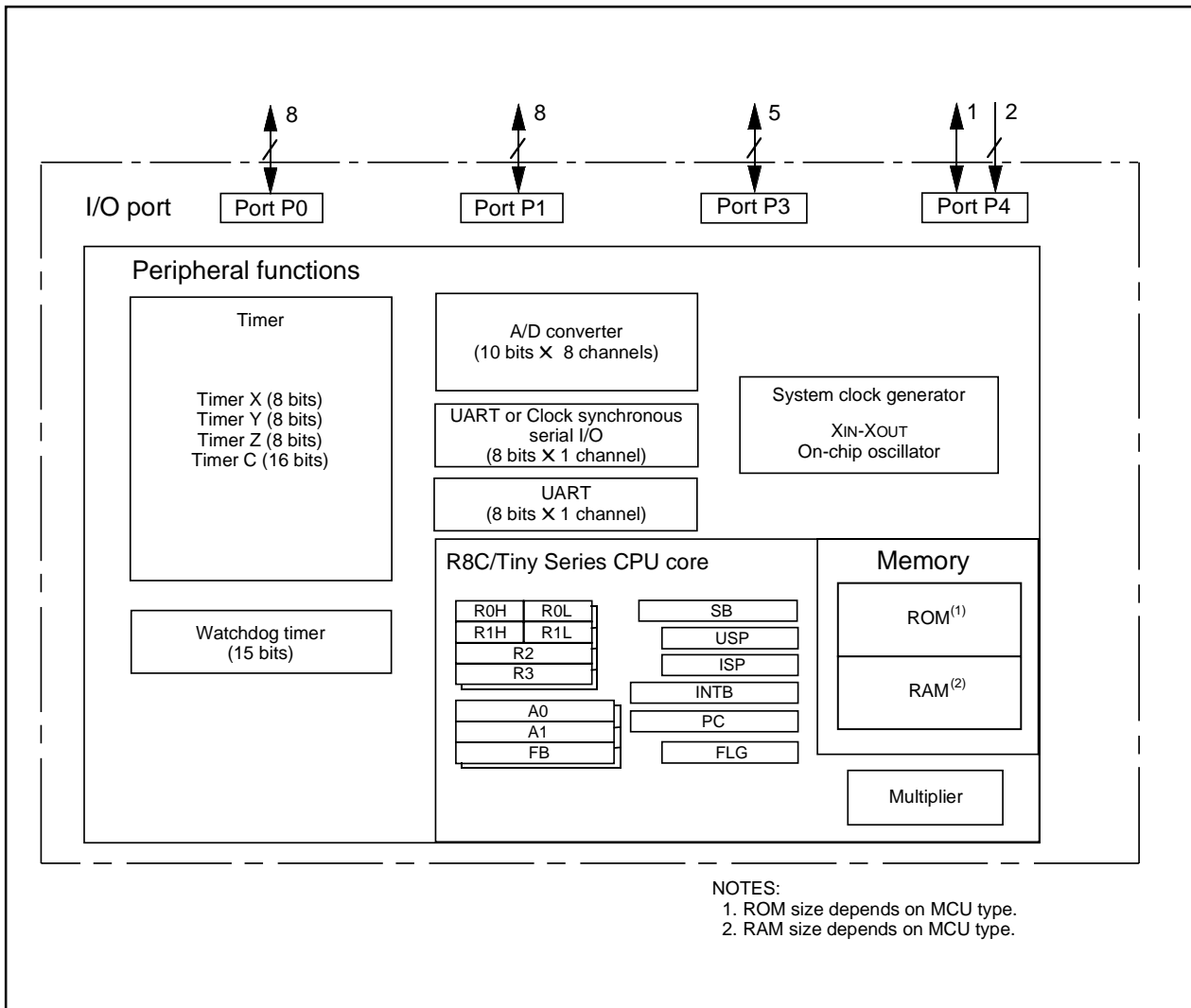


Figure 1.1 Block Diagram

1.4 Product Information

Table 1.2 lists the product information.

Table 1.2 Product Information

As of January 2006

Type No.	ROM capacity	RAM capacity	Package type	Remarks
R5F21102FP	8K bytes	512 bytes	PLQP0032GB-A	Flash memory version
R5F21103FP	12K bytes	768 bytes	PLQP0032GB-A	
R5F21104FP	16K bytes	1K bytes	PLQP0032GB-A	
R5F21102DFP	8K bytes	512 bytes	PLQP0032GB-A	D version
R5F21103DFP	12K bytes	768 bytes	PLQP0032GB-A	
R5F21104DFP	16K bytes	1K bytes	PLQP0032GB-A	

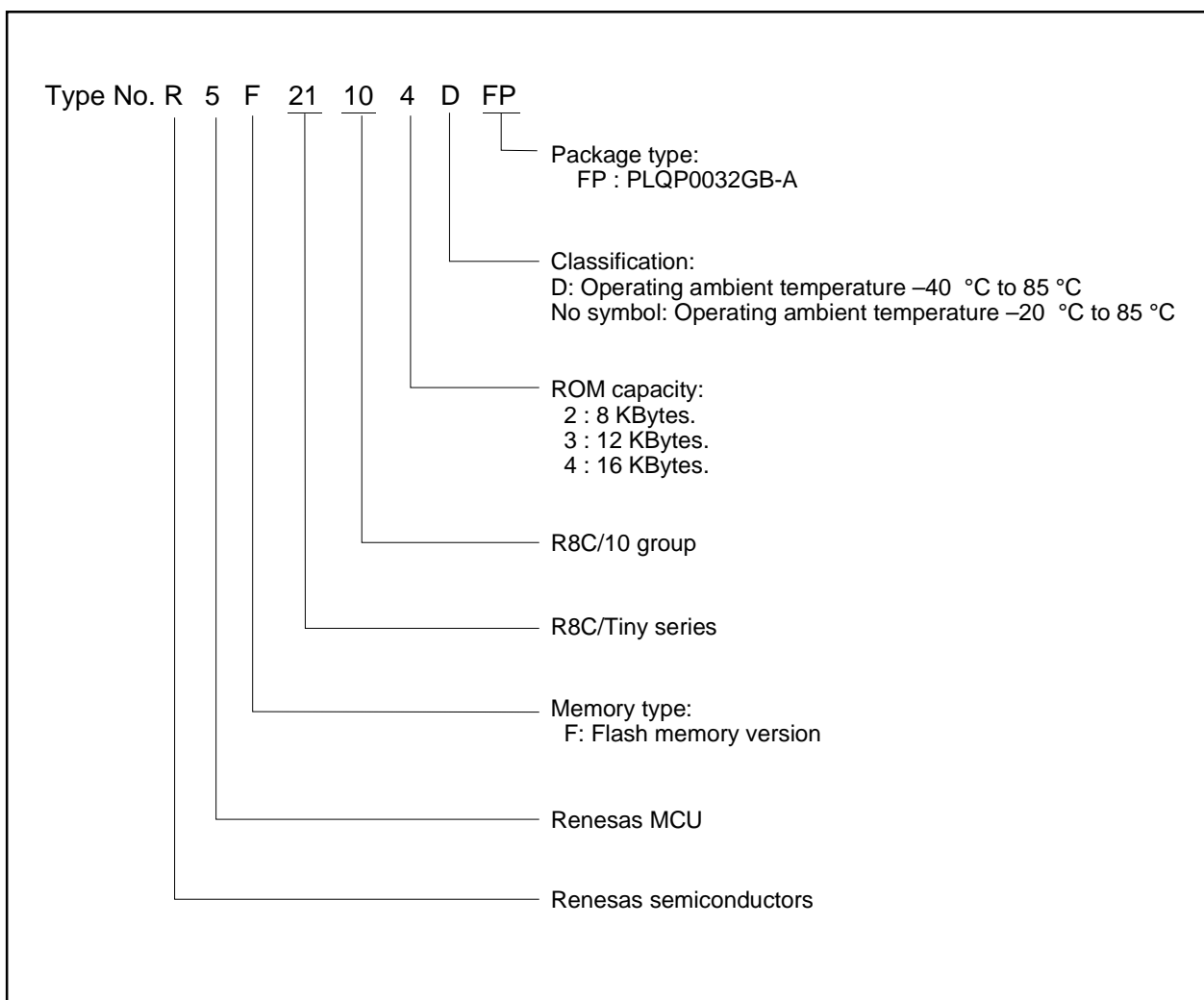
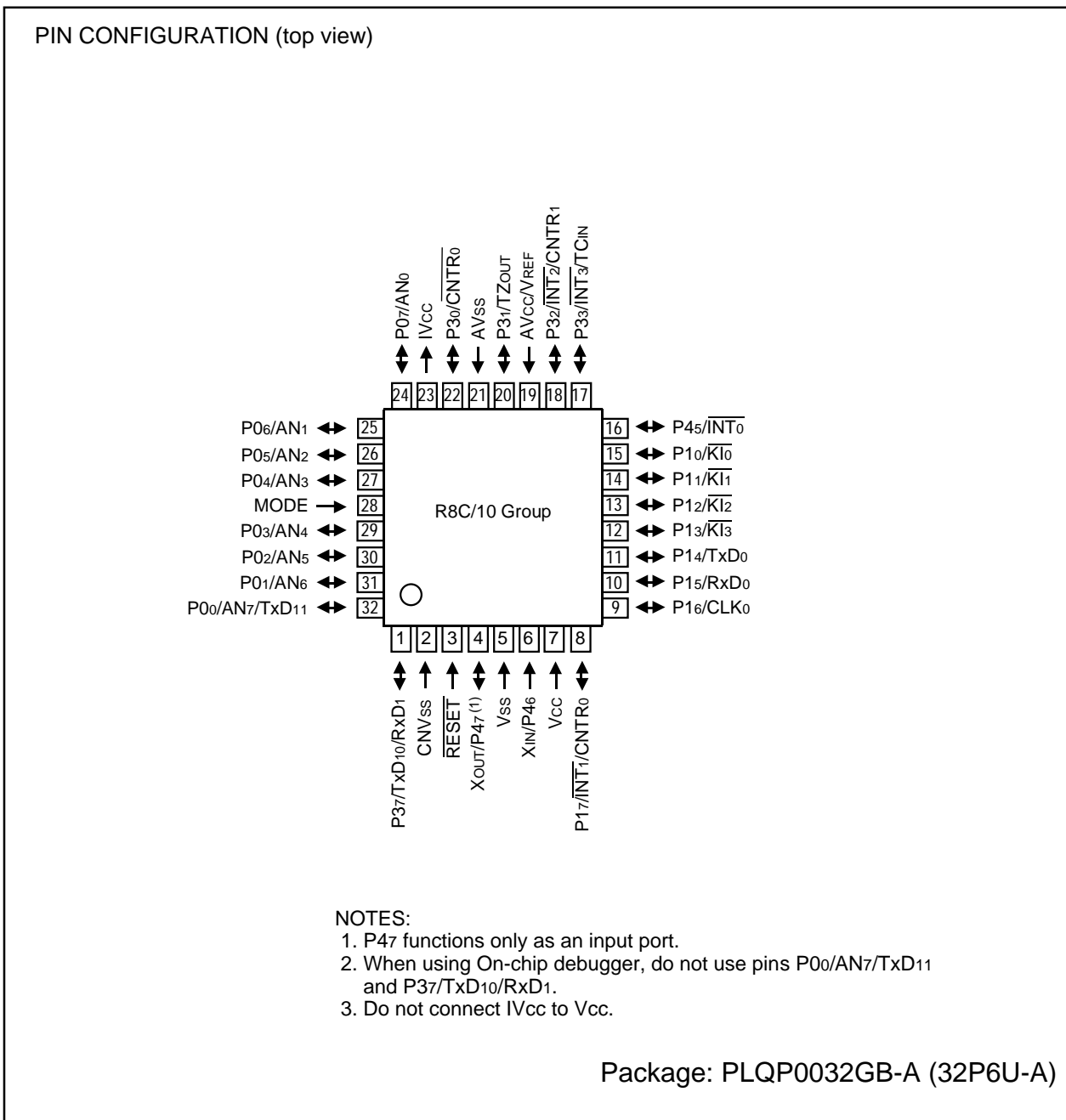


Figure 1.2 Type No., Memory Size, and Package

1.5 Pin Assignment

Figure 1.3 shows the pin Assignments (top view).



NOTES:

1. P47 functions only as an input port.
2. When using On-chip debugger, do not use pins P00/AN7/TxD11 and P37/TxD10/RxD1.
3. Do not connect IVcc to Vcc.

Figure 1.3 Pin Assignments (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	I	Apply 2.7 V to 5.5 V to the Vcc pin. Apply 0 V to the Vss pin.
IVcc	IVcc	O	This pin is to stabilize internal power supply. Connect this pin to Vss via a capacitor (0.1 μF). Do not connect to Vcc.
Analog power supply input	AVcc, AVss	I	Power supply input pins for A/D converter. Connect the AVcc pin to Vcc. Connect the AVss pin to Vss. Connect a capacitor between pins AVcc and AVss.
Reset input	RESET	I	Input "L" on this pin resets the MCU.
CNVss	CNVss	I	Connect this pin to Vss via a resistor. ⁽¹⁾
MODE	MODE	I	Connect this pin to Vcc via a resistor.
Main clock input	XIN	I	These pins are provided for the main clock generating circuit I/O. 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	O	
INT interrupt input	INT ₀ to INT ₃	I	INT interrupt input pins.
Key input interrupt	KI ₀ to KI ₃	I	Key input interrupt pins.
Timer X	CNTR ₀	I/O	Timer X I/O pin
	CNTR ₀	O	Timer X output pin
Timer Y	CNTR ₁	I/O	Timer Y I/O pin
Timer Z	TZOUT	O	Timer Z output pin
Timer C	TCIN	I	Timer C input pin
Serial interface	CLK ₀	I/O	Transfer clock I/O pin.
	RxD ₀ , RxD ₁	I	Serial data input pins.
	TxD ₀ , TxD ₁₀ , TxD ₁₁	O	Serial data output pins.
Reference voltage input	VREF	I	Reference voltage input pin for A/D converter. Connect the VREF pin to Vcc.
A/D converter	AN ₀ to AN ₇	I	Analog input pins for A/D converter
I/O port	P ₀₀ to P ₀₇ , P ₁₀ to P ₁₇ , P ₃₀ to P ₃₃ , P ₃₇ , P ₄₅	I/O	These are 8-bit CMOS I/O ports. Each port has an I/O 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. P ₁₀ to P ₁₇ also function as LED drive ports.
Input port	P ₄₆ , P ₄₇	I	Port for input-only.

NOTES :

1. Refer to "19.8 Noise" for the connecting reference resistor value.

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. Two sets of register banks are provided.

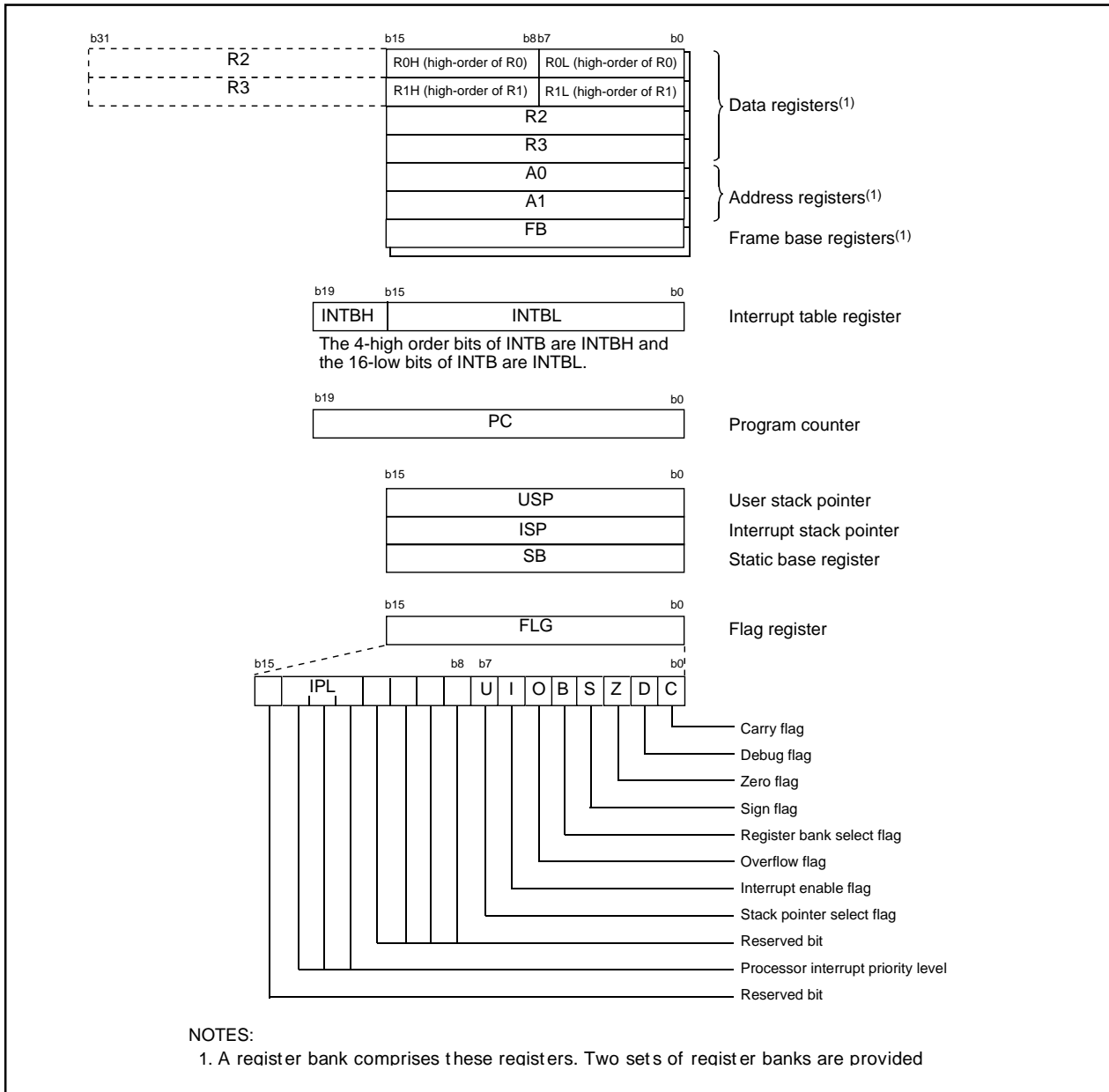


Figure 2.1 CPU Register

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

R0 is a 16-bit register for transfer, arithmetic and logic operations. The same applies to R1 to R3. The R0 can be split into high-order bit (R0H) and low-order bit (R0L) to be used separately as 8-bit data registers. The same applies to R1H and R1L as R0H and R0L. R2 can be combined with R0 to be used as a 32-bit data register (R2R0). The same applies to R3R1 as R2R0.

2.2 Address Registers (A0 and A1)

A0 is a 16-bit register for address register indirect addressing and address register relative addressing. They also are used for transfer, arithmetic and logic operations. The same applies to A1 as A0. A0 can be combined with A1 to be used as a 32-bit address register (A1A0).

2.3 Frame Base Register (FB)

FB is a 16-bit register for FB relative addressing.

2.4 Interrupt Table Register (INTB)

INTB is a 20-bit register indicates the start address of an interrupt vector table.

2.5 Program Counter (PC)

PC, 20 bits wide, indicates the address of an instruction to be executed.

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

The stack pointer (SP), USP and ISP, are 16 bits wide each. The U flag of FLG is used to switch between USP and ISP.

2.7 Static Base Register (SB)

SB is a 16-bit register for SB relative addressing.

2.8 Flag Register (FLG)

FLG is a 11-bit register indicating the CPU state.

2.8.1 Carry Flag (C)

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

2.8.2 Debug Flag (D)

The D flag is for debug only. Set to "0".

2.8.3 Zero Flag (Z)

The Z flag is set to "1" when an arithmetic operation resulted in 0; otherwise, "0".

2.8.4 Sign Flag (S)

The S flag is set to "1" when an arithmetic operation resulted in a negative value; otherwise, "0".

2.8.5 Register Bank Select Flag (B)

The register bank 0 is selected when the B flag is "0". The register bank 1 is selected when this flag is set to "1".

2.8.6 Overflow Flag (O)

The O flag is set to "1" when the operation resulted in an overflow; otherwise, "0".

2.8.7 Interrupt Enable Flag (I)

The I flag enables a maskable interrupt.

An interrupt is disabled when the I flag is set to "0", and are enabled when the I flag is set to "1". The I flag is set to "0" when an interrupt request is acknowledged.

2.8.8 Stack Pointer Select Flag (U)

ISP is selected when the U flag is set to "0", USP is selected when the U flag is set to "1".

The U flag is set to "0" when a hardware interrupt request is acknowledged or the INT instruction of software interrupt numbers 0 to 31 is executed.

2.8.9 Processor Interrupt Priority Level (IPL)

IPL, 3 bits wide, assigns processor interrupt priority levels from level 0 to level 7.

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

2.8.10 Reserved Bit

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

3. Memory

Figure 3.1 is a memory map of this MCU. This MCU provides 1-Mbyte address space from addresses 00000₁₆ to FFFFF₁₆.

The internal ROM is allocated lower addresses beginning with address 0FFFF₁₆. For example, a 16-Kbyte internal ROM is allocated addresses from 0C000₁₆ to 0FFFF₁₆.

The fixed interrupt vector table is allocated addresses 0FFDC₁₆ to 0FFFF₁₆. They store the starting address of each interrupt routine.

The internal RAM is allocated higher addresses beginning with address 00400₁₆. For example, a 1-Kbyte internal RAM is allocated addresses 00400₁₆ to 007FF₁₆. The internal RAM is used not only for storing data, but for calling subroutines and stacks when interrupt request is acknowledged.

Special function registers (SFR) are allocated addresses 00000₁₆ to 002FF₁₆. The peripheral function control registers are located there. All addresses, which have nothing allocated within the SFR, are reserved area and cannot be accessed by users.

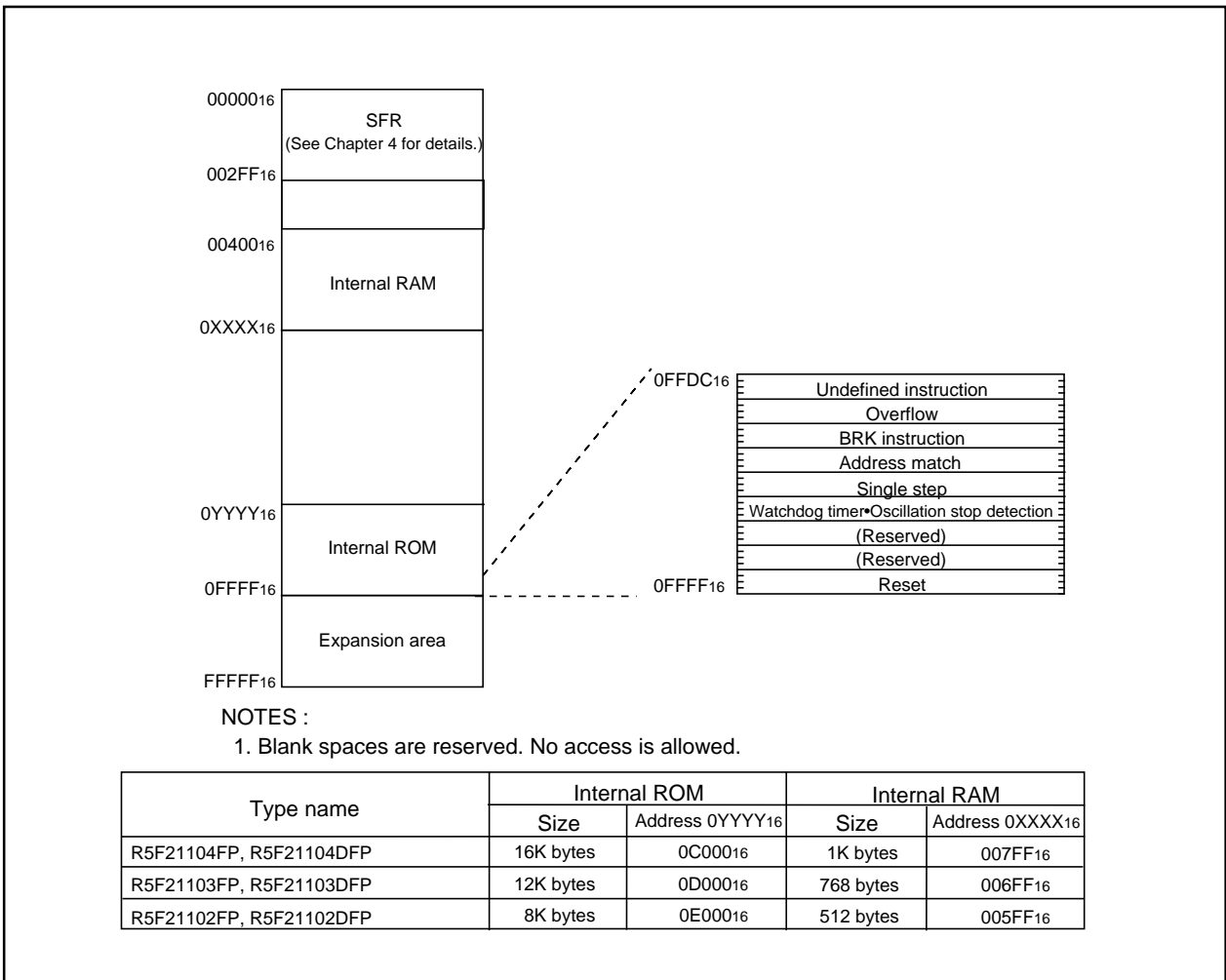


Figure 3.1 Memory Map

4. Special Function Register (SFR)

SFR(Special Function Register) is the control register of peripheral functions. Tables 4.1 to 4.4 list the SFR information

Table 4.1 SFR Information(1)(1)

Address	Register	Symbol	After reset
0000 ₁₆			
0001 ₁₆			
0002 ₁₆			
0003 ₁₆			
0004 ₁₆	Processor mode register 0	PM0	XXXX0X002
0005 ₁₆	Processor mode register 1	PM1	00XXX0X02
0006 ₁₆	System clock control register 0	CM0	011010002
0007 ₁₆	System clock control register 1	CM1	001000002
0008 ₁₆			
0009 ₁₆	Address match interrupt enable register	AIER	XXXXXX002
000A ₁₆	Protect register	PRCR	00XXX0002
000B ₁₆			
000C ₁₆	Oscillation stop detection register	OCD	000001002
000D ₁₆	Watchdog timer reset register	WDTR	XX16
000E ₁₆	Watchdog timer start register	WDTS	XX16
000F ₁₆	Watchdog timer control register	WDC	000111112
0010 ₁₆	Address match interrupt register 0	RMAD0	0016
0011 ₁₆			0016
0012 ₁₆			X016
0013 ₁₆			
0014 ₁₆	Address match interrupt register 1	RMAD1	0016
0015 ₁₆			0016
0016 ₁₆			X016
0017 ₁₆			
0018 ₁₆			
0019 ₁₆			
001A ₁₆			
001B ₁₆			
001C ₁₆			
001D ₁₆			
001E ₁₆	INT0 input filter select register	INT0F	XXXXX0002
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			

NOTES:

- Blank spaces are reserved. No access is allowed.

X : Undefined

Table 4.2 SFR Information(2)⁽¹⁾

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	XXXXX0002
004E ₁₆	AD conversion interrupt control register	ADIC	XXXXX0002
004F ₁₆			
0050 ₁₆			
0051 ₁₆	UART0 transmit interrupt control register	S0TIC	XXXXX0002
0052 ₁₆	UART0 receive interrupt control register	S0RIC	XXXXX0002
0053 ₁₆	UART1 transmit interrupt control register	S1TIC	XXXXX0002
0054 ₁₆	UART1 receive interrupt control register	S1RIC	XXXXX0002
0055 ₁₆	INT2 interrupt control register	INT2IC	XXXXX0002
0056 ₁₆	Timer X interrupt control register	TXIC	XXXXX0002
0057 ₁₆	Timer Y interrupt control register	TYIC	XXXXX0002
0058 ₁₆	Timer Z interrupt control register	TZIC	XXXXX0002
0059 ₁₆	INT1 interrupt control register	INT1IC	XXXXX0002
005A ₁₆	INT3 interrupt control register	INT3IC	XXXXX0002
005B ₁₆	Timer C interrupt control register	TCIC	XXXXX0002
005C ₁₆			
005D ₁₆	INT0 interrupt control register	INT0IC	XX00X0002
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			

NOTES :

- Blank spaces are reserved. No access is allowed.

X : Undefined

Table 4.3 SFR Information(3)⁽¹⁾

Address	Register	Symbol	After reset
0080 ₁₆	Timer Y, Z mode register	TYZMR	0016
0081 ₁₆	Prescaler Y register	PREY	FF16
0082 ₁₆	Timer Y secondary register	TYSC	FF16
0083 ₁₆	Timer Y primary register	TYPR	FF16
0084 ₁₆	Timer Y, Z waveform output control register	PUM	0016
0085 ₁₆	Prescaler Z register	PREZ	FF16
0086 ₁₆	Timer Z secondary register	TZSC	FF16
0087 ₁₆	Timer Z primary register	TZPR	FF16
0088 ₁₆			
0089 ₁₆			
008A ₁₆	Timer Y, Z output control register	TYZOC	0016
008B ₁₆	Timer X mode register	TXMR	0016
008C ₁₆	Prescaler X register	PREX	FF16
008D ₁₆	Timer X register	TX	FF16
008E ₁₆	Count source set register	TCSS	0016
008F ₁₆			
0090 ₁₆	Timer C register	TC	0016
0091 ₁₆			0016
0092 ₁₆			
0093 ₁₆			
0094 ₁₆			
0095 ₁₆			
0096 ₁₆	External input enable register	INTEN	0016
0097 ₁₆			
0098 ₁₆	Key input enable register	KIEN	0016
0099 ₁₆			
009A ₁₆	Timer C control register 0	TCC0	0016
009B ₁₆	Timer C control register 1	TCC1	0016
009C ₁₆	Capture register	TM0	0016
009D ₁₆			0016
009E ₁₆			
009F ₁₆			
00A0 ₁₆	UART0 transmit/receive mode register	U0MR	0016
00A1 ₁₆	UART0 bit rate generator	U0BRG	XX16
00A2 ₁₆	UART0 transmit buffer register	U0TB	XX16
00A3 ₁₆			XX16
00A4 ₁₆	UART0 transmit/receive control register 0	U0C0	000010002
00A5 ₁₆	UART0 transmit/receive control register 1	U0C1	000000102
00A6 ₁₆	UART0 receive buffer register	U0RB	XX16
00A7 ₁₆			XX16
00A8 ₁₆	UART1 transmit/receive mode register	U1MR	0016
00A9 ₁₆	UART1 bit rate generator	U1BRG	XX16
00AA ₁₆	UART1 transmit buffer register	U1TB	XX16
00AB ₁₆			XX16
00AC ₁₆	UART1 transmit/receive control register 0	U1C0	000010002
00AD ₁₆	UART1 transmit/receive control register 1	U1C1	000000102
00AE ₁₆	UART1 receive buffer register	U1RB	XX16
00AF ₁₆			XX16
00B0 ₁₆	UART transmit/receive control register 2	UCON	0016
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			

NOTES:

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

X : Undefined

Table 4.4 SFR Information(4)⁽¹⁾

Address	Register	Symbol	After reset
00C0 ₁₆	AD 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	AD control register 2	ADCON2	0016
00D5 ₁₆			
00D6 ₁₆	AD control register 0	ADCON0	00000XXX ₂
00D7 ₁₆	AD control register 1	ADCON1	0016
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	0016
00E3 ₁₆	Port P1 direction register	PD1	0016
00E4 ₁₆			
00E5 ₁₆	Port P3 register	P3	XX ₁₆
00E6 ₁₆			
00E7 ₁₆	Port P3 direction register	PD3	0016
00E8 ₁₆	Port P4 register	P4	XX ₁₆
00E9 ₁₆			
00EA ₁₆	Port P4 direction register	PD4	0016
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	XXXXXXXX ₂
00FE ₁₆	Port P1 drive capacity control register	DRR	0016
00FF ₁₆			
01B3 ₁₆	Flash memory control register 4	FMR4	01000000 ₂
01B4 ₁₆			
01B5 ₁₆	Flash memory control register 1	FMR1	0100XX0X ₂
01B6 ₁₆			
01B7 ₁₆	Flash memory control register 0	FMR0	00000012

NOTES:

1. Blank columns, 0100₁₆ to 01B2₁₆ and 01B8₁₆ to 02FF₁₆ are all reserved. No access is allowed.

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. After reset, the on-chip oscillator clock divided by 8 is automatically selected for the CPU. 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 for 500 μs ($1/\text{fRING} \times 20$).
 - (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 $t_d(\text{P-R})$ or more until the internal power supply stabilizes.
 - (4) Wait for 500 μs ($1/\text{fRING} \times 20$).
 - (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
	CNV _{SS} = V _{SS}
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. After reset, the on-chip oscillator clock divided by 8 is automatically selected for the CPU.

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. After reset, the on-chip oscillator clock divided by 8 is automatically selected for the CPU.

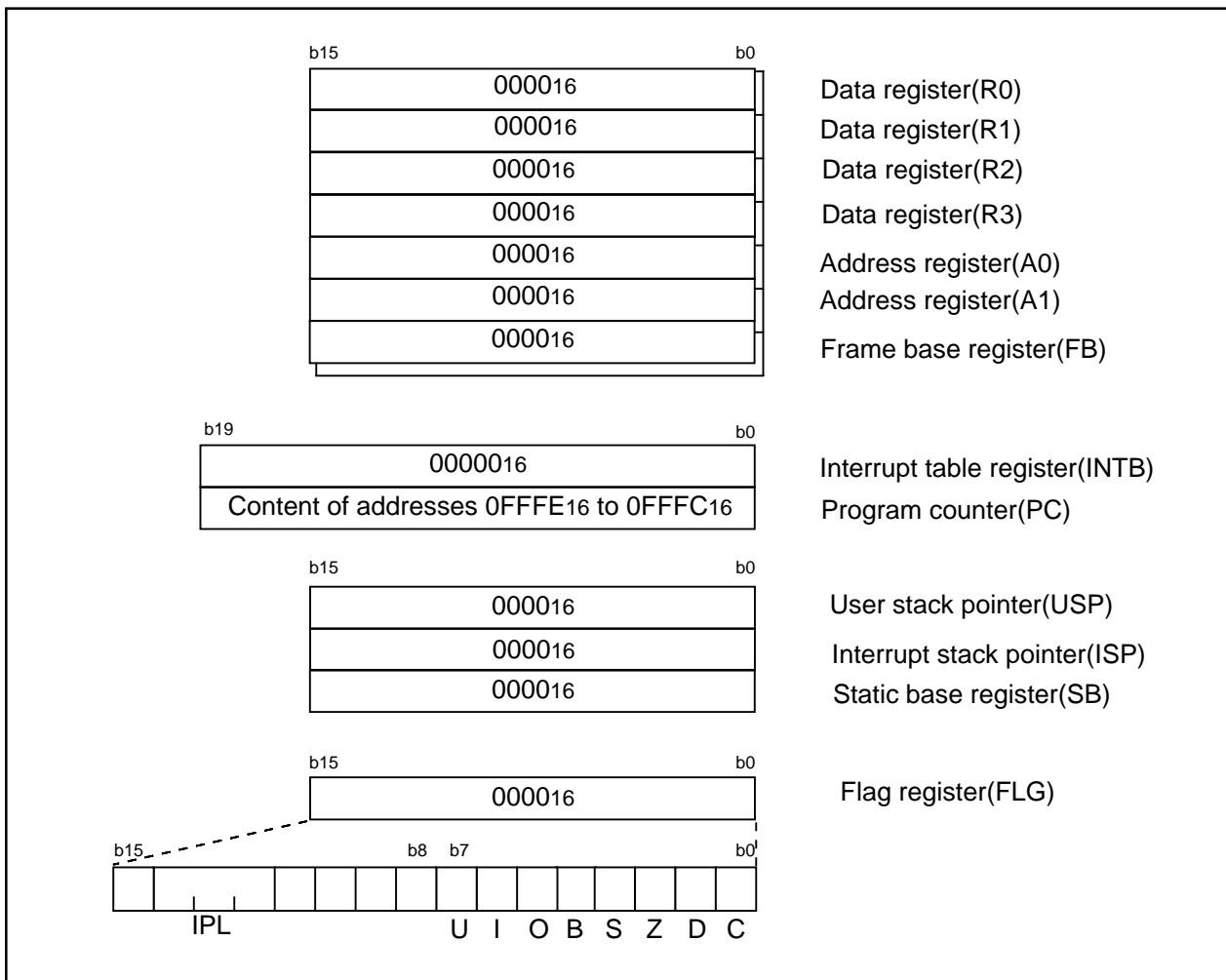


Figure 5.1 CPU Register Status After Reset

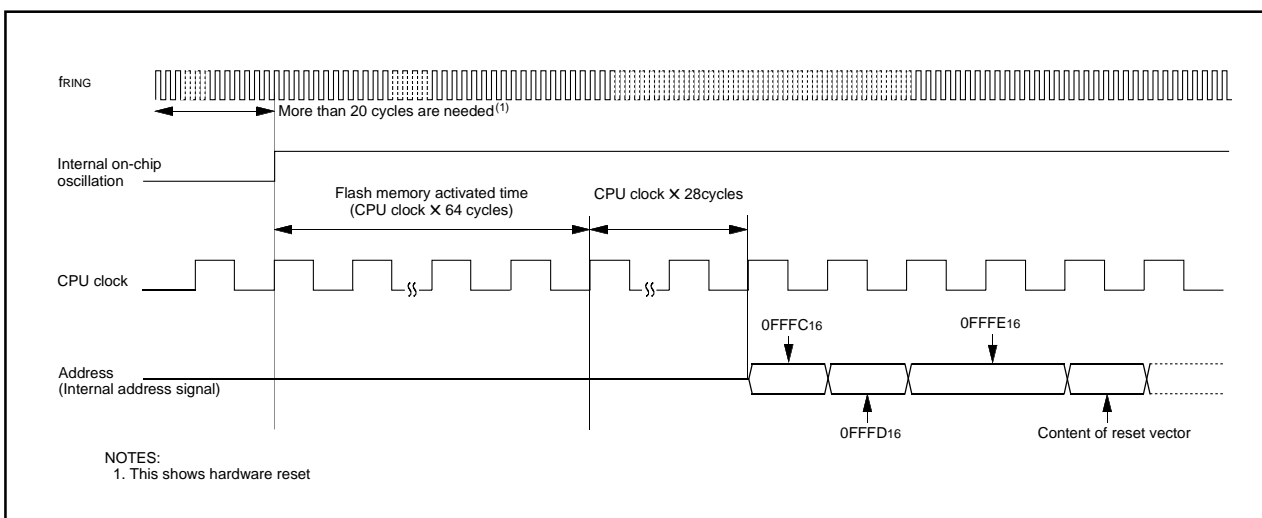


Figure 5.2 Reset Sequence

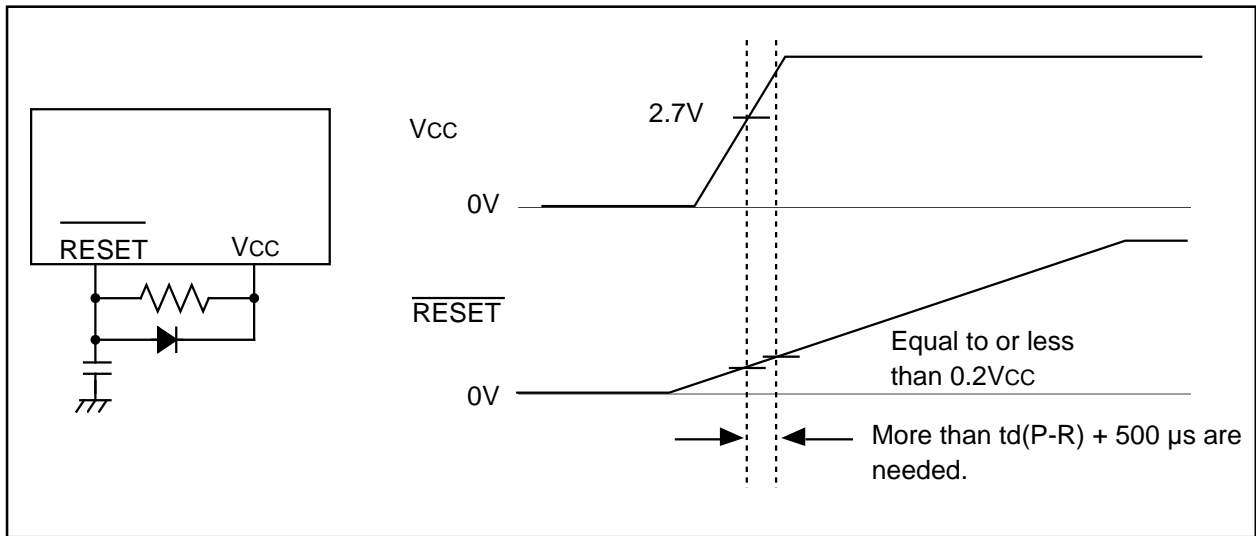


Figure 5.3 Example Reset Circuit

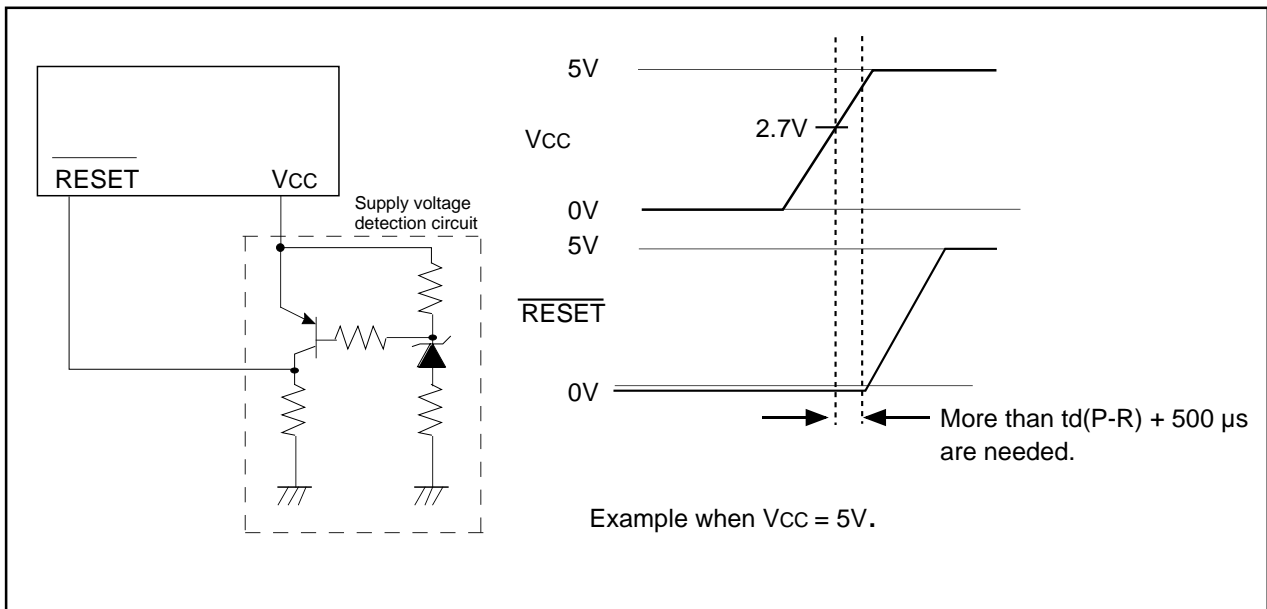


Figure 5.4 Example Reset Circuit (Voltage Check Circuit)

6. Clock Generation Circuit

The clock generation circuit contains two oscillator circuits as follows:

- Main clock oscillation circuit
- On-chip oscillator (with oscillation stop detection 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	On-chip oscillator
Use of clock	<ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when the main clock stops oscillating 	<ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when the main clock stops oscillating
Clock frequency	0 to 16 MHz	About 125 kHz
Usable oscillator	<ul style="list-style-type: none"> • Ceramic resonator • Crystal oscillator 	_____
Pins to connect oscillator	XIN, XOUT ⁽¹⁾	(Note 1)
Oscillation starts and stops	Present	Present
Oscillator status after reset	Stopped	Oscillating
Other	Externally derived clock can be input	_____

NOTES:

1. Can be used as P46 and P47 when the on-chip oscillator clock is used for CPU clock while the main clock oscillation circuit is not used.

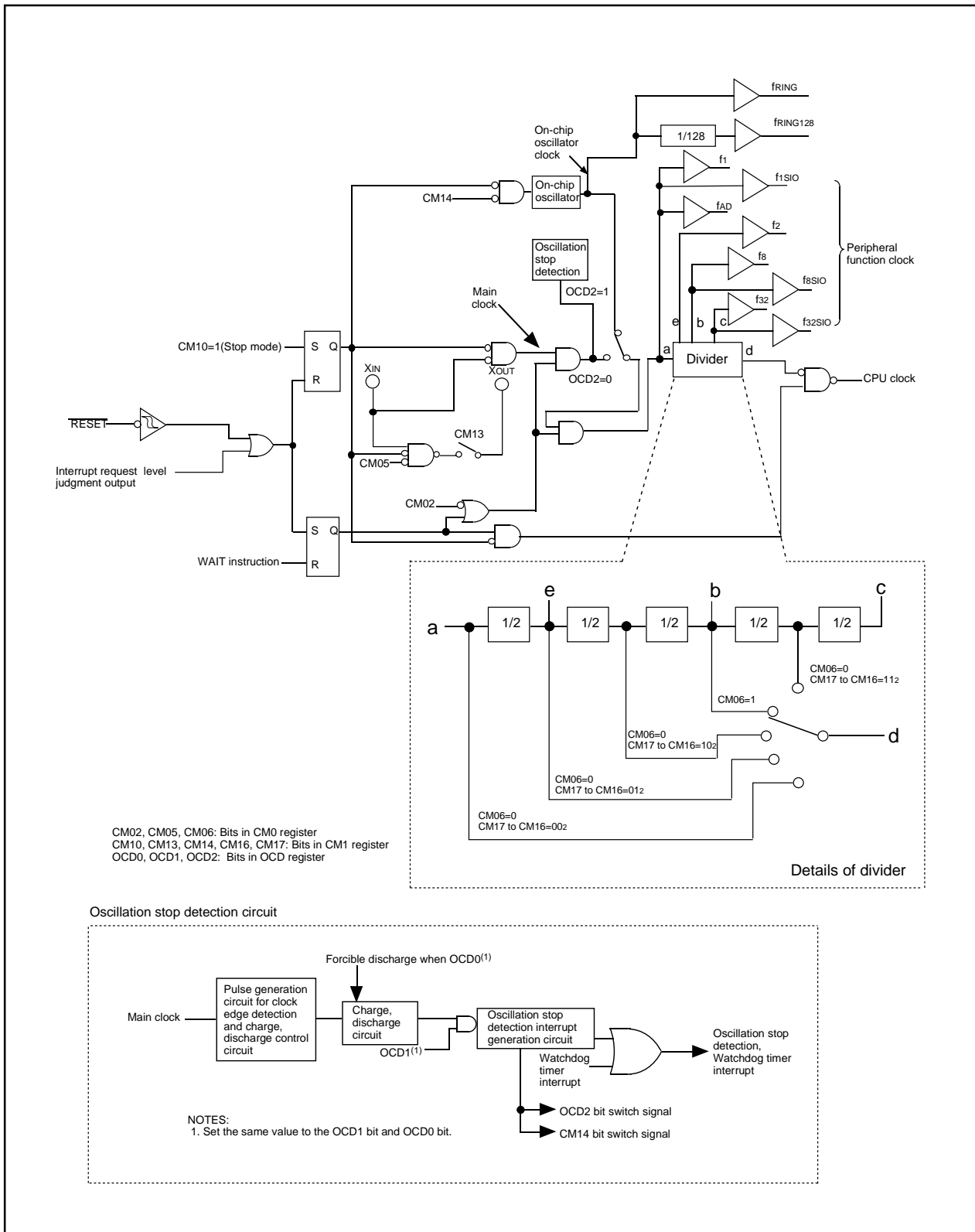


Figure 6.1 Clock Generation Circuit

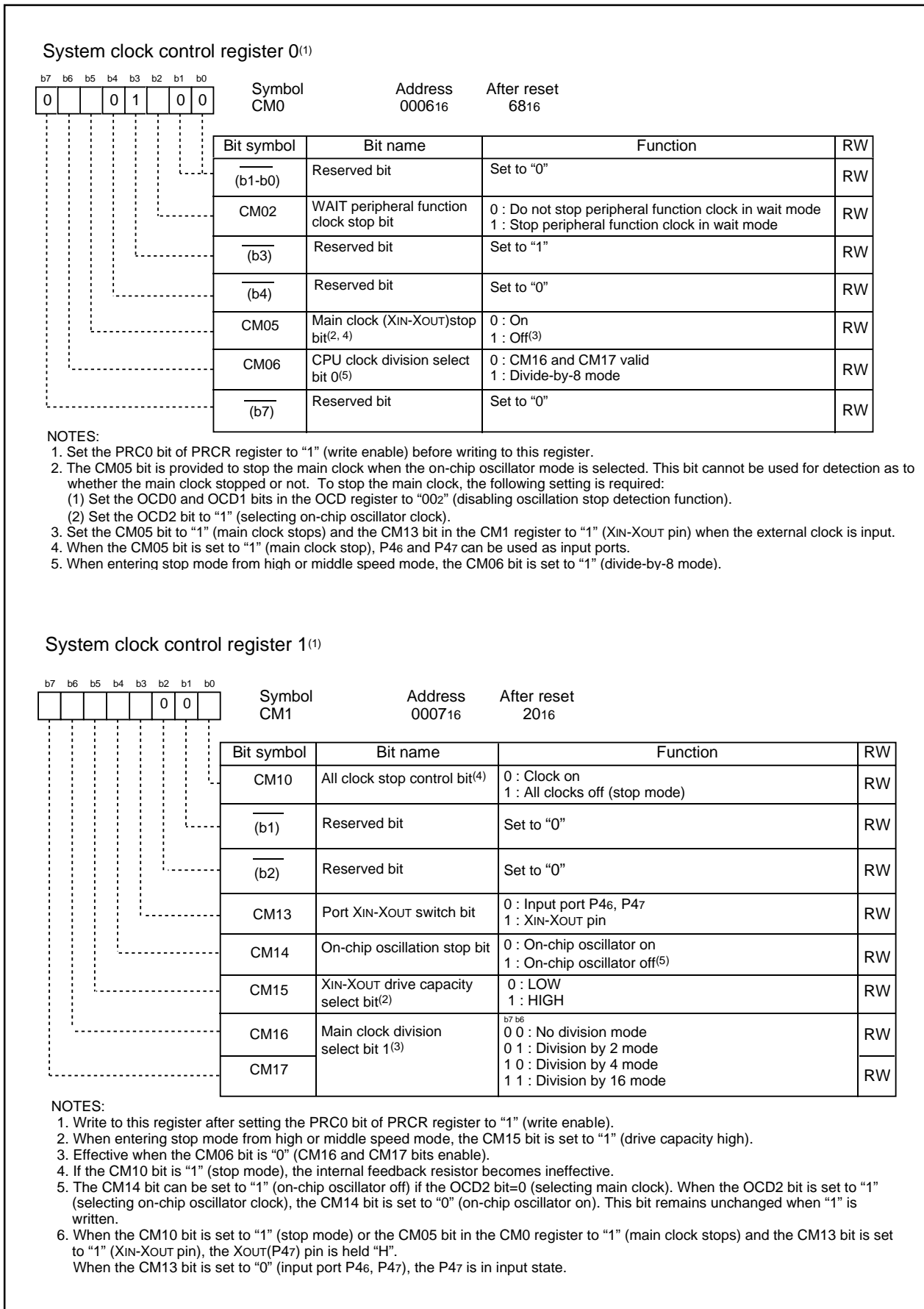


Figure 6.2 CM0 Register and CM1 Register

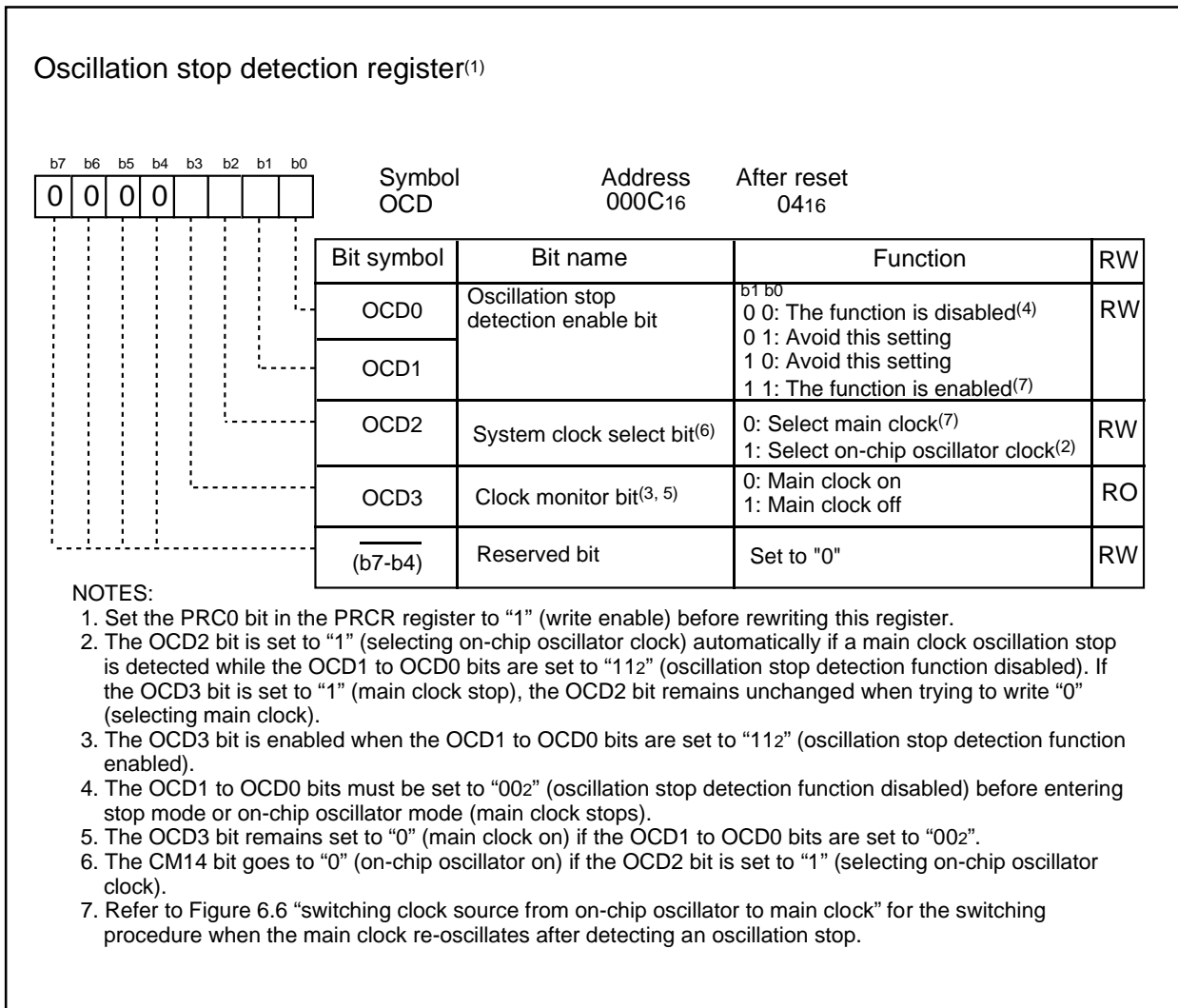


Figure 6.3 OCD Register

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.

During reset and 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 on-chip 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."

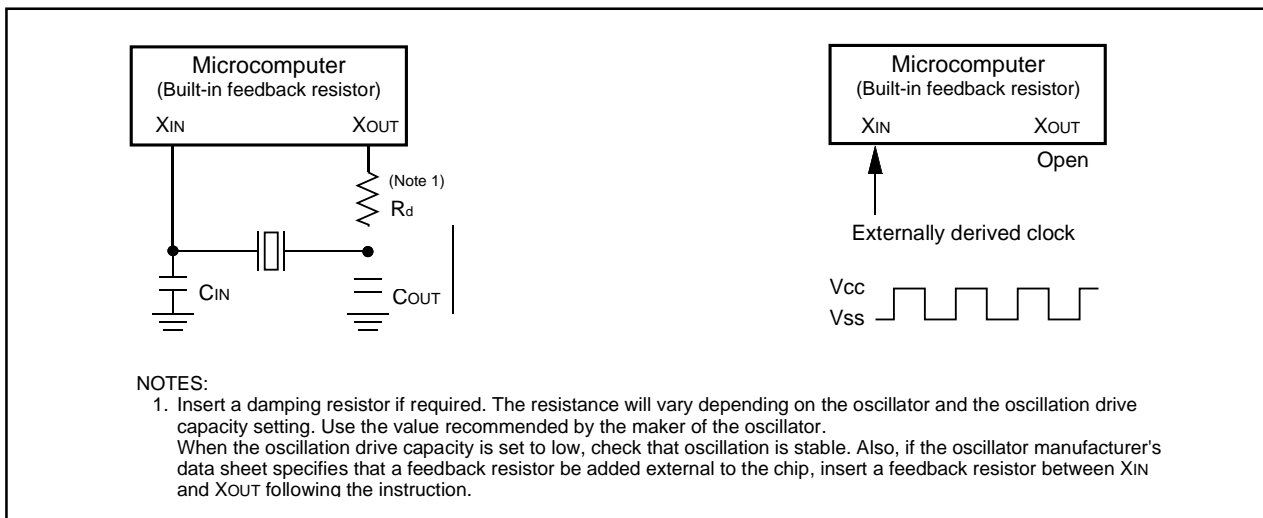


Figure 6.4 Examples of Main Clock Connection Circuit

6.2 On-Chip Oscillator Clock

This clock, approximately 125 kHz, is supplied by the on-chip oscillator. This clock is used as the clock source for the CPU clock, peripheral function clock, FRING, and FRING128.

After reset, the on-chip 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 "112" (oscillation stop detection function enabled), the on-chip oscillator automatically starts operating, supplying the necessary clock for the microcomputer.

The frequency of the on-chip oscillator varies depending on the supply voltage and the operation ambient temperature. The application products must be designed with sufficient margin for the frequency change.

6.3 CPU Clock and Peripheral Function Clock

There are two types of 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 on-chip 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 on-chip oscillator clock divided by 8 provides the CPU clock.

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 (f₁, f₂, f₈, f₃₂, f_{AD}, f_{1SIO}, f_{8SIO}, f_{32SIO})

These are operating clocks for the peripheral functions.

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

The clock f_{jSIO} (j=1, 8, 32) is derived from the main clock or on-chip oscillator clock by dividing them by j. The clock f_{jSIO} is used for serial interface.

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 f_i, f_{jSIO}, 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 on-chip 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 be 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. All modes other than wait and stop modes are referred to as normal operation mode.

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" (on-chip oscillator on), the FRING 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" (on-chip oscillator on), the FRING is used as the count source for timer Y.

- **On-Chip Oscillator Mode**

The on-chip oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The on-chip oscillator clock is also the clock source for the peripheral function clocks.

Table 6.2 Setting Clock Related Bit and Modes

Modes		OCD register	CM1 register		CM0 register	
		OCD2	CM17, CM16	CM13	CM06	CM05
High-speed mode		0	002	1	0	0
Medium-speed mode	divided by 2	0	012	1	0	0
	divided by 4	0	102	1	0	0
	divided by 8	0	—	1	1	0
	divided by 16	0	112	1	0	0
On-chip oscillator mode	no division	1	002	—	0	0 or 1
	divided by 2	1	012	—	0	0 or 1
	divided by 4	1	102	—	0	0 or 1
	divided by 8	1	—	—	1	0 or 1
	divided by 16	1	112	—	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 on-chip 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 f1, f2, f8, f32, f1SIO, f8SIO, f32SIO, and fAD 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 “0002” (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 “0002” (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 interface interrupt	Can be used when operating with internal or external clock	Can be used when operating with external clock
Key input interrupt	Can be used	Can be used
A/D conversion interrupt	Can be used in one-shot mode	— (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
Timer Z interrupt	Can be used in all modes	— (Do not use)
Timer C interrupt	Can be used in all modes	— (Do not use)
INT interrupt	Can be used	Can be used (INT0 and INT3 can be used if there is no filter.)
Voltage detection interrupt	Can be used	Can be used
Oscillation stop detection interrupt	Can be used	— (Do not use)

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
- $\overline{\text{INT}}0$ to $\overline{\text{INT}}2$ interrupts ($\overline{\text{INT}}0$ 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 interface 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 capacity high).

Before entering stop mode, set the OCD1 to OCD0 bits to "002" (oscillation stop detection function disable).

• Pin Status in Stop Mode

The status before wait mode is retained.

However, the XOUT(P47) pin is held "H" when the CM13 bit in the CM1 register is set to "1" (XIN-XOUT pin). The P47(XOUT) is in input state when the CM13 bit is set to "0" (input port P46, P47).

• 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 "0002" (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 "0002".

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

In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt 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 of Power control

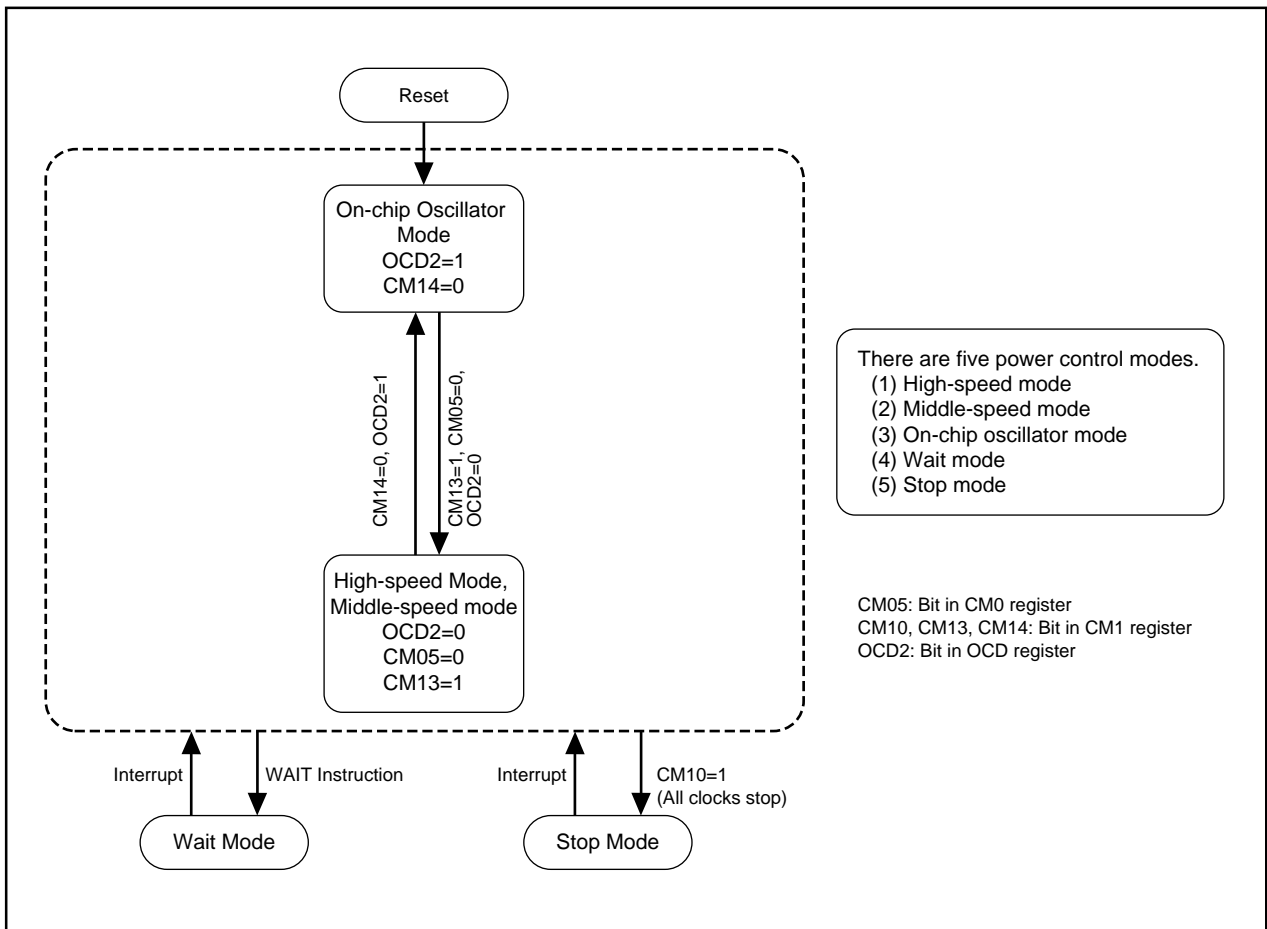


Figure 6.5 State Transition of Power Control

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 “112” (oscillation stop detection function enabled), the system is placed in the following state if the main clock comes to a halt:

- The on-chip oscillator starts oscillation, and the on-chip oscillator clock becomes the clock source for CPU clock and peripheral functions in place of the main clock
- OCD register OCD2 bit = 1 (selecting on-chip oscillator clock)
- OCD register OCD3 bit = 1 (main clock stopped)
- CM1 register CM14 bit = 0 (on-chip 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	$f(X_{IN}) \geq 2 \text{ MHz}$
Enabling condition for oscillation stop detection function	Set OCD1 to OCD0 bits to “112” (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 factor must be determined. Table 6.5 shows to determine the interrupt factor with the oscillation stop detection interrupt, watchdog timer interrupt and voltage detection interrupt.
- 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 6.6 shows the procedure for switching the clock source from the on-chip 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 “002” (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 when the main clock frequency is below 2 MHz. Set the OCD1 to OCD0 bits to “002” (oscillation stop detection function disabled).

Table 6.5 Determination of Interrupt Factor of Oscillation Stop Detection

Generated Interrupt Factor	Bit showing interrupt factor
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 = 112 and the OCD2 bit = 1

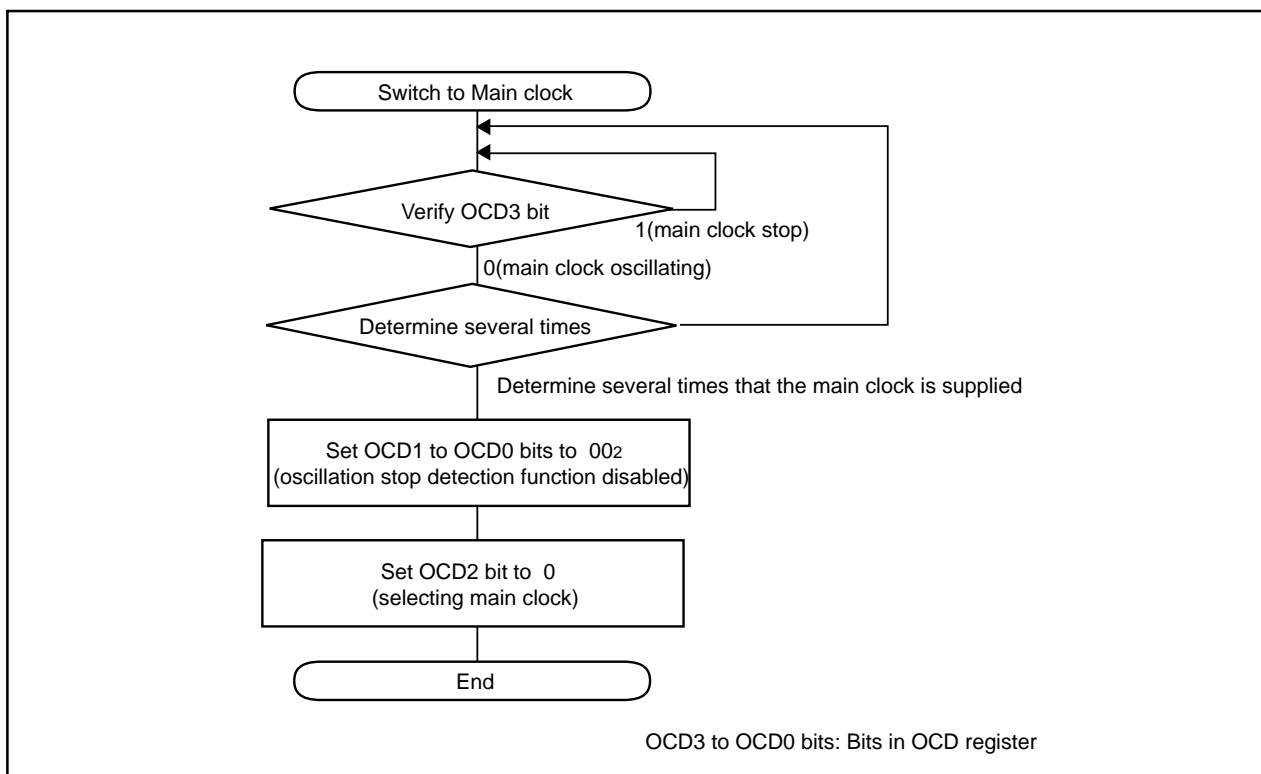


Figure 6.6 Switching Clock Source From On-Chip Oscillator to Main Clock

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.

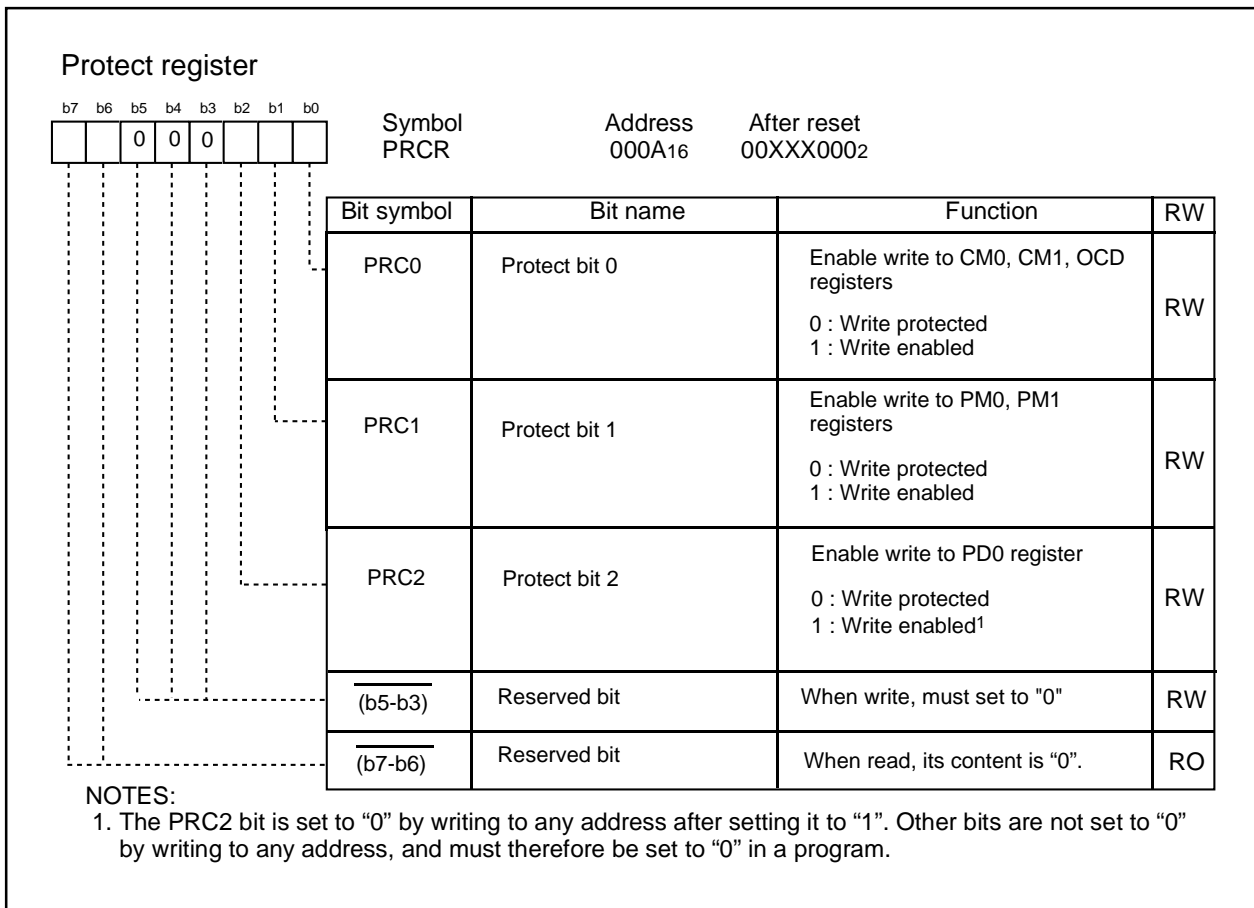


Figure 7.1 PRCR Register

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

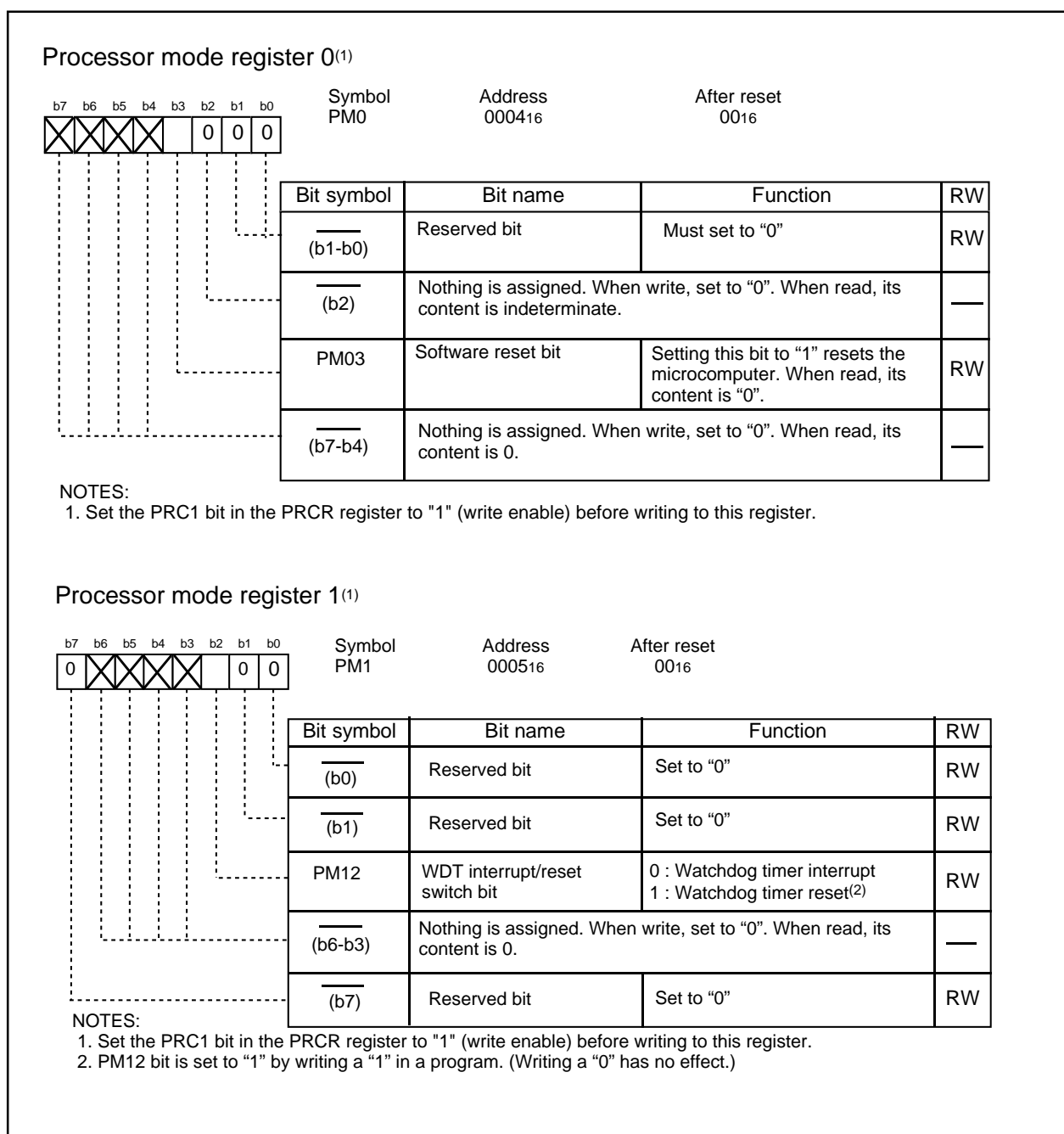


Figure 8.1 PM0 Register and PM1 Register

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/Data flash	2 CPU clock cycles
Program ROM/RAM	1 CPU clock cycles

Table 9.2 Access Unit and Bus Operation

Space	SFR, Data flash	Program ROM/RAM
Even address byte access		
Odd address byte access		
Even address word access		
Odd address word access		

10. Interrupt

10.1 Interrupt Overview

10.1.1 Type of Interrupts

Figure 10.1 shows types of interrupts.

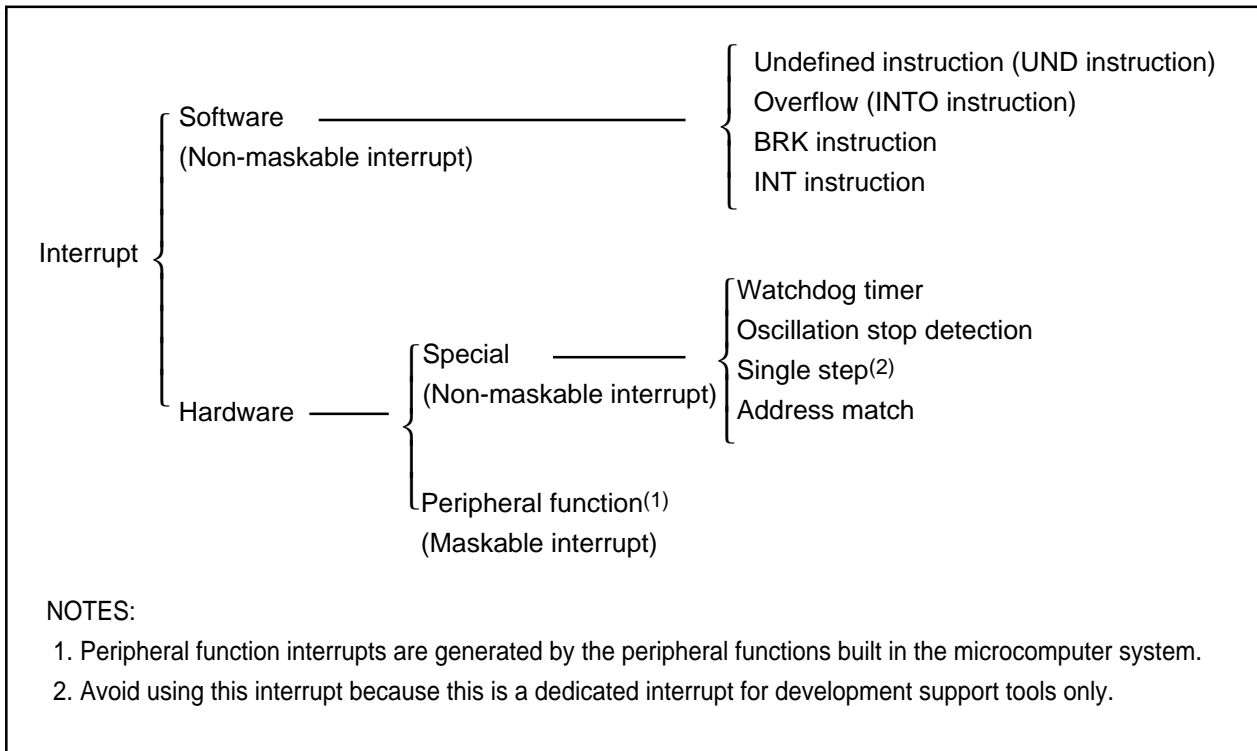


Figure 10.1 Interrupts

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

10.1.2 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 numbers 0 to 63 can be specified for the INT instruction. Because software interrupt numbers 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 numbers 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 numbers 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 factors 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.

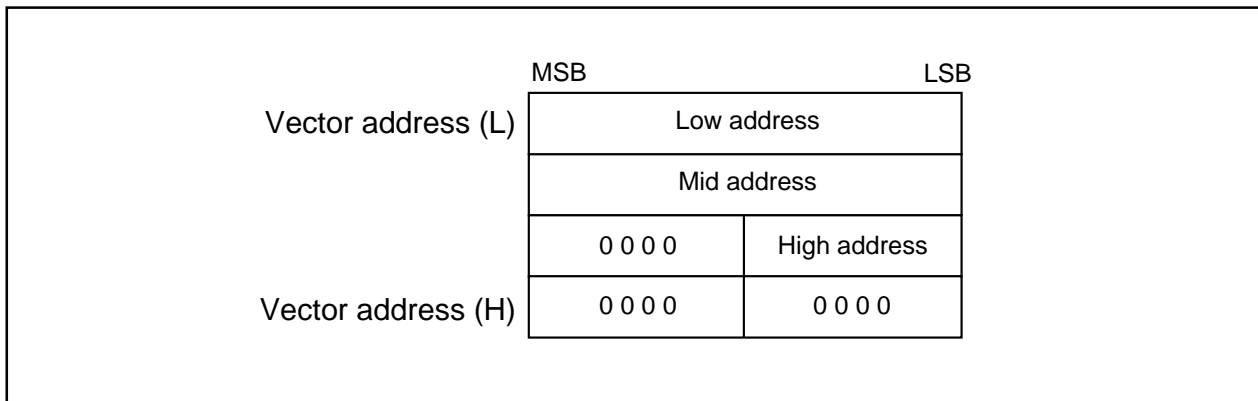


Figure 10.2 Interrupt Vector

• Fixed Vector Tables

The fixed vector tables are allocated to the addresses from 0FFDC₁₆ to 0FFFF₁₆. 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 factor	Vector addresses Address (L) to address (H)	Remarks	Reference
Undefined instruction	0FFDC ₁₆ to 0FFDF ₁₆	Interrupt on UND instruction	R8C/Tiny 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		10.4 Address match interrupt
Single step ⁽¹⁾	0FFEC ₁₆ to 0FFEF ₁₆		
Watchdog timer, oscillation stop detection	0FFF0 ₁₆ to 0FFF3 ₁₆		11. Watchdog timer, 6. Clock generation circuit
(Reserved)	0FFF4 ₁₆ to 0FFF7 ₁₆		
(Reserved)	0FFF8 ₁₆ to 0FFFB ₁₆		
Reset	0FFFC ₁₆ to 0FFFF ₁₆		5. Reset

NOTES:

- 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 Relocatable Vector Tables

Interrupt factor	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	+52 to +55 (0034 ₁₆ to 0037 ₁₆)	13	10.3 Key input interrupt
A/D Conversion	+56 to +59 (0038 ₁₆ to 003B ₁₆)	14	14. A/D converter
——— (Reserved)		15, 16	
UART0 transmit	+68 to +71 (0044 ₁₆ to 0047 ₁₆)	17	13. Serial interface
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	
$\overline{\text{INT}}_2$	+84 to +87 (0054 ₁₆ to 0057 ₁₆)	21	10.2.3 $\overline{\text{INT}}_2$ interrupt
Timer X	+88 to +91 (0058 ₁₆ to 005B ₁₆)	22	12.1 Timer X
Timer Y	+92 to +95 (005C ₁₆ to 005F ₁₆)	23	12.2 Timer Y
Timer Z	+96 to +99 (0060 ₁₆ to 0063 ₁₆)	24	12.3 Timer Z
$\overline{\text{INT}}_1$	+100 to +103 (0064 ₁₆ to 0067 ₁₆)	25	12.2.3 $\overline{\text{INT}}_1$ interrupt
$\overline{\text{INT}}_3$	+104 to +107 (0068 ₁₆ to 006B ₁₆)	26	12.2.4 $\overline{\text{INT}}_3$ interrupt
Timer C	+108 to +111 (006C ₁₆ to 006F ₁₆)	27	12.4 Timer C
——— (Reserved)		28	
$\overline{\text{INT}}_0$	+116 to +119 (0074 ₁₆ to 0077 ₁₆)	29	10.2.1 $\overline{\text{INT}}_0$ 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.

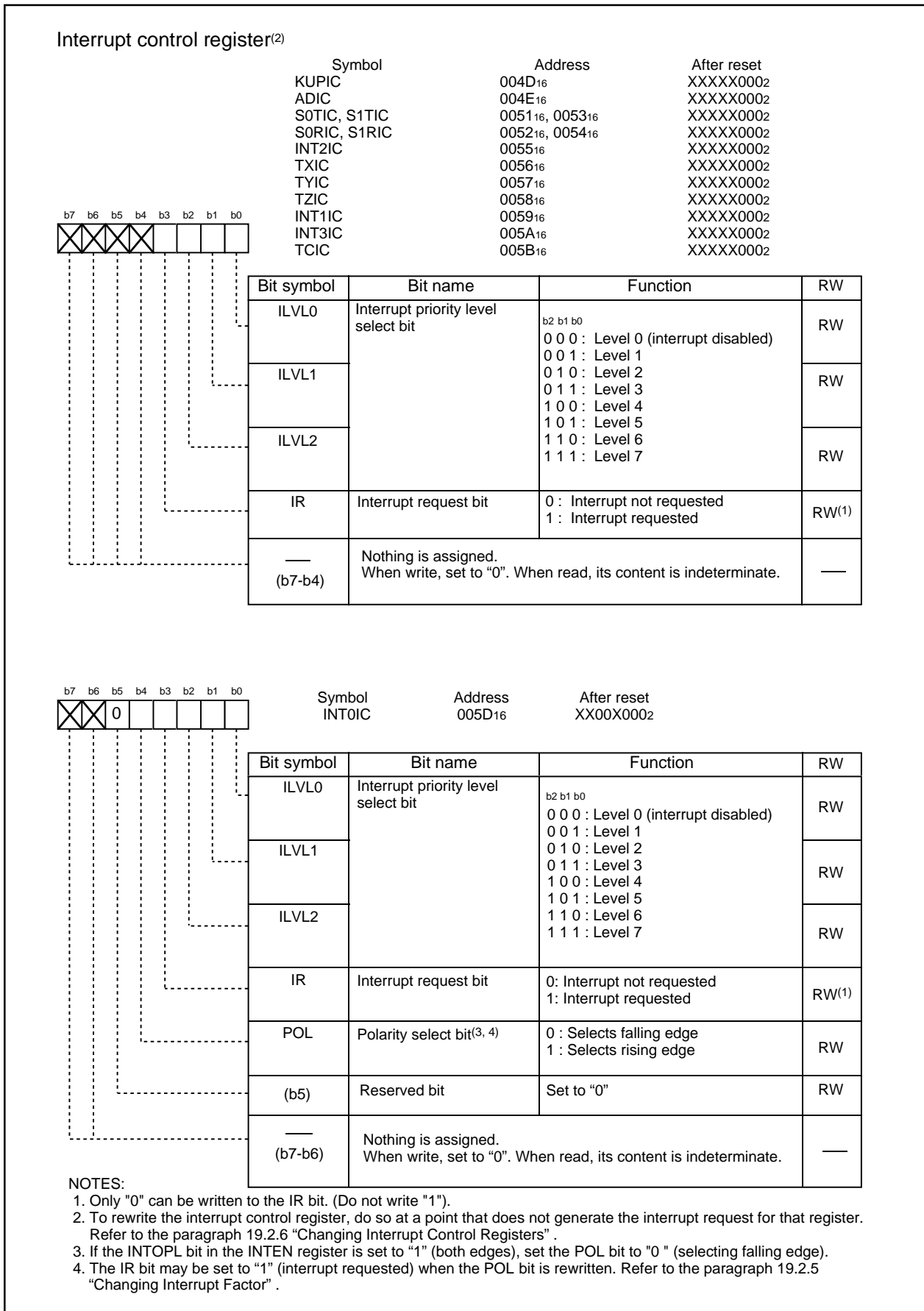


Figure 10.3 Interrupt Control Registers

- **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
0002	Level 0 (interrupt disabled)	————
0012	Level 1	Lowest  Highest
0102	Level 2	
0112	Level 3	
1002	Level 4	
1012	Level 5	
1102	Level 6	
1112	Level 7	

Table 10.4 Interrupt Priority Levels Enabled by IPL

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

• 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 00000₁₆. 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 internal temporary register⁽¹⁾.
- (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 numbers 32 to 63 is executed.
- (4) The CPU's internal temporary register⁽¹⁾ 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.

NOTES:

1. This register cannot be used by user.

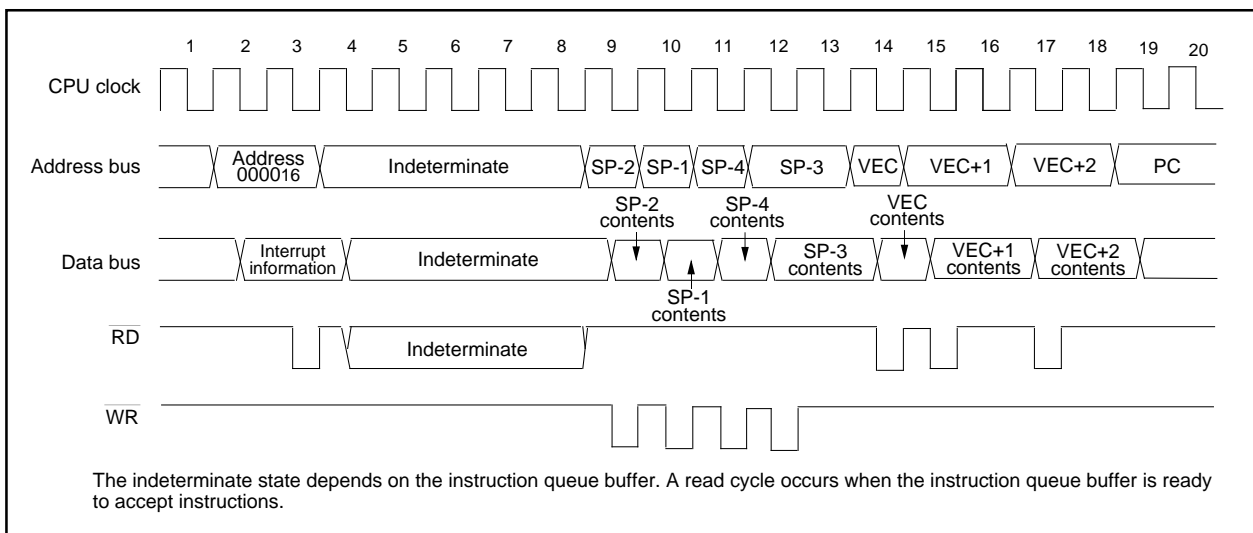


Figure 10.4 Time Required for Executing Interrupt Sequence

• 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).

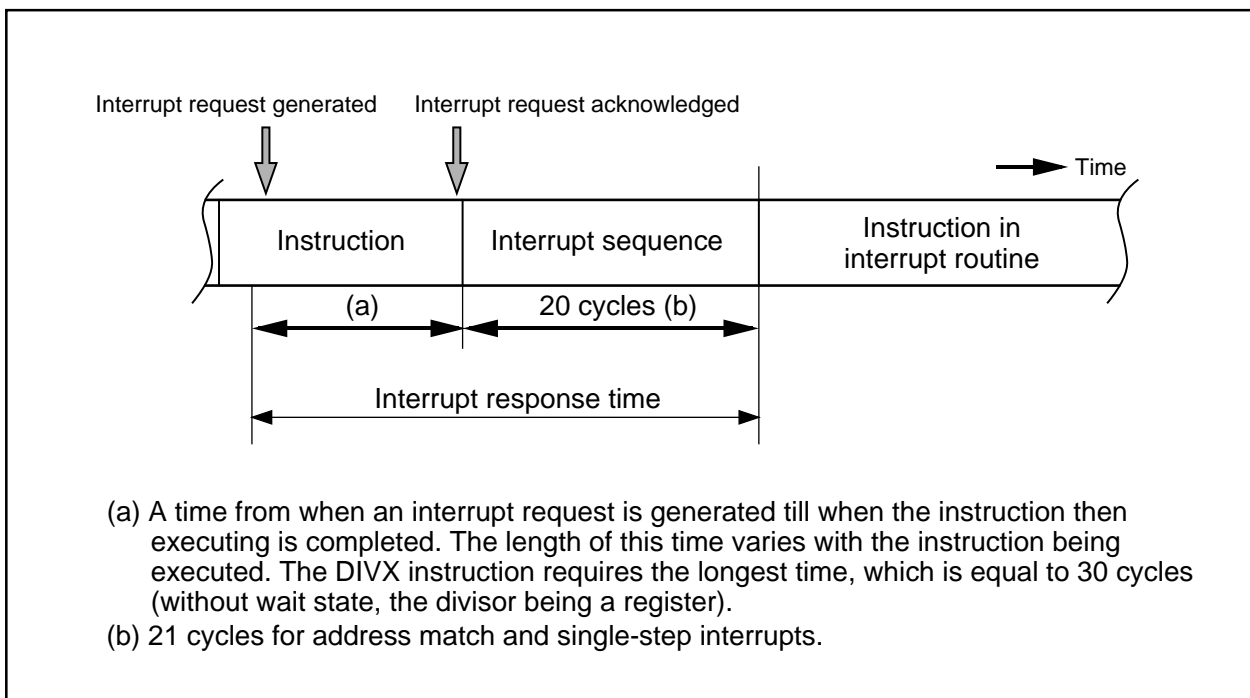


Figure 10.5 Interrupt Response Time

• 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 factors	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. The PUSHM instruction can save several registers in the register bank being currently used⁽¹⁾ with a single instruction.

NOTES:

1. Selectable from registers R0, R1, R2, R3, A0, A1, SB, and FB.

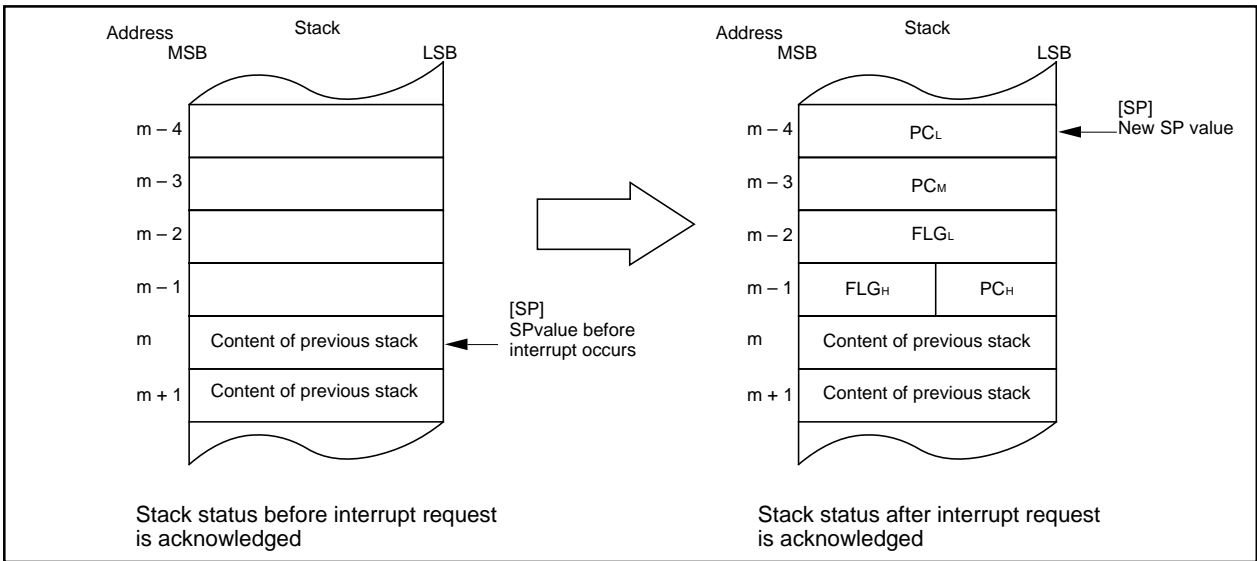
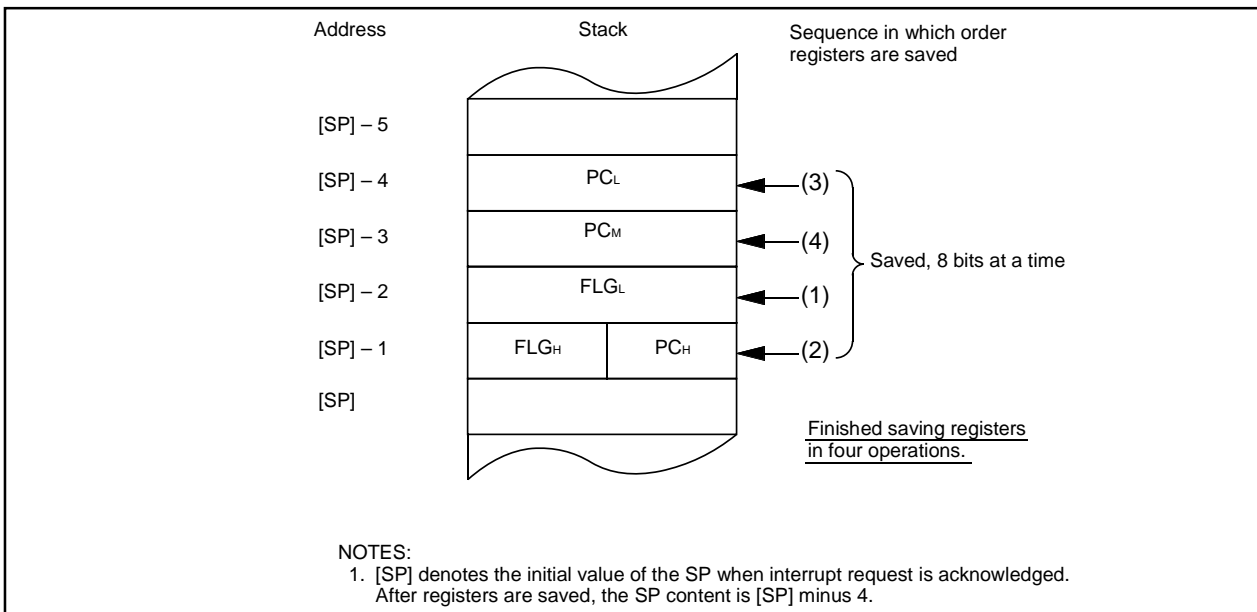


Figure 10.6 Stack Status Before and After Acceptance of Interrupt Request

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

NOTES:

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



NOTES:

1. [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 Hardware Interrupt Priority.

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 Interrupts Priority Select Circuit.

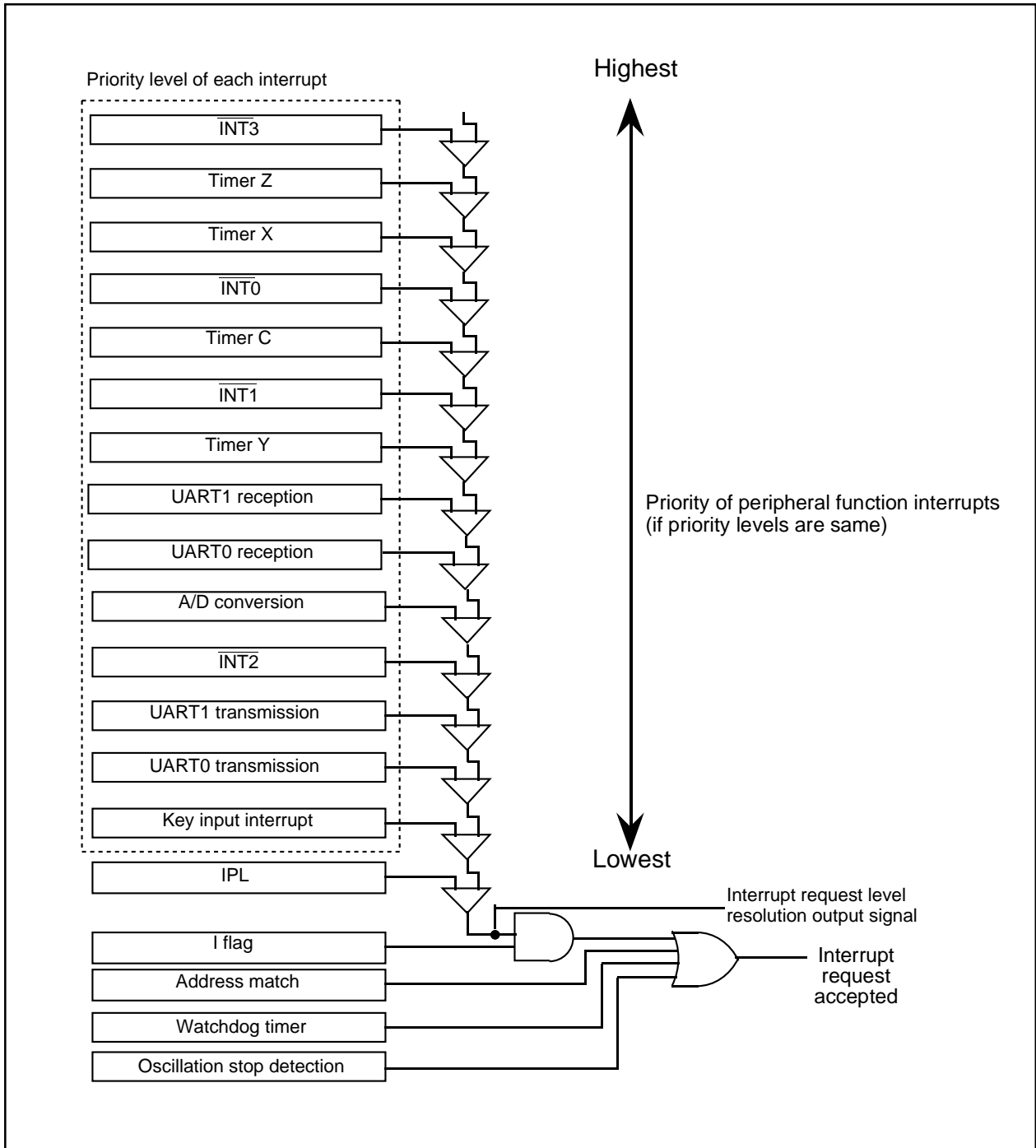


Figure 10.9 Interrupts Priority Select Circuit

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

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

$\overline{\text{INT0}}$ interrupt is triggered by an $\overline{\text{INT0}}$ input. When using $\overline{\text{INT0}}$ interrupts, the $\overline{\text{INT0EN}}$ bit in the $\overline{\text{INTEN}}$ register must be set to "1" (enabling). The edge polarity is selected using the $\overline{\text{INT0PL}}$ bit in the $\overline{\text{INTEN}}$ register and the $\overline{\text{POL}}$ bit in the $\overline{\text{INT0IC}}$ register.

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

The $\overline{\text{INT0}}$ pin is shared with the external trigger input pin of Timer Z.

Figure 10.10 shows the $\overline{\text{INTEN}}$ and $\overline{\text{INT0F}}$ registers.

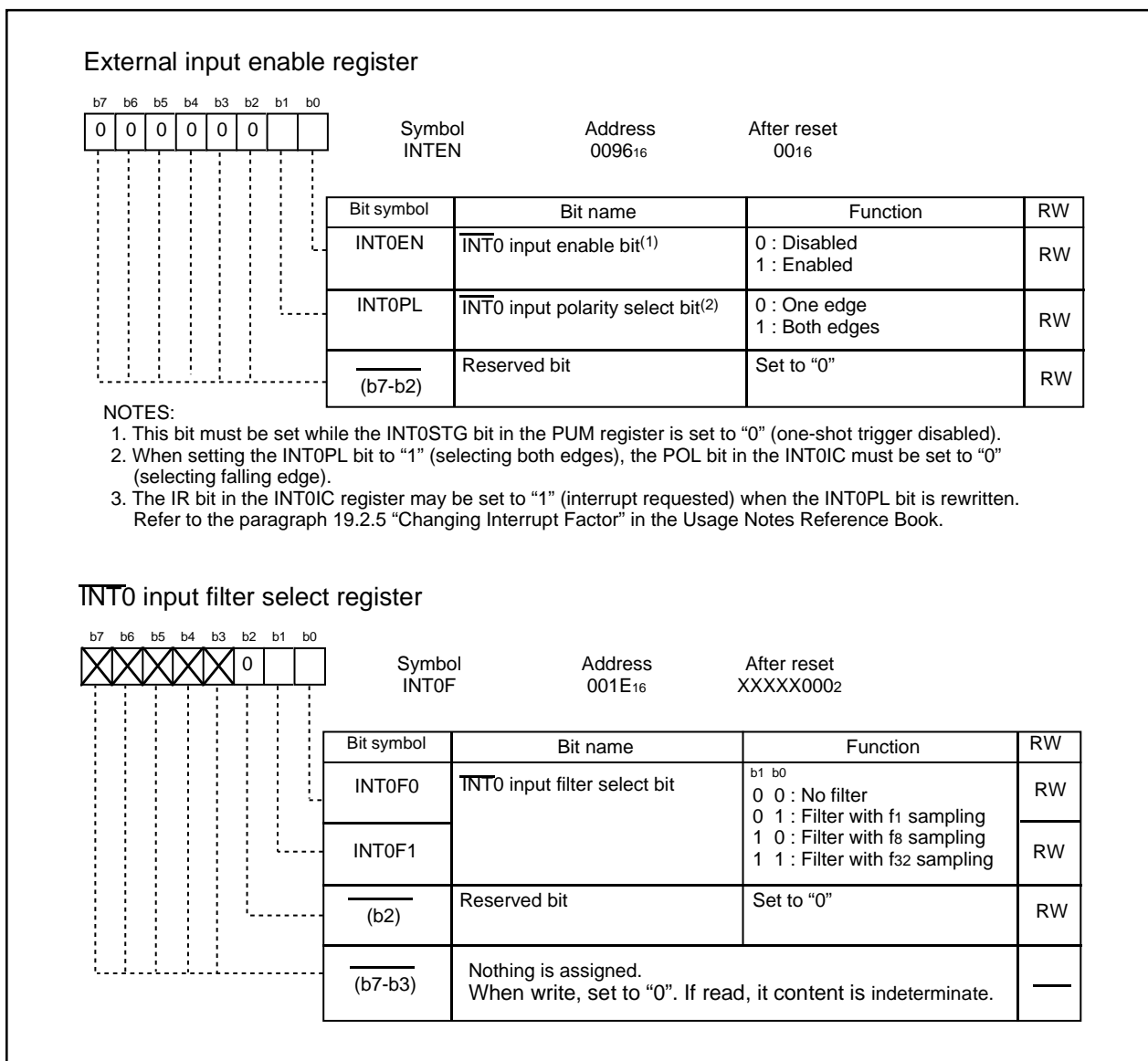


Figure 10.10 $\overline{\text{INTEN}}$ Register and $\overline{\text{INT0F}}$ Register

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 "012", "102", or "112", 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.

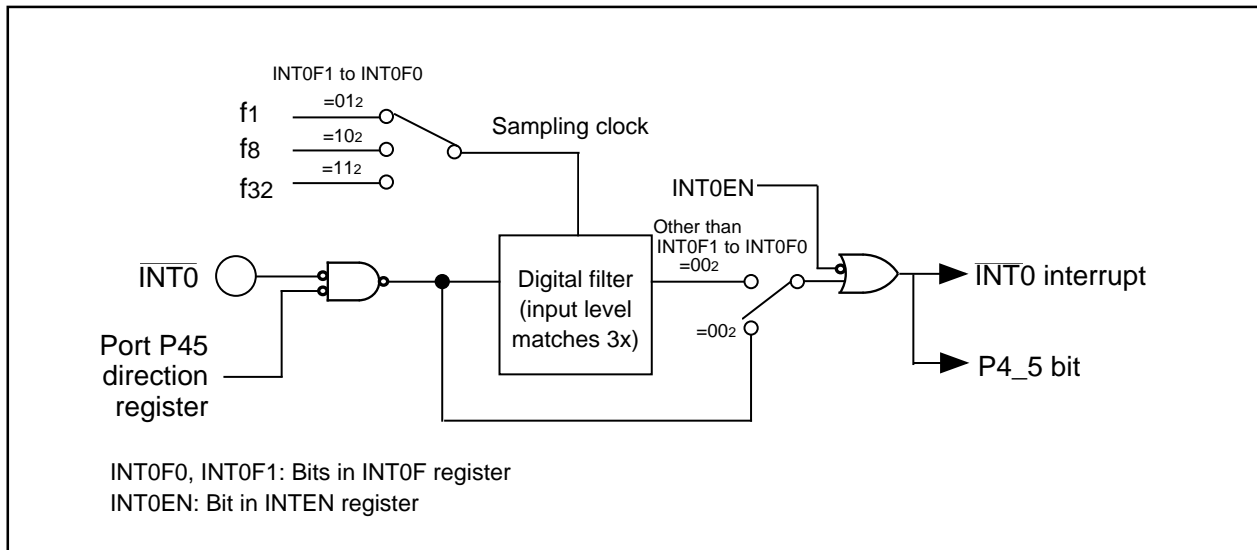


Figure 10.11 $\overline{\text{INT0}}$ Input Filter

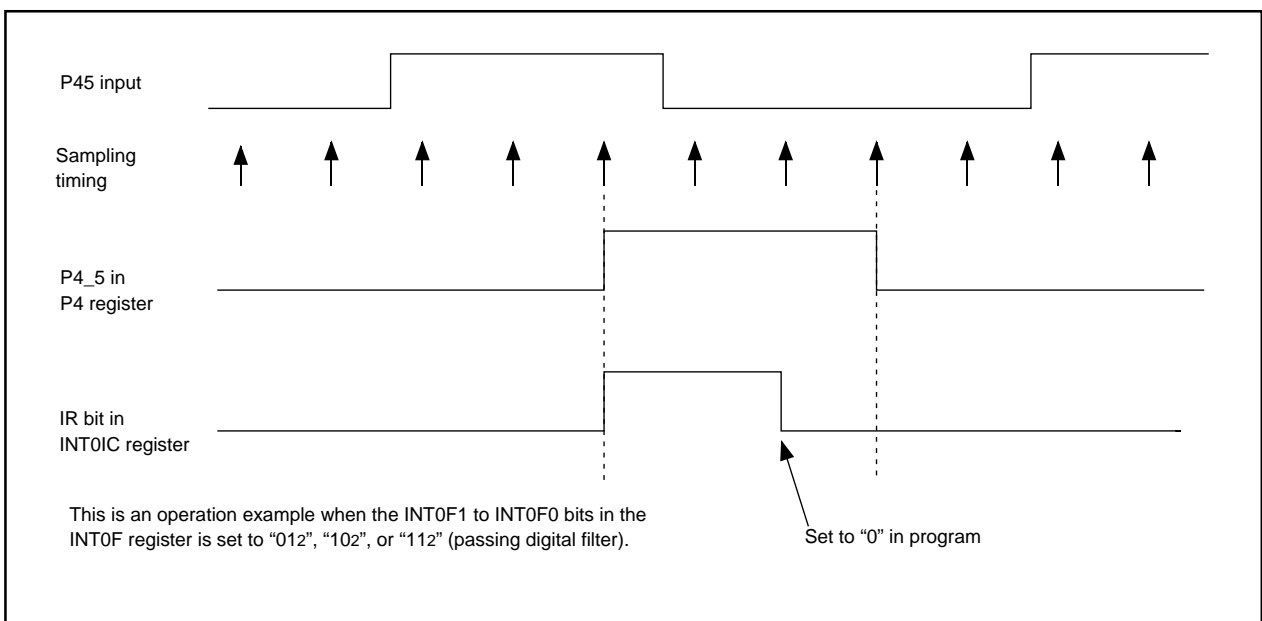


Figure 10.12 Operation Example of $\overline{\text{INT0}}$ Input Filter

10.2.3 $\overline{\text{INT1}}$ Interrupt and $\overline{\text{INT2}}$ Interrupt

$\overline{\text{INT1}}$ interrupts are triggered by $\overline{\text{INT1}}$ inputs. The edge polarity is selected with the R0EDG bit in the TXMR register. The $\overline{\text{INT1}}$ pin is shared with the CNTR0 pin.

$\overline{\text{INT2}}$ interrupts are triggered by $\overline{\text{INT2}}$ inputs. The edge polarity is selected with the R1EDG bit in the TYZMR register. The $\overline{\text{INT2}}$ pin is shared with the CNTR1 pin.

Figure 10.13 shows the TXMR and TYZMR registers when using $\overline{\text{INT1}}$ and $\overline{\text{INT2}}$ interrupts.

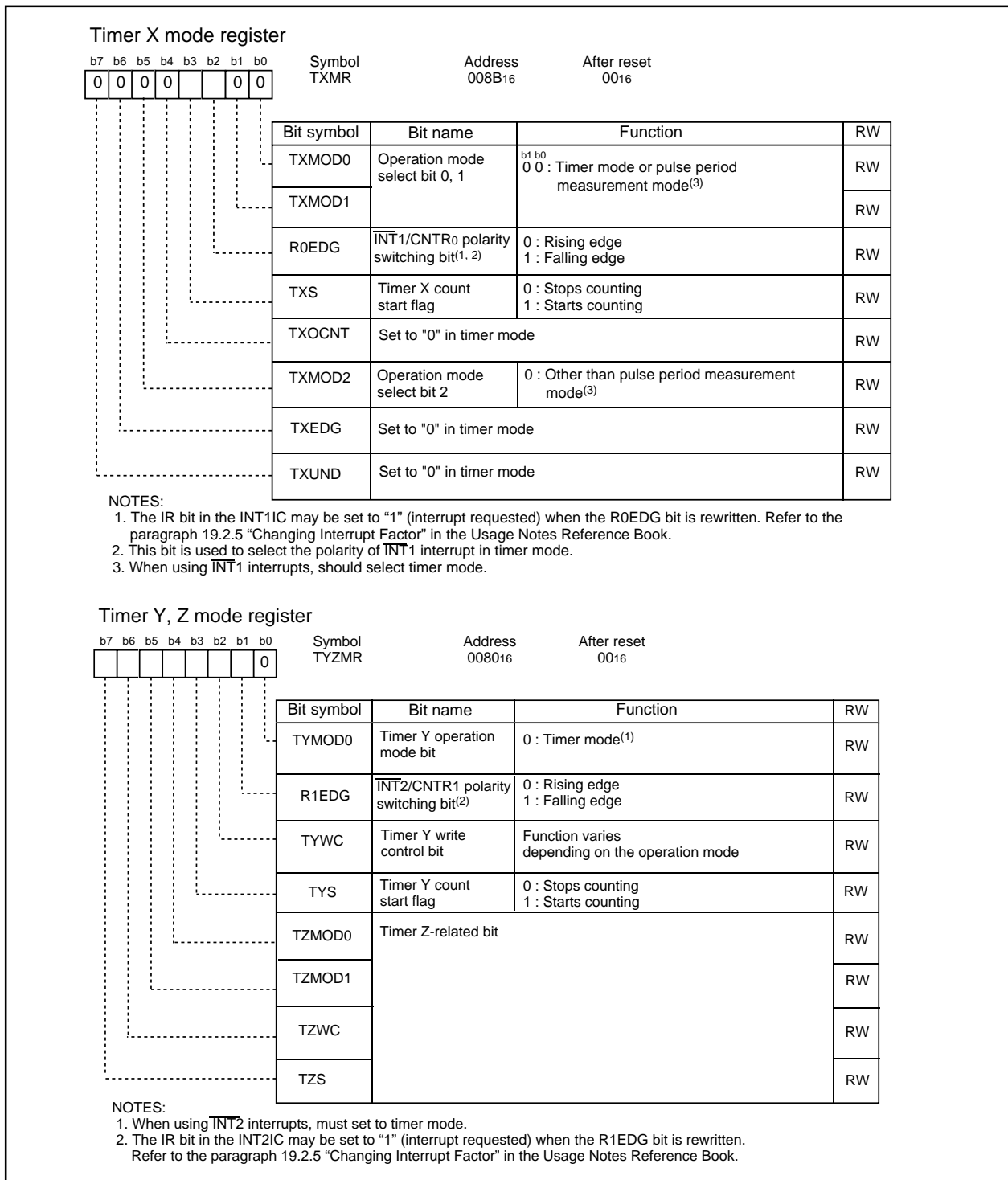


Figure 10.13 TXMR Register and TYZMR Register when $\overline{\text{INT1}}$ and $\overline{\text{INT2}}$ Interrupt Used

10.2.4 $\overline{\text{INT}}$ 3 Interrupt

$\overline{\text{INT}}$ 3 interrupts are triggered by $\overline{\text{INT}}$ 3 inputs. The TCC07 bit in the TCC0 register should be set to "0" ($\overline{\text{INT}}$ 3). The $\overline{\text{INT}}$ 3 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.

The $\overline{\text{INT}}$ 3 pin is shared with the TCIN pin.

When setting the TCC07 bit to "1" (fRING128), $\overline{\text{INT}}$ 3 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.

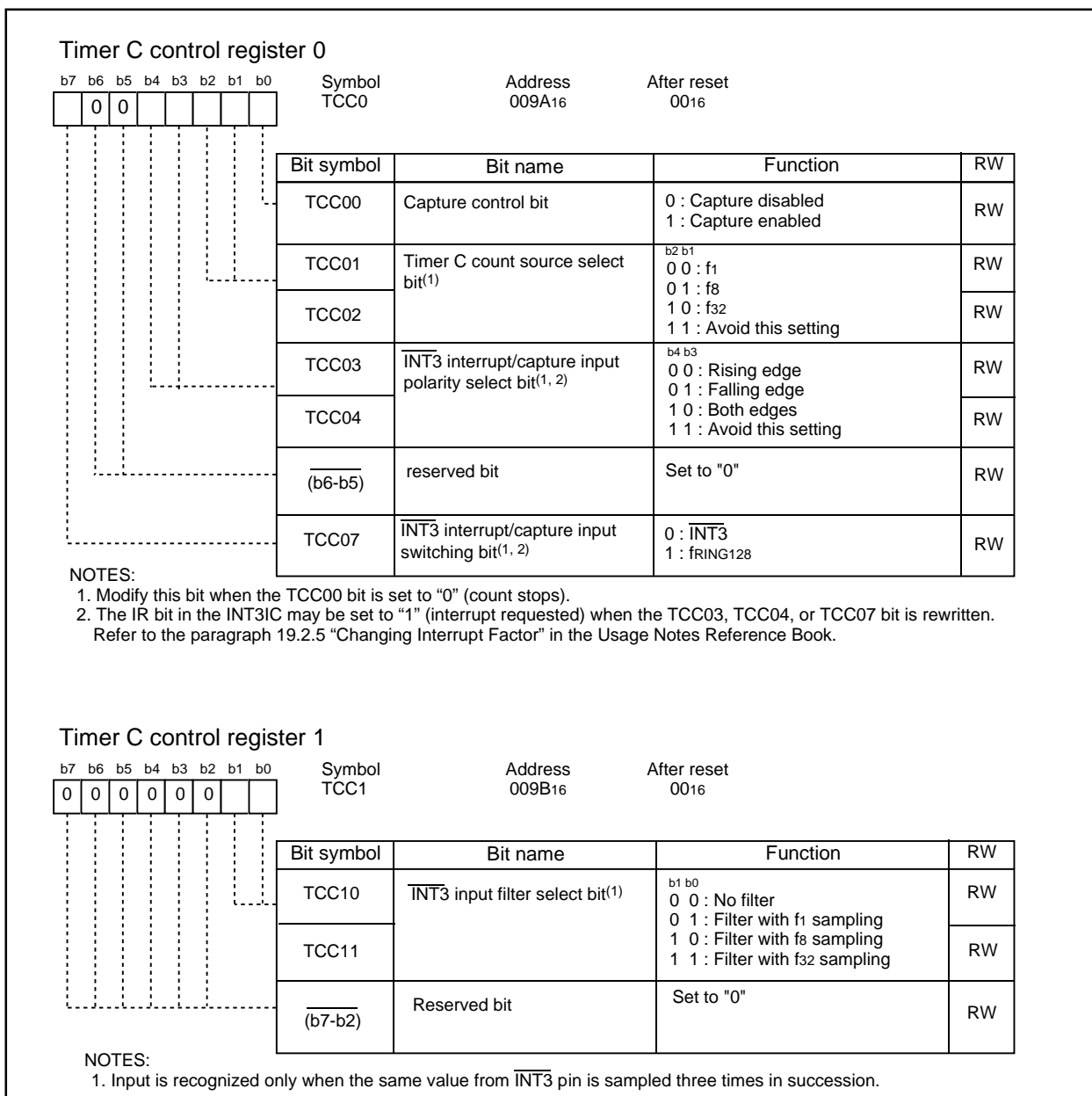


Figure 10.14 TCC0 Register and TCC1 Register

10.3 Key Input Interrupt

A key input interrupt is generated on an input edge of any of the $\overline{K10}$ to $\overline{K13}$ pins. Key input interrupts can be used as a key-on wakeup function to exit wait or stop mode. \overline{Kli} input can be enabled or disabled selecting with the $KliEN$ ($i=0$ to 3) bit in the KIEN register. The edge polarity can be rising edge or falling edge selecting with the $KliPL$ bit in the KIEN register. Note, however, that while input on any \overline{Kli} pin which has had the $KliPL$ 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{Kli} pin which has had the $KliPL$ 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.

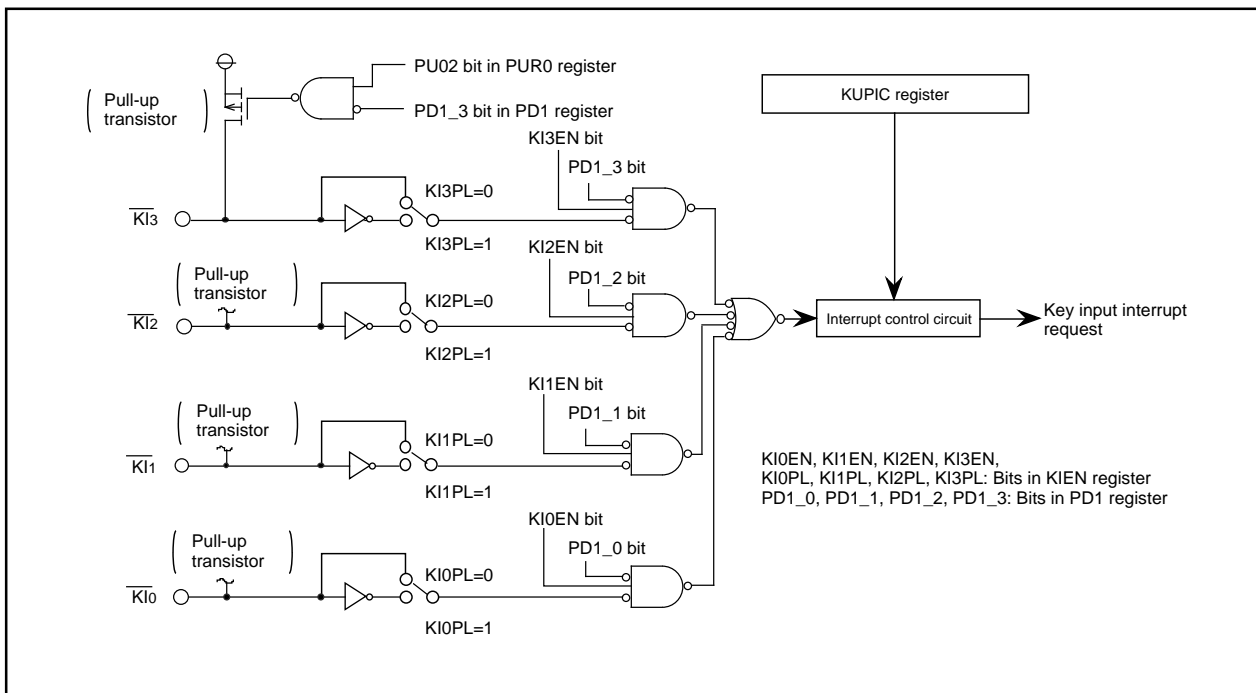


Figure 10.15 Key Input Interrupt

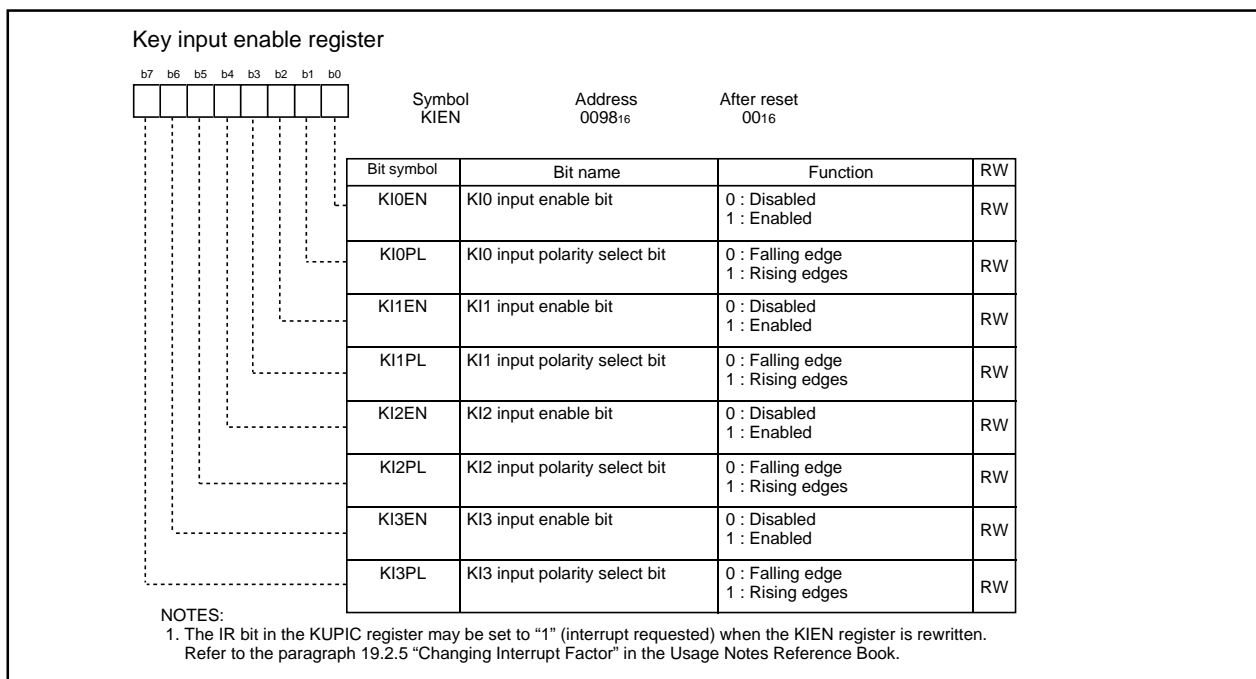


Figure 10.16 KIEN Register

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 ⁽¹⁾
<ul style="list-style-type: none"> • 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
<ul style="list-style-type: none"> • Instructions other than the above 	Address indicated by RMADi register + 1

NOTES:

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

Table 10.7 Relationship Between Address Match Interrupt Factors and Associated Registers

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

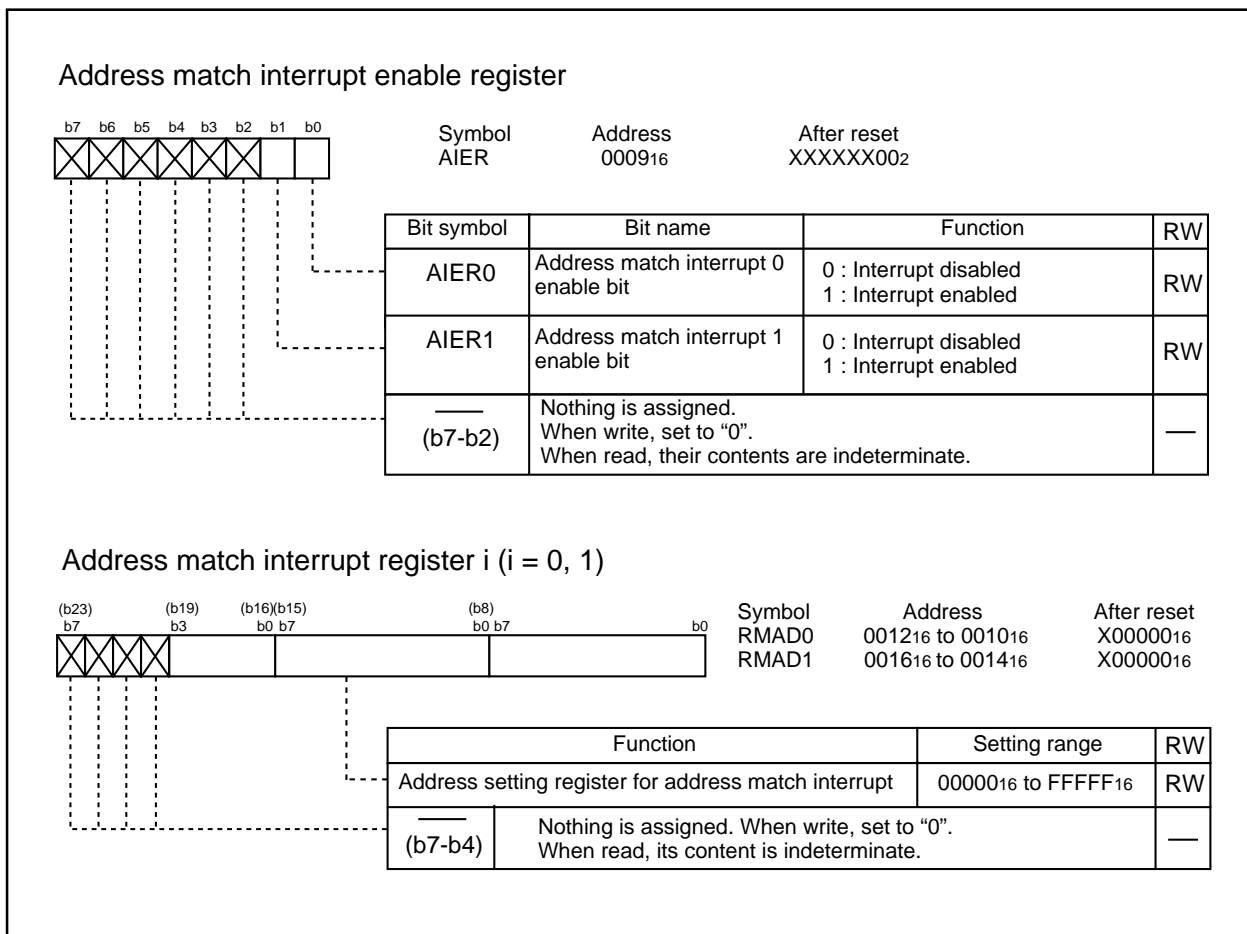


Figure 10.17 AIER Register and RMAD0 to RMAD1 Registers

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 5.3, "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.

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

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

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.

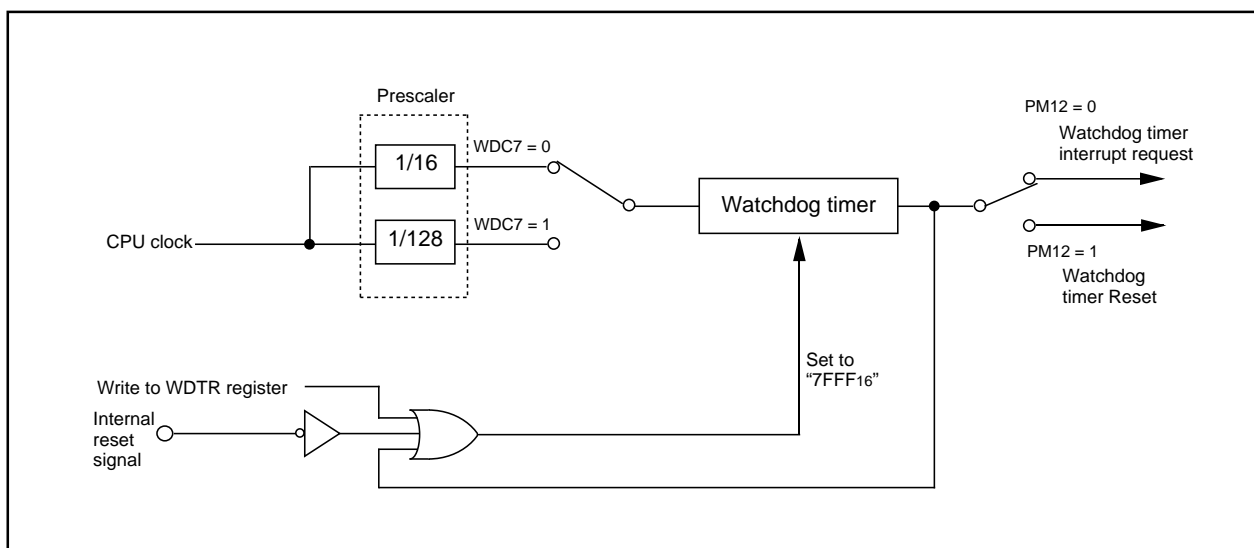


Figure 11.1 Watchdog Timer Block Diagram

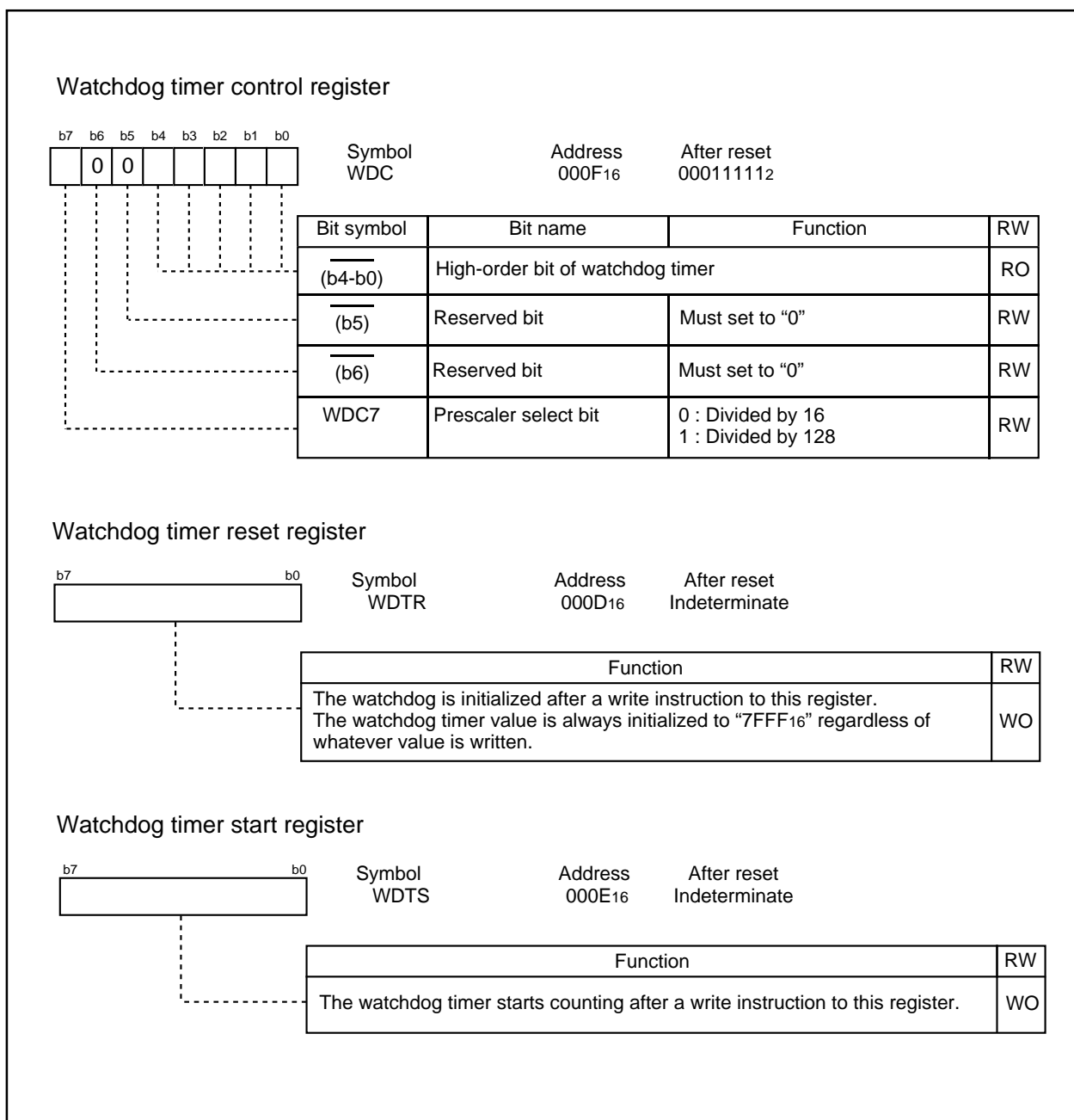


Figure 11.2 WDC Register, WDTR Register, and WDTS Register

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	<ul style="list-style-type: none"> •f1 •f2 •f8 •f32 	<ul style="list-style-type: none"> •f1 •f8 •fRING •Input from CNTR1 pin 	<ul style="list-style-type: none"> •f1 •f2 •f8 •Timer Y underflow 	<ul style="list-style-type: none"> •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	INT0	TCIN	
Output pin	CNTR0 CNTR0	CNTR1	TZOUT	not provided	
Related interrupt	Timer X int INT1 int	Timer Y int INT2 int	Timer Z int INT0 int	Timer C int INT3 int	
Timer stop	provided	provided	provided	provided	

NOTES:

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

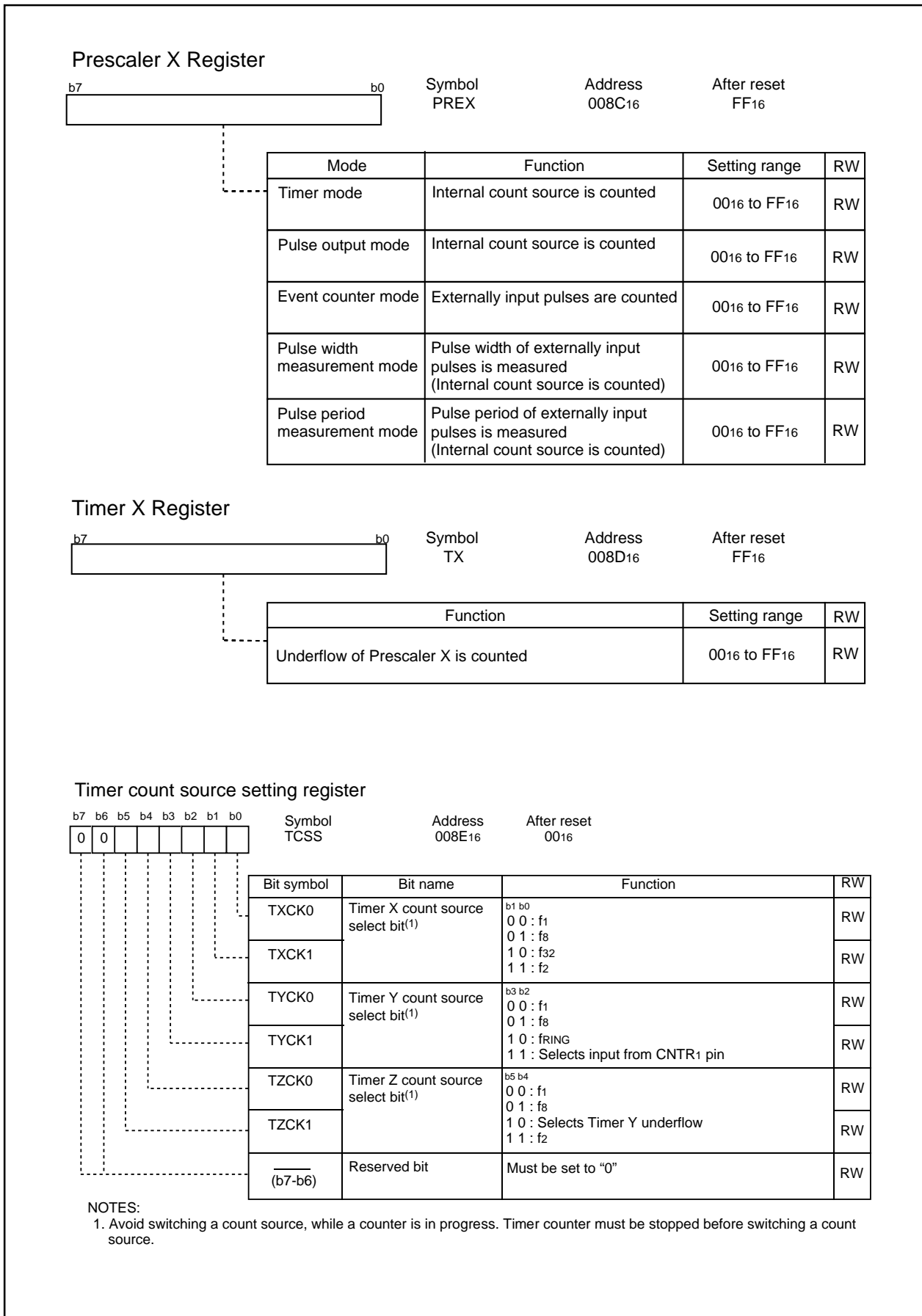


Figure 12.3 PREX Register, TX Register, and TCSS Register

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	<ul style="list-style-type: none"> Down-count When the timer underflows, the contents in the reload register is reloaded and the count is continued
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]
INT1/CNTR0 pin function	Programmable I/O port, or INT1 interrupt 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.

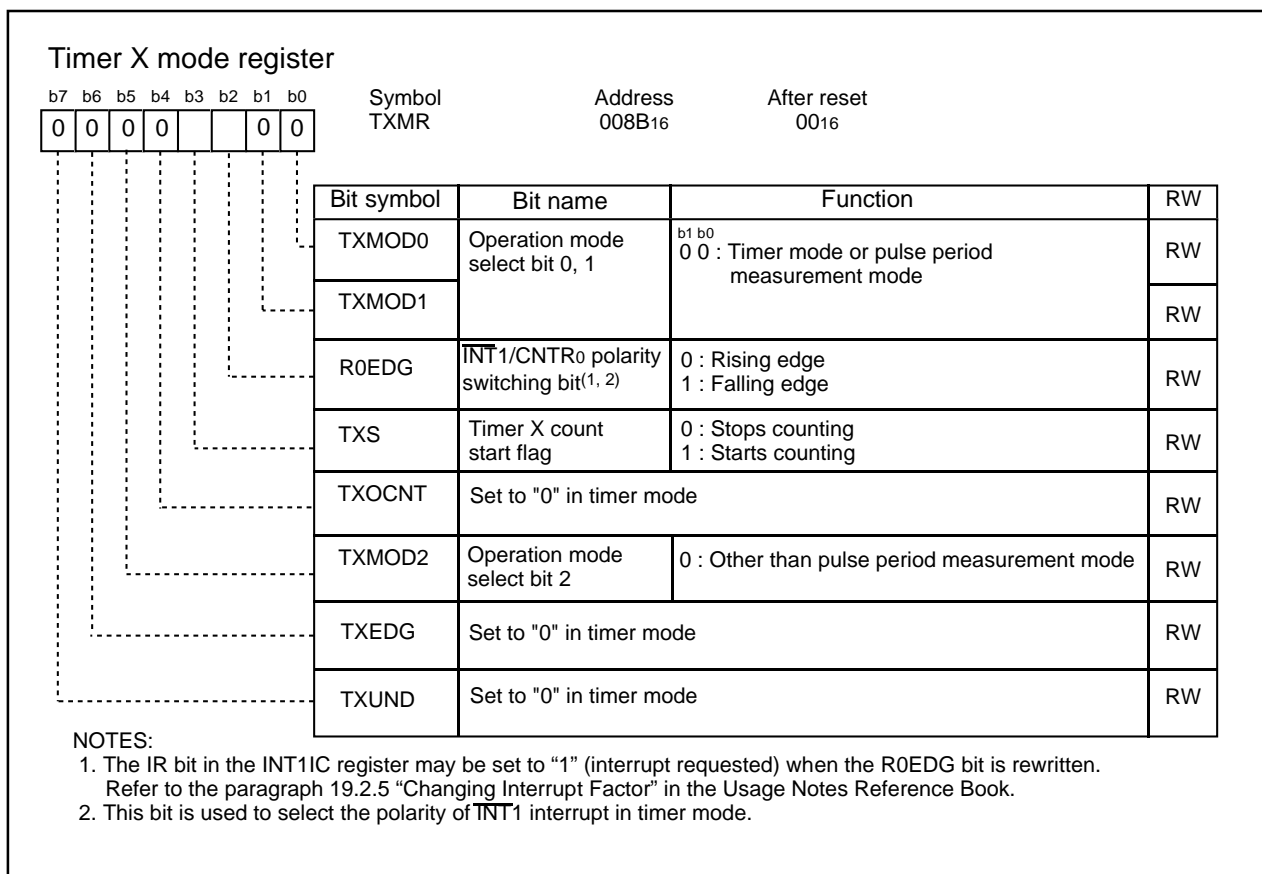


Figure 12.4 TXMR Register in Timer Mode

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	f1, f2, f8, f32
Count operation	<ul style="list-style-type: none"> Down-count When the timer underflows, the contents in the reload register is reloaded and the count is continued
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	<ul style="list-style-type: none"> When Timer X underflows [Timer X interruption]
INT1/CNTR0 pin function	Pulse output
CNTR0 pin function	Programmable I/O port or inverted output of CNTR0
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	<ul style="list-style-type: none"> INT1/CNTR0 polarity switching function Polarity level at starting of pulse output can be selected with R0EDG bit⁽¹⁾ Inverted pulse output function The inverted pulse of CNTR0 output polarity can be output from the $\overline{\text{CNTR0}}$ pin (selected by the TXOCNT bit)

NOTES:

- The level of the output pulse becomes the level when the pulse output starts when the TX register is written to.

Timer X mode register															
b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After reset					
0	0	0				0	1	TXMR	008B ₁₆	00 ₁₆					
TXMOD0		TXMOD1		R0EDG		TXS		TXOCNT		TXMOD2		TXEDG		TXUND	
Bit symbol		Bit name		Function		RW									
TXMOD0		Operation mode select bit 0, 1		b1 b0 0 1 : Pulse output mode		RW									
TXMOD1						RW									
R0EDG		INT1/CNTR0 polarity switching bit ⁽¹⁾		0: CNTR0 output starts at "H" 1: CNTR0 output starts at "L"		RW									
TXS		Timer X count start flag		0 : Stops counting 1 : Starts counting		RW									
TXOCNT		P30/ $\overline{\text{CNTR0}}$ select bit		0 : Port P30 1 : $\overline{\text{CNTR0}}$ output		RW									
TXMOD2		Set to "0" in pulse output mode				RW									
TXEDG		Set to "0" in pulse output mode				RW									
TXUND		Set to "0" in pulse output mode				RW									

NOTES:

- 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 Factor" 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 $\overline{\text{INT1/CNTR0}}$ 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	<ul style="list-style-type: none"> Down count When the timer underflows, the contents in the reload register is reloaded and the count is continued
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	<ul style="list-style-type: none"> When Timer X underflows [Timer X interrupt]
INT1/CNTR0 pin function	Count source input (INT1 interrupt 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	<ul style="list-style-type: none"> INT1/CNTR0 polarity switching function Active edge of count source can be selected with R0EDG.

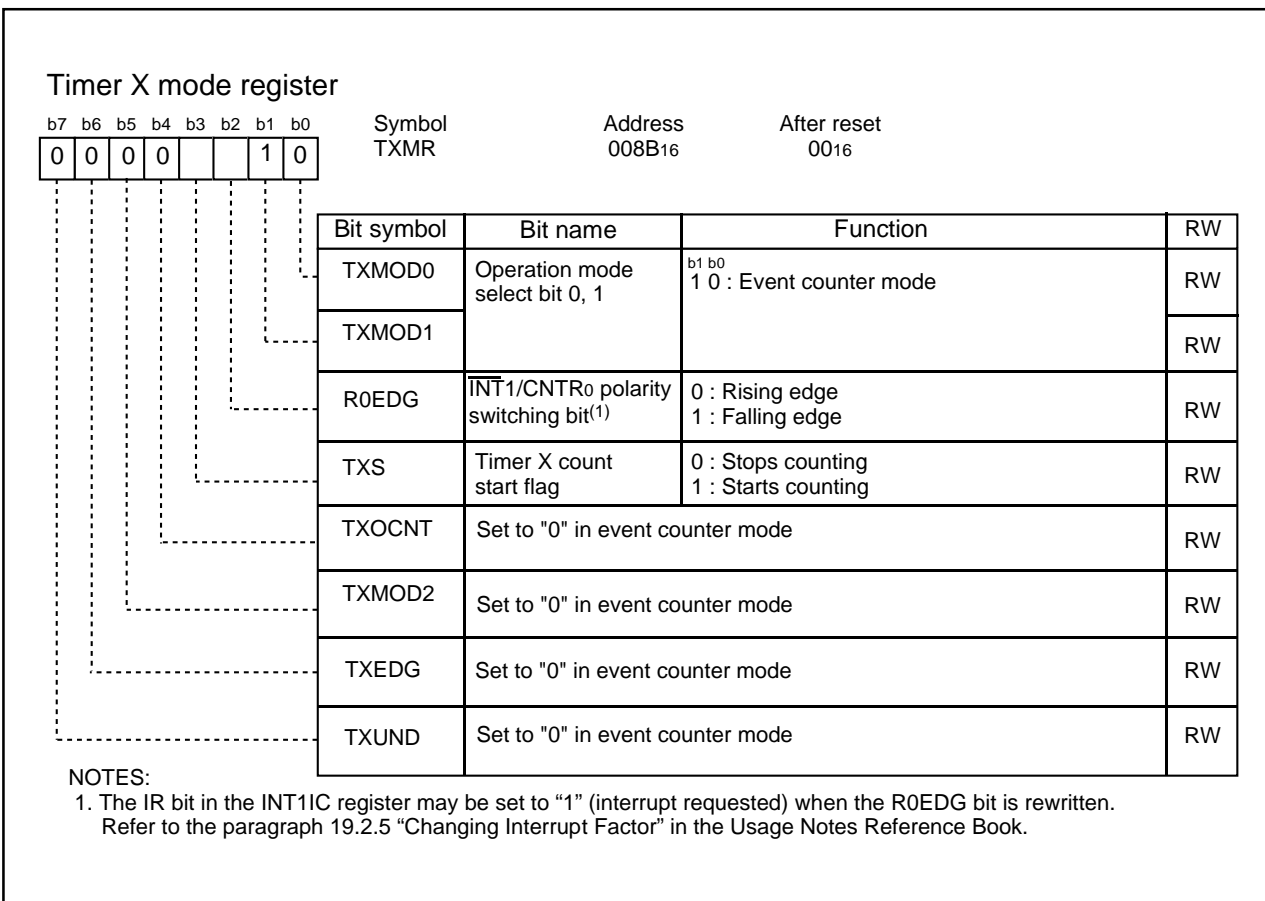


Figure 12.6 TXMR Register in Event Counter Mode

12.1.4 Pulse Width Measurement Mode

In this mode, the timer measures the pulse width of an external signal fed to $\overline{\text{INT1}}/\text{CNTR0}$ 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	f1, f2, f8, f32
Count operation	<ul style="list-style-type: none"> Down-count Continuously counts the selected signal only when the measurement pulse is "H" level, or conversely only "L" level. When the timer underflows, the contents in the reload register is reloaded and the count is continued
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	<ul style="list-style-type: none"> When Timer X underflows [Timer X interruption] Rising or falling of CNTR0 input (end of measurement period) [$\overline{\text{INT1}}$ interrupt]
$\overline{\text{INT1}}/\text{CNTR0}$ pin function	Measurement pulse 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	<ul style="list-style-type: none"> $\overline{\text{INT1}}/\text{CNTR0}$ polarity switching function "H" or "L" level duration can be selected with R0EDG bit as the input pulse measurement

Timer X mode register		Symbol	Address	After reset							
b7	b6	b5	b4	b3	b2	b1	b0	TXMR	008B16	0016	
0	0	0	0			1	1				
								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	$\overline{\text{INT1}}/\text{CNTR0}$ polarity switching bit ⁽¹⁾	[CNTR0] 0 : Measures "H" level width 1 : Measures "L" level width [INT1] 0 : Rising edge 1 : Falling edge	RW
								TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
								TXOCNT	Set to "0" in pulse width measurement mode		RW
								TXMOD2	Set to "0" in pulse width measurement mode		RW
								TXEDG	Set to "0" in pulse width measurement mode		RW
								TXUND	Set to "0" in pulse width measurement mode		RW

NOTES:
1. If 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 Factor" in the Usage Notes Reference Book.

Figure 12.7 TXMR Register in Pulse Width Measurement Mode

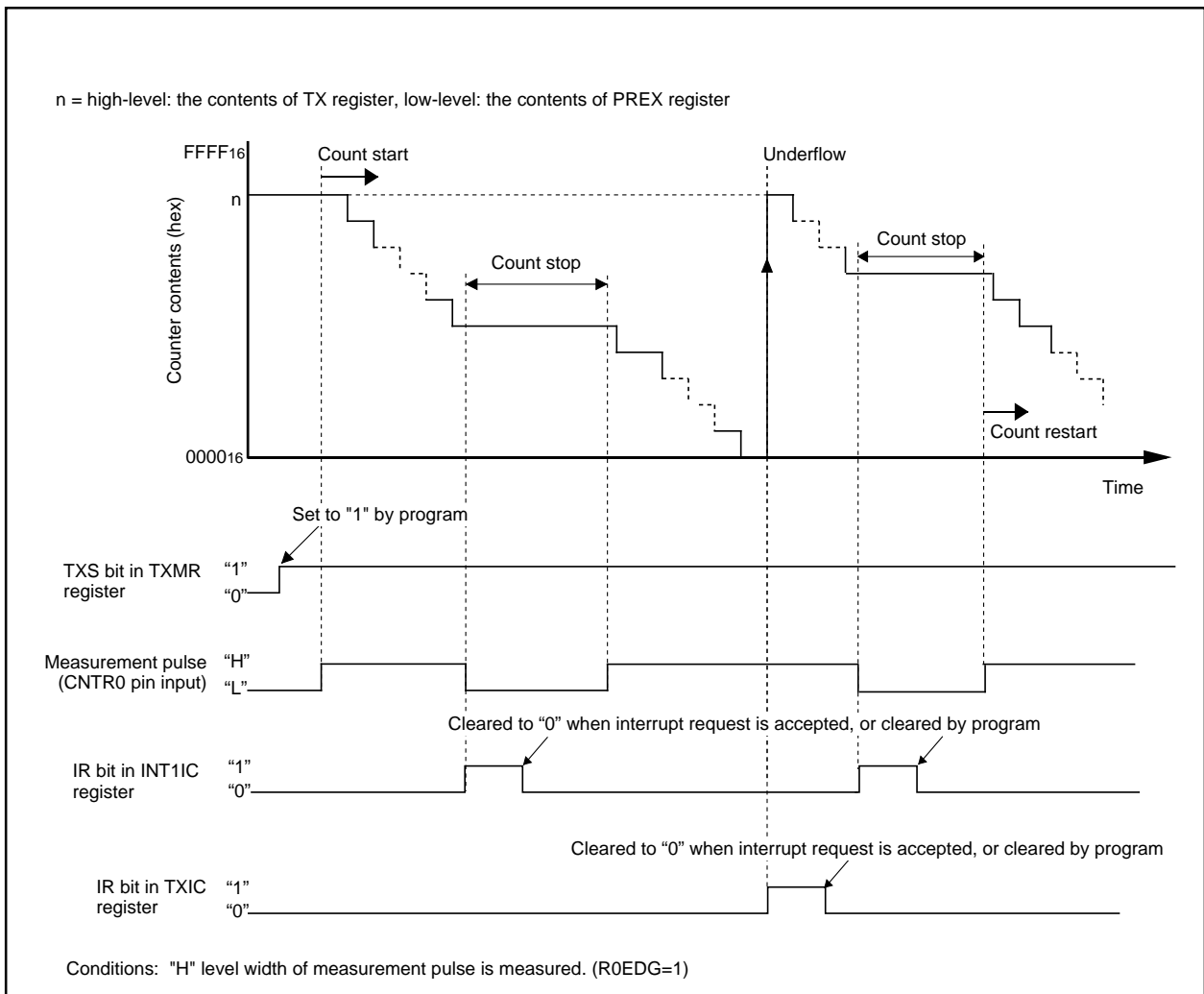


Figure 12.8 Operation Example in Pulse Width Measurement Mode

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	f1, f2, f8, f32
Count operation	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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⁾ ($\overline{\text{INT1}}$ interrupt input)
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	<ul style="list-style-type: none"> $\overline{\text{INT1/CNTR0}}$ polarity switching function Measurement period of input pulse can be selected with R0EDG bit.

NOTES:

- 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.

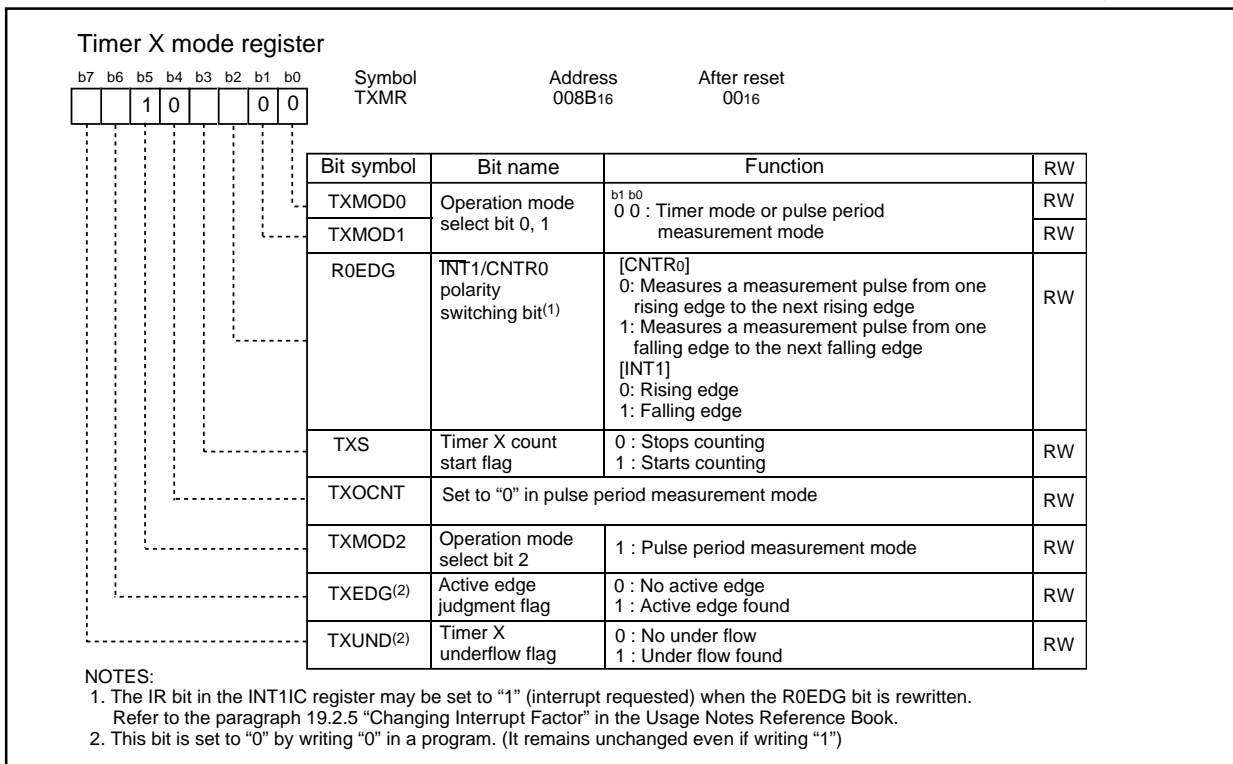


Figure 12.9 TXMR Register in Pulse Period Measurement Mode

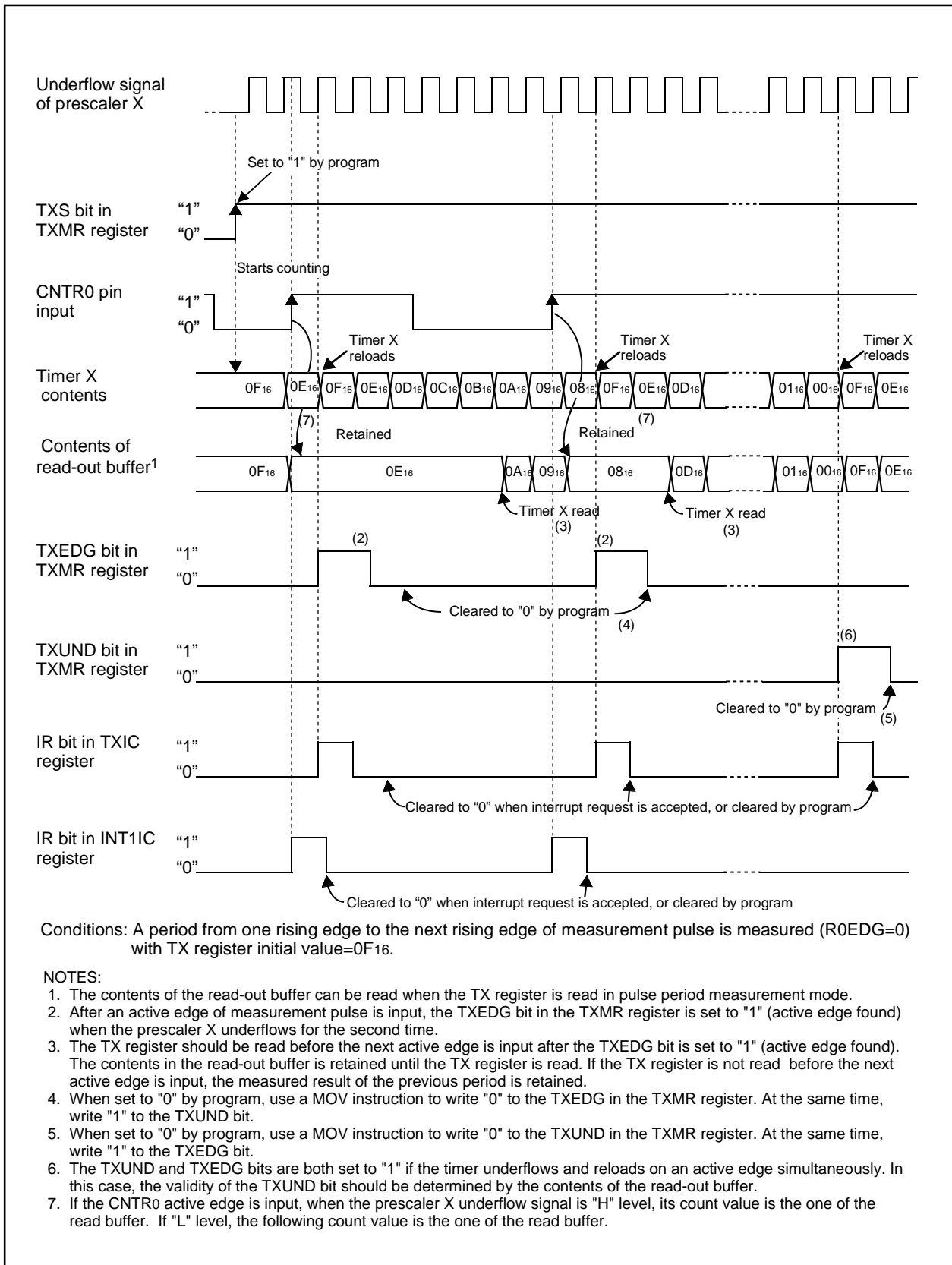


Figure 12.10 Operation Example in Pulse Period Measurement Mode

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.
- Programmable waveform generation mode: The timer outputs pulses of a given width successively.

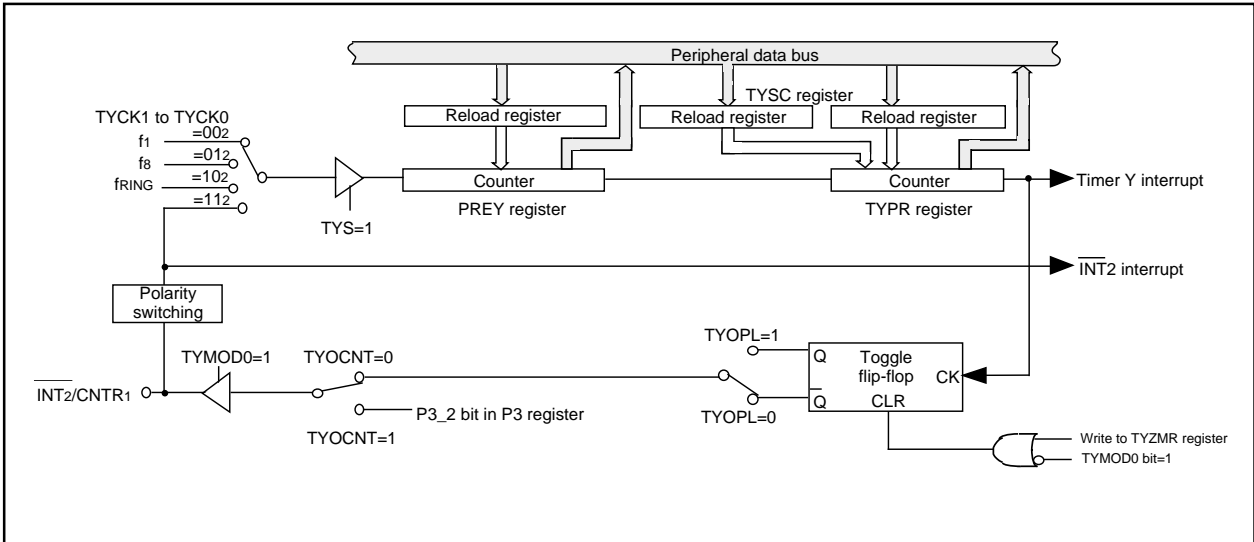


Figure 12.11 Timer Y Block Diagram

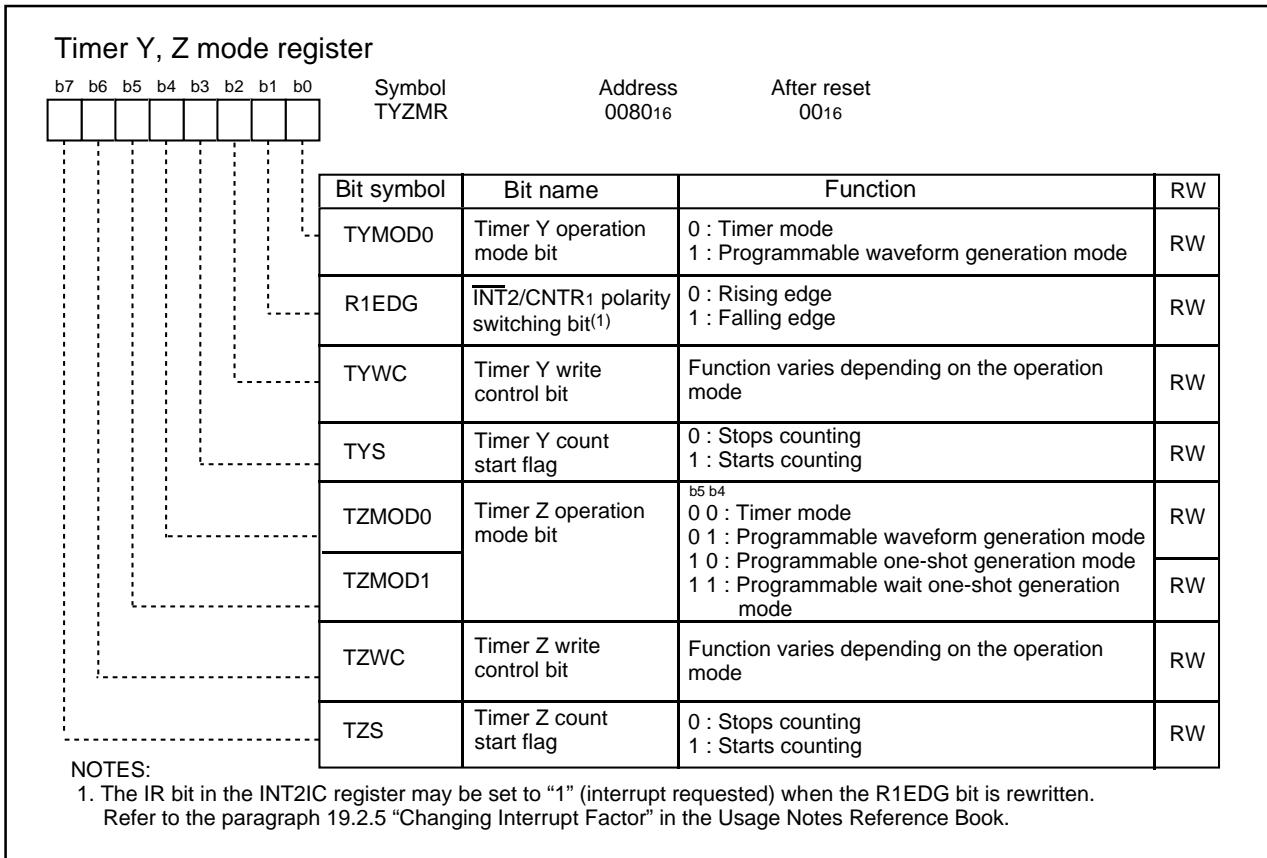


Figure 12.12 TYZMR Register

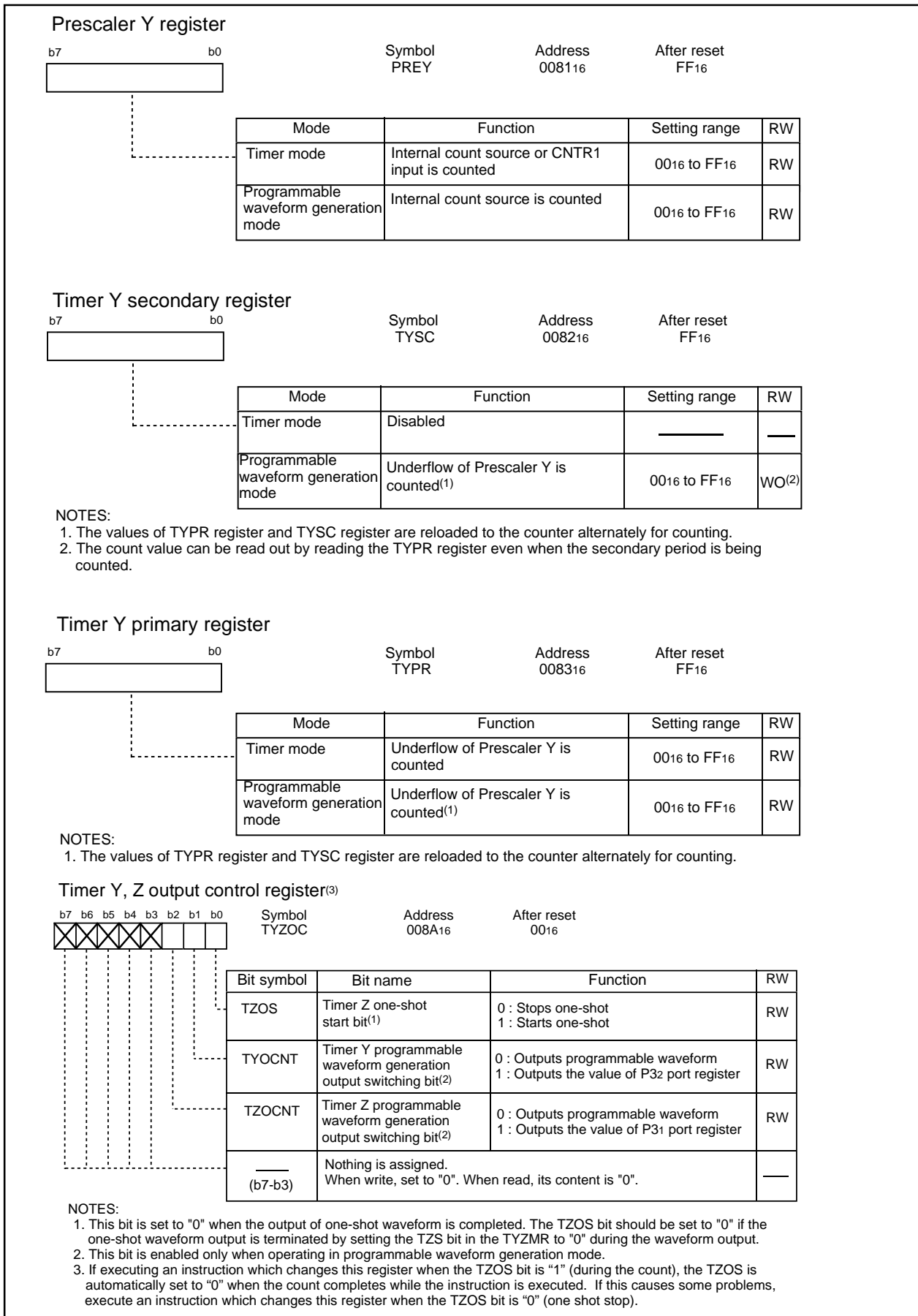


Figure 12.13 PREY Register, TYSC Register, TYPR Register, and TYZOC Register

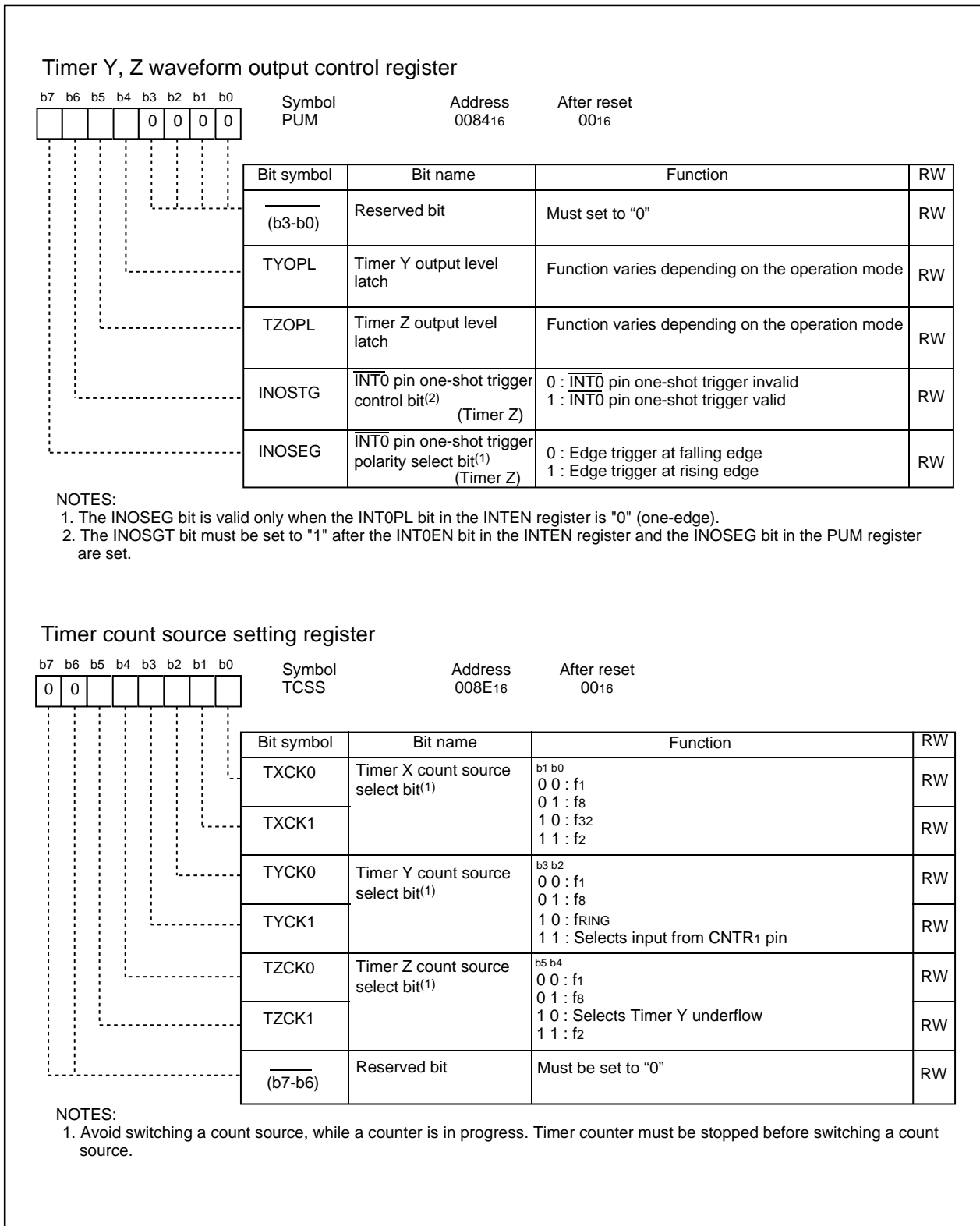


Figure 12.14 PUM Register and TCSS Register

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	f1, f8, fRING, external signal fed to CNTR1 pin
Count operation	<ul style="list-style-type: none"> • 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	$1/(n+1)(m+1)$ 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	<ul style="list-style-type: none"> • When Timer Y underflows [Timer Y interrupt]
INT2/CNTR1 pin function	Programmable I/O port, count source input or $\overline{\text{INT2}}$ interrupt input <ul style="list-style-type: none"> • When the TYCK1 to TYCK0 bits in the TCSS register are set to “00b”, “01b” or “10b” (Timer Y count source is f1, f8 or fRING), programmable I/O port or $\overline{\text{INT2}}$ interrupt input • When the TYCK1 to TYCK0 bits are set to “11b” (Timer Y count source is CNTR1 input), count source input ($\overline{\text{INT2}}$ interrupt input)
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	<ul style="list-style-type: none"> • Event counter function When setting TYCK1 to TYCK0 bits to “112”, an external signal fed to CNTR1 pin is counted. • 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.

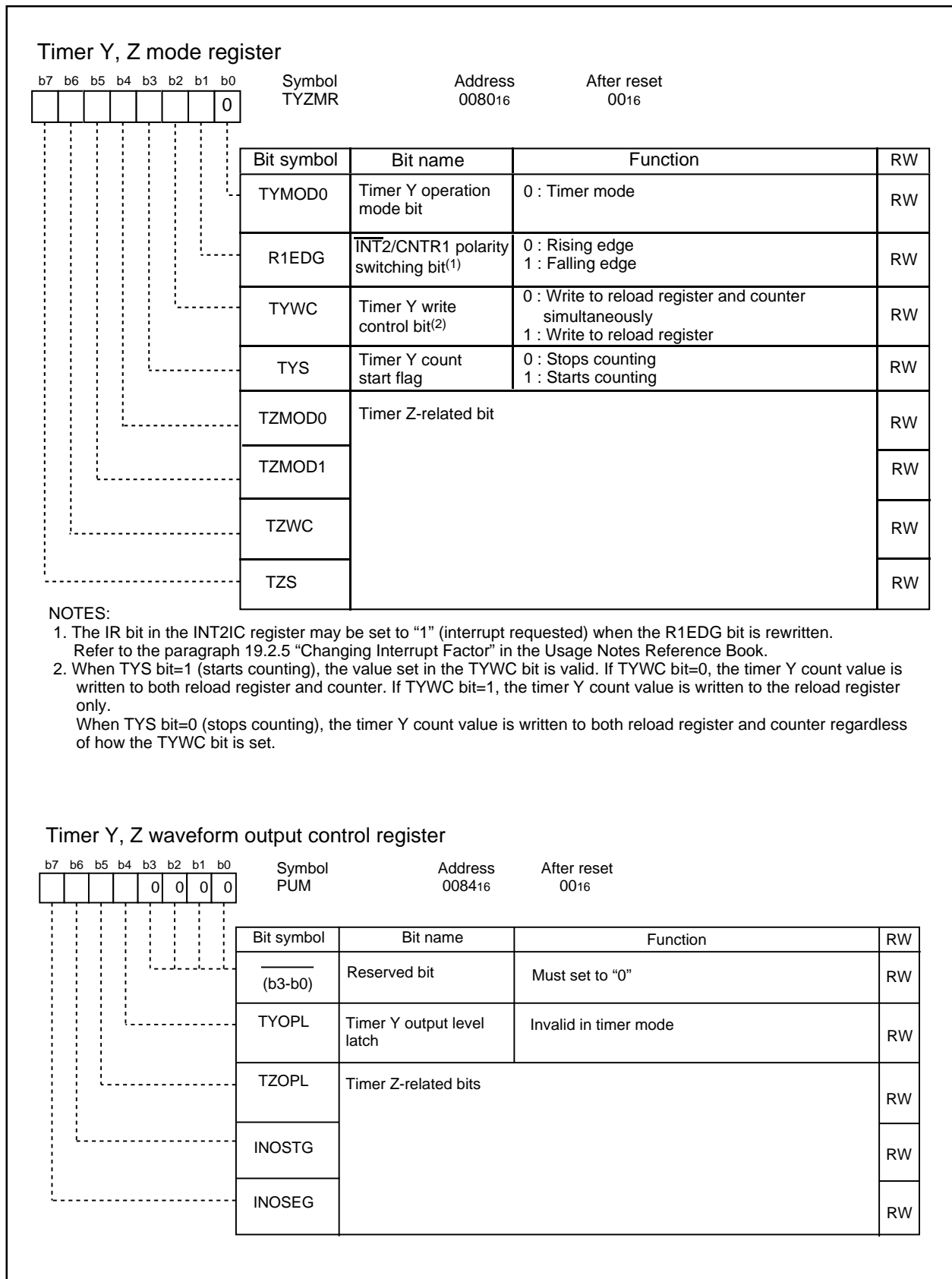


Figure 12.15 TYZMR Register and PUM Register in Timer Mode

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	<ul style="list-style-type: none"> Down count When the timer underflows, it reloads the contents of primary reload register and secondary reload register alternately before continuing counting.
Output waveform width and period	Primary period : $(n+1)(m+1)/f_i$ Secondary period : $(n+1)(p+1)/f_i$ Period : $(n+1)\{(m+1)+(p+1)\}/f_i$ n: set value in PREY register, m: set value in TYPR register, p: set value in TYSC register f _i : Count source frequency
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]
INT2/CNTR1 pin functions	Pulse output Use timer mode when using this pin as a programmable I/O port.
Read from timer	Count value can be read out by reading TYPR register. Same applies to PREY register ⁽¹⁾ .
Write to timer	Value written to TYPR register is written to only reload register. Same applies to TYSC register and PREY register ⁽²⁾ .
Select function	<ul style="list-style-type: none"> 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⁽³⁾.

NOTES:

- Even when counting the secondary period, read out the TYPR register.
- 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.
- The TYOCNTbit is enabled in the following timings
 - When count starts
 - When Timer Y interrupt request is generated

Therefore, pulse is output from the next primary period depending on the setting value of the TYOCNT bit.

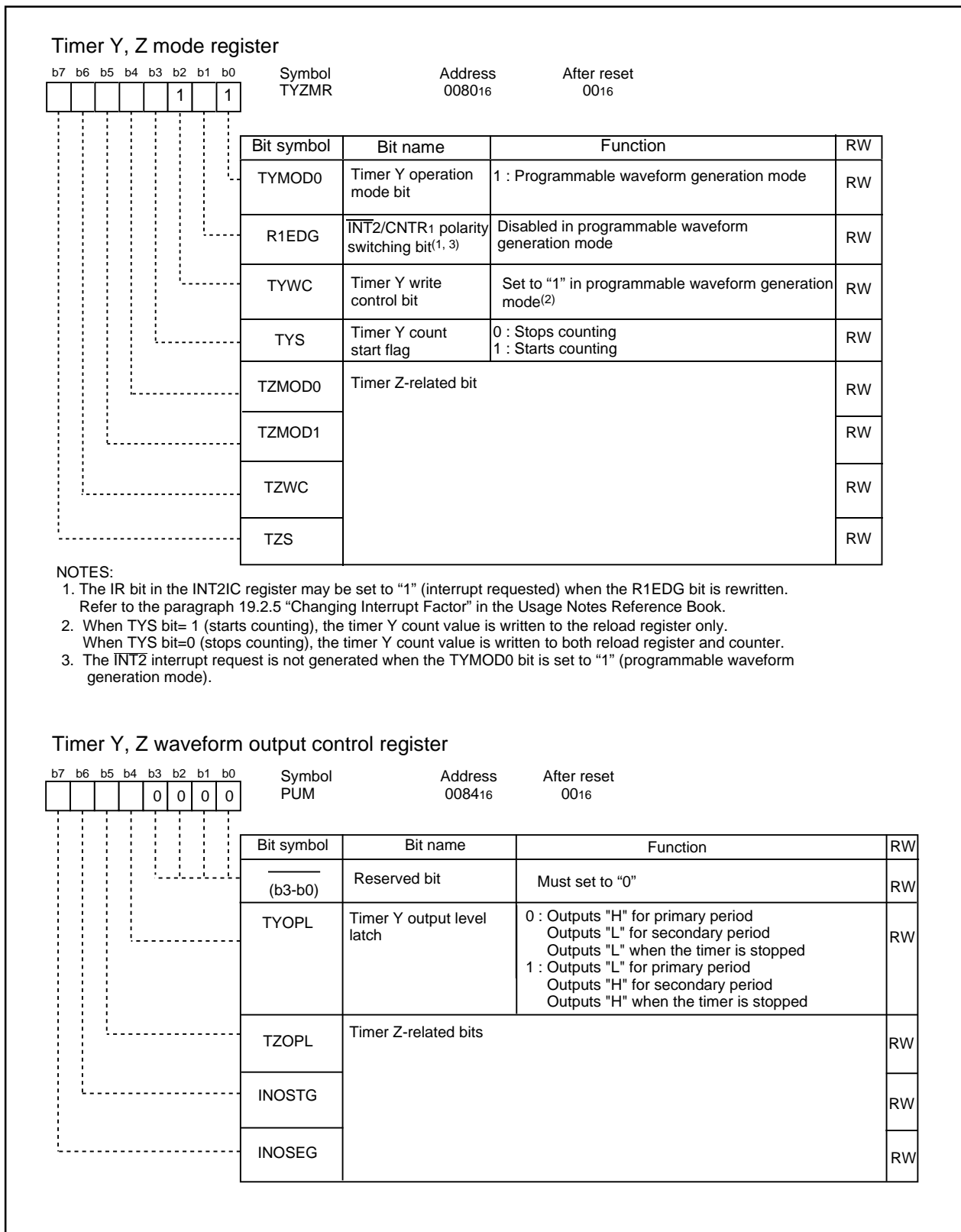


Figure 12.16 TYZMR Register and PUM Register in Programmable Waveform Generation Mode

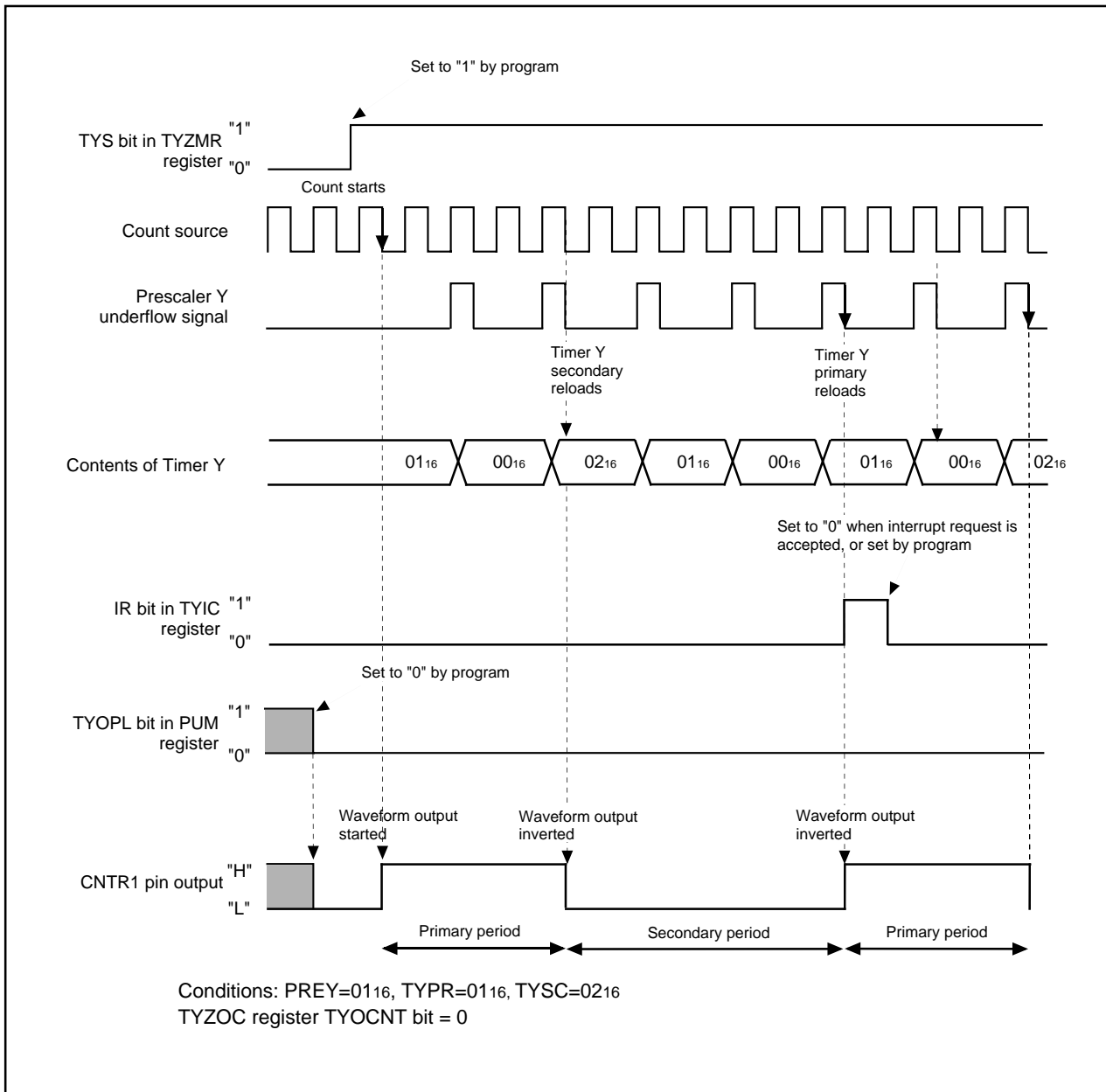


Figure 12.17 Timer Y Operation Example in Programmable Waveform Generation Mode

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 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.

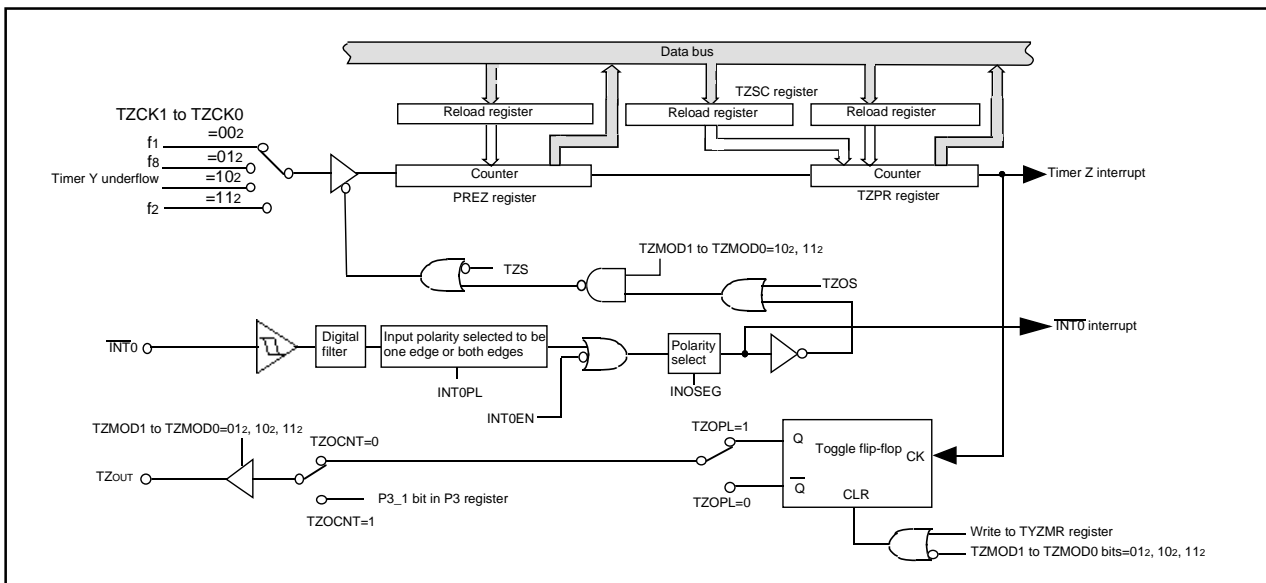


Figure 12.18 Timer Z Block Diagram

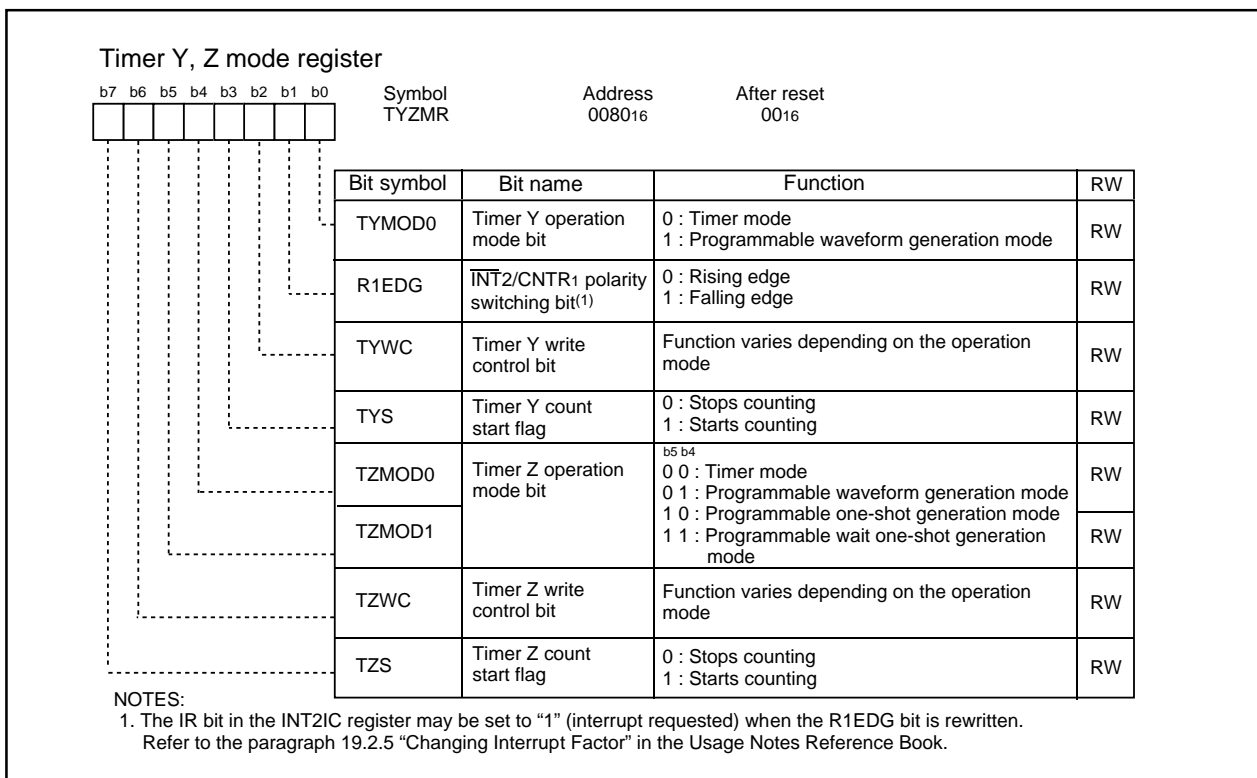


Figure 12.19 TYZMR Register

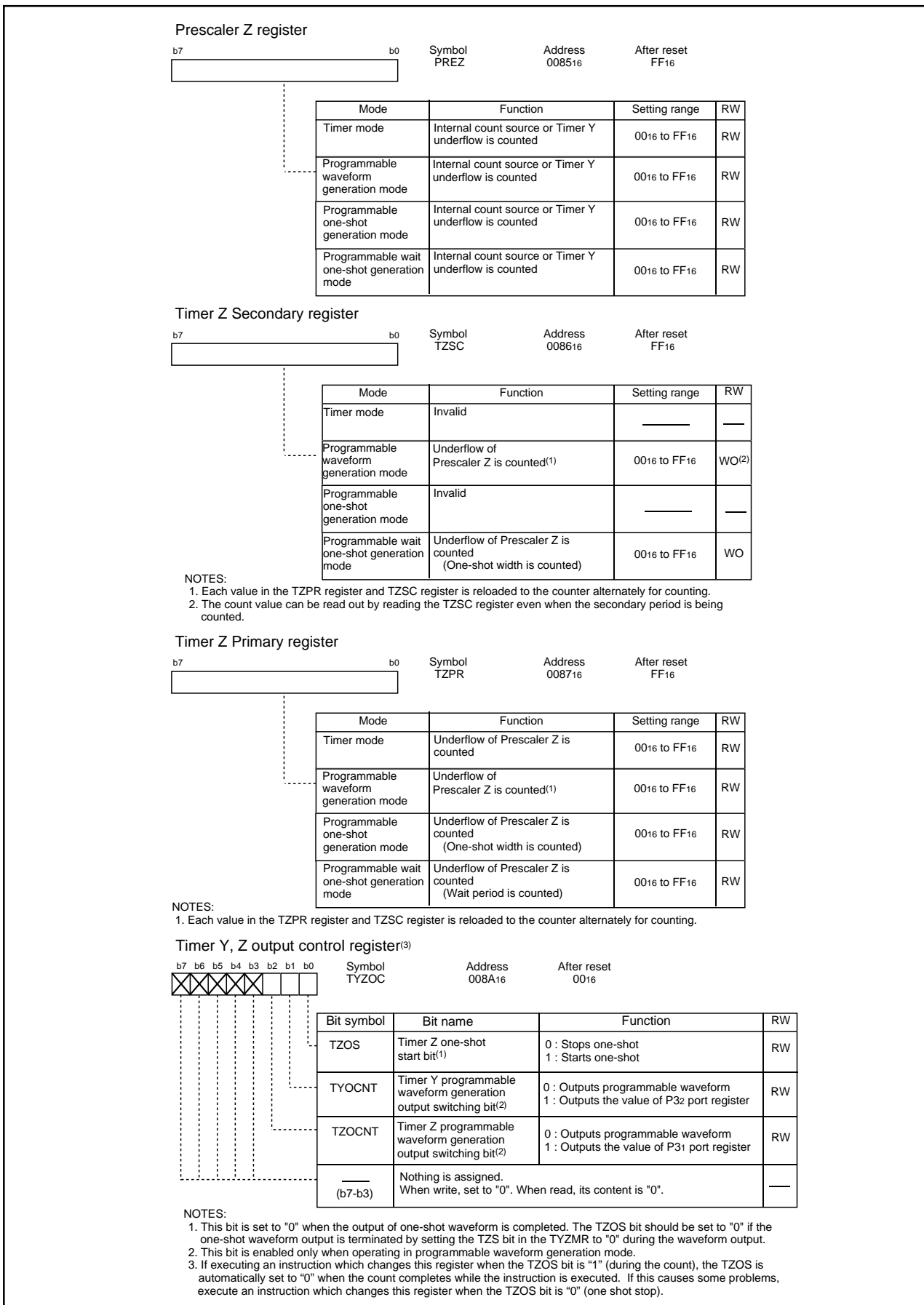


Figure 12.20 PREZ Register, TZSC Register, TZPR Register, and TYZOC Register

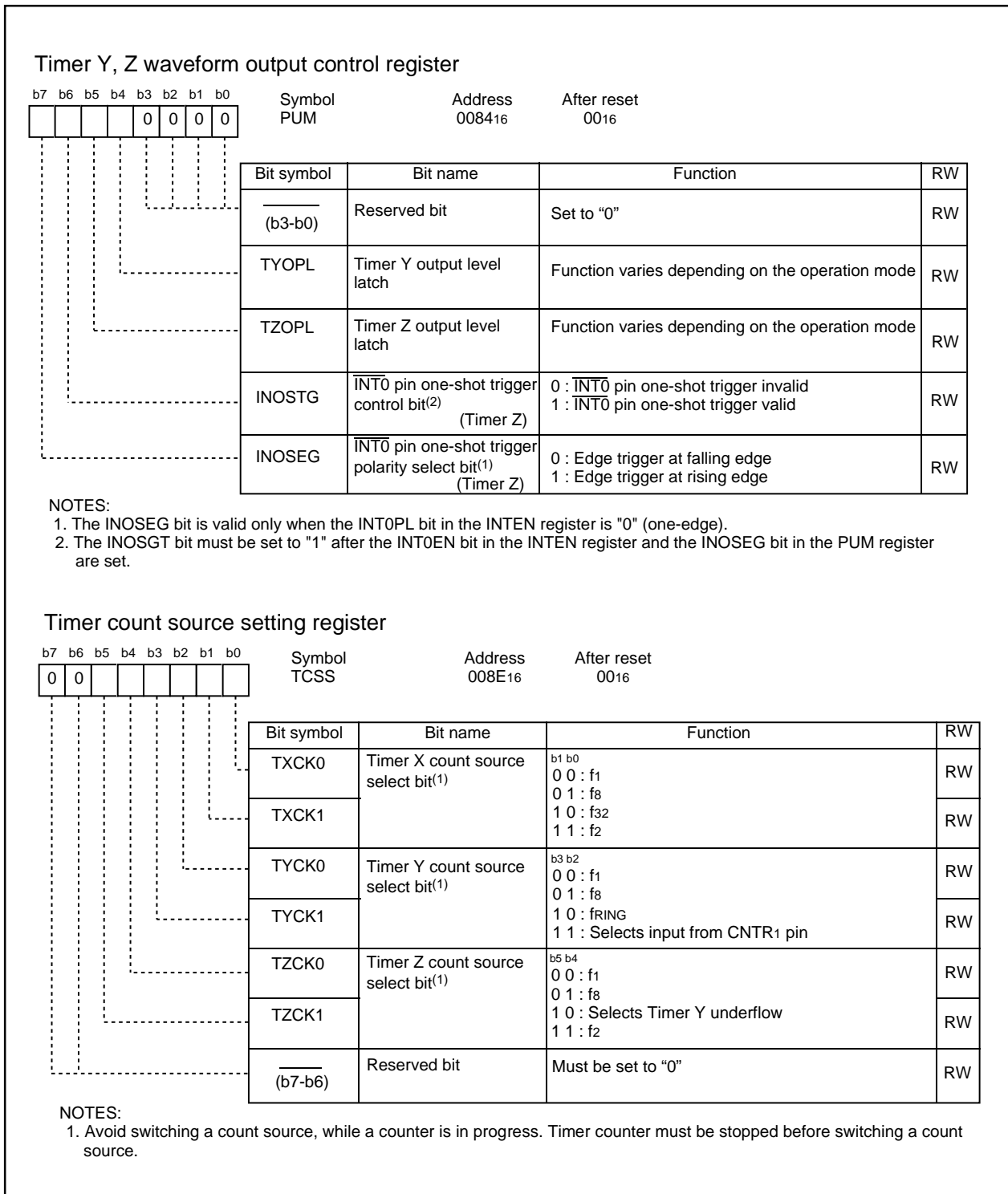


Figure 12.21 PUM Register and TCSS Register

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	f1, f2, f8, Timer Y underflow
Count operation	<ul style="list-style-type: none"> • 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	$1/(n+1)(m+1)$ 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	<ul style="list-style-type: none"> • When Timer Z underflows [Timer Z interrupt]
TZOUT pin function	Programmable I/O port
INT0 pin function	Programmable I/O port, or INT0 interrupt input
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.

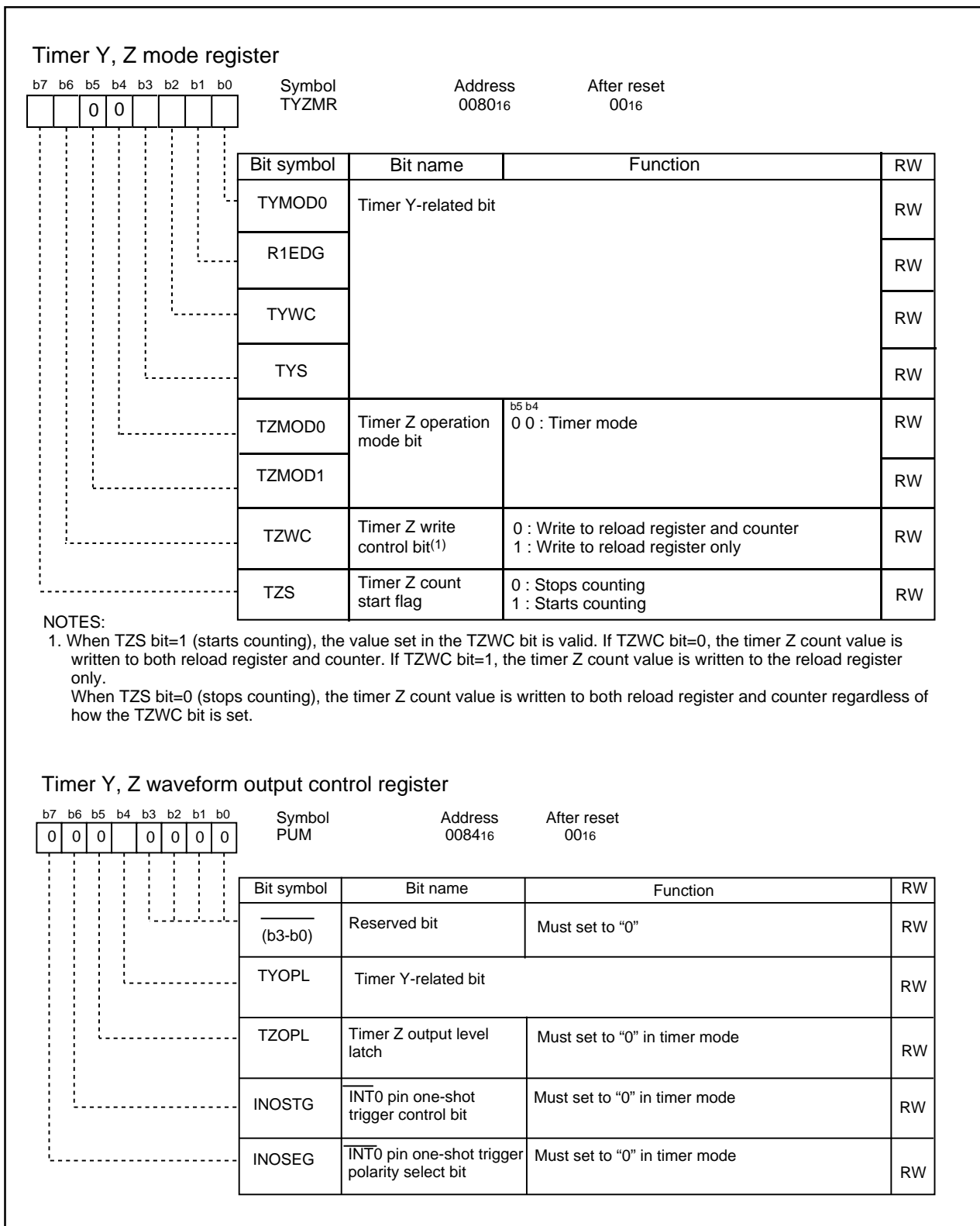


Figure 12.22 TYZMR Register and PUM Register in Timer Mode

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	f1, f2, f8, Timer Y underflow
Count operation	<ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the contents of primary reload register and secondary reload register alternately before continuing counting.
Output waveform width and period	Primary period : $(n+1)(m+1)/f_i$ Secondary period : $(n+1)(p+1)/f_i$ Period : $(n+1)\{(m+1)+(p+1)\}/f_i$ f_i : Count source frequency 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 Use timer mode when using this pin as a programmable I/O port.
INT0 pin functions	Programmable I/O port, or INT0 interrupt input
Read from timer	Count value can be read out by reading TZPR register. Same applies to PREZ register ⁽¹⁾ .
Write to timer	Value written to TZPR register is written to reload register only. Same applies to TZSC register and PREZ register ⁽²⁾ .
Select function	<ul style="list-style-type: none"> 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 The output from TZOUT is inverted synchronously when Timer Z underflows by setting the TZOCNT bit in the TYZOC register to “0”. A value in the P3_1 bit is output from the TZOUT by setting to “1”⁽³⁾.

NOTES:

- Even when counting the secondary period, read out the TZPR register.
- 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.
- The TZOCNTbit is enabled in the following timings
 - When count starts
 - When Timer Z interrupt request is generated
 Therefore, pulse is output from the next primary period depending on the setting value of the TZOCNT bit.

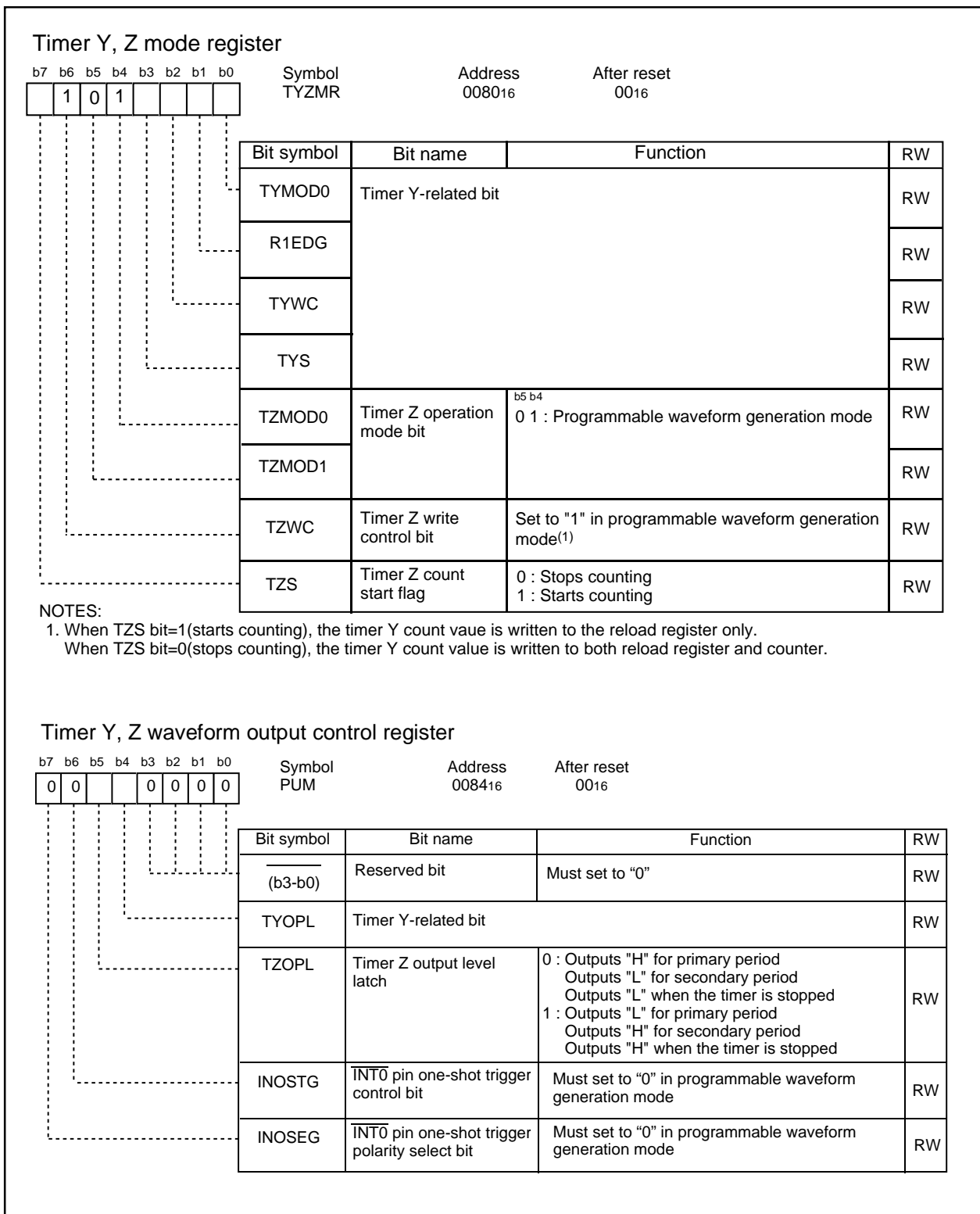


Figure 12.23 TYZMR Register and PUM Register in Programmable Waveform Generation Mode

12.3.3 Programmable One-shot Generation Mode

In this mode, upon program command or external trigger input (input to the $\overline{\text{INT0}}$ 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	f1, f2, f8, Timer Y underflow
Count operation	<ul style="list-style-type: none"> • Downcounts set value in TZPR register • When the timer underflows, it reloads the contents of reload register before completing counting and the TZOS bit is "0". • When a count stops, the timer reloads the contents of the reload register before it stops.
One-shot pulse output duration	$(n+1)(m+1)/f_i$ f_i : count source frequency, n: set value in PREZ register, m: set value in TZPR register
Count start condition	<ul style="list-style-type: none"> • Set TZOS bit in TYZOC register to "1" (start one-shot)⁽¹⁾ • Input active trigger to INT0 pin⁽²⁾
Count stop condition	<ul style="list-style-type: none"> • When reloading is completed after count value was set to "0016" • 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 Use timer mode when using this pin as a programmable I/O port.
$\overline{\text{INT0}}$ pin function	Programmable I/O port, $\overline{\text{INT0}}$ interrupt input or external trigger input <ul style="list-style-type: none"> • When the INOSTG bit in the PUM register is set to "0" ($\overline{\text{INT0}}$ one-shot trigger disabled) Programmable I/O port or $\overline{\text{INT0}}$ interrupt input • When the INOSTG bit in the PUM register is set to "1" ($\overline{\text{INT0}}$ one-shot trigger enabled) external trigger ($\overline{\text{INT0}}$ interrupt input)
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	<ul style="list-style-type: none"> • 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 INTOEN 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).
Although the trigger input during counting cannot be acknowledged, the $\overline{\text{INT0}}$ interrupt request is generated.
3. The set values are reflected beginning with the next one-shot pulse after writing to the TZPR register.

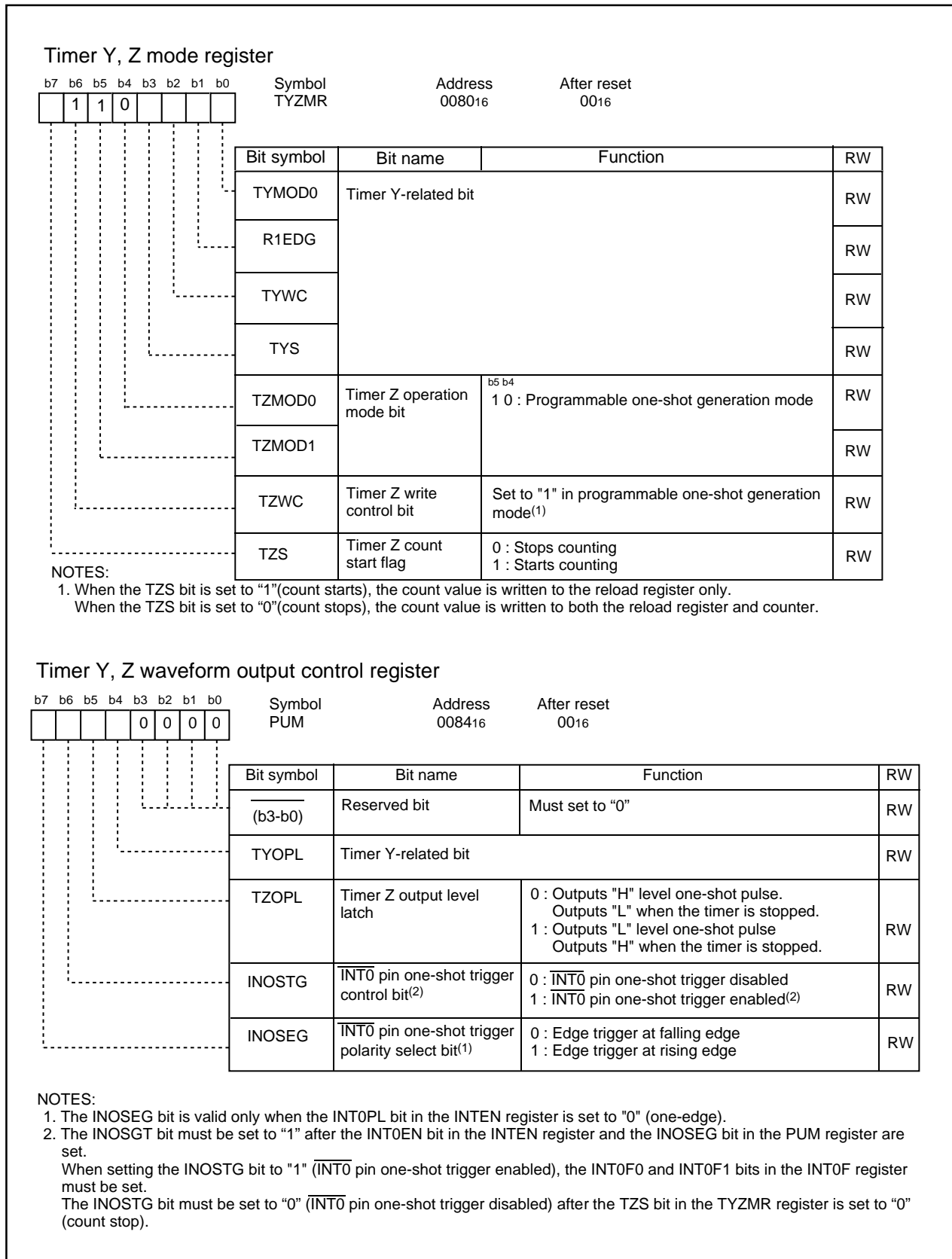


Figure 12.24 TYZMR Register and PUM Register in Programmable One-shot Generation Mode

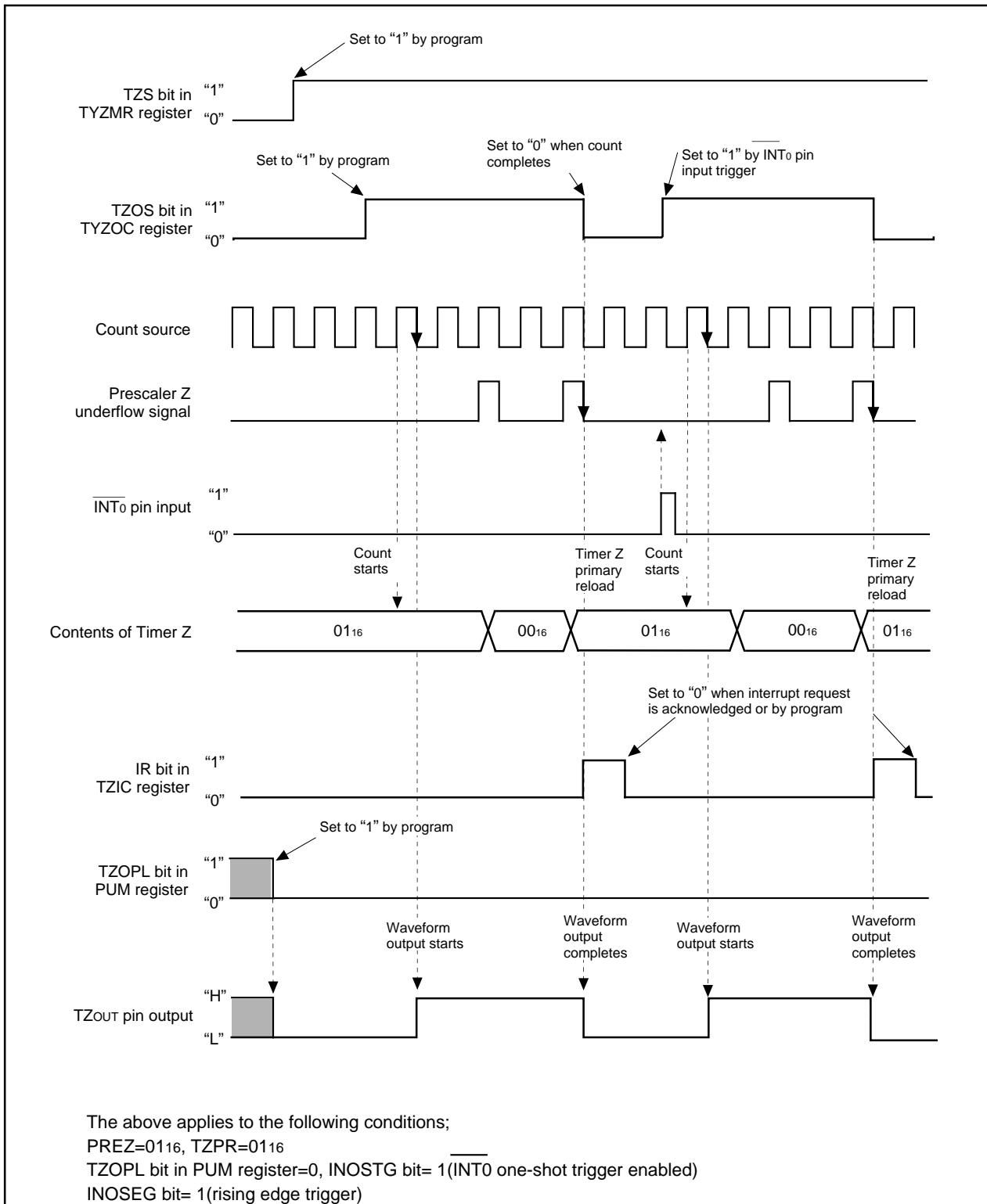


Figure 12.25 Operation Example in Programmable One-shot Generation Mode

12.3.4 Programmable Wait One-shot Generation Mode

In this mode, upon program or external trigger input (input to the 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	f1, f2, f8, Timer Y underflow
Count operation	<ul style="list-style-type: none"> • 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 completing counting and the TZOS bit is "0". • When a count stops, the timer reloads the contents of the reload register before it stops.
Wait time	$(n+1)(m+1)/f_i$ n: set value in PREZ register, m: set value in TZPR register
One-shot pulse output time	$(n+1)(p+1)/f_i$ n: set value in PREZ, p: set value in TZSC register
Count start condition	<ul style="list-style-type: none"> • Set TZOS bit in TYZOC register to "1" (start one-shot)⁽¹⁾ • Input active trigger to $\overline{\text{INT0}}$ pin⁽²⁾
Count stop condition	<ul style="list-style-type: none"> • When reloading is completed after count value at counting TZSC register was set to "0016" • 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 "0016" (at the same time as the TZout output change) [Timer Z interrupt]
TZOUT pin function	Pulse output Use timer mode when using this pin as a programmable I/O port.
INT0 pin function	Programmable I/O port, $\overline{\text{INT0}}$ interrupt input or external trigger input <ul style="list-style-type: none"> • When the INOSTG bit in the PUM register is set to "0" ($\overline{\text{INT0}}$ one-shot trigger disabled) Programmable I/O port or $\overline{\text{INT0}}$ interrupt input • When the INOSTG bit in the PUM register is set to "1" ($\overline{\text{INT0}}$ one-shot trigger enabled) external trigger ($\overline{\text{INT0}}$ interrupt input)
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	<ul style="list-style-type: none"> • 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 INT0 input), and the INOSTG bit in the PUM register to "1" (enabling INT0 one-shot trigger)
Although the trigger input during counting cannot be acknowledged, the $\overline{\text{INT0}}$ interrupt request is generated.
3. The set values are reflected beginning with the next one-shot pulse after writing to the TZPR register.

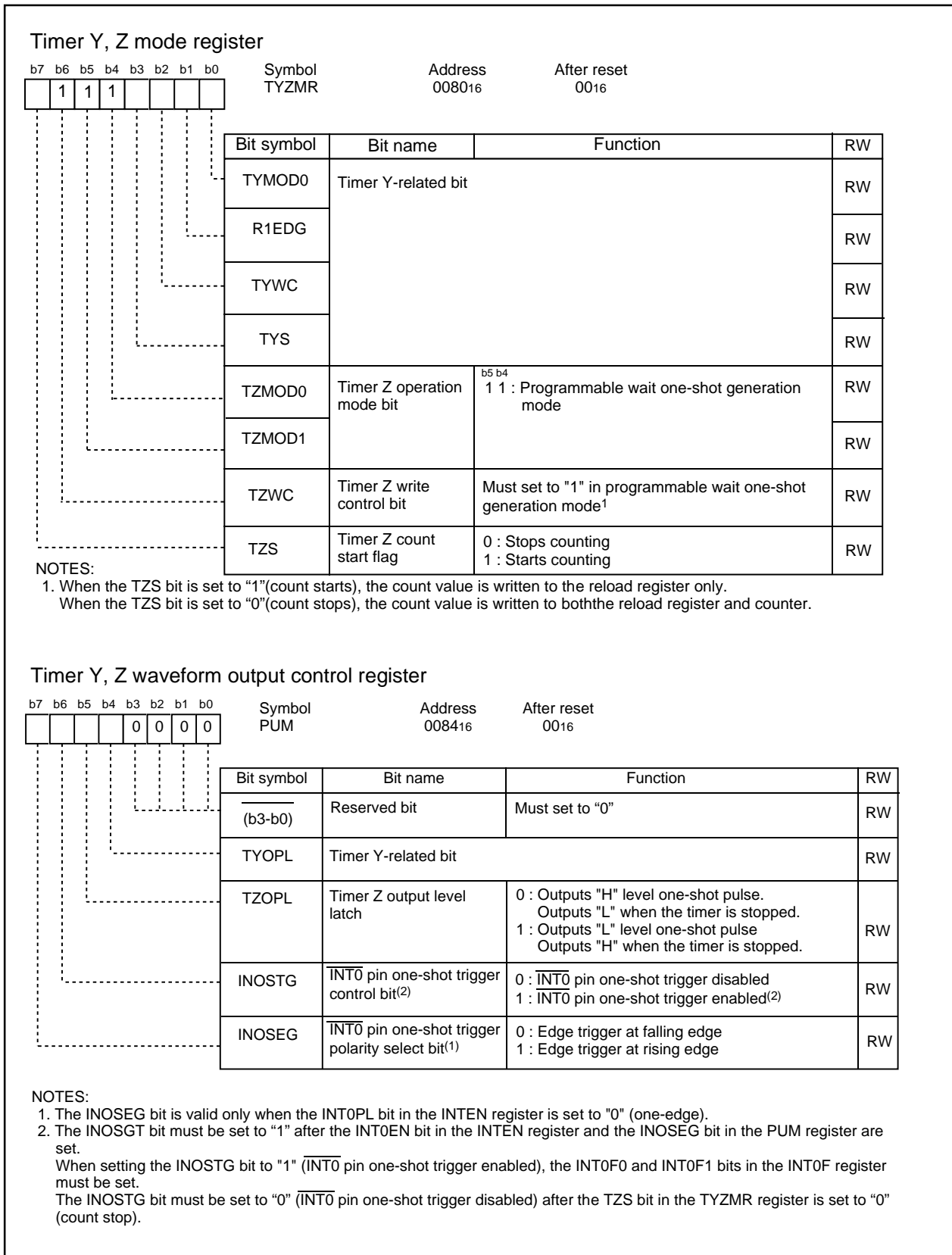


Figure 12.26 TYZMR Register and PUM Register in Programmable Wait One-shot Generation Mode

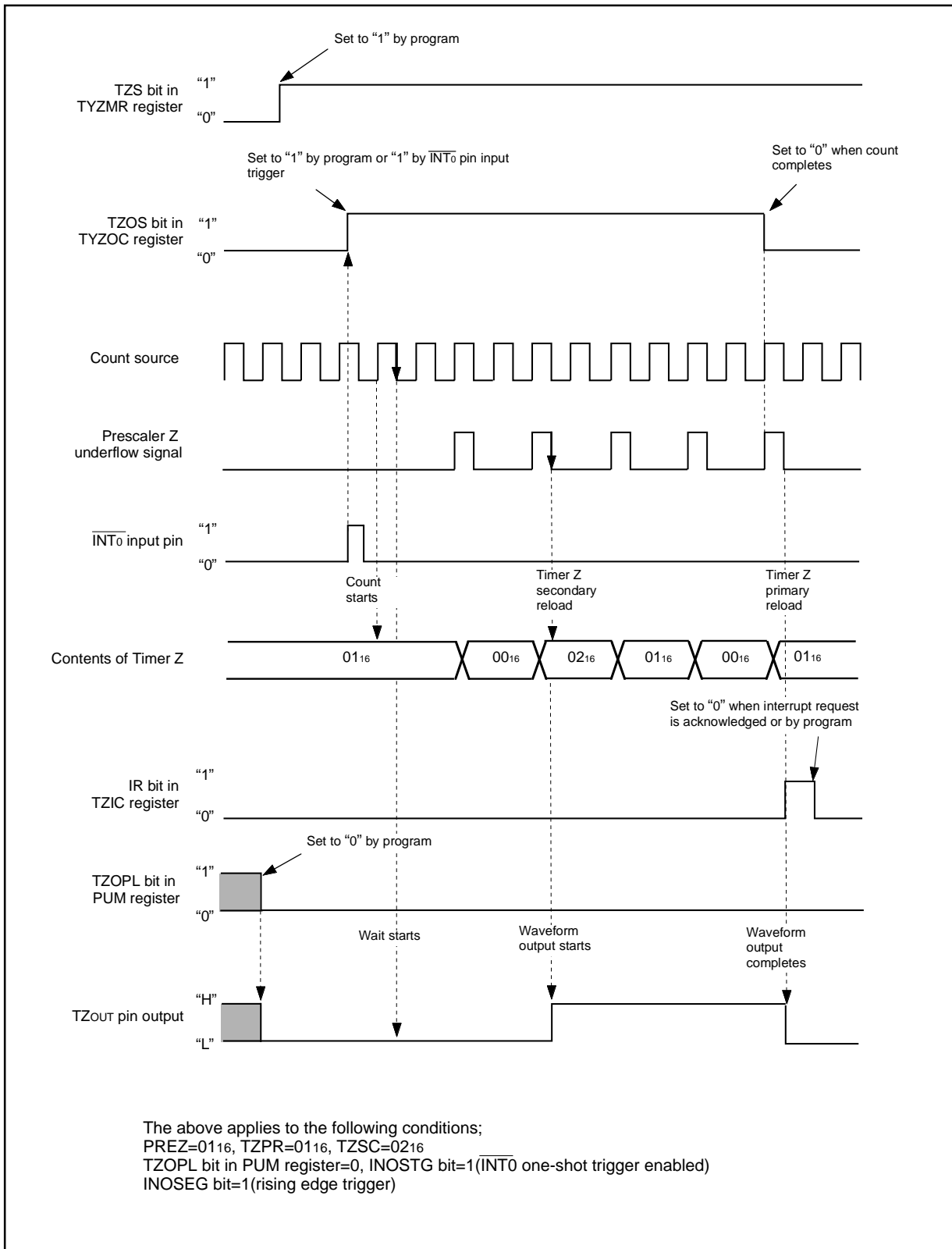


Figure 12.27 Operation Example in Programmable Wait One-shot Generation Mode

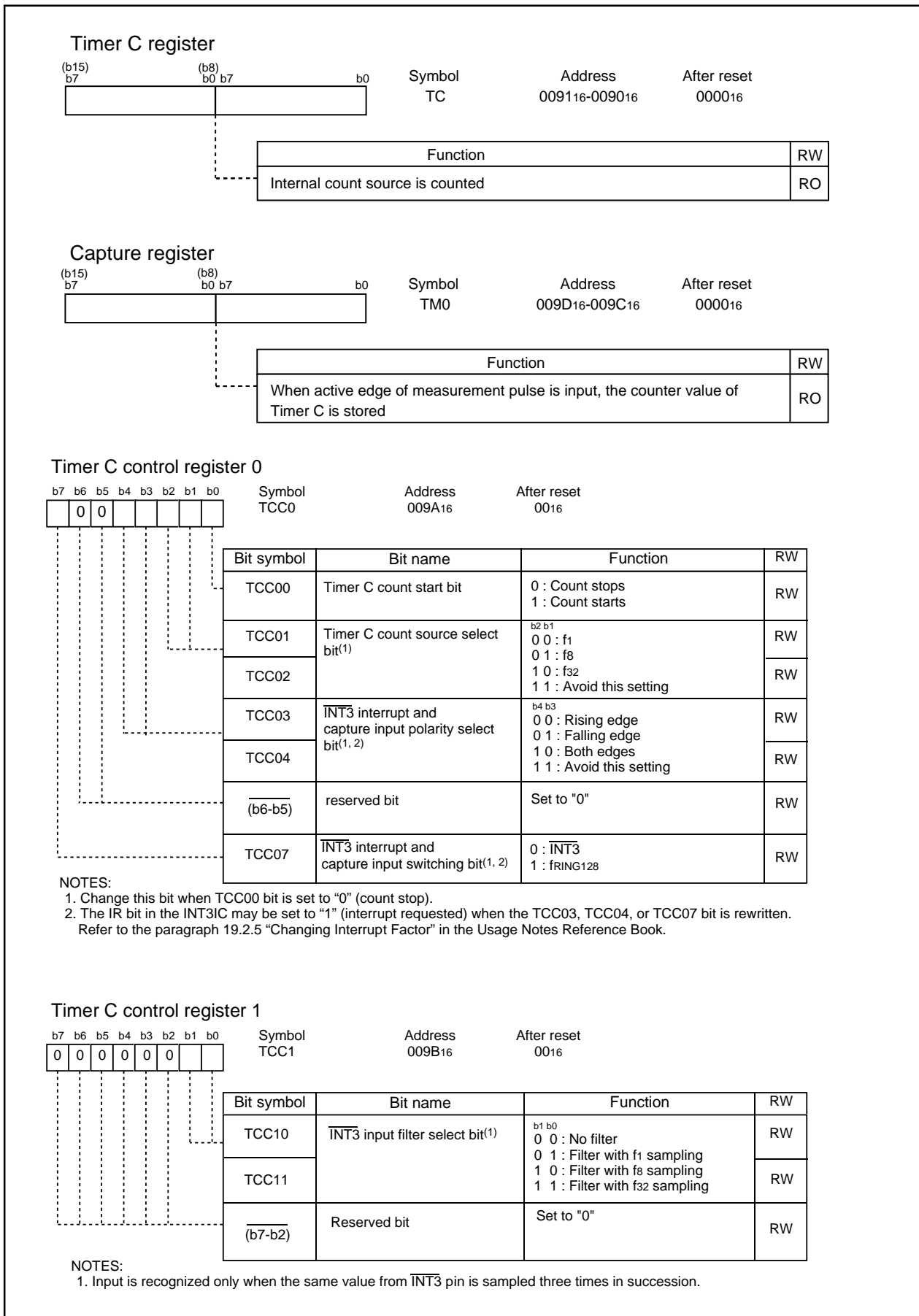


Figure 12.29 TC Register, TM0 Register, TCC0 Register, and TCC1 Register

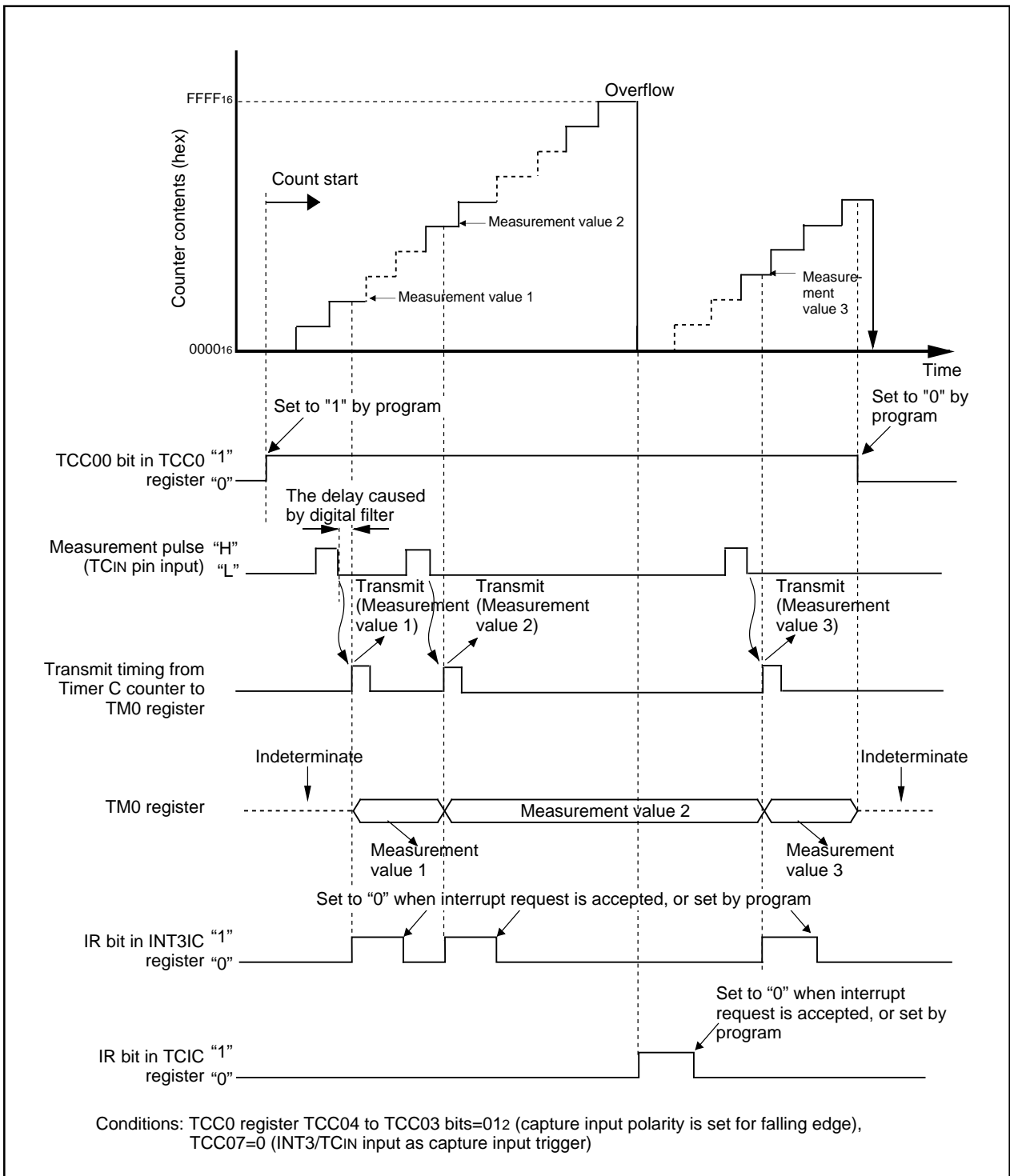


Figure 12.30 Operation Example of Timer C

13. Serial Interface

Serial interface 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.

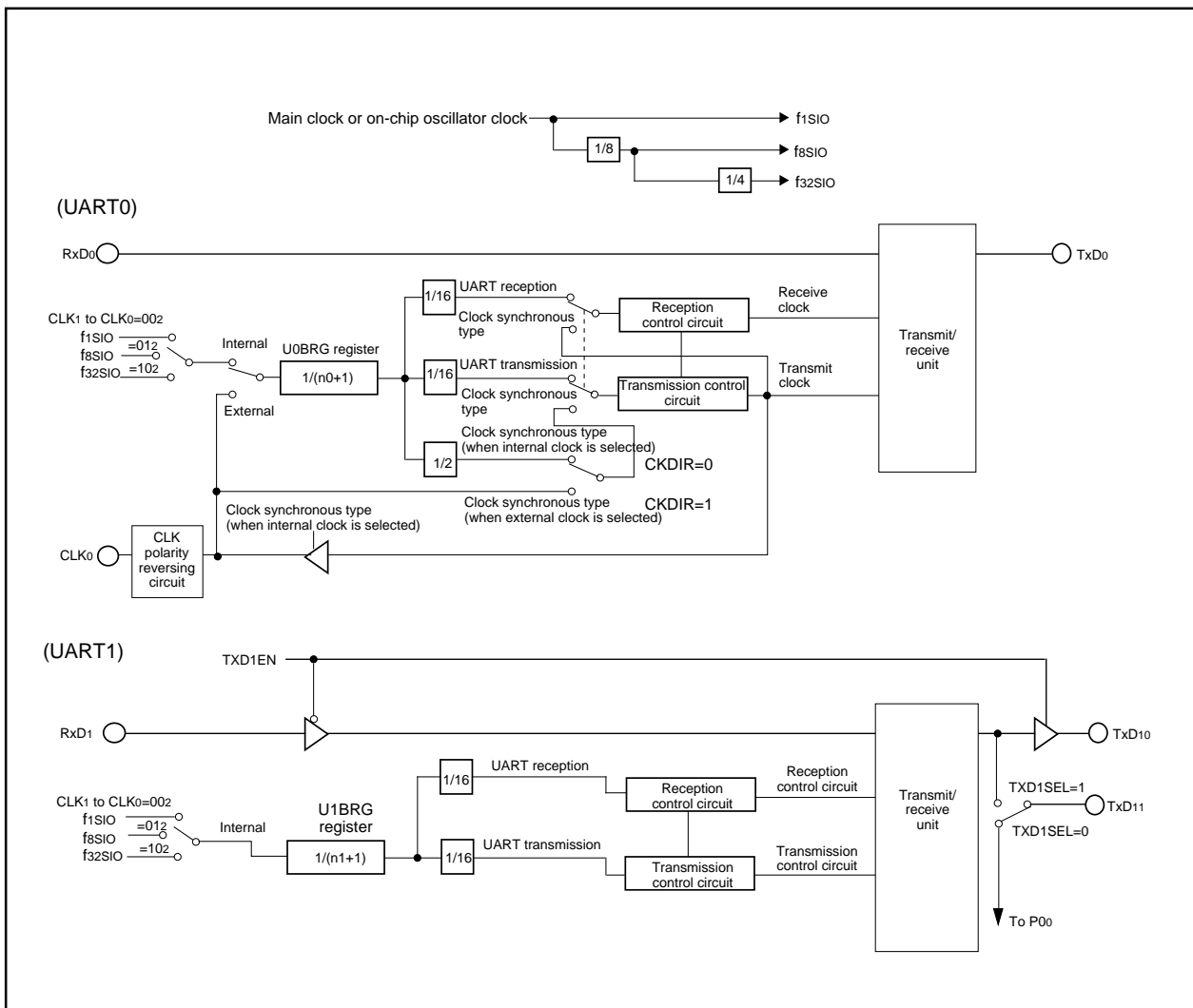


Figure 13.1 UARTi (i=0, 1) Block Diagram

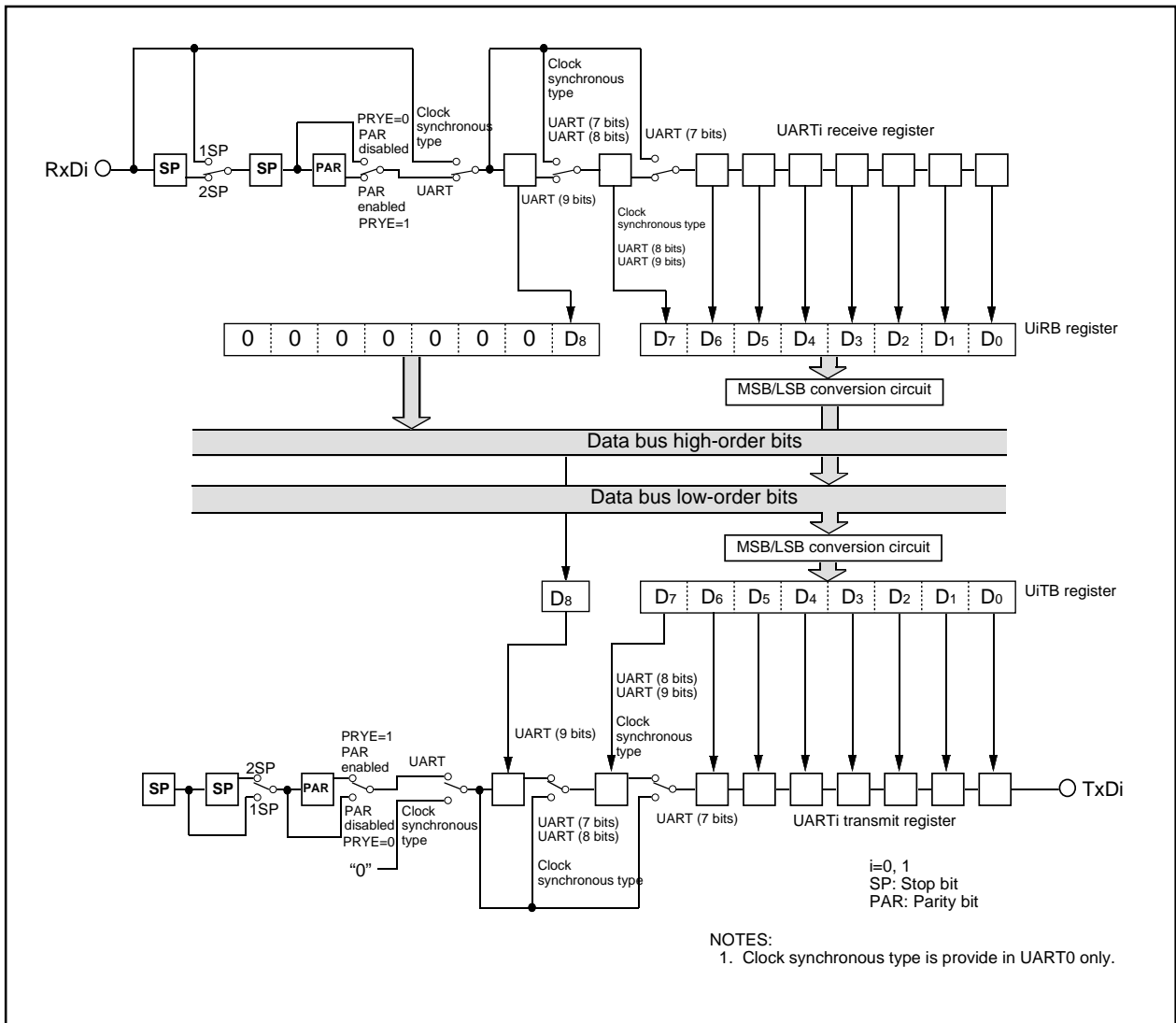


Figure 13.2 UARTi Transmit/Receive Unit

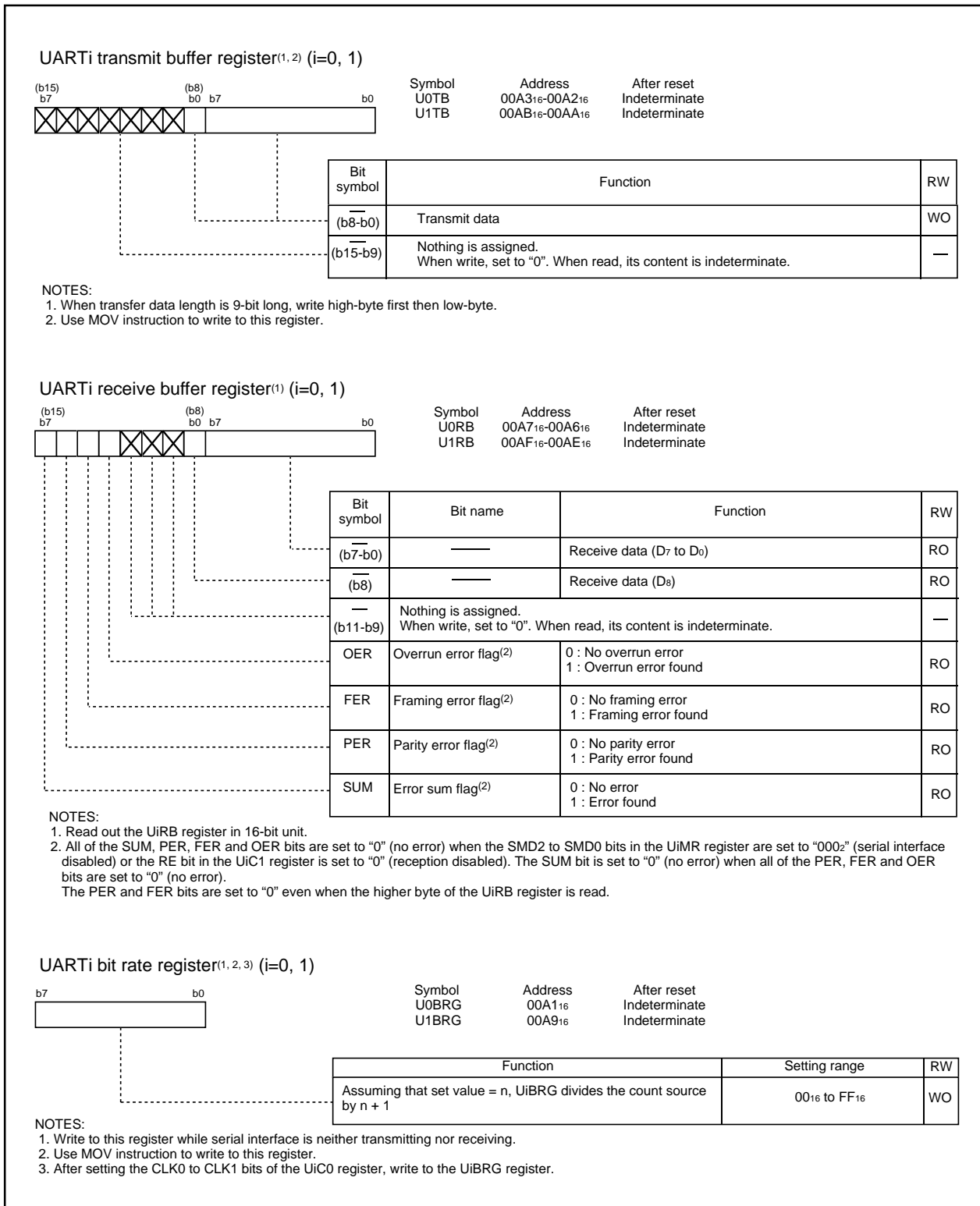


Figure 13.3 U0TB and U1TB Registers, U0RB and U1RB Registers, and U0BRG and U1BRG Registers

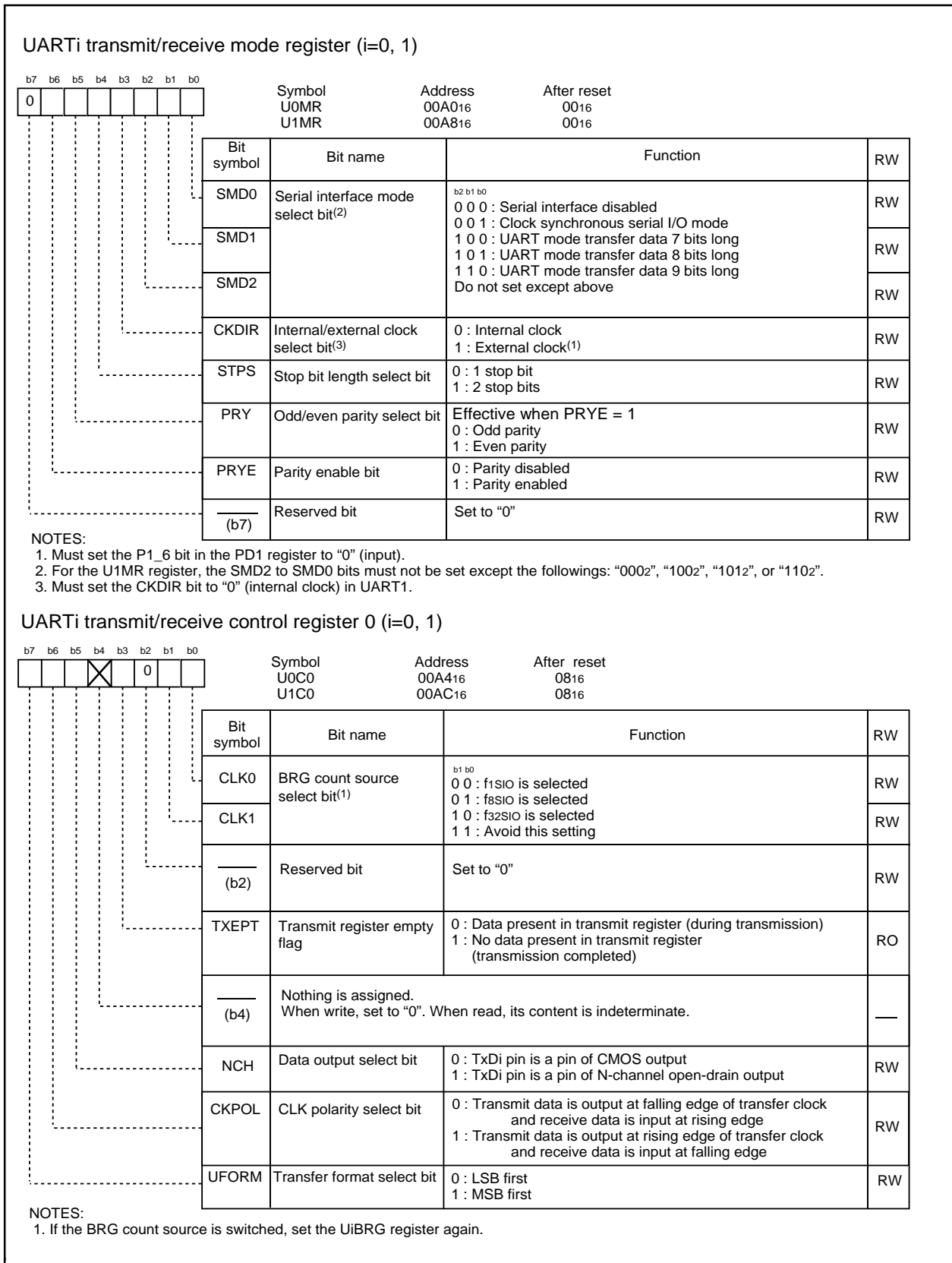


Figure 13.4 U0MR and U1MR Registers and U0C0 and U1C0 Registers

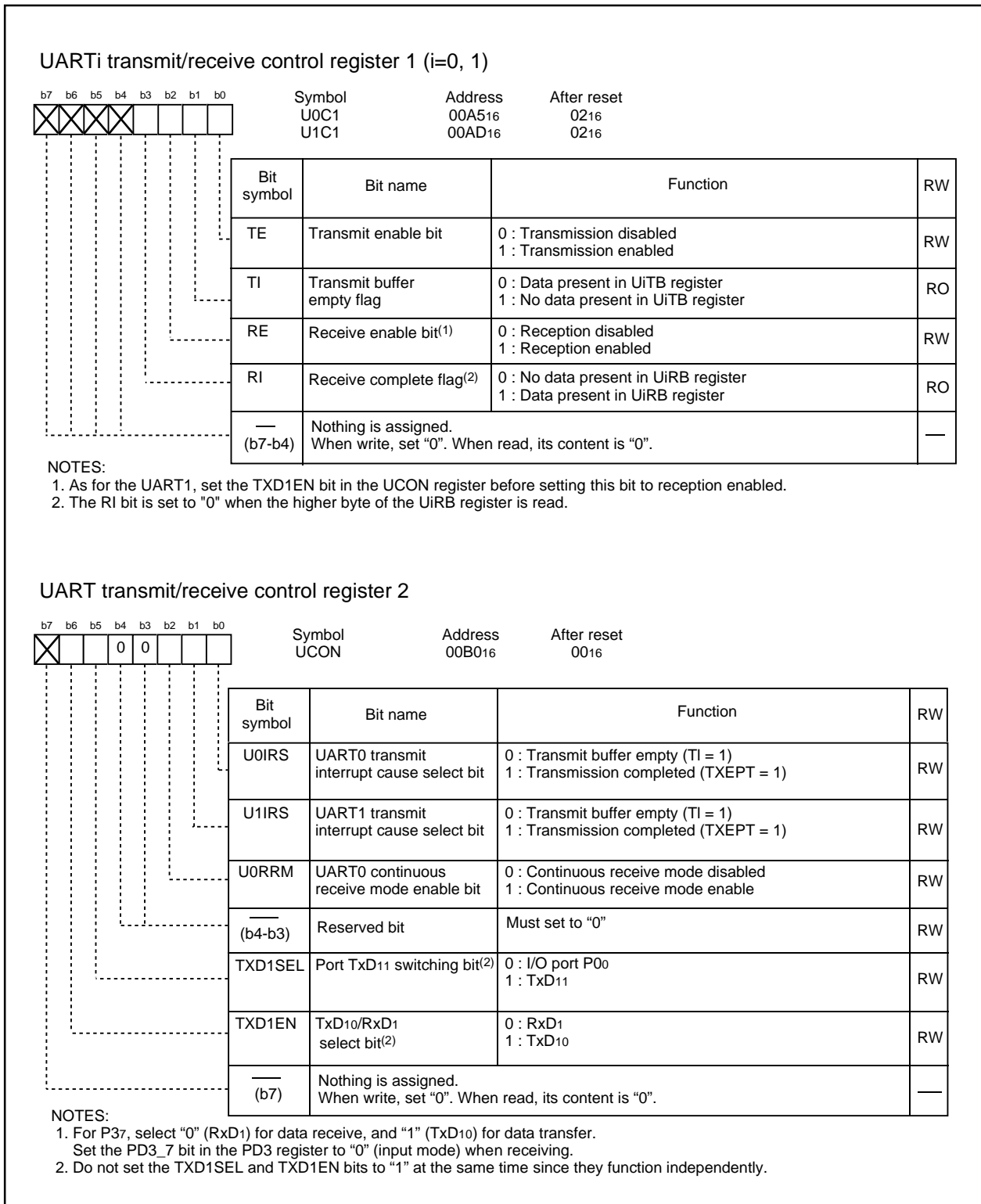


Figure 13.5 U0C1 and U1C1 Registers and UCON Register

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	<ul style="list-style-type: none"> Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> CKDIR bit in U0MR register is set to "0" (internal clock): $f_i/(2(n+1))$ $f_i=f_{1SIO}, f_{8SIO}, f_{32SIO}$ n=setting value in UiBRG register: 00₁₆ to FF₁₆ CKDIR bit is set to "1" (external clock): input from CLK0 pin
Transmission start condition	<ul style="list-style-type: none"> Before transmission can start, the following requirements must be met⁽¹⁾ <ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Before reception can start, the following requirements must be met⁽¹⁾ <ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> 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 interface finished sending data from UARTi transmit register For reception <ul style="list-style-type: none"> When transferring data from the UART0 receive register to the U0RB register (at completion of reception)
Error detection	<ul style="list-style-type: none"> Overrun error⁽²⁾ This error occurs if serial interface started receiving the next data before reading the U0RB register and received the 7th bit of the next data
Select function	<ul style="list-style-type: none"> CLK polarity selection Transfer data I/O 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:

- 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.
- 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 bit rate
U0MR	SMD2 to SMD0	Set to "0012"
	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 I/O 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"(N-channel open-drain output), this pin is in 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

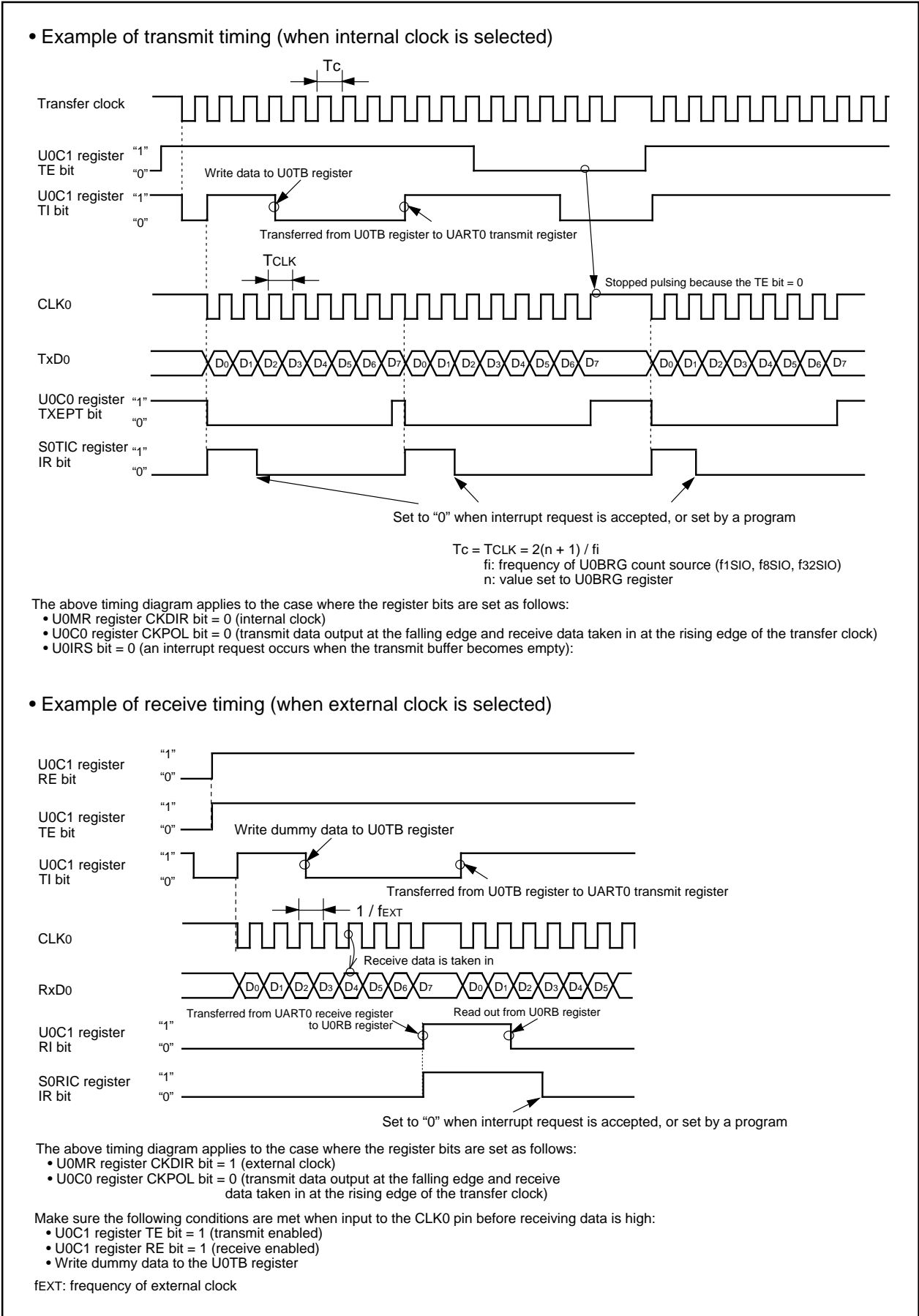


Figure 13.6 Transmit and Receive Operation

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.

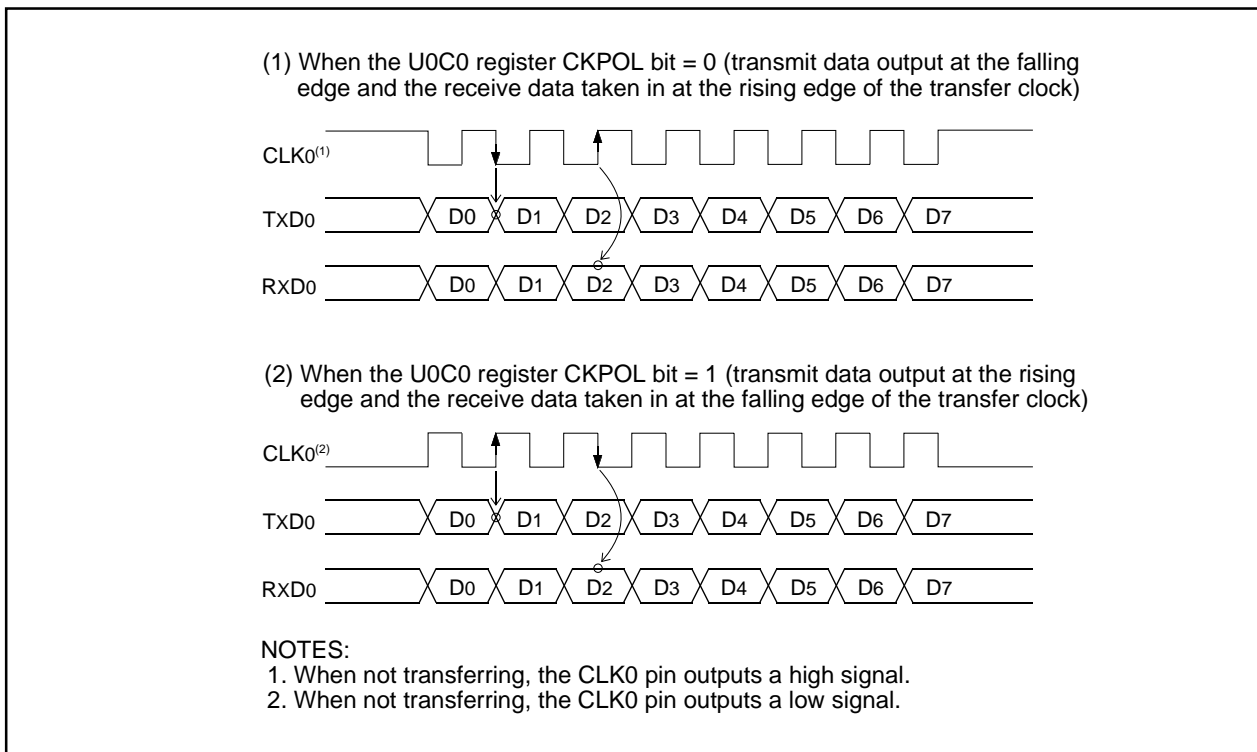


Figure 13.7 Transfer Clock Polarity

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.

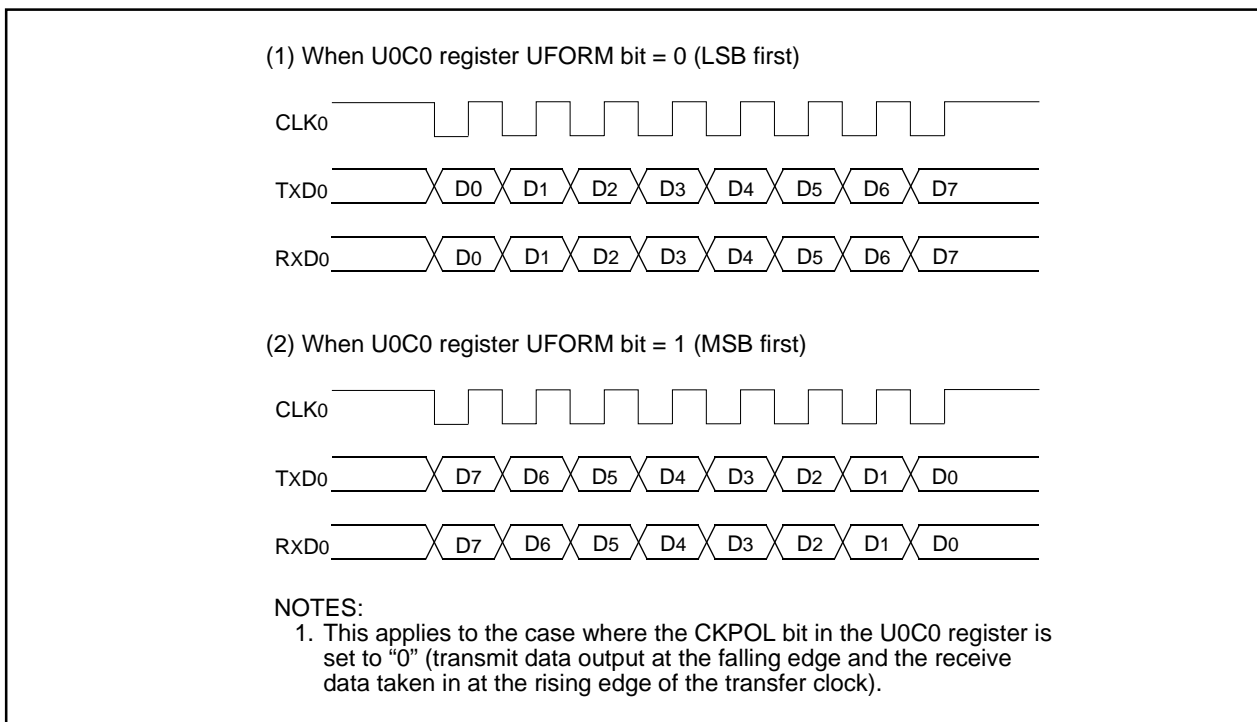


Figure 13.8 Transfer Format

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 sets the TI bit in the U0C1 register to "0"(data in the U0TB register). When the U0RRM bit is set to "1", do not write dummy data to the 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 bit 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	<ul style="list-style-type: none"> • Character bit (transfer data): selectable from 7, 8 or 9 bits • Start bit: 1 bit • Parity bit: selectable from odd, even, or none • Stop bit: selectable from 1 or 2 bits
Transfer clock	<ul style="list-style-type: none"> • UiMR(i=0, 1) register CKDIR bit = 0 (internal clock) : $f_j/(16(n+1))$ $f_j=f_{1SIO}, f_{8SIO}, f_{32SIO}$ n=setting value in UiBRG register: 00₁₆ to FF₁₆ • CKDIR bit = "1" (external clock) : $f_{EXT}/(16(n+1))$ f_{EXT}: input from CLKi pin n=setting value in UiBRG register: 00₁₆ to FF₁₆
Transmission start condition	<ul style="list-style-type: none"> • Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> – TE bit in UiC1 register= 1 (transmission enabled) – TI bit in UiC1 register = 0 (data present in UiTB register)
Reception start condition	<ul style="list-style-type: none"> • Before reception can start, the following requirements must be met <ul style="list-style-type: none"> – RE bit in UiC1 register= 1 (reception enabled) – Start bit detection
Interrupt request generation timing	<ul style="list-style-type: none"> • For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> – 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 interface finished sending data from UARTi transmit register • For reception <ul style="list-style-type: none"> When transferring data from UARTi receive register to UiRB register (at completion of reception)
Error detection	<ul style="list-style-type: none"> • Overrun error⁽¹⁾ This error occurs if serial interface 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	<ul style="list-style-type: none"> • TXD10, RXD1 selection (UART) P37 pin can be used as RXD1 pin or TXD10 pin in UART1. Select by a program. • TXD11 pin selection (UART1) P00 pin can be used as TXD11 pin in UART1 or port P00. Select by a program.

NOTES:

1. If an overrun error occurs, the value of UORB register will be indeterminate. The IR bit in the SORIC 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	0 to 7	Set a bit rate
UiMR	SMD2 to SMD0	Set these bits to '1002' when transfer data is 7 bits long Set these bits to '1012' when transfer data is 8 bits long Set these bits to '1102' when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock ⁽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
	CKPOL	Set to "0"
	UFORM	LSB first or MSB first can be selected when transfer data is 8 bits long. Set this bit to "0" when transfer data is 7 or 9 bits long.
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
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 TxD10 or RxD1 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 NCH bit is set to "1"(N-channel open-drain output), this pin is in high-impedance state.)

Table 13.6 I/O Pin Functions in UART Mode

Pin name	Function	Method of selection
TxD0 (P14)	Serial data output	(Cannot be used as a port when performing reception only)
RxD0 (P15)	Serial data input	PD1 register PD1_5 bit=0 (Can be used as an input port when performing transmission only)
CLK0 (P16)	Programmable I/O port	U0MR register CKDIR bit=0
	Transfer clock input	U0MR register CKDIR bit=1 PD1 register PD1_6 bit=0
TxD10/RxD1 (P37)	Serial data output	TXD1EN=1
	Serial data input	TXD1EN=0, PD3 register PD3_7 bit=0
TxD11 (P00)	Serial data output	Serial data output, TXD1SEL=1

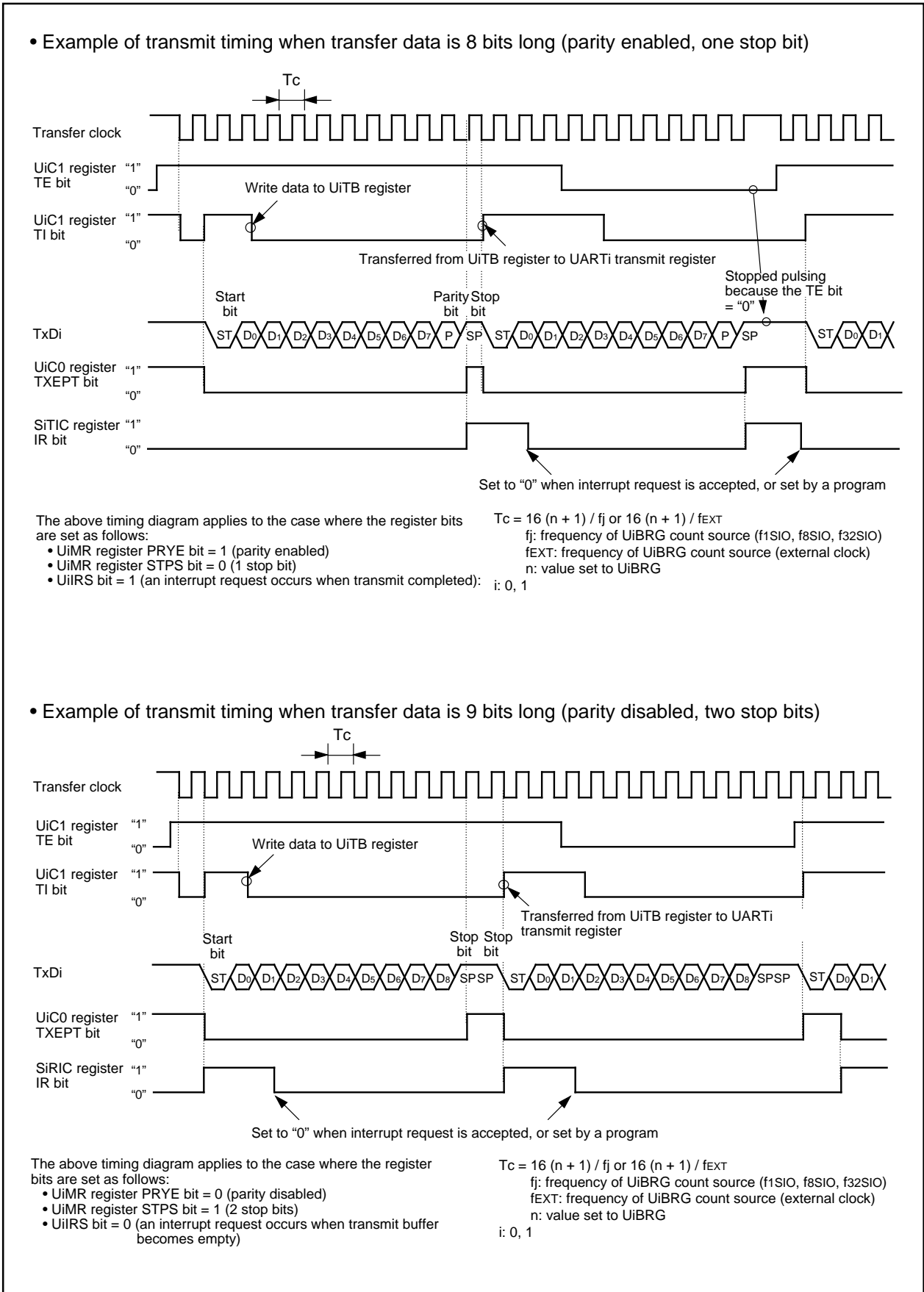


Figure 13.9 Transmit Operation

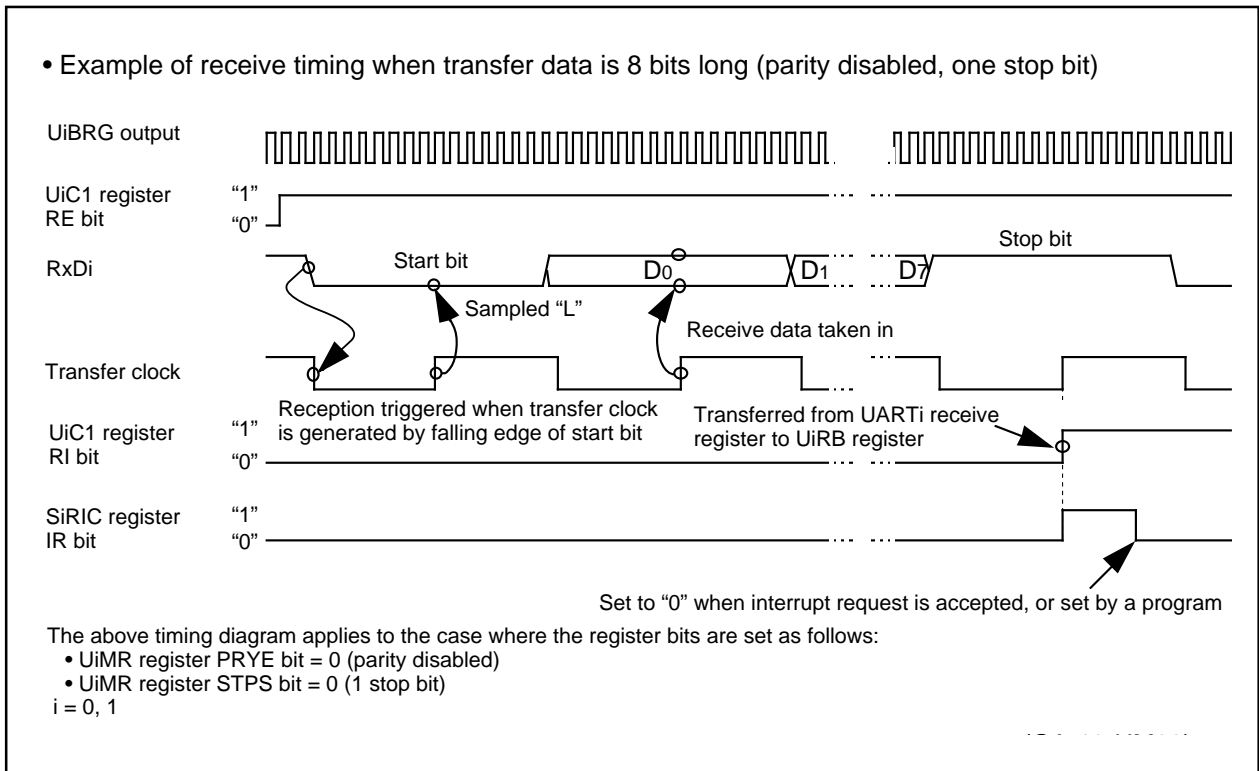


Figure 13.10 Receive Operation

13.2.1 TxD10/RxD1 Select Function (UART1)

P37 can be used as TxD10 output pin or RxD1 input pin by selecting with the TXD1EN bit in the UCON register. P37 is used as TxD10 output pin if the TXD1EN bit is set to "1" (TxD10) and used as RxD1 input pin if set to "0" (RxD1).

13.2.2 TxD11 Select Function (UART1)

P00 can be used as TxD11 output pin or a port by selecting with the TXD1SEL bit in the UCON register. P00 is used as TxD11 output pin if the TXD1SEL bit is set to "1" (TxD11) and used as an I/O port if set to "0" (P00).

13.2.3 Bit Rate

Divided-by-16 of frequency by the UiBRG (i=0 to 1) register in UART mode is a bit rate.

<UART Mode>

- When selecting internal clock
 Setting value to the UiBRG register = $\frac{f_j}{\text{Bit Rate} \times 16} - 1$

f_j : Count source frequency of the UiBRG register (f1SIO, f8SIO and f32SIO)

- When selecting external clock
 Setting value to the UiBRG register = $\frac{f_{EXT}}{\text{Bit Rate} \times 16} - 1$

f_{EXT} : Count source frequency of the UiBRG register (external clock)

Figure 13.11 Calculation Formula of UiBRG (i=0 to 1) Register Setting Value

Table 13.7 Bit Rate Setting Example in UART Mode

Bit Rate (bps)	BRG Count Source	System Clock = 16MHz			System Clock = 8MHz		
		BRG Setting Value	Actual Time(bps)	Error(%)	BRG Setting Value	Actual Time(bps)	Error(%)
1200	f8	103 (67 ₁₆)	1201.92	0.16	51 (33 ₁₆)	1201.92	0.16
2400	f8	51 (33 ₁₆)	2403.85	0.16	25 (19 ₁₆)	2403.85	0.16
4800	f8	25 (19 ₁₆)	4807.69	0.16	12 (0C ₁₆)	4807.69	0.16
9600	f1	103 (67 ₁₆)	9615.38	0.16	51 (33 ₁₆)	9615.38	0.16
14400	f1	68 (44 ₁₆)	14492.75	0.64	34 (22 ₁₆)	14285.71	-0.79
19200	f1	51 (33 ₁₆)	19230.77	0.16	25 (19 ₁₆)	19230.77	0.16
28800	f1	34 (22 ₁₆)	28571.43	-0.79	16 (10 ₁₆)	29411.76	2.12
31250	f1	31 (1F ₁₆)	31250.00	0.00	15 (0F ₁₆)	31250.00	0.00
38400	f1	25 (19 ₁₆)	38461.54	0.16	12 (0C ₁₆)	38461.54	0.16
51200	f1	19 (13 ₁₆)	50000.00	-2.34	9 (09 ₁₆)	50000.00	-2.34

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 P0₀ to P0₇. 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⁾	AVCC = 5V fAD, divide-by-2 of fAD, divide-by-4 of fAD AVCC = 3V divide-by-2 of fAD, divide-by-4 of fAD
Resolution	8-bit or 10-bit (selectable)
Integral nonlinearity error	AVCC = Vref = 5V <ul style="list-style-type: none"> • 8-bit resolution ± 2LSB • 10-bit resolution ± 3LSB AVCC = Vref = 3.3V <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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 AVCC is less than 4.2V, ϕ_{AD} must be fAD/2 or less by dividing fAD.
 - 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.

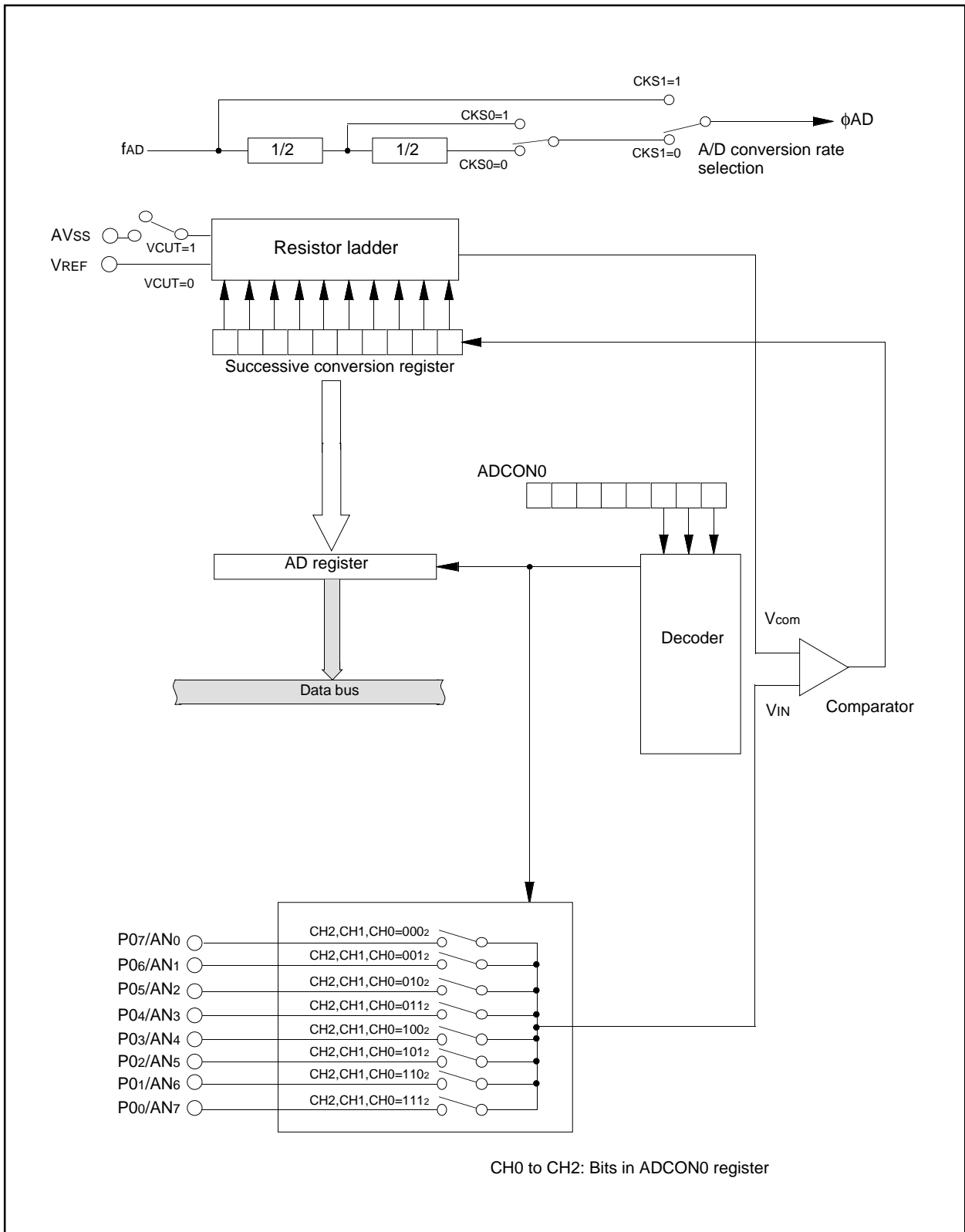


Figure 14.1 A/D Converter Block Diagram

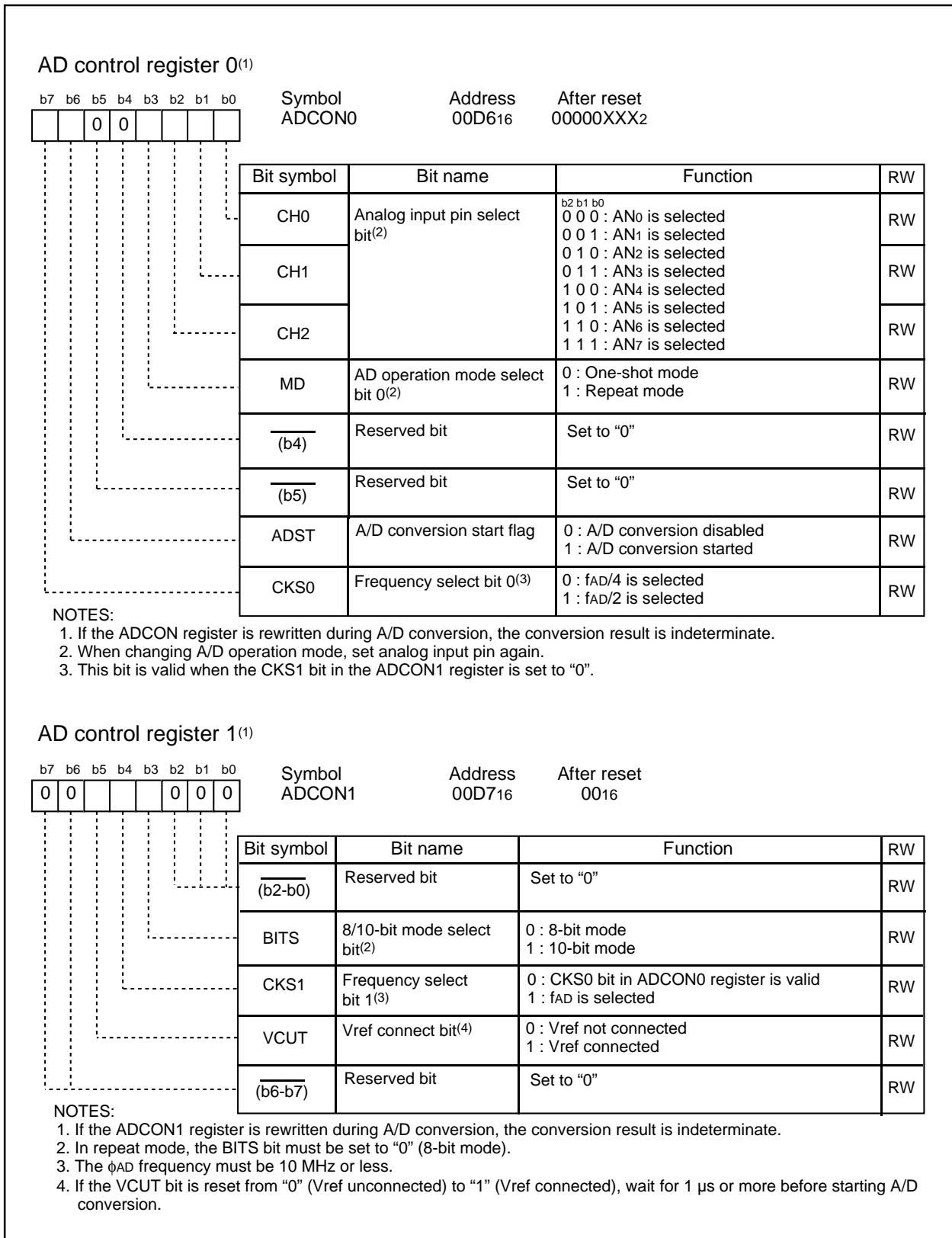


Figure 14.2 ADCON0 Register and ADCON1 Register

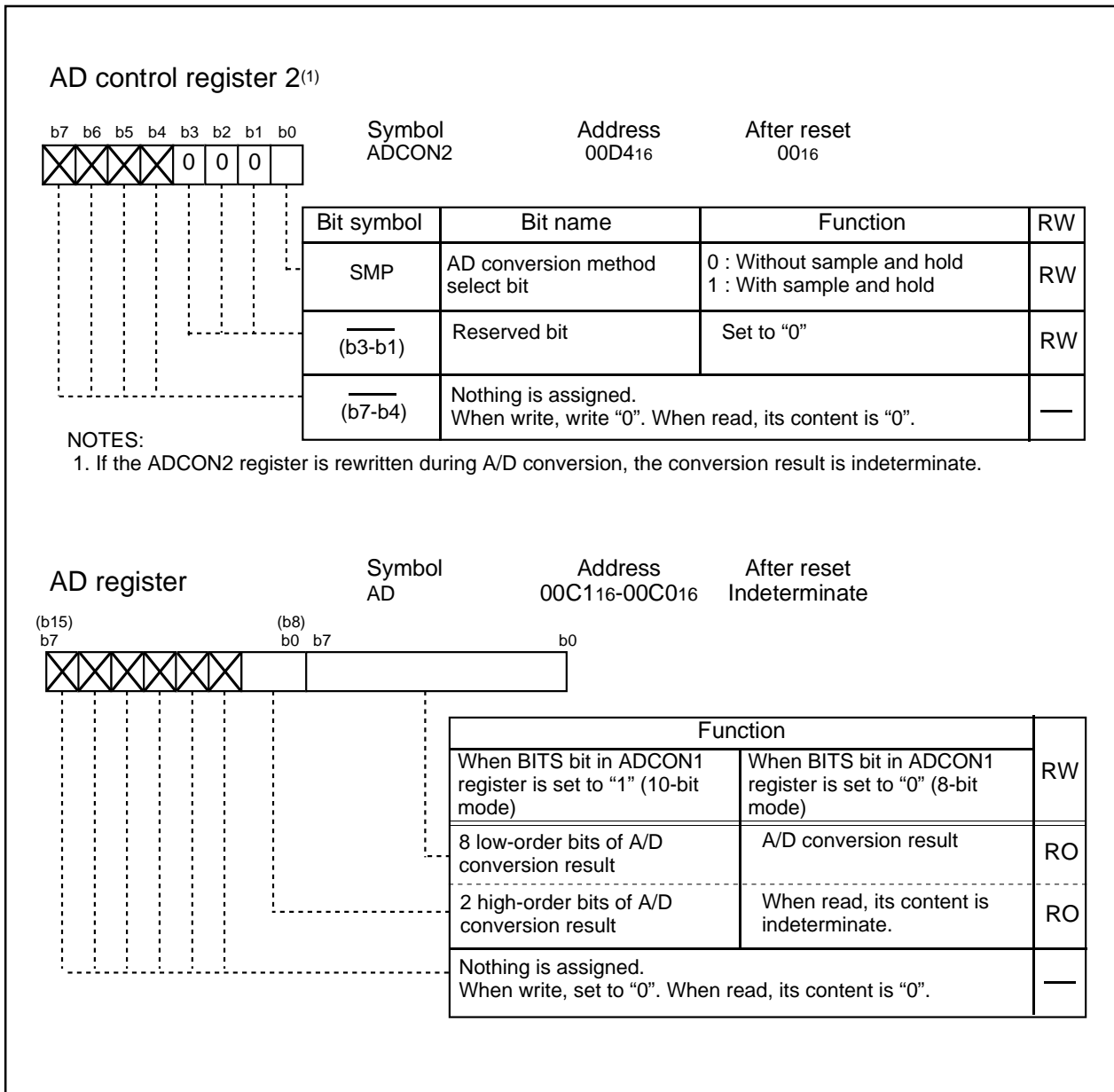


Figure 14.3 ADCON2 Register and AD Register

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	<ul style="list-style-type: none"> • 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 AN0 to AN7, as selected
Reading of result of A/D converter	Read AD register

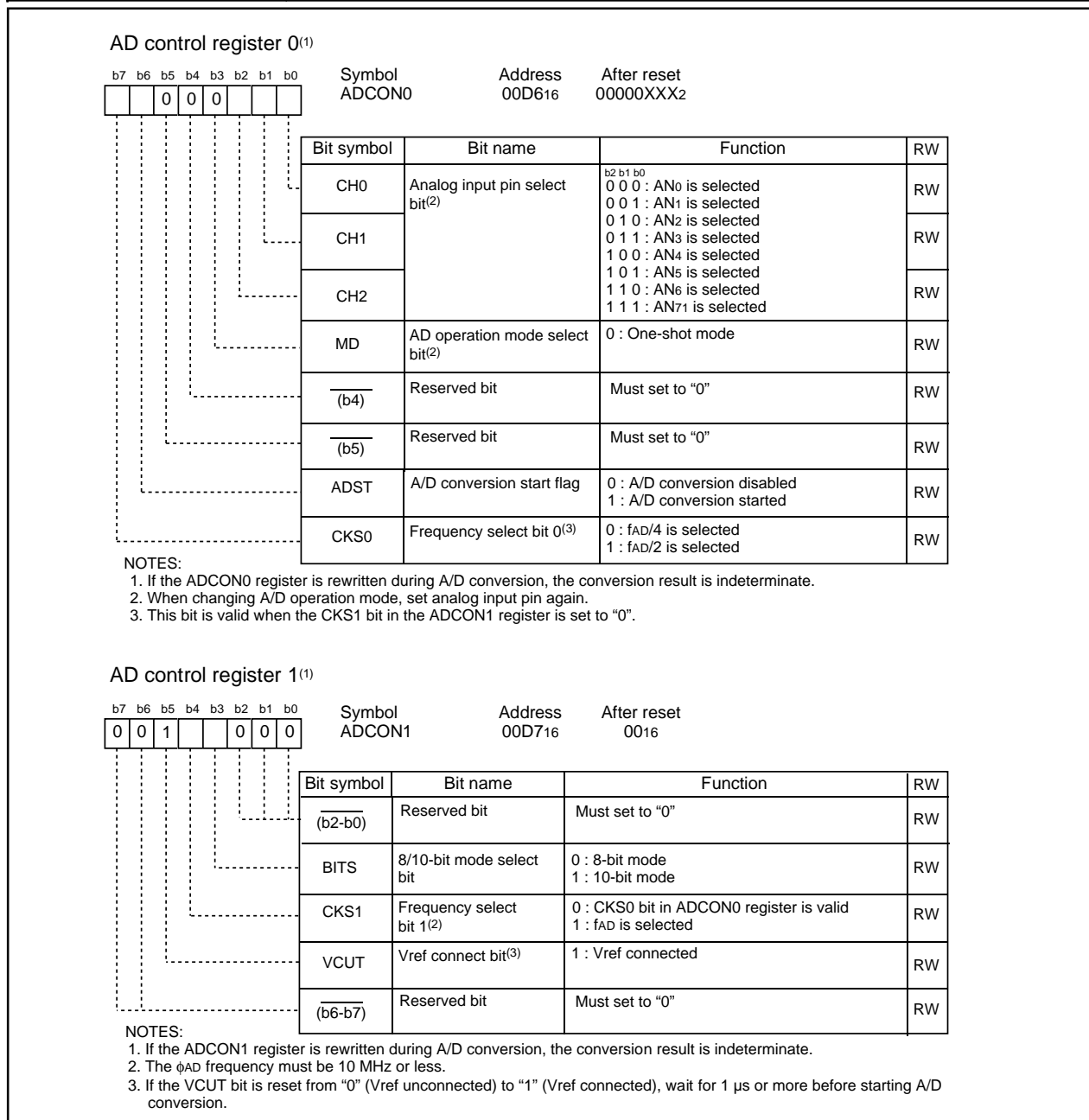


Figure 14.4 ADCON0 Register and ADCON1 Registers in One-shot Mode

14.2 Repeat Mode

In repeat mode, the input voltage 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

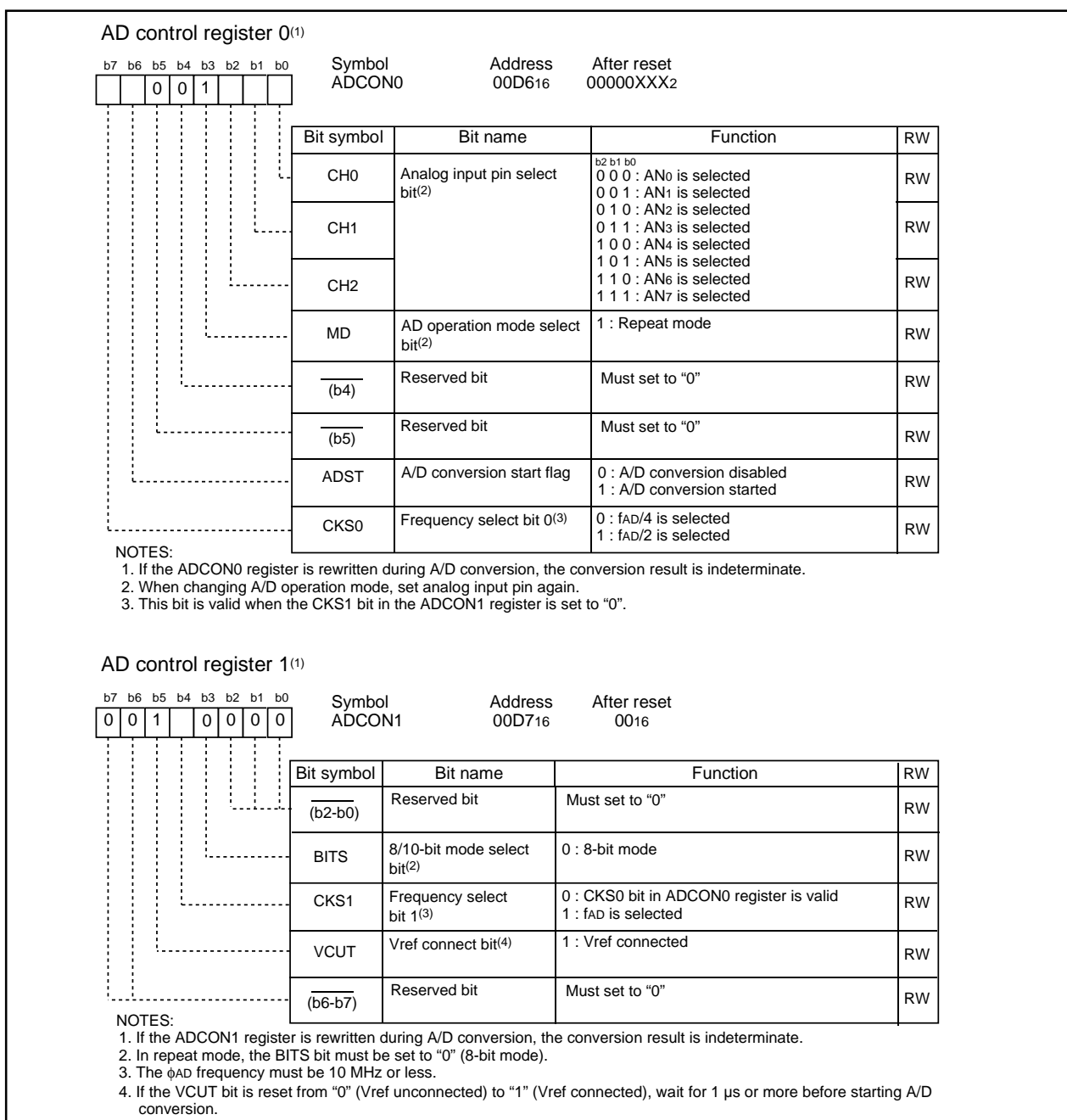


Figure 14.5 ADCON0 Register and ADCON1 Register in Repeat Mode

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 ϕ AD cycles for 8-bit resolution or 33 ϕ AD cycles for 10-bit resolution. Sample-and-hold is effective in all operation modes. Select whether or not to use the sample-and-hold function before starting A/D conversion.

When performing the A/D conversion, charge the comparator capacitor inside the microcomputer. Figure 14.6 shows the A/D conversion timing diagram.

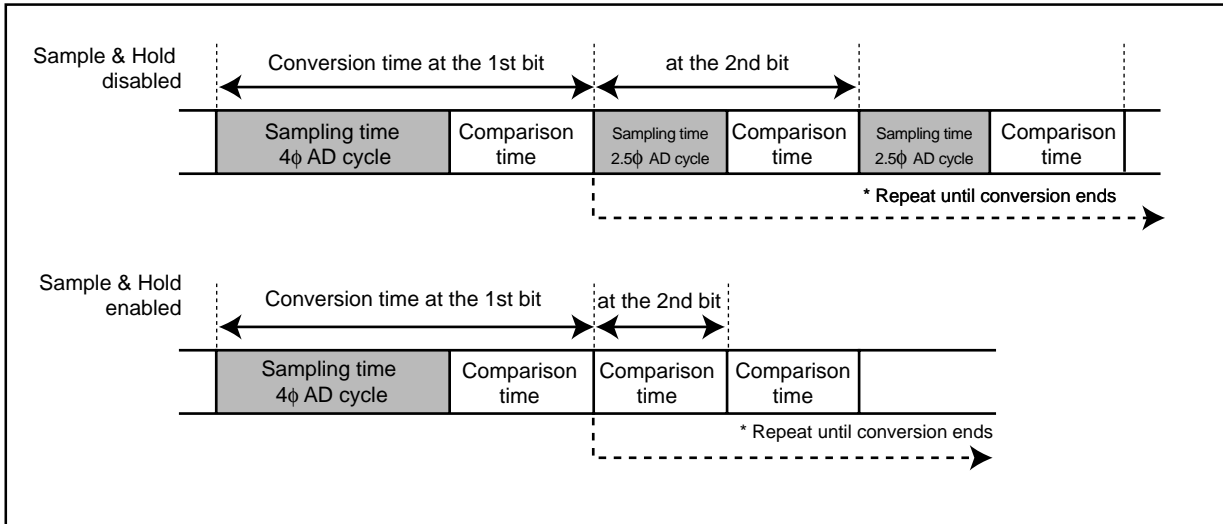


Figure 14.6 A/D Conversion Timing Diagram

14.4 A/D conversion cycles

Figure 14.7 shows the A/D conversion cycles.

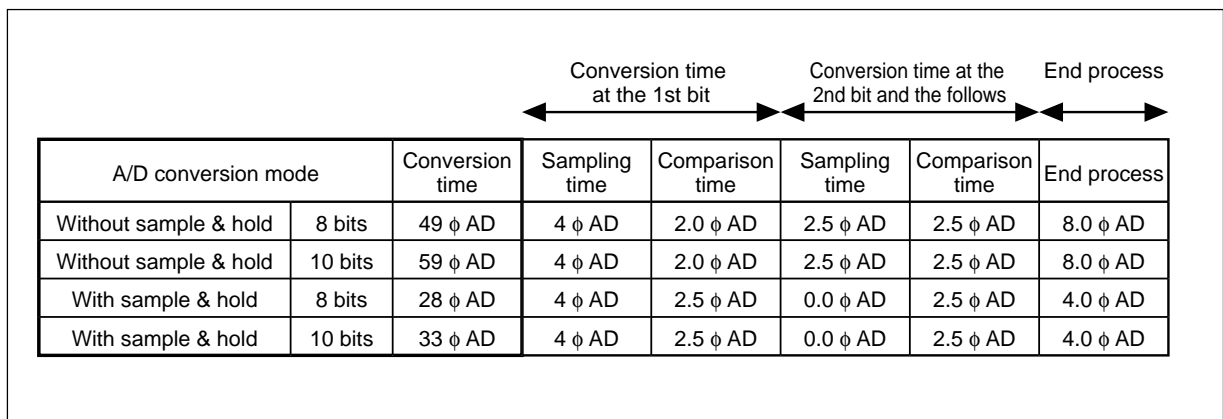


Figure 14.7 A/D Conversion Cycles

14.5 Internal Equivalent Circuit of Analog Input

Figure 14.8 shows the internal equivalent circuit of analog input.

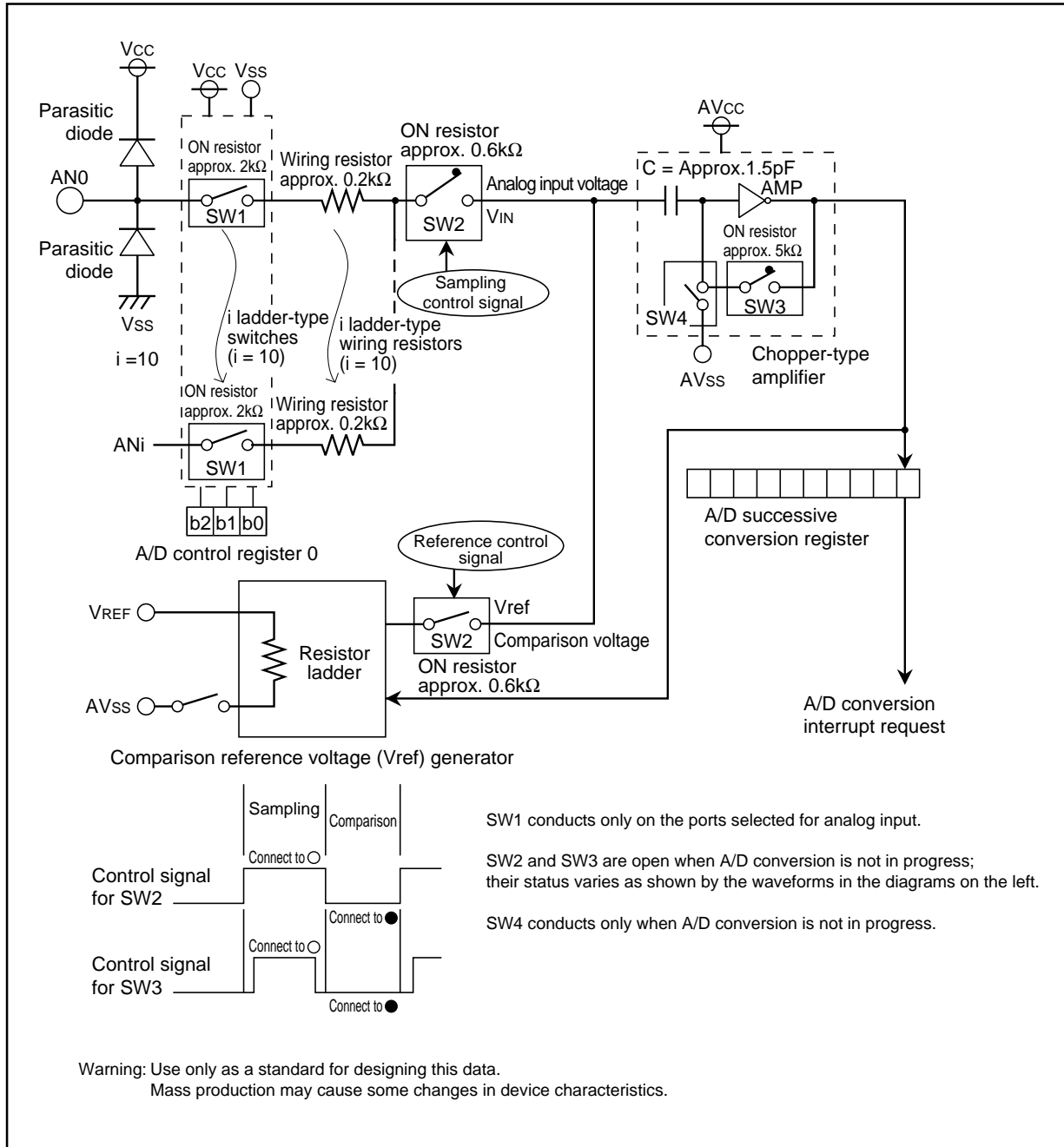


Figure 14.8 Internal Equivalent Circuit to Analog Input

14.6 Inflow Current Bypass Circuit

Figure 14.9 shows the configuration of the inflow current bypass circuit, figure 14.10 shows the example of an inflow current bypass circuit where VCC or more is applied.

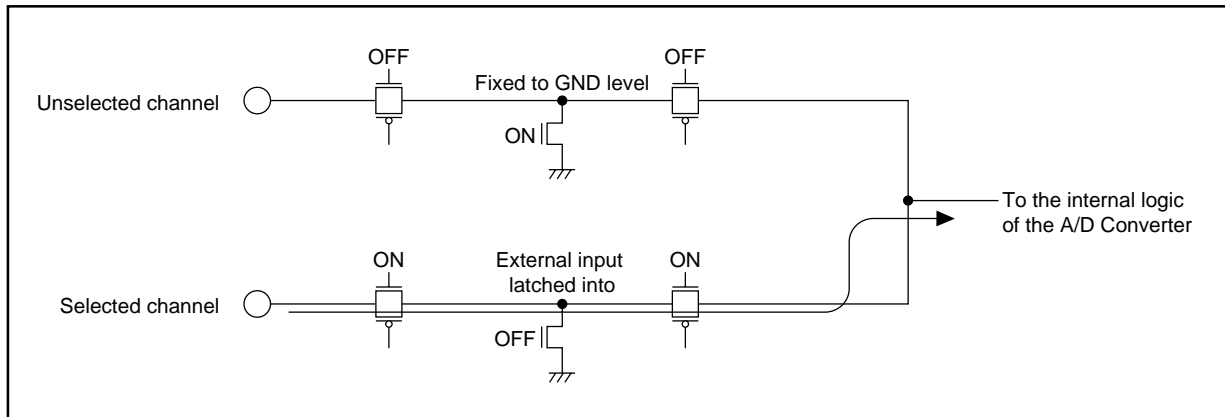


Figure 14.9 Configuration of the Inflow Current Bypass Circuit

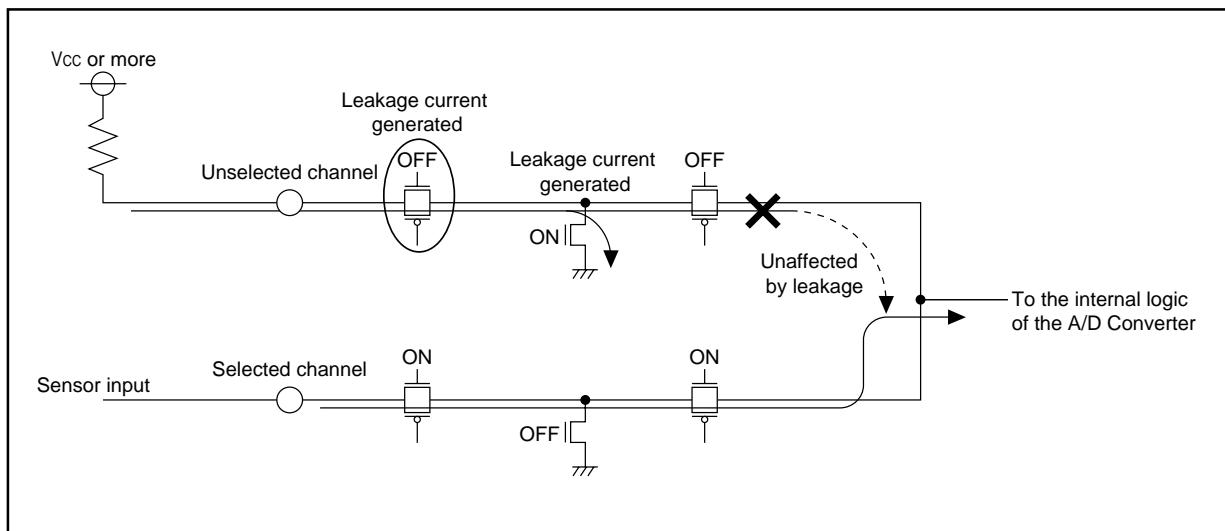


Figure 14.10 Example of an Inflow Current Bypass Circuit where Vcc or More is Applied

14.7 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 14.11 has to be completed within a specified period of time. T (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be R0, microcomputer's internal resistance be R, precision (error) of the A/D converter be X, and the A/D converter's resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

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

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

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

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

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

Figure 14.11 shows analog input pin and external sensor equivalent circuit. When the difference between VIN and VC becomes 0.1 LSB, we find impedance R0 when voltage between pins VC changes from 0 to VIN - (0.1/1024) VIN in time T. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1 LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1 LSB. When f(XIN) = 10 MHz, T = 0.25 μs in the A/D conversion mode with sample & hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.25 μs, R = 2.8 kΩ, C = 1.5 pF, X = 0.1, and Y = 1024. Hence,

$$R0 = -\frac{0.25 \times 10^{-6}}{6.0 \times 10^{-12} \cdot \ln \frac{0.1}{1024}} - 2.8 \times 10^3 \approx 7.3 \times 10^3$$

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A/D converter turns out to be approximately 7.3 kΩ.

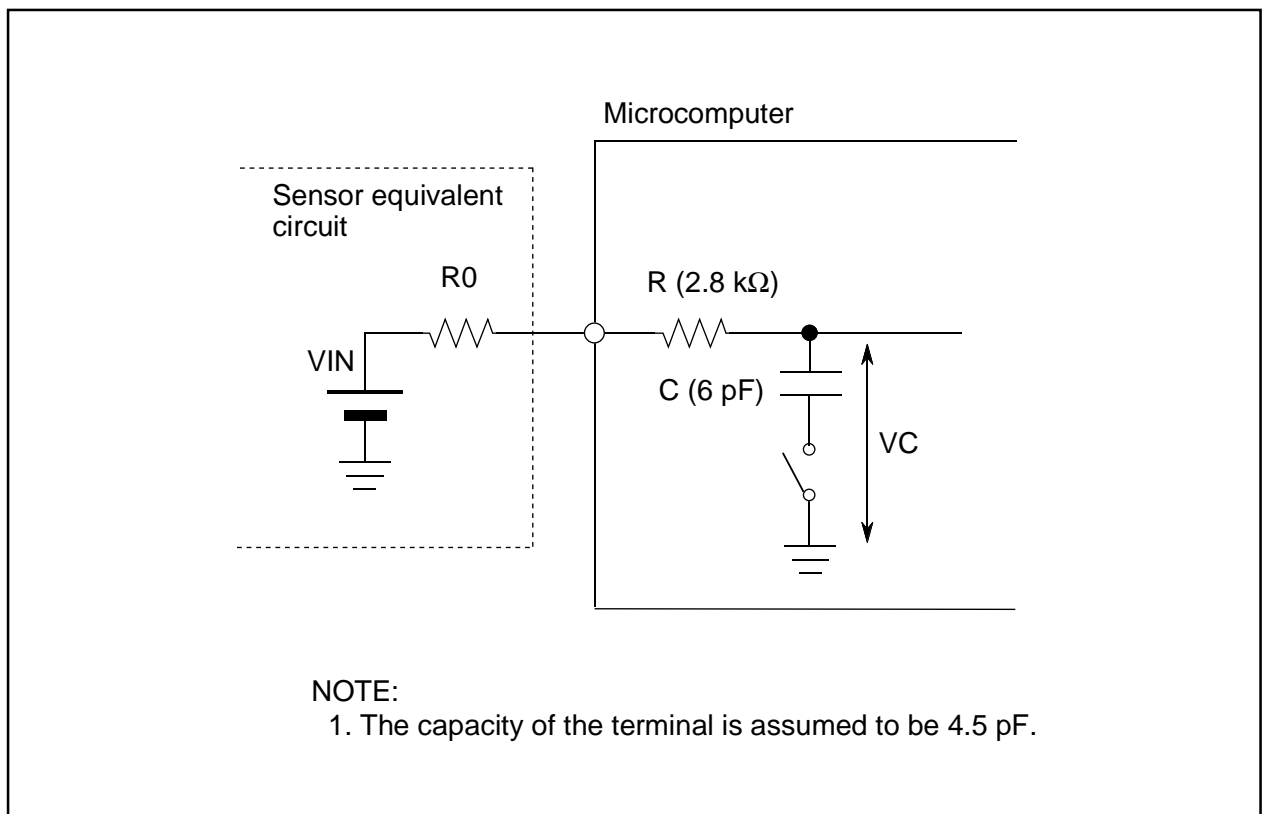


Figure 14.11 Analog Input Pin and External Sensor Equivalent Circuit

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

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

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.

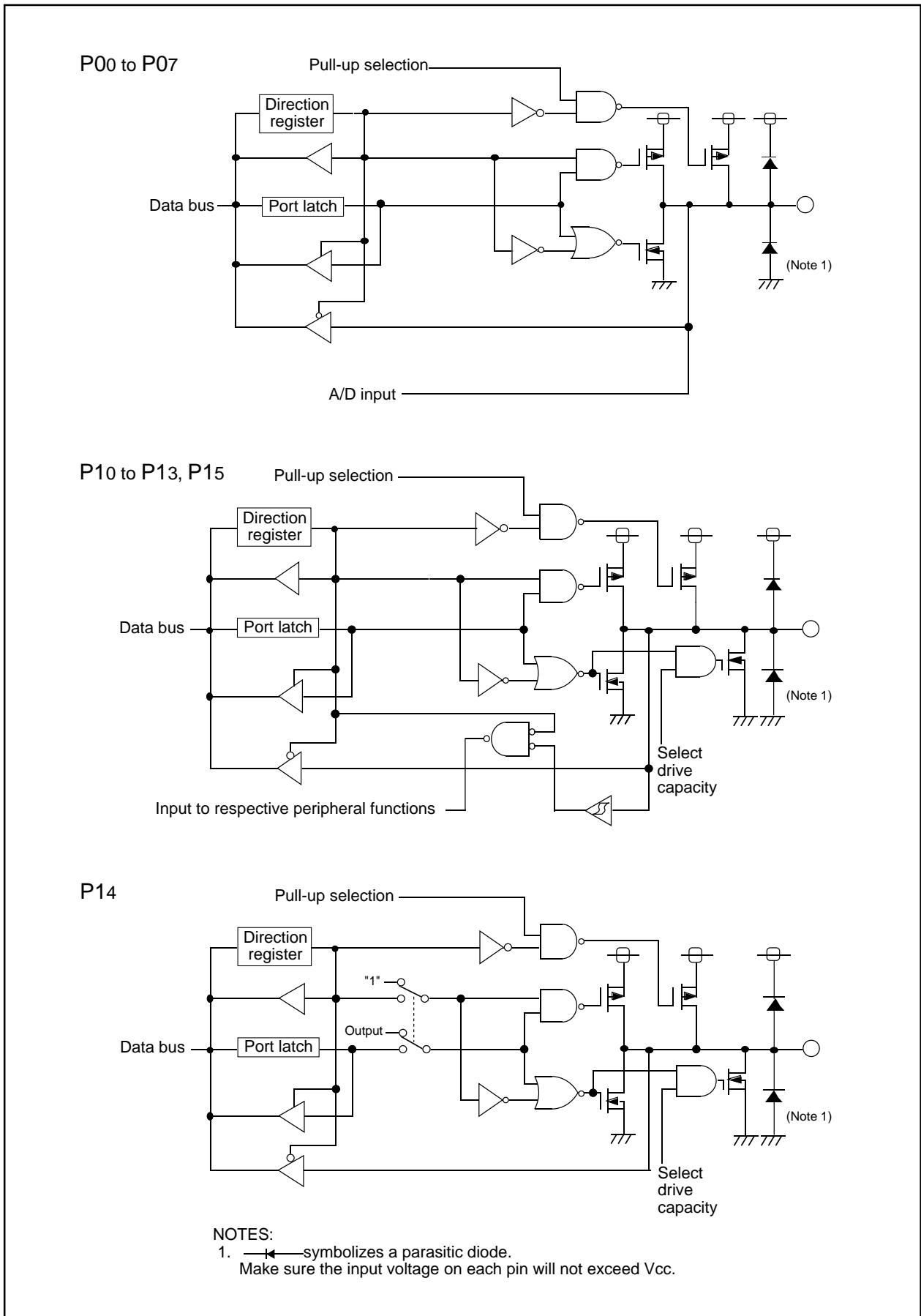


Figure 15.1 Programmable I/O Ports (1)

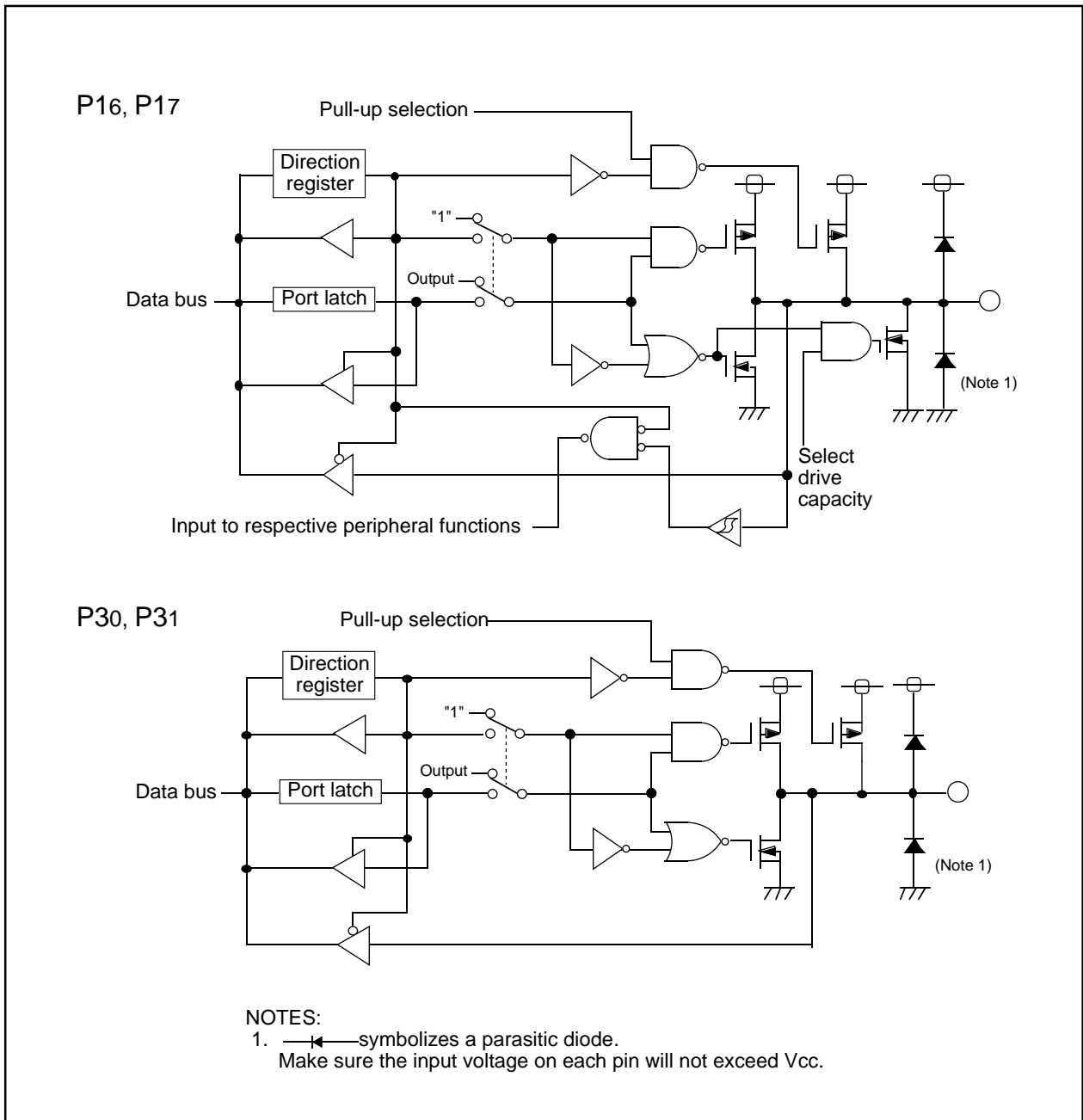


Figure 15.2 Programmable I/O Ports (2)

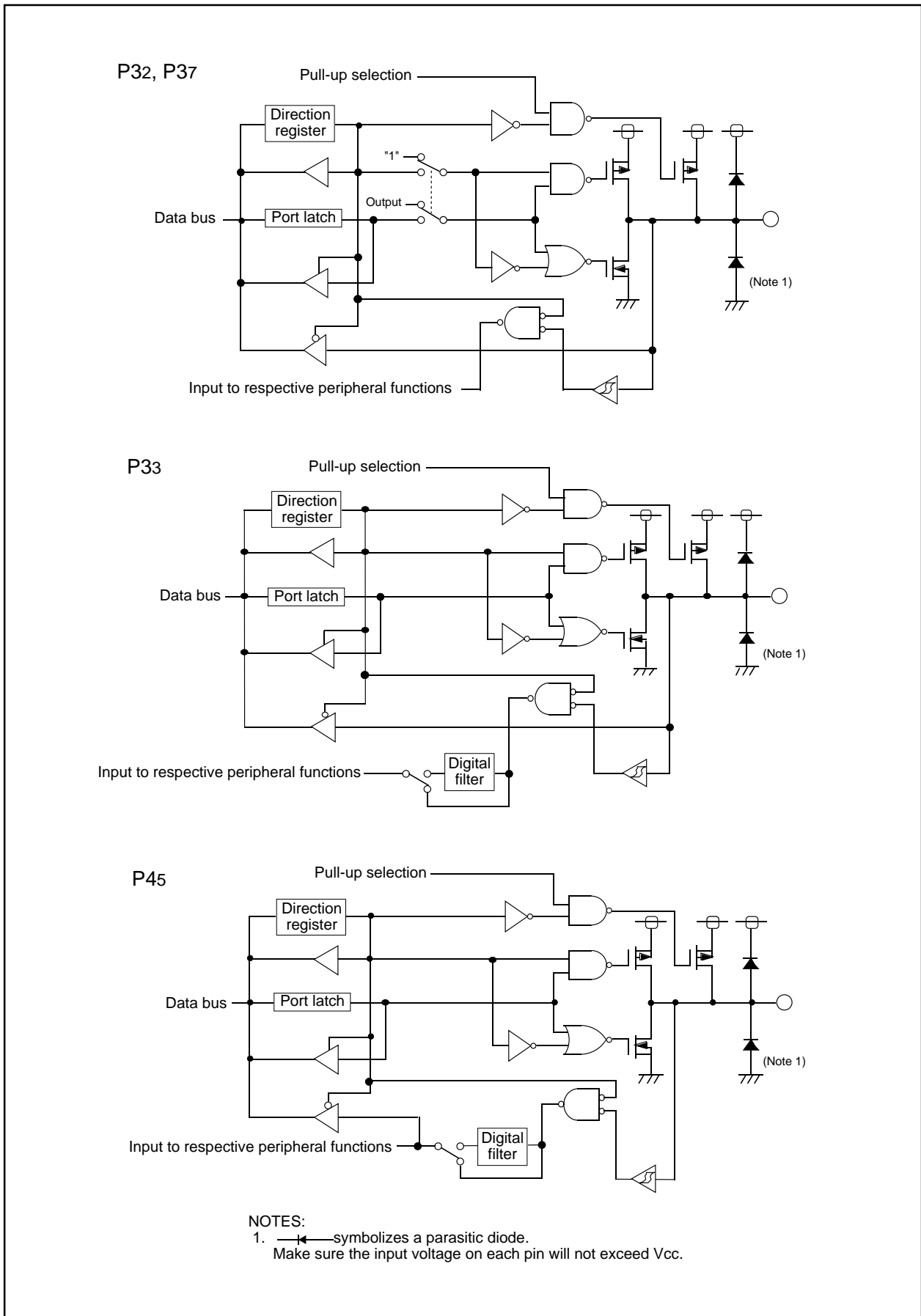


Figure 15.3 Programmable I/O Ports (3)

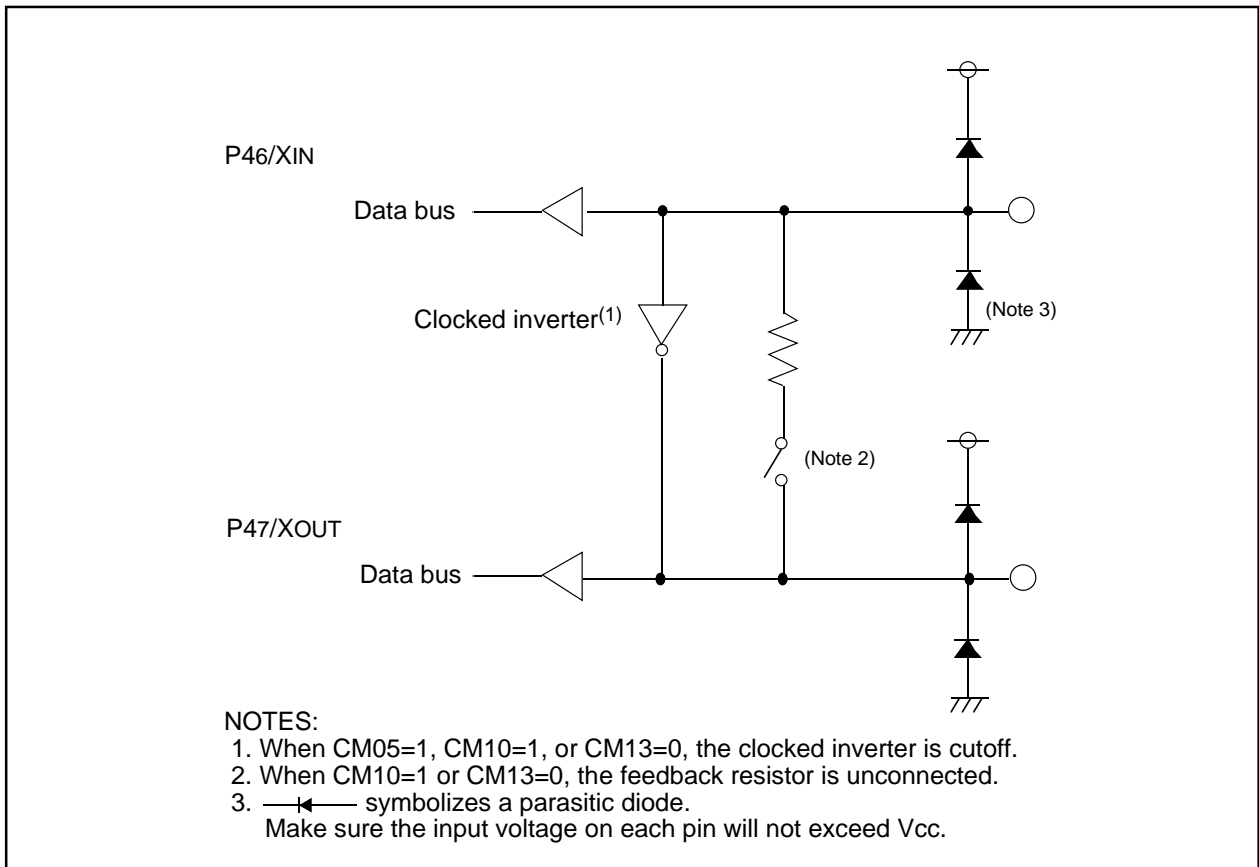


Figure 15.4 Programmable I/O Port (4)

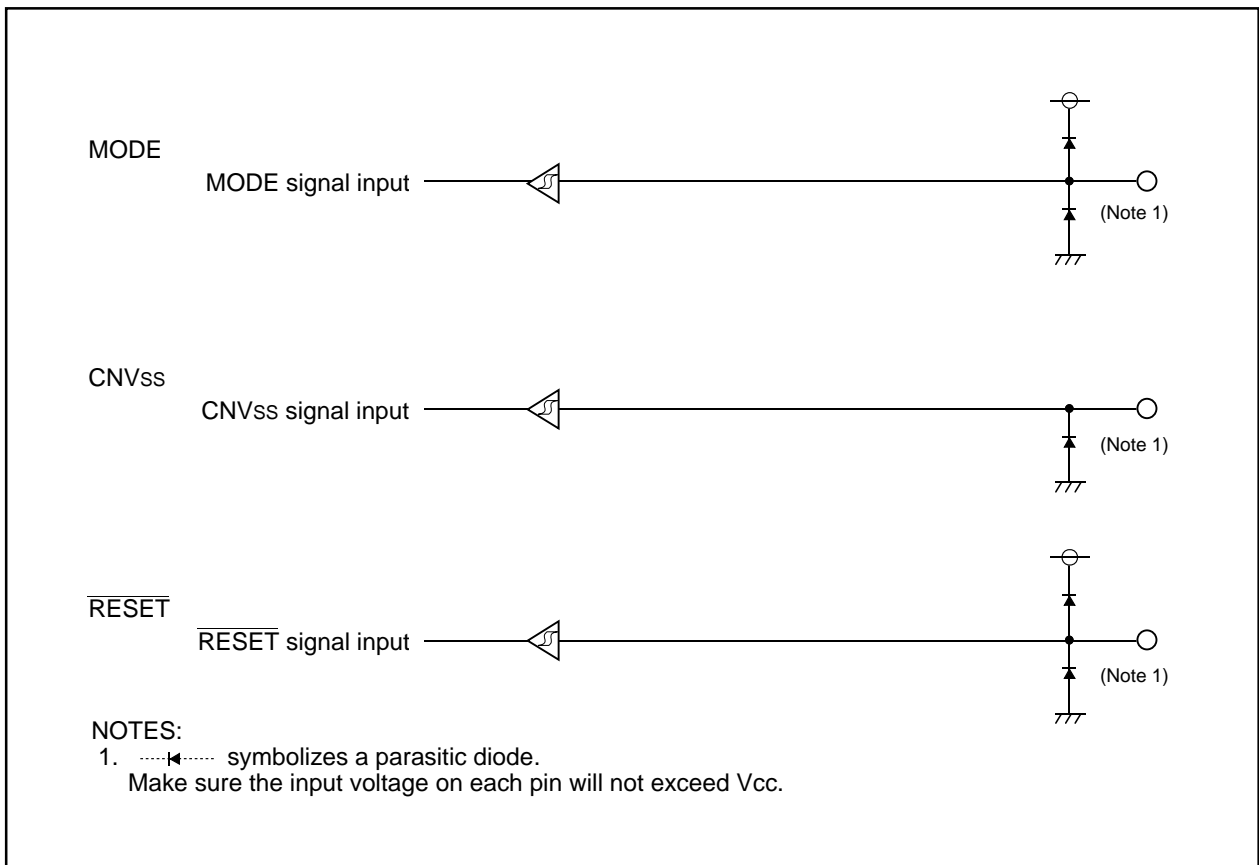


Figure 15.5 I/O Pins

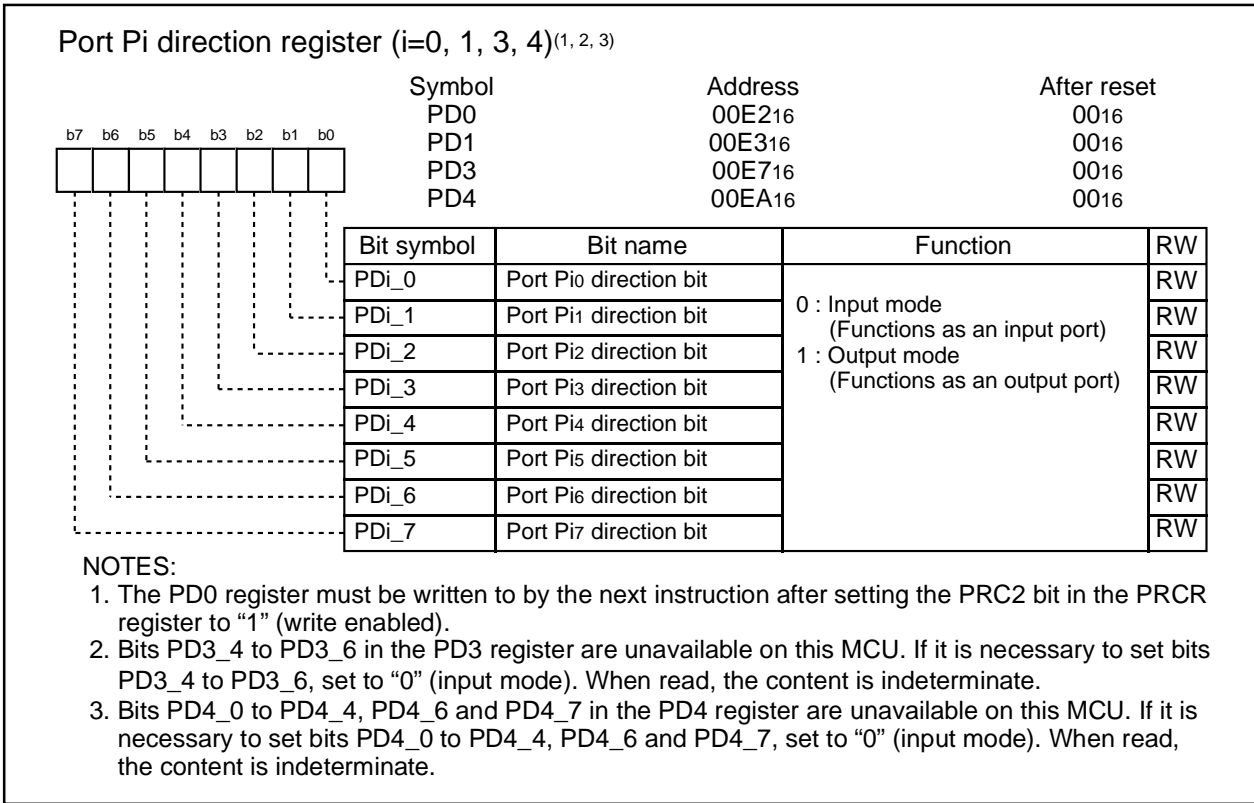


Figure 15.6 PD0 Register, PD1 Register, PD3 Register, and PD4 Register

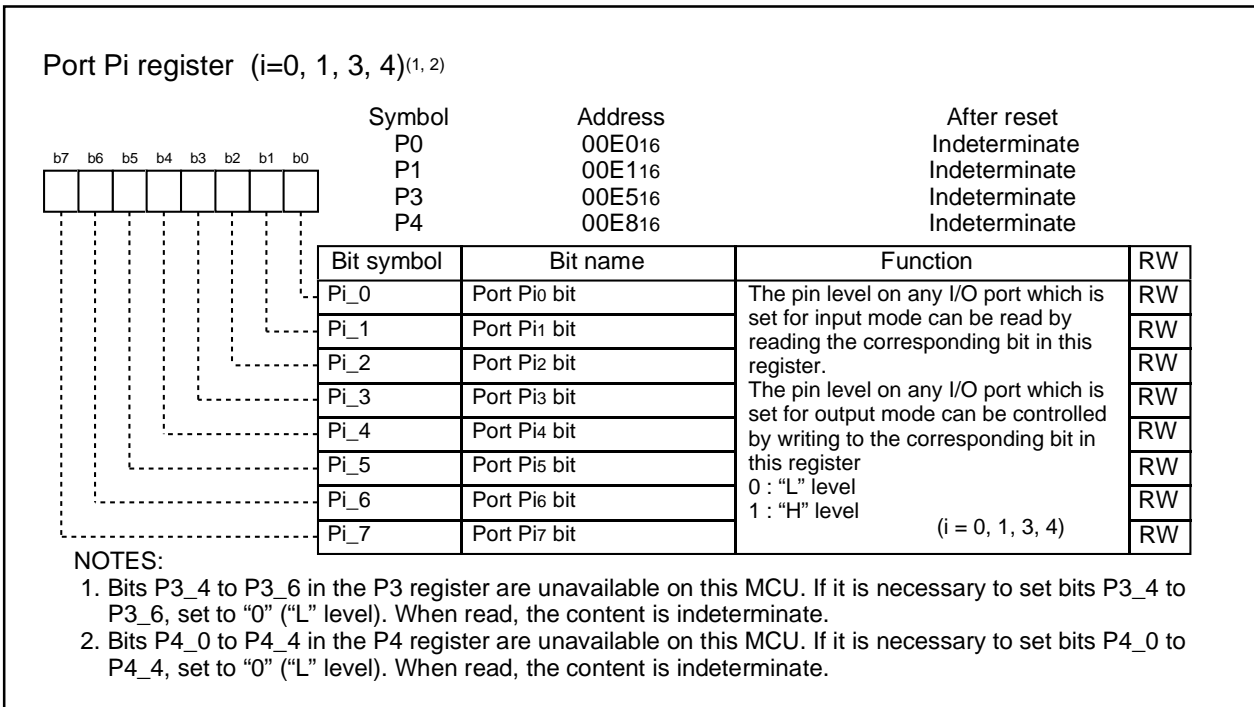
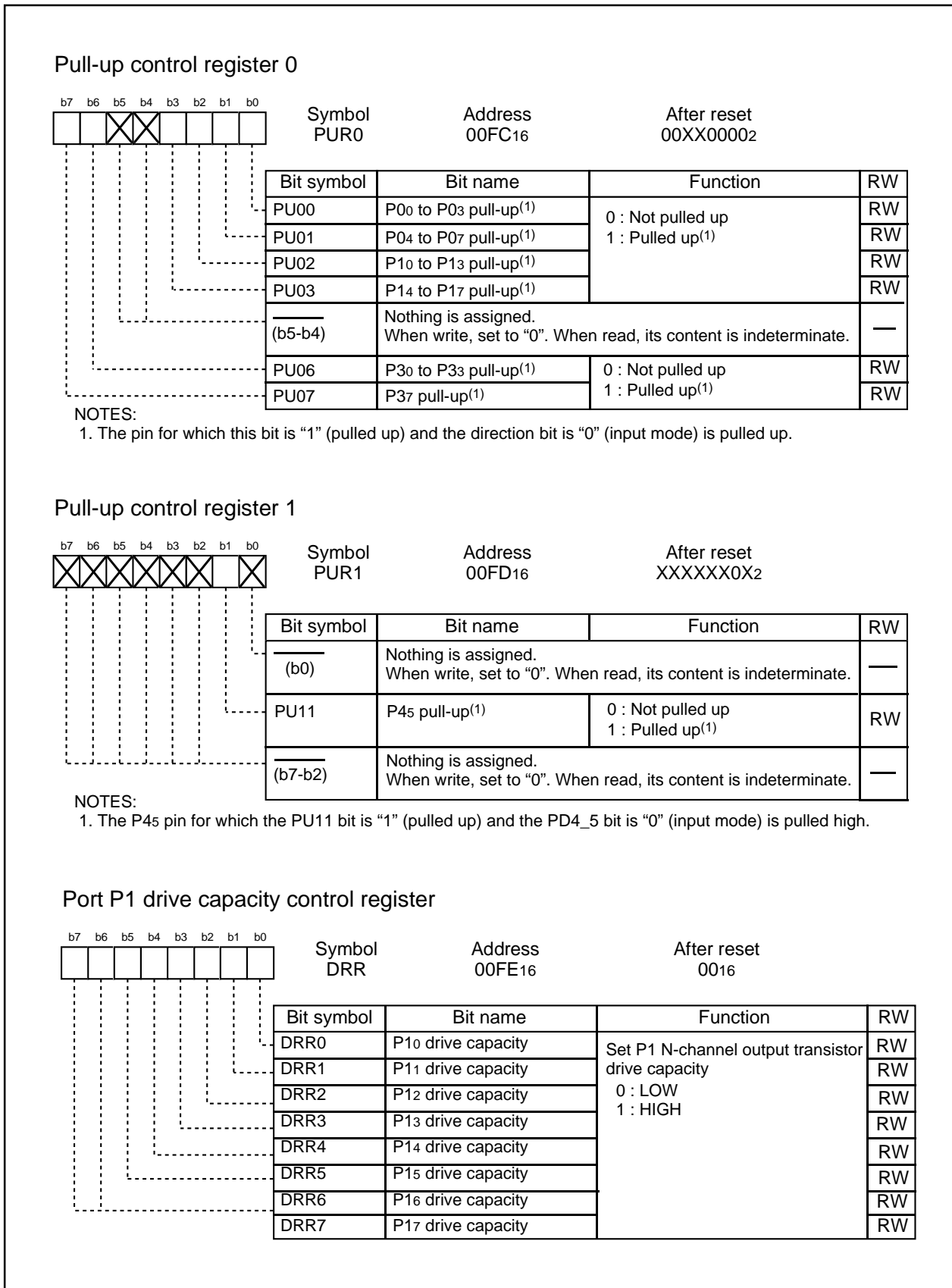


Figure 15.7 P0 Register, P1 Register, P3 Register, and P4 Register



15.2 Port setting

Table 15.1 to Table 15.23 list the port setting.

Table 15.1 Port P00/AN7/TxD11 Setting

Register	PD0	PUR0	ADCON0	UCON	U1MR	U1C0	Function
Bit	PD0_0	PU00	CH2, CH1, CH0	TXD1SEL	SMD2, SMD0	NCH	
Setting value	0	0	XXX	X	002	X	Input port (not pulled up)
				0	XX		
	0	1	XXX	X	002	X	Input port (pulled up)
				0	XX		
	0	0	1112	X	002	X	A/D input (AN7)
				0	XX		
	1	X	XXX	X	002	X	Output port
				0	XX		
	X	X	XXX	1	1X	0	TxD11
					X1		
	X	0	XXX	1	1X	1	TxD11, N-channel open output
					X1		

X: "0" or "1"

Table 15.2 Port P01/AN6 Setting

Register	PD0	PUR0	ADCON0	Function
Bit	PD0_1	PU00	CH2, CH1, CH0	
Setting value	0	0	XXX	Input port (not pulled up)
			1	XXX
	0	0	1102	A/D input (AN6)
			1	X

X: "0" or "1"

Table 15.3 Port P02/AN5 Setting

Register	PD0	PUR0	ADCON0	Function
Bit	PD0_2	PU00	CH2, CH1, CH0	
Setting value	0	0	XXX	Input port (not pulled up)
			1	XXX
	0	0	1012	A/D input (AN5)
			1	X

X: "0" or "1"

Table 15.4 Port P03/AN4 Setting

Register	PD0	PUR0	ADCON0	Function
Bit	PD0_3	PU00	CH2, CH1, CH0	
Setting value	0	0	XXX	Input port (not pulled up)
			1	XXX
	0	0	1002	A/D input (AN4)
			1	X

X: "0" or "1"

Table 15.5 Port P04/AN3 Setting

Register	PD0	PUR0	ADCON0	Function
Bit	PD0_4	PU01	CH2, CH1, CH0	
Setting value	0	0	XXX	Input port (not pulled up)
	0	1	XXX	Input port (pulled up)
	0	0	0112	A/D input (AN3)
	1	X	XXX	Output port

X: "0" or "1"

Table 15.6 Port P05/AN2 setting

Register	PD0	PUR0	ADCON0	Function
Bit	PD0_5	PU01	CH2, CH1, CH0	
Setting value	0	0	XXX	Input port (not pulled up)
	0	1	XXX	Input port (pulled up)
	0	0	0102	A/D input (AN2)
	1	X	XXX	Output port

X: "0" or "1"

Table 15.7 Port P06/AN1 Setting

Register	PD0	PUR0	ADCON0	Function
Bit	PD0_6	PU01	CH2, CH1, CH0	
Setting value	0	0	XXX	Input port (not pulled up)
	0	1	XXX	Input port (pulled up)
	0	0	0012	A/D input (AN1)
	1	X	XXX	Output port

X: "0" or "1"

Table 15.8 Port P07/AN0 Setting

Register	PD0	PUR0	ADCON0	Function
Bit	PD0_7	PU01	CH2, CH1, CH0	
Setting value	0	0	XXX	Input port (not pulled up)
	0	1	XXX	Input port (pulled up)
	0	0	0002	A/D input (AN0)
	1	X	XXX	Output port

X: "0" or "1"

Table 15.9 Port P10/ $\overline{KI0}$ Setting

Register	PD1	PUR0	DRR	KIEN	P1	Function
Bit	PD1_0	PU02	DRR0	KI0EN	P1_0	
Setting value	0	0	X	X	X	Input port (not pulled up)
	0	1	X	X	X	Input port (pulled up)
	0	0	X	1	X	$\overline{KI0}$ input
	1	X	0	X	X	Output port
	1	X	1	X	X	Output port (High drive)

X: "0" or "1"

Table 15.10 Port P11/ $\overline{KI1}$ Setting

Register	PD1	PUR0	DRR	KIEN	P1	Function
Bit	PD1_1	PU02	DRR1	KI1EN	P1_1	
Setting value	0	0	X	X	X	Input port (not pulled up)
	0	1	X	X	X	Input port (pulled up)
	0	0	X	1	X	$\overline{KI1}$ input
	1	X	0	X	X	Output port
	1	X	1	X	X	Output port (High drive)

X: "0" or "1"

Table 15.11 Port P12/ $\overline{KI2}$ Setting

Register	PD1	PUR0	DRR	KIEN	P1	Function
Bit	PD1_2	PU02	DRR2	KI2EN	P1_2	
Setting value	0	0	X	X	X	Input port (not pulled up)
	0	1	X	X	X	Input port (pulled up)
	0	0	X	1	X	$\overline{KI2}$ input
	1	X	0	X	X	Output port
	1	X	1	X	X	Output port (High drive)

X: "0" or "1"

Table 15.12 Port P13/ $\overline{KI3}$ Setting

Register	PD1	PUR0	DRR	KIEN	Function
Bit	PD1_3	PU02	DRR3	KI3EN	
Setting value	0	0	X	X	Input port (not pulled up)
	0	1	X	X	Input port (pulled up)
	0	0	X	1	$\overline{KI3}$ input
	1	X	0	X	Output port
	1	X	1	X	Output port (High drive)

X: "0" or "1"

Table 15.13 Port P14/TxD0 Setting

Register	PD1	PUR0	DRR	U0MR	U0C0	Function
Bit	PD1_4	PU03	DRR4	SMD2, SMD0	NCH	
Setting value	0	0	X	002	X	Input port (not pulled up)
	0	1	X	002	X	Input port (pulled up)
	1	X	0	002	X	Output port
	1	X	1	002	X	Output port (High drive)
	X	X	0	X1 1X	0	TxD0 output, CMOS output
	X	X	1	X1 1X	0	TxD0 output, CMOS output (High drive)
	X	X	0	X1 1X	1	TxD0 output, N-channel open output
	X	X	1	X1 1X	1	TxD0 output, N-channel open output (High drive)

X: "0" or "1"

Table 15.14 Port P15/RxD0 Setting

Register	PD1	PUR0	DRR	Function
Bit	PD1_5	PU03	DRR5	
Setting value	0	0	X	Input port (not pulled up)
	0	1	X	Input port (pulled up)
	0	0	X	RxD0 input
	1	X	0	Output port
	1	X	1	Output port (High drive)

X: "0" or "1"

Table 15.15 Port P16/CLK0 Setting

Register	PD1	PUR0	DRR	U0MR	Function
Bit	PD1_6	PU03	DRR6	SMD2, SMD0, CKDIR	
Setting value	0	0	X	Other than 0102	Input port (not pulled up)
	0	1	X	Other than 0102	Input port (pulled up)
	0	0	X	XX1	CLK0 (external clock) input
	1	X	0	Other than 0102	Output port
	1	X	1	Other than 0102	Output port (High drive)
	X	X	0	0102	CLK0 (internal clock) output
	X	X	1	0102	CLK0 (internal clock) output (High drive)

X: "0" or "1"

Table 15.16 Port P17/ $\overline{\text{INT}}_1/\text{CNTR}_0$ Setting

Register	PD1	PUR0	DRR	TXMR	Function
Bit	PD1_7	PU03	DRR5	TXMOD1, TXMOD0	
Setting value	0	0	X	Other than 012	Input port (not pulled up)
	0	1	X	Other than 012	Input port (pulled up)
	0	0	X	Other than 012	$\text{CNTR}_0/\overline{\text{INT}}_1$ input
	1	X	0	Other than 012	Output port
	1	X	1	Other than 012	Output port (High drive)
	X	X	0	012	CNTR_0 output
	X	X	1	012	CNTR_0 (High drive)

X: "0" or "1"

Table 15.17 Port P30/ CNTR_0 Setting

Register	PD3	PUR0	TXMR	P3	Function
Bit	PD3_0	PU06	TXOCNT	P3_0	
Setting value	0	0	0	X	Input port (not pulled up)
	0	1	0	X	Input port (pulled up)
	1	X	0	X	Output port
	X	X	1	X	CNTR_0 output

X: "0" or "1"

Table 15.18 Port P31/ TZOUT Setting

Register	PD3	PUR0	TYZMR	TYZOC	P3	Function
Bit	PD3_1	PU06	TZMOD1, TZMOD0	TZOCNT	P3_1	
Setting value	0	0	002	X	X	Input port (not pulled up)
			012	1		
	0	1	002	X	X	Input port (pulled up)
			012	1		
	1	X	002	X	X	Output port
			012	1		
	X	X	1X	X	X	TZOUT output
			012	0		

X: "0" or "1"

Table 15.19 Port P32/ $\overline{\text{INT}}_2/\text{CNTR}_1$ Setting

Register	PD3	PUR0	TYZMR	TYZOC	P3	Function
Bit	PD3_2	PU06	TYMOD1	TZOCNT	P3_2	
Setting value	0	0	0	1	X	Input port (not pulled up)
	0	1	0	1	X	Input port (pulled up)
	0	0	0	1	X	$\text{CNTR}_1/\overline{\text{INT}}_2$ input
	1	X	0	1	X	Output port
	X	X	1	0	X	CNTR_1 output

X: "0" or "1"

Table 15.20 Port P33/ $\overline{\text{INT}}_3/\text{TCIN}$ Setting

Register	PD3	PUR0	Function
Bit	PD3_3	PU06	
Setting value	0	0	Input port (not pulled up)
	0	1	Input port (pulled up)
	0	0	$\text{TCIN}/\overline{\text{INT}}_3$ input
	1	X	Output port

X: "0" or "1"

Table 15.21 Port P37/TxD10/RxD1 Setting

Register	PD3	PUR0	UCON	U1MR	U1C0	Function
Bit	PD3_7	PU07	TXD1EN	SMD2, SMD0	NCH	
Setting value	0	0	X	002	X	Input port (not pulled up)
	0	1	X	002	X	Input port (pulled up)
	0	0	0	1X	X	RxD1
				X1		
	1	X	X	002	X	Output port
	X	X	1	1X	0	TxD0 output, CMOS output
				X1		
X	X	1	1X	1	TxD10 output, N-channel open output	
			X1			

X: "0" or "1"

Table 15.22 Port P45/ $\overline{\text{INT}}_0$ Setting

Register	PD4	PUR1	INTEN	Function
Bit	PD4_5	PU11	INT0EN	
Setting value	0	0	0	Input port (not pulled up)
	0	1	0	Input port (pulled up)
	0	0	1	$\overline{\text{INT}}_0$ input
	1	X	X	Output port

X: "0" or "1"

Table 15.23 Port XIN/P46, XOUT/P47 Setting

Register	CM1	CM1	CM0	Circuit specification		Function
Bit	CM13	CM10	CM05	Oscillation buffer	Feedback resistance	
Setting value	1	1	1	OFF	OFF	XIN-XOUT oscillatoin stop
	1	0	1	OFF	ON	External input to XIN pin, "H" output from XOUT pin
	1	0	1	OFF	ON	XIN-XOUT oscillatoin stop
	1	0	0	ON	ON	XIN-XOUT oscillatoin
	0	X	X	OFF	OFF	Input port

X: "0" or "1"

15.3 Unassigned Pin Handling

Table 15.24 lists the handling of unassigned pins.

Table 15.24 Unassigned Pin Handling

Pin name	Connection
Ports P0, P1, P30 to P33, P37, P45	<ul style="list-style-type: none"> •After setting for input mode, connect every pin to VSS via a resistor(pull-down) or connect every pin to VCC via a resistor(pull-up) •Set to output mode and leave these pins open^(1, 2)
Ports P46, P47	Connect to VCC via resistor (pull-up) ⁽²⁾
AVCC, VREF	Connect to VCC
AVSS	Connect to VSS

NOTES:

- 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 current 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.
- Connect these unassigned pins to the microcomputer using the shortest wire length (within 2 cm) possible.

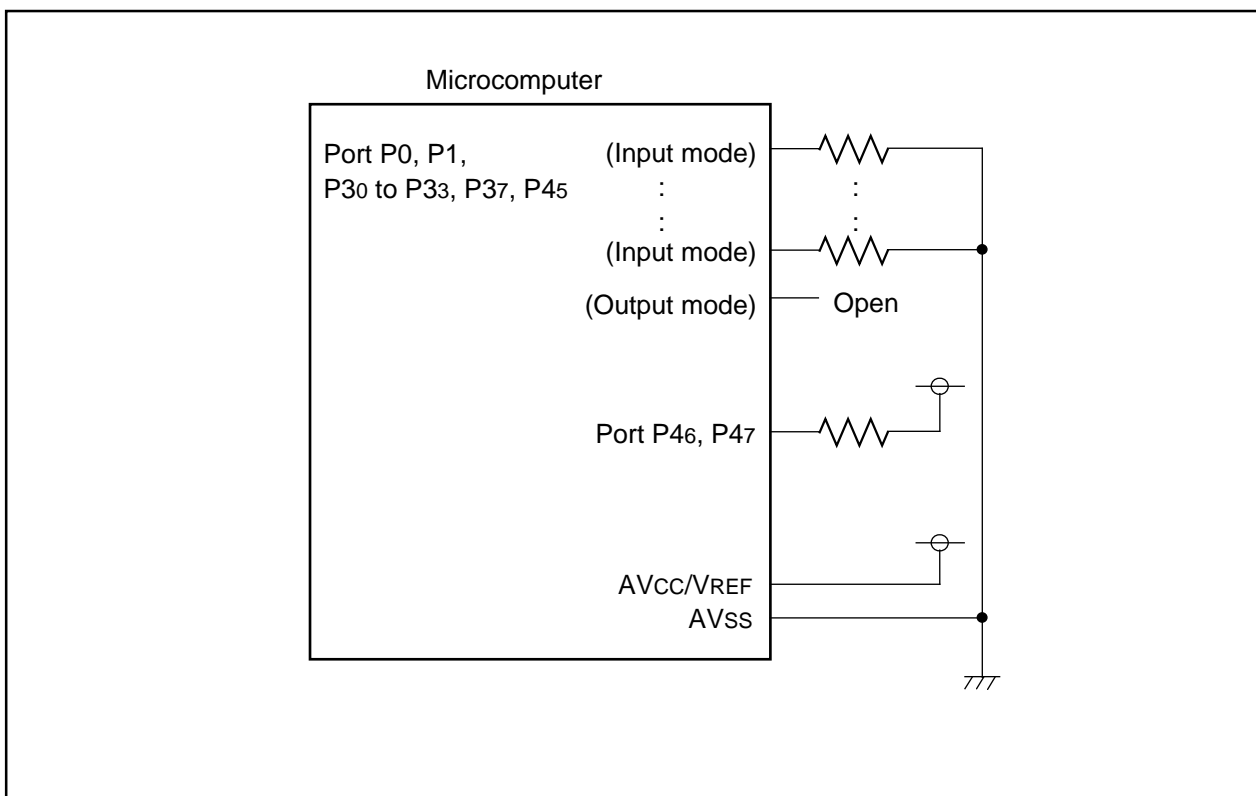


Figure 15.9 Unassigned Pin Handling

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	T _{opr} =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	—	5.5	V	
AV _{cc}	Analog supply voltage		—	V _{cc} (3)	—	V	
V _{ss}	Supply voltage		—	0	—	V	
AV _{ss}	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 P10 to P17	—	—	10	mA	
		P10 to P17	Drive capacity HIGH	—	—	30	mA
			Drive capacity LOW	—	—	10	mA
I _{OL} (avg)	"L" average output current	Except P10 to P17	—	—	5	mA	
		P10 to P17	Drive capacity HIGH	—	—	15	mA
			Drive capacity LOW	—	—	5	mA
f (XIN)	Main clock input oscillation frequency	3.0V ≤ V _{cc} ≤ 5.5V	0	—	16	MHz	
		2.7V ≤ V _{cc} < 3.0V	0	—	10	MHz	

NOTES:

1. V_{cc} = AV_{cc} = 2.7 to 5.5V at T_{opr} = -20 to 85 °C / -40 to 85 °C, unless otherwise specified.
2. The typical values when average output current is 100ms.
3. Hold V_{cc}=AV_{cc}.

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	$\phi_{AD} = 10 \text{ MHz}$, $V_{REF} = V_{CC} = 5.0 \text{ V}$	—	—	± 3	LSB
		8 bit mode	$\phi_{AD} = 10 \text{ MHz}$, $V_{REF} = V_{CC} = 5.0 \text{ V}$	—	—	± 2	LSB
		10 bit mode	$\phi_{AD} = 10 \text{ MHz}$, $V_{REF} = V_{CC} = 3.3 \text{ V}^{(3)}$	—	—	± 5	LSB
		8 bit mode	$\phi_{AD} = 10 \text{ MHz}$, $V_{REF} = V_{CC} = 3.3 \text{ V}^{(3)}$	—	—	± 2	LSB
R_{LADDER}	Ladder resistance		$V_{REF} = V_{CC}$	10	—	40	$k\Omega$
t_{CONV}	Conversion time	10 bit mode	$\phi_{AD} = 10 \text{ MHz}$, $V_{REF} = V_{CC} = 5.0 \text{ V}$	3.3	—	—	μs
		8 bit mode	$\phi_{AD} = 10 \text{ MHz}$, $V_{REF} = V_{CC} = 5.0 \text{ V}$	2.8	—	—	μs
V_{REF}	Reference voltage			—	$V_{CC}^{(4)}$	—	V
V_{IA}	Analog input voltage			0	—	V_{REF}	V
—	A/D operating clock frequency ⁽²⁾	Without sample & hold		0.25	—	10	MHz
		With sample & hold		1.0	—	10	MHz

NOTES:

1. $V_{CC} = AV_{CC} = 2.7$ to 5.5 V at $T_{OPR} = -20$ to $85 \text{ }^\circ\text{C}$ / -40 to $85 \text{ }^\circ\text{C}$, unless otherwise specified.
2. If f_{AD} exceeds 10 MHz , divide the f_{AD} and hold A/D operating clock frequency (ϕ_{AD}) 10 MHz or below.
3. If the AV_{CC} is less than 4.2 V , divide the f_{AD} and hold A/D operating clock frequency (ϕ_{AD}) $f_{AD}/2$ or below.
4. Hold $V_{CC} = V_{REF}$.

Table 16.4 Flash Memory Version Electrical Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
—	Program/erase endurance		100	—	—	times
—	Byte program time		—	50	400	μs
—	Block erase time		—	0.4	9	s
$t_{d(SR-ES)}$	Time delay from suspend request until erase suspend		—	—	8	ms
—	Erase Suspend Request Interval		10	—	—	ms
—	Program, Erase voltage		2.7	—	5.5	V
—	Read voltage		2.7	—	5.5	V
—	Program, Erase temperature		0	—	60	$^\circ\text{C}$
—	Data hold time ⁽²⁾	Ambient temperature = $55 \text{ }^\circ\text{C}$	20	—	—	year

NOTES:

1. $V_{CC1} = AV_{CC} = 2.7$ to 5.5 V at $T_{OPR} = 0$ to $60 \text{ }^\circ\text{C}$, unless otherwise specified.
2. The data hold time includes time that the power supply is off or the clock is not supplied.

Table 16.5 Power Circuit Timing Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Time for internal power supply stabilization during powering-on ⁽²⁾		1	—	2000	μs
$t_{d(R-S)}$	STOP release time ⁽³⁾		—	—	150	μs

NOTES:

1. The measuring condition is $V_{CC} = AV_{CC} = 2.7$ to 5.5 V and $T_{OPR} = 25 \text{ }^\circ\text{C}$.
2. This shows the waiting time until the internal power supply generating circuit is stabilized during powering-on.
3. This shows the time until BCLK starts from the interrupt acknowledgement to cancel stop mode.

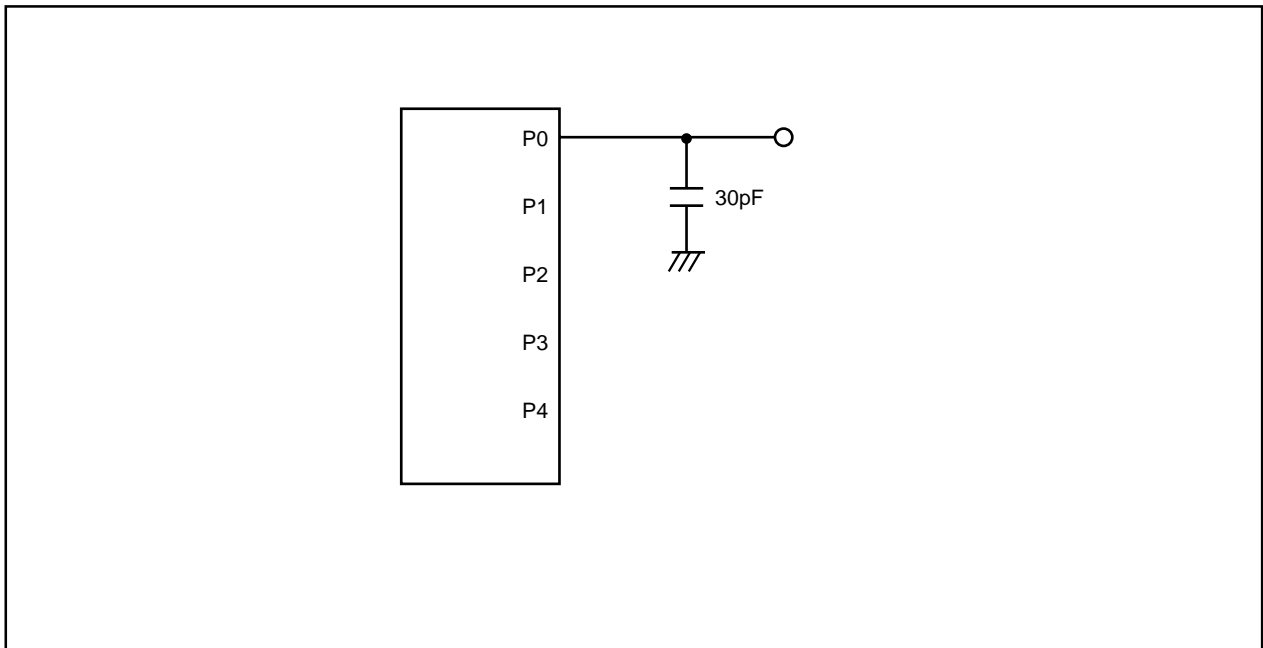


Figure 16.1 Port P0 to P4 measurement circuit

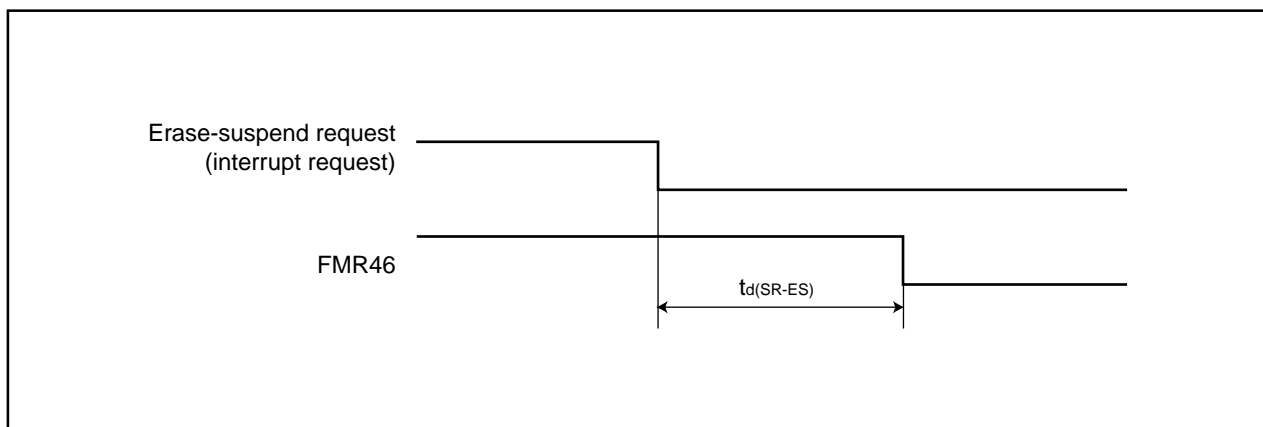


Figure 16.2 Time delay from Suspend Request until Erase Suspend

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

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
VOH	"H" output voltage	Except XOUT	IOH=-5mA	Vcc-2.0		Vcc	V
			IOH=-200μA	Vcc-0.3		Vcc	V
	XOUT		Drive capacity HIGH IOH=-1 mA	Vcc-2.0		Vcc	V
			Drive capacity LOW IOH=-500μA	Vcc-2.0		Vcc	V
VOL	"L" output voltage	Except P10 to P17, XOUT	IOL= 5 mA			2.0	V
			IOL= 200 μA			0.45	V
	P10 to P17		Drive capacity HIGH IOL= 15 mA			2.0	V
			Drive capacity LOW IOL= 5 mA			2.0	V
			Drive capacity LOW IOL= 200 μA			0.45	V
	XOUT		Drive capacity HIGH IOL= 1 mA			2.0	V
Drive capacity LOW IOL=500μA					2.0	V	
VT+-VT-	Hysteresis	INT0, INT1, INT2, INT3, KI0, KI1, KI2, KI3, CNTR0, CNTR1, TCIN, RxD0, RxD1, P45 RESET		0.2		1.0	V
				0.2		2.2	V
IiH	"H" input current		VI=5V			5.0	μA
IiL	"L" input current		VI=0V			-5.0	μA
RPULLUP	Pull-up resistance		VI=0V	30	50	167	kΩ
RiXIN	Feedback resistance	XIN			1.0		MΩ
fRING	On-chip oscillator frequency			40	125	250	kHz
V _{RAM}	RAM retention voltage		At stop mode	2.0			V

NOTES:

1. Referenced to Vcc=AVcc=4.2 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C, f(XIN)=20MHz unless otherwise specified.

Table 16.7 Electrical Characteristics (2) [V_{CC}=5V]

Symbol	Parameter	Measuring condition	Standard			Unit	
			Min.	Typ.	Max.		
I _{CC}	Power supply current (V _{CC} =3.3 to 5.5V) In single-chip mode, the output pins are open and other pins are V _{SS}	High-speed mode	X _{IN} =16 MHz (square wave) On-chip oscillator on=125 kHz No division	—	8	14	mA
			X _{IN} =10 MHz (square wave) On-chip oscillator on=125 kHz No division	—	5	—	mA
		Medium-speed mode	X _{IN} =16 MHz (square wave) On-chip oscillator on=125 kHz Division by 8	—	3	—	mA
			X _{IN} =10 MHz (square wave) On-chip oscillator on=125 kHz Division by 8	—	2	—	mA
		On-chip oscillator mode	Main clock off On-chip oscillator on=125 kHz Division by 8	—	470	900	μA
		Wait mode	Main clock off On-chip oscillator on=125 kHz When a WAIT instruction is executed ⁽¹⁾ Peripheral clock operation	—	40	80	μA
		Wait mode	Main clock off On-chip oscillator on=125 kHz When a WAIT instruction is executed ⁽¹⁾ Peripheral clock off	—	38	76	μA
Stop mode	Main clock off, T _{opr} = 25 °C On-chip oscillator off CM10="1" Peripheral clock off	—	0.8	3.0	μA		

NOTES:

1. Timer Y is operated with timer mode.
2. Referenced to V_{CC} = AV_{CC} = 4.2 to 5.5V at T_{opr} = -20 to 85 °C / -40 to 85 °C, f(X_{IN})=20MHz unless otherwise specified.

Timing requirements (Unless otherwise noted: $V_{CC} = 5V$, $V_{SS} = 0V$ at $T_{opr} = 25\text{ }^{\circ}C$) [$V_{CC}=5V$]**Table 16.8 XIN input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_C(XIN)$	XIN input cycle time	62.5	–	ns
$t_{WH}(XIN)$	XIN input HIGH pulse width	30	–	ns
$t_{WL}(XIN)$	XIN input LOW pulse width	30	–	ns

Table 16.9 CNTR0 input, CNTR1 input, $\overline{INT2}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_C(CNTR0)$	CNTR0 input cycle time	100	–	ns
$t_{WH}(CNTR0)$	CNTR0 input HIGH pulse width	40	–	ns
$t_{WL}(CNTR0)$	CNTR0 input LOW pulse width	40	–	ns

Table 16.10 TCIN input, $\overline{INT3}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_C(TCIN)$	TCIN input cycle time	400 ⁽¹⁾	–	ns
$t_{WH}(TCIN)$	TCIN input HIGH pulse width	200 ⁽²⁾	–	ns
$t_{WL}(TCIN)$	TCIN input LOW pulse width	200 ⁽²⁾	–	ns

NOTES:

1. When using the Timer C capture function, adjust the cycle time above (1/ Timer C count source frequency x 3).
2. When using the Timer C capture function, adjust the pulse width above (1/ Timer C count source frequency x 1.5).

Table 16.11 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_C(CK)$	CLKi input cycle time	200	–	ns
$t_W(CKH)$	CLKi input HIGH pulse width	100	–	ns
$t_W(CKL)$	CLKi input LOW pulse width	100	–	ns
$t_d(C-Q)$	TxDi output delay time	–	80	ns
$t_h(C-Q)$	TxDi hold time	0	–	ns
$t_{su}(D-C)$	RxDi input setup time	35	–	ns
$t_h(C-D)$	RxDi input hold time	90	–	ns

Table 16.12 External interrupt $\overline{INT0}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_W(INH)$	$\overline{INT0}$ input HIGH pulse width	250 ⁽¹⁾	–	ns
$t_W(INL)$	$\overline{INT0}$ input LOW pulse width	250 ⁽²⁾	–	ns

NOTES:

1. When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input HIGH pulse width to the greater value, either (1/ digital filter clock frequency x 3) or the minimum value of standard.
2. When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input LOW pulse width to the greater value, either (1/ digital filter clock frequency x 3) or the minimum value of standard.

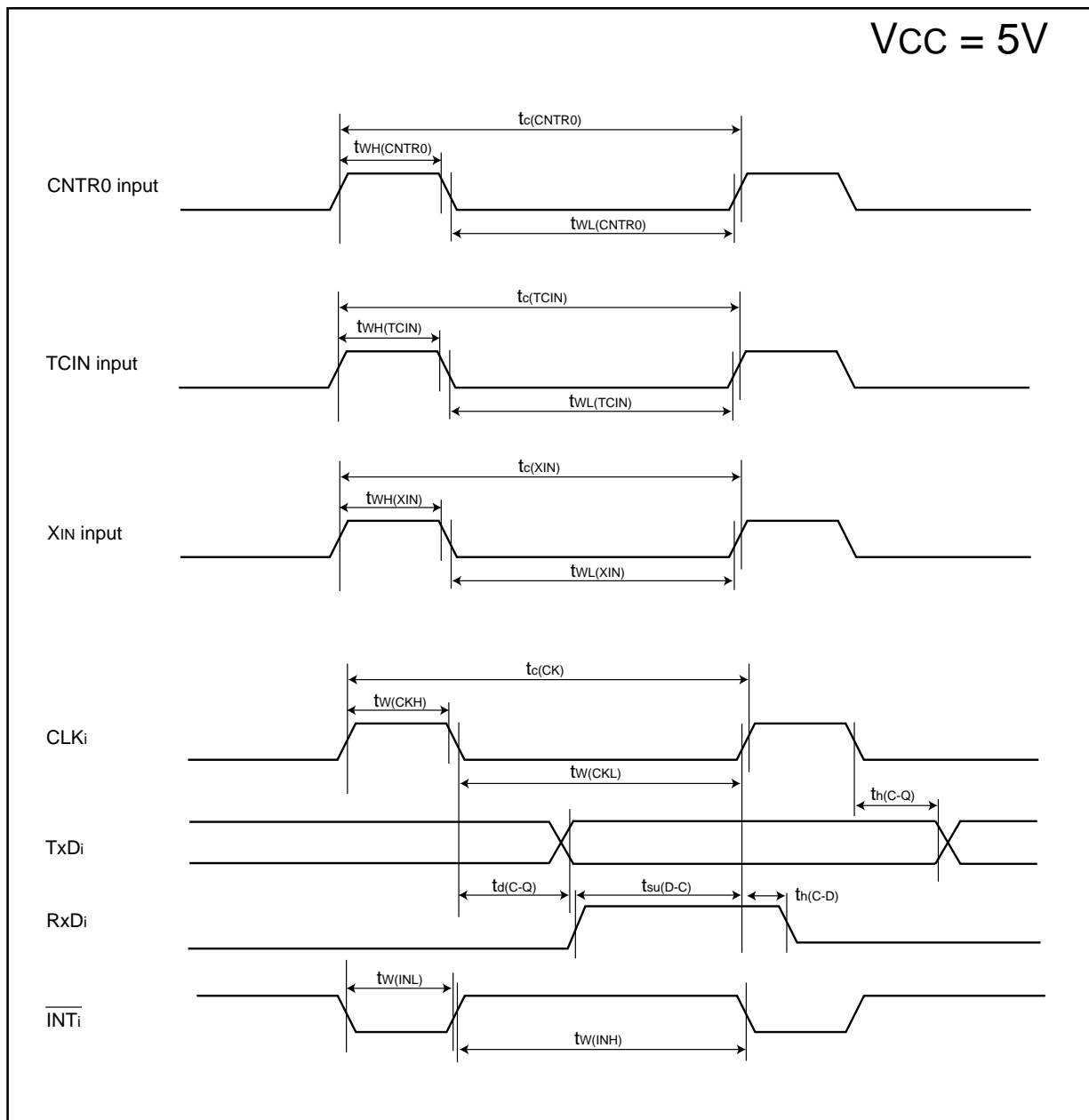


Figure 16.3 Vcc=5V timing diagram

Table 16.13 Electrical Characteristics (3) [Vcc=3V]

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
V _{OH}	"H" output voltage	Except X _{OUT}	I _{OH} =-1mA		V _{CC} -0.5	—	V _{CC}	V
		X _{OUT}	Drive capacity HIGH	I _{OH} =-0.1 mA	—	—	V _{CC}	V
			Drive capacity LOW	I _{OH} =-50 μA	V _{CC} -0.5	—	V _{CC}	V
V _{OL}	"L" output voltage	Except P10 to P17, X _{OUT}	I _{OL} = 1 mA		—	—	0.5	V
		P10 to P17	Drive capacity HIGH	I _{OL} = 2 mA	—	—	0.5	V
			Drive capacity LOW	I _{OL} = 1 mA	—	—	0.5	V
		X _{OUT}	Drive capacity HIGH	I _{OL} = 0.1 mA	—	—	0.5	V
			Drive capacity LOW	I _{OL} =50 μA	—	—	0.5	V
V _{T+} -V _{T-}	Hysteresis	INT0, INT1, INT2, INT3, KI0, KI1, KI2, KI3, CNTR0, CNTR1, TCIN, RxD0, RxD1, P45			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 _{XIN}	Feedback resistance	X _{IN}			—	3.0	—	MΩ
f _{RING}	On-chip oscillator frequency				40	125	250	kHz
V _{RAM}	RAM retention voltage			At stop mode	2.0	—	—	V

NOTES:

1. Referenced to V_{CC}=AV_{CC}=2.7 to 3.3V at T_{opr} = -20 to 85 °C / -40 to 85 °C, f(X_{IN})=10MHz unless otherwise specified.

Table 16.14 Electrical Characteristics (4) [V_{CC}=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) On-chip oscillator on=125 kHz No division	—	7	12	mA
			X _{IN} =10 MHz (square wave) On-chip oscillator on=125 kHz No division	—	5	—	mA
		Medium-speed mode	X _{IN} =16 MHz (square wave) On-chip oscillator on=125 kHz Division by 8	—	2.5	—	mA
			X _{IN} =10 MHz (square wave) On-chip oscillator on=125 kHz Division by 8	—	1.6	—	mA
		On-chip oscillator mode	Main clock off On-chip oscillator on=125 kHz Division by 8	—	420	800	μA
		Wait mode	Main clock off On-chip oscillator on=125 kHz When a WAIT instruction is executed ⁽¹⁾ Peripheral clock operation	—	37	74	μA
		Wait mode	Main clock off On-chip oscillator on=125 kHz When a WAIT instruction is executed ⁽¹⁾ Peripheral clock off	—	35	70	μA
Stop mode	Main clock off, T _{opr} = 25 °C On-chip oscillator off CM10="1" Peripheral clock off	—	0.7	3.0	μA		

NOTES:

1. Timer Y is operated with timer mode.
2. Referenced to V_{CC}=AV_{CC}=2.7 to 3.3V at T_{opr} = -20 to 85 °C / -40 to 85 °C, f(X_{IN})=10MHz unless otherwise specified.

Timing requirements (Unless otherwise noted: $V_{CC} = 3V$, $V_{SS} = 0V$ at $T_{opr} = 25\text{ }^{\circ}C$) [$V_{CC}=3V$]**Table 16.15 XIN input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tC(XIN)	XIN input cycle time	100	–	ns
tWH(XIN)	XIN input HIGH pulse width	40	–	ns
tWL(XIN)	XIN input LOW pulse width	40	–	ns

Table 16.16 CNTR0 input, CNTR1 input, $\overline{INT2}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tC(CNTR0)	CNTR0 input cycle time	300	–	ns
tWH(CNTR0)	CNTR0 input HIGH pulse width	120	–	ns
tWL(CNTR0)	CNTR0 input LOW pulse width	120	–	ns

Table 16.17 TCIN input, $\overline{INT3}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tC(TCIN)	TCIN input cycle time	1200 ⁽¹⁾	–	ns
tWH(TCIN)	TCIN input HIGH pulse width	600 ⁽²⁾	–	ns
tWL(TCIN)	TCIN input LOW pulse width	600 ⁽²⁾	–	ns

NOTES:

1. When using the Timer C capture function, adjust the cycle time above (1 / Timer C count source frequency x 3).
2. When using the Timer C capture function, adjust the pulse width above (1 / Timer C count source frequency x 1.5).

Table 16.18 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tC(CK)	CLKi input cycle time	300	–	ns
tW(CKH)	CLKi input HIGH pulse width	150	–	ns
tW(CKL)	CLKi input LOW pulse width	150	–	ns
td(C-Q)	TxDi output delay time	–	160	ns
th(C-Q)	TxDi hold time	0	–	ns
tsu(D-C)	RxDi input setup time	55	–	ns
th(C-D)	RxDi input hold time	90	–	ns

Table 16.19 External interrupt $\overline{INT0}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tW(INH)	$\overline{INT0}$ input HIGH pulse width	380 ⁽¹⁾	–	ns
tW(INL)	$\overline{INT0}$ input LOW pulse width	380 ⁽²⁾	–	ns

NOTES:

1. When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input HIGH pulse width to the greater value, either (1 / digital filter clock frequency x 3) or the minimum value of standard.
2. When selecting the digital filter by the $\overline{INT0}$ input filter select bit, use the $\overline{INT0}$ input LOW pulse width to the greater value, either (1 / digital filter clock frequency x 3) or the minimum value of standard.

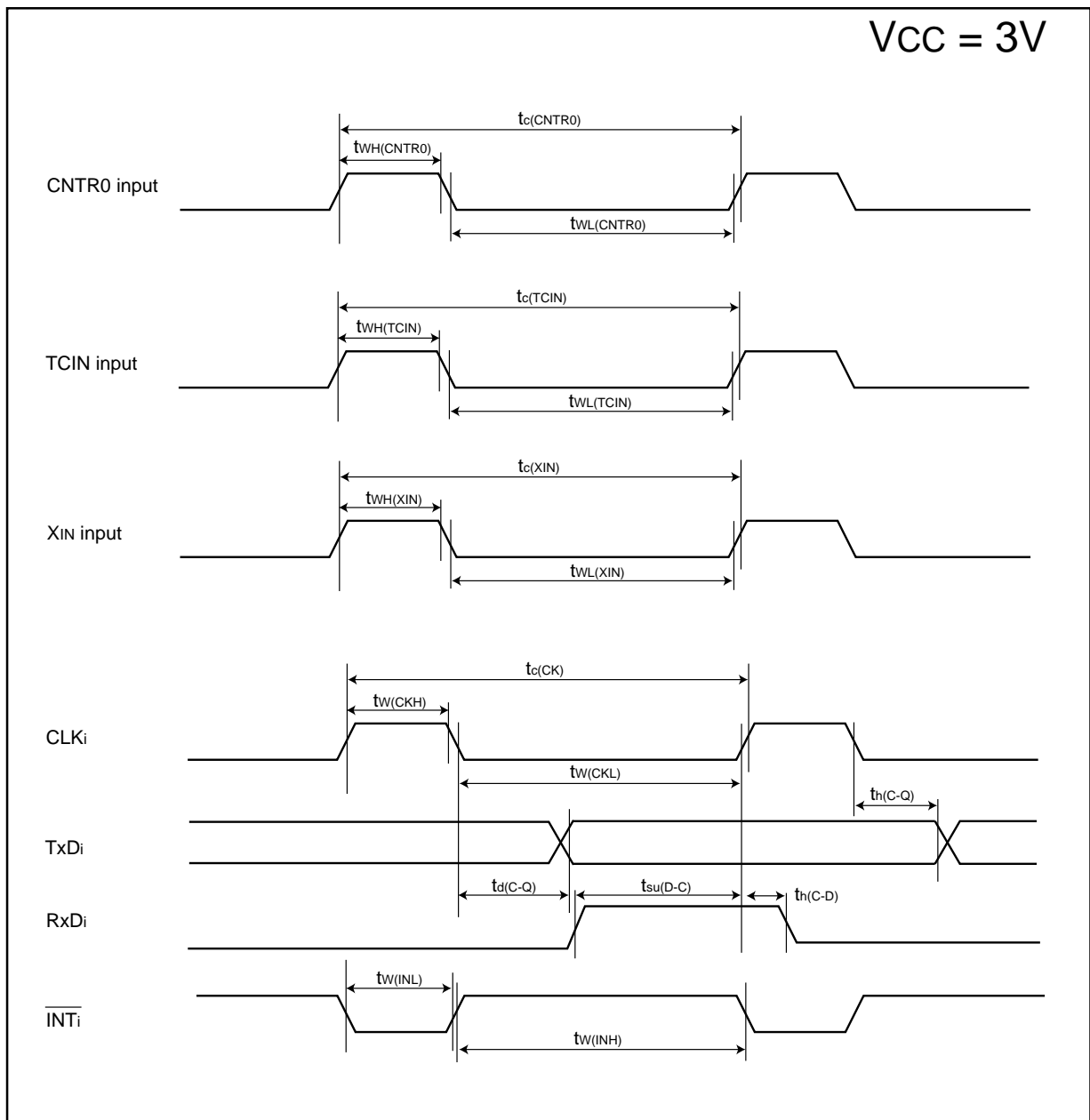


Figure 16.4 $V_{CC}=3V$ timing diagram

17. Flash Memory Version

17.1 Overview

The flash memory version has two modes—CPU rewrite and standard serial I/O—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
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 synchronous 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 I/O 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. The boot ROM area and the user ROM area share the same address, but have an another memory.

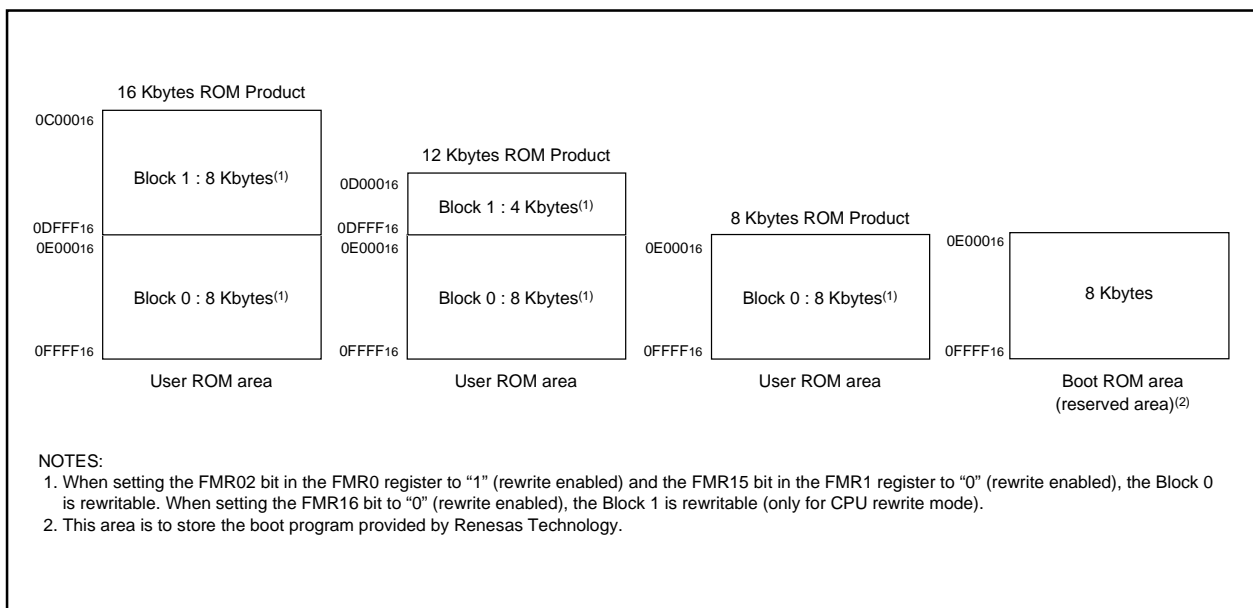


Figure 17.1 Flash Memory Block Diagram

17.3 Functions To Prevent Flash Memory from Rewriting

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

17.3.1 ID Code Check Function

Use this function in standard serial I/O 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 00FFDF₁₆, 00FFE3₁₆, 00FFEB₁₆, 00FFEF₁₆, 00FFF3₁₆, 00FFF7₁₆, and 00FFFB₁₆. Prepare a program in which the ID codes are preset at these addresses and write it in the flash memory.

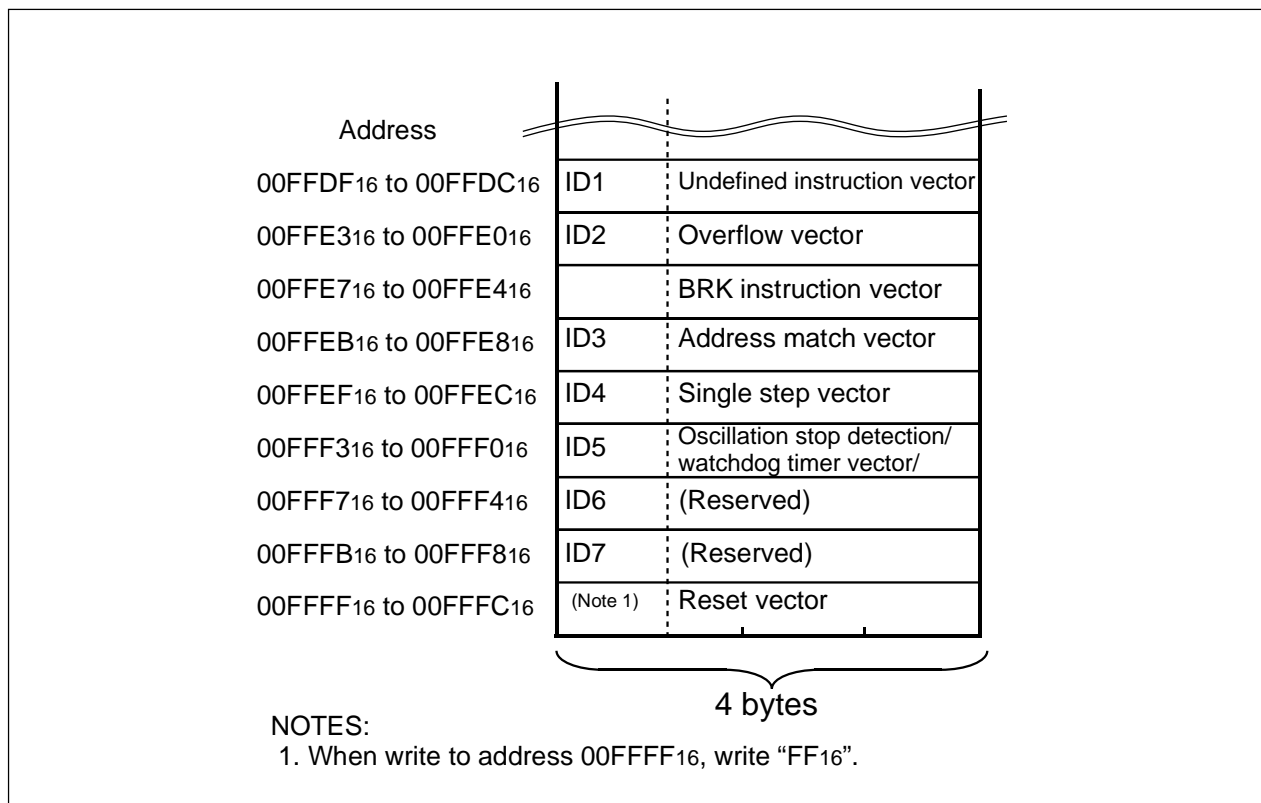


Figure 17.2 Address for ID Code Stored

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	<ul style="list-style-type: 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	<ul style="list-style-type: none"> • 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 during auto-erase, set the FMR40 bit to "1" (erase-suspend enabled) and the FMR41 bit to "1" (erase-suspend requested). Make sure that the FMR46 bit is set to "1" (enables reading) before accessing the user ROM space. The auto-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. After passing td(SR-ES) since the block erase command is executed, an interrupt request can be acknowledged.

The FMR41 bit is automatically set to "1" (erase-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 register. Figure 17.4 shows the FMR1 and FMR4 registers.

- **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 on-chip oscillator mode (main clock stop).

Figure 17.6 shows a flow chart to be followed before and after entering on-chip 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” (erase-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”(disables reading) during auto-erase execution and set to “1”(enables reading) during erase-suspend mode. Do not access to the flash memory when this bit is set to “0”.

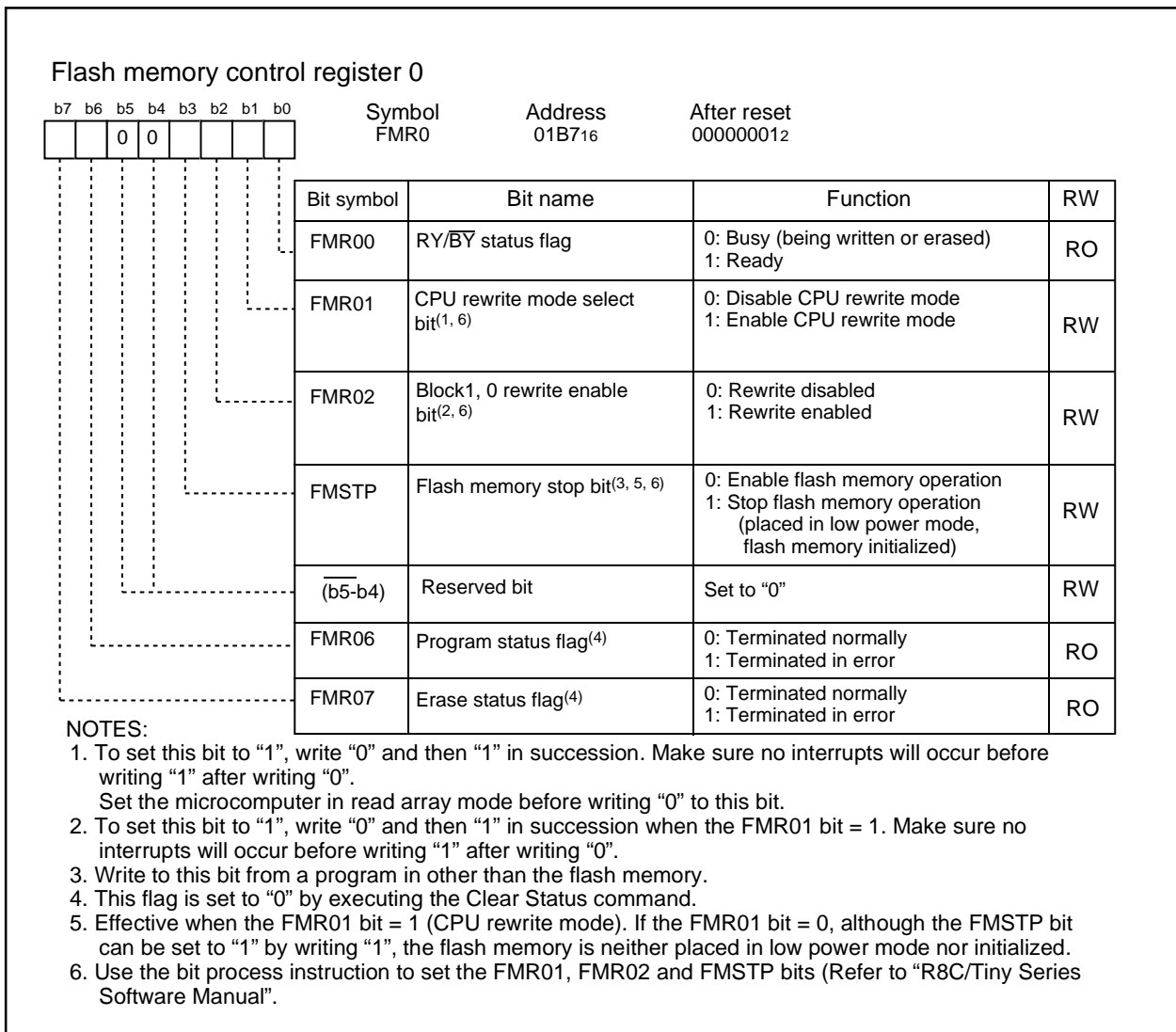


Figure 17.3 FMR0 Register

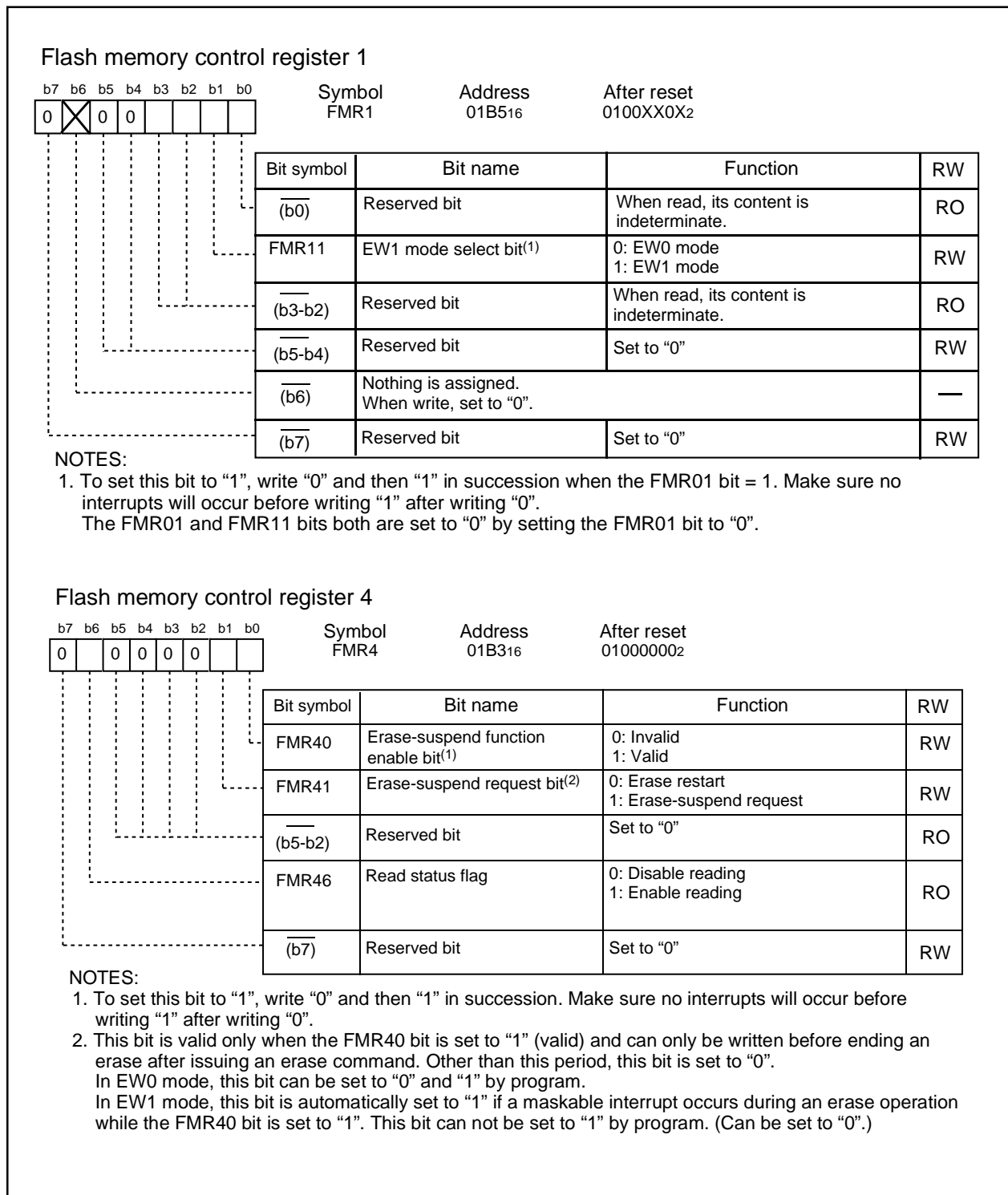


Figure 17.4 FMR1 and FMR4

Figures 17.5 shows the timing on suspend operation.

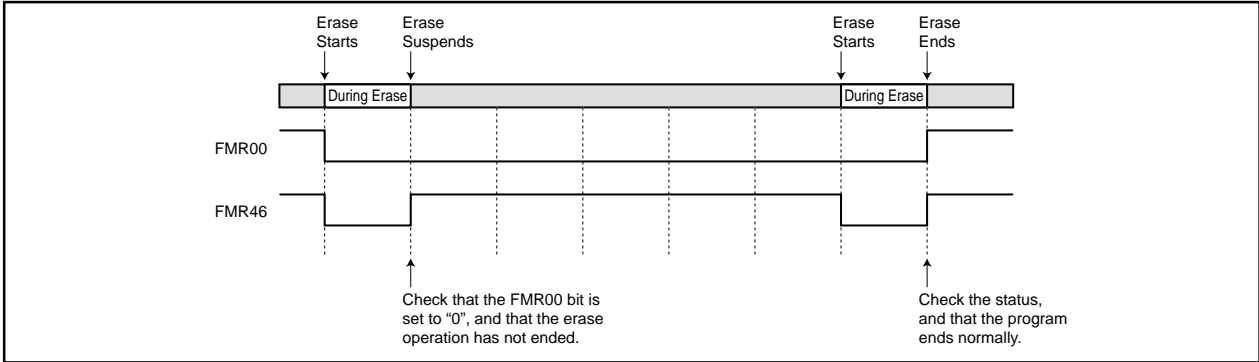


Figure 17.5 Timing on Suspend Operation

Figures 17.6 and 17.7 show the setting and resetting of EW0 mode and EW1 mode, respectively.

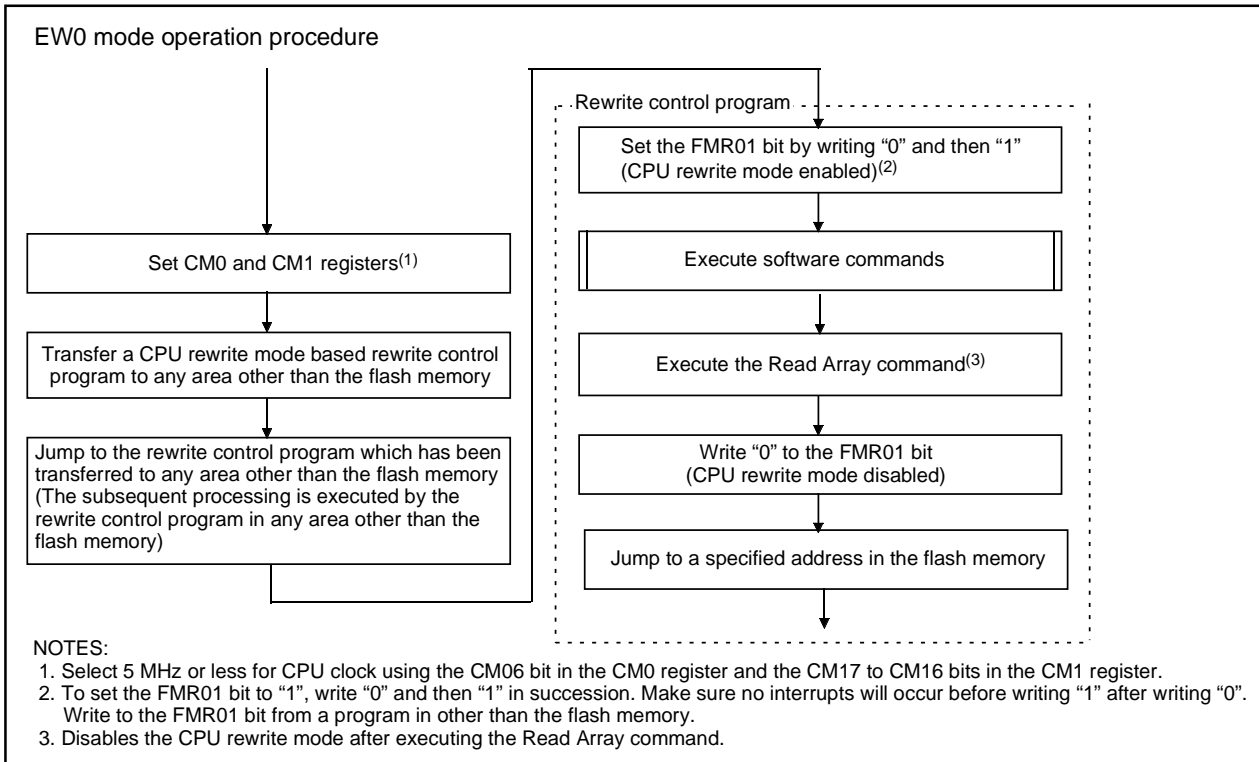


Figure 17.6 Setting and Resetting of EW0 Mode

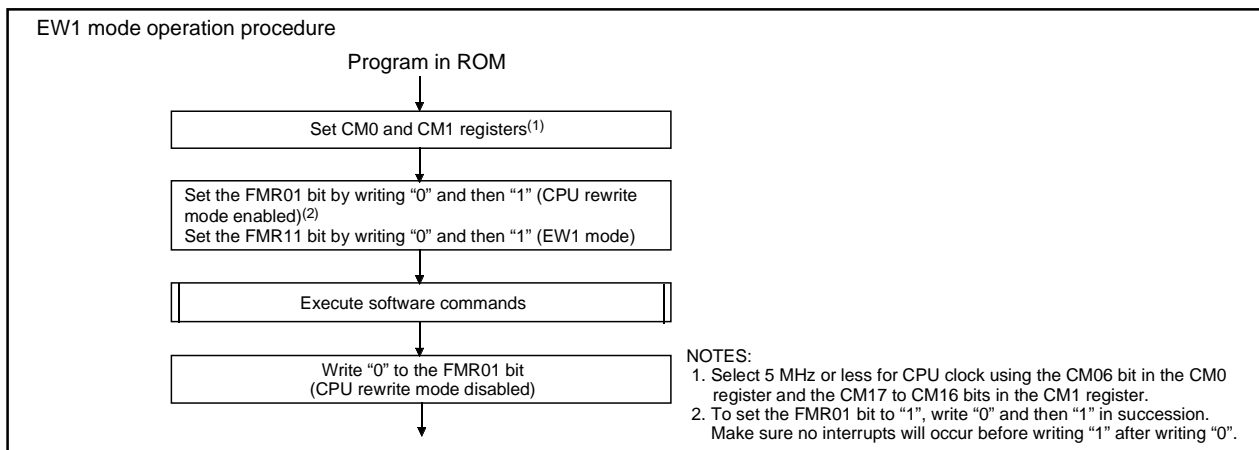


Figure 17.7 Setting and Resetting of EW1 Mode

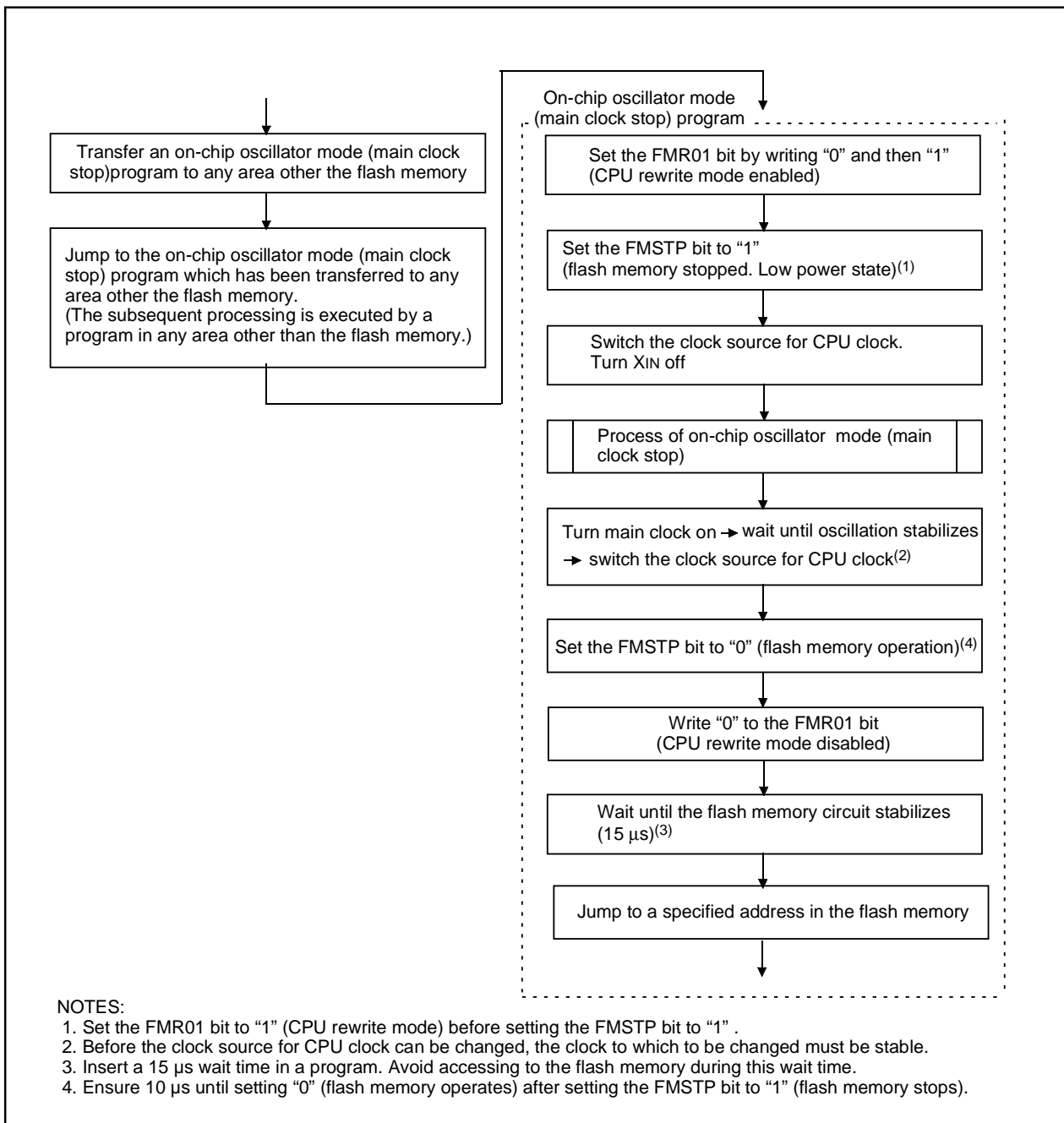


Figure 17.8 Process to Reduce Power Consumption in On-Chip Oscillator Mode (Main Clock Stop)

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 (D7 to D0)	Mode	Address	Data (D7 to D0)
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 (D7 to D0)

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: Given block address

X: Any address in the user ROM area

• Read Array Command

This command reads the flash memory.

Writing 'FF₁₆' 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 '70₁₆' 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 '50₁₆' 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 '4016' 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 17.4.5, "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 array command is written next. The result of auto programming can be known by reading the status register after auto programming has finished.

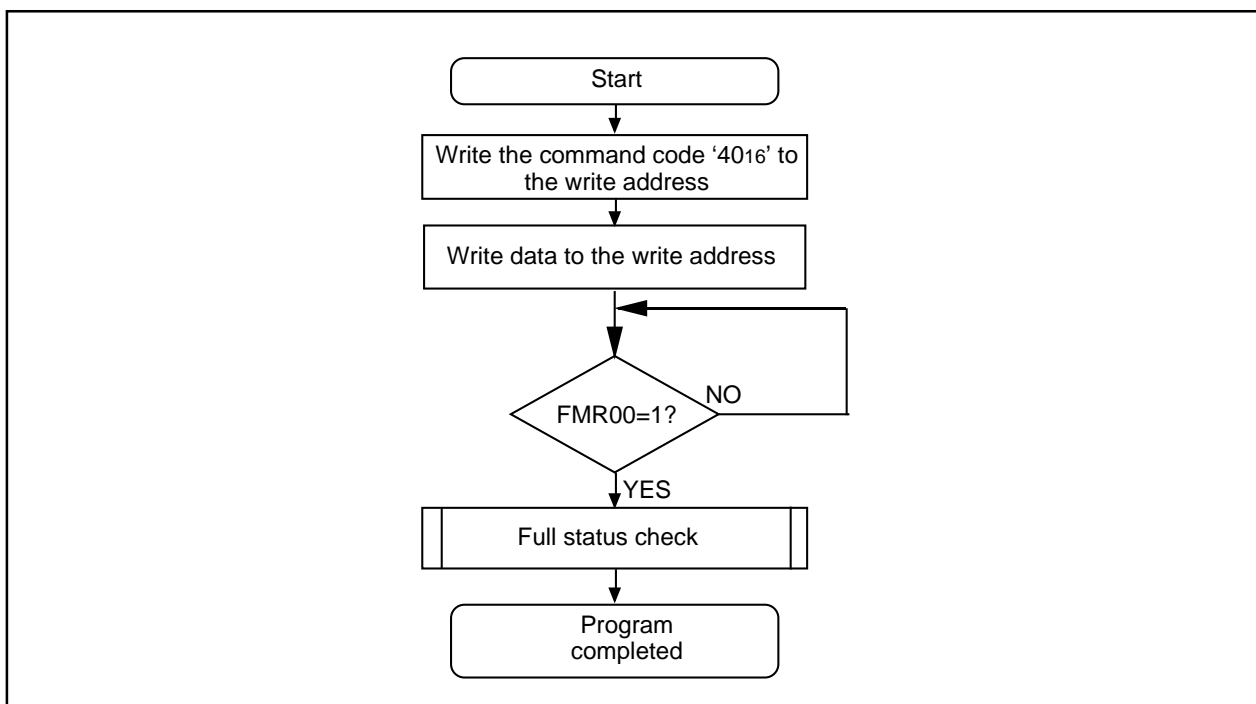


Figure 17.9 Program Flow Chart

• Block Erase

Write '2016' in the first bus cycle and write 'D016' to the given 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.

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 17.4.5, "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 17.10 shows an example of a block erase flowchart when the erase-suspend function is not used. Figure 17.11 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.

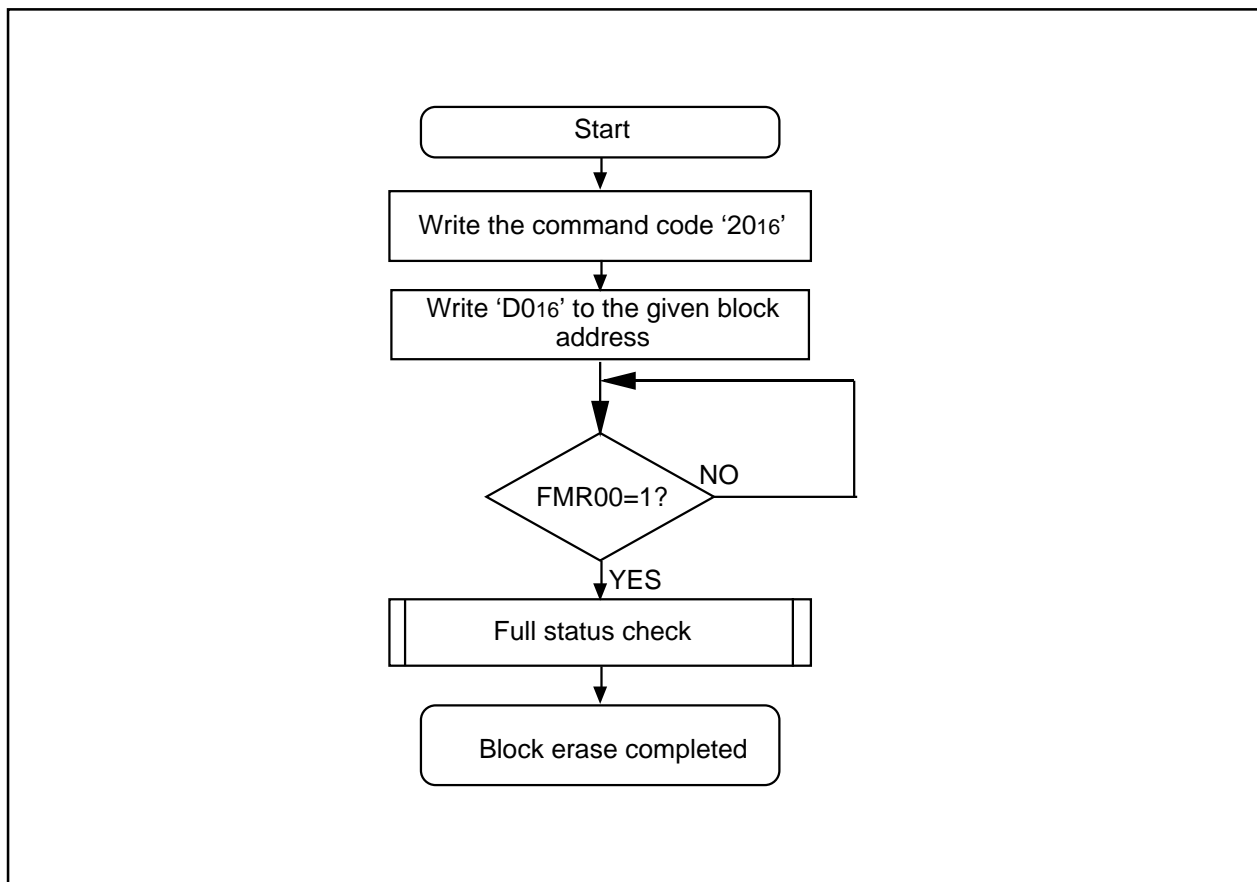


Figure 17.10 Block Erase Flow Chart (When Not Using Erase-suspend Function)

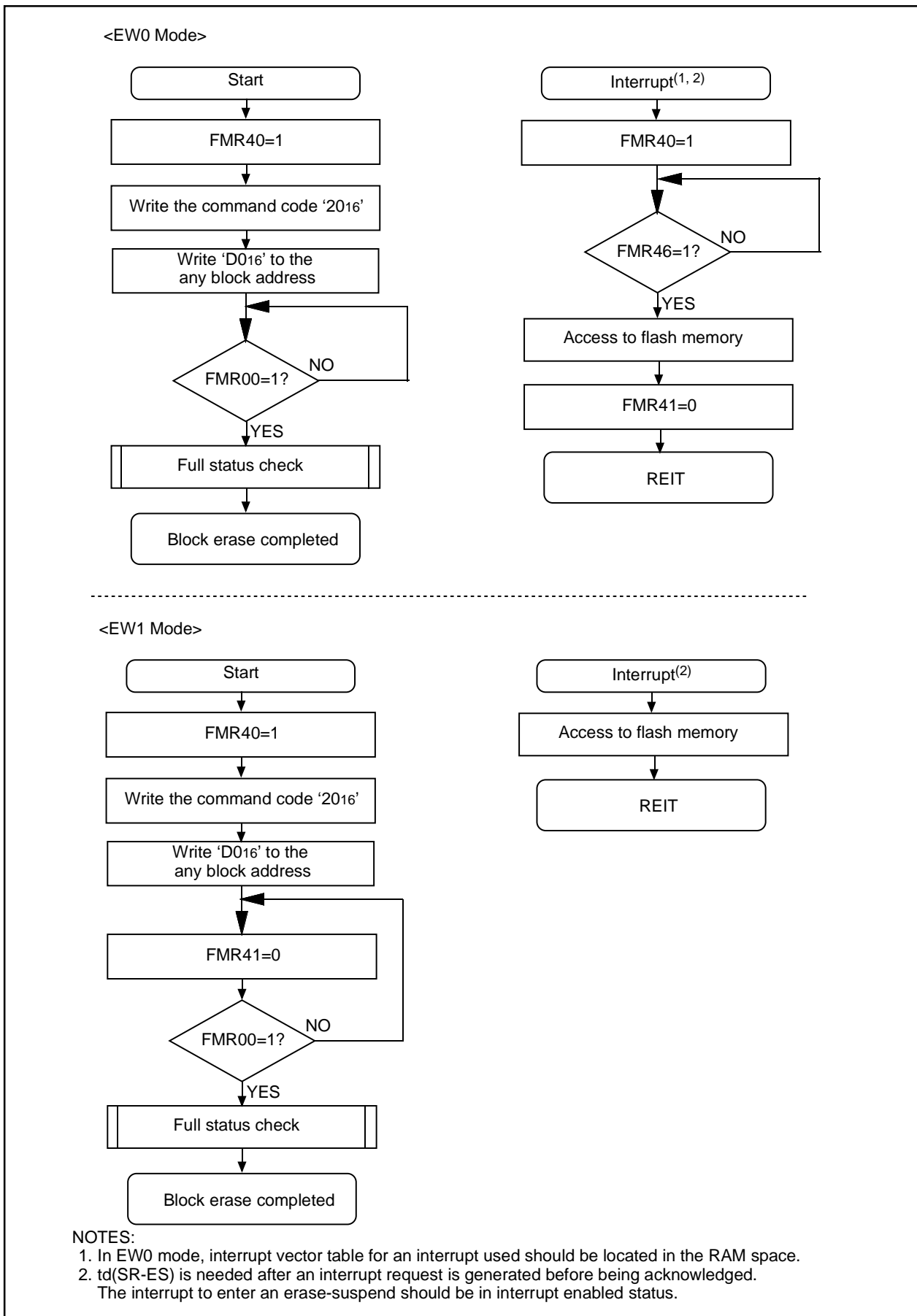


Figure 17.11 Block Erase Command (When Using Erase-suspend Function)

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.

- **Sequence 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.

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

Refer to Section 17.4.5, "Full Status Check."

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

Refer to Section 17.4.5, "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 17.12 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	<ul style="list-style-type: none"> • 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 'D016' or 'FF16')⁽¹⁾ • When executing the program command or block erase command while rewriting is disabled using the FMR02 bit in the FMR0 register, the FMR15 or FMR16 bit in the FMR1 register. • When inputting and erasing the address in which the Flash memory is not allocated during the erase command input. • When executing to erase the block which disables rewriting during the erase command input. • When inputting and writing the address in which the Flash memory is not allocated during the write command input. • When executing to write the block which disables rewriting during the write command input.
1	0	Erase error	<ul style="list-style-type: none"> • When the Block Erase command was executed but not automatically erased correctly
0	1	Program error	<ul style="list-style-type: none"> • When the Program command was executed but not automatically programmed correctly.

NOTES:

1. Writing 'FF16' 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.

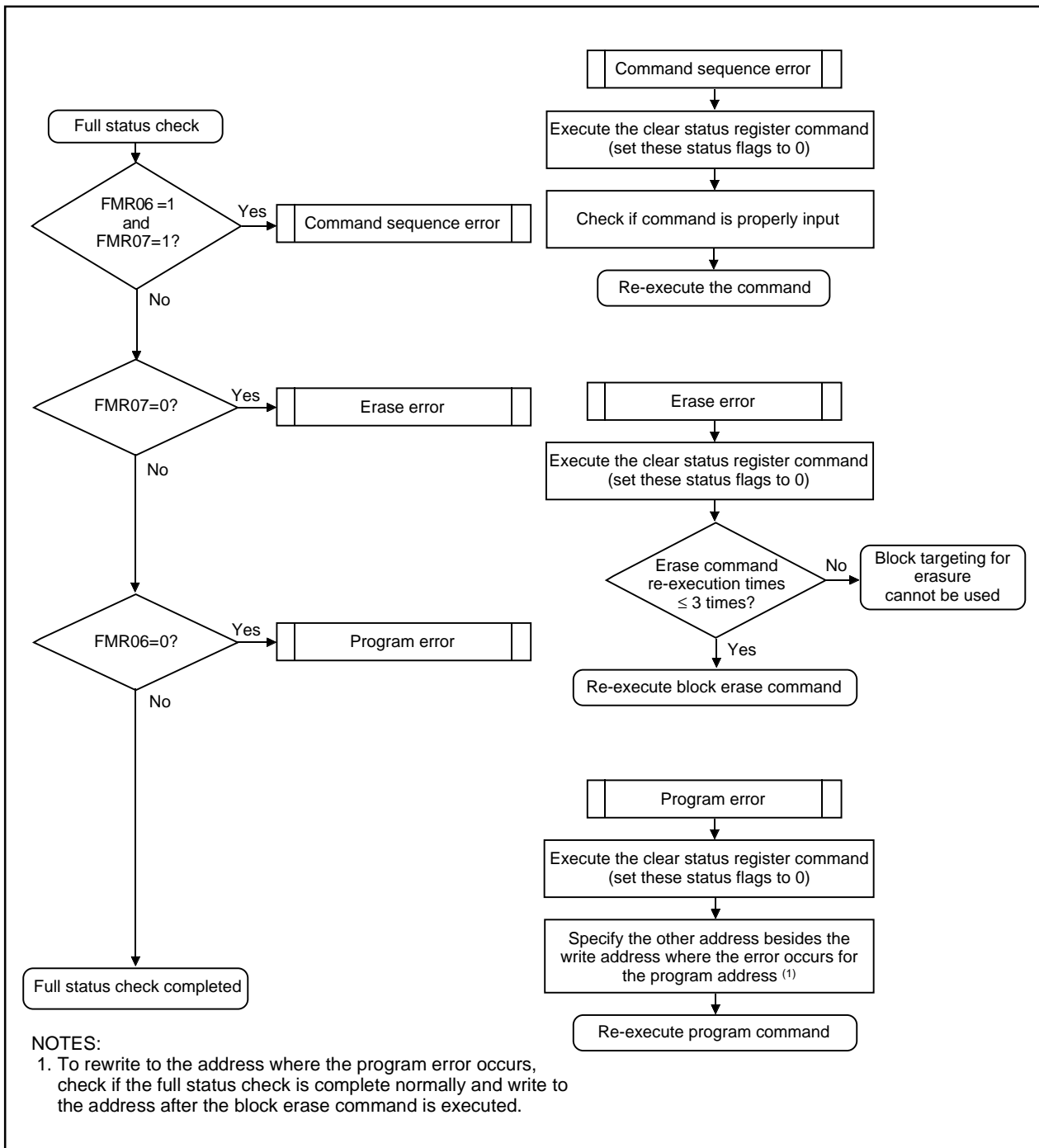


Figure 17.12 Full Status Check and Handling Procedure for Each Error

17.5 Standard Serial I/O Mode

In standard serial I/O mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer suitable for this microcomputer. Standard serial I/O mode has standard serial I/O mode 1 of the clock synchronous serial and standard serial I/O mode 2 of the clock asynchronous serial. Refer to "Appendix 2 Connecting Examples for Serial Writer and On-chip Debugging Emulator". 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 17.13 to 17.15 show pin connections for standard serial I/O 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 μ F) to Vss.
RESET	Reset input	I	Reset input 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. When using the main clock in standard serial I/O mode 1, connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. When not using the main clock in standard serial I/O mode 1, connect this pin to Vcc via a resistor (pull-up).
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	I	Serial data input pin

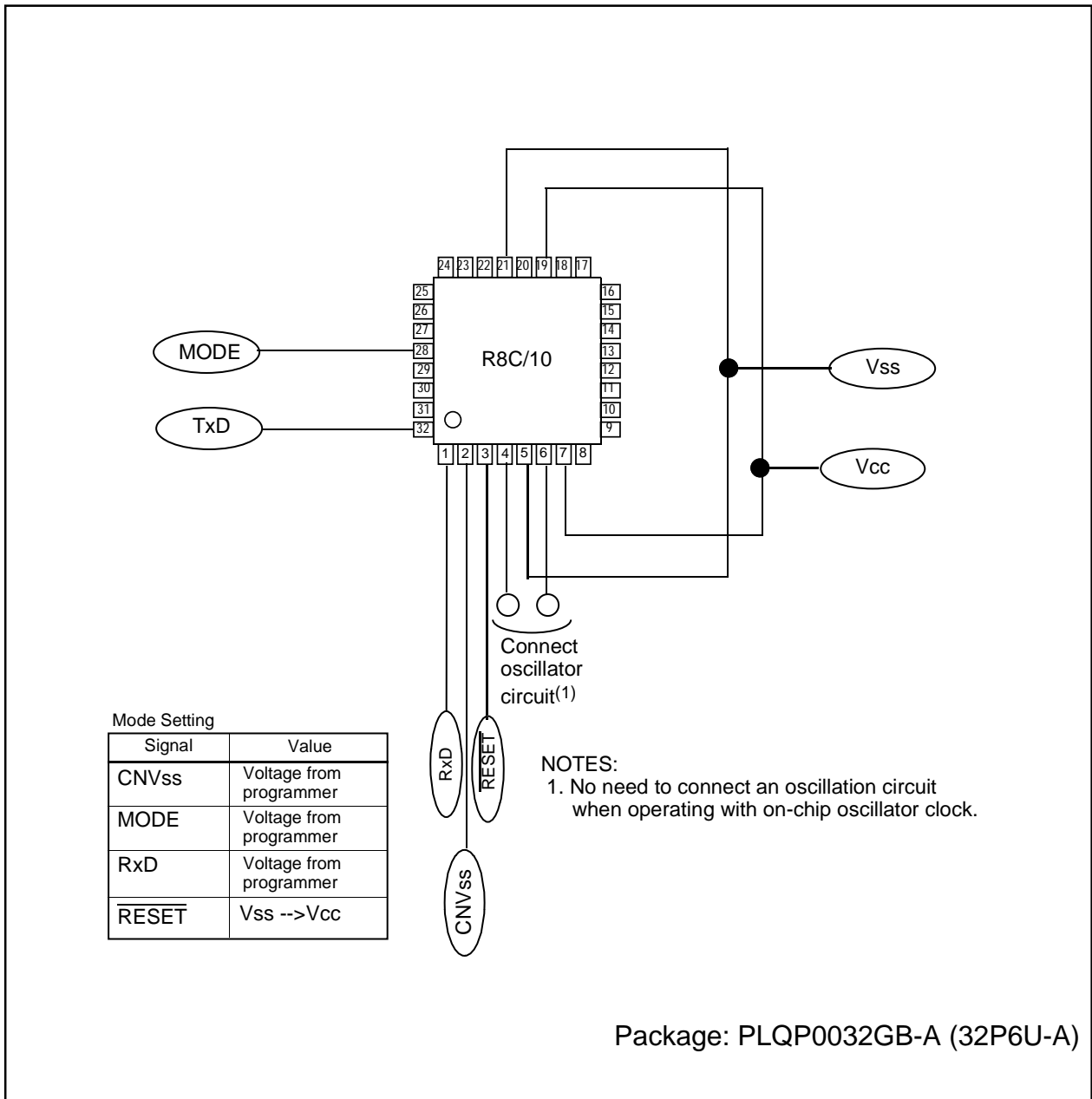


Figure 17.13 Pin Connections for Standard Serial I/O Mode

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

Figures 17.14 and 17.15 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.

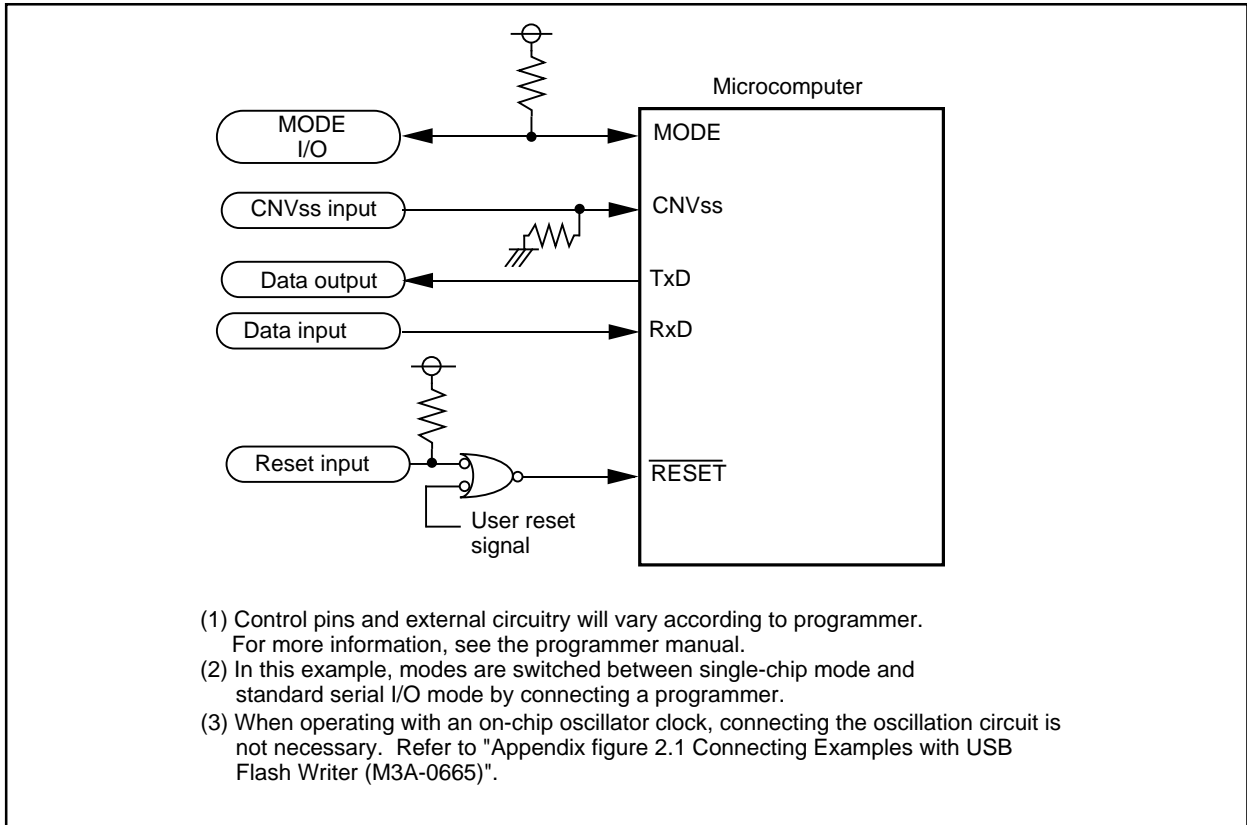


Figure 17.14 Circuit Application in Standard Serial I/O Mode 1

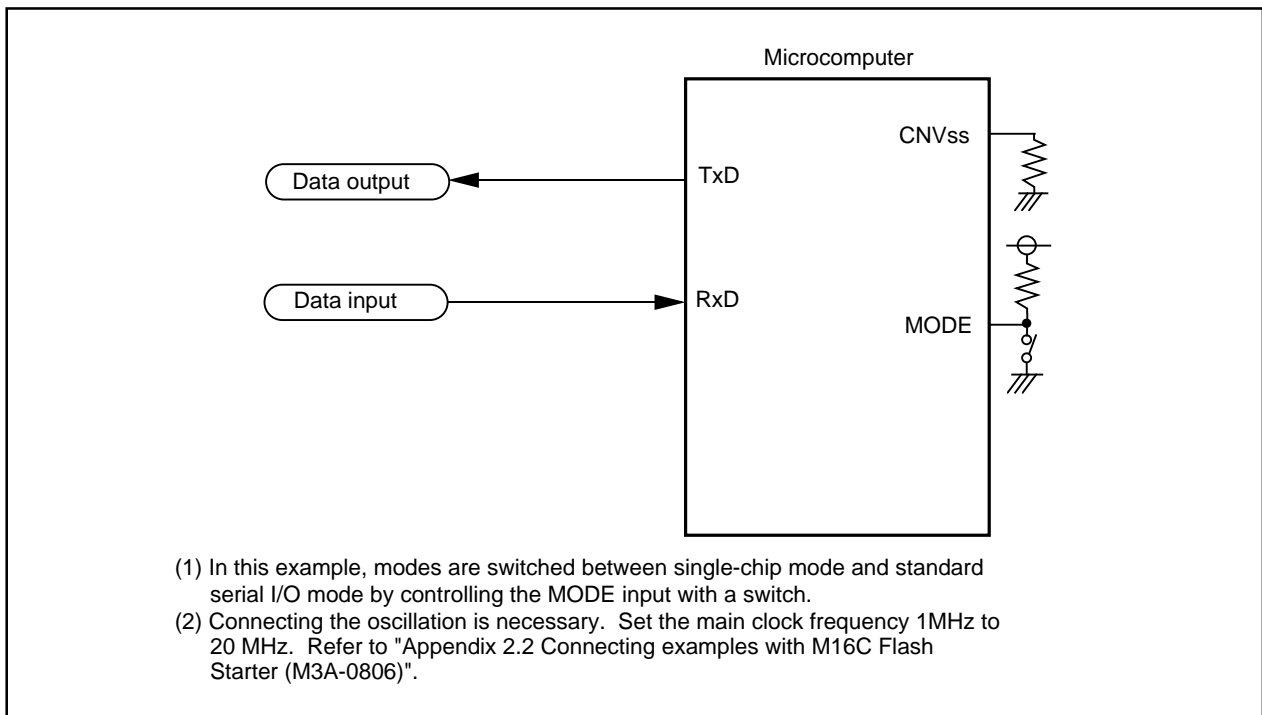


Figure 17.15 Circuit Application in Standard Serial I/O Mode 2

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 0C000₁₆ to 0C7FF₁₆ 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 setting the address match interrupt (the registers of AIER, RMAD0, RMAD1 and the fixed vector tables) 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 Interface" for the details of UART1. Also, avoid using the UART1 and the functions (P0₀/AN₇ and P3₇) 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

19.1.1 Stop Mode

When entering stop mode, set the CM10 bit to “1” (stop mode) after setting the FMR01 bit to “0” (CPU rewrite mode disabled). The instruction queue pre-reads 4 bytes from the instruction which sets the CM10 bit in the CM1 register to “1” (stop mode) and the program stops. Insert at least 4 NOP instructions after inserting the JMP.B instruction immediately after the instruction which sets the CM10 bit to “1”.

Use the next program to enter stop mode.

- Program of entering stop mode

```

        BCLR    1, FMR0    ; CPU rewrite mode disabled
        BSET    0, PRCR    ; Protect exited
        BSET    0, CM1     ; Stop mode
        JMP.B   LABEL_001
LABEL_001:
        NOP
        NOP
        NOP
        NOP

```

19.1.2 Wait Mode

When entering wait mode, execute the WAIT instruction after setting the FMR01 bit to “0” (CPU rewrite mode disabled). The instruction queue pre-reads 4 bytes from the WAIT instruction and the program stops. Insert at least 4 NOP instructions after the WAIT instruction.

Also, the value in the specific internal RAM area may be rewritten when exiting wait mode if writing to the internal RAM area before executing the WAIT instruction and entering wait mode. The area for a maximum of 3 bytes is rewritten from the following address of the internal RAM in which the writing is performed before the WAIT instruction. If this causes a problem, avoid by inserting the JMP.B instruction between the writing instruction to the internal RAM area and WAIT instruction as shown in the following program example.

- Example to execute WAIT instruction

```

Program Example  MOV.B   #055h,0601h    ; Write to internal RAM area
                ...
                JMP.B   LABEL_001
LABEL_001 :
                FSET    I                ; Interrupt enabled
                BCLR    1,FMR0          ; CPU rewrite mode disabled
                WAIT                    ; Wait mode
                NOP
                NOP
                NOP
                NOP

```

When accessing any area other than the internal RAM area between the writing instruction to the internal RAM area and execution of the WAIT instruction, this situation will not occur.

19.2 Interrupt

19.2.1 Reading Address 00000₁₆

Do not read the address 00000₁₆ by a program. When a maskable interrupt request is acknowledged, the CPU reads interrupt information (interrupt number and interrupt request level) from 00000₁₆ in the interrupt sequence. At this time, the acknowledged interrupt IR bit is set to "0".

If the address 00000₁₆ is read by 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 an interrupt is acknowledged. The SP is set to "0000₁₆" after reset. Therefore, if an interrupt is acknowledged before setting any value in the SP, the program may run out of control.

19.2.3 External Interrupt and Key Input Interrupt

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

19.2.4 Watchdog Timer Interrupt

Reset the watchdog timer after a watchdog timer interrupt is generated.

19.2.5 Changing Interrupt Factor

The IR bit in the interrupt control register may be set to “1” (interrupt requested) when the interrupt factor is changed. When using an interrupt, set the IR bit to “0” (interrupt not request) after changing the interrupt factor. In addition, the changes of interrupt factors include all elements that change the interrupt factors assigned to individual software interrupt numbers, polarities, and timing. Therefore, when a mode change of the peripheral functions involves interrupt factors, edge polarities, and timing, set the IR bit to “0” (interrupt not requested) after the change. Refer to each peripheral function for the interrupts caused by the peripheral functions.

Figure 19.1 shows an Example of Procedure for Changing Interrupt Factor.

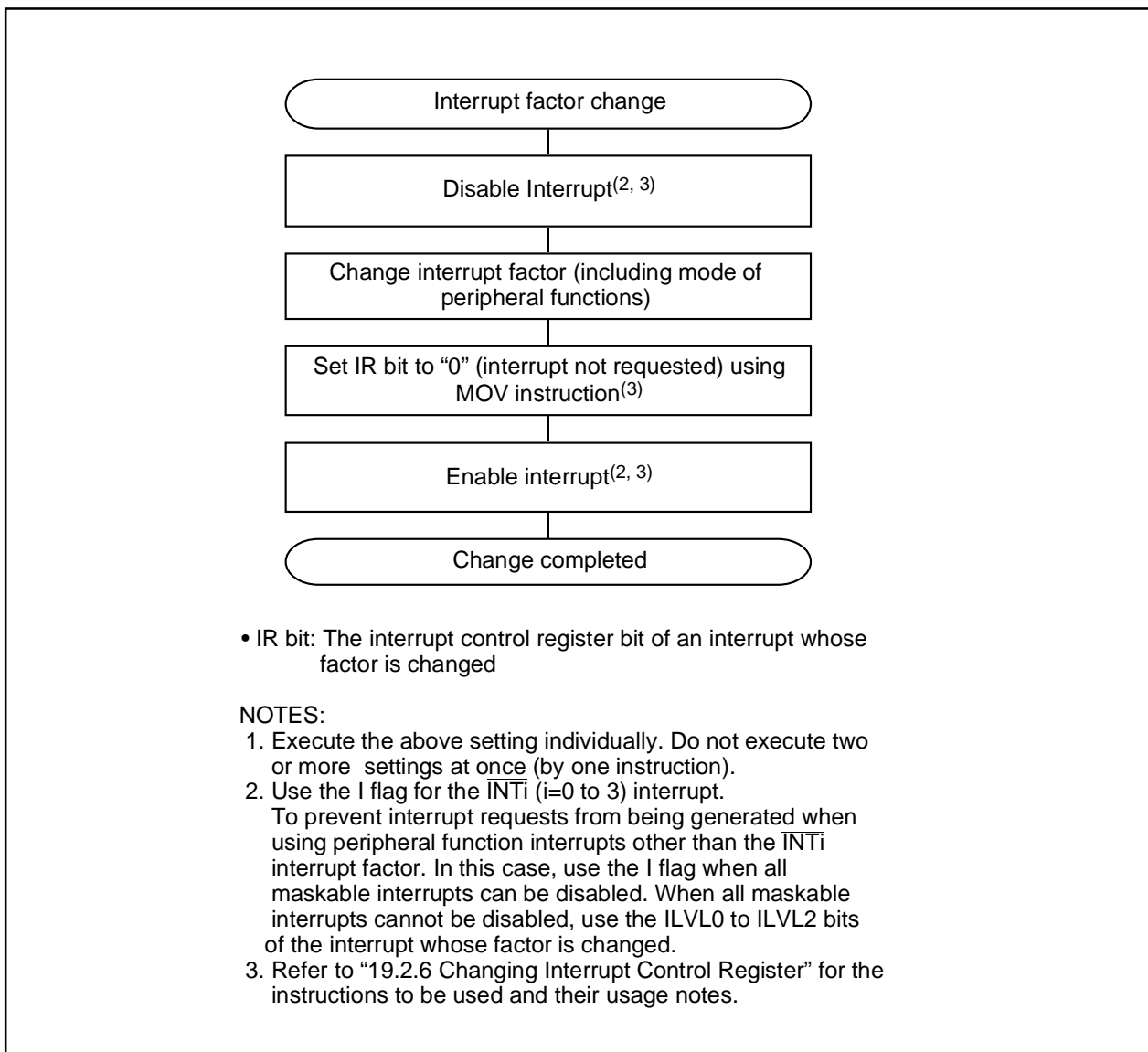


Figure 19.1 Example of Procedure for Changing Interrupt Factor

19.2.6 Changing Interrupt Control Register

(1) Each interrupt control register can only be changed while interrupt requests corresponding to that register are not generated. If interrupt requests may be generated, disable the interrupts before changing the interrupt control register.

(2) When changing any interrupt control register after disabling interrupts, be careful with the instruction to be used.

When Changing Any Bit 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), and the interrupt request may be ignored. If this causes a problem, use the following instructions to change the register.

Instructions to use: AND, OR, BCLR, BSET

When Changing IR Bit

If the IR bit is set to "0" (interrupt not requested), it may not be set to "0" depending on the instruction used. 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 preventing 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 Clock Generation Circuit

19.3.1 Oscillation Stop Detection Function

Since the oscillation stop detection function cannot be used if the main clock frequency is below 2MHz, set the OCD1 to OCD0 bits to "002" (oscillation stop detection function disabled).

19.3.2 Oscillation Circuit Constants

Ask the maker of the oscillator to specify the best oscillation circuit constants on your system.

19.4 Timers

19.4.1 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.

19.4.2 Timer X

- (1) Do not rewrite the TXMOD0 to TXMOD1 bits, the TXMOD2 bit and TXS bit simultaneously.
- (2) 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
```

- (3) 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.
- (4) The prescaler X underflow which is generated for the first time after the count start may cause that the TXEDG bit is set to "1". When using the pulse period measurement mode, leave more than two periods of the prescaler X right after count starts and set the TXEDG bit to "0".

19.4.3 Timer Y

- (1) Do not rewrite the TYMOD0 and TYS bits simultaneously.

19.4.4 Timer Z

- (1) Do not rewrite the TZMOD0 to TZMOD1 bits and the TZS bit simultaneously.
- (2) In programmable one-shot generation mode and programmable wait one-shot generation mode, when setting the TZS bit in the TC register to "0" (stops counting) or setting the TZOS bit in the TZOC register to "0" (stops one-shot), 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.

19.4.5 Timer C

- (1) Access the TC, TM0 and TM1 registers in 16-bit units.
This prevents the timer value from being updated between the low-order byte and high-order byte are being read.

Example (when Timer C is read):

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

19.5 Serial Interface

- (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 UORB 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
```

19.6 A/D Converter

- (1) When writing to each bit but except bit 6 in the ADCON0 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 μ s 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 or the ADST bit in the ADCON0 register can indicate whether the A/D conversion is completed or not.
- (4) In repeat mode, the undivided main clock must be used for the CPU clock.
- (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 set to "0" in a program, ignore the value of AD register.
- (6) A 0.1 μ F capacitor should be connected between the AVcc/VREF pin and AVss pin.

19.7 Flash Memory Version

19.7.1 CPU Rewrite Mode

- Operation Speed

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

- Instructions Disabled Against Use

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

- How to Access

Write "0" to the corresponding bits before writing "1" when setting the FMR01, FMR02, and FMR11 bits to "1". Do not generate an interrupt between writing "0" and "1".

- Rewriting User ROM Area

In EW0 mode, if the power supply voltage drops while rewriting any block in which the rewrite control program is stored, the flash memory may not be able to be rewritten because the rewrite control program cannot be rewritten correctly. In this case, use standard serial I/O mode.

- Reset Flash Memory

Since the CPU stops and cannot return when setting the FMSTP bit in the FMR0 register to "1" (flash memory stops) during erase suspend in EW1 mode, do not set the FMSTP bit to "1".

- Entering Stop Mode or Wait Mode

Do not enter stop mode or wait mode during erase-suspend.

● Interrupt

Table 19.1 list the Interrupt in EW0 Mode and Table 19.2 lists the Interrupt in EW1 Mode.

Table 19.1 Interrupt in EW0 Mode

Mode	Status	When maskable interrupt request is acknowledged	When watchdog timer, oscillation stop detection, and voltage detection interrupt request are acknowledged
EW0	During auto- matic erasing	Any interrupt can be used by allocating a vector to RAM	Once an interrupt request is acknowledged, the auto-programming or auto-erasing is forcibly stopped and resets the flash memory. An interrupt process starts after the fixed period and the flash memory restarts. Since the block during the auto-erasing or the address during the auto-programming is forcibly stopped, the normal value may not be read. Execute the auto-erasing again and ensure the auto-erasing is completed normally. Since the watchdog timer does not stop during the command operation, the interrupt request may be generated. Reset the watchdog timer regularly.
	Automatic writing		

NOTES:

1. Do not use the address match interrupt while the command is executed because the vector of the address match interrupt is allocated on ROM.
2. Do not use the non-maskable interrupt while Block 0 is automatically erased because the fixed vector is allocated Block 0.

Table 19.2 Interrupt in EW1 Mode

Mode	Status	When maskable interrupt request is acknowledged	When watchdog timer, oscillation stop detection and voltage detection interrupt request area acknowledged
EW1	During automatic erasing (erase-suspend function is enabled)	The auto-erasing is suspended and the interrupt process is executed. The auto-erasing can be restarted by setting the FMR41 bit in the FMR4 register to "0" (erase restart) after the interrupt process completes	Once an interrupt request is acknowledged, the auto-programming or auto-erasing is forcibly stopped and resets the flash memory. An interrupt process starts after the fixed period and the flash memory restarts. Since the block during the auto-erasing or the address during the auto-programming is forcibly stopped, the normal value may not be read. Execute the auto-erasing again and ensure the auto-erasing is completed normally. Since the watchdog timer does not stop during the command operation, the interrupt request may be generated. Reset the watchdog timer regularly using the erase-suspend function.
	During automatic erasing (erase-suspend function is disabled)	The auto-erasing has a priority and the interrupt request acknowledgement is waited. The interrupt process is executed after the auto-erasing completes	
	Auto programming	The auto-programming has a priority and the interrupt request acknowledgement is waited. The interrupt process is executed after the auto-programming completes	

NOTES:

1. Do not use the address match interrupt while the command is executed because the vector of the address match interrupt is allocated on ROM.
2. Do not use the non-maskable interrupt while Block 0 is automatically erased because the fixed vector is allocated Block 0.

19.8 Noise

(1) Bypass Capacitor between Vcc and Vss Pins

Insert a bypass capacitor (at least 0.1 μ F) 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 be 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 Ω) between CNVss and Vss, and place it as close as possible to the CNVss pin.

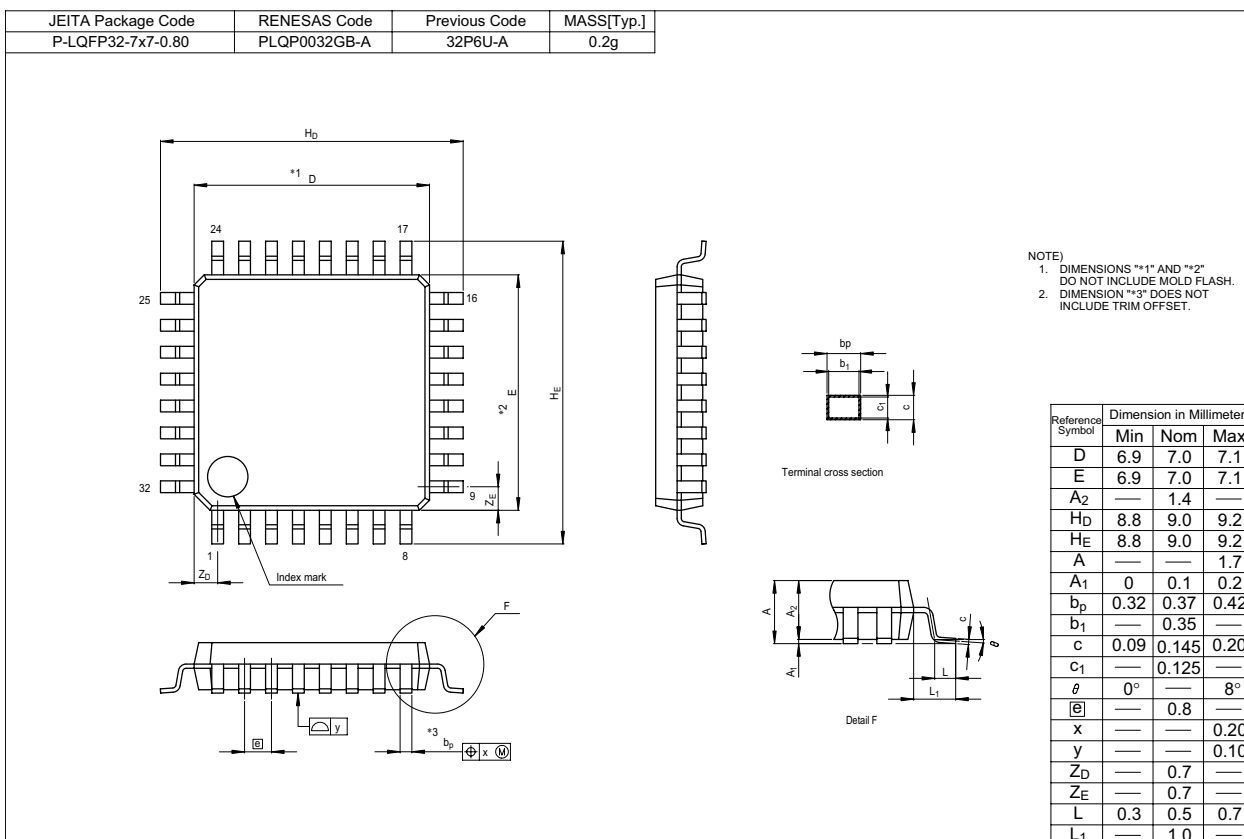
20. Usage notes for on-chip debugger

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

- (1) Do not use P0₀/AN₇/TxD₁₁ pin and P3₇/TxD₁₀/RxD₁ pin.
- (2) When write in the PD3 register (00E7₁₆ address), set bit 7 to "0".
- (3) Do not access the related serial interface 1 register.
- (4) Do not use from OC000₁₆ address to OC7FF₁₆ address because the on-chip debugger uses these addresses.
- (5) Do not set the address match interrupt (the registers of AIER, RMAD0, RMAD1 and the fixed vector tables) in a user system.
- (6) Do not use the BRK instruction in a user system.
- (7) Do not set the b5 to "0" by a user program since the on-chip debugger uses after setting the b5 in the FMR0 register to "1".
- (8) The stack pointer with upto 8 bytes is used during the user program break. Therefore, save space of 8 bytes for the stack area.

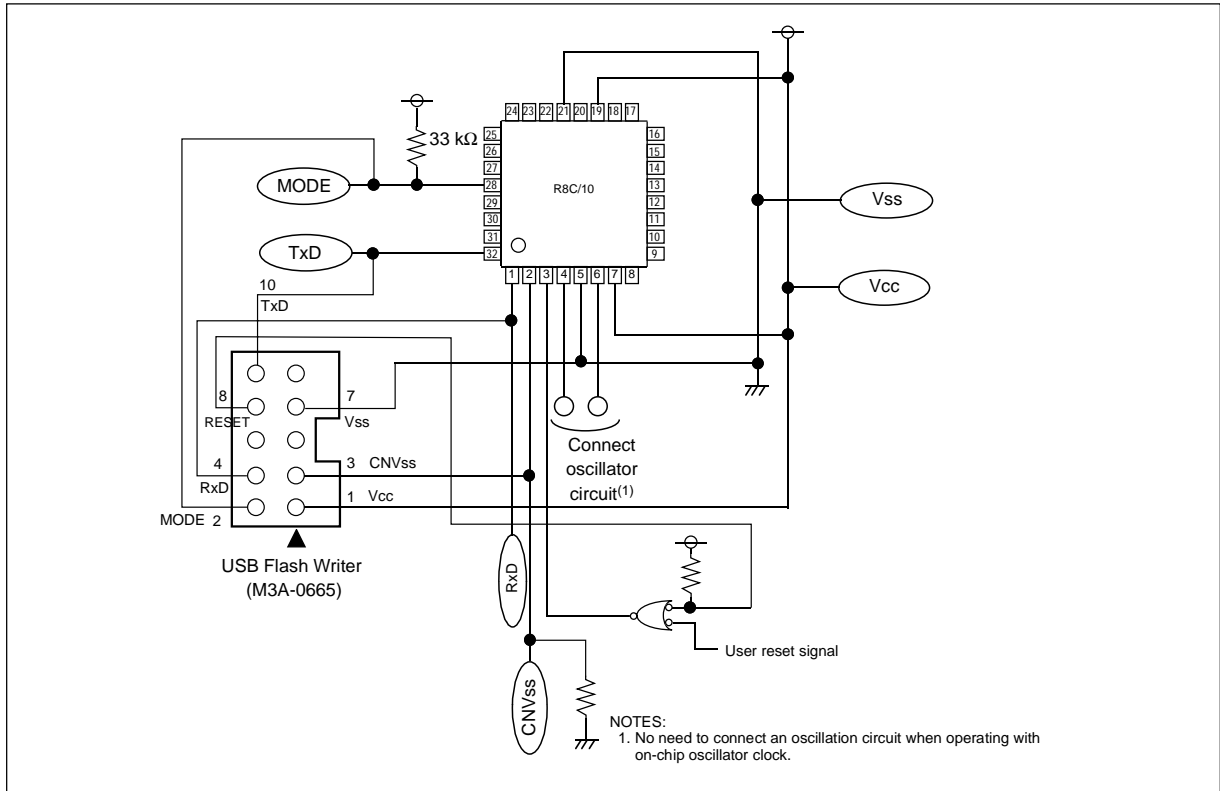
Connecting and using the on-chip debugger has some peculiar restrictions. Refer to each on-chip debugger manual for on-chip debugger details.

Appendix 1. Package Dimensions

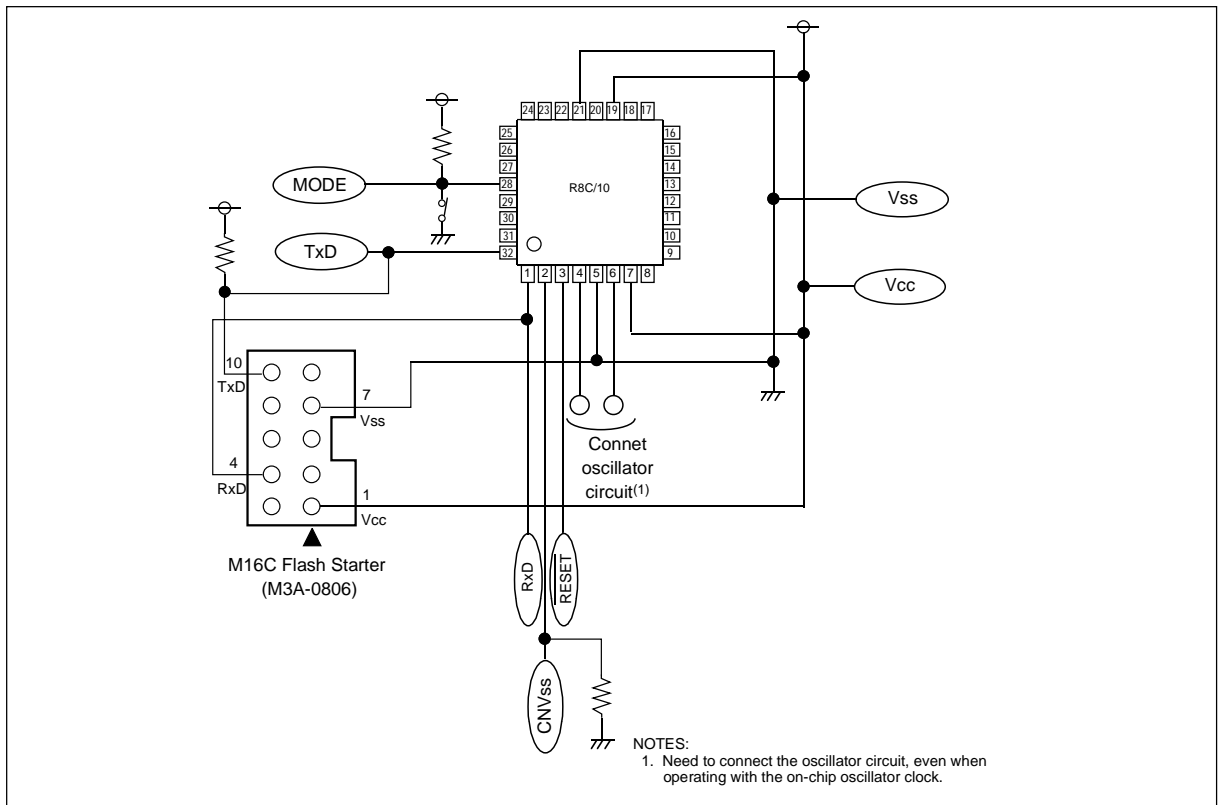


Appendix 2. Connecting examples for serial writer and on-chip debugging emulator

Appendix figure. 2.1 shows connecting examples with USB Flash Writer and appendix figure 2.2 shows connecting examples with M16C Flash Starter.



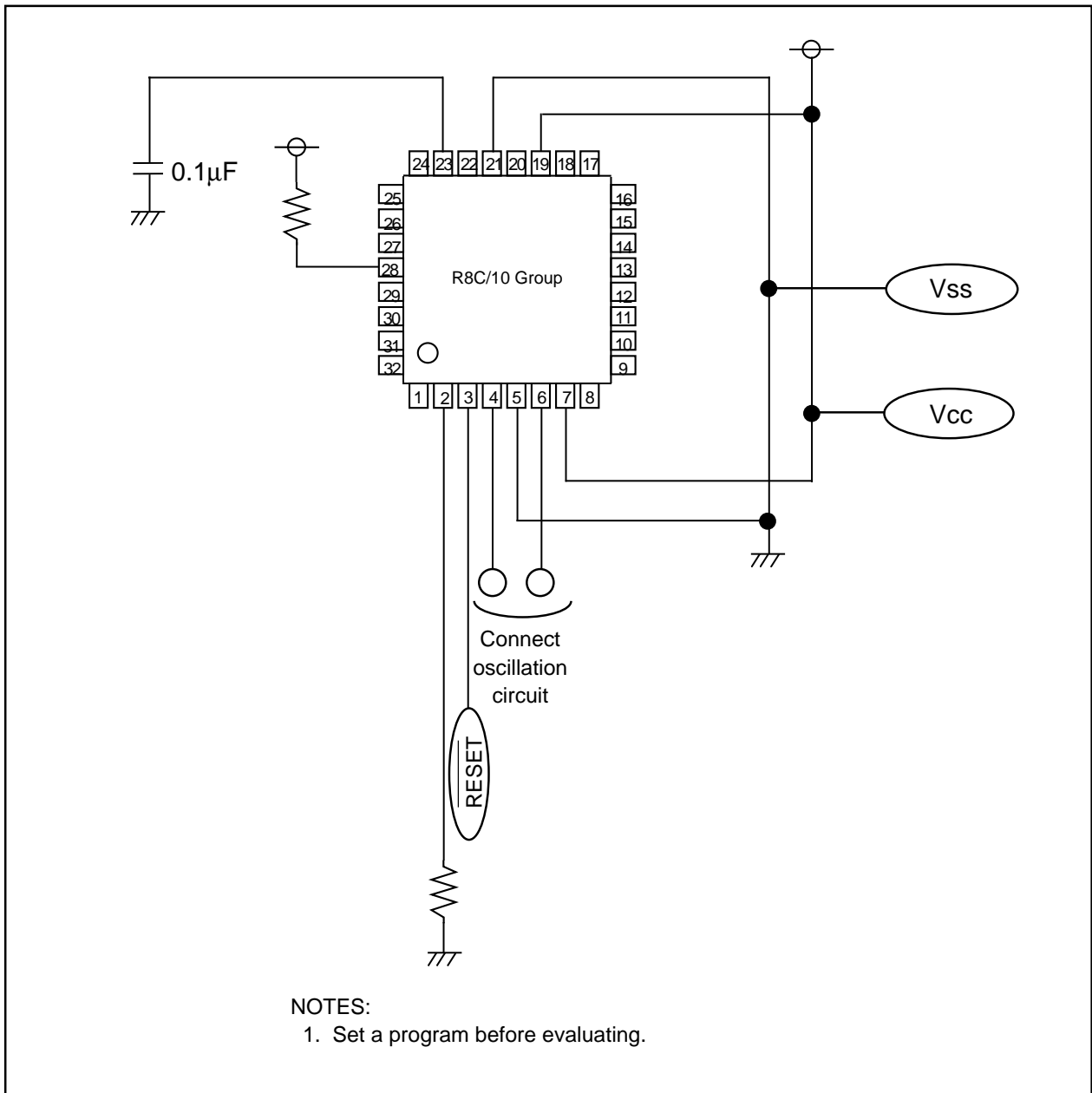
Appendix figure 2.1 Connecting examples with USB Flash Writer (M3A-0665)



Appendix figure 2.2 Connecting examples with M16C Flash Starter (M3A-0806)

Appendix 3. Example of Oscillation Evaluation Circuit

Appendix Figure 3.1 shows the Example of Oscillation Evaluation Circuit.



Appendix figure 3.1 Example of Oscillation Evaluation Circuit

Register Index

A

AD 107
 ADCON0 106, 108, 109
 ADCON1 106, 108, 109
 ADCON2 107
 ADIC 39
 AIER 52

C

CM0 19
 CM1 19

D

DRR 121

F

FMR0 146
 FMR1 147
 FMR4 147

I

INT0F 46
 INT0IC 39
 INT1IC 39
 INT2IC 39
 INT3IC 39
 INTEN 46

K

KIEN 50
 KUPIC 39

O

OCD 20

P

P0 120
 P1 120
 P3 120
 P4 120
 PD0 120
 PD1 120
 PD3 120

PD4 120
 PM0 31
 PM1 31
 PRCR 30
 PREX 57
 PREY 66
 PREZ 74
 PUM 67, 69, 71, 75, 77, 79, 81, 84
 PUR0 121
 PUR1 121

R

RMAD0 52
 RMAD1 52

S

S0RIC 39
 S0TIC 39
 S1RIC 39
 S1TIC 39

T

TC 87
 TCC0 49, 87
 TCC1 49, 87
 TCIC 39
 TCSS 57, 67, 75
 TM0 87
 TX 57
 TXIC 39
 TXMR 48, 56, 58, 59, 60, 61, 63
 TYIC 39
 TYPR 66
 TYSC 66
 TYZMR 48, 65, 69, 71, 73, 77, 79, 81, 84
 TYZOC 66, 74
 TZIC 39
 TZPR 74

TZSC 74

U

U0BRG 91

U0C0 92

U0C1 93

U0MR 92

U0RB 91

U0TB 91

U1BRG 91

U1C0 92

U1C1 93

U1MR 92

U1RB 91

U1TB 91

UCON 93

W

WDC 54

WDTR 54

WDTS 54

REVISION HISTORY

R8C/10 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.91	Sep 8, 2003	–	First edition issued
0.92	Nov 5, 2003	2	Table1.1 Add on Power Consumption Internal : Revise 10 sources to 9 sources
		4	Table 1.2 Delete **
		6	Table1.3 CNVss and MODE : Delete (5 kΩ) CNVss : Add NOTES 1 Analog power supply input : Add one sentence Reference voltage input : Add one sentence
		14	Section 5.1 Add one sentence Section 5.2 Add one sentence Delete sentences Section 5.3 Add one sentence Delet sentecnes
		19	Figure 6.2 Revise “on-chip oscillator” to “on-chip oscillator clock” on NOTES 2, CMO
		22	Section 6.2 Revise 100 kHz to 125 kHz
		26	Section 6.4.3 is revised
		30	Table 6.5 Delete “watchdog timer”
		49	Table 10.13 Revise “Function of the TYWC bit in the TYZMR” to “Function varies depending on the operation mode”
		54	Figure 11.1 is revised
		57	Figure 12.1 Delete $\overline{\text{CLR}}$ of $\overline{\text{CLR}}$
		62	Table 12.5 Select function is revised
		65	Figure 12.10 Add NOTE 7 and revise 0D ₁₆ to 0E ₁₆ Revise some parts in Figure
		66	Figure 12.11 Delete $\overline{\text{CLR}}$ of $\overline{\text{CLR}}$
		67	Figure 12.13 TYPR, NOTES 1 : Revise PYSC register to TYSC register TYZOC, NOTES 1 : Revise TYS bit to TZS bit Add NOTE 3
74	Figure 12.18 Delete $\overline{\text{CLR}}$ of $\overline{\text{CLR}}$		
75	Figure 12.20 TYZOC, NOTE 1 : Revise TYS bit to TZS bit Add NOTE 3		
81	Table 12.11 revised		

REVISION HISTORY

R8C/10 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.92	Nov 5, 2003	84	Table 12.12 revised
		87	Figure 12.28 Add "Sampling clock"
		101	Table 13.5 Add bit "CKPOL", "Set to "0"'"
		110	Section 14.3 Add under the sixth line Add Figure 14.6
		116	Figure 15.6 NOTES 2 and 3 : Revise its content is "0" to its content is indeterminate
			Figure 15.7 NOTES 1 and 2 : Revise its content is "0" to its content is indeterminate
		119	Table 16.2 Delete NOTES 3 and 4
		120	Table 16.3 Delete Tsamp
			Revise Table 16.4
		121	Add Figure 16.1
			Add Figure 16.2
		123	Table 16.7 Revise on-chip oscillator 100 kHz to 125 kHz High-speed mode, Medium-speed mode : Revise XIN = 5 MHz to XIN = 10 MHz
		124	Add Table 16.8 to 16.12
			Table 16.10 Revise NOTES 1 and 2
		125	Revise Figure 16.2 to 16.3
		126	Table 16.13 Revise fRING-S to fRING Revise "Low-speed on-chip oscillator frequency" to "On-chip oscillator frequency"
			NOTES 1 : Revise f(BCLK) = 5 MHz to 10 MHz
127	Table 16.14 On-chip oscillator : Revise 100 kHz to 125 kHz		
	High-speed mode, Medium-speed mode : Revise XIN = 5 MHz to 10MHz		
128	Add Table 16.15 to 16.19		
	Table 16.17 Revise NOTES 1 and 2		
129	Add Figure 16.3		
130	Table 17.1 Delete Data Retention		
131	Section 17.2 Add under the eighth line		
137	Figure 17.4 FMR1, bit 6 : Delete "When read, its content is indeterminate" Revise RO to —		

REVISION HISTORY

R8C/10 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.92	Nov 5, 2003	147	Section 17.5 Add sentences
		148	Table 17.7 Revise P46/XIN and P47/XOUT
		150	Figure 17.13 Add NOTES 3 Figure 17.14 Add NOTES 2
		155	Section 19.3.2 Add (1) and (4) Add Section 19.3.3 (1) Revise 19.3.3 Timer Z to 19.3.4 Timer Z Section 19.3.3 Add (1)
		161	Section 20 Add (5) and (6) Add sentences
0.93	Feb 18, 2004	147- 150	Add 4 pages
1.00	Sep 24, 2004	all pages	Words standardized (on-chip oscillator, serial interface, A/D)
		2	Table 1.1 revised
		5	Figure 1.3, NOTES 3 added
		6	Table 1.3 revised
		9	Figure 3.1, NOTES added
		10-13	One body sentence in chapter 4 added; Titles of Table 4.1 to 4.4 added
		12	Table 4.3 revised ; Table 4.4 revised
		22	Figure 6.2 revised (CM0 and CM1)
		25	Table 6.3, Timer Z and Timer C interrupt added
		26	6.4.3 Stop Mode, in "Pin Status in Stop Mode", one sentence added
		29	One sentence in 6.5.1 moves to Chapter 19
		47	One body sentence in 10.2.1 added
		49	One body sentence in 10.2.3 added
		50	One body sentence in 10.2.4 added
		51	Figure 10.15 revised
		54	Figure 11.1 revised
		57	Line 4 in 12.1 revised
		60	Table 12.3 revised
		61	Table 12.4 revised
		62	Figure 12.7 revised
64	Table 12.6 revised; Figure 12.9 revised		
69	Table 12.7 revised		
71	Table 12.8 revised, NOTES revised		
72	Figure 12.16 revised		
74	5 line in 12.3 revised ; Figure 12.18 revised		

REVISION HISTORY

R8C/10 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Sep 24, 2004	77	Table 12.9 revised
		79	Table 12.10 revised, NOTES revised
		81	Table 12.11 revised, NOTES revised
		83	Figure 12.25 revised
		84	Table 12.12 revised, NOTES revised
		86	Figure 12.27 revised
		88	Figure 12.29 revised
		91	Figure 13.2 revised
		99	13.1.3 revised
		103	Figure 13.10 revised
		111	Figure numbers in 15.1.1, 15.1.2, 15.1.3 and 15.1.4 revised
		118	Table 15.1 revised
		119	Table 16.2 revised
		120	Table 16.3 revised
			Table 16.4 revised
		122	Table 16.6 revised
		123	Table 16.7 revised
		124	Table 16.8 revised
			Table 16.12 revised
		126	Table 16.13 revised
		127	Table 16.14 revised
		128	Table 16.15 revised
			Table 16.17 revised
			Table 16.19 revised
		134	Line 2 and 8 in 17.4.2 revised
		135	FMR46 bit revised
		136	Figure 17.3 revised
		137	Figure 17.4 revised (FMR4)
		139	Figure 17.7 revised ; Figure title revised
		140	Table 17.4 revised
142	Figure 17.9 revised		
143	Figure 17.10 revised		
145	Table 17.6 revised		
152-163	Compositions in Chapter 19 modified ;19.3 added ; 19.4.5 revised ; 19.7 revised		
164	(7) in Chapter 20 added		
168	Appendix 3 added		
169-170	Page numbers in Register Index revised		

REVISION HISTORY

R8C/10 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.10	Apr.27.2005	4	Table 1.2, Figure 1.2 package name revised
		5	Figure 1.3 package name revised
		10	Table 4.1 revised
		12	Table 4.3 revised
		14	5.1 partly revised
		15	Figure 5.2 partly revised
		17	Table 6.1 partly added
		19	Figure 6.2 partly revised
		21	6.1 partly revised
		23	6.3.1 partly deleted
		24	6.4.1 partly revised
		27	Figure 6.5 revised
		28	Figure 6.6 deleted
		54	Figure 11.2 partly revised
		63	Figure 12.9 partly revised
		80	Table 12.11 partly revised
		83	Table 12.12 partly revised
		87	Figure 12.29 partly revised
		100	Table 13.6 partly revised
		103	13.2.3 Bit Rate added
		110	Figure 14.6 partly revised
			14.4 added
		111	14.5 added
		112	14.6 added
		114	Figure 15.1 revised
		115	Figure 15.2 revised
		116	Figure 15.3 revised
		120-125	15.2 added
		126	Table 15.24 partly revised
			Figure 15.9 added
128	Table 16.3 partly revised		
	Table 16.4 partly added		
	Table 16.5 partly revised		
130	Table 16.6 partly revised		
134	Table 16.13 partly revised		
138	Figure 17.1 revised		
146	Figure 17.5 added		
149	•Program partly revised		
151	Figure 17.11 partly added		
157	Figure 17.13 package name revised		
159	18.1 partly revised		
164	19.3.2 added		
165	19.4.4 partly revised		
167	19.6 partly revised		
168	19.7.1 partly added		
172	20 partly revised		
173	Package Dimensions revised		

REVISION HISTORY

R8C/10 Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.20	Jan.27.2006	2	Table 1.1 Performance outline revised
		3	Figure 1.1 Block diagram partly revised
		4	1.4 Product Information, title of Table 1.2 "Product List" → "Product Informaton" revised
		6	Figure 1.2 Type No., Memory Size, and Package partly revised
		7-8	Table 1.3 Pin description revised 2 Central Processing Unit (CPU) revised
		10	Figure 2.1 CPU register revised
		11	Table 4.1 SFR Information(1) NOTES:1 revised
		12	Table 4.2 SFR Information(2) NOTES:1 revised Table 4.3 SFR Information(3); 0081 ₁₆ : "Prescaler Y" → "Prescaler Y Register" 0082 ₁₆ : "Timer Y Secondary" → "Timer Y Secondary Register" 0083 ₁₆ : "Timer Y Primary" → "Timer Y Primary Register" 0085 ₁₆ : "Prescaler Z" → "Prescaler Z Register" 0086 ₁₆ : "Timer Z Secondary" → "Timer Z Secondary Register" 0087 ₁₆ : "Timer Z Primary" → "Timer Z Primary Register" 008C ₁₆ : "Prescaler X" → "Prescaler X Register" revised NOTES:1 revised
		13	Table 4.4 SFR Information(4) NOTES:1 revised
		15	Figure 5.2 Reset Sequence; "72cycles" → "64cycles" revised
		17	6 Clock Generation Circuit; "(oscillation stop detect function)" → "(oscillation stop detection function)" revised
		20	Table 6.1 Clock Generation Circuit Specifications NOTES: 2 deleted
		22	Figure 6.3 OCD Register NOTES: 3 partly deleted 6.2.1 On-Chip Oscillator Clock; "The application products ... to accommodate the frequency range." → "The application products ... for the frequency change." revised
		24	Table 6.2 Setting Clock Related Bit and Modes CM13 added
		28	6.5.1 How to Use Oscillation Stop Detection Function: "This function cannot ... is below 2 MHz." added
		32	Table 9.1 Bus Cycles for Access Space, Table 9.2 Access Unit and Bus Operation; "SFR" → "SFR, Data flash", ROM/RAM" → "Program ROM/RAM" revised
		37	Table 10.2 Relocatable Vector Tables; "A/D" → "A/D Conversion" revised
		45	Figure 10.9 Interrupts Priority Select Circuit NOTES: 1 deleted
		56	Figure 12.1 Timer X Block Diagram; "Peripheral data bus" → "Data bus" revised
		59	Table 12.3 Pulse Output Mode Specifications NOTES: 1 added
73	Figure 12.18 Timer Z Block Diagram; "Peripheral data bus" → "Data bus" revised		
91	Figure 13.3 U0TB to U1TB Registers, U0RB and U1RB Registers, and U0BRG and U1BRG Registers; UARTi transmit buffer register (i=0, 1) revised UARTi bit rate register (i=0, 1); NOTES: 3 added		
92	Figure 13.4 U0MR to U1MR Registers and U0C0 and U1C0 Registers; UARTi transmit/receive control register 0 (i=0, 1); NOTES: 1 added		
93	Figure 13.5 U0C1 and U1C1 Registers and UCON Register; UART transmit/receive control register 2; NOTES: 2 added		
100	Table 13.5 Registers to Be Used and Settings in UART Mode; UiBRG: "-" → "0 to 7" revised		
105	Figure 14.1 A/D Converter Block Diagram "Vref" → "Vcom" revised		

REVISION HISTORY

R8C/10 Group Hardware Manual

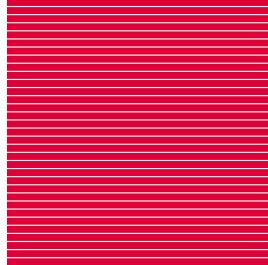
Rev.	Date	Description	
		Page	Summary
1.20	Jan.27.2006	113	14.7 Output Impedance of Sensor under A/D Conversion added
		116	Figure 15.1 Programmable I/O Ports (1); NOTES: 1 added
		117	Figure 15.2 Programmable I/O Ports (2); NOTES: 1 added
		118	Figure 15.3 Programmable I/O Ports (3); NOTES: 1 added
		119	Figure 15.4 Programmable I/O Ports (4); NOTES: 3 added
		127	Table 15.20 Port P3 ₃ /INT ₃ /TC _{IN} Setting; Bit: "PD3_1" → "PD3_3" Table 15.22 Port P4 ₅ /INT ₀ Setting; Bit: "PD3_3" → "PD4_5" Table 15.23 Port X _{IN} /P4 ₆ , X _{OUT} /P4 ₇ Setting; Setting value: External input to X _{IN} pin, "H" output from X _{OUT} pin; CM1: "1" → "0", CM0: "0" → "1", Feedback resistance: "OFF" → "ON"
		129	Table 16.2 Recommended Operating Conditions; NOTES: 1, 2, 3 revised
		130	Table 16.3 A/D Conversion Characteristics; "A/D operation clock frequency" → "A/D operating clock frequency" revised NOTES: 1, 2, 3, 4 revised
			Table 16.4 Flash Memory (Program ROM) Electrical Characteristics; "Data retention duration" → "Data hold time" revised "Topr" → "Ambient temperature" revised NOTES: 2 added Measuring condition of byte program time and block erase time deleted
		132	Table 16.6 Electrical Characteristics (1) [V _{CC} =5V]; "P1 ₀ to P1 ₇ Except X _{OUT} " → "Except P1 ₀ to P1 ₇ , X _{OUT} " revised
		133	Table 16.7 Electrical Characteristics (2) [V _{CC} =5V] NOTES: 1, 2 revised Measuring condition: Stop mode "Topr = 25 °C" added
		136	Table 16.13 Electrical Characteristics (3) [V _{CC} =3V] "P1 ₀ to P1 ₇ Except X _{OUT} " → "Except P1 ₀ to P1 ₇ , X _{OUT} " revised
		137	Table 16.14 Electrical Characteristics (4) [V _{CC} =3V] NOTES: 1, 2 revised Measuring condition: Stop mode "Topr = 25 °C" added
		147	Figure 17.4 FMR1 and FMR4; Flash memory control register 4 NOTES: 2 "Other than this period, this bit is set to "0"." revised
		153	Figure 17.11 Block Erase Command (When Using Erase-suspend Function); "Write 'D0 ₁₆ ' to the uppermost block address" → "Write 'D0 ₁₆ ' to the any block address" revised
		156	Figure 17.12 Full Status Check and Handling Procedure for Each Error revised
		158	Table 17.7 Pin Functions (Flash Memory Standard Serial I/O Mode); RESET: revised
162	19.1.1 Stop Mode "Use the next program to enter stop mode." added "• Example of entering stop mode" → "• Program of entering stop mode" "Program Example" deleted		
166	19.3.1 Oscillation Stop Detection Function "Since the oscillation stop ... is 2 MHz or below, ..." → "Since the oscillation stop ... is below 2 MHz, ..." revised		
176	Appendix figure 2.2 Connecting examples with M16C Flash Starter (M3A-0806); Pulled up added NOTES: 1 revised		

**RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER
HARDWARE MANUAL
R8C/10 Group**

**Publication Data : Rev.0.93 Feb 18, 2004
Rev.1.20 Jan 27, 2006**

**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