REJ09B0179-0122

Everywhere you imagine. RENESAS

16

M16C/30P Group

Hardware Manual

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER
M16C FAMILY / M16C/30 SERIES

Hardware Manual

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

Rev.1.22
Revision Date:Mar 29, 2007

RenesasTechnology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.
The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the M16C/30P Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

Document Type	Description	Document Title	Document No.
Datasheet	Hardware overview and electrical characteristics	M16C/30P Group Datasheet	REJ03B0088
Hardware manual	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	M16C/30P Group Hardware Manual	This hardware manual
Application note	Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C	Available from Renesas Technology Web site.	
Renesas technical update	Product specifications, updates on documents, etc.		

2. Notation of Numbers and Symbols

The notation conventions for register names, bit names, numbers, and symbols used in this manual are described below.

(1) Register Names, Bit Names, and Pin Names

Registers, bits, and pins are referred to in the text by symbols. The symbol is accompanied by the word “register,” “bit,” or “pin” to distinguish the three categories.

Examples the PM03 bit in the PM0 register
P3_5 pin, VCC pin

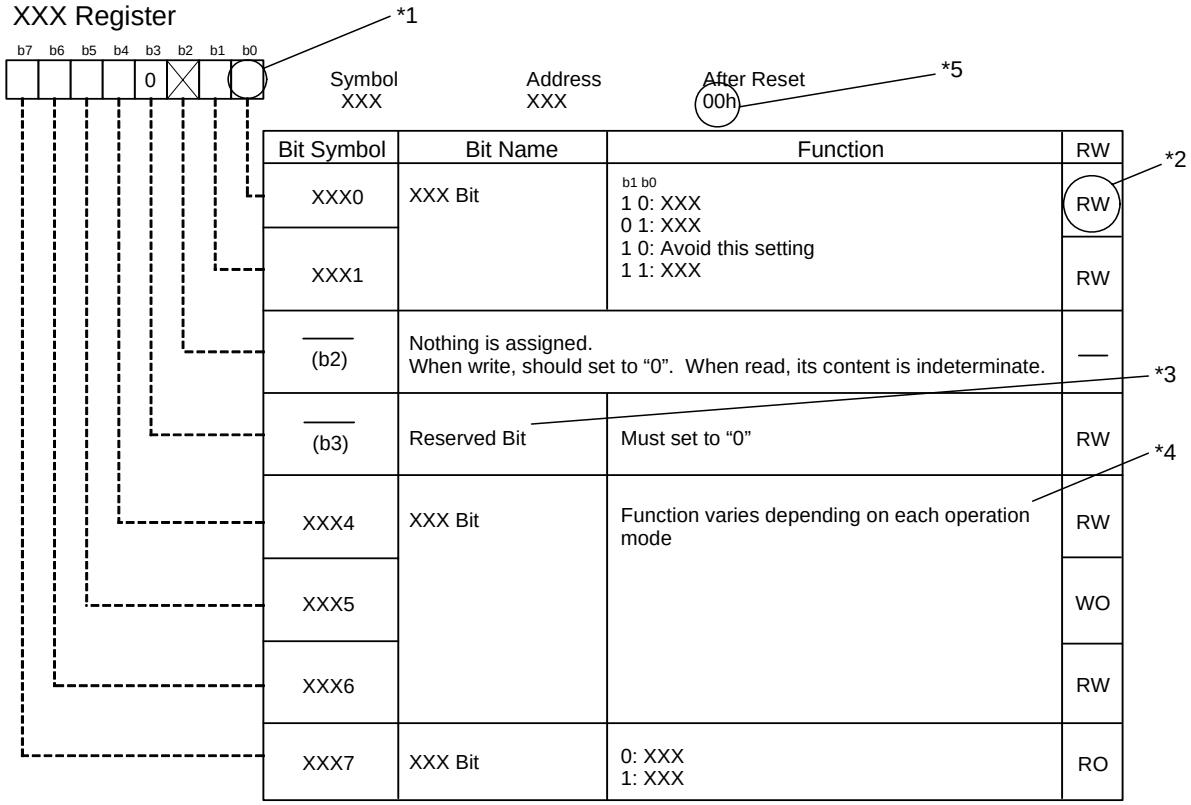
(2) Notation of Numbers

The indication “b” is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication “h” is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11b
Hexadecimal: EFA0h
Decimal: 1234

3. Register Notation

The symbols and terms used in register diagrams are described 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. As the bit may be used for future functions, if necessary, set to 0.
- Do not set to a value
  Operation is not guaranteed when a value is set.
- Function varies according to the operating mode.
  The function of the bit varies with the peripheral function mode. Refer to the register diagram for information on the individual modes.

4. List of Abbreviations and Acronyms

Abbreviation	Full Form
ACIA	Asynchronous Communication Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment bus
I/O	Input/Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connection
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SFR	Special Function Registers
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

Table of Contents

SFR Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	30
0005h	Processor Mode Register 1	PM1	31
0006h	System Clock Control Register 0	CM0	47
0007h	System Clock Control Register 1	CM1	48
0008h	Chip Select Control Register	CSR	35
0009h	Address Match Interrupt Enable Register	AIER	78
000Ah	Protect Register	PRCR	60
000Bh			
000Ch			
000Dh			
000Eh	Watchdog Timer Start Register	WDTS	80
000Fh	Watchdog Timer Control Register	WDC	28, 80
0010h 0011h 0012h	Address Match Interrupt Register 0	RMAD0	78
0013h			
0014h 0015h 0016h	Address Match Interrupt Register 1	RMAD1	78
0017h			
0018h			
0019h			
001Ah			
001Bh			
001Ch			
001Dh			
001Eh			
001Fh			
0020h 0021h 0022h	DMA0 Source Pointer	SAR0	86
0023h			
0024h 0025h 0026h	DMA0 Destination Pointer	DAR0	86
0027h			
0028h 0029h	DMA0 Transfer Counter	TCR0	86
002Ah			
002Bh			
002Ch	DMA0 Control Register	DM0CON	85
002Dh			
002Eh			
002Fh			
0030h 0031h 0032h	DMA1 Source Pointer	SAR1	86
0033h			
0034h 0035h 0036h	DMA1 Destination Pointer	DAR1	86
0037h			
0038h 0039h	DMA1 Transfer Counter	TCR1	86
003Ah			
003Bh			
003Ch	DMA1 Control Register	DM1CON	85
003Dh			
003Eh			
003Fh			

Address	Register	Symbol	Page
0040h			
0041h			
0042h			
0043h			
0044h	INT3 Interrupt Control Register	INT3IC	67
0045h			
0046h	UART1 BUS Collision Detection Interrupt Control Register	U1BCNIC	66
0047h	UART0 BUS Collision Detection Interrupt Control Register	U0BCNIC	66
0048h			
0049h	INT4 Interrupt Control Register	INT4IC	67
004Ah	UART2 Bus Collision Detection Interrupt Control Register	BCNIC	66
004Bh	DMA0 Interrupt Control Register	DM0IC	66
004Ch	DMA1 Interrupt Control Register	DM1IC	66
004Dh	Key Input Interrupt Control Register	KUPIC	66
004Eh	A/D Conversion Interrupt Control Register	ADIC	66
004Fh	UART2 Transmit Interrupt Control Register	S2TIC	66
0050h	UART2 Receive Interrupt Control Register	S2RIC	66
0051h	UART0 Transmit Interrupt Control Register	S0TIC	66
0052h	UART0 Receive Interrupt Control Register	S0RIC	66
0053h	UART1 Transmit Interrupt Control Register	S1TIC	66
0054h	UART1 Receive Interrupt Control Register	S1RIC	66
0055h	Timer A0 Interrupt Control Register	TA0IC	66
0056h	Timer A1 Interrupt Control Register	TA1IC	66
0057h	Timer A2 Interrupt Control Register	TA2IC	66
0058h			
0059h			
005Ah	Timer B0 Interrupt Control Register	TB0IC	66
005Bh	Timer B1 Interrupt Control Register	TB1IC	66
005Ch	Timer B2 Interrupt Control Register	TB2IC	66
005Dh	INT0 Interrupt Control Register	INT0IC	67
005Eh	INT1 Interrupt Control Register	INT1IC	67
005Fh	INT2 Interrupt Control Register	INT2IC	67
0060h to 01B3h			
01B4h			
01B5h	Flash Memory Control Register 1(2)	FMR1	204
01B6h			
01B7h	Flash Memory Control Register 0(2)	FMR0	203
01B8h to 0253h			
0254h			
0255h			
0256h			
0257h			
0258h			
0259h			
025Ah			
025Bh			
025Ch			
025Dh			
025Eh	Peripheral Clock Select Register	PCLKR	48
025Fh			
0260h to 0335h			
0336h			
0337h			
0338h			
0339h			
033Ah			
033Bh			
033Ch			
033Dh			
033Eh			
033Fh			

NOTES:
1. Blank columns are all reserved space. No access is allowed.
2. This register is included in the flash memory version.
3. This register is included in the One time flash version.

Address	Register	Symbol	Page
0340h			
0341h			
0342h			
0343h			
0344h			
0345h			
0346h			
0347h			
0348h			
0349h			
034Ah			
034Bh			
034Ch			
034Dh			
034Eh			
034Fh			
0350h			
0351h			
0352h			
0353h			
0354h			
0355h			
0356h			
0357h			
0358h			
0359h			
035Ah			
035Bh			
035Ch			
035Dh			
035Eh	Interrupt Factor Select Register 2	IFSR2A	75
035Fh	Interrupt Factor Select Register	IFSR	75
0360h			
0361h			
0362h			
0363h			
0364h			
0365h			
0366h			
0367h			
0368h			
0369h			
036Ah			
036Bh			
036Ch	UART0 Special Mode Register 4	U0SMR4	129
036Dh	UART0 Special Mode Register 3	U0SMR3	128
036Eh	UART0 Special Mode Register 2	U0SMR2	128
036Fh	UART0 Special Mode Register	U0SMR	127
0370h	UART1 Special Mode Register 4	U1SMR4	129
0371h	UART1 Special Mode Register 3	U1SMR3	128
0372h	UART1 Special Mode Register 2	U1SMR2	128
0373h	UART1 Special Mode Register	U1SMR	127
0374h	UART2 Special Mode Register 4	U2SMR4	129
0375h	UART2 Special Mode Register 3	U2SMR3	128
0376h	UART2 Special Mode Register 2	U2SMR2	128
0377h	UART2 Special Mode Register	U2SMR	127
0378h	UART2 Transmit/Receive Mode Register	U2MR	124
0379h	UART2 Bit Rate Generator	U2BRG	124
037Ah 037Bh	UART2 Transmit Buffer Register	U2TB	123
037Ch	UART2 Transmit/Receive Control Register 0	U2C0	125
037Dh	UART2 Transmit/Receive Control Register 1	U2C1	126
037Eh 037Fh	UART2 Receive Buffer Register	U2RB	123

Address	Register	Symbol	Page
0380h	Count Start Flag	TABSR	96, 112
0381h	Clock Prescaler Reset Fag	CPSRF	98, 112
0382h	One-Shot Start Flag	ONSF	97
0383h	Trigger Select Register	TRGSR	97
0384h	Up-Down Flag	UDF	96
0385h			
0386h 0387h	Timer A0 Register	TA0	95
0388h 0389h	Timer A1 Register	TA1	95
038Ah 038Bh	Timer A2 Register	TA2	95
038Ch			
038Dh			
038Eh			
038Fh			
0390h 0391h	Timer B0 Register	TB0	111
0392h 0393h	Timer B1 Register	TB1	111
0394h 0395h	Timer B2 Register	TB2	111
0396h	Timer A0 Mode Register	TA0MR	95
0397h	Timer A1 Mode Register	TA1MR	95
0398h	Timer A2 Mode Register	TA2MR	95
0399h			
039Ah			
039Bh	Timer B0 Mode Register	TB0MR	111
039Ch	Timer B1 Mode Register	TB1MR	111
039Dh	Timer B2 Mode Register	TB2MR	111
039Eh			
039Fh			
03A0h	UART0 Transmit/Receive Mode Register	U0MR	124
03A1h	UART0 Bit Rate Generator	U0BRG	124
03A2h 03A3h	UART0 Transmit Buffer Register	U0TB	123
03A4h	UART0 Transmit/Receive Control Register 0	U0C0	125
03A5h	UART0 Transmit/Receive Control Register 1	U0C1	126
03A6h 03A7h	UART0 Receive Buffer Register	U0RB	123
03A8h	UART1 Transmit/Receive Mode Register	U1MR	124
03A9h	UART1 Bit Rate Generator	U1BRG	124
03AAh 03ABh	UART1 Transmit Buffer Register	U1TB	123
03ACh	UART1 Transmit/Receive Control Register 0	U1C0	125
03ADh	UART1 Transmit/Receive Control Register 1	U1C1	126
03AEh 03AFh	UART1 Receive Buffer Register	U1RB	123
03B0h	UART Transmit/Receive Control Register 2	UCON	127
03B1h			
03B2h			
03B3h			
03B4h			
03B5h			
03B6h			
03B7h			
03B8h	DMA0 Request Factor Select Register	DM0SL	83
03B9h			
03BAh	DMA1 Request Factor Select Register	DM1SL	84
03BBh			
03BCh 03BDh	CRC Data Register	CRCD	180
03BEh	CRC Input Register	CRCIN	180
03BFh			

NOTES:
1. Blank columns are all reserved space. No access is allowed.

Address	Register	Symbol	Page
03C0h 03C1h	A/D Register 0	AD0	171
03C2h 03C3h	A/D Register 1	AD1	171
03C4h 03C5h	A/D Register 2	AD2	171
03C6h 03C7h	A/D Register 3	AD3	171
03C8h 03C9h	A/D Register 4	AD4	171
03CAh 03CBh	A/D Register 5	AD5	171
03CCh 03CDh	A/D Register 6	AD6	171
03CEh 03CFh	A/D Register 7	AD7	171
03D0h			
03D1h			
03D2h			
03D3h			
03D4h	A/D Control Register 2	ADCON2	171
03D5h			
03D6h	A/D Control Register 0	ADCON0	170
03D7h	A/D Control Register 1	ADCON1	170
03D8h			
03D9h			
03DAh			
03DBh			
03DCh			
03DDh			
03DEh			
03DFh			
03E0h	Port P0 Register	P0	189
03E1h	Port P1 Register	P1	189
03E2h	Port P0 Direction Register	PD0	188
03E3h	Port P1 Direction Register	PD1	188
03E4h	Port P2 Register	P2	189
03E5h	Port P3 Register	P3	189
03E6h	Port P2 Direction Register	PD2	188
03E7h	Port P3 Direction Register	PD3	188
03E8h	Port P4 Register	P4	189
03E9h	Port P5 Register	P5	189
03EAh	Port P4 Direction Register	PD4	188
03EBh	Port P5 Direction Register	PD5	188
03ECh	Port P6 Register	P6	189
03EDh	Port P7 Register	P7	189
03EEh	Port P6 Direction Register	PD6	188
03EFh	Port P7 Direction Register	PD7	188
03F0h	Port P8 Register	P8	189
03F1h	Port P9 Register	P9	189
03F2h	Port P8 Direction Register	PD8	188
03F3h	Port P9 Direction Register	PD9	188
03F4h	Port P10 Register	P10	189
03F5h			
03F6h	Port P10 Direction Register	PD10	188
03F7h			
03F8h			
03F9h			
03FAh			
03FBh			
03FCh	Pull-Up Control Register 0	PUR0	190
03FDh	Pull-Up Control Register 1	PUR1	190
03FEh	Pull-Up Control Register 2	PUR2	191
03FFh	Port Control Register	PCR	191

NOTES:

1. Blank columns are all reserved space. No access is allowed.

1. Overview

The M16C/30P Group of single-chip microcomputers is built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core and is packaged in a 100-pin plastic molded QFP.
These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 1 Mbyte of address space, they are capable of executing instructions at high speed. In addition, these microcomputers contain a multiplier and DMAC which combined with fast instruction processing capability, make it suitable for control of various OA, communication, and industrial equipment which requires high-speed arithmetic/logic operations.

1.1 Applications

Audio, cameras, TV, home appliance, office/communications/portable/industrial equipment, etc.

1.2 Performance Outline

Table 1.1 lists Performance Outline of M16C/30P Group.

Table 1.1 Performance Outline of M16C/30P Group

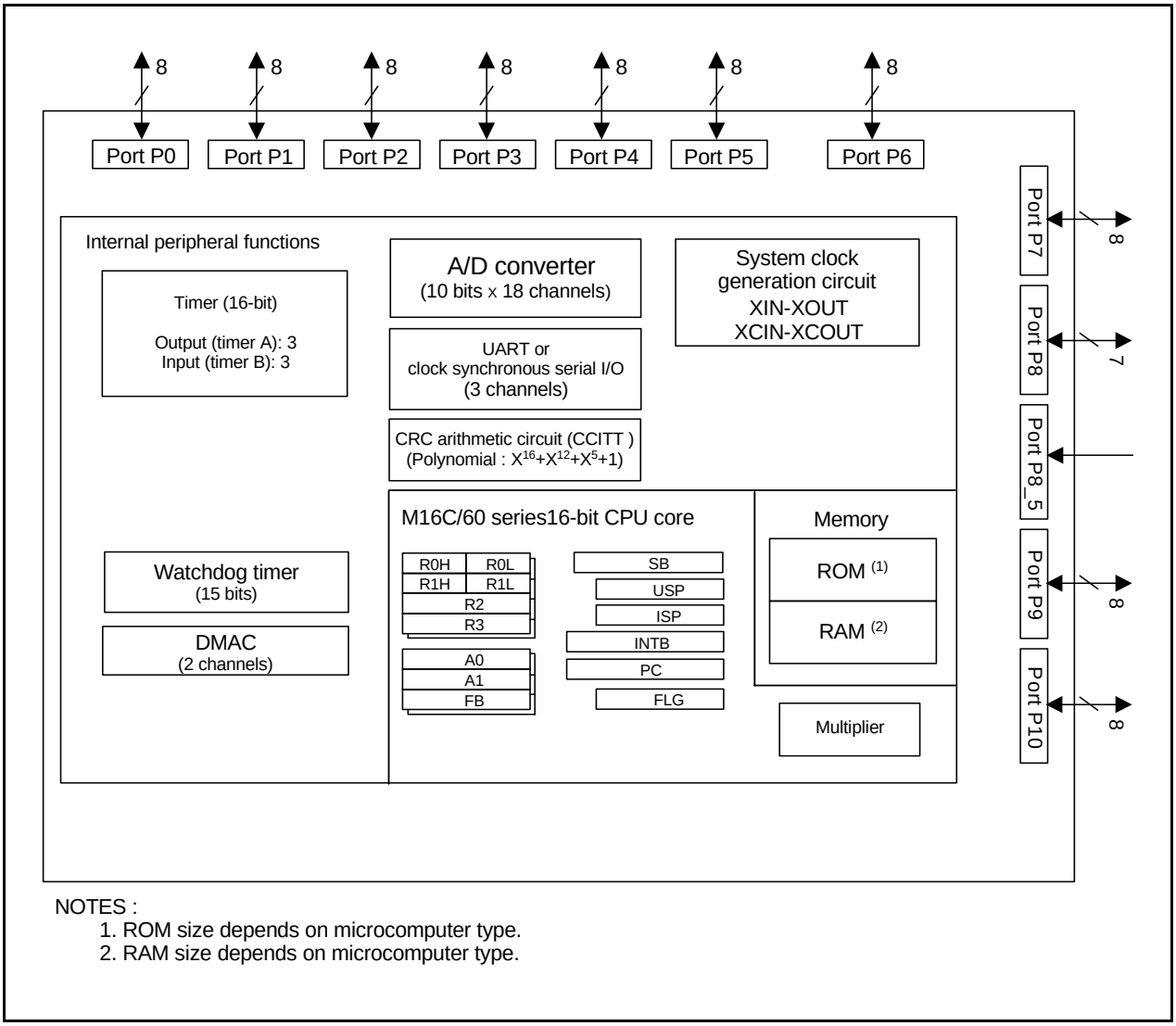
Item		Performance
CPU	Number of Basic Instructions	91 instructions
	Minimum Instruction Execution Time	62.5ns(f(XIN)=16MHz, VCC1=VCC2=3.0 to 5.5V, no wait) 100ns(f(XIN)=10MHz, VCC1=VCC2=2.7 to 5.5V, no wait)
	Operation Mode	Single-chip, memory expansion and microprocessor mode
	Memory Space	1 Mbyte
	Memory Capacity	See **Table 1.2 Product List**
Peripheral Function	Port	Input/Output : 87 pins, Input : 1 pin
	Multifunction Timer	Timer A : 16 bits x 3 channels, Timer B : 16 bits x 3 channels
	Serial Interface	1 channels Clock synchronous, UART, I2CBus(1), IEBus(2) 2 channels Clock synchronous, UART, I2CBus(1)
	A/D Converter	10-bit A/D converter: 1 circuit, 18 channels
	DMAC	2 channels
	CRC Calculation Circuit	CCITT-CRC
	Watchdog Timer	15 bits x 1 channel (with prescaler)
	Interrupt	Internal: 20 sources, External: 7 sources, Software: 4 sources, Priority level: 7 levels
	Clock Generating Circuit	2 circuits Main clock generation circuit (*), Subclock generation circuit (*), (*)Equipped with a built-in feedback resistor.
Electric Characteristics	Supply Voltage	VCC1=VCC2=3.0 to 5.5 V (f(XIN)=16MHz) VCC1=VCC2=2.7 to 5.5 V (f(XIN)=10MHz, no wait)
	Power Consumption	10 mA (VCC1=VCC2=5V, f(XIN)=16MHz) 8 mA (VCC1=VCC2=3V, f(XIN)=10MHz) 1.8 μA (VCC1=VCC2=3V, f(XCIN)=32kHz, wait mode) 0.7 μA(VCC1=VCC2=3V, stop mode)
One time flash version	Program Supply Voltage	3.3±0.3 V or 5.0±0.5 V
Flash memory version	Program/Erase Supply Voltage	3.3±0.3 V or 5.0±0.5 V
	Program and Erase Endurance	100 times (all area)
Operating Ambient Temperature		-20 to 85°C, -40 to 85°C
Package		100-pin plastic mold QFP, LQFP

NOTES:

1. I2C bus is a registered trademark of Koninklijke Philips Electronics N. V.
2. IEBus is a registered trademark of NEC Electronics Corporation.
3. Use the M16C/30P on VCC1 = VCC2.

1.3 Block Diagram

Figure 1.1 is a M16C/30P Group Block Diagram.



NOTES :
1. ROM size depends on microcomputer type.
2. RAM size depends on microcomputer type.

Figure 1.1 M16C/30P Group Block Diagram

1.4 Product List

Table 1.2 lists the M16C/30P group products and Figure 1.2 shows the Part No., Memory Size, and Package. Table 1.4 lists Product Code of MASK ROM version for M16C/30P. Figure 1.3 shows the Marking Diagram of Mask ROM Version for M16C/30P (Top View). Table 1.5 lists Product Code of One Time Flash version, Flash Memory version, and ROM-less version for M16C/30P. Figure 1.4 shows the Marking Diagram of One Time Flash version, Flash Memory version, and ROM-less Version for M16C/30P (Top View). Please specify the marking for M16C30P (MASK ROM version) when placing an order for ROM.

Table 1.2 Product List (1) **As of March 2007**

Part No.	ROM Capacity	RAM Capacity	package code (1)	Remarks
M30302MAP-XXXFP	96 Kbytes	5 Kbytes	PRQP0100JB-A	Mask ROM version
M30302MAP-XXXGP			PLQP0100KB-A	
M30302MCP-XXXFP	128 Kbytes		PRQP0100JB-A	
M30302MCP-XXXGP			PLQP0100KB-A	
M30302MDP-XXXFP	160 Kbytes	6 Kbytes	PRQP0100JB-A	
M30302MDP-XXXGP			PLQP0100KB-A	
M30302MEP-XXXFP	192 Kbytes		PRQP0100JB-A	
M30302MEP-XXXGP			PLQP0100KB-A	
M30302GAPFP	96 Kbytes	5 Kbytes	PRQP0100JB-A	One Time Flash version (blank product)
M30302GAPGP (D)			PLQP0100KB-A	
M30302GCPFP	128 Kbytes		PRQP0100JB-A	
M30302GCPGP (D)			PLQP0100KB-A	
M30302GDPFP	160 Kbytes	6 Kbytes	PRQP0100JB-A	
M30302GDPGP (D)			PLQP0100KB-A	
M30304GDPFP (D)		12 Kbytes	PRQP0100JB-A	
M30304GDPGP (D)			PLQP0100KB-A	
M30302GEPFP	192 Kbytes	6 Kbytes	PRQP0100JB-A	
M30302GEPGP (D)			PLQP0100KB-A	
M30304GEPFP (D)		12 Kbytes	PRQP0100JB-A	
M30304GEPGP (D)			PLQP0100KB-A	
M30302GGPFP (D)	256 Kbytes	12 Kbytes	PRQP0100JB-A	
M30302GGPGP (D)			PLQP0100KB-A	
M30302GAP-XXXFP	96 Kbytes	5 Kbytes	PRQP0100JB-A	One Time Flash version (factory programmed product)
M30302GAPvGP (D)			PLQP0100KB-A	
M30302GCP-XXXFP	128 Kbytes		PRQP0100JB-A	
M30302GCP-XXXGP (D)			PLQP0100KB-A	
M30302GDP-XXXFP	160 Kbytes	6 Kbytes	PRQP0100JB-A	
M30302GDP-XXXGP (D)			PLQP0100KB-A	
M30304GDP-XXXFP (D)		12 Kbytes	PRQP0100JB-A	
M30304GDP-XXXGP (D)			PLQP0100KB-A	
M30302GEP-XXXFP	192 Kbytes	6 Kbytes	PRQP0100JB-A	
M30302GEP-XXXGP (D)			PLQP0100KB-A	
M30304GEP-XXXFP (D)		12 Kbytes	PRQP0100JB-A	
M30304GEP-XXXGP (D)			PLQP0100KB-A	
M30302GGP-XXXFP (D)	256 Kbytes	12 Kbytes	PRQP0100JB-A	
M30302GGP-XXXGP (D)			PLQP0100KB-A	

(D): Under development
(P): Under planning
NOTES:
1. Previous package codes are as follows.
   PRQP0100JB-A : 100P6S-A,
   PLQP0100KB-A : 100P6Q-A
2. Block A (4-Kbytes space) is available in flash memory version.

Table 1.3 Product List (2) **As of March 2007**

Part No.	ROM Capacity	RAM Capacity	package code (1)	Remarks
M30302FAPFP	96 K + 4 Kbytes	5 Kbytes	PRQP0100JB-A	Flash memory version(2)
M30302FAPGP			PLQP0100KB-A	
M30302FCPFP	128 K + 4 Kbytes		PRQP0100JB-A	
M30302FCPGP			PLQP0100KB-A	
M30302FEPFP	192 K + 4 Kbytes	6 Kbytes	PRQP0100JB-A	
M30302FEPGP			PLQP0100KB-A	
M30302SPFP	-	6 Kbytes	PRQP0100JB-A	ROM-less version
M30302SPGP			PLQP0100KB-A	

(D): Under development
(P): Under planning
NOTES:
1. Previous package codes are as follows.
   PRQP0100JB-A : 100P6S-A,
   PLQP0100KB-A : 100P6Q-A
2. Block A (4-Kbytes space) is available in flash memory version.

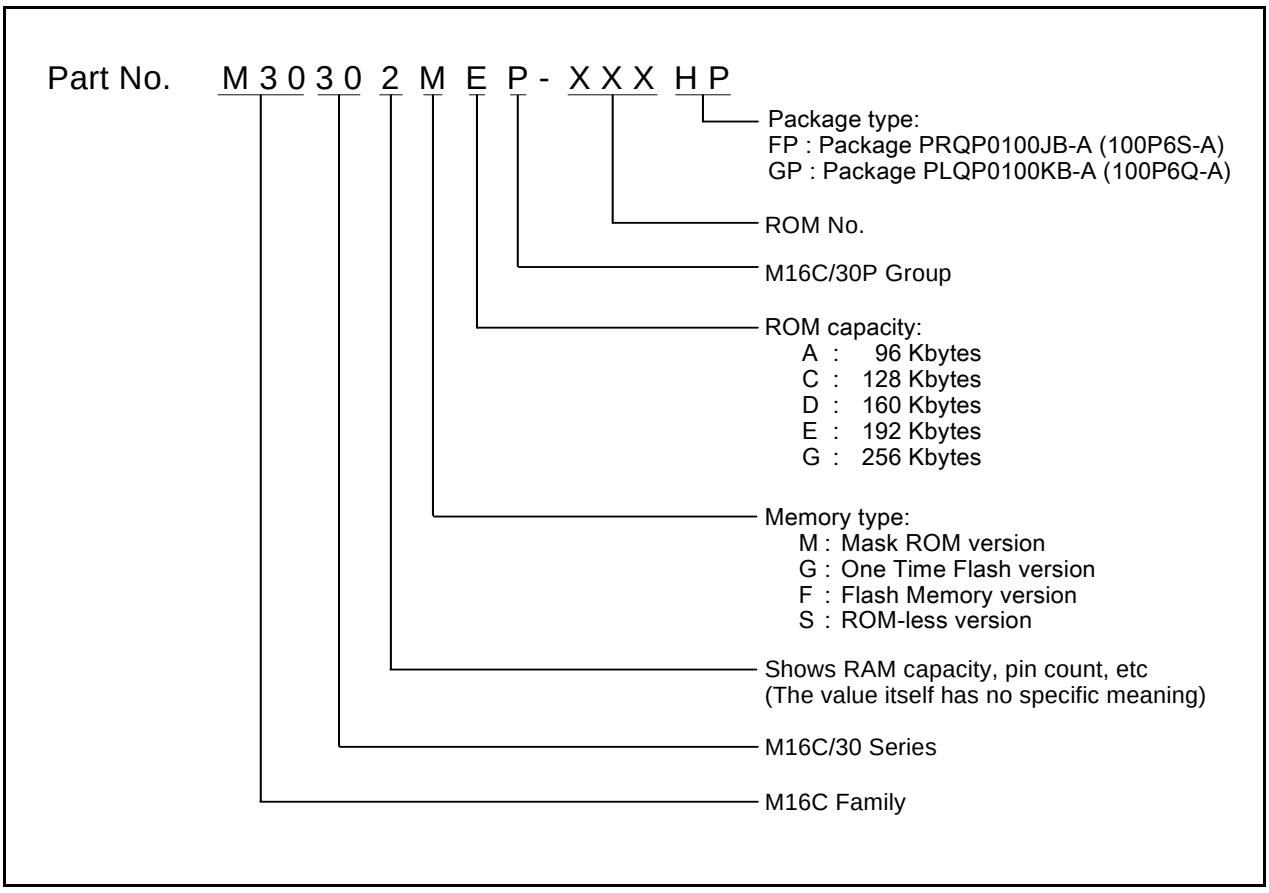


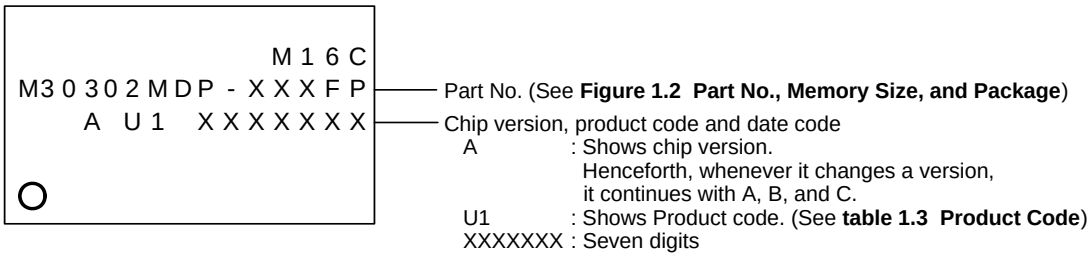
Figure 1.2 Part No., Memory Size, and Package

Table 1.4 Product Code of MASK ROM version for M16C/30P

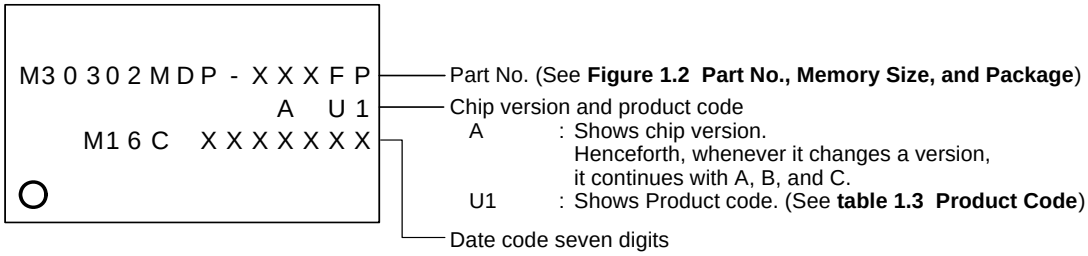
Product Code	Package	Operating Ambient Temperature
U1	Lead-free	-20°C to 85°C
U4		-40°C to 85°C

PRQP0100JB-A (100P6S-A)

1. Standard Renesas Mark

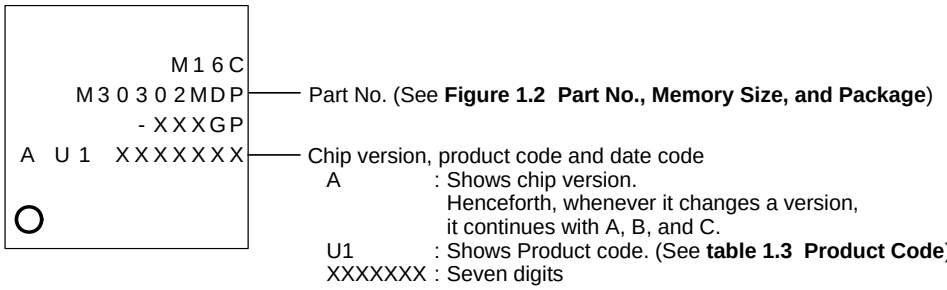


2. Customer's Parts Number + Renesas catalog name

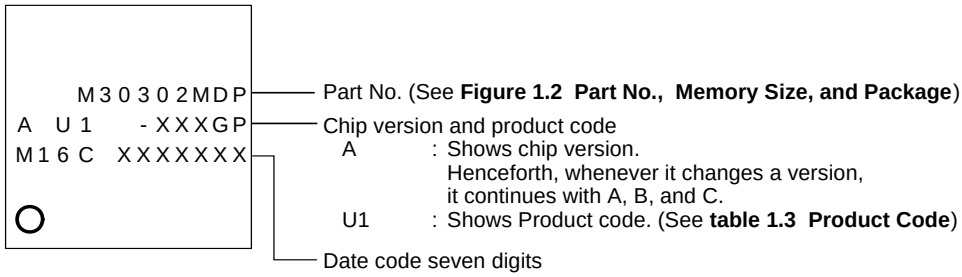


PLQP0100KB-A (100P6Q-A)

1. Standard Renesas Mark



2. Customer's Parts Number + Renesas catalog name



NOTES:
1. Refer to the **mark specification form** for details of the Mask ROM version marking.

Figure 1.3 Marking Diagram of Mask ROM Version for M16C/30P (Top View)

Table 1.5 Product Code of One Time Flash version, Flash Memory version, and ROM-less version for M16C/30P

	Product Code	Package	Internal ROM		Operating Ambient Temperature
			Program and Erase Endurance	Temperature Range	
One Time Flash version	U3	Lead-free	0	0°C to 60°C	-40°C to 85°C
	U5				-20°C to 85°C
Flash Memory version	U3	Lead-free	100	0°C to 60°C	-40°C to 85°C
	U5				-20°C to 85°C
ROM-less version	U3	Lead-free	–	–	-40°C to 85°C
	U5				-20°C to 85°C

NOTES:The one time flash version can be written once only.

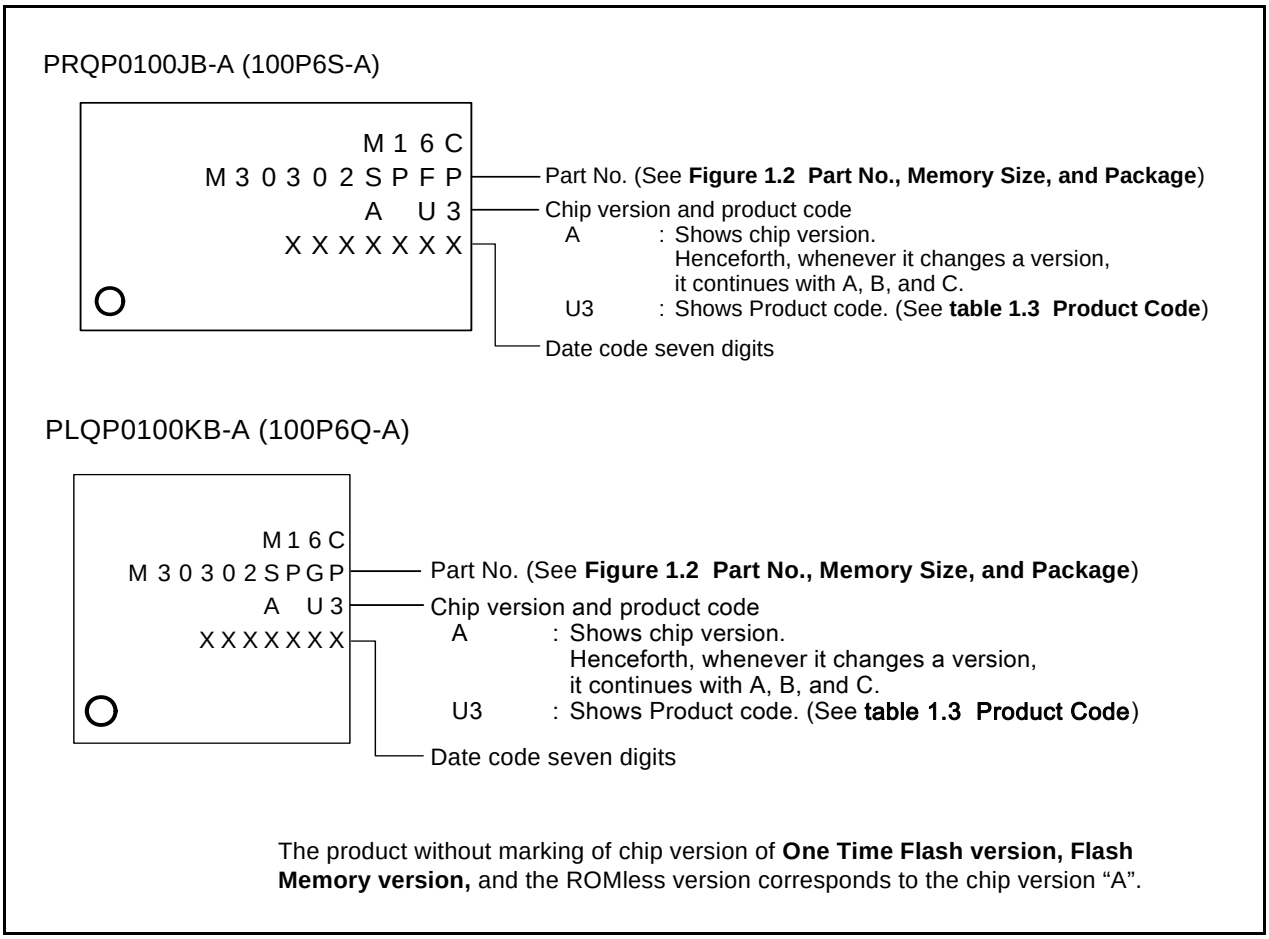


Figure 1.4 Marking Diagram of One Time Flash version, Flash Memory version, and ROM-less Version for M16C/30P (Top View)

1.5 Pin Configuration

Figures 1.5 to 1.6 show the pin configurations (top view).

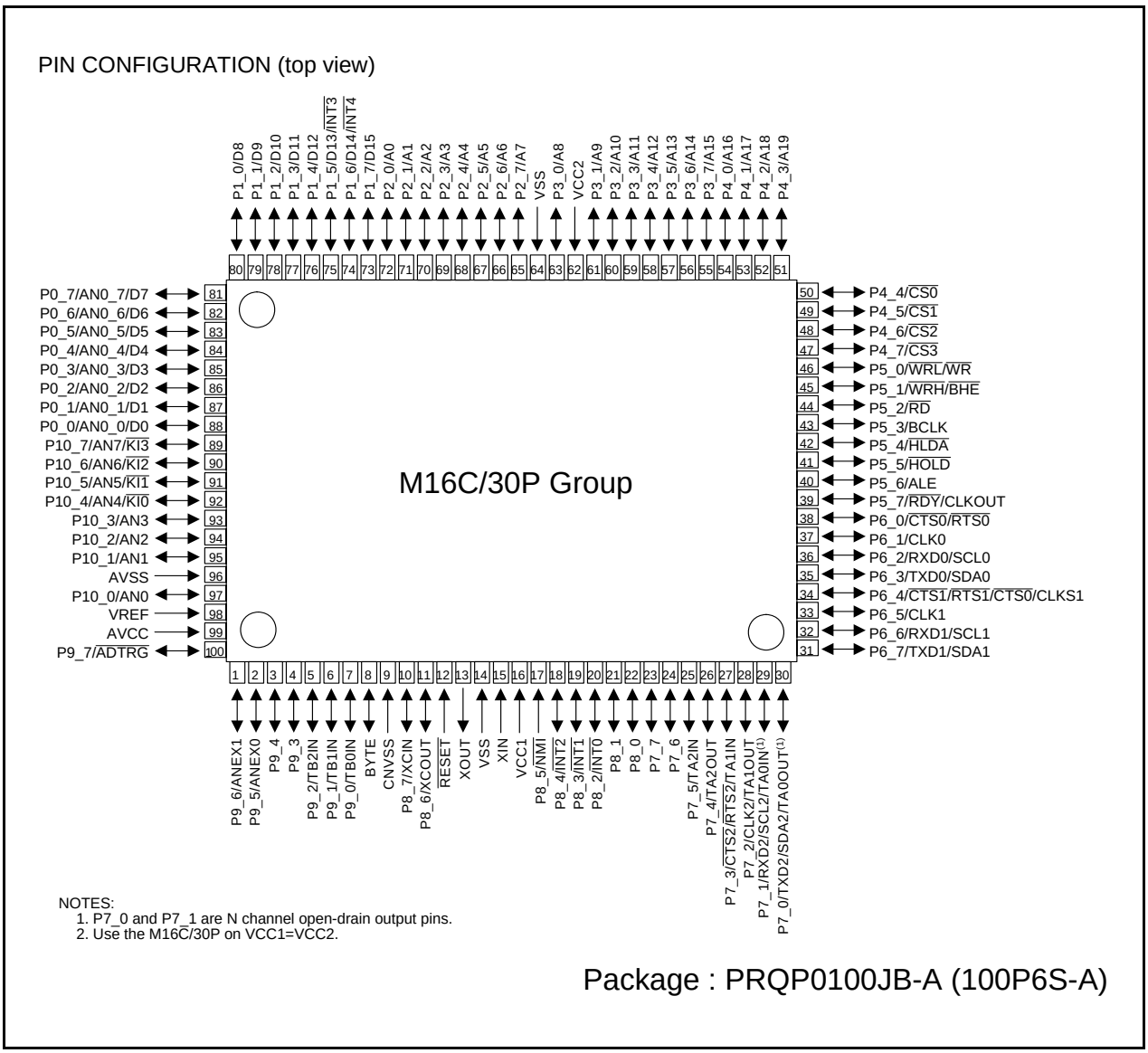
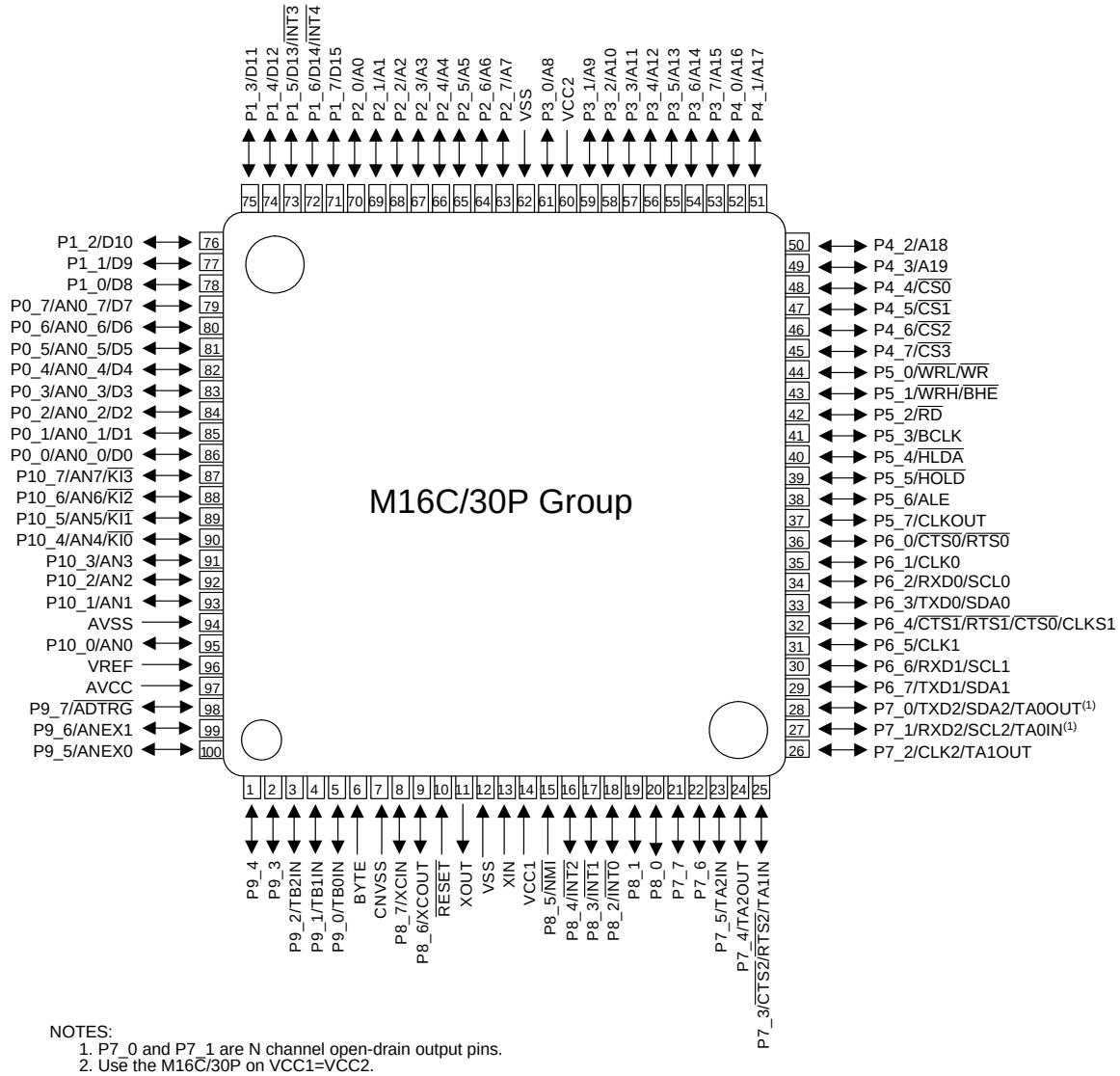


Figure 1.5 Pin Configuration (Top View)

PIN CONFIGURATION (top view)



NOTES:
1. P7_0 and P7_1 are N channel open-drain output pins.
2. Use the M16C/30P on VCC1=VCC2.

Package : PLQP0100KB-A (100P6Q-A)

Figure 1.6 Pin Configuration (Top View)

Table 1.6 Pin Characteristics (1)

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

Table 1.7 Pin Characteristics (2)

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

1.6 Pin Description

Table 1.8 Pin Description (1)

Signal Name	Pin Name	I/O Type	Description
Power supply input	VCC1, VCC2 VSS	I	Apply 2.7 to 5.5 V to the VCC1 and VCC2 pins and 0 V to the VSS pin. The VCC apply condition is that VCC1 = VCC2.
Analog power supply input	AVCC AVSS	I	Applies the power supply for the A/D converter. Connect the AVCC pin to VCC1. Connect the AVSS pin to VSS.
Reset input	$\overline{\text{RESET}}$	I	The microcomputer is in a reset state when applying “L” to the this pin.
CNVSS	CNVSS	I	Switches processor mode. Connect this pin to VSS to when after a reset to start up in single-chip mode. Connect this pin to VCC1 to start up in microprocessor mode.
External data bus width select input	BYTE	I	Switches the data bus in external memory space. The data bus is 16 bits long when the this pin is held "L" and 8 bits long when the this pin is held "H". Set it to either one. Connect this pin to VSS when an single-chip mode.
Bus control pins	D0 to D7	I/O	Inputs and outputs data (D0 to D7) when these pins are set as the separate bus.
	D8 to D15	I/O	Inputs and outputs data (D8 to D15) when external 16-bit data bus is set as the separate bus.
	A0 to A19	O	Output address bits (A0 to A19).
	$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	O	Output $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ signals. $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ are chip-select signals to specify an external space.
	$\overline{\text{WRL}}$/$\overline{\text{WR}}$ $\overline{\text{WRH}}$/$\overline{\text{BHE}}$ $\overline{\text{RD}}$	O	Output $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, ($\overline{\text{WR}}$, $\overline{\text{BHE}}$), $\overline{\text{RD}}$ signals. $\overline{\text{WRL}}$ and $\overline{\text{WRH}}$ or $\overline{\text{BHE}}$ and $\overline{\text{WR}}$ can be switched by program. • $\overline{\text{WRL}}$, $\overline{\text{WRH}}$ and $\overline{\text{RD}}$ are selected The $\overline{\text{WRL}}$ signal becomes "L" by writing data to an even address in an external memory space. The $\overline{\text{WRH}}$ signal becomes "L" by writing data to an odd address in an external memory space. The $\overline{\text{RD}}$ pin signal becomes "L" by reading data in an external memory space. • $\overline{\text{WR}}$, $\overline{\text{BHE}}$ and $\overline{\text{RD}}$ are selected The $\overline{\text{WR}}$ signal becomes "L" by writing data in an external memory space. The $\overline{\text{RD}}$ signal becomes "L" by reading data in an external memory space. The $\overline{\text{BHE}}$ signal becomes "L" by accessing an odd address. Select $\overline{\text{WR}}$, $\overline{\text{BHE}}$ and $\overline{\text{RD}}$ for an external 8-bit data bus.
	ALE	O	ALE is a signal to latch the address.
	$\overline{\text{HOLD}}$	I	While the $\overline{\text{HOLD}}$ pin is held "L", the microcomputer is placed in a hold state.
	$\overline{\text{HLDA}}$	O	In a hold state, $\overline{\text{HLDA}}$ outputs a "L" signal.
	$\overline{\text{RDY}}$	I	While applying a "L" signal to the $\overline{\text{RDY}}$ pin, the microcomputer is placed in a wait state.

I : Input O : Output I/O : Input and output

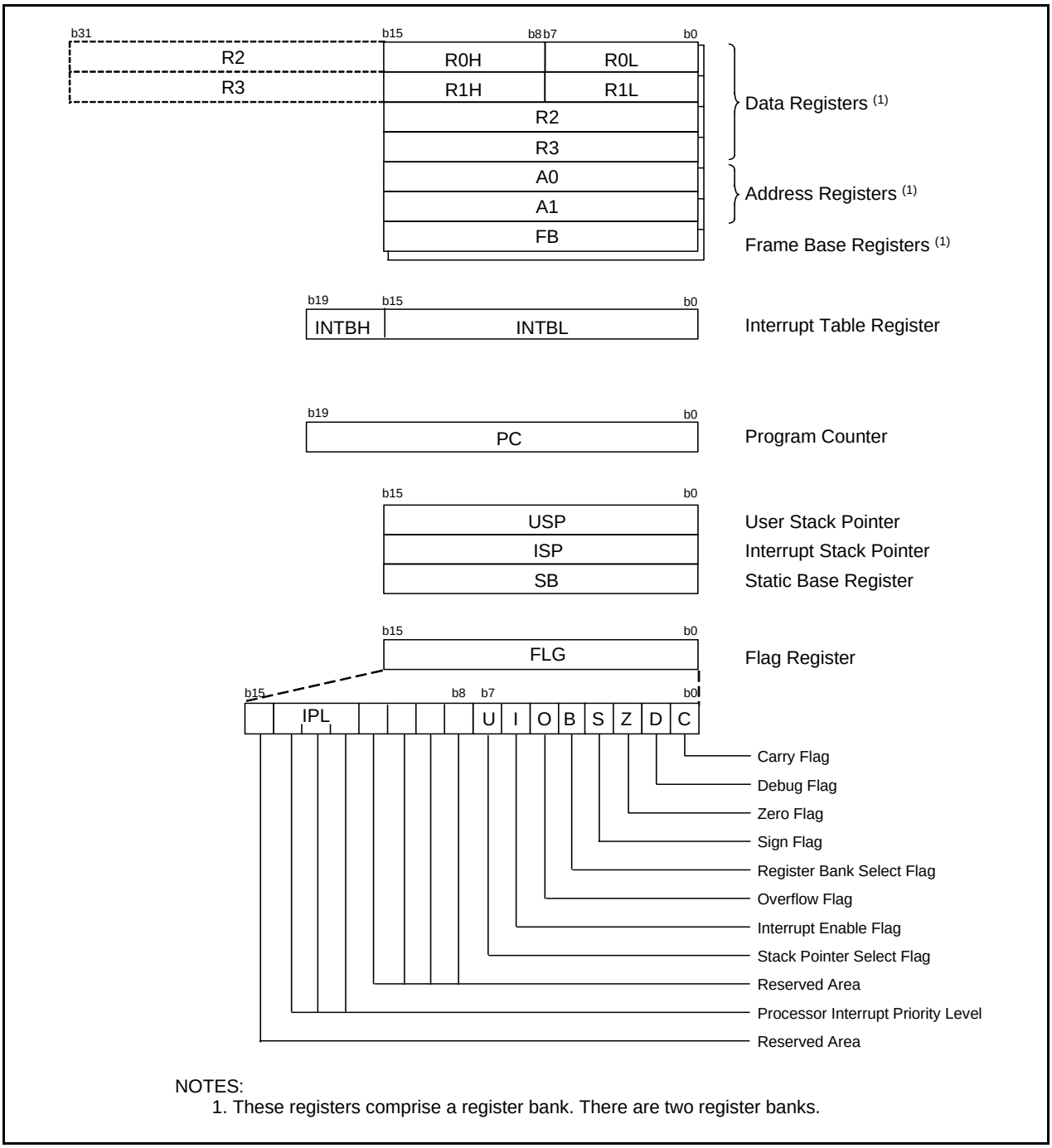
Table 1.9 Pin Description (2)

Signal Name	Pin Name	I/O Type	Description
Main clock input	XIN	I	I/O pins for the main clock generation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT. To use the external clock, input the clock from XIN and leave XOUT open.
Main clock output	XOUT	O	
Sub clock input	XCIN	I	I/O pins for a sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOUT. To use the external clock, input the clock from XCIN and leave XCOUT open.
Sub clock output	XCOUT	O	
Clock output	CLKOUT	O	The clock of the same cycle as fC, f8, or f32 is outputted.
$\overline{\text{INT}}$ interrupt input	$\overline{\text{INT0}}$ to $\overline{\text{INT4}}$	I	Input pins for the $\overline{\text{INT}}$ interrupt.
$\overline{\text{NMI}}$ interrupt input	$\overline{\text{NMI}}$	I	Input pin for the $\overline{\text{NMI}}$ interrupt.
Key input interrupt input	$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$	I	Input pins for the key input interrupt.
Timer A	TA0OUT to TA2OUT	I/O	These are timer A0 to timer A2 I/O pins. (however, the output of TA0OUT for the N-channel open drain output.)
	TA0IN to TA2IN	I	These are timer A0 to timer A2 input pins.
Timer B	TB0IN to TB2IN	I	These are timer B0 to timer B2 input pins.
Serial interface	$\overline{\text{CTS0}}$ to $\overline{\text{CTS2}}$	I	These are send control input pins.
	$\overline{\text{RTS0}}$ to $\overline{\text{RTS2}}$	O	These are receive control output pins.
	CLK0 to CLK2	I/O	These are transfer clock I/O pins.
	RXD0 to RXD2	I	These are serial data input pins.
	TXD0 to TXD2	O	These are serial data output pins. (however, TXD2 for the N-channel open drain output.)
	CLKS1	O	This is output pin for transfer clock output from multiple pins function.
I^2C mode	SDA0 to SDA2	I/O	These are serial data I/O pins. (however, SDA2 for the N-channel open drain output.)
	SCL0 to SCL2	I/O	These are transfer clock I/O pins. (however, SCL2 for the N-channel open drain output.)
Reference voltage input	VREF	I	Applies the reference voltage for the A/D converter.
A/D converter	AN0 to AN7, AN0_0 to AN0_7	I	Analog input pins for the A/D converter.
	$\overline{\text{ADTRG}}$	I	This is an A/D trigger input pin.
	ANEX0	I/O	This is the extended analog input pin for the A/D converter, and is the output in external op-amp connection mode.
	ANEX1	I	This is the extended analog input pin for the A/D converter.
I/O port	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P9_0 to P9_7, P10_0 to P10_7	I/O	8-bit I/O ports in CMOS, having a direction register to select an input or output. Each pin is set as an input port or output port. An input port can be set for a pull-up or for no pull-up in 4-bit unit by program. (however, P7_0 and P7_1 for the N-channel open drain output.)
	P8_0 to P8_4, P8_6, P8_7	I/O	I/O ports having equivalent functions to P0.
Input port	P8_5	I	Input pin for the $\overline{\text{NMI}}$ interrupt. Pin states can be read by the P8_5 bit in the P8 register.

I : Input O : Output I/O : Input and output

2. Central Processing Unit (CPU)

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



NOTES:
1. These registers comprise a register bank. There are two register banks.

Figure 2.1 Central Processing Unit Register

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

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

The R0 register can be separated between high (R0H) and low (R0L) for use as two 8-bit data registers.

R1H and R1L are the same as R0H and R0L. Conversely, R2 and R0 can be combined for use as a 32-bit data register (R2R0). R3R1 is the same as R2R0.

2.2 Address Registers (A0 and A1)

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

2.3 Frame Base Register (FB)

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

2.4 Interrupt Table Register (INTB)

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

2.5 Program Counter (PC)

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

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

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

2.7 Static Base Register (SB)

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

2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

2.8.1 Carry Flag (C Flag)

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

2.8.2 Debug Flag (D Flag)

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

2.8.3 Zero Flag (Z Flag)

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

2.8.4 Sign Flag (S Flag)

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

2.8.5 Register Bank Select Flag (B Flag)

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

2.8.6 Overflow Flag (O Flag)

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

2.8.7 Interrupt Enable Flag (I Flag)

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

2.8.8 Stack Pointer Select Flag (U Flag)

ISP is selected when the U flag is “0”; USP is selected when the U flag is “1”.
The U flag is cleared to “0” when a hardware interrupt request is accepted or an INT instruction for software interrupt Nos. 0 to 31 is executed.

2.8.9 Processor Interrupt Priority Level (IPL)

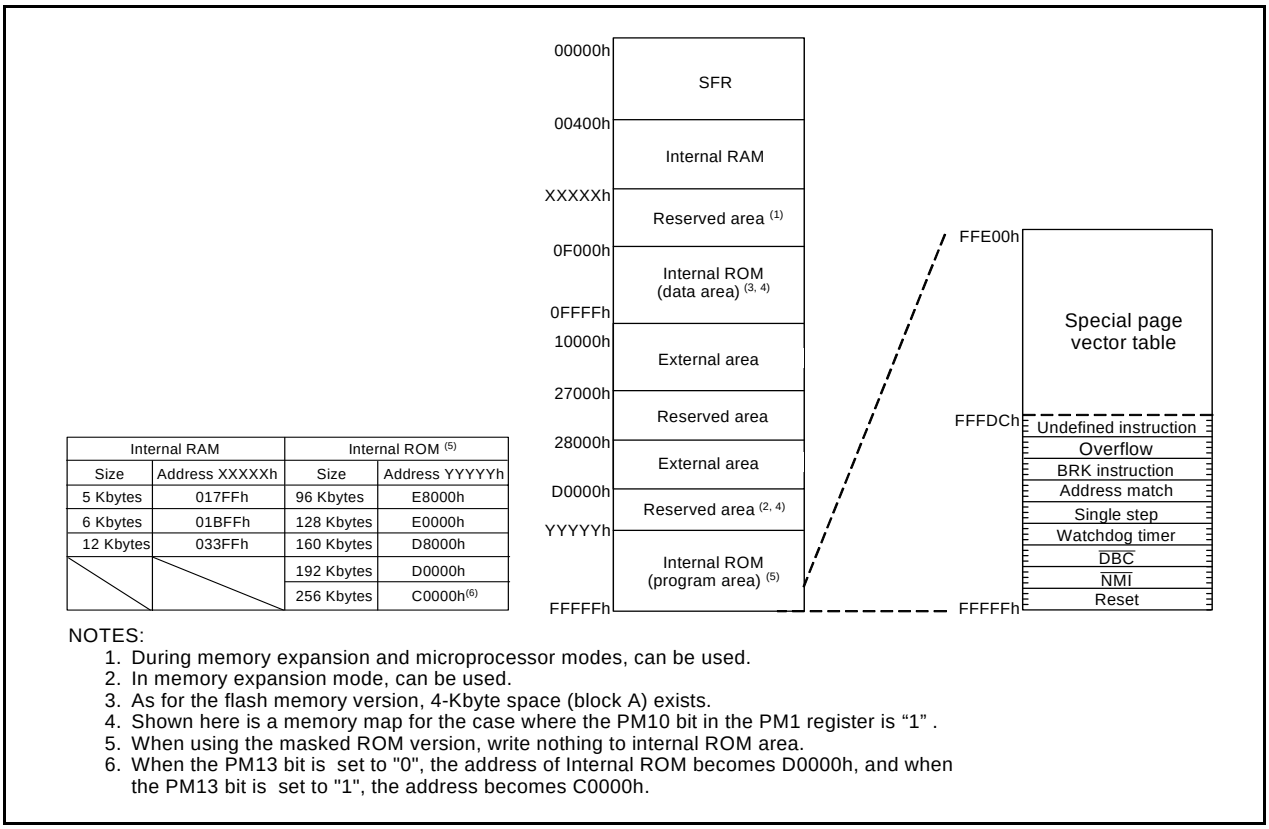
IPL is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.
If a requested interrupt has priority greater than IPL, the interrupt is enabled.

2.8.10 Reserved Area

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

3. Memory

Figure 3.1 is a Memory Map of the M16C/30P group. The address space extends the 1 Mbyte from address 00000h to FFFFFh.
The internal ROM is allocated in a lower address direction beginning with address FFFFFh. For example, a 64-Kbyte internal ROM is allocated to the addresses from F0000h to FFFFFh.
The fixed interrupt vector table is allocated to the addresses from FFFDCh to FFFFFh. Therefore, store the start address of each interrupt routine here.
The internal RAM is allocated in an upper address direction beginning with address 00400h. For example, a 5-Kbyte internal RAM is allocated to the addresses from 00400h to 017FFh. In addition to storing data, the internal RAM also stores the stack used when calling subroutines and when interrupts are generated. The SFR is allocated to the addresses from 00000h to 003FFh. Peripheral function control registers are located here. Of the SFR, any area which has no functions allocated is reserved for future use and cannot be used by users.
The special page vector table is allocated to the addresses from FFE00h to FFFDBh. This vector is used by the JMPS or JSRS instruction. For details, refer to the **M16C/60 and M16C/20 Series Software Manual**.



Internal RAM		Internal ROM (5)	
Size	Address XXXXXh	Size	Address YYYYYh
5 Kbytes	017FFh	96 Kbytes	E8000h
6 Kbytes	01BFFh	128 Kbytes	E0000h
12 Kbytes	033FFh	160 Kbytes	D8000h
		192 Kbytes	D0000h
		256 Kbytes	C0000h(6)

NOTES:
1. During memory expansion and microprocessor modes, can be used.
2. In memory expansion mode, can be used.
3. As for the flash memory version, 4-Kbyte space (block A) exists.
4. Shown here is a memory map for the case where the PM10 bit in the PM1 register is “1” .
5. When using the masked ROM version, write nothing to internal ROM area.
6. When the PM13 bit is set to "0", the address of Internal ROM becomes D0000h, and when the PM13 bit is set to "1", the address becomes C0000h.

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 4.5 list the SFR information.

Table 4.1 SFR Information (1) (1)

Address	Register	Symbol	After Reset
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0 (2)	PM0	00000000b(CNVSS pin is "L") 00000011b(CNVSS pin is "H")
0005h	Processor Mode Register 1	PM1	00XXX0X0b
0006h	System Clock Control Register 0	CM0	01001000b
0007h	System Clock Control Register 1	CM1	00100000b
0008h	Chip Select Control Register	CSR	00000001b
0009h	Address Match Interrupt Enable Register	AIER	XXXXXX00b
000Ah	Protect Register	PRCR	XX000000b
000Bh			
000Ch			
000Dh			
000Eh	Watchdog Timer Start Register	WDTS	XXh
000Fh	Watchdog Timer Control Register	WDC	00XXXXXXb
0010h 0011h 0012h	Address Match Interrupt Register 0	RMAD0	00h 00h X0h
0013h			
0014h 0015h 0016h	Address Match Interrupt Register 1	RMAD1	00h 00h X0h
0017h			
0018h			
0019h			
001Ah			
001Bh			
001Ch			
001Dh			
001Eh			
001Fh			
0020h 0021h 0022h	DMA0 Source Pointer	SAR0	XXh XXh XXh
0023h			
0024h 0025h 0026h	DMA0 Destination Pointer	DAR0	XXh XXh XXh
0027h			
0028h 0029h	DMA0 Transfer Counter	TCR0	XXh XXh
002Ah			
002Bh			
002Ch	DMA0 Control Register	DM0CON	00000X00b
002Dh			
002Eh			
002Fh			
0030h 0031h 0032h	DMA1 Source Pointer	SAR1	XXh XXh XXh
0033h			
0034h 0035h 0036h	DMA1 Destination Pointer	DAR1	XXh XXh XXh
0037h			
0038h 0039h	DMA1 Transfer Counter	TCR1	XXh XXh
003Ah			
003Bh			
003Ch	DMA1 Control Register	DM1CON	00000X00b
003Dh			
003Eh			
003Fh			

NOTES:
1. The blank areas are reserved and cannot be accessed by users.
2. The PM00 and PM01 bits do not change at software reset.

X : Nothing is mapped to this bit

Table 4.2 SFR Information (2) (1)

Address	Register	Symbol	After Reset
0040h			
0041h			
0042h			
0043h			
0044h	INT3 Interrupt Control Register	INT3IC	XX00X000b
0045h			
0046h	UART1 BUS Collision Detection Interrupt Control Register	U1BCNIC	XXXXX000b
0047h	UART0 BUS Collision Detection Interrupt Control Register	U0BCNIC	XXXXX000b
0048h			
0049h	INT4 Interrupt Control Register	INT4IC	XX00X000b
004Ah	UART2 Bus Collision Detection Interrupt Control Register	BCNIC	XXXXX000b
004Bh	DMA0 Interrupt Control Register	DM0IC	XXXXX000b
004Ch	DMA1 Interrupt Control Register	DM1IC	XXXXX000b
004Dh	Key Input Interrupt Control Register	KUPIC	XXXXX000b
004Eh	A/D Conversion Interrupt Control Register	ADIC	XXXXX000b
004Fh	UART2 Transmit Interrupt Control Register	S2TIC	XXXXX000b
0050h	UART2 Receive Interrupt Control Register	S2RIC	XXXXX000b
0051h	UART0 Transmit Interrupt Control Register	S0TIC	XXXXX000b
0052h	UART0 Receive Interrupt Control Register	S0RIC	XXXXX000b
0053h	UART1 Transmit Interrupt Control Register	S1TIC	XXXXX000b
0054h	UART1 Receive Interrupt Control Register	S1RIC	XXXXX000b
0055h	Timer A0 Interrupt Control Register	TA0IC	XXXXX000b
0056h	Timer A1 Interrupt Control Register	TA1IC	XXXXX000b
0057h	Timer A2 Interrupt Control Register	TA2IC	XXXXX000b
0058h			
0059h			
005Ah	Timer B0 Interrupt Control Register	TB0IC	XXXXX000b
005Bh	Timer B1 Interrupt Control Register	TB1IC	XXXXX000b
005Ch	Timer B2 Interrupt Control Register	TB2IC	XXXXX000b
005Dh	INT0 Interrupt Control Register	INT0IC	XX00X000b
005Eh	INT1 Interrupt Control Register	INT1IC	XX00X000b
005Fh	INT2 Interrupt Control Register	INT2IC	XX00X000b
0060h to 01AFh			
01B0h			
01B1h			
01B2h			
01B3h			
01B4h			
01B5h	Flash Memory Control Register 1 (2)	FMR1	0X00XX0Xb
01B6h			
01B7h	Flash Memory Control Register 0 (3)	FMR0	00000001b
01B8h			
01B9h			
01BAh			
01BBh			
01BCh			
01BDh			
01BEh			
01BFh			
01C0h to 024Fh			
0250h			
0251h			
0252h			
0253h			
0254h			
0255h			
0256h			
0257h			
0258h			
0259h			
025Ah			
025Bh			
025Ch			
025Dh			
025Eh	Peripheral Clock Select Register	PCLKR	00000011b
025Fh			
0260h to 033Fh			

NOTES:
1. The blank areas are reserved and cannot be accessed by users.
2. This register is included in the flash memory version.
3. This register is included in the flash memory version and one time flash version.

X : Nothing is mapped to this bit

Table 4.3 SFR Information (3) (1)

Address	Register	Symbol	After Reset
0340h			
0341h			
0342h			
0343h			
0344h			
0345h			
0346h			
0347h			
0348h			
0349h			
034Ah			
034Bh			
034Ch			
034Dh			
034Eh			
034Fh			
0350h			
0351h			
0352h			
0353h			
0354h			
0355h			
0356h			
0357h			
0358h			
0359h			
035Ah			
035Bh			
035Ch			
035Dh			
035Eh	Interrupt Factor Select Register 2	IFSR2A	00XXXXXXb
035Fh	Interrupt Factor Select Register	IFSR	00h
0360h			
0361h			
0362h			
0363h			
0364h			
0365h			
0366h			
0367h			
0368h			
0369h			
036Ah			
036Bh			
036Ch	UART0 Special Mode Register 4	U0SMR4	00h
036Dh	UART0 Special Mode Register 3	U0SMR3	000X0X0Xb
036Eh	UART0 Special Mode Register 2	U0SMR2	X0000000b
036Fh	UART0 Special Mode Register	U0SMR	X0000000b
0370h	UART1 Special Mode Register 4	U1SMR4	00h
0371h	UART1 Special Mode Register 3	U1SMR3	000X0X0Xb
0372h	UART1 Special Mode Register 2	U1SMR2	X0000000b
0373h	UART1 Special Mode Register	U1SMR	X0000000b
0374h	UART2 Special Mode Register 4	U2SMR4	00h
0375h	UART2 Special Mode Register 3	U2SMR3	000X0X0Xb
0376h	UART2 Special Mode Register 2	U2SMR2	X0000000b
0377h	UART2 Special Mode Register	U2SMR	X0000000b
0378h	UART2 Transmit/Receive Mode Register	U2MR	00h
0379h	UART2 Bit Rate Generator	U2BRG	XXh
037Ah	UART2 Transmit Buffer Register	U2TB	XXh
037Bh			XXh
037Ch	UART2 Transmit/Receive Control Register 0	U2C0	00001000b
037Dh	UART2 Transmit/Receive Control Register 1	U2C1	00000010b
037Eh	UART2 Receive Buffer Register	U2RB	XXh
037Fh			XXh

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

X : Nothing is mapped to this bit

Table 4.4 SFR Information (4) (1)

Address	Register	Symbol	After Reset
0380h	Count Start Flag	TABSR	000XX000b
0381h	Clock Prescaler Reset Fag	CPSRF	0XXXXXXXb
0382h	One-Shot Start Flag	ONSF	00XXX000b
0383h	Trigger Select Register	TRGSR	XXXX0000b
0384h	Up-Down Flag	UDF	XX0XX000b (2)
0385h			
0386h 0387h	Timer A0 Register	TA0	XXh XXh
0388h 0389h	Timer A1 Register	TA1	XXh XXh
038Ah 038Bh	Timer A2 Register	TA2	XXh XXh
038Ch			
038Dh			
038Eh			
038Fh			
0390h 0391h	Timer B0 Register	TB0	XXh XXh
0392h 0393h	Timer B1 Register	TB1	XXh XXh
0394h 0395h	Timer B2 Register	TB2	XXh XXh
0396h	Timer A0 Mode Register	TA0MR	00h
0397h	Timer A1 Mode Register	TA1MR	00h
0398h	Timer A2 Mode Register	TA2MR	00h
0399h			
039Ah			
039Bh	Timer B0 Mode Register	TB0MR	00XX0000b
039Ch	Timer B1 Mode Register	TB1MR	00XX0000b
039Dh	Timer B2 Mode Register	TB2MR	00XX0000b
039Eh			
039Fh			
03A0h	UART0 Transmit/Receive Mode Register	U0MR	00h
03A1h	UART0 Bit Rate Generator	U0BRG	XXh
03A2h 03A3h	UART0 Transmit Buffer Register	U0TB	XXh XXh
03A4h	UART0 Transmit/Receive Control Register 0	U0C0	00001000b
03A5h	UART0 Transmit/Receive Control Register 1	U0C1	00000010b
03A6h 03A7h	UART0 Receive Buffer Register	U0RB	XXh XXh
03A8h	UART1 Transmit/Receive Mode Register	U1MR	00h
03A9h	UART1 Bit Rate Generator	U1BRG	XXh
03AAh 03ABh	UART1 Transmit Buffer Register	U1TB	XXh XXh
03ACh	UART1 Transmit/Receive Control Register 0	U1C0	00001000b
03ADh	UART1 Transmit/Receive Control Register 1	U1C1	00000010b
03AEh 03AFh	UART1 Receive Buffer Register	U1RB	XXh XXh
03B0h	UART Transmit/Receive Control Register 2	UCON	X0000000b
03B1h			
03B2h			
03B3h			
03B4h			
03B5h			
03B6h			
03B7h			
03B8h	DMA0 Request Factor Select Register	DM0SL	00h
03B9h			
03BAh	DMA1 Request Factor Select Register	DM1SL	00h
03BBh			
03BCh 03BDh	CRC Data Register	CRCD	XXh XXh
03BEh	CRC Input Register	CRCIN	XXh
03BFh			

NOTES:
1. The blank areas are reserved and cannot be accessed by users.
2. Bit 5 in the Up-down flag is “0” by reset. However, The values in these bits when read are indeterminate.

X : Nothing is mapped to this bit

Table 4.5 SFR Information (5) (1)

Address	Register	Symbol	After Reset
03C0h	A/D Register 0	AD0	XXh
03C1h			XXh
03C2h	A/D Register 1	AD1	XXh
03C3h			XXh
03C4h	A/D Register 2	AD2	XXh
03C5h			XXh
03C6h	A/D Register 3	AD3	XXh
03C7h			XXh
03C8h	A/D Register 4	AD4	XXh
03C9h			XXh
03CAh	A/D Register 5	AD5	XXh
03CBh			XXh
03CCh	A/D Register 6	AD6	XXh
03CDh			XXh
03CEh	A/D Register 7	AD7	XXh
03CFh			XXh
03D0h			
03D1h			
03D2h			
03D3h			
03D4h	A/D Control Register 2	ADCON2	XXX000X0b
03D5h			
03D6h	A/D Control Register 0	ADCON0	000X0XXXb
03D7h	A/D Control Register 1	ADCON1	00000XXXb
03D8h			
03D9h			
03DAh			
03DBh			
03DCh			
03DDh			
03DEh			
03DFh			
03E0h	Port P0 Register	P0	XXh
03E1h	Port P1 Register	P1	XXh
03E2h	Port P0 Direction Register	PD0	00h
03E3h	Port P1 Direction Register	PD1	00h
03E4h	Port P2 Register	P2	XXh
03E5h	Port P3 Register	P3	XXh
03E6h	Port P2 Direction Register	PD2	00h
03E7h	Port P3 Direction Register	PD3	00h
03E8h	Port P4 Register	P4	XXh
03E9h	Port P5 Register	P5	XXh
03EAh	Port P4 Direction Register	PD4	00h
03EBh	Port P5 Direction Register	PD5	00h
03ECh	Port P6 Register	P6	XXh
03EDh	Port P7 Register	P7	XXh
03EEh	Port P6 Direction Register	PD6	00h
03EFh	Port P7 Direction Register	PD7	00h
03F0h	Port P8 Register	P8	XXh
03F1h	Port P9 Register	P9	XXh
03F2h	Port P8 Direction Register	PD8	00X00000b
03F3h	Port P9 Direction Register	PD9	00h
03F4h	Port P10 Register	P10	XXh
03F5h			
03F6h	Port P10 Direction Register	PD10	00h
03F7h			
03F8h			
03F9h			
03FAh			
03FBh			
03FCh	Pull-Up Control Register 0	PUR0	00h
03FDh	Pull-Up Control Register 1	PUR1	00000000b (2) 00000010b (2)
03FEh	Pull-Up Control Register 2	PUR2	00h
03FFh	Port Control Register	PCR	00h

NOTES:
1. The blank areas are reserved and cannot be accessed by users.
2. At hardware reset, the register is as follows:
   - "00000000b" where "L" is inputted to the CNVSS pin
   - "00000010b" where "H" is inputted to the CNVSS pin

   At software reset, the register is as follows:
   - "00000000b" where the PM01 to PM00 bits in the PM0 register are "00b" (single-chip mode).
   - "00000010b" where the PM01 to PM00 bits in the PM0 register are "01b" (memory expansion mode) or "11b" (microprocessor mode).

X : Nothing is mapped to this bit

5. Reset

Hardware reset and software reset are available to reset the microcomputer.

5.1 Hardware Reset

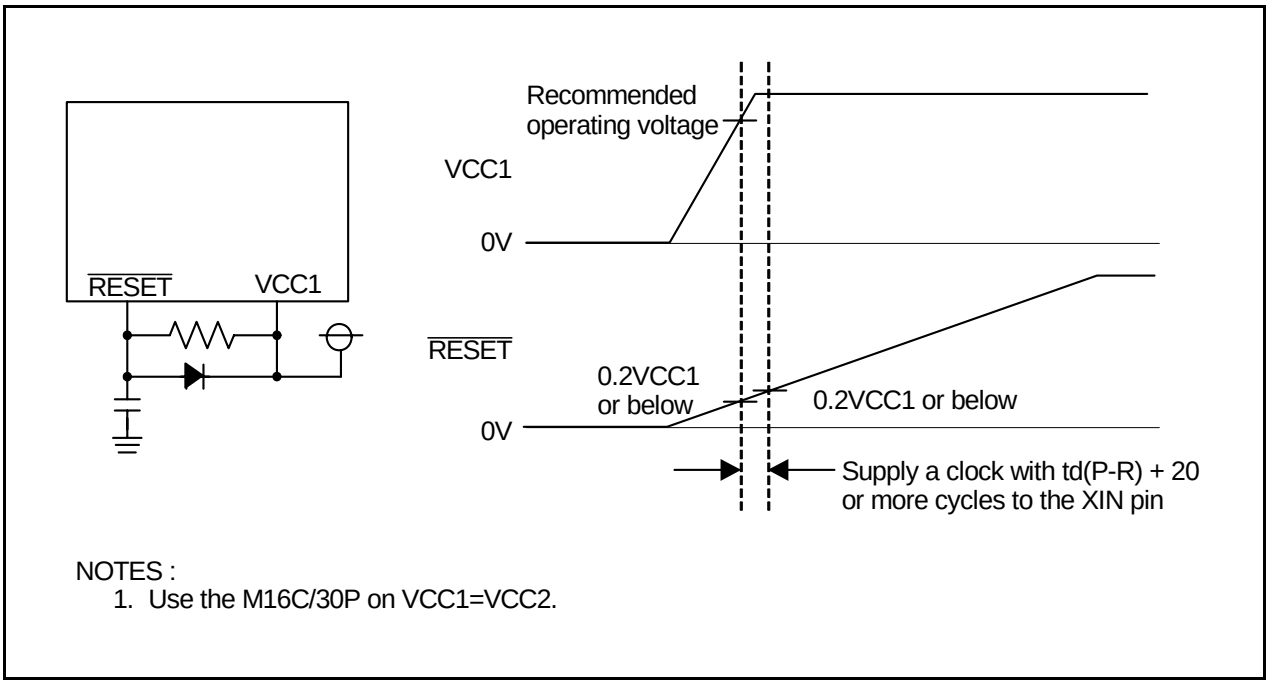
The microcomputer resets pins, the CPU and SFR by setting the $\overline{\text{RESET}}$ pin. If the supply voltage meets the recommended operating conditions, the microcomputer resets all pins when an "L" signal is applied to the $\overline{\text{RESET}}$ pin (see **Table 5.1 Pin Status When $\overline{\text{RESET}}$ Pin Level is "L"**). The oscillation circuit is also reset and the main clock starts oscillation. The microcomputer resets the CPU and SFR when the signal applied to the $\overline{\text{RESET}}$ pin changes low ("L") to high ("H"). The microcomputer executes the program in an address indicated by the reset vector. The internal RAM is not reset. When an "L" signal is applied to the $\overline{\text{RESET}}$ pin while writing data to the internal RAM, the internal RAM is in an indeterminate state.
Figure 5.1 shows an Example Reset Circuit. Figure 5.2 shows a Reset Sequence. Table 5.1 lists pin states while the $\overline{\text{RESET}}$ pin is held low ("L").

5.1.1 Reset on a Stable Supply Voltage

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

5.1.2 Power-on Reset

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



NOTES :
1. Use the M16C/30P on VCC1=VCC2.

Figure 5.1 Example Reset Circuit

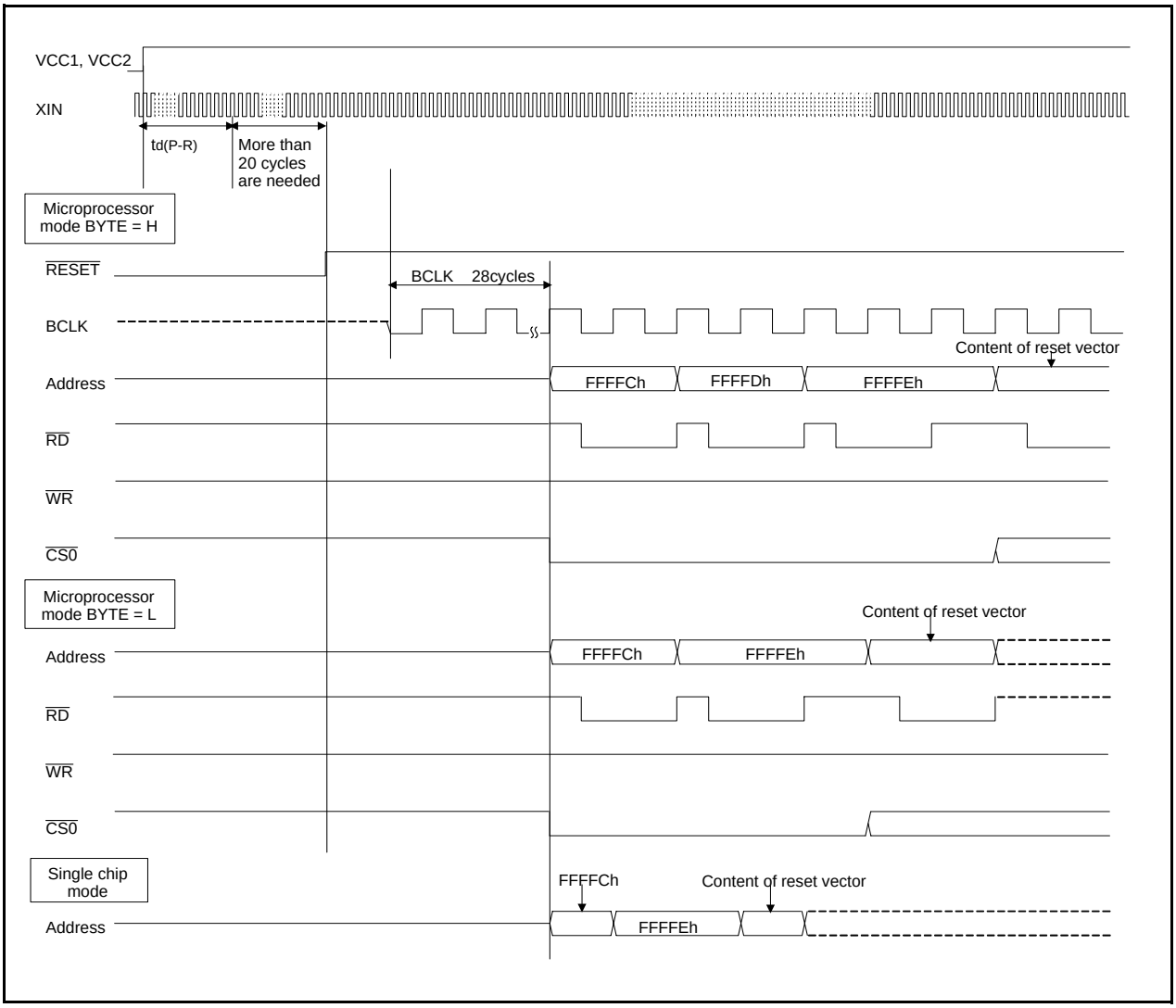


Figure 5.2 Reset Sequence

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

Pin Name	Status		
	CNVSS = VSS	CNVSS = VCC1	
		BYTE = VSS	BYTE = VCC1
P0	Input port	Data input	Data input
P1	Input port	Data input	Input port
P2, P3, P4_0 to P4_3	Input port	Address output (undefined)	Address output (undefined)
P4_4	Input port	$\overline{\text{CS0}}$ output ("H" is output)	$\overline{\text{CS0}}$ output ("H" is output)
P4_5 to P4_7	Input port	Input port (Pulled high)	Input port (Pulled high)
P5_0	Input port	$\overline{\text{WR}}$ output ("H" is output)	$\overline{\text{WR}}$ output ("H" is output)
P5_1	Input port	$\overline{\text{BHE}}$ output (undefined)	$\overline{\text{BHE}}$ output (undefined)
P5_2	Input port	$\overline{\text{RD}}$ output ("H" is output)	$\overline{\text{RD}}$ output ("H" is output)
P5_3	Input port	BCLK output	BCLK output
P5_4	Input port	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)
P5_5	Input port	$\overline{\text{HOLD}}$ input	$\overline{\text{HOLD}}$ input
P5_6	Input port	ALE output ("L" is output)	ALE output ("L" is output)
P5_7	Input port	$\overline{\text{RDY}}$ input	$\overline{\text{RDY}}$ input
P6, P7, P8_0 to P8_4, P8_6, P8_7, P9, P10	Input port	Input port	Input port

NOTES:

1. Shown here is the valid pin state when the internal power supply voltage has stabilized after power on.
   When CNVSS = VCC1, the pin state is indeterminate until the internal power supply voltage stabilizes.

5.2 Software Reset

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

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

5.3 Internal Space

Figure 5.3 shows CPU Register Status After Reset. Refer to **4. Special Function Register (SFR)** for SFR states after reset.

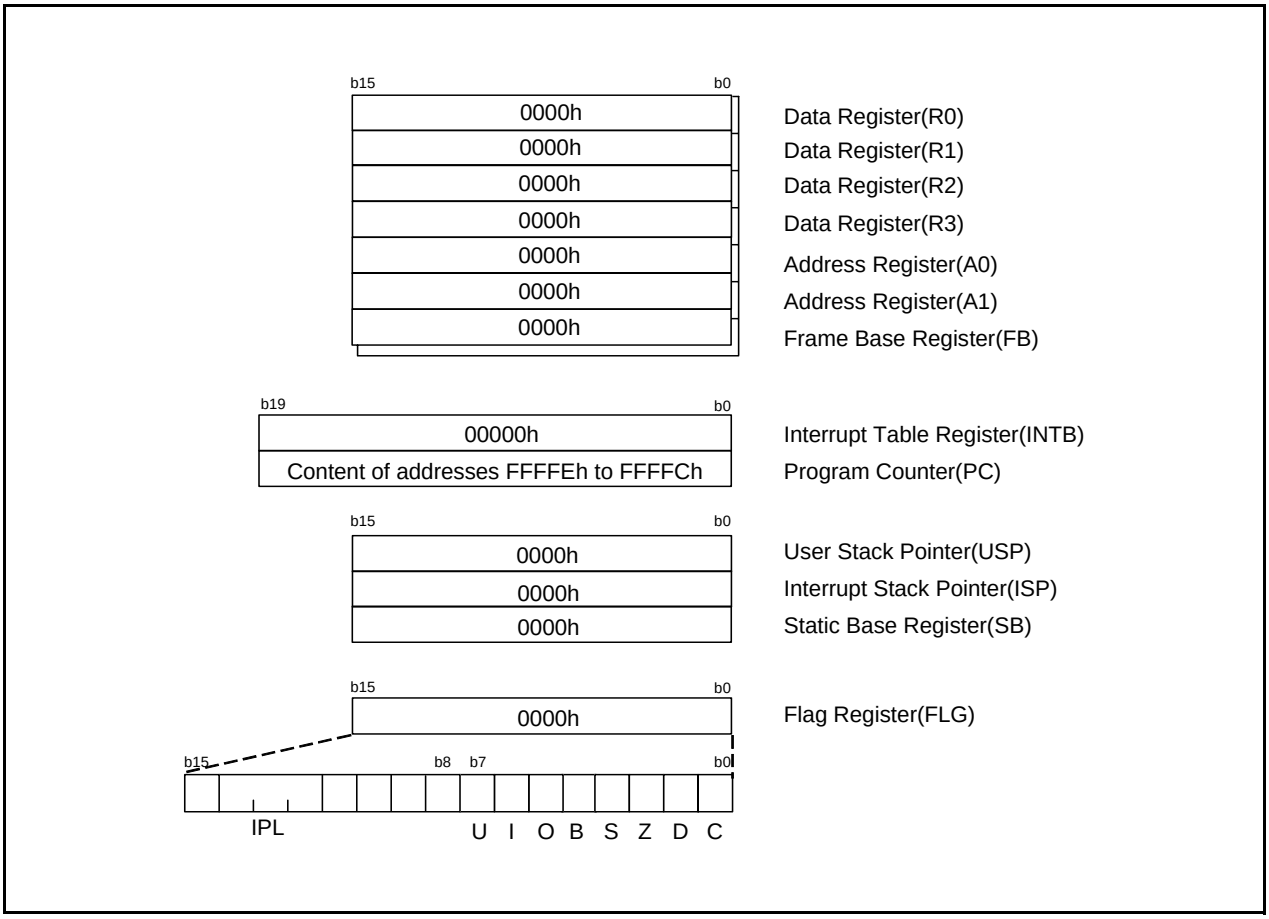


Figure 5.3 CPU Register Status After Reset

5.4 Cold Start-up / Warm Start-up Determine Function

As for the cold start-up/warm start-up determine function, the WDC5 flag in the WDC register determines either cold start-up (reset process) when power-on or warm start-up (reset process) when reset signal is applied during the microcomputer running.
Default value of the WDC5 bit is “0” (cold start-up) when power-on. It is set to “1” (warm start-up) by writing desired values to the WDC register. The WDC bit is not reset, regardless of a software reset or a reset operation.
Figure 5.4 shows Cold Start-up/Warm Start-up Determine Function Block Diagram. Figure 5.5 shows the Cold Start-up/Warm Start-up Determine Function Operation Example. Figure 5.6 shows WDC Register.

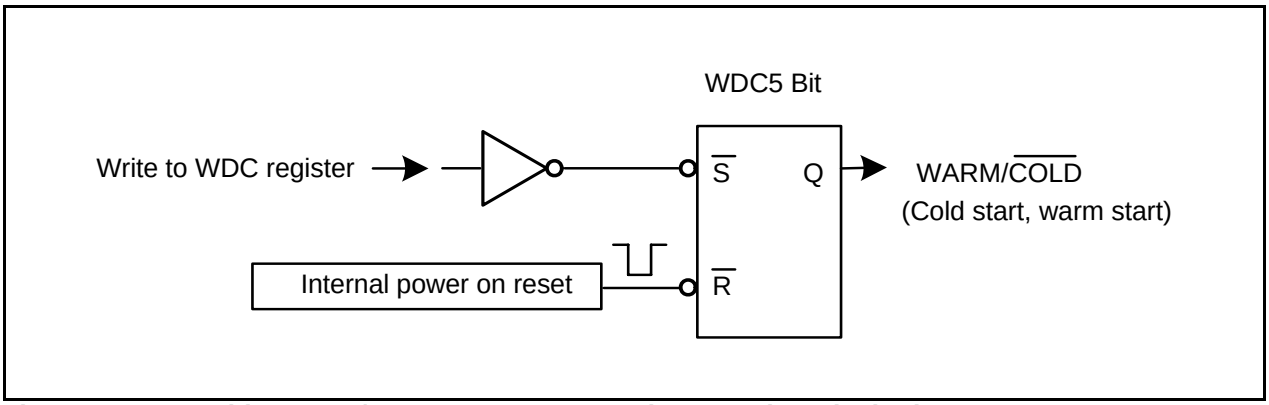
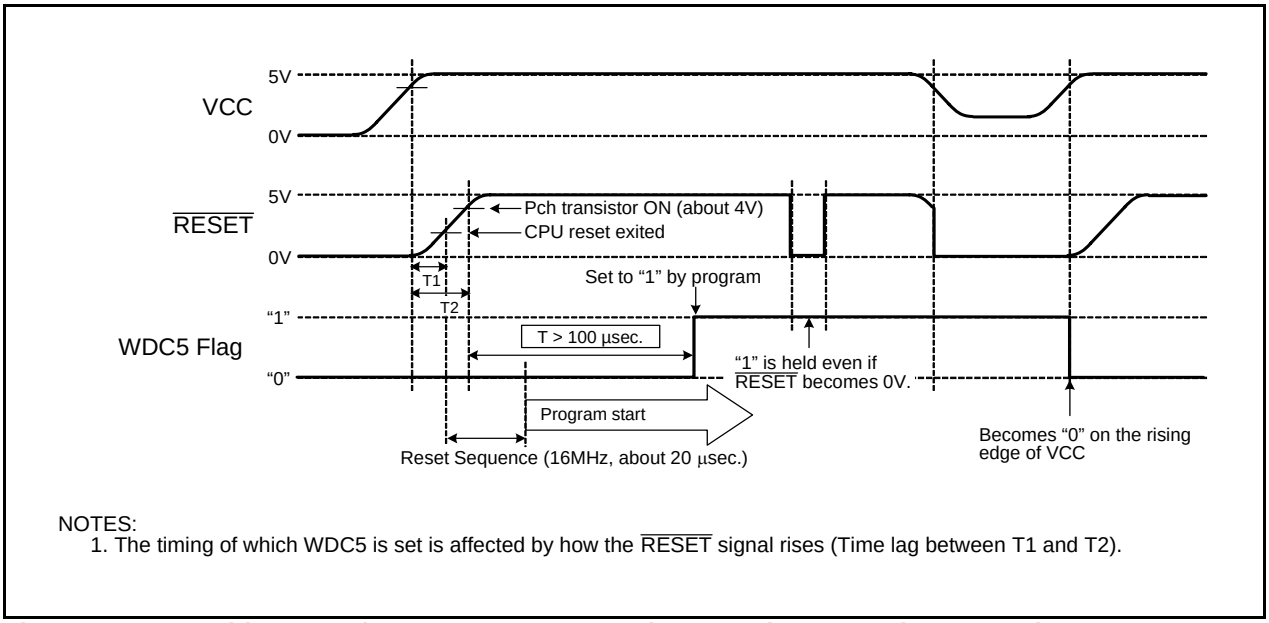


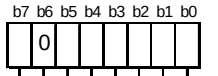
Figure 5.4 Cold Start-up/Warm Start-up Determine Function Block Diagram



NOTES:
1. The timing of which WDC5 is set is affected by how the RESET signal rises (Time lag between T1 and T2).

Figure 5.5 Cold Start-up/Warm Start-up Determine Function Operation Example

Watchdog Timer Control Register



Symbol	Address	After Reset
WDC	000Fh	00XXXXXXb

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	High-order Bit of Watchdog Timer		RO
WDC5	Cold Start / Warm Start Discrimination Flag(1, 2)	0 : Cold Start 1 : Warm Start	RW
— (b6)	Reserved Bit	Set to “0”	RW
WDC7	Prescaler Select Bit	0 : Divided by 16 1 : Divided by 128	RW

NOTES :
1. Writing to the WDC register causes the WDC5 bit to be set to “1” (warm start). If the voltage applied to VCC1 is less than 4.0 V, either write to this register when the CPU clock frequency is 2 MHz or write twice.
2. The WDC5 bit is set to “0” (cold start) when power is turned on and can be set to “1” by program only.

Figure 5.6 WDC Register

6. Processor Mode

6.1 Types of Processor Mode

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

Table 6.1 Features of Processor Modes

Processor Modes	Access Space	Pins which are Assigned I/O Ports
Single-Chip Mode	SFR, Internal RAM, Internal ROM	All pins are I/O ports or peripheral function I/O pins
Memory Expansion Mode	SFR, Internal RAM, Internal ROM, External Area (1)	Some pins serve as bus control pins (1)
Microprocessor Mode	SFR, Internal RAM, External Area (1)	Some pins serve as bus control pins (1)

NOTES:
1. Refer to **7. Bus**.

6.2 Setting Processor Modes

Processor mode is set by using the CNVSS pin and the PM01 to PM00 bits in the PM0 register.
Table 6.2 shows the Processor Mode After Hardware Reset. Table 6.3 shows the PM01 to PM00 Bits Set Values and Processor Modes.

Table 6.2 Processor Mode After Hardware Reset

CNVSS Pin Input Level	Processor Modes
VSS	Single-Chip Mode
VCC1 (1)	Microprocessor Mode

NOTES:
1. If the microcomputer is reset in hardware by applying VCC1 to the CNVSS pin (hardware reset), the internal ROM cannot be accessed regardless of PM01 to PM00 bits.

Table 6.3 PM01 to PM00 Bits Set Values and Processor Modes

PM01 to PM00 Bits	Processor Modes
00b	Single-Chip Mode
01b	Memory Expansion Mode
10b	Do not set
11b	Microprocessor Mode

Rewriting the PM01 to PM00 bits places the microcomputer in the corresponding processor mode regardless of whether the input level on the CNVSS pin is “H” or “L”. Note, however, that the PM01 to PM00 bits cannot be rewritten to “01b” (memory expansion mode) or “11b” (microprocessor mode) at the same time the PM07 to PM02 bits are rewritten. Note also that these bits cannot be rewritten to enter microprocessor mode in the internal ROM, nor can they be rewritten to exit microprocessor mode in areas overlapping the internal ROM.
If the microcomputer is reset in hardware by applying VCC1 to the CNVSS pin (hardware reset 1 or brown-out detection reset (hardware reset 2)), the internal ROM cannot be accessed regardless of PM01 to PM00 bits.
Figures 6.1 and 6.2 show the PM0 Register and PM1 Register (1). Figure 6.4 show the Memory Map in Single Chip Mode.

Processor Mode Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0				

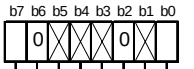
Symbol	Address	After Reset (3)
PM0	0004h	00000000b (CNVSS pin = L) 00000011b (CNVSS pin = H)

Bit Symbol	Bit Name	Function	RW
PM00	Processor Mode Bit (2)	b1 b0 0 0 : Single-chip mode 0 1 : Memory expansion mode 1 0 : Do not set 1 1 : Microprocessor mode	RW
PM01			RW
PM02	R/W Mode Select Bit (3)	0 : $\overline{RD}$, $\overline{BHE}$, $\overline{WR}$ 1 : $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$	RW
PM03	Software Reset Bit	Setting this bit to “1” resets the microcomputer. When read, its content is “0”.	RW
— (b4)	Reserved Bit	Set to “0”.	RW
— (b5)	Reserved Bit	Set to “0”.	RW
PM06	Port P4_0 to P4_3 Function Select Bit (3)	0 : Address output 1 : Port function (Address is not output)	RW
PM07	BCLK Output Disable Bit (3)	0 : BCLK is output 1 : BCLK is not output (Pin is left high-impedance)	RW

NOTES :
1. Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable).
2 The PM01 to PM00 bits do not change at software reset.
3 Effective when the PM01 to PM00 bits are set to “01b” (memory expansion mode) or “11b” (microprocessor mode).

Figure 6.1 PM0 Register

Processor Mode Register 1 (1)



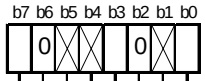
Symbol	Address	After Reset
PM1	0005h	00XXX0X0b

Bit Symbol	Bit Name	Function	RW
PM10	CS2 Area Switch Bit (Data Block Enable Bit) (2)	0 : 08000h to 26FFFh (Block A disable) 1 : 10000h to 26FFFh (Block A enable)	RW
— (b1)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
— (b2)	Reserved Bit	Set to “0”.	RW
— (b3)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
— (b4)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
— (b5)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
— (b6)	Reserved Bit	Set to “0”.	RW
PM17	Wait Bit(3)	0 : No wait state 1 : With wait state (1 wait)	RW

NOTES :
1. Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable).
2. Set the PM10 bit to “0” for Mask ROM version and one time flash version. For flash memory version, the PM10 bit controls whether Block A is enabled or disabled. When the PM10 bit is set to “1”, 0F000h to 0FFFFh can be used as internal ROM area.
   In addition, for the flash memory version and one time flash version, the PM10 bit is automatically set to “1” while the FMR01 bit in the FMR0 register is set to “1” (CPU rewrite mode or the FMSTP bit enabled).
3. When PM17 bit is set to “1” (with wait state), one wait state is inserted when accessing the internal RAM, or internal ROM.
   When PM17 bit is set to “1” and accesses an external area, set the CSiW bit in the CSR register (i=0 to 3) to “0” (with wait state).

Figure 6.2 PM1 Register (1)

Processor Mode Register 1 (1)



Symbol	Address	After Reset
PM1	0005h	00XXX0X0b

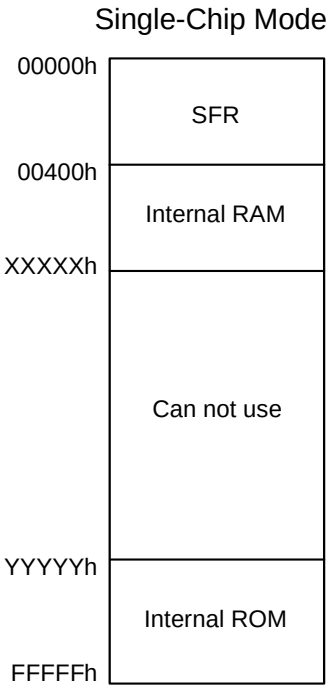
Bit Symbol	Bit Name	Function	RW
PM10	CS2 Area Switch Bit (2)	0 : 08000h to 26FFFh 1 : 10000h to 26FFFh	RW
— (b1)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
— (b2)	Reserved Bit	Set to “0”.	RW
PM13	Internal Reserved Area Expansion Bit (4)	(NOTE 5)	RW
— (b4)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
— (b5)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
— (b6)	Reserved Bit	Set to “0”.	RW
PM17	Wait Bit (3)	0 : No wait state 1 : With wait state (1 wait)	RW

NOTES :

1. Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable).
2. Set the PM10 bit to “0” for one time flash version.
3. When PM17 bit is set to “1” (with wait state), one wait state is inserted when accessing the internal RAM, or internal ROM.
   When PM17 bit is set to “1” and accesses an external area, set the CSiW bit in the CSR register (i=0 to 3) to “0” (with
4. The PM13 bit is automatically set to “1” when the FMR01 bit in the FMR0 register is “1” (CPU rewrite mode).
5. The access area is changed by the PM13 bit as listed in the table below.

Access Area		PM13=0	PM13=1
Internal	RAM	Up to Addresses 00400h to 03FFFh (15 Kbytes	The entire area is usable
	ROM	Up to Addresses D0000h to FFFFFh (192 Kbyte	The entire area is usable
External		Address 04000h to 07FFFh are usable Address 80000h to CFFFFh are usable	Address 04000h to 07FFFh are reserved Address 80000h to CFFFFh are reserved (Memory expansion mode)

Figure 6.3 PM1 Register (2) (M30304GDPFP, M30304GDPGP, M30304GEPFP, M30304GEPGP, M30302GGPFP, M30302GGPGP)



PM13=0 or without PM13 bit

Internal RAM		Internal ROM	
Size	Address XXXXXh	Size	Address YYYYYh
5 Kbytes	017FFh	96 Kbytes	E8000h
6 Kbytes	01BFFh	128 Kbytes	E0000h
12 Kbytes	033FFh	160 Kbytes	D8000h
		192 Kbytes	D0000h
		256 Kbytes	D0000h(2)

PM13=1

Internal RAM		Internal ROM	
Size	Address XXXXXh	Size	Address YYYYYh
12 Kbytes	033FFh	160 Kbytes	D8000h
		192 Kbytes	D0000h
		256 Kbytes	C0000h

NOTE :
1. For the mask ROM version and one time flash version set the PM10 bit to “0” (08000h to 26FFFh for $\overline{CS2}$ area).
2. 192-Kbyte internal ROM space is available for use.

Figure 6.4 Memory Map in Single Chip Mode

7. Bus

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

7.1 Bus Mode

The bus mode is the "separate bus mode" that separates data and address.

7.2 Bus Control

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

7.2.1 Address Bus

The address bus consists of 20 lines, A0 to A19. The address bus width can be chosen to be 16 or 20 bits by using the PM06 bit in the PM0 register. Table 7.1 shows the PM06 Bit Set Value and Address Bus Width.

Table 7.1 PM06 Bit Set Value and Address Bus Width

Set Value (1)	Pin Function	Address Bus Width
PM06=1	P4_0 to P4_3	16 bits
PM06=0	A16 to A19	20 bits

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

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

7.2.2 Data Bus

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

7.2.3 Chip Select Signal

The chip select (hereafter referred to as the $\overline{CS}$) signals are output from the $\overline{CSi}$ (i = 0 to 3) pins. These pins can be chosen to function as I/O ports or as $\overline{CS}$ by using the $\overline{CS}$ bit in the CSR register.
Figure 7.1 shows the CSR Register.
During 1-Mbyte mode, the external area can be separated into up to 4 by the $\overline{CSi}$ signal which is output from the $\overline{CSi}$ pin.
Figure 7.2 shows the Example of Address Bus and $\overline{CSi}$ Signal Output in 1-Mbyte mode.

Chip Select Control Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
CSR	0008h	01h

Bit Symbol	Bit Name	Function	RW
CS0	$\overline{CS0}$ Output Enable Bit	0 : Chip select output disabled (functions as I/O port) 1 : Chip select output enabled	RW
CS1	$\overline{CS1}$ Output Enable Bit		RW
CS2	$\overline{CS2}$ Output Enable Bit		RW
CS3	$\overline{CS3}$ Output Enable Bit		RW
CS0W	$\overline{CS0}$ Wait Bit	0 : With wait state (1 wait) 1 : Without wait state (1, 2)	RW
CS1W	$\overline{CS1}$ Wait Bit		RW
CS2W	$\overline{CS2}$ Wait Bit		RW
CS3W	$\overline{CS3}$ Wait Bit		RW

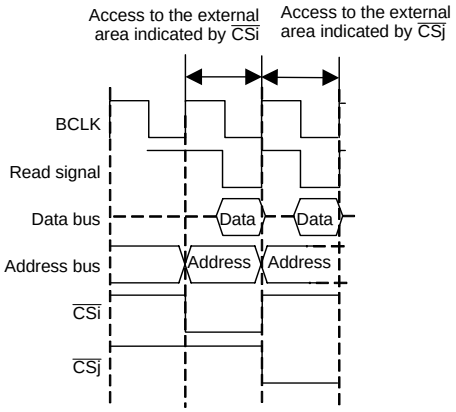
NOTES :
1. Where the $\overline{RDY}$ signal is used in the area indicated by $\overline{CSi}$ (i = 0 to 3) or the multiplex bus is used, set the CSiW bit to "0" (with wait state).
2. If the PM17 bit in the PM1 register is set to "1" (with wait state), set the CSiW bit to "0" (with wait state).

Figure 7.1 CSR Register

Example 1

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

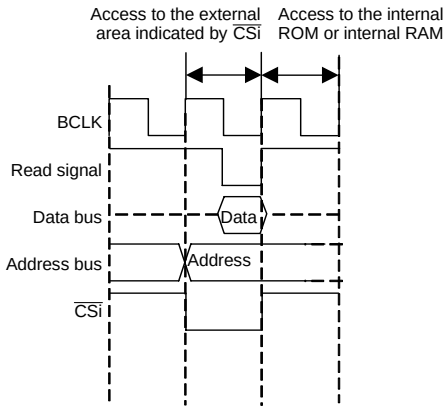
The address bus and the chip select signal both change state between these two cycles.



Example 2

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

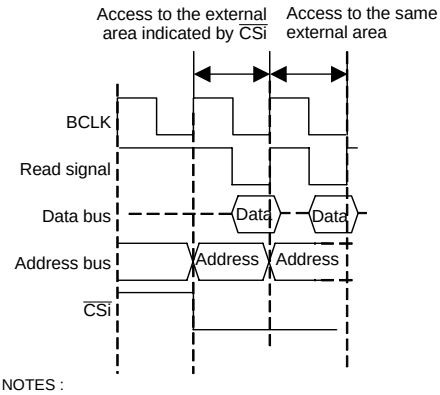
The chip select signal changes state but the address bus does not change state



Example 3

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

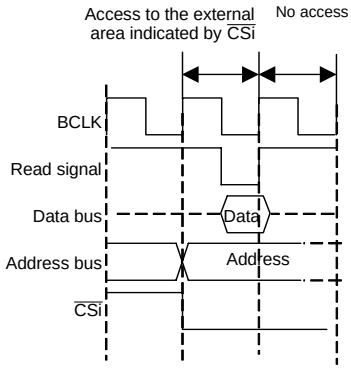
The address bus changes state but the chip select signal does not change state



Example 4

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

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



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

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

Figure 7.2 Example of Address Bus and $\overline{CSi}$ Signal Output in 1-Mbyte mode

7.2.4 Read and Write Signals

When the data bus is 16 bits wide, the read and write signals can be chosen to be a combination of $\overline{RD}$, $\overline{BHE}$ and $\overline{WR}$ or a combination of $\overline{RD}$, $\overline{WRL}$ and $\overline{WRH}$ by using the PM02 bit in the PM0 register. When the data bus is 8 bits wide, use a combination of $\overline{RD}$, $\overline{WR}$ and $\overline{BHE}$.
Table 7.2 shows the Operation of $\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$ Signals. Table 7.3 shows the Operation of $\overline{RD}$, $\overline{WRL}$, and $\overline{BHE}$ Signals.

Table 7.2 Operation of $\overline{RD}$, $\overline{WRL}$ and $\overline{WRH}$ Signals

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

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

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

7.2.5 ALE Signal

The ALE signal latches the address.

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

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

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

Then, when the input on the $\overline{\text{RDY}}$ pin is detected high at the falling edge of BCLK, the remaining bus cycle is executed. Figure 7.3 shows Example in which the Wait State was Inserted into Read Cycle by $\overline{\text{RDY}}$ Signal. To use the $\overline{\text{RDY}}$ signal, set the corresponding bit (CS3W to CS0W bits) in the CSR register to "0" (with wait state). When not using the $\overline{\text{RDY}}$ signal, the $\overline{\text{RDY}}$ pin must be pulled-up.

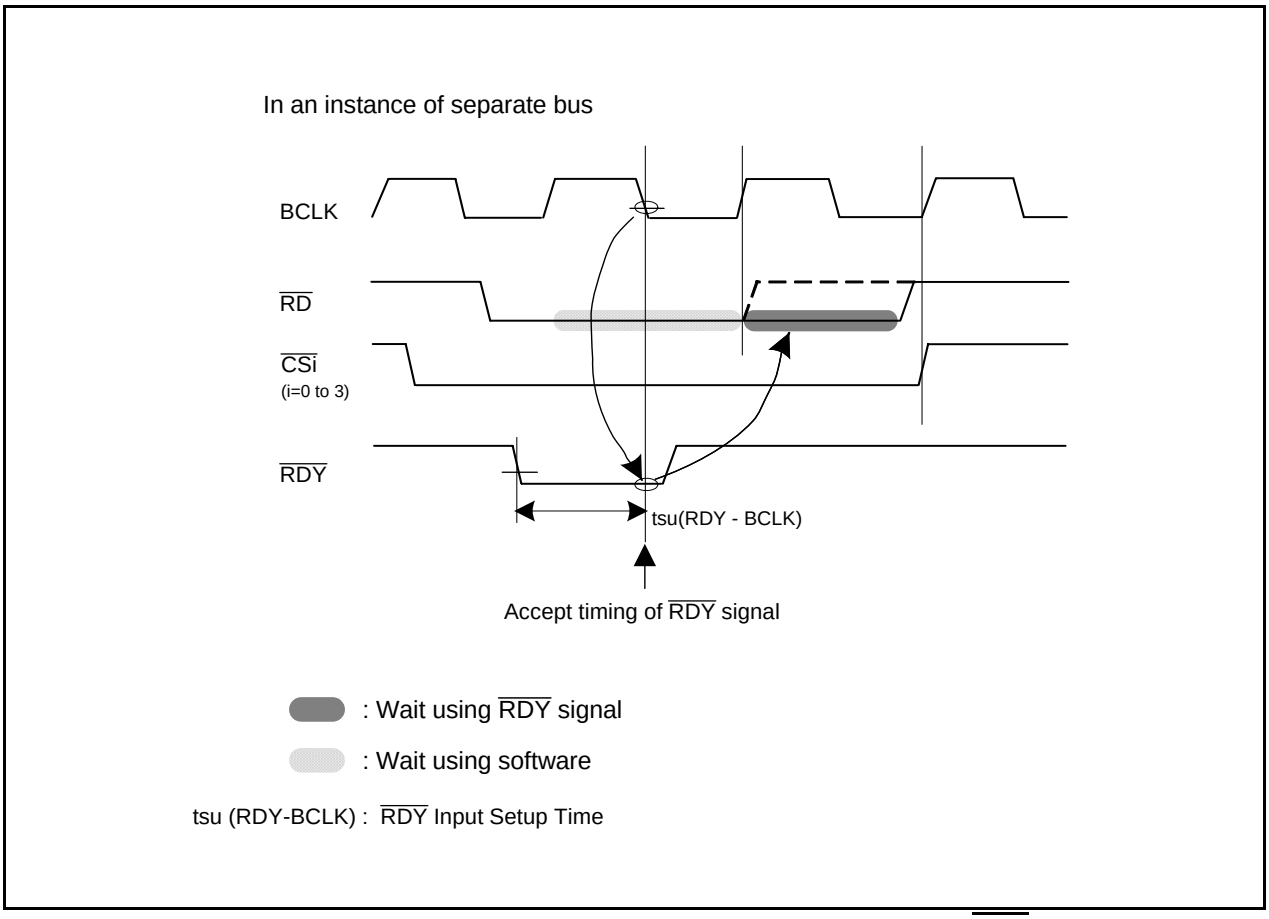


Figure 7.3 Example in which Wait State was Inserted into Read Cycle by $\overline{\text{RDY}}$ Signal

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

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

Table 7.4 shows the Microcomputer Status in Hold State.

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

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

Figure 7.4 Bus-Using Priorities

Table 7.4 Microcomputer Status in Hold State

Item		Status
BCLK		Output
A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}$,$\overline{\text{WRH}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$		High-impedance
I/O ports	P0, P1, P3, P4 (1)	High-impedance
	P6 to P10	Maintains status when $\overline{\text{HOLD}}$ signal is received
$\overline{\text{HLDA}}$		Output “L”
Internal Peripheral Circuits		ON (but watchdog timer stops)
ALE Signal		Undefined

NOTES:
1. When I/O port function is selected.

7.2.8 BCLK Output

If the PM07 bit in the PM0 register is set to “0” (output enable), a clock with the same frequency as that of the CPU clock is output as BCLK from the BCLK pin. Refer to **9.2 CPU Clock and Peripheral Function Clock**.

Table 7.5 Pin Functions for Each Processor Mode

Processor Mode		Memory Expansion Mode or Microprocessor Mode	
Data Bus Width BYTE Pin		8 bits "H"	16 bits "L"
P0_0 to P0_7		D0 to D7	D0 to D7
P1_0 to P1_7		I/O ports	D8 to D15
P2_0		A0	A0
P2_1 to P2_7		A1 to A7	A1 to A7
P3_0		A8	A8
P3_1 to P3_7		A9 to A15	
P4_0 to P4_3	PM06=0	A16 to A19	
	PM06=1	I/O ports	
P4_4	CS0=0	I/O ports	
	CS0=1	$\overline{CS0}$	
P4_5	CS1=0	I/O ports	
	CS1=1	$\overline{CS1}$	
P4_6	CS2=0	I/O ports	
	CS2=1	$\overline{CS2}$	
P4_7	CS3=0	I/O ports	
	CS3=1	$\overline{CS3}$	
P5_0	PM02=0	$\overline{WR}$	
	PM02=1	– (1)	$\overline{WRL}$
P5_1	PM02=0	$\overline{BHE}$	
	PM02=1	– (1)	$\overline{WRH}$
P5_2		$\overline{RD}$	
P5_3		BCLK	
P5_4		$\overline{HLDA}$	
P5_5		$\overline{HOLD}$	
P5_6		ALE	
P5_7		$\overline{RDY}$	

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

NOTES:

1. If the data bus is 8 bits wide, make sure the PM02 bit is set to "0" ($\overline{RD}$, $\overline{BHE}$, $\overline{WR}$).

7.2.9 External Bus Status When Internal Area Accessed

Table 7.6 shows the External Bus Status When Internal Area Accessed.

Table 7.6 External Bus Status When Internal Area Accessed

Item		SFR Accessed	Internal ROM, RAM Accessed
A0 to A19		Address output	Maintain status before accessed address of external area or SFR
D0 to D15	When Read	High-impedance	High-impedance
	When Write	Output data	Undefined
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$ output	Output “H”
$\overline{BHE}$		$\overline{BHE}$ output	Maintain status before accessed status of external area or SFR
$\overline{CS0}$ to $\overline{CS3}$		Output “H”	Output “H”
ALE		Output “L”	Output “L”

7.2.10 Software Wait

Software wait states can be inserted by using the PM17 bit in the PM1 register, the CS0W to CS3W bits in the CSR register, and the CSE register. The SFR area is unaffected by these control bits. This area is always accessed in 2 BCLK or 3 BCLK cycles as determined by the PM20 bit in the PM2 register. See **Table 7.7 Bit and Bus Cycle Related to Software Wait** for details.

To use the $\overline{RDY}$ signal, set the corresponding CS3W to CS0W bit to “0” (with wait state). Table 7.7 shows the Bit and Bus Cycle Related to Software Wait. Figure 7.5 shows the Typical Bus Timings Using Software Wait.

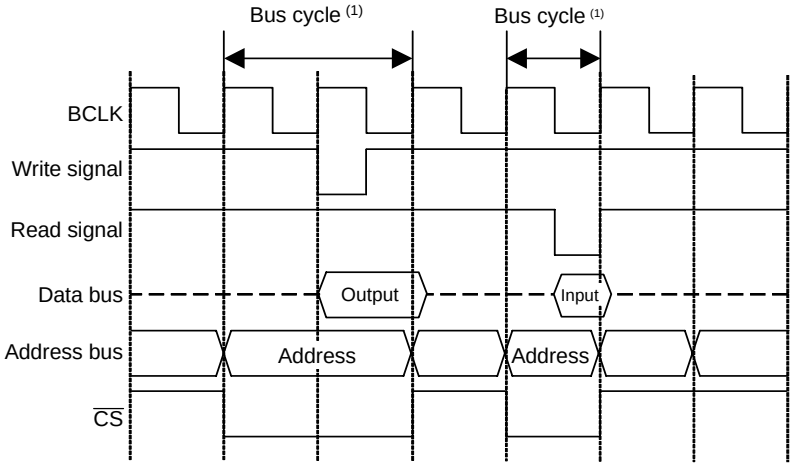
Table 7.7 Bit and Bus Cycle Related to Software Wait

Area	PM1 Register PM17 Bit (3)	CSR Register CS3W Bit (1) CS2W Bit (1) CS1W Bit (1) CS0W Bit (1)	Software Wait	Bus Cycle
Internal RAM, ROM	0	–	No wait	1 BCLK cycle (2)
	1	–	1 wait	2 BCLK cycles
External Area	0	1	No wait	1 BCLK cycle (read)
				2 BCLK cycles (write)
	–	0	1 wait	2 BCLK cycle (2)
	1	0	1 wait	2 BCLK cycle

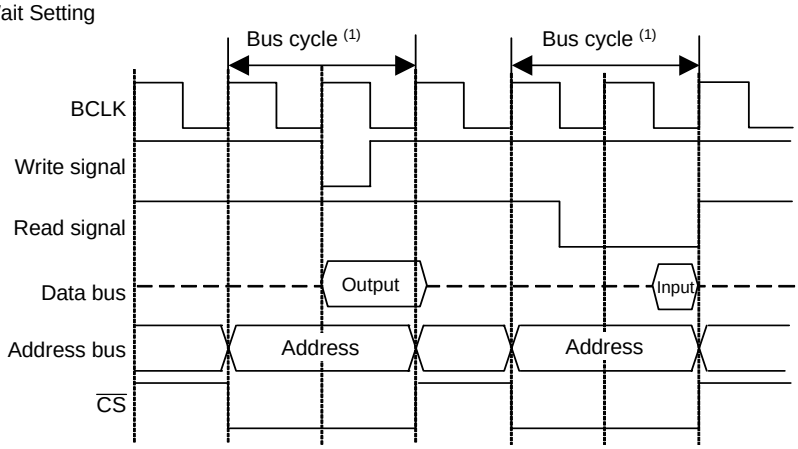
NOTES:

1. To use the $\overline{RDY}$ signal, set this bit to “0” (with wait state).
2. After reset, the PM17 bit is set to “0” (without wait state), all of the CS0W to CS3W bits are set to “0” (with wait state). Therefore, the internal RAM and internal ROM are accessed with no wait states, and all external areas are accessed with one wait state.
3. When PM17 bit is set to “1” and accesses an external area, set the CSiW (i=0 to 3) bits to “0” (with wait state).

(1) Separate Bus, No Wait Setting



(2) Separate Bus, 1-Wait Setting



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

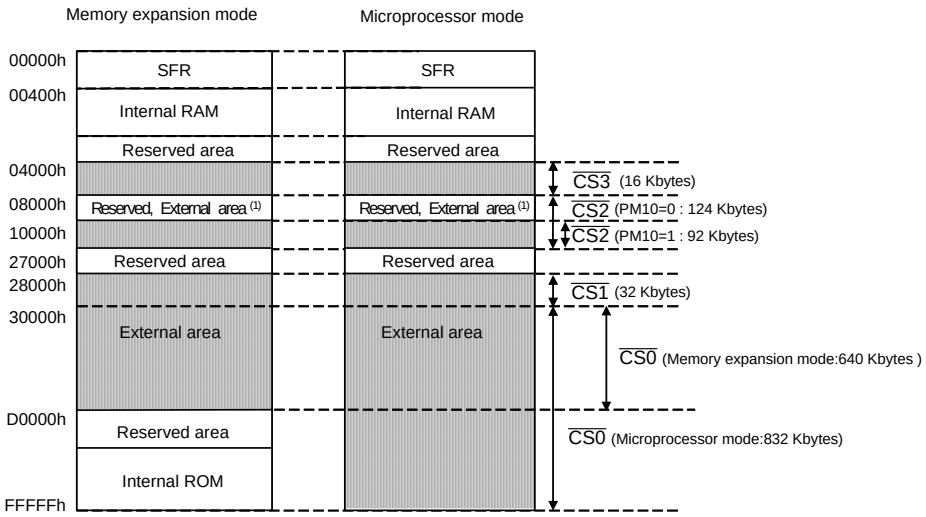
Figure 7.5 Typical Bus Timings Using Software Wait

8. Memory Space Expansion Function

The following describes a memory space extension function.
During memory expansion or microprocessor mode, the memory space expansion function allows the access space to be expanded using the appropriate register bits.

8.1 1-Mbyte Mode

In this mode, the memory space is 1 Mbyte. In 1-Mbyte mode, the external area to be accessed is specified using the $\overline{CSi}$ (i = 0 to 3) signals (hereafter referred to as the $\overline{CSi}$ area). Figure 8.1 and 8.2 show Memory Map and $\overline{CS}$ area. Refer to Figure 8.1 for models without the PM13 bit. See **Figure 6.2 PM1 Register (1)** and **Figure 6.3 PM1 Register (2)** to check presence or absence of the PM13 bit.

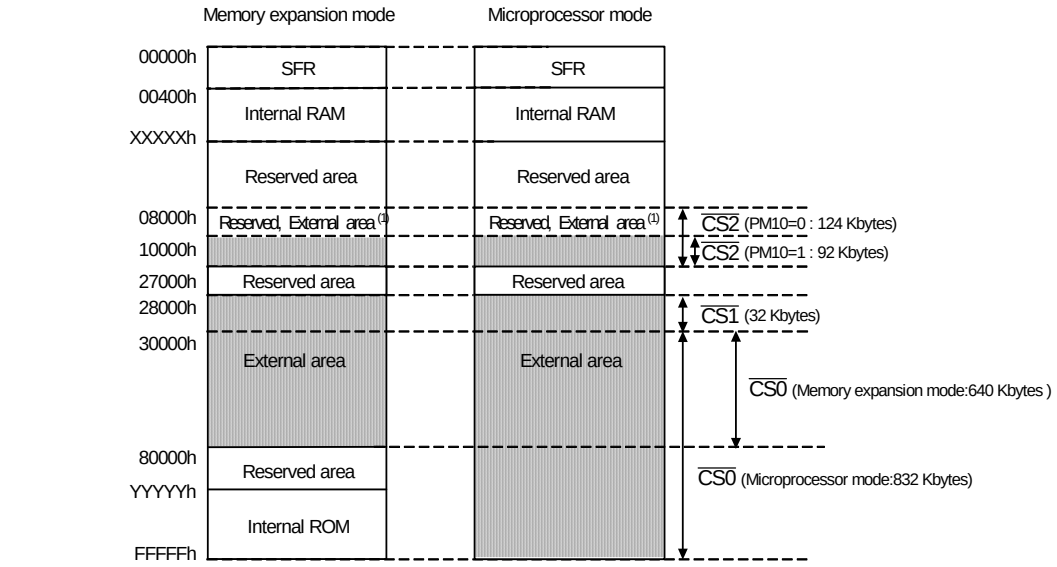


PM13=0

Internal RAM		Internal ROM		External Area			
Size	Address XXXXXh	Size	Address YYYYYh	$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$	$\overline{CS3}$
5 Kbytes	017FFh	96 Kbytes	E8000h	Memory expansion mode 30000h to CFFFFh Microprocessor mode 30000h to FFFFFh	28000h to 2FFFFh	When PM10=0 08000h to 26FFFh When PM10=1 10000h to 26FFFh	04000h to 07FFFh
6 Kbytes	01BFFh	128 Kbytes	E0000h				
12 Kbytes	033FFh	160 Kbytes	D8000h				
		192 Kbytes	D0000h				
		256 Kbytes	D0000h(2)				

NOTES :
1. For flash memory version, when the PM10 bit in the PM1 register is set to “1”, 0F000h to 0FFFFh can be used as internal ROM area. For mask ROM version and one-time flash version, set the PM10 bit to "0" (08000h to 26FFFh for $\overline{CS}$ area).
2. 192-Kbyte internal ROM space is available for use.

Figure 8.1 Memory Mapping and $\overline{CS}$ Area in 1-Mbyte mode (PM13=0)



PM13=1

Internal RAM		Internal ROM		External Area			
Size	Address XXXXXh	Size	Address YYYYYh	CS0	CS1	CS2	CS3
12 Kbytes	033FFh	160 Kbytes	D8000h	Memory expansion mode 30000h to 7FFFFh Microprocessor mode 30000h to FFFFFh	28000h to 2FFFFh	When PM10=0 08000h to 26FFFh When PM10=1 10000h to 26FFFh	No area
		192 Kbytes	D0000h				
		256 Kbytes	C0000h				

NOTE :
1. For flash memory version, when the PM10 bit in the PM1 register is set to “1”, 0F000h to 0FFFFh can be used as internal ROM area.

Figure 8.2 Memory Mapping and CS Area in 1-Mbyte mode (PM13=1)

9. Clock Generating Circuit

9.1 Types of the Clock Generating Circuit

Two circuits are incorporated to generate the system clock signal :

- Main clock oscillation circuit
- Sub clock oscillation circuit

Table 9.1 lists the Clock Generation Circuit Specifications. Figure 9.1 shows the clock generation circuit. Figures 9.2 to 9.4 show the clock-related registers.

Table 9.1 Clock Generation Circuit Specifications

Item	Main Clock Oscillation Circuit	Sub Clock Oscillation Circuit
Use of Clock	• CPU clock source • Peripheral function clock source	• CPU clock source • Timer A, B's clock source
Clock Frequency	0 to 16 MHz	32.768 kHz
Usable Oscillator	• Ceramic oscillator • Crystal oscillator	• Crystal oscillator
Pins to Connect Oscillator	XIN, XOUT	XCIN, XCOUT
Oscillation Stop, Restart Function	Presence	Presence
Oscillator Status After Reset	Oscillating	Stopped
Other	Externally derived clock can be input	

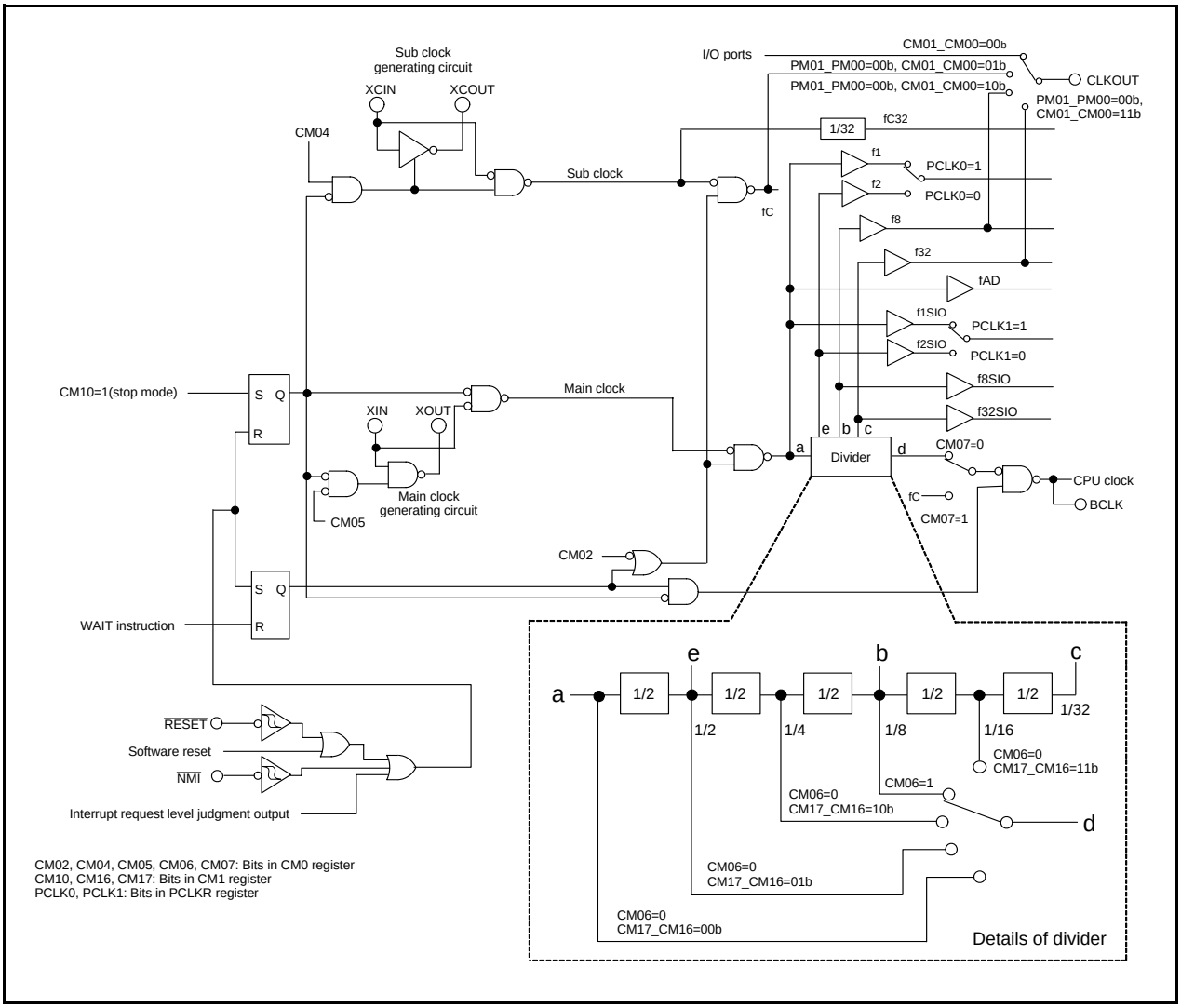
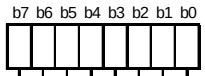


Figure 9.1 Clock Generation Circuit

System Clock Control Register 0 (1)



Symbol	Address	After Reset
CM0	0006h	01001000b

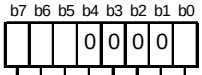
Bit Symbol	Bit Name	Function	RW
CM00	Clock Output Function Select Bit (Valid only in single-chip mode)	b1 b0 0 0 : I/O port P5_7 0 1 : fC output 1 0 : f8 output 1 1 : f32 output	RW
CM01			RW
CM02	WAIT Mode Peripheral Function Clock Stop Bit	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (8)	RW
CM03	XCIN-XCOUT Drive Capacity Select Bit (2)	0 : LOW 1 : HIGH	RW
CM04	Port XC Select Bit (2)	0 : I/O port P8_6, P8_7 1 : XCIN-XCOUT generation function (9)	RW
CM05	Main Clock Stop Bit (3, 10, 11)	0 : On 1 : Off (4, 5)	RW
CM06	Main Clock Division Select Bit 0 (7, 11)	0 : CM16 and CM17 valid 1 : Division by 8 mode	RW
CM07	System Clock Select Bit (6, 10)	0 : Main clock 1 : Sub-clock	RW

NOTES :

1. Write to this register after setting the PRC0 bit in the PRCR register to “1” (write enable).
2. The CM03 bit is set to “1” (high) while the CM04 bit is set to “0,” or when entered to stop mode.
3. This bit is provided to stop the main clock when the low power dissipation mode is selected. This bit cannot be used for detection as to whether the main clock stopped or not. To stop the main clock, set bits in the following order.
   (a) Set the CM07 bit to “1” (Sub-clock select) with the sub-clock stably oscillating.
   (b) Set the CM05 bit to “1” (Stop).
4. During external clock input, Set the CM05 bit to “0” (oscillate).
5. When CM05 bit is set to “1”, the XOUT pin goes “H”. Furthermore, because the internal feedback resistor remains connected, the XIN pin is pulled “H” to the same level as XOUT via the feedback resistor.
6. After setting the CM04 bit to “1” (XCIN-XCOUT oscillator function), wait until the sub-clock oscillates stably before switching the CM07 bit from “0” to “1” (sub-clock).
7. When entering stop mode from high or middle speed mode, the CM06 bit is set to “1” (divide-by-8 mode).
8. The fC32 clock does not stop. During low speed or low power dissipation mode, do not set this bit to “1” (peripheral clock turned off when in wait mode).
9. To use a sub-clock, set this bit to “1”. Also make sure ports P8_6 and P8_7 are directed for input, with no pull-ups.
10. To use the main clock as the clock source for the CPU clock, set bits in the following order.
    (a) Set the CM05 bit to “0” (oscillate).
    (b) Wait the main clock oscillation stabilizes.
    (c) Set the CM07 bit to “0”.
11. When the CM05 bit = 1 (main clock turned off), the CM06 bit is fixed to “1” (divide-by-8 mode) and the CM15 bit is fixed to “1” drive capability High).

Figure 9.2 CM0 Register

System Clock Control Register 1(1)



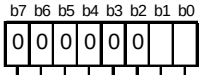
Symbol	Address	After Reset
CM1	0007h	00100000b

Bit Symbol	Bit Name	Function	RW
CM10	All Clock Stop Control Bit (4)	0 : Clock on 1 : All clocks off (stop mode)	RW
— (b4-b1)	Reserved Bit	Set to “0”	RW
CM15	XIN-XOUT Drive Capacity Select Bit(2)	0 : LOW 1 : HIGH	RW
CM16	Main Clock Division Select Bit 1(3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	RW
CM17			RW

NOTES :
1. Write to this register after setting the PRC0 bit in the PRCR register to “1” (write enable).
2. When entering stop mode from high or middle speed mode, or when the CM05 bit is set to “1” (main clock turned off) in low speed mode, the CM15 bit is set to “1” (drive capability high).
3. Effective when the CM06 bit is “0” (CM16 and CM17 bits enable).
4. If the CM10 bit is “1” (stop mode), XOUT goes “H” and the internal feedback resistor is disconnected. The XCIN and XCOUT pins are placed in the high-impedance state.

Figure 9.3 CM1 Register

Peripheral Clock Select Register (1)



Symbol	Address	After Reset
PCLKR	025Eh	00000011b

Bit Symbol	Bit Name	Function	RW
PCLK0	Timers A, B Clock Select Bit (Clock source for the timers A and B)	0 : f2 1 : f1	RW
PCLK1	SI/O Clock Select Bit (Clock source for UART0 to UART2)	0 : f2SIO 1 : f1SIO	RW
— (b7-b2)	Reserved bit	Set to “0”	RW

NOTES :
1. Write to this register after setting the PRC0 bit in the PRCR register to “1” (write enable).

Figure 9.4 PCLKR Register

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

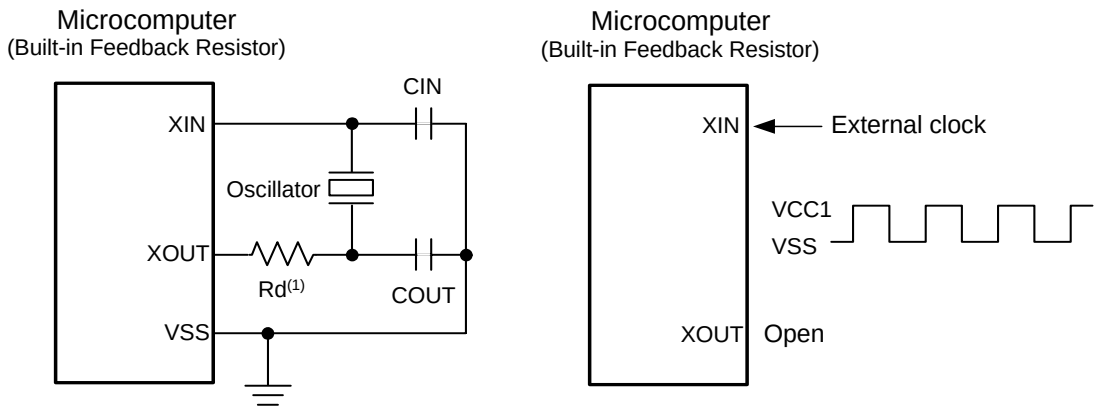
9.1.1 Main Clock

This clock is used as the clock source for the CPU and peripheral function clocks. 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 9.5 shows the examples of main clock connection circuit.

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

The power consumption in the chip can be reduced by setting the CM05 bit in the CM0 register to "1" (main clock oscillator circuit turned off) after switching the clock source for the CPU clock to a sub clock or on-chip oscillator clock. In this case, XOUT goes "H". Furthermore, because the internal feedback resistor remains on, XIN is pulled "H" to XOUT via the feedback resistor. Note that if an externally generated clock is fed into the XIN pin, the main clock cannot be turned off by setting the CM05 bit to "1" unless the sub clock is chosen as a CPU clock. If necessary, use an external circuit to turn off the clock.

During stop mode, all clocks including the main clock are turned off. Refer to **9.4 Power Control**.



NOTES :
1. Place a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by each oscillator the oscillator manufacturer.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, place a feedback resistor between XIN and XOUT if the oscillator manufacturer recommends placing the resistor externally.

Figure 9.5 Examples of Main Clock Connection Circuit

9.1.2 Sub Clock

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

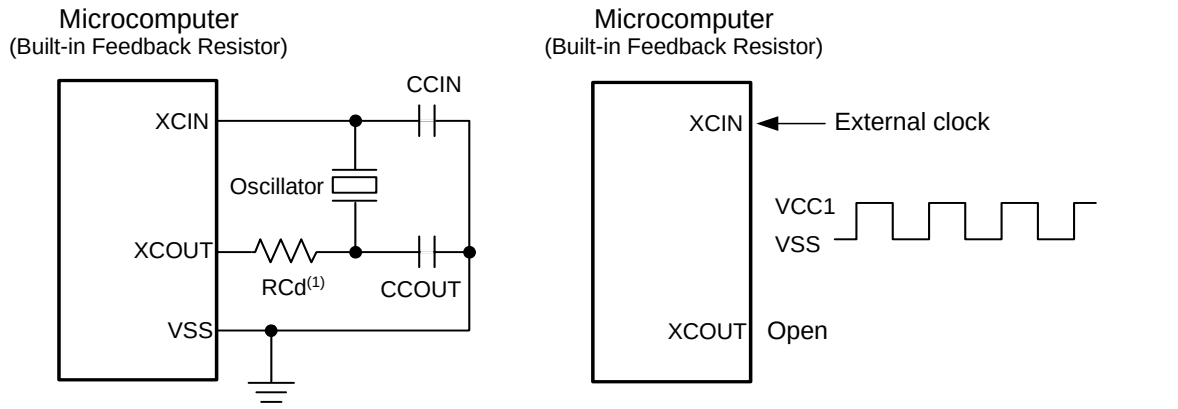
The sub clock oscillator circuit is configured by connecting a crystal resonator between the XCIN and XCOUT pins. The sub clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The sub clock oscillator circuit may also be configured by feeding an externally generated clock to the XCIN pin.

Figure 9.6 shows the examples of sub clock connection circuit.

After reset, the sub clock is turned off. At this time, the feedback resistor is disconnected from the oscillator circuit.

To use the sub clock for the CPU clock, set the CM07 bit in the CM0 register to "1" (sub clock) after the sub clock becomes oscillating stably.

During stop mode, all clocks including the sub clock are turned off. Refer to **9.4 Power Control**.



NOTES :
1. Place a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by each oscillator the oscillator manufacturer.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, place a feedback resistor between XCIN and XCOUT if the oscillator manufacturer recommends placing the resistor externally.

Figure 9.6 Examples of Sub Clock Connection Circuit

9.2 CPU Clock and Peripheral Function Clock

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

9.2.1 CPU Clock and BCLK

These are operating clocks for the CPU and watchdog timer.
The clock source for the CPU clock can be chosen to be the main clock or sub clock.
If the main clock is selected as the clock source for the CPU clock, the selected clock source can be divided by 1 (undivided), 2, 4, 8 or 16 to produce the CPU clock. Use the CM06 bit in CM0 register and the CM17 to CM16 bits in the CM1 register to select the divide-by-n value.
After reset, the main clock divided by 8 provides the CPU clock.
Note that when entering stop mode from high or middle speed mode, or when the CM05 bit in the CM0 register is set to “1” (main clock turned off) in low-speed mode, the CM06 bit in the CM0 register is set to “1” (divide-by-8 mode).

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

These are operating clocks for the peripheral functions.
Of these, fi (i = 1, 2, 8, 32) and fiSIO are derived from the main clock by dividing them by i. The clock fi is used for timers A and B, and fiSIO is used for serial interface. The f8 and f32 clocks can be output from the CLKOUT pin.
The fAD clock is produced from the main clock, and 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), or when the microcomputer is in low power dissipation mode, the fi, fiSIO and fAD clocks are turned off.
The fC32 clock is produced from the sub clock, and is used for timers A and B. This clock can be used when the sub clock is on.

9.3 Clock Output Function

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

9.4 Power Control

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

9.4.1 Normal Operation Mode

Normal operation mode is further classified into 4 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 or sub clock, allow a sufficient wait time in a program until it becomes oscillating stably.

9.4.1.1 High-speed Mode

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

9.4.1.2 Medium-speed Mode

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

9.4.1.3 Low-speed Mode

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

9.4.1.4 Low Power Dissipation Mode

In this mode, the main clock is turned off after being placed in low speed mode. The sub clock provides the CPU clock. The fC32 clock can be used as the count source for timers A and B.
Simultaneously when this mode is selected, the CM06 bit becomes “1” (divided by 8 mode). In the low power dissipation mode, do not change the CM06 bit. Consequently, the medium speed (divided by 8) mode is to be selected when the main clock is operated next.

Table 9.2 Setting Clock Related Bit and Modes

Modes		CM1 Register	CM0 Register			
		CM17, CM16	CM07	CM06	CM05	CM04
High-Speed Mode		00b	0	0	0	–
Medium-Speed Mode	divided by 2	01b	0	0	0	–
	divided by 4	10b	0	0	0	–
	divided by 8	–	0	1	0	–
	divided by 16	11b	0	0	0	–
Low-Speed Mode		–	1	–	0	1
Low Power Dissipation Mode		–	1	1(1)	1(1)	1

– : “0” or ”1”

NOTES:
1. When the CM05 bit is set to “1” (main clock turned off) in low-speed mode, the mode goes to low power dissipation mode and CM06 bit is set to “1” (divided by 8 mode) simultaneously.

9.4.2 Wait Mode

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

9.4.2.1 Peripheral Function Clock Stop Function

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

9.4.2.2 Entering Wait Mode

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

9.4.2.3 Pin Status During Wait Mode

Table 9.3 lists Pin Status During Wait Mode.

Table 9.3 Pin Status During Wait Mode

Pin		Memory Expansion Mode Microprocessor Mode	Single-Chip Mode
A0 to A19, D0 to D15, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$		Retains status before wait mode	Does not become a bus control pin
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		"H"	
$\overline{HLDA}$, BCLK		"H"	
$\overline{ALE}$		"L"	
I/O ports		Retains status before wait mode	Retains status before wait mode
CLKOUT	When fC selected	Does not become a CLKOUT pin	Does not stop
	When f8, f32 selected		Does not stop when the CM02 bit is "0". When the CM02 bit is "1", the status immediately prior to entering wait mode is maintained.

9.4.2.4 Exiting Wait Mode

The microcomputer is moved out of wait mode by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

If the microcomputer is to be moved out of exit wait mode by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority ILVL2 to ILVL0 bits to “000b” (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), 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 9.4 Interrupts to Exit Wait Mode and Use Conditions

Interrupt	CM02=0	CM02=1
$\overline{\text{NMI}}$ Interrupt	Can be used	Can be used
Serial Interface Interrupt	Can be used when operating with internal or external clock	Can be used when operating with external clock
Key Input Interrupt	Can be used	Can be used
A/D Conversion Interrupt	Can be used in one-shot mode	–(Do not use)
Timer A Interrupt Timer B Interrupt	Can be used in all modes	Can be used in event counter mode or when the count source is fC32
$\overline{\text{INT}}$ Interrupt	Can be used	Can be used

Table 9.4 lists the Interrupts to Exit Wait Mode and Use Conditions.

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

(1) Set the ILVL2 to ILVL0 bits in the interrupt control register, for peripheral function interrupts used to exit wait mode.
The ILVL2 to ILVL0 bits in all other interrupt control registers, for peripheral function interrupts not used to exit wait mode, are set to “000b” (interrupt disable).
(2) Set the I flag to “1”.
(3) Start operating the peripheral functions used to exit wait mode.
When the peripheral function interrupt is used, an interrupt routine is performed as soon as an interrupt request is acknowledged and the CPU clock is supplied again.

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

9.4.3 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 VCC1 and VCC2 pins is VRAM or more, the internal RAM is retained. When applying 2.7 or less voltage to VCC1 and VCC2 pins, make sure VCC1 = VCC2 ≥ VRAM.

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

Table 9.5 Interrupts to Exit Stop Mode and Use Conditions

Interrupt	CM02=1
$\overline{\text{NMI}}$ Interrupt	Can be used
Key Input Interrupt	Can be used
$\overline{\text{INT}}$ Interrupt	Can be used
Timer A Interrupt Timer B Interrupt	Can be used when counting external pulses in event counter mode
Serial Interface Interrupt	Can be used when operating with external clock

9.4.3.1 Entering Stop Mode

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

9.4.3.2 Pin Status in Stop Mode

Table 9.6 lists Pin Status in Stop Mode.

Table 9.6 Pin Status in Stop Mode

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

9.4.3.3 Exiting Stop Mode

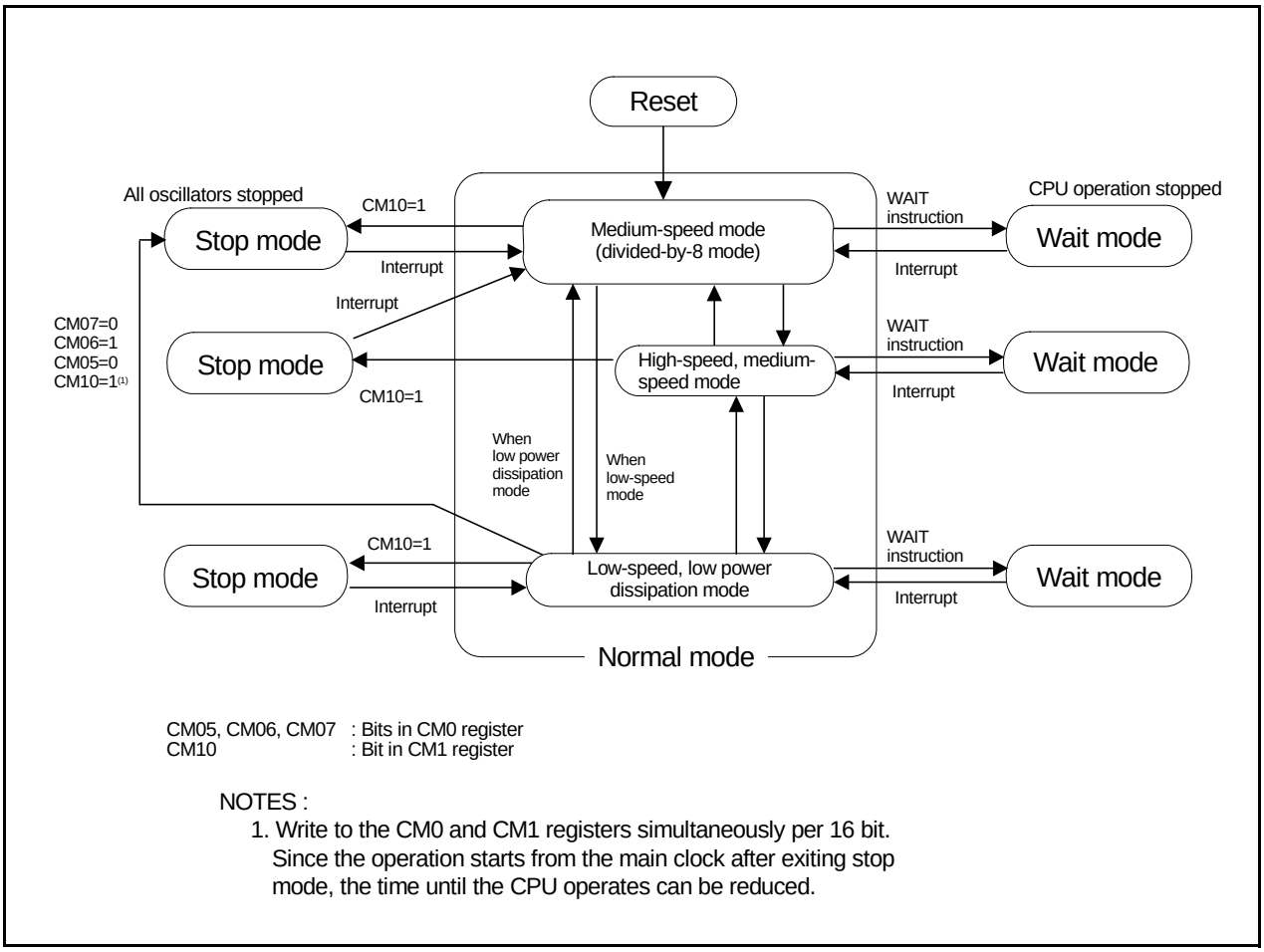
Stop mode is exited by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.
When the hardware reset or $\overline{\text{NMI}}$ interrupt is used to exit wait mode, set all ILVL2 to ILVL0 bits in the interrupt control registers for the peripheral function interrupt to “000b” (interrupt disabled) before setting the CM10 bit to “1”.
When the peripheral function interrupt is used to exit stop mode, set the CM10 bit to “1” after the following settings are completed.

(1) Set the ILVL2 to ILVL0 bits in the interrupt control registers to decide the peripheral priority level of the peripheral function interrupt.
Set the interrupt priority levels of the interrupts, not being used to exit stop mode, to “0” by setting the all ILVL2 to ILVL0 bits to “000b”
(2) Set the I flag to “1”.
(3) Start operation of peripheral function being used to exit wait mode.
When exiting stop mode by the peripheral function interrupt, the interrupt routine is performed when an interrupt request is generated and the CPU clock is supplied again.

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

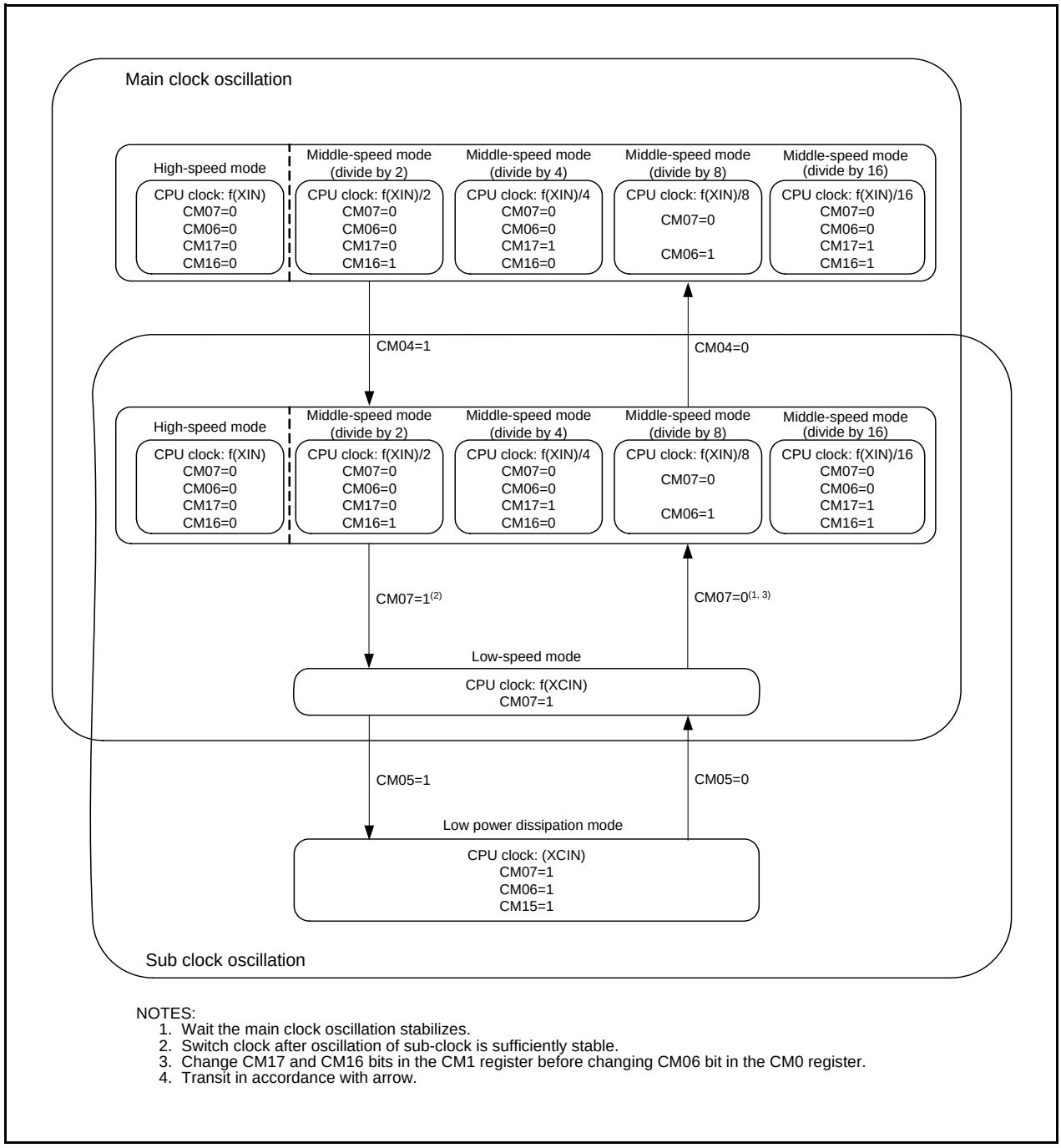
- When the sub clock is the CPU clock before entering stop mode : Sub clock
- When the main clock is the CPU clock source before entering stop mode : Main clock divided by 8

Figure 9.7 shows the state transition from normal operation mode to stop mode and wait mode. Figure 9.8 shows the State Transition in Normal Operation Mode.
Table 9.7 shows a state transition matrix describing Allowed Transition and Setting. The vertical line shows current state and horizontal line shows state after transition.



NOTES :
1. Write to the CM0 and CM1 registers simultaneously per 16 bit. Since the operation starts from the main clock after exiting stop mode, the time until the CPU operates can be reduced.

Figure 9.7 State Transition to Stop Mode and Wait Mode



NOTES:
1. Wait the main clock oscillation stabilizes.
2. Switch clock after oscillation of sub-clock is sufficiently stable.
3. Change CM17 and CM16 bits in the CM1 register before changing CM06 bit in the CM0 register.
4. Transit in accordance with arrow.

Figure 9.8 State Transition in Normal Operation Mode

Table 9.7 Allowed Transition and Setting

		State After Transition				
		High-Speed Mode, Middle-Speed Mode	Low-Speed Mode	Low Power Dissipation Mode	Stop Mode	Wait Mode
Current State	High-Speed Mode, Middle-Speed Mode	(NOTE 4)	(9)(NOTE 3)	–	(12)	(13)
	Low-Speed Mode	(8)		(11)(NOTE 2)	(12)	(13)
	Low Power Dissipation Mode	–	(10)		(12)	(13)
	Stop Mode	(14)(NOTE 1)	(14)	(14)		–
	Wait Mode	(14)	(14)	(14)	–	

–: Cannot transit

NOTES:
1. When exiting stop mode, the CM06 bit is set to “1” (division by 8 mode).
2. If the CM05 bit set to “1” (main clock stop), then the CM06 bit is set to “1” (division by 8 mode).
3. A transition can be made only when sub clock is oscillating.
4. State transitions within the same mode (divide-by-n values changed or subclock oscillation turned on or off) are shown in the table below.

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

–: Cannot transit

5. ( ) : setting method. See the following table.

	Setting	Operation
(1)	CM04 = 0	Sub clock turned off
(2)	CM04 = 1	Sub clock oscillating
(3)	CM06 = 0, CM17 = 0, CM16 = 0	CPU clock no division mode
(4)	CM06 = 0, CM17 = 0, CM16 = 1	CPU clock division by 2 mode
(5)	CM06 = 0, CM17 = 1, CM16 = 0	CPU clock division by 4 mode
(6)	CM06 = 0, CM17 = 1, CM16 = 1	CPU clock division by 16 mode
(7)	CM06 = 1	CPU clock division by 8 mode
(8)	CM07 = 0	Main clock selected
(9)	CM07 = 1	Sub clock selected
(10)	CM05 = 0	Main clock oscillating
(11)	CM05 = 1	Main clock turned off
(12)	CM10 = 1	Transition to stop mode
(13)	Wait Instruction	Transition to wait mode
(14)	Hardware Interrupt	Exit stop mode or wait mode

CM04, CM05, CM06, CM07 : Bits in CM0 register
CM10, CM16, CM17 : Bits in CM1 register

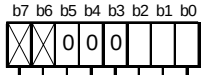
10. Protection

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 10.1 shows the PRCR Register. The following lists the registers protected by the PRCR register.

- The PRC0 bit protects the CM0, CM1 and PCLKR registers;
- The PRC1 bit protects the PM0 and PM1 registers;
- The PRC2 bit protects the PD9 register;

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

Protect Register



Symbol	Address	After Reset
PRCR	000Ah	XX000000b

Bit Symbol	Bit Name	Function	RW
PRC0	Protect Bit 0	Enable write to CM0, CM1 and PCLKR registers 0 : Write protected 1 : Write enabled	RW
PRC1	Protect Bit 1	Enable write to PM0 and PM1 registers 0 : Write protected 1 : Write enabled (1)	RW
PRC2	Protect Bit 2	Enable write to PD9 register 0 : Write protected 1 : Write enabled (1)	RW
— (b5-b3)	Reserved Bit	Set to "0".	RW
— (b7-b6)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

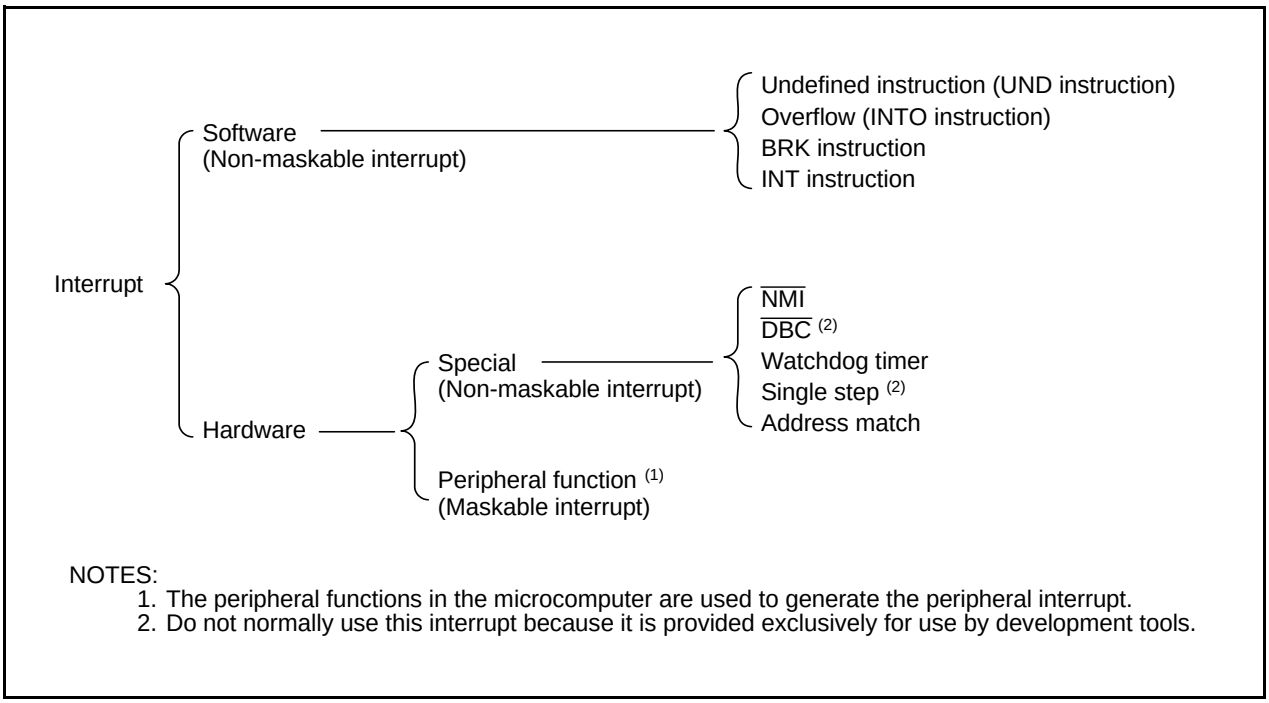
NOTES :
1. The PRC2 bit is set to "0" by writing to any address after setting it to "1". Other bits are not set to "0" by writing to any address, and must therefore be set in a program.

Figure 10.1 PRCR Register

11. Interrupt

11.1 Type of Interrupts

Figure 11.1 shows Type of Interrupts.



NOTES:
1. The peripheral functions in the microcomputer are used to generate the peripheral interrupt.
2. Do not normally use this interrupt because it is provided exclusively for use by development tools.

Figure 11.1 Type of Interrupts

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

11.2 Software Interrupts

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

11.2.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

11.2.2 Overflow Interrupt

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

11.2.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

11.2.4 INT Instruction Interrupt

An INT instruction interrupt occurs when executing the INT instruction. Software interrupt Nos. 0 to 63 can be specified for the INT instruction. Because software interrupt Nos. 4 to 31 are assigned to peripheral function interrupts, the same interrupt routine as for peripheral function interrupts can be executed by executing the INT instruction.
In software interrupt Nos. 0 to 31, the U flag is saved to the stack during instruction execution and is cleared to “0” (ISP selected) before executing an interrupt sequence. The U flag is restored from the stack when returning from the interrupt routine. In software interrupt Nos. 32 to 63, the U flag does not change state during instruction execution, and the SP then selected is used.

11.3 Hardware Interrupts

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

11.3.1 Special Interrupts

Special interrupts are non-maskable interrupts.

11.3.1.1 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. For details about the $\overline{\text{NMI}}$ interrupt, refer to the **11.7 NMI Interrupt**.

11.3.1.2 $\overline{\text{DBC}}$ Interrupt

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

11.3.1.3 Watchdog Timer Interrupt

Generated by the watchdog timer. Once a watchdog timer interrupt is generated, be sure to initialize the watchdog timer. For details about the watchdog timer, refer to the **12. Watchdog Timer**.

11.3.1.4 Single-Step Interrupt

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

11.3.1.5 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 AIER0 or AIER1 bit in the AIER register which is “1” (address match interrupt enabled). For details about the address match interrupt, refer to the **11.9 Address Match Interrupt**.

11.3.2 Peripheral Function Interrupts

The peripheral function interrupt occurs when a request from the peripheral functions in the microcomputer is acknowledged. The peripheral function interrupt is a maskable interrupt. See **Table 11.2 Relocatable Vector Tables** about how the peripheral function interrupt occurs. Refer to the descriptions of each function for details.

11.4 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 11.2 shows the Interrupt Vector.

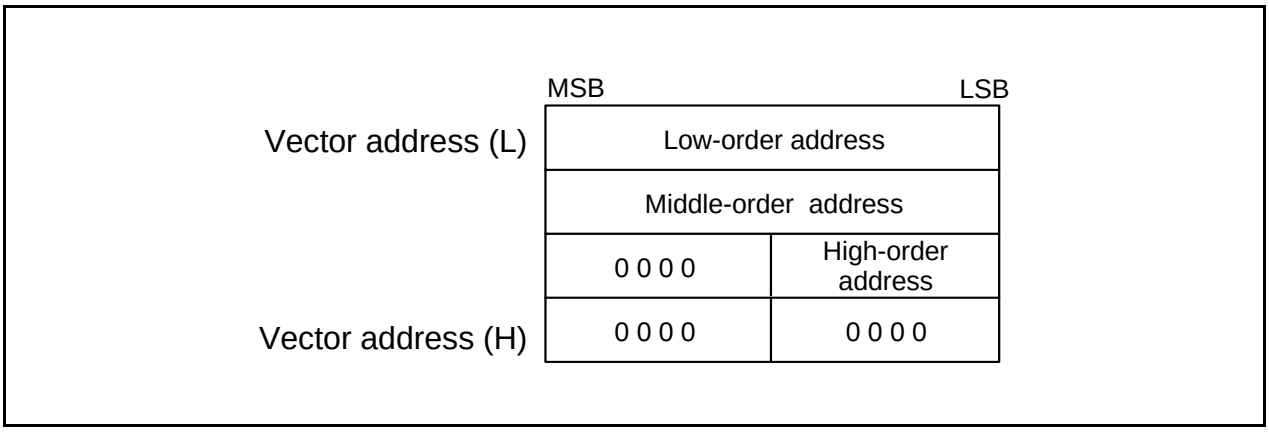


Figure 11.2 Interrupt Vector

11.4.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDCh to FFFFFh. Table 11.1 lists the Fixed Vector Tables. In the one time flash memory and 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 the 19.2 Functions To Prevent Flash Memory from Rewriting .

Table 11.1 Fixed Vector Tables

Interrupt Source	Vector Table Addresses Address (L) to Address (H)	Reference
Undefined Instruction (UND instruction)	FFFDCh to FFFDFh	M16C/60, M16C/20 Series software manual
Overflow (INTO instruction)	FFFE0h to FFFE3h	
BRK Instruction (2)	FFFE4h to FFFE7h	
Address Match	FFFE8h to FFFEBh	11.9 Address Match Interrupt
Single Step (1)	FFFECh to FFFEFh	
Watchdog Timer	FFFF0h to FFFF3h	12. Watchdog Timer
$\overline{\text{DBC}}$ (1)	FFFF4h to FFFF7h	
$\overline{\text{NMI}}$	FFFF8h to FFFFBh	11.7 $\overline{\text{NMI}}$ interrupt
Reset	FFFFCh to FFFFFh	5. Reset

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

11.4.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a reloacatable vector table area. Table 11.2 lists the Relocatable Vector Tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than in the case of odd addresses.

Table 11.2 Relocatable Vector Tables

Interrupt Source	Vector Address (1) Address (L) to Address (H)	Software Interrupt Number	Reference
BRK Instruction (5)	+0 to +3 (0000h to 0003h)	0	M16C/60, M16C/20 Series software manual
–(Reserved)	–	1 to 3	
$\overline{\text{INT3}}$	+16 to +19 (0010h to 0013h)	4	11.6 $\overline{\text{INT}}$ interrupt
–	–	5	–
UART1 Bus Collision Detect (4, 6)	+24 to +27 (0018h to 001Bh)	6	15. Serial Interface
UART0 Bus Collision Detect (4, 6)	+28 to +31 (001Ch to 001Fh)	7	
–	–	8	–
$\overline{\text{INT4}}$ (2)	+36 to +39 (0024h to 0027h)	9	11.6 $\overline{\text{INT}}$ interrupt
UART 2 Bus Collision Detection (6)	+40 to +43 (0028h to 002Bh)	10	15. Serial Interface
DMA0	+44 to +47 (002Ch to 002Fh)	11	13. DMAC
DMA1	+48 to +51 (0030h to 0033h)	12	
Key Input Interrupt	+52 to +55 (0034h to 0037h)	13	11.8 Key Input Interrupt
A/D	+56 to +59 (0038h to 003Bh)	14	16. A/D Converter
UART2 Transmit, NACK2 (3)	+60 to +63 (003Ch to 003Fh)	15	15. Serial Interface
UART2 Receive, ACK2 (3)	+64 to +67 (0040h to 0043h)	16	
UART0 Transmit, NACK0 (3)	+68 to +71 (0044h to 0047h)	17	
UART0 Receive, ACK0 (3)	+72 to +75 (0048h to 004Bh)	18	
UART1 Transmit, NACK1 (3)	+76 to +79 (004Ch to 004Fh)	19	
UART1 Receive, ACK1 (3)	+80 to +83 (0050h to 0053h)	20	
Timer A0	+84 to +87 (0054h to 0057h)	21	14. Timers
Timer A1	+88 to +91 (0058h to 005Bh)	22	
Timer A2	+92 to +95 (005Ch to 005Fh)	23	
–	–	24	–
–	–	25	–
Timer B0	+104 to +107 (0068h to 006Bh)	26	14. Timers
Timer B1	+108 to +111 (006Ch to 006Fh)	27	
Timer B2	+112 to +115 (0070h to 0073h)	28	
$\overline{\text{INT0}}$	+116 to +119 (0074h to 0077h)	29	11.6 $\overline{\text{INT}}$ interrupt
$\overline{\text{INT1}}$	+120 to +123 (0078h to 007Bh)	30	
$\overline{\text{INT2}}$	+124 to +127 (007Ch to 007Fh)	31	
Software Interrupt (5)	+128 to +131 (0080h to 0083h) to +252 to +255 (00FCh to 00FFh)	32 to 63	M16C/60, M16C/20 Series software manual

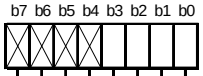
NOTES:
1. Address relative to address in INTB.
2. Use the IFSR6 bit in the IFSR register to select.
3. During I^2C mode, NACK and ACK interrupts comprise the interrupt source.
4. Use the IFSR26 and IFSR27 bits in the IFSR2A register to select.
5. These interrupts cannot be disabled using the I flag.
6. Bus collision detection : During IE mode, this bus collision detection constitutes the factor of an interrupt. During I^2C mode, however, a start condition or a stop condition detection constitutes the factor of an interrupt.

11.5 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 I flag in the FLG register, IPL, and ILVL2 to ILVL0 bits in the each interrupt control register to enable/ disable the maskable interrupts. Whether an interrupt is requested is indicated by the IR bit in the each interrupt control register.
Figure 11.3 and 11.4 shows the Interrupt Control Registers.

Interrupt Control Register (2)

Symbol	Address	After Reset
U1BCNIC(3)	0046h	XXXXX000b
U0BCNIC(3)	0047h	XXXXX000b
BCNIC	004Ah	XXXXX000b
DM0IC, DM1IC	004Bh, 004Ch	XXXXX000b
KUPIC	004Dh	XXXXX000b
ADIC	004Eh	XXXXX000b
S0TIC to S2TIC	0051h, 0053h, 004Fh	XXXXX000b
S0RIC to S2RIC	0052h, 0054h, 0050h	XXXXX000b
TA0IC to TA2IC	0055h to 0057h	XXXXX000b
TB0IC to TB2IC	005Ah to 005Ch	XXXXX000b

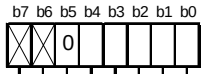


Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt Priority Level Select Bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt Request Bit	0 : Interrupt not requested 1 : Interrupt requested	RW(1)
— (b7-b4)	Nothing is assigned.When write, set to “0”. When read, their contents are indeterminate.		—

NOTES :
1. This bit can only be reset by writing “0” (Do not write “1”).
2. To rewrite the interrupt control registers, do so at a point that does not generate the interrupt request for that register. For details, refer to **21.5 Precautions for Interrupt**.
3. Use the IFSR2A register to select.

Figure 11.3 Interrupt Control Registers (1)

INTi (0 to 4) Interrupt Control Register (2)



Symbol	Address	After Reset
INT3IC (4)	0044h	XX00X000b
INT4IC (4)	0049h	XX00X000b
INT0IC to INT2IC	005Dh to 005Fh	XX00X000b

Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt Priority Level Select Bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt Request Bit	0: Interrupt not requested 1: Interrupt requested	RW(1)
POL	Polarity Select Bit	0 : Selects falling edge (3) 1 : Selects rising edge	RW
— (b5)	Reserved Bit	Set to “0”.	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

NOTES :
1. This bit can only be reset by writing “0” (Do not write “1”).
2. To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. For details, refer to **21.5 Precautions for Interrupt**.
3. If the IFSRi bit (i = 0 to 4) in the IFSR register are “1” (both edges), set the POL bit in the INTiIC register to “0” (falling edge).
4. When the BYTE pin is low and the processor mode is memory expansion or microprocessor mode, set the ILVL2 to ILVL0 bits in the INT4IC to INT3IC registers to “000b” (interrupts disabled).

Figure 11.4 Interrupt Control Registers (2)

11.5.1 I Flag

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

11.5.2 IR Bit

The IR bit is set to “1” (= interrupt requested) when an interrupt request is generated. Then, when the interrupt request is accepted and the CPU branches to the corresponding interrupt vector, the IR bit is 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.

11.5.3 ILVL2 to ILVL0 Bits and IPL

Interrupt priority levels can be set using the ILVL2 to ILVL0 bits.
Table 11.3 shows the Settings of Interrupt Priority Levels and Table 11.4 shows the Interrupt Priority Levels Enabled by 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 11.3 Settings of Interrupt Priority Levels

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

Table 11.4 Interrupt Priority Levels Enabled by IPL

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

11.5.4 Interrupt Sequence

An interrupt sequence – what are performed over a period from the instant an interrupt is accepted to the instant the interrupt routine is executed – is described here.
If an interrupt 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 11.5 shows Time Required for Executing Interrupt Sequence.

(1) The CPU obtains interrupt information (interrupt number and interrupt request level) by reading address 000000h. Then, the IR bit applicable to the interrupt information is set to “0” (interrupt requested).
(2) The FLG register, prior to an interrupt sequence, is saved to a temporary register (1) within the CPU.
(3) The I, D and U flags in the FLG register become as follows:
- The I flag is set to “0” (interrupt disabled)
- The D flag is set to “0” (single-step interrupt disabled)
- The U flag is set to “0” (ISP selected)
  However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.

(4) The temporary register (1) within the CPU is saved to the stack.
(5) The PC is saved to the stack.
(6) The interrupt priority level of the acknowledged interrupt in IPL is set.
(7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

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

NOTES:
1.Temporary register cannot be modified by users.

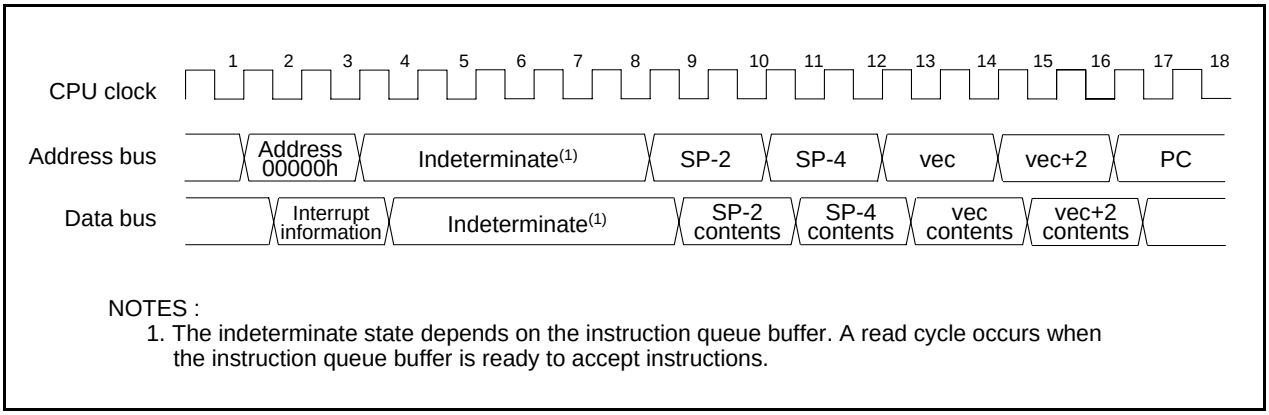
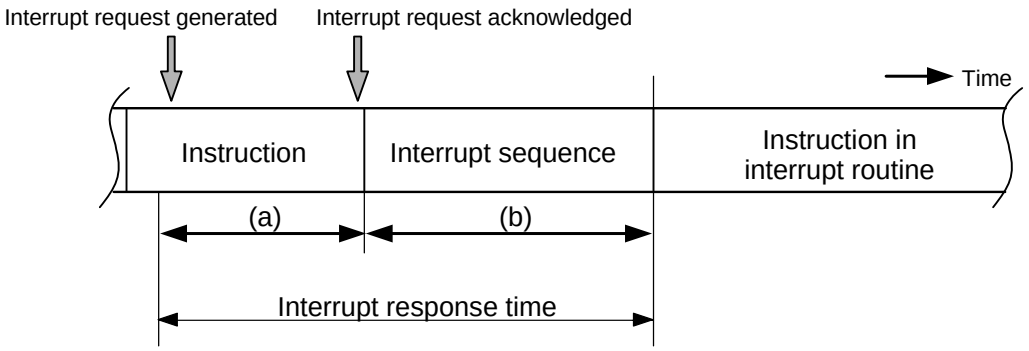


Figure 11.5 Time Required for Executing Interrupt Sequence

11.5.5 Interrupt Response Time

Figure 11.6 shows the Interrupt Response Time. The interrupt response or interrupt acknowledge time denotes a time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of a time from when an interrupt request is generated till when the instruction then executing is completed ((a) on Figure 11.6) and a time during which the interrupt sequence is executed ((b) on Figure 11.6).



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

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

Interrupt Vector Address	SP Value	16-Bit Bus, Without Wait	8-Bit Bus, Without Wait
Even	Even	18 cycles	20 cycles
Even	Odd	19 cycles	20 cycles
Odd	Even	19 cycles	20 cycles
Odd	Odd	20 cycles	20 cycles

Figure 11.6 Interrupt Response Time

11.5.6 Variation of IPL when Interrupt Request is Accepted

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

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in Table 11.5 is set in the IPL. Table 11.5 lists the IPL Level That is Set to IPL When a Software or Special Interrupt is Accepted.

Table 11.5 IPL Level That is Set to IPL When a Software or Special Interrupt is Accepted

Interrupt Sources	Level that is Set to IPL
Watchdog Timer, $\overline{\text{NMI}}$	7
Software, Address Match, $\overline{\text{DBC}}$, Single-Step	Not changed

11.5.7 Saving Registers

In the interrupt sequence, the FLG register and PC are saved to the stack.
At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits in the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits of the PC are saved.
Figure 11.7 shows the Stack Status Before and After Acceptance of Interrupt Request.
The other necessary registers must be saved in a program at the beginning of the interrupt routine. Use the PUSHM instruction, and all registers except SP can be saved with a single instruction.

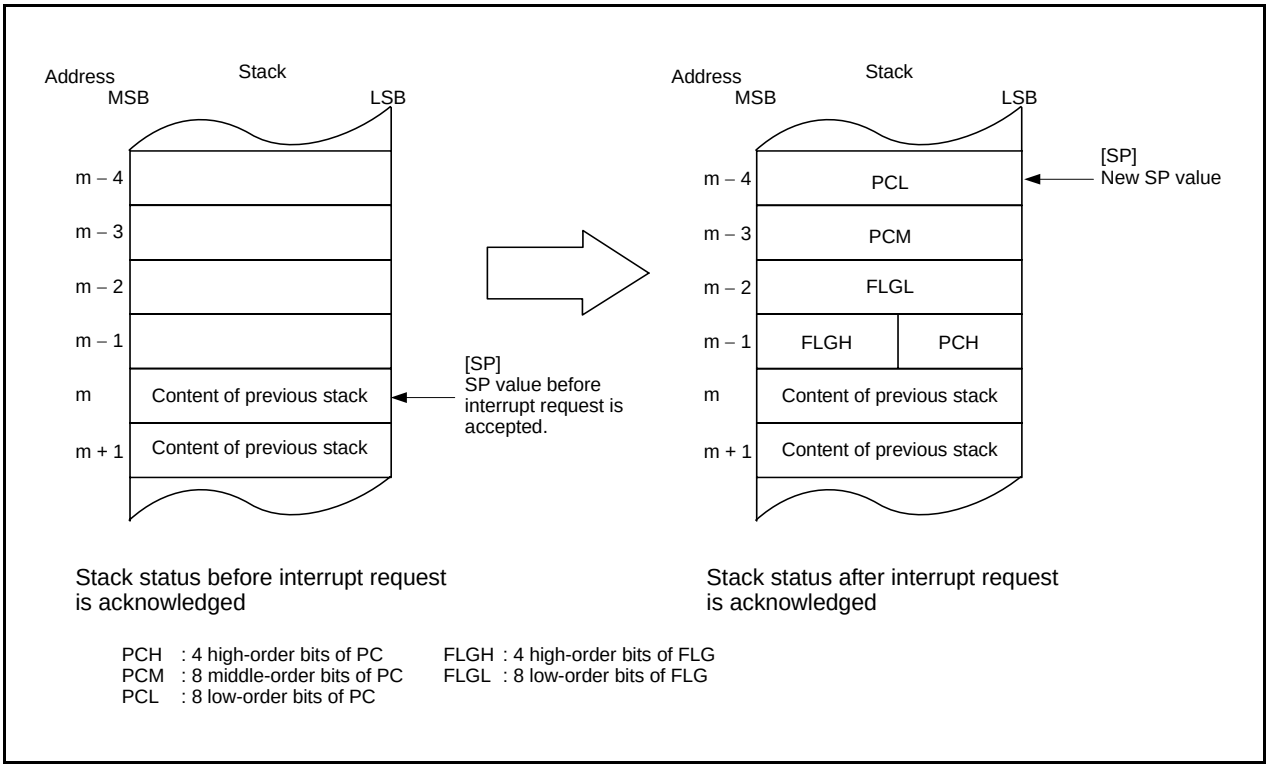


Figure 11.7 Stack Status Before and After Acceptance of Interrupt Request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP (1), at the time of acceptance of an interrupt request, is even or odd. If the stack pointer (1) is even, the FLG register and the PC are saved,16 bits at a time. If odd, they are saved in two steps, 8 bits at a time. Figure 11.8 shows the Operation of Saving Register.

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.

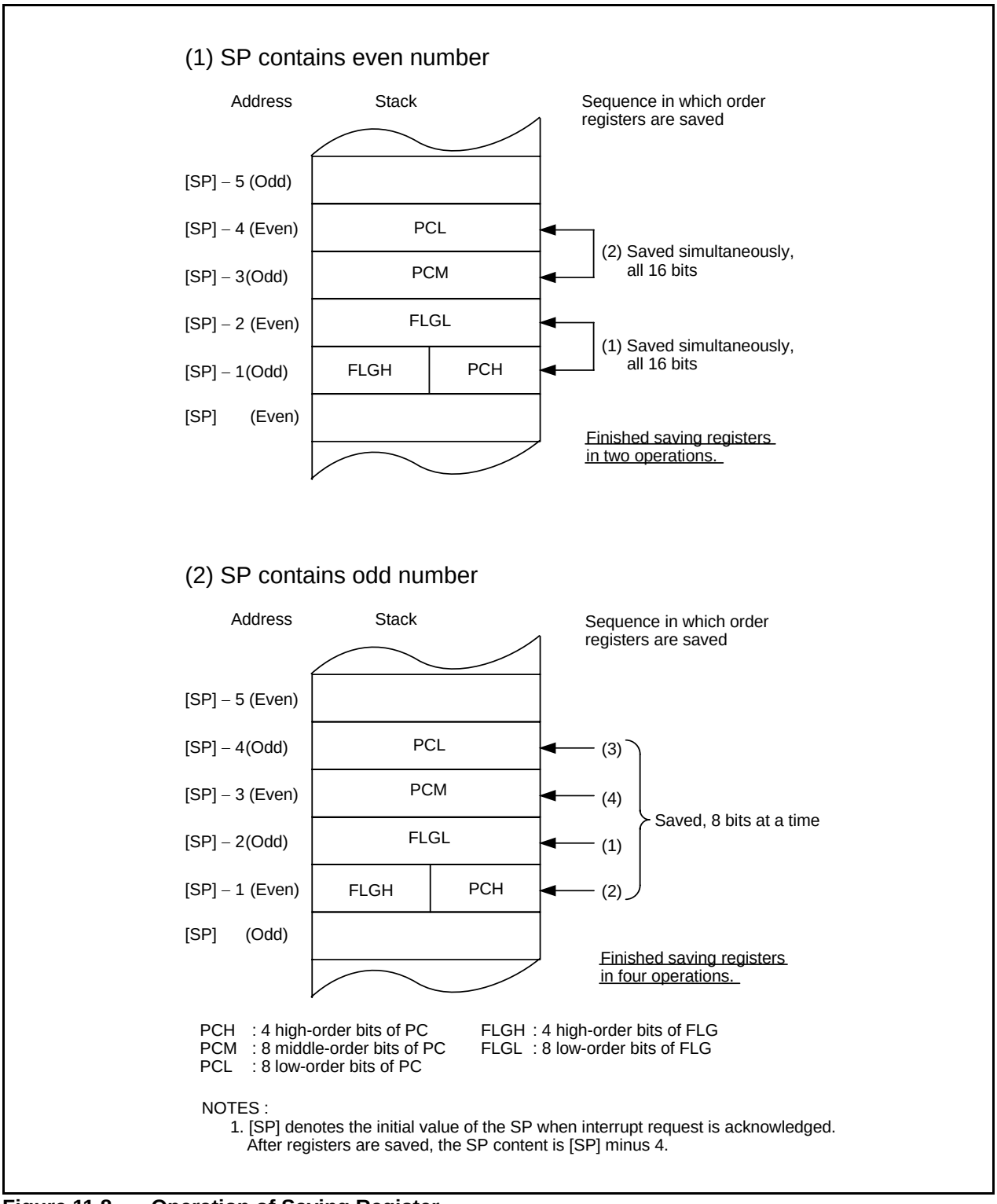


Figure 11.8 Operation of Saving Register

11.5.8 Returning from an Interrupt Routine

The FLG register and PC in the state in which they were immediately before entering the interrupt sequence are restored from the stack by executing the REIT instruction at the end of the interrupt routine.
Thereafter the CPU returns to the program which was being executed before accepting the interrupt request.
Return the other registers saved by a program within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

11.5.9 Interrupt Priority

If two or more interrupt requests are generated while executing one instruction, the interrupt request that has the highest priority is accepted.
For maskable interrupts (peripheral functions), any desired priority level can be selected using the ILVL2 to ILVL0 bits. However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, with the highest priority interrupt accepted.
The watchdog timer and other special interrupts have their priority levels set in hardware. Figure 11.9 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.

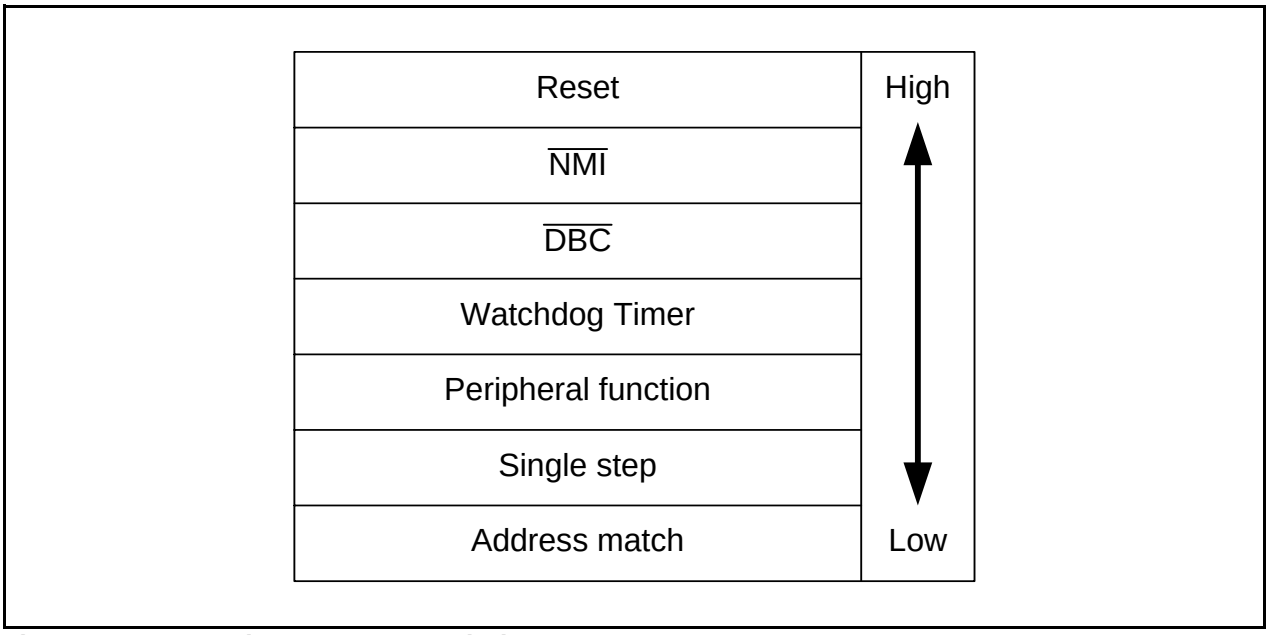


Figure 11.9 Hardware Interrupt Priority

11.5.10 Interrupt Priority Resolution Circuit

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

Figure 11.10 shows the Interrupts Priority Select Circuit.

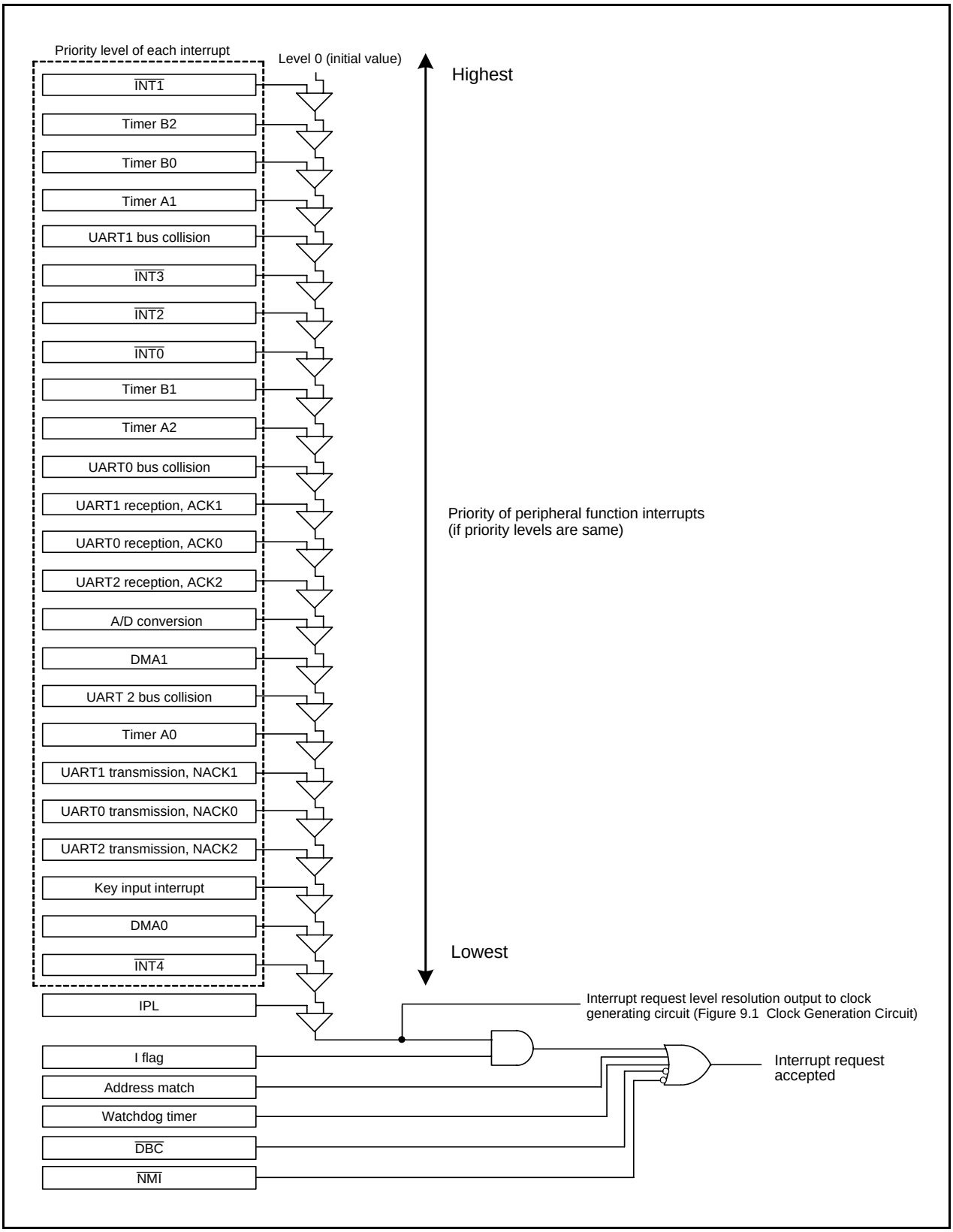
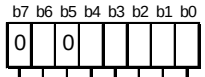


Figure 11.10 Interrupts Priority Select Circuit

11.6 $\overline{\text{INT}}$ Interrupt

$\overline{\text{INTi}}$ interrupt (i=0 to 4) is triggered by the edges of external inputs. The edge polarity is selected using the IFSRi bit in the IFSR register.
To use the $\overline{\text{INT4}}$ interrupt, set the IFSR6 bit in the IFSR register to "1" (= $\overline{\text{INT4}}$).
After modifying the IFSR6 bit, clear the corresponding IR bit to "0" (= interrupt not requested) before enabling the interrupt.
Figure 11.11 shows the IFSR and IFSR2A Registers.

Interrupt Factor Select Register

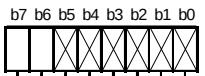


Symbol	Address	After Reset
IFSR	035Fh	00h

Bit Symbol	Bit Name	Function	RW
IFSR0	INT0 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges (1)	RW
IFSR1	INT1 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges (1)	RW
IFSR2	INT2 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges (1)	RW
IFSR3	INT3 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges (1)	RW
IFSR4	INT4 Interrupt Polarity Switching Bit	0 : One edge 1 : Both edges (1)	RW
— (b5)	Reserved Bit	Set to "0".	RW
IFSR6	Interrupt Request Factor Select Bit	0 : Do not set 1 : $\overline{\text{INT4}}$	RW
— (b7)	Reserved Bit	Set to "0".	RW

NOTES :
1. When setting this bit to "1" (= both edges), make sure the POL bit in the INT0IC to INT4IC register are set to "0" (= falling edge).

Interrupt Factor Select Register 2



Symbol	Address	After Reset
IFSR2A	035Eh	00XXXXXXb

Bit Symbol	Bit Name	Function	RW
— (b5-b0)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—
IFSR26	Interrupt Request Factor Select Bit (1)	0 : Do not set 1 : UART0 bus collision detection	RW
IFSR27	Interrupt Request Factor Select Bit (2)	0 : Do not set 1 : UART1 bus collision detection	RW

NOTES :
1. When using UART0 bus collision detection, set the IFSR26 bit to "1".
2. When using UART1 bus collision detection, set the IFSR27 bit to "1".

Figure 11.11 IFSR and IFSR2A Registers

11.7 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt request is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. The $\overline{\text{NMI}}$ interrupt is a non-maskable interrupt.
The input level of this $\overline{\text{NMI}}$ interrupt input pin can be read by accessing the P8_5 bit in the P8 register.
This pin cannot be used as an input port.

11.8 Key Input Interrupt

Of P10_4 to P10_7, a key input interrupt is generated when input on any of the P10_4 to P10_7 pins which has had the PD10_4 to PD10_7 bits in the PD10 register set to "0" (= input) goes low. Key input interrupts can be used as a key-on wake up function, the function which gets the microcomputer out of wait or stop mode. However, if you intend to use the key input interrupt, do not use P10_4 to P10_7 as analog input ports. Figure 11.12 shows the block diagram of the Key Input Interrupt. Note, however, that while input on any pin which has had the PD10_4 to PD10_7 bits set to "0" (= input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

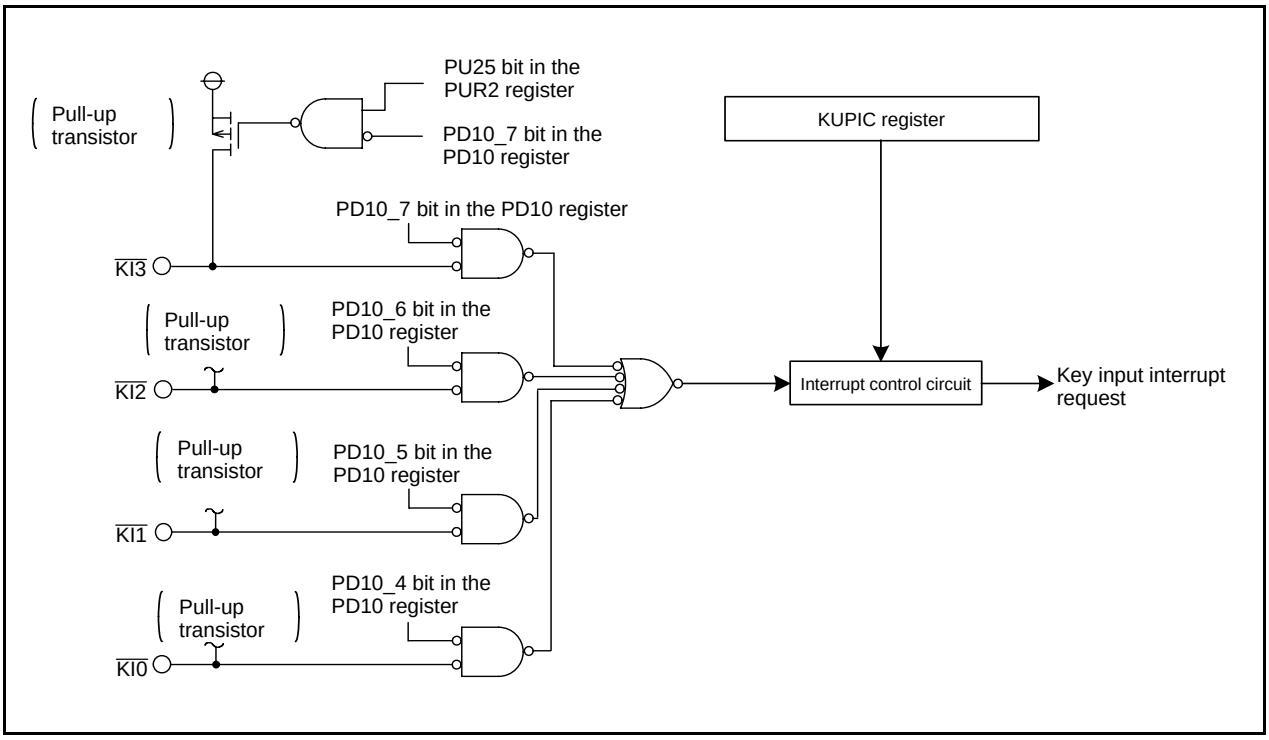


Figure 11.12 Key Input Interrupt

11.9 Address Match Interrupt

An address match interrupt request is generated immediately before executing the instruction at the address indicated by the RMADi register (i=0 to 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. For address match interrupts, the value of the PC that is saved to the stack area varies depending on the instruction being executed (refer to **11.5.7 Saving Registers**).
(The value of the PC that is saved to the stack area is not the correct return address.) Therefore, follow one of the methods described below to return from the address match interrupt.

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

Table 11.6 shows the Value of the PC that is Saved to the Stack Area when an Address Match Interrupt Request is Accepted
The address match interrupt is not available for an external space when an 8-bit wide external data bus is used.
Figure 11.13 shows the AIER and RMAD0 to RMAD1 Registers.

Table 11.6 Value of the PC that is Saved to the Stack Area when an Address Match Interrupt Request is Accepted

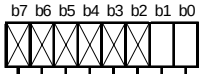
Instruction at the Address Indicated by the RMADi Register	Value of the PC that is saved to the stack area
• 16-bit op-code instruction • Instruction shown below among 8-bit operation code instructions ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ.B:S #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)	The address indicated by the RMADi register +2
Instructions other than the above	The address indicated by the RMADi register +1

Value of the PC that is saved to the stack area : Refer to **11.5.7 Saving Registers**.

Table 11.7 Relationship Between Address Match Interrupt Sources and Associated Registers

Address Match Interrupt sources	Address Match Interrupt Enable Bit	Address Match Interrupt Register
Address Match Interrupt 0	AIER0	RMAD0
Address Match Interrupt 1	AIER1	RMAD1

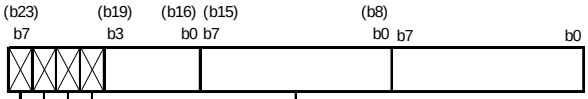
Address Match Interrupt Enable Register



Symbol	Address	After Reset
AIER	0009h	XXXXXX00b

Bit Symbol	Bit Name	Function	RW
AIER0	Address Match Interrupt 0 Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER1	Address Match Interrupt 1 Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
— (b7-b2)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

Address Match Interrupt Register i (i = 0 to 1)



Symbol	Address	After Reset
RMAD0	0012h to 0010h	X00000h
RMAD1	0016h to 0014h	X00000h

Function	Setting Range	RW
Address setting register for address match interrupt	00000h to FFFFFh	RW
Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

Figure 11.13 AIER and RMAD0 to RMAD1 Registers

12. 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. A watchdog timer interrupt is generated when an underflow occurs in the watchdog timer.
When the main clock source is selected for CPU clock, the divide-by-N value for the prescaler can be chosen to be 16 or 128. If a sub-clock is selected for CPU clock, the divide-by-N value for the prescaler is always 2 no matter how the WDC7 bit is set. The period of watchdog timer can be calculated as given below. The period of watchdog timer is, however, subject to an error due to the prescaler.

With main clock chosen for CPU clock

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

With sub-clock chosen for CPU clock

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

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

The watchdog timer is initialized by writing to the WDTS register. The prescaler is initialized after reset. Note that the watchdog timer and the prescaler both are inactive after reset, so that the watchdog timer is activated to start counting by writing to the WDTS register.
In stop mode, wait mode, and hold state, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.
Figure 12.1 shows the Watchdog Timer Block Diagram. Figure 12.2 shows the WDC and WDTS Register.

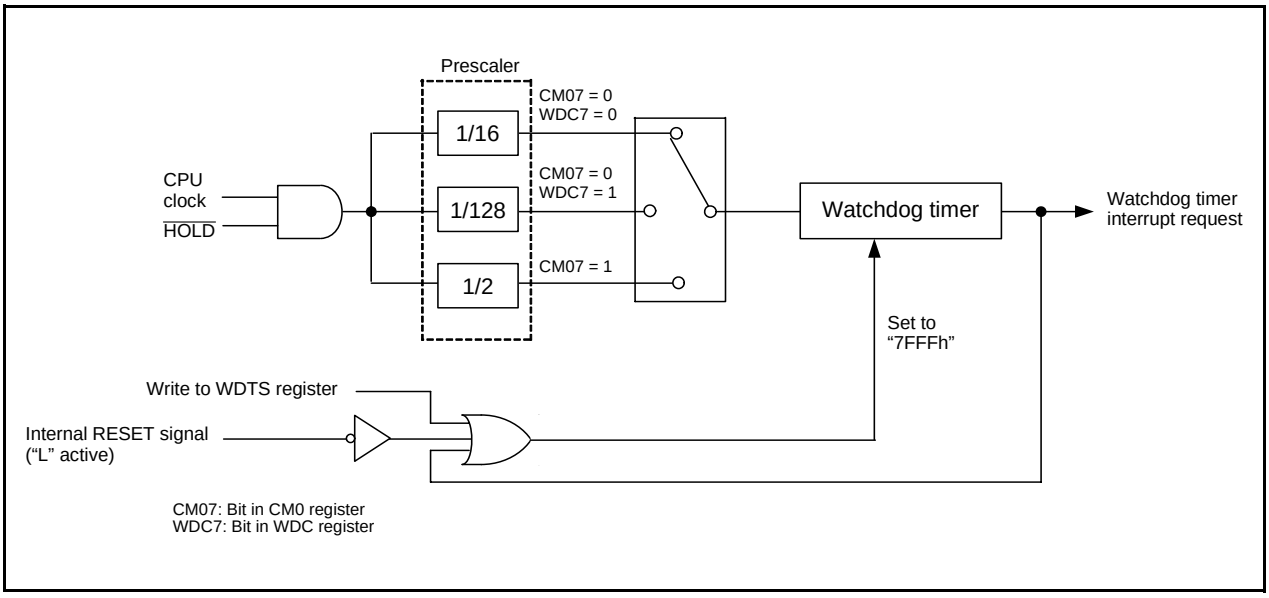
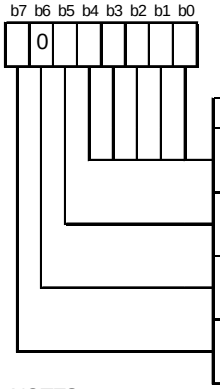


Figure 12.1 Watchdog Timer Block Diagram

Watchdog Timer Control Register

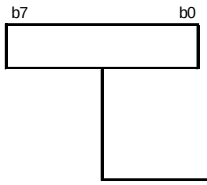


Symbol	Address	After Reset
WDC	000Fh	00XXXXXXb

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	High-order Bit of Watchdog Timer		RO
WDC5	Cold Start / Warm Start Discrimination Flag(1, 2)	0 : Cold Start 1 : Warm Start	RW
— (b6)	Reserved Bit	Set to “0”	RW
WDC7	Prescaler Select Bit	0 : Divided by 16 1 : Divided by 128	RW

NOTES :
1. Writing to the WDC register causes the WDC5 bit to be set to “1” (warm start). If the voltage applied to VCC1 is less than 4.0 V, either write to this register when the CPU clock frequency is 2 MHz or write twice.
2. The WDC5 bit is set to “0” (cold start) when power is turned on and can be set to “1” by program only.

Watchdog Timer Start Register (1)



Symbol	Address	After Reset
WDTS	000Eh	Indeterminate

Function	RW
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to “7FFFh” regardless of whatever value is written.	WO

NOTES :
1. Write to the WDTS register after the watchdog timer interrupt occurs.

Figure 12.2 WDC and WDTS Register

13. DMAC

The DMAC (Direct Memory Access Controller) allows data to be transferred without the CPU intervention.
Two DMAC channels are included. Each time a DMA request occurs, the DMAC transfers one (8 or 16-bit) data from the source address to the destination address. The DMAC uses the same data bus as used by the CPU. Because the DMAC has higher priority of bus control than the CPU and because it makes use of a cycle steal method, it can transfer one word (16 bits) or one byte (8 bits) of data within a very short time after a DMA request is generated. Figure 13.1 shows the DMAC Block Diagram. Table 13.1 lists the DMAC Specifications. Figures 13.2 to 13.4 shows the DMAC-related registers.

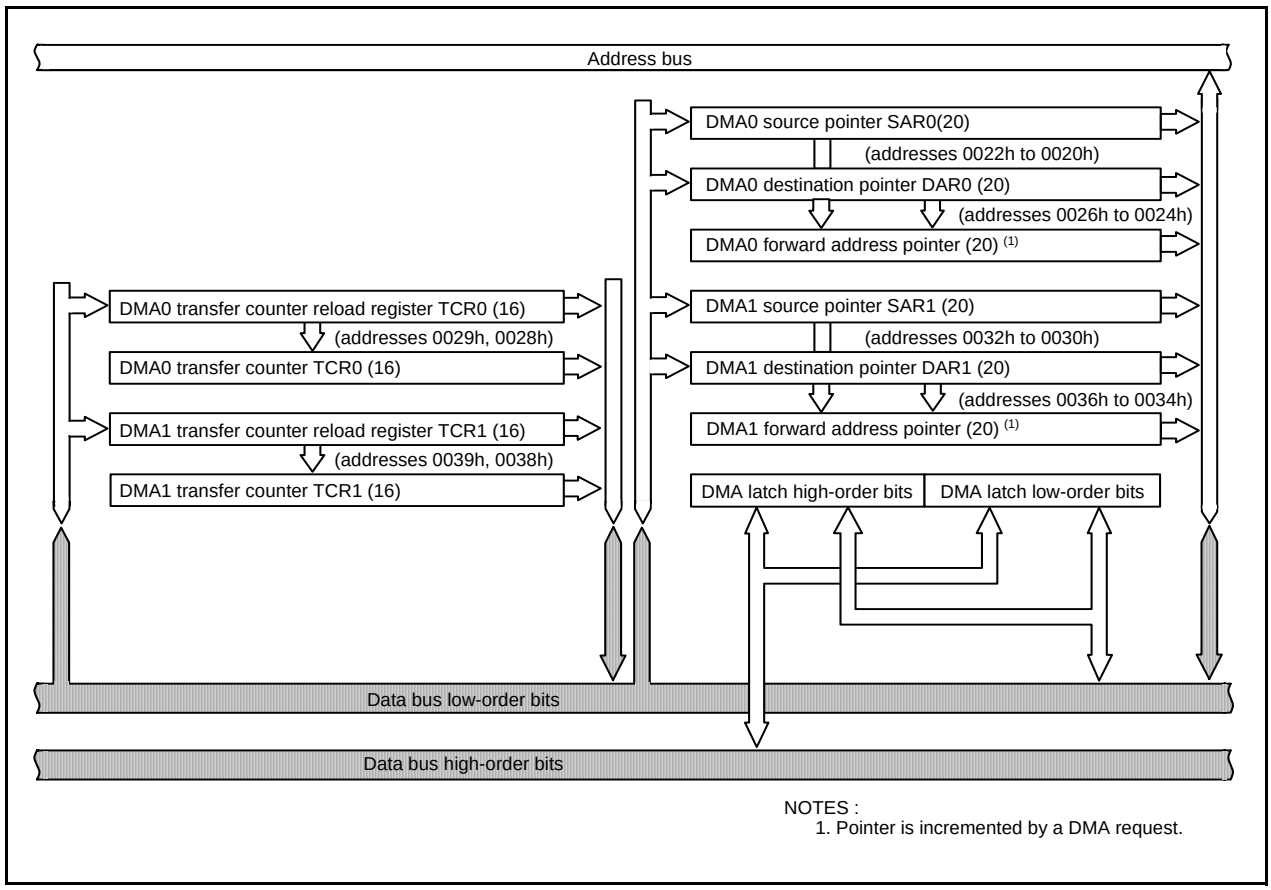


Figure 13.1 DMAC Block Diagram

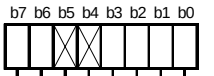
A DMA request is generated by a write to the DSR bit in the DMiSL register (i = 0 to 1), as well as by an interrupt request which is generated by any function specified by the DMS and DSEL3 to DSEL0 bits in the DMiSL register. However, unlike in the case of interrupt requests, DMA requests are not affected by the I flag and the interrupt control register, so that even when interrupt requests are disabled and no interrupt request can be accepted, DMA requests are always accepted. Furthermore, because the DMAC does not affect interrupts, the IR bit in the interrupt control register does not change state due to a DMA transfer.
A data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMiCON register = 1 (DMA enabled). However, if the cycle in which a DMA request is generated is faster than the DMA transfer cycle, the number of transfer requests generated and the number of times data is transferred may not match. Refer to **13.4 DMA Request** for details.

Table 13.1 DMAC Specifications

Item		Specification
No. of Channels		2 (cycle steal method)
Transfer Memory Space		• From any address in the 1-Mbyte space to a fixed address • From a fixed address to any address in the 1-Mbyte space • From a fixed address to a fixed address
Maximum No. of Bytes Transferred		128 Kbytes (with 16-bit transfers) or 64 Kbytes (with 8-bit transfers)
DMA Request Factors (1, 2)		Falling edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Both edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Timer A0 to timer A2 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transfer, UART0 reception interrupt requests UART1 transfer, UART1 reception interrupt requests UART2 transfer, UART2 reception interrupt requests A/D conversion interrupt requests Software triggers
Channel Priority		DMA0 > DMA1 (DMA0 takes precedence)
Transfer Unit		8 bits or 16 bits
Transfer Address Direction		Forward or fixed (The source and destination addresses cannot both be in the forward direction.)
Transfer Mode	Single Transfer	Transfer is completed when the DMAi transfer counter (i = 0 to 1) underflows after reaching the terminal count.
	Repeat Transfer	When the DMAi transfer counter underflows, it is reloaded with the value of the DMAi transfer counter reload register and a DMA transfer is continued with it.
DMA Interrupt Request Generation Timing		When the DMAi transfer counter underflowed
DMA Start up		Data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMAiCON register = 1 (enabled).
DMA Shutdown	Single Transfer	• When the DMAE bit is set to “0” (disabled) • After the DMAi transfer counter underflows
	Repeat Transfer	When the DMAE bit is set to “0” (disabled)
Reload Timing for Forward Address Pointer and Transfer Counter		When a data transfer is started after setting the DMAE bit to “1” (enabled), the forward address pointer is reloaded with the value of the SARi or the DARi pointer whichever is specified to be in the forward direction and the DMAi transfer counter is reloaded with the value of the DMAi transfer counter reload register.
DMA Transfer Cycles		Minimum 3 cycles between SFR and internal RAM

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

DMA0 Request Factor Select Register



Symbol	Address	After Reset
DM0SL	03B8h	00h

Bit Symbol	Bit Name	Function	RW
DSEL0	DMA Request Factor Select Bit	(NOTE 1)	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
— (b5-b4)	Nothing is assigned. When write, set to “0”. When read, their content are “0”.		—
DMS	DMA Request Factor Expansion Select Bit	0: Basic factor of request 1: Extended factor of request	RW
DSR	Software DMA Request Bit	A DMA request is generated by setting this bit to “1” when the DMS bit is “0” (basic factor) and the DSEL3 to DSEL0 bits are “0001b” (software trigger). The value of this bit when read is “0”.	RW

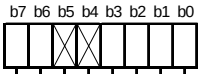
NOTES :

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

DSEL3 to DSEL0	DMS=0(Basic Factor of Request)	DMS=1(Extended Factor of Request)
0 0 0 0 b	Falling Edge of $\overline{INT0}$ Pin	—
0 0 0 1 b	Software Trigger	—
0 0 1 0 b	Timer A0	—
0 0 1 1 b	Timer A1	—
0 1 0 0 b	Timer A2	—
0 1 0 1 b	—	—
0 1 1 0 b	—	Two Edges of $\overline{INT0}$ Pin
0 1 1 1 b	Timer B0	—
1 0 0 0 b	Timer B1	—
1 0 0 1 b	Timer B2	—
1 0 1 0 b	UART0 Transmit	—
1 0 1 1 b	UART0 Receive	—
1 1 0 0 b	UART2 Transmit	—
1 1 0 1 b	UART2 Receive	—
1 1 1 0 b	A/D Conversion	—
1 1 1 1 b	UART1 Transmit	—

Figure 13.2 DM0SL Register

DMA1 Request Factor Select Register



Symbol	Address	After Reset
DM1SL	03BAh	00h

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

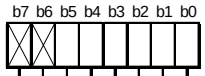
NOTES :

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

DSEL3 to DSEL0	DMS=0(Basic Factor of Request)	DMS=1(Extended Factor of Request)
0 0 0 0 b	Falling Edge of $\overline{\text{INT1}}$ Pin	—
0 0 0 1 b	Software Trigger	—
0 0 1 0 b	Timer A0	—
0 0 1 1 b	Timer A1	—
0 1 0 0 b	Timer A2	—
0 1 0 1 b	—	—
0 1 1 0 b	—	—
0 1 1 1 b	Timer B0	Two Edges of $\overline{\text{INT1}}$ Pin
1 0 0 0 b	Timer B1	—
1 0 0 1 b	Timer B2	—
1 0 1 0 b	UART0 Transmit	—
1 0 1 1 b	UART0 Receive/ACK0	—
1 1 0 0 b	UART2 Transmit	—
1 1 0 1 b	UART2 Receive/ACK2	—
1 1 1 0 b	A/D Conversion	—
1 1 1 1 b	UART1 Receive/ACK1	—

Figure 13.3 DM1SL Register

DMAi Control Register (i=0,1)



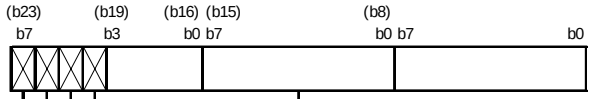
Symbol	Address	After Reset
DM0CON	002Ch	00000X00b
DM1CON	003Ch	00000X00b

Bit Symbol	Bit Name	Function	RW
DMBIT	Transfer Unit Bit Select Bit	0 : 16 bits 1 : 8 bits	RW
DMASL	Repeat Transfer Mode Select Bit	0 : Single transfer 1 : Repeat transfer	RW
DMAS	DMA Request Bit	0 : DMA not requested 1 : DMA requested	RW(1)
DMAE	DMA Enable Bit	0 : Disabled 1 : Enabled	RW
DSD	Source Address Direction Select Bit (2)	0 : Fixed 1 : Forward	RW
DAD	Destination Address Direction Select Bit (2)	0 : Fixed 1 : Forward	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, their contents are “0”.		—

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

Figure 13.4 DM0CON and DM1CON Register

DMAi Source Pointer (i = 0, 1) (1)

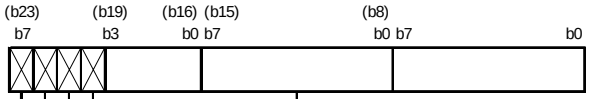


Symbol	Address	After Reset
SAR0	0022h to 0020h	Indeterminate
SAR1	0032h to 0030h	Indeterminate

Function	Setting Range	RW
Set the source address of transfer	00000h to FFFFFh	RW
Nothing is assigned. When write, set "0". When read, their contents are "0".		—

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

DMAi Destination Pointer (i = 0, 1) (1)

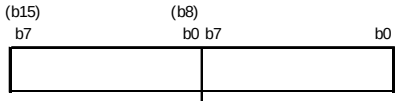


Symbol	Address	After Reset
DAR0	0026h to 0024h	Indeterminate
DAR1	0036h to 0034h	Indeterminate

Function	Setting Range	RW
Set the destination address of transfer	00000h to FFFFFh	RW
Nothing is assigned. When write, set "0". When read, their contents are "0".		—

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

DMAi Transfer Counter (i = 0, 1)



Symbol	Address	After Reset
TCR0	0029h to 0028h	Indeterminate
TCR1	0039h to 0038h	Indeterminate

Function	Setting Range	RW
Set the transfer count minus 1. The written value is stored in the DMAi transfer counter reload register, and when the DMAE bit in the DMiCON register is set to "1" (DMA enabled) or the DMAi transfer counter underflows when the DMASL bit in the DMiCON register is "1" (repeat transfer), the value of the DMAi transfer counter reload register is transferred to the DMAi transfer counter. When read, the DMAi transfer counter is read.	0000h to FFFFh	RW

Figure 13.5 SAR0, SAR1, DAR0, DAR1, TCR0 and TCR1 Registers

13.1 Transfer Cycles

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

13.1.1 Effect of Source and Destination Addresses

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

13.1.2 Effect of BYTE Pin Level

During memory extension and microprocessor modes, if 16 bits of data are to be transferred on an 8-bit data bus (input on the BYTE pin = high), the operation is accomplished by transferring 8 bits of data twice. Therefore, this operation requires two bus cycles to read data and two bus cycles to write data. Furthermore, if the DMAC is to access the internal area (internal ROM, internal RAM, or SFR), unlike in the case of the CPU, the DMAC does it through the data bus width selected by the BYTE pin.

13.1.3 Effect of Software Wait

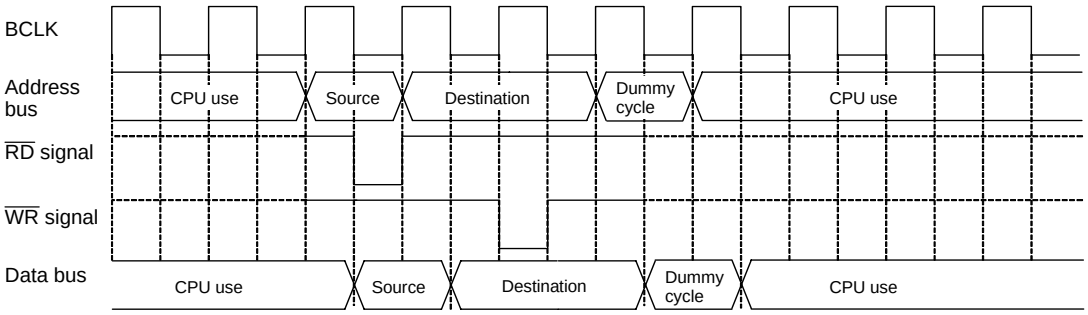
For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

13.1.4 Effect of $\overline{\text{RDY}}$ Signal

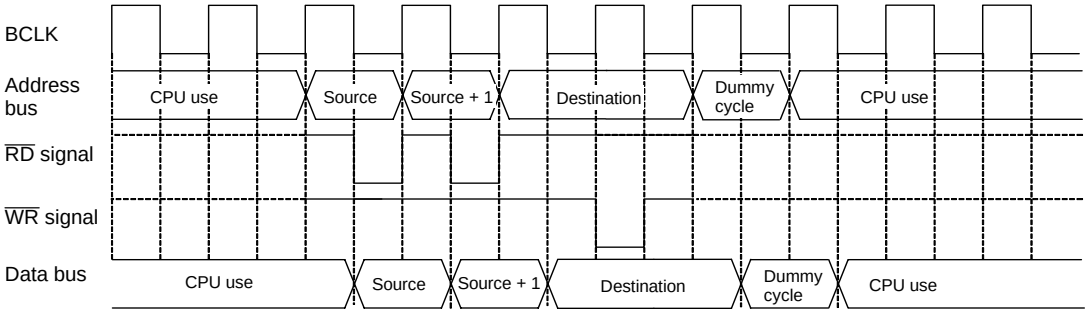
During memory extension and microprocessor modes, DMA transfers to and from an external area are affected by the RDY signal. Refer to **7.2.6 $\overline{\text{RDY}}$ Signal**.

Figure 13.6 shows the example of the Transfer Cycles for Source Read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16 bit units using an 8-bit bus ((2) on Figure 13.6), two source read bus cycles and two destination write bus cycles are required.

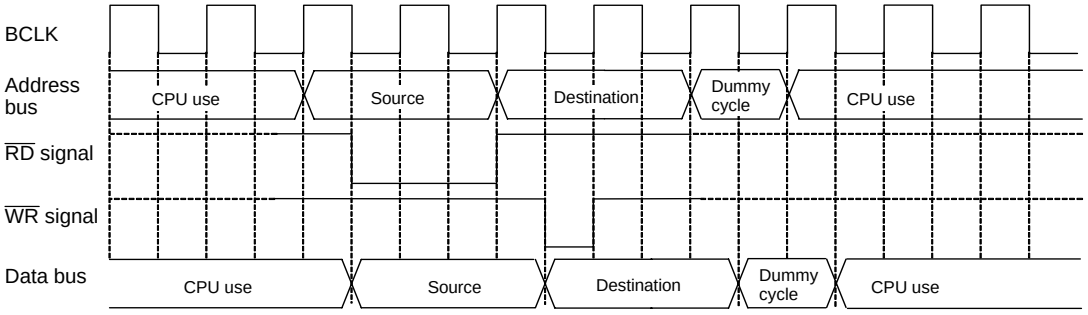
(1) When the transfer unit is 8 or 16 bits and the source of transfer is an even address



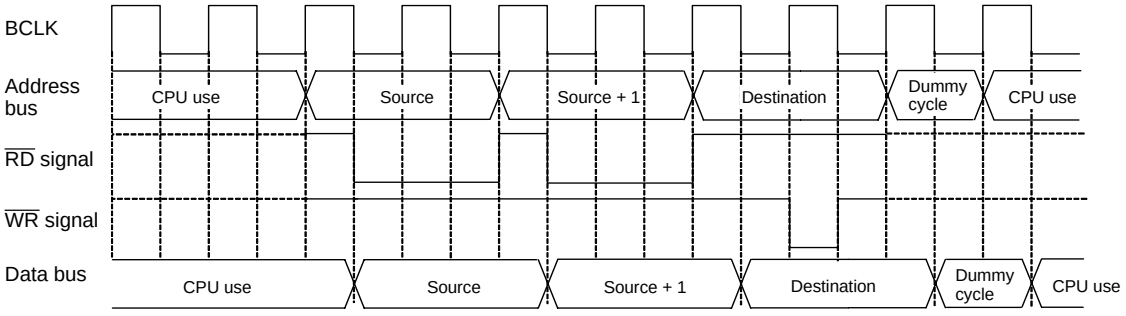
(2) When the transfer unit is 16 bits and the source address of transfer is an odd address, or when the transfer unit is 16 bits and an 8-bit bus is used



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



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



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

Figure 13.6 Transfer Cycles for Source Read

13.2 DMA Transfer Cycles

Any combination of even or odd transfer read and write addresses is possible. Table 13.2 lists the DMA Transfer Cycles. Table 13.3 lists the Coefficient j, k.
The number of DMAC transfer cycles can be calculated as follows:

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

Table 13.2 DMA Transfer Cycles

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

— : This condition does not exist.

Table 13.3 Coefficient j, k

	Internal Area			External Area	
	Internal ROM, RAM		SFR	Separate Bus	
	No Wait	With Wait	1-Wait	No Wait	1-Wait
j	1	2	2	1	2
k	1	2	2	2	2

13.3 DMA Enable

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

(1) Reload the forward address pointer with the SARi register value when the DSD bit in the DMiCON register is “1” (forward) or the DARi register value when the DAD bit in the DMiCON register is “1” (forward).

(2) Reload the DMAi transfer counter with the DMAi transfer counter reload register value.

If the DMAE bit is set to “1” again while it remains set, the DMAC performs the above operation. However, if a DMA request may occur simultaneously when the DMAE bit is being written, follow the steps below.

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

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

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

13.4 DMA Request

The DMAC can generate a DMA request as triggered by the factor of request that is selected with the DMS and DSEL3 to DSEL0 bits in the DMiSL register (i = 0, 1) on either channel. Table 13.4 lists the Timing at Which the DMAS Bit Changes State.

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

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

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

Table 13.4 Timing at Which the DMAS Bit Changes State

DMA Factor	DMAS Bit of the DMiCON Register	
	Timing at which the bit is set to “1”	Timing at which the bit is set to “0”
Software Trigger	When the DSR bit in the DMiSL register is set to “1”	• Immediately before a data transfer starts • When set by writing “0” in a program
Peripheral Function	When the interrupt control register for the peripheral function that is selected by the DSEL3 to DSEL0 and DMS bits in the DMiSL register has its IR bit set to “1”	

13.5 Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of BCLK), the DMAS bit on each channel is set to "1" (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1. The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period. Figure 13.7 shows an example of DMA Transfer by External Factors.

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

In addition, DMA requests cannot be counted up since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in Figure 13.7, occurs more than one time, the DMAS bit is set to "0" as soon as getting the bus arbitration. The bus arbitration is returned to the CPU when one transfer is completed.

Refer to **7.2.7 $\overline{\text{HOLD}}$ Signal** for details about bus arbitration between the CPU and DMA.

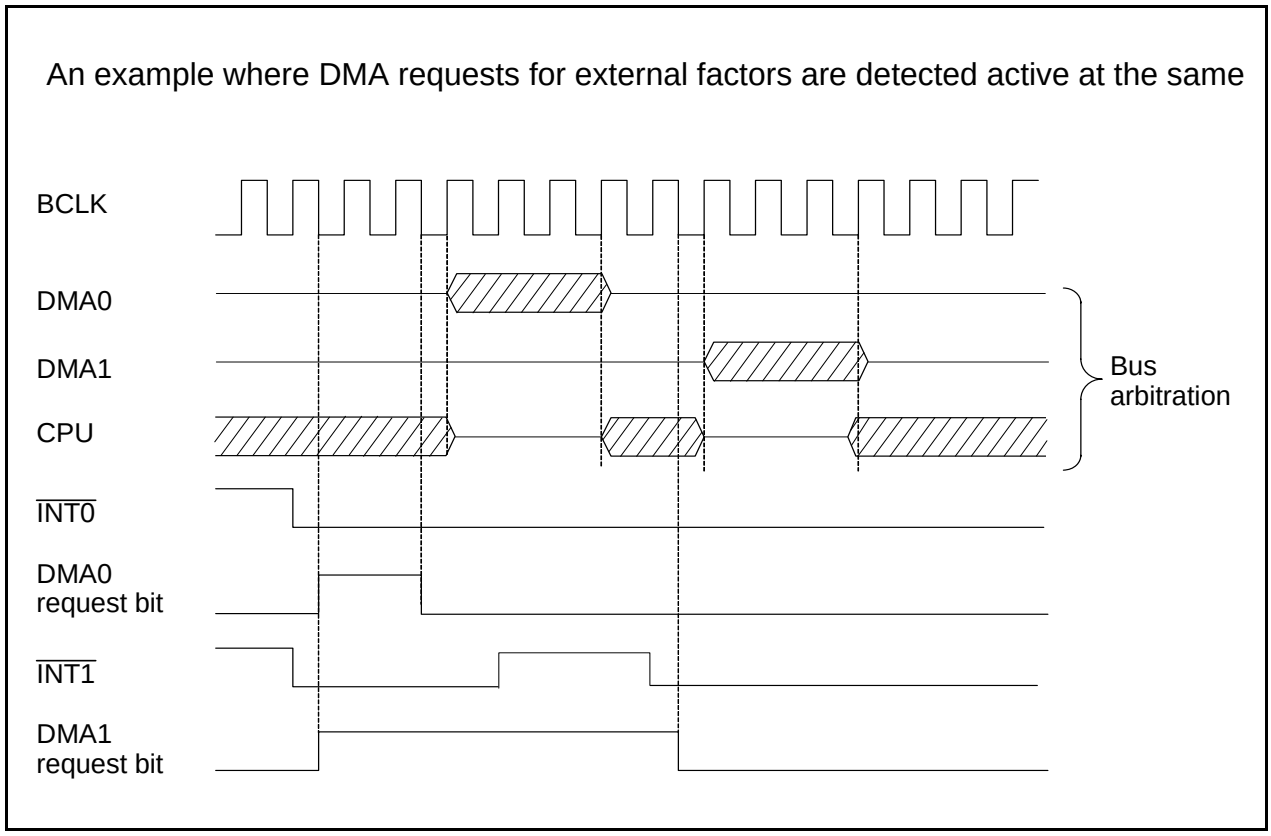


Figure 13.7 DMA Transfer by External Factors

14. Timers

Six 16-bit timers, each capable of operating independently of the others, can be classified by function as either Timer A (three) and Timer B (three). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc. Figures 14.1 and 14.2 show block diagrams of Timer A and Timer B configuration, respectively.

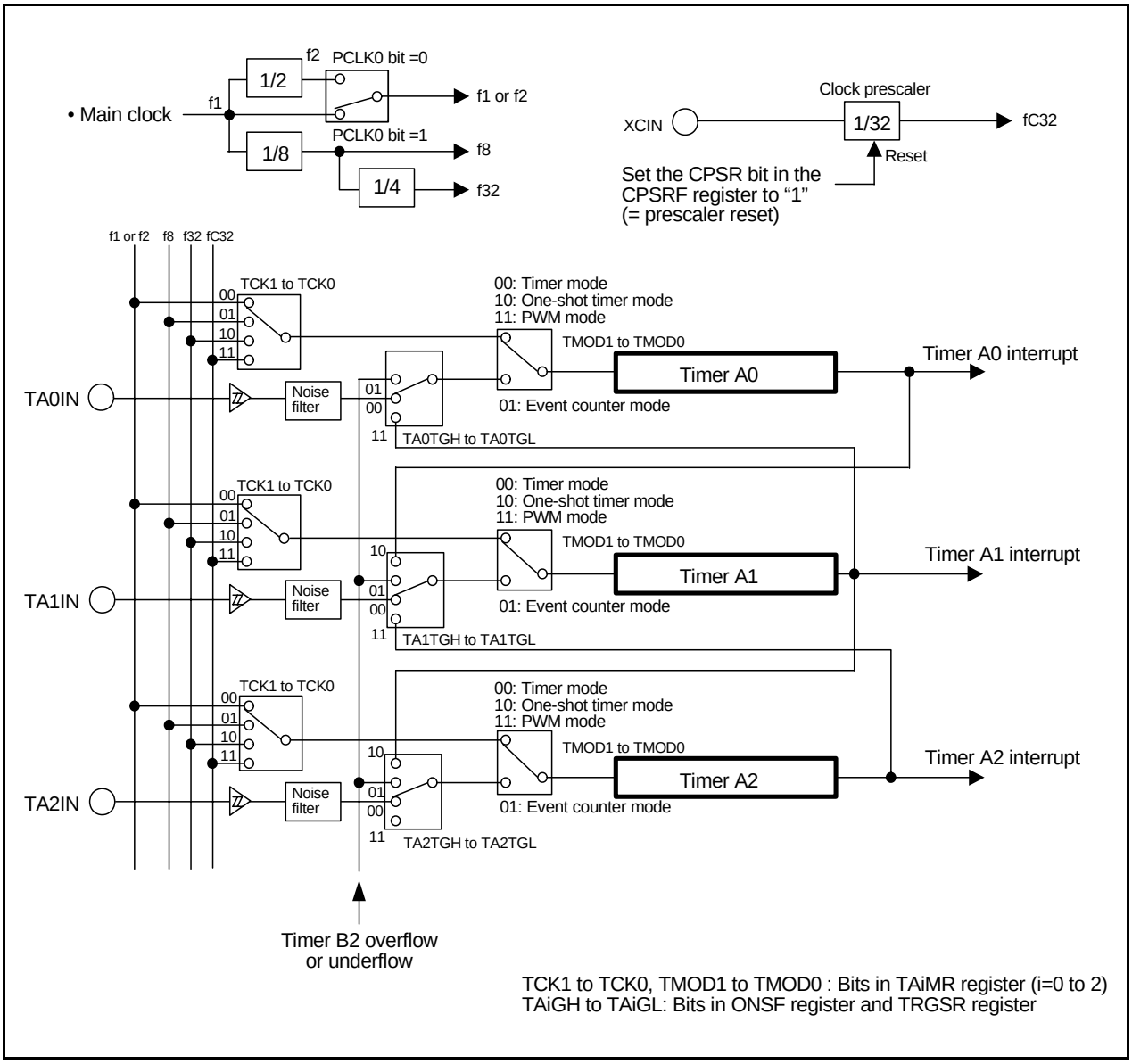


Figure 14.1 Timer A Configuration

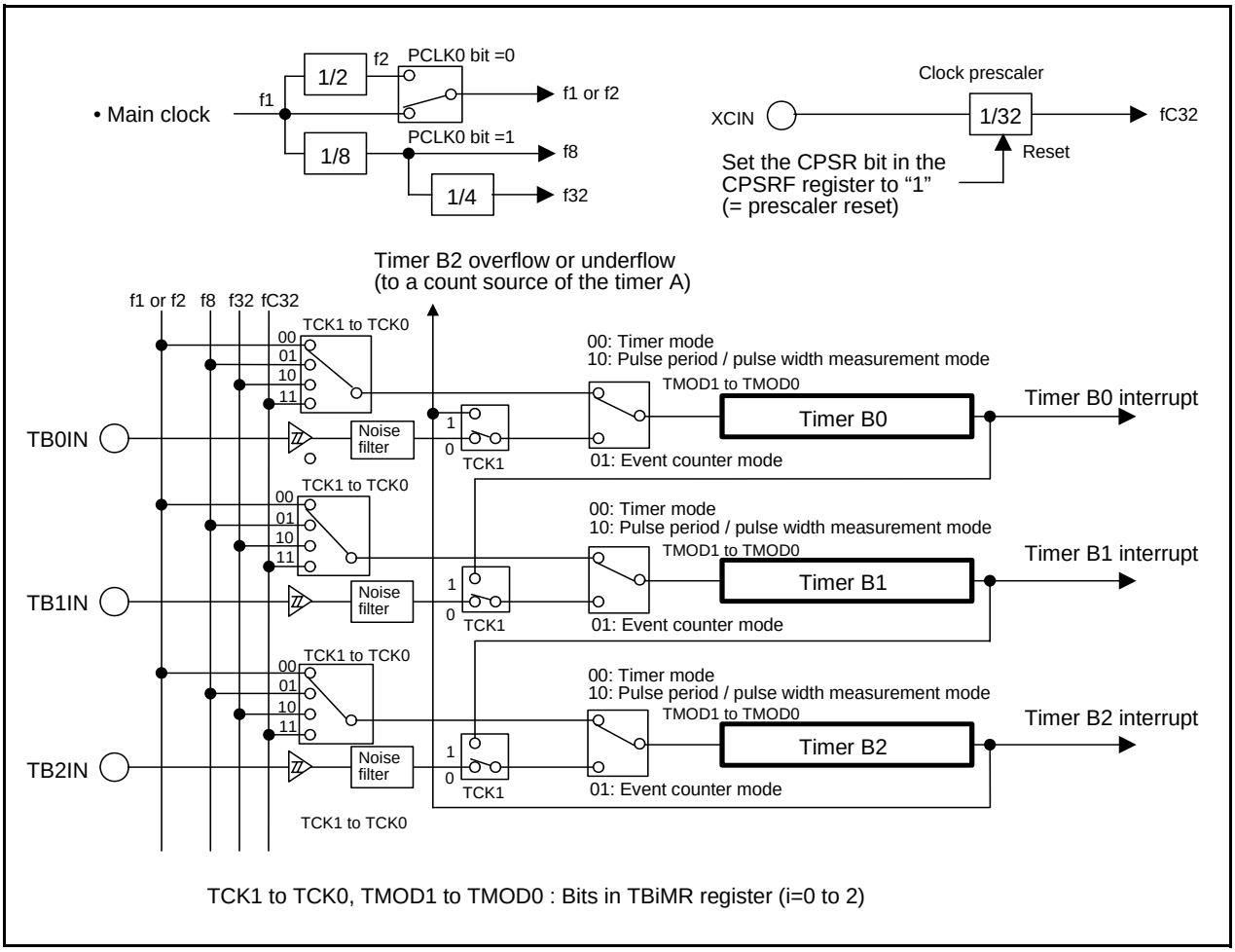


Figure 14.2 Timer B Configuration

14.1 Timer A

Figure 14.3 shows a Timer A Block Diagram. Figures 14.4 to 14.6 show registers related to the Timer A. The Timer A supports the following four modes. Except in event counter mode, Timers A0 to A2 all have the same function. Use the TMOD1 to TMOD0 bits in the TAiMR register (i = 0 to 2) to select the desired mode.

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

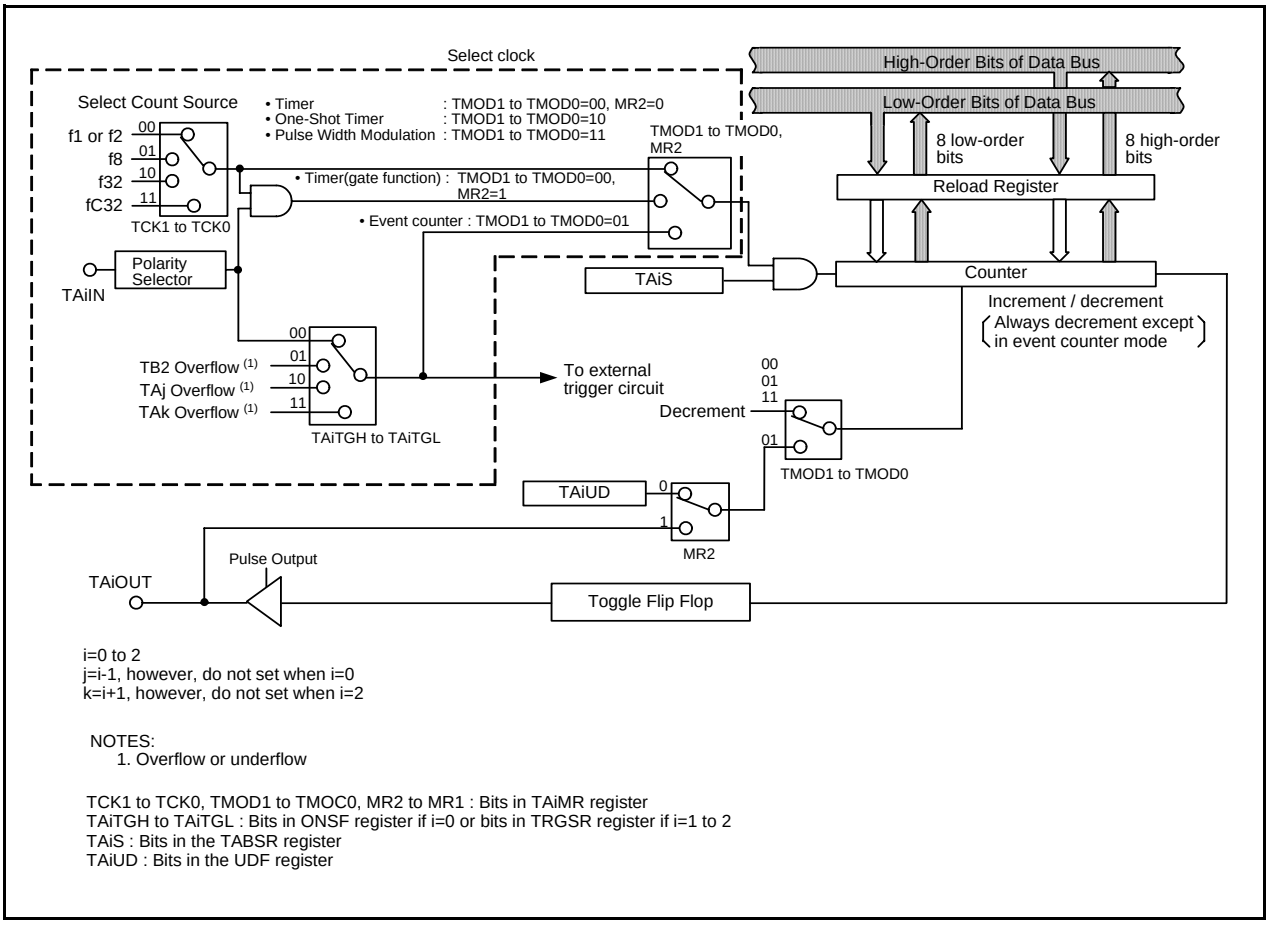


Figure 14.3 Timer A Block Diagram

Timer Ai Mode Register (i=0 to 2)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TA0MR to TA2MR	0396h to 0398h	00h

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

Timer Ai Register (i= 0 to 2) (1)

(b15) (b8)
b7 b0 b7 b0

Symbol	Address	After Reset
TA0	0387h, 0386h	Indeterminate
TA1	0389h, 0388h	Indeterminate
TA2	038Bh, 038Ah	Indeterminate

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

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

Figure 14.4 TAiMR and TAi Registers

Count Start Flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TABSR	0380h	000XX000b

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 Count Start Flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 Count Start Flag		RW
TA2S	Timer A2 Count Start Flag		RW
— (b4-b3)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—
TB0S	Timer B0 Count Start Flag	0 : Stops counting 1 : Starts counting	RW
TB1S	Timer B1 Count Start Flag		RW
TB2S	Timer B2 Count Start Flag		RW

Up/Down Flag (1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
UDF	0384h	XX0XX000b

Bit Symbol	Bit Name	Function	RW
TA0UD	Timer A0 Up/Down Flag	0 : Down count 1 : Up count Enabled by setting the MR2 bit in the TAiMR register to “0” (= switching source in UDF register) during event counter mode.	RW
TA1UD	Timer A1 Up/Down Flag		RW
TA2UD	Timer A2 Up/Down Flag		RW
— (b4-b3)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—
TA2P	Timer A2 Two-Phase Pulse Signal Processing Select Bit	0 : Two-phase pulse signal processing disabled 1 : Two-phase pulse signal processing enabled (2, 3)	WO
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

NOTES :
1. Use MOV instruction to write to this register.
2. Make sure the port direction bits for the TA2IN and TA2OUT pins are set to “0” (input mode).
3. When not using the two-phase pulse signal processing function, set the bit corresponding to timer A2 to “0”.

Figure 14.5 TABSR and UDF Registers

One-Shot Start Flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
ONSF	0382h	00XXX000b

Bit Symbol	Bit Name	Function	RW
TA0OS	Timer A0 One-Shot Start Flag	The timer starts counting by setting this bit to "1" while the TMOD1 to TMOD0 bits in the TAiMR register i = 0 to 2) = 10b (= one-shot timer mode) and the MR2 bit in the TAiMR register = 0 (=TAiOS bit enabled). When read, its content is 0".	RW
TA1OS	Timer A1 One-Shot Start Flag		RW
TA2OS	Timer A2 One-Shot Start Flag		RW
— (b5-b3)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—
TA0TGL	Timer A0 Event/Trigger Select Bit	b7 b6 0 0 : Input on TA0IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : Do not set 1 1 : TA1 is selected (2)	RW
TA0TGH			RW

NOTES :
1. Make sure the PD7_1 bit in the PD7 register is set to "0" (= input mode).
2. Overflow or underflow.

Trigger Select Register

b7 b6 b5 b4 b3 b2 b1 b0

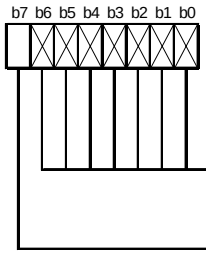
Symbol	Address	After Reset
TRGSR	0383h	XXXX0000b

Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 Event/Trigger Select Bit	b1 b0 0 0 : Input on TA1IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA0 is selected (2) 1 1 : TA2 is selected (2)	RW
TA1TGH			RW
TA2TGL	Timer A2 Event/Trigger Select Bit	b3 b2 0 0 : Input on TA2IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA1 is selected (2) 1 1 : Do not set	RW
TA2TGH			RW
— (b7-b4)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—

NOTES :
1. Make sure the port direction bits for the TA1IN to TA2IN pins are set to "0" (= input mode).
2. Overflow or underflow.

Figure 14.6 ONSF and TRGSR Registers

Clock Prescaler Reset Flag



Symbol	Address	After Reset
CPSRF	0381h	0XXXXXXXb

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

Figure 14.7 CPSRF Register

14.1.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 14.1). Figure 14.8 shows TAiMR Register in Timer Mode.

Table 14.1 Specifications in Timer Mode

Item	Specification
Count source	f1, f2, f8, f32, fC32
Count Operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide Ratio	1/(n+1) n: set value of TAi register (i= 0 to 2) 0000h to FFFFh
Count Start Condition	Set TAiS bit in TABSR register to “1” (= start counting)
Count Stop Condition	Set TAiS bit to “0” (= stop counting)
Interrupt Request Generation Timing	Timer underflow
TAiIN Pin Function	I/O port or gate input
TAiOUT Pin Function	I/O port or pulse output
Read from Timer	Count value can be read by reading TAi register
Write to Timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select Function	• Gate function Counting can be started and stopped by an input signal to TAiIN pin • Pulse output function Whenever the timer underflows, the output polarity of TAiOUT pin is inverted. When TAiS bit is set to “0” (stop counting), the pin outputs a low.

Timer Ai Mode Register (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol	Address	After Reset
TA0MR to TA2MR	0396h to 0398h	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Pulse Output Function Select Bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (1) (TAiOUT pin is a pulse output pin)	RW
MR1	Gate Function Select Bit	b4 b3 0 0 : Gate function not available 0 1 : (TAiIN pin functions as I/O port) 1 0 : Counts while input on the TAiIN pin is low (2) 1 1 : Counts while input on the TAiIN pin is high (2)	RW
MR2			RW
MR3	Set to “0” in timer mode		RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 or f2 (3) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTES :
1. TA0OUT pin is N-channel open drain output.
2. The port direction bit for the TAiIN pin is set to “0” (= input mode).
3. Selected by PCLK0 bit in the PCLKR register.

Figure 14.8 TAiMR Register in Timer Mode

14.1.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timer A2 can count two-phase external signals. Table 14.2 lists Specifications in Event Counter Mode (when not processing two-phase pulse signal). Figure 14.9 shows TAiMR Register in Event Counter Mode (when not using two-phase pulse signal processing).

Table 14.2 Specifications in Event Counter Mode (when not processing two-phase pulse signal)

Item	Specification
Count Source	• External signals input to TAiIN pin (i=0 to 2) (effective edge can be selected in program) • Timer B2 overflows or underflows, Timer Aj (j=i-1, however, do not set when i=0) overflows or underflows, Timer Ak (k=i+1, however, do not set when i=2) overflows or underflows
Count Operation	• Up-count or down-count can be selected by external signal or program • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divided Ratio	1/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count n : set value of TAi register 0000h to FFFFh
Count Start Condition	Set TAiS bit in the TABSR register to “1” (= start counting)
Count Stop Condition	Set TAiS bit to “0” (= stop counting)
Interrupt Request Generation Timing	Timer overflow or underflow
TAiIN Pin Function	I/O port or count source input
TAiOUT Pin Function	I/O port, pulse output, or up/down-count select input
Read from Timer	Count value can be read by reading TAi register
Write to Timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select Function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Whenever the timer underflows or underflows, the output polarity of TAiOUT pin is inverted. When TAiS bit is set to “0” (stop counting), the pin outputs a low.

Timer Ai Mode Register (i=0 to 2)
(when not using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol	Address	After Reset
TA0MR to TA2MR	0396h to 0398h	00h

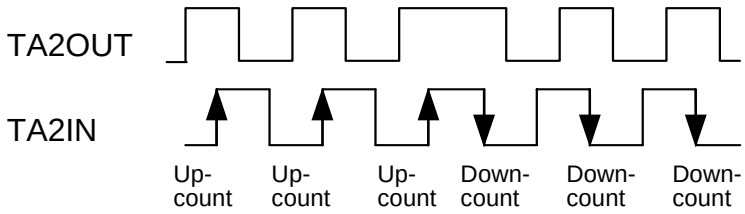
Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 0 1 : Event counter mode (1)	RW
TMOD1			RW
MR0	Pulse Output Function Select Bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (2) (TAiOUT pin functions as pulse output pin)	RW
MR1	Count Polarity Select Bit (3)	0 : Counts falling edge of external signal 1 : Counts rising edge of external signal	RW
MR2	Up/Down Switching Factor Select Bit	0 : UDF register 1 : Input signal to TAiOUT pin (4)	RW
MR3	Set to “0” in event counter mode		RW
TCK0	Count Operation Type Select Bit	0 : Reload type 1 : Free-run type	RW
TCK1	Can be “0” or “1” when not using two-phase pulse signal processing		RW

NOTES :
1. During event counter mode, the count source can be selected using the ONSF and TRGSR registers.
2. TA0OUT pin is N-channel open drain output.
3. Effective when the TAiTGH and TAiTGL bits in the ONSF or TRGSR register are “00b” (TAiIN pin input).
4. Count down when input on TAiOUT pin is low or count up when input on that pin is high. The port direction bit for TAiOUT pin is set to “0” (= input mode).

Figure 14.9 TAiMR Register in Event Counter Mode (when not using two-phase pulse signal processing)

Table 14.3 lists Specifications in Event Counter Mode (when processing two-phase pulse signal with Timer A2). Figure 14.10 shows TA2MR Register in Event Counter Mode (when using two-phase pulse signal processing with Timer A2).

Table 14.3 Specifications in Event Counter Mode (when processing two-phase pulse signal with Timer A2)

Item	Specification
Count Source	• Two-phase pulse signals input to TA2IN or TA2OUT pins
Count Operation	• Up-count or down-count can be selected by two-phase pulse signal • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divide Ratio	1/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count n : set value of TA2 register 0000h to FFFFh
Count Start Condition	Set TA2S bit of TABSR register to “1” (= start counting)
Count Stop Condition	Set TA2S bit to “0” (= stop counting)
Interrupt Request Generation Timing	Timer overflow or underflow
TA2IN Pin Function	Two-phase pulse input
TA2OUT Pin Function	Two-phase pulse input
Read from Timer	Count value can be read by reading Timer A2 register
Write to Timer	• When not counting and until the 1st count source is input after counting start Value written to TA2 register is written to both reload register and counter • When counting (after 1st count source input) Value written to TA2 register is written to reload register (Transferred to counter when reloaded next)
Select Function	• The timer counts up rising edges or counts down falling edges on TA2IN pin when input signals on TA2OUT pin is “H”. 

Timer A2 Mode Register
(when using two-phase pulse signal processing)

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

Symbol	Address	After Reset
TA2MR	0398h	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	To use two-phase pulse signal processing, set this bit to “0”.		RW
MR1	To use two-phase pulse signal processing, set this bit to “0”.		RW
MR2	To use two-phase pulse signal processing, set this bit to “1”.		RW
MR3	To use two-phase pulse signal processing, set this bit to “0”.		RW
TCK0	Count Operation Type Select Bit	0 : Reload type 1 : Free-run type	RW
TCK1	To use two-phase pulse signal processing, set this bit to “0”. (1)		RW

NOTES :
1. If two-phase pulse signal processing is desired, following register settings are required:
   • Set the TA2P bit in the UDF register to “1” (two-phase pulse signal processing function enabled).
   • Set the TA2TGH and TA2TGL bits in the TRGSR register to “00b” (TA2IN pin input).
   • Set the port direction bits for TA2IN and TA2OUT to “0” (input mode).

Figure 14.10 TA2MR Register in Event Counter Mode (when using two-phase pulse signal processing with Timer A2)

14.1.3 One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger (see Table 14.4). When the trigger occurs, the timer starts up and continues operating for a given period. Figure 14.11 shows the TAiMR Register in One-Shot Timer Mode.

Table 14.4 Specifications in One-shot Timer Mode

Item	Specification
Count Source	f1, f2, f8, f32, fC32
Count Operation	• Down-count • When the counter reaches 0000h, it stops counting after reloading a new value • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide Ratio	1/n n : set value of TAi register (i=0 to 2) 0000h to FFFFh However, the counter does not work if the divide-by-n value is set to 0000h.
Count start Condition	TAiS bit in the TABSR register = 1 (start counting) and one of the following triggers occurs. • External trigger input from the TAiIN pin • Timer B2 overflow or underflow, Timer Aj (j=i-1, however, do not set when i=0) overflow or underflow, Timer Ak (k=i+1, however, do not set when i=2) overflow or underflow • The TAiOS bit in the ONSF register is set to “1”(= timer starts)
Count Stop Condition	• When the counter is reloaded after reaching “0000h” • TAiS bit is set to “0” (= stop counting)
Interrupt Request Generation Timing	When the counter reaches “0000h”
TAiIN Pin Function	I/O port or trigger input
TAiOUT Pin Function	I/O port or pulse output
Read from Timer	An indeterminate value is read by reading TAi register
Write to Timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select Function	• Pulse output function The timer outputs a low when not counting and a high when counting.

Timer Ai Mode Register (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol	Address	After Reset
TA0MR to TA2MR	0396h to 0398h	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0	RW
TMOD1		1 0 : One-shot timer mode	RW
MR0	Pulse Output Function Select Bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (1) (TAiOUT pin functions as a pulse output pin)	RW
MR1	External Trigger Select Bit (2)	0 : Falling edge of input signal to TAiIN pin (3) 1 : Rising edge of input signal to TAiIN pin (3)	RW
MR2	Trigger Select Bit	0 : TAiOS bit is enabled 1 : Selected by TAiTGH to TAiTGL bits	RW
MR3	Set to “0” in one-shot timer mode		RW
TCK0	Count Source Select Bit	b7 b6	RW
TCK1		0 0 : f1 or f2 (4) 0 1 : f8 1 0 : f32 1 1 : fC32	RW

NOTES :
1. TA0OUT pin is N-channel open drain output.
2. Effective when the TAiTGH and TAiTGL bits in the ONSF or TRGSR register are “00b” (TAiIN pin input).
3. The port direction bit for the TAiIN pin is set to “0” (= input mode).
4. Selected by PCLK0 bit in the PCLKR register.

Figure 14.11 TAiMR Register in One-Shot Timer Mode

14.1.4 Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see Table 14.5). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 14.12 shows TAiMR Register in PWM Mode. Figures 14.13 and 14.14 show Example of 16-bit Pulse Width Modulator Operation and Example of 8-bit Pulse Width Modulator Operation.

Table 14.5 Specifications in PWM Mode

Item	Specification
Count Source	f1, f2, f8, f32, fC32
Count Operation	• Down-count (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new value at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs during counting
16-bit PWM	• High level width n / fj n : set value of TAi register (i=o to 2) • Cycle time $(2^{16}-1)$ / fj fixed fj: count source frequency (f1, f2, f8, f32, fC32)
8-bit PWM	• High level width n × (m+1) / fj n : set value of TAi register high-order address • Cycle time $(2^{8}-1)$ × (m+1) / fj m : set value of TAi register low-order address
Count Start Condition	• TAiS bit of TABSR register is set to “1” (= start counting) • The TAiS bit = 1 and external trigger input from the TAiIN pin • The TAiS bit = 1 and one of the following external triggers occurs • Timer B2 overflow or underflow, Timer Aj (j=i-1, however, do not set when i=0) overflow or underflow, Timer Ak (k=i+1, however, do not set when i=2) overflow or underflow
Count Stop Condition	TAiS bit is set to “0” (= stop counting)
Interrupt Request Generation Timing	PWM pulse goes “L”
TAiIN Pin Function	I/O port or trigger input
TAiOUT Pin Function	Pulse output
Read from Timer	An indeterminate value is read by reading TAi register
Write to Timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)

Timer Ai Mode Register (i= 0 to 2)

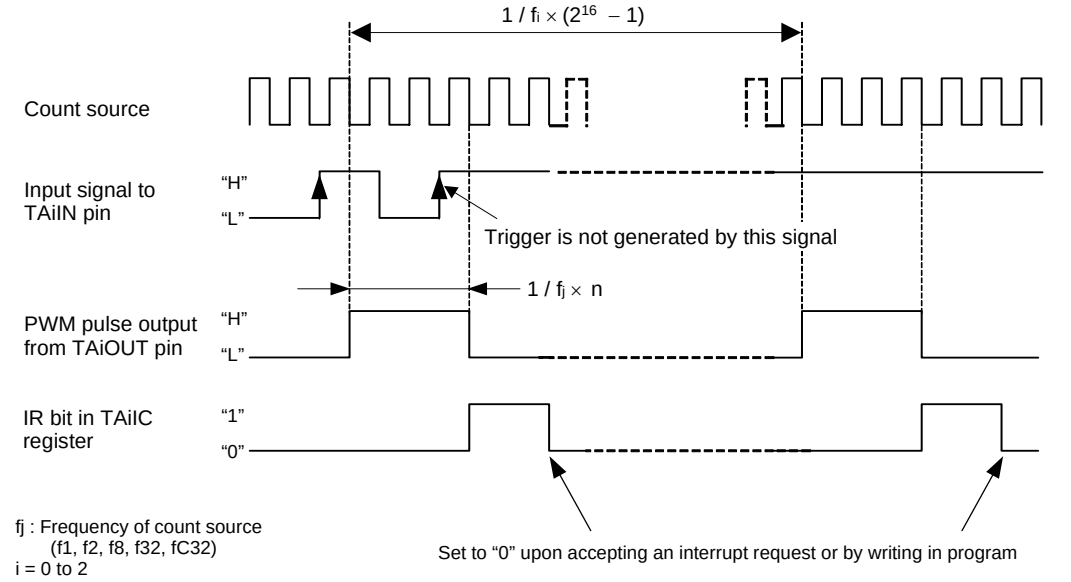
b7	b6	b5	b4	b3	b2	b1	b0
						1	1

Symbol	Address	After Reset
TA0MR to TA2MR	0396h to 0398h	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 1 1 : PWM mode (1)	RW
TMOD1			RW
MR0	Pulse Output Function Select Bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (4) (TAiOUT pin functions as a pulse output pin)	RW
MR1	External Trigger Select Bit (2)	0 : Falling edge of input signal to TAiIN pin (3) 1 : Rising edge of input signal to TAiIN pin (3)	RW
MR2	Trigger Select Bit	0 : Write “1” to TAiS bit in the TASF register 1 : Selected by TAiTGH to TAiTGL bits	RW
MR3	16/8-Bit PWM Mode Select Bit	0 : Functions as a 16-bit pulse width modulator 1 : Functions as an 8-bit pulse width modulator	RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 or f2 (5) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTES :
1. TA0OUT pin is N-channel open drain output.
2. Effective when the TAiTGH and TAiTGL bits in the ONSF or TRGSR register are “00b” (TAiIN pin input).
3. The port direction bit for the TAiIN pin is set to “0” (= input mode).
4. Set this bit to “1” (Pulse is output) to output PWM pulse.
5. Selected by PCLK0 bit in the PCLKR register.

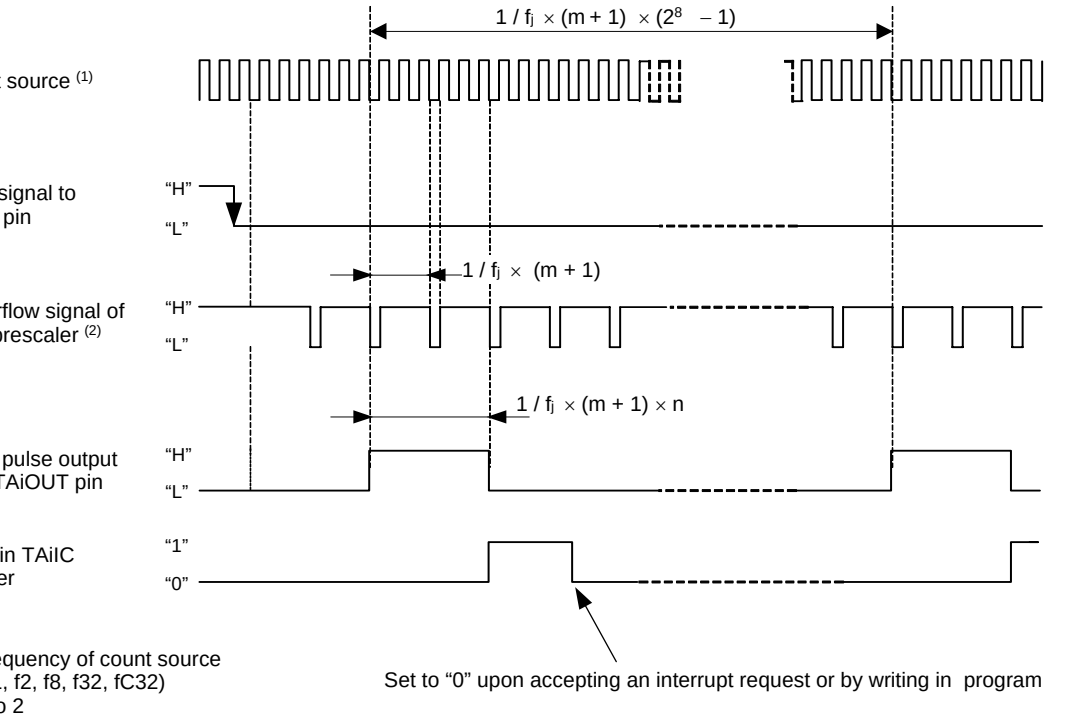
Figure 14.12 TAiMR Register in PWM Mode



fj : Frequency of count source
(f1, f2, f8, f32, fC32)
i = 0 to 2

NOTES :
1. n = 0000h to FFFEh.
2. This timing diagram is for the case where the TAi register is "0003h", the TAiTGH and TAiTGL bits in the ONSF or TRGSR register = 00b (TAiIN pin input), the MR1 bit of TAiMR register = 1 (rising edge), and the MR2 bit in the TAiMR register = 1 (trigger selected by TAiTGH and TAiTGL bits).

Figure 14.13 Example of 16-bit Pulse Width Modulator Operation



fj : Frequency of count source
(f1, f2, f8, f32, fC32)
i = 0 to 2

NOTES :
1. The 8-bit prescaler counts the count source.
2. The 8-bit pulse width modulator counts the output from the 8-bit prescaler underflow signal.
3. m = 00h to FFh; n = 00h to FEh.
4. This timing diagram is for the case where the TAi register is "0202h", the TAiTGH and TAiTGL bits of ONSF or TRGSR register = 00b (TAiIN pin input), the MR1 bit of TAiMR register = 0 (falling edge), and the MR2 bit of TAiMR register = 1 (trigger selected by TAiTGH and TAiTGL bits).

Figure 14.14 Example of 8-bit Pulse Width Modulator Operation

14.2 Timer B

Figure 14.15 shows a Timer B Block Diagram. Figures 14.16 and 14.17 show registers related to the Timer B. Timer B supports the following three modes. Use the TMOD1 and TMOD0 bits in the TBiMR register (i = 0 to 2) to select the desired mode.

- Timer Mode: The timer counts an internal count source.
- Event Counter Mode: The timer counts pulses from an external device or overflows or underflows of other timers.
- Pulse Period/Pulse Width Measurement Mode: The timer measures pulse period or pulse width of an external signal.

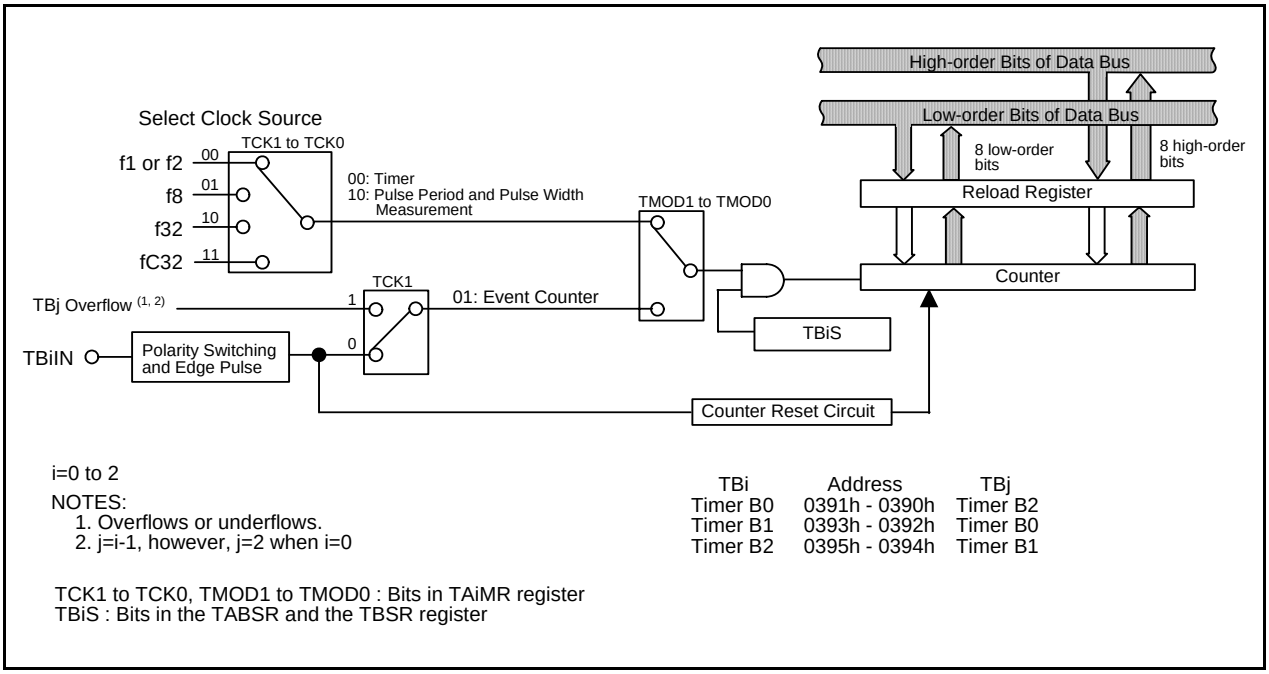
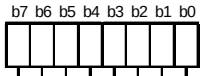


Figure 14.15 Timer B Block Diagram

Timer Bi Mode Register (i=0 to 2)

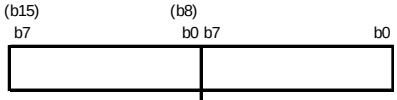


Symbol	Address	After Reset
TB0MR to TB2MR	039Bh to 039Dh	00XX0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period measurement mode, pulse width measurement mode 1 1 : Do not set to this value	RW
TMOD1			RW
MR0	—	Function varies with each operation mode	RW
MR1			RW
MR2			RW(1) —(2)
MR3			RO
TCK0	Count Source Select Bit	Function varies with each operation mode	RW
TCK1			RW

NOTES :
1. Timer B0.
2. Timer B1, Timer B2.

Timer Bi Register (i=0 to 2)(1)



Symbol	Address	After Reset
TB0	0391h, 0390h	Indeterminate
TB1	0393h, 0392h	Indeterminate
TB2	0395h, 0394h	Indeterminate

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

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

Figure 14.16 TBiMR and TBi Registers

Count Start Flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TABSR	0380h	000XX000b

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 Count Start Flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 Count Start Flag		RW
TA2S	Timer A2 Count Start Flag		RW
— (b4-b3)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—
TB0S	Timer B0 Count Start Flag	0 : Stops counting 1 : Starts counting	RW
TB1S	Timer B1 Count Start Flag		RW
TB2S	Timer B2 Count Start Flag		RW

Clock Prescaler Reset Flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
CPSRF	0381h	0XXXXXXXb

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

Figure 14.17 TABSR and CPSRF Registers

14.2.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 14.6). Figure 14.18 shows TBiMR Register in Timer Mode.

Table 14.6 Specifications in Timer Mode

Item	Specification
Count Source	f1, f2, f8, f32, fC32
Count Operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide Ratio	1/(n+1) n: set value of TBi register (i= 0 to 2) 0000h to FFFFh
Count Start Condition	Set TBiS bit(1) to “1” (= start counting)
Count Stop Condition	Set TBiS bit to “0” (= stop counting)
Interrupt Request Generation Timing	Timer underflow
TBiIN Pin Function	I/O port
Read from Timer	Count value can be read by reading TBi register
Write to Timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

NOTES:
1. The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.

Timer Bi Mode Register (i= 0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol	Address	After Reset
TB0MR to TB2MR	039Bh to 039Dh	00XX0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Has no effect in timer mode		RW
MR1	Can be set to “0” or “1”		RW
MR2	TB0MR register Set to “0” in timer mode		RW
	TB1MR, TB2MR registers Nothing is assigned. When write, set to “0”. When read, its content is indeterminate		—
MR3	When write in timer mode, set to “0”. When read in timer mode, its content is indeterminate.		RO
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 or f2 (1) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTES :
1. Selected by PCLK0 bit in the PCLKR register.

Figure 14.18 TBiMR Register in Timer Mode

14.2.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers (see Table 14.7). Figure 14.19 shows TBiMR Register in Event Counter Mode.

Table 14.7 Specifications in Event Counter Mode

Item	Specification
Count Source	• External signals input to TBiIN pin (i=0 to 2) (effective edge can be selected in program) • Timer Bj overflow or underflow (j=i-1, however, j=2 if i=0)
Count Operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide Ratio	1/(n+1) n: set value of TBi register 0000h to FFFFh
Count Start Condition	Set TBiS bit(1) to “1” (= start counting)
Count Stop Condition	Set TBiS bit to “0” (= stop counting)
Interrupt Request Generation Timing	Timer underflow
TBiIN Pin Function	Count source input
Read from Timer	Count value can be read by reading TBi register
Write to Timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

NOTES:
1. The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7.

Timer Bi Mode Register (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol	Address	After Reset
TB0MR to TB2MR	039Bh to 039Dh	00XX0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	Count Polarity Select Bit (1)	b3 b2 0 0 : Counts falling edges of external signal 0 1 : Counts rising edges of external signal 1 0 : Counts falling and rising edges external signal 1 1 : Do not set to this value	RW
MR1			RW
MR2	TB0MR register Set to “0” in event counter mode		RW
	TB1MR, TB2MR registers Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
MR3	When write in event counter mode, set to “0”. When read in event counter mode, its content is indeterminate.		RO
TCK0	Has no effect in event counter mode. Can be set to “0” or “1”.		RW
TCK1	Event Clock Select	0 : Input from TBiIN pin (2) 1 : TBj overflow or underflow (j = i – 1, however, j = 2 if i = 0)	RW

NOTES :
1. Effective when the TCK1 bit = 0 (input from TBiIN pin). If the TCK1 bit = 1 (TBj overflow or underflow), these bits can be set to “0” or “1”.
2. The port direction bit for the TBiIN pin must be set to “0” (= input mode).

Figure 14.19 TBiMR Register in Event Counter Mode

14.2.3 Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal (see Table 14.8). Figure 14.20 shows TBiMR Register in Pulse Period and Pulse Width Measurement Mode. Figure 14.21 shows the Operation Timing when Measuring a Pulse Period. Figure 14.22 shows the Operation Timing when Measuring a Pulse Width.

Table 14.8 Specifications in Pulse Period and Pulse Width Measurement Mode

Item	Specification
Count Source	f1, f2, f8, f32, fC32
Count Operation	• Up-count • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to “0000h” to continue counting.
Count Start Condition	Set TBiS (i=0 to 2) bit (3) to “1” (= start counting)
Count Stop Condition	Set TBiS bit to “0” (= stop counting)
Interrupt Request Generation Timing	• When an effective edge of measurement pulse is input (1) • Timer overflow. When an overflow occurs, MR3 bit in the TBiMR register is set to “1” (overflowed) simultaneously. MR3 bit is set to “0” (no overflow) by writing to TBiMR register at the next count timing or later after MR3 bit was set to “1”. At this time, make sure TBiS bit is set to “1” (start counting).
TBiIN Pin Function	Measurement pulse input
Read from Timer	Contents of the reload register (measurement result) can be read by reading TBi register (2)
Write to Timer	Value written to TBi register is written to neither reload register nor counter

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

Timer Bi Mode Register (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
						1	0

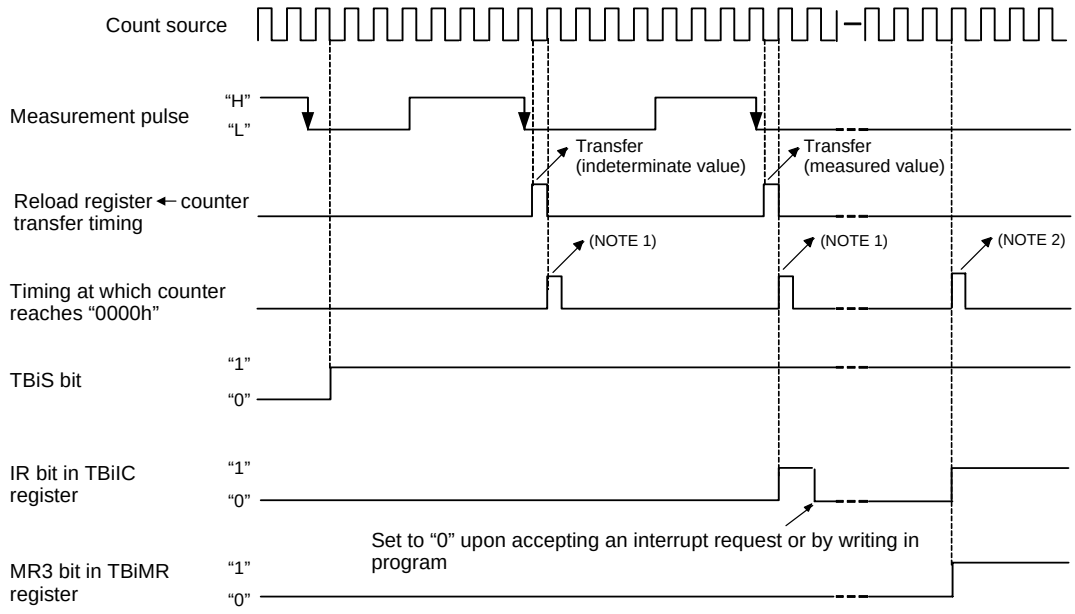
Symbol	Address	After Reset
TB0MR to TB2MR	039Bh to 039Dh	00XX0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation Mode Select Bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	RW
TMOD1			RW
MR0	Measurement Mode Select Bit	b3 b2 0 0 : Pulse period measurement (Measurement between a falling edge and the next falling edge of measured pulse) 0 1 : Pulse period measurement (Measurement between a rising edge and the next rising edge of measured pulse) 1 0 : Pulse width measurement (Measurement between a falling edge and the next rising edge of measured pulse and between a rising edge and the next falling edge) 1 1 : Do not set to this value	RW
MR1			RW
MR2	TB0MR register Set to “0” in pulse period and pulse width measurement mode		RW
	TB1MR, TB2MR registers Nothing is assigned. When write, set to “0”. When read, its content turns out to be indeterminate.		—
MR3	Timer Bi Overflow Flag (1)	0 : Timer did not overflow 1 : Timer has overflowed	RO
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 or f2 (2) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTES :

1. This flag is indeterminate after reset. When the TBiS bit = 1 (start counting), the MR3 bit is cleared to “0” (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to “1” (overflowed). The MR3 bit cannot be set to “1” in a program. The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.
2. Selected by PCLK0 bit in the PCLKR register.

Figure 14.20 TBiMR Register in Pulse Period and Pulse Width Measurement Mode

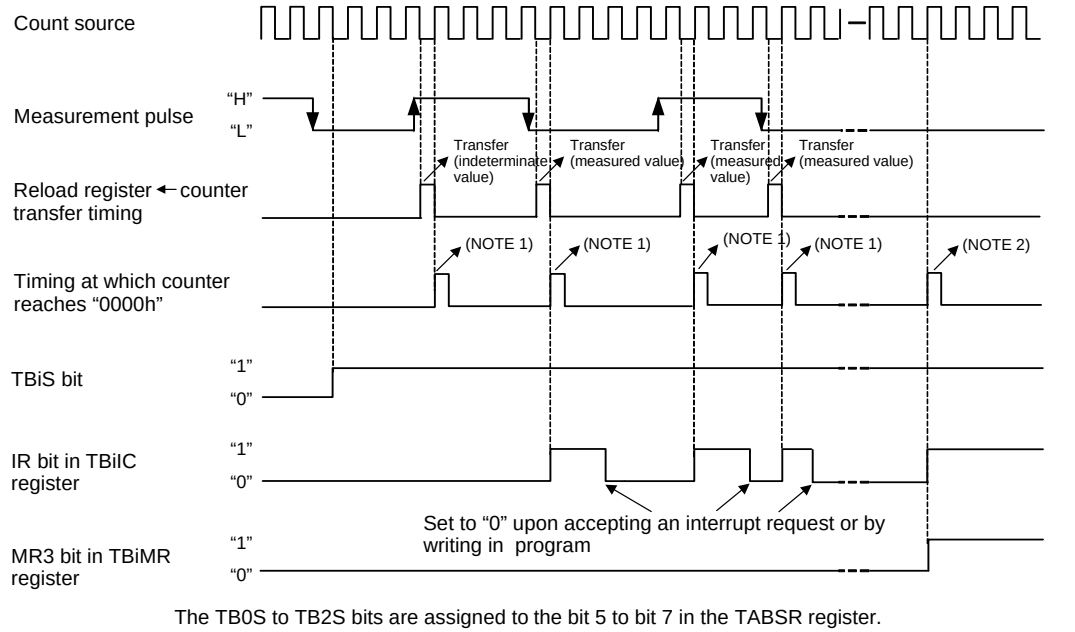


The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.

i = 0 to 2

NOTES :
1. Counter is initialized at completion of measurement.
2. Timer has overflowed.
3. This timing diagram is for the case where the MR1 to MR0 bits in the TBiMR register are "00b" (measure the interval from falling edge to falling edge of the measurement pulse).

Figure 14.21 Operation Timing when Measuring a Pulse Period



The TB0S to TB2S bits are assigned to the bit 5 to bit 7 in the TABSR register.

i = 0 to 2

NOTES :
1. Counter is initialized at completion of measurement.
2. Timer has overflowed.
3. This timing diagram is for the case where the MR1 to MR0 bits in the TBiMR register are "10b" (measure the interval from a falling edge to the next rising edge and the interval from a rising edge to the next falling edge of the measurement pulse).

Figure 14.22 Operation Timing when Measuring a Pulse Width

15. Serial Interface

Serial interface is configured with 3 channels: UART0 to UART2.

15.1 UARTi (i=0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other. Figures 15.1 to 15.3 shows the block diagram of UART0 to UART2. Figure 15.4 shows the UARTi Transmit/ Receive Unit.

UARTi has the following modes:

- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode).
- Special mode 1 (I^2C mode)
- Special mode 2
- Special mode 3 (Bus collision detection function, IE mode) : UART2
- Special mode 4 (SIM mode) : UART2

Figures 15.5 to 15.11 show the UARTi-related registers.
Refer to tables listing each mode for register setting.

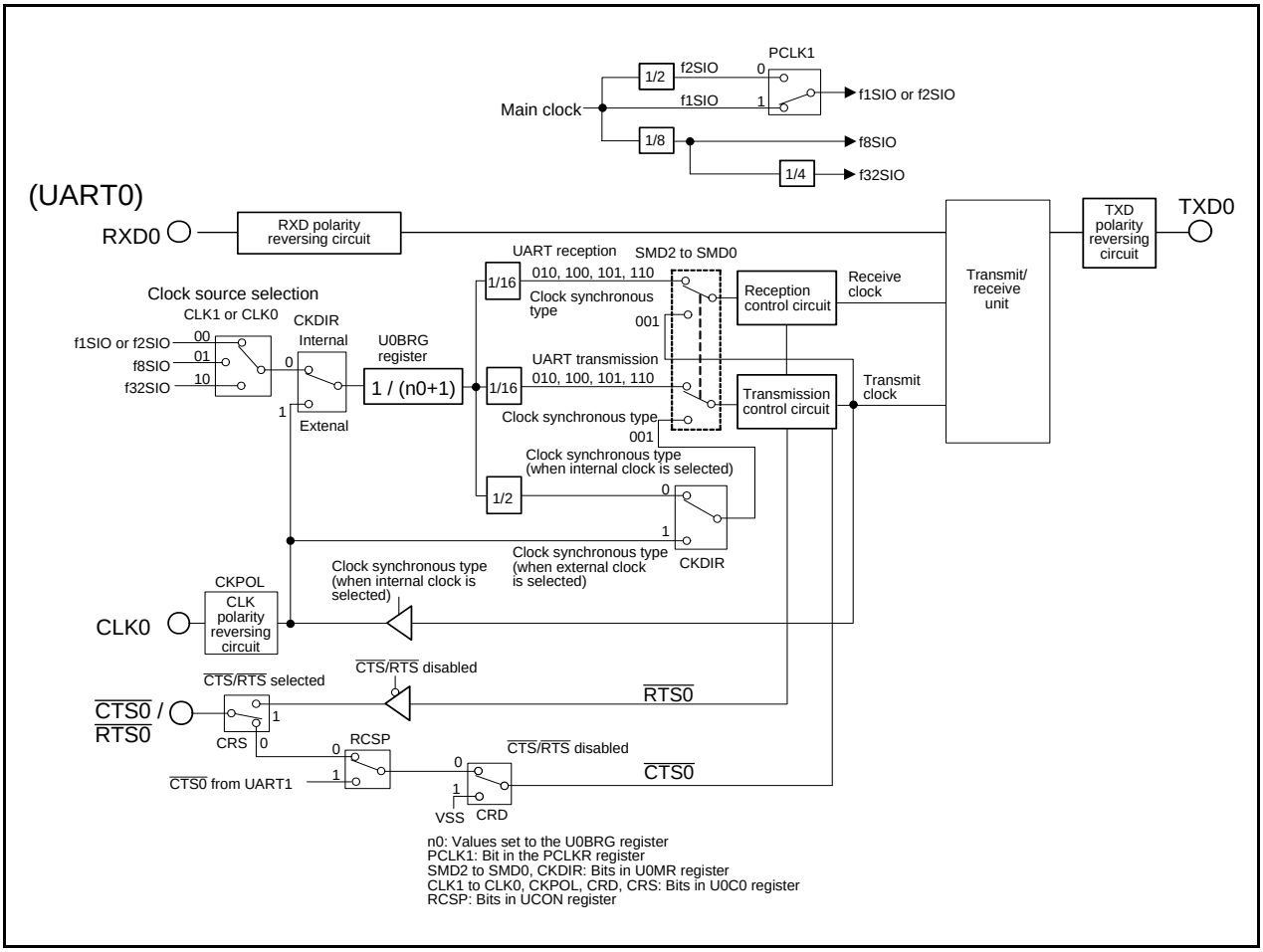


Figure 15.1 UART0 Block Diagram

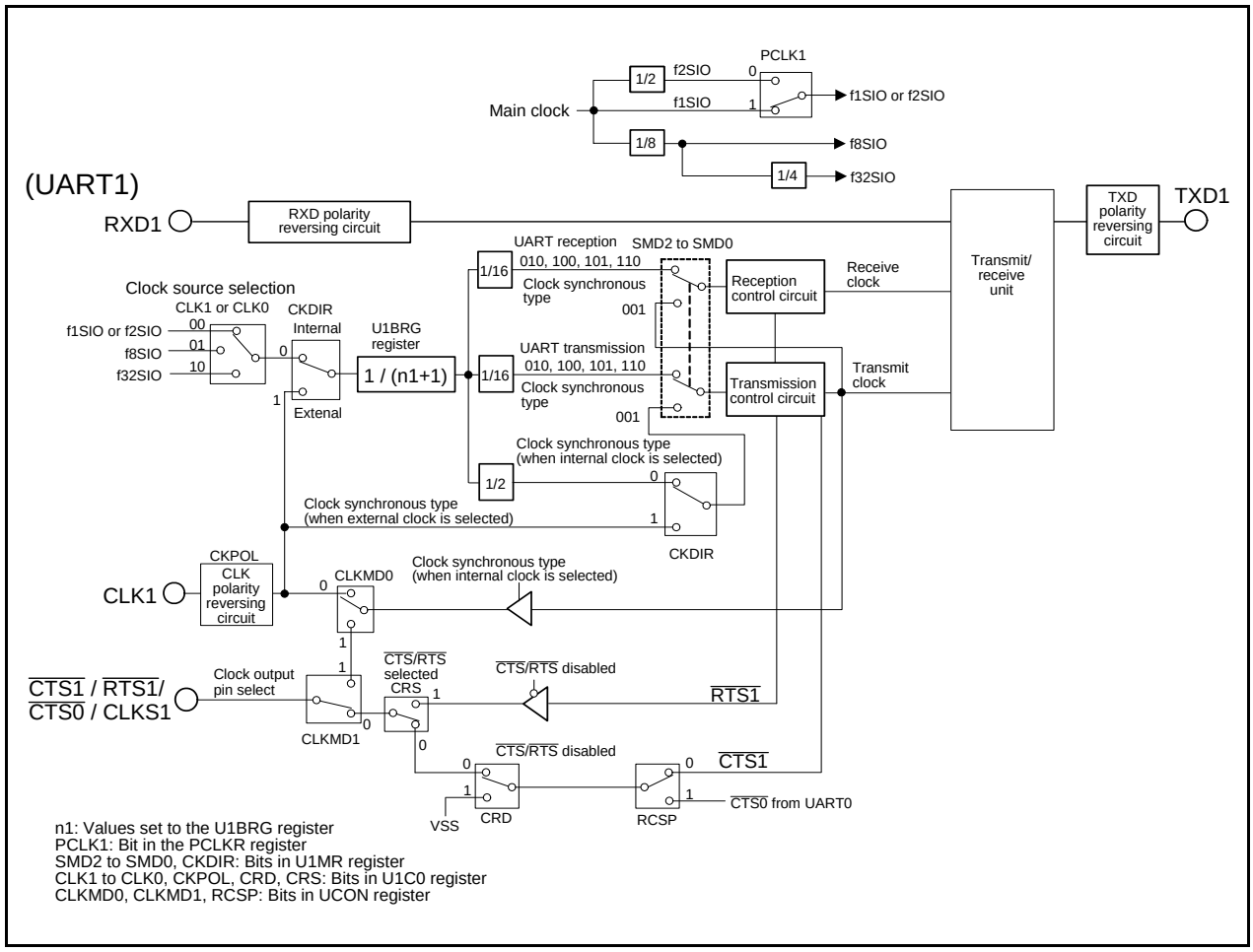
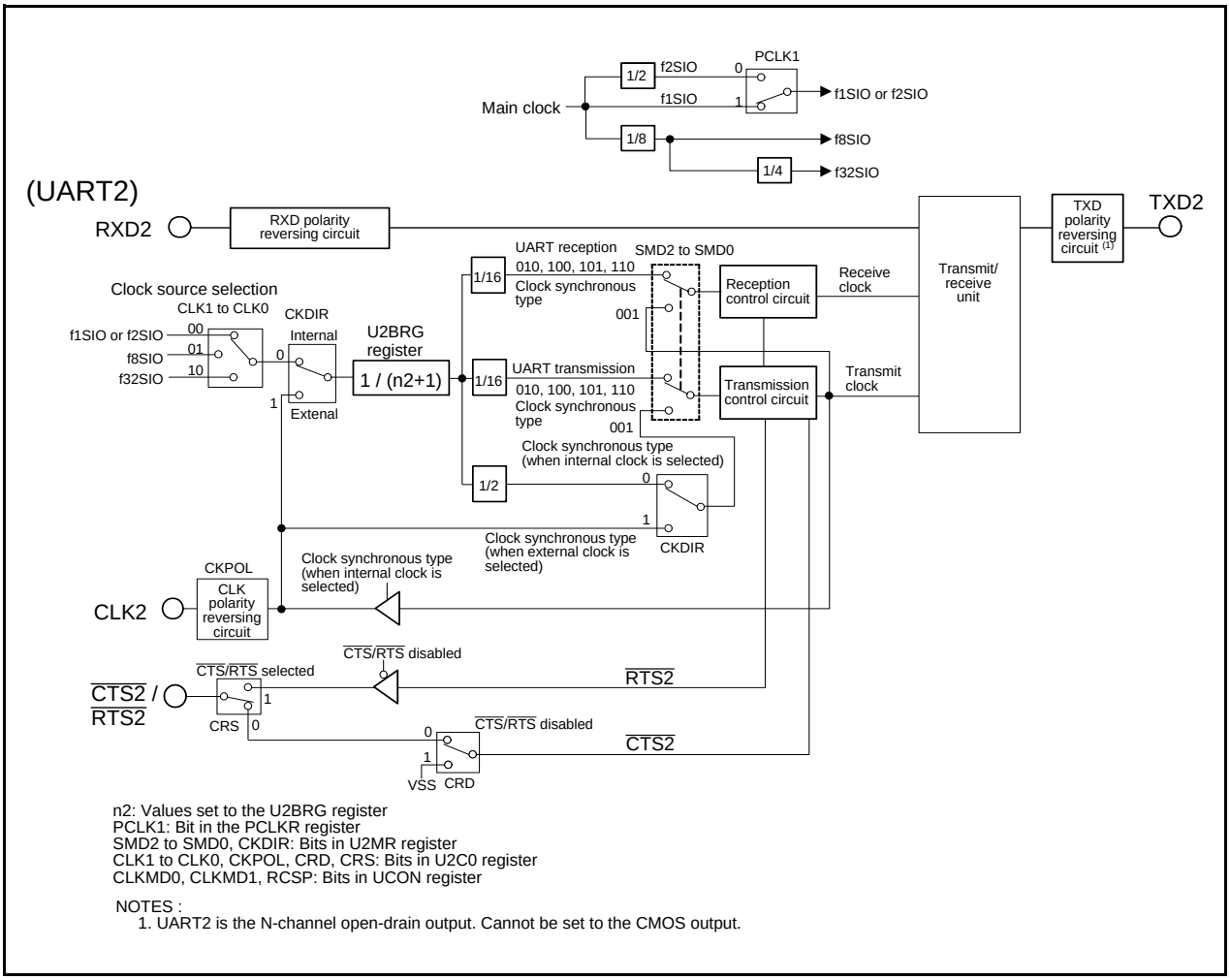


Figure 15.2 UART1 Block Diagram



n2: Values set to the U2BRG register
PCLK1: Bit in the PCLKR register
SMD2 to SMD0, CKDIR: Bits in U2MR register
CLK1 to CLK0, CKPOL, CRD, CRS: Bits in U2C0 register
CLKMD0, CLKMD1, RCSP: Bits in UCON register

NOTES :
1. UART2 is the N-channel open-drain output. Cannot be set to the CMOS output.

Figure 15.3 UART2 Block Diagram

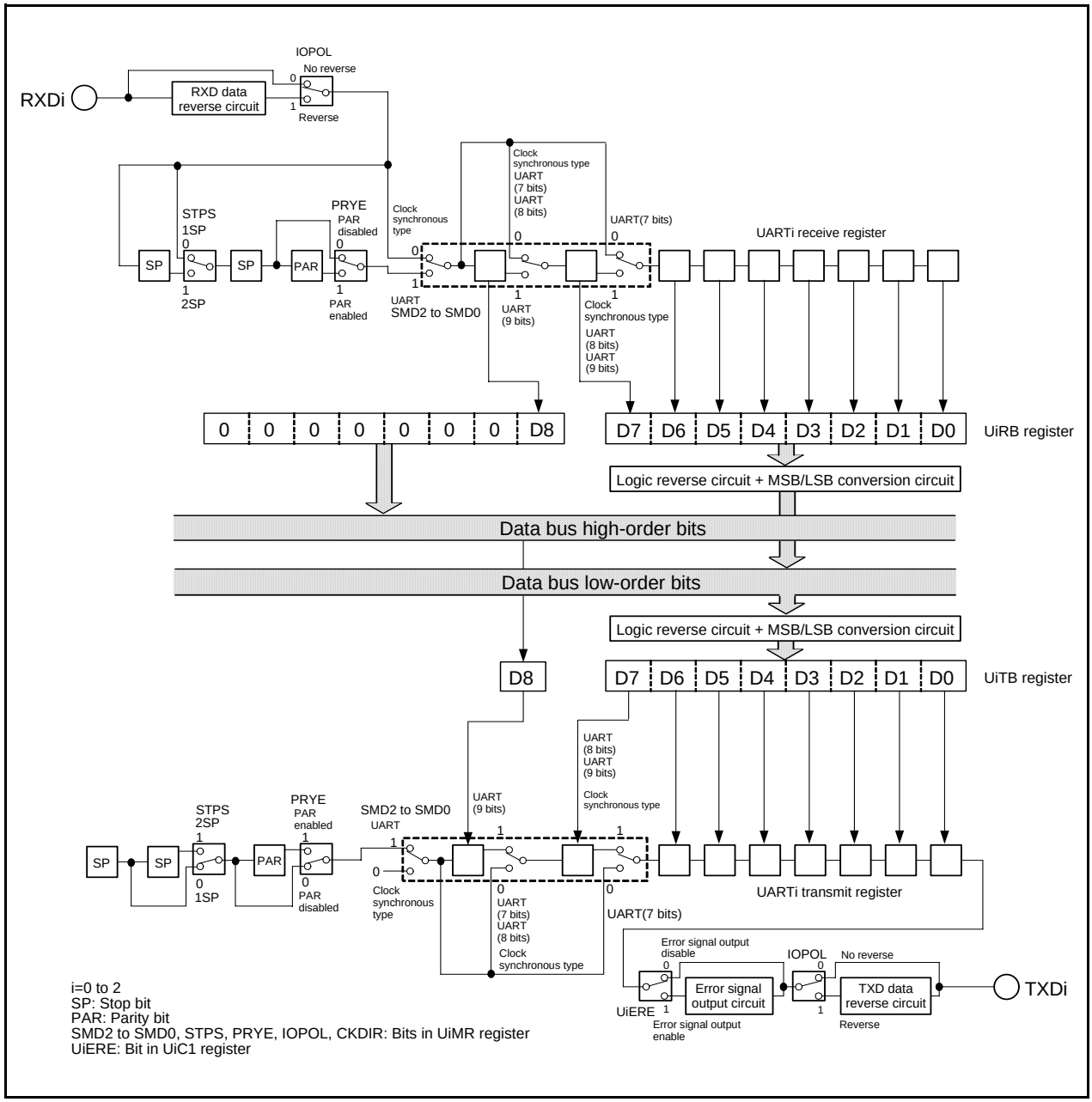
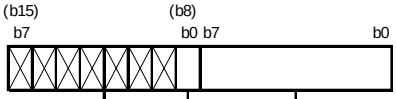


Figure 15.4 UARTi Transmit/Receive Unit

UARTi Transmit Buffer Register (i=0 to 2)(1)

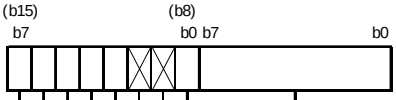


Symbol	Address	After Reset
U0TB	03A3h to 03A2h	Indeterminate
U1TB	03ABh to 03AAh	Indeterminate
U2TB	037Bh to 037Ah	Indeterminate

Function	RW
Transmit data	WO
Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.	—

NOTES :
1. Use MOV instruction to write to this register.

UARTi Receive Buffer Register (i=0 to 2)



Symbol	Address	After Reset
U0RB	03A7h to 03A6h	Indeterminate
U1RB	03AFh to 03AEh	Indeterminate
U2RB	037Fh to 037Eh	Indeterminate

Bit Symbol	Bit Name	Function	RW
— (b7-b0)	—	Receive data (D7 to D0)	RO
— (b8)	—	Receive data (D8)	RO
— (b10-b9)	Nothing is assigned. When write, set to "0". When read, their contents are "0".		—
ABT	Arbitration Lost Detecting Flag (2)	0 : Not detected 1 : Detected	RW
OER	Overrun Error Flag (1)	0 : No overrun error 1 : Overrun error found	RO
FER	Framing Error Flag (1, 3)	0 : No framing error 1 : Framing error found	RO
PER	Parity Error Flag (1, 3)	0 : No parity error 1 : Parity error found	RO
SUM	Error Sum Flag (1,3)	0 : No error 1 : Error found	RO

NOTES :
1. When the SMD2 to SMD0 bits in the UiMR register = 000b (serial interface disabled) or the RE bit in the UiC1 register = 0 (reception disabled), all of the SUM, PER, FER and OER bits are set to "0" (no error). The SUM bit is set to "0" (no error) when all of the PER, FER and OER bits = 0 (no error).
   Also, the PER and FER bits are set to "0" by reading the lower byte of the UiRB register.
2. The ABT bit is set to "0" by writing "0" in a program. (Writing "1" has no effect)
3. These error flags are disabled when the SMD2 to SMD0 bits are set to "001b" (clock synchronous serial I/O mode) or to "010b" (I^2C mode). When read, the contents are indeterminate.

Figure 15.5 UiTB and UiRB Registers

UARTi Bit Rate Generation Register (i=0 to 2) (1, 2, 3)

b7 b0

Symbol	Address	After Reset
U0BRG	03A1h	Indeterminate
U1BRG	03A9h	Indeterminate
U2BRG	0379h	Indeterminate

Function	Setting Range	RW
Assuming that set value = n, UiBRG divides the count source by n + 1	00h to FFh	WO

NOTES :
1. Write to this register while serial interface is neither transmitting nor receiving.
2. Use MOV instruction to write to this register.
3. Write to this register after setting the CLK1 to CLK0 bits in the UiC0 register.

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

b7 b6 b5 b4 b3 b2 b1 b0

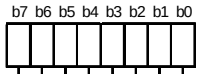
Symbol	Address	After Reset
U0MR to U2MR	03A0h, 03A8h, 0378h	00h

Bit Symbol	Bit Name	Function	RW
SMD0	Serial I/O Mode Select Bit (2)	b2 b1 b0 0 0 0 : Serial interface disabled 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : I²C mode (3) 1 0 0 : UART mode transfer data 7 bits long 1 0 1 : UART mode transfer data 8 bits long 1 1 0 : UART mode transfer data 9 bits long Do not set except above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/External Clock Select Bit	0 : Internal clock 1 : External clock (1)	RW
STPS	Stop Bit Length Select Bit	0 : 1 stop bit 1 : 2 stop bits	RW
PRY	Odd/Even Parity Select Bit	Effective when PRYE = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity Enable Bit	0 : Parity disabled 1 : Parity enabled	RW
IOPOL	TXD, RXD I/O Polarity Reverse Bit	0 : No reverse 1 : Reverse	RW

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

Figure 15.6 UiBRG and UiMR Registers

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



Symbol	Address	After Reset
U0C0 to U2C0	03A4h, 03ACh, 037Ch	00001000b

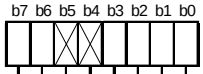
Bit Symbol	Bit Name	Function	RW
CLK0	BRG Count Source Select Bit (5)	b1 b0 0 0 : f1SIO or f2SIO is selected(6) 0 1 : f8SIO is selected 1 0 : f32SIO is selected 1 1 : Do not set to this value	RW
CLK1			RW
CRS	$\overline{\text{CTS}}/\overline{\text{RTS}}$ Function Select Bit (4)	Effective when CRD = 0 0 : $\overline{\text{CTS}}$ function is selected (1) 1 : $\overline{\text{RTS}}$ function is selected	RW
TXEPT	Transmit Register Empty Flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	RO
CRD	$\overline{\text{CTS}}/\overline{\text{RTS}}$ Disable Bit	0 : $\overline{\text{CTS}}/\overline{\text{RTS}}$ function enabled 1 : $\overline{\text{CTS}}/\overline{\text{RTS}}$ function disabled (P6_0, P6_4 and P7_3 can be used as I/O ports)	RW
NCH	Data Output Select Bit (2)	0 : TXDi/SDAi and SCLi pins are CMOS output 1 : TXDi/SDAi and SCLi pins are N-channel open-drain output	RW
CKPOL	CLK Polarity Select Bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	RW
UFORM	Transfer Format Select Bit (3)	0 : LSB first 1 : MSB first	RW

NOTES :

1. Set the corresponding port direction bit for each $\overline{\text{CTS}}$i pin to “0” (input mode).
2. TXD2/SDA2 and SCL2 are N-channel open-drain output. Cannot be set to the CMOS output. No NCH bit in U2C0 register is assigned. When write, set to “0”.
3. The UFORM bit is enabled when the SMD2 to SMD0 bits in the UiMR register are set to “001b” (clock synchronous serial I/O mode), or “101b” (UART mode, 8-bit transfer data).
   Set this bit to “1” when the SMD2 to SMD0 bits are set to “010b” (I^2C mode), and to “0” when the SMD2 to SMD0 bits are set to “100b” (UART mode, 7-bit transfer data) or “110b” (UART mode, 9-bit transfer data).
4. $\overline{\text{CTS1}}/\overline{\text{RTS1}}$ can be used when the CLKMD1 bit in the UCON register = 0 (only CLK1 output) and the RCSP bit in the UCON register = 0 ($\overline{\text{CTS0}}/\overline{\text{RTS0}}$ not separated).
5. When changing the CLK1 to CLK0 bits, set the UiBRG register.
6. Selected by PCLK1 bit in the PCLKR register.

Figure 15.7 UiC0 to UiC2 Register

UARTi Transmit/Receive Control Register 1 (i=0, 1)

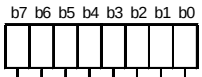


Symbol	Address	After Reset
U0C1, U1C1	03A5h, 03ADh	00XX0010b

Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit Buffer Empty Flag	0 : Data present in UiTB register 1 : No data present in UiTB register	RO
RE	Receive Enable Bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive Complete Flag	0 : No data present in UiRB register 1 : Data present in UiRB register	RO
— (b5-b4)	Nothing is assigned. When write, set to “0”. When read, these contents are indeterminate.		—
UiLCH	Data Logic Select Bit (1)	0 : No reverse 1 : Reverse	RW
UiERE	Error Signal Output Enable Bit	0 : Output disabled 1 : Output enabled	RW

NOTES :
1. The UiLCH bit is enabled when the SMD2 to SMD0 bits in the UiMR register are set to “001b” (clock synchronous serial I/O mode), “100b” (UART mode, 7-bit transfer data), or “101b” (UART mode, 8-bit transfer data).
Set this bit to “0” when the SMD2 to SMD0 bits are set to “010b” (I²C mode) or “110b” (UART mode, 9-bit transfer data).

UART2 Transmit/Receive Control Register 1



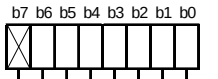
Symbol	Address	After Reset
U2C1	037Dh	00000010b

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

NOTES :
1. The U2LCH bit is enabled when the SMD2 to SMD0 bits in the U2MR register are set to “001b” (clock synchronous serial I/O mode), “100b” (UART mode, 7-bit transfer data), or “101b” (UART mode, 8-bit transfer data).
Set this bit to “0” when the SMD2 to SMD0 bits are set to “010b” (I²C mode) or “110b” (UART mode, 9-bit transfer data).

Figure 15.8 U0C1 to U2C1 Registers

UART Transmit/Receive Control Register 2

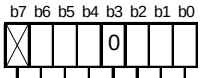


Symbol	Address	After Reset
UCON	03B0h	X0000000b

Bit Symbol	Bit Name	Function	RW
U0IRS	UART0 Transmit Interrupt Factor Select Bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U1IRS	UART1 Transmit Interrupt Factor Select Bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U0RRM	UART0 Continuous Receive Mode Enable Bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	RW
U1RRM	UART1 Continuous Receive Mode Enable Bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
CLKMD0	UART1 CLK/CLKS Select Bit 0	Effective when CLKMD1 = 1 0 : Clock output from CLK1 1 : Clock output from CLKS1	RW
CLKMD1	UART1 CLK/CLKS Select Bit 1 (1)	0 : CLK output is only CLK1 1 : Transfer clock output from multiple pins function selected	RW
RCSP	Separate UART0 $\overline{CTS}/\overline{RTS}$ Bit	0 : $\overline{CTS}/\overline{RTS}$ shared pin 1 : $\overline{CTS}/\overline{RTS}$ separated (CTS0 supplied from the P6_4 pin)	RW
— (b7)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—

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

UARTi Special Mode Register (i=0 to 2)



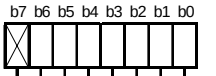
Symbol	Address	After Reset
U0SMR to U2SMR	036Fh, 0373h, 0377h	X0000000b

Bit Symbol	Bit Name	Function	RW
IICM	I^2C Mode Select Bit	0 : Other than I^2C mode 1 : I^2C mode	RW
ABC	Arbitration Lost Detecting Flag Control Bit	0 : Update per bit 1 : Update per byte	RW
BBS	Bus Busy Flag	0 : STOP condition detected 1 : START condition detected (busy)	RW(1)
— (b3)	Reserved Bit	Set to “0”	RW
ABSCS	Bus Collision Detect Sampling Clock Select Bit	0 : Rising edge of transfer clock 1 : Underflow signal of timer Aj (2)	RW
ACSE	Auto Clear Function Select Bit of Transmit Enable Bit	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	RW
SSS	Transmit Start Condition Select Bit	0 : Not synchronized to RXDi 1 : Synchronized to RXDi (3)	RW
— (b7)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—

NOTES :
1. The BBS bit is set to “0” by writing “0” in a program (Writing “1” has no effect).
2. Underflow signal of timer A0 in UART2.
3. When a transfer begins, the SSS bit is set to “0” (Not synchronized to RXDi).

Figure 15.9 UCON and UiSMR Registers

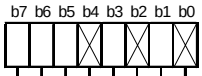
UARTi Special Mode Register 2 (i=0 to 2)



Symbol	Address	After Reset
U0SMR2 to U2SMR2	036Eh, 0372h, 0376h	X0000000b

Bit Symbol	Bit Name	Function	RW
IICM2	I²C Mode Select Bit 2	See **Table 15.13 I²C Mode Functions**	RW
CSC	Clock-Synchronous Bit	0 : Disabled 1 : Enabled	RW
SWC	SCL Wait Output Bit	0 : Disabled 1 : Enabled	RW
ALS	SDA Output Stop Bit	0 : Disabled 1 : Enabled	RW
STAC	UARTi Initialization Bit	0 : Disabled 1 : Enabled	RW
SWC2	SCL Wait Output Bit 2	0: Transfer clock 1: “L” output	RW
SDHI	SDA Output Disable Bit	0: Enabled 1: Disabled (high-impedance)	RW
— (b7)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—

UARTi special mode register 3 (i=0 to 2)



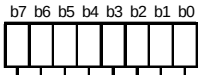
Symbol	Address	After Reset
U0SMR3 to U2SMR3	036Dh, 0371h, 0375h	000X0X0Xb

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

NOTES :
1. The DL2 to DL0 bits are used to generate a delay in SDAi output by digital means during I²C mode. In other than I²C mode, set these bits to “000b” (no delay).
2. The amount of delay varies with the load on SCLi and SDAi pins. Also, when using an external clock, the amount of delay increases by about 100 ns.

Figure 15.10 UiSMR2 and UiSMR3 Registers

UARTi Special Mode Register 4 (i=0 to 2)



Symbol	Address	After Reset
U0SMR4 to U2SMR4	036Ch, 0370h, 0374h	00h

Bit Symbol	Bit Name	Function	RW
STAREQ	Start Condition Generate Bit (1)	0 : Clear 1 : Start	RW
RSTAREQ	Restart Condition Generate Bit (1)	0 : Clear 1 : Start	RW
STPREQ	Stop Condition Generate Bit (1)	0 : Clear 1 : Start	RW
STSPSEL	SCL,SDA Output Select Bit	0 : Start and stop conditions not output 1 : Start and stop conditions output	RW
ACKD	ACK Data Bit	0 : ACK 1 : NACK	RW
ACKC	ACK Data Output Enable Bit	0 : Serial interface data output 1 : ACK data output	RW
SCLHI	SCL Output Stop Enable Bit	0 : Disabled 1 : Enabled	RW
SWC9	SCL Wait Bit 3	0 : SCL “L” hold disabled 1 : SCL “L” hold enabled	RW

NOTES :
1. Set to “0” when each condition is generated.

Figure 15.11 UiSMR4 Register

15.1.1 Clock Synchronous Serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 15.1 lists the Clock Synchronous Serial I/O Mode Specifications. Table 15.2 lists the Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode.

Table 15.1 Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	• CKDIR bit in the UiMR(i=0 to 2) register = 0 (internal clock) : fj/ (2(n+1)) fj = f1SIO, f2SIO, f8SIO, f32SIO n: Setting value of UiBRG register 00h to FFh • CKDIR bit = 1 (external clock) : Input from CLKi pin
Transmission, Reception Control	Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}$/$\overline{RTS}$ function disable
Transmission Start Condition	Before transmission can start, the following requirements must be met (1) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in UiTB register) • If $\overline{CTS}$ function is selected, input on the $\overline{CTSi}$ pin = L
Reception Start Condition	Before reception can start, the following requirements must be met (1) • The RE bit in the UiC1 register = 1 (reception enabled) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected • The UiIRS bit (3) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit =1 (transfer completed): when the serial interface finished sending data from the UARTi transmit register For reception • When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error Detection	Overrun error (2) This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the 7th bit of the next data
Select Function	• CLK polarity selection Transfer data input/output can be 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 UiRB register • Switching serial data logic This function reverses the logic value of the transmit/receive data • Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set • Separate $\overline{CTS}$/$\overline{RTS}$ pins (UART0) $\overline{CTS0}$ and $\overline{RTS0}$ are input/output from separate pins

NOTES:
1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. If an overrun error occurs, bits 8 to 0 in the UiRB register are undefined. The IR bit in the SiRIC register does not change to “1” (interrupt requested).
3. The U0IRS and U1IRS bits respectively are the bits 0 and 1 in the UCON register; the U2IRS bit is the bit 4 in the U2C1 register.

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

Register	Bit	Function
UiTB (3)	0 to 7	Set transmission data
UiRB (3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a bit rate
UiMR (3)	SMD2 to SMD0	Set to “001b”
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to “0”
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ function
	NCH	Select TXDi pin output mode (2)
	CKPOL	Select the transfer clock polarity
	UFORM	Select the LSB first or MSB first
UiC1	TE	Set this bit to “1” to enable transmission/reception
	TI	Transmit buffer empty flag
	RE	Set this bit to “1” to enable reception
	RI	Reception complete flag
	U2IRS (1)	Select the source of UART2 transmit interrupt
	U2RRM (1)	Set this bit to “1” to use continuous receive mode
	UiLCH	Set this bit to “1” to use inverted data logic
	UiERE	Set to “0”
UiSMR	0 to 7	Set to “0”
UiSMR2	0 to 7	Set to “0”
UiSMR3	0 to 2	Set to “0”
	NODC	Select clock output mode
	4 to 7	Set to “0”
UiSMR4	0 to 7	Set to “0”
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set this bit to “1” to use continuous receive mode
	CLKMD0	Select the transfer clock output pin when CLKMD1 = 1
	CLKMD1	Set this bit to “1” to output UART1 transfer clock from two pins
	RCSP	Set this bit to “1” to accept as input the $\overline{\text{CTS0}}$ signal of the UART0 from the P6_4 pin
	7	Set to “0”

NOTES:
1. Set the bit 4 and bit 5 in the U0C1 and U1C1 register to “0”. The U0IRS, U1IRS, U0RRM and U1RRM bits in the UCON register.
2. TXD2 pin is N channel open-drain output. Set the NCH bit in the U2C0 register to “0”.
3. Not all register bits are described above. Set those bits to “0” when writing to the registers in clock synchronous serial I/O mode.

i=0 to 2

Table 15.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Table 15.3 shows pin functions for the case where the multiple transfer clock output pin select function is deselected. Table 15.4 lists the P6_4 Pin Functions during clock synchronous serial I/O mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TXDi pin outputs an "H" (If the N-channel open-drain output is selected, this pin is in a high-impedance state).

Table 15.3 Pin Functions (when not select multiple transfer clock output pin function)

Pin Name	Function	Method of Selection
TXDi (i = 0 to 2) (P6_3, P6_7, P7_0)	Serial Data Output	(Outputs dummy data when performing reception only)
RXDi (P6_2, P6_6, P7_1)	Serial Data Input	PD6_2 bit and PD6_6 bit in the PD6 register = 0, PD7_1 bit in the PD7 register = 0 (Can be used as an input port when performing transmission only)
CLKi (P6_1, P6_5, P7_2)	Transfer Clock Output	CKDIR bit in the UiMR register = 0
	Transfer Clock Input	CKDIR bit = 1 PD6_1 bit and PD6_5 bit in the PD6 register = 0, PD7_2 bit in the PD7 register = 0
$\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ (P6_0, P6_4, P7_3)	$\overline{\text{CTS}}$ Input	CRD bit in the UiC0 register = 0 CRS bit in the UiC0 register = 0 PD6_0 and PD6_4 bit in the PD6 register = 0, PD7_3 bit in the PD7 register = 0
	$\overline{\text{RTS}}$ Output	CRD bit = 0 CRS bit = 1
	I/O Port	CRD bit = 1

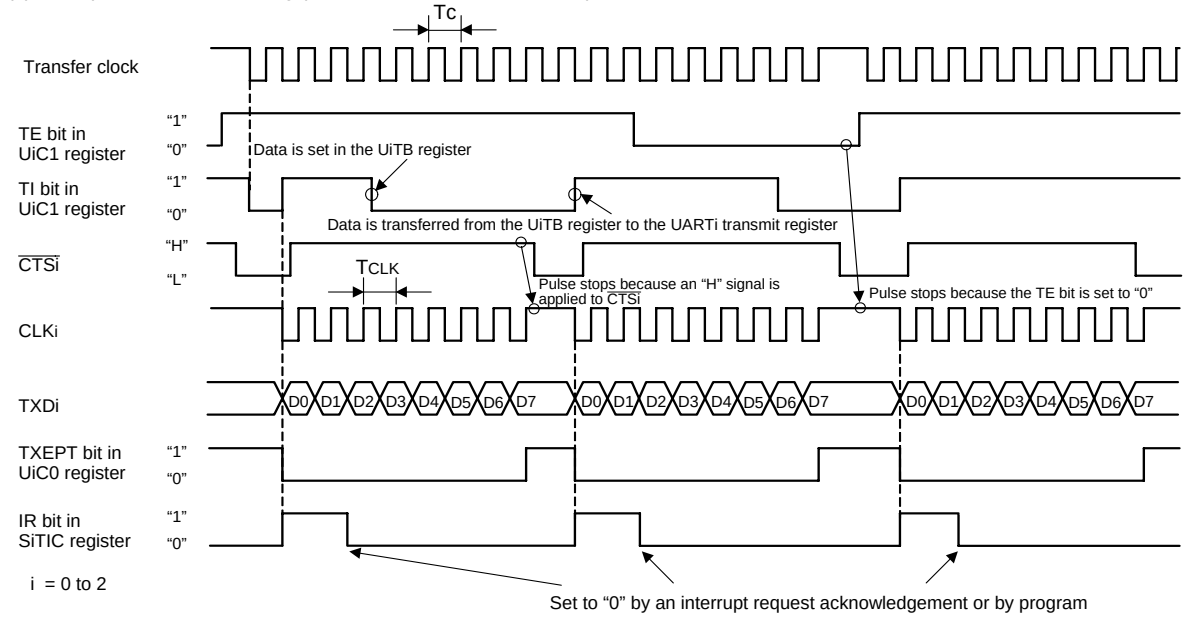
Table 15.4 P6_4 Pin Functions

Pin Function	Bit Set Value					
	U1C0 Register		UCON Register			PD6 Register
	CRD	CRS	RCSP	CLKMD1	CLKMD0	PD6_4
P6_4	1	–	0	0	–	Input: 0, Output: 1
$\overline{\text{CTS1}}$	0	0	0	0	–	0
$\overline{\text{RTS1}}$	0	1	0	0	–	–
$\overline{\text{CTS0}}$ (1)	0	0	1	0	–	0
CLKS1	–	–	–	1 (2)	1	–

– : "0" or "1"

NOTES:
1. In addition to this, set the CRD bit in the U0C0 register to "0" ($\overline{\text{CTS0}}$/$\overline{\text{RTS0}}$ enabled) and the CRS bit in the U0C0 register to "1" ($\overline{\text{RTS0}}$ selected).
2. When the CLKMD1 bit = 1 and the CLKMD0 bit = 0, the following logic levels are output:
   - •High if the CLKPOL bit in the U1C0 register = 0
   - •Low if the CLKPOL bit = 1

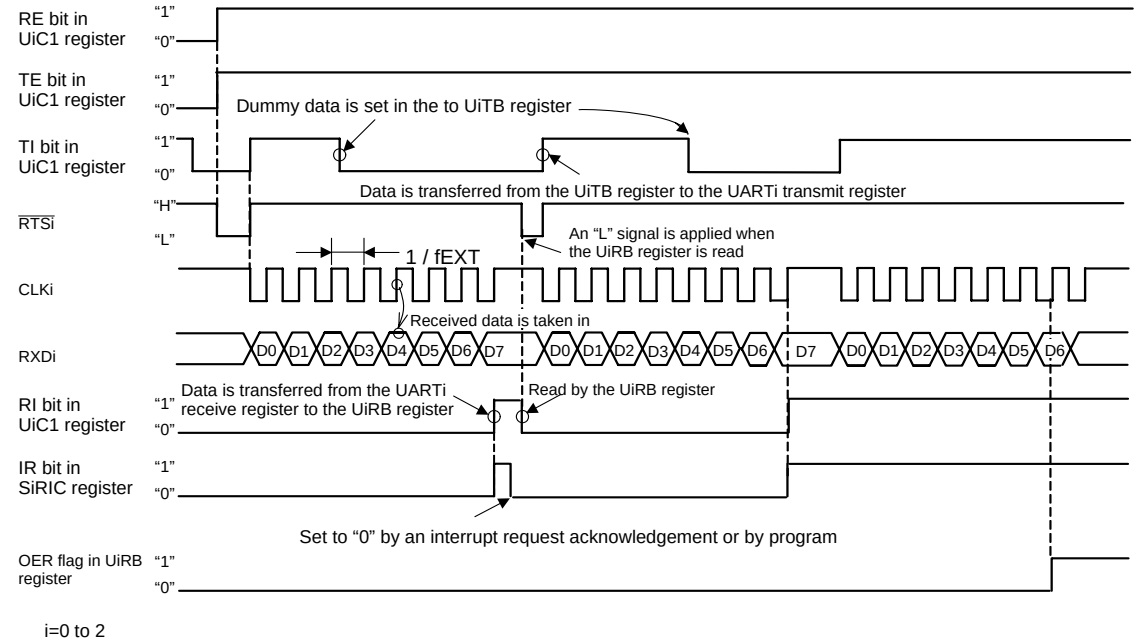
(1) Example of Transmit Timing (when internal clock is selected)



The above timing diagram applies to the case where the register bits are set as follows:
- CKDIR bit in UiMR register = 0 (internal clock)
- CRD bit in UiC0 register = 0 (CTS/RTS enabled), CRS bit = 0 (CTS selected)
- CKPOL bit in UiC0 register = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- UiIRS bit = 0 (an interrupt request occurs when the transmit buffer becomes empty):
  U0IRS bit is bit 0 in UCON register
  U1IRS bit is bit 1 in UCON register
  U2IRS bit is bit 4 in U2C1 register

TC = TCLK = 2(n + 1) / fj
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
n: value set to UiBRG register

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



The above timing diagram applies to the case where the register bits are set as follows:
- CKDIR bit in UiMR register = 1 (external clock)
- CRD bit in UiC0 register = 0 (CTS/RTS enabled), CRS bit = 1 (RTS selected)
- CKPOL bit in UiC0 register = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

fEXT: frequency of external clock

Make sure the following conditions are met when input to the CLKi pin before receiving data is high:
- TE bit in UiC0 register = 1 (transmit enabled)
- RE bit in UiC0 register = 1 (receive enabled)
- Write dummy data to the UiTB register

Figure 15.12 Transmit and Receive Operation

15.1.1.1 Counter Measure for Communication Error Occurs

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

• Resetting the UiRB register (i=0 to 2)

(1) Set the RE bit in the UiC1 register to "0" (reception disabled)
(2) Set the SMD2 to SMD0 bits in the UiMR register to "000b" (Serial interface disabled)
(3) Set the SMD2 to SMD0 bits in the UiMR register to "001b" (Clock synchronous serial I/O mode)
(4) Set the RE bit in the UiC1 register to "1" (reception enabled)

• Resetting the UiTB register (i=0 to 2)

(1) Set the SMD2 to SMD0 bits in the UiMR register "000b" (Serial interface disabled)
(2) Set the SMD2 to SMD0 bits in the UiMR register "001b" (Clock synchronous serial I/O mode)
(3) "1" is written to RE bit in the UiC1 register (transmission enabled), regardless of the TE bit in the UiCi register

15.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register (i = 0 to 2) to select the transfer clock polarity. Figure 15.13 shows the Transfer Clock Polarity.

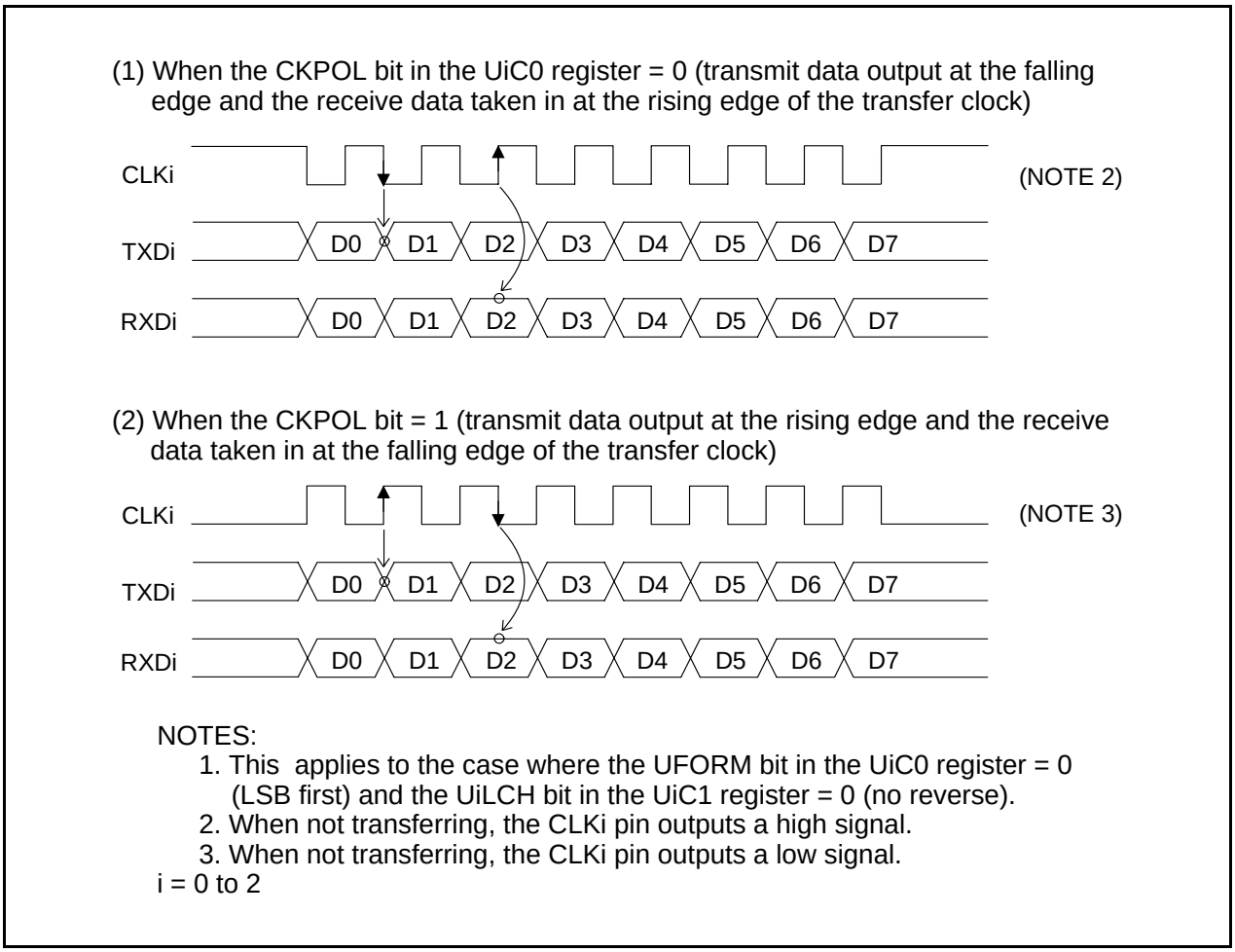


Figure 15.13 Transfer Clock Polarity

15.1.1.3 LSB First/MSB First Select Function

Use the UFORM bit in the UiC0 register (i = 0 to 2) to select the transfer format. Figure 15.14 shows the Transfer Format.

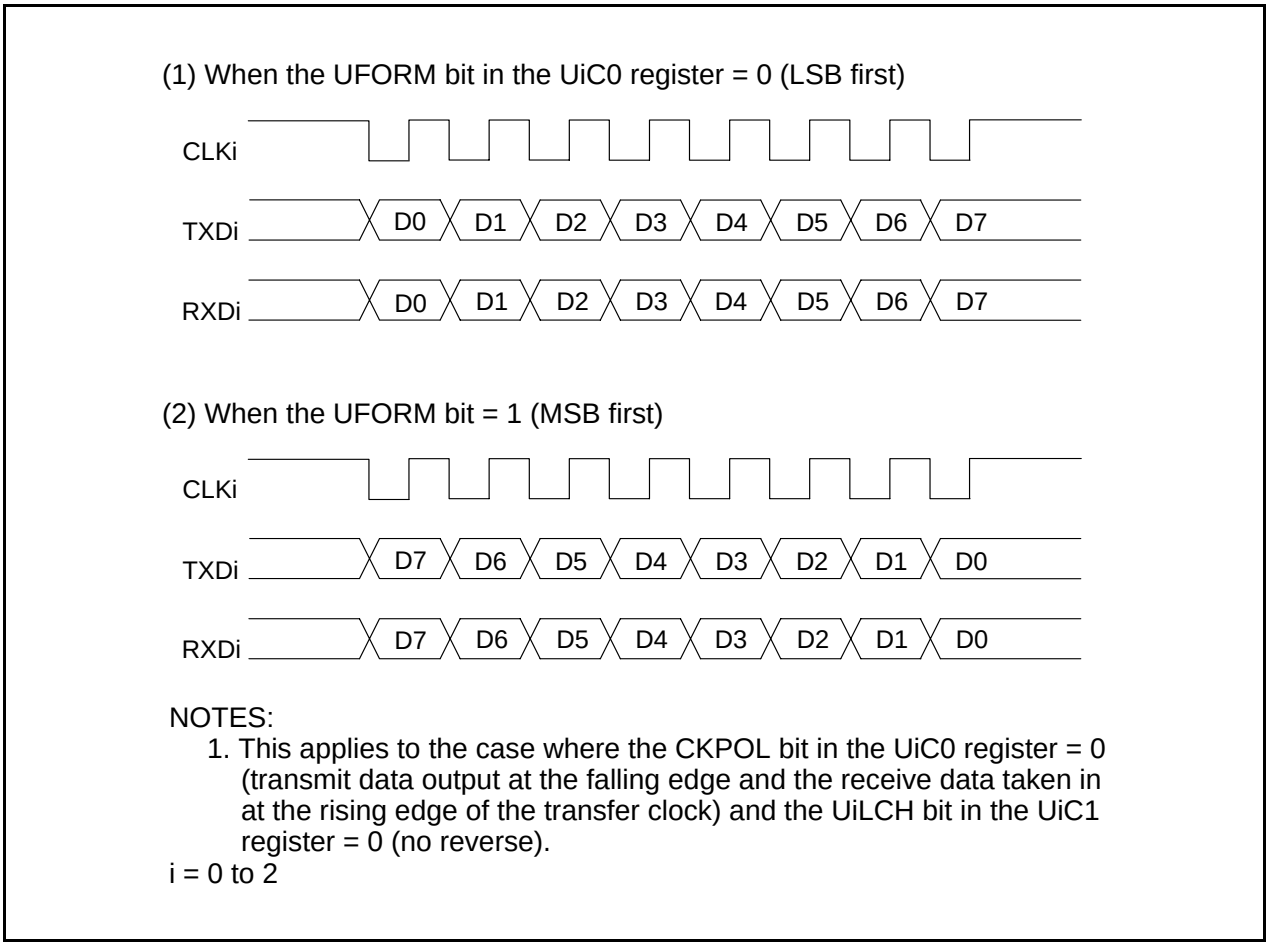


Figure 15.14 Transfer Format

15.1.1.4 Continuous Receive Mode

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

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

15.1.1.5 Serial Data Logic Switching Function

When the UiLCH bit in the UiC1 register (i = 0 to 2) = 1 (reverse), the data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 15.15 shows Serial Data Logic Switching.

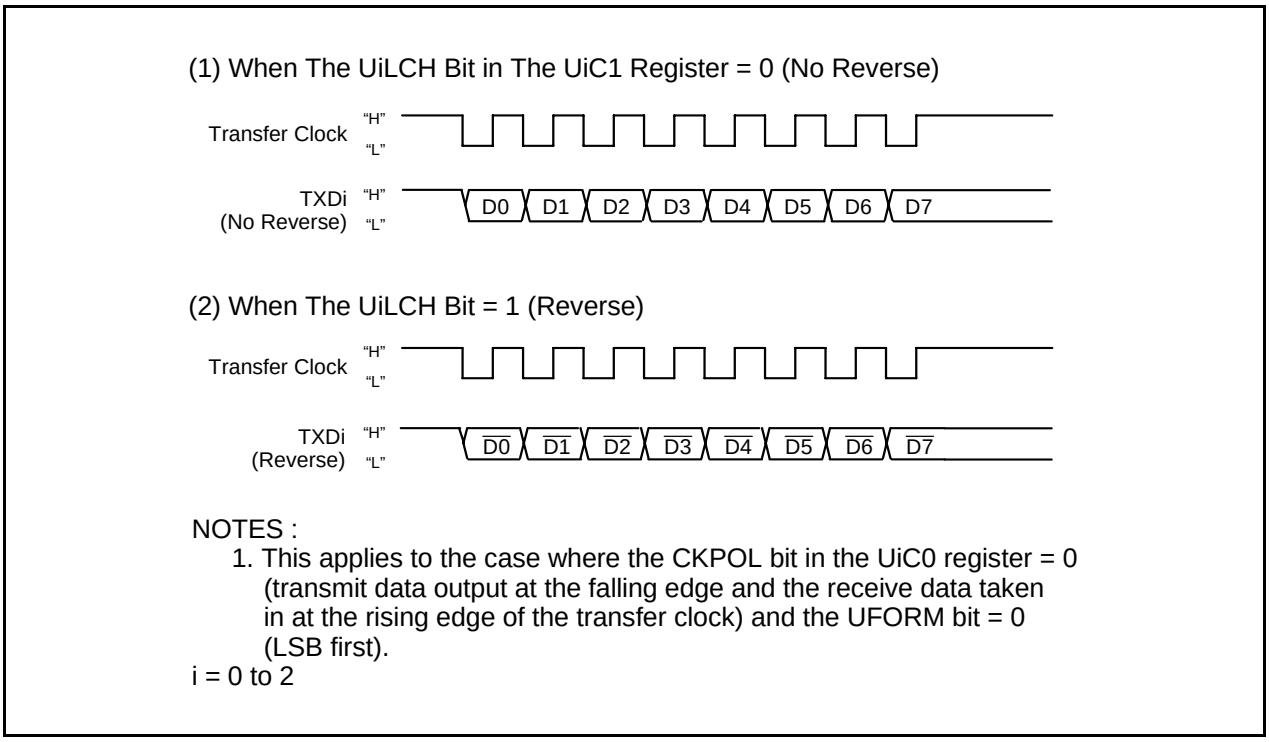


Figure 15.15 Serial Data Logic Switching

15.1.1.6 Transfer Clock Output From Multiple Pins (UART1)

Use the CLKMD1 to CLKMD0 bits in the UCON register to select one of the two transfer clock output pins (see Figure 15.16). This function can be used when the selected transfer clock for UART1 is an internal clock.

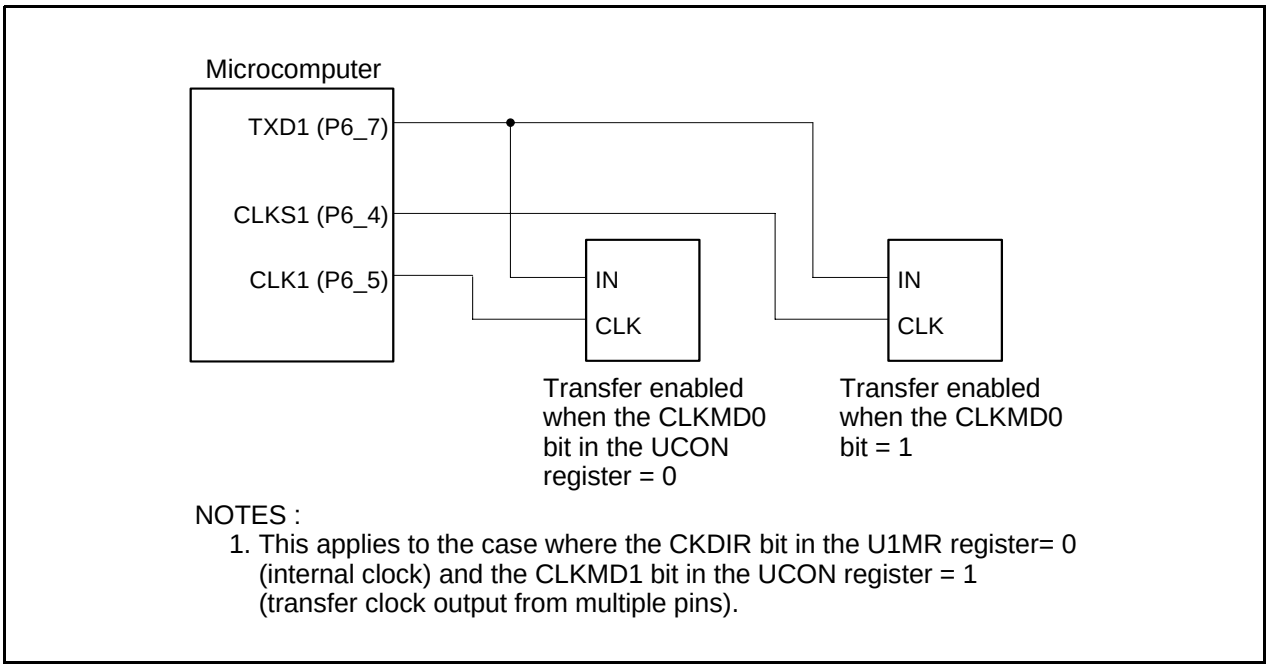


Figure 15.16 Transfer Clock Output from Multiple Pins

15.1.1.7 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Function

When the $\overline{\text{CTS}}$ function is used transmit and receive operation start when "L" is applied to the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ (i=0 to 2) pin. Transmit and receive operation begins when the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin is held "L". If the "L" signal is switched to "H" during a transmit or receive operation, the operation stops before the next data.
When the $\overline{\text{RTS}}$ function is used, the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin outputs on "L" signal when the microcomputer is ready to receive. The output level becomes "H" on the first falling edge of the CLKi pin.

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

15.1.1.8 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function (UART0)

This function separates $\overline{\text{CTS0}}/\overline{\text{RTS0}}$, outputs $\overline{\text{RTS0}}$ from the P6_0 pin, and accepts as input the $\overline{\text{CTS0}}$ from the P6_4 pin. To use this function, set the register bits as shown below.

- CRD bit in U0C0 register = 0 (enable $\overline{\text{CTS}}/\overline{\text{RTS}}$ of UART0)
- CRS bit in U0C0 register = 1 (output $\overline{\text{RTS}}$ of UART0)
- CRD bit in U1C0 register = 0 (enable $\overline{\text{CTS}}/\overline{\text{RTS}}$ of UART1)
- CRS bit in U1C0 register = 0 (input $\overline{\text{CTS}}$ of UART1)
- RCSP bit in UCON register = 1 (inputs $\overline{\text{CTS0}}$ from the P6_4 pin)
- CLKMD1 bit in UCON register = 0 (CLKS1 not used)

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

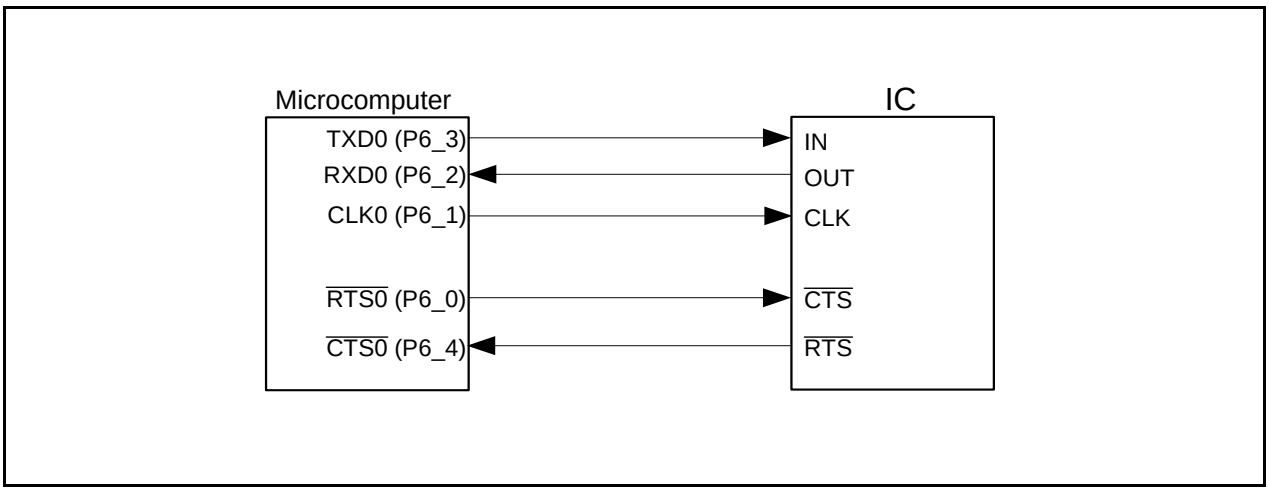


Figure 15.17 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

15.1.2 Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows transmitting and receiving data after setting the desired bit rate and transfer data format. Table 15.5 lists the UART Mode Specifications.

Table 15.5 UART Mode Specifications

Item	Specification
Transfer Data Format	• Character bit (transfer data): Selectable from 7, 8 or 9 bits • Start bit: 1 bit • Parity bit: Selectable from odd, even, or none • Stop bit: Selectable from 1 or 2 bits
Transfer Clock	• CKDIR bit in the UiMR(i=0 to 2) register = 0 (internal clock) : fj/ (16(n+1)) fj = f1SIO, f2SIO, f8SIO, f32SIO n: Setting value of UiBRG register 00h to FFh • CKDIR bit = 1 (external clock) : fEXT/(16(n+1)) fEXT: Input from CLKi pin n :Setting value of UiBRG register 00h to FFh
Transmission, Reception Control	Selectable from $\overline{\text{CTS}}$ function, $\overline{\text{RTS}}$ function or $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function disable
Transmission Start Condition	Before transmission can start, the following requirements must be met • The TE bit in the UiC1 register= 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in UiTB register) • If $\overline{\text{CTS}}$ function is selected, input on the $\overline{\text{CTSi}}$ pin = L
Reception Start Condition	Before reception can start, the following requirements must be met • The RE bit in the UiC1 register = 1 (reception enabled) • Start bit detection
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected • The UiIRS bit (2) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit =1 (transfer completed): when the serial interface finished sending data from the UARTi transmit register For reception • When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error Detection	• Overrun error (1) This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the bit one before the last stop bit of the next data • Framing error (3) This error occurs when the number of stop bits set is not detected • Parity error (3) This error occurs when if parity is enabled, the number of “1” in parity and character bits does not match the number of “1” set • Error sum flag This flag is set to “1” when any of the overrun, framing or parity errors occur
Select Function	• LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Serial data logic switch This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. • TXD, RXD I/O polarity switch This function reverses the polarities of the TXD pin output and RXD pin input. The logic levels of all I/O data is reversed. • Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins (UART0) $\overline{\text{CTS0}}$ and $\overline{\text{RTS0}}$ are input/output from separate pins

NOTES:
1. If an overrun error occurs, bits 8 to 0 in the UiRB register are undefined. The IR bit in the SiRIC register does not change to "1" (interrupt requested).
2. The U0IRS and U1IRS bits are bits 0 and 1 in the UCON register. The U2IRS bit is bit 4 in the U2C1 register.
3. The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UARTi receive register to the UiRB register.

Table 15.6 Registers to Be Used and Settings in UART Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data (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 “100b” when transfer data is 7 bits long
		Set these bits to “101b” when transfer data is 8 bits long
		Set these bits to “110b” when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
	IOPOL	Select the TXD/RXD input/output polarity
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	CRS	Select $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ function
	NCH	Select TXDi pin output mode (3)
	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
	U2IRS (2)	Select the source of UART2 transmit interrupt
	U2RRM (2)	Set to “0”
	UiLCH	Set this bit to “1” to use inverted data logic
	UiERE	Set to “0”
UiSMR	0 to 7	Set to “0”
UiSMR2	0 to 7	Set to “0”
UiSMR3	0 to 7	Set to “0”
UiSMR4	0 to 7	Set to “0”
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to “0”
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1	Set to “0”
	RCSP	Set this bit to “1” to accept as input $\overline{\text{CTS0}}$ signal of UART0 from the P6_4 pin
	7	Set to “0”

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. Set the bit 4 to bit 5 in the U0C1 and U1C1 registers to “0”. The U0IRS, U1IRS, U0RRM and U1RRM bits are included in the UCON register.
3. TXD2 pin is N channel open-drain output. Set the NCH bit in the U2C0 register to “0”.

i=0 to 2

Table 15.7 lists the functions of the input/output pins during UART mode. Table 15.8 lists the P6_4 Pin Functions. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TXDi pin outputs an "H" (If the N-channel open-drain output is selected, this pin is in a high-impedance state).

Table 15.7 I/O Pin Functions

Pin Name	Function	Method of Selection
TXDi (i = 0 to 2) (P6_3, P6_7, P7_0)	Serial Data Output	("H" outputs when performing reception only)
RXDi (P6_2, P6_6, P7_1)	Serial Data Input	PD6_2 bit and PD6_6 bit in the PD6 register = 0, PD7_1 bit in the PD7 register = 0 (Can be used as an input port when performing transmission only)
CLKi (P6_1, P6_5, P7_2)	Input/Output Port	CKDIR bit in the UiMR register = 0
	Transfer Clock Input	CKDIR bit = 1 PD6_1 bit and PD6_5 bit in the PD6 register = 0, PD7_2 bit in the PD7 register = 0
$\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ (P6_0, P6_4, P7_3)	$\overline{\text{CTS}}$ Input	CRD bit in the UiC0 register = 0 CRS bit in the UiC0 register = 0 PD6_0 bit and PD6_4 bit in the PD6 register = 0, PD7_3 bit in the PD7 register = 0
	$\overline{\text{RTS}}$ Output	CRD bit = 0 CRS bit = 1
	Input/Output Port	CRD bit = 1

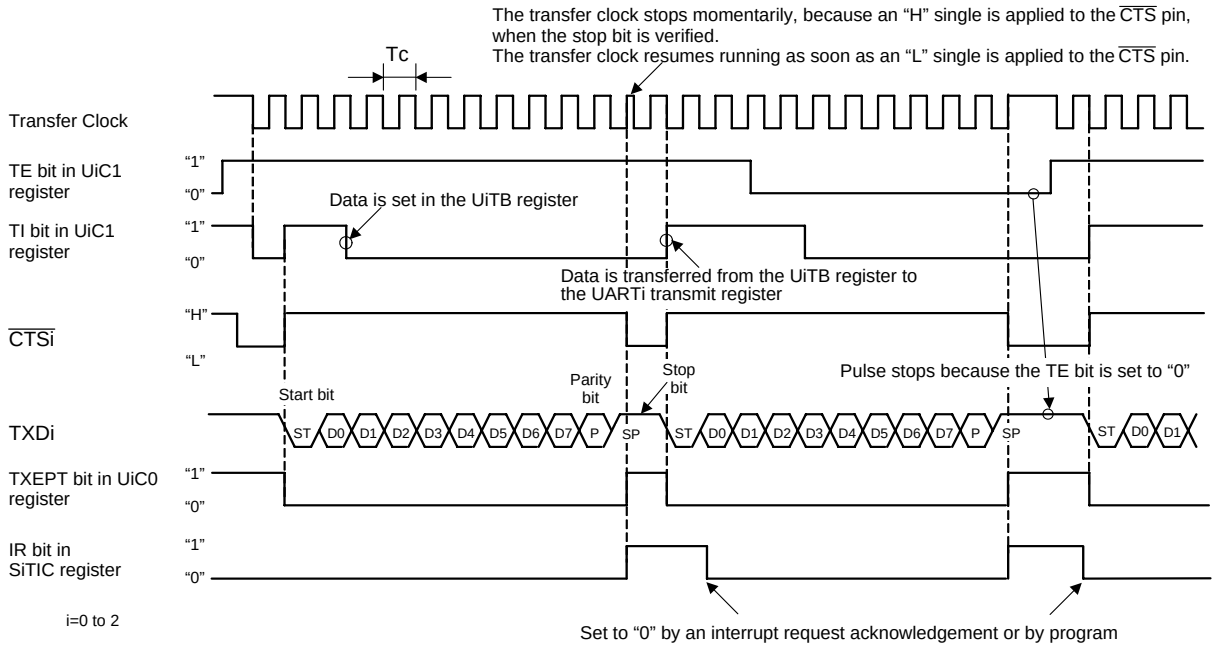
Table 15.8 P6_4 Pin Functions

Pin Function	Bit Set Value				
	U1C0 Register		UCON Register		PD6 Register
	CRD	CRS	RCSP	CLKMD1	PD6_4
P6_4	1	–	0	0	Input: 0, Output: 1
$\overline{\text{CTS1}}$	0	0	0	0	0
$\overline{\text{RTS1}}$	0	1	0	0	–
$\overline{\text{CTS0}}$ (1)	0	0	1	0	0

– : "0" or "1"

NOTES:
1. In addition to this, set the CRD bit in the U0C0 register to "0" ($\overline{\text{CTS0}}$/$\overline{\text{RTS0}}$ enabled) and the CRS bit in the U0C0 register to "1" ($\overline{\text{RTS0}}$ selected).

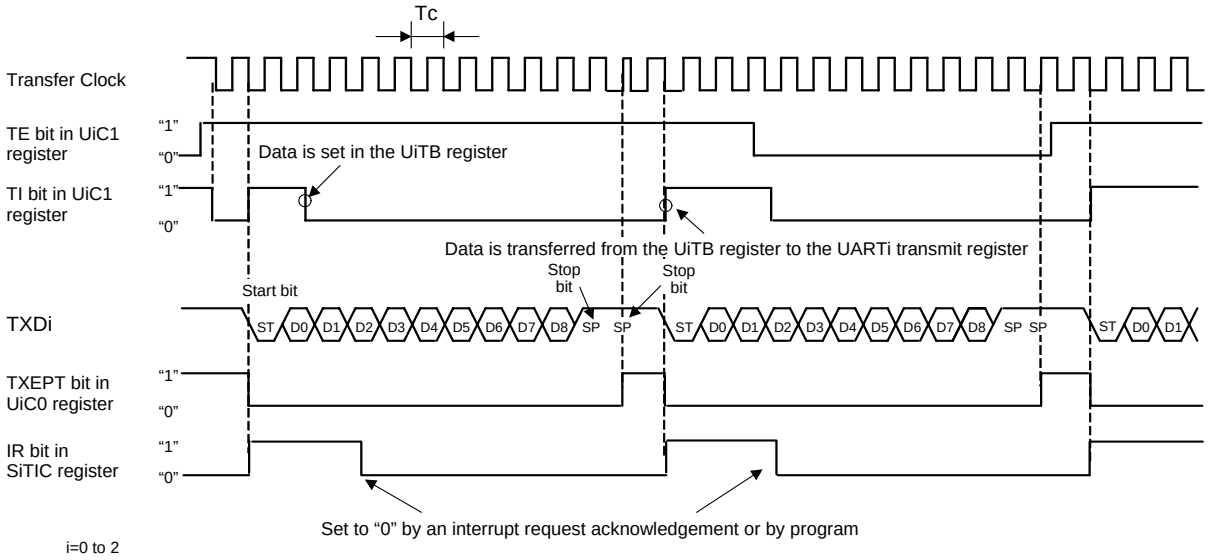
(1) 8-bit Data Transmit Timing (with a parity and 1 stop bit)



The above timing diagram applies to the case where the register bits are set as follows:
- PRYE bit in UiMR register = 1 (parity enabled)
- STPS bit in UiMR register = 0 (1 stop bit)
- CRD bit in UiC0 register = 0 (CTS/RTS enabled) and CRS bit = 0 (CTS selected)
- UiIRS bit = 1 (an interrupt request occurs when transmit completed):
  - U0IRS bit is bit 0 in UCON register
  - U1IRS bit is bit 1 in UCON register
  - U2IRS bit is bit 4 in U2C1 register

TC = 16 (n + 1) / fj or 16 (n + 1) / fEXT
fj : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of UiBRG count source (external clock)
n : value set to UiBRG

(1) 9-bit Data Transmit Timing (with no parity and 2 stop bits)

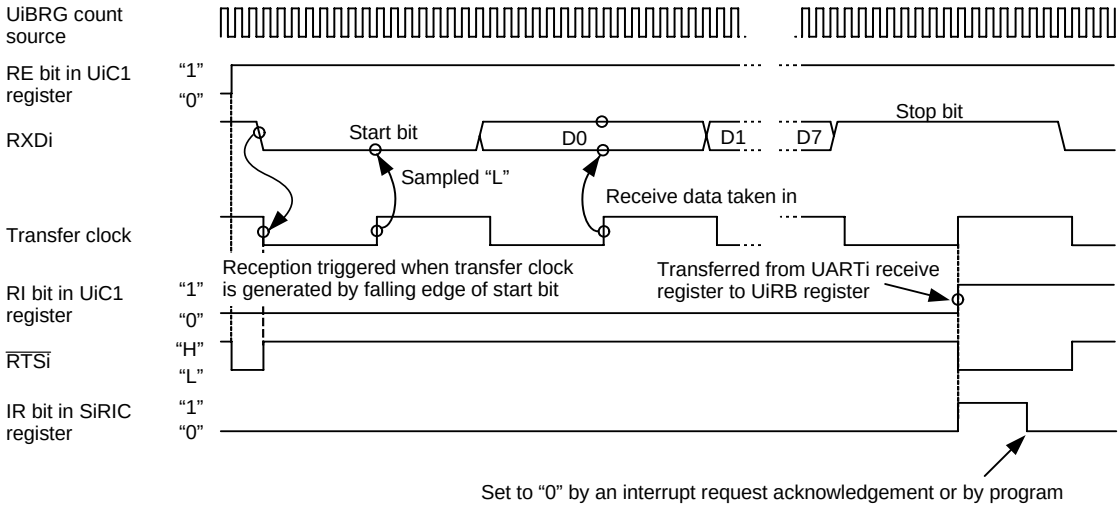


The above timing diagram applies to the case where the register bits are set as follows:
- PRYE bit in UiMR register = 0 (parity disabled)
- STPS bit in UiMR register = 1 (2 stop bits)
- CRD bit in UiC0 register = 1 (CTS/RTS disabled)
- UiIRS bit = 0 (an interrupt request occurs when transmit buffer becomes empty):
  - U0IRS bit is bit 0 in UCON register
  - U1IRS bit is bit 1 in UCON register
  - U2IRS bit is bit 4 in U2C1 register

TC = 16 (n + 1) / fj or 16 (n + 1) / fEXT
fj : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of UiBRG count source (external clock)
n : value set to UiBRG

Figure 15.18 Transmit Operation

• Example of Receive Timing when Transfer Data is 8 Bits Long (parity disabled, one stop bit)



The above timing diagram applies to the case where the register bits are set as follows:
- PRYE bit in UiMR register = 0 (parity disabled)
- STPS bit in UiMR register = 0 (1 stop bit)
- CRD bit in UiC0 register = 0 ($\overline{CTSi}$/$\overline{RTSi}$ enabled) and CRS bit = 1 ($\overline{RTSi}$ selected)

i = 0 to 2

Figure 15.19 Receive Operation

15.1.2.1 Bit Rate

In UART mode, the frequency set by the UiBRG register (i=0 to 2) divided by 16 become the bit rates. Table 15.9 lists Example of Bit Rates and Settings.

Table 15.9 Example of Bit Rates and Settings

Bit Rate (bps)	Count Source of UiBRG	Peripheral Function Clock : 16MHz	
		Set Value of UiBRG : n	Bit Rate (bps)
1200	f8	103 (67h)	1202
2400	f8	51 (33h)	2404
4800	f8	25 (19h)	4808
9600	f1	103 (67h)	9615
14400	f1	68 (44h)	14493
19200	f1	51 (33h)	19231
28800	f1	34 (22h)	28571
31250	f1	31 (1Fh)	31250
38400	f1	25 (19h)	38462
51200	f1	19 (13h)	50000

15.1.2.2 Counter Measure for Communication Error Occurs

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

• Resetting the UiRB register (i=0 to 2)

(1) Set the RE bit in the UiC1 register to “0” (reception disabled)

(2) Set the RE bit in the UiC1 register to “1” (reception enabled)

• Resetting the UiTB register (i=0 to 2)

(1) Set the SMD2 to SMD0 bits in the UiMR register “000b” (Serial interface disabled)

(2) Set the SMD2 to SMD0 bits in the UiMR register “001b”, “101b”, “110b”.

(3) “1” is written to RE bit in the UiC1 register (transmission enabled), regardless of the TE bit in the UiCi register

15.1.2.3 LSB First/MSB First Select Function

As shown in Figure 15.20, use the UFORM bit in the UiC0 register to select the transfer format. This function is valid when transfer data is 8 bits long.

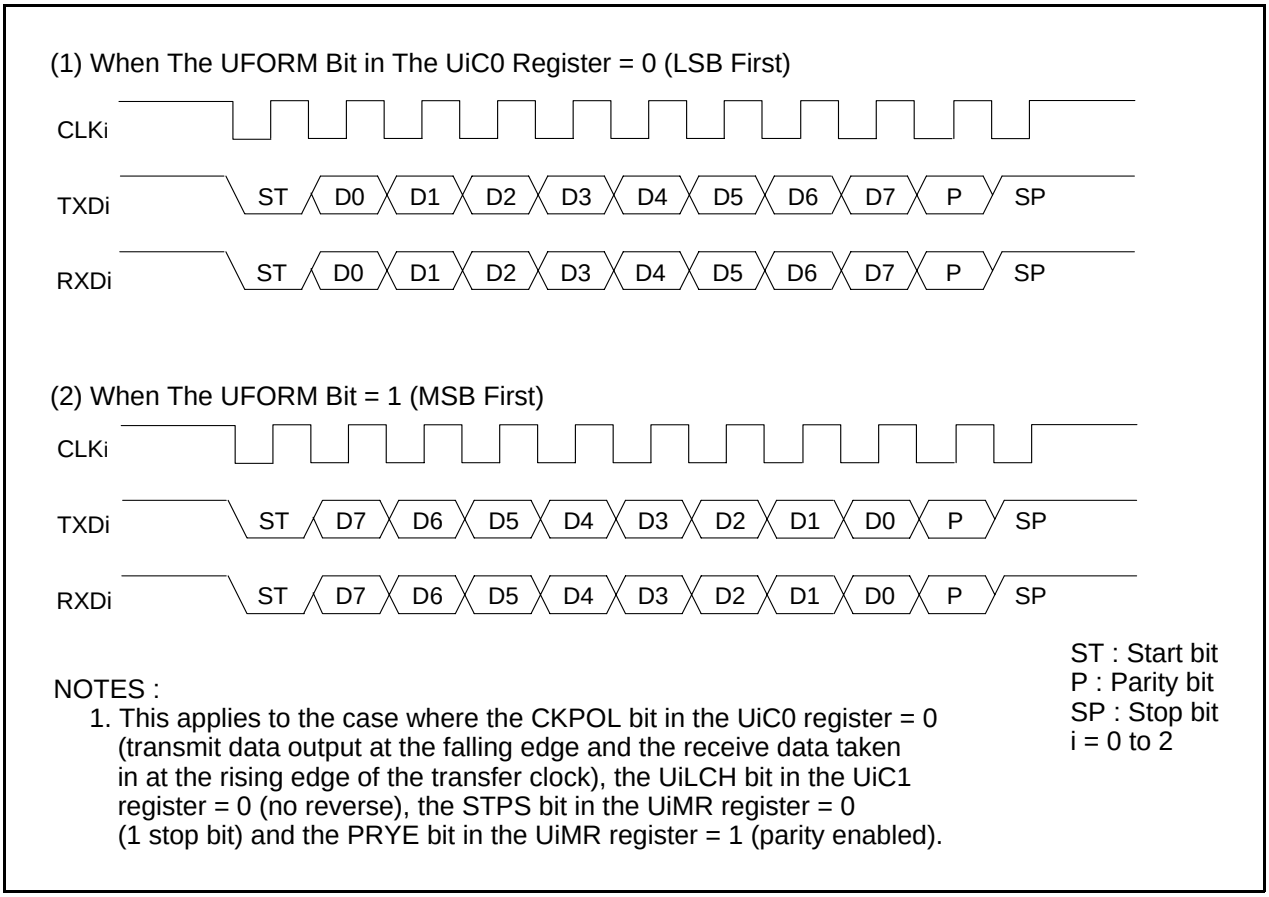


Figure 15.20 Transfer Format

15.1.2.4 Serial Data Logic Switching Function

The data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 15.21 shows Serial Data Logic Switching.

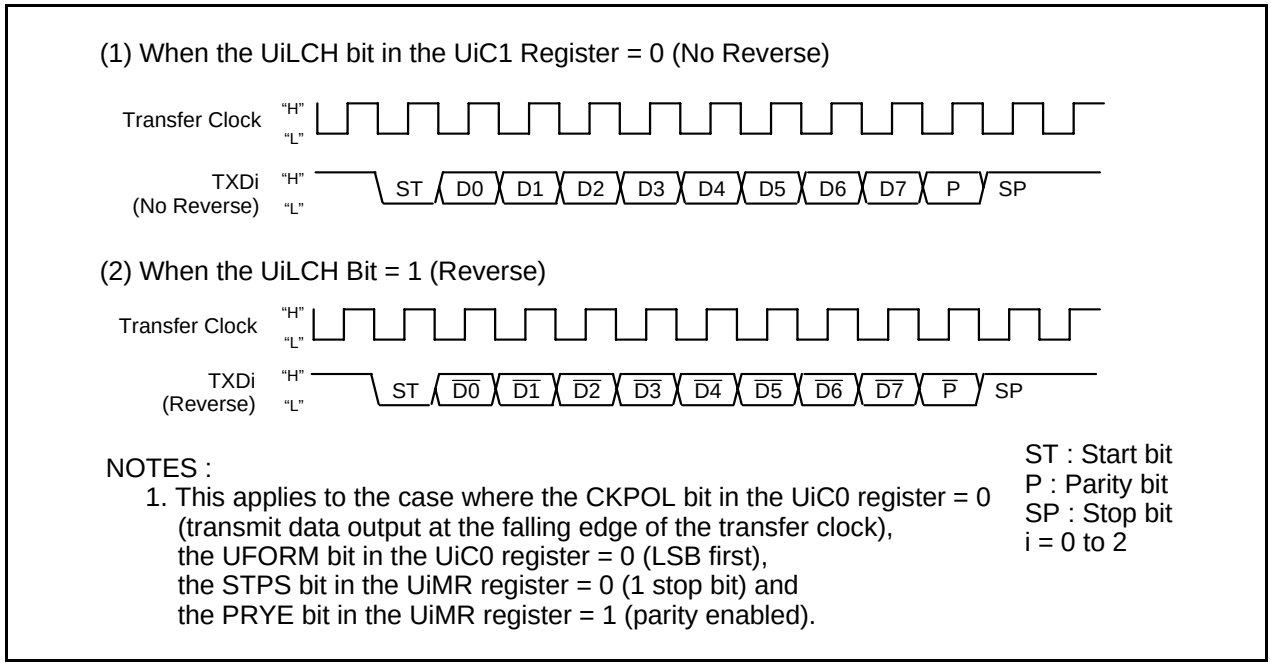


Figure 15.21 Serial Data Logic Switching

15.1.2.5 TXD and RXD I/O Polarity Inverse Function

This function inverses the polarities of the TXDi pin output and RXDi pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inversed. Figure 15.22 shows the TXD and RXD I/O Polarity Inverse.

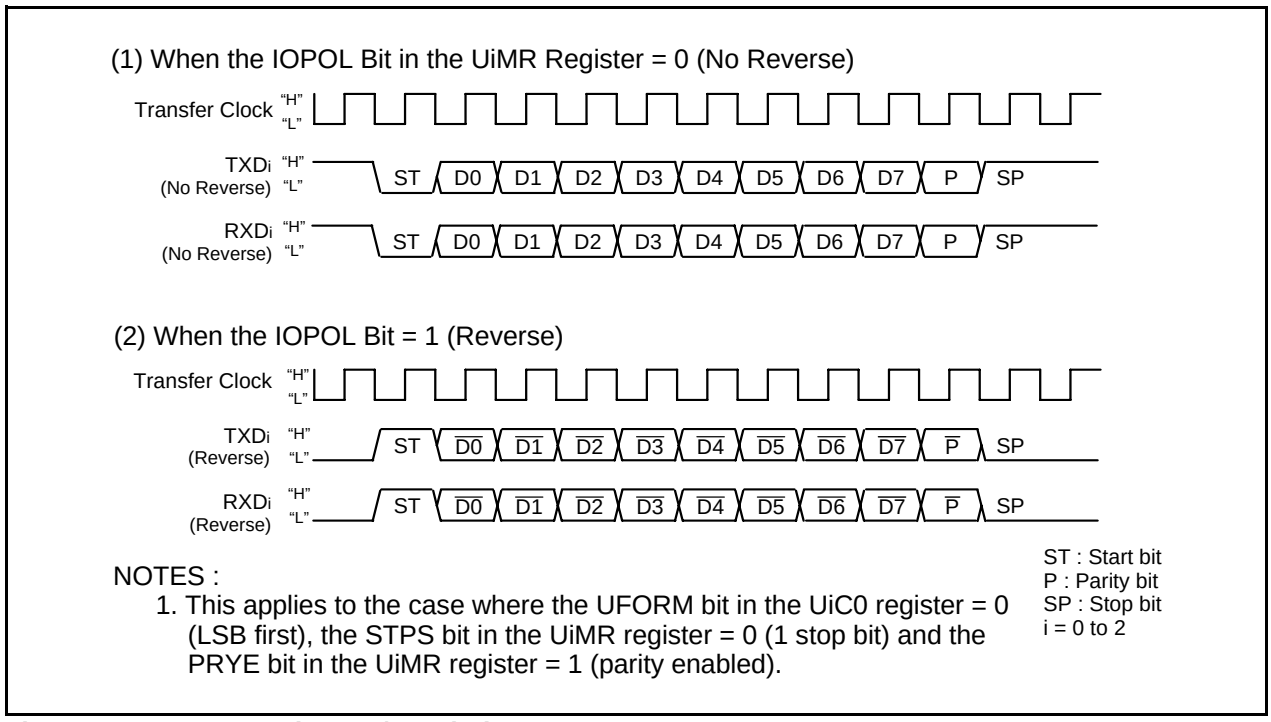


Figure 15.22 TXD and RXD I/O Polarity Inverse

15.1.2.6 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Function

When the $\overline{\text{CTS}}$ function is used transmit operation start when "L" is applied to the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ (i=0 to 2) pin. Transmit operation begins when the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin is held "L". If the "L" signal is switched to "H" during a transmit operation, the operation stops before the next data.
When the $\overline{\text{RTS}}$ function is used, the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin outputs on "L" signal when the microcomputer is ready to receive. The output level becomes "H" on the first falling edge of the CLKi pin.

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

15.1.2.7 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function (UART0)

This function separates $\overline{\text{CTS0}}/\overline{\text{RTS0}}$, outputs $\overline{\text{RTS0}}$ from the P6_0 pin, and accepts as input the $\overline{\text{CTS0}}$ from the P6_4 pin. To use this function, set the register bits as shown below.

- CRD bit in U0C0 register = 0 (enable $\overline{\text{CTS}}/\overline{\text{RTS}}$ of UART0)
- CRS bit in U0C0 register = 1 (output $\overline{\text{RTS}}$ of UART0)
- CRD bit in U1C0 register = 0 (enable $\overline{\text{CTS}}/\overline{\text{RTS}}$ of UART1)
- CRS bit in U1C0 register = 0 (input $\overline{\text{CTS}}$ of UART1)
- RCSP bit in UCON register = 1 (inputs $\overline{\text{CTS0}}$ from the P6_4 pin)
- CLKMD1 bit in UCON register = 0 (CLKS1 not used)

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

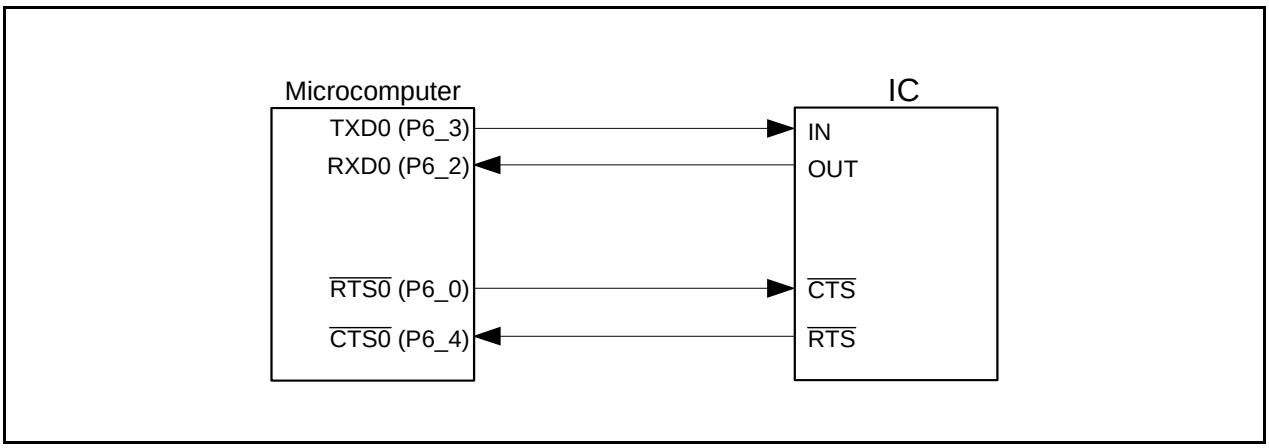


Figure 15.23 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

15.1.3 Special Mode 1 (I^2C mode)

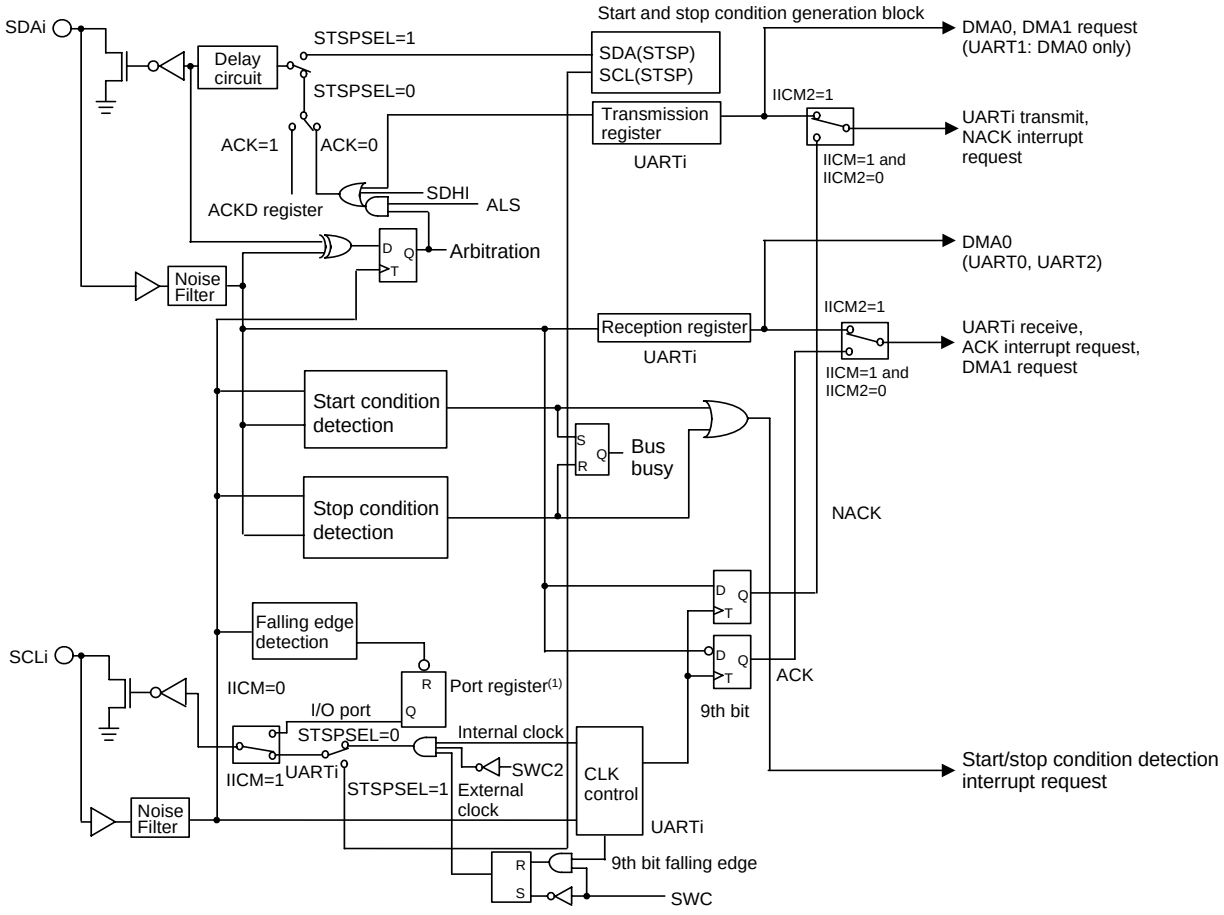
I^2C mode is provided for use as a simplified I^2C interface compatible mode. Table 15.10 lists the specifications of the I^2C mode. Table 15.11 to 15.12 lists the registers used in the I^2C mode and the register values set. Table 15.13 lists the I^2C Mode Functions. Figure 15.24 shows the block diagram for I^2C mode. Figure 15.25 shows Transfer to UiRB Register and Interrupt Timing.

As shown in Table 15.13, the microcomputer is placed in I^2C mode by setting the SMD2 to SMD0 bits to "010b" and the IICM bit to "1". Because SDAi transmit output has a delay circuit attached, SDAi output does not change state until SCLi goes low and remains stably low.

Table 15.10 I^2C Mode Specifications

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	• During master CKDIR bit in the UiMR (i=0 to 2) register = 0 (internal clock) : fj/ (2(n+1)) fj = f1SIO, f2SIO, f8SIO, f32SIO n: Setting value of UiBRG register 00h to FFh • During slave CKDIR bit = 1 (external clock) : Input from SCLi pin
Transmission Start Condition	Before transmission can start, the following requirements must be met (1) • The TE bit in the UiC1 register= 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in UiTB register)
Reception Start Condition	Before reception can start, the following requirements must be met (1) • The RE bit in UiC1 register= 1 (reception enabled) • The TE bit in UiC1 register= 1 (transmission enabled) • The TI bit in UiC1 register= 0 (data present in the UiTB register)
Interrupt Request Generation Timing	When start or stop condition is detected, acknowledge undetected, and acknowledge detected
Error Detection	Overrun error (2) This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the 8th bit of the next data
Select Function	• Arbitration lost Timing at which the ABT bit in the UiRB register is updated can be selected • SDAi digital delay No digital delay or a delay of 2 to 8 UiBRG count source clock cycles selectable • Clock phase setting With or without clock delay selectable

NOTES:
1. When an external clock is selected, the conditions must be met while the external clock is in the high state.
2. If an overrun error occurs, bits 8 to 0 in the UiRB register are undefined. The IR bit in the SiRIC register does not change to "1" (interrupt requested).



This diagram applies to the case where the SMD2 to SMD0 bits in the UiMR register = 010b and the IICM bit in the UiSMR register = 1.

IICM : Bit in UiSMR register
IICM2, SWC, ALS, SWC2, SDHI : Bit in UiSMR2 register
STSPSEL, ACKD, ACKC : Bit in UiSMR4 register

i=0 to 2

NOTES :
1. If the IICM bit = 1, the pin can be read even when the PD6_2, PD6_6 or PD7_1 bit = 1 (output mode).

Figure 15.24 I²C Mode Block Diagram

Table 15.11 Registers to Be Used and Settings in I²C Mode (1)

Register	Bit	Function	
		Master	Slave
UiTB	0 to 7	Set transmission data	Set transmission data
UiRB (3)	0 to 7	Reception data can be read	Reception data can be read
	8	ACK or NACK is set in this bit	ACK or NACK is set in this bit
	ABT	Arbitration lost detection flag	Invalid
	OER	Overrun error flag	Overrun error flag
UiBRG	0 to 7	Set a bit rate	Invalid
UiMR (3)	SMD2 to SMD0	Set to “010b”	Set to “010b”
	CKDIR	Set to “0”	Set to “1”
	IOPOL	Set to “0”	Set to “0”
UiC0	CLK1, CLK0	Select the count source for the UiBRG register	Invalid
	CRS	Invalid because CRD = 1	Invalid because CRD = 1
	TXEPT	Transmit buffer empty flag	Transmit buffer empty flag
	CRD (4)	Set to “1”	Set to “1”
	NCH	Set to “1” (2)	Set to “1” (2)
	CKPOL	Set to “0”	Set to “0”
	UFORM	Set to “1”	Set to “1”
UiC1	TE	Set this bit to “1” to enable transmission	Set this bit to “1” to enable transmission
	TI	Transmit buffer empty flag	Transmit buffer empty flag
	RE	Set this bit to “1” to enable reception	Set this bit to “1” to enable reception
	RI	Reception complete flag	Reception complete flag
	U2IRS (1)	Invalid	Invalid
	U2RRM (1), UiLCH, UiERE	Set to “0”	Set to “0”
UiSMR	IICM	Set to “1”	Set to “1”
	ABC	Select the timing at which arbitration-lost is detected	Invalid
	BBS	Bus busy flag	Bus busy flag
	3 to 7	Set to “0”	Set to “0”
UiSMR2	IICM2	See **Table 15.13 I²C Mode Functions**	See **Table 15.13 I²C Mode Functions**
	CSC	Set this bit to “1” to enable clock synchronization	Set to “0”
	SWC	Set this bit to “1” to have SCLi output fixed to “L” at the falling edge of the 9th bit of clock	Set this bit to “1” to have SCLi output fixed to “L” at the falling edge of the 9th bit of clock
	ALS	Set this bit to “1” to have SDAi output stopped when arbitration-lost is detected	Set to “0”
	STAC	Set to “0”	Set this bit to “1” to initialize UARTi at start condition detection
	SWC2	Set this bit to “1” to have SCLi output forcibly pulled low	Set this bit to “1” to have SCLi output forcibly pulled low
	SDHI	Set this bit to “1” to disable SDAi output	Set this bit to “1” to disable SDAi output
	7	Set to “0”	Set to “0”

NOTES:
1. Set the bit 4 and bit 5 in the U0C1 and U1C1 register to “0”. The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.
2. TXD2 pin is N channel open-drain output. No NCH bit in the U2C0 register is assigned. When write, set to “0”.
3. Not all register bits are described above. Set those bits to “0” when writing to the registers in I²C mode.
4. When using UART1 in I²C mode and enabling the $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function of UART0, set the CRD bit in the U1C0 register to “0” ($\overline{\text{CTS}}/\overline{\text{RTS}}$ enable) and the CRS bit to “0” ($\overline{\text{CTS}}$ input).

i=0 to 2

Table 15.12 Registers to Be Used and Settings in I²C Mode (2)

Register	Bit	Function	
		Master	Slave
UiSMR3	0, 2, 4 and NODC	Set to “0”	Set to “0”
	CKPH	See **Table 15.13 I²C Mode Functions**	See **Table 15.13 I²C Mode Functions**
	DL2 to DL0	Set the amount of SDAi digital delay	Set the amount of SDAi digital delay
UiSMR4	STAREQ	Set this bit to “1” to generate start condition	Set to “0”
	RSTAREQ	Set this bit to “1” to generate restart condition	Set to “0”
	STPREQ	Set this bit to “1” to generate stop condition	Set to “0”
	STSPSEL	Set this bit to “1” to output each condition	Set to “0”
	ACKD	Select ACK or NACK	Select ACK or NACK
	ACKC	Set this bit to “1” to output ACK data	Set this bit to “1” to output ACK data
	SCLHI	Set this bit to “1” to have SCLi output stopped when stop condition is detected	Set to “0”
	SWC9	Set to “0”	Set this bit to “1” to set the SCLi to “L” hold at the falling edge of the 9th bit of clock
IFSR2A	IFSR26, ISFR27	Set to “1”	Set to “1”
UCON	U0IRS, U1IRS	Invalid	Invalid
	U0RRM, U1RRM	Set to “0”	Set to "0"
	CLKMD0	Set to "0"	Set to "0"
	CLKMD1	Set to "0"	Set to "0"
	RCSP	Set to "0"	Set to "0"
	7	Set to "0"	Set to "0"

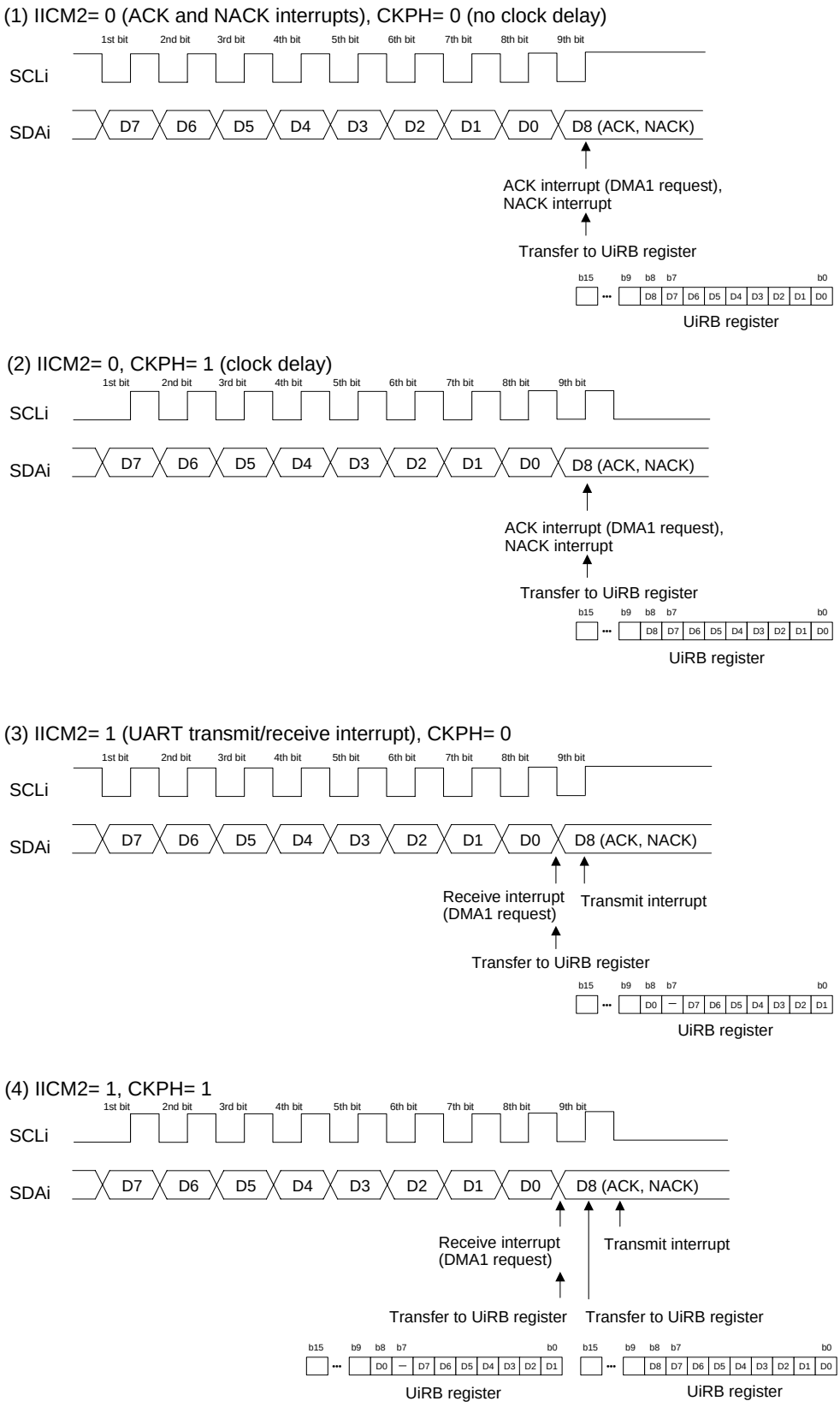
i=0 to 2

Table 15.13 I²C Mode Functions

Function	Clock Synchronous Serial I/O Mode (SMD2 to SMD0 = 001b, IICM = 0)	I²C Mode (SMD2 to SMD0 = 010b, IICM = 1)			
		IICM2 = 0 (NACK/ACK interrupt)		IICM2 = 1 (UART transmit/receive interrupt)	
		CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)	CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)
Factor of Interrupt Number 6, 7 and 10 (1, 5, 7)	–	Start condition detection or stop condition detection (See **Table 15.14 STSPSEL Bit Functions**)			
Factor of Interrupt Number 15, 17 and 19 (1, 6)	UARTi transmission Transmission started or completed (selected by UiIRS)	No acknowledgment detection (NACK) Rising edge of SCLi 9th bit		UARTi transmission Rising edge of SCLi 9th bit	UARTi transmission Falling edge of SCLi next to the 9th bit
Factor of Interrupt Number 16, 18 and 20 (1, 6)	UARTi reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) Rising edge of SCLi 9th bit		UARTi reception Falling edge of SCLi 9th bit	
Timing for Transferring Data From the UART Reception Shift Register to the UiRB Register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Rising edge of SCLi 9th bit		Falling edge of SCLi 9th bit	Falling and rising edges of SCLi 9th bit
UARTi Transmission Output Delay	Not delayed	Delayed			
Functions of P6_3, P6_7 and P7_0 Pins	TXDi output	SDAi input/output			
Functions of P6_2, P6_6 and P7_1 Pins	RXDi input	SCLi input/output			
Functions of P6_1, P6_5 and P7_2 Pins	CLKi input or output selected	– (Cannot be used in I²C mode)			
Noise Filter Width	15ns	200ns			
Read RXDi and SCLi Pin Levels	Possible when the corresponding port direction bit = 0	Always possible no matter how the corresponding port direction bit is set			
Initial Value of TXDi and SDAi Outputs	CKPOL = 0 (H) CKPOL = 1 (L)	The value set in the port register before setting I²C mode (2)			
Initial and End Values of SCLi	–	H	L	H	L
DMA1 Factor (6)	UARTi reception	Acknowledgment detection (ACK)		UARTi reception Falling edge of SCLi 9th bit	
Store Received Data	1st to 8th bits of the received data are stored into bits 7 to 0 in the UiRB register	1st to 8th bits of the received data are stored into bits 7 to 0 in the UiRB register		1st to 7th bits of the received data are stored into bits 6 to 0 in the UiRB register. 8th bit is stored into bit 8 in the UiRB register.	
					1st to 8th bits are stored into bits 7 to 0 in the UiRB register (3)
Read Received Data	The UiRB register status is read				Bits 6 to 0 in the UiRB register (4) are read as bits 7 to 1. Bit 8 in the UiRB register is read as bit 0.

NOTES:
1. If the source or factor of any interrupt is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to “1” (interrupt requested). (Refer to **22.6 Precautions for Interrupt**)
   If one of the bits shown below is changed, the interrupt source, the interrupt timing, etc. change. Therefore, always be sure to clear the IR bit to “0” (interrupt not requested) after changing those bits.
   SMD2 to SMD0 bits in the UiMR register, IICM bit in the UiSMR register, IICM2 bit in the UiSMR2 register, CKPH bit in the UiSMR3 register
2. Set the initial value of SDAi output while the SMD2 to SMD0 bits in the UiMR register = 000b (serial interface disabled).
3. Second data transfer to UiRB register (Rising edge of SCLi 9th bit)
4. First data transfer to UiRB register (Falling edge of SCLi 9th bit)
5. See **Figure 15.27 STSPSEL Bit Functions**.
6. See **Figure 15.25 Transfer to UiRB Register and Interrupt Timing**.
7. When using UART0, be sure to set the IFSR26 bit in the IFSR2A register to “1” (factor of interrupt: UART0 bus collision).
   When using UART1, be sure to set the IFSR27 bit to “1” (factor of interrupt: UART1 bus collision).

i = 0 to 2



i=0 to 2

This diagram applies to the case where the following condition is met.
• UiMR register CKDIR bit = 0 (Slave selected)

Figure 15.25 Transfer to UiRB Register and Interrupt Timing

15.1.3.1 Detection of Start and Stop Condition

Whether a start or a stop condition has been detected is determined.
A start condition-detected interrupt request is generated when the SDAi pin changes state from high to low while the SCLi pin is in the high state. A stop condition-detected interrupt request is generated when the SDAi pin changes state from low to high while the SCLi pin is in the high state.
Because the start and stop condition-detected interrupts share the interrupt control register and vector, check the BBS bit in the UiSMR register to determine which interrupt source is requesting the interrupt.

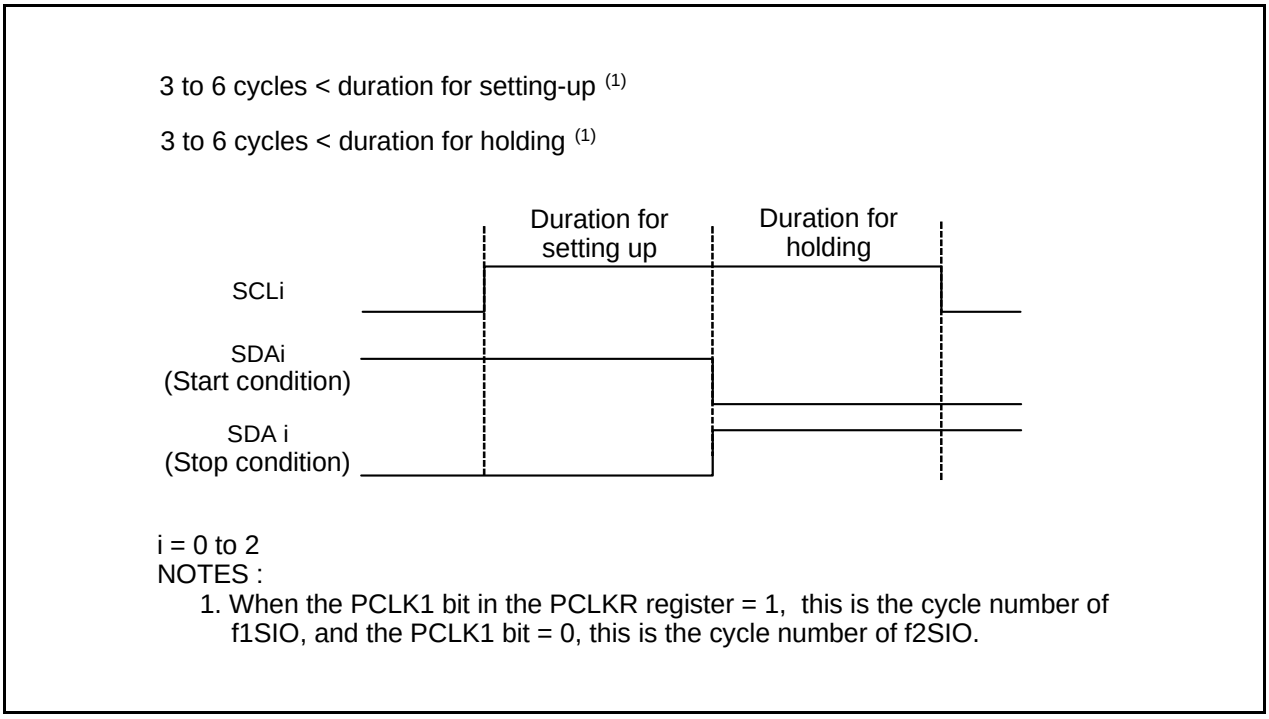


Figure 15.26 Detection of Start and Stop Condition

15.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the STAREQ bit in the UiSMR4 register (i = 0 to 2) to “1” (start).
A restart condition is generated by setting the RSTAREQ bit in the UiSMR4 register to “1” (start).
A stop condition is generated by setting the STPREQ bit in the UiSMR4 register to “1” (start).
The output procedure is described below.

(1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to “1” (start).
(2) Set the STSPSEL bit in the UiSMR4 register to “1” (output).

The function of the STSPSEL bit is shown in Table 15.14 and Figure 15.27.

Table 15.14 STSPSEL Bit Functions

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

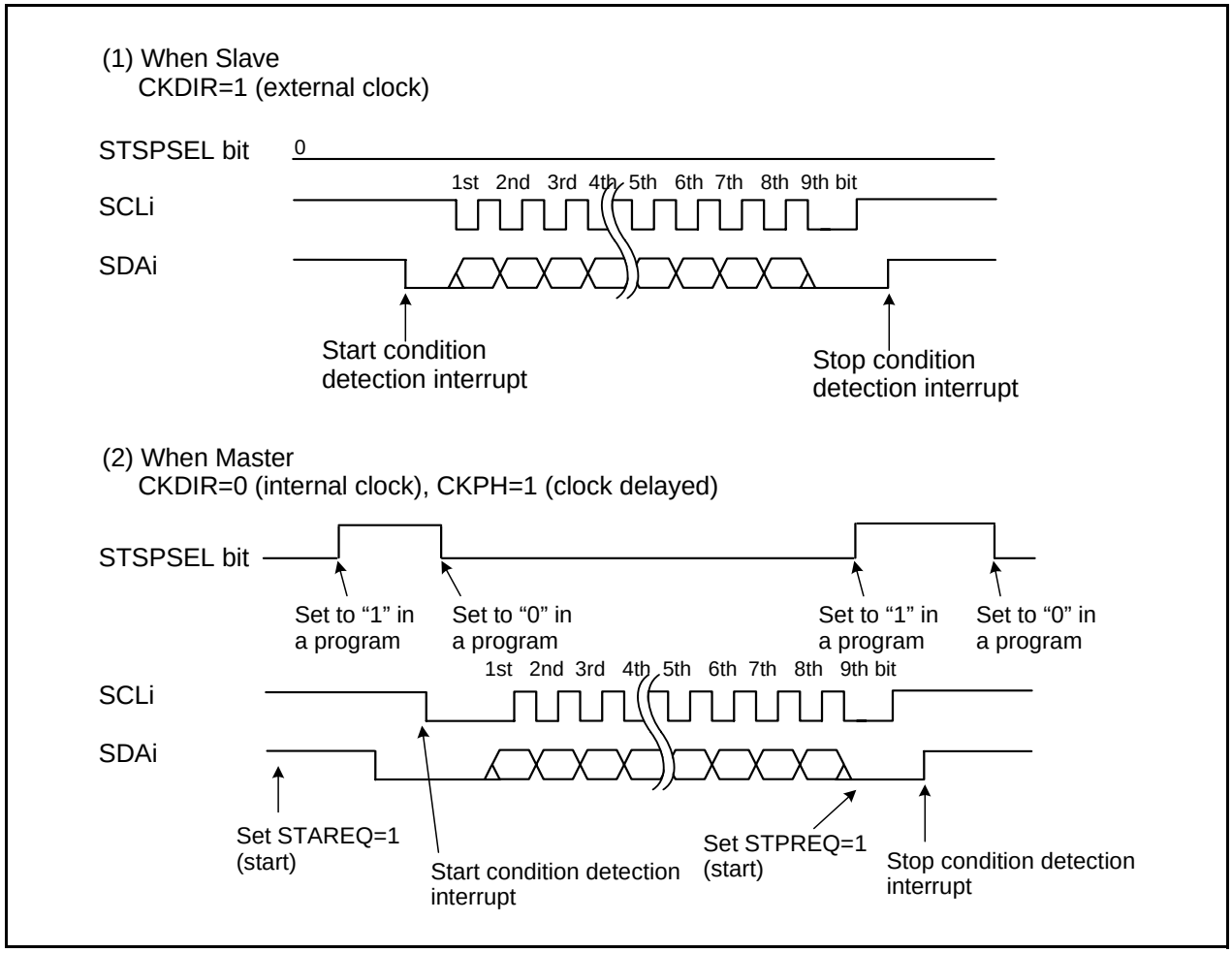


Figure 15.27 STSPSEL Bit Functions

15.1.3.3 Arbitration

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

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

15.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 15.25 Transfer to UiRB Register and Interrupt Timing.

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

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

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

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

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

If the SWC9 bit in the UiSMR4 register is set to "1" (SCL hold low enabled) when the CKPH bit in the UiSMR3 register = 1, the SCLi pin is fixed to low-level output at the falling edge of the clock pulse next to the 9th. Setting the SWC9 bit = 0 (SCL hold low disabled) frees the SCLi pin from low-level output.

15.1.3.5 SDA Output

The data written to the UiTB register bit 7 to bit 0 (D7 to D0) is sequentially output beginning with D7.

The 9th bit (D8) is ACK or NACK.

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

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

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

15.1.3.6 SDA Input

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

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

15.1.3.7 ACK and NACK

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

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

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

15.1.3.8 Initialization of Transmission/Reception

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

- The transmit shift register is initialized, and the content of the UiTB register is transferred to the transmit shift register. In this way, the serial interface starts sending data synchronously with the next clock pulse applied. However, the UARTi output value does not change state and remains the same as when a start condition was detected until the first bit of data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial interface starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to “1” (SCL wait output enabled). Consequently, the SCLi pin is pulled low at the falling edge of the 9th clock pulse.

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

15.1.4 Special Mode 2

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. Table 15.15 lists the Special Mode 2 Specifications. Table 15.16 lists the Registers to Be Used and Settings in Special Mode 2. Figure 15.28 shows Serial Bus Communication Control Example (UART2).

Table 15.15 Special Mode 2 Specifications

Item	Specification
Transfer Data Format	Transfer data length: 8 bits
Transfer Clock	• Master mode CKDIR bit in UiMR(i=0 to 2) register = 0 (internal clock) : fj/ (2(n+1)) fj = f1SIO, f2SIO, f8SIO, f32SIO n: Setting value of UiBRG register 00h to FFh • Slave mode CKDIR bit = 1 (external clock selected) : Input from CLKi pin
Transmit/Receive Control	Controlled by input/output ports
Transmission Start Condition	Before transmission can start, the following requirements must be met (1) • The TE bit in UiC1 register= 1 (transmission enabled) • The TI bit in UiC1 register = 0 (data present in UiTB register)
Reception Start Condition	Before reception can start, the following requirements must be met (1) • The RE bit in UiC1 register= 1 (reception enabled) • The TE bit in UiC1 register= 1 (transmission enabled) • The TI bit in UiC1 register= 0 (data present in the UiTB register)
Interrupt Request Generation Timing	For transmission, one of the following conditions can be selected • The UiIRS bit in UiC1 register = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit =1 (transfer completed): when the serial interface finished sending data from the UARTi transmit register For reception • When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error Detection	Overrun error (2) This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the 7th bit of the next data
Select Function	• CLK polarity selection Transfer data input/output can be 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 UiRB register • Switching serial data logic This function reverses the logic value of the transmit/receive dataClock phase setting • Selectable from four combinations of transfer clock polarities and phases

NOTES:
1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. If an overrun error occurs, bits 8 to 0 in the UiRB register are undefined. The IR bit in the SiRIC register does not change to "1" (interrupt requested).

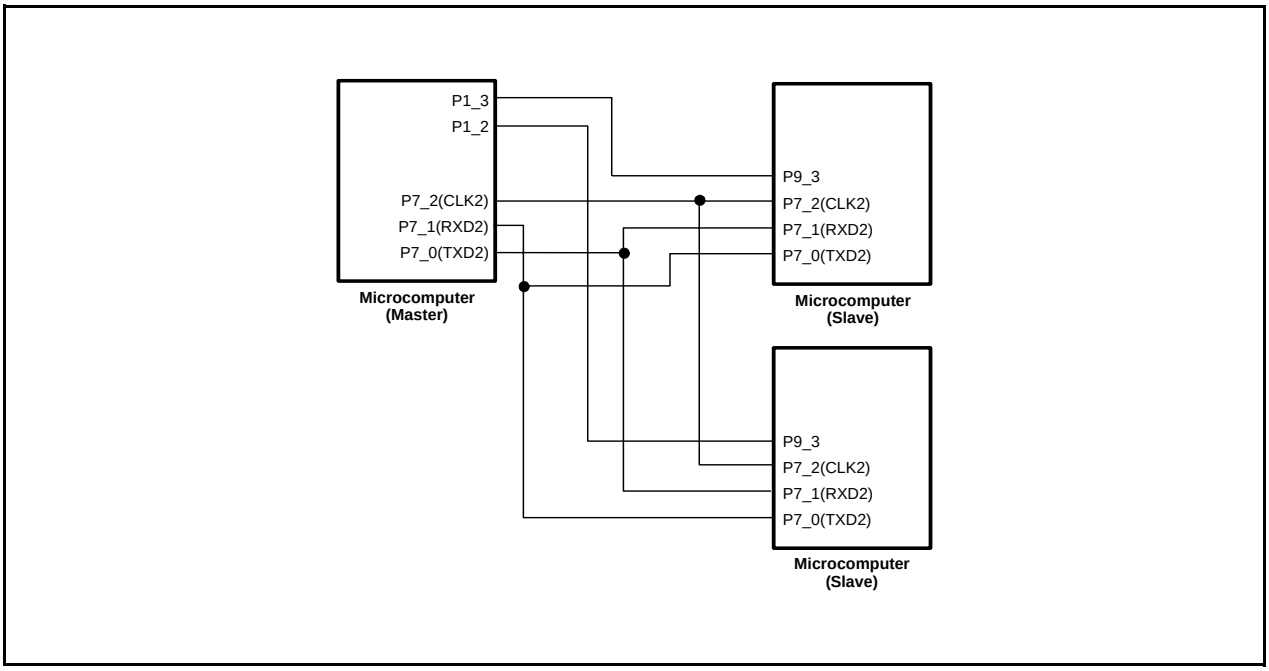


Figure 15.28 Serial Bus Communication Control Example (UART2)

Table 15.16 Registers to Be Used and Settings in Special Mode 2

Register	Bit	Function
UiTB (3)	0 to 7	Set transmission data
UiRB (3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a bit rate
UiMR (3)	SMD2 to SMD0	Set to “001b”
	CKDIR	Set this bit to “0” for master mode or “1” for slave mode
	IOPOL	Set to “0”
UiC0	CLK1, CLK0	Select the count source for the UiBRG register
	CRS	Invalid because CRD = 1
	TXEPT	Transmit register empty flag
	CRD	Set to “1”
	NCH	Select TXDi pin output format (2)
	CKPOL	Clock phases can be set in combination with the CKPH bit in the UiSMR3 register
	UFORM	Select the LSB first or MSB first
UiC1	TE	Set this bit to “1” to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to “1” to enable reception
	RI	Reception complete flag
	U2IRS (1)	Select UART2 transmit interrupt factor
	U2RRM (1)	Set this bit to “1” to use continuous receive mode
	UiLCH	Set this bit to “1” to use inverted data logic
	UiERE	Set to “0”
UiSMR	0 to 7	Set to “0”
UiSMR2	0 to 7	Set to “0”
UiSMR3	CKPH	Clock phases can be set in combination with the CKPOL bit in the UiC0 register
	NODC	Set to “0”
	0, 2, 4 to 7	Set to “0”
UiSMR4	0 to 7	Set to “0”
UCON	U0IRS, U1IRS	Select UART0 and UART1 transmit interrupt factor
	U0RRM, U1RRM	Set this bit to “1” to use continuous receive mode
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1, RCSP, 7	Set to “0”

NOTES:
1. Set the bit 4 and bit 5 in the U0C0 and U1C1 register to “0”. The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.
2. TXD2 pin is N channel open-drain output. No NCH bit in the U2C0 register is assigned. When write, set to “0”.
3. Not all register bits are described above. Set those bits to “0” when writing to the registers in Special Mode 2.

i = 0 to 2

15.1.4.1 Clock Phase Setting Function

One of four combinations of transfer clock phases and polarities can be selected using the CKPH bit in the UiSMR3 register and the CKPOL bit in the UiC0 register.
Make sure the transfer clock polarity and phase are the same for the master and salves to be communicated.
Figure 15.29 shows the Transmission and Reception Timing in Master Mode (Internal Clock).
Figure 15.30 shows the Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock) while Figure 15.31 shows the Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock).

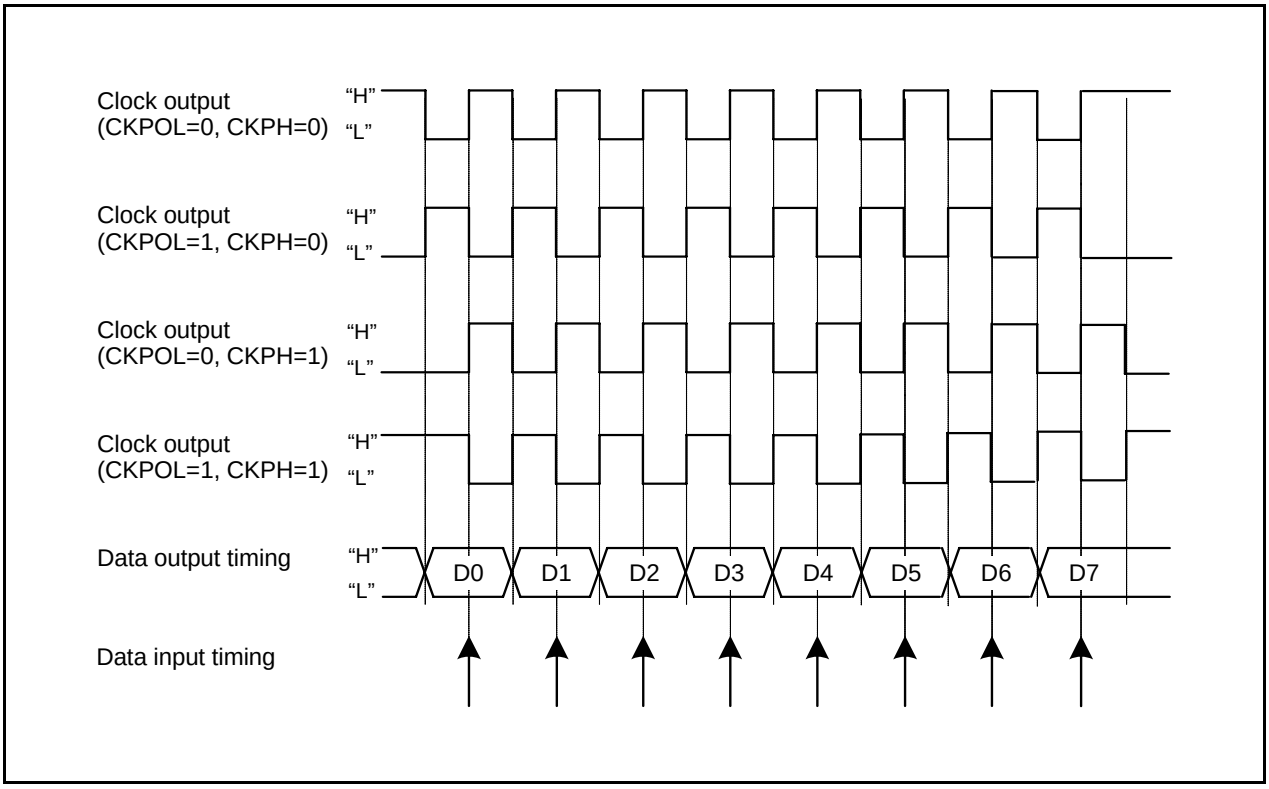
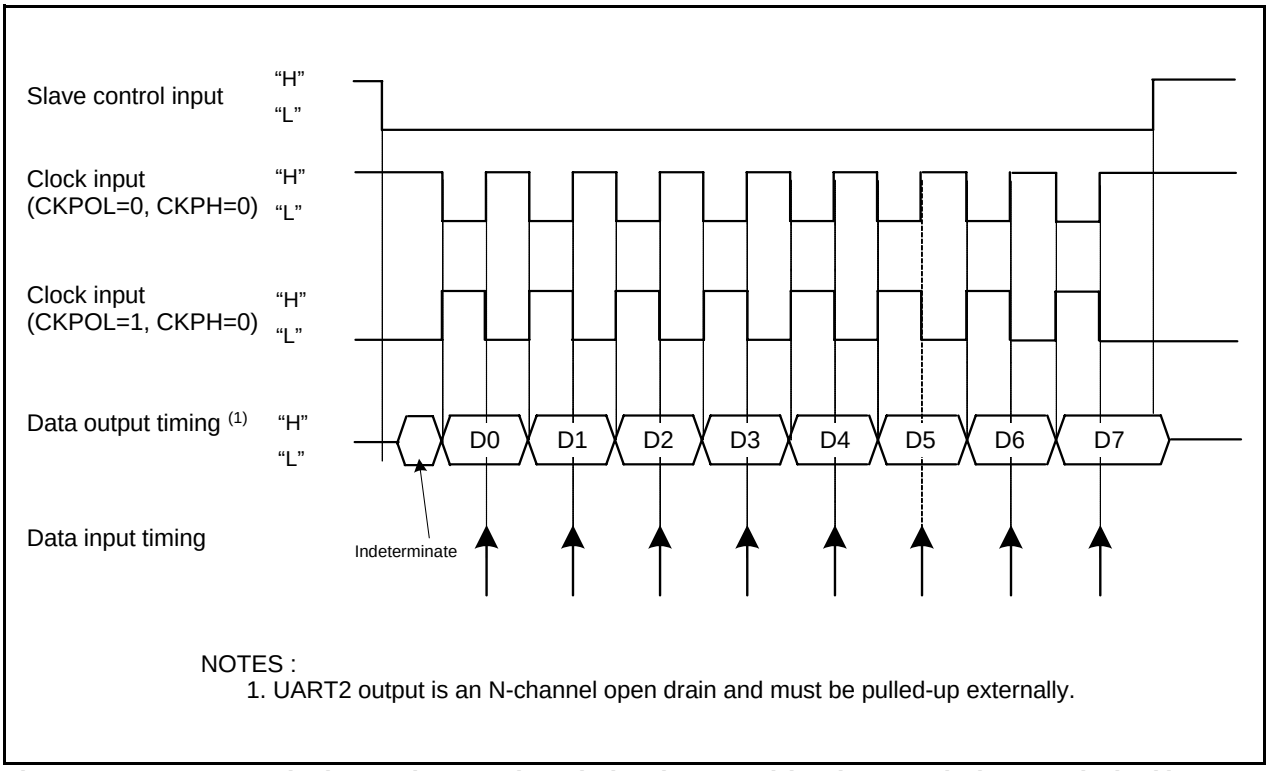
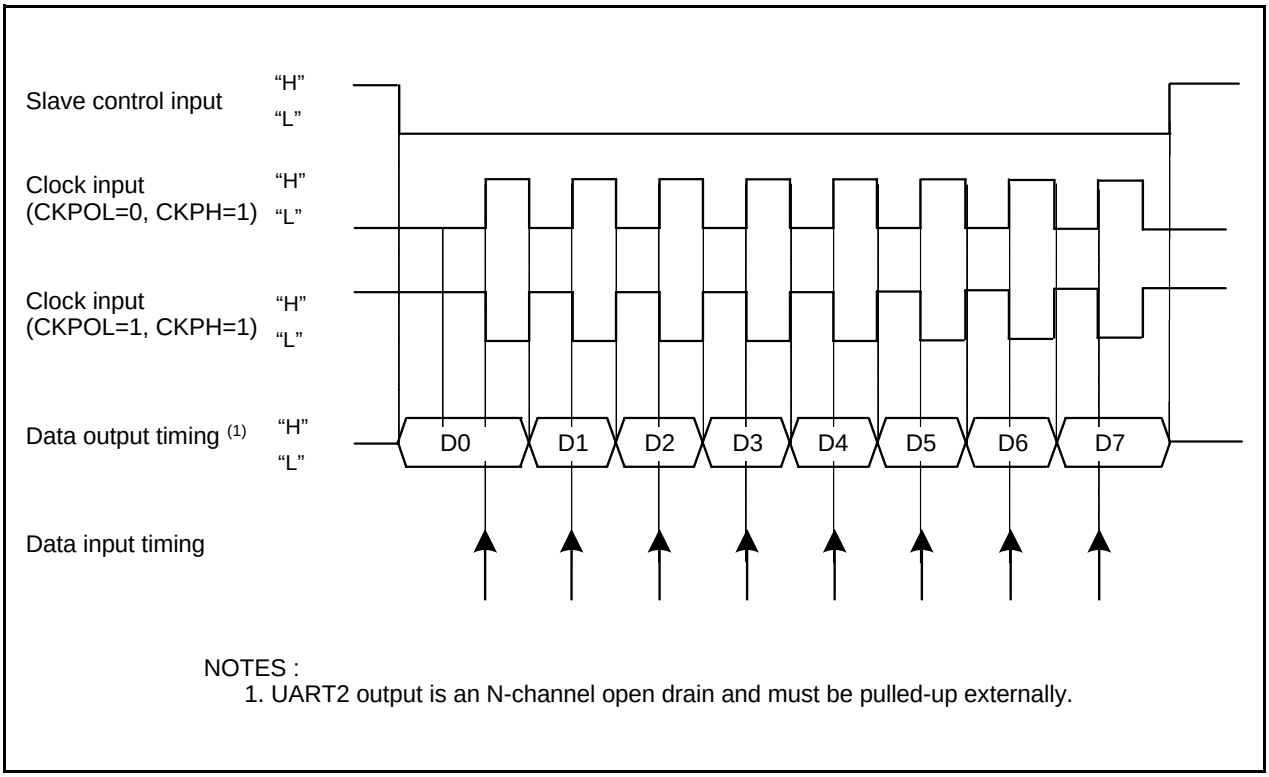


Figure 15.29 Transmission and Reception Timing in Master Mode (Internal Clock)



NOTES :
1. UART2 output is an N-channel open drain and must be pulled-up externally.

Figure 15.30 Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)



NOTES :
1. UART2 output is an N-channel open drain and must be pulled-up externally.

Figure 15.31 Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)

15.1.5 Special Mode 3 (IE mode)(UART2)

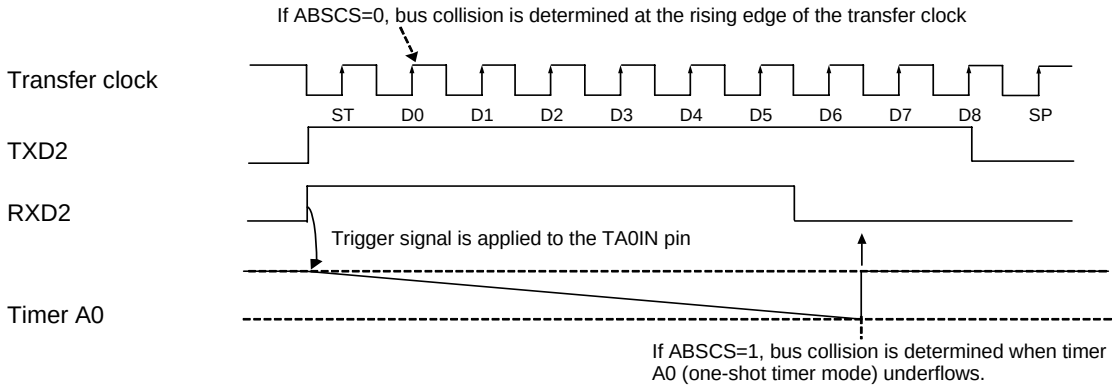
In this mode, one bit of IEBus is approximated with one byte of UART mode waveform.
Table 15.17 lists the Registers to Be Used and Settings in IE Mode. Figure 15.32 shows the Bus Collision Detect Function-Related BitsBus Collision Detect Function-Related Bits.
If the TXD2 pin output level and RXD2 pin input level do not match, a UART2 bus collision detect interrupt request is generated.

Table 15.17 Registers to Be Used and Settings in IE Mode

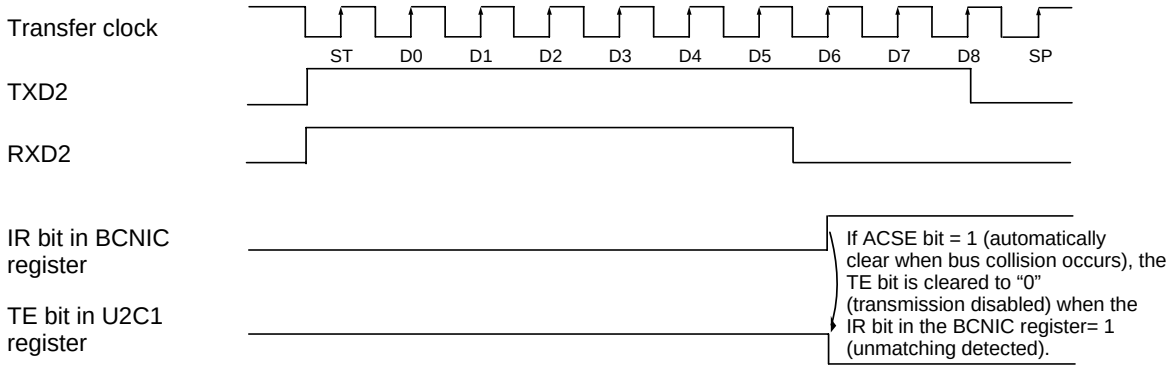
Register	Bit	Function
U2TB	0 to 8	Set transmission data
U2RB (2)	0 to 8	Reception data can be read
	OER, FER, PER, SUM	Error flag
U2BRG	0 to 7	Set a bit rate
U2MR	SMD2 to SMD0	Set to “110b”
	CKDIR	Select the internal clock or external clock
	STPS	Set to “0”
	PRY	Invalid because PRYE=0
	PRYE	Set to “0”
	IOPOL	Select the TXD/RXD input/output polarity
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRD=1
	TXEPT	Transmit register empty flag
	CRD	Set to “1”
	NCH	Select TXD2 pin output mode (1)
	CKPOL	Set to “0”
	UFORM	Set to “0”
U2C1	TE	Set this bit to “1” to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to “1” to enable reception
	RI	Reception complete flag
	U2IRS	Select the source of UART2 transmit interrupt
	U2RRM, U2LCH, U2ERE	Set to “0”
U2SMR	0 to 3, 7	Set to “0”
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to “1” to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
U2SMR2	0 to 7	Set to “0”
U2SMR3	0 to 7	Set to “0”
U2SMR4	0 to 7	Set to “0”

NOTES:
1. TXD2 pin is N channel open-drain output. No NCH bit in the U2C0 register is assigned. When write, set to “0”.
2. Not all register bits are described above. Set those bits to “0” when writing to the registers in IE mode.

(1) The ABSCS Bit in the U2SMR Register (Bus collision detect sampling clock select)

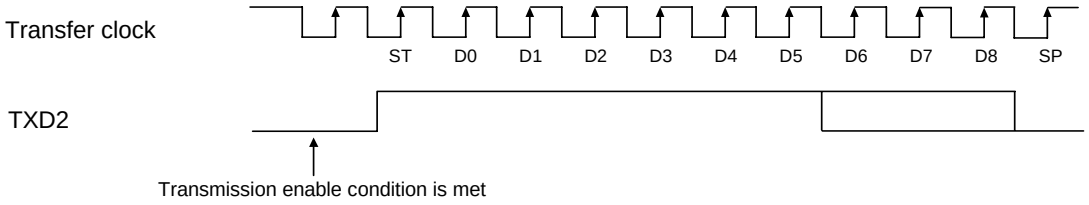


(2) The ACSE Bit in the U2SMR Register (Auto clear of transmit enable bit)

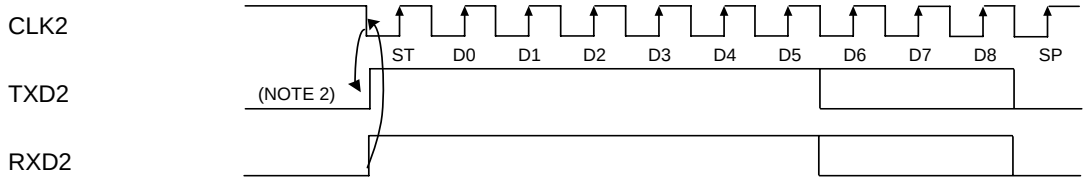


(3) The SSS Bit in the U2SMR Register (Transmit start condition select)

If SSS bit = 0, the serial interface starts sending data one transfer clock cycle after the transmission enable condition is met.



If SSS bit = 1, the serial interface starts sending data at the rising edge (1) of RXD2



NOTES :
1. The falling edge of RXD2 when IOPOL=0; the rising edge of RXD2 when IOPOL =1.
2. The transmit condition must be met before the falling edge (1) of RXD.

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

Figure 15.32 Bus Collision Detect Function-Related Bits

15.1.6 Special Mode 4 (SIM Mode) (UART2)

Based on UART mode, this is an SIM interface compatible mode. Direct and inverse formats can be implemented, and this mode allows to output a low from the TXD2 pin when a parity error is detected.
Table 15.18 lists the SIM Mode Specifications. Table 15.19 lists the Registers to Be Used and Settings in SIM Mode.

Table 15.18 SIM Mode Specifications

Item	Specification
Transfer Data Format	• Direct format • Inverse format
Transfer Clock	• CKDIR bit in U2MR register = 0 (internal clock) : fi/ (16(n+1)) fi = f1SIO, f2SIO, f8SIO, f32SIO n: Setting value of U2BRG register 00h to FFh • CKDIR bit = 1 (external clock) : fEXT/(16(n+1)) fEXT: Input from CLK2 pin n: Setting value of U2BRG register 00h to FFh
Transmission Start Condition	Before transmission can start, the following requirements must be met • The TE bit in the U2C1 register = 1 (transmission enabled) • The TI bit in the U2C1 register = 0 (data present in U2TB register)
Reception Start Condition	Before reception can start, the following requirements must be met • The RE bit in the U2C1 register = 1 (reception enabled) • Start bit detection
Interrupt Request Generation Timing (2)	• For transmission When the serial interface finished sending data from the U2TB transfer register (U2IRS bit =1) • For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception)
Error Detection	• Overrun error (1) This error occurs if the serial interface started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data • Framing error (3) This error occurs when the number of stop bits set is not detected • Parity error (3) During reception, if a parity error is detected, parity error signal is output from the TXD2 pin. During transmission, a parity error is detected by the level of input to the RXD2 pin when a transmission interrupt occurs • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered

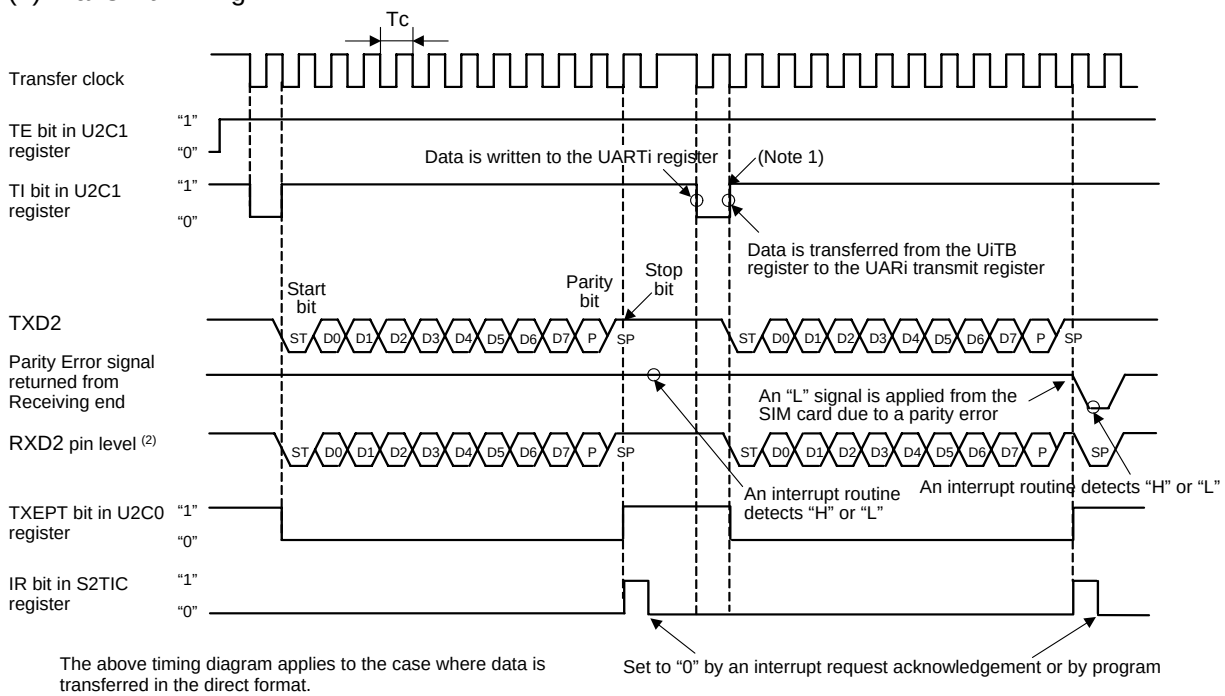
NOTES:
1. If an overrun error occurs, bits 8 to 0 in the U2RB register are undefined. The IR bit in the S2RIC register does not change to “1” (interrupt requested).
2. A transmit interrupt request is generated by setting the U2IRS bit to “1” (transmission complete) and U2ERE bit to “1” (error signal output) in the U2C1 register after reset. Therefore, when using SIM mode, set the IR bit to “0” (no interrupt request) after setting these bits.
3. The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UARTi receive register to the UiRB register.

Table 15.19 Registers to Be Used and Settings in SIM Mode

Register	Bit	Function
U2TB (1)	0 to 7	Set transmission data
U2RB (1)	0 to 7	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	0 to 7	Set a bit rate
U2MR	SMD2 to SMD0	Set to “101b”
	CKDIR	Select the internal clock or external clock
	STPS	Set to “0”
	PRY	Set this bit to “1” for direct format or “0” for inverse format
	PRYE	Set to “1”
	IOPOL	Set to “0”
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRD = 1
	TXEPT	Transmit register empty flag
	CRD	Set to “1”
	NCH	Set to “0”
	CKPOL	Set to “0”
	UFORM	Set this bit to “0” for direct format or “1” for inverse format
U2C1	TE	Set this bit to “1” to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to “1” to enable reception
	RI	Reception complete flag
	U2IRS	Set to “1”
	U2RRM	Set to “0”
	U2LCH	Set this bit to “0” for direct format or “1” for inverse format
	U2ERE	Set to “1”
U2SMR (1)	0 to 3	Set to “0”
U2SMR2	0 to 7	Set to “0”
U2SMR3	0 to 7	Set to “0”
U2SMR4	0 to 7	Set to “0”

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

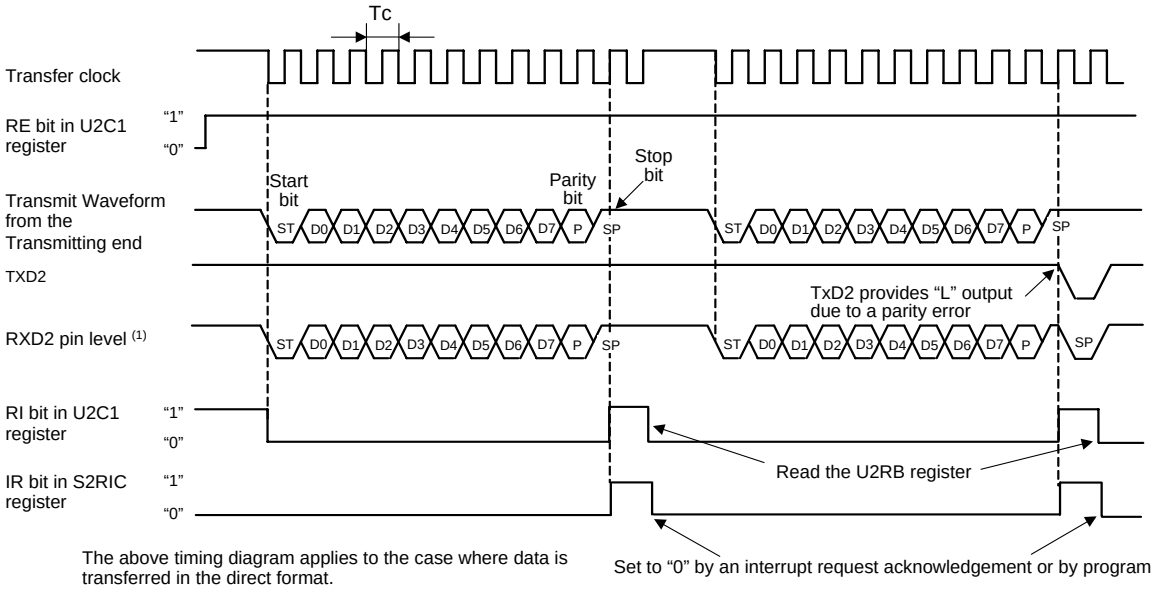
(1) Transmit Timing



The above timing diagram applies to the case where data is transferred in the direct format.
- STPS bit in U2MR register = 0 (1 stop bit)
- PRY bit in U2MR register = 1 (even)
- UFORM bit in U2C0 register = 0 (LSB first)
- U2LCH bit in U2C1 register = 0 (no reverse)
- U2IRSCH bit in U2C1 register = 1 (transmit is completed)

TC = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of U2BRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of U2BRG count source (external clock)
n : value set to U2BRG

(2) Receive Timing



The above timing diagram applies to the case where data is transferred in the direct format.
- STPS bit in U2MR register = 0 (1 stop bit)
- PRY bit in U2MR register = 1 (even)
- UFORM bit in U2C0 register = 0 (LSB first)
- U2LCH bit in U2C1 register = 0 (no reverse)
- U2IRSCH bit in U2C1 register = 1 (transmit is completed)

TC = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of U2BRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of U2BRG count source (external clock)
n : value set to U2BRG

NOTES:
1. Data transmission starts when BRG overflows after a value is set to the U2TB register on the rising edge of the TI bit.
2. Because the TxD2 and RxD2 pins are connected, a composite waveform, consisting of transmit waveform from the TxD2 pin and parity error signal from the receiving end, is generated.
3. Because the TxD2 and RxD2 pins are connected, a composite waveform, consisting of transmit waveform from the transmitting end and parity error signal from the TxD2 pin, is generated.

Figure 15.33 Transmit and Receive Timing in SIM Mode

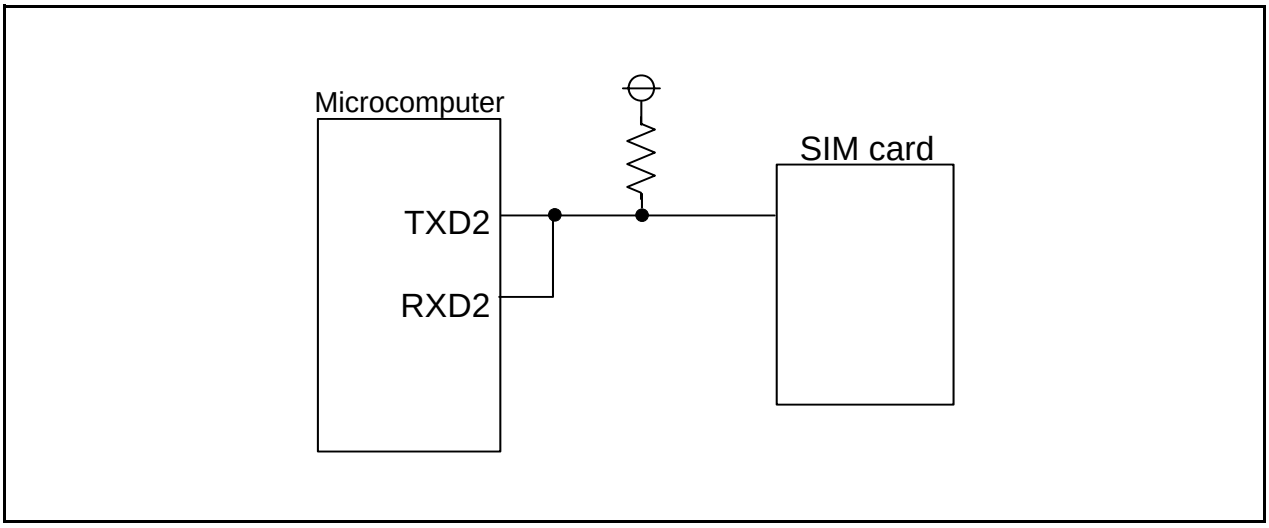


Figure 15.34 SIM Interface Connection

15.1.6.1 Parity Error Signal Output

The parity error signal is enabled by setting the U2ERE bit in the U2C1 register to "1".
The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TXD2 output low with the timing shown in Figure 15.35. If the R2RB register is read while outputting a parity error signal, the PER bit is cleared to "0" and at the same time the TXD2 output is returned high.
When transmitting, a transmission-finished interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity signal has been returned can be determined by reading the port that shares the RXD2 pin in a transmission-finished interrupt routine.

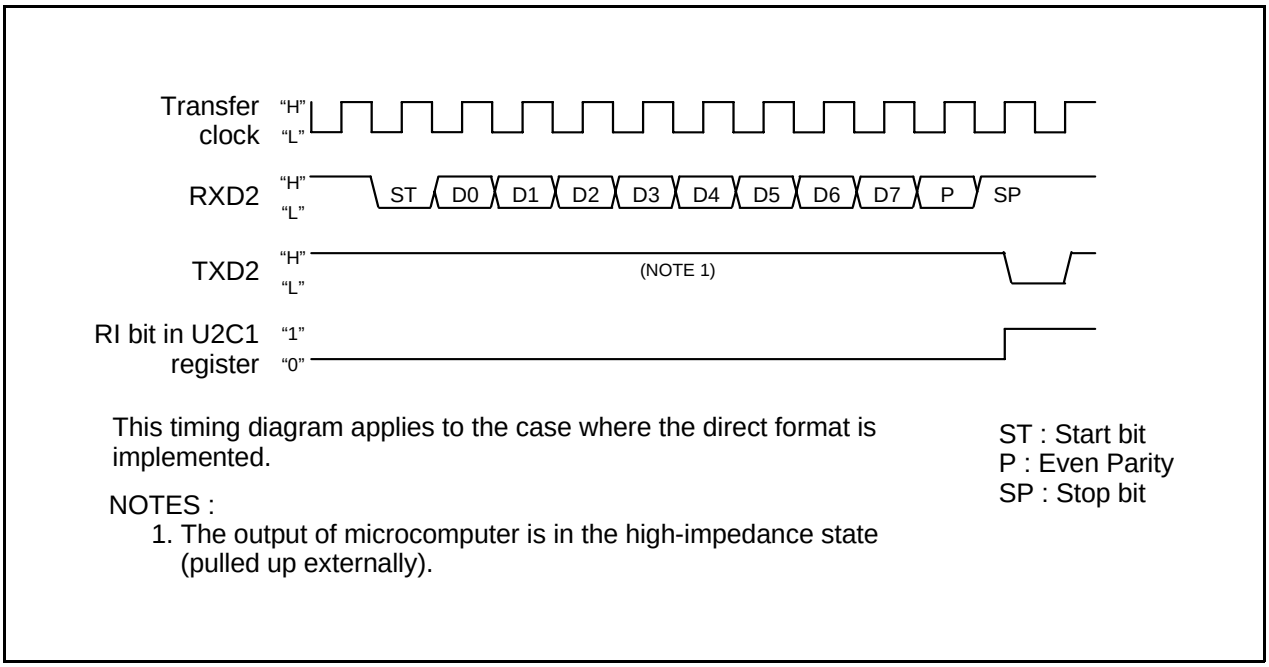


Figure 15.35 Parity Error Signal Output Timing

15.1.6.2 Format

When direct format, set the PRYE bit in the U2MR register to “1”, the PRY bit to “1”, the UFORM bit in the U2C0 register to “0” and the U2LCH bit in the U2C1 register to “0”. When data are transmitted, data set in the U2TB register are transmitted with the even-numbered parity, starting from D0. When data are received, received data are stored in the U2RB register, starting from D0. The even-numbered parity determines whether a parity error occurs.

When inverse format, set the PRYE bit to “1”, the PRY bit to “0”, the UFORM bit to “1” and the U2LCH bit to “1”. When data are transmitted, values set in the U2TB register are logically inversed and are transmitted with the odd-numbered parity, starting from D7. When data are received, received data are logically inversed to be stored in the U2RB register, starting from D7. The odd-numbered parity determines whether a parity error occurs.

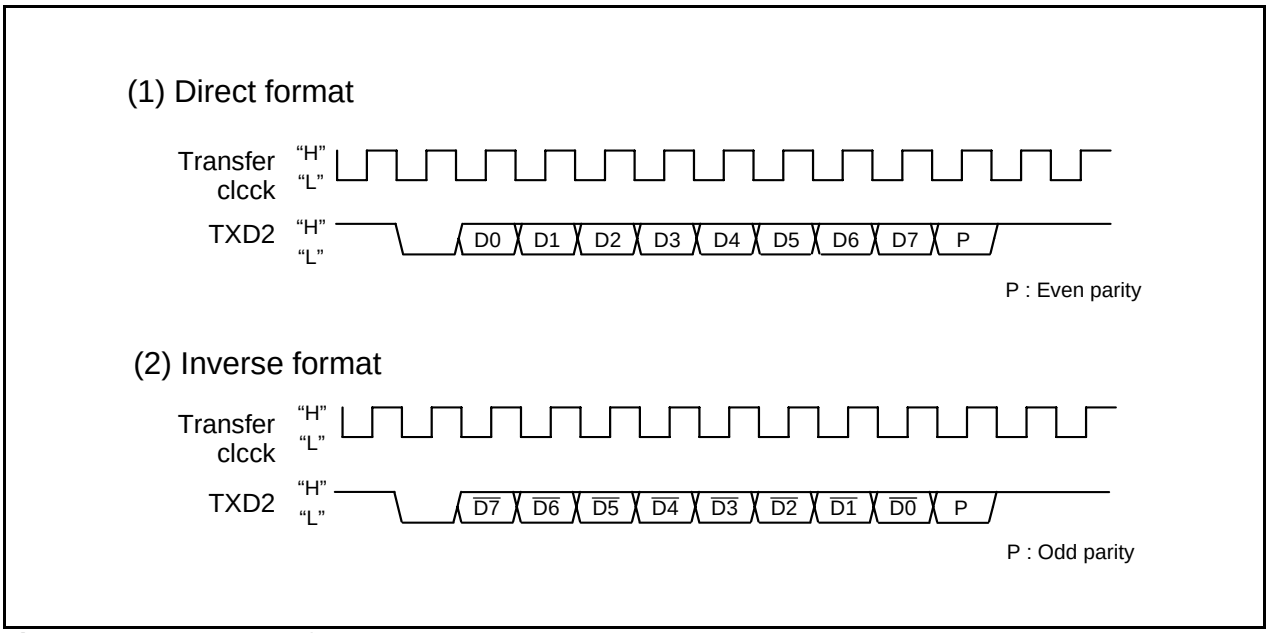


Figure 15.36 SIM Interface Format

16. A/D Converter

The microcomputer contains one A/D converter circuit based on 10-bit successive approximation method configured with a capacitive-coupling amplifier. The analog inputs share the pins with P10_0 to P10_7, P9_5, P9_6, and P0_0 to P0_7. Similarly, $\overline{\text{ADTRG}}$ input shares the pin with P9_7. Therefore, when using these inputs, make sure the corresponding port direction bits are set to "0" (= input mode).
When not using the A/D converter, set the VCUT bit to "0" (= Vref unconnected), so that no current will flow from the VREF pin into the resistor ladder, helping to reduce the power consumption of the chip.
The A/D conversion result is stored in the ADi register bits for ANi, and AN0_i pins (i = 0 to 7).
Table 16.1 shows the Performance of A/D Converter. Figure 16.1 shows the A/D Converter Block Diagram, and Figures 16.2 and 16.3 show the A/D converter-related registers.

Table 16.1 Performance of A/D Converter

Item	Performance
Method of A/D Conversion	Successive approximation (capacitive coupling amplifier)
Analog input Voltage (1)	0V to AVCC (VCC1)
Operating clock ϕAD (2)	fAD/divide-by-2 of fAD/divide-by-3 of fAD/divide-by-4 of fAD/divide-by-6 of fAD/divide-by-12 of fAD
Resolution	8-bit or 10-bit (selectable)
Integral Nonlinearity Error	When AVCC = VREF = 5V • With 8-bit resolution: ±2LSB • With 10-bit resolution AN0 to AN7, AN0_0 to AN0_7, ANEX0 and ANEX1 input : ±5LSB When AVCC = VREF = 3.3V • With 8-bit resolution: ±2LSB • With 10-bit resolution AN0 to AN7, AN0_0 to AN0_7, ANEX0 and ANEX1 input : ±7LSB
Operating Modes	One-shot mode and repeat mode
Analog Input Pins	8 pins (AN0 to AN7) + 2 pins (ANEX0 and ANEX1) + 8 pins (AN0_0 to AN0_7)
A/D Conversion Start Condition	• Software trigger The ADST bit in the ADCON0 register is set to "1" (A/D conversion starts) • External trigger (retriggerable) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to "1" (A/D conversion starts)
Conversion Speed	• Without sample and hold function 8-bit resolution: 49 ϕAD cycles, 10-bit resolution: 59 ϕAD cycles • With sample and hold function 8-bit resolution: 28 ϕAD cycles, 10-bit resolution: 33 ϕAD cycles

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

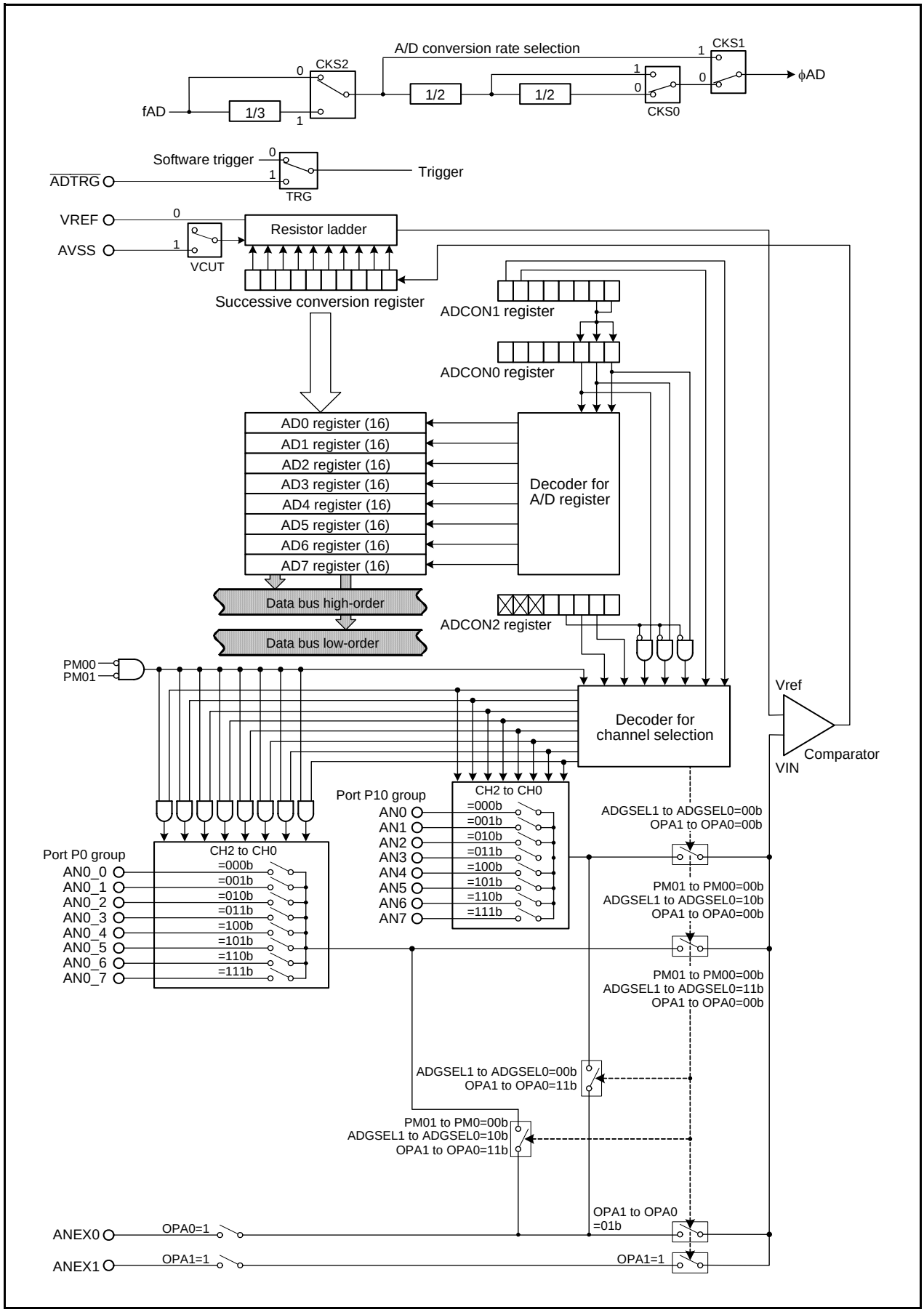
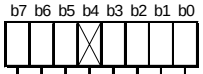


Figure 16.1 A/D Converter Block Diagram

A/D Control Register 0 (1)

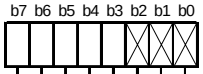


Symbol	Address	After Reset
ADCON0	03D6h	000X0XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0	0 : One-shot mode 1 : Repeat mode	RW
— (b4)	Nothing is assigned. When write, set to “0”. When read, its content is ndeterminate.		—
TRG	Trigger Select Bit	0 : Software trigger 1 : $\overline{\text{ADTRG}}$ trigger	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **NOTE 2 for the ADCON2 Register**	RW

NOTES :
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.

A/D Control Register 1 (1)



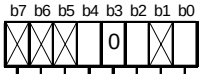
Symbol	Address	After Reset
ADCON1	03D7h	00000XXXb

Symbol	Address	After Reset	RW
— (b0)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
— (b1)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
— (b2)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **NOTE 2 for the ADCON2 Register**	RW
VCUT	Vref Connect Bit (2)	0 : Vref not connected 1 : Vref connected	RW
OPA0	External Op-Amp Connection Mode Bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : Do not set	RW
OPA1			RW

NOTES :
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. If the VCUT bit is reset from “0” (Vref unconnected) to “1” (Vref connected), wait for 1 µs or more before starting A/D conversion.

Figure 16.2 ADCON0 to ADCON1 Registers

A/D Control Register 2 (1)



Symbol	Address	After Reset
ADCON2	03D4h	XXX00000b

Bit Symbol	Bit Name	Function	RW
SMP	A/D Conversion Method Select Bit	0 : Without sample and hold 1 : With sample and hold	RW
— (b1)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
ADGSEL1	A/D Input Group Select Bit	0 : Port P10 group is selected 1 : Port P0 group is selected	RW
— (b3)	Reserved Bit	Set to “0”	RW
CKS2	Frequency Select Bit 2 (2)	0: Selects fAD, fAD divided by 2, or fAD divided by 4. 1: Selects fAD divided by 3, fAD divided by 6, or fAD divided by 12.	RW
— (b7-b5)	Nothing is assigned. When write, set to “0”. When read, their contents are “0”.		—

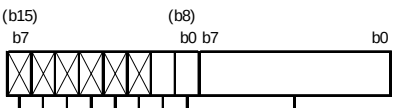
NOTES :

1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. The ØAD frequency must be 10 MHz or less. The selected ØAD frequency is determined by a combination of the CKS0 bit in the ADCON0 register, the CKS1 bit in the ADCON1 register, and the CKS2 bit in the ADCON2 register.

CKS2	CKS1	CKS0	ØAD
0	0	0	Divide-by-4 of fAD
0	0	1	Divide-by-2 of fAD
0	1	0	fAD
0	1	1	
1	0	0	Ddivide-by-12 of fAD
1	0	1	Divide-by-6 of fAD
1	1	0	Divide-by-3 of fAD
1	1	1	

A/D Register i (i=0 to 7)

Symbol	Address	After Reset
AD0	03C1h to 03C0h	Indeterminate
AD1	03C3h to 03C2h	Indeterminate
AD2	03C5h to 03C4h	Indeterminate
AD3	03C7h to 03C6h	Indeterminate
AD4	03C9h to 03C8h	Indeterminate
AD5	03CBh to 03CAh	Indeterminate
AD6	03CDh to 03CCh	Indeterminate
AD7	03CFh to 03CEh	Indeterminate



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

Figure 16.3 ADCON2 and AD0 to AD7 Registers

16.1 Mode Description

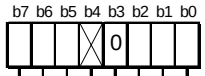
16.1.1 One-Shot Mode

In one-shot mode, analog voltage applied to a selected pin is converted to a digital code once. Table 16.2 shows the One-Shot Mode Specifications. Figures 16.4 and 16.5 shows the ADCON0 and ADCON1 registers in one-shot mode.

Table 16.2 One-Shot Mode Specifications

Item	Specification
Function	The CH2 to CH0 bits in the ADCON0 register, the ADGSEL1 to ADGSEL0 bits in the ADCON2 register and the OPA1 to OPA0 bits in the ADCON1 register select a pin. Analog voltage applied to the pin is converted to a digital code once.
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is “0” (software trigger) The ADST bit in the ADCON0 register is set to “1” (A/D conversion starts) • When the TRG bit is “1” ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to “1” (A/D conversion starts)
A/D Conversion Stop Condition	• Completion of A/D conversion (If a software trigger is selected, the ADST bit is cleared to “0” (A/D conversion halted)) • Set the ADST bit to “0”
Interrupt Request Generation Timing	Completion of A/D conversion
Analog Input Pin	Select one pin from AN0 to AN7, AN0_0 to AN0_7, ANEX0 to ANEX1
Reading of Result of A/D Converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

A/D Control Register 0 (1)



Symbol	Address	After Reset
ADCON0	03D6h	000X0XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit (2, 3)	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0 (3)	0 : One-shot mode	RW
— (b4)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
TRG	Trigger Select Bit	0 : Software trigger 1 : $\overline{\text{ADTRG}}$ trigger	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **NOTE 2 for the ADCON2 Register**	RW

NOTES :
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN0_0 to AN0_7 can be used in the same way as AN0 to AN7. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. After rewriting the MD0 bit, set the CH2 to CH0 bits over again using another instruction.

Figure 16.4 ADCON0 Register (One-shot Mode)

A/D Control Register 1 (1)

b7 b6 b5 b4 b3 b2 b1 b0

1

Symbol	Address	After Reset
ADCON1	03D7h	00000XXXb

Symbol	Address	After Reset	RW
— (b0)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
— (b1)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
— (b2)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **NOTE 2 for the ADCON2 Register**	RW
VCUT	Vref Connect Bit (2)	1 : Vref connected	RW
OPA0	External Op-Amp Connection Mode Bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : Do not set	RW
OPA1			RW

NOTES :
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A/D conversion.

Figure 16.5 ADCON1 Register (One-shot Mode)

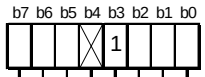
16.1.2 Repeat Mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code. Table 16.3 shows the Repeat Mode Specifications. Figures 16.6 and 16.7 shows the ADCON0 to ADCON1 registers in repeat mode.

Table 16.3 Repeat Mode Specifications

Item	Specification
Function	The CH2 to CH0 bits in the ADCON0 register, the ADGSEL1 to ADGSEL0 bits in the ADCON2 register and the OPA1 to OPA0 bits in the ADCON1 register select a pin. Analog voltage applied to this pin is repeatedly converted to a digital code.
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is “0” (software trigger) The ADST bit in the ADCON0 register is set to “1” (A/D conversion starts) • When the TRG bit is “1” ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to “1” (A/D conversion starts)
A/D Conversion Stop Condition	Set the ADST bit to “0” (A/D conversion halted)
Interrupt Request Generation timing	None generated
Analog Input Pin	Select one pin from AN0 to AN7, AN0_0 to AN0_7, ANEX0 to ANEX1
Reading of Result of A/D Converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

A/D Control Register 0 (1)



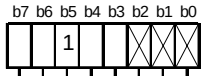
Symbol	Address	After Reset
ADCON0	03D6h	000X0XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit (2, 3)	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected	RW
CH1			RW
CH2			RW
MD0	A/D Operation Mode Select Bit 0 (3)	1 : Repeat mode	RW
— (b4)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
TRG	Trigger Select Bit	0 : Software trigger 1 : $\overline{\text{ADTRG}}$ trigger	RW
ADST	A/D Conversion Start Flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency Select Bit 0	Refer to **NOTE 2 for the ADCON2 Register**	RW

NOTES :
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. AN0_0 to AN0_7 can be used in the same way as AN0 to AN7. Use the ADGSEL1 to ADGSEL0 bits in the ADCON2 register to select the desired pin.
3. After rewriting the MD0 bit, set the CH2 to CH0 bits over again using another instruction.

Figure 16.6 ADCON0 Register (Repeat Mode)

A/D Control Register 1 (1)



Symbol	Address	After Reset
ADCON1	03D7h	00000XXXb

Symbol	Address	After Reset	RW
— (b0)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
— (b1)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
— (b2)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
BITS	8/10-Bit Mode Select Bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency Select Bit 1	Refer to **NOTE 2 for the ADCON2 Register**	RW
VCUT	Vref Connect Bit (2)	1 : Vref connected	RW
OPA0	External Op-Amp Connection Mode Bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : Do not set	RW
OPA1			RW

NOTES :
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be indeterminate.
2. If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A/D conversion.

Figure 16.7 ADCON1 Register (Repeat Mode)

16.2 Function

16.2.1 Resolution Select Function

The desired resolution can be selected using the BITS bit in the ADCON1 register. If the BITS bit is set to “1” (10-bit conversion accuracy), the A/D conversion result is stored in the bit 0 to bit 9 in the ADi register (i = 0 to 7). If the BITS bit is set to “0” (8-bit conversion accuracy), the A/D conversion result is stored in the bit 0 to bit 7 in the ADi register.

16.2.2 Sample and Hold

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

16.2.3 Extended Analog Input Pins

In one-shot and repeat modes, the ANEX0 and ANEX1 pins can be used as analog input pins. Use the OPA1 to OPA0 bits in the ADCON1 register to select whether or not use ANEX0 and ANEX1.
The A/D conversion results of ANEX0 and ANEX1 inputs are stored in the AD0 and AD1 registers, respectively.

16.2.4 Current Consumption Reducing Function

When not using the A/D converter, its resistor ladder and reference voltage input pin (VREF) can be separated using the VCUT bit in the ADCON1 register. When separated, no current will flow from the VREF pin into the resistor ladder, helping to reduce the power consumption of the chip.
To use the A/D converter, set the VCUT bit to “1” (Vref connected) and then set the ADST bit in the ADCON0 register to “1” (A/D conversion start). The VCUT and ADST bits cannot be set to “1” at the same time.
Nor can the VCUT bit be set to “0” (Vref unconnected) during A/D conversion.

16.2.5 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 16.8 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 is generally $$VC = VIN\left\{1 - e^{-\frac{1}{C(R0+R)}t}\right\}$$

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

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

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

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

Figure 16.8 shows Analog Input Pin and External Sensor Equivalent Circuit. When the difference between VIN and VC becomes 0.1LSB, we find impedance R0 when voltage between pins VC changes from 0 to VIN-(0.1/1024) VIN in time T. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB. When f(ϕAD) = 10 MHz, T = 0.3 µs in the A/D conversion mode with sample & hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3 µs, R = 7.8 kΩ, C = 1.5 pF, X = 0.1, and Y = 1024. Hence,

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

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

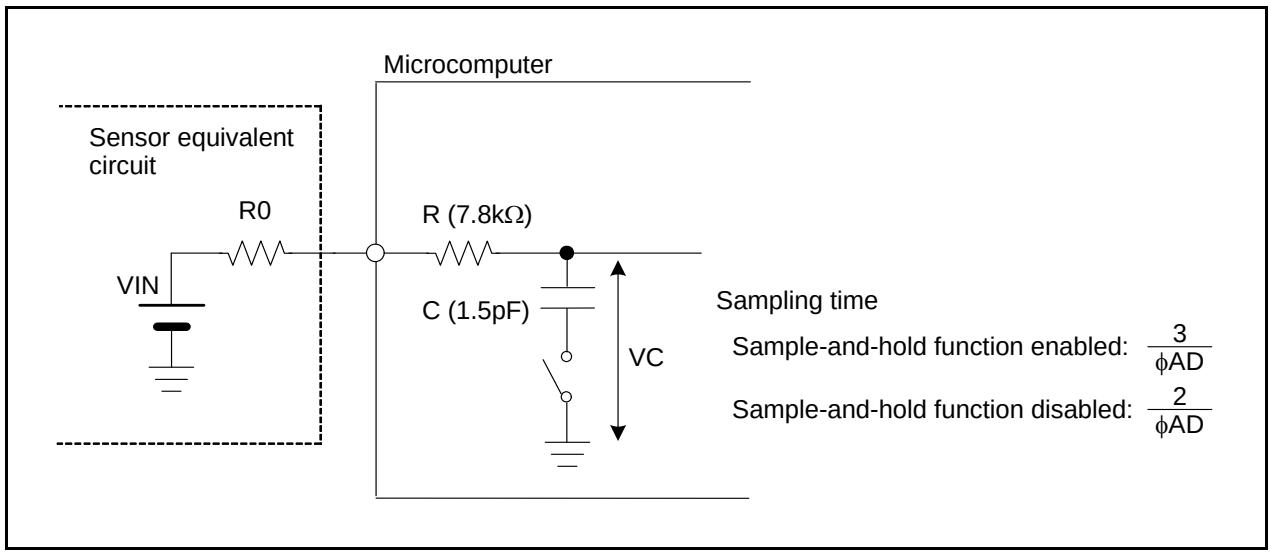


Figure 16.8 Analog Input Pin and External Sensor Equivalent Circuit

17. CRC Calculation

The Cyclic Redundancy Check (CRC) operation detects an error in data blocks. The microcomputer uses a generator polynomial of CRC_CCITT ($X^{16} + X^{12} + X^5 + 1$) to generate CRC code.
The CRC code consists of 16 bits which are generated for each data block in given length, separated in 8 bit units. After the initial value is set in the CRCD register, the CRC code is set in that register each time one byte of data is written to the CRCIN register. CRC code generation for one-byte data is finished in two cycles.
Figure 17.1 shows the CRC Circuit Block Diagram. Figure 17.2 shows the CRC-related registers.
Figure 17.3 shows the calculation example using the CRC operation.

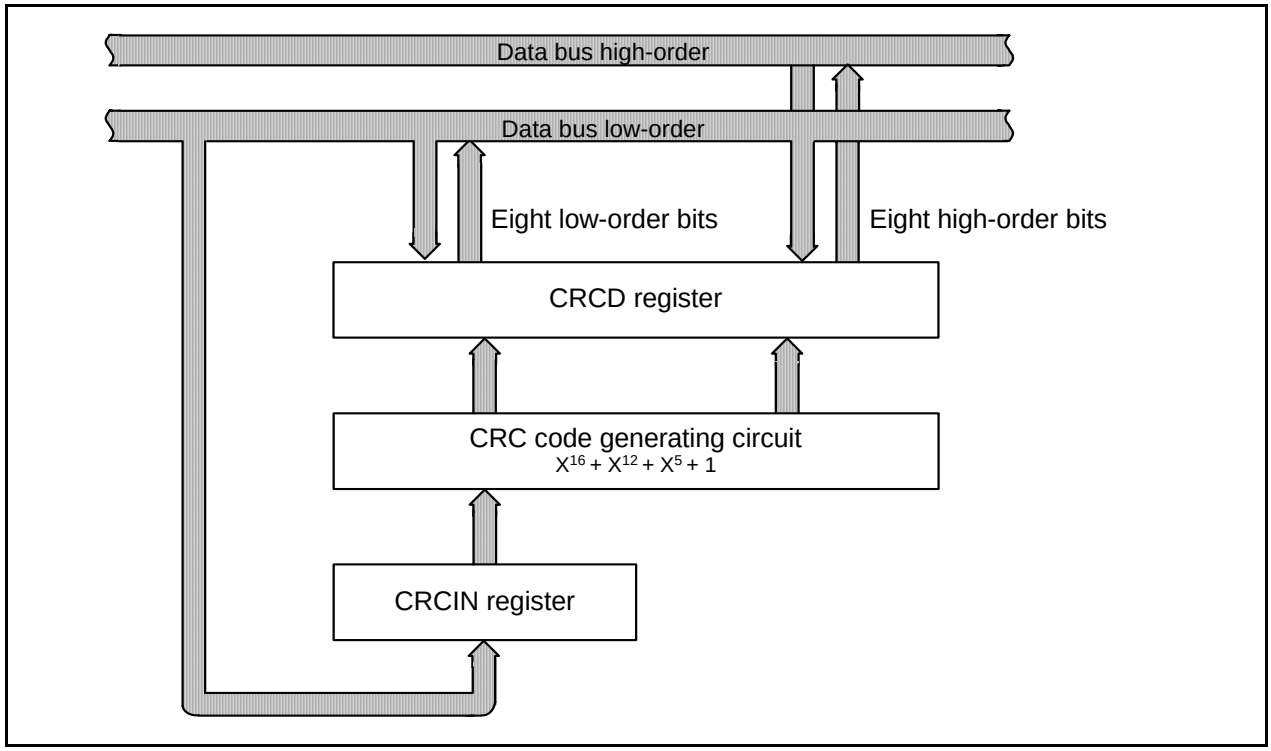


Figure 17.1 CRC Circuit Block Diagram

CRC Data Register

(b15) (b8)
b7 b0 b7 b0

Symbol	Address	After Reset
CRCD	03BDh to 03BCh	Indeterminate

Function	Setting Range	RW
When data is written to the CRCIN register after setting the initial value in the CRCD register, the CRC code can be read out from the CRCD register.	0000h to FFFFh	RW

CRC Input Register

b7 b0

Symbol	Address	After Reset
CRCIN	03BEh	Indeterminate

Function	Setting Range	RW
Data input	00h to FFh	RW

Figure 17.2 CRCD and CRCIN Registers

Setup procedure and CRC operation when generating CRC code “80C4h”

• CRC operation performed by the M16C

CRC code: Remainder of a division in which the value written to the CRCIN register with its bit positions reversed is divided by the generator polynomial

Generator polynomial: $X^{16} + X^{12} + X^5 + 1$ (1 0001 0000 0010 0001b)

• Setting procedure

(1) Reverse the bit positions of the value “80C4h” by program in 1-byte units.

“80h” → “01h”, “C4h” → “23h”

(2) Write 0000h (initial value) → b15 b0 CRCD register

(3) Write 01h → b7 b0 CRCIN register

Two cycles later, the CRC code for “80h,” i.e., 9188h, has its bit positions reversed to become “1189h” which is stored in the CRCD register.

b15 b0 1189h CRCD register

(4) Write 23h → b7 b0 CRCIN register

Two cycles later, the CRC code for “80C4h,” i.e., 8250h, has its bit positions reversed to become “0A41h” which is stored in the CRCD register.

b15 b0 0A41h CRCD register

• Details of CRC operation

As shown in (3) above, bit position of “01h” (00000001b) written to the CRCIN register is inversed and becomes “10000000b”.
Add “1000 0000 0000 0000 0000 0000b”, as “10000000b” plus 16 digits, to “0000 0000 0000 0000 0000 0000b”, as “0000 0000 0000 0000b” plus 8 digits as the default value of the CRCD register to perform the modulo-2 division.

```
                                  1000 1000
1 0001 0000 0010 0001 ) 1000 0000 0000 0000 0000 0000   <- Data
                        1000 1000 0001 0000 1
                             1000 0001 0000 1000 0
                             1000 1000 0001 0000 1
                                  1001 0001 1000 1000
```

Generator polynomial

CRC code

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

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

“0001 0001 1000 1001b (1189h)”, the remainder “1001 0001 1000 1000b (9188h)” with inversed bit position, can be read from the CRCD register.

When going on to (4) above, “23h (00100011b)” written in the CRCIN register is inversed and becomes “11000100b”.
Add “1100 0100 0000 0000 0000 0000b”, as “11000100b” plus 16 digits, to “1001 0001 1000 1000 0000 0000b”, as “1001 0001 1000 1000b” plus 8 digits as a remainder of (3) left in the CRCD register to perform the modulo-2 division.
“0000 1010 0100 0001b (0A41h)”, the remainder with inversed bit position, can be read from CRCD register.

Figure 17.3 CRC Calculation

18. Programmable I/O Ports

The programmable input/output ports (hereafter referred to simply as I/O ports) consist of 87 lines P0 to P10 (except P8_5) for the 100-pin version. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high every 4 lines. P8_5 is an input-only port and does not have a pull-up resistor. Port P8_5 shares the pin with $\overline{\text{NMI}}$, so that the $\overline{\text{NMI}}$ input level can be read from the P8 register P8_5 bit.
Figures 18.1 to 18.5 show the I/O ports. Figure 18.6 shows the I/O Pins.
Each pin functions as an I/O port, a peripheral function input/output, or a bus control pin.
For details on how to set peripheral functions, refer to each functional description in this manual. If any pin is used as a peripheral function input, set the direction bit for that pin to "0" (input mode). Any pin used as an output pin for peripheral functions output, no matter how the corresponding direction bit is set.
When using any pin as a bus control pin, refer to **7.2 Bus Control**.

18.1 Port Pi Direction Register (PDi Register, i = 0 to 10)

Figure 18.7 shows the PDi Registers.
This register selects whether the I/O port is to be used for input or output. The bits in this register correspond one for one to each port.
During memory extension and microprocessor modes, the PDi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}$/$\overline{\text{WR}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$, ALE, $\overline{\text{RDY}}$, $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$, and BCLK) cannot be modified.
No direction register bit for P8_5 is available.

18.2 Port Pi Register (Pi Register, i = 0 to 10)

Figure 18.8 shows the Pi Registers.
Data input/output to and from external devices are accomplished by reading and writing to the Pi register.
The Pi register consists of a port latch to hold the input/output data and a circuit to read the pin status. For ports set for input mode, the input level of the pin can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register.
For ports set for output mode, the port latch can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register. The data written to the port latch is output from the pin. The bits in the Pi register correspond one for one to each port.
During memory extension and microprocessor modes, the Pi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}$/$\overline{\text{WR}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$, ALE, $\overline{\text{RDY}}$, $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$, and BCLK) cannot be modified.

18.3 Pull-up Control Register 0 to Pull-up Control Register 2 (PUR0 to PUR2 Registers)

Figure 18.9 and Figure 18.10 show the PUR0 to PUR2 Registers.
The PUR0 to PUR2 registers 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.
However, the pull-up control register has no effect on P0 to P3, P4_0 to P4_3, and P5 during memory extension and microprocessor modes. Although the register contents can be modified, no pull-up resistors are connected.

18.4 Port Control Register (PCR Register)

Figure 18.10 shows the PCR Register.
When the P1 register is read after setting the PCR0 bit in the PCR register to "1", the corresponding port latch can be read no matter how the PD1 register is set.

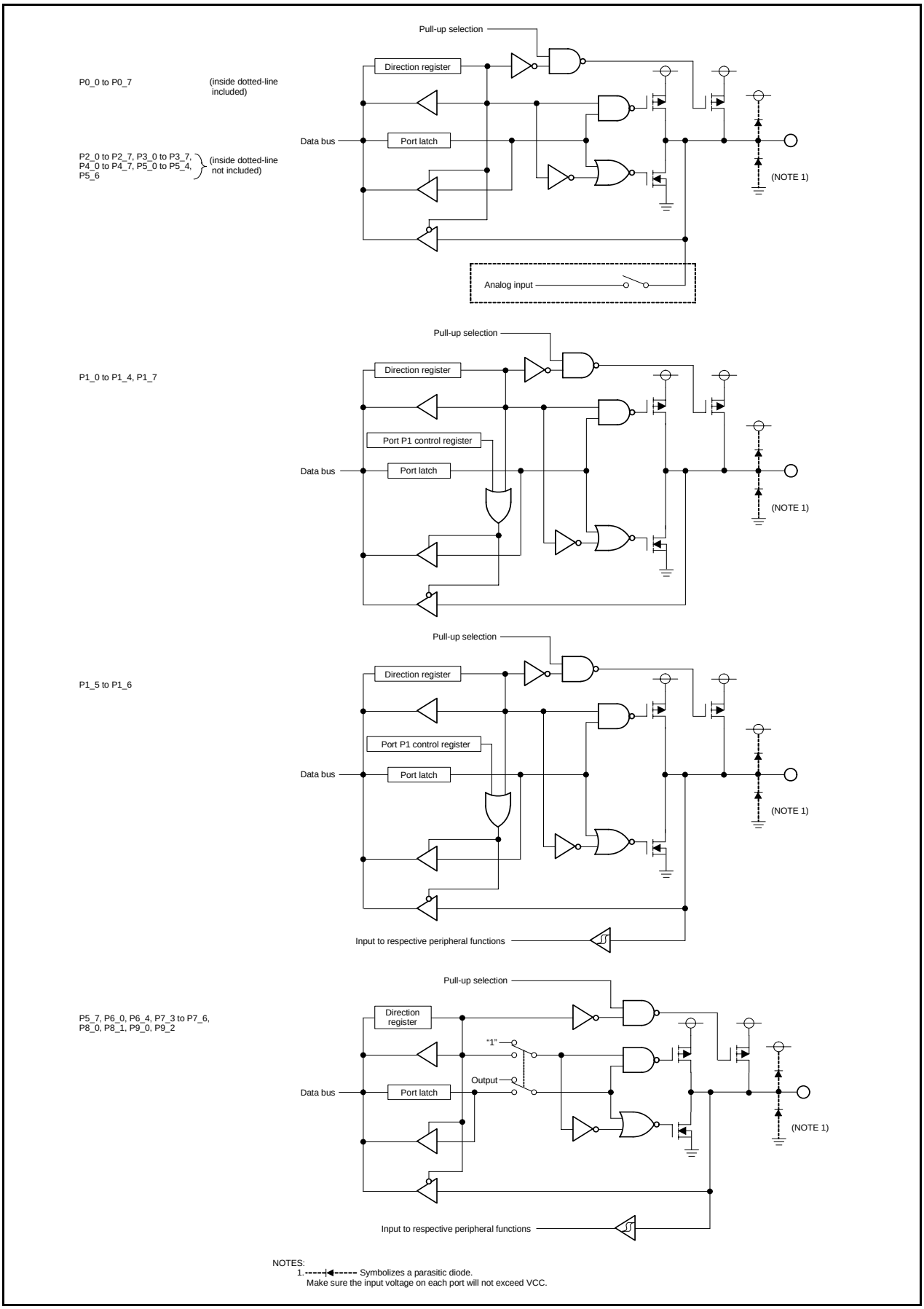
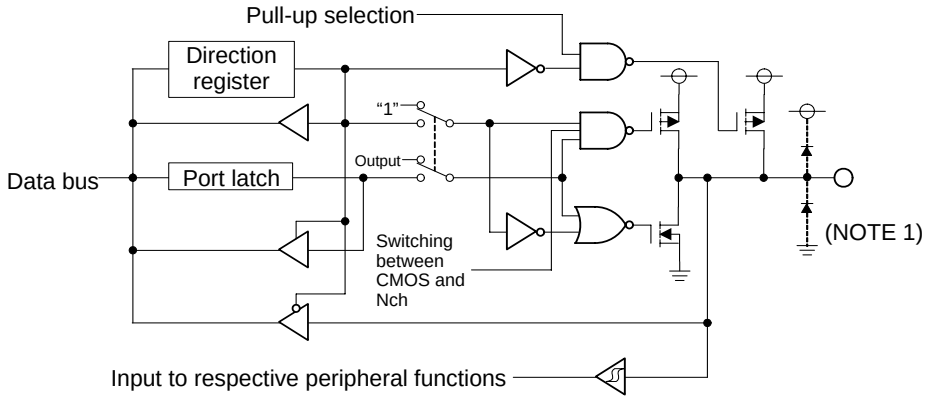
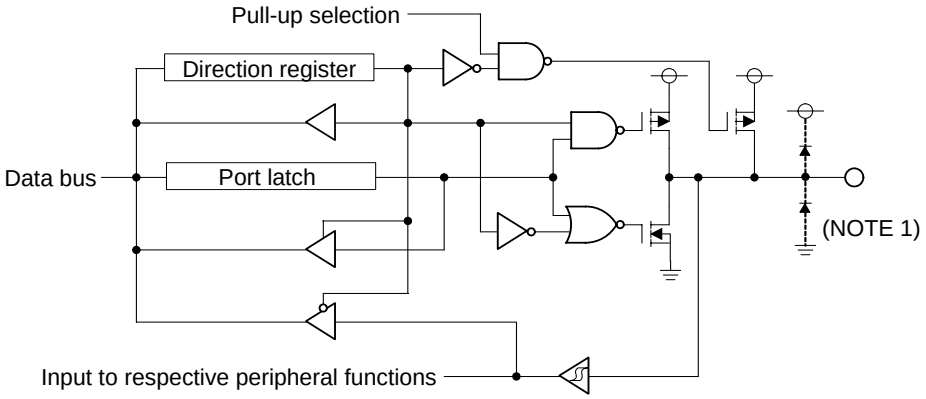


Figure 18.1 I/O Ports (1)

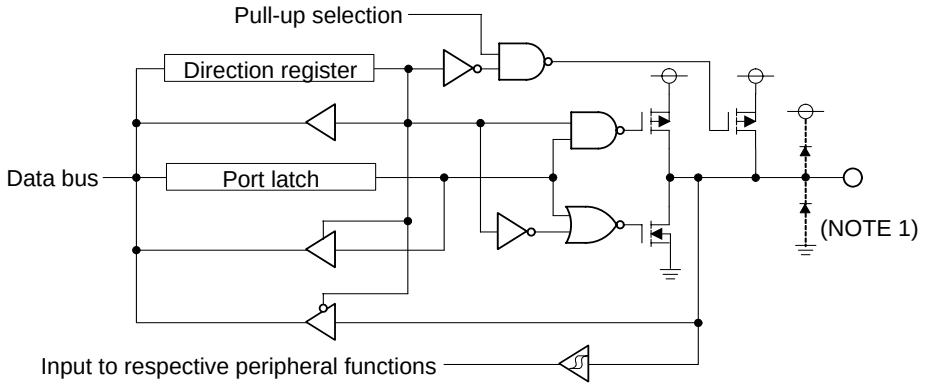
P6_1, P6_5, P7_2



P8_2 to P8_4



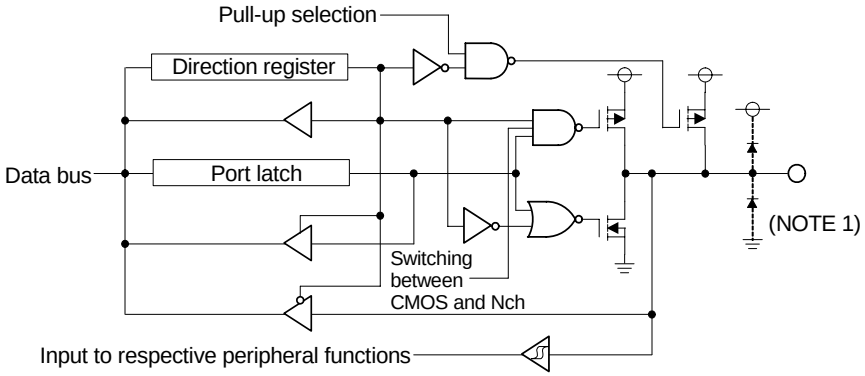
P5_5, P7_7, P9_1, P9_3, P9_4, P9_7



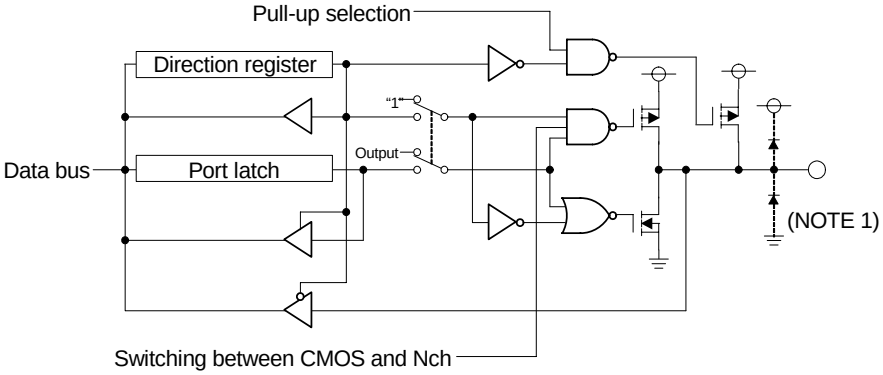
NOTES:
1. ----◂---- Symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed VCC.

Figure 18.2 I/O Ports (2)

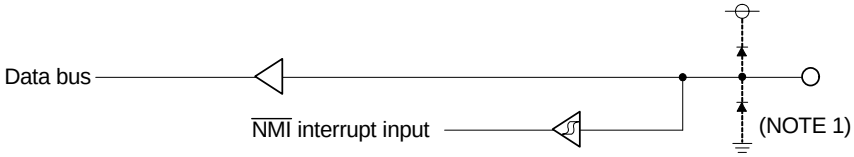
P6_2, P6_6



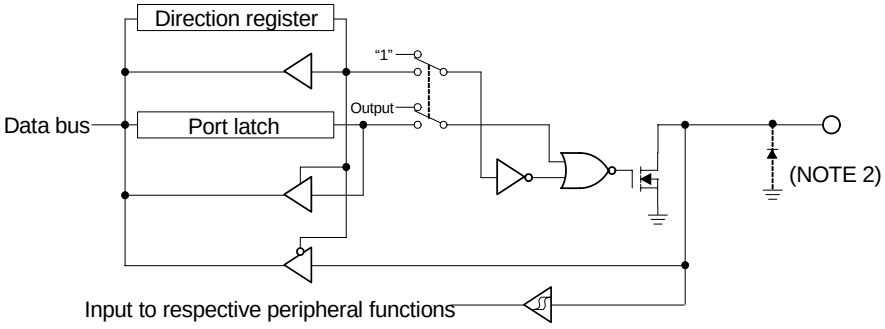
P6_3, P6_7



P8_5



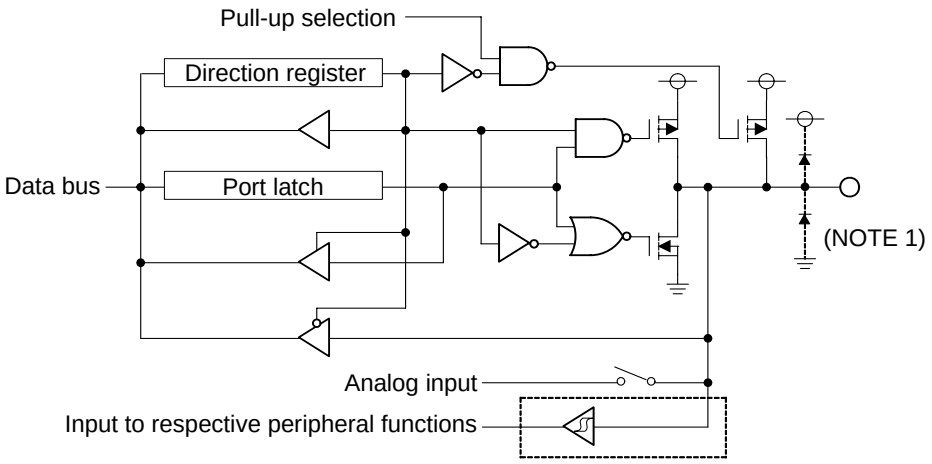
P7_0, P7_1



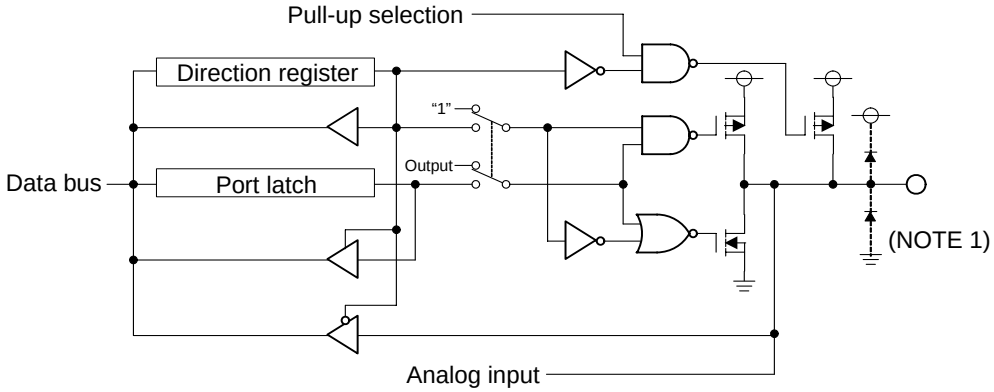
NOTES:
1. ----|◄---- Symbolizes a parasitic diode.
   Make sure the input voltage on each port will not exceed VCC.
2. ----|◄---- Symbolizes a parasitic diode.

Figure 18.3 I/O Ports (3)

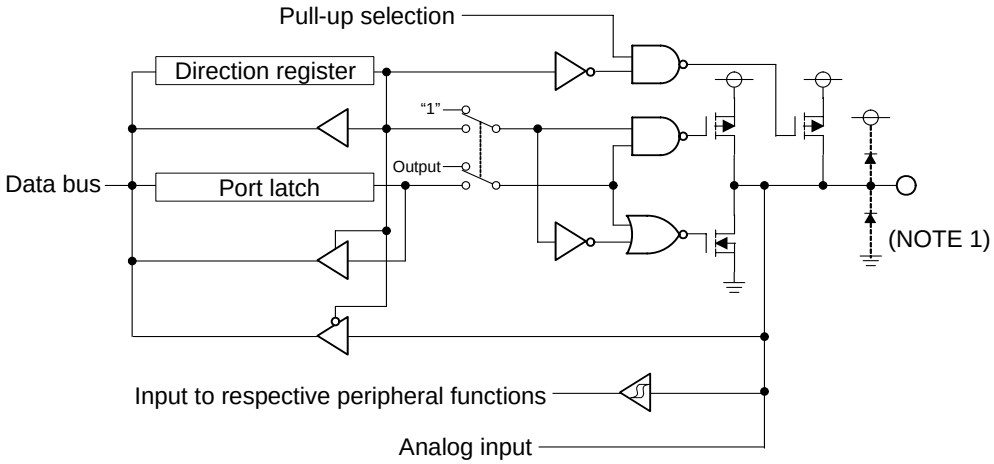
P10_0 to P10_3
(Inside dotted-line not included)
P10_4 to P10_7
(Inside dotted-line included)



P9_6



P9_5



NOTES:
1. ----|◄----- Symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed VCC.

Figure 18.4 I/O Ports (4)

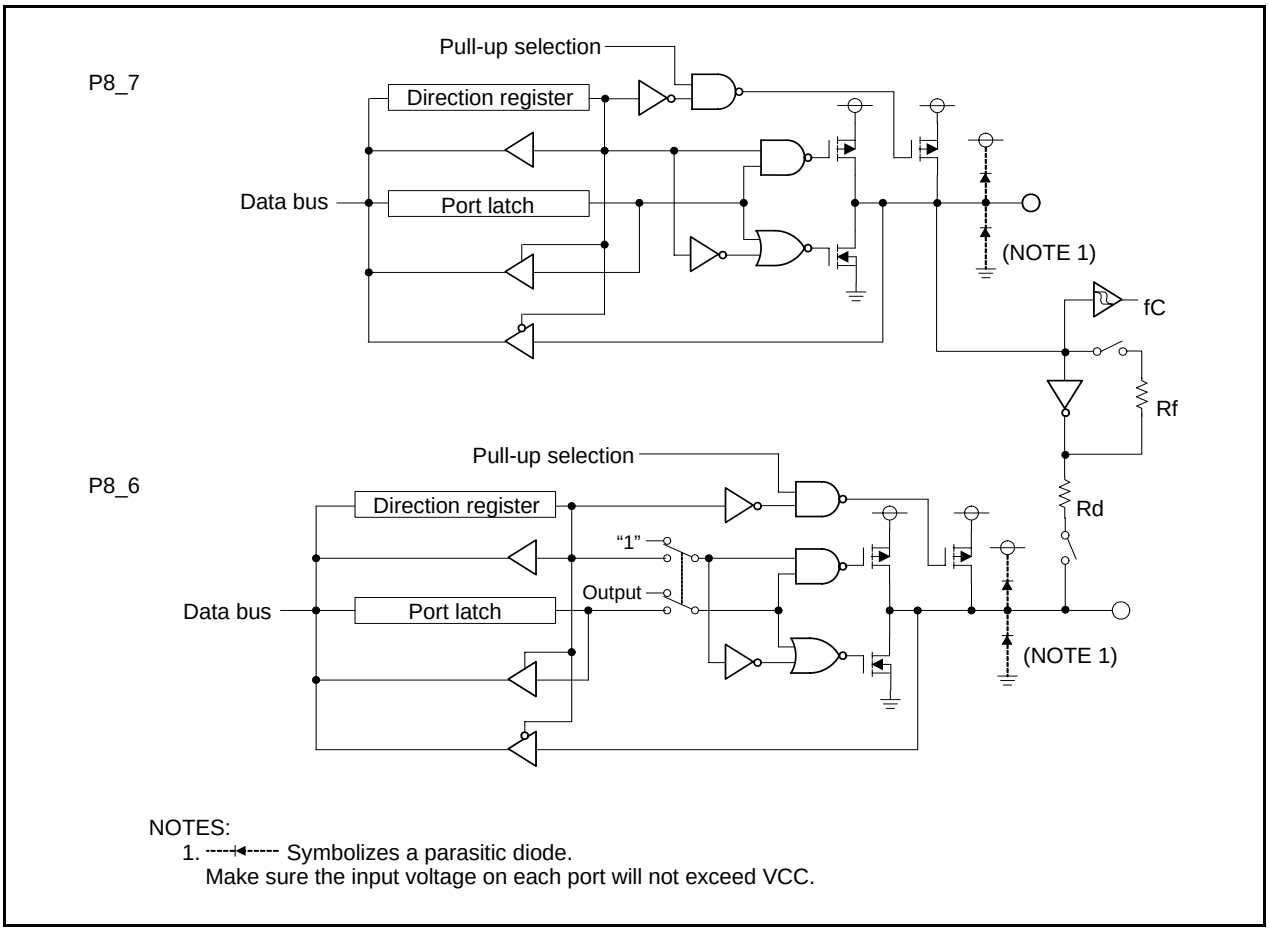


Figure 18.5 I/O Ports (5)

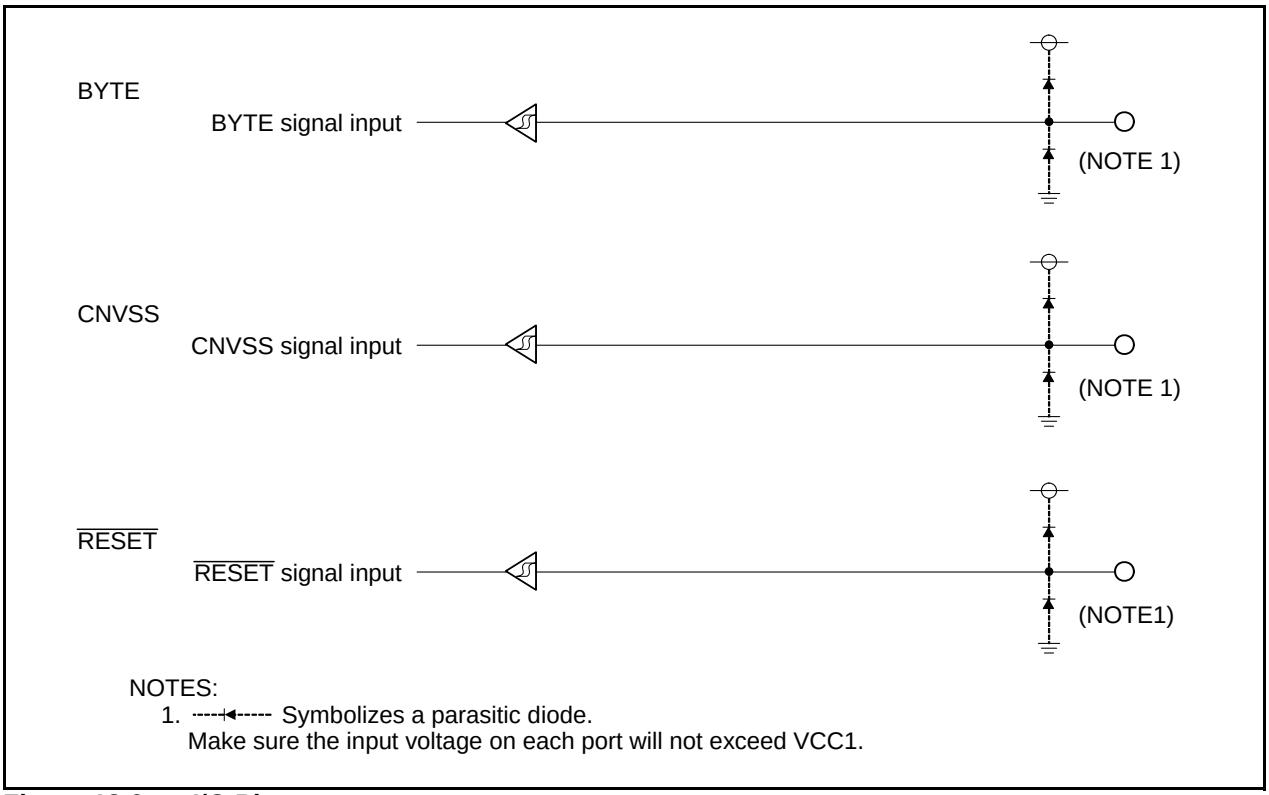
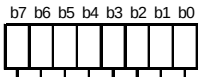


Figure 18.6 I/O Pins

Port Pi Direction Register (i=0 to 7 and 9 to 10) (1, 2)

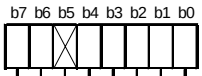


Symbol	Address	After Reset
PD0 to PD3	03E2h, 03E3h, 03E6h, 03E7h	00h
PD4 to PD7	03EAh, 03EBh, 03EEh, 03EFh	00h
PD9 to PD10	03F3h, 03F6h	00h

Bit Symbol	Bit Name	Function	RW
PDi_0	Port Pi_0 Direction Bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 7 and 9 to 10)	RW
PDi_1	Port Pi_1 Direction Bit		RW
PDi_2	Port Pi_2 Direction Bit		RW
PDi_3	Port Pi_3 Direction Bit		RW
PDi_4	Port Pi_4 Direction Bit		RW
PDi_5	Port Pi_5 Direction Bit		RW
PDi_6	Port Pi_6 Direction Bit		RW
PDi_7	Port Pi_7 Direction Bit		RW

NOTES :
1. Make sure the PD9 register is written to by the next instruction after setting the PRC2 bit in the PRCR register to “1” (write enabled).
2. During memory extension and microprocessor modes, the PDi register for the pins functioning as bus control pins (A0 to A19, D0 to D15, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK) cannot be modified.

Port P8 Direction Register



Symbol	Address	After Reset
PD8	03F2h	00X00000b

Bit Symbol	Bit Name	Function	RW
PD8_0	Port P8_0 Direction Bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_1	Port P8_1 Direction Bit		RW
PD8_2	Port P8_2 Direction Bit		RW
PD8_3	Port P8_3 Direction Bit		RW
PD8_4	Port P8_4 Direction Bit		RW
— (b5)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
PD8_6	Port P8_6 Direction Bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_7	Port P8_7 Direction Bit		RW

Figure 18.7 PDi Registers

Port Pi Register (i=0 to 7 and 9 to 10) (2)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
P0 to P3	03E0h, 03E1h, 03E4h, 03E5h	Indeterminate
P4 to P7	03E8h, 03E9h, 03ECh, 03EDh	Indeterminate
P9 to P10	03F1h, 03F4h	Indeterminate

Bit Symbol	Bit Name	Function	RW
Pi_0	Port Pi_0 Bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level (1) (i = 0 to 7 and 9 to 10)	RW
Pi_1	Port Pi_1 Bit		RW
Pi_2	Port Pi_2 Bit		RW
Pi_3	Port Pi_3 Bit		RW
Pi_4	Port Pi_4 Bit		RW
Pi_5	Port Pi_5 Bit		RW
Pi_6	Port Pi_6 Bit		RW
Pi_7	Port Pi_7 Bit		RW

NOTES :
1. Since P7_0 and P7_1 are N-channel open drain ports, the data is high-impedance.
2. During memory extension and microprocessor modes, the Pi register for the pins functioning as bus control pins (A0 to A19, D0 to D15, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK) cannot be modified.

Port P8 Register

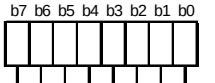
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
P8	03F0h	Indeterminate

Bit Symbol	Bit Name	Function	RW
P8_0	Port P8_0 Bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register (except for P8_5) 0 : "L" level 1 : "H" level	RW
P8_1	Port P8_1 Bit		RW
P8_2	Port P8_2 Bit		RW
P8_3	Port P8_3 Bit		RW
P8_4	Port P8_4 Bit		RW
P8_5	Port P8_5 Bit		RO
P8_6	Port P8_6 Bit		RW
P8_7	Port P8_7 Bit		RW

Figure 18.8 Pi Registers

Pull-up Control Register 0 (2)

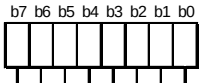


Symbol	Address	After Reset
PUR0	03FCh	00h

Bit Symbol	Bit Name	Function	RW
PU00	P0_0 to P0_3 Pull-Up	0 : Not pulled high 1 : Pulled high (1)	RW
PU01	P0_4 to P0_7 Pull-Up		RW
PU02	P1_0 to P1_3 Pull-Up		RW
PU03	P1_4 to P1_7 Pull-Up		RW
PU04	P2_0 to P2_3 Pull-Up		RW
PU05	P2_4 to P2_7 Pull-Up		RW
PU06	P3_0 to P3_3 Pull-Up		RW
PU07	P3_4 to P3_7 Pull-Up		RW

NOTES :
1. The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.
2. During memory extension and microprocessor modes, the pins are not pulled high although their corresponding register contents can be modified.

Pull-Up Control Register 1



Symbol	Address	After Reset (5)
PUR1	03FDh	00000000b 00000010b

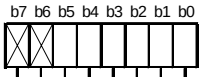
Bit Symbol	Bit Name	Function	RW
PU10	P4_0 to P4_3 Pull-Up (3)	0 : Not pulled high 1 : Pulled high (2)	RW
PU11	P4_4 to P4_7 Pull-Up (4)		RW
PU12	P5_0 to P5_3 Pull-Up (3)		RW
PU13	P5_4 to P5_7 Pull-Up (3)		RW
PU14	P6_0 to P6_3 Pull-Up		RW
PU15	P6_4 to P6_7 Pull-Up		RW
PU16	P7_2 to P7_3 Pull-Up (1)		RW
PU17	P7_4 to P7_7 Pull-Up		RW

NOTES :
1. The P7_0 and P7_1 pins do not have pull-ups.
2. The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.
3. During memory extension and microprocessor modes, the pins are not pulled high although the contents of these bits can be modified.
4. If the PM01 to PM00 bits in the PM0 register are set to “01b” (memory expansion mode) or “11b” (microprocessor mode) in a program during single-chip mode, the PU11 bit becomes “1”.
5. The values after hardware reset is as follows:
   • 00000000b when input on CNVSS pin is “L”
   • 00000010b when input on CNVSS pin is “H”

   The values after software reset is as follows:
   • 00000000b when PM01 to PM00 bits are “00b” (single-chip mode)
   • 00000010b when PM01 to PM00 bits are “01b” (memory expansion mode) or “11b” (microprocessor mode)

Figure 18.9 PUR0 to PUR1 Registers

Pull-Up Control Register 2

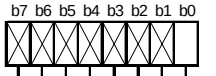


Symbol	Address	After Reset
PUR2	03FEh	00h

Bit Symbol	Bit Name	Function	RW
PU20	P8_0 to P8_3 Pull-Up	0 : Not pulled high	RW
PU21	P8_4 to P8_7 Pull-Up (2)	1 : Pulled high (1)	RW
PU22	P9_0 to P9_3 Pull-Up		RW
PU23	P9_4 to P9_7 Pull-Up		RW
PU24	P10_0 to P10_3 Pull-Up		RW
PU25	P10_4 to P10_7 Pull-Up		RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, their contenta are “0”.		—

NOTES :
1. The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.
2. The P8_5 pin does not have pull-up.

Port Control Register



Symbol	Address	After Reset
PCR	03FFh	00h

Bit Symbol	Bit Name	Function	RW
PCR0	Port P1 Control Bit	Operation performed when the P1 register is read 0 : When the port is set for input, the input levels of P1_0 to P1_7 pins are read. When set for output, the port latch is read. 1 : The port latch is read regardless of whether the port is set for input or output.	RW
— (b7-b1)	Nothing is assigned. When write, set to “0”. When read, their contents are “0”.		—

Figure 18.10 PUR2 and PCR Registers

Table 18.1 Unassigned Pin Handling in Single-chip Mode

Pin Name	Connection
Ports P0 to P7, P8_0 to P8_4, P8_6 to P8_7, P9 to P10	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open. (1, 2, 3)
XOUT (4)	Open
$\overline{\text{NMI}}$ (P8_5)	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF, BYTE	Connect to VSS

NOTES:

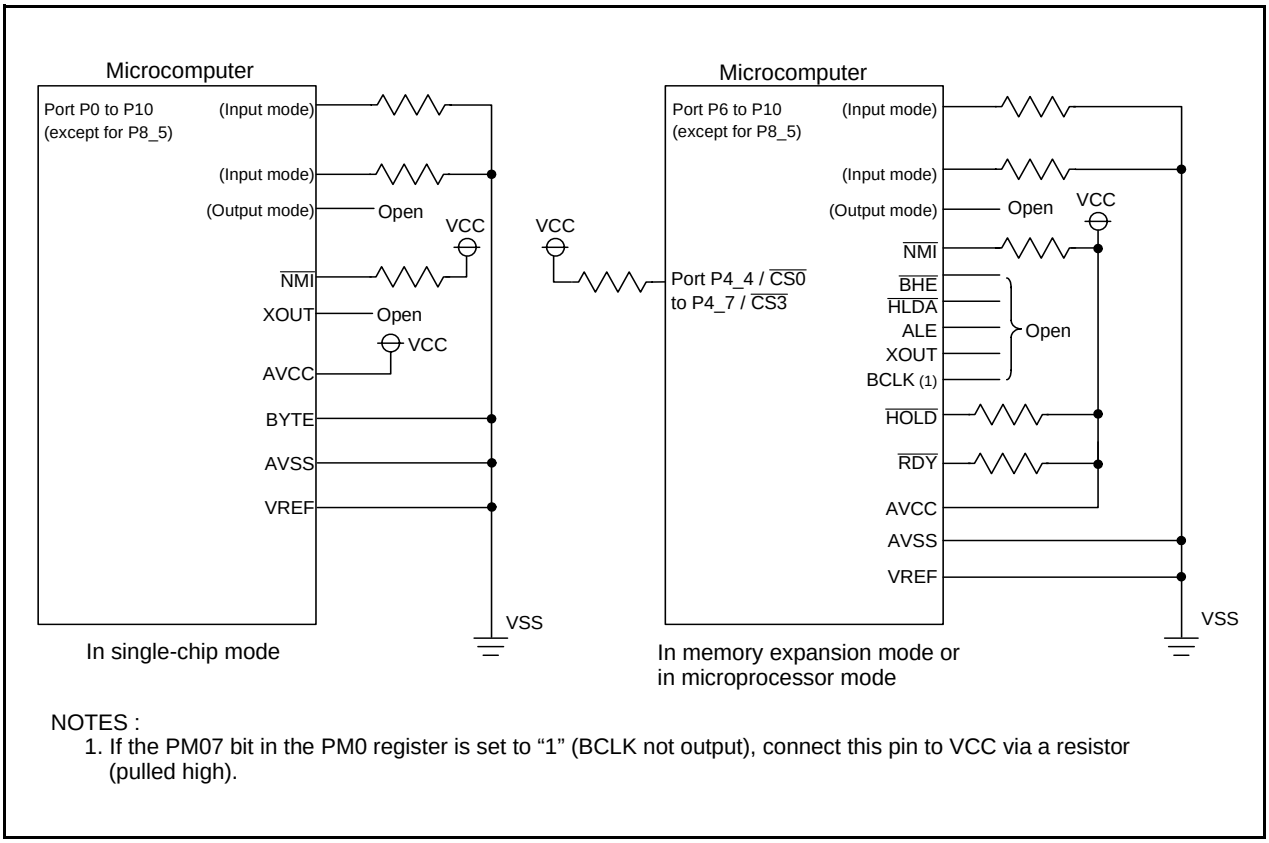
1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes indeterminate, causing the power supply current to increase while the port remains in input mode.
   Furthermore, by considering a possibility that the contents of the direction registers could be changed by noise or noise-induced runaway, it is recommended that the contents of the direction registers be periodically reset in software, for the increased reliability of the program.
2. Make sure the unused pins are processed with the shortest possible wiring from the microcomputer pins (within 2 cm).
3. When the ports P7_0 and P7_1 are set for output mode, make sure a low-level signal is output from the pins.
   The ports P7_0 and P7_1 are N-channel open-drain outputs.
4. With external clock input to XIN pin.

Table 18.2 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode

Pin Name	Connection
Ports P0 to P7, P8_0 to P8_4, P8_6 to P8_7, P9 to P10	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open. (1, 2, 3, 4)
P4_4/$\overline{CS0}$ to P4_7/$\overline{CS3}$	Connect to VCC via a resistor (pulled high) by setting the corresponding direction bit in the PD4 register for $\overline{CSi}$ (i=0 to 3) to “0” (input mode) and the CSi bit in the CSR register to “0” (chip select disabled).
$\overline{BHE}$, ALE, $\overline{HLDA}$, XOUT (5), BCLK (6)	Open
$\overline{HOLD}$, $\overline{RDY}$	Connect via resistor to VCC (pull-up)
$\overline{NMI}$ (P8_5)	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF	Connect to VSS

NOTES:

1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes indeterminate, causing the power supply current to increase while the port remains in input mode.
   Furthermore, by considering a possibility that the contents of the direction registers could be changed by noise or noise-induced runaway, it is recommended that the contents of the direction registers be periodically reset in software, for the increased reliability of the program.
2. Make sure the unused pins are processed with the shortest possible wiring from the microcomputer pins (within 2 cm).
3. If the CNVSS pin has the VSS level applied to it, these pins are set for input ports until the processor mode is switched over in a program after reset. For this reason, the voltage levels on these pins become indeterminate, causing the power supply current to increase while they remain set for input ports.
4. When the ports P7_0 and P7_1 are set for output mode, make sure a low-level signal is output from the pins.
   The ports P7_0 and P7_1 are N-channel open-drain outputs.
5. With external clock input to XIN pin.
6. If the PM07 bit in the PM0 register is set to “1” (BCLK not output), connect this pin to VCC via a resistor (pulled high).



NOTES :
1. If the PM07 bit in the PM0 register is set to “1” (BCLK not output), connect this pin to VCC via a resistor (pulled high).

Figure 18.11 Unassigned Pins Handling

19. Flash Memory Version

Aside from the built-in flash memory, the flash memory version microcomputer has the same functions as the masked ROM version.
In the flash memory version, the flash memory can perform in three rewrite modes: CPU rewrite mode, standard serial I/O mode and parallel I/O mode.
Table 19.1 lists specifications of the flash memory version. See **Table 1.1 Performance Outline of M16C/30P Group** for the items not listed in Table 19.1.

Table 19.1 Flash Memory Version Specifications

Item		Specification
Flash Memory Rewrite Mode		3 modes (CPU rewrite, standard serial I/O, parallel I/O)
Erase Block	User ROM Area	See **Figure 19.1 Flash Memory Block Diagram**
	Boot ROM Area	1 block (4 Kbytes) (1)
Program Method		In units of word
Erase Method		Block erase
Program and Erase Control Method		Program and erase controlled by software command
Protect Method		The lock bit protects each block
Number of Commands		8 commands
Program and Erase Endurance		100 times (2)
Data Retention		10 years
ROM Code Protection		Parallel I/O and standard serial I/O modes are supported

NOTES:
1. The boot ROM area contains a standard serial I/O mode rewrite control program which is stored in it when shipped from the factory. This area can only be rewritten in parallel input/output mode.
2. Definition of program and erase endurance
   The programming and erasure times are defined to be per-block erasure times. For example, assume a case where a 4-Kbyte block A is programmed in 2,048 operations by writing one word at a time and erased thereafter.
   In this case, the block is reckoned as having been programmed and erased once.
   If a product is 100 times of programming and erasure, each block in it can be erased up to 100 times.

Table 19.2 Flash Memory Rewrite Modes Overview

Flash Memory Rewrite Mode	CPU rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	The User ROM area is rewritten when the CPU executes software commands. EW0 mode: Rewrite in areas other than flash memory (1) EW1 mode: Can be rewritten in the flash memory	The user ROM area is rewritten using a dedicated serial programmer. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART	The boot ROM area and user ROM area is rewritten using a dedicated parallel programmer.
Areas which can be Rewritten	User ROM area	User ROM area	User ROM area Boot ROM area
Operating Mode	Single-chip mode Memory expansion mode (EW0 mode) Boot mode (EW0 mode)	Boot mode	Parallel I/O mode
ROM Programmer	None	Serial programmer	Parallel programmer

NOTES:
1. When in CPU mode, the PM10 bit in the PM1 register is set to "1". Execute the rewrite control program in the internal RAM or in an external area usable when the PM10 bit is "1".

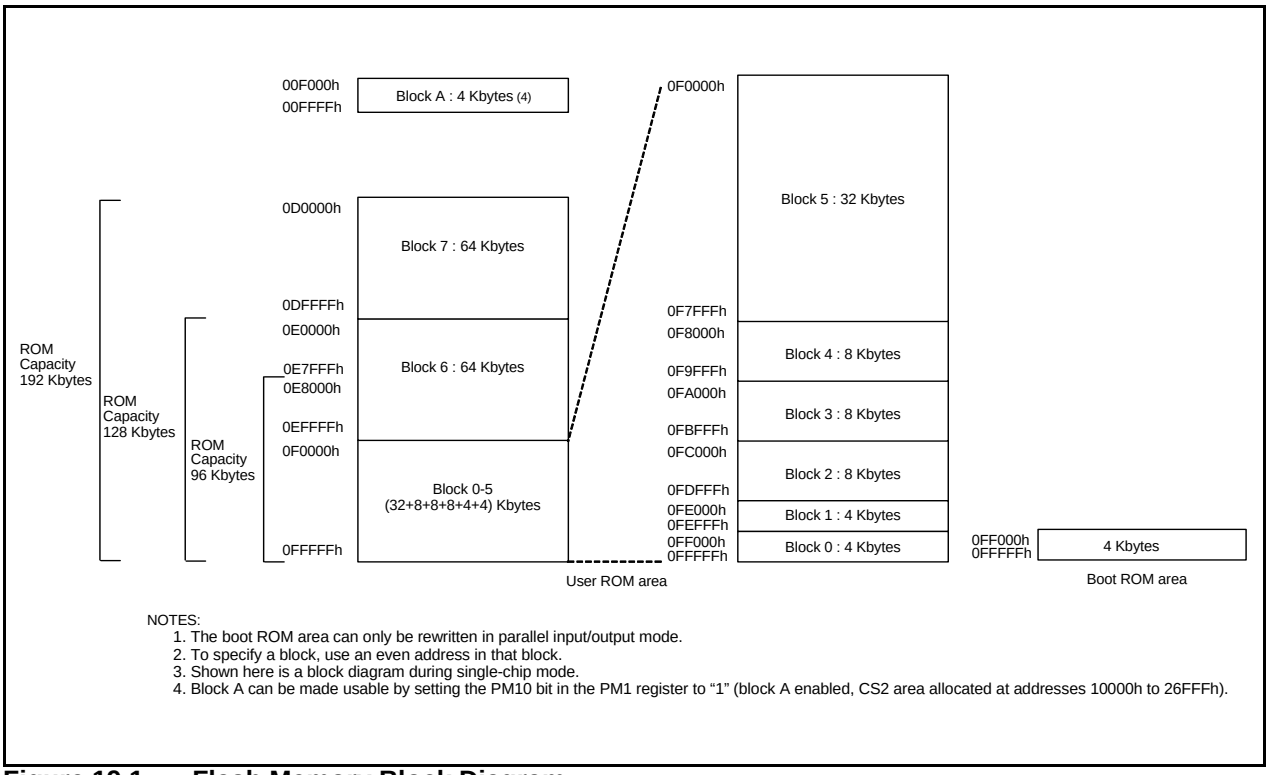
19.1 Memory Map

The flash memory contains the user ROM area and the boot ROM area. The user ROM area has space to store the microcomputer operating program in single-chip mode or memory expansion mode and a separate 4-Kbyte space as the block A. Figure 19.1 shows a Flash Memory Block Diagram.

The user ROM area is divided into several blocks, each of which can be protected (locked) from program or erase. The user ROM area can be rewritten in CPU rewrite, standard serial I/O and parallel I/O modes.

Block A is enabled for use by setting the PM10 bit in the PM1 register to “1” (block A enabled, CS2 area at addresses 10000h to 26FFFh).

The boot ROM area is located at the same addresses as the user ROM area. It can only be rewritten in parallel I/O mode (refer to **19.1.1 Boot Mode**). A program in the boot ROM area is executed after a hardware reset occurs while an “H” signal is applied to the CNVSS and P5_0 pins and an “L” signal is applied to the P5_5 pin (refer to **19.1.1 Boot Mode**). A program in the user ROM area is executed after a hardware reset occurs while an “L” signal is applied to the CNVSS pin. However, the boot ROM area cannot be read.



NOTES:
1. The boot ROM area can only be rewritten in parallel input/output mode.
2. To specify a block, use an even address in that block.
3. Shown here is a block diagram during single-chip mode.
4. Block A can be made usable by setting the PM10 bit in the PM1 register to “1” (block A enabled, CS2 area allocated at addresses 10000h to 26FFFh).

Figure 19.1 Flash Memory Block Diagram

19.1.1 Boot Mode

The microcomputer enters boot mode when a hardware reset occurs while an “H” signal is applied to the CNVSS and P5_0 pins and an “L” signal is applied to the P5_5 pin. A program in the boot ROM area is executed.
In boot mode, the FMR05 bit in the FMR0 register selects access to the boot ROM area or the user ROM area.
The rewrite control program for standard serial I/O mode is stored in the boot ROM area before shipment.
The boot ROM area can be rewritten in parallel I/O mode only. If any rewrite control program using erase-write mode (EW0 mode) is written in the boot ROM area, the flash memory can be rewritten according to the system implemented.

19.2 Functions To Prevent Flash Memory from Rewriting

The flash memory has a built-in ROM code protect function for parallel I/O mode and a built-in ID code check function for standard I/O mode to prevent the flash memory from reading or rewriting.

19.2.1 ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten during parallel input/output mode. Figure 19.2 shows the ROMCP Address. The ROMCP address is located in the user ROM area.
The ROM code protect function is enabled when the ROMCR bits are set to other than “11b”. In this case, set the bit 5 to bit 0 to “111111b”.
When exiting ROM code protect, erase the block including the ROMCP address by the CPU rewrite mode or the standard serial I/O mode.

19.2.2 ID Code Check Function

Use the ID code check function in standard serial I/O mode. The ID code sent from the serial programmer is compared with the ID code written in the flash memory for a match. If the ID codes do not match, commands sent from the serial programmer are not accepted. However, if the four bytes of the reset vector are “FFFFFFFFh”, ID codes are not compared, allowing all commands to be accepted.

The ID codes are 7-byte data stored consecutively, starting with the first byte, into addresses 0FFFDFh, 0FFFE3h, 0FFFEBh, 0FFFEFh, 0FFFF3h, 0FFFF7h, and 0FFFFBh. The flash memory must have a program with the ID codes set in these addresses.

Figure 19.3 shows address for ID code stored.

Reserved character sequence of the ASCII codes: “A”, “L”, “e”, “R”, “A”, “S”, and “E” are used for forced erase function. Table 19.3 lists reserved character sequence.

When the ID codes stored in the ID code addresses in the user ROM area are set to the ASCII codes: “A”, “L”, “e”, “R”, “A”, “S”, and “E” as the combination table listed in Table 19.3, forced erase function becomes active. Use the sequence only when forced erase function is necessary.

Table 19.3 Reserved Character Sequence (Reserved Word)

ID Code Address		ID Code
		Hexadecimal Code (ASCII)
FFFDFh	ID1	41h (A)
FFFE3h	ID2	4Ch (L)
FFFEBh	ID3	65h (e)
FFFEFh	ID4	52h (R)
FFFF3h	ID5	41h (A)
FFFF7h	ID6	53h (S)
FFFFBh	ID7	45h (E)

Reserve word for forced erase function: A set of reserved characters that match all the ID code addresses in sequence as the combination table listed in Table 19.3.

19.2.3 Forced Erase Function

This function is available only in standard serial I/O mode.
When the reserved characters, “A”, “L”, “e”, “R”, “A”, “S”, and “E” in ASCII code, are sent from the serial programmer as ID codes, the content of the user ROM area will be erased at once. However, if the ID codes stored in the ID code addresses in the user ROM area are set to other than a reserved word “ALeRASE” (other than the combination table listed in Table 19.3) when the ROMCP bit in the ROMCP address is set to other than 11b (ROM code protect enabled), forced erase function is ignored and ID code check is executed. Table 19.4 lists conditions and functions for forced erase function.
When both the ID codes sent from the serial programmer and the ID codes stored in the ID code addresses correspond to the reserved word ALeRASE”, the user ROM area will be erased. However, when the serial programmer sends other than “ALeRASE”, even if the ID codes stored in the ID code addresses are “ALeRASE”, there is no ID match and any command is ignored. The user ROM area remains protected accordingly.

Table 19.4 Forced Erase Function

Condition			Function
ID code from serial programmer	Code in ID code stored address	ROMCP1 bit in the ROMCP address	
ALeRASE	ALeRASE	–	User ROM area all erase (forced erase function)
	Other than ALeRASE	11b (ROM code protect disabled)	
		00b, 01b, 10b (ROM code protect enabled)	ID code check
Other than ALeRASE	ALeRASE	–	ID code check (no ID match)
	Other than ALeRASE	–	ID code check

ROM Code Protect Control Address (5)

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

Symbol: ROMCP — Address: 0FFFFFh — Factory Setting: FFh (4)

Bit Symbol	Bit Name	Function	RW
— (b5-b0)	Reserved Bit	Set to "1"	RW
ROMCP1	ROM Code Protect Level 1 Set Bit (1, 2, 3, 4)	b7 b6 0 0 : 0 1 : 1 0 : ROM code protection active 1 1 : ROM code protection inactive	RW

NOTES :
1. When the ROM code protection is active by the ROMCP1 bit setting, the flash memory is protected against reading or rew riting in parallel I/O mode.
2. Set the bit 5 to bit 0 to "111111b" w hen the ROMCP1 bit is set to a value other than "11b".
   If the bit 5 to bit 0 are set to values other than "111111b", the ROM code protection may not become active by setting the ROMCP1 bit to a value other than "11b".
3. To make the ROM code protection inactive, erase a block including the ROMCP address in standard serial I/O mode or CPU rew rite mode.
4. The ROMCP address is set to "FFh" w hen a block, including the ROMCP address, is erased.
5. When a value of the ROMCP address is "00h" or "FFh", the ROM code protect function is disabled.

Figure 19.2 ROMCP Address

Address		
0FFFDFh to 0FFFDCh	ID1	Undefined instruction vector
0FFFE3h to 0FFFE0h	ID2	Overflow vector
0FFFE7h to 0FFFE4h		BRK instruction vector
0FFFEBh to 0FFFE8h	ID3	Address match vector
0FFFEFh to 0FFFECh	ID4	Single step vector
0FFFF3h to 0FFFF0h	ID5	Watchdog timer vector
0FFFF7h to 0FFFF4h	ID6	$\overline{\text{DBC}}$ vector
0FFFFBh to 0FFFF8h	ID7	$\overline{\text{NMI}}$ vector
0FFFFFh to 0FFFFCh	ROMCP	Reset vector

4 bytes

Figure 19.3 Address for ID Code Stored

19.3 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten when the CPU executes software commands.
The user ROM area can be rewritten with the microcomputer mounted on a board without using a parallel or serial programmer.
In CPU rewrite mode, only the user ROM area shown in Figure 19.1 can be rewritten. The boot ROM area cannot be rewritten. Program and the block erase command are executed only in the user ROM area.
Erase-write 0 (EW0) mode and erase-write 1 (EW1) mode are provided as CPU rewrite mode. Table 19.5 lists differences between erase-write 0 (EW0) and erase-write 1 (EW1) modes.

Table 19.5 EW0 Mode and EW1 Mode

Item	EW0 Mode	EW1 Mode
Operating Mode	• Single-chip mode • Memory expansion mode • Boot mode	• Single-chip mode
Space where the rewrite control program can be placed	• User ROM area • Boot ROM area	• User ROM area
Space where the rewrite control program can be executed	The rewrite control program must be transferred to any space other than the flash memory (e.g., RAM) before being executed (2)	The rewrite control program can be executed in the user ROM area
Space which can be rewritten	User ROM area	User ROM area However, this excludes blocks with the rewrite control program
Software Command Restriction	None	• Program and block erase commands cannot be executed in a block having the rewrite control program. • Read status register command cannot be used.
Mode after Program or Erasing	Read status register mode	Read array mode
CPU State during Auto Write and Auto Erase	Operating	Maintains hold state (I/O ports maintains the state before the command was executed) (1)
Flash Memory State Detection	• Read the FMR00, FMR06 and FMR07 bits in the FMR0 register by program • Execute the read status register command to read the SR7, SR5 and SR4 bits in the status register.	Read the FMR00, FMR06 and FMR07 bits in the FMR0 register by program

NOTES:
1. Do not generate an interrupt (except $\overline{\text{NMI}}$ interrupt) or DMA transfer.
2. When in CPU mode, the PM10 bit in the PM1 register is set to "1". Execute the rewrite control program in the internal RAM or in an external area usable when the PM10 bit is "1".

19.3.1 EW0 Mode

The microcomputer enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to “1” (CPU rewrite mode enabled) and is ready to accept commands. EW0 mode is selected by setting the FMR11 bit in the FMR1 register to “0”. To set the FMR01 bit to “1”, set to “1” after first writing “0”.
The software commands control programming and erasing. The FMR0 register or the status register indicates whether a program or erase operation is completed as expected or not.

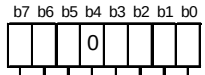
19.3.2 EW1 Mode

EW1 mode is selected by setting the FMR11 bit to “1” after the FMR01 bit is set to “1”. (Both bits must be set to “0” first before setting to “1”.)
The FMR0 register indicates whether or not a program or erase operation has been completed as expected. The status register cannot be read in EW1 mode.
When an erase/program operation is initiated the CPU halts all program execution until the operation is completed or erase-suspend is requested.

19.3.3 Flash Memory Control Register (FMR0 and FMR1 registers)

Figure 19.4 to Figure 19.5 show the FMR0 and FMR1 Registers.

Flash Memory Control Register 0



Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{BY}$ Status Flag	0 : Busy (being written or erased) 1 : Ready	RO
FMR01	CPU Rewrite Mode Select Bit (1)	0 : Disables CPU rewrite mode 1 : Enables CPU rewrite mode	RW
FMR02	Lock Bit Disable Select Bit (2)	0 : Enables lock bit 1 : Disables lock bit	RW
FMSTP	Flash Memory Stop Bit (3, 5)	0 : Enables flash memory operation 1 : Stops flash memory operation (placed in low power mode, flash memory initialized)	RW
— (b4)	Reserved Bit	Set to “0”	RW
FMR05	User ROM Area Select Bit (3) (Effective in Only Boot Mode)	0 : Boot ROM area is accessed 1 : User ROM area is accessed	RW
FMR06	Program Status Flag (4)	0 : Terminated normally 1 : Terminated in error	RO
FMR07	Erase Status Flag (4)	0 : Terminated normally 1 : Terminated in error	RO

NOTES :

1. To set this bit to “1,” write “0” and then “1” in succession. Make sure no interrupts or DMA transfers will occur before writing “1” after writing “0”.
   Write to this bit when the $\overline{NMI}$ pin is in the high state. Also, while in EW0 mode, write to this bit from a program in than the flash memory.
   Enter read array mode and set this bit to “0”.
2. To set this bit to “1,” write “0” and then “1” in succession when the FMR01 bit = 1. Make sure no interrupts or no DMA transfers will occur before writing “1” after writing “0”.
   Write to this bit when the $\overline{NMI}$ pin is in the high state.
3. Write to this bit from a program in other than the flash memory.
4. This flag is cleared to “0” by executing the Clear Status command.
5. Effective when the FMR01 bit = 1 (CPU rewrite mode). If the FMR01 bit = 0, although the FMR03 bit can be set to “1” by writing “1” in a program, the flash memory is neither placed in low power mode nor initialized.
6. This status includes writing or reading with the Lock Bit Program or Read Lock Bit Status command.

Figure 19.4 FMR0 Register

Flash Memory Control Register 1

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

Symbol	Address	After Reset
FMR1	01B5h	0X00XX0Xb

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved Bit	The value in this bit when read is indeterminate	RO
FMR11	EW1 Mode Select Bit (1)	0: EW0 mode 1: EW1 mode	RW
— (b3-b2)	Reserved Bit	The value in this bit when read is indeterminate	RO
— (b5-b4)	Reserved Bit	Set to “0”	RW
FMR16	Lock Bit Status Flag	0: Lock 1: Unlock	RO
— (b7)	Reserved Bit	Set to “0”	RW

NOTES :

1. To set this bit to “1,” write “0” and then “1” in succession when the FMR01 bit = 1. Make sure no interrupts or DMA transfers will occur before writing “1” after writing “0”.
   Write to this bit when the $\overline{\text{NMI}}$ pin is in the high state.
   The FMR01 and FMR11 bits both are cleared to “0” by setting the FMR01 bit to “0”.

Figure 19.5 FMR1 Register

19.3.3.1 FMR00 Bit

This bit indicates the flash memory operating state. It is set to “0” while the program, block erase, lock bit program, or read lock bit status command is being executed; otherwise, it is set to “1”.

19.3.3.2 FMR01 Bit

The microcomputer can accept commands when the FMR01 bit is set to “1” (CPU rewrite mode). Set the FMR05 bit to “1” (user ROM area access) as well if in boot mode.

19.3.3.3 FMR02 Bit

The lock bit is disabled by setting the FMR02 bit to “1” (lock bit disabled). (Refer to **19.3.6 Data Protect Function**.) The lock bit is enabled by setting the FMR02 bit to “0” (lock bit enabled).
The FMR02 bit does not change the lock bit status but disables the lock bit function. If the block erase command is executed when the FMR02 bit is set to “1”, the lock bit status changes “0” (locked) to “1” (unlocked) after command execution is completed.

19.3.3.4 FMSTP Bit

The FMSTP bit resets the flash memory control circuits and minimizes power consumption in the flash memory. Access to the flash memory is disabled when the FMSTP bit is set to “1”. Set the FMSTP bit by program in a space other than the flash memory.

- Set the FMSTP bit to “1” if one of the followings occurs: A flash memory access error occurs while erasing or programming in EW0 mode (FMR00 bit does not switch back to “1” (ready)).
- Low-power consumption mode is entered

Use the following procedure to change the FMSTP bit setting.
To stop the flash memory,

(1) Set the FMSTP bit to “1”
(2) Set tps (the wait time to stabilize flash memory circuit)

To restart the flash memory,

(1) Set the FMSTP bit to “0”
(2) Set tps (the wait time to stabilize flash memory circuit)

Figure 19.8 shows a Flow Chart Illustrating How To Start and Stop the Flash Memory Processing Before and After Low Power Dissipation Mode. Follow the procedure on this flow chart.
When entering stop or wait mode, the flash memory is automatically turned off. When exiting stop or wait mode, the flash memory is turned back on. The FMR0 register does not need to be set.

19.3.3.5 FMR05 Bit

This bit selects the boot ROM or user ROM area in boot mode. Set to “0” to access (read) the boot ROM area or to “1” (user ROM access) to access (read, write or erase) the user ROM area.

19.3.3.6 FMR06 Bit

This is a read-only bit indicating an auto program operation state. The FMR06 bit is set to “1” when a program error occurs; otherwise, it is set to “0”. Refer to **19.3.8 Full Status Check**.

19.3.3.7 FMR07 Bit

This is a read-only bit indicating the auto erase operation status. The FMR07 bit is set to “1” when an erase error occurs; otherwise, it is set to “0”. For details, refer to **19.3.8 Full Status Check**.

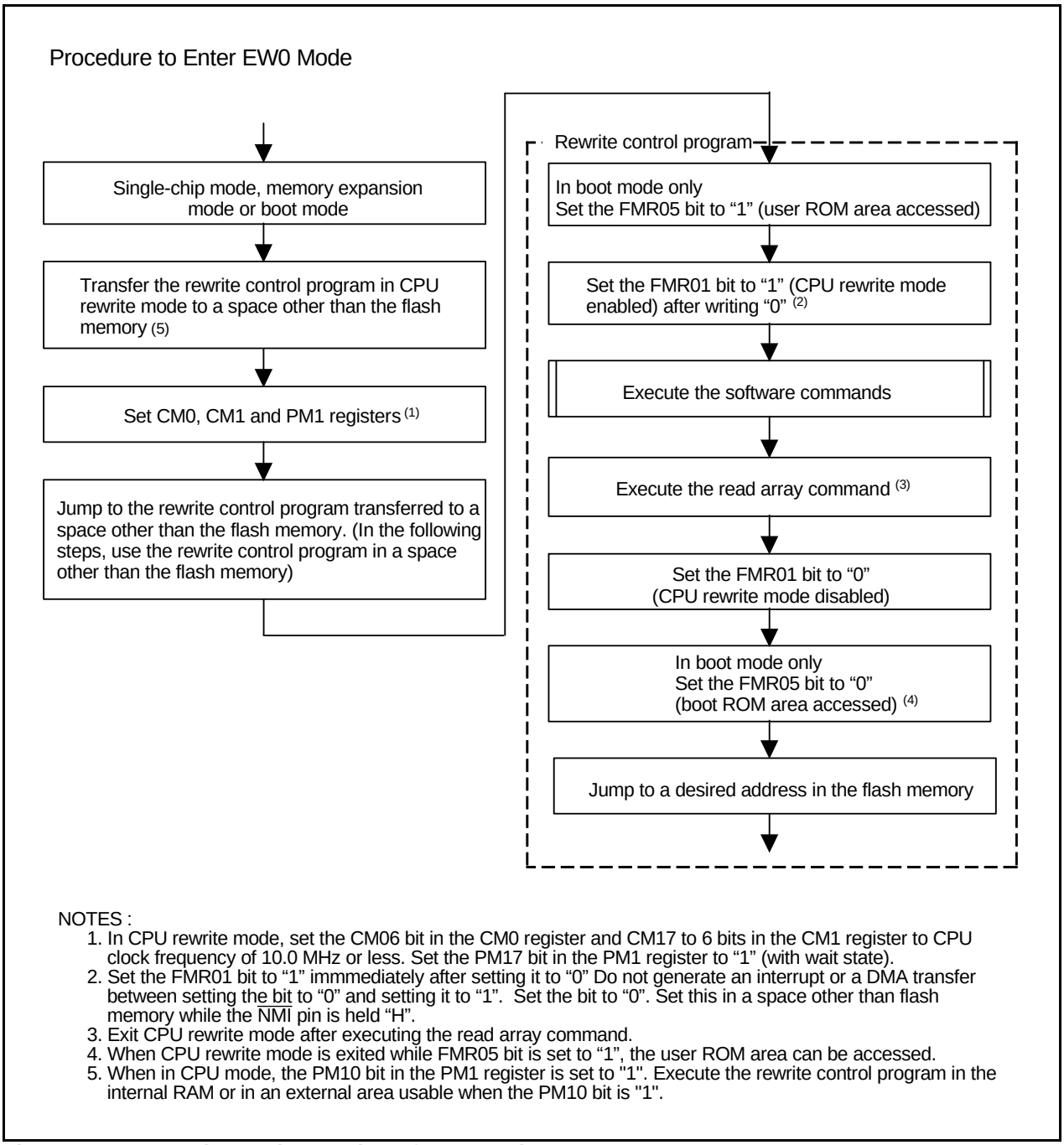
19.3.3.8 FMR11 Bit

EW0 mode is entered by setting the FMR11 bit to “0” (EW0 mode).
EW1 mode is entered by setting the FMR11 bit to “1” (EW1 mode).

19.3.3.9 FMR16 Bit

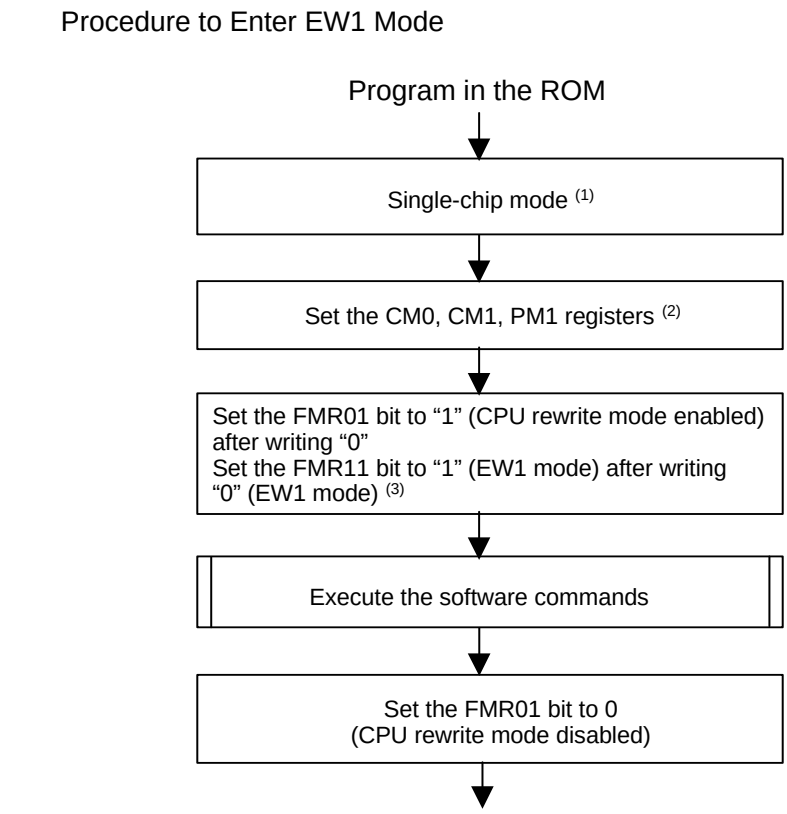
This is a read-only bit indicating the execution result of the read lock bit status command.
When the block, where the read lock bit status command is executed, is locked, the FMR16 bit is set to “0”.
When the block, where the read lock bit status command is executed, is unlocked, the FMR16 bit is set to “1”.

Figure 19.6 shows Setting and Resetting of EW0 Mode. Figure 19.7 show Setting and Resetting of EW1 Mode.



NOTES :
1. In CPU rewrite mode, set the CM06 bit in the CM0 register and CM17 to 6 bits in the CM1 register to CPU clock frequency of 10.0 MHz or less. Set the PM17 bit in the PM1 register to "1" (with wait state).
2. Set the FMR01 bit to "1" immmediately after setting it to "0" Do not generate an interrupt or a DMA transfer between setting the bit to "0" and setting it to "1". Set the bit to "0". Set this in a space other than flash memory while the $\overline{\text{NMI}}$ pin is held "H".
3. Exit CPU rewrite mode after executing the read array command.
4. When CPU rewrite mode is exited while FMR05 bit is set to "1", the user ROM area can be accessed.
5. When in CPU mode, the PM10 bit in the PM1 register is set to "1". Execute the rewrite control program in the internal RAM or in an external area usable when the PM10 bit is "1".

Figure 19.6 Setting and Resetting of EW0 Mode



NOTES:
1. In EW1 mode, do not enter memory expansion or boot mode.
2. In CPU rewrite mode, set the CM06 bit in the CM0 register and the CM17 to 6 bits in the CM1 register to CPU clock frequency of 10.0 MHz or less. Set the PM17 bit in the PM1 register to “1” (with wait state).
3. Set the FMR01 bit to “1” immediately after setting it to “0”. Do not generate an interrupt or a DMA transfer between setting the bit to “0” and setting it to “1”.
   Set the FMR11 bit to “1” immediately after setting it to “0” while the FMR01 bit is set to “1”. Do not generate an interrupt or a DMA transfer between setting the FMR11 bit to 0 and setting it to “1”.
   Set the FMR01 and FMR11 bits while “H” is applied to the $\overline{\text{NMI}}$ pin.

Figure 19.7 Setting and Resetting of EW1 Mode

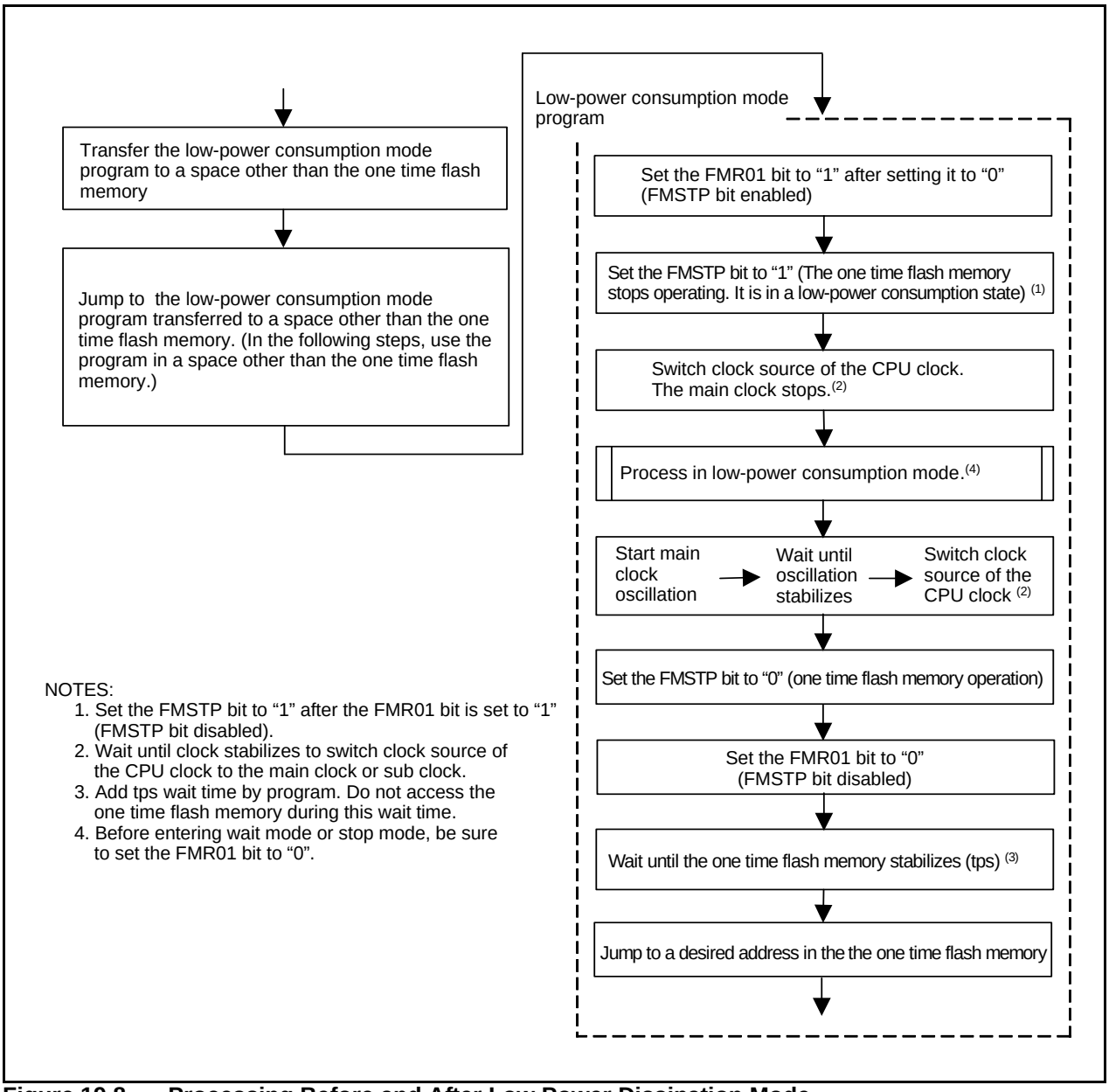


Figure 19.8 Processing Before and After Low Power Dissipation Mode

19.3.4 Precautions on CPU Rewrite Mode

19.3.4.1 Operating Speed

Set the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register to a CPU clock frequency of 10 MHz or less before entering CPU rewrite mode (EW0 or EW1 mode). Also, set the PM17 bit in the PM1 register to “1” (wait state).

19.3.4.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the CPU tries to read data in the flash memory: the UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction.

19.3.4.3 Interrupts (EW0 mode)

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts are available since the FMR0 and FMR1 registers are forcibly reset when either interrupt occurs. Allocate the jump addresses for each interrupt service routines to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ or watchdog timer interrupt occurs. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

19.3.4.4 Interrupts (EW1 mode)

- Do not acknowledge any interrupts with vectors in the relocatable vector table or address match interrupt during the auto program or auto erase period.
- Do not use the watchdog timer interrupt.
- The $\overline{\text{NMI}}$ interrupt is available since the FMR0 and FMR1 registers are forcibly reset when the interrupt occurs. Allocate the jump address for the interrupt service routine to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ interrupt occurs. Execute the rewrite program again after exiting the interrupt service routine.

19.3.4.5 How to Access

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

19.3.4.6 Rewriting in the User ROM Area (EW0 mode)

If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory cannot be rewritten because the rewrite control program is not correctly rewritten. If this error occurs, rewrite the user ROM area while in standard serial I/O mode or parallel I/O mode.

19.3.4.7 Rewriting in the User ROM Area (EW1 mode)

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

19.3.4.8 DMA Transfer

In EW1 mode, do not perform a DMA transfer while the FMR00 bit in the FMR0 register is set to “0” (auto programming or auto erasing).

19.3.4.9 Writing Command and Data

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

19.3.4.10 Wait Mode

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

19.3.4.11 Stop Mode

When entering stop mode, the following settings are required:

- Set the FMR01 bit to “0” (CPU rewrite mode disabled). Disable DMA transfer before setting the CM10 bit to “1” (stop mode).

19.3.4.12 Low-Power Consumption Mode

If the CM05 bit is set to “1” (main clock stopped), do not execute the following commands:

- Program
- Block erase
- Lock bit program
- Read lock bit status

19.3.5 Software Commands

Software commands are described below. The command code and data must be read and written in 16-bit units, to and from even addresses in the user ROM area. When writing command code, the 8 high-order bits (D15 to D8) are ignored.

Table 19.6 Software Commands

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

SRD: Data in the SRD register (D7 to D0)
WA: Address to be written (The address specified in the first bus cycle is the same even address as the address specified in the second bus cycle.)
WD: 16-bit write data
BA: Highest-order block address (must be an even address)
X: Any even address in the user ROM space
xx: 8 high-order bits of command code (ignored)

19.3.5.1 Read Array Command (FFh)

The read array command reads the flash memory.
By writing command code “xxFFh” in the first bus cycle, read array mode is entered. Content of a specified address can be read in 16-bit units after the next bus cycle.
The microcomputer remains in read array mode until another command is written. Therefore, contents from multiple addresses can be read consecutively.

19.3.5.2 Read Status Register Command (70h)

The read status register command reads the status register (refer to **19.3.7 Status Register** for detail).
By writing command code “xx70h” in the first bus cycle, the status register can be read in the second bus cycle. Read an even address in the user ROM area.
Do not execute this command in EW1 mode.

19.3.5.3 Clear Status Register Command (50h)

The clear status register command clears the status register. By writing “xx50h” in the first bus cycle, the FMR07 to FMR06 bits in the FMR0 register are set to “00b” and the SR5 to SR4 bits in the status register are set to “00b”.

19.3.5.4 Program Command (40h)

The program command writes 2-byte data to the flash memory. By writing “xx40h” in the first bus cycle and data to the write address in the second bus cycle, an auto program operation (data program and verify) will start. The address value specified in the first bus cycle must be the same even address as the write address specified in the second bus cycle.
The FMR00 bit in the FMR0 register indicates whether an auto program operation has been completed. The FMR00 bit is set to “0” (busy) during auto program and to “1” (ready) when an auto program operation is completed.
After the completion of an auto program operation, the FMR06 bit in the FMR0 register indicates whether or not the auto program operation has been completed as expected. (Refer to **19.3.8 Full Status Check**.)
An address that is already written cannot be altered or rewritten.
Figure 19.9 shows a Flow Chart of the Program Command Programming.
The lock bit protects each block from being programmed inadvertently. (Refer to **19.3.6 Data Protect Function**.)

In EW1 mode, do not execute this command on the block where the rewrite control program is allocated.
In EW0 mode, the microcomputer enters read status register mode as soon as an auto program operation starts. The status register can be read. The SR7 bit in the status register is set to “0” at the same time an auto program operation starts. It is set to “1” when auto program operation is completed. The microcomputer remains in read status register mode until the read array command is written. After completion of an auto program operation, the status register indicates whether or not the auto program operation has been completed as expected.

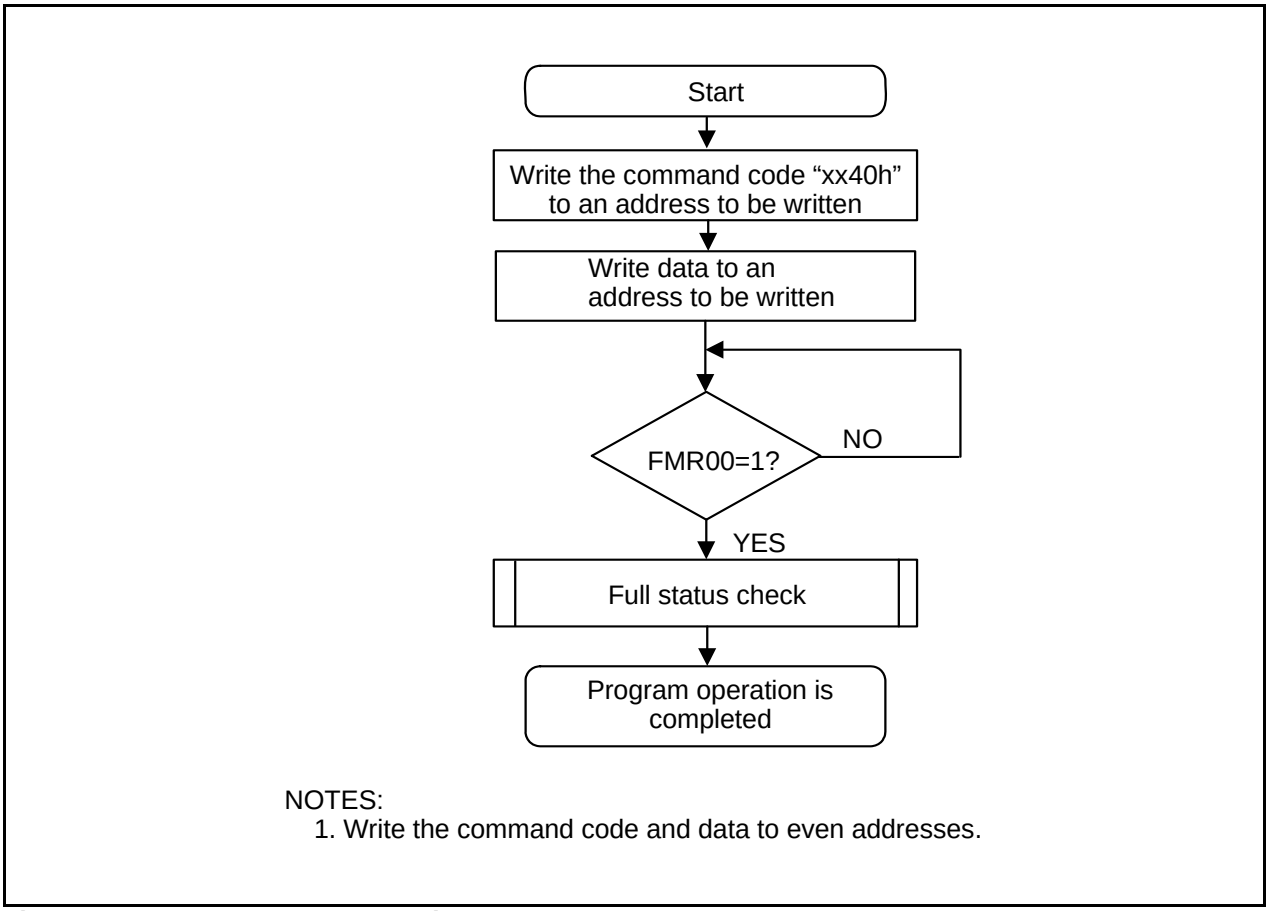
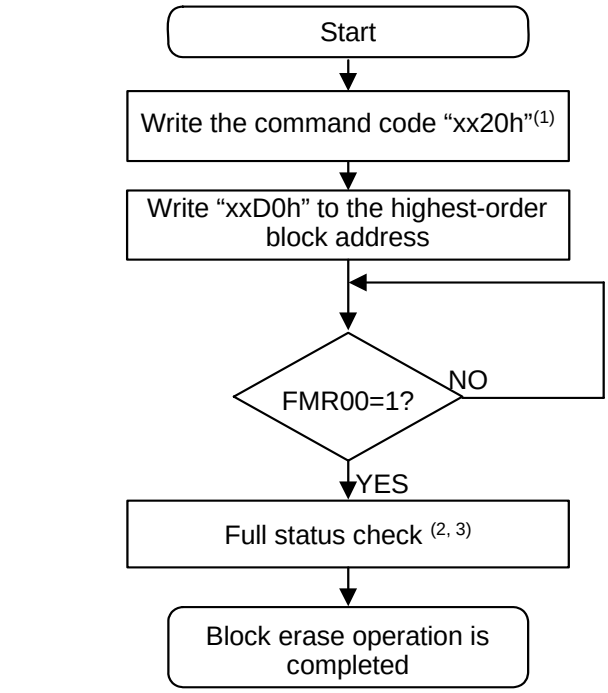


Figure 19.9 Program Command

19.3.5.5 Block Erase Command

The block erase command erases each block.
By writing “xx20h” in the first bus cycle and “xxD0h” to the highest-order even address of a block in the second bus cycle, an auto erase operation (erase and verify) will start in the specified block.
The FMR00 bit in the FMR0 register indicates whether an auto erase operation has been completed.
The FMR00 bit is set to “0” (busy) during auto erase and to “1” (ready) when the auto erase operation is completed.
After the completion of an auto erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto erase operation has been completed as expected. (Refer to **19.3.8 Full Status Check**.)
Figure 19.10 shows a Flow Chart of the Block Erase Command Programming.
The lock bit protects each block from being programmed inadvertently. (Refer to **19.3.6 Data Protect Function**.)
In EW1 mode, do not execute this command on the block where the rewrite control program is allocated. In EW0 mode, the microcomputer enters read status register mode as soon as an auto erase operation starts. The status register can be read. The SR7 bit in the status register is set to “0” at the same time an auto erase operation starts. It is set to “1” when an auto erase operation is completed. The microcomputer remains in read status register mode until the read array command or read lock bit status command is written. Also execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.



NOTES:
1. Write the command code and data to even addresses.
2. Refer to “Figure 19.13 Full Status Check and Handling Procedure for Each Error”.
3. Execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.

Figure 19.10 Block Erase Command

19.3.5.6 Lock Bit Program Command

The lock bit program command sets the lock bit for a specified block to “0” (locked).
By writing “xx77h” in the first bus cycle and “xxD0h” to the highest-order even address of a block in the second bus cycle, the lock bit for the specified block is set to “0”. The address value specified in the first bus cycle must be the same highest-order even address of a block specified in the second bus cycle.
Figure 19.11 shows a Flow Chart of the Lock Bit Program Command Programming. Execute read lock bit status command to read lock bit state (lock bit data).
The FMR00 bit in the FMR0 register indicates whether a lock bit program operation is completed.
Refer to **19.3.6 Data Protect Function** for details on lock bit functions and how to set it to “1” (unlocked).

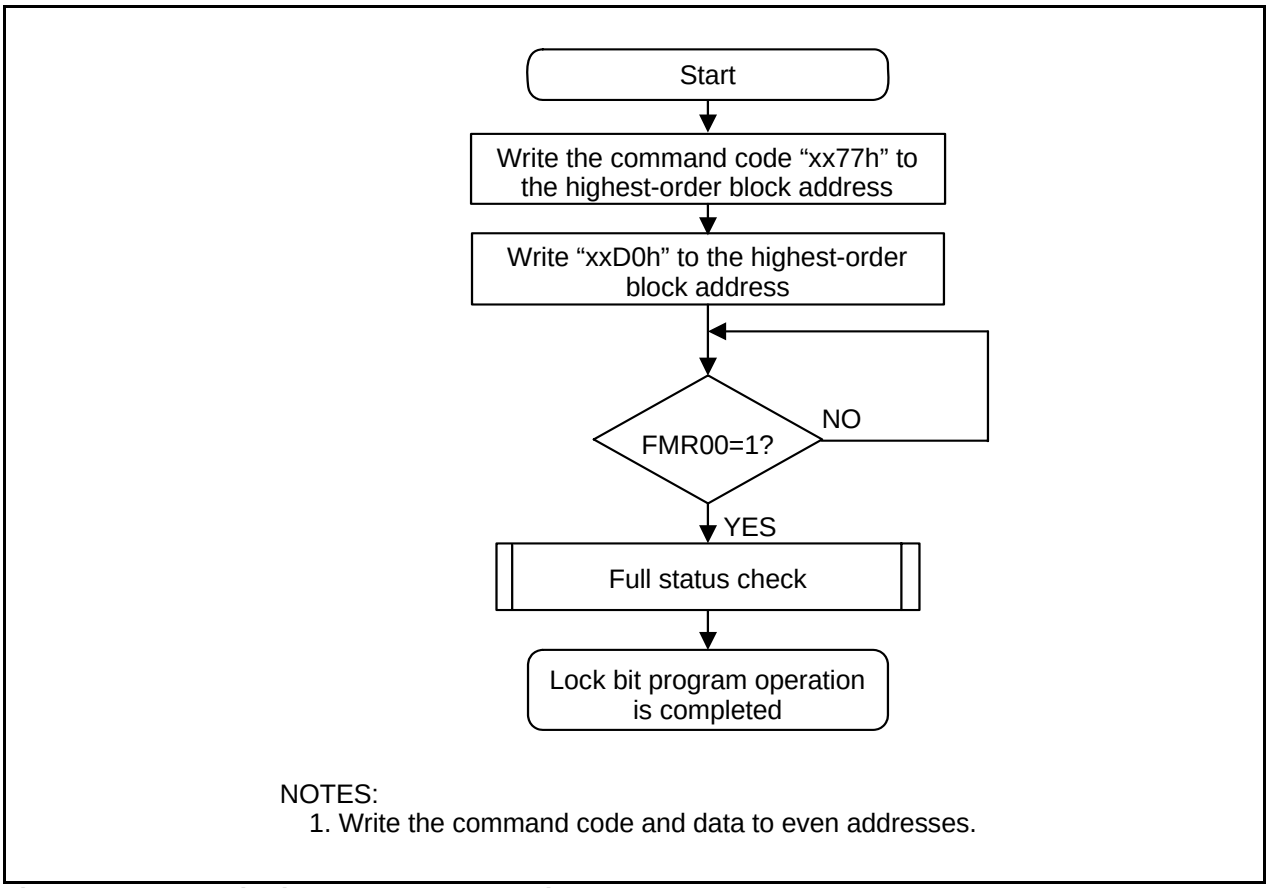


Figure 19.11 Lock Bit Program Command

19.3.5.7 Read Lock Bit Status Command (71h)

The read lock bit status command reads the lock bit state of a specified block.
By writing “xx71h” in the first bus cycle and “xxD0h” to the highest-order even address of a block in the second bus cycle, the FMR16 bit in the FMR1 register stores information on whether or not the lock bit of a specified block is locked. Read the FMR16 bit after the FMR00 bit in the FMR0 register is set to “1” (ready).
Figure 19.12 shows a Flow Chart of the Read Lock Bit Status Command Programming.

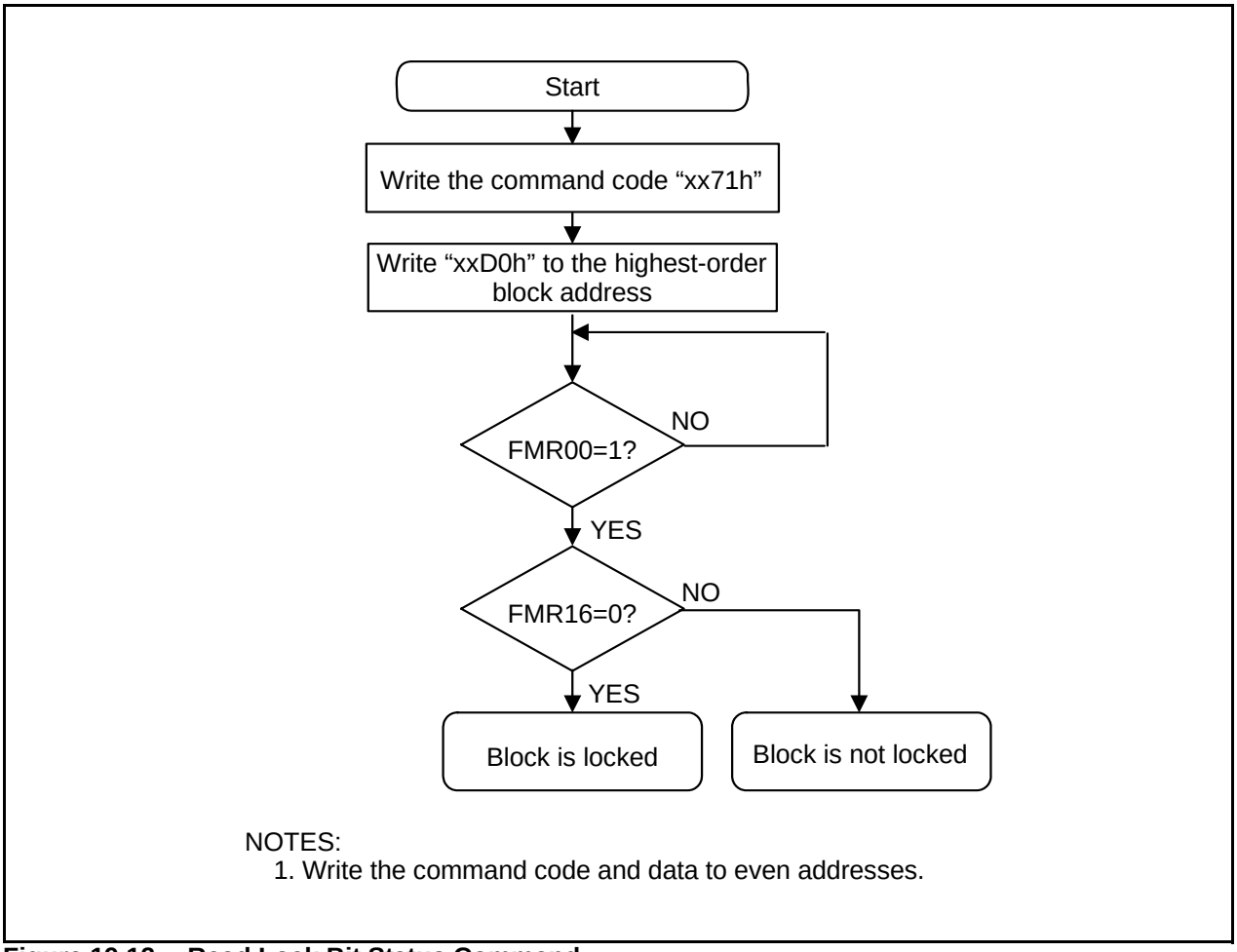


Figure 19.12 Read Lock Bit Status Command

19.3.6 Data Protect Function

Each block in the flash memory has a nonvolatile lock bit. The lock bit is enabled by setting the FMR02 bit to "0" (lock bit enabled). The lock bit allows each block to be individually protected (locked) against program and erase. This helps prevent data from being inadvertently written to or erased from the flash memory.

- When the lock bit status is set to "0", the block is locked (block is protected against program and erase).
- When the lock bit status is set to "1", the block is not locked (block can be programmed or erased).

The lock bit status is set to "0" (locked) by executing the lock bit program command and to "1" (unlocked) by erasing the block. The lock bit status cannot be set to "1" by any commands.
The lock bit status can be read by the read lock bit status command.

The lock bit function is disabled by setting the FMR02 bit to "1". All blocks are unlocked. However, individual lock bit status remains unchanged. The lock bit function is enabled by setting the FMR02 bit to "0". Lock bit status is retained.
If the block erase command is executed while the FMR02 bit is set to "1", the target block is erased regardless of lock bit status. The lock bit status of each block are set to "1" after an erase operation is completed.
Refer to **19.3.5 Software Commands** for details on each command.

19.3.7 Status Register

The status register indicates the flash memory operation state and whether or not an erase or program operation is completed as expected. The FMR00, FMR06 and FMR07 bits in the FMR0 register indicate status register states.
Table 19.7 shows the Status Register.
In EW0 mode, the status register can be read when the followings occur.

- Any even address in the user ROM area is read after writing the read status register command.
- Any even address in the user ROM area is read from when the program, block erase, or lock bit program command is executed until when the read array command is executed.

19.3.7.1 Sequence Status (SR7 and FMR00 Bits)

The sequence status indicates the flash memory operation state. It is set to "0" while the program, block erase, lock bit program, or read lock bit status command is being executed; otherwise, it is set to "1".

19.3.7.2 Erase Status (SR5 and FMR07 Bits)

Refer to **19.3.8 Full Status Check**.

19.3.7.3 Program Status (SR4 and FMR06 Bits)

Refer to **19.3.8 Full Status Check**.

Table 19.7 Status Register

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

- D0 to D7: These data buses are read 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) is set to "1," the program, block erase, and lock bit program commands are not accepted.

19.3.8 Full Status Check

If an error occurs when a program or erase operation is completed, the FMR06 to FMR07 bits in the FMR0 register are set to "1", indicating a specific error. Therefore, execution results can be confirmed by checking these bits (full status check).
Table 19.8 lists Errors and FMR0 Register State. Figure 19.13 shows a flow chart of the Full Status Check and Handling Procedure for Each Error.

Table 19.8 Errors and FMR0 Register State

FMR0 Register (Status Register) State		Error	Error Occurrence Conditions
FMR07 bit (SR5 bit)	FMR06 bit (SR4 bit)		
1	1	Command Sequence error	• Command is written incorrectly • A value other than "xxD0h" or "xxFFh" is written in the second bus cycle of the lock bit program or block erase command (1)
1	0	Erase error	• The block erase command is executed on a locked block • The block erase command is executed on an unlock block and auto erase operation is not completed as expected (2)
0	1	Program error	• The program command is executed on locked blocks • The program command is executed on unlocked blocks but program operation is not completed as expected • The lock bit program command is executed but program operation is not completed as expected (2)

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

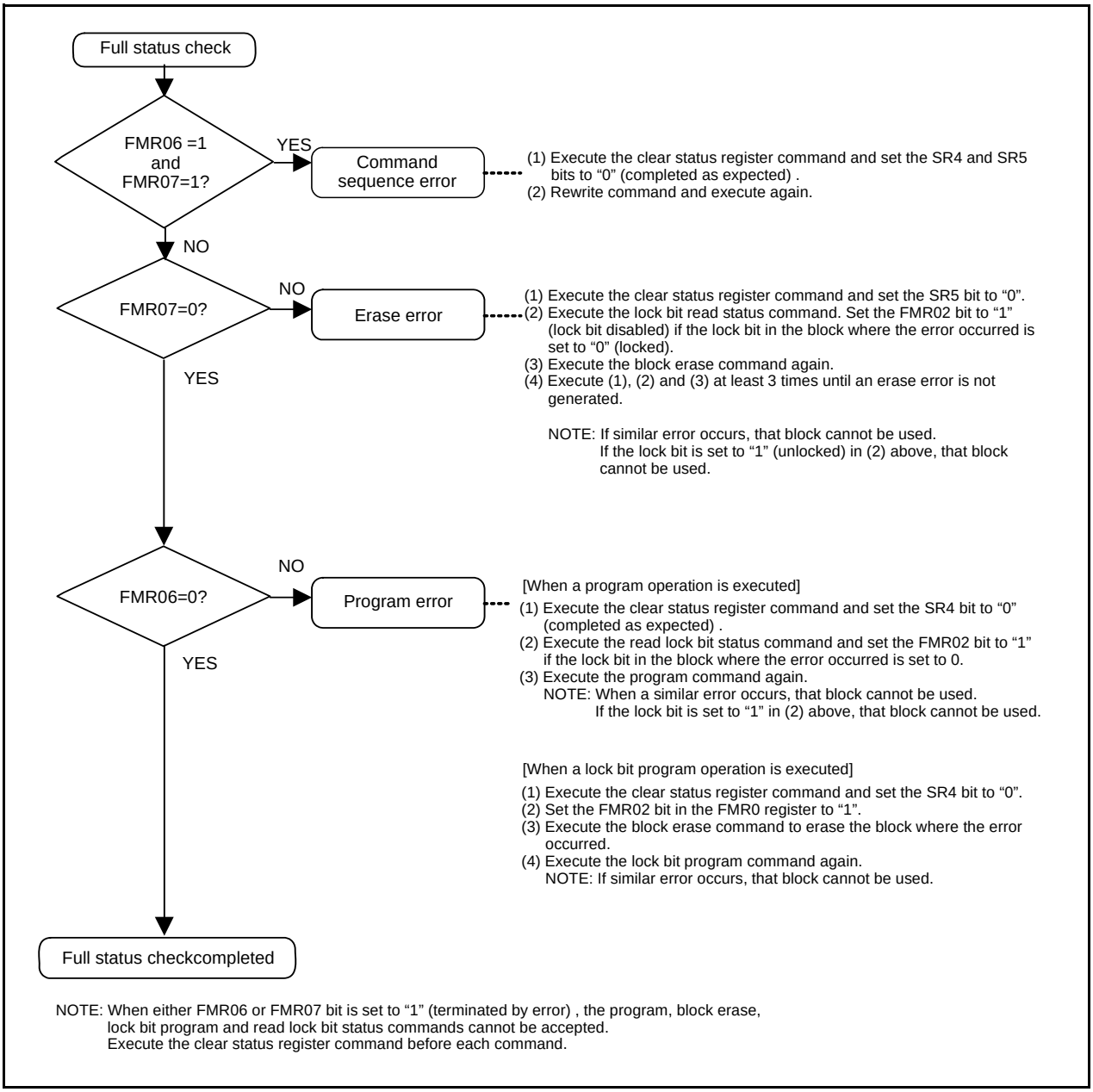


Figure 19.13 Full Status Check and Handling Procedure for Each Error

19.4 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the M16C/30P Group can be used to rewrite the flash memory user ROM area in the microcomputer mounted on a board.
For more information about the serial programmer, contact your serial programmer manufacturer. Refer to the user's manual included with your serial programmer for instructions.
Table 19.9 lists Pin Functions (Flash Memory Standard Serial I/O Mode). Figure 19.14 to Figure 19.15 show Pin Connections in Serial I/O Mode.

19.4.1 ID Code Check Function

The ID code check function determines whether the ID codes sent from the serial programmer matches those written in the flash memory. (Refer to **19.2 Functions To Prevent Flash Memory from Rewriting**.)

Table 19.9 Pin Functions (Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Description
VCC1, VCC2, VSS	Power Input		Apply the Flash Program, Erase Voltage to VCC1 pin and VCC2 pin. Apply 0 V to VSS pin.
CNVSS	CNVSS	I	Connect to VCC1 pin.
$\overline{\text{RESET}}$	Reset Input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is “L” level, input a 20 cycle or longer clock to XIN pin.
XIN	Clock Input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock Output	O	
BYTE	BYTE	I	Connect this pin to VCC1 or VSS.
AVCC, AVSS	Analog Power Supply Input		Connect AVSS to VSS and AVCC to VCC1, respectively.
VREF	Reference Voltage Input	I	Enter the reference voltage for A/D from this pin.
P0_0 to P0_7	Input Port P0	I	Input “H” or “L” level signal or open.
P1_0 to P1_7	Input Port P1	I	Input “H” or “L” level signal or open.
P2_0 to P2_7	Input Port P2	I	Input “H” or “L” level signal or open.
P3_0 to P3_7	Input Port P3	I	Input “H” or “L” level signal or open.
P4_0 to P4_7	Input Port P4	I	Input “H” or “L” level signal or open.
P5_1 to P5_4, P5_6, P5_7	Input Port P5	I	Input “H” or “L” level signal or open.
P5_0	$\overline{\text{CE}}$ Input	I	Input “H” level signal.
P5_5	$\overline{\text{EPM}}$ Input	I	Input “L” level signal.
P6_0 to P6_3	Input Port P6	I	Input “H” or “L” level signal or open.
P6_4/RTS1	BUSY Output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitors the boot program operation check signal output pin.
P6_5/CLK1	SCLK Input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input “L”.
P6_6/RXD1	RXD Input	I	Serial data input pin.
P6_7/TXD1	TXD Input	O	Serial data output pin. (1)
P7_0 to P7_7	Input Port P7	I	Input “H” or “L” level signal or open.
P8_0 to P8_3, P8_6, P8_7	Input Port P8	I	Input “H” or “L” level signal or open.
P8_4	P8_4 input	I	Input “L” level signal. (2)
P8_5/$\overline{\text{NMI}}$	$\overline{\text{NMI}}$ Input	I	Connect this pin to VCC1.
P9_0 to P9_7	Input Port P9	I	Input “H” or “L” level signal or open.
P10_0 to P10_7	Input Port P10	I	Input “H” or “L” level signal or open.

NOTES:

1. When using the standard serial I/O mode, the internal pull-up is enabled for the TXD1 (P6_7) pin while the $\overline{\text{RESET}}$ pin is “L”.
2. When using the standard serial I/O mode, the P0_0 to P0_7, P1_0 to P1_7 pins may become indeterminate while the P8_4 pin is “H” and the $\overline{\text{RESET}}$ pin is “L”. If this causes a program, apply “L” to the P8_4 pin.

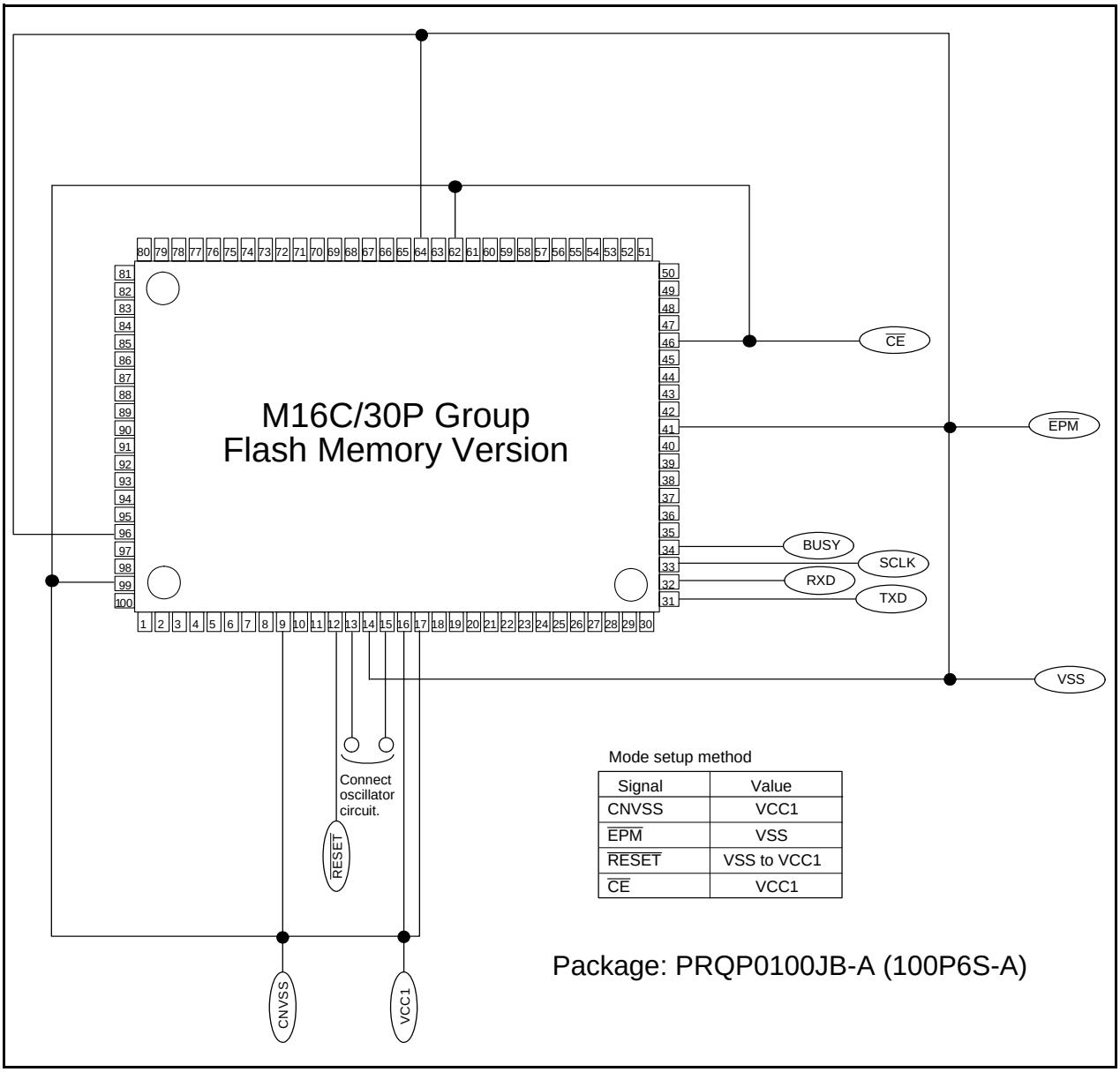


Figure 19.14 Pin Connections for Serial I/O Mode (1)

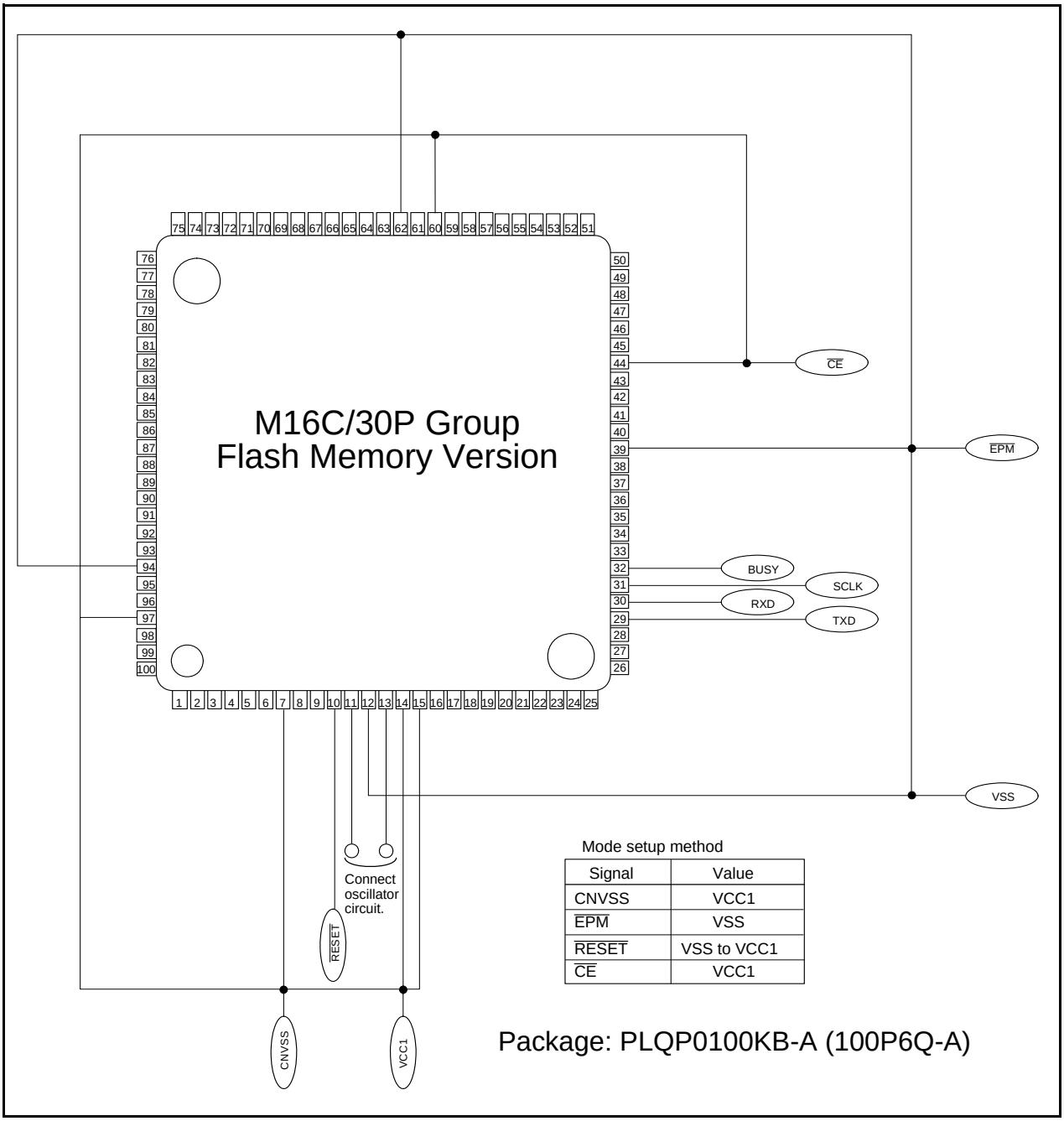
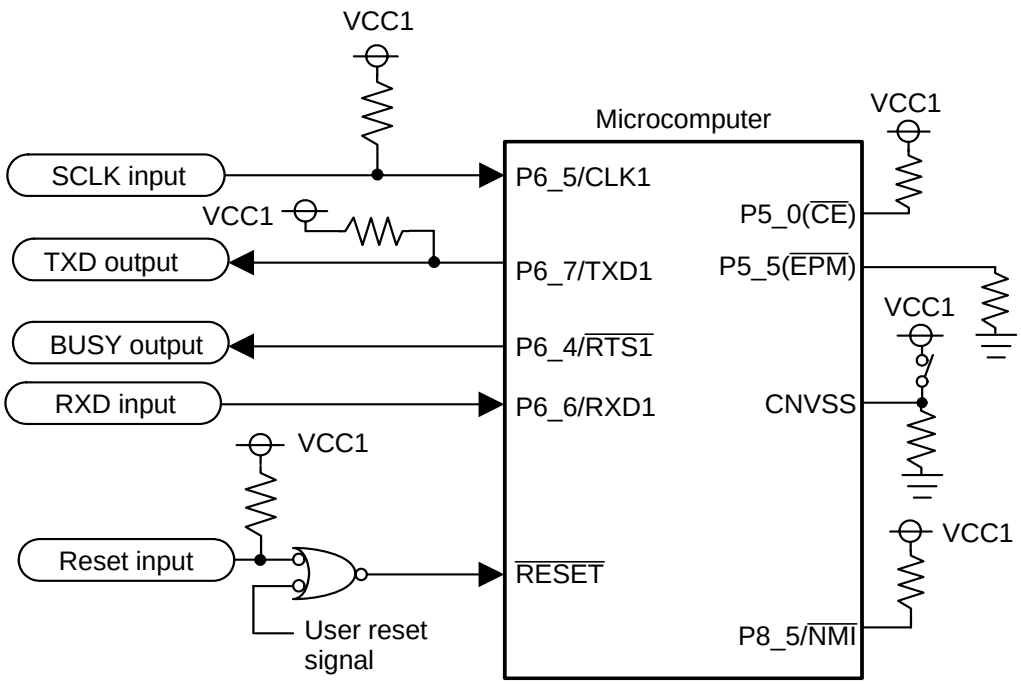


Figure 19.15 Pin Connections for Serial I/O Mode (2)

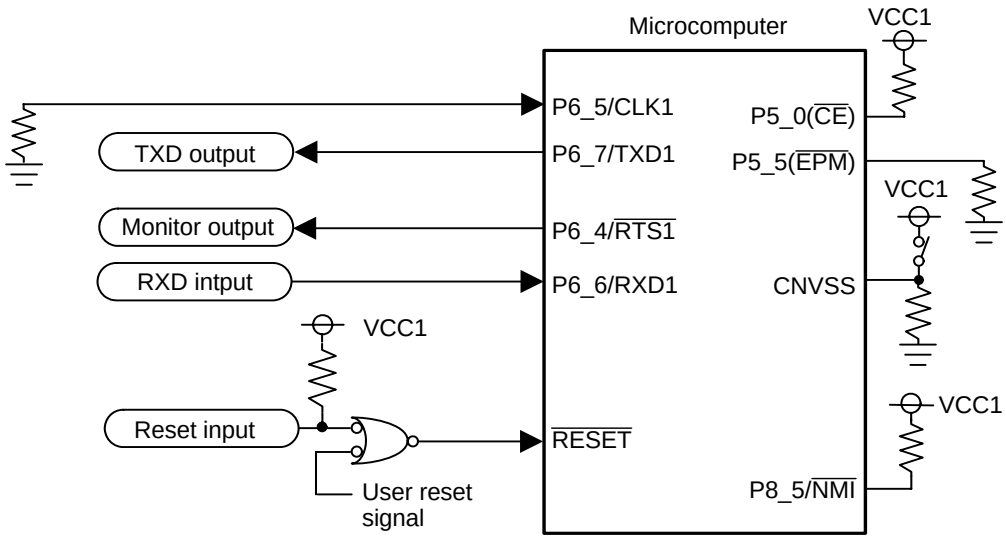
19.4.2 Example of Circuit Application in the Standard Serial I/O Mode

Figure 19.16 and Figure 19.17 show example of Circuit Application in Standard Serial I/O Mode 1 and Mode 2, respectively. Refer to the user's manual of your serial programmer to handle pins controlled by the serial programmer.



NOTES:
1. Control pins and external circuitry will vary depending on a programmer. For more information, see the programmer manual.
2. In this example, modes are switched between single-chip mode and standard serial input/output mode by controlling the CNVSS input with a switch.
3. If in standard serial input/output mode 1 there is a possibility that the user reset signal will go low during serial input/output mode, break the connection between the user reset signal and $\overline{\text{RESET}}$ pin by using, for example, a jumper switch.

Figure 19.16 Circuit Application in Standard Serial I/O Mode 1



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

Figure 19.17 Circuit Application in Standard Serial I/O Mode 2

19.5 Parallel I/O Mode

In parallel I/O mode, the user ROM area and the boot ROM area can be rewritten by a parallel programmer supporting the M16C/30P Group. Contact your parallel programmer manufacturer for more information on the parallel programmer. Refer to the user's manual included with your parallel programmer for instructions.

19.5.1 User ROM and Boot ROM Areas

An erase block operation in the boot ROM area is applied to only one 4 Kbyte block. The rewrite control program in standard serial I/O mode is written in the boot ROM area before shipment. Do not rewrite the boot ROM area if using the serial programmer.
In parallel I/O mode, the boot ROM area is located in addresses 0FF000h to 0FFFFFh. Rewrite this address range only if rewriting the boot ROM area. (Do not access addresses other than addresses 0FF000h to 0FFFFFh.)

19.5.2 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read and rewritten in parallel I/O mode. (Refer to **19.2 Functions To Prevent Flash Memory from Rewriting**.)

20. One Time Flash Version

The one time flash version microcomputer has the same functions as the masked ROM version except the built-in flash memory. The flash memory will be referred to as the one time flash memory in the one time flash version chapter.
The one time flash memory can be written in standard serial I/O mode. It cannot be erased. Table 20.1 lists One Time Flash Memory Version Specifications. See **Table 1.1 Performance Outline of M16C/30P Group** for the items not listed in Table 20.1.

Table 20.1 One Time Flash Memory Version Specifications

Item	Specification
Program Method	In units of word
Program Endurance	1 time
Data Retention	10 years
ROM Code Protection	Parallel I/O modes and standard serial I/O modes are supported

Table 20.2 One Time Flash Memory Rewrite Modes Overview

Flash Memory Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	The user ROM area is written using a dedicated serial programmer. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART	Boot ROM area and user ROM area are rewritten using a dedicated parallel programmer.
ROM Programmer	Serial programmer	Parallel programmer

20.1 Low Consumption Mode

The one time flash memory version enters the low power consumption mode to reduce the power consumption by stopping the one time flash memory. Figure 20.1 shows the FMR0 Register in One Time Flash Memory and Figure 20.2 shows the processing before and after low power consumption mode.
To enter stop mode or wait mode, set the FMR01 bit in the FMR0 register to 0 (FMSTP bit disabled).
For models including the PM13 bit, when the FMR01 bit in the FMR0 register is 1 (FMSTP bit enabled), the PM13 bit in the PM1 register automatically becomes 1. Store the program to change the FMSTP bit either in external area that is usable when the PM13 bit is set to 1. When the FMR01 bit is changed to 0 (FMSTP bit disabled), the PM13 bit is set back to the value before the change. However, when the PM13 bit value is changed while the FMR01 bit is 1, the changed value is reflected after the FMR01 bit is set to 0. External area does not change depending on FMR01 bit status for models without the PM13 bit. Refer to **Figure 6.2 PM1 Register (1)** and **Figure 6.3 PM1 Register (2)** for availability of the PM13 bit in the PM1 Register.

Flash Memory Control Register 0

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

Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
— (b1)	Reserved Bit	The value in this bit when read is indeterminate	RO
FMR01	FMSTP Enable Bit (1)	0 : FMSTP bit disabled 1 : FMSTP bit enabled	RW
— (b2)	Reserved Bit	Set to “0”	RW
FMSTP	One time Flash Memory Stop Bit (2, 3)	0 : Enables one time flash memory operation 1 : Stops one time flash memory operation (placed in low power mode, flash memory initialized)	RW
— (b5-b4)	Reserved Bit	Set to “0”	RW
— (b7-b6)	Reserved Bit	Set to “0”	RO

NOTES :
1. To set this bit to “1,” write “0” and then “1” in succession. Make sure no interrupts or DMA transfers will occur before writing “1” after writing “0”.
   Write to this bit when the $\overline{\text{NMI}}$ pin is in the high state.
2. Set this bit in a program area other than the one time flash memory area.
3. Effective when the FMR01 bit = 1 (FMSTP bit enabled). If the FMR01 bit = 0, although the FMSTP bit can be set to “1” by writing “1” by program, the one time flash memory is neither placed in low power mode nor initialized.

Figure 20.1 FMR0 Register in One Time Flash Version

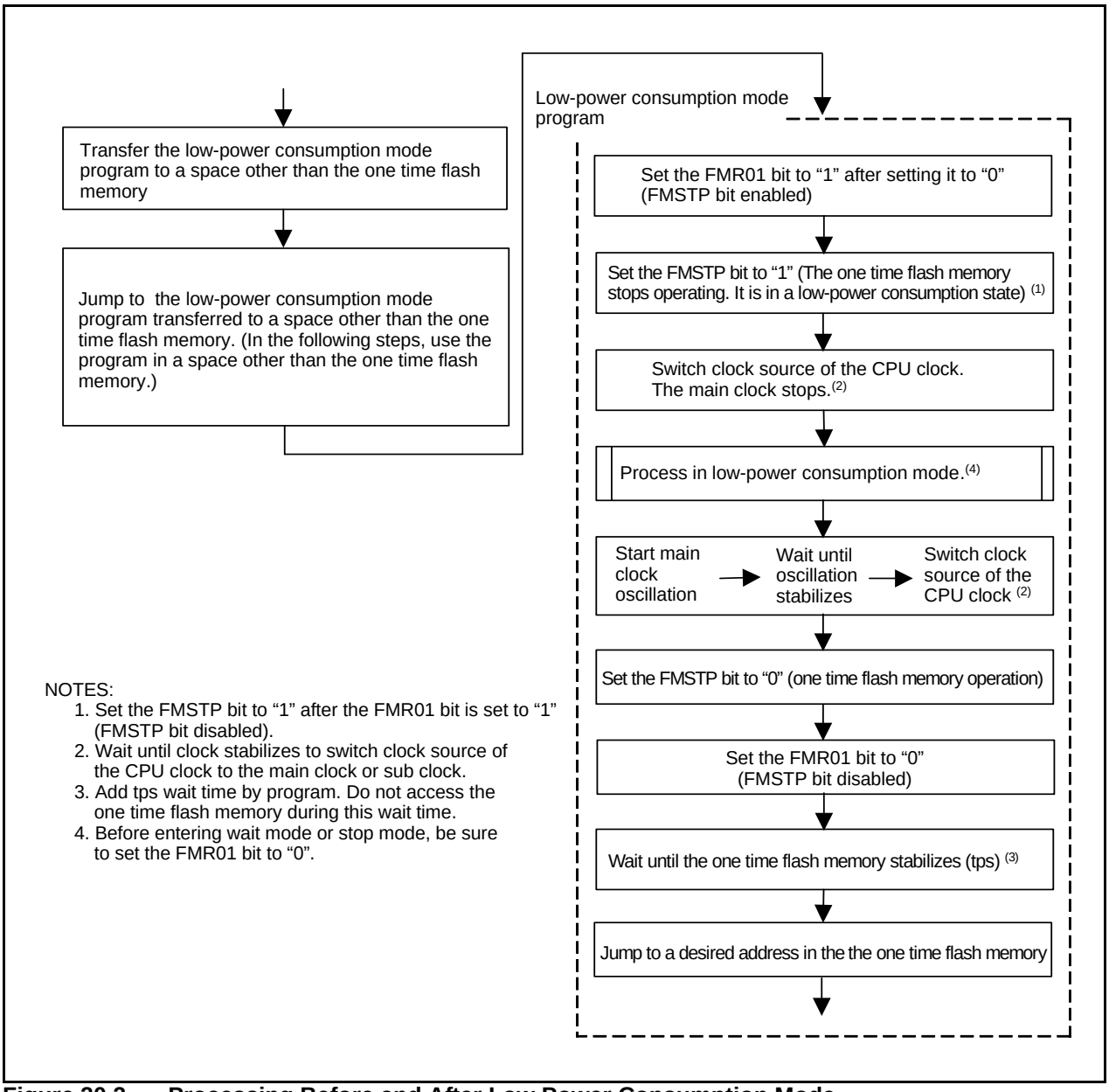


Figure 20.2 Processing Before and After Low Power Consumption Mode

20.2 Functions to Prevent One Time Flash Version from Being Read

Parallel I/O mode of one time flash has a ROM code protect function, and Standard I/O mode of one time flash has an ID code check function.

20.2.1 ROM Code Protect Function

The ROM code protect function prevents the one time flash being read in parallel I/O mode. The ROM code protect function is enabled when the address 0FFFFFh is set to "3Fh". The ROM code protect function is disabled when the address 0FFFFFh is set to "00h" or "FFh". Write a program with "3Fh", "00h" or "FFh" set in the address to one time flash version.
Table 20.3 lists the values and functions of the address 0FFFFFh.

Table 20.3 The Values and Functions of Address 0FFFFFh

Set Value	Function
3Fh	ROM code protect enabled
00h	ROM code protect disabled
FFh	
Other than the above	Do not set.

20.2.2 ID Code Check Function

The ID code check function is used in standard serial I/O mode. The ID code sent from the serial programmer and the ID code written in the one time flash memory are checked to see if they match. If these ID codes do not match, the commands sent from the serial programmer are not acknowledged. However, if the four bytes of the reset vector are FFFFFFFFh, the ID code is not checked and all commands can be acknowledged.
The ID codes is 7-byte data stored consecutively, beginning with the first byte, into addresses 0FFFDFh, 0FFFE3h, 0FFFEBh, 0FFFEFh, 0FFFF3h, 0FFFF7h, and 0FFFFBh. Write a programs with the ID codes set at these addresses to the one time flash memory.

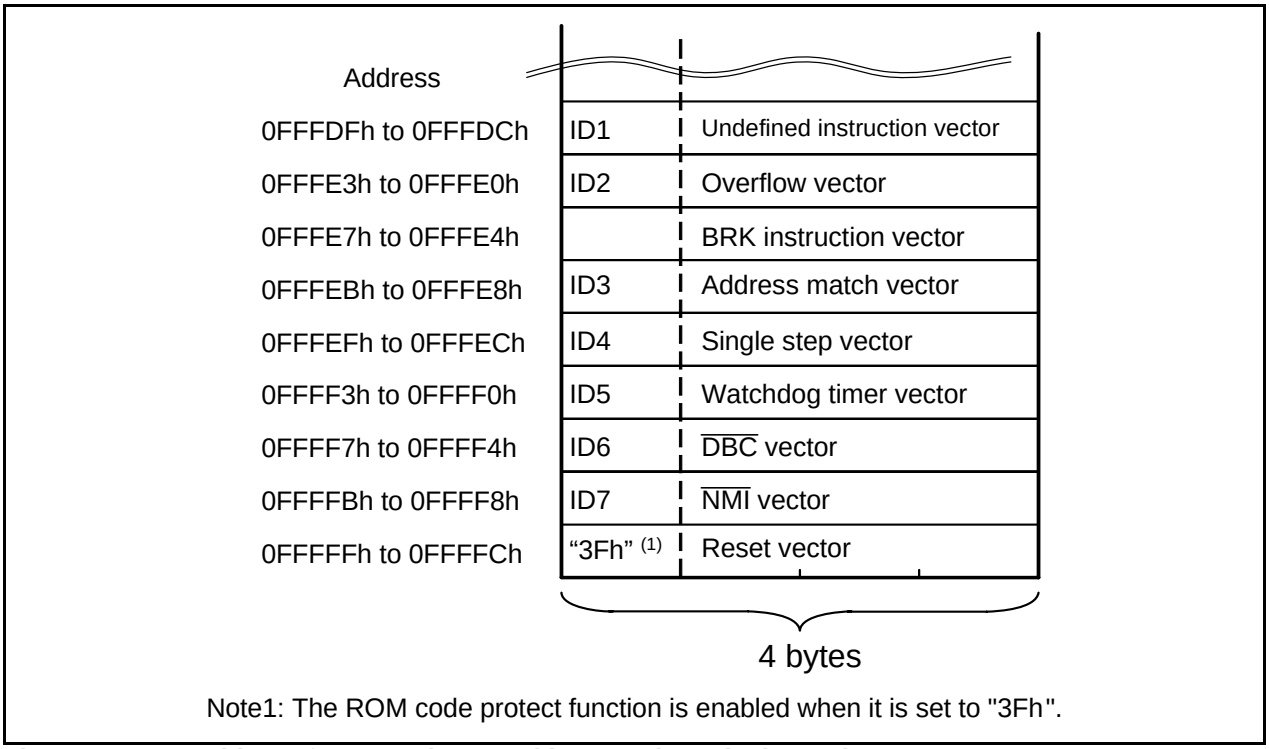


Figure 20.3 Address for ID Code Stored in One Time Flash Version

20.3 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the M16C/30P Group can be used to write (the one time flash memory) user ROM area in the microcomputer mounted on a board.
For more information about the serial programmer, contact your serial programmer manufacturer. Refer to the user's manual included with your serial programmer for instructions.
Table 20.4 lists Pin Functions (One Time Flash Memory Standard Serial I/O Mode). Figure 20.4 show Pin Connections for Serial I/O Mode.

20.3.1 ID Code Check Function

The ID code check function determines whether the ID codes sent from the serial programmer matches those written in the one time flash memory. (Refer to **20.2 Functions to Prevent One Time Flash Version from Being Read**.)

Table 20.4 Pin Functions (One Time Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Description
VCC1, VCC2, VSS	Power Input		Apply the Flash Program Voltage to VCC1 pin and VCC2 pin. Apply 0 V to VSS pin.
CNVSS	CNVSS	I	Connect to VCC1 pin.
$\overline{\text{RESET}}$	Reset Input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is “L” level, input a 20 cycle or longer clock to XIN pin.
XIN	Clock Input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock Output	O	
BYTE	BYTE	I	Connect this pin to VCC1 or VSS.
AVCC, AVSS	Analog Power Supply Input		Connect AVSS to VSS and AVCC to VCC1, respectively.
VREF	Reference Voltage Input	I	Enter the reference voltage for A/D from this pin.
P0_0 to P0_7	Input Port P0	I	Input “H” or “L” level signal or open.
P1_0 to P1_7	Input Port P1	I	Input “H” or “L” level signal or open.
P2_0 to P2_7	Input Port P2	I	Input “H” or “L” level signal or open.
P3_0 to P3_7	Input Port P3	I	Input “H” or “L” level signal or open.
P4_0 to P4_7	Input Port P4	I	Input “H” or “L” level signal or open.
P5_1 to P5_4, P5_6, P5_7	Input Port P5	I	Input “H” or “L” level signal or open.
P5_0	$\overline{\text{CE}}$ Input	I	Input “H” level signal.
P5_5	$\overline{\text{EPM}}$ Input	I	Input “L” level signal.
P6_0 to P6_3	Input Port P6	I	Input “H” or “L” level signal or open.
P6_4/RTS1	BUSY Output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitors the boot program operation check signal output pin.
P6_5/CLK1	SCLK Input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input “L”.
P6_6/RXD1	RXD Input	I	Serial data input pin.
P6_7/TXD1	TXD Input	O	Serial data output pin. (1)
P7_0 to P7_7	Input Port P7	I	Input “H” or “L” level signal or open.
P8_0 to P8_3, P8_6, P8_7	Input Port P8	I	Input “H” or “L” level signal or open.
P8_4	P8_4 input	I	Input “L” level signal. (2)
P8_5/$\overline{\text{NMI}}$	$\overline{\text{NMI}}$ Input	I	Connect this pin to VCC1.
P9_0 to P9_7	Input Port P9	I	Input “H” or “L” level signal or open.
P10_0 to P10_7	Input Port P10	I	Input “H” or “L” level signal or open.

NOTES:

1. When using the standard serial I/O mode, the internal pull-up is enabled for the TXD1 (P6_7) pin while the $\overline{\text{RESET}}$ pin is “L”.
2. When using the standard serial I/O mode, the P0_0 to P0_7, P1_0 to P1_7 pins may become indeterminate while the P8_4 pin is “H” and the $\overline{\text{RESET}}$ pin is “L”. If this causes a program, apply “L” to the P8_4 pin.

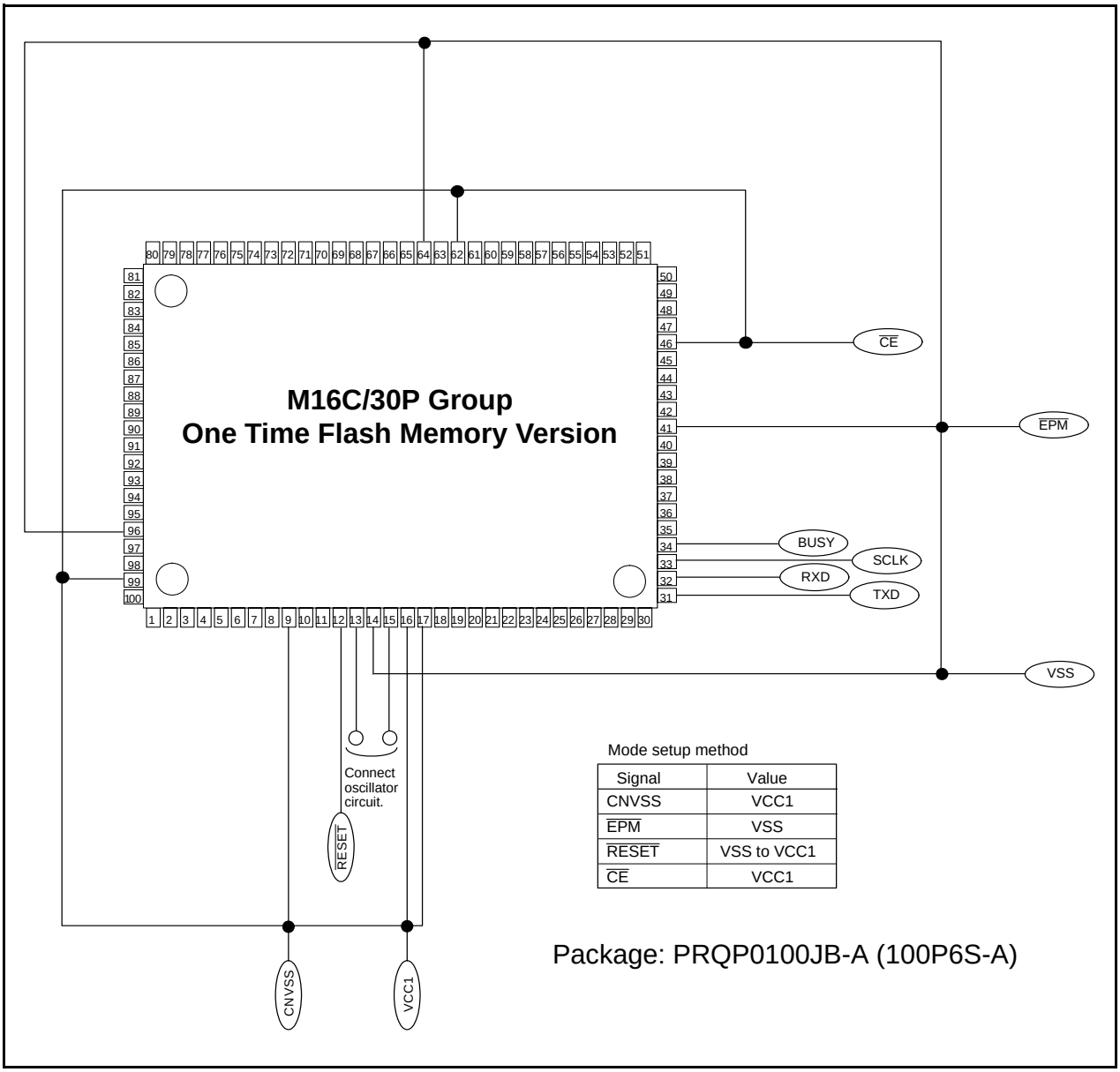
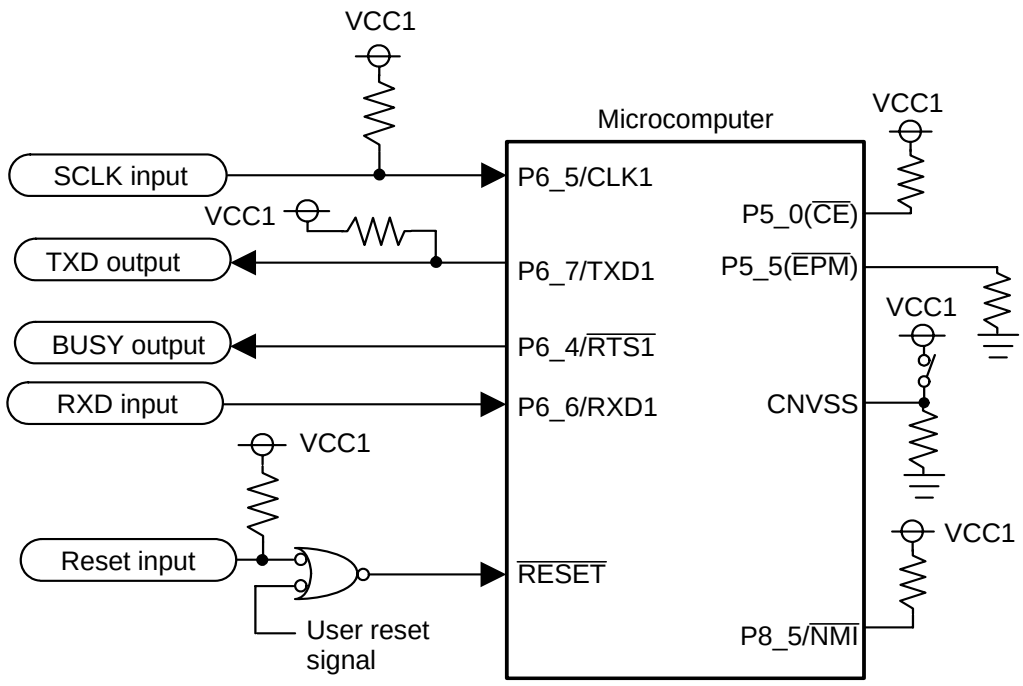


Figure 20.4 Pin Connections for Serial I/O Mode

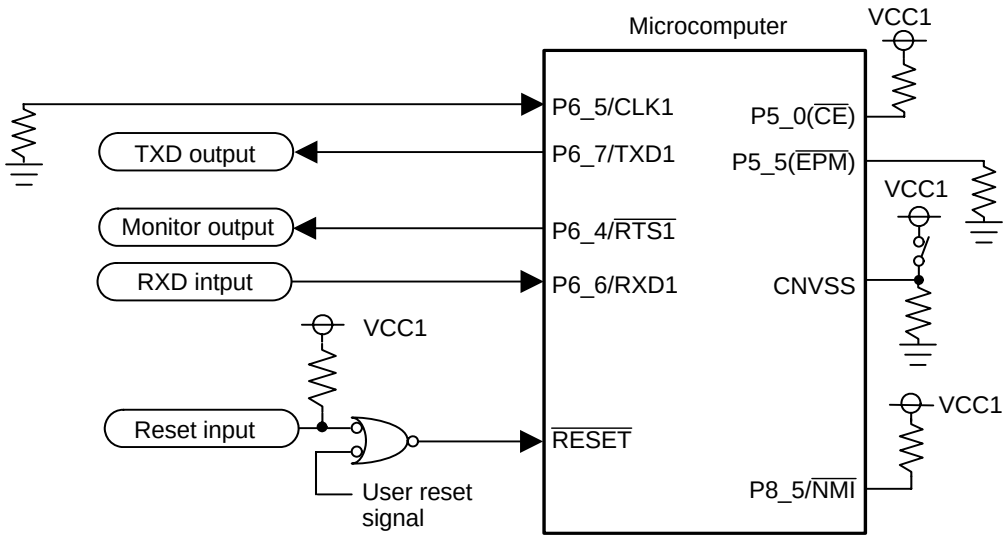
20.3.2 Example of Circuit Application in the Standard Serial I/O Mode

Figure 20.5 and Figure 20.6 show example of Circuit Application in Standard Serial I/O Mode 1 and Mode 2, respectively. Refer to the user's manual of your serial programmer to handle pins controlled by the serial programmer.



NOTES:
1. Control pins and external circuitry will vary depending on a programmer. For more information, see the programmer manual.
2. In this example, modes are switched between single-chip mode and standard serial input/output mode by controlling the CNVSS input with a switch.
3. If in standard serial input/output mode 1 there is a possibility that the user reset signal will go low during serial input/output mode, break the connection between the user reset signal and RESET pin by using, for example, a jumper switch.

Figure 20.5 Circuit Application in Standard Serial I/O Mode 1



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

Figure 20.6 Circuit Application in Standard Serial I/O Mode 2

21. Electrical Characteristics

Table 21.1 Absolute Maximum Ratings

Symbol	Parameter		Condition	Rated Value	Unit
V_{CC}	Supply Voltage(V_{CC1}=V_{CC2})		V_{CC1}=V_{CC2}=AVcc	–0.3 to 6.5	V
AVcc	Analog Supply Voltage		V_{CC1}=V_{CC2}=AVcc	–0.3 to 6.5	V
V_I	Input Voltage	$\overline{RESET}$, CNVSS, BYTE, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, VREF, XIN		–0.3 to Vcc+0.3	V
		P7_0, P7_1		–0.3 to 6.5	V
V_O	Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, XOUT		–0.3 to Vcc+0.3	V
		P7_0, P7_1		–0.3 to 6.5	V
P_d	Power Dissipation		–40°C<T_{opr}≤85°C	300	mW
T_{opr}	Operating Ambient Temperature	When the Microcomputer is Operating		–20 to 85 / –40 to 85	°C
		One Time Flash Program Erase		0 to 60	
		Flash Program Erase		0 to 60	
T_{stg}	Storage Temperature			–65 to 150	°C

Table 21.2 Recommended Operating Conditions (1)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply Voltage (V_{CC1}=V_{CC2})		2.7	5.0	5.5	V
AVcc	Analog Supply Voltage			Vcc		V
Vss	Supply Voltage			0		V
AVss	Analog Supply Voltage			0		V
V_{IH}	HIGH Input Voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7	0.8Vcc		Vcc	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (during single-chip mode)	0.8Vcc		Vcc	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (data input during memory expansion and microprocessor mode)	0.5Vcc		Vcc	V
		P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	0.8Vcc		Vcc	V
		P7_0, P7_1	0.8Vcc		6.5	V
V_{IL}	LOW Input Voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7	0		0.2Vcc	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (during single-chip mode)	0		0.2Vcc	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (data input during memory expansion and microprocessor mode)	0		0.16Vcc	V
		P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	0		0.2Vcc	V
$I_{OH(peak)}$	HIGH Peak Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			−10.0	mA
$I_{OH(avg)}$	HIGH Average Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			−5.0	mA
$I_{OL(peak)}$	LOW Peak Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			10.0	mA
$I_{OL(avg)}$	LOW Average Output Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7			5.0	mA
f(XIN)	Main Clock Input Oscillation Frequency (4)	Vcc=3.0V to 5.5V	0		16	MHz
		Vcc=2.7V to 3.0V	0		20×V_{CC1}−44	MHz
f(XCIN)	Sub-Clock Oscillation Frequency			32.768	50	kHz
f(BCLK)	CPU Operation Clock		0		16	MHz

NOTES:
1. Referenced to V_{CC1} = V_{CC2} = 2.7 to 5.5V at T_{opr} = −20 to 85°C / −40 to 85°C unless otherwise specified.
2. The Average Output Current is the mean value within 100ms.
3. The total $I_{OL(peak)}$ for ports P0, P1, P2, P8_6, P8_7, P9 and P10 must be 80mA max. The total $I_{OL(peak)}$ for ports P3, P4, P5, P6, P7 and P8_0 to P8_4 must be 80mA max. The total $I_{OH(peak)}$ for ports P0, P1, and P2 must be −40mA max. The total $I_{OH(peak)}$ for ports P3, P4 and P5 must be −40mA max. The total $I_{OH(peak)}$ for ports P6, P7, and P8_0 to P8_4 must be −40mA max.
   The total $I_{OH(peak)}$ for ports P8_6, P8_7 and P9 must be −40mA max. Set Average Output Current to 1/2 of peak.
4. Relationship between main clock oscillation frequency, and supply voltage.

Main clock input oscillation frequency

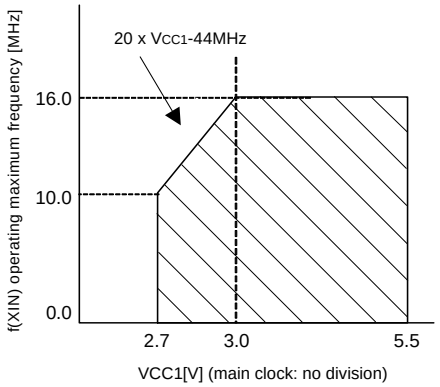


Table 21.3 A/D Conversion Characteristics (1)

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
–	Resolution		$V_{REF}=V_{CC}$				10	Bits
INL	Integral Non-Linearity Error	10bit	$V_{REF}=V_{CC}=5V$	AN0 to AN7 input, AN0_0 to AN0_7 input, ANEX0, ANEX1 input			±5	LSB
			$V_{REF}=V_{CC}=3.3V$	AN0 to AN7 input, AN0_0 to AN0_7 input, ANEX0, ANEX1 input			±7	LSB
		8bit	$V_{REF}=V_{CC}=5V, 3.3V$				±2	LSB
–	Absolute Accuracy	10bit	$V_{REF}=V_{CC}=5V$	AN0 to AN7 input, AN0_0 to AN0_7 input, ANEX0, ANEX1 input			±5	LSB
			$V_{REF}=V_{CC}=3.3V$	AN0 to AN7 input, AN0_0 to AN0_7 input, ANEX0, ANEX1 input			±7	LSB
		8bit	$V_{REF}=V_{CC}=5V, 3.3V$				±2	LSB
–	Tolerance Level Impedance					3		kΩ
DNL	Differential Non-Linearity Error						±2	LSB
–	Offset Error						±5	LSB
–	Gain Error						±5	LSB
R_{LADDER}	Ladder Resistance		$V_{REF}=V_{CC}$		10		40	kΩ
t_{CONV}	10-bit Conversion Time, Sample & Hold Function Available		$V_{REF}=V_{CC}=5V$, ϕAD=10MHz		3.3			µs
t_{CONV}	8-bit Conversion Time, Sample & Hold Function Available		$V_{REF}=V_{CC}=5V$, ϕAD=10MHz		2.8			µs
t_{SAMP}	Sampling Time				0.3			µs
V_{REF}	Reference Voltage				3.0		V_{CC}	V
V_{IA}	Analog Input Voltage				0		V_{REF}	V

NOTES:
1. Referenced to $V_{CC}=AV_{CC}=V_{REF}=3.3$ to 5.5V, $V_{SS}=AV_{SS}=0V$ at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified.
2. ϕAD frequency must be 10 MHz or less.
3. When sample & hold function is disabled, ϕAD frequency must be 250 kHz or more, in addition to the limitation in Note 2.
4. When sample & hold function is enabled, ϕAD frequency must be 1MHz or more, in addition to the limitation in Note 2.

Table 21.4 Flash Memory Version Electrical Characteristics (1)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
–	Program and Erase Endurance (2)		100(3)			cycle
–	Word Program Time (V_{CC1}=5.0V)			25	200	µs
–	Lock Bit Program Time			25	200	µs
–	Block Erase Time (V_{CC1}=5.0V)	4-Kbyte block		0.3	4	s
–		8-Kbyte block		0.3	4	s
–		32-Kbyte block		0.5	4	s
–		64-Kbyte block		0.8	4	s
t_{PS}	Flash Memory Circuit Stabilization Wait Time				15	µs
–	Data Hold Time (4)		10			year

NOTES:
1. Referenced to V_{CC1}=4.5 to 5.5V, 3.0 to 3.6V at Topr = 0 to 60 °C (U3, U5) unless otherwise specified.
2. Program and Erase Endurance refers to the number of times a block erase can be performed.
   If the program and erase endurance is 100, each block can be erased 100 times.
   For example, if a 4 Kbytes block A is erased after writing 1 word data 2,048 times, each to a different address, this counts as one program and erase endurance. Data cannot be written to the same address more than once without erasing the block. (Rewrite prohibited)
3. Maximum number of E/W cycles for which operation is guaranteed.
4. Topr = -40 to 85 °C (U3) / -20 to 85 °C (U5).

Table 21.5 Flash Memory Version Program / Erase Voltage and Read Operation Voltage Characteristics

Flash Program, Erase Voltage	Flash Read Operation Voltage
V_{CC1} = 3.3 ± 0.3 V or 5.0 ± 0.5 (T_{opr} = 0°C to 60°C)	V_{CC1}=2.7 to 5.5 V (T_{opr} = -40°C to 85°C (U3) -20°C to 85°C (U5))

Table 21.6 One Time Flash Version Electrical Characteristics (1)

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
–	Program Endurance			1	cycle
–	Word Program Time (V_{CC1}=5.0V)		50	500	µs
tPS	One Time Flash Memory Circuit Stabilization Wait Time			15	µs
–	Data Hold Time (4)	10			year

NOTES:
1. Referenced to V_{CC1}=4.5 to 5.5V, 3.0 to 3.6V at Topr = 0 to 60 °C (U3, U5) unless otherwise specified.
2. Topr = -40 to 85 °C (U3) / -20 to 85 °C (U5).

Table 21.7 One Time Flash Version Program Voltage and Read Operation Voltage Characteristics

Flash Program Voltage	Flash Read Operation Voltage
V_{CC1} = 3.3 ± 0.3 V or 5.0 ± 0.5 (T_{opr} = 0°C to 60°C)	V_{CC1}=2.7 to 5.5 V (T_{opr} = -40°C to 85°C (U3) -20°C to 85°C (U5))

Table 21.8 Power Supply Circuit Timing Characteristics

Symbol	Parameter	Measuring Condition	Standard			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Time for Internal Power Supply Stabilization During Powering-On	V_{CC}=2.7V to 5.5V			2	ms
$t_{d(R-S)}$	STOP Release Time				1500	µs
$t_{d(W-S)}$	Low Power Dissipation Mode Wait Mode Release Time				1500	µs

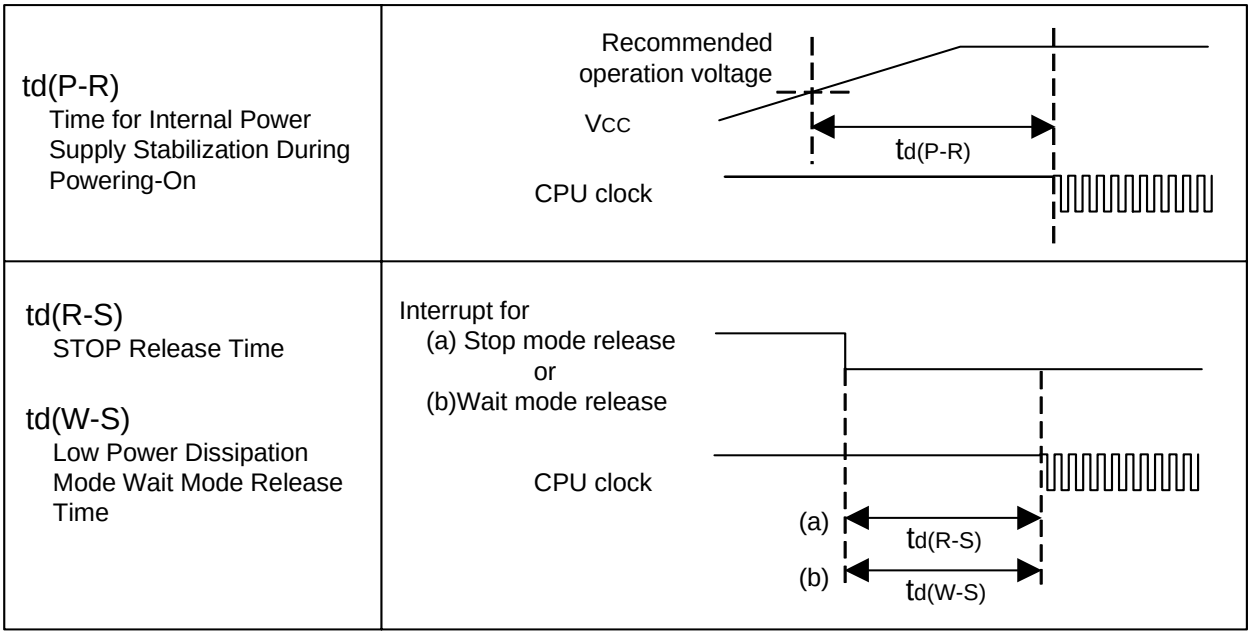


Figure 21.1 Power Supply Circuit Timing Diagram

$V_{CC1}=V_{CC2}=5V$

Table 21.9 Electrical Characteristics(1) (1)

Symbol	Parameter			Measuring Condition	Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7		I_{OH}=–5mA	V_{CC}–2.0		V_{CC}	V
V_{OH}	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7		I_{OH}=–200µA	V_{CC}–0.3		V_{CC}	V
V_{OH}	HIGH Output Voltage XOUT		HIGHPOWER	I_{OH}=–1mA	V_{CC}–2.0		V_{CC}	V
			LOWPOWER	I_{OH}=–0.5mA	V_{CC}–2.0		V_{CC}	
	HIGH Output Voltage XCOUT		HIGHPOWER	With no load applied		2.5		V
			LOWPOWER	With no load applied		1.6		
V_{OL}	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7		I_{OL}=5mA			2.0	V
V_{OL}	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7		I_{OL}=200µA			0.45	V
V_{OL}	LOW Output Voltage XOUT		HIGHPOWER	I_{OL}=1mA			2.0	V
			LOWPOWER	I_{OL}=0.5mA			2.0	
	LOW Output Voltage XCOUT		HIGHPOWER	With no load applied		0		V
			LOWPOWER	With no load applied		0		
V_{T+}-V_{T-}	Hysteresis	TA0IN to TA2IN, TB0IN to TB2IN, $\overline{INT0}$ to $\overline{INT4}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$ to $\overline{CTS2}$, CLK0 to CLK2, TA0OUT to TA2OUT, $\overline{KI0}$ to $\overline{KI3}$, RXD0 to RXD2, SCL0 to SCL2, SDA0 to SDA2			0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$			0.2		2.5	V
I_{IH}	HIGH Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, $\overline{RESET}$, CNVSS, BYTE		V_I=5V			5.0	µA
I_{IL}	LOW Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, $\overline{RESET}$, CNVSS, BYTE		V_I=0V			–5.0	µA
R_{PULLUP}	Pull-Up Resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7		V_I=0V	30	50	170	kΩ
R_{fXIN}	Feedback Resistance XIN					1.5		MΩ
R_{fXCIN}	Feedback Resistance XCIN					15		MΩ
V_{RAM}	RAM Retention Voltage			At stop mode	2.0			V

NOTES:
1. Referenced to $V_{CC1}=V_{CC2}$=4.2 to 5.5V, V_{SS} = 0V at T_{opr} = –20 to 85°C / –40 to 85°C, f(XIN) =16MHz unless otherwise specified.

Table 21.10 Electrical Characteristics (2) (1)

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power Supply Current (V_{CC1}=V_{CC2}=4.0V to 5.5V)	In single-chip mode, the output pins are open and other pins are V_{SS}	Mask ROM	f(XIN)=16MHz No division		10	15	mA
			One Time Flash	f(XIN)=16MHz, No division		10	18	mA
			Flash Memory	f(XIN)=16MHz, No division		12	18	mA
			One Time Flash	f(XIN)=10MHz, VCC1=5.0V		15		mA
			Flash Memory Program	f(XIN)=10MHz, VCC1=5.0V		15		mA
			Flash Memory Erase	f(XIN)=10MHz, VCC1=5.0V		25		mA
			Mask ROM	f(XCIN)=32kHz Low power dissipation mode, ROM (3)		25		µA
			One Time Flash	f(XCIN)=32kHz Low power dissipation mode, RAM (3)		25		µA
				f(XCIN)=32kHz Low power dissipation mode, Flash Memory (3)		350		µA
			Flash Memory	f(XCIN)=32kHz Low power dissipation mode, RAM (3)		25		µA
				f(XCIN)=32kHz Low power dissipation mode, Flash Memory (3)		420		µA
			Mask ROM One Time Flash Flash Memory	f(XCIN)=32kHz Wait mode (2), Oscillation capability High		7.5		µA
				f(XCIN)=32kHz Wait mode (2), Oscillation capability Low		2.0		µA
				Stop mode Topr =25°C		0.8	3.0	µA

NOTES:
1. Referenced to V_{CC1}=V_{CC2}=4.2 to 5.5V, V_{SS} = 0V at T_{opr} = –20 to 85°C / –40 to 85°C, f(XIN)=16MHz unless otherwise specified.
2. With one timer operated using fC32.
3. This indicates the memory in which the program to be executed exists.

$V_{CC1}=V_{CC2}=5V$

Timing Requirements

(V_{CC1} = V_{CC2} = 5V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.11 External Clock Input (XIN input) (1)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External Clock Input Cycle Time	62.5		ns
$t_{w(H)}$	External Clock Input HIGH Pulse Width	25		ns
$t_{w(L)}$	External Clock Input LOW Pulse Width	25		ns
t_r	External Clock Rise Time		15	ns
t_f	External Clock Fall Time		15	ns

NOTES:
1. The condition is V_{CC1}=V_{CC2}=3.0 to 5.0V.

Table 21.12 Memory Expansion Mode and Microprocessor Mode

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data Input Access Time (for setting with no wait)		(NOTE 1)	ns
$t_{ac2(RD-DB)}$	Data Input Access Time (for setting with wait)		(NOTE 2)	ns
$t_{su(DB-RD)}$	Data Input Setup Time	40		ns
$t_{su(RDY-BCLK)}$	$\overline{RDY}$ Input Setup Time	30		ns
$t_{su(HOLD-BCLK)}$	$\overline{HOLD}$ Input Setup Time	40		ns
$t_{h(RD-DB)}$	Data Input Hold Time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{RDY}$ Input Hold Time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{HOLD}$ Input Hold Time	0		ns

NOTES:
1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 45[ns]$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 45[ns]$$ n is "2" for 1-wait setting.

$V_{CC1}=V_{CC2}=5V$

Timing Requirements

(V_{CC1} = V_{CC2} = 5V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.13 Timer A Input (Counter Input in Event Counter Mode)

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

Table 21.14 Timer A Input (Gating Input in Timer Mode)

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

Table 21.15 Timer A Input (External Trigger Input in One-shot Timer Mode)

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

Table 21.16 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN Input HIGH Pulse Width	100		ns
$t_{w(TAL)}$	TAiIN Input LOW Pulse Width	100		ns

Table 21.17 Timer A Input (Counter Increment/Decrement Input in Event Counter Mode)

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

Table 21.18 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

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

$V_{CC1}=V_{CC2}=5V$

Timing Requirements

(V_{CC1} = V_{CC2} = 5V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.19 Timer B Input (Counter Input in Event Counter Mode)

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

Table 21.20 Timer B Input (Pulse Period Measurement Mode)

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

Table 21.21 Timer B Input (Pulse Width Measurement Mode)

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

Table 21.22 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{\text{ADTRG}}$ Input Cycle Time	1000		ns
$t_{w(ADL)}$	$\overline{\text{ADTRG}}$ input LOW Pulse Width	125		ns

Table 21.23 Serial Interface

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

Table 21.24 External Interrupt $\overline{\text{INTi}}$ Input

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

$V_{CC1}=V_{CC2}=5V$

Switching Characteristics

(V_{CC1} = V_{CC2} = 5V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.25 Memory Expansion and Microprocessor Modes (for setting with no wait)

Symbol	Parameter		Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address Output Delay Time	See **Figure 21.2**		25	ns
th(BCLK-AD)	Address Output Hold Time (in relation to BCLK)		–3		ns
th(RD-AD)	Address Output Hold Time (in relation to RD)		0		ns
th(WR-AD)	Address Output Hold Time (in relation to WR)		(NOTE 2)		ns
td(BCLK-CS)	Chip Select Output Delay Time			25	ns
th(BCLK-CS)	Chip Select Output Hold Time (in relation to BCLK)		–3		ns
td(BCLK-ALE)	ALE Signal Output Delay Time			15	ns
th(BCLK-ALE)	ALE Signal Output Hold Time		–4		ns
td(BCLK-RD)	RD Signal Output Delay Time			25	ns
th(BCLK-RD)	RD Signal Output Hold Time		0		ns
td(BCLK-WR)	WR Signal Output Delay Time			25	ns
th(BCLK-WR)	WR Signal Output Hold Time		0		ns
td(BCLK-DB)	Data Output Delay Time (in relation to BCLK)			40	ns
th(BCLK-DB)	Data Output Hold Time (in relation to BCLK) (3)		4		ns
td(DB-WR)	Data Output Delay Time (in relation to WR)		(NOTE 1)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR) (3)		(NOTE 2)		ns
td(BCLK-HLDA)	HLDA Output Delay Time			40	ns

NOTES:

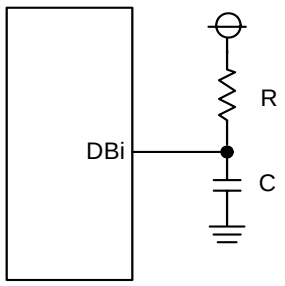
1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5\times10^{9}}{f(BCLK)} - 40[ns]$$ f(BCLK) is 12.5MHz or less.

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5\times10^{9}}{f(BCLK)} - 10[ns]$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.
   Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
   Hold time of data bus is expressed in
   $t = -CR \times \ln(1 - V_{OL} / V_{CC1})$
   by a circuit of the right figure.
   For example, when V_{OL} = 0.2V_{CC1}, C = 30pF, R = 1kΩ, hold time of output "L" level is
   $t = -30pF \times 1k\Omega \times \ln(1 - 0.2V_{CC1} / V_{CC1})$
   = 6.7ns.



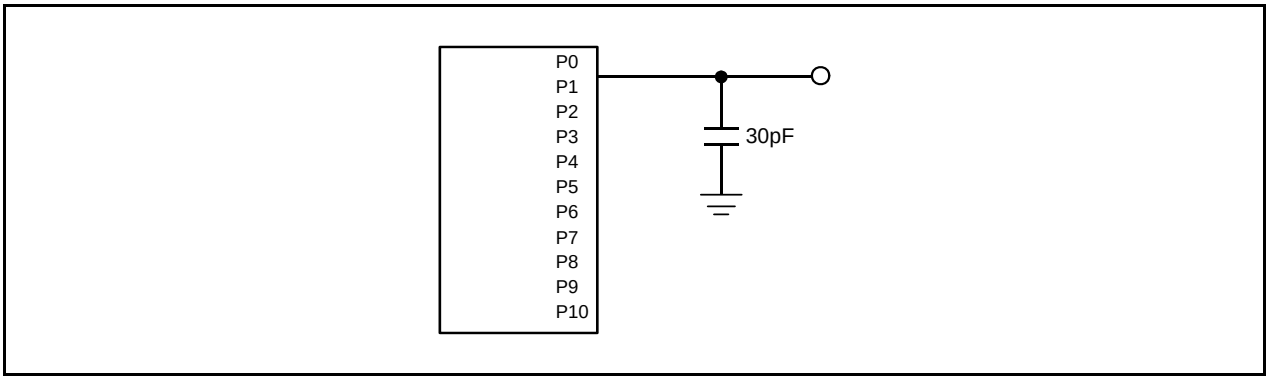


Figure 21.2 Ports P0 to P10 Measurement Circuit

$V_{CC1}=V_{CC2}=5V$

Switching Characteristics

(V_{CC1} = V_{CC2} = 5V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.26 Memory Expansion and Microprocessor Modes (for 1 wait setting and external area access)

Symbol	Parameter		Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address Output Delay Time	See **Figure 21.2**		25	ns
$t_{h(BCLK-AD)}$	Address Output Hold Time (in relation to BCLK)		–3		ns
$t_{h(RD-AD)}$	Address Output Hold Time (in relation to RD)		0		ns
$t_{h(WR-AD)}$	Address Output Hold Time (in relation to WR)		(NOTE 2)		ns
$t_{d(BCLK-CS)}$	Chip Select Output Delay Time			25	ns
$t_{h(BCLK-CS)}$	Chip Select Output Hold Time (in relation to BCLK)		–3		ns
$t_{d(BCLK-ALE)}$	ALE Signal Output Delay Time			15	ns
$t_{h(BCLK-ALE)}$	ALE Signal Output Hold Time		–4		ns
$t_{d(BCLK-RD)}$	RD Signal Output Delay Time			25	ns
$t_{h(BCLK-RD)}$	RD Signal Output Hold Time		0		ns
$t_{d(BCLK-WR)}$	WR Signal Output Delay Time			25	ns
$t_{h(BCLK-WR)}$	WR Signal Output Hold Time		0		ns
$t_{d(BCLK-DB)}$	Data Output Delay Time (in relation to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data Output Hold Time (in relation to BCLK) (3)		4		ns
$t_{d(DB-WR)}$	Data Output Delay Time (in relation to WR)		(NOTE 1)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR)(3)		(NOTE 2)		ns
$t_{d(BCLK-HLDA)}$	HLDA Output Delay Time			40	ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

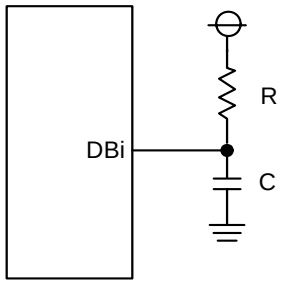
$$\frac{(n-0.5)\times 10^9}{f(BCLK)} - 40[ns]$$

n is “1” for 1-wait setting, f(BCLK) is 12.5MHz or less.

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5\times 10^9}{f(BCLK)} - 10[ns]$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in
t = –CR X ln (1–V_{OL} / V_{CC1})
by a circuit of the right figure.
For example, when V_{OL} = 0.2V_{CC1}, C = 30pF, R = 1kΩ, hold time of output ”L” level is
t = –30pF X 1kΩ X ln(1–0.2V_{CC1} / V_{CC1})
= 6.7ns.



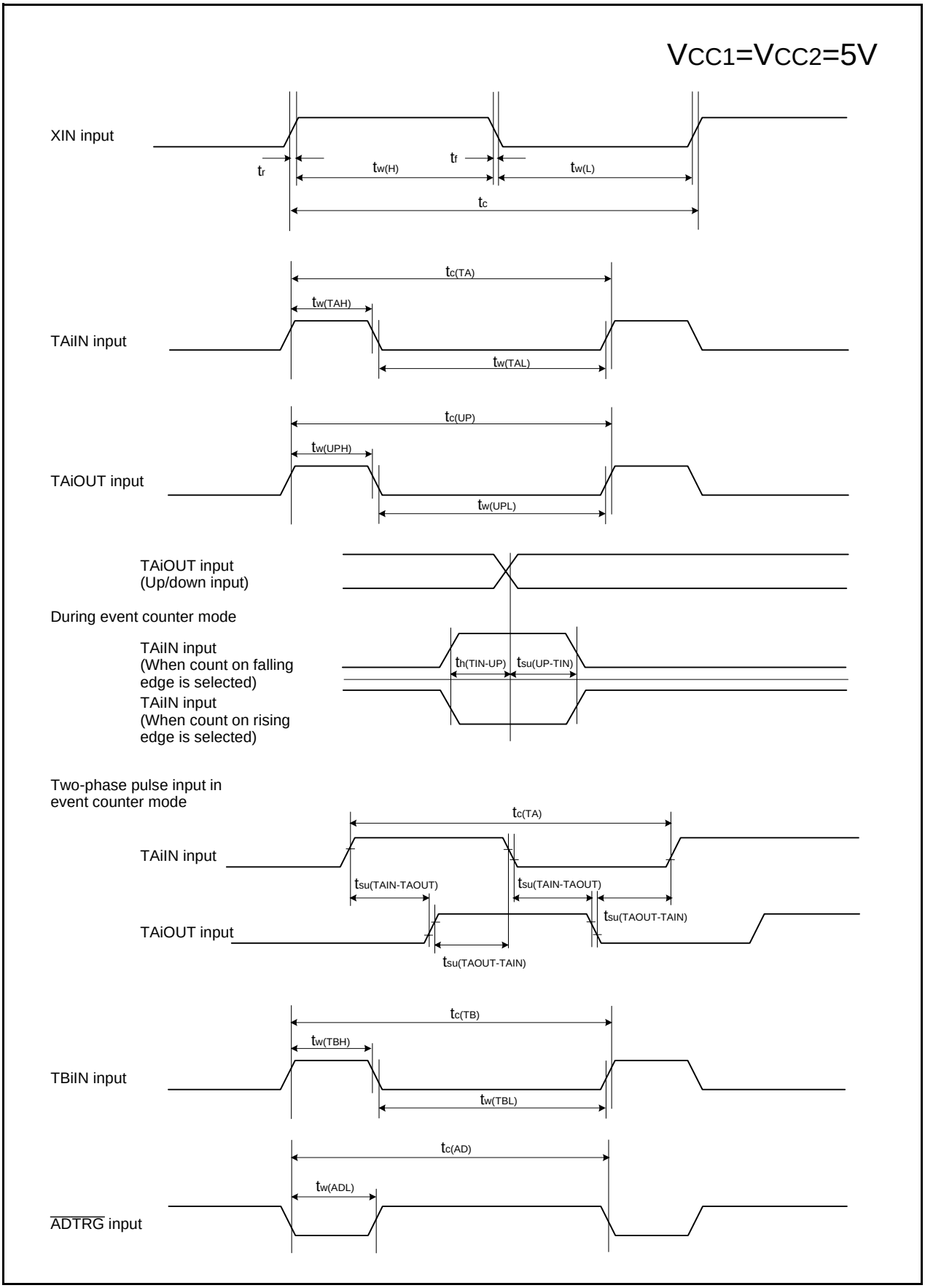


Figure 21.3 Timing Diagram (1)

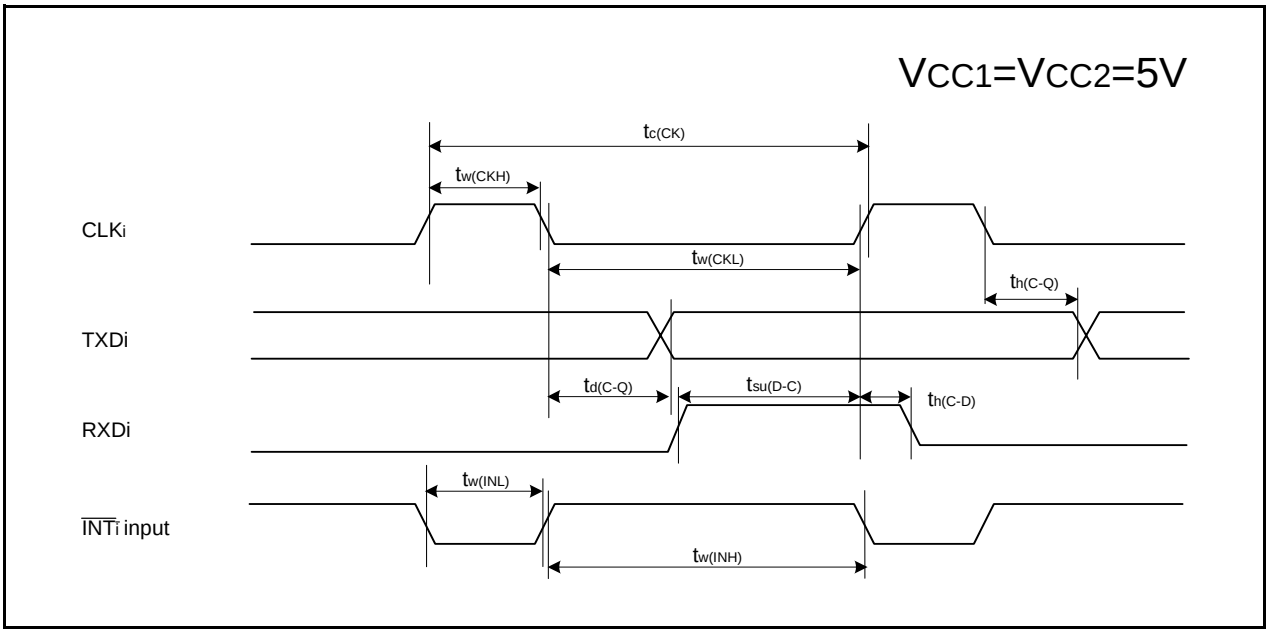
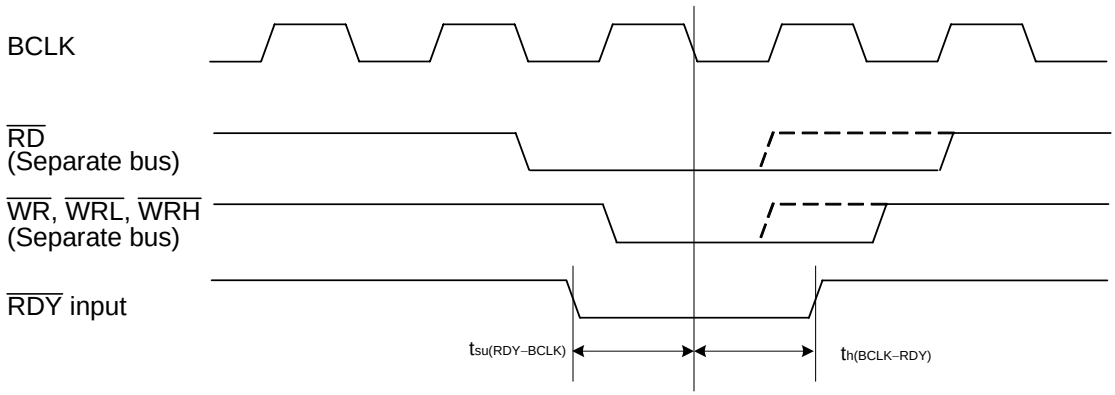


Figure 21.4 Timing Diagram (2)

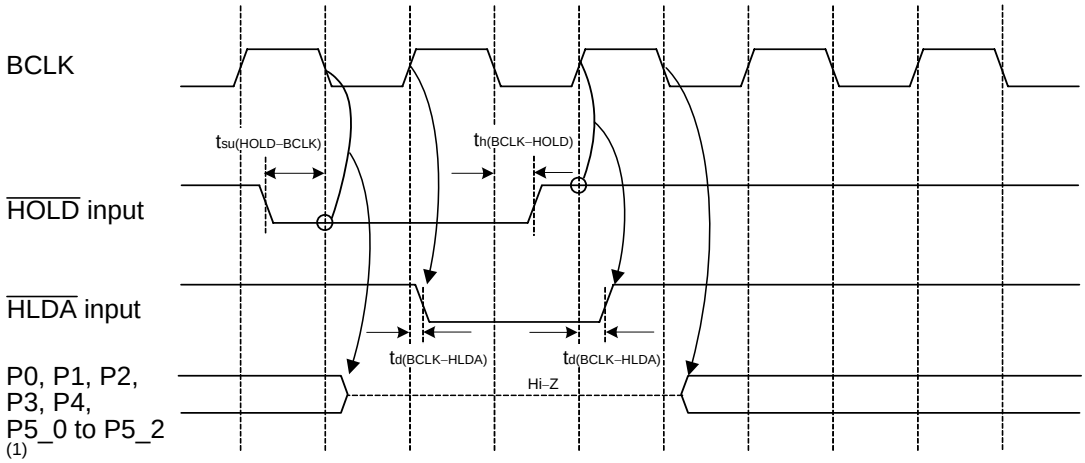
$V_{CC1}=V_{CC2}=5V$

Memory Expansion Mode, Microprocessor Mode

(Effective for setting with wait**)**



(Common to setting with wait and setting without wait**)**



NOTES:

1. These pins are set to high-impedance regardless of the input level of the BYTE pin, PM06 bit in PM0 register.

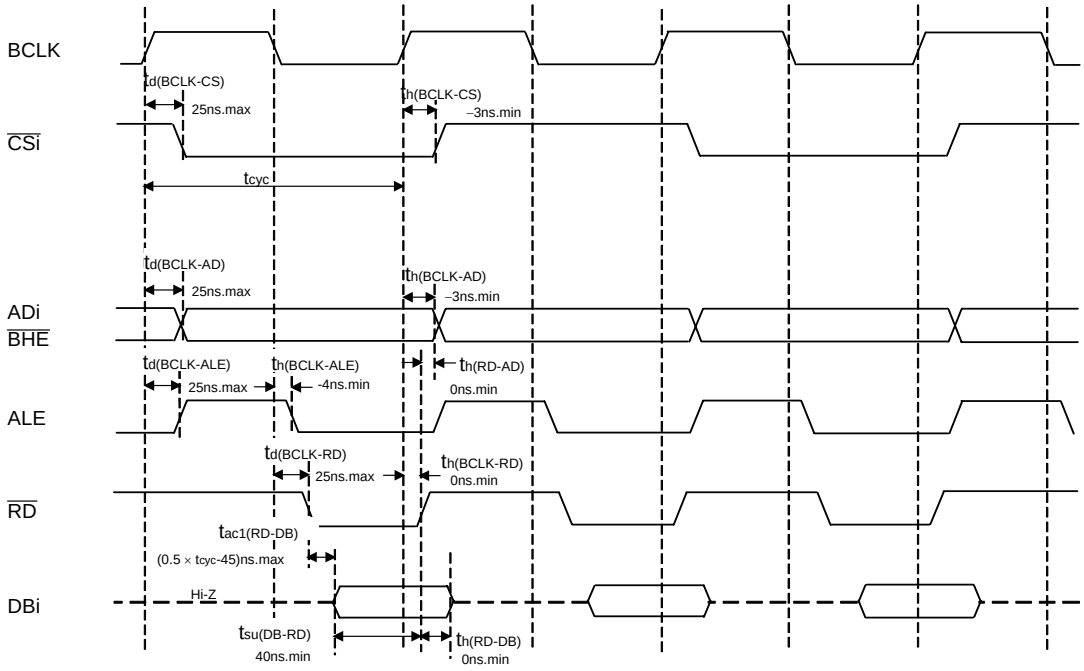
· Measuring conditions :
· $V_{CC1}=V_{CC2}=5V$
· Input timing voltage : Determined with $V_{IL}=1.0V$, $V_{IH}=4.0V$
· Output timing voltage : Determined with $V_{OL}=2.5V$, $V_{OH}=2.5V$

Figure 21.5 Timing Diagram (3)

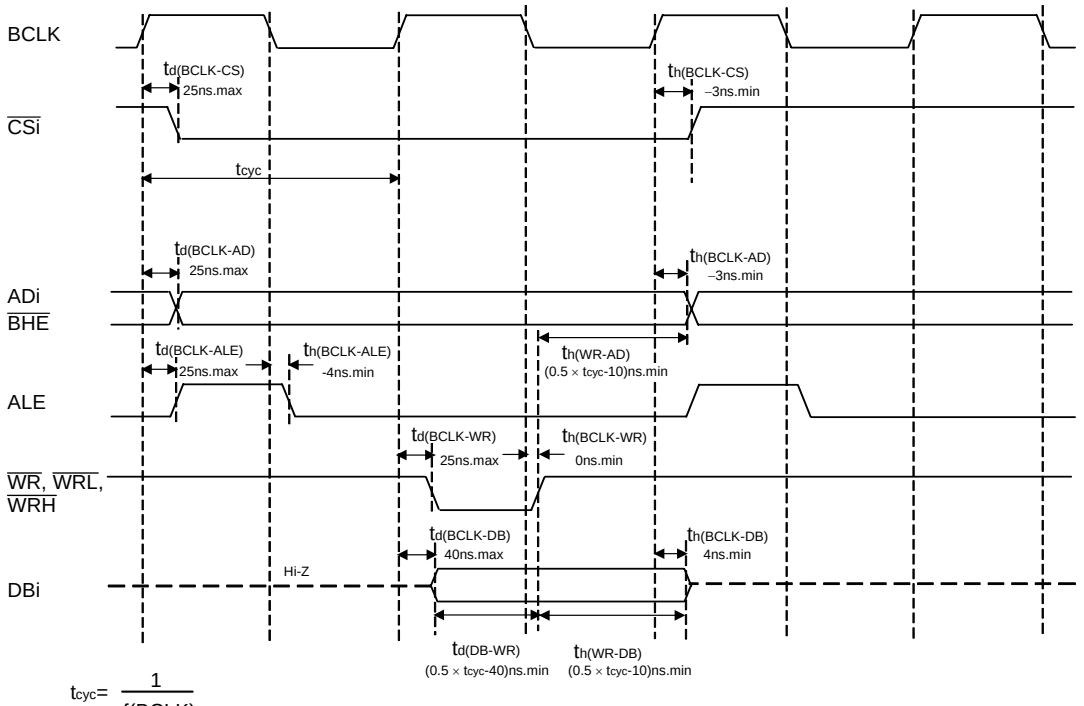
Memory Expansion Mode, Microprocessor Mode
(For setting with no wait**)**

$V_{CC1}=V_{CC2}=5V$

Read timing



Write timing



$$t_{cyc}=\frac{1}{f(BCLK)}$$

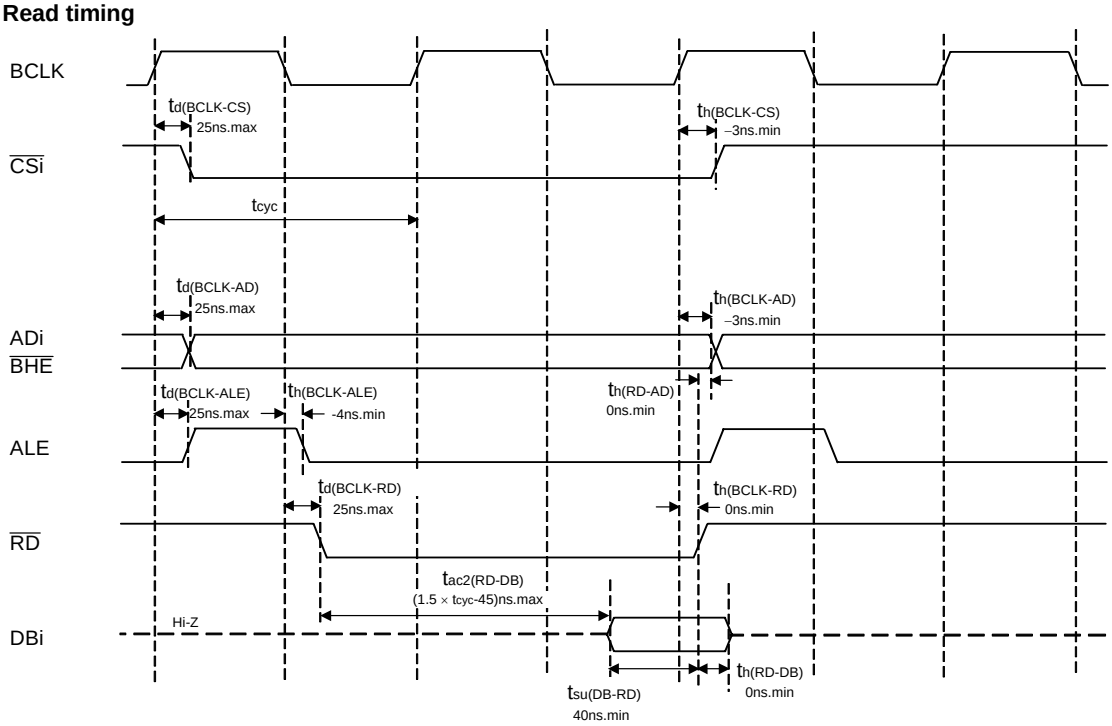
Measuring conditions
· $V_{CC1}=V_{CC2}=5V$
· Input timing voltage : $V_{IL}=0.8V$, $V_{IH}=2.0V$
· Output timing voltage : $V_{OL}=0.4V$, $V_{OH}=2.4V$

Figure 21.6 Timing Diagram (4)

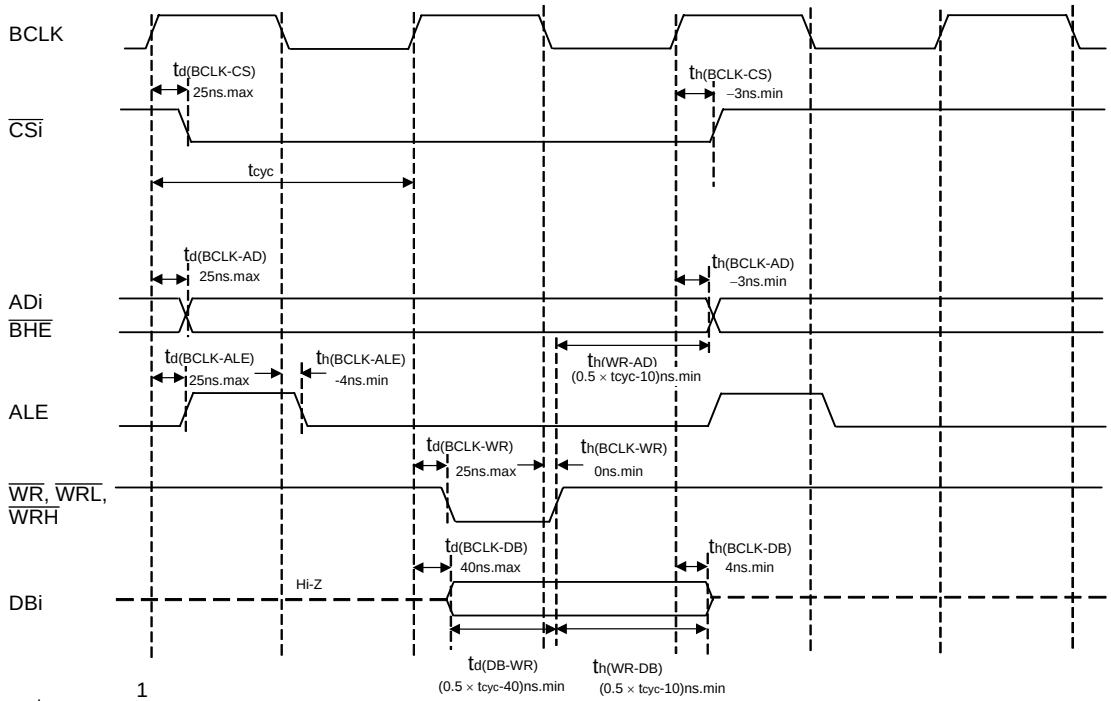
$V_{CC1}=V_{CC2}=5V$

Memory Expansion Mode, Microprocessor Mode
(for 1-wait setting and external area access**)**

Read timing



Write timing



$$t_{cyc}=\frac{1}{f(BCLK)}$$

Measuring conditions
· $V_{CC1}=V_{CC2}=5V$
· Input timing voltage : $V_{IL}=0.8V$, $V_{IH}=2.0V$
· Output timing voltage : $V_{OL}=0.4V$, $V_{OH}=2.4V$

Figure 21.7 Timing Diagram (5)

$V_{CC1}=V_{CC2}=3V$

Table 21.27 Electrical Characteristics (1) (1)

Symbol	Parameter			Measuring Condition	Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	HIGH Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7		I_{OH}=–1mA	V_{CC}–0.5		V_{CC}	V
V_{OH}	HIGH Output Voltage XOUT		HIGHPOWER	I_{OH}=–0.1mA	V_{CC}–0.5		V_{CC}	V
			LOWPOWER	I_{OH}=–50µA	V_{CC}–0.5		V_{CC}	
	HIGH Output Voltage XCOUT		HIGHPOWER	With no load applied		2.5		V
			LOWPOWER	With no load applied		1.6		
V_{OL}	LOW Output Voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7		I_{OL}=1mA			0.5	V
V_{OL}	LOW Output Voltage XOUT		HIGHPOWER	I_{OL}=0.1mA			0.5	V
			LOWPOWER	I_{OL}=50µA			0.5	
	LOW Output Voltage XCOUT		HIGHPOWER	With no load applied		0		V
			LOWPOWER	With no load applied		0		
V_{T+}-V_{T-}	Hysteresis	TA0IN to TA2IN, TB0IN to TB2IN, $\overline{INT0}$ to $\overline{INT4}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$ to $\overline{CTS2}$, RXD0 to RXD2, CLK0 to CLK2, TA0OUT to TA2OUT, $\overline{KI0}$ to $\overline{KI3}$, SCL0 to SCL2, SDA0 to SDA2			0.2		0.8	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$			0.2	(0.7)	1.8	V
I_{IH}	HIGH Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, $\overline{RESET}$, CNVSS, BYTE		V_I=3V			4.0	µA
I_{IL}	LOW Input Current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, XIN, $\overline{RESET}$, CNVSS, BYTE		V_I=0V			–4.0	µA
R_{PULLUP}	Pull-Up Resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7		V_I=0V	50	100	500	kΩ
R_{fXIN}	Feedback Resistance	XIN				3.0		MΩ
R_{fXCIN}	Feedback Resistance	XCIN				25		MΩ
V_{RAM}	RAM Retention Voltage			At stop mode	2.0			V

NOTES:
1. Referenced to V_{CC1} = V_{CC2} = 2.7 to 3.3V, V_{SS} = 0V at Topr = –20 to 85°C / –40 to 85°C, f(XIN)=10MHz no wait unless otherwise specified.

Table 21.28 Electrical Characteristics (2) (1)

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power Supply Current (V_{CC1}=V_{CC2}=2.7V to 3.6V)	In single-chip mode, the output pins are open and other pins are V_{SS}	Mask ROM	f(XIN)=10MHz No division		8	11	mA
			One Time Flash	f(XIN)=10MHz, No division		8	13	mA
			Flash Memory	f(XIN)=10MHz, No division		8	13	mA
			Flash Memory Program	f(XIN)=10MHz, VCC1=3.0V		12		mA
			One Time Flash Program	f(XIN)=10MHz, VCC1=3.0V		12		mA
			Flash Memory Erase	f(XIN)=10MHz, VCC1=3.0V		22		mA
			Mask ROM	f(XCIN)=32kHz Low power dissipation mode, ROM (3)		25		µA
			One Time Flash	f(XCIN)=32kHz Low power dissipation mode, RAM (3)		25		µA
				f(XCIN)=32kHz Low power dissipation mode, Flash Memory (3)		350		µA
			Flash Memory	f(XCIN)=32kHz Low power dissipation mode, RAM (3)		25		µA
				f(XCIN)=32kHz Low power dissipation mode, Flash Memory (3)		420		µA
			Mask ROM One Time Flash Flash Memory	f(XCIN)=32kHz Wait mode (2), Oscillation capability High		6.0		µA
				f(XCIN)=32kHz Wait mode (2), Oscillation capability Low		1.8		µA
				Stop mode Topr =25°C		0.7	3.0	µA

NOTES:
1. Referenced to V_{CC1}=V_{CC2}=2.7 to 3.3V, V_{SS} = 0V at T_{opr} = –20 to 85°C / –40 to 85°C, f(XIN)=10MHz unless otherwise specified.
2. With one timer operated using fC32.
3. This indicates the memory in which the program to be executed exists.

$V_{CC1}=V_{CC2}=3V$

Timing Requirements

(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.29 External Clock Input (XIN input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External Clock Input Cycle Time	(NOTE 2)		ns
$t_{w(H)}$	External Clock Input HIGH Pulse Width	(NOTE 3)		ns
$t_{w(L)}$	External Clock Input LOW Pulse Width	(NOTE 3)		ns
t_r	External Clock Rise Time		(NOTE 4)	ns
t_f	External Clock Fall Time		(NOTE 4)	ns

NOTES:

1. The condition is V_{CC1}=V_{CC2}=2.7 to 3.0V.
2. Calculated according to the V_{CC1} voltage as follows:

$$\frac{10^{-6}}{20 \times V_{CC1} - 44} \text{ [ns]}$$

3. Calculated according to the V_{CC1} voltage as follows:

$$\frac{10^{-6}}{20 \times V_{CC1} - 44} \times 0.4 \text{ [ns]}$$

4. Calculated according to the V_{CC1} voltage as follows:

$$-10 \times V_{CC1} + 45 \text{ [ns]}$$

Table 21.30 Memory Expansion Mode and Microprocessor Mode

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data Input Access Time (for setting with no wait)		(NOTE 1)	ns
$t_{ac2(RD-DB)}$	Data Input Access Time (for setting with wait)		(NOTE 2)	ns
$t_{su(DB-RD)}$	Data Input Setup Time	50		ns
$t_{su(RDY-BCLK)}$	$\overline{RDY}$ Input Setup Time	40		ns
$t_{su(HOLD-BCLK)}$	$\overline{HOLD}$ Input Setup Time	50		ns
$t_{h(RD-DB)}$	Data Input Hold Time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{RDY}$ Input Hold Time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{HOLD}$ Input Hold Time	0		ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 60 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 60 \text{ [ns]}$$

n is "2" for 1-wait setting.

$V_{CC1}=V_{CC2}=3V$

Timing Requirements
(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.31 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN Input Cycle Time	150		ns
$t_{w(TAH)}$	TAiIN Input HIGH Pulse Width	60		ns
$t_{w(TAL)}$	TAiIN Input LOW Pulse Width	60		ns

Table 21.32 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN Input Cycle Time	600		ns
$t_{w(TAH)}$	TAiIN Input HIGH Pulse Width	300		ns
$t_{w(TAL)}$	TAiIN Input LOW Pulse Width	300		ns

Table 21.33 Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN Input Cycle Time	300		ns
$t_{w(TAH)}$	TAiIN Input HIGH Pulse Width	150		ns
$t_{w(TAL)}$	TAiIN Input LOW Pulse Width	150		ns

Table 21.34 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN Input HIGH Pulse Width	150		ns
$t_{w(TAL)}$	TAiIN Input LOW Pulse Width	150		ns

Table 21.35 Timer A Input (Counter Increment/Decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT Input Cycle Time	3000		ns
$t_{w(UPH)}$	TAiOUT Input HIGH Pulse Width	1500		ns
$t_{w(UPL)}$	TAiOUT Input LOW Pulse Width	1500		ns
$t_{su(UP-TIN)}$	TAiOUT Input Setup Time	600		ns
$t_{h(TIN-UP)}$	TAiOUT Input Hold Time	600		ns

Table 21.36 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN Input Cycle Time	2		µs
$t_{su(TAIN-TAOUT)}$	TAiOUT Input Setup Time	500		ns
$t_{su(TAOUT-TAIN)}$	TAiIN Input Setup Time	500		ns

$V_{CC1}=V_{CC2}=3V$

Timing Requirements

(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.37 Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on one edge)	60		ns
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width (counted on both edges)	120		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width (counted on both edges)	120		ns

Table 21.38 Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	600		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	300		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	300		ns

Table 21.39 Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	600		ns
$t_{w(TBH)}$	TBiIN Input HIGH Pulse Width	300		ns
$t_{w(TBL)}$	TBiIN Input LOW Pulse Width	300		ns

Table 21.40 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{\text{ADTRG}}$ Input Cycle Time	1500		ns
$t_{w(ADL)}$	$\overline{\text{ADTRG}}$ Input LOW Pulse Width	200		ns

Table 21.41 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi Input Cycle Time	300		ns
$t_{w(CKH)}$	CLKi Input HIGH Pulse Width	150		ns
$t_{w(CKL)}$	CLKi Input LOW Pulse Width	150		ns
$t_{d(C-Q)}$	TXDi Output Delay Time		160	ns
$t_{h(C-Q)}$	TXDi Hold Time	0		ns
$t_{su(D-C)}$	RXDi Input Setup Time	100		ns
$t_{h(C-D)}$	RXDi Input Hold Time	90		ns

Table 21.42 External Interrupt $\overline{\text{INTi}}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{\text{INTi}}$ Input HIGH Pulse Width	380		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ Input LOW Pulse Width	380		ns

$V_{CC1}=V_{CC2}=3V$

Switching Characteristics

(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.43 Memory Expansion and Microprocessor Modes (for setting with no wait)

Symbol	Parameter		Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address Output Delay Time	See **Figure 21.8**		30	ns
th(BCLK-AD)	Address Output Hold Time (in relation to BCLK)		0		ns
th(RD-AD)	Address Output Hold Time (in relation to RD)		0		ns
th(WR-AD)	Address Output Hold Time (in relation to WR)		(NOTE 2)		ns
td(BCLK-CS)	Chip Select Output Delay Time			30	ns
th(BCLK-CS)	Chip Select Output Hold Time (in relation to BCLK)		0		ns
td(BCLK-ALE)	ALE Signal Output Delay Time			25	ns
th(BCLK-ALE)	ALE Signal Output Hold Time		–4		ns
td(BCLK-RD)	RD Signal Output Delay Time			30	ns
th(BCLK-RD)	RD Signal Output Hold Time		0		ns
td(BCLK-WR)	WR Signal Output Delay Time			30	ns
th(BCLK-WR)	WR Signal Output Hold Time		0		ns
td(BCLK-DB)	Data Output Delay Time (in relation to BCLK)			40	ns
th(BCLK-DB)	Data Output Hold Time (in relation to BCLK) (3)		4		ns
td(DB-WR)	Data Output Delay Time (in relation to WR)		(NOTE 1)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR) (3)		(NOTE 2)		ns
td(BCLK-HLDA)	HLDA Output Delay Time			40	ns

NOTES:

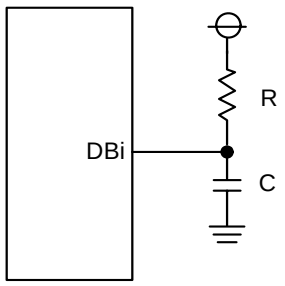
1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5\times10^{9}}{f(BCLK)} - 40[ns]$$ f(BCLK) is 12.5MHz or less.

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5\times10^{9}}{f(BCLK)} - 10[ns]$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.
   Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
   Hold time of data bus is expressed in
   $t = -CR \times \ln(1 - V_{OL} / V_{CC1})$
   by a circuit of the right figure.
   For example, when V_{OL} = 0.2V_{CC1}, C = 30pF, R = 1kΩ, hold time of output "L" level is
   $t = -30pF \times 1k\Omega \times \ln(1 - 0.2V_{CC1} / V_{CC1})$
   = 6.7ns.



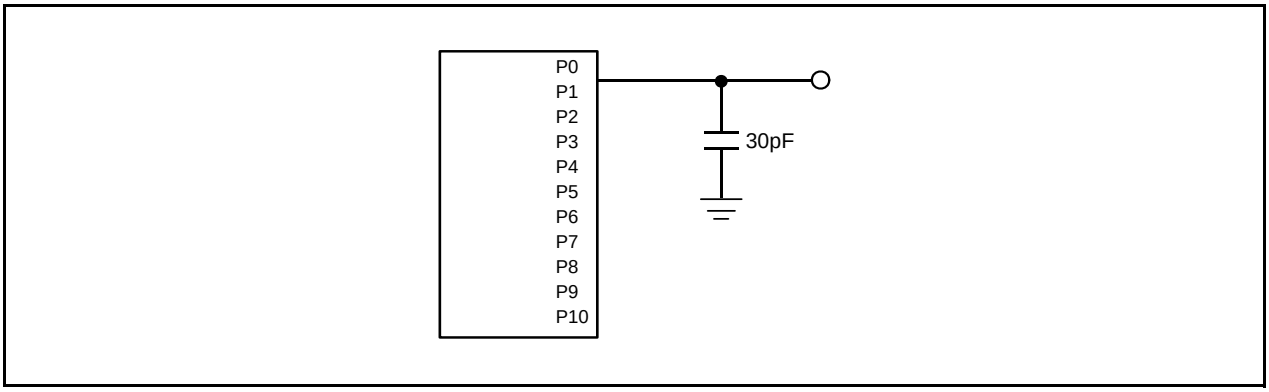


Figure 21.8 Ports P0 to P10 Measurement Circuit

$V_{CC1}=V_{CC2}=3V$

Switching Characteristics

(V_{CC1} = V_{CC2} = 3V, V_{SS} = 0V, at T_{opr} = –20 to 85°C / –40 to 85°C unless otherwise specified)

Table 21.44 Memory Expansion and Microprocessor Modes (for 1 wait setting and external area access)

Symbol	Parameter		Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address Output Delay Time	See **Figure 21.8**		30	ns
$t_{h(BCLK-AD)}$	Address Output Hold Time (in relation to BCLK)		0		ns
$t_{h(RD-AD)}$	Address Output Hold Time (in relation to RD)		0		ns
$t_{h(WR-AD)}$	Address Output Hold Time (in relation to WR)		(NOTE 2)		ns
$t_{d(BCLK-CS)}$	Chip Select Output Delay Time			30	ns
$t_{h(BCLK-CS)}$	Chip Select Output Hold Time (in relation to BCLK)		0		ns
$t_{d(BCLK-ALE)}$	ALE Signal Output Delay Time			25	ns
$t_{h(BCLK-ALE)}$	ALE Signal Output Hold Time		-4		ns
$t_{d(BCLK-RD)}$	RD Signal Output Delay Time			30	ns
$t_{h(BCLK-RD)}$	RD Signal Output Hold Time		0		ns
$t_{d(BCLK-WR)}$	WR Signal Output Delay Time			30	ns
$t_{h(BCLK-WR)}$	WR Signal Output Hold Time		0		ns
$t_{d(BCLK-DB)}$	Data Output Delay Time (in relation to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data Output Hold Time (in relation to BCLK) (3)		4		ns
$t_{d(DB-WR)}$	Data Output Delay Time (in relation to WR)		(NOTE 1)		ns
th(WR-DB)	Data Output Hold Time (in relation to WR)(3)		(NOTE 2)		ns
$t_{d(BCLK-HLDA)}$	HLDA Output Delay Time			40	ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

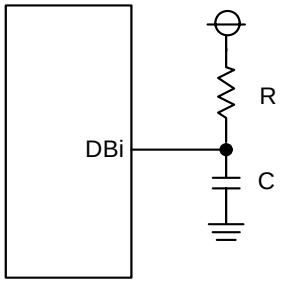
$$\frac{(n-0.5)\times 10^9}{f(BCLK)} - 40[ns]$$

n is “1” for 1-wait setting, f(BCLK) is 12.5MHz or less.

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5\times 10^9}{f(BCLK)} - 10[ns]$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in
t = –CR X ln (1–V_{OL} / V_{CC1})
by a circuit of the right figure.
For example, when V_{OL} = 0.2V_{CC1}, C = 30pF, R = 1kΩ, hold time of output ”L” level is
t = –30pF X 1kΩ X ln(1–0.2V_{CC1} / V_{CC1})
= 6.7ns.



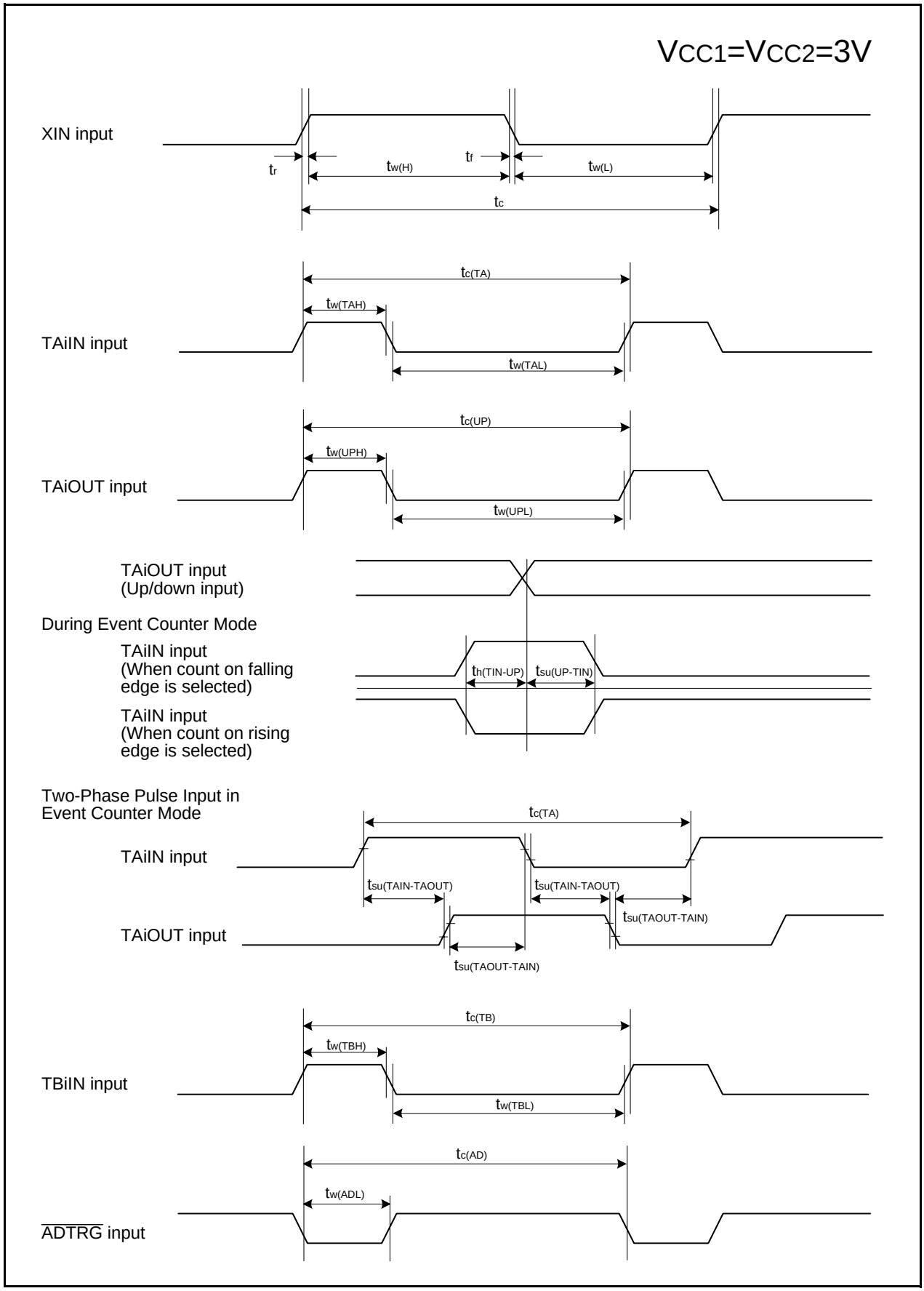


Figure 21.9 Timing Diagram (1)

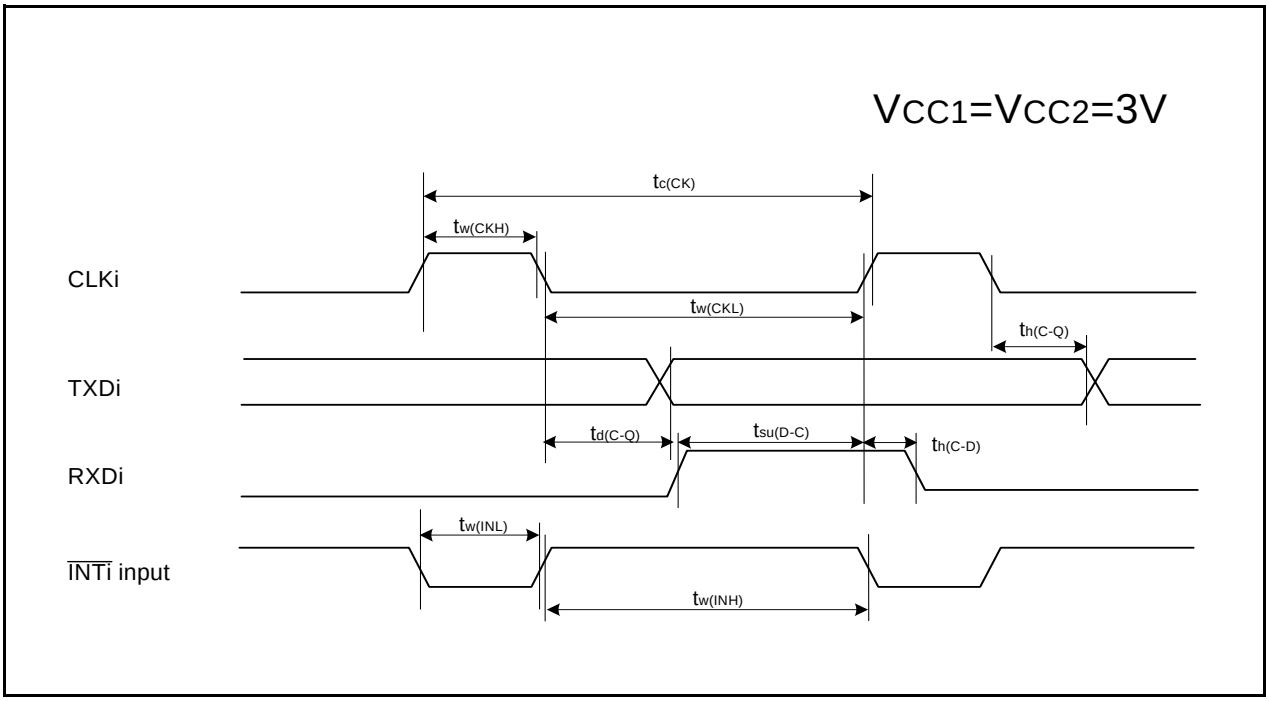
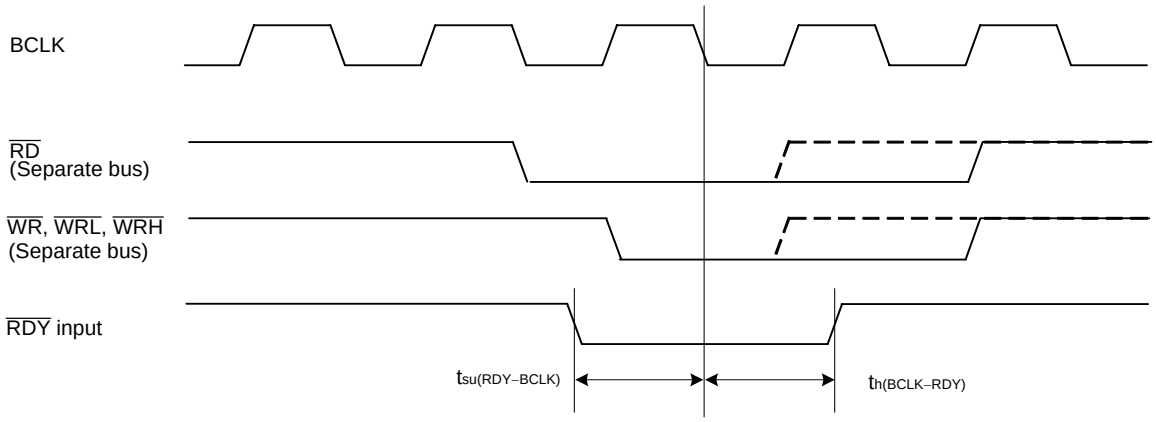


Figure 21.10 Timing Diagram (2)

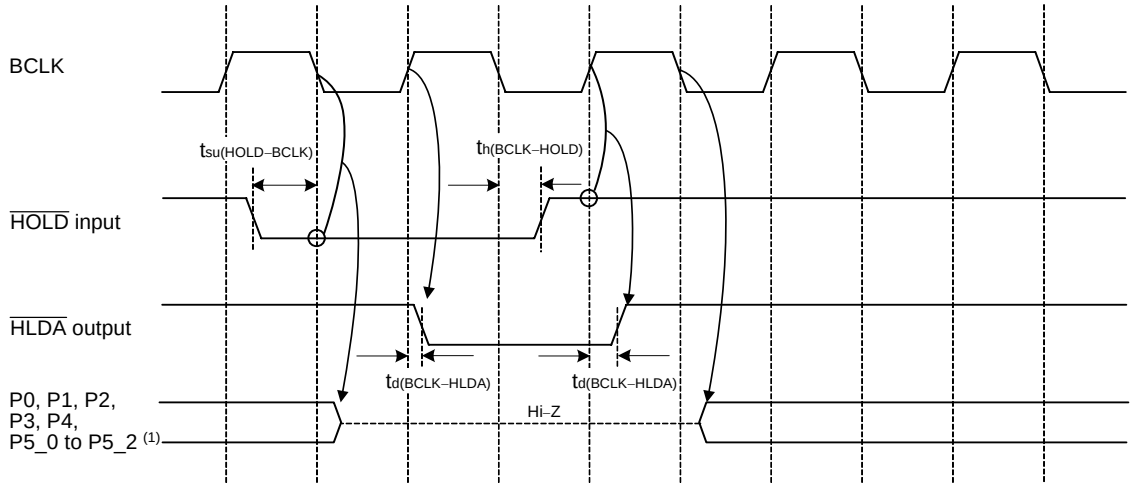
Memory Expansion Mode, Microprocessor Mode

$V_{CC1}=V_{CC2}=3V$

(Effective for setting with wait)



(Common to setting with wait and setting without wait)



NOTES:
1. These pins are set to high-impedance regardless of the input level of the BYTE pin, PM06 bit in PM0 register.

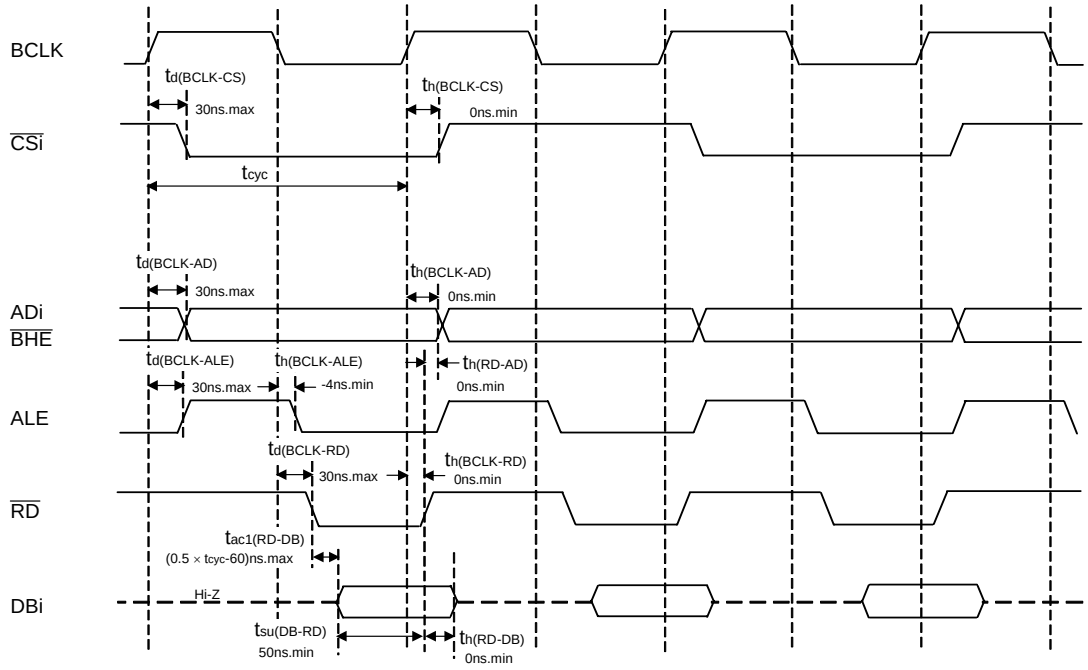
Measuring conditions :
· $V_{CC1}=V_{CC2}=3V$
· Input timing voltage : Determined with $V_{IL}=0.6V$, $V_{IH}=2.4V$
· Output timing voltage : Determined with $V_{OL}=1.5V$, $V_{OH}=1.5V$

Figure 21.11 Timing Diagram (3)

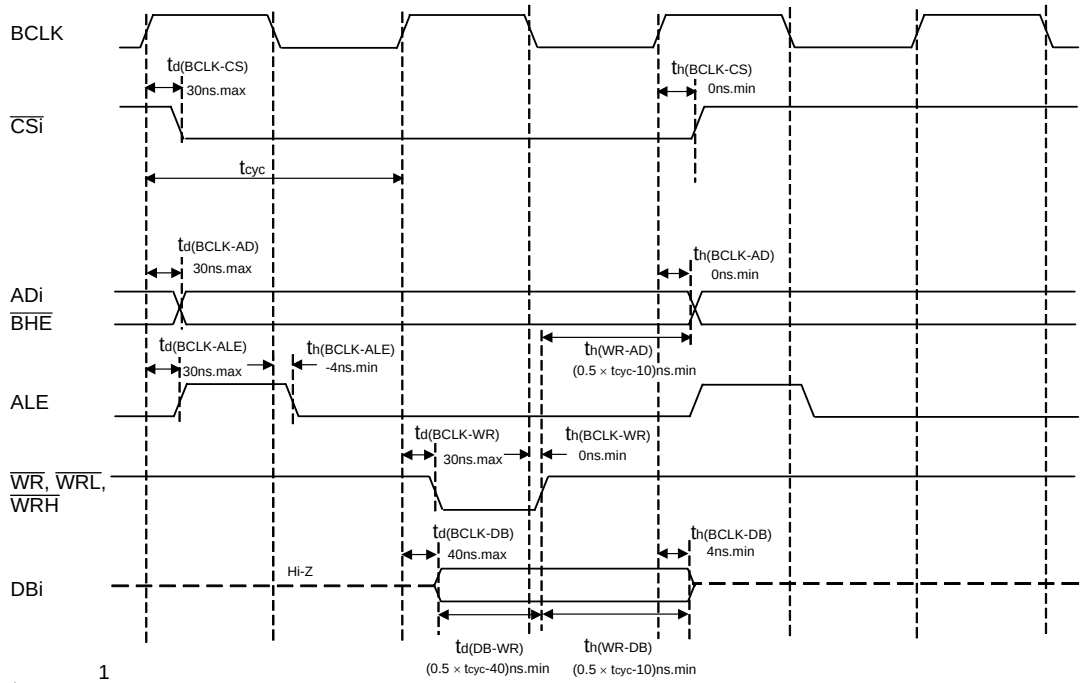
Memory Expansion Mode, Microprocessor Mode
(for setting with no wait**)**

$V_{CC1}=V_{CC2}=3V$

Read timing



Write timing



$$t_{cyc}=\frac{1}{f(BCLK)}$$

Measuring conditions
· $V_{CC1}=V_{CC2}=3V$
· Input timing voltage : $V_{IL}=0.6V$, $V_{IH}=2.4V$
· Output timing voltage : $V_{OL}=1.5V$, $V_{OH}=1.5V$

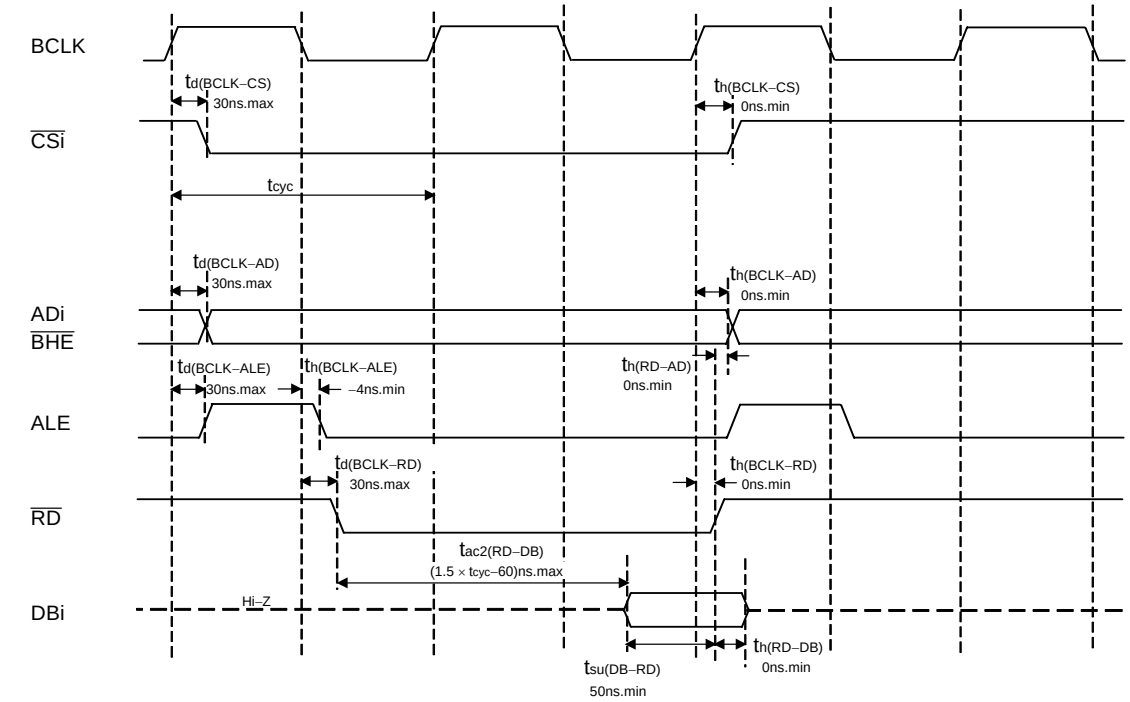
Figure 21.12 Timing Diagram (4)

Memory Expansion Mode, Microprocessor Mode

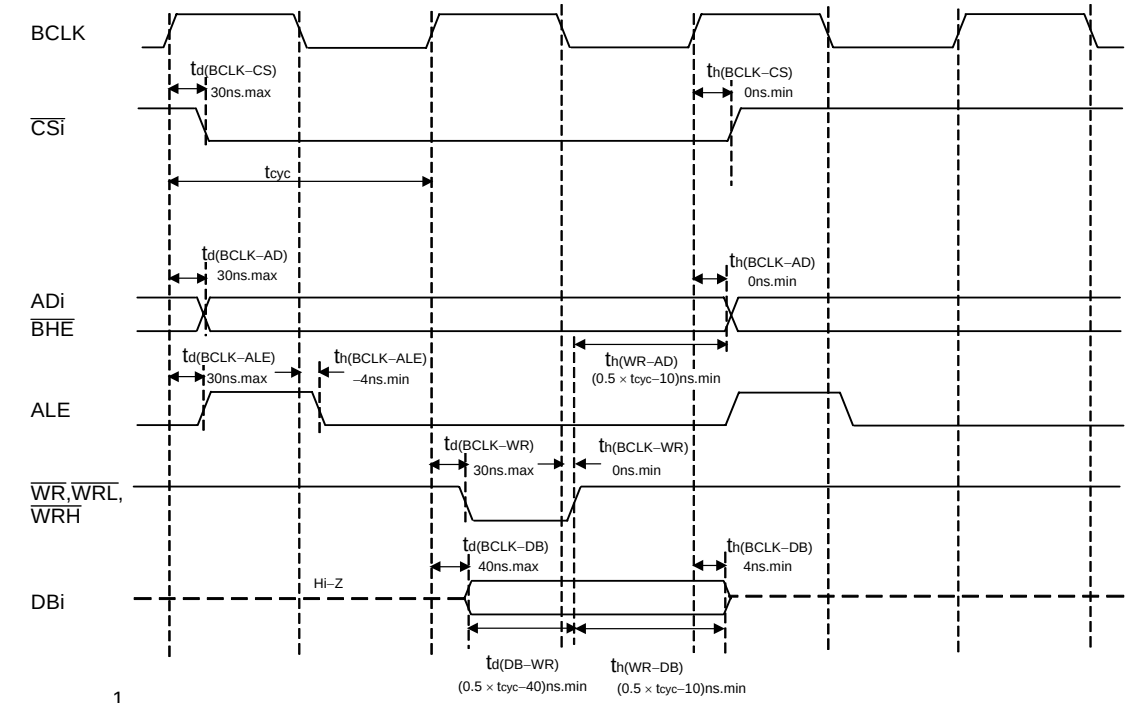
(for 1-wait setting and external area access)

$V_{CC1}=V_{CC2}=3V$

Read timing



Write timing



$$t_{cyc}=\frac{1}{f(BCLK)}$$

Measuring conditions

- $V_{CC1}=V_{CC2}=3V$
- Input timing voltage : $V_{IL}=0.6V$, $V_{IH}=2.4V$
- Output timing voltage : $V_{OL}=1.5V$, $V_{OH}=1.5V$

Figure 21.13 Timing Diagram (5)

22. Usage Precaution

22.1 SFR

22.1.1 Register Settings

Table 22.1 Registers with Write-only Bits which can only be written to. Set these registers with immediate values. When establishing the next value by altering the present value, write the present value to the RAM as well as to the register. Transfer the next value to the register after making changes in the RAM.

Table 22.1 Registers with Write-only Bits

Register	Symbol	Address
Watchdog timer start register	WDC	000E
UART0 bit rateregister	U0BRG	03A1
UART1 bit rateregister	U1BRG	03A9
UART2 bit rate register	U2BRG	0379
UART0 Transmit buffer register	U0TB	03A3 to 03A2
UART1 Transmit buffer register	U1TB	03AB to 03AA
UART2 Transmit buffer register	U2TB	037B to 037A
Ups and downs flag	UDF	0384
Timer 0 register	TA0	0387 to 0386
Timer 1 register	TA1	0389 to 0388
Timer 2 register	TA2	038B to 038A

22.2 Reset

When supplying power to the microcomputer, the power supply voltage applied to the VCC1 and VCC2 pins must meet the conditions of SVCC.

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
SVCC	Power supply rising gradient (VCC1)	0.05			V/ms

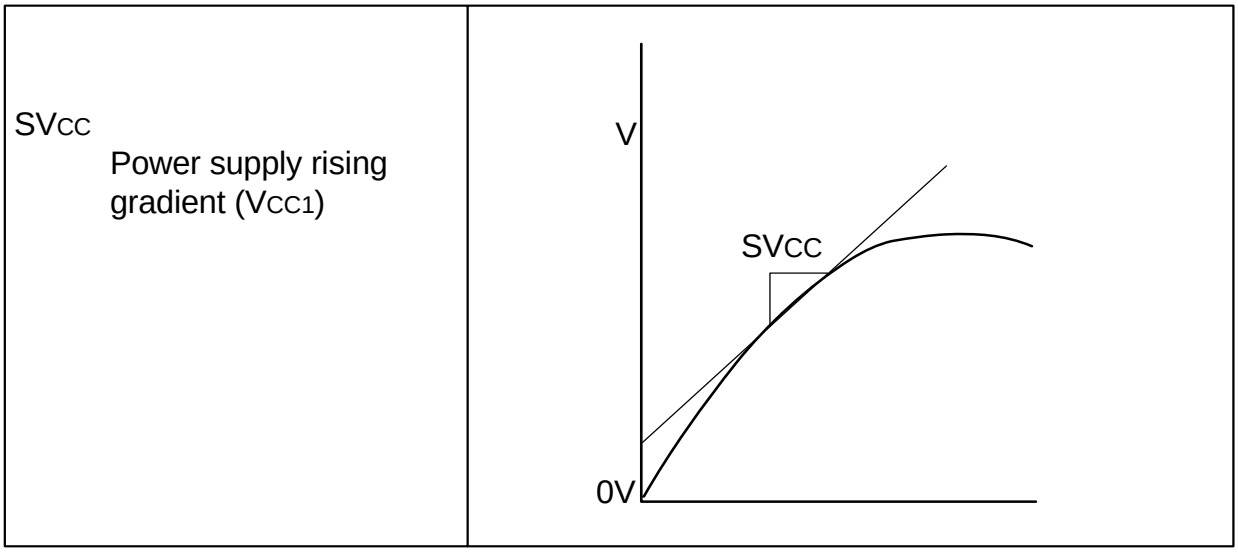


Figure 22.1 Timing of SVCC

22.3 Bus

- The ROMless version can operate only in the microprocessor mode, connect the CNVSS pin to VCC1.
- When resetting CNVSS pin with “H” input, contents of internal ROM cannot be read out.

22.4 Precautions for Power Control

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

- Set the MR0 bit in the TAiMR register (i=0 to 2) to “0” (pulse is not output) to use the timer A to exit stop mode.

- When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not execute any instructions which can generate a write to RAM between the JMP.B and WAIT instructions. Disable the DMA transfers, if a DMA transfer may occur between the JMP.B and WAIT instructions. After the WAIT instruction, insert at least 4 NOP instructions. When entering wait mode, the instruction queue roadstead the instructions following WAIT, and depending on timing, some of these may execute before the microcomputer enters wait mode.

Program example when entering wait mode

```
Program Example:        JMP.B   L1      ; Insert JMP.B instruction before WAIT instruction
                L1:
                        FSET    I       ;
                        WAIT            ; Enter wait mode
                        NOP             ; More than 4 NOP instructions
                        NOP
                        NOP
                        NOP
```

- When entering stop mode, insert a JMP.B instruction immediately after executing an instruction which sets the CM10 bit in the CM1 register to Åg1Åh, and then insert at least 4 NOP instructions. When entering stop mode, the instruction queue reads ahead the instructions following the instruction which sets the CM10 bit to “1” (all clock stops), and, some of these may execute before the microcomputer enters stop mode or before the interrupt routine for returning from stop mode.

Program example when entering stop mode

```
Program Example:        FSET    I
                        BSET    CM10    ; Enter stop mode
                        JMP.B   L2      ; Insert JMP.B instruction
                L2:
                        NOP             ; More than 4 NOP instructions
                        NOP
                        NOP
                        NOP
```

- Wait the main clock oscillation stabilizes, before switching the clock source for CPU clock to the main clock.
- Similarly, wait until the sub clock oscillates stably before switching the clock source for CPU clock to the sub clock.

• Suggestions to reduce power consumption

Ports

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A pass current flows in input ports that high-impedance state. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

A/D converter

When A/D conversion is not performed, set the VCUT bit of ADiCON1 register to “0” (no VREF connection).

When A/D conversion is performed, start the A/D conversion at least 1 µs or longer after setting the VCUT bit to “1” (VREF connection).

Stopping peripheral functions

Use the CM0 register CM02 bit to stop the unnecessary peripheral functions during wait mode.

However, because the peripheral function clock (fC32) generated from the sub-clock does not stop, this measure is not conducive to reducing the power consumption of the chip. If low speed mode or low power dissipation mode is to be changed to wait mode, set the CM02 bit to “0” (do not peripheral function clock stopped when in wait mode), before changing wait mode.

Switching the oscillation-driving capacity

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

22.5 Precautions for Protect

Set the PRC2 bit to “1” (write enabled) and then write to any address, and the PRC2 bit will be cleared to “0” (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to “1”. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to “1” and the next instruction.

22.6 Precautions for Interrupt

22.6.1 Reading address 00000h

Do not read the address 00000h in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000h during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to “0”.
If the address 00000h is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to “0”. This factors a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

22.6.2 Setting the SP

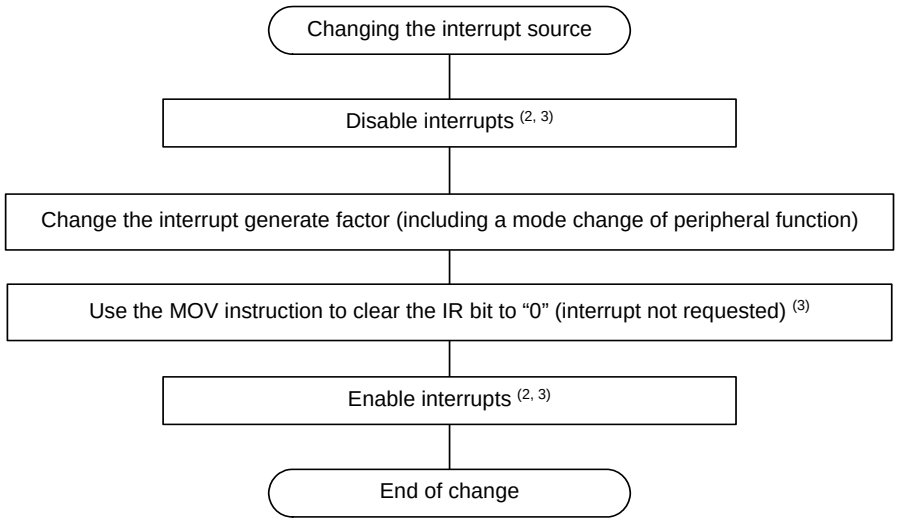
Set any value in the SP(USP, ISP) before accepting an interrupt. The SP(USP, ISP) is cleared to “0000h” after reset. Therefore, if an interrupt is accepted before setting any value in the SP(USP, ISP), the program may go out of control.
Especially when using $\overline{\text{NMI}}$ interrupt, set a value in the ISP at the beginning of the program. For the first and only the first instruction after reset, all interrupts including $\overline{\text{NMI}}$ interrupt are disabled.

22.6.3 The $\overline{\text{NMI}}$ Interrupt

The $\overline{\text{NMI}}$ interrupt cannot be disabled. If this interrupt is unused, connect the $\overline{\text{NMI}}$ pin to VCC1 via a resistor (pull-up).
The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8_5 bit in the P8 register. Note that the P8_5 bit can only be read when determining the pin level in $\overline{\text{NMI}}$ interrupt routine.
Stop mode cannot be entered into while input on the $\overline{\text{NMI}}$ pin is low. This is because while input on the $\overline{\text{NMI}}$ pin is low the CM10 bit in the CM1 register is fixed to “0”.
Do not go to wait mode while input on the $\overline{\text{NMI}}$ pin is low. This is because when input on the $\overline{\text{NMI}}$ pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
The low and high level durations of the input signal to the $\overline{\text{NMI}}$ pin must each be 2 CPU clock cycles + 300 ns or more.

22.6.4 Changing the Interrupt Generate Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to "1" (interrupt requested). If you changed the interrupt generate factor for an interrupt that needs to be used, be sure to clear the IR bit for that interrupt to "0" (interrupt not requested).
Changing the interrupt generate factor refered to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the generate factor, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to "0" (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.
Figure 22.2 shows the Procedure for Changing the Interrupt Generate Factor.



IR bit: A bit in the interrupt control register for the interrupt whose interrupt generate factor is to be changed

NOTES :
1. The above settings must be executed individually. Do not execute two or more settings simultaneously (using one instruction).
2. Use the I flag for the $\overline{\text{INTi}}$ interrupt (i = 0 to 4).
For the interrupts from peripheral functions other than the $\overline{\text{INTi}}$ interrupt, turn off the peripheral function that is the source of the interrupt in order not to generate an interrupt request before changing the interrupt generate factor. In this case, if the maskable interrupts can all be disabled without causing a problem, use the I flag. Otherwise, use the corresponding ILVL2 to ILVL0 bit for the interrupt whose interrupt generate factor is to be changed.
3. Refer to **20.5.6 Rewrite the Interrupt Control Register** for details about the instructions to use and the notes to be taken for instruction execution.

Figure 22.2 Procedure for Changing the Interrupt Generate Factor

22.6.5 $\overline{\text{INT}}$ Interrupt

- Either an "L" level of at least tW(INH) or an "H" level of at least tW(INL) width is necessary for the signal input to pins $\overline{\text{INT0}}$ through $\overline{\text{INT4}}$ regardless of the CPU operation clock.
- If the POL bit in the INT0IC to INT4IC registers or the IFSR7 to IFSR0 bits in the IFSR register are changed, the IR bit may inadvertently set to 1 (interrupt requested). Be sure to clear the IR bit to 0 interrupt not requested) after changing any of those register bits.

22.6.6 Rewrite the Interrupt Control Register

(a) The interrupt control register for any interrupt should be modified in places where no requests for that interrupt may occur. Otherwise, disable the interrupt before rewriting the interrupt control register.

(b) To rewrite the interrupt control register for any interrupt after disabling that interrupt, be careful with the instruction to be used.

- Changing any bit other than the IR bit

  If while executing an instruction, a request for an interrupt controlled by the register being modified occurs, the IR bit in the register may not be set to "1" (interrupt requested), with the result that the interrupt request is ignored. If such a situation presents a problem, use the instructions shown below to modify the register.
  Usable instructions: AND, OR, BCLR, BSET

- Changing the IR bit

  Depending on the instruction used, the IR bit may not always be cleared to "0" (interrupt not requested). Therefore, be sure to use the MOV instruction to clear the IR bit.

(c) When using the I flag to disable an interrupt, refer to the sample program fragments shown below as you set the I flag. (Refer to (b) for details about rewrite the interrupt control registers in the sample program fragments.)

Examples 1 through 3 show how to prevent the I flag from being set to "1" (interrupts enabled) before the interrupt control register is rewrited, owing to the effects of the internal bus and the instruction queue buffer.

Example 1:Using the NOP instruction to keep the program waiting until the interrupt control register is modified

```
INT_SWITCH1:
    FCLR    I           ; Disable interrupts.
    AND.B   #00h, 0055h ; Set the TA0IC register to "00h".
    NOP                 ;
    NOP
    FSET    I           ; Enable interrupts.
```

The number of NOP instruction is as follows.
When using HOLD function : 4.

Example 2:Using the dummy read to keep the FSET instruction waiting

```
INT_SWITCH2:
    FCLR    I           ; Disable interrupts.
    AND.B   #00h, 0055h ; Set the TA0IC register to "00h".
    MOV.W   MEM, R0     ; Dummy read.
    FSET    I           ; Enable interrupts.
```

Example 3:Using the POPC instruction to changing the I flag

```
INT_SWITCH3:
    PUSHC   FLG
    FCLR    I           ; Disable interrupts.
    AND.B   #00h, 0055h ; Set the TA0IC register to "00h".
    POPC    FLG         ; Enable interrupts.
```

22.6.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

22.7 Precautions for DMAC

22.7.1 Write to DMAE Bit in DMiCON Register

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

Conditions

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

Steps

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

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

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

NOTES:

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

22.8 Precautions for Timers

22.8.1 Timer A

22.8.1.1 Timer A (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 2) register and the TAi register before setting the TAiS bit in the TABSR register to “1” (count starts).
Always make sure the TAiMR register is modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

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

22.8.1.2 Timer A (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 2) register, the TAi register, the UDF register, the ONSF register TAZIE, TA0TGL and TA0TGH bits and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
Always make sure the TAiMR register, the UDF register, the TAZIE, TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

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

22.8.1.3 Timer A (One-shot Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 2) register, the TAi register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
Always make sure the TAiMR register, the TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

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

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

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

The IR bit is set to “1” when timer operation mode is set with any of the following procedures:

- Select one-shot timer mode after reset.
- Change an operation mode from timer mode to one-shot timer mode.
- Change an operation mode from event counter mode to one-shot timer mode.

To use the Timer Ai interrupt (the IR bit), set the IR bit to “0” after the changes listed above have been made.

When a trigger occurs, while counting, a counter reloads the reload register to continue counting after generating a re-trigger and counting down once. To generate a trigger while counting, generate a second trigger between occurring the previous trigger and operating longer than one cycle of a timer count source.

22.8.1.4 Timer A (Pulse Width Modulation Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 2) register, the TAi register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
Always make sure the TAiMR register, TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

The IR bit is set to “1” when setting a timer operation mode with any of the following procedures:

- Select the PWM mode after reset.
- Change an operation mode from timer mode to PWM mode.
- Change an operation mode from event counter mode to PWM mode.

To use the Timer Ai interrupt (interrupt request bit), set the IR bit to “0” by program after the above listed changes have been made.

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

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

22.8.2 Timer B

22.8.2.1 Timer B (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR or the TBSR register to “1” (count starts). Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not.

A value of a counter, while counting, can be read in TBi register at any time. “FFFFh” is read while reloading. Setting value is read between setting values in TBi register at count stop and starting a counter.

22.8.2.2 Timer B (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR or the TBSR register to “1” (count starts). Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not.

The counter value can be read out on-the-fly at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always “FFFFh”. If the TBi register is read after setting a value in it while not counting but before the counter starts counting, the read value is the one that has been set in the register.

22.8.2.3 Timer B (Pulse Period/pulse Width Measurement Mode)

The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR (i = 0 to 2) register before setting the TBiS bit in the TABSR or the TBSR register to “1” (count starts).
Always make sure the TBiMR register is modified while the TBiS bit remains “0” (count stops) regardless whether after reset or not. To clear the MR3 bit to “0” by writing to the TBiMR register while the TBiS bit = 1 (count starts), be sure to write the same value as previously written to the TM0D0, TM0D1, MR0, MR1, TCK0 and TCK1 bits and a 0 to the MR2 bit.

The IR bit in the TBiIC register (i=0 to 2) goes to “1” (interrupt request), when an effective edge of a measurement pulse is input or Timer Bi is overflowed. The factor of interrupt request can be determined by use of the MR3 bit in the TBiMR register within the interrupt routine.

If the source of interrupt cannot be identified by the MR3 bit such as when the measurement pulse input and a timer overflow occur at the same time, use another timer to count the number of times Timer B has overflowed.

To set the MR3 bit to “0” (no overflow), set TBiMR register with setting the TBiS bit to “1” and counting the next count source after setting the MR3 bit to “1” (overflow).

Use the IR bit to detect only overflows. Use the MR3 bit only to determine the interrupt factor.

When a count is started and the first effective edge is input, an indeterminate value is transferred to the reload register. At this time, Timer Bi interrupt request is not generated.

A value of the counter is indeterminate at the beginning of a count. MR3 may be set to “1” and Timer Bi interrupt request may be generated between a count start and an effective edge input.

For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an “H” level width or an “L” level width.

22.9 Precautions for Serial interface

22.9.1 Clock Synchronous Serial I/O

22.9.1.1 Transmission/reception

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

22.9.1.2 Transmission

When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit in the UiC1 register= 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in UiTB register)
- If $\overline{\text{CTS}}$ function is selected, input on the $\overline{\text{CTSi}}$ pin = L

22.9.1.3 Reception

In operating the clock-synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TXDi pin when receiving data.

When an internal clock is selected, set the TE bit in the UiC1 register (i = 0 to 2) to 1 (transmission enabled) and write dummy data to the UiTB register, and the shift clock will thereby be generated.
When an external clock is selected, set the TE bit to 1 and write dummy data to the UiTB register, and the shift clock will be generated when the external clock is fed to the CLKi input pin.

When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the RE bit in the UiC1 register (i = 0 to 2) = 1 (data present in the UiRB register), an overrun error occurs and the OER bit in the UiRB register is set to “1” (overrun error occurred). In this case, because the content of the UiRB register is indeterminate, a corrective measure must be taken by programs on the transmit and receive sides so that the valid data before the overrun error occurred will be retransmitted. Note that when an overrun error occurred, the IR bit in the SiRIC register does not change state.

To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.

When an external clock is selected, the conditions must be met while if the CKPOL bit = 0, the external clock is in the high state; if the CKPOL bit = 1, the external clock is in the low state.

- The RE bit in the UiC1 register= 1 (reception enabled)
- The TE bit in the UiC1 register= 1 (transmission enabled)
- The TI bit in the UiC1 register= 0 (data present in the UiTB register)

22.9.2 UART

22.9.2.1 Special Mode 1(I²C Mode)

When generating start, stop and restart conditions, set the STSPSEL bit in the UiSMR4 register to “0” and wait for more than half cycle of the transfer clock before setting each condition generate bit (STAREQ, RSTAREQ and STPREQ) from “0” to “1”.

22.9.2.2 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting the U2C1 register U2IRS bit to “1” (transmission complete) and U2ERE bit to “1” (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to “0” (no interrupt request) after setting these bits.

22.10 A/D Converter

Set ADCON0 (except bit 6), ADCON1 and ADCON2 registers when A/D conversion is stopped (before a trigger occurs).

When the VCUT bit in the ADCON1 register is changed from “0” (Vref not connected) to “1” (Vref connected), start A/D conversion after passing 1 μs or longer.

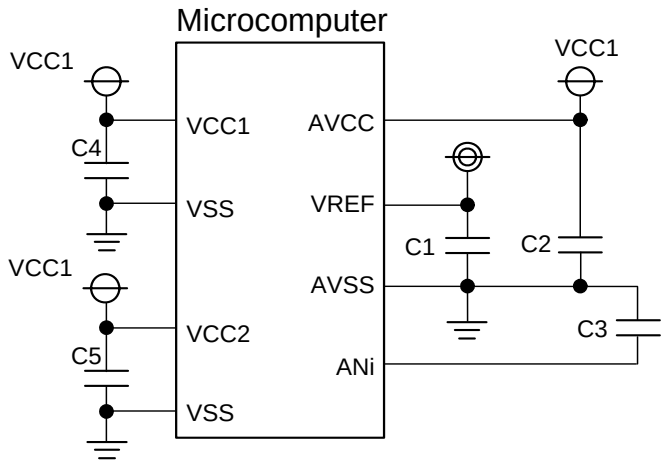
To prevent noise-induced device malfunction or latchup, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins (ANi(i=0 to 7), AN0_i) each and the AVSS pin. Similarly, insert a capacitor between the VCC1 pin and the VSS pin. Figure 22.3 is an example connection of each pin.

Make sure the port direction bits for those pins that are used as analog inputs are set to “0” (input mode). Also, if the TGR bit in the ADCON0 register = 1 (external trigger), make sure the port direction bit for the $\overline{\text{ADTRG}}$ pin is set to “0” (input mode).

When using key input interrupts, do not use any of the four AN4 to AN7 pins as analog inputs. (A key input interrupt request is generated when the A/D input voltage goes low.)

The ϕAD frequency must be 10MHz or less. Without sample-and-hold function, limit the ϕAD frequency to 250kHz or more. With the sample and hold function, limit the ϕAD frequency to 1MHz or more.

When changing an A/D operation mode, select analog input pin again in the CH2 to CH0 bits in the ADCON0 register.



ANi: ANi, AN0_i (i=0 to 7)

NOTES :
1. C1≥0.47μF, C2≥0.47μF, C3≥100pF, C4≥0.1μF, C5≥0.1μF (reference)
2. Use thick and shortest possible wiring to connect capacitors.

Figure 22.3 Use of Capacitors to Reduce Noise

If the CPU reads the ADi register (i = 0 to 7) at the same time the conversion result is stored in the ADi register after completion of A/D conversion, an incorrect value may be stored in the ADi register. This problem occurs when a divide-by-n clock derived from the main clock or a subclock is selected for CPU clock.

- When operating in one-shot mode
  Check to see that A/D conversion is completed before reading the target ADi register. (Check the IR bit in the ADIC register to see if A/D conversion is completed.)
- When operating in repeat mode
  Use the main clock for CPU clock directly without dividing it.

If A/D conversion is forcibly terminated while in progress by setting the ADST bit in the ADCON0 register to “0” (A/D conversion halted), the conversion result of the A/D converter is indeterminate. The contents of ADi registers irrelevant to A/D conversion may also become indeterminate. If while A/D conversion is underway the ADST bit is cleared to “0” in a program, ignore the values of all ADi registers.

The applied intermediate potential may cause more increase in power consumption than other analog input pins (AN0 to AN3 and AN0_0 to AN0_7), since the AN4 to AN7 are used with the KI0 to KI3.

22.11 Precautions for Programmable I/O Ports

The input threshold voltage of pins differs between programmable input/output ports and peripheral functions. Therefore, if any pin is shared by a programmable input/output port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions VIH and VIL (neither “high” nor “low”), the input level may be determined differently depending on which side-the programmable input/output port or the peripheral function-is currently selected.

22.12 Electric Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers

Flash memory version and mask ROM version may have different characteristics, operating margin, noise tolerated dose, noise width dose in electrical characteristics due to internal ROM, different layout pattern, etc. When switching to the mask ROM version, conduct equivalent tests as system evaluation tests conducted in the flush memory version.

22.13 Mask ROM

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

22.14 Flash Memory Version

22.14.1 Functions to Inhibit Rewriting Flash Memory Rewrite

ID codes are stored in addresses 0FFFDFh, 0FFFE3h, 0FFFEBh, 0FFFEFh, 0FFFF3h, 0FFFF7h, and 0FFFFBh. If wrong data are written to theses addresses, the flash memory cannot be read or written in standard serial I/O mode.
The ROMCP address is mapped in address 0FFFFFh. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.
In the flash memory version of MCU, these addresses are allocated to the vector addresses (H) of fixed vectors.

22.14.2 Stop mode

When the MCU enters stop mode, execute the instruction which sets the CM10 bit to “1” (stop mode) after setting the FMR01 bit to “0” (CPU rewrite mode disabled) and disabling the DMA transfer.

22.14.3 Wait mode

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

22.14.4 Low power dissipation mode

If the CM05 bit is set to “1” (main clock stop), the following commands must not be executed.

- Program
- Block erase
- Lock bit program
- Read Lock bit Status

22.14.5 Writing command and data

Write the command code and data at even addresses.

22.14.6 Program Command

Write “xx40h” in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

22.14.7 Lock Bit Program Command

Write “77h” in the first bus cycle and write “xxD0h” to the uppermost address of a block (even address, however) in the second bus cycle, and the lock bit for the specified block is cleared to “0”. Make sure then address value specified in the first bus cycle is the same uppermost block address that is specified in the second bus cycle.

22.14.8 Operation speed

Before entering CPU rewrite mode (EW0 or EW1 mode), set the CM11 bit in the CM1 register to “0” (main clock), select 10 MHz or less for CPU clock using the CM06 bit in the CM0 register and CM17 to CM16 bits in the CM1 register. Also, set the PM17 bit in the PM1 register to “1” (with wait state).

22.14.9 Prohibited instructions

The following instructions cannot be used in EW0 mode because they reference data in the flash memory: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction.

22.14.10 Interrupts

EW0 Mode

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated in the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts are available because the FMR0 and FMR1 registers are forcibly reset when either interrupt occurs. Allocate jump addresses for individual interrupt routine to the fixed vector table. Rewrite operation is aborted when the $\overline{\text{NMI}}$ or watchdog timer interrupt occurs. Execute a rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available because the CPU references data in the flash memory.

EW1 Mode

- Do not acknowledge any interrupts having vectors in the relocatable vector table or address match interrupt during auto-program or auto-erase operation.
- Do not use the watchdog timer interrupt.
- The $\overline{\text{NMI}}$ interrupt is available since registers FMR0 and FMR1 are forcibly reset when the interrupt occurs. Allocate a jump address for individual interrupt routine to the fixed vector table. Rewrite operation is aborted when the $\overline{\text{NMI}}$ interrupt occurs. Execute a rewrite program again after exiting the interrupt routine.

22.14.11 How to access

To set the FMR01, FMR02, or FMR11 bit to “1”, write “0” and then “1” in succession. This is necessary to ensure that no interrupts or DMA transfers will occur before writing “1” after writing “0”. Also only when $\overline{\text{NMI}}$ pin is “H” level.

22.14.12 Writing in the user ROM area

EW0 Mode

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

EW1 Mode

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

22.14.13 DMA transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR00 bit in the FMR0 register = 0 (during the auto program or auto erase period).

22.15 One Time Flash Version

22.15.1 Stop mode

When the MCU enters stop mode, execute the instruction which sets the CM10 bit to “1” (stop mode) after setting the FMR01 bit to “0” and disabling the DMA transfer.

22.15.2 Wait mode

When shifting to wait mode, set the FMR01 bit to “0” before executing the WAIT instruction.

22.15.3 Operation speed

Before the FMR01 bit is set to “1” , set the CM11 bit in the CM1 register to “0” (main clock), select 10 MHz or less for CPU clock using the CM06 bit in the CM0 register and CM17 to CM16 bits in the CM1 register. Also, set the PM17 bit in the PM1 register to “1” (with wait state).

22.15.4 Prohibited Instructions

The following instructions cannot be used when the FMR01 bit is set to “1” because they reference data in the flash memory: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction.

22.15.5 Interrupts

When the FMR01 bit is set to “1”,

- To use interrupts having vectors in relocatable vector table, the vectors must be relocated in the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts are available because the FMR0 and FMR1 registers are forcibly reset when either interrupt occurs. Allocate jump addresses for individual interrupt routine to fixed vector table. When the $\overline{\text{NMI}}$ or watchdog timer interrupt occurs, wait until the interrupt routine is completed and then set the FMR01 bit to “1” in order to set the FMSTP bit in the FMR0 register to “1” again.
- The address match interrupt is not available because the CPU references data in the flash memory.

22.15.6 How to access

Set the FMR01 bit to 1 immediately after setting them first to 0 while a high-level (“H”) signal is applied to the $\overline{\text{NMI}}$ pin. Do not generate an interrupt or a DMA transfer between setting the FMR01 bit to 0 and setting them to 1.

22.16 Precautions for Noise

Connect a bypass capacitor (approximately 0.1 μF) across the VCC1 and XSS pins, and VCC2 and VSS pins using the shortest and thicker possible wiring. Figure 22.4 shows the Bypass Capacitor Connection.

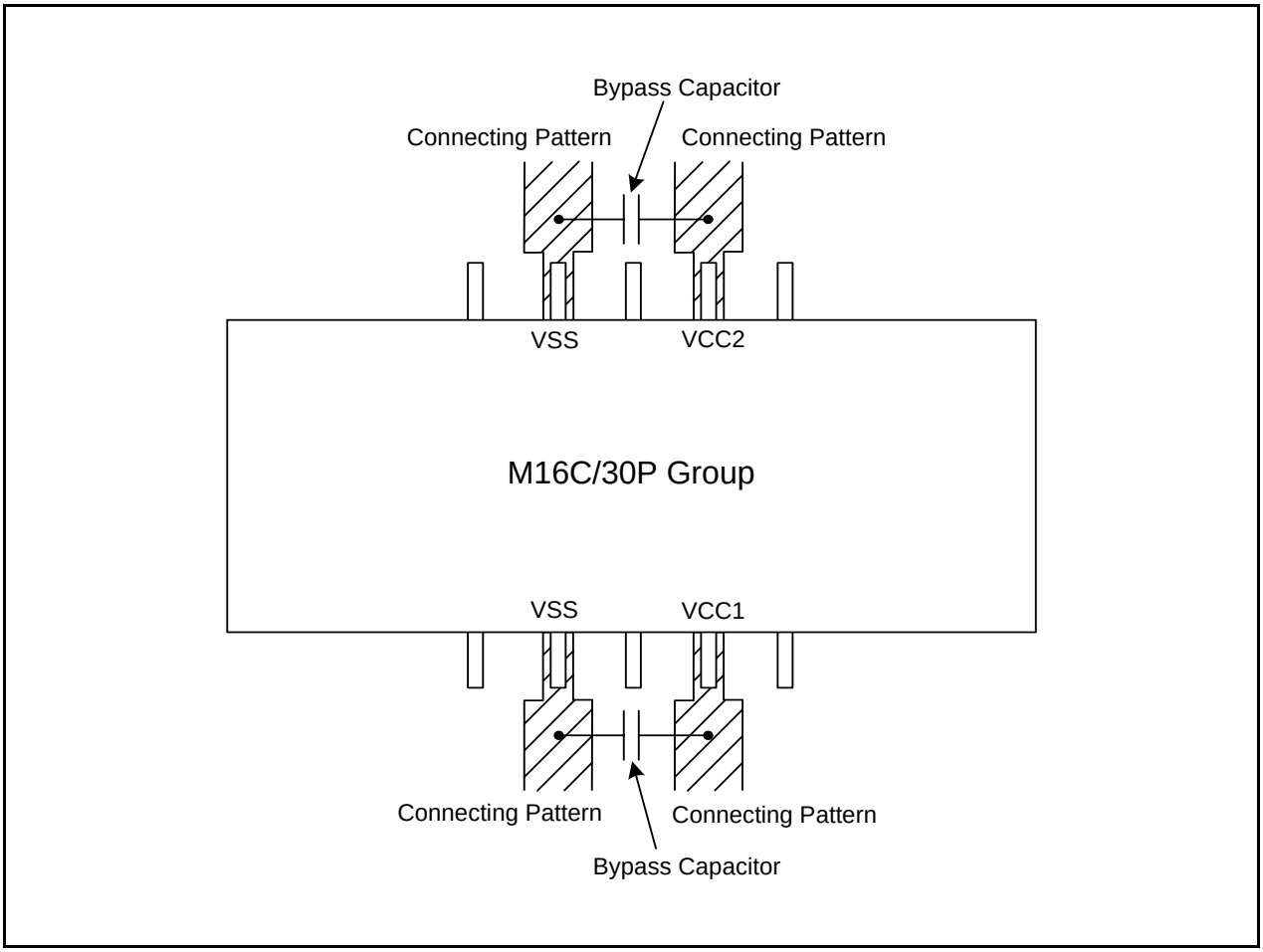
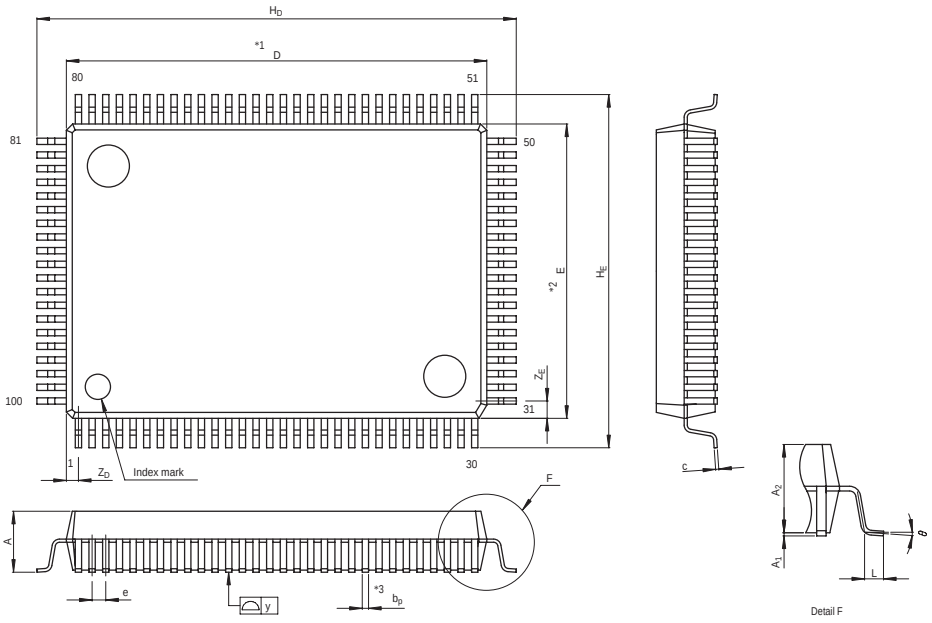


Figure 22.4 Bypass Capacitor Connection

Appendix 1. Package Dimensions

Diagrams showing the latest package dimensions and mounting information are available in the "Packages" section of the Renesas Technology website.

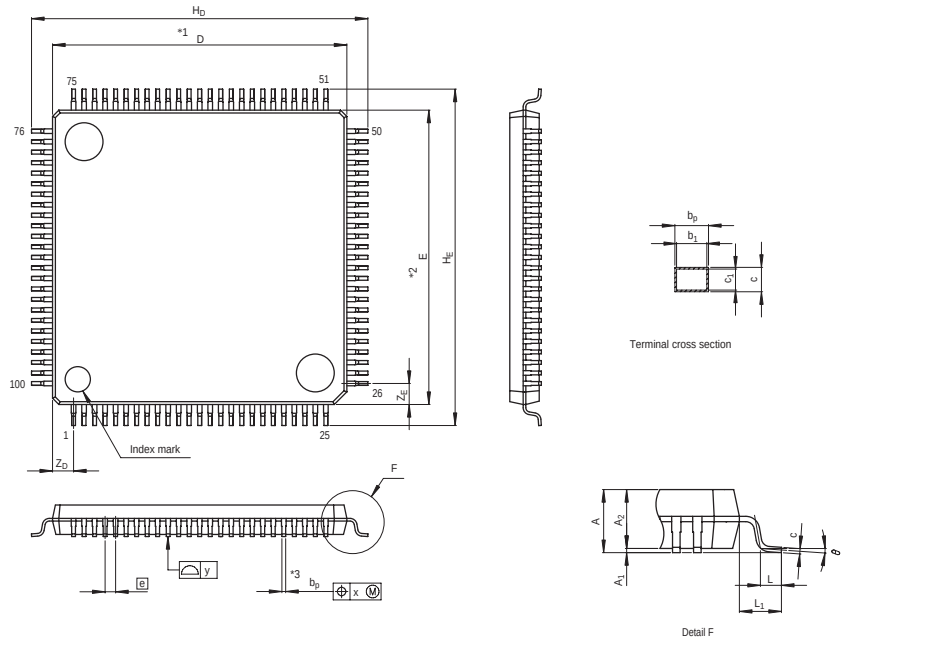
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-QFP100-14x20-0.65	PRQP0100JB-A	100P6S-A	1.6g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.8	20.0	20.2
E	13.8	14.0	14.2
A_2	—	2.8	—
H_D	22.5	22.8	23.1
H_E	16.5	16.8	17.1
A	—	—	3.05
A_1	0	0.1	0.2
b_p	0.25	0.3	0.4
c	0.13	0.15	0.2
θ	0°	—	10°
e	0.5	0.65	0.8
y	—	—	0.10
Z_D	—	0.575	—
Z_E	—	0.825	—
L	0.4	0.6	0.8

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Appendix 2. Difference between M16C/62P and M16C/30P

Appendix Table 2.1 Function Difference (1)(1)

Item	M16C/62P	M16C/30P
Minimum instruction Execution Time	41.7ns(f(BCLK)=24MHz, VCC1=3.0 to 5.5V) 100ns(f(BCLK)=10MHz, VCC1=2.7 to 5.5V)	62.5ns(f(XIN)=16MHz, VCC=3.0 to 5.5V) 100ns(f(XIN)=10MHz, VCC=2.7 to 5.5V)
Supply Voltage	VCC1=3.0 to 5.5V, VCC2=3.0V to VCC1 (f(BCLK)=24MHz) VCC1=VCC2=2.7 to 5.5V (f(BCLK)=10MHz)	VCC1=VCC2=3.0 to 5.5V(f(XIN)=16MHz) VCC1=VCC2=2.7 to 5.5V(f(XIN)=10MHz)
I/O Power Supply	Double (VCC1, VCC2)	Single (VCC1=VCC2)
Package	80-pin, 100-pin, 128-pin plastic mold QFP	100-pin plastic mold QFP
Memory	Mask ROM Flash Memory ROMless	Mask ROM Flash Memory One time Flash Memory ROMless
Voltage Detection Circuit	Built-in Vdet3, Vdet4 detect Voltage down detect interrupt Voltage down detect reset (hardware reset 2)	None
Clock Generating Circuit	PLL, XIN, XCIN, On-chip oscillator	XIN, XCIN
System Clock Protective Function	Built-in	None (protected by protect register)
Oscillation Stop, Re-oscillation Detection Function	Built-in	None
Power Consumption	18mA(VCC1=VCC2=5V, f(BCLK)=24MHz) 8mA(VCC1=VCC2=3V, f(BCLK)=10MHz) 1.8μA(VCC1=VCC2=3V, f(XCIN)=32kHz, wait mode)	10mA(VCC=5V, f(XIN)=16MHz) 8mA(VCC=3V, f(XIN)=10MHz) 1.8μA(VCC=3V, f(XCIN)=32kHz, wait mode)
Memory Area	Memory area expandable (4M bytes)	1 M bytes fixed
External Device Connect Area	04000h to 07FFFh (PM13=0) 08000h to 0FFFFh(PM10=0) 10000h to 26FFFh 28000h to 7FFFFh 80000h to CFFFFh (PM13=0) D0000h to FFFFFh (Microprocessor mode)	04000h to 07FFFh 08000h to 0FFFFh (PM10=0) 10000h to 26FFFh 28000h to 7FFFFh 80000h to CFFFFh (PM13=0 or without the PM13 bit) D0000h to FFFFFh (Microprocessor mode)
Bus Mode	Separate bus Multiplexed bus	Separate bus

NOTES:
1. About the details and the characteristics, refer to Hardware Manual.

Appendix Table 2.2 Function Difference (2)(1)

Item	M16C/62P	M16C/30P
Upper Address Memory Expansion Mode and Microprocessor mode	P4_0 to P4_3(A16 to A19), P3_4 to P3_7(A12 to A15) : Switchable between address bus and I/O port	P4_0 to P4_3(A16 to A19) : Switchable between address bus and I/O port P3_4 to P3_7(A12 to A15) : Can not be switched
Access to SFR	Variable (1 to 2 waits)	1 wait fixed
Software Wait to External Area	Variable (0 to 3 waits)	Variable (0 to 1 wait)
Protect	Can be set for PM0, PM1, PM2, CM0, CM1, CM2, PLC0, INVC0, INVC1, PD9, S3C, S4C, TB2SC, PCLKR, VCR2, D4INT registers	Can be set for PM0, PM1, CM0, CM1, PD9 registers
Watchdog Timer	Watchdog timer interrupt or watchdog timer reset is selected Count source protective mode is available	Watchdog timer interrupt No count source protective mode
$\overline{\text{INT}}$ Interrupt	6 ($\overline{\text{INT0}}$ to $\overline{\text{INT5}}$)	5 ($\overline{\text{INT0}}$ to $\overline{\text{INT4}}$)
Address Match Interrupt	4	2
Multifunction Timer	11 channels Timer A x 5 channels, Timer B x 6 channels	6 channels Timer A x 3 channels, Timer B x 3 channels
Timer A two-phase pulse signal processing	Function Z-phase (counter reset) input	No function Z-phase (counter reset) input
Timer Functions for Three-phase Motor Control	Built-in	None
Serial Interface (UART0 to UART2)	(UART, Clock synchronous, I2C bus™ (2), IEBus™ (3))x 3	(UART, Clock synchronous, I2C bus™ (2)) x 2 (UART, Clock synchronous, I2C bus™ (2), IEBus™ (3)) x 1
Clock Synchronous Serial I/O (SI/O3, SI/O4)	2 channels	None
A/D Converter	10 bits x 8 channels Expandable up to 26 channels	10 bits x 8 channels Expandable up to 18 channels
A/D Converter Operation Mode	One-shot mode, Repeat mode, Single sweep mode, Repeat sweep mode 0, Repeat sweep mode 1 With External Op-amp mode	One-shot mode, Repeat mode Without External Op-amp mode
A/D Converter Input Pin	Select from ports P0, P2, P10	Select from ports P0, P10
D/A Converter	8 bits x 2 channels	None
Forced erase function of Flash Memory	None	Built-in

NOTES:
1. About the details and the characteristics, refer to Hardware Manual.
2. I2C bus is a trademark of Koninklijke Philips Electronics N. V.
3. IEBus is a trademark of NEC Electronics Corporation.

Register Index

REVISION HISTORY	M16C/30P Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.70	Aug 26, 2004	–	First Edition issued
0.80	Mar 18, 2005	–	development support tools -> development tools
		–	BCLK -> CPU clock
		2	Table 1.1 Performance Outline of M16C/30P Group Serial interface is revised.
		4	Figure 1.2 Type., Memory Size, and Package is partly revised.
		8	Table 1.4 Pin Detection (2) is partly revised.
		18	5.1 Hardware Reset 1 -> 5.1 Hardware Reset
		21	6.1 Types of Processor Modes is revised.
		25	Figure 7.1 Clock Generation Circuit is partly revised.
		26	Note 3 of Figure 7.2 CM0 Register is revised.
		30	Tittle of 7.2.1 CPU Clock and BCLK is revised.
		31	7.4.1 Normal Operation Mode is partly revised.
		33	Table 7.4 Interrupts to Exit Wait Mode is partly revised.
		57	Figure 10.1 Watchdog Timer Block Diagram is partly revised.
		59	Figure 10.3 Typical Operation of Cold start / Warm start is revised.
		63	Figure 11.3 DM1SL Register is partly revised.
		71	Figure 12.1 Timer A Configuration is revised.
		73	Figure 12.3 Timer A Block Diagram is revised.
		83	Table 12.4 Specifications in One-shot Timer Mode is revised.
		85	Table 12.5 Specifications in PWM Mode is revised.
		88	Figure 12.14 Timer B Block Diagram is partly revised.
		92	Table 12.7 Specifications in Event Counter Mode is partly revised.
		93	Figure 12.18 TBiMR Register in Event counter Mode is partly revised.
		95	Figure 12.19 TBiMR Register in Pulse Period and Pulse Width Measurement Mode is partly revised.
		96	Figure 12.20 Operation Timing when Measuring a Pulse Period is partly revised.
			Figure 12.21 Operation Timing when Measuring a Pulse Width is partly revised.
		108	Table 13.1 Clock Synchronous Serial I/O Mode Specifications is partly revised.
		111	Figure 13.12 Transmit and Receive Operation is partly revised.
		112	13.1.1.1 Counter Measure for Communication Error Occurs is partly revised.
		120	13.1.2.1 Bit Rate is partly revised.
			Table 13.9 Example of Rates and Settings is revised.
		124	Table 13.10 I^2C Mode Specifications is partly revised.
		134	Table 13.15 Special Mode 2 Specifications is partly revised.
		141	Table 13.18 SIM Mode Specifications is partly revised.

Rev.	Date	Description	
		Page	Summary
		146	Table 14.1 Performance of A/D Converter is revised.
		148	ADCON1 Register of Figure 14.2 ADCON0 to ADCON1 Registers is partly revised.
		149	ADCON2 Register of Figure 14.3 ADCON2 and AD0 to AD7 Registers is partly revised.
		152	Figure 14.5 ADCON1 Register (One-shot Mode) is partly revised.
		155	Figure 14.7 ADCON1 Register (Repeat Mode) is partly revised.
		158	Figure 14.8 Analog Input Pin and External Sensor Equivalent Circuit is partly revised.
		167	Figure 16.7 PDi Registers is partly revised.
		174	Note 2 Table 17.3 A/D Conversion Characteristics is partly revised.
		175	Symbol of Table 17.4 Power Supply Circuit Timing Characteristics is partly revised.
		176	Table 17.5 Electrical Characteristics is revised.
		182	Table 17.19 Electrical Characteristics is revised.
		189	A/D converter of 18.2 Precautions for Power Control is partly revised.
		191	Note 2 of Figure 18.2 Procedure for Changing the Interrupt Generate Factor is partly revised.
		199	18.8 Precautions for A/D Converter is partly revised.
		200	18.8 Precautions for A/D Converter is partly revised.
		203-204	Appendix 2. Difference between M16C/62P and M16C/30P is added.
1.00	Sep 01, 2005	2	Table 1.1 Performance Outline of M16C/30P Group is partly revised.
		4	Table 1.2 Product List is partly revised.
			Figure 1.2 Type No., Memory Size, and Package is partly revised.
		5	Figure 1.3 Pin Configuration is partly revised.
		6	Figure 1.4 Pin Configuration is partly revised.
		7-8	Tables 1.3 to 1.4 Pin Characteristics are added.
		9	Table 1.5 Pin Description is revised.
		14	3. Memory is partly revised.
		15	Table 4.1 SFR Information is partly revised.
		19	Table 4.5 SFR Information is partly revised
		20-23	Change Sections in Chapter 5.
		21	Figure 5.2 Reset Sequence is revised.
		22	Table 5.1 Pin Status When RESET Pin Level is “L” is revised.
		25-26	5.4 Cold Start-up / Warm Start-up Determine Function is added.
		27-30	6. Processor Mode is revised.
		31-39	7. Bus is Added.
		40	8. Memory Space Expansion Function is added.
		45	Figure 9.5 Example of Main Clock Connection Circuit is partly revised.

Rev.	Date	Description	
		Page	Summary
		46	Figure 9.6 Example of Sub Clock Connection Circuit is partly revised.
		49	Table 9.3 Pin Status During Wait Mode is partly revised.
		50	Table 9.4 Interrupts to Exit Wait Mode and Use Conditions is partly revised.
		51	9.4.3 Stop Mode is partly revised.
			Table 9.5 Interrupts to Exit Stop Mode and Use Conditions id added.
		76	12.1 Cold Start / Warm Start moved to 5. Reset.
		78	Table 13.1 DMAC Specifications is partly revised.
		83	13.1.2 Effect of BYTE Pin Level is added.
		85	Table 13.2 DMA Transfer Cycles is partly revised.
			Table 13.3 Coefficient J, k is partly revised.
		115	Figure 15.1 UART0 Block Diagram is partly revised.
		116	Figure 15.2 UART1 Block Diagram is partly revised.
		117	Figure 15.3 UART2 Block Diagram is partly revised.
		119	Note 3 is added in Figure 15.5 UiRB Register.
		126	Note 2 is partly revised in Table 15.1 Clock Synchronous Serial I/O Mode Specifications.
		129	Figure 15.12 Transmit and Receive Operation is revised.
		134	Note 1 is partly revised in Table 15.5 UART Mode Specifications.
		137	Figure 15.18 Transmit Operation is revised.
		138	Table 15.9 Example of Bit Rates and Settings is partly revised.
		144	Note 4 is added in table 15.11 Registers to Be Used and Settings in I^2C Mode.
		161	Figure 15.33 Transmit and Receive Timing in SIM Mode is revised.
		163	15.1.6.2 Format is revised.
		178	Figure 17.3 CRC Calculation is partly revised.
		179	18.1 Port Pi Direction Register is partly revised.
			18.2 Port Pi Register is partly revised.
			18.3 Pull-up Control Register 0 to Pull-up Control Register 2 is partly revised.
		184	Figure 18.6 I/O Pins is partly revised.
		185	Note 2 is added in Figure 18.7 PDi Registers.
		186	Note 2 is added in Figure 18.8 Pi Registers.
		187	Note 2 is added in Figure 18.9 PUR0 Register.
			Note 3 to 5 are added in Figure 18.9 PUR1 Register.
		190	Table 18.2 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode is added.
		191	Figure 18.11 Unassigned Pins Handling is revised.
		193	Table 19.2 Recommended Operating Conditions is partly revised.
		194	Table 19.3 A/D Conversion Characteristics is partly revised.

Rev.	Date	Description	
		Page	Summary
		197	Note 1 is added in Table 19.6 External Clock Input (XIN input)
			Table 19.7 Memory Expansion Mode and Microprocessor Mode is added.
		200	Table 19.20 Memory Expansion Mode and Microprocessor Modes (for setting with no wait) is added.
			Figure 19.2 Ports P0 to P10 Measurement Circuit is added.
		201	Table 19.21 Memory Expansion Mode and Microprocessor Modes (for 1- to 3-wait setting and external area access) is added.
		204	Figure 19.5 Timing Diagram (3) is added.
		205	Figure 19.6 Timing Diagram (4) is added.
		206	Figure 19.7 Timing Diagram (5) is added.
		208	Note 1 to 4 are added in Table 19.23 External Clock Input (XIN input)
			Table 19.24 Memory Expansion Mode and Microprocessor Mode is added.
		211	Table 19.37 Memory Expansion Mode and Microprocessor Modes (for setting with no wait) is added.
			Figure 19.8 Ports P0 to P10 Measurement Circuit is added.
		212	Table 19.38 Memory Expansion Mode and Microprocessor Modes (for 1- to 3-wait setting and external area access) is added.
		215	Figure 19.11 Timing Diagram (3) is added.
		216	Figure 19.12 Timing Diagram (4) is added.
		217	Figure 19.13 Timing Diagram (5) is added.
		219	20.2 Bus is added.
		220	20.3 Precautions for Power Control is revised.
		231	Figure 20.3 Use of Capacitors to Reduce Noise is partly revised.
		232	20.8 Precautions for A/D Converter is partly revised.
		235-236	Appendix Table 2.1 to 2.2 are partly revised.

Rev.	Date	Description	
		Page	Summary
1.10	Oct 01, 2005	2	Table 1.1 Performance Outline of M16C/30P Group is partly revised.
		4	Table 1.2 Product List is partly revised.
			Figure 1.2 Type No., Memory Size, and Package is partly revised.
		5	Table 1.3 Product Code of Mask ROM version Version for M16C/30P is added.
			Figure 1.3 Marking Diagram of Mask ROM Version for M16C/30P is added.
		6	Figure 1.4 Marking Diagram of ROM -less Version for M16C/30P is added.
		6	Table 1.4 Product Code of ROM-less version for M16C/30P is added.
		16	Figure 3.1 Memory Map is partly added.
		32	Figure 6.3 Memory Map is partly added.
		54	9.4.3.3 Exiting Stop Mode is partly revised.
		85	13.1 Transfer Cycles information is added.
		99	14.1.2 Event Counter Mode is partly revised.
		101	Information is added
		123	Note 5 is added in Figure 15.7 UiC0 Register.
		194	Table 19.2 information is revised.
		236	Appendix Table 2.1 Function Difference Memory is partly added.
1.11	May 31, 2006	1	A note is add in Chapter 1. Overview.
		2	Table 1.1 Performance Outline of M16C/30P Group is partly revised.
		4	1.4 Product List information is added. Table 1.2 Product List is partly revised.
		5	Figure 1.2 Type No., Memory Size, and Package is added.
		7	Table 1.4 Product Code of Flash Memory version and ROM-less version for M16C/30P is added.
			Figure 1.4 Marking Diagram of Flash Memory version and ROM-less Version for M16C/30P (Top View) is partly added.
		17	3. Memory information is revised. Figure 3.1 Memory Map is partly revised.
		18	Table 4.1 SFR Information(1) is partly revised.
		19	Table 4.2 SFR Information(2) is partly added.
		23	5.1 Hardware Reset information is deleted.
		29	Table 6.1 Features of Processor Modes is partly deleted.
		31	Figure 6.2 PM1 Register is partly revised.
		32	Figure 6.3 Memory Map in Single Chip Mode is partly added.
		39	Table 7.5 Pin Functions for Each Processor Mode NOTES is partly deleted.
		42	Figure 8.1 Memory Mapping and CS Area in 1-Mbyte mode is partly revised.
		62	11.4.1 Fixed Vector Tables Information is added.
		63	Table 11.2 Relocatable Vector Tables is partly added.
		65	Figure 11.4 Interrupt Control Registers (2) NOTES 4 is added.

Rev.	Date	Description	
		Page	Summary
		73	Figure 11.11 IFSR and IFSR2A Registers NOTES is added.
		75	11.9 Address Match Interrupt information is added.
		77	Figure 12.1 Watchdog Timer Block Diagram is partly revised.
		85	13.1 Transfer Cycles is entirely revised.
		89	13.5 Channel Priority and DMA Transfer Timing is partly added.
		90	Figure 14.1 Timer A Configuration is partly deleted.
		98	Figure 14.8 TAiMR Register in Timer Mode is partly added.
		104	Figure 14.11 TAiMR Register in One-Shot Timer Mode is partly added.
		106	Figure 14.12 TAiMR Register in PWM Mode is partly added.
		111	Figure 14.18 TBiMR Register in Timer Mode is partly added.
		115	Figure 14.20 TBiMR Register in Pulse Period and Pulse Width Measurement Mode is partly added.
		128	Table 15.1 Clock Synchronous Serial I/O Mode Specifications is partly added.
		136	Table 15.5 UART Mode Specifications is partly added.
		145	Figure 15.24 I^2C Mode Block Diagram is partly revised.
		147	Table 15.12 Registers to Be Used and Settings in I2C Mode (2) is partly added.
		154	Table 15.15 Special Mode 2 Specifications is partly added.
		156	Table 15.16 Registers to Be Used and Settings in Special Mode 2 is partly revised.
		161	Table 15.18 SIM Mode Specifications is partly added.
		180	18. Programmable I/O Ports Information is revised.
		190	Table 18.1 Unassigned Pin Handling in Single-chip Mode is partly revised.
		191	Table 18.2 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode is partly revised.
		192	Figure 18.11 Unassigned Pins Handling is partly revised.
		193	19. Flash Memory Version is added.
		208	Table 19.4 Software Commands NOTES: 1 is partly revised.
		211	19.3.5.6 Erase All Unlocked Block is partly revised.
		219	Figure 19.14 Pin Connections for Serial I/O Mode (1)
		220	Figure 19.15 Pin Connections for Serial I/O Mode (2) is partly revised.
		224	Table 20.1 Absolute Maximum Ratings is partly revised.
		225	Table 20.2 Recommended Operating Conditions is partly revised.
		227	Table 20.4 Flash Memory Version Electrical Characteristic sum Ratings and Table 20.5 Flash Memory Version Program / Erase Voltage and Read Operation Voltage Characteristics is added.
		229	Table 20.7 Electrical Characteristics (1) is partly revised.
		230	Table 20.8 Electrical Characteristics (2) is partly revised.
		234	Table 20.23 Memory Expansion and Microprocessor Modes (for setting with no wait) is partly revised.
		235	Table 20.24 Memory Expansion and Microprocessor Modes (for 1 wait setting and external area access) is partly revised.

Rev.	Date	Description	
		Page	Summary
		241	Table 20.25 Electrical Characteristics (1) is partly revised.
		242	Table 20.26 Electrical Characteristics (2) is partly revised.
		246	Table 20.41 Memory Expansion and Microprocessor Modes (for setting with no wait) is partly revised.
		247	Table 20.42 Memory Expansion and Microprocessor Modes (for 1 wait setting and external area access) is partly revised.
		253	21.1 SFR, 21.1.1 Register Settings, and Table 21.1 Registers with Write-only Bits is added.
		255	21.3 Bus information is added.
		267	Figure 21.3 Use of Capacitors to Reduce Noise is partly revised.
		269	21.12 Electric Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers information is added.
		270-272	21.14 Flash Memory Version information is added.
1.21	Jan 24, 2007	All chapters	Added the contents relevant to one-time flash to body texts, tables, and diagrams to all chapters. Deleted Erase all unlock block command in all chapters. Type No. Æ Part No.
		2	Table 1.1 Performance Outline of M16C/30P Group is partly revised.
		4	Table 1.2 Product List is partly revised.
		17	Figure 3.1 Memory Map is partly revised.
		32	Figure 6.3 PM1 Register (2) (M30304GDPFP, M30304GDPGP, M30304GEPFP, M30304GEPGP, M30302GGPFP, M30302GGPGP) is partly revised.
		33	Figure 6.4 Memory Map in Single Chip Mode is partly revised.
		43	8. Memory Space Expansion Function Figure 8.1 Memory Mapping and CS Area in 1-Mbyte mode (PM13=0) is partly revised.
		44	Figure 8.2 Memory Mapping and CS Area in 1-Mbyte mode (PM13=1) is partly revised.
		130	Table 15.1 Clock Synchronous Serial I/O Mode Specifications is partly revised.
		138	Table 15.5 UART Mode Specifications is partly revised.
		146	Table 15.10 I^2C Mode Specifications is partly revised.
		148	Table 15.11 Registers to Be Used and Settings in I2C Mode (1) is partly revised.
		156	Table 15.15 Special Mode 2 Specifications is partly revised.
		163	Table 15.18 SIM Mode Specifications is partly revised.
		197	Figure 19.1 Flash Memory Block Diagram
		199	19.2.2 ID Code Check Function, Table 19.3 Reserved Character Sequence (Reserved Word) is added.
		200	19.2.3 Forced Erase Function, Table 19.4 Forced Erase Function is added.
		229	Chapter 20. One Time Flash Version is added.
		241	Table 21.4 Flash Memory Version Electrical Characteristics (1) is partly revised.
		285	22.14.4 Low power dissipation mode is partly added.
		290	Appendix Table 2.1 Function Difference (1) is partly added.
		291	Appendix Table 2.2 Function Difference (2) is partly added.

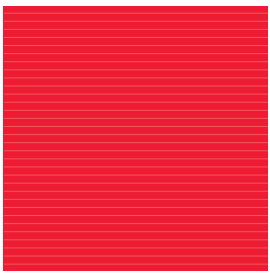
Rev.	Date	Description	
		Page	Summary
1.22	Mar 29, 2007	4	Table 1.2 Product List (1) is partly revised.
		5	Table 1.3 Product List (2) is partly revised.
		19	Table 4.2 SFR Information (2) is partly revised.
		31	Figure 6.2 PM1 Register (1) is partly revised.
		32	Figure 6.3 PM1 Register (2) (M30304GDPFP, M30304GDPGP, M30304GEPFP, M30304GEPGP, M30302GGPFP, M30302GGPGP) is partly deleted.
		195	Table 19.1 Flash Memory Version Specifications is partly deleted. Table 19.2 Flash Memory Rewrite Modes Overview is partly revised.
		201	Table 19.5 EW0 Mode and EW1 Mode is partly revised.
		203	Figure 19.4 FMR0 Register is partly added.
		205	19.3.3.4 FMSTP Bit is partly added.
		228	20. One Time Flash Version is partly revised. Table 20.1 One Time Flash Memory Version Specifications is partly added. Table 20.2 One Time Flash Memory Rewrite Modes Overview is partly added.

M16C/30P Group Hardware Manual

Publication Date : Rev.1.22 Mar 29, 2007

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

M16C/30P Group
Hardware Manual



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