

## Overview

The M16C/62P group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core and are packaged in a 100-pin and 128-pin plastic molded QFP. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 1M bytes of address space, they are capable of executing instructions at high speed. In addition, this microcomputer contains a multiplier and DMAC which combined with fast instruction processing capability, makes it suitable for control of various OA, communication, and industrial equipment which requires high-speed arithmetic/logic operations.

## Applications

Audio, cameras, office/communications/portable/industrial equipment, etc

## -----Table of Contents-----

Overview .....	1	Serial I/O .....	139
Central Processing Unit (CPU) .....	12	Clock Synchronous serial I/O Mode .....	148
Special Function Registers (SFR) .....	14	UART Mode .....	155
Reset .....	20	Special Mode 1 (I <sup>2</sup> C mode) .....	162
Processor Mode .....	29	Special Mode 2 .....	172
Clock Generation Circuit .....	51	Special Mode 3 (IE mode) .....	177
Protection .....	74	Special Mode 4 (SIM mode) (UART2) .....	179
Interrupts .....	75	SI/O3 and SI/O4 .....	184
Watchdog Timer .....	95	A-D Converter .....	189
DMAC .....	97	D-A Converter .....	206
Timers .....	107	CRC Calculation .....	208
Timer A .....	109	Programmable I/O Ports .....	210
Timer B .....	123	Electrical Characteristics .....	223
Three-phase Motor Control Timer Function	129	Flash Memory .....	261

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

## Performance Outline

Table 1.1.1 lists performance outline of M16C/62P group.

**Table 1.1.1. Performance outline of M16C/62P group**

Item		Performance	
Number of basic instructions		91 instructions	
Shortest instruction execution time		41.7 ns (f(BCLK)= 24MHz, Vcc1= 3.0V to 5.5V) 100 ns (f(BCLK)= 10MHz, Vcc1= 2.7V to 5.5V)	
Memory capacity	ROM	(See the product list)	
	RAM	(See the product list)	
I/O port	100-pin version P0 to P10 (except P85)	8 bits x 10, 7 bits x 1	P0 to P5: Vcc2 ports P6 to P10: Vcc1 ports
	128-pin version P0 to P14 (except P85)	8 bits x 13, 7 bits x 1, 2 bits x 1	P0 to P5, P12, P13: Vcc2 ports P6 to P10, P11, P14: Vcc1 ports
Input port	P85	1 bit x 1 (NMI pin level judgment): Vcc1 ports	
Multifunction timer			
Output		16 bits x 5 channels (TA0, TA1, TA2, TA3, TA40)	
Input		16 bits x 6 channels (TB0, TB1, TB2, TB3, TB4, TB5)	
Serial I/O		3 channels (UART0, UART1, UART2) UART, clock synchronous, I <sup>2</sup> C bus <sup>1</sup> (option <sup>3</sup> ), or IE bus <sup>2</sup> (option <sup>3</sup> ) 2 channels (SI/O3, SI/O4) Clock synchronous	
A-D converter		10 bits x (8 x 3 + 2) channels	
D-A converter		8 bits x 2	
DMAC		2 channels (trigger: 25 sources)	
CRC calculation circuit		CRC-CCITT	
Watchdog timer		15 bits x 1 (with prescaler)	
Interrupt		25 internal and 8 external sources, 4 software sources, 7 levels	
Clock generation circuit		4 circuits <ul style="list-style-type: none"> <li>• Main clock</li> <li>• Sub-clock</li> <li>• Ring oscillator (for main-clock oscillation stop detect function)</li> <li>• PLL frequency synthesizer</li> </ul> (These circuits contain a built-in feedback resistor and external ceramic/quartz oscillator)	
Voltage detection circuit		Present (option <sup>3</sup> )	
Power supply voltage		Vcc1=3.0V to 5.5V, Vcc2=3.0V to Vcc1(f(BCLK)=24MHz) Vcc1=Vcc2=2.7V to 5.5V (f(BCLK)=10MHz)	
Flash memory	Program/erase voltage	3.3V ± 0.3V or 5.0V ± 0.5V	
	Number of program/erase	100 times	
Power consumption		14mA (Vcc1=Vcc2=5V, f(BCLK)=24MHz) 8mA (Vcc1=Vcc2=3V, f(BCLK)=10MHz) 1.8μA (Vcc1=Vcc2=3V, f(XCIN)=32kHz, when wait mode)	
I/O characteristics	I/O withstand voltage	5.0V	
	Output current	5mA	
Memory expansion		Available (to 4M bytes)	
Operating ambient temperature		-20 to 85°C -40 to 85°C (option <sup>3</sup> )	
Device configuration		CMOS high performance silicon gate	
Package		100-pin and 128-pin plastic mold QFP	

Notes:

1. I<sup>2</sup>C Bus is a registered trademark of PHILIPS.
2. IE Bus is a registered trademark of NEC.
3. If you desire this option, please so specify.

## Block Diagram

Figure 1.1.1 is a block diagram of the M16C/62P group.

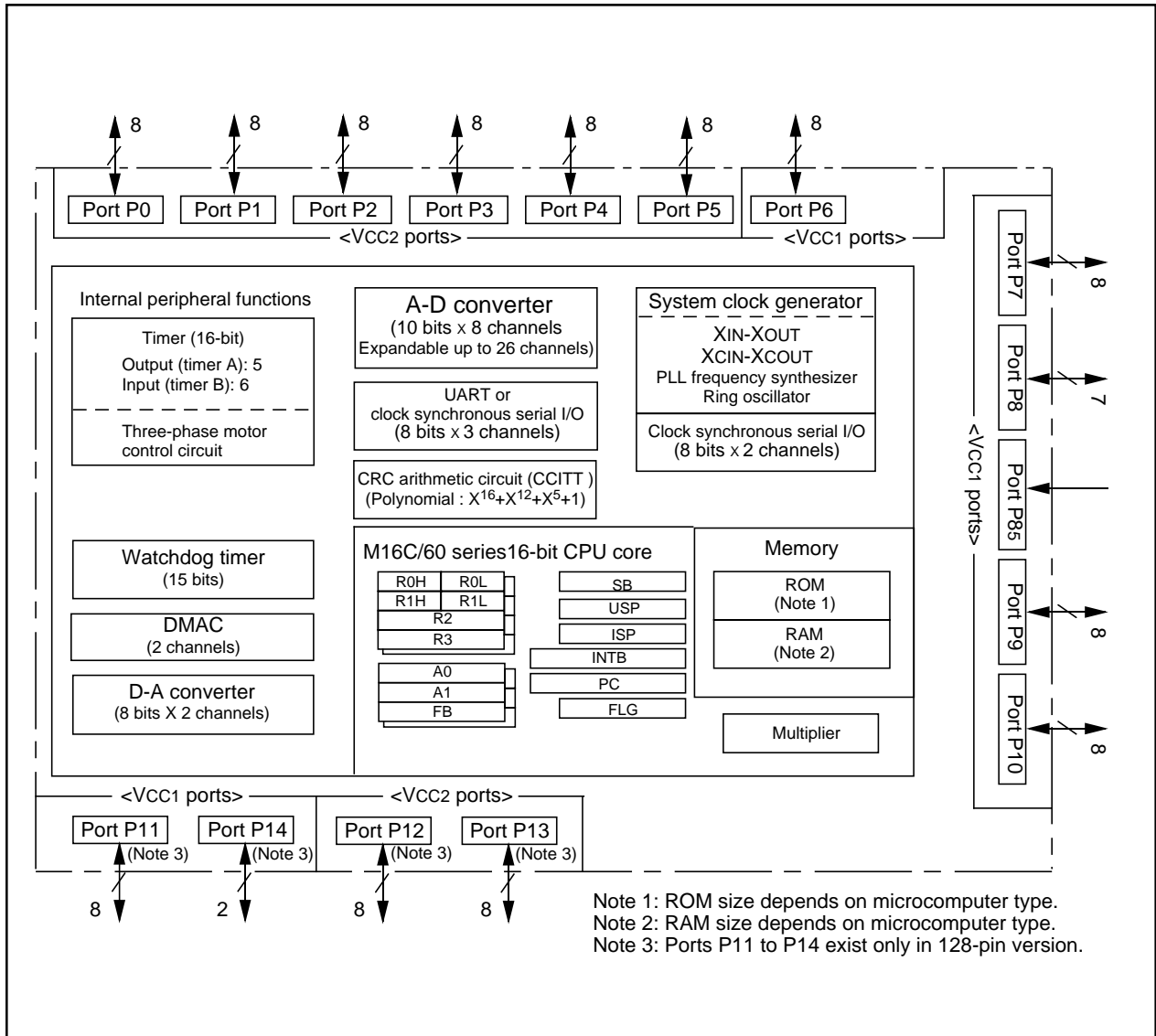


Figure 1.1.1. Block Diagram

## Overview

## Product List

Tables 1.1.2 and 1.1.3 list the M16C/62P group products and Figure 1.1.2 shows the type numbers, memory sizes and packages.

Table 1.1.2. Product List (1)

As of January 2003

Type No.	ROM capacity	RAM capacity	Package type	Remarks
M30622M6P-XXXFP **	48K bytes	4K bytes	100P6S-A	MASK ROM version
M30622M6P-XXXGP **			100P6Q-A	
M30622M8P-XXXFP **	64K bytes	4K bytes	100P6S-A	
M30622M8P-XXXGP **			100P6Q-A	
M30622MAP-XXXFP **	96K bytes	5K bytes	100P6S-A	
M30622MAP-XXXGP **			100P6Q-A	
M30620MCP-XXXFP **	128K bytes	10K bytes	100P6S-A	
M30620MCP-XXXGP **			100P6Q-A	
M30622MEP-XXXFP **	192K bytes	12K bytes	100P6S-A	
M30622MEP-XXXGP **			100P6Q-A	
M30623MEP-XXXGP **			128P6Q-A	
M30622MGP-XXXFP **	256K bytes	12K bytes	100P6S-A	
M30622MGP-XXXGP **			100P6Q-A	
M30623MGP-XXXGP **			128P6Q-A	
M30624MGP-XXXFP **		20K bytes	100P6S-A	
M30624MGP-XXXGP **			100P6Q-A	
M30625MGP-XXXGP **			128P6Q-A	
M30622MWP-XXXFP **	320K bytes	16K bytes	100P6S-A	
M30622MWP-XXXGP **			100P6Q-A	
M30623MWP-XXXGP **			128P6Q-A	
M30624MWP-XXXFP **			100P6S-A	
M30624MWP-XXXGP **		24K bytes	100P6Q-A	
M30625MWP-XXXGP **		128P6Q-A		
M30626MWP-XXXFP **		31K bytes	100P6S-A	
M30626MWP-XXXGP **			100P6Q-A	
M30627MWP-XXXGP **	128P6Q-A			
M30622MHP-XXXFP **	384K bytes	16K bytes	100P6S-A	
M30622MHP-XXXGP **			100P6Q-A	
M30623MHP-XXXGP **			128P6Q-A	
M30624MHP-XXXFP **		24K bytes	100P6S-A	
M30624MHP-XXXGP **			100P6Q-A	
M30625MHP-XXXGP **			128P6Q-A	
M30626MHP-XXXFP **		31K bytes	100P6S-A	
M30626MHP-XXXGP **			100P6Q-A	
M30627MHP-XXXGP **		128P6Q-A		

\* : Under planning

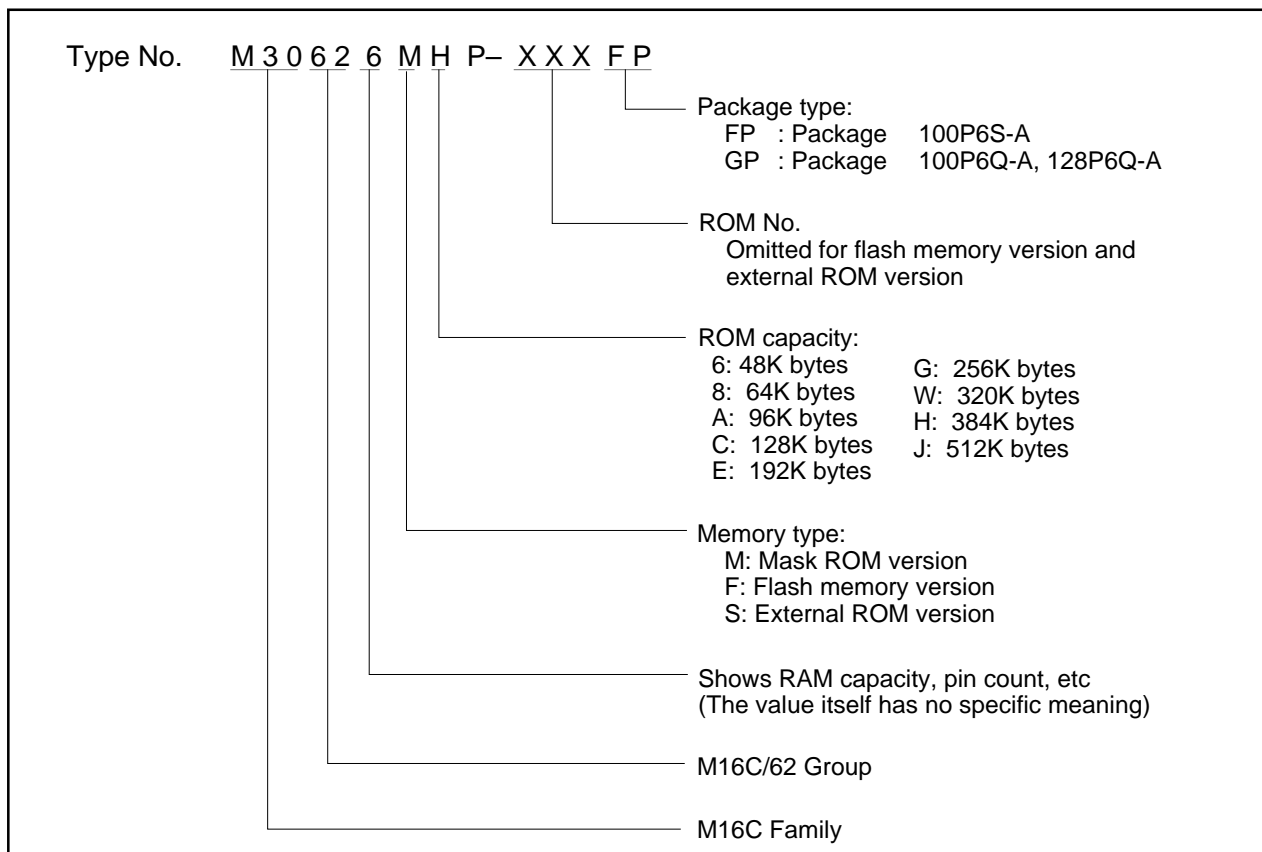
\*\* : Under development

**Table 1.1.3. Product List (2)**

As of January 2003

Type No.	ROM capacity	RAM capacity	Package type	Remarks
M30622F8PFP **	64K bytes	4K bytes	100P6S-A	Flash memory version
M30622F8PGP **			100P6Q-A	
M30620FCPFP **	128K bytes	10K bytes	100P6S-A	
M30620FCPGP **			100P6Q-A	
M30624FGPFP **	256K bytes	20K bytes	100P6S-A	
M30624FGPGP **			100P6Q-A	
M30625FGPGP **			128P6Q-A	
M30626FHPFP **	384K bytes	31K bytes	100P6S-A	
M30626FHPGP **			100P6Q-A	
M30627FHPGP **			128P6Q-A	
M30626FJPFP *	512K bytes	31K bytes	100P6S-A	
M30626FJPGP *			100P6Q-A	
M30627FJPGP *			128P6Q-A	
M30620SPFP **	—	10K bytes	100P6S-A	External ROM version
M30620SPGP **			100P6Q-A	
M30622SPFP **		4K bytes	100P6S-A	
M30622SPGP **			100P6Q-A	

\* : Under planning  
\*\* : Under development



**Figure 1.1.2. Type No., Memory Size, and Package**

### Pin Configuration

Figures 1.1.3 to 1.1.5 show the pin configurations (top view).

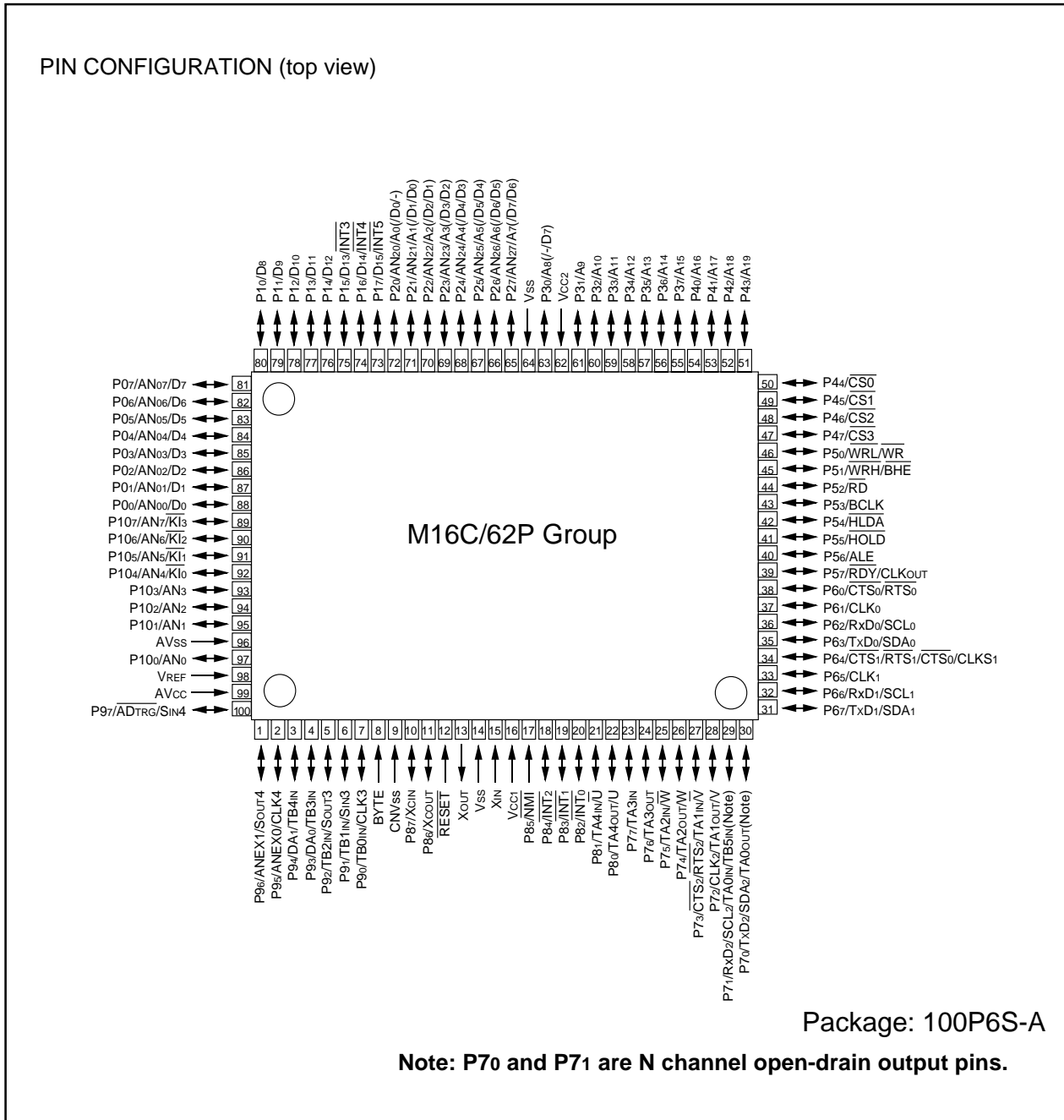


Figure 1.1.3. Pin Configuration (Top View)

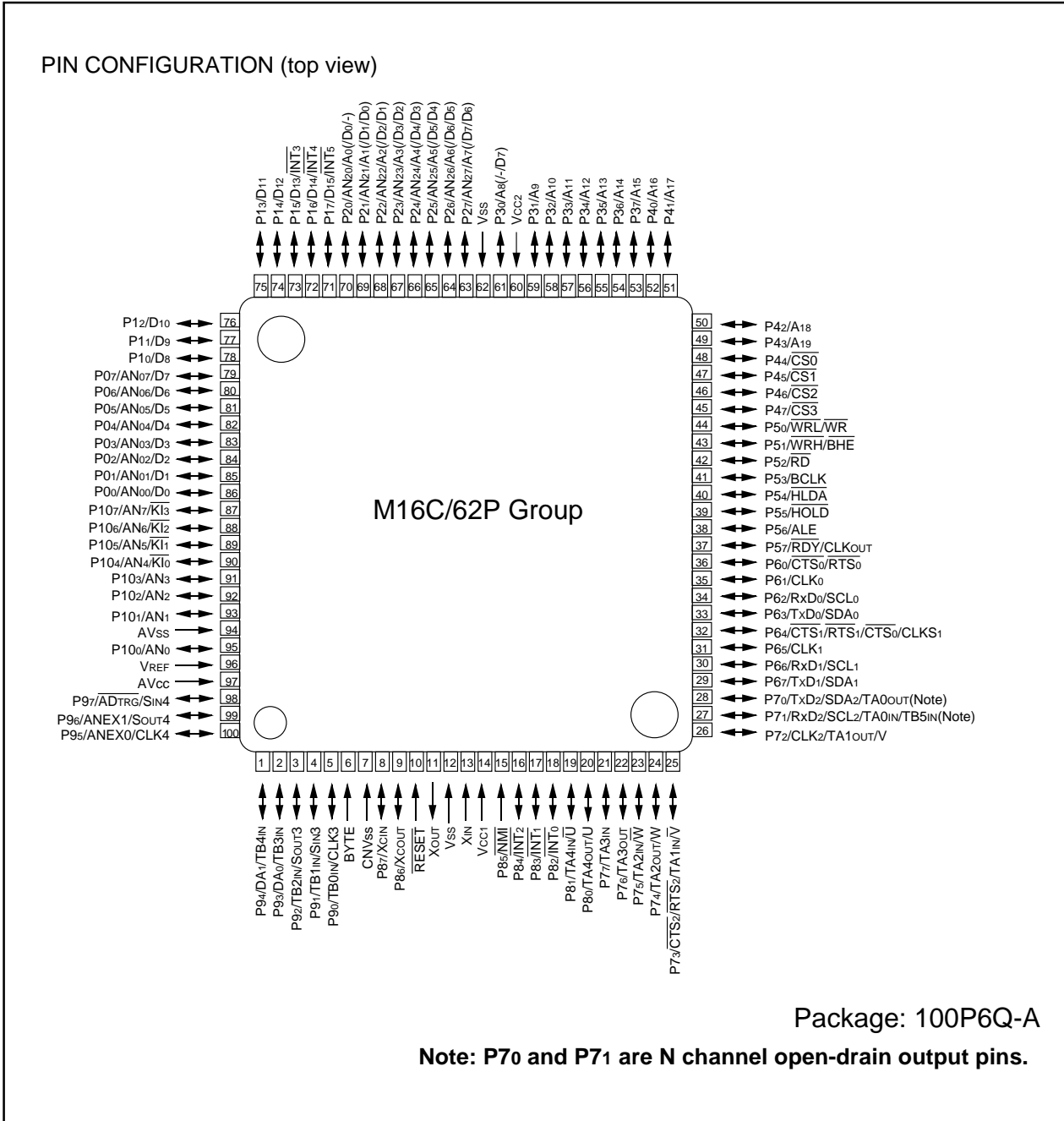


Figure 1.1.4. Pin Configuration (Top View)

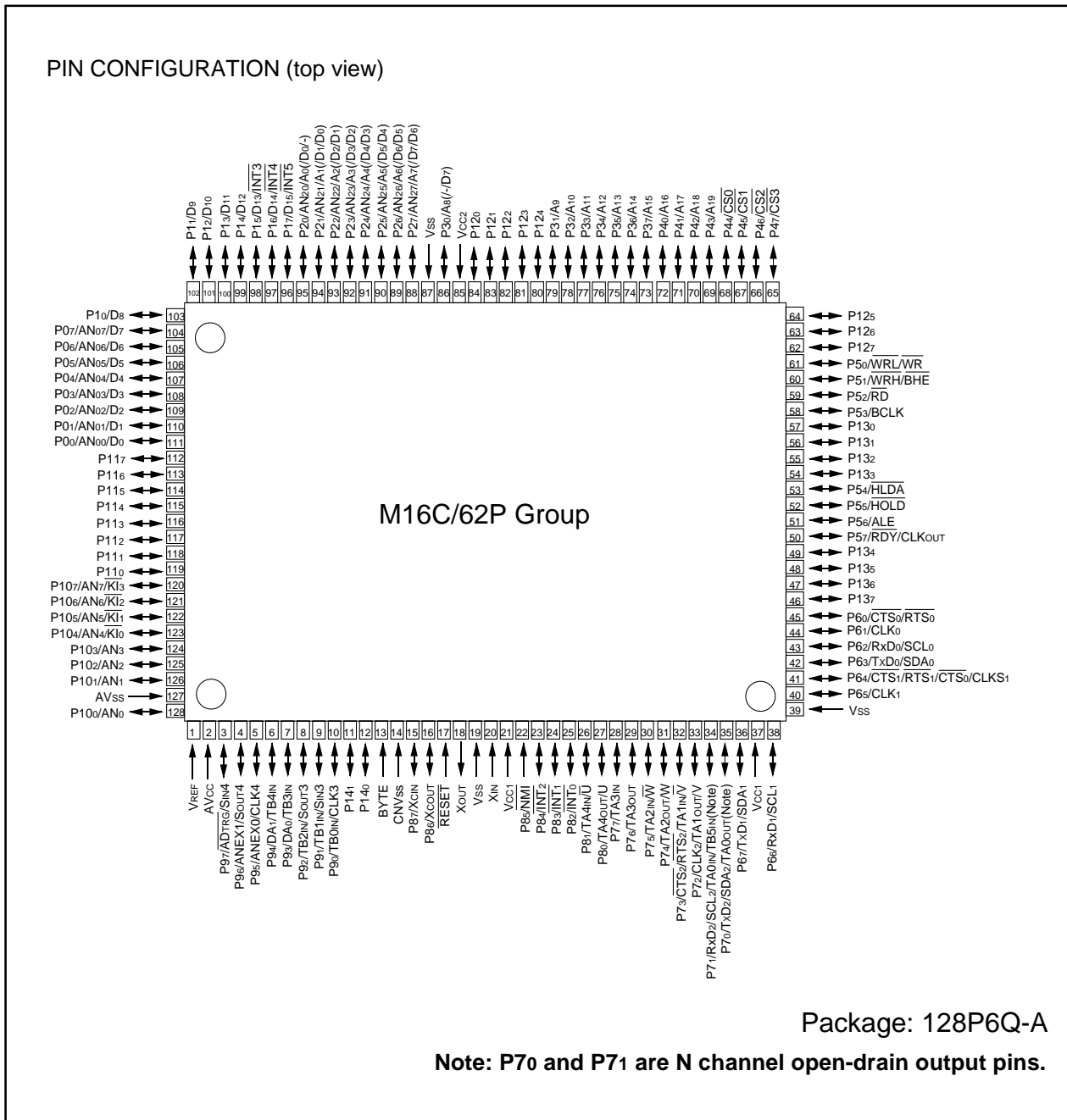


Figure 1.1.5. Pin Configuration (Top View)



## Overview

Table 1.1.4 Pin Description (100-pin and 128-pin Packages) (Continued)

Pin name	Signal name	I/O type	Power supply	Function
VCC1, VCC2, VSS	Power supply input		—	Apply 2.7V to 5.5 V to the VCC1 and VCC2 pins and 0 V to the VSS pin. The Vcc apply condition is that $VCC2 \leq VCC1$ (Note)
CNVSS	CNVSS	Input	VCC1	This pin switches between processor modes. Connect this pin to VSS pin when after a reset you want to start operation in single-chip mode (memory expansion mode) or the VCC1 pin when starting operation in microprocessor mode.
RESET	Reset input	Input	VCC1	"L" on this input resets the microcomputer.
XIN XOUT	Clock input Clock output	Input Output	VCC1	These pins are provided for the main clock generating circuit input/output. Connect a ceramic resonator or crystal between the XIN and the XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
BYTE	External data bus width select input	Input		This pin selects the width of an external data bus. A 16-bit width is selected when this input is "L"; an 8-bit width is selected when this input is "H". This input must be fixed to either "H" or "L". Connect this pin to the VSS pin when operating in single-chip mode.
AVCC	Analog power supply input			This pin is a power supply input for the A-D converter. Connect this pin to VCC1.
AVSS	Analog power supply input			This pin is a power supply input for the A-D converter. Connect this pin to VSS.
VREF	Reference voltage input	Input		This pin is a reference voltage input for the A-D converter.
P00 to P07	I/O port P0	Input/output	VCC2	This is an 8-bit CMOS I/O port. This port has an input/output select direction register, allowing each pin in that port to be directed for input or output individually. If any port is set for input, selection can be made for it in a program whether or not to have a pull-up resistor in 4 bit units. This selection is unavailable in memory extension and microprocessor modes. This port can function as input pins for the A-D converter when so selected in a program.
D0 to D7		Input/output		When set as a separate bus, these pins input and output data (D0–D7).
P10 to P17	I/O port P1	Input/output	VCC2	This is an 8-bit I/O port equivalent to P0. P15 to P17 also function as INT interrupt input pins as selected by a program.
D8 to D15		Input/output		When set as a separate bus, these pins input and output data (D8–D15).
P20 to P27	I/O port P2	Input/output	VCC2	This is an 8-bit I/O port equivalent to P0. This port can function as input pins for the A-D converter when so selected in a program.
A0 to A7		Output		These pins output 8 low-order address bits (A0 to A7).
A0/D0 to A7/D7		Input/output		If the external bus is set as an 8-bit wide multiplexed bus, these pins input and output data (D0 to D7) and output 8 low-order address bits (A0 to A7) separated in time by multiplexing.
A0 A1/D0 to A7/D6		Output Input/output		If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D0 to D6) and output address (A1 to A7) separated in time by multiplexing. They also output address (A0).
P30 to P37	I/O port P3	Input/output	VCC2	This is an 8-bit I/O port equivalent to P0.
A8 to A15		Output		These pins output 8 middle-order address bits (A8 to A15).
A8/D7, A9 to A15		Input/output Output		If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D7) and output address (A8) separated in time by multiplexing. They also output address (A9 to A15).
P40 to P47	I/O port P4	Input/output	VCC2	This is an 8-bit I/O port equivalent to P0.
A16 to A19, CS0 to CS3		Output Output		These pins output A16 to A19 and CS0 to CS3 signals. A16 to A19 are 4 high-order address bits. CS0 to CS3 are chip select signals used to specify an access space.

Note: In this manual, hereafter, VCC refers to VCC1 unless otherwise noted.

## Overview

Table 1.1.5 Pin Description (100-pin and 128-pin Packages) (Continued)

Pin name	Signal name	I/O type	Power supply	Function
P50 to P57	I/O port P5	Input/output	VCC2	This is an 8-bit I/O port equivalent to P0. In single-chip mode, P57 in this port outputs a divide-by-8 or divide-by-32 clock of XIN or a clock of the same frequency as XCIN as selected by program.
	WRL / WR, WRH / BHE, RD, BCLK, HLDA, HOLD, ALE, RDY	Output Output Output Output Input Output Input		Output WRL/WR, WRH/BHE, RD, BCLK, HLDA, and ALE signals. WRL/WR and WRH/BHE are switchable in a program. Note that WRL and WRH are always used as a pair, so as WR and BHE. <ul style="list-style-type: none"> <li>■ WRL, WRH, and RD selected</li> </ul> If the external data bus is 16 bits wide, data are written to even addresses when the WRL signal is low, and written to odd addresses when the WRH signal is low. Data are read out when the RD signal is low. <ul style="list-style-type: none"> <li>■ WR, BHE, and RD selected</li> </ul> Data are written when the WR signal is low, or read out when the RD signal is low. Odd addresses are accessed when the BHE signal is low. Use this mode when the external data bus is 8 bits wide. The microcomputer goes to a hold state when input to the HOLD pin is held low. While in the hold state, HLDA outputs a low level. ALE is used to latch the address. While the input level of the RDY pin is low, the bus of the microcomputer goes to a wait state.
P60 to P67	I/O port P6	Input/output	VCC1	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as UART0 and UART1 I/O pins as selected by program.
P70 to P77	I/O port P7	Input/output	VCC1	This is an 8-bit I/O port equivalent to P0 (P70 and P71 are N channel open-drain output). This port can function as input/output pins for timers A0 to A3 when so selected in a program. Furthermore, P70 to P75, P71, and P72 to P75 can also function as input/output pins for UART2, an input pin for timer B5, and output pins for the three-phase motor control timer, respectively.
P80 to P84, P86, P87, P85	I/O port P8  I/O port P85	Input/output Input/output Input/output Input	VCC1	P80 to P84, P86, and P87 are I/O ports with the same functions as P0. When so selected in a program, P80 to P81 and P82 to P84 can function as input/output pins for timer A4 or output pins for the three-phase motor control timer and INT interrupt input pins, respectively. P86 and P87, when so selected in a program, both can function as input/output pins for the subclock oscillator circuit. In that case, connect a crystal resonator between P86 (XCOUT pin) and P87 (XCIN pin). P85 is an input-only port shared with NMI. An NMI interrupt is generated when input on this pin changes state from high to low. The NMI function cannot be disabled in a program. A pull-up cannot be set for this pin.
P90 to P97	I/O port P9	Input/output	VCC1	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as SI/O3 and SI/O4 I/O pins, Timer B0 to B4 input pins, D-A converter output pins, A-D converter input pins, or A-D trigger input pins as selected by program.
P100 to P107	I/O port P10	Input/output	VCC1	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as A-D converter input pins as selected by program. Furthermore, P104 to P107 also function as input pins for the key input interrupt function.

Table 1.1.6 Pin Description (3) (128-pin Package) (Continued)

Pin name	Signal name	I/O type	Power supply circuit block	Function
P110 to P117	I/O port P11	Input/output	VCC1	This is an 8-bit I/O port equivalent to P0.
P120 to P127	I/O port P12	Input/output	VCC2	This is an 8-bit I/O port equivalent to P0.
P130 to P137	I/O port P13	Input/output	VCC2	This is an 8-bit I/O port equivalent to P0.
P140, P141	I/O port P14	Input/output	VCC1	This is an 2-bit I/O port equivalent to P0.

Memory

# Memory

Figure 1.2.1 is a memory map of the M16C/62P group. The address space extends the 1M bytes from address 00000<sub>16</sub> to FFFFF<sub>16</sub>.

The internal ROM is allocated in a lower address direction beginning with address FFFFF<sub>16</sub>. For example, a 64-Kbyte internal ROM is allocated to the addresses from F0000<sub>16</sub> to FFFFF<sub>16</sub>.

The fixed interrupt vector table is allocated to the addresses from FFFDC<sub>16</sub> to FFFFF<sub>16</sub>. Therefore, store the start address of each interrupt routine here.

The internal RAM is allocated in an upper address direction beginning with address 00400<sub>16</sub>. For example, a 10-Kbytes internal RAM is allocated to the addresses from 00400<sub>16</sub> to 02BFF<sub>16</sub>. In addition to storing data, the internal RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SRF is allocated to the addresses from 00000<sub>16</sub> to 003FF<sub>16</sub>. 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 FFE00<sub>16</sub> to FFFDB<sub>16</sub>. This vector is used by the JMPS or JSRS instruction. For details, refer to the "M16C/60 and M16C/20 Series Software Manual." In memory expansion and microprocessor modes, some areas are reserved for future use and cannot be used by users.

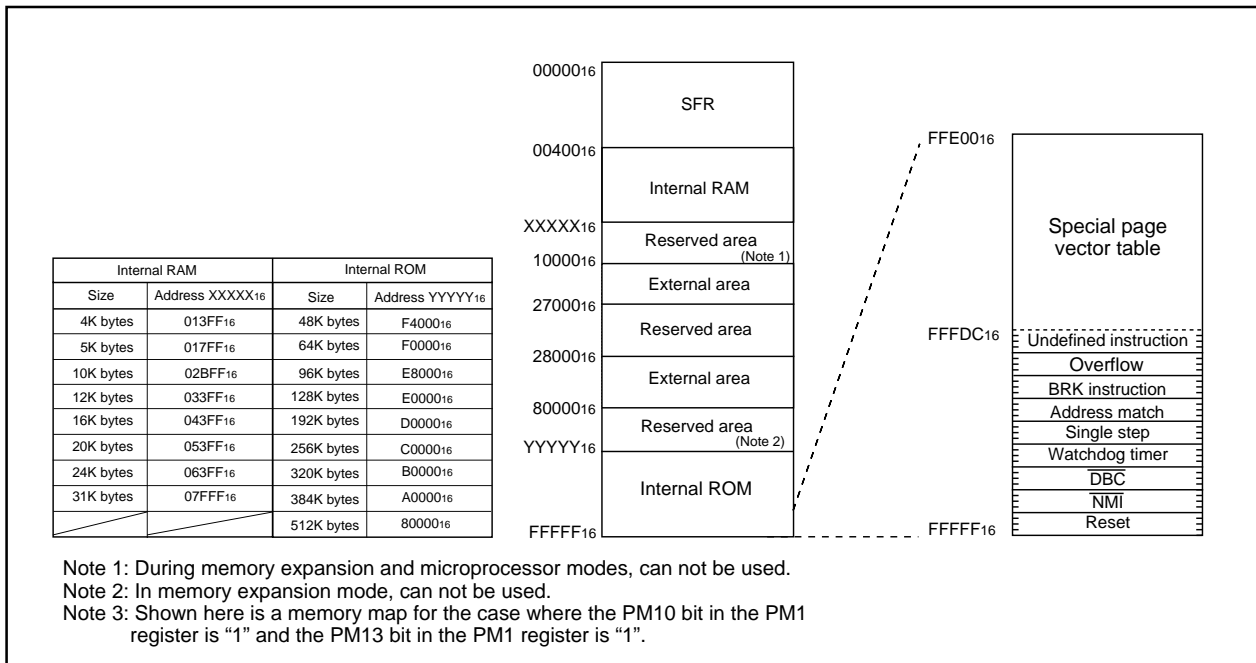
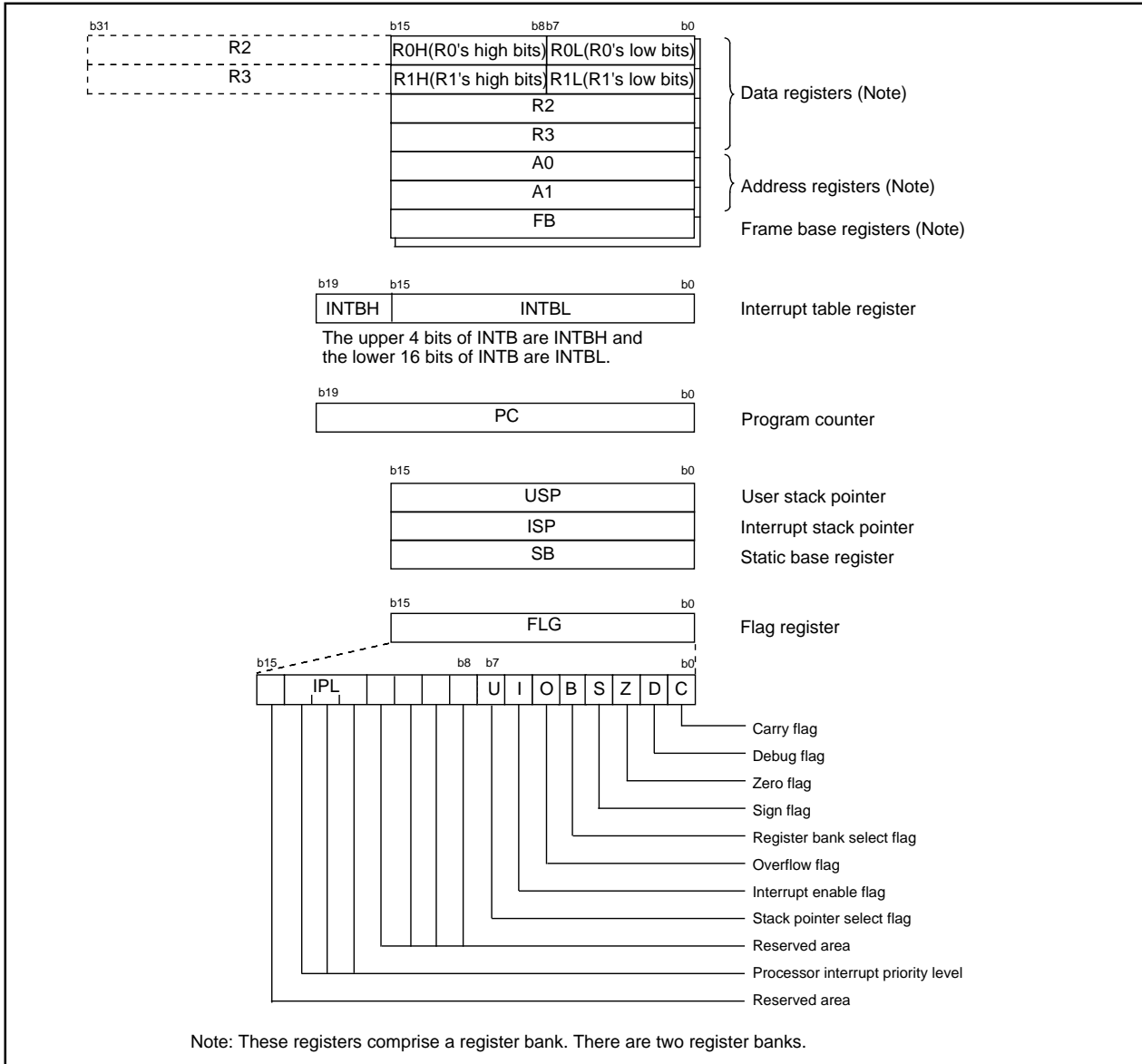


Figure 1.2.1. Memory Map

# Central Processing Unit (CPU)

## Central Processing Unit (CPU)

Figure 1.3.1 shows the CPU registers. The CPU has 13 registers. Of these, R0, R1, R2, R3, A0, A1 and FB comprise a register bank. There are two register banks.



**Figure 1.3.1. Central Processing Unit Register**

### (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) 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).

## Central Processing Unit (CPU)

---

### (3) Frame Base Register (FB)

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

### (4) Interrupt Table Register (INTB)

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

### (5) Program Counter (PC)

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

### (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.

### (7) Static Base Register (SB)

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

### (8) Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

- **Carry Flag (C Flag)**

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

- **Debug Flag (D Flag)**

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

- **Zero Flag (Z Flag)**

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

- **Sign Flag (S Flag)**

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

- **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".

- **Overflow Flag (O Flag)**

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

- **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.

- **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.

- **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.

- **Reserved Area**

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

## SFR

Address	Register	Symbol	After reset
0000 <sub>16</sub>			
0001 <sub>16</sub>			
0002 <sub>16</sub>			
0003 <sub>16</sub>			
0004 <sub>16</sub>	Processor mode register 0 (Note 2)	PM0	0000000z(CNVss pin is "L") 00000011z(CNVss pin is "H")
0005 <sub>16</sub>	Processor mode register 1	PM1	000010002
0006 <sub>16</sub>	System clock control register 0	CM0	010010002
0007 <sub>16</sub>	System clock control register 1	CM1	001000002
0008 <sub>16</sub>	Chip select control register	CSR	000000012
0009 <sub>16</sub>	Address match interrupt enable register	AIER	XXXXXXXX002
000A <sub>16</sub>	Protect register	PRCR	XX0000002
000B <sub>16</sub>	Data bank register	DBR	0016
000C <sub>16</sub>	Oscillation stop detection register (Note 3)	CM2	0000X0002
000D <sub>16</sub>			
000E <sub>16</sub>	Watchdog timer start register	WDTS	??16
000F <sub>16</sub>	Watchdog timer control register	WDC	00?????2(Note 4)
0010 <sub>16</sub>	Address match interrupt register 0	RMAD0	0016
0011 <sub>16</sub>			0016
0012 <sub>16</sub>			X016
0013 <sub>16</sub>			
0014 <sub>16</sub>	Address match interrupt register 1	RMAD1	0016
0015 <sub>16</sub>			0016
0016 <sub>16</sub>			X016
0017 <sub>16</sub>			
0018 <sub>16</sub>			
0019 <sub>16</sub>	Power supply detection register 1 (Note 5)	VCR1	000010002
001A <sub>16</sub>	Power supply detection register 2 (Note 5)	VCR2	0016
001B <sub>16</sub>	Chip select expansion control register	CSE	0016
001C <sub>16</sub>	PLL control register 0	PLC0	0001X0102
001D <sub>16</sub>			
001E <sub>16</sub>	Processor mode register 2	PM2	XXX000002
001F <sub>16</sub>	Power supply down detection interrupt register	D4INT	0016
0020 <sub>16</sub>	DMA0 source pointer	SAR0	??16
0021 <sub>16</sub>			??16
0022 <sub>16</sub>			X?16
0023 <sub>16</sub>			
0024 <sub>16</sub>	DMA0 destination pointer	DAR0	??16
0025 <sub>16</sub>			??16
0026 <sub>16</sub>			X?16
0027 <sub>16</sub>			
0028 <sub>16</sub>	DMA0 transfer counter	TCR0	??16
0029 <sub>16</sub>			??16
002A <sub>16</sub>			
002B <sub>16</sub>			
002C <sub>16</sub>	DMA0 control register	DM0CON	00000?002
002D <sub>16</sub>			
002E <sub>16</sub>			
002F <sub>16</sub>			
0030 <sub>16</sub>	DMA1 source pointer	SAR1	??16
0031 <sub>16</sub>			??16
0032 <sub>16</sub>			X?16
0033 <sub>16</sub>			
0034 <sub>16</sub>	DMA1 destination pointer	DAR1	??16
0035 <sub>16</sub>			??16
0036 <sub>16</sub>			X?16
0037 <sub>16</sub>			
0038 <sub>16</sub>	DMA1 transfer counter	TCR1	??16
0039 <sub>16</sub>			??16
003A <sub>16</sub>			
003B <sub>16</sub>			
003C <sub>16</sub>	DMA1 control register	DM1CON	00000?002
003D <sub>16</sub>			
003E <sub>16</sub>			
003F <sub>16</sub>			

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

Note 2: The PM00 and PM01 bits do not change at software reset, watchdog timer reset and oscillation stop detection reset.

Note 3: The CM20, CM21, and CM27 bits do not change at oscillation stop detection reset.

Note 4: The WDC5 bit is "0" (cold start) immediately after power-on. It can only be set to "1" in a program. It is set to "0" when the input voltage at the V<sub>CC1</sub> pin drops to V<sub>dets</sub> or less while the VC25 bit in the VCR2 register is set to "1" (RAM retention limit detection circuit enable).

Note 5: This register does not change at software reset, watchdog timer reset and oscillation stop detection reset.

X : Nothing is mapped to this bit

? : Undefined

## SFR

Address	Register	Symbol	After reset
0040 <sub>16</sub>			
0041 <sub>16</sub>			
0042 <sub>16</sub>			
0043 <sub>16</sub>			
0044 <sub>16</sub>	INT3 interrupt control register	INT3IC	XX00?0002
0045 <sub>16</sub>	Timer B5 interrupt control register	TB5IC	XXXX?0002
0046 <sub>16</sub>	Timer B4 interrupt control register, UART1 BUS collision detection interrupt control register	TB4IC, U1BCNIC	XXXX?0002
0047 <sub>16</sub>	Timer B3 interrupt control register, UART0 BUS collision detection interrupt control register	TB3IC, U0BCNIC	XXXX?0002
0048 <sub>16</sub>	SI/O4 interrupt control register (S4IC), INT5 interrupt control register	S4IC, INT5IC	XX00?0002
0049 <sub>16</sub>	SI/O3 interrupt control register, INT4 interrupt control register	S3IC, INT4IC	XX00?0002
004A <sub>16</sub>	UART2 Bus collision detection interrupt control register	BCNIC	XXXX?0002
004B <sub>16</sub>	DMA0 interrupt control register	DM0IC	XXXX?0002
004C <sub>16</sub>	DMA1 interrupt control register	DM1IC	XXXX?0002
004D <sub>16</sub>	Key input interrupt control register	KUPIC	XXXX?0002
004E <sub>16</sub>	A-D conversion interrupt control register	ADIC	XXXX?0002
004F <sub>16</sub>	UART2 transmit interrupt control register	S2TIC	XXXX?0002
0050 <sub>16</sub>	UART2 receive interrupt control register	S2RIC	XXXX?0002
0051 <sub>16</sub>	UART0 transmit interrupt control register	S0TIC	XXXX?0002
0052 <sub>16</sub>	UART0 receive interrupt control register	S0RIC	XXXX?0002
0053 <sub>16</sub>	UART1 transmit interrupt control register	S1TIC	XXXX?0002
0054 <sub>16</sub>	UART1 receive interrupt control register	S1RIC	XXXX?0002
0055 <sub>16</sub>	Timer A0 interrupt control register	TA0IC	XXXX?0002
0056 <sub>16</sub>	Timer A1 interrupt control register	TA1IC	XXXX?0002
0057 <sub>16</sub>	Timer A2 interrupt control register	TA2IC	XXXX?0002
0058 <sub>16</sub>	Timer A3 interrupt control register	TA3IC	XXXX?0002
0059 <sub>16</sub>	Timer A4 interrupt control register	TA4IC	XXXX?0002
005A <sub>16</sub>	Timer B0 interrupt control register	TB0IC	XXXX?0002
005B <sub>16</sub>	Timer B1 interrupt control register	TB1IC	XXXX?0002
005C <sub>16</sub>	Timer B2 interrupt control register	TB2IC	XXXX?0002
005D <sub>16</sub>	INT0 interrupt control register	INT0IC	XX00?0002
005E <sub>16</sub>	INT1 interrupt control register	INT1IC	XX00?0002
005F <sub>16</sub>	INT2 interrupt control register	INT2IC	XX00?0002
0060 <sub>16</sub>			
0061 <sub>16</sub>			
0062 <sub>16</sub>			
0063 <sub>16</sub>			
0064 <sub>16</sub>			
0065 <sub>16</sub>			
0066 <sub>16</sub>			
0067 <sub>16</sub>			
0068 <sub>16</sub>			
0069 <sub>16</sub>			
006A <sub>16</sub>			
006B <sub>16</sub>			
006C <sub>16</sub>			
006D <sub>16</sub>			
006E <sub>16</sub>			
006F <sub>16</sub>			
0070 <sub>16</sub>			
0071 <sub>16</sub>			
0072 <sub>16</sub>			
0073 <sub>16</sub>			
0074 <sub>16</sub>			
0075 <sub>16</sub>			
0076 <sub>16</sub>			
0077 <sub>16</sub>			
0078 <sub>16</sub>			
0079 <sub>16</sub>			
007A <sub>16</sub>			
007B <sub>16</sub>			
007C <sub>16</sub>			
007D <sub>16</sub>			
007E <sub>16</sub>			
007F <sub>16</sub>			

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

X : Nothing is mapped to this bit

? : Undefined

**SFR**

Address	Register	Symbol	After reset
0080 <sub>16</sub>			
0081 <sub>16</sub>			
0082 <sub>16</sub>			
0083 <sub>16</sub>			
0084 <sub>16</sub>			
0085 <sub>16</sub>			
0086 <sub>16</sub>			
≈			≈
01B0 <sub>16</sub>			
01B1 <sub>16</sub>			
01B2 <sub>16</sub>			
01B3 <sub>16</sub>			
01B4 <sub>16</sub>	Flash identification register (Note 2)	FIDR	XXXXXX002
01B5 <sub>16</sub>	Flash memory control register 1 (Note 2)	FMR1	0?00??0?2
01B6 <sub>16</sub>			
01B7 <sub>16</sub>	Flash memory control register 0 (Note 2)	FMR0	??0000012
01B8 <sub>16</sub>	Address match interrupt register 2	RMAD2	0016
01B9 <sub>16</sub>			0016
01BA <sub>16</sub>			X016
01BB <sub>16</sub>	Address match interrupt enable register 2	AIER2	XXXXXX002
01BC <sub>16</sub>	Address match interrupt register 3	RMAD3	0016
01BD <sub>16</sub>			0016
01BE <sub>16</sub>			X016
01BF <sub>16</sub>			
≈			≈
0250 <sub>16</sub>			
0251 <sub>16</sub>			
0252 <sub>16</sub>			
0253 <sub>16</sub>			
0254 <sub>16</sub>			
0255 <sub>16</sub>			
0256 <sub>16</sub>			
0257 <sub>16</sub>			
0258 <sub>16</sub>			
0259 <sub>16</sub>			
025A <sub>16</sub>			
025B <sub>16</sub>			
025C <sub>16</sub>			
025D <sub>16</sub>			
025E <sub>16</sub>	Peripheral clock select register	PCLKR	000000112
025F <sub>16</sub>			
≈			≈
0330 <sub>16</sub>			
0331 <sub>16</sub>			
0332 <sub>16</sub>			
0333 <sub>16</sub>			
0334 <sub>16</sub>			
0335 <sub>16</sub>			
0336 <sub>16</sub>			
0337 <sub>16</sub>			
0338 <sub>16</sub>			
0339 <sub>16</sub>			
033A <sub>16</sub>			
033B <sub>16</sub>			
033C <sub>16</sub>			
033D <sub>16</sub>			
033E <sub>16</sub>			
033F <sub>16</sub>			

Note 1: The blank areas are reserved and cannot be used by users.  
Note 2: This register is included in the flash memory version.

X : Nothing is mapped to this bit  
? : Undefined



## SFR

Address	Register	Symbol	After reset
0340 <sub>16</sub>	Timer B3, 4, 5 count start flag	TBSR	000XXXXX2
0341 <sub>16</sub>			
0342 <sub>16</sub> 0343 <sub>16</sub>	Timer A1-1 register	TA11	?? <sub>16</sub> ?? <sub>16</sub>
0344 <sub>16</sub> 0345 <sub>16</sub>	Timer A2-1 register	TA21	?? <sub>16</sub> ?? <sub>16</sub>
0346 <sub>16</sub> 0347 <sub>16</sub>	Timer A4-1 register	TA41	?? <sub>16</sub> ?? <sub>16</sub>
0348 <sub>16</sub>	Three-phase PWM control register 0	INVC0	00 <sub>16</sub>
0349 <sub>16</sub>	Three-phase PWM control register 1	INVC1	00 <sub>16</sub>
034A <sub>16</sub>	Three-phase output buffer register 0	IDB0	00 <sub>16</sub>
034B <sub>16</sub>	Three-phase output buffer register 1	IDB1	00 <sub>16</sub>
034C <sub>16</sub>	Dead time timer	DTT	?? <sub>16</sub>
034D <sub>16</sub> 034E <sub>16</sub> 034F <sub>16</sub>	Timer B2 interrupt occurrence frequency set counter	ICTB2	?? <sub>16</sub>
0350 <sub>16</sub> 0351 <sub>16</sub>	Timer B3 register	TB3	?? <sub>16</sub> ?? <sub>16</sub>
0352 <sub>16</sub> 0353 <sub>16</sub>	Timer B4 register	TB4	?? <sub>16</sub> ?? <sub>16</sub>
0354 <sub>16</sub> 0355 <sub>16</sub> 0356 <sub>16</sub>	Timer B5 register	TB5	?? <sub>16</sub> ?? <sub>16</sub> ?? <sub>16</sub>
0357 <sub>16</sub> 0358 <sub>16</sub> 0359 <sub>16</sub> 035A <sub>16</sub>			
035B <sub>16</sub>	Timer B3 mode register	TB3MR	00??00002
035C <sub>16</sub>	Timer B4 mode register	TB4MR	00?X00002
035D <sub>16</sub>	Timer B5 mode register	TB5MR	00?X00002
035E <sub>16</sub>	Interrupt cause select register 2	IFSR2A	00XXXXXX2
035F <sub>16</sub>	Interrupt cause select register	IFSR	00 <sub>16</sub>
0360 <sub>16</sub> 0361 <sub>16</sub>	SI/O3 transmit/receive register	S3TRR	?? <sub>16</sub>
0362 <sub>16</sub>	SI/O3 control register	S3C	010000002
0363 <sub>16</sub>	SI/O3 bit rate generator	S3BRG	?? <sub>16</sub>
0364 <sub>16</sub> 0365 <sub>16</sub>	SI/O4 transmit/receive register	S4TRR	?? <sub>16</sub>
0366 <sub>16</sub>	SI/O4 control register	S4C	010000002
0367 <sub>16</sub> 0368 <sub>16</sub> 0369 <sub>16</sub> 036A <sub>16</sub> 036B <sub>16</sub>	SI/O4 bit rate generator	S4BRG	?? <sub>16</sub>
036C <sub>16</sub>	UART0 special mode register 4	U0SMR4	00 <sub>16</sub>
036D <sub>16</sub>	UART0 special mode register 3	U0SMR3	000X0X0X2
036E <sub>16</sub>	UART0 special mode register 2	U0SMR2	X00000002
036F <sub>16</sub>	UART0 special mode register	U0SMR	X00000002
0370 <sub>16</sub>	UART1 special mode register 4	U1SMR4	00 <sub>16</sub>
0371 <sub>16</sub>	UART1 special mode register 3	U1SMR3	000X0X0X2
0372 <sub>16</sub>	UART1 special mode register 2	U1SMR2	X00000002
0373 <sub>16</sub>	UART1 special mode register	U1SMR	X00000002
0374 <sub>16</sub>	UART2 special mode register 4	U2SMR4	00 <sub>16</sub>
0375 <sub>16</sub>	UART2 special mode register 3	U2SMR3	000X0X0X2
0376 <sub>16</sub>	UART2 special mode register 2	U2SMR2	X00000002
0377 <sub>16</sub>	UART2 special mode register	U2SMR	X00000002
0378 <sub>16</sub>	UART2 transmit/receive mode register	U2MR	00 <sub>16</sub>
0379 <sub>16</sub>	UART2 bit rate generator	U2BRG	?? <sub>16</sub>
037A <sub>16</sub> 037B <sub>16</sub>	UART2 transmit buffer register	U2TB	????????2 XXXXXXXX?2
037C <sub>16</sub>	UART2 transmit/receive control register 0	U2C0	000010002
037D <sub>16</sub>	UART2 transmit/receive control register 1	U2C1	000000102
037E <sub>16</sub> 037F <sub>16</sub>	UART2 receive buffer register	U2RB	????????2 ?????XX?2

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

X : Nothing is mapped to this bit

? : Undefined

## SFR

Address	Register	Symbol	After reset
0380 <sub>16</sub>	Count start flag	TABSR	0016
0381 <sub>16</sub>	Clock prescaler reset flag	CPSRF	0XXXXXXX <sub>2</sub>
0382 <sub>16</sub>	One-shot start flag	ONSF	0016
0383 <sub>16</sub>	Trigger select register	TRGSR	0016
0384 <sub>16</sub>	Up-down flag	UDF	0016
0385 <sub>16</sub>			
0386 <sub>16</sub> 0387 <sub>16</sub>	Timer A0 register	TA0	?? <sub>16</sub> ?? <sub>16</sub>
0388 <sub>16</sub> 0389 <sub>16</sub>	Timer A1 register	TA1	?? <sub>16</sub> ?? <sub>16</sub>
038A <sub>16</sub> 038B <sub>16</sub>	Timer A2 register	TA2	?? <sub>16</sub> ?? <sub>16</sub>
038C <sub>16</sub> 038D <sub>16</sub>	Timer A3 register	TA3	?? <sub>16</sub> ?? <sub>16</sub>
038E <sub>16</sub> 038F <sub>16</sub>	Timer A4 register	TA4	?? <sub>16</sub> ?? <sub>16</sub>
0390 <sub>16</sub> 0391 <sub>16</sub>	Timer B0 register	TB0	?? <sub>16</sub> ?? <sub>16</sub>
0392 <sub>16</sub> 0393 <sub>16</sub>	Timer B1 register	TB1	?? <sub>16</sub> ?? <sub>16</sub>
0394 <sub>16</sub> 0395 <sub>16</sub>	Timer B2 register	TB2	?? <sub>16</sub> ?? <sub>16</sub>
0396 <sub>16</sub>	Timer A0 mode register	TA0MR	0016
0397 <sub>16</sub>	Timer A1 mode register	TA1MR	0016
0398 <sub>16</sub>	Timer A2 mode register	TA2MR	0016
0399 <sub>16</sub>	Timer A3 mode register	TA3MR	0016
039A <sub>16</sub>	Timer A4 mode register	TA4MR	0016
039B <sub>16</sub>	Timer B0 mode register	TB0MR	00??0000 <sub>2</sub>
039C <sub>16</sub>	Timer B1 mode register	TB1MR	00?X0000 <sub>2</sub>
039D <sub>16</sub>	Timer B2 mode register	TB2MR	00?X0000 <sub>2</sub>
039E <sub>16</sub>	Timer B2 special mode register	TB2SC	XXXXXX00 <sub>2</sub>
039F <sub>16</sub>			
03A0 <sub>16</sub>	UART0 transmit/receive mode register	U0MR	0016
03A1 <sub>16</sub>	UART0 bit rate generator	U0BRG	?? <sub>16</sub>
03A2 <sub>16</sub> 03A3 <sub>16</sub>	UART0 transmit buffer register	U0TB	??????? <sub>2</sub> XXXXXXX <sub>2</sub>
03A4 <sub>16</sub>	UART0 transmit/receive control register 0	U0C0	00001000 <sub>2</sub>
03A5 <sub>16</sub>	UART0 transmit/receive control register 1	U0C1	00000010 <sub>2</sub>
03A6 <sub>16</sub> 03A7 <sub>16</sub>	UART0 receive buffer register	U0RB	??????? <sub>2</sub> ?????XX <sub>2</sub>
03A8 <sub>16</sub>	UART1 transmit/receive mode register	U1MR	0016
03A9 <sub>16</sub>	UART1 bit rate generator	U1BRG	?? <sub>16</sub>
03AA <sub>16</sub> 03AB <sub>16</sub>	UART1 transmit buffer register	U1TB	??????? <sub>2</sub> XXXXXXX <sub>2</sub>
03AC <sub>16</sub>	UART1 transmit/receive control register 0	U1C0	00001000 <sub>2</sub>
03AD <sub>16</sub>	UART1 transmit/receive control register 1	U1C1	00000010 <sub>2</sub>
03AE <sub>16</sub> 03AF <sub>16</sub>	UART1 receive buffer register	U1RB	??????? <sub>2</sub> ?????XX <sub>2</sub>
03B0 <sub>16</sub>	UART transmit/receive control register 2	UCON	X0000000 <sub>2</sub>
03B1 <sub>16</sub>			
03B2 <sub>16</sub>			
03B3 <sub>16</sub>			
03B4 <sub>16</sub>			
03B5 <sub>16</sub>			
03B6 <sub>16</sub>			
03B7 <sub>16</sub>			
03B8 <sub>16</sub>	DMA0 request cause select register	DM0SL	0016
03B9 <sub>16</sub>			
03BA <sub>16</sub>	DMA1 request cause select register	DM1SL	0016
03BB <sub>16</sub>			
03BC <sub>16</sub>	CRC data register	CRCD	?? <sub>16</sub>
03BD <sub>16</sub>			?? <sub>16</sub>
03BE <sub>16</sub>	CRC input register	CRCIN	?? <sub>16</sub>
03BF <sub>16</sub>			

Note : The blank areas are reserved and cannot be used by users.  
X : Nothing is mapped to this bit  
? : Undefined

Address	Register	Symbol	After reset
03C0 <sub>16</sub> 03C1 <sub>16</sub>	A-D register 0	AD0	???????2 XXXXXX??2
03C2 <sub>16</sub> 03C3 <sub>16</sub>	A-D register 1	AD1	???????2 XXXXXX??2
03C4 <sub>16</sub> 03C5 <sub>16</sub>	A-D register 2	AD2	???????2 XXXXXX??2
03C6 <sub>16</sub> 03C7 <sub>16</sub>	A-D register 3	AD3	???????2 XXXXXX??2
03C8 <sub>16</sub> 03C9 <sub>16</sub>	A-D register 4	AD4	???????2 XXXXXX??2
03CA <sub>16</sub> 03CB <sub>16</sub>	A-D register 5	AD5	???????2 XXXXXX??2
03CC <sub>16</sub> 03CD <sub>16</sub>	A-D register 6	AD6	???????2 XXXXXX??2
03CE <sub>16</sub> 03CF <sub>16</sub>	A-D register 7	AD7	???????2 XXXXXX??2
03D0 <sub>16</sub>			
03D1 <sub>16</sub>			
03D2 <sub>16</sub>			
03D3 <sub>16</sub>			
03D4 <sub>16</sub> 03D5 <sub>16</sub>	A-D control register 2	ADCON2	0016
03D6 <sub>16</sub>	A-D control register 0	ADCON0	00000???2
03D7 <sub>16</sub>	A-D control register 1	ADCON1	0016
03D8 <sub>16</sub> 03D9 <sub>16</sub>	D-A register 0	DA0	??16
03DA <sub>16</sub> 03DB <sub>16</sub>	D-A register 1	DA1	??16
03DC <sub>16</sub> 03DD <sub>16</sub>	D-A control register	DACON	0016
03DE <sub>16</sub>	Port P14 control register	PC14	XX00XXXX2
03DF <sub>16</sub>	Pull-up control register 3	PUR3	0016
03E0 <sub>16</sub>	Port P0 register	P0	??16
03E1 <sub>16</sub>	Port P1 register	P1	??16
03E2 <sub>16</sub>	Port P0 direction register	PD0	0016
03E3 <sub>16</sub>	Port P1 direction register	PD1	0016
03E4 <sub>16</sub>	Port P2 register	P2	??16
03E5 <sub>16</sub>	Port P3 register	P3	??16
03E6 <sub>16</sub>	Port P2 direction register	PD2	0016
03E7 <sub>16</sub>	Port P3 direction register	PD3	0016
03E8 <sub>16</sub>	Port P4 register	P4	??16
03E9 <sub>16</sub>	Port P5 register	P5	??16
03EA <sub>16</sub>	Port P4 direction register	PD4	0016
03EB <sub>16</sub>	Port P5 direction register	PD5	0016
03EC <sub>16</sub>	Port P6 register	P6	??16
03ED <sub>16</sub>	Port P7 register	P7	??16
03EE <sub>16</sub>	Port P6 direction register	PD6	0016
03EF <sub>16</sub>	Port P7 direction register	PD7	0016
03F0 <sub>16</sub>	Port P8 register	P8	??16
03F1 <sub>16</sub>	Port P9 register	P9	??16
03F2 <sub>16</sub>	Port P8 direction register	PD8	00X000002
03F3 <sub>16</sub>	Port P9 direction register	PD9	0016
03F4 <sub>16</sub>	Port P10 register	P10	??16
03F5 <sub>16</sub>	Port P11 register	P11	??16
03F6 <sub>16</sub>	Port P10 direction register	PD10	0016
03F7 <sub>16</sub>	Port P11 direction register	PD11	0016
03F8 <sub>16</sub>	Port P12 register	P12	??16
03F9 <sub>16</sub>	Port P13 register	P13	??16
03FA <sub>16</sub>	Port P12 direction register	PD12	0016
03FB <sub>16</sub>	Port P13 direction register	PD13	0016
03FC <sub>16</sub>	Pull-up control register 0	PUR0	0016
03FD <sub>16</sub>	Pull-up control register 1	PUR1	00000000 <sub>2</sub> 00000010 <sub>2</sub> (Note 2)
03FE <sub>16</sub>	Pull-up control register 2	PUR2	0016
03FF <sub>16</sub>	Port control register	PCR	0016

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

Note 2: At hardware reset 1 or hardware reset 2, the register is as follows:

- "00000000<sub>2</sub>" where "L" is inputted to the CNV<sub>ss</sub> pin
- "00000010<sub>2</sub>" where "H" is inputted to the CNV<sub>ss</sub> pin

At software reset, watchdog timer reset and oscillation stop detection reset, the register is as follows:

- "00000000<sub>2</sub>" where the PM01 to PM00 bits in the PM0 register are "00<sub>2</sub>" (single-chip mode)
- "00000010<sub>2</sub>" where the PM01 to PM00 bits in the PM0 register are "01<sub>2</sub>" (memory expansion mode) or "11<sub>2</sub>" (microprocessor mode)

X : Nothing is mapped to this bit

? : Undefined

## Reset

There are four types of resets: a hardware reset, a software reset, an watchdog timer reset, and an oscillation stop detection reset.

### Hardware Reset

There are two types of hardware resets: a hardware reset 1 and a hardware reset 2.

#### Hardware Reset 1

A reset is applied using the  $\overline{\text{RESET}}$  pin. When an "L" signal is applied to the  $\overline{\text{RESET}}$  pin while the power supply voltage is within the recommended operating condition, the pins are initialized (see Table 1.5.1). The oscillation circuit is initialized and the main clock starts oscillating. When the input level at the  $\overline{\text{RESET}}$  pin is released from "L" to "H", the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. The internal RAM is not initialized. If the  $\overline{\text{RESET}}$  pin is pulled "L" while writing to the internal RAM, the internal RAM becomes indeterminate.

Figure 1.5.1 shows the example reset circuit. Figure 1.5.2 shows the reset sequence. Table 1.5.1 shows the statuses of the other pins while the  $\overline{\text{RESET}}$  pin is "L". Figure 1.5.3 shows the CPU register status after reset. Refer to "SFR" for SFR status after reset.

1. When the power supply is stable

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

2. Power on

- (1) Apply an "L" signal to the  $\overline{\text{RESET}}$  pin.
- (2) Let the power supply voltage increase until it meets the recommended operating condition.
- (3) Wait  $t_d(\text{P-R})$  or more until the internal power supply stabilizes.
- (4) Supply a clock for 20 cycles or more to the XIN pin.
- (5) Apply an "H" signal to the  $\overline{\text{RESET}}$  pin.

#### Hardware Reset 2

This reset is generated by the microcomputer's internal voltage detection circuit. The voltage detection circuit monitors the voltage supplied to the VCC1 pin.

If the VC26 bit in the VCR2 register is set to "1" (reset level detection circuit enabled), the microcomputer is reset when the voltage at the VCC1 input pin drops below Vdet3.

Similarly, if the VC25 bit in the VCR2 register is set to "1" (RAM retention limit detection circuit enabled), the microcomputer is reset when the voltage at the VCC1 input pin drops below Vdet2.

Conversely, when the input voltage at the VCC1 pin rises to Vdet3 or more, the pins and the CPU and SFR are initialized, and the program is executed starting from the address indicated by the reset vector. It takes about  $t_d(\text{S-R})$  before the program starts running after Vdet3 is detected. The initialized pins and registers and the status thereof are the same as in hardware reset 1.

Set the CM10 bit in the CM1 register to "1" (stop mode) after setting the VC25 bit to "1" (RAM retention limit detection circuit enabled), and the microcomputer will be reset when the voltage at the VCC1 input pin drops below Vdet2 and comes out of reset when the voltage at the VCC1 input pin rises above Vdet3. During stop mode, the value set in the VC26 bit has no effect. Therefore, no reset is generated even when the input voltage at the VCC1 pin drops to Vdet3 or less.

## Reset

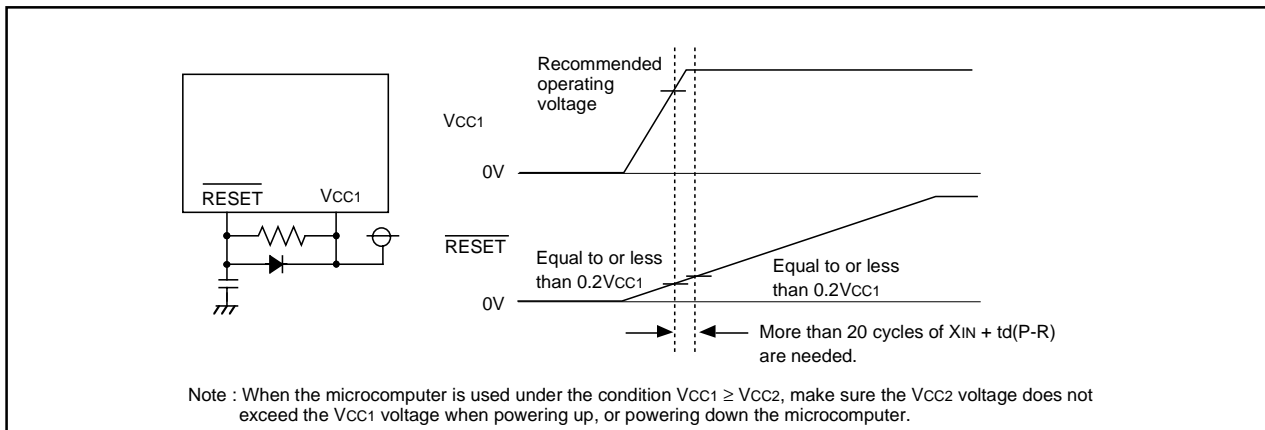


Figure 1.5.1. Example Reset Circuit

### Software Reset

When the PM03 bit in the PM0 register is set to “1” (microcomputer reset), the microcomputer has its pins, CPU, and SFR initialized. Then the program is executed starting from the address indicated by the reset vector.

Select the main clock for the CPU clock source, and set the PM03 bit to “1” with main clock oscillation satisfactorily stable.

At software reset, some SFR’s are not initialized. Refer to “SFR”. Also, since the PM01 to PM00 bits in the PM0 register are not initialized, the processor mode remains unchanged.

### Watchdog Timer Reset

Where the PM12 bit in the PM1 register is “1” (reset when watchdog timer underflows), the microcomputer initializes its pins, CPU and SFR if the watchdog timer underflows. Then the program is executed starting from the address indicated by the reset vector.

At watchdog timer reset, some SFR’s are not initialized. Refer to “SFR”. Also, since the PM01 to PM00 bits in the PM0 register are not initialized, the processor mode remains unchanged.

### Oscillation Stop Detection Reset

Where the CM27 bit in the CM2 register is “0” (reset at oscillation stop detection), the microcomputer initializes its pins, CPU and SFR, coming to a halt if it detects main clock oscillation circuit stop. Refer to the section “oscillation stop, re-oscillation detection function”.

At oscillation stop detection reset, some SFR’s are not initialized. Refer to the section “SFR”. Also, since the PM01 to PM00 bits in the PM0 register are not initialized, the processor mode remains unchanged.

# Reset

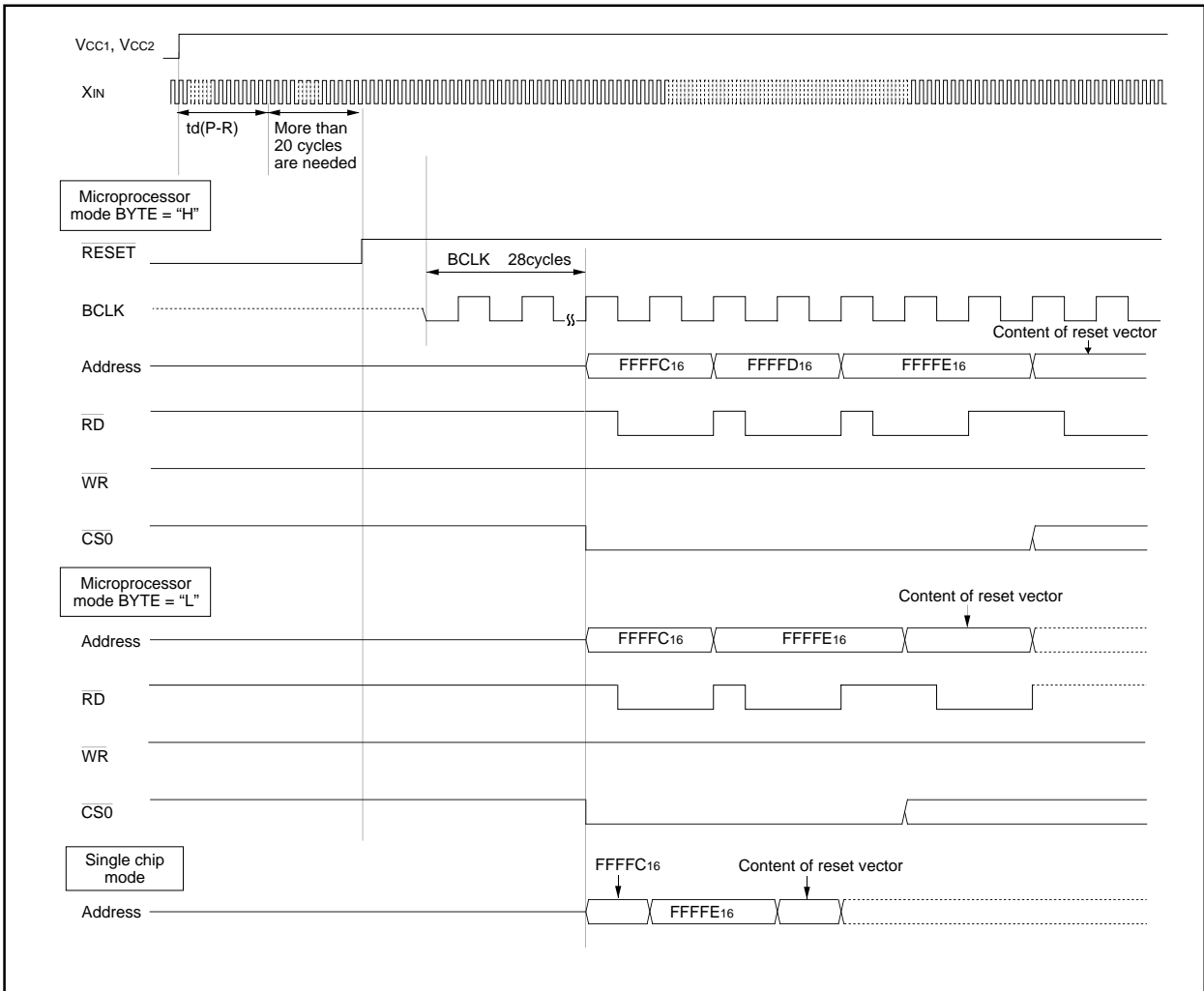


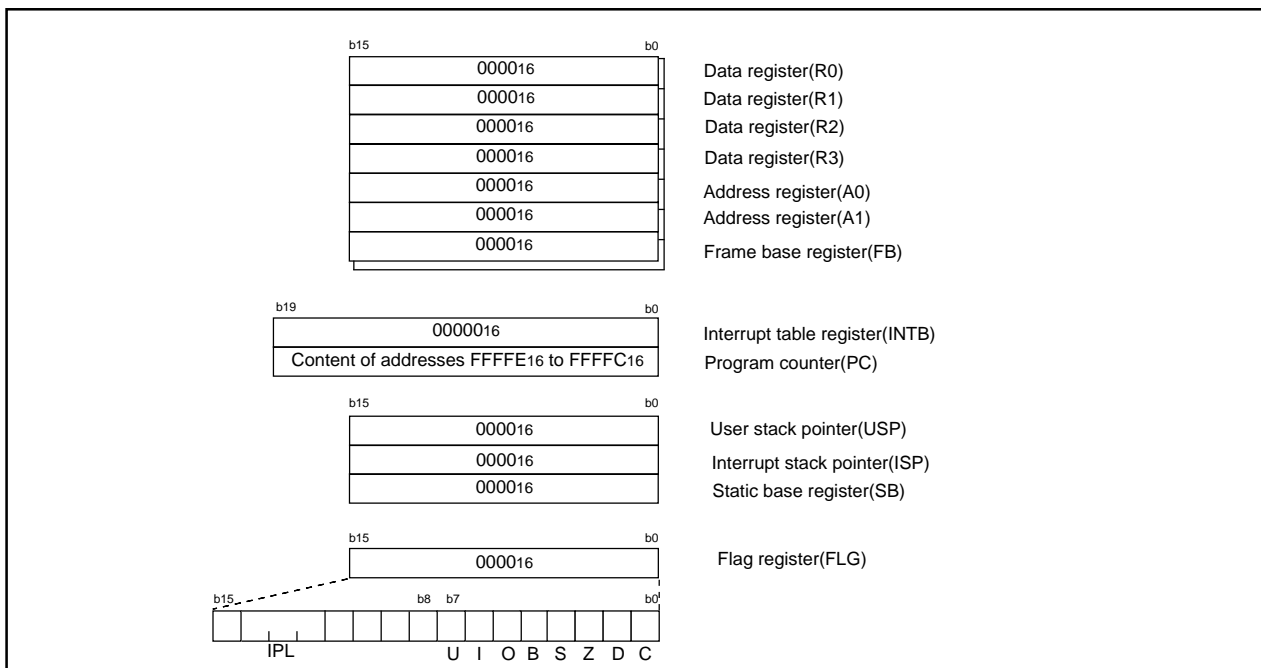
Figure 1.5.2. Reset Sequence

Reset

**Table 1.5.1. Pin Status When RESET Pin Level is “L”**

Pin name	Status		
	CNVss = Vss	CNVss = Vcc1	
		BYTE = Vss	BYTE = Vcc
P0	Input port	Data input	Data input
P1	Input port	Data input	Input port
P2, P3, P40 to P43	Input port	Address output (undefined)	Address output (undefined)
P44	Input port	$\overline{CS0}$ output (“H” is output)	$\overline{CS0}$ output (“H” is output)
P45 to P47	Input port	Input port (Pulled high)	Input port (Pulled high)
P50	Input port	$\overline{WR}$ output (“H” is output)	$\overline{WR}$ output (“H” is output)
P51	Input port	$\overline{BHE}$ output (undefined)	$\overline{BHE}$ output (undefined)
P52	Input port	$\overline{RD}$ output (“H” is output)	$\overline{RD}$ output (“H” is output)
P53	Input port	BCLK output	BCLK output
P54	Input port	$\overline{HLDA}$ output (The output value depends on the input to the HOLD pin)	$\overline{HLDA}$ output (The output value depends on the input to the HOLD pin)
P55	Input port	$\overline{HOLD}$ input	$\overline{HOLD}$ input
P56	Input port	ALE output (“L” is output)	ALE output (“L” is output)
P57	Input port	$\overline{RDY}$ input	$\overline{RDY}$ input
P6, P7, P80 to P84, P86, P87, P9, P10	Input port	Input port	Input port
P11, P12, P13, P140, P141 (Note)	Input port	Input port	Input port

Note : P11, P12, P13, P140, P141 pins exist in 128-pin version.



**Figure 1.5.3. CPU Register Status After Reset**

Reset

**Voltage Detection Circuit**

The voltage detection circuit has circuits to monitor the input voltage at the VCC1 pin, each checking the input voltage with respect to Vdet2, Vdet3, and Vdet4, respectively. Use the VC25 to VC27 bits in the VCR2 register to select whether or not to enable these circuits.

Enable the RAM retention limit detection circuit when using hardware reset 2 in stop mode, or when using the WDC5 bit in the WDC register. The WDC5 bit indicates that the RAM is retained.

Use the reset level detection circuit for hardware reset 2.

The power supply down detection circuit can be set to detect whether the input voltage is equal to or greater than Vdet4 or less than Vdet4 by using the VC13 bit in the VCR1 register. Furthermore, a power supply down detection interrupt can be used.

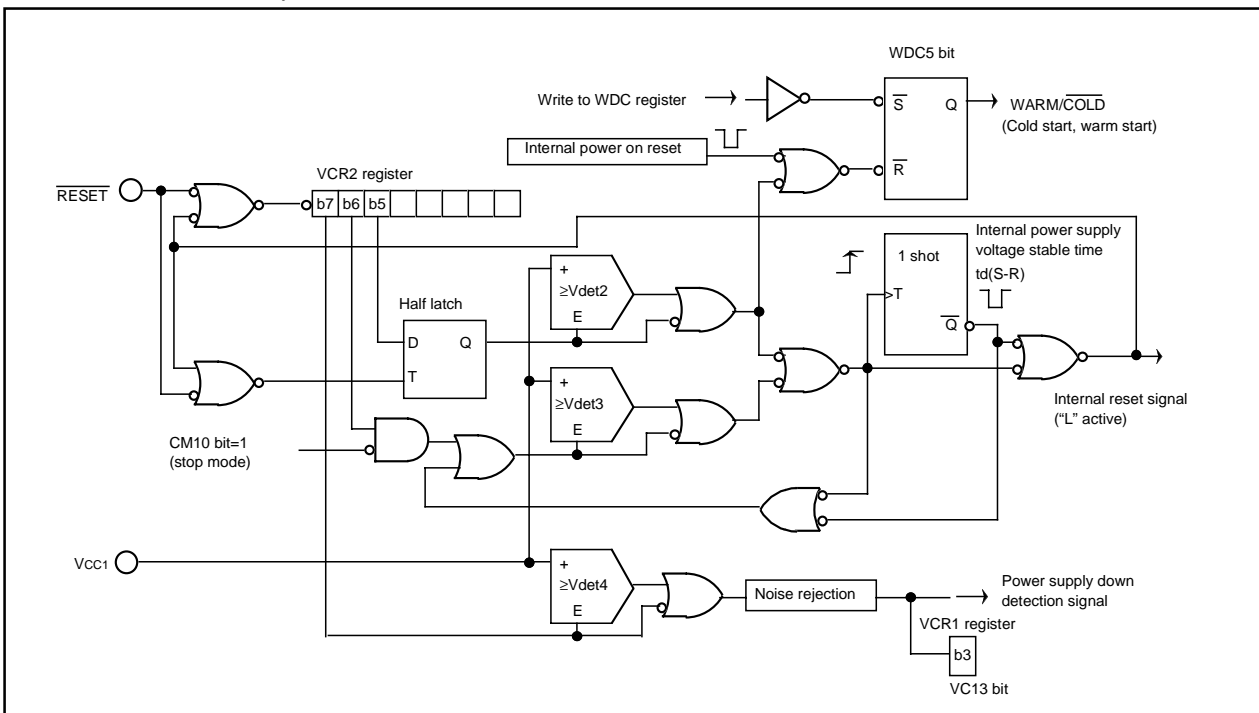


Figure 1.5.4. Reset Circuit Block

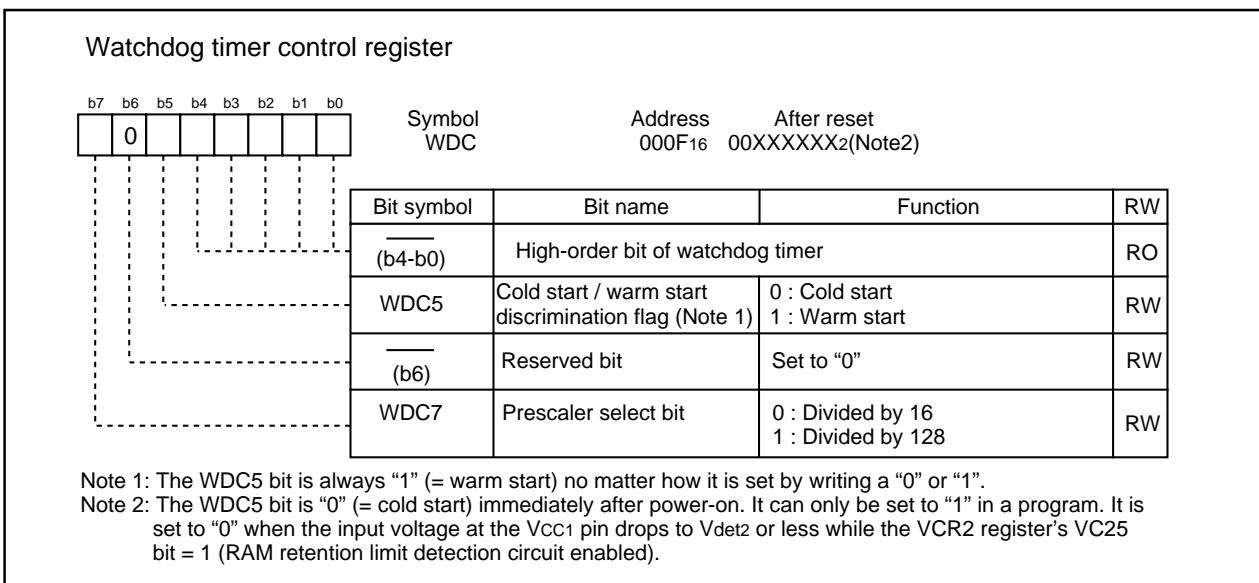


Figure 1.5.5. WDC Register



Reset

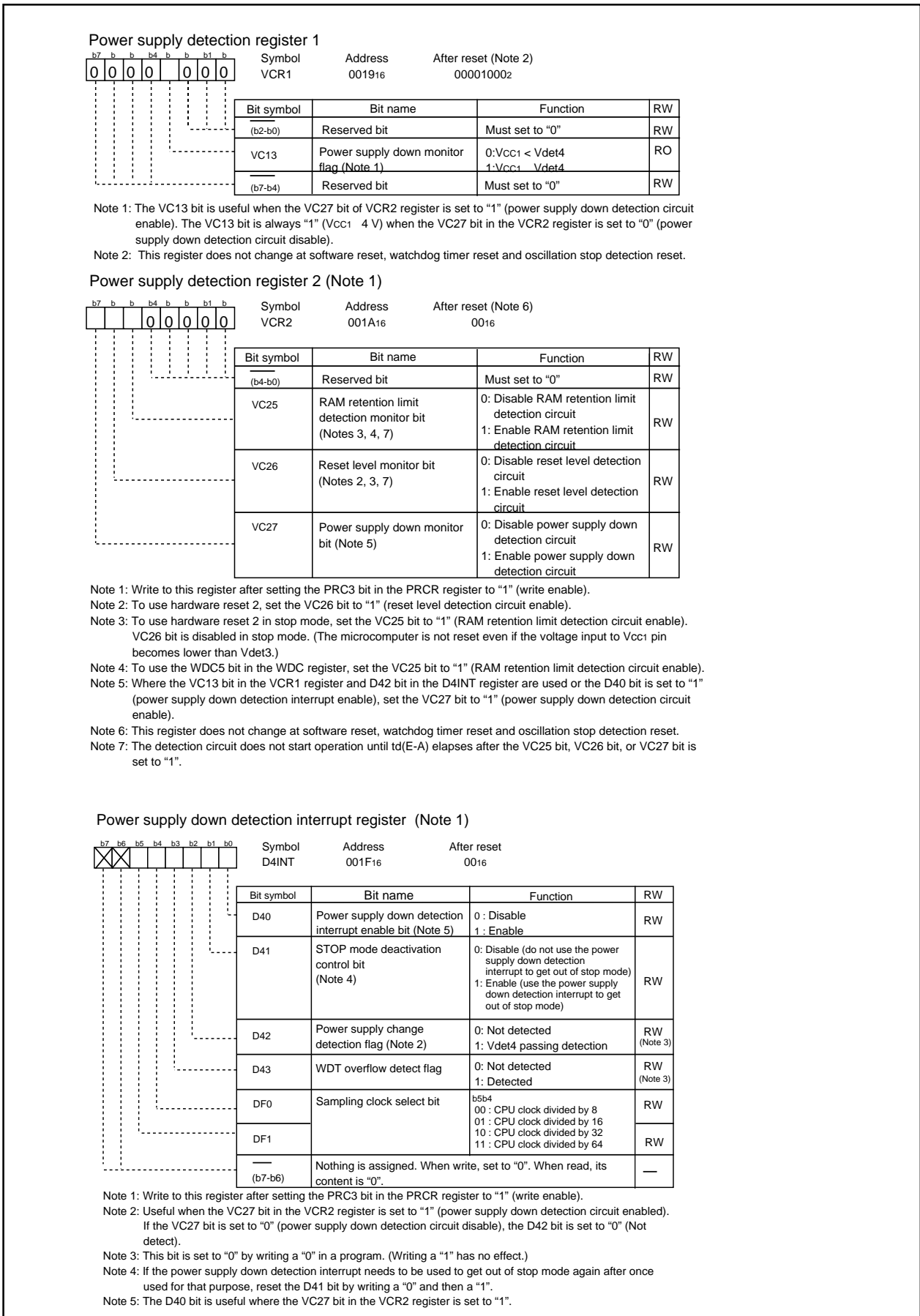
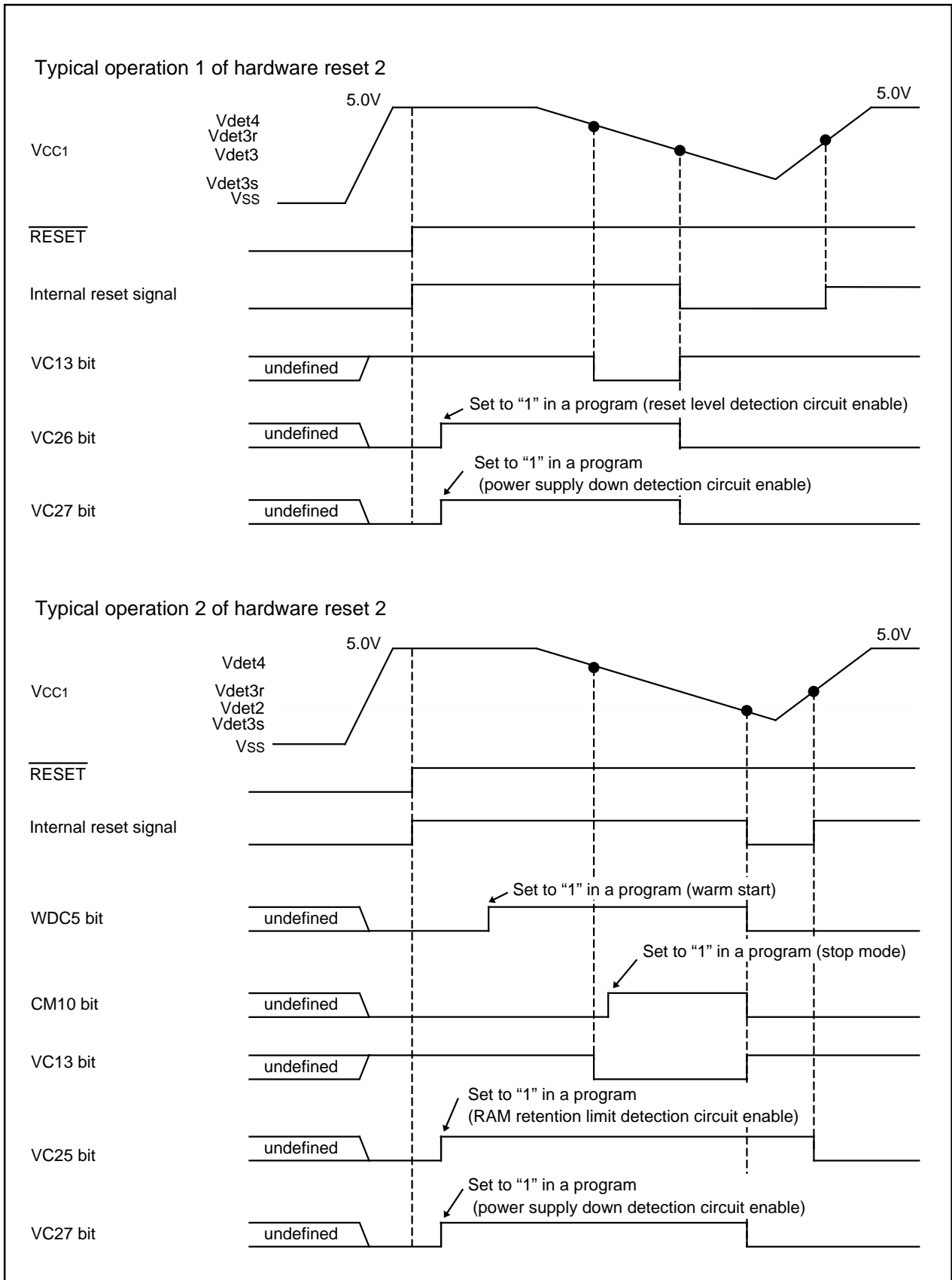


Figure 1.5.6. VCR1 Register, VCR2 Register, and D4INT Register

**Reset**



**Figure 1.5.7. Typical Operation of Hardware Reset 2**

## Power Supply Down Detection Interrupt

A power supply down detection interrupt request is generated when the input voltage at the VCC1 pin rises to Vdet4 or more or drops below Vdet4 while the D40 bit in the D4INT register is set to "1" (power supply down detection interrupt enable). The power supply down detection interrupt shares the interrupt vector with the watchdog timer interrupt and oscillation stop, re-oscillation detection interrupt.

To use the power supply down detection interrupt to get out of stop mode, set the D41 bit in the D4INT register to "1" (enable).

The D42 bit in the D4INT register becomes "1" when passing through Vdet4 is detected after the voltage inputted to the VCC1 pin is up or down.

A power supply down detection interrupt is generated when the D42 bit changes state from "0" to "1". The D42 bit needs to be set to "0" in a program. However, where the D41 bit is "1" and the stop mode is selected, the power supply down detection interrupt request arises, and the microcomputer is reset from the stop mode with no regard for the status of D42 bit if it is detected that the voltage applied to the VCC1 pin has increased, passing through Vdet4.

Table 1.5.2 shows the power supply down detection interrupt request generation conditions.

It is possible to set the sampling clock detecting that the voltage applied to the VCC1 pin has passed through Vdet4 with the DF1 to DF0 bits of D4INT register. Table 1.5.3 shows sampling clock periods.

**Table 1.5.2. Power Supply Down Detection Interrupt Request Generation Conditions**

Bit, Vdet4 passing detection, operation mode condition							Power supply down detection Interrupt request
VC27 bit	D40 bit	Vdet4 passing detection	D42 bit	D41 bit	VC13 bit	Operation mode (Notes 1, 2)	
0	—	—	—	—	—	—	Not generated
1	0	—	—	—	—	—	
		1	Not detected	—	—	—	—
	Detected		From 0 to 1	0	—	Normal, wait	Generated
				—	—	Stop	Not generated
	1		Detected	From 0 to 1 (Up)	1	—	—
		0			—	—	Not generated
1	Detected	From 1 to 1 (No change)	1	From 0 to 1 (Up)	Normal, wait	Generated	
				From 1 to 0 (Down)	—	Not generated	

Note 1: The status except the wait mode and stop mode is handled as the normal mode. (Refer to "Clock generating circuit")

Note 2: Refer to "Limitations on stop mode", "Limitations on wait mode".

**Table 1.5.3. Sampling Clock Periods**

CPU clock (MHz)	Sampling clock ( $\mu$ s)			
	divided by 8	divided by 16	divided by 32	divided by 64
16	1.5	3.0	6.0	12.0

## Precautions

### 1. Limitations on Stop Mode

If the CM10 bit in the CM1 register is set to "1" (stop mode) when the VC13 bit in the VCR1 register is "1" ( $V_{CC1} \geq V_{det4}$ ) while the VC27 bit in the VCR2 register is "1" (power supply down detection circuit enable) and the D40 bit in the D4INT register is "1" (power supply down detection interrupt enable) and D41 bit in the D4INT register is "1" (power supply down detection interrupt is used to get out of stop mode), a power supply down detection interrupt is immediately generated, causing the microcomputer to exit stop mode. In systems where the microcomputer enters stop mode when the input voltage at the VCC1 pin drops below Vdet4 and exits stop mode when the input voltage rises to Vdet4 or more, make sure the CM10 bit is set to "1" when VC13 bit is "0" ( $V_{CC1} < V_{det4}$ ).

## 2. Limitations on WAIT Instruction

If the WAIT instruction is executed when the VC13 bit in the VCR1 register is "1" ( $V_{CC1} \geq V_{det4}$ ) while the VC27 bit in the VCR2 register is "1" (power supply down detection circuit enable) and the D40 bit in the D4INT register is "1" (power supply down detection interrupt enable), a power supply down detection interrupt is immediately generated, causing the microcomputer to exit wait mode.

In systems where the microcomputer enters wait mode when the input voltage at the VCC1 pin drops below  $V_{det4}$  and exits wait mode when the input voltage rises to  $V_{det4}$  or more, make sure the WAIT instruction is executed when VC13 bit is "0" ( $V_{CC1} < V_{det4}$ ).

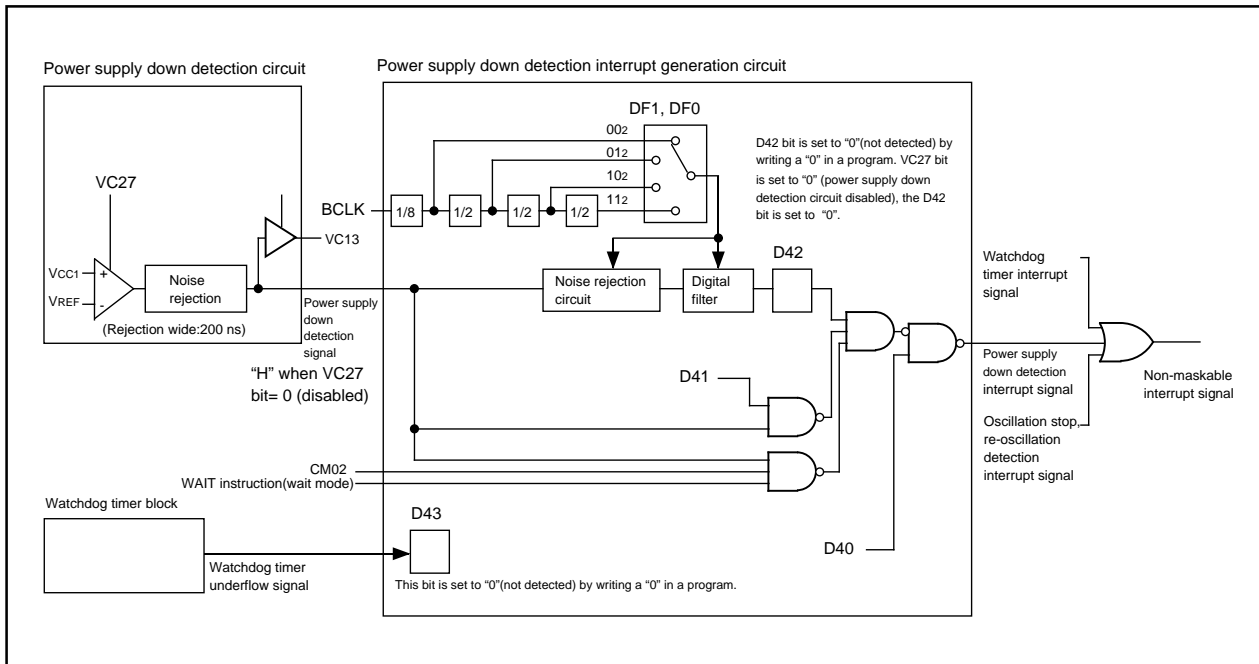


Figure 1.5.8. Power Supply Down Detection Interrupt Generation Block

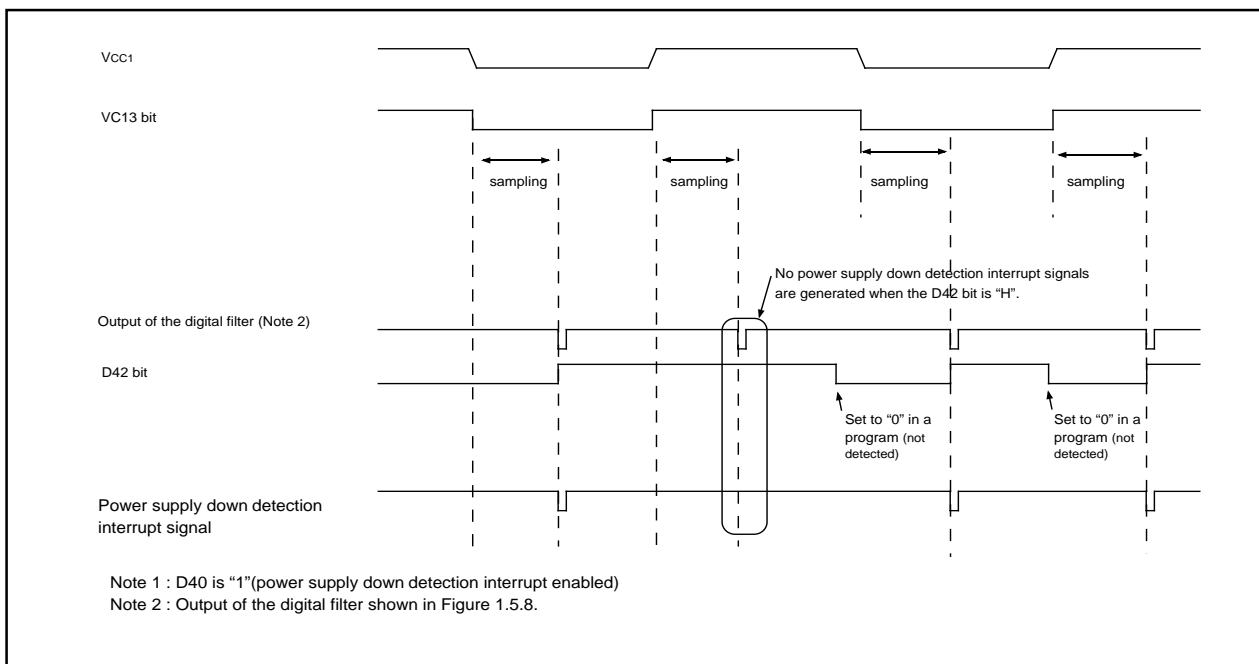


Figure 1.5.9. Power Supply Down Detection Interrupt Generation Circuit Operation Example

# Processor Mode

## (1) Types of Processor Mode

Three processor modes are available to choose from: single-chip mode, memory expansion mode, and microprocessor mode. Table 1.6.1 shows the features of these processor modes.

**Table 1.6.1. 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 (Note)	Some pins serve as bus control pins (Note)
Microprocessor mode	SFR, internal RAM, external area (Note)	Some pins serve as bus control pins (Note)

Note : Refer to "Bus".

## (2) Setting Processor Modes

Processor mode is set by using the CNVss pin and the PM01 to PM00 bits in the PM0 register.

Table 1.6.2 shows the processor mode after hardware reset. Table 1.6.3 shows the PM01 to PM00 bit set values and processor modes.

**Table 1.6.2. Processor Mode After Hardware Reset**

CNVss pin input level	Processor mode
Vss	Single-chip mode
Vcc1 (Note 1, Note 2)	Microprocessor mode

Note 1: If the microcomputer is reset in hardware by applying Vcc1 to the CNVSS pin (hardware reset 1 or hardware reset 2), the internal ROM cannot be accessed regardless of PM10 to PM00 bits.

Note 2: The multiplexed bus cannot be assigned to the entire CS space.

**Table 1.6.3. PM01 to PM00 Bits Set Values and Processor Modes**

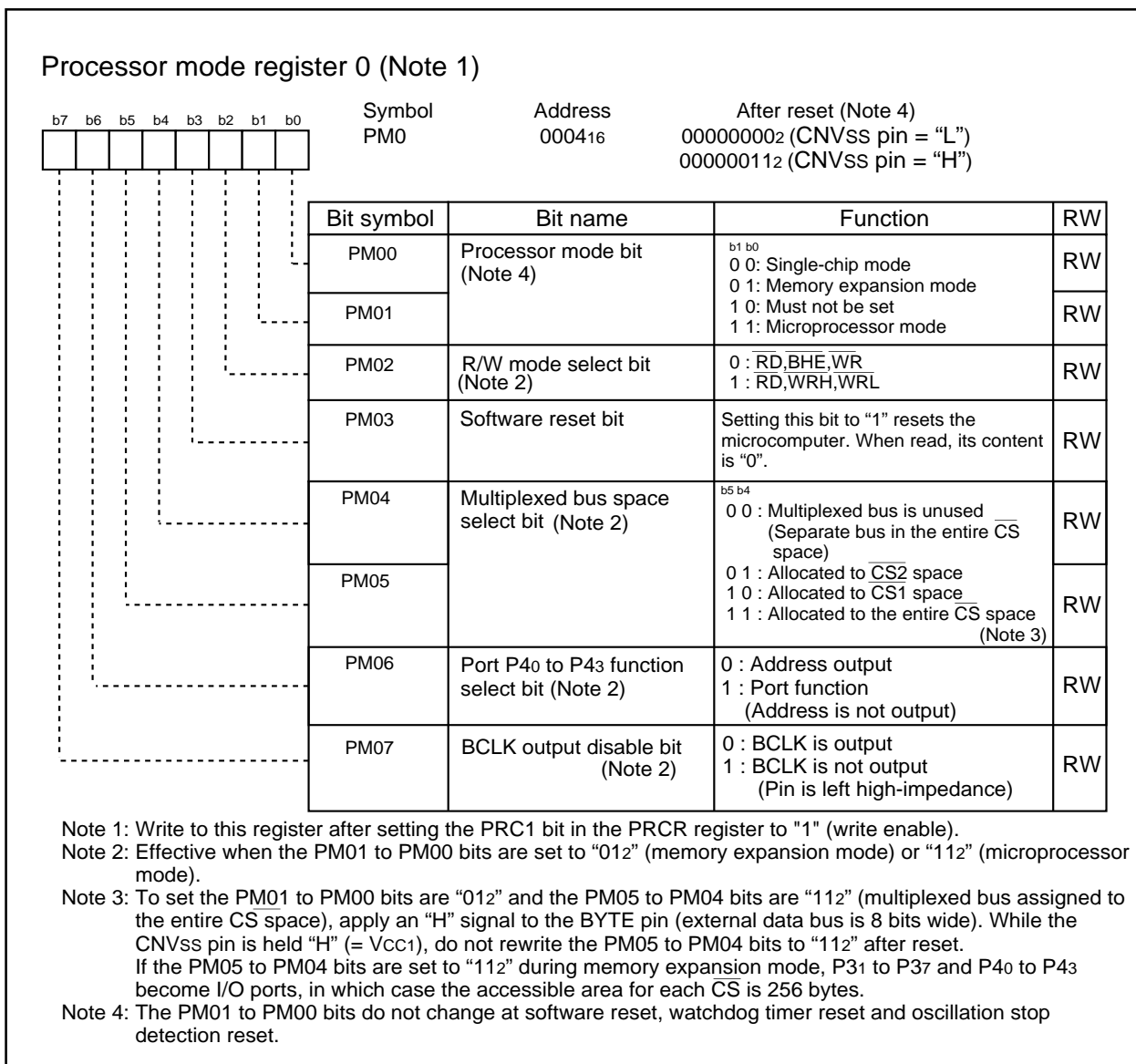
PM01 to PM00 bits	Processor modes
002	Single-chip mode
012	Memory expansion mode
102	Must not be set
112	Microprocessor mode

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 "012" (memory expansion mode) or "112" (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 hardware reset 2), the internal ROM cannot be accessed regardless of PM01 to PM00 bits.

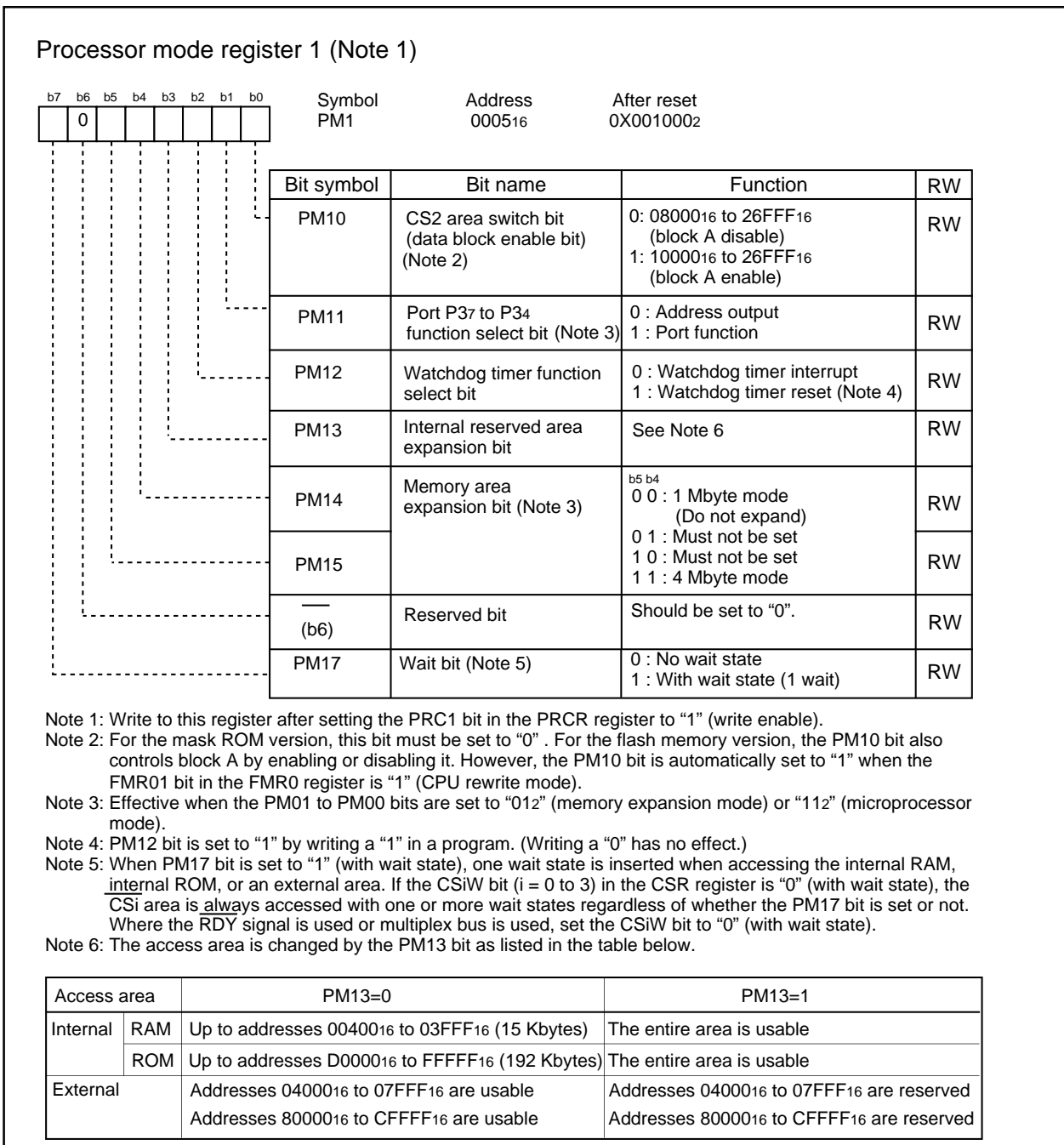
Figures 1.6.1 and 1.6.2 show the registers associated with processor modes. Figure 1.6.3 show the memory map in single chip mode.

**Processor Mode**



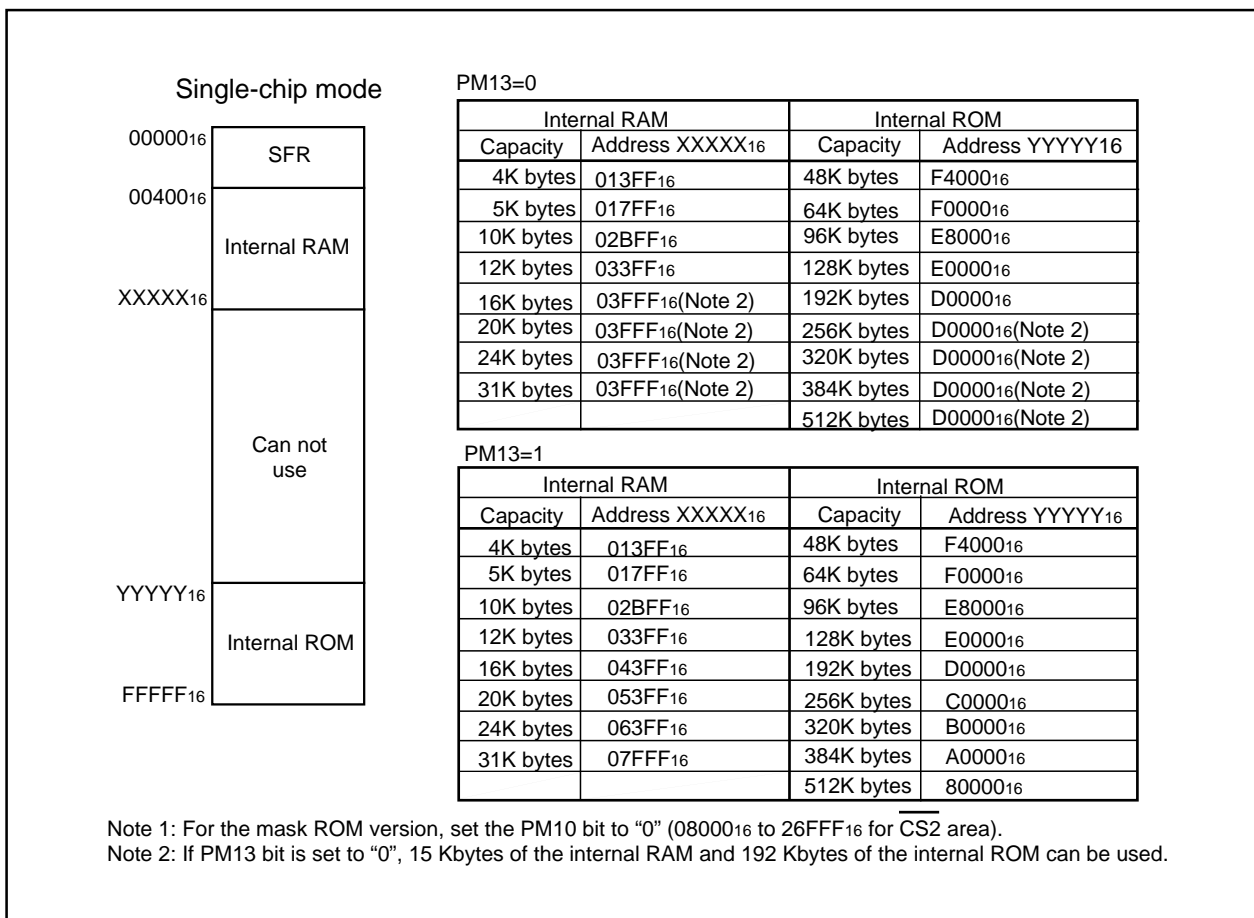
**Figure 1.6.1. PM0 Register**

**Processor Mode**



**Figure 1.6.2. PM1 Register**

**Processor Mode**



**Figure 1.6.3. Memory Map in Single Chip Mode**



## 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  $\overline{CS3}$ ,  $\overline{RD}$ ,  $\overline{WRL/WR}$ ,  $\overline{WRH/BHE}$ ,  $\overline{ALE}$ ,  $\overline{RDY}$ ,  $\overline{HOLD}$ ,  $\overline{HLDA}$  and  $\overline{BCLK}$ .

### Bus Mode

The bus mode, either multiplexed or separate, can be selected using the PM05 to PM04 bits.

#### Separate Bus

In this bus mode, data and address are separate.

#### Multiplexed Bus

In this bus mode, data and address are multiplexed. If the data bus is 8 bits wide, D0 to D7 and A0 to A7 are multiplexed. If the data bus is 16 bits wide, D0 to D7 and A1 to A8 are multiplexed, with D8 to D15 not multiplexed. In this case, external devices connecting to the multiplexed bus are mapped to the even addresses of the microcomputer.

## Bus Control

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

### (1) Address Bus

The address bus consists of 20 lines, A0 to A19. The address bus width can be chosen to be 12, 16 or 20 bits by using the PM06 bit in the PM0 register and the PM11 bit in the PM1 register. Table 1.7.1 shows the PM06 and PM11 bit set values and address bus widths.

**Table 1.7.1. PM06 and PM11 Bits Set Value and Address Bus Width**

Set value(Note)	Pin function	Address bus wide
PM11=1 PM06=1	P34 to P37 P40 to P43	12 bits
PM11=0 PM06=1	A12 to A15 P40 to P43	16 bits
PM11=0 PM06=0	A12 to A15 A16 to A19	20 bits

Note 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.

### (2) Data Bus

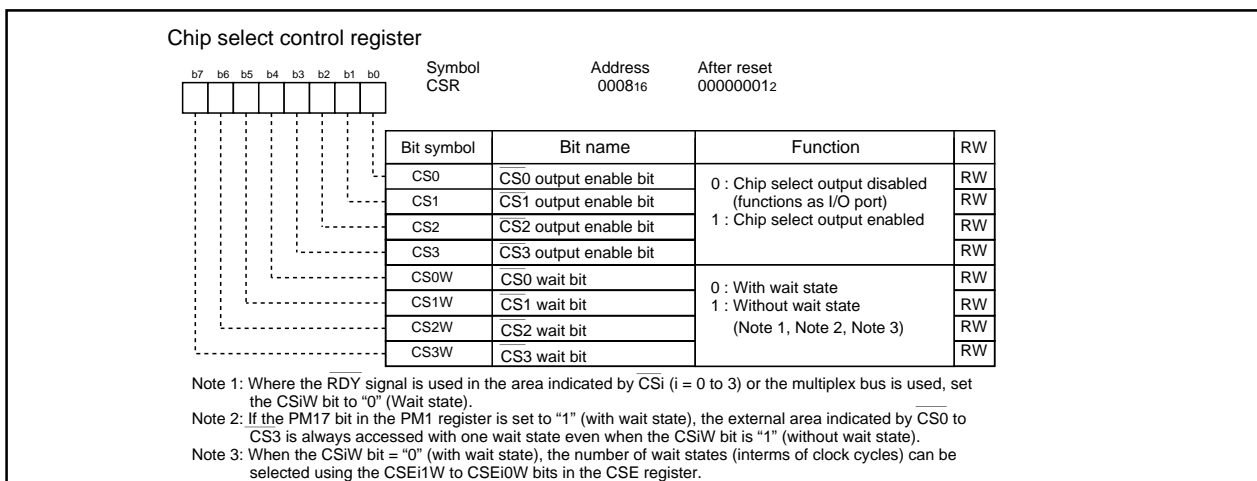
When input on the BYTE pin is high, 8 lines D0 to D7 comprise the data bus; when input on the BYTE pin is low, 16 lines D0 to D15 comprise the data bus.

Do not change the input level on the BYTE pin while in operation.

### (3) Chip Select Signal

The chip select (hereafter referred to as the  $\overline{CS}_i$ ) signals are output from the  $\overline{CS}_i$  ( $i = 0$  to 3) pins. These pins can be chosen to function as I/O ports or as  $\overline{CS}$  by using the CSi bit in the CSR register. Figure 1.7.1 shows the CSR register.

During 1 Mbyte mode, the external area can be separated into up to 4 by the  $\overline{CS}_i$  signal which is output from the  $\overline{CS}_i$  pin. During 4 Mbyte mode,  $\overline{CS}_i$  signal or bank number is output from the  $\overline{CS}_i$  pin. Refer to "Memory space expansion function". Figure 1.7.2 shows the example of address bus and  $\overline{CS}_i$  signal output in 1 Mbyte mode.



**Figure 1.7.1. CSR Register**

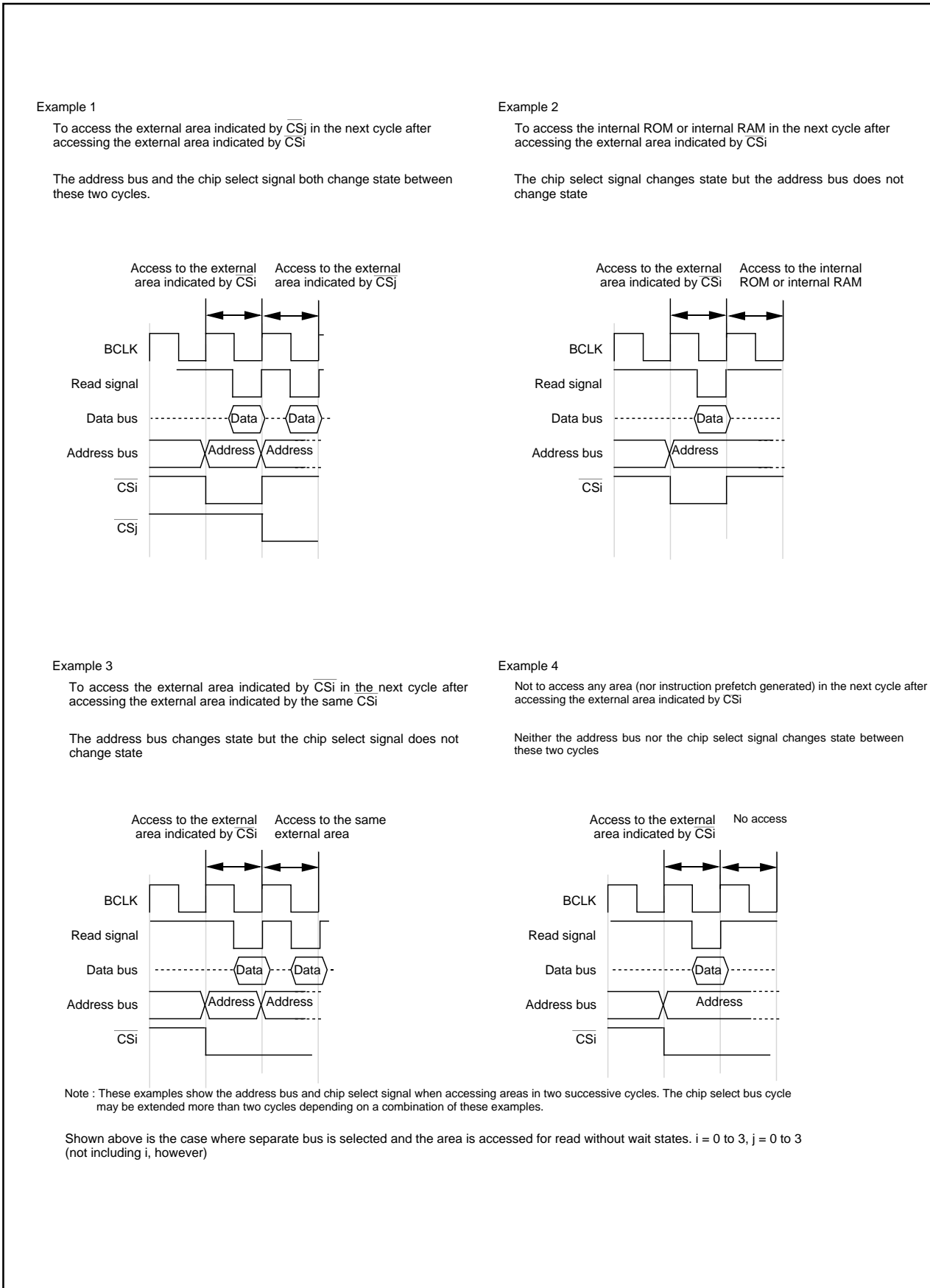


Figure 1.7.2. Example of Address Bus and  $\overline{CS}_i$  Signal Output in 1 Mbyte Mode

#### (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 1.7.2 shows the operation of  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  signals. Table 1.7.3 shows the operation of operation of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$  signals.

**Table 1.7.2. 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

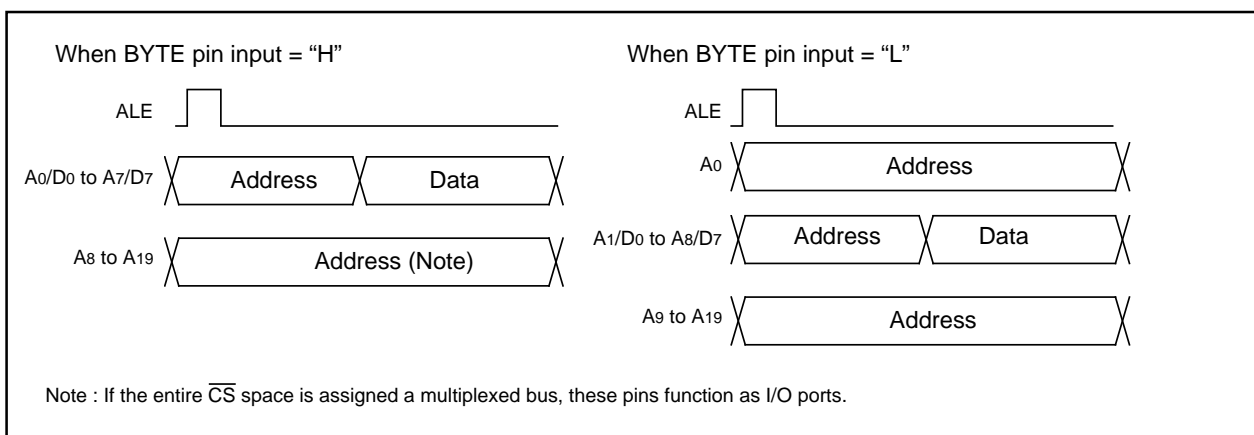
**Table 1.7.3. Operation of  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{BHE}$  Signals**

Data bus width	$\overline{RD}$	$\overline{WR}$	$\overline{BHE}$	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	— (Note)	H or L	Write 1 byte of data
	L	H	— (Note)	H or L	Read 1 byte of data

Note : Do not use.

#### (5) ALE Signal

The ALE signal latches the address when accessing the multiplex bus space. Latch the address when the ALE signal falls.



**Figure 1.7.3. ALE Signal, Address Bus, Data Bus**

### (6) The RDY Signal

This signal is provided for accessing external devices which need to be accessed at low speed. If input on the 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 RDY signal was acknowledged.

A0 to A19, D0 to D15, CS0 to CS3, RD, WRL, WRH, WR, BHE, ALE, HLDA

Then, when the input on the RDY pin is detected high at the falling edge of BCLK, the remaining bus cycle is executed. Figure 1.7.4 shows example in which the wait state was inserted into the read cycle by the RDY signal. To use the RDY signal, set the corresponding bit (CS3W to CS0W bits) in the CSR register to "0" (with wait state). When not using the RDY signal, process the RDY pin as an unused pin.

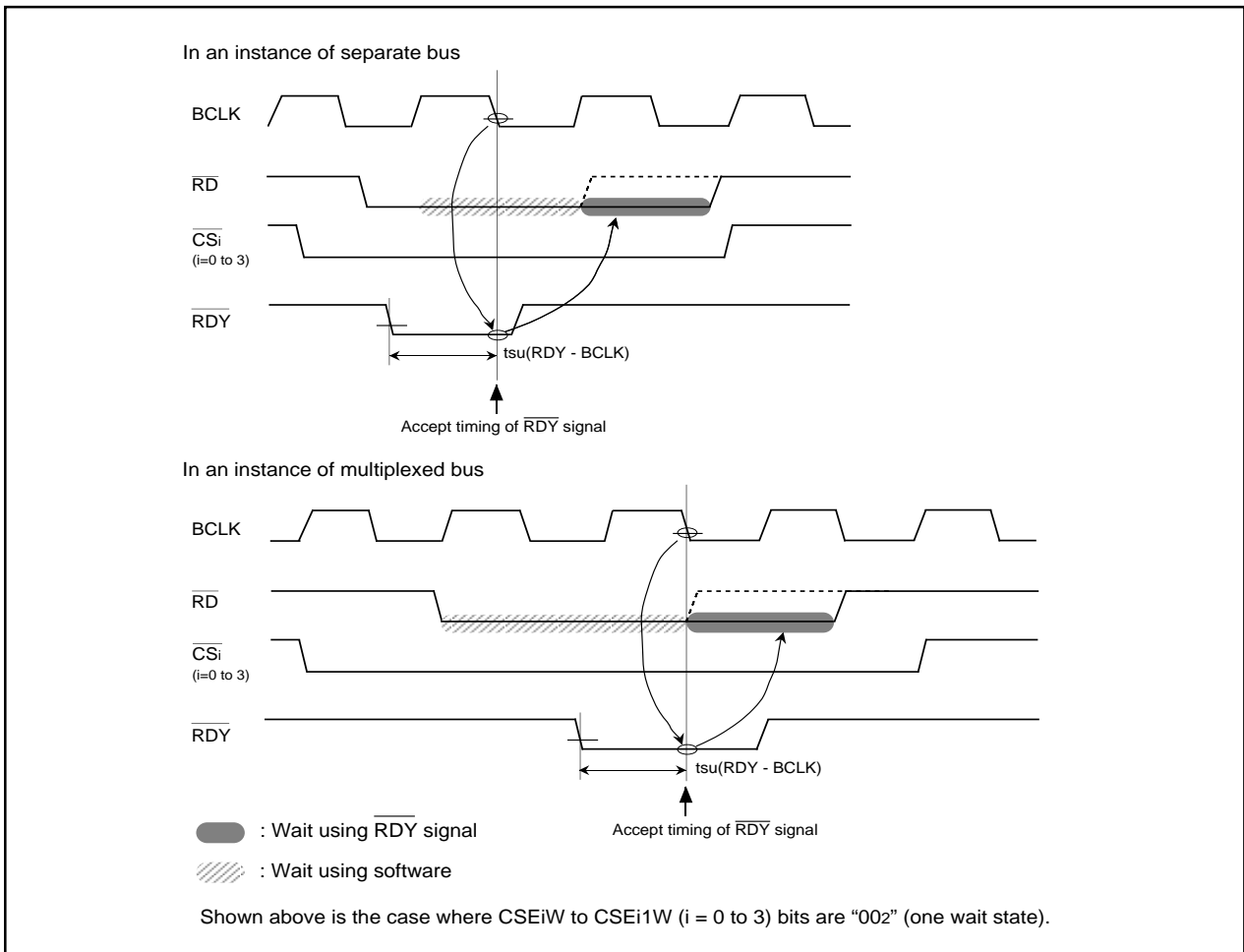


Figure 1.7.4. Example in which Wait State was Inserted into Read Cycle by RDY Signal

## (7) Hold Signal

This signal is used to transfer control of the bus from the CPU or DMA to an external circuit. When input on the  $\overline{\text{HOLD}}$  pin is asserted "L", the microcomputer goes to a hold state after completing the bus access then in progress. While the  $\overline{\text{HOLD}}$  pin is held "L", the microcomputer remains in a hold state, outputting a low signal from the  $\overline{\text{HLDA}}$  pin.

Table 1.7.4 shows the microcomputer status in the hold state.

Bus-using priorities are given to  $\overline{\text{HOLD}}$ , DMAC, and CPU in order of decreasing precedence.

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

Figure 1.7.5. Bus-using Priorities

Table 1.7.4. Microcomputer Status in Hold State

Item		Status
BCLK		Output
$A_0$ to $A_{19}$ , $D_0$ to $D_{15}$ , $\overline{\text{CS}}_0$ to $\overline{\text{CS}}_3$ , $\overline{\text{RD}}$ , $\overline{\text{WRL}}$ , $\overline{\text{WRH}}$ , $\overline{\text{WR}}$ , $\overline{\text{BHE}}$		High-impedance
I/O ports	P0, P1, P3, P4(Note 2)	High-impedance
	P6 to P14(Note 1)	Maintains status when hold signal is received
$\overline{\text{HLDA}}$		Output "L"
Internal peripheral circuits		ON (but watchdog timer stops)
ALE signal		Undefined

Note 1: P11 to P14 are included in the 128-pin version.

Note 2: When I/O port function is selected.

## (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 "CPU clock and peripheral clock".

Table 1.7.5. Pin Functions for Each Processor Mode

Processor mode	Memory expansion mode or microprocessor mode				Memory expansion mode
PM05–PM04 bits	002(separate bus)		012(CS2 is for multiplexed bus and others are for separate bus) 102(CS1 is for multiplexed bus and others are for separate bus)		112(multiplexed bus for the entire space) (Note 1)
Data bus width BYTE pin	8 bits “H”	16 bits “L”	8 bits “H”	16 bits “L”	8 bits “H”
P00 to P07	D0 to D7	D0 to D7	D0 to D7	D0 to D7	I/O ports
P10 to P17	I/O ports	D8 to D15	I/O ports	D8 to D15	I/O ports
P20	A0	A0	A0/D0(Note 2)	A0	A0/D0
P21 to P27	A1 to A7	A1 to A7	A1 to A7/D1 to D7 (Note 2)	A1 to A7/D0 to D6 (Note 2)	A1 to A7/D1 to D7
P30	A8	A8	A8	A8/D7(Note 2)	A8
P31 to P33	A9 to A11				I/O ports
P34 to P37	PM11=0	A12 to A15			I/O ports
	PM11=1	I/O ports			
P40 to P43	PM06=0	A16 to A19			I/O ports
	PM06=1	I/O ports			
P44	CS0=0	I/O ports			
	CS0=1	$\overline{\text{CS0}}$			
P45	CS1=0	I/O ports			
	CS1=1	$\overline{\text{CS1}}$			
P46	CS2=0	I/O ports			
	CS2=1	$\overline{\text{CS2}}$			
P47	CS3=0	I/O ports			
	CS3=1	$\overline{\text{CS3}}$			
P50	PM02=0	WR			
	PM02=1	— (Note 3)	WRL	— (Note 3)	WRL
P51	PM02=0	BHE			
	PM02=1	— (Note 3)	WRH	— (Note 3)	WRH
P52	$\overline{\text{RD}}$				
P53	BCLK				
P54	$\overline{\text{HLDA}}$				
P55	HOLD				
P56	ALE				
P57	$\overline{\text{RDY}}$				

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

Note 1: To set the PM01 to PM00 bits are set to “012” and the PM05 to PM04 bits are set to “112” (multiplexed bus assigned to the entire CS space), apply “H” to the BYTE pin (external data bus 8 bits wide). While the CNVss pin is held “H” (= Vcc1), do not rewrite the PM05 to PM04 bits to “112” after reset. If the PM05 to PM04 bits are set to “112” during memory expansion mode, P31 to P37 and P40 to P43 become I/O ports, in which case the accessible area for each CS is 256 bytes.

Note 2: In separate bus mode, these pins serve as the address bus.

Note 3: If the data bus is 8 bits wide, make sure the PM02 bit is set to “0” ( $\overline{\text{RD}}$ , BHE, WR).

### (9) External Bus Status When Internal Area Accessed

Table 1.7.6 shows the external bus status when the internal area is accessed.

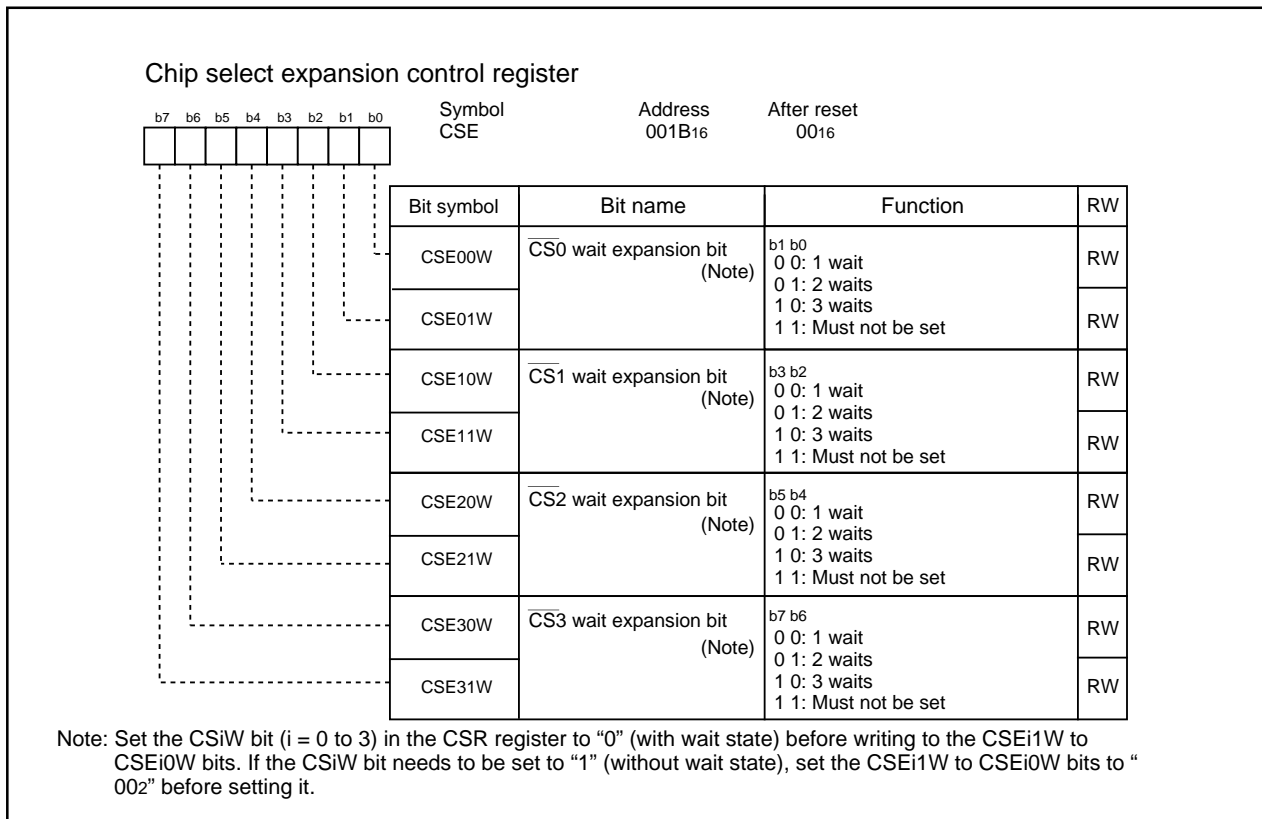
**Table 1.7.6. External Bus Status When Internal Area Accessed**

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
	When write	Output data
$\overline{RD}$ , $\overline{WR}$ , $\overline{WRL}$ , $\overline{WRH}$	$\overline{RD}$ , $\overline{WR}$ , $\overline{WRL}$ , $\overline{WRH}$ 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"

### (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.

To use the  $\overline{RDY}$  signal, set the corresponding CS3W to CS0W bit to "0". Figure 1.7.6 shows the CSE register. Table 1.7.7 shows the software wait related bits and bus cycles. Figure 1.7.7 and 1.7.8 show the typical bus timings using software wait.



**Figure 1.7.6. CSE Register**



Table 1.7.7. Bit and Bus Cycle Related to Software Wait

Area	Bus mode	PM1 register PM17 bit	CSR register CS3W bit (Note 1) CS2W bit (Note 1) CS1W bit (Note 1) CS0W bit (Note 1)	CSE register CSE31W to CSE30W bit CSE21W to CSE20W bit CSE11W to CSE10W bit CSE01W to CSE00W bit	Software wait	Bus cycle
Internal RAM, ROM	—	0	—	—	No wait	1 BCLK cycle (Note 3)
	—	1	—	—	1 wait	2 BCLK cycles
External area	Separate bus	0	1	002	No wait	1 BCLK cycle (read) 2 BCLK cycles (write)
		—	0	002	1 wait	2 BCLK cycles (Note 3)
		—	0	012	2 waits	3 BCLK cycles
		—	0	102	3 waits	4 BCLK cycles
		1	1	002	1 wait	2 BCLK cycles
	Multiplexed bus (Note 2)	—	0	002	1 wait	3 BCLK cycles
		—	0	012	2 waits	3 BCLK cycles
		—	0	102	3 waits	4 BCLK cycles
		—	0	002	1 wait	3 BCLK cycles
		1	0	002	1 wait	3 BCLK cycles

Note 1: To use the  $\overline{\text{RDY}}$  signal, set this bit to "0".

Note 2: To access in multiplexed bus mode, set the corresponding bit of CS0W to CS3W to "0" (with wait state).

Note 3: 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), and the CSE register is set to "0016" (one wait state for  $\overline{\text{CS0}}$  to  $\overline{\text{CS3}}$ ). Therefore, the internal RAM and internal ROM are accessed with no wait states, and all external areas are accessed with one wait state.

Bus

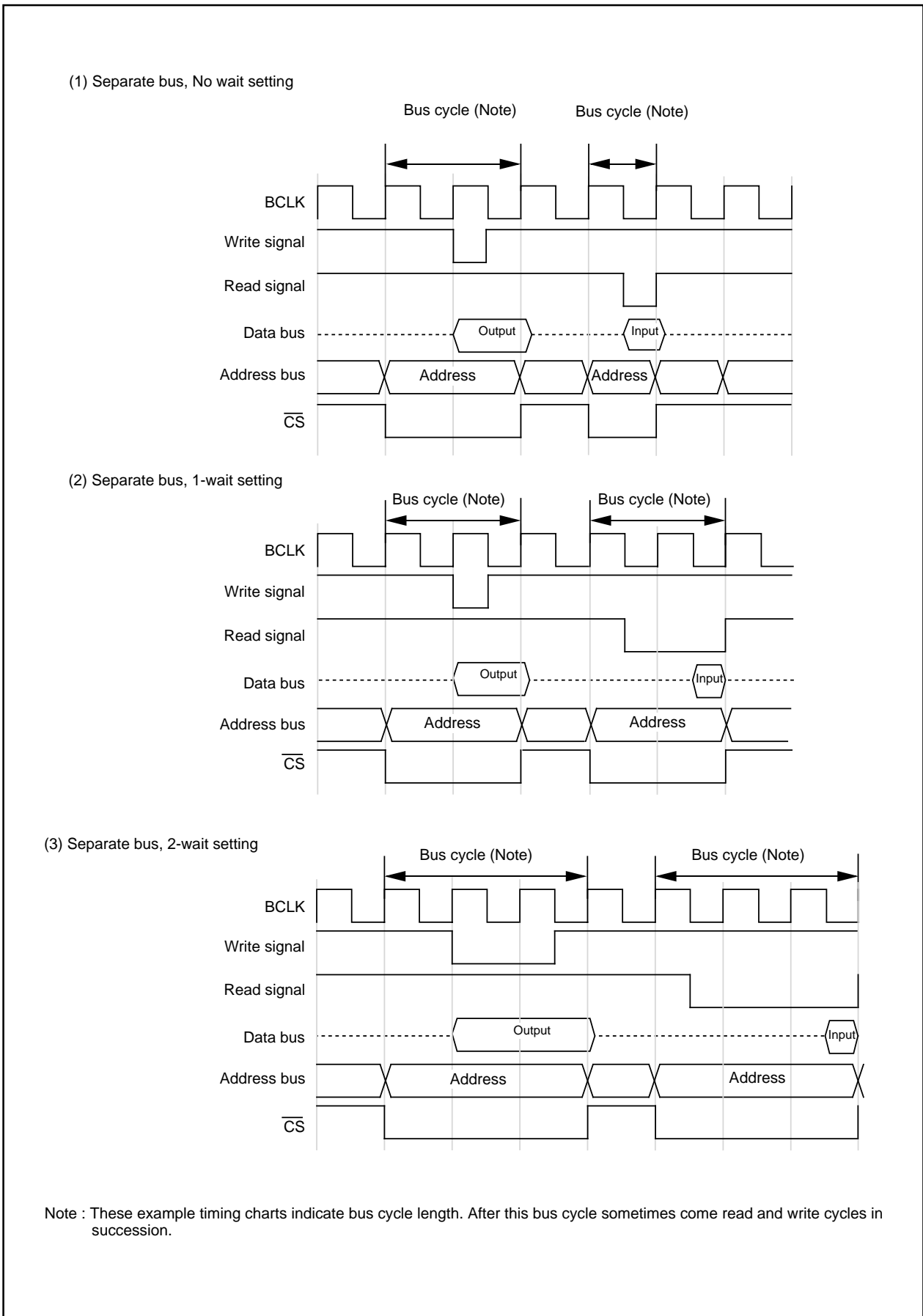


Figure 1.7.7. Typical Bus Timings Using Software Wait (1)

Bus

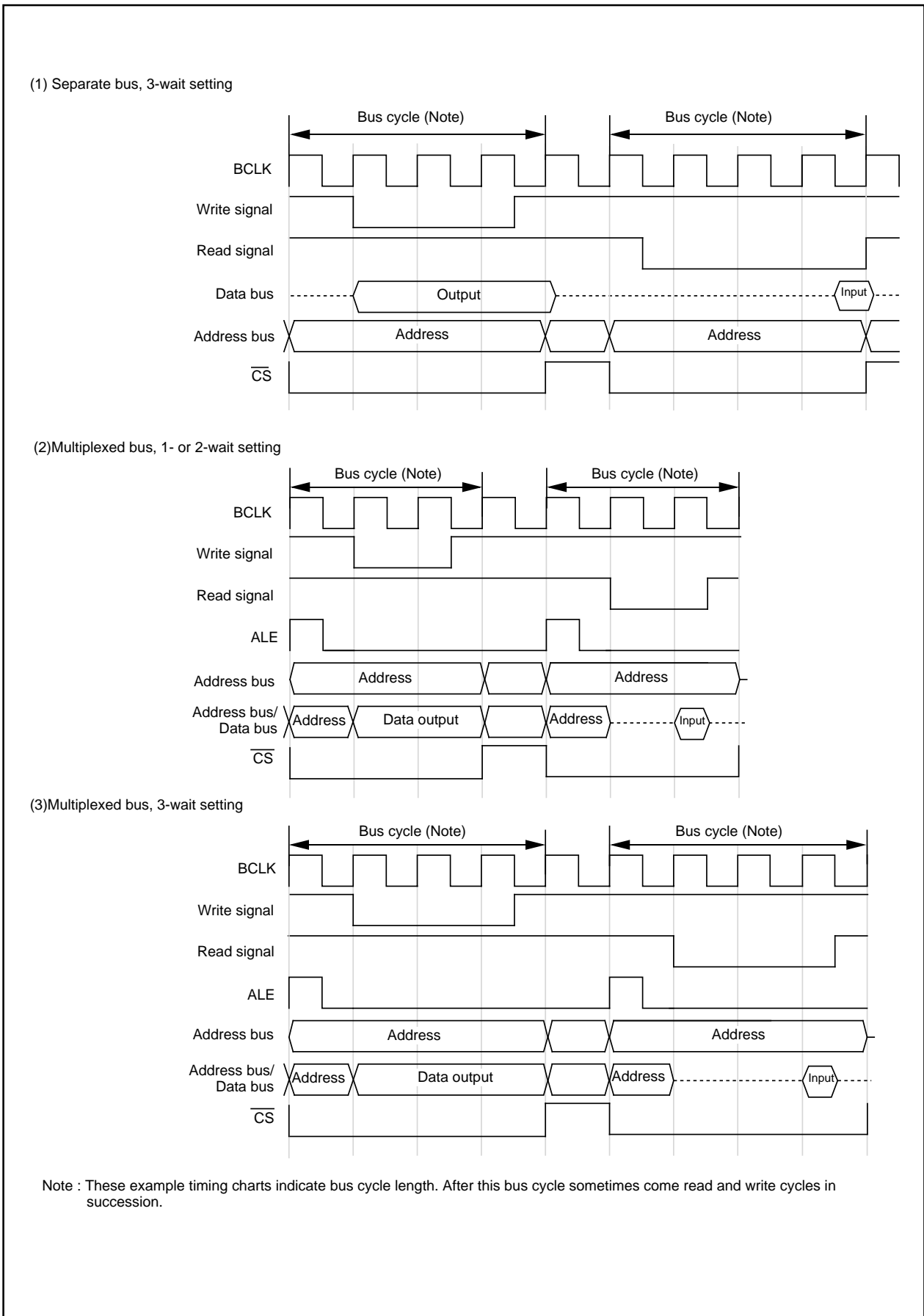


Figure 1.7.8. Typical Bus Timings Using Software Wait (2)

## Memory Space Expansion Function

## 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.

Table 1.8.1 shows the way of setting memory space expansion function, memory spaces.

**Table 1.8.1. The Way of Setting Memory Space Expansion Function, Memory Space**

Memory space expansion function	How to set (PM15 to PM14)	Memory space
1 Mbytes mode	00 <sub>2</sub>	1 Mbytes (no expansion)
4 Mbytes mode	11 <sub>2</sub>	4 Mbytes

### (1) 1 Mbyte Mode

In this mode, the memory space is 1 Mbytes. In 1 Mbyte mode, the external area to be accessed is specified using the  $\overline{CS}_i$  ( $i = 0$  to 3) signals (hereafter referred to as the  $\overline{CS}_i$  area). Figures 1.8.2 to 1.8.3 show the memory mapping and  $\overline{CS}$  area in 1 Mbyte mode.

### (2) 4 Mbyte Mode

In this mode, the memory space is 4 Mbytes. Figure 1.8.1 shows the DBR register. The BSR2 to BSR0 bits select a bank number which is to be accessed to read or write data. Setting the OFS bit to "1" (with offset) allows the accessed address to be offset by 40000<sub>16</sub>.

In 4 Mbyte mode, the  $\overline{CS}_i$  ( $i=0$  to 3) pin functions differently for each area to be accessed.

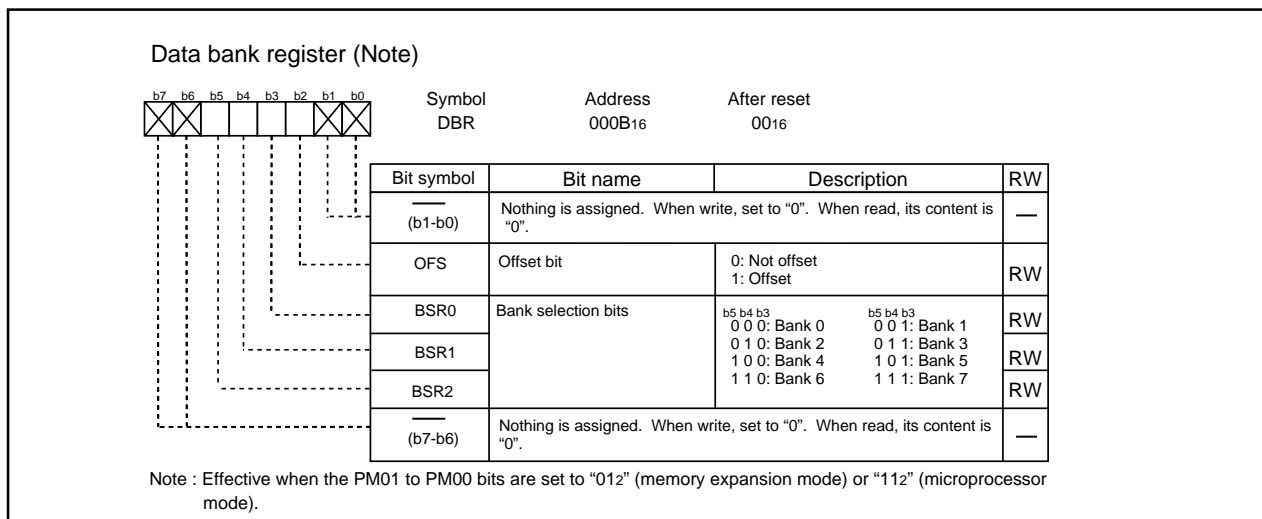
Addresses 04000<sub>16</sub> to 3FFFF<sub>16</sub>, C0000<sub>16</sub> to FFFFF<sub>16</sub>

- The  $\overline{CS}_i$  signal is output from the  $\overline{CS}_i$  pin (same operation as 1 Mbyte mode. However the last address of  $\overline{CS}_1$  area is 3FFFF<sub>16</sub>)

Addresses 40000<sub>16</sub> to BFFFF<sub>16</sub>

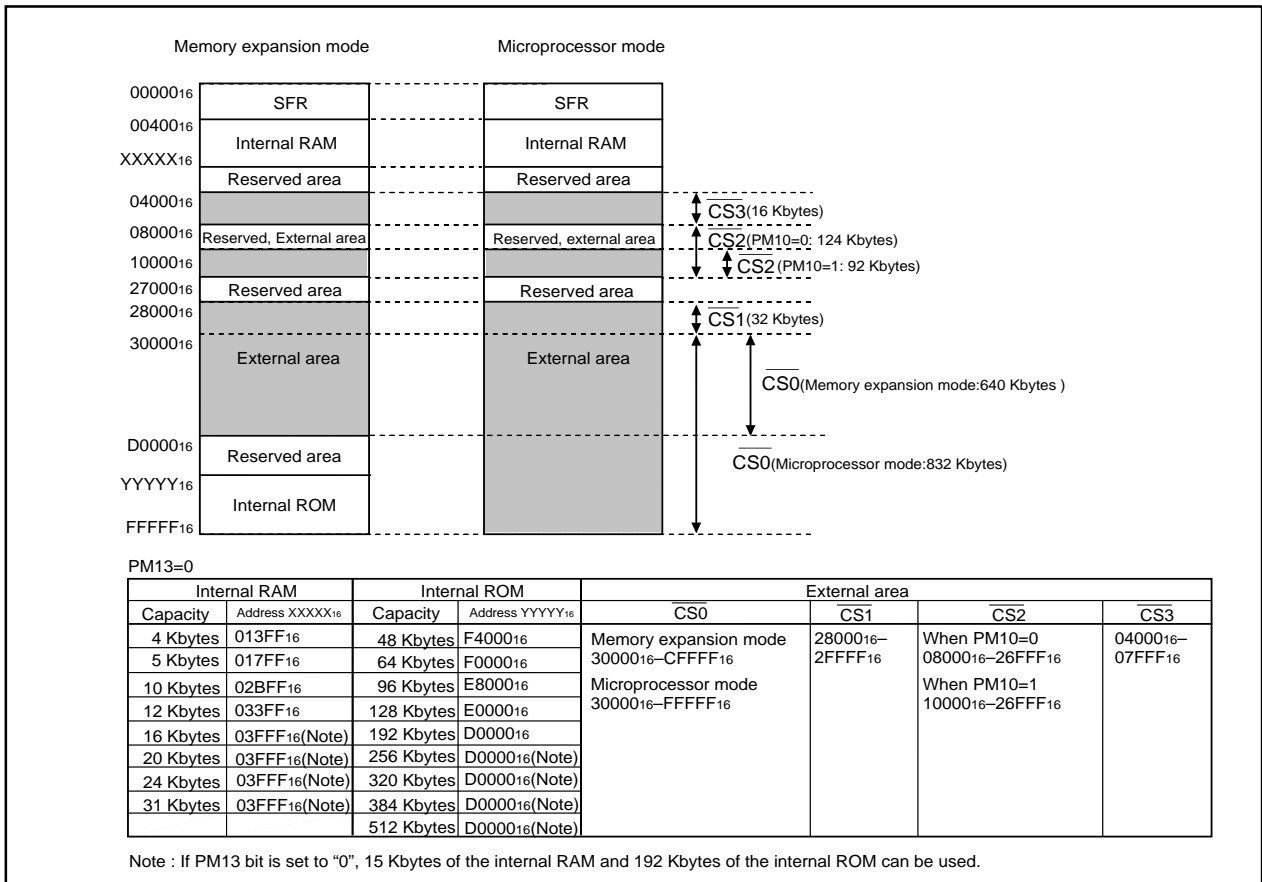
- The  $\overline{CS}_0$  pin outputs "L"
- The  $\overline{CS}_1$  to  $\overline{CS}_3$  pins output the value of the BSR2 to BSR0 bits (bank number)

Figures 1.8.4 to 1.8.5 show the memory mapping and  $\overline{CS}$  area in 4 Mbyte mode. Note that banks 0 to 6 are data-only areas. Locate the program in bank 7 or the  $\overline{CS}_i$  area.

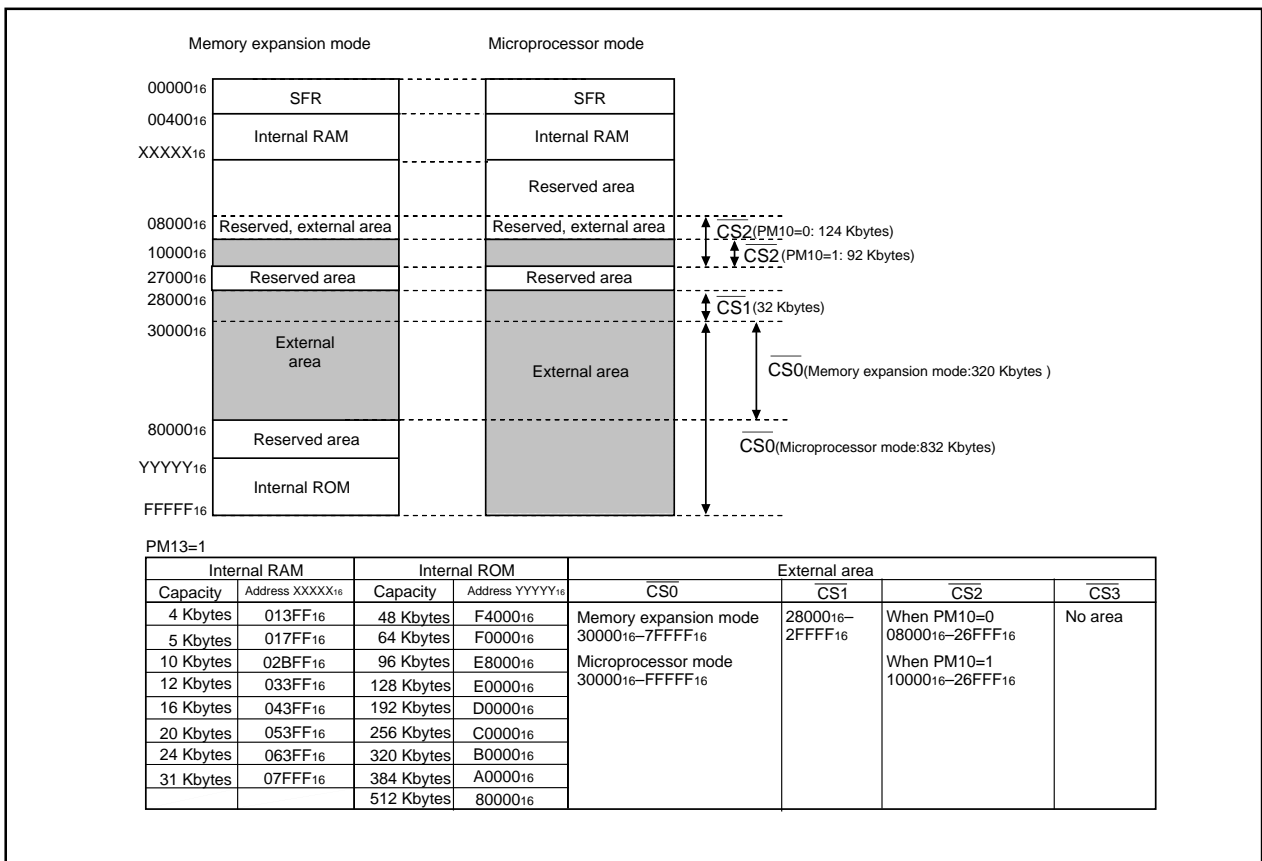


**Figure 1.8.1. DBR Register**

**Memory Space Expansion Function**



**Figure 1.8.2. Memory Mapping and CS Area in 1 Mbyte Mode (PM13=0)**



**Figure 1.8.3. Memory Mapping and CS Area in 1 Mbyte Mode (PM13=1)**

### Memory Space Expansion Function

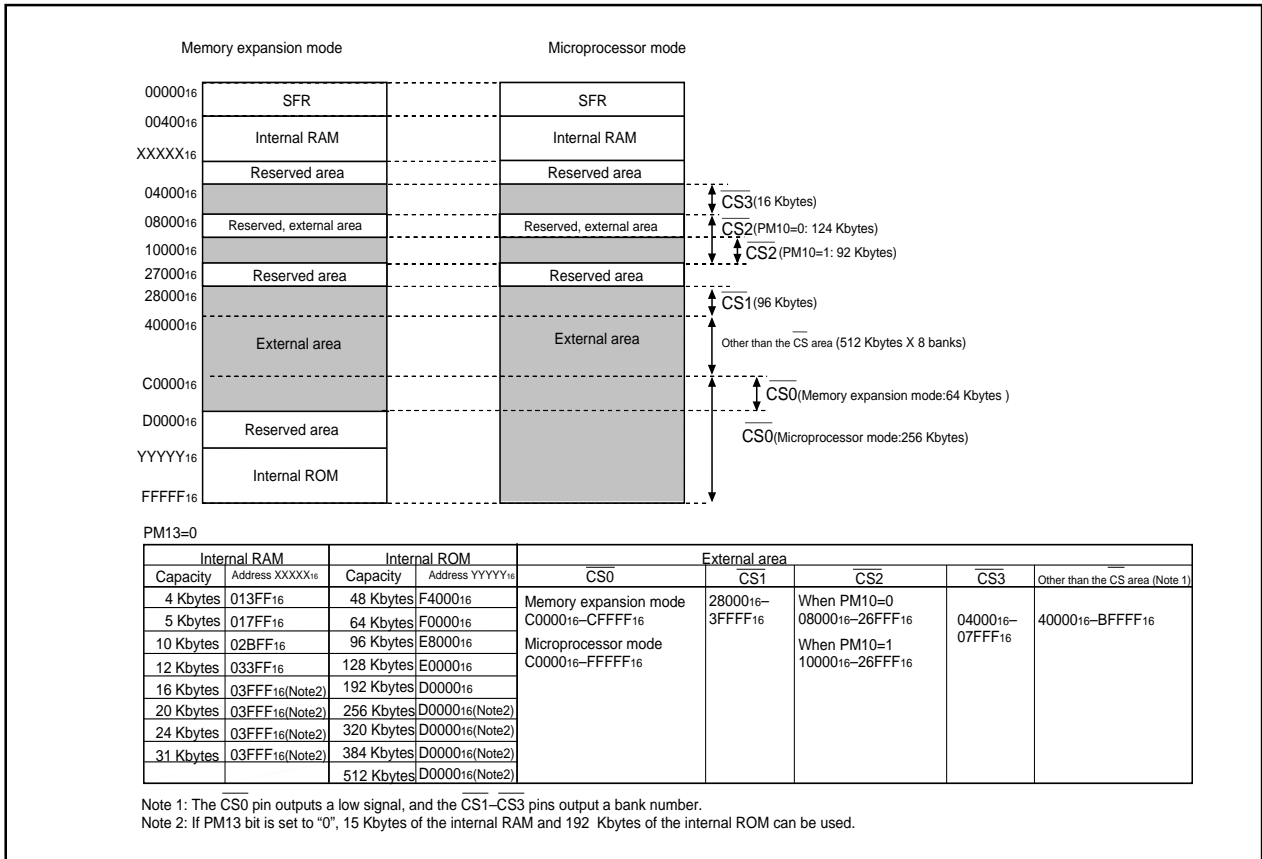


Figure 1.8.4. Memory Mapping and CS Area in 4 Mbyte Mode (PM13=0)

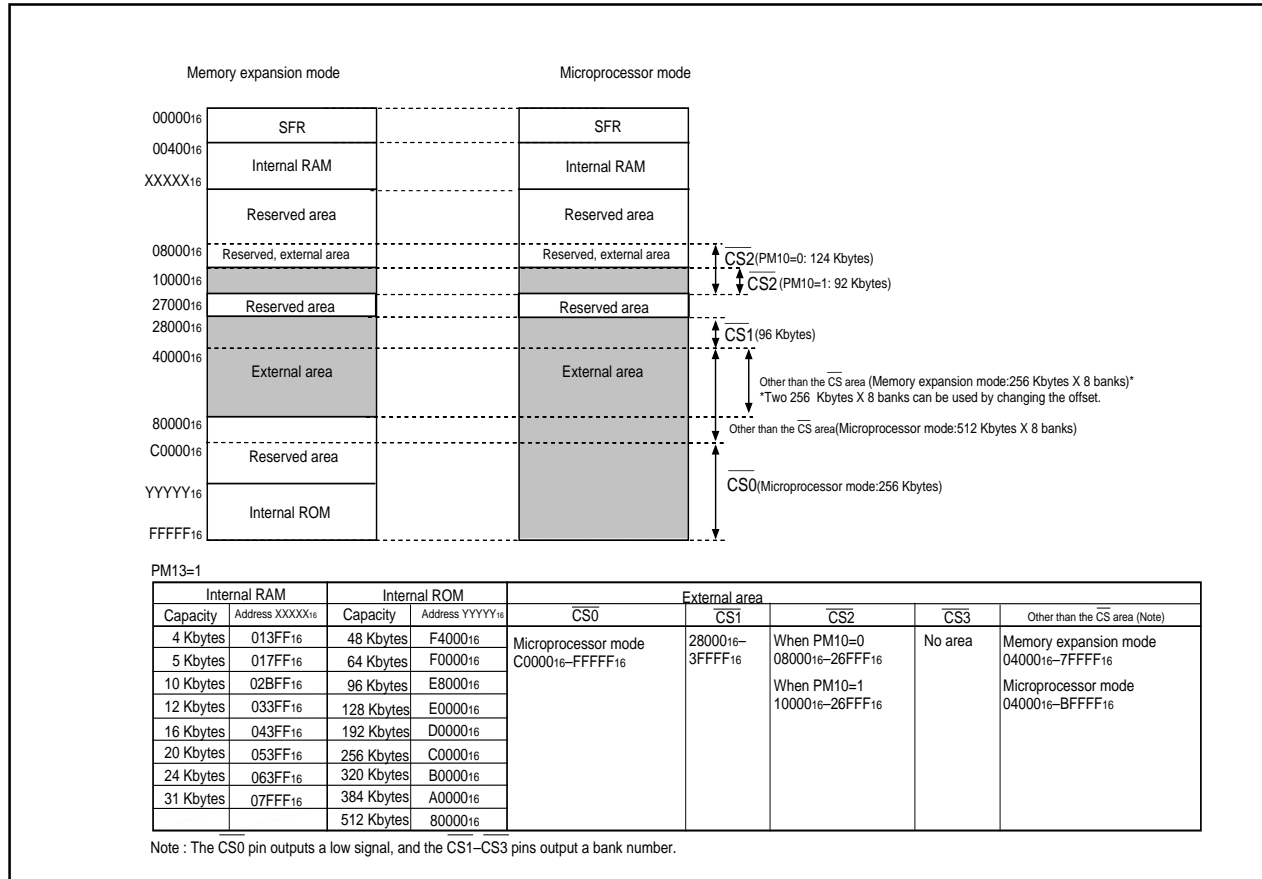


Figure 1.8.5. Memory Mapping and CS Area in 4 Mbyte Mode (PM13=1)

## Memory Space Expansion Function

Figure 1.8.6 shows the external memory connect example in 4 Mbyte mode.

In this example, the  $\overline{CS}$  pin of 4-Mbyte ROM is connected to the  $\overline{CS0}$  pin of microcomputer. The address input AD21, AD20 and AD19 pins are connected to the  $\overline{CS3}$ ,  $\overline{CS2}$  and  $\overline{CS1}$  pins of microcomputer, respectively. The address input AD18 pin is connected to the A19 pin of microcomputer. Figures 1.8.7 to 1.8.9 show the relationship of addresses between the 4-Mbyte ROM and the microcomputer for the case of a connection example in Figure 1.8.6.

In microprocessor mode, or in memory expansion mode where the PM13 bit is "0", banks are located every 512 Kbytes. Setting the OFS bit to "1" allows the accessed address to be offset by  $40000_{16}$ , so that even the data overlapping a bank boundary can be accessed in succession.

In memory expansion mode where the PM13 bit is "1", each 512-Kbyte bank can be accessed in 256 Kbyte units by switching them over with the OFS bit.

Because the SRAM can be accessed on condition that the chip select signals  $S2 = "H"$  and  $\overline{S1} = "L"$ ,  $\overline{CS0}$  and  $\overline{CS2}$  can be connected to  $\overline{S2}$  and  $\overline{S1}$ , respectively. If the SRAM does not have the input pins to accept "H" active and "L" active chip select signals,  $\overline{CS0}$  and  $\overline{CS2}$  should be decoded external to the chip.

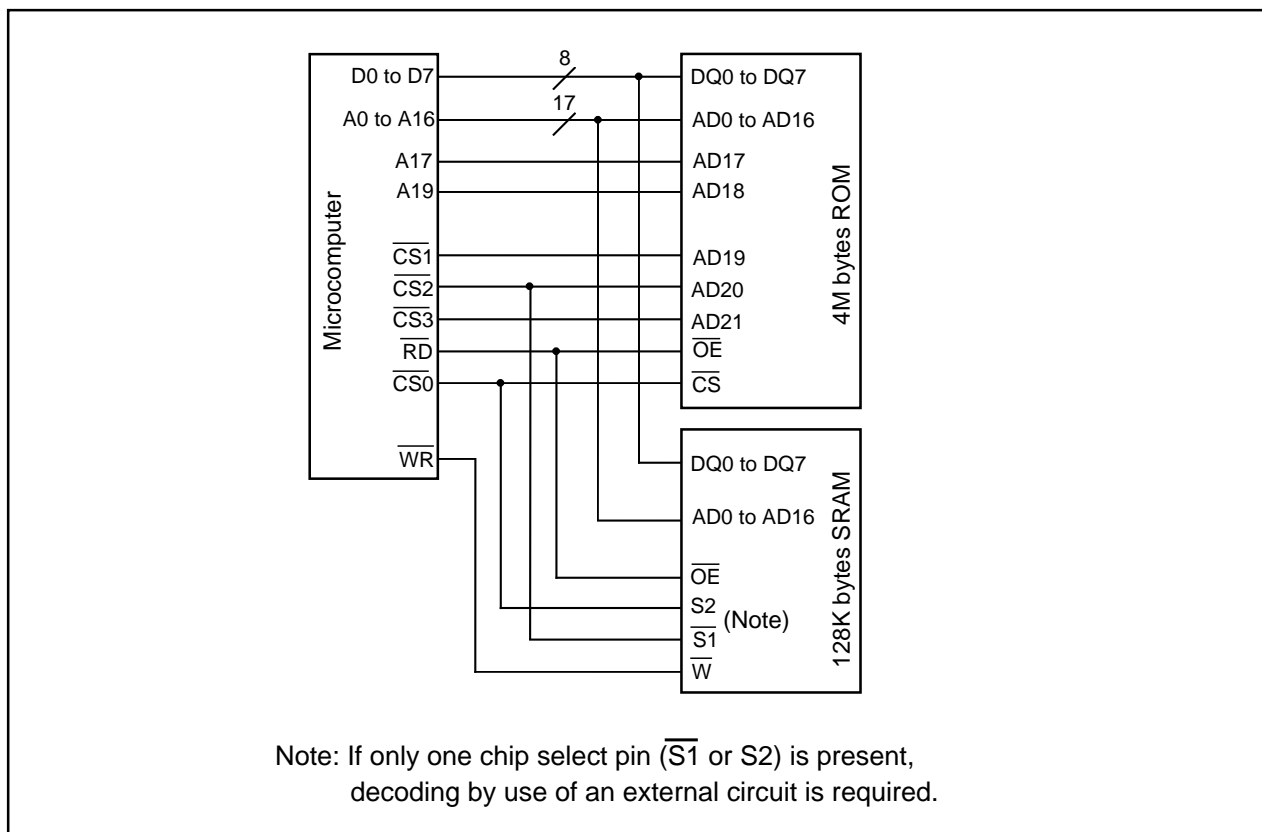
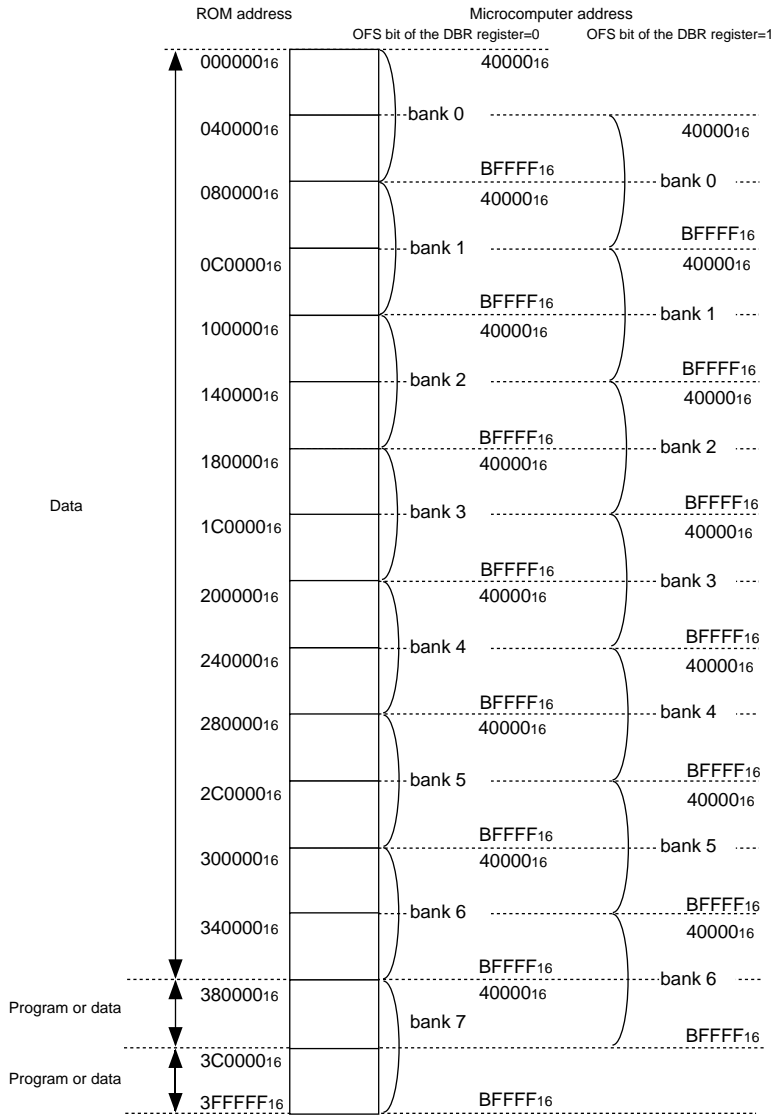


Figure 1.8.6. External Memory Connect Example in 4M Byte Mode

Memory Space Expansion Function



Memory expansion mode where PM13 =0

Bank number	OFS	Access area	Output from the microcomputer pins								
			CS output			Address output					
			CS3	CS2	CS1	A19	A18	A17	A16	A15-A0	
0	0	40000 <sub>16</sub>	0	0	0	0	1	0	0	0000 <sub>16</sub>	000000 <sub>16</sub>
		BFFFF <sub>16</sub>	0	0	0	1	0	1	1	FFFF <sub>16</sub>	07FFFF <sub>16</sub>
0	1	40000 <sub>16</sub>	0	0	0	1	0	0	0	0000 <sub>16</sub>	040000 <sub>16</sub>
		BFFFF <sub>16</sub>	0	0	1	0	1	1	1	FFFF <sub>16</sub>	0BFFFF <sub>16</sub>
1	0	40000 <sub>16</sub>	0	0	1	0	1	0	0	0000 <sub>16</sub>	080000 <sub>16</sub>
		BFFFF <sub>16</sub>	0	0	1	1	0	1	1	FFFF <sub>16</sub>	0FFFFF <sub>16</sub>
1	1	40000 <sub>16</sub>	0	0	1	1	0	0	0	0000 <sub>16</sub>	0C0000 <sub>16</sub>
		BFFFF <sub>16</sub>	0	1	0	0	1	1	1	FFFF <sub>16</sub>	13FFFF <sub>16</sub>
2	0	40000 <sub>16</sub>	0	1	0	0	1	0	0	0000 <sub>16</sub>	100000 <sub>16</sub>
		BFFFF <sub>16</sub>	0	1	0	1	0	1	1	FFFF <sub>16</sub>	17FFFF <sub>16</sub>
2	1	40000 <sub>16</sub>	0	1	0	1	0	0	0	0000 <sub>16</sub>	140000 <sub>16</sub>
		BFFFF <sub>16</sub>	0	1	1	0	1	1	1	FFFF <sub>16</sub>	1BFFFF <sub>16</sub>
3	0	40000 <sub>16</sub>	0	1	1	0	1	0	0	0000 <sub>16</sub>	180000 <sub>16</sub>
		BFFFF <sub>16</sub>	0	1	1	1	0	1	1	FFFF <sub>16</sub>	1FFFFF <sub>16</sub>
3	1	40000 <sub>16</sub>	0	1	1	1	0	0	0	0000 <sub>16</sub>	1C0000 <sub>16</sub>
		BFFFF <sub>16</sub>	1	0	0	0	1	1	1	FFFF <sub>16</sub>	23FFFF <sub>16</sub>
4	0	40000 <sub>16</sub>	1	0	0	0	1	0	0	0000 <sub>16</sub>	200000 <sub>16</sub>
		BFFFF <sub>16</sub>	1	0	0	1	0	1	1	FFFF <sub>16</sub>	27FFFF <sub>16</sub>
4	1	40000 <sub>16</sub>	1	0	0	1	0	0	0	0000 <sub>16</sub>	240000 <sub>16</sub>
		BFFFF <sub>16</sub>	1	0	1	0	1	1	1	FFFF <sub>16</sub>	2BFFFF <sub>16</sub>
5	0	40000 <sub>16</sub>	1	0	1	0	1	0	0	0000 <sub>16</sub>	280000 <sub>16</sub>
		BFFFF <sub>16</sub>	1	0	1	1	0	1	1	FFFF <sub>16</sub>	2FFFFF <sub>16</sub>
5	1	40000 <sub>16</sub>	1	0	1	1	0	0	0	0000 <sub>16</sub>	2C0000 <sub>16</sub>
		BFFFF <sub>16</sub>	1	1	0	0	1	1	1	FFFF <sub>16</sub>	33FFFF <sub>16</sub>
6	0	40000 <sub>16</sub>	1	1	0	0	1	0	0	0000 <sub>16</sub>	300000 <sub>16</sub>
		BFFFF <sub>16</sub>	1	1	0	1	0	1	1	FFFF <sub>16</sub>	37FFFF <sub>16</sub>
6	1	40000 <sub>16</sub>	1	1	0	1	0	0	0	0000 <sub>16</sub>	340000 <sub>16</sub>
		BFFFF <sub>16</sub>	1	1	1	0	1	1	1	FFFF <sub>16</sub>	3BFFFF <sub>16</sub>
7	0	40000 <sub>16</sub>	1	1	1	0	1	0	0	0000 <sub>16</sub>	380000 <sub>16</sub>
		7FFFF <sub>16</sub>	1	1	1	0	1	1	1	FFFF <sub>16</sub>	3BFFFF <sub>16</sub>
		80000 <sub>16</sub>	1	1	1	1	0	0	0	0000 <sub>16</sub>	3C0000 <sub>16</sub>
		BFFFF <sub>16</sub>	1	1	1	1	0	1	1	FFFF <sub>16</sub>	3FFFFF <sub>16</sub>
		C0000 <sub>16</sub>	1	1	1	1	1	0	0	0000 <sub>16</sub>	3C0000 <sub>16</sub>
		CFFFF <sub>16</sub>	1	1	1	1	1	0	0	FFFF <sub>16</sub>	3CFFFF <sub>16</sub>
		D0000 <sub>16</sub>								Internal ROM access	
		DFFFF <sub>16</sub>								Internal ROM access	
		D0000 <sub>16</sub>								Internal ROM access	
		DFFFF <sub>16</sub>								Internal ROM access	
			A21	A20	A19	A18	N.C.	A17	A16	A15-A0	Address input for 4-Mbyte ROM

N.C.: No connected

Figure 1.8.7. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (1)



Memory Space Expansion Function

Memory expansion mode where PM13 = 1

Bank number	OFS	Access area	Output from the microcomputer pins								
			CS output			Address output					
			CS3	CS2	CS1	A19	A18	A17	A16	A15-A0	
0	0	40000 <sub>16</sub>	0	0	0	0	1	0	0	0000 <sub>16</sub>	000000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	0	0	0	0	1	1	1	FFFF <sub>16</sub>	03FFFF <sub>16</sub>
1	0	40000 <sub>16</sub>	0	0	1	0	1	0	0	0000 <sub>16</sub>	040000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	0	0	1	1	0	1	1	FFFF <sub>16</sub>	07FFFF <sub>16</sub>
2	0	40000 <sub>16</sub>	0	1	0	0	1	0	0	0000 <sub>16</sub>	080000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	0	1	0	1	0	1	1	FFFF <sub>16</sub>	0BFFFF <sub>16</sub>
3	0	40000 <sub>16</sub>	0	1	1	0	1	0	0	0000 <sub>16</sub>	0C0000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	0	1	1	1	0	1	1	FFFF <sub>16</sub>	0FFFFF <sub>16</sub>
4	0	40000 <sub>16</sub>	0	1	0	0	1	0	0	0000 <sub>16</sub>	100000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	0	1	0	1	0	1	1	FFFF <sub>16</sub>	13FFFF <sub>16</sub>
5	0	40000 <sub>16</sub>	0	1	0	1	0	0	0	0000 <sub>16</sub>	140000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	0	1	0	1	1	0	1	FFFF <sub>16</sub>	17FFFF <sub>16</sub>
6	0	40000 <sub>16</sub>	1	0	0	0	1	0	0	0000 <sub>16</sub>	180000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	0	0	1	0	1	1	FFFF <sub>16</sub>	1BFFFF <sub>16</sub>
7	0	40000 <sub>16</sub>	1	0	1	0	1	0	0	0000 <sub>16</sub>	1C0000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	0	1	1	0	1	1	FFFF <sub>16</sub>	1FFFFF <sub>16</sub>
8	0	40000 <sub>16</sub>	1	0	0	0	1	0	0	0000 <sub>16</sub>	200000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	0	0	1	1	1	1	FFFF <sub>16</sub>	23FFFF <sub>16</sub>
9	0	40000 <sub>16</sub>	1	0	0	1	0	0	0	0000 <sub>16</sub>	240000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	0	0	1	0	1	1	FFFF <sub>16</sub>	27FFFF <sub>16</sub>
10	0	40000 <sub>16</sub>	1	0	1	0	1	0	0	0000 <sub>16</sub>	280000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	0	1	0	1	1	1	FFFF <sub>16</sub>	2BFFFF <sub>16</sub>
11	0	40000 <sub>16</sub>	1	0	1	1	0	0	0	0000 <sub>16</sub>	2C0000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	0	1	1	0	1	1	FFFF <sub>16</sub>	2FFFFF <sub>16</sub>
12	0	40000 <sub>16</sub>	1	1	0	0	1	0	0	0000 <sub>16</sub>	300000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	1	0	0	1	1	1	FFFF <sub>16</sub>	33FFFF <sub>16</sub>
13	0	40000 <sub>16</sub>	1	1	0	1	0	0	0	0000 <sub>16</sub>	340000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	1	0	1	0	1	1	FFFF <sub>16</sub>	37FFFF <sub>16</sub>
14	0	40000 <sub>16</sub>	1	1	1	0	1	0	0	0000 <sub>16</sub>	380000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	1	1	0	1	1	1	FFFF <sub>16</sub>	3BFFFF <sub>16</sub>
15	0	80000 <sub>16</sub>									Internal ROM access
	1	FFFFF <sub>16</sub>									Internal ROM access
16	0	40000 <sub>16</sub>	1	1	1	1	0	0	0	0000 <sub>16</sub>	3C0000 <sub>16</sub>
	1	7FFFF <sub>16</sub>	1	1	1	1	0	1	1	FFFF <sub>16</sub>	3FFFFF <sub>16</sub>
17	0	80000 <sub>16</sub>									Internal ROM access
	1	FFFFF <sub>16</sub>									Internal ROM access
			A21	A20	A19	A18	N.C.	A17	A16	A15-A0	Address input for 4-Mbyte ROM

N.C.: No connected

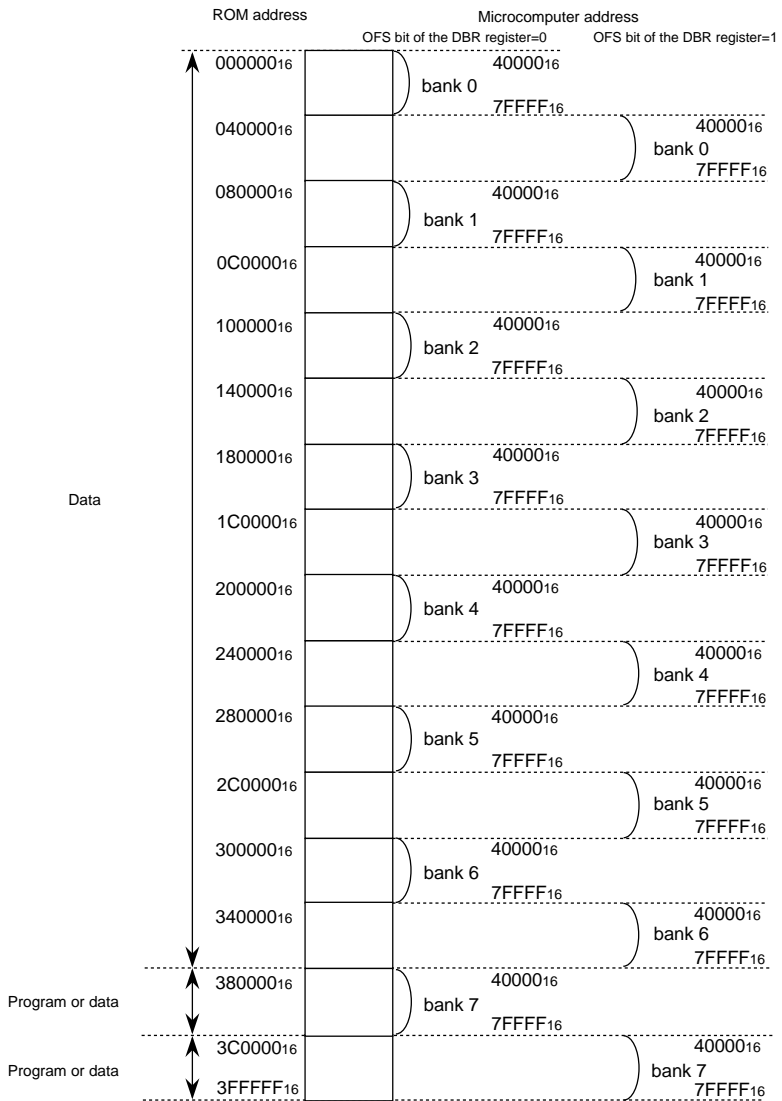
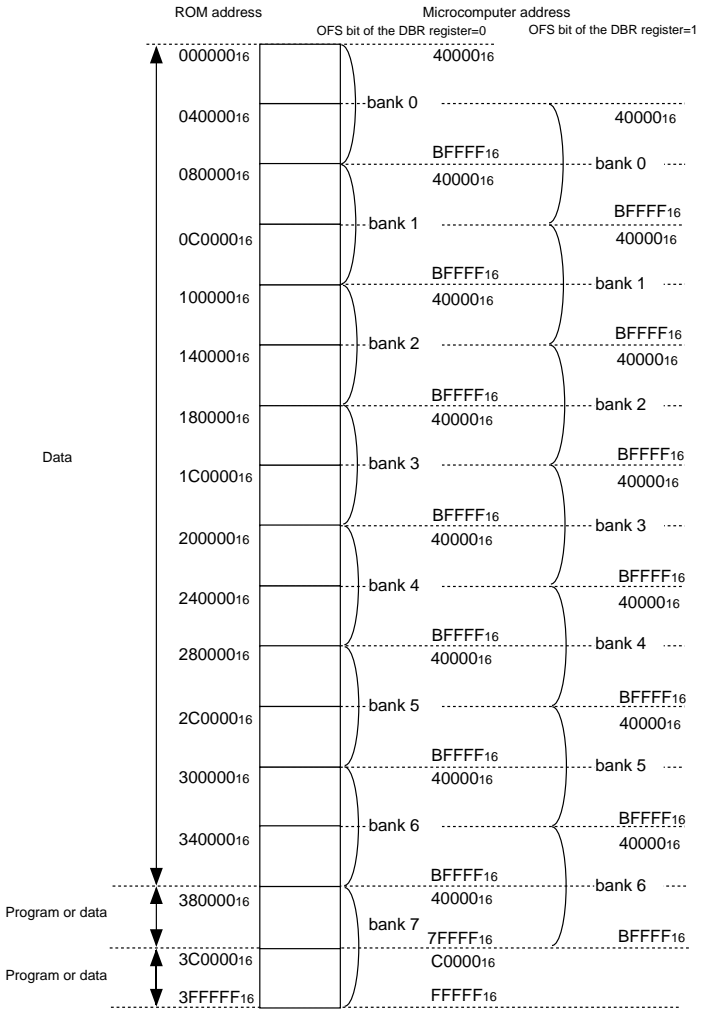


Figure 1.8.8. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (2)





Microprocessor mode

Bank number	OFS	Access area	Output from the microcomputer pins									
			CS output			Address output						
			CS3	CS2	CS1	A19	A18	A17	A16	A15-A0		
0	0	40000 <sub>16</sub>	0	0	0	0	1	0	0	0000 <sub>16</sub>	000000 <sub>16</sub>	
		BFFFF <sub>16</sub>	0	0	0	1	0	1	1	FFFF <sub>16</sub>	07FFFF <sub>16</sub>	
0	1	40000 <sub>16</sub>	0	0	0	1	0	0	0	0000 <sub>16</sub>	040000 <sub>16</sub>	
		BFFFF <sub>16</sub>	0	0	1	0	1	1	1	FFFF <sub>16</sub>	0BFFFF <sub>16</sub>	
1	0	40000 <sub>16</sub>	0	0	1	0	1	0	0	0000 <sub>16</sub>	080000 <sub>16</sub>	
		BFFFF <sub>16</sub>	0	0	1	1	0	1	1	FFFF <sub>16</sub>	0FFFFF <sub>16</sub>	
1	1	40000 <sub>16</sub>	0	0	1	1	0	0	0	0000 <sub>16</sub>	0C0000 <sub>16</sub>	
		BFFFF <sub>16</sub>	0	1	0	0	1	1	1	FFFF <sub>16</sub>	13FFFF <sub>16</sub>	
2	0	40000 <sub>16</sub>	0	1	0	0	1	0	0	0000 <sub>16</sub>	100000 <sub>16</sub>	
		BFFFF <sub>16</sub>	0	1	0	1	0	1	1	FFFF <sub>16</sub>	17FFFF <sub>16</sub>	
2	1	40000 <sub>16</sub>	0	1	0	1	0	0	0	0000 <sub>16</sub>	140000 <sub>16</sub>	
		BFFFF <sub>16</sub>	0	1	1	0	1	1	1	FFFF <sub>16</sub>	1BFFFF <sub>16</sub>	
3	0	40000 <sub>16</sub>	0	1	1	0	1	0	0	0000 <sub>16</sub>	180000 <sub>16</sub>	
		BFFFF <sub>16</sub>	0	1	1	1	0	1	1	FFFF <sub>16</sub>	1FFFFF <sub>16</sub>	
3	1	40000 <sub>16</sub>	0	1	1	1	0	0	0	0000 <sub>16</sub>	1C0000 <sub>16</sub>	
		BFFFF <sub>16</sub>	1	0	0	0	1	1	1	FFFF <sub>16</sub>	23FFFF <sub>16</sub>	
4	0	40000 <sub>16</sub>	1	0	0	0	1	0	0	0000 <sub>16</sub>	200000 <sub>16</sub>	
		BFFFF <sub>16</sub>	1	0	0	1	0	1	1	FFFF <sub>16</sub>	27FFFF <sub>16</sub>	
4	1	40000 <sub>16</sub>	1	0	0	1	0	0	0	0000 <sub>16</sub>	240000 <sub>16</sub>	
		BFFFF <sub>16</sub>	1	0	1	0	1	1	1	FFFF <sub>16</sub>	2BFFFF <sub>16</sub>	
5	0	40000 <sub>16</sub>	1	0	1	0	1	0	0	0000 <sub>16</sub>	280000 <sub>16</sub>	
		BFFFF <sub>16</sub>	1	0	1	1	0	1	1	FFFF <sub>16</sub>	2FFFFF <sub>16</sub>	
5	1	40000 <sub>16</sub>	1	0	1	1	0	0	0	0000 <sub>16</sub>	2C0000 <sub>16</sub>	
		BFFFF <sub>16</sub>	1	1	0	0	1	1	1	FFFF <sub>16</sub>	33FFFF <sub>16</sub>	
6	0	40000 <sub>16</sub>	1	1	0	0	1	0	0	0000 <sub>16</sub>	300000 <sub>16</sub>	
		BFFFF <sub>16</sub>	1	1	0	1	0	1	1	FFFF <sub>16</sub>	37FFFF <sub>16</sub>	
6	1	40000 <sub>16</sub>	1	1	0	1	0	0	0	0000 <sub>16</sub>	340000 <sub>16</sub>	
		BFFFF <sub>16</sub>	1	1	1	0	1	1	1	FFFF <sub>16</sub>	3BFFFF <sub>16</sub>	
7	0	40000 <sub>16</sub>	1	1	1	0	1	0	0	0000 <sub>16</sub>	380000 <sub>16</sub>	
		7FFFF <sub>16</sub>	1	1	1	0	1	1	1	FFFF <sub>16</sub>	3BFFFF <sub>16</sub>	
		80000 <sub>16</sub>	1	1	1	1	0	0	0	0	0000 <sub>16</sub>	3C0000 <sub>16</sub>
		BFFFF <sub>16</sub>	1	1	1	1	0	1	1	1	FFFF <sub>16</sub>	3FFFFF <sub>16</sub>
7	1	C0000 <sub>16</sub>	1	1	1	1	1	0	0	0000 <sub>16</sub>	3C0000 <sub>16</sub>	
		FFFFF <sub>16</sub>	1	1	1	1	1	1	1	FFFF <sub>16</sub>	3FFFFF <sub>16</sub>	
		A21	A20	A19	A18	N.C.	A17	A16	A15-A0	Address input for 4-Mbyte ROM		
		Address input for 4-Mbyte ROM										

N.C.: No connected

Figure 1.8.9. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (3)

## Clock Generation Circuit

# Clock Generation Circuit

The clock generation circuit contains four oscillator circuits as follows:

- (1) Main clock oscillation circuit
- (2) Sub clock oscillation circuit
- (3) Ring oscillator (oscillation stop detect function)
- (4) PLL frequency synthesizer

Table 1.9.1 lists the clock generation circuit specifications. Figure 1.9.1 shows the clock generation circuit. Figures 1.9.2 to 1.9.6 show the clock-related registers.

**Table 1.9.1. Clock Generation Circuit Specifications**

Item	Main clock oscillation circuit	Sub clock oscillation circuit	Ring oscillator	PLL frequency synthesizer
Use of clock	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Timer A, B's clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> <li>• CPU and peripheral function clock sources when the main clock stops oscillating</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>
Clock frequency	0 to 16 MHz	32.768 kHz	About 1 MHz	10 to 24 MHz
Usable oscillator	<ul style="list-style-type: none"> <li>• Ceramic oscillator</li> <li>• Crystal oscillator</li> </ul>	<ul style="list-style-type: none"> <li>• Crystal oscillator</li> </ul>	_____	_____
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	_____	_____
Oscillation stop, restart function	Presence	Presence	Presence	Presence
Oscillator status after reset	Oscillating	Stopped	Stopped	Stopped
Other	Externally derived clock can be input		_____	_____

Clock Generation Circuit

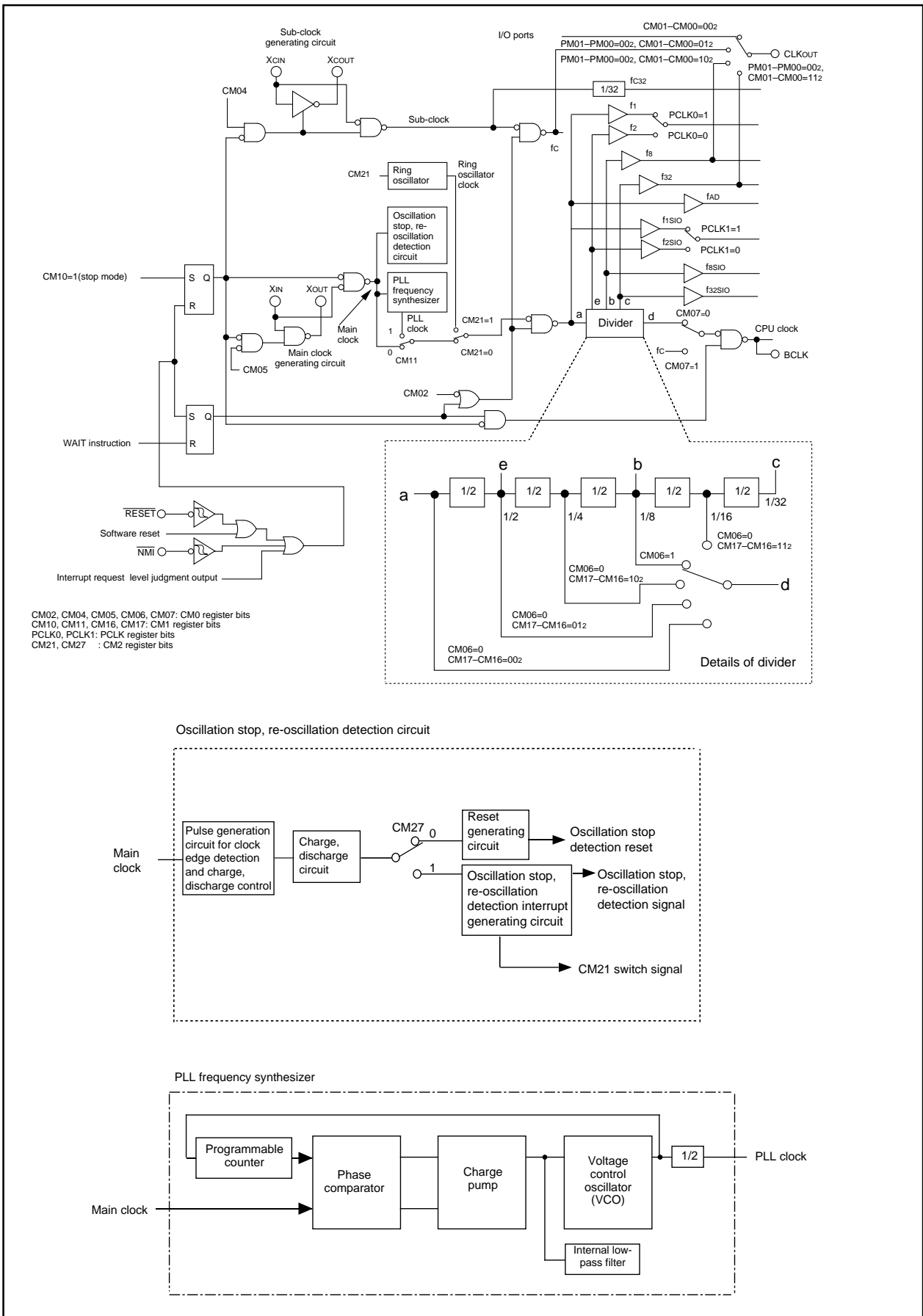
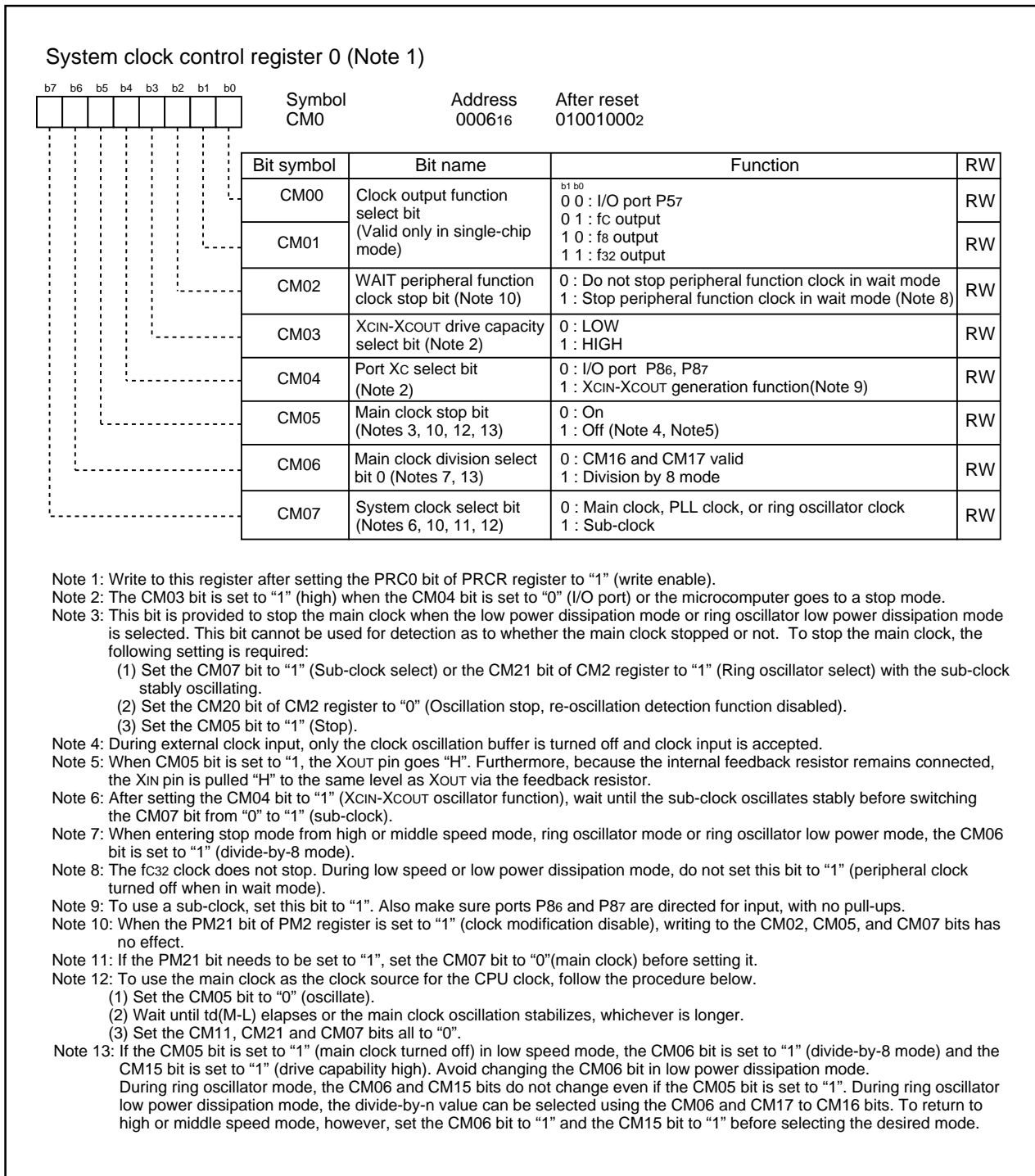


Figure 1.9.1. Clock Generation Circuit

**Clock Generation Circuit**



**Figure 1.9.2. CM0 Register**

Clock Generation Circuit

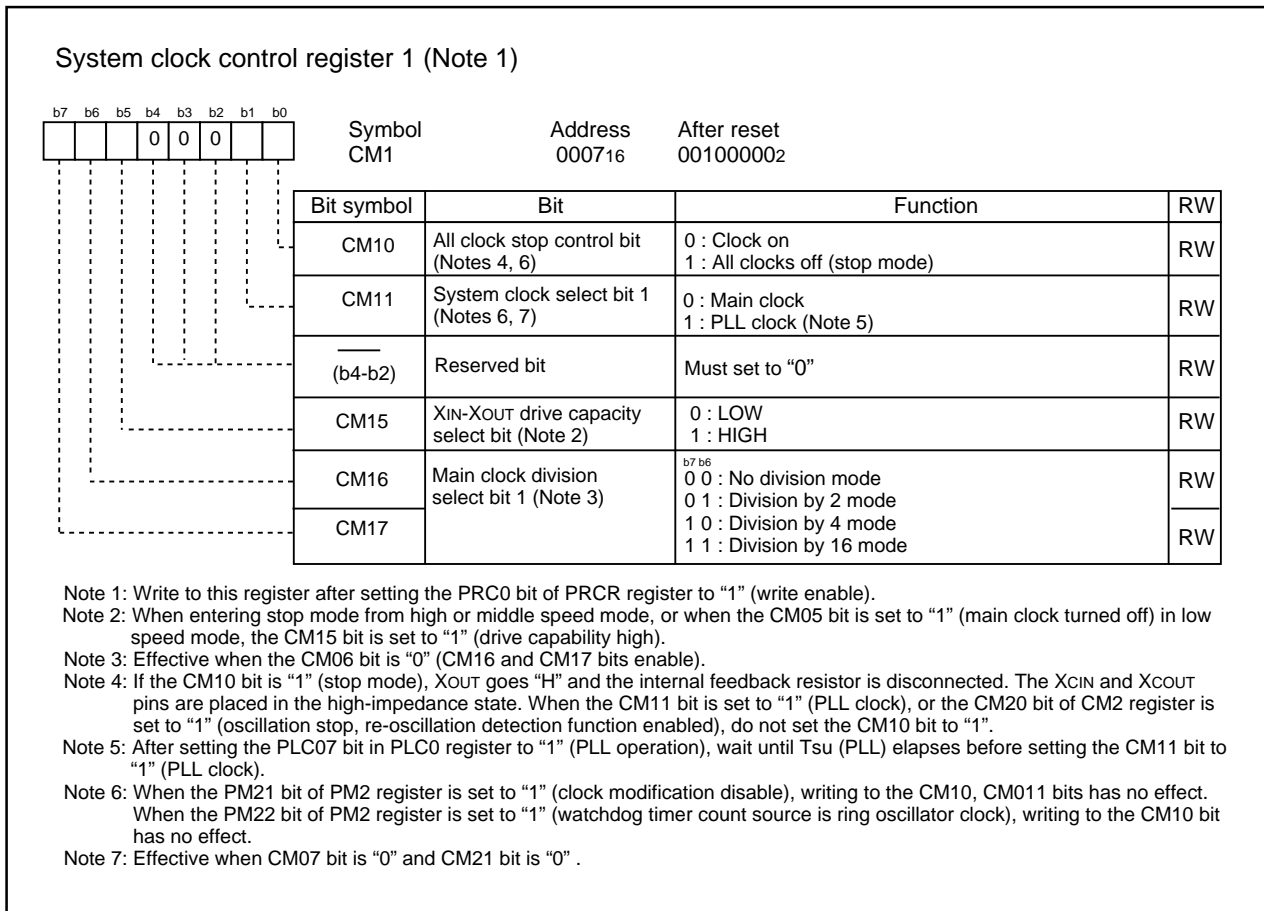
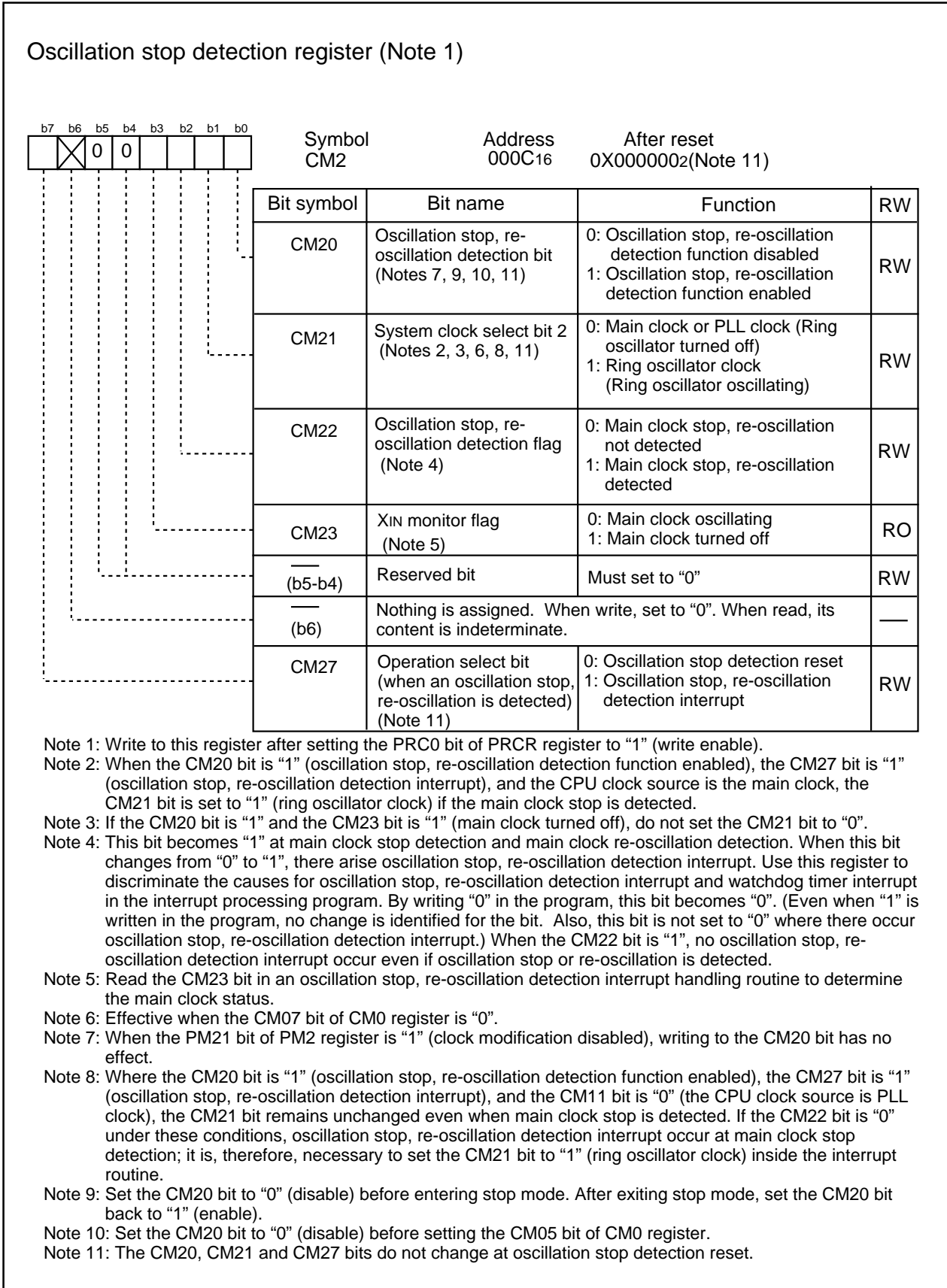


Figure 1.9.3. CM1 Register

**Clock Generation Circuit**



**Figure 1.9.4. CM2 Register**

Clock Generation Circuit

Peripheral clock select register (Note)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol PCLKR	Address 025E16	When reset 00000112	
0	0	0	0	0	0						
								Bit symbol	Bit name	Function	RW
								PCLK0	Timers A, B clock select bit (Clock source for the timers A, B, and the dead time timer)	0 : f2 1 : f1	RW
								PCLK1	SI/O clock select bit (Clock source for UART0 to UART2, SI/O3, SI/O4)	0 : f2SIO 1 : f1SIO	RW
								— (b7-b2)	Reserved bit	Must set to "0"	RW

Note: Write to this register after setting the PRC0 bit of PRCR register to "1" (write enable).

Processor mode register 2 (Note 1)

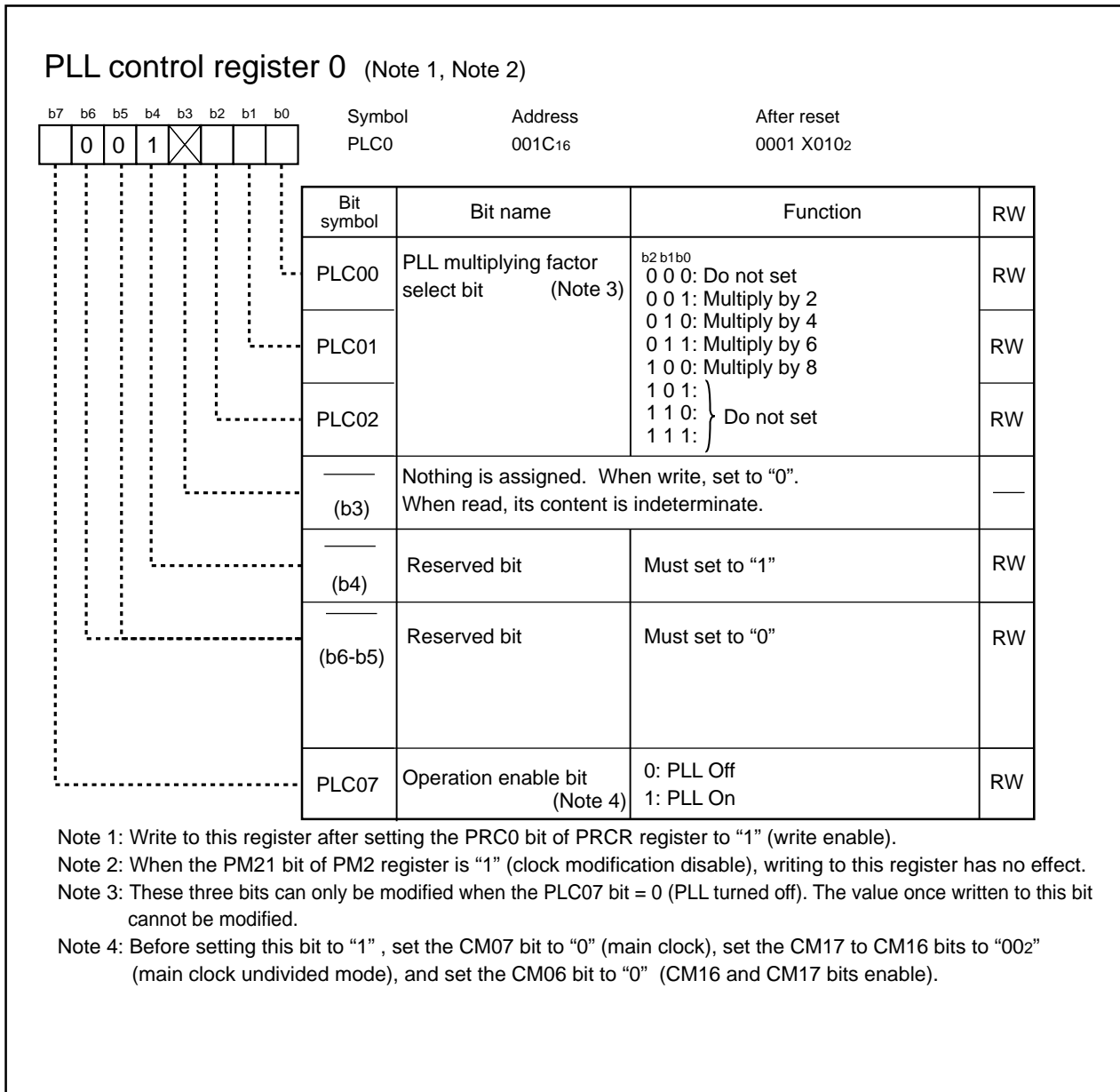
b7	b6	b5	b4	b3	b2	b1	b0	Symbol PM2	Address 001E16	After reset XXX000002	
X	X	X	0	0							
								Bit symbol	Bit name	Function	RW
								PM20	Specifying wait when accessing SFR at PLL operation (Note 2)	0 : 2 waits 1 : 1 wait	RW
								PM21	System clock protective bit (Note 3, Note 4)	0 : Clock is protected by PRCR register 1 : Clock modification disabled	RW
								PM22	WDT count source protective bit (Note 3, Note 5)	0 : CPU clock is used for the watchdog timer count source 1 : Ring oscillator clock is used for the watchdog timer count source	RW
								— (b4-b3)	Reserved bit	Must set to "0"	RW
								— (b7-b5)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

- Note 1: Write to this register after setting the PRC1 bit of PRCR register to "1" (write enable).
- Note 2: This bit can only be rewritten while the PLC07 bit is "0" (PLL turned off). Also, to select a 16 MHz or higher PLL clock, set this bit to "0" (2 waits). Note that if the clock source for the CPU clock is to be changed from the PLL clock to another, the PLC07 bit must be set to "0" before setting the PM20 bit.
- Note 3: Once this bit is set to "1", it cannot be cleared to "0" in a program.
- Note 4: Setting the PM21 bit to "1" results in the following conditions:
- The BCLK is not halted by executing the WAIT instruction.
  - Writing to the following bits has no effect.
    - CM02 bit of CM0 register
    - CM05 bit of CM0 register (main clock is not halted)
    - CM07 bit of CM0 register (CPU clock source does not change)
    - CM10 bit of CM1 register (stop mode is not entered)
    - CM11 bit of CM1 register (CPU clock source does not change)
    - CM20 bit of CM2 register (oscillation stop, re-oscillation detection function settings do not change)
    - All bits of PLC0 register (PLL frequency synthesizer settings do not change)
- Note 5: Setting the PM22 bit to "1" results in the following conditions:
- The ring oscillator starts oscillating, and the ring oscillator clock becomes the watchdog timer count source.
  - The CM10 bit of CM1 register is disabled against write. (Writing a "1" has no effect, nor is stop mode entered.)
  - The watchdog timer does not stop when in wait mode or hold state.

Figure 1.9.5. PCLKR Register and PM2 Register



**Clock Generation Circuit**



**Figure 1.9.6. PLC0 Register**

## Clock Generation Circuit

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

### (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 1.9.7 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 of CM0 register to "1" (main clock oscillator circuit turned off) after switching the clock source for the CPU clock to a sub clock or ring 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". 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 "power control".

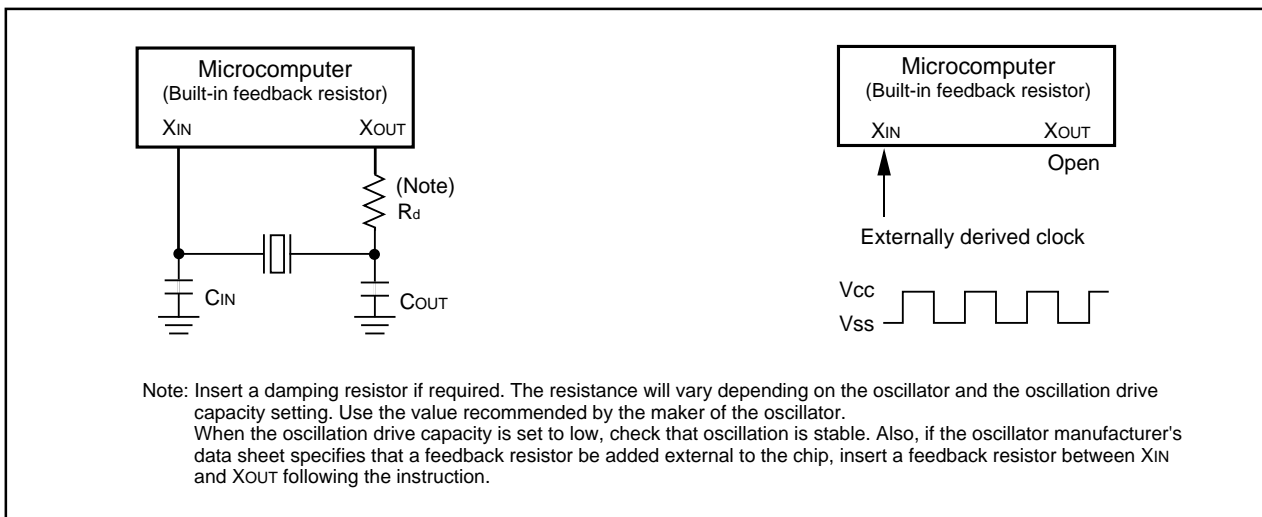


Figure 1.9.7. Examples of Main Clock Connection Circuit

## Clock Generation Circuit

**(2) Sub Clock**

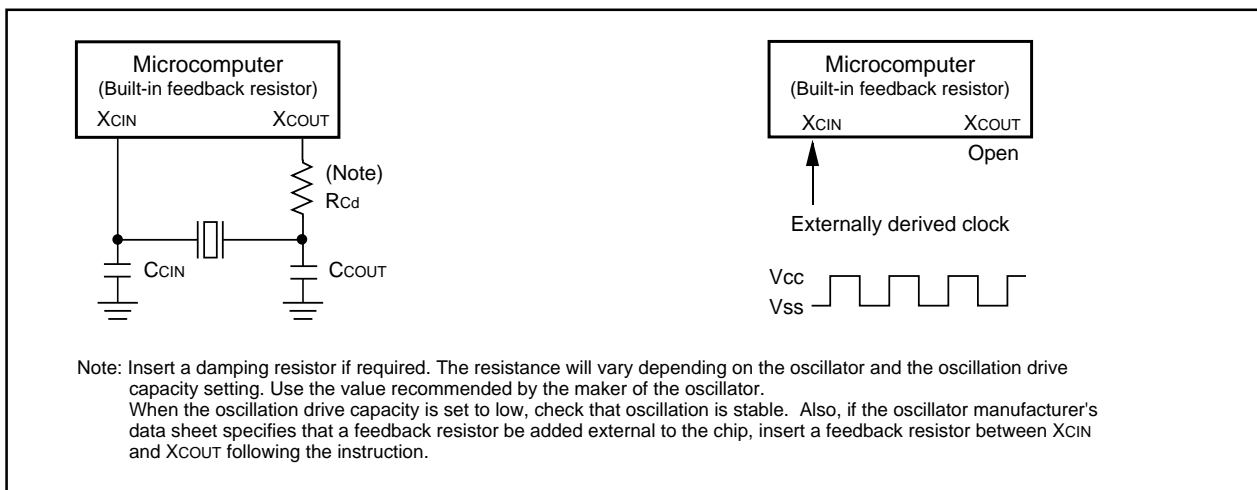
The sub clock is generated by the sub clock oscillation circuit. This clock is used as the clock source for the CPU clock, as well as the timer A and timer B count sources. In addition, an fc clock with the same frequency as that of the sub clock can be output from the CLKOUT pin.

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 1.9.8 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 of 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 "power control".



**Figure 1.9.8. Examples of Sub Clock Connection Circuit**

## Clock Generation Circuit

### (3) Ring Oscillator Clock

This clock, approximately 1 MHz, is supplied by a ring oscillator. This clock is used as the clock source for the CPU and peripheral function clocks. In addition, if the PM22 bit of PM2 register is "1" (ring oscillator clock for the watchdog timer count source), this clock is used as the count source for the watchdog timer. After reset, the ring oscillator clock is turned off. It is turned on by setting the CM21 bit of CM2 register to "1" (ring oscillator clock), and is used as the clock source for the CPU and peripheral function clocks, in place of the main clock. If the main clock stops oscillating when the CM20 bit of CM2 register is "1" (oscillation stop, re-oscillation detection function enabled) and the CM27 bit is "1" (oscillation stop, re-oscillation detection interrupt), the ring oscillator automatically starts operating, supplying the necessary clock for the microcomputer.

### (4) PLL Clock

The PLL clock is generated from the main clock by a PLL frequency synthesizer. This clock is used as the clock source for the CPU and peripheral function clocks. After reset, the PLL clock is turned off. The PLL frequency synthesizer is activated by setting the PLC07 bit to "1" (PLL operation). When the PLL clock is used as the clock source for the CPU clock, wait  $t_{su}(PLL)$  for the PLL clock to be stable, and then set the CM11 bit in the CM1 register to "1".

To enter wait or stop mode, set the CM11 bit to "0" (main clock for the CPU clock source) and then the PLC07 bit of PLC0 register to "0" (PLL off) before entering that mode. Figure 1.9.9 shows the procedure for using the PLL clock as the clock source for the CPU.

The PLL clock frequency is determined by the equation below.

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

(However,  $10 \text{ MHz} \leq \text{PLL clock frequency} \leq 24 \text{ MHz}$ )

The PLC02 to PLC00 bits can be set only once after reset. Table 1.9.2 shows the example for setting PLL clock frequencies.

**Table 1.9.2. Example for Setting PLL Clock Frequencies**

XIN (MHz)	PLC02	PLC01	PLC00	Multiplying factor	PLL clock (MHz)(Note)
10	0	0	1	2	20
5	0	1	0	4	
3.33	0	1	1	6	
2.5	1	0	0	8	
12	0	0	1	2	24
6	0	1	0	4	
4	0	1	1	6	
3	1	0	0	8	

Note:  $10\text{MHz} \leq \text{PLL clock frequency} \leq 24\text{MHz}$ .

## Clock Generation Circuit

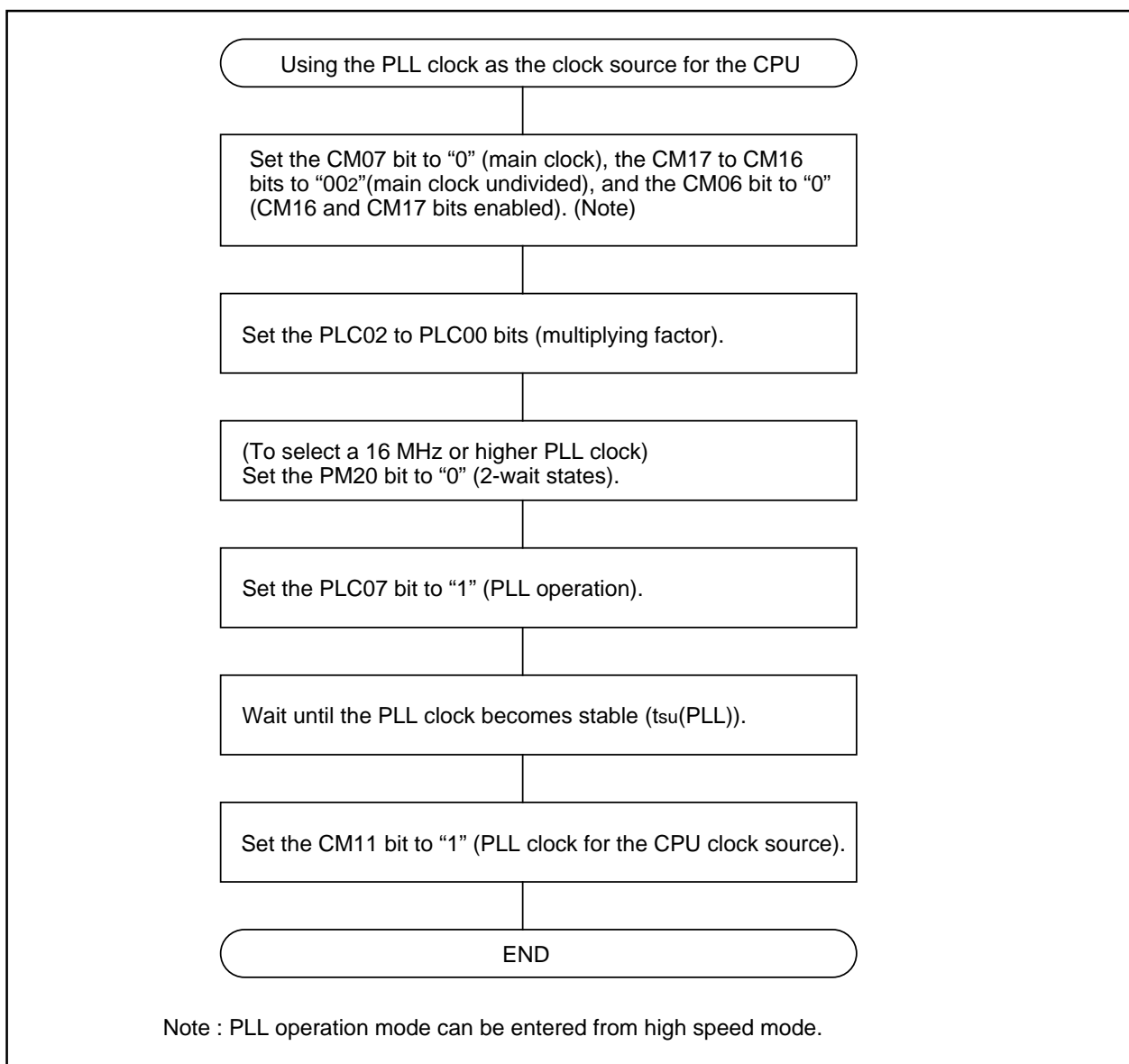


Figure 1.9.9. Procedure to Use PLL Clock as CPU Clock Source

## CPU Clock and Peripheral Function Clock

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

### (1) CPU Clock and BCLK

These are operating clocks for the CPU and watchdog timer.

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

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

When the PLL clock is selected as the clock source for the CPU clock, the CM06 bit should be set to "0" and the CM17 to CM16 bits to "002" (undivided).

After reset, the main clock divided by 8 provides the CPU clock.

During memory expansion or microprocessor mode, a BCLK signal with the same frequency as the CPU clock can be output from the BCLK pin by setting the PM07 bit of PM0 register to "0" (output enabled).

Note that when entering stop mode from high or middle speed mode, ring oscillator mode or low power ring oscillator mode, or when the CM05 bit of CM0 register is set to "1" (main clock turned off) in low-speed mode, the CM06 bit of CM0 register is set to "1" (divide-by-8 mode).

### (2) Peripheral Function Clock(f<sub>1</sub>, f<sub>2</sub>, f<sub>8</sub>, f<sub>32</sub>, f<sub>1SIO</sub>, f<sub>2SIO</sub>, f<sub>8SIO</sub>, f<sub>32SIO</sub>, f<sub>AD</sub>, f<sub>C32</sub>)

These are operating clocks for the peripheral functions.

Of these, f<sub>i</sub> (i = 1, 2, 8, 32) and f<sub>SIO</sub> are derived from the main clock, PLL clock or ring oscillator clock by dividing them by i. The clock f<sub>i</sub> is used for timers A and B, and f<sub>SIO</sub> is used for serial I/O. The f<sub>8</sub> and f<sub>32</sub> clocks can be output from the CLKOUT pin.

The f<sub>AD</sub> clock is produced from the main clock, PLL clock or ring oscillator clock, and is used for the A-D converter.

When the WAIT instruction is executed after setting the CM02 bit of CM0 register to "1" (peripheral function clock turned off during wait mode), or when the microcomputer is in low power dissipation mode, the f<sub>i</sub>, f<sub>SIO</sub> and f<sub>AD</sub> clocks are turned off.

The f<sub>C32</sub> 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.

## Clock Output Function

During single-chip mode, the f<sub>8</sub>, f<sub>32</sub> or f<sub>C</sub> clock can be output from the CLKOUT pin. Use the CM01 to CM00 bits of CM0 register to select.

## Clock Generation Circuit

---

### Power Control

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

#### (1) Normal Operation Mode

Normal operation mode is further classified into seven modes.

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

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

Note that operation modes cannot be changed directly from low speed or low power dissipation mode to ring oscillator or ring oscillator low power dissipation mode. Nor can operation modes be changed directly from ring oscillator or ring oscillator low power dissipation mode to low speed or low power dissipation mode. Where the CPU clock source is changed from the ring oscillator to the main clock, change the operation mode to the medium speed mode (divided by 8 mode) after the clock was divided by 8 (the CM06 bit of CM0 register was set to "1") in the ring oscillator mode.

- **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.

- **PLL Operation Mode**

The main clock multiplied by 2, 4, 6 or 8 provides the PLL clock, and this PLL clock serves as the CPU clock. If the sub clock is on, fc32 can be used as the count source for timers A and B. PLL operation mode can be entered from high speed mode. If PLL operation mode is to be changed to wait or stop mode, first go to high speed mode before changing.

- **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.

- **Low-speed Mode**

The sub clock provides the CPU clock. The main clock is used as the clock source for the peripheral function clock when the CM21 bit is set to "0" (ring oscillator turned off), and the ring oscillator clock is used when the CM21 bit is set to "1" (ring oscillator oscillating).

The fc32 clock can be used as the count source for timers A and B.

- **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. fc32 is the only peripheral function clock available when the CM21 bit is set to "0" (ring oscillator turned off). If the CM21 bit is set to "1" (ring oscillator oscillating), then fc32 and the ring oscillator clock can be used.

Simultaneously when this mode is selected, the CM06 bit of CM0 register 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.

## Clock Generation Circuit

### • Ring Oscillator Mode

The ring oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The ring oscillator clock is also the clock source for the peripheral function clocks. If the sub clock is on, fc32 can be used as the count source for timers A and B.

### • Ring Oscillator Low Power Dissipation Mode

The main clock is turned off after being placed in ring oscillator mode. The CPU clock can be selected as in the ring oscillator mode. The ring oscillator clock is the clock source for the peripheral function clocks. If the sub clock is on, fc32 can be used as the count source for timers A and B. When the operation mode is returned to the high and medium speed modes, set the CM06 bit to "1" (divided by 8 mode).

**Table 1.9.3. Setting Clock Related Bit and Modes**

Modes	CM2 register	CM1 register		CM0 register			
	CM21	CM11	CM17, CM16	CM07	CM06	CM05	CM04
PLL operation mode	0	1	00z	0	0	0	—
High-speed mode	0	0	00z	0	0	0	—
Medium-speed mode	divided by 2	0	0	01z	0	0	—
	divided by 4	0	0	10z	0	0	—
	divided by 8	0	0	—	0	1	—
	divided by 16	0	0	11z	0	0	—
Low-speed mode	—	—	—	1	—	0	1
Low power dissipation mode	—	—	—	1	1 (Note 1)	1 (Note 1)	1
Ring oscillator mode	divided by 1	1	—	00z	0	0	—
	divided by 2	1	—	01z	0	0	—
	divided by 4	1	—	10z	0	0	—
	divided by 8	1	—	—	0	1	—
	divided by 16	1	—	11z	0	0	—
Ring oscillator low power dissipation mode	1	—	(Note 2)	0	(Note 2)	1	—

Note 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.

Note 2: The divide-by-n value can be selected the same way as in ring oscillator mode.

## (2) Wait Mode

In wait mode, the CPU clock is turned off, so are the CPU (because operated by the CPU clock) and the watchdog timer. However, if the PM22 bit of PM2 register is "1" (ring oscillator clock for the watchdog timer count source), the watchdog timer remains active. Because the main clock, sub clock, ring oscillator clock and PLL clock all are on, the peripheral functions using these clocks keep operating.

### • Peripheral Function Clock Stop Function

If the CM02 bit is "1" (peripheral function clocks turned off during wait mode), the f1, f2, f8, f32, f1SIO, f8SIO, f32SIO and fAD clocks are turned off when in wait mode, with the power consumption reduced that much. However, fc32 remains on.

### • Entering Wait Mode

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

If the CM11 bit is "1" (PLL clock for the CPU clock source), set the CM11 bit to "0" (main clock for the CPU clock source) and then the PLC07 bit to "0" (PLL turned off) before entering wait mode.

### • Pin Status During Wait Mode

Table 1.9.4 lists pin status during wait mode

### • 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 "000z" (interrupts disabled) before executing the WAIT instruction.

The peripheral function interrupts are affected by the CM02 bit. If CM02 bit is "0" (peripheral function clocks not turned off during wait mode), all peripheral function interrupts can be used to exit wait



## Clock Generation Circuit

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 1.9.4. Pin Status During Wait Mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
A <sub>0</sub> to A <sub>19</sub> , D <sub>0</sub> to D <sub>15</sub> , $\overline{CS0}$ to $\overline{CS3}$ , $\overline{BHE}$		Retains status before wait mode	/
RD, WR, WRL, WRH		"H"	
HLDA, BCLK		"H"	
ALE		"H"	
I/O ports		Retains status before wait mode	
CLKOUT	When f <sub>c</sub> selected		Does not stop
	When f <sub>8</sub> , f <sub>32</sub> 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.

Table 1.9.5. Interrupts to Exit Wait Mode

Interrupt	CM02=0	CM02=1
NMI interrupt	Can be used	Can be used
Serial I/O interrupt	Can be used when operating with internal or external clock	Can be used when operating with external clock
key input interrupt	Can be used	Can be used
A-D conversion interrupt	Can be used in one-shot mode or single sweep mode	— (Do not use)
Timer A interrupt Timer B interrupt	Can be used in all modes	Can be used in event counter mode or when the count source is f <sub>C32</sub>
INT interrupt	Can be used	Can be used

Table 1.9.5 lists the interrupts to exit wait mode.

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

1. In the ILVL2 to ILVL0 bits of interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit wait mode.

Also, for all of the peripheral function interrupts not used to exit wait mode, set the ILVL2 to ILVL0 bits to "0002" (interrupt disable).

2. Set the I flag to "1".

3. Enable the peripheral function whose interrupt is to be used to exit wait mode.

In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt routine is executed.

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

## Clock Generation Circuit

### (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 V<sub>RAM</sub> or more, the internal RAM is retained. When applying 2.7 or less voltage to Vcc1 and Vcc2 pins, make sure  $V_{cc1} \geq V_{cc2} \geq V_{RAM}$ .

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

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

#### • Entering Stop Mode

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

Before entering stop mode, set the CM20 bit to "0" (oscillation stop, re-oscillation detection function disable).

Also, if the CM11 bit is "1" (PLL clock for the CPU clock source), set the CM11 bit to "0" (main clock for the CPU clock source) and the PLC07 bit to "0" (PLL turned off) before entering stop mode.

#### • Pin Status in Stop Mode

Table 1.9.6 lists pin status during stop mode

#### • Exiting Stop Mode

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

If the microcomputer is to be moved out of stop mode by a hardware reset or  $\overline{NMI}$  interrupt, set the peripheral function interrupt priority ILVL2 to ILVL0 bits to "0002" (interrupts disable) before setting the CM10 bit to "1".

If the microcomputer is to be moved out of stop mode by a peripheral function interrupt, set up the following before setting the CM10 bit to "1".

1. In the ILVL2 to ILVL0 bits of interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit stop mode.

Also, for all of the peripheral function interrupts not used to exit stop mode, set the ILVL2 to ILVL0 bits to "0002".

2. Set the I flag to "1".

3. Enable the peripheral function whose interrupt is to be used to exit stop mode.

In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt service routine is executed.

Which CPU clock will be used after exiting stop mode by a peripheral function or  $\overline{NMI}$  interrupt is determined by the CPU clock that was on when the microcomputer was placed into stop mode as follows:

If the CPU clock before entering stop mode was derived from the sub clock: sub clock

If the CPU clock before entering stop mode was derived from the main clock: main clock divide-by-8

If the CPU clock before entering stop mode was derived from the ring oscillator clock: ring oscillator clock divide-by-8

## Clock Generation Circuit

Table 1.9.6. Pin Status in Stop Mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
A <sub>0</sub> to A <sub>19</sub> , D <sub>0</sub> to D <sub>15</sub> , $\overline{CS0}$ to $\overline{CS3}$ , $\overline{BHE}$		Retains status before stop mode	/
RD, WR, $\overline{WRL}$ , $\overline{WRH}$		"H"	
$\overline{HLDA}$ , BCLK		"H"	
ALE		"H"	
I/O ports		Retains status before stop mode	
CLKOUT	When fc selected		"H"
	When f8, f32 selected		Retains status before stop mode

Clock Generation Circuit

Figure 1.9.10 shows the state transition from normal operation mode to stop mode and wait mode. Figure 1.9.11 shows the state transition in normal operation mode.

Table 1.9.7 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line shows state after transition.

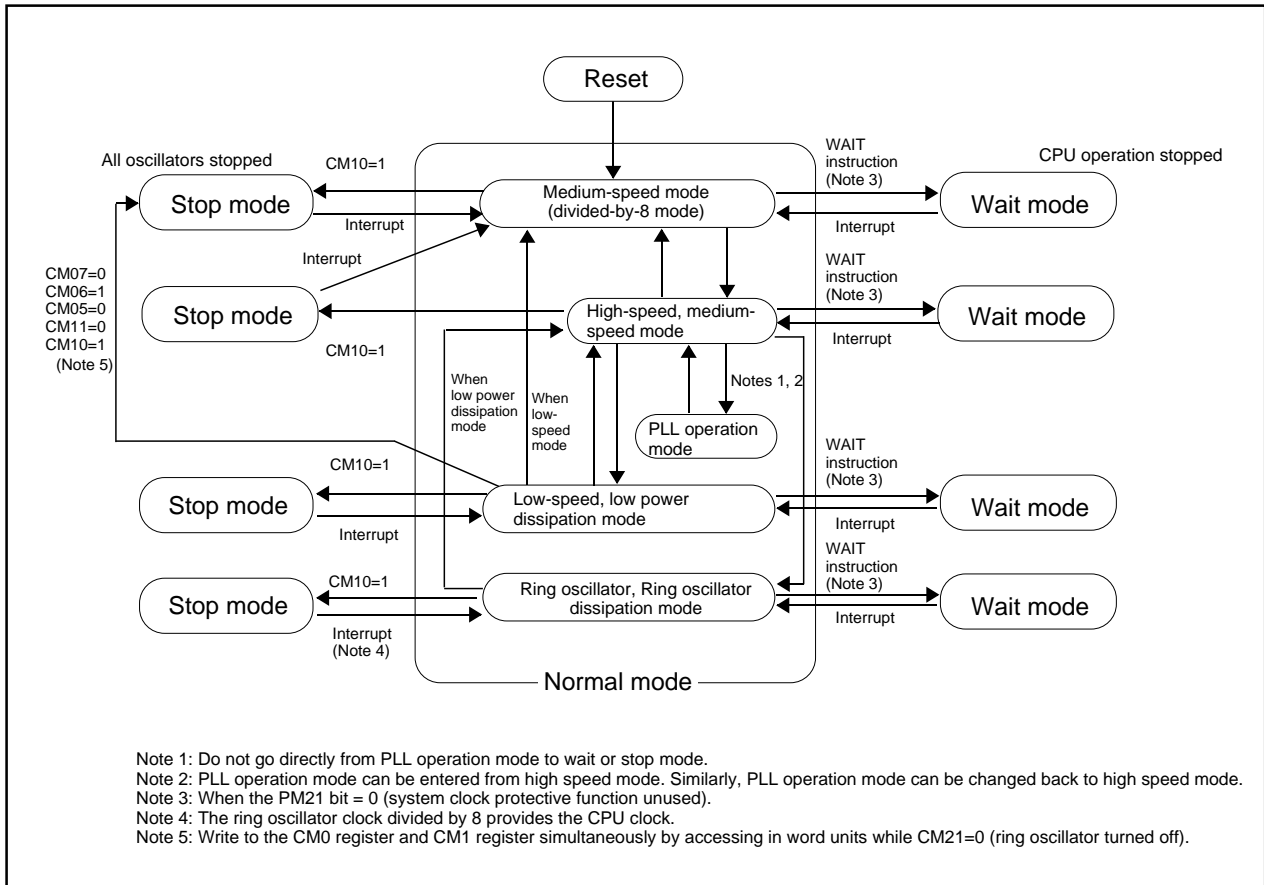


Figure 1.9.10. State Transition to Stop Mode and Wait Mode

Clock Generation Circuit

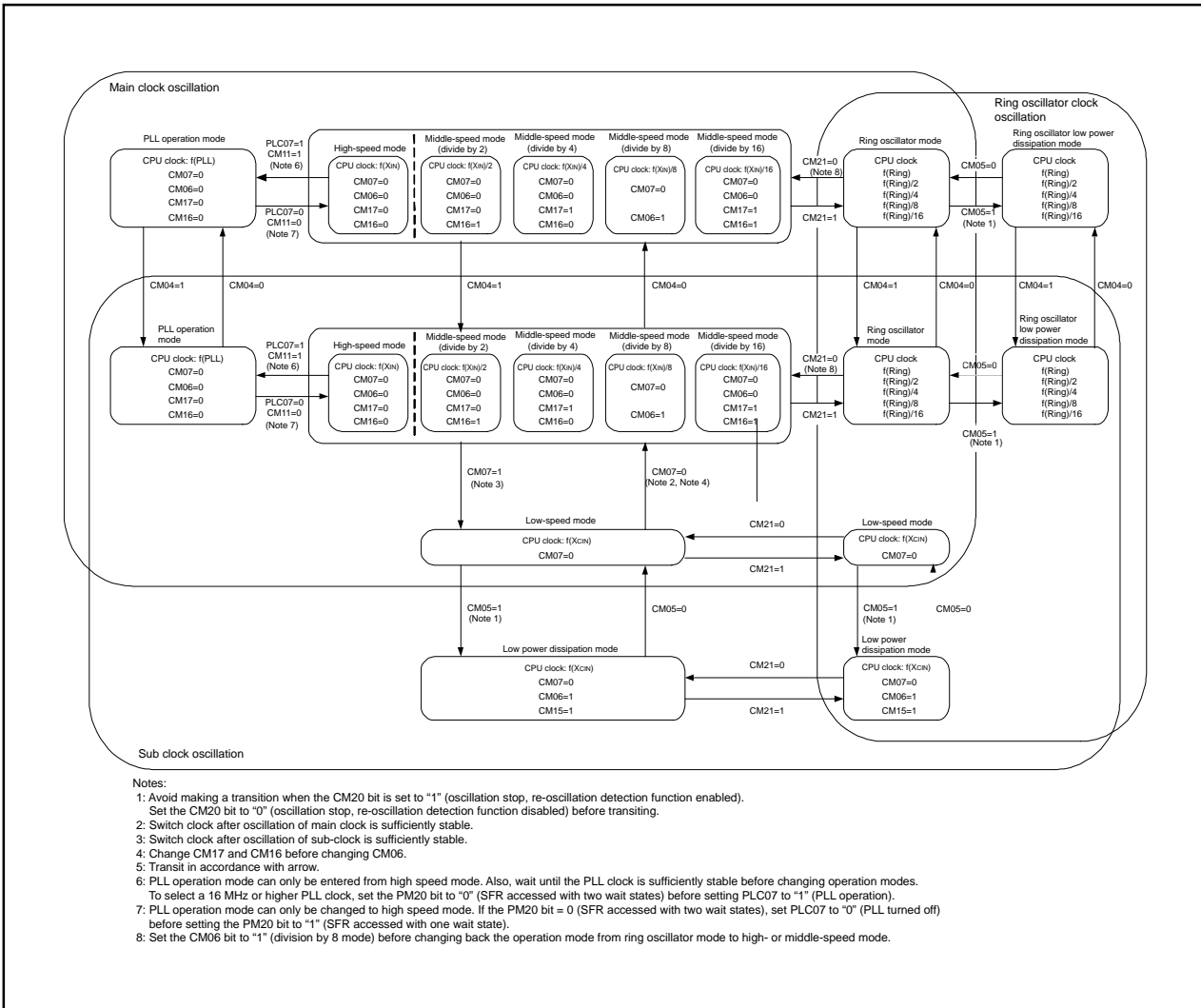


Figure 1.9.11. State Transition in Normal Mode

Clock Generation Circuit

Table 1.9.7. Allowed Transition and Setting

		State after transition							
		High-speed mode, middle-speed mode	Low-speed mode <sup>1</sup>	Low power dissipation mode <sup>1</sup>	PLL operation mode <sup>1</sup>	Ring oscillator mode	Ring oscillator low power dissipation mode	Stop mode	Wait mode
Current state	High-speed mode, middle-speed mode	See Table A	(9) <sup>7</sup>	--	(13) <sup>3</sup>	(15)	--	(16) <sup>1</sup>	(17)
	Low-speed mode <sup>2</sup>	(8)		(11) <sup>1, 6</sup>	--	--	--	(16) <sup>1</sup>	(17)
	Low power dissipation mode <sup>2</sup>	--	(10)		--	--	--	(16) <sup>1</sup>	(17)
	PLL operation mode <sup>2</sup>	(12) <sup>3</sup>	--	--		--	--	--	--
	Ring oscillator mode	(14) <sup>4</sup>	--	--	--	See Table A	(11) <sup>1</sup>	(16) <sup>1</sup>	(17)
	Ring oscillator low power dissipation mode	--	--	--	--	(10)	See Table A	(16) <sup>1</sup>	(17)
	Stop mode	(18) <sup>5</sup>	(18)	(18)	--	(18) <sup>5</sup>	(18) <sup>5</sup>		--
	Wait mode	(18)	(18)	(18)	--	(18)	(18)	--	

--: Cannot transit

Table 1. State Transition with Main Clock Division Ratio in High- or Middle-speed Mode, Ring Oscillator Mode, and Ring Oscillator Low Power Dissipation Mode

		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 16	--	--	--	--	(2)	(3)	(4)	(5)	(7)	

--: Cannot transit

Notes:

1. Avoid making a transition when the CM21 bit is set to "1" (oscillation stop, re-oscillation detection function enabled). Set the CM21 bit to "0" (oscillation stop, re-oscillation detection function disabled) before transiting.
2. Ring oscillator clock oscillates and stops in low-speed mode and low power dissipation mode. In these mode, the ring oscillator can be used as peripheral function clock.  
Sub clock oscillates and stops in PLL operation mode. In this mode, sub clock can be used as peripheral function clock.
3. PLL operation mode can only be entered from and changed to high-speed mode.
4. Set the CM06 bit to "1" (division by 8 mode) before transiting from ring oscillator mode to high- or middle-speed mode.
5. When exiting stop mode, the CM06 bit is set to "1" (division by 8 mode).
6. If the CM05 bit is set to "1" (main clock stop), then the CM06 bit is set to "1" (division by 8 mode).
7. A transition can be made only when sub clock is oscillating.

Table B. Setting and Operation

	Setting	Operation
(1)	CM04 = 0	Sub clock turned off
(2)	CM04 = 1	Sub clock oscillating
(3)	CM06 = 0, CM17 = 0, CM16 = 0	CPU clock no division mode
(4)	CM06 = 0, CM17 = 0, CM16 = 1	CPU clock division by 2 mode
(5)	CM06 = 0, CM17 = 1, CM16 = 0	CPU clock division by 4 mode
(6)	CM06 = 0, CM17 = 1, CM16 = 1	CPU clock division by 16 mode
(7)	CM06 = 1	CPU clock division by 8 mode
(8)	CM07 = 0	Main clock, PLL clock, or ring oscillator clock selected
(9)	CM07 = 1	Sub clock selected
(10)	CM05 = 0	Main clock oscillating
(11)	CM05 = 1	Main clock turned off
(12)	PLC07 = 0, CM11 = 0	Main clock selected
(13)	PLC07 = 1, CM11 = 1	PLL clock selected
(14)	CM21 = 0	Main clock or PLL clock selected
(15)	CM21 = 1	Ring oscillator clock selected
(16)	CM10 = 1	Transition to stop mode
(17)	wait	Transition to wait mode
(18)	Hardware interrupt	Exit stop mode or wait mode

## Clock Generation Circuit

---

### System Clock Protective Function

When the main clock is selected for the CPU clock source, this function disables the clock against modifications in order to prevent the CPU clock from becoming halted by run-away.

If the PM21 bit of PM2 register is set to "1" (clock modification disabled), the following bits are protected against writes:

- CM02, CM05, and CM07 bits in CM0 register
- CM10, CM11 bits in CM1 register
- CM20 bit in CM2 register
- All bits in PLC0 register

Before the system clock protective function can be used, the following register settings must be made while the CM05 bit of CM0 register is "0" (main clock oscillating) and CM07 bit is "0" (main clock selected for the CPU clock source):

- (1) Set the PRC1 bit of PRCR register to "1" (enable writes to PM2 register).
- (2) Set the PM21 bit of PM2 register to "1" (disable clock modification).
- (3) Set the PRC1 bit of PRCR register to "0" (disable writes to PM2 register).

Do not execute the WAIT instruction when the PM21 bit is "1".

## Clock Generation Circuit

**Oscillation Stop and Re-oscillation Detect Function**

The oscillation stop and re-oscillation detect function is such that main clock oscillation circuit stop and re-oscillation are detected. At oscillation stop, re-oscillation detection, reset or oscillation stop, re-oscillation detection interrupt are generated. Which is to be generated can be selected using the CM27 bit of CM2 register. Table 1.9.4 lists an specification overview of the oscillation stop and re-oscillation detect function.

**Table 1.9.7. Specification Overview of Oscillation Stop and Re-oscillation Detect Function**

Item	Specification
Oscillation stop detectable clock and frequency bandwidth	$f(X_{IN}) \geq 2 \text{ MHz}$
Enabling condition for oscillation stop, re-oscillation detection function	Set CM20 bit to "1"(enable)
Operation at oscillation stop, re-oscillation detection	<ul style="list-style-type: none"> <li>•Reset occurs (when CM27 bit =0)</li> <li>•Oscillation stop, re-oscillation detection interrupt occurs(when CM27 bit =1)</li> </ul>

**(1) Operation When CM27 bit = 0 (Oscillation Stop Detection Reset)**

Where main clock stop is detected when the CM20 bit is "1" (oscillation stop, re-oscillation detection function enabled), the microcomputer is initialized, coming to a halt (oscillation stop reset; refer to "SFR", "Reset").

This status is reset with hardware reset 1 or hardware reset 2. Also, even when re-oscillation is detected, the microcomputer can be initialized and stopped; it is, however, necessary to avoid such usage. (During main clock stop, do not set the CM20 bit to "1" and the CM27 bit to "0".)

**(2) Operation When CM27 bit = 0 (Oscillation Stop and Re-oscillation Detect Interrupt)**

Where the main clock corresponds to the CPU clock source and the CM20 bit is "1" (oscillation stop and re-oscillation detect function enabled), the system is placed in the following state if the main clock comes to a halt:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- The ring oscillator starts oscillation, and the ring oscillator clock becomes the CPU clock and clock source for peripheral functions in place of the main clock.
- CM21 bit = 1 (ring oscillator clock for CPU clock source)
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)

Where the PLL clock corresponds to the CPU clock source and the CM20 bit is "1", the system is placed in the following state if the main clock comes to a halt: Since the CM21 bit remains unchanged, set it to "1" (ring oscillator clock) inside the interrupt routine.

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)
- CM21 bit remains unchanged

Where the CM20 bit is "1", the system is placed in the following state if the main clock re-oscillates from the stop condition:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock re-oscillation detected)
- CM23 bit = 0 (main clock oscillation)
- CM21 bit remains unchanged



## Clock Generation Circuit

**How to Use Oscillation Stop and Re-oscillation Detect Function**

- The oscillation stop and re-oscillation detect interrupt shares the vector with the watchdog timer interrupt. If the re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, return the main clock to the CPU clock and peripheral function clock source in the program. Figure 1.9.12 shows the procedure for switching the clock source from the ring oscillator to the main clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt occurrence, the CM22 bit becomes "1". When the CM22 bit is set at "1", oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to "0" in the program, oscillation stop, re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is "1", an oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the ring oscillator starts oscillating. In this case, although the CPU clock is derived from the sub clock as it was before the interrupt occurred, the peripheral function clocks now are derived from the ring oscillator clock.
- To enter wait mode while using the oscillation stop, re-oscillation detection function, set the CM02 bit to "0" (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop, re-oscillation detection function is provided in preparation for main clock stop due to external factors, set the CM20 bit to "0" (Oscillation stop, re-oscillation detection function disabled) where the main clock is stopped or oscillated in the program, that is where the stop mode is selected or the CM05 bit is altered.
- This function cannot be used if the main clock frequency is 2 MHz or less. In that case, set the CM20 bit to "0".

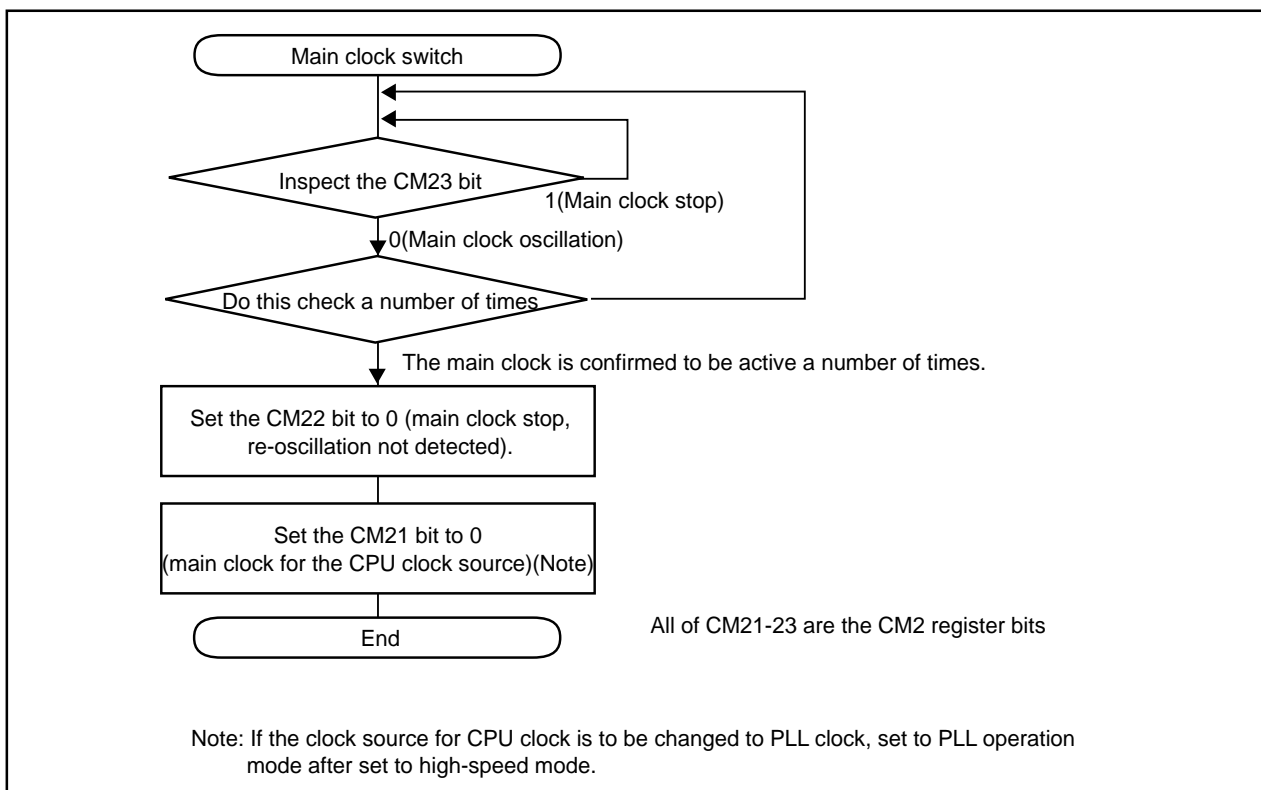


Figure 1.9.12. Procedure to Switch Clock Source From Ring Oscillator to Main Clock

## Protection

## 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 1.10.1 shows the PRCR register. The following lists the registers protected by the PRCR register.

- Registers protected by PRC0 bit: CM0, CM1, CM2, PLC0 and PCLKR registers
- Registers protected by PRC1 bit: PM0, PM1, PM2, TB2SC, INVC0 and INVC1 registers
- Registers protected by PRC2 bit: PD9, S3C and S4C registers
- Registers protected by PRC3 bit: VCR2 and D4INT registers

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, PRC1 and PRC3 bits are not automatically cleared to "0" by writing to any address. They can only be cleared in a program.

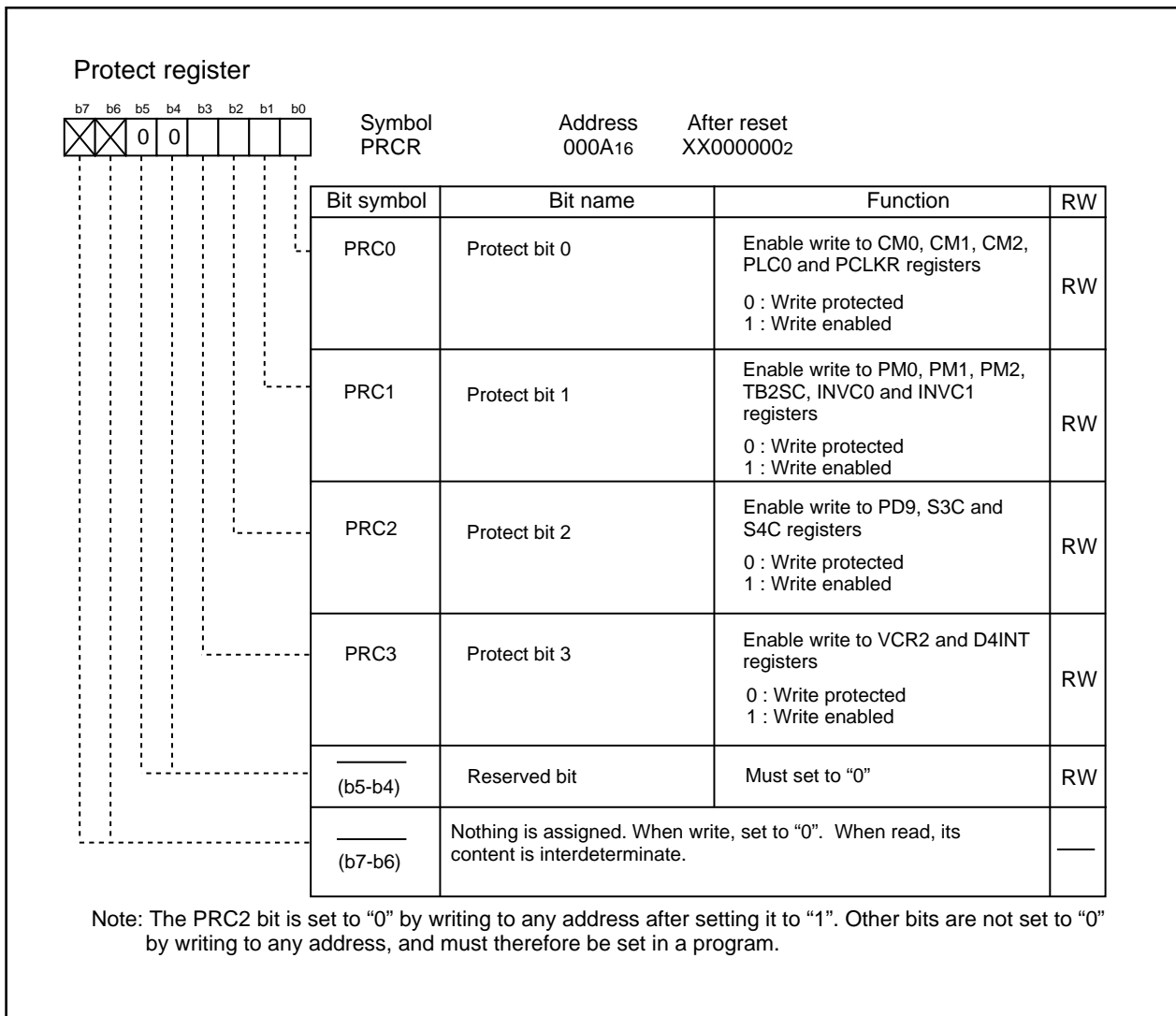


Figure 1.10.1. PRCR Register

# Interrupts

## Type of Interrupts

Figure 1.11.1 shows types of interrupts.

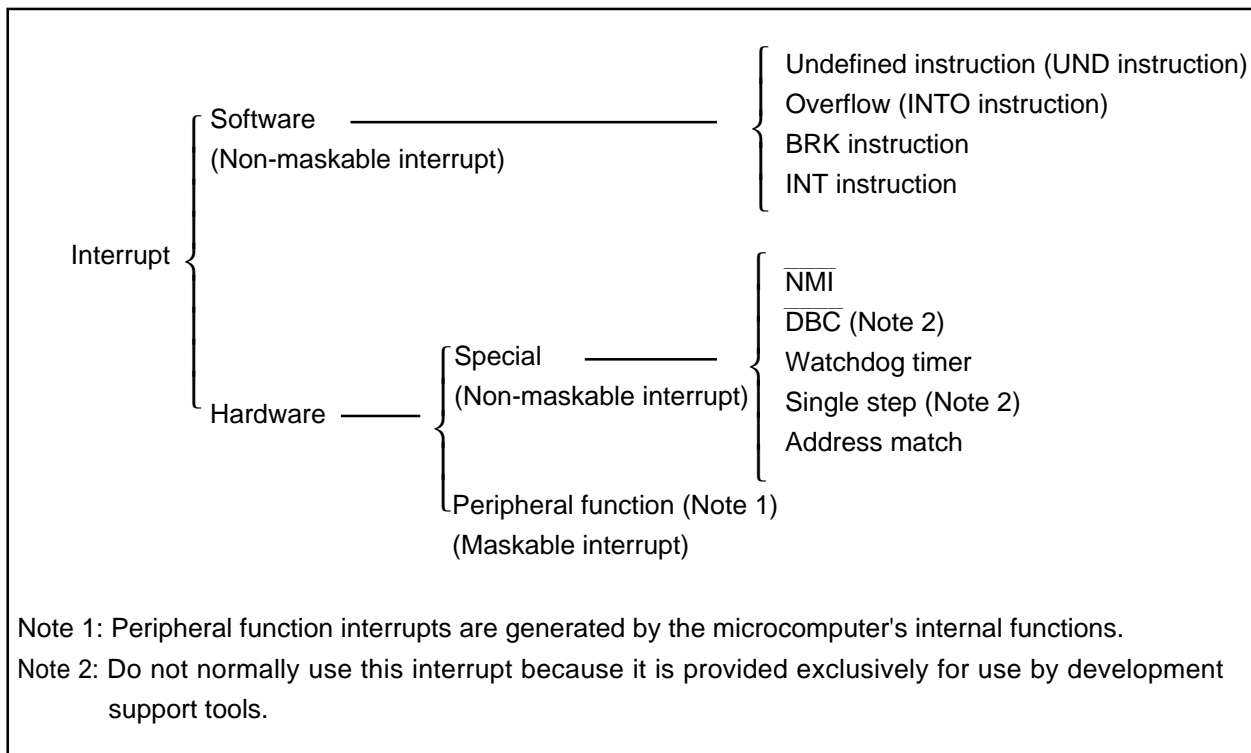


Figure 1.11.1. Interrupts

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

## Software Interrupts

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

- **Undefined Instruction Interrupt**

An undefined instruction interrupt occurs when executing the UND instruction.

- **Overflow Interrupt**

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

- **BRK Interrupt**

A BRK interrupt occurs when executing the BRK instruction.

- **INT Instruction Interrupt**

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

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

## Hardware Interrupts

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

### (1) Special Interrupts

Special interrupts are non-maskable interrupts.

- **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 section "NMI interrupt".

- **DBC Interrupt**

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

- **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 section "watchdog timer".

- **Oscillation Stop and Re-oscillation Detection Interrupt**

Generated by the oscillation stop and re-oscillation detection function. For details about the oscillation stop detection function, refer to the section "clock generating circuit".

- **Power Supply Down Detection Interrupt**

Generated by the voltage detection circuit. For details about the voltage detection circuit, refer to the section "voltage detection circuit".

- **Single-step Interrupt**

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

- **Address Match Interrupt**

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMAD0 to RMAD3 register that corresponds to one of the AIER register's AIER0 or AIER1 bit or the AIER2 register's AIER20 or AIER21 bit which is "1" (address match interrupt enabled). For details about the address match interrupt, refer to the section "address match interrupt".

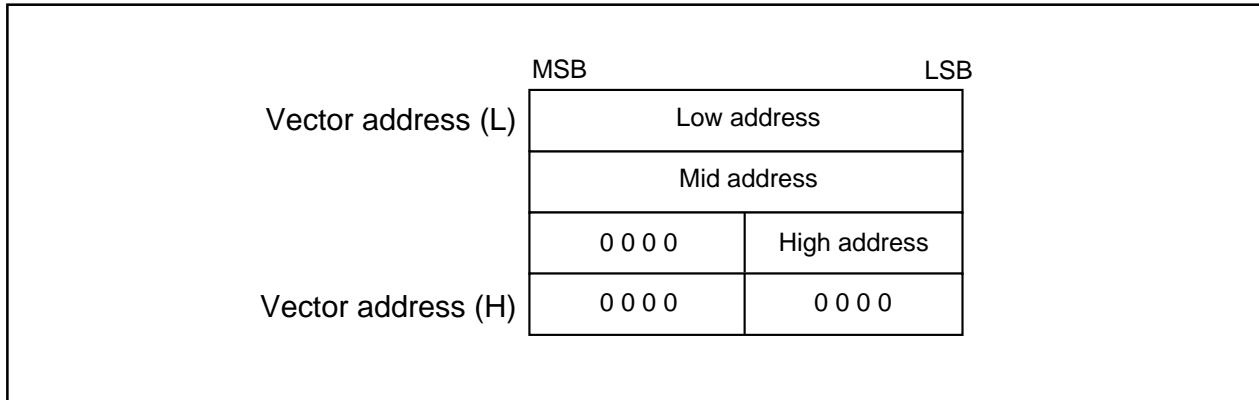
### (2) Peripheral Function Interrupts

Peripheral function interrupts are maskable interrupts and generated by the microcomputer's internal functions. The interrupt sources for peripheral function interrupts are listed in Table 1.11.2. For details about the peripheral functions, refer to the description of each peripheral function in this manual.

## Interrupts

### 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 1.11.2 shows the interrupt vector.



**Figure 1.11.2. Interrupt Vector**

- **Fixed Vector Tables**

The fixed vector tables are allocated to the addresses from FFFDC<sub>16</sub> to FFFFF<sub>16</sub>. Table 1.11.1 lists the fixed vector tables. In the flash memory version of microcomputer, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to the section "flash memory rewrite disabling function".

**Table 1.11.1. Fixed Vector Tables**

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks	Reference
Undefined instruction	FFFD <sub>16</sub> C to FFFD <sub>16</sub> F	Interrupt on UND instruction	M16C/60, M16C/20 serise software maual
Overflow	FFFE <sub>16</sub> 0 to FFFE <sub>16</sub> 3	Interrupt on INTO instruction	
BRK instruction	FFFE <sub>16</sub> 4 to FFFE <sub>16</sub> 7	If the contents of address FFFE <sub>16</sub> 7 is FF <sub>16</sub> , program execution starts from the address shown by the vector in the relocatable vector table.	
Address match	FFFE <sub>16</sub> 8 to FFFE <sub>16</sub> B		Address match interrupt
Single step (Note)	FFFE <sub>16</sub> C to FFFE <sub>16</sub> F		
Watchdog timer Oscillation stop and re-oscillation detection Power supply down detection	FFFF <sub>16</sub> 0 to FFFF <sub>16</sub> 3		Watchdog timer  Clock generating circuit  Voltage detection circuit
DBC (Note)	FFFF <sub>16</sub> 4 to FFFF <sub>16</sub> 7		
NMI	FFFF <sub>16</sub> 8 to FFFF <sub>16</sub> B		NMI interrupt
Reset	FFFF <sub>16</sub> C to FFFF <sub>16</sub> F		Reset

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

## Interrupts

## • Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. Table 1.11.2 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 1.11.2. Relocatable Vector Tables

Interrupt source	Vector address (Note 1) Address (L) to address (H)	Software interrupt number	Reference
BRK instruction (Note 5)	+0 to +3 (0000 <sub>16</sub> to 0003 <sub>16</sub> )	0	M16C/60, M16C/20 series software manual
———— (Reserved)		1 to 3	
$\overline{\text{INT3}}$	+16 to +19 (0010 <sub>16</sub> to 0013 <sub>16</sub> )	4	$\overline{\text{INT}}$ interrupt
Timer B5	+20 to +23 (0014 <sub>16</sub> to 0017 <sub>16</sub> )	5	Timer
Timer B4, UART1 bus collision (Note 4) detect	+24 to +27 (0018 <sub>16</sub> to 001B <sub>16</sub> )	6	Timer Serial I/O
Timer B3, UART0 bus collision (Note 4) detect	+28 to +31 (001C <sub>16</sub> to 001F <sub>16</sub> )	7	
SI/O4, $\overline{\text{INT5}}$ (Note 2)	+32 to +35 (0020 <sub>16</sub> to 0023 <sub>16</sub> )	8	$\overline{\text{INT}}$ interrupt
SI/O3, $\overline{\text{INT4}}$ (Note 2)	+36 to +39 (0024 <sub>16</sub> to 0027 <sub>16</sub> )	9	Serial I/O
UART 2 bus collision detection	+40 to +43 (0028 <sub>16</sub> to 002B <sub>16</sub> )	10	Serial I/O
DMA0	+44 to +47 (002C <sub>16</sub> to 002F <sub>16</sub> )	11	DMAC
DMA1	+48 to +51 (0030 <sub>16</sub> to 0033 <sub>16</sub> )	12	
Key input interrupt	+52 to +55 (0034 <sub>16</sub> to 0037 <sub>16</sub> )	13	Key input interrupt
A-D	+56 to +59 (0038 <sub>16</sub> to 003B <sub>16</sub> )	14	A-D convertor
UART2 transmit, NACK2 (Note 3)	+60 to +63 (003C <sub>16</sub> to 003F <sub>16</sub> )	15	Serial I/O
UART2 receive, ACK2 (Note 3)	+64 to +67 (0040 <sub>16</sub> to 0043 <sub>16</sub> )	16	
UART0 transmit, NACK0 (Note 3)	+68 to +71 (0044 <sub>16</sub> to 0047 <sub>16</sub> )	17	
UART0 receive, ACK0 (Note 3)	+72 to +75 (0048 <sub>16</sub> to 004B <sub>16</sub> )	18	
UART1 transmit, NACK1 (Note 3)	+76 to +79 (004C <sub>16</sub> to 004F <sub>16</sub> )	19	
UART1 receive, ACK1 (Note 3)	+80 to +83 (0050 <sub>16</sub> to 0053 <sub>16</sub> )	20	
Timer A0	+84 to +87 (0054 <sub>16</sub> to 0057 <sub>16</sub> )	21	Timer
Timer A1	+88 to +91 (0058 <sub>16</sub> to 005B <sub>16</sub> )	22	
Timer A2	+92 to +95 (005C <sub>16</sub> to 005F <sub>16</sub> )	23	
Timer A3	+96 to +99 (0060 <sub>16</sub> to 0063 <sub>16</sub> )	24	
Timer A4	+100 to +103 (0064 <sub>16</sub> to 0067 <sub>16</sub> )	25	
Timer B0	+104 to +107 (0068 <sub>16</sub> to 006B <sub>16</sub> )	26	
Timer B1	+108 to +111 (006C <sub>16</sub> to 006F <sub>16</sub> )	27	
Timer B2	+112 to +115 (0070 <sub>16</sub> to 0073 <sub>16</sub> )	28	
$\overline{\text{INT0}}$	+116 to +119 (0074 <sub>16</sub> to 0077 <sub>16</sub> )	29	$\overline{\text{INT}}$ interrupt
$\overline{\text{INT1}}$	+120 to +123 (0078 <sub>16</sub> to 007B <sub>16</sub> )	30	
$\overline{\text{INT2}}$	+124 to +127 (007C <sub>16</sub> to 007F <sub>16</sub> )	31	
Software interrupt (Note 5)	+128 to +131 (0080 <sub>16</sub> to 0083 <sub>16</sub> ) to +252 to +255 (00FC <sub>16</sub> to 00FF <sub>16</sub> )	32 to 63	M16C/60, M16C/20 series software manual

Note 1: Address relative to address in INTB.

Note 2: Use the IFSR register's IFSR6 and IFSR7 bits to select.

Note 3: During I<sup>2</sup>C mode, NACK and ACK interrupts comprise the interrupt source.

Note 4: Use the IFSR2A register's IFSR26 and IFSR27 bits to select.

Note 5: These interrupts cannot be disabled using the I flag.

## Interrupts

---

### Interrupt Control

The following describes how to enable/disable the maskable interrupts, and how to set the priority in which order they are accepted. What is explained here does not apply to nonmaskable interrupts.

Use the FLG register's I flag, IPL, and each interrupt control register's ILVL2 to ILVL0 bits to enable/disable the maskable interrupts. Whether an interrupt is requested is indicated by the IR bit in each interrupt control register.

Figure 1.11.3 shows the interrupt control registers.



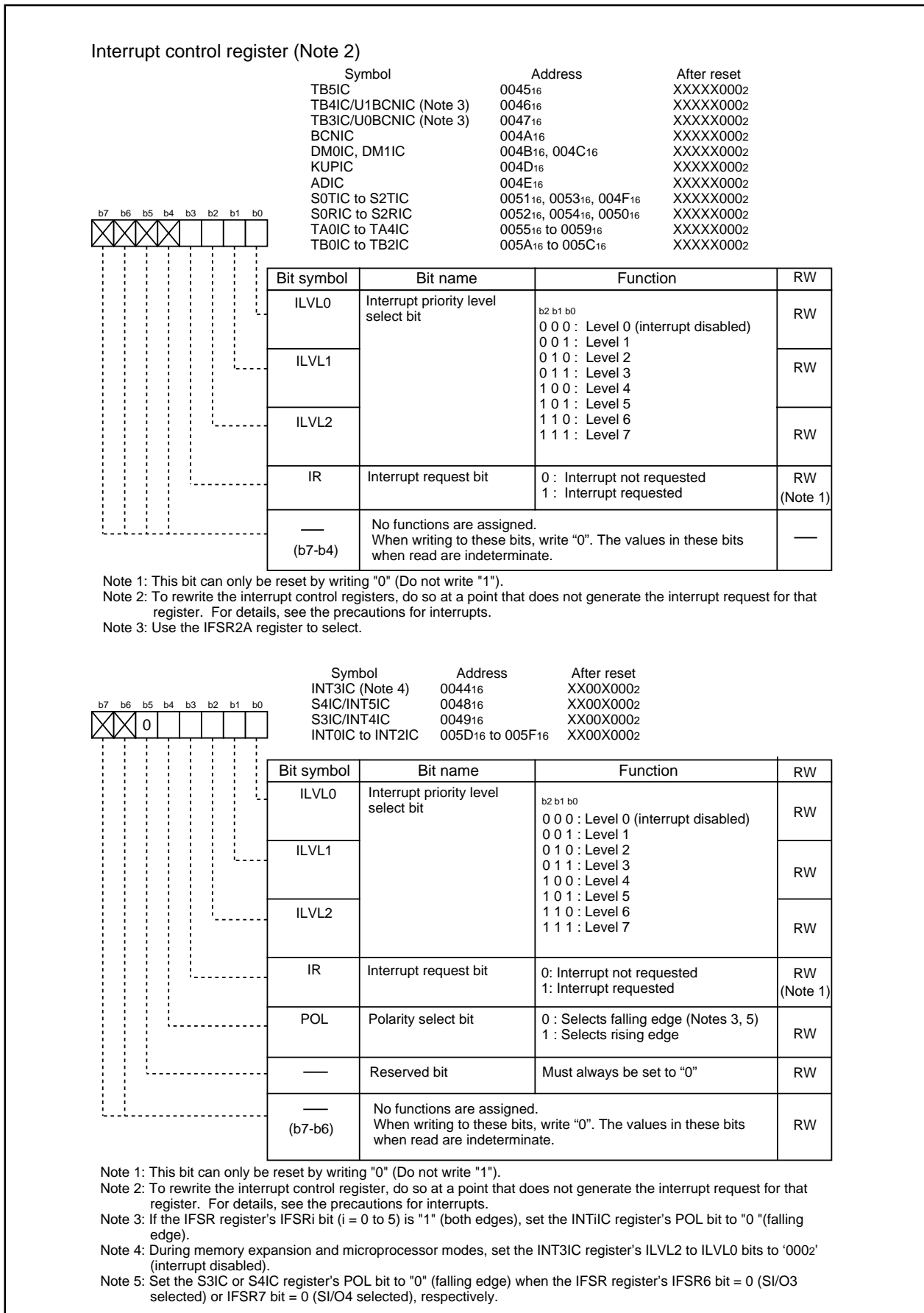


Figure 1.11.3. Interrupt Control Registers

## Interrupts

### I Flag

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

### IR Bit

The IR bit is set to "1" (= interrupt requested) when an interrupt request is generated. Then, when the interrupt request is accepted and the CPU branches to the corresponding interrupt vector, the IR bit is cleared to "0" (= interrupt not requested).

The IR bit can be cleared to "0" in a program. Note that do not write "1" to this bit.

### ILVL2 to ILVL0 Bits and IPL

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

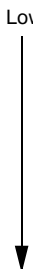
Table 1.11.3 shows the settings of interrupt priority levels and Table 1.11.4 shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:

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

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

**Table 1.11.3. Settings of Interrupt Priority Levels**

ILVL2 to ILVL0 bits	Interrupt priority level	Priority order
0002	Level 0 (interrupt disabled)	————
0012	Level 1	Low  High
0102	Level 2	
0112	Level 3	
1002	Level 4	
1012	Level 5	
1102	Level 6	
1112	Level 7	

**Table 1.11.4. Interrupt Priority Levels Enabled by IPL**

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

**Interrupts**

**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.

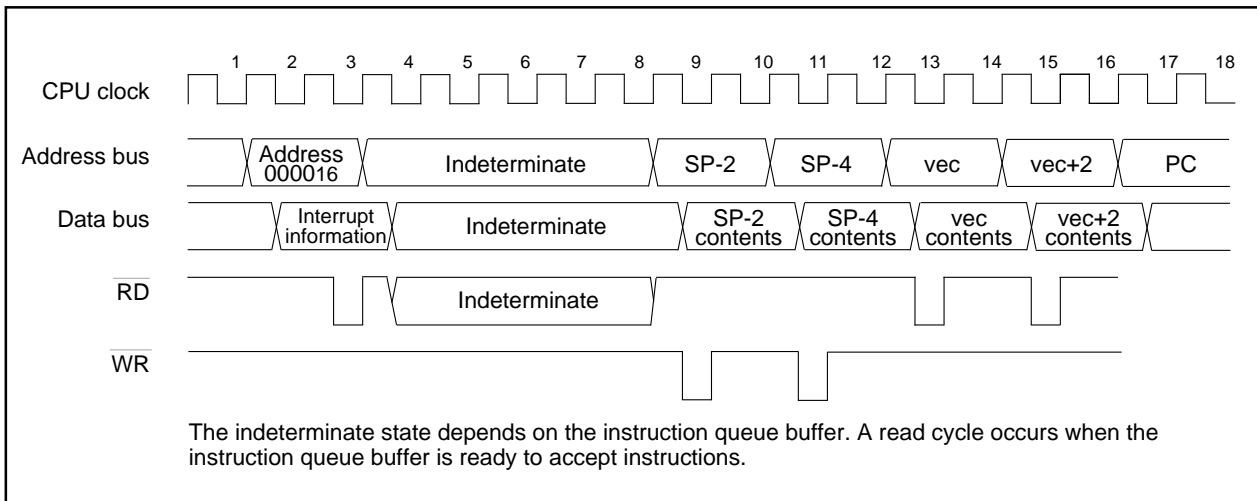
If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instruction, 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 1.11.4 shows time required for executing the interrupt sequence.

- (1) The CPU gets interrupt information (interrupt number and interrupt request priority level) by reading the address 0000016. Then it clears the IR bit for the corresponding interrupt to “0” (interrupt not requested).
- (2) The FLG register immediately before entering the interrupt sequence is saved to the CPU’s internal temporary register<sup>(Note 1)</sup>.
- (3) The I, D and U flags in the FLG register become as follows:  
 The I flag is cleared to “0” (interrupts disabled).  
 The D flag is cleared to “0” (single-step interrupt disabled).  
 The U flag is cleared to “0” (ISP selected).  
 However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.
- (4) The CPU’s internal temporary register <sup>(Note 1)</sup> is saved to the stack.
- (5) The PC is saved to the stack.
- (6) The interrupt priority level of the accepted interrupt is set in the IPL.
- (7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

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

Note: This register cannot be used by user.

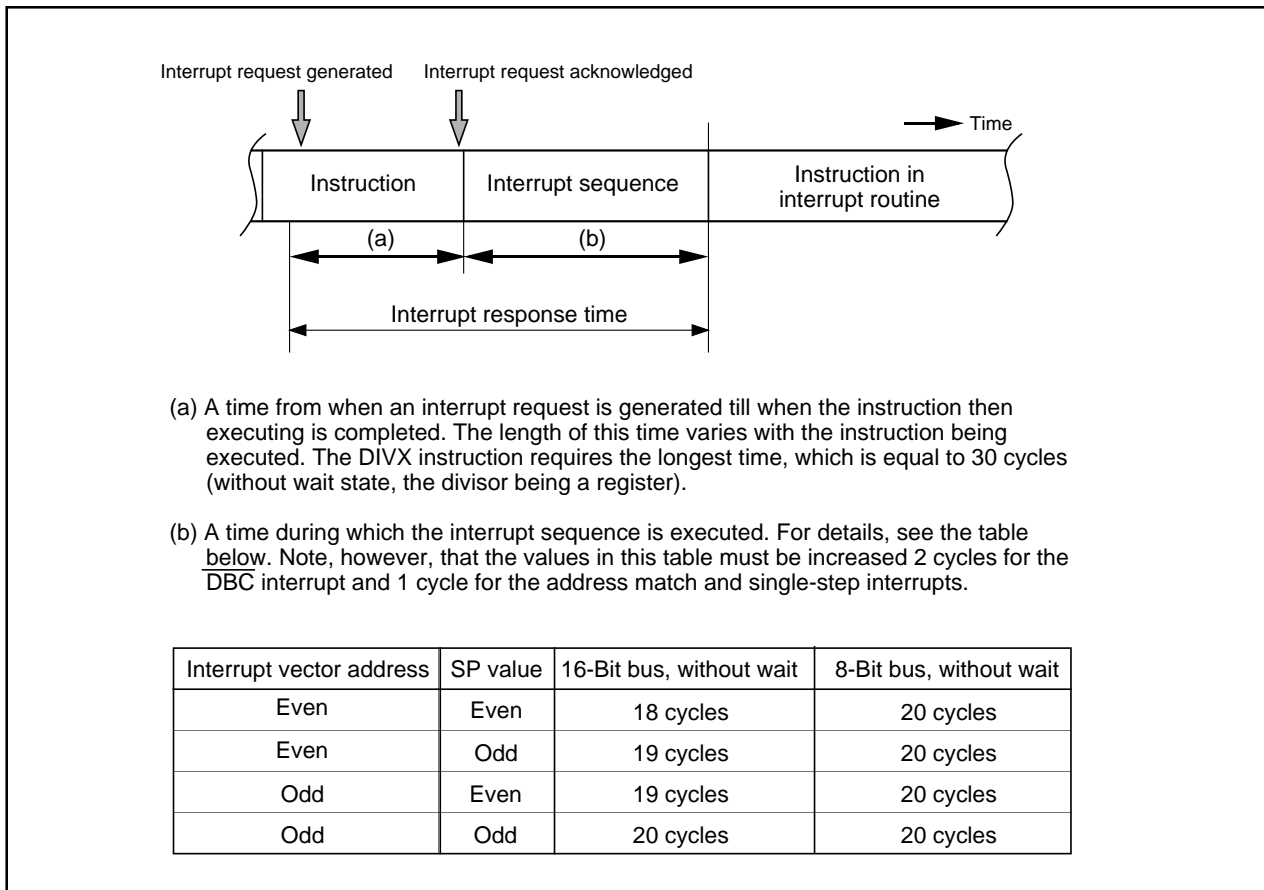


**Figure 1.11.4. Time Required for Executing Interrupt Sequence**

**Interrupts**

**Interrupt Response Time**

Figure 1.11.5 shows the interrupt response time. The interrupt response or interrupt acknowledge time denotes a time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of a time from when an interrupt request is generated till when the instruction then executing is completed ((a) in Figure 1.11.5) and a time during which the interrupt sequence is executed ((b) in Figure 1.11.5).



**Figure 1.11.5. Interrupt response time**

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

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

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in Table 1.11.5 is set in the IPL. Shown in Table 1.11.5 are the IPL values of software and special interrupts when they are accepted.

**Table 1.11.5. 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

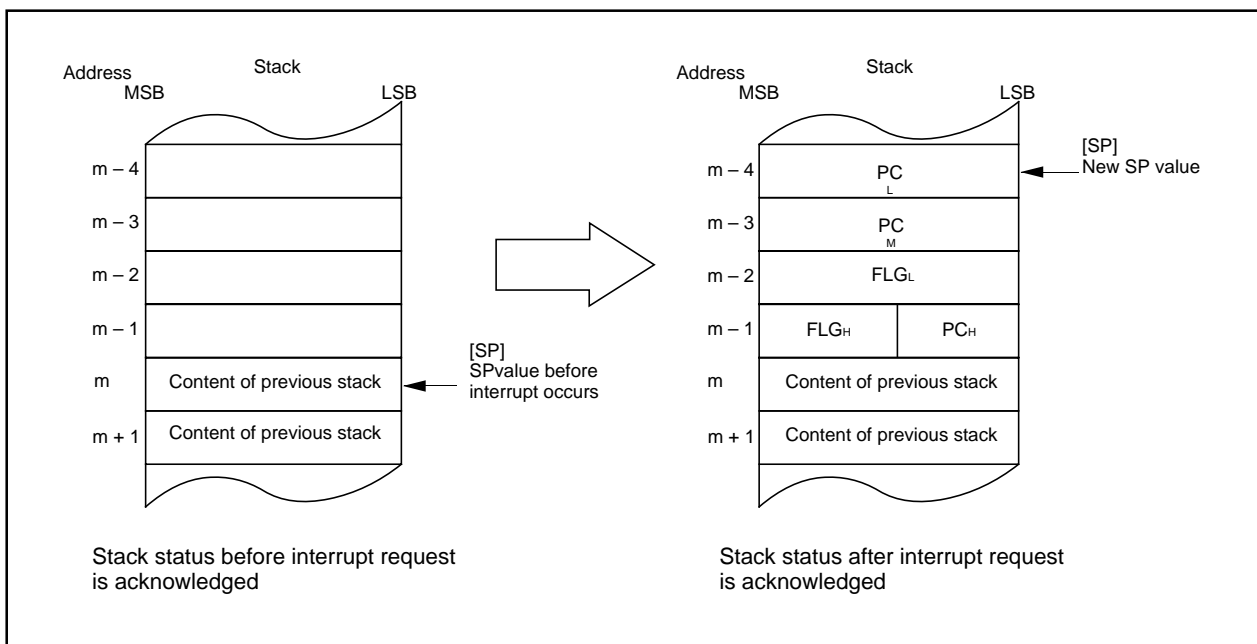
## Interrupts

### 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 of 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 1.11.6 shows the stack status before and after an interrupt request is accepted.

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

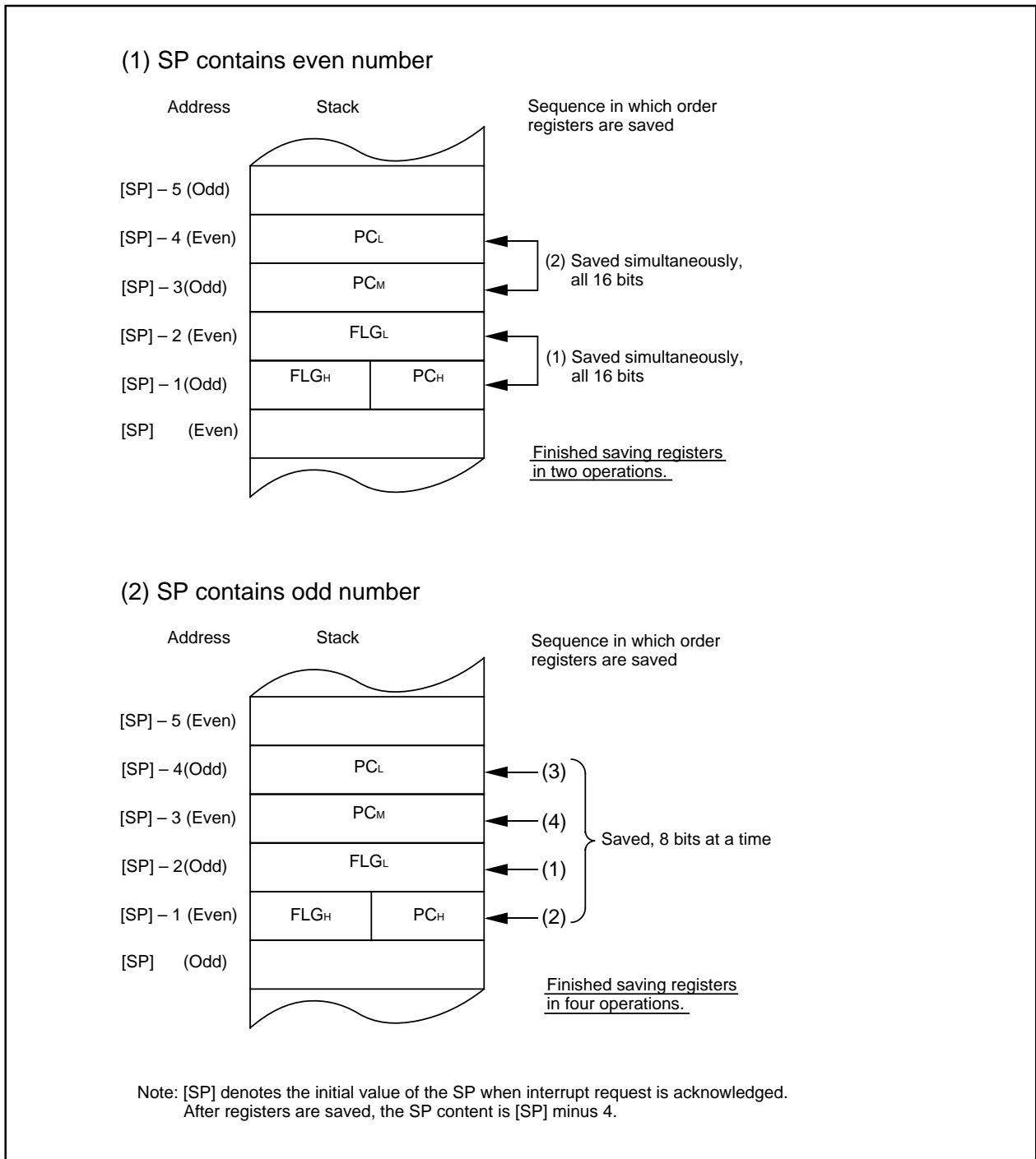


**Figure 1.11.6. Stack Status Before and After Acceptance of Interrupt Request**

**Interrupts**

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP<sup>(Note)</sup>, at the time of acceptance of an interrupt request, is even or odd. If the stack pointer <sup>(Note)</sup> 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 1.11.7 shows the operation of the saving registers.

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



**Figure 1.11.7. Operation of Saving Register**

## Returning from an Interrupt Routine

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

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

## Interrupt Priority

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

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

The watchdog timer and other special interrupts have their priority levels set in hardware. Figure 1.11.8 shows the priorities of hardware interrupts.

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

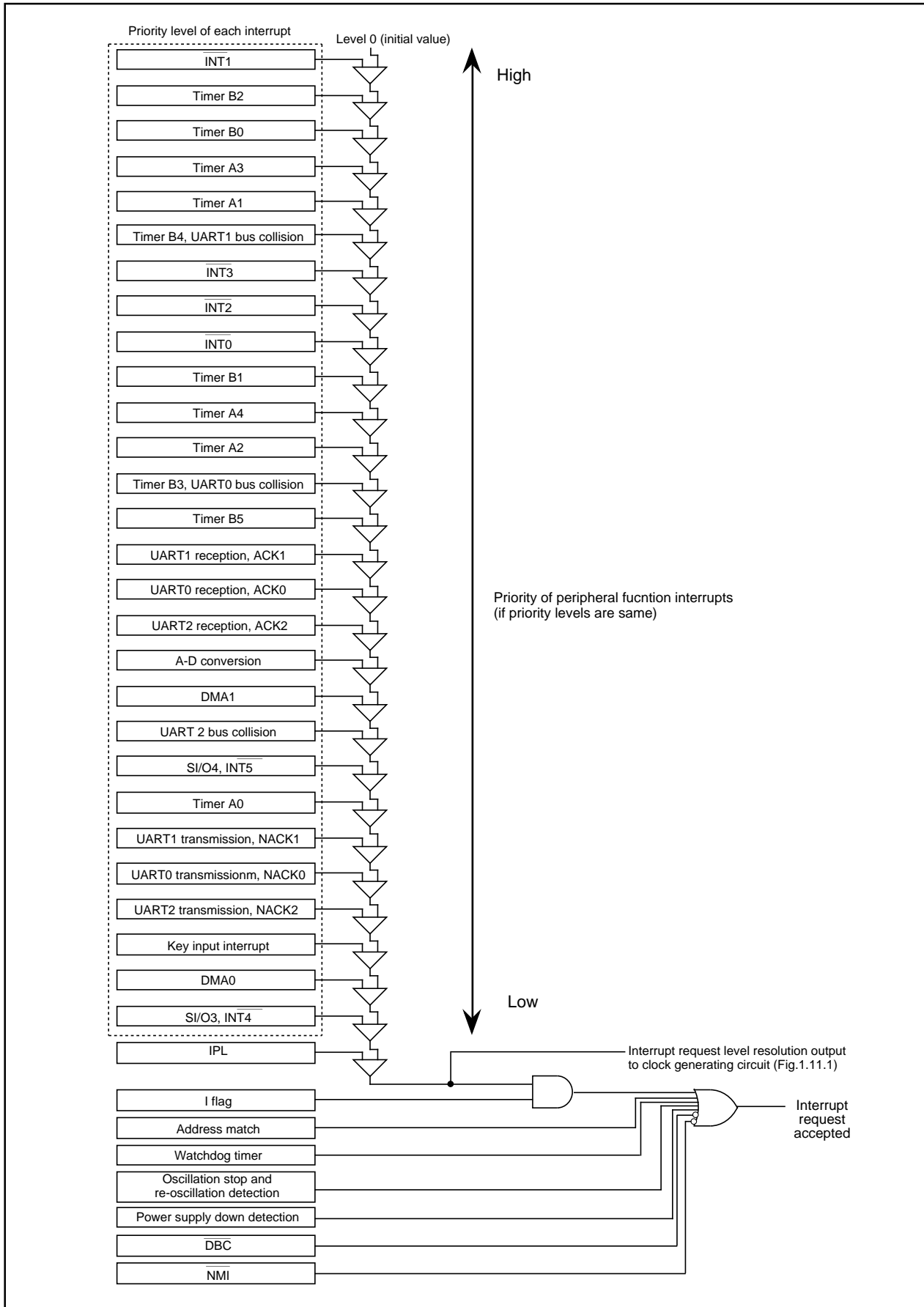
Reset >  $\overline{\text{NMI}}$  >  $\overline{\text{DBC}}$  > WDT > Peripheral function > Single step > Address match

**Figure 1.11.8. Hardware Interrupt Priority**

## Interrupt Priority Resolution Circuit

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

Figure 1.11.9 shows the circuit that judges the interrupt priority level.



**Figure 1.11.9. Interrupts Priority Select Circuit**



Interrupts

**INT Interrupt**

$\overline{INT}_i$  interrupt ( $i=0$  to  $5$ ) is triggered by the edges of external inputs. The edge polarity is selected using the IFSR register's IFSR $i$  bit.

$\overline{INT}_4$  and  $\overline{INT}_5$  share the interrupt vector and interrupt control register with SI/O3 and SI/O4, respectively. To use the  $\overline{INT}_4$  interrupt, set the IFSR register's IFSR6 bit to "1" (=  $\overline{INT}_4$ ). To use the  $\overline{INT}_5$  interrupt, set the IFSR register's IFSR7 bit to "1" (=  $\overline{INT}_5$ ).

After modifying the IFSR6 or IFSR7 bit, clear the corresponding IR bit to "0" (= interrupt not requested) before enabling the interrupt.

Figure 1.11.10 shows the IFSR and IFSR2A registers.

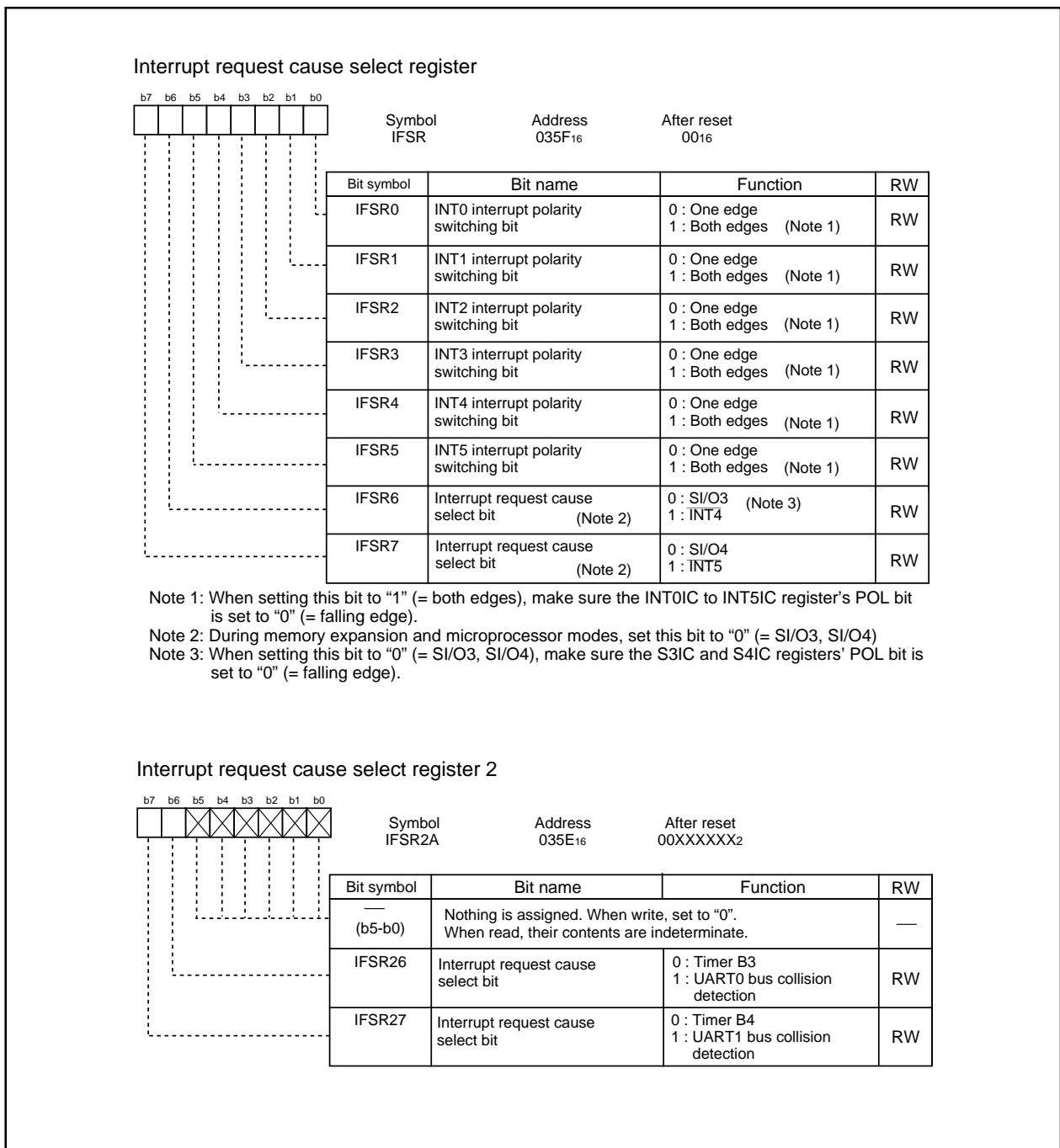


Figure 1.11.10. IFSR Register and IFSR2A Register

## Interrupts

### NMI Interrupt

An  $\overline{\text{NMI}}$  interrupt 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 register's P8\_5 bit.

This pin cannot be used as an input port.

### Key Input Interrupt

Of P104 to P107, a key input interrupt is generated when input on any of the P104 to P107 pins which has had the PD10 register's PD10\_4 to PD10\_7 bits set to "0" (= input) goes low. Key input interrupts can be used as a key-on wakeup 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 P104 to P107 as analog input ports. Figure 1.11.11 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.

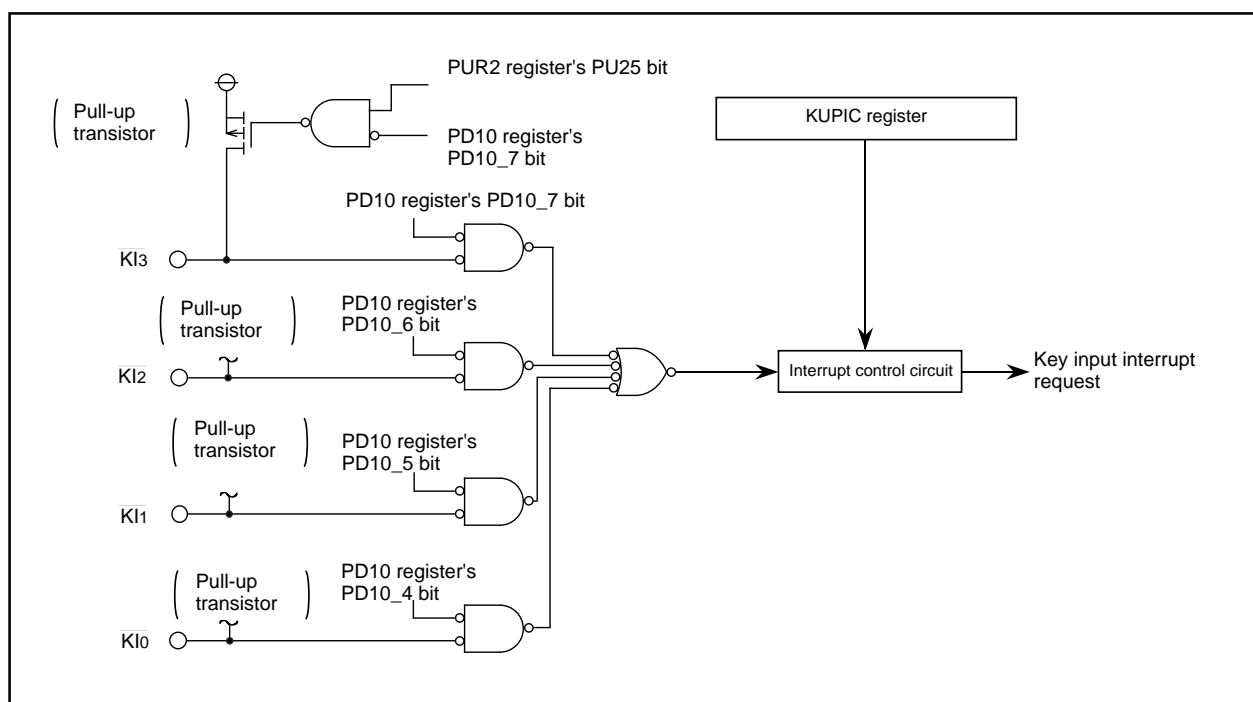


Figure 1.11.11. Key Input Interrupt

## Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMAD<sub>i</sub> register (i=0 to 3). Set the start address of any instruction in the RMAD<sub>i</sub> register. Use the AIER register's AIER0 and AIER1 bits and the AIER2 register's AIER20 and AIER21 bits 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. Figure 1.11.12 shows the instruction just before execution and address stored in the stack when there occurs interruption.

Note that when using the external data bus in width of 8 bits, the address match interrupt cannot be used for external area.

Figure 1.11.13 shows the AIER, AIER2, and RMAD0 to RMAD3 registers.

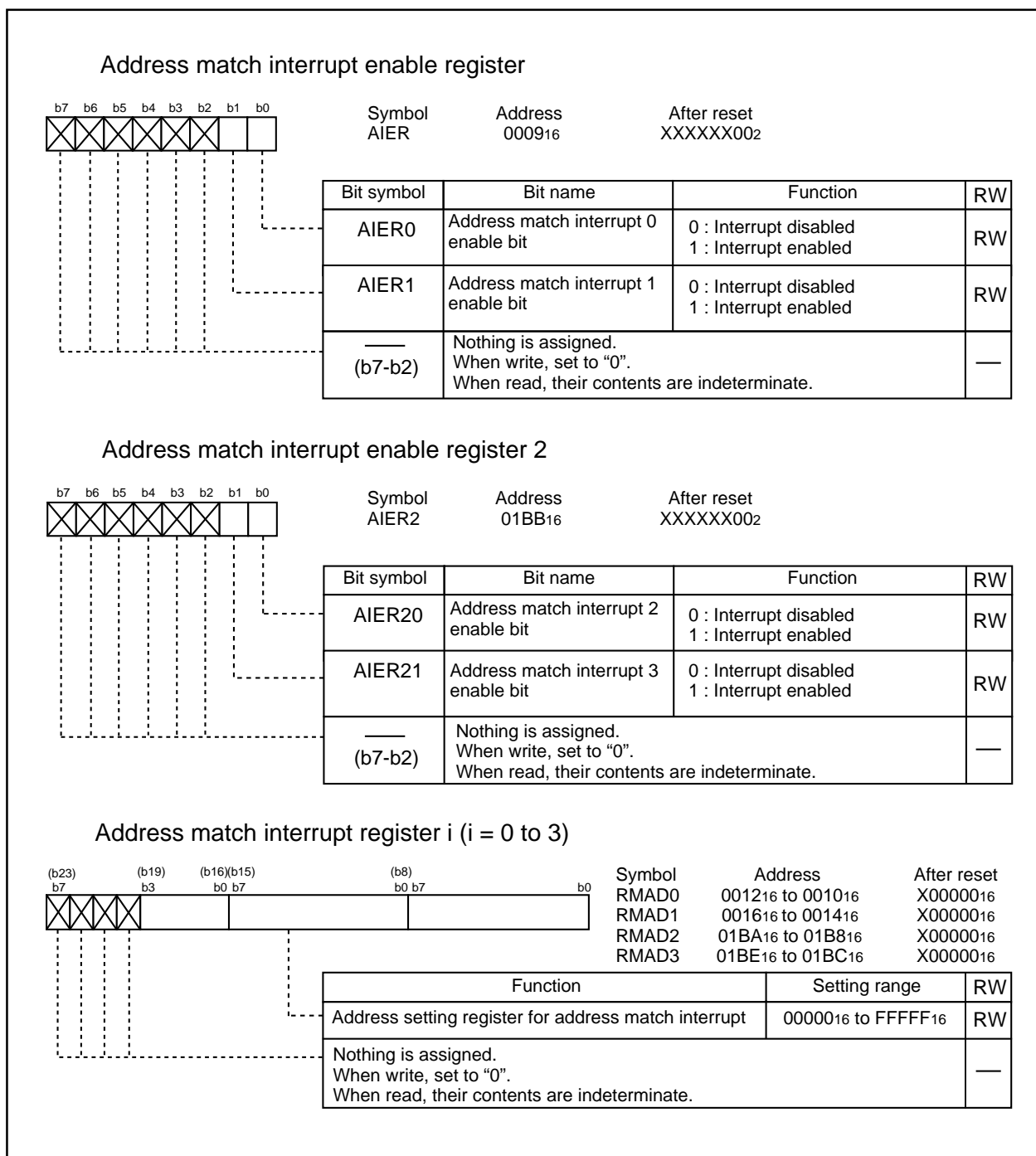
(1) Instructions in which the "return destination + 2" address is stored in the stack when address match interrupt occurs					
• 16-bit operation code					
• Instruction shown below among 8-bit operation code instructions					
ADD.B:S	#IMM8,dest	SUB.B:S	#IMM8,dest	AND.B:S	#IMM8,dest
OR.B:S	#IMM8,dest	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)			
(2) Instructions in which the "return destination + 1" address is stored in the stack when address match interrupt occurs					
• Instructions other than the above					

**Figure 1.11.12. Instruction Just Before Execution and Address Stored in Stack When There Occurs Interrupts**

**Table 1.11.6. Relationship Between Address Match Interrupt Sources and Associated Registers**

Address match interrupt sources	Address match interrupt enable bit	Address match interrupt register
Address match interrupt 0	AIER0	RMAD0
Address match interrupt 1	AIER1	RMAD1
Address match interrupt 2	AIER20	RMAD2
Address match interrupt 3	AIER21	RMAD3

**Interrupts**



**Figure 1.11.13. AIER Register, AIER2 Register and RMAD0 to RMAD3 Registers**

## Precautions for Interrupts

### (1) Reading Address 00000<sub>16</sub>

- Do not read the address 00000<sub>16</sub> in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000<sub>16</sub> during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to "0". If the address 00000<sub>16</sub> is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to "0". This causes a problem that the interrupt is canceled, or an unexpected interrupt is generated.

### (2) SP Setting

- Set any value in the SP before accepting an interrupt. The SP is cleared to '0000<sub>16</sub>' after reset. Therefore, if an interrupt is accepted before setting any value in the SP, the program may go out of control. Especially when using  $\overline{\text{NMI}}$  interrupt, set a value in the SP 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.

### (3) $\overline{\text{NMI}}$ Interrupt

- The  $\overline{\text{NMI}}$  interrupt cannot be disabled. If this interrupt is unused, connect the  $\overline{\text{NMI}}$  pin to VCC via a resistor (pull-up).
- The input level of the  $\overline{\text{NMI}}$  pin can be read by accessing the P8 register's P8\_5 bit. Note that the P8\_5 bit can only be read when determining the pin level after an  $\overline{\text{NMI}}$  interrupt is generated.
- 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 CM1 register's CM10 bit 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.

### (4) $\overline{\text{INT}}$ Interrupt

- Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to the  $\overline{\text{INT}}$  through  $\overline{\text{INT}}_5$  pins regardless of the CPU clock.
- When the polarity of the  $\overline{\text{INT}}_0$  to  $\overline{\text{INT}}_5$  pins is changed, the IR bit is sometimes set to "1" (=interrupt requested). After changing the polarity, set the IR bit to "0" (=interrupt not requested). Figure 1.11.13 shows the procedure for changing the  $\overline{\text{INT}}$  interrupt generate factor.

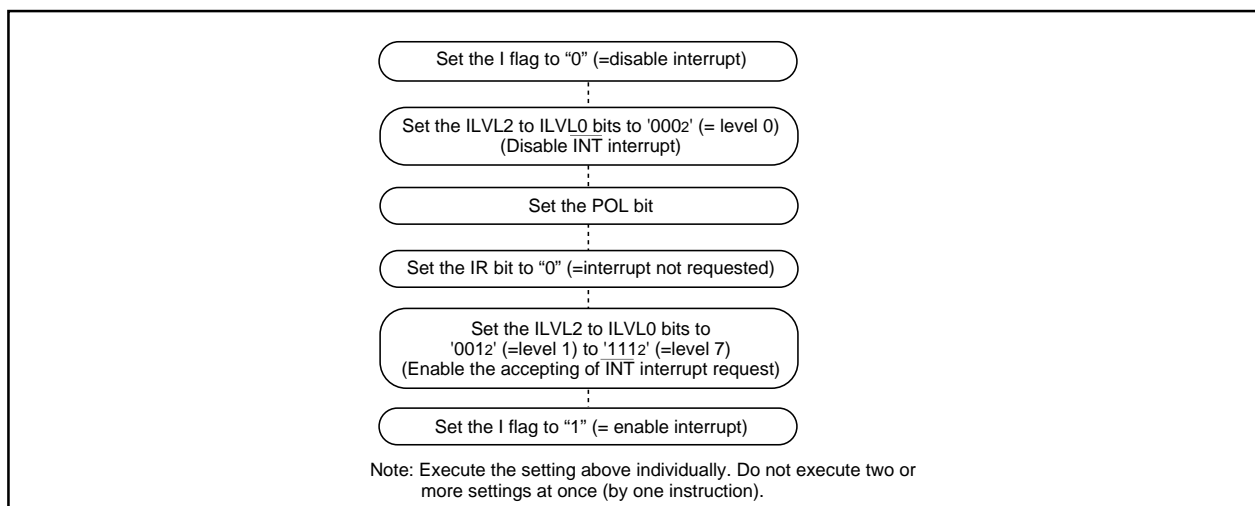


Figure 1.11.14. Switching Procedure for  $\overline{\text{INT}}$  Interrupt Request

## (5) Watchdog Timer Interrupt

- Initialize the watchdog timer after the watchdog timer interrupt occurs.

## (6) Modifying Interrupt Control Register

- Each interrupt control register can only be modified while no interrupt requests corresponding to that register are generated. If interrupt requests managed by any interrupt control register are likely to occur, disable the interrupts before modifying the register. A sample program is shown below.

### Example 1:

```
INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Set the TA0IC register to "0016".
  NOP                               ; Four NOP instructions are required when using HOLD function.
  NOP
  FSET  I           ; Enable interrupts.
```

### Example 2:

```
INT_SWITCH2:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Set the TA0IC register to "0016".
  MOV.W MEM, R0    ; Dummy read.
  FSET  I           ; Enable interrupts.
```

### Example 3:

```
INT_SWITCH3:
  PUSHC FLG       ; Push Flag register onto stack
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Set the TA0IC register to "0016".
  POPC  FLG       ; Enable interrupts.
```

Why the FSET I instruction is preceded by two NOP instructions (four when using HOLD function) in Example 1 and why the FSET I instruction is preceded by a dummy read in Example 2

This is to prevent the I flag from being set to "1" before writing to the interrupt control register for reasons of the instruction queue buffer.

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

### Modifying other than the IR bit

If an interrupt request corresponding to that register is generated while executing the instruction, the IR bit may not be set to "1" (= interrupt requested), with the result that the interrupt request is ignored. If this presents a problem, use the following instructions to modify the register.

Instructions to use: AND, OR, BCLR, BSET

### Modifying the IR bit

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

## Watchdog Timer

The watchdog timer is the function of detecting when the program is out of control. Therefore, we recommend using the watchdog timer to improve reliability of a system. The watchdog timer contains a 15-bit counter which counts down the clock derived by dividing the CPU clock using the prescaler. Whether to generate a watchdog timer interrupt request or apply a watchdog timer reset as an operation to be performed when the watchdog timer underflows after reaching the terminal count can be selected using the PM12 bit of PM1 register. The PM12 bit can only be set to "1" (reset). Once this bit is set to "1", it cannot be set to "0" (watchdog timer interrupt) in a program.

The pin, CPU and SFR initialized where the monitor timer underflows when the PM12 bit is "1" are the same as in software reset.

When the main clock 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

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

With sub-clock chosen for CPU clock

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

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

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 1.12.1 shows the block diagram of the watchdog timer. Figure 1.12.2 shows the watchdog timer-related registers.

- Count source protective mode

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

Before this mode can be used, the following register settings are required:

- (1) Set the PRC1 bit of PRCR register to "1" (enable writes to PM1 and PM2 registers).
- (2) Set the PM12 bit of PM1 register to "1" (reset when the watchdog timer underflows).
- (3) Set the PM22 bit of PM2 register to "1" (ring oscillator clock used for the watchdog timer count source).
- (4) Set the PRC1 bit of PRCR register to "0" (disable writes to PM1 and PM2 registers).
- (5) Write to the WDTS register (watchdog timer starts counting).

Setting the PM22 bit to "1" results in the following conditions

- The ring oscillator starts oscillating, and the ring oscillator clock becomes the watchdog timer count source.
- The CM10 bit of CM1 register is disabled against write. (Writing a "1" has no effect, nor is stop mode entered.)
- The watchdog timer does not stop when in wait mode or hold state.

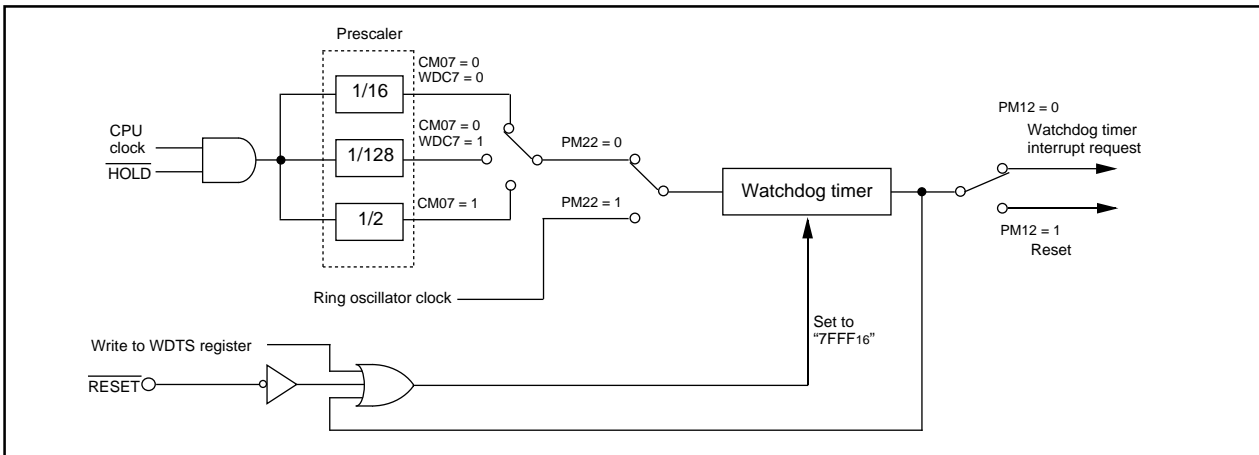


Figure 1.12.1. Watchdog Timer Block Diagram

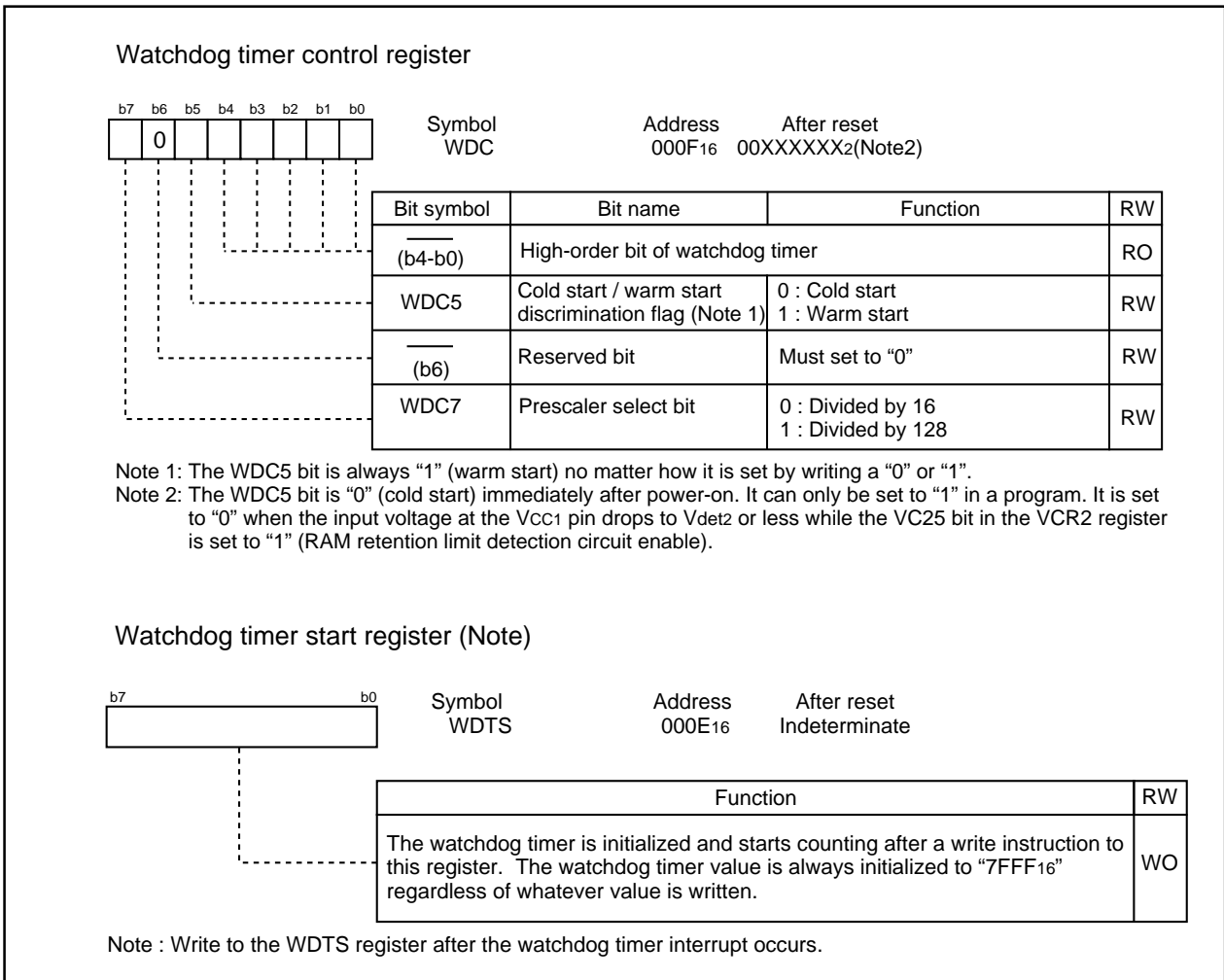


Figure 1.12.2. WDC Register and WDTS Register



## 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 1.13.1 shows the block diagram of the DMAC. Table 1.13.1 shows the DMAC specifications. Figures 1.13.2 to 1.13.4 show the DMAC-related registers.

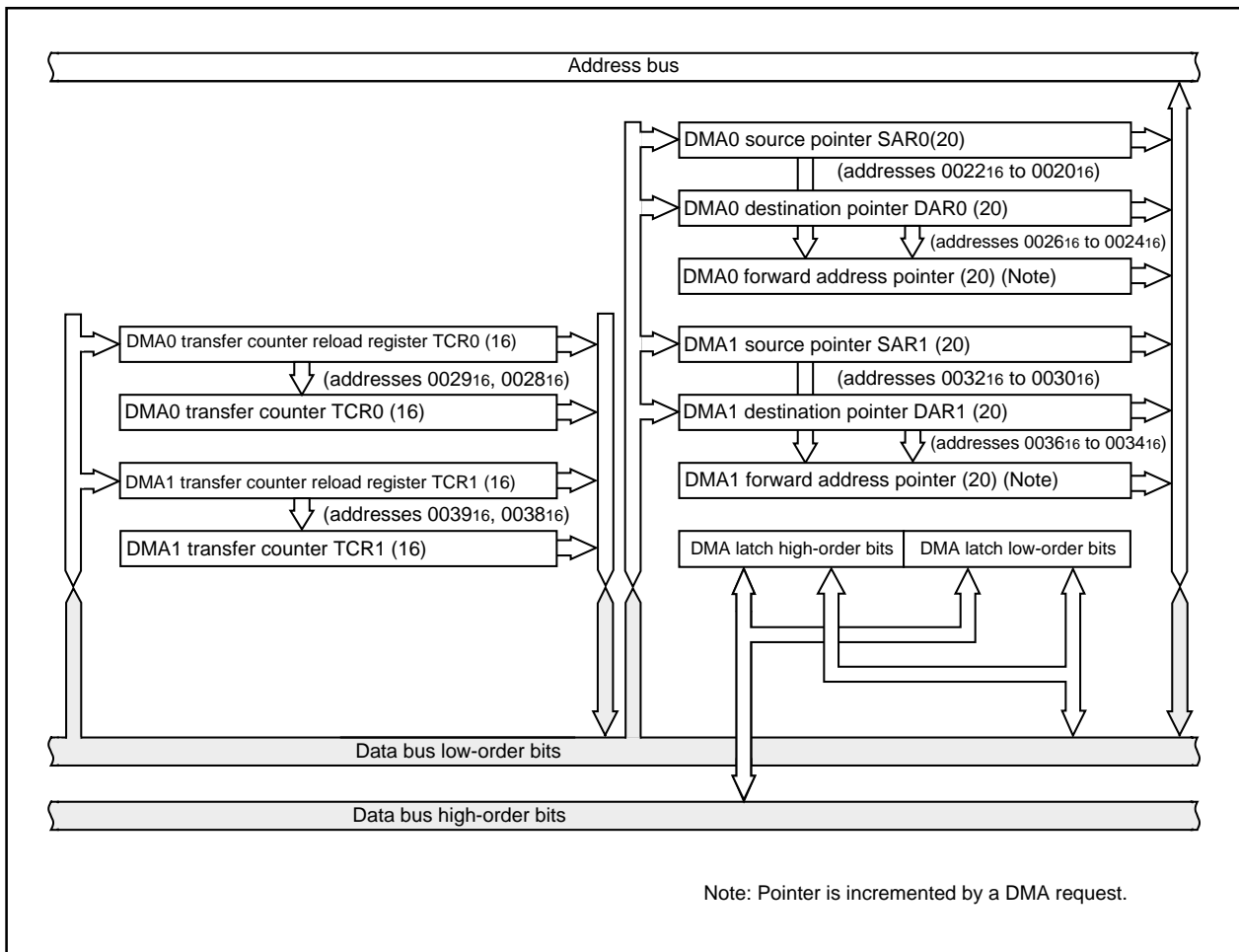


Figure 1.13.1. DMAC Block Diagram

A DMA request is generated by a write to the DMiSL register ( $i = 0-1$ )'s DSR bit, as well as by an interrupt request which is generated by any function specified by the DMiSL register's DMS and DSEL3–DSEL0 bits. 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 interrupt control register's IR bit does not change state due to a DMA transfer.

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

Table 1.13.1. DMAC Specifications

Item	Specification	
No. of channels	2 (cycle steal method)	
Transfer memory space	<ul style="list-style-type: none"> <li>• From any address in the 1M bytes space to a fixed address</li> <li>• From a fixed address to any address in the 1M bytes space</li> <li>• From a fixed address to a fixed address</li> </ul>	
Maximum No. of bytes transferred	128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)	
DMA request factors (Note 1, Note 2)	Falling edge of $\overline{INT0}$ or $\overline{INT1}$ Both edge of $\overline{INT0}$ or $\overline{INT1}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 transfer, UART0 reception interrupt requests UART1 transfer, UART1 reception interrupt requests UART2 transfer, UART2 reception interrupt requests SI/O3, SI/O4 interrupt requests A-D conversion interrupt requests Software triggers	
Channel priority	DMA0 > DMA1 (DMA0 takes precedence)	
Transfer unit	8 bits or 16 bits	
Transfer address direction	forward or fixed (The source and destination addresses cannot both be in the forward direction.)	
Transfer mode	•Single transfer	Transfer is completed when the DMA <sub>i</sub> transfer counter ( $i = 0-1$ ) underflows after reaching the terminal count.
	•Repeat transfer	When the DMA <sub>i</sub> transfer counter underflows, it is reloaded with the value of the DMA <sub>i</sub> transfer counter reload register and a DMA transfer is continued with it.
DMA interrupt request generation timing	When the DMA <sub>i</sub> transfer counter underflowed	
DMA startup	Data transfer is initiated each time a DMA request is generated when the DMA <sub>i</sub> CON register's DMAE bit = "1" (enabled).	
DMA shutdown	•Single transfer	<ul style="list-style-type: none"> <li>• When the DMAE bit is set to "0" (disabled)</li> <li>• After the DMA<sub>i</sub> transfer counter underflows</li> </ul>
	•Repeat transfer	When the DMAE bit is set to "0" (disabled)
Reload timing for forward address pointer and transfer counter	When a data transfer is started after setting the DMAE bit to "1" (enabled), the forward address pointer is reloaded with the value of the SAR <sub>i</sub> or the DAR <sub>i</sub> pointer whichever is specified to be in the forward direction and the DMA <sub>i</sub> transfer counter is reloaded with the value of the DMA <sub>i</sub> transfer counter reload register.	

## Notes:

1. DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the I flag nor by the interrupt control register.
2. The selectable causes of DMA requests differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0020<sub>16</sub>–003F<sub>16</sub>) are accessed by the DMAC.

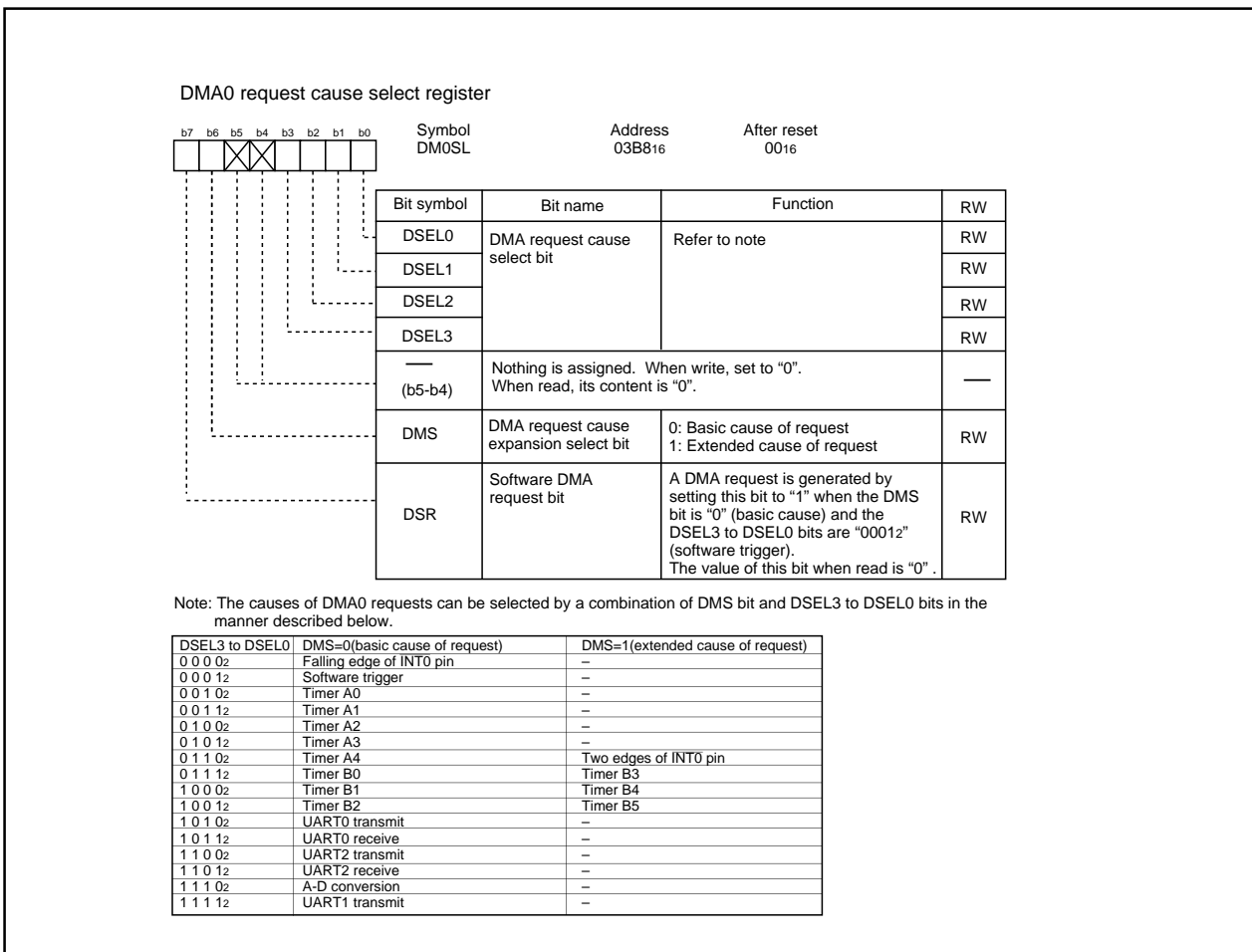


Figure 1.13.2. DM0SL Register

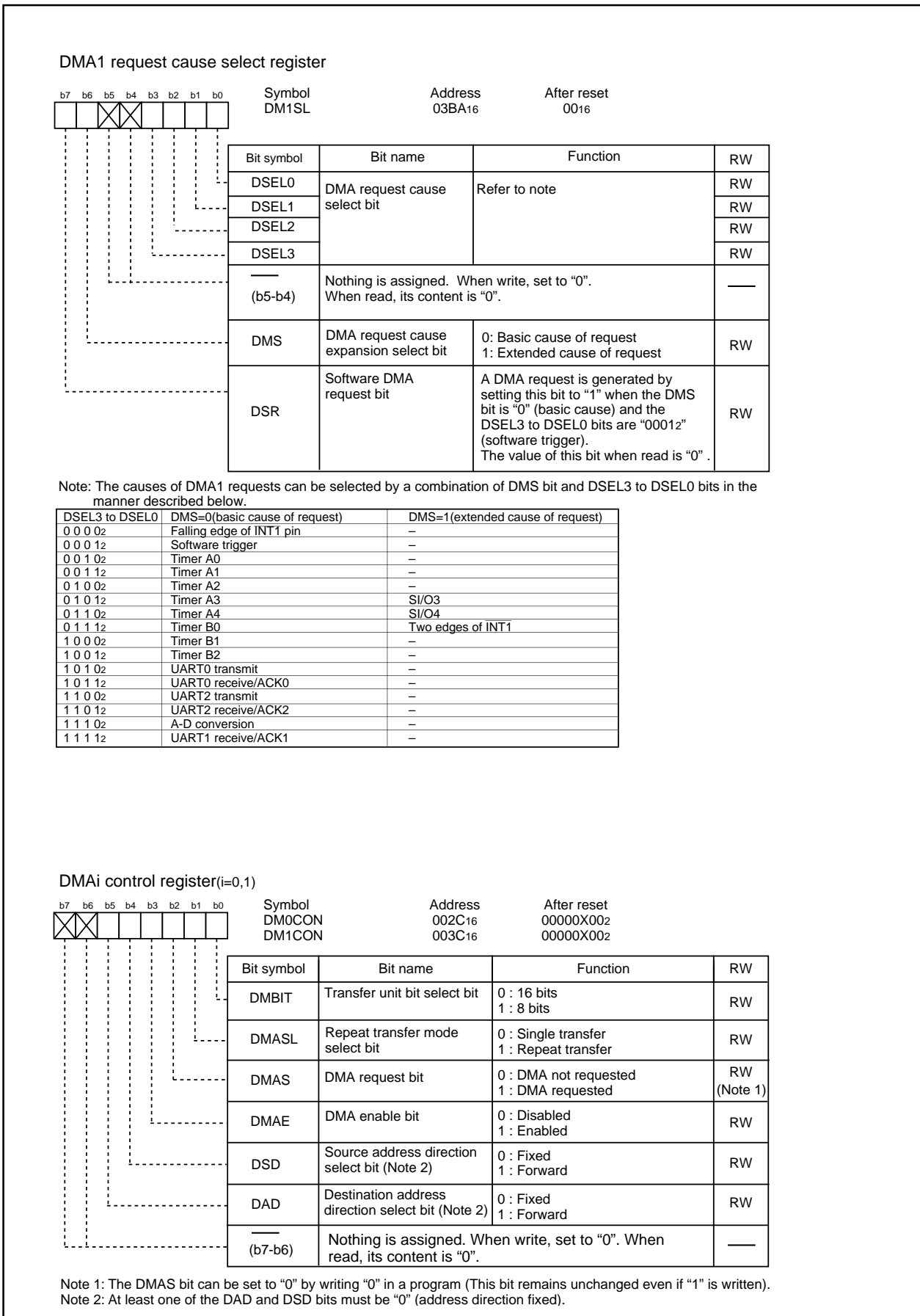


Figure 1.13.3. DM1SL Register, DM0CON Register, and DM1CON Registers

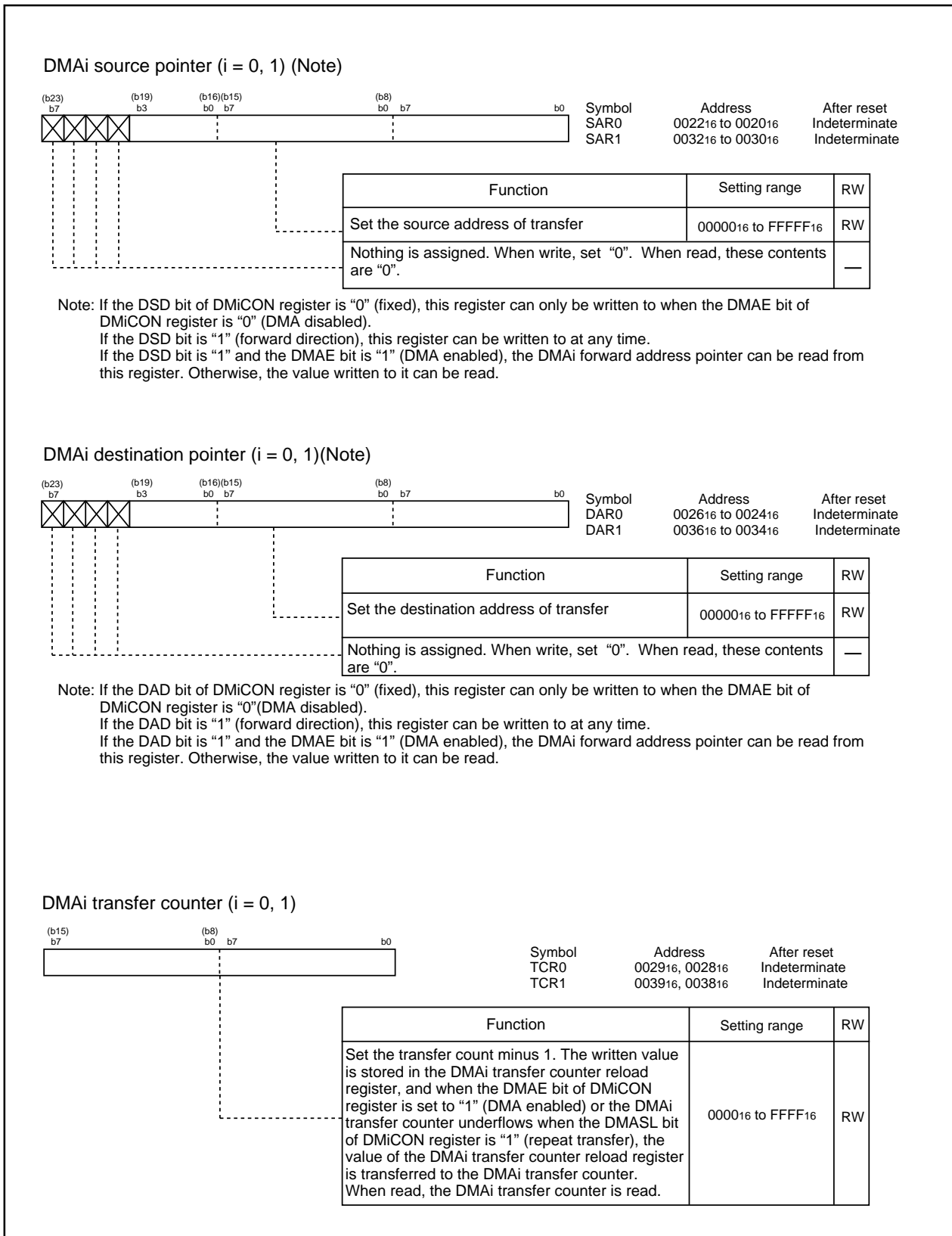


Figure 1.13.4. SAR0, SAR1, DAR0, DAR1, TCR0, and TCR1 Registers

## 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.

### (a) 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.

### (b) 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.

### (c) 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.

### (d) Effect of $\overline{\text{RDY}}$ Signal

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

Figure 1.13.5 shows the example of the cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16 bit units using an 8-bit bus ((2) in Figure 1.13.5), two source read bus cycles and two destination write bus cycles are required.

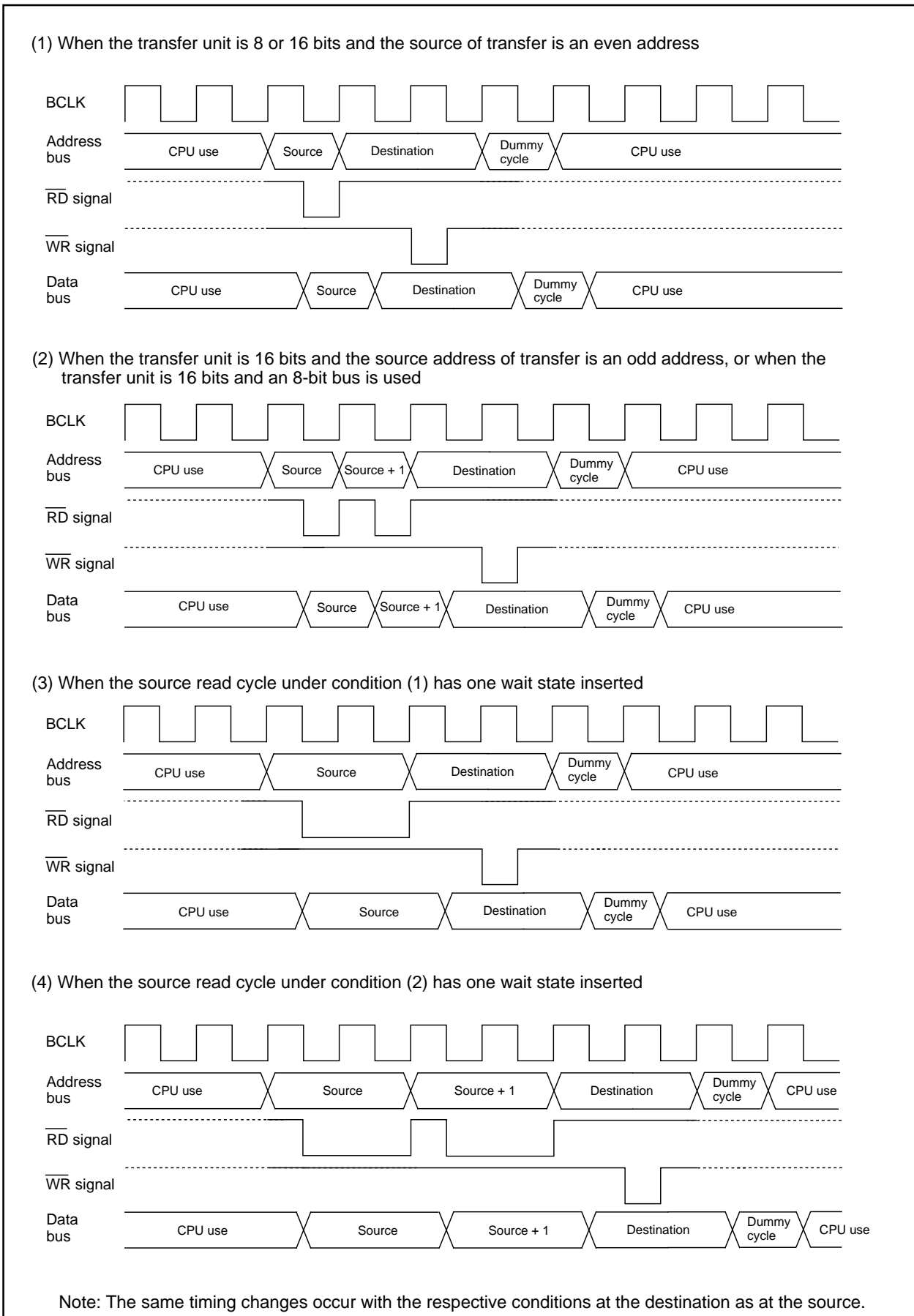


Figure 1.13.5. Transfer Cycles for Source Read

## 2. DMA Transfer Cycles

Any combination of even or odd transfer read and write addresses is possible. Table 1.13.2 shows the number of DMA transfer cycles. Table 1.13.3 shows the Coefficient j, k.

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

$$\text{No. of transfer cycles per transfer unit} = \text{No. of read cycles} \times j + \text{No. of write cycles} \times k$$

**Table 1.13.2. 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

**Table 1.13.3. Coefficient j, k**

	Internal area				External area						
	Internal ROM, RAM		SFR		Separate bus			Multiplex bus			
	No wait	With wait	1-wait <sup>2</sup>	2-wait <sup>2</sup>	No wait	With wait <sup>1</sup>			With wait <sup>1</sup>		
						1 wait	2 waits	3 waits	1wait	2 waits	3 waits
j	1	2	2	3	1	2	3	4	3	3	4
k	1	2	2	3	2	2	3	4	3	3	4

Notes:

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



### 3. DMA Enable

When a data transfer starts after setting the DMAE bit in 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 DMiCON register is "1" (forward) or the DARi register value when the DAD bit of 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 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.

### 4. DMA Request

The DMAC can generate a DMA request as triggered by the cause of request that is selected with the DMS and DSEL3 to DSEL0 bits of DMiSL register (i = 0, 1) on either channel. Table 1.13.4 shows the timing at which the DMAS bit changes state.

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

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

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

**Table 1.13.4. 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 of DMiCON register is set to "1"	<ul style="list-style-type: none"> <li>• Immediately before a data transfer starts</li> <li>• When set by writing "0" in a program</li> </ul>
Peripheral function	When the interrupt control register for the peripheral function that is selected by the DSEL3 to DSEL0 and DMS bits of DMiCON register has its IR bit set to "1"	

## 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 1.13.6 shows an example of DMA transfer effected by external factors.

In Figure 1.13.6, because DMA0 and DMA1 requests occurred at the same time, DMA0 which has higher channel priority is accepted first and a DMA transfer on it starts. When DMA0 finishes one transfer unit, it relinquishes control of the bus to the CPU, and when the CPU finishes one bus access, DMA1 starts a transfer next and after completion of one transfer unit, returns control of the bus to the CPU.

Note that because there is only one DMAS bit on each channel, the number of times DMA is requested cannot be counted. Therefore, even if multiple DMA requests occurred before gaining control of the bus as in the case of DMA1 in Figure 1.13.6, the DMAS bit is set to "0" when control of the bus is gained and after completion of one transfer unit, control of the bus is returned to the CPU.

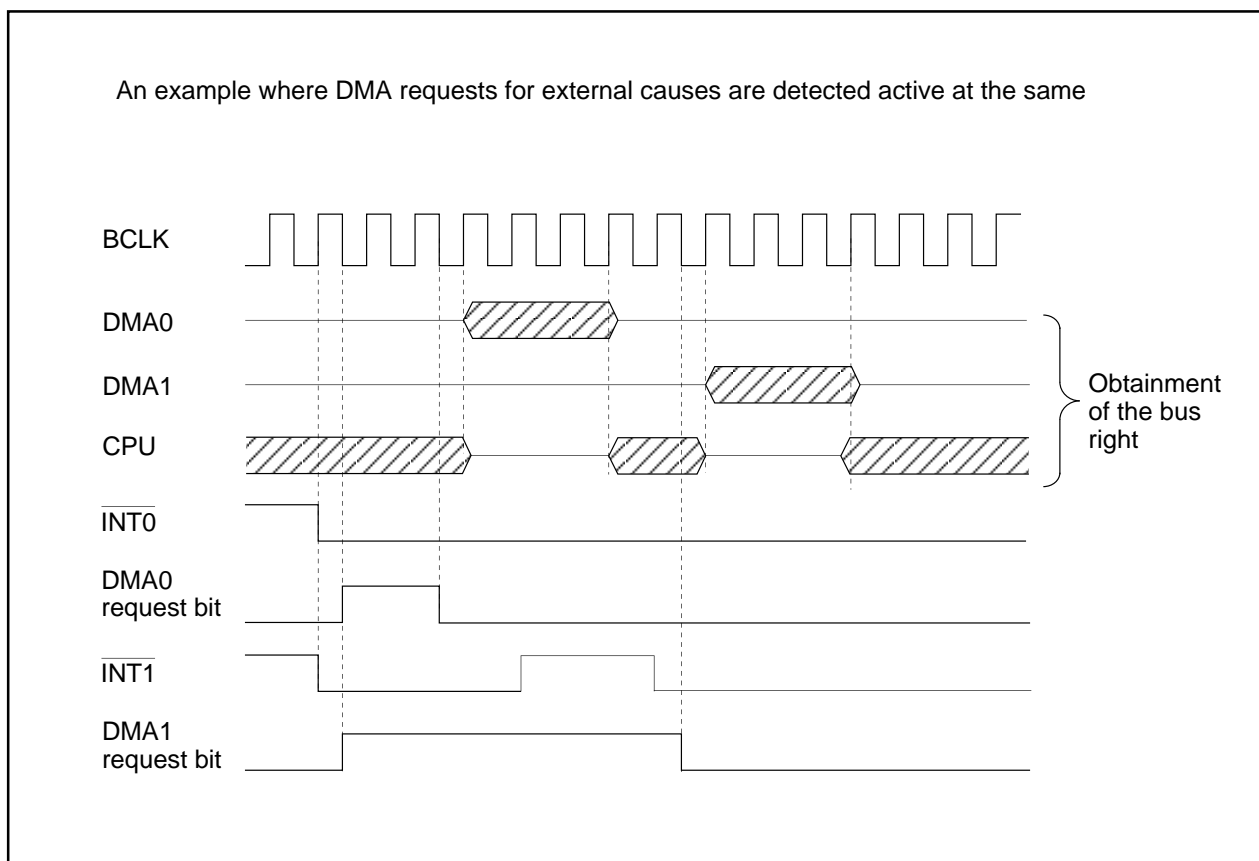


Figure 1.13.6. DMA Transfer by External Factors

# Timers

Eleven 16-bit timers, each capable of operating independently of the others, can be classified by function as either timer A (five) and timer B (six). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc. Figures 1.14.1 and 1.14.2 show block diagrams of timer A and timer B configuration, respectively.

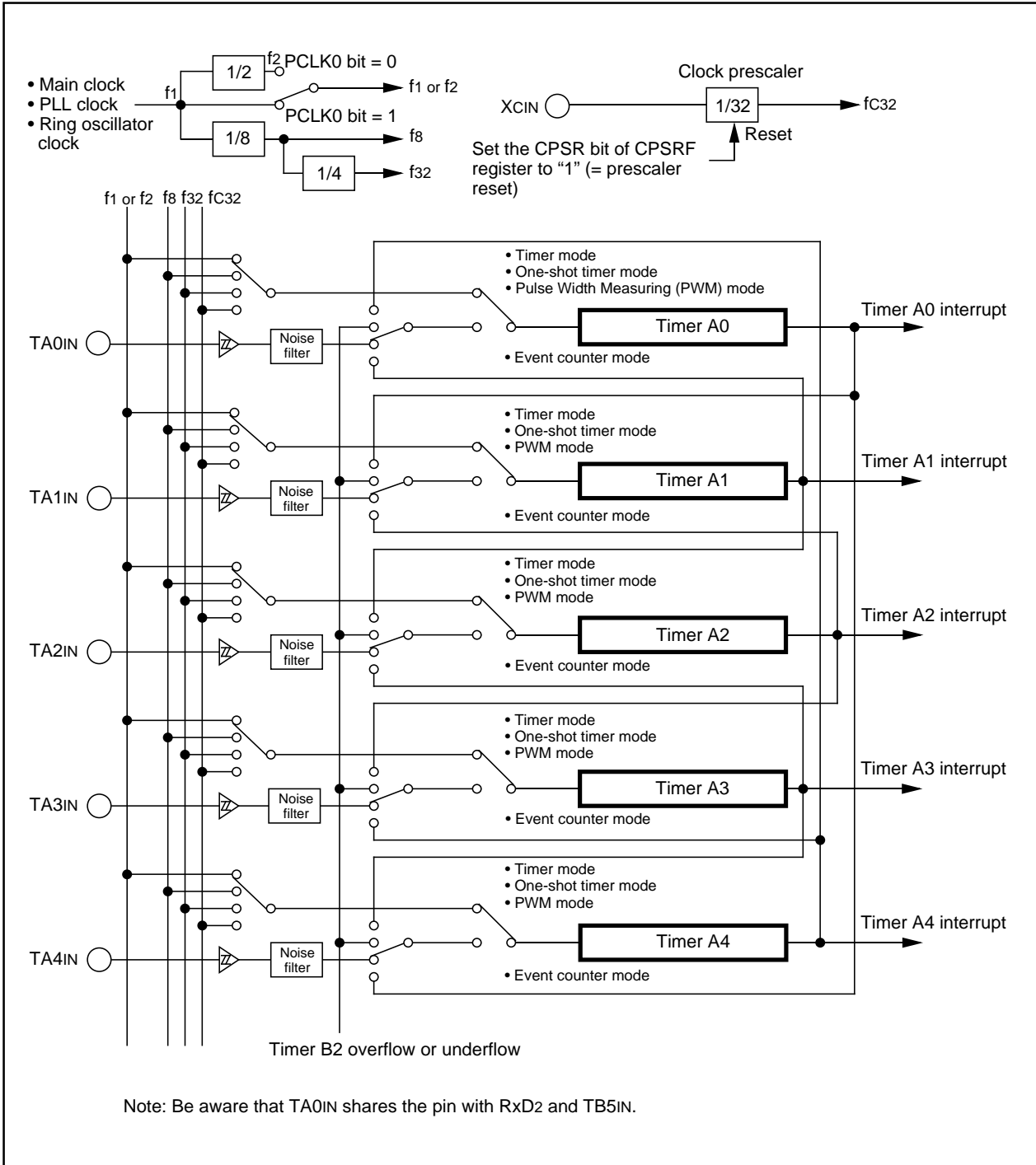


Figure 1.14.1. Timer A Configuration

Timers

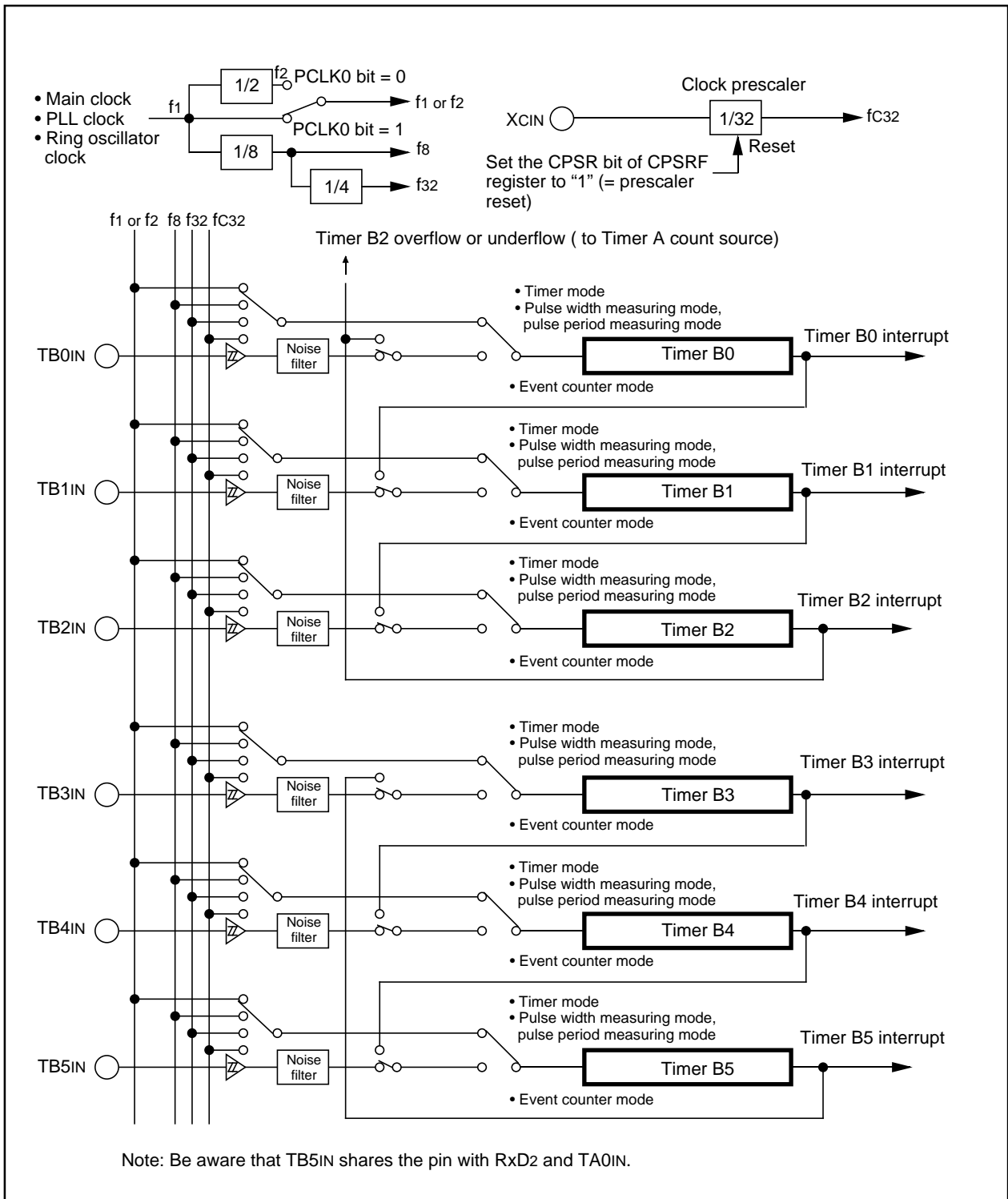


Figure 1.14.2. Timer B Configuration

### Timer A

Figure 1.14.3 shows a block diagram of the timer A. Figures 1.14.4 to 1.14.6 show registers related to the timer A.

The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use the TMOD1 to TMOD0 bits of TAiMR register (i = 0 to 4) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows and underflows of other timers.
- One-shot timer mode: The timer outputs a pulse only once before it reaches the minimum count "0000<sub>16</sub>."
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.

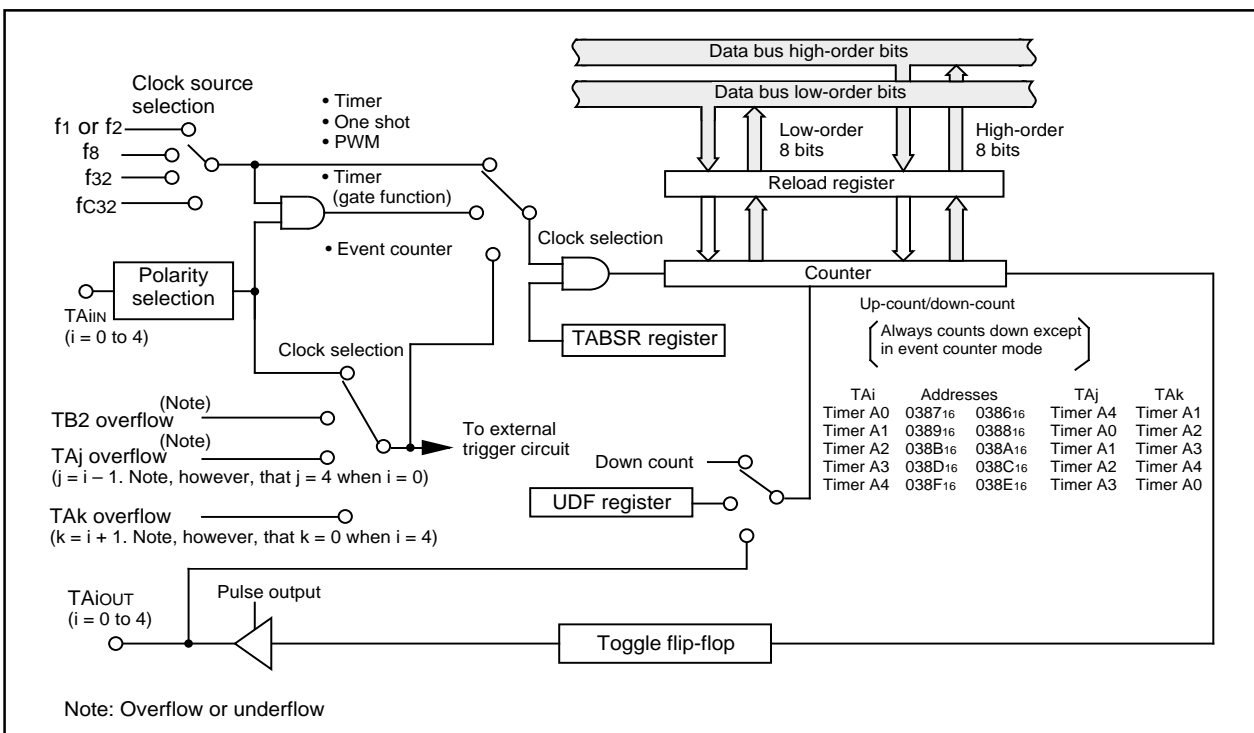


Figure 1.14.3. Timer A Block Diagram

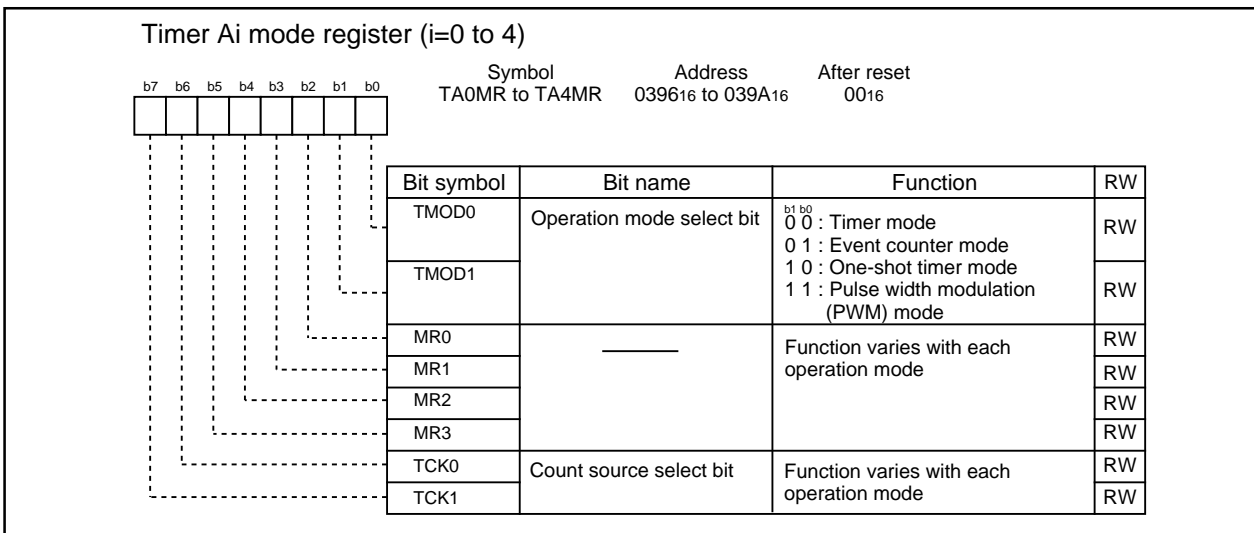


Figure 1.14.4. TA0MR to TA4MR Registers

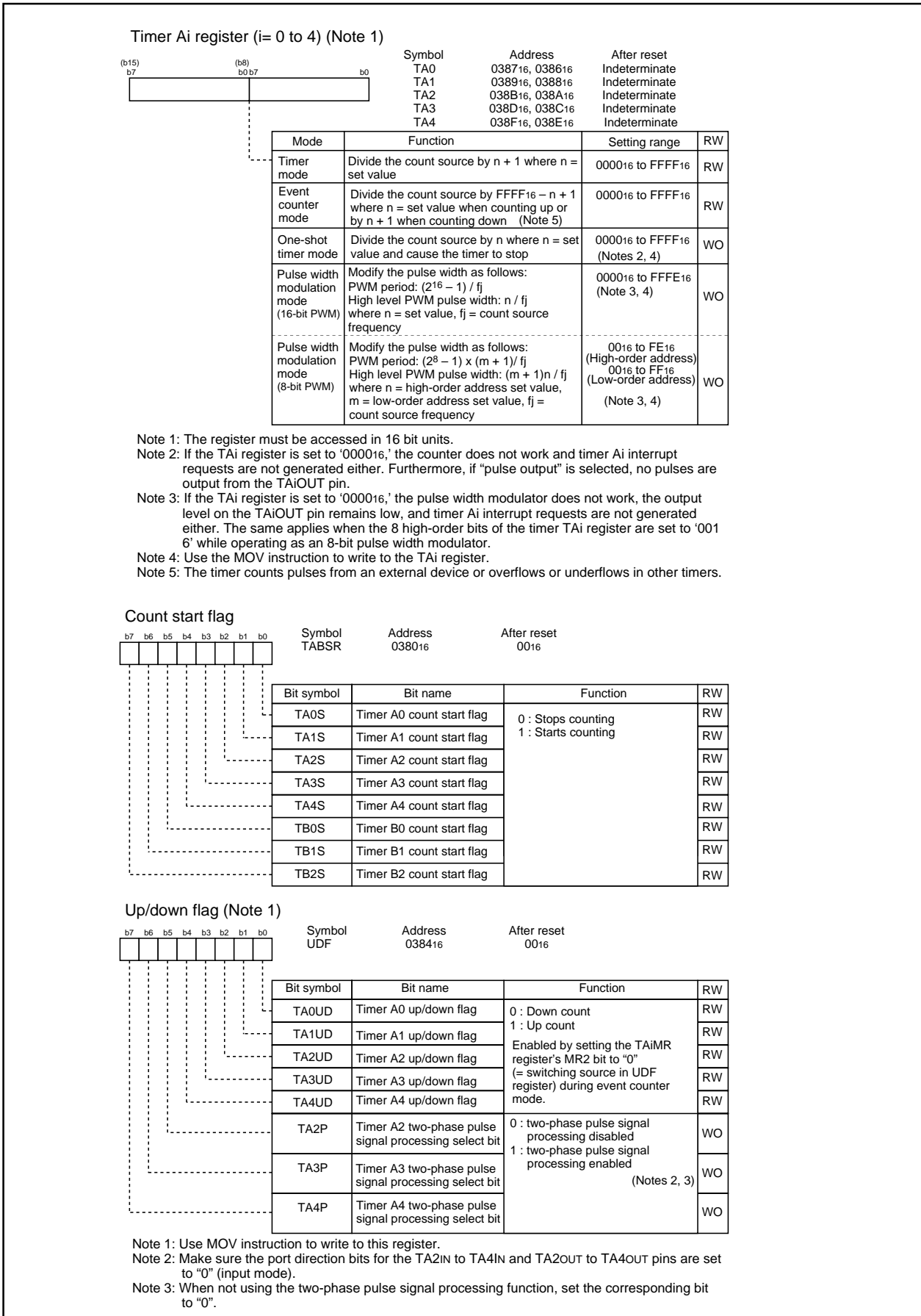


Figure 1.14.5. TA0 to TA4 Registers, TABSR Register, and UDF Register

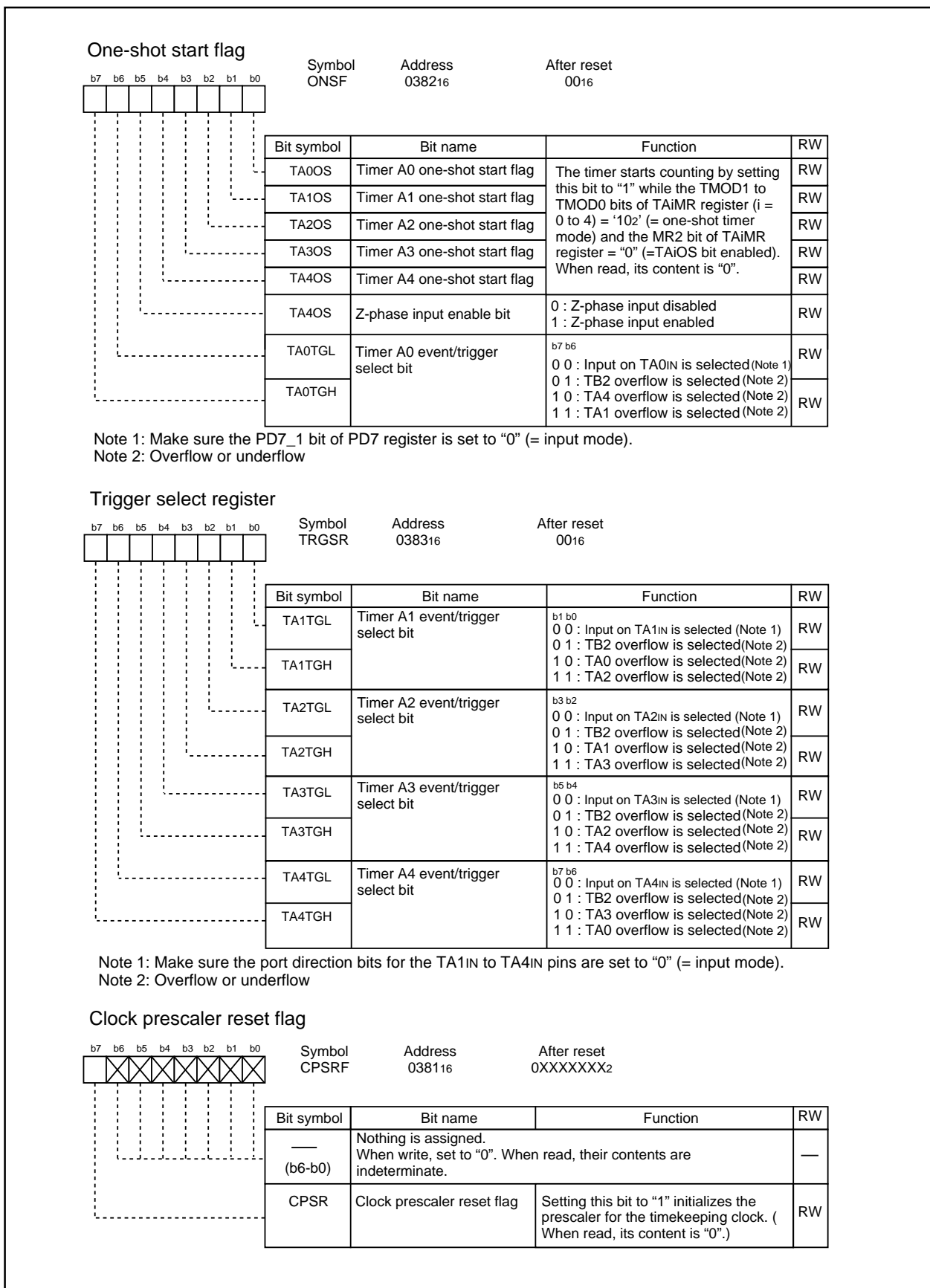


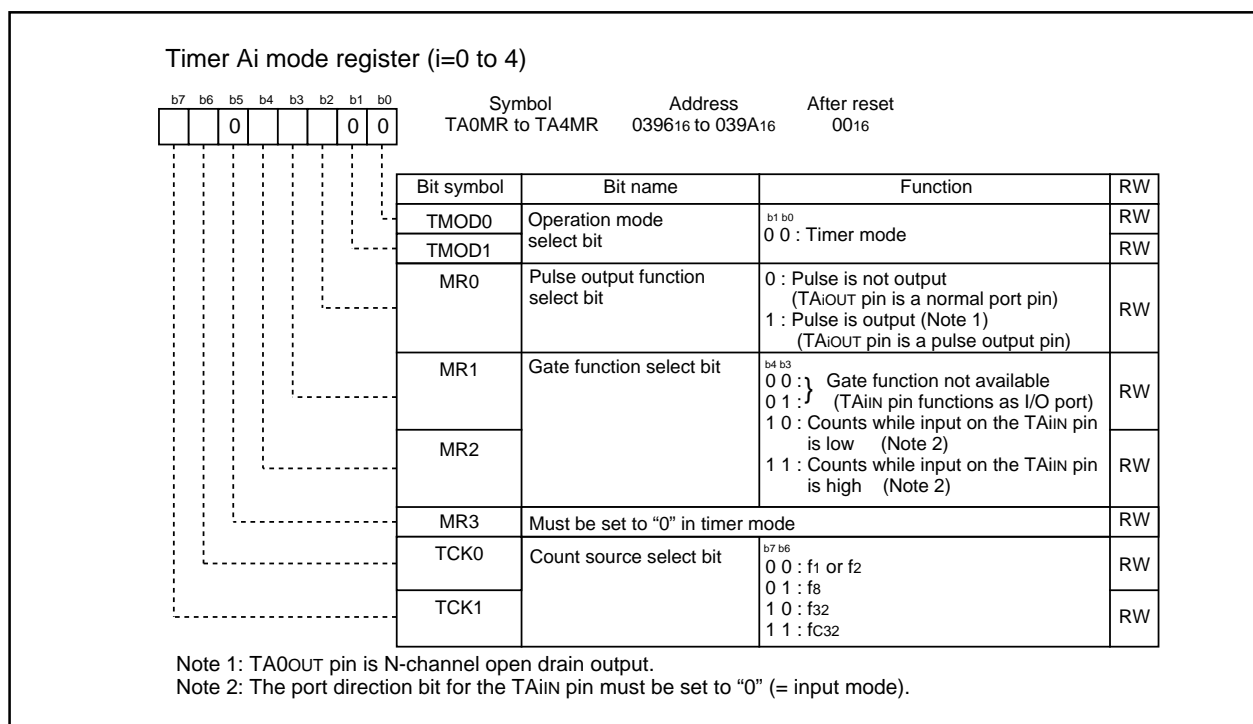
Figure 1.14.6. ONSF Register, TRGSR Register, and CPSRF Register

## 1. Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 1.14.1). Figure 1.14.7 shows TAI<sub>i</sub>MR register in timer mode.

**Table 1.14.1. Specifications in Timer Mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>2</sub> , f <sub>8</sub> , f <sub>32</sub> , f <sub>C32</sub>
Count operation	<ul style="list-style-type: none"> <li>Down-count</li> <li>When the timer underflows, it reloads the reload register contents and continues counting</li> </ul>
Divide ratio	1/(n+1) n: set value of TAI <sub>i</sub> MR register (i= 0 to 4) 0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	Set TAI <sub>S</sub> bit of TABSR register to "1" (= start counting)
Count stop condition	Set TAI <sub>S</sub> bit to "0" (= stop counting)
Interrupt request generation timing	Timer underflow
TAI <sub>i</sub> N pin function	I/O port or gate input
TAI <sub>i</sub> OUT pin function	I/O port or pulse output
Read from timer	Count value can be read by reading TAI register
Write to timer	<ul style="list-style-type: none"> <li>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</li> <li>When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Gate function Counting can be started and stopped by an input signal to TAI<sub>i</sub>N pin</li> <li>Pulse output function Whenever the timer underflows, the output polarity of TAI<sub>i</sub>OUT pin is inverted. When not counting, the pin outputs a low.</li> </ul>



**Figure 1.14.7. Timer Ai Mode Register in Timer Mode**



## 2. Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3 and A4 can count two-phase external signals. Table 1.14.2 lists specifications in event counter mode (when not processing two-phase pulse signal). Table 1.14.3 lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4). Figure 1.14.8 shows TAI<sub>i</sub>MR register in event counter mode (when not processing two-phase pulse signal). Figure 1.14.9 shows TA2MR to TA4MR registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

**Table 1.14.2. Specifications in Event Counter Mode (when not processing two-phase pulse signal)**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signals input to TAI<sub>i</sub>N pin (i=0 to 4) (effective edge can be selected in program)</li> <li>Timer B2 overflows or underflows,</li> <li>timer A<sub>j</sub> (j=i-1, except j=4 if i=0) overflows or underflows,</li> <li>timer A<sub>k</sub> (k=i+1, except k=0 if i=4) overflows or underflows</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Up-count or down-count can be selected by external signal or program</li> <li>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.</li> </ul>
Divided ratio	1/ (FFFF <sub>16</sub> - n + 1) for up-count 1/ (n + 1) for down-count    n : set value of TAI register    0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	Set TAI <sub>i</sub> S bit of TABSR register to "1" (= start counting)
Count stop condition	Set TAI <sub>i</sub> S bit to "0" (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAI <sub>i</sub> N pin function	I/O port or count source input
TAI <sub>i</sub> OUT 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	<ul style="list-style-type: none"> <li>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</li> <li>When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it</li> <li>Pulse output function Whenever the timer underflows or underflows, the output polarity of TAI<sub>i</sub>OUT pin is inverted . When not counting, the pin outputs a low.</li> </ul>

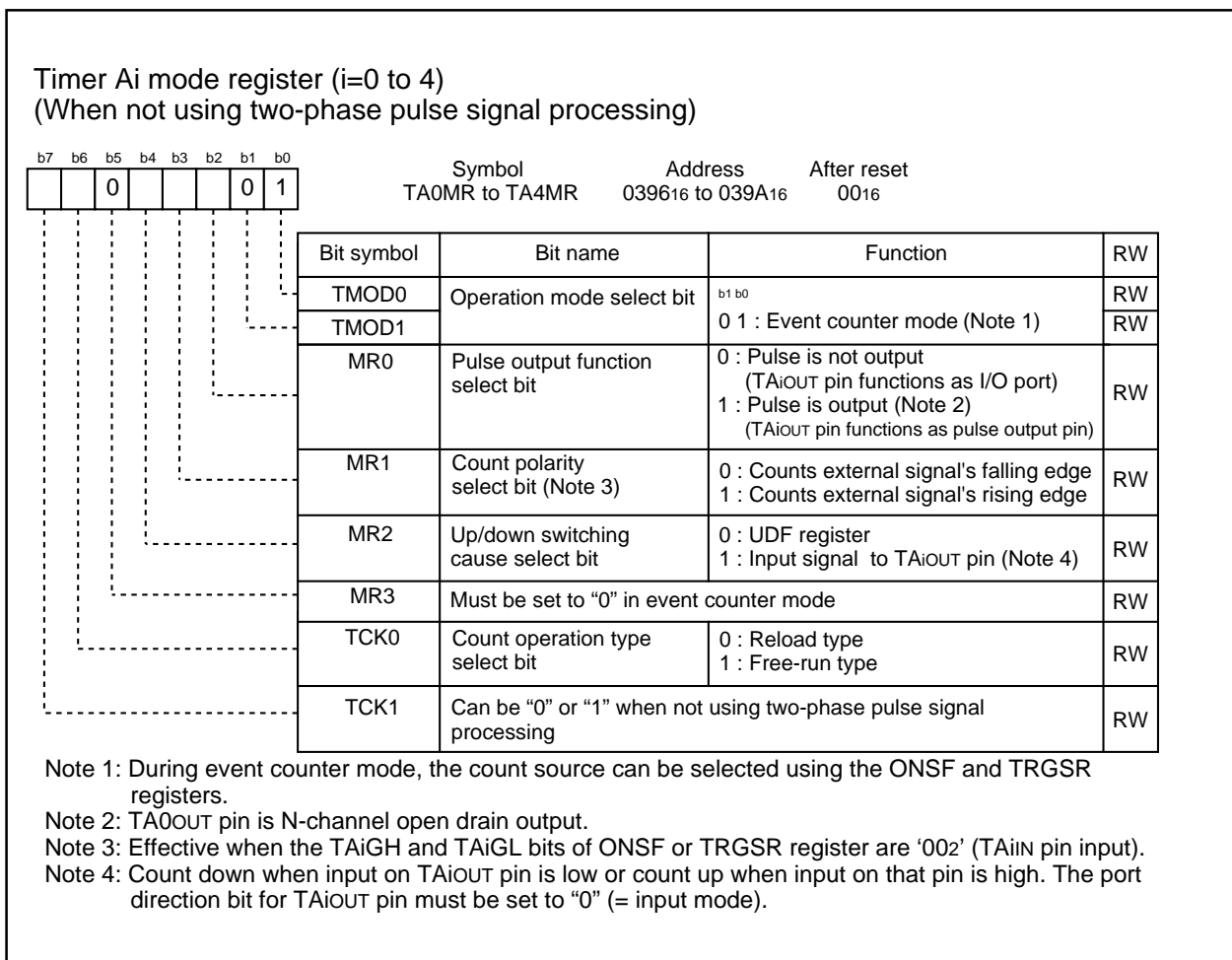
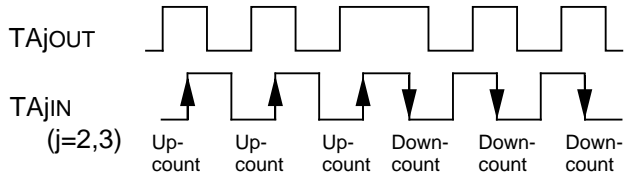
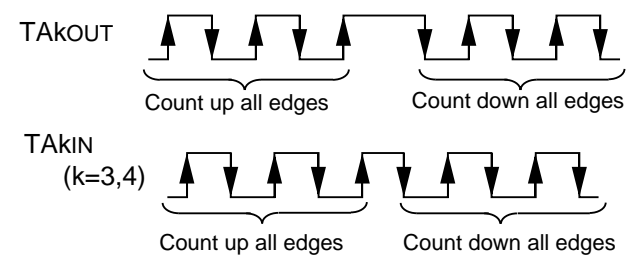


Figure 1.14.8. TAiMR Register in Event Counter Mode (when not using two-phase pulse signal processing)

Table 1.14.3. Specifications in Event Counter Mode (when processing two-phase pulse signal with timers A2, A3 and A4)

Item	Specification
Count source	• Two-phase pulse signals input to TAIIN or TAIOUT pins (i = 2 to 4)
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/ (FFFF <sub>16</sub> - n + 1) for up-count 1/ (n + 1) for down-count      n : set value of TAI register    0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	Set TAI <sub>S</sub> bit of TABSR register to "1" (= start counting)
Count stop condition	Set TAI <sub>S</sub> bit to "0" (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read by reading timer A2, A3 or A4 register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAI register is written to reload register (Transferred to counter when reloaded next)
Select function (Note)	<ul style="list-style-type: none"> <li>• Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on TAJIN pin when input signals on TAJOUT pin is "H".</li> </ul>  <ul style="list-style-type: none"> <li>• Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAKIN(k=3, 4) pin goes "H" when the input signal on TAKOUT pin is "H", the timer counts up rising and falling edges on TAKOUT and TAKIN pins. If the phase relationship is such that TAKIN pin goes "L" when the input signal on TAKOUT pin is "H", the timer counts down rising and falling edges on TAKOUT and TAKIN pins.</li> </ul>  <ul style="list-style-type: none"> <li>• Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input.</li> </ul>

Notes:

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

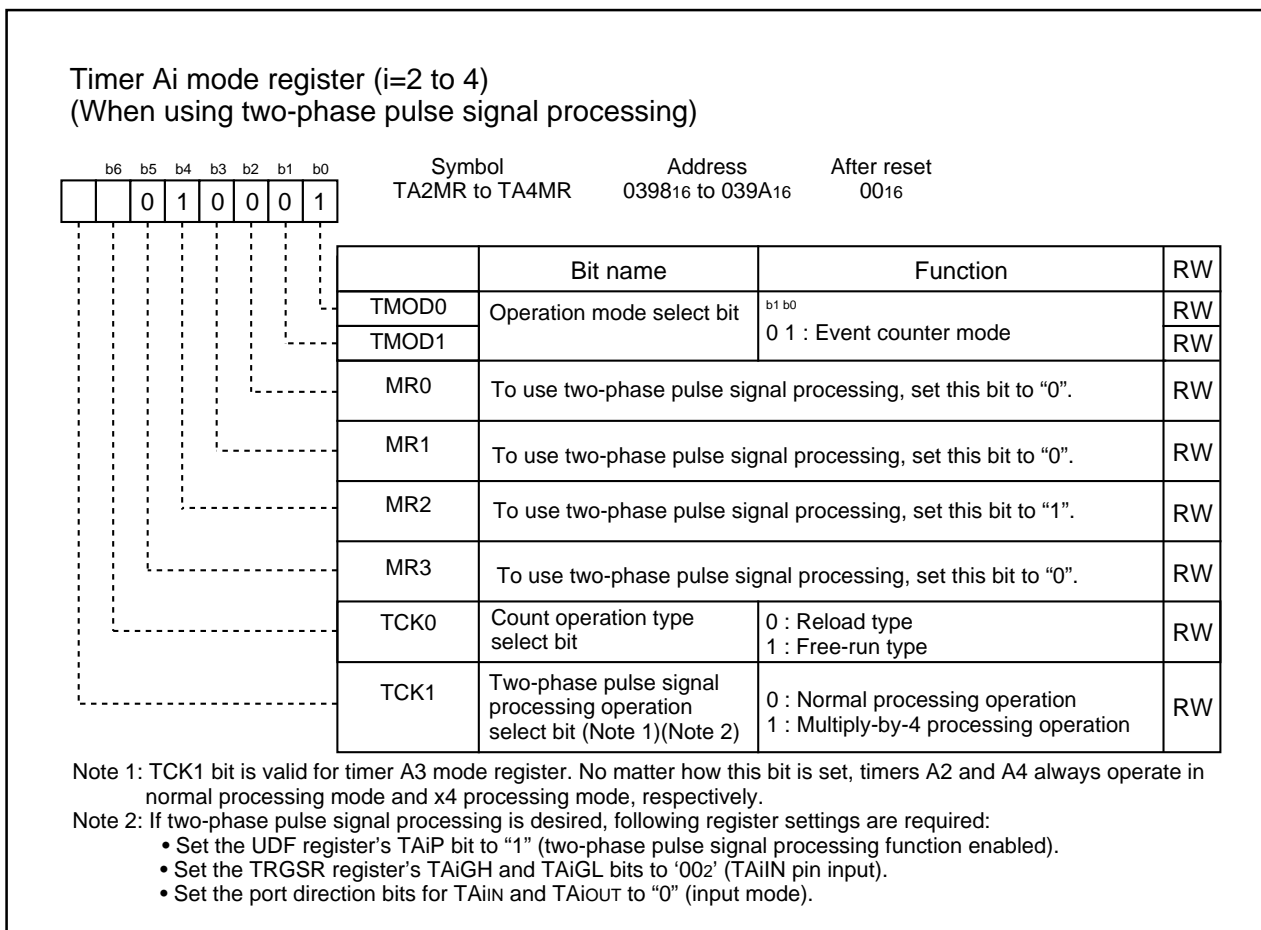


Figure 1.14.9. TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2, A3 or A4)

### • Counter Initialization by Two-Phase Pulse Signal Processing

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

This function can only be used in timer A3 event counter mode during two-phase pulse signal processing, free-running type, x4 processing, with Z-phase entered from the  $\overline{\text{INT2}}$  pin.

Counter initialization by Z-phase input is enabled by writing "0000<sub>16</sub>" to the TA3 register and setting the TAZIE bit in ONSF register to "1" (= Z-phase input enabled).

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

The counter is initialized at the next count timing after recognizing Z-phase input. Figure 1.14.10 shows the relationship between the two-phase pulse (A phase and B phase) and the Z phase.

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

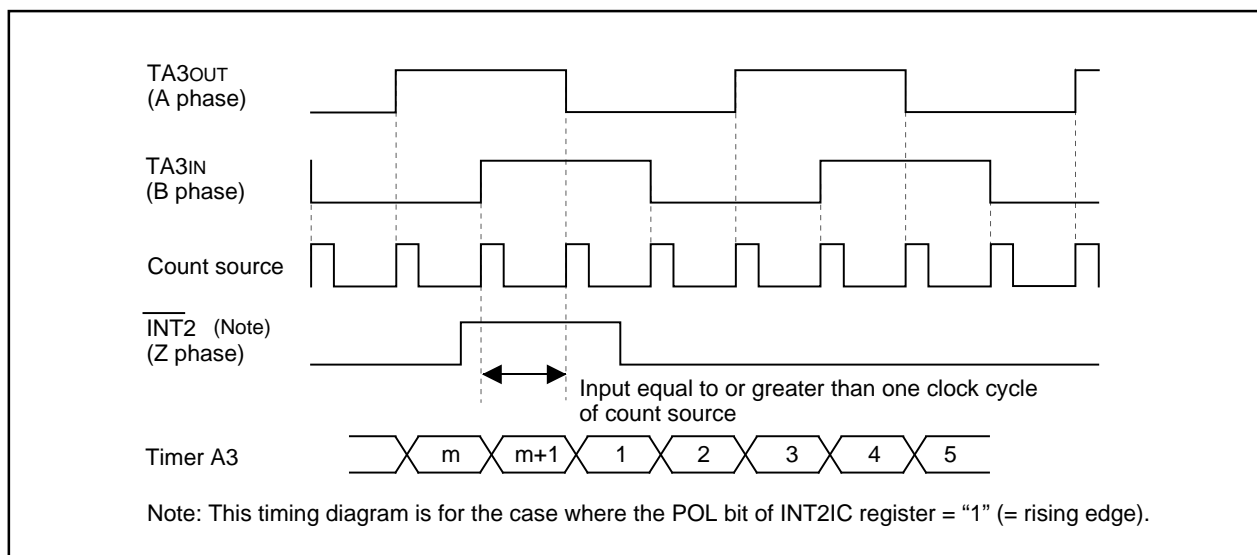


Figure 1.14.10. Two-phase Pulse (A phase and B phase) and the Z Phase

### 3. One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. (See Table 1.14.4.) When the trigger occurs, the timer starts up and continues operating for a given period. Figure 1.14.12 shows the TAI<sub>MR</sub> register in one-shot timer mode.

**Table 1.14.4. Specifications in One-shot Timer Mode**

Item	Specification
Count source	f1, f2, f8, f32, fC32
Count operation	<ul style="list-style-type: none"> <li>• Down-count</li> <li>• When the counter reaches 0000<sub>16</sub>, it stops counting after reloading a new value</li> <li>• If a trigger occurs when counting, the timer reloads a new count and restarts counting</li> </ul>
Divide ratio	$1/n$ $n$ : set value of TAI register    0000 <sub>16</sub> to FFFF <sub>16</sub> However, the counter does not work if the divide-by-n value is set to 0000 <sub>16</sub> .
Count start condition	TAI <sub>S</sub> bit of TABSR register = "1" (start counting) and one of the following triggers occurs. <ul style="list-style-type: none"> <li>• External trigger input from the TAI<sub>IN</sub> pin</li> <li>• Timer B2 overflow or underflow, timer A<sub>j</sub> (<math>j=i-1</math>, except <math>j=4</math> if <math>i=0</math>) overflow or underflow, timer A<sub>k</sub> (<math>k=i+1</math>, except <math>k=0</math> if <math>i=4</math>) overflow or underflow</li> <li>• The TAI<sub>OS</sub> bit of ONSF register is set to "1" (= timer starts)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>• When the counter is reloaded after reaching "0000<sub>16</sub>"</li> <li>• TAI<sub>S</sub> bit is set to "0" (= stop counting)</li> </ul>
Interrupt request generation timing	When the counter reaches "0000 <sub>16</sub> "
TAI <sub>IN</sub> pin function	I/O port or trigger input
TAI <sub>OUT</sub> pin function	I/O port or pulse output
Read from timer	An indeterminate value is read by reading TAI register
Write to timer	<ul style="list-style-type: none"> <li>• 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</li> <li>• When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>• Pulse output function The timer outputs a low when not counting and a high when counting.</li> </ul>

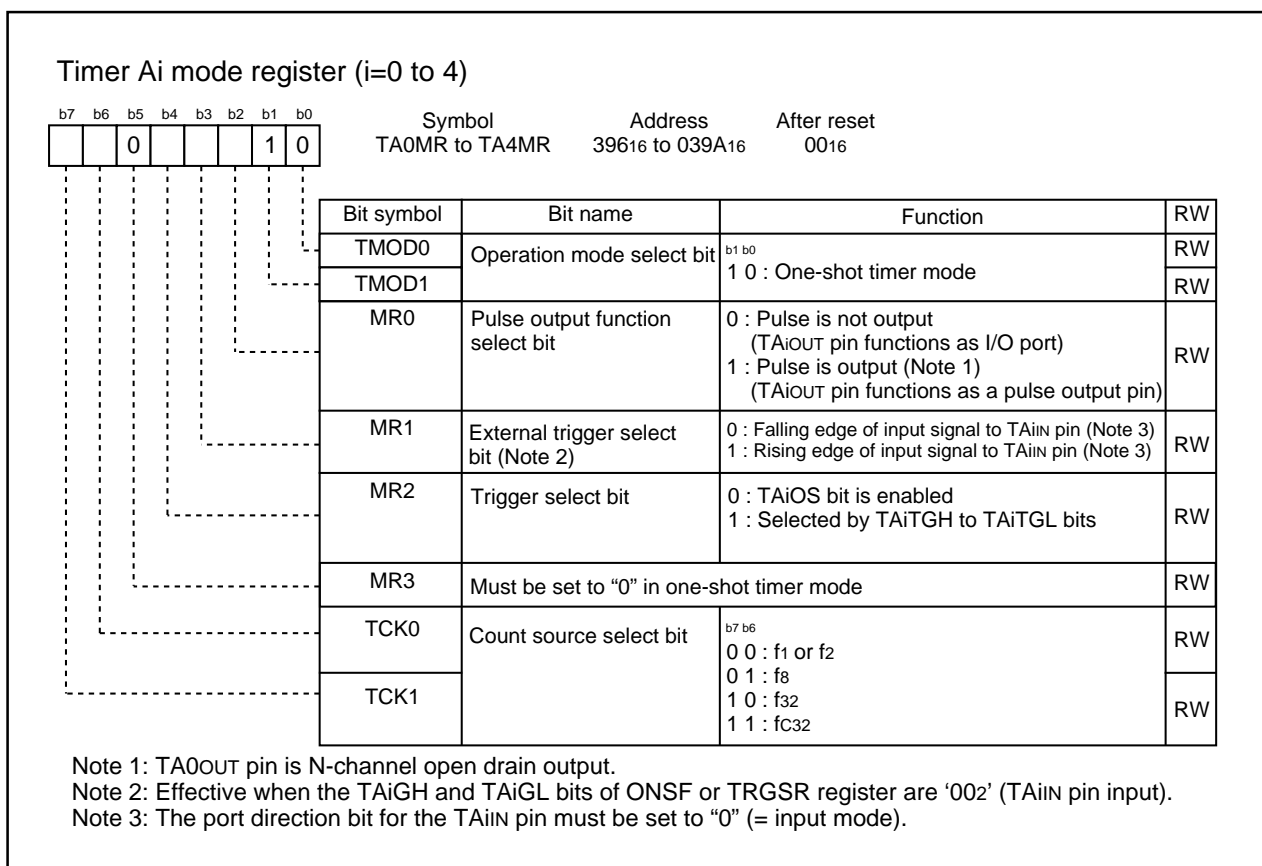


Figure 1.14.12. TAiMR Register in One-shot Timer Mode

#### 4. Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see Table 1.14.5). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 1.14.13 shows TAIiMR register in pulse width modulation mode. Figures 1.14.14 and 1.14.15 show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates.

**Table 1.14.5. Specifications in PWM Mode**

Item	Specification
Count source	f1, f2, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>• Down-count (operating as an 8-bit or a 16-bit pulse width modulator)</li> <li>• The timer reloads a new value at a rising edge of PWM pulse and continues counting</li> <li>• The timer is not affected by a trigger that occurs during counting</li> </ul>
16-bit PWM	<ul style="list-style-type: none"> <li>• High level width <math>n / f_j</math> <math>n</math>: set value of TAI register (<math>i=0</math> to 4)</li> <li>• Cycle time <math>(2^{16}-1) / f_j</math> fixed <math>f_j</math>: count source frequency (f1, f2, f8, f32, fc32)</li> </ul>
8-bit PWM	<ul style="list-style-type: none"> <li>• High level width <math>n \times (m+1) / f_j</math> <math>n</math>: set value of TAIiMR register high-order address</li> <li>• Cycle time <math>(2^8-1) \times (m+1) / f_j</math> <math>m</math>: set value of TAIiMR register low-order address</li> </ul>
Count start condition	<ul style="list-style-type: none"> <li>• External trigger input from the TAIiN pin</li> <li>• Timer B2 overflow or underflow, timer Aj (<math>j=i-1</math>, except <math>j=4</math> if <math>i=0</math>) overflow or underflow, timer Ak (<math>k=i+1</math>, except <math>k=0</math> if <math>i=4</math>) overflow or underflow</li> <li>• TAIiS bit of TABSR register is set to "1" (= start counting)</li> </ul>
Count stop condition	TAIiS bit is set to "0" (= stop counting)
Interrupt request generation timing	PWM pulse goes "L"
TAIiN pin function	I/O port or trigger input
TAIiOUT pin function	Pulse output
Read from timer	An indeterminate value is read by reading TAI register
Write to timer	<ul style="list-style-type: none"> <li>• 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</li> <li>• When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>



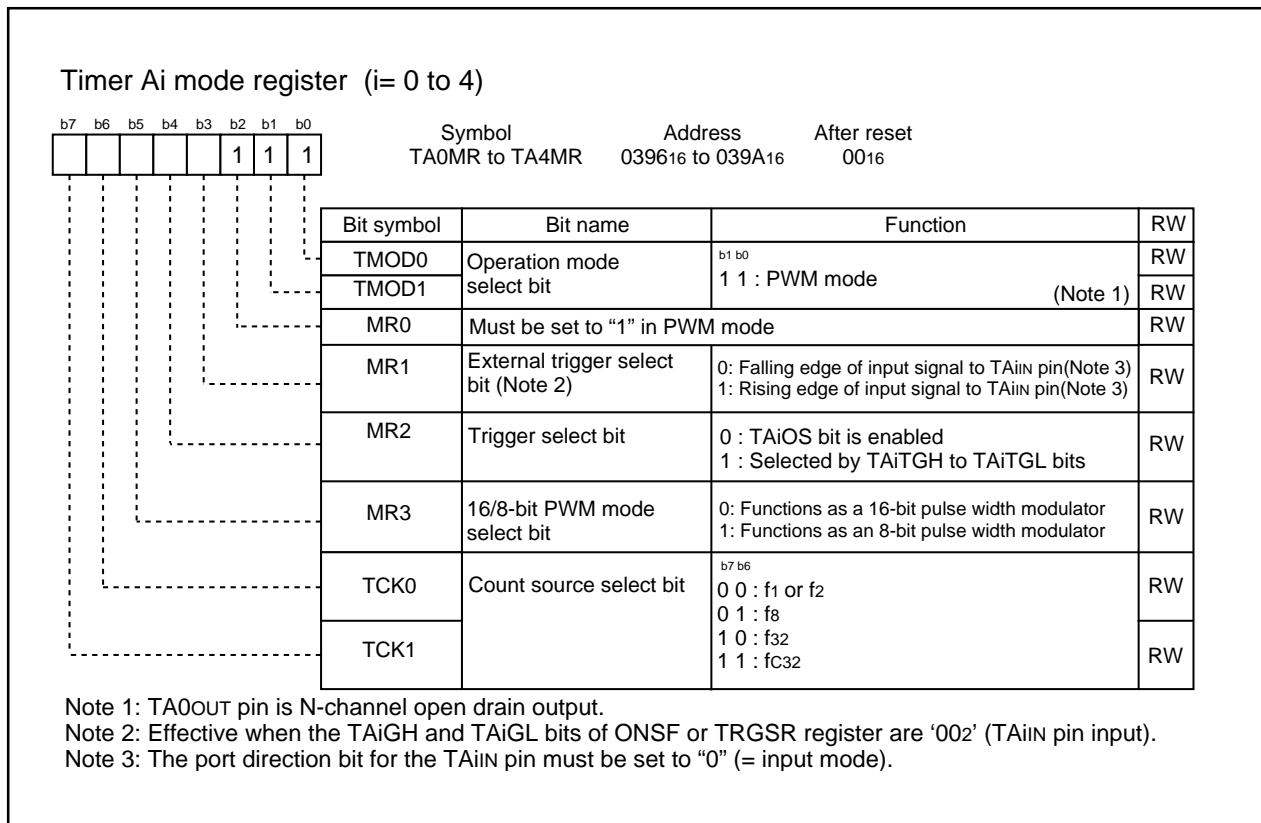


Figure 1.14.13. TAiMR Register in PWM Mode

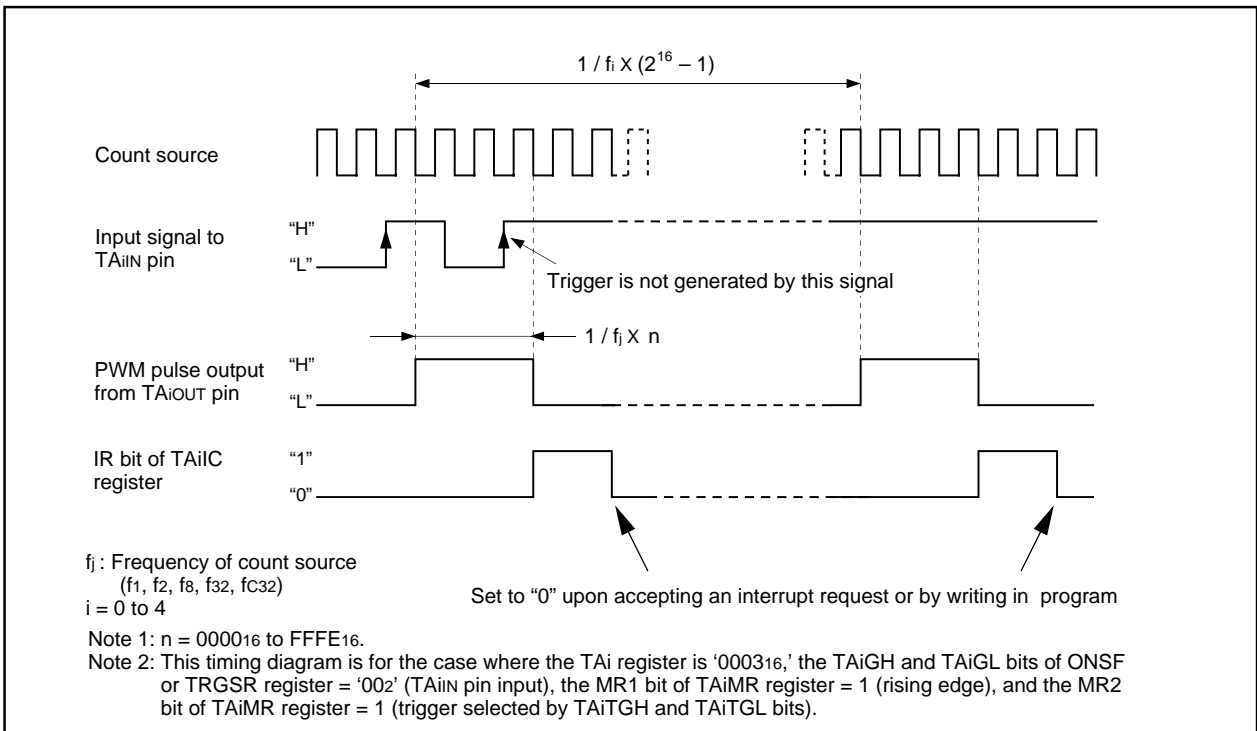


Figure 1.14.14. Example of 16-bit Pulse Width Modulator Operation

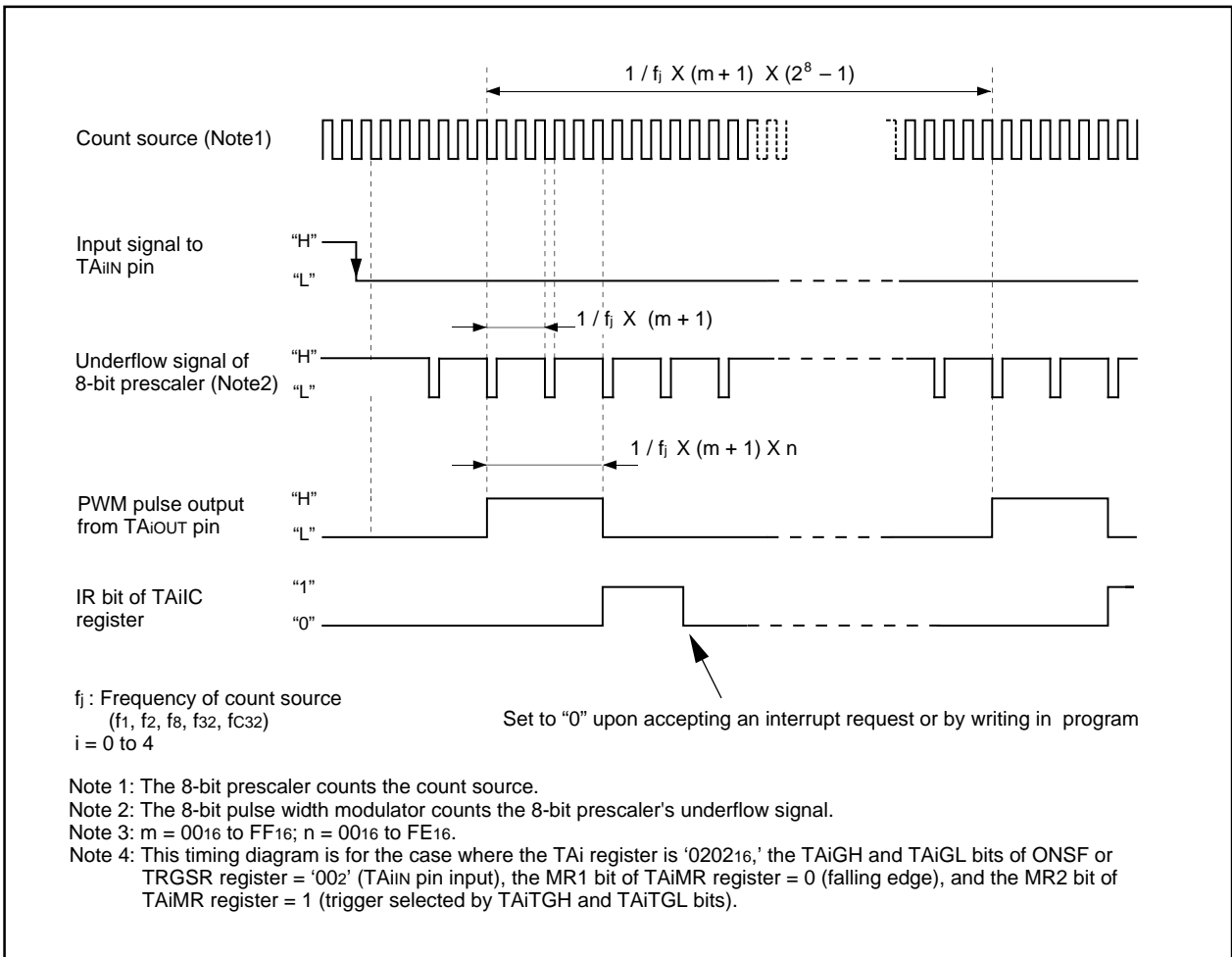


Figure 1.14.15. Example of 8-bit Pulse Width Modulator Operation

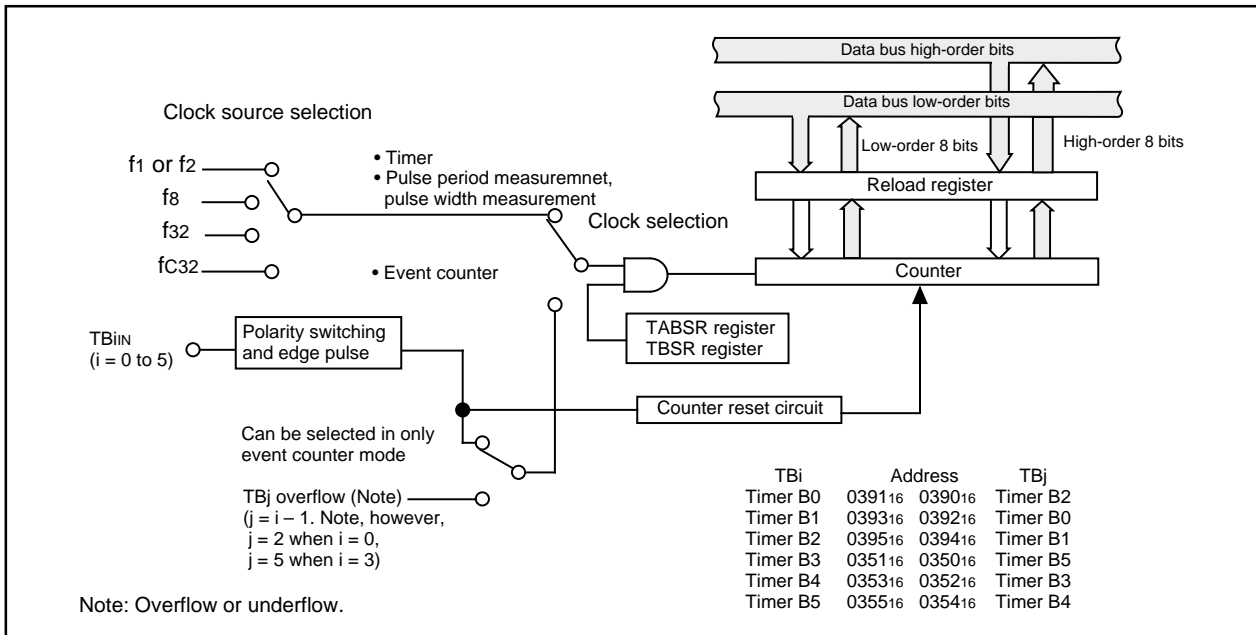
**Timers (Timer B)**

**Timer B**

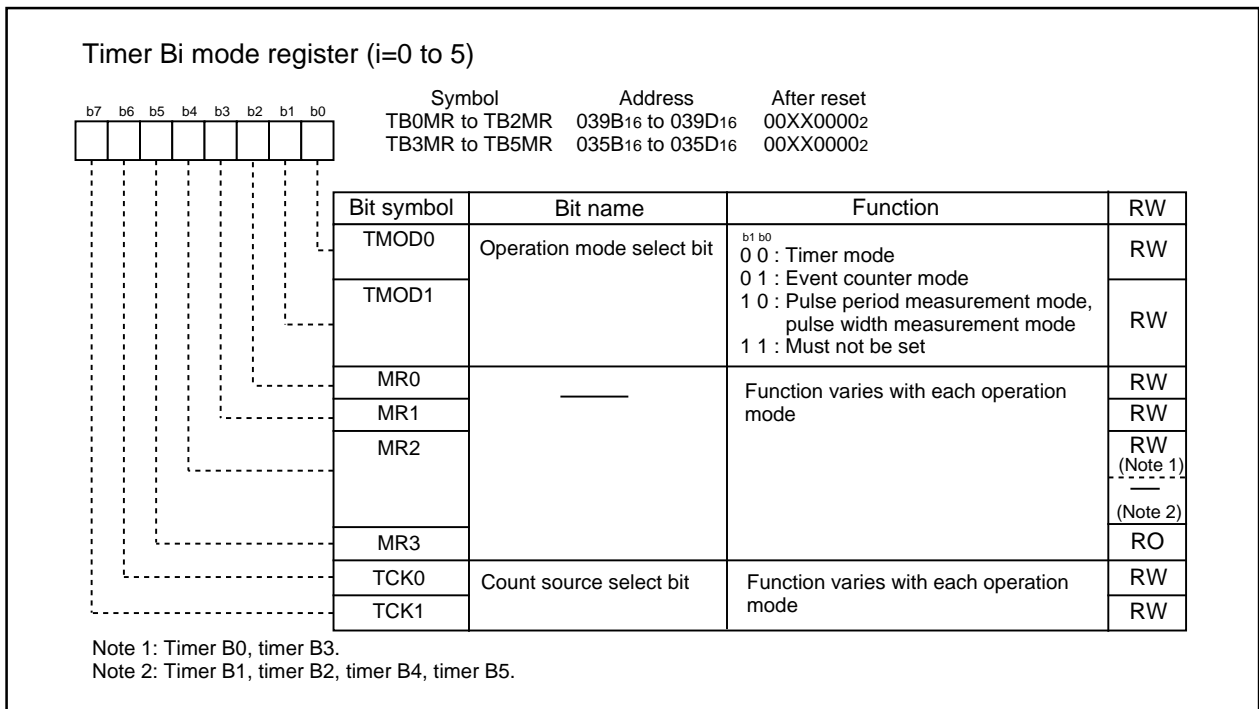
Figure 1.15.1 shows a block diagram of the timer B. Figures 1.15.2 and 1.15.3 show registers related to the timer B.

Timer B supports the following three modes. Use the TMOD1 and TMOD0 bits of TBiMR register (i = 0 to 5) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows or underflows of other timers.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.



**Figure 1.15.1. Timer B Block Diagram**



**Figure 1.15.2. TB0MR to TB5MR Registers**

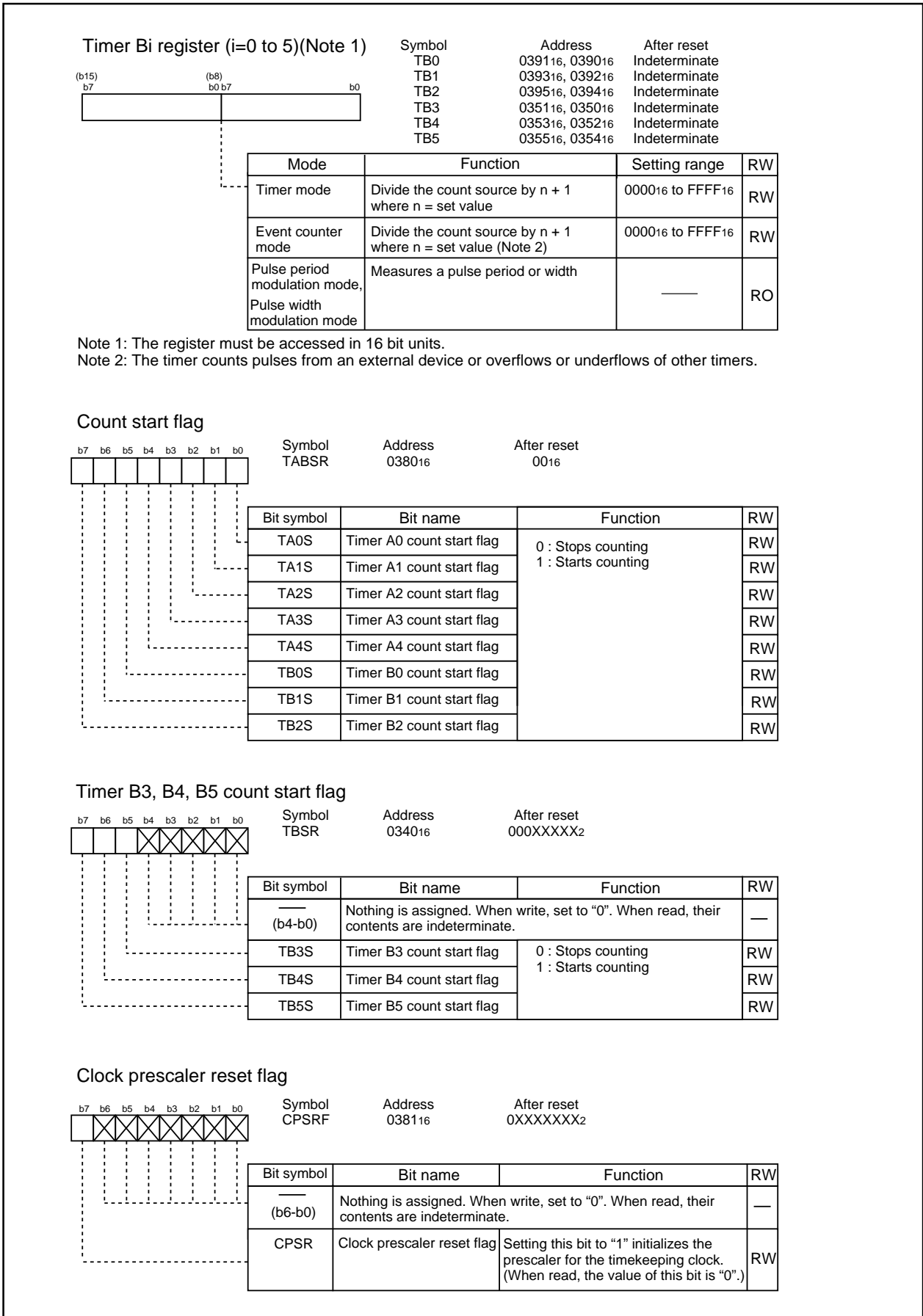


Figure 1.15.3. TB0 to TB5 Registers, TABSR Register, TBSR Register, CPSRF Register

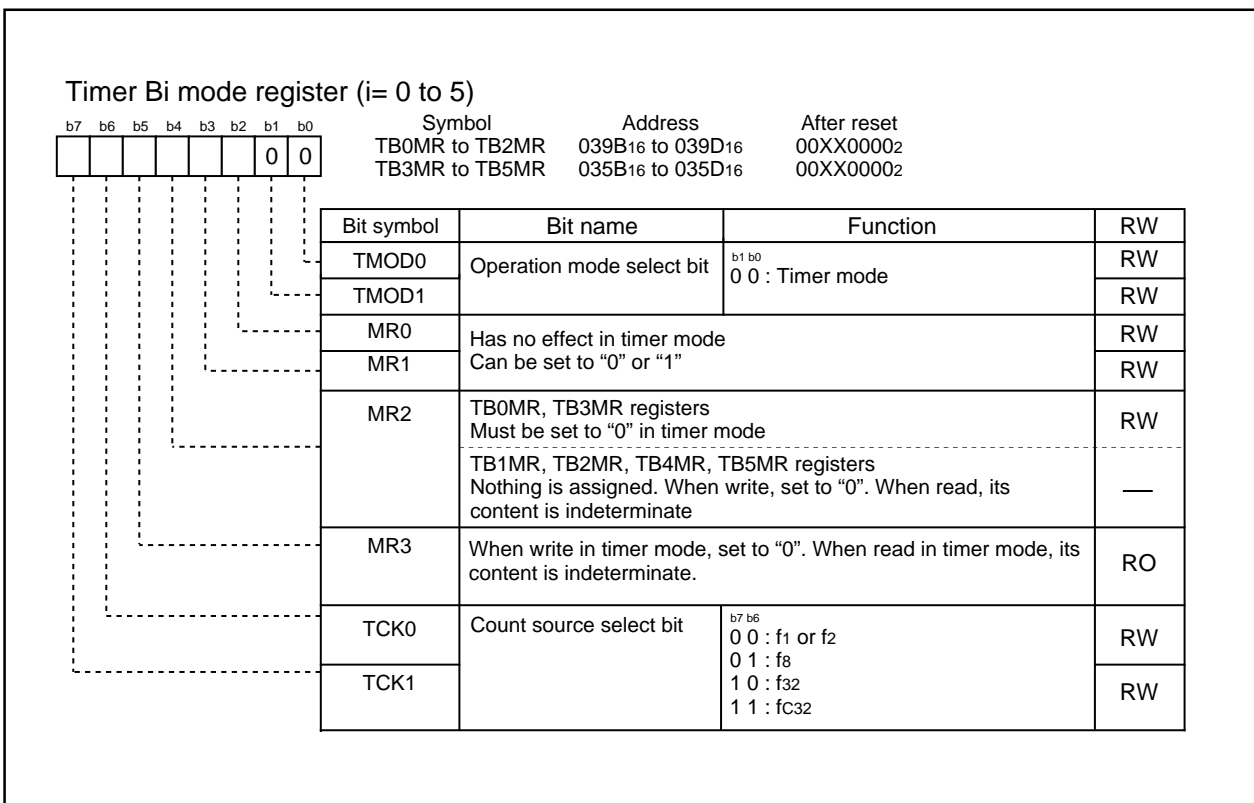
### 1. Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 1.15.1). Figure 1.15.4 shows TBiMR register in timer mode.

**Table 1.15.1. Specifications in Timer Mode**

Item	Specification
Count source	f1, f2, f8, f32, fC32
Count operation	<ul style="list-style-type: none"> <li>Down-count</li> <li>When the timer underflows, it reloads the reload register contents and continues counting</li> </ul>
Divide ratio	1/(n+1) n: set value of TBiMR register (i= 0 to 5) 0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	Set TBiS bit <sup>(Note)</sup> 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	<ul style="list-style-type: none"> <li>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</li> <li>When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>

Note : The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7, and the TB3S to TB5S bits are assigned to the TBSR register bit 5 to bit 7.



**Figure 1.15.4. TBiMR Register in Timer Mode**

## 2. Event Counter Mode

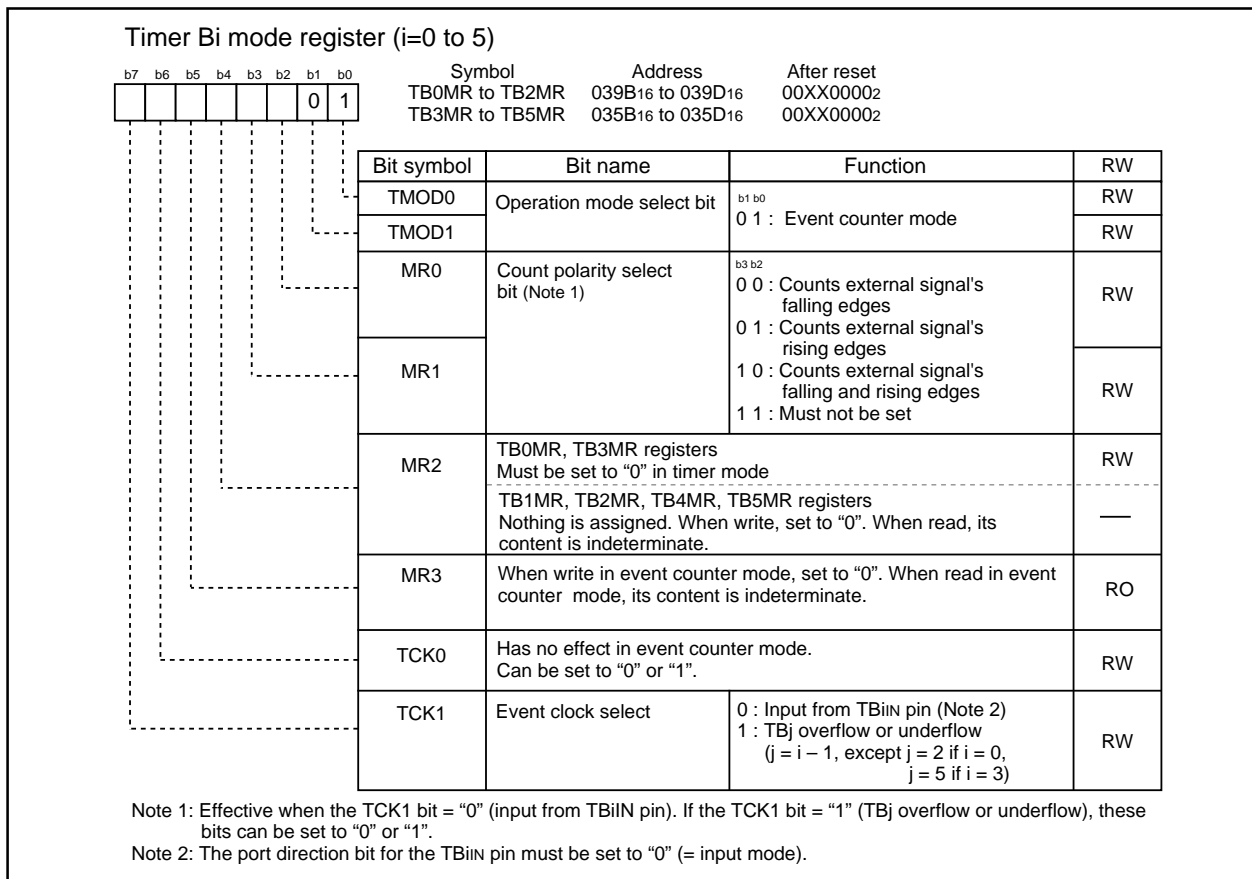
In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers (see Table 1.15.2) . Figure 1.15.5 shows TBiMR register in event counter mode.

**Table 1.15.2. Specifications in Event Counter Mode**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signals input to TBiIN pin (i=0 to 5) (effective edge can be selected in program)</li> <li>Timer Bj overflow or underflow (j=i-1, except j=2 if i=0, j=5 if i=3)</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Down-count</li> <li>When the timer underflows, it reloads the reload register contents and continues counting</li> </ul>
Divide ratio	1/(n+1)      n: set value of TBi register      0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	Set TBiS bit <sup>1</sup> 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	<ul style="list-style-type: none"> <li>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</li> <li>When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)</li> </ul>

Notes:

- The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7, and the TB3S to TB5S bits are assigned to the TBSR register bit 5 to bit 7.



**Figure 1.15.5. TBiMR Register in Event Counter Mode**

### 3. Pulse Period and Pulse Width Measurement Mode

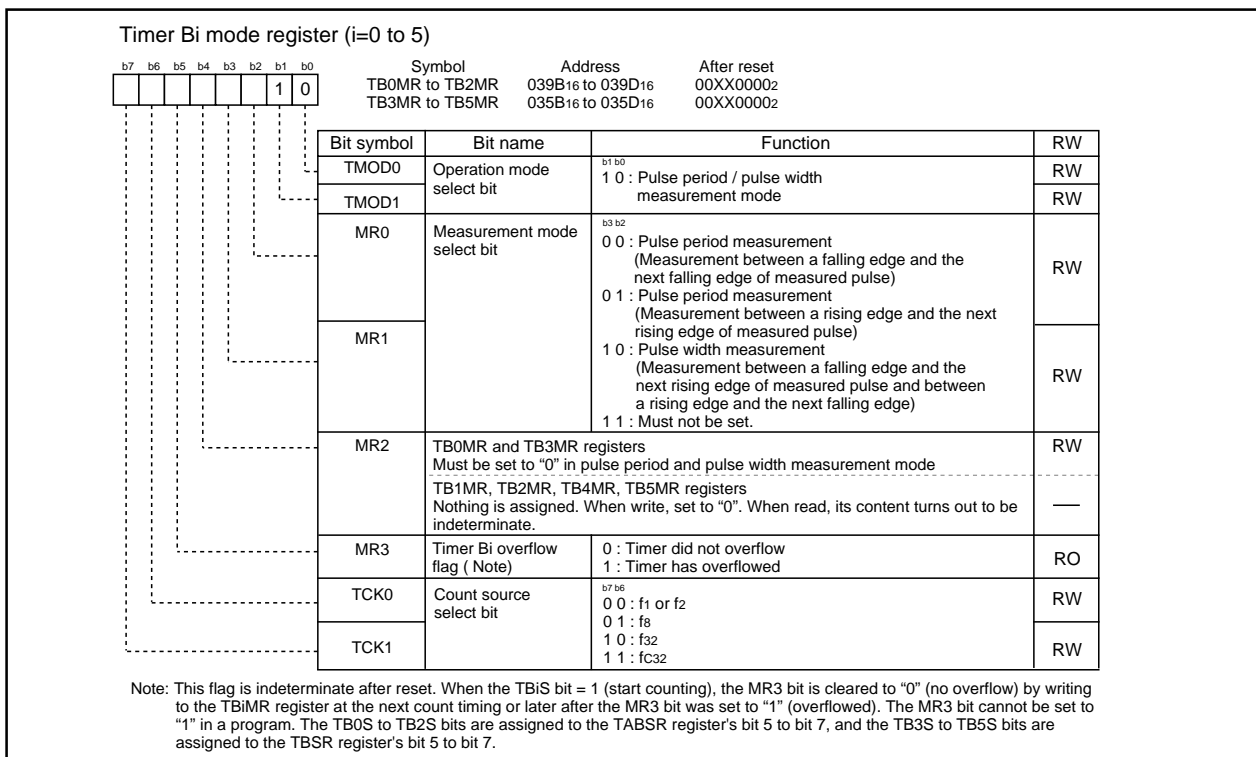
In pulse period and pulse width measurement mode, the timer measures pulse period<sup>1</sup> or pulse width of an external signal (see Table 1.15.3). Figure 1.15.6 shows TBiMR register in pulse period and pulse width measurement mode. Figure 1.15.7 shows the operation timing when measuring a pulse period. Figure 1.15.8 shows the operation timing when measuring a pulse width.

**Table 1.15.3. Specifications in Pulse Period and Pulse Width Measurement Mode**

Item	Specification
Count source	f1, f2, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>Up-count</li> <li>Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to "000016" to continue counting.</li> </ul>
Count start condition	Set TBiS (i=0 to 5) bit <sup>3</sup> to "1" (= start counting)
Count stop condition	Set TBiS bit to "0" (= stop counting)
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When an effective edge of measurement pulse is input<sup>1</sup></li> <li>Timer overflow. When an overflow occurs, MR3 bit of TBiMR register is set to "1" (overflowed) simultaneously. MR3 bit is cleared 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).</li> </ul>
TBiIN pin function	Measurement pulse input
Read from timer	Contents of the reload register (measurement result) can be read by reading TBi register <sup>2</sup>
Write to timer	Value written to TBi register is written to neither reload register nor counter

Notes:

- Interrupt request is not generated when the first effective edge is input after the timer started counting.
- Value read from TBi register is indeterminate until the second valid edge is input after the timer starts counting.
- The TB0S to TB2S bits are assigned to the TABSR register bit 5 to bit 7, and the TB3S to TB5S bits are assigned to the TBSR register bit 5 to bit 7.



**Figure 1.15.6. TBiMR Register in Pulse Period and Pulse Width Measurement Mode**

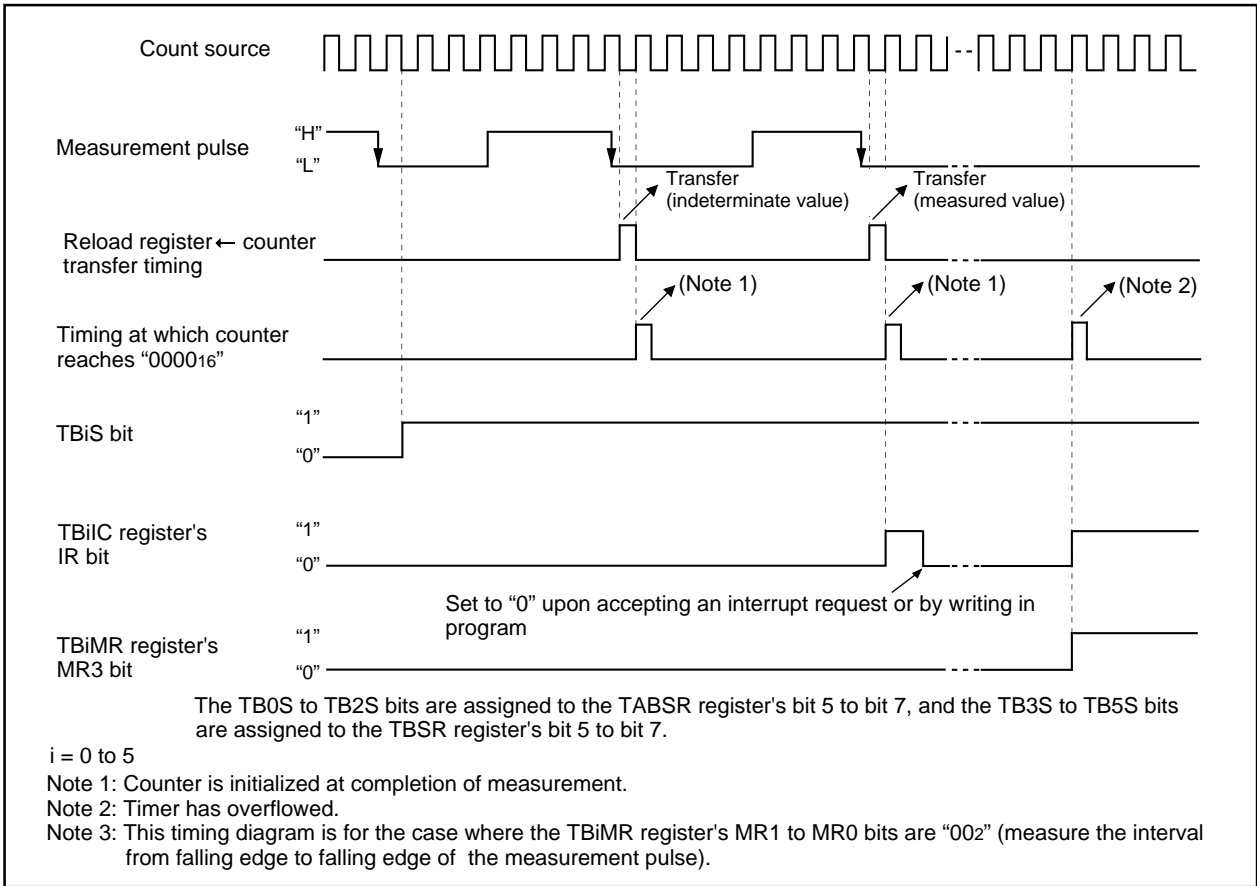


Figure 1.15.7. Operation timing when measuring a pulse period

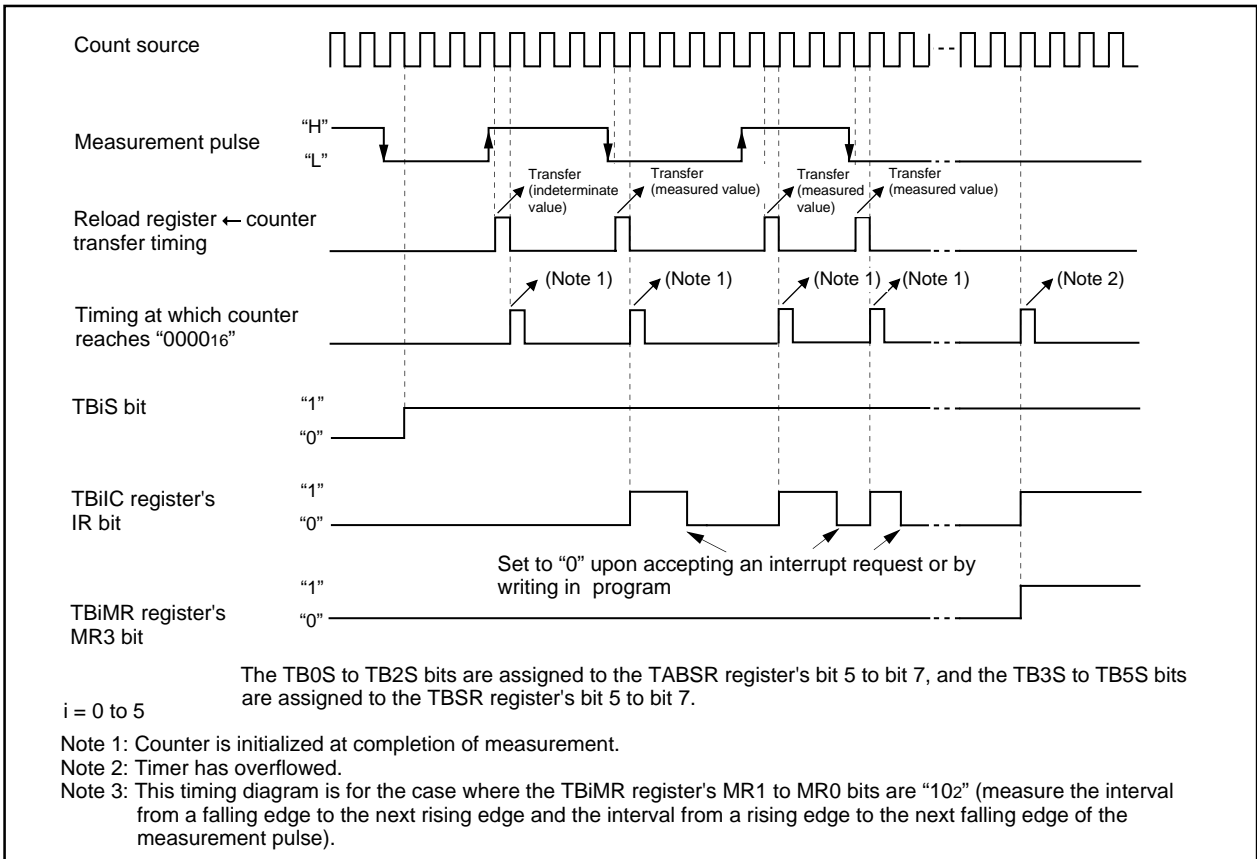


Figure 1.15.8. Operation timing when measuring a pulse width



## Three-phase Motor Control Timer Function

Timers A1, A2, A4 and B2 can be used to output three-phase motor drive waveforms. Table 1.16.1 lists the specifications of the three-phase motor control timer function. Figure 1.16.1 shows the block diagram for three-phase motor control timer function. Also, the related registers are shown on Figure 1.16.2 to Figure 1.16.7.

**Table 1.16.1. Three-phase Motor Control Timer Functions Specifications**

Item	Specification
Three-phase waveform output pin	Six pins (U, U, V, V, W, W)
Forced cutoff input <sup>1</sup>	Input "L" to NMI pin
Used Timers	Timer A4, A1, A2 (used in the one-shot timer mode) Timer A4: U- and $\bar{U}$ -phase waveform control Timer A1: V- and $\bar{V}$ -phase waveform control Timer A2: W- and $\bar{W}$ -phase waveform control Timer B2 (used in the timer mode) Carrier wave cycle control Dead timer timer (3 eight-bit timer and shared reload register) Dead time control
Output waveform	Triangular wave modulation, Sawtooth wave modification Enable to output "H" or "L" for one cycle Enable to set positive-phase level and negative-phase level respectively
Carrier wave cycle	Triangular wave modulation: count source x (m+1) x 2 Sawtooth wave modulation: count source x (m+1) m: Setting value of TB2 register, 0 to 65535 Count source: f <sub>1</sub> , f <sub>2</sub> , f <sub>8</sub> , f <sub>32</sub> , f <sub>32</sub>
Three-phase PWM output width	Triangular wave modulation: count source x n x 2 Sawtooth wave modulation: count source x n n: Setting value of TA4, TA1 and TA2 register (of TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11 bit to "1"), 1 to 65535 Count source: f <sub>1</sub> , f <sub>2</sub> , f <sub>8</sub> , f <sub>32</sub> , f <sub>32</sub>
Dead time active disable function	Count source x p, or no dead time p: Setting value of DTT register, 1 to 255 Count source: f <sub>1</sub> , f <sub>2</sub> , f <sub>1</sub> divided by 2, f <sub>2</sub> divided by 2
Active level	Enable to select "H" or "L"
Positive and negative-phase concurrent	Positive and negative-phases concurrent active disable function Positive and negative-phases concurrent active detect function
Interrupt frequency	For Timer B2 interrupt, select a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis

### Notes:

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

Related pins      P72/CLK2/TA1OUT/V  
                       P73/CTS2/RTS2/TA1IN/ $\bar{V}$   
                       P74/TA2OUT/W  
                       P75/TA2IN/ $\bar{W}$   
                       P80/TA4OUT/U  
                       P81/TA4IN/ $\bar{U}$

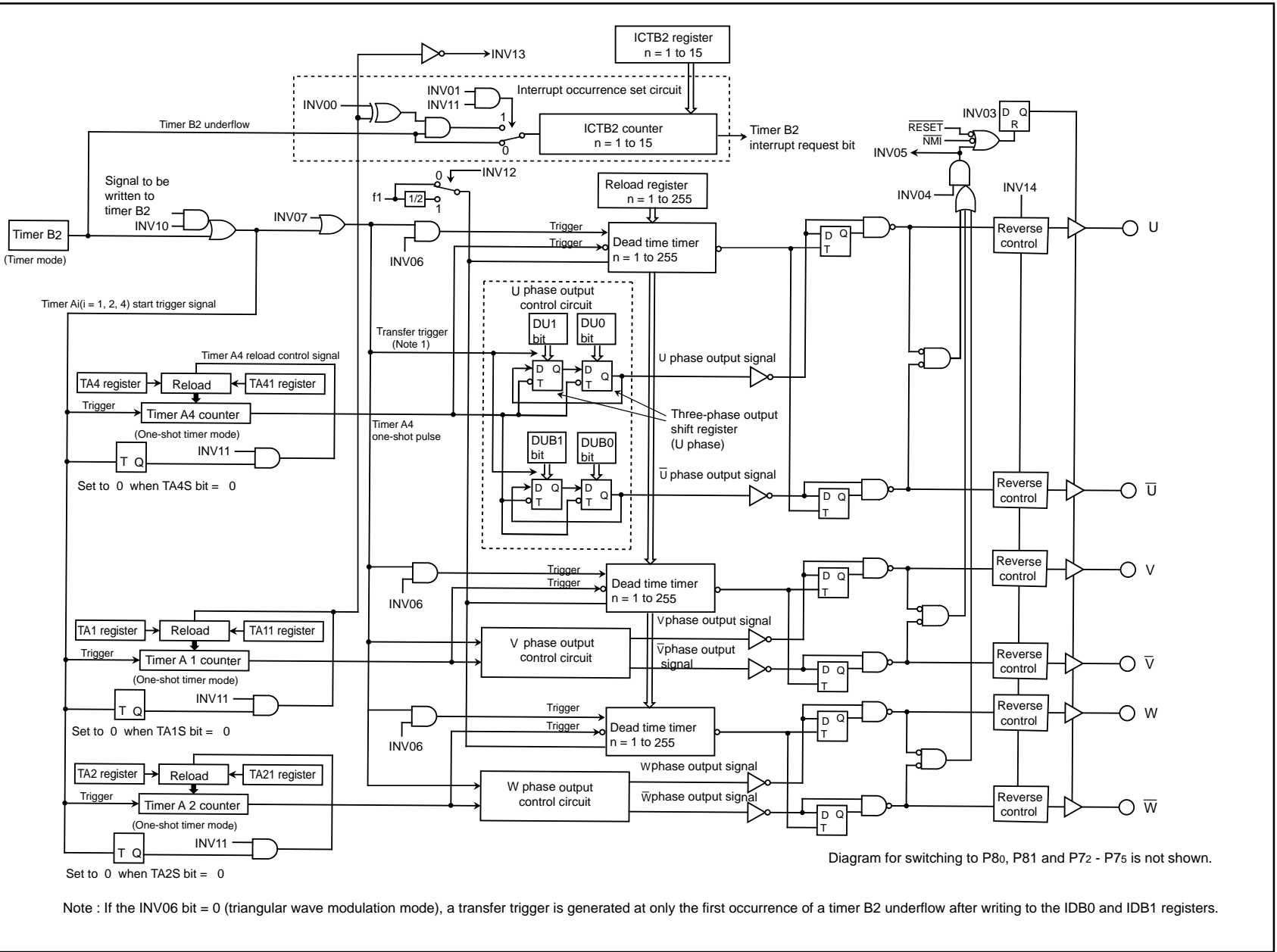


Diagram for switching to P80, P81 and P72 - P75 is not shown.

Note : If the INV06 bit = 0 (triangular wave modulation mode), a transfer trigger is generated at only the first occurrence of a timer B2 underflow after writing to the IDB0 and IDB1 registers.

Figure 1.16.1. Three-phase Motor Control Timer Functions Block Diagram

Three-phase Motor Control Timer Functions

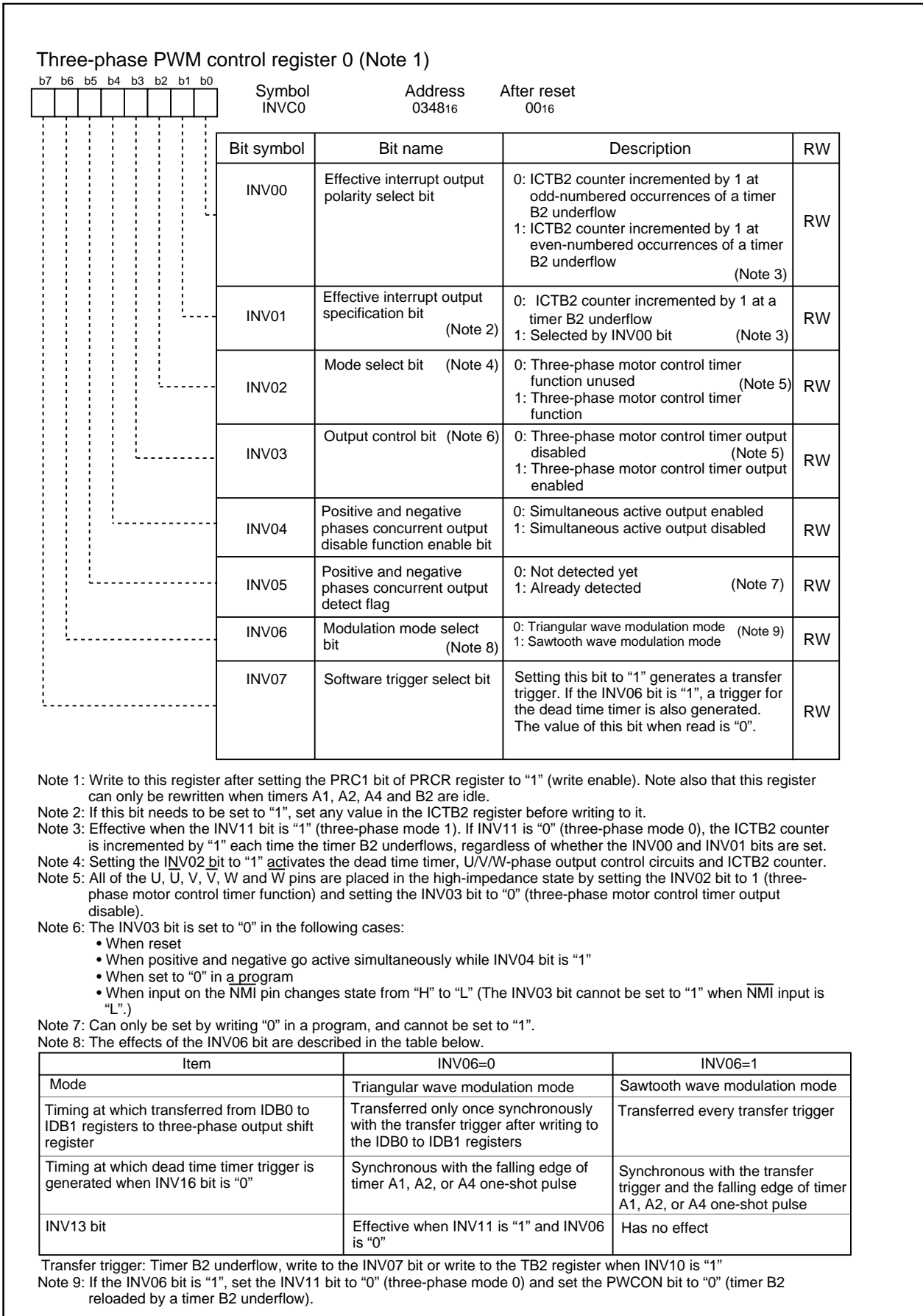


Figure 1.16.2. INVC0 Register

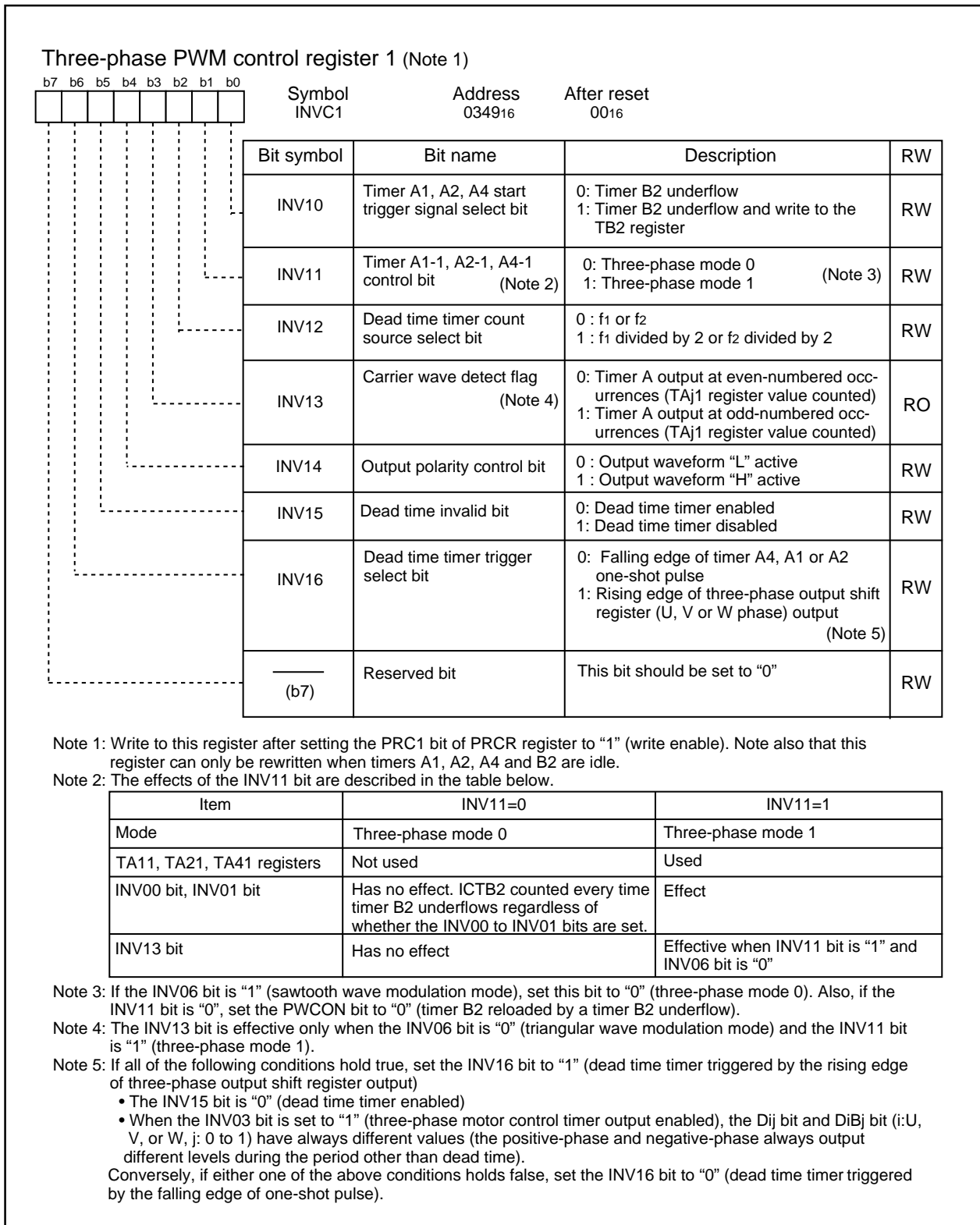


Figure 1.16.3. INVC1 Register

Three-phase Motor Control Timer Functions

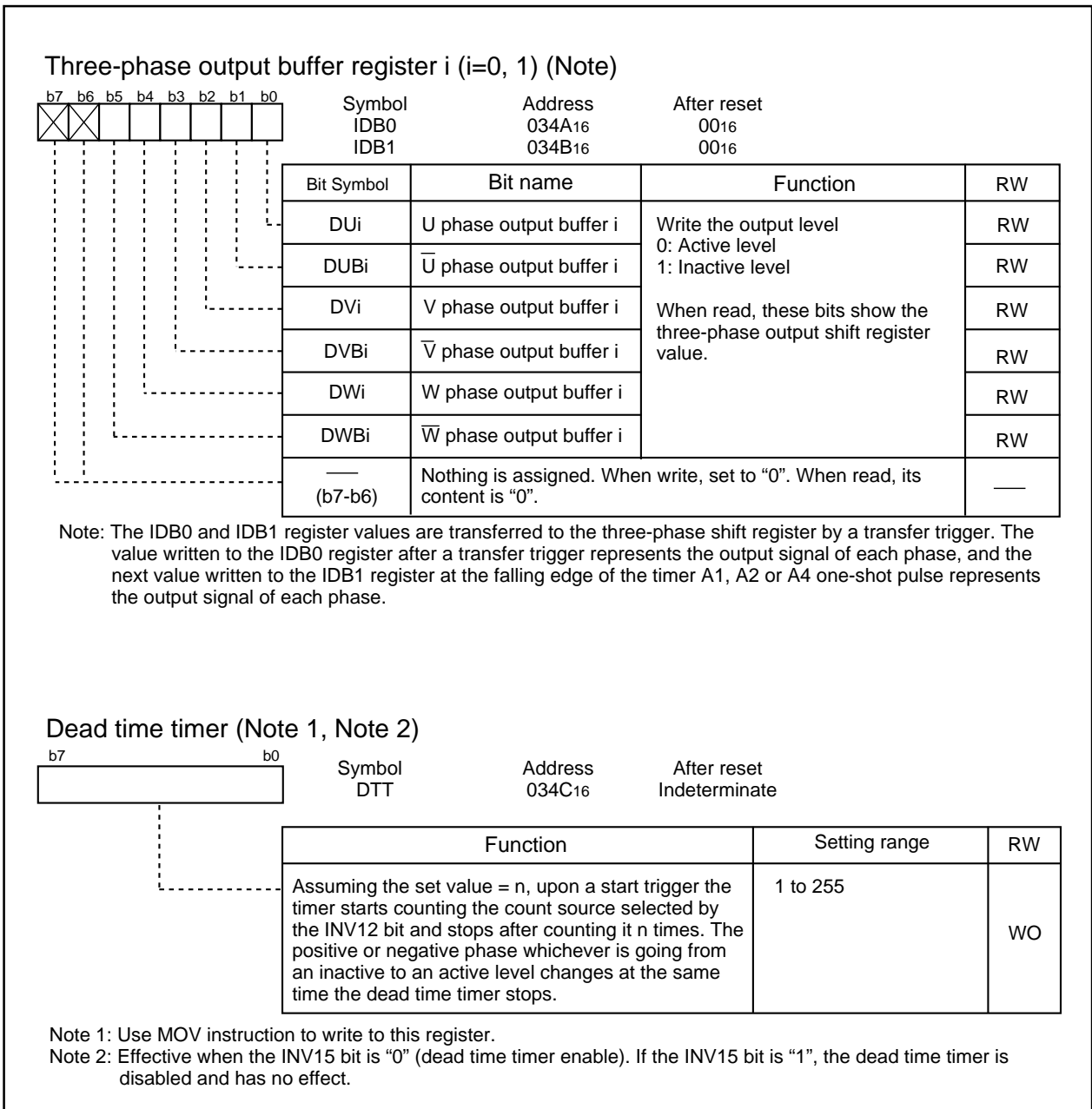


Figure 1.16.4. IDB0 Register, IDB1 Register, and DTT Register

Three-phase Motor Control Timer Functions

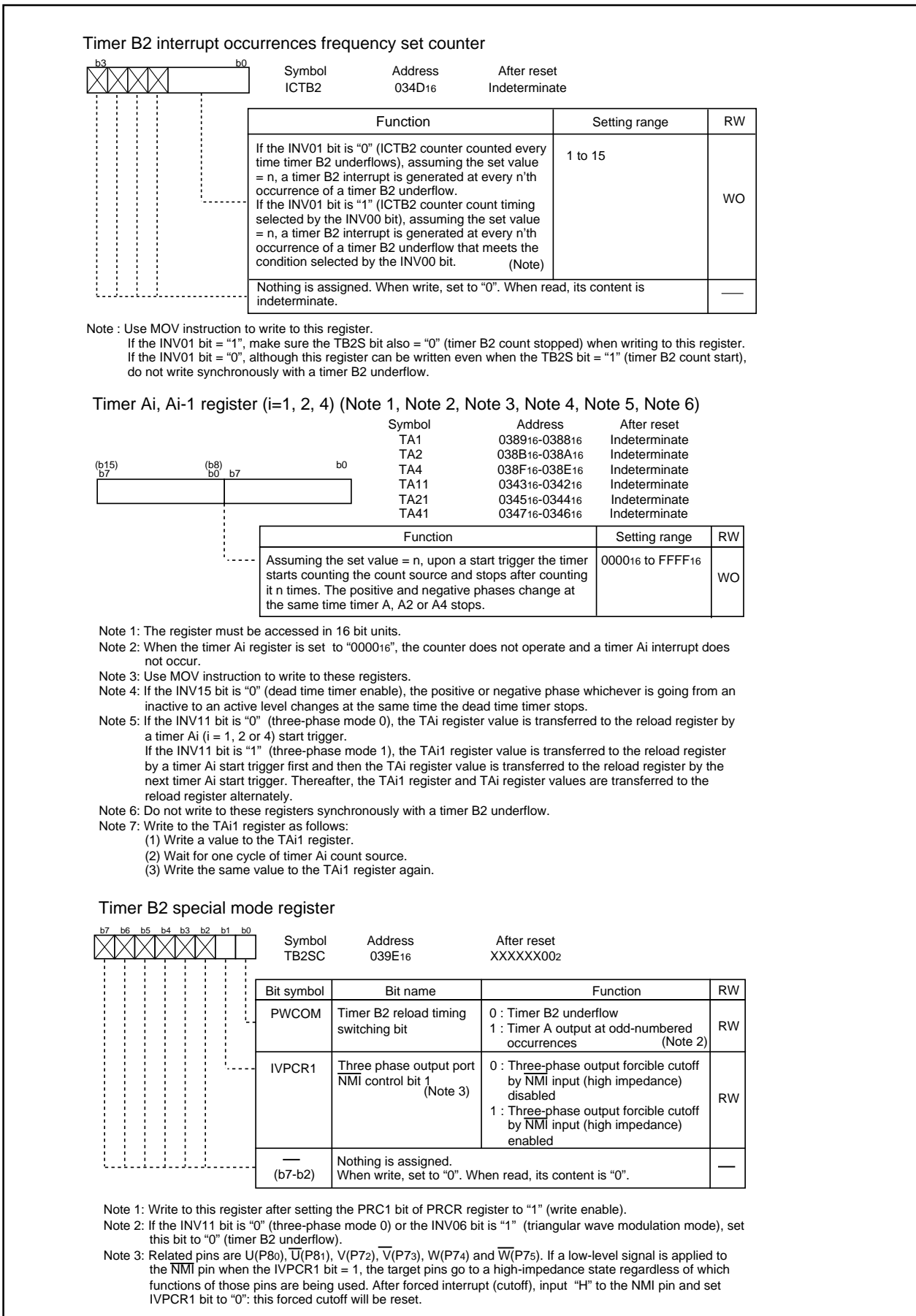


Figure 1.16.5. ICTB2 Register, TA1, TA2, TA4, TA11, TA21 and TA41 Registers, and TB2SC Registers

Three-phase Motor Control Timer Functions

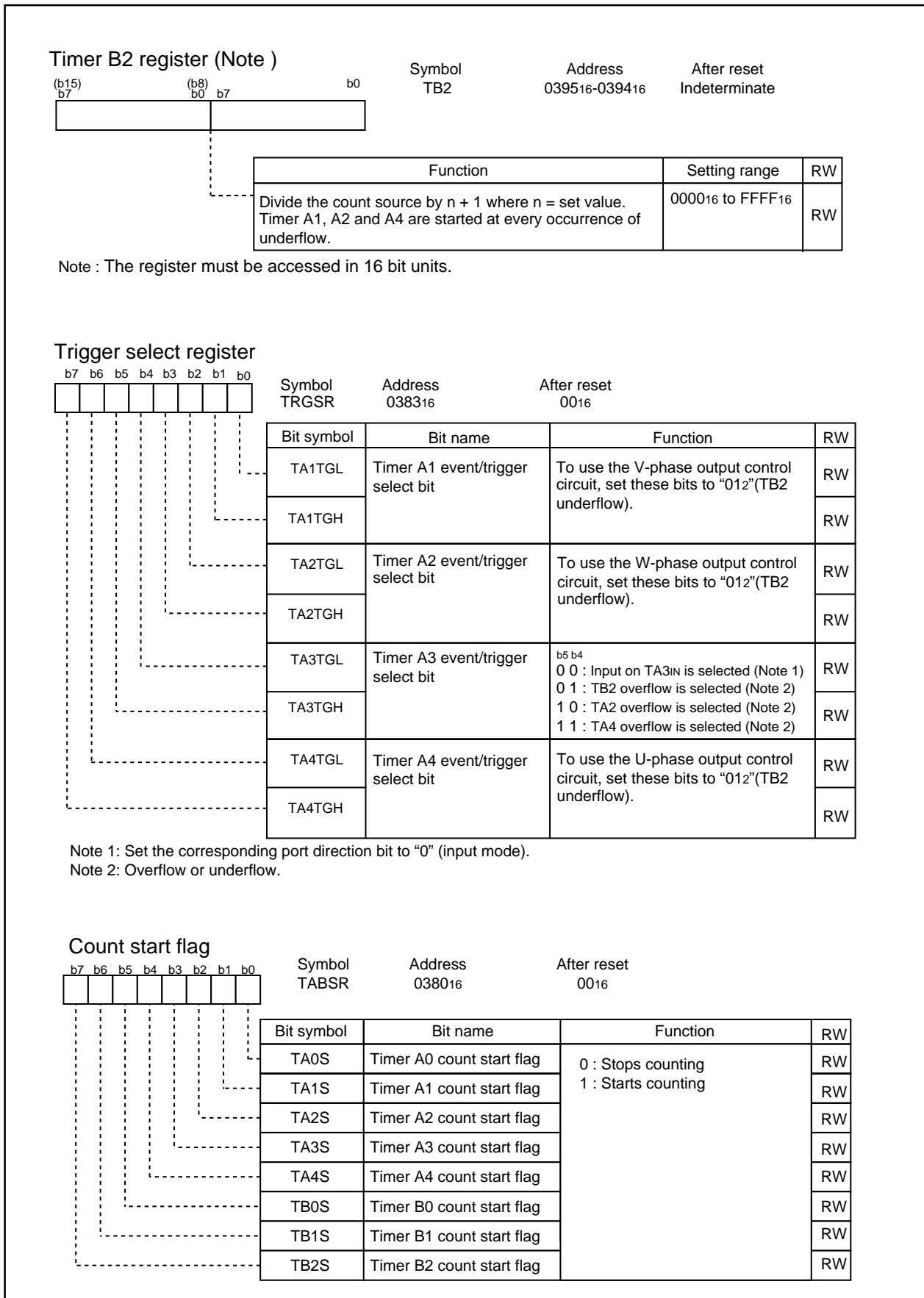


Figure 1.16.6. TB2 Register, TRGSR Register, and TABSR Register

Three-phase Motor Control Timer Functions

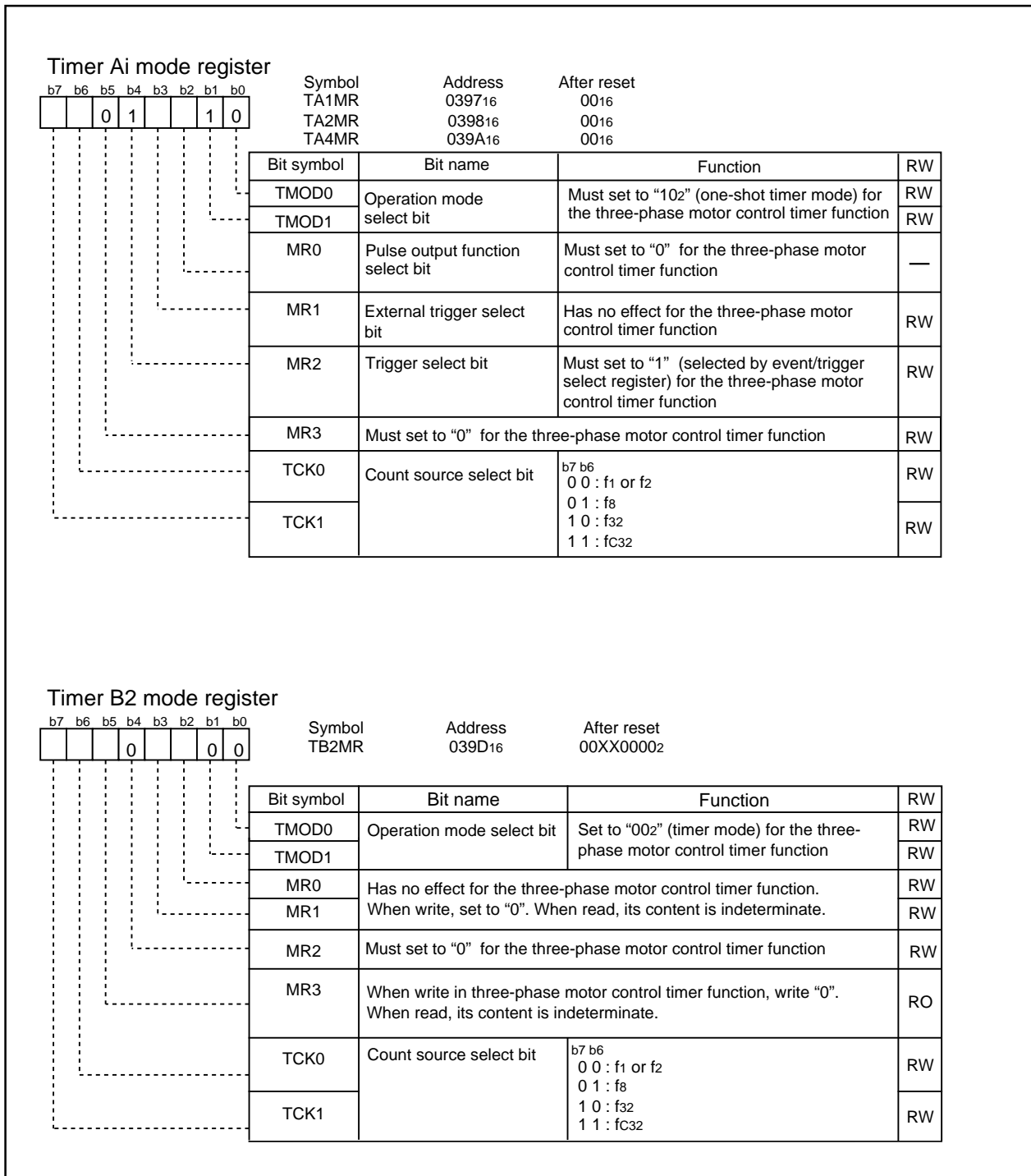


Figure 1.16.7. TA1MR, TA2MR, TA4MR, and TB2MR Registers



Three-phase Motor Control Timer Functions

The three-phase motor control timer function is enabled by setting the INV02 bit of INVC0 register to "1". When this function is on, timer B2 is used to control the carrier wave, and timers A4, A1 and A2 are used to control three-phase PWM outputs (U,  $\bar{U}$ , V,  $\bar{V}$ , W and  $\bar{W}$ ). The dead time is controlled by a dedicated dead time timer. Figure 1.16.8 shows the example of triangular modulation waveform and Figure 1.16.9 shows the example of sawtooth modulation waveform.

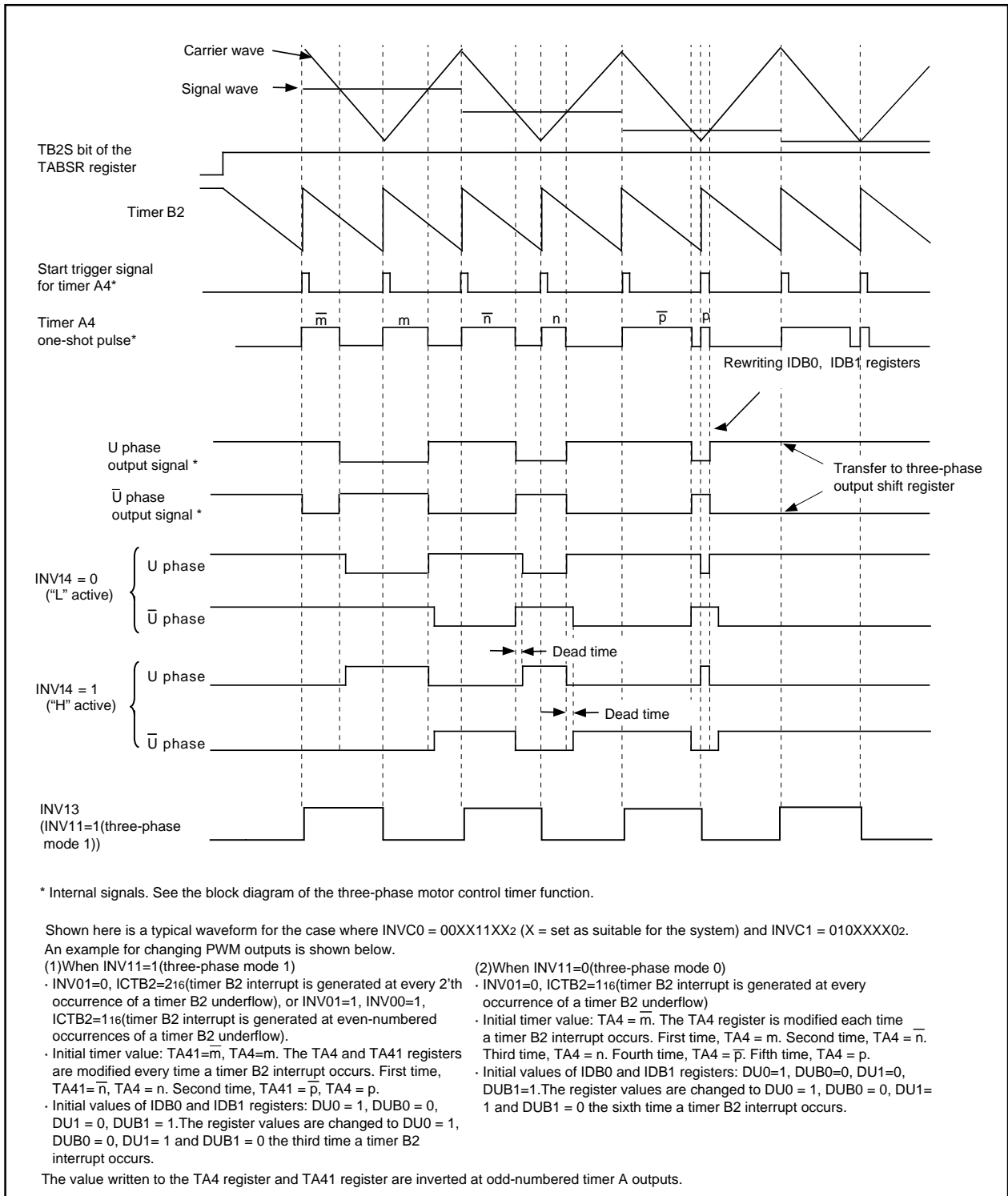
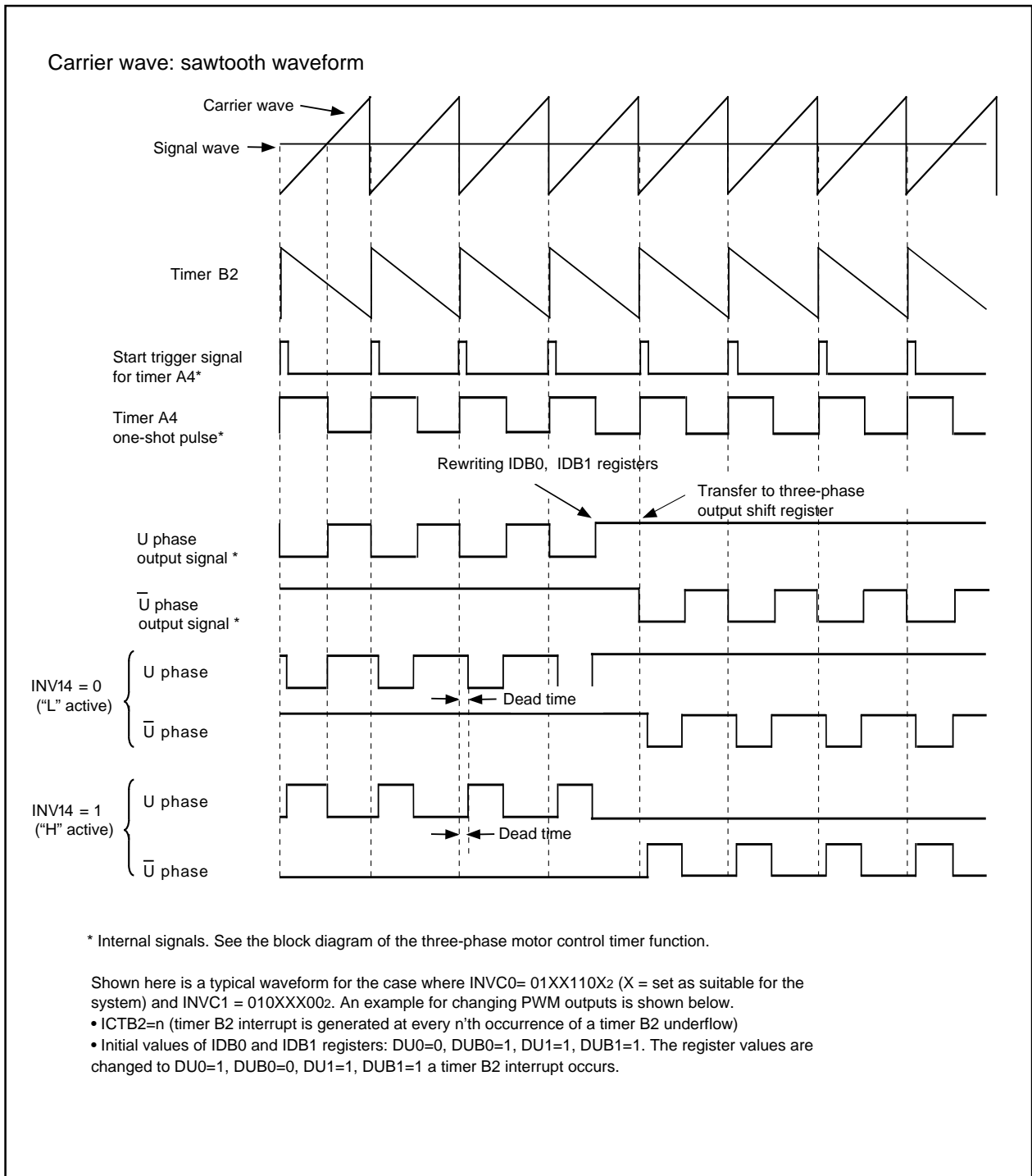


Figure 1.16.8. Triangular Wave Modulation Operation



**Figure 1.16.9. Sawtooth Wave Modulation Operation**

## Serial I/O

Serial I/O is configured with five channels: UART0 to UART2, SI/O3 and SI/O4.

### UARTi (i=0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 1.17.1 shows the block diagram of UARTi. Figures 1.17.2 shows the block diagram of the UARTi transmit/receive.

UARTi has the following modes:

- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode).
- Special mode 1 (I<sup>2</sup>C mode)
- Special mode 2
- Special mode 3 (Bus collision detection function, IE mode) : UART0, UART1
- Special mode 4 (SIM mode) : UART2

Figures 1.17.3 to 1.17.8 show the UARTi-related registers.

Refer to tables listing each mode for register setting.

Serial I/O

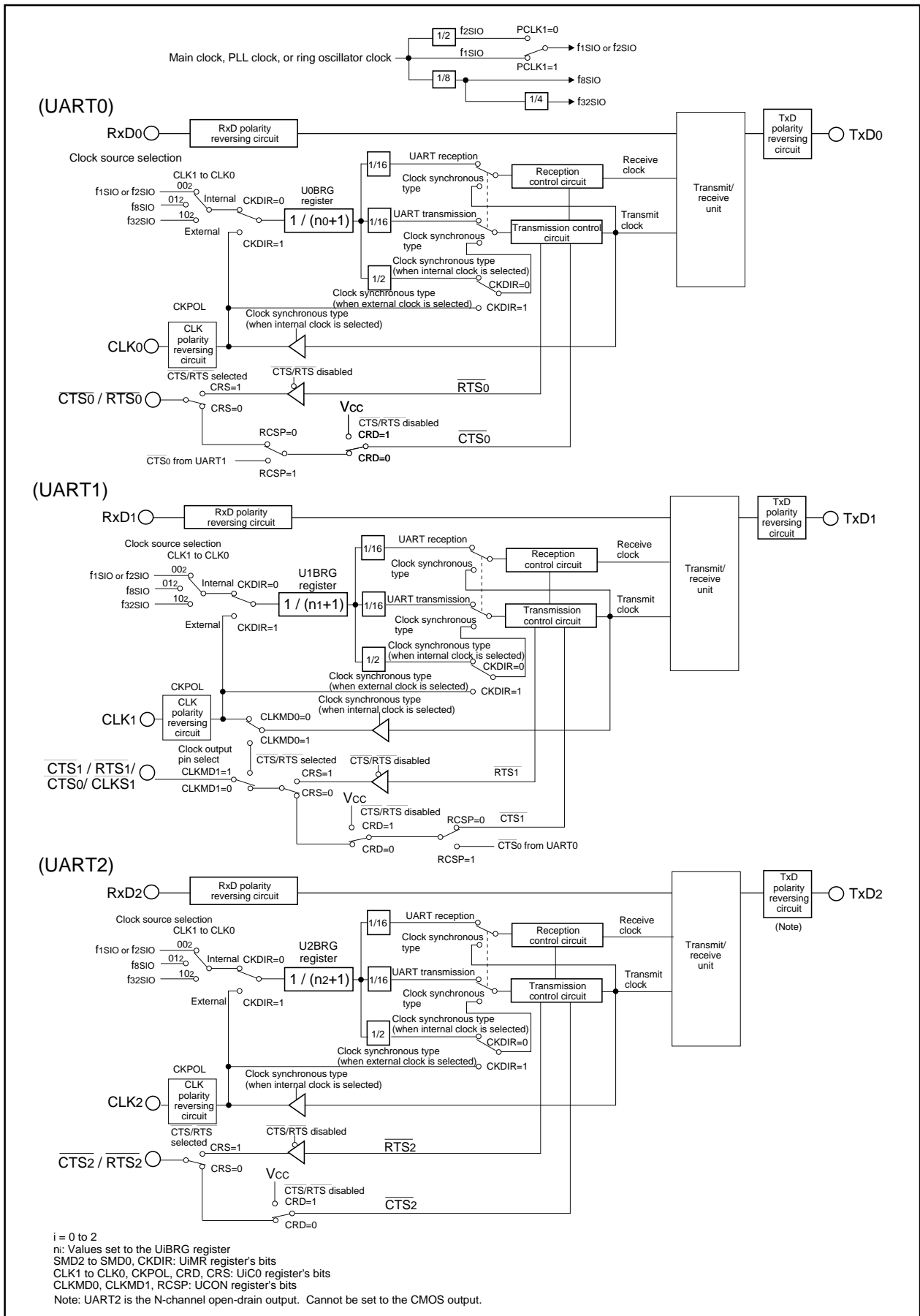


Figure 1.17.1. UARTi Block Diagram

Serial I/O

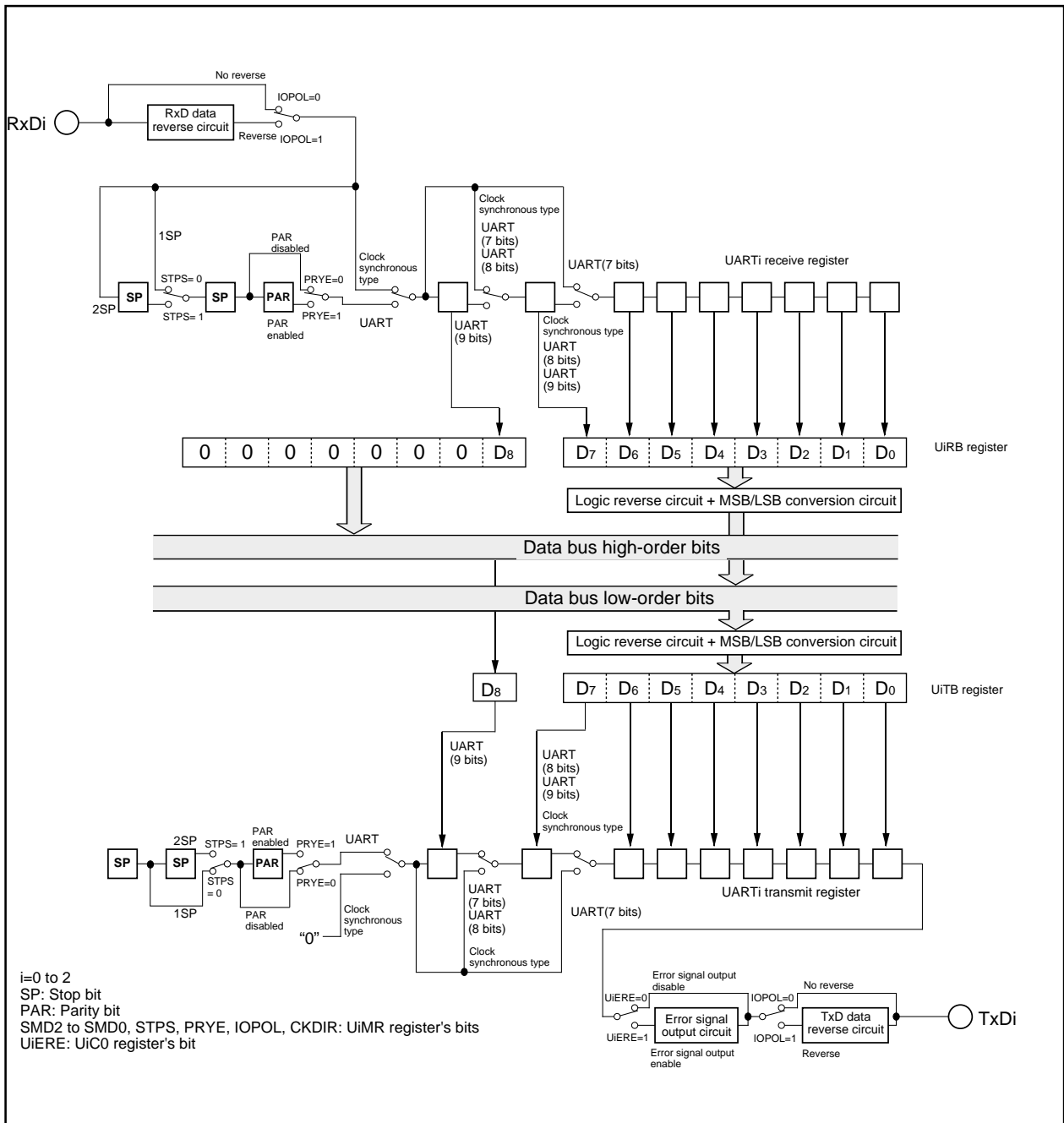


Figure 1.17.2. UARTi Transmit/Receive Unit

Serial I/O

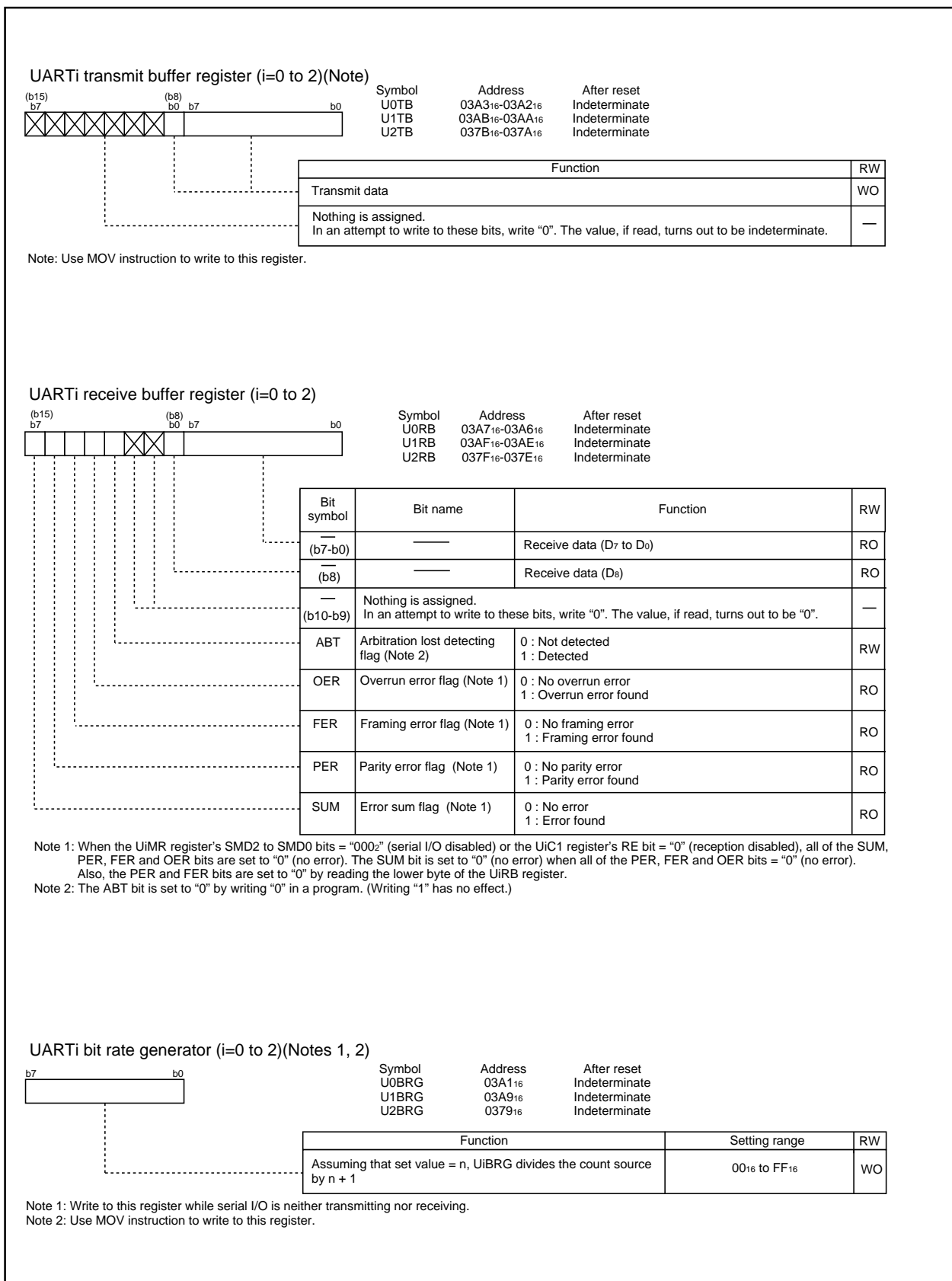


Figure 1.17.3. U0TB to U2TB Register, U0RB to U2RB Register, and U0BRG to U2BRG Register

Serial I/O

UART<sub>i</sub> transmit/receive mode register (i=0 to 2)

Bit symbol	Bit name	Function	RW
b7			
b6			
b5			
b4			
b3			
b2			
b1			
b0			
Symbol: U0MR to U2MR    Address: 03A0 <sub>16</sub> , 03A8 <sub>16</sub> , 0378 <sub>16</sub> After reset: 00 <sub>16</sub>			
SMD0	Serial I/O mode select bit (Note 2)	<sup>b2 b1 b0</sup> 0 0 0 : Serial I/O disabled 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : I <sup>2</sup> C mode (Note 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 Must not be set except above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 1)	RW
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	RW
PRY	Odd/even parity select bit	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

Note 1: Set the corresponding port direction bit for each CLK<sub>i</sub> pin to "0" (input mode).  
 Note 2: To receive data, set the corresponding port direction bit for each RxD<sub>i</sub> pin to "0" (input mode).  
 Note 3: Set the corresponding port direction bit for SCL and SDA pins to "0" (input mode).

UART<sub>i</sub> transmit/receive control register 0 (i=0 to 2)

Bit symbol	Bit name	Function	RW
b7			
b6			
b5			
b4			
b3			
b2			
b1			
b0			
Symbol: U0C0 to U2C0    Address: 03A4 <sub>16</sub> , 03AC <sub>16</sub> , 037C <sub>16</sub> After reset: 00001000 <sub>2</sub>			
CLK0	BRG count source select bit	<sup>b1 b0</sup> 0 0 : f <sub>1SIO</sub> or f <sub>2SIO</sub> is selected 0 1 : f <sub>8SIO</sub> is selected 1 0 : f <sub>32SIO</sub> is selected 1 1 : Must not be set	RW
CLK1			RW
CRS	CTS/RTS function select bit (Note 4)	Effective when CRD = 0 0 : CTS function is selected (Note 1) 1 : 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	CTS/RTS disable bit	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled (P6 <sub>0</sub> , P6 <sub>4</sub> and P7 <sub>3</sub> can be used as I/O ports)	RW
NCH	Data output select bit (Note 2)	0 : TxD <sub>i</sub> /SDA <sub>i</sub> and SCL <sub>i</sub> pins are CMOS output 1 : TxD <sub>i</sub> /SDA <sub>i</sub> and SCL <sub>i</sub> 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 (Note 3)	0 : LSB first 1 : MSB first	RW

Note 1: Set the corresponding port direction bit for each CTS<sub>i</sub> pin to "0" (input mode).  
 Note 2: TxD<sub>2</sub>/SDA<sub>2</sub> and SCL<sub>2</sub> are N-channel open-drain output. Cannot be set to the CMOS output. Set the NCH bit of the U2C0 register to "0".  
 Note 3: Effective for clock synchronous serial I/O mode and UART mode transfer data 8 bits long.  
 Note 4: CTS<sub>1</sub>/RTS<sub>1</sub> can be used when the UCON register's CLKMD1 bit = "0" (only CLK<sub>1</sub> output) and the UCON register's RCSP bit = "0" (CTS<sub>0</sub>/RTS<sub>0</sub> not separated).

Figure 1.17.4. U0MR to U2MR Register and U0C0 to U2C0 Register

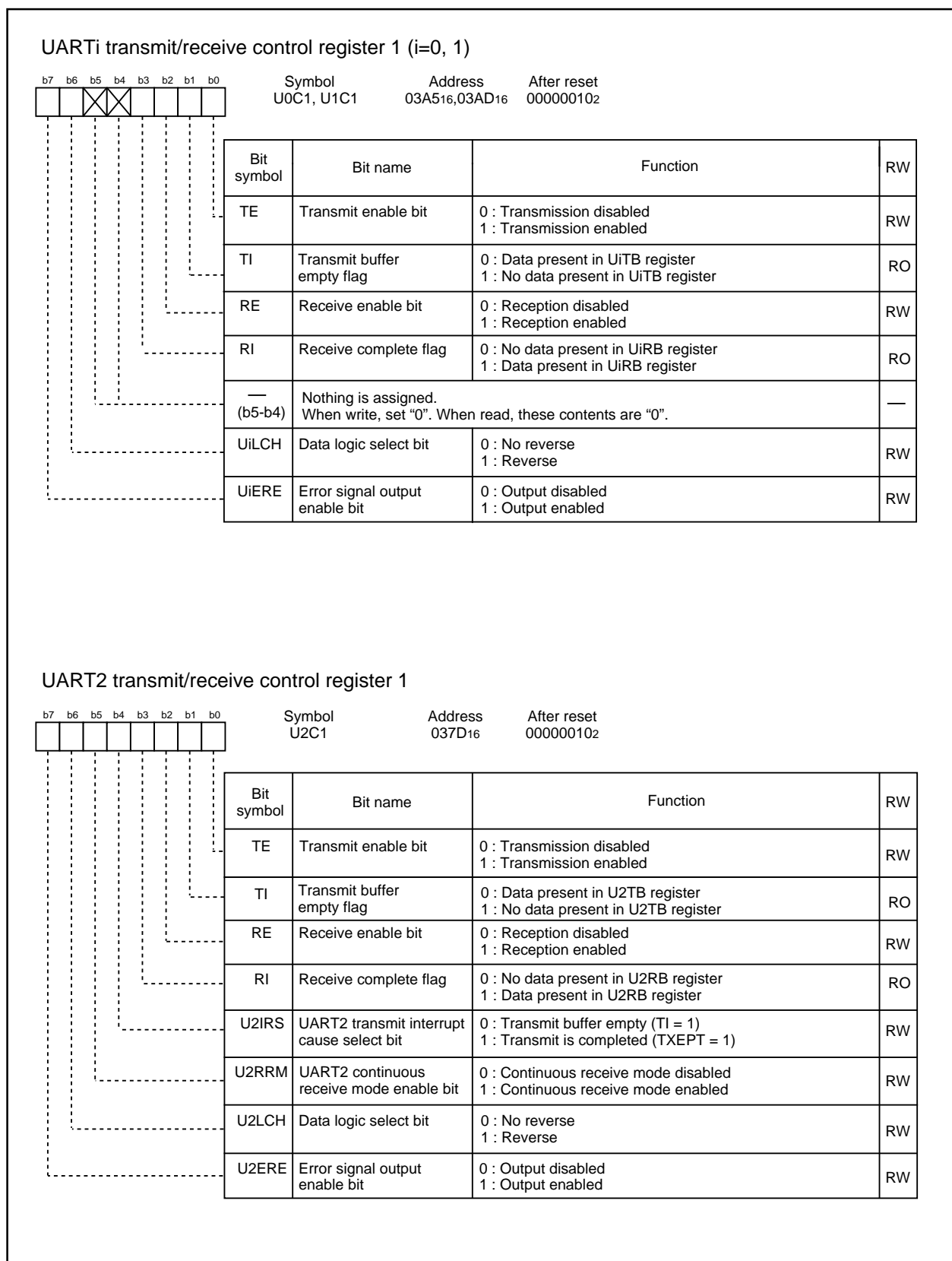


Figure 1.17.5. U0C1 to U2C1 Registers



Serial I/O

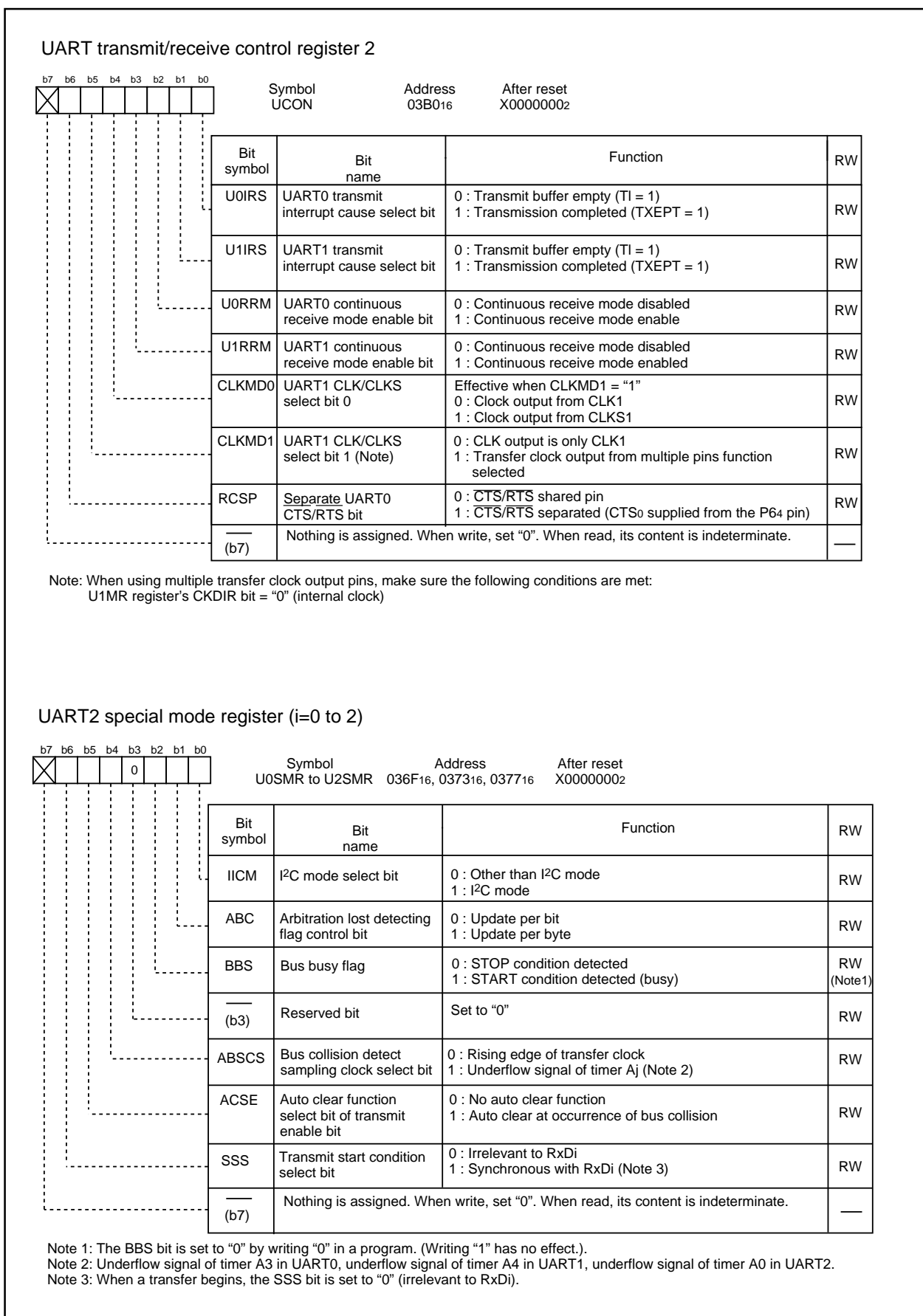


Figure 1.17.6. UCON Register and U0SMR to U2SMR Registers

Serial I/O

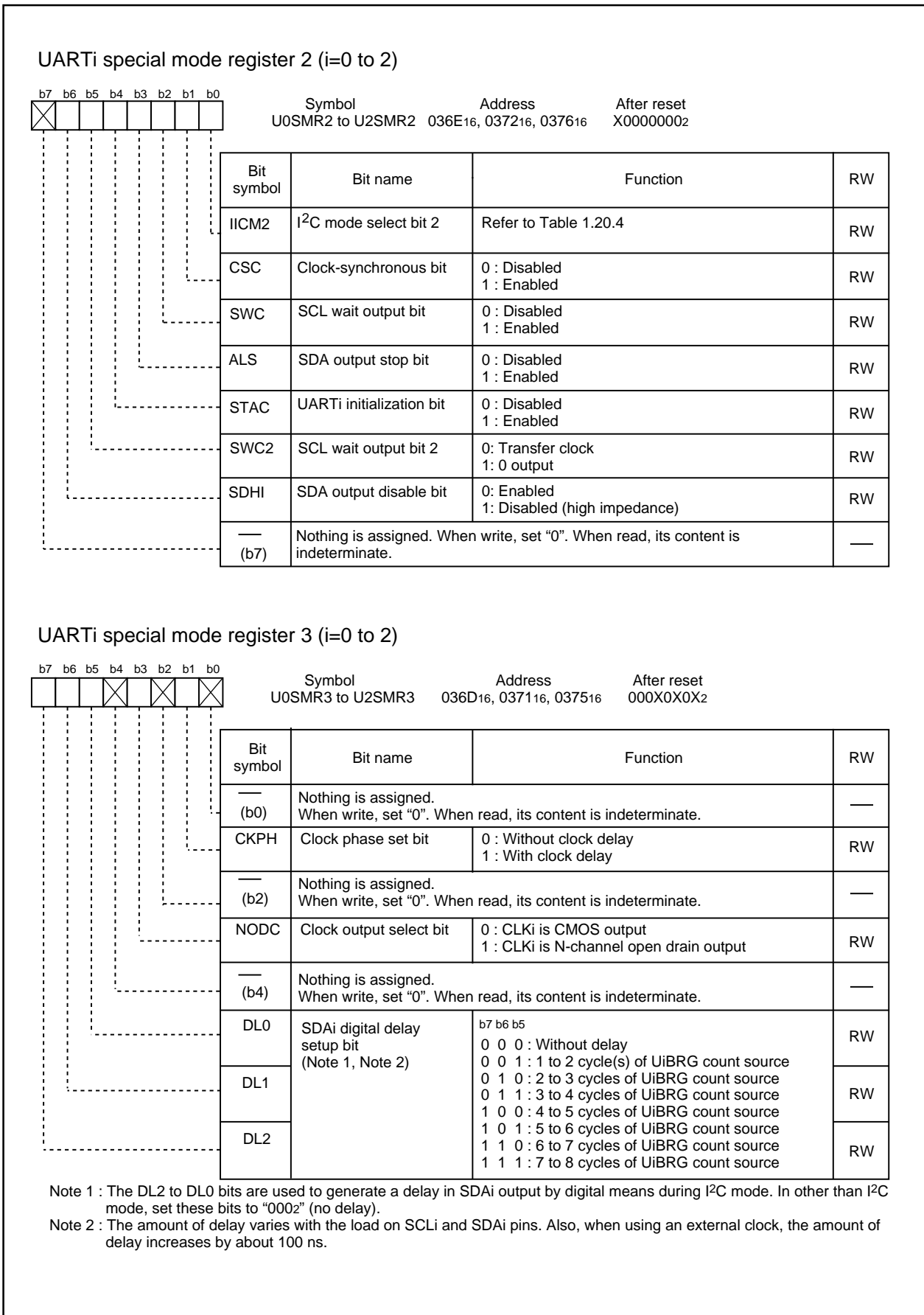


Figure 1.17.7. U0SMR2 to U2SMR2 Registers and U0SMR3 to U2SMR3 Registers

Serial I/O

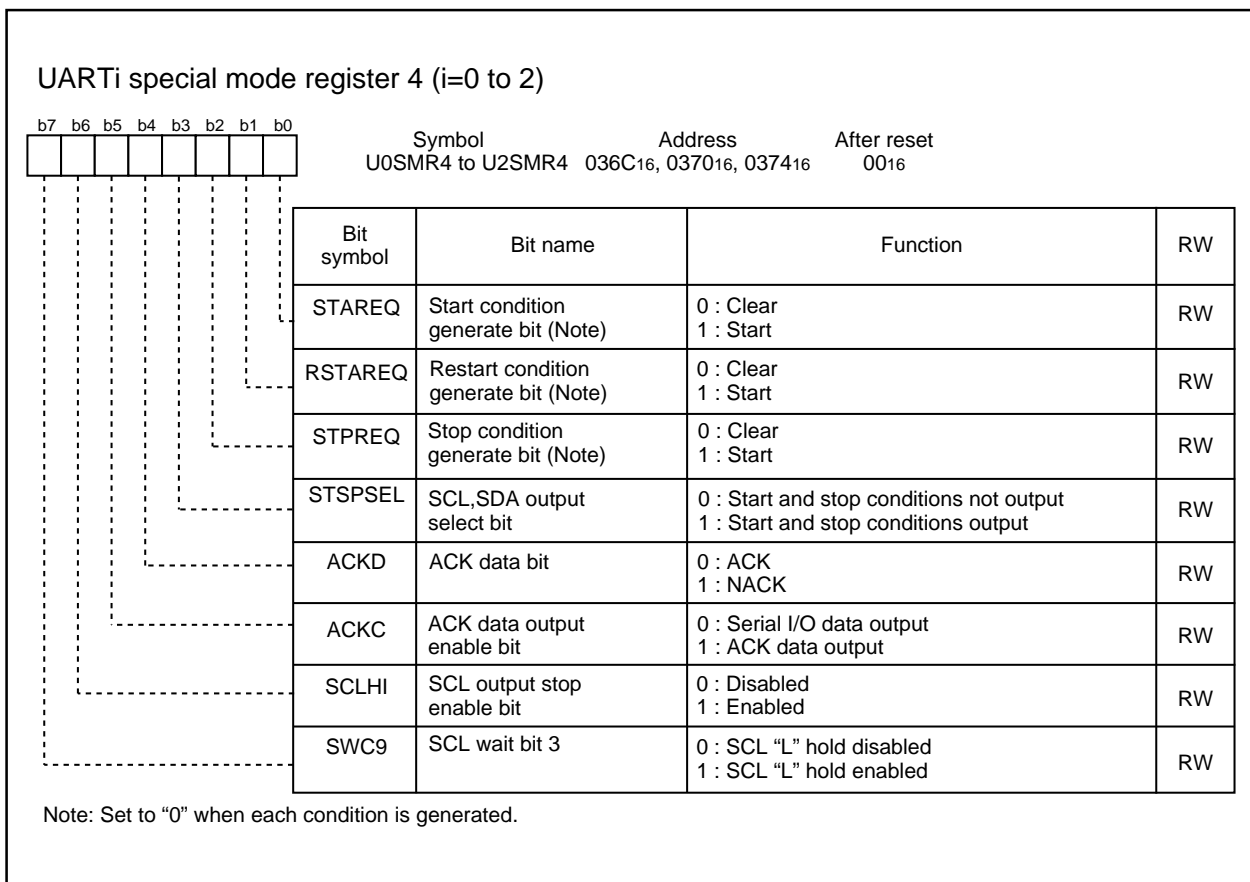


Figure 1.17.8. U0SMR4 to U2SMR4 Registers

## Serial I/O (Clock Synchronous Serial I/O)

**Clock Synchronous serial I/O Mode**

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 1.18.1 lists the specifications of the clock synchronous serial I/O mode. Table 1.18.2 lists the registers used in clock synchronous serial I/O mode and the register values set.

**Table 1.18.1. Clock Synchronous Serial I/O Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>UiMR(i=0 to 2) register's CKDIR bit = "0" (internal clock) : <math>f_j / 2(n+1)</math>  <math>f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math>. n: Setting value of UiBRG register 0016 to FF16</li> <li>CKDIR bit = "1" (external clock) : Input from CLKi pin</li> </ul>
Transmission, reception control	<ul style="list-style-type: none"> <li>Selectable from CTS function, RTS function or CTS/RTS function disable</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>Before transmission can start, the following requirements must be met (Note 1) <ul style="list-style-type: none"> <li>The TE bit of UiC1 register= 1 (transmission enabled)</li> <li>The TI bit of UiC1 register = 0 (data present in UiTB register)</li> <li>If CTS function is selected, input on the CTSi pin = "L"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>Before reception can start, the following requirements must be met (Note 1) <ul style="list-style-type: none"> <li>The RE bit of UiC1 register= 1 (reception enabled)</li> <li>The TE bit of UiC1 register= 1 (transmission enabled)</li> <li>The TI bit of UiC1 register= 0 (data present in the UiTB register)</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> <li>The UiIRS bit (Note 3) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)</li> <li>The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register</li> </ul> </li> <li>For reception <ul style="list-style-type: none"> <li>When transferring data from the UARTi receive register to the UiRB register (at completion of reception)</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error (Note 2) <ul style="list-style-type: none"> <li>This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 7th bit of the next data</li> </ul> </li> </ul>
Select function	<ul style="list-style-type: none"> <li>CLK polarity selection <ul style="list-style-type: none"> <li>Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock</li> </ul> </li> <li>LSB first, MSB first selection <ul style="list-style-type: none"> <li>Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> </ul> </li> <li>Continuous receive mode selection <ul style="list-style-type: none"> <li>Reception is enabled immediately by reading the UiRB register</li> </ul> </li> <li>Switching serial data logic <ul style="list-style-type: none"> <li>This function reverses the logic value of the transmit/receive data</li> </ul> </li> <li>Transfer clock output from multiple pins selection (UART1) <ul style="list-style-type: none"> <li>The output pin can be selected in a program from two UART1 transfer clock pins that have been set</li> </ul> </li> <li>Separate CTS/RTS pins (UART0) <ul style="list-style-type: none"> <li>CTS0 and RTS0 are input/output from separate pins</li> </ul> </li> </ul>

Note 1: When an external clock is selected, the conditions must be met while if the UiC0 register's CKPOL bit = "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the UiC0 register's 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.

Note 2: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.

Note 3: The U0IRS and U1IRS bits respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

## Serial I/O (Clock Synchronous Serial I/O)

Table 1. 18. 2. Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode

Register	Bit	Function
UiTB(Note3)	0 to 7	Set transmission data
UiRB(Note3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a transfer rate
UiMR(Note3)	SMD2 to SMD0	Set to "0012"
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to "0"
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select CTS or RTS to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the CTS or RTS function
	NCH	Select TxDi pin output mode (Note 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 (Note 1)	Select the source of UART2 transmit interrupt
	U2RRM (Note 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 UART0 CTS <sub>0</sub> signal from the P64 pin
	7	Set to "0"

Note 1: Set the U0C1 and U1C1 register bit 4 and bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.

Note 2: TxD2 pin is N channel open-drain output. Set the U2C0 register's NCH bit to "0".

Note 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

## Serial I/O (Clock Synchronous Serial I/O)

Table 1.18.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Table 1.18.3 shows pin functions for the case where the multiple transfer clock output pin select function is deselected. Table 1.18.4 lists the P64 pin functions during clock synchronous serial I/O mode. Note that for a period from when the UART<sub>i</sub> 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 1.18.3. Pin Functions (When Not Select Multiple Transfer Clock Output Pin Function)**

Pin name	Function	Method of selection
TxD <sub>i</sub> (i = 0 to 2) (P63, P67, P70)	Serial data output	(Outputs dummy data when performing reception only)
RxD <sub>i</sub> (P62, P66, P71)	Serial data input	PD6 register's PD6_2 bit=0, PD6_6 bit=0, PD7 register's PD7_1 bit=0 (Can be used as an input port when performing transmission only)
CLK <sub>i</sub> (P61, P65, P72)	Transfer clock output	UiMR register's CKDIR bit=0
	Transfer clock input	UiMR register's CKDIR bit=1 PD6 register's PD6_1 bit=0, PD6_5 bit=0, PD7 register's PD7_2 bit=0
CTS <sub>i</sub> /RTS <sub>i</sub> (P60, P64, P73)	CTS input	UiC0 register's CRD bit=0 UiC0 register's CRS bit=0 PD6 register's PD6_0 bit=0, PD6_4 bit=0, PD7 register's PD7_3 bit=0
	RTS output	UiC0 register's CRD bit=0 UiC0 register's CRS bit=1
	I/O port	UiC0 register's CRD bit=1

**Table 1.18.4. P64 Pin Functions**

Pin function	Bit set value					
	U1C0 register		UCON register			PD6 register
	CRD	CRS	RCSP	CLKMD1	CLKMD0	PD6_4
P64	1	—	0	0	—	Input: 0, Output: 1
CTS <sub>1</sub>	0	0	0	0	—	0
RTS <sub>1</sub>	0	1	0	0	—	—
CTS <sub>0</sub> (Note1)	0	0	1	0	—	0
CLKS <sub>1</sub>	—	—	—	1(Note 2)	1	—

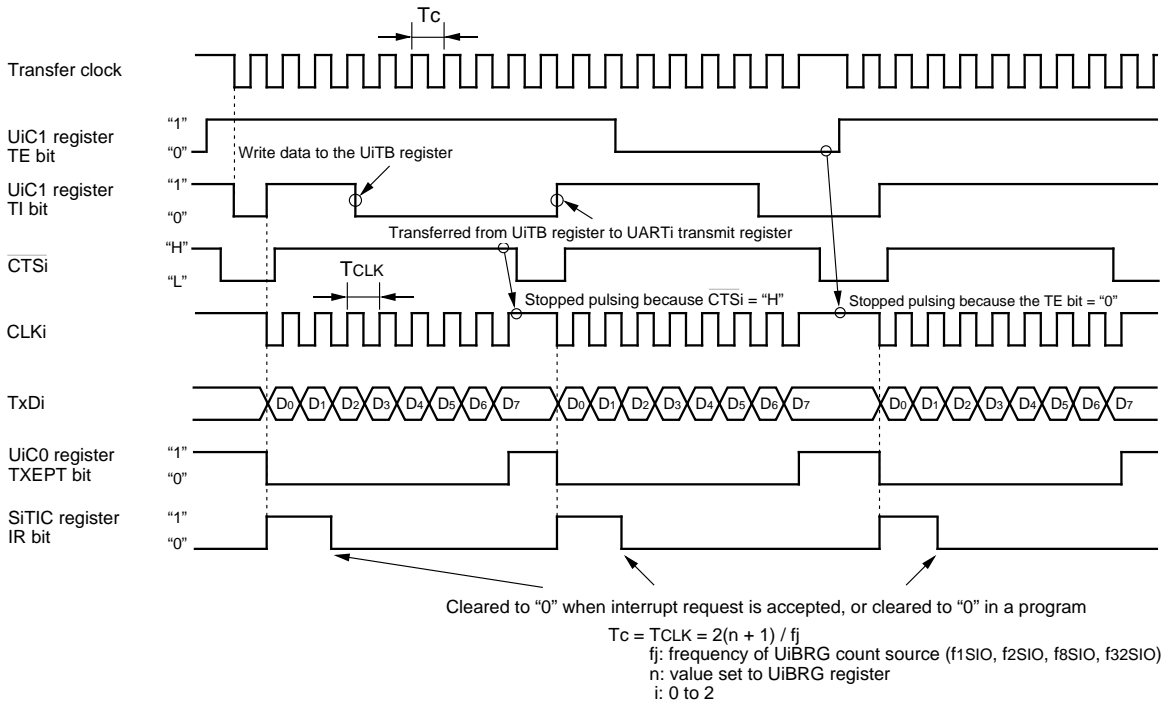
Note 1: In addition to this, set the U0C0 register's CRD bit to "0" (CTS<sub>0</sub>/RTS<sub>0</sub> enabled) and the U0C0 register's CRS bit to "1" (RTS<sub>0</sub> selected).

Note 2: When the CLKMD1 bit = 1 and the CLKMD0 bit = 0, the following logic levels are output:

- High if the U1C0 register's CLKPOL bit = 0
- Low if the U1C0 register's CLKPOL bit = 1

**Serial I/O (Clock Synchronous Serial I/O)**

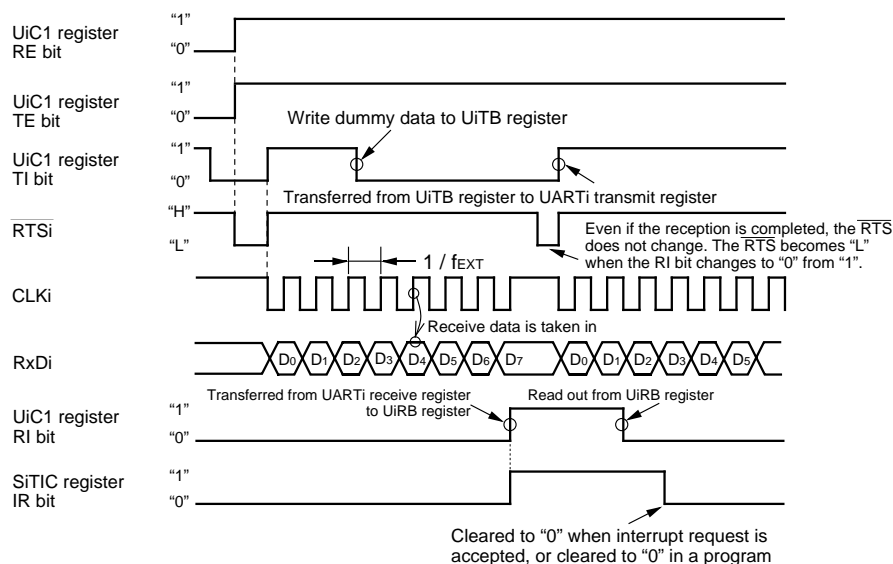
(1) Example of transmit timing (when internal clock is selected)



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

- UiMR register CKDIR bit = 0 (internal clock)
- UIC0 register CRD bit = 0 (CTS/RTS enabled), CRS bit = 0 (CTS selected)
- UIC0 register CKPOL bit = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- UIRS bit = 0 (an interrupt request occurs when the transmit buffer becomes empty): UIORS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

(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:

- UiMR register CKDIR bit = 1 (external clock)
- UIC0 register CRD bit = 0 (CTS/RTS enabled), CRS bit = 1 (RTS selected)
- UIC0 register CKPOL bit = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

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

- UIC0 register TE bit = 1 (transmit enabled)
- UIC0 register RE bit = 1 (Receive enabled)
- Write dummy data to the UiTB register

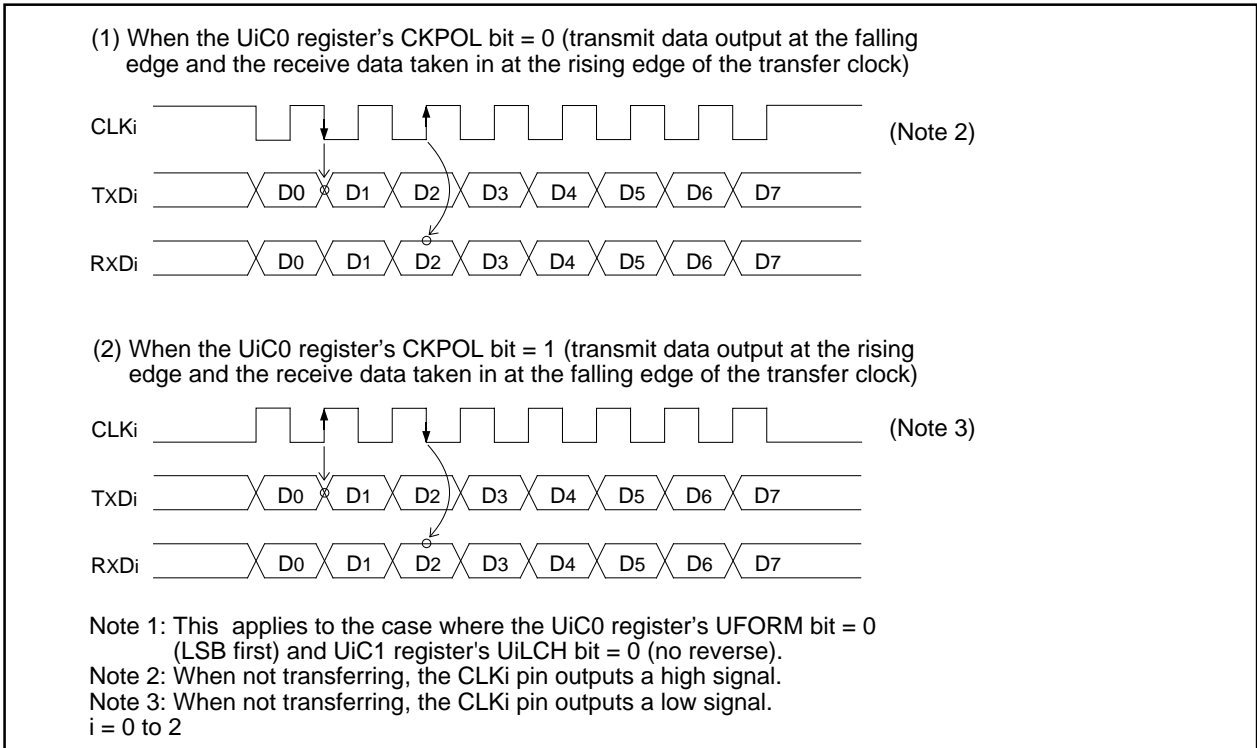
fEXT: frequency of external clock

**Figure 1.18.1. Transmit and Receive Operation**

**Serial I/O (Clock Synchronous Serial I/O)**

**(a) CLK Polarity Select Function**

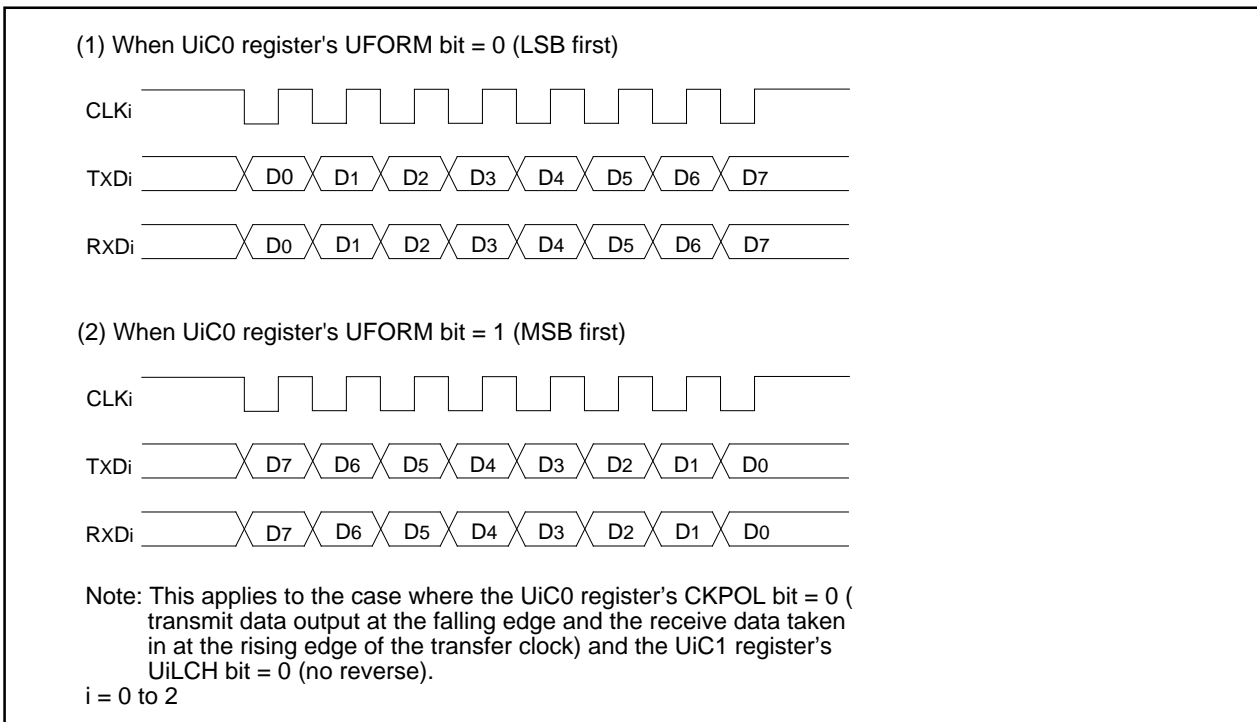
Use the UiC0 register (i = 0 to 2)'s CKPOL bit to select the transfer clock polarity. Figure 1.18.2 shows the polarity of the transfer clock.



**Figure 1.18.2. Transfer Clock Polarity**

**(b) LSB First/MSB First Select Function**

Use the UiC0 register (i = 0 to 2)'s UFORM bit to select the transfer format. Figure 1.18.3 shows the transfer format.



**Figure 1.18.3. Transfer Format**



Serial I/O (Clock Synchronous Serial I/O)

**(c) Continuous Receive Mode**

When the UiRRM bit ( $i = 0$  to  $2$ ) = 1 (continuous receive mode), the UiC1 register's TI bit is set to "1" (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 UCON register bit 2 and bit 3, respectively, and the U2RRM bit is the U2C1 register bit 4.

**(d) Serial Data Logic Switching Function**

When the UiC1 register ( $i = 0$  to  $2$ )'s UiLCH bit = 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 1.18.4 shows serial data logic.

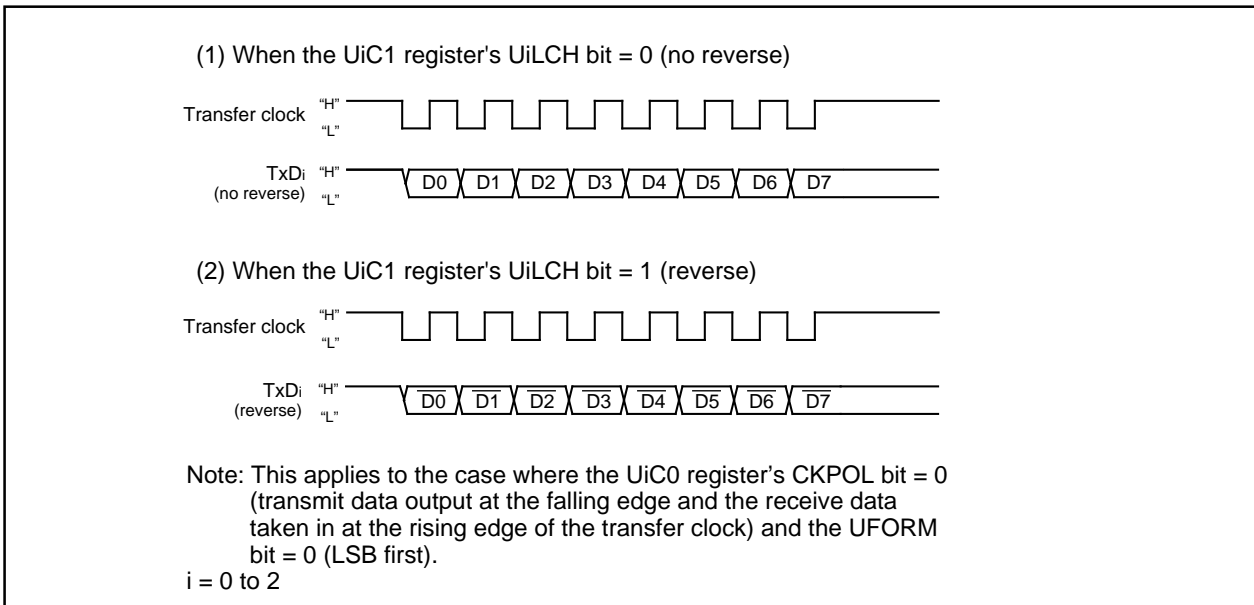


Figure 1.18.4. Serial Data Logic Switching

**(e) Transfer Clock Output From Multiple Pins (UART1)**

Use the UCON register's CLKMD1 to CLKMD0 bits to select one of the two transfer clock output pins. (See Figure 1.18.5.) This function can be used when the selected transfer clock for UART1 is an internal clock.

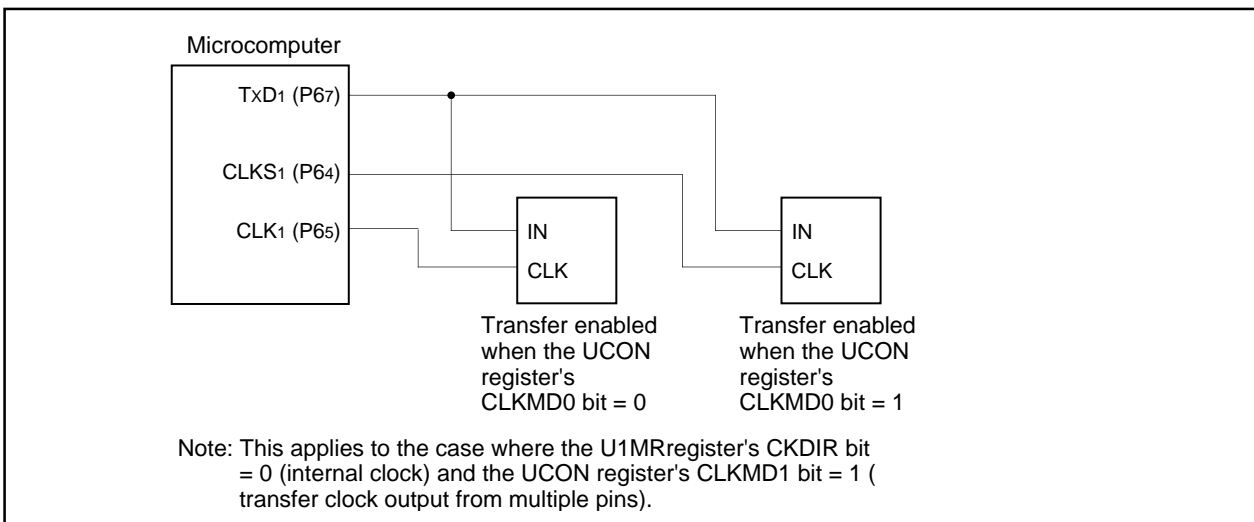


Figure 1.18.5. Transfer Clock Output From Multiple Pins

## Serial I/O (Clock Synchronous Serial I/O)

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

This function separates  $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$ , outputs  $\overline{\text{RTS}}_0$  from the P60 pin, and accepts as input the  $\overline{\text{CTS}}_0$  from the P64 pin. To use this function, set the register bits as shown below.

- U0C0 register's CRD bit = 0 (enables UART0  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- U0C0 register's CRS bit = 1 (outputs UART0  $\overline{\text{RTS}}$ )
- U1C0 register's CRD bit = 0 (enables UART1  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- U1C0 register's CRS bit = 0 (inputs UART1  $\overline{\text{CTS}}$ )
- UCON register's RCSP bit = 1 (inputs  $\overline{\text{CTS}}_0$  from the P64 pin)
- UCON register's CLKMD1 bit = 0 (CLKS1 not used)

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

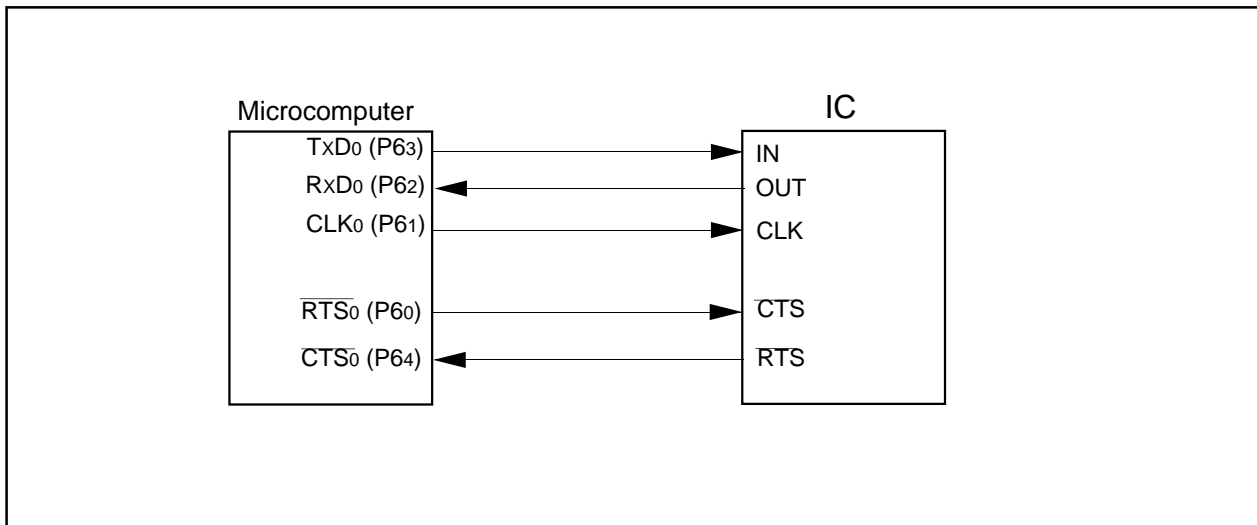


Figure 1.18.6.  $\overline{\text{CTS}}/\overline{\text{RTS}}$  Separat Function

## Serial I/O (UART)

**Clock Asynchronous Serial I/O (UART) Mode**

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables 1.19.1 lists the specifications of the UART mode.

**Table 1.19.1. UART Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Character bit (transfer data): Selectable from 7, 8 or 9 bits</li> <li>• Start bit: 1 bit</li> <li>• Parity bit: Selectable from odd, even, or none</li> <li>• Stop bit: Selectable from 1 or 2 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• UiMR(i=0 to 2) register's CKDIR bit = 0 (internal clock) : <math>f_j / 16(n+1)</math>  <math>f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math>. n: Setting value of UiBRG register 00<sub>16</sub> to FF<sub>16</sub></li> <li>• CKDIR bit = "1" (external clock) : <math>f_{EXT}/16(n+1)</math>  <math>f_{EXT}</math>: Input from CLKi pin. n: Setting value of UiBRG register 00<sub>16</sub> to FF<sub>16</sub></li> </ul>
Transmission, reception control	<ul style="list-style-type: none"> <li>• Selectable from <math>\overline{CTS}</math> function, <math>\overline{RTS}</math> function or <math>\overline{CTS}/\overline{RTS}</math> function disable</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>• Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> <li>– The TE bit of UiC1 register= 1 (transmission enabled)</li> <li>– The TI bit of UiC1 register = 0 (data present in UiTB register)</li> <li>– If <math>\overline{CTS}</math> function is selected, input on the <math>\overline{CTS}_i</math> pin = "L"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>• Before reception can start, the following requirements must be met <ul style="list-style-type: none"> <li>– The RE bit of UiC1 register= 1 (reception enabled)</li> <li>– Start bit detection</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> <li>– The UiIRS bit (Note 2) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)</li> <li>– The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register</li> </ul> </li> <li>• For reception <ul style="list-style-type: none"> <li>When transferring data from the UARTi receive register to the UiRB register (at completion of reception)</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error (Note 1) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the bit one before the last stop bit of the next data</li> <li>• Framing error This error occurs when the number of stop bits set is not detected</li> <li>• Parity error This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set</li> <li>• Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered</li> </ul>
Select function	<ul style="list-style-type: none"> <li>• LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> <li>• Serial data logic switch This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed.</li> <li>• 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.</li> <li>• Separate <math>\overline{CTS}/\overline{RTS}</math> pins (UART0) <math>\overline{CTS}_0</math> and <math>\overline{RTS}_0</math> are input/output from separate pins</li> </ul>

Note 1: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.

Note 2: The U0IRS and U1IRS bits respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

## Serial I/O (UART)

Table 1. 19. 2. Registers to Be Used and Settings in UART Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data (Note 1)
UiRB	0 to 8	Reception data can be read (Note 1)
	OER,FER,PER,SUM	Error flag
UiBRG	---	Set a transfer rate
UiMR	SMD2 to SMD0	Set these bits to '1002' when transfer data is 7 bits long Set these bits to '1012' when transfer data is 8 bits long Set these bits to '1102' when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
	IOPOL	Select the TxD/RxD input/output polarity
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	CRS	Select CTS or RTS to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the CTS or RTS function
	NCH	Select TxDi pin output mode (Note 2)
	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 (Note 2)	Select the source of UART2 transmit interrupt
	U2RRM (Note 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 the UART0 CTS <sub>0</sub> signal from the P64 pin
	7	Set to "0"

Note 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.

Note 2: Set the U0C1 and U1C1 registers bit 4 to bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are included in the UCON register.

Note 3: TxD2 pin is N channel open-drain output. Set the U2C0 register's NCH bit to "0".

i=0 to 2

## Serial I/O (UART)

Table 1.19.3 lists the functions of the input/output pins during UART mode. Table 1.19.4 lists the P64 pin functions during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an "H". (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

**Table 1.19.3. I/O Pin Functions**

Pin name	Function	Method of selection
TxDi (i = 0 to 2) (P63, P67, P70)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P62, P66, P71)	Serial data input	PD6 register's PD6_2 bit=0, PD6_6 bit=0, PD7 register's PD7_1 bit=0 (Can be used as an input port when performing transmission only)
CLKi (P61, P65, P72)	Input/output port	UiMR register's CKDIR bit=0
	Transfer clock input	UiMR register's CKDIR bit=1 PD6 register's PD6_1 bit=0, PD6_5 bit=0, PD7 register's PD7_2 bit=0
$\overline{\text{CTS}}_i/\text{RTS}_i$ (P60, P64, P73)	$\overline{\text{CTS}}$ input	UiC0 register's CRD bit=0 UiC0 register's CRS bit=0 PD6 register's PD6_0 bit=0, PD6_4 bit=0, PD7 register's PD7_3 bit=0
	RTS output	UiC0 register's CRD bit=0 UiC0 register's CRS bit=1
	Input/output port	UiC0 register's CRD bit=1

**Table 1.19.4. P64 Pin Functions**

Pin function	Bit set value				
	U1C0 register		UCON register		PD6 register
	CRD	CRS	RCSP	CLKMD1	PD6_4
P64	1	—	0	0	Input: 0, Output: 1
$\overline{\text{CTS}}_1$	0	0	0	0	0
$\text{RTS}_1$	0	1	0	0	—
$\overline{\text{CTS}}_0$ (Note)	0	0	1	0	0

Note: In addition to this, set the U0C0 register's CRD bit to "0" ( $\overline{\text{CTS}}_0/\text{RTS}_0$  enabled) and the U0C0 register's CRS bit to "1" ( $\text{RTS}_0$  selected).

Serial I/O (UART)

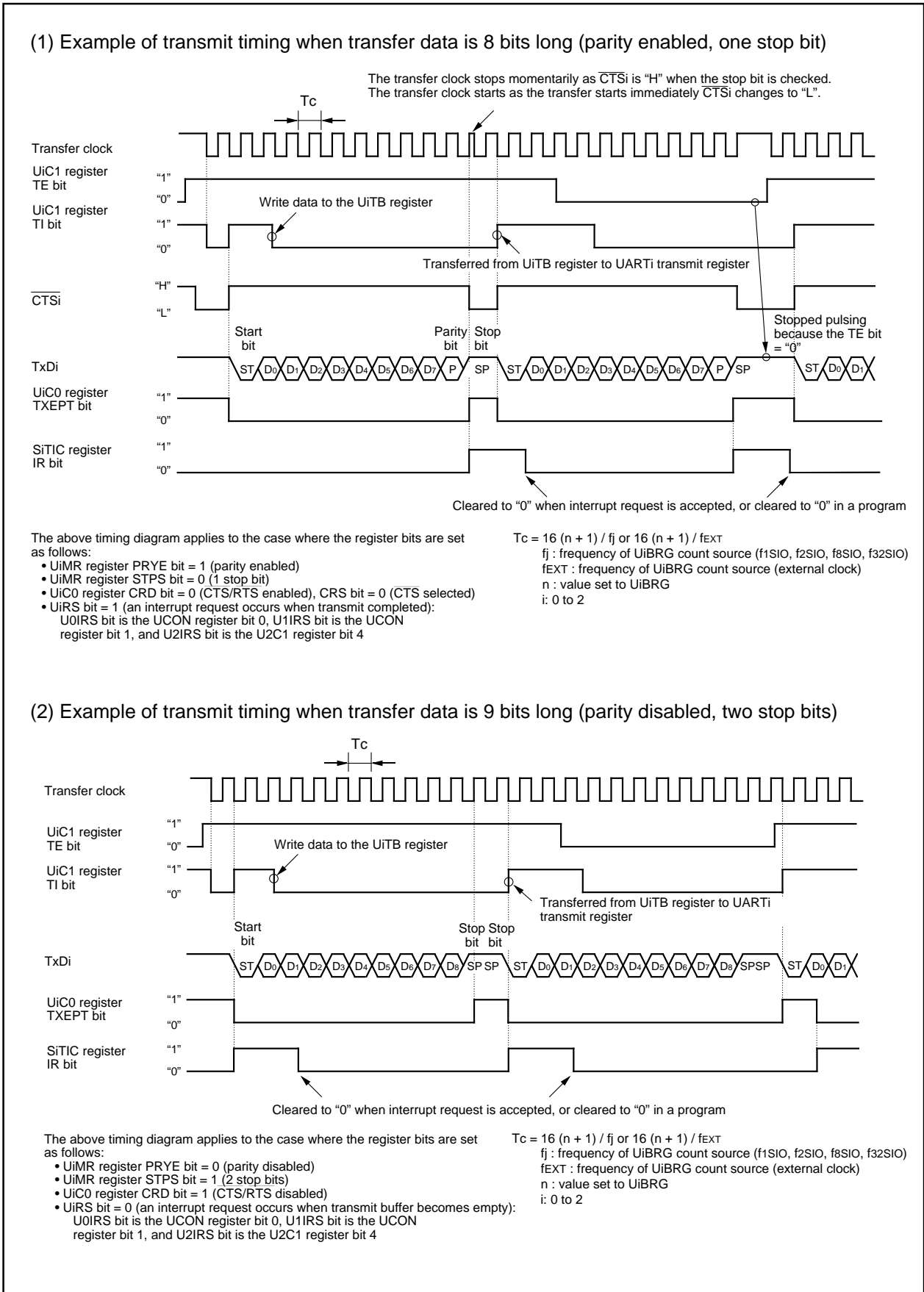
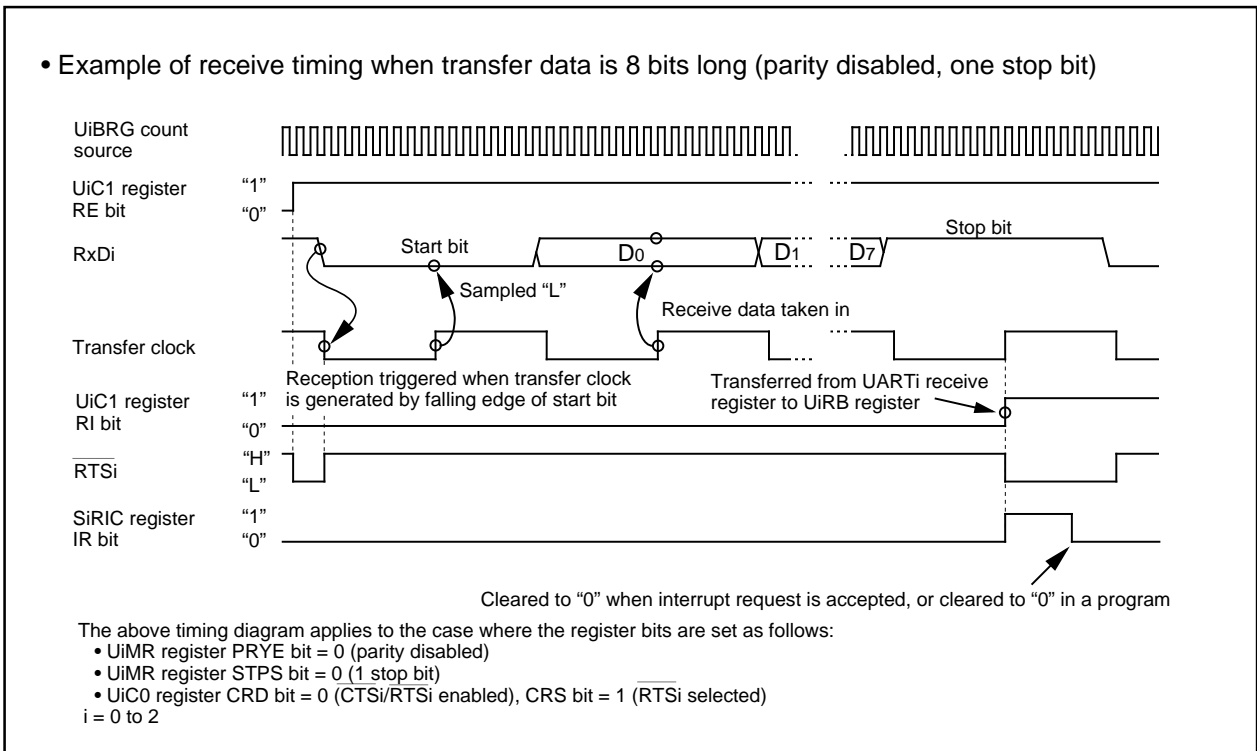


Figure 1.19.1. Transmit Operation

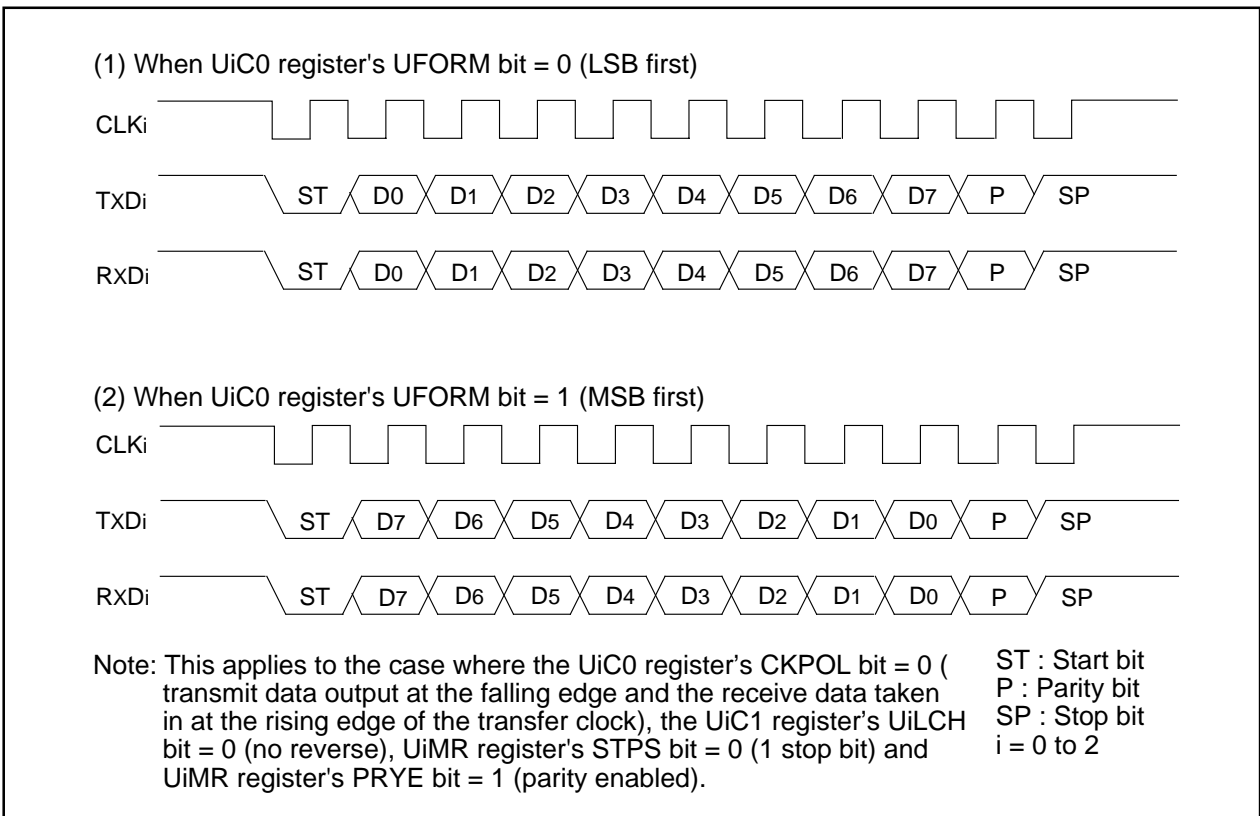
**Serial I/O (UART)**



**Figure 1.19.2. Receive Operation**

**(a) LSB First/MSB First Select Function**

As shown in Figure 1.19.3, use the UiC0 register's UFORM bit to select the transfer format. This function is valid when transfer data is 8 bits long.

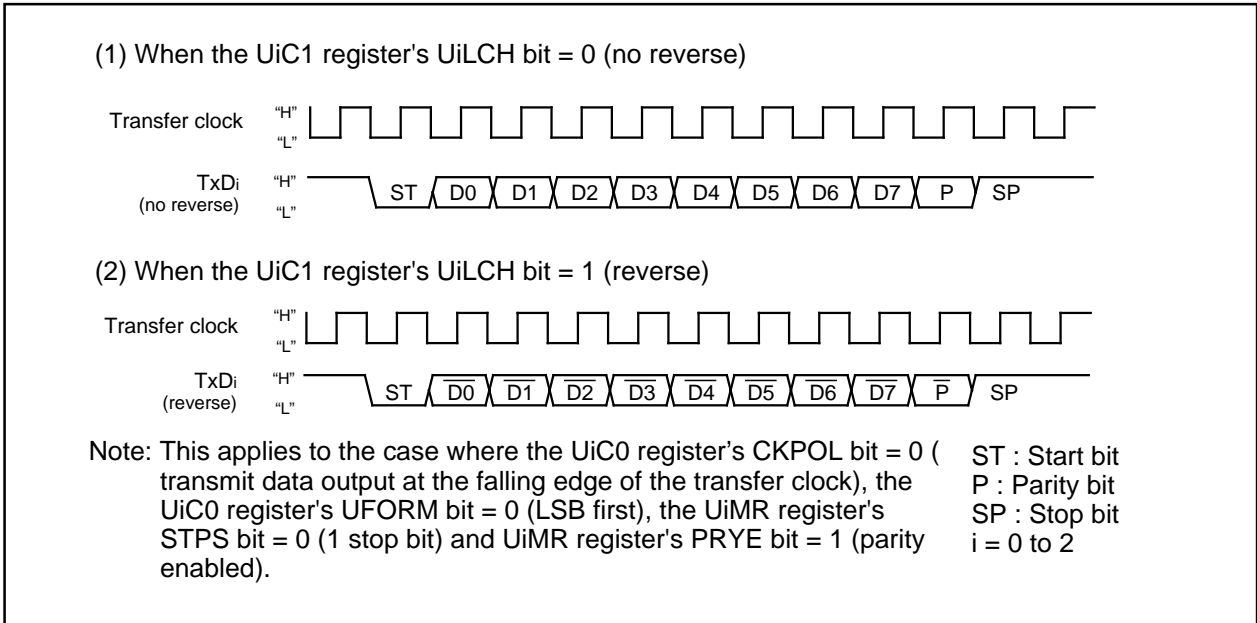


**Figure 1.19.3. Transfer Format**

**Serial I/O (UART)**

**(b) Serial Data Logic Switching Function**

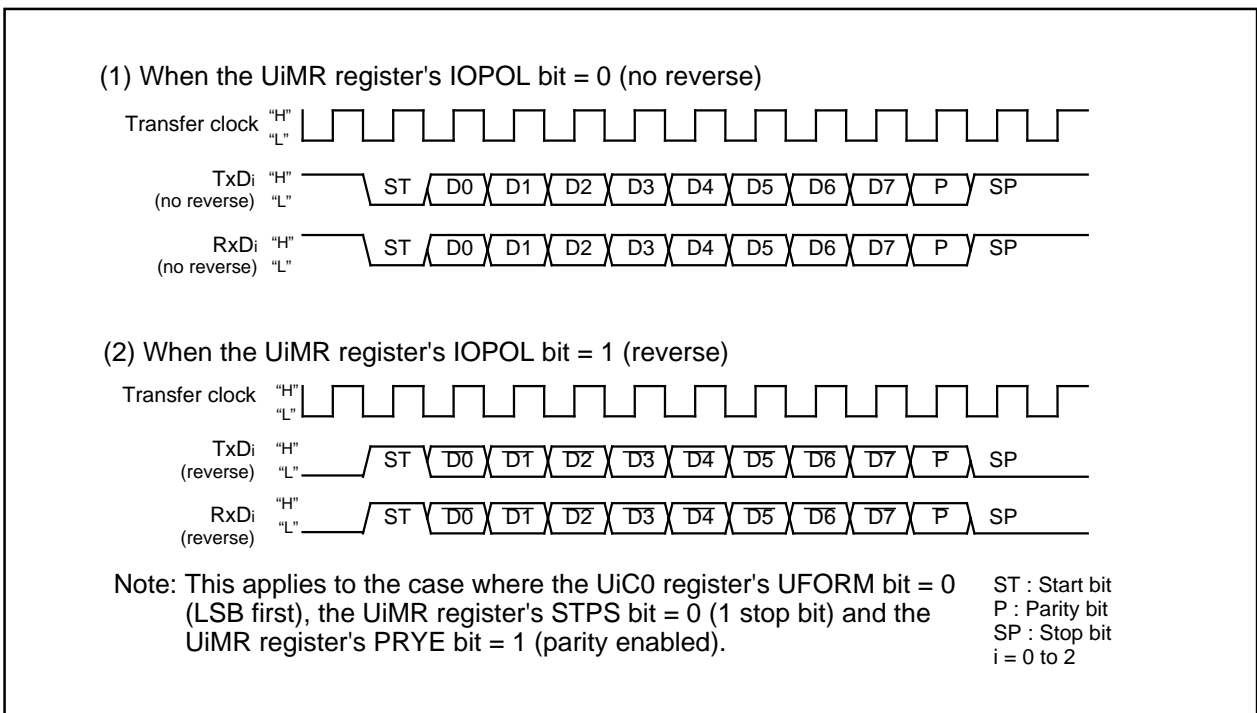
The data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 1.19.4 shows serial data logic.



**Figure 1.19.4. Serial Data Logic Switching**

**(c) TxD and RxD I/O Polarity Inverse Function**

This function inverses the polarities of the TxDi pin output and RxDi pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inverted. Figure 1.19.5 shows the TxD pin output and RxD pin input polarity inverse.



**Figure 1.19.5. TxD and RxD I/O Polarity Inverse**



## Serial I/O (UART)

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

This function separates  $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$ , outputs  $\overline{\text{RTS}}_0$  from the P60 pin, and accepts as input the  $\overline{\text{CTS}}_0$  from the P64 pin. To use this function, set the register bits as shown below.

- U0C0 register's CRD bit = 0 (enables UART0  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- U0C0 register's CRS bit = 1 (outputs UART0  $\overline{\text{RTS}}$ )
- U1C0 register's CRD bit = 0 (enables UART1  $\overline{\text{CTS}}/\overline{\text{RTS}}$ )
- U1C0 register's CRS bit = 0 (inputs UART1  $\overline{\text{CTS}}$ )
- UCON register's RCSP bit = 1 (inputs  $\overline{\text{CTS}}_0$  from the P64 pin)
- UCON register's CLKMD1 bit = 0 (CLKS1 not used)

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

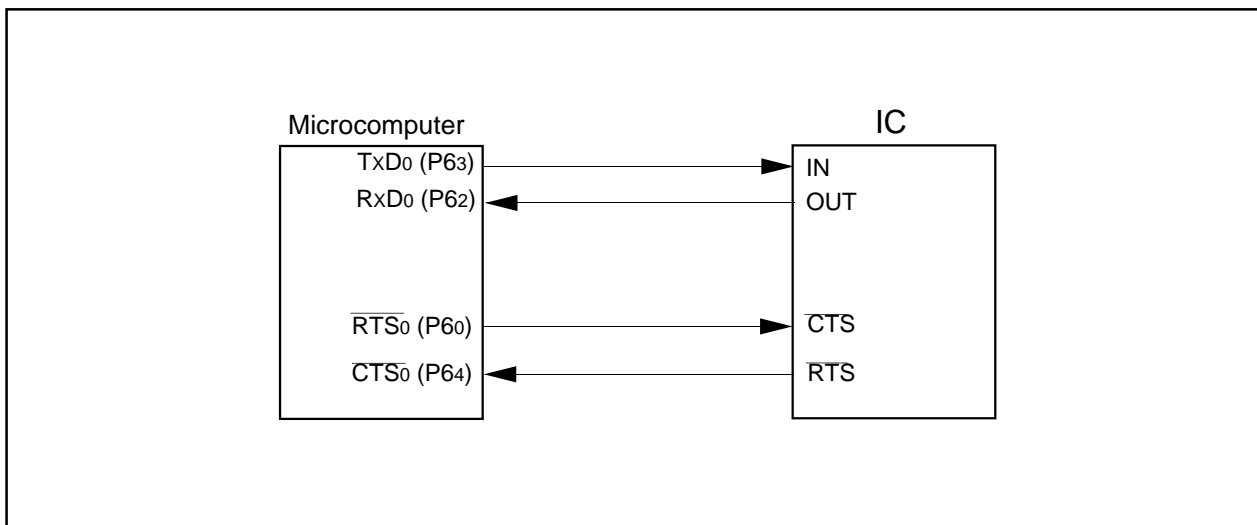


Figure 1.19.6.  $\overline{\text{CTS}}/\overline{\text{RTS}}$  Separate Function

## Serial I/O (Special Modes)

**Special Mode 1 (I<sup>2</sup>C mode)**

I<sup>2</sup>C mode is provided for use as a simplified I<sup>2</sup>C interface compatible mode. Table 1.20.1 lists the specifications of the I<sup>2</sup>C mode. Table 1.20.2 lists the registers used in the I<sup>2</sup>C mode and the register values set. Figure 1.20.1 shows the block diagram for I<sup>2</sup>C mode. Figure 1.20.2 shows SCLi timing.

As shown in Table 1.20.3, the microcomputer is placed in I<sup>2</sup>C mode by setting the SMD2 to SMD0 bits to '0102' 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.

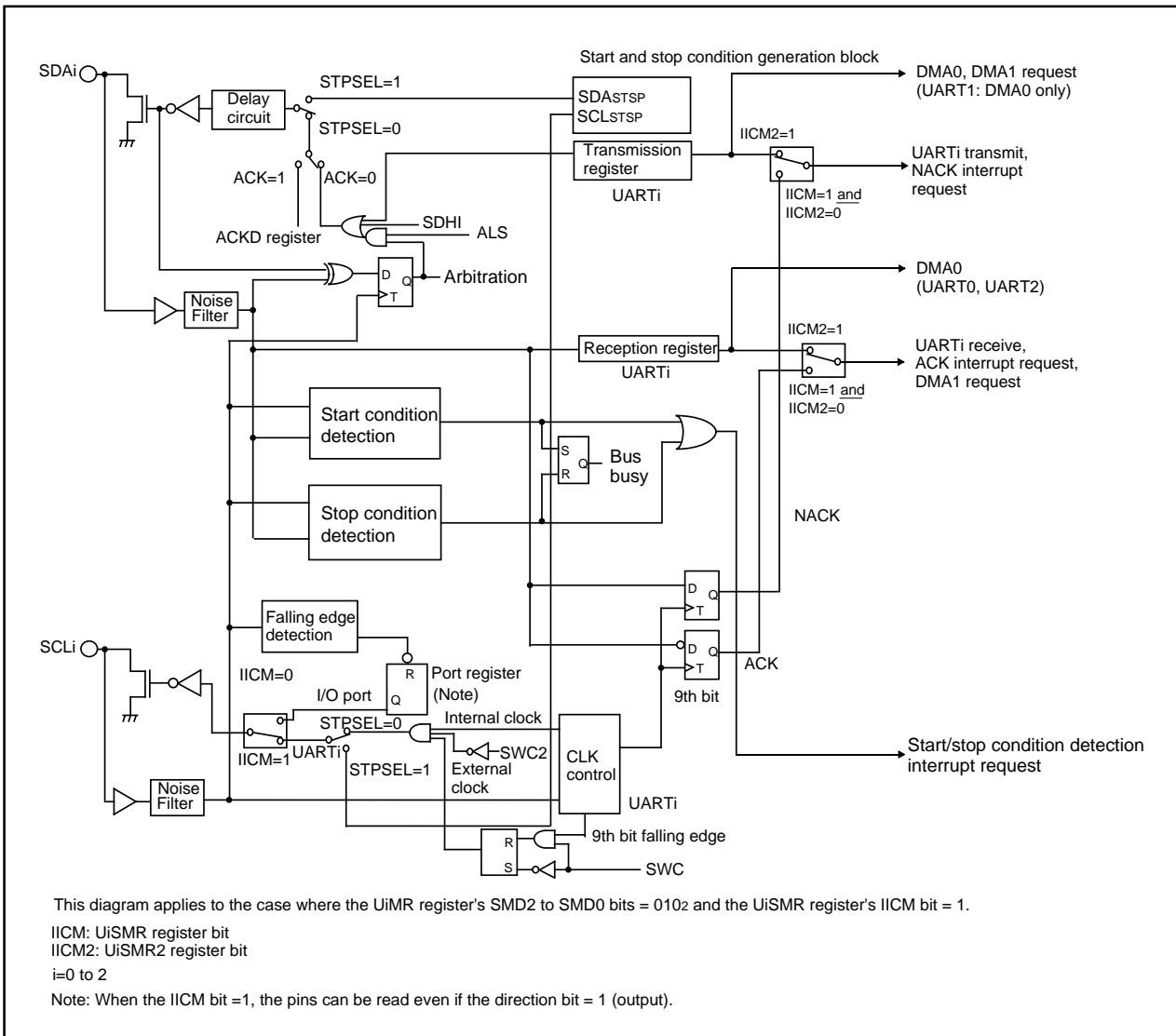
**Table 1.20.1. I<sup>2</sup>C Mode Specifications**

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• During master UiMR(i=0 to 2) register's CKDIR bit = "0" (internal clock) : $f_j / 2(n+1)$ $f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}$ . n: Setting value of UiBRG register 0016 to FF16 • During slave CKDIR bit = "1" (external clock) : Input from CLKi pin
Transmission start condition	• Before transmission can start, the following requirements must be met (Note 1) – The TE bit of UiC1 register= 1 (transmission enabled) – The TI bit of UiC1 register = 0 (data present in UiTB register)
Reception start condition	• Before reception can start, the following requirements must be met (Note 1) – The RE bit of UiC1 register= 1 (reception enabled) – The TE bit of UiC1 register= 1 (transmission enabled) – The TI bit of 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 (Note 2) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 8th bit of the next data
Select function	• Arbitration lost Timing at which the UiRB register's ABT bit 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

Note 1: When an external clock is selected, the conditions must be met while the external clock is in the high state.

Note 2: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.

**Serial I/O (Special Modes)**



**Figure 1.20.1. I²C Mode Block Diagram**

## Serial I/O (Special Modes)

Table 1. 20. 2. Registers to Be Used and Settings in I<sup>2</sup>C Mode (1) (Continued)

Register	Bit	Function	
		Master	Slave
UiTB <sup>3</sup>	0 to 7	Set transmission data	Set transmission data
UiRB <sup>3</sup>	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	---	Set a transfer rate	Invalid
UiMR <sup>3</sup>	SMD2 to SMD0	Set to '0102'	Set to '0102'
	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	Set to "1"	Set to "1"
	NCH	Set to "1" <sup>2</sup>	Set to "1" <sup>2</sup>
	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 <sup>1</sup>	Invalid	Invalid
	U2RRM <sup>1</sup> , 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	Refer to Table 1.20.4.	Refer to Table 1.20.4.
	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"
UiSMR3	0, 2, 4 and NODC	Set to "0"	Set to "0"
	CKPH	Refer to Table 1.20.4	Refer to Table 1.20.4
	DL2 to DL0	Set the amount of SDAi digital delay	Set the amount of SDAi digital delay

i=0 to 2

Notes:

1. Set the U0C1 and U1C1 register bit 4 and bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are 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 I<sup>2</sup>C mode.

## Serial I/O (Special Modes)

Table 1. 20. 3. Registers to Be Used and Settings in I<sup>2</sup>C Mode (2) (Continued)

Register	Bit	Function	
		Master	Slave
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
	2 to 7	Set to "0"	Set to "0"

i=0 to 2

## Serial I/O (Special Modes)

Table 1.20.4. I<sup>2</sup>C Mode Functions

Function	Clock synchronous serial I/O mode (SMD2 to SMD0 = 001 <sub>2</sub> , IICM = 0)	I <sup>2</sup> C mode (SMD2 to SMD0 = 010 <sub>2</sub> , 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 (Note 1)(Refer to Fig 1.20.2)	—————	Start condition detection or stop condition detection (Refer to Fig 1.20.4)			
Factor of interrupt number 15, 17 and 19 (Note 1)(Refer to Fig 1.20.2)	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 (Note 1)(Refer to Fig 1.20.2)	UARTi reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) Rising edge of SCLi 9th bit		UARTi transmission 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 P63, P67 and P70 pins	TxDi output	SDAi input/output			
Functions of P62, P66 and P71 pins	RxDi input	SCLi input/output			
Functions of P61, P65 and P72 pins	CLKi input or output selected	————— (Cannot be used in I <sup>2</sup> C 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 I <sup>2</sup> C mode (Note 2)			
Initial and end values of SCLi	—————	H	L	H	L
DMA1 factor (Refer to Fig 1.20.2)	UARTi reception	Acknowledgment detection (ACK)		UARTi reception Falling edge of SCLi 9th bit	
Store received data	1st to 8th bits are stored in UiRB register bit 0 to bit 7	1st to 8th bits are stored in UiRB register bit 7 to bit 0		1st to 7th bits are stored in UiRB register bit 6 to bit 0, with 8th bit stored in UiRB register bit 8  1st to 8th bits are stored in UiRB register bit 7 to bit 0 (Note 3)	
Read received data	UiRB register status is read directly as is				Read UiRB register Bit 6 to bit 0 as bit 7 to bit 1, and bit 8 as bit 0 (Note 4)

i = 0 to 2

Note 1: To change the interrupt sources from one to another, follow the procedure described below.

1. Disable the interrupt of the corresponding interrupt number to be changed.
2. Change interrupt sources from one to another.
3. Set the IR bit for the corresponding interrupt number to 0 (no interrupt request).
4. Set the IPL2 to IPL0 bits for the corresponding interrupt number.

Note 2: Set the initial value of SDAi output while the UiMR register's SMD2 to SMD0 bits = '000<sub>2</sub>' (serial I/O disabled).

Note 3: Second data transfer to UiRB register (Rising edge of SCLi 9th bit)

Note 4: First data transfer to UiRB register (Falling edge of SCLi 9th bit)

Serial I/O (Special Modes)

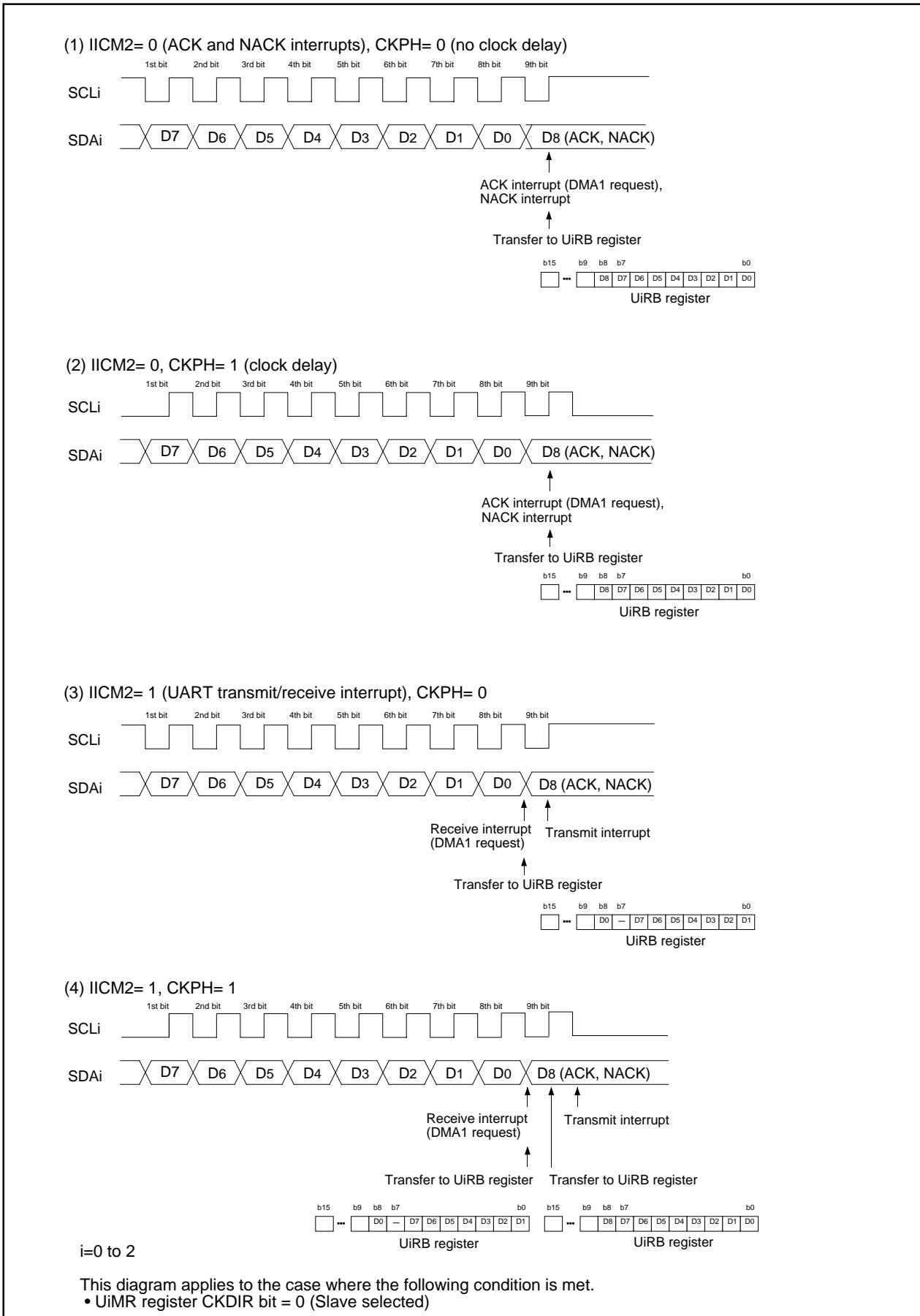


Figure 1.20.2. Transfer to UiRB Register and Interrupt Timing

## Serial I/O (Special Modes)

### • 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 SDA<sub>i</sub> pin changes state from high to low while the SCL<sub>i</sub> pin is in the high state. A stop condition-detected interrupt request is generated when the SDA<sub>i</sub> pin changes state from low to high while the SCL<sub>i</sub> pin is in the high state.

Because the start and stop condition-detected interrupts share the interrupt control register and vector, check the UiSMR register's BBS bit to determine which interrupt source is requesting the interrupt.

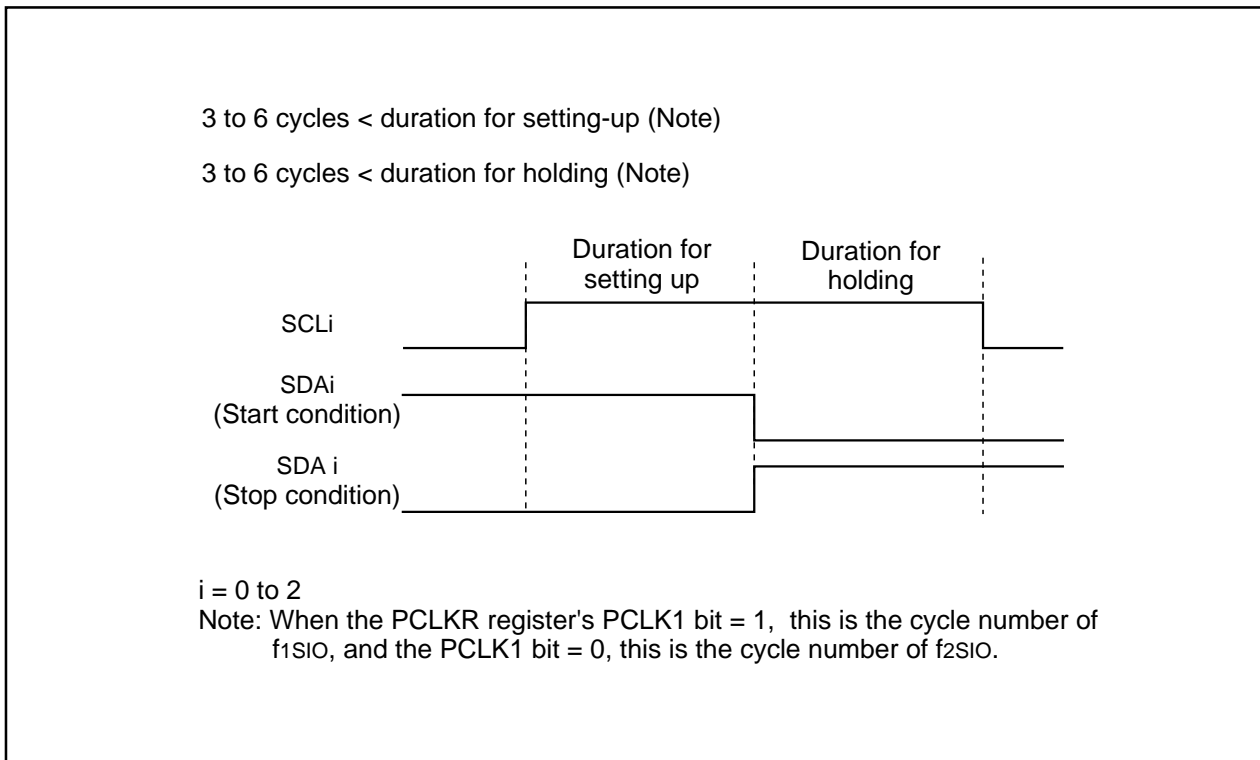


Figure 1.20.3. Detection of Start and Stop Condition

### • Output of Start and Stop Condition

A start condition is generated by setting the UiSMR4 register ( $i = 0$  to  $2$ )'s STAREQ bit to "1" (start).

A restart condition is generated by setting the UiSMR4 register's RSTAREQ bit to "1" (start).

A stop condition is generated by setting the UiSMR4 register's STPREQ bit to "1" (start).

A start condition is output by setting the STAREQ bit to "1" and then the UiSMR4 register's STSPSEL bit to "1" (start). Similarly, a restart condition is output by setting the RSTAREQ bit to "1" and then the STSPSEL bit to "1", and a stop condition is output by setting the STPREQ bit to "1" and then the STSPSEL bit to "1".

Table 1.20.5 and Figure 1.20.4 show the functions of the STSPSEL.

If start, stop and restart conditions are to be output, make sure no interrupts will occur between the instruction that sets the STAREQ, STPREQ or RSTAREQ bit to "1" and the instruction that sets the STSPSEL bit to "1".

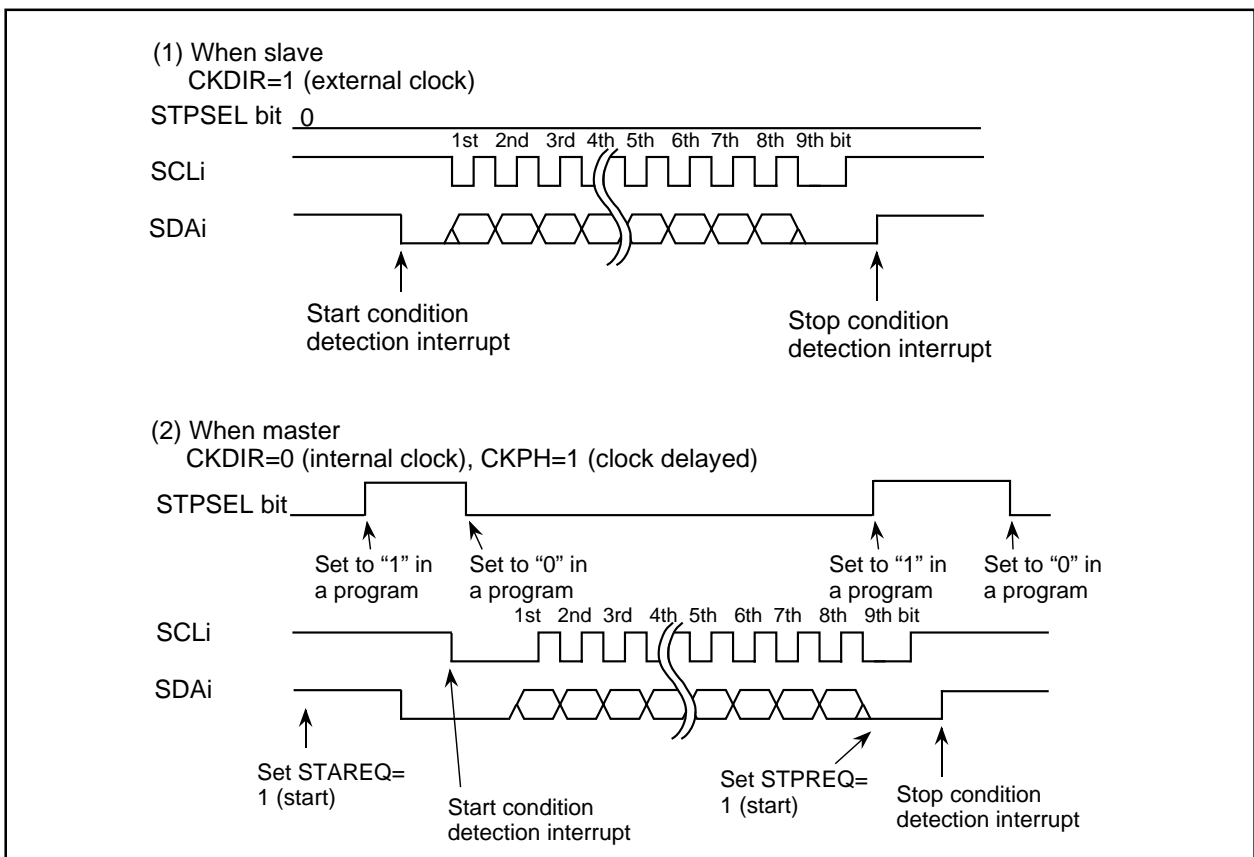
Also, if a start condition is to be output, make sure the STAREQ bit is set to "1" before setting the STSPSEL bit to "1".



Serial I/O (Special Modes)

**Table 1.20.5. 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
Star/stop condition interrupt request generation timing	Start/stop condition detection	Finish generating start/stop condition



**Figure 1.20.4. STSPSEL Bit Functions**

• Arbitration

Unmatching of the transmit data and SDAi pin input data is checked synchronously with the rising edge of SCLi. Use the UiSMR register's ABC bit to select the timing at which the UiRB register's ABT bit 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 byte-wise, clear the ABT bit to "0" (undetected) after detecting acknowledge in the first byte, before transferring the next byte.

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

### • Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 1.20.4.

The UiSMR2 register's CSC bit 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 UiBRG register value 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 UiSMR2 register's SWC bit 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 UiSMR4 register's SCLHI bit is set to "1" (enabled), SCLi output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the UiSMR2 register's SWC2 bit = 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 UiSMR4 register's SWC9 bit is set to "1" (SCL hold low enabled) when the UiSMR3 register's CKPH bit = 1, the SCLi pin is fixed to low-level output at the falling edge of the clock pulse next to the ninth. Setting the SWC9 bit = 0 (SCL hold low disabled) frees the SCLi pin from low-level output.

### • SDA Output

The data written to the UiTB register bit 7 to bit 0 (D7 to D0) is sequentially output beginning with D7. The ninth bit (D8) is ACK or NACK.

The initial value of SDAi transmit output can only be set when IICM = 1 (I<sup>2</sup>C mode) and the UiMR register's SMD2 to SMD0 bits = '0002' (serial I/O disabled).

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

Setting the UiSMR2 register's SDHI bit = 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).

### • 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.

## Serial I/O (Special Modes)

---

### • ACK and NACK

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

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

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

### • Initialization of Transmission/Reception

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

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

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

## Serial I/O (Special Modes)

## Special Mode 2

Multiple slaves can be serially communicated from one master. Synchronous clock polarity and phase are selectable. Table 1.20.6 lists the specifications of Special Mode 2. Table 1.20.7 lists the registers used in Special Mode 2 and the register values set. Figure 1.20.5 shows communication control example for Special Mode 2.

**Table 1.20.6. Special Mode 2 Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>Master mode               <ul style="list-style-type: none"> <li>UiMR(i=0 to 2) register's CKDIR bit = "0" (internal clock) : <math>f_j / 2(n+1)</math></li> <li><math>f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math>. n: Setting value of UiBRG register 0016 to FF16</li> </ul> </li> <li>Slave mode               <ul style="list-style-type: none"> <li>CKDIR bit = "1" (external clock selected) : Input from CLKi pin</li> </ul> </li> </ul>
Transmit/receive control	Controlled by input/output ports
Transmission start condition	<ul style="list-style-type: none"> <li>Before transmission can start, the following requirements must be met (Note 1)               <ul style="list-style-type: none"> <li>The TE bit of UiC1 register= 1 (transmission enabled)</li> <li>The TI bit of UiC1 register = 0 (data present in UiTB register)</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>Before reception can start, the following requirements must be met (Note 1)               <ul style="list-style-type: none"> <li>The RE bit of UiC1 register= 1 (reception enabled)</li> <li>The TE bit of UiC1 register= 1 (transmission enabled)</li> <li>The TI bit of UiC1 register= 0 (data present in the UiTB register)</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>For transmission, one of the following conditions can be selected               <ul style="list-style-type: none"> <li>The UiIRS bit of UiC1 register = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission)</li> <li>The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register</li> </ul> </li> <li>For reception               <ul style="list-style-type: none"> <li>When transferring data from the UARTi receive register to the UiRB register (at completion of reception)</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error (Note 2)               <ul style="list-style-type: none"> <li>This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 7th bit of the next data</li> </ul> </li> </ul>
Select function	<ul style="list-style-type: none"> <li>Clock phase setting               <ul style="list-style-type: none"> <li>Selectable from four combinations of transfer clock polarities and phases</li> </ul> </li> </ul>

Note 1: When an external clock is selected, the conditions must be met while if the UiC0 register's CKPOL bit = "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the UiC0 register's 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.

Note 2: If an overrun error occurs, the value of UiRB register will be indeterminate. The IR bit of SiRIC register does not change.

### Serial I/O (Special Modes)

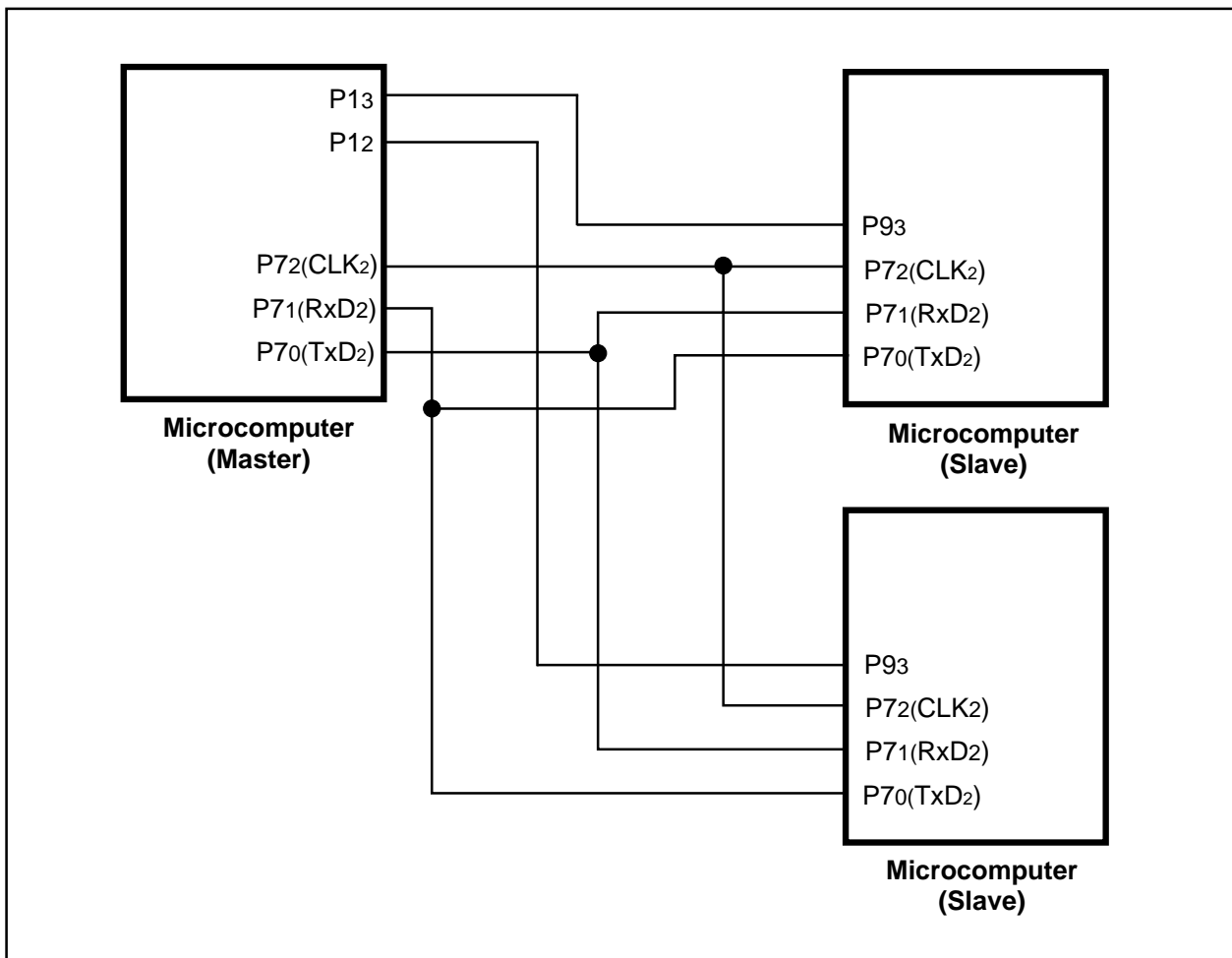


Figure 1.20.5. Serial Bus Communication Control Example (UART2)

## Serial I/O (Special Modes)

Table 1. 20. 7. Registers to Be Used and Settings in Special Mode 2

Register	Bit	Function
UiTB(Note3)	0 to 7	Set transmission data
UiRB(Note3)	0 to 7	Reception data can be read
	OER	Overflow error flag
UiBRG	0 to 7	Set a transfer rate
UiMR(Note3)	SMD2 to SMD0	Set to '0012'
	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(Note 2)
	CKPOL	Clock phases can be set in combination with the UiSMR3 register's CKPH bit
	UFORM	Set to "0"
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS (Note 1)	Select UART2 transmit interrupt cause
	U2RRM(Note 1), U2LCH, 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 UiC0 register's CKPOL bit
	NODC	Set to "0"
	0, 2, 4 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select UART0 and UART1 transmit interrupt cause
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1, RCSP, 7	Set to "0"

Note 1: Set the U0C0 and U1C1 register bit 4 and bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.

Note 2: TxDi pin is N channel open-drain output. Set the U2C0 register's NCH bit to "0".

Note 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

**Serial I/O (Special Modes)**

• **Clock Phase Setting Function**

One of four combinations of transfer clock phases and polarities can be selected using the UiSMR3 register's CKPH bit and the UiC0 register's CKPOL bit.

Make sure the transfer clock polarity and phase are the same for the master and slaves to be communicated.

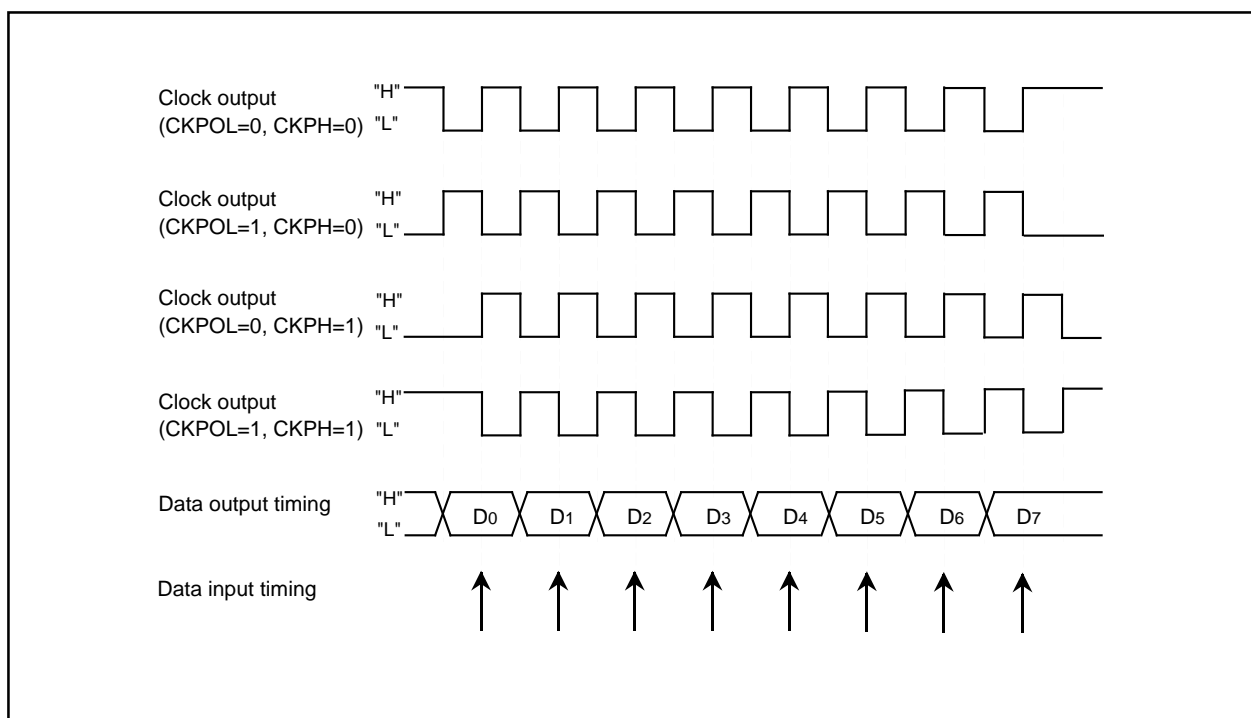
**(a) Master (Internal Clock)**

Figure 1.20.6 shows the transmission and reception timing in master (internal clock).

**(b) Slave (External Clock)**

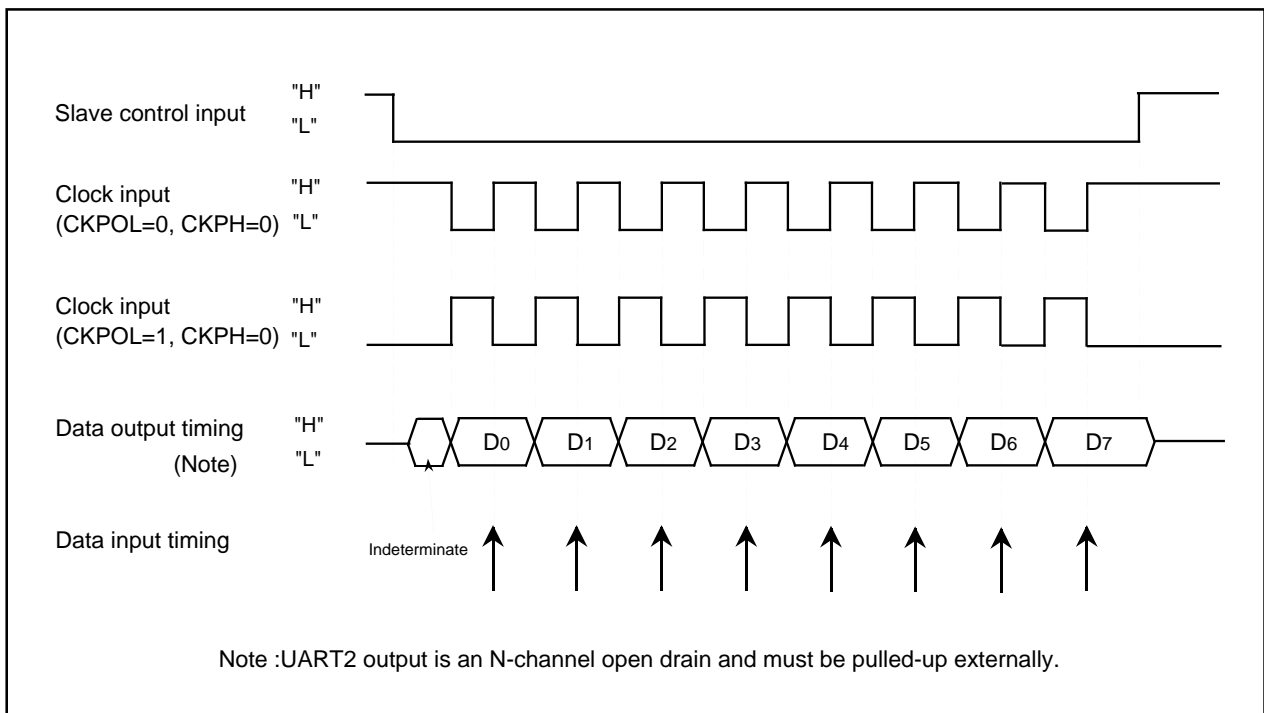
Figure 1.20.7 shows the transmission and reception timing (CKPH=0) in slave (external clock) while

Figure 1.20.8 shows the transmission and reception timing (CKPH=1) in slave (external clock).

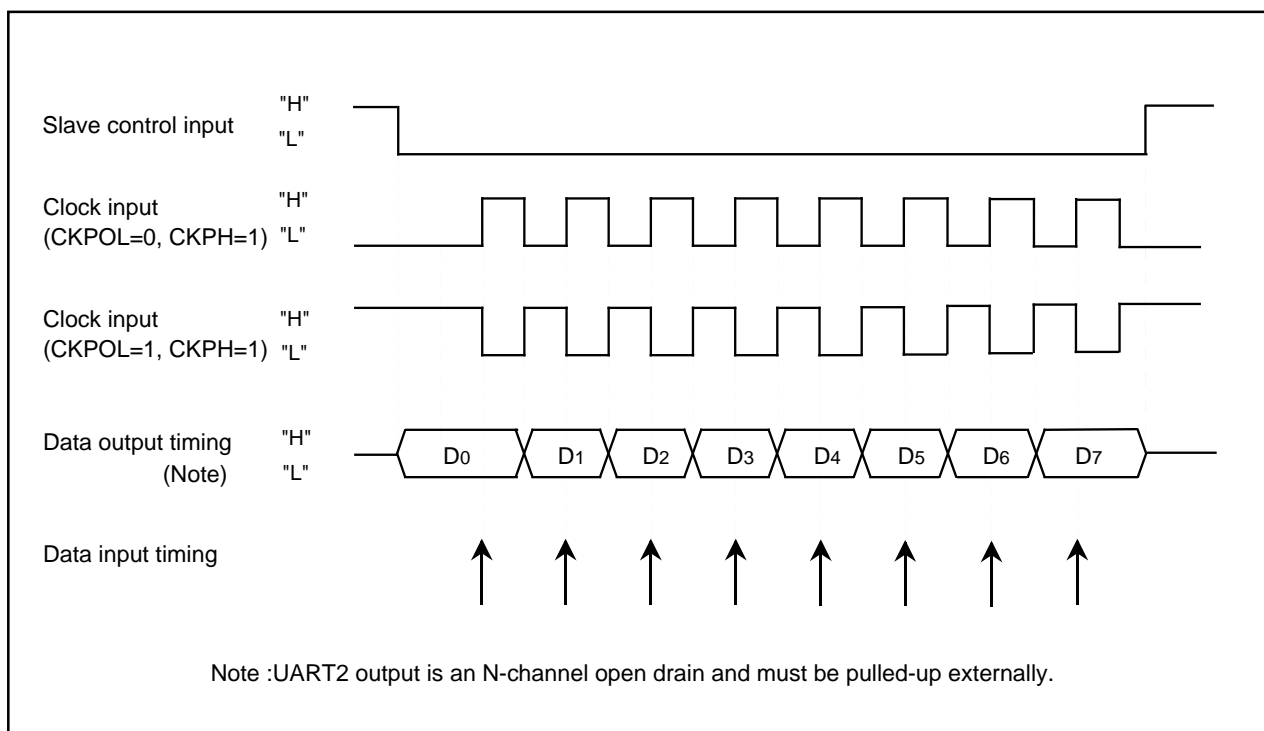


**Figure 1.20.6. Transmission and Reception Timing in Master Mode (Internal Clock)**

**Serial I/O (Special Modes)**



**Figure 1.20.7. Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)**



**Figure 1.20.8. Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)**



## Serial I/O (Special Modes)

**Special Mode 3 (IE mode)**

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

Table 1.20.8 lists the registers used in IE mode and the register values set. Figure 1.20.9 shows the functions of bus collision detect function related bits.

If the TxDi pin (i = 0 to 2) output level and RxDi pin input level do not match, a UARTi bus collision detect interrupt request is generated.

Use the IFSR2A register's IFSR26 and IFSR27 bits to enable the UART0/UART1 bus collision detect function.

**Table 1. 20. 8. Registers to Be Used and Settings in IE Mode**

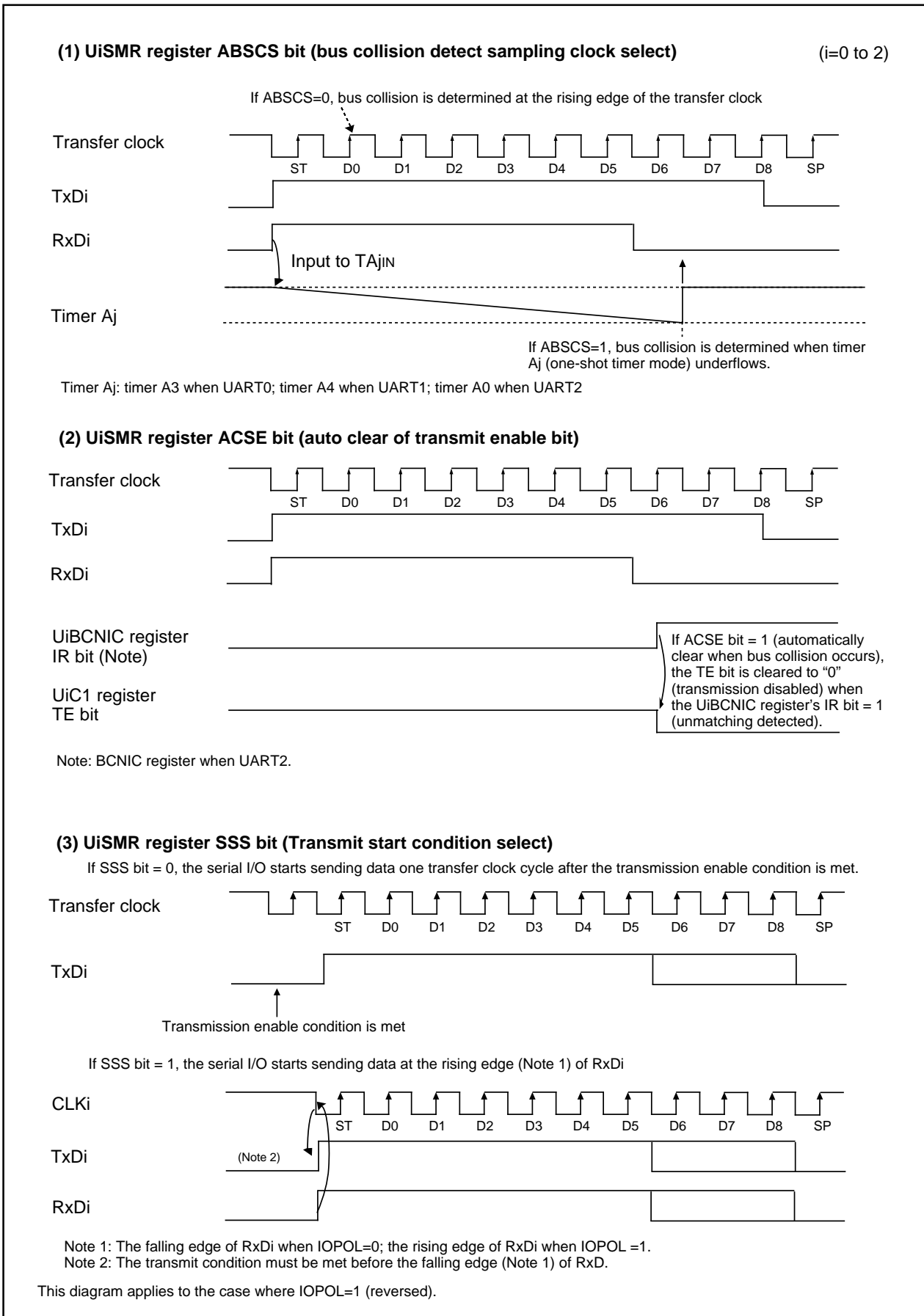
Register	Bit	Function
UiTB	0 to 8	Set transmission data
UiRB(Note3)	0 to 8	Reception data can be read
	OER,FER,PER,SUM	Error flag
UiBRG	---	Set a transfer rate
UiMR	SMD2 to SMD0	Set to '1102'
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Invalid because PRYE=0
	PRYE	Set to "0"
	IOPOL	Select the TxD/RxD input/output polarity
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 mode (Note 2)
	CKPOL	Set to "0"
	UFORM	Set to "0"
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS (Note 1)	Select the source of UART2 transmit interrupt
	UiRRM (Note 1), UiLCH, UiERE	Set to "0"
UiSMR	0 to 3, 7	Set to "0"
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to "1" to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
IFSR2A	IFSR26, IFSR27	Set to "1"
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1,RCSP,7	Set to "0"

Note 1: Set the U0C0 and U1C1 registers bit 4 and bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.

Note 2: TxD2 pin is N channel open-drain output. Set the U2C0 register's NCH bit to "0".

Note 3: Not all register bits are described above. Set those bits to "0" when writing to the registers in IEmode. i= 0 to 2

**Serial I/O (Special Modes)**



**Figure 1.20.9. Bus Collision Detect Function-Related Bits**

## Serial I/O (Special Modes)

**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.

Tables 1.20.9 lists the specifications of SIM mode. Table 1.20.10 lists the registers used in the SIM mode and the register values set.

**Table 1.20.9. SIM Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Direct format</li> <li>• Inverse format</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• U2MR register's CKDIR bit = "0" (internal clock) : <math>f_i / 16(n+1)</math>  <math>f_i = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}</math>. n: Setting value of U2BRG register 00<sub>16</sub> to FF<sub>16</sub></li> <li>• CKDIR bit = "1" (external clock) : <math>f_{EXT} / 16(n+1)</math>  <math>f_{EXT}</math>: Input from CLK2 pin. n: Setting value of U2BRG register 00<sub>16</sub> to FF<sub>16</sub></li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>• Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> <li>– The TE bit of U2C1 register= 1 (transmission enabled)</li> <li>– The TI bit of U2C1 register = 0 (data present in U2TB register)</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>• Before reception can start, the following requirements must be met <ul style="list-style-type: none"> <li>– The RE bit of U2C1 register= 1 (reception enabled)</li> <li>– Start bit detection</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• For transmission When the serial I/O finished sending data from the U2TB transfer register (U2IRS bit =1)</li> <li>• For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception)</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error (Note) This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data</li> <li>• Framing error This error occurs when the number of stop bits set is not detected</li> <li>• Parity error 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</li> <li>• Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered</li> </ul>

Note: If an overrun error occurs, the value of U2RB register will be indeterminate. The IR bit of S2RIC register does not change.

## Serial I/O (Special Modes)

Table 1. 20. 10. Registers to Be Used and Settings in SIM Mode

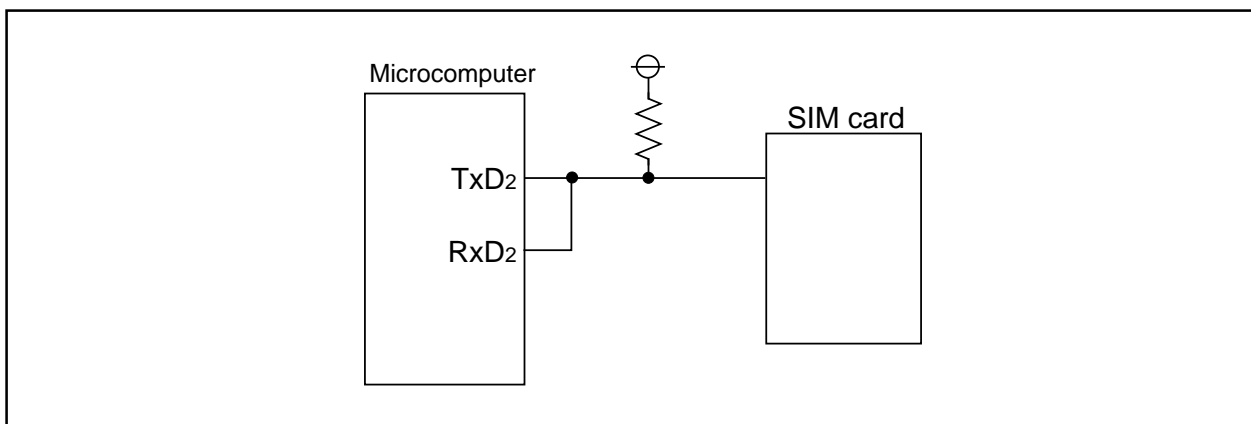
Register	Bit	Function
U2TB(Note)	0 to 7	Set transmission data
U2RB(Note)	0 to 7	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	---	Set a transfer rate
U2MR	SMD2 to SMD0	Set to '1012'
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Set this bit to "1" for direct 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(Note)	0 to 3	Set to "0"
U2SMR2	0 to 7	Set to "0"
U2SMR3	0 to 7	Set to "0"
U2SMR4	0 to 7	Set to "0"

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



**Serial I/O (Special Modes)**

Figure 1.20.11 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.



**Figure 1.20.11. SIM Interface Connection**

**(a) Parity Error Signal Output**

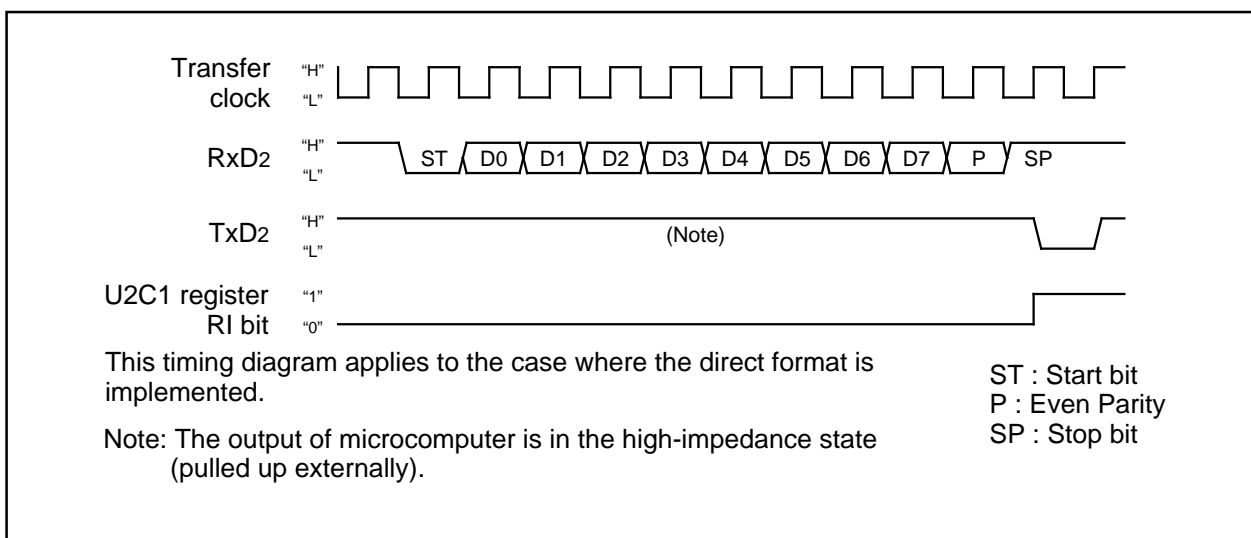
The parity error signal is enabled by setting the U2C1 register's U2ERE bit to "1".

- When receiving

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 1.20.12. 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 service routine.



**Figure 1.20.12. Parity Error Signal Output Timing**

Serial I/O (Special Modes)

(b) Format

• Direct Format

Set the U2MR register's PRY bit to "1", U2C0 register's UFORM bit to "0" and U2C1 register's U2LCH bit to "0".

• Inverse Format

Set the PRY bit to "0", UFORM bit to "1" and U2LCH bit to "1".

Figure 1.20.13 shows the SIM interface format.

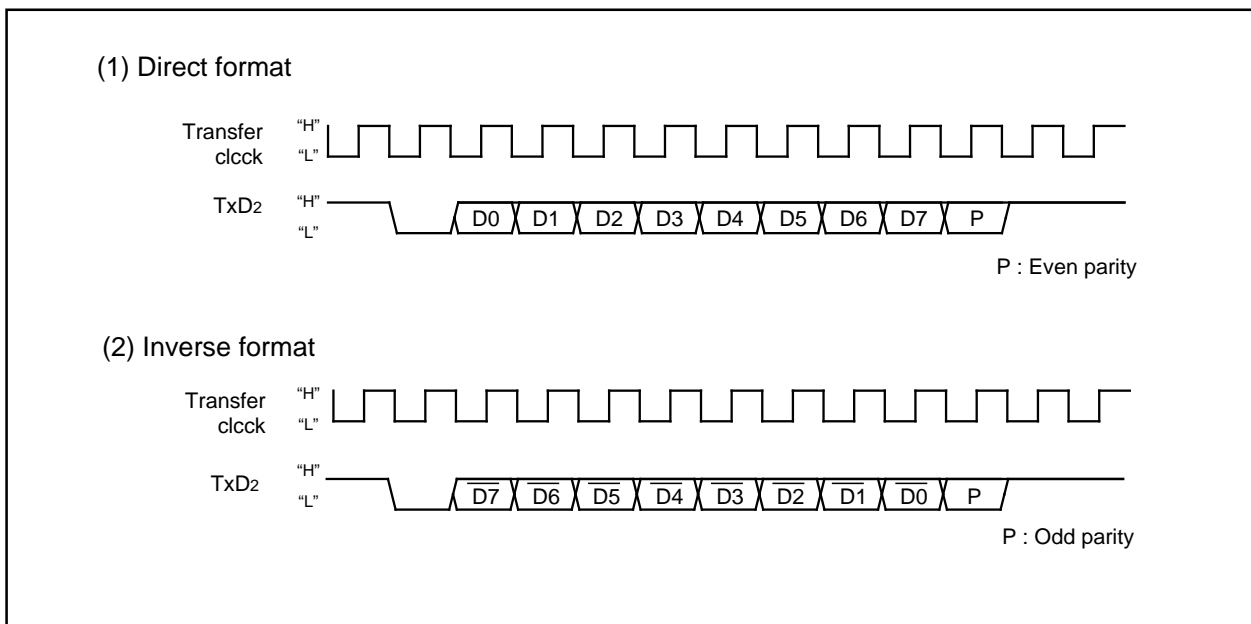


Figure 1.20.13. SIM Interface Format

SI/O3, SI/O4

## SI/O3 and SI/O4

SI/O3 and SI/O4 are exclusive clock-synchronous serial I/Os.

Figure 1.21.1 shows the block diagram of SI/O3 and SI/O4, and Figure 1.21.2 shows the SI/O3 and SI/O4-related registers.

Table 1.21.1 shows the specifications of SI/O3 and SI/O4.

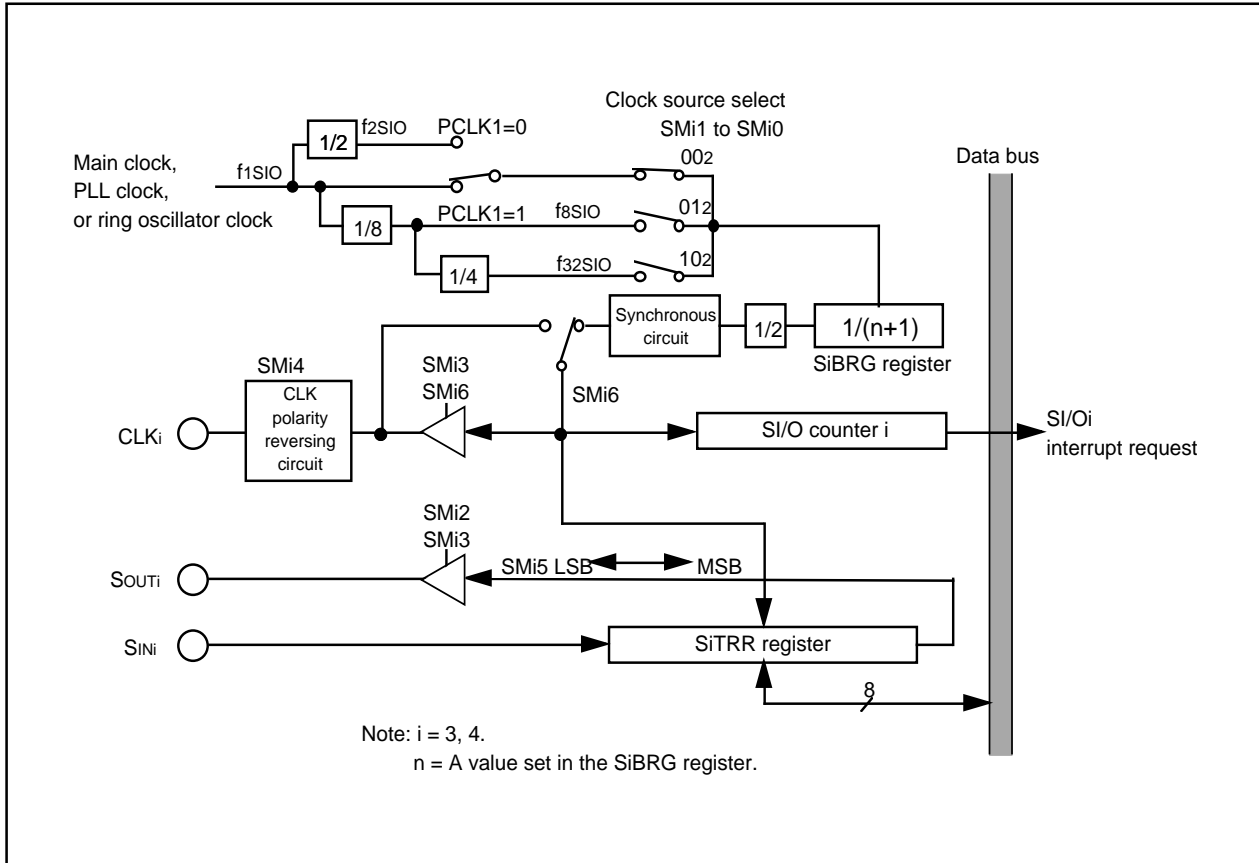
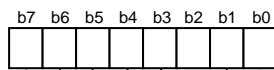


Figure 1.21.1. SI/O3 and SI/O4 Block Diagram



S I/Oi control register (i = 3, 4) (Note 1)



Symbol	Address	After reset
S3C	0362 <sub>16</sub>	0100000 <sub>16</sub>
S4C	0366 <sub>16</sub>	0100000 <sub>16</sub>

Bit symbol	Bit name	Description	RW
SMi0	Internal synchronous clock select bit	b1 b0 0 0 : Selecting f1SIO or f2SIO	RW
		0 1 : Selecting f8SIO	RW
SMi1		1 0 : Selecting f32SIO	RW
		1 1 : Must not be set.	
SMi2	Souti output disable bit (Note 4)	0 : Souti output 1 : Souti output disable(high impedance)	RW
SMi3	S I/Oi port select bit	0 : Input/output port 1 : Souti output, CLKi function	RW
SMi4	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
SMi5	Transfer direction select bit	0 : LSB first 1 : MSB first	RW
SMi6	Synchronous clock select bit	0 : External clock (Note 2) 1 : Internal clock (Note 3)	RW
SMi7	Souti initial value set bit	Effective when SMi3 = 0 0 : "L" output 1 : "H" output	RW

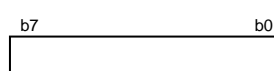
Note 1: Make sure this register is written to by the next instruction after setting the PRCR register's PRC2 bit to "1" (write enable).

Note 2: Set the SMi3 bit to "1" (Souti output, CLKi function).

Note 3: Set the SMi3 bit to "1" and the corresponding port direction bit to "0" (input mode).

Note 4: Effective when SMi3 bit = 1.

SI/Oi bit rate generator (i = 3, 4) (Notes 1, 2)



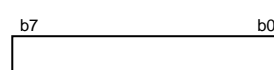
Symbol	Address	After reset
S3BRG	0363 <sub>16</sub>	Indeterminate
S4BRG	0367 <sub>16</sub>	Indeterminate

Description	Setting range	RW
Assuming that set value = n, BRGi divides the count source by n + 1	00 <sub>16</sub> to FF <sub>16</sub>	WO

Note 1: Write to this register while serial I/O is neither transmitting nor receiving.

Note 2: Use MOV instruction to write to this register.

SI/Oi transmit/receive register (i = 3, 4) (Note 1, 2)



Symbol	Address	After reset
S3TRR	0360 <sub>16</sub>	Indeterminate
S4TRR	0364 <sub>16</sub>	Indeterminate

Description	RW
Transmission/reception starts by writing transmit data to this register. After transmission/reception finishes, reception data can be read by reading this register.	RW

Note 1: Write to this register while serial I/O is neither transmitting nor receiving.

Note 2: To receive data, set the corresponding port direction bit for Sini to "0" (input mode).

Figure 1.21.2. S3C and S4C Registers, S3BRG and S4BRG Registers, and S3TRR and S4TRR Registers

Table 1.21.1. SI/O3 and SI/O4 Specifications

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>SiC (i=3, 4) register's SMi6 bit = "1" (internal clock) : <math>f_j / 2^{(n+1)}</math>  <math>f_j = f_{1SIO}, f_{8SIO}, f_{32SIO}</math>. n=Setting value of SiBRG register 0016 to FF16.</li> <li>SMi6 bit = "0" (external clock) : Input from CLKi pin (Note 1)</li> </ul>
Transmission/reception start condition	<ul style="list-style-type: none"> <li>Before transmission/reception can start, the following requirements must be met Write transmit data to the SiTRR register (Notes 2, 3)</li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When SiC register's SMi4 bit = 0 The rising edge of the last transfer clock pulse (Note 4)</li> <li>When SMi4 = 1 The falling edge of the last transfer clock pulse (Note 4)</li> </ul>
CLKi pin function	I/O port, transfer clock input, transfer clock output
SOUTi pin function	I/O port, transmit data output, high-impedance
SINI pin function	I/O port, receive data input
Select function	<ul style="list-style-type: none"> <li>LSB first or MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> <li>Function for setting an SOUTi initial value set function When the SiC register's SMi6 bit = 0 (external clock), the SOUTi pin output level while not transmitting can be selected.</li> <li>CLK polarity selection Whether transmit data is output/input timing at the rising edge or falling edge of transfer clock can be selected.</li> </ul>

Note 1: To set the SiC register's SMi6 bit to "0" (external clock), follow the procedure described below.

- If the SiC register's SMi4 bit = 0, write transmit data to the SiTRR register while input on the CLKi pin is high. The same applies when rewriting the SiC register's SMi7 bit.
- If the SMi4 bit = 1, write transmit data to the SiTRR register while input on the CLKi pin is low. The same applies when rewriting the SMi7 bit.
- Because shift operation continues as long as the transfer clock is supplied to the SI/Oi circuit, stop the transfer clock after supplying eight pulses. If the SMi6 bit = 1 (internal clock), the transfer clock automatically stops.

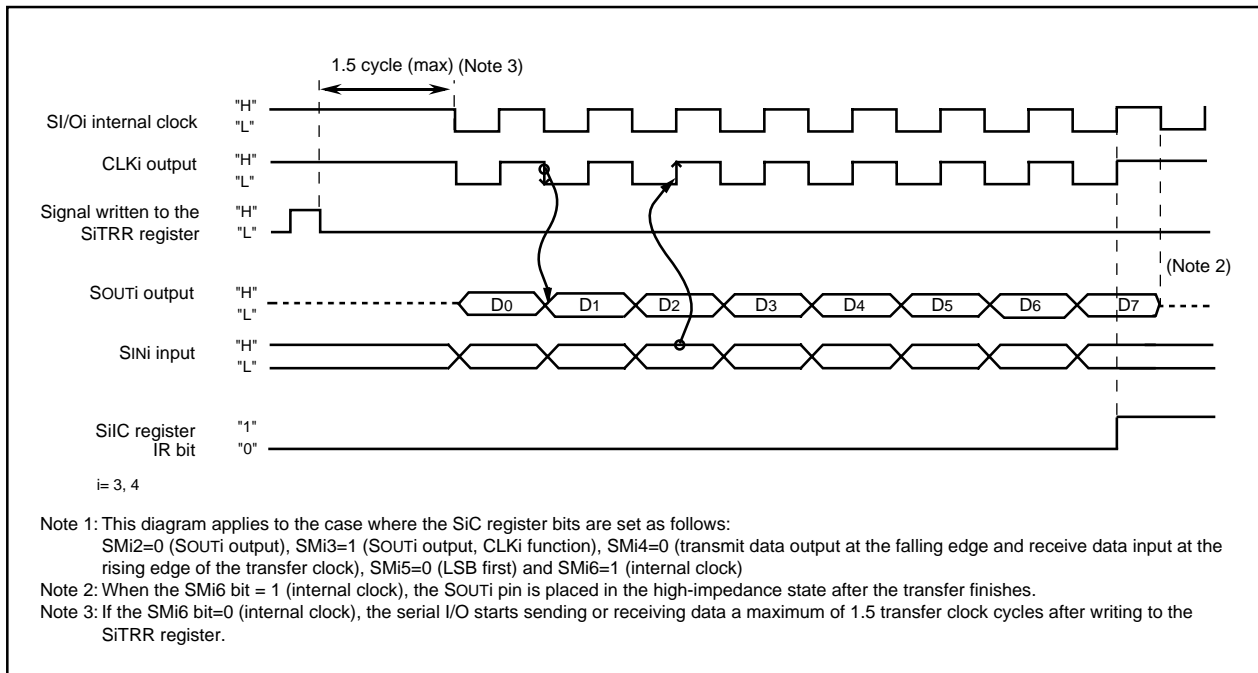
Note 2: Unlike UART0 to UART2, SI/Oi (i = 3 to 4) is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the SiTRR register during transmission.

Note 3: When the SiC register's SMi6 bit = 1 (internal clock), SOUTi retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to a high-impedance state. However, if transmit data is written to the SiTRR register during this period, SOUTi immediately goes to a high-impedance state, with the data hold time thereby reduced.

Note 4: When the SiC register's SMi6 bit = 1 (internal clock), the transfer clock stops in the high state if the SMi4 bit = 0, or stops in the low state if the SMi4 bit = 1.

**(a) SI/Oi Operation Timing**

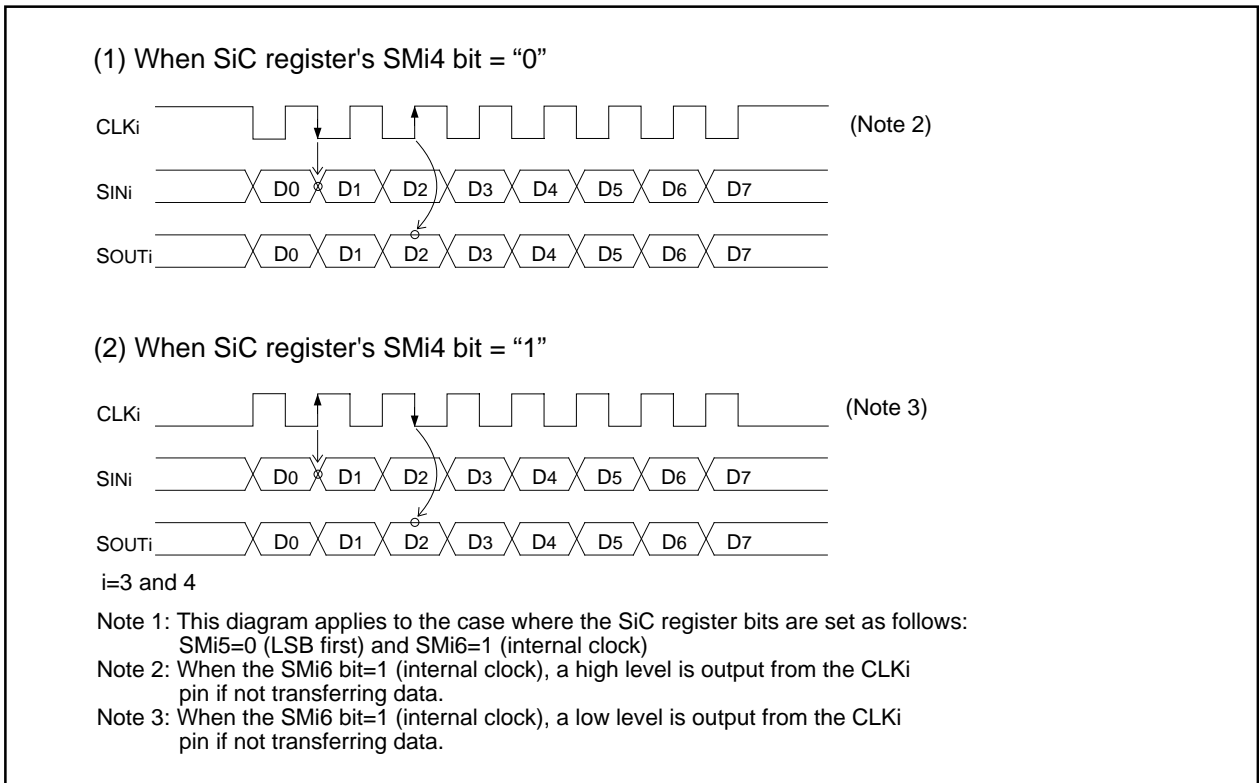
Figure 1.21.3 shows the SI/Oi operation timing



**Figure 1.21.3. SI/Oi Operation Timing**

**(b) CLK Polarity Selection**

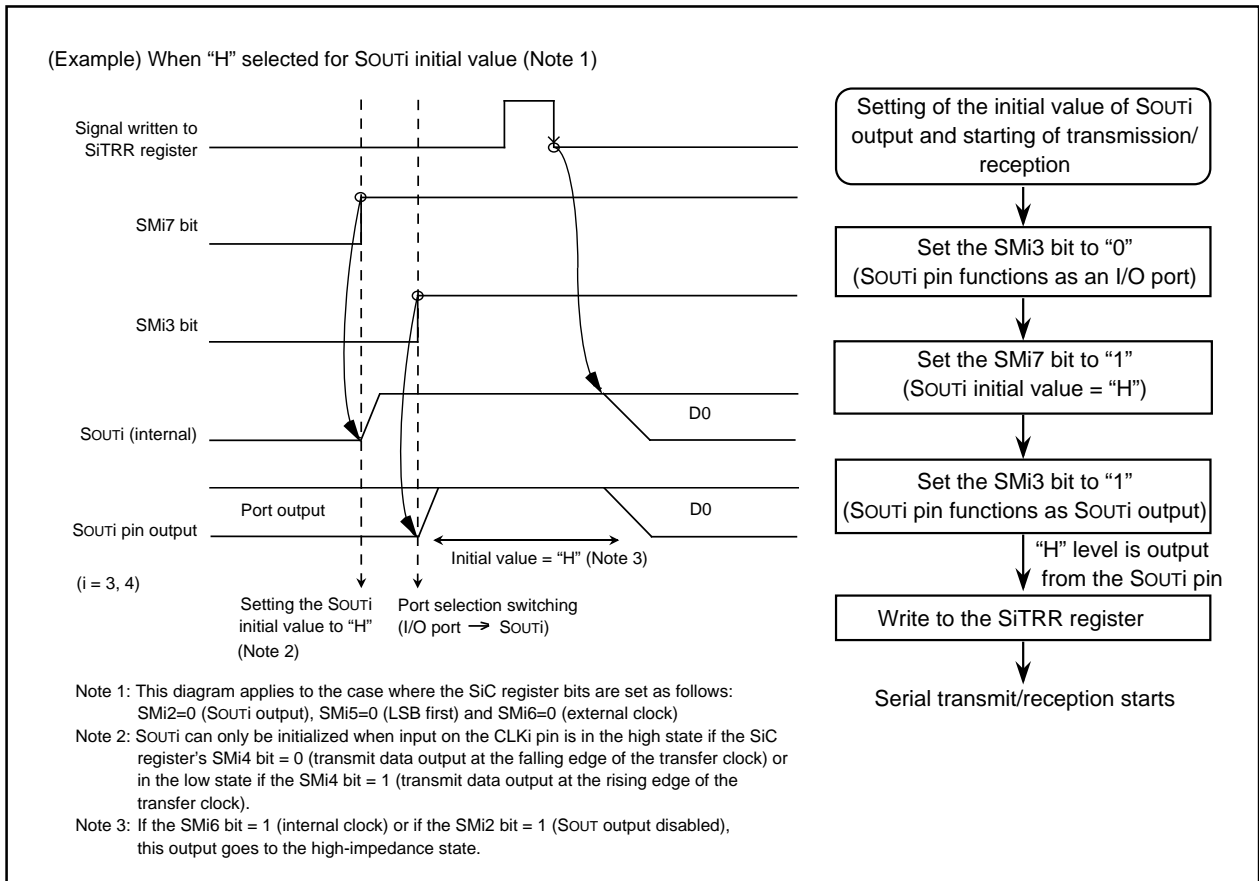
The SiC register's SMI4 bit allows selection of the polarity of the transfer clock. Figure 1.21.4 shows the polarity of the transfer clock.



**Figure 1.21.4. Polarity of Transfer Clock**

**(c) Functions for Setting an SOUTi Initial Value**

If the SiC register's SMi6 bit = 0 (external clock), the SOUTi pin output can be fixed high or low when not transferring. Figure 1.21.5 shows the timing chart for setting an SOUTi initial value and how to set it.



**Figure 1.21.5. SOUTi's Initial Value Setting**

## A-D Converter

## 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<sub>0</sub> to P10<sub>7</sub>, P9<sub>5</sub>, P9<sub>6</sub>, P0<sub>0</sub> to P0<sub>7</sub>, and P2<sub>0</sub> to P2<sub>7</sub>. Similarly,  $\overline{ADTRG}$  input shares the pin with P9<sub>7</sub>. 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 AN<sub>i</sub>, AN0<sub>i</sub>, and AN2<sub>i</sub> pins (i = 0 to 7).

Table 1.22.1 shows the performance of the A-D converter. Figure 1.22.1 shows the block diagram of the A-D converter, and Figures 1.22.2 and 1.22.3 show the A-D converter-related registers.

**Table 1.22.1. Performance of A-D Converter**

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AVCC (VCC1)
Operating clock $\phi_{AD}$ (Note 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 <ul style="list-style-type: none"> <li>• With 8-bit resolution: <math>\pm 2</math>LSB</li> <li>• With 10-bit resolution               <ul style="list-style-type: none"> <li>- AN<sub>0</sub> to AN<sub>7</sub> input : <math>\pm 3</math>LSB</li> <li>- AN<sub>00</sub> to AN<sub>07</sub> input and AN<sub>20</sub> to AN<sub>27</sub> input : <math>\pm 7</math>LSB</li> <li>- ANEX0 and ANEX1 input (including mode in which external operation amp is connected) : <math>\pm 7</math>LSB</li> </ul> </li> </ul> When AVCC = VREF = 3.3V <ul style="list-style-type: none"> <li>• With 8-bit resolution: <math>\pm 2</math>LSB</li> <li>• With 10-bit resolution               <ul style="list-style-type: none"> <li>- AN<sub>0</sub> to AN<sub>7</sub> input : <math>\pm 5</math>LSB</li> <li>- AN<sub>00</sub> to AN<sub>07</sub> input and AN<sub>20</sub> to AN<sub>27</sub> input : <math>\pm 7</math>LSB</li> <li>- ANEX0 and ANEX1 input (including mode in which external operation amp is connected) : <math>\pm 7</math>LSB</li> </ul> </li> </ul>
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8 pins (AN <sub>0</sub> to AN <sub>7</sub> ) + 2 pins (ANEX0 and ANEX1) + 8 pins (AN <sub>00</sub> to AN <sub>07</sub> ) + 8 pins (AN <sub>20</sub> to AN <sub>27</sub> )
A-D conversion start condition	<ul style="list-style-type: none"> <li>• Software trigger               <ul style="list-style-type: none"> <li>The ADCON0 register's ADST bit is set to "1" (A-D conversion starts)</li> </ul> </li> <li>• External trigger (retriggerable)               <ul style="list-style-type: none"> <li>Input on the <math>\overline{ADTRG}</math> pin changes state from high to low after the ADST bit is set to "1" (A-D conversion starts)</li> </ul> </li> </ul>
Conversion speed per pin	<ul style="list-style-type: none"> <li>• Without sample and hold function               <ul style="list-style-type: none"> <li>8-bit resolution: 49 <math>\phi_{AD}</math> cycles, 10-bit resolution: 59 <math>\phi_{AD}</math> cycles</li> </ul> </li> <li>• With sample and hold function               <ul style="list-style-type: none"> <li>8-bit resolution: 28 <math>\phi_{AD}</math> cycles, 10-bit resolution: 33 <math>\phi_{AD}</math> cycles</li> </ul> </li> </ul>

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

Note 2: The fAD frequency must be 10 MHz or less.

Without sample-and-hold function, limit the fAD frequency to 250kHz or less.

With the sample and hold function, limit the fAD frequency to 1MHz or less.

Note 3: If VCC2 < VCC1, do not use AN<sub>00</sub> to AN<sub>07</sub> and AN<sub>20</sub> to AN<sub>27</sub> as analog input pins.



A-D Converter

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol ADCON0	Address 03D616	After reset 0000XXX2
Bit symbol	Bit name	Function	RW							
CH0	Analog input pin select bit	Function varies with each operation mode	RW							
CH1			RW							
CH2			RW							
MD0	A-D operation mode select bit 0	<sup>b4:b3</sup> 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW							
MD1			RW							
TRG	Trigger select bit	0 : Software trigger 1 : 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	See Note 3 for the ADCON2 register	RW							

Note: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

A-D control register 1 (Note 1)

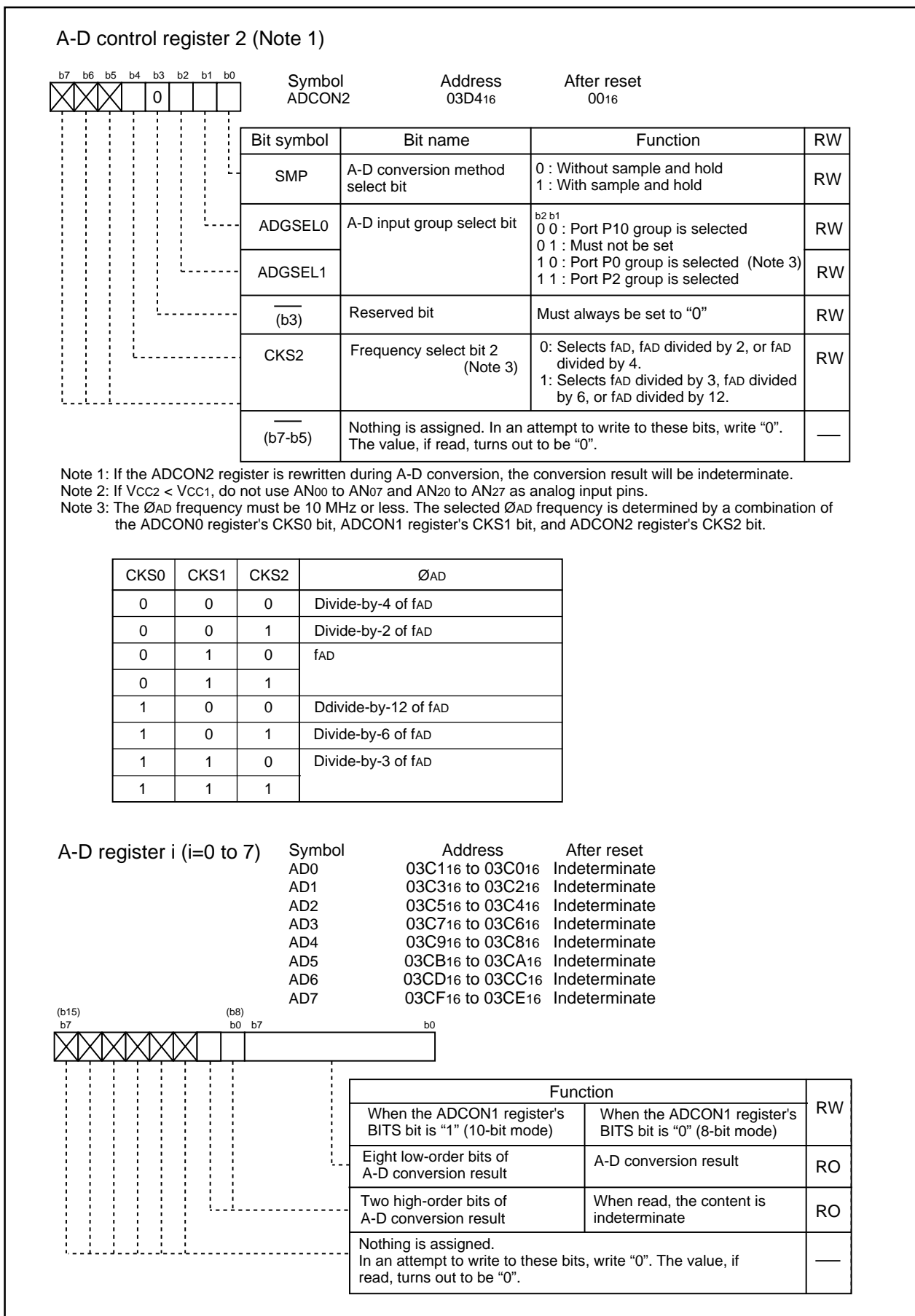
b7	b6	b5	b4	b3	b2	b1	b0	Symbol ADCON1	Address 03D716	After reset 0016
Bit symbol	Bit name	Function	RW							
SCAN0	A-D sweep pin select bit	Function varies with each operation mode	RW							
SCAN1			RW							
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	RW							
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW							
CKS1	Frequency select bit 1	See Note 3 for the ADCON2 register	RW							
VCUT	Vref connect bit (Note 2)	0 : Vref not connected 1 : Vref connected	RW							
OPA0	External op-amp connection mode bit	Function varies with each operation mode	RW							
OPA1			RW							

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.

Note 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 1.22.2. ADCON0 to ADCON1 Registers

**A-D Converter**



**Figure 1.22.3. ADCON2 Register, and AD0 to AD7 Registers**



## A-D Converter

**(1) One-shot Mode**

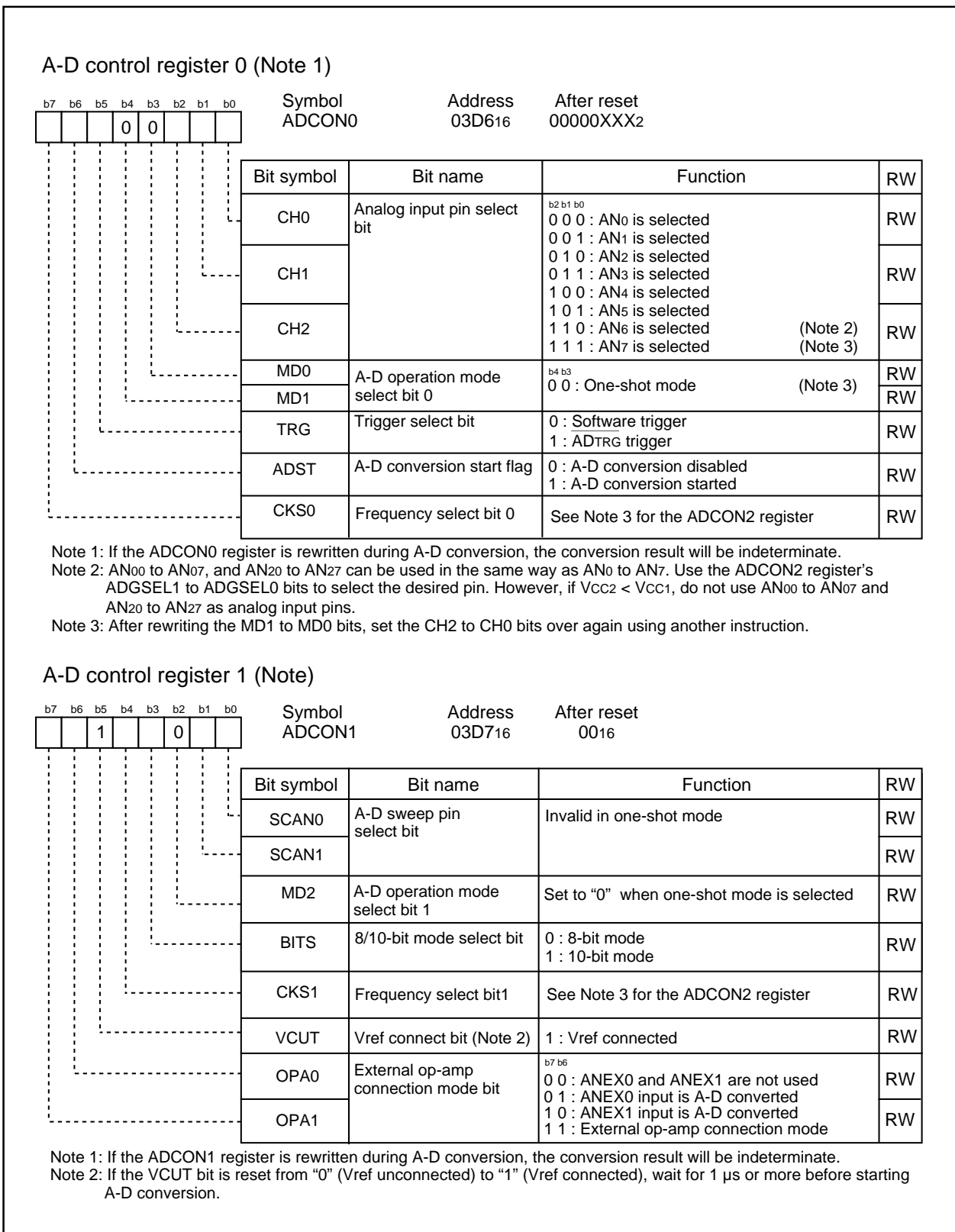
In this mode, the input voltage on one selected pin is A-D converted once. Table 1.22.2 shows the specifications of one-shot mode. Figure 1.22.4 shows the ADCON0 to ADCON1 registers in one-shot mode.

**Table 1.22.2. One-shot Mode Specifications**

Item	Specification
Function	The input voltage on one pin selected by the ADCON0 register's CH2 to CH0 bits and ADCON2 register's ADGSEL1 to ADGSEL0 bits or the ADCON1 register's OPA1 to OPA0 bits is A-D converted once.
A-D conversion start condition	<ul style="list-style-type: none"> <li>• When the ADCON0 register's TRG bit is "0" (software trigger) The ADCON0 register's ADST bit is set to "1" (A-D conversion starts)</li> <li>• When the TRG bit is "1" (ADTRG trigger) Input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to "1" (A-D conversion starts)</li> </ul>
A-D conversion stop condition	<ul style="list-style-type: none"> <li>• Completion of A-D conversion (If a software trigger is selected, the ADST bit is cleared to "0" (A-D conversion halted).)</li> <li>• Set the ADST bit to "0"</li> </ul>
Interrupt request generation timing	Completion of A-D conversion
Analog input pin (Note)	Select one pin from AN0 to AN7, AN00 to AN07, AN20 to AN27, ANEX0 to ANEX1
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: If  $V_{CC2} < V_{CC1}$ , do not use AN00–AN07 and AN20–AN27 as analog input pins.

**A-D Converter**



**Figure 1.22.4. ADCON0 Register and ADCON1 Register (One-shot Mode)**

## A-D Converter

**(2) Repeat mode**

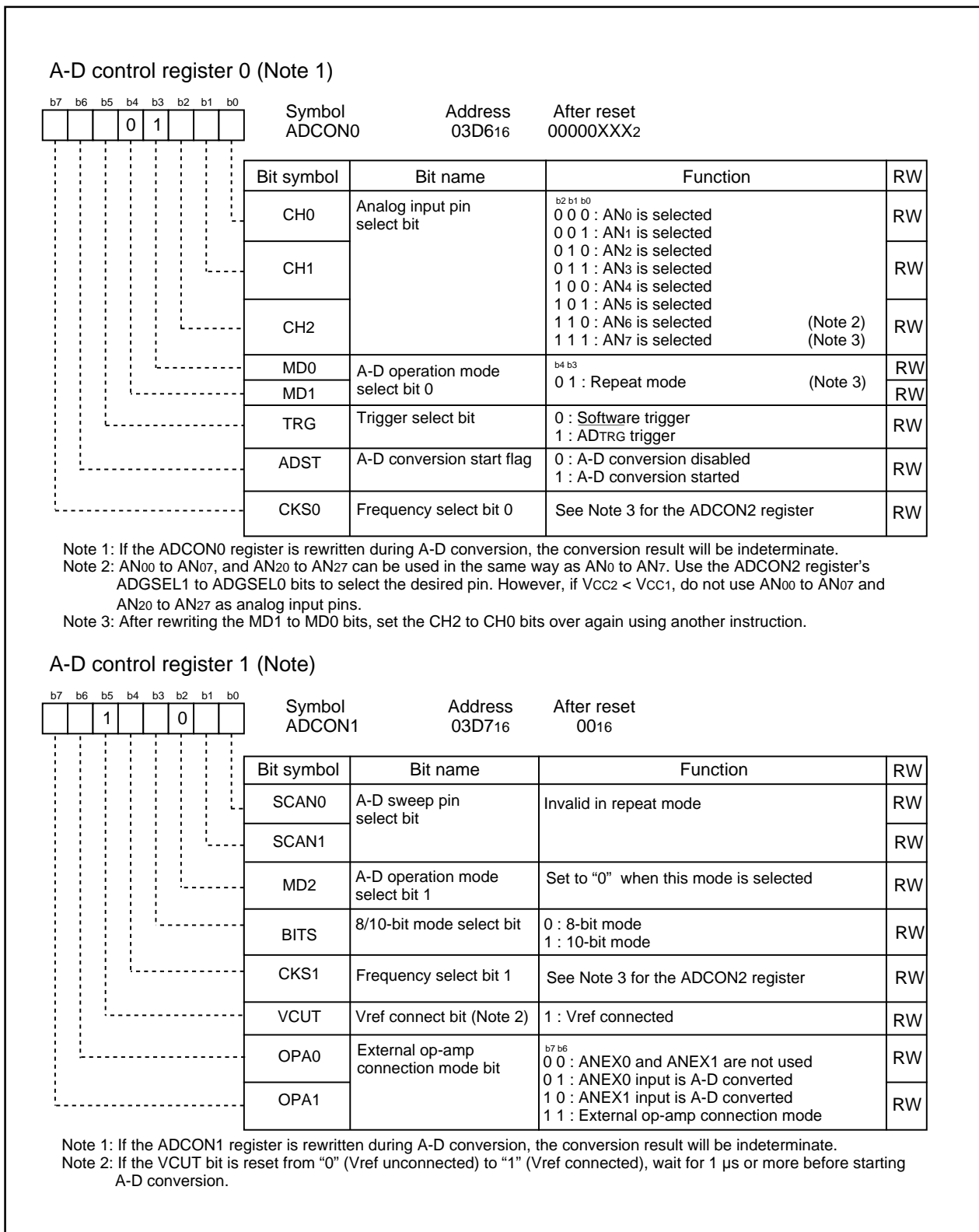
In this mode, the input voltage on one selected pin is A-D converted repeatedly. Table 1.22.3 shows the specifications of repeat mode. Figure 1.22.5 shows the ADCON0 to ADCON1 registers in repeat mode.

**Table 1.22.3. Repeat Mode Specifications**

Item	Specification
Function	The input voltage on one pin selected by the ADCON0 register's CH2 to CH0 bits and ADCON2 register's ADGSEL1 to ADGSEL0 bits or the ADCON1 register's OPA1 to OPA0 bits is A-D converted repeatedly.
A-D conversion start condition	<ul style="list-style-type: none"> <li>• When the ADCON0 register's TRG bit is "0" (software trigger) The ADCON0 register's ADST bit is set to "1" (A-D conversion starts)</li> <li>• When the TRG bit is "1" (<math>\overline{\text{ADTRG}}</math> trigger) Input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to "1" (A-D conversion starts)</li> </ul>
A-D conversion stop condition	Set the ADST bit to "0" (A-D conversion halted)
Interrupt request generation timing	None generated
Analog input pin (Note)	Select one pin from AN0 to AN7, AN00 to AN07, AN20 to AN27, ANEX0 to ANEX1
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: If  $V_{CC2} < V_{CC1}$ , do not use AN00–AN07 and AN20–AN27 as analog input pins.

**A-D Converter**



**Figure 1.22.5. ADCON0 Register and ADCON1 Register (Repeat Mode)**

## A-D Converter

**(3) Single Sweep Mode**

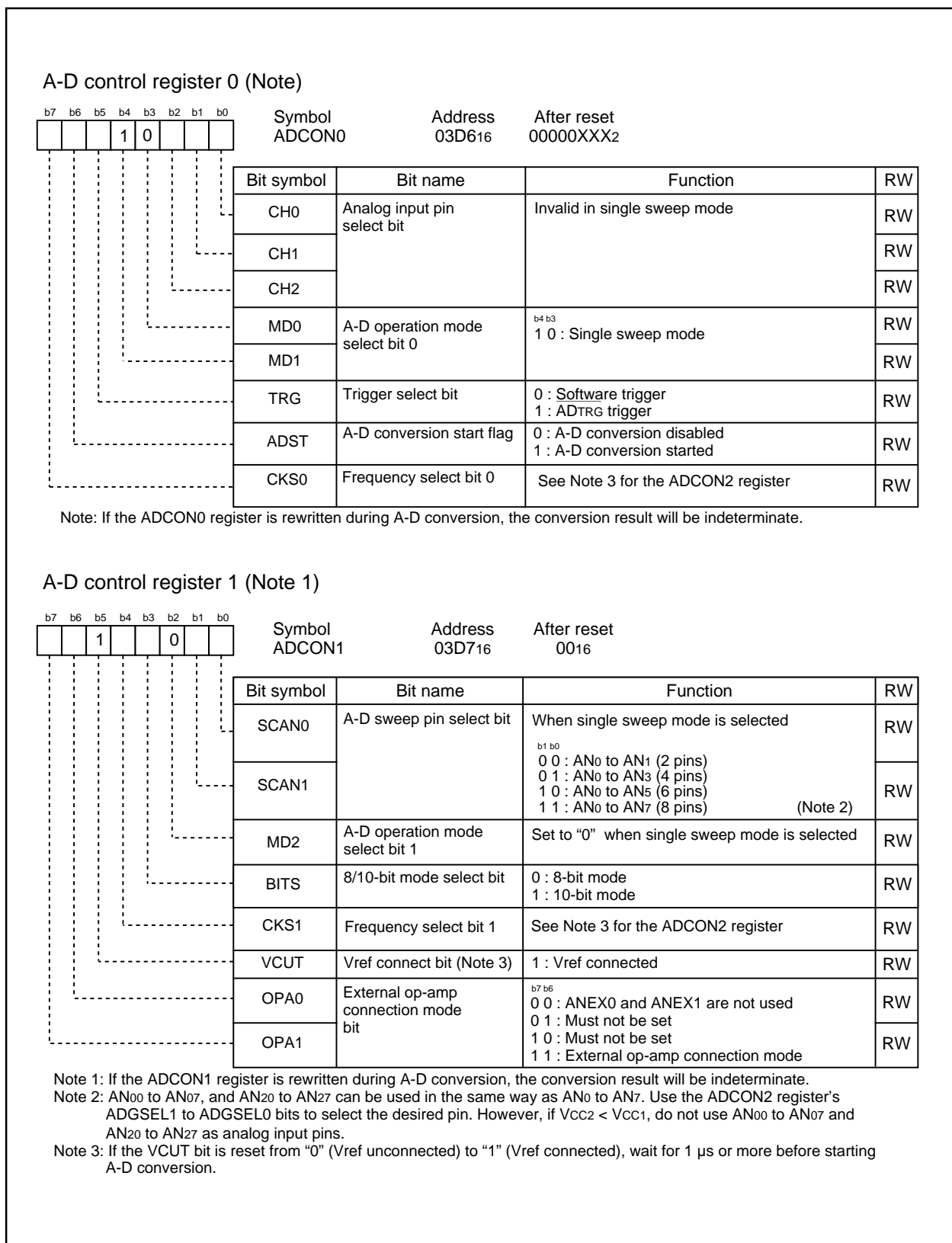
In this mode, the input voltages on selected pins are A-D converted, one pin at a time. Table 1.22.4 shows the specifications of single sweep mode. Figure 1.22.6 shows the ADCON0 to ADCON1 registers in single sweep mode.

**Table 1.22.4. Single Sweep Mode Specifications**

Item	Specification
Function	The input voltages on pins selected by the ADCON1 register's SCAN1 to SCAN0 bits and ADCON2 register's ADGSEL1 to ADGSEL0 bits are A-D converted, one pin at a time.
A-D conversion start condition	<ul style="list-style-type: none"> <li>• When the ADCON0 register's TRG bit is "0" (software trigger) The ADCON0 register's ADST bit is set to "1" (A-D conversion starts)</li> <li>• When the TRG bit is "1" (<math>\overline{\text{ADTRG}}</math> trigger) Input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to "1" (A-D conversion starts)</li> </ul>
A-D conversion stop condition	<ul style="list-style-type: none"> <li>• Completion of A-D conversion (If a software trigger is selected, the ADST bit is cleared to "0" (A-D conversion halted).)</li> <li>• Set the ADST bit to "0"</li> </ul>
Interrupt request generation timing	Completion of A-D conversion
Analog input pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), AN0 to AN7 (8 pins) (Note)
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: AN00 to AN07, and AN20 to AN27 can be used in the same way as AN0 to AN7. However, if  $V_{CC2} < V_{CC1}$ , do not use AN00–AN07 and AN20–AN27 as analog input pins.

**A-D Converter**



**Figure 1.22.6. ADCON0 Register and ADCON1 Register (Single Sweep Mode)**

## A-D Converter

**(4) Repeat Sweep Mode 0**

In this mode, the input voltages on selected pins are A-D converted repeatedly. Table 1.22.5 shows the specifications of repeat sweep mode 0. Figure 1.22.7 shows the ADCON0 to ADCON1 registers in repeat sweep mode 0.

**Table 1.22.5. Repeat Sweep Mode 0 Specifications**

Item	Specification
Function	The input voltages on pins selected by the ADCON1 register's SCAN1 to SCAN0 bits and ADCON2 register's ADGSEL1 to ADGSEL0 bits are A-D converted repeatedly.
A-D conversion start condition	<ul style="list-style-type: none"> <li>• When the ADCON0 register's TRG bit is "0" (software trigger) The ADCON0 register's ADST bit is set to "1" (A-D conversion starts)</li> <li>• When the TRG bit is "1" (<math>\overline{\text{ADTRG}}</math> trigger) Input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to "1" (A-D conversion starts)</li> </ul>
A-D conversion stop condition	Set the ADST bit to "0" (A-D conversion halted)
Interrupt request generation timing	None generated
Analog input pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), AN0 to AN7 (8 pins) (Note)
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: AN00 to AN07, and AN20 to AN27 can be used in the same way as AN0 to AN7. However, if  $V_{CC2} < V_{CC1}$ , do not use AN00–AN07 and AN20–AN27 as analog input pins.

**A-D Converter**

**A-D control register 0 (Note)**

b7	b6	b5	b4	b3	b2	b1	b0	Symbol ADCON0	Address 03D6 <sub>16</sub>	After reset 0000XXX <sub>2</sub>	
			1	1							
								Bit symbol	Bit name	Function	RW
								CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	RW
							CH1	RW			
							CH2	RW			
								MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
							MD1	RW			
								TRG	Trigger select bit	0 : Software trigger 1 : 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	See Note 3 for the ADCON2 register	RW

Note: If the ADCON0 register is rewritten during A-D conversion, the conversion result will be indeterminate.

**A-D control register 1 (Note 1)**

b7	b6	b5	b4	b3	b2	b1	b0	Symbol ADCON1	Address 03D7 <sub>16</sub>	After reset 00 <sub>16</sub>	
		1				0					
								Bit symbol	Bit name	Function	RW
								SCAN0	A-D sweep pin select bit	When repeat sweep mode 0 is selected	RW
							SCAN1	b1 b0 0 0 : AN <sub>0</sub> , AN <sub>1</sub> (2 pins) 0 1 : AN <sub>0</sub> to AN <sub>3</sub> (4 pins) 1 0 : AN <sub>0</sub> to AN <sub>5</sub> (6 pins) 1 1 : AN <sub>0</sub> to AN <sub>7</sub> (8 pins) (Note 2)			RW
								MD2	A-D operation mode select bit 1	Set to "0" when repeat sweep mode 0 is selected	RW
								BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
								CKS1	Frequency select bit 1	See Note 3 for the ADCON2 register	RW
								VCUT	Vref connect bit (Note 3)	1 : Vref connected	RW
								OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : Must not be set 1 0 : Must not be set 1 1 : External op-amp connection mode	RW
							OPA1	RW			

Note 1: If the ADCON1 register is rewritten during A-D conversion, the conversion result will be indeterminate.

Note 2: AN<sub>0</sub> to AN<sub>7</sub>, and AN<sub>20</sub> to AN<sub>27</sub> can be used in the same way as AN<sub>0</sub> to AN<sub>7</sub>. Use the ADCON2 register's ADGSEL1 to ADGSEL0 bits to select the desired pin. However, if V<sub>CC2</sub> < V<sub>CC1</sub>, do not use AN<sub>0</sub> to AN<sub>7</sub> and AN<sub>20</sub> to AN<sub>27</sub> as analog input pins.

Note 3: If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 μs or more before starting A-D conversion.

**Figure 1.22.7. ADCON0 Register and ADCON1 Registers (Repeat Sweep Mode 0)**



## A-D Converter

**(5) Repeat Sweep Mode 1**

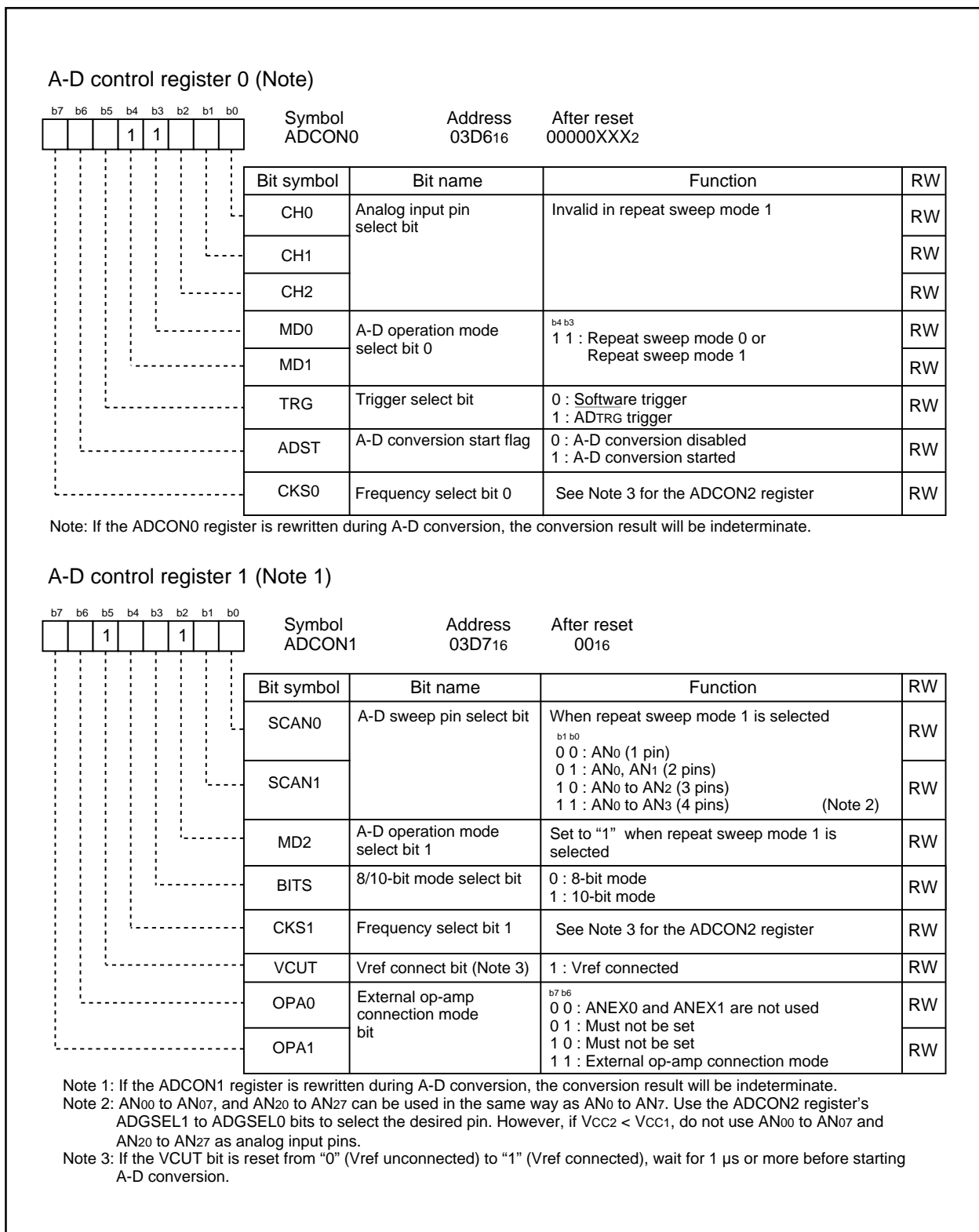
In this mode, the input voltages on all pins are A-D converted repeatedly, with priority given to the selected pins. Table 1.22.6 shows the specifications of repeat sweep mode 1. Figure 1.22.8 shows the ADCON0 to ADCON1 registers in repeat sweep mode 1.

**Table 1.22.6. Repeat Sweep Mode 1 Specifications**

Item	Specification
Function	The input voltages on all pins selected by the ADCON2 register's ADGSEL1 to ADGSEL0 bits are A-D converted repeatedly, with priority given to pins selected by the ADCON1 register's SCAN1 to SCAN0 bits and ADGSEL1 to ADGSEL0 bits. Example : If AN <sub>0</sub> selected, input voltages are A-D converted in order of AN <sub>0</sub> → AN <sub>1</sub> → AN <sub>0</sub> → AN <sub>2</sub> → AN <sub>0</sub> → AN <sub>3</sub> , and so on.
A-D conversion start condition	<ul style="list-style-type: none"> <li>When the ADCON0 register's TRG bit is "0" (software trigger) The ADCON0 register's ADST bit is set to "1" (A-D conversion starts)</li> <li>When the TRG bit is "1" (<math>\overline{\text{ADTRG}}</math> trigger) Input on the <math>\overline{\text{ADTRG}}</math> pin changes state from high to low after the ADST bit is set to "1" (A-D conversion starts)</li> </ul>
A-D conversion stop condition	Set the ADST bit to "0" (A-D conversion halted)
Interrupt request generation timing	None generated
Analog input pins to be given priority when A-D converted	Select from AN <sub>0</sub> (1 pins), AN <sub>0</sub> to AN <sub>1</sub> (2 pins), AN <sub>0</sub> to AN <sub>2</sub> (3 pins), AN <sub>0</sub> to AN <sub>3</sub> (4 pins) (Note)
Reading of result of A-D converter	Read one of the AD0 to AD7 registers that corresponds to the selected pin

Note: AN<sub>0</sub> to AN<sub>7</sub>, and AN<sub>20</sub> to AN<sub>27</sub> can be used in the same way as AN<sub>0</sub> to AN<sub>7</sub>. However, if VCC2 < VCC1, do not use AN<sub>0</sub>–AN<sub>7</sub> and AN<sub>20</sub>–AN<sub>27</sub> as analog input pins.

**A-D Converter**



**Figure 1.22.8. ADCON0 Register and ADCON1 Register (Repeat Sweep Mode 1)**

## A-D Converter

**(a) Resolution Select Function**

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

**(b) Sample and Hold**

If the ADCON2 register's SMP bit is set to "1" (with sample-and-hold), the conversion speed per pin is increased to  $28 \varnothing_{AD}$  cycles for 8-bit resolution or  $33 \varnothing_{AD}$  cycles for 10-bit resolution. Sample-and-hold is effective in all operation modes. Select whether or not to use the sample-and-hold function before starting A-D conversion.

**(c) Extended Analog Input Pins**

In one-shot and repeat modes, the ANEX0 and ANEX1 pins can be used as analog input pins. Use the ADCON1 register's OPA1 to OPA0 bits 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.

**(d) External Operation Amp Connection Mode**

Multiple analog inputs can be amplified using a single external op-amp via the ANEX0 and ANEX1 pins. Set the ADCON1 register's OPA1 OPA0 bits to '112' (external op-amp connection mode). The inputs from ANi (i = 0 to 7) (Note 1) are output from the ANEX0 pin. Amplify this output with an external op-amp before sending it back to the ANEX1 pin. The A-D conversion result is stored in the corresponding ADi register. The A-D conversion speed depends on the response characteristics of the external op-amp. Note that the ANEX0 and ANEX1 pins cannot be directly connected to each other. Figure 1.22.9 is an example of how to connect the pins in external operation amp.

Note: AN0i and AN2i can be used the same as ANi. However, if  $V_{CC2} < V_{CC1}$ , do not use AN0i and AN2i as analog input pins.

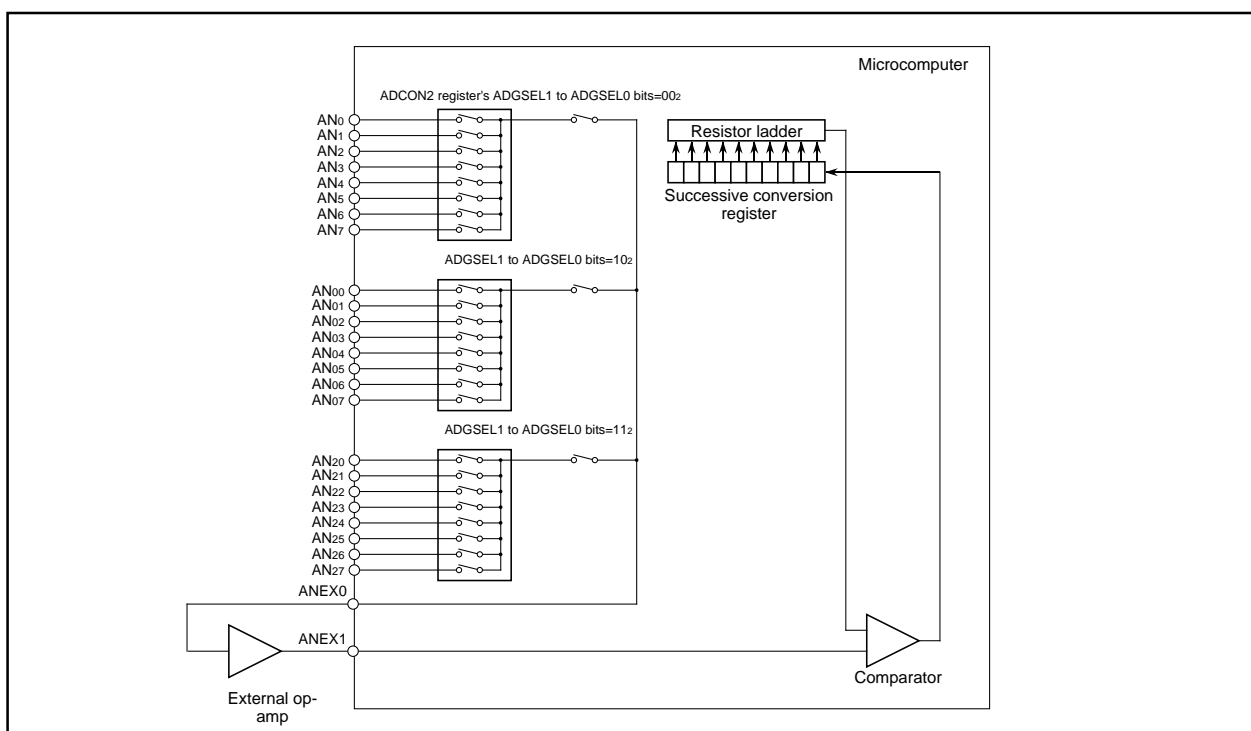


Figure 1.22.9. External Op-amp Connection

## A-D Converter

### (e) 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 ADCON1 register's VCUT bit. 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 ADCON0 register's ADST bit to "1" (A-D conversion start). The VCUT and ADST bits cannot be set to "1" at the same time. Nor can the VCUT bit be set to "0" (VREF unconnected) during A-D conversion.

Note that this does not affect VREF for the D-A converter (irrelevant).

### (f) Analog Input Pin and External Sensor Equivalent Circuit Example

Figure 1.22.10 shows analog input pin and external sensor equivalent circuit example.

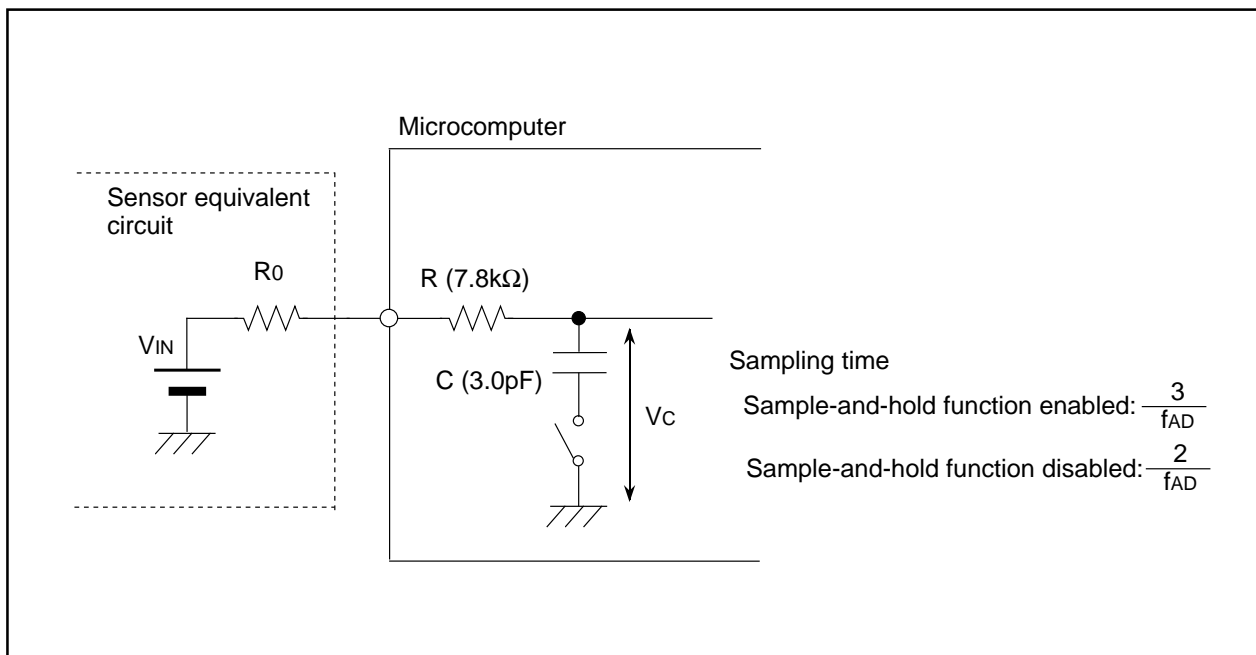


Figure 1.22.10. Analog Input Pin and External Sensor Equivalent Circuit

## A-D Converter

**(g) Caution of Using A-D Converter**

- (1) Make sure the port direction bits for those pins that are used as analog inputs are set to "0" (input mode). Also, if the ADCON0 register's TGR bit = 1 (external trigger), make sure the port direction bit for the ADTRG pin is set to "0" (input mode).
- (2) 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.)
- (3) 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), ANoi, and ANzi) each and the AVSS pin. Similarly, insert a capacitor between the VCC pin and the VSS pin. Figure 1.22.11 is an example connection of each pin.
- (4) If  $V_{CC2} < V_{CC1}$ , do not use AN00 to AN07 and AN20 to AN27 as analog input pins.
- (5) 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 or single-sweep mode  
Check to see that A-D conversion is completed before reading the target ADi register. (Check the IR bit in the ADIC register to see if A-D conversion is completed.)
  - When operating in repeat mode or repeat sweep mode 0 or 1  
Use the main clock for CPU clock directly without dividing it.
- (6) If A-D conversion is forcibly terminated while in progress by setting the ADCON0 register's ADST bit 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.

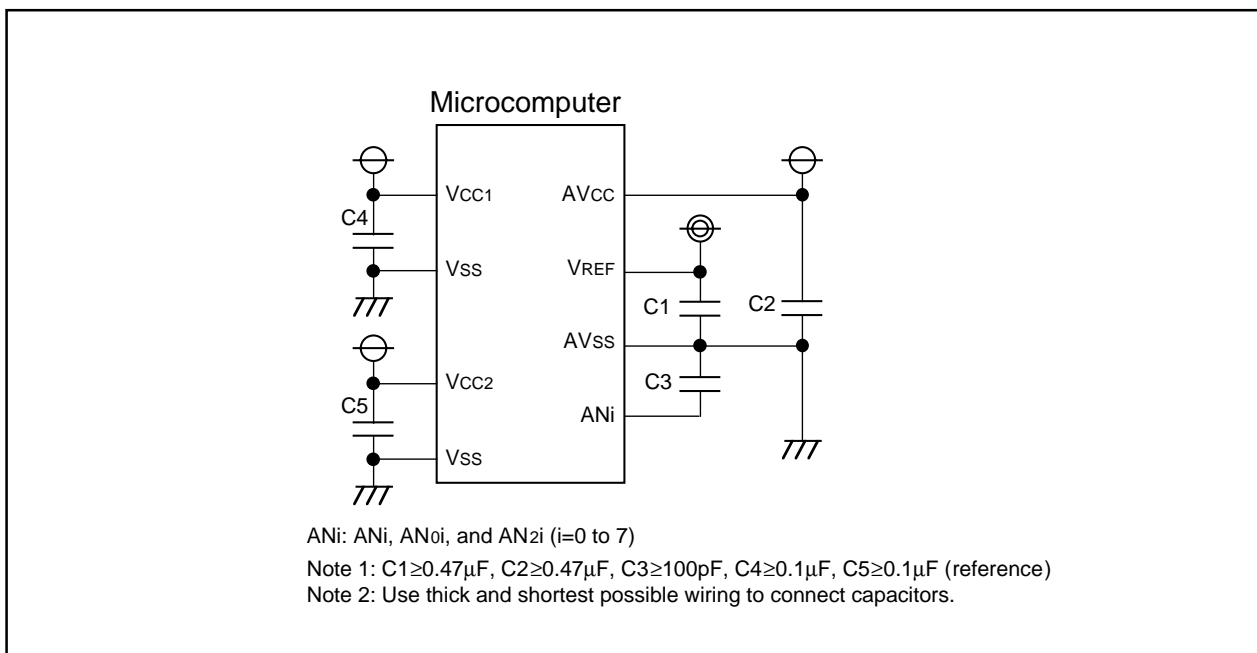


Figure 1.22.11. VCC, VSS, AVCC, AVSS, VREF and ANi Connection

D-A Converter

# D-A Converter

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

D-A conversion is performed by writing to the DA<sub>i</sub> register (i = 0 to 1). To output the result of conversion, set the DACON register's DA<sub>i</sub>E bit to "1" (output enabled). Before D-A conversion can be used, the corresponding port direction bit must be cleared to "0" (input mode). Setting the DA<sub>i</sub>E bit to "1" removes a pull-up from the corresponding port.

Output analog voltage (V) is determined by a set value (n : decimal) in the DA<sub>i</sub> register.

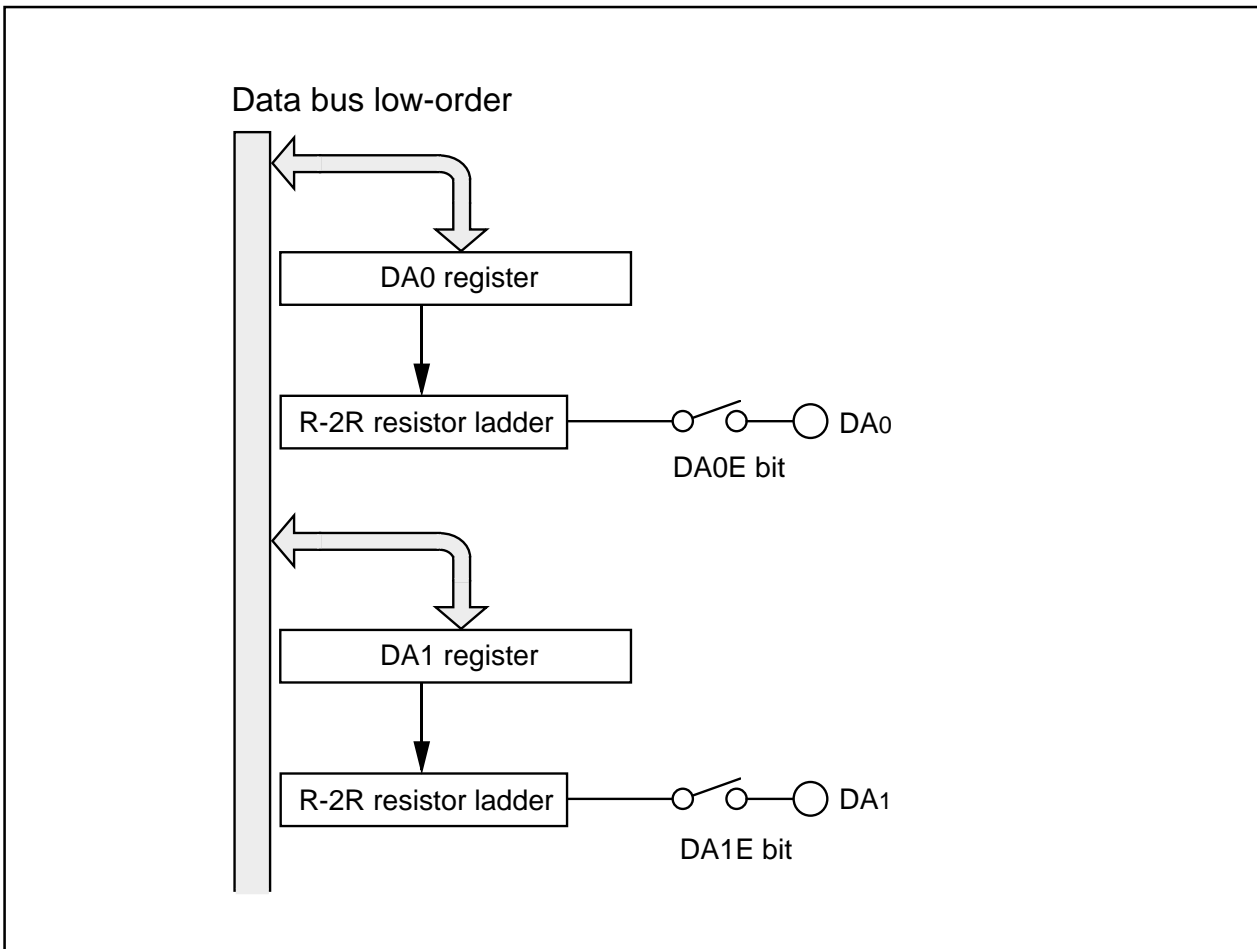
$$V = V_{REF} \times n / 256 \quad (n = 0 \text{ to } 255)$$

V<sub>REF</sub> : reference voltage

Table 1.23.1 lists the performance of the D-A converter. Figure 1.23.1 shows the block diagram of the D-A converter. Figure 1.23.2 shows the D-A converter related registers. Figure 1.23.3 shows the D-A converter equivalent circuit.

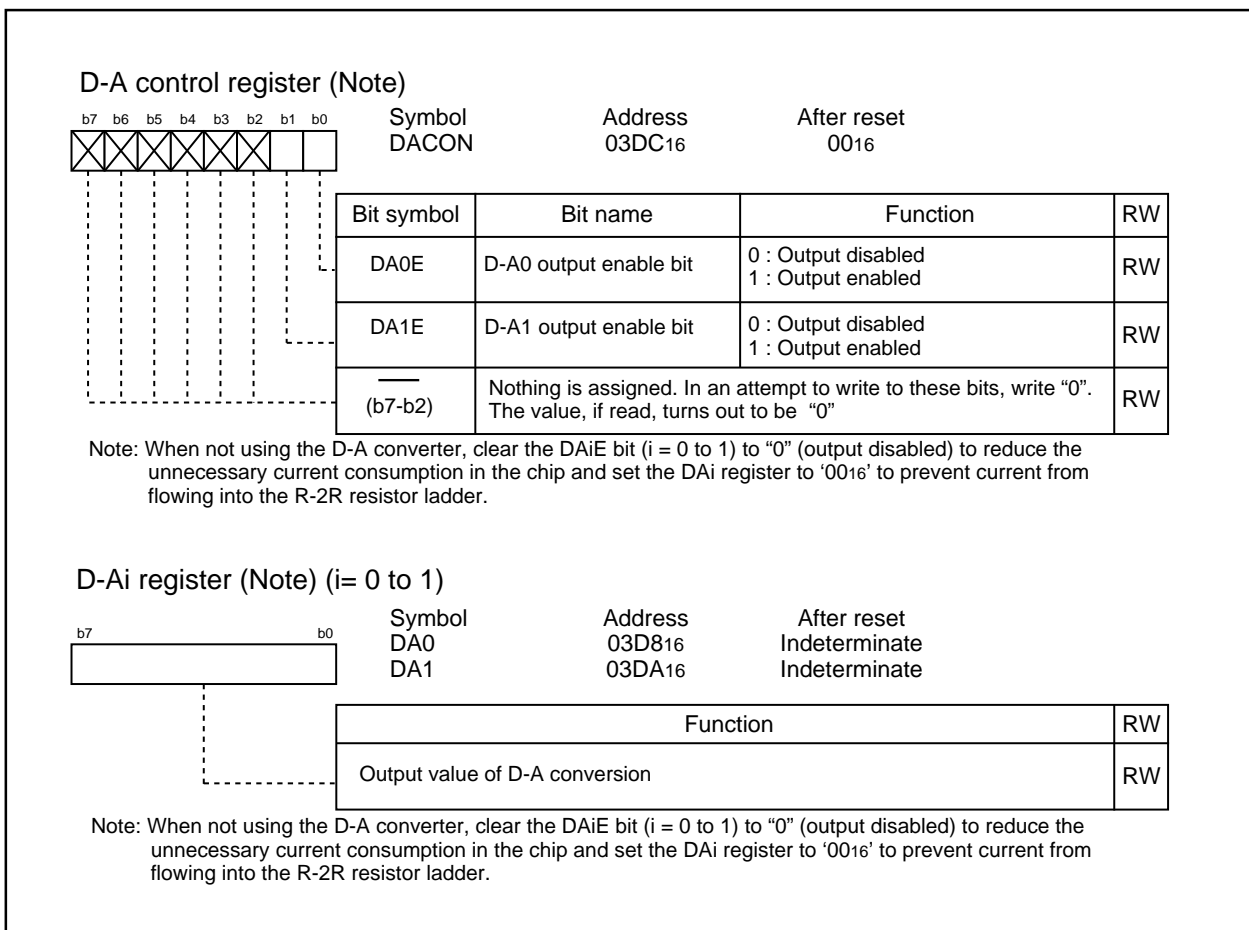
**Table 1.23.1. D-A Converter Performance**

Item	Performance
D-A conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 (DA0 and DA1)

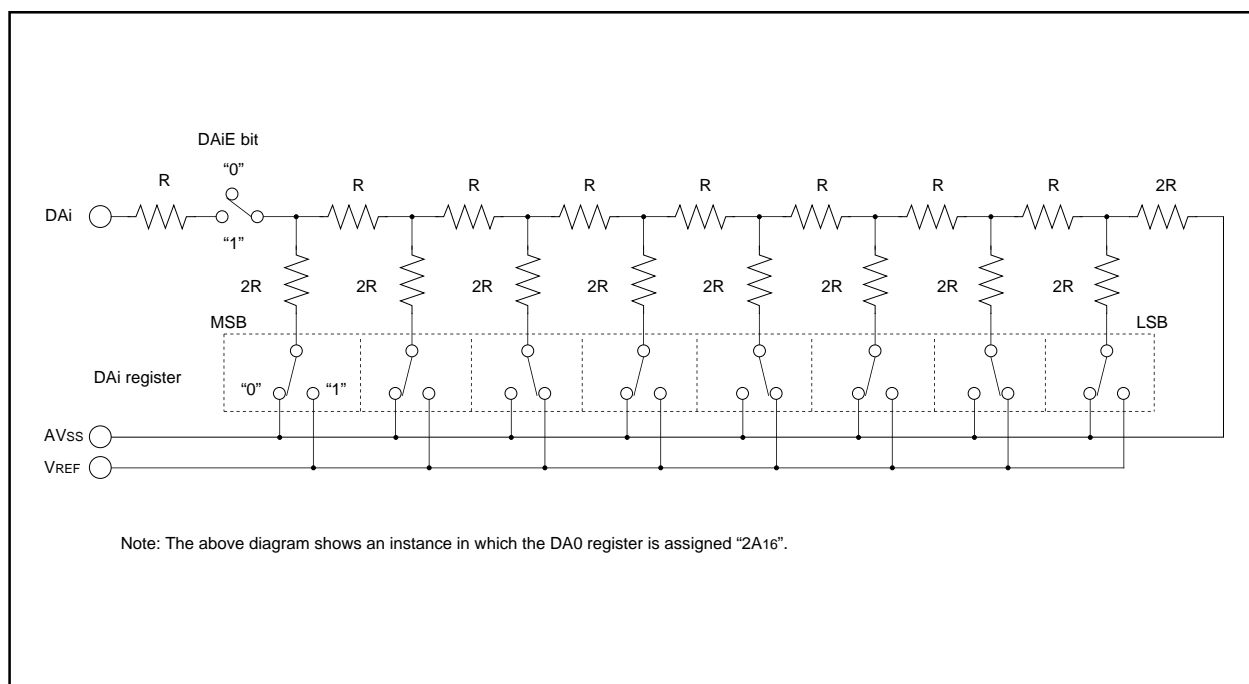


**Figure 1.23.1. D-A Converter Block Diagram**

### D-A Converter



**Figure 1.23.2. DACON Register, DA0 Register, and DA1 Register**



**Figure 1.23.3. D-A Converter Equivalent Circuit**

## CRC Calculation

### 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 1.24.1 shows the block diagram of the CRC circuit. Figure 1.24.2 shows the CRC-related registers. Figure 1.24.3 shows the calculation example using the CRC operation.

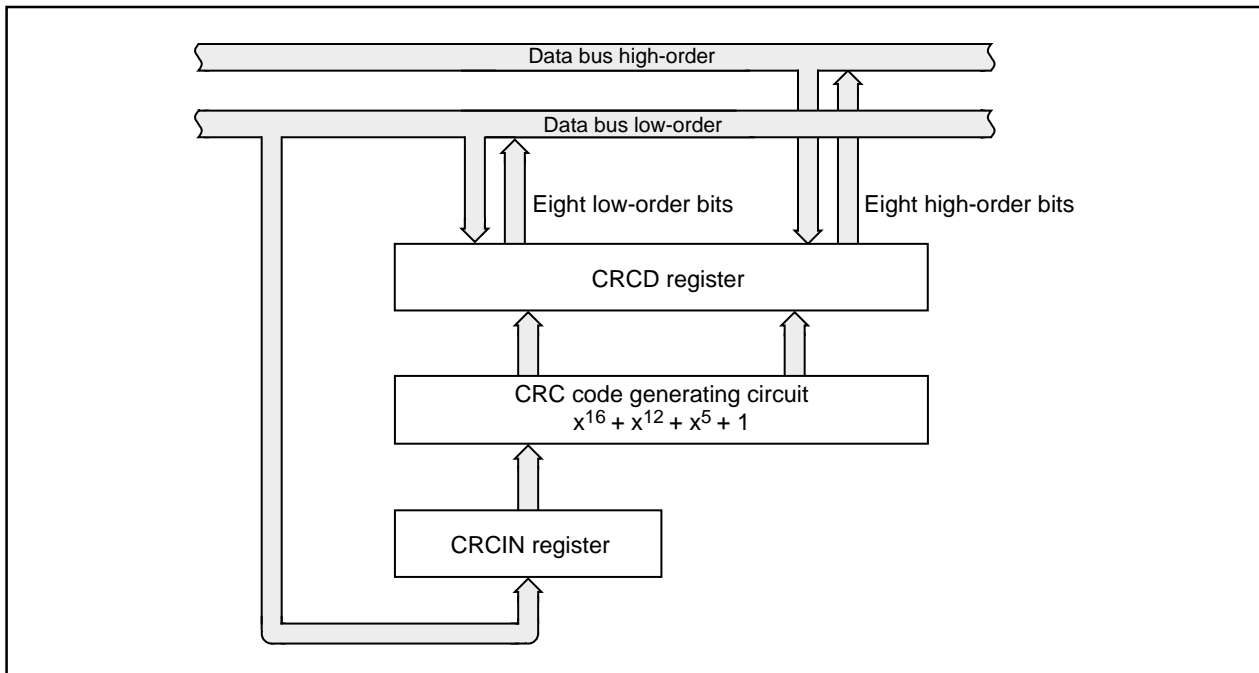


Figure 1.24.1. CRC Circuit Block Diagram

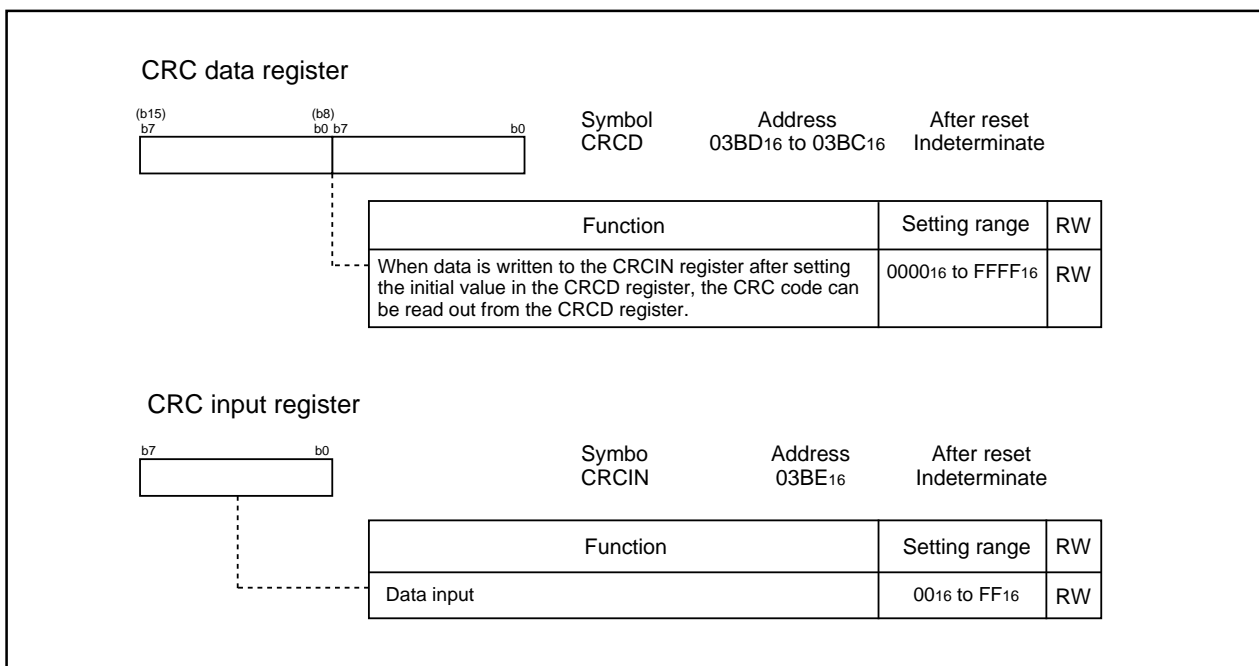


Figure 1.24.2. CRCD Register and CRCIN Register



**CRC Calculation**

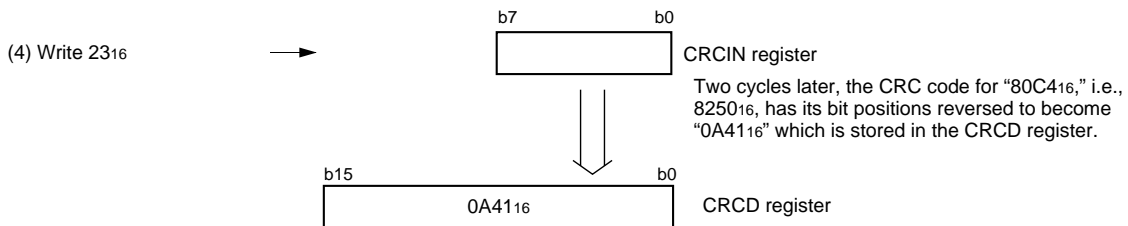
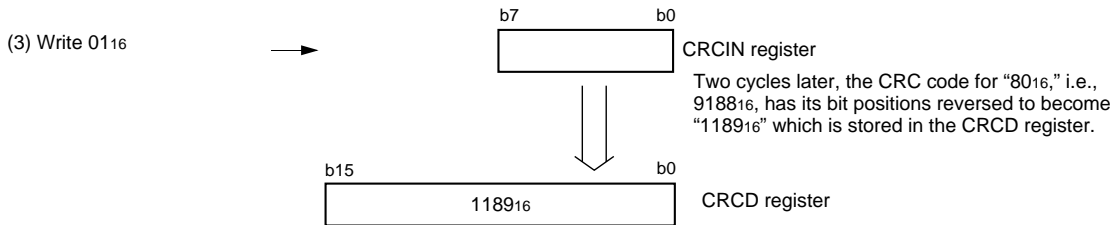
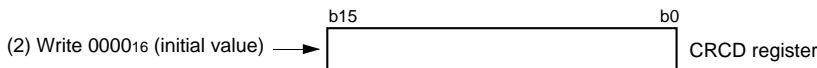
**Setup procedure and CRC operation when generating CRC code "80C4<sub>16</sub>"**

(a) 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 0001<sub>2</sub>)

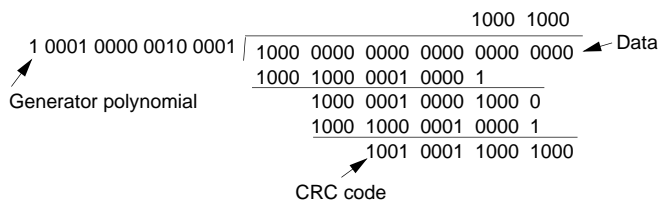
(b) Setting procedure

(1) Reverse the bit positions of the value "80C4<sub>16</sub>" bitwise in a program.  
"80<sub>16</sub>" → "01<sub>16</sub>", "C4<sub>16</sub>" → "23<sub>16</sub>"



(c) Details of CRC operation

In the case of (3) above, the value written to the CRCIN register "01<sub>16</sub> (00000001<sub>2</sub>)" has its bit positions reversed to become "10000002." The value "1000 0000 0000 0000 0000 0002" derived from that by adding 16 digits and the CRCD register's initial value "0000<sub>16</sub>" are added, the result of which is divided by the generator polynomial using modulo-2 arithmetic.



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

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

The value "0001 0001 1000 1001<sub>2</sub> (1189<sub>16</sub>)" derived from the remainder "1001 0001 1000 1000<sub>2</sub> (9188<sub>16</sub>)" by reversing its bit positions may be read from the CRCD register.

If operation (4) above is performed subsequently, the value written to the CRCIN register "23<sub>16</sub> (00100011<sub>2</sub>)" has its bit positions reversed to become "11000100<sub>2</sub>. The value "1100 0100 0000 0000 0000 0000<sub>2</sub>" derived from that by adding 16 digits and the remainder in (3) "1001 0001 1000 1000<sub>2</sub>" which is left in the CRCD register are added, the result of which is divided by the generator polynomial using modulo-2 arithmetic. The value "0000 1010 0100 0001<sub>2</sub> (0A41<sub>16</sub>)" derived from the remainder by reversing its bit positions may be read from the CRCD register.

**Figure 1.24.3. CRC Calculation**

## 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 P85) for the 100-pin version, or 113 lines P0 to P14 (except P85) for the 128-pin version. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high every 4 lines. P85 is an input-only port and does not have a pull-up resistor. Port P85 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 1.25.1 to 1.25.4 show the I/O ports. Figure 1.25.5 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 or D-A converter output pin, set the direction bit for that pin to "0" (input mode). Any pin used as an output pin for peripheral functions other than the D-A converter is directed for output no matter how the corresponding direction bit is set.

When using any pin as a bus control pin, refer to "Bus Control."

P0 to P5, P12, and P13 are capable of VCC2-level input/output; P6 to P11 and P14 are capable of VCC1-level input/output.

### (1) Port Pi Direction Register (PDi Register, i = 0 to 13)

Figure 1.25.6 shows the direction 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, CS0 to CS3, RD, WRL/WR, WRH/BHE, ALE, RDY, HOLD, HLDA, and BCLK) cannot be modified.

No direction register bit for P85 is available.

### (2) Port Pi Register (Pi Register, i = 0 to 13)

Figure 1.25.7 and 1.25.8 show 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 PDi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, CS0 to CS3, RD, WRL/WR, WRH/BHE, ALE, RDY, HOLD, HLDA, and BCLK) cannot be modified.

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

Figure 1.25.9 shows the PUR0 to PUR2 registers.

The PUR0 to PUR2 register bits can be used to select whether or not to pull the corresponding port high in 4 bit units. The port chosen to be pulled high has a pull-up resistor connected to it when the direction bit is set for input mode.

However, the pull-up control register has no effect on P0 to P3, P40 to P43, and P5 during memory extension and microprocessor modes. Although the register contents can be modified, no pull-up resistors are connected.

### (4) Port Control Register

Figure 1.25.10 shows the port control register.

When the P1 register is read after setting the PCR register's PCR0 bit to "1", the corresponding port latch can be read no matter how the PD1 register is set.

Programmable I/O Ports

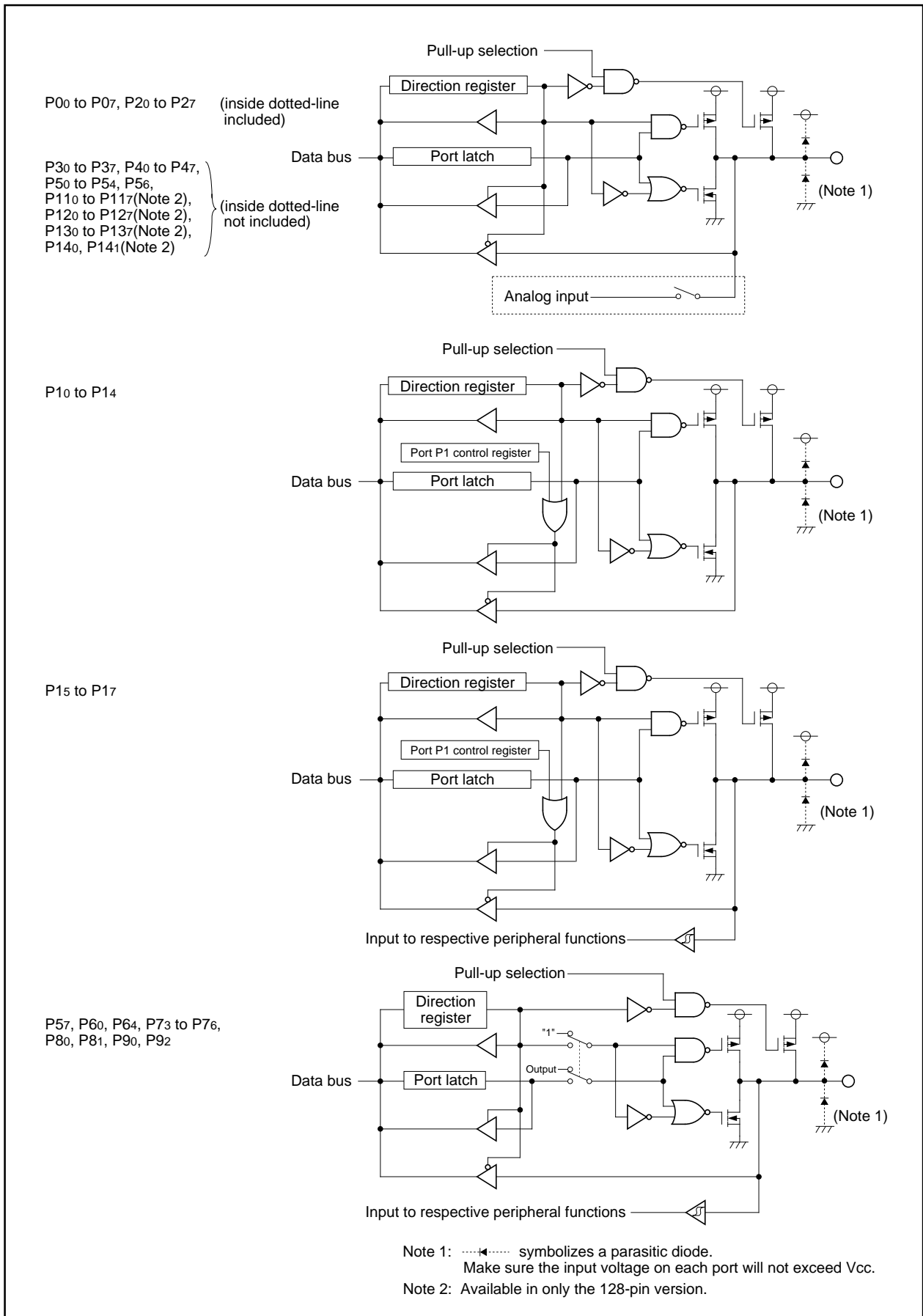


Figure 1.25.1. I/O Ports (1)

Programmable I/O Ports

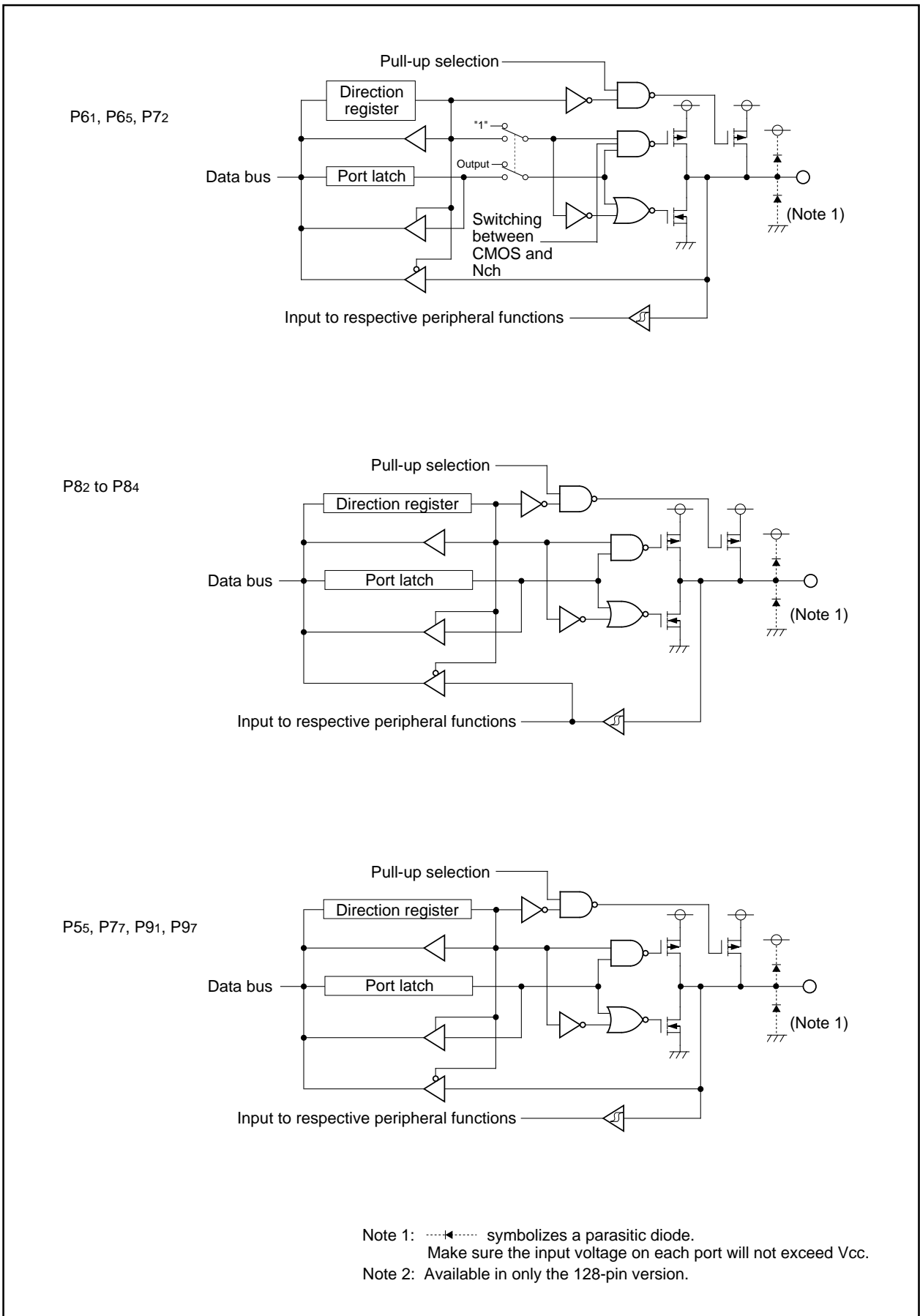


Figure 1.25.2. I/O Ports (2)

Programmable I/O Ports

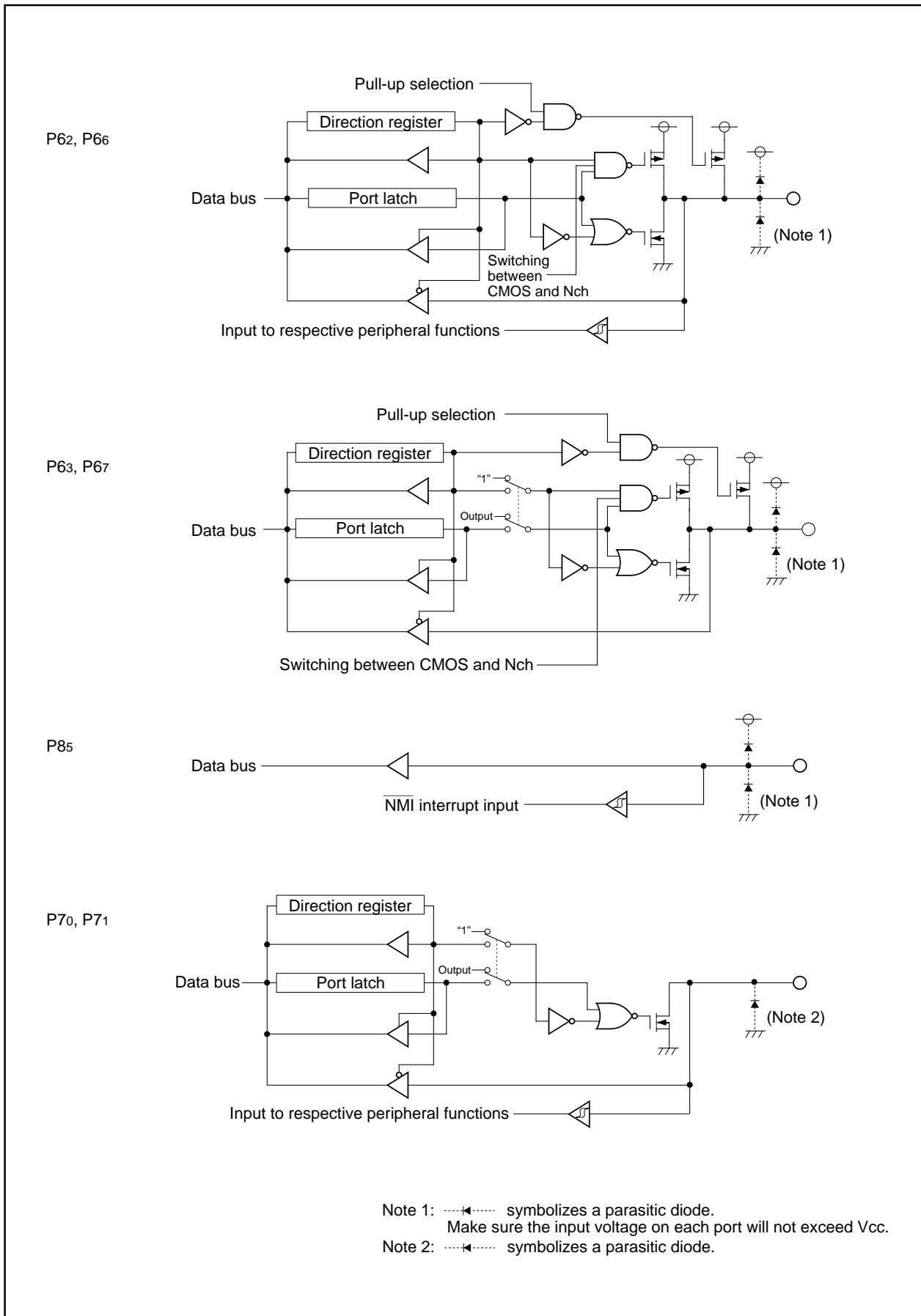


Figure 1.25.3. I/O Ports (3)

Programmable I/O Ports

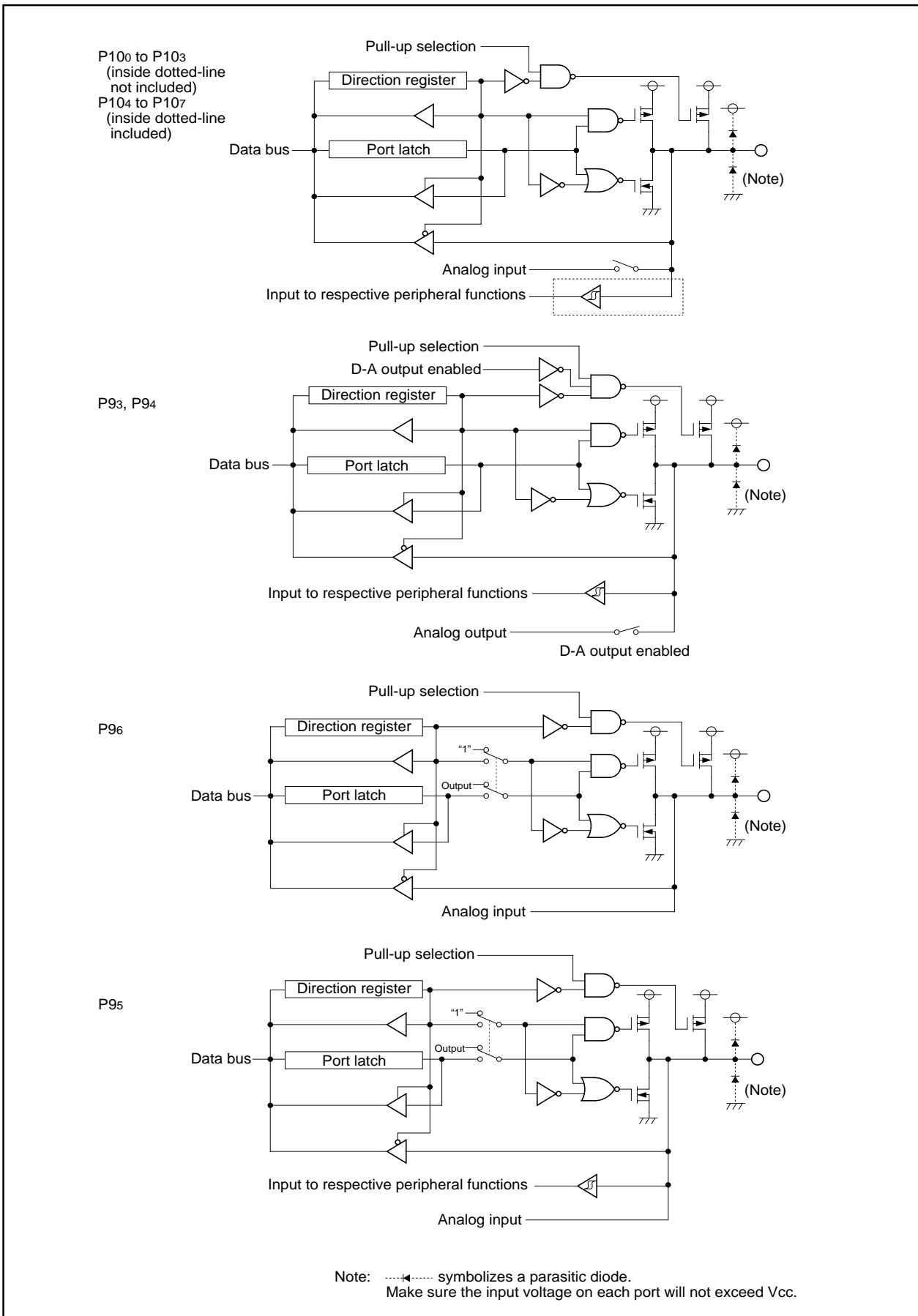


Figure 1.25.4. I/O Ports (4)

Programmable I/O Ports

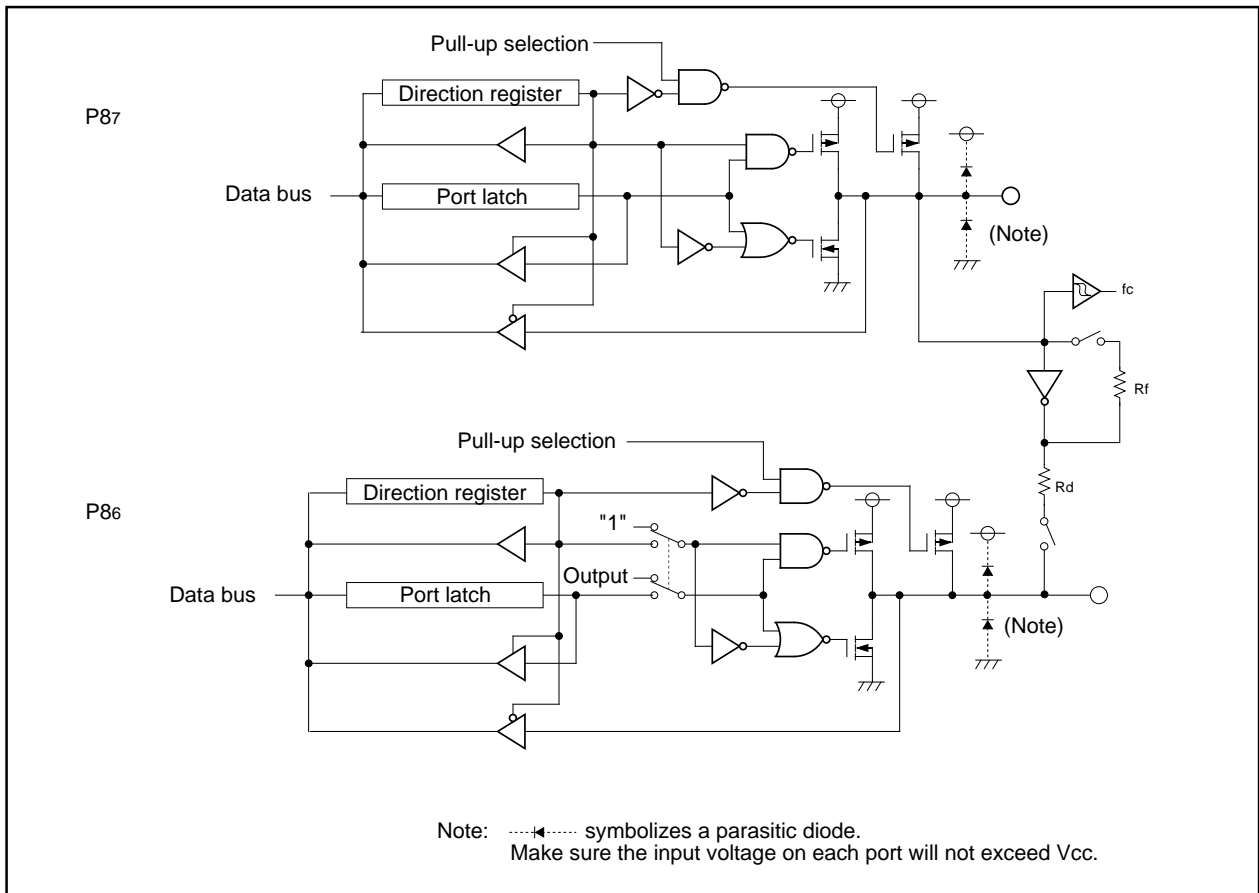


Figure 1.25.5. I/O Ports (5)

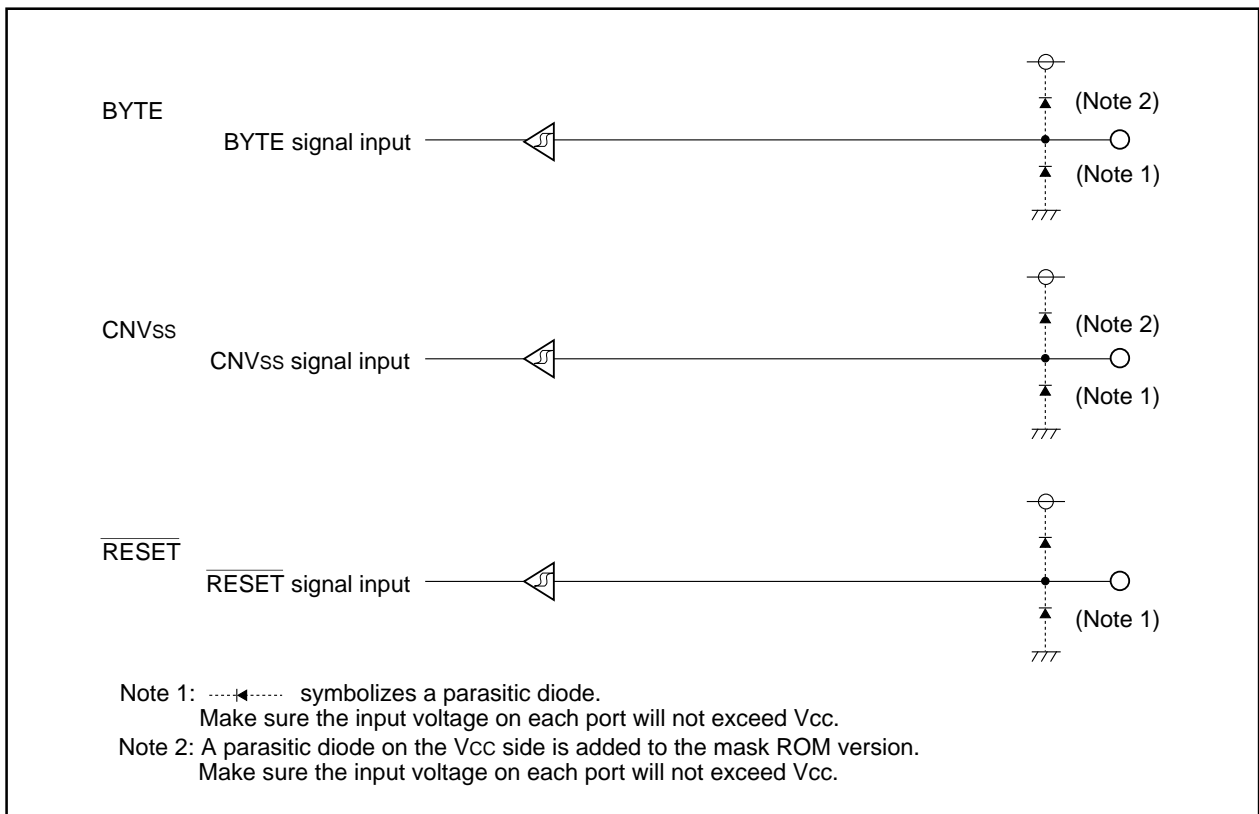


Figure 1.25.6. I/O Pins

Programmable I/O Ports

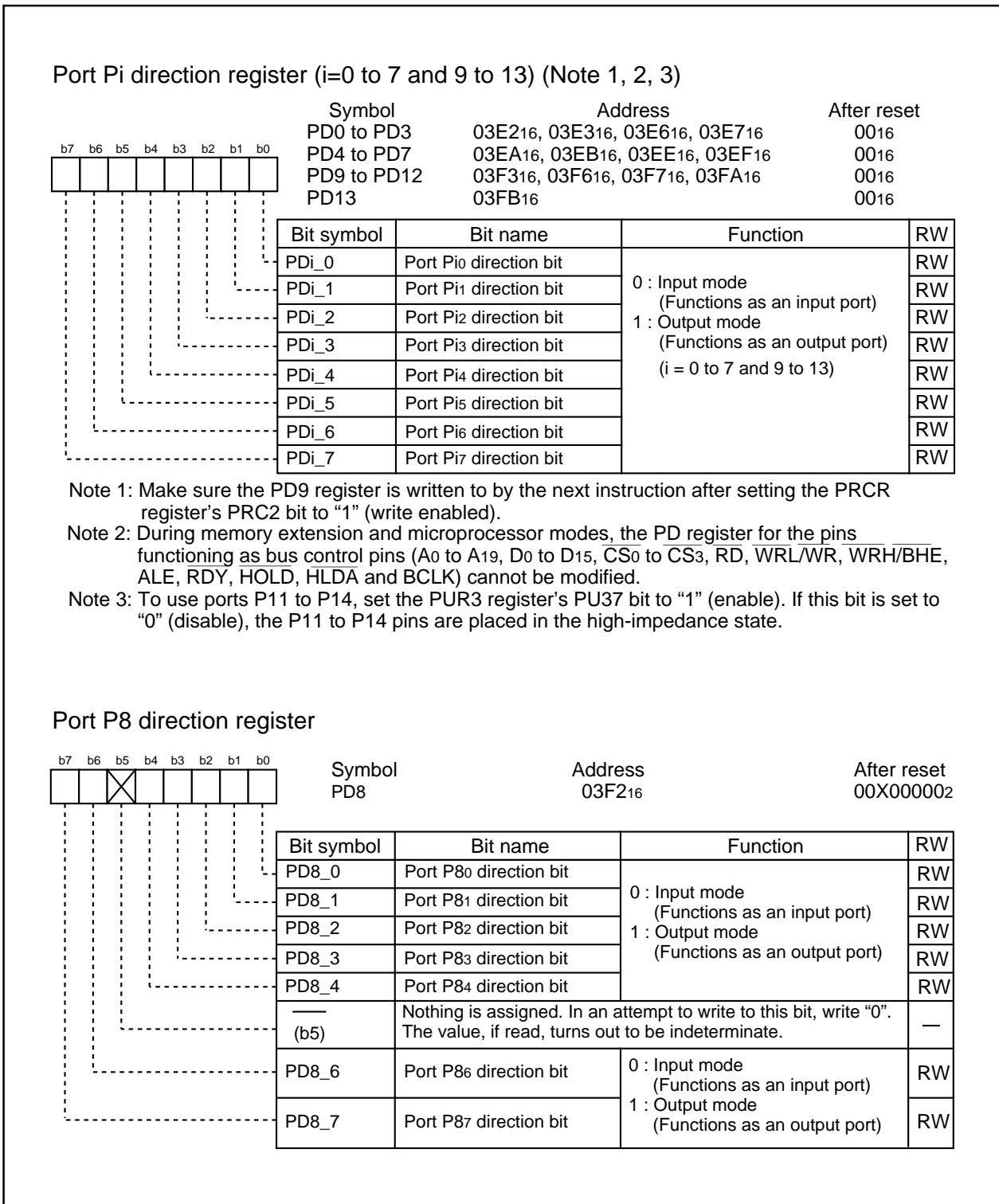


Figure 1.25.7. PD0 to PD13 Registers



Programmable I/O Ports

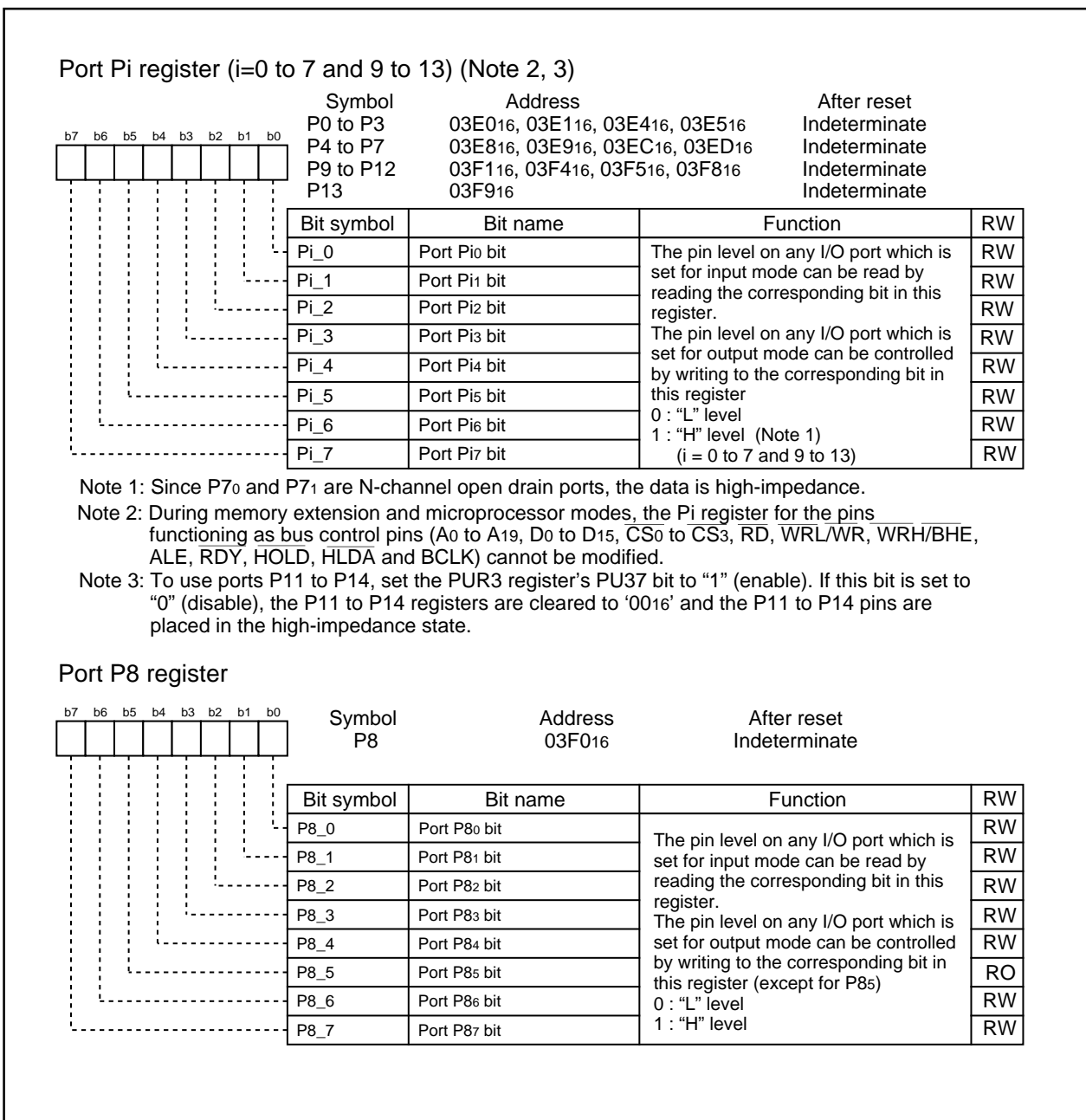


Figure 1.25.8. P0 to P13 Registers

Programmable I/O Ports

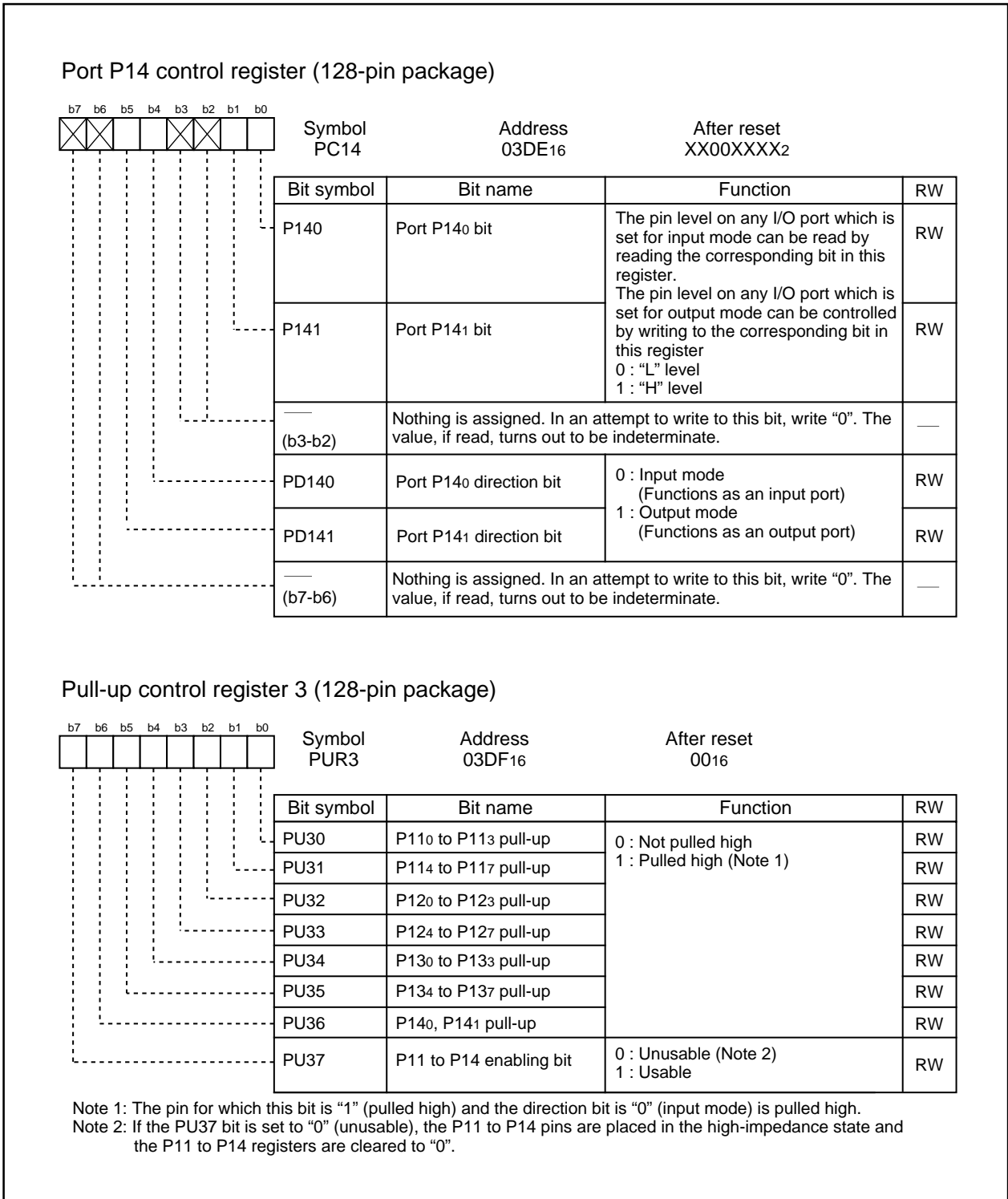


Figure 1.25.9. PC14 Register and PUR3 Register

Programmable I/O Ports

Pull-up control register 0 (Note 1)

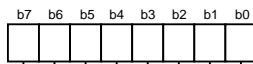


Symbol: PUR0  
 Address: 03FC<sub>16</sub>  
 After reset: 00<sub>16</sub>

Bit symbol	Bit name	Function	RW
PU00	P00 to P03 pull-up	0 : Not pulled high 1 : Pulled high (Note 2)	RW
PU01	P04 to P07 pull-up		RW
PU02	P10 to P13 pull-up		RW
PU03	P14 to P17 pull-up		RW
PU04	P20 to P23 pull-up		RW
PU05	P24 to P27 pull-up		RW
PU06	P30 to P33 pull-up		RW
PU07	P34 to P37 pull-up		RW

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

Pull-up control register 1

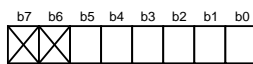


Symbol: PUR1  
 Address: 03FD<sub>16</sub>  
 After reset(Note 5): 00000000<sub>2</sub>  
 00000010<sub>2</sub>

Bit symbol	Bit name	Function	RW
PU10	P40 to P43 pull-up (Note 2)	0 : Not pulled high 1 : Pulled high (Note 3)	RW
PU11	P44 to P47 pull-up (Note 4)		RW
PU12	P50 to P53 pull-up (Note 2)		RW
PU13	P54 to P57 pull-up (Note 2)		RW
PU14	P60 to P63 pull-up		RW
PU15	P64 to P67 pull-up		RW
PU16	P72 to P73 pull-up (Note 1)		RW
PU17	P74 to P77 pull-up		RW

Note 1: The P70 and P71 pins do not have pull-ups.  
 Note 2: During memory extension and microprocessor modes, the pins are not pulled high although the contents of these bits can be modified.  
 Note 3: The pin for which this bit is "1" (pulled high) and the direction bit is "0" (input mode) is pulled high.  
 Note 4: If the PM01 to PM00 bits are set to "012" (memory expansion mode) or "112" (microprocessor mode) in a program during single-chip mode, the PU11 bit becomes "1".  
 Note 5: The values after hardware reset 1 and 2 are as follows:  
 • 00000000<sub>2</sub> when input on CNVss pin is "L"  
 • 00000010<sub>2</sub> when input on CNVss pin is "H"  
 The values after software reset, watchdog timer reset and oscillation stop detection reset are as follows:  
 • 00000000<sub>2</sub> when PM 01 to PM00 bits of PM0 register are "002" (single-chip mode)  
 • 00000010<sub>2</sub> when PM 01 to PM00 bits of PM0 register are "012" (memory expansion mode) or "112" (microprocessor mode)

Pull-up control register 2



Symbol: PUR2  
 Address: 03FE<sub>16</sub>  
 After reset: 00<sub>16</sub>

Bit symbol	Bit name	Function	RW
PU20	P80 to P83 pull-up	0 : Not pulled high 1 : Pulled high (Note 1)	RW
PU21	P84 to P87 pull-up (Note 2)		RW
PU22	P90 to P93 pull-up		RW
PU23	P94 to P97 pull-up		RW
PU24	P100 to P103 pull-up		RW
PU25	P104 to P107 pull-up		RW
— (b7-b6)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

Note 1: The pin for which this bit is "1" (pulled high) and the direction bit is "0" (input mode) is pulled high.  
 Note 2: The P85 pin does not have pull-up.

Figure 1.25.10. PUR0 to PUR2 Registers

Programmable I/O Ports

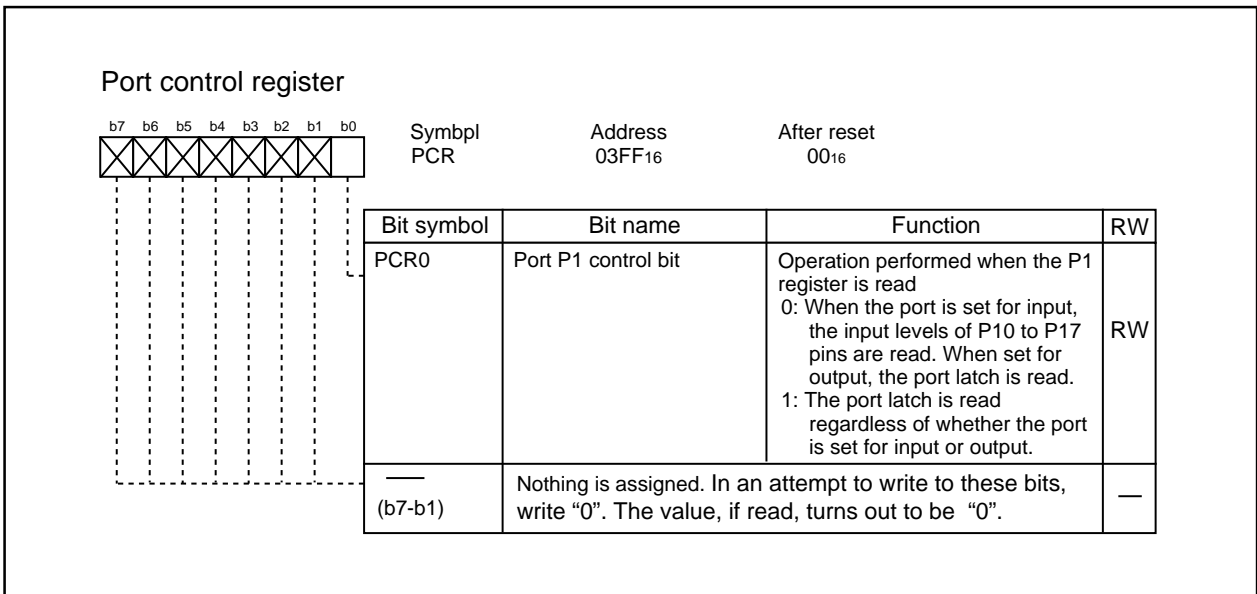


Figure 1.25.11. PCR Register

## Programmable I/O Ports

**Table 1.25.1. Unassigned Pin Handling in Single-chip Mode**

Pin name	Connection
Ports P0 to P14 (excluding P85)	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. (Note 2)
XOUT (Note 1)	Open
$\overline{\text{NMI}}$	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF, BYTE	Connect to VSS

Note 1: With external clock input to XIN pin.

Note 2: When not using all of the P11 to P14, the P11 to P14 pins may be left open by setting the PUR3 register's PU37 bit to "0" (unusable) without causing any problem.

**Table 1.25.2. Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode**

Pin name	Connection
Ports P6 to P10 (excluding P85)	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.
P45 / $\overline{\text{CS1}}$ to P47 / $\overline{\text{CS3}}$	Connect to VCC via a resistor (pulled high) by setting the PD4 register's corresponding direction bit for $\overline{\text{CSi}}$ (i=1 to 3) to "0" (input mode) and the CSR register's $\overline{\text{CSi}}$ bit to "0" (chip select disabled).
$\overline{\text{BHE}}$ , $\overline{\text{ALE}}$ , $\overline{\text{HLDA}}$ , XOUT (Note 1), BCLK (Note 2)	Open
$\overline{\text{HOLD}}$ , $\overline{\text{RDY}}$ , $\overline{\text{NMI}}$	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF	Connect to VSS

Note 1: With external clock input to XIN pin.

Note 2: If the PM0 register's PM07 bit is set to "1" (BCLK not output), connect this pin to VCC via a resistor (pulled high).

Note 3: When not using all of the P11 to P14, the P11 to P14 pins may be left open by setting the PUR3 register's PU37 bit to "0" (unusable) without causing any problem.

Programmable I/O Ports

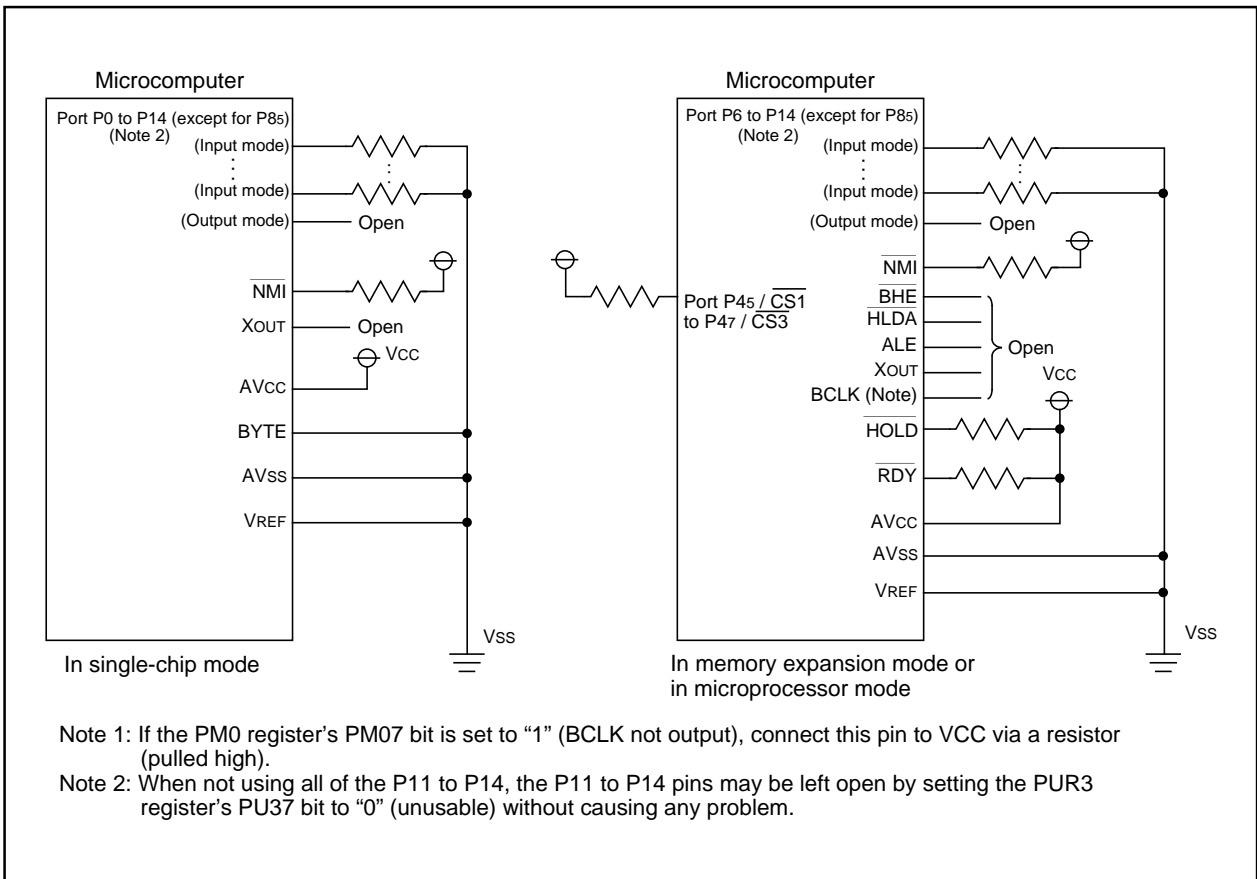


Figure 1.25.12. Unassigned Pins Handling

Electrical Characteristics

## Electrical Characteristics

**Table 1.26.1. Absolute Maximum Ratings**

Symbol	Parameter		Condition	Rated value	Unit
Vcc1, Vcc2	Supply voltage		Vcc1=AVcc	-0.3 to 6.5	V
Vcc2	Supply voltage		Vcc2	-0.3 to Vcc1+0.1	V
AVcc	Analog supply voltage		Vcc1=AVcc	-0.3 to 6.5	V
Vi	Input voltage	RESET, CNVss, BYTE, P60 to P67, P72 to P77, P80 to P87, P90 to P97, P100 to P107, P110 to P117, P140, P141, VREF, XIN		-0.3 to Vcc1+0.3	V
		P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P120 to P127, P130 to P137		-0.3 to Vcc2+0.3	V
		P70, P71		-0.3 to 6.5	V
Vo	Output voltage	P60 to P67, P72 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P110 to P117, P140, P141, XOUT		-0.3 to Vcc1+0.3	V
		P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P120 to P127, P130 to P137		-0.3 to Vcc2+0.3	V
		P70, P71		-0.3 to 6.5	V
Pd	Power dissipation		Topr=25 °C	300	mW
Topr	Operating ambient temperature			-20 to 85 / -40 to 85	°C
Tstg	Storage temperature			-65 to 150	°C

Electrical Characteristics

Table 1.26.2. Recommended Operating Conditions (Note 1)

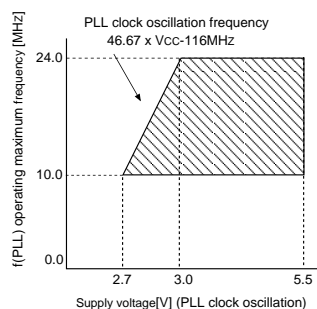
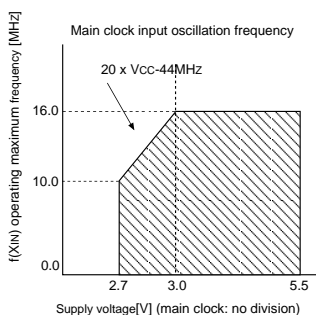
Symbol	Parameter	Standard			Unit	
		Min.	Typ.	Max.		
VCC1, VCC2	Supply voltage(VCC1≥VCC2)	2.7	5.0	5.5	V	
AVcc	Analog supply voltage		VCC1		V	
Vss	Supply voltage		0		V	
AVss	Analog supply voltage		0		V	
VIH	HIGH input voltage	P31 to P37, P40 to P47, P50 to P57, P120 to P127, P130 to P137	0.8VCC2		VCC2	V
		P00 to P07, P10 to P17, P20 to P27, P30 (during single-chip mode)	0.8VCC2		VCC2	V
		P00 to P07, P10 to P17, P20 to P27, P30 (data input function during memory expansion and microprocessor modes)	0.5VCC2		VCC2	V
		P60 to P67, P72 to P77, P80 to P87, P90 to P97, P100 to P107, P110 to P117, P140, P141, XIN, RESET, CNVss, BYTE P70, P71	0.8VCC1		VCC1	V
VIL	LOW input voltage	P31 to P37, P40 to P47, P50 to P57, P120 to P127, P130 to P137	0		0.2VCC2	V
		P00 to P07, P10 to P17, P20 to P27, P30 (during single-chip mode)	0		0.2VCC2	V
		P00 to P07, P10 to P17, P20 to P27, P30 (data input function during memory expansion and microprocessor modes)	0		0.16VCC2	V
		P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P107, P110 to P117, P140, P141, XIN, RESET, CNVss, BYTE	0		0.2VCC1	V
IOH (peak)	HIGH peak output current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P72 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P140, P141			-10.0	mA
IOH (avg)	HIGH average output current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P72 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P140, P141			-5.0	mA
IOL (peak)	LOW peak output current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P140, P141			10.0	mA
IOL (avg)	LOW average output current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P140, P141			5.0	mA
f (XIN)	Main clock input oscillation frequency (Note 4)	VCC=3.0 to 5.5V	0		16	MHz
		VCC=2.7 to 3.0V	0		20 X VCC-44	MHz
f (XCIN)	Sub-clock oscillation frequency			32.768	50	kHz
f (Ring)	Ring oscillation frequency			1		MHz
f (PLL)	PLL clock oscillation frequency (Note 4)	VCC=3.0 to 5.5V	10		24	MHz
		VCC=2.7 to 3.0V	10		46.67 X VCC-116	MHz
f (BCLK)	CPU operation clock		0		24	MHz
Tsu(PLL)	PLL frequency synthesizer stabilization wait time	VCC=5.0V			20	ms
		VCC=3.0V			50	ms

Note 1: Referenced to VCC = VCC1 = VCC2 = 2.7 to 5.5V at Topr = -20 to 85 °C / -40 to 85 °C unless otherwise specified.

Note 2: The mean output current is the mean value within 100ms.

Note 3: The total IOL (peak) for ports P0, P1, P2, P86, P87, P9, P10, P11, P140 and P141 must be 80mA max. The total IOL (peak) for ports P3, P4, P5, P6, P7, P80 to P84, P12, and P13 must be 80mA max. The total IOH (peak) for ports P0, P1, and P2 must be -40mA max. The total IOH (peak) for ports P3, P4, P5, P12, and P13 must be -40mA max. The total IOH (peak) for ports P6, P7, and P80 to P84 must be -40mA max. The total IOH (peak) for ports P86, P87, P9, P10, P11, P140, and P141 must be -40mA max.

Note 4: Relationship between main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.





**Electrical Characteristics**

**Table 1.26.3. A-D Conversion Characteristics (Note 1)**

Symbol	Parameter		Measuring condition	Standard			Unit	
				Min.	Typ.	Max.		
–	Resolution		$V_{REF} = V_{CC1}$			10	Bits	
INL	Integral non-linearity error	10 bit	$V_{REF} = V_{CC1} = 5V$	AN0 to AN7 input ANEX0, ANEX1 input External operation amp connection mode AN00 to AN07 input AN20 to AN27 input			$\pm 3$	LSB
			$V_{REF} = V_{CC1} = 3.3V$	AN0 to AN7 input ANEX0, ANEX1 input External operation amp connection mode AN00 to AN07 input AN20 to AN27 input			$\pm 7$	LSB
		8 bit	$V_{REF} = V_{CC1} = 3.3V$	AN0 to AN7 input ANEX0, ANEX1 input External operation amp connection mode AN00 to AN07 input AN20 to AN27 input			$\pm 5$	LSB
			$V_{REF} = V_{CC1} = 3.3V$	AN0 to AN7 input ANEX0, ANEX1 input External operation amp connection mode AN00 to AN07 input AN20 to AN27 input			$\pm 7$	LSB
DNL	Differential non-linearity error					$\pm 2$	LSB	
–	Offset error					$\pm 3$	LSB	
–	Gain error					$\pm 3$	LSB	
RLADDER	Ladder resistance		$V_{REF} = V_{CC1}$	10		40	k $\Omega$	
tCONV	Conversion time(10bit), Sample & hold function available		$V_{REF} = V_{CC1} = 5V, \phi_{AD} = 10MHz$	3.3			$\mu s$	
tCONV	Conversion time(8bit), Sample & hold function available		$V_{REF} = V_{CC1} = 5V, \phi_{AD} = 10MHz$	2.8			$\mu s$	
tsAMP	Sampling time			0.3			$\mu s$	
VREF	Reference voltage			2.0		$V_{CC1}$	V	
VIA	Analog input voltage			0		$V_{REF}$	V	

Note 1: Referenced to  $V_{CC1} = AV_{CC} = V_{REF} = 3.3$  to  $5.5V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^\circ C$  /  $-40$  to  $85^\circ C$  unless otherwise specified.

Note 2: If  $V_{CC1} > V_{CC2}$ , do not use AN00 to AN07 and AN20 to AN27 as analog input pins.

Note 3: AD operation clock frequency ( $\phi_{AD}$  frequency) must be 10 MHz or less. And divide the  $f_{AD}$  if  $V_{CC1}$  is less than 4.2V, and make  $\phi_{AD}$  frequency equal to or lower than  $f_{AD}/2$ .

Note 4: A case without sample & hold function turn  $\phi_{AD}$  frequency into 250 kHz or more in addition to a limit of Note 3.  
A case with sample & hold function turn  $\phi_{AD}$  frequency into 1MHz or more in addition to a limit of Note 3.

**Table 1.26.4. D-A Conversion Characteristics (Note 1)**

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy				1.0	%
t <sub>su</sub>	Setup time				3	$\mu s$
R <sub>O</sub>	Output resistance		4	10	20	k $\Omega$
I <sub>VREF</sub>	Reference power supply input current	(Note 2)			1.5	mA

Note 1: Referenced to  $V_{CC1} = V_{REF} = 3.3$  to  $5.5V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^\circ C$  /  $-40$  to  $85^\circ C$  unless otherwise specified.

Note 2: This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "0016". The A-D converter's ladder resistance is not included. Also, when D-A register contents are not "0016", the current I<sub>VREF</sub> always flows even though V<sub>ref</sub> may have been set to be unconnected by the A-D control register.

**Table 1.26.5. Flash Memory Version Electrical Characteristics (Note 1)**

Parameter	Standard			Unit
	Min.	Typ.	Max	
Word program time		30	200	$\mu s$
Block erase time		1	4	s
Erase all unlocked blocks time		1 X n	4 X n	s
Lock bit program time		30	200	$\mu s$

Note 1: Referenced to  $V_{CC1} = 4.5$  to  $5.5V$ ,  $3.0$  to  $3.6V$  at  $T_{opr} = 0$  to  $60^\circ C$  unless otherwise specified.

Note 2: n denotes the number of block erases.

**Table 1.26.6. Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics (at  $T_{opr} = 0$  to  $60^\circ C$ )**

Flash program, erase voltage	Flash read operation voltage
$V_{CC1} = 3.3V \pm 0.3V$ or $5.0V \pm 0.5V$	$V_{CC1} = 2.7$ to $5.5V$

**Electrical Characteristics**

**Table 1.26.7. Low Voltage Detection Circuit Electrical Characteristics (Note 1)**

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
Vdet4	Power supply down detection voltage (Notes 1, 2)	V <sub>CC1</sub> =0.8 to 5.5V	3.3	3.8	4.4	V
Vdet3	Reset level detection voltage (Notes 1, 2)		2.2	2.8	3.6	V
Vdet3s	Low voltage reset retention voltage		0.8			V
Vdet3r	Low voltage reset release voltage (Note 3)		2.2	2.9	4.0	V
Vdet2	RAM retention limit detection voltage (Notes 1, 2)		1.4	2.0	2.7	V

Note 1: Vdet4 > Vdet3 > Vdet2

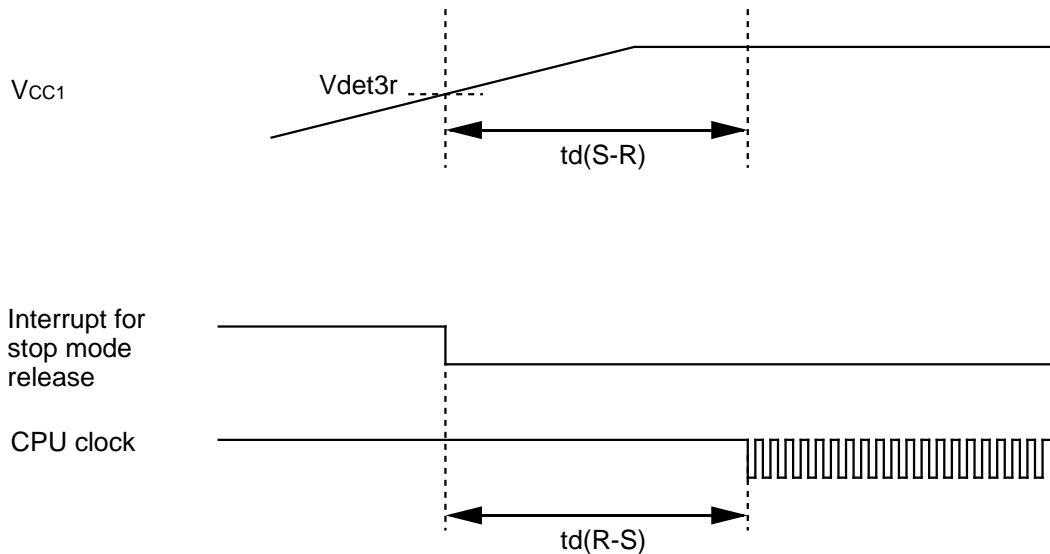
Note 2: Where reset level detection voltage is less than 2.7 V, if the supply power voltage is greater than the reset level detection voltage, the operation at f(BCLK) ≤ 10MHz is guaranteed.

Note 3: Vdet3r > Vdet3 is not guaranteed.

**Table 1.26.8. Power Supply Circuit Timing Characteristics**

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Time for internal power supply stabilization during powering-on	V <sub>CC1</sub> =2.7 to 5.5V			2	ms
td(R-S)	STOP release time				150	μs
td(M-L)	Time for internal power supply stabilization when main clock oscillation starts				50	μs
td(S-R)	Hardware reset 2 release wait time	V <sub>CC1</sub> =Vdet3r to 5.5V		6 (Note)	20	ms
td(E-A)	Low voltage detection circuit operation start time	V <sub>CC1</sub> =2.7 to 5.5V			20	μs

Note : When V<sub>CC1</sub> = 5V



Electrical Characteristics (Vcc1 = Vcc2 = 5V)

VCC1 = VCC2 = 5V

Table 1.26.9. Electrical Characteristics (Note 1)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
VOH	HIGH output voltage	P60 to P67,P72 to P77,P80 to P84,P86,P87,P90 to P97, P100 to P107,P110 to P117,P140,P141	IOH=-5mA	Vcc1-2.0		Vcc1	V
		P00 to P07,P10 to P17,P20 to P27,P30 to P37, P40 to P47,P50 to P57,P120 to P127,P130 to P137	IOH=-5mA(Note 2)	Vcc2-2.0		Vcc2	
VOH	HIGH output voltage	P60 to P67,P72 to P77,P80 to P84,P86,P87,P90 to P97, P100 to P107,P110 to P117,P140,P141	IOH=-200μA	Vcc1-0.3		Vcc1	V
		P00 to P07,P10 to P17,P20 to P27,P30 to P37, P40 to P47,P50 to P57,P120 to P127,P130 to P137	IOH=-200μA(Note 2)	Vcc2-0.3		Vcc2	
VOH	HIGH output voltage	XOUT	HIGHPOWER	IOH=-1mA		Vcc1-2.0	V
			LOWPOWER	IOH=-0.5mA		Vcc1-2.0	
	HIGH output voltage	XCOUT	HIGHPOWER	With no load applied		2.5	V
			LOWPOWER	With no load applied		1.6	
VOL	LOW output voltage	P60 to P67,P70 to P77,P80 to P84,P86,P87,P90 to P97, P100 to P107,P110 to P117,P140,P141	IOL=5mA			2.0	V
		P00 to P07,P10 to P17,P20 to P27,P30 to P37, P40 to P47,P50 to P57,P120 to P127,P130 to P137	IOL=5mA(Note 2)			2.0	
VOL	LOW output voltage	P60 to P67,P70 to P77,P80 to P84,P86,P87,P90 to P97, P100 to P107,P110 to P117,P140,P141	IOL=200μA			0.45	V
		P00 to P07,P10 to P17,P20 to P27,P30 to P37, P40 to P47,P50 to P57,P120 to P127,P130 to P137	IOL=200μA(Note 2)			0.45	
VOL	LOW output voltage	XOUT	HIGHPOWER	IOL=1mA		2.0	V
			LOWPOWER	IOL=0.5mA		2.0	
	LOW output voltage	XCOUT	HIGHPOWER	With no load applied		0	V
			LOWPOWER	With no load applied		0	
VT+-VT-	Hysteresis	HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT5, NMI, ADTRG, CTS0 to CTS2, SCL, SDA, CLK0 to CLK4,TA2OUT to TA4OUT, Kl0 to Kl3, RxD0 to RxD2, SIn3, SIn4		0.2		1.0	V
VT+-VT-	Hysteresis	RESET		0.2		2.2	V
IiH	HIGH input current	P00 to P07,P10 to P17,P20 to P27,P30 to P37, P40 to P47,P50 to P57,P60 to P67,P70 to P77, P80 to P87,P90 to P97,P100 to P107,P110 to P117, P120 to P127,P130 to P137,P140,P141, XIN, RESET, CNVss, BYTE	Vi=5V			5.0	μA
IiL	LOW input current	P00 to P07,P10 to P17,P20 to P27,P30 to P37, P40 to P47,P50 to P57,P60 to P67,P70 to P77, P80 to P87,P90 to P97,P100 to P107,P110 to P117, P120 to P127,P130 to P137,P140,P141, XIN, RESET, CNVss, BYTE	Vi=0V			-5.0	μA
RPULLUP	Pull-up resistance	P00 to P07,P10 to P17,P20 to P27,P30 to P37, P40 to P47,P50 to P57,P60 to P67,P72 to P77, P80 to P84,P86,P87,P90 to P97,P100 to P107, P110 to P117,P120 to P127,P130 to P137,P140,P141	Vi=0V	30	50	170	kΩ
RixIN	Feedback resistance	XIN				1.5	MΩ
RixCIN	Feedback resistance	XCIN				15	MΩ
Vram	RAM retention voltage		At stop mode	2.0			V

Note 1: Referenced to Vcc=Vcc1=Vcc2=4.2 to 5.5V, Vss=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=24MHz unless otherwise specified.

Note 2: Where the product is used at Vcc1 = 5 V and Vcc2 = 3 V, refer to the 3 V version value for the pin specified value on the Vcc2 port side.

$$V_{CC1} = V_{CC2} = 5V$$

Table 1.26.10. Electrical Characteristics (2) (Note 1)

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
I <sub>CC</sub>	Power supply current (V <sub>CC</sub> =4.0 to 5.5V)	In single-chip mode, the output pins are open and other pins are V <sub>SS</sub>	Mask ROM	f(BCLK)=24MHz, No division, PLL operation		14	20	mA
				No division, Ring oscillation		1		mA
			Flash memory	f(BCLK)=24MHz, No division, PLL operation		18	27	mA
				No division, Ring oscillation		1.8		mA
			Flash memory Program	f(BCLK)=10MHz, V <sub>CC</sub> =5.0V		15		mA
			Flash memory Erase	f(BCLK)=10MHz, V <sub>CC</sub> =5.0V		25		mA
			Mask ROM	f(X <sub>CIN</sub> )=32kHz, Low power dissipation mode, ROM(Note 3)		25		μA
			Flash memory	f(BCLK)=32kHz, Low power dissipation mode, RAM(Note 3)		25		μA
				f(BCLK)=32kHz Low power dissipation mode, Flash memory(Note 3)		420		μA
				Ring oscillation, Wait mode		50		μA
			Mask ROM Flash memory	f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity High		7.5		μA
f(BCLK)=32kHz, Wait mode(Note 2), Oscillation capacity Low		2.0			μA			
Stop mode, T <sub>OPR</sub> =25°C		0.8		3.0	μA			
I <sub>det4</sub>	Power supply down detection dissipation current (Note 4)				0.7	4	μA	
I <sub>det3</sub>	Reset area detection dissipation current (Note 4)				1.2	8	μA	
I <sub>det2</sub>	RAM retention limit detection dissipation current (Note 4)				1.1	6	μA	

Note 1: Referenced to V<sub>CC</sub>=V<sub>CC1</sub>=V<sub>CC2</sub>=4.2 to 5.5V, V<sub>SS</sub>=0V at T<sub>OPR</sub> = -20 to 85 °C / -40 to 85 °C, f(BCLK)=24MHz unless otherwise specified.

Note 2: With one timer operated using fc32.

Note 3: This indicates the memory in which the program to be executed exists.

Note 4: I<sub>det</sub> is dissipation current when the following bit is set to "1" (detection circuit enabled).

I<sub>det4</sub>: VC27 bit of VCR2 register

I<sub>det3</sub>: VC26 bit of VCR2 register

I<sub>det2</sub>: VC25 bit of VCR2 register

$$V_{CC1} = V_{CC2} = 5V$$

### Timing Requirements

(V<sub>CC1</sub> = V<sub>CC2</sub> = 5V, V<sub>SS</sub> = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

**Table 1.26.11. External Clock Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c</sub>	External clock input cycle time	62.5		ns
t <sub>w(H)</sub>	External clock input HIGH pulse width	25		ns
t <sub>w(L)</sub>	External clock input LOW pulse width	25		ns
t <sub>r</sub>	External clock rise time		15	ns
t <sub>f</sub>	External clock fall time		15	ns

**Table 1.26.12. Memory Expansion Mode and Microprocessor Mode**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>ac1</sub> (RD-DB)	Data input access time (for setting with no wait)		(Note 1)	ns
t <sub>ac2</sub> (RD-DB)	Data input access time (for setting with wait)		(Note 2)	ns
t <sub>ac3</sub> (RD-DB)	Data input access time (when accessing multiplex bus area)		(Note 3)	ns
t <sub>su</sub> (DB-RD)	Data input setup time	40		ns
t <sub>su</sub> (RDY-BCLK)	RDY input setup time	30		ns
t <sub>su</sub> (HOLD-BCLK)	HOLD input setup time	40		ns
t <sub>h</sub> (RD-DB)	Data input hold time	0		ns
t <sub>h</sub> (BCLK -RDY)	RDY input hold time	0		ns
t <sub>h</sub> (BCLK-HOLD)	HOLD input hold time	0		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time		40	ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 45 \quad [\text{ns}]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 45 \quad [\text{ns}] \quad \text{n is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 45 \quad [\text{ns}] \quad \text{n is "2" for 2-wait setting, "3" for 3-wait setting.}$$

$$V_{CC1} = V_{CC2} = 5V$$

**Timing Requirements****( $V_{CC1} = V_{CC2} = 5V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)****Table 1.26.13. 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 1.26.14. 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 1.26.15. 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 1.26.16. 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 1.26.17. 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 1.26.18. 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  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)

**Table 1.26.19. 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 1.26.20. 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 1.26.21. 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 1.26.22. A-D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	ADTRG input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	ADTRG input LOW pulse width	125		ns

**Table 1.26.23. Serial I/O**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

**Table 1.26.24. External Interrupt  $\overline{INT}_i$  Input**

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

Electrical Characteristics (Vcc1 = Vcc2 = 5V)

VCC1 = VCC2 = 5V

Switching Characteristics

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = - 20 to 85°C / - 40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.25. Memory Expansion and Microprocessor Modes (for setting with no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 1.26.1		25	ns
th(BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
th(RD-AD)	Address output hold time (refers to RD)		0		ns
th(WR-AD)	Address output hold time (refers to WR)		(Note 2)		ns
td(BCLK-CS)	Chip select output delay time			25	ns
th(BCLK-CS)	Chip select output hold time (refers to BCLK)		4		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			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 (refers to BCLK)			40	ns
th(BCLK-DB)	Data output hold time (refers to BCLK)		4		ns
td(DB-WR)	Data output delay time (refers to WR)		(Note 1)		ns
th(WR-DB)	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \quad [\text{ns}]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} \quad [\text{ns}]$$

Note 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

$$t = -CR \times \ln(1 - V_{OL} / V_{CC2})$$

by a circuit of the right figure.

For example, when  $V_{OL} = 0.2V_{CC2}$ ,  $C = 30\text{pF}$ ,  $R = 1\text{k}\Omega$ , hold time of output "L" level is

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2V_{CC2} / V_{CC2}) = 6.7\text{ns}.$$

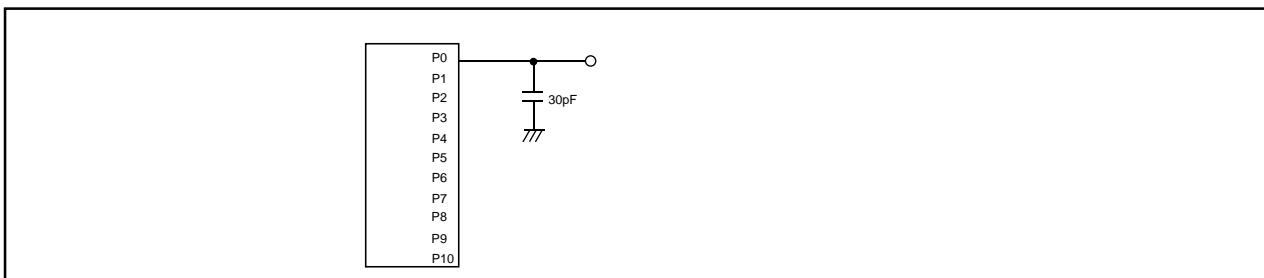
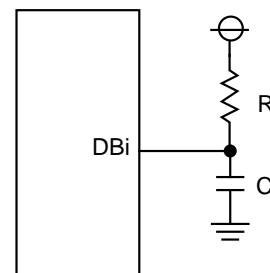


Figure 1.26.1. Ports P0 to P10 Measurement Circuit



Electrical Characteristics (Vcc1 = Vcc2 = 5V)

VCC1 = VCC2 = 5V

Switching Characteristics

(VCC1 = VCC2 = 5V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.26. Memory Expansion and Microprocessor Modes  
(for 1- to 3-wait setting and external area access)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 1.26.1		25	ns
th(BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
th(RD-AD)	Address output hold time (refers to RD)		0		ns
th(WR-AD)	Address output hold time (refers to WR)		(Note 2)		ns
td(BCLK-CS)	Chip select output delay time			25	ns
th(BCLK-CS)	Chip select output hold time (refers to BCLK)		4		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			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 (refers to BCLK)			40	ns
th(BCLK-DB)	Data output hold time (refers to BCLK)		4		ns
td(DB-WR)	Data output delay time (refers to WR)		(Note 1)		ns
th(WR-DB)	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 40 \quad [\text{ns}] \quad n \text{ is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting.}$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} \quad [\text{ns}]$$

Note 3: This standard value shows the timing when the output is off, and does not show hold time of data bus.

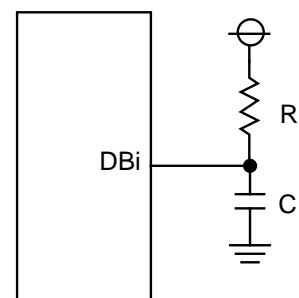
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.

Hold time of data bus is expressed in  
 $t = -CR \times \ln(1 - VOL / VCC2)$

by a circuit of the right figure.

For example, when VOL = 0.2Vcc2, C = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2V_{CC2} / V_{CC2}) = 6.7\text{ns.}$$



Electrical Characteristics ( $V_{CC1} = V_{CC2} = 5V$ )

$$V_{CC1} = V_{CC2} = 5V$$

**Switching Characteristics****( $V_{CC1} = V_{CC2} = 5V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$ ,  $CM15="1"$  unless otherwise specified)****Table 1.26.27. Memory Expansion and Microprocessor Modes**  
**(for 2- to 3-wait setting, external area access and multiplex bus selection)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.26.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (refers to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (refers to RD)		(Note 1)		ns
$t_{h(WR-AD)}$	Address output hold time (refers to WR)		(Note 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (refers to BCLK)		4		ns
$t_{h(RD-CS)}$	Chip select output hold time (refers to RD)		(Note 1)		ns
$t_{h(WR-CS)}$	Chip select output hold time (refers to WR)		(Note 1)		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 (refers to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (refers to BCLK)		4		ns
$t_{d(DB-WR)}$	Data output delay time (refers to WR)		(Note 2)		ns
$t_{h(WR-DB)}$	Data output hold time (refers to WR)		(Note 1)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (refers to BCLK)			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (refers to BCLK)		-4		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (refers to Address)		(Note 3)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (refers to Address)		30		ns
$t_{d(AD-RD)}$	RD signal output delay from the end of Address	0		ns	
$t_{d(AD-WR)}$	WR signal output delay from the end of Address	0		ns	
$t_{dZ(RD-AD)}$	Address output floating start time		8	ns	

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} \quad [\text{ns}]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} \quad -40 \quad [\text{ns}] \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} \quad -25 \quad [\text{ns}]$$

Electrical Characteristics ( $V_{CC1} = V_{CC2} = 5V$ )

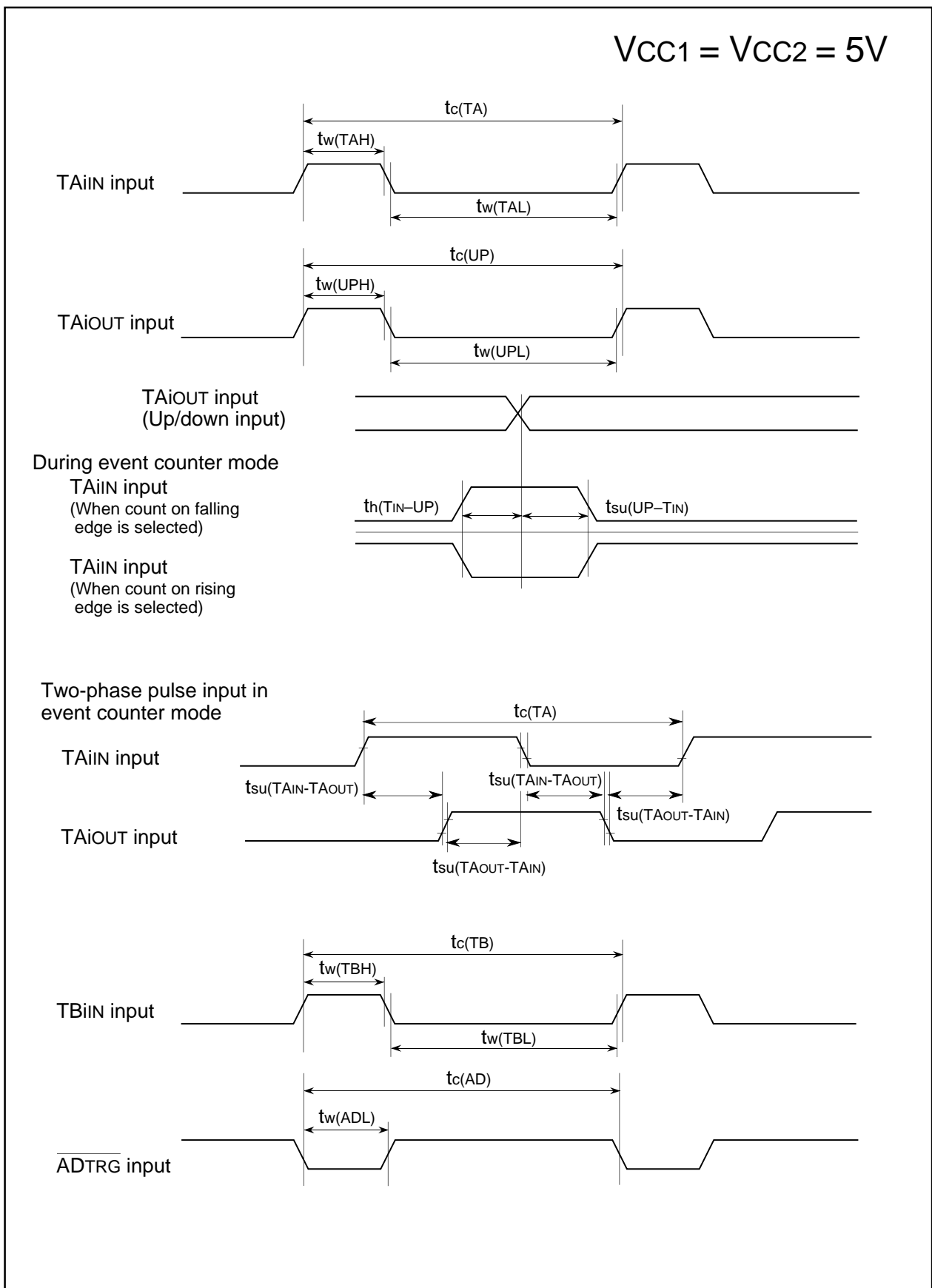


Figure 1.26.2. Timing Diagram (1)

Electrical Characteristics ( $V_{CC1} = V_{CC2} = 5V$ )

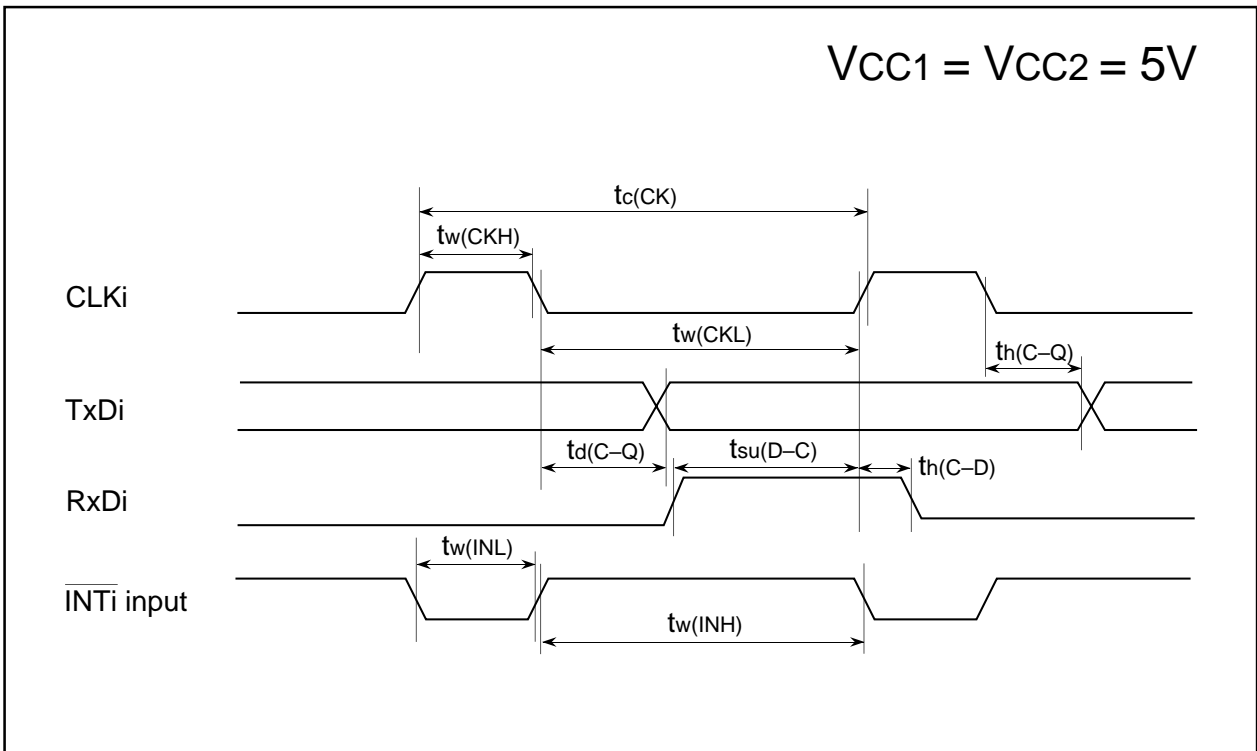


Figure 1.26.3. Timing Diagram (2)

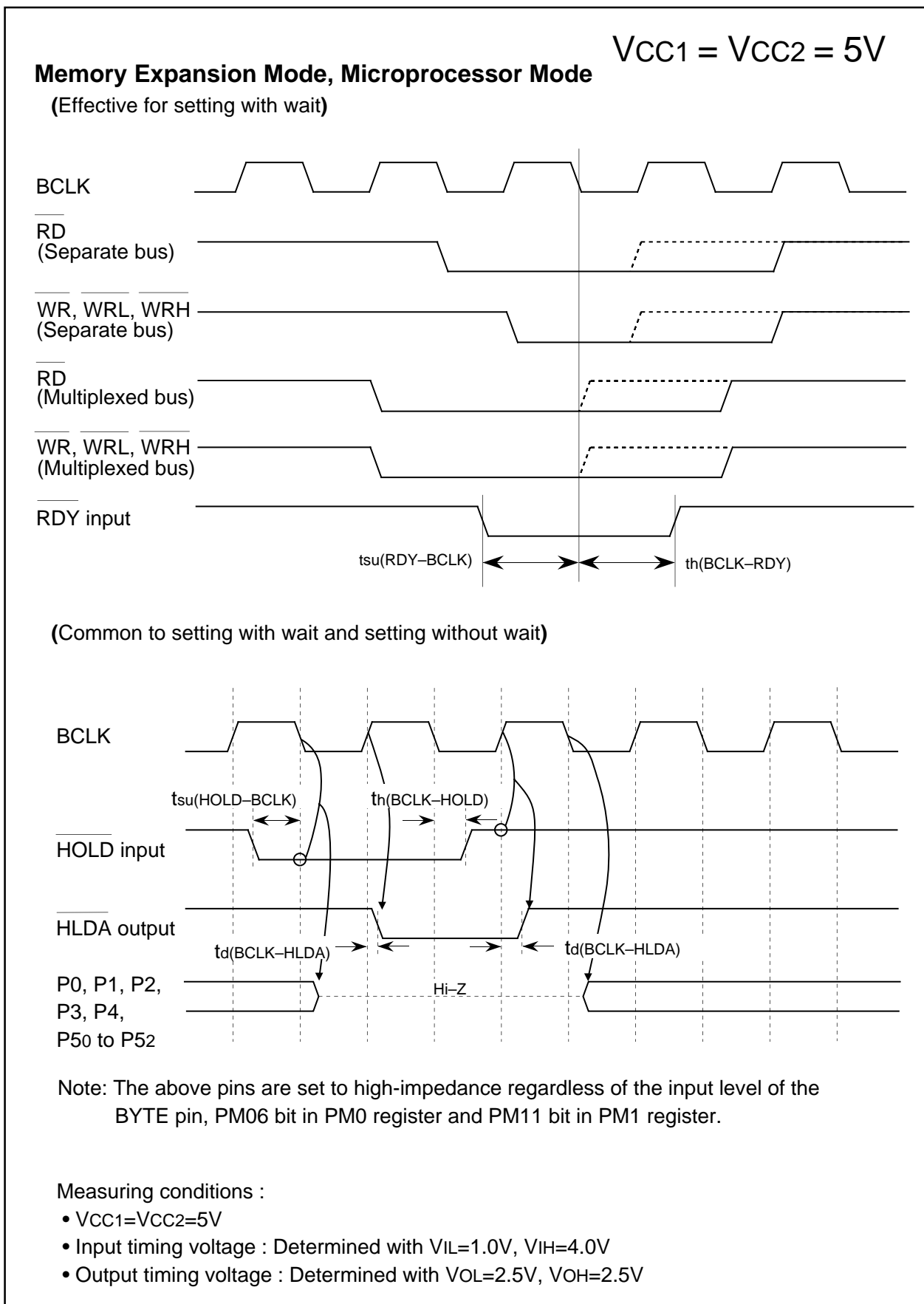


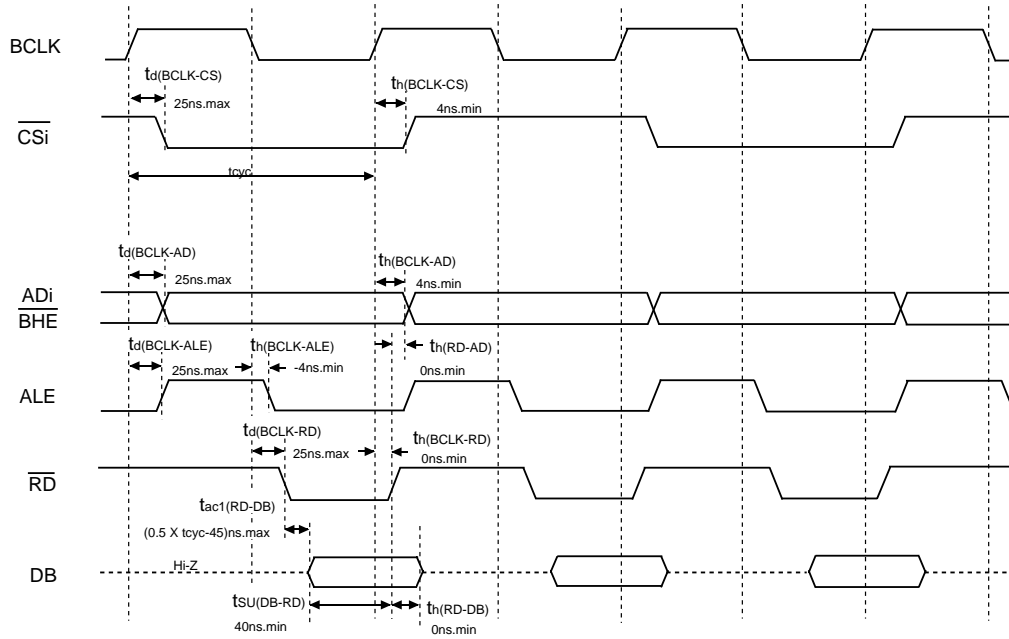
Figure 1.26.4. Timing Diagram (3)

Electrical Characteristics (Vcc1 = Vcc2 = 5V)

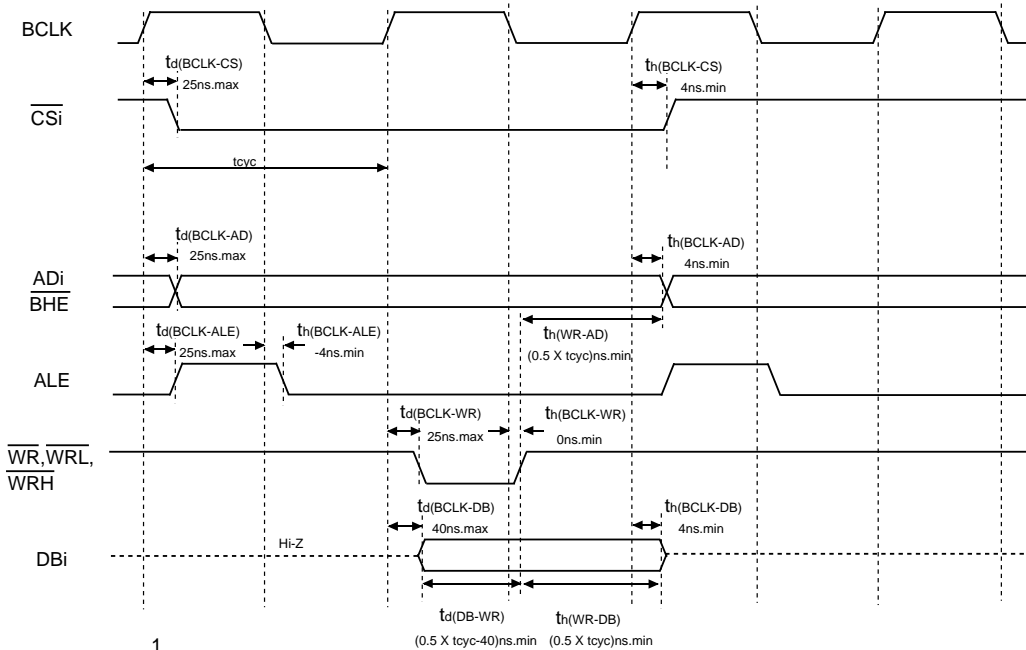
VCC1 = VCC2 = 5V

Memory Expansion Mode, Microprocessor Mode  
(For setting with no wait)

Read timing



Write timing



$$t_{\text{CYC}} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions

- VCC1=VCC2=5V
- Input timing voltage : VIL=0.8V, VIH=2.0V
- Output timing voltage : VOL=0.4V, VOH=2.4V

Figure 1.26.5. Timing Diagram (4)

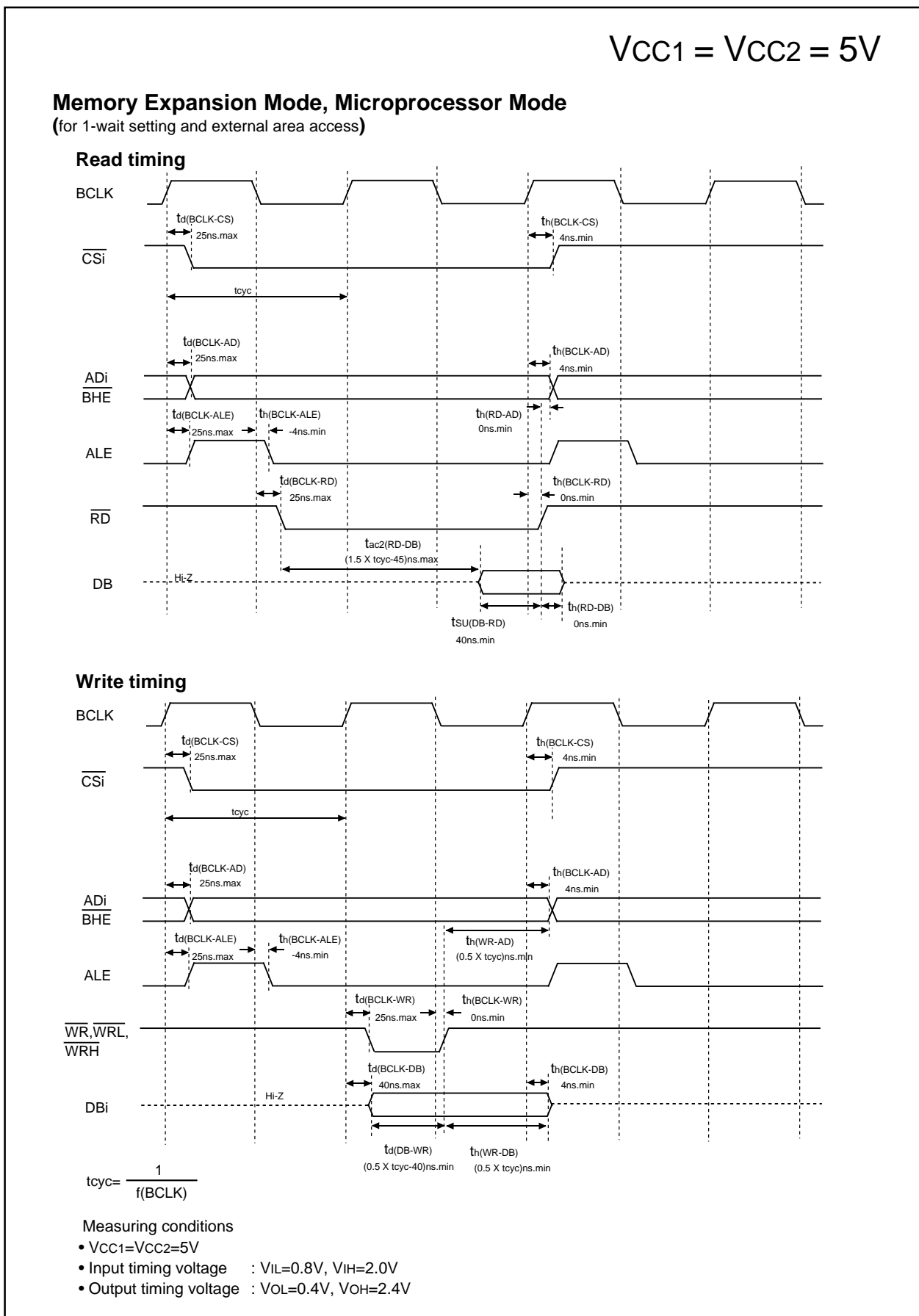


Figure 1.26.6. Timing Diagram (5)

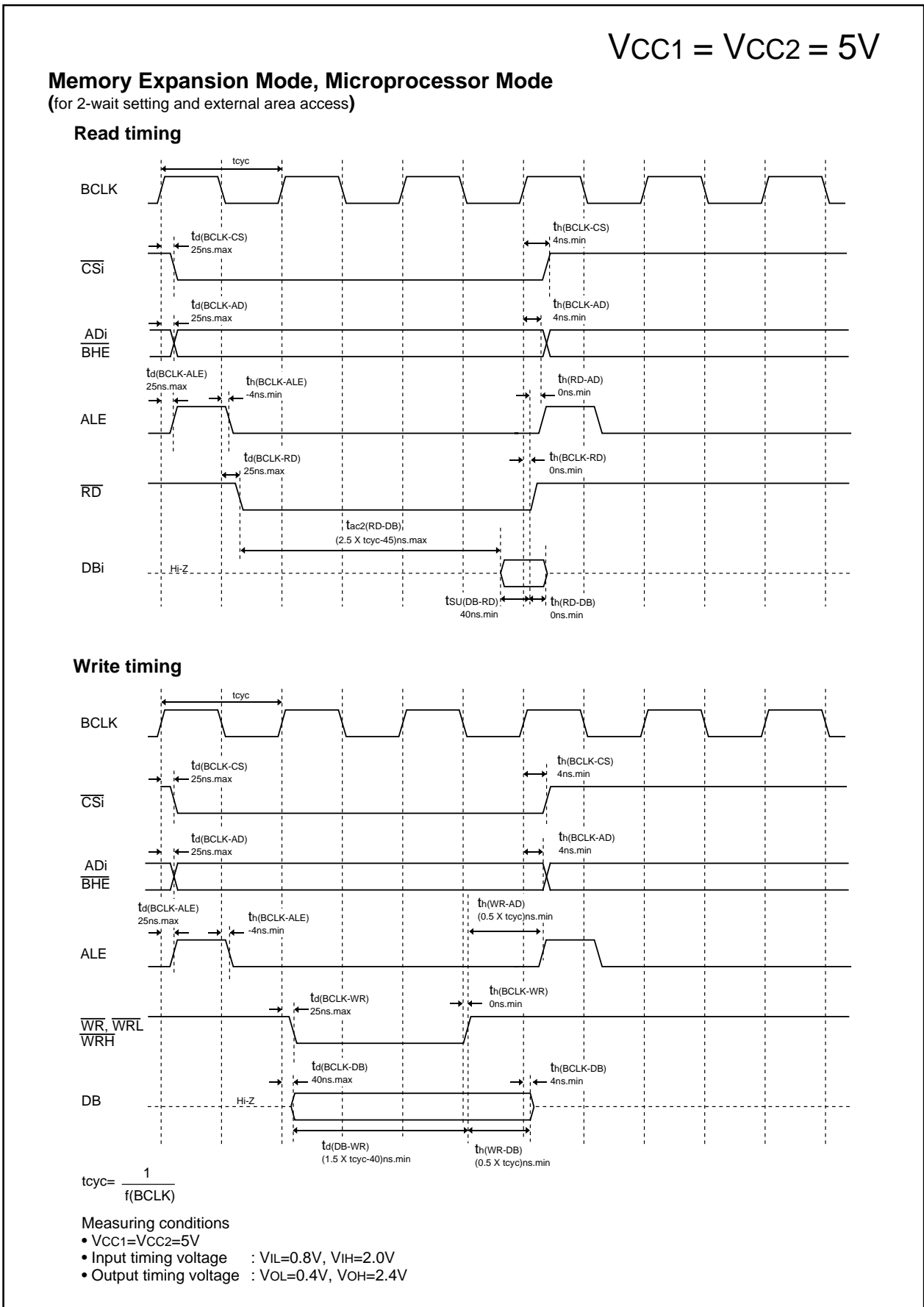


Figure 1.26.7. Timing Diagram (6)

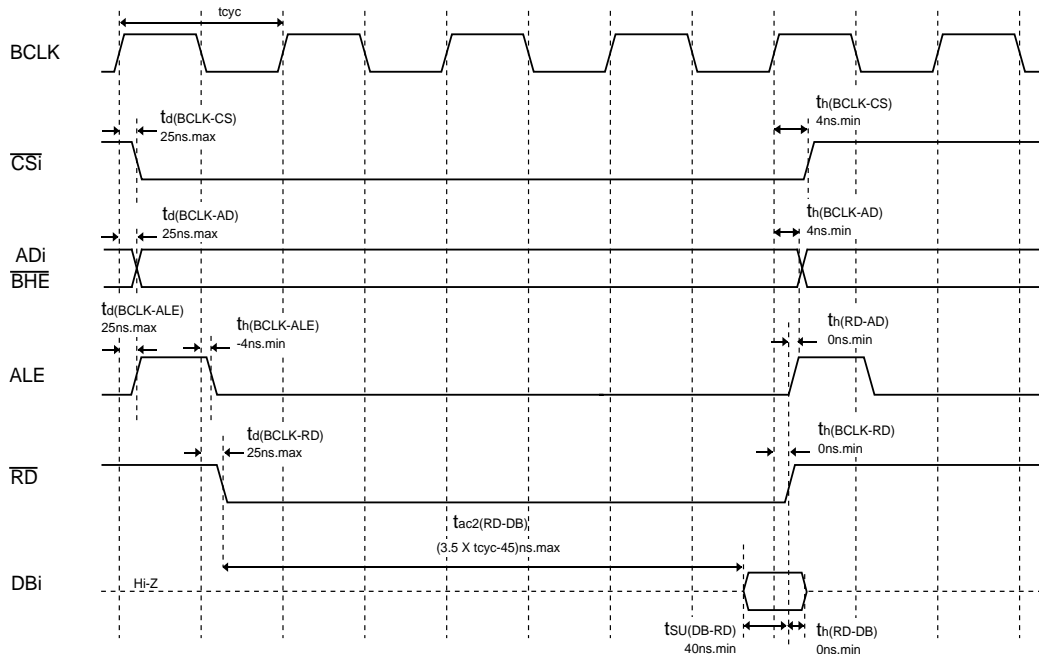


Electrical Characteristics (Vcc1 = Vcc2 = 5V)

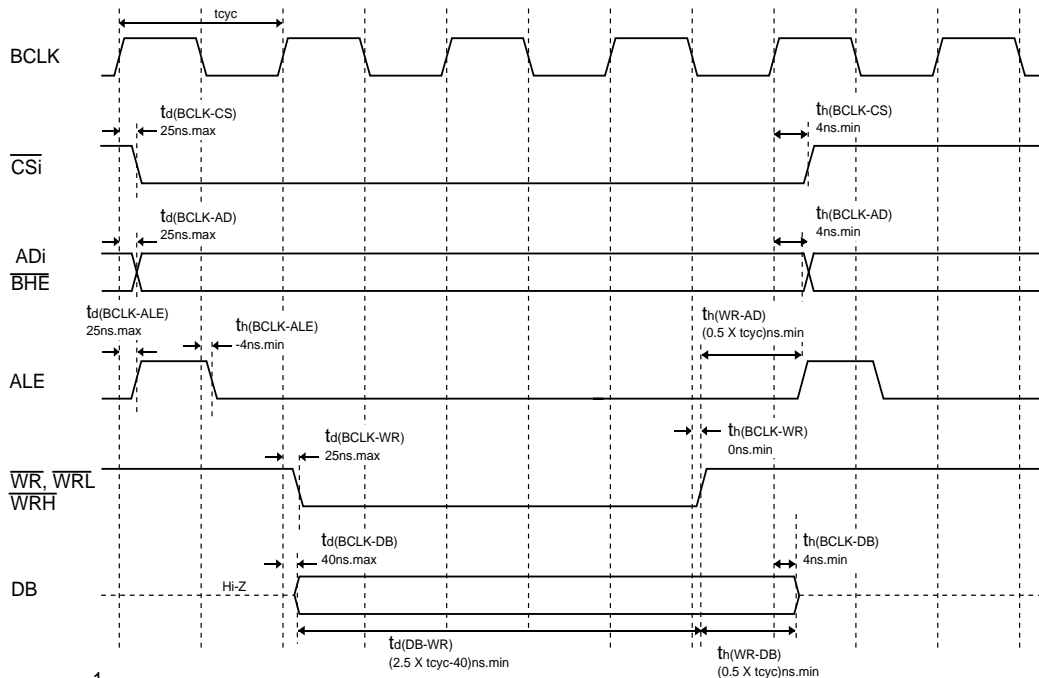
VCC1 = VCC2 = 5V

**Memory Expansion Mode, Microprocessor Mode**  
(for 3-wait setting and external area access)

**Read timing**



**Write timing**



$$t_{cy} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions

- VCC1=VCC2=5V
- Input timing voltage : VIL=0.8V, VIH=2.0V
- Output timing voltage : VOL=0.4V, VOH=2.4V

Figure 1.26.8. Timing Diagram (7)

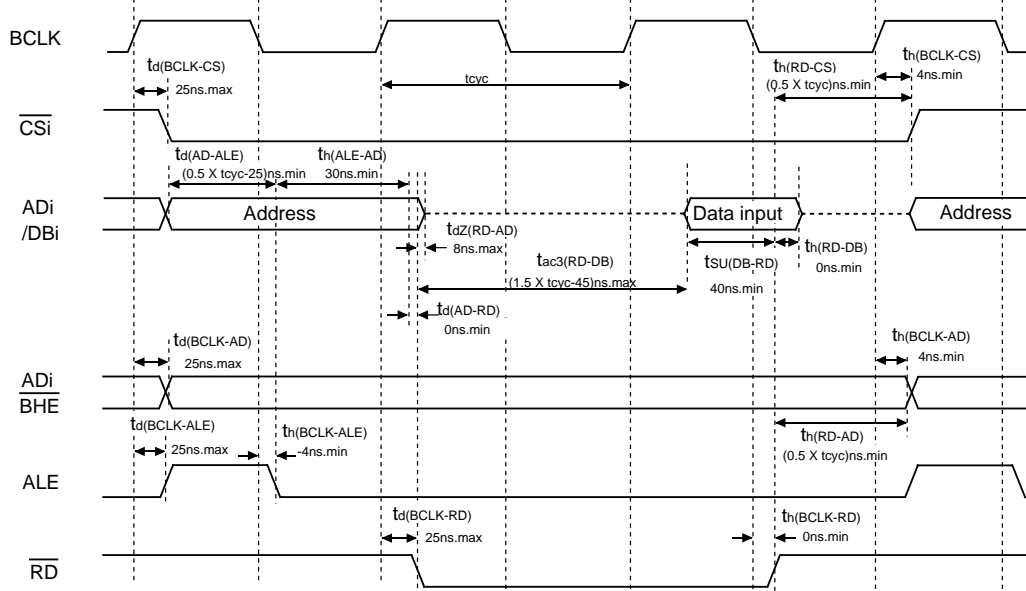
Electrical Characteristics (Vcc1 = Vcc2 = 5V)

VCC1 = VCC2 = 5V

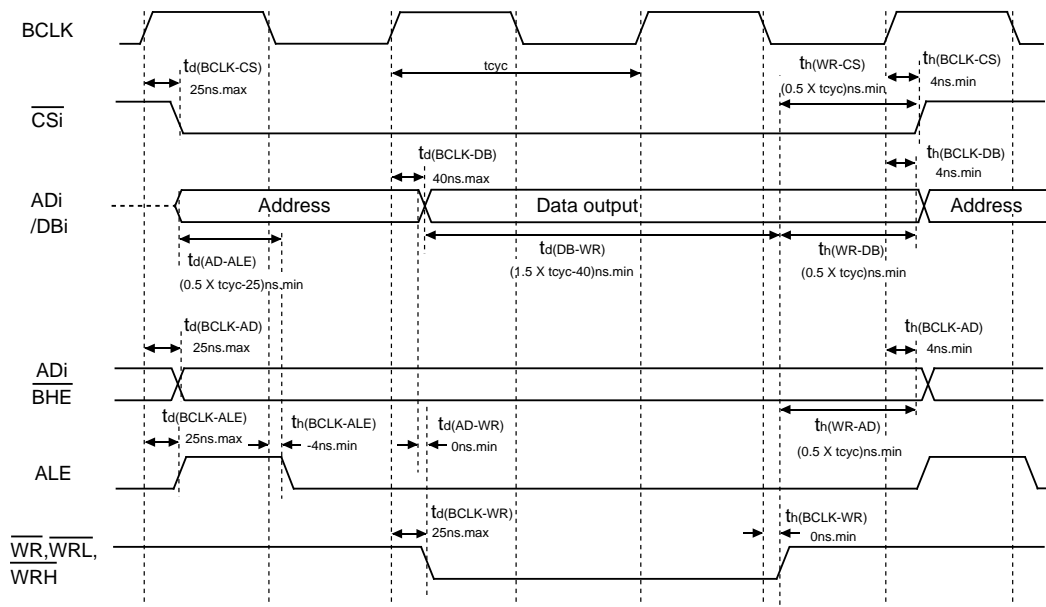
**Memory Expansion Mode, Microprocessor Mode**

(For 1- or 2-wait setting, external area access and multiplex bus selection)

**Read timing**



**Write timing**



$$t_{\text{cyc}} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions

- VCC1=VCC2=5V
- Input timing voltage : VIL=0.8V, VIH=2.0V
- Output timing voltage : VOL=0.4V, VOH=2.4V

Figure 1.26.9. Timing Diagram (8)

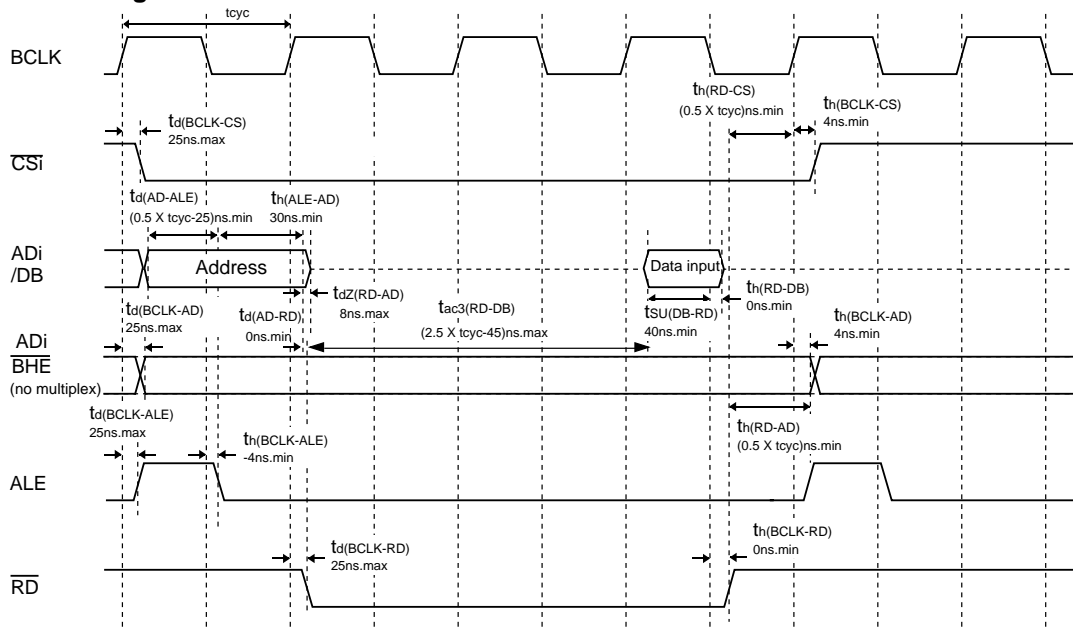
Electrical Characteristics (Vcc1 = Vcc2 = 5V)

VCC1 = VCC2 = 5V

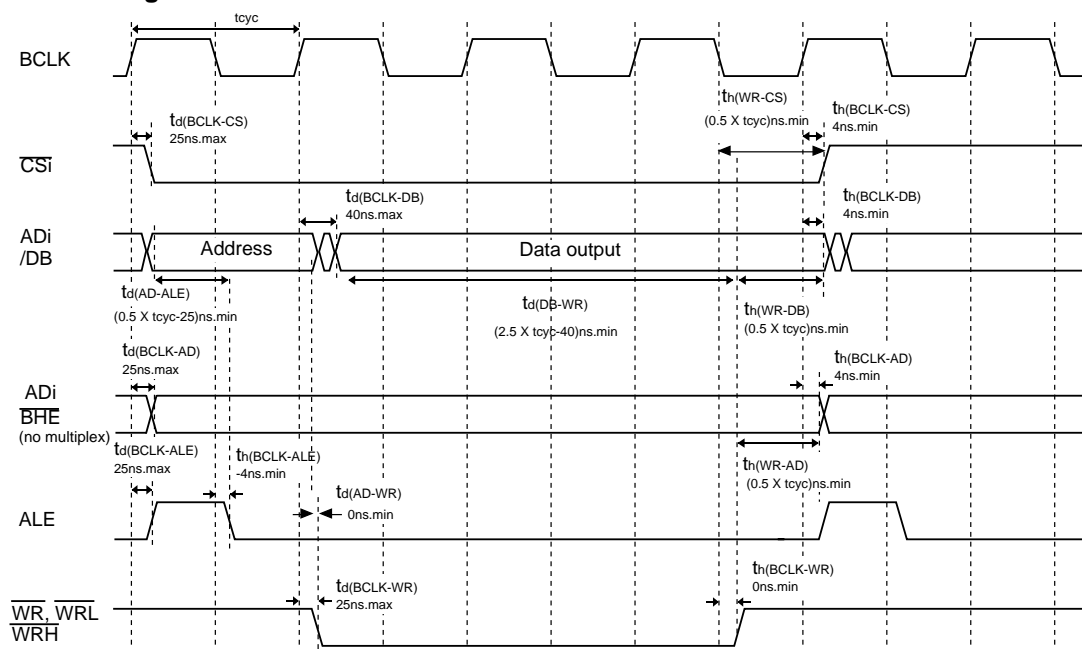
Memory Expansion Mode, Microprocessor Mode

(For 3-wait setting, external area access and multiplex bus selection)

Read timing



Write timing



$$t_{cy} = \frac{1}{f(\overline{BCLK})}$$

Measuring conditions

- Vcc1=Vcc2=5V
- Input timing voltage : VIL=0.8V, VIH=2.0V
- Output timing voltage : VOL=0.4V, VOH=2.4V

Figure 1.26.10. Timing Diagram (9)

VCC1 = VCC2 = 3V

Table 1.26.28. Electrical Characteristics (Note)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
V <sub>OH</sub>	HIGH output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , P11 <sub>0</sub> to P11 <sub>7</sub> , P12 <sub>0</sub> to P12 <sub>7</sub> , P13 <sub>0</sub> to P13 <sub>7</sub> , P14 <sub>0</sub> , P14 <sub>1</sub>	I <sub>OH</sub> =-1mA	V <sub>CC</sub> -0.5		V <sub>CC</sub>	V
V <sub>OH</sub>	HIGH output voltage	X <sub>OUT</sub>	HIGHPOWER			V <sub>CC</sub>	V
			LOWPOWER	I <sub>OH</sub> =-0.1mA	V <sub>CC</sub> -0.5	V <sub>CC</sub>	
V <sub>OH</sub>	HIGH output voltage	X <sub>COU</sub> T	HIGHPOWER	With no load applied	2.5		V
			LOWPOWER	With no load applied	1.6		
V <sub>OL</sub>	LOW output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , P11 <sub>0</sub> to P11 <sub>7</sub> , P12 <sub>0</sub> to P12 <sub>7</sub> , P13 <sub>0</sub> to P13 <sub>7</sub> , P14 <sub>0</sub> , P14 <sub>1</sub>	I <sub>OL</sub> =1mA			0.5	V
V <sub>OL</sub>	LOW output voltage	X <sub>OUT</sub>	HIGHPOWER			0.5	V
			LOWPOWER	I <sub>OL</sub> =0.1mA		0.5	
V <sub>OL</sub>	LOW output voltage	X <sub>COU</sub> T	HIGHPOWER	With no load applied	0		V
			LOWPOWER	With no load applied	0		
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	HOLD, RDY, TA0 <sub>IN</sub> to TA4 <sub>IN</sub> , TB0 <sub>IN</sub> to TB5 <sub>IN</sub> , INT0 to INT5, NMI, AD <sub>TRG</sub> , CT5 <sub>0</sub> to CT5 <sub>2</sub> , SCL, SDA, CLK0 to CLK4, TA2 <sub>OUT</sub> to TA4 <sub>OUT</sub> , K <sub>I0</sub> to K <sub>I3</sub> , RxD0 to RxD2, S <sub>IN3</sub> , S <sub>IN4</sub>		0.2		0.8	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	RESET		0.2	(0.7)	1.8	V
I <sub>IH</sub>	HIGH input current	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , P11 <sub>0</sub> to P11 <sub>7</sub> , P12 <sub>0</sub> to P12 <sub>7</sub> , P13 <sub>0</sub> to P13 <sub>7</sub> , P14 <sub>0</sub> , P14 <sub>1</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	V <sub>I</sub> =3V			4.0	μA
I <sub>IL</sub>	LOW input current	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , P11 <sub>0</sub> to P11 <sub>7</sub> , P12 <sub>0</sub> to P12 <sub>7</sub> , P13 <sub>0</sub> to P13 <sub>7</sub> , P14 <sub>0</sub> , P14 <sub>1</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	V <sub>I</sub> =0V			-4.0	μA
R <sub>PULLUP</sub>	Pull-up resistance	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , P11 <sub>0</sub> to P11 <sub>7</sub> , P12 <sub>0</sub> to P12 <sub>7</sub> , P13 <sub>0</sub> to P13 <sub>7</sub> , P14 <sub>0</sub> , P14 <sub>1</sub>	V <sub>I</sub> =0V	66	160	500	kΩ
R <sub>XIN</sub>	Feedback resistance	X <sub>IN</sub>			3.0		MΩ
R <sub>XICIN</sub>	Feedback resistance	X <sub>CIN</sub>			25		MΩ
V <sub>RAM</sub>	RAM retention voltage		At stop mode	2.0			V

Note : Referenced to V<sub>CC</sub>=V<sub>CC1</sub>=V<sub>CC2</sub>=2.7 to 3.3V, V<sub>SS</sub>=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=10MHz unless otherwise specified.

**Electrical Characteristics (Vcc1 = Vcc2 = 3V)**

**VCC1 = VCC2 = 3V**

**Table 1.26.29. Electrical Characteristics (2) (Note 1)**

Symbol	Parameter		Measuring condition		Standard			Unit		
					Min.	Typ.	Max.			
Icc	Power supply current (Vcc=2.7 to 3.6V)	In single-chip mode, the output pins are open and other pins are Vss	Mask ROM	f(BCLK)=10MHz, No division		8	11	mA		
				No division, Ring oscillation		1		mA		
			Flash memory	f(BCLK)=10MHz, No division		8	13	mA		
				No division, Ring oscillation		1.8		mA		
			Flash memory Program	f(BCLK)=10MHz, Vcc1=3.0V		12		mA		
			Flash memory Erase	f(BCLK)=10MHz, Vcc1=3.0V		22		mA		
			Mask ROM	f(XCIN)=32kHz, Low power dissipation mode, ROM(Note 3)		25		μA		
					Flash memory	f(BCLK)=32kHz, Low power dissipation mode, RAM(Note 3)		25		μA
						f(BCLK)=32kHz, Low power dissipation mode, Flash memory(Note 3)		420		μA
			Mask ROM Flash memory	f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity High	Ring oscillation, Wait mode		45		μA	
f(BCLK)=32kHz, Wait mode (Note 2), Oscillation capacity Low		6.0				μA				
Stop mode, Topr=25°C		0.7			3.0	μA				
Idet4	Power supply down detection dissipation current (Note 4)				0.6	4	μA			
Idet3	Reset level detection dissipation current (Note 4)				0.4	2	μA			
Idet2	RAM retention limit detection dissipation current (Note 4)				0.9	4	μA			

Note 1: Referenced to Vcc=Vcc1=Vcc2=2.7 to 3.3V, Vss=0V at Topr = -20 to 85 °C / -40 to 85 °C, f(BCLK)=10MHz unless otherwise specified.

Note 2: With one timer operated using fc32.

Note 3: This indicates the memory in which the program to be executed exists.

Note 4: Idet is dissipation current when the following bit is set to "1" (detection circuit enabled).

- Idet4: VC27 bit of VCR2 register
- Idet3: VC26 bit of VCR2 register
- Idet2: VC25 bit of VCR2 register

Electrical Characteristics ( $V_{CC1} = V_{CC2} = 3V$ )

$$V_{CC1} = V_{CC2} = 3V$$

**Timing Requirements****( $V_{CC1} = V_{CC2} = 3V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)****Table 1.26.30. External Clock Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_c$	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
$t_r$	External clock rise time		18	ns
$t_f$	External clock fall time		18	ns

**Table 1.26.31. Memory Expansion and Microprocessor Modes**

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
$t_{ac3(RD-DB)}$	Data input access time (when accessing multiplex bus area)		(Note 3)	ns
$t_{su(DB-RD)}$	Data input setup time	50		ns
$t_{su(RDY-BCLK)}$	RDY input setup time	40		ns
$t_{su(HOLD-BCLK)}$	HOLD input setup time	50		ns
$t_h(RD-DB)$	Data input hold time	0		ns
$t_h(BCLK-RDY)$	RDY input hold time	0		ns
$t_h(BCLK-HOLD)$	HOLD input hold time	0		ns
$t_d(BCLK-HLDA)$	HLDA output delay time		40	ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 60 \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 60 \quad [ns] \quad n \text{ is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 60 \quad [ns] \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

Electrical Characteristics ( $V_{CC1} = V_{CC2} = 3V$ )

$$V_{CC1} = V_{CC2} = 3V$$

**Timing Requirements****( $V_{CC1} = V_{CC2} = 3V$ ,  $V_{SS} = 0V$ , at  $T_{opr} = -20$  to  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)****Table 1.26.32. 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 1.26.33. 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 1.26.34. 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 1.26.35. 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 1.26.36. 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 1.26.37. 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		$\mu s$
$t_{su(TAiN-TAiOUT)}$	TAiOUT input setup time	500		ns
$t_{su(TAiOUT-TAiN)}$	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  $85^{\circ}C$  /  $-40$  to  $85^{\circ}C$  unless otherwise specified)****Table 1.26.38. 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)	160		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	160		ns

**Table 1.26.39. 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 1.26.40. 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 1.26.41. A-D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	ADTRG input cycle time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	ADTRG input LOW pulse width	200		ns

**Table 1.26.42. Serial I/O**

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

**Table 1.26.43. External Interrupt  $\overline{INT}_i$  Input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT}_i$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INT}_i$ input LOW pulse width	380		ns



Electrical Characteristics (Vcc1 ≥ Vcc2 = 3V)

VCC1 ≥ VCC2 = 3V

Switching Characteristics

(VCC1 = VCC2 = 3V, VSS = 0V, at Topr = - 20 to 85°C / - 40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.44. Memory Expansion, Microprocessor Modes (for setting with no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 1.26.11		30	ns
th(BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
th(RD-AD)	Address output hold time (refers to RD)		0		ns
th(WR-AD)	Address output hold time (refers to WR)		(Note 2)		ns
td(BCLK-CS)	Chip select output delay time			30	ns
th(BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
td(BCLK-ALE)	ALE signal output delay time			30	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 (refers to BCLK)			40	ns
th(BCLK-DB)	Data output hold time (refers to BCLK)		4		ns
td(DB-WR)	Data output delay time (refers to WR)		(Note 1)		ns
th(WR-DB)	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \quad [\text{ns}]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} \quad [\text{ns}]$$

Note 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  $t = -CR \times \ln(1 - V_{OL} / V_{CC2})$

by a circuit of the right figure.

For example, when  $V_{OL} = 0.2V_{CC2}$ ,  $C = 30\text{pF}$ ,  $R = 1\text{k}\Omega$ , hold time of output "L" level is

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2V_{CC2} / V_{CC2}) = 6.7\text{ns}.$$

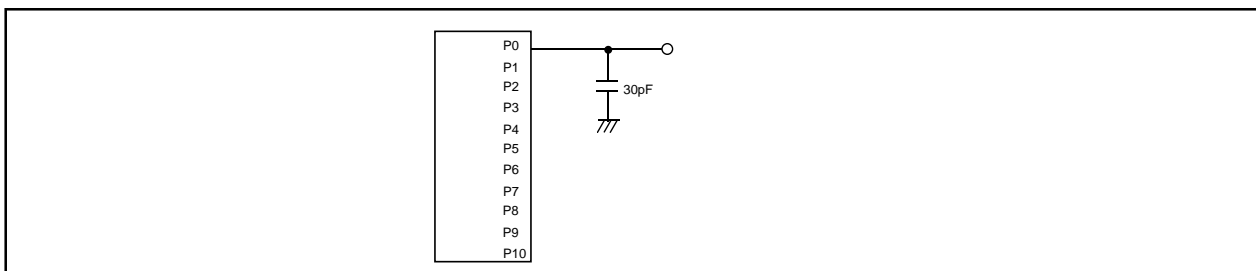
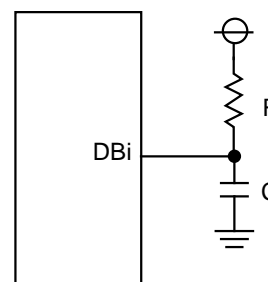


Figure 1.26.11. Ports P0 to P10 Measurement Circuit

Electrical Characteristics (Vcc1 ≥ Vcc2 = 3V)

$$V_{CC1} \geq V_{CC2} = 3V$$

Switching Characteristics

(VCC1 = VCC2 = 3V, VSS = 0V, at Topr = -20 to 85°C / -40 to 85°C, CM15="1" unless otherwise specified)

Table 1.26.45. Memory expansion and Microprocessor Modes  
(for 1- to 3-wait setting and external area access)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 1.26.11		30	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (refers to RD)		0		ns
t <sub>h</sub> (WR-AD)	Address output hold time (refers to WR)		(Note 2)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			30	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time			30	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time		-4		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			30	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			30	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (refers to BCLK)			40	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (refers to WR)		(Note 1)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 40 \quad [\text{ns}]$$

n is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting.

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} \quad [\text{ns}]$$

Note 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.

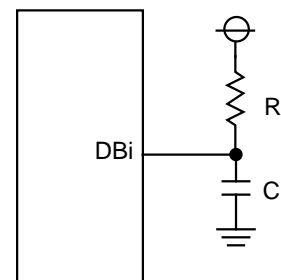
Hold time of data bus is expressed in

$$t = -CR \times \ln(1 - V_{OL} / V_{CC2})$$

by a circuit of the right figure.

For example, when V<sub>OL</sub> = 0.2V<sub>CC2</sub>, C = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2V_{CC2} / V_{CC2}) = 6.7\text{ns}.$$



Electrical Characteristics (Vcc1 ≥ Vcc2 = 3V)

$$V_{CC1} \geq V_{CC2} = 3V$$

**Switching Characteristics**

(VCC1 = VCC2 = 3V, VSS = 0V, at Topr = - 20 to 85°C / - 40 to 85°C, CM15="1" unless otherwise specified)

**Table 1.26.46. Memory expansion and Microprocessor Modes**  
(for 2- to 3-wait setting, external area access and multiplex bus selection)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 1.26.11		50	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (refers to RD)		(Note 1)		ns
t <sub>h</sub> (WR-AD)	Address output hold time (refers to WR)		(Note 1)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			50	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t <sub>h</sub> (RD-CS)	Chip select output hold time (refers to RD)		(Note 1)		ns
t <sub>h</sub> (WR-CS)	Chip select output hold time (refers to WR)		(Note 1)		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			40	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			40	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (refers to BCLK)			50	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (refers to BCLK)		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (refers to WR)		(Note 2)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (refers to WR)		(Note 1)		ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time (refers to BCLK)			40	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time (refers to BCLK)		- 4		ns
t <sub>d</sub> (AD-ALE)	ALE signal output delay time (refers to Address)		(Note 3)		ns
t <sub>h</sub> (ALE-AD)	ALE signal output hold time (refers to Address)		30		ns
t <sub>d</sub> (AD-RD)	RD signal output delay from the end of Address	0		ns	
t <sub>d</sub> (AD-WR)	WR signal output delay from the end of Address	0		ns	
t <sub>dZ</sub> (RD-AD)	Address output floating start time		8	ns	

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} \quad [\text{ns}]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 50 \quad [\text{ns}] \quad \text{n is "2" for 2-wait setting, "3" for 3-wait setting.}$$

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \quad [\text{ns}]$$

Electrical Characteristics ( $V_{CC1} = V_{CC2} = 3V$ )

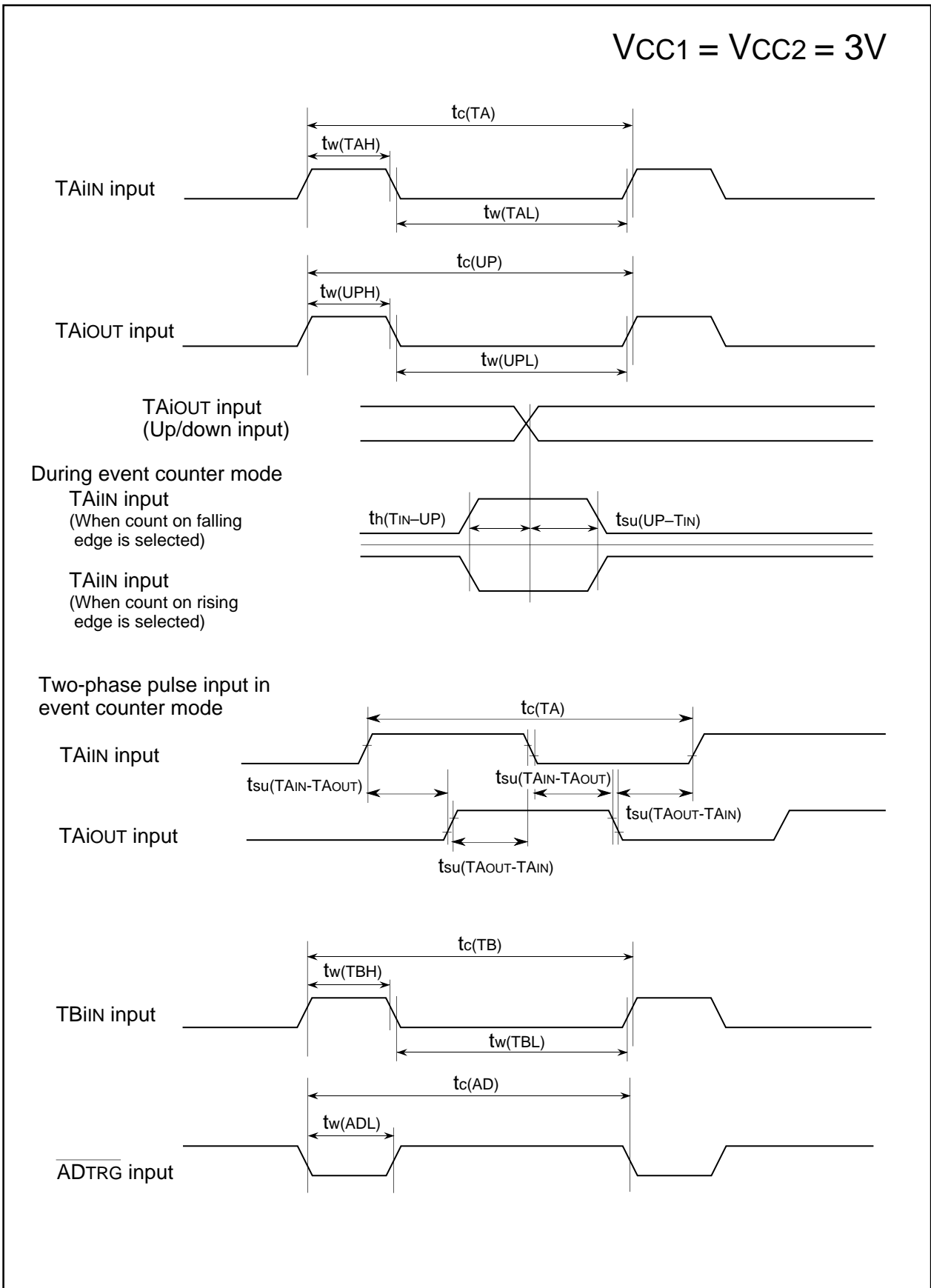


Figure 1.26.12. Timing Diagram (1)

Electrical Characteristics ( $V_{CC1} = V_{CC2} = 3V$ )

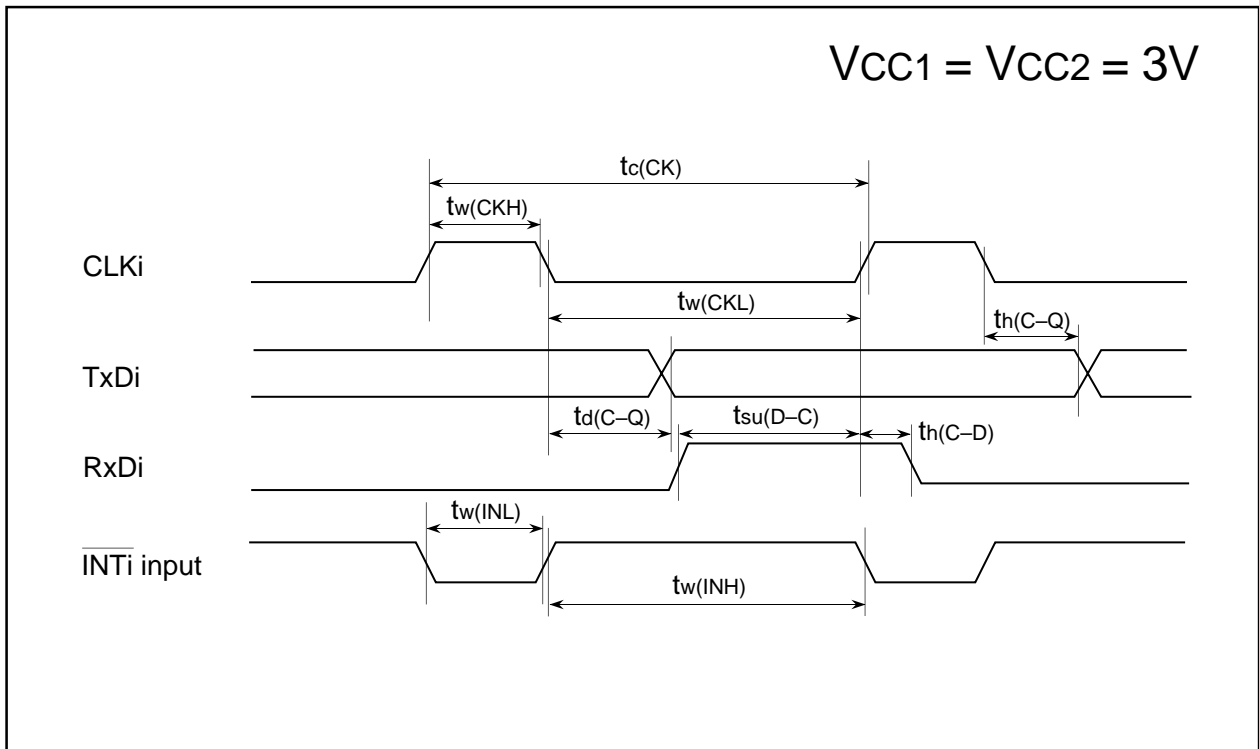


Figure 1.26.13. Timing Diagram (2)

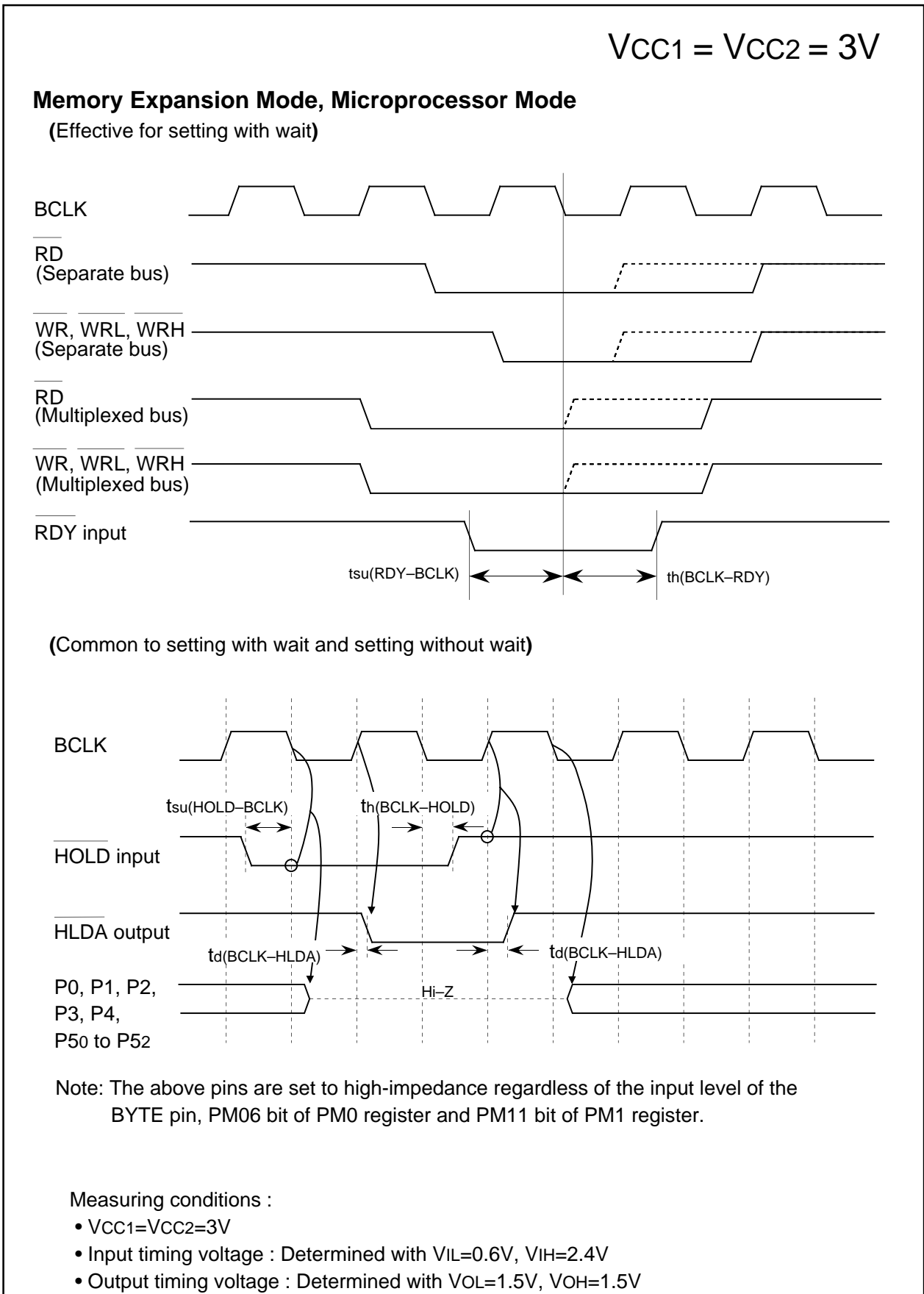


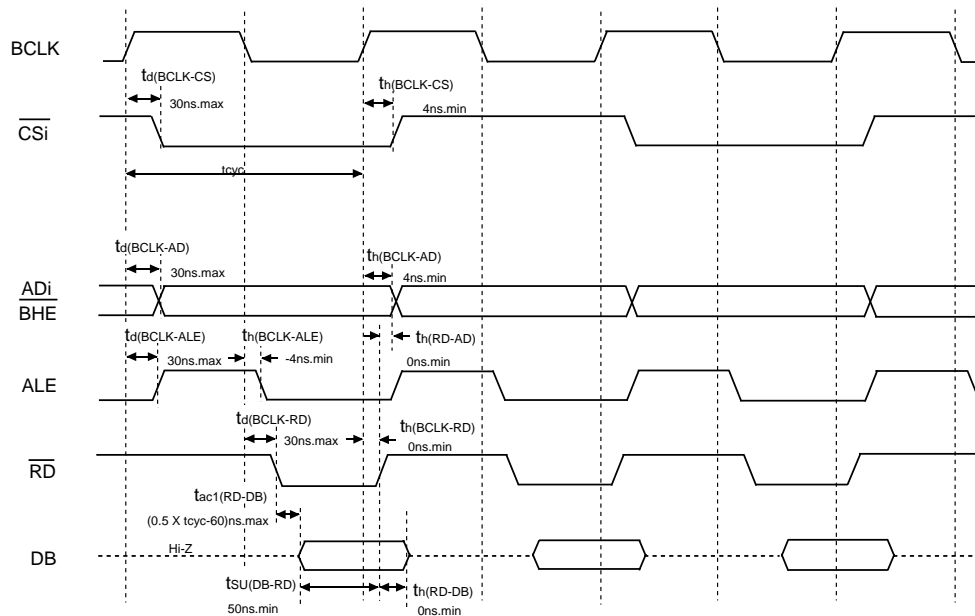
Figure 1.26.14. Timing Diagram (3)

$V_{CC1} \geq V_{CC2} = 3V$

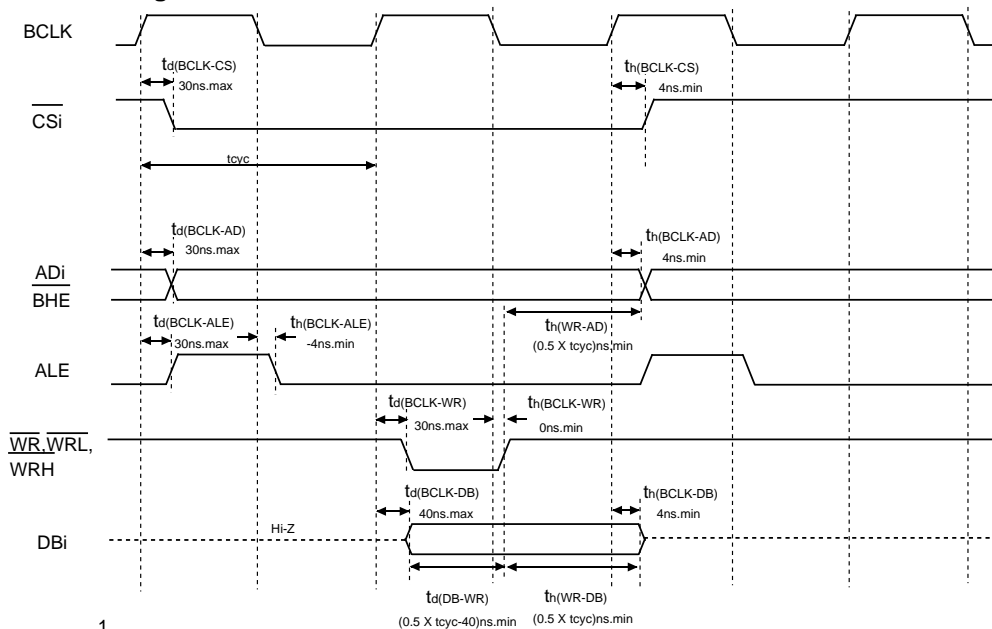
**Memory Expansion Mode, Microprocessor Mode**

(For setting with no wait)

**Read timing**



**Write timing**



$$t_{\text{cyc}} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions

- $V_{CC1} = V_{CC2} = 3V$
- Input timing voltage :  $V_{IL} = 0.6V$ ,  $V_{IH} = 2.4V$
- Output timing voltage :  $V_{OL} = 1.5V$ ,  $V_{OH} = 1.5V$

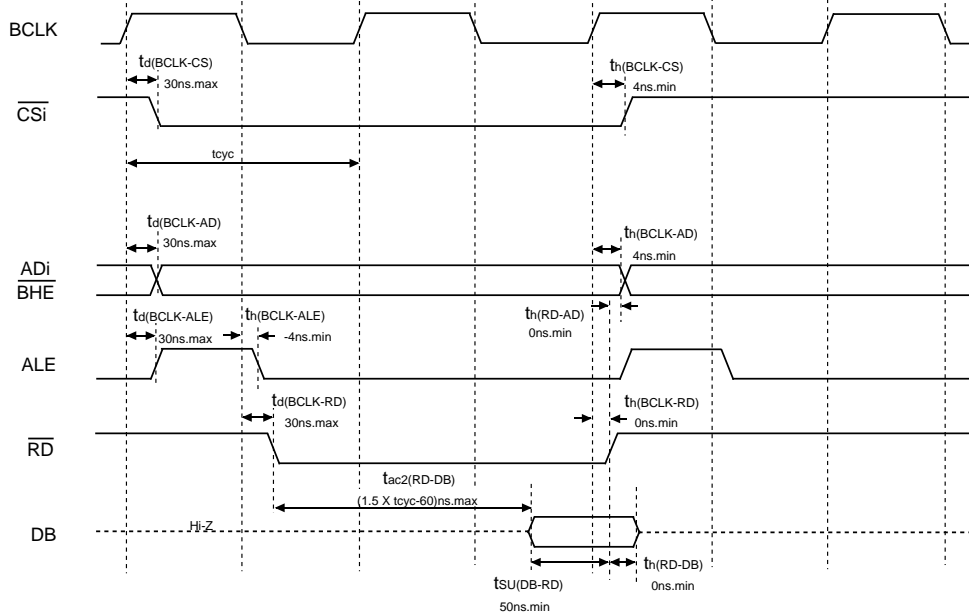
Figure 1.26.15. Timing Diagram (4)

VCC1 ≥ VCC2 = 3V

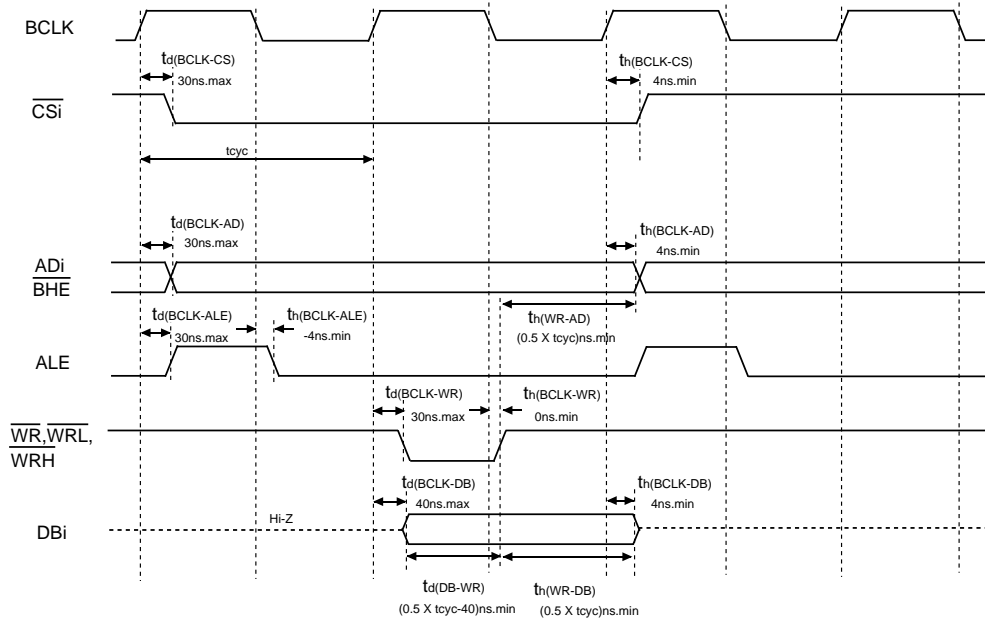
**Memory Expansion Mode, Microprocessor Mode**

(for 1-wait setting and external area access)

**Read timing**



**Write timing**



$$t_{cyc} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions

- VCC1=VCC2=3V
- Input timing voltage : VIL=0.6V, VIH=2.4V
- Output timing voltage : VOL=1.5V, VOH=1.5V

Figure 1.26.16. Timing Diagram (5)

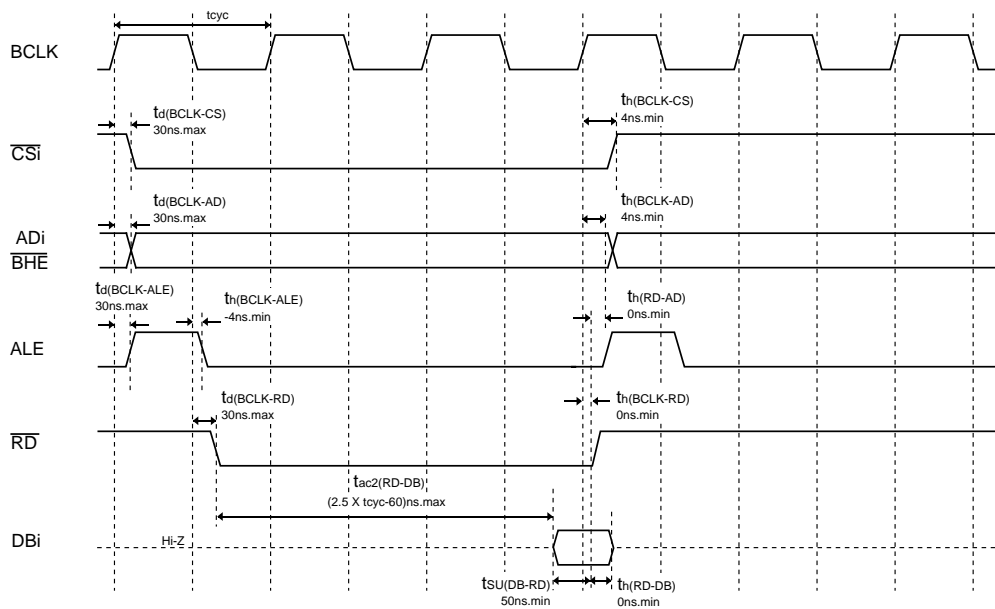


$V_{CC1} \geq V_{CC2} = 3V$

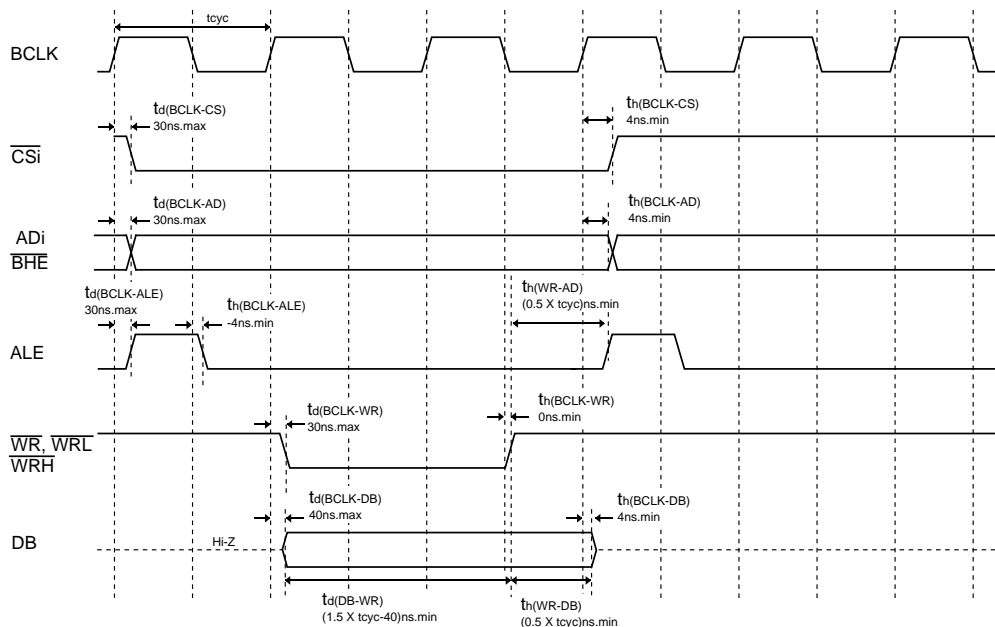
**Memory Expansion Mode, Microprocessor Mode**

(for 2-wait setting and external area access)

**Read timing**



**Write timing**



$$t_{cy} = \frac{1}{f(BCLK)}$$

- Measuring conditions
- $V_{CC1} = V_{CC2} = 3V$
  - Input timing voltage :  $V_{IL} = 0.6V, V_{IH} = 2.4V$
  - Output timing voltage :  $V_{OL} = 1.5V, V_{OH} = 1.5V$

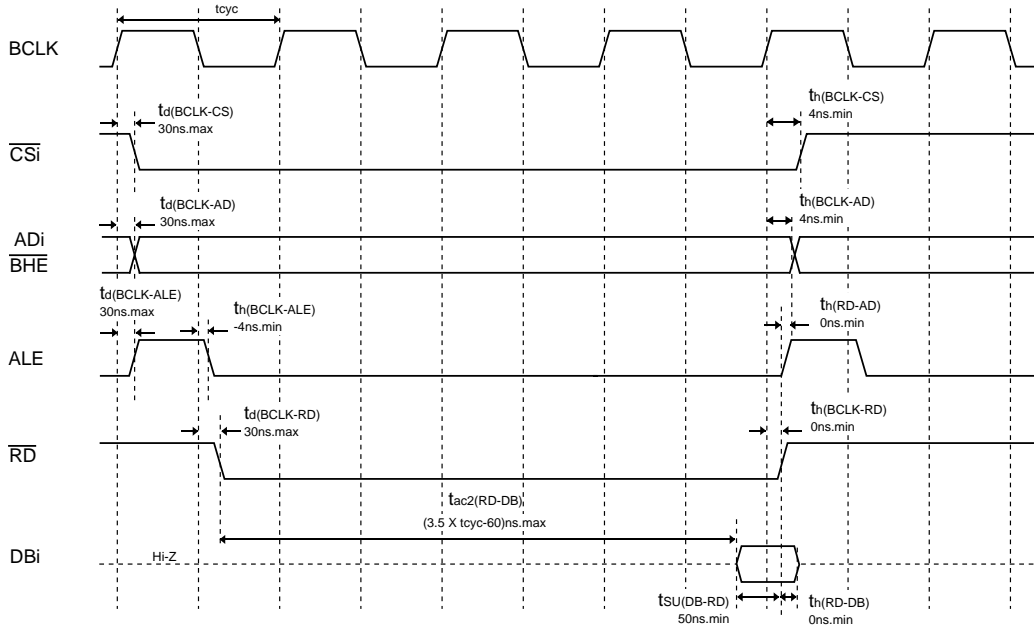
Figure 1.26.17. Timing Diagram (6)

$V_{CC1} \geq V_{CC2} = 3V$

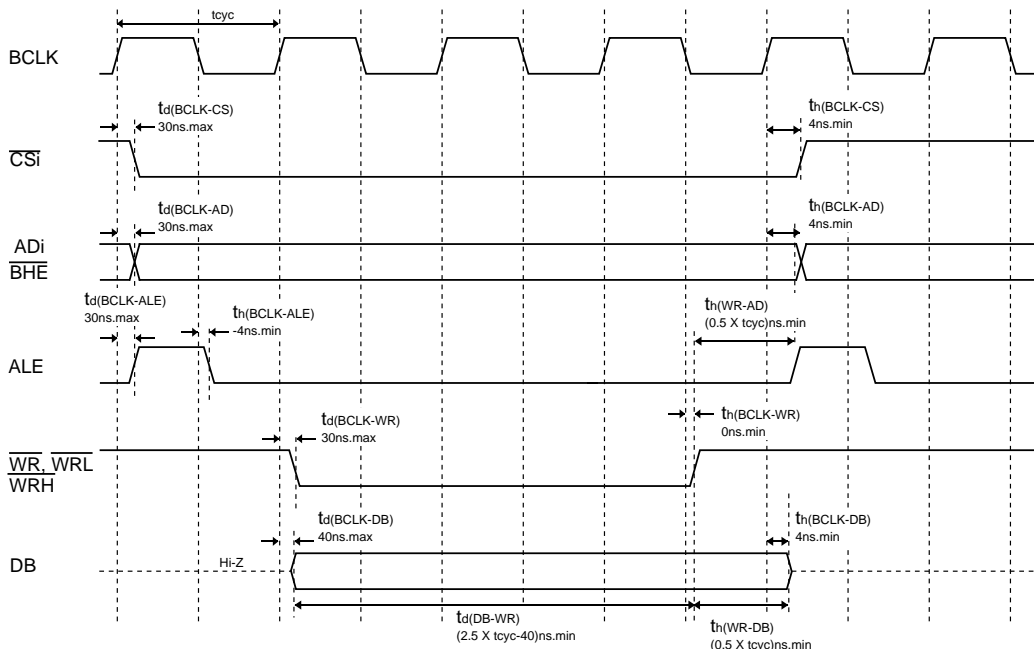
**Memory Expansion Mode, Microprocessor Mode**

(for 3-wait setting and external area access)

**Read timing**



**Write timing**



$$t_{cy} = \frac{1}{f(\text{BCLK})}$$

Measuring conditions

- $V_{CC1} = V_{CC2} = 3V$
- Input timing voltage :  $V_{IL} = 0.6V, V_{IH} = 2.4V$
- Output timing voltage :  $V_{OL} = 1.5V, V_{OH} = 1.5V$

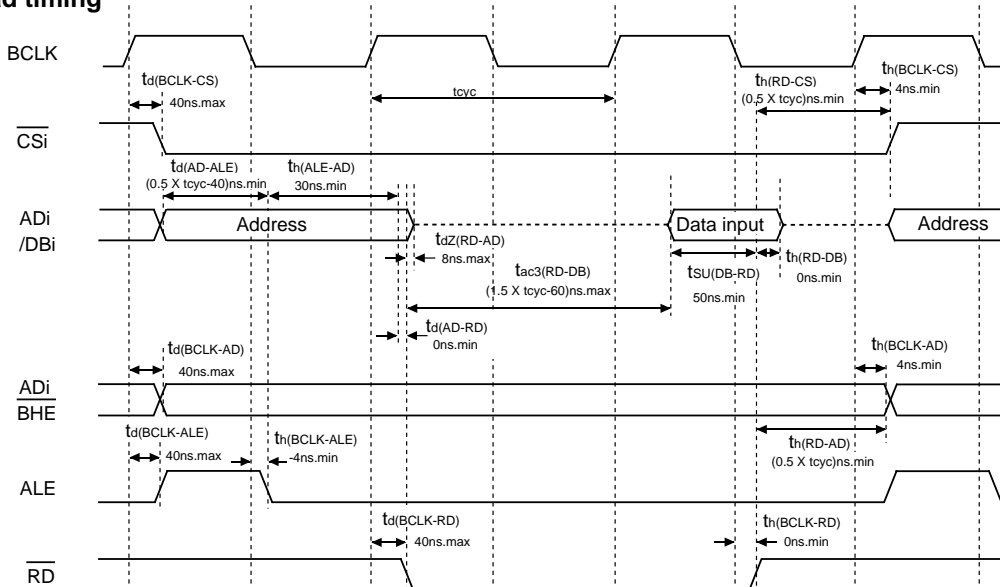
Figure 1.26.18. Timing Diagram (7)

VCC1 ≥ VCC2 = 3V

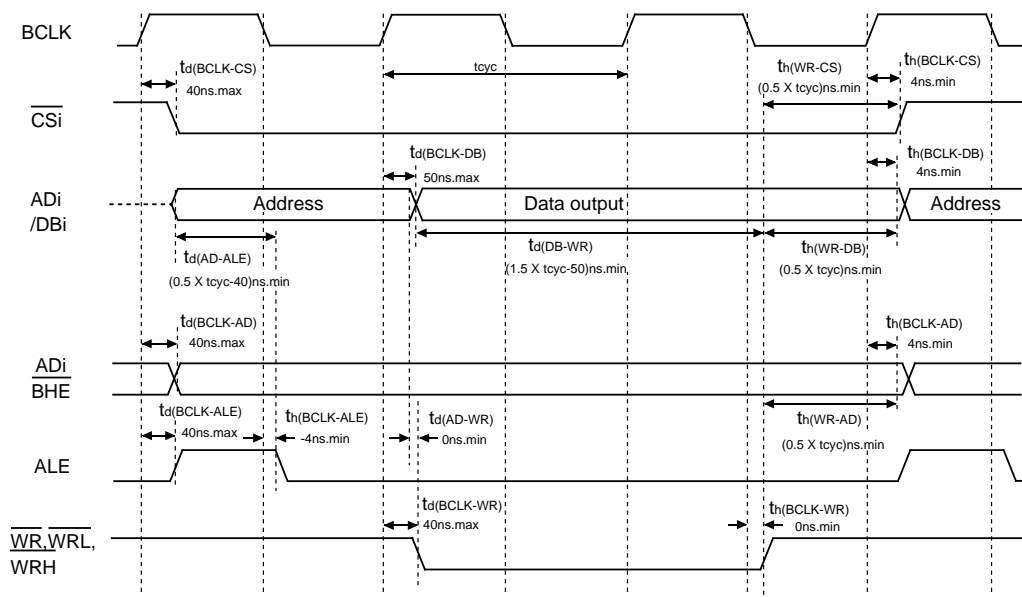
**Memory Expansion Mode, Microprocessor Mode**

(For 2-wait setting, external area access and multiplex bus selection)

**Read timing**



**Write timing**



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions

- VCC1=VCC2=3V
- Input timing voltage : VIL=0.6V, VIH=2.4V
- Output timing voltage : VOL=1.5V, VOH=1.5V

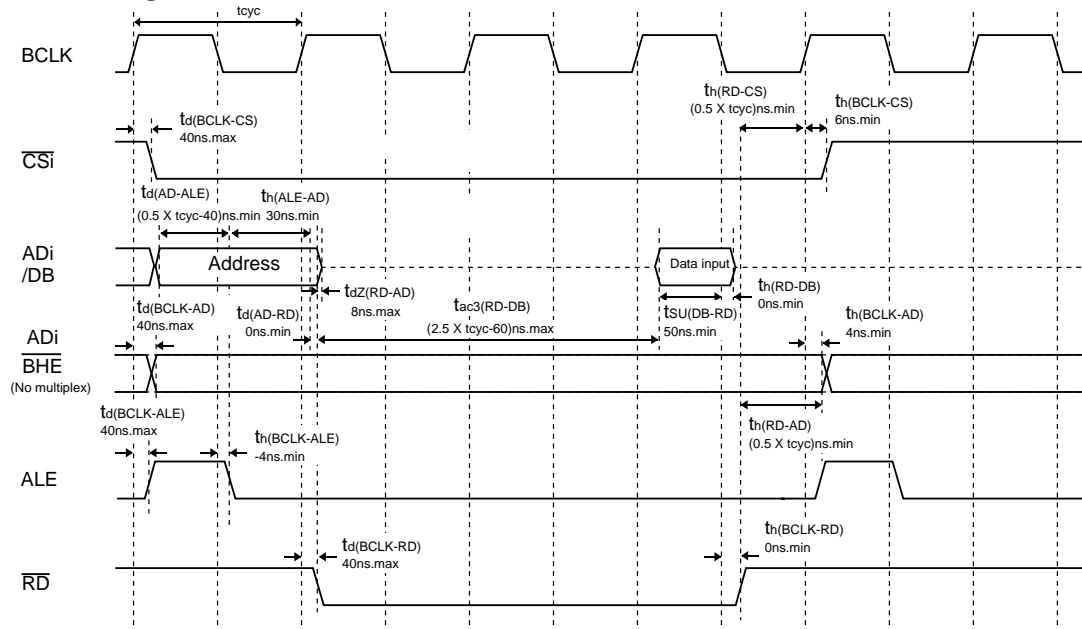
Figure 1.26.19. Timing Diagram (8)

VCC1 ≥ VCC2 = 3V

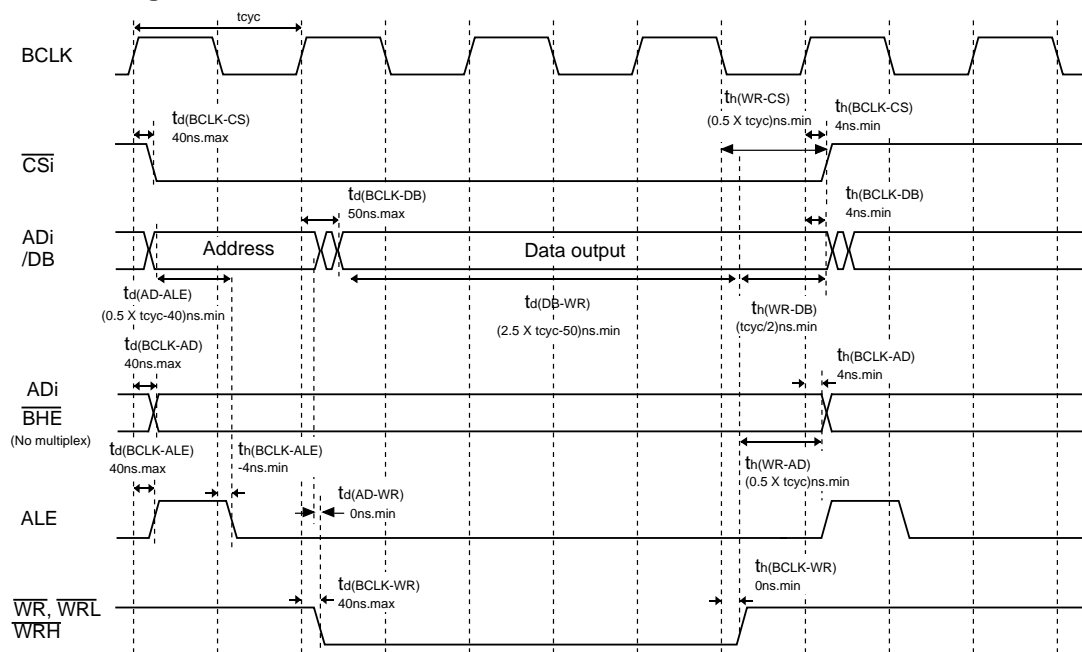
### Memory Expansion Mode, Microprocessor Mode

(For 3-wait setting, external area access and multiplex bus selection)

#### Read timing



#### Write timing



$$t_{cyc} = \frac{1}{f(BCLK)}$$

Measuring conditions

- VCC1=VCC2=3V
- Input timing voltage : VIL=0.6V, VIH=2.4V
- Output timing voltage : VOL=1.5V, VOH=1.5V

Figure 1.26.20. Timing Diagram (9)

# Flash Memory Version

## Flash Memory Performance

The flash memory version is functionally the same as the mask ROM version except that it internally contains flash memory.

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

Table 1.27.1 shows the outline performance of flash memory version (see Table 1.1.1 for the items not listed in Table 1.27.1.).

**Table 1.27.1. Flash Memory Version Specifications**

Item		Specification
Flash memory operating mode		3 modes (CPU rewrite, standard serial I/O, parallel I/O)
Erase block	User ROM area	See Figure 1.27.1
	Boot ROM area	1 block (4 Kbytes) (Note 1)
Method for program		In units of word, in units of byte (Note 2)
Method for erasure		Collective erase, block erase
Program, erase control method		Program and erase controlled by software command
Protect method		Protected for each block by lock bit
Number of commands		8 commands
Number of program and erasure		100 times
Data Retention		10 years
ROM code protection		Parallel I/O and standard serial I/O modes are supported.

Note 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.

Note 2: Can be programmed in byte units in only parallel input/output mode.

**Table 1.27.2. 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 by executing software commands from the CPU. EW0 mode: Can be rewritten in any area other than the flash memory EW1 mode: Can be rewritten in the flash memory	The user ROM area is rewritten by using a dedicated serial programmer. Standard serial I/O mode 1: Clock sync serial I/O Standard serial I/O mode 2: UART	The boot ROM and user ROM areas are rewritten by using a dedicated parallel programmer.
Areas which can be rewritten	User ROM area	User ROM area	User ROM area Boot ROM area
Operation 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

Flash Memory Version

### 1. Memory Map

The ROM in the flash memory version is separated between a user ROM area and a boot ROM area. Figure 1.27.1 shows the block diagram of flash memory. The user ROM area has a 4K-byte block A, in addition to the area that stores a program for microcomputer operation during single-chip or memory expansion mode.

The user ROM area is divided into several blocks, each of which can individually be protected (locked) against programming or erasure. The user ROM area can be rewritten in all of CPU rewrite, standard serial input/output, and parallel input/output modes. Block A is enabled for use by setting the PM1 register's PM10 bit to "1" (block A enabled, CS2 area at addresses 10000<sub>16</sub> to 26FFF<sub>16</sub>).

The boot ROM area is located at addresses that overlap the user ROM area, and can only be rewritten in parallel input/output mode. After a hardware reset that is performed by applying a high-level signal to the CNVss and P50 pins and a low-level signal to the P55 pin, the program in the boot ROM area is executed. After a hardware reset that is performed by applying a low-level signal to the CNVss pin, the program in the user ROM area is executed (but the boot ROM area cannot be read).

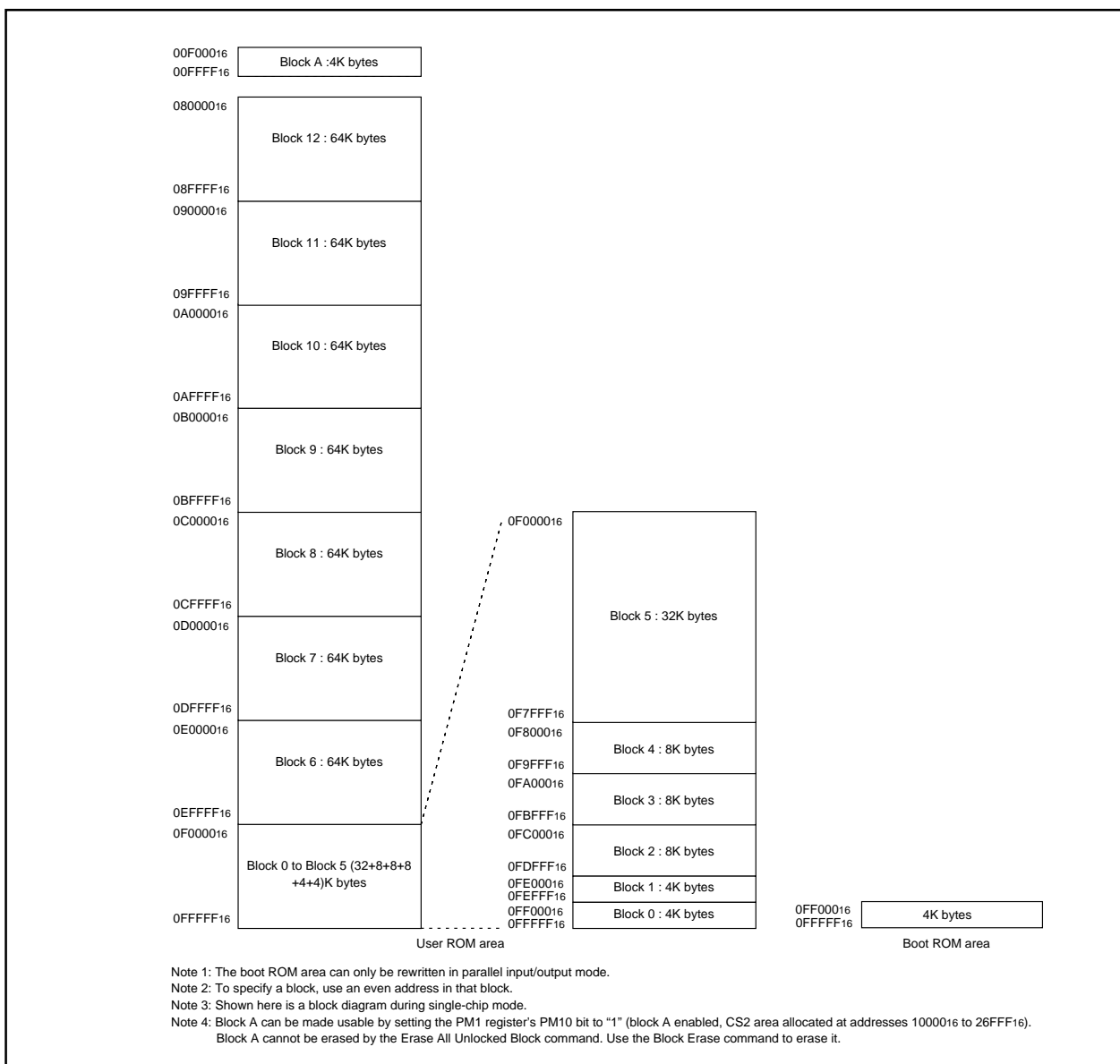


Figure 1.27.1. Flash Memory Block Diagram

## Flash Memory

---

### Boot Mode

After a hardware reset which is performed by applying a low-level signal to the P55 pin and a high-level signal to the CNVss and P50 pins, the microcomputer is placed in boot mode, thereby executing the program in the boot ROM area.

During boot mode, the boot ROM and user ROM areas are switched over by the FMR05 bit in the FMR0 register.

The boot ROM area contains a standard serial input/output mode based rewrite control program which was stored in it when shipped from the factory.

The boot ROM area can be rewritten in parallel input/output mode. Prepare an EW0 mode based rewrite control program and write it in the boot ROM area, and the flash memory can be rewritten as suitable for the system.

### Functions To Prevent Flash Memory from Rewriting

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

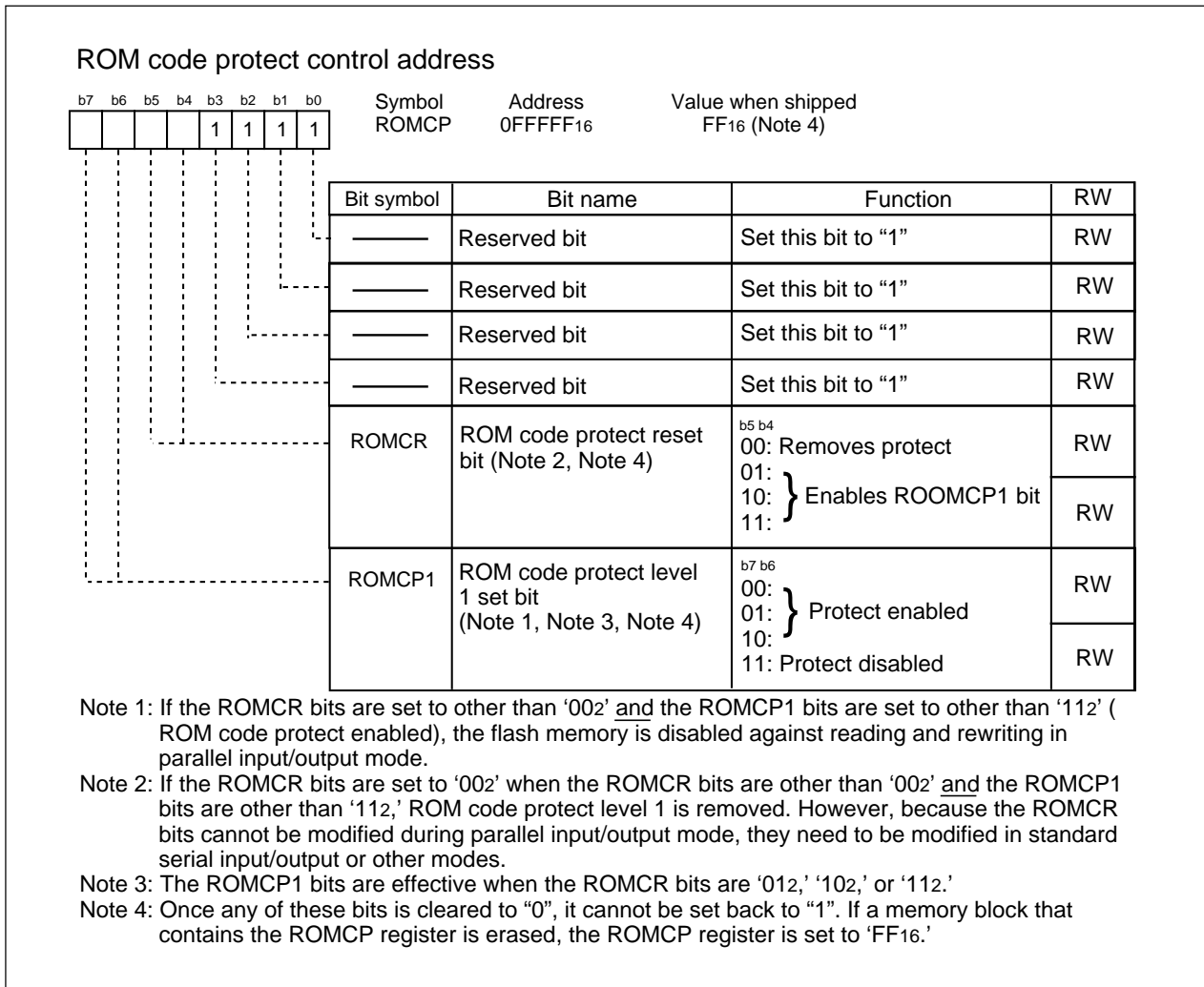
#### • ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten during parallel input/output mode. Figure 1.27.2 shows the ROMCP register.

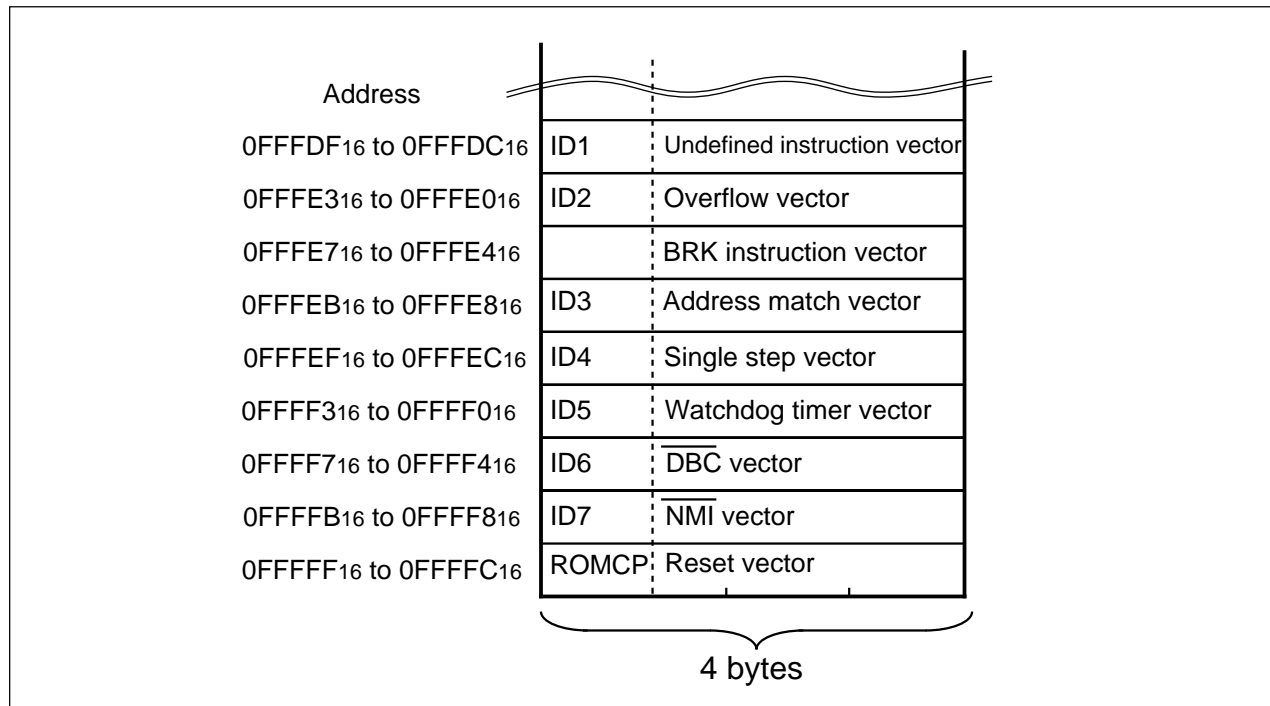
The ROMCP register is located in the user ROM area. The ROMCP1 bit consists of two bits. The ROM code protect function is enabled by clearing one or both of two ROMCP1 bits to "0" when the ROMCR bits are not '002,' with the flash memory thereby protected against reading or rewriting. Conversely, when the ROMCR bits are '002' (ROM code protect removed), the flash memory can be read or rewritten. Once the ROM code protect function is enabled, the ROMCR bits cannot be changed during parallel input/output mode. Therefore, use standard serial input/output or other modes to rewrite the flash memory.

#### • ID Code Check Function

Use this function in standard serial input/output mode. Unless the flash memory is blank, the ID codes sent from the programmer and the ID codes written in the flash memory are compared to see if they match. If the ID codes do not match, the commands sent from the programmer are not accepted. The ID code consists of 8-bit data, the areas of which, beginning with the first byte, are 0FFFDF<sub>16</sub>, 0FFFE3<sub>16</sub>, 0FFFE8<sub>16</sub>, 0FFFEF<sub>16</sub>, 0FFFF3<sub>16</sub>, 0FFFF7<sub>16</sub>, and 0FFFFB<sub>16</sub>. Prepare a program in which the ID codes are preset at these addresses and write it in the flash memory.



**Figure 1.27.2. ROMCP Register**



**Figure 1.27.3. Address for ID Code Stored**



## Flash Memory

## CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten by executing software commands from the CPU. Therefore, the user ROM area can be rewritten directly while the microcomputer is mounted on-board without having to use a ROM programmer, etc.

In CPU rewrite mode, only the user ROM area shown in Figure 1.27.1 can be rewritten and the boot ROM area cannot be rewritten. Make sure the Program and the Block Erase commands are executed only on each block in the user ROM area.

During CPU rewrite mode, the user ROM area be operated on in either Erase Write 0 (EW0) mode or Erase Write 1 (EW1) mode. Table 1.27.3 lists the differences between Erase Write 0 (EW0) and Erase Write 1 (EW1) modes.

Table 1.27.3. EW0 Mode and EW1 Mode

Item	EW0 mode	EW1 mode
Operation mode	<ul style="list-style-type: none"> <li>• Single chip mode</li> <li>• Memory expansion mode</li> <li>• Boot mode</li> </ul>	Single chip mode
Areas in which a rewrite control program can be located	<ul style="list-style-type: none"> <li>• User ROM area</li> <li>• Boot ROM area</li> </ul>	User ROM area
Areas in which a rewrite control program can be executed	Must be transferred to any area other than the flash memory (e.g., RAM) before being executed	Can be executed directly in the user ROM area
Areas which can be rewritten	User ROM area	User ROM area However, this does not include the area in which a rewrite control program exists
Software command limitations	None	<ul style="list-style-type: none"> <li>• Program, Block Erase command Cannot be executed on any block in which a rewrite control program exists</li> <li>• Erase All Unlocked Block command Cannot be executed when the lock bit for any block in which a rewrite control program exists is set to "1" (unlocked) or the FMR0 register's FMR02 bit is set to "1" (lock bit disabled)</li> <li>• Read Status Register command Cannot be executed</li> </ul>
Modes after Program or Erase	Read Status Register mode	Read Array mode
CPU status during Auto Write and Auto Erase	Operating	Hold state (I/O ports retain the state in which they were before the command was executed) <sup>(Note)</sup>
Flash memory status detection	<ul style="list-style-type: none"> <li>• Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program</li> <li>• Execute the Read Status Register command to read the status register's SR7, SR5, and SR4 flags.</li> </ul>	Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program

Note: Make sure no interrupts (except NMI and watchdog timer interrupts) and DMA transfers will occur.

## Flash Memory

---

- **EW0 Mode**

The microcomputer is placed in CPU rewrite mode by setting the FMR0 register's FMR01 bit to "1" (CPU rewrite mode enabled), ready to accept commands. In this case, because the FMR1 register's FMR11 bit = 0, EW0 mode is selected. The FMR01 bit can be set to "1" by writing "0" and then "1" in succession. Use software commands to control program and erase operations. Read the FMR0 register or status register to check the status of program or erase operation at completion.

- **EW1 Mode**

EW1 mode is selected by setting FMR11 bit to "1" (by writing "0" and then "1" in succession) after setting the FMR01 bit to "1" (by writing "0" and then "1" in succession).

Read the FMR0 register to check the status of program or erase operation at completion. The status register cannot be read during EW1 mode.

## Flash Memory

---

Figure 1.27.4 shows the FIDR, FMR0 and FMR1 registers.

### **FMR00 Bit**

This bit indicates the operating status of the flash memory. The bit is “0” when the Program, Erase, or Lock Bit program is running; otherwise, the bit is “1”.

### **FMR01 Bit**

The microcomputer is made ready to accept commands by setting the FMR01 bit to “1” (CPU rewrite mode). During boot mode, make sure the FMR05 bit also is “1” (user ROM area access).

### **FMR02 Bit**

The lock bit set for each block can be disabled by setting the FMR02 bit to “1” (lock bit disabled). (Refer to the description of the data protect function.) The lock bits set are enabled by setting the FMR02 bit to “0”. The FMR02 bit only disables the lock bit function and does not modify the lock bit data (lock bit status flag). However, if the Erase command is executed while the FMR02 bit is set to “1”, the lock bit data changes state from “0” (locked) to “1” (unlocked) after Erase is completed.

### **FMSTP Bit**

This bit is provided for initializing the flash memory control circuits, as well as for reducing the amount of current consumed in the flash memory. The internal flash memory is disabled against access by setting the FMSTP bit to “1”. Therefore, the FMSTP bit must be written to by a program in other than the flash memory.

In the following cases, set the FMSTP bit to “1”:

- When flash memory access resulted in an error while erasing or programming in EW0 mode (FMR00 bit not reset to “1” (ready))
- When entering low power mode or ring low power mode

Figure 1.27.7 shows a flow chart to be followed before and after entering low power mode.

Note that when going to stop or wait mode, the FMR0 register does not need to be set because the power for the internal flash memory is automatically turned off and is turned back on again after returning from stop or wait mode.

### **FMR05 Bit**

This bit switches between the boot ROM and user ROM areas during boot mode. Set this bit to “0” when accessing the boot ROM area (for read) or “1” (user ROM access) when accessing the user ROM area (for read, write, or erase).

### **FMR06 Bit**

This is a read-only bit indicating the status of auto program operation. The bit is set to “1” when a program error occurs; otherwise, it is cleared to “0”. For details, refer to the description of the full status check.

### **FMR07 Bit**

This is a read-only bit indicating the status of auto erase operation. The bit is set to “1” when an erase error occurs; otherwise, it is cleared to “0”. For details, refer to the description of the full status check.

Figure 1.27.5 and 1.27.6 show the setting and resetting of EW0 mode and EW1 mode, respectively.

### **FMR11 Bit**

Setting this bit to “1” places the microcomputer in EW1 mode.

### **FMR16 Bit**

This is a read-only bit indicating the execution result of the Read Lock Bit Status command.

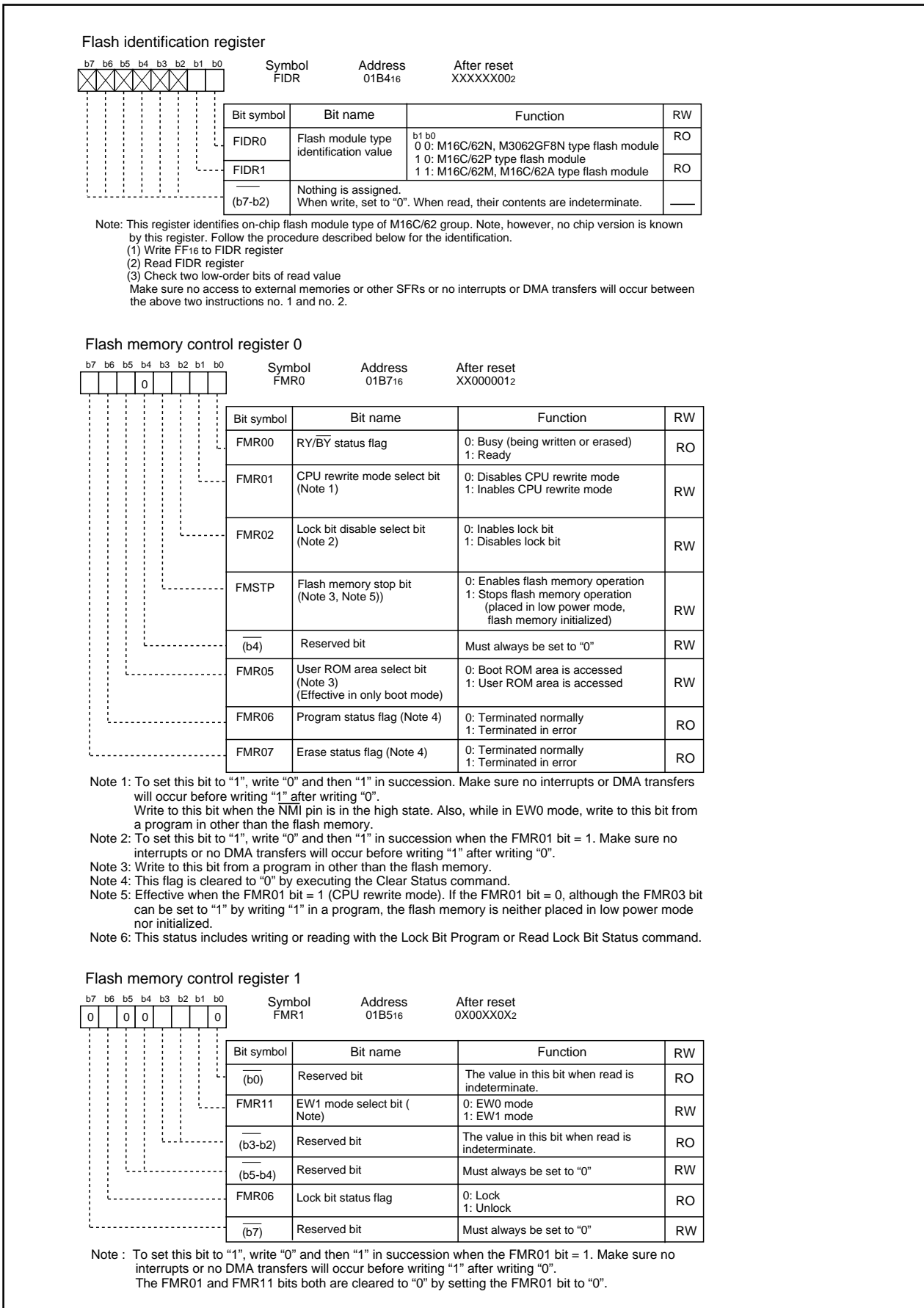


Figure 1.27.4. FIDR Register and FMR0 and FMR1 Registers

# Flash Memory

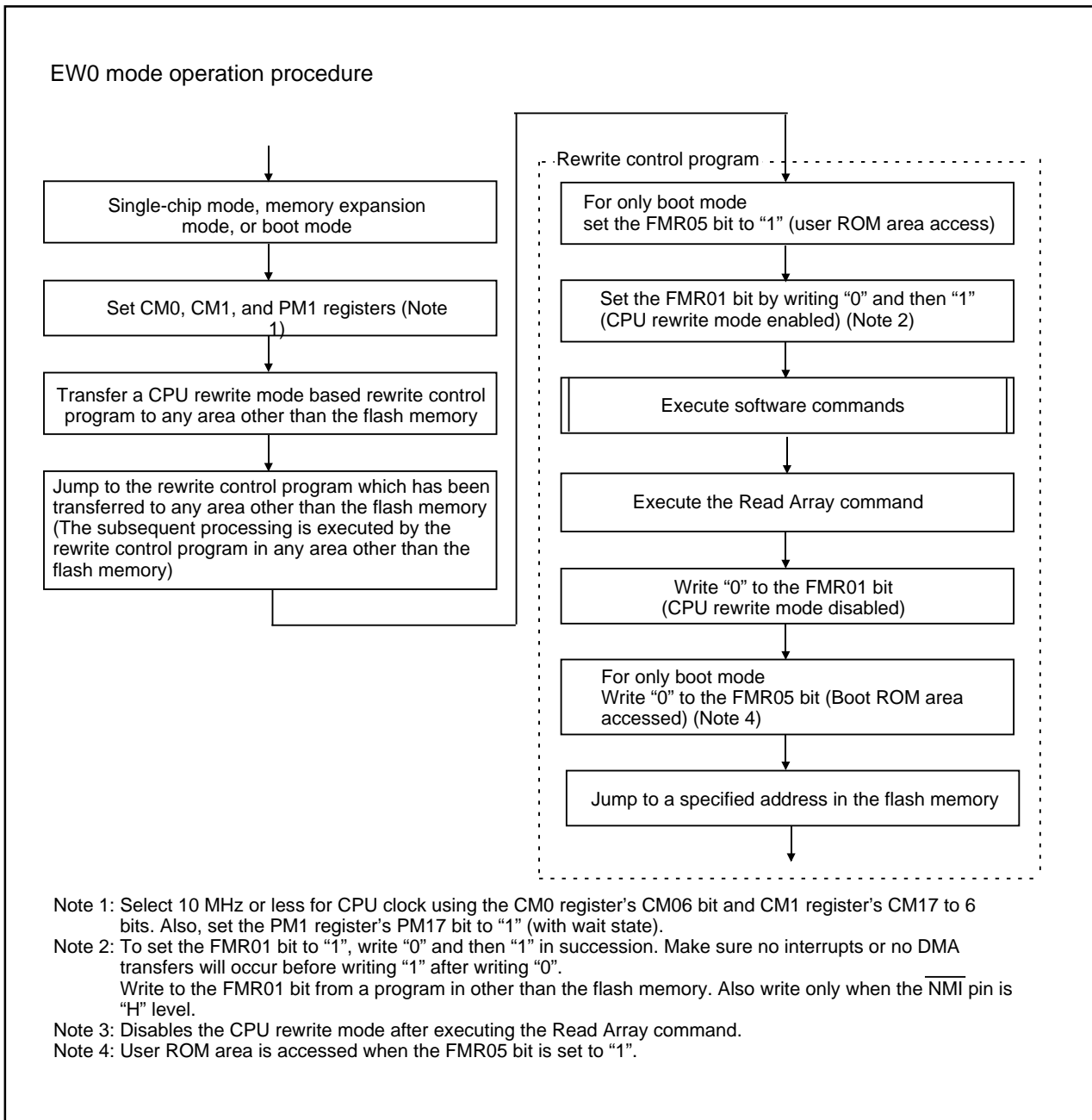


Figure 1.27.5. Setting and Tresetting of EW0 Mode

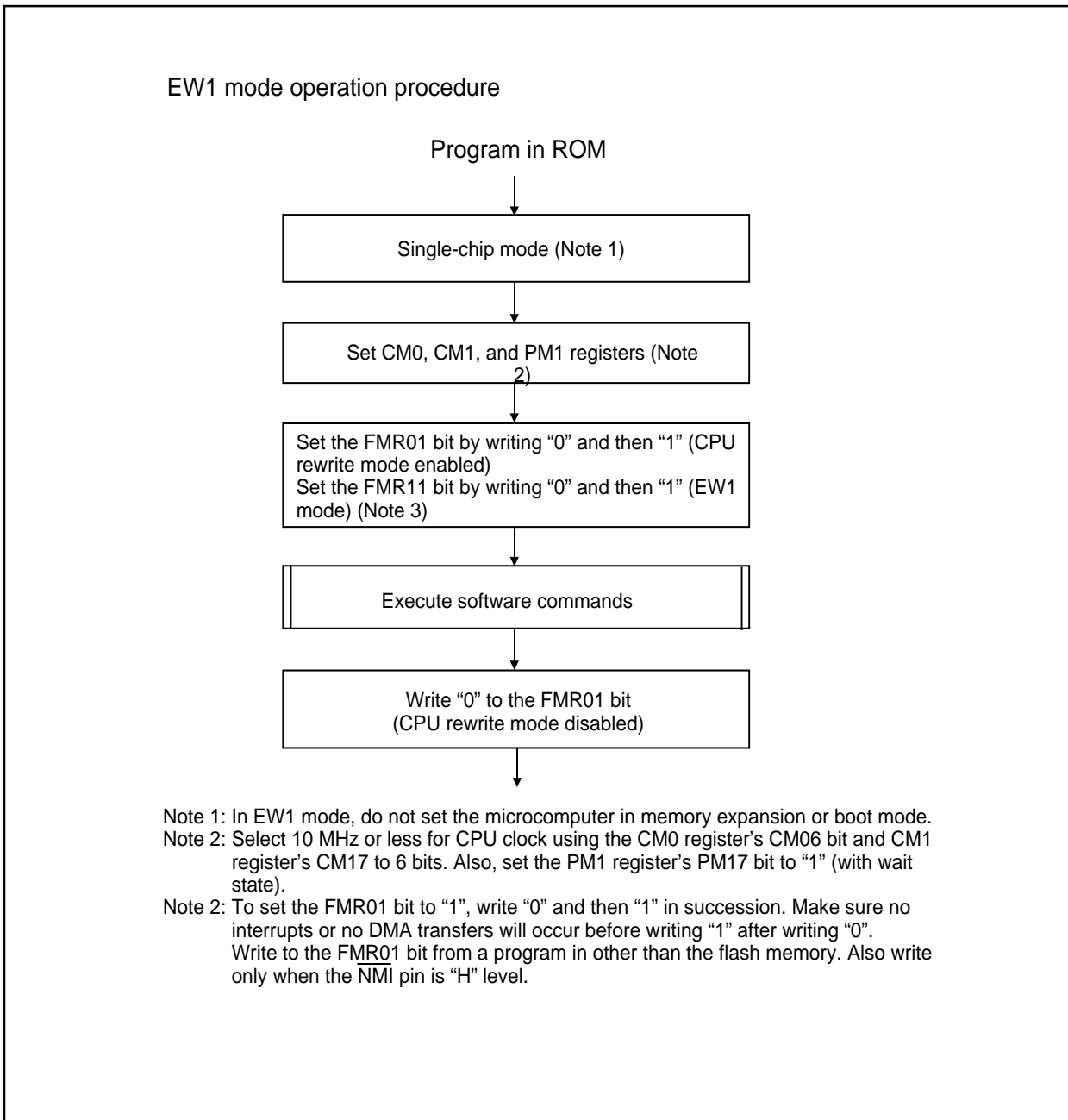
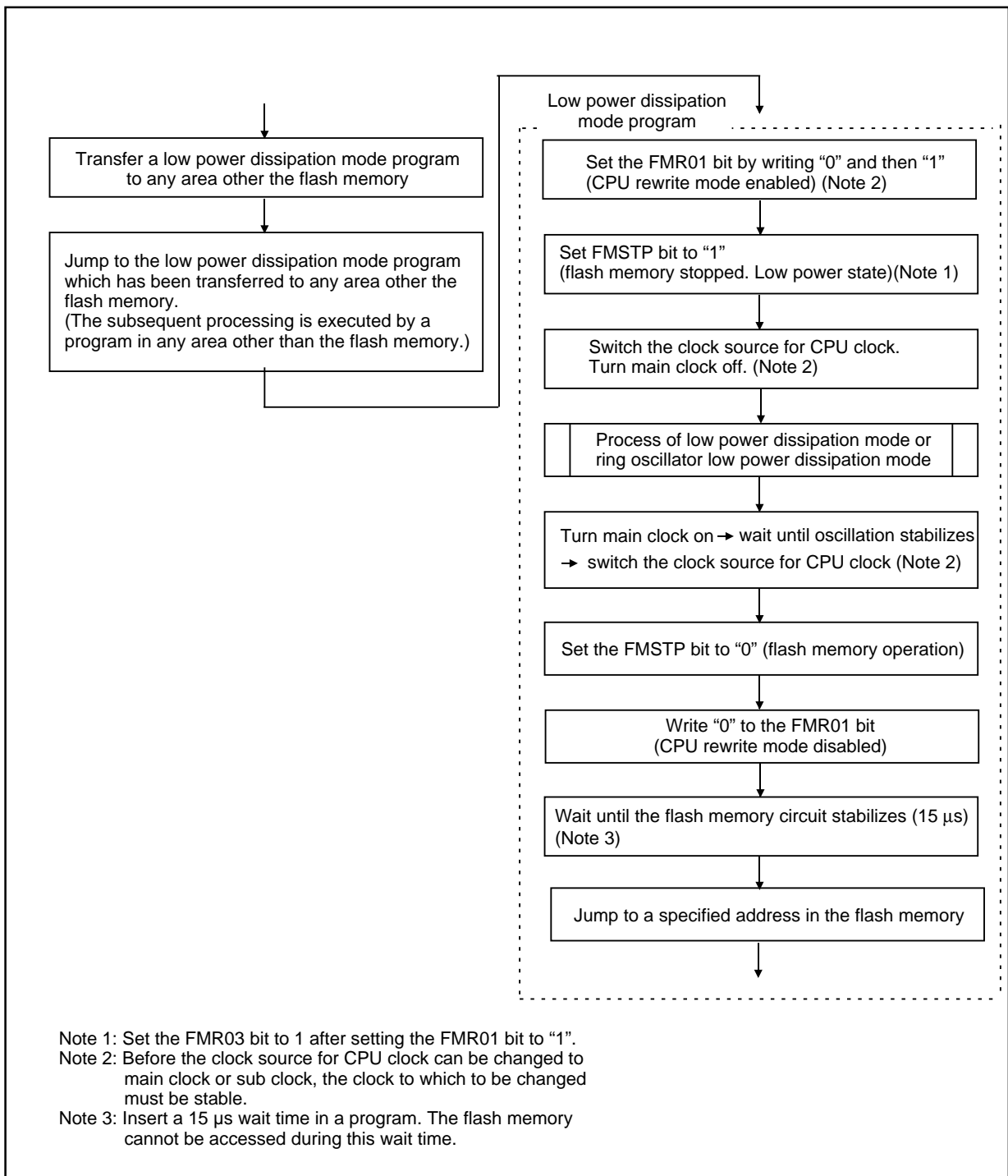


Figure 1.27.6. Setting and Resetting of EW1 Mode

**Flash Memory**



**Figure 1.27.7. Processing Before and After Low Power Sissipation Mode**

## Precautions on CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

### (1) Operation Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or less for BCLK using the CM06 bit in the CM0 register and the CM17 to CM16 bits in the CM1 register. Also, set the PM17 bit in the PM1 register to "1" (with wait state).

### (2) Instructions to Prevent from Using

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

### (3) Interrupts

EW0 Mode

- Any interrupt which has a vector in the variable vector table can be used providing that its vector is transferred into the RAM area.
- The  $\overline{\text{NMI}}$  and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.

Because the rewrite operation is halted when a  $\overline{\text{NMI}}$  or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

- The address match interrupt cannot be used because the flash memory's internal data is referenced.

EW1 Mode

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program or auto erase period.
- Avoid using watchdog timer interrupts.
- The  $\overline{\text{NMI}}$  interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.

Because the rewrite operation is halted when a  $\overline{\text{NMI}}$  interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

### (4) How to Access

To set the FMR01, FMR02, or FMR11 bit to "1", write "0" and then "1" in succession. This is necessary to ensure that no interrupts or DMA transfers will occur before writing "1" after writing "0". Also only when  $\overline{\text{NMI}}$  pin is "H" level.

### (5) Writing in the User ROM Space

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.



## Flash Memory

---

### (6) DMA Transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR0 register's FMR00 bit = 0 (during the auto program or auto erase period).

### (7) Writing Command and Data

Write the command code and data at even addresses.

### (8) Wait Mode

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

### (9) Stop Mode

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

- Set the FMR01 bit to "0" (CPU rewrite mode disabled) and disable DMA transfers before setting the CM10 bit to "1" (stop mode).
- Execute the JMP.B instruction subsequent to the instruction which sets the CM10 bit to "1" (stop mode)

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

L1:

Program after returning from stop mode

### (10) Low Power Dissipation Mode and Ring Oscillator Low Power Dissipation Mode

If the CM05 bit is set to "1" (main clock stop), the following commands must not be executed.

- Program
- Block erase
- Erase all unlocked blocks
- Lock bit program

## Software Commands

Software commands are described below. The command code and data must be read and written in 16-bit units, to and from even addresses in the user ROM area. When writing command code, the 8 high-order bits (D<sub>11</sub>–D<sub>8</sub>) are ignored.

**Table 1.27.4. Software Commands**

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D <sub>0</sub> to D <sub>7</sub> )	Mode	Address	Data (D <sub>0</sub> to D <sub>7</sub> )
Read array	Write	X	xxFF <sub>16</sub>			
Read status register	Write	X	xx70 <sub>16</sub>	Read	X	SRD
Clear status register	Write	X	xx50 <sub>16</sub>			
Program	Write	WA	xx40 <sub>16</sub>	Write	WA	WD
Block erase	Write	X	xx20 <sub>16</sub>	Write	BA	xxD0 <sub>16</sub>
Erase all unlocked block <sup>(Note)</sup>	Write	X	xxA7 <sub>16</sub>	Write	X	xxD0 <sub>16</sub>
Lock bit program	Write	BA	xx77 <sub>16</sub>	Write	BA	xxD0 <sub>16</sub>
Read lock bit status	Write	X	xx71 <sub>16</sub>	Write	BA	xxD0 <sub>16</sub>

Note: It is only blocks 0 to 12 that can be erased by the Erase All Unlocked Block command.

Block A cannot be erased. Use the Block Erase command to erase block A.

SRD: Status register data (D<sub>7</sub> to D<sub>0</sub>)

WA: Write address (Make sure the address value specified in the the first bus cycle is the same even address as the write address specified in the second bus cycle.)

WD: Write data (16 bits)

BA: Uppermost block address (even address, however)

X: Any even address in the user ROM area

x: High-order 8 bits of command code (ignored)

### Read Array Command (FF<sub>16</sub>)

This command reads the flash memory.

Writing 'xxFF<sub>16</sub>' in the first bus cycle places the microcomputer in read array mode. Enter the read address in the next or subsequent bus cycles, and the content of the specified address can be read in 16-bit units.

Because the microcomputer remains in read array mode until another command is written, the contents of multiple addresses can be read in succession.

### Read Status Register Command (70<sub>16</sub>)

This command reads the status register.

Write 'xx70<sub>16</sub>' in the first bus cycle, and the status register can be read in the second bus cycle. (Refer to "Status Register.") When reading the status register too, specify an even address in the user ROM area.

Do not execute this command in EW1 mode.

## Flash Memory

**Clear Status Register Command (50<sub>16</sub>)**

This command clears the status register to "0".

Write 'xx50<sub>16</sub>' in the first bus cycle, and the FMR06 to FMR07 bits in the FMR0 register and SR4 to SR5 in the status register will be cleared to "0".

**Program Command (40<sub>16</sub>)**

This command writes data to the flash memory in 1 word (2 byte) units.

Write 'xx40<sub>16</sub>' in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

Check the FMR00 bit in the FMR0 register to see if auto programming has finished. The FMR00 bit is "0" during auto programming and set to "1" when auto programming is completed.

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

Each block can be protected against programming by a lock bit. (Refer to "Data Protect Function.")

Writing over already programmed addresses is inhibited.

In EW1 mode, do not execute this command on any address at which the rewrite control program is located.

In EW0 mode, the microcomputer goes to read status register mode at the same time auto programming starts, making it possible to read the status register. The status register bit 7 (SR7) is cleared to "0" at the same time auto programming starts, and set back to "1" when auto programming finishes. In this case, the microcomputer remains in read status register mode until a read command is written next. The result of auto programming can be known by reading the status register after auto programming has finished.

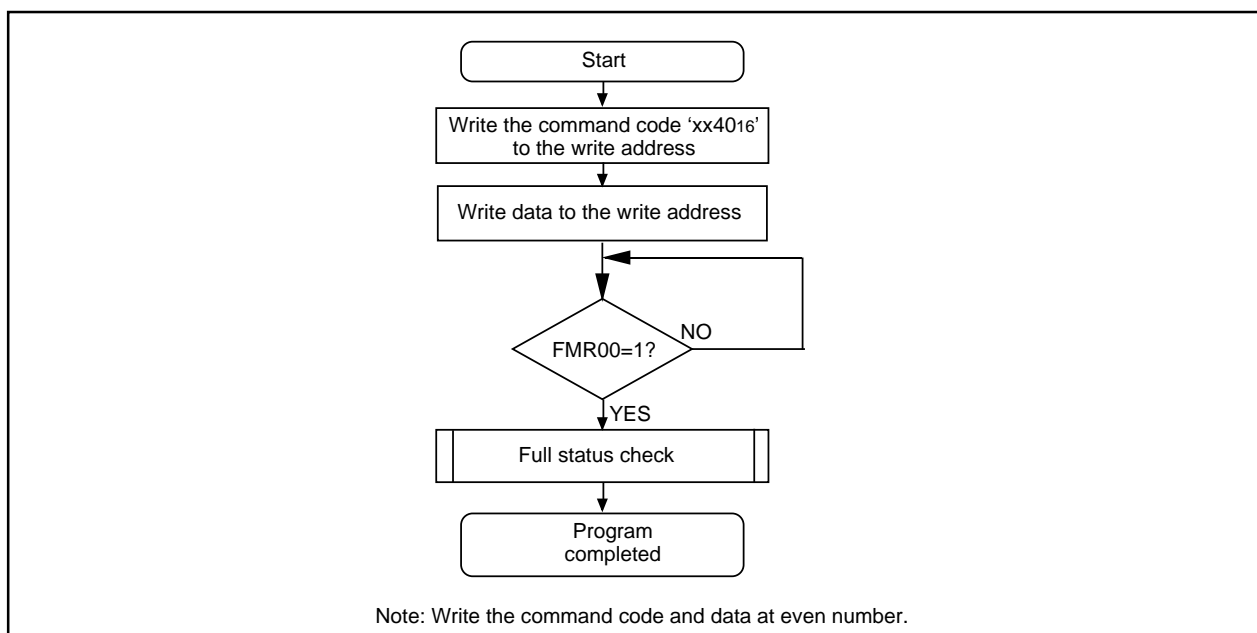


Figure 1.27.8. Program Command

### Block Erase

Write 'xx2016' in the first bus cycle and write 'xxD016' to the uppermost address of a block (even address, however) in the second bus cycle, and an auto erase operation (erase and verify) will start. Check the FMR0 register's FMR00 bit to see if auto erasing has finished.

The FMR00 bit is "0" during auto erasing and set to "1" when auto erasing is completed.

Check the FMR0 register's FMR07 bit after auto erasing has finished, and the result of auto erasing can be known. (Refer to "Full Status Check.")

Figure 1.27.9 shows an example of a block erase flowchart.

Each block can be protected against erasing by a lock bit. (Refer to "Data Protect Function.")

Writing over already programmed addresses is inhibited.

In EW1 mode, do not execute this command on any address at which the rewrite control program is located.

In EW0 mode, the microcomputer goes to read status register mode at the same time auto erasing starts, making it possible to read the status register. The status register bit 7 (SR7) is cleared to "0" at the same time auto erasing starts, and set back to "1" when auto erasing finishes. In this case, the microcomputer remains in read status register mode until the Read Array or Read Lock Bit Status command is written next.

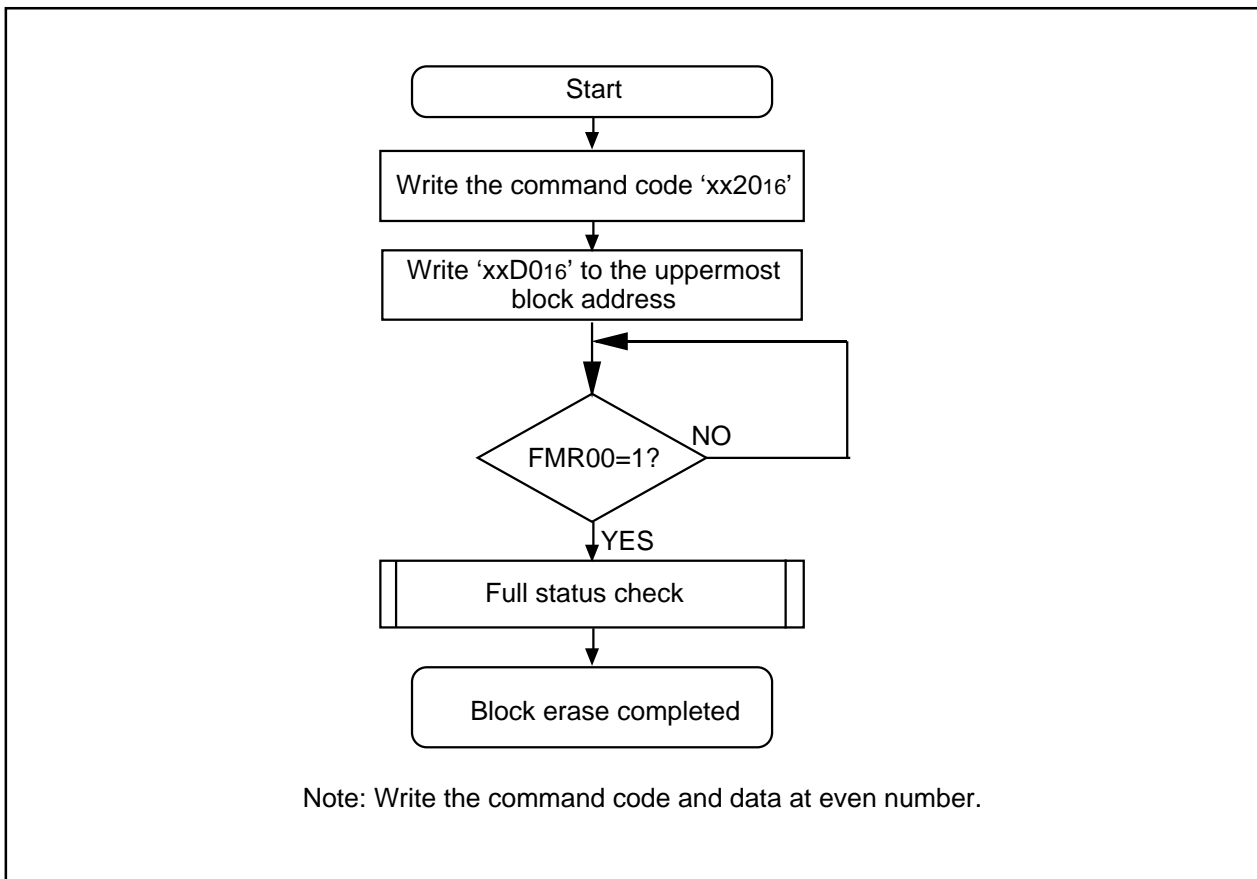


Figure 1.27.9. Block Erase Command

## Flash Memory

**Erase All Unlocked Block**

Write 'xxA716' in the first bus cycle and write 'xxD016' in the second bus cycle, and all blocks except block A will be erased successively, one block at a time.

Check the FMR0 register's FMR00 bit to see if auto erasing has finished. The result of the auto erase operation can be known by inspecting the FMR0 register's FMR07 bit.

Each block can be protected against erasing by a lock bit. (Refer to "Data Protect Function.")

In EW1 mode, do not execute this command when the lock bit for any block = 1 (unlocked) in which the rewrite control program is stored, or when the FMR0 register's FMR02 bit = 1 (lock bit disabled).

In EW0 mode, the microcomputer goes to read status register mode at the same time auto erasing starts, making it possible to read the status register. The status register bit 7 (SR7) is cleared to "0" at the same time auto erasing starts, and set back to "1" when auto erasing finishes. In this case, the microcomputer remains in read status register mode until the Read Array or Read Lock Bit Status command is written next.

Note that only blocks 0 to 12 can be erased by the Erase All Unlocked Block command. Block A cannot be erased. Use the Block Erase command to erase block A.

**Lock Bit Program Command (7716/D016)**

This command sets the lock bit for a specified block to "0" (locked).

Write 'xx7716' in the first bus cycle and write 'xxD016' 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 the address value specified in the first bus cycle is the same uppermost block address that is specified in the second bus cycle.

Figure 1.27.10 shows an example of a lock bit program flowchart. The lock bit status (lock bit data) can be read using the Read Lock Bit Status command.

Check the FMR0 register's FMR00 bit to see if writing has finished.

For details about the lock bit function, and on how to set the lock bit to "1", refer to "Data Protect Function."

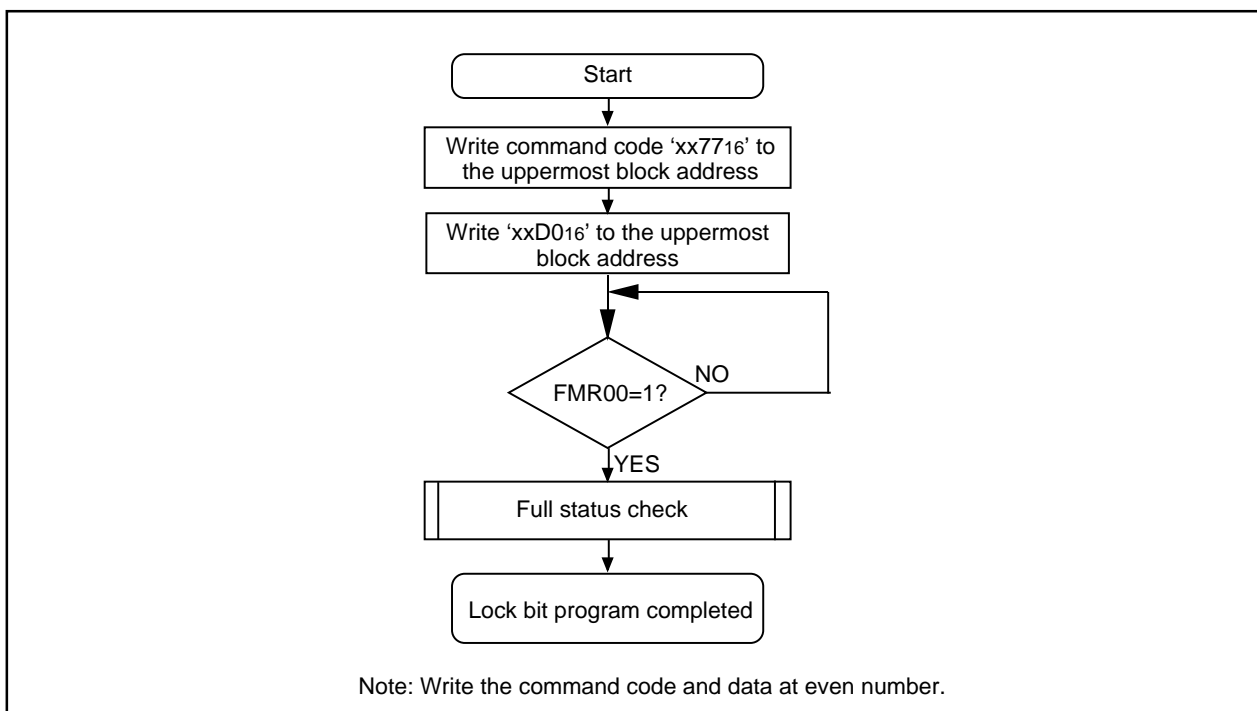


Figure 1.27.10. Lock Bit Program Command

### Read Lock Bit Status Command (7116)

This command reads the lock bit status of a specified block.

Write 'xx7116' in the first bus cycle and write 'xxD016' to the uppermost address of a block (even address, however) in the second bus cycle, and the lock bit status of the specified block is stored in the FMR1 register's FMR16 bit. Read the FMR16 bit after the FMR0 register's FMR00 bit is set to "1" (ready).

Figure 1.27.11 shows an example of a read lock bit status flowchart.

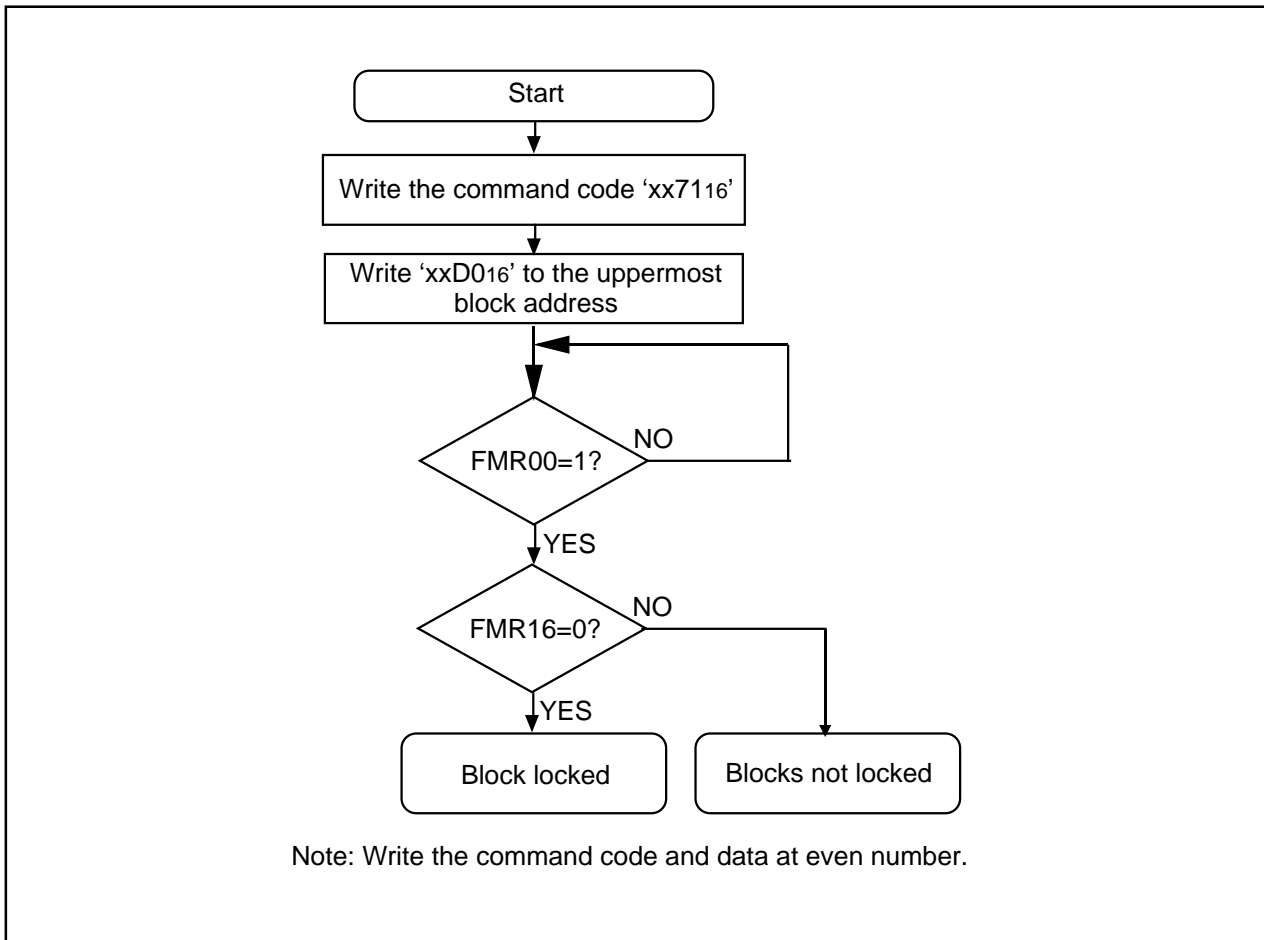


Figure 1.27.11. Read Lock Bit Status Command

## Flash Memory

---

### Data Protect Function

Each block in the flash memory has a nonvolatile lock bit. The lock bit is effective when the FMR02 bit = 0 (lock bit enabled). The lock bit allows each block to be individually protected (locked) against programming and erasure. This helps to prevent data from inadvertently written to or erased from the flash memory. The following shows the relationship between the lock bit and the block status.

- When the lock bit = 0, the block is locked (protected against programming and erasure).
- When the lock bit = 1, the block is not locked (can be programmed or erased).

The lock bit is cleared to "0" (locked) by executing the Lock Bit Program command, and is set to "1" (unlocked) by erasing the block. The lock bit cannot be set to "1" by a command.

The lock bit status can be read using the Read Lock Bit Status command

The lock bit function is disabled by setting the FMR02 bit to "1", with all blocks placed in an unlocked state. (The lock bit data itself does not change state.) Setting the FMR02 bit to "0" enables the lock bit function (lock bit data retained).

If the Block Erase or Erase All Unlocked Block command is executed while the FMR02 bit = 1, the target block or all blocks are erased irrespective of how the lock bit is set. The lock bit for each block is set to "1" after completion of erasure.

For details about the commands, refer to "Software Commands."

### Status Register

The status register indicates the operating status of the flash memory and whether an erase or programming operation terminated normally or in error. The status of the status register can be known by reading the FMR0 register's FMR00, FMR06, and FMR07 bits.

Table 1.27.5 shows the status register.

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

- (1) When a given even address in the user ROM area is read after writing the Read Status Register command
- (2) When a given even address in the user ROM area is read after executing the Program, Block Erase, Erase All Unlocked Block, or Lock Bit Program command but before executing the Read Array command.

#### Sequencer Status (SR7 and FMR00 Bits )

The sequence status indicates the operating status of the flash memory. SR7 = 0 (busy) during auto programming, auto erase, and lock bit write, and is set to "1" (ready) at the same time the operation finishes.

#### Erase Status (SR5 and FMR07 Bits)

Refer to "Full Status Check."

#### Program Status (SR4 and FMR06 Bits)

Refer to "Full Status Check."

Table 1.27.5. Status Register

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

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



## Flash Memory

## Full Status Check

When an error occurs, the FMR0 register's FMR06 to FMR07 bits are set to "1", indicating occurrence of each specific error. Therefore, execution results can be verified by checking these status bits (full status check). Table 1.27.6 lists errors and FMR0 register status. Figure 1.27.12 shows a full status check flowchart and the action to be taken when each error occurs.

**Table 1.27.6. Errors and FMR0 Register Status**

FMR00 register (status register) status		Error	Error occurrence condition
FMR07 (SR5)	FMR06 (SR4)		
1	1	Command sequence error	<ul style="list-style-type: none"> <li>• When any command is not written correctly</li> <li>• When invalid data was written other than those that can be written in the second bus cycle of the Lock Bit Program, Block Erase, or Erase All Unlocked Block command (i.e., other than 'xxD016' or 'xxFF16') (Note 1)</li> </ul>
1	0	Erase error	<ul style="list-style-type: none"> <li>• When the Block Erase command was executed on locked blocks (Note 2)</li> <li>• When the Block Erase or Erase All Unlocked Block command was executed on unlocked blocks but the blocks were not automatically erased correctly</li> </ul>
0	1	Program error	<ul style="list-style-type: none"> <li>• When the Block Erase command was executed on locked blocks (Note 2)</li> <li>• When the Program command was executed on unlocked blocks but the blocks were not automatically programmed correctly.</li> <li>• When the Lock Bit Program command was executed but not programmed correctly</li> </ul>

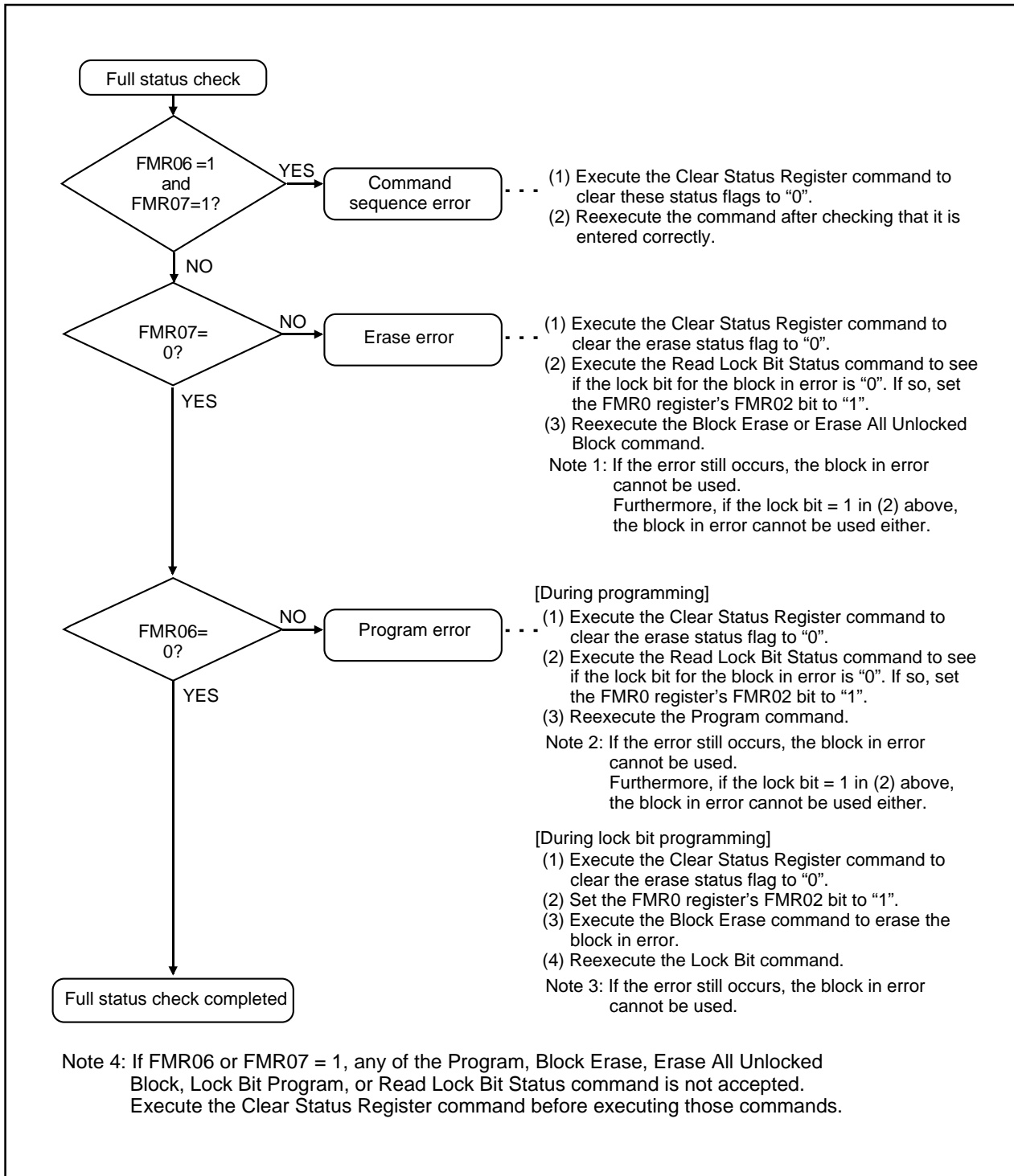


Figure 1.27.12. Full Status Check and Handling Procedure for Each Error

### **Standard Serial I/O Mode**

In standard serial input/output mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer suitable for the M16C/62P group. For more information about serial programmers, contact the manufacturer of your serial programmer. For details on how to use, refer to the user's manual included with your serial programmer.

Table 1.27.7 lists pin functions (flash memory standard serial input/output mode). Figures 1.27.13 to 1.27.15 show pin connections for serial input/output mode.

### **ID Code Check Function**

This function determines whether the ID codes sent from the serial programmer and those written in the flash memory match. (Refer to the description of the functions to inhibit rewriting flash memory version.)

Table 1.27.7. Pin Functions (Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Description
Vcc, Vss	Power input		Apply the voltage guaranteed for Program and Erase to Vcc pin and 0 V to Vss pin.
CNVss	CNVss	I	Connect to Vcc pin.
RESET	Reset input	I	Reset input pin. While 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 Vcc or Vss.
AVcc, AVss	Analog power supply input		Connect AVss to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for AD from this pin.
P00 to P07	Input port P0	I	Input "H" or "L" level signal or open.
P10 to P17	Input port P1	I	Input "H" or "L" level signal or open.
P20 to P27	Input port P2	I	Input "H" or "L" level signal or open.
P30 to P37	Input port P3	I	Input "H" or "L" level signal or open.
P40 to P47	Input port P4	I	Input "H" or "L" level signal or open.
P51 to P54, P56, P57	Input port P5	I	Input "H" or "L" level signal or open.
P50	CE input	I	Input "H" level signal.
P55	EPM input	I	Input "L" level signal.
P60 to P63	Input port P6	I	Input "H" or "L" level signal or open.
P64	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.
P65	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input "L".
P66	RxD input	I	Serial data input pin
P67	TxD output	O	Serial data output pin (Note 1)
P70 to P77	Input port P7	I	Input "H" or "L" level signal or open.
P80 to P84, P86, P87	Input port P8	I	Input "H" or "L" level signal or open.
P85	NMI input	I	Connect this pin to Vcc.
P90 to P97	Input port P9	I	Input "H" or "L" level signal or open. (Note 2)
P100 to P107	Input port P10	I	Input "H" or "L" level signal or open. (Note 2)
P110 to P117	Input port P11	I	Input "H" or "L" level signal or open. (Note 2)
P120 to P127	Input port P12	I	Input "H" or "L" level signal or open. (Note 2)
P130 to P137	Input port P13	I	Input "H" or "L" level signal or open. (Note 2)
P140 to P147	Input port P14	I	Input "H" or "L" level signal or open. (Note 2)

Note 1: When using standard serial input/output mode 1, the TxD pin must be held high while the RESET pin is pulled low. Therefore, connect this pin to Vcc via a resistor. Because this pin is directed for data output after reset, adjust the pull-up resistance value in the system so that data transfers will not be affected.

Note 2: Available in only the 128-pin version.

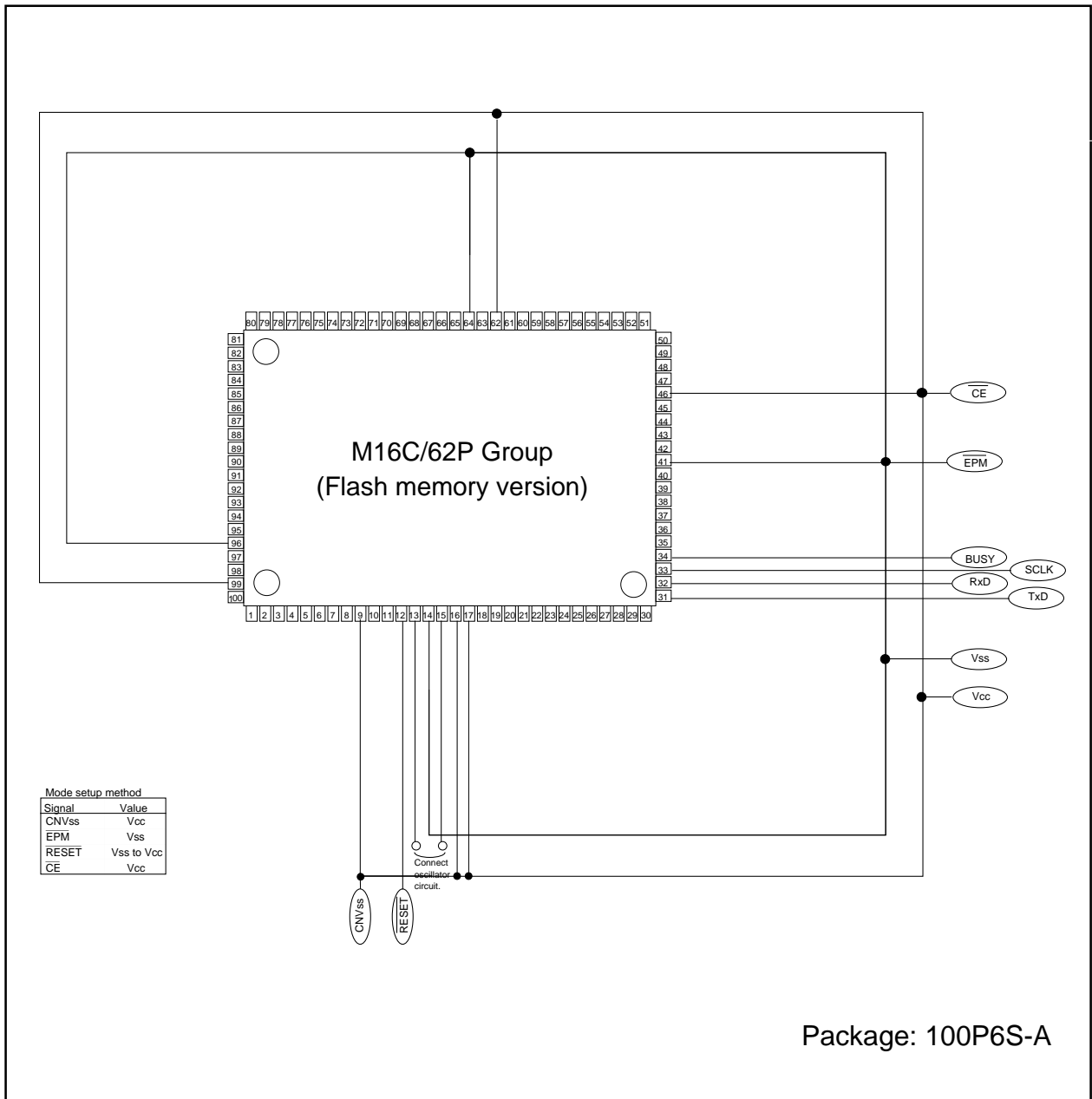


Figure 1.27.13. Pin Connections for Serial I/O Mode (1)

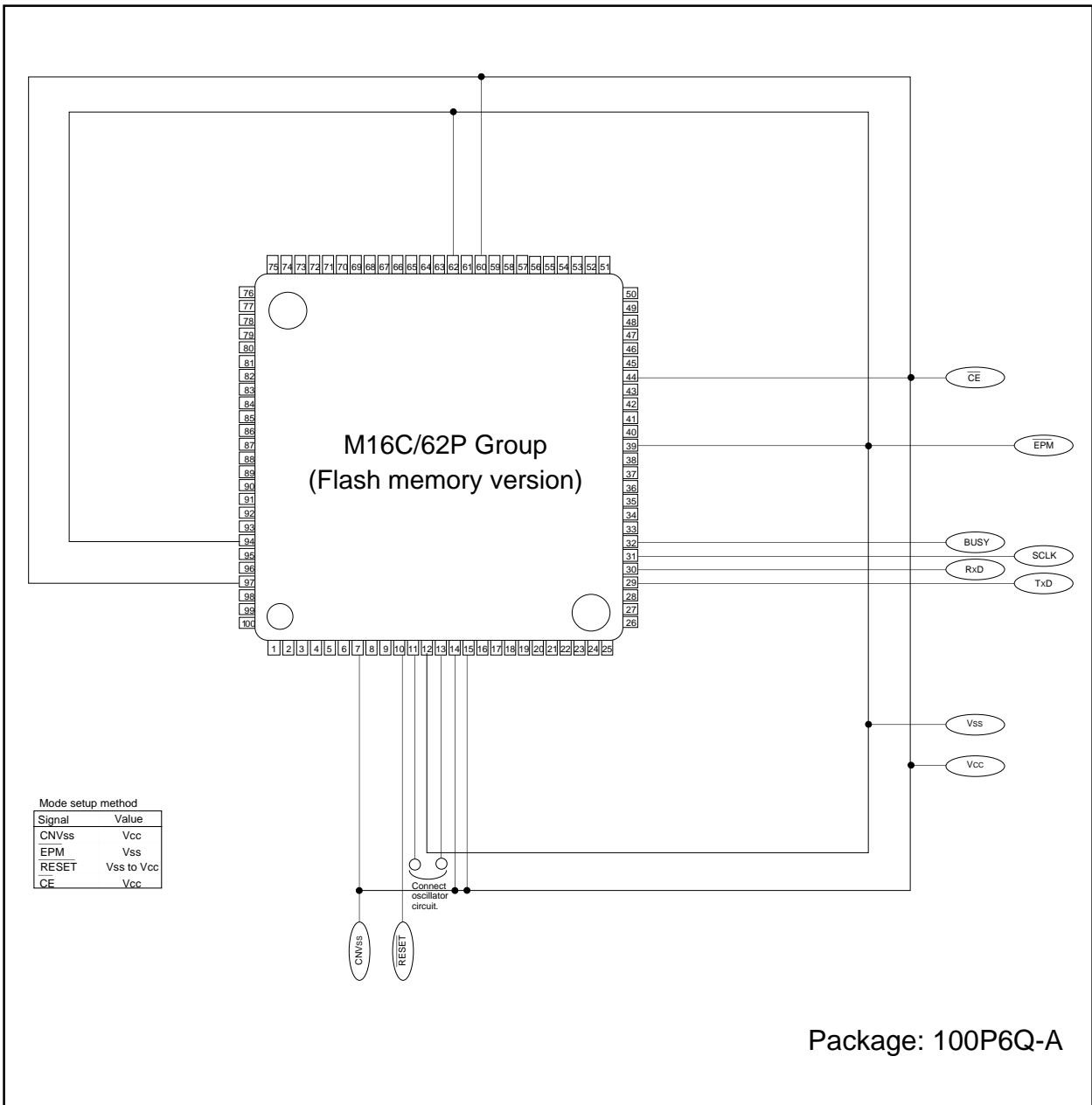


Figure 1.27.14. Pin Connections for Serial I/O Mode (2)

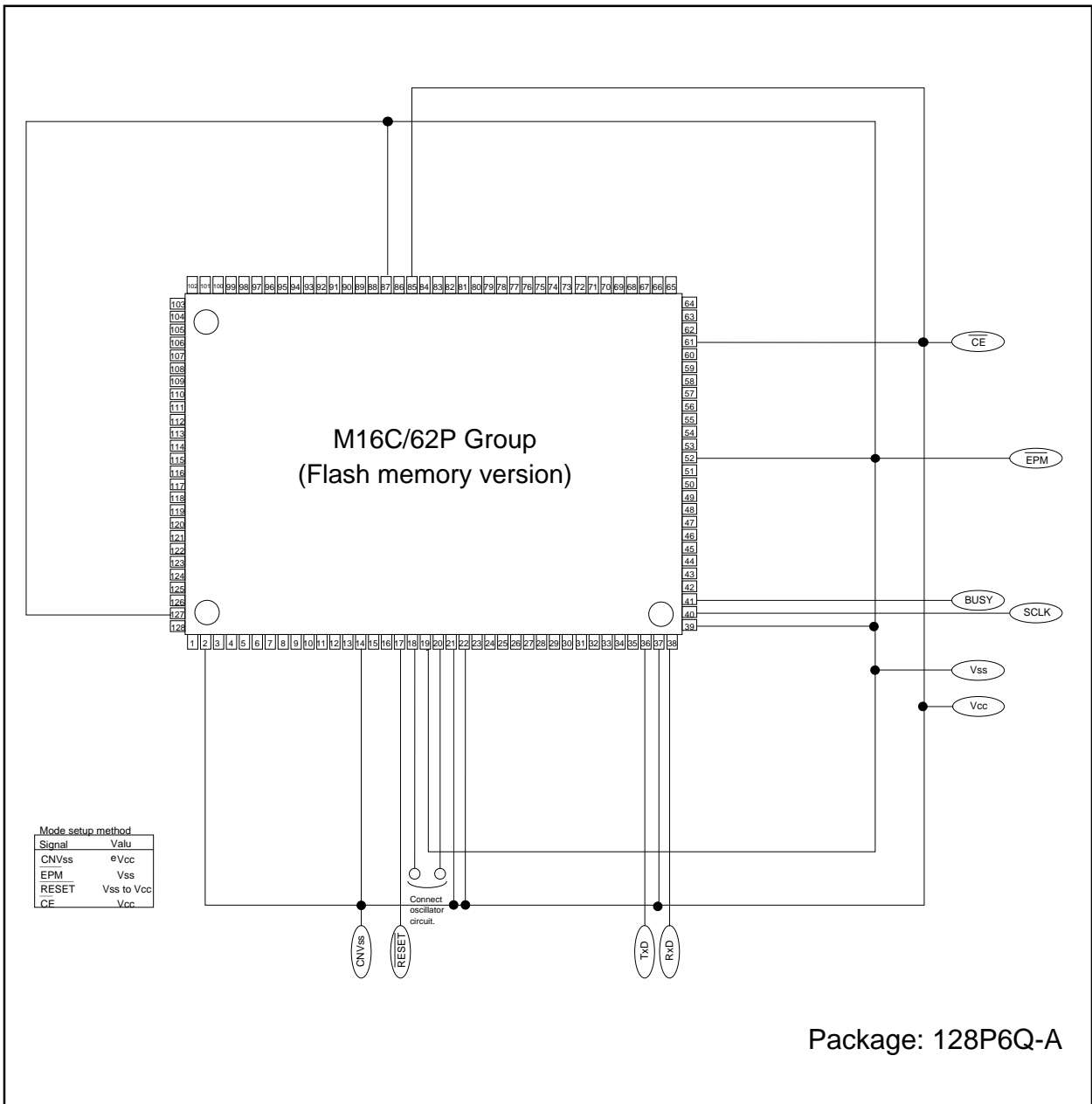
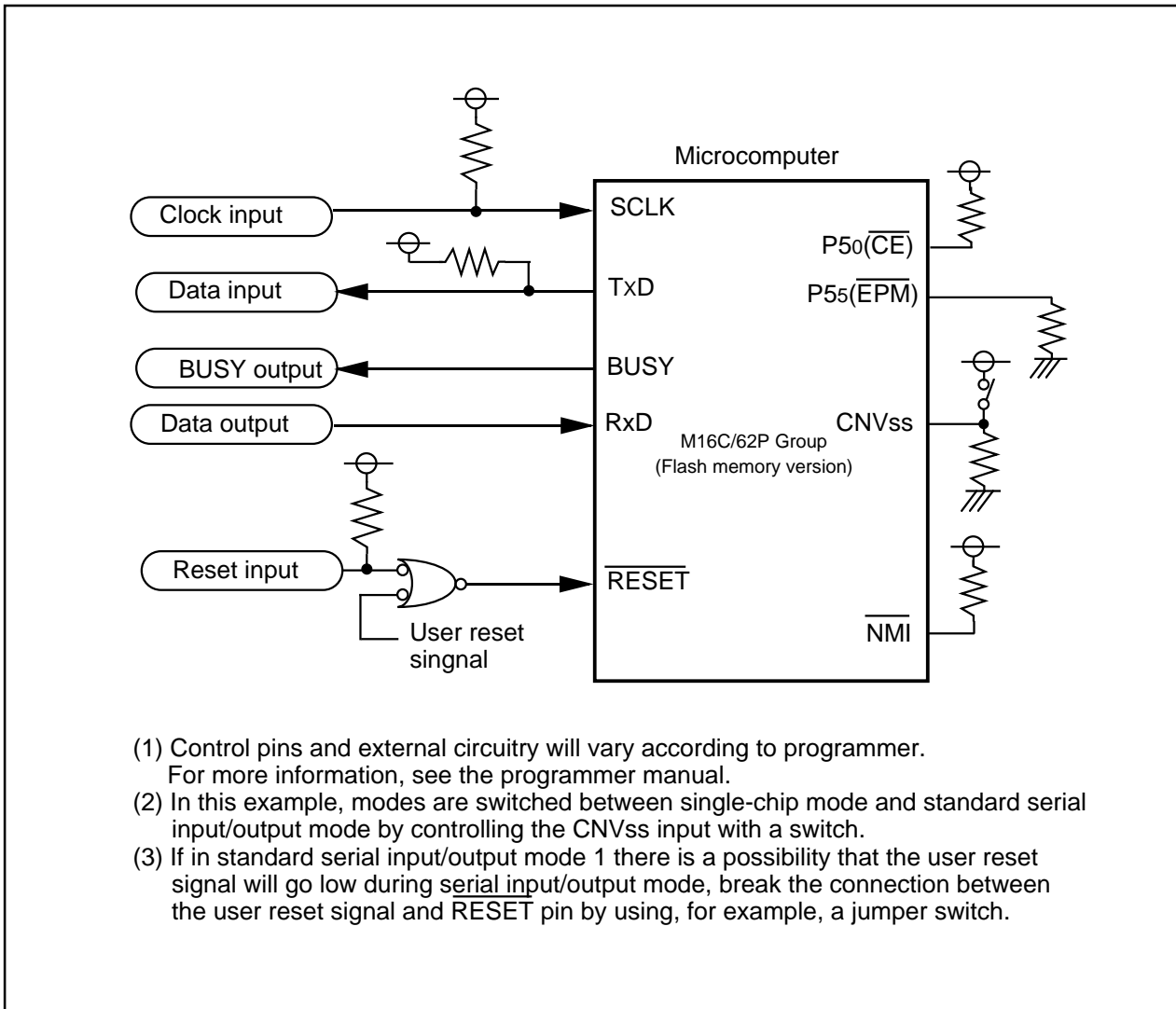


Figure 1.27.15. Pin Connections for Serial I/O Mode (3)

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

Figure 1.27.16 and 1.27.17 show example of circuit application in standard serial I/O mode 1 and mode 2, respectively. Refer to the user's manual for serial writer to handle pins controlled by a serial writer.



- (1) Control pins and external circuitry will vary according to programmer.  
 For more information, see the programmer manual.
- (2) In this example, modes are switched between single-chip mode and standard serial input/output mode by 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 1.27.16. Circuit Application in Standard Serial I/O Mode 1



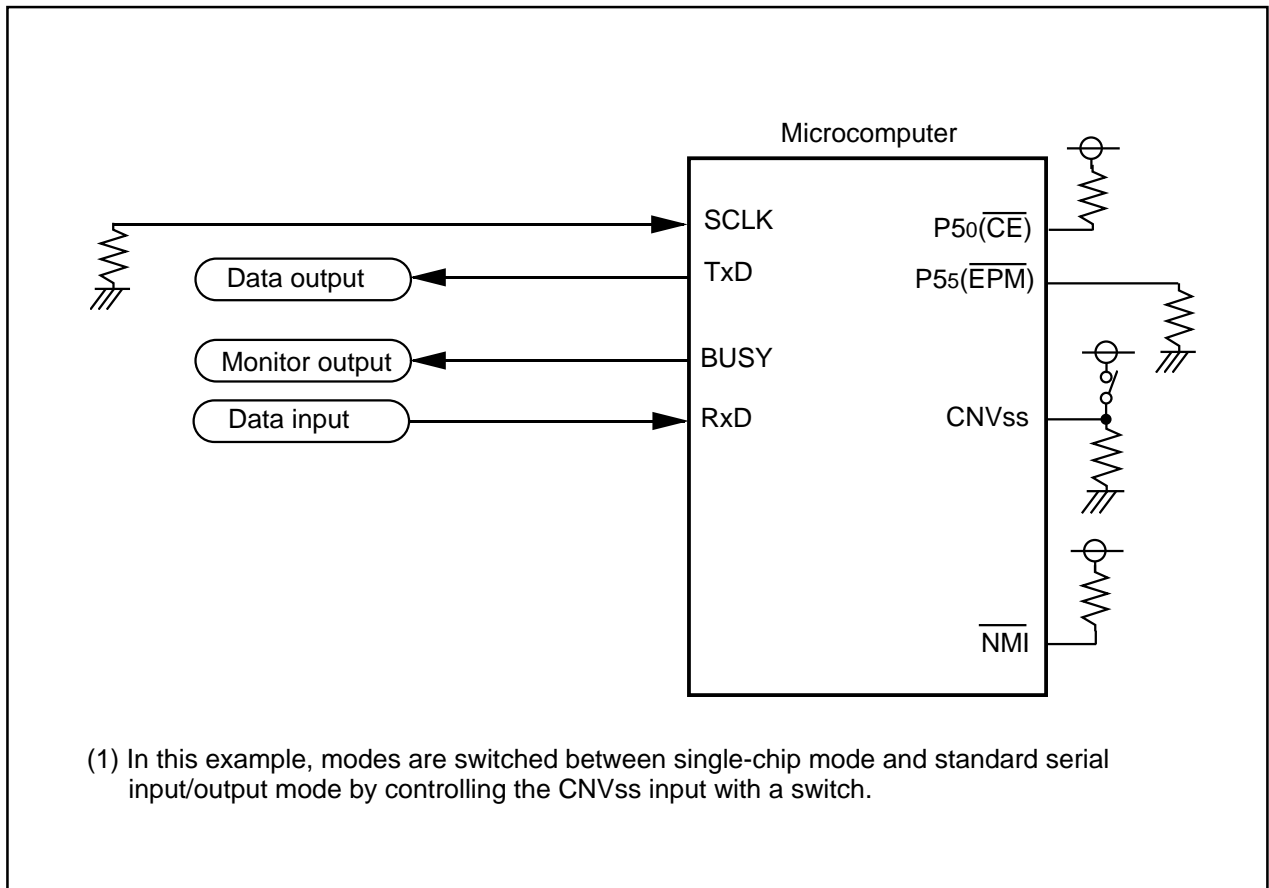


Figure 1.27.17. Circuit Application in Standard Serial I/o Mode 2

## Parallel I/O Mode

In parallel input/output mode, the user ROM and boot ROM areas can be rewritten by using a parallel programmer suitable for the M16C/62P group. For more information about parallel programmers, contact the manufacturer of your parallel programmer. For details on how to use, refer to the user's manual included with your parallel programmer.

## User ROM and Boot ROM Areas

In the boot ROM area, an erase block operation is applied to only one 4 Kbyte block. The boot ROM area contains a standard serial input/output mode based rewrite control program which was written in it when shipped from the factory. Therefore, when using a serial programmer, be careful not to rewrite the boot ROM area.

When in parallel output mode, the boot ROM area is located at addresses  $0FF000_{16}$  to  $0FFFFFF_{16}$ . When rewriting the boot ROM area, make sure that only this address range is rewritten. (Do not access other than the addresses  $0FF000_{16}$  to  $0FFFFFF_{16}$ .)

## ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten. (Refer to the description of the functions to inhibit rewriting flash memory version.)

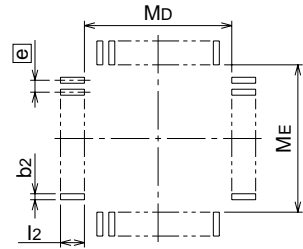
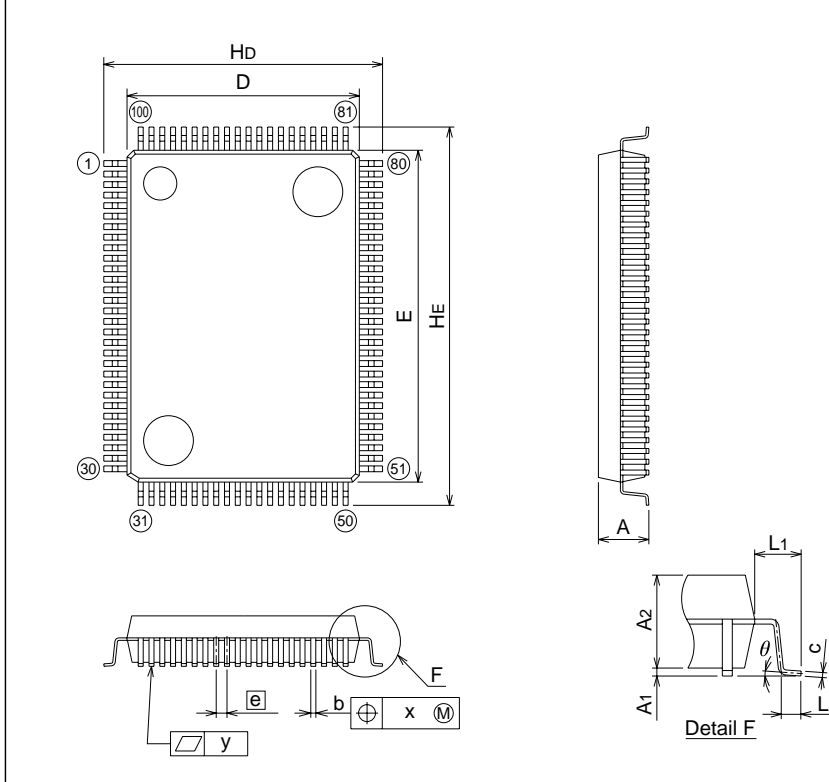
Package Outline

Package Outline

100P6S-A (MMP)

Plastic 100pin 14X20mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP100-P-1420-0.65	-	1.58	Alloy 42



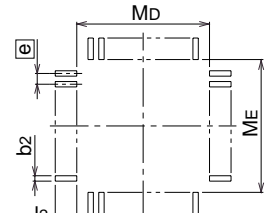
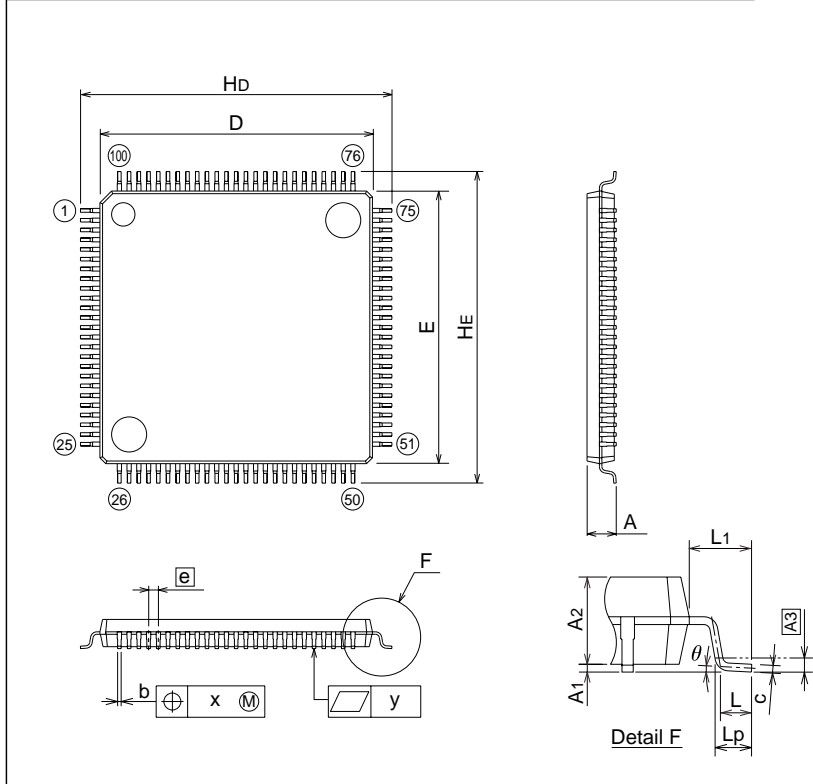
Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	3.05
A1	0	0.1	0.2
A2	-	2.8	-
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	-	0.65	-
Hd	16.5	16.8	17.1
HE	22.5	22.8	23.1
L	0.4	0.6	0.8
L1	-	1.4	-
x	-	-	0.13
y	-	-	0.1
θ	0°	-	10°
b2	-	0.35	-
l2	1.3	-	-
Md	-	14.6	-
ME	-	20.6	-

100P6Q-A (MMP)

Plastic 100pin 14X14mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP100-P-1414-0.50	-	0.63	Cu Alloy



Recommended Mount Pad

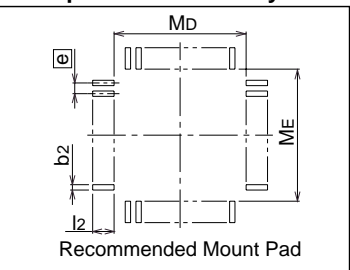
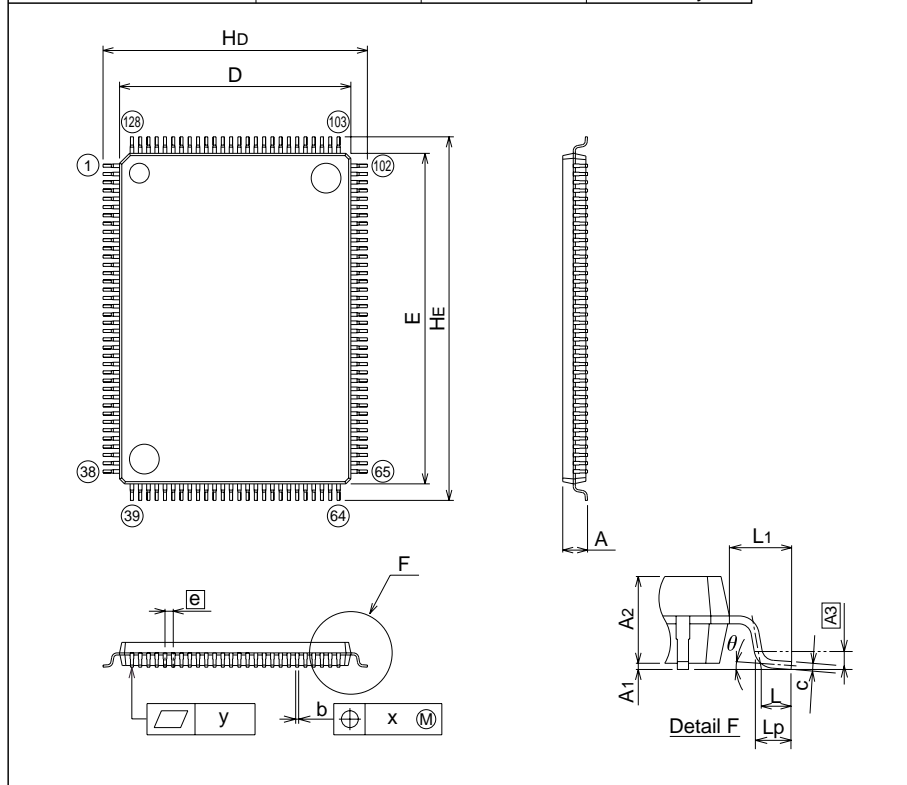
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	1.7
A1	0	0.1	0.2
A2	-	1.4	-
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	13.9	14.0	14.1
E	13.9	14.0	14.1
e	-	0.5	-
Hd	15.8	16.0	16.2
HE	15.8	16.0	16.2
L	0.3	0.5	0.7
L1	-	1.0	-
Lp	0.45	0.6	0.75
A3	-	0.25	-
x	-	-	0.08
y	-	-	0.1
θ	0°	-	10°
b2	-	0.225	-
l2	0.9	-	-
MD	-	14.4	-
ME	-	14.4	-

**128P6Q-A**

(MMP)

Plastic 128pin 14X20mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP128-P-1420-0.50	-	-	Cu Alloy



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	1.4	1.5	1.7
A1	0.05	0.125	0.2
A2	-	1.4	-
b	0.17	0.22	0.27
c	0.105	0.125	0.175
D	13.9	14.0	14.1
E	19.9	20.0	20.1
e	-	0.5	-
Hd	15.8	16.0	16.2
HE	21.8	22.0	22.2
L	0.35	0.5	0.65
L1	-	1.0	-
Lp	0.45	0.6	0.75
A3	-	0.25	-
x	-	-	0.08
y	-	-	0.1
θ	0°	-	8°
b2	-	0.225	-
l2	-	1.0	-
MD	-	14.4	-
ME	-	20.4	-

## Differences Between M16C/62P and M16C/62A

## Differences in Mask ROM Version and Flash Memory Version (1) (Note)

Item	M16C/62P	M16C/62A
Shortest 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=4.2V to 5.5V) 100ns (f(XIN)=10MHz, VCC=2.7V to 5.5V with software one-wait)
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)	4.2V to 5.5V (f(XIN)=16MHz, without software wait) 2.7V to 5.5V (f(XIN)=10MHz, with software one-wait)
I/O power supply	Double (VCC1, VCC2)	Single (VCC)
Package	100-pin, 128-pin plastic mold QFP	80-pin, 100-pin plastic mold QFP
Voltage detection circuit	Built-in Vdet2, Vdet3, Vdet4 detect Power supply voltage down detect interrupt Hardware reset 2	None
Clock Generating Circuit	PLL, XIN, XCIN, ring oscillator Main clock division rate when main clock is stopped: Divide-by-8 frequency XIN drive capacity when main clock is stopped: HIGH	XIN, XCIN Main clock division rate when main clock is stopped: No change XIN drive capacity when main clock is stopped: No change
System clock protective function	Built-in	None (protected by protect register)
Oscillation stop, re-oscillation detection function	Built-in	None
Low power consumption	18mA (VCC1=VCC2=5V, f(BCLK)=24MHz) 8mA (VCC1=VCC2=3V, f(BCLK)=10MHz) 1.8μA (VCC1=VCC2=3V, f(XCIN)=32kHz, when wait mode)	32.5mA (VCC=5V, f(XIN)=16MHz) 8.5mA (VCC=3V, f(XCIN)=10MHz with software one-wait) 0.9μA (VCC=3V, f(XCIN)=32kHz, when wait mode)
Memory area	Memory area expandable (4 Mbytes)	1 Mbytes fixed
External device connect area	0400016-07FFF16(PM13=0) 0800016-0FFFF16(PM10=0) 1000016-26FFF16 2800016-7FFF16 8000016-CFFF16(PM13=0) D000016-FFFF16(Microprocessor mode)	0400016-05FFF16(PM13=0) 0600016-CFFF16 D000016-FFFF16(Microprocessor mode)
Upper address in memory expansion mode and microprocessor mode	P40 to P43 (A16 to A19), P34 to P37 (A12 to A15) : Switchable between address bus and port	P40 to P43 (A16 to A19) : Switchable between address bus and port
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, S3C, S4C 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
Address match interrupt	4	2

Note: About the details and the electric characteristics, refer to data sheet.

## Differences in Mask ROM version and Flash memory version (2) (Note)

Item	M16C/62P	M16C/62A
Timers A, B count source	Selectable: f1, f2, f8, f32, fc32	Selectable: f1, f8, f32, fc32
Timer A two-phase pulse signal processing	Z-phase (counter reset) input is available	No Z-phase (counter reset input)
Timer functions for three-phase motor control	Function protect by protect register Count source is selectable: f1, f2, f8, f32, fc32 Dead time timer count source is selectable: f1, f1 divided by 2, f2, f2 divided by 2 Output polarity is selectable Carrier wave phase detectable Three-phase output port NMI control	No function protect by protect register Count source is selectable: f1, f8, f32, fc32 Dead time timer count source is fixed at f1/2
Serial I/O (UART0 to UART2)	(UART, clock synchronous, I <sup>2</sup> C bus, IE bus) x 3	(UART, clock synchronous,) x 2 (UART, clock synchronous, IIC bus, IE bus) x 1
UART0 to UART2, SI/O3, SI/O4 count source	Selectable: f1SIO, f2SIO, f8SIO, f32SIO	Selectable: f1, f8, f32
Serial I/O $\overline{\text{RTS}}$ timing	Assert low when receive buffer is read	Assert low when reception is completed
CTS/RTS separate function	Have	None
UART2 data transmit timing	After data was written, transfer starts at the 2nd BRG overflow timing (same as UART0 and UART1)	After data was written, transfer starts at the 1st BRG overflow timing (Output starts one cycle of BRG overflow earlier than UART0 and UART1)
Serial I/O sleep function	None	Have
Serial I/O I <sup>2</sup> C mode	Start condition, stop condition: Auto-generationable	Start condition, stop condition: Not auto-generationable
Serial I/O I <sup>2</sup> C mode SDA delay	Only digital delay is selected as SDA delay SDA digital delay count source: BRG	Analog or digital delay is selected as SDA delay SDA digital delay count source: 1/f(XIN)
SI/O3, SI/O4 clock polarity selection	Selectable	Not selectable
A-D converter	10 bits X 8 channels Expandable up to 26 channels	10 bits X 8 channels Expandable up to 10 channels
A-D converter operation clock	Selectable: fAD, fAD divided by 2, 3, 4, 6, 12	Selectable: fAD, fAD/2, fAD/4
A-D converter input pin	Selectable: ports P0, P2, P10	Fixed at port P10

Note: About the details and the electric characteristics, refer to data sheet.

Under development

Differences in Flash memory version(Note)

Item	M16C/62P	M16C/62A
User ROM blocks	14 blocks: 4 Kbytes x 3, 8 Kbytes x 3, 32 Kbytes x 1, 64 Kbytes x 7 (Flash memory: max. 512 Kbytes)	7 blocks: 8 Kbytes x 2, 16 Kbytes x 1, 32 Kbytes x 1, 64 Kbytes x 3 (Flash memory: max. 256 Kbytes)
Program manner	Word	Page
Program command (software command)	Page program command: none Program command: have (program method: in units of word, in units of byte)	Page program command: have Program command: none (program method: in units of page)
Block status after program function	Have	None
CPU rewrite mode	EW1 mode is available	No EW1 mode

Note: About the details and the electric characteristics, refer to data sheet.

# Register Index

## A

AD0 to AD7 192

ADCON0 191

ADCON1 191

ADCON2 192

AIER 92

AIER2 92

## C

CM0 53

CM1 54

CM2 55

CPSRF 111, 124

CRCD 208

CRCIN 208

CSE 40

CSR 34

## D

D4INT 25

DA0 207

DA1 207

DACON 207

DAR0 101

DAR1 101

DBR 44

DM0CON 100

DM1CON 100

DM1SL 100

DTT 133

## F

FIDR 268

FMR0 268

## I

ICTB2 134

IDB0 133

IDB1 133

IFSR 89

IFSR2A 89

INVC0 131

INVC1 132

## O

ONSF 111

## P

P0 to P13 217

PC14 218

PCLKR 56

PCR 220

PD0 to PD13 216

PLC0 57

PM0 30

PM1 31

PM2 56

PRCR 74

PUR0 to PUR2 219

PUR3 218

## R

RMAD0 to RMAD3 92

ROMCP 264

## S

S3BRG 185

S3C 185

S3TRR 185

S4BRG 185

S4C 185

S4TRR 185

SAR0 101

SAR1 101

## T

TA0 to TA4 110

TA0MR to TA4MR 109

TA1 134

TA11 134

TA1MR 136

TA2 134

TA21 134



**Register Index**

TA2MR 136  
TA4 134  
TA41 134  
TA4MR 136  
TABSR 110, 124, 135  
TB0 to TB5 124  
TB0MR to TB5MR 123  
TB2 135  
TB2MR 136  
TB2SC 134  
TBSR 124  
TCR0 101  
TCR1 101  
TRGSR 111, 135

**U**

U0BRG to U2BRG 142  
U0C0 to U2C0 143  
U0C1 to U2C1 144  
U0MR to U2MR 143  
U0RB to U2RB 142  
U0SMR to U2SMR 145  
U0SMR2 to U2SMR2 146  
U0SMR3 to U2SMR3 146  
U0SMR4 to U2SMR4 147  
U0TB to U2TB 142  
UCON 145  
UDF 110

**V**

VCR1 25  
VCR2 25

**W**

WDC 24, 96  
WDTS 96

REVISION HISTORY

M16C/62P GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
1.0	Jan/31/Y03 (Continued)	1	Applications are partly revised.
		2	Table 1.1.1 is partly revised.
		5	Table 1.1.3 is partly revised.
		5	Figure 1.1.2 is partly revised.
		11	Explanation of "Memory" is partly revised.
		20	Explanation of "Hardware Reset 1" is partly revised.
		21	Figure 1.5.1 is partly revised.
		22	Figure 1.5.2 is partly revised.
		24	Figure 1.5.4 is partly revised.
		25	VCR2 Register in Figure 1.5.6 is partly revised.
		26	Figure 1.5.6 is partly revised.
		27	Explanation of "Power Supply Down Detection Interrupt" is partly revised.
		30	Figure 1.6.1 is partly revised.
		31	Figure 1.6.2 is partly revised.
		39	Table 1.7.5 is partly revised.
		41	Table 1.7.7 is partly revised.
		43	Figure 1.7.8 is partly revised.
		44	Explanation of "4 Mbyte Mode" is partly revised.
		53	Notes 12 and 13 in Figure 1.9.2 is partly revised.
		54	Notes 2 and 5 in Figure 1.9.3 is partly revised.
		55	Figure 1.9.4 is partly revised.
		57	Note 4 in Figure 1.9.6 is partly revised.
		60	Explanation of "PLL Clock" is partly revised.
		61	Figure 1.9.9 is partly revised.
		62	Explanation of "CPU Clock and BCLK" is partly revised.
		63	Explanation of "Low-speed Mode" is partly revised.
		63	Explanation of "Low Power Dissipation Mode" is partly revised.
		64	Explanation of "Ring Oscillator Low Power Dissipation Mode" is partly revised.
		64	Table 1.9.3 is partly revised.
		65	Table 1.9.5 is partly revised.
68	Figure 1.9.10 is partly revised.		
69	Figure 1.9.11 is partly revised.		
70	Table 1.9.7 is added.		
71	Explanation of "System Clock Protective Function" is partly revised.		
77	Explanation of "Power Supply Down Detection Interrupt" is partly revised.		
78	Table 1.11.1 is partly revised.		
88	Figure 1.11.9 is partly revised.		
96	WDTS Register in Figure 1.12.2 is partly revised.		
99	Figure 1.13.2 is partly revised.		
100	Figure 1.13.3 is partly revised.		
103	Figure 1.13.5 is partly revised.		
104	Table 1.13.3 is partly revised.		
105	Explanation of "DMA Enable" is partly revised.		
109	Figure 1.14.3 is partly revised.		
115	Table 1.14.3 is partly revised.		
117	Explanation of "Counter Initialization by Two-Phase Pulse Signal Processing" is partly revised.		
117	Figure 1.14.10 is partly revised.		
122	Figure 1.14.14 is partly revised.		
122	Figure 1.14.15 is partly revised.		

REVISION HISTORY

M16C/62P GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
1.0	Jan/31/Y03 (Continued)	124	Figure 1.15.3 is partly revised.
		128	Figure 1.15.7 is partly revised.
		128	Figure 1.15.8 is partly revised.
		130	Figure 1.16.1 is partly revised.
		132	Figure 1.16.3 is partly revised.
		134	Note 7 is added to TAI, TAI1 Register in Figure 1.16.5.
		137	Figure 1.16.8 is partly revised.
		146	UiSMR2 Register in Figure 1.17.7 is partly revised.
		163	Figure 1.20.1 is partly revised.
		164, 165	Table 1.20.2 and Table 1.20.3 are partly revised.
		169	Figure 1.20.4 is partly revised.
		169	Explanation of "Arbitration" is partly revised.
		170	Explanation of "Transfer Clock" is partly revised.
		171	Explanation of "ACK and NACK" is partly revised.
		179	Explanation of "Special Mode 4 (SIM Mode)" is partly revised.
		179	Table 1.20.9 is partly revised.
		184	Figure 1.21.1 is partly revised.
		187	Figure 1.21.4 is partly revised.
		203	Explanation of "External Operation Amp Connection Mode" is partly revised.
		205	Explanation of "Caution of Using A-D Converter" is partly revised.
		205	Figure 1.22.11 is partly revised
		206	Table 1.23.1 is partly revised.
		207	Figure 1.23.3 is partly revised.
		218	Figure 1.25.9 is partly revised.
		223	Table 1.26.1 is partly revised.
		224	Table 1.26.2 is partly revised.
		225	Note 1 of Table 1.26.3 is partly revised.
		225	Note 1 of Table 1.26.4 is partly revised.
		225	Table 1.26.6 is partly revised.
		227	Note 1 of Table 1.26.9 is partly revised.
		228	Note 1 of Table 1.26.10 is partly revised.
		229	Measurement conditions of timing requirements are partly revised.
		229	Table 1.26.11 is partly revised.
		230	Measurement conditions of timing requirements are partly revised.
230	Table 1.26.18 is added.		
231	Measurement conditions of timing requirements are partly revised.		
232	Measurement conditions of switching characteristics are partly revised.		
233	Measurement conditions of switching characteristics are partly revised.		
234	Measurement conditions of switching characteristics are partly revised.		
235	Figure 1.26.2 is partly revised.		
242	Figure 1.26.9 is partly revised.		
244	Note of Table 1.26.28 is partly revised.		
245	Figure 1.26.29 is partly revised.		
246	Measurement conditions of timing requirements are partly revised.		
246	Table 1.26.30 is partly revised.		
247	Measurement conditions of timing requirements are partly revised.		
247	Table 1.26.37 is added.		
248	Measurement conditions of timing requirements are partly revised.		
249	Measurement conditions of switching characteristics are partly revised.		
250	Measurement conditions of switching characteristics are partly revised.		

REVISION HISTORY

M16C/62P GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
1.0 (Continued)	Jan/31/Y03	251	Measurement conditions of switching characteristics are partly revised.
		252	Figure 1.26.12 is partly revised.
		255	Figure 1.26.15 is partly revised.
		256	Figure 1.26.16 is partly revised.
		257	Figure 1.26.17 is partly revised.
		258	Figure 1.26.18 is partly revised.
		259	Figure 1.26.19 is partly revised.
		260	Figure 1.26.20 is partly revised.
		262	Explanation of "Memory Map" is partly revised.
		263	Explanation of "Boot Mode" is partly revised.
		264	Figure 1.27.3 is partly revised.
		268	Note of FIDR Register in Figure 1.27.4 is partly revised.
		271	Figure 1.27.7 is partly revised.
		272	Explanation of "Interrupts" is partly revised.
		272	Explanation of "Writing in the User ROM Space" is partly revised.
		274	Table 1.27.4 is partly revised.
		274	Explanation of "Read Array Command" is partly revised.
278	Explanation of "Program Command" is partly revised.		
287	Figure 1.27.15 is partly revised.		
293	Partly revised.		

— Keep safety first in your circuit designs! —

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

— Notes regarding these materials —

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.  
The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.  
Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (<http://www.mitsubishichips.com>).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.  
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

mitsubishi 16-BIT SINGLE-CHIP MICROCOMPUTER  
M16C FAMILY / M16C/60 SERIES

**M16C/62**

(M16C/62P)

**G r o u p**

Usage Notes Reference Book

<http://www.infocom.maec.co.jp/indexe.htm>

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

### Keep safety first in your circuit designs!

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

### Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein. The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.  
Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (<http://www.mitsubishichips.com>).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.  
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

## **Preface**

This book describes the M16C/62 (M16C/62P) group's precautions for use, which contains paragraphs describing precautions of the user's manual and technical news relevant to these paragraphs. Please refer to this book when developing your systems. However, all of precautions are not contained in this book, please perform sufficient evaluation under systems development.



## 1.1 Precautions for Interrupts

---

### 1. Usage Precaution

#### 1.1 Precautions for Interrupts

##### 1.1.1 Reading address 00000<sub>16</sub>

Do not read the address 00000<sub>16</sub> in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000<sub>16</sub> during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to "0". If the address 00000<sub>16</sub> is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to "0". This causes a problem that the interrupt is canceled, or an unexpected interrupt is generated.

## 1.1 Precautions for Interrupts

---

### 1.1.2 Setting the SP

Set any value in the SP before accepting an interrupt. The SP is cleared to '000016' after reset. Therefore, if an interrupt is accepted before setting any value in the SP, the program may go out of control.

Especially when using  $\overline{\text{NMI}}$  interrupt, set a value in the SP 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.

## 1.1 Precautions for Interrupts

---

### 1.1.3 The $\overline{\text{NMI}}$ Interrupt

1. The  $\overline{\text{NMI}}$  interrupt cannot be disabled. If this interrupt is unused, connect the  $\overline{\text{NMI}}$  pin to VCC via a resistor (pull-up).
2. The input level of the  $\overline{\text{NMI}}$  pin can be read by accessing the P8 register's P8\_5 bit. Note that the P8\_5 bit can only be read when determining the pin level after an  $\overline{\text{NMI}}$  interrupt is generated.
3. 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 CM1 register's CM10 bit is fixed to "0".
4. 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.
5. 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.

## 1.1 Precautions for Interrupts

### 1.1.4 INT Interrupt

1. Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to pins  $\overline{INT0}$  through  $\overline{INT5}$  regardless of the CPU operation clock.
2. When the polarity of the  $\overline{INT0}$  to  $\overline{INT5}$  pins is changed or the interrupt request cause of the software interrupt numbers 8 to 9 is changed, the IR bit is sometimes set to "1" (interrupt request). After these changes were made, set the interrupt request bit to "0" (no interrupt request). Figure 1.1.1 shows the procedure for changing the  $\overline{INT}$  interrupt generate factor.

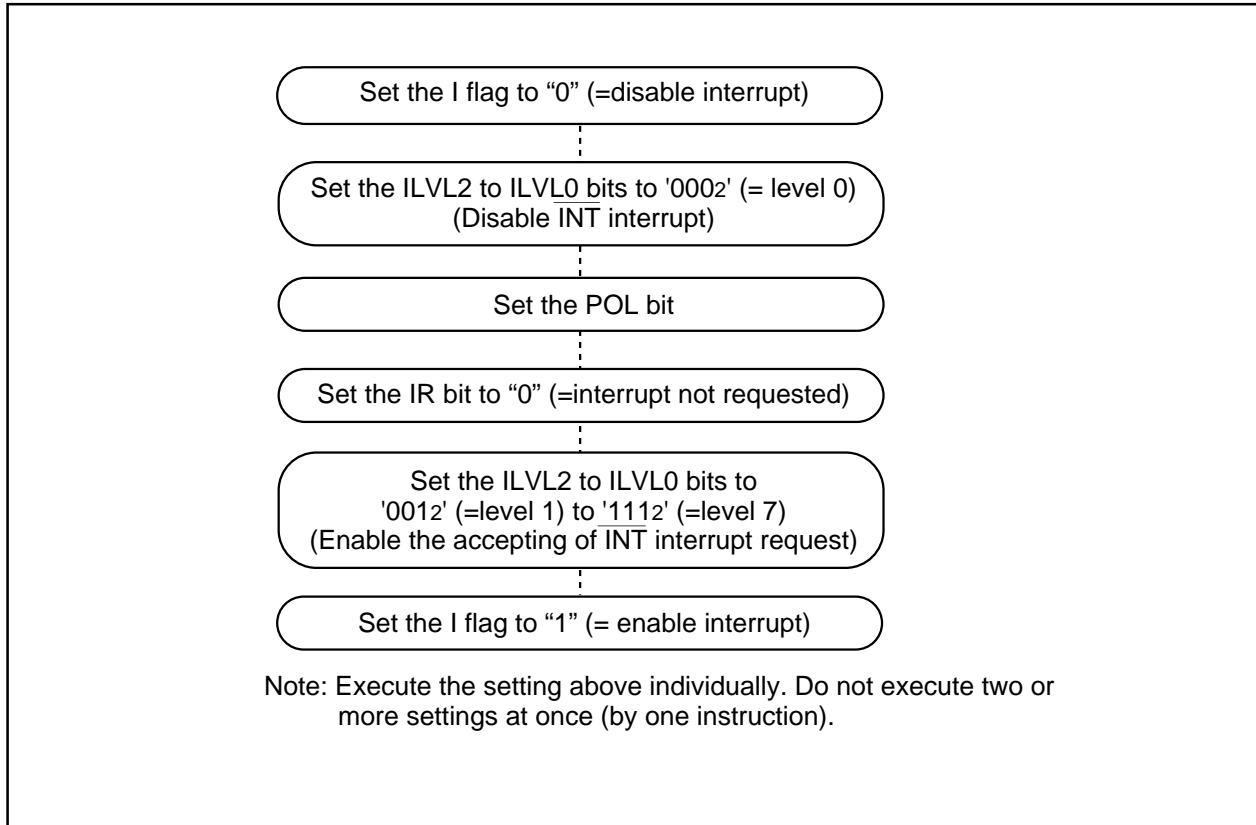


Figure 1.1.1. Procedure for Changing the  $\overline{INT}$  Interrupt Generate Factor

## 1.1 Precautions for Interrupts

### 1.1.5 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

## 1.1 Precautions for Interrupts

### 1.1.6 Rewrite the Interrupt Control Register

Each interrupt control register can only be modified while no interrupt requests corresponding to that register are generated. If interrupt requests managed by any interrupt control register are likely to occur, disable the interrupts before modifying the register. A sample program is shown below.

#### Example 1:

```
INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Set the TA0IC register to "0016".
  NOP                               ; Four NOP instructions are required when using HOLD function.
  NOP
  FSET  I           ; Enable interrupts.
```

#### Example 2:

```
INT_SWITCH2:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Set the TA0IC register to "0016".
  MOV.W MEM, R0     ; Dummy read.
  FSET  I           ; Enable interrupts.
```

#### Example 3:

```
INT_SWITCH3:
  PUSHC FLG        ; Push Flag register onto stack
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Set the TA0IC register to "0016".
  POPC  FLG        ; Enable interrupts.
```

Why the FSET I instruction is preceded by two NOP instructions (four when using HOLD function) in Example 1 and why the FSET I instruction is preceded by a dummy read in Example 2  
This is to prevent the I flag from being set to "1" before writing to the interrupt control register for reasons of the instruction queue buffer.

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

#### (1) Modifying Other Than the IR Bit

If an interrupt request corresponding to that register is generated while executing the instruction, the IR bit may not be set to "1" (= interrupt requested), with the result that the interrupt request is ignored. If this presents a problem, use the following instructions to modify the register.

Instructions to use: AND, OR, BCLR, BSET

#### (2) Modifying the IR Bit

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

## 1.2 Precautions for Protect

---

### 1.2 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.

## 1.3 Precautions for DMAC

---

### 1.3 Precautions for DMAC

#### 1.3.1 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.

Step 1: Write "1" to the DMAE bit and DMAS bit in DMiCON register simultaneously<sup>(\*1)</sup>.

Step 2: Make sure that the DMAi is in an initial state<sup>(\*2)</sup> 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.



## 1.4 Precautions for Timers

---

### 1.4 Precautions for Timers

#### 1.4.1 Timers A and B

This section describes precautions for timers A and B. Precautions for each mode should be referred as well.

1. After reset, timers stop. After setting mode, count source or counter value, the TAI<sub>S</sub> bit (i=0 to 4) or TB<sub>j</sub>S bit (j=0 to 5) in the TABSR or TBSR register should be set to "1" (starts counting). Make sure that the TAI<sub>S</sub> bit or TB<sub>j</sub>S bit is set to "0" (stops counting) before changing the registers and bits listed below.
  - TAI<sub>M</sub>R register and TB<sub>j</sub>M<sub>R</sub> register
  - TAI register and TB<sub>j</sub> register
  - UDF register
  - TAZIE, TA0TGL and TA0TGH bits in ONSF register
  - TRGSR register

## 1.4 Precautions for Timers

---

### 1.4.2 Timer A

#### 1.4.2.1 Timer A (Timer Mode)

1. After reset, the TABSR register TAI<sub>S</sub> bit (i = 0 to 4) is cleared to "0" (stops counting). Select operation mode and set a value in the TAI register before setting the TAI<sub>S</sub> bit to "1" (starts counting).
2. 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 "FFFF<sub>16</sub>" 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.
3. If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the TB2SC register IVPCR1 bit = "1" (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the TA1<sub>OUT</sub>, TA2<sub>OUT</sub> and TA4<sub>OUT</sub> pins go to a high-impedance state.

## 1.4 Precautions for Timers

---

### 1.4.2.2 Timer A (Event Counter Mode)

1. After reset, the TABSR register TAI<sub>S</sub> bit ( $i = 0$  to 4) is cleared to "0" (stopped counting). Select operation mode and set a value in the TAI register before setting the TAI<sub>S</sub> bit to "1" (start counting).
2. While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, "FFFF<sub>16</sub>" can be read in underflow, while reloading, and "0000<sub>16</sub>" 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.
3. If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the TB2SC register IVPCR1 bit = "1" (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the TA1<sub>OUT</sub>, TA2<sub>OUT</sub> and TA4<sub>OUT</sub> pins go to a high-impedance state.

## 1.4 Precautions for Timers

---

### 1.4.2.3 Timer A (One-shot Timer Mode)

1. After reset, the TABSR register TAI<sub>S</sub> bit ( $i = 0$  to 4) is cleared to "0" (stopped counting). Select operation mode and set a value in the TAI register before setting the TAI<sub>S</sub> bit to "1" (start counting).
2. When setting TABSR register to "0" (count stop), the followings occur:
  - A counter stops counting and a content of reload register is reloaded.
  - TAI<sub>OUT</sub> pin outputs "L".
  - After one cycle of the CPU clock, the IR bit of TAI<sub>IC</sub> register is set to "1" (interrupt request).
3. Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger has been selected, one-cycle delay of a count source as maximum occurs between a trigger input to TAI<sub>IN</sub> pin and output in one-shot timer mode.
4. 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 A<sub>i</sub> interrupt (the IR bit), set the IR bit to "0" after the changes listed above have been made.
5. 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.
6. If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the TB2SC register IVPCR1 bit = "1" (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the TA1<sub>OUT</sub>, TA2<sub>OUT</sub> and TA4<sub>OUT</sub> pins go to a high-impedance state.

## 1.4 Precautions for Timers

---

### 1.4.2.4 Timer A (Pulse Width Modulation Mode)

1. After reset, the TABSR register TAI<sub>S</sub> bit ( $i = 0$  to 4) is cleared to "0" (stopped counting). Select operation mode and set a value in the TAI register before setting the TAI<sub>S</sub> bit to "1" (start counting).
2. 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.
3. When setting TAI<sub>S</sub> register to "0" (count stop) during PWM pulse output, the following action occurs:
  - Stop counting.
  - When TAI<sub>OUT</sub> pin is output "H", output level is set to "L" and the IR bit is set to "1".
  - When TAI<sub>OUT</sub> pin is output "L", both output level and the IR bit remains unchanged.
4. If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the TB2SC register IVPCR1 bit = "1" (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the TA1<sub>OUT</sub>, TA2<sub>OUT</sub> and TA4<sub>OUT</sub> pins go to a high-impedance state.

## 1.4 Precautions for Timers

---

### 1.4.3 Timer B

#### 1.4.3.1 Timer B (Timer Mode and Event Counter Mode)

1. After reset, the TBiS bit ( $i = 0$  to  $5$ ) is cleared to "0" (stopped counting). Select operation mode and set a value in the TBi register before setting the TBiS bit to "1" (start counting).

The TB0S to TB2S bits are the bits 5 to 7 of TABSR register, the TB3S to TB5S bits are the bits 5 to 7 of TBSR register.

2. A value of a counter, while counting, can be read in TBi register at any time. "FFFF<sub>16</sub>" is read while reloading. Setting value is read between setting values in TBi register at count stop and starting a counter.

## 1.4 Precautions for Timers

---

### 1.4.3.2 Timer B (Pulse Period/pulse Width Measurement Mode)

1. The IR bit of TBiC register (i=0 to 5) goes to "1" (overflow), 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 of TBiMR register within the interrupt routine.
2. 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.
3. 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).
4. Use the IR bit of TBiC register to detect only overflows. Use the MR3 bit only to determine the interrupt factor within the interrupt routine.
5. 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.
6. 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.
7. When changing the MR1 to MR0 bits of TBiMR after a count is started, the IR bit of TBiC register may be set to "1" (interrupt request). Note that the IR bit does not change if the same value as before is written to the MR1 to MR0 bits.
8. 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.

## 1.5 Precautions for Serial I/O (Clock-synchronous Serial I/O)

---

### 1.5 Precautions for Serial I/O (Clock-synchronous Serial I/O)

#### 1.5.1 Transmission/reception

1. With an external clock selected, and choosing the  $\overline{\text{RTS}}$  function, the output level of the  $\overline{\text{RTSi}}$  pin goes to "L" when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the  $\overline{\text{RTSi}}$  pin goes to "H" when reception starts. So if the  $\overline{\text{RTSi}}$  pin is connected to the  $\overline{\text{CTS}}$  pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the  $\overline{\text{RTS}}$  function has no effect.
2. If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the TB2SC register IVPCR1 bit = "1" (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the  $\overline{\text{RTS2}}$  and CLK2 pins go to a high-impedance state.



## 1.5 Precautions for Serial I/O (Clock-synchronous Serial I/O)

---

### 1.5.2 Transmission

When an external clock is selected, the conditions must be met while if the UiC0 register's CKPOL bit = "0" (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the UiC0 register's CKPOL bit = "1" (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit of UiC1 register= "1" (transmission enabled)
- The TI bit of UiC1 register = "0" (data present in UiTB register)
- If CTS function is selected, input on the CTSi pin = "L"

## 1.5 Precautions for Serial I/O (Clock-synchronous Serial I/O)

---

### 1.5.3 Reception

1. 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.
2. When an internal clock is selected, set the UiC1 register (i = 0 to 2)'s TE bit 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 UiC1 register (i = 0 to 2)'s 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.
3. When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the UiC1 register (i = 0 to 2)'s RE bit = "1" (data present in the UiRB register), an overrun error occurs and the UiRB register OER bit 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 SiRIC register IR bit does not change state.
4. To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.
5. 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 of UiC1 register = "1" (reception enabled)
  - The TE bit of UiC1 register = "1" (transmission enabled)
  - The TI bit of UiC1 register = "0" (data present in the UiTB register)

## 1.6 Precautions for Serial I/O (UART Mode, Special Mode 2)

---

### 1.6 Precautions for Serial I/O (UART Mode, Special Mode 2)

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

## 1.7 Precautions for A-D Converter

### 1.7 Precautions for A-D Converter

1. Set ADCON0 (except bit 6), ADCON1 and ADCON2 registers when A-D conversion is stopped (before a trigger occurs).
2. When the VCUT bit of ADCON1 register is changed from "0" (Vref not connected) to "1" (Vref connected), start A-D conversion after passing 1  $\mu$ s or longer.
3. 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) each and the AVSS pin. Similarly, insert a capacitor between the VCC pin and the VSS pin. Figure 1.7.1 is an example connection of each pin.
4. Make sure the port direction bits for those pins that are used as analog inputs are set to "0" (input mode). Also, if the ADCON0 register's TGR bit = 1 (external trigger), make sure the port direction bit for the  $\overline{\text{ADTRG}}$  pin is set to "0" (input mode).
5. 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.)
6. The  $\phi_{\text{AD}}$  frequency must be 10 MHz or less. Without sample-and-hold function, limit the  $\phi_{\text{AD}}$  frequency to 250kHz or more. With the sample and hold function, limit the  $\phi_{\text{AD}}$  frequency to 1MHz or more.
7. When changing an A-D operation mode, select analog input pin again in the CH2 to CH0 bits of ADCON0 register and the SCAN1 to SCAN0 bits of ADCON1 register.

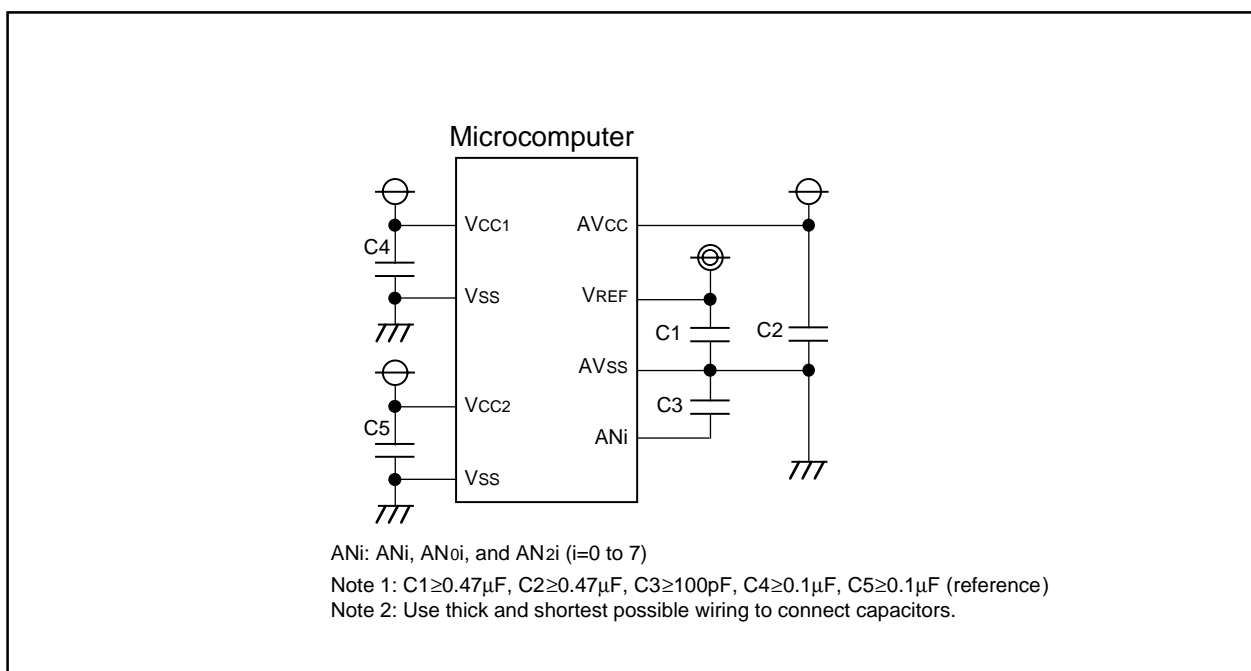


Figure 1.7.1. Use of capacitors to reduce noise

## 1.7 Precautions for A-D Converter

---

8. If the CPU reads the AD<sub>i</sub> register (i = 0 to 7) at the same time the conversion result is stored in the AD<sub>i</sub> register after completion of A-D conversion, an incorrect value may be stored in the AD<sub>i</sub> 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 or single-sweep mode

Check to see that A-D conversion is completed before reading the target AD<sub>i</sub> register. (Check the AD<sub>i</sub>C register's IR bit to see if A-D conversion is completed.)

- When operating in repeat mode or repeat sweep mode 0 or 1

Use the main clock for CPU clock directly without dividing it.

9. If A-D conversion is forcibly terminated while in progress by setting the ADCON0 register's ADST bit to "0" (A-D conversion halted), the conversion result of the A-D converter is indeterminate. The contents of AD<sub>i</sub> 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 AD<sub>i</sub> registers.

10. If  $V_{CC2} < V_{CC1}$ , do not use AN<sub>00</sub> to AN<sub>07</sub> and AN<sub>20</sub> to AN<sub>27</sub> as analog input pins.

## 1.8 Precautions for Power Control

---

### 1.8 Precautions for Power Control

1. When exiting stop mode by hardware reset, set  $\overline{\text{RESET}}$  pin to "L" until a main clock oscillation is stabilized.
2. Insert more than four NOP instructions after an WAIT instruction or a instruction to set the CM10 bit of CM1 register to "1". When shifting to wait mode or stop mode, an instruction queue reads ahead to the next instruction to halt a program by an WAIT instruction and an instruction to set the CM10 bit to "1" (all clocks stopped). The next instruction may be executed before entering wait mode or stop mode, depending on a combination of instruction and an execution timing.
3. Wait until the  $t_{su(M-L)}$  elapses or main clock oscillation stabilization time, whichever is longer, before switching the clock source for CPU clock to the main clock.  
Similarly, wait until the sub clock oscillates stably before switching the clock source for CPU clock to the sub clock.

#### 4. Suggestions to reduce power consumption

##### (a) 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.

##### (b) 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  $\mu\text{s}$  or longer after setting the VCUT bit to "1" (VREF connection).

##### (c) D-A converter

When not performing D-A conversion, set the DAi bit ( $i=0, 1$ ) of DACON register to "0" (input inhibited) and DAi register to "0016".

##### (d) 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. During low speed mode and low power dissipation mode, do not set the CM02 bit to "1" (peripheral function clock stopped when in wait mode) before entering wait mode.

##### (e) Switching the oscillation-driving capacity

Set the driving capacity to "LOW" when oscillation is stable.

##### (f) External clock

When using an external clock input for the CPU clock, set the CM0 register CM05 bit to "1" (stop). Setting the CM05 bit to "1" disables the XOUT pin from functioning, which helps to reduce the amount of current drawn in the chip. (When using an external clock input, note that the clock remains fed into the chip regardless of how the CM05 bit is set.)

## 1.9 Precautions for External Bus

---

### 1.9 Precautions for External Bus

1. The external ROM version can operate only in the microprocessor mode, connect the CNVss pin to Vcc.
2. When resetting CNVss pin with "H" input, contents of internal ROM cannot be read out.

## 1.10 Electric Characteristic Differences Between Mask ROM and Flash Memory Version Microcomputers

---

### 1.10 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 flash memory version.



## 1.11 Precautions for Flash Memory Version

---

### 1.11 Precautions for Flash Memory Version

#### 1.11.1 Precautions for Functions to Inhibit Rewriting Flash Memory Rewrite

ID codes are stored in addresses 0FFFDF<sub>16</sub>, 0FFFE3<sub>16</sub>, 0FFFE<sub>16</sub>, 0FFFEF<sub>16</sub>, 0FFFF3<sub>16</sub>, 0FFFF7<sub>16</sub>, and 0FFFFB<sub>16</sub>. If wrong data are written to these addresses, the flash memory cannot be read or written in standard serial I/O mode.

The ROMCP register is mapped in address 0FFFFFF<sub>16</sub>. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.

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

## 1.11 Precautions for Flash Memory Version

---

### 1.11.2 Precautions for Program Command

Write 'xx4016' 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.

## 1.11 Precautions for Flash Memory Version

---

### 1.11.3 Precautions for Lock Bit Program Command

Write 'xx7716' in the first bus cycle and write 'xxD016' 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 the address value specified in the first bus cycle is the same uppermost block address that is specified in the second bus cycle.

## 1.11 Precautions for Flash Memory Version

---

### 1.11.4 Precautions for Stop mode

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

- Set the FMR01 bit to "0" (CPU rewrite mode disabled) and disable DMA transfers before setting the CM10 bit to "1" (stop mode).
- Execute the JMP.B instruction subsequent to the instruction which sets the CM10 bit to "1" (stop mode)

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

L1:

Program after returning from stop mode

## 1.11 Precautions for Flash Memory Version

---

### 1.11.5 Precautions for Wait mode

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

## 1.11 Precautions for Flash Memory Version

---

### 1.11.6 Precautions for CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

#### 1.11.6.1 Operation speed

Before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or less for BCLK using the CM0 register's CM06 bit and CM1 register's CM17–6 bits. Also, set the PM1 register's PM17 bit to 1 (with wait state).

## 1.11 Precautions for Flash Memory Version

---

### 1.11.6.2 Instructions inhibited against use

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

## 1.11 Precautions for Flash Memory Version

---

### 1.11.6.3 Interrupts

#### EW0 Mode

- Any interrupt which has a vector in the variable vector table can be used providing that its vector is transferred into the RAM area.
- The  $\overline{\text{NMI}}$  and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.

Because the rewrite operation is halted when a  $\overline{\text{NMI}}$  or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

- The address match interrupt cannot be used because the flash memory's internal data is referenced.

#### EW1 Mode

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program or auto erase period.
- Avoid using watchdog timer interrupts.
- The  $\overline{\text{NMI}}$  interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.

Because the rewrite operation is halted when a  $\overline{\text{NMI}}$  interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.



## 1.11 Precautions for Flash Memory Version

---

### 1.11.6.4 How to access

To set the FMR01, FMR02, or FMR11 bit to "1", write "0" and then "1" in succession. This is necessary to ensure that no interrupts or DMA transfers will occur before writing "1" after writing "0". Also only when  $\overline{\text{NMI}}$  pin is "H" level.

## 1.11 Precautions for Flash Memory Version

---

### 1.11.6.5 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.

## 1.11 Precautions for Flash Memory Version

---

### 1.11.6.6 DMA transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR0 register's FMR00 bit = 0 (during the auto program or auto erase period).

## 1.11 Precautions for Flash Memory Version

---

### 1.11.6.7 Writing command and data

Write the command code and data at even addresses.

## 1.11 Precautions for Flash Memory Version

---

### 1.11.7 Precautions for Low power dissipation mode, ring oscillator low power dissipation mode

If the CM05 bit is set to "1" (main clock stop), the following commands must not be executed.

- Program
- Block erase
- Erase all unlocked blocks
- Lock bit program

## 1.12 Precautions for PLL Frequency Synthesizer

---

### 1.12 Precautions for PLL Frequency Synthesizer

Make the supply voltage stable to use the PLL frequency synthesizer.

For ripple with the supply voltage 5V, keep below 10kHz as frequency, below 0.5V (peak to peak) as voltage fluctuation band and below 1V/mS as voltage fluctuation rate.

For ripple with the supply voltage 3V, keep below 10kHz as frequency, below 0.3V (peak to peak) as voltage fluctuation band and below 0.6V/mS as voltage fluctuation rate.

## 1.13 Precautions for Programmable I/O Ports

---

### 1.13 Precautions for Programmable I/O Ports

1. If a low-level signal is applied to the  $\overline{\text{NMI}}$  pin when the TB2SC register IVPCR1 bit = "1" (three-phase output forcible cutoff by input on  $\overline{\text{NMI}}$  pin enabled), the P72 to P75, P80 and P81 pins go to a high-impedance state.

## 2.1 Vdet2 Detection

---

## 2. Differences Made Depending on Manufactured Time

### 2.1 Vdet2 Detection

The present version of the products may not detect the Vdet2 voltage in the voltage detection circuit properly. Therefore, the followings should be noted.

- (1) When the VC25 bit in the VCR2 register is set to "1" (enabling the RAM retention limit detection circuit), the present version may not be reset even if the voltage at the Vcc1 input pin drops below Vdet2.
- (2) The WD5 bit in the WDC register may not change properly.

### Supplementary Explanation

Normally, during the stop mode, the Vdet3 voltage is not detected, and thus no reset is generated even when the input voltage at the Vcc1 pin drops to Vdet3 or less. Therefore, if the microcomputer is not reset when the VCC1 voltage drops below Vdet2 due to the reason described in the above No.1, the microcomputer cannot get out of the stop mode with Hardware Reset 2.



## 2.2 RESET Input

---

### 2.2 RESET Input

Ensure that pin RESET must hold valid-low state during powering-up.

When using a reset IC, use a CMOS type IC. When using an open-drain type reset IC, insert a capacitor between the reset input and Vss and a resistor between the input and Vcc respectively. The R-C time constant of the capacitor and resistor must provide a low state at least 10 times longer than the Vcc rise time.

REVISION HISTORY

M16C/62P GROUP USAGE NOTES

Rev.	Date	Description	
		Page	Summary
1.0	Jan/31/Y03	1 8 9 15 18 19 22 25 26 38	Figure 1.1.1 is partly revised. The section "1.3 Precautions for DMAC" is added. The section "1.4.1 Timers A and B" is added. The section "1.4.3.2 Timer B (Pulse Period/Pulse Width Measurement Mode)" is partly revised. The section "1.5.3 Reception" is partly revised. The section "1.6 Precautions for Serial I/O (UART Mode, Special Mode 2)" is partly revised. The section "1.8 Precautions for Power Control" is partly revised. The section "1.11.1 Precautions for Functions to Inhibit Rewriting Flash Memory Rewrite" is partly revised. The section "1.11.2 Precautions for Program Command" is partly revised. The section "1.12 Precautions for PLL Frequency Synthesizer" is partly revised.

**MITSUBISHI SEMICONDUCTORS  
USAGE NOTES REFERENCE BOOK  
M16C/62 (M16C/62P) Group**

---

February First Edition 2003

Editioned by  
Committee of editing of Mitsubishi Semiconductor Usage Notes Reference  
Book

Published by  
Mitsubishi Electric Corp., Semiconductor Marketing Division

---

This book, or parts thereof, may not be reproduced in any form without permission  
of Mitsubishi Electric Corporation.

**©2003 MITSUBISHI ELECTRIC CORPORATION**

# Usage Notes Reference Book

## M16C/62 (M16C/62P) Group

 **mitsubishi electric corporation**  
HEAD OFFICE: 2-2-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN