

DESCRIPTION

The M306V8FJFP are single-chip microcomputers using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core and are packaged in a 116-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. They also feature a built-in OSD display function and data slicer, making them ideal for closed caption and ID1 for TV control.

Applications

TV

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

DESCRIPTION	1	A/D Converter	156
Central Processing Unit (CPU)	9	Multi-master I ² C-BUS Interface	170
Reset	22	Data Slicer	190
Processor Mode	28	HSYNC Counter	203
Clock Generating Circuit	50	OSD Functions	204
Protection	67	Programmable I/O Ports	260
Interrupts	68	ELECTRICAL CHARACTERISTICS	295
Watchdog Timer	89	Flash Memory Version	316
DMAC	91	Usage Precaution	348
Timer	101	PACKAGE OUTLINE	363
Serial I/O	123		

Performance Outline

Table 1.1. Performance outline of M306V8FJFP

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		62.5 ns (f(BCLK)= 16MHz)
Memory capacity	ROM	(See the product list)
	RAM	(See the product list)
I/O port	P0 to P10	75
Multifunction timer	TA0, TA1, TA2, TA3, TA4	16 bits output x 5 channels
	TB0, TB1, TB2, TB3, TB4, TB5	16 bits input x 6 channels
Serial I/O	UART0, UART1, UART2	(UART, clock sync. serial I/O, IEBus (Note 2)) x 3
A/D converter		8 bits x 13 channels
Data slicer		2 circuits
Hsync counter		1 circuit 2 lines
OSD function		1 circuit
Multi-master I ² Cbus interface (Note 1)		3 circuits 4 lines
DMAC		2 channels (trigger: 29 sources)
Watchdog timer		15 bits x 1 (with prescaler)
Interrupt		31 internal and 5 external sources, 4 software sources, 7 levels
Clock generation circuit		3 circuits <ul style="list-style-type: none"> • Main clock • Sub-clock • OSD clock } (These circuits contain a built-in feedback resistor and external ceramic/quartz oscillator)
Power supply voltage		3.15 to 3.45V
Flash memory	Program/erase voltage	3.15 to 3.45V
	Number of program/erase	100 times
Power consumption		500mW
I/O characteristics	I/O withstand voltage	3.3V
	Output current	5mA
Memory expansion		Available (to 4M bytes)
Operating ambient temperature		-20 to 70°C
Device configuration		CMOS high performance silicon gate
Package		116-pin plastic mold QFP

Notes:

1. I²C bus is a registered trademark of Koninklijke Philips Electronics N. V.
2. IEBus is a trademark of NEC Electronics Corporation.
When you use option function, please specify that.

Block Diagram

Figure 1.1 is a block diagram of the M306V8FJFP.

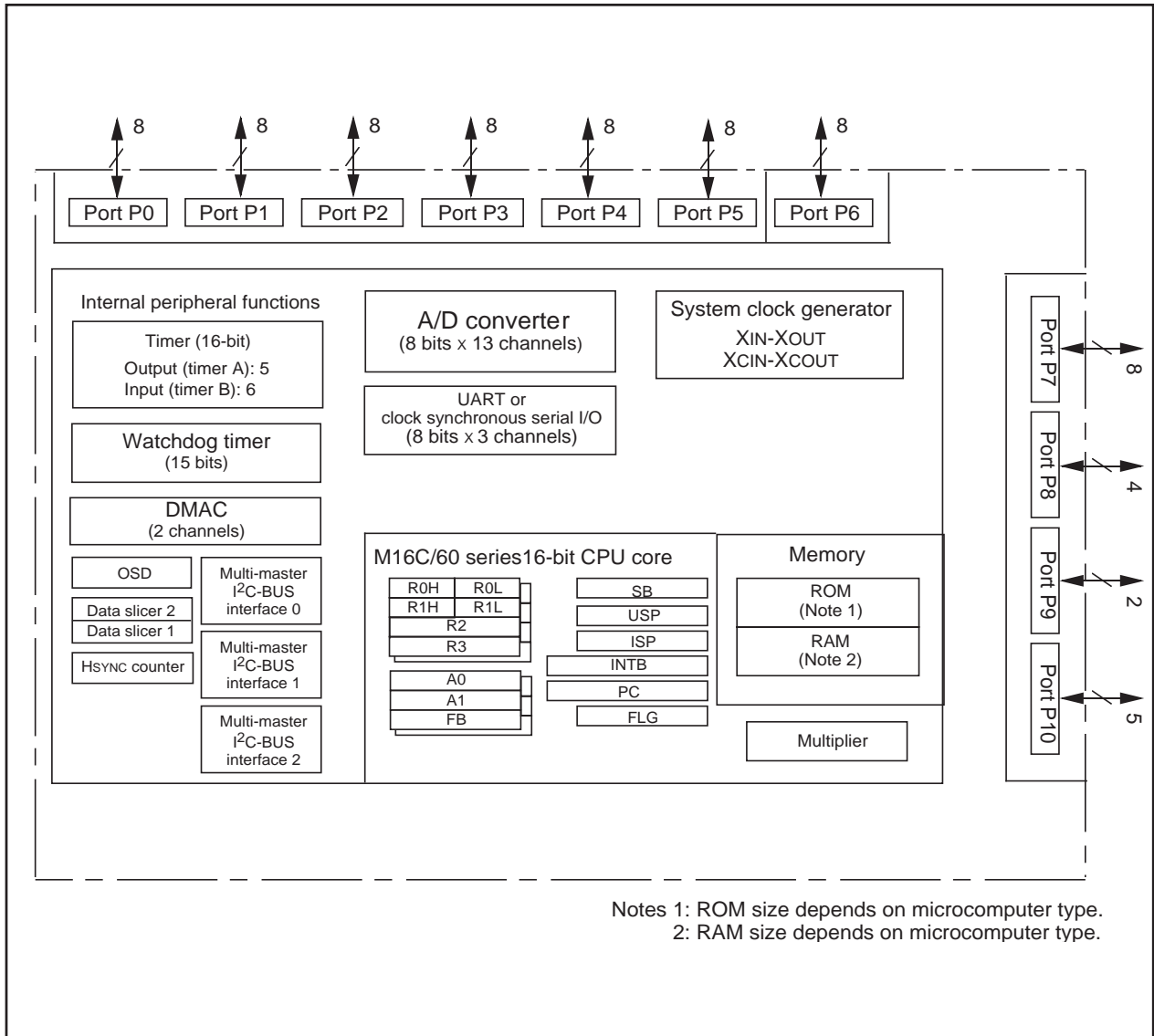


Figure 1.1. Block Diagram

Product List

Product list is show in Table 1.2 type No., memory size and package type are show in Figure 1.2.

Table 1.2. Product List

Type No.	ROM capacity	RAM capacity	Package type	Remarks
M306V8FJFP	512K bytes	16K bytes	116P6A-A	Flash memory version

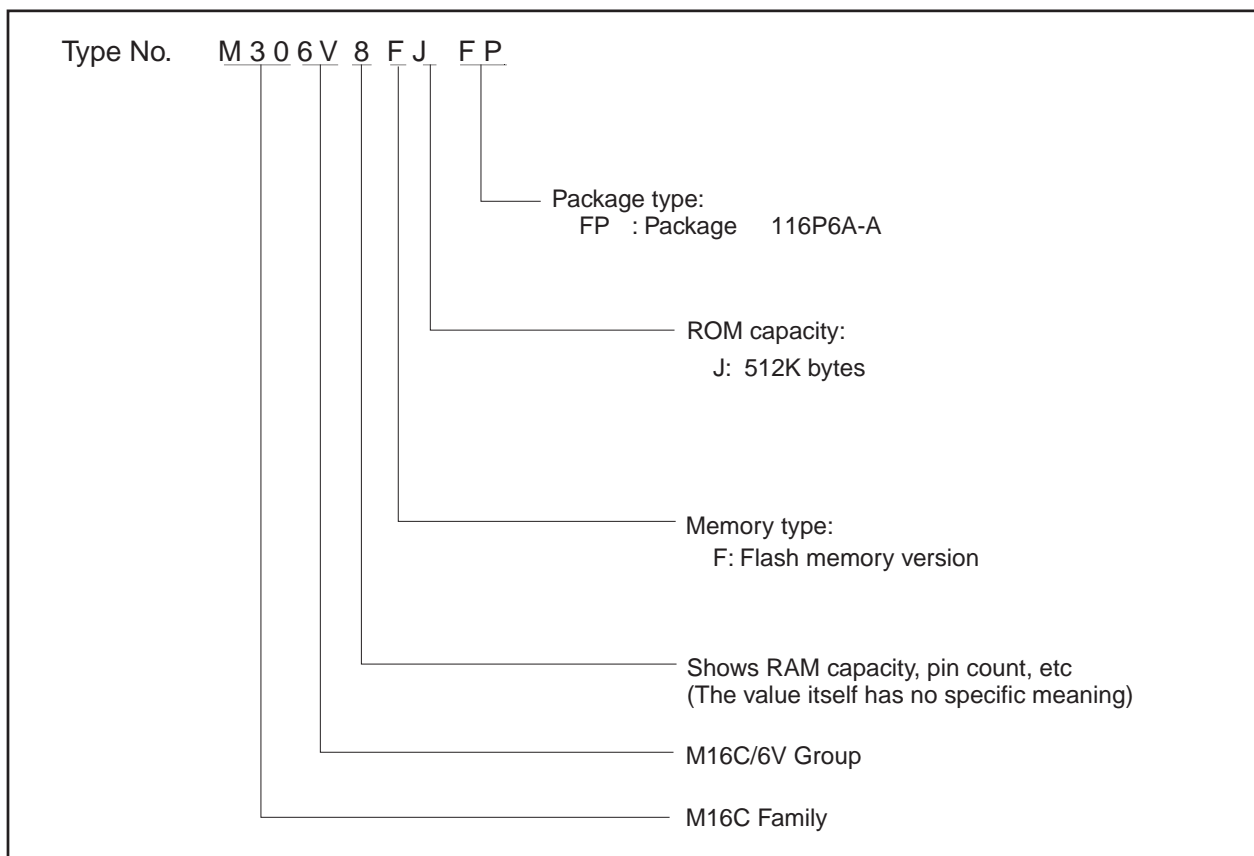


Figure 1.2. Type No., Memory Size, and Package

Pin Configuration

Figures 1.3 show the pin configuration.

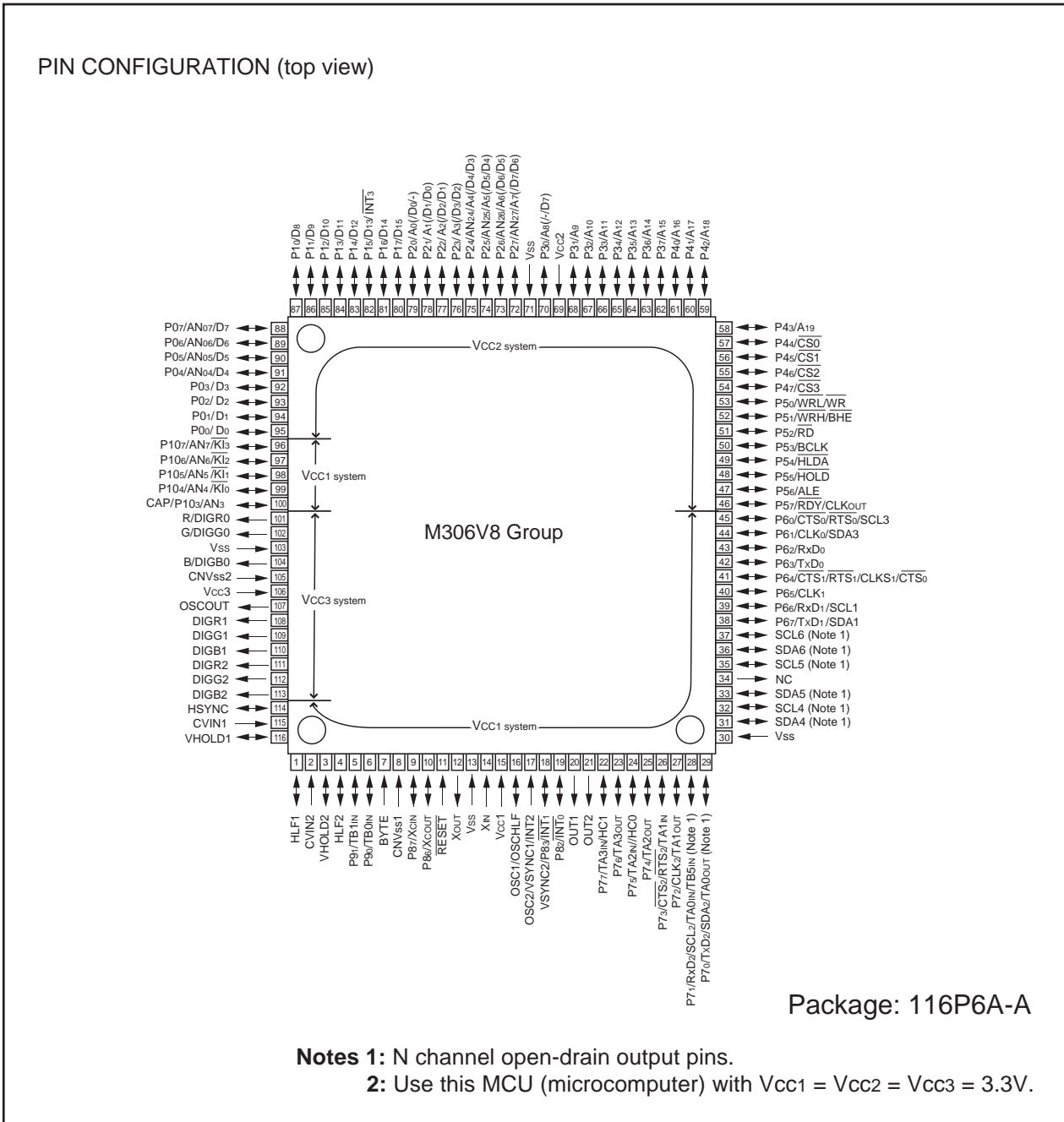


Figure 1.3. Pin Configuration (Top View)

Pin Description

Table 1.3. Pin Description (1)

Pin name	Signal name	I/O type	Power supply	Function
VCC1, VCC2, VCC3, VSS	Power supply input		—	Apply 3.3 V to the VCC1, VCC2 and VCC3 pins and 0 V to the VSSpin. (Note 1) Insert a bypass capacitor between power supply and GND. (Note 2)
CNVSS1, CNVSS2	CNVSS1/ CNVSS2	Input	VCC	CNVSS1 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. Always connect CNVSS2 pin to VSS.
RESET	Reset input	Input	VCC	"L" on this input resets the microcomputer.
XIN XOUT	Clock input Clock output	Input Output	VCC	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	VCC	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.
P00 to P07	I/O port P0	I/O	VCC	This is an 8-bit CMOS I/O port. This port has an I/O 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		I/O		When set as a separate bus, these pins input and output data (D0–D7).
P10 to P17	I/O port P1	I/O	VCC	This is an 8-bit I/O port equivalent to P0. P15 also function as $\overline{\text{INT}}$ interrupt input pins as selected by a program.
D8 to D15		I/O		When set as a separate bus, these pins input and output data (D8–D15).
P20 to P27	I/O port P2	I/O	VCC	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		I/O		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 I/O		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	I/O	VCC	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		I/O 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	I/O	VCC	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.

Notes 1: In this manual, hereafter, VCC refers to VCC1 unless otherwise noted.

2: Insert capacitors between each power supply pin and GND to prevent errors or latch-up by noise.
Also, use thick and shortest possible wiring to connect capacitors.

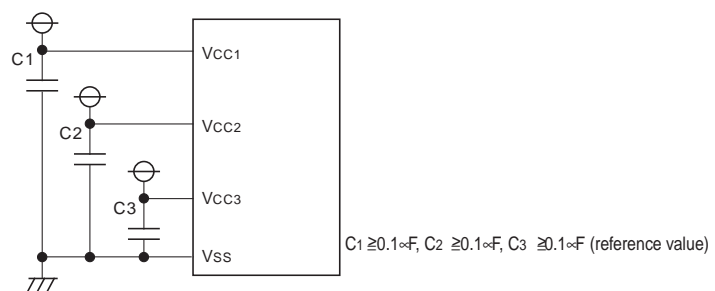


Table 1.4. Pin Description (2)

Pin name	Signal name	I/O type	Power supply	unction
P50 to P57	I/O port P5	I/O	Vcc	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.
\overline{WRL} / \overline{WR} , \overline{WRH} / \overline{BHE} , RD, BCLK, HLDA, HOLD, ALE, RDY		Output Output Output Output Input Output Input		Output $\overline{WRL}/\overline{WRH}$, ($\overline{BHE}/\overline{WR}$), RD, BCLK, HLDA, and ALE signals. $\overline{WRL}/\overline{WRH}$ and $\overline{BHE}/\overline{WR}$ are switch able in a program. Note that \overline{WRL} and \overline{WRH} are always used as a pair, so as \overline{WR} and \overline{BHE} . <ul style="list-style-type: none"> ■ \overline{WRL}, \overline{WRH}, and RD selected If the external data bus is 16 bits wide, data are written to even addresses when the \overline{WRL} signal is low, and written to odd addresses when the \overline{WRH} signal is low. Data are read out when the RD signal is low. ■ \overline{WR}, \overline{BHE}, and RD selected Data are written when the \overline{WR} signal is low, or read out when the RD signal is low. Odd addresses are accessed when the \overline{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 \overline{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	I/O	Vcc	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as UART0, UART1 and multi-master I ² C bus I/O pins as selected by program.
P70 to P77	I/O port P7	I/O	Vcc	This is an 8-bit I/O port equivalent to P0. (However, P70 and P71 are the pins of N-channel opendrain output) This port can function as I/O pins for timers A0 to A3 and B5 by selecting in a program. And, UART2, I ² C bus I/O pin, P75 and P77 can also function as input pin for Hsync center.
P82, P84, P86, P87,	I/O port P8	I/O	Vcc	They are I/O ports with the same functions as P0. When so selected in a program, they can function as I/O pins for INT interrupt, Vsync input pins and the sub clock oscillator circuit. In that case, connect a crystal resonator between P86 (XCOUT pin) and P87 (XCIN pin).
P90, P91	I/O port P9	I/O	Vcc	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as timer B0 and B1 input pins as selected by program.
P103 to P107	I/O port P10	I/O	Vcc	This is an I/O port equivalent to P0. Pins in this port also function as A/D converter input pins and the capacitor connection pin for analog RGB operation.
SCL4 to 6, SDA4 to 6	Multi-master I ² C-bus interface	I/O		These are exclusive pins for multi-master I ² C-bus interface (N-channel open drain output.)
Hsync	Hsync input	Input	Vcc	OSD function Hsync input pins.
R/DIGR0, B/DIGB0, G/DIGG0, OUT1, OUT2, DIGR1, DIGB1, DIGG1, DIGR2, DIGG2, DIGB2, OSCOU	OSD function output pin	Output	Vcc	These are exclusive pins for OSD functions.
CVin1, VHOLD1, HLF1, CVin2, VHOLD2, HLF2	Data slicer function I/O pin	I/O	Vcc	These are exclusive pins for data slicer function.
OSC1/ OSDHLF, OSC2	Oscillation pin for OSD function	I/O	Vcc	These are oscillation pins for OSD function. Using the same pins as external interrupt and Vsync input pin.

Memory

Figure 1.4 is a memory map of the M306V8FJFP. The address space extends the 1M bytes from address 00000₁₆ to FFFFF₁₆.

The internal ROM is allocated in a lower address direction beginning with address FFFFF₁₆. For example, a 64 Kbytes internal ROM is allocated to the addresses from F0000₁₆ to FFFFF₁₆.

The fixed interrupt vector table is allocated to the addresses from FFFDC₁₆ to FFFFF₁₆. Therefore, store the start address of each interrupt routine here.

The internal RAM is allocated in an upper address direction beginning with address 00400₁₆. For example, a 10 Kbytes internal RAM is allocated to the addresses from 00400₁₆ to 02BFF₁₆. 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 00000₁₆ to 003FF₁₆. 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 FFE00₁₆ to FFFDB₁₆. 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.

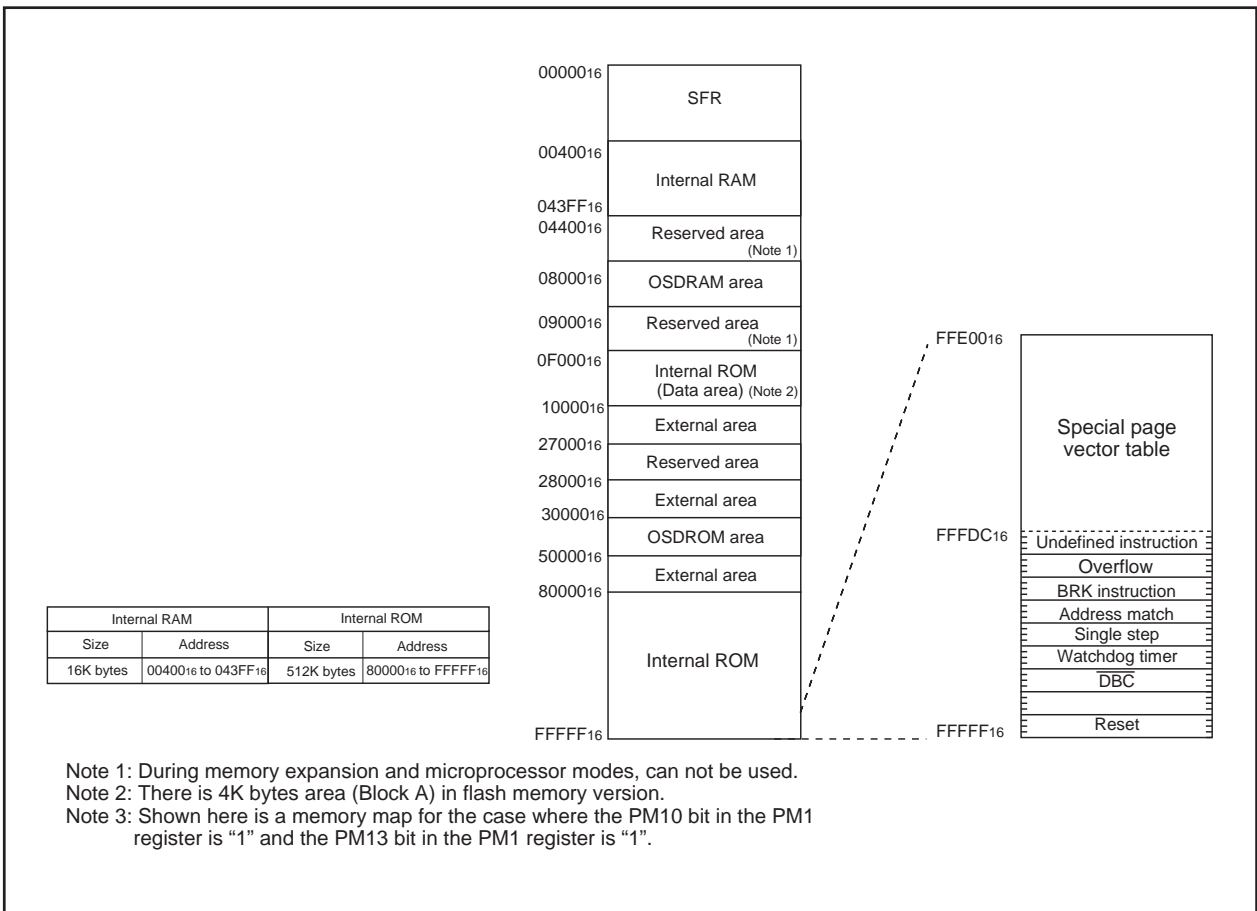


Figure 1.4. Memory Map

Central Processing Unit (CPU)

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

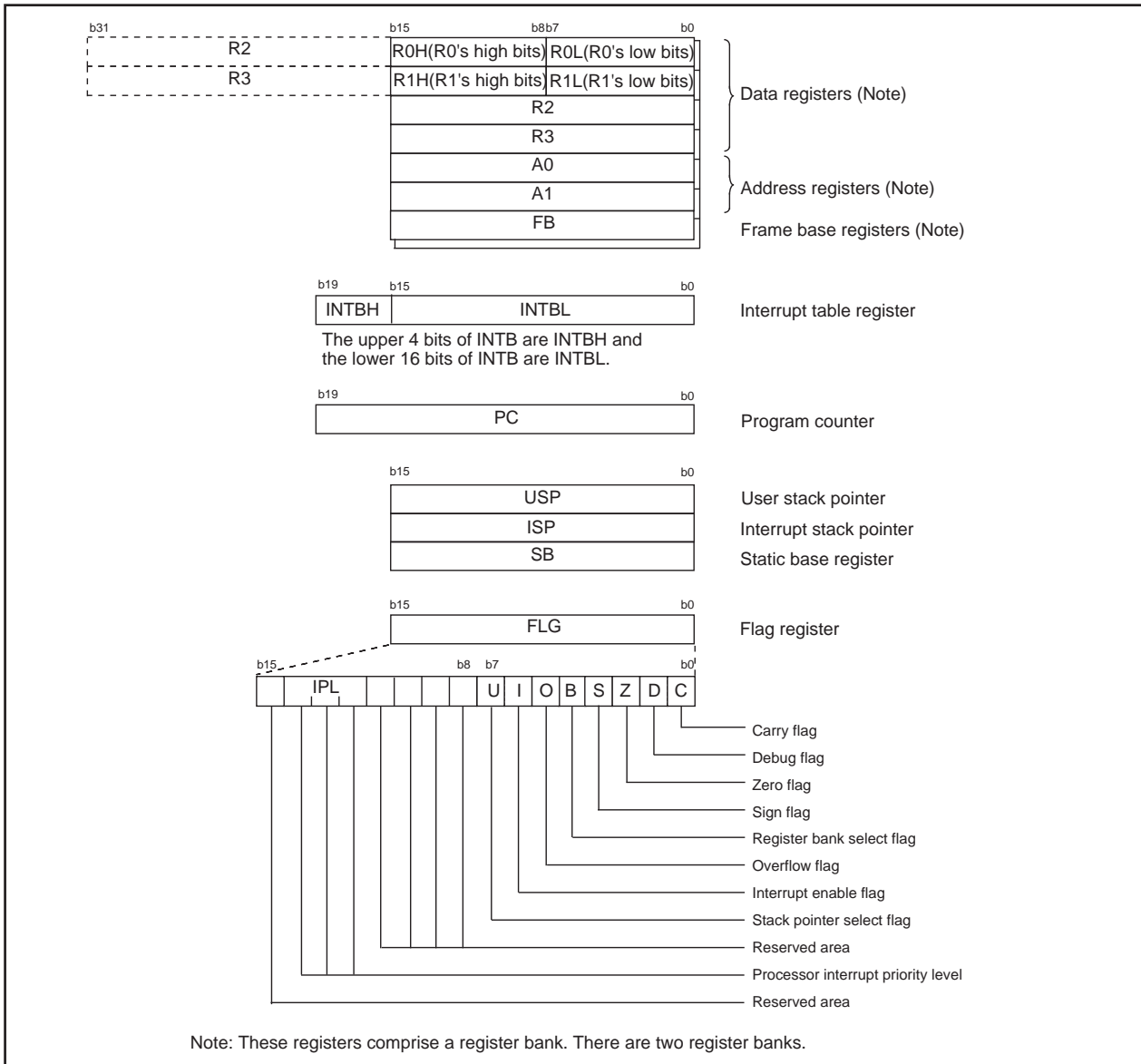


Figure 2.1. CPU registers

(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).

(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 ₁₆			
0001 ₁₆			
0002 ₁₆			
0003 ₁₆			
0004 ₁₆	Processor mode register 0 (Note 2)	PM0	00000002(CNVss1 pin is "L") 00000011z(CNVss1 pin is "H")
0005 ₁₆	Processor mode register 1	PM1	000010002
0006 ₁₆	System clock control register 0	CM0	010010002
0007 ₁₆	System clock control register 1	CM1	001000002
0008 ₁₆	Chip select control register	CSR	000000012
0009 ₁₆	Address match interrupt enable register	AIER	XXXXXX002
000A ₁₆	Protect register	PRCR	XX0000002
000B ₁₆	Data bank register	DBR	0016
000C ₁₆	System clock control register 2	CM2	0000X0002
000D ₁₆			
000E ₁₆	Watchdog timer start register	WDTS	??16
000F ₁₆	Watchdog timer control register	WDC	00??????2
0010 ₁₆	Address match interrupt register 0	RMAD0	0016
0011 ₁₆			0016
0012 ₁₆			X016
0013 ₁₆			
0014 ₁₆	Address match interrupt register 1	RMAD1	0016
0015 ₁₆			0016
0016 ₁₆			X016
0017 ₁₆			
0018 ₁₆			
0019 ₁₆	Reserved register	RSVREG0019	000010002
001A ₁₆	Reserved register	RSVREG001A	0016
001B ₁₆	Chip select expansion control register 2	CSE	0016
001C ₁₆	Reserved register	RSVREG001C	0001X0102
001D ₁₆			
001E ₁₆	Reserved register	RSVREG001E	XXX000002
001F ₁₆	Reserved register	RSVREG001F	0016
0020 ₁₆	DMA0 source pointer	SAR0	??16
0021 ₁₆			??16
0022 ₁₆			X?16
0023 ₁₆			
0024 ₁₆	DMA0 destination pointer	DAR0	??16
0025 ₁₆			??16
0026 ₁₆			X?16
0027 ₁₆			
0028 ₁₆	DMA0 transfer counter	TCR0	??16
0029 ₁₆			??16
002A ₁₆			
002B ₁₆			
002C ₁₆	DMA0 control register	DM0CON	00000?002
002D ₁₆			
002E ₁₆			
002F ₁₆			
0030 ₁₆	DMA1 source pointer	SAR1	??16
0031 ₁₆			??16
0032 ₁₆			X?16
0033 ₁₆			
0034 ₁₆	DMA1 destination pointer	DAR1	??16
0035 ₁₆			??16
0036 ₁₆			X?16
0037 ₁₆			
0038 ₁₆	DMA1 transfer counter	TCR1	??16
0039 ₁₆			??16
003A ₁₆			
003B ₁₆			
003C ₁₆	DMA1 control register	DM1CON	00000?002
003D ₁₆			
003E ₁₆			
003F ₁₆			

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

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

X : Nothing is mapped to this bit

? : This bit is indeterminate.

Address	Register	Symbol	After reset
0040 ₁₆			
0041 ₁₆			
0042 ₁₆			
0043 ₁₆			
0044 ₁₆	INT3 interrupt control register	INT3IC	XX00?0002
0045 ₁₆	Timer B5 interrupt control register	TB5IC	XXXX?0002
0046 ₁₆	Timer B4 interrupt control register	TB4IC	XXXX?0002
0047 ₁₆	Timer B3 interrupt control register	TB3IC	XXXX?0002
0048 ₁₆	Slicer 1 interrupt control register	DSC1IC	XX00?0002
0049 ₁₆	Slicer 2 interrupt control register	DSC2IC	XX00?0002
004A ₁₆	Bus collision detection interrupt control register	BCNIC	XXXX?0002
004B ₁₆	DMA0 interrupt control register	DM0IC	XXXX?0002
004C ₁₆	DMA1 interrupt control register	DM1IC	XXXX?0002
004D ₁₆	Key input interrupt control register	KUPIC	XXXX?0002
004E ₁₆	A/D conversion interrupt control register	ADIC	XXXX?0002
004F ₁₆	UART2 transmit interrupt control register	S2TIC	XXXX?0002
0050 ₁₆	UART2 receive interrupt control register	S2RIC	XXXX?0002
0051 ₁₆	UART0 transmit interrupt control register	S0TIC	XXXX?0002
0052 ₁₆	UART0 receive interrupt control register	S0RIC	XXXX?0002
0053 ₁₆	UART1 transmit interrupt control register	S1TIC	XXXX?0002
0054 ₁₆	UART1 receive interrupt control register	S1RIC	XXXX?0002
0055 ₁₆	Timer A0 interrupt control register	TA0IC	XXXX?0002
0056 ₁₆	Timer A1 interrupt control register	TA1IC	XXXX?0002
0057 ₁₆	Timer A2 interrupt control register	TA2IC	XXXX?0002
0058 ₁₆	Timer A3 interrupt control register	TA3IC	XXXX?0002
0059 ₁₆	Timer A4 interrupt control register	TA4IC	XXXX?0002
005A ₁₆	Timer B0 interrupt control register	TB0IC	XXXX?0002
005B ₁₆	Timer B1 interrupt control register	TB1IC	XXXX?0002
005C ₁₆	Timer B2 interrupt control register	TB2IC	XXXX?0002
005D ₁₆	INT0 interrupt control register	INT0IC	XX00?0002
005E ₁₆	INT1 interrupt control register	INT1IC	XX00?0002
005F ₁₆	INT2 interrupt control register	INT2IC	XX00?0002
0060 ₁₆			
0061 ₁₆			
0062 ₁₆			
0063 ₁₆			
0064 ₁₆			
0065 ₁₆			
0066 ₁₆			
0067 ₁₆			
0068 ₁₆			
0069 ₁₆			
006A ₁₆			
006B ₁₆			
006C ₁₆			
006D ₁₆			
006E ₁₆			
006F ₁₆			
0070 ₁₆			
0071 ₁₆			
0072 ₁₆			
0073 ₁₆			
0074 ₁₆			
0075 ₁₆			
0076 ₁₆			
0077 ₁₆			
0078 ₁₆			
0079 ₁₆			
007A ₁₆			
007B ₁₆			
007C ₁₆			
007D ₁₆			
007E ₁₆			
007F ₁₆			

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

X : Nothing is mapped to this bit ? : This bit is indeterminate.

Address	Register	Symbol	After reset
0080 ₁₆			
0081 ₁₆			
0082 ₁₆			
0083 ₁₆			
0084 ₁₆			
0085 ₁₆			
0086 ₁₆			
~			~
01B0 ₁₆			
01B1 ₁₆			
01B2 ₁₆			
01B3 ₁₆			
01B4 ₁₆	Flash identification register (Note 2)	FIDR	XXXXXX002
01B5 ₁₆	Flash memory control register 1 (Note 2)	FMR1	0?00??0?2
01B6 ₁₆			
01B7 ₁₆	Flash memory control register 0 (Note 2)	FMR0	??0000012
01B8 ₁₆	Address match interrupt register 2	RMAD2	0016
01B9 ₁₆			0016
01BA ₁₆			X016
01BB ₁₆	Address match interrupt enable register 2	AIER2	XXXXXX002
01BC ₁₆	Address match interrupt register 3	RMAD3	0016
01BD ₁₆			0016
01BE ₁₆			X016
01BF ₁₆			
01C0 ₁₆			
01C1 ₁₆			
01C2 ₁₆			
~			~
01E0 ₁₆			
01E1 ₁₆			
01E2 ₁₆			
01E3 ₁₆			
01E4 ₁₆			
01E5 ₁₆			
01E6 ₁₆			
01E7 ₁₆			
01E8 ₁₆			
01E9 ₁₆			
01EA ₁₆			
01EB ₁₆			
01EC ₁₆			
01ED ₁₆			
01EE ₁₆			
01EF ₁₆			
01F0 ₁₆			
01F1 ₁₆			
01F2 ₁₆			
01F3 ₁₆			
01F4 ₁₆			
01F5 ₁₆			
01F6 ₁₆			
01F7 ₁₆			
01F8 ₁₆			
01F9 ₁₆			
01FA ₁₆			
01FB ₁₆			
01FC ₁₆			
01FD ₁₆			
01FE ₁₆			
01FF ₁₆			

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

Note 2: This register is included in the flash memory version.

X : Nothing is mapped to this bit ? : This bit is indeterminate.

Address	Register	Symbol	After reset
0200 ₁₆			
0201 ₁₆	Sprite OSD control register	SC	XXX000002
0202 ₁₆	OSD control register 1	OC1	0016
0203 ₁₆	OSD control register 2	OC2	0016
0204 ₁₆	Horizontal position register	HP	0016
0205 ₁₆	Clock control register 1	CS	0016
0206 ₁₆	I/O polarity control register	PC	100000002
0207 ₁₆	OSD control register 3	OC3	0016
0208 ₁₆	Raster color register	RSC	0016
0209 ₁₆			0016
020A ₁₆	OSD reserved register 5	OR5	0016
020B ₁₆	Clock control register 2	CG	0016
020C ₁₆ 020D ₁₆	Top border control register	TBR	??16
020E ₁₆ 020F ₁₆	Bottom border control register	BBR	??16
0210 ₁₆	Block control register 1	BC1	??16
0211 ₁₆	Block control register 2	BC2	??16
0212 ₁₆	Block control register 3	BC3	??16
0213 ₁₆	Block control register 4	BC4	??16
0214 ₁₆	Block control register 5	BC5	??16
0215 ₁₆	Block control register 6	BC6	??16
0216 ₁₆	Block control register 7	BC7	??16
0217 ₁₆	Block control register 8	BC8	??16
0218 ₁₆	Block control register 9	BC9	??16
0219 ₁₆	Block control register 10	BC10	??16
021A ₁₆	Block control register 11	BC11	??16
021B ₁₆	Block control register 12	BC12	??16
021C ₁₆	Block control register 13	BC13	??16
021D ₁₆	Block control register 14	BC14	??16
021E ₁₆	Block control register 15	BC15	??16
021F ₁₆	Block control register 16	BC16	??16
0220 ₁₆ 0221 ₁₆	Vertical position register 1	VP1	??16 ??16
0222 ₁₆ 0223 ₁₆	Vertical position register 2	VP2	??16 ??16
0224 ₁₆ 0225 ₁₆	Vertical position register 3	VP3	??16 ??16
0226 ₁₆ 0227 ₁₆	Vertical position register 4	VP4	??16 ??16
0228 ₁₆ 0229 ₁₆	Vertical position register 5	VP5	??16 ??16
022A ₁₆ 022B ₁₆	Vertical position register 6	VP6	??16 ??16
022C ₁₆ 022D ₁₆	Vertical position register 7	VP7	??16 ??16
022E ₁₆ 022F ₁₆	Vertical position register 8	VP8	??16 ??16
0230 ₁₆ 0231 ₁₆	Vertical position register 9	VP9	??16 ??16
0232 ₁₆ 0233 ₁₆	Vertical position register 10	VP10	??16 ??16
0234 ₁₆ 0235 ₁₆	Vertical position register 11	VP11	??16 ??16
0236 ₁₆ 0237 ₁₆	Vertical position register 12	VP12	??16 ??16
0238 ₁₆ 0239 ₁₆	Vertical position register 13	VP13	??16 ??16
023A ₁₆ 023B ₁₆	Vertical position register 14	VP14	??16 ??16
023C ₁₆ 023D ₁₆	Vertical position register 15	VP15	??16 ??16
023E ₁₆ 023F ₁₆	Vertical position register 16	VP16	??16 ??16

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

X : Nothing is mapped to this bit ? : This bit is indeterminate.

Address	Register	Symbol	After reset
0240 ₁₆ 0241 ₁₆	Color palette register 1	CR1	?? ₁₆ ?? ₁₆
0242 ₁₆ 0243 ₁₆	Color palette register 2	CR2	?? ₁₆ ?? ₁₆
0244 ₁₆ 0245 ₁₆	Color palette register 3	CR3	?? ₁₆ ?? ₁₆
0246 ₁₆ 0247 ₁₆	Color palette register 4	CR4	?? ₁₆ ?? ₁₆
0248 ₁₆ 0249 ₁₆	Color palette register 5	CR5	?? ₁₆ ?? ₁₆
024A ₁₆ 024B ₁₆	Color palette register 6	CR6	?? ₁₆ ?? ₁₆
024C ₁₆ 024D ₁₆	Color palette register 7	CR7	?? ₁₆ ?? ₁₆
024E ₁₆ 024F ₁₆	Color palette register 9	CR9	?? ₁₆ ?? ₁₆
0250 ₁₆ 0251 ₁₆	Color palette register 10	CR10	?? ₁₆ ?? ₁₆
0252 ₁₆ 0253 ₁₆	Color palette register 11	CR11	?? ₁₆ ?? ₁₆
0254 ₁₆ 0255 ₁₆	Color palette register 12	CR12	?? ₁₆ ?? ₁₆
0256 ₁₆ 0257 ₁₆	Color palette register 13	CR13	?? ₁₆ ?? ₁₆
0258 ₁₆ 0259 ₁₆	Color palette register 14	CR14	?? ₁₆ ?? ₁₆
025A ₁₆ 025B ₁₆	Color palette register 15	CR15	?? ₁₆ ?? ₁₆
025C ₁₆			
025D ₁₆	OSD reserved register 1	OR1	00 ₁₆
025E ₁₆	Peripheral clock selection register	PCLKR	03 ₁₆
025F ₁₆	OSD control register 4	OC4	XXXXXX00 ₂
0260 ₁₆	Data slicer 0 control register 1	DSC01	00 ₁₆
0261 ₁₆	Data slicer 0 control register 2	DSC02	?0?0??0?2
0262 ₁₆ 0263 ₁₆	Caption data register 01	CD01	????????2 ????????2
0264 ₁₆ 0265 ₁₆	Caption data register 02	CD02	????????2 ????????2
0266 ₁₆	Caption position register 0	CPS0	00?00000 ₂
0267 ₁₆	Slice standard voltage selection register	SBV0	00 ₁₆
0268 ₁₆	Data slicer 0 reserved register 1	DR01	00 ₁₆
0269 ₁₆	Clock run-in detection register 0	CRD0	00 ₁₆
026A ₁₆	Data clock position register 0	DPS0	X0000000 ₂
026B ₁₆	ID1 control register 0	IDC0	00 ₁₆
026C ₁₆	Standard clock detection register 0	BCD0	XX??????2
026D ₁₆	CRCC data register 0	CRC0	XX000000 ₂
026E ₁₆	Test reservation register 0	IDT0	00 ₁₆
026F ₁₆	Reserved register	RSVREG026F	XXXXXXXX00 ₂
0270 ₁₆ 0271 ₁₆	Left border control register	LBR	XXXXX000 ₂ 00 ₁₆
0272 ₁₆ 0273 ₁₆	Right border control register	RBR	00 ₁₆ XXXXX000 ₂
0274 ₁₆ 0275 ₁₆	Sprite vertical position register 1	VS1	?? ₁₆ ?? ₁₆
0276 ₁₆ 0277 ₁₆	Sprite vertical position register 2	VS2	?? ₁₆ ?? ₁₆
0278 ₁₆ 0279 ₁₆	Sprite horizontal position register	HS	?? ₁₆ XXXXX000 ₂
027A ₁₆	OSD reserved register 4	OR4	X0000000 ₂
027B ₁₆	OSD reserved register 3	OR3	00 ₁₆
027C ₁₆	OSD reserved register 2	OR2	00 ₁₆
027D ₁₆	Peripheral mode register	PM	000XXXXX ₂
027E ₁₆	HSYNC counter register	HC	XXX00X00 ₂
027F ₁₆	HSYNC counter latch		?? ₁₆

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

X : Nothing is mapped to this bit ? : This bit is indeterminate.

Address	Register	Symbol	After reset
0280 ₁₆	Internal oscillation control register 1	DIV0	0016
0281 ₁₆	Internal oscillation control register 2	DIV1	0016
0282 ₁₆	Internal oscillation control register 3	VCO	0016
0283 ₁₆			
0284 ₁₆			
0285 ₁₆			
0286 ₁₆			
0287 ₁₆			
0288 ₁₆			
0289 ₁₆			
028A ₁₆			
028B ₁₆			
028C ₁₆			
028D ₁₆			
028E ₁₆			
028F ₁₆			
0290 ₁₆			
0291 ₁₆			
0292 ₁₆			
0293 ₁₆			
0294 ₁₆			
0295 ₁₆			
0296 ₁₆			
0297 ₁₆			
0298 ₁₆			
0299 ₁₆			
029A ₁₆			
029B ₁₆			
029C ₁₆			
029D ₁₆			
029E ₁₆			
029F ₁₆			
02A0 ₁₆	Flash memory (USER/OSD) change register	FMSEL	0016
02A1 ₁₆			
02A2 ₁₆			
02A3 ₁₆	Flash memory OSD1 control register 4	FMOSA4	X0XXXX002
02A4 ₁₆			
02A5 ₁₆	Flash memory OSD1 control register 1	FMOSA1	XXXXXX0X2
02A6 ₁₆			
02A7 ₁₆	Flash memory OSD1 control register 0	FMOSA0	XX0000012
02A8 ₁₆			
02A9 ₁₆			
02AA ₁₆			
02AB ₁₆			
02AC ₁₆			
02AD ₁₆			
02AE ₁₆			
02AF ₁₆			
02B0 ₁₆			
02B1 ₁₆			
02B2 ₁₆			
02B3 ₁₆	Flash memory OSD2 control register 4	FMOSB4	X0XXXX002
02B4 ₁₆			
02B5 ₁₆	Flash memory OSD2 control register 1	FMOSB1	XXXXXX0X2
02B6 ₁₆			
02B7 ₁₆	Flash memory OSD2 control register 0	FMOSB0	XX0000012
02B8 ₁₆			
02B9 ₁₆			
02BA ₁₆			
02BB ₁₆			
02BC ₁₆			
02BD ₁₆			
02BE ₁₆			
02BF ₁₆			

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

X : Nothing is mapped to this bit ? : This bit is indeterminate.

Address	Register	Symbol	After reset
02C0 ₁₆	Extended register 00	EXTREG02C0	0016
02C1 ₁₆	Extended register 01	EXTREG02C1	0016
02C2 ₁₆	Extended register 02	EXTREG02C2	0016
02C3 ₁₆	Extended register 03	EXTREG02C3	0016
02C4 ₁₆	Extended register 04	EXTREG02C4	0016
02C5 ₁₆	Extended register 05	EXTREG02C5	0016
02C6 ₁₆	Extended register 06	EXTREG02C6	0016
02C7 ₁₆	Extended register 07	EXTREG02C7	0016
02C8 ₁₆	Extended register 08	EXTREG02C8	0016
02C9 ₁₆	Extended register 09	EXTREG02C9	0016
02CA ₁₆	Extended register 0A	EXTREG02CA	0016
02CB ₁₆	Extended register 0B	EXTREG02CB	0016
02CC ₁₆	Extended register 0C	EXTREG02CC	0016
02CD ₁₆	Extended register 0D	EXTREG02CD	0016
02CE ₁₆	Extended register 0E	EXTREG02CE	0016
02CF ₁₆	Extended register 0F	EXTREG02CF	0016
02D0 ₁₆	Extended register 10	EXTREG02D0	0016
02D1 ₁₆	Extended register 11	EXTREG02D1	0016
02D2 ₁₆	Extended register 12	EXTREG02D2	0016
02D3 ₁₆	Extended register 13	EXTREG02D3	0016
02D4 ₁₆	Extended register 14	EXTREG02D4	0016
02D5 ₁₆	Extended register 15	EXTREG02D5	0016
02D6 ₁₆	Extended register 16	EXTREG02D6	0016
02D7 ₁₆	Extended register 17	EXTREG02D7	0016
02D8 ₁₆	Extended register 18	EXTREG02D8	0016
02D9 ₁₆	Extended register 19	EXTREG02D9	0016
02DA ₁₆	Extended register 1A	EXTREG02DA	0016
02DB ₁₆	Extended register 1B	EXTREG02DB	0016
02DC ₁₆	Extended register 1C	EXTREG02DC	0016
02DD ₁₆	Extended register 1D	EXTREG02DD	0016
02DE ₁₆	Extended register 1E	EXTREG02DE	0016
02DF ₁₆	Extended register 1F	EXTREG02DF	0016
02E0 ₁₆	I ² C0 data shift register	IIC0S0	??16
02E1 ₁₆	I ² C0 address register	IIC0S0D	0016
02E2 ₁₆	I ² C0 status register	IIC0S1	0001000?2
02E3 ₁₆	I ² C0 control register	IIC0S1D	0016
02E4 ₁₆	I ² C0 clock control register	IIC0S2	0016
02E5 ₁₆	Reserved register	RSVREG02E5	00?000002
02E6 ₁₆	I ² C0 transmitting buffer register	IIC0S0S	??16
02E7 ₁₆			
02E8 ₁₆	I ² C1 data shift register	IIC1S0	??16
02E9 ₁₆	I ² C1 address register	IIC1S0D	0016
02EA ₁₆	I ² C1 status register	IIC1S1	0001000?2
02EB ₁₆	I ² C1 control register	IIC1S1D	0016
02EC ₁₆	I ² C1 clock control register	IIC1S2	0016
02ED ₁₆	Reserved register	RSVREG02ED	00?000002
02EE ₁₆	I ² C1 transmitting buffer register	IIC1S0S	??16
02EF ₁₆			
02F0 ₁₆	I ² C2 data shift register	IIC2S0	??16
02F1 ₁₆	I ² C2 address register	IIC2S0D	0016
02F2 ₁₆	I ² C2 status register	IIC2S1	0001000?2
02F3 ₁₆	I ² C2 control register	IIC2S1D	0016
02F4 ₁₆	I ² C2 clock control register	IIC2S2	0016
02F5 ₁₆	Reserved register	RSVREG02F5	00?000002
02F6 ₁₆	I ² C2 transmitting buffer register	IIC2S0S	??16
02F7 ₁₆			
02F8 ₁₆			
02F9 ₁₆			
02FA ₁₆			
02FB ₁₆			
02FC ₁₆			
02FD ₁₆			
02FE ₁₆			
02FF ₁₆			

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

X : Nothing is mapped to this bit ? : This bit is indeterminate.

Address	Register	Symbol	After reset
0300 ₁₆	Data slicer 1 control register 1	DSC11	0016
0301 ₁₆	Data slicer 1 control register 2	DSC12	?0?0???2
0302 ₁₆ 0303 ₁₆	Caption data register 11	CD11	????????2 ????????2
0304 ₁₆ 0305 ₁₆	Caption data register 12	CD12	????????2 ????????2
0306 ₁₆	Caption position register 1	CPS1	00?000002
0307 ₁₆	Slice standard voltage selection register	SBV1	0016
0308 ₁₆	Data slicer 1 reserved register 1	DR11	0016
0309 ₁₆	Clock run-in detection register 1	CRD1	0016
030A ₁₆	Data clock position register 1	DPS1	X00000002
030B ₁₆	ID1 control register 1	IDC1	0016
030C ₁₆	Standard clock detection register 1	BCD1	XX??????2
030D ₁₆	CRCC data register 1	CRC1	XX0000002
030E ₁₆	Test reservation register 1	IDT1	0016
030F ₁₆	Reserved register	RSVREG030F	XXXXXXXX02
0310 ₁₆			
0311 ₁₆			
0312 ₁₆			
0313 ₁₆			
0314 ₁₆			
0315 ₁₆			
0316 ₁₆			
0317 ₁₆			
0318 ₁₆			
0319 ₁₆			
031A ₁₆			
031B ₁₆			
031C ₁₆	ID1 reserved register 0	IRSV0	00??????2
031D ₁₆	ID1 reserved register 1	IRSV1	00??????2
031E ₁₆			
031F ₁₆			
0320 ₁₆			
0321 ₁₆			
0322 ₁₆			
0323 ₁₆			
0324 ₁₆			
0325 ₁₆			
0326 ₁₆			
0327 ₁₆			
0328 ₁₆			
0329 ₁₆			
032A ₁₆			
032B ₁₆			
032C ₁₆			
032D ₁₆			
032E ₁₆			
032F ₁₆			
0330 ₁₆			
0331 ₁₆			
0332 ₁₆			
0333 ₁₆			
0334 ₁₆			
0335 ₁₆			
0336 ₁₆			
0337 ₁₆			
0338 ₁₆			
0339 ₁₆			
033A ₁₆			
033B ₁₆			
033C ₁₆			
033D ₁₆			
033E ₁₆			
033F ₁₆			

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

X : Nothing is mapped to this bit ? : This bit is indeterminate.

Address	Register	Symbol	After reset
0340 ₁₆	Timer B3, 4, 5 count start flag	TBSR	000XXXXX2
0341 ₁₆			
0342 ₁₆	Reserved register	RSVREG0342	??16
0343 ₁₆	Reserved register	RSVREG0343	??16
0344 ₁₆	Reserved register	RSVREG0344	??16
0345 ₁₆	Reserved register	RSVREG0345	??16
0346 ₁₆	Reserved register	RSVREG0346	??16
0347 ₁₆	Reserved register	RSVREG0347	??16
0348 ₁₆	Reserved register	RSVREG0348	0016
0349 ₁₆	Reserved register	RSVREG0349	0016
034A ₁₆	Reserved register	RSVREG034A	0016
034B ₁₆	Reserved register	RSVREG034B	0016
034C ₁₆	Reserved register	RSVREG034C	??16
034D ₁₆	Reserved register	RSVREG034D	??16
034E ₁₆			
034F ₁₆			
0350 ₁₆	Timer B3 register	TB3	??16
0351 ₁₆			??16
0352 ₁₆	Timer B4 register	TB4	??16
0353 ₁₆			??16
0354 ₁₆	Timer B5 register	TB5	??16
0355 ₁₆			??16
0356 ₁₆			
0357 ₁₆			
0358 ₁₆			
0359 ₁₆			
035A ₁₆			
035B ₁₆	Timer B3 mode register	TB3MR	00??00002
035C ₁₆	Timer B4 mode register	TB4MR	00?X00002
035D ₁₆	Timer B5 mode register	TB5MR	00?X00002
035E ₁₆	Interrupt cause select register 2	IFSR2A	00XXXXXX2
035F ₁₆	Interrupt cause select register	IFSR	0016
0360 ₁₆	Reserved register	RSVREG0360	??16
0361 ₁₆			
0362 ₁₆	Reserved register	RSVREG0362	010000002
0363 ₁₆	Reserved register	RSVREG0363	??16
0364 ₁₆	Reserved register	RSVREG0364	??16
0365 ₁₆			
0366 ₁₆	Reserved register	RSVREG0366	010000002
0367 ₁₆	Reserved register	RSVREG0367	??16
0368 ₁₆			
0369 ₁₆			
036A ₁₆			
036B ₁₆			
036C ₁₆	UART0 special mode register 4	U0SMR4	0016
036D ₁₆	UART0 special mode register 3	U0SMR3	000X0X0X2
036E ₁₆	UART0 special mode register 2	U0SMR2	X00000002
036F ₁₆	UART0 special mode register	U0SMR	X00000002
0370 ₁₆	UART1 special mode register 4	U1SMR4	0016
0371 ₁₆	UART1 special mode register 3	U1SMR3	000X0X0X2
0372 ₁₆	UART1 special mode register 2	U1SMR2	X00000002
0373 ₁₆	UART1 special mode register	U1SMR	X00000002
0374 ₁₆	UART2 special mode register 4	U2SMR4	0016
0375 ₁₆	UART2 special mode register 3	U2SMR3	000X0X0X2
0376 ₁₆	UART2 special mode register 2	U2SMR2	X00000002
0377 ₁₆	UART2 special mode register	U2SMR	X00000002
0378 ₁₆	UART2 transmit/receive mode register	U2MR	0016
0379 ₁₆	UART2 bit rate generator	U2BRG	??16
037A ₁₆	UART2 transmit buffer register	U2TB	????????2
037B ₁₆			XXXXXXXX?2
037C ₁₆	UART2 transmit/receive control register 0	U2C0	000010002
037D ₁₆	UART2 transmit/receive control register 1	U2C1	000000102
037E ₁₆	UART2 receive buffer register	U2RB	????????2
037F ₁₆			?????XX?2

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

X : Nothing is mapped to this bit ? : This bit is indeterminate.

Address	Register	Symbol	After reset
0380 ₁₆	Count start flag	TABSR	0016
0381 ₁₆	Clock prescaler reset flag	CPSRF	0XXXXXXX2
0382 ₁₆	One-shot start flag	ONSF	0016
0383 ₁₆	Trigger select register	TRGSR	0016
0384 ₁₆	Up-down flag	UDF	0016
0385 ₁₆			
0386 ₁₆	Timer A0 register	TA0	??16
0387 ₁₆			??16
0388 ₁₆	Timer A1 register	TA1	??16
0389 ₁₆			??16
038A ₁₆	Timer A2 register	TA2	??16
038B ₁₆			??16
038C ₁₆	Timer A3 register	TA3	??16
038D ₁₆			??16
038E ₁₆	Timer A4 register	TA4	??16
038F ₁₆			??16
0390 ₁₆	Timer B0 register	TB0	??16
0391 ₁₆			??16
0392 ₁₆	Timer B1 register	TB1	??16
0393 ₁₆			??16
0394 ₁₆	Timer B2 register	TB2	??16
0395 ₁₆			??16
0396 ₁₆	Timer A0 mode register	TA0MR	0016
0397 ₁₆	Timer A1 mode register	TA1MR	0016
0398 ₁₆	Timer A2 mode register	TA2MR	0016
0399 ₁₆	Timer A3 mode register	TA3MR	0016
039A ₁₆	Timer A4 mode register	TA4MR	0016
039B ₁₆	Timer B0 mode register	TB0MR	00??00002
039C ₁₆	Timer B1 mode register	TB1MR	00?X00002
039D ₁₆	Timer B2 mode register	TB2MR	00?X00002
039E ₁₆	Reserved register	RSVREG039E	XXXXXX002
039F ₁₆			
03A0 ₁₆	UART0 transmit/receive mode register	U0MR	0016
03A1 ₁₆	UART0 bit rate generator	U0BRG	??16
03A2 ₁₆	UART0 transmit buffer register	U0TB	???????2
03A3 ₁₆			XXXXXXXX?2
03A4 ₁₆	UART0 transmit/receive control register 0	U0C0	000010002
03A5 ₁₆	UART0 transmit/receive control register 1	U0C1	000000102
03A6 ₁₆	UART0 receive buffer register	U0RB	???????2
03A7 ₁₆			?????XX?2
03A8 ₁₆	UART1 transmit/receive mode register	U1MR	0016
03A9 ₁₆	UART1 bit rate generator	U1BRG	??16
03AA ₁₆	UART1 transmit buffer register	U1TB	???????2
03AB ₁₆			XXXXXXXX?2
03AC ₁₆	UART1 transmit/receive control register 0	U1C0	000010002
03AD ₁₆	UART1 transmit/receive control register 1	U1C1	000000102
03AE ₁₆	UART1 receive buffer register	U1RB	???????2
03AF ₁₆			?????XX?2
03B0 ₁₆	UART transmit/receive control register 2	UCON	X00000002
03B1 ₁₆			
03B2 ₁₆			
03B3 ₁₆			
03B4 ₁₆			
03B5 ₁₆			
03B6 ₁₆			
03B7 ₁₆			
03B8 ₁₆	DMA0 request cause select register	DM0SL	0016
03B9 ₁₆			
03BA ₁₆	DMA1 request cause select register	DM1SL	0016
03BB ₁₆			
03BC ₁₆	Reserved register	RSVREG03BC	??16
03BD ₁₆	Reserved register	RSVREG03BD	??16
03BE ₁₆	Reserved register	RSVREG03BE	??16
03BF ₁₆			

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

X : Nothing is mapped to this bit ? : This bit is indeterminate.

Address	Register	Symbol	After reset
03C0 ₁₆	Reserved register	RSVREG03C0	?????????
03C1 ₁₆	Reserved register	RSVREG03C1	XXXXXX???
03C2 ₁₆	Reserved register	RSVREG03C2	?????????
03C3 ₁₆	Reserved register	RSVREG03C3	XXXXXX???
03C4 ₁₆	Reserved register	RSVREG03C4	?????????
03C5 ₁₆	Reserved register	RSVREG03C5	XXXXXX???
03C6 ₁₆ 03C7 ₁₆	A/D register 3	AD3	????????? XXXXXX???
03C8 ₁₆ 03C9 ₁₆	A/D register 4	AD4	????????? XXXXXX???
03CA ₁₆ 03CB ₁₆	A/D register 5	AD5	????????? XXXXXX???
03CC ₁₆ 03CD ₁₆	A/D register 6	AD6	????????? XXXXXX???
03CE ₁₆ 03CF ₁₆	A/D register 7	AD7	????????? XXXXXX???
03D0 ₁₆			
03D1 ₁₆			
03D2 ₁₆			
03D3 ₁₆			
03D4 ₁₆ 03D5 ₁₆	A/D control register 2	ADCON2	0016
03D6 ₁₆	A/D control register 0	ADCON0	00000???
03D7 ₁₆	A/D control register 1	ADCON1	0016
03D8 ₁₆ 03D9 ₁₆	Reserved register	RSVREG03D8	??16
03DA ₁₆ 03DB ₁₆	Reserved register	RSVREG03DA	??16
03DC ₁₆ 03DD ₁₆	Reserved register	RSVREG03DC	0016
03DE ₁₆	Reserved register	RSVREG03DE	XX00XXXX2
03DF ₁₆	Reserved register	RSVREG03DF	0016
03E0 ₁₆	Port P0 register	P0	??16
03E1 ₁₆	Port P1 register	P1	??16
03E2 ₁₆	Port P0 direction register	PD0	0016
03E3 ₁₆	Port P1 direction register	PD1	0016
03E4 ₁₆	Port P2 register	P2	??16
03E5 ₁₆	Port P3 register	P3	??16
03E6 ₁₆	Port P2 direction register	PD2	0016
03E7 ₁₆	Port P3 direction register	PD3	0016
03E8 ₁₆	Port P4 register	P4	??16
03E9 ₁₆	Port P5 register	P5	??16
03EA ₁₆	Port P4 direction register	PD4	0016
03EB ₁₆	Port P5 direction register	PD5	0016
03EC ₁₆	Port P6 register	P6	??16
03ED ₁₆	Port P7 register	P7	??16
03EE ₁₆	Port P6 direction register	PD6	0016
03EF ₁₆	Port P7 direction register	PD7	0016
03F0 ₁₆	Port P8 register	P8	??16
03F1 ₁₆	Port P9 register	P9	??16
03F2 ₁₆	Port P8 direction register	PD8	00X000002
03F3 ₁₆	Port P9 direction register	PD9	0016
03F4 ₁₆	Port P10 register	P10	??16
03F5 ₁₆	Reserved register	RSVREG03F5	??16
03F6 ₁₆	Port P10 direction register	PD10	0016
03F7 ₁₆	Reserved register	RSVREG03F7	0016
03F8 ₁₆	Reserved register	RSVREG03F8	??16
03F9 ₁₆	Reserved register	RSVREG03F9	??16
03FA ₁₆	Reserved register	RSVREG03FA	0016
03FB ₁₆	Reserved register	RSVREG03FB	0016
03FC ₁₆	Pull-up control register 0	PUR0	0016
03FD ₁₆	Pull-up control register 1	PUR1	000000002 000000102 (Note 2)
03FE ₁₆	Pull-up control register 2	PUR2	0016
03FF ₁₆	Port control register	PCR	0016

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

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

- "000000002" where "L" is inputted to the CNVss1 pin
- "000000102" where "H" is inputted to the CNVss1 pin

At software reset, watchdog timer reset, the register is as follows:

- "000000002" where the PM01 to PM00 bits in the PM0 register are "002" (single-chip mode)
- "000000102" where the PM01 to PM00 bits in the PM0 register are "002" (memory expansion mode) or "112" (microprocessor mode)

X : Nothing is mapped to this bit

? : This bit is indeterminate.

Reset

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

Hardware Reset

A reset is applied using the $\overline{\text{RESET}}$ pin. When an “L” signal is applied to the $\overline{\text{RESET}}$ pin while the power supply voltage is within the recommended operating condition, the pins are initialized (see Table 3.1. Pin Status When $\overline{\text{RESET}}$ Pin Level is “L”). 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 3.1 shows the example reset circuit. Figure 3.2 shows the reset sequence. Table 3.1 shows the status of the other pins while the $\overline{\text{RESET}}$ pin is “L”. Figure 3.3 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 X_{IN} 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 X_{IN} pin.
- (5) Apply an “H” signal to the $\overline{\text{RESET}}$ pin.

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.

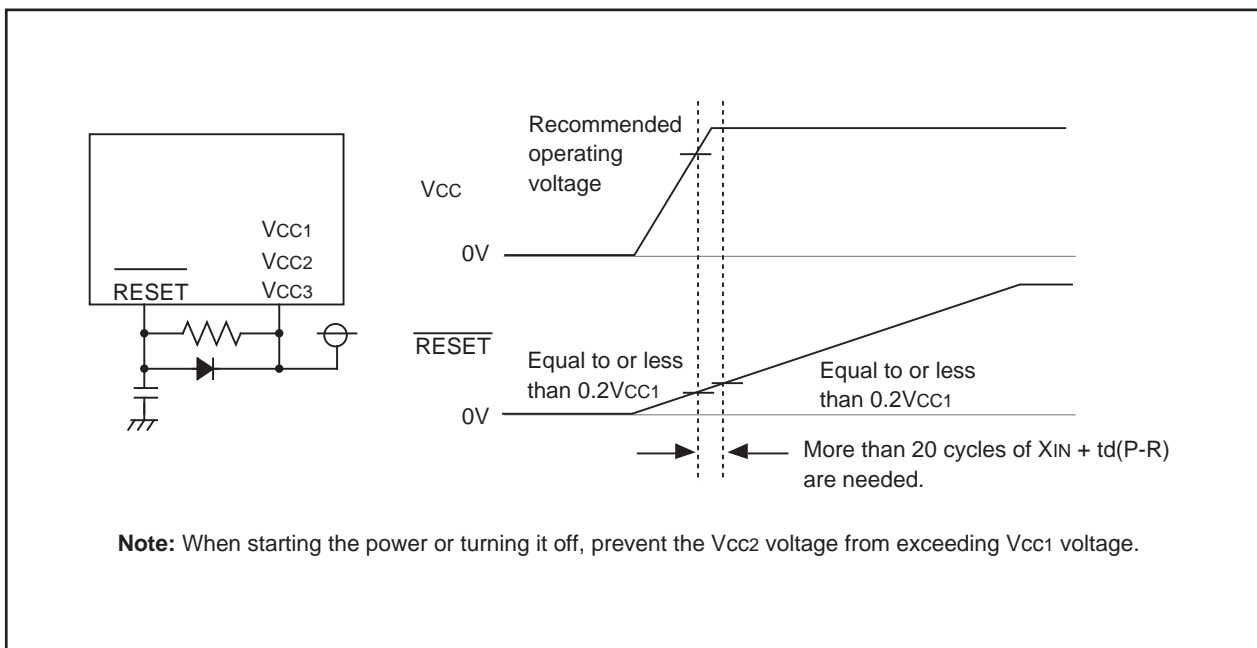


Figure 3.1 shows the example reset circuit

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.

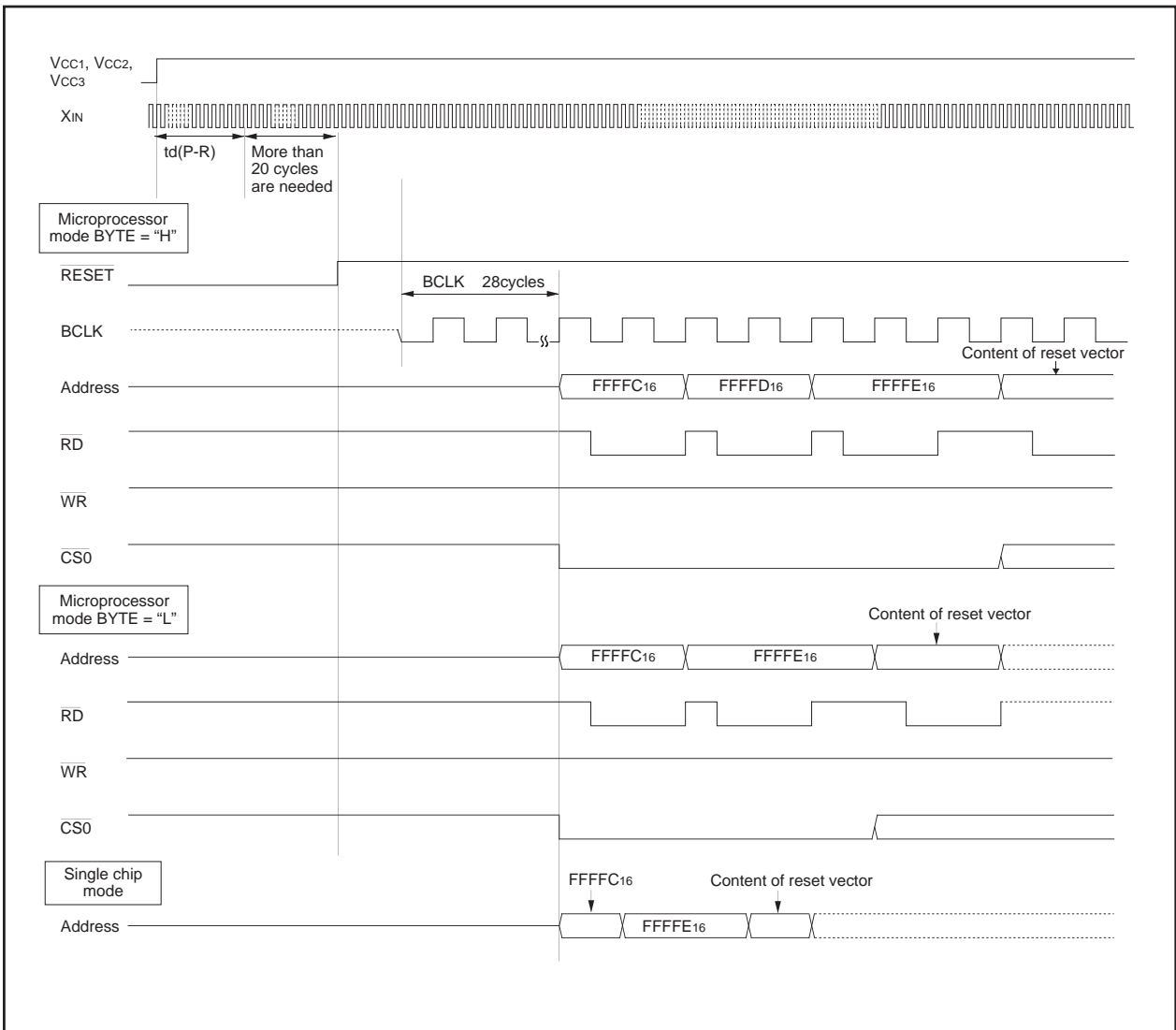


Figure 3.2. Reset sequence

Table 3.1. Pin Status When $\overline{\text{RESET}}$ Pin Level is "L"

Pin name	Status		
	CNVss1 = Vss	CNVss1 = 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	$\overline{\text{CS0}}$ output ("H" is output)	$\overline{\text{CS0}}$ output ("H" is output)
P45 to P47	Input port	Input port (Pulled high)	Input port (Pulled high)
P50	Input port	$\overline{\text{WR}}$ output ("H" is output)	$\overline{\text{WR}}$ output ("H" is output)
P51	Input port	$\overline{\text{BHE}}$ output (indeterminate)	$\overline{\text{BHE}}$ output (indeterminate)
P52	Input port	$\overline{\text{RD}}$ output ("H" is output)	$\overline{\text{RD}}$ output ("H" is output)
P53	Input port	BCLK output	BCLK output
P54	Input port	$\overline{\text{HLDA}}$ output (The output value depends on the input to the HOLD pin)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the HOLD pin)
P55	Input port	HOLD input	HOLD input
P56	Input port	ALE output ("L" is output)	ALE output ("L" is output)
P57	Input port	$\overline{\text{RDY}}$ input	$\overline{\text{RDY}}$ input
P6, P7, P8, P9, P10	Input port	Input port	Input port
OSC2/VSYNC1, OSCOOUT, DIGR1, HSYNC SCL4, SDA4, SCL5 SDA5, SCL6, SDA6	Output state		
DIGR1, DIGG1 DIGB1, DIGR2, DIGG2 DIGB2	Output state (undefined)		
R/DIGR0, G/DIGG0, B/DIGB0, OUT1, OUT2, OSC1/OSCHLF, CVIN1 VHOLD1, HLF1, CVIN2, VHOLD2, HLF2	Input state		

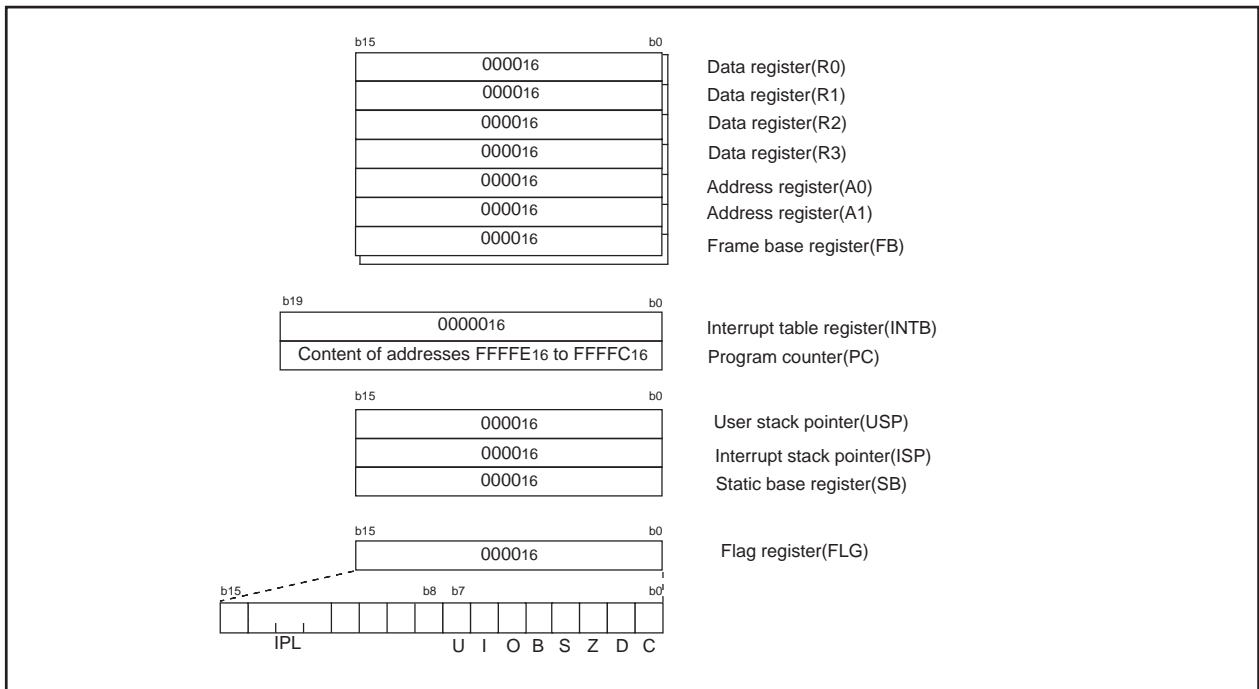


Figure 3.3. CPU Register Status After Reset

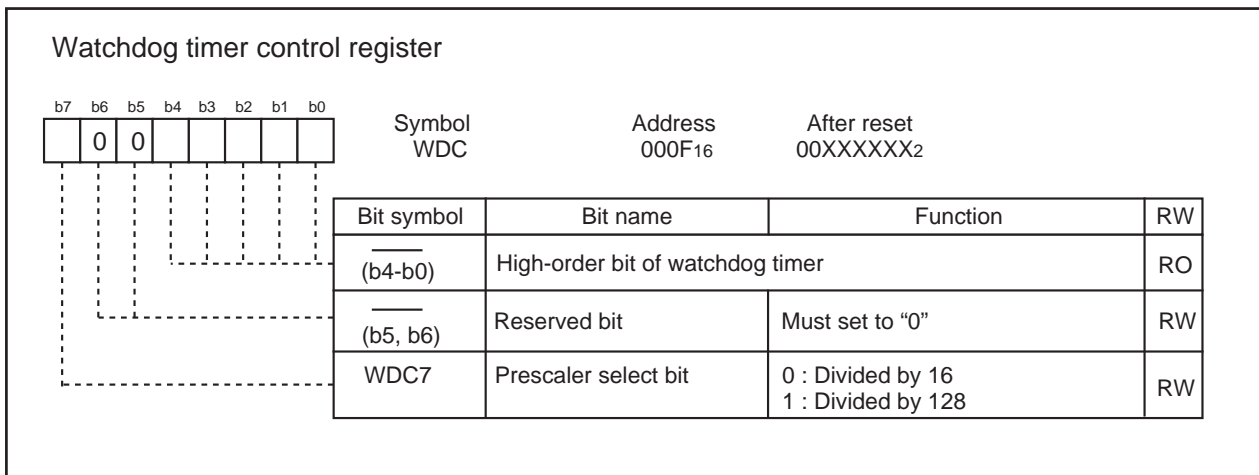


Figure 3.4. WDC Register

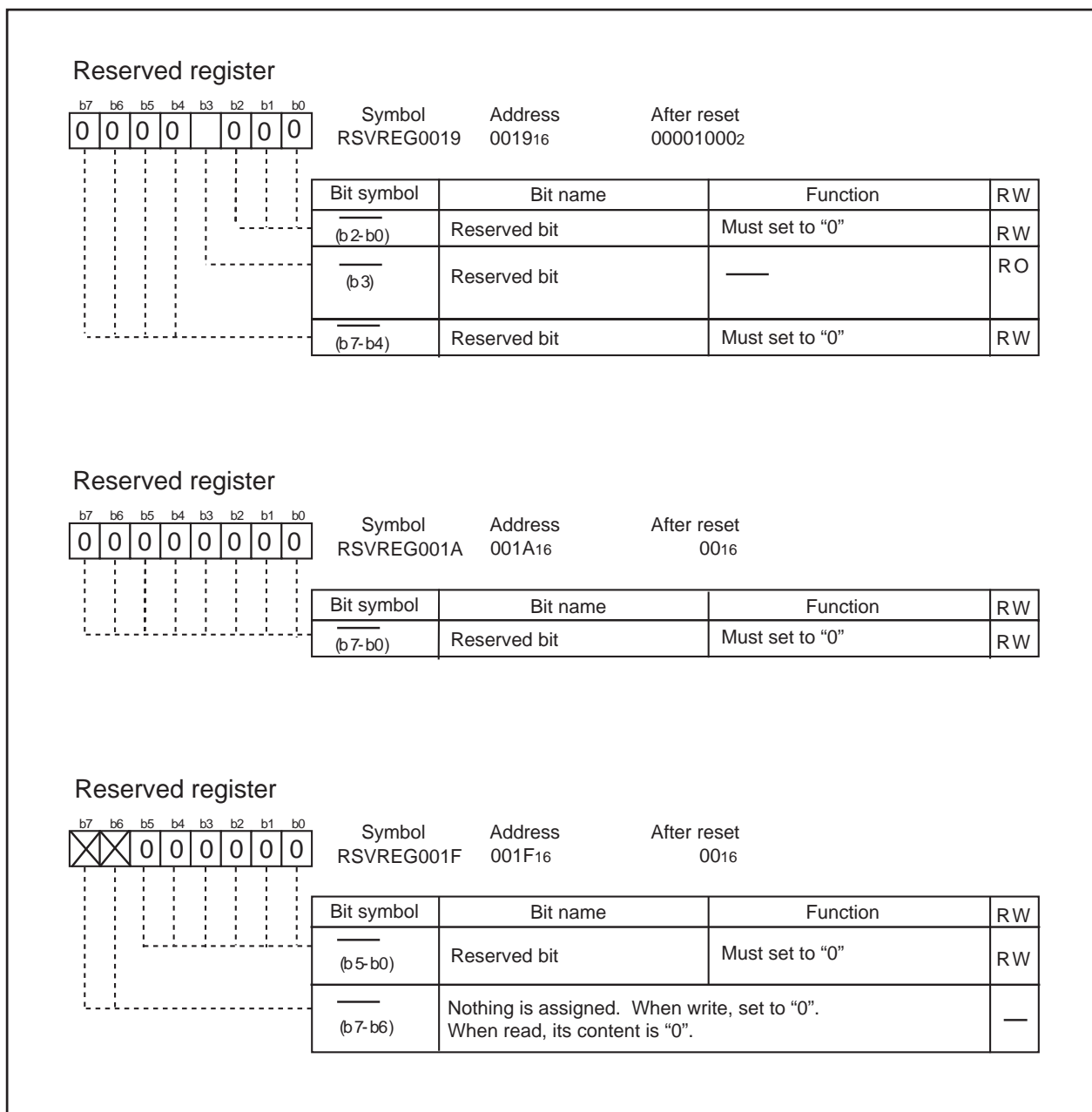


Figure 3.5. Reserved register

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 4.1 shows the features of these processor modes.

Table 4.1. Features of Processor Modes

Processor modes	Access space	Pins which are assigned I/O ports
Single-chip mode	SFR, internal RAM, internal ROM, OSDRAM, OSDROM	All pins are I/O ports or peripheral function I/O pins
Memory expansion mode	SFR, internal RAM, internal ROM, external area (Note), OSDRAM, OSDROM	Some pins serve as bus control pins (Note)
Microprocessor mode	SFR, internal RAM, external area (Note), OSDRAM, OSDROM	Some pins serve as bus control pins (Note)

Note : Refer to "Bus".

(2) Setting Processor Modes

Processor mode is set by using the CNVSS1 pin and the PM01 to PM00 bits in the PM0 register.

Table 4.2 shows the processor mode after hardware reset. Table 4.3 shows the PM01 to PM00 bit set values and processor modes.

Table 4.2. Processor Mode After Hardware Reset

CNVSS1 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 CNVSS1 pin (hardware reset 1 or hardware reset 2), the internal ROM cannot be accessed regardless of PM01 to PM00 bits.

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

Table 4.3. 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 CNVSS1 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 CNVSS1 pin (hardware reset 1 or hardware reset 2), the internal ROM cannot be accessed regardless of PM01 to PM00 bits.

Figures 4.1 and 4.2 show the registers associated with processor modes. Figure 4.3 show the memory map in single chip mode.

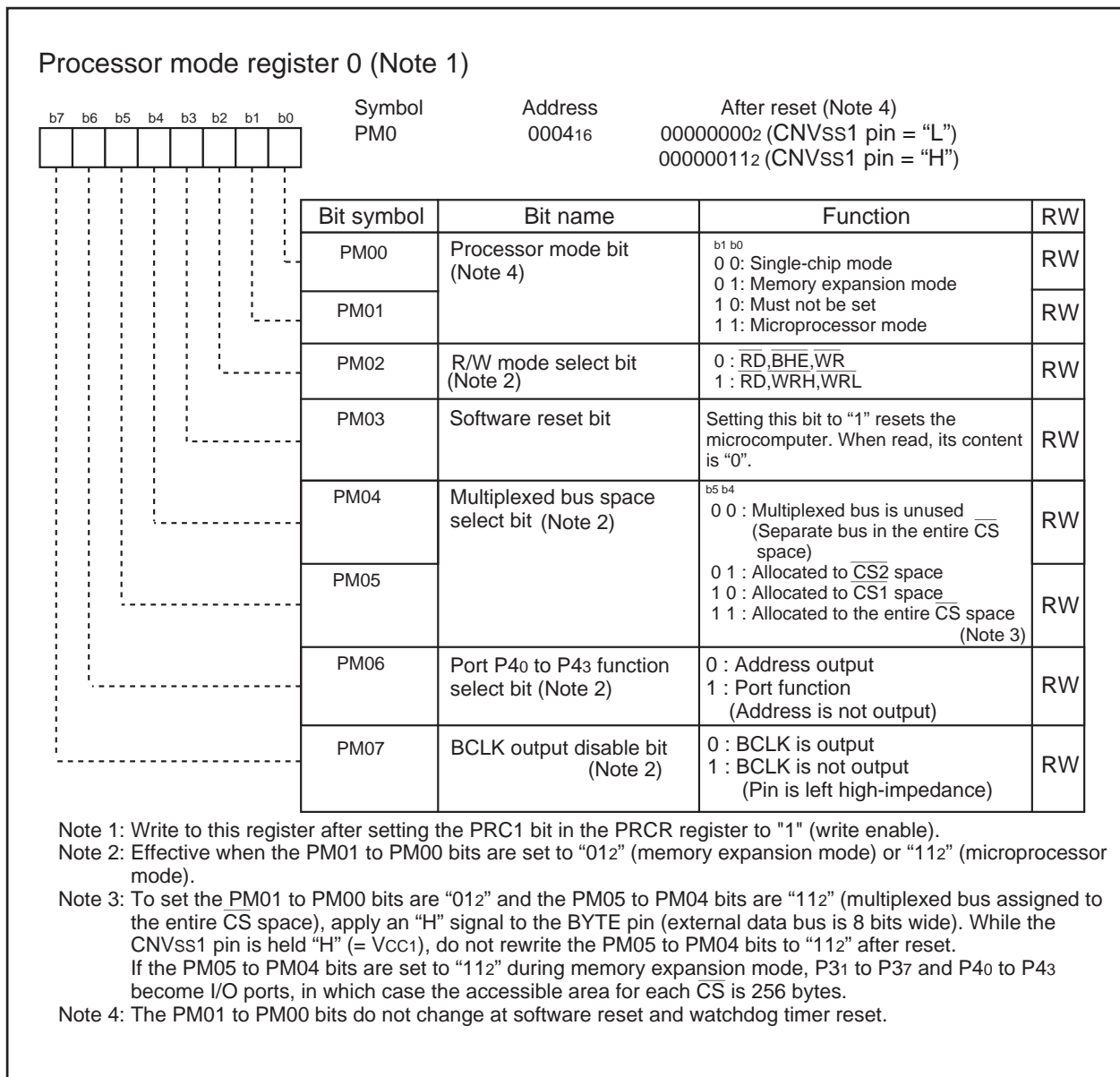


Figure 4.1. PM0 Register

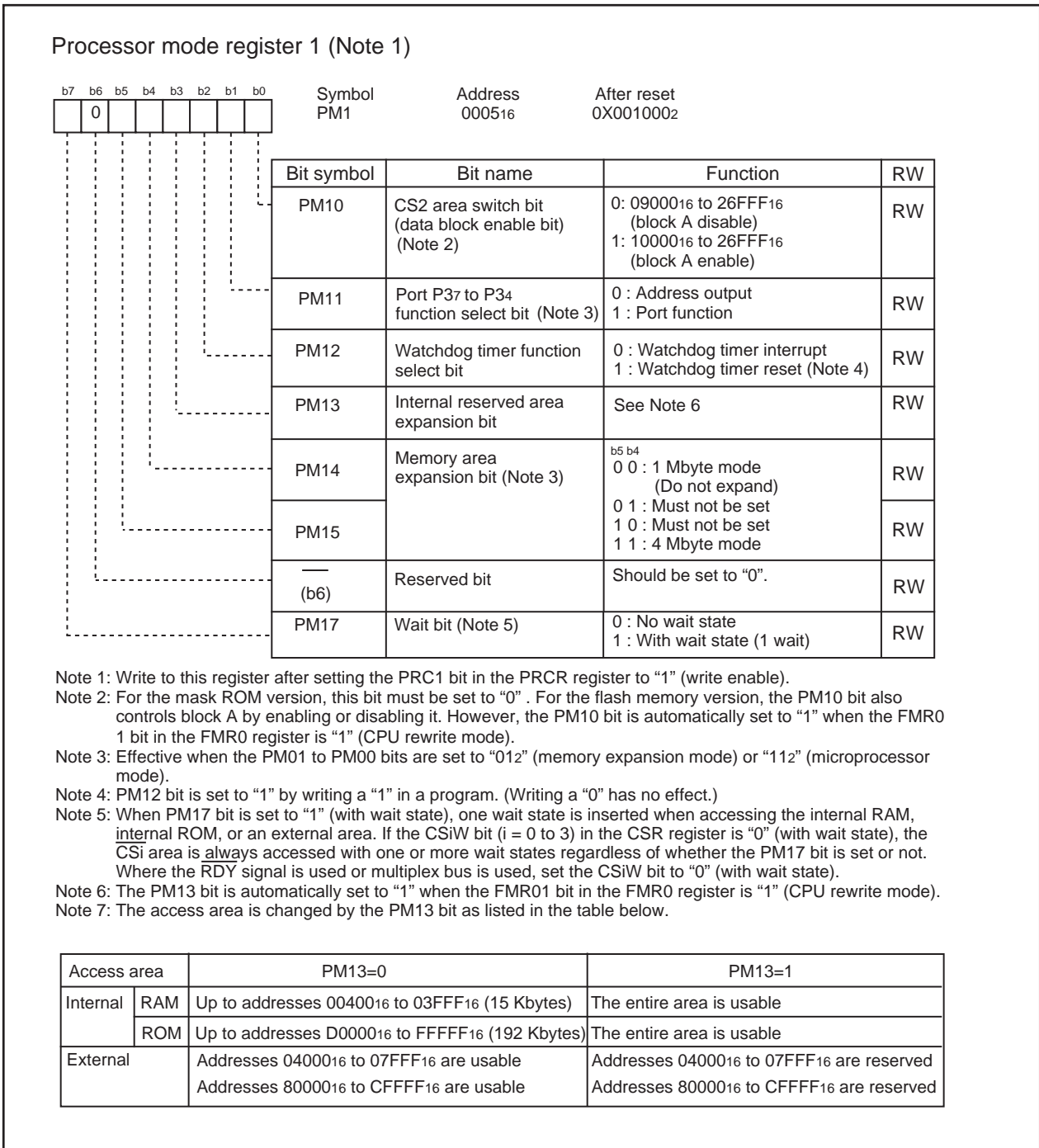


Figure 4.2. PM1 Register

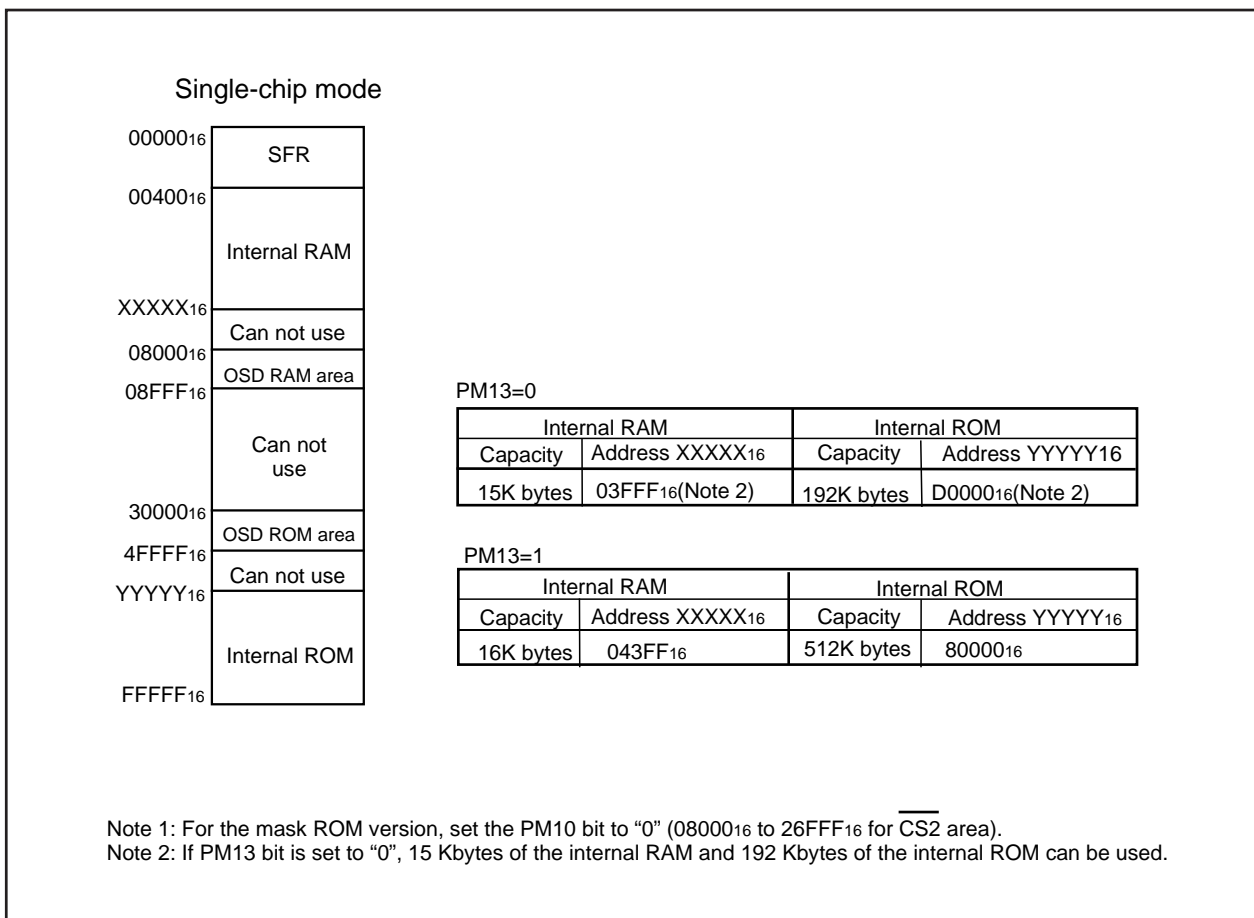


Figure 4.3. Memory Map in Single Chip Mode

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 BCLK.

Bus Mode

The bus mode, either multiplexed or separate, can be selected using the PM05 to PM04 bits in the PM0 register.

Separate Bus

In this bus mode, data and address are separate.

Multiplexed Bus

D0 to D7 and A1 to A8 are multiplexed. D8 to D15 are not multiplexed. Do not use D8 to D15. External buses connecting to a multiplexed bus are allocated to only the even addresses of the micro-computer. Odd addresses cannot be accessed.

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 4.4 shows the PM06 and PM11 bit set values and address bus widths.

Table 4.4. 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

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}) 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 4.4 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 4.5 shows the example of address bus and \overline{CS}_i signal output in 1 Mbyte mode.

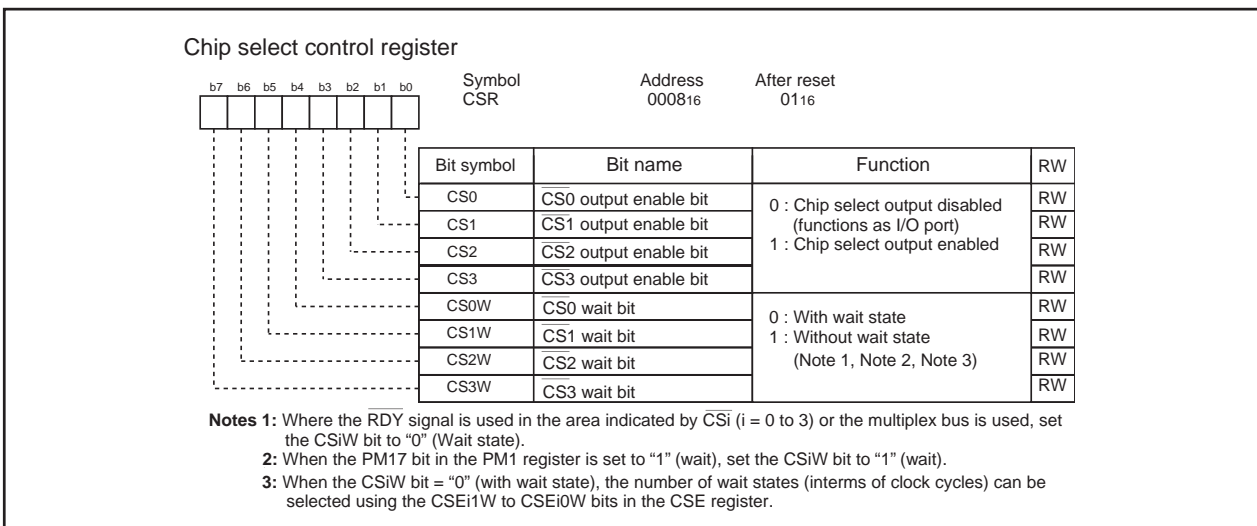


Figure 4.4. CSR Register

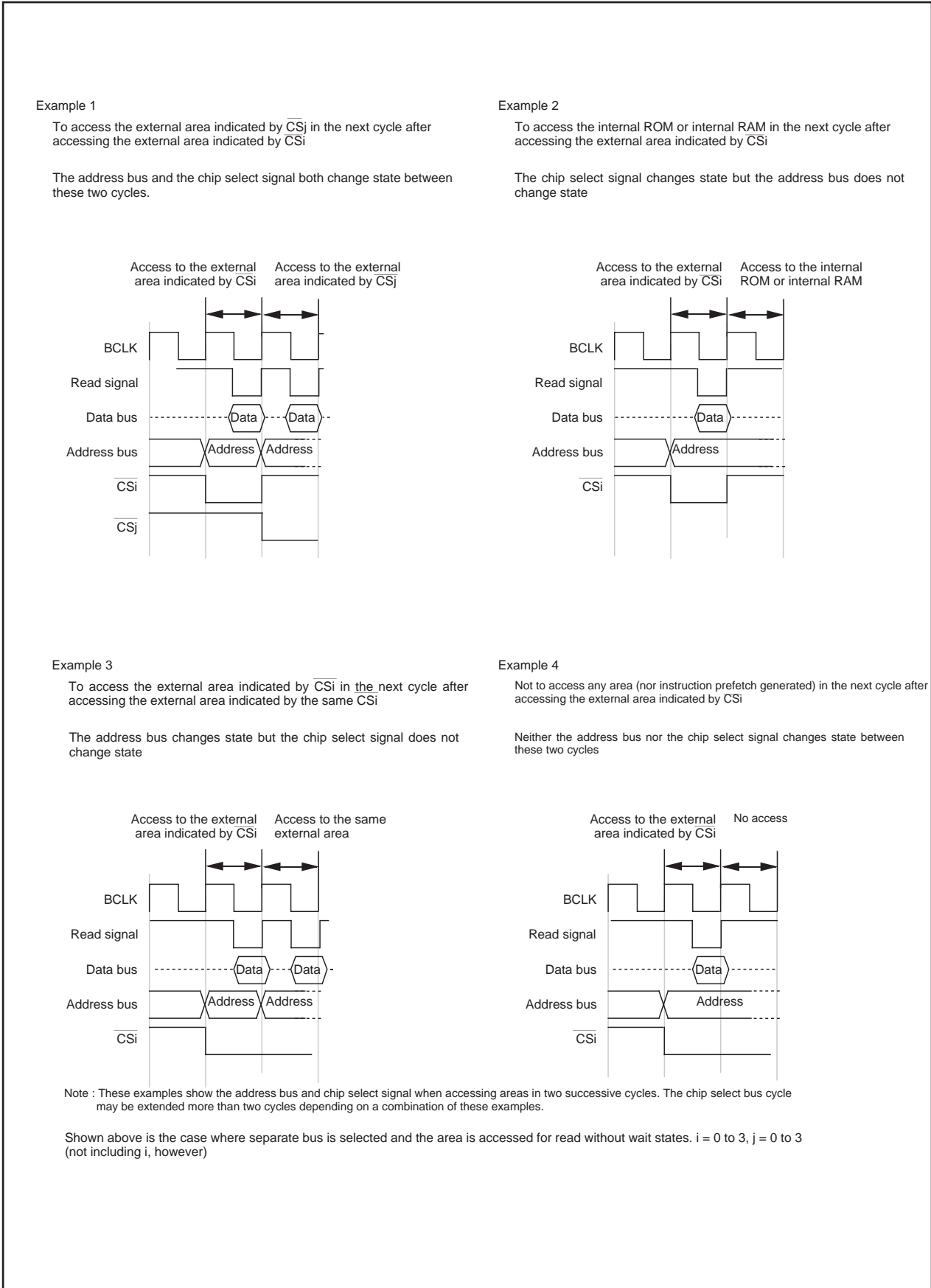


Figure 4.5. Example of Address Bus and \overline{CSi} Signal Output in 1 Mbyte Mode

(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 4.5 shows the operation of \overline{RD} , \overline{WRL} , and \overline{WRH} signals. Table 4.6 shows the operation of operation of \overline{RD} , \overline{WR} , and \overline{BHE} signals.

Table 4.5. 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 4.6. 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

(5) ALE Signal

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

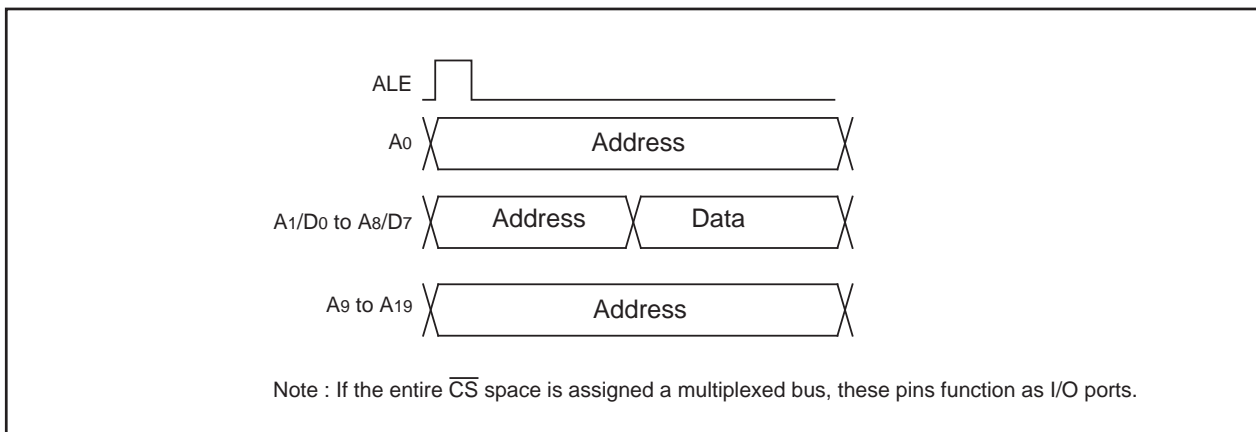


Figure 4.6. ALE Signal, Address Bus, Data Bus

(6) The $\overline{\text{RDY}}$ Signal

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

A0 to A19, D0 to D15, $\overline{\text{CS}}_0$ to $\overline{\text{CS}}_3$, $\overline{\text{RD}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, $\overline{\text{ALE}}$, $\overline{\text{HLDA}}$

Then, when the input on the $\overline{\text{RDY}}$ pin is detected high at the falling edge of BCLK, the remaining bus cycle is executed. Figure 4.7 shows example in which the wait state was inserted into the read cycle by the $\overline{\text{RDY}}$ signal. To use the $\overline{\text{RDY}}$ signal, set the corresponding bit (CS3W to CS0W bits) in the CSR register to "0" (with wait state). When not using the $\overline{\text{RDY}}$ signal, process the $\overline{\text{RDY}}$ pin as an unused pin.

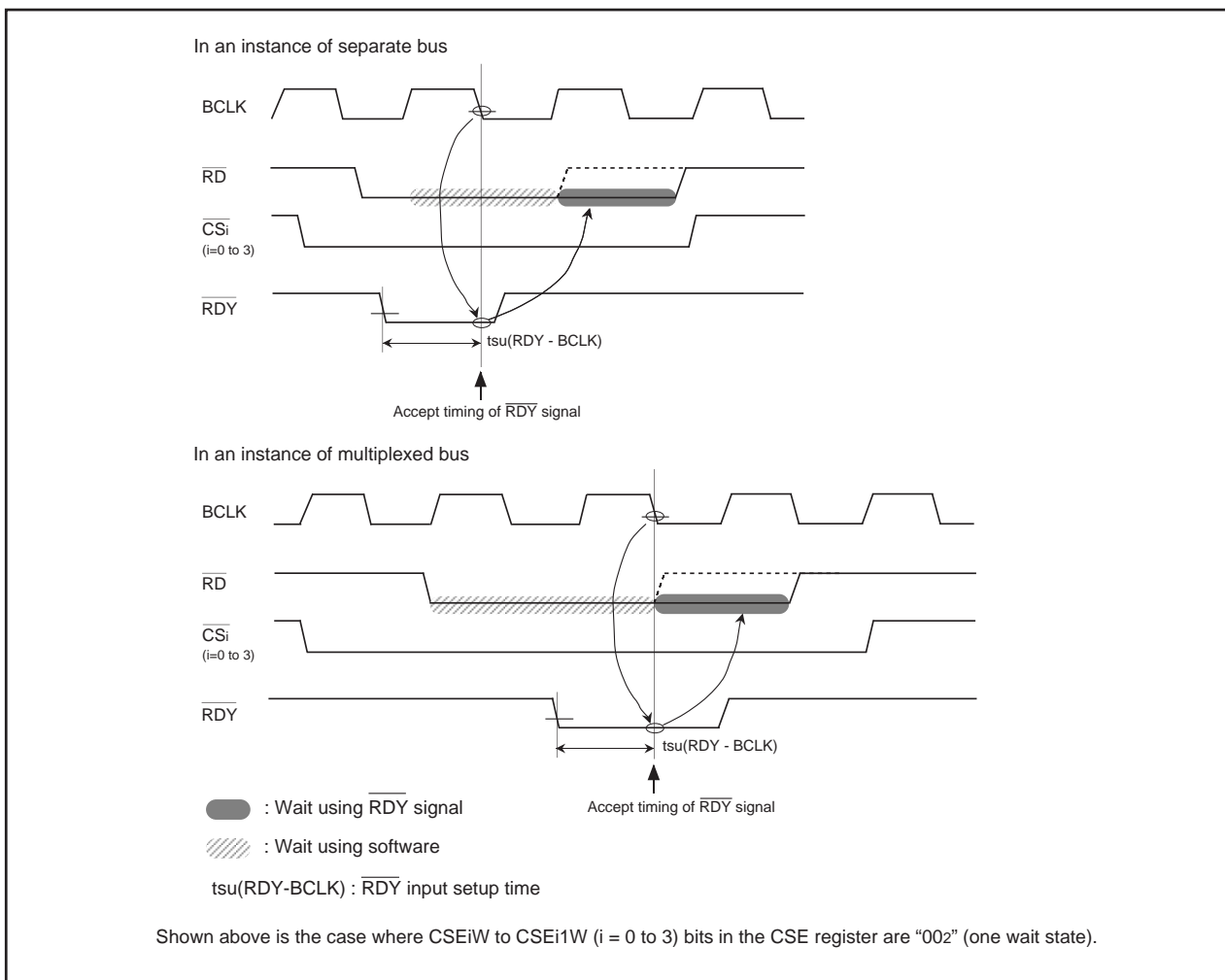


Figure 4.7. Example in which Wait State was Inserted into Read Cycle by $\overline{\text{RDY}}$ Signal

(7) $\overline{\text{HOLD}}$ Signal

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

Table 4.7 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. However, if the CPU is accessing an odd address in word units, the DMAC cannot gain control of the bus during two separate accesses.

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

Figure 4.8. Bus-using Priorities

Table 4.7. Microcomputer Status in Hold State

Item		Status
BCLK		Output
A ₀ to A ₁₉ , D ₀ to D ₁₅ , $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, RD, WRL, WRH, WR, BHE		High-impedance
I/O ports	P0, P1, P3, P4(Note 1)	High-impedance
	P6 to P10	Maintains status when hold signal is received
$\overline{\text{HLDA}}$		Output "L"
Internal peripheral circuits		ON (but watchdog timer stops)
ALE signal		Indeterminate

Note 1: 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.

Table 4.8. Pin Functions for Each Processor Mode

Processor mode	Memory expansion mode or microprocessor 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)
Data bus width BYTE pin	16 bits "L"	16 bits "L"
P00 to P07	D0 to D7	D0 to D7(Note)
P10 to P17	D8 to D15	D8 to D15(Note)
P20	A0	A0
P21 to P27	A1 to A7	A1 to A7/D0 to D6 (Note 2)
P30	A8	A8/D7(Note 2)
P31 to P33	A9 to A11	
P34 to P37	PM11=0	A12 to A15
	PM11=1	I/O ports
P40 to P43	PM06=0	A16 to A19
	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	$\overline{\text{WR}}$
	PM02=1	WRL
P51	PM02=0	$\overline{\text{BHE}}$
	PM02=1	WRH
P52	RD	
P53	BCLK	
P54	HLDA	
P55	$\overline{\text{HOLD}}$	
P56	ALE	
P57	RDY	

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

Note 1: When Vcc1 is inputted into CNVss1 pin, do not set bits PM05 and PM04 to "112" after reset.

Since P31 to P37, and P40 to P43 become I/O ports when bits PM05 and PM04 are set to "112" in memory extension mode, the area which can be accessed is 256 bytes per CS.

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

Note 3: When accessing the area that uses a multiplexed bus, these pins output an indeterminate value during a write.

(9) External Bus Status When Internal Area Accessed

Table 4.9 shows the external bus status when the internal area is accessed.

Table 4.9. 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. The SFR area is unaffected by these control bits. This area is always accessed in 2 BCLK.

To use the RDY signal, set the corresponding CS0W to CS3W bit to "0"(with wait state). Figure 4.9 shows the CSE register. Table 4.10 shows the software wait related bits and bus cycles. Figure 4.10 and 4.11 show the typical bus timings using software wait.

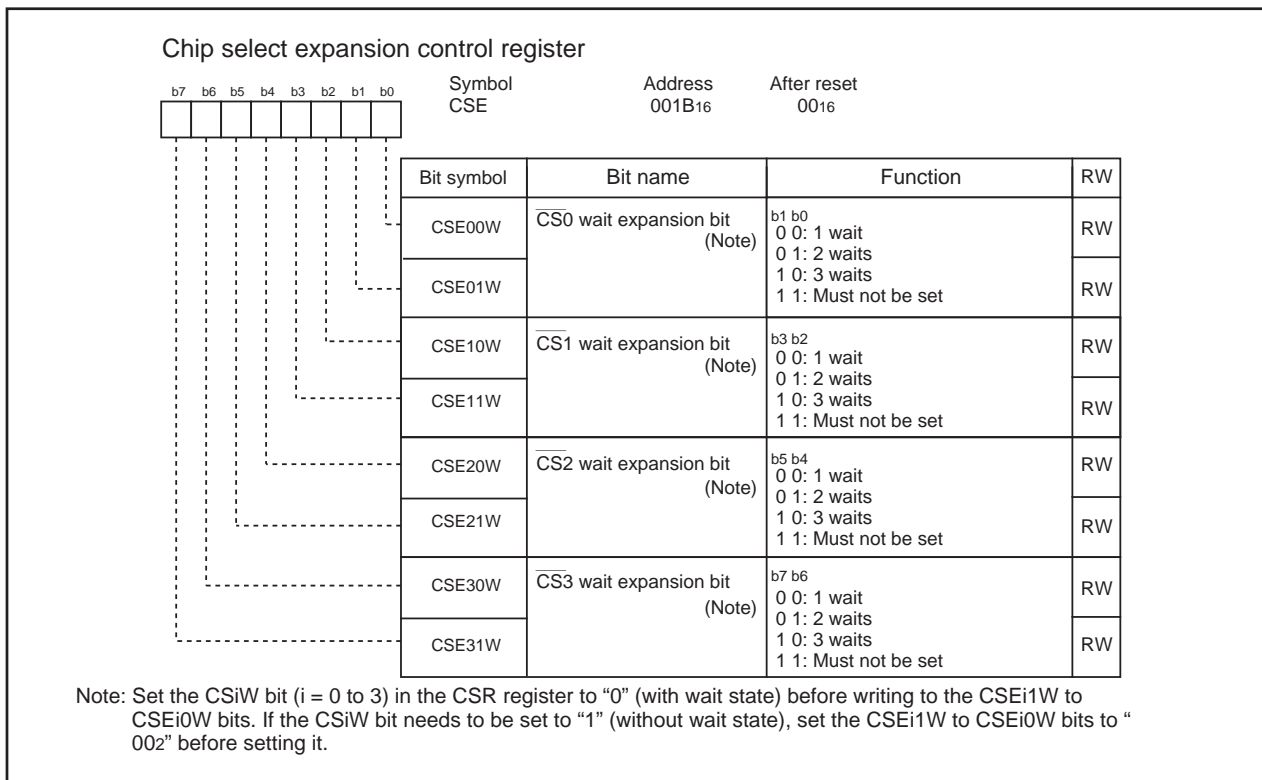


Figure 4.9. CSE Register

Table 4.10. 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
SFR	—	—	—	—	—	2 BCLK 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
		1	0	002	1 wait	3 BCLK cycles

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

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

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 CS0 to CS3). Therefore, the internal RAM and internal ROM are accessed with no wait states, and all external areas are accessed with one wait state.

4: When the PM17 bit is set to "1" and accessing external area, set the CSiW bit (0 to 3) in the CSR register to "0" (wait).

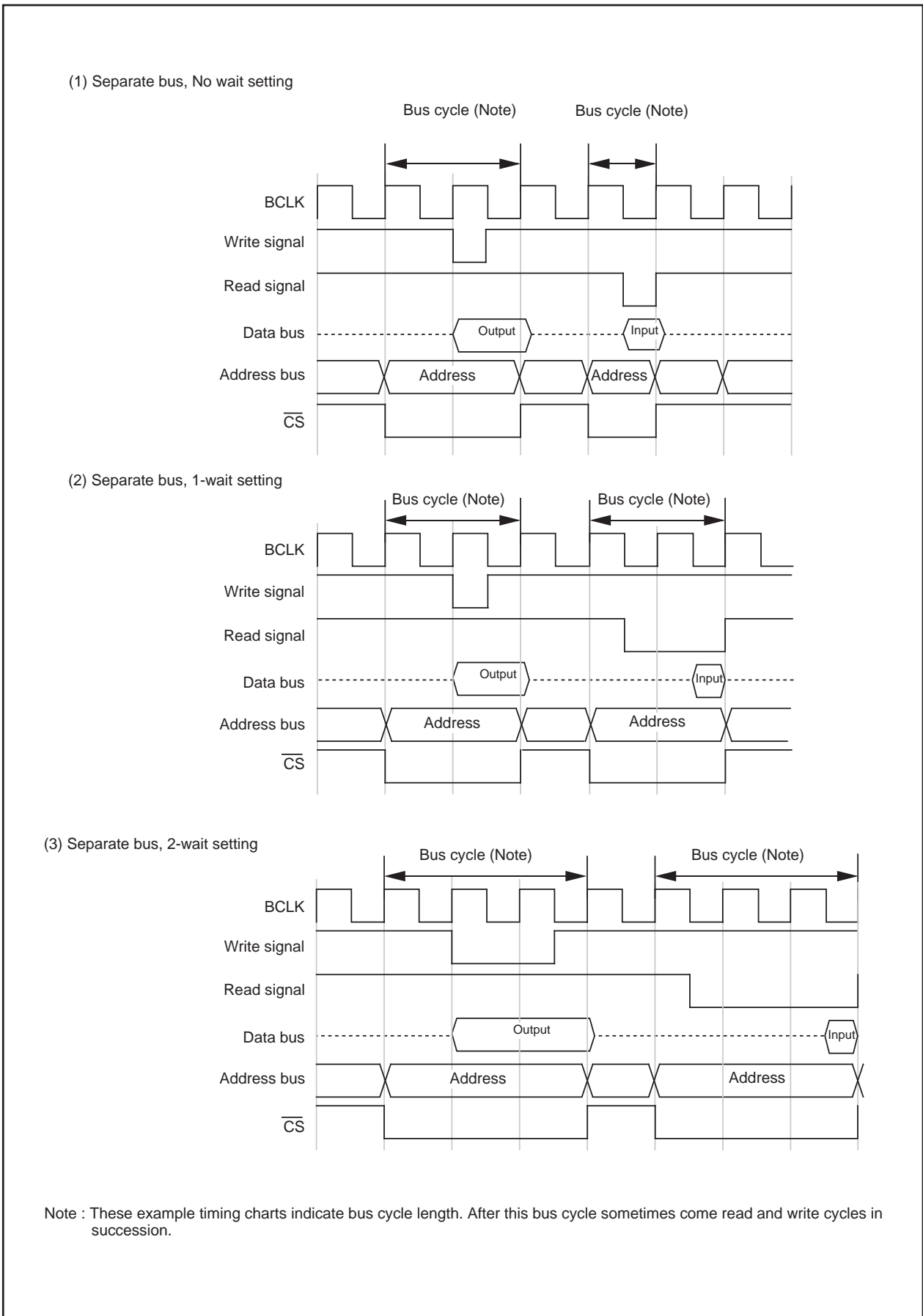


Figure 4.10. Typical Bus Timings Using Software Wait (1)

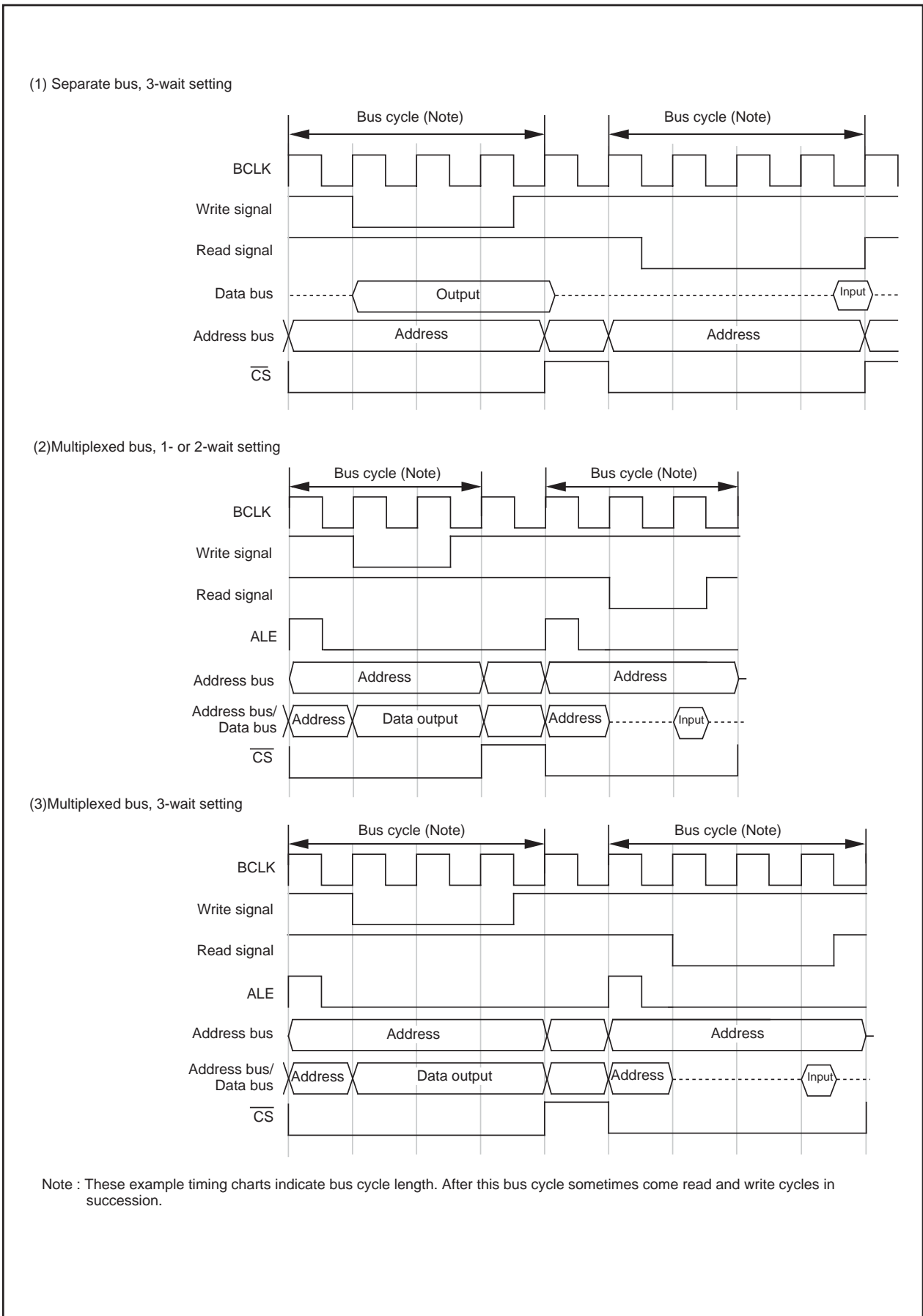


Figure 4.11. Typical Bus Timings Using Software Wait (2)

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

Table 4.11. 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 ₂	1 Mbytes (no expansion)
4 Mbytes mode	11 ₂	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 4.13 to 4.14 show the memory mapping and CS area in 1 Mbyte mode.

(2) 4 Mbyte Mode

In this mode, the memory space is 4 Mbytes. Figure 4.12 shows the DBR register. The BSR2 to BSR0 bits in the DBR register 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 40000₁₆.

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

Addresses 04000₁₆ to 3FFFF₁₆, C0000₁₆ to FFFFF₁₆

- 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 3FFFF₁₆)

Addresses 40000₁₆ to BFFFF₁₆

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

Figures 4.15 to 4.16 show the memory mapping and \overline{CS}_i 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.

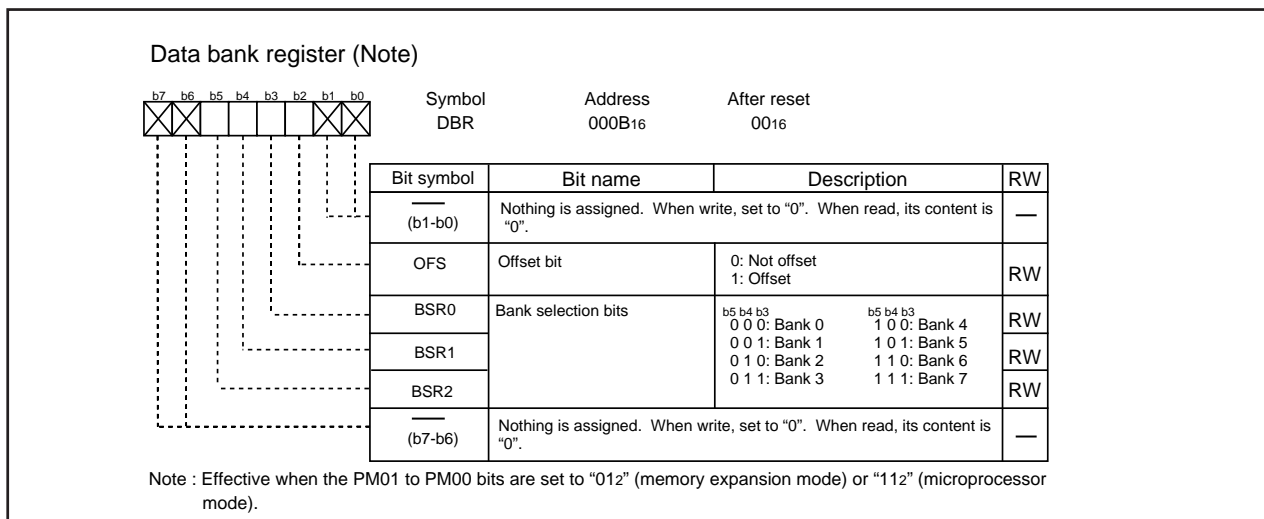


Figure 4.12. DBR Register

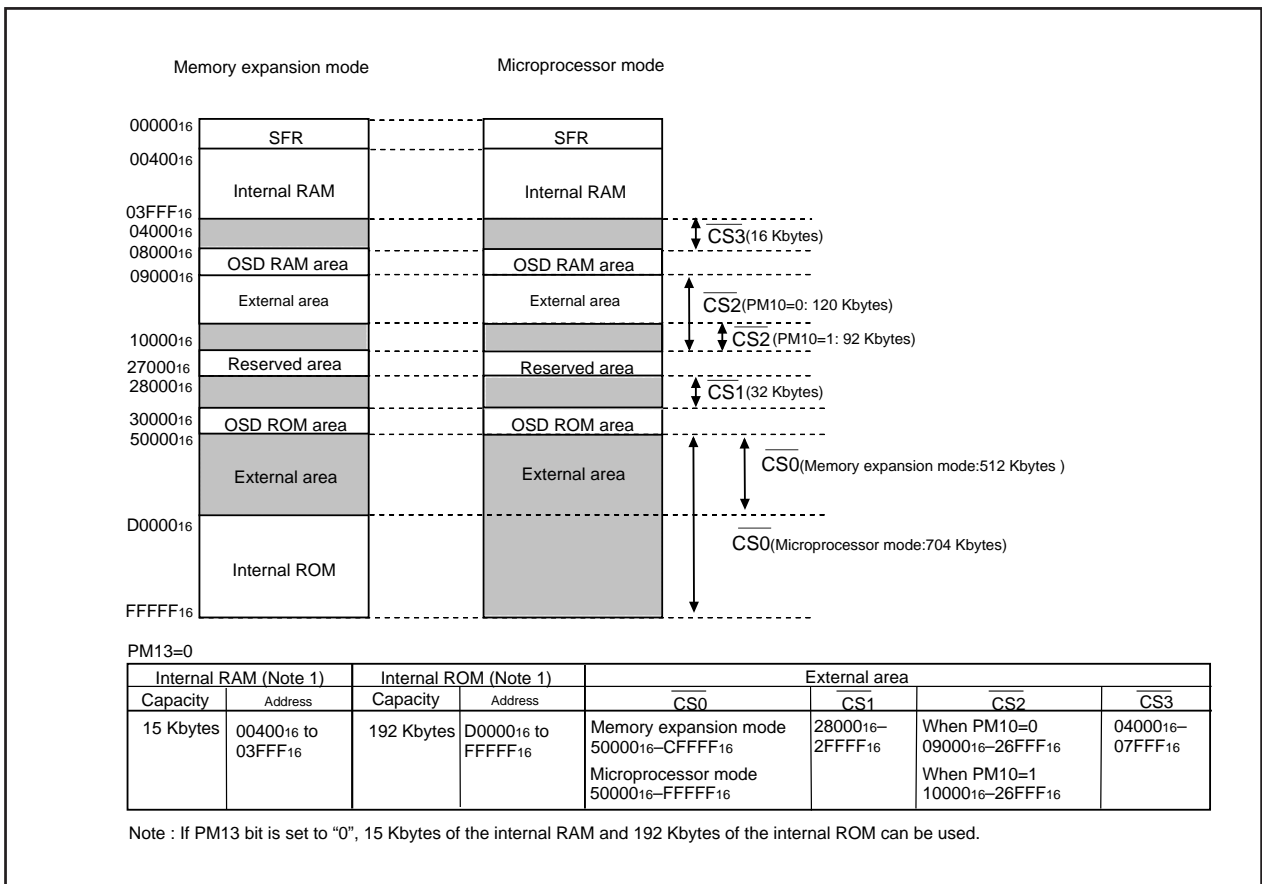


Figure 4.13. Memory Mapping and CS Area in 1 Mbyte Mode (PM13=0)

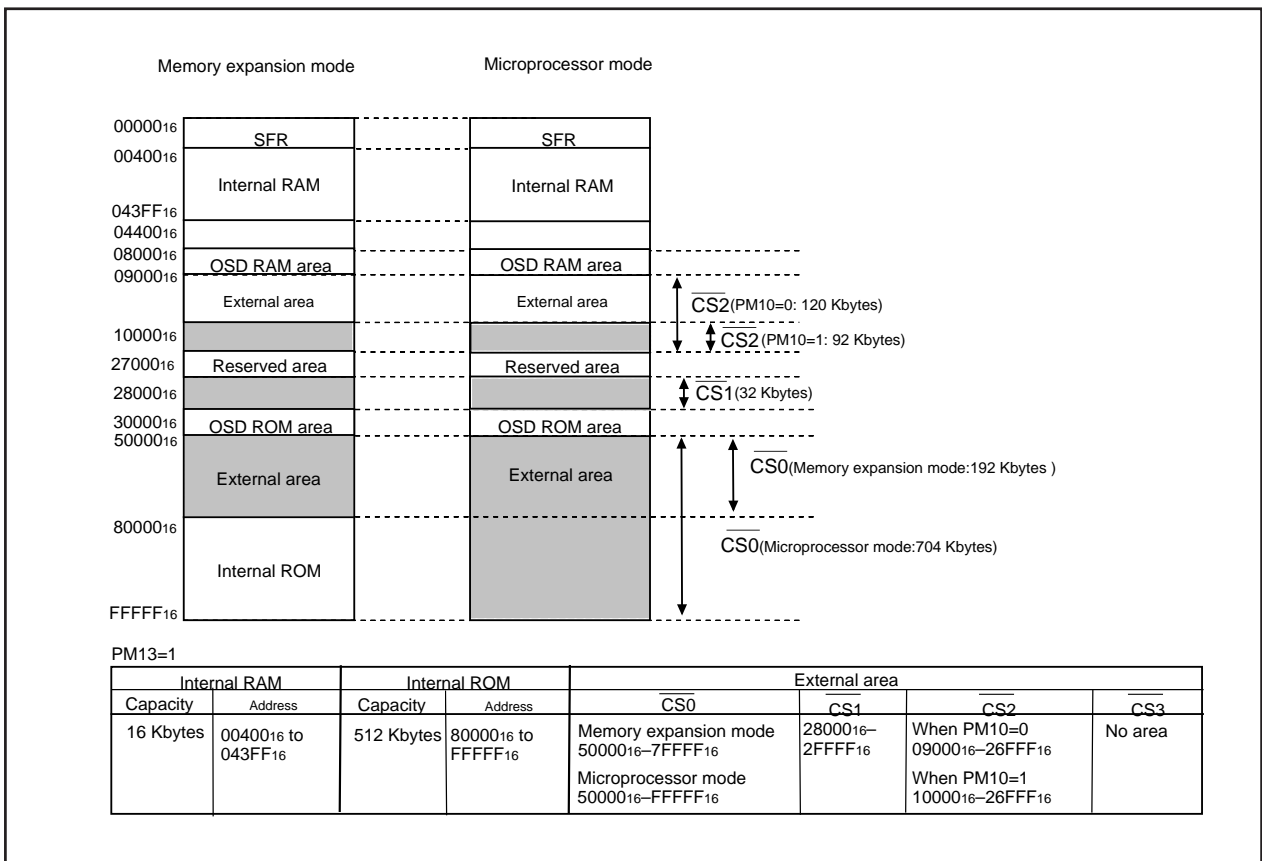


Figure 4.14. Memory Mapping and CS Area in 1 Mbyte Mode (PM13=1)

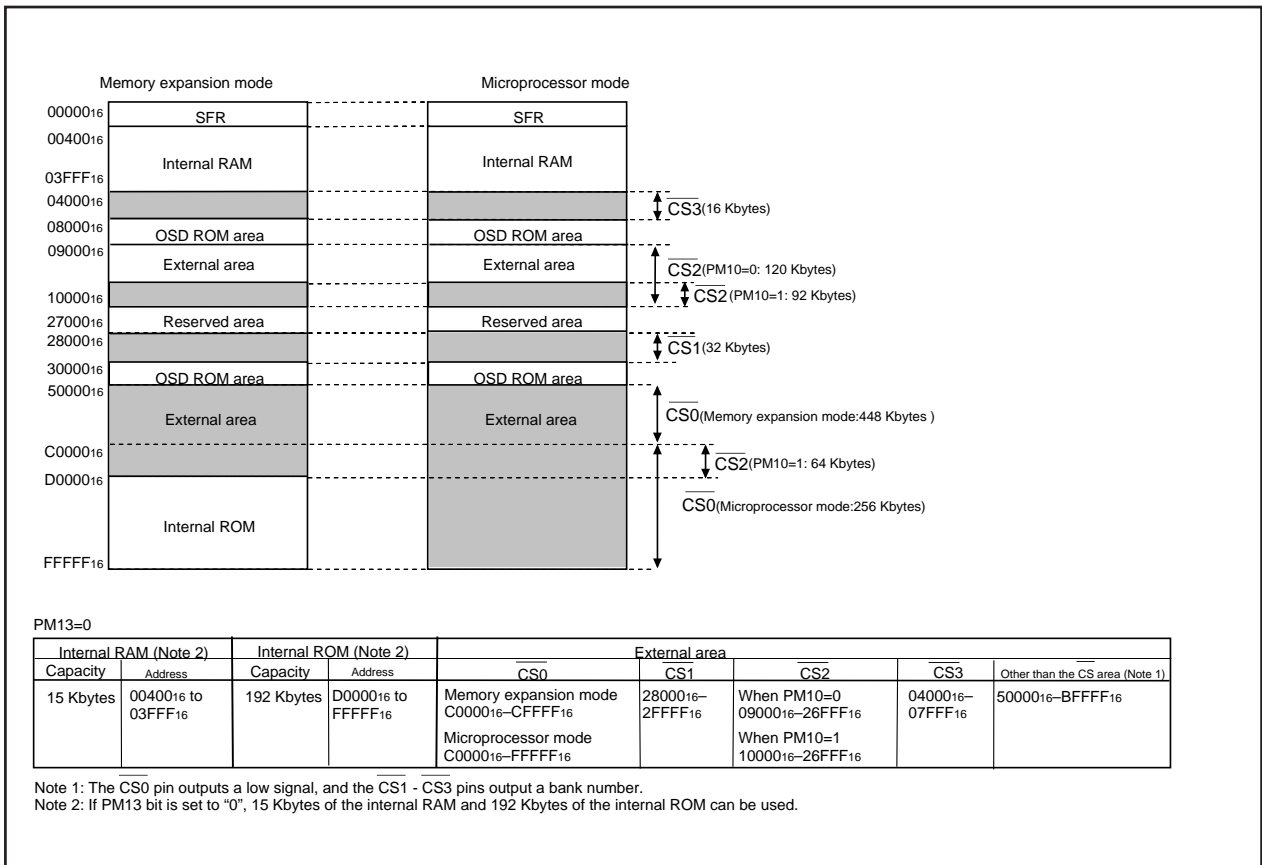


Figure 4.15. Memory Mapping and CS Area in 4 Mbyte Mode (PM13=0)

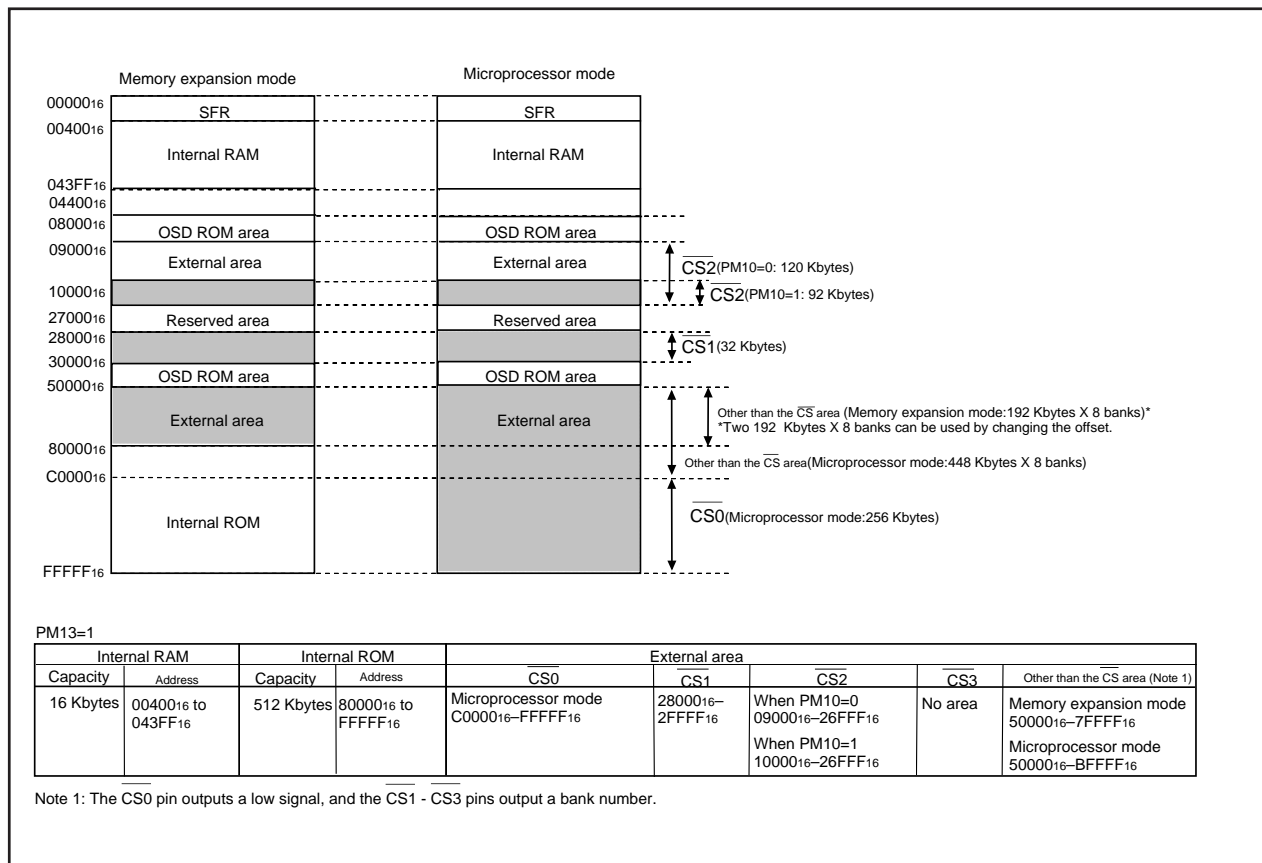


Figure 4.16. Memory Mapping and CS Area in 4 Mbyte Mode (PM13=1)

Figure 4.17 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 4 Mbyte ROM 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 4.18 to 4.20 show the relationship of addresses between the 4-Mbyte ROM and the microcomputer for the case of a connection example in Figure 4.17.

In microprocessor mode, or in memory expansion mode where the PM13 bit in the PM1 register is "0", banks are located every 512 Kbytes. Setting the OFS bit in the DBR register to "1"(offset) 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 $S2$ and $\overline{S1}$, respectively. If the SRAM does not have the input pins to accept "H" active and "L" active chip select signals($\overline{S1}$, $S2$), $\overline{CS0}$ and $\overline{CS2}$ should be decoded external to the chip.

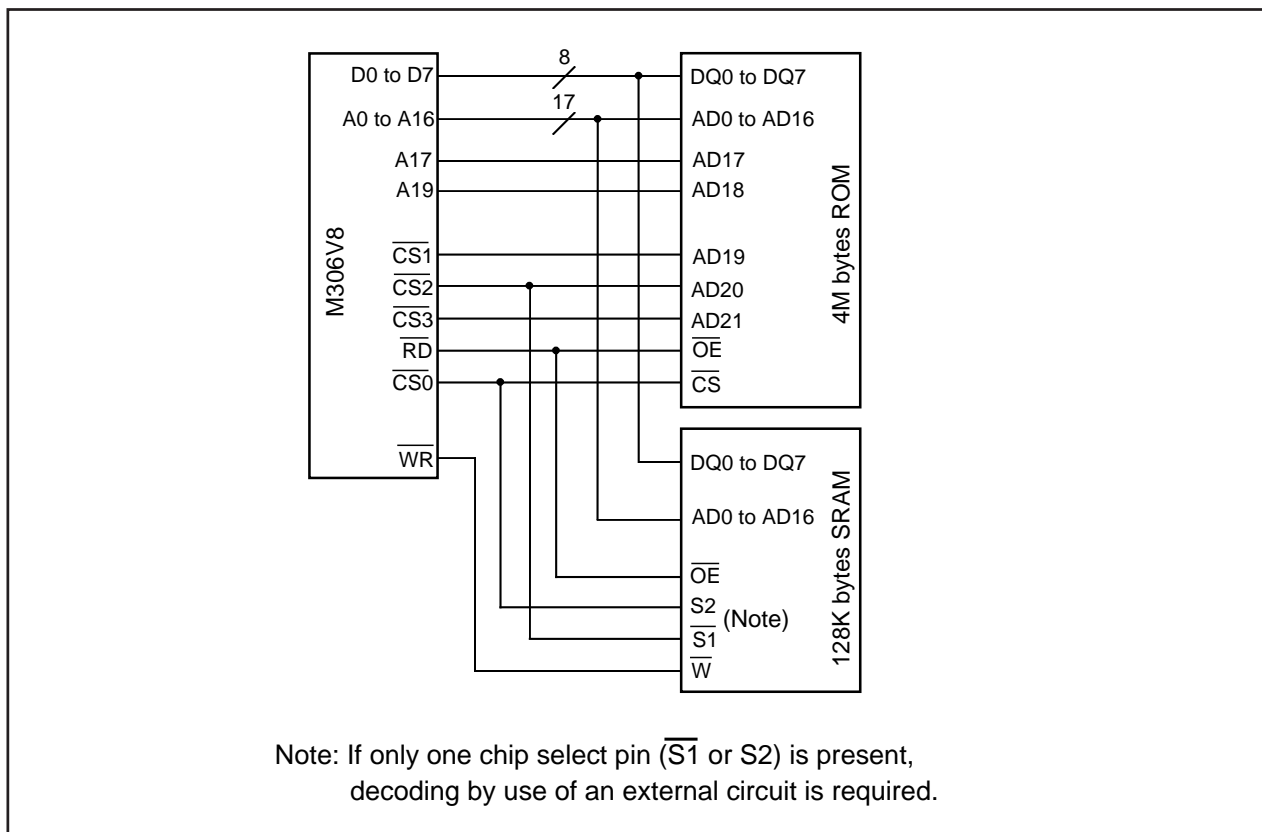


Figure 4.17. External Memory Connect Example in 4M Byte Mode

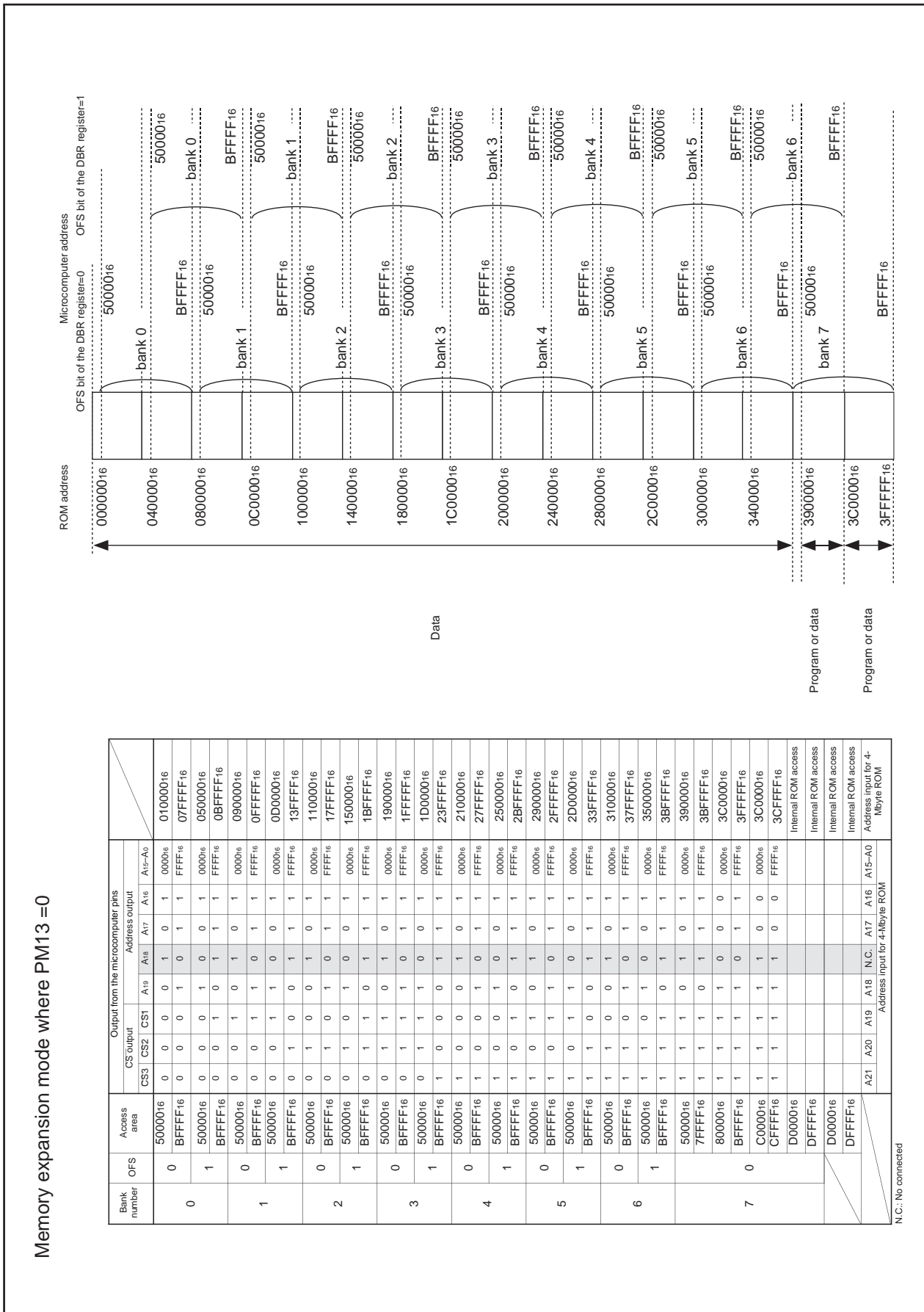


Figure 4.18. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (1)

Memory expansion mode where PM13 = 1

Bank number	OFS	Access area	Output from the microcomputer pins										Address input for 4-Mbyte ROM
			CS output			Address output							
			CS3	CS2	CS1	A18	A18	A17	A16	A15-A0			
0	0	50000 ₁₆	0	0	0	0	1	0	1	0000 ₁₆	010000 ₁₆		
	1	7FFF ₁₆	0	0	0	1	1	1	1	FFFF ₁₆	03FFFF ₁₆		
1	0	50000 ₁₆	0	0	0	1	0	0	1	0000 ₁₆	050000 ₁₆		
	1	7FFF ₁₆	0	0	0	1	0	1	1	FFFF ₁₆	07FFFF ₁₆		
2	0	50000 ₁₆	0	0	1	0	1	0	1	0000 ₁₆	090000 ₁₆		
	1	7FFF ₁₆	0	0	1	0	1	1	1	FFFF ₁₆	0BFFFF ₁₆		
3	0	50000 ₁₆	0	0	1	1	0	0	1	0000 ₁₆	0D0000 ₁₆		
	1	7FFF ₁₆	0	0	1	1	0	1	1	FFFF ₁₆	0FFFFF ₁₆		
4	0	50000 ₁₆	0	1	0	0	1	0	1	0000 ₁₆	110000 ₁₆		
	1	7FFF ₁₆	0	1	0	0	1	1	1	FFFF ₁₆	13FFFF ₁₆		
5	0	50000 ₁₆	0	1	1	0	0	1	1	0000 ₁₆	150000 ₁₆		
	1	7FFF ₁₆	0	1	1	0	1	1	1	FFFF ₁₆	17FFFF ₁₆		
6	0	50000 ₁₆	0	1	1	0	1	0	1	0000 ₁₆	190000 ₁₆		
	1	7FFF ₁₆	0	1	1	0	1	1	1	FFFF ₁₆	1BFFFF ₁₆		
7	0	50000 ₁₆	0	1	1	1	0	0	1	0000 ₁₆	1D0000 ₁₆		
	1	7FFF ₁₆	0	1	1	1	0	1	1	FFFF ₁₆	1FFFFF ₁₆		
7	0	50000 ₁₆	1	0	0	0	1	0	1	0000 ₁₆	210000 ₁₆		
	1	7FFF ₁₆	1	0	0	0	1	1	1	FFFF ₁₆	23FFFF ₁₆		
7	0	50000 ₁₆	1	0	0	1	0	0	1	0000 ₁₆	250000 ₁₆		
	1	7FFF ₁₆	1	0	0	1	0	1	1	FFFF ₁₆	27FFFF ₁₆		
7	0	50000 ₁₆	1	0	1	0	1	0	1	0000 ₁₆	290000 ₁₆		
	1	7FFF ₁₆	1	0	1	0	1	1	1	FFFF ₁₆	2BFFFF ₁₆		
7	0	50000 ₁₆	1	0	1	1	0	0	1	0000 ₁₆	2D0000 ₁₆		
	1	7FFF ₁₆	1	0	1	1	0	1	1	FFFF ₁₆	2FFFFF ₁₆		
7	0	50000 ₁₆	1	1	0	0	1	0	1	0000 ₁₆	310000 ₁₆		
	1	7FFF ₁₆	1	1	0	0	1	1	1	FFFF ₁₆	33FFFF ₁₆		
7	0	50000 ₁₆	1	1	0	1	0	0	1	0000 ₁₆	350000 ₁₆		
	1	7FFF ₁₆	1	1	0	1	0	1	1	FFFF ₁₆	37FFFF ₁₆		
7	0	50000 ₁₆	1	1	1	0	1	0	1	0000 ₁₆	390000 ₁₆		
	1	7FFF ₁₆	1	1	1	0	1	1	1	FFFF ₁₆	3BFFFF ₁₆		
		80000 ₁₆									Internal ROM access		
		FFFF ₁₆									Internal ROM access		
		50000 ₁₆	1	1	1	1	0	0	1	0000 ₁₆	3D0000 ₁₆		
		7FFF ₁₆	1	1	1	1	0	1	1	FFFF ₁₆	3FFFFF ₁₆		
		80000 ₁₆									Internal ROM access		
		FFFF ₁₆									Internal ROM access		

N.C.: No connected

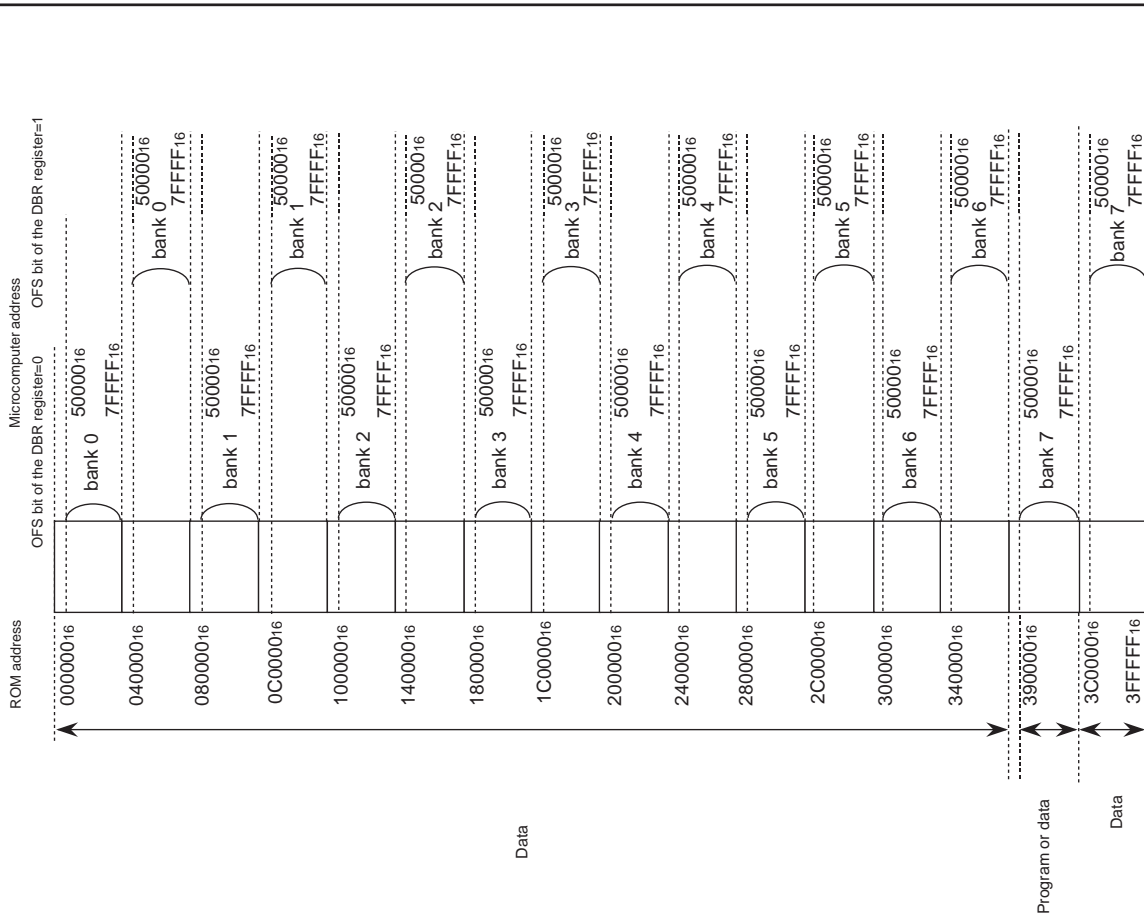


Figure 4.19. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (2)

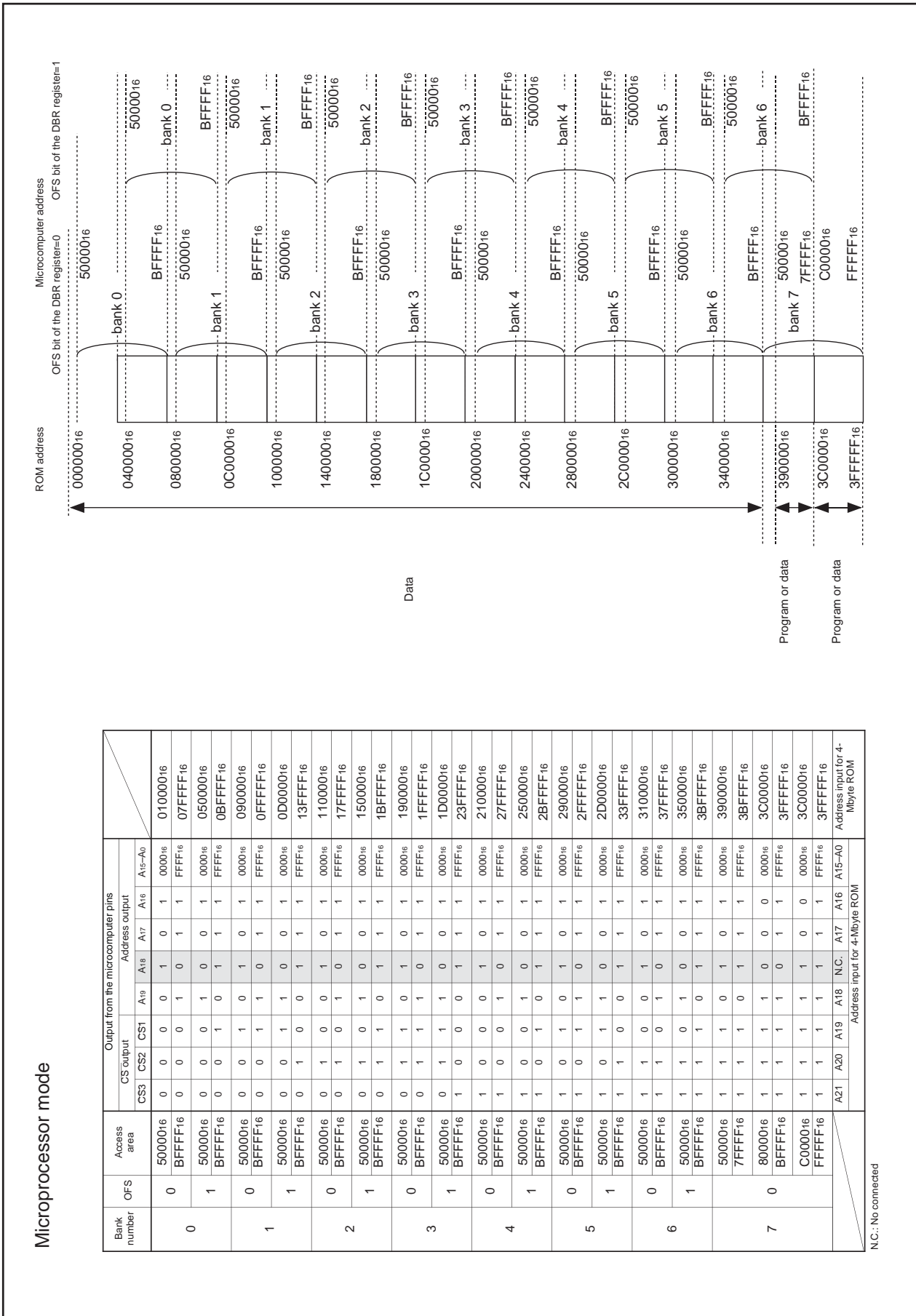


Figure 4.20. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (3)

Clock Generation Circuit

The clock generation circuit contains two oscillator circuits as follows:

- (1) Main clock oscillation circuit
- (2) Sub clock oscillation circuit

Table 5.1 lists the clock generation circuit specifications. Figure 5.1 shows the clock generation circuit. Figures 5.2 to 5.6 show the clock-related registers.

Table 5.1. Clock Generation Circuit Specifications

Item	Main clock oscillation circuit	Sub clock oscillation circuit
Use of clock	<ul style="list-style-type: none"> • CPU clock source • Peripheral function clock source 	<ul style="list-style-type: none"> • CPU clock source • Timer A, B's clock source
Clock frequency	16 MHz	32.768 kHz
Usable oscillator	<ul style="list-style-type: none"> • Ceramic oscillator • Crystal oscillator 	<ul style="list-style-type: none"> • Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XCIN, XCOU
Oscillation stop, restart function	Presence	Presence
Oscillator status after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

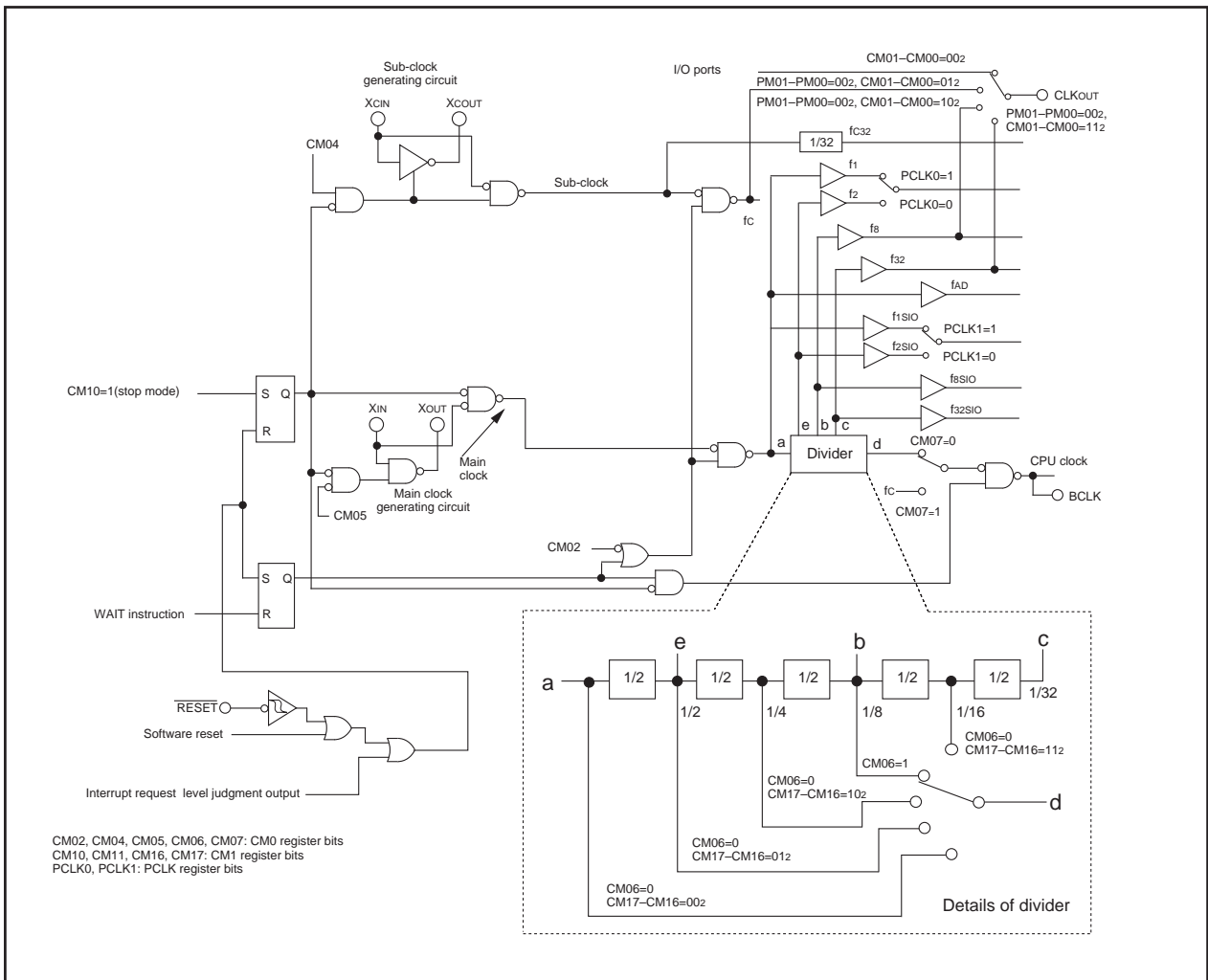


Figure 5.1. Clock Generation Circuit

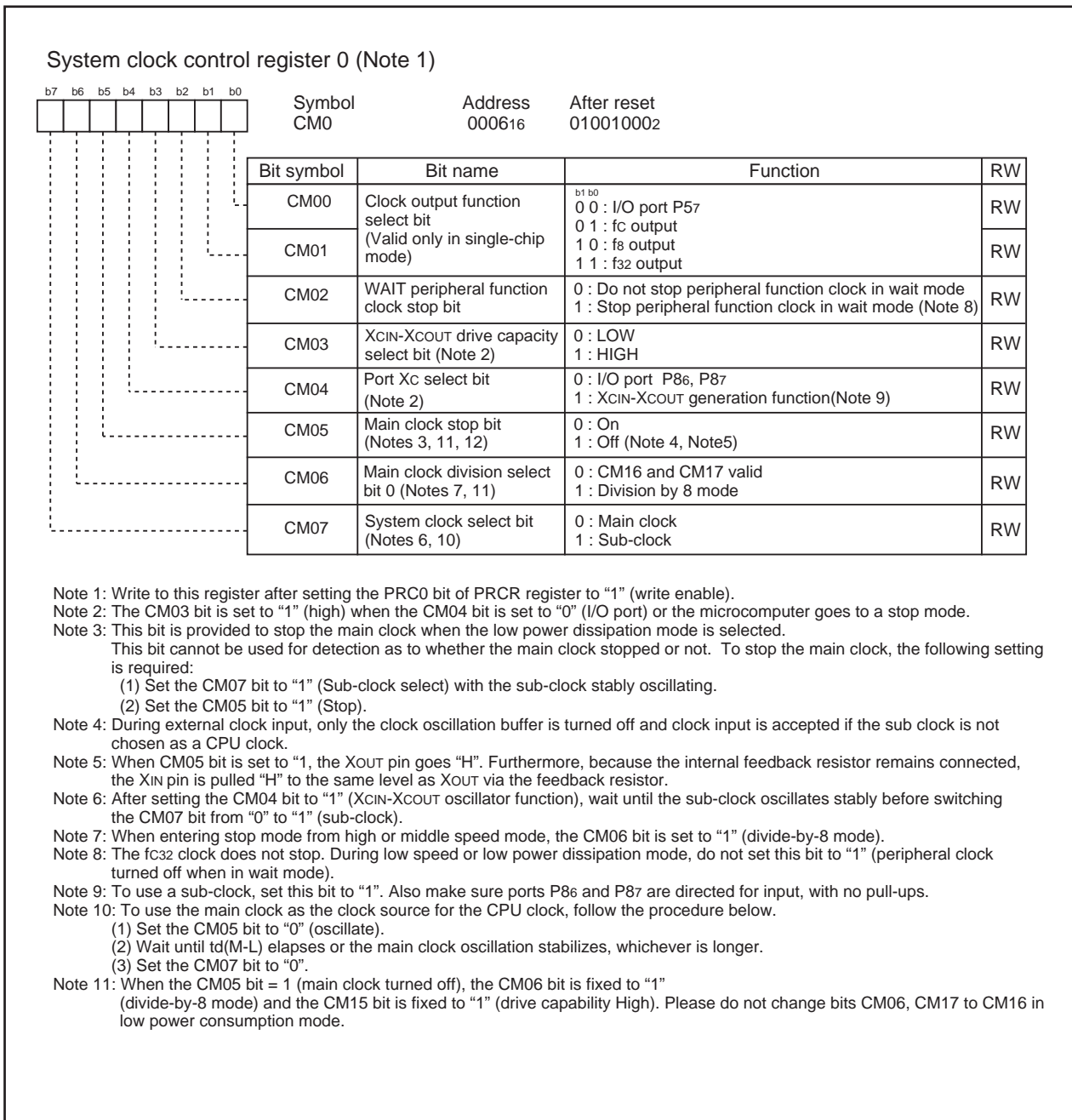


Figure 5.2. CM0 Register

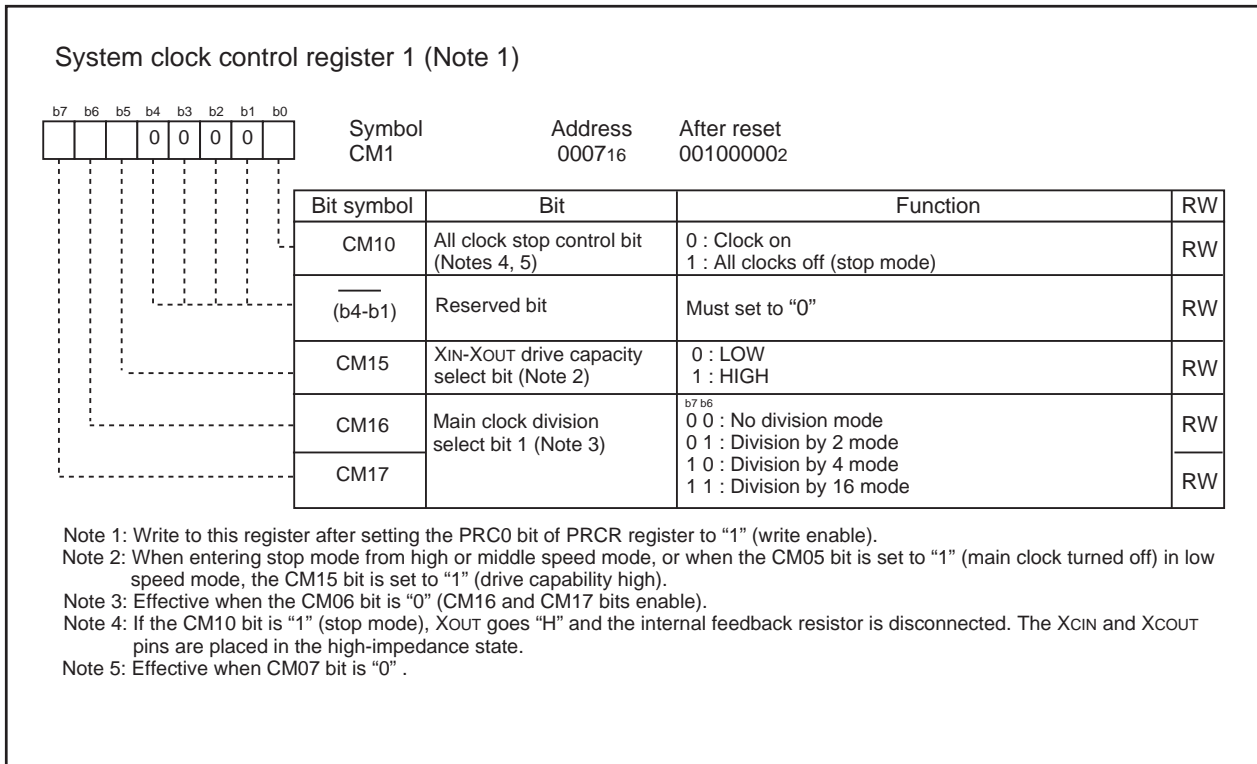


Figure 5.3. CM1 Register

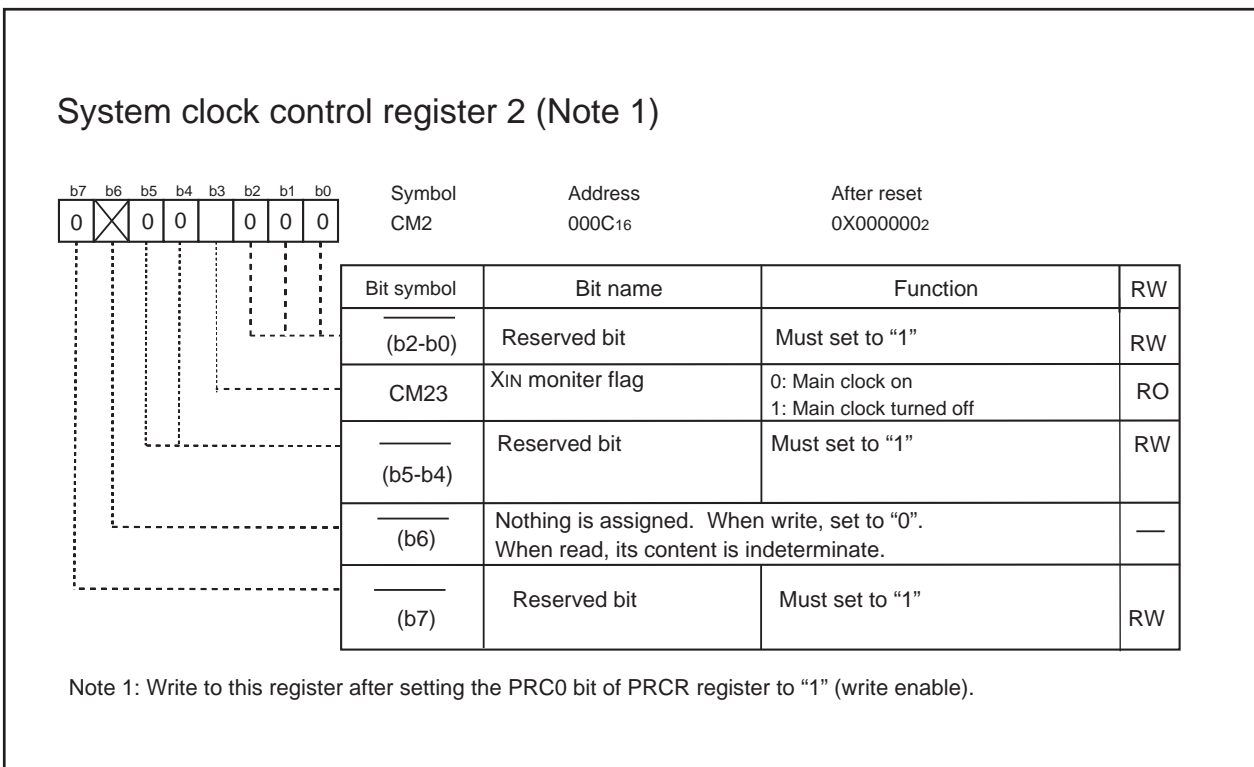


Figure 5.4. CM2 Register

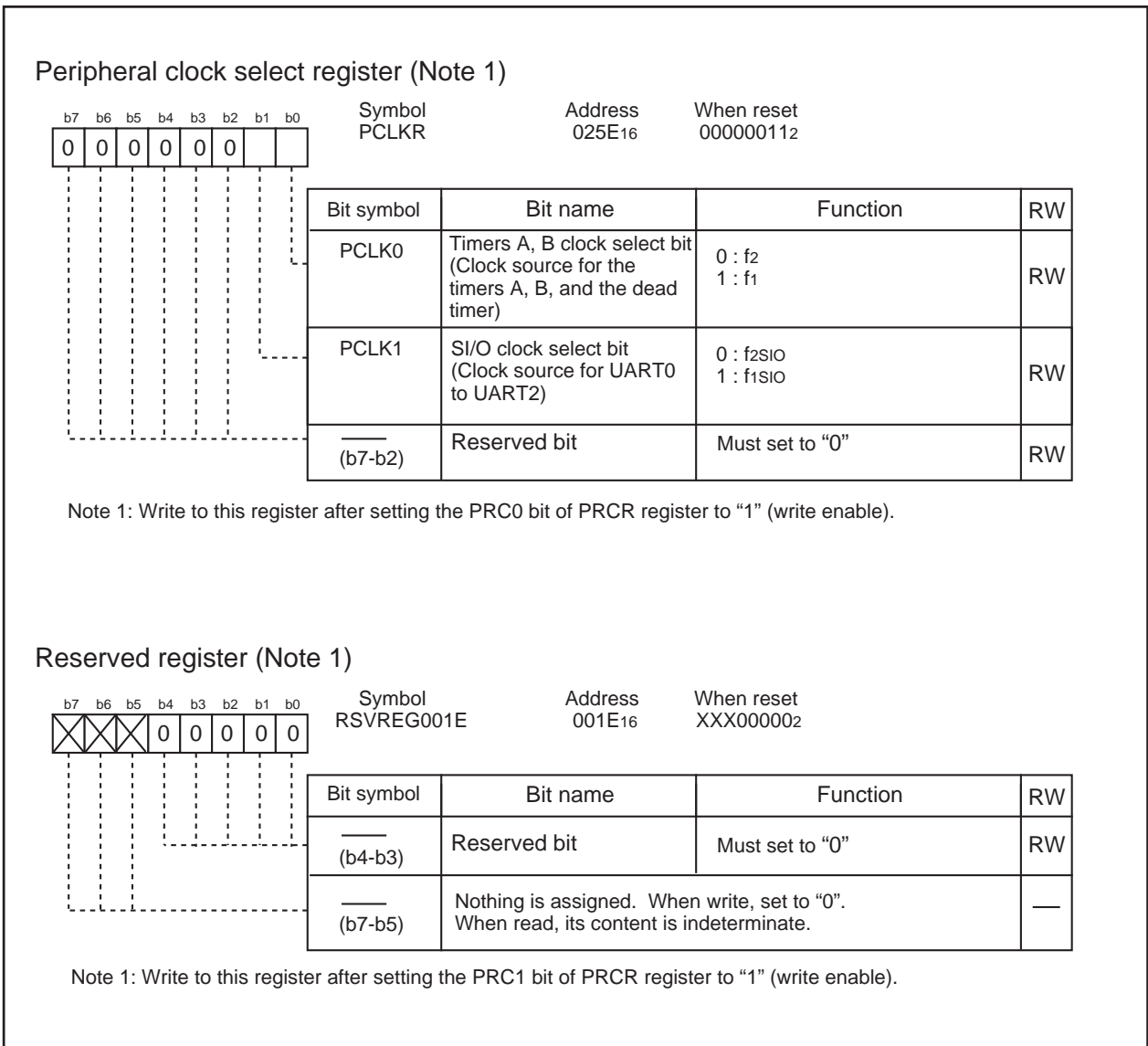


Figure 5.5. PCLKR Register and PM2 Register

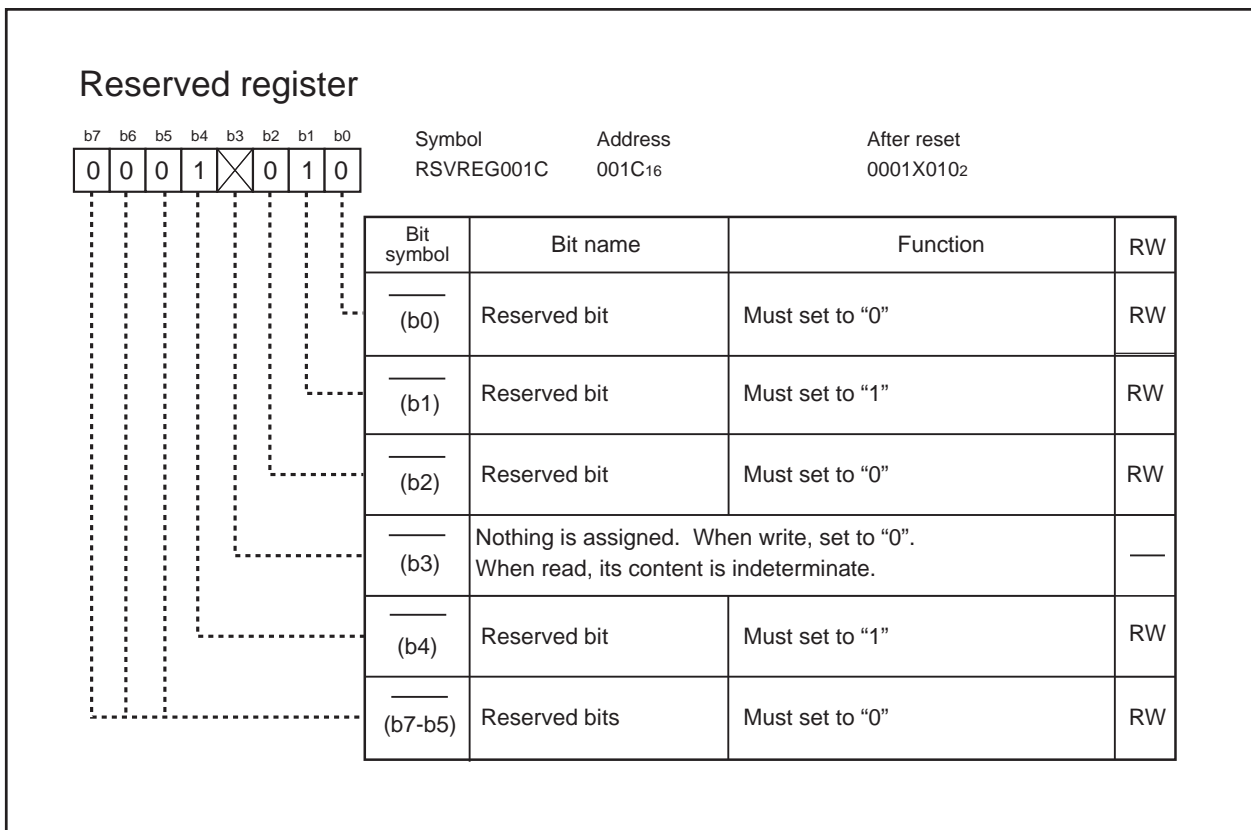


Figure 5.6. PLC0 Register

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 5.7 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. In this case, XOUT goes "H". Furthermore, because the internal feedback resistor remains on, XIN is pulled "H" to XOUT via the feedback resistor. Note that if an externally generated clock is fed into the XIN pin, the main clock cannot be turned off by setting the CM05 bit to "1", unless the sub clock is chosen as a CPU clock. If necessary, use an external circuit to turn off the clock.

During stop mode, all clocks including the main clock are turned off. Refer to "power control".

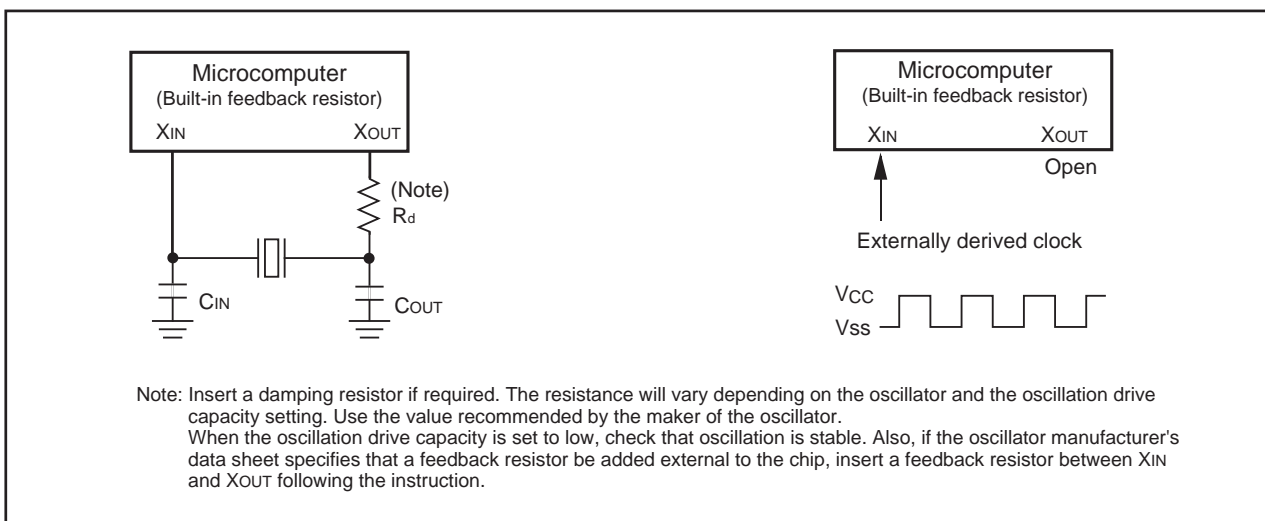


Figure 5.7. Examples of Main Clock Connection Circuit

(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 XCOU 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 5.8 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.

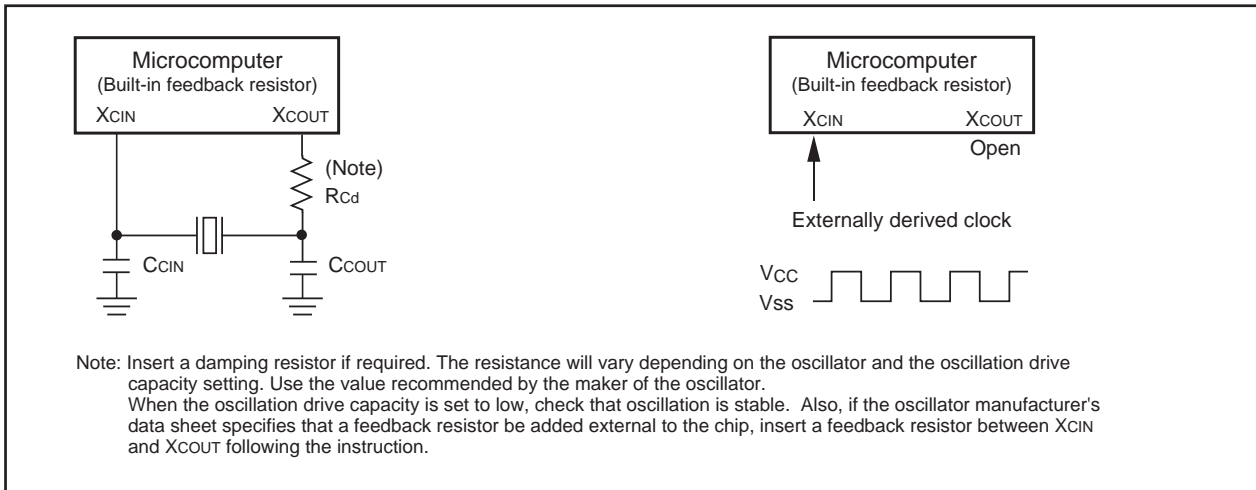


Figure 5.8. Examples of Sub Clock Connection Circuit

OSD Oscillation Circuit

The OSD clock oscillation circuit can be chosen to be an external oscillator circuit comprised of an LC oscillator or a ceramic resonator (or a quartz-crystal oscillator) connected between the OSC1 and OSC2 pins, or an internal oscillator circuit with a filter connected to the OSC1 pin. Which of LC oscillator or a ceramic resonator (or a quartz-crystal oscillator) is selected by setting bits 0, 1 and 2 of the clock control register (address 020516) and bit 1 of the extended register 1C (address 02DC16).

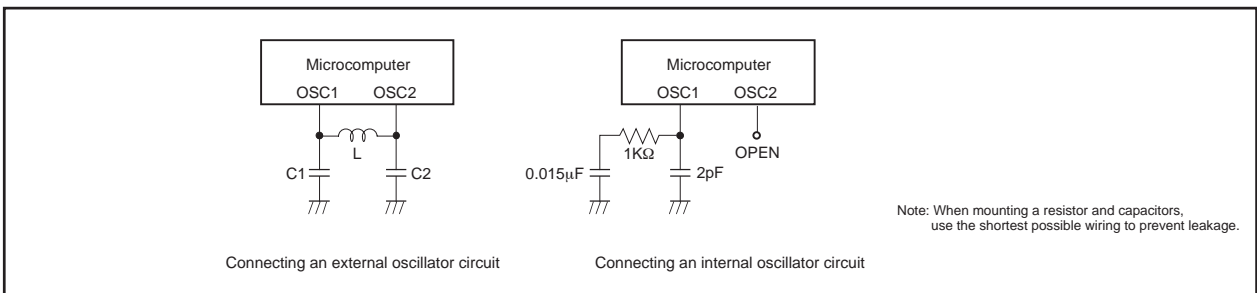


Figure 5.9. OSD clock connection example

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 or sub clock.

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

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, 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(f1, f2, f8, f32, f1SIO, f2SIO, f8SIO, f32SIO, fAD, fC32)

These are operating clocks for the peripheral functions.

Of these, fi (i = 1, 2, 8, 32) and fiSIO are derived from the main clock. The clock fi is used for timers A and B, and fiSIO is used for serial I/O. The f8 and f32 clocks can be output from the CLKOUT pin.

The fAD clock is produced from the main clock, and is used for the A/D converter.

When the WAIT instruction is executed after setting the CM02 bit of CM0 register to "1" (peripheral function clock turned off during wait mode), or when the microcomputer is in low power dissipation mode, the fi, fiSIO and fAD clocks are turned off.

The fC32 clock is produced from the sub clock, and is used for timers A and B. This clock can be used when the sub clock is on.

Clock Output Function

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

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 four modes.

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

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

- **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.

- **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 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.

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.

Table 5.2. Setting Clock Related Bit and Modes

Modes		CM1 register		CM0 register		
		CM17, CM16	CM07	CM06	CM05	CM04
High-speed mode		002	0	0	0	—
Medium-speed mode	divided by 2	012	0	0	0	—
	divided by 4	102	0	0	0	—
	divided by 8	—	0	1	0	—
	divided by 16	112	0	0	0	—
Low-speed mode		—	1	—	0	1
Low power dissipation mode		—	1	1(Note 1)	1(Note 1)	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.

(2) Wait Mode

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

- **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.

- **Pin Status During Wait Mode**

Table 5.3 lists pin status during wait mode

- **Exiting Wait Mode**

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

If the microcomputer is to be moved out of exit wait mode by a hardware reset, set the peripheral function interrupt priority ILVL2 to ILVL0 bits to “0002” (interrupts disabled) before executing the WAIT instruction.

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

Table 5.3. Pin Status During Wait Mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
A ₀ to A ₁₉ , D ₀ to D ₁₅ , $\overline{CS0}$ to $\overline{CS3}$, BHE		Retains status before wait mode	
RD, WR, WRL, WRH		"H"	
HLDA, BCLK		"H"	
ALE		"L"	
I/O ports		Retains status before wait mode	Retains status before wait mode
CLKOUT	When fc selected		Does not stop
	When f8, f32 selected		Does not stop when the CM02 bit is "0". When the CM02 bit is "1", the status immediately prior to entering wait mode is maintained.

Table 5.4. Interrupts to Exit Wait Mode

Interrupt	CM02=0	CM02=1
Serial I/O interrupt	Can be used when operating with internal or external clock	Can be used when operating with external clock
key input interrupt	Can be used	Can be used
A/D conversion interrupt	Can be used in one-shot mode 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 fc32
INT interrupt	Can be used	Can be used

Table 5.4 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.

(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.

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

- Key interrupt
- $\overline{\text{INT}}$ interrupt
- 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 CM1 register is set to “1” (main clock oscillator circuit drive capability high).

• Pin Status in Stop Mode

Table 5.5 lists pin status during stop mode

• Exiting Stop Mode

The microcomputer is moved out of stop mode by a hardware reset, or peripheral function interrupt. If the microcomputer is to be moved out of stop mode by a hardware reset or, 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 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

Table 5.5. Pin Status in Stop Mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
A ₀ to A ₁₉ , D ₀ to D ₁₅ , $\overline{\text{CS}}_0$ to $\overline{\text{CS}}_3$, $\overline{\text{BHE}}$		Retains status before stop mode	
RD, WR, WRL, WRH		"H"	
$\overline{\text{HLDA}}$, BCLK		"H"	
ALE		undefined	
I/O ports		Retains status before stop mode	Retains status before stop mode
CLKOUT	When fc selected		"H"
	When f8, f32 selected		Retains status before stop mode

Figure 5.10 shows the state transition from normal operation mode to stop mode and wait mode. Figure 5.11 shows the state transition in normal operation mode.

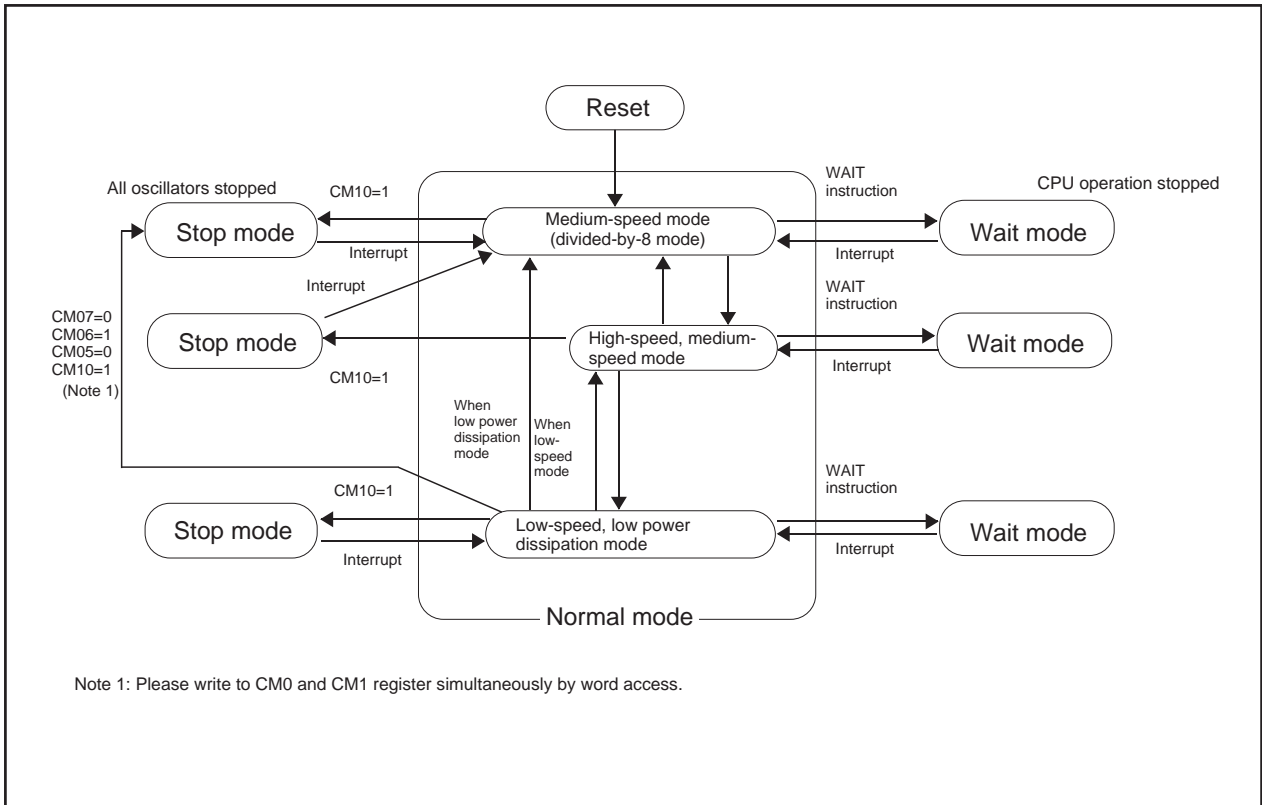


Figure 5.10. State Transition to Stop Mode and Wait Mode

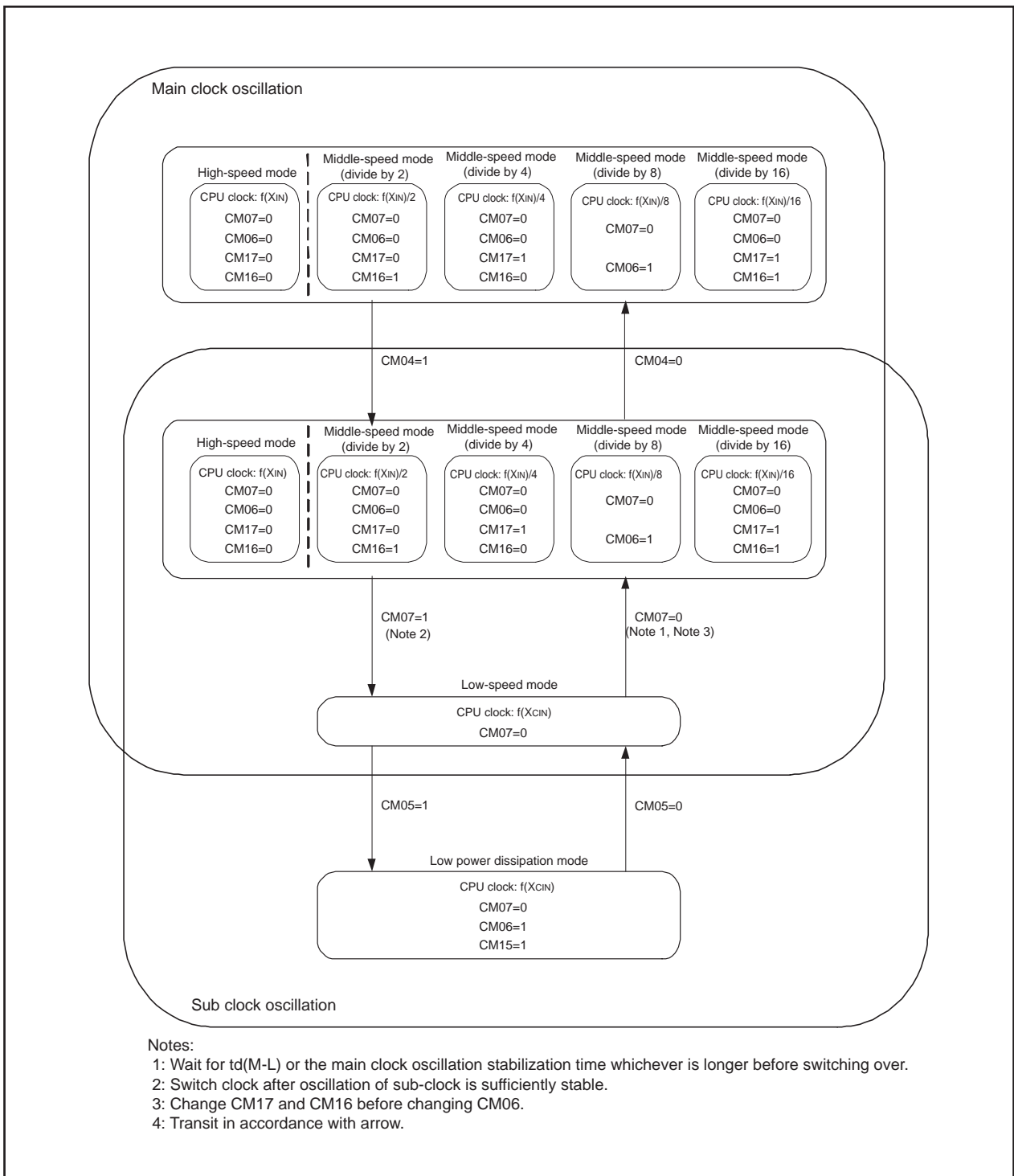


Figure 5.11. State Transition in Normal Mode

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 6.1 shows the PRCR register. The following lists the registers protected by the PRCR register.

- Registers protected by PRC0 bit: CM0, CM1, CM2 and PCLKR registers, reserved register address 001C₁₆.
- Registers protected by PRC1 bit: PM0 and PM1 registers, reserved registers address 001E₁₆, 039E₁₆, 0348₁₆, and 0349₁₆
- Registers protected by PRC2 bit: PD9 register, reserved registers address 0362₁₆ and 0366₁₆

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.

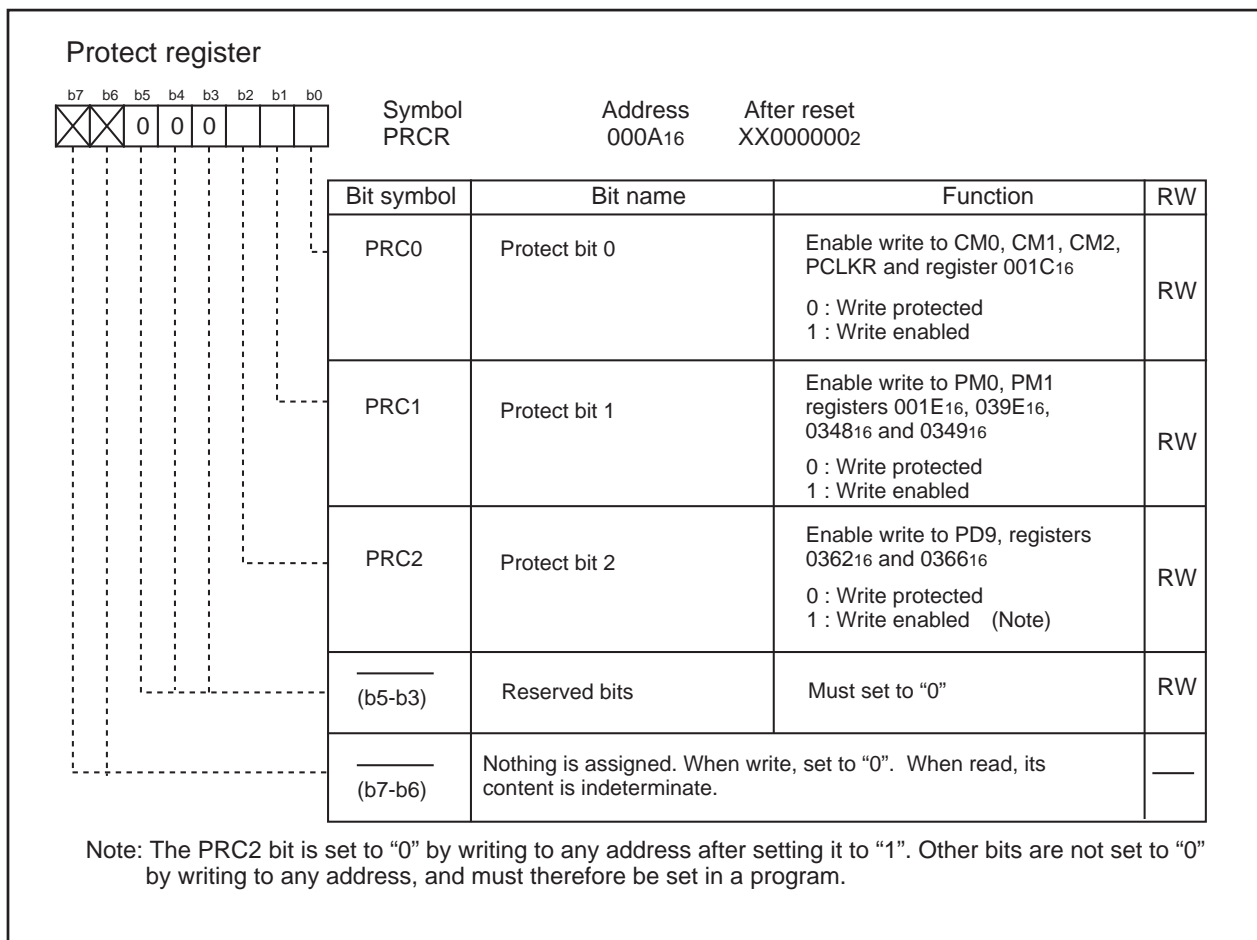


Figure 6.1. PRCR Register

Interrupts

Type of Interrupts

Figure 7.1 shows types of interrupts.

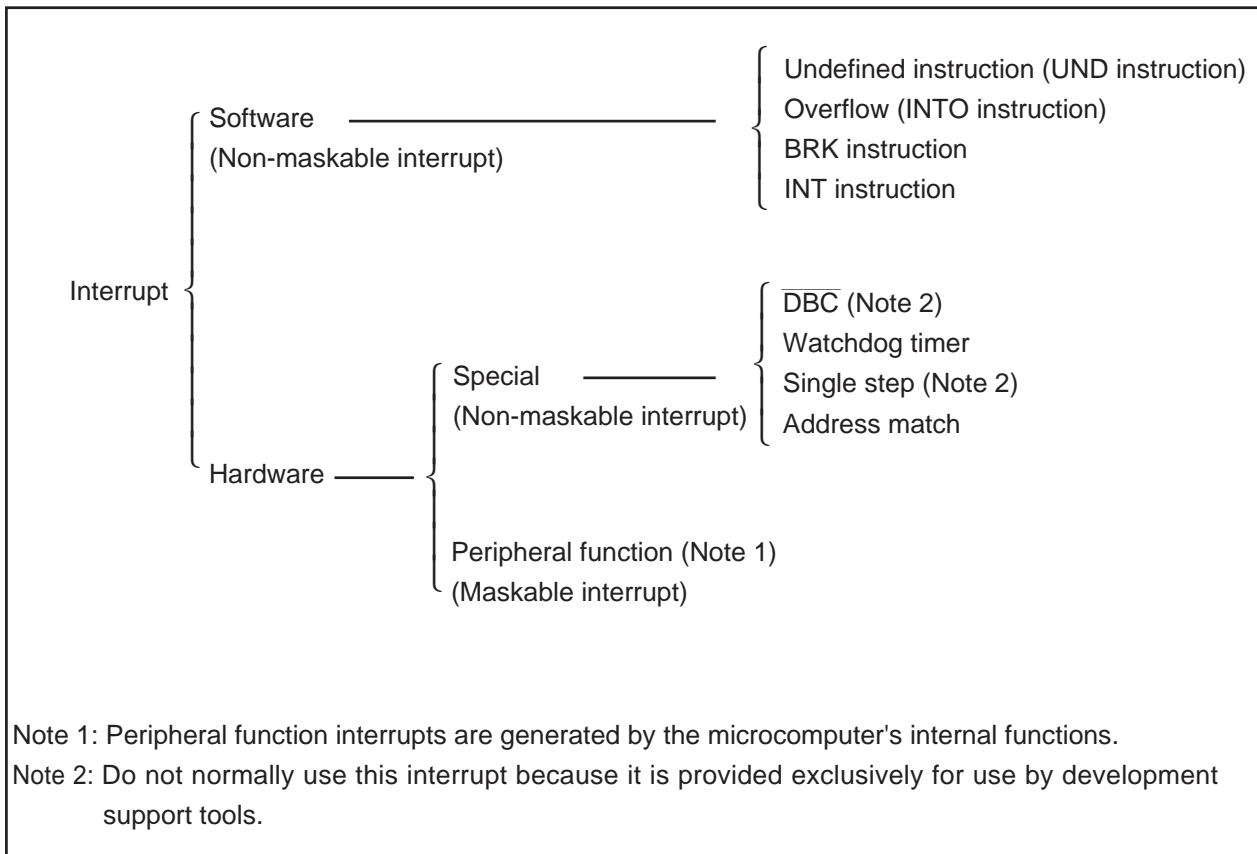


Figure 7.1. Interrupts

- 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.

- **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".

- **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 7.2. Relocatable Vector Tables". For details about the peripheral functions, refer to the description of each peripheral function in this manual.

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 7.2 shows the interrupt vector.

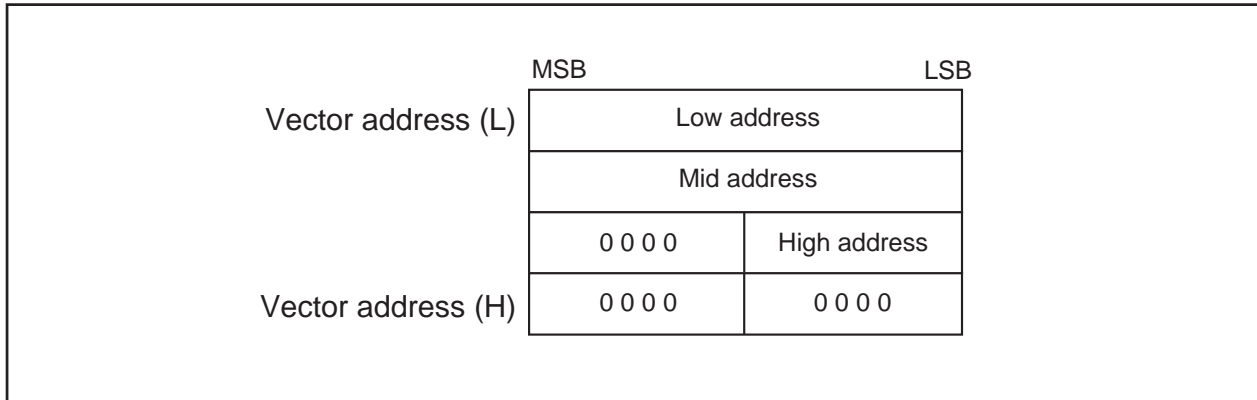


Figure 7.2. Interrupt Vector

• Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDC₁₆ to FFFFF₁₆. Table 7.1 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 7.1. Fixed Vector Tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks	Reference
Undefined instruction	FFFD _{C16} to FFFD _{F16}	Interrupt on UND instruction	M16C/60, M16C/20 serise software maual
Overflow	FFFE ₀₁₆ to FFFE ₃₁₆	Interrupt on INTO instruction	
BRK instruction	FFFE ₄₁₆ to FFFE ₇₁₆	If the contents of address FFFE ₇₁₆ is FF ₁₆ , program execution starts from the address shown by the vector in the relocatable vector table.	
Address match	FFFE ₈₁₆ to FFFEB ₁₆		Address match interrupt
Single step (Note)	FFFE _{C16} to FFFE _{F16}		
Watchdog timer	FFFF ₀₁₆ to FFFF ₃₁₆		Watchdog timer
DBC (Note)	FFFF ₄₁₆ to FFFF ₇₁₆		
Reset	FFFF _{C16} to FFFF _{F16}		Reset

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

• Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. Table 7.2 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 7.2. Relocatable Vector Tables

Interrupt source	Vector address (Note 1) Address (L) to address (H)	Software interrupt number	Reference
BRK instruction (Note 3)	+0 to +3 (0000 ₁₆ to 0003 ₁₆)	0	M16C/60, M16C/20 series software manual
———— (Reserved)		1 to 3	
$\overline{\text{INT3}}$	+16 to +19 (0010 ₁₆ to 0013 ₁₆)	4	
Timer B5/OSD1	+20 to +23 (0014 ₁₆ to 0017 ₁₆)	5	Timer
Timer B4, UART1 bus collision detect (Note 2)	+24 to +27 (0018 ₁₆ to 001B ₁₆)	6	Timer Serial I/O
Timer B3, UART0 bus collision detect (Note 2)	+28 to +31 (001C ₁₆ to 001F ₁₆)	7	
Data slicer 1	+32 to +35 (0020 ₁₆ to 0023 ₁₆)	8	Data slicer
Data slicer 2	+36 to +39 (0024 ₁₆ to 0027 ₁₆)	9	
UART 2 bus collision detection/I ² C-bus0	+40 to +43 (0028 ₁₆ to 002B ₁₆)	10	Serial I/O/I ² C-bus
DMA0	+44 to +47 (002C ₁₆ to 002F ₁₆)	11	DMAC
DMA1	+48 to +51 (0030 ₁₆ to 0033 ₁₆)	12	
Key input interrupt/VSYNC	+52 to +55 (0034 ₁₆ to 0037 ₁₆)	13	Key input interrupt, OSD
A/D/I ² C-bus1NACK	+56 to +59 (0038 ₁₆ to 003B ₁₆)	14	A/D convertor/I ² C-bus
UART2 transmit	+60 to +63 (003C ₁₆ to 003F ₁₆)	15	Serial I/O
UART2 receive	+64 to +67 (0040 ₁₆ to 0043 ₁₆)	16	
UART0 transmit	+68 to +71 (0044 ₁₆ to 0047 ₁₆)	17	
UART0 receive	+72 to +75 (0048 ₁₆ to 004B ₁₆)	18	
UART1 transmit	+76 to +79 (004C ₁₆ to 004F ₁₆)	19	
UART1 receive	+80 to +83 (0050 ₁₆ to 0053 ₁₆)	20	
Timer A0/I ² C-bus0	+84 to +87 (0054 ₁₆ to 0057 ₁₆)	21	Timer/I ² C-bus
Timer A1/I ² C-bus1	+88 to +91 (0058 ₁₆ to 005B ₁₆)	22	
Timer A2/OSD2	+92 to +95 (005C ₁₆ to 005F ₁₆)	23	
Timer A3/VSYNC	+96 to +99 (0060 ₁₆ to 0063 ₁₆)	24	
Timer A4/I ² C-bus0NACK	+100 to +103 (0064 ₁₆ to 0067 ₁₆)	25	
Timer B0/I ² C-bus1NACK	+104 to +107 (0068 ₁₆ to 006B ₁₆)	26	
Timer B1/I ² C-bus2NACK	+108 to +111 (006C ₁₆ to 006F ₁₆)	27	
Timer B2/I ² C-bus2	+112 to +115 (0070 ₁₆ to 0073 ₁₆)	28	
$\overline{\text{INT0}}$	+116 to +119 (0074 ₁₆ to 0077 ₁₆)	29	$\overline{\text{INT}}$ interrupt, OSD
$\overline{\text{INT1}}$	+120 to +123 (0078 ₁₆ to 007B ₁₆)	30	
$\overline{\text{INT2}}$ /OSD2	+124 to +127 (007C ₁₆ to 007F ₁₆)	31	
Software interrupt (Note 3)	+128 to +131 (0080 ₁₆ to 0083 ₁₆) to +252 to +255 (00FC ₁₆ to 00FF ₁₆)	32 to 63	M16C/60, M16C/20 series software manual

Notes 1: Address relative to address in INTB

2: Use the IFSR2A register's IFSR26 and IFSR27 bits to select

3: These interrupts cannot be disabled using the I flag

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 7.3 shows the interrupt control registers.

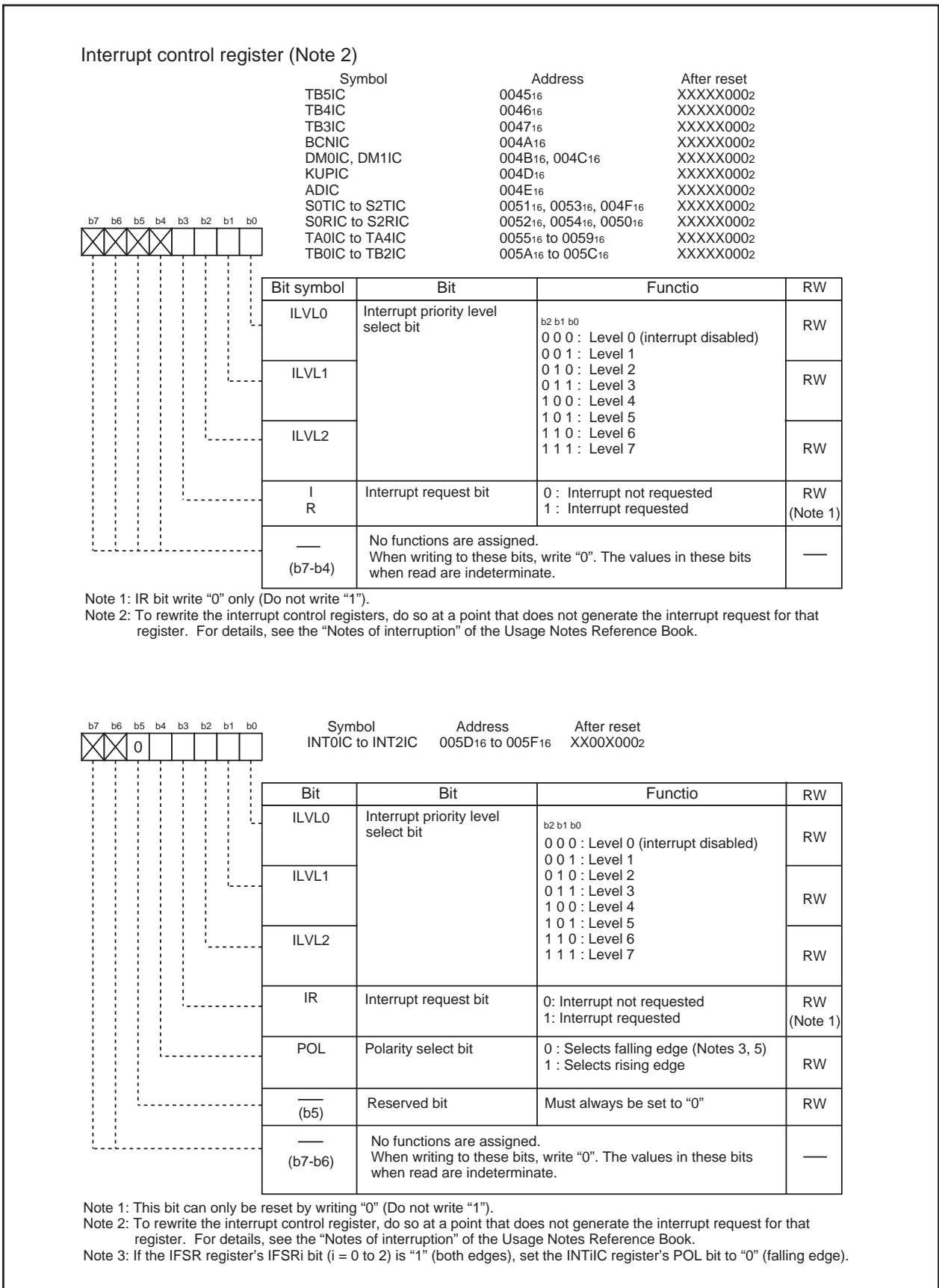


Figure 7.3. Interrupt Control Registers

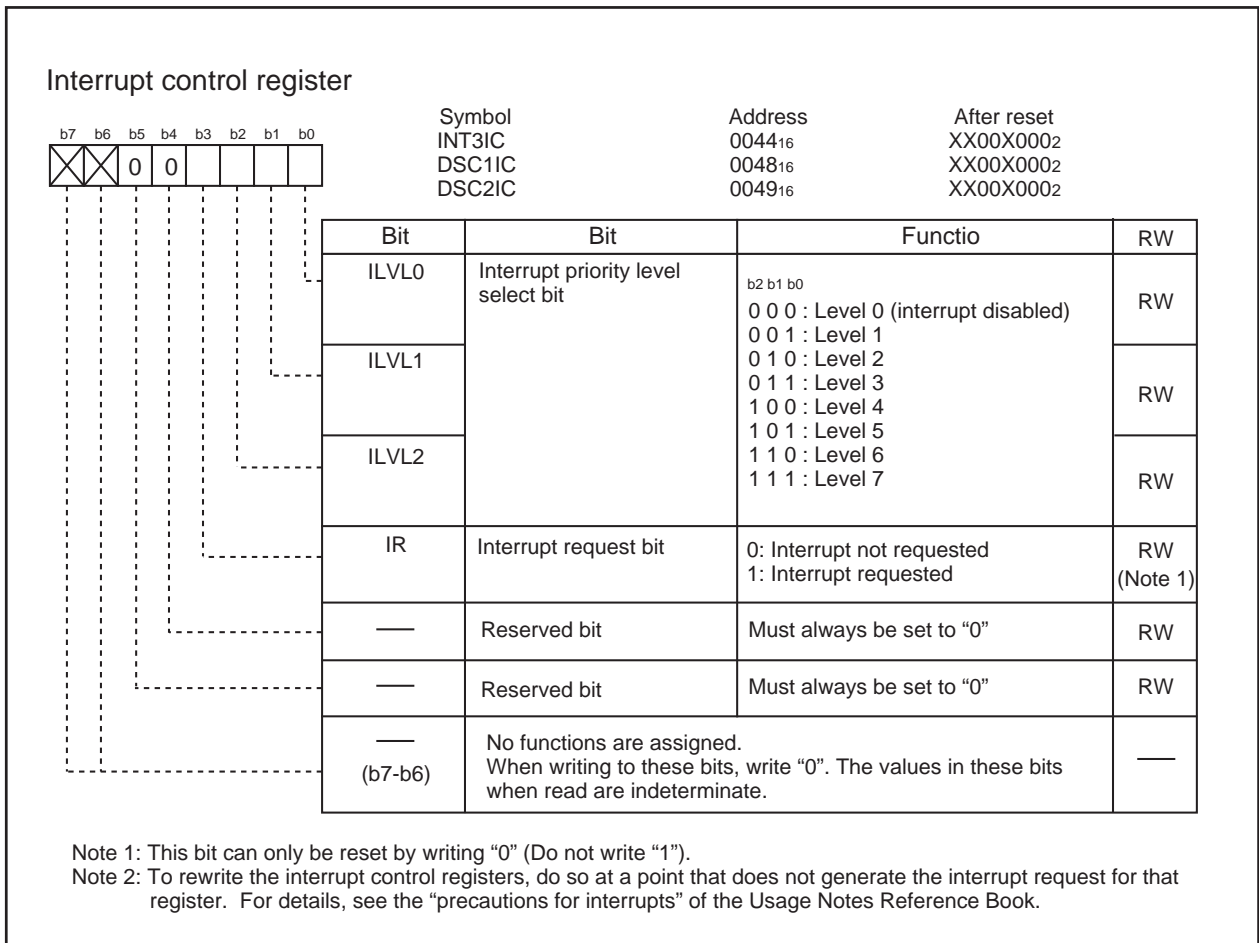


Figure 7.4. Interrupt Control Registers

I Flag

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

IR Bit

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

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

ILVL2 to ILVL0 Bits and IPL

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

Table 7.3 shows the settings of interrupt priority levels and Table 7.4 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 7.3. 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 7.4. Interrupt Priority Levels Enabled by IPL

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

Interrupt Sequence

An interrupt sequence — what are performed over a period from the instant an interrupt is accepted to the instant the interrupt routine is executed — is described here.

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

The CPU behavior during the interrupt sequence is described below. Figure 7.5 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 000016. Then it clears the IR bit for the corresponding interrupt to “0” (interrupt not requested).
- (2) The FLG register immediately before entering the interrupt sequence is saved to the CPU’s internal temporary register^(Note 1).
- (3) The I, D and U flags in the FLG register become as follows:
 - The I flag is cleared to “0” (interrupts disabled).
 - The D flag is cleared to “0” (single-step interrupt disabled).
 - The U flag is cleared to “0” (ISP selected).
 However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.
- (4) The CPU’s internal temporary register ^(Note 1) is saved to the stack.
- (5) The PC is saved to the stack.
- (6) The interrupt priority level of the accepted interrupt is set in the IPL.
- (7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

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

Note: This register cannot be used by user.

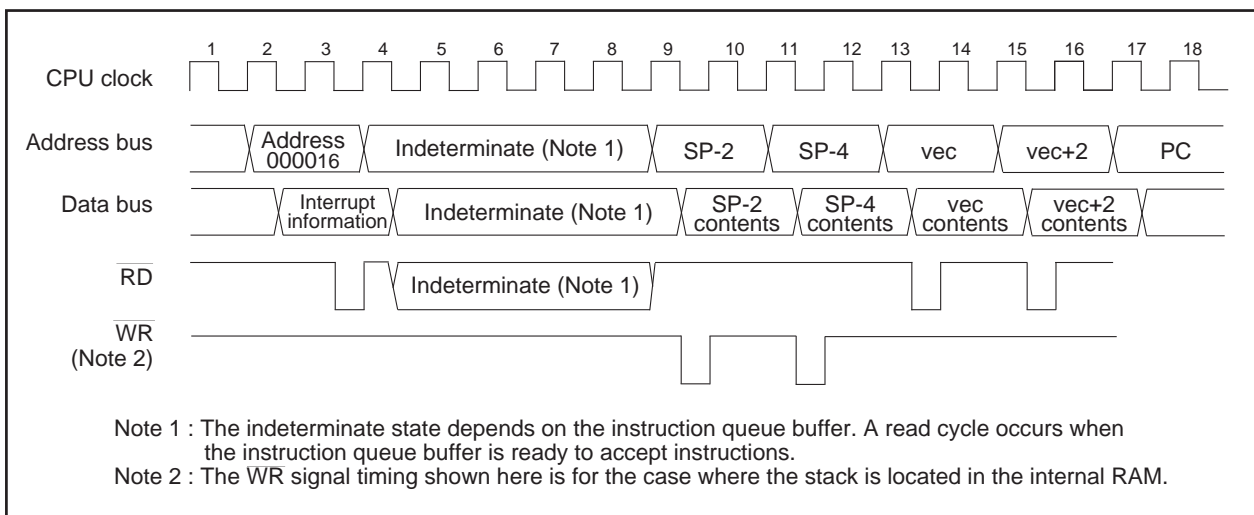


Figure 7.5. Time Required for Executing Interrupt Sequence

Interrupt Response Time

Figure 7.6 shows the interrupt response time. The interrupt response or interrupt acknowledge time denotes a time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of a time from when an interrupt request is generated till when the instruction then executing is completed ((a) in Figure 7.6) and a time during which the interrupt sequence is executed ((b) in Figure 7.6).

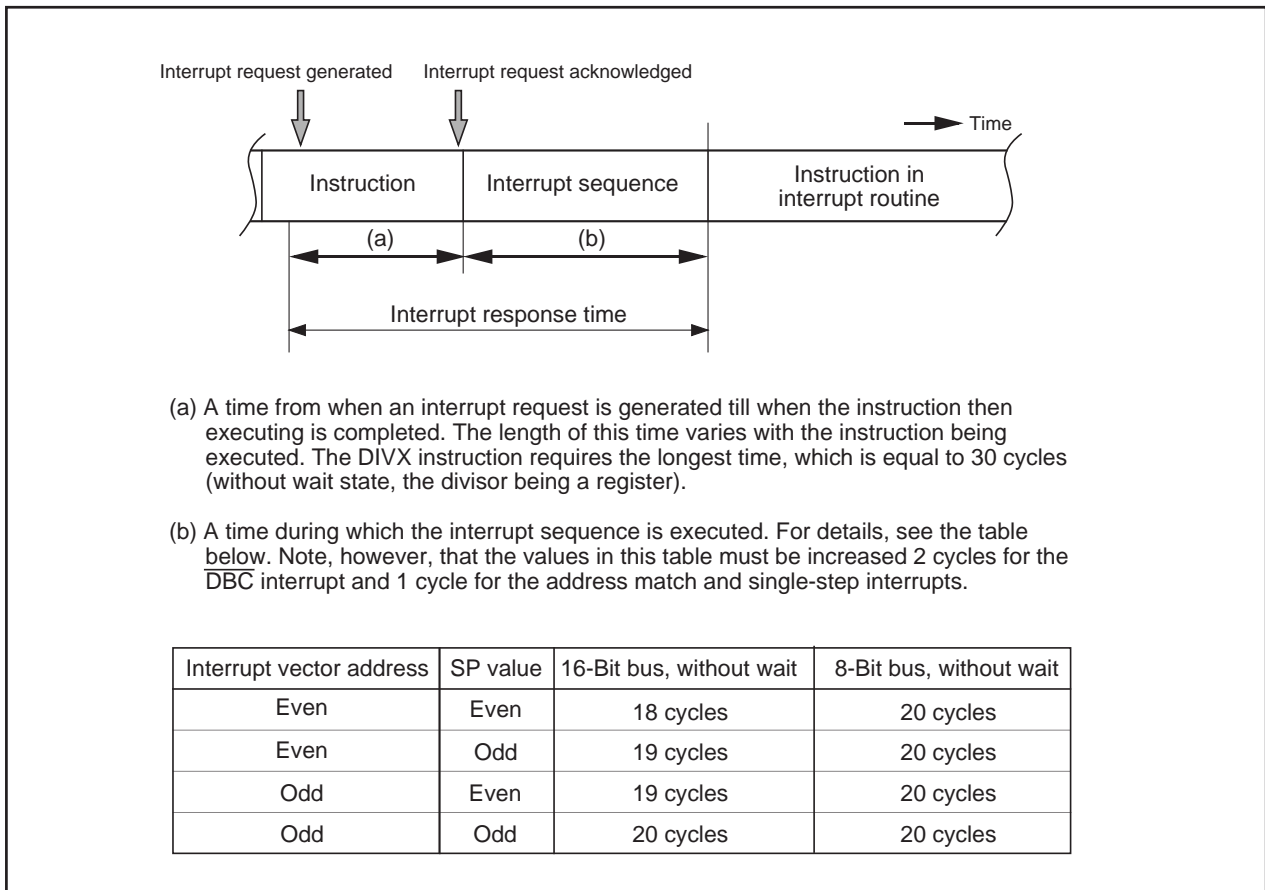


Figure 7.6. Interrupt response time

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 7.5 is set in the IPL. Shown in Table 7.5 are the IPL values of software and special interrupts when they are accepted.

Table 7.5. 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	7
Software, address match, DBC, single-step	Not changed

Saving Registers

In the interrupt sequence, the FLG register and PC are saved to the stack.

At this time, the 4 high-order bits 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 7.7 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.

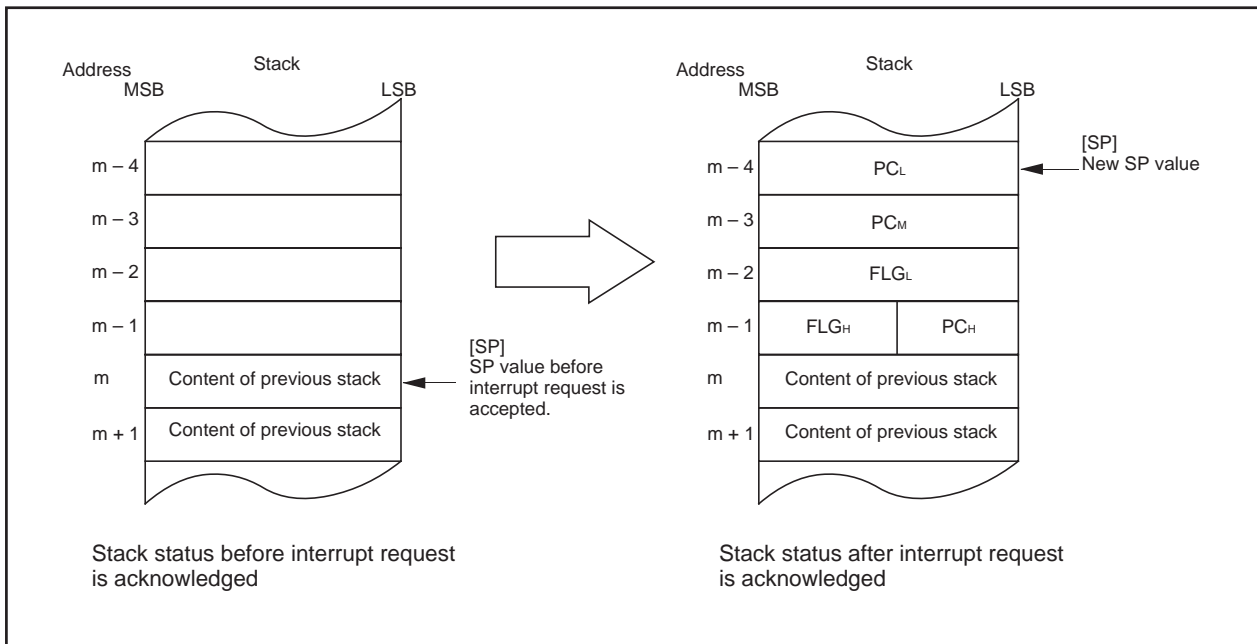


Figure 7.7. Stack Status Before and After Acceptance of Interrupt Request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP^(Note), at the time of acceptance of an interrupt request, is even or odd. If the stack pointer ^(Note) 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 7.8 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.

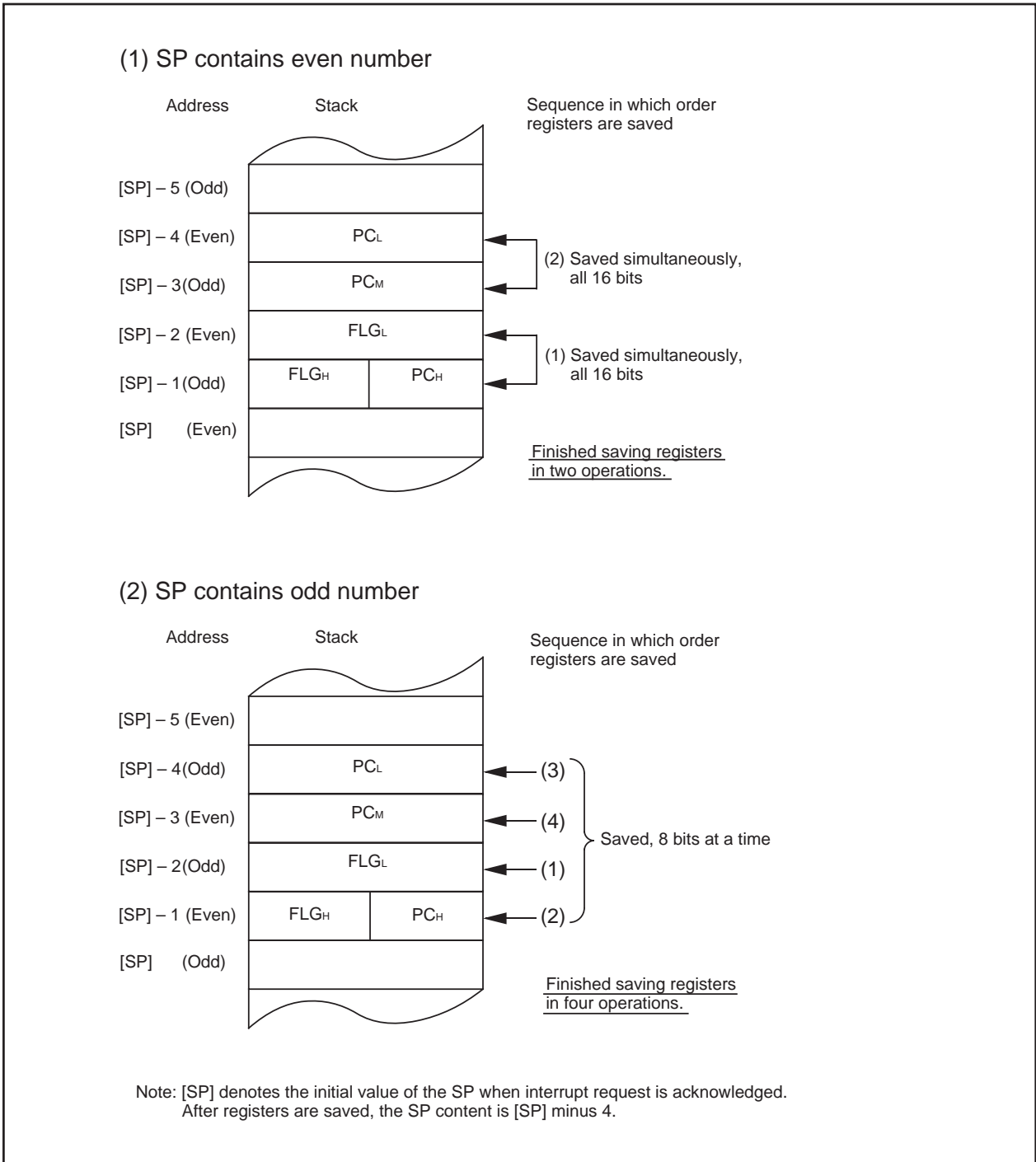


Figure 7.8. Operation of Saving Register

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 7.9 shows the priorities of hardware interrupts.

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

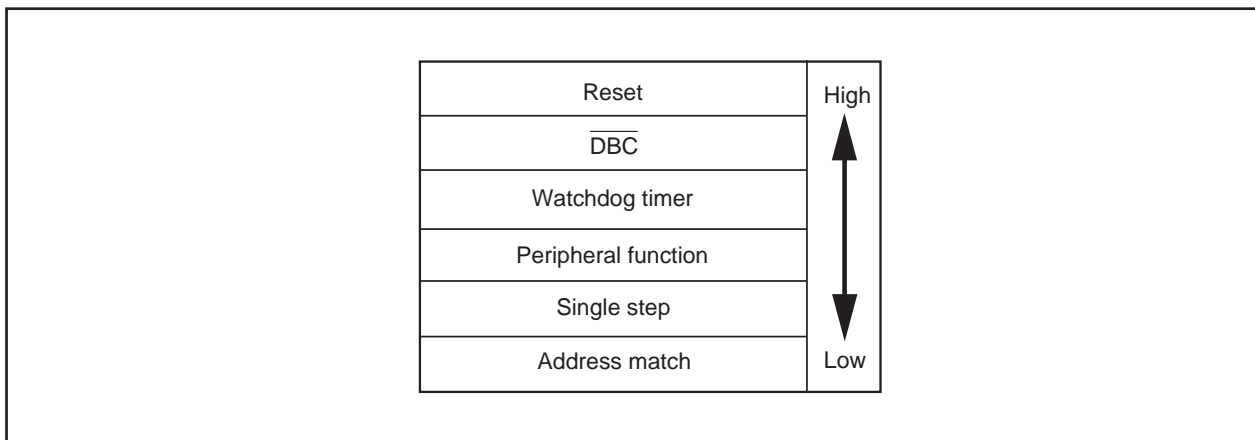


Figure 7.9. Hardware Interrupt Priority

Interrupt Priority Resolution Circuit

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

Figure 7.10 shows the circuit that judges the interrupt priority level.

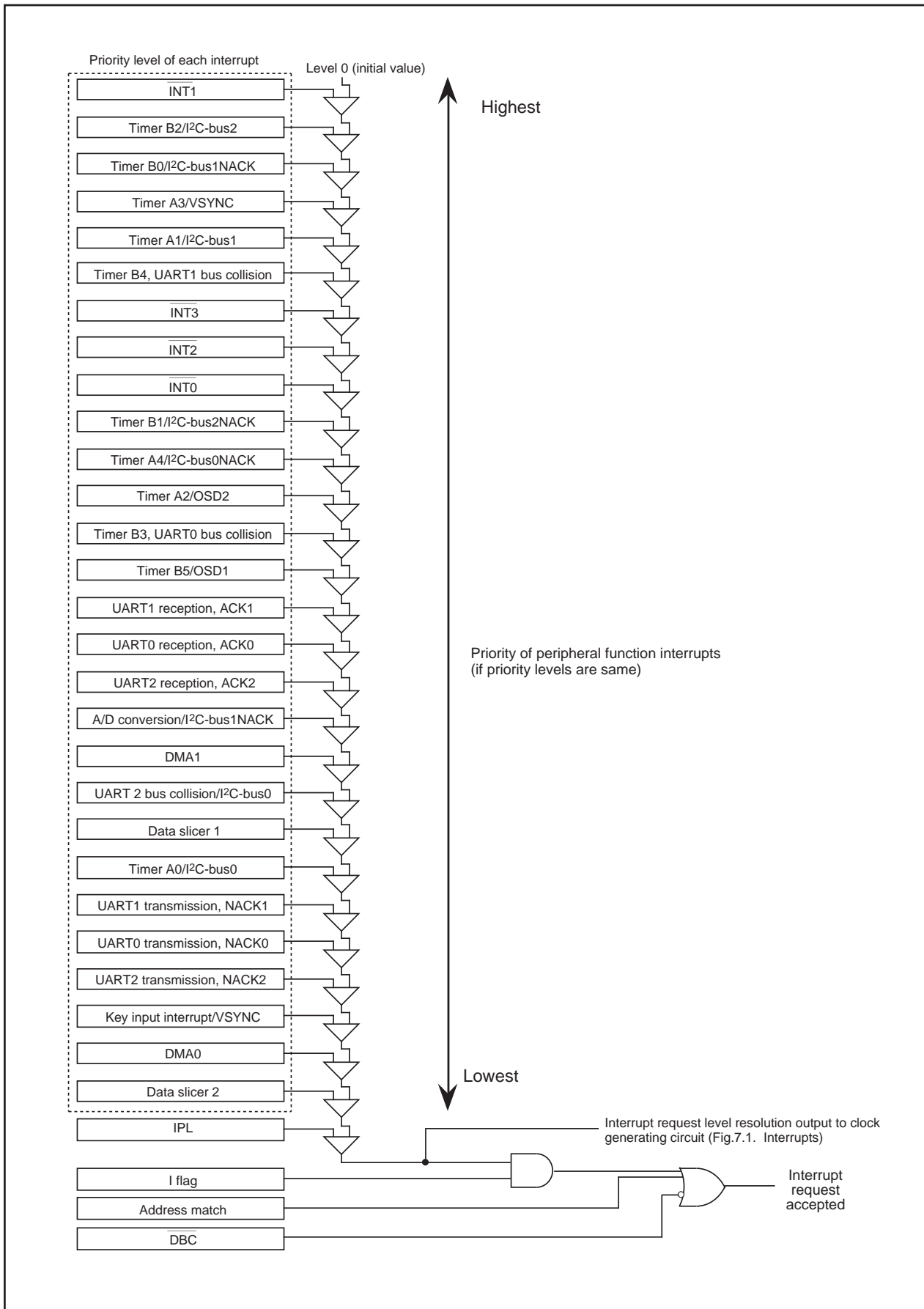


Figure 7.10. Interrupts Priority Select Circuit

INT Interrupt

INT_i interrupt (i=0 to 3) is triggered by the edges of external inputs. The edge polarity is selected using the IFSR register's IFSR_i bit.

Figure 7.11 shows the IFSR and IFSR2A registers.

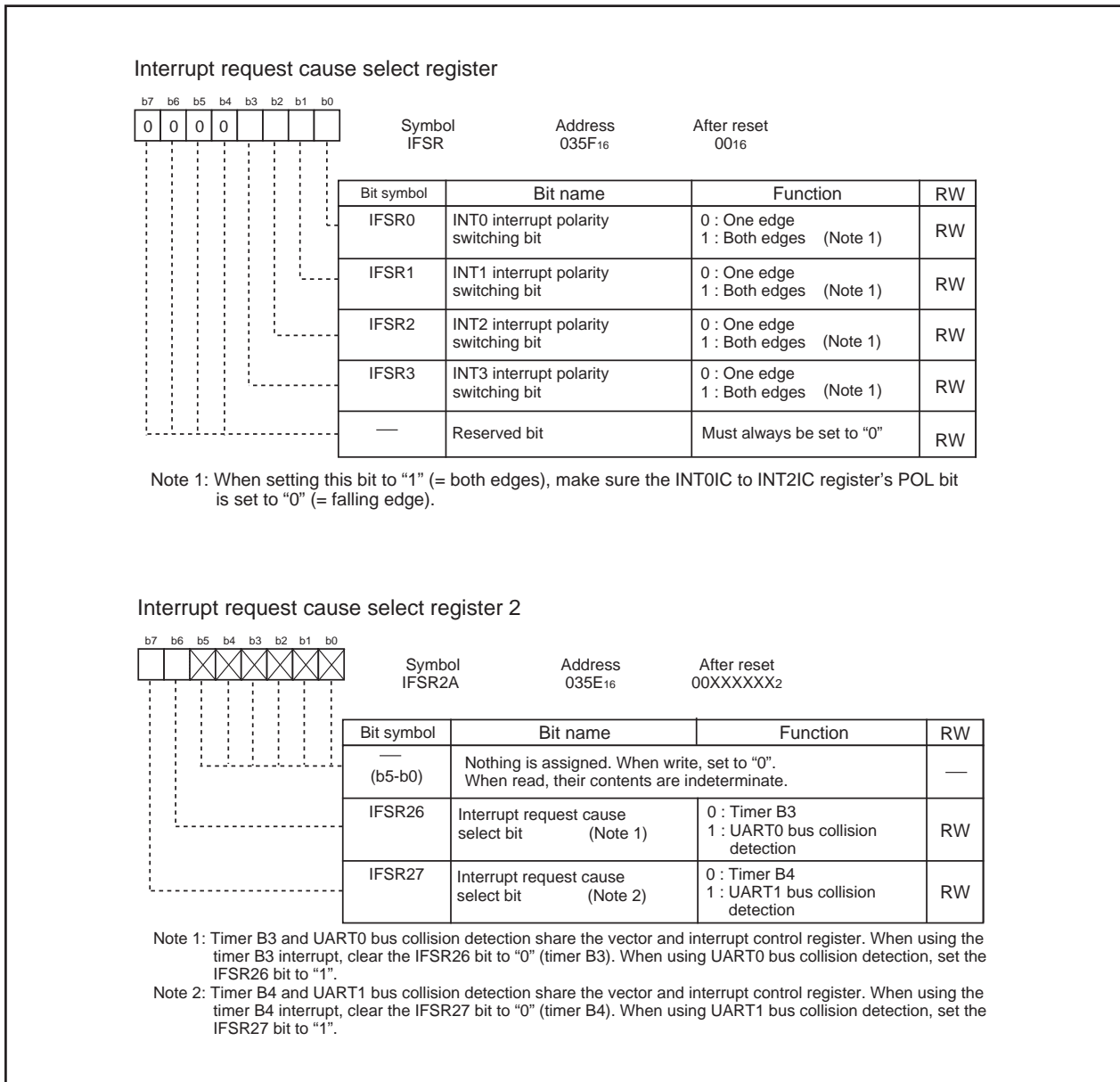


Figure 7.11. IFSR Register and IFSR2A Register

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 7.12 shows the block diagram of the key input interrupt. Note, however, that while input on any pin which has had the PD10_4 to PD10_7 bits set to "0" (= input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

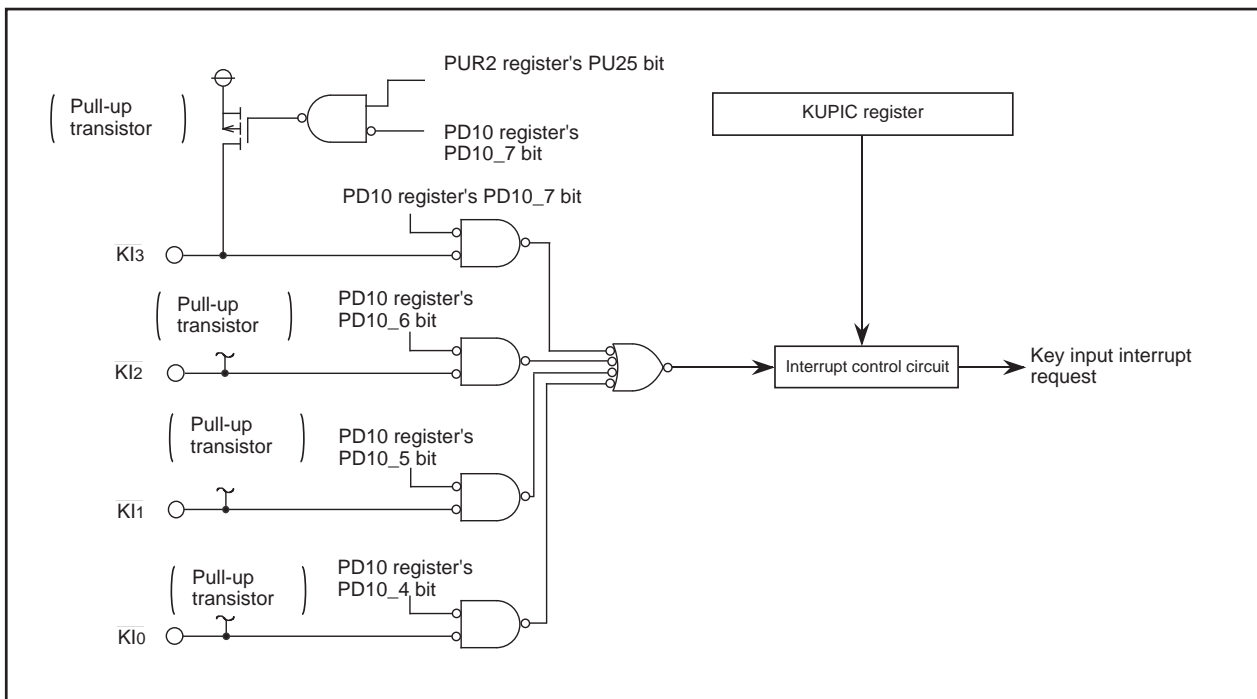


Figure 7.12. Key Input Interrupt

Address Match Interrupt

An address match interrupt request is generated immediately before executing the instruction at the address indicated by the RMAD_i register (i=0 to 3). Set the start address of any instruction in the RMAD_i 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 (refer to "Saving Registers").

(The value of the PC that is saved to the stack area is not the correct return address.) Therefore, follow one of the methods described below to return from the address match interrupt.

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

Table 7.6 shows the value of the PC that is saved to the stack area when an address match interrupt request is accepted.

Note that when using the external bus in 8 bits width, no address match interrupts can be used for external areas.

Figure 7.13 shows the AIER, AIER2, and RMAD0 to RMAD3 registers.

Table 7.6. Value of the PC that is saved to the stack area when an address match interrupt request is accepted.

Instruction at the address indicated by the RMAD _i register	Value of the PC that is saved to the stack area
<ul style="list-style-type: none"> • 16-bit op-code instruction • Instruction shown below among 8-bit operation code instructions <pre> 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) </pre>	The address indicated by the RMAD _i register +2
Instructions other than the above	The address indicated by the RMAD _i register +1

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

Table 7.7. Relationship Between Address Match Interrupt Sources and Associated Registers

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

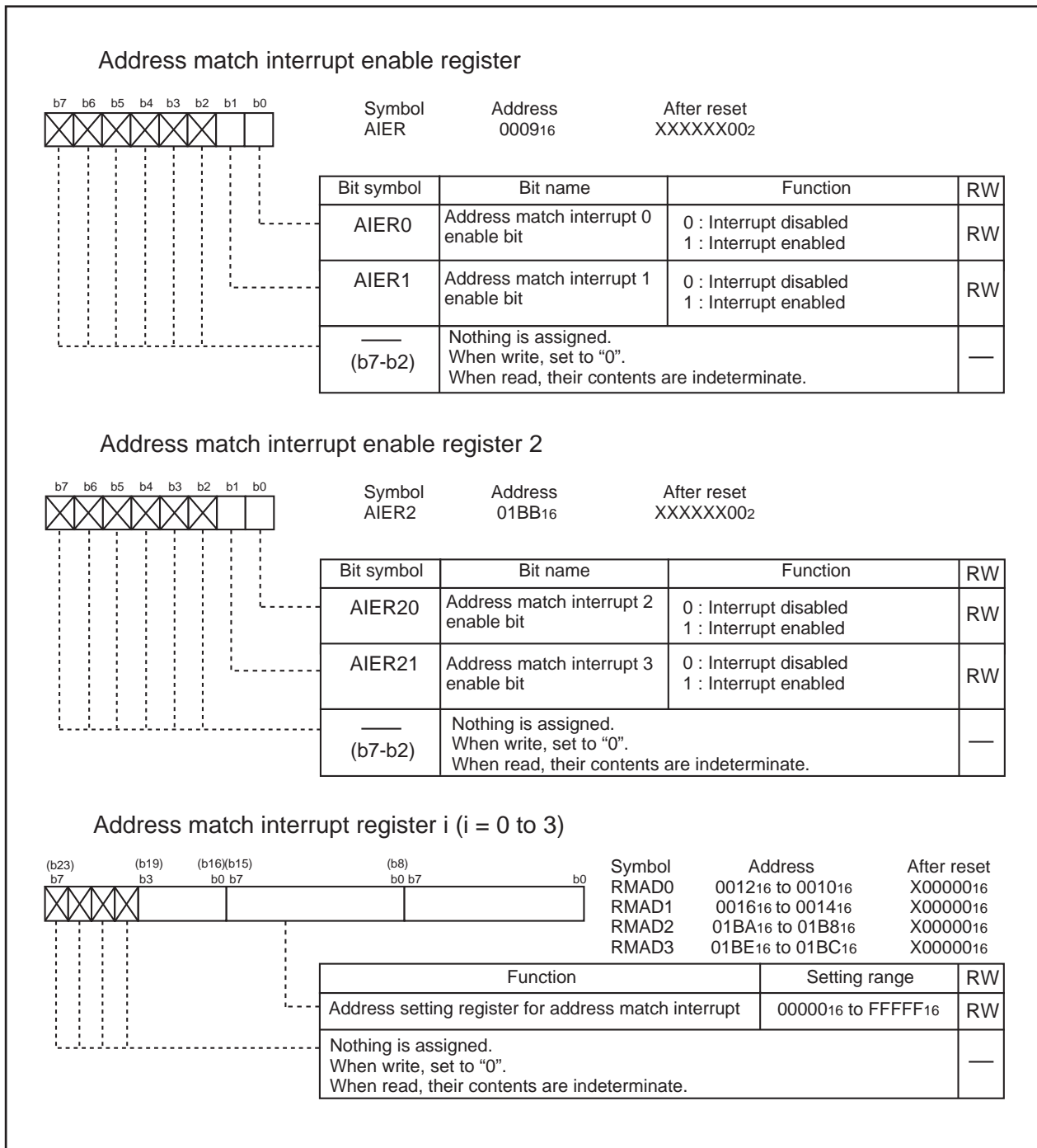


Figure 7.13. AIER Register, AIER2 Register and RMAD0 to RMAD3 Registers

Notes of interruption

(1) Address 00000₁₆ read-out

- Please do not read address 00000₁₆ by the program. When the interruption demand of maskable interruption is received, CPU interrupts in an interruption sequence and reads information (it interrupts with an interruption number and is a demand level) from address 00000₁₆. IR bit of received interruption is set to "0" at this time. If address 00000₁₆ is read by the program, IR bit of high interruption of a priority will be most set to "0" among interruption permitted. Therefore, interruption may be canceled or unexpected interruption may occur.

(2) Setting of SP

- Please assign a value to SP before receiving interruption. After reset, SP is "0000₁₆." Therefore, if interruption is received before assigning a value to SP, it will become the factor of a reckless run.

(3) $\overline{\text{INT}}$ interrupt

- Regardless of CPU clock, "L" width or "H" width for 250ns or more is required for the signal inputted into pins $\overline{\text{INT}}_0$ to $\overline{\text{INT}}_3$.
- IR bit may be set to "1" (those with an interruption demand) when changing the polarity of pins $\overline{\text{INT}}_0$ to $\overline{\text{INT}}_3$. Please set IR bit to "0" (with no interruption demand) after changing. The example of a change procedure of an $\overline{\text{INT}}$ interruption generating factor is shown in Fig. 7.14.

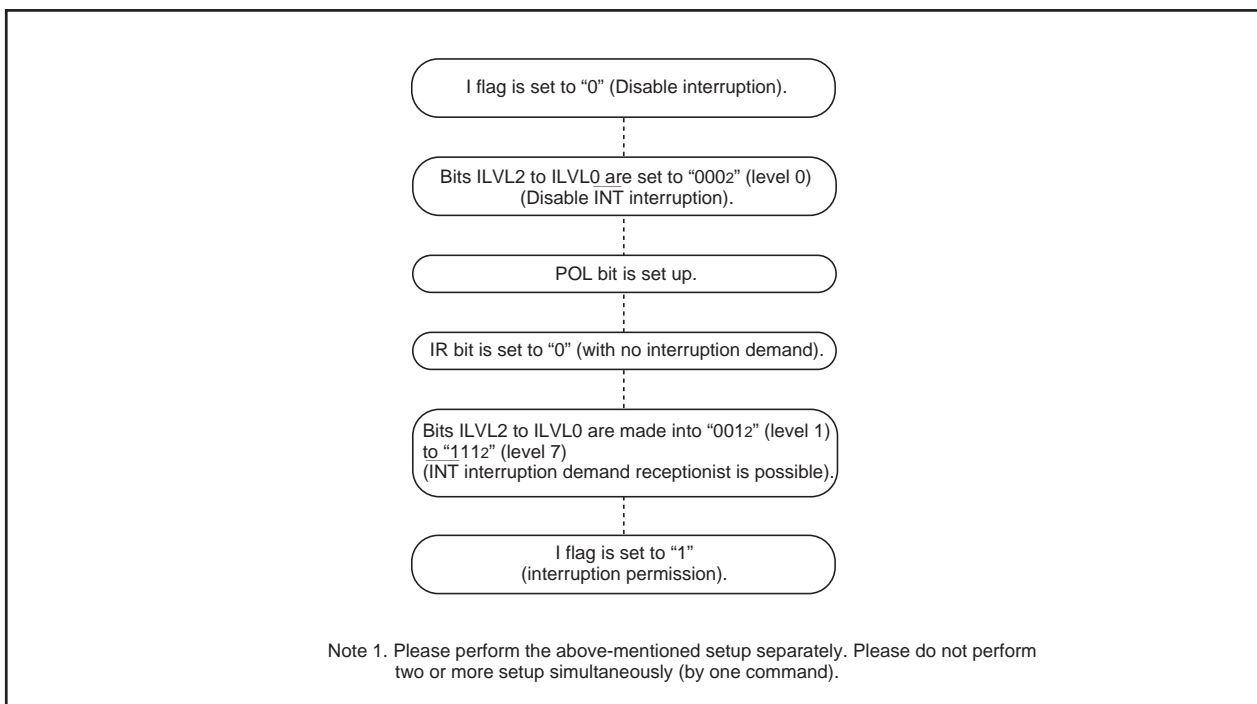


Figure 7.14. The example of change procedure of INT interruption generating factor

(4) Watchdog timer interruption

After watchdog timer interruption generating should initialize watchdog timer.

(5) Change of an interrupt control register

● Please make a change of an interrupt control register in the part which the interruption demand corresponding to the register does not generate. When an interruption demand may occur, please change after forbidding interruption. The example of a reference program is shown below.

<The example of program which rewrites an interruption control register>

Example 1:

```
INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00H, 0055H ; TA0IC register is set 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 ; TA0IC register is set to "0016."
  MOV.W MEM, R0    ; Dummy read.
  FSET  I           ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
  PUSHC FLG
  FCLR  I           ; Disable interrupts.
  AND.B #00H, 0055H ; TA0IC register is set to "0016."
  POPC  FLG        ; Enable interrupts.
```

The reason which has a dummy lead in Example 1 before two pieces (they are four pieces at the time of HOLD functional use) and an FSET I command in Example 2 of an NOP command before an FSET I command.
Under the influence of a command cue buffer, before writing to an interruption control register, it prevents setting I flag to "1."

When you forbid interruption, you interrupt and you change a control register, be careful of the command to be used.

Change of bits other than IR bit

During execution of a command, when the interruption demand corresponding to the register occurs, IR bit is not set to "1" (those with an interruption demand), but interruption may be disregarded.

The target command : AND, OR, BCLR, BSET

Change of IR bit

When setting IR bit to "0" (with no interruption demand), IR bit may not be set to "0" depending on the command to be used. Please set IR bit to "0" using MOV command.

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. Refer to “Watchdog Timer Reset” for the details of watchdog timer reset.

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

With main clock source 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 8.1 shows the block diagram of the watchdog timer. Figure 8.2 shows the watchdog timer-related registers.

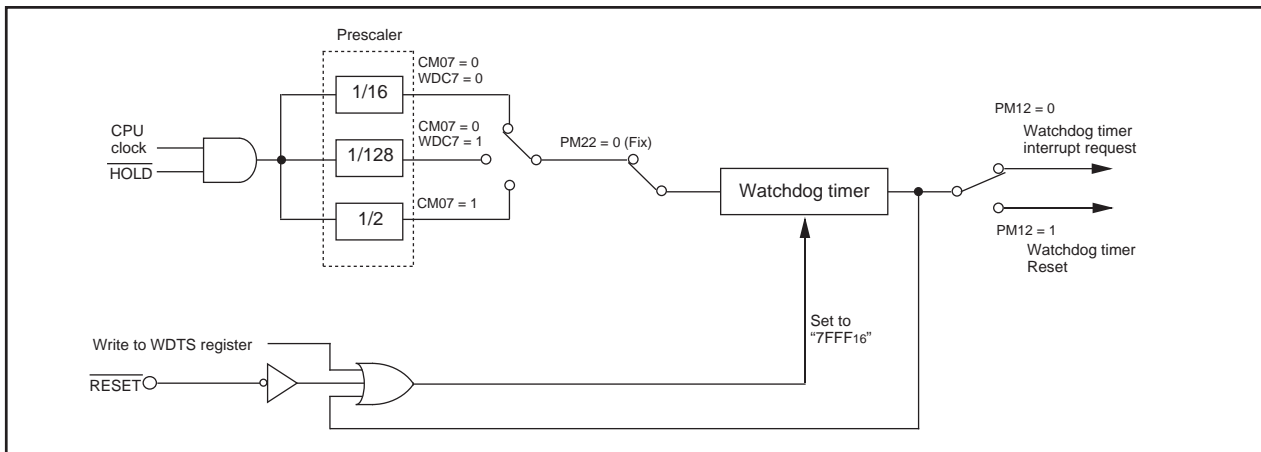


Figure 8.1. Watchdog Timer Block Diagram

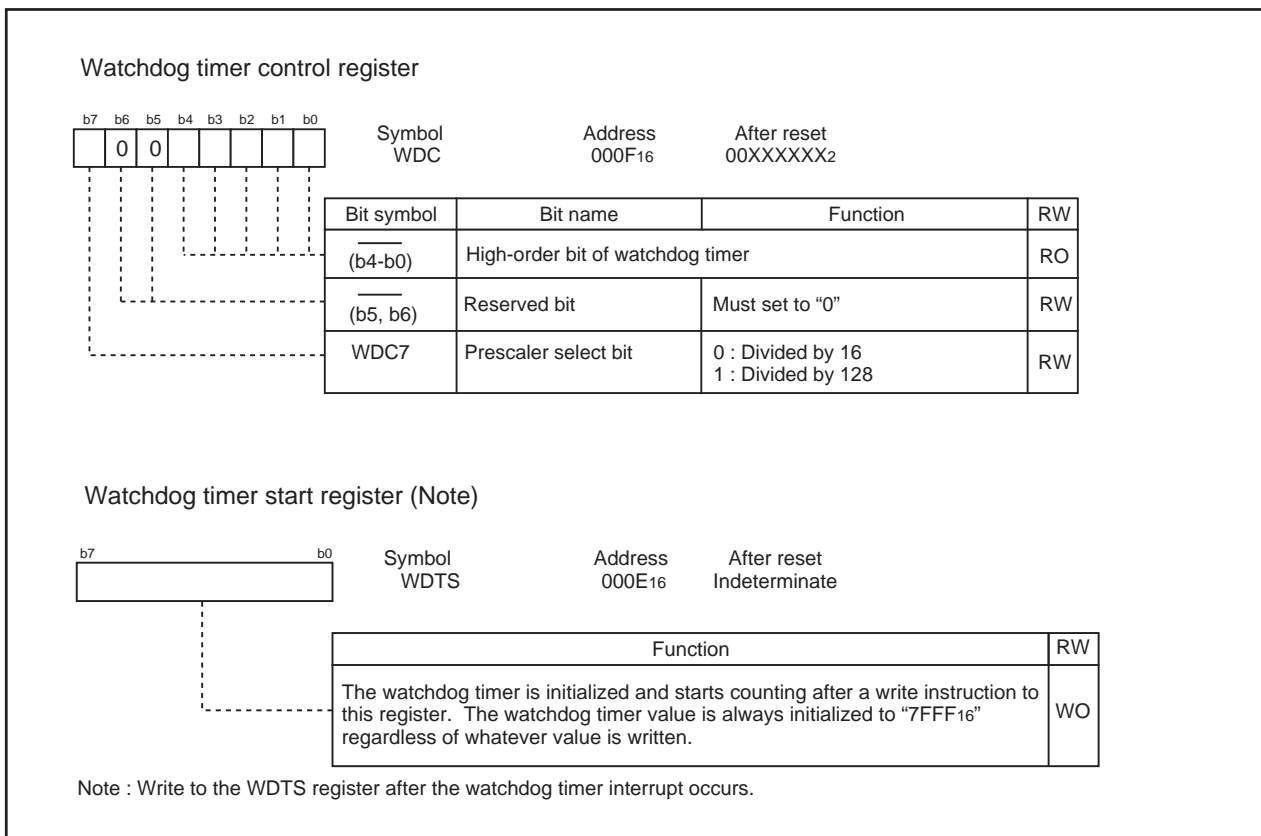


Figure 8.2. WDC Register and WDTS Register

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 9.1 shows the block diagram of the DMAC. Table 9.1 shows the DMAC specifications. Figures 9.2 to 9.4 show the DMAC-related registers.

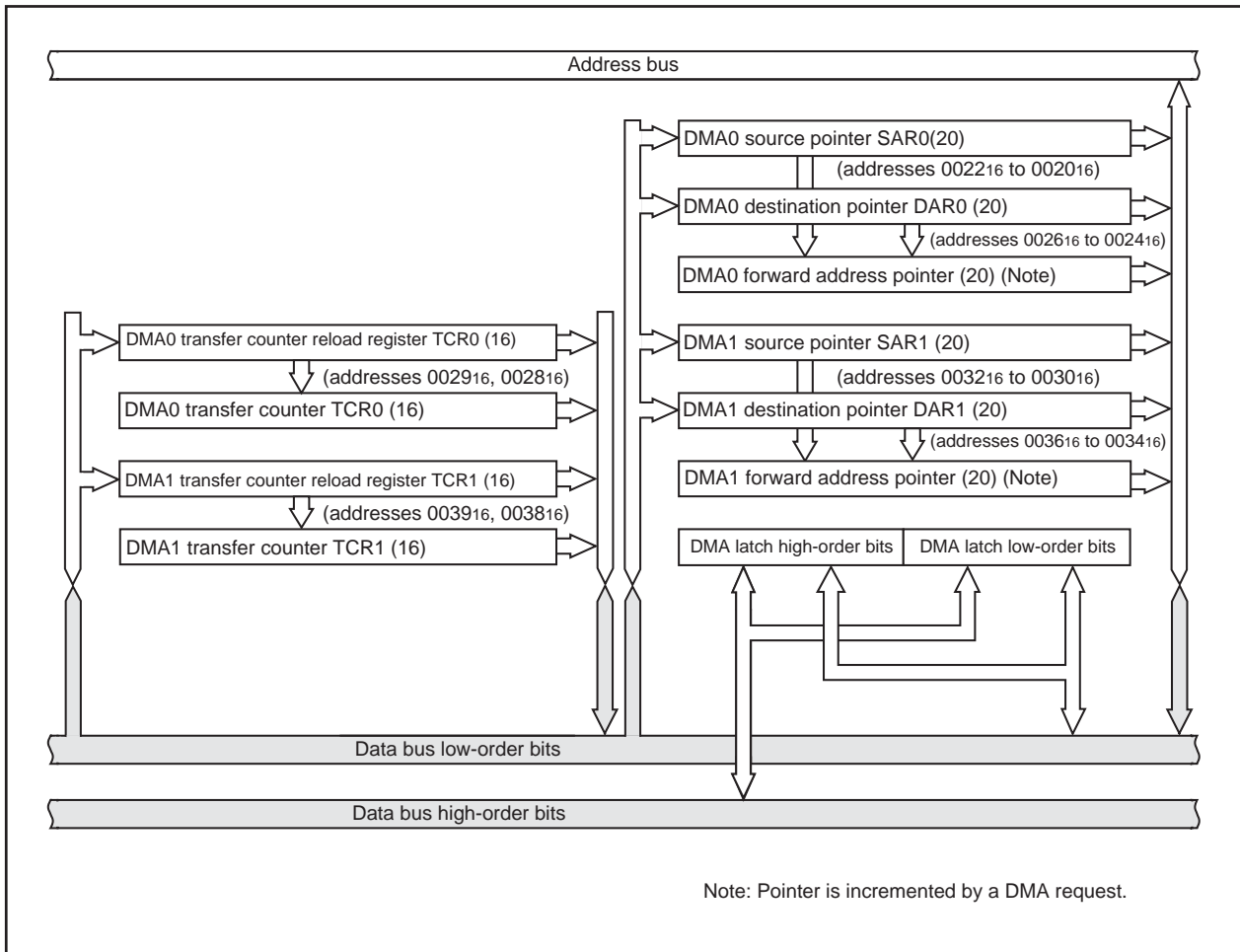


Figure 9.1. DMAC Block Diagram

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 9.1. DMAC Specifications

Item	Specification	
No. of channels	2 (cycle steal method)	
Transfer memory space	<ul style="list-style-type: none"> • From any address in the 1M bytes space to a fixed address • From a fixed address to any address in the 1M bytes space • From a fixed address to a fixed address 	
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 A/D conversion interrupt requests Software triggers OSD1, OSD2 interrupt VSYNC interrupt Multi-master I ² C-bus interface 0, 1, 2 interrupt I ² C-bus 0, 1, 2 NACK interrupt	
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 _i transfer counter (i = 0–1) underflows after reaching the terminal count.
	•Repeat transfer	When the DMA _i transfer counter underflows, it is reloaded with the value of the DMA _i transfer counter reload register and a DMA transfer is continued with it.
DMA interrupt request generation timing	When the DMA _i transfer counter underflowed	
DMA startup	Data transfer is initiated each time a DMA request is generated when the DMA _i CON register's DMAE bit = "1" (enabled).	
DMA shutdown	•Single transfer	• When the DMAE bit is set to "0" (disabled)
	•Repeat transfer	<ul style="list-style-type: none"> • After the DMA_i transfer counter underflows • 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 _i or the DAR _i pointer whichever is specified to be in the forward direction and the DMA _i transfer counter is reloaded with the value of the DMA _i 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 002016–003F16) are accessed by the DMAC.

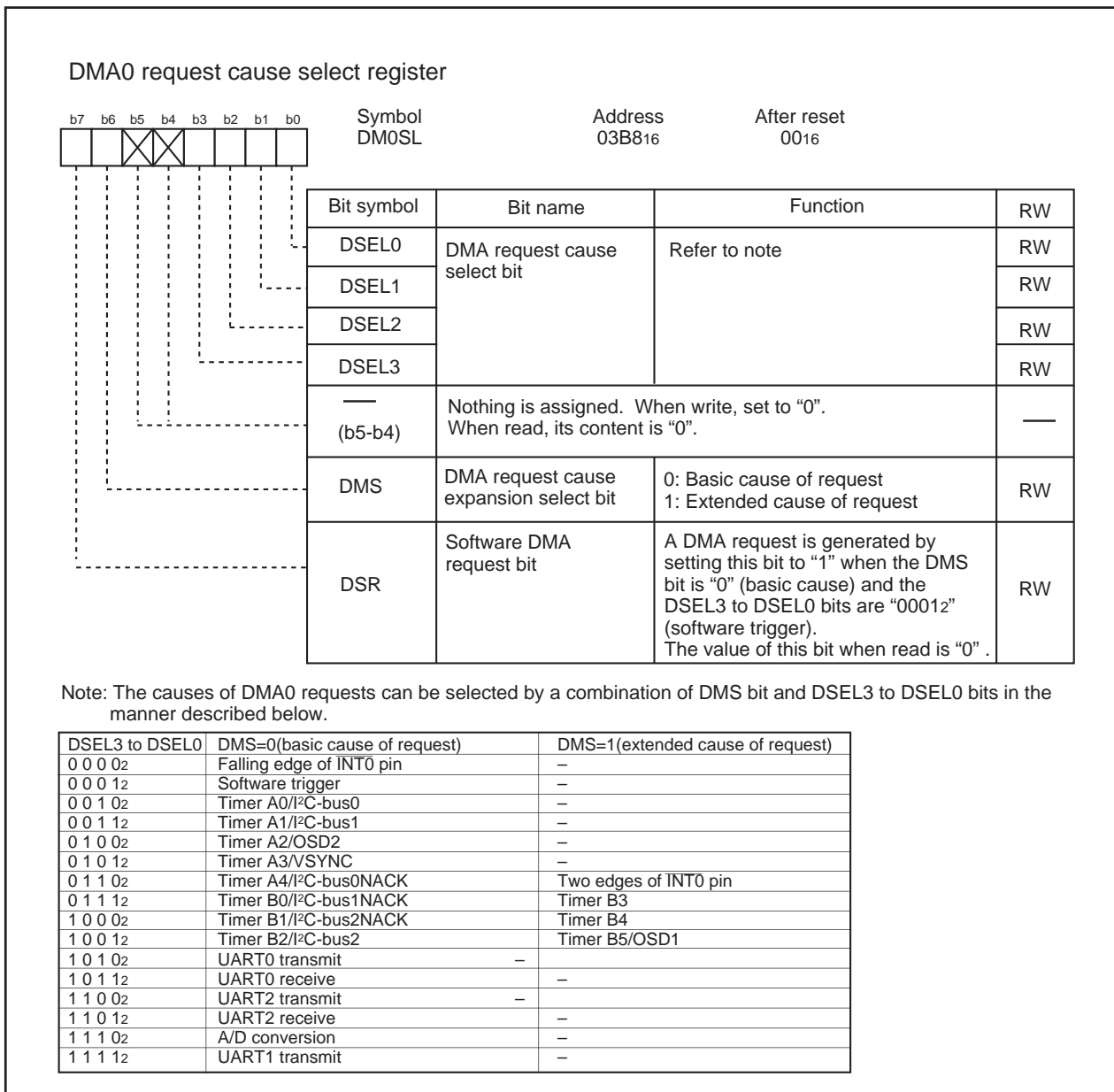


Figure 9.2. DM0SL Register

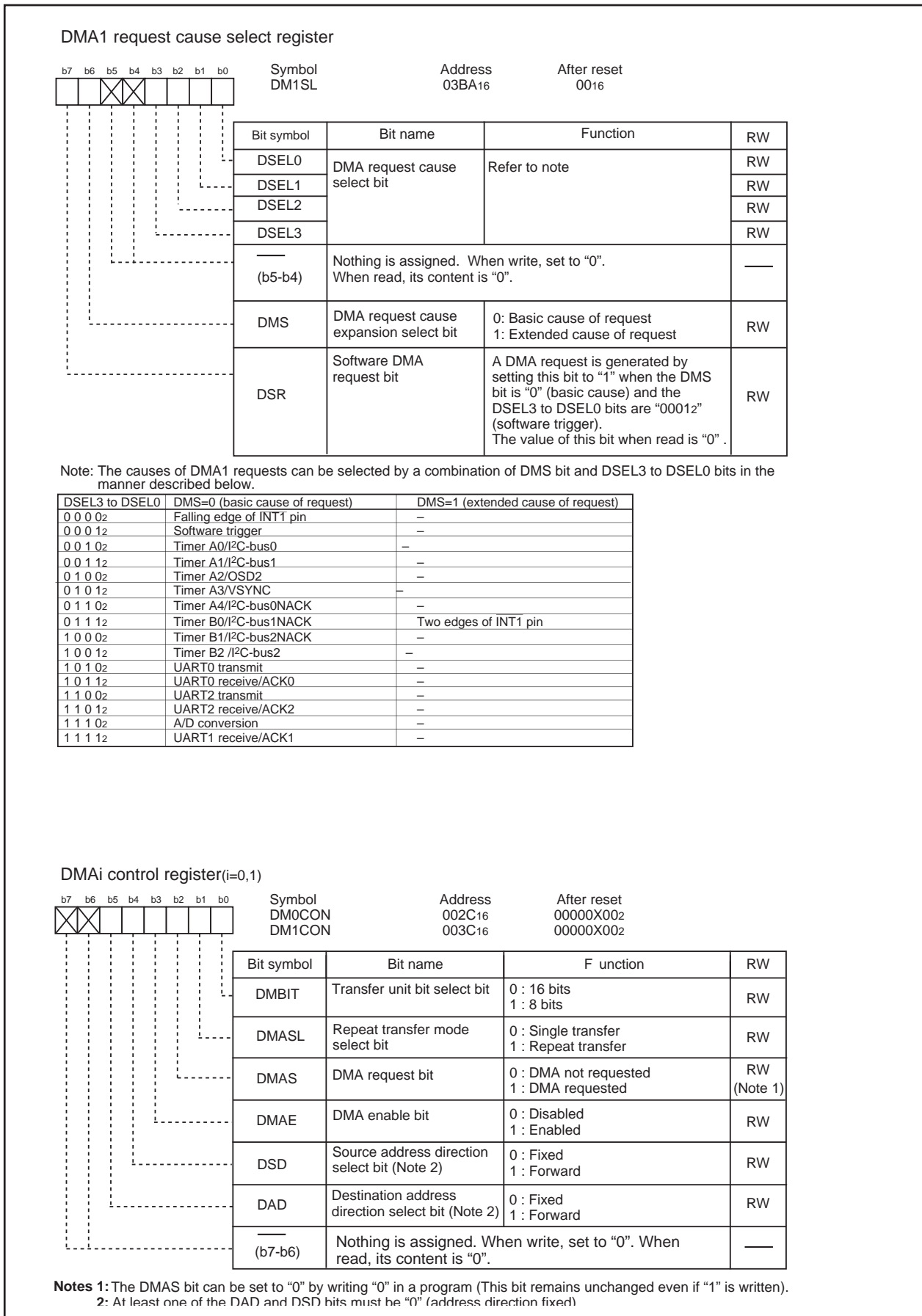


Figure 9.3. DM1SL Register, DM0CON Register, and DM1CON Registers

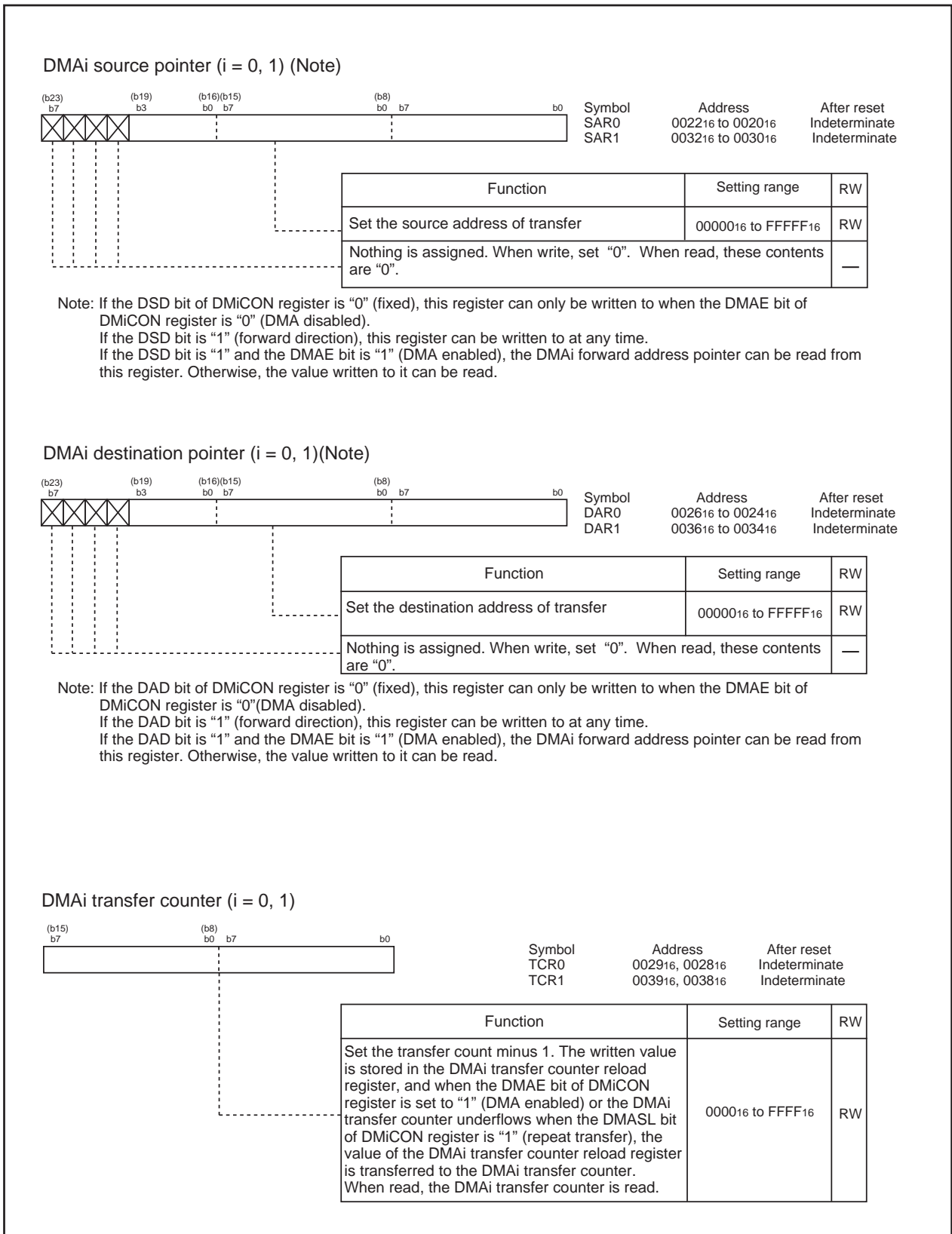


Figure 9.4. SAR0, SAR1, DAR0, DAR1, TCR0, and TCR1 Registers

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 9.5 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 9.5), two source read bus cycles and two destination write bus cycles are required.

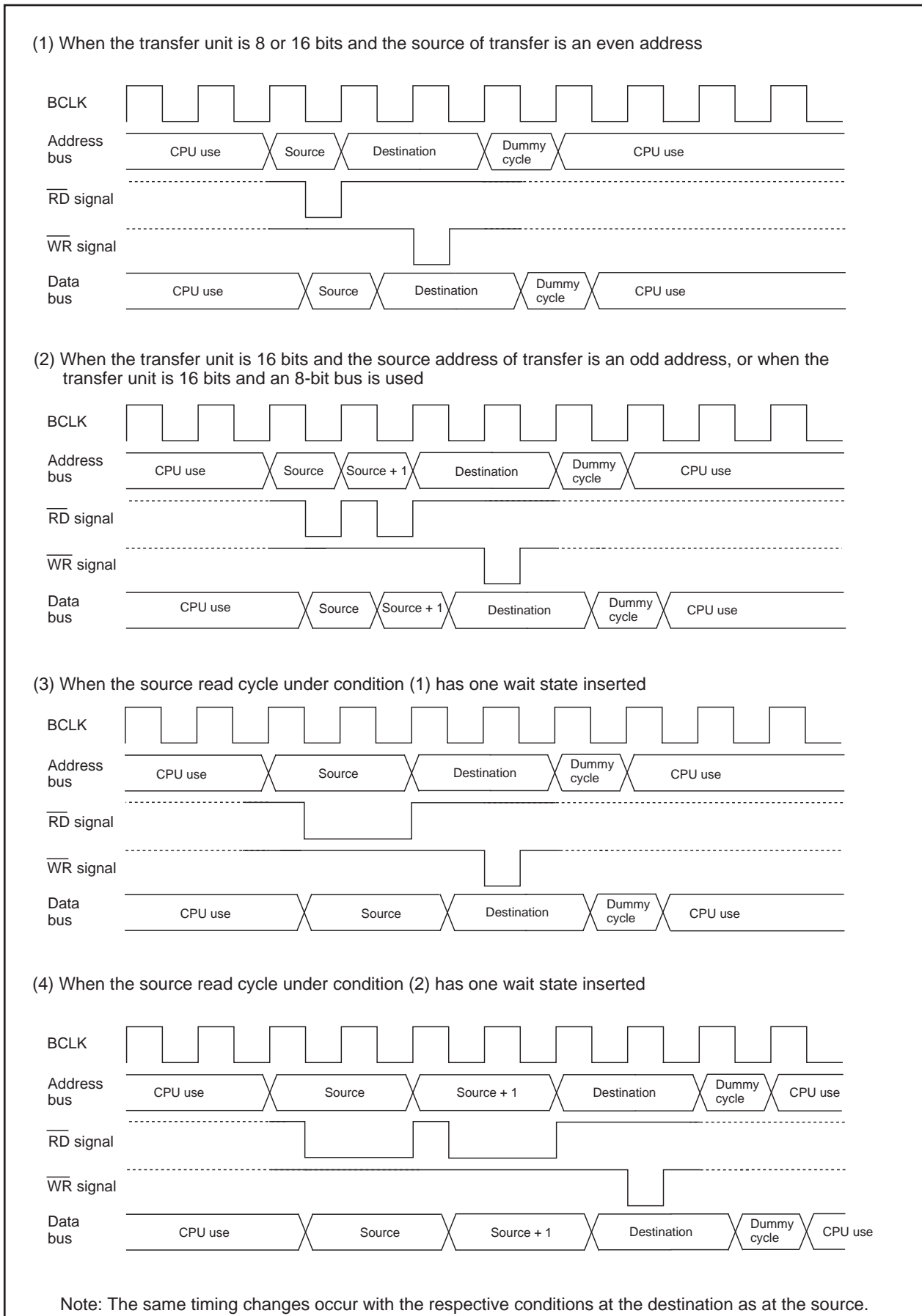


Figure 9.5. Transfer Cycles for Source Read

2. DMA Transfer Cycles

Any combination of even or odd transfer read and write addresses is possible. Table 9.2 shows the number of DMA transfer cycles. Table 9.3 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 9.2. 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 9.3. Coefficient j, k

	Internal area				External area						
	Internal ROM, RAM		SFR		Separate bus			Multiplex bus			
	No wait	With wait	1-wait ²	2-wait ²	No wait	With wait ¹			With wait ¹		
						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 9.4 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 9.4. 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"> • Immediately before a data transfer starts • When set by writing “0” in a program
Peripheral function	When the interrupt control register for the peripheral function that is selected by the DSEL3 to DSEL0 and DMS bits of DMiCON register has its IR bit set to “1”	

5. Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of BCLK), the DMAS bit on each channel is set to "1" (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1. The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period. Figure 9.6 shows an example of DMA transfer effected by external factors.

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

In addition, DMA requests cannot be counted up since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in Figure 9.6, occurs more than one time, the DMAS bit is set to "0" as soon as getting the bus arbitration. The bus arbitration is returned to the CPU when one transfer is completed. Refer to "(7) HOLD Signal in Bus Control" for details about bus arbitration between the CPU and DMA.

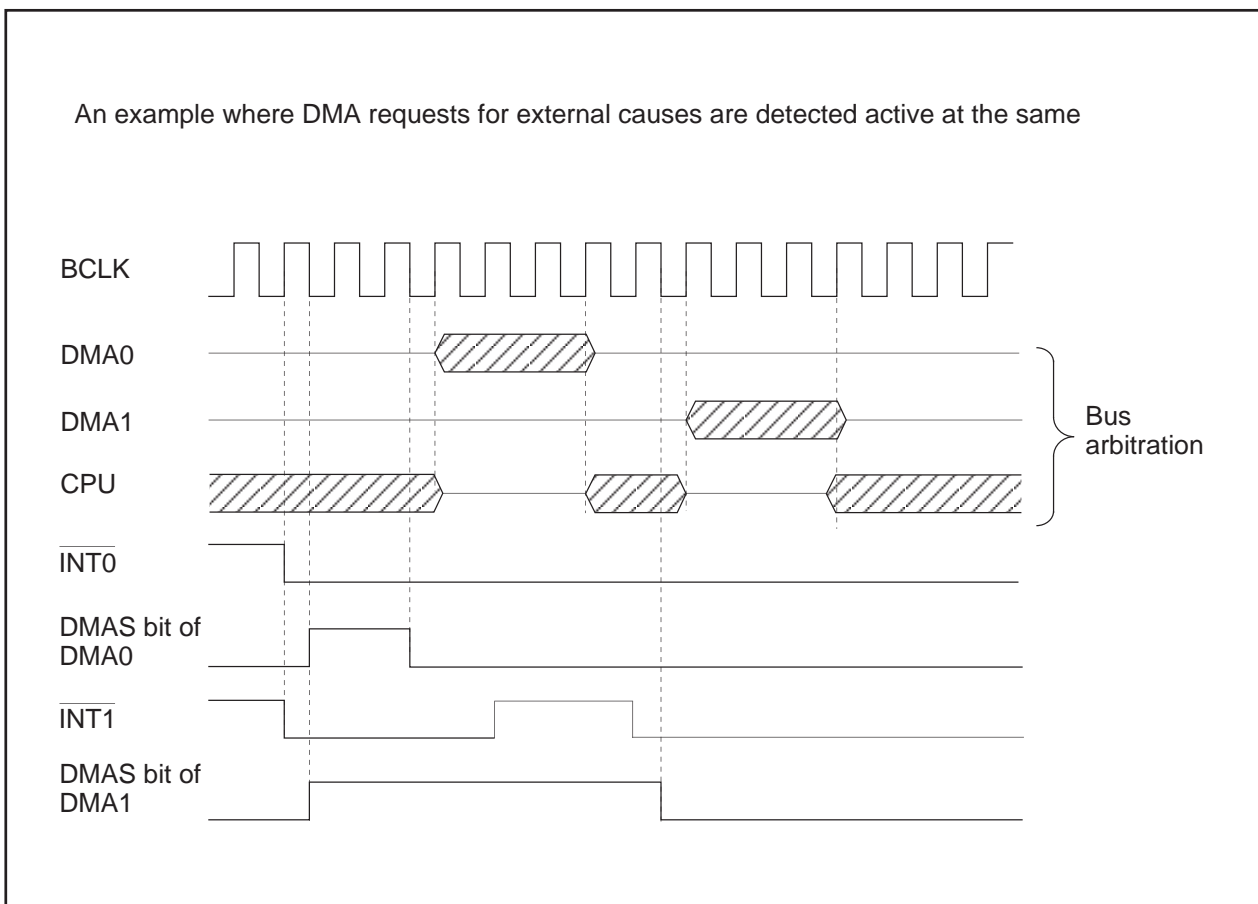


Figure 9.6. DMA Transfer by External Factors

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 10.1 and 10.2 show block diagrams of timer A and timer B configuration, respectively.

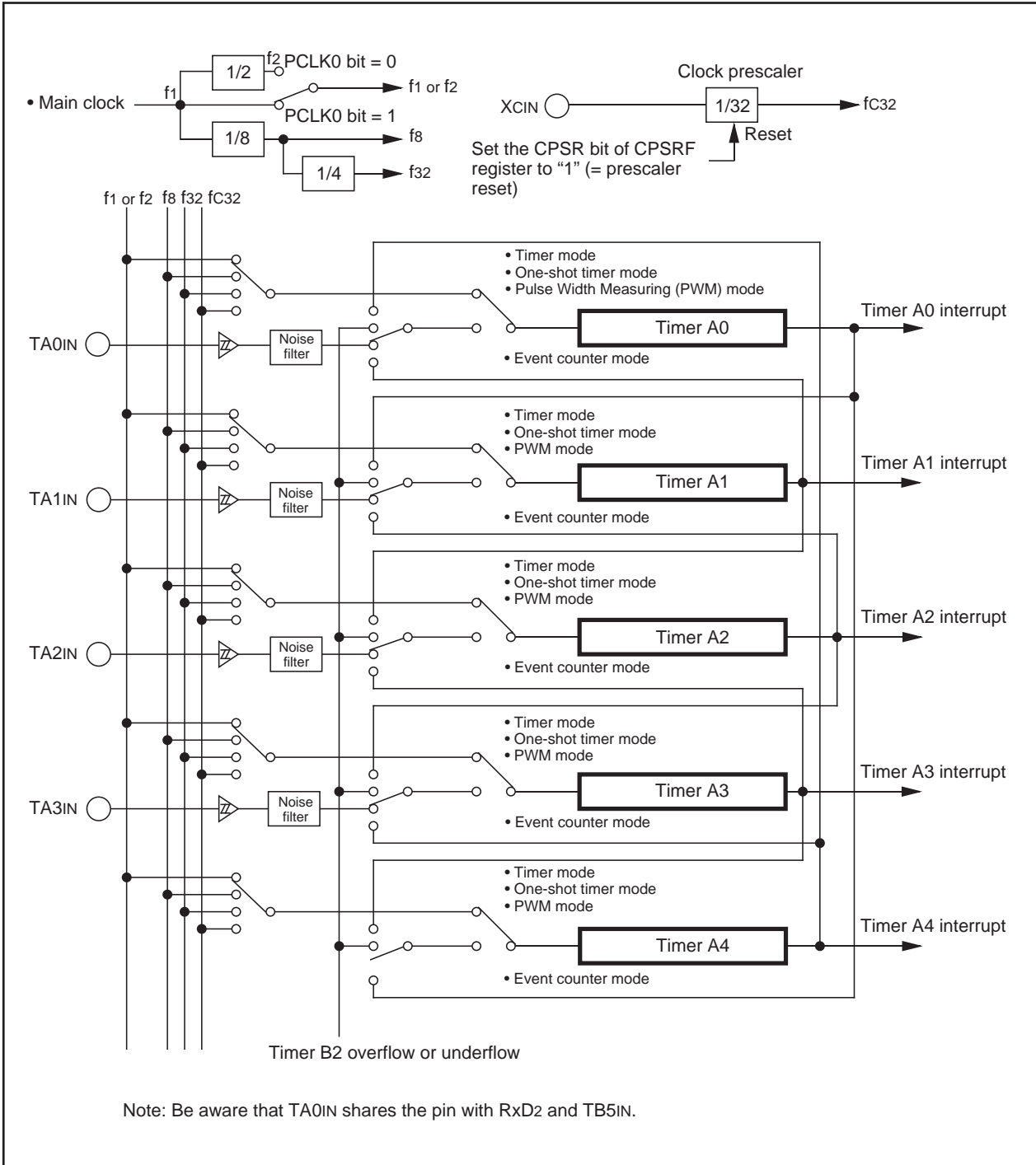


Figure 10.1. Timer A Configuration

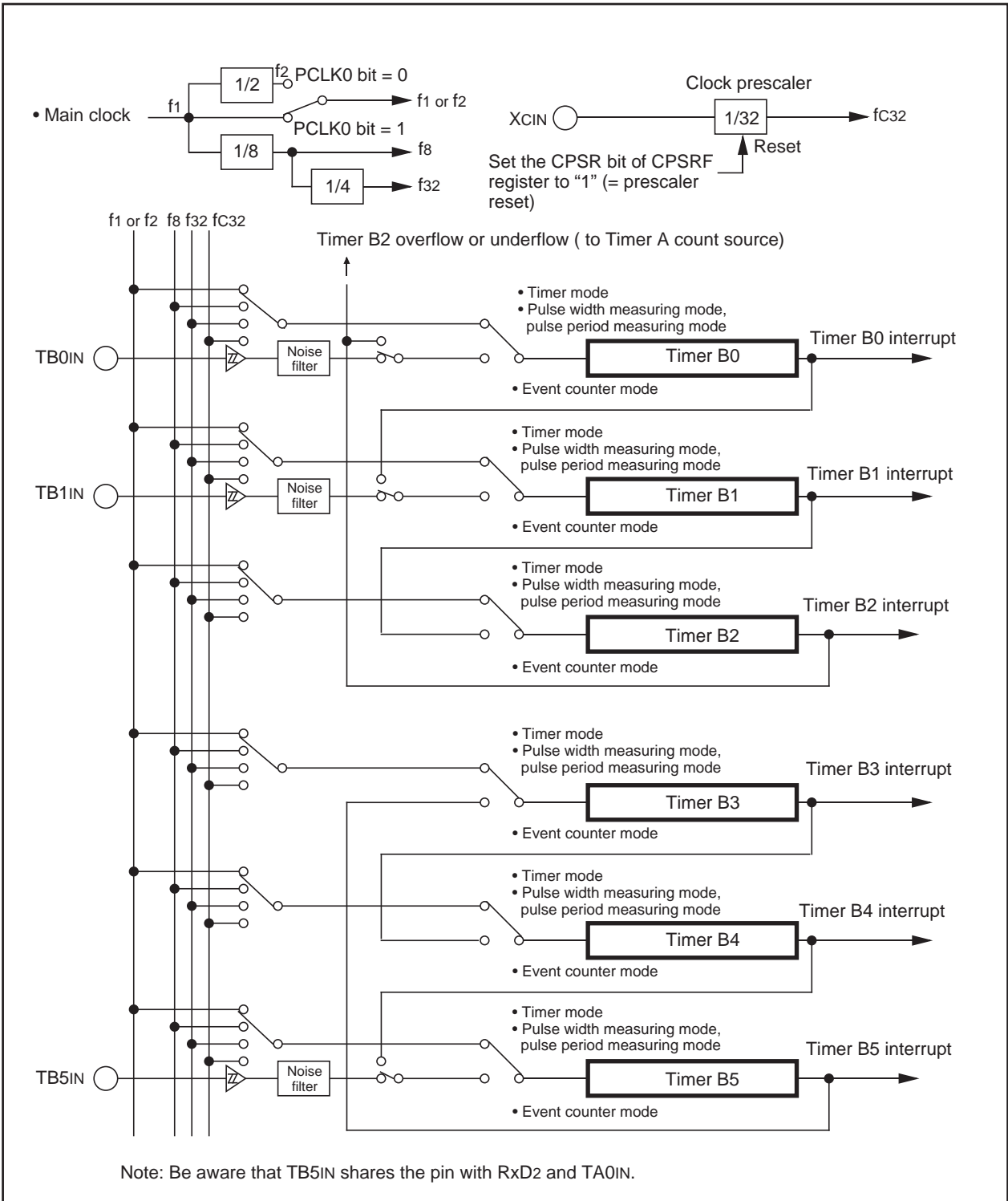


Figure 10.2. Timer B Configuration

Timer A

Figure 10.3 shows a block diagram of the timer A. Figures 10.4 to 10.6 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 “0000₁₆.”
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.

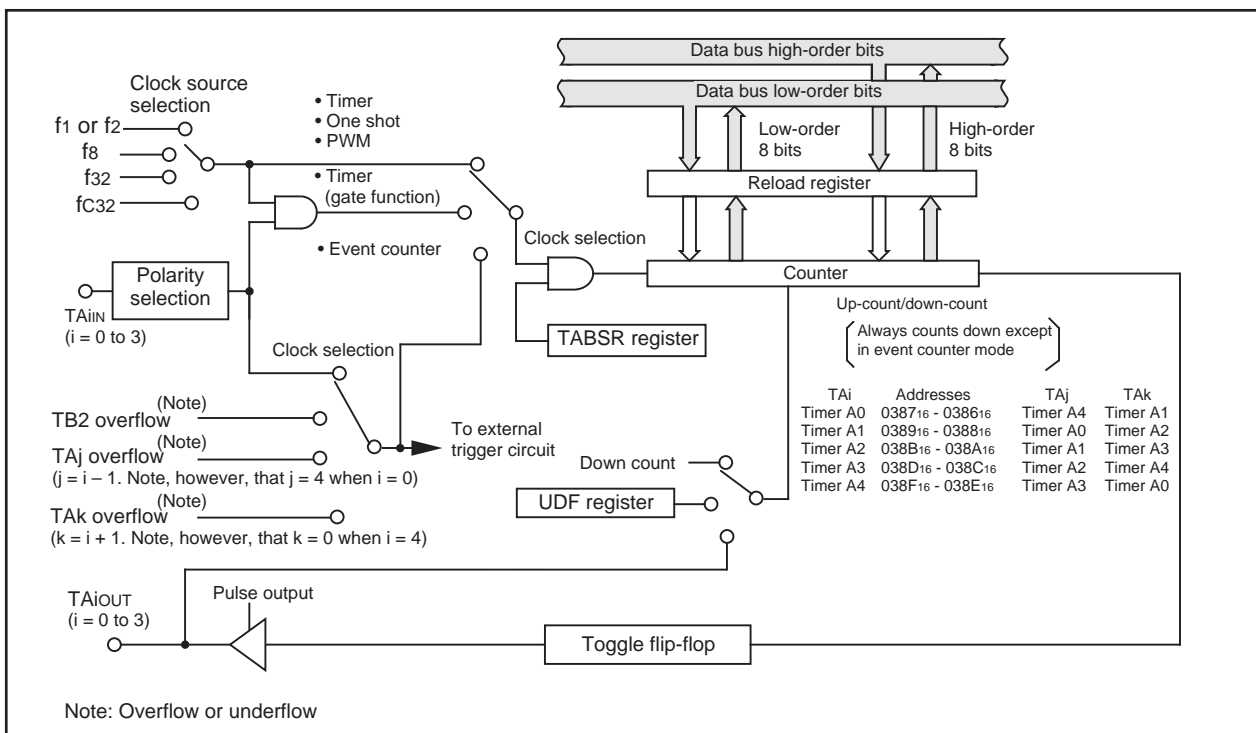


Figure 10.3. Timer A Block Diagram

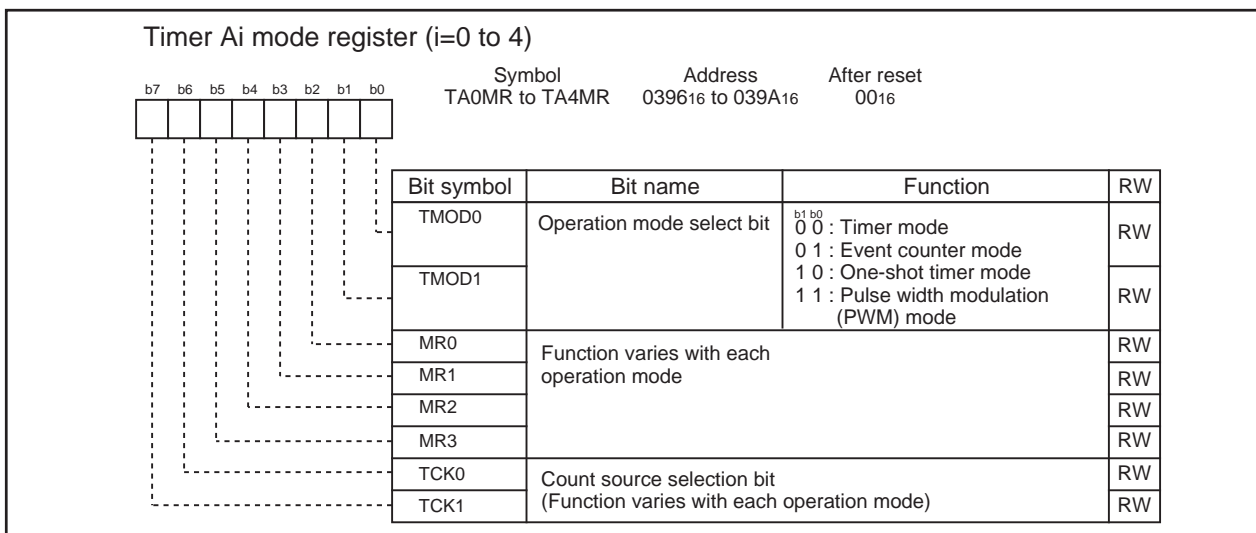


Figure 10.4. TA0MR to TA4MR Registers

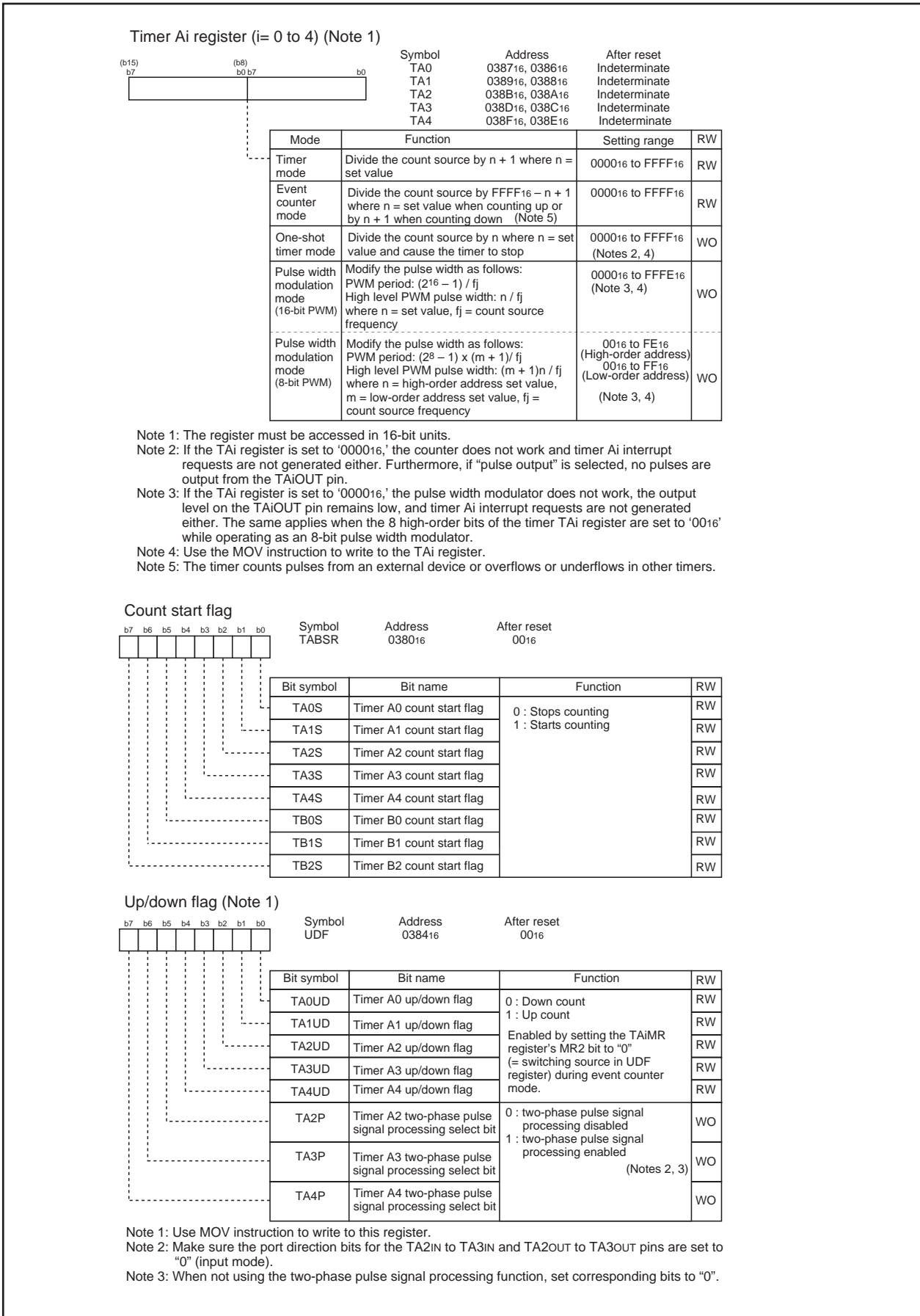


Figure 10.5. TA0 to TA4 Registers, TABSR Register, and UDF Register

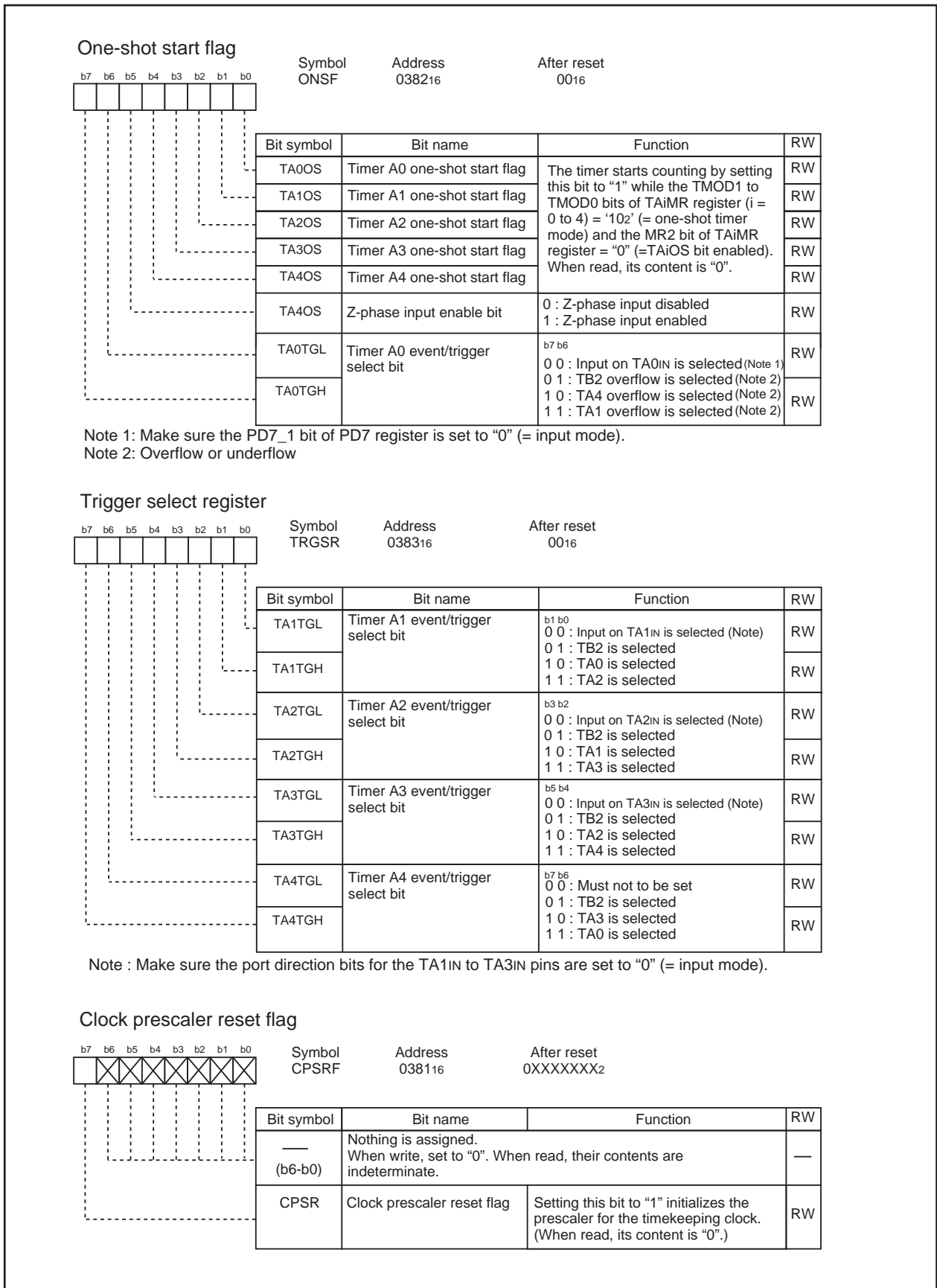


Figure 10.6. ONSF Register, TRGSR Register, and CPSRF Register

1. Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 10.1). Figure 10.7 shows TAIiMR register in timer mode.

Table 10.1. Specifications in Timer Mode

Item	Specification
Count source	f1, f2, f8, f32, fC32
Count operation	<ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TAIiMR register (i= 0 to 4) 0000 ₁₆ to FFFF ₁₆
Count start condition	Set TAIiS bit of TABSR register to "1" (= start counting)
Count stop condition	Set TAIiS bit to "0" (= stop counting)
Interrupt request generation timing	At underflow
TAiIN pin function	I/O port or gate input
TAiOUT pin function	I/O port or pulse output
Read from timer	Count value can be read by reading TAIi register
Write to timer	<ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to TAIi register is written to both reload register and counter When counting (after 1st count source input) Value written to TAIi register is written to only reload register (Transferred to counter when reloaded next)
Select function	<ul style="list-style-type: none"> Gate function Counting can be started and stopped by an input signal to TAIiIN pin Pulse output function Whenever the timer underflows, the output polarity of TAIiOUT pin is inverted. While the TAIiS bit is set to "0", the pin outputs an "L" level signal during the count stop.

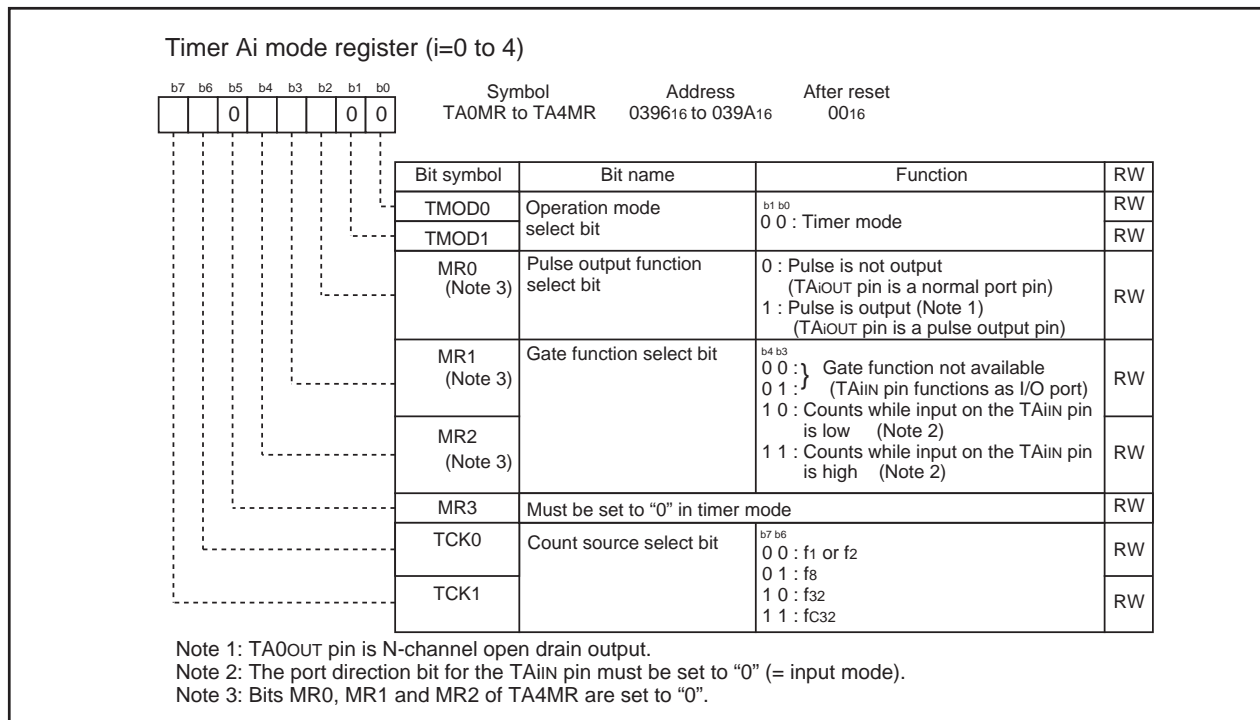


Figure 10.7. TAIiMR Register in Timer Mode

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 10.2 lists specifications in event counter mode (when not processing two-phase pulse signal). Figure 10.8 shows TAI_{MR} register in event counter mode (when not processing two-phase pulse signal).

Table 10.2. Specifications in Event Counter Mode (when not processing two-phase pulse signal)

Item	Specification
Count source	<ul style="list-style-type: none"> External signals input to TAI_{IN} pin (i=0 to 3) (effective edge can be selected in program) Timer B2 overflows or underflows, timer A_j (j=i-1, except j=4 if i=0) overflows or underflows, timer A_k (k=i+1, except k=0 if i=4) overflows or underflows
Count operation	<ul style="list-style-type: none"> Up-count or down-count can be selected by external signal or program When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divided ratio	$1 / (FFFF_{16} - n + 1)$ for up-count $1 / (n + 1)$ for down-count n : set value of TAI register 0000 ₁₆ to FFFF ₁₆
Count start condition	Set TAI _S bit of TABSR register to "1" (= start counting)
Count stop condition	Set TAI _S bit to "0" (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAI _{IN} pin function	I/O port or count source input
TAI _{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"> When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)
Select function	<ul style="list-style-type: none"> Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it Pulse output function Whenever the timer underflows or underflows, the output polarity of TAI_{OUT} pin is inverted . When not counting, the pin outputs a low.

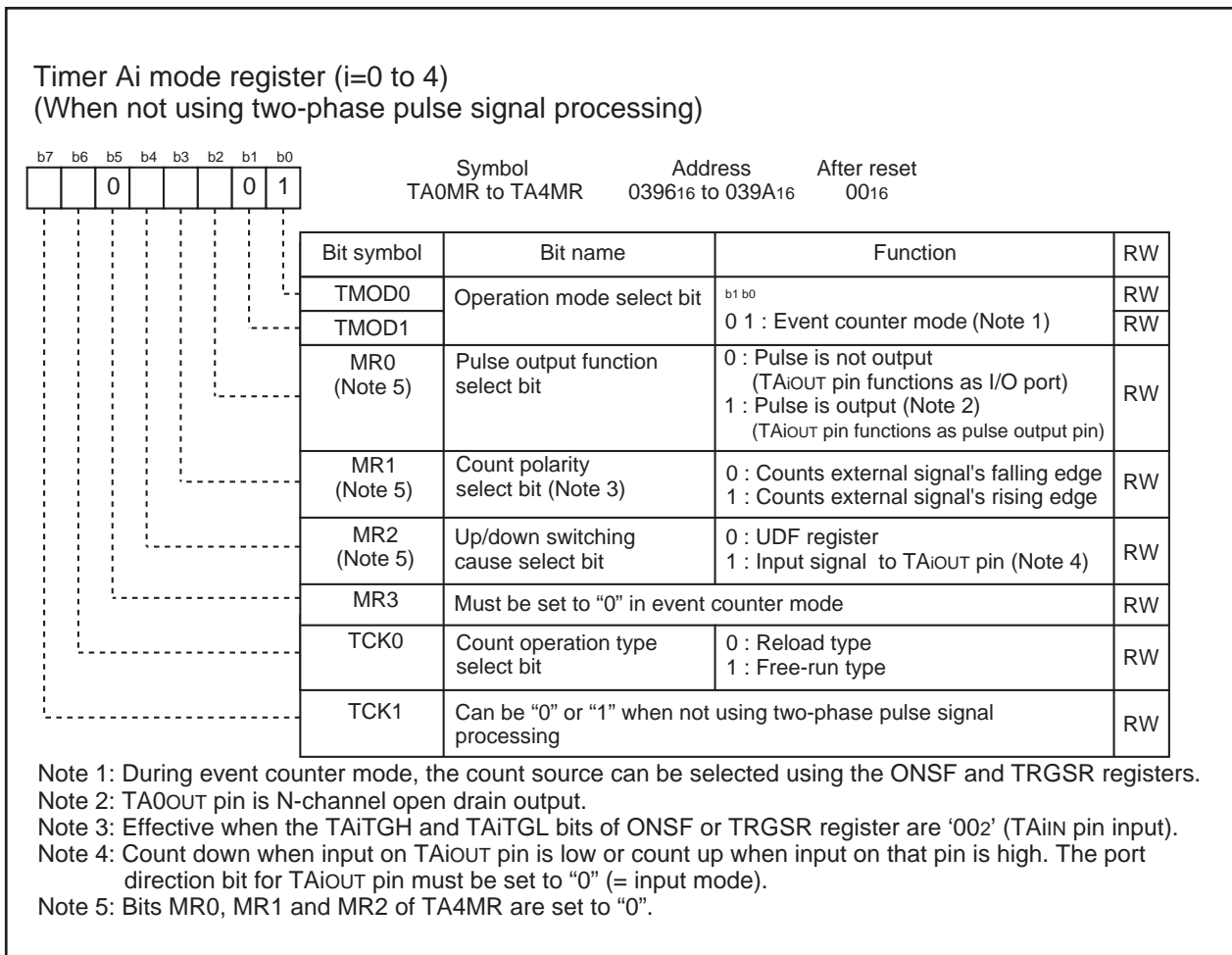
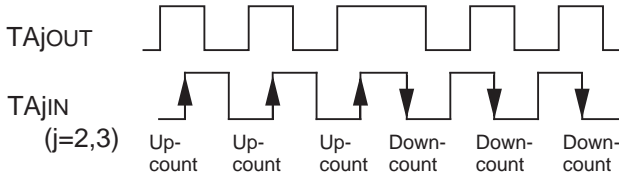
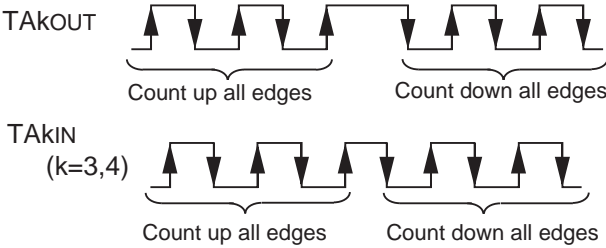


Figure 10.8. TAI_iMR Register in Event Counter Mode (when not using two-phase pulse signal processing)

Table 10.3 lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

Figure 10.9 shows TA2MR to TA4MR registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

Table 10.3. 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 3)
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 ₁₆ - n + 1) for up-count 1/ (n + 1) for down-count n : set value of TAI register 0000 ₁₆ to FFFF ₁₆
Count start condition	Set TAI _i S bit of TABSR register to "1" (= start counting)
Count stop condition	Set TAI _i S 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"> 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".  <ul style="list-style-type: none"> Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAKIN(k=3) 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.  <ul style="list-style-type: none"> Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input.

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

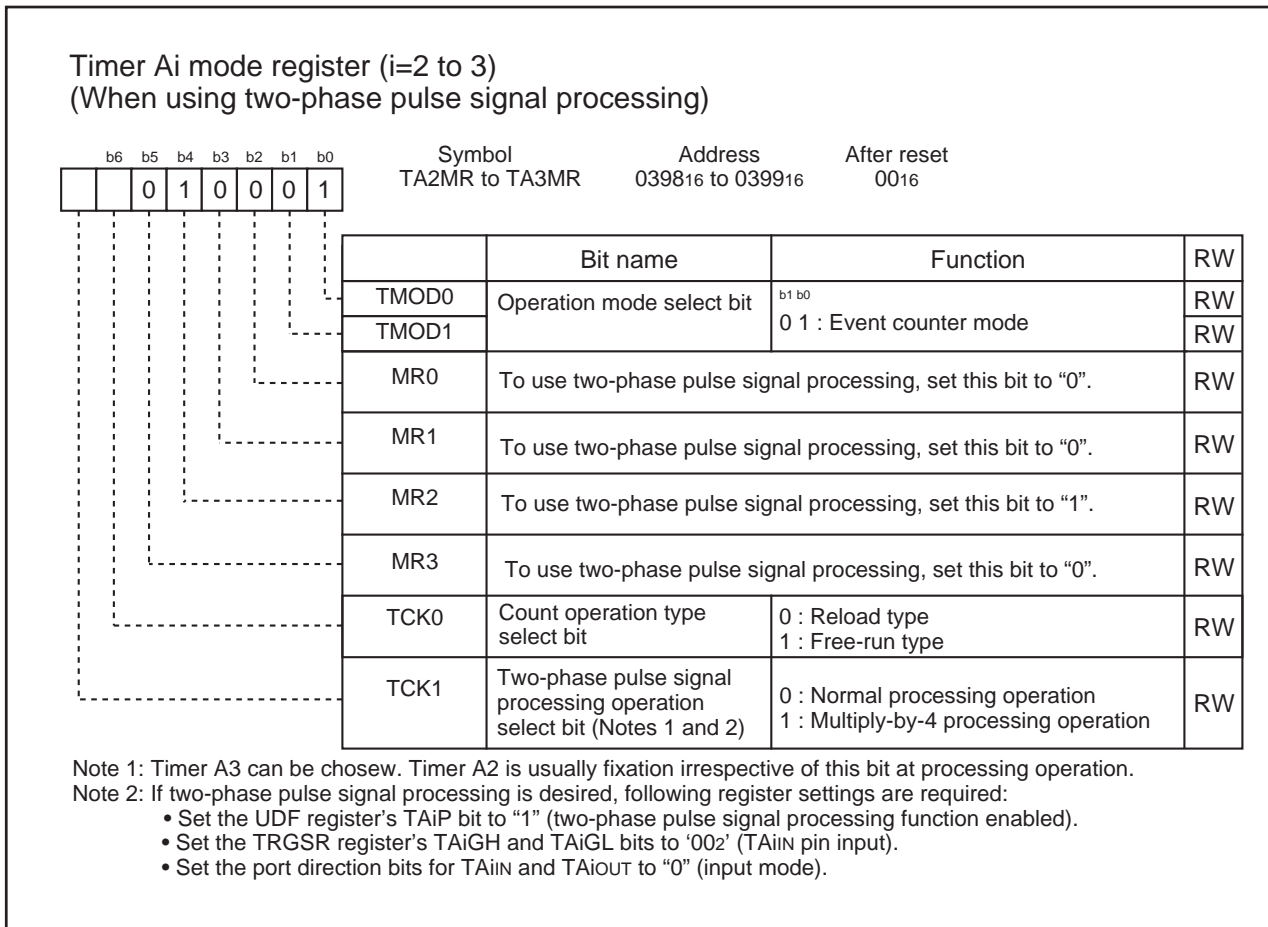


Figure 10.9. TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2 or A3)

• 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 “0000₁₆” 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 10.10 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.

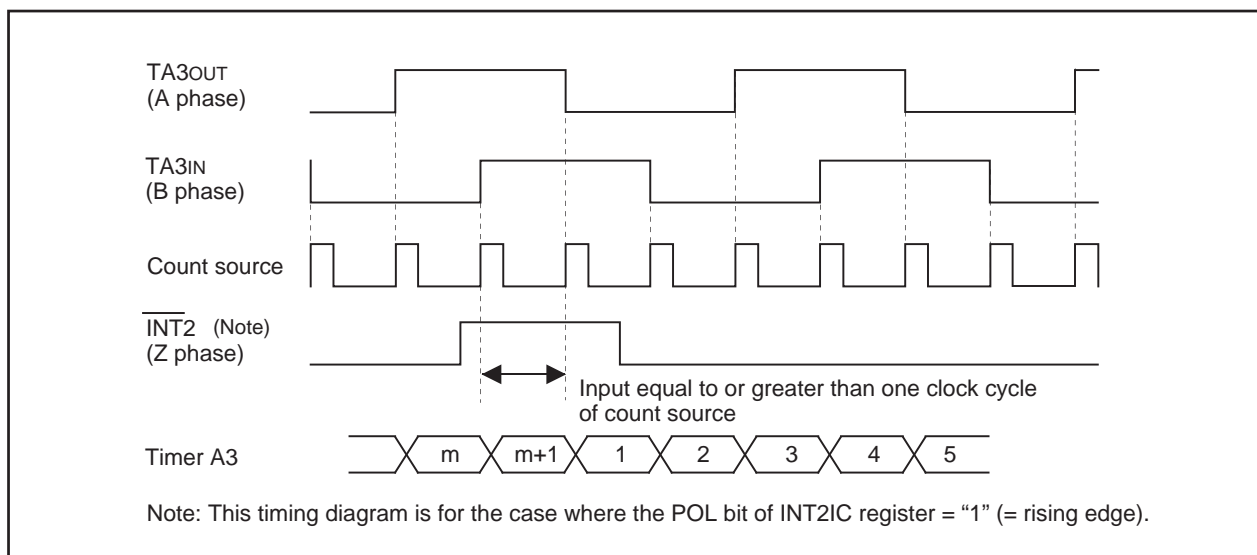


Figure 10.10. Two-phase Pulse (A phase and B phase) and the Z Phase

3. One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. (See Table 10.4.) When the trigger occurs, the timer starts up and continues operating for a given period. Figure 10.11 shows the TAIMR register in one-shot timer mode.

Table 10.4. Specifications in One-shot Timer Mode

Item	Specification
Count source	f1, f2, f8, f32, fC32
Count operation	<ul style="list-style-type: none"> Down-count When the counter reaches 0000₁₆, it stops counting after reloading a new value If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	$1/n$ n : set value of TAI register 0000 ₁₆ to FFFF ₁₆ However, the counter does not work if the divide-by-n value is set to 0000 ₁₆ .
Count start condition	TAI _S bit of TABSR register = "1" (start counting) and one of the following triggers occurs. <ul style="list-style-type: none"> External trigger input from the TAI_{IN} pin Timer B2 overflow or underflow, timer A_j (j=i-1, except j=4 if i=0) overflow or underflow, timer A_k (k=i+1, except k=0 if i=4) overflow or underflow The TAI_{OS} bit of ONSF register is set to "1" (= timer starts)
Count stop condition	<ul style="list-style-type: none"> When the counter is reloaded after reaching "0000₁₆" TAI_S bit is set to "0" (= stop counting)
Interrupt request generation timing	When the counter reaches "0000 ₁₆ "
TAI _{IN} pin function	I/O port or trigger input
TAI _{OUT} 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"> When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)
Select function	<ul style="list-style-type: none"> Pulse output function The timer outputs a low when not counting and a high when counting.

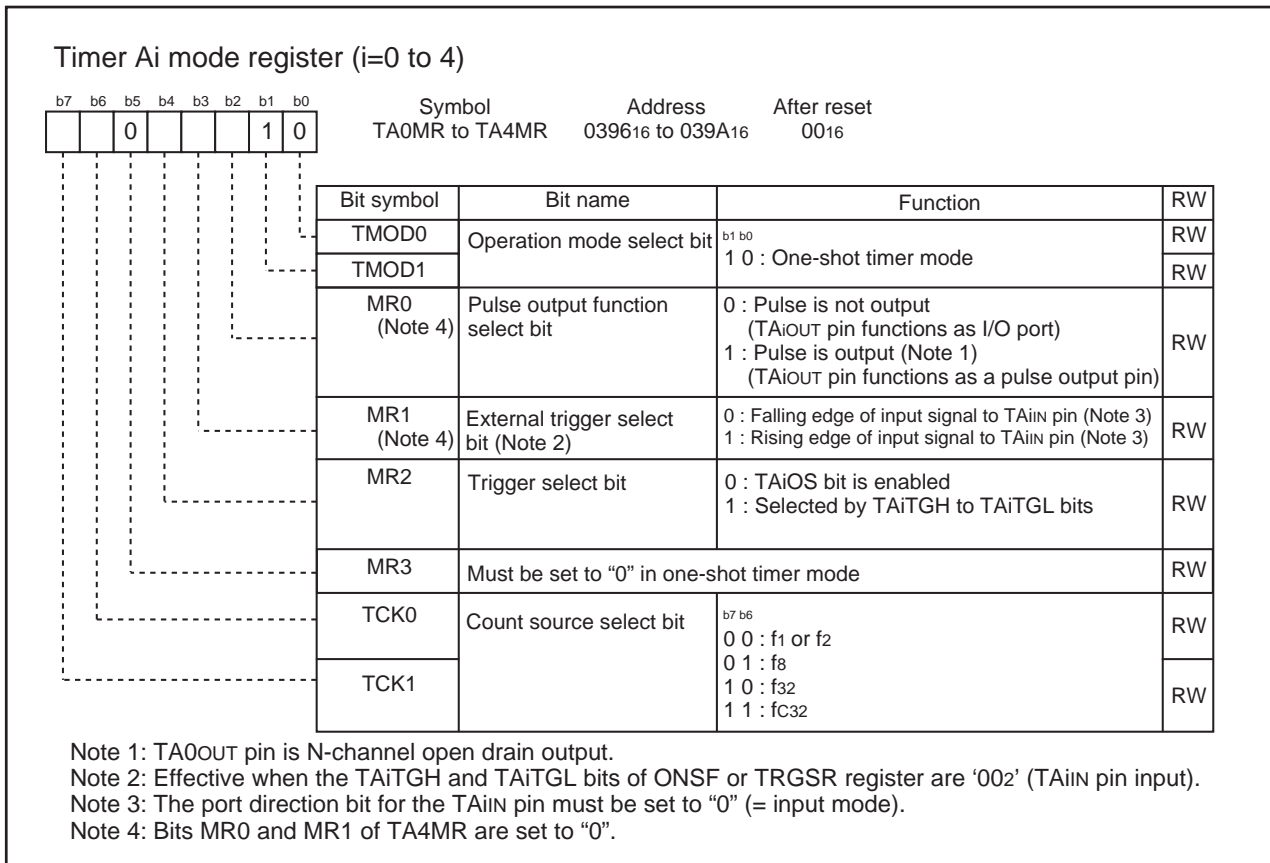


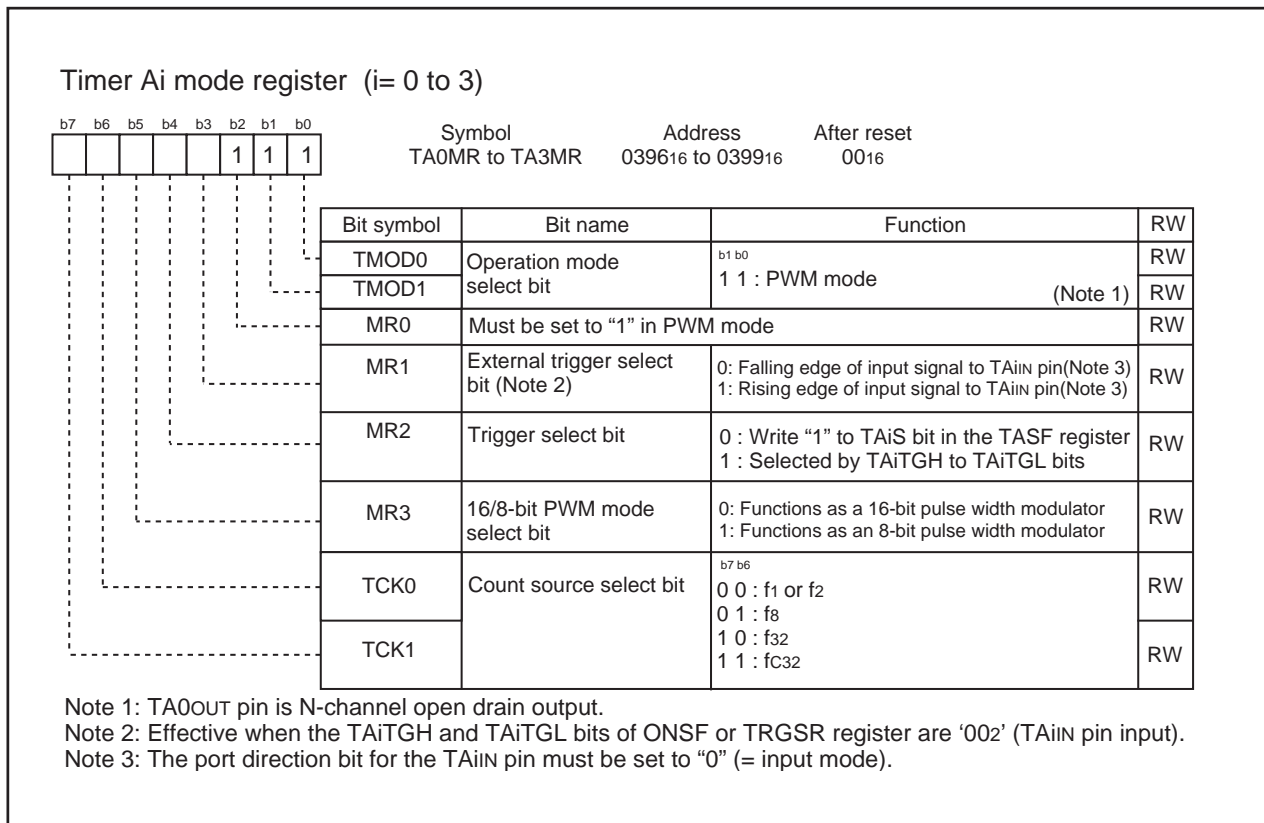
Figure 10.11. TAIiMR Register in One-shot Timer Mode

4. Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see Table 10.5). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 10.12 shows TAI_{MR} register in pulse width modulation mode. Figures 10.13 and 10.14 show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates.

Table 10.5. Specifications in PWM Mode

Item	Specification
Count source	f ₁ , f ₂ , f ₈ , f ₃₂ , f _{C32}
Count operation	<ul style="list-style-type: none"> • Down-count (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new value at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs during counting
16-bit PWM	<ul style="list-style-type: none"> • High level width n / f_j n: set value of TAI register ($i=0$ to 3) • Cycle time $(2^{16}-1) / f_j$ fixed f_j: count source frequency (f₁, f₂, f₈, f₃₂, f_{C32})
8-bit PWM	<ul style="list-style-type: none"> • High level width $n \times (m+1) / f_j$ n: set value of TAI_{MR} register high-order address • Cycle time $(2^8-1) \times (m+1) / f_j$ m: set value of TAI_{MR} register low-order address
Count start condition	<ul style="list-style-type: none"> • TAI_S bit of TABSR register is set to "1" (= start counting) • The TAI_S bit = 1 and external trigger input from the TAI_{IN} pin • The TAI_S bit = 1 and one of the following external triggers occurs • Timer B2 overflow or underflow, timer A_j ($j=i-1$, except $j=4$ if $i=0$) overflow or underflow, timer A_k ($k=i+1$) overflow or underflow
Count stop condition	TAI _S bit is set to "0" (= stop counting)
Interrupt request generation timing	PWM pulse goes "L"
TAI _{IN} pin function	I/O port or trigger input
TAI _{OUT} 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"> • When not counting and until the 1st count source is input after counting start Value written to TAI register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAI register is written to only reload register (Transferred to counter when reloaded next)

Figure 10.12. TAI_{MR} Register in PWM Mode

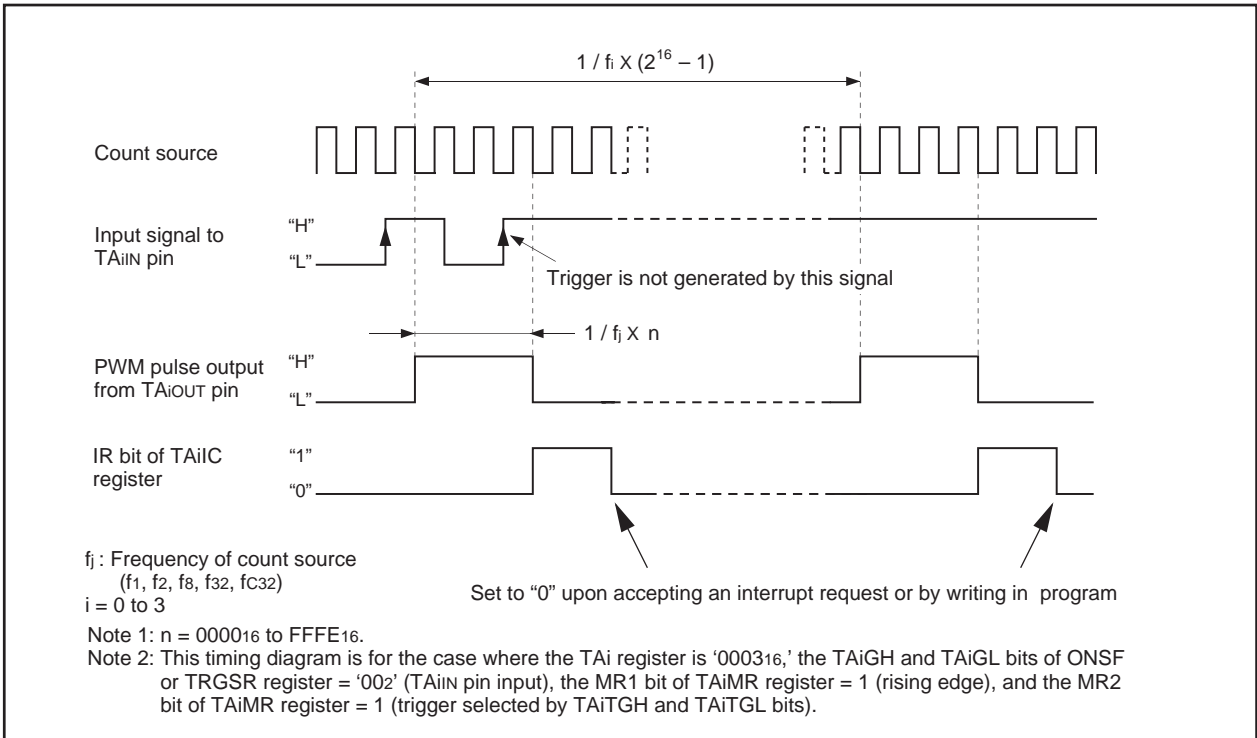


Figure 10.13. Example of 16-bit Pulse Width Modulator Operation

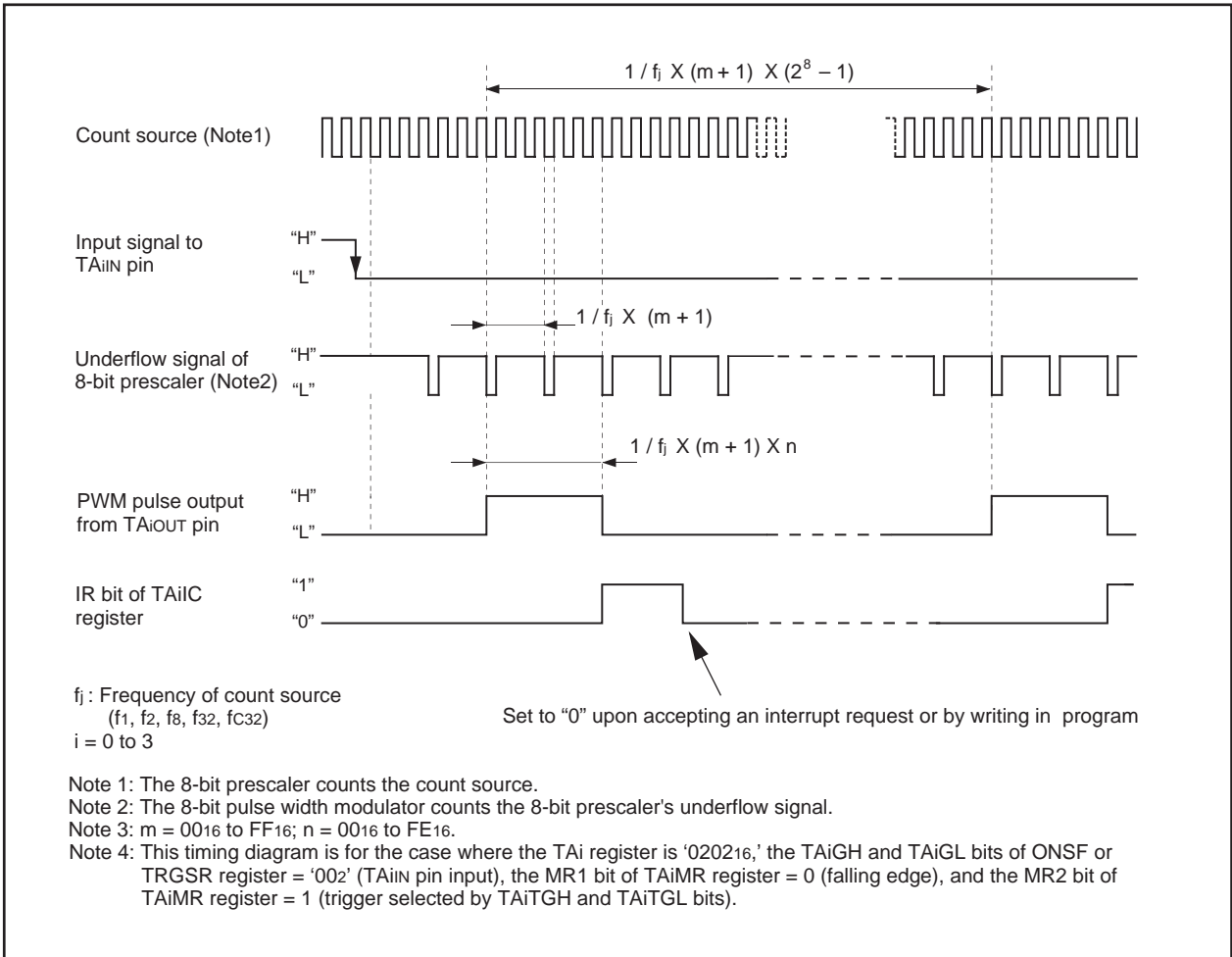


Figure 10.14. Example of 8-bit Pulse Width Modulator Operation

Timer B

Figure 10.15 shows a block diagram of the timer B. Figures 10.16 and 10.17 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.

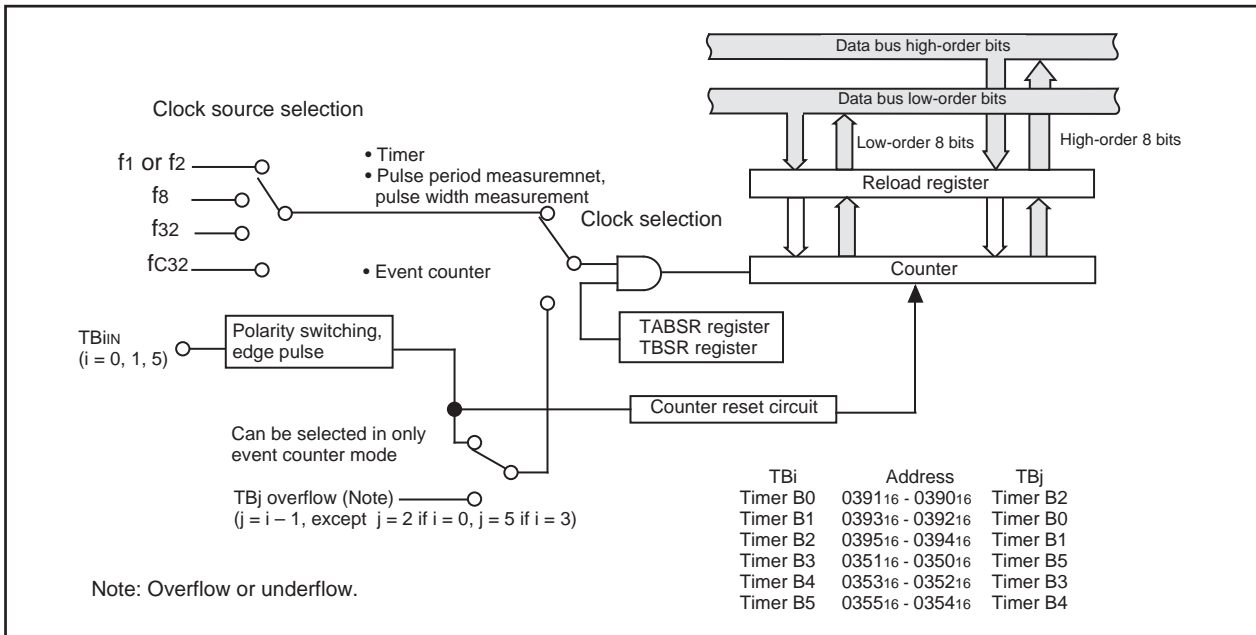


Figure 10.15. Timer B Block Diagram

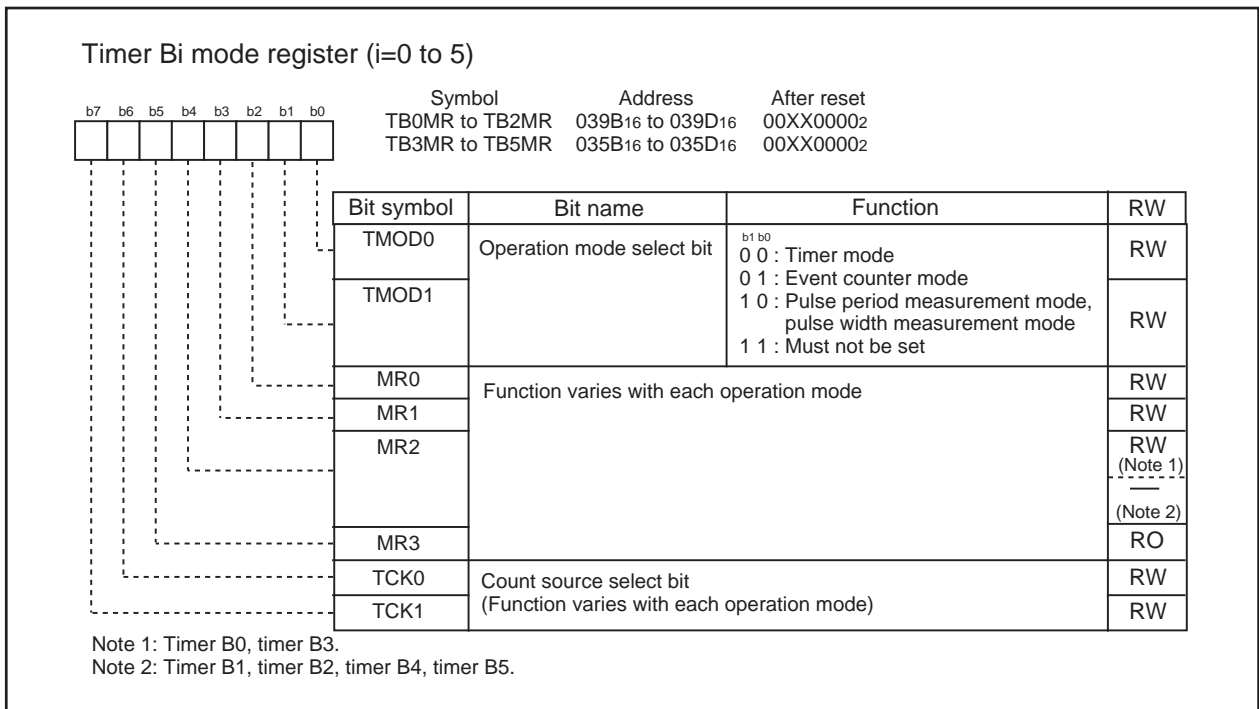


Figure 10.16. TB0MR to TB5MR Registers

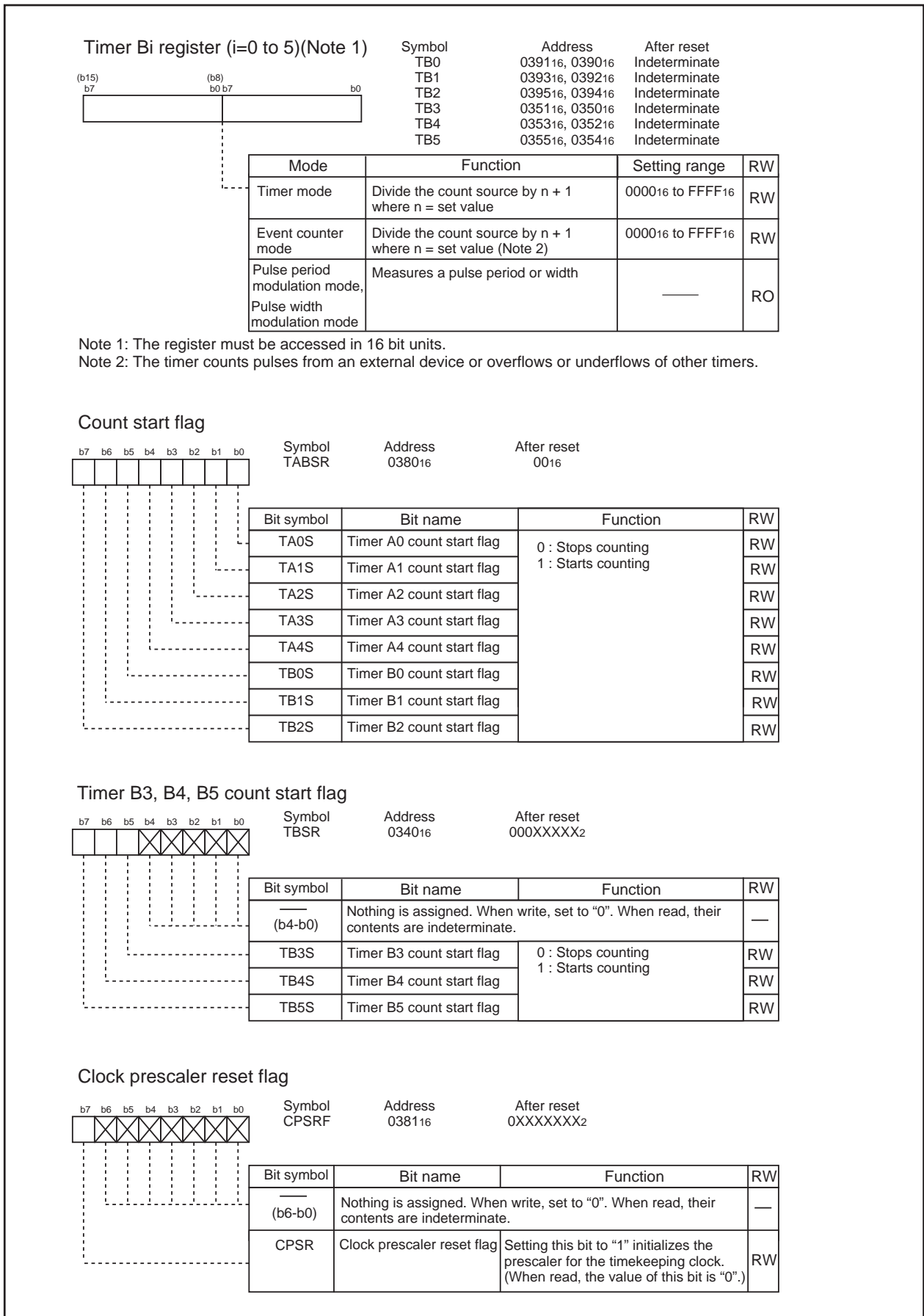


Figure 10.17. TB0 to TB5 Registers, TABSR Register, TBSR Register, CPSRF Register

1. Timer Mode

In timer mode, the timer counts a count source generated internally (see Table 10.6). Figure 10.18 shows TBiMR register in timer mode.

Table 10.6. Specifications in Timer Mode

Item	Specification
Count source	f1, f2, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TB register (i= 0 to 5) 0000 ₁₆ to FFFF ₁₆
Count start condition	Set TBiS bit ^(Note) 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"> When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

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.

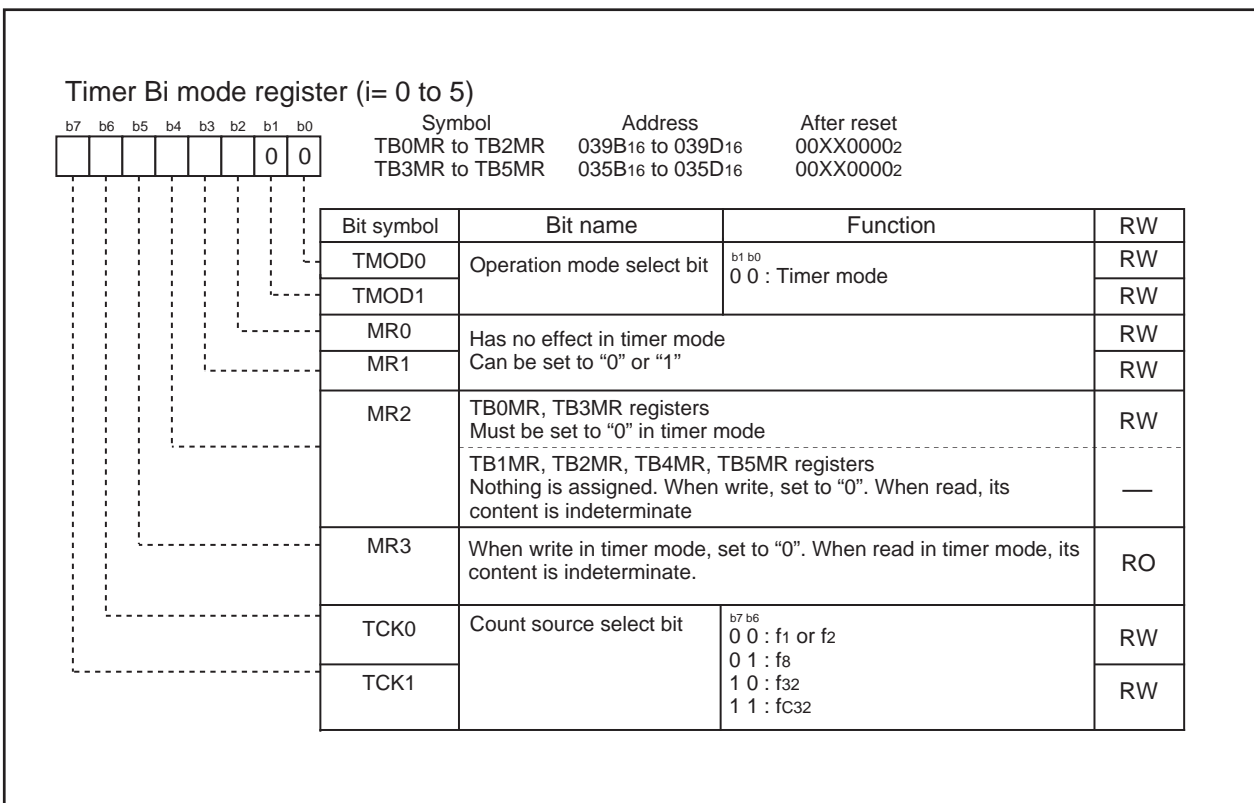


Figure 10.18. TBiMR Register in Timer Mode

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 10.7). Figure 10.19 shows TBiMR register in event counter mode.

Table 10.7. Specifications in Event Counter Mode

Item	Specification
Count source	<ul style="list-style-type: none"> External signals input to TBiIN pin ($i=0, 1, 5$) (effective edge can be selected in program) Timer Bj overflow or underflow ($j=i-1$, except $j=2$ if $i=0$, $j=5$ if $i=3$)
Count operation	<ul style="list-style-type: none"> Down-count When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	$1/(n+1)$ n: set value of TBi register 0000 ₁₆ to FFFF ₁₆
Count start condition	Set TBiS bit ¹ to "1" (= start counting)
Count stop condition	Set TBiS bit to "0" (= stop counting)
Interrupt request generation timing	Timer underflow
TBiIN pin function	Count source input
Read from timer	Count value can be read by reading TBi register
Write to timer	<ul style="list-style-type: none"> When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

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.

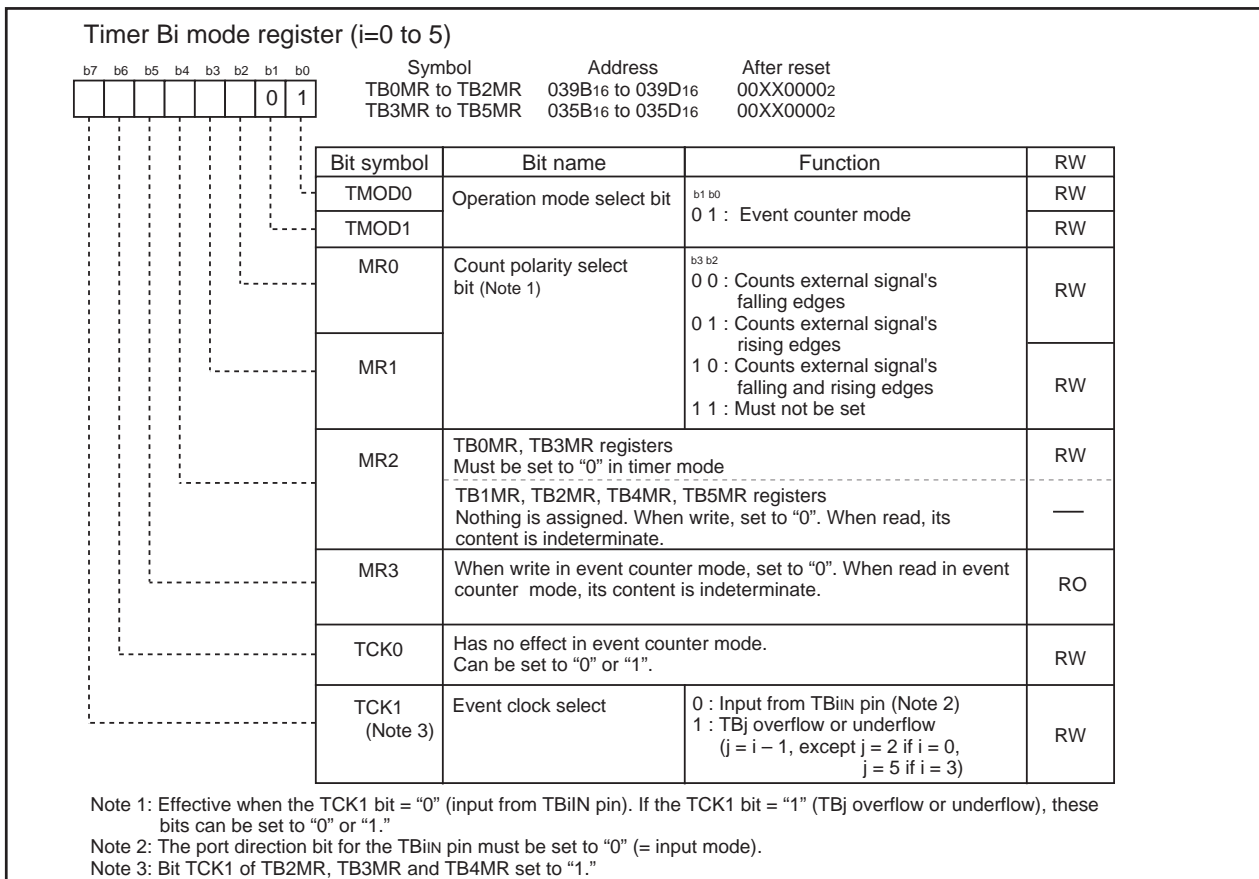


Figure 10.19. TBiMR Register in Event Counter Mode

3. Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal (see Table 10.8). Figure 10.20 shows TBiMR register in pulse period and pulse width measurement mode. Figure 10.21 shows the operation timing when measuring a pulse period. Figure 10.22 shows the operation timing when measuring a pulse width.

Table 10.8. 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"> • Up-count • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to "000016" to continue counting.
Count start condition	Set TBiS (i=0, 1, 5) bit (Note 3) to "1" (= start counting)
Count stop condition	Set TBiS bit to "0" (= stop counting)
Interrupt request generation timing	<ul style="list-style-type: none"> • When an effective edge of measurement pulse is input (Note 1) • 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).
TBiIN pin function	Measurement pulse input
Read from timer	Contents of the reload register (measurement result) can be read by reading TBi register (Note 2)
Write to timer	Value written to TBi register is written to neither reload register nor counter

Notes:

1. Interrupt request is not generated when the first effective edge is input after the timer started counting.
2. Value read from TBi register is indeterminate until the second valid edge is input after the timer starts counting.
3. The TB0S to TB1S bits are assigned to the TABSR register bit 5 to bit 6, and the TB5S bit is assigned to the TBSR register bit 7.

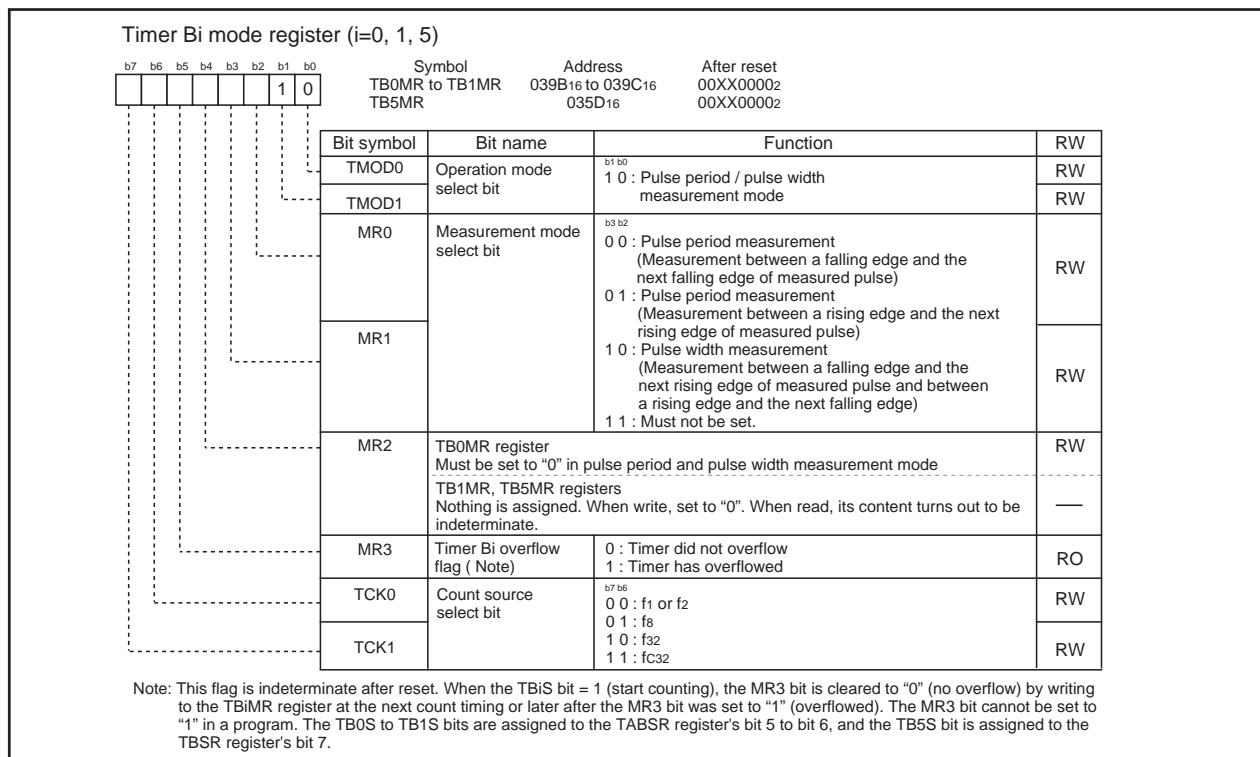


Figure 10.20. TBiMR Register in Pulse Period and Pulse Width Measurement Mode

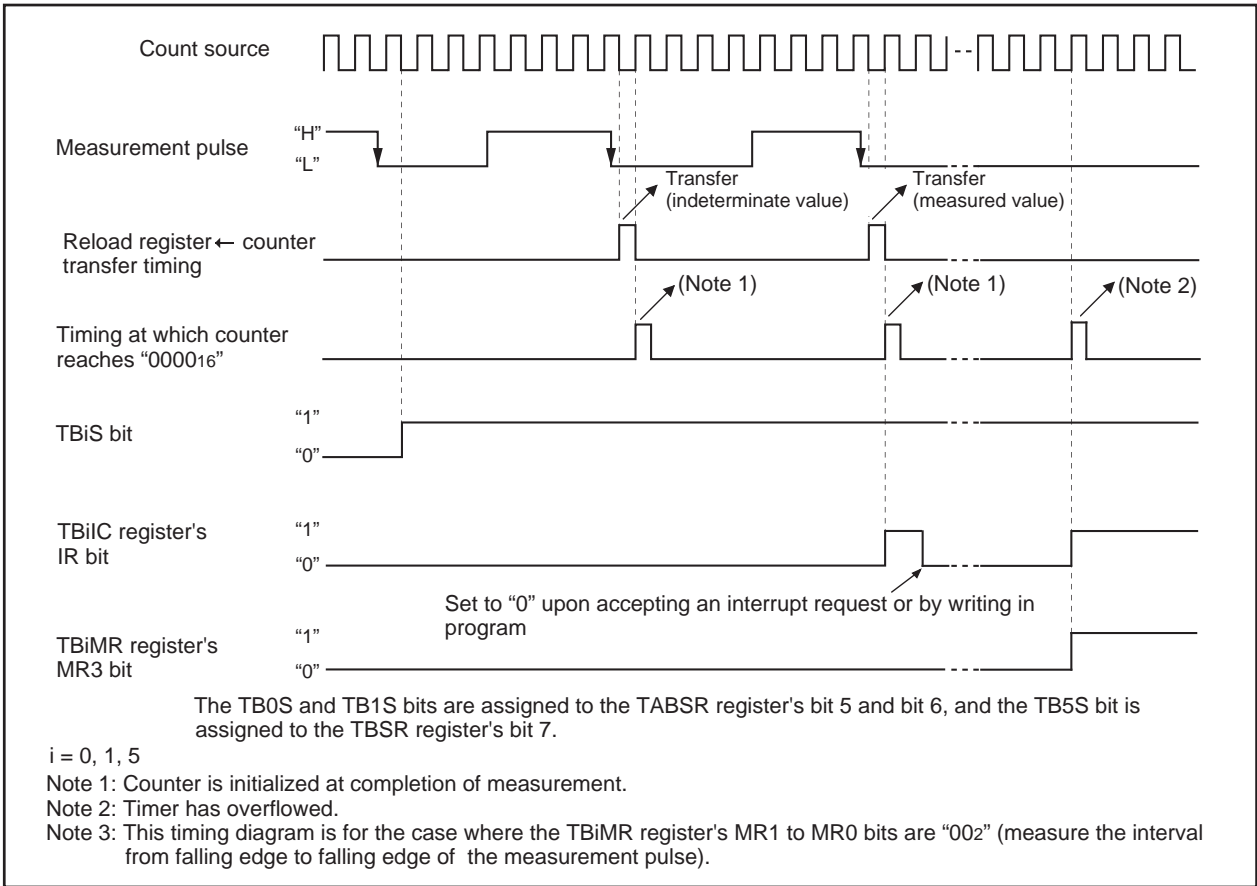


Figure 10.21. Operation timing when measuring a pulse period

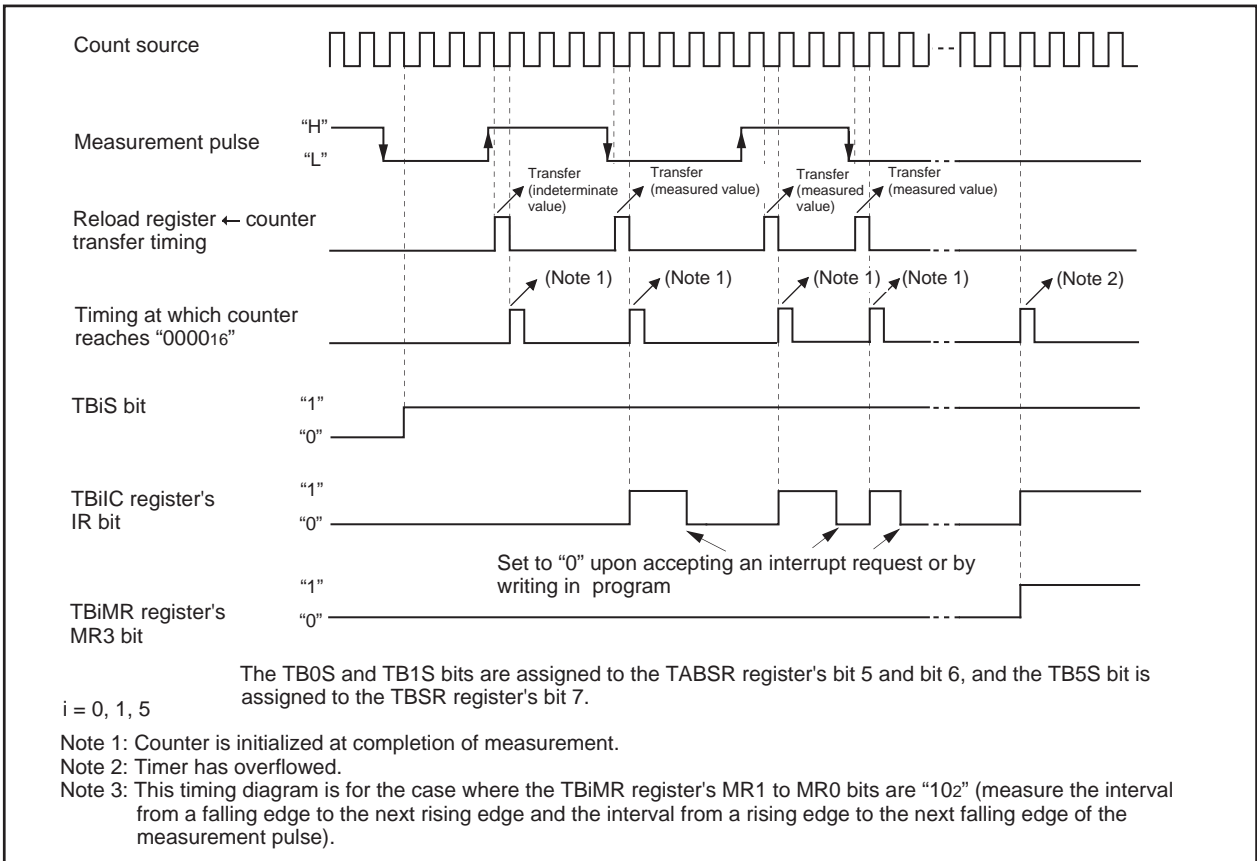


Figure 10.22. Operation timing when measuring a pulse width

Serial I/O

Serial I/O is configured with 3 channels: UART0 to UART2.

Each is explained below.

UARTi (i=0 to 2)

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

Figure 11.1 shows the block diagram of UARTi. Figure 11.2 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 2
- Special mode 1 (Bus collision detection function, IE mode) : UART0, UART1

Figures 11.3 to 11.8 show the UARTi-related registers.

Refer to tables listing each mode for register setting.

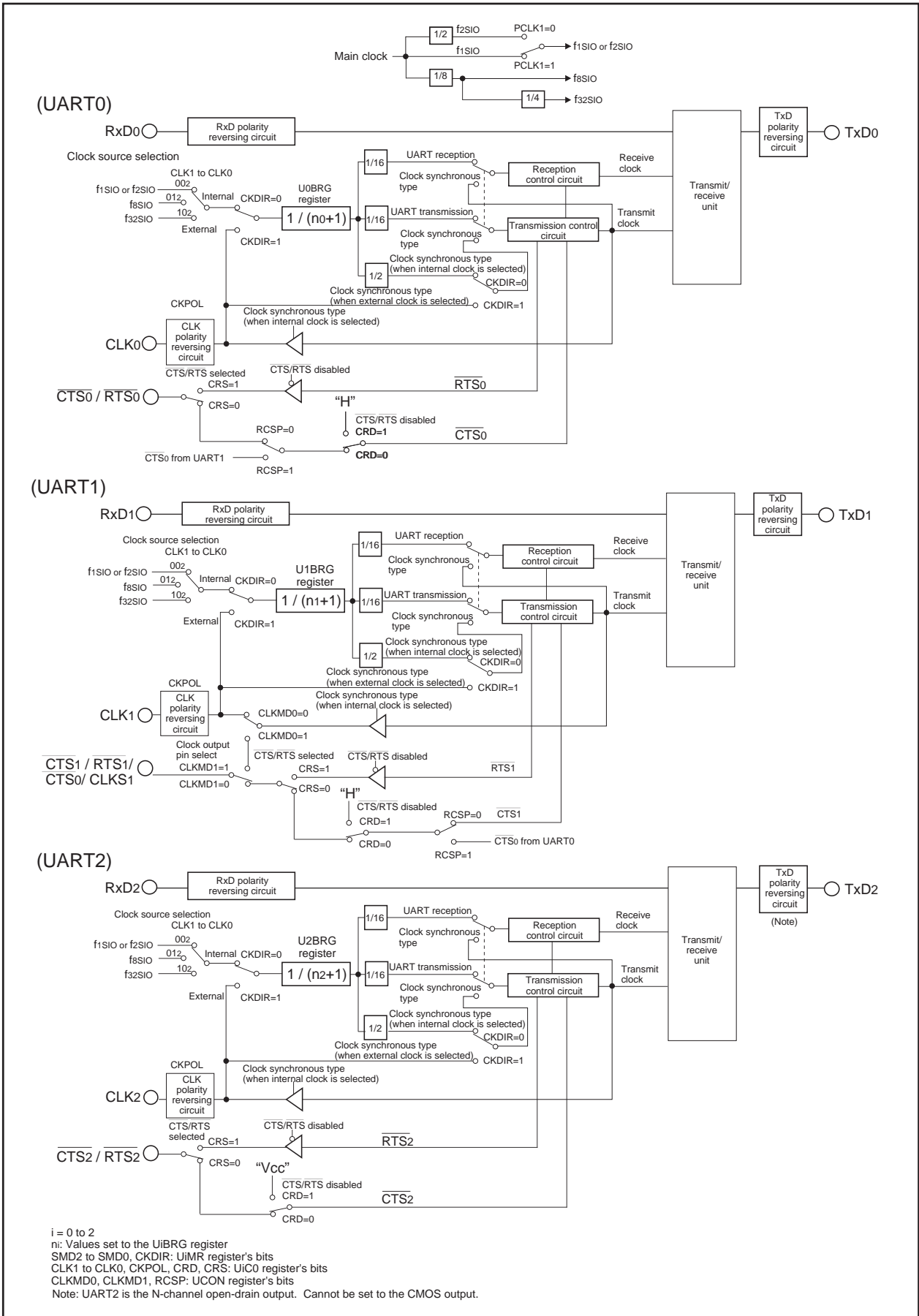


Figure 11.1. UARTi Block Diagram

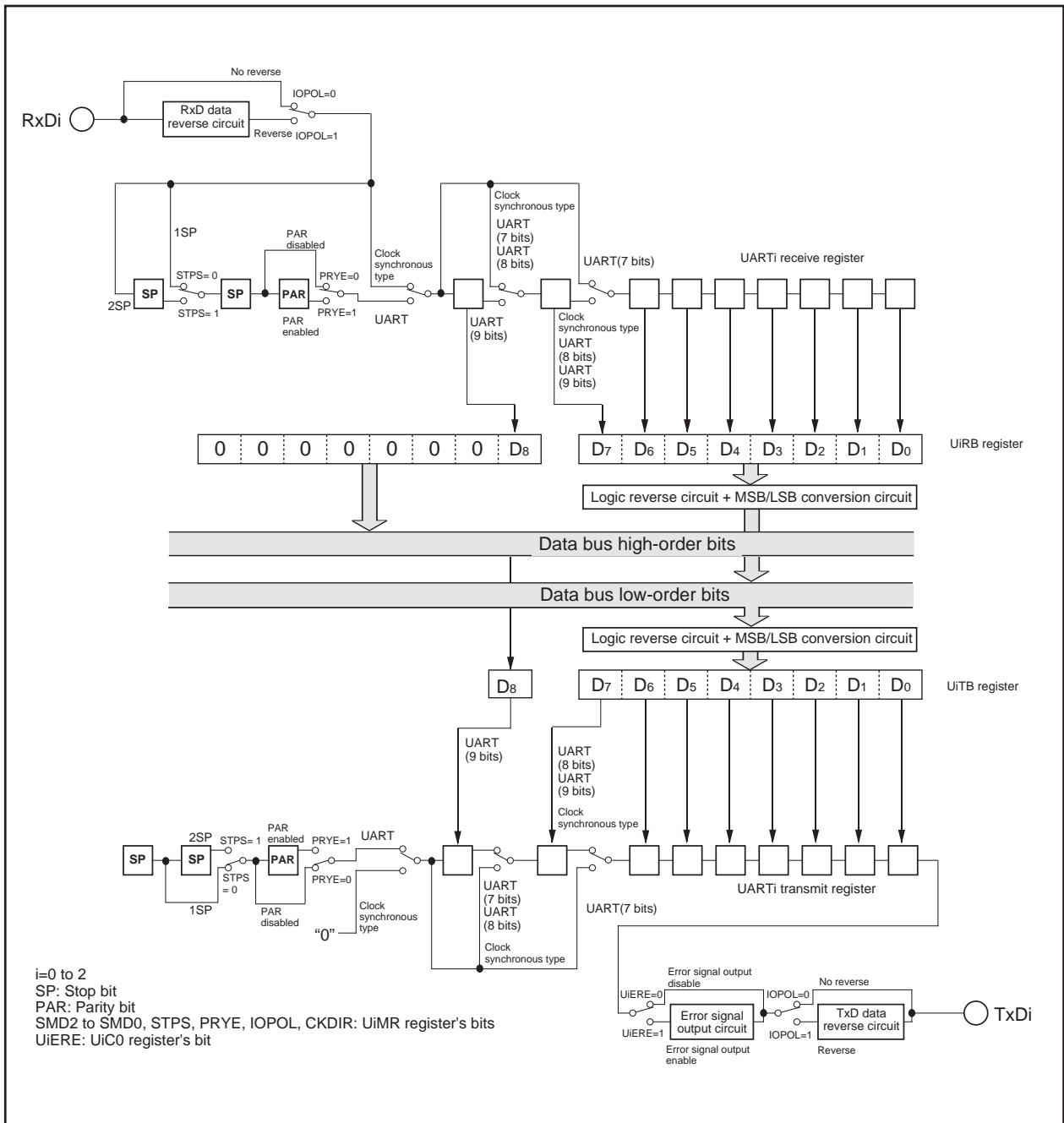


Figure 11.2. UARTi Transmit/Receive Unit

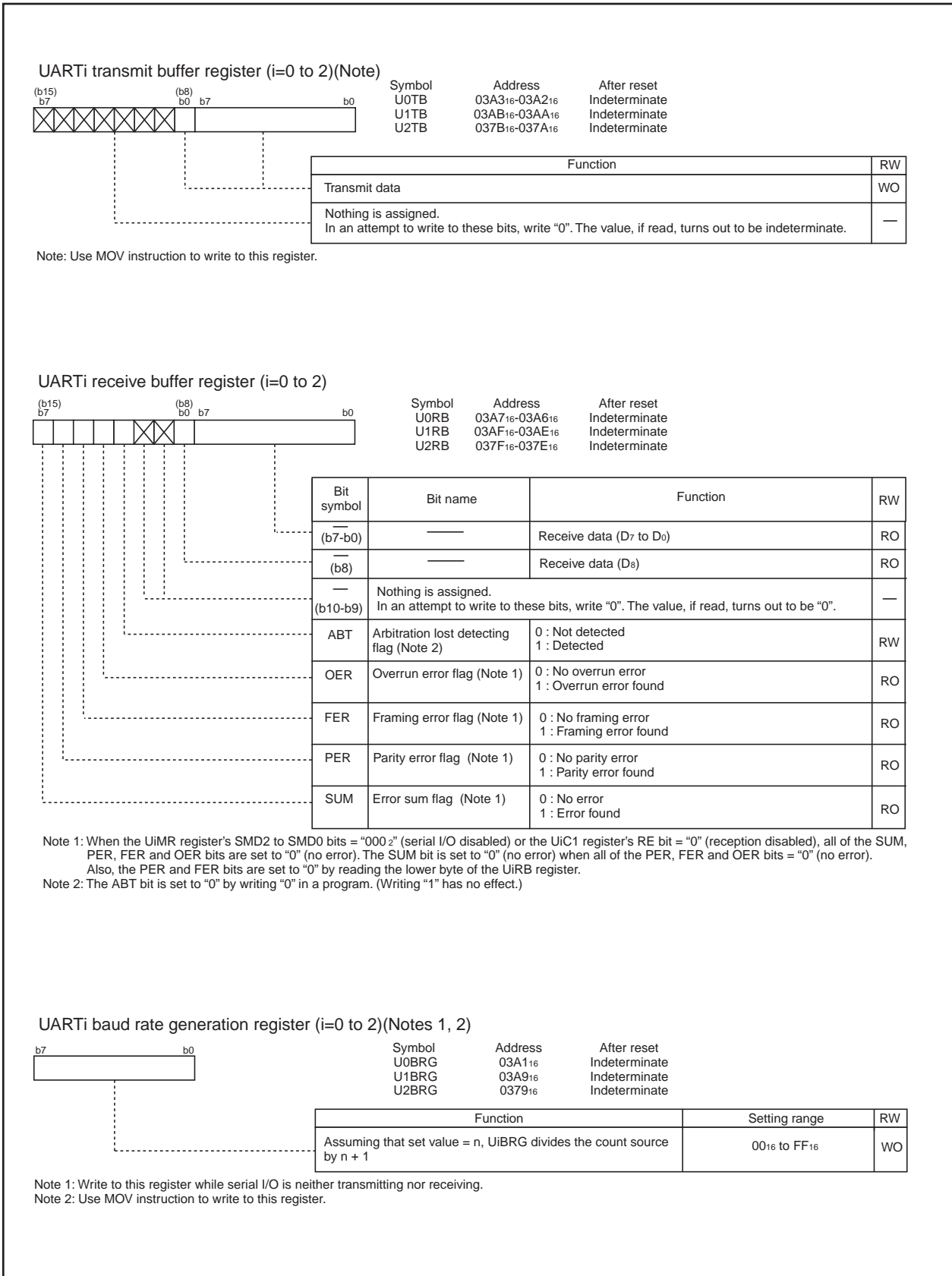


Figure 11.3. U0TB to U2TB Register, U0RB to U2RB Register, and U0BRG to U2BRG Register

UARTi transmit/receive mode register (i=0 to 2)

Bit symbol	Bit name	Function	RW																
<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">b7</td><td style="text-align: center;">b6</td><td style="text-align: center;">b5</td><td style="text-align: center;">b4</td><td style="text-align: center;">b3</td><td style="text-align: center;">b2</td><td style="text-align: center;">b1</td><td style="text-align: center;">b0</td> </tr> <tr> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> </tr> </table>				b7	b6	b5	b4	b3	b2	b1	b0								
b7	b6	b5	b4	b3	b2	b1	b0												
		Symbol	Address	After reset															
		U0MR to U2MR	03A0 ₁₆ , 03A8 ₁₆ , 0378 ₁₆	00 ₁₆															
SMD0	Serial I/O mode select bit (Note 2)	b2 b1 b0 0 0 0 : Serial I/O disabled 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : Must not be set 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 CLKi pin to "0" (input mode).
 Note 2: To receive data, set the corresponding port direction bit for each RxDi pin to "0" (input mode).

UARTi transmit/receive control register 0 (i=0 to 2)

Bit symbol	Bit name	Function	RW																
<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">b7</td><td style="text-align: center;">b6</td><td style="text-align: center;">b5</td><td style="text-align: center;">b4</td><td style="text-align: center;">b3</td><td style="text-align: center;">b2</td><td style="text-align: center;">b1</td><td style="text-align: center;">b0</td> </tr> <tr> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> </tr> </table>				b7	b6	b5	b4	b3	b2	b1	b0								
b7	b6	b5	b4	b3	b2	b1	b0												
		Symbol	Address	After reset															
		U0C0 to U2C0	03A4 ₁₆ , 03AC ₁₆ , 037C ₁₆	00001000 ₂															
CLK0	BRG count source select bit	b1 b0 0 0 : f1SIO or f2SIO is selected 0 1 : f8SIO is selected 1 0 : f32SIO 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 ₀ , P6 ₄ and P7 ₃ can be used as I/O ports)		RW															
NCH	Data output select bit (Note 2)	0 : TxDi/SDAi and SCLi pins are CMOS output 1 : TxDi/SDAi and SCLi pins are N-channel open-drain output		RW															
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge		RW															
UFORM	Transfer format select bit (Note 3)	0 : LSB first 1 : MSB first		RW															

Note 1: Set the corresponding port direction bit for each CTSi pin to "0" (input mode).
 Note 2: TxD2 is 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: CTS1/RTS1 can be used when the UCON register's CLKMD1 bit = "0" (only CLK1 output) and the UCON register's RCSP bit = "0" (CTS0/RTS0 not separated).

Figure 11.4. U0MR to U2MR Register and U0C0 to U2C0 Register

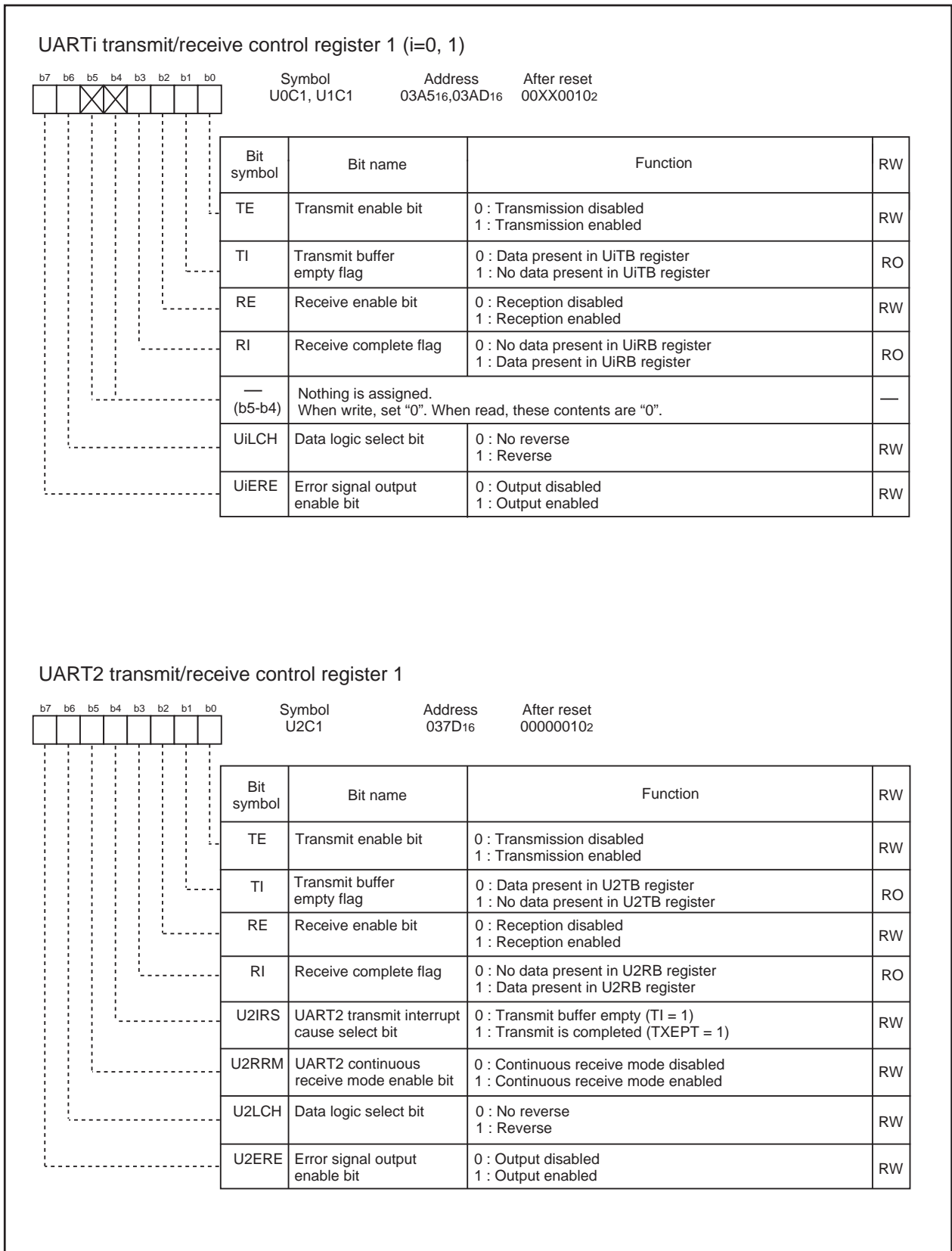


Figure 11.5. U0C1 to U2C1 Registers

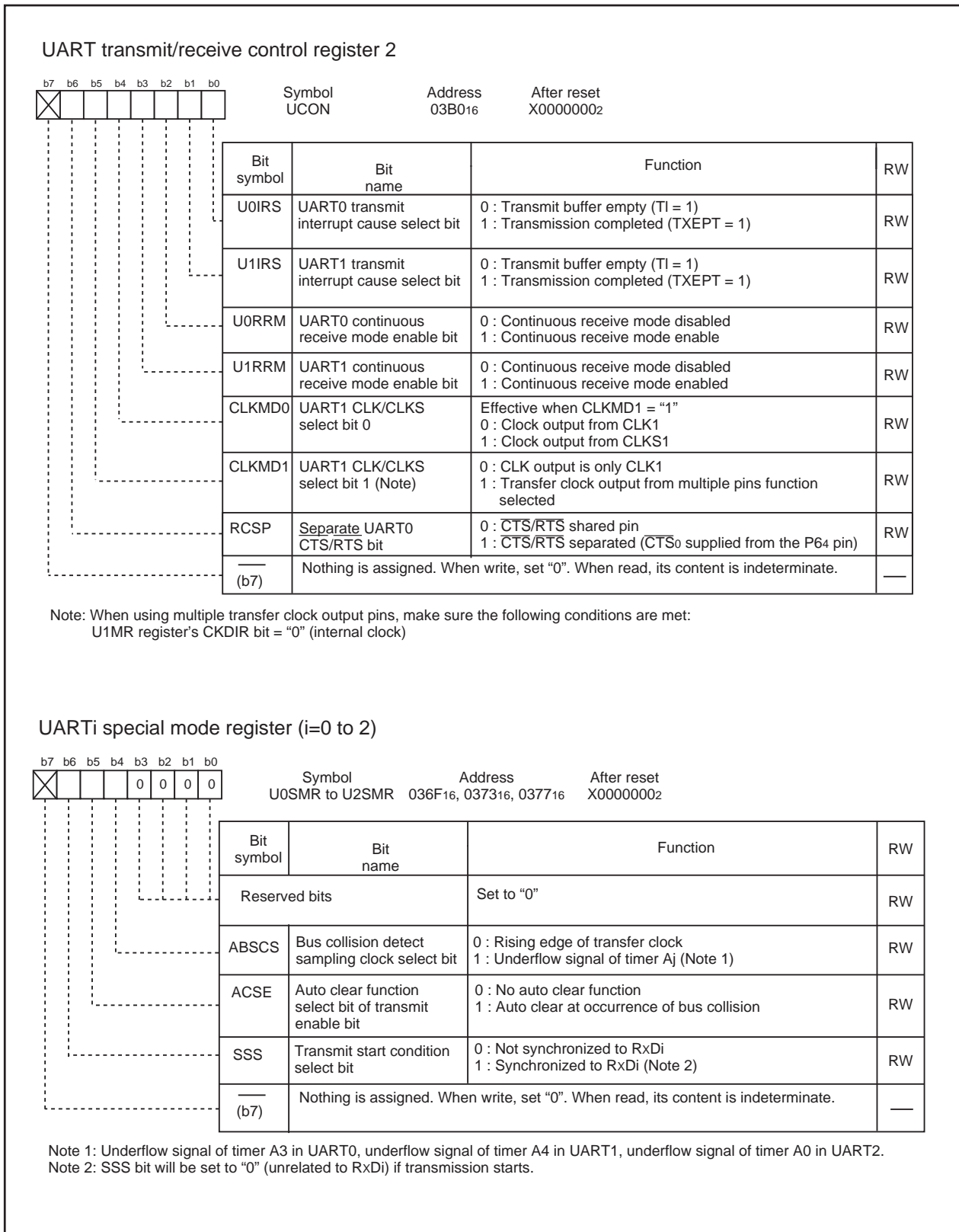


Figure 11.6. UCON Register and U0SMR to U2SMR Registers

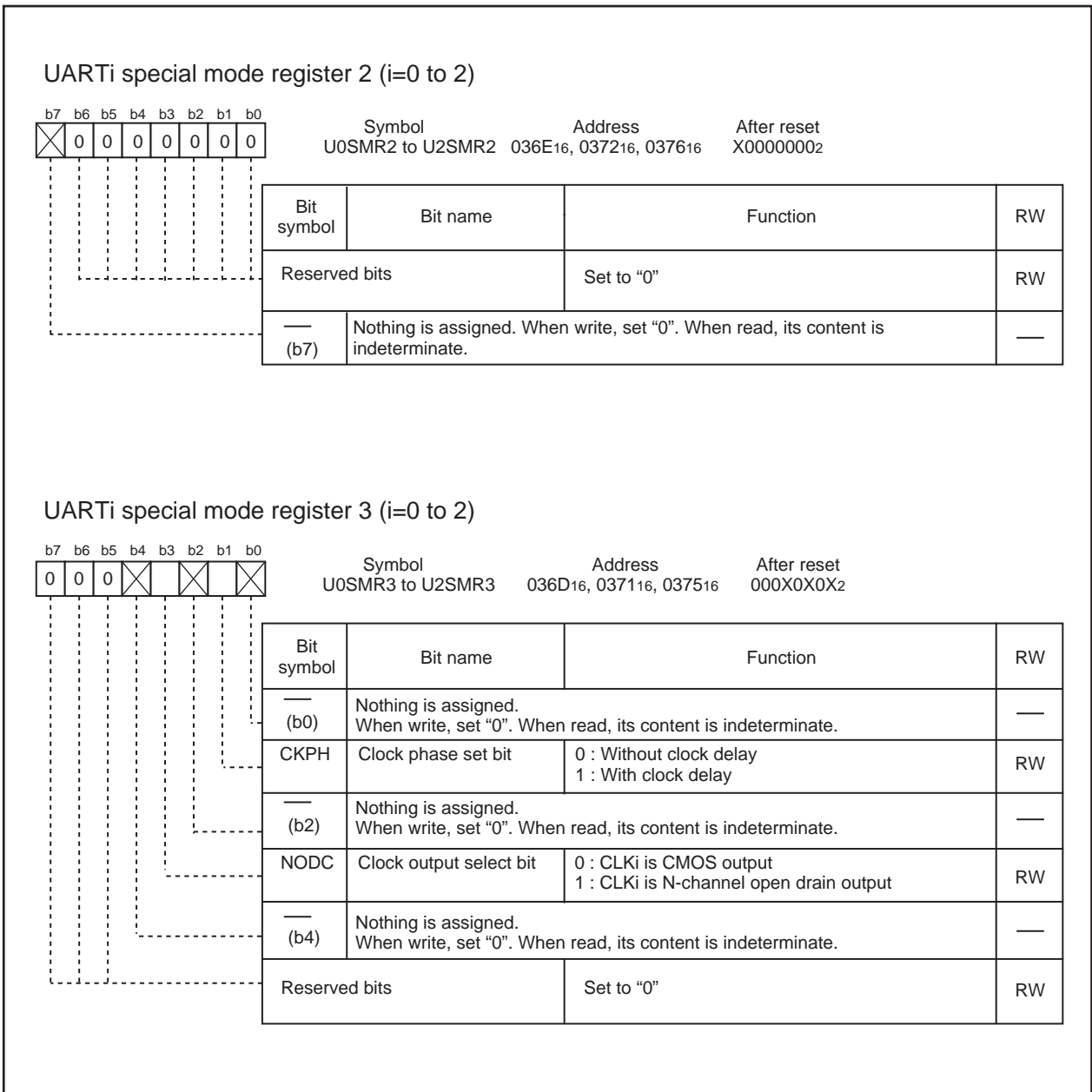


Figure 11.7. U0SMR2 to U2SMR2 Registers and U0SMR3 to U2SMR3 Registers

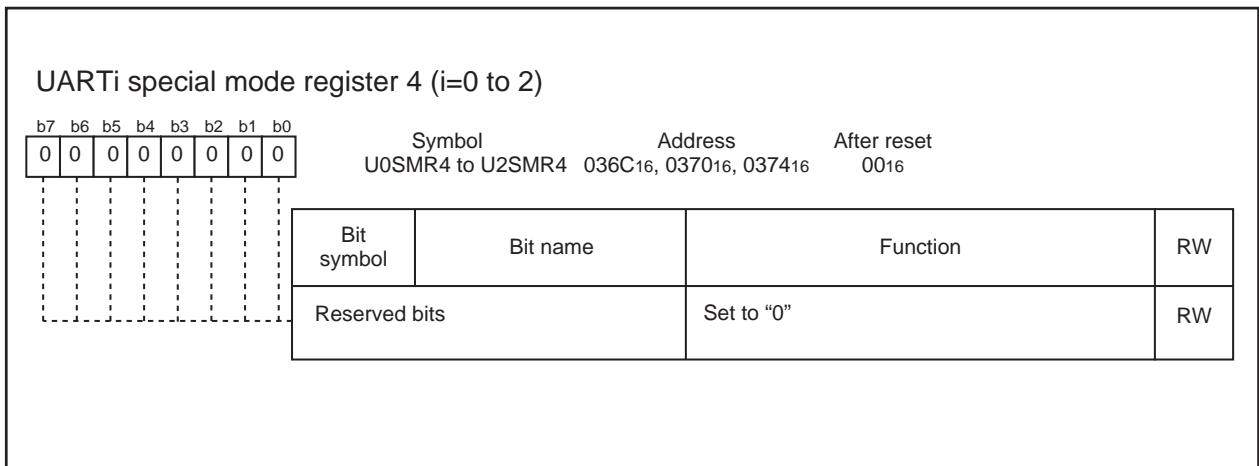


Figure 11.8. U0SMR4 to U2SMR4 Registers

Clock Synchronous serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 11.1 lists the specifications of the clock synchronous serial I/O mode. Table 11.2 lists the registers used in clock synchronous serial I/O mode and the register values set.

Table 11.1. Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer data format	<ul style="list-style-type: none"> Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> 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 CKDIR bit = "1" (external clock) : Input from CLKi pin
Transmission, reception control	<ul style="list-style-type: none"> Selectable from CTS function, RTS function or CTS/RTS function disable
Transmission start condition	<ul style="list-style-type: none"> Before transmission can start, the following requirements must be met (Note 1) <ul style="list-style-type: none"> 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"
Reception start condition	<ul style="list-style-type: none"> Before reception can start, the following requirements must be met (Note 1) <ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> 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) The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register For reception When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	<ul style="list-style-type: none"> 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 7th bit of the next data
Select function	<ul style="list-style-type: none"> CLK polarity selection Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected Continuous receive mode selection Reception is enabled immediately by reading the UiRB register Switching serial data logic This function reverses the logic value of the transmit/receive data Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set Separate CTS/RTS pins (UART0) CTS0 and RTS0 are input/output from separate pins

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.

Table 11. 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 ₀ 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

Table 11.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. Table 11.3 shows pin functions for the case where the multiple transfer clock output pin select function is deselected. Table 11.4 lists the P64 pin functions during clock synchronous serial I/O mode. Note that for a period from when the UART_i operation mode is selected to when transfer starts, the TxD_i pin outputs an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 11.3. Pin Functions (When Not Select Multiple Transfer Clock Output Pin Function)

Pin name	Function	Method of selection
TxD _i (i = 0 to 2) (P63, P67, P70)	Serial data output	(Outputs dummy data when performing reception only)
RxD _i (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 _i (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 _i /RTS _i (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 11.4. P64 Pin Functions

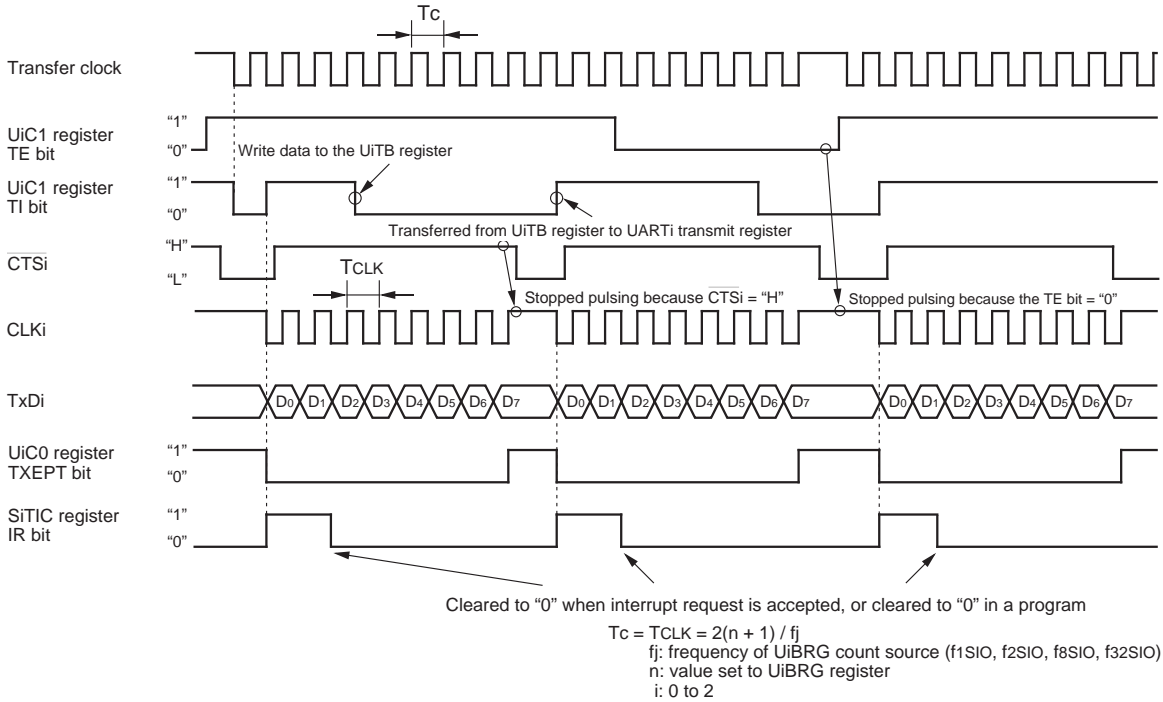
Pin function	Bit set value					
	U1C0 register		UCON register			PD6 register
	CRD	CRS	RCSP	CLKMD1	CLKMD0	PD6_4
P64	1	—	0	0	—	Input: 0, Output: 1
CTS ₁	0	0	0	0	—	0
RTS ₁	0	1	0	0	—	—
CTS ₀ (Note1)	0	0	1	0	—	0
CLKS ₁	—	—	—	1(Note 2)	1	—

Note 1: In addition to this, set the U0C0 register's CRD bit to “0” (CTS₀/RTS₀ enabled) and the U0C0 register's CRS bit to “1” (RTS₀ 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 CKPOL bit = 0
- Low if the U1C0 register's CKPOL bit = 1

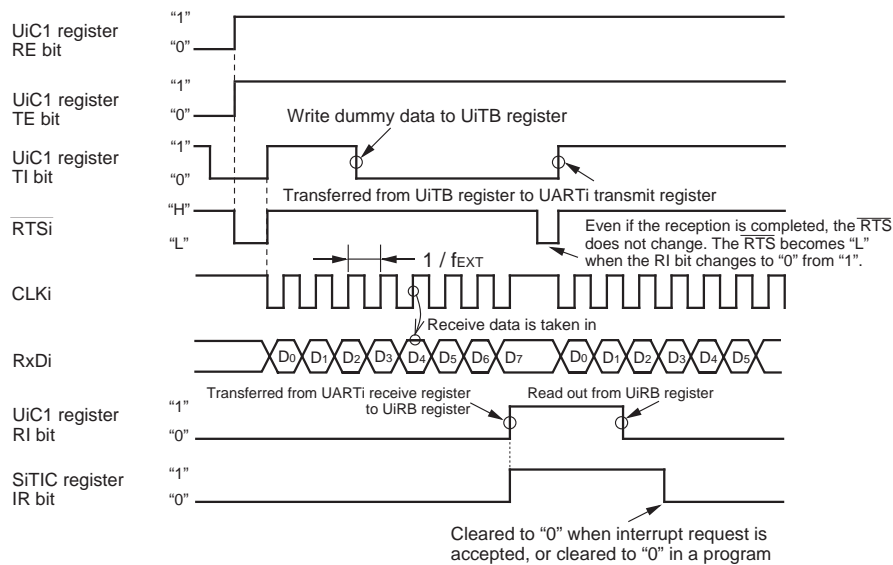
(1) Example of transmit timing (when internal clock is selected)



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

- 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)
- UiIRS bit = 0 (an interrupt request occurs when the transmit buffer becomes empty): U0IRS 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)
- 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 11.9. Transmit and Receive Operation

Counter Measure for Communication Error Occurs

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

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

- Resetting the UiTB register (i=0 to 2)
 - (1) Set the SMD2 to SMD0 bits in the UiMR register "000b" (Serial I/O disabled)
 - (2) Set the SMD2 to SMD0 bits in the UiMR register "001b" (Clock synchronous serial I/O mode)
 - (3) "1" is written to RE bit in the UiC1 register (reception enabled), regardless of the TE bit in the UiCi register

(a) CLK Polarity Select Function

Use the UiC0 register (i = 0 to 2)'s CKPOL bit to select the transfer clock polarity. Figure 11.10 shows the polarity of the transfer clock.

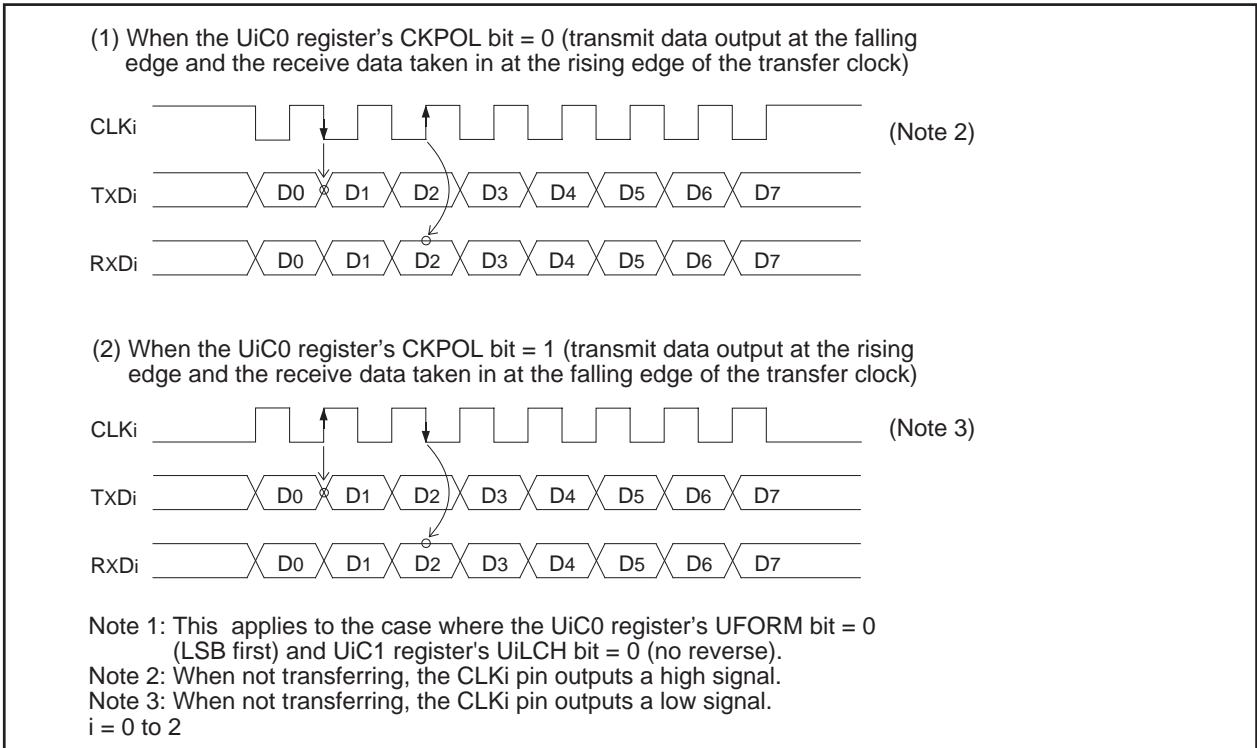


Figure 11.10. Transfer Clock Polarity

(b) LSB First/MSB First Select Function

Use the UiC0 register (i = 0 to 2)'s UFORM bit to select the transfer format. Figure 11.11 shows the transfer format.

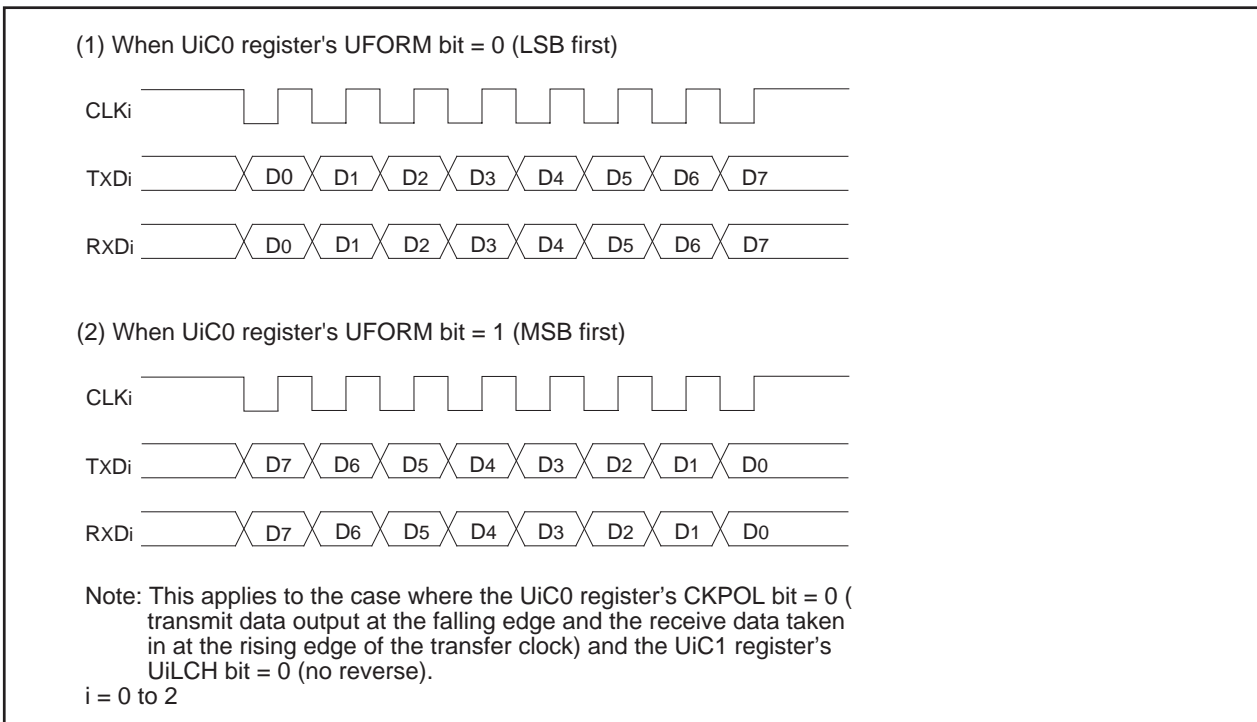


Figure 11.11. Transfer Format

(c) Continuous Receive Mode

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

When the U_iRRM bit ($i = 0$ to 2) = 1 (continuous receive mode), the U_iC1 register's TI bit is set to "0" (data present in the U_iTB register) by reading the U_iRB register. In this case, i.e., U_iRRM bit = 1, do not write dummy data to the U_iTB 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 U_iC1 register ($i = 0$ to 2)'s U_iLCH bit = 1 (reverse), the data written to the U_iTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U_iRB register. Figure 11.12 shows serial data logic.

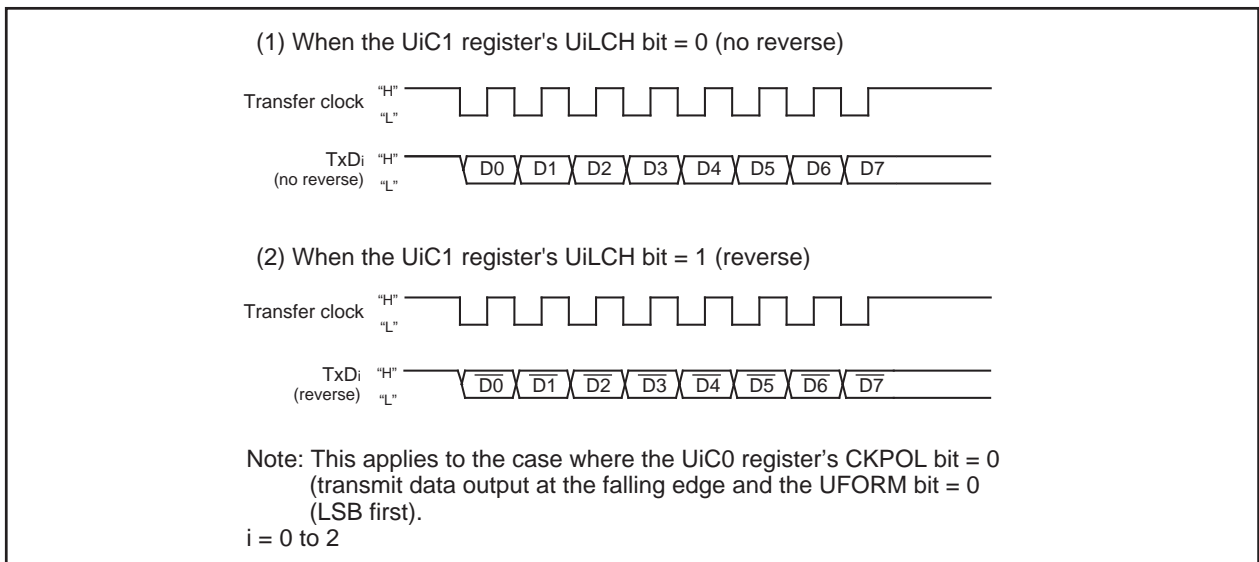


Figure 11.12. Serial Data Logic Switching

(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 11.13.) This function can be used when the selected transfer clock for UART1 is an internal clock.

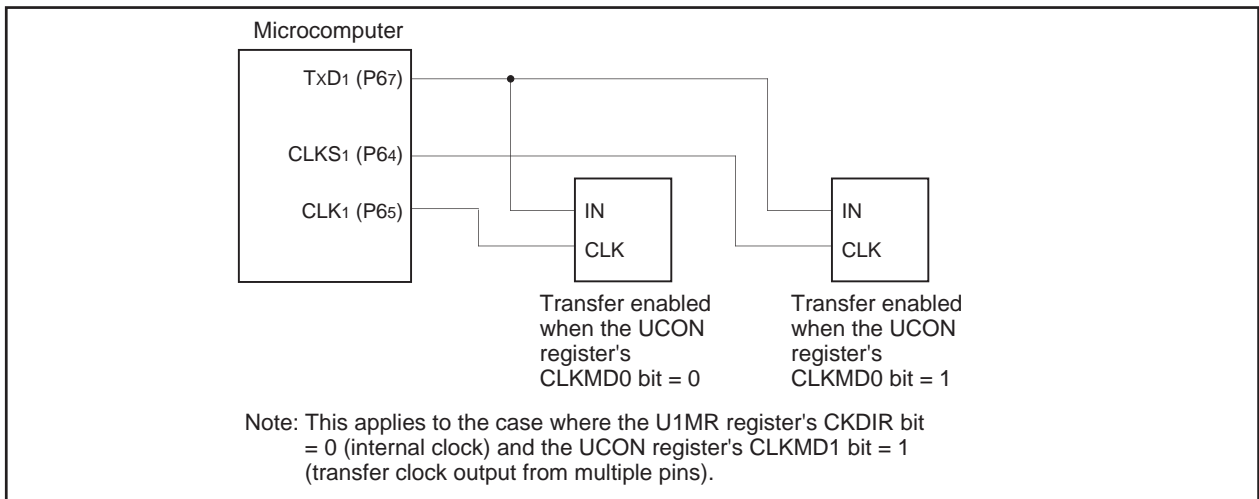


Figure 11.13. Transfer Clock Output From Multiple Pins

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

When the $\overline{\text{CTS}}$ function is used transmit and receive operation start when “L” is applied to the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ ($i=0$ to 2) pin. Transmit and receive operation begins when the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is held “L”. If the “L” signal is switched to “H” during a transmit or receive operation, the operation stops before the next data.

When the $\overline{\text{RTS}}$ function is used, the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin outputs on “L” signal when the microcomputer is ready to receive. The output level becomes “H” on the first falling edge of the CLK_i pin.

- CRD bit in U_iC0 register = 1 ($\overline{\text{CTS}}/\overline{\text{RTS}}$ function disabled) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is programmable I/O function
- CRD bit = 0, CRS bit = 0 ($\overline{\text{CTS}}$ function is selected) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is $\overline{\text{CTS}}$ function
- CRD bit = 0, CRS bit = 1 ($\overline{\text{RTS}}$ function is selected) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is $\overline{\text{RTS}}$ function

(g) $\overline{\text{CTS}}/\overline{\text{RTS}}$ 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.

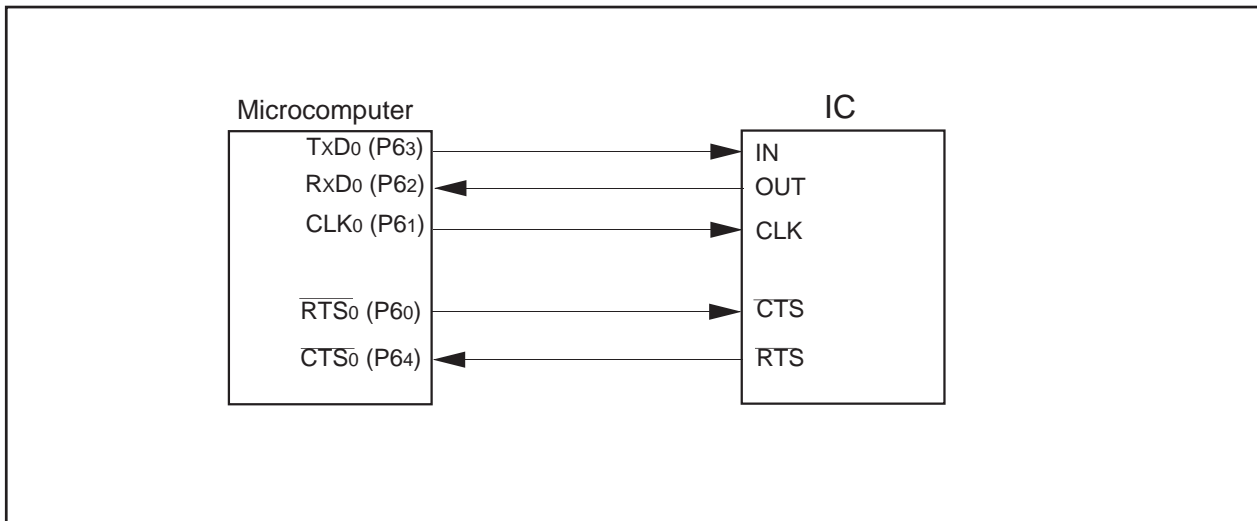


Figure 11.14. $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

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 11.5 lists the specifications of the UART mode.

Table 11.5. UART Mode Specifications

Item	Specification
Transfer data format	<ul style="list-style-type: none"> ● Character bit (transfer data): Selectable from 7, 8 or 9 bits ● Start bit: 1 bit ● Parity bit: Selectable from odd, even, or none ● Stop bit: Selectable from 1 or 2 bits
Transfer clock	<ul style="list-style-type: none"> ● UiMR(i=0 to 2) register's CKDIR bit = 0 (internal clock) : $f_j / 16(n+1)$ $f_j = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}$. n: Setting value of UiBRG register 00₁₆ to FF₁₆ ● CKDIR bit = "1" (external clock) : $f_{EXT} / 16(n+1)$ f_{EXT}: Input from CLKi pin. n: Setting value of UiBRG register 00₁₆ to FF₁₆
Transmission, reception control	<ul style="list-style-type: none"> ● Selectable from CTS function, RTS function or CTS/RTS function disable
Transmission start condition	<ul style="list-style-type: none"> ● Before transmission can start, the following requirements must be met <ul style="list-style-type: none"> • 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"
Reception start condition	<ul style="list-style-type: none"> ● Before reception can start, the following requirements must be met <ul style="list-style-type: none"> • The RE bit of UiC1 register = 1 (reception enabled) • Start bit detection
Interrupt request generation timing	<ul style="list-style-type: none"> ● For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> • 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) • The UiIRS bit = 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register ● For reception <ul style="list-style-type: none"> • When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	<ul style="list-style-type: none"> ● 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 ● Framing error (Note 3) This error occurs when the number of stop bits set is not detected ● Parity error (Note 3) This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set ● Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered
Select function	<ul style="list-style-type: none"> ● LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected ● Serial data logic switch This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. ● Tx/D, Rx/D I/O polarity switch This function reverses the polarities of the Tx/D pin output and Rx/D pin input. The logic levels of all I/O data is reversed. ● Separate CTS/RTS pins (UART0) CTS₀ and RTS₀ are input/output from separate pins

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

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

3: The timing when the framing error flag or parity flag are generated is detected when the data is transferred from the UARTi receive register to the UiRB register.

Table 11. 6. 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 I/O 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 3)
	CKPOL	Set to "0"
	UFORM	LSB first or MSB first can be selected when transfer data is 8 bits long. Set this bit to "0" when transfer data is 7 or 9 bits long.
UiC1	TE	Set this bit to "1" to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to "1" to enable reception
	RI	Reception complete flag
	U2IRS (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 ₀ 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

Table 11.7 lists the functions of the input/output pins during UART mode. Table 11.8 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 11.7. I/O Pin Functions

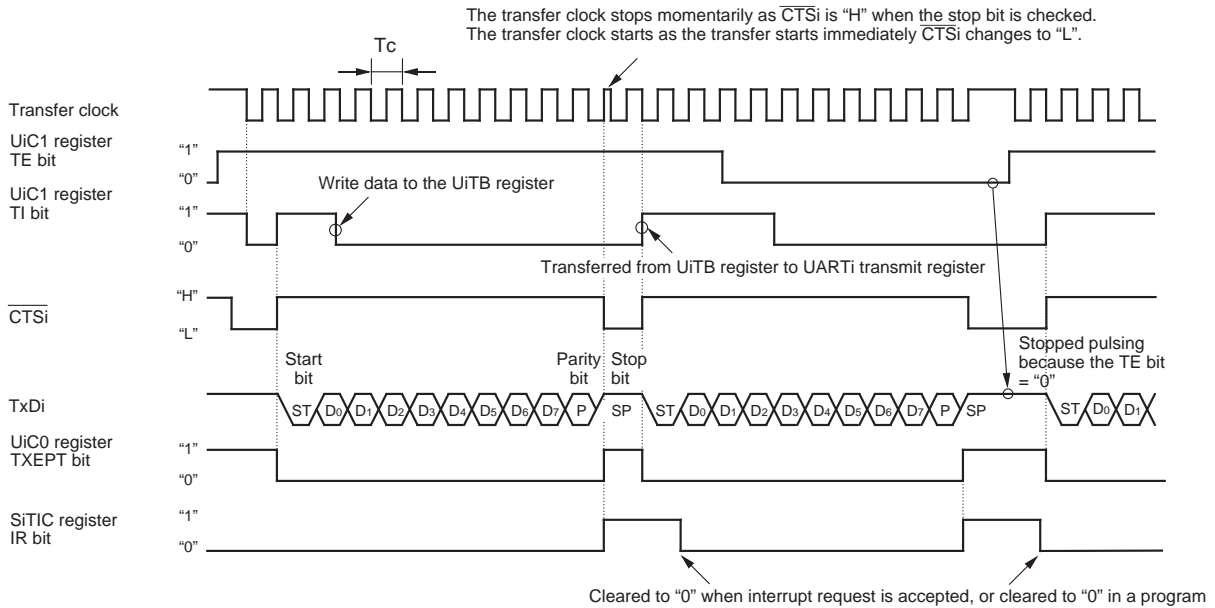
Pin name	Function	Method of selection
TxDi (i = 0 to 2) (P63, P67, P70)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P62, P66, P71)	Serial data input	PD6 register's PD6_2 bit=0, PD6_6 bit=0, PD7 register's PD7_1 bit=0 (Can be used as an input port when performing transmission 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
$\overline{\text{CTS}}/\overline{\text{RTS}}_i$ (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
	$\overline{\text{RTS}}$ output	UiC0 register's CRD bit=0 UiC0 register's CRS bit=1
	Input/output port	UiC0 register's CRD bit=1

Table 11.8. 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
$\overline{\text{RTS}}_1$	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” ($\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$ enabled) and the U0C0 register's CRS bit to “1” ($\overline{\text{RTS}}_0$ selected).

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



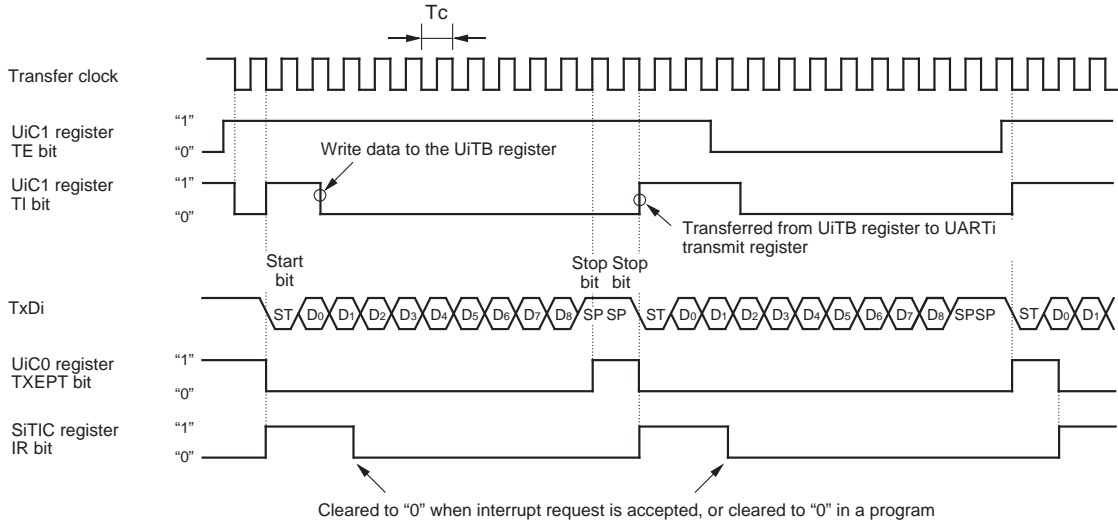
The above timing diagram applies to the case where the register bits are set as follows:

- UiMR register PRYE bit = 1 (parity enabled)
- UiMR register STPS bit = 0 (1 stop bit)
- UiC0 register CRD bit = 0 (CTS/RTS enabled), CRS bit = 0 (CTS selected)
- UiIRS bit = 1 (an interrupt request occurs when transmit completed):
 U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$$T_c = 16(n + 1) / f_j \text{ or } 16(n + 1) / f_{EXT}$$

f_j : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
 f_{EXT} : frequency of UiBRG count source (external clock)
 n : value set to UiBRG
 i : 0 to 2

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



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

- UiMR register PRYE bit = 0 (parity disabled)
- UiMR register STPS bit = 1 (2 stop bits)
- UiC0 register CRD bit = 1 (CTS/RTS disabled)
- UiIRS bit = 0 (an interrupt request occurs when transmit buffer becomes empty):
 U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$$T_c = 16(n + 1) / f_j \text{ or } 16(n + 1) / f_{EXT}$$

f_j : frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
 f_{EXT} : frequency of UiBRG count source (external clock)
 n : value set to UiBRG
 i : 0 to 2

Figure 11.15. Transmit Operation

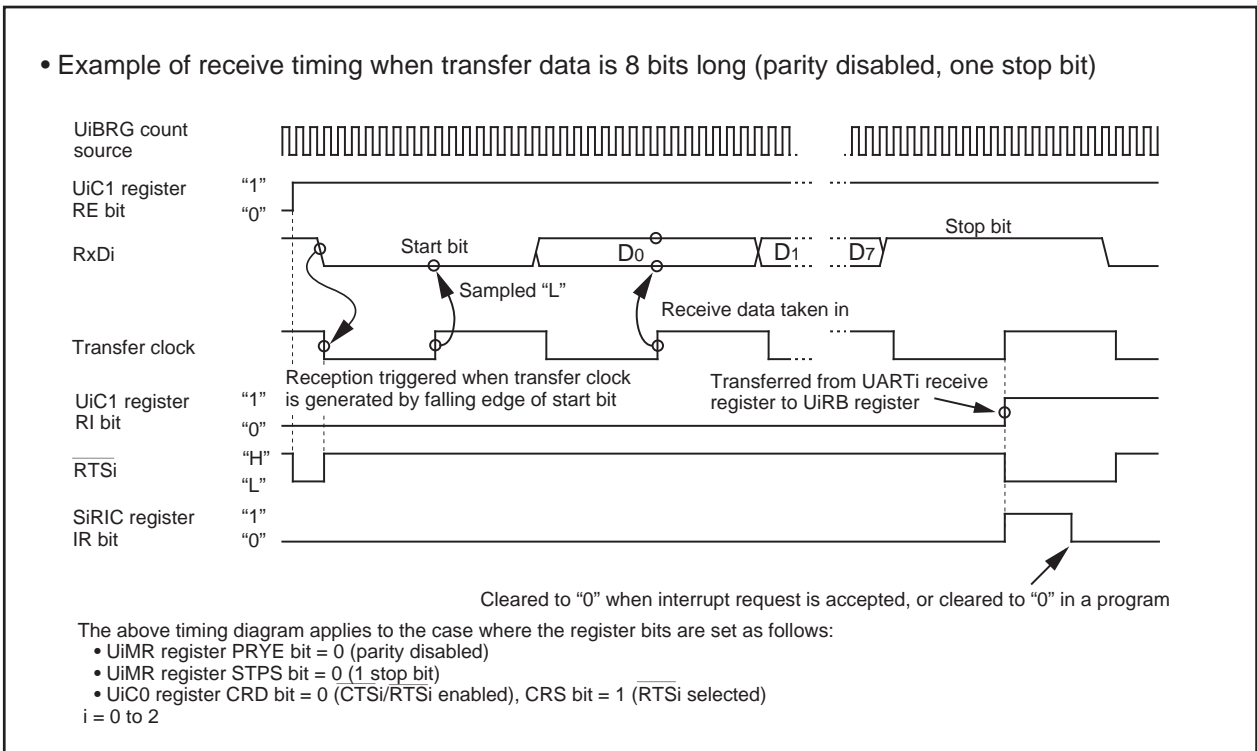


Figure 11.16. Receive Operation

(a) Bit Rates

In UART mode, the frequency set by the UiBRG register (i=0 to 2) divided by 16 become the bit rates. Table 11.9 lists example of bit rates and settings.

Table 11.9 Example of Bit Rates and Settings

Bit Rate (bps)	Count Source of BRG	Peripheral Function Clock : 16MHz		Peripheral Function Clock : 24MHz	
		Set Value of BRG : n	Actual Time (bps)	Set value of BRG : n	Actual Time (bps)
1200	f8	103 (67h)	1202	155 (96h)	1202
2400	f8	51 (33h)	2404	77 (46h)	2404
4800	f8	25 (19h)	4808	38 (26h)	4808
9600	f1	103 (67h)	9615	155 (96h)	9615
14400	f1	68 (44h)	14493	103 (67h)	14423
19200	f1	51 (33h)	19231	77 (46h)	19231
28800	f1	34 (22h)	28571	51 (33h)	28846
31250	f1	31 (1Fh)	31250	47 (2Fh)	31250
38400	f1	25 (19h)	38462	38 (26h)	38462
51200	f1	19 (13h)	50000	28 (1Ch)	51724

(b) Counter Measure for Communication Error Occurs

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

- Resetting the UiRB register (i=0 to 2)
 - (1) Set the RE bit in the UiC1 register to “0” (reception disabled)
 - (2) Set the RE bit in the UiC1 register to “1” (reception enabled)

- Resetting the UiTB register (i=0 to 2)
 - (1) Set the SMD2 to SMD0 bits in the UiMR register “000b” (Serial I/O disabled)
 - (2) Set the SMD2 to SMD0 bits in the UiMR register “001b”, “101b”, “110b”.
 - (3) “1” is written to RE bit in the UiC1 register (reception enabled), regardless of the TE bit in the UiCi register

(c) LSB First/MSB First Select Function

As shown in Figure 11.17, use the UiC0 register's UFORM bit to select the transfer format. This function is valid when transfer data is 8 bits long.

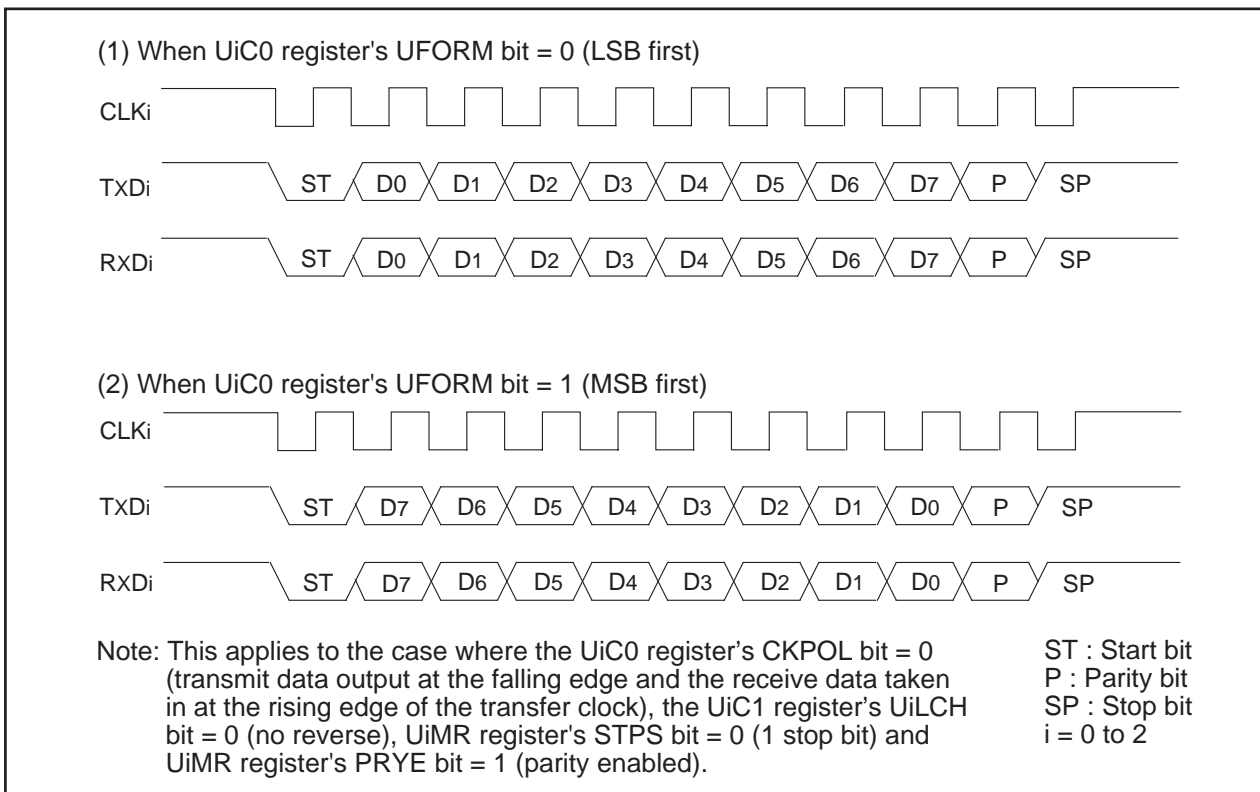


Figure 11.17. Transfer Format

(d) 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 11.18 shows serial data logic.

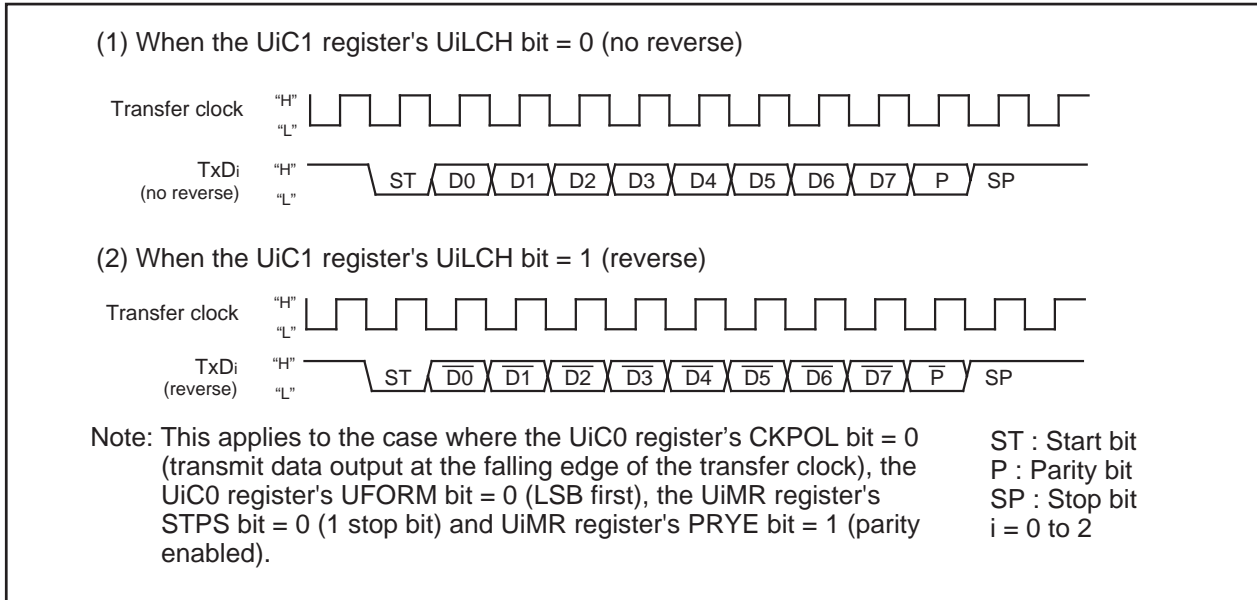


Figure 11.18. Serial Data Logic Switching

(e) TxD and RxD I/O Polarity Inverse Function

This function inverts the polarities of the TxD_i pin output and RxD_i pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inverted. Figure 11.19 shows the TxD pin output and RxD pin input polarity inverse.

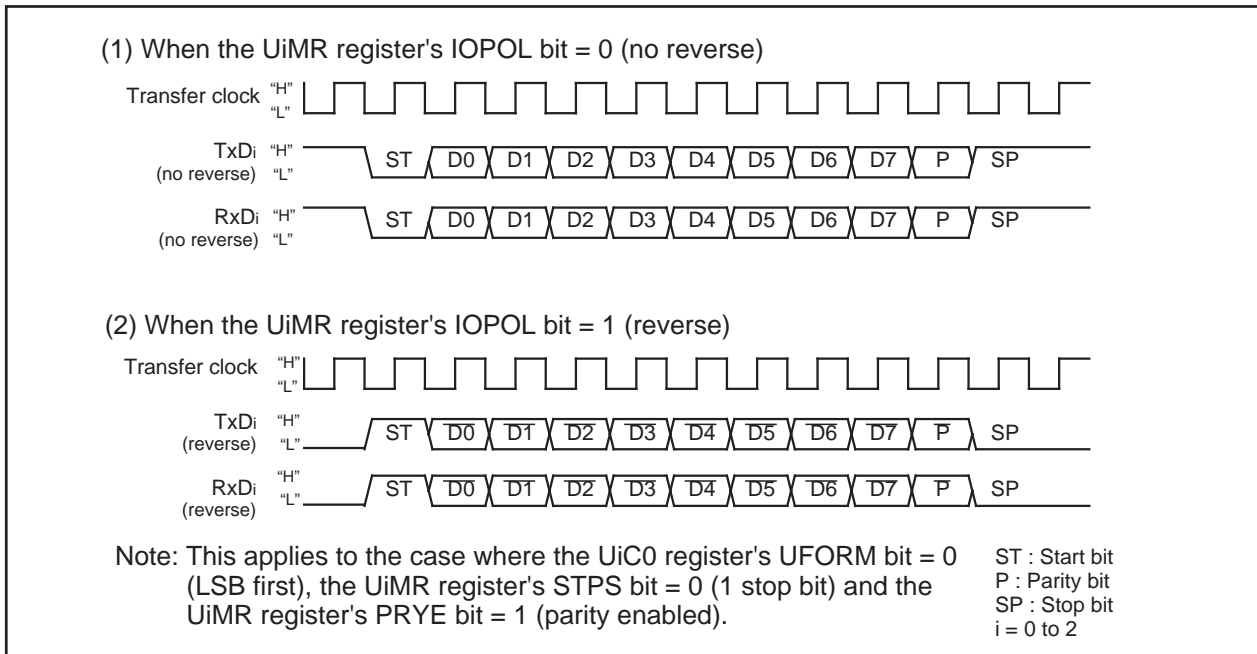


Figure 11.19. TxD and RxD I/O Polarity Inverse

(f) CTS/RTS Function

When the $\overline{\text{CTS}}$ function is used transmit operation starts when "L" is applied to the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ ($i=0$ to 2) pin. Transmit operation begins when the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is held "L". If the "L" signal is switched to "H" during a transmit operation, the operation stops before the next data.

When the $\overline{\text{RTS}}$ function is used, the $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin outputs on "L" signal when the microcomputer is ready to receive. The output level becomes "H" on the first falling edge of the CLK_i pin.

- CRD bit in UiC0 register = 1 (disable $\overline{\text{CTS}}/\overline{\text{RTS}}$ function of UART0)

$\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is programmable I/O function

- CRD bit = 0, CRS bit = 0 ($\overline{\text{CTS}}$ function is selected) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is $\overline{\text{CTS}}$ function
- CRD bit = 0, CRS bit = 1 ($\overline{\text{RTS}}$ function is selected) $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ pin is $\overline{\text{RTS}}$ function

(g) $\overline{\text{CTS}}/\overline{\text{RTS}}$ 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.

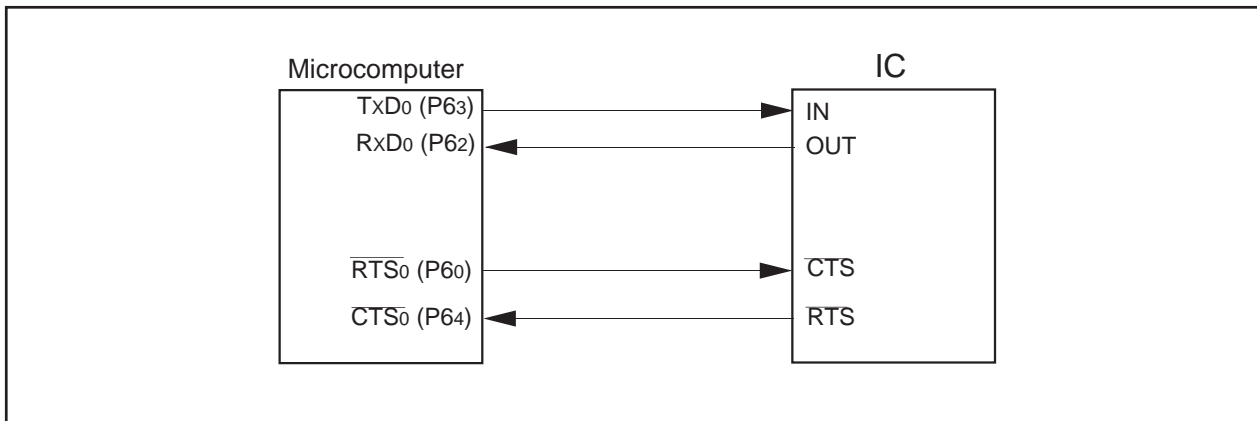


Figure 11.20. $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

Special Mode 2

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. Table 11.10 lists the specifications of Special Mode 2. Table 11.11 lists the registers used in Special Mode 2 and the register values set. Figure 11.21 shows communication control example for Special Mode 2.

Table 11.10. Special Mode 2 Specifications

Item	Specification
Transfer data format	<ul style="list-style-type: none"> Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> Master mode 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 Slave mode CKDIR bit = "1" (external clock selected) : Input from CLKi pin
Transmit/receive control	Controlled by input/output ports
Transmission start condition	<ul style="list-style-type: none"> Before transmission can start, the following requirements must be met (Note 1) <ul style="list-style-type: none"> The TE bit of UiC1 register= 1 (transmission enabled) The TI bit of UiC1 register = 0 (data present in UiTB register)
Reception start condition	<ul style="list-style-type: none"> Before reception can start, the following requirements must be met (Note 1) <ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> 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) The UiIRS bit =1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register For reception When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	<ul style="list-style-type: none"> 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 7th bit of the next data
Select function	<ul style="list-style-type: none"> Clock phase setting Selectable from four combinations of transfer clock polarities and phases

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.

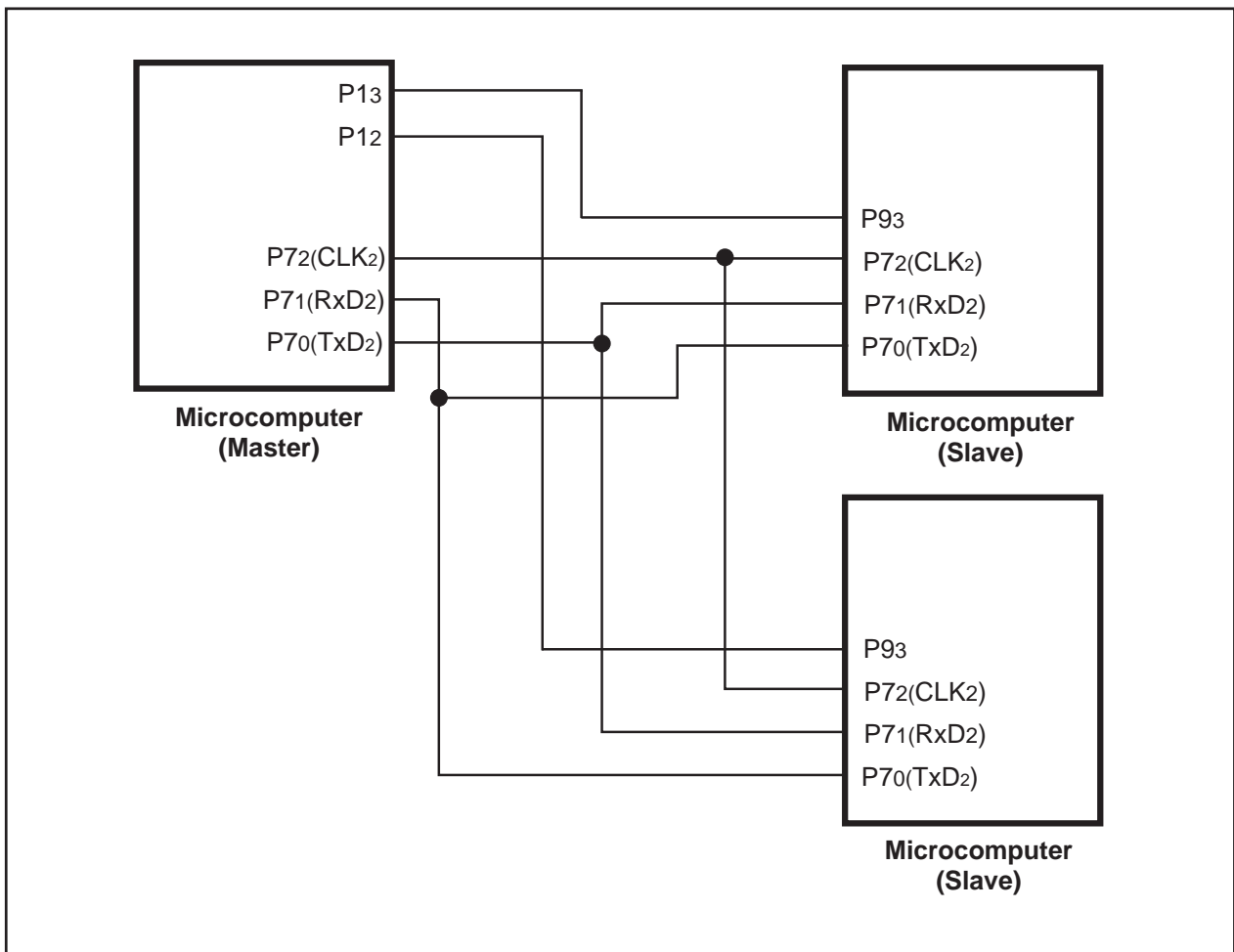


Figure 11.21. Serial Bus Communication Control Example (UART2)

Table 11. 11. 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"

Notes 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.

2: TxD2 pin is N channel open-drain output. Nothing is assigned. When writing, 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 Special Mode 2.

i = 0 to 2

- **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 11.22 shows the transmission and reception timing in master (internal clock).

(b) Slave (External Clock)

Figure 11.23 shows the transmission and reception timing (CKPH=0) in slave (external clock) while

Figure 11.24 shows the transmission and reception timing (CKPH=1) in slave (external clock).

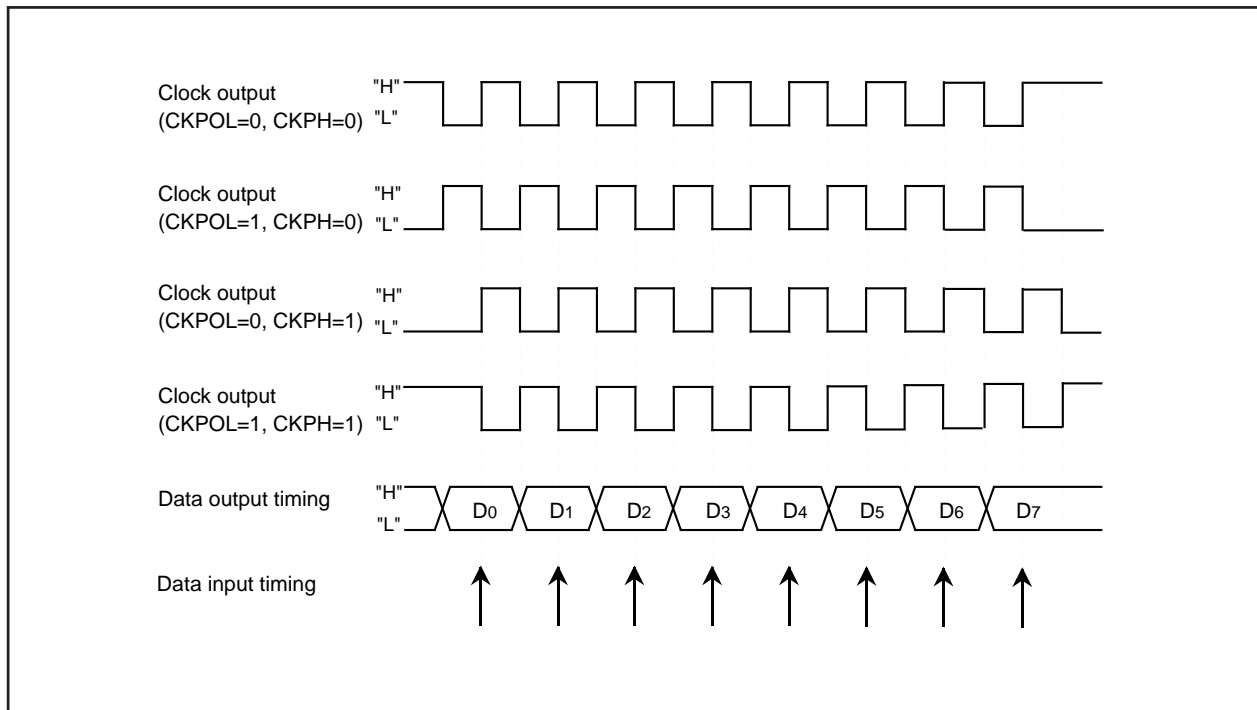


Figure 11.22. Transmission and Reception Timing in Master Mode (Internal Clock)

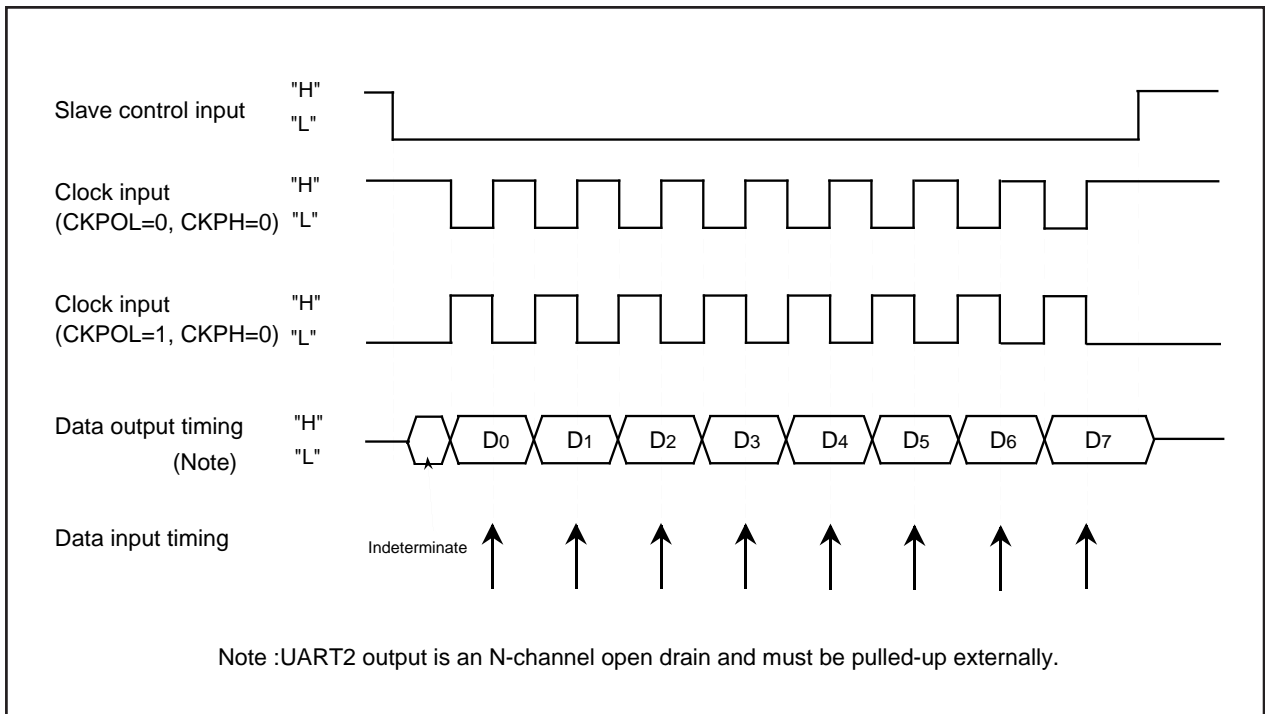


Figure 11.23. Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)

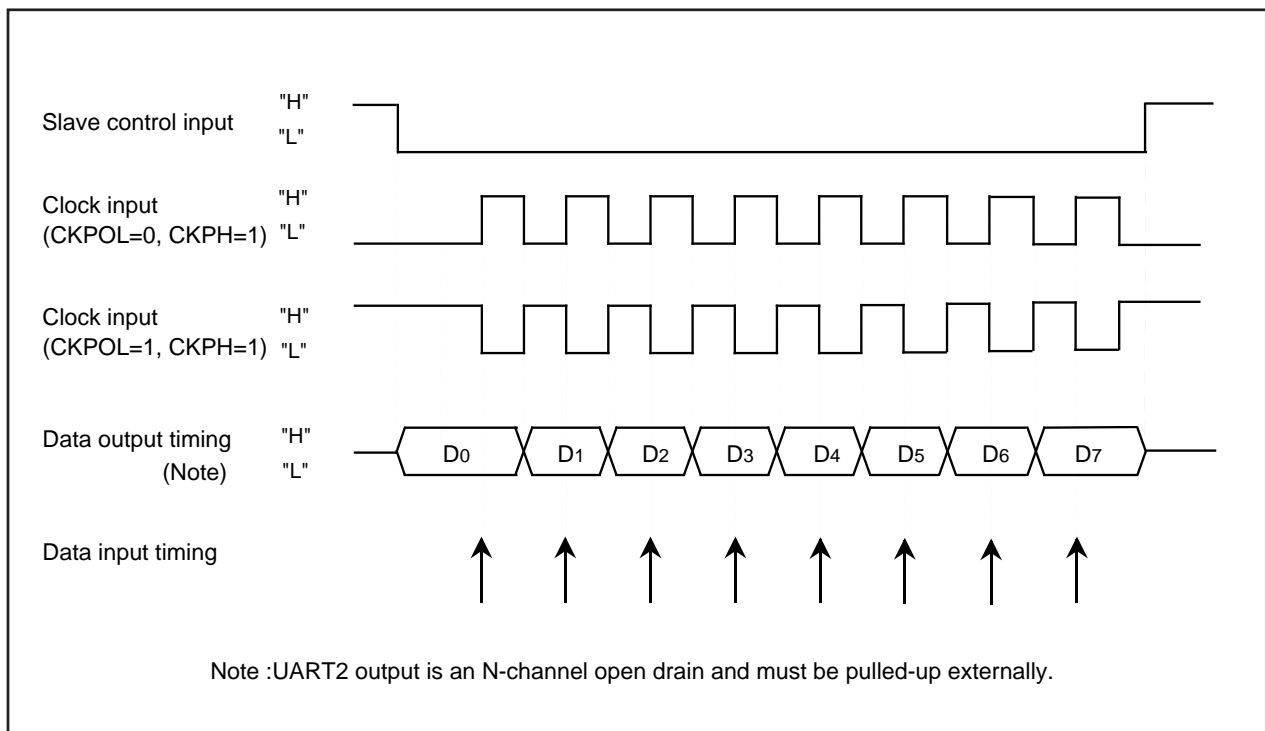


Figure 11.24. Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)

Special Mode 3 (IE mode)

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

Table 11.12 lists the registers used in IE mode and the register values set. Figure 11.25 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 UART i 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 11. 12. 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	0 to 7	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"

Notes 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.

2: TxD2 pin is N channel open-drain output. Nothing is assigned. When writing, 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 IE mode.

$i = 0$ to 2

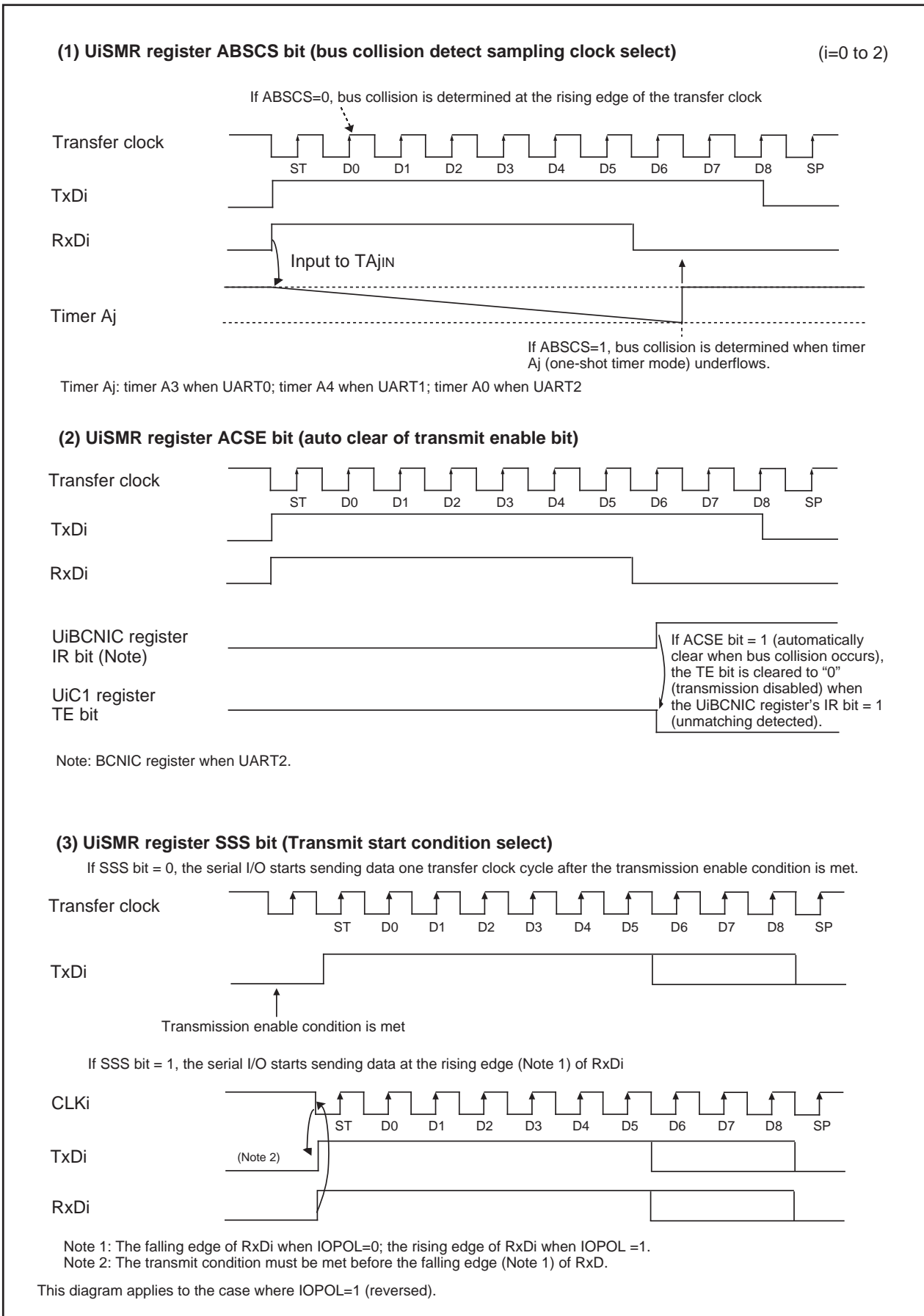


Figure 11.25. Bus Collision Detect Function-Related Bits

A/D Converter

The microcomputer contains one A/D converter circuit based on 8-bit successive approximation method configured with a capacitive-coupling amplifier. The analog inputs share the pins with P103 to P107, P04 to P07, and P24 to P27.

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

The A/D conversion result is stored in the ADi register bits for ANi, AN0i, and AN2i pins (i = 0 to 7).

Table 12.1 shows the performance of the A/D converter. Figure 12.1 shows the block diagram of the A/D converter, and Figures 12.2 and 12.3 show the A/D converter-related registers.

Table 12.1. Performance of A/D Converter

Item	Performance
Method of A/D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to VCC1
Operating clock ϕ_{AD} (Note 2)	$f_{AD}/\text{divide-by-2}$ of $f_{AD}/\text{divide-by-3}$ of $f_{AD}/\text{divide-by-4}$ of $f_{AD}/\text{divide-by-6}$ of $f_{AD}/\text{divide-by-12}$ of f_{AD}
Resolution	8-bit
Integral nonlinearity error	$\pm 5\text{LSB}$
Operating modes	One-shot mode, repeat mode, single sweep mode and repeat sweep mode 0
Analog input pins	5 pins (AN3 to AN7) + 4 pins (AN04 to AN07) + 4 pins (AN24 to AN27)
A/D conversion start condition	<ul style="list-style-type: none"> Software trigger The ADCON0 register's ADST bit is set to "1" (A/D conversion starts)
Conversion speed per pin	<ul style="list-style-type: none"> Without sample and hold function 49 ϕ_{AD} cycles With sample and hold function 28 ϕ_{AD} cycles

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

Note 2: Operation clock frequency (ϕ_{AD} frequency) must be 10 MHz or less.

A case without sample and hold function turn (ϕ_{AD} frequency) into 250kHz or more .

A case with the sample and hold function turn (ϕ_{AD} frequency) into 1MHz or more.

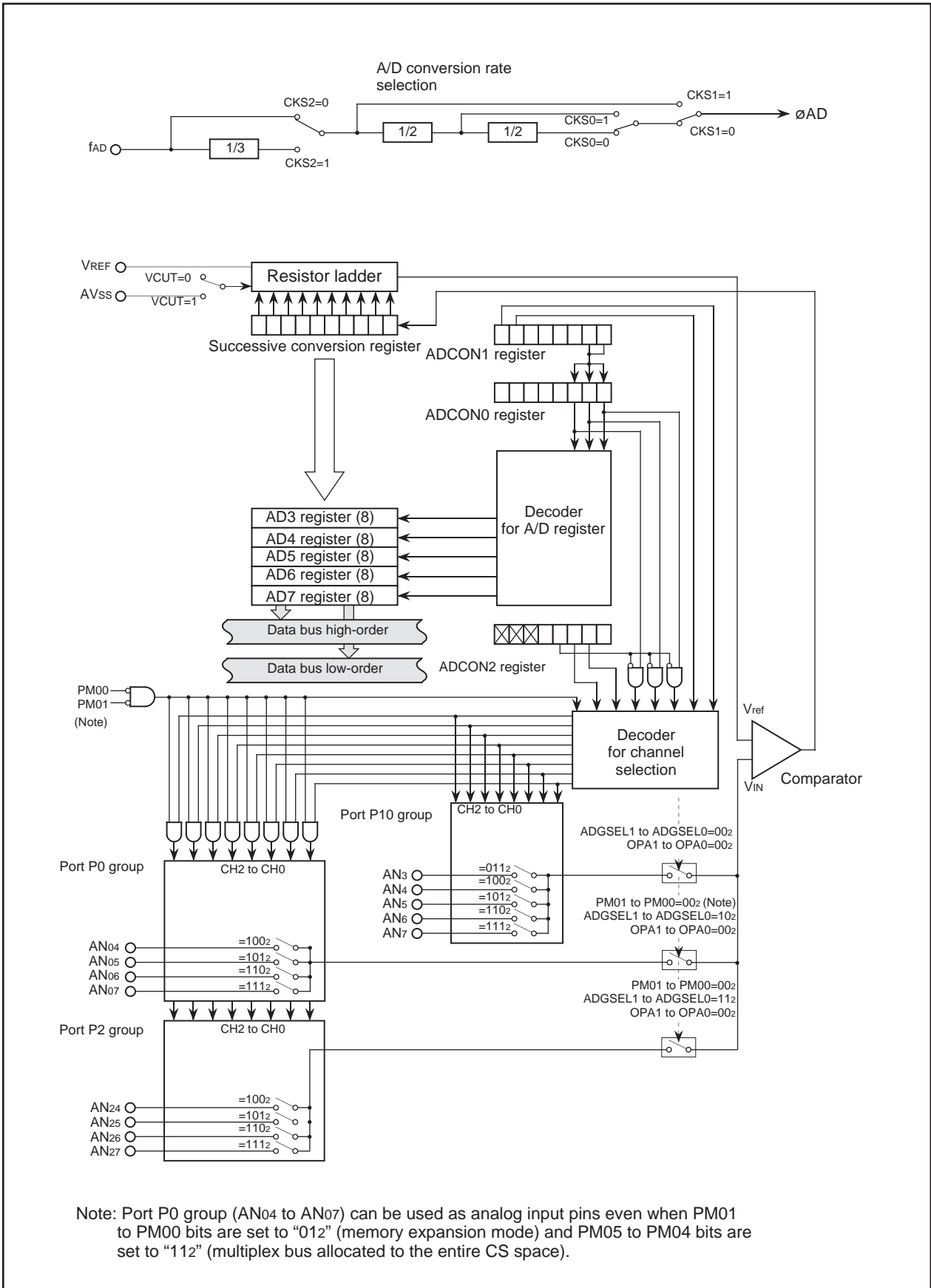


Figure 12.1. A/D Converter Block Diagram

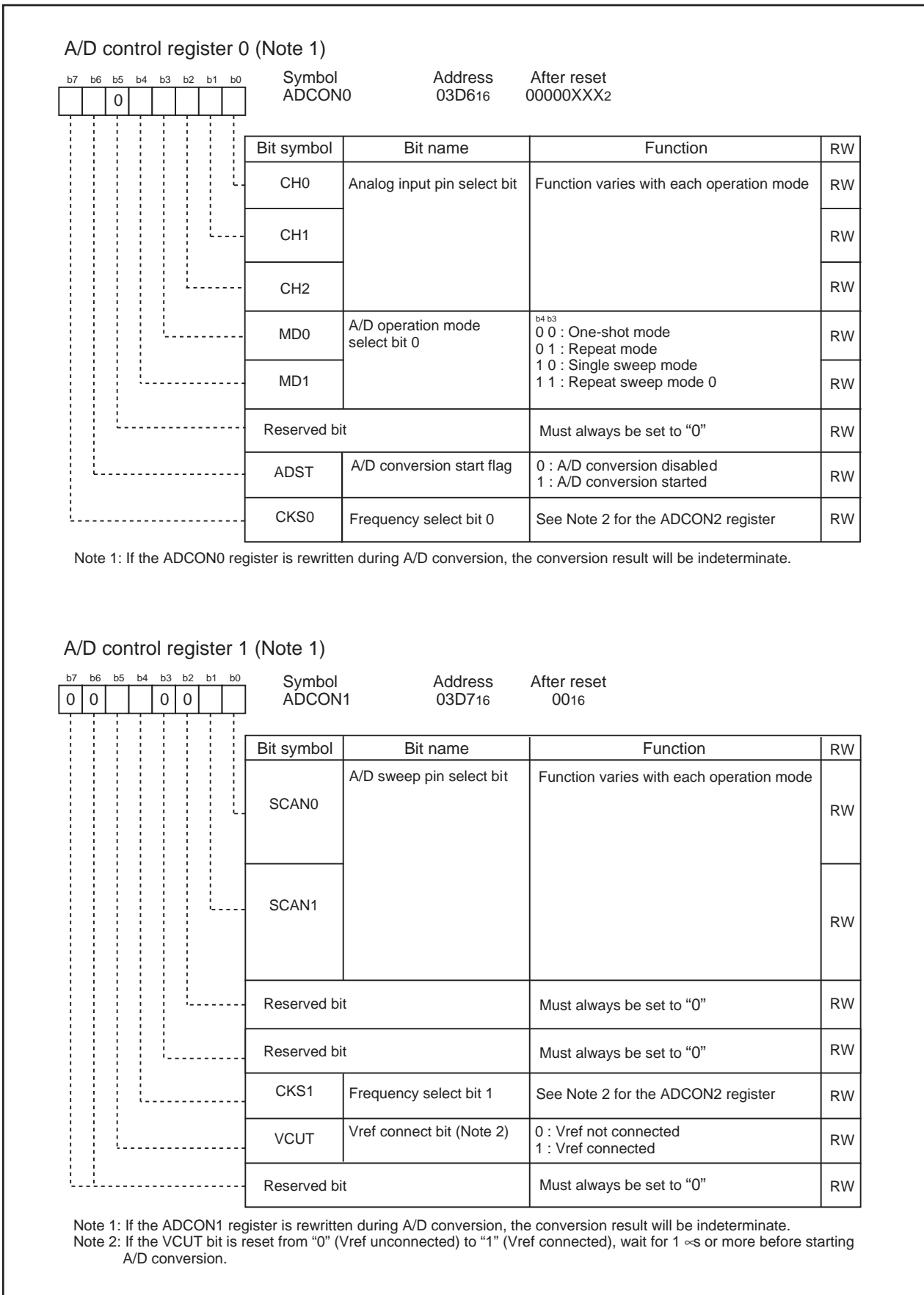
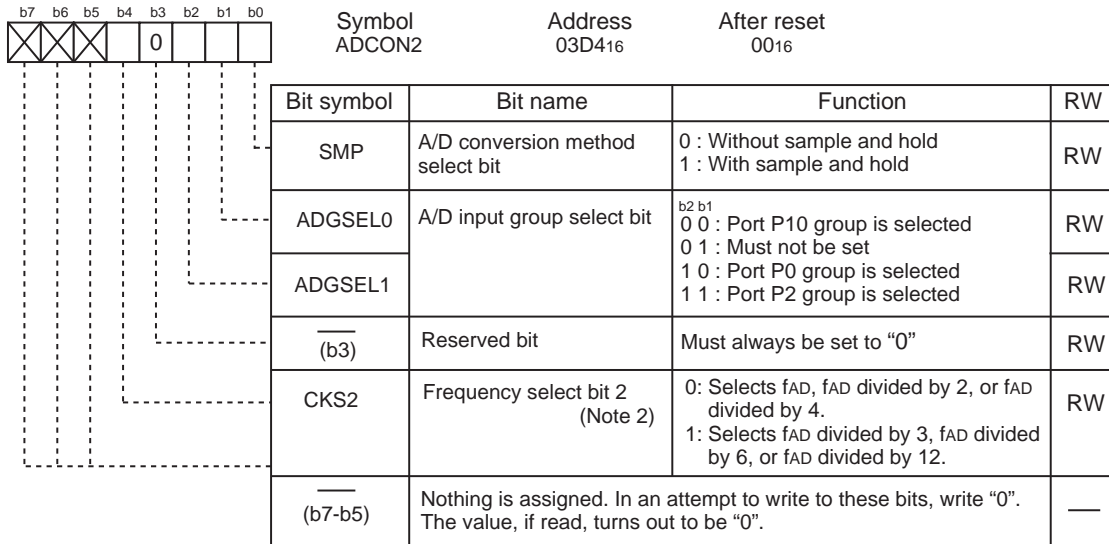


Figure 12.2. ADCON0 to ADCON1 Registers

A/D control register 2 (Note 1)



Notes 1: If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate
2: The \varnothing_{AD} frequency must be 10 MHz or less. The selected \varnothing_{AD} frequency is determined by a combination of the ADCON0 register's CKS0 bit, ADCON1 register's CKS1 bit, and ADCON2 register's CKS2 bit.

CKS2	CKS1	CKS0	\varnothing_{AD}
0	0	0	Divide-by-4 of f _{AD}
0	0	1	Divide-by-2 of f _{AD}
0	1	0	f _{AD}
0	1	1	
1	0	0	Divide-by-12 of f _{AD}
1	0	1	Divide-by-6 of f _{AD}
1	1	0	Divide-by-3 of f _{AD}
1	1	1	

A/D register i (i=3 to 7)	Symbol	Address	After reset
	AD3	03C6 ₁₆	Indeterminate
	AD4	03C8 ₁₆	Indeterminate
	AD5	03CA ₁₆	Indeterminate
	AD6	03CC ₁₆	Indeterminate
	AD7	03CE ₁₆	Indeterminate

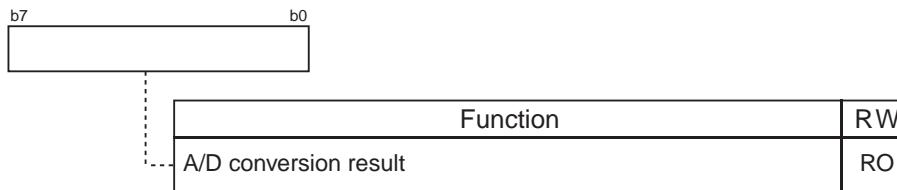


Figure 12.3. ADCON2 Register, and AD3 to AD7 Registers

(1) One-shot Mode

In this mode, the input voltage on one selected pin is A/D converted once. Table 12.2 shows the specifications of one-shot mode. Figure 12.4 shows the ADCON0 to ADCON1 registers in one-shot mode.

Table 12.2. One-shot Mode Specifications

Item	Specification
Function	Bits CH2 to CH0 of ADCON0 register and bits ADGSEL1 to ADGSEL0 bit of ADCON2 register
Start condition	Writing "1" to ADST bit of ADCON0 register
Stop condition	<ul style="list-style-type: none"> • End of A/D conversion • Writing "0" to ADST bit
Interrupt request generation timing	End of A/D conversion
Input pin	One of AN3 to AN7, AN04 to AN07, AN24 to AN27, as selected
Reading of result of A/D converter	Read AD3 to AD7 registers corresponding to selected pin

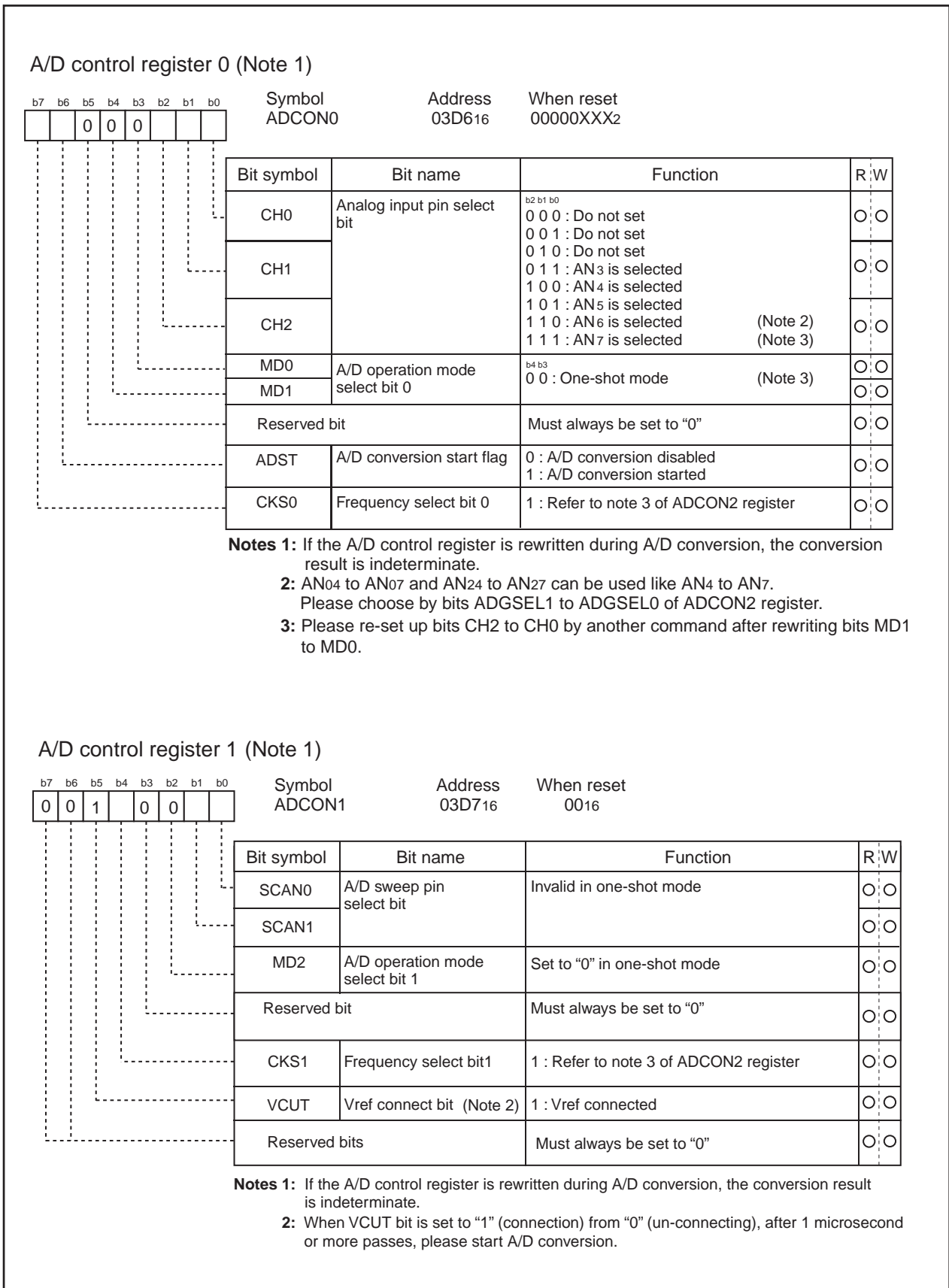


Figure 12.4 ADCON0 Register and ADCON1 Register (One-shot Mode)

(2) Repeat mode

In this mode, the input voltage on one selected pin is A/D converted repeatedly. Table 12.3 shows the specifications of repeat mode. Figure 12.5 shows the ADCON0 to ADCON1 registers in repeat mode.

Table 12.3. Repeat Mode Specifications

Item	Specification
Function	Bits CH2 to CH0 of ADCON0 register and bits ADGSEL1 to ADGSEL0 of ADCON2 register.
A/D conversion start conditions	ADST bit of ADCON0 register is set to "1" (A/D conversion start).
A/D conversion stop conditions	ADST bit is set to "0" (A/D conversion stop).
Interruption demand generating timing	At the time of a A/D conversion end
Analog input pin	One pin is chosen from AN3 to AN7 and AN04 to AN07 and AN24 to AN27.
Read-out of A/D conversion value	Read out of registers AD3 to AD7 corresponding to the selected pin.

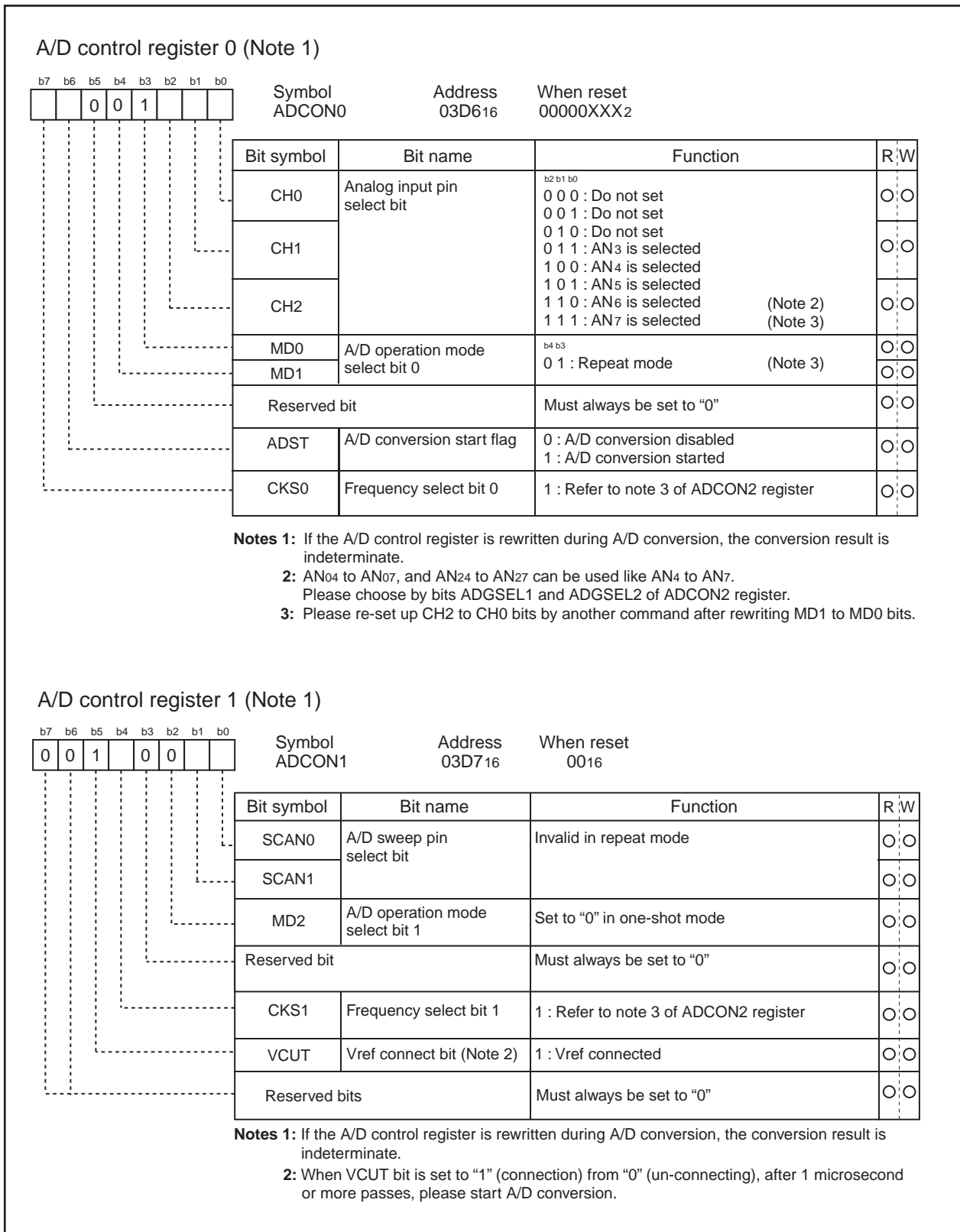


Figure 12.5. ADCON0 Register and ADCON1 Register (Repeat Mode)

(3) Single Sweep Mode

In this mode, the input voltages on selected pins are A/D converted, one pin at a time. Table 12.4 shows the specifications of single sweep mode. Figure 12.6 shows the ADCON0 to ADCON1 registers in single sweep mode.

Table 12.4. Single Sweep Mode Specifications

Item	Specification
Function	A/D conversion of the input voltage of pin chosen by bits SCAN1 to SCAN0 of ADCON1 register and bits ADGSEL1 to ADGSEL0 of ADCON2 register is carried out by a unit of 1 time.
A/D conversion start conditions	ADST bit of ADCON0 register is set to "1" (A/D conversion start).
A/D conversion stop conditions	<ul style="list-style-type: none"> ● A/D conversion end ● ADST bit is set to "0"
Interruption demand generating timing	At the time of a A/D conversion end
Analog input pin	From ANi4 to ANi5 (two pins), and ANi4 to ANi7 (four pins) to selection (i=0, 2) (Note 1)
Read-out of A/D conversion value	Read out of registers AD4 to AD7 corresponding to the selected pin.

Note: AN4 to AN7 can be used like AN04 to AN07, and AN24 to AN27. In this case, it becomes selection from AN4 to AN5 (two pins), and AN4 to AN7 (four pins).

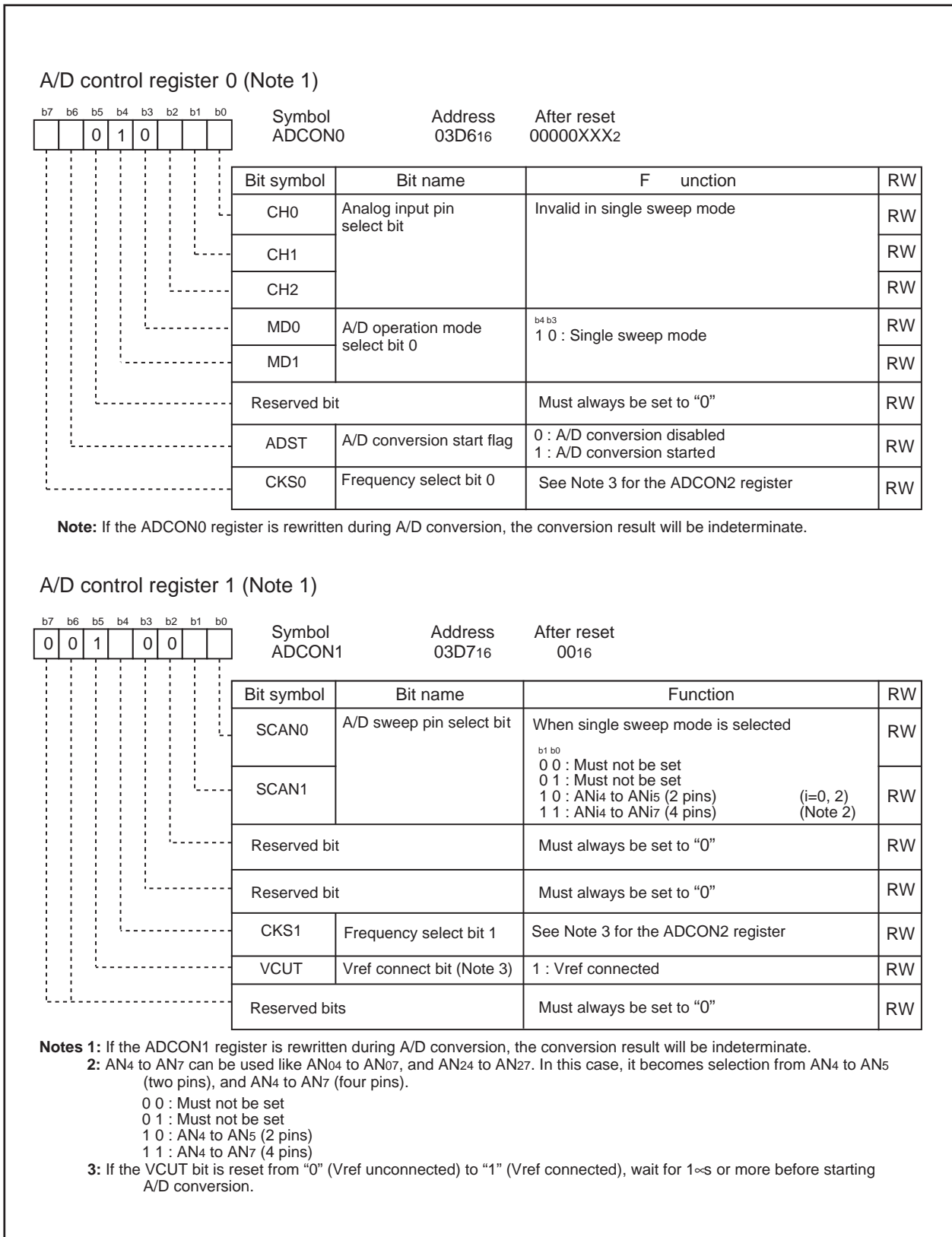


Figure 12.6. ADCON0 Register and ADCON1 Register (Single Sweep Mode)

(4) Repeat Sweep Mode 0

In this mode, the input voltages on selected pins are A/D converted repeatedly. Table 12.5 shows the specifications of repeat sweep mode 0. Figure 12.7 shows the ADCON0 to ADCON1 registers in repeat sweep mode 0.

Table 12.5. Repeat Sweep Mode 0 Specifications

Item	Specification
Function	A/D conversion of the input voltage of pin chosen by bits SCAN1 to SCAN0 of ADCON1 register and bits ADGSEL1 to ADGSEL0 of ADCON2 register is carried out by a unit of 1 time.
A/D conversion start conditions	ADST bit of ADCON0 register is set to "1" (A/D conversion start).
A/D conversion stop conditions	ADST bit is set to "0" (A/D conversion stop).
Interruption demand generating timing	An interruption demand is not generated.
Analog input pin	From ANi4 to ANi5 (two pins), and ANi4 to ANi7 (four pins) to selection (i=0, 2) (Note 1)
Read-out of A/D conversion value	Read out of registers AD3 to AD7 corresponding to the selected pin.

Note: AN4 to AN7 can be used like AN04 to AN07, and AN24 to AN27. In this case, it becomes selection from AN4 to AN5 (two pins), and AN4 to AN7 (four pins).

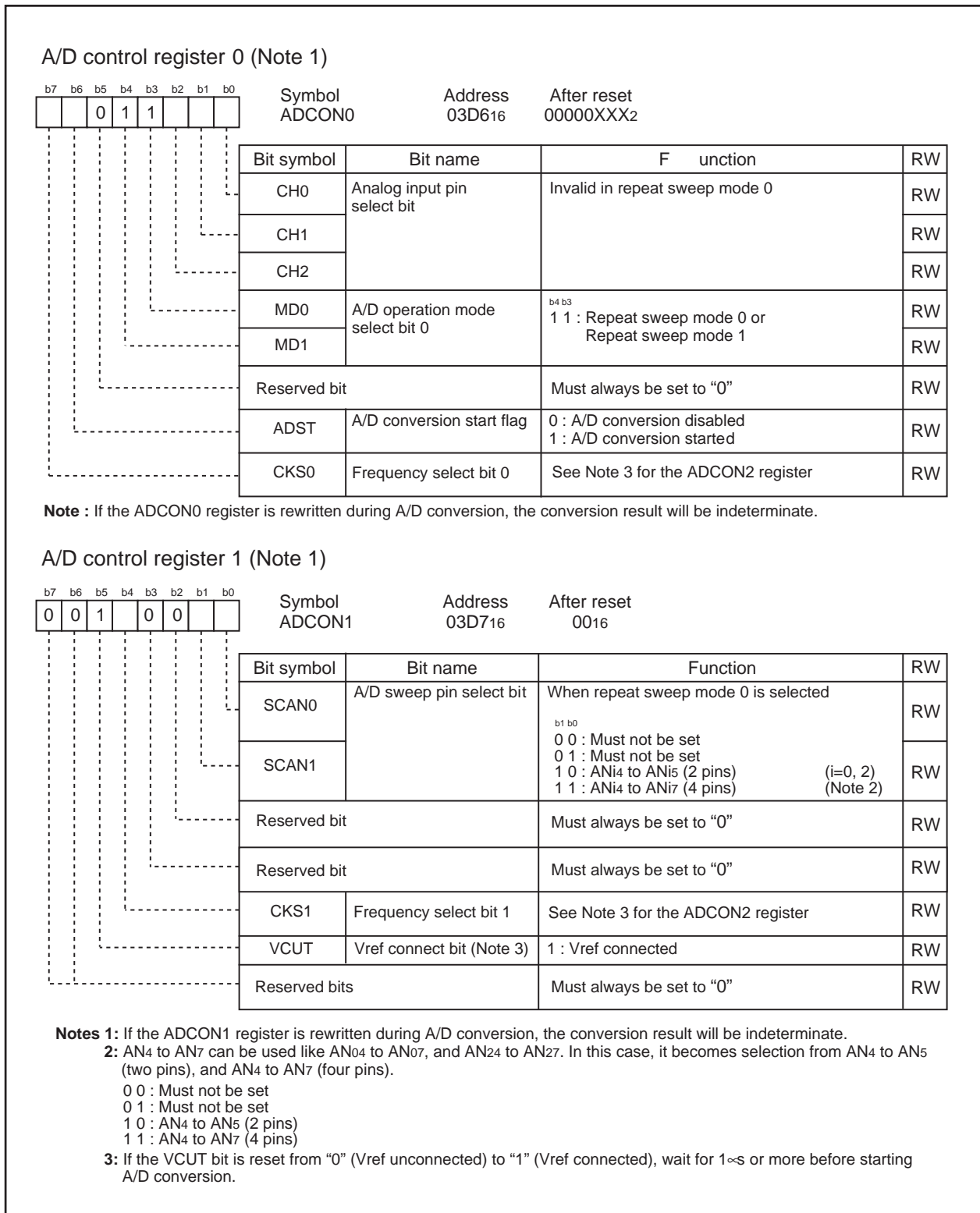


Figure 12.7. ADCON0 Register and ADCON1 Registers (Repeat Sweep Mode 0)

Sample and Hold

If the SMP bit of ADCON2 register is set to "1" (those with a sample & hold), the conversion speed per one pin will improve and it will become a 28 ϕ AD cycle. However, in all modes, be sure to specify before starting A/D conversion whether sample and hold is to be used.

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.

Notes at the time of using A/D converter

- (1) Please set to "0" (input mode) the direction bit of a port corresponding to the pin used as an analog input pin.
- (2) When you use key input interruption, please do not use all of pins AN4 to AN7 as an analog input pin (if A/D input voltage is set to "L", a key input interruption demand will occur).
- (3) In order to reduce prevention of incorrect operation and the latch rise by the noise, and a conversion error, please insert a capacitor, respectively between VCC1 pin, VCC2 pin, an analog input pin (AN_i (i=3 to 7), AN0_i, AN2_i), and a VSS pin. The example of processing of each pin is shown in Fig.12.8.
- (4) A/D conversion is completed, and the mistaken value is stored in an AD_i register when CPU reads an AD_i register to the timing which stores the result in an AD_i register (i=0 to 7). This phenomenon is generated when the clock which divided the main clock, or a sub clock is chosen as a CPU clock.

● When using it in one-shot mode or single sweep mode

Please read the target AD_i register after checking that A/D conversion has been completed (completion of A/D conversion can be judged in IR bit of ADIC register).

● When using it in repeat mode, repeat sweep mode 0 or repeat sweep mode 1

Please use a CPU clock, without divide a main clock.

- (5) When the ADST bit of ADCON0 register is set to "0" (A/D conversion stop) and it forces by the program during A/D conversion operation to terminate, the conversion result of a A/D conversion machine becomes unfixed. Moreover, the AD_i register which omits A/D conversion may also become unfixed. During A/D conversion operation, when an ADST bit is set to "0" by the program, please use no value of AD_i registers.

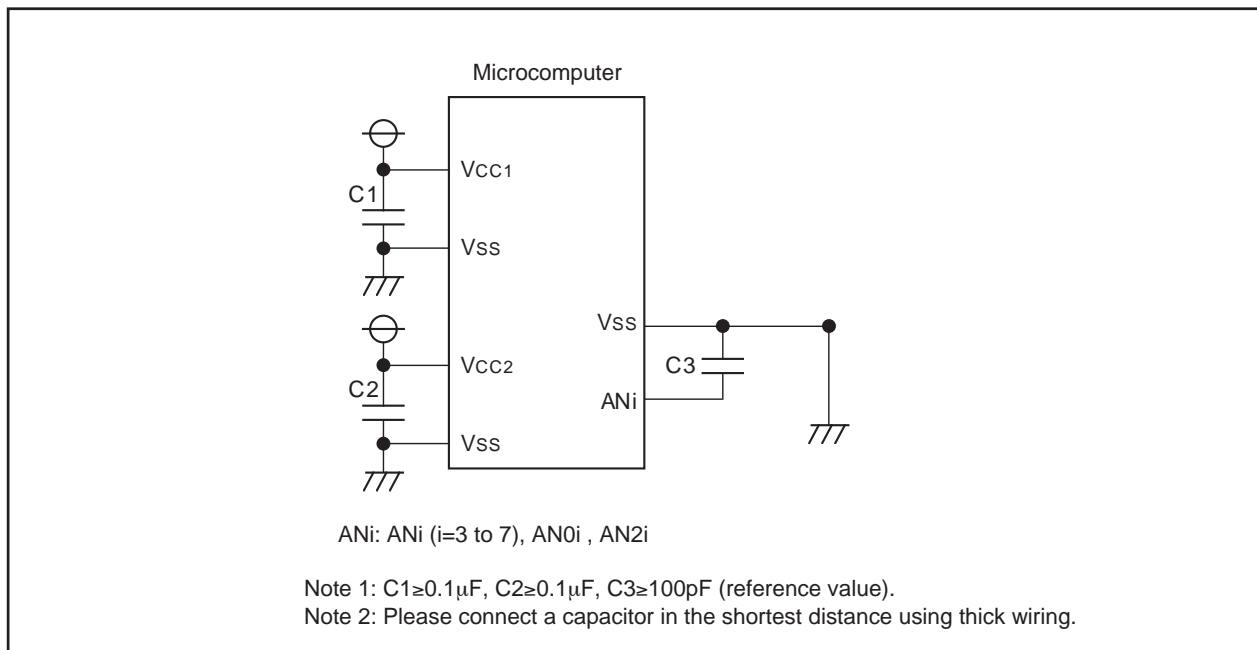


Figure 12.8. Example of noise measure processing of each pin

Multi-master I²C-BUS Interface 0 to 2

The multi-master I²C-BUS interface *i* (*i*=0 to 2) have each dedicated circuit and operate independently. The multi-master I²C-BUS interface *i* is a serial communications circuit, conforming to the Philips I²C-BUS data transfer format. This interface *i*, offering both arbitration lost detection and a synchronous functions, is useful for the multi-master serial communications.

Figures 13.1 and 13.2 show a block diagram of the multi-master I²C-BUS interface *i* and Table 13.1 shows multi-master I²C-BUS interface *i* functions.

This multi-master I²C-BUS interface *i* consists of the I²C_{*i*} address register, the I²C_{*i*} data shift register, the I²C_{*i*} clock control register, the I²C_{*i*} control register, the I²C_{*i*} status register, the I²C_{*i*} port selection register and other control circuits.

Table 13.1 Multi-master I²C-BUS Interface Functions

Item	Function
Format	In conformity with Philips I ² C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	In conformity with Philips I ² C-BUS standard: Master transmission Master reception Slave transmission Slave reception
SCL clock frequency	16.1 kHz to 400 kHz (at BCLK = 16 MHz)
Power supply voltage on bus line	(SCL1/SDA1), (SCL3/SDA3), (SCL5/SDA5), (SCL6/SDA6) : 3.3V (SCL2/SDA2), (SCL4/SDA4) : 3.3V or 5V

Note : We are not responsible for any third party's infringement of patent rights or other rights attributable to the use of the control function (bits 1 and 0 of the I²C control register at address 02D9₁₆) for connections between the I²C-BUS interface 0, 1 and ports (SCL1, SCL3, SCL5, SCL6, SDA1, SDA3, SDA5, SDA6).

Multi-master I²C-BUS Interface Ports

Communication control is more possible for I²C-BUS0 to I²C-BUS2 than the following pin. Please choose the pin used by register set up.

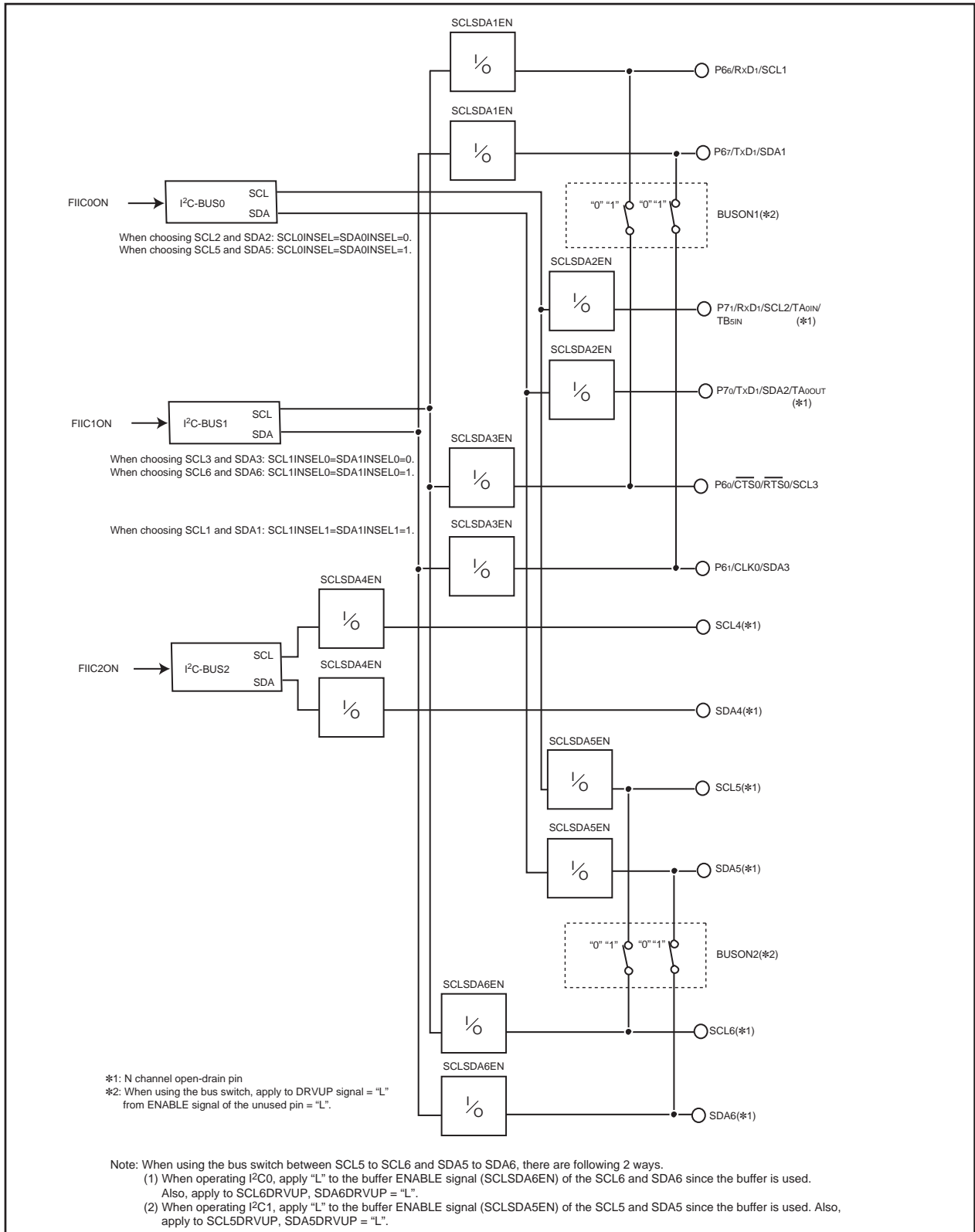


Figure 13.1 Block diagram of multi-master I²C-BUS interface i (i=0 to 2)

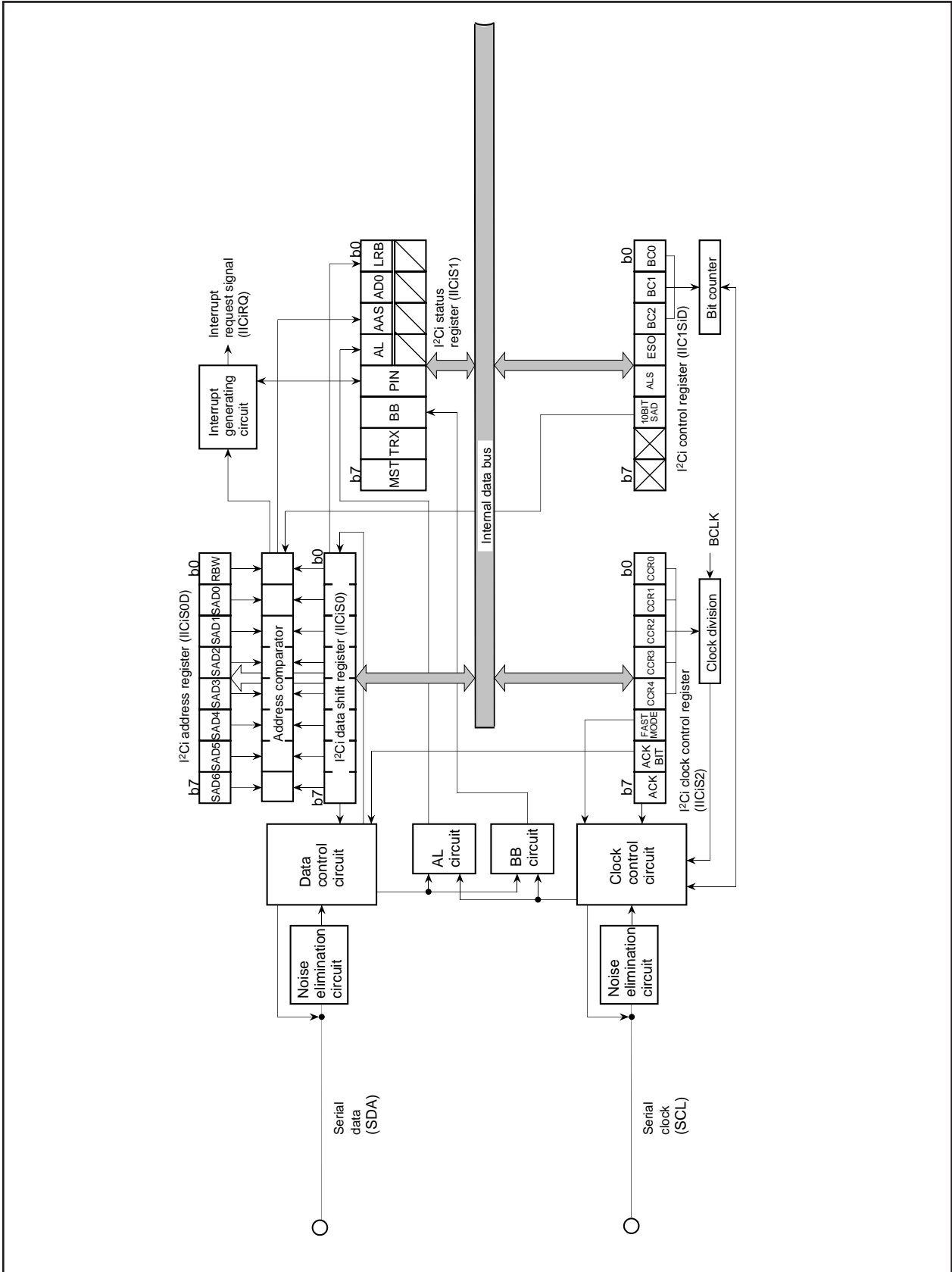


Fig. 13.2 Block Diagram of Multi-master I²C-BUS Interface i (i = 0 to 2)

(1) Reserved register

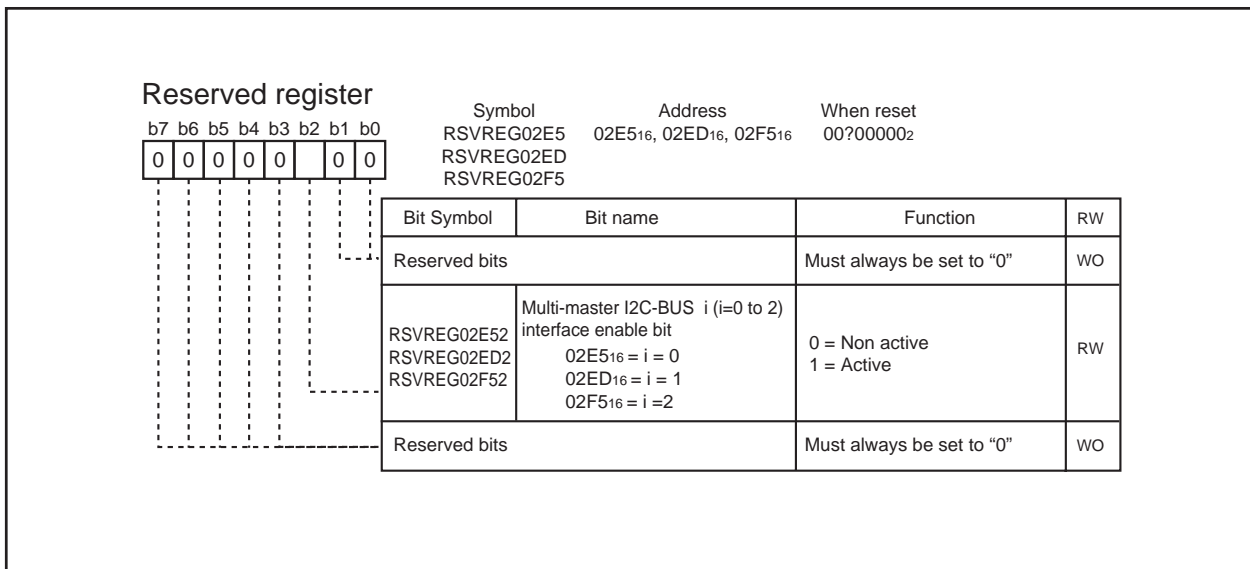


Fig. 13.3 Reserved register

(2) I²Ci data shift register, I²Ci transmit buffer register (i = 0 to 2)

The I²Ci data shift register is an 8-bit shift register to store receive data and write transmit data.

When transmit data is written into this register, it is transferred to the outside from bit 7 in synchronization with the SCL clock, and each time one-bit data is output, the data of this register are shifted one bit to the left. When data is received, it is input to this register from bit 0 in synchronization with the SCL clock, and each time one-bit data is input, the data of this register are shifted one bit to the left.

The I²Ci data shift register is in a write enable status only when the ESO bit of the I²Ci control register is "1." The bit counter is reset by a write instruction to the I²Ci data shift register. When both the ESO bit and the MST bit of the I²Ci status register are "1," the SCL is output by a write instruction to the I²Ci data shift register. Reading data from the I²Ci data shift register is always enabled regardless of the ESO bit value.

The I²Ci transmit buffer register is a register to store transmit data (slave address) to the I²Ci data shift register before RESTART condition generation. That is, in master, transmit data written to the I²Ci transmit buffer register is written to the I²Ci data shift register simultaneously. However, the SCL is not output. The I²Ci transmit buffer register can be written only when the ESO bit is "1," reading data from the I²Ci transmit buffer register is disabled regardless of the ESO bit value.

Notes 1: To write data into the I²Ci data shift register or the I²Ci transmit buffer register after the MST bit value changes from "1" to "0" (slave mode), keep an interval of 20 BCLK or more.

2: To generate START/RESTART condition after the I²Ci data shift register or the I²Ci transmit buffer register is written, keep an interval of 4 BCLK or more.

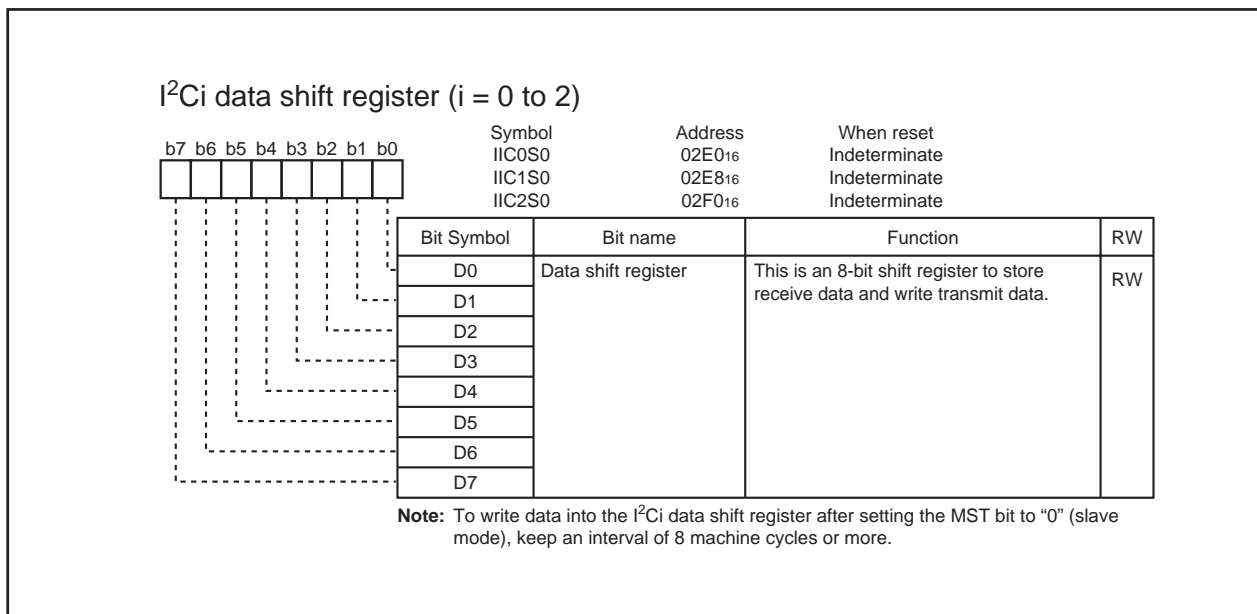


Fig. 13.4 I²Ci data shift register (i = 0 to 2)

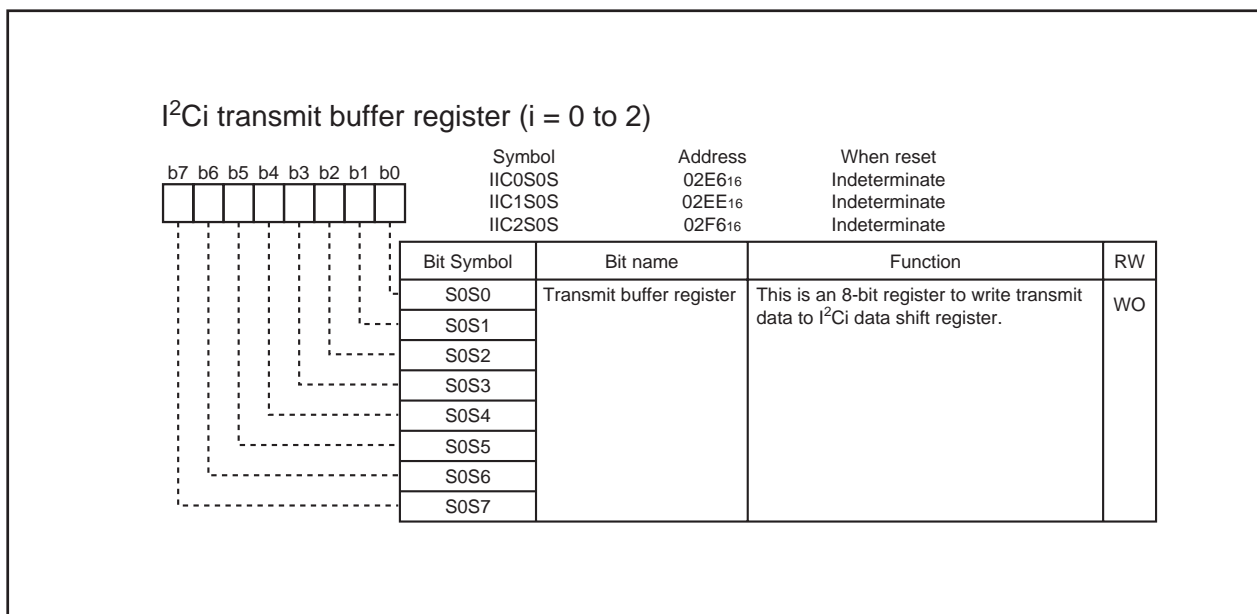


Fig. 13.5 I²Ci transmit buffer register (i = 0 to 2)

(3) I²Ci address register (i = 0 to 2)

The I²Ci address register consists of a 7-bit slave address and a $\overline{\text{read/write}}$ bit. In the addressing mode, the slave address written in this register is compared with the address data to be received immediately after the START condition are detected.

■ **Bit 0: $\overline{\text{read/write}}$ bit (RBW)**

Not used when comparing addresses, in the 7-bit addressing mode. In the 10-bit addressing mode, the first address data to be received is compared with the contents (SAD6 to SAD0 + RBW) of the I²Ci address register.

The RBW bit is cleared to “0” automatically when the stop condition is detected.

■ **Bits 1 to 7: slave address (SAD0 to SAD6)**

These bits store slave addresses. Regardless of the 7-bit addressing mode and the 10-bit addressing mode, the address data transmitted from the master is compared with the contents of these bits.

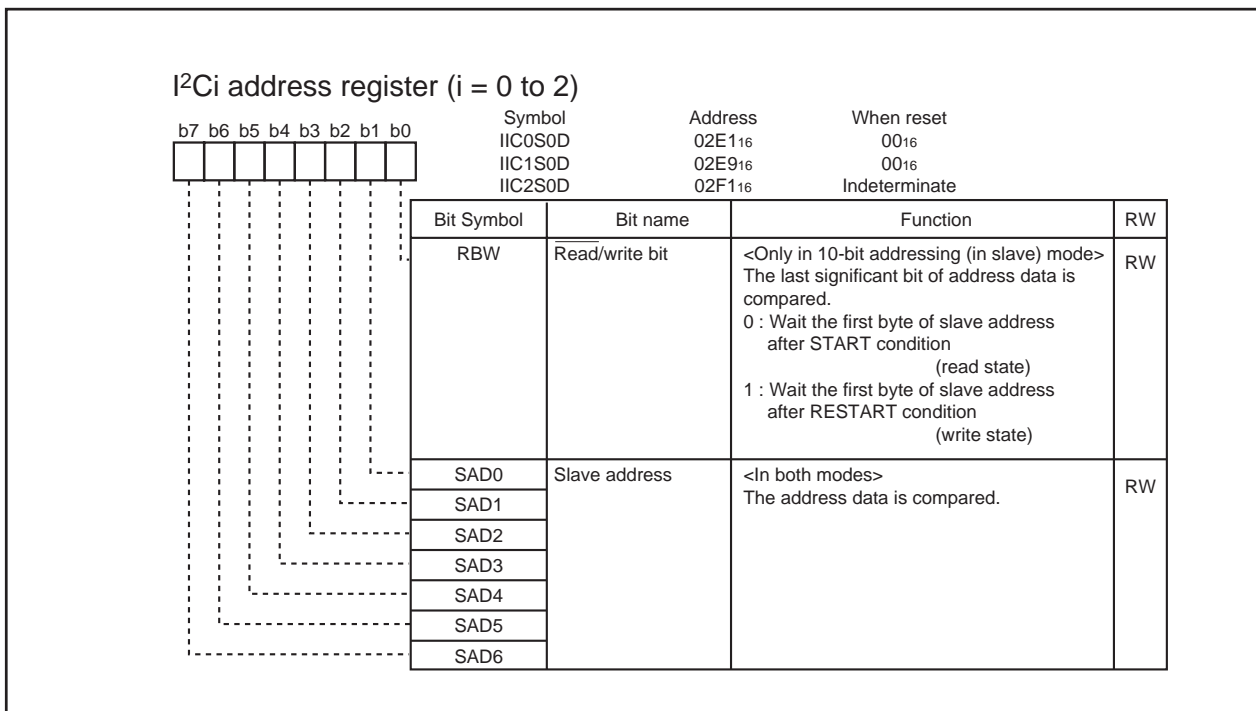


Fig. 13.6 I²Ci address register (i = 0 to 2)

(4) I²Ci clock control register (i = 0 to 2)

The I²Ci clock control register is used to set ACK control, SCL mode and SCL frequency.

■ Bits 0 to 4: SCL frequency control bits (CCR0–CCR4)

These bits control the SCL frequency.

■ Bit 5: SCL mode specification bit (FAST MODE)

This bit specifies the SCL mode. When this bit is set to “0,” the standard clock mode is set. When the bit is set to “1,” the high-speed clock mode is set.

■ Bit 6: ACK bit (ACK BIT)

This bit sets the SDA status when an ACK clock* is generated. When this bit is set to “0,” the ACK return mode is set and SDA goes to LOW at the occurrence of an ACK clock. When the bit is set to “1,” the ACK non-return mode is set. The SDA is held in the HIGH status at the occurrence of an ACK clock.

However, when the slave address matches the address data in the reception of address data at ACK BIT = “0,” the SDA is automatically made LOW (ACK is returned). If there is a mismatch between the slave address and the address data, the SDA is automatically made HIGH (ACK is not returned).

*ACK clock: Clock for acknowledgement

■ Bit 7: ACK clock bit (ACK)

This bit specifies a mode of acknowledgment which is an acknowledgment response of data transmission. When this bit is set to “0,” the no ACK clock mode is set. In this case, no ACK clock occurs after data transmission. When the bit is set to “1,” the ACK clock mode is set and the master generates an ACK clock upon completion of each 1-byte data transmission. The device for transmitting address data and control data releases the SDA at the occurrence of an ACK clock (make SDA HIGH) and receives the ACK bit generated by the data receiving device.

Note: Do not write data into the I²Ci clock control register during transmission. If data is written during transmission, the I²Ci clock generator is reset, so that data cannot be transmitted normally.

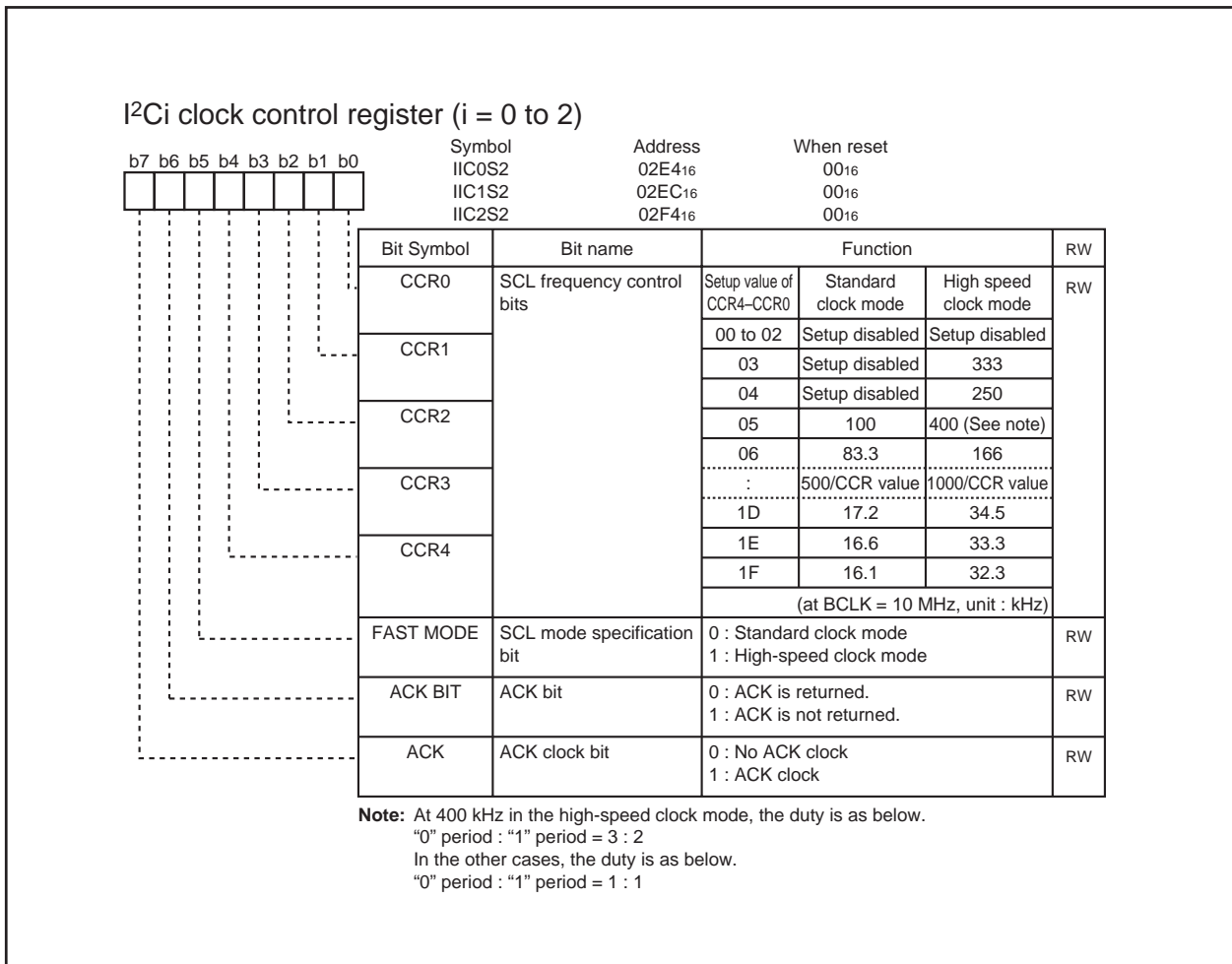


Fig. 13.7 I²Ci clock control register (i = 0 to 2)

(5) I²Ci control register (i = 0 to 2)

The I²Ci control register controls the data communication format.

■ Bits 0 to 2: bit counter (BC0–BC2)

These bits decide the number of bits for the next 1-byte data to be transmitted. An interrupt request signal occurs immediately after the number of bits specified with these bits are transmitted.

When a START condition is received, these bits become “0002” and the address data is always transmitted and received in 8 bits.

Note: When the bit counter value = “1112,” a STOP condition and START condition cannot be waited.

■ Bit 3: I²C-BUS interface i use enable bit (ESO)

This bit enables usage of the multimaster I²C-BUS interface i. When this bit is set to “0,” the use disable status is provided, so the SDA and the SCL become high-impedance. When the bit is set to “1,” use of the interface is enabled.

When ESO = “0,” the following is performed.

- PIN = “1,” BB = “0” and AL = “0” are set (they are bits of the I²Ci status register).
- Writing data to the I²Ci data shift register and the I²Ci transmit buffer register is disabled.

■ Bit 4: data format selection bit (ALS)

This bit decides whether or not to recognize slave addresses. When this bit is set to “0,” the addressing format is selected, so that address data is recognized. When a match is found between a slave address and address data as a result of comparison or when a general call (refer to “(6) I²Ci status register,” bit 1) is received, transmission processing can be performed. When this bit is set to “1,” the free data format is selected, so that slave addresses are not recognized.

■ Bit 5: addressing format selection bit (10BIT SAD)

This bit selects a slave address specification format. When this bit is set to “0,” the 7-bit addressing format is selected. In this case, only the high-order 7 bits (slave address) of the I²Ci address register are compared with address data. When this bit is set to “1,” the 10-bit addressing format is selected, all the bits of the I²Ci address register are compared with address data.

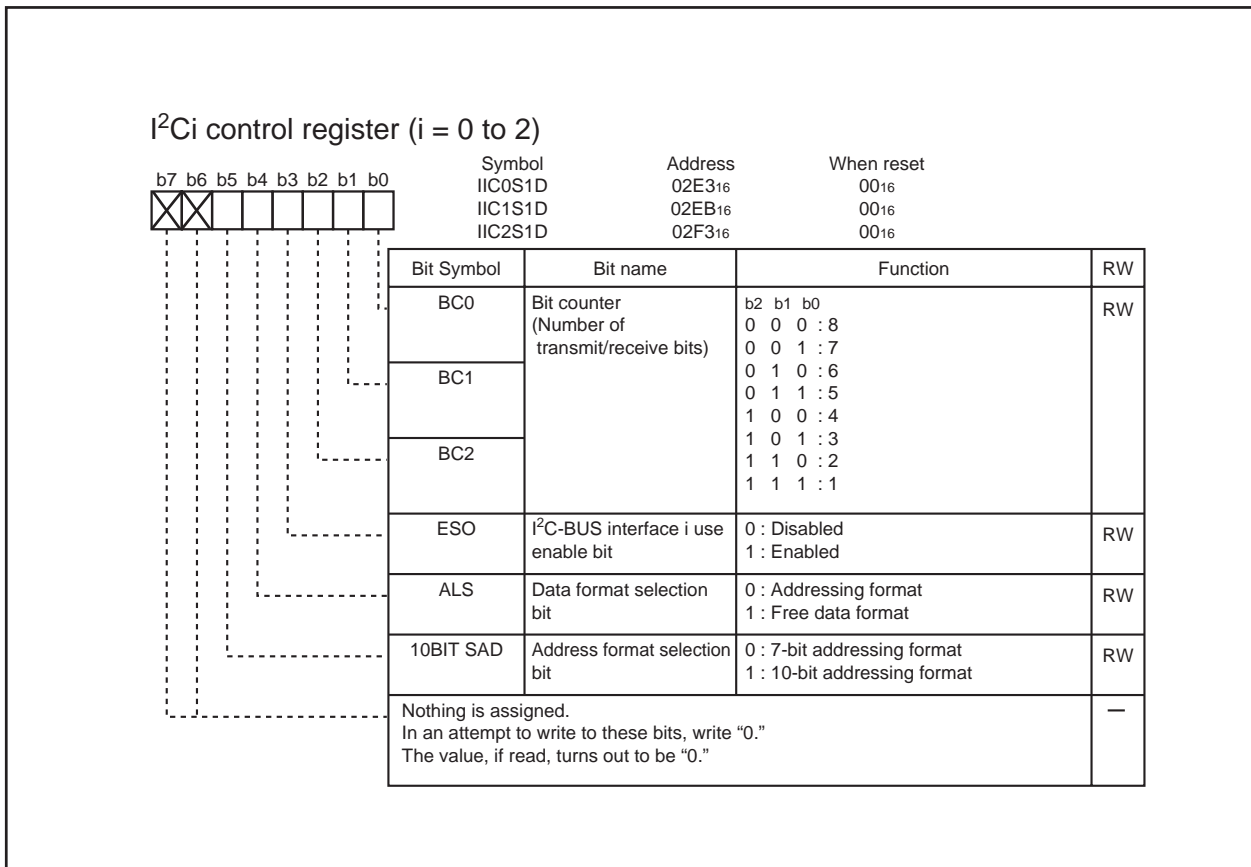


Fig. 13.8 I²C_i control register (i = 0 to 2)

(6) I²Ci status register (i = 0 to 2)

The I²Ci status register controls the I²C-BUS interface i status. Bits 0 to 3, 5 are read-only bits and bits 4, 6, 7 can be read out and written to.

■ Bit 0: last receive bit (LRB)

This bit stores the last bit value of received data and can also be used for ACK receive confirmation. If ACK is returned when an ACK clock occurs, the LRB bit is set to "0." If ACK is not returned, this bit is set to "1." Except in the ACK mode, the last bit value of received data is input. The state of this bit is changed from "1" to "0" by executing a write instruction to the I²Ci data shift register or the I²Ci transmit buffer register.

■ Bit 1: general call detecting flag (AD0)

This bit is set to "1" when a general call* whose address data is all "0" is received in the slave mode. By a general call of the master device, every slave device receives control data after the general call. The AD0 bit is set to "0" by detecting the STOP condition or START condition.

*General call: The master transmits the general call address "0016" to all slaves.

■ Bit 2: slave address comparison flag (AAS)

This flag indicates a comparison result of address data.

<<In the slave receive mode, when the 7-bit addressing format is selected, this bit is set to "1" in one of the following conditions.>>

- The address data immediately after occurrence of a START condition matches the slave address stored in the high-order 7 bits of the I²Ci address register.
- A general call is received.

<<In the slave reception mode, when the 10-bit addressing format is selected, this bit is set to "1" with the following condition.>>

- When the address data is compared with the I²Ci address register (8 bits consists of slave address and RBW), the first bytes match.

<<The state of this bit is changed from "1" to "0" by executing a write instruction to the I²Ci data shift register or the I²Ci transmit buffer register.>>

■ Bit 3: arbitration lost* detecting flag (AL)

In the master transmission mode, when a device other than the microcomputer sets the SDA to "L", arbitration is judged to have been lost, so that this bit is set to "1." At the same time, the TRX bit is set to "0," so that immediately after transmission of the byte whose arbitration was lost is completed, the MST bit is set to "0." When arbitration is lost during slave address transmission, the TRX bit is set to "0" and the reception mode is set. Consequently, it becomes possible to receive and recognize its own slave address transmitted by another master device.

<<This bit changes "1" to "0" by writing instruction to I²Ci data shift register or I²Ci transmit buffer register.>>

*Arbitration lost: The status in which communication as a master is disabled.

■ Bit 4: I²C-BUS interface i interrupt request bit (PIN)

This bit generates an interrupt request signal. Each time 1-byte data is transmitted, the state of the PIN bit changes from “1” to “0.” At the same time, an interrupt request signal is sent to the CPU. The PIN bit is set to “0” in synchronization with a falling edge of the last clock (including the ACK clock) of an internal clock and an interrupt request signal occurs in synchronization with a falling edge of the PIN bit. When detecting the STOP condition in slave, the multi-master I²C-BUS interface interrupt request bit (IR) is set to “1” (interrupt requested) regardless of falling of PIN bit. When the PIN bit is “0,” the SCL is kept in the “0” state and clock generation is disabled. Figure 13.10 shows an interrupt request signal generating timing chart.

The PIN bit is set to “1” in any one of the following conditions.

- Writing “1” to the PIN bit
- Executing a write instruction to the I²Ci data shift register or the I²Ci transmit buffer register (See note).
- When the ESO bit is “0”
- At reset

Note: It takes 12 BCLK cycles or more until PIN bit becomes “1” after write instructions are executed to these registers.

The conditions in which the PIN bit is set to “0” are shown below:

- Immediately after completion of 1-byte data transmission (including when arbitration lost is detected)
- Immediately after completion of 1-byte data reception
- In the slave reception mode, with ALS = “0” and immediately after completion of slave address or general call address reception
- In the slave reception mode, with ALS = “1” and immediately after completion of address data reception

■ Bit 5: bus busy flag (BB)

This bit indicates the status of use of the bus system. When this bit is set to “0,” this bus system is not busy and a START condition can be generated. When this bit is set to “1,” this bus system is busy and the occurrence of a START condition is disabled by the START condition duplication prevention function (See note).

This flag can be written by software only in the master transmission mode. In the other modes, this bit is set to “1” by detecting a START condition and set to “0” by detecting a STOP condition. When the ESO bit of the I²Ci control register is “0” and at reset, the BB flag is kept in the “0” state.

■ Bit 6: communication mode specification bit (transfer direction specification bit: TRX)

This bit decides the direction of transfer for data communication. When this bit is “0,” the reception mode is selected and the data of a transmitting device is received. When the bit is “1,” the transmission mode is selected and address data and control data are output into the SDA in synchronization with the clock generated on the SCL.

When the ALS bit of the I²Ci control register is “0” in the slave reception mode is selected, the TRX bit is set to “1” (transmit) if the least significant bit (R/ \bar{W} bit) of the address data transmitted by the master is “1.” When the ALS bit is “0” and the R/ \bar{W} bit is “0,” the TRX bit is cleared to “0” (receive).

The TRX bit is cleared to “0” in one of the following conditions.

- When arbitration lost is detected.
- When a STOP condition is detected.
- When occurrence of a START condition is disabled by the START condition duplication prevention function (Note).
- With MST = “0” and when a START condition is detected.
- With MST = “0” and when ACK non-return is detected.
- At reset

■ **Bit 7: Communication mode specification bit (master/slave specification bit: MST)**

This bit is used for master/slave specification for data communication. When this bit is “0,” the slave is specified, so that a START condition and a STOP condition generated by the master are received, and data communication is performed in synchronization with the clock generated by the master. When this bit is “1,” the master is specified and a START condition and a STOP condition are generated, and also the clocks required for data communication are generated on the SCL.

The MST bit is cleared to “0” in one of the following conditions.

- Immediately after completion of 1-byte data transmission when arbitration lost is detected
- When a STOP condition is detected.
- When occurrence of a START condition is disabled by the START condition duplication preventing function (See note).
- At reset

Note: The START condition duplication prevention function disables the following: the START condition generation; bit counter reset, and SCL output with the generation. This bit is valid from setting of BB flag to the completion of 1-byte transmission/reception (occurrence of transmission/reception interrupt request) <IICIRQ>.

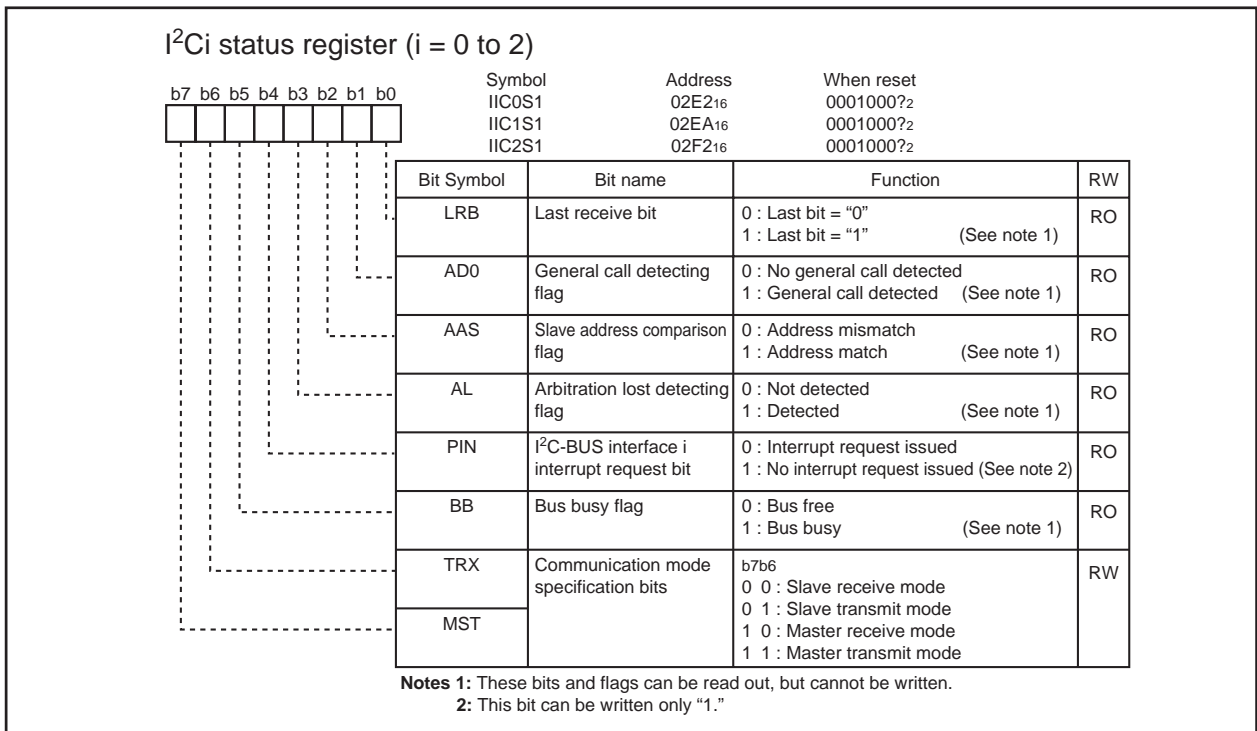


Fig. 13.9 I²Ci status register (i = 0 to 2)

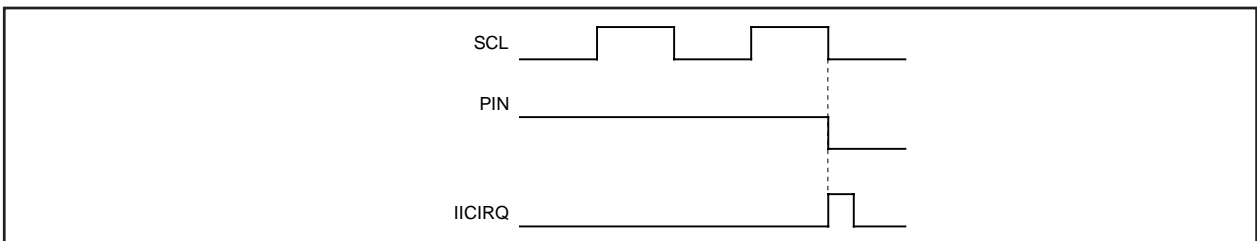


Fig. 13.10 Interrupt request signal generation timing

(7) START condition generation method

When the ESO bit of the I²Ci control register is “1,” execute a write instruction to the I²Ci status register to set the MST, TRX and BB bits to “1.” A START condition will then be generated. After that, the bit counter becomes “0002” and an SCL for 1 byte is output. The START condition generation timing and BB bit set timing are different in the standard clock mode and the high-speed clock mode. Refer to Figure 13.11 for the START condition generation timing diagram, and Table 13.2 for the START condition/STOP condition generation timing table.

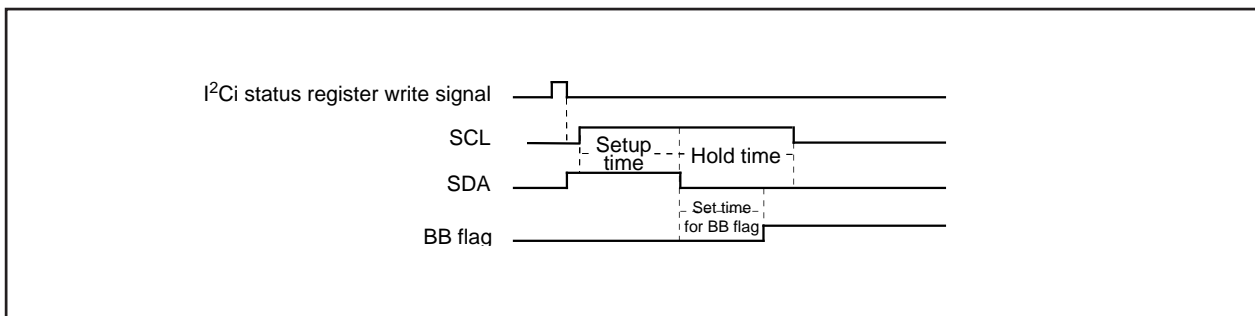


Fig. 13.11 START condition generation timing diagram

(8) STOP condition generation method

When the ESO bit of the I²Ci control register is “1,” execute a write instruction to the I²Ci status register for setting the MST bit and the TRX bit to “1” and the BB bit to “0.” A STOP condition will then be generated. The STOP condition generation timing and the BB flag reset timing are different in the standard clock mode and the high-speed clock mode. Refer to Figure 13.12 for the STOP condition generation timing diagram, and Table 13.2 for the START condition/STOP condition generation timing table.

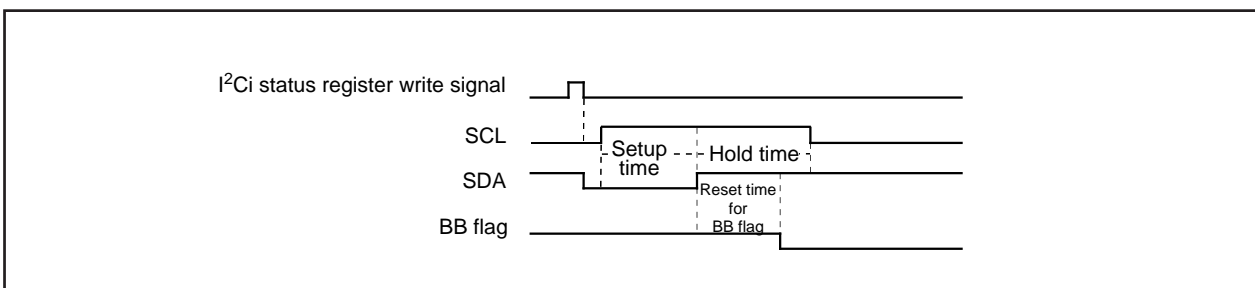


Fig. 13.12 STOP condition generation timing diagram

Table 13.2 START condition/STOP condition generation timing table

Item	Standard Clock Mode	High-speed Clock Mode
Setup time (Min.)	5.6 μ s	2.1 μ s
Hold time (Min.)	4.8 μ s	2.3 μ s
Set/reset time for BB flag	3.5 μ s	0.75 μ s

(9) START/STOP condition detect conditions

The START/STOP condition detect conditions are shown in Figure 13.13 and Table 13.3. Only when the 3 conditions of Table 13.3 are satisfied, a START/STOP condition can be detected.

Note: When a STOP condition is detected in the slave mode (MST = 0), an interrupt request signal <IICIRQ> is generated to the CPU.

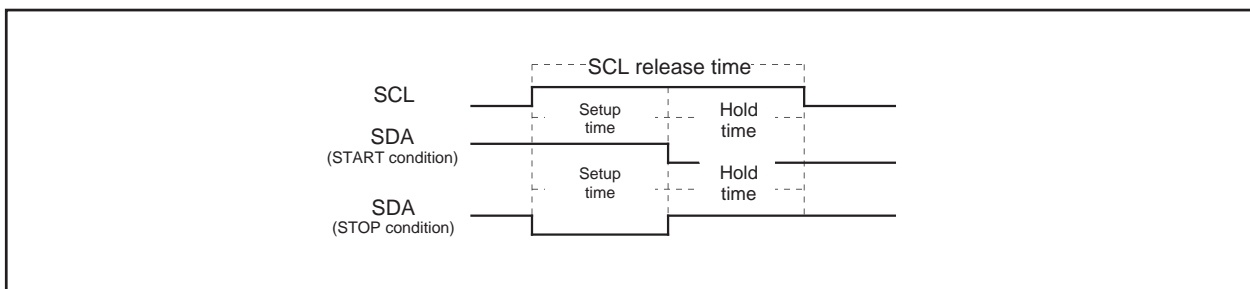


Fig. 13.13 START condition/STOP condition detect timing diagram

Table 13.3 START condition/STOP condition detect conditions

Standard Clock Mode	High-speed Clock Mode
6.5 μs < SCL release time	1.0 μs < SCL release time
3.25 μs < Setup time	0.5 μs < Setup time
3.25 μs < Hold time	0.5 μs < Hold time

(10) Address data communication

There are two address data communication formats, namely, 7-bit addressing format and 10-bit addressing format. The respective address communication formats is described below.

■ 7-bit addressing format

To meet the 7-bit addressing format, set the 10BIT SAD bit of the I²Ci control register to "0." The first 7-bit address data transmitted from the master is compared with the high-order 7-bit slave address stored in the I²Ci address register. At the time of this comparison, address comparison of the RBW bit of the I²Ci address register is not made. For the data transmission format when the 7-bit addressing format is selected, refer to Figure 13.14, (1) and (2).

■ 10-bit addressing format

To meet the 10-bit addressing format, set the 10BIT SAD bit of the I²Ci control register to "1." An address comparison is made between the first-byte address data transmitted from the master and the 7-bit slave address stored in the I²Ci address register. At the time of this comparison, an address comparison between the RBW bit of the I²Ci address register and the R/ \bar{W} bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the R/ \bar{W} bit which is the last bit of the address data not only specifies the direction of communication for control data but also is processed as an address data bit.

When the first-byte address data matches the slave address, the AAS bit of the I²Ci status register is set to "1." After the second-byte address data is stored into the I²Ci data shift register, make an address comparison between the second-byte data and the slave address by software. When the address data of the 2nd bytes matches the slave address, set the RBW bit of the I²Ci address register to "1" by software. This processing can match the 7-bit slave address and R/ \bar{W} data, which are received after a RESTART condition is detected, with the value of the I²Ci address register. For the data transmission format when the 10-bit addressing format is selected, refer to Figure 13.14, (3) and (4).

(11) Example of Master Transmission

An example of master transmission in the standard clock mode, at the SCL frequency of 100 kHz and in the ACK return mode is shown below.

- ① Set a slave address in the high-order 7 bits of the I²Ci address register and “0” in the RBW bit.
- ② Set the ACK return mode and SCL = 100 kHz by setting “85₁₆” in the I²Ci clock control register.
- ③ Set “10₁₆” in the I²Ci status register and hold the SCL at the HIGH.
- ④ Set a communication enable status by setting “08₁₆” in the I²Ci control register.
- ⑤ Set the address data of the destination of transmission in the high-order 7 bits of the I²Ci data shift register and set “0” in the least significant bit.
- ⑥ Set “F0₁₆” in the I²Ci status register to generate a START condition. At this time, an SCL for 1 byte and an ACK clock automatically occurs.
- ⑦ Set transmit data in the I²Ci data shift register. At this time, an SCL and an ACK clock automatically occurs.
- ⑧ When transmitting control data of more than 1 byte, repeat step ⑦.
- ⑨ Set “D0₁₆” in the I²Ci status register. After this, if ACK is not returned or transmission ends, a STOP condition will be generated.

(12) Example of Slave Reception

An example of slave reception in the high-speed clock mode, at the SCL frequency of 400 kHz, in the ACK non-return mode, using the addressing format, is shown below.

- ① Set a slave address in the high-order 7 bits of the I²Ci address register and “0” in the RBW bit.
- ② Set the no ACK clock mode and SCL = 400 kHz by setting “25₁₆” in the I²Ci clock control register.
- ③ Set “10₁₆” in the I²Ci status register and hold the SCL at the HIGH.
- ④ Set a communication enable status by setting “08₁₆” in the I²Ci control register.
- ⑤ When a START condition is received, an address comparison is made.
- ⑥
 - When all transmitted address are “0” (general call):
AD0 of the I²Ci status register is set to “1” and an interrupt request signal occurs.
 - When the transmitted addresses match the address set in ①:
ASS of the I²Ci status register is set to “1” and an interrupt request signal occurs.
 - In the cases other than the above:
AD0 and AAS of the I²Ci status register are set to “0” and no interrupt request signal occurs.
- ⑦ Set dummy data in the I²Ci data shift register.
- ⑧ When receiving control data of more than 1 byte, repeat step ⑦.
- ⑨ When a STOP condition is detected, the communication ends.

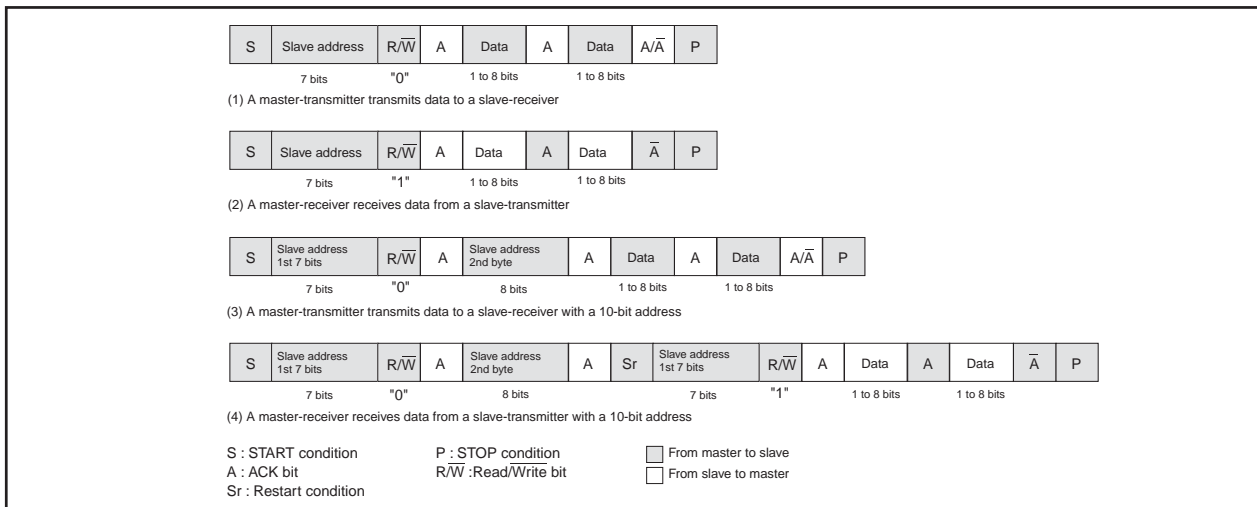


Fig. 13.14 Address data communication format

(13) Precautions when using multi-master I²C-BUS interface i

■ BCLK operation mode

Select the no-division mode.

■ Used instructions

Specify byte (.B) as data size to access multi-master I²C-BUS interface i-related registers.

■ Read-modify-write instruction

The precautions when the read-modify-write instruction such as BSET, BCLR etc. is executed for each register of the multi-master I²C-BUS interface i are described below.

• I²Ci data shift register (IICiS0)

When executing the read-modify-write instruction for this register during transfer, data may become a value not intended.

• I²Ci address register (IICiS0D)

When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become a value not intended. It is because hardware changes the $\overline{\text{read/write}}$ bit (RBW) at the above timing.

• I²Ci status register (IICiS1)

Do not execute the read-modify-write instruction for this register because all bits of this register are changed by hardware.

• I²Ci control register (IICiS1D)

When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended. Because hardware changes the bit counter (BC0–BC2) at the above timing.

• I²Ci clock control register (IICiS2)

The read-modify-write instruction can be executed for this register.

• I²Ci port selection register (IICiS2D)

Since the read value of high-order 4 bits is indeterminate, the read-modify-write instruction cannot be used.

• I²Ci transmit buffer register (IICiS0S)

Since the value of all bits is indeterminate, the read-modify-write instruction cannot be used.

■ START condition generating procedure using multi-master

```

:
FCLR      I                      (Interrupt disabled)
BTST     5, IICiS1              (BB flag confirming and branch process)
JC       BUSBUSY
BUSFREE:
MOV.B    SA, IICiS0             (Writing of slave address value <SA>)
NOP
NOP
NOP
NOP
MOV.B    #F0H, IICiS1          (Trigger of START condition generating)
FSET     I                      (Interrupt enabled)
:
BUSBUSY:
FSET     I                      (Interrupt enabled)
:

```

① Be sure to add NOP instruction X 4 between writing the slave address value and setting trigger of START condition generating shown the above procedure example.

② When using multi-master system, disable interrupts during the following three process steps:

- BB flag confirming
- Writing of slave address value
- Trigger of START condition generating

When the condition of the BB flag is bus busy, enable interrupts immediately.

When using single-master system, it is not necessary to disable interrupts above.

■ RESTART condition generating procedure

```

:
MOV.B    SA, IICiS0S           (Writing of slave address value <SA>) ———①
NOP
NOP
MOV.B    #F0H, IICiS1          (Trigger of RESTART condition generating)
:

```

① Use the I²Ci transmit buffer register to write the slave address value to the I²Ci data shift register. And also, be sure to add NOP instruction X 4.

■ Writing to I²Ci status register

Do not execute an instruction to set the PIN bit to “1” from “0” and an instruction to set the MST and TRX bits to “0” from “1” simultaneously. It is because it may enter the state that the SCL pin is released and the SDA pin is released after about one machine cycle. Do not execute an instruction to set the MST and TRX bits to “0” from “1” simultaneously when the PIN bit is “1.” It is because it may become the same as above.

■ Process of after STOP condition generating

Do not write data in the I²Ci data shift register (IICiS0) and the I²Ci status register (IICiS1) until the bus busy flag BB becomes “0” after generating the STOP condition in the master mode. It is because the STOP condition waveform might not be normally generated. Reading to the above registers do not have the problem.

Data Slicer

This microcomputer includes the data slicer function for the closed caption decoder (referred to as the CCD) and video ID (referred to as the ID1). This function takes out CC and ID1 (note 1) superimposed in the vertical blanking interval of a composite video signal. A composite video signal which makes the sync. tip's polarity negative is input to the CVIN pin.

When the data slicer function is not used, the data slicer circuit and the timing signal generating circuit can be cut off by setting bit 0 of the data slicer control register 1 (address 026016/030016) to "0." These settings can realize the low-power dissipation.

- Notes**
- 525i (480i)/525p (480p):ID1 data slice can be performed. No CC data slice at 525p (480p).
 - When there is no specification, it becomes the publication about 525i (480i) below.

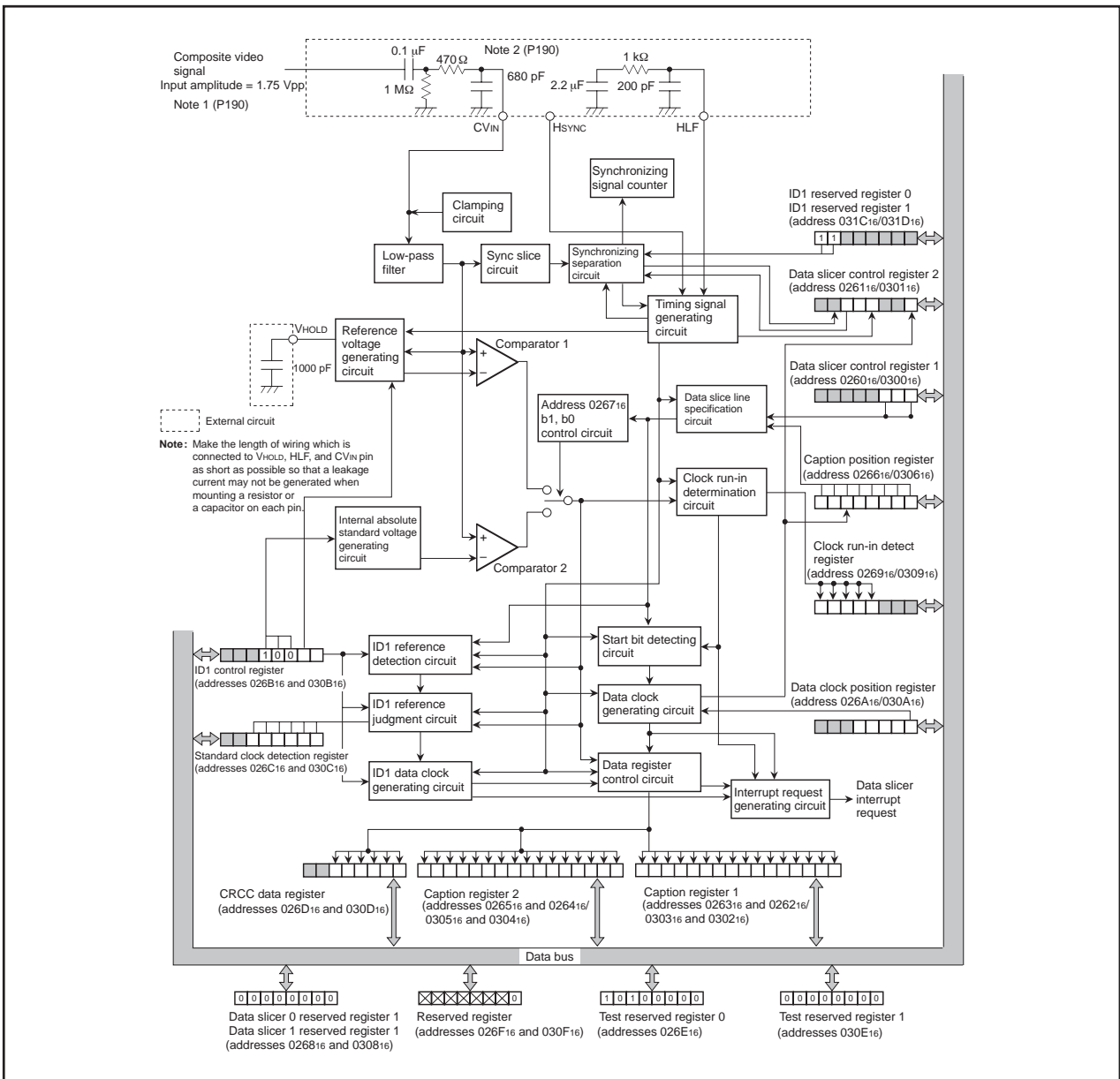


Figure 14.1 Data slicer block diagram

Note 1 : Set up the amplitude inputted from CVIN pin to satisfy the following conditions.

(1) Set up as below :

input amplitude + synchronized chip clamp potential < $V_{CCi} + 0.3 V$.

V_{cc_i} shows V_{cc} power supply pin voltage.

Sink tip clamp pin serves as $(43/120) \times V_{CCi}$.

Example) In the case of $V_{CCi} = 3.3V$ input amplitude = 2.0V

$$2.0V + 1.18 V = 3.18 V < 3.6 V = 3.3 V + 0.3 V$$

(2) Each signal level to input amplitude of CVIN pin is shown in Figure 14.2.

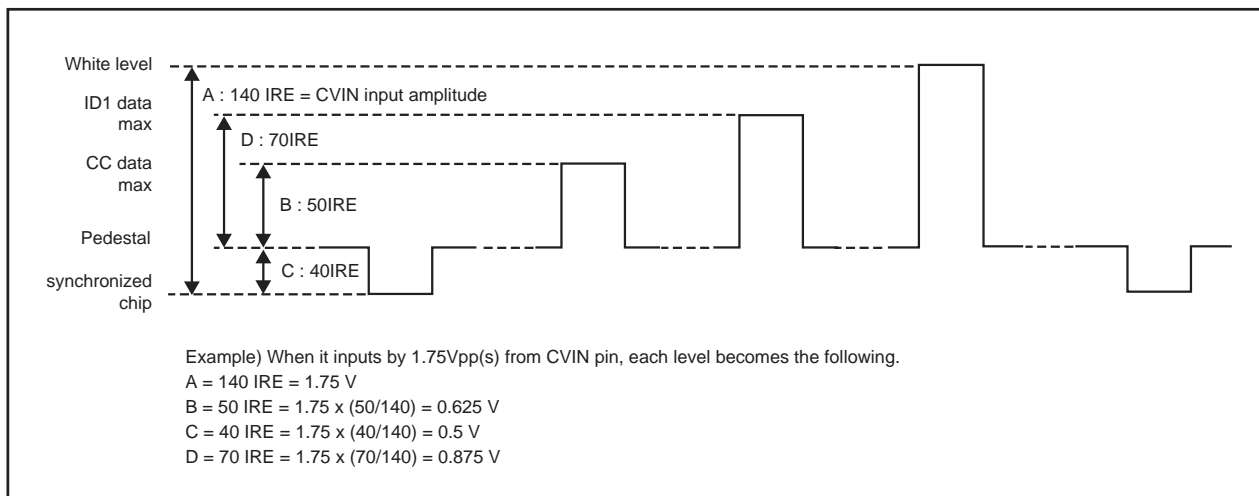


Figure 14.2 Each signal level to input amplitude of CVIN pin

Note 2 : External each constant shown in Figure 14.1 is an example, and is greatly influenced by video signal output impedance, substrate capacity, etc. on a system. Evaluate input amplitude and external each constant perfectly, and determine it.

Notes when not Using Data Slicer

When bit 0 of data slicer control register 1 (address 026016/030016) is "0," terminate the pins as shown in Figure 14.3

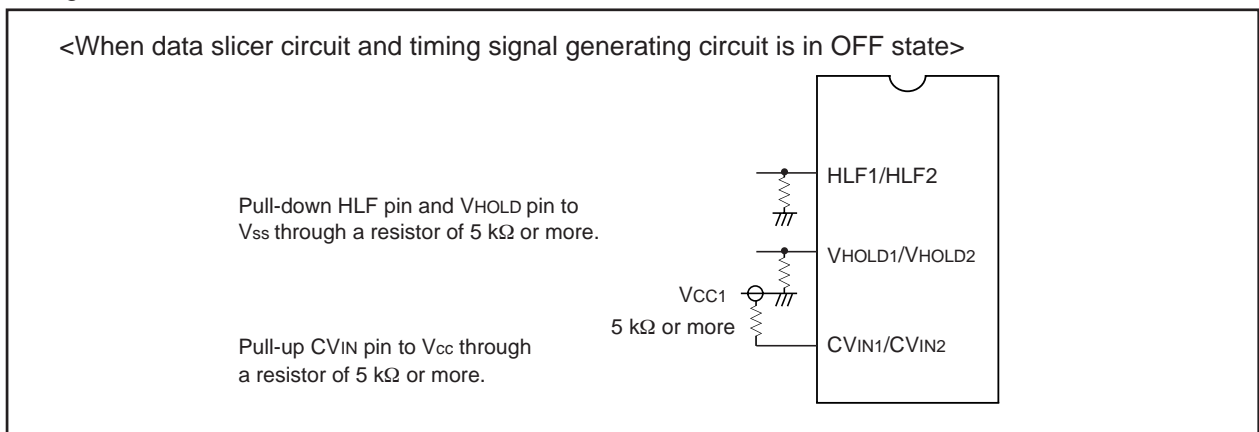


Figure 14.3 Termination of data slicer input/output pins when data slicer circuit and timing generating circuit is in OFF state

Figures 14.4 and 14.5 the data slicer control registers.

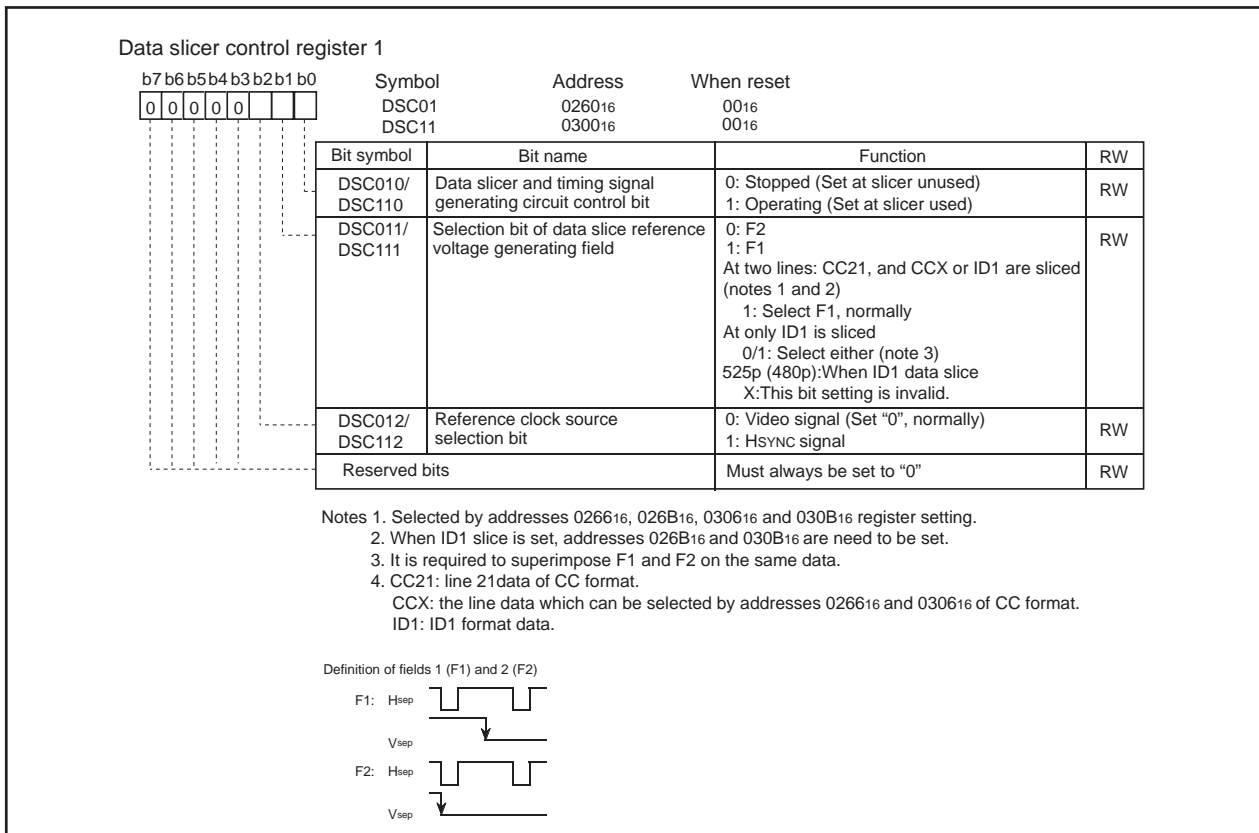


Figure 14.4 Data slicer control register 1

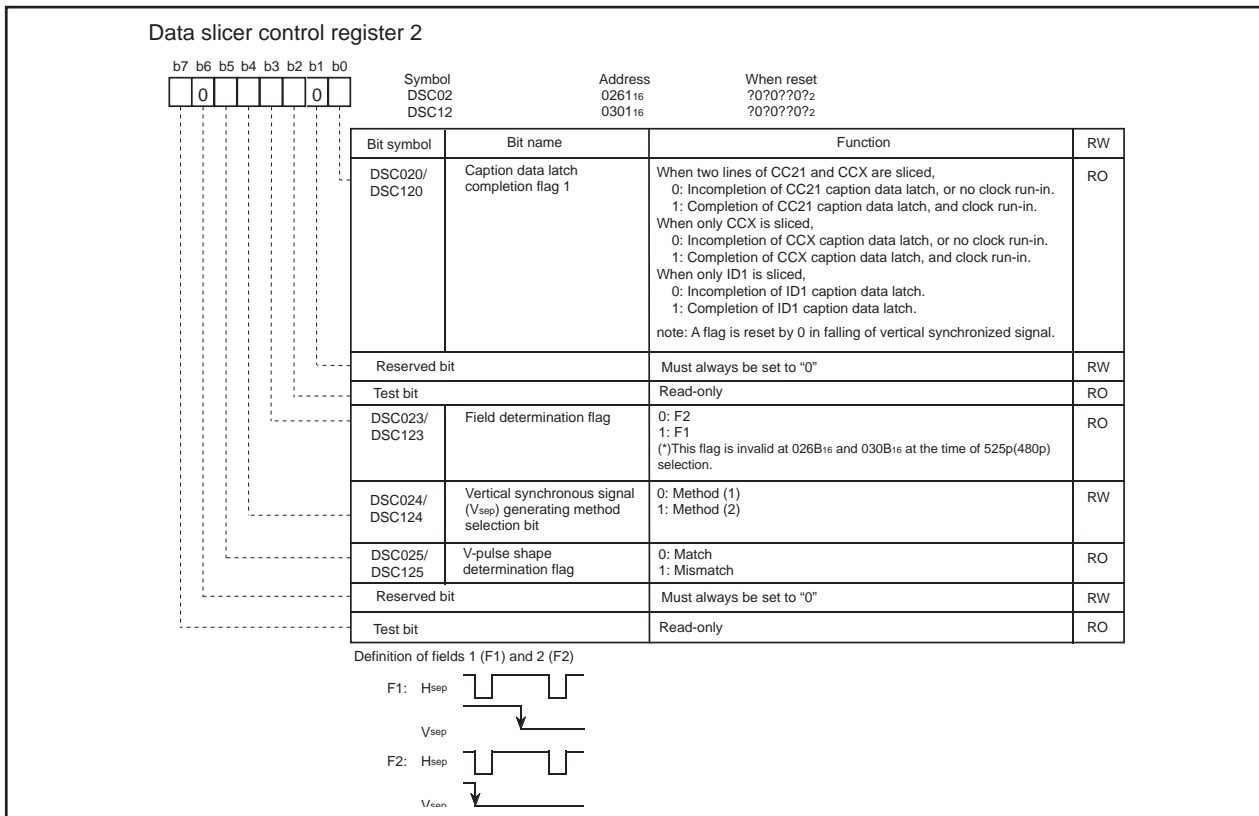


Figure 14.5 Data slicer control register 2

Clamping Circuit and Low-pass Filter

The clamp circuit clamps the sync. tip part of the composite video signal input from the CVIN pin. The low-pass filter attenuates the noise of clamped composite video signal. The CVIN pin to which composite video signal is input requires a capacitor (0.1 μF) coupling outside. Pull down the CVIN pin with a resistor of hundreds of kilohms to 1 MΩ. In addition, we recommend to install externally a simple low-pass filter using a resistor and a capacitor at the CVIN pin (refer to Figure 14.1 and notes).

Sync Slice Circuit

This circuit takes out a composite sync signal from the output signal of the low-pass filter. Set bit 6 and 7 to 11b of ID1 reserved register 0 and 1 (addresses 031C16 and 031D16) show in Fig 14.21.

Synchronous Signal Separation Circuit

This circuit separates a horizontal synchronous signal and a vertical synchronous signal from the composite sync signal taken out in the sync slice circuit.

(1) Horizontal synchronous signal (Hsep)

A one-shot horizontal synchronizing signal Hsep is generated at the falling edge of the composite sync signal.

(2) Vertical synchronous signal (Vsep)

As a Vsep signal generating method, it is possible to select one of the following 2 methods by using bit 4 of the data slicer control register 2 (address 026116/030116).

- Method 1 The “L” level width of the composite sync signal is measured. If this width exceeds a certain time, a Vsep signal is generated in synchronization with the rising of the timing signal immediately after this “L” level.
- Method 2 The “L” level width of the composite sync signal is measured. If this width exceeds a certain time, it is detected whether a falling of the composite sync signal exits or not in the “L” level period of the timing signal immediately after this “L” level. If a falling exists, a Vsep signal is generated in synchronization with the rising of the timing signal (refer to Figure 14.6).

Figure 14.6 shows a Vsep generating timing. The timing signal shown in the figure is generated from the reference clock which the timing generating circuit outputs.

Reading bit 5 of data slicer control register 2 permits determining the shape of the v-pulse portion of the composite sync signal. As shown in Figure 14.7, when the A level matches the B level, this bit is “0.” In the case of a mismatch, the bit is “1.”

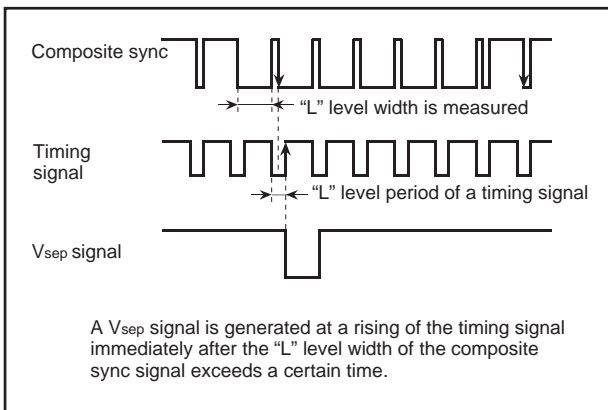


Figure 14.6 Vsep generating timing (method 2)

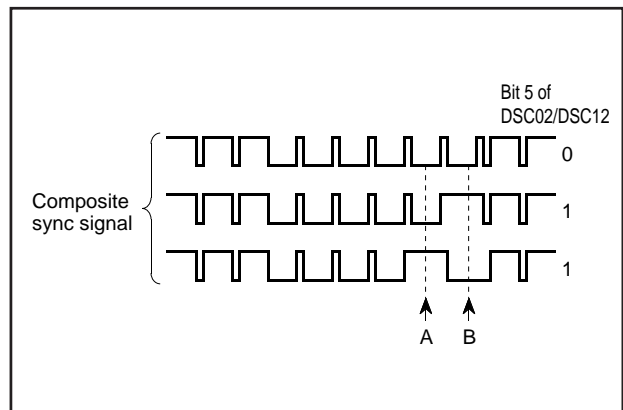


Figure 14.7 Determination of v-pulse waveform

Timing Signal Generating Circuit

This circuit generates a reference clock which is 832 times as large as the horizontal synchronous signal frequency. It also generates various timing signals on the basis of the reference clock, horizontal synchronous signal and vertical synchronizing signal. The circuit operates by setting bit 0 of data slicer control register 1 (address 0260₁₆/0300₁₆) to "1."

The reference clock is the HSYNC signal can be used as a count source instead of the composite sync signal. However, when the HSYNC signal is selected, the data slicer cannot be used. A count source of the reference clock can be selected by bit 2 of data slicer control register 1 (address 0260₁₆/0300₁₆).

For the pins HLF, connect a resistor and a capacitor as shown in Figure 14.1 Make the length of wiring which is connected to these pins as short as possible so that a leakage current may not be generated.

Note: It takes a few tens of milliseconds until the reference clock becomes stable after the data slicer and the timing signal generating circuit are started. In this period, various timing signals, H_{sep} signals and V_{sep} signals become unstable. For this reason, take stabilization time into consideration when programming.

Data Slice Line Specification Circuit

(1) Specification of data slice line

This circuit decides a line on which caption data is superimposed. The line 21 (fixed), 1 appropriate line for a period of 1 field (total 2 line for a period of 1 field), and both fields (F1 and F2) are sliced their data. The caption position register (address 0266₁₆/0306₁₆) is used for each setting (refer to Table 14.1).

The counter is reset at the falling edge of V_{sep} and is incremented by 1 every H_{sep} pulse. When the counter value matched the value specified by bits 4 to 0 of the caption position register, this H_{sep} is sliced.

The values of "00₁₆" to "1F₁₆" can be set in the caption position register (at setting only 1 appropriate line, refer to Table 14.1). Figure 14.8 shows the signals in the vertical blanking interval. Figure 14.9 shows the caption position register.

When slice ID1, set bits 0 to 4 of addresses 0266₁₆ and 0306₁₆ = 10000b.

525p (480p):When ID1 data slice, set up addresses 0266₁₆/0306₁₆ bit 4-0 = 00001b and the data clock position register (addresses 026A₁₆ and 030A₁₆) bit 6, and 5 = 01b.

(2) Specification of line to set slice voltage

When slice CC21 and CCX, the reference voltage for slicing (slice voltage) is generated for the clock run-in pulse in the particular line (refer to Table 14.1). The field to generate slice voltage is specified by bit 1 of data slicer control register 1. The line to generate slice voltage 1 field is specified by bits 6, 7 of the caption position register (refer to Table 14.1).

When slice ID1, set bit 6 and 7 of addresses 0266₁₆ and 0306₁₆ = 00b or 01b.

525p (480p):When ID1 data slice, set up the addresses 0266₁₆ and 0306₁₆ bit 7 and 6 = 01b.

(3) Field determination

The field determination flag can be read out by bit 3 of data slicer control register 2. This flag change at the falling edge of V_{sep}.

525p (480p):When ID1 data slice, this bit setting is invalid.

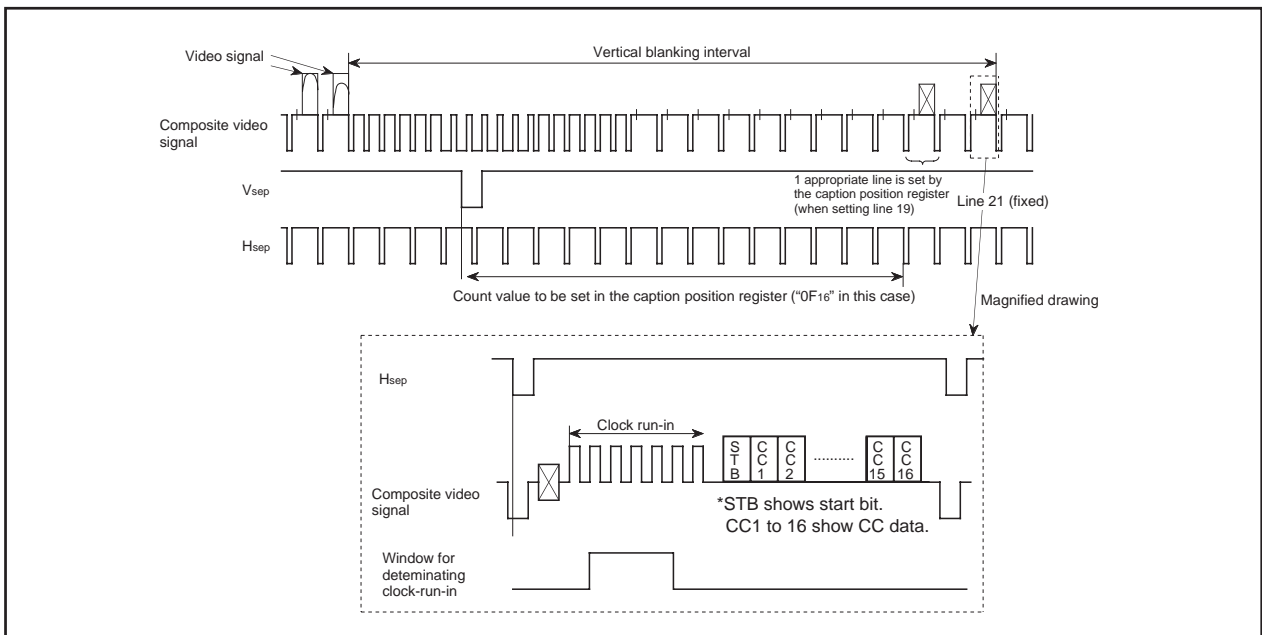


Figure 14.8 Signals in vertical blanking interval

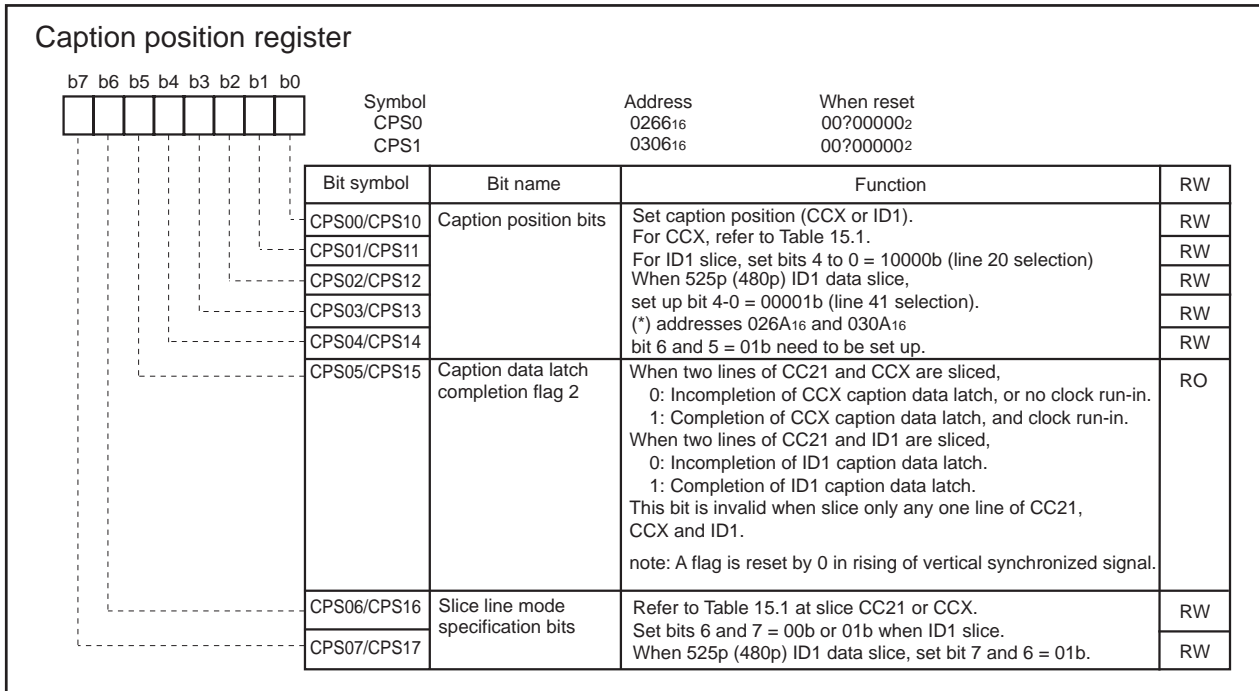


Figure 14.9 Caption position register

Table 14.1 Specification of data slice line

CPS0/CPS1		Field and Line to Be Sliced Data	Field and Line to Generate Slice Voltage
b7	b6		
0	0	<ul style="list-style-type: none"> Both fields of F1 and F2 Line 21 and a line specified by bits 4 to 0 of CPS0/CPS1 (total 2 lines) (See note 2) 	<ul style="list-style-type: none"> Field specified by bit 1 of DSC01/DSC11 Line 21 (total 1 line)
0	1	<ul style="list-style-type: none"> Both fields of F1 and F2 A line specified by bits 4 to 0 of CPS0/CPS1 (total 1 line) (See note 3) 	<ul style="list-style-type: none"> Field specified by bit 1 of DSC01/DSC11 A line specified by bits 4 to 0 of CPS0/CPS1 (total 1 line) (See note 3)
1	0	<ul style="list-style-type: none"> Both fields of F1 and F2 Line 21 (total 1 line) 	<ul style="list-style-type: none"> Field specified by bit 1 of DSC01/DSC11 Line 21 (total 1 line)
1	1	<ul style="list-style-type: none"> Both fields of F1 and F2 Line 21 and a line specified by bits 4 to 0 of CPS0/CPS1 (total 2 lines) (See note 2) 	<ul style="list-style-type: none"> Field specified by bit 1 of DSC01/DSC11 Line 21 and a line specified by bits 4 to 0 of CPS0/CPS1 (total 2 lines) (See note 2)

- Notes 1: DSC01/DSC11 is data slicer control register 1.
CPS0/CPS1 is caption position register.
2: Set the value of "00₁₆" – "10₁₆" to bits 4 to 0 of CPS0/CPS1.
3: Set the value of "00₁₆" – "1F₁₆" to bits 4 to 0 of CPS0/CPS1.

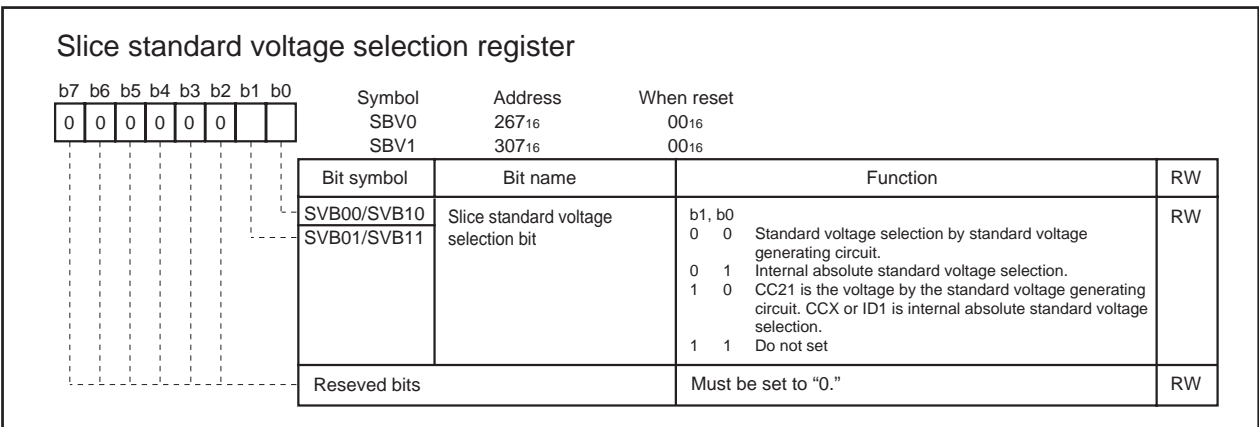


Figure 14.10 Slice standard voltage selection register

Reference Voltage Generating Circuit and Comparator

The composite video signal clamped by the clamping circuit is input to the reference voltage generating circuit and the comparator 1 and 2.

(1) Reference voltage generating circuit

This circuit generates a reference voltage (slice voltage) by using the amplitude of the clock run-in pulse in line specified by the data slice line specification circuit. Connect a capacitor between the VHOLD pin and the VSS pin, and make the length of wiring as short as possible so that a leakage current may not be generated.

Note: It takes a few tens of lines to generate slice voltage until the slice voltage becomes stable after the data slicer is started. In this period, the slice data becomes unstable. For this reason, take stabilization time into consideration when programming.

(2) Comparator 1

The comparator 1 compares the voltage of the composite video signal with the voltage (reference voltage) generated in the reference voltage generating circuit, and converts the composite video signal into a digital value.

(3) Comparator 2

The comparator 2 compares the absolute standard voltage generated inside from the voltage and power supply voltage of a composite video signal, and converts the composite video signal into a digital value.

CC Start Bit • ID1 Reference Bit Detection Circuit

This circuit detects a CC start bit • ID1 reference bit at line decided in the data slice line specification circuit.

In the case of CC start bit

- 1) Detect a clock run impulse at counting the input pulse of a data slice line.
- 2) When a clock run impulse is detected, the sampling clock outputted from a timing generating circuit detects a start bit pattern, and judge CC start bit.

In the case of ID1 reference bit

- 1) Detect ID1 reference bit all over the window generated after fixed time from Hsep in a timing signal generating circuit.

Clock Run-in Determination Circuit • ID1 Reference Bit Detection Circuit

Clock run in judging

By counting the number of pulses all over the specific window of a data slice line, it judges that it is clock run in. When it judges with having no clock run in, the completion flag of a caption data latch is not set to 1. Moreover, the number of standard clocks counted in clock run impulse 1 cycle is stored in the bits 7-3 of a clock run in detection register (addresses 0269₁₆/0309₁₆).

ID1 reference bit judging

The number of standard clocks counted during fixed of ID1 reference bit is stored in the bits 5-0 of a standard clock detection register (addresses 0269₁₆/the 030C₁₆). Read these bits after generating of data slicer interruption ("Interrupt Request Generating Circuit").

Clock run-in detection register is shown in Fig. 14.11, standard clock detection register is shown in Fig. 14.12.

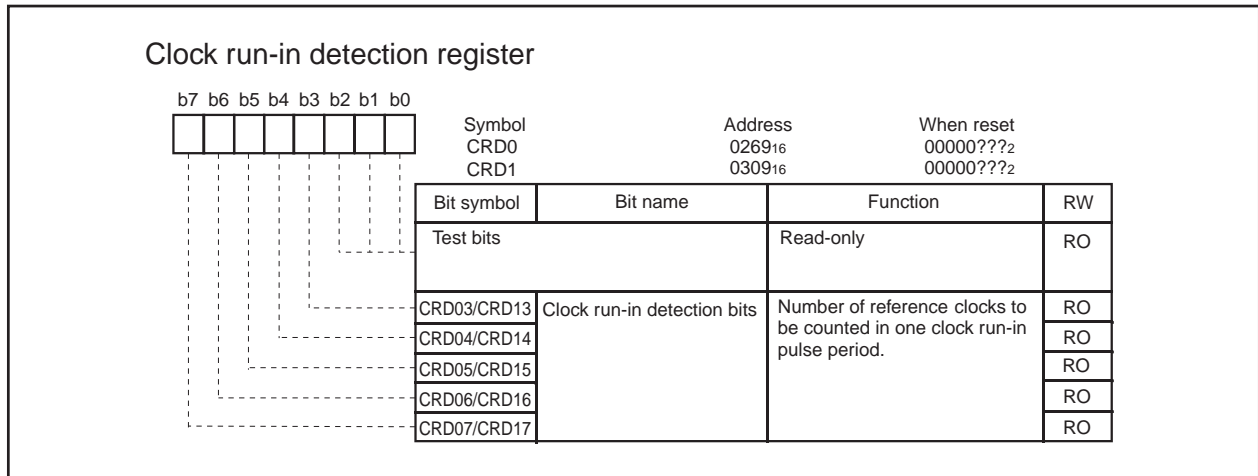


Figure 14.11 Clock run-in detection register

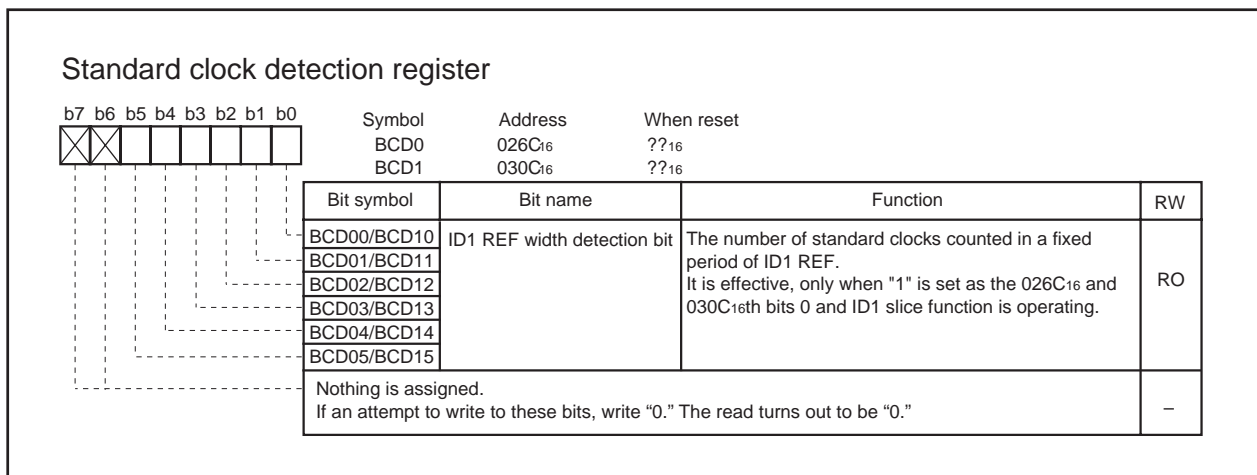


Figure 14.12 Standard clock detection register

Data Clock Generating Circuit

At the time of CC data slice

It synchronizes with CC start bit detected in CC start bit detection circuit, and a data clock is generated after the fixed offset set up by the data clock position register (addresses 026A₁₆/030A₁₆). A data clock is a clock for storing caption data in a caption register. When 16-bit data is stored in a caption register and judged in a clock run in judging circuit that has clock run in, the completion flag of a caption data latch is set.

A data clock position register is shown in Fig. 14.13.

At the time of ID1 data slice

The data clock which synchronized with ID1 reference bit is generated. With this data clock, the 6 bit data of the remaining CRCC is stored in a caption register for 14-bit data among 20-bit data at a CRCC data register (addresses 026D₁₆/030D₁₆). If 20-bit data is stored in each register, the completion flag of a caption data latch will be set.

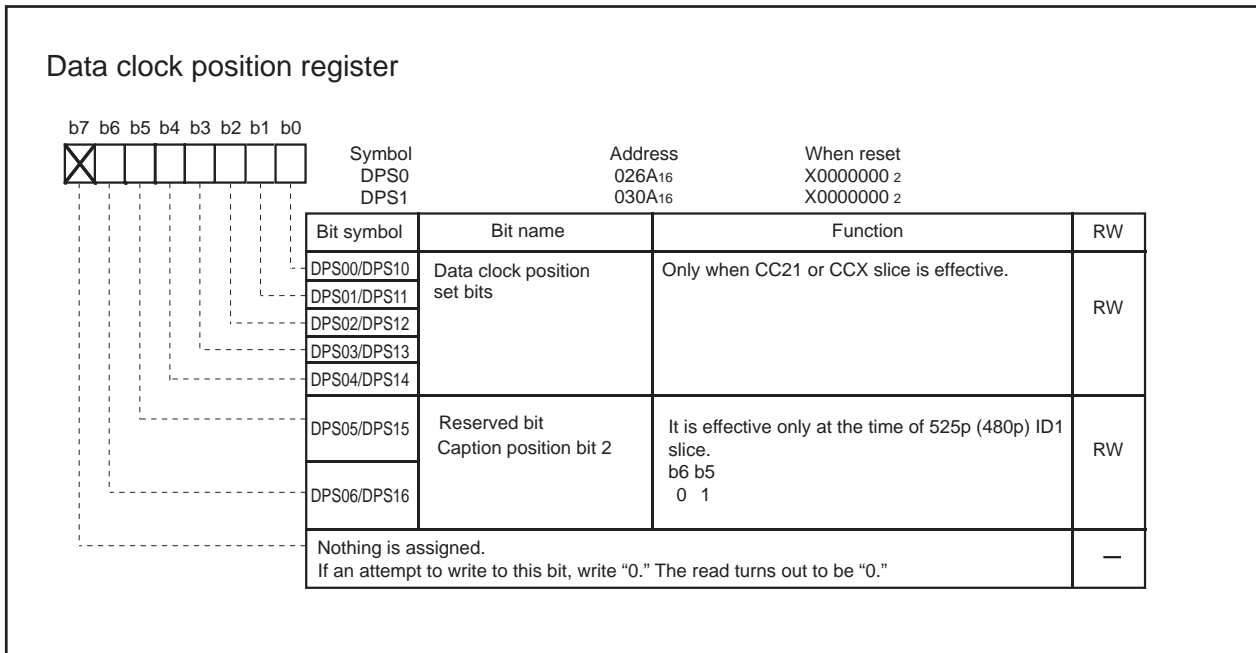


Figure 14.13 Data clock position register

Caption Register and CRCC Data Register

The caption data converted into a digital value by the comparator is stored into the caption register and CRCC data register in synchronization with the data clock. The contents of the stored caption data can be obtained by reading out the stored caption register and CRCC data register. These registers are reset to "0" at a falling edge of V_{sep}. Read out these registers after the occurrence of a data slicer interrupt.

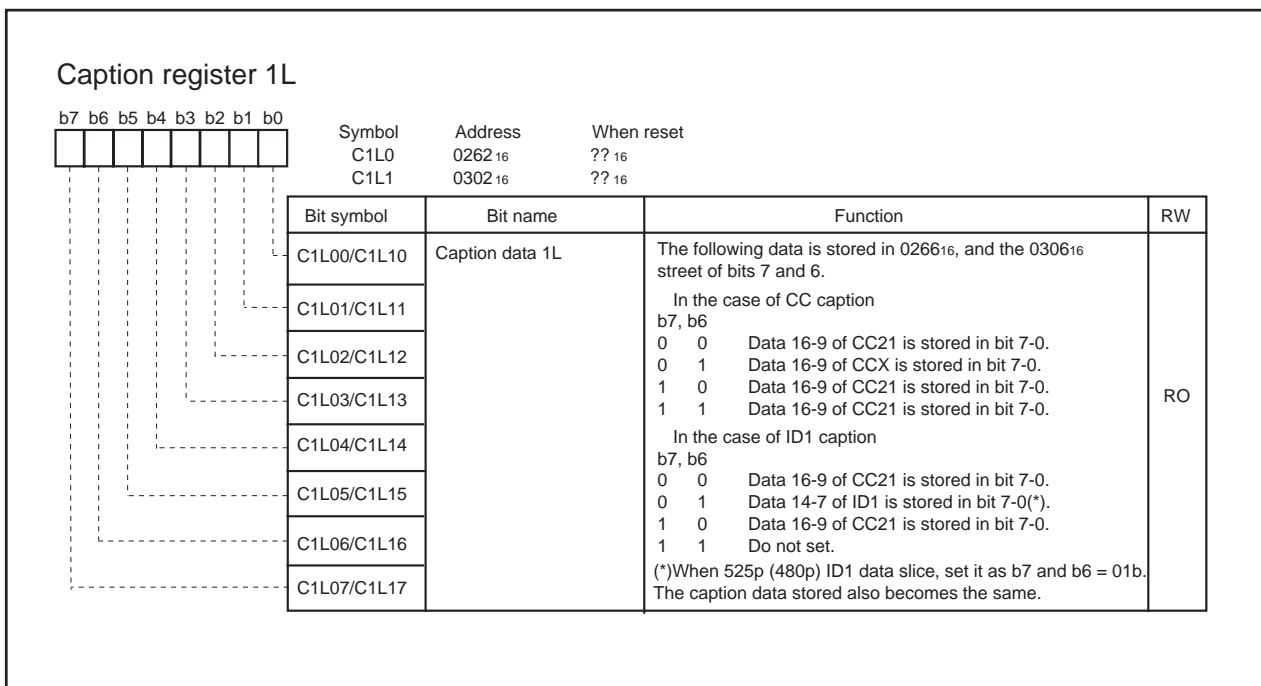


Figure 14.14 Caption register 1L

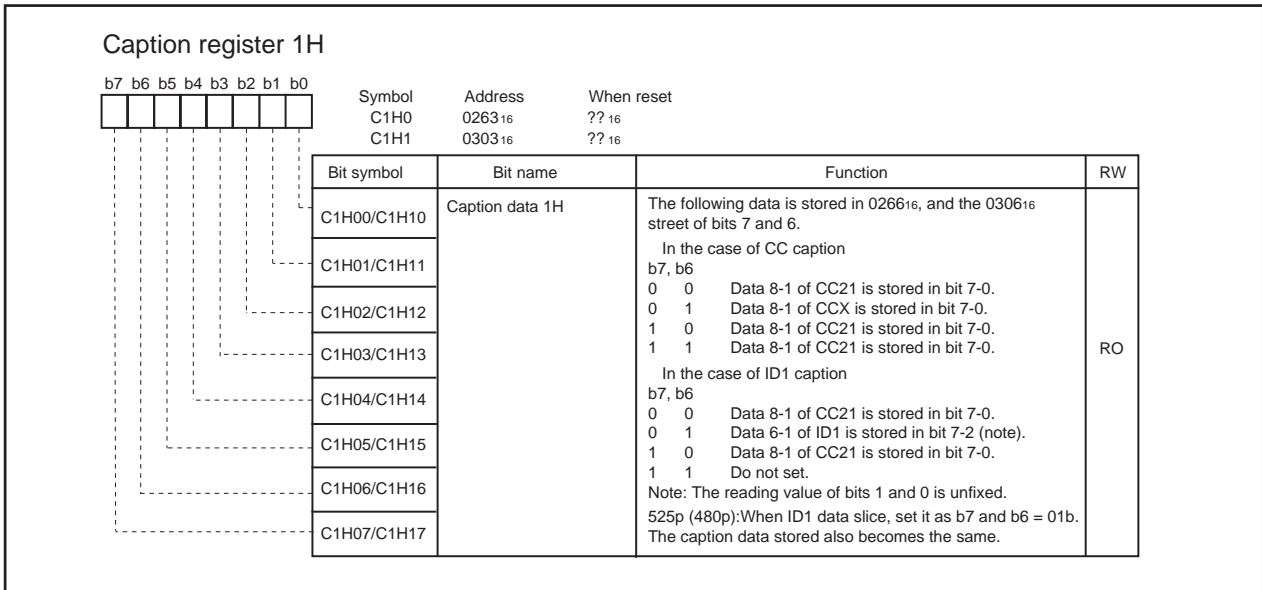


Figure 14.15 Caption register 1H

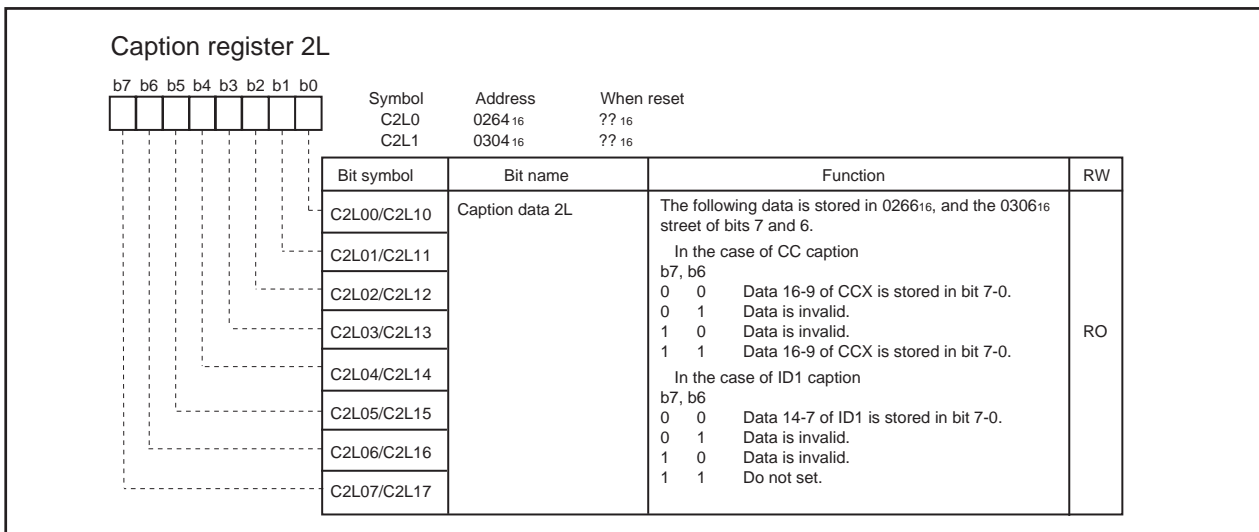


Figure 14.16 Caption register 2L

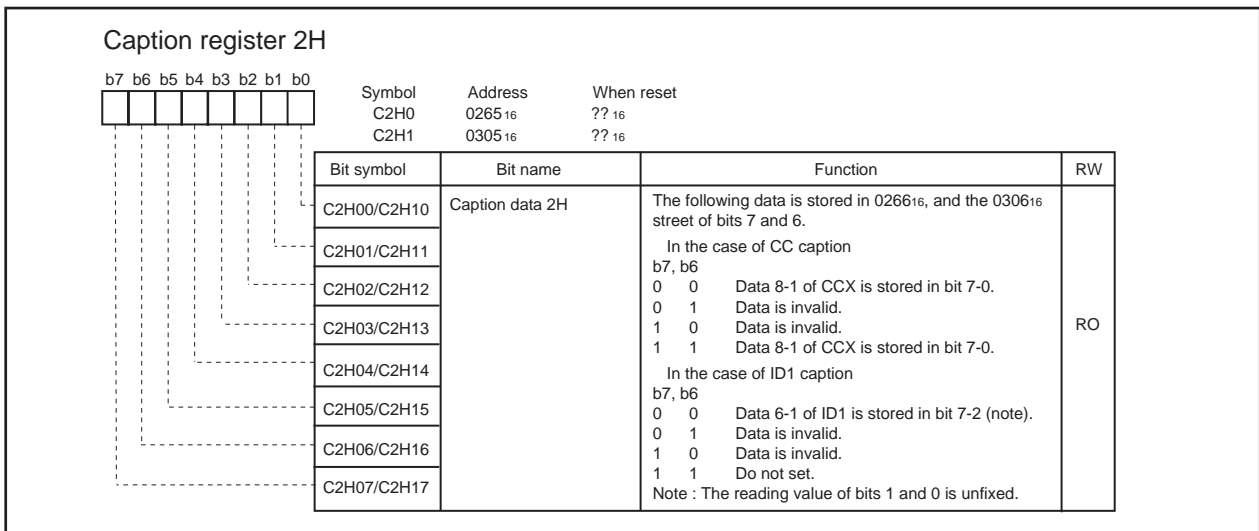


Figure 14.17 Caption register 2H

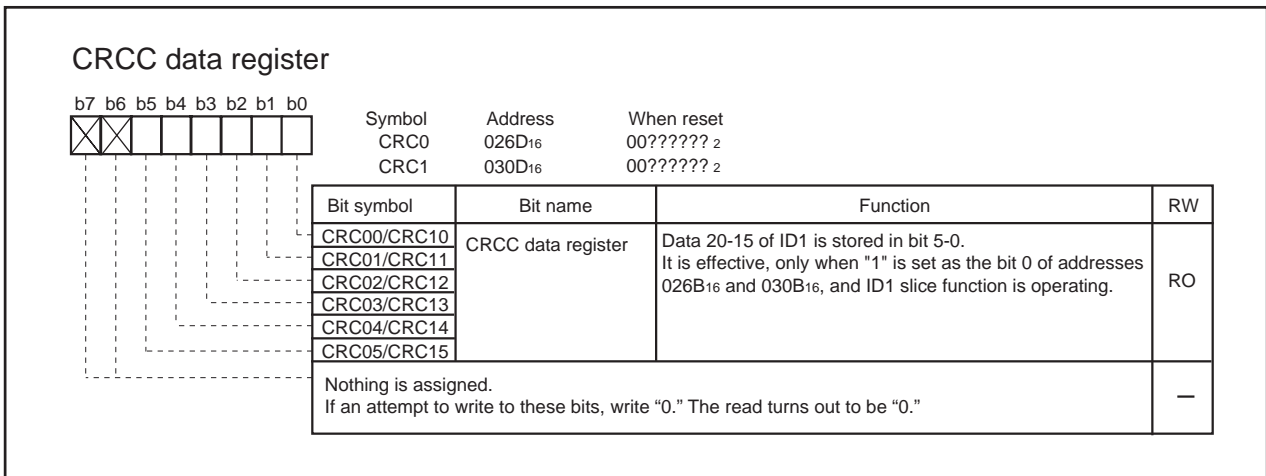


Figure 14.18 CRCC data register

Interrupt Request Generating Circuit

The interrupt requests as shown in Table 14.2 are generated by combination of the following bits; bits 6 and 7 of the caption position register (addresses 0266₁₆/0306₁₆). Read out the contents of caption data registers 1 and 2, CRCC data register, clock run-in detection register and standard clock detect register after the occurrence of a data slicer interrupt request.

Table 14.2 Occurrence sources of Interrupt request

CPS		Occurrence Sources of Interrupt Request at End of Data Slice Line
b7	b6	
0	0	After slicing line 21
	1	After a line specified by bits 4 to 0 of CPS (Note)
1	0	After slicing line 21
	1	After slicing line 21

CPS: Caption position register

Note: When 525p (480p), it becomes the one-line back specified caption position register bits 4 to 0 and the data clock position register bits 6 and 5.

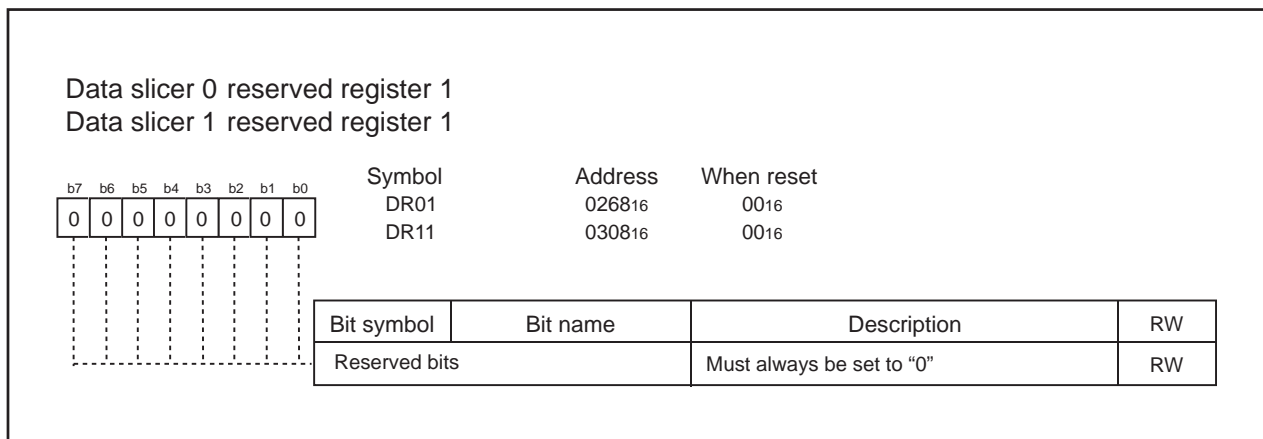


Figure 14.19 Data slicer i reserved register 1 (i = 1, 2)

ID1 data slice

When data slice ID1, ID1 control register of Fig 14.20 needs to be set.

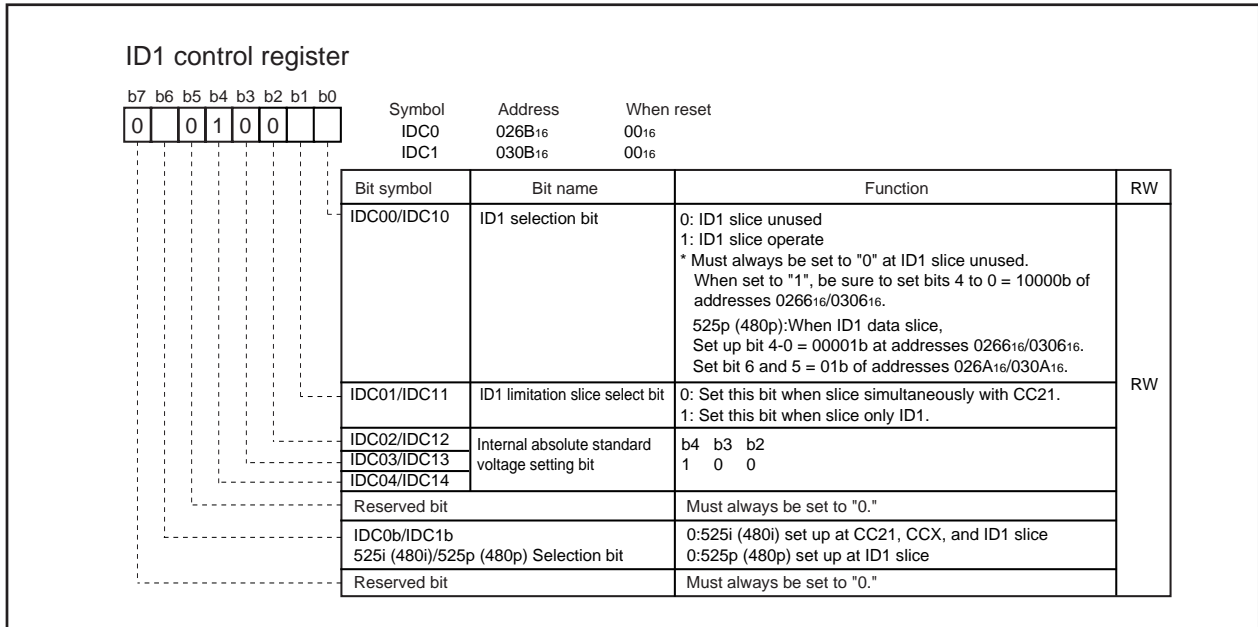


Figure 14.20 ID1 control register

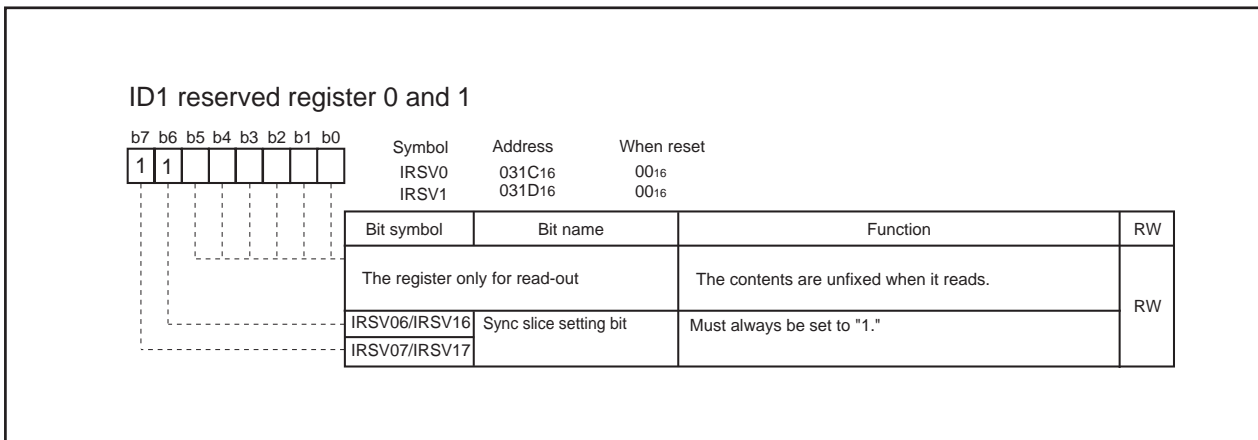


Figure 14.21 ID1 reserved register

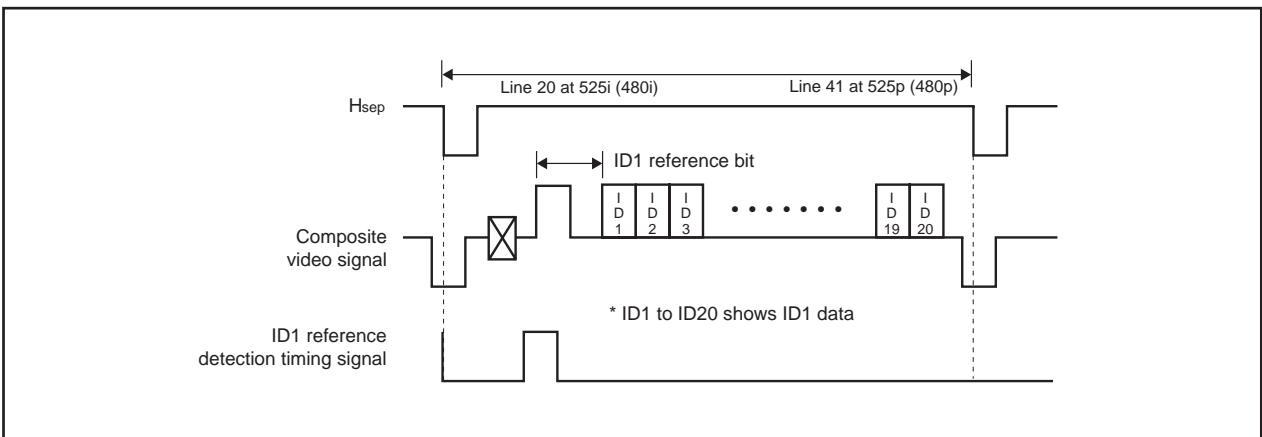


Figure 14.22 ID1 signal in vertical blanking interval

HSYNC Counter

The synchronous signal counter counts HSYNC from HSYNC count input pins (HC0 and HC1) as a count source.

The count value in a certain time (T time; 1024 μ s, 2048 μ s, 4096 μ s and 8192 μ s) divided system clock is stored into the 8-bit latch. Accordingly, the latch value changes in the cycle of T time. When the count value exceeds "FF16," "FF16" is stored into the latch.

The latch value can be obtained by reading out the HSYNC counter latch (address 027F16). A count source and count update cycle (T time) are selected by bits 0, 3 and 4 of the HSYNC counter register.

Figure 15.1 shows the HSYNC counter and Figure 15.2 shows the synchronous signal counter block diagram.

Note: HSYNC counter latch is a register only for read-out.

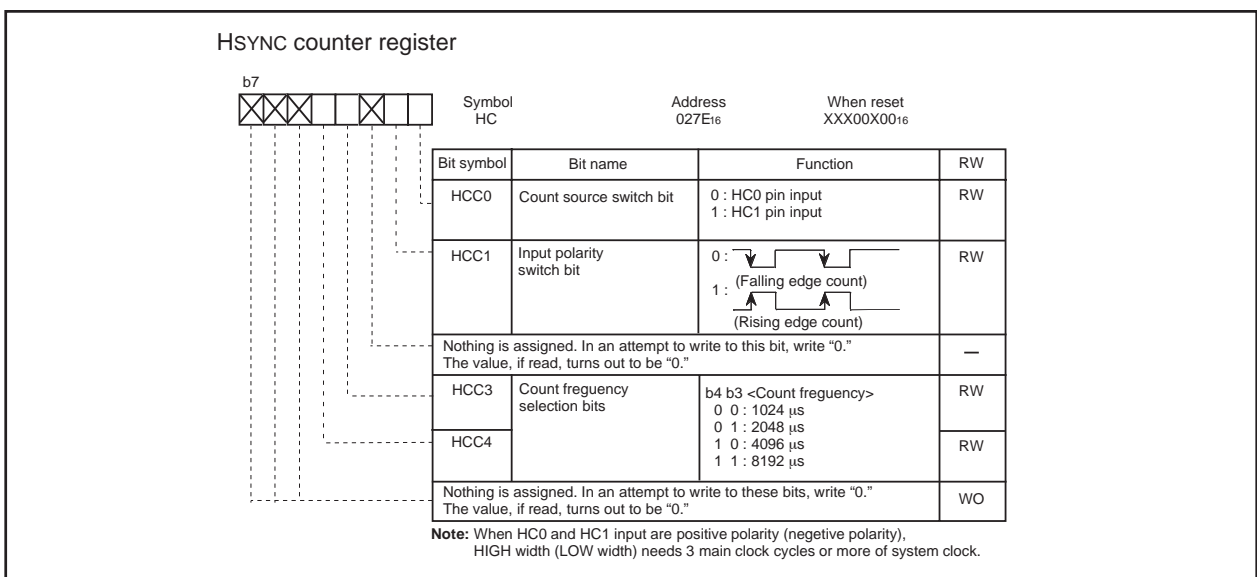


Figure 15.1 HSYNC counter register

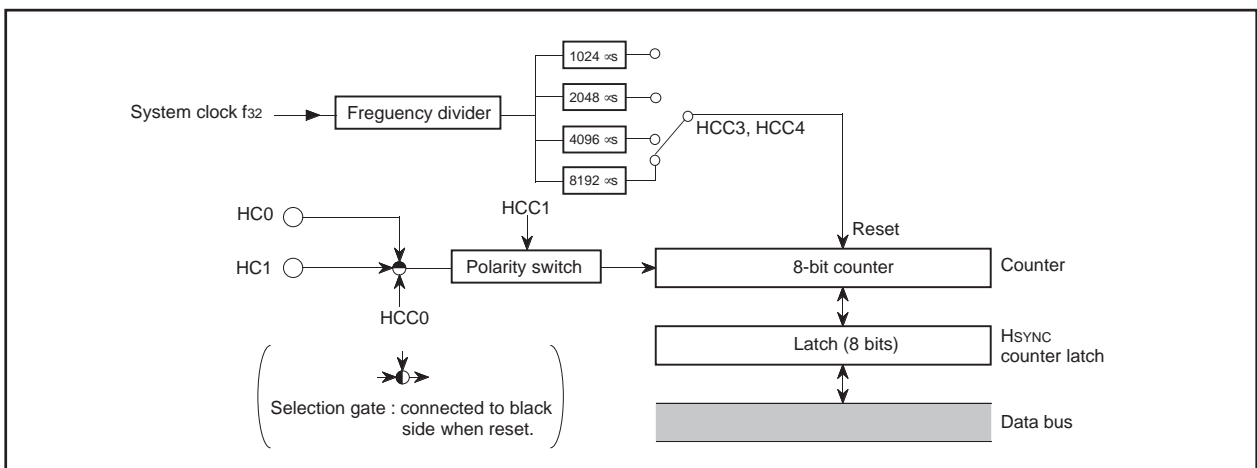


Figure 15.2 HSYNC counter block diagram

OSD Functions

Table 16.1 outlines the OSD functions of this microcomputer. This OSD function can display the following: the block display (32 characters X 16 lines or 42 characters X 16 lines) and the SPRITE display, and can display the both display at the same time. There are 3 display modes and they are selected by a block unit. The display modes are selected by block control register i ($i = 1$ to 16). The features of each display are described below.

Note: When using OSD function, select “No-division mode” as BCLK operating mode and set the main clock frequency to $f(XIN) = 16$ MHz.

Table 16.1 Features of each display style

Display style Parameter		Block display				SPRITE display	
		CC mode (Closed caption mode)	OSD mode (On-screen display mode)				CDOSD mode (Color dot on-screen display mode)
			OSDS mode	OSDP mode	OSDL mode		
Number of display characters		32 characters X 16 lines/42 characters X 16 lines				1 character X 2 lines	
Dot structure		16 X 20 dots (Character display area: 16 X 26 dots)	16 X 20 dots 12 X 20 dots 8 X 20 dots 4 X 20 dots	24 X 32 dots	16 X 26 dots	32 X 20 dots	
Kinds of character ROM	OSDL enable mode	254 kinds		254 kinds	126 kinds	2 kinds of RAM font	
	OSDL disable mode	508 kinds	254 kinds				
Kinds of character sizes (See note 1)		4 kinds	14 kinds	12 kinds	14 kinds	8 kinds	
Pre-divide ratio (Note)		X 1, X 2	X 1, X 2, X 3			X 1, X 2	
Dot size		1Tc X 1/2H, 1Tc X 1H	1Tc X 1/2H, 1Tc X 1H, 1.5Tc X 1/2H, 1.5Tc X 1H, 2Tc X 2H, 3Tc X 3H	1Tc X 1/2H, 1Tc X 1H, 2Tc X 2H, 3Tc X 3H	1Tc X 1/2H, 1Tc X 1H, 1.5Tc X 1/2H, 1.5Tc X 1H, 2Tc X 2H, 3Tc X 3H	1Tc X 1/2H, 1Tc X 1H, 2Tc X 1H, 2Tc X 2H	
Attribute		Smooth italic, under line, flash	Border				
Character font coloring		1 screen: 8 kinds (a character unit) Max. 512 kinds	1 screen: 16 kinds (a character unit) Max. 512 kinds		1 screen: 16 kinds (a dot unit) (only specified dots are colored by a character unit) Max. 512 kinds	1 screen: 16 kinds (a dot unit) Max. 512 kinds	
Character background coloring		Possible (a character unit, 1 screen: 4 kinds, Max. 512 kinds)	Possible (a character unit, 1 screen: 16 kinds, Max. 512 kinds)				
Display layer		Layer 1	Layers 1, 2	Layer 1	Layers 1, 2	Layer 3 (with highest priority)	
OSD output (See note 2)		Analog R, G, B output (each 8 adjustment levels: 512 colors), Digital OUT1, OUT2 output					
Raster coloring		Possible (a screen unit, max 512 kinds)					
Other function (See note 3)		Auto solid space function	Triple layer OSD function, Window function, Blank function				
Display expansion (multiline display)		Possible					

Notes 1: The character size is specified with dot size and pre-divide ratio (refer to “Dot Size”).

2: As for SPRITE display, OUT2 is not output.

3: As for SPRITE display, the window function does not operate.

The OSD circuit has an extended display mode. This mode allows multiple lines (16 lines or more) to be displayed on the screen by interrupting the display each time one line is displayed and rewriting data in the block for which display is terminated by software.

Figure 16.1 shows the display-enable fonts for each display style. Figure 16.2 shows the block diagram of the OSD circuit. Figure 16.3 shows the OSD control register 1. Figure 16.4 shows the block control register i.

Display Styles	Display-enable Fonts
CC Mode	
OSDS Mode	
OSDP Mode	<p>* : Character code fixation ** : Blank font</p>
OSDL Mode	
CDOSD Mode	
SPRITE	

Figure 16.1 Display-enable fonts for each display style

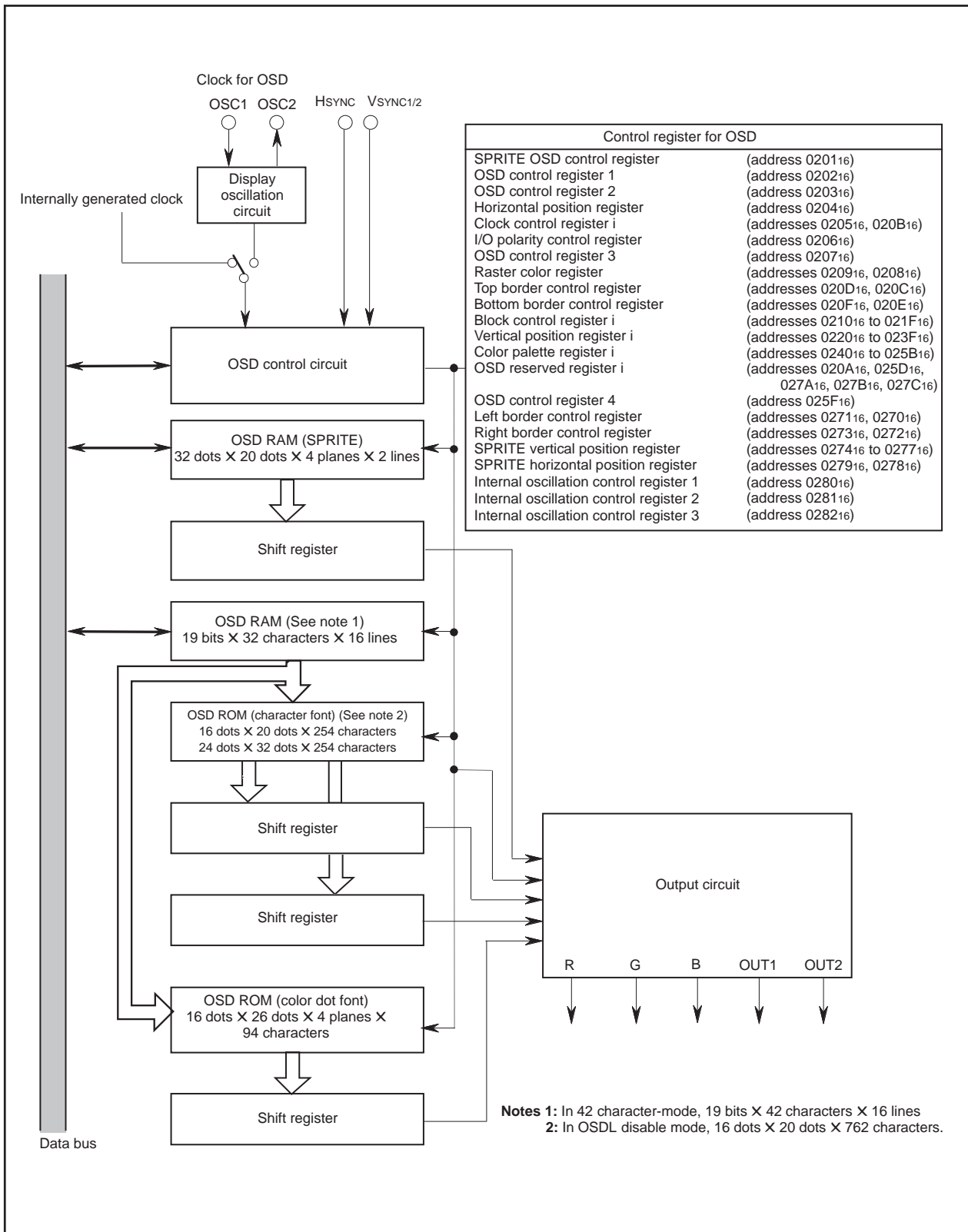


Figure 16.2 Block diagram of OSD circuit

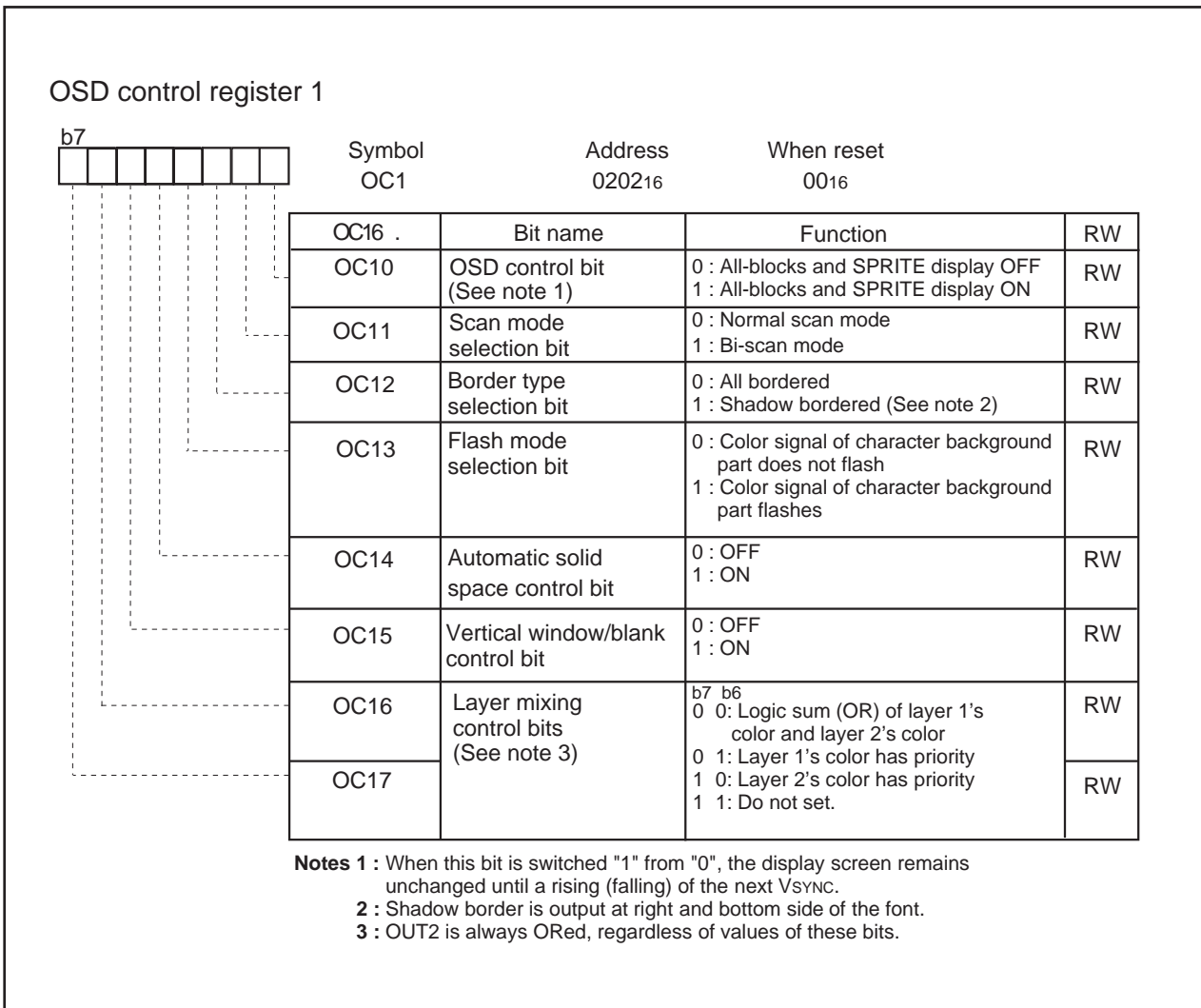


Figure 16.3 OSD control register 1

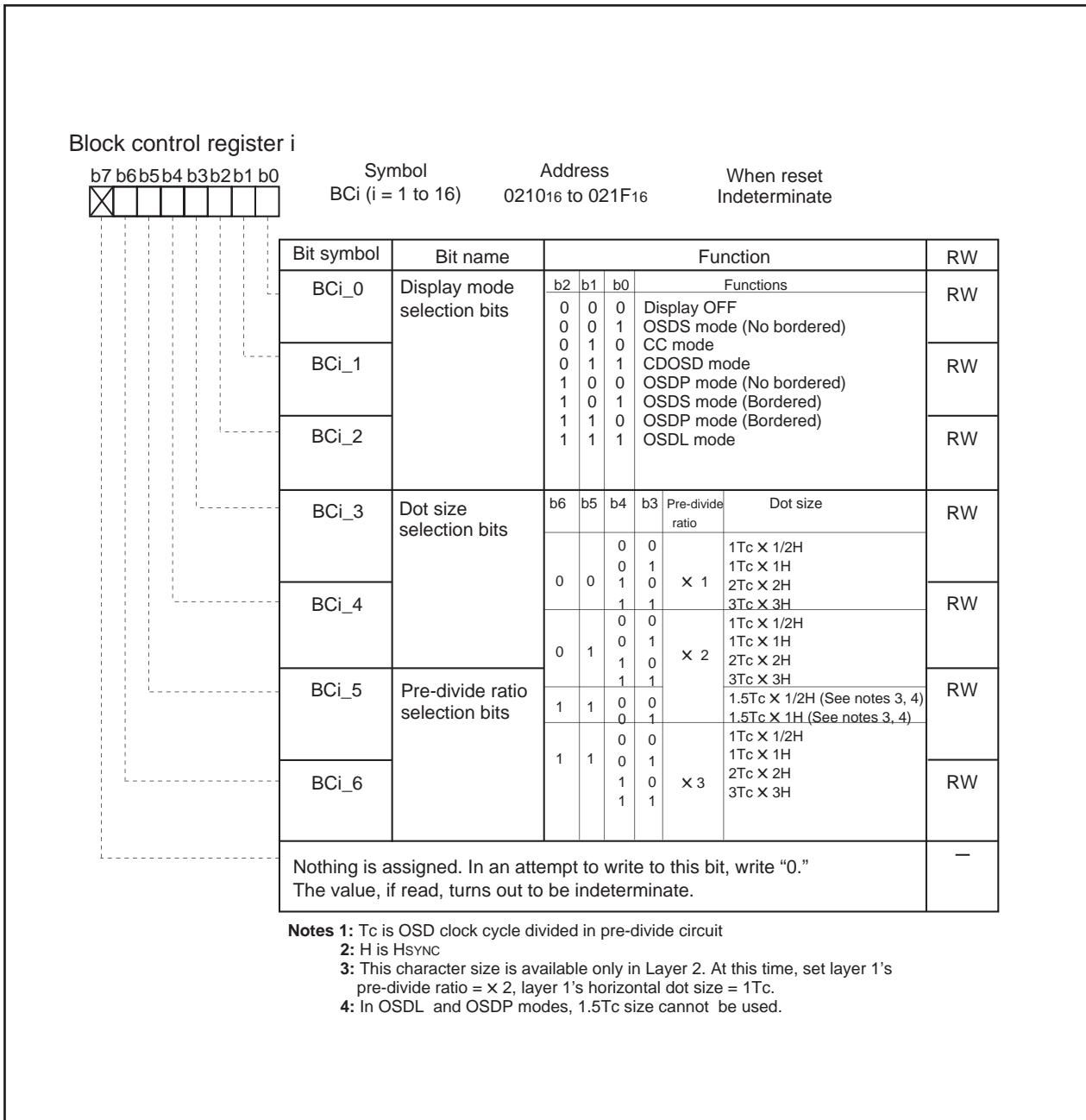


Figure 16.4 Block control register i (i = 1 to 16)

Triple Layer OSD

Three built-in layers of display screens accommodate triple display of channels, volume, etc., closed caption, and sprite displays within layers 1 to 3.

The layer to be displayed in each block is selected by bit 0 or 1 of the OSD control register 2 for each display mode (refer to Figure 16.7). Layer 3 always displays the sprite display.

When the layer 1 block and the layer 2 block overlay, the screen is composed with layer mixing by bit 6 or 7 of the OSD control register 1, as shown in Figure 16.5. Layer 3 always takes display priority of layers 1 and 2.

Notes 1: When mixing layer 1 and layer 2, note Table 16.2.

2: OSDP mode is always displayed on layer 1. And also, it cannot be overlapped with layer 2's block.

3: OUT2 is always ORed, regardless of values of bits 6, 7 of the OSD control register 1. And besides, even when OUT2 (layer 1 and layer 2) overlaps with SPRITE display (layer 3), OUT2 is output without masking.

Table 16.2 Mixing layer 1 and layer 2

Parameter \ Block	Block in Layer 1	Block in Layer 2	
Display mode	CC, OSDS/L, CDOSD mode	OSDS/L, CDOSD mode	
Pre-divide ratio	X 1, X 2 (CC mode) X 1 to X 3 (OSD, CDOSD mode)	Same as layer 1 (See note)	
Dot size	1Tc X 1/2H, 1Tc X 1H (CC mode)	Pre-divide ratio = X 1	Pre-divide ratio = X 2
		1Tc X 1/2H 1Tc X 1H	1TcX1/2H, 1.5TcX1/2H 1TcX1H, 1.5TcX1H(Seenote)
	1Tc X 1H, 1Tc X 1/2H, 2Tc X 2H, 3Tc X 3H (OSDS/L, CDOSD mode)	<ul style="list-style-type: none"> • Same size as layer 1 • 1.5Tc can be selected only when: layer 1's pre-divide ratio = X 2 AND layer 1's horizontal dot size = 1Tc. As this time, vertical dot size is the same as layer 1. 	
Horizontal display start position	Arbitrary	Same position as layer 1	
Vertical display start position	Arbitrary However, when dot size is 2Tc X 2H or 3Tc X 3H, set difference between vertical display position of layer 1 and that of layer 2 as follows. <ul style="list-style-type: none"> •2Tc X 2H: 2H units •3Tc X 3H: 3H units 		

Note: In the OSDL mode, 1.5Tc size cannot be used.

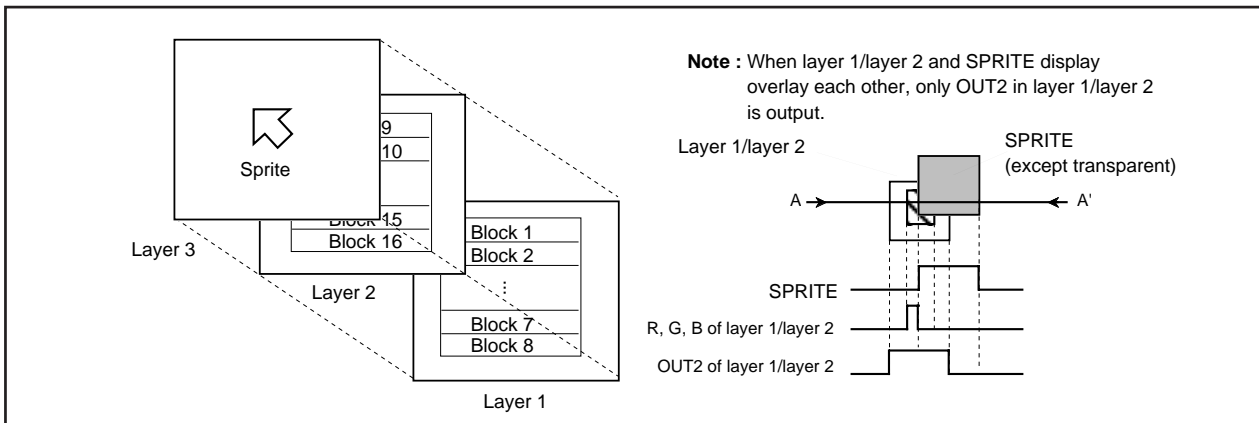


Figure 16.5 Triple layer OSD

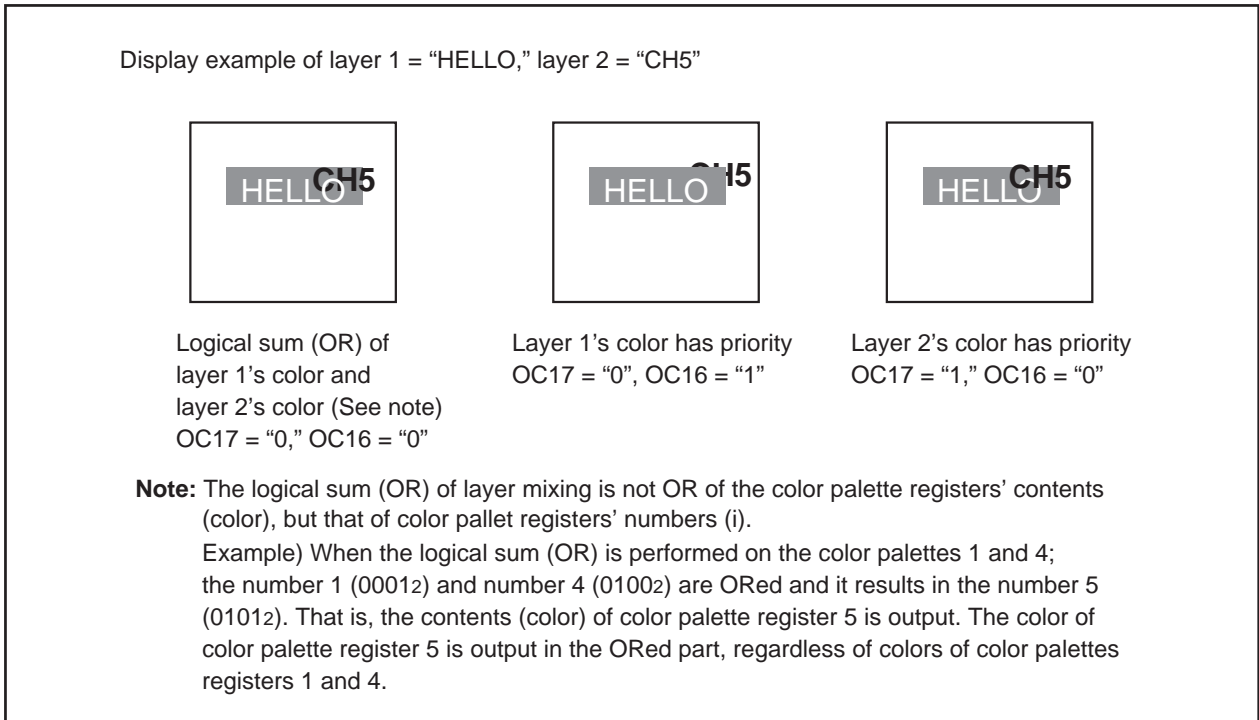


Figure 16.6 Display example of layer mixing OSD

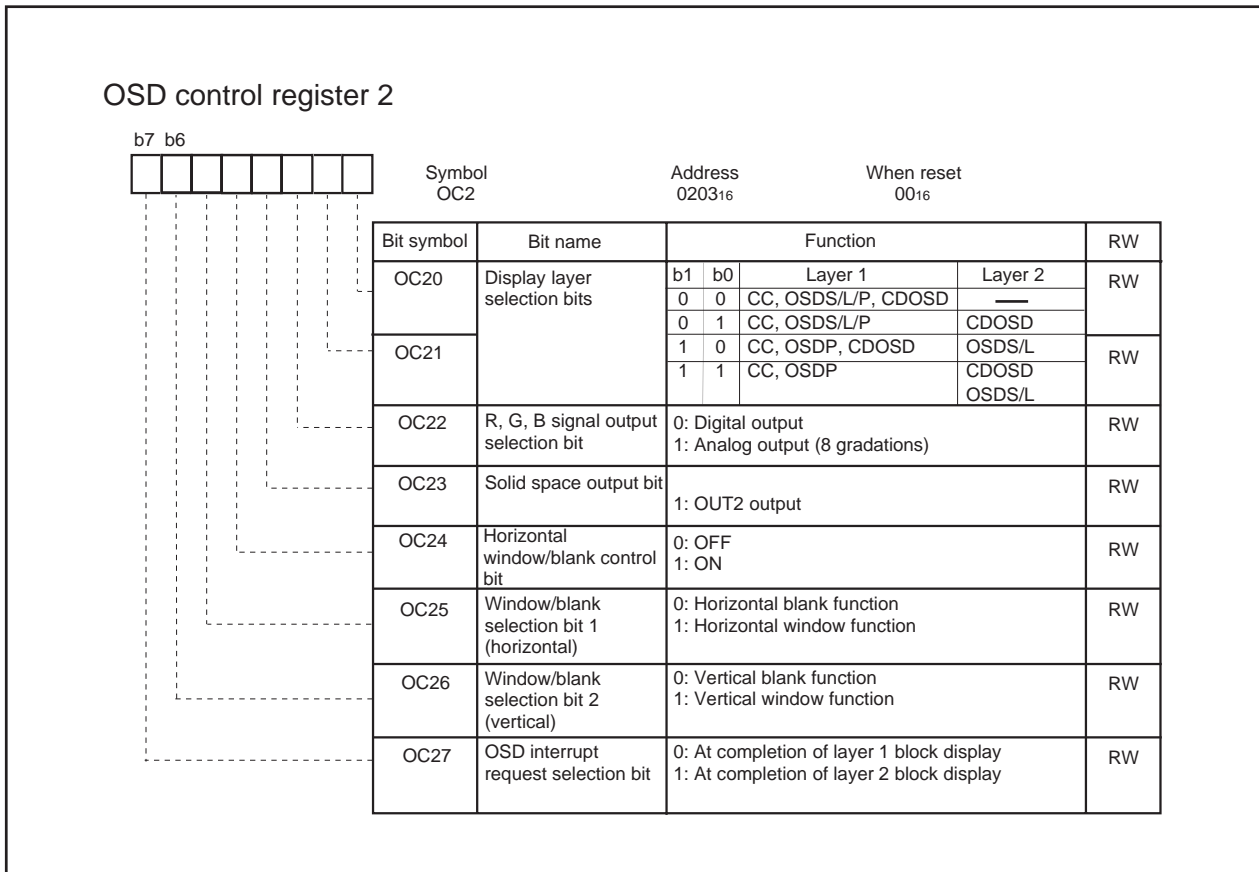


Figure 16.7 OSD control register 2

Display Position

The display positions of characters are specified by a block. There are 16 blocks, blocks 1 to 16. Up to 32 characters (32-character mode)/42 characters (42-character mode)/ can be displayed in each block (refer to Memory for OSD).

The display position of each block can be set in both horizontal and vertical directions by software.

The display position in the horizontal direction can be selected for all blocks in common from 256-step display positions in units of 4 TOSC (TOSC = OSD oscillation cycle).

The display position in the vertical direction for each block can be selected from 1024-step display positions in units of 1 TH (TH = HSYNC cycle).

Blocks are displayed in conformance with the following rules:

- When the display position is overlapped with another block in the same layer (Figure 16.8 (b)), a low block number (1 to 16) is displayed on the front.
- When another block display position appears while one block is displayed in the same layer (Figure 16.8 (c)), the block with a larger set value as the vertical display start position is displayed. However, do not display block with the dot size of $2Tc \times 2H$ or $3Tc \times 3H$ during display period (*) of another block.
 - * In the case of OSDSP mode block: 20 dots in vertical from the vertical display start position.
 - * In the case of OSDL mode block: 32 dots in vertical from the vertical display start position.
 - * In the case of CC or CDOSD mode block: 26 dots in vertical from the vertical display start position.

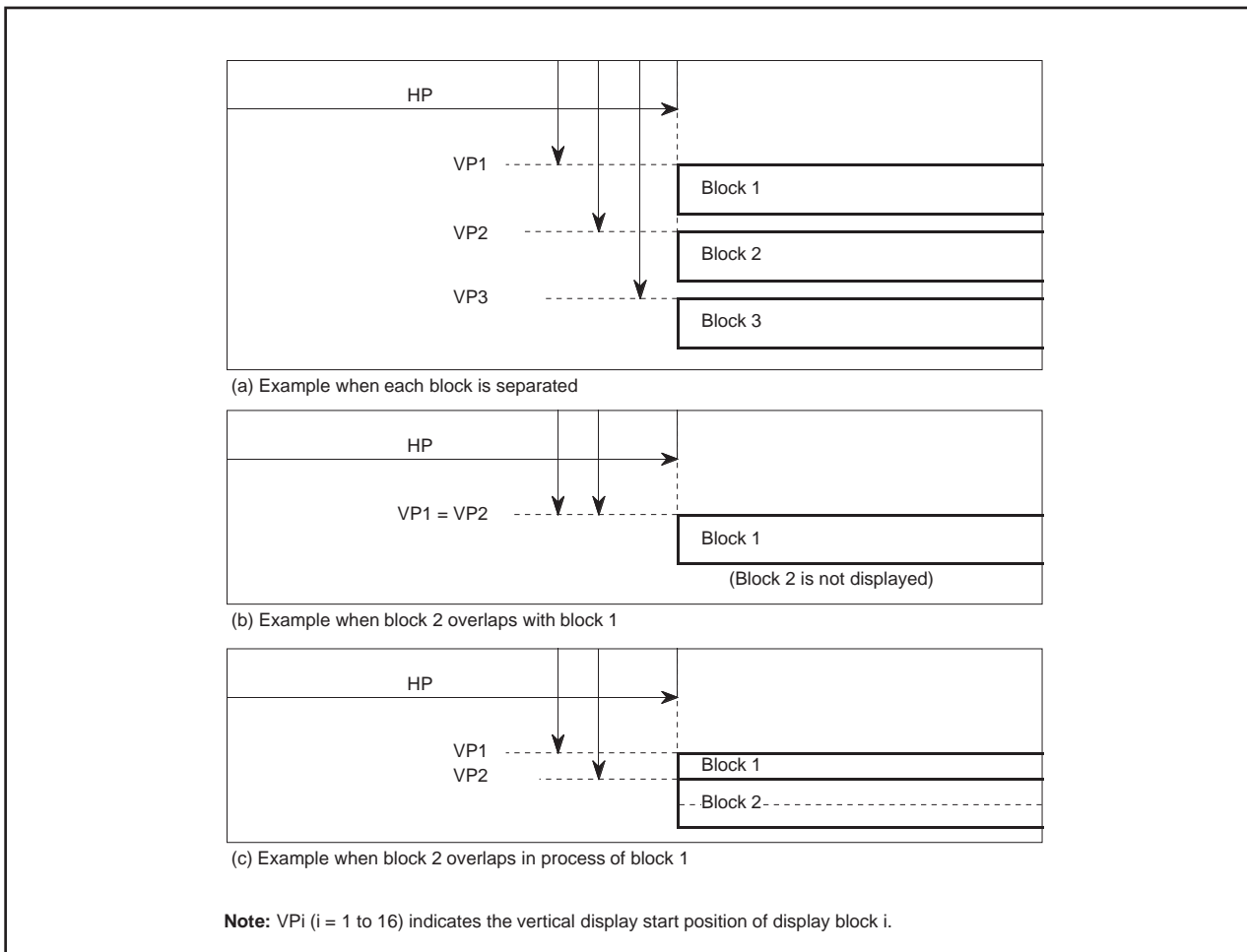


Figure 16.8 Display position

The display position in the vertical direction is determined by counting the horizontal sync signal (HSYNC). At this time, when VSYNC and HSYNC are positive polarity (negative polarity), it starts to count the rising edge (falling edge) of HSYNC signal from after fixed cycle of rising edge (falling edge) of VSYNC signal. So interval from rising edge (falling edge) of VSYNC signal to rising edge (falling edge) of HSYNC signal needs enough time ($2 \times \text{BCLK}$ cycles or more) for avoiding jitter. The polarity of HSYNC and VSYNC signals can select with the I/O polarity control register (address 0206₁₆).

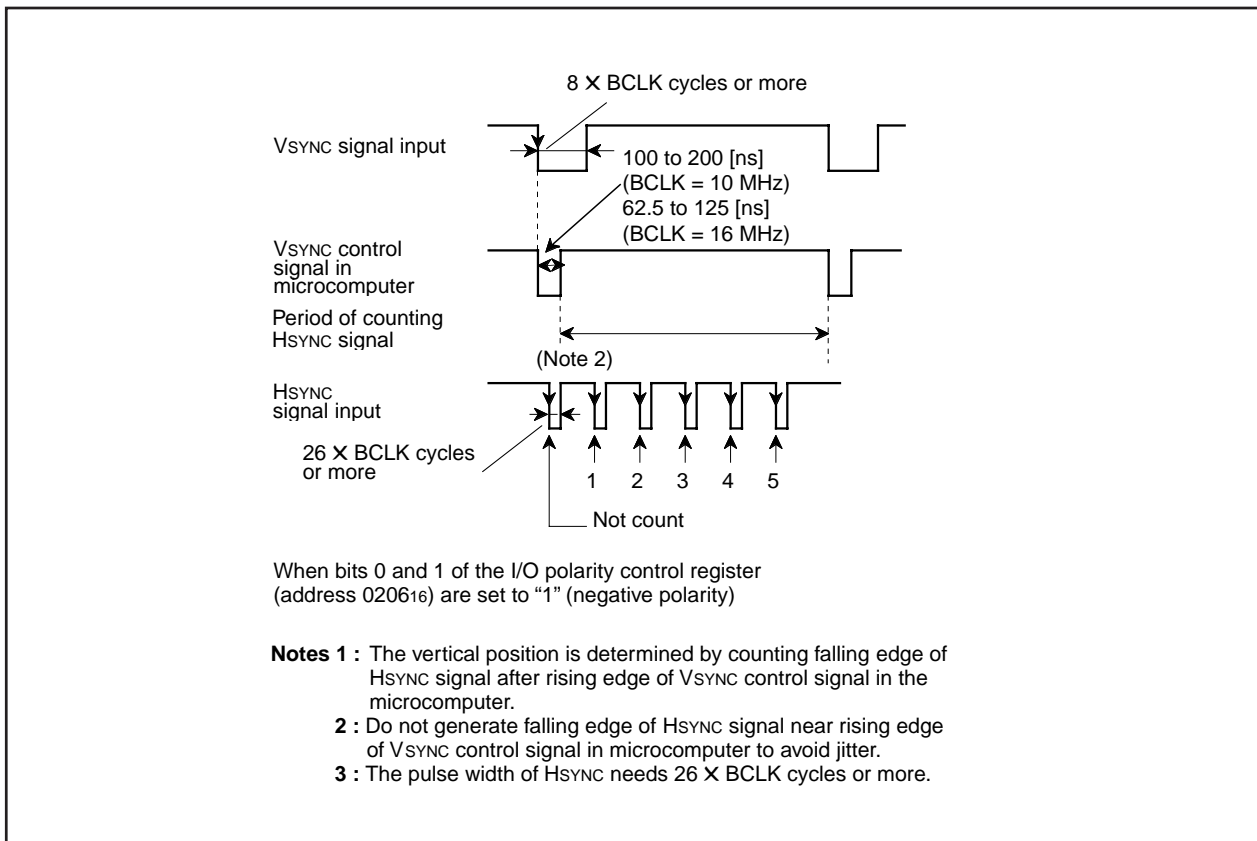


Figure 16.9 Supplement explanation for display position

The vertical position for each block can be set in 1024 steps (where each step is 1_{TH} (TH: HSYNC cycle)) as values “002₁₆” to “3FF₁₆” in vertical position register i (i = 1 to 16) (addresses 0220₁₆ to 023F₁₆). The vertical position register i is shown in Figure 16.10.

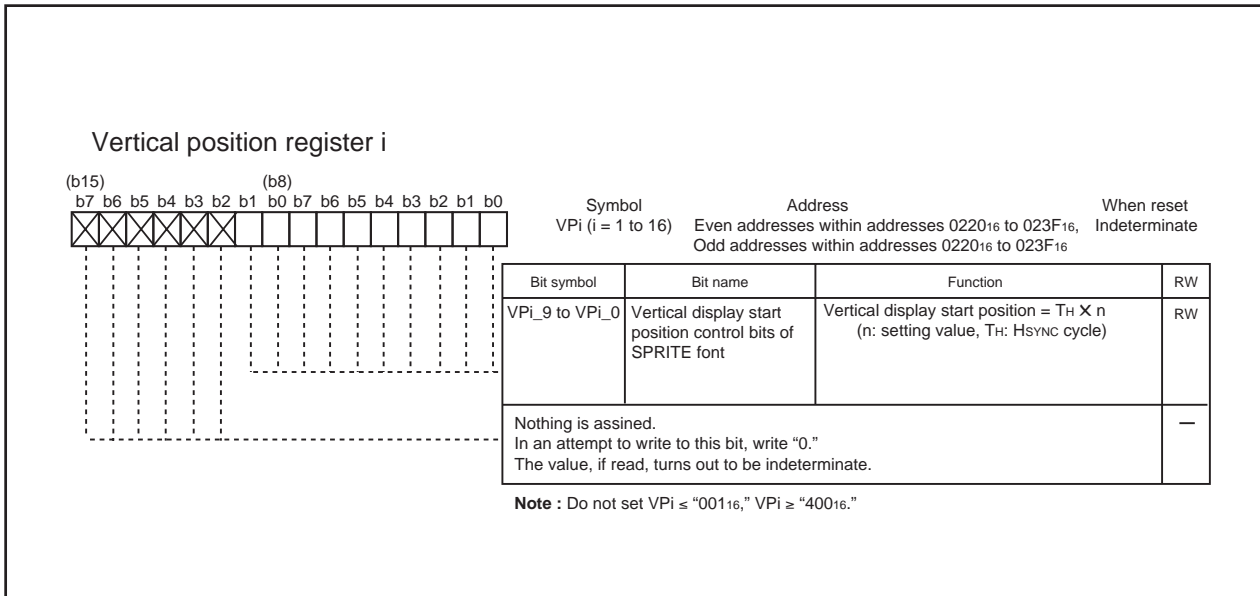


Figure 16.10 Vertical position register i (i = 1 to 16)

The horizontal position is common to all blocks, and can be set in 256 steps (where 1 step is 4_{Tosc}, _{Tosc} being OSD oscillation cycle) as values “00₁₆” to “FF₁₆” in bits 0 to 7 of the horizontal position register (address 0204₁₆). The horizontal position register is shown in Figure 16.11.

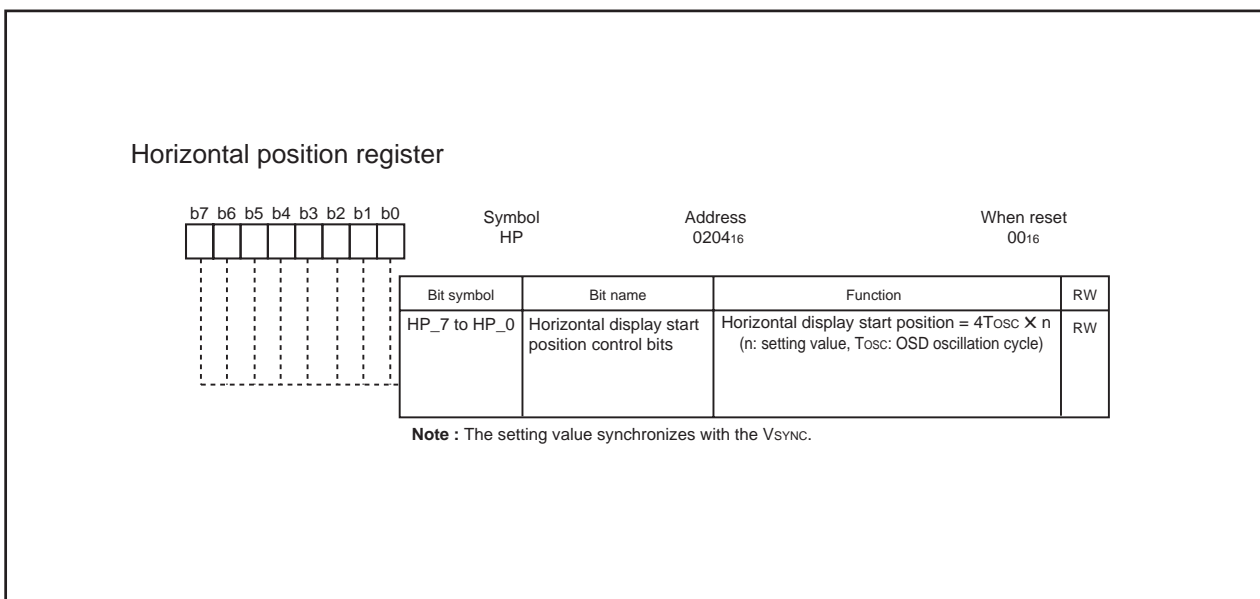


Figure 16.11 Horizontal position register

Note : $1T_c$ (T_c : OSD clock cycle divided in pre-divide circuit) gap occurs between the horizontal display start position set by the horizontal position register and the most left dot of the 1st block. Accordingly, when 2 blocks have different pre-divide ratios, their horizontal display start position will not match.

Ordinary, this gap is $1T_c$ regardless of character sizes, however, the gap is $1.5T_c$ only when the character size is $1.5T_c$.

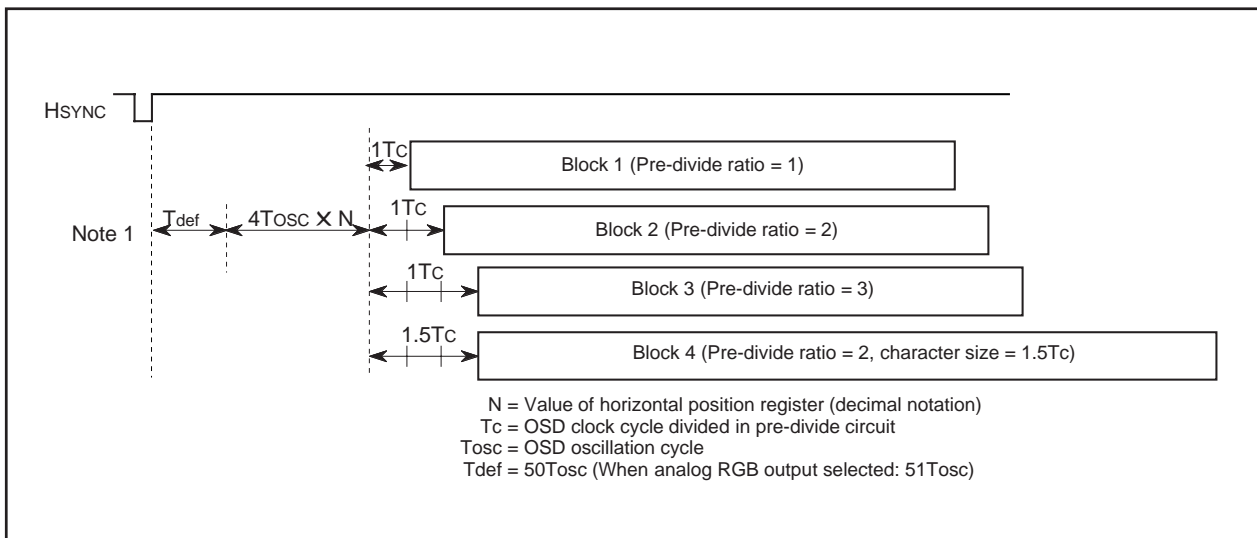


Figure 16.12 Notes on horizontal display start position

Dot Size

The dot size can be selected by a block unit. The dot size in vertical direction is determined by dividing HSYNC in the vertical dot size control circuit. The dot size in horizontal is determined by dividing the following clock in the horizontal dot size control circuit : the clock gained by dividing the OSD clock source (internally generated clock, OSC1, main clock) in the pre-divide circuit. The clock cycle divided in the pre-divide circuit is defined as 1Tc.

The dot size is specified by bits 3 to 6 of the block control register.

Refer to Figure 16.4 (the block control register i), refer to Figure 16.15 (the clock control register).

The block diagram of dot size control circuit is shown in Figure 16.13.

Notes 1 : The pre-divide ratio = 3 cannot be used in the CC mode.

2 : The pre-divide ratio of the layer 2 must be same as that of the layer 1 by the block control register i.

3 : In the bi-scan mode, the dot size in the vertical direction is 2 times as compared with the normal mode. Refer to “Scan Mode” about the scan mode.

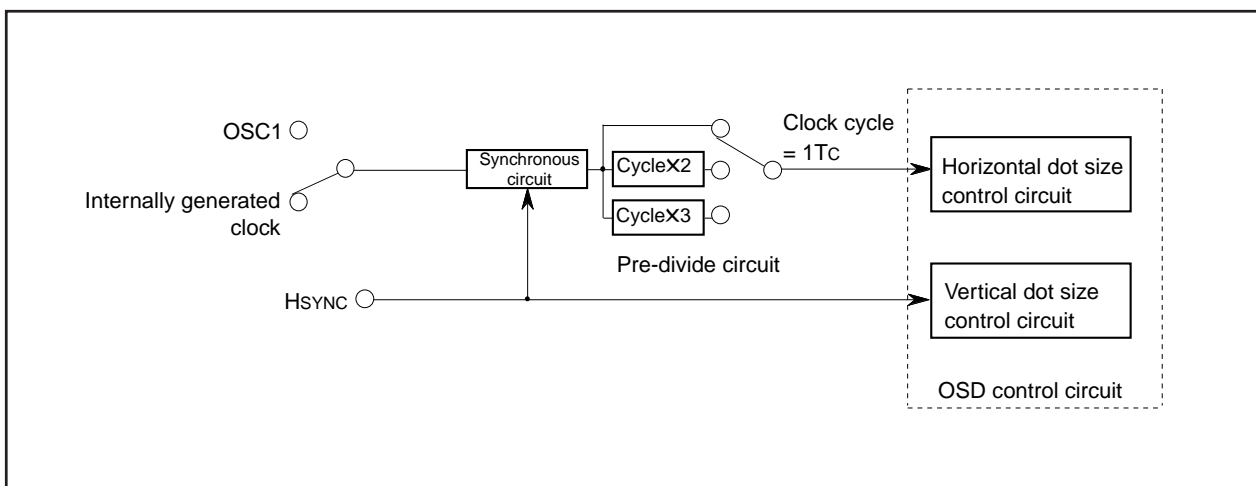


Figure 16.13 Block diagram of dot size control circuit

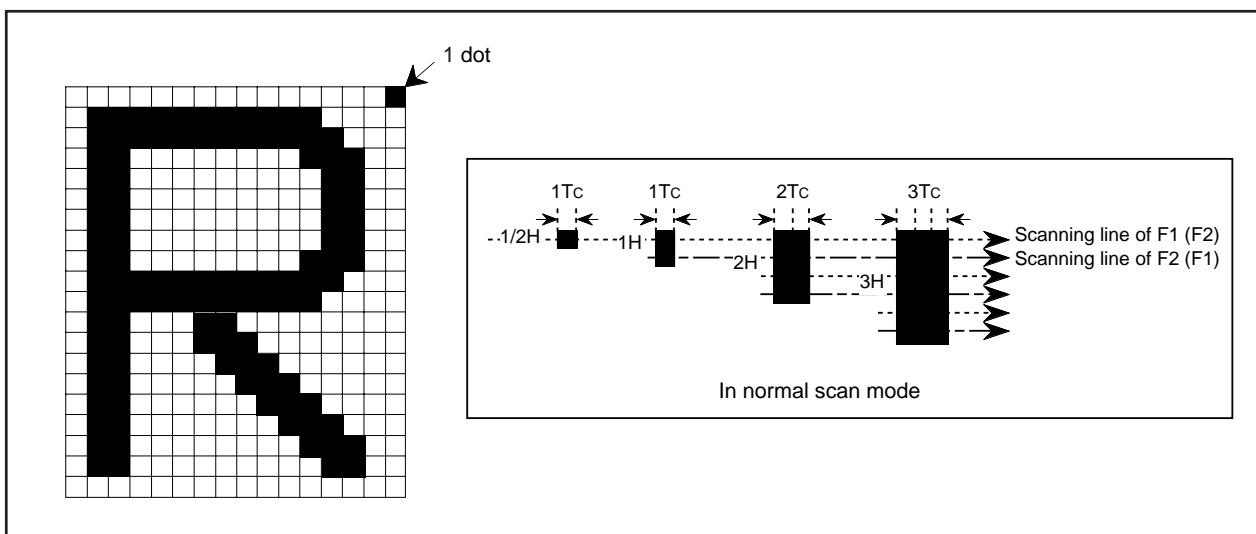


Figure 16.14 Definition of dot sizes

Clock for OSD

As a clock for display to be used for OSD, it is possible to select one of the following 3 types.

- Internally generated clock output by the internal oscillator
- Clock from the LC oscillator supplied from the pin OSC1
- Clock from the ceramic resonator (or the quartz-crystal oscillator) from the pin OSC1

When the clock control register i ($i=1-2$) is set to choose an internally generated clock for the OSD clock, use the internal oscillation control register i ($i=1-3$) to select the oscillation frequency.

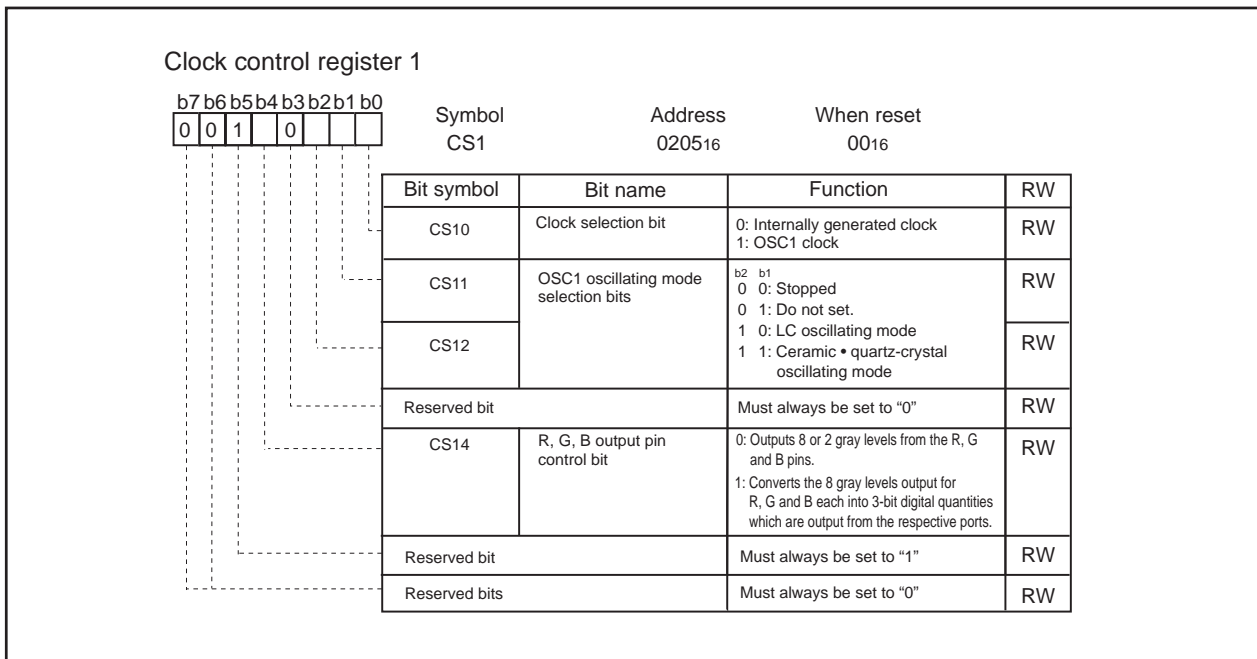


Figure 16.15 Clock control register 1

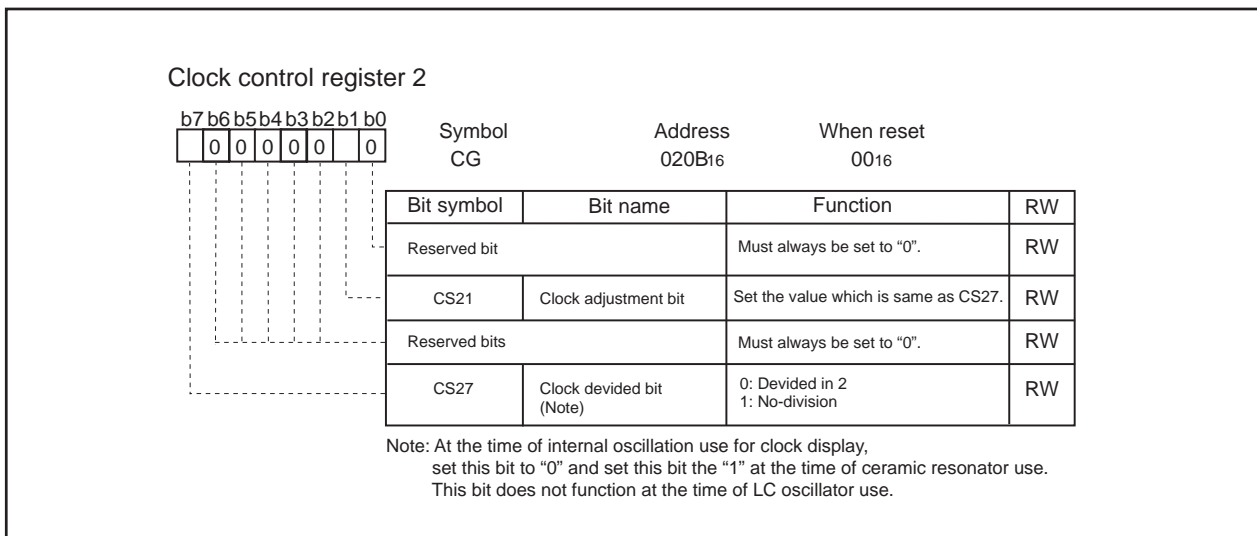


Figure 16.16 Clock control register 2

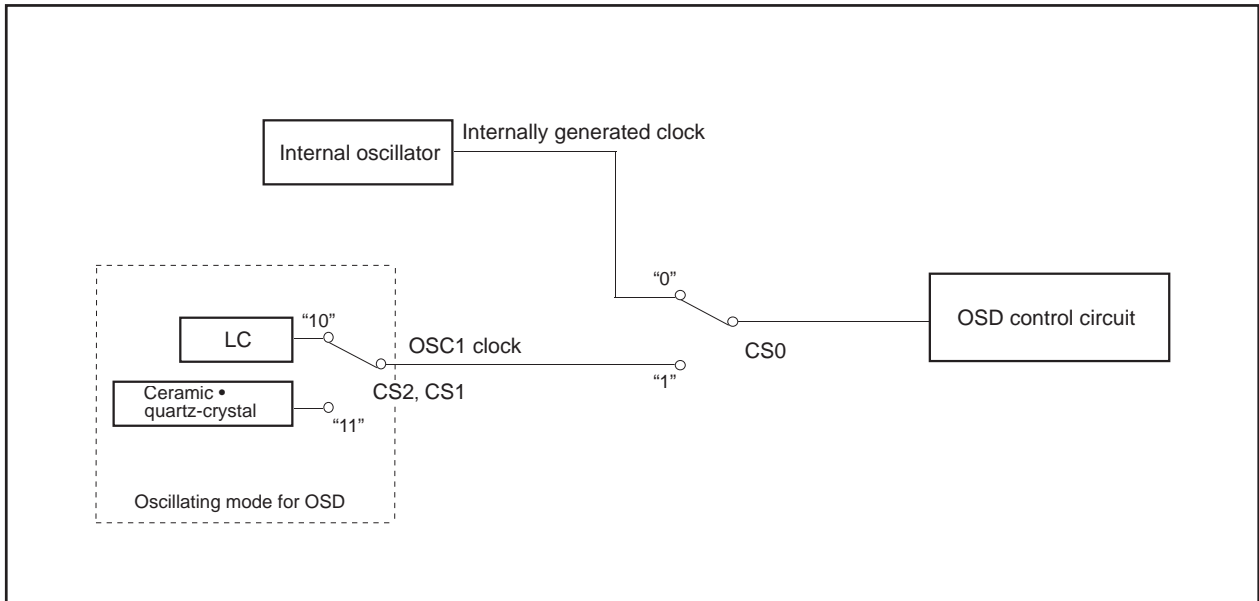


Figure 16.17 Block Diagram of OSD selection circuit

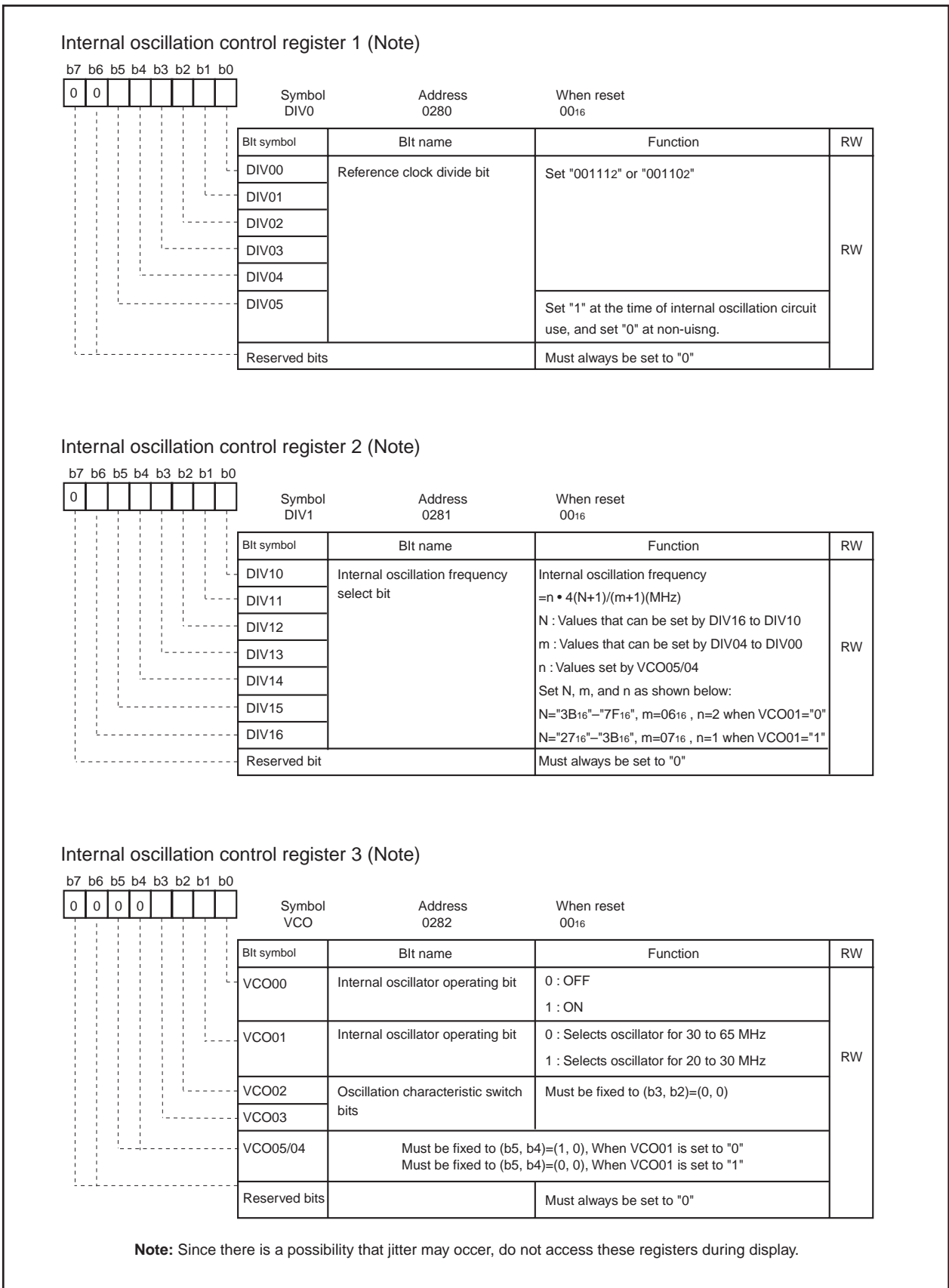


Figure 16.18 Internal oscillation control register i (i=1 to 3)

2.16.5 Field Determination Display

To display the block with vertical dot size of 1/2H, whether an even field or an odd field is determined through differences in a synchronizing signal waveform of interlacing system. The dot line 0 or 1 (refer to Figure 16.20) corresponding to the field is displayed alternately.

In the following, the field determination standard for the case where both the horizontal sync signal and the vertical sync signal are negative-polarity inputs will be explained. A field determination is determined by detecting the time from a falling edge of the horizontal sync signal until a falling edge of the VSYNC control signal (refer to Figure 16.9) in the microcomputer and then comparing this time with the time of the previous field. When the time is longer than the comparing time, it is regarded as even field. When the time is shorter, it is regarded as odd field.

The field determination flag changes at a rising edge of VSYNC control signal in the microcomputer .

The contents of this field can be read out by the field determination flag (bit 7 of the I/O polarity control register at address 0206₁₆). A dot line is specified by bit 6 of the I/O polarity control register (refer to Figure 16.19).

However, the field determination flag read out from the CPU is fixed to "0" at even field or "1" at odd field, regardless of bit 6.

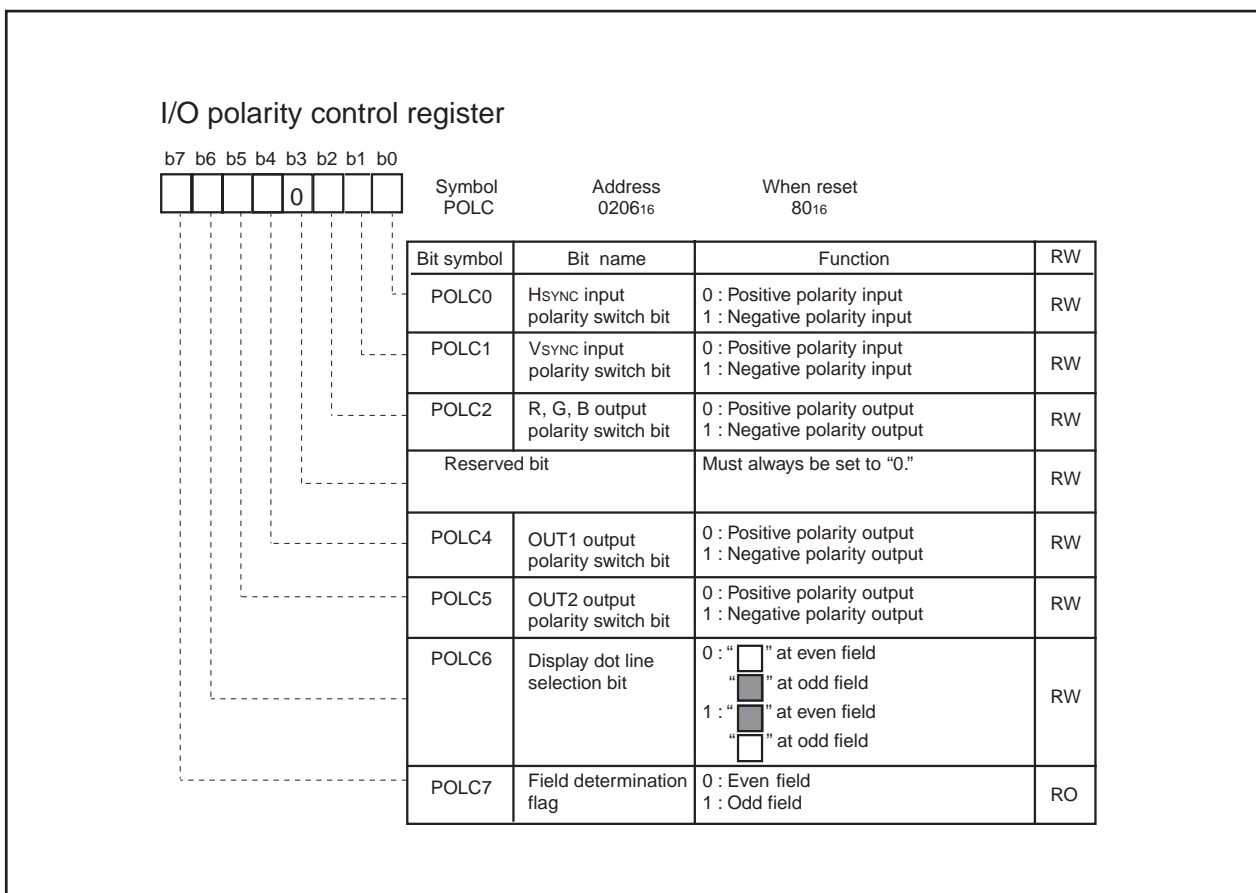
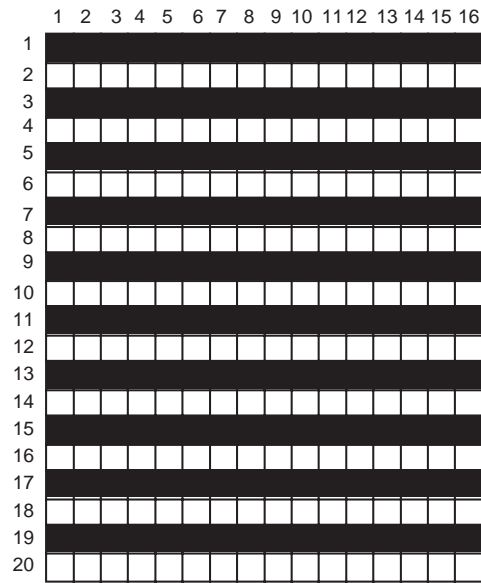
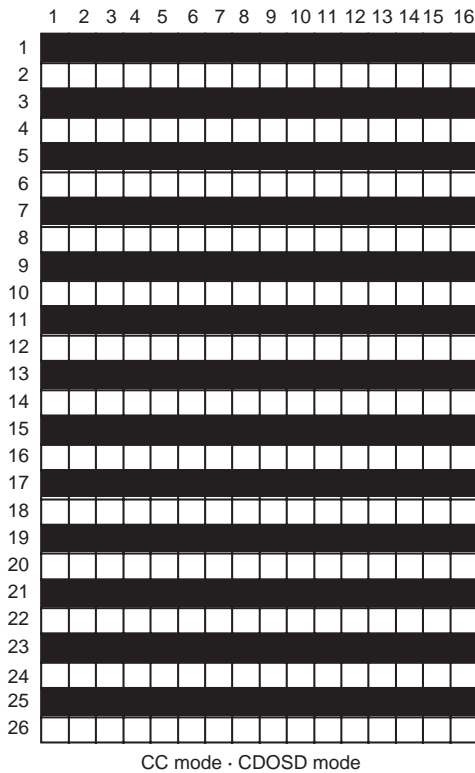


Figure 16.19 I/O polarity control register

Both Hsync signal and Vsync signal are negative-polarity input

Hsync		Field	Field determination flag(Note)	Display dot line selection bit	Display dot line
Vsync and Vsync control signal in micro-computer	(n - 1) field (Odd-numbered)	Odd	/	/	/
	Upper : Vsync signal (n) field (Even-numbered)	Even	0 (T2 > T1)	0	Dot line 1 <input type="checkbox"/>
	Lower : Vsync control signal in micro-computer (n + 1) field (Odd-numbered)	Odd	1 (T3 < T2)	1	Dot line 0 <input checked="" type="checkbox"/>



When the display dot line selection bit is "0," the "□" font is displayed at even field, the "■" font is displayed at odd field. Bit 7 of the I/O polarity control register can be read as the field determination flag : "1" is read at odd field, "0" is read at even field.

OSD ROM font configuration diagram

Note : The field determination flag changes at a rising edge of the Vsync control signal (negative-polarity input) in the microcomputer.

Figure 16.20 Relation between field determination flag and display font

Memory for OSD

There are 2 types of memory for OSD : OSD ROM (addresses 30000_{16} to $4FFFF_{16}$) used to store character dot data and OSD RAM (addresses 8000_{16} to $8FFF_{16}$) used to specify the kinds of display characters, display colors, and SPRITE display. The following describes each type of memory.

(1) ROM for OSD (addresses 30000_{16} to $4FFFF_{16}$)

The dot pattern data for OSD characters is stored in the character font area in the OSD ROM and the CD font data for OSD characters is stored in the color dot font area in the OSD ROM. To specify the kinds of the character font and the CD font, it is necessary to write the character code into the OSD RAM.

For character font, there are the following 2 mode.

- OSDL enable mode
16 X 20-dot font and 24 X 32-dot font
- OSDL disable mode
16 X 20-dot font

The modes are selected by bit 0 of the OSD control register 4 for each screen.

The conditions for each OSDL enable/disable mode are shown in Figure 16.22.

During OSDL enable mode, character codes 000_{16} through $1FF_{16}$ can be used. In this case, the character codes 000_{16} through $0FF_{16}$ are turned to 16 X 20-dot fonts, whereas the character codes 100_{16} through $1FF_{16}$ are turned to 24 X 32-dot fonts. Of these, however, character codes $0FE_{16}$, $0FF_{16}$, 100_{16} , and 180_{16} cannot be used.

During OSDL disable mode, character codes 000_{16} through $2FF_{16}$ can be used. In this case, all characters are turned to 16 X 20-dots. Of these, however, character codes $0FE_{16}$, $0FF_{16}$, 100_{16} , 180_{16} , 200_{16} , and 280_{16} cannot be used.

CD codes 00_{16} through $7F_{16}$ can be used. In this case, all characters are turned to 16 X 26-dot fonts. Of these, however, CD codes $3F_{16}$ and 40_{16} cannot be used.

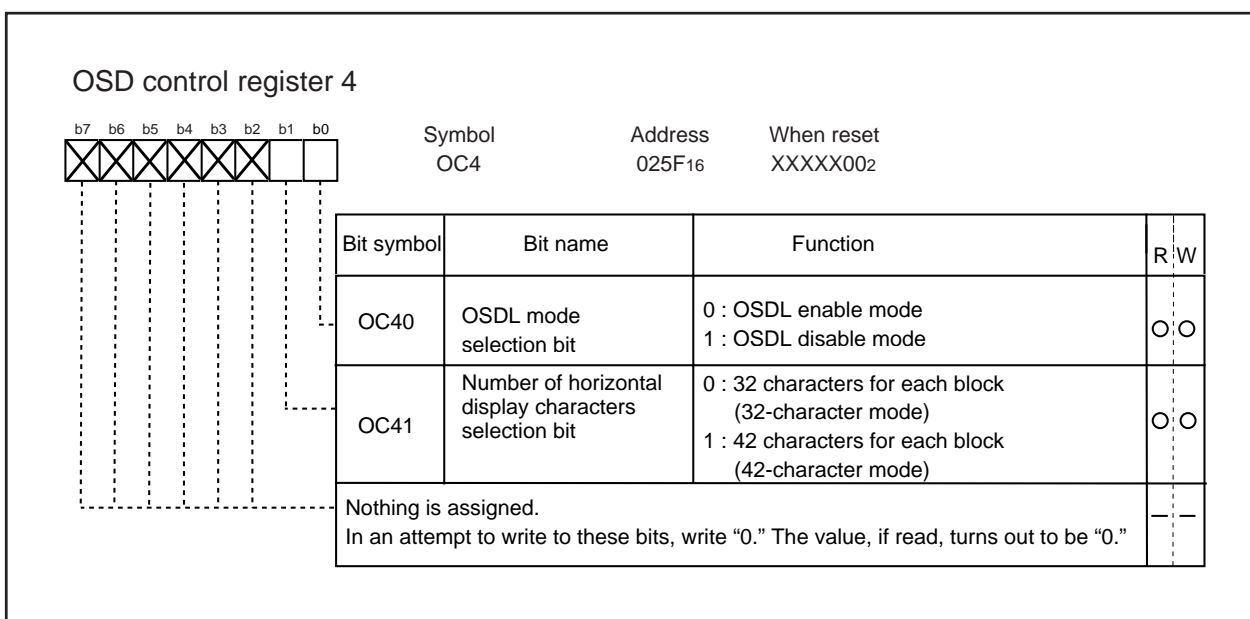
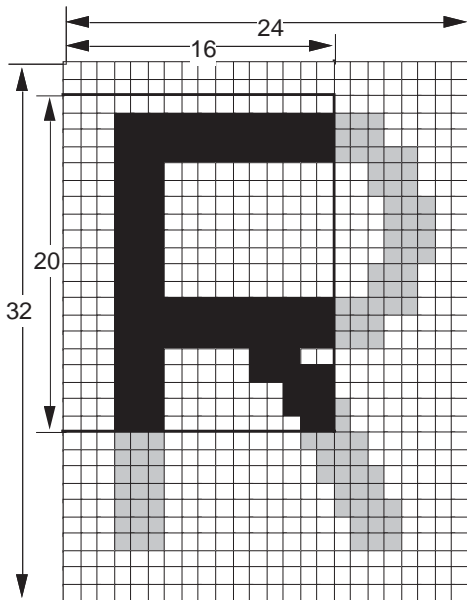


Figure 16.21 OSD control register 4

Depending on the relationship of OSDL enable/disable mode, display mode and character code, note the conditions below.

OSDL enable/ disable mode		OSDL enable mode (Bit 0 of OSD control register 4 = "0")				OSDL disable mode (Bit 0 of OSD control register 4 = "1")			
		Character size	CC	OSDS/P	OSDL	Character size	CC	OSDS/P	OSDL
Display mode & character code		Character size	CC	OSDS/P	OSDL	Character size	CC	OSDS/P	OSDL
Specified character code	000 ₁₆ to 0FF ₁₆	S	Used	Used	Not used (See note 3)	S	Used	Used	Display OFF
	100 ₁₆ to 1FF ₁₆	L	Used (See note 1)	Used (See note 1)	Used		Used	Used	Display OFF
	200 ₁₆ to 27F ₁₆	Not used (See note 3)	Not used (See note 3)	Not used (See note 3)	Not used (See note 3)		Not used (See note 3)	Used	Display OFF
	280 ₁₆ to 2FF ₁₆							Used (No border) (See note 2)	Display OFF
	300 ₁₆ to 3FF ₁₆							Not used (See note 3)	Display OFF



- Notes 1:** Part of 24 X 32 font is displayed.
- 2:** In OSDL disable mode, character codes "280₁₆" to "2FF₁₆" are used in OSDS/P mode (no border).
- 3:** As setting this make output of font data indeterminate, do not use. However, "3FE₁₆" and "3FF₁₆" can be used as character codes of blank font output in OSDP mode.

Figure 16.22 Conditions for each OSDL enable/disable mode

(2) OSD RAM (OSD RAM for character, addresses 8400₁₆ to 8EFF₁₆)

The OSD RAM for character is allocated at addresses 8400₁₆ to 8EFF₁₆, and is divided into a display character code specification part, color code 1 specification part, and color code 2 specification part for each block. The number of characters for 1 block (32- or 42-character mode) is selected by bit 1 of the OSD control register 4. Tables 16.3 to 16.7 show the address map.

For example, to display 1 character position (the left edge) in block 1, write the character code in address 8400₁₆, write color code 1 at 8401₁₆, and write color code 2 at 8480₁₆. The structure of the OSD RAM is shown in Figure 16.23.

Note : For blocks of the following dot sizes, the 3rd (n = 1 to 14) character is skipped as compared with ordinary block.

■ In OSDL mode: all dot size.

■ In OSDS and CDOSD modes of layer 2: 1.5Tc X 1/2H or 1.5Tc X 1H

Accordingly, maximum 22 characters (32-character mode)/28 characters (42-character mode) are only displayed in 1 block (refer to Fig 16.22). The RAM data for the 3rd character does not effect the display. Any character data can be stored here. And also, note the following only in 32-character mode. As the character is displayed in the 28th's character area in 42-character mode, set ordinarily.

- In OSDS mode

The character is not displayed, and only the left 1/3 part of the 22nd character background is displayed in the 22nd's character area. When not displaying this background, set transparent for character background color.

- In OSDL mode

Set a blank character or a character of transparent color to the 22nd character.

- In CDOSD mode

The character is not displayed, and color palette color specified by bits 3 to 6 of color code 1 can be output in the 22nd's character area (left 1/3 part).

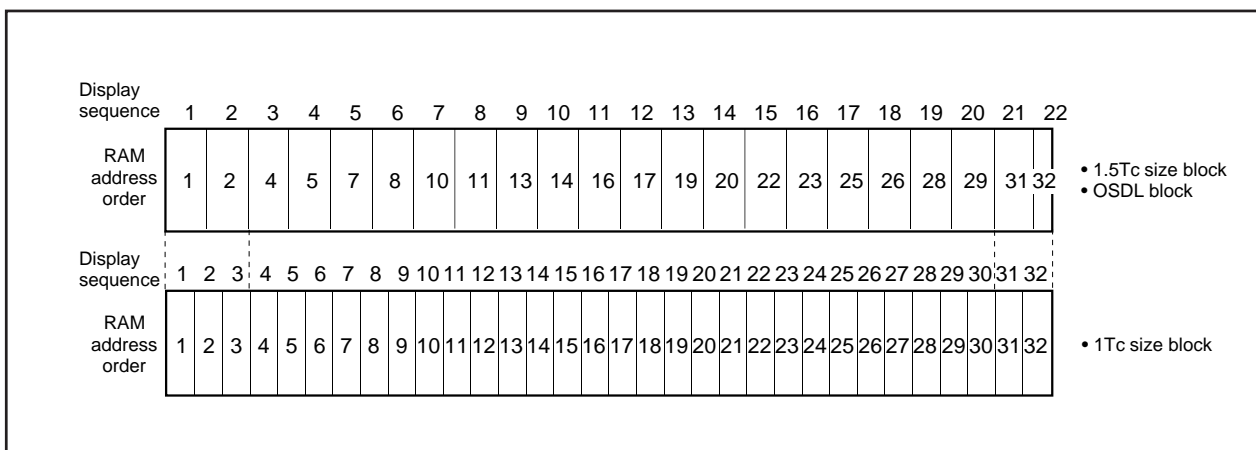


Figure 16.23 RAM data for 3rd character (in 32-character mode)

Table 16.3 Contents of OSD RAM (1st to 32nd character)

Block	Display Position (from left)	Character Code Specification	Color Code 1 Specification	Color Code 2 Specification
Block 1	1st character	8400 ₁₆	8401 ₁₆	8480 ₁₆
	2nd character	8402 ₁₆	8403 ₁₆	8482 ₁₆
	⋮	⋮	⋮	⋮
	31st character	843C ₁₆	843D ₁₆	84BC ₁₆
	32nd character	843E ₁₆	843F ₁₆	84BE ₁₆
Block 2	1st character	8440 ₁₆	8441 ₁₆	84C0 ₁₆
	2nd character	8442 ₁₆	8443 ₁₆	84C2 ₁₆
	⋮	⋮	⋮	⋮
	31st character	847C ₁₆	847D ₁₆	84FC ₁₆
	32nd character	847E ₁₆	847F ₁₆	84FE ₁₆
Block 3	1st character	8500 ₁₆	8501 ₁₆	8580 ₁₆
	2nd character	8502 ₁₆	8503 ₁₆	8582 ₁₆
	⋮	⋮	⋮	⋮
	31st character	853C ₁₆	853D ₁₆	85BC ₁₆
	32nd character	853E ₁₆	853F ₁₆	85BE ₁₆
Block 4	1st character	8540 ₁₆	8541 ₁₆	85C0 ₁₆
	2nd character	8542 ₁₆	8543 ₁₆	85C2 ₁₆
	⋮	⋮	⋮	⋮
	31st character	857C ₁₆	857D ₁₆	85FC ₁₆
	32nd character	857E ₁₆	857F ₁₆	85FE ₁₆
Block 5	1st character	8600 ₁₆	8601 ₁₆	8680 ₁₆
	2nd character	8602 ₁₆	8603 ₁₆	8682 ₁₆
	⋮	⋮	⋮	⋮
	31st character	863C ₁₆	863D ₁₆	86BC ₁₆
	32nd character	863E ₁₆	863F ₁₆	86BE ₁₆
Block 6	1st character	8640 ₁₆	8641 ₁₆	86C0 ₁₆
	2nd character	8642 ₁₆	8643 ₁₆	86C2 ₁₆
	⋮	⋮	⋮	⋮
	31st character	867C ₁₆	867D ₁₆	86FC ₁₆
	32nd character	867E ₁₆	867F ₁₆	86FE ₁₆
Block 7	1st character	8700 ₁₆	8701 ₁₆	8780 ₁₆
	2nd character	8702 ₁₆	8703 ₁₆	8782 ₁₆
	⋮	⋮	⋮	⋮
	31st character	873C ₁₆	873D ₁₆	87BC ₁₆
	32nd character	873E ₁₆	873F ₁₆	87BE ₁₆
Block 8	1st character	8740 ₁₆	8741 ₁₆	87C0 ₁₆
	2nd character	8742 ₁₆	8743 ₁₆	87C2 ₁₆
	⋮	⋮	⋮	⋮
	31st character	877C ₁₆	877D ₁₆	87FC ₁₆
	32nd character	877E ₁₆	877F ₁₆	87FE ₁₆
Block 9	1st character	8800 ₁₆	8801 ₁₆	8880 ₁₆
	2nd character	8802 ₁₆	8803 ₁₆	8882 ₁₆
	⋮	⋮	⋮	⋮
	31st character	883C ₁₆	883D ₁₆	88BC ₁₆
	32nd character	883E ₁₆	883F ₁₆	88BE ₁₆
Block 10	1st character	8840 ₁₆	8841 ₁₆	88C0 ₁₆
	2nd character	8842 ₁₆	8843 ₁₆	88C2 ₁₆
	⋮	⋮	⋮	⋮
	31st character	887C ₁₆	887D ₁₆	88FC ₁₆
	32nd character	887E ₁₆	887F ₁₆	88FE ₁₆

Table 16.4 Contents of OSD RAM (1st to 32nd character) (continued)

Block	Display Position (from left)	Character Code Specification	Color Code 1 Specification	Color Code 2 Specification
Block 11	1st character	8900 ₁₆	8901 ₁₆	8980 ₁₆
	2nd character	8902 ₁₆	8903 ₁₆	8982 ₁₆
	⋮	⋮	⋮	⋮
	31st character	893C ₁₆	893D ₁₆	89BC ₁₆
	32nd character	893E ₁₆	893F ₁₆	89BE ₁₆
Block 12	1st character	8940 ₁₆	8941 ₁₆	89C0 ₁₆
	2nd character	8942 ₁₆	8943 ₁₆	89C2 ₁₆
	⋮	⋮	⋮	⋮
	31st character	897C ₁₆	897D ₁₆	89FC ₁₆
	32nd character	897E ₁₆	897F ₁₆	89FE ₁₆
Block 13	1st character	8A00 ₁₆	8A01 ₁₆	8A80 ₁₆
	2nd character	8A02 ₁₆	8A03 ₁₆	8A82 ₁₆
	⋮	⋮	⋮	⋮
	31st character	8A3C ₁₆	8A3D ₁₆	8ABC ₁₆
	32nd character	8A3E ₁₆	8A3F ₁₆	8ABE ₁₆
Block 14	1st character	8A40 ₁₆	8A41 ₁₆	8AC0 ₁₆
	2nd character	8A42 ₁₆	8A43 ₁₆	8AC2 ₁₆
	⋮	⋮	⋮	⋮
	31st character	8A7C ₁₆	8A7D ₁₆	8AFC ₁₆
	32nd character	8A7E ₁₆	8A7F ₁₆	8AFE ₁₆
Block 15	1st character	8B00 ₁₆	8B01 ₁₆	8B80 ₁₆
	2nd character	8B02 ₁₆	8B03 ₁₆	8B82 ₁₆
	⋮	⋮	⋮	⋮
	31st character	8B3C ₁₆	8B3D ₁₆	8BBC ₁₆
	32nd character	8B3E ₁₆	8B3F ₁₆	8BBE ₁₆
Block 16	1st character	8B40 ₁₆	8B41 ₁₆	8BC0 ₁₆
	2nd character	8B42 ₁₆	8B43 ₁₆	8BC2 ₁₆
	⋮	⋮	⋮	⋮
	31st character	8B7C ₁₆	8B7D ₁₆	8BF0 ₁₆
	32nd character	8B7E ₁₆	8B7F ₁₆	8BFE ₁₆

Table 16.5 Contents of OSD RAM (33rd to 42nd character)

Block	Display Position (from left)	Character Code Specification	Color Code 1 Specification	Color Code 2 Specification
Block 1	33rd character	8C00 ₁₆	8C01 ₁₆	8C80 ₁₆
	34th character	8C02 ₁₆	8C03 ₁₆	8C82 ₁₆
	:	:	:	:
	39th character	8C0C ₁₆	8C0D ₁₆	8C8C ₁₆
	40th character	8C0E ₁₆	8C0F ₁₆	8C8E ₁₆
	41st character	8E00 ₁₆	8E01 ₁₆	8E80 ₁₆
Block 2	42nd character	8E02 ₁₆	8E03 ₁₆	8E82 ₁₆
	33rd character	8C10 ₁₆	8C11 ₁₆	8C90 ₁₆
	34th character	8C12 ₁₆	8C13 ₁₆	8C92 ₁₆
	:	:	:	:
	39th character	8C1C ₁₆	8C1D ₁₆	8C9C ₁₆
	40th character	8C1E ₁₆	8C1F ₁₆	8C9E ₁₆
Block 3	41st character	8E08 ₁₆	8E09 ₁₆	8E88 ₁₆
	42nd character	8E0A ₁₆	8E0B ₁₆	8E8A ₁₆
	33rd character	8C20 ₁₆	8C21 ₁₆	8CA0 ₁₆
	34th character	8C22 ₁₆	8C23 ₁₆	8CA2 ₁₆
	:	:	:	:
	39th character	8C2C ₁₆	8C2D ₁₆	8CAC ₁₆
Block 4	40th character	8C2E ₁₆	8C2F ₁₆	8CAE ₁₆
	41st character	8E10 ₁₆	8E11 ₁₆	8E90 ₁₆
	42nd character	8E12 ₁₆	8E13 ₁₆	8E92 ₁₆
	33rd character	8C30 ₁₆	8C31 ₁₆	8CB0 ₁₆
	34th character	8C32 ₁₆	8C33 ₁₆	8CB2 ₁₆
	:	:	:	:
Block 5	39th character	8C3C ₁₆	8C3D ₁₆	8CBC ₁₆
	40th character	8C3E ₁₆	8C3F ₁₆	8CBE ₁₆
	41st character	8E18 ₁₆	8E19 ₁₆	8E98 ₁₆
	42nd character	8E1A ₁₆	8E1B ₁₆	8E9A ₁₆
	33rd character	8C40 ₁₆	8C41 ₁₆	8CC0 ₁₆
	34th character	8C42 ₁₆	8C43 ₁₆	8CC2 ₁₆
Block 6	:	:	:	:
	39th character	8C4C ₁₆	8C4D ₁₆	8CCC ₁₆
	40th character	8C4E ₁₆	8C4F ₁₆	8CCE ₁₆
	41st character	8E20 ₁₆	8E21 ₁₆	8EA0 ₁₆
	42nd character	8E22 ₁₆	8E23 ₁₆	8EA2 ₁₆
	33rd character	8C50 ₁₆	8C51 ₁₆	8CD0 ₁₆
Block 7	34th character	8C52 ₁₆	8C53 ₁₆	8CD2 ₁₆
	:	:	:	:
	39th character	8C5C ₁₆	8C5D ₁₆	8CDC ₁₆
	40th character	8C5E ₁₆	8C5F ₁₆	8CDE ₁₆
	41st character	8E28 ₁₆	8E29 ₁₆	8EA8 ₁₆
	42nd character	8E2A ₁₆	8E2B ₁₆	8EAA ₁₆
Block 7	33rd character	8C60 ₁₆	8C61 ₁₆	8CE0 ₁₆
	34th character	8C62 ₁₆	8C63 ₁₆	8CE2 ₁₆
	:	:	:	:
	39th character	8C6C ₁₆	8C6D ₁₆	8CEC ₁₆
	40th character	8C6E ₁₆	8C6F ₁₆	8CEE ₁₆
	41st character	8E30 ₁₆	8E31 ₁₆	8EB0 ₁₆
42nd character	8E32 ₁₆	8E33 ₁₆	8EB2 ₁₆	

Table 16.6 Contents of OSD RAM (33rd to 42nd character) (continued)

Block	Display Position (from left)	Character Code Specification	Color Code 1 Specification	Color Code 2 Specification
Block 8	33rd character	8C70 ₁₆	8C71 ₁₆	8CF0 ₁₆
	34th character	8C72 ₁₆	8C73 ₁₆	8CF2 ₁₆
	⋮	⋮	⋮	⋮
	39th character	8C7C ₁₆	8C7D ₁₆	8CFC ₁₆
	40th character	8C7E ₁₆	8C7F ₁₆	8CFE ₁₆
	41st character	8E38 ₁₆	8E39 ₁₆	8EB8 ₁₆
Block 9	42nd character	8E3A ₁₆	8E3B ₁₆	8EBA ₁₆
	33rd character	8D00 ₁₆	8D01 ₁₆	8D80 ₁₆
	34th character	8D02 ₁₆	8D03 ₁₆	8D82 ₁₆
	⋮	⋮	⋮	⋮
	39th character	8D0C ₁₆	8D0D ₁₆	8D8C ₁₆
	40th character	8D0E ₁₆	8D0F ₁₆	8D8E ₁₆
Block 10	41st character	8E40 ₁₆	8E41 ₁₆	8EC0 ₁₆
	42nd character	8E42 ₁₆	8E43 ₁₆	8EC2 ₁₆
	33rd character	8D10 ₁₆	8D11 ₁₆	8D90 ₁₆
	34th character	8D12 ₁₆	8D13 ₁₆	8D92 ₁₆
	⋮	⋮	⋮	⋮
	39th character	8D1C ₁₆	8D1D ₁₆	8D9C ₁₆
Block 11	40th character	8D1E ₁₆	8D1F ₁₆	8D9E ₁₆
	41st character	8E48 ₁₆	8E49 ₁₆	8EC8 ₁₆
	42nd character	8E4A ₁₆	8E4B ₁₆	8ECA ₁₆
	33rd character	8D20 ₁₆	8D21 ₁₆	8DA0 ₁₆
	34th character	8D22 ₁₆	8D23 ₁₆	8DA2 ₁₆
	⋮	⋮	⋮	⋮
Block 12	39th character	8D2C ₁₆	8D2D ₁₆	8DAC ₁₆
	40th character	8D2E ₁₆	8D2F ₁₆	8DAE ₁₆
	41st character	8E50 ₁₆	8E51 ₁₆	8ED0 ₁₆
	42nd character	8E52 ₁₆	8E53 ₁₆	8ED2 ₁₆
	33rd character	8D30 ₁₆	8D31 ₁₆	8DB0 ₁₆
	34th character	8D32 ₁₆	8D33 ₁₆	8DB2 ₁₆
Block 13	⋮	⋮	⋮	⋮
	39th character	8D3C ₁₆	8D3D ₁₆	8DBC ₁₆
	40th character	8D3E ₁₆	8D3F ₁₆	8DBE ₁₆
	41st character	8E58 ₁₆	8E59 ₁₆	8ED8 ₁₆
	42nd character	8E5A ₁₆	8E5B ₁₆	8EDA ₁₆
	33rd character	8D40 ₁₆	8D41 ₁₆	8DC0 ₁₆
Block 14	34th character	8D42 ₁₆	8D43 ₁₆	8DC2 ₁₆
	⋮	⋮	⋮	⋮
	39th character	8D4C ₁₆	8D4D ₁₆	8DCC ₁₆
	40th character	8D4E ₁₆	8D4F ₁₆	8DCE ₁₆
	41st character	8E60 ₁₆	8E61 ₁₆	8EE0 ₁₆
	42nd character	8E62 ₁₆	8E63 ₁₆	8EE2 ₁₆
Block 14	33rd character	8D50 ₁₆	8D51 ₁₆	8DD0 ₁₆
	34th character	8D52 ₁₆	8D53 ₁₆	8DD2 ₁₆
	⋮	⋮	⋮	⋮
	39th character	8D5C ₁₆	8D5D ₁₆	8DDC ₁₆
	40th character	8D5E ₁₆	8D5F ₁₆	8DDE ₁₆
	41st character	8E68 ₁₆	8E69 ₁₆	8EE8 ₁₆
42nd character	8E6A ₁₆	8E6B ₁₆	8EEA ₁₆	

Table 16.7 Contents of OSD RAM (33rd to 42nd character) (continued)

Block	Display Position (from left)	Character Code Specification	Color Code 1 Specification	Color Code 2 Specification
Block 15	33rd character	8D60 ₁₆	8D61 ₁₆	8DE0 ₁₆
	34th character	8D62 ₁₆	8D63 ₁₆	8DE2 ₁₆
	⋮	⋮	⋮	⋮
	39th character	8D6C ₁₆	8D6D ₁₆	8DEC ₁₆
	40th character	8D6E ₁₆	8D6F ₁₆	8DEE ₁₆
	41st character	8E70 ₁₆	8E71 ₁₆	8EF0 ₁₆
Block 16	42nd character	8E72 ₁₆	8E73 ₁₆	8EF2 ₁₆
	33rd character	8D70 ₁₆	8D71 ₁₆	8DF0 ₁₆
	34th character	8D72 ₁₆	8D73 ₁₆	8DF2 ₁₆
	⋮	⋮	⋮	⋮
	39th character	8D7C ₁₆	8D7D ₁₆	8DFC ₁₆
	40th character	8D7E ₁₆	8D7F ₁₆	8DFE ₁₆
Block 16	41st character	8E78 ₁₆	8E79 ₁₆	8EF8 ₁₆
	42nd character	8E7A ₁₆	8E7B ₁₆	8EFA ₁₆

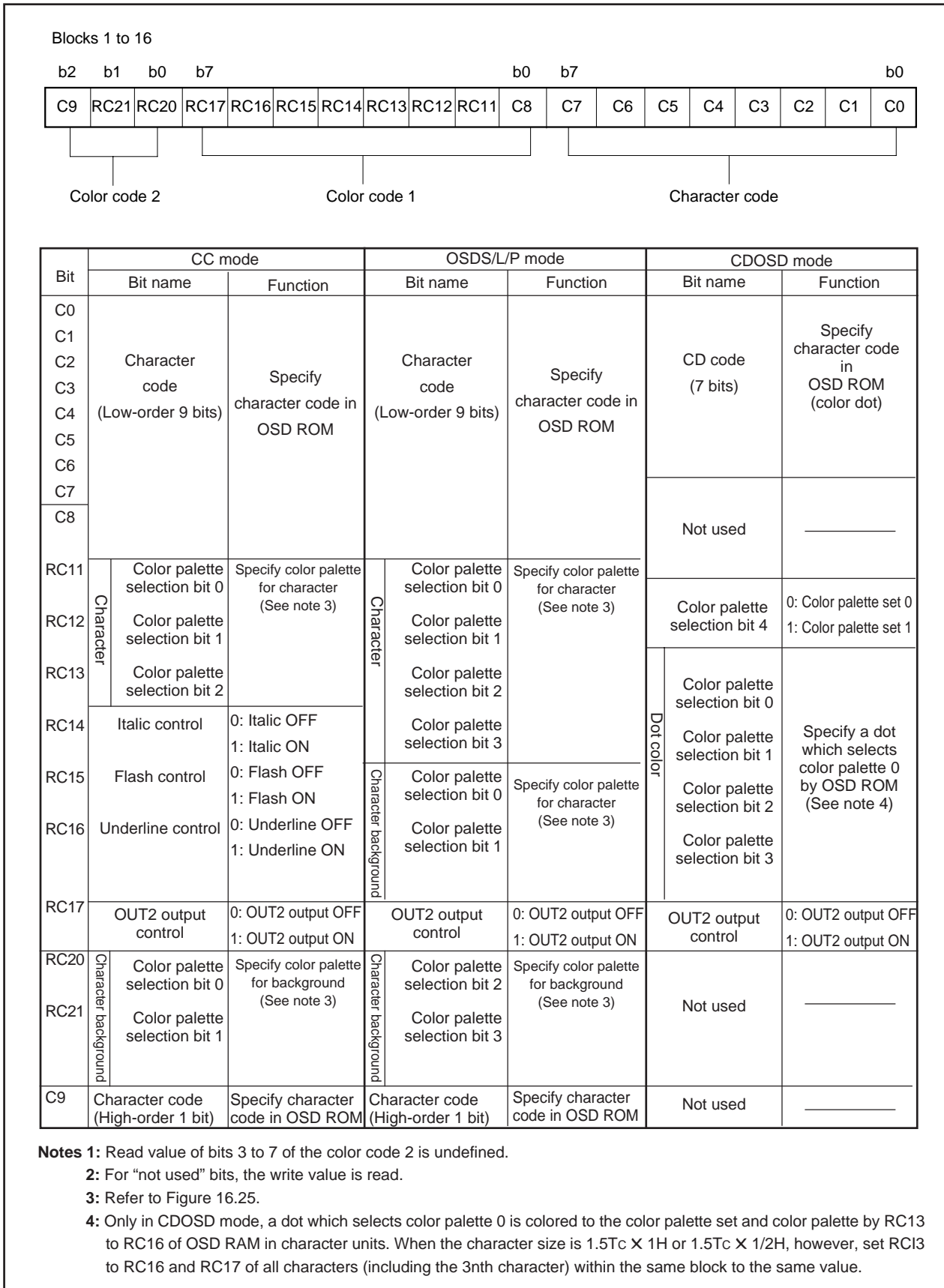


Figure 16.24 Structure of OSD RAM

(3) OSD RAM (OSD RAM for SPRITE, addresses 8000₁₆ to 83E7₁₆)

The OSD RAM for SPRITE fonts 1 and 2, consisting of 4 planes for each font, is assigned to addresses 8000₁₆ to 83E7₁₆. Each plane corresponds to each color palette selection bit and the color palette of each dot is determined from among 16 kinds.

Table 16.8 OSD RAM address (SPRITE font 1)

Planes	Plane 3 (Color palette selection bit 3)				Plane 2 (Color palette selection bit 2)				Plane 1 (Color palette selection bit 1)				Plane 0 (Color palette selection bit 0)			
	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32
Dots	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32
Bits	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0
Line 1	80C0 ₁₆	80C1 ₁₆	81C0 ₁₆	81C1 ₁₆	8080 ₁₆	8081 ₁₆	8180 ₁₆	8181 ₁₆	8040 ₁₆	8041 ₁₆	8140 ₁₆	8141 ₁₆	8000 ₁₆	8001 ₁₆	8100 ₁₆	8101 ₁₆
Line 2	80C2 ₁₆	80C3 ₁₆	81C2 ₁₆	81C3 ₁₆	8082 ₁₆	8083 ₁₆	8182 ₁₆	8183 ₁₆	8042 ₁₆	8043 ₁₆	8142 ₁₆	8143 ₁₆	8002 ₁₆	8003 ₁₆	8102 ₁₆	8103 ₁₆
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Line 19	80E4 ₁₆	80E5 ₁₆	81E4 ₁₆	81E5 ₁₆	80A4 ₁₆	80A5 ₁₆	81A4 ₁₆	81A5 ₁₆	8064 ₁₆	8065 ₁₆	8164 ₁₆	8165 ₁₆	8024 ₁₆	8025 ₁₆	8124 ₁₆	8125 ₁₆
Line 20	80E6 ₁₆	80E7 ₁₆	81E6 ₁₆	81E7 ₁₆	80A6 ₁₆	80A7 ₁₆	81A6 ₁₆	81A7 ₁₆	8066 ₁₆	8067 ₁₆	8166 ₁₆	8167 ₁₆	8026 ₁₆	8027 ₁₆	8126 ₁₆	8127 ₁₆

Table 16.9 OSD RAM address (SPRITE font 2)

Planes	Plane 3 (Color palette selection bit 3)				Plane 2 (Color palette selection bit 2)				Plane 1 (Color palette selection bit 1)				Plane 0 (Color palette selection bit 0)			
	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32
Dots	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32
Bits	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0
Line 1	82C0 ₁₆	82C1 ₁₆	83C0 ₁₆	83C1 ₁₆	8280 ₁₆	8281 ₁₆	8380 ₁₆	8381 ₁₆	8240 ₁₆	8241 ₁₆	8340 ₁₆	8341 ₁₆	8200 ₁₆	8201 ₁₆	8300 ₁₆	8301 ₁₆
Line 2	82C2 ₁₆	82C3 ₁₆	83C2 ₁₆	83C3 ₁₆	8282 ₁₆	8283 ₁₆	8382 ₁₆	8383 ₁₆	8242 ₁₆	8243 ₁₆	8342 ₁₆	8343 ₁₆	8202 ₁₆	8203 ₁₆	8302 ₁₆	8303 ₁₆
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Line 19	82E4 ₁₆	82E5 ₁₆	83E4 ₁₆	83E5 ₁₆	82A4 ₁₆	82A5 ₁₆	83A4 ₁₆	83A5 ₁₆	8264 ₁₆	8265 ₁₆	8364 ₁₆	8365 ₁₆	8224 ₁₆	8225 ₁₆	8324 ₁₆	8325 ₁₆
Line 20	82E6 ₁₆	82E7 ₁₆	83E6 ₁₆	83E7 ₁₆	82A6 ₁₆	82A7 ₁₆	83A6 ₁₆	83A7 ₁₆	8266 ₁₆	8267 ₁₆	8366 ₁₆	8367 ₁₆	8226 ₁₆	8227 ₁₆	8326 ₁₆	8327 ₁₆

Character Color

As shown in Figure 16.25, there are 16 built-in color codes. Color palette 0 is fixed at transparent, and color palette 8 is fixed at black. The remaining 14 colors can be set to any of the 512 colors available. The setting procedure for character colors is as follows:

- CC mode 8 kinds

Color palette selection range (color palettes 0 to 7 or 8 to 15) can be selected by bit 0 of the OSD control register 3 (address 0207₁₆). Color palettes are set by bits RC11 to RC13 of the OSD RAM from among the selection range.

- OSDS/L/P mode 16 kinds

Color palettes are set by bits RC11 to RC14 of the OSD RAM.

- CDOSD mode 16 kinds

Color palettes are set in dot units according to CD font data.

Only in CDOSD mode, a dot which selects color palette 0 or 8 is colored to the color palette set by RC13 to RC16 of OSD RAM in character units (refer to Figure 16.25). And, selection of color palette set is possible by RC12 of OSDRAM.

- SPRITE display 16 kinds

Color palettes are set in dot units according to the CD font data.

Notes 1: Color palette 8 is always selected for bordering and solid space output (OUT 1 output) regardless of the set value in the register.

2: Color palette 0 (transparent) and the transparent setting of other color palettes will differ. When there are multiple layers overlapping (on top of each other, piled up), and the priority layer is color palette 0 (transparent), the bottom layer is displayed, but if the priority layer is the transparent setting of any other color palette, the background is displayed without displaying the bottom layer (refer to Figure 16.27).

Character Background Color

The display area around the characters can be colored in with a character background color. Character background colors are set in character units.

- CC mode 4 kinds

Color palette selection range (color codes 0 to 3, 4 to 7, 8 to 11, or 12 to 15) can be selected by bits 1 and 2 of the OSD control register 3 (address 0207₁₆). Color palettes are set by bits RC20 and RC21 of the OSD RAM from among the selection range.

- OSDS/L/P mode 16 kinds

Color palettes are set by bits RC15, RC16, RC20, and RC21 of the OSD RAM.

Note: The character background is displayed in the following part:

(character display area) – (character font) – (border).

Accordingly, the character background color and the color signal for these two sections cannot be mixed.

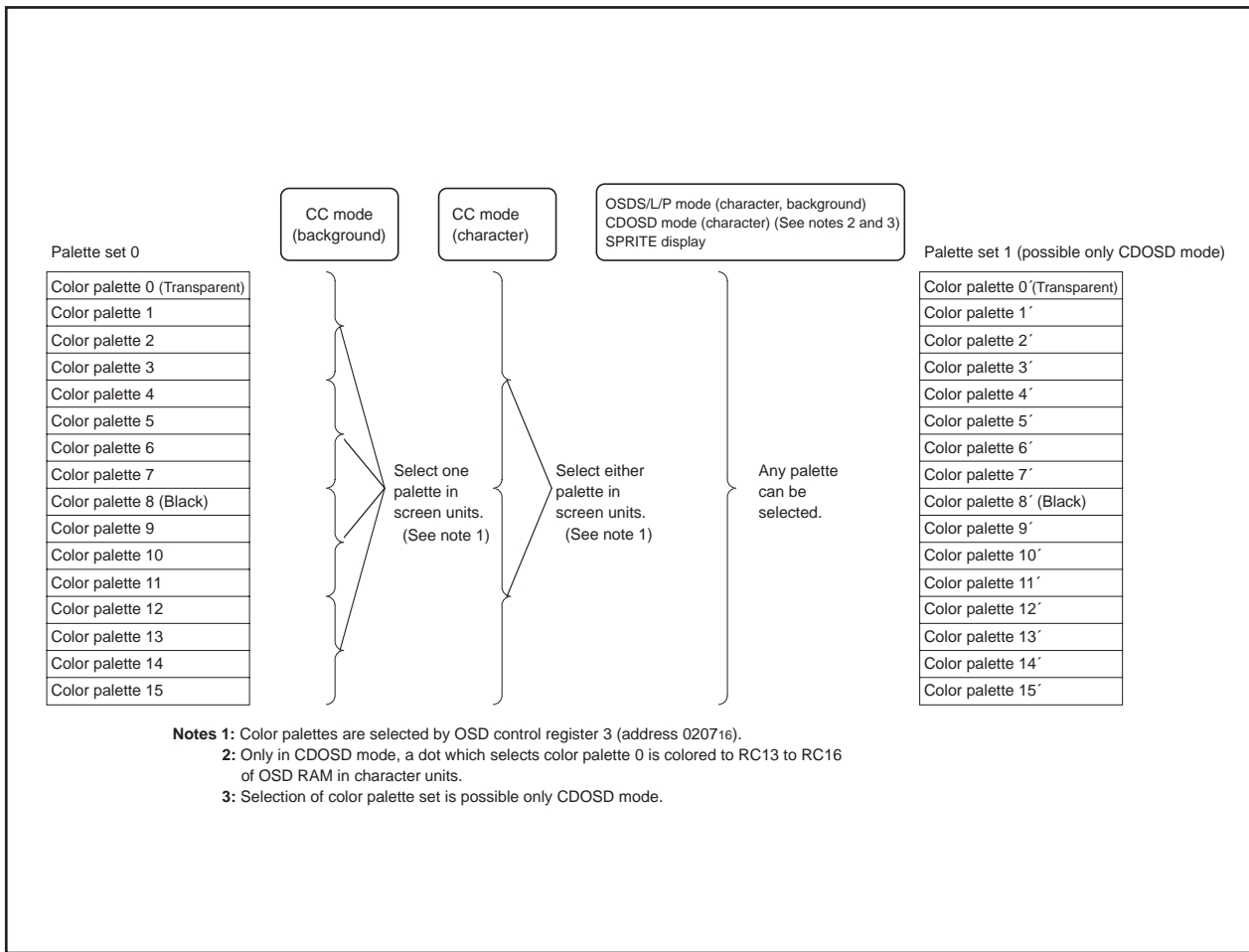


Figure 16.25 Color palette selection

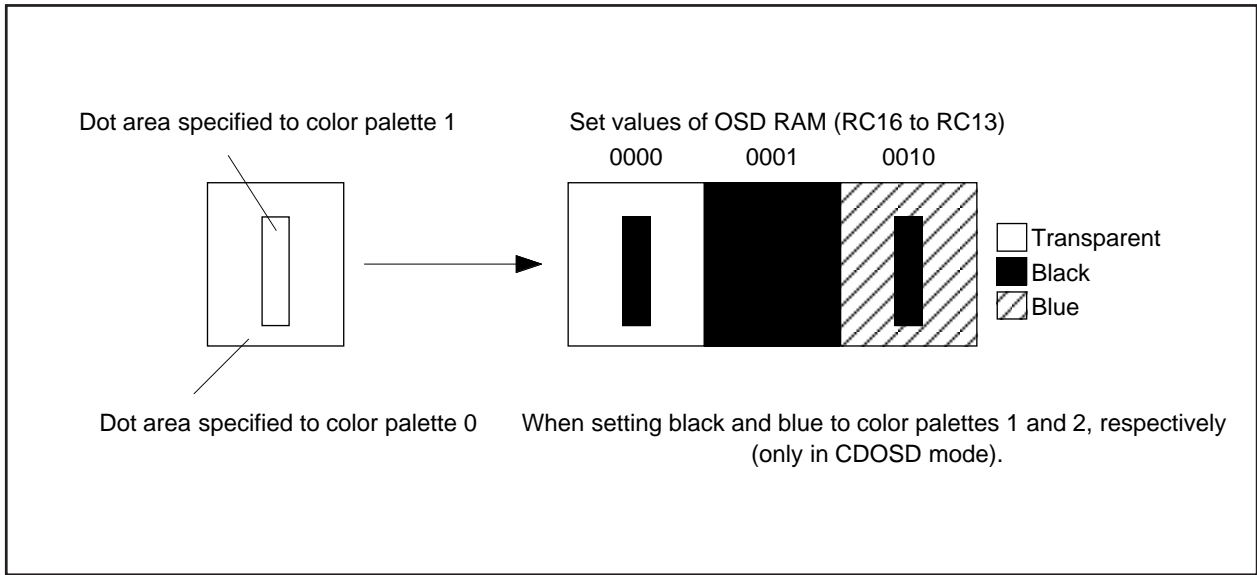


Figure 16.26 Set of color palette 0 or 8 in CDOSD mode

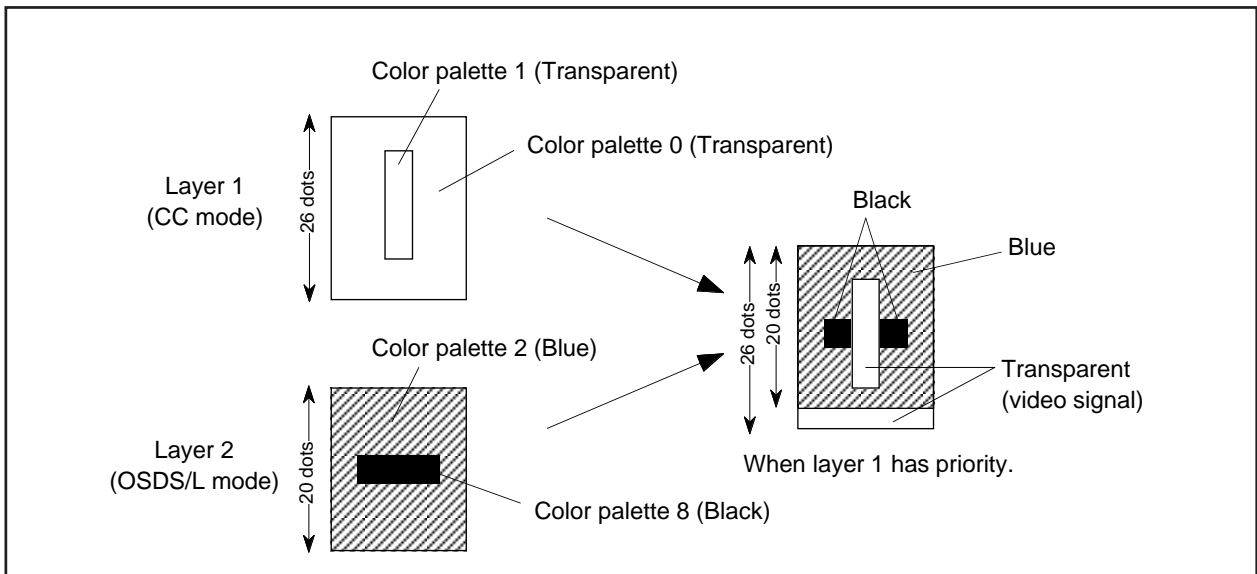


Figure 16.27 Difference between color palette 0 (transparent) and transparent setting of other color palettes

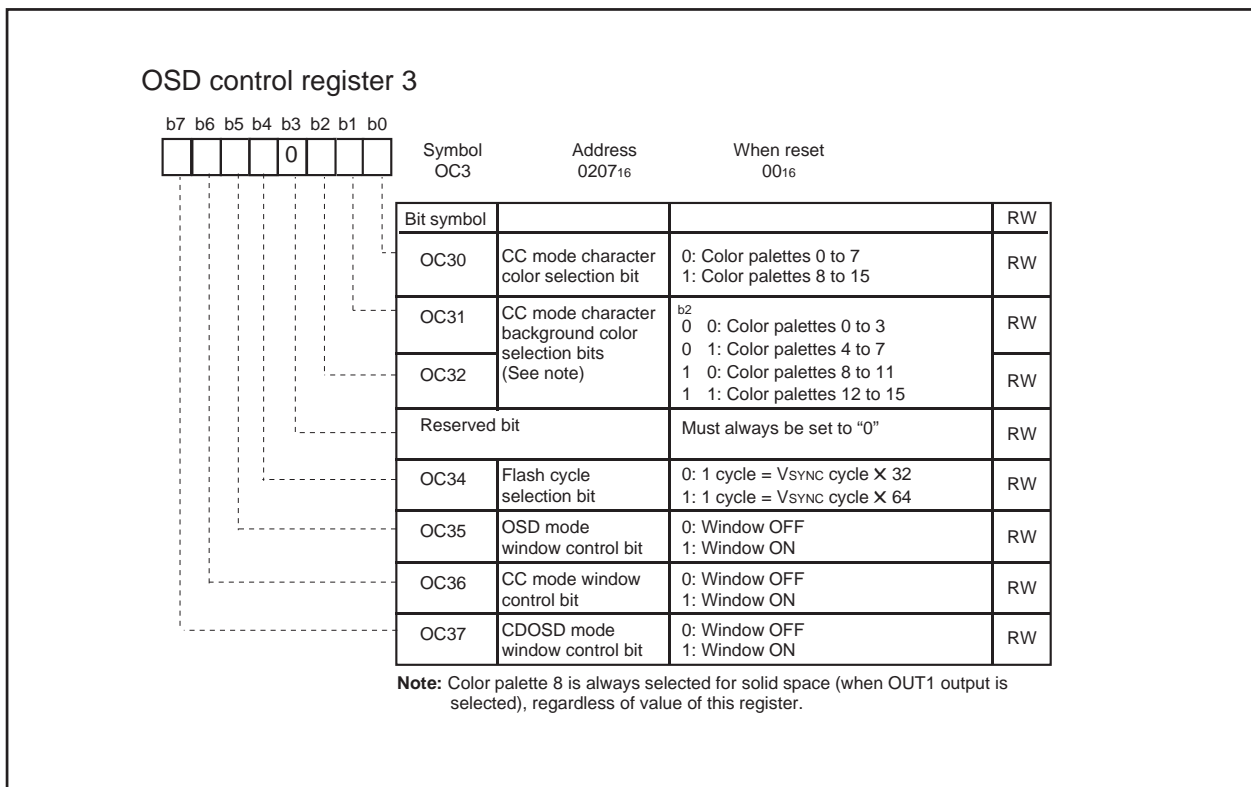


Figure 16.28 OSD control register 3

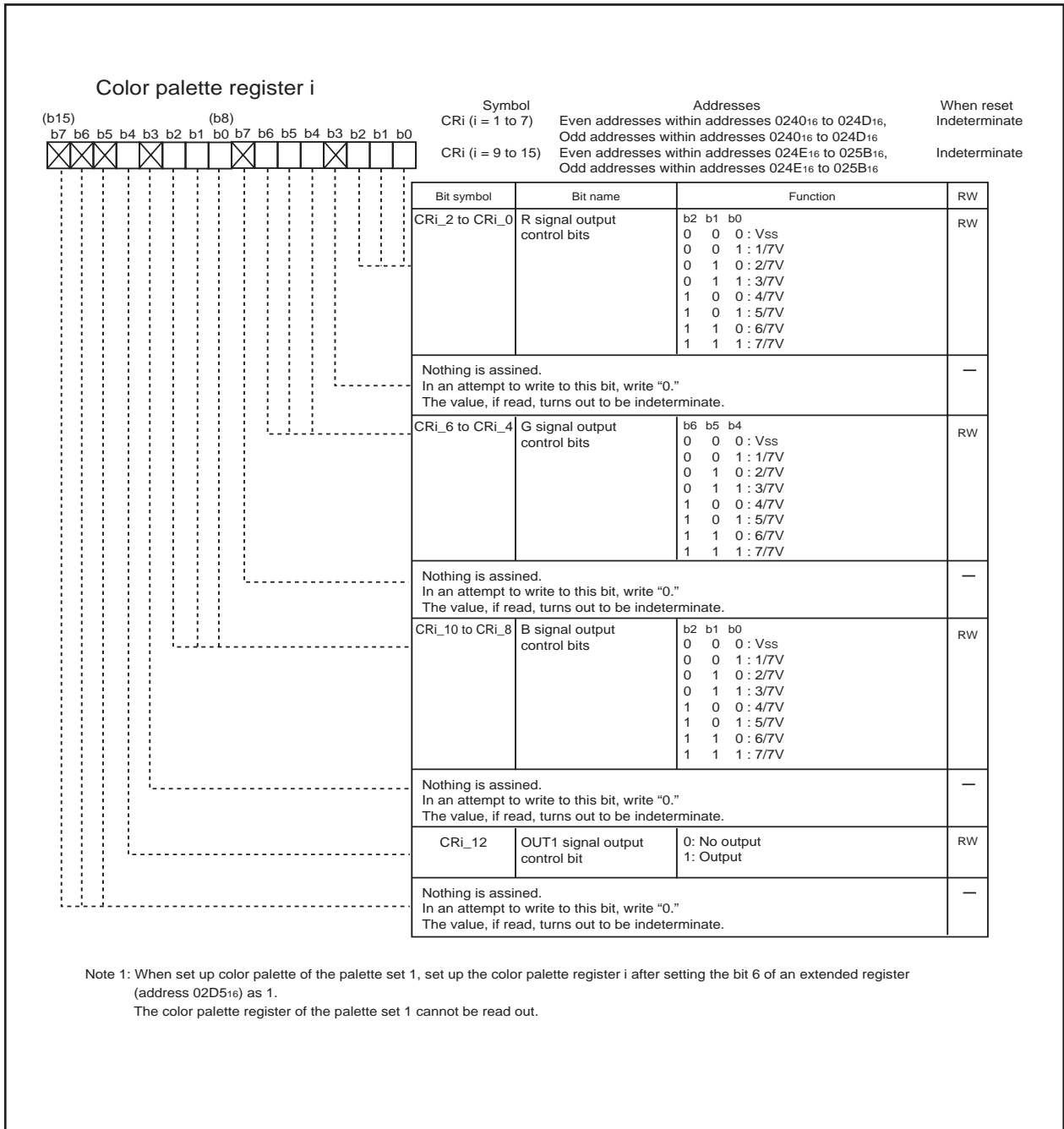


Figure 16.29 Color palette register i (i = 1 to 7, 9 to 15)

OUT1, OUT2 Signals

The OUT1, OUT2 signals are used to control the luminance of the video signal. The output waveform of the OUT1, OUT2 signals is controlled by bit 6 of the color palette register *i* (refer to Figure 16.29), bits 0 to 2 of the block control register *i* (refer to Figure 16.4) and RC17 of OSD RAM. The setting values for controlling OUT1, OUT2 and the corresponding output waveform is shown in Figure 16.30.

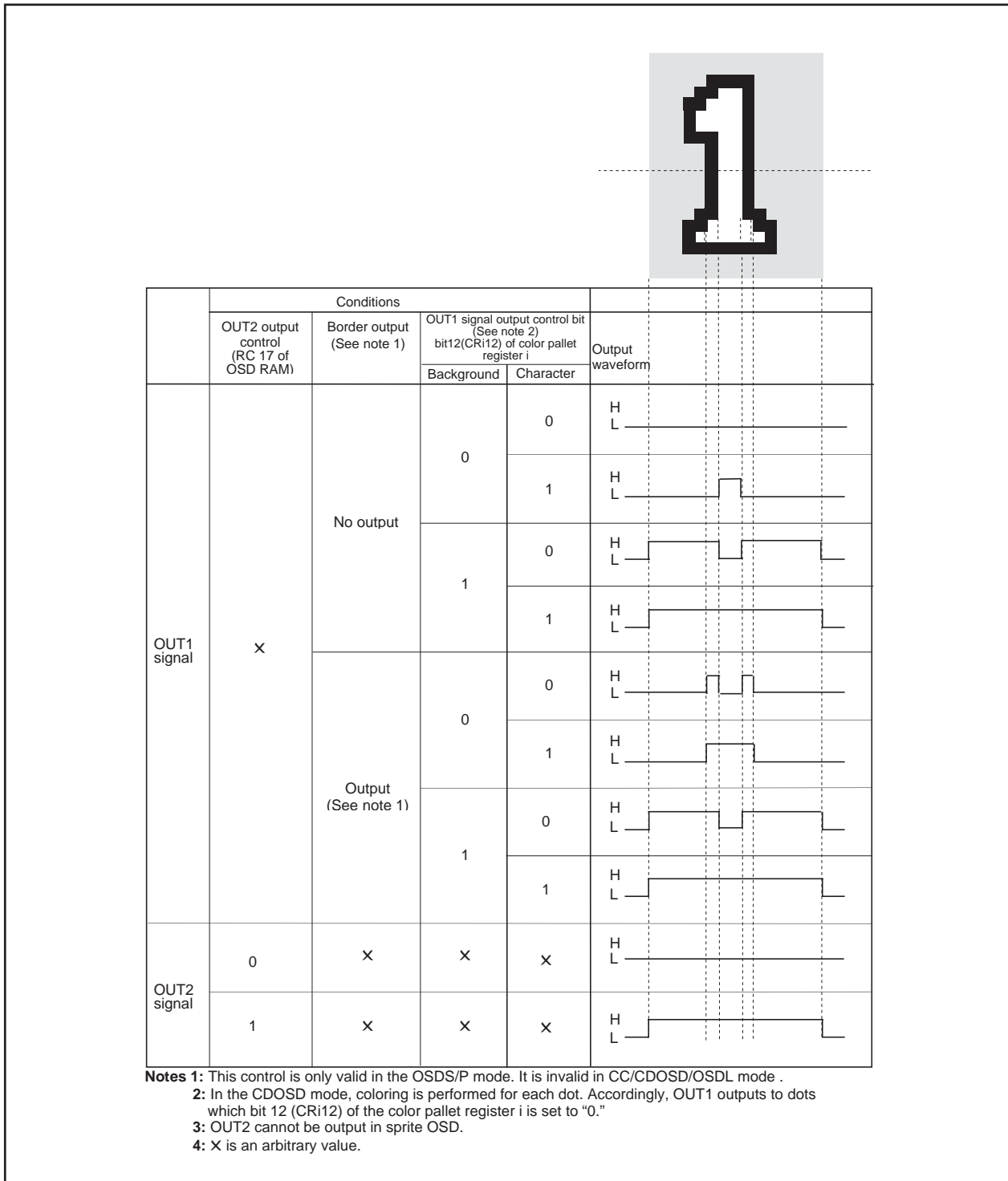


Figure 16.30 Setting value for controlling OUT1, OUT2 and corresponding output waveform

Attribute

The attributes (flash, underline, italic fonts) are controlled to the character font. The attributes to be controlled are different depending on each mode.

CC mode Flash, underline, italic for each character

OSDS/P mode Border (all bordered, shadow bordered can be selected) for each block

(1) Underline

The underline is output at the 23rd and 24th lines in vertical direction only in the CC mode. The underline is controlled by RC16 of OSD RAM. The color of underline is the same color as that of the character font.

(2) Flash

The parts of the character font, the underline, and the character background are flashed only in the CC mode. The flash for each character is controlled by RC15 of OSD RAM. The ON/OFF for flash is controlled by bit 3 of the OSD control register 1 (refer to Figure 16.3). When this bit is "0," only character font and underline flash. When "1," for a character without solid space output, R, G, B and OUT1 (all display area) flash, for a character with solid space output, only R, G, and B (all display area) flash. The flash cycle bases on the VSYNC count and is selected by bit 4 of OSD control register 3.

<NTSC method>

- When bit 4 = "0"
 - VSYNC cycle X 24 ≈ 400 ms (at flash ON)
 - VSYNC cycle X 8 ≈ 133 ms (at flash OFF)
- When bit 4 = "1"
 - VSYNC cycle X 48 ≈ 800 ms (at flash ON)
 - VSYNC cycle X 8 ≈ 267 ms (at flash OFF)

(3) Italic

The italic is made by slanting the font stored in OSD ROM to the right only in the CC mode. The italic is controlled by RC14 of OSD RAM.

The display example attribute is shown in Figure 16.31. In this case, "R" is displayed.

Notes 1: When setting both the italic and the flash, the italic character flashes.

2: When a flash character (with flash character background) adjoin on the right side of a non-flash italic character, parts out of the non-flash italic character is also flashed.

3: OUT2 is not flashed.

4: When the pre-divide ratio = 1, the italic character with slant of 1 dot X 5 steps is displayed ; when the pre-divide ratio = 2, the italic character with slant of 1/2 dot X 10 steps is displayed (refer to Figure 16.32 (c), (d)).

5: The boundary of character color is displayed in italic. However, the boundary of character background color is not affected by the italic (refer to Figure 16.32).

6: The adjacent character (one side or both side) to an italic character is displayed in italic even when the character is not specified to display in italic (refer to Figure 16.32).

7: When displaying the 32nd character (in 32-character mode)/42nd character (in 42-character mode) in the italic and when solid space is off (OC14 = "0"), parts out of character area is not displayed (refer to Figure 16.32).

8: When use the italic character which the pre-divide ratio = 1, do not use the character in which dot data exists for the right end of a font.

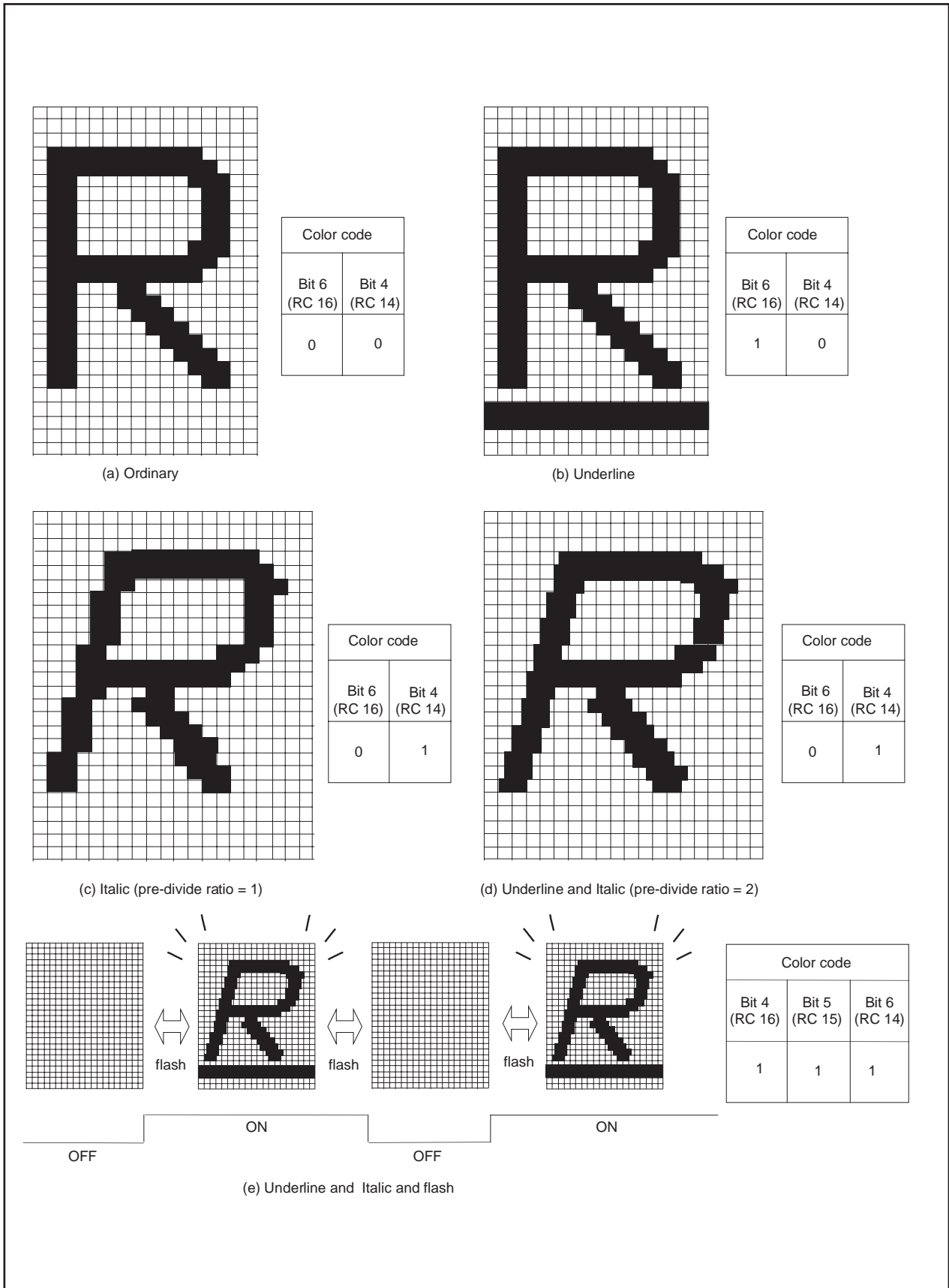


Figure 16.31 Example of attribute display (in CC mode)

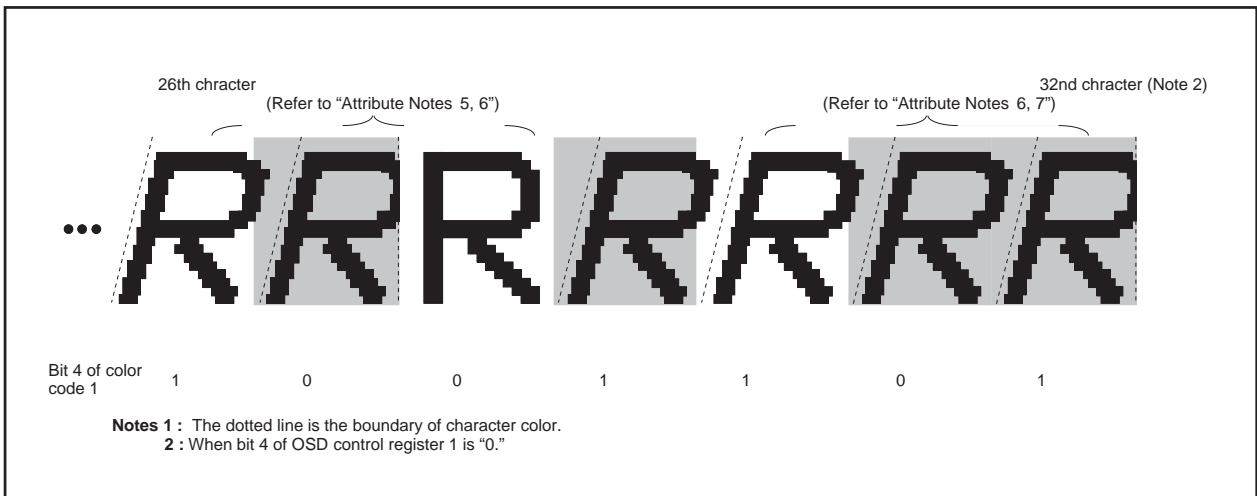


Figure 16.32 Example of italic display

(4) Border

The border is output in the OSD/S/P mode. The all bordered (bordering around of character font) and the shadow bordered (bordering right and bottom sides of character font) are selected (refer to Figure 16.33) by bit 2 of the OSD control register 1 (refer to Figure 16.3). The ON/OFF switch for borders can be controlled in block units by bits 0 to 2 of the block control register i (refer to Figure 16.4).

The OUT1 signal is used for border output. The border color is fixed at color palette 8 (black). The border color for each screen is specified by the border color register i.

The horizontal size (x) of border is 1Tc (OSD clock cycle divided in the pre-divide circuit) regardless of the character font dot size. However, only when the pre-divide ratio = 2 and character size = 1.5Tc, the horizontal size is 1.5Tc. The vertical size (y) different depending on the screen scan mode and the vertical dot size of character font.

Notes 1 : The border dot area is the shaded area as shown in Figure 16.33.

2 : When the border dot overlaps on the next character font, the character font has priority (refer to Figure 16.36 A). When the border dot overlaps on the next character background, the border has priority (refer to Figure 16.36 B).

3 : The border in vertical out of character area is not displayed (refer to Figure 16.36).

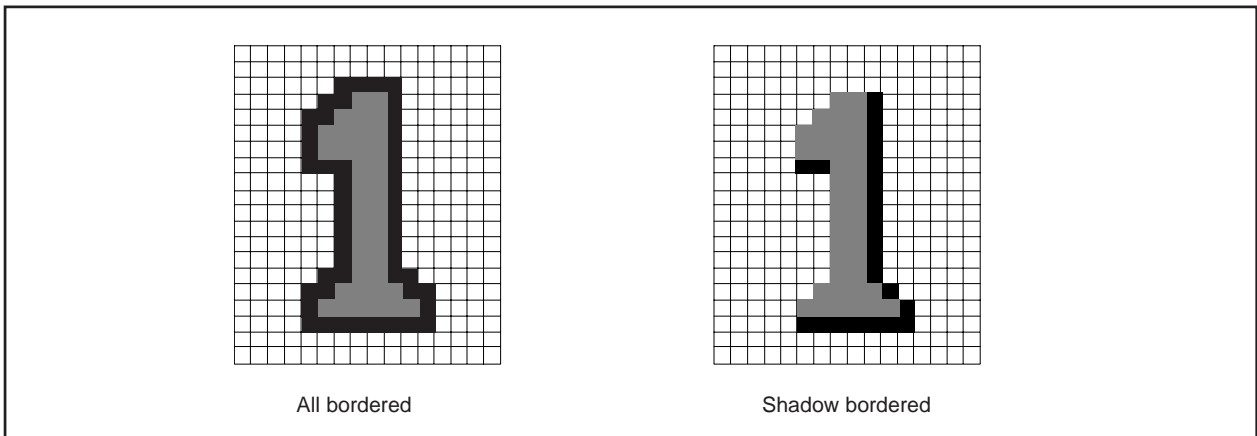


Figure 16.33 Example of border display

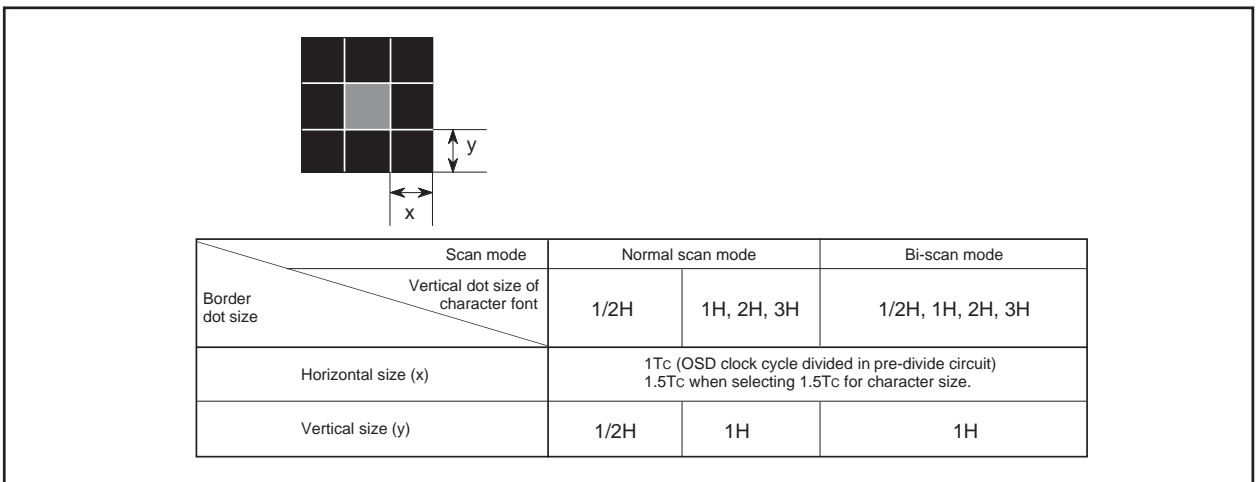


Figure 16.34 Horizontal and vertical size of border

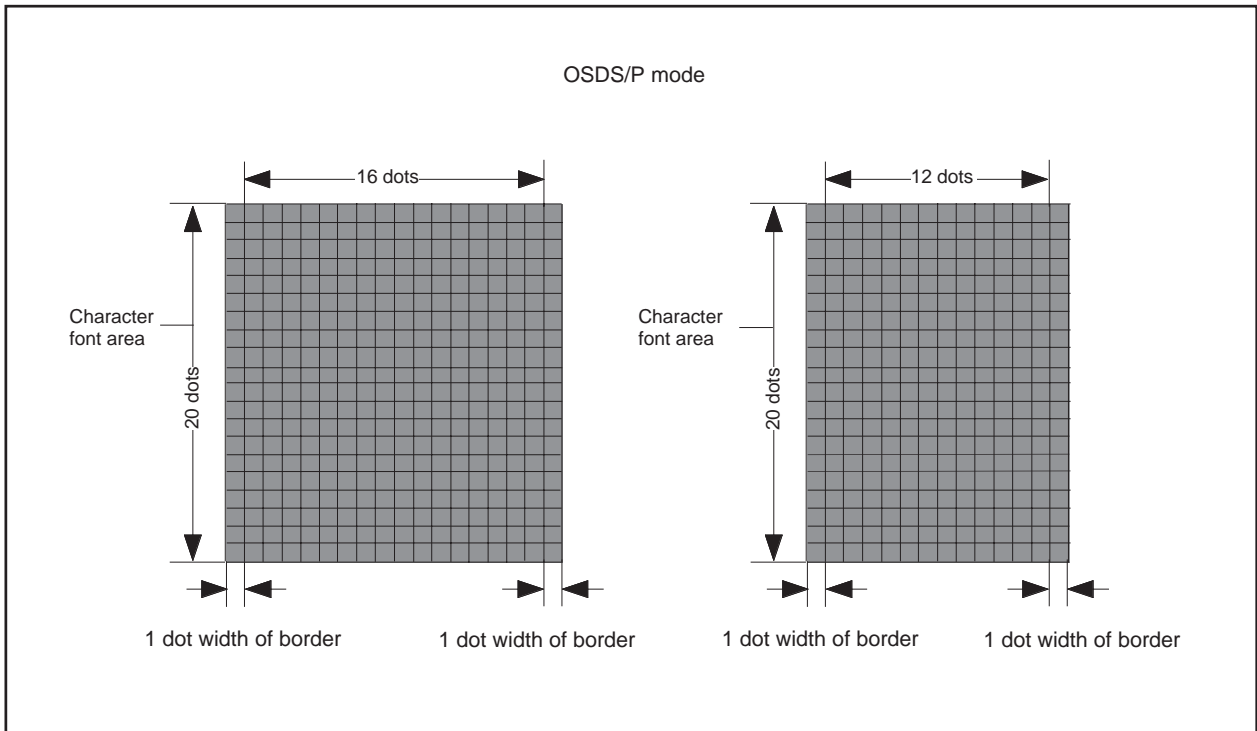


Figure 16.35 Border area

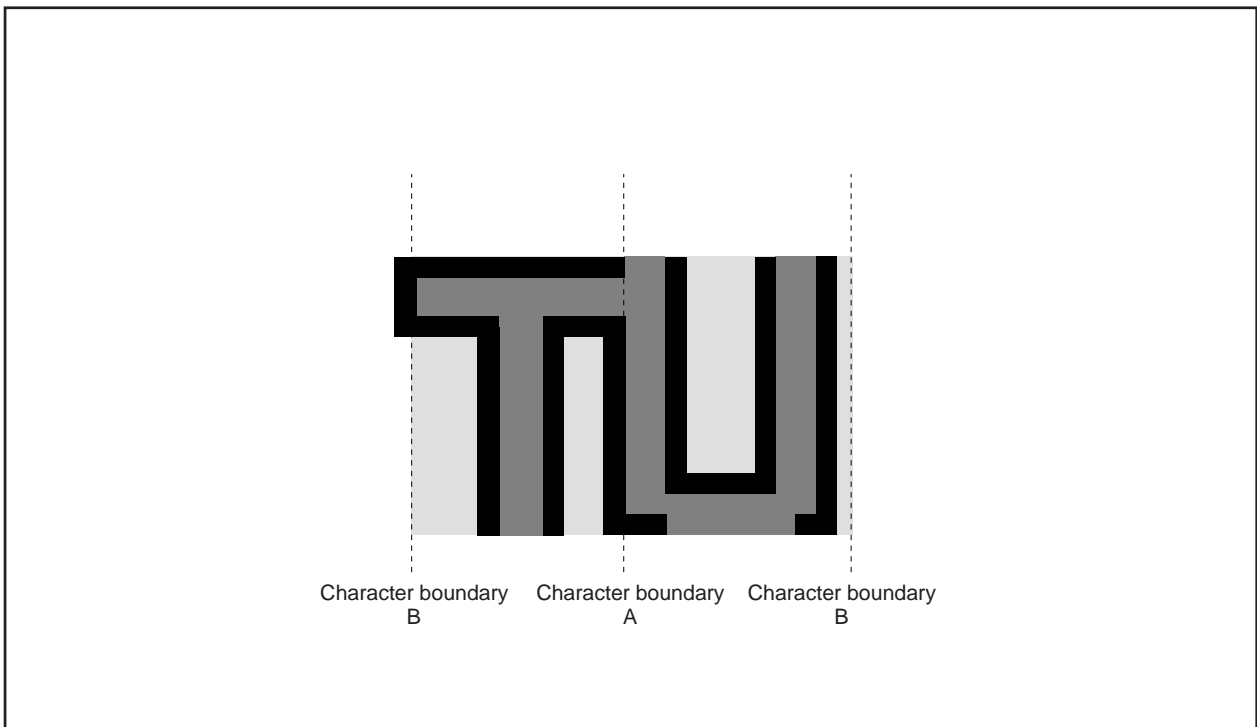


Figure 16.36 Border priority

Automatic Solid Space Function

This function generates automatically the solid space (OUT1 or OUT2 blank output) of the character area in the CC mode.

The solid space is output in the following area :

- the character area except character code “00916 ”
- the character area on the left and right sides

This function is turned on and off by bit 4 of the OSD control register 1 (refer to Figure 16.3).

OUT1 or OUT2 output is selected by bit 3 of the OSD control register 2.

Notes 1: When selecting OUT1 as solid space output, character background color with solid space output is fixed to color palette 8 (black) regardless of setting.

2: When selecting any font except blank font as the character code “00916,” the set font is output.

Table 16.10 Setting for automatic solid space

Bit 4 of OSD control register 1	0				1			
Bit 3 of OSD control register 2	0		1		0		1	
RC17 of OSD RAM	0	1	0	1	0	1	0	1
OUT1 output signal	•Character font area •Character background area		•Character font area •Character background area		•Solid space area		•Character font area •Character background area	
OUT2 output signal	OFF	•Character display area	OFF	•Character display area	OFF	•Character display area	•Solid space	•Solid space •Character display area

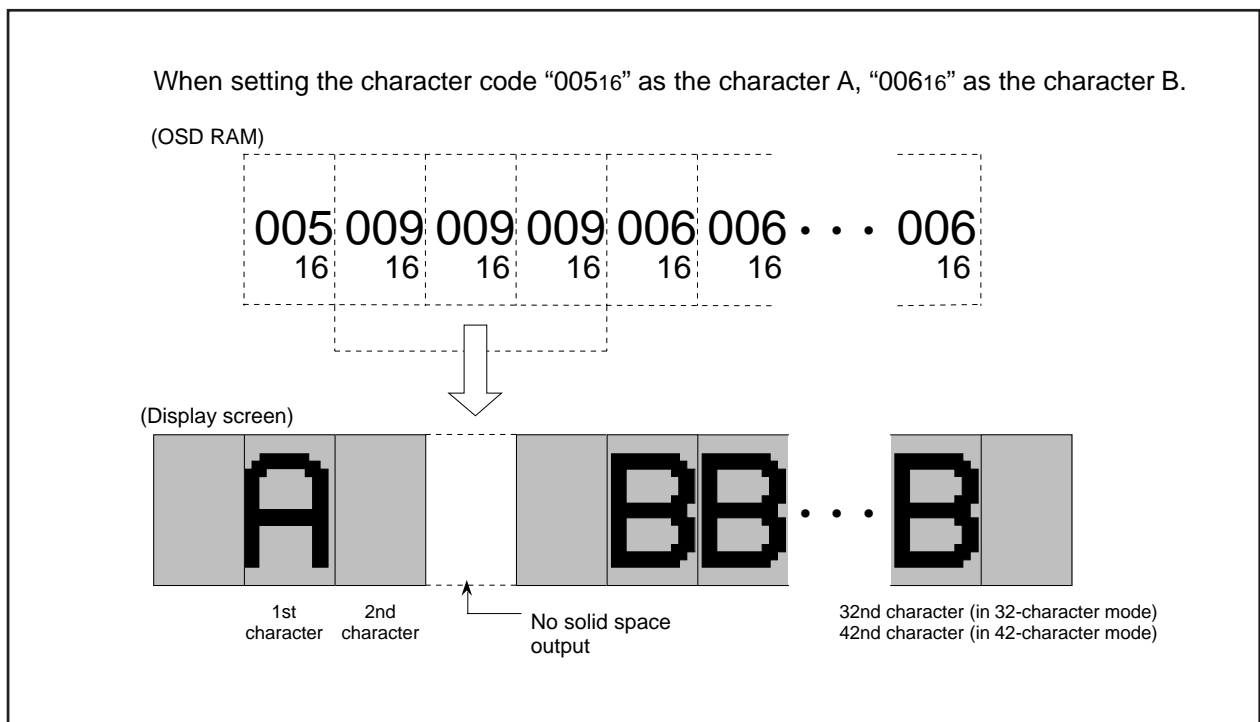


Figure 16.37 Display screen example of automatic solid space

Particular OSD Mode Block

This function can display with mixing the fonts below within the OSD mode block.

<horizontal dot structure with vertical dot structure of 20 dots>

- 16 dots
- 12 dots
- 8 dots
- 4 dots

Each font is selected by a character code. Figure 16.38 shows the display example of particular OSD mode block and Table 16.11 shows the corresponding between character codes and display fonts.

Note: As for 8 X 20-dot and 4 X 20-dot fonts, only these character background color can be displayed. And also, any character is not displayed on the right side area nor any following areas of these fonts.

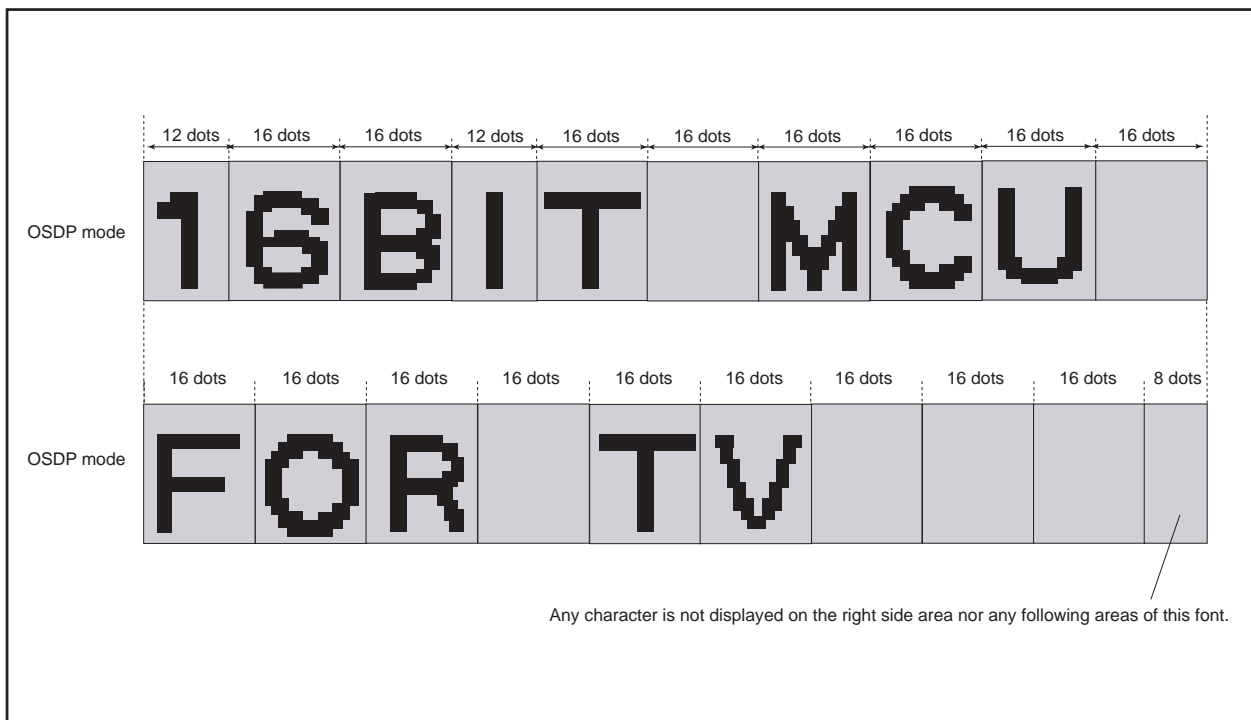
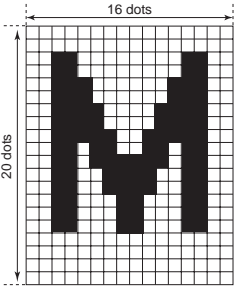
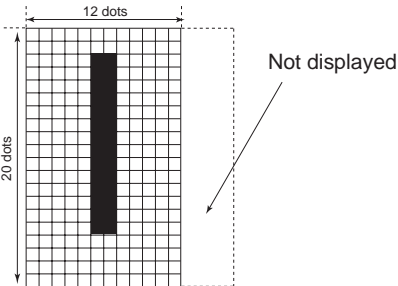
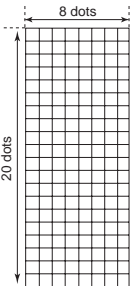
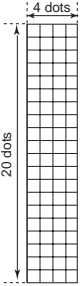


Figure 16.38 Display example of OSD mode block

Table 16.11 Corresponding between character codes and display fonts

Character code	Display fonts	Notes
000 ₁₆ to 0EF ₁₆ , 100 ₁₆ to 2FF ₁₆ (except 100 ₁₆ , 180 ₁₆ , 200 ₁₆ , 280 ₁₆)		
0F0 ₁₆ to 0FD ₁₆		<ul style="list-style-type: none"> • The left 12-dot part (16 X 12 dots) of set font is displayed. • In CC and OSDS modes, entire part (16 X 20 dots) of set font is displayed.
3FE ₁₆		<ul style="list-style-type: none"> • The blank font (only character background) is displayed. • Any character is not displayed on the right side area nor any following areas of this font. • Do not set this font for the 1st character (left edge) of a block.
3FF ₁₆		<ul style="list-style-type: none"> • The blank font (only character background) is displayed. • Any character is not displayed on the right side area nor any following areas of this font. • Do not set this font for the 1st character (left edge) of a block.

Multiline Display

This microcomputer can ordinarily display 16 lines on the CRT screen by displaying 16 blocks at different vertical positions. In addition, it can display up to 16 lines by using OSD1 interrupts.

An OSD1 interrupt request occurs at the point at which display of each block has been completed. In other words, when a scanning line reaches the point of the display position (specified by the vertical position registers) of a certain block, the character display of that block starts, and an interrupt occurs at the point at which the scanning line exceeds the block. The mode in which an OSD1 interrupt occurs is different depending on the setting of the OSD control register 2 (refer to Figure 16.7).

- When bit 7 of the OSD control register 2 is "0"
An OSD1 interrupt request occurs at the completion of layer 1 block display.
- When bit 7 of the OSD control register 2 is "1"
An OSD1 interrupt request occurs at the completion of layer 2 block display.

Notes 1: An OSD1 interrupt does not occur at the end of display when the block is not displayed. In other words, if a block is set to off display by the display control bit of the block control register *i* (addresses 0210₁₆ to 021F₁₆), an OSD1 interrupt request does not occur (refer to Figure 16.39 (A)).

2: When another block display appears while one block is displayed, an OSD1 interrupt request occurs only once at the end of the another block display (refer to Figure 16.39 (B)).

3: On the screen setting window, an OSD1 interrupt occurs even at the end of the CC mode block (off display) out of window (refer to Figure 16.39 (C)).

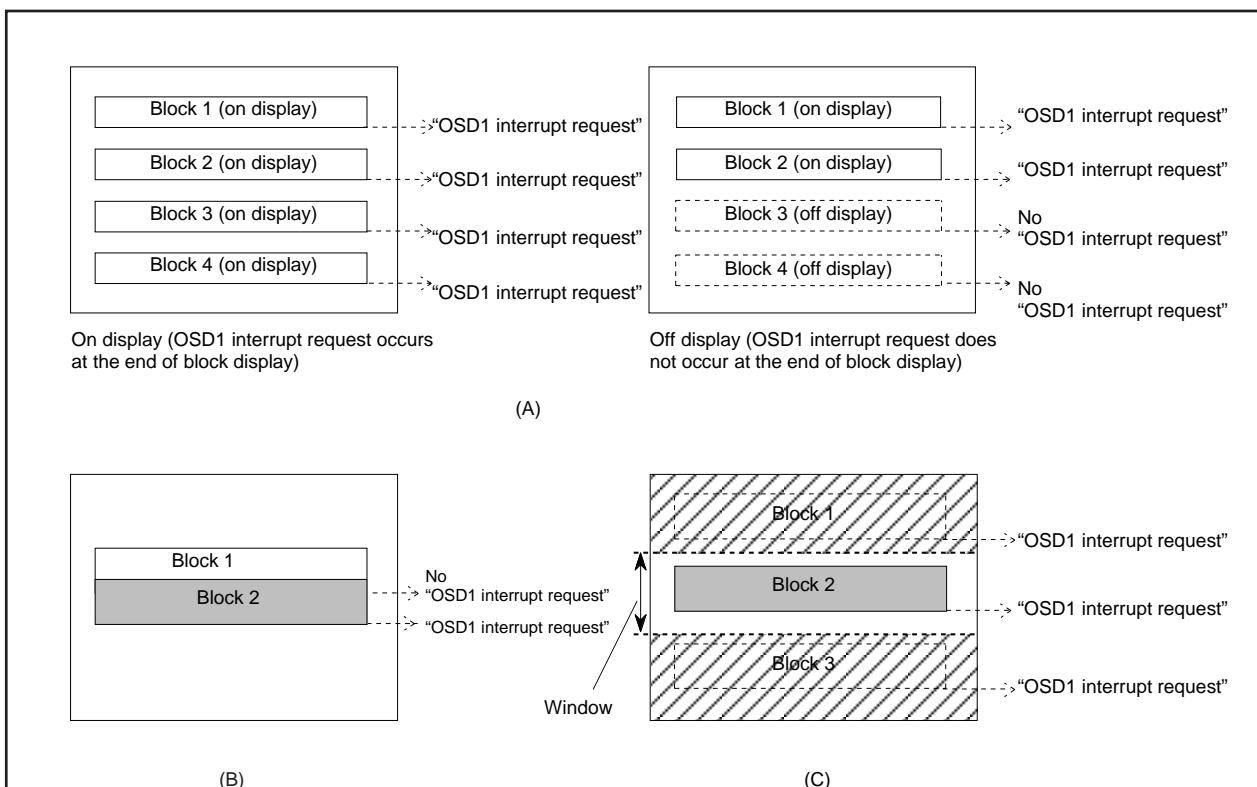


Figure 16.39 Note on occurrence of OSD1 interrupt

SPRITE OSD Function

This is especially suitable for cursor and other displays as its function allows for display in any position, regardless of the validity of block OSD displays or display positions. SPRITE font consists of 2 characters: SPRITE fonts 1 and 2. Each SPRITE font is a RAM font consisting of 32 horizontal dots X 20 vertical dots, 4 planes, and 4 bits of data per dot. Each plane has corresponding color palette selection bit, and 16 kinds of color palettes can be selected by the plane bit combination (three bits) for each dot. The color palette is set in dot units according to the OSD RAM (SPRITE) contents from among the selection range. It is possible to add arbitrary font data by software as the SPRITE fonts consist of RAM font.

The SPRITE OSD control register can control SPRITE display and dot size. The display position can also be set independently of the block display by the SPRITE horizontal position registers and the sprite horizontal vertical position registers. The vertical fonts 1 and 2 can be set independently. OSD2 interrupt request occurs at each completion of font display. The horizontal position is set in 2048 steps in 2Tosc units, and the vertical position is set in 1024 steps in 1TH units.

When SPRITE display overlaps with other OSD displays, SPRITE display is always given priority. However, the SPRITE display overlaps with the display which includes OUT2 output, OUT2 in the OSD is output without masking.

Notes 1: The SPRITE OSD function cannot output OUT2.

2: When using SPRITE OSD, do not set $HS \leq "00316"$, $HS \geq "80016"$.

3: When using SPRITE OSD, do not set $VSi = "00016"$, $VSi \geq "40016"$.

4: When displaying with SPRITE fonts 1 and 2 overlapped, the SPRITE font with a larger set value as the vertical display start position is displayed. When the set values of the vertical display start position are the same, the SPRITE font 1 is displayed.

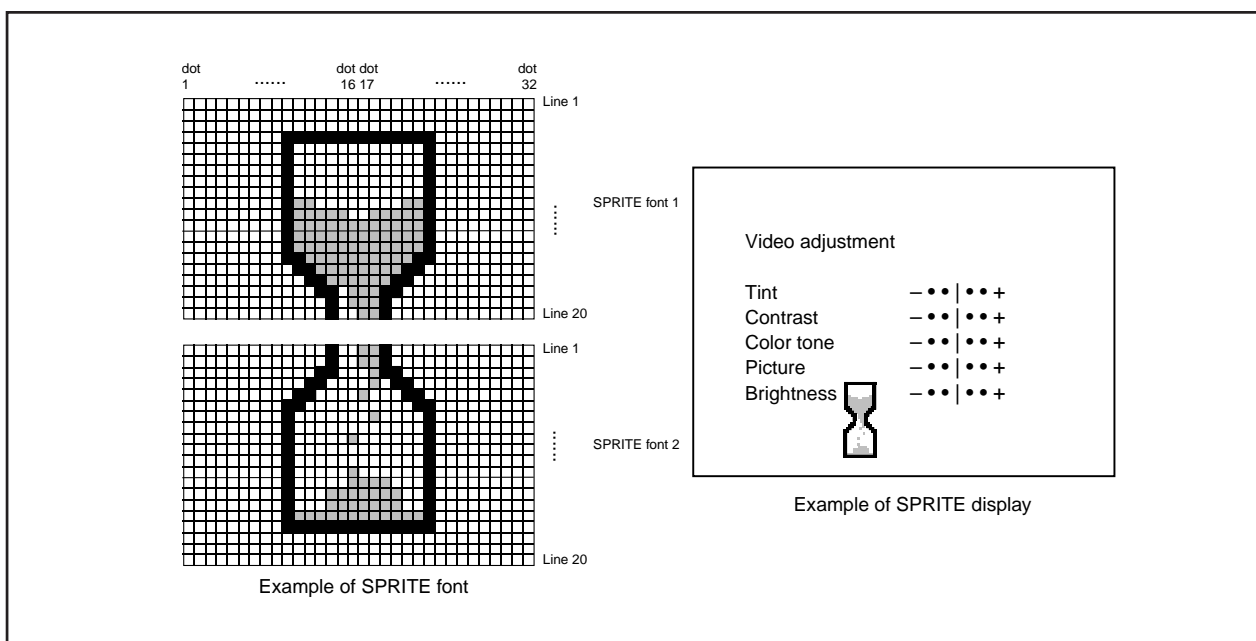


Figure 16.40 SPRITE OSD display example

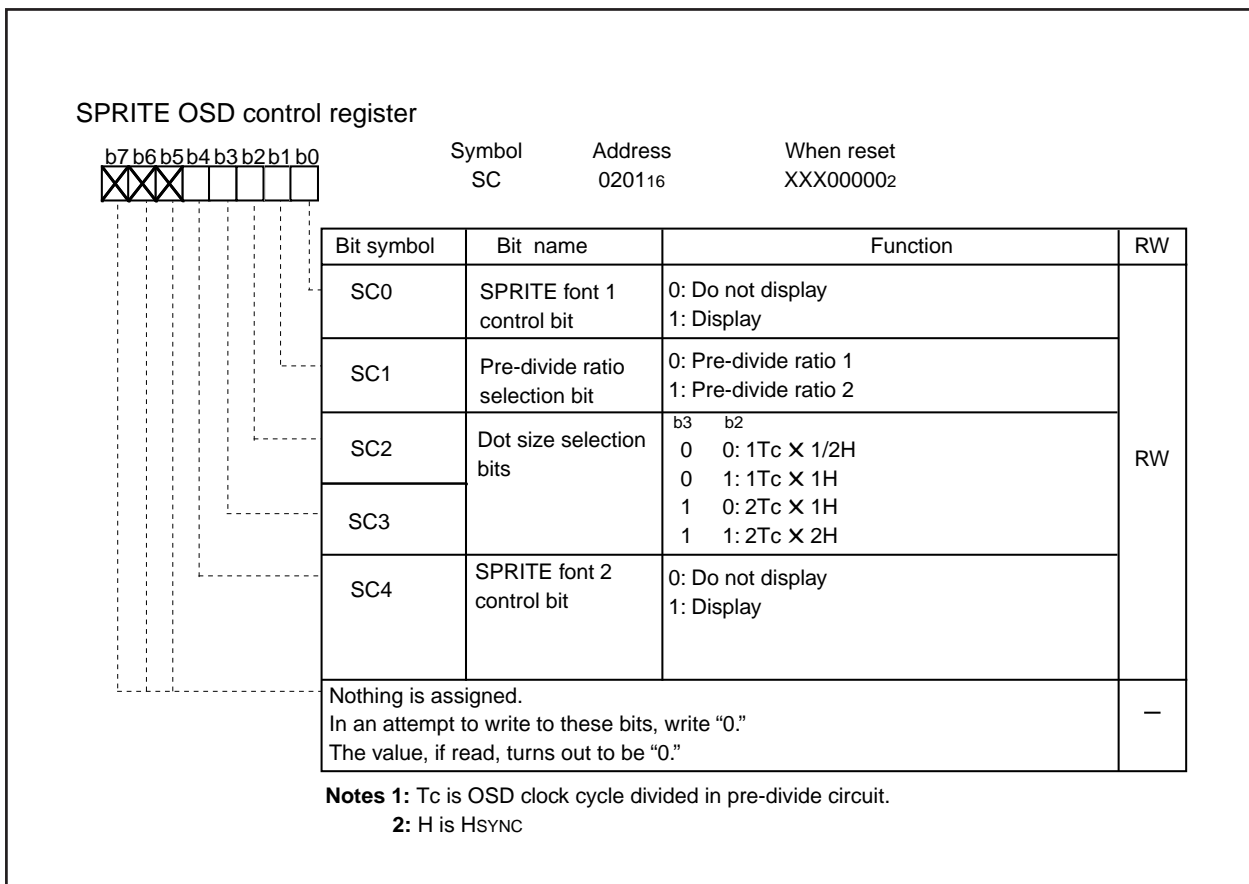


Figure 16.41 SPRITE OSD control register

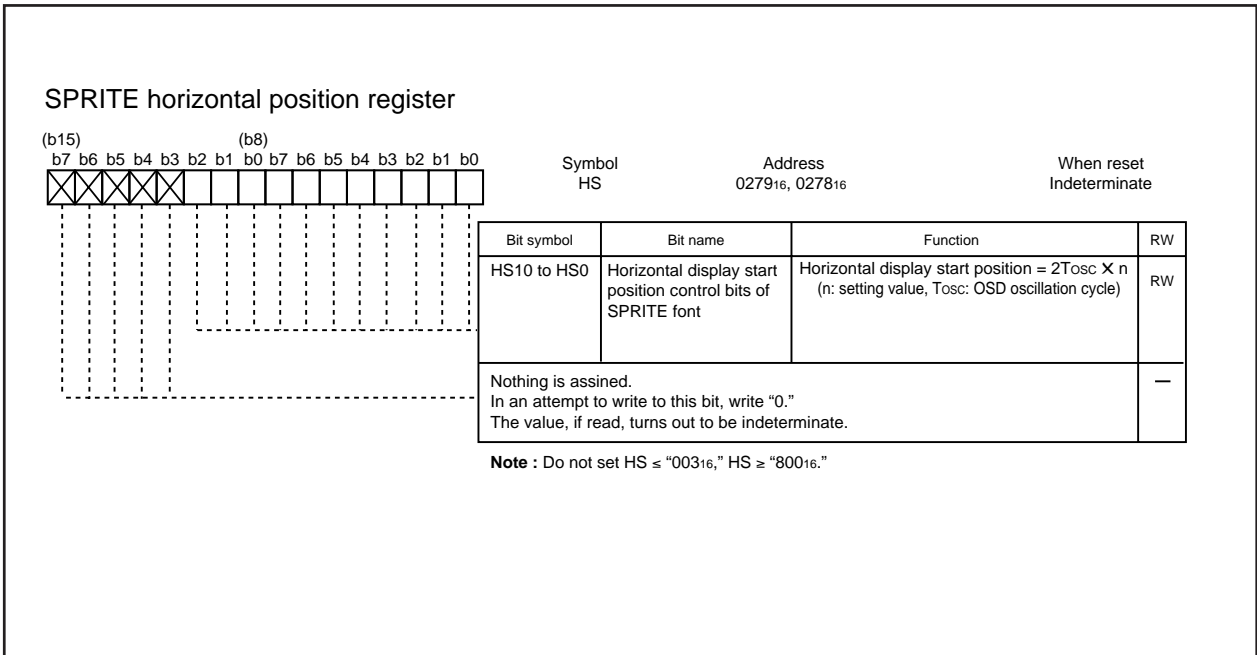


Figure 16.42 SPRITE horizontal position register

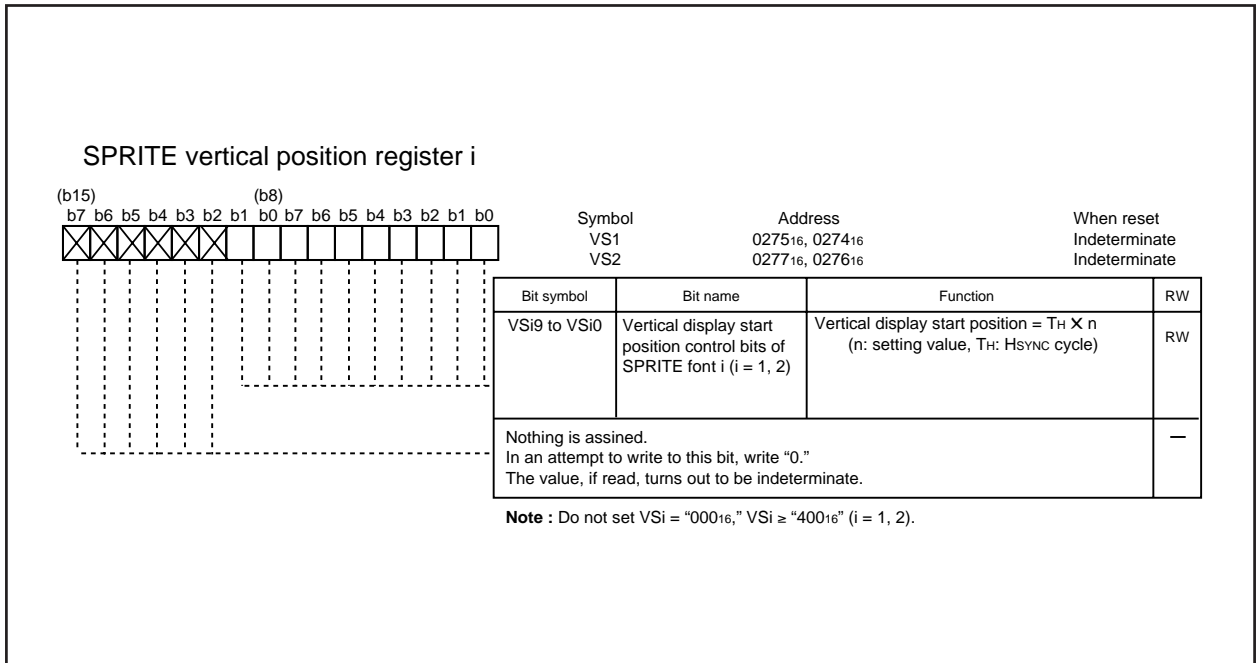


Figure 16.43 SPRITE vertical position register i (i = 1, 2)

Window Function

The window function can be set windows on-screen and output OSD within only the area where the window is set.

The ON/OFF for vertical window function is performed by bit 5 of the OSD control register 1 and is used to select vertical window function or vertical blank function by bit 6 of the OSD control register 2. Accordingly, the vertical window function cannot be used simultaneously with the vertical blank function. The display mode to validate the window function is selected by bits 5 to 7 of the OSD control register 3. The top border is set by the top border control register (TBR) and the bottom border is set by the bottom border control register (BBR).

The ON/OFF for horizontal window function is performed by bit 4 of the OSD control register 2 and is used interchangeably for the horizontal blank function with bit 5 of the OSD control register 2. Accordingly, the horizontal blank function cannot be used simultaneously with the horizontal window function. The display mode to validate the window function is selected by bits 5 to 7 of the OSD control register 3. The left border is set by the left border control register (LBR), and the right border is set by the right border control register (RBR).

Notes 1: Horizontal blank and horizontal window, as well as vertical blank and vertical window can not be used simultaneously.

2: When the window function is ON by OSD control registers 1 and 2, the window function of OUT2 is valid in all display mode regardless of setting value of the OSD control register 3 (bits 5 to 7). For example, even when make the window function valid in only CC mode, the function of OUT2 is valid in OSDS/L/P and CDOSD modes.

3: As for SPRITE display, the window function does not operate.

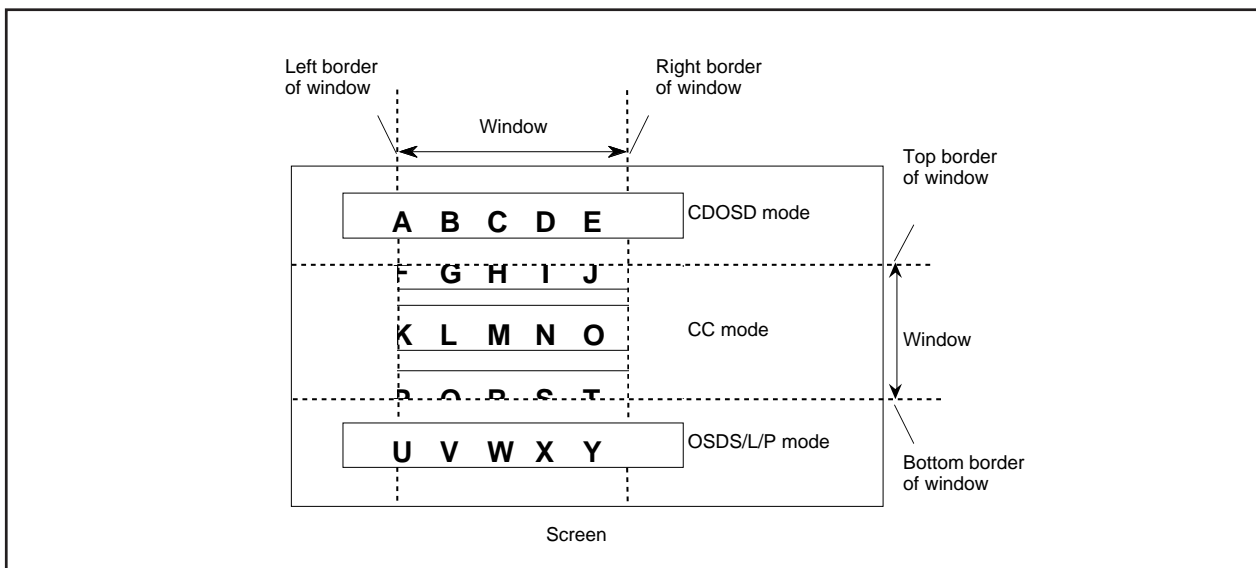


Figure 16.44 Example of window function (When CC mode is valid)

Blank Function

The blank function can output blank (OUT1) area on all sides (vertical and horizontal) of the screen. This provides the blank signal, wipe function, etc., when outputting a 3 : 4 image on a wide screen.

The ON/OFF for vertical blank function is performed by bit 5 of the OSD control register 1 and is used to select vertical window function or vertical blank function by bit 6 of the OSD control register 2. Accordingly, the vertical blank function cannot be used simultaneously with the vertical window function. The top border is set by the top border control register (TBR), and the bottom border is set by the bottom border control register (BBR), in 1H units.

The ON/OFF for horizontal blank function is performed by bit 4 of the OSD control register 2 and is used interchangeably for the horizontal window function with bit 5 of the OSD control register 2. Accordingly, the horizontal blank function cannot be used simultaneously with the horizontal window function. The left border is set by the left border control register (LBR) and the right border is set by the right border control register (RBR), in 4TOSC units.

The OSD output (except raster) in area with blank output is not deleted.

These blank signals are not output in the horizontal/vertical blanking interval.

- Notes 1.** Horizontal blank and horizontal window, as well as vertical blank and vertical window can not be used simultaneously.
- 2.** When using the window function, be sure to set "1" to bit 0 of OSD control register 1.

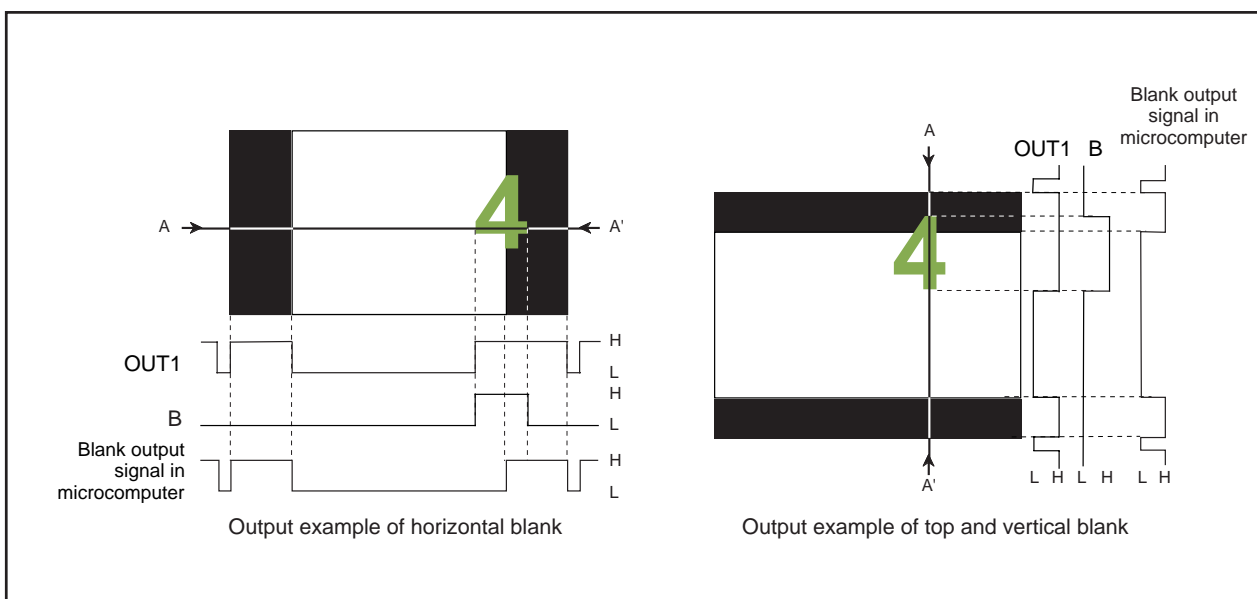


Figure 16.45 Blank output example (when OSD output is B + OUT1)

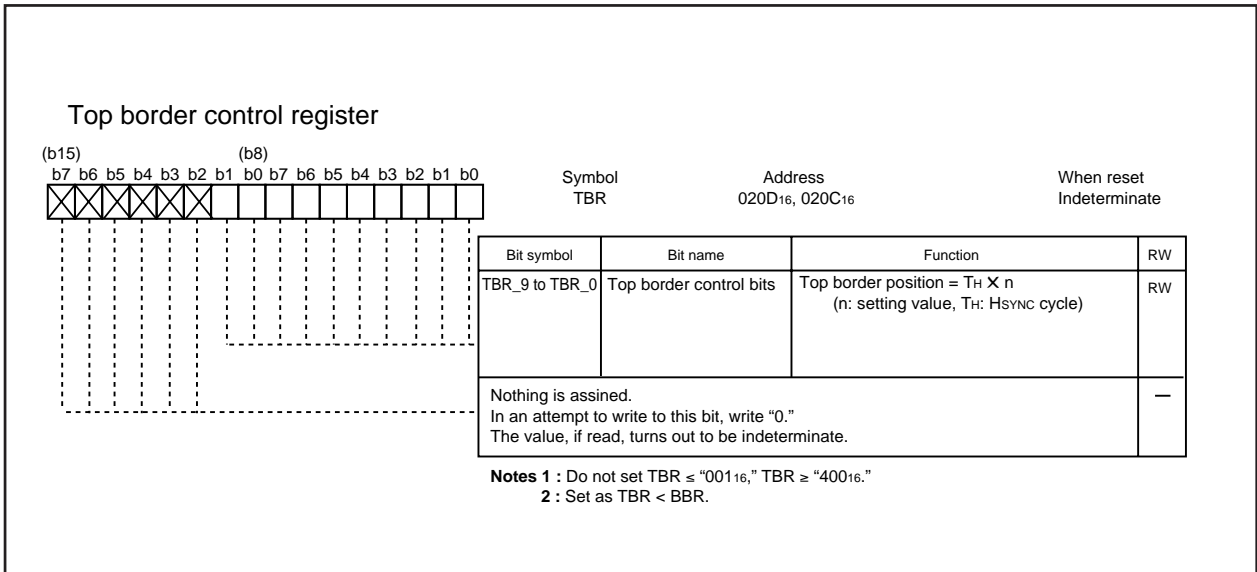


Figure 16.46 Top border control register

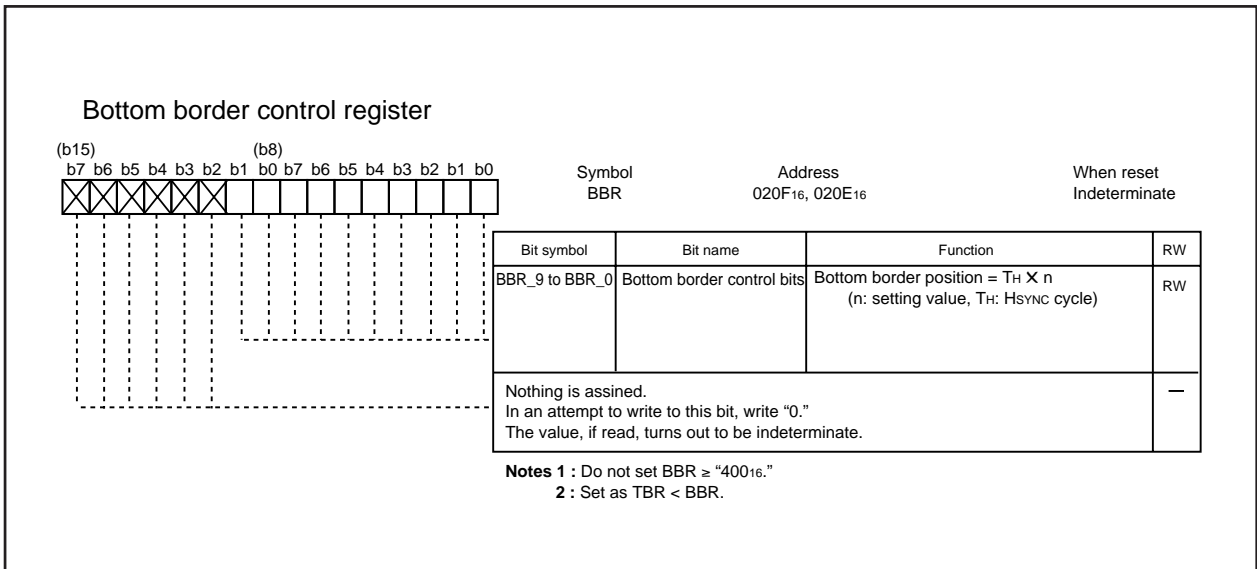


Figure 16.47 Bottom border control register

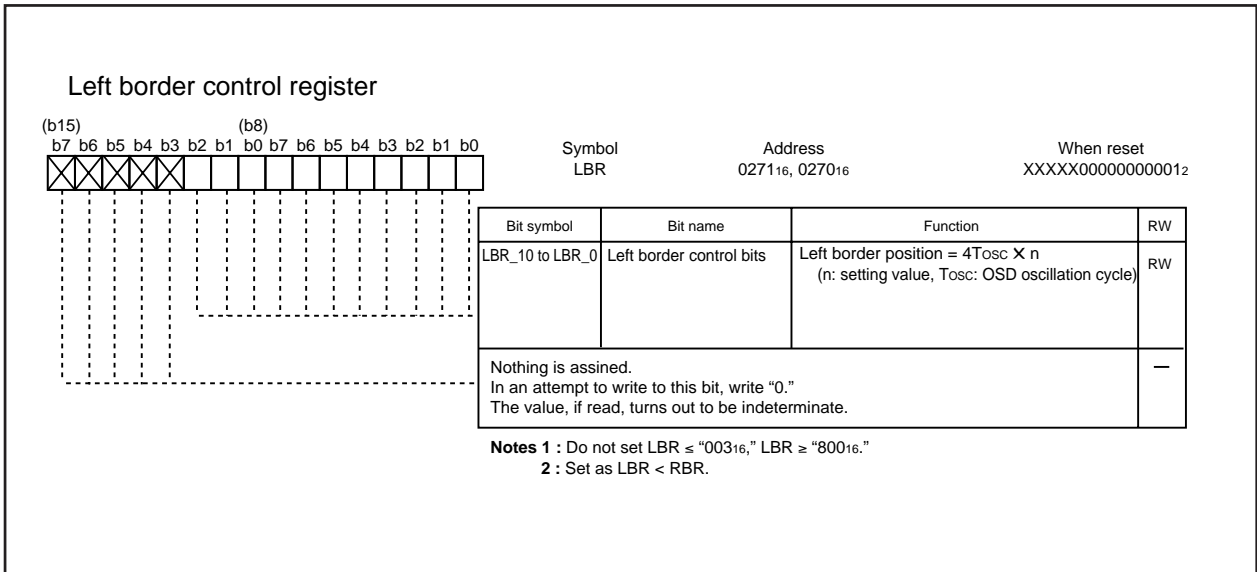


Figure 16.48 Left border control register

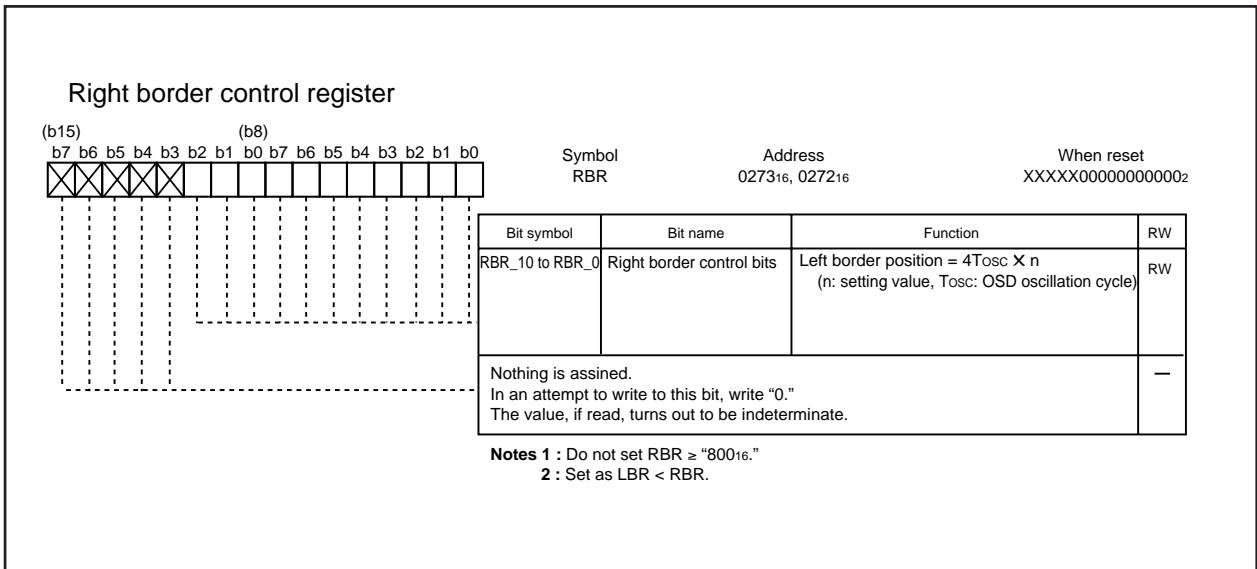


Figure 16.49 Right border control register

Raster Coloring Function

An entire screen (raster) can be colored by setting the bits 6 to 0 of the raster color register. Since each of the R, G, B, OUT1, and OUT2 pins can be switched to raster coloring output, 512 raster colors can be obtained.

When the character color/the character background color overlaps with the raster color, the color (R, G, B, OUT1, OUT2), specified for the character color/the character background color, takes priority of the raster color. This ensures that the character color/the character background color is not mixed with the raster color.

The raster color register is shown in Figure 16.50, the example of raster coloring is shown in Figure 16.51.

Note: Raster is not output to the area which includes blank area.

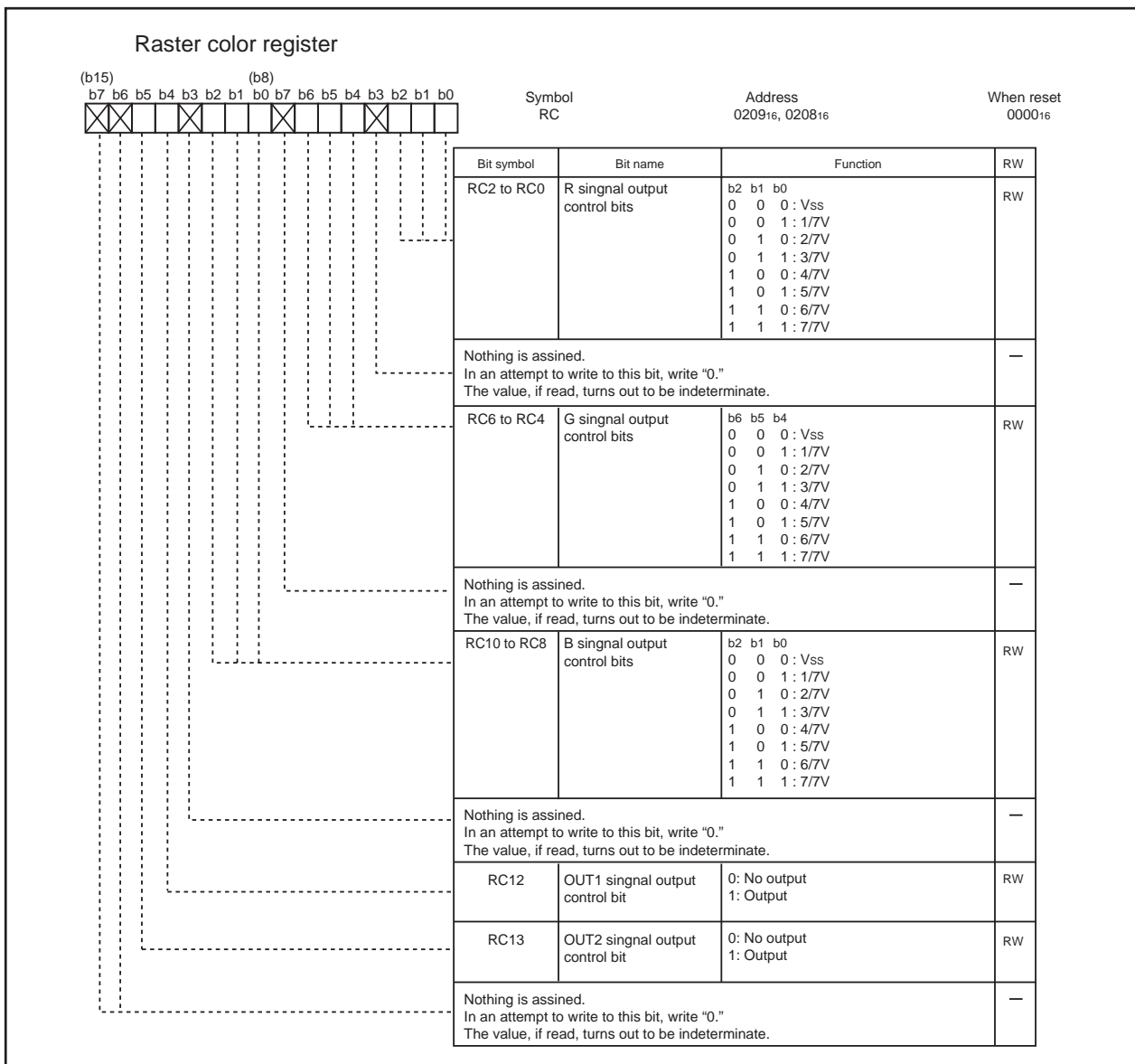


Figure 16.50 Raster color register

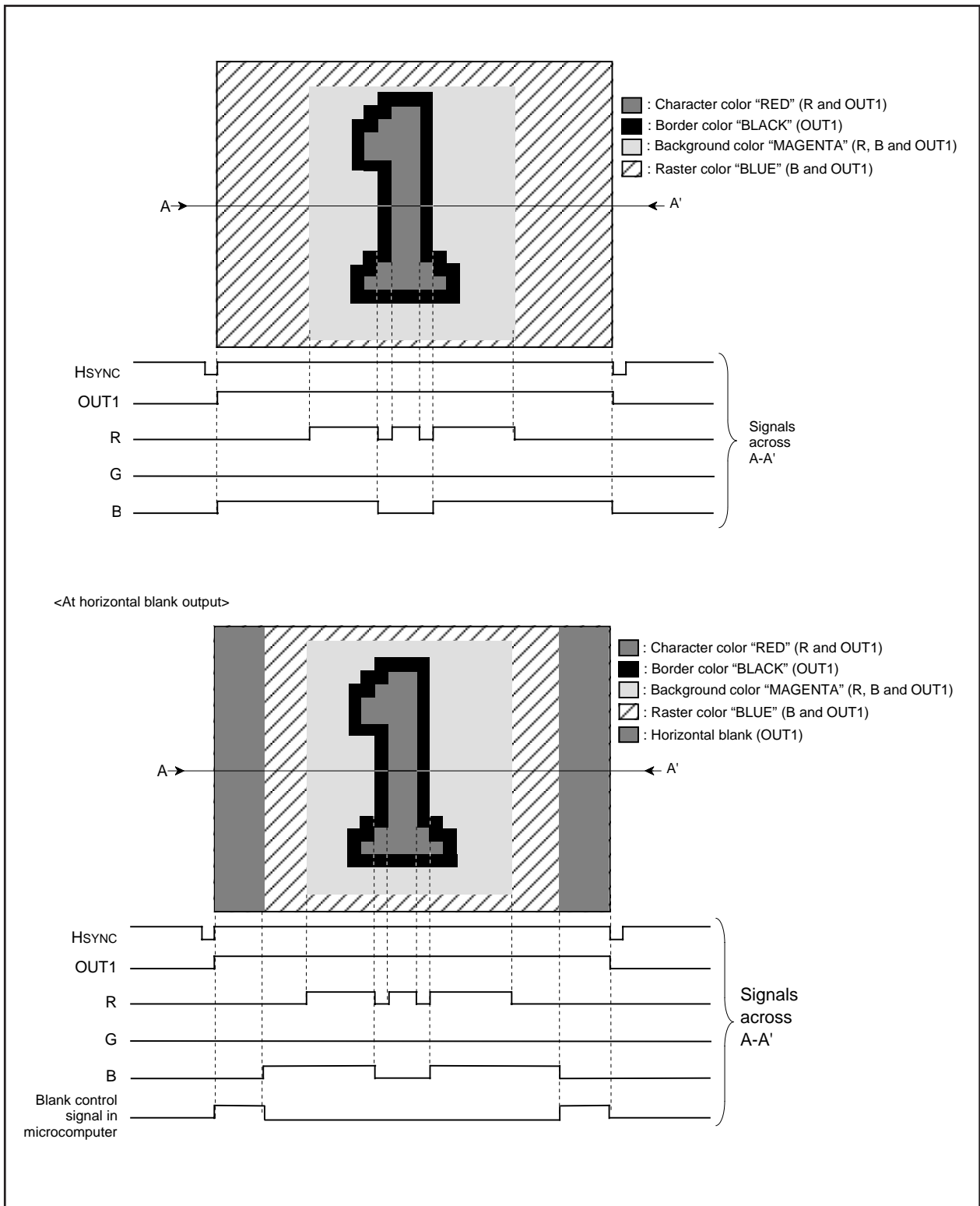


Figure 16.51 Example of raster coloring

Scan Mode

This microcomputer has the bi-scan mode for corresponding to HSYNC of double speed frequency. In the bi-scan mode, the vertical start display position and the vertical size is two times as compared with the normal scan mode. The scan mode is selected by bit 1 of the OSD control register 1 (refer to Figure 16.3).

Table 16.12 Setting for scan mode

Parameter	Scan Mode	Normal Scan	Bi-Scan
Bit 1 of OSD control register 1		0	1
Vertical display start position		Value of vertical position register X 1H	Value of vertical position register X 2H
Vertical dot size		1Tc X 1/2H	1Tc X 1H
		1Tc X 1H	1Tc X 2H
		2Tc X 2H	2Tc X 4H
		3Tc X 3H	3Tc X 6H

R, G, B Signal Output Control

The form of R, G, B signal output is controlled by bit 4 of the clock register and bit 2 of the OSD control register 2 as the table below.

Table 16.13 R, G, B signal output control

Bit 4 of clock control register 1 (address 0205 ₁₆)	Bit 2 of OSD control register 2 (address 0203 ₁₆)	Bit 0 of extended register (address 02D5 ₁₆)	Form of R, G, B signal output
0	0	1	Each R, G, B pin outputs 2 values (digital output).
	1	0	Each R, G, B pin outputs 8 values (analog output). (Note 1)
1	0	0	DIGR0, DIGR1, DIGR2 DIGG0, DIGG1, DIGG2 DIGB0, DIGB1, DIGB2 Each of these pins output two-level values. (Corresponding to each signal output control bit in color palette register i) DIGR0~2 correspond to CRi0~2, respectively. DIGG0~2 correspond to CRi4~6, respectively. DIGB0~2 correspond to CRi8~10, respectively.

Note 1: In addition to this, set the ANARGBCLKEN bit (address 02DE₁₆, bit 4) and the RGBRON bit (address 025D₁₆, bit 6) to "1." Also, set the ANARGBCAPON bit (address 2DE, bit3) and attach the capacitor for analog RGB internal operation stability (0.47 μ F (reference value)) to the CAP pin.

2: To use the OUT1 and OUT2 pin, set the OUT1EN bit (address 02DB₁₆, bit 4) and the OUT2EN bit (address 02DB₁₆, bit 5) to "1."

OSD Reserved Register

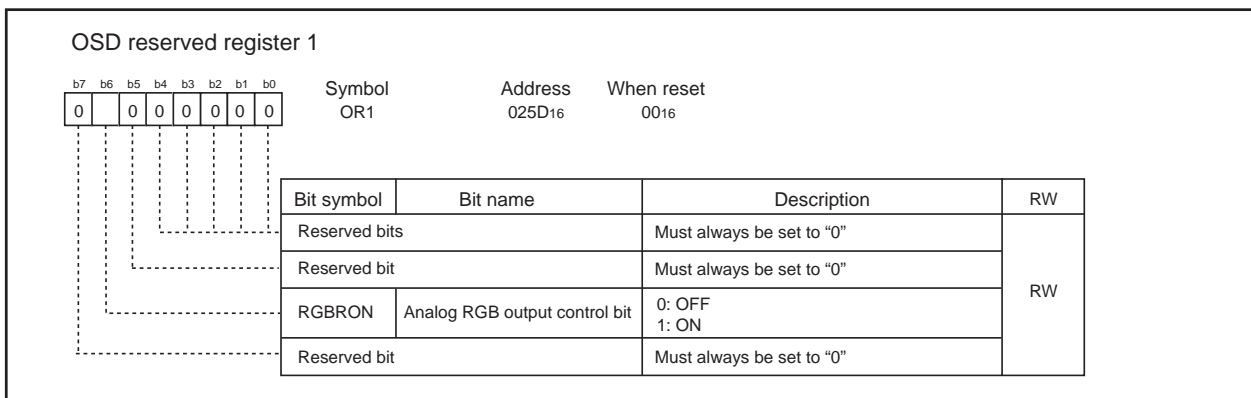


Figure 16.52 OSD reserved register 1

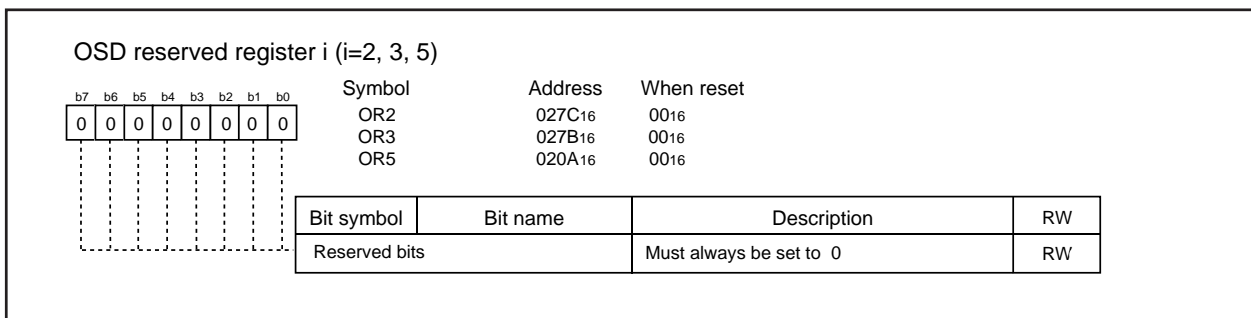


Figure 16.53 OSD reserved register i (i=2, 3, 5)

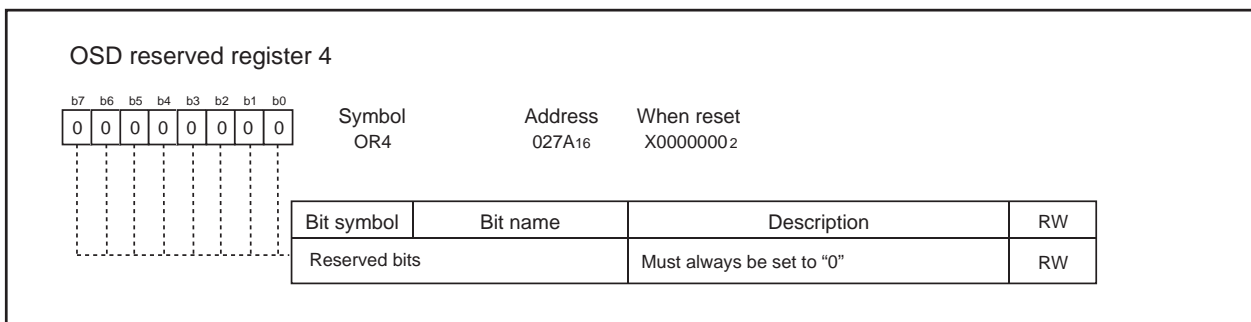


Figure 16.54 OSD reserved register 4

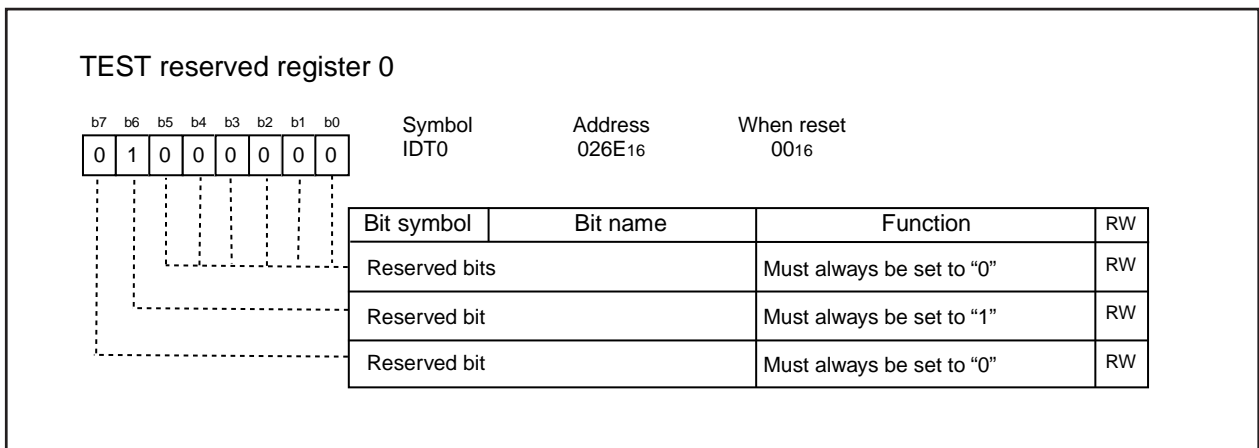


Figure 16.55 TEST reserved register 0

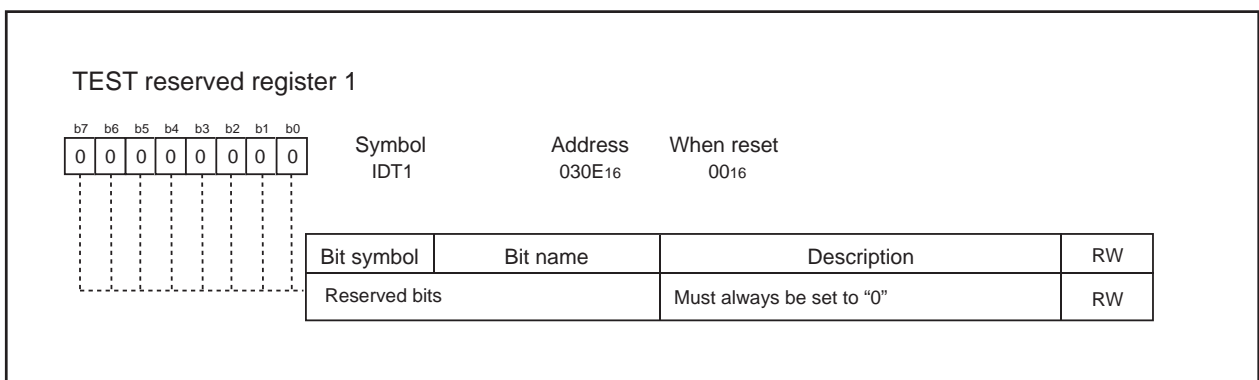


Figure 16.56 TEST reserved register 1

TB0IN noise filter

The noise filter is built in the input of a TB0IN pin. ON/OFF of a noise filter and selection of a filter clock are performed in the bit 2 to bit 4 of extended register 1D.

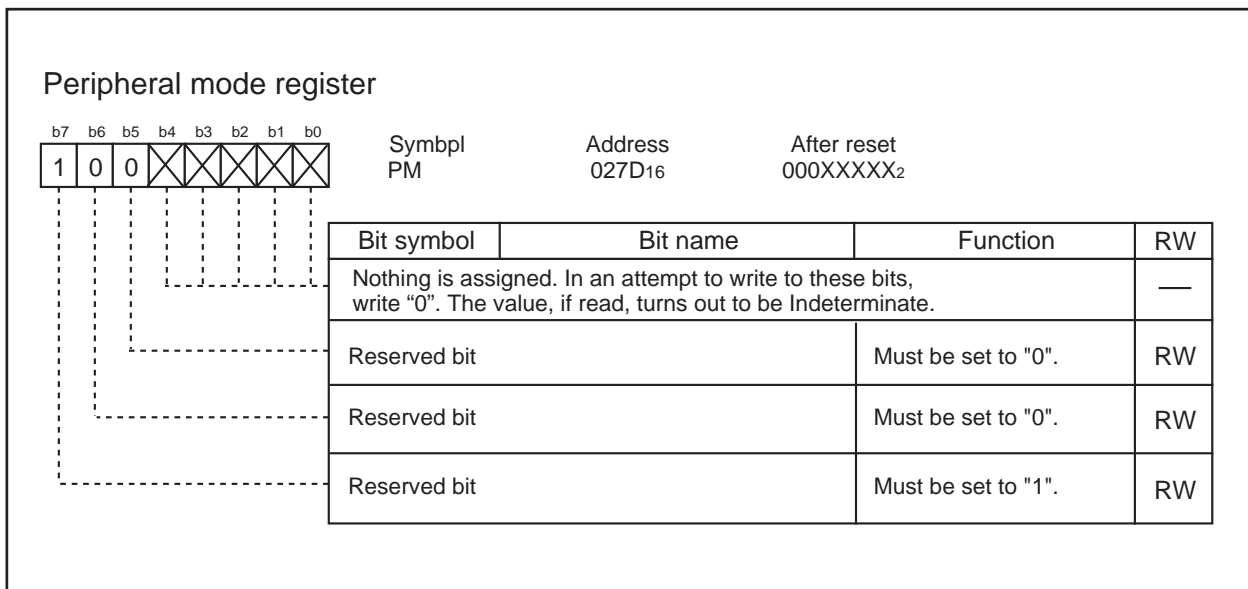


Figure 16.57 Peripheral mode register

Programmable I/O Ports

The programmable input/output ports (hereafter referred to simply as “I/O ports”) consist of 75 lines P0 to P10. 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.

Figures 17.1 to 17.5 show the I/O ports. Figure 17.6 shows the I/O pins.

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

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

Figure 17.7 shows the PDi registers.

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

During memory extension and microprocessor modes, the PDi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, 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 10)

Figure 17.8 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 17.10 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 (PCR Register)

Figure 17.11 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.

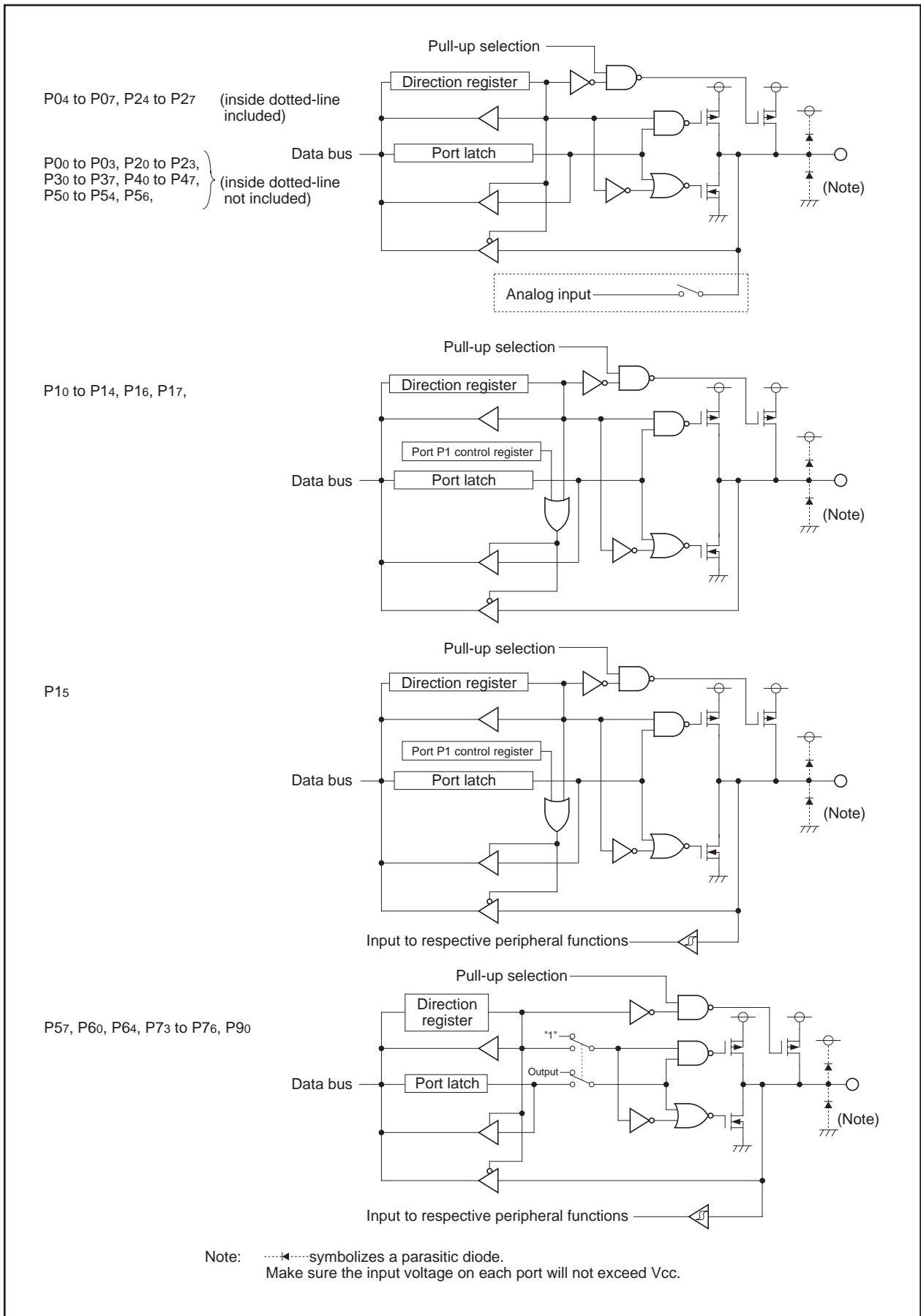


Figure 17.1. I/O Ports (1)

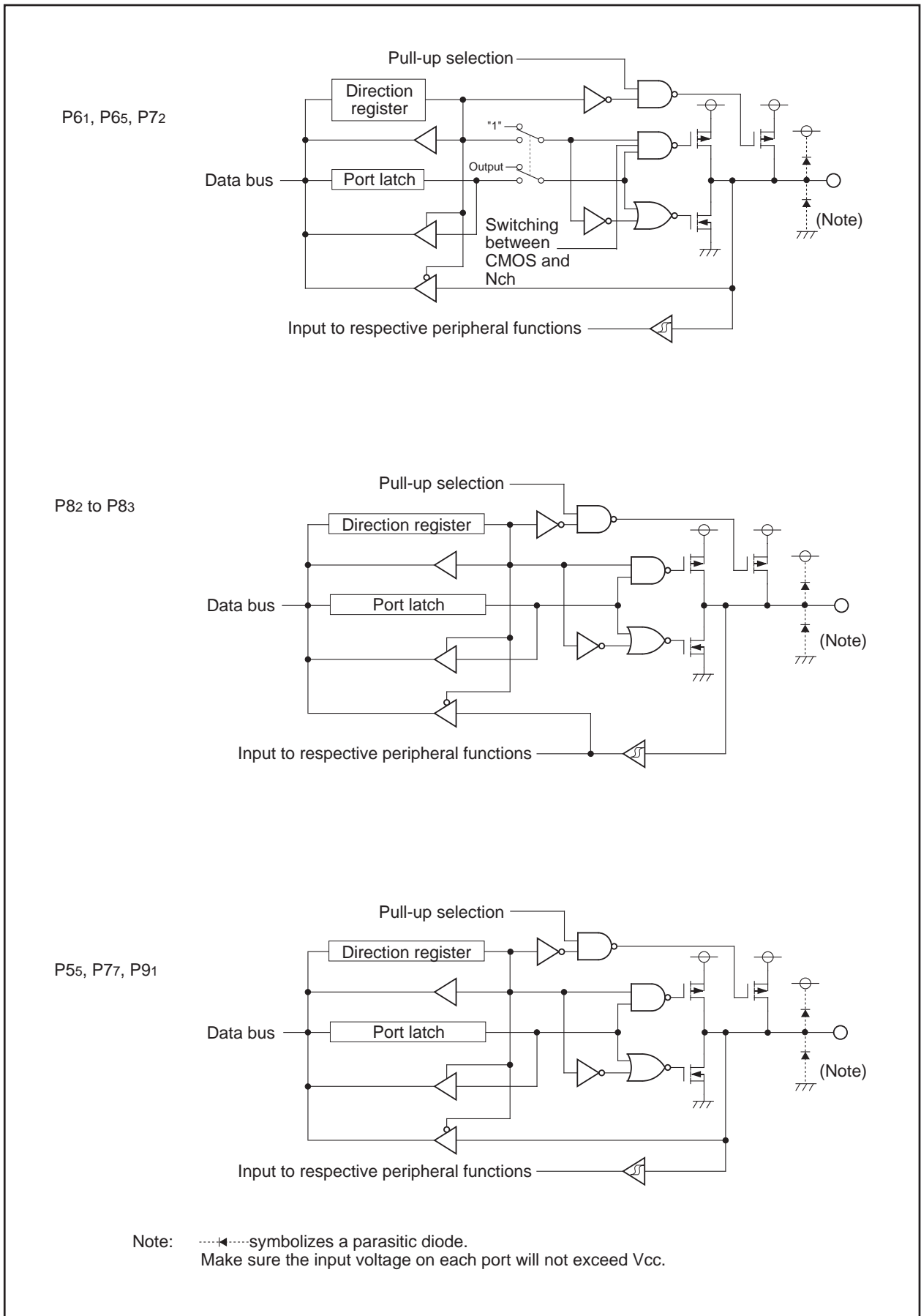


Figure 17.2. I/O Ports (2)

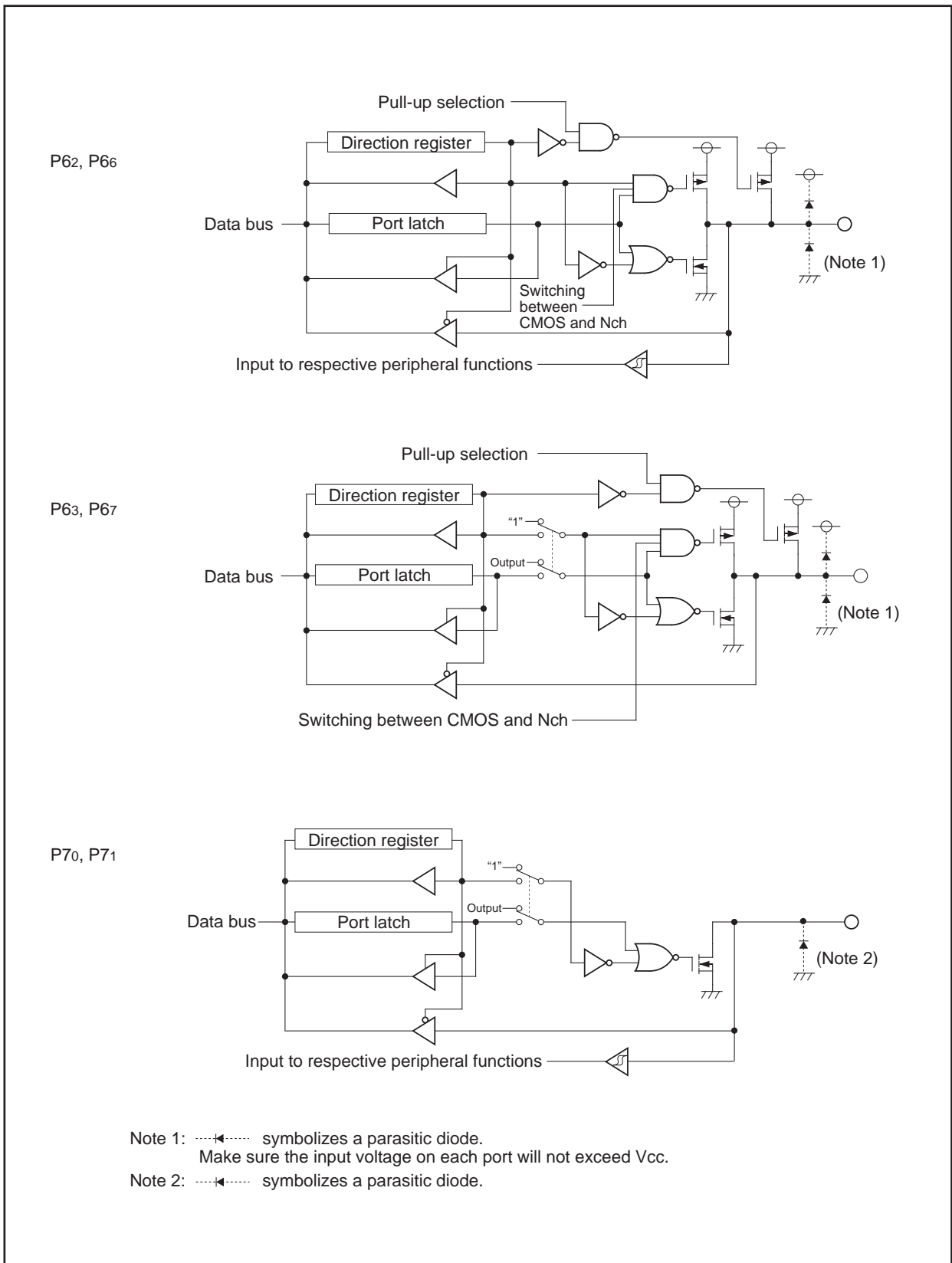


Figure 17.3. I/O Ports (3)

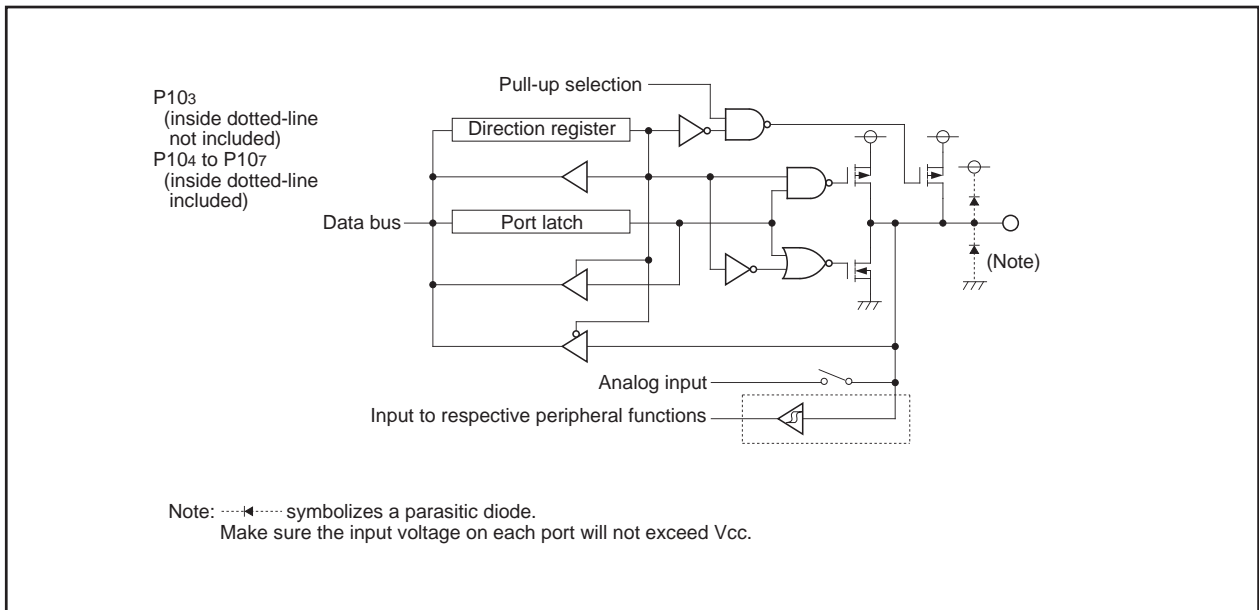


Figure 17.4. I/O Ports (4)

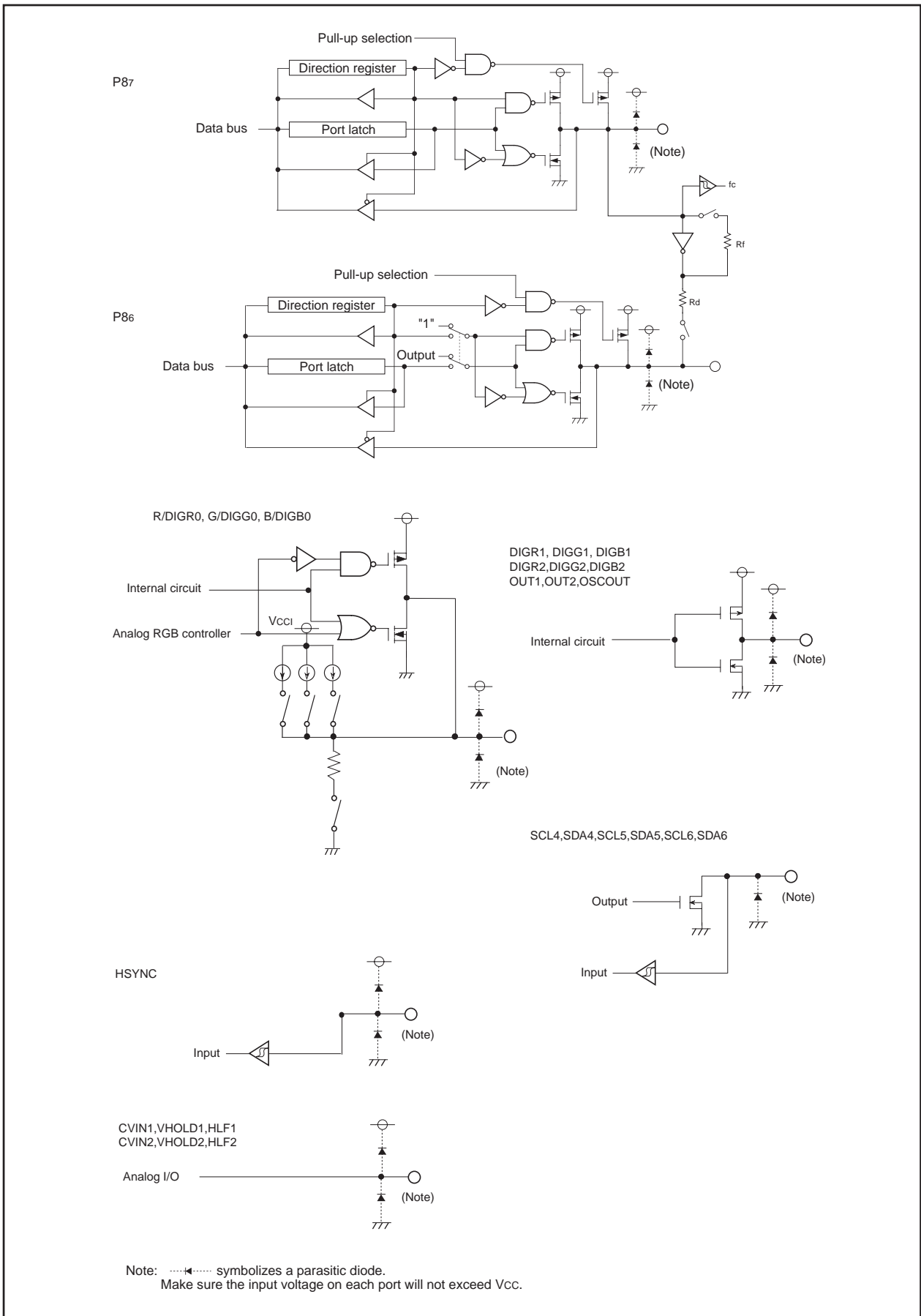


Figure 17.5. I/O Ports (5)

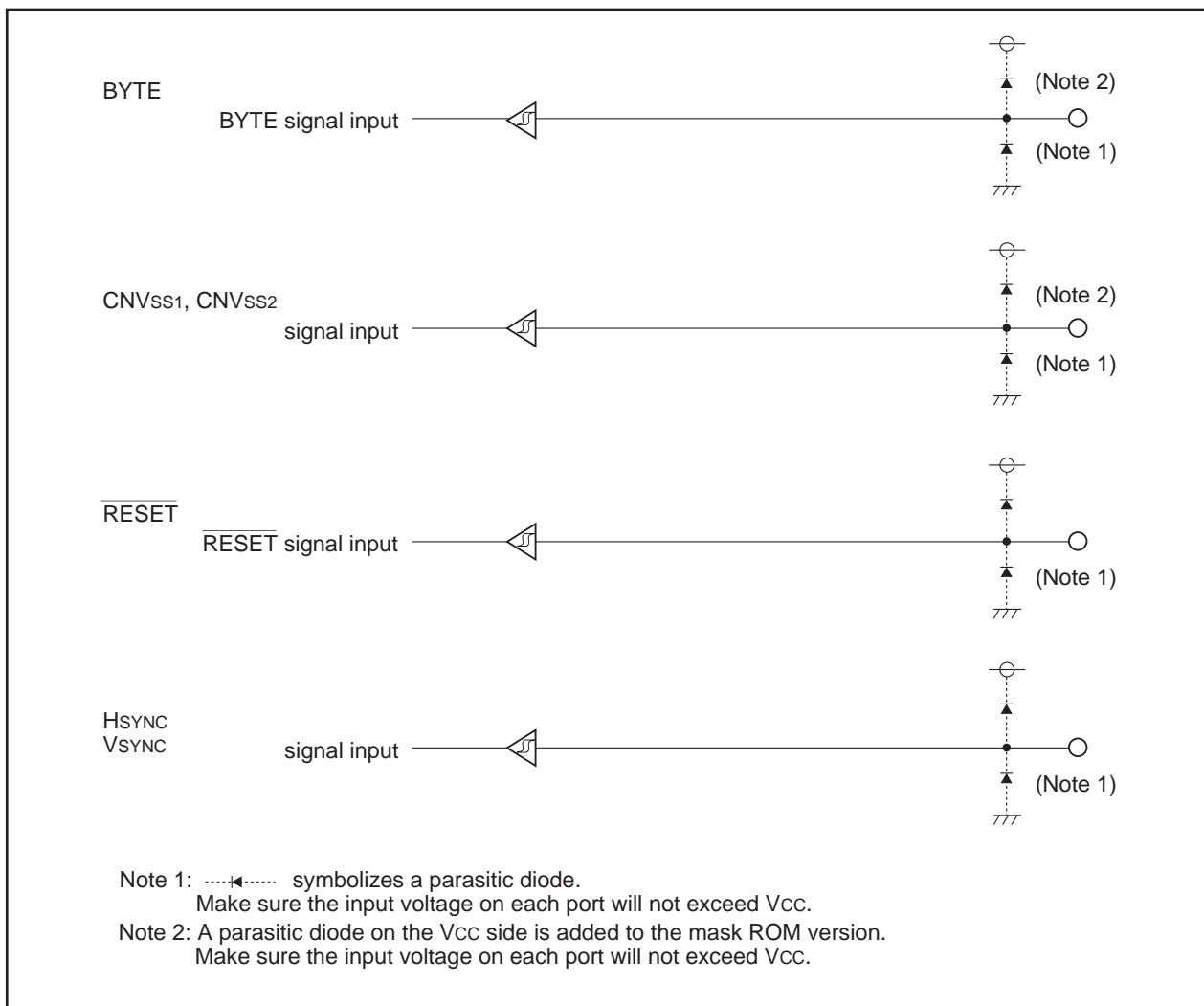


Figure 17.6. I/O Pins

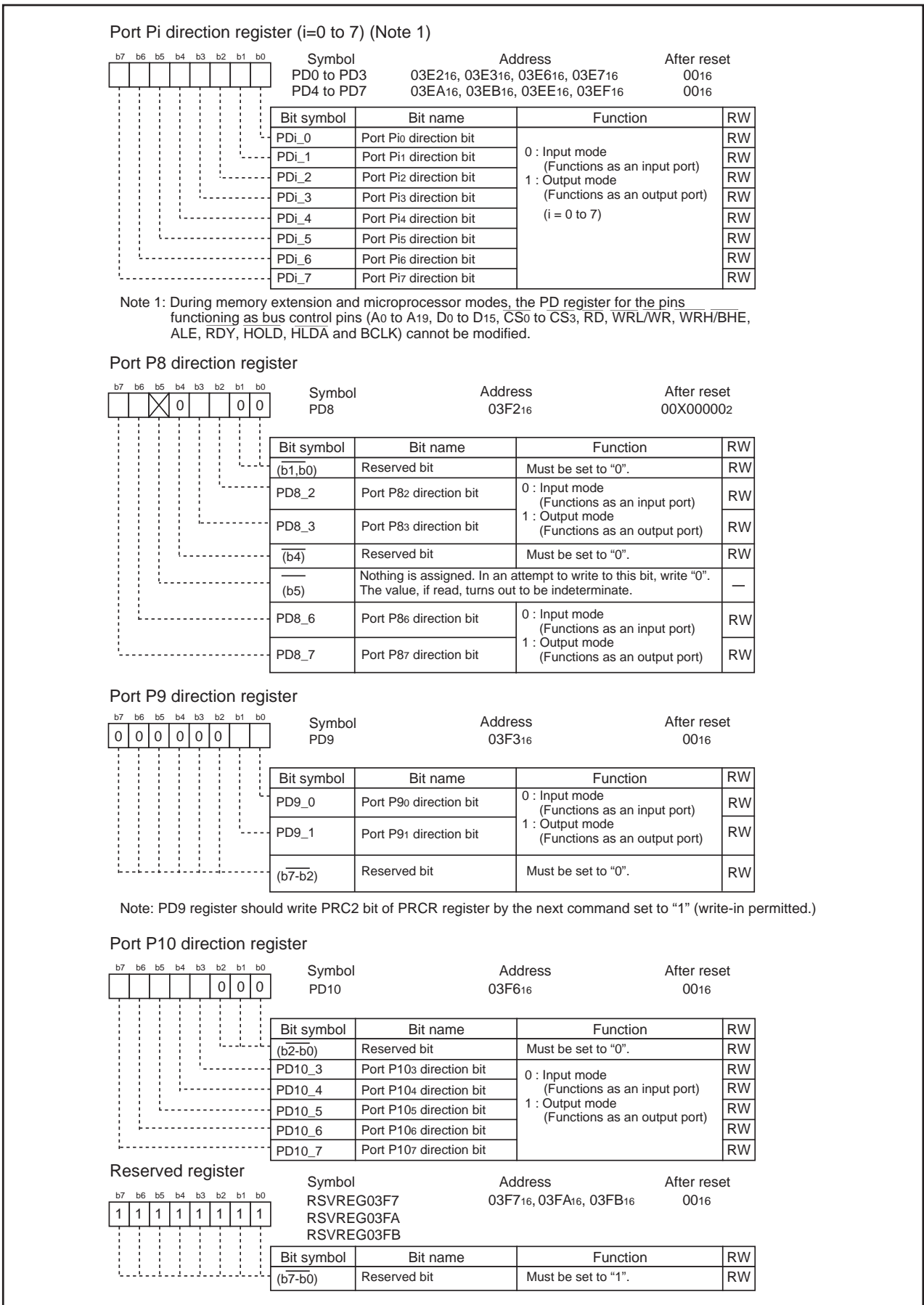


Figure 17.7. PD0 to PD10 Registers

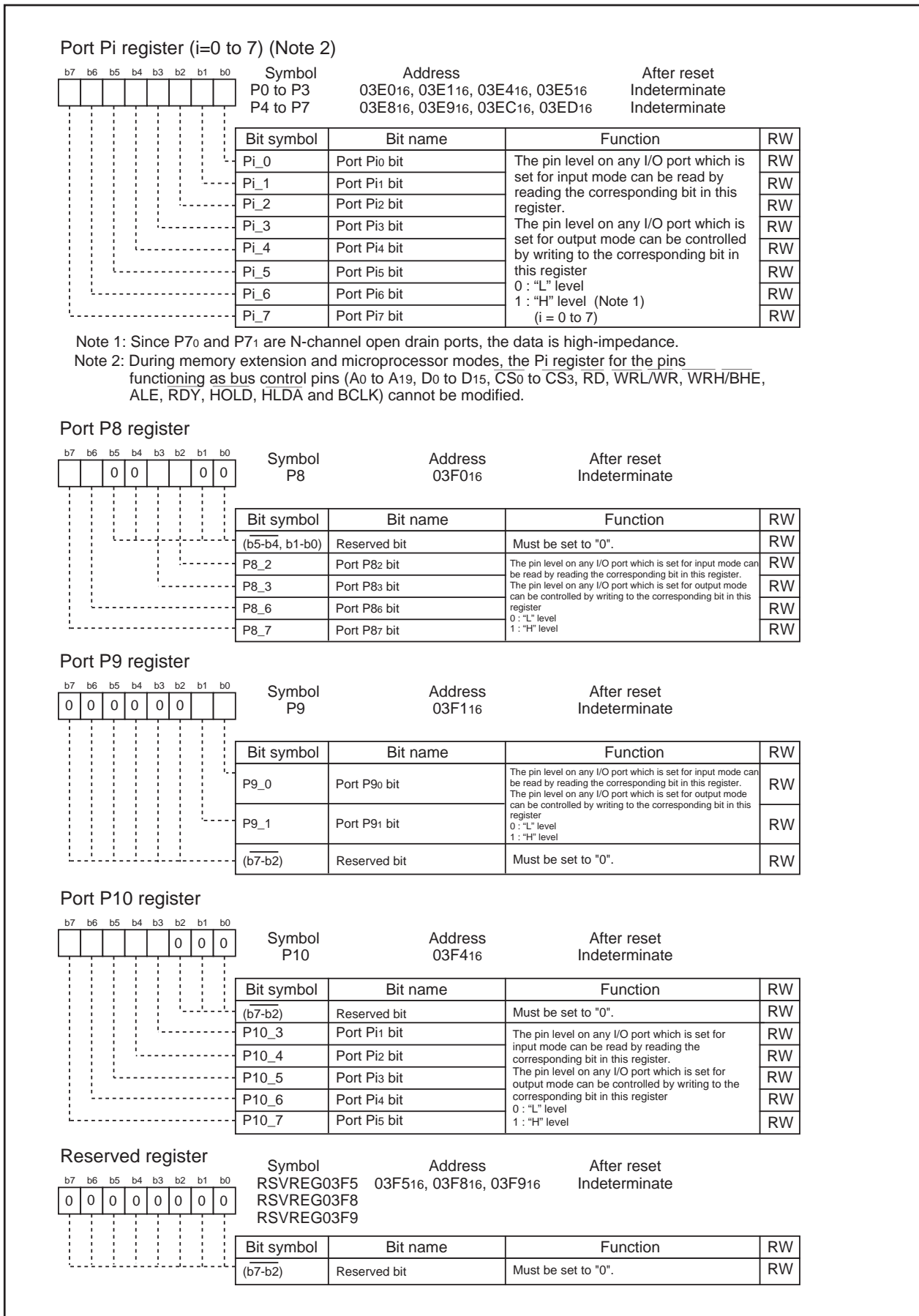


Figure 17.8. P0 to P10 Registers

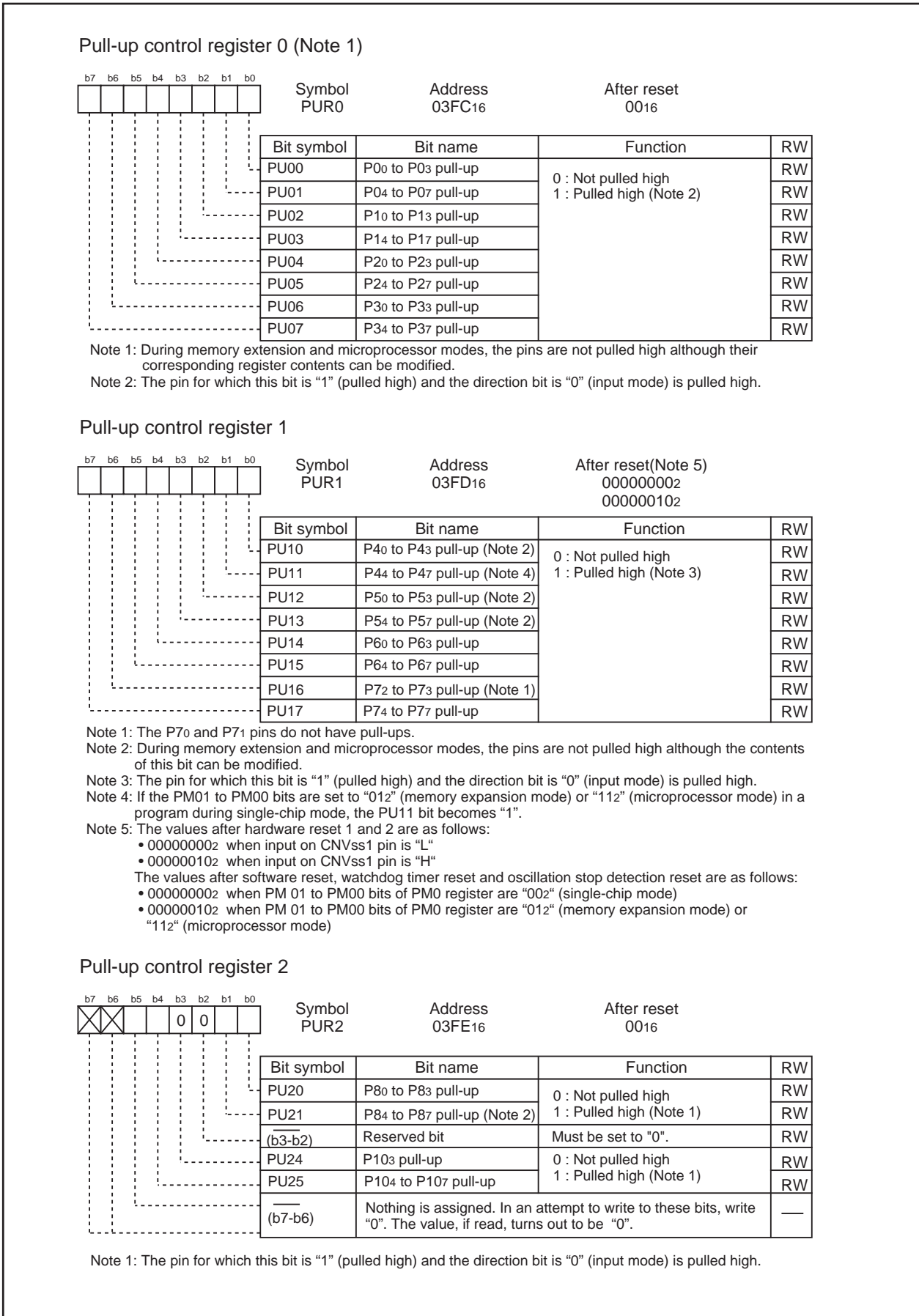


Figure 17.9. PUR0 to PUR2 Registers

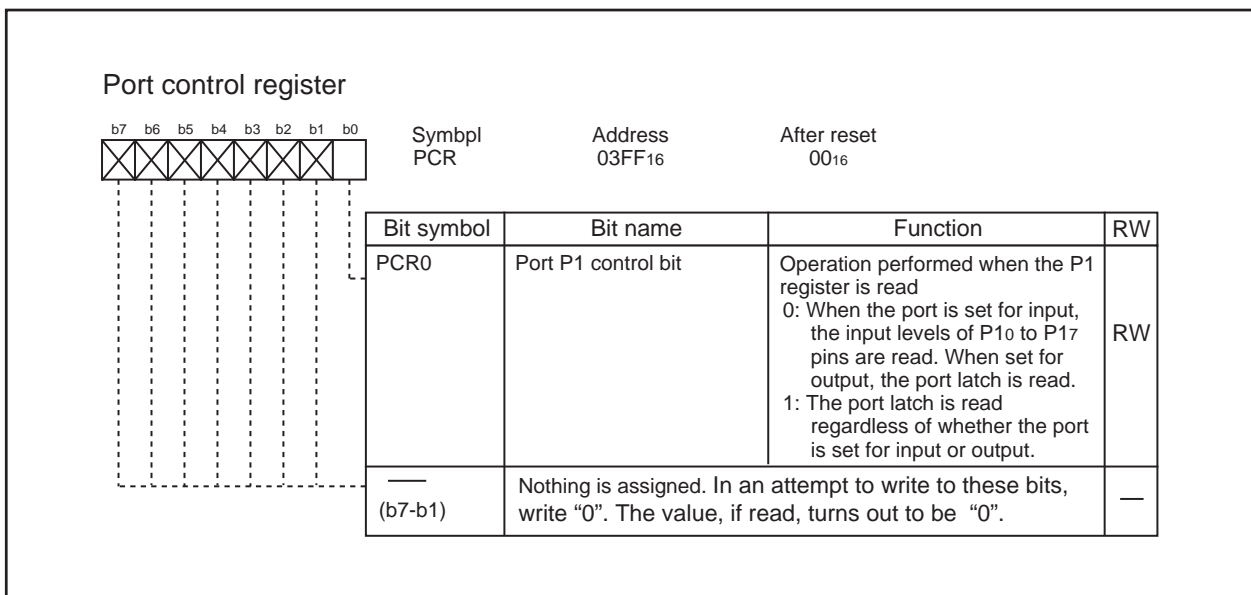


Figure 17.10. PCR Register

Table 17.1. Unassigned Pin Handling in Single-chip Mode

Pin name	Connection
Ports P0 to P10	After setting for input mode, connect every pin to VSS via a resistor(pull-down); or after setting for output mode, leave these pins open. (Note 1)
XOUT (Note 2)	Open
BYTE	Connect to VSS

Note 1: When the ports P70 and P71 are set for output mode, make sure a low-level signal is output from the pins. The ports P70 and P71 are N-channel open-drain outputs.

Note 2: With external clock input to XIN pin.

Table 17.2. Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode

Pin name	Connection
Ports P6 to P10	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open. (Note 1, 2)
P45 / CS1 to P47 / CS3	Connect to VCC via a resistor (pulled high) by setting the PD4 register's corresponding direction bit for CSi (i=1 to 3) to "0" (input mode) and the CSR register's CSi bit to "0" (chip select disabled).
BHE, ALE, HLDA, XOUT(Note 3), BCLK (Note 4)	Open
HOLD, RDY	Connect via resistor to VCC2 (pull-up)

Note 1: If the CNVSS1 pin has the VSS level applied to it, these pins are set for input ports until the processor mode is switched over in a program after reset. For this reason, the voltage levels on these pins become indeterminate, causing the power supply current to increase while they remain set for input ports.

Note 2: When the ports P70 and P71 are set for output mode, make sure a low-level signal is output from the pins. The ports P70 and P71 are N-channel open-drain outputs.

Note 3: With external clock input to XIN pin.

Note 4: If the PM07 bit in the PM0 register is set to "1" (BCLK output is not carried out), connect this pin to Vcc via a resistor (pulled high).

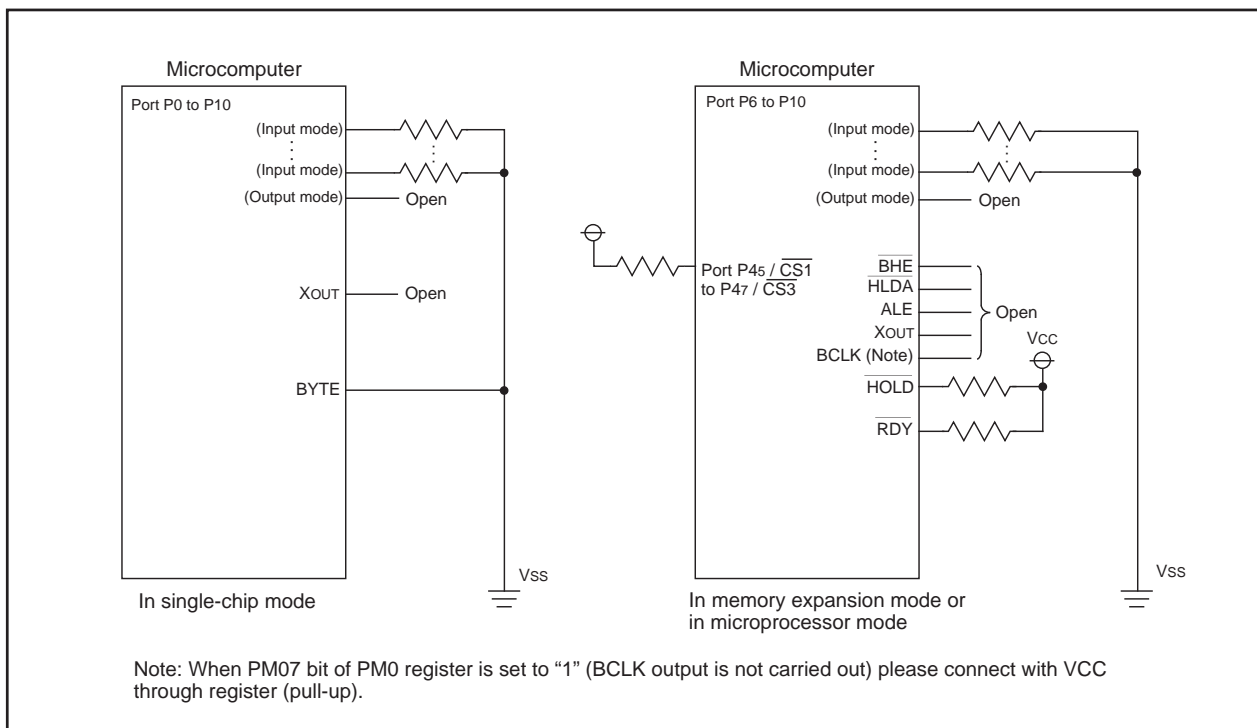
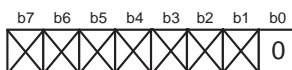


Figure 17.11. Unassigned Pins Handling

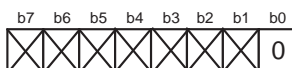
Reserved register



Symbpl RSVREG026F Address 026F₁₆ After reset XXXXXXX0₂

Bit symbol	Bit name	Function	RW
$\overline{\text{(b0)}}$	Reserved bit	Must be set to "0".	RW
$\overline{\text{(b7-b1)}}$	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

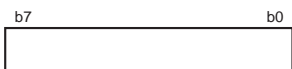
Reserved register



Symbpl RSVREG030F Address 030F₁₆ After reset XXXXXXX0₂

Bit symbol	Bit name	Function	RW
$\overline{\text{(b0)}}$	Reserved bit	Must be set to "0".	RW
$\overline{\text{(b7-b1)}}$	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

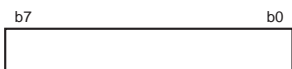
Reserved register



Symbpl RSVREG0342 Address 0342₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

Reserved register



Symbpl RSVREG0343 Address 0343₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

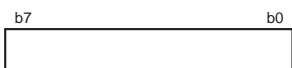
Reserved register



Symbpl RSVREG0344 Address 0344₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

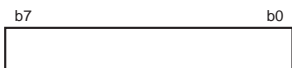
Reserved register



Symbpl RSVREG0345 Address 0345₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

Reserved register



Symbpl RSVREG0346 Address 0346₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

Reserved register



Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

Reserved register



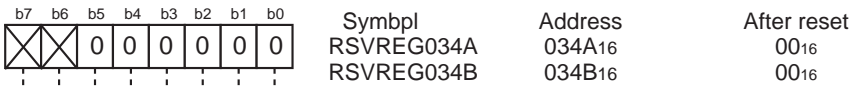
Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Must be set to "0".	RW

Reserved register



Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Must be set to "0".	RW

Reserved register



Bit symbol	Bit name	Function	RW
(b5-b0)	Reserved bit	Must be set to "0".	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".		—

Reserved register



Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

Reserved register


 Symbpl RSVREG03BC Address 03BC₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

Reserved register


 Symbpl RSVREG03BD Address 03BD₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

Reserved register


 Symbpl RSVREG03BE Address 03BE₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	WO

Reserved register


 Symbpl RSVREG03C0 Address 03C0₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	RO

Reserved register


 Symbpl RSVREG03C1 Address 03C1₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	RO

Reserved register

b7 b0 Symbpl Address After reset
 RSVREG03C2 03C2₁₆ Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	RO

Reserved register

b7 b0 Symbpl Address After reset
 RSVREG03C3 03C3₁₆ Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	RO

Reserved register

b7 b0 Symbpl Address After reset
 RSVREG03C4 03C4₁₆ Indeterminate

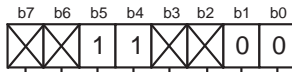
Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	RO

Reserved register

b7 b0 Symbpl Address After reset
 RSVREG03C5 03C5₁₆ Indeterminate

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Setup is arbitrary.	RO

Reserved register



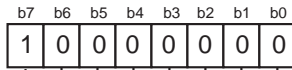
Symbpl
RSVREG03DE

Address
03DE₁₆

After reset
XX00XXXX₂

Bit symbol	Bit name	Function	RW
$\overline{(b1-b0)}$	Reserved bits	Must be set to "0".	RW
$\overline{(b3-b2)}$	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.		—
$\overline{(b5-b4)}$	Reserved bits	Must be set to "1".	RW
$\overline{(b7-b6)}$	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.		—

Reserved register



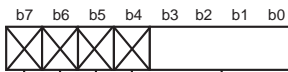
Symbpl
RSVREG03DF

Address
03DF₁₆

After reset
00₁₆

Bit symbol	Bit name	Function	RW
$\overline{(b6-b0)}$	Reserved bits	Must be set to "0".	RW
$\overline{(b7)}$	Reserved bit	Must be set to "1".	RW

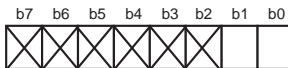
Reserved register



Symbpl RSVREG034D Address 034D₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b3-b0)	Reserved bit	Setup is arbitrary.	WO
(b7-b4)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be Indeterminate		—

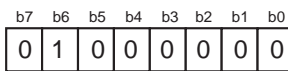
Reserved register



Symbpl RSVREG039E Address 039E₁₆ After reset XXXXXX00₂

Bit symbol	Bit name	Function	RW
(b1-b0)	Reserved bit	Must be set to "0".	RW
(b7-b2)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

Reserved register



Symbpl RSVREG0362 Address 0362₁₆ After reset 01000000₂
RSVREG0366 Address 0366₁₆ After reset 01000000₂

Bit symbol	Bit name	Function	RW
(b5-b0)	Reserved bit	Must be set to "0".	RW
(b6)	Reserved bit	Must be set to "1".	RW
(b7)	Reserved bit	Must be set to "0".	RW

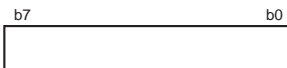
Reserved register



Symbpl RSVREG0363 Address 0363₁₆ After reset Indeterminate
RSVREG0367 Address 0367₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b0)	Reserved bit	Setup is arbitrary.	WO

Reserved register



Symbpl RSVREG0360 Address 0360₁₆ After reset Indeterminate
RSVREG0364 Address 0364₁₆ After reset Indeterminate

Bit symbol	Bit name	Function	RW
(b0)	Reserved bit	Setup is arbitrary.	RW

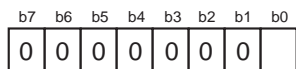
Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02C0)	Address (02C0 ₁₆)	After reset (00 ₁₆)	
								Bit symbol	Bit	Function	R
								IREQSELSIG0	Interrupt DMA factor selection	0 : TIMERA0 1 : I ² C-bus0	○:○
								IREQSELSIG1	Interrupt DMA factor selection	0 : TIMERA1 1 : I ² C-bus1	○:○
								IREQSELSIG2	Interrupt DMA factor selection	0 : TIMERA2 1 : OSD2	○:○
								IREQSELSIG3	Interrupt DMA factor selection	0 : TIMERA3 1 : VSYNC	○:○
								IREQSELSIG4	Interrupt DMA factor selection	0 : TIMERA4 1 : I ² C-bus0NACK	○:○
								IREQSELSIG5	Interrupt DMA factor selection	0 : TIMERB0 1 : I ² C-bus1NACK	○:○
								IREQSELSIG6	Interrupt DMA factor selection	0 : TIMERB1 1 : I ² C-bus2NACK	○:○
								IREQSELSIG7	Interrupt DMA factor selection	0 : TIMERB2 1 : I ² C-bus2	○:○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02C1)	Address (02C1 ₁₆)	After reset (00 ₁₆)	
		1	1								
								Bit symbol	Bit	Function	R
								IREQSELSIG8	Interrupt DMA factor selection	0 : TIMERB3 1 : Do not set	○:○
								IREQSELSIG9	Interrupt DMA factor selection	0 : TIMERB4 1 : Do not set	○:○
								IREQSELSIG10	Interrupt DMA factor selection	0 : TIMERB5 1 : OSD1	○:○
								IREQSELSIGA1	Interrupt factor selection	0 : UART2 bus shock detection 1 : I ² C-bus0	○:○
								Reserved bit	Must be set to "1".		○:○
								Reserved bit	Must be set to "1".		○:○
								IREQSELSIGA4	Interrupt DMA factor selection	0 : Key input 1 : VSYNC	○:○
								IREQSELSIGA5	Interrupt DMA factor selection	0 : AD 1 : I ² C-bus1NACK	○:○

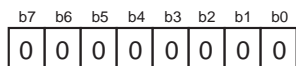
Extended register



Symbpl (EXTREG02C2) Address (02C2₁₆) After reset (00₁₆)

Bit symbol	Bit name	Function	R	W
IREQSELSIGA6	Interrupt DMA factor selection	0 : INT2 1 : OSD2	○	○
Reserved bit	Must be set to "0".		○	○

Extended register



Symbpl (EXTREG02C3) Address (02C3₁₆) After reset (00₁₆)

Bit symbol	Bit name	Function	R	W
Reserved bit	Must be set to "0".		○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02C4)	Address (02C4 ₁₆)	After reset (00 ₁₆)
0	0	0	0	0	0	0	0			

Bit symbol	Bit name	Function	R	W
Reserved bit	Must be set to "0".		○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02C5)	Address (02C5 ₁₆)	After reset (00 ₁₆)
0	0	0	0	0	0	0	0			

Bit symbol	Bit name	Function	R	W
Reserved bit	Must be set to "0".		○	○

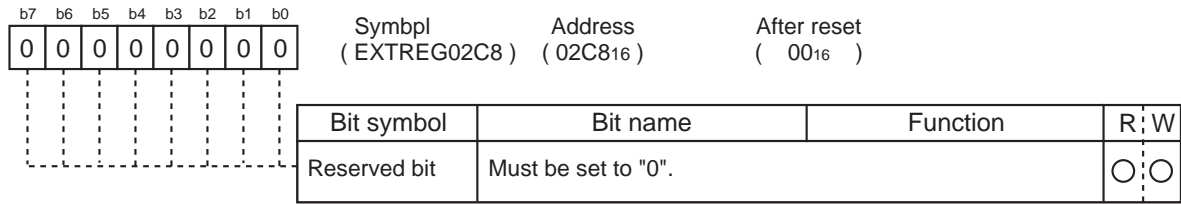
Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02C6)	Address (02C6 ₁₆)	After reset (00 ₁₆)
0	0	0	0	0	0	0	0			
<div style="display: flex; justify-content: space-around; width: 100%;"> </div>										
Bit symbol	Bit name	Function	R:W							
Reserved bit	Must be set to "0".		○:○							

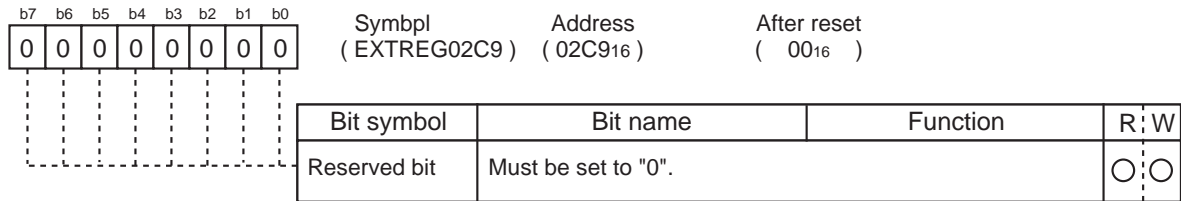
Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02C7)	Address (02C7 ₁₆)	After reset (00 ₁₆)
0	0	0	0	0	0	0	0			
<div style="display: flex; justify-content: space-around; width: 100%;"> </div>										
Bit symbol	Bit name	Function	R:W							
Reserved bit	Must be set to "0".		○:○							

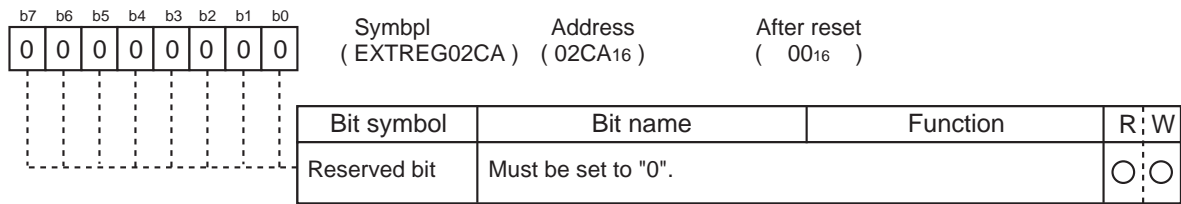
Extended register



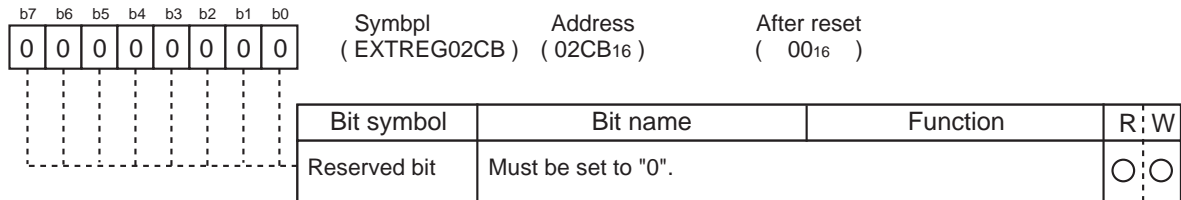
Extended register



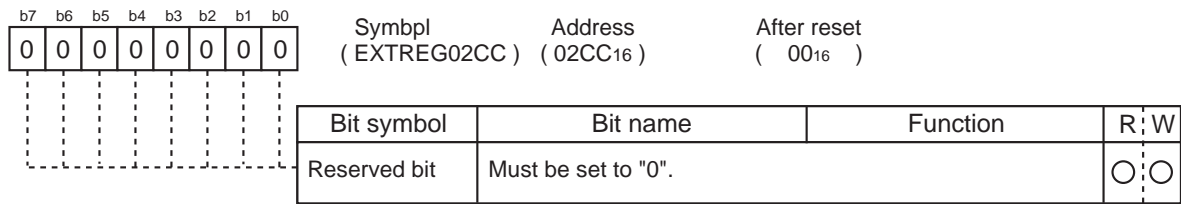
Extended register



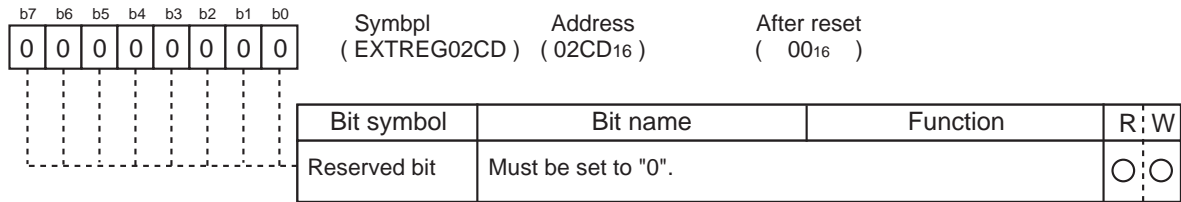
Extended register



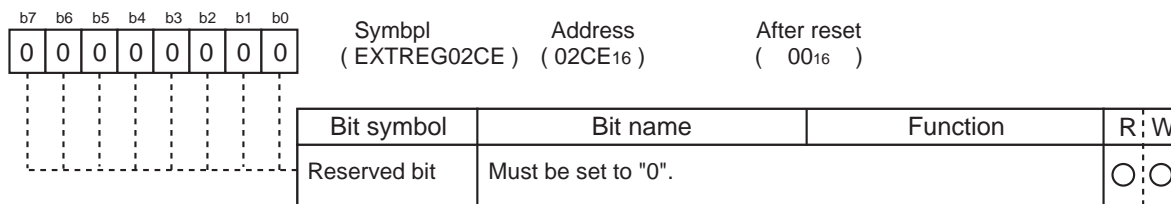
Extended register



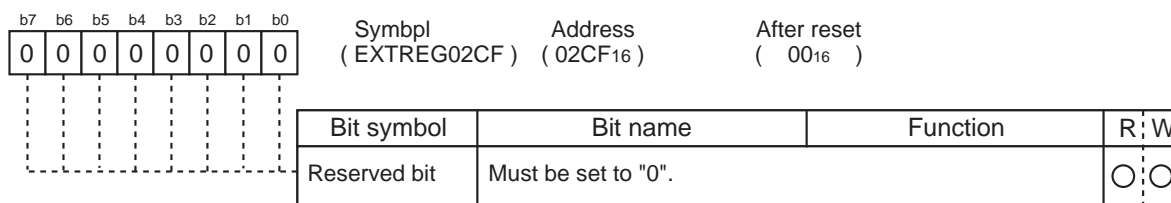
Extended register



Extended register



Extended register



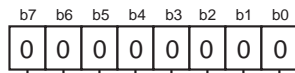
Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02D0)	Address (02D0 ₁₆)	After reset (00 ₁₆)			
0	0	0	0	0	0	0	0						
								Bit symbol	Bit name	Function	R	W	
								Reserved bit	Must be set to "0".			○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02D1)	Address (02D1 ₁₆)	After reset (00 ₁₆)			
0	0	0	0	0	0	0	0						
								Bit symbol	Bit name	Function	R	W	
								Reserved bit	Must be set to "0".			○	○

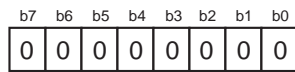
Extended register



Symbpl (EXTREG02D2) Address (02D2₁₆) After reset (00₁₆)

Bit symbol	Bit name	Function	R	W
Reserved bit	Must be set to "0".		○	○

Extended register



Symbpl (EXTREG02D3) Address (02D3₁₆) After reset (00₁₆)

Bit symbol	Bit name	Function	R	W
Reserved bit	Must be set to "0".		○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02D4)	Address (02D4 ₁₆)	After reset (00 ₁₆)
0	0	0	0	0	0	0	0			

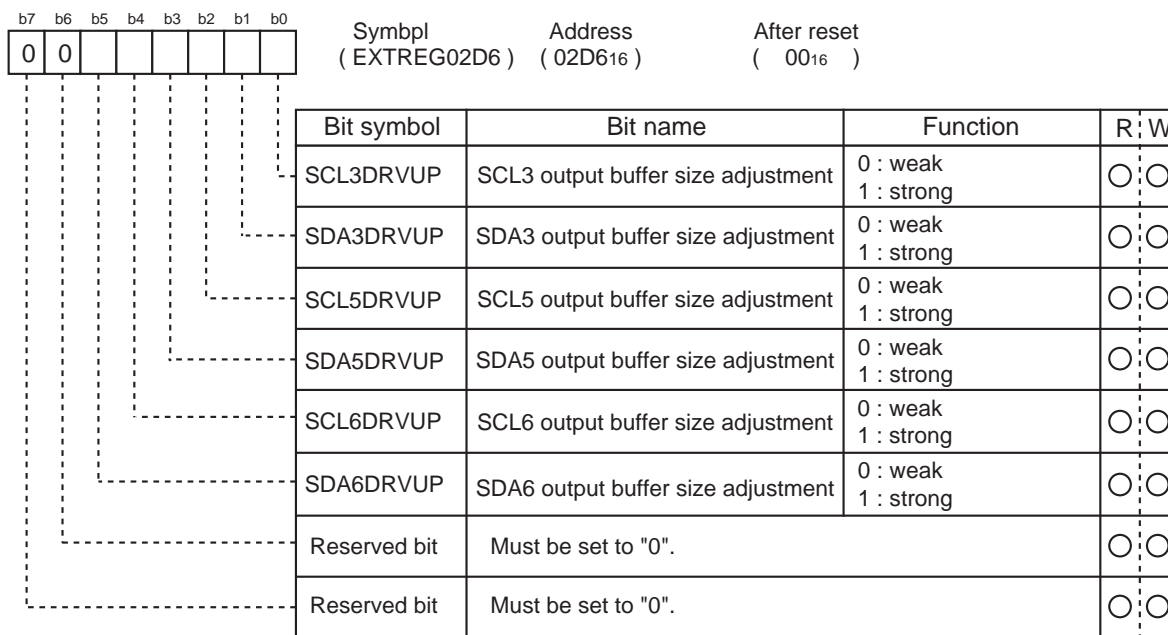
Bit symbol	Bit name	Function	R	W
— (b7-b0)	Reserved bits	Must be set to "0".	○	○

Extended register

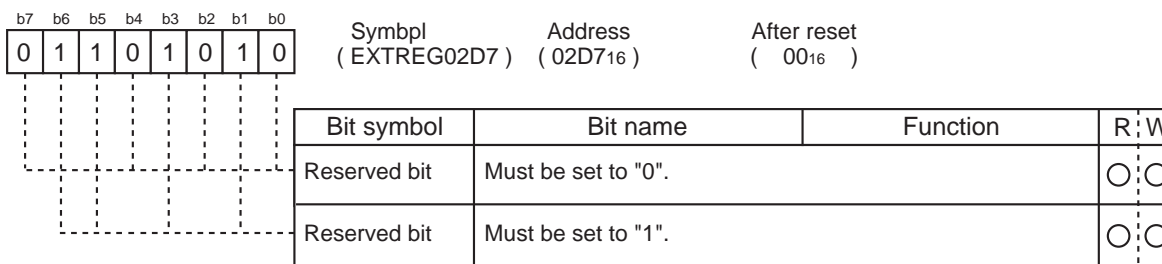
b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02D5)	Address (02D5 ₁₆)	After reset (00 ₁₆)
1		0	0	0	0	0				

Bit symbol	Bit name	Function	R	W
RGBSEL	RGB signal output selection bit	0: RGB 3-bit or analog output 1: RGB 2 values output	○	○
— (b5-b1)	Reserved bits	Must be set to "0".	○	○
CREGCPUSEL	Palette register selection	0: Palette register setting of palette set 0 1: Palette register setting of palette set 1	○	○
— (b7)	Reserved bit	Must be set to "1".	○	○

Extended register



Extended register



Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02D8)	Address (02D8 ₁₆)	After reset (00 ₁₆)
1	1									

Bit symbol	Bit name	Function	R	W
SCL0INSEL	I ² C-bus0 SCL input pin selection	0 : Pin "SCL2" 1 : Pin "SCL5"	○	○
SDA0INSEL	I ² C-bus0 SDA input pin selection	0 : Pin "SDA2" 1 : Pin "SDA5"	○	○
SCL1INSEL0	I ² C-bus1 SCL input pin selection	0 : Pin "SCL3" 1 : Pin "SCL6"	○	○
SDA1INSEL0	I ² C-bus1 SDA input pin selection	0 : Pin "SDA3" 1 : Pin "SDA6"	○	○
SCL1INSEL1	I ² C-bus1 SCL input pin selection	0 : SCL3 or SCL6 (SCL1INSEL0 available) 1 : Pin "SCL1"	○	○
SDA1INSEL1	I ² C-bus1 SDA input pin selection	0 : SDA3 or SDA6 (SDA1INSEL0 available) 1 : Pin "SDA1"	○	○
(b7-b6)	Reserved bits	Must be set to "1".	○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02D9)	Address (02D9 ₁₆)	After reset (00 ₁₆)

Bit symbol	Bit name	Function	R	W
BUSON1	SCL1-SCL3, SDA1-SDA3 bus switch	0 : OFF 1 : ON	○	○
BUSON2	SCL5-SCL6, SDA5-SDA6 bus switch	0 : OFF 1 : ON	○	○
SCLSDA1EN	SCL1, SDA1 pin control	0 : SCL1 and SDA1 are not used 1 : SCL1 and SDA1 are used.	○	○
SCLSDA2EN	SCL2, SDA2 pin control	0 : SCL2 and SDA2 are not used 1 : SCL2 and SDA2 are used.	○	○
SCLSDA3EN	SCL3, SDA3 pin control	0 : SCL3 and SDA3 are not used 1 : SCL3 and SDA3 are used.	○	○
SCLSDA4EN	SCL4, SDA4 pin control	0 : SCL4 and SDA4 are not used 1 : SCL4 and SDA4 are used.	○	○
SCLSDA5EN	SCL5, SDA5 pin control	0 : SCL5 and SDA5 are not used 1 : SCL5 and SDA5 are used.	○	○
SCLSDA6EN	SCL6, SDA6 pin control	0 : SCL6 and SDA6 are not used 1 : SCL6 and SDA6 are used.	○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02DA)	Address (02DA ₁₆)	After reset (00 ₁₆)
0	0	0	0	0	0	0	0			

Bit symbol (b7-b0)	Bit name	Function	R	W
	Reserved bit	Must be set to "0".	○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02DB)	Address (02DB ₁₆)	After reset (00 ₁₆)

Bit symbol	Bit name	Function	R	W
SELVIN	VSYNC input selection	0 : VSYNC1 1 : VSYNC2	○	○
DIGREN	R digital 3BIT output control	0 : DISABLE 1 : ENABLE	○	○
DIGGEN	Q digital 3BIT output control	0 : DISABLE 1 : ENABLE	○	○
DIGBEN	B digital 3BIT output control	0 : DISABLE 1 : ENABLE	○	○
OUT1EN	OUT1 output control	0 : DISABLE 1 : ENABLE	○	○
OUT2EN	OUT2 output control	0 : DISABLE 1 : ENABLE	○	○
OSCOUTEN	OSCOUT output control	0 : DISABLE 1 : ENABLE	○	○
OSCEN	Selection of OSD2/VSYNC1/INT2 function	0 : VSYNC1/INT2 input 1 : OSC2 output	○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02DC)	Address (02DC ₁₆)	After reset (00 ₁₆)
0	0	0	0	0	0	0	0			

Bit symbol	Bit name	Function	R	W
$\overline{\text{(b0)}}$	Reserved bit	Must be set to "0".	○	○
TST11	OSD oscillation circuit	0: Used (LC or ceramic) 1: Not used	○	○
$\overline{\text{(b7-b2)}}$	Reserved bits	Must be set to "0".	○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02DD)	Address (02DD ₁₆)	After reset (00 ₁₆)
0	0	0				0	0			

Bit symbol	Bit name	Function	R	W
$\overline{\text{(b1-b0)}}$	Reserved bits	Must be set to "0".	○	○
WSWL0	TB0IN pin noise filter clock selection bit	b3 b2 0 0 : 0.25 μ s (The removable maximum bus width=1 μ s) 0 1 : 8 μ s (The removable maximum bus width=32 μ s) 1 0 : 16 μ s (The removable maximum bus width=64 μ s) 1 1 : 32 μ s (The removable maximum bus width=128 μ s)	○	○
WSWL1			○	○
NFON	TB0IN pin noise filter ON/OFF selection	0 : Noise filter OFF 1 : Noise filter ON	○	○
$\overline{\text{(b7-b5)}}$	Reserved bits	Must be set to "0".	○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02DE)	Address (02DE ₁₆)	After reset (00 ₁₆)
0	0	0	0	0	0	0	0			

Bit symbol	Bit name	Function	R	W
$\overline{\text{---}}$ (b2-b0)	Reserved bit	Must be set to "0".	○	○
ANARGBCAPON	The CAP pin for inside operation of analog RGB stable	0 : CAP pin is not used 1 : CAP pin is used	○	○
ANARGBCLKEN	Analog RGB internal clock input control	0 : OFF 1 : ON	○	○
$\overline{\text{---}}$ (b6-b5)	Reserved bits	Must be set to "0".	○	○
$\overline{\text{---}}$ (b7)	Reserved bit	Must be set to "0".	○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0	Symbpl (EXTREG02DF)	Address (02DF ₁₆)	After reset (00 ₁₆)
0	0	0	0	0	0	0	0			

Bit symbol	Bit name	Function	R	W
$\overline{\text{---}}$ (b7-b0)	Reserved bit	Must be set to "0".	○	○

Electrical Characteristics

Table 18.1. Absolute Maximum Ratings

Symbol	Parameter		Condition	Rated value	Unit
VCC1,VCC2, VCC3	Supply voltage		VCC1=VCC2= VCC3	-0.3 to 6.5	V
VI	Input voltage	RESET, CNVss1, BYTE, CNVss2, P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P72 to P77, P82, P8, P87 P90 to P91, P103 to P107, VSYNC1, OSC1, HLF2, VHOLD2, CVIN2, HLF1,VHOLD1, CVIN1, HSYNC, XIN, SCL5, SDA5,SCL6, SDA6		-0.3 to Vcc1+0.3	V
		P70, P71, SCL4, SDA4		-0.3 to 6.5	V
VO	Output voltage	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P72 to P77, P82, P83, P86, P87 P90 to P91, P103 to P107, OUT1, OUT2, OSC2, OSCHLF, HLF2, VHOLD2, CVIN2, HLF1, VHOLD1, CVIN1, Xout, DIGR1, DIGR2, DIGG1, DIGG2, DIGB1, DIGB2, OSCOUT, R, G, B, SCL5, SDA5, SCL6,SDA6		-0.3 to Vcc1+0.3	V
		P70, P71, SCL4, SDA4		-0.3 to 6.5	V
Pd	Power dissipation		Topr=25°C	500	mW
Topr	Operating ambient temperature	At microcomputer operate		-20 to 70	°C
		At flash memory erase		0 to 60	°C
Tstg	Storage temperature			-40 to 125	°C

Table 18.2. Recommended Operating Conditions (Note 1)

Symbol	Parameter		Standard			Unit	
			Min.	Typ.	Max.		
VCC1, VCC2, VCC3	Supply voltage(VCC1=VCC2=VCC3)		3.15	3.3	3.45	V	
VSS	Supply voltage			0		V	
VIH	HIGH input voltage	P31 to P37, P40 to P47, P50 to P57	0.8VCC1		VCC1	V	
		P00 to P07, P10 to P17, P20 to P27, P30 (during single-chip mode)	0.8VCC1		VCC1	V	
		P00 to P07, P10 to P17, P20 to P27, P30 (data input during memory expansion and microprocessor modes)	0.5VCC1		VCC1	V	
		P60 to P67, P72 to P77, P82, P83, P86, 87, P90 to P91, P103 to P107, XIN, RESET, CNVSS1, BYTE, VSYNC1, OSC1, HSYNC, SCL5, SDA5, SCL6, SDA6	0.8VCC1		VCC1	V	
		P70, P71, SCL4, SDA4	0.8VCC1		6.5	V	
VIL	LOW input voltage	P31 to P37, P40 to P47, P50 to P57	0		0.2VCC1	V	
		P00 to P07, P10 to P17, P20 to P27, P30 (during single-chip mode)	0		0.2VCC1	V	
		P00 to P07, P10 to P17, P20 to P27, P30 (data input during memory expansion and microprocessor modes)	0		0.16VCC1	V	
		P60 to P67, P70 to P77, P82, P83, P86, P87, P90 to P91, P103 to P107, XIN, RESET, CNVSS1, BYTE, VSYNC1, OSC1, HSYNC, SCL4, SDA4, SCL5, SDA5, SCL6, SDA6	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, P82 to P83, P86, P87, P90, P91, P103 to P107, R, G, B, OUT1, OUT2, OSCOUT, DIGR1, DIGR2, DIGG1, DIGG2, DIGB1, DIGB2			-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, P82, P83, P86, P87, P90, P91, P103 to P107, R, G, B, OUT1, OUT2, OSCOUT, DIGR1, DIGR2, DIGG1, DIGG2, DIGB1, DIGB2			-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, P82, P83, P86, P87, P90, P91, P103 to P107, R, G, B, OUT1, OUT2, OSCOUT, DIGR1, DIGR2, DIGG1, DIGG2, DIGB1, DIGB2, SCL4, SDA4, SCL5, SDA5, SCL6, SDA6			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, P82, P83, P86, P87, P90, P91, P103 to P107, R, G, B, OUT1, OUT2, OSCOUT, DIGR1, DIGR2, DIGG1, DIGG2, DIGB1, DIGB2, SCL4, SDA4, SCL5, SDA5, SCL6, SDA6			5.0	mA	
f (XIN)	Main clock input oscillation frequency			16		MHz	
f (XCIN)	Sub-clock oscillation frequency			32.768		kHz	
f OSC	Oscillation frequency (for OSD)	OSC1	LC oscillation mode	8.0		30	MHz
			Ceramic oscillation mode	20.0		30	
			Internal oscillation mode (XIN=16MHz)	20.0		65	
f CVIN	Input frequency	The level synchronized signal of 525i (480i) video signal		15.262	15.734	16.206	kHz
		The level synchronized signal of 525p (480p) video signal		–	31.47	–	
VI	Input amplitude	Video signal CVIN1, CVIN2	1.5	1.75	2.00	V	

Note 1: Referenced to VCC = VCC1 = VCC2 = VCC3 = 3.3V ± 0.15V at Topr = -20 to 70 °C unless otherwise specified.

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

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

Table 18.3. A/D Conversion Characteristics (Note 1)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF} = V_{CC1}$			8	Bits
INL	Absolute accuracy	$V_{REF} = V_{CC1} = 3.3V$			+5	LSB
tCONV	Conversion time, Sample & hold function available	$\varnothing_{AD} = 10 \text{ MHz}$	2.8			μs
tsAMP	Sampling time		0.3			μs
VREF	Reference voltage			V_{CC1}		V
VIA	Analog input voltage		0		V_{CC1}	V

Note 1: Referenced to $V_{CC1} = 3.3V$, $V_{SS} = 0V$ at $T_{opr} = -20$ to 70°C unless otherwise specified.

Note 2: AD operation clock frequency (\varnothing_{AD} frequency) must be 10 MHz or less. And divide the f_{AD} and make \varnothing_{AD} frequency equal to or lower than $f_{AD}/2$.

Note 3: A case without sample & hold function turn \varnothing_{AD} frequency into 250 kHz or more in addition to a limit of Note 2.
 A case with sample & hold function turn \varnothing_{AD} frequency into 1MHz or more in addition to a limit of Note 2.

Table 18.4. Analog R,G,B output specifications ($V_{CC1} = 3.3V$, $V_{SS} = 0V$, $T_a = 25^\circ\text{C}$, Load capacity RI=nothing, Load capacity CI=nothing, unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
Vppm	Maximum output amplitude	RGB each output control bit=111b setup	$\alpha \times 0.8$	$\alpha = \frac{100}{140} \approx 0.71$	$\alpha \times 1.2$	V
Voe	Output deviation				± 20	%
Io	Maximum output current	RGB each output control bit=111b setup	2.2	4.0	5.8	mA
Ro	Output register		190		400	ΩA
Tst	Set ring time	30 to 70% or 70 to 30%			33	nS

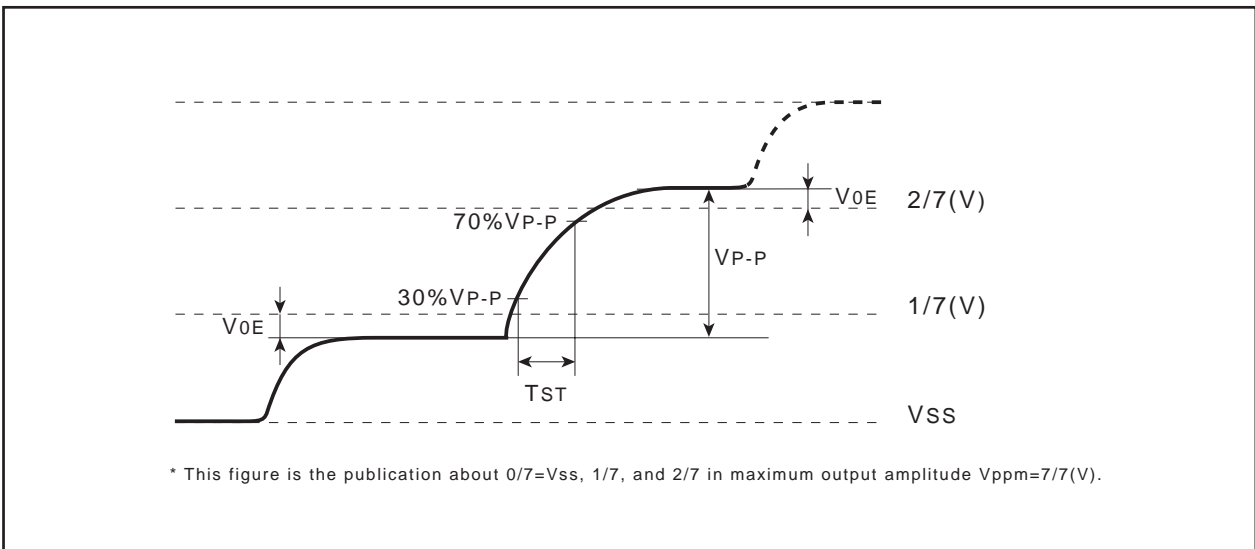


Figure 18.1. Analog RGB output characteristic

Table 18.5. Flash Memory Version Electrical Characteristics (Note 1)

Parameter	Standard			Unit
	Min.	Typ.	Max.	
Word program time		30	200	μs
Block erase time		1	4	s
Erase all unlocked blocks time		1 X n	4 X n	s
Lock bit program time		30	200	μs

Note 1: Referenced to VCC1=3.3V at Topr=0 to 60°C unless otherwise specified.
 Note 2: n denotes the number of block erases.

Table 18.6. Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics (at Topr = 0 to 60°C)

Flash program, erase voltage	Flash read operation voltage
VCC1 = 3.3 V ± 0.15V	VCC1=3.3 to 0.15 V

Table 18.7. 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	VCC1=2.7 to 3.45V			2	ms
td(R-S)	STOP Release Time				150	μs
td(W-S)	Low Power Dissipation Mode Wait Mode Release Time				150	μs

Note: When VCC1 = 5V.

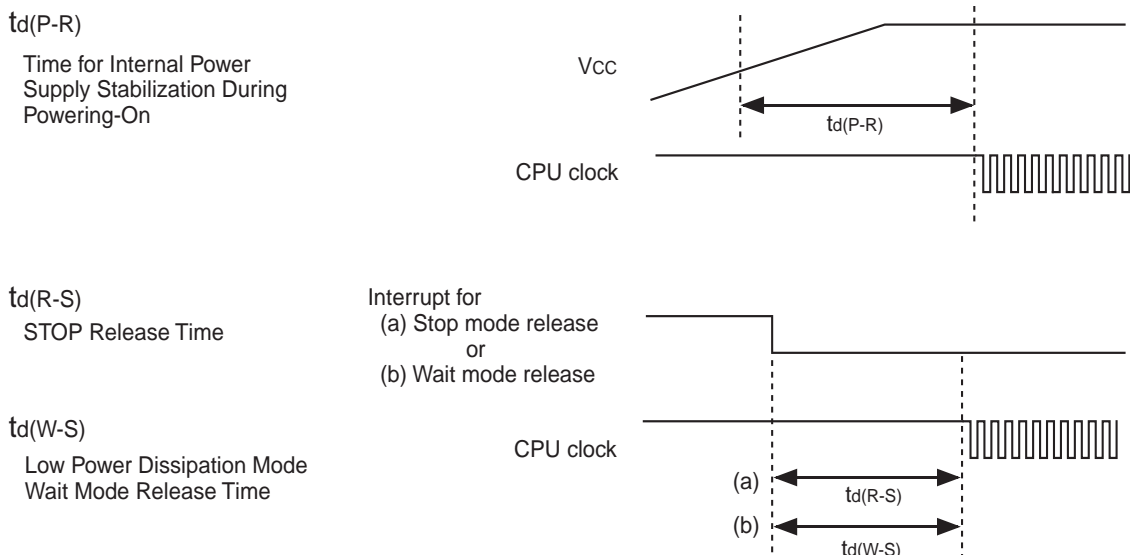


Table 18.8. Electrical Characteristics (Note 1)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
V _{OH}	HIGH output voltage	P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P4 ₀ to P4 ₇ , P5 ₀ to P5 ₇ , P6 ₀ to P6 ₇ , P7 ₂ to P7 ₇ , P8 ₂ , P8 ₃ , P8 ₈ , P8 ₇ , P9 ₀ , P9 ₁ , P10 ₃ to P10 ₇ , R, G, B, OUT1, OUT2, OSCOUT, DIGR1, DIGR2, DIGG1, DIGG2, DIGB1, DIGB2	I _{OH} = -1mA	V _{CC} - 0.5		V _{CC}	V
V _{OH}	HIGH output voltage	X _{OUT}	HIGHPOWER	I _{OH} = -0.1mA	V _{CC} - 0.5	V _{CC}	V
			LOWPOWER	I _{OH} = -50μA	V _{CC} - 0.5	V _{CC}	
	HIGH output voltage	X _{COU} T	HIGHPOWER	With no load applied		2.5	V
			LOWPOWER	With no load applied		1.6	
V _{OL}	LOW output voltage	P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P4 ₀ to P4 ₇ , P5 ₀ to P5 ₇ , P6 ₀ to P6 ₇ , P7 ₂ to P7 ₇ , P8 ₂ , P8 ₃ , P8 ₈ , P8 ₇ , P9 ₀ , P9 ₁ , P10 ₃ to P10 ₇ , R, G, B, OUT1, OUT2, OSCOUT, DIGR1, DIGR2, DIGG1, DIGG2, DIGB1, DIGB2, SCL4, SDA4, SCL5, SDA5, SCL6, SDA6	I _{OL} = 1mA			0.5	V
V _{OL}	LOW output voltage	X _{OUT}	HIGHPOWER	I _{OL} = 0.1mA		0.5	V
			LOWPOWER	I _{OL} = 50μA		0.5	
	LOW output voltage	X _{COU} T	HIGHPOWER	With no load applied		0	V
			LOWPOWER	With no load applied		0	
V _{T+} -V _{T-}	Hysteresis	HOLD, RDY, TA0 _{IN} to TA3 _{IN} , TB0 _{IN} , TB1 _{IN} , INT ₀ to INT ₃ , CTS ₀ to CTS ₂ , SCL0 to SCL6, SDA0 to SDA6, CLK ₀ to CLK ₂ , TA0 _{OUT} to TA3 _{OUT} , K1 to K3, RxD ₀ to RxD ₂ , VSYNC1, VSYNC2, HC0, HC1, HSYNC2		0.2		0.8	V
V _{T+} -V _{T-}	Hysteresis	RESET		0.2	(0.7)	1.8	V
V _{T+} -V _{T-}	Hysteresis	X _{IN}		0.2		0.8	V
I _{IH}	HIGH input current	P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P4 ₀ to P4 ₇ , P5 ₀ to P5 ₇ , P6 ₀ to P6 ₇ , P7 ₂ to P7 ₇ , P8 ₂ , P8 ₃ , P8 ₈ , P8 ₇ , P9 ₀ , P9 ₁ , P10 ₃ to P10 ₇ , X _{IN} , RESET, CNV _{SS} 1, BYTE, VSYNC1, OSC1, HSYNC, SCL4, SDA4, SCL5, SDA5, SCL6, SDA6	V _I = 3V			40	μA
I _{IL}	LOW input current	P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P4 ₀ to P4 ₇ , P5 ₀ to P5 ₇ , P6 ₀ to P6 ₇ , P7 ₂ to P7 ₇ , P8 ₂ , P8 ₃ , P8 ₈ , P8 ₇ , P9 ₀ , P9 ₁ , P10 ₃ to P10 ₇ , X _{IN} , RESET, CNV _{SS} 1, BYTE, VSYNC1, OSC1, HSYNC, SCL4, SDA4, SCL5, SDA5, SCL6, SDA6	V _I = 0V			-4.0	μA
R _{PULLUP}	Pull-up resistance	P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P4 ₀ to P4 ₇ , P5 ₀ to P5 ₇ , P6 ₀ to P6 ₇ , P7 ₂ to P7 ₇ , P8 ₂ , P8 ₃ , P8 ₈ , P8 ₇ , P9 ₀ , P9 ₁ , P10 ₃ to P10 ₇	V _I = 0V	66	160	500	kΩ
R _{XIN}	Feedback resistance	X _{IN}			3.0		MΩ
R _{X CIN}	Feedback resistance	X _{CIN}			25		MΩ
R _{BS}	I ² C-bus, Bus switch					130	Ω

Note 1: Referenced to V_{CC} = V_{CC1} = V_{CC2} = V_{CC3} = 3.3V, V_{SS} = 0V, T_{opr} = -20 to 70 °C unless otherwise specified.

Table 18.9. Electrical Characteristics (2) (Note 1)

Symbol	Parameter		Measuring condition			Standard			Unit
						Min.	Typ.	Max.	
I _{cc}	Power supply current	In single-chip mode, the output pins are open and other pins are V _{SS}	Mask ROM	f(BCLK)=16MHz, No division	OSD ON Data slicer ON		100	140	mA
					OSD OFF Data slicer OFF		15		mA
			Flash memory	f(BCLK)=16MHz, No division	OSD ON Data slicer ON		120	170	mA
					OSD OFF Data slicer OFF		15		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			6.0		μA
					f(BCLK)=32kHz, Wait mode(Note 2), Oscillation capacity Low		1.8		μA
					Stop mode, T _{opr} =25°C		0.7	3.0	μA

Note 1: Referenced to VCC = VCC1 = VCC2 = VCC3 = 3.3V, VSS = 0V, Topr = -20 to 70 °C 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.

Timing Requirements

(VCC1 = VCC2 = VCC3 = 3.3V, VSS = 0V, at Topr = – 20 to 70°C unless otherwise specified)

Table 18.10. External Clock Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c	External clock input cycle time	62		ns
t _{w(H)}	External clock input HIGH pulse width	25		ns
t _{w(L)}	External clock input LOW pulse width	25		ns
t _r	External clock rise time		15	ns
t _f	External clock fall time		15	ns

Table 18.11. Memory Expansion Mode and Microprocessor Mode

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{ac1} (RD-DB)	Data input access time (for setting with no wait)		(Note 1)	ns
t _{ac2} (RD-DB)	Data input access time (for setting with wait)		(Note 2)	ns
t _{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

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 60 \quad [\text{ns}]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 60 \quad [\text{ns}] \quad \text{n is "2" for 1-wait setting, "3" for 2-wait setting and "4" for 3-wait setting.}$$

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 60 \quad [\text{ns}] \quad \text{n is "2" for 2-wait setting, "3" for 3-wait setting.}$$

Timing Requirements**(V_{CC1} = V_{CC2} = V_{CC3} = 3.3V, V_{SS} = 0V, at Topr = – 20 to 70°C unless otherwise specified)****Table 18.12. Timer A Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(TA)}	TAi _{IN} input cycle time	300		ns
t _{w(TAH)}	TAi _{IN} input HIGH pulse width	60		ns
t _{w(TAL)}	TAi _{IN} input LOW pulse width	60		ns

Table 18.13. Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(TA)}	TAi _{IN} input cycle time	600		ns
t _{w(TAH)}	TAi _{IN} input HIGH pulse width	300		ns
t _{w(TAL)}	TAi _{IN} input LOW pulse width	300		ns

Table 18.14. Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(TA)}	TAi _{IN} input cycle time	300		ns
t _{w(TAH)}	TAi _{IN} input HIGH pulse width	150		ns
t _{w(TAL)}	TAi _{IN} input LOW pulse width	150		ns

Table 18.15. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{w(TAH)}	TAi _{IN} input HIGH pulse width	150		ns
t _{w(TAL)}	TAi _{IN} input LOW pulse width	150		ns

Table 18.16. Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(UP)}	TAi _{OUT} input cycle time	3000		ns
t _{w(UPH)}	TAi _{OUT} input HIGH pulse width	1500		ns
t _{w(UPL)}	TAi _{OUT} input LOW pulse width	1500		ns
t _{su(UP-TiN)}	TAi _{OUT} input setup time	600		ns
t _{h(TiN-UP)}	TAi _{OUT} input hold time	600		ns

Table 18.17. Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _{c(TA)}	TAi _{IN} input cycle time	2		∞s
t _{su(TAiN-TAiOUT)}	TAi _{OUT} input setup time	500		ns
t _{su(TAiOUT-TAiN)}	TAi _{IN} input setup time	500		ns

Timing Requirements**(VCC1 = VCC2 = VCC3 = 3.3V, VSS = 0V, at Topr = – 20 to 70°C unless otherwise specified)****Table 18.18. Timer B Input (Counter Input in Event Counter Mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN input cycle time (counted on one edge)	150		ns
tw(TBH)	TBiIN input HIGH pulse width (counted on one edge)	60		ns
tw(TBL)	TBiIN input LOW pulse width (counted on one edge)	60		ns
tc(TB)	TBiIN input cycle time (counted on both edges)	300		ns
tw(TBH)	TBiIN input HIGH pulse width (counted on both edges)	160		ns
tw(TBL)	TBiIN input LOW pulse width (counted on both edges)	160		ns

Table 18.19. Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN input cycle time	600		ns
tw(TBH)	TBiIN input HIGH pulse width	300		ns
tw(TBL)	TBiIN input LOW pulse width	300		ns

Table 18.20. Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN input cycle time	600		ns
tw(TBH)	TBiIN input HIGH pulse width	300		ns
tw(TBL)	TBiIN input LOW pulse width	300		ns

Table 18.21. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	CLKi input cycle time	300		ns
tw(CKH)	CLKi input HIGH pulse width	150		ns
tw(CKL)	CLKi input LOW pulse width	150		ns
td(C-Q)	TxDi output delay time		160	ns
th(C-Q)	TxDi hold time	0		ns
tsu(D-C)	RxDi input setup time	100		ns
th(C-D)	RxDi input hold time	90		ns

Table 18.22. External Interrupt $\overline{\text{INTi}}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(INH)	$\overline{\text{INTi}}$ input HIGH pulse width	380		ns
tw(INL)	$\overline{\text{INTi}}$ input LOW pulse width	380		ns

Switching Characteristics

(VCC1 = VCC2 = VCC3 = 3.3V, VSS = 0V, at Topr = - 20 to 70°C, CM15 = “1” unless otherwise specified)

Table 18.23. Memory Expansion and Microprocessor Modes (for setting with no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t _d (BCLK-AD)	Address output delay time	Fig.19.11		30	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)		0		ns
t _h (WR-AD)	Address output hold time (refers to WR)		(Note 2)		ns
t _d (BCLK-CS)	Chip select output delay time			30	ns
t _h (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t _d (BCLK-ALE)	ALE signal output delay time			30	ns
t _h (BCLK-ALE)	ALE signal output hold time		-4		ns
t _d (BCLK-RD)	RD signal output delay time			30	ns
t _h (BCLK-RD)	RD signal output hold time		0		ns
t _d (BCLK-WR)	WR signal output delay time			30	ns
t _h (BCLK-WR)	WR signal output hold time		0		ns
t _d (BCLK-DB)	Data output delay time (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 1)		ns
t _h (WR-DB)	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns
t _d (BCLK-HLDA)	HLDA output delay time			40	ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \quad [\text{ns}]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \quad [\text{ns}]$$

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

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2V_{CC2} / V_{CC2}) = 6.7\text{ns}.$$

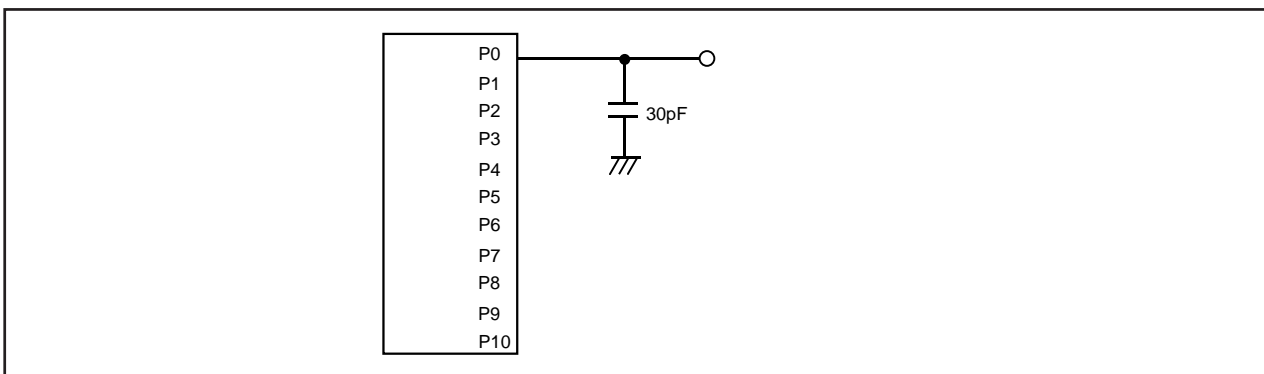
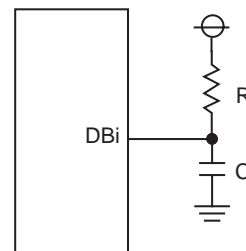


Figure 18.2. Ports P0 to P10 Measurement Circuit

Switching Characteristics

(VCC1 = VCC2 = VCC3 = 3.3V, VSS = 0V, at Topr = – 20 to 70°C, CM15 = “1” unless otherwise specified)

Table 18.24. Memory Expansion and Microprocessor Modes
(for 1- to 3-wait setting and external area access)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t _d (BCLK-AD)	Address output delay time	Fig.19.11		30	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)		0		ns
t _h (WR-AD)	Address output hold time (refers to WR)		(Note 2)		ns
t _d (BCLK-CS)	Chip select output delay time			30	ns
t _h (BCLK-CS)	Chip select output hold time (refers to BCLK)		4		ns
t _d (BCLK-ALE)	ALE signal output delay time			30	ns
t _h (BCLK-ALE)	ALE signal output hold time		–4		ns
t _d (BCLK-RD)	RD signal output delay time			30	ns
t _h (BCLK-RD)	RD signal output hold time		0		ns
t _d (BCLK-WR)	WR signal output delay time			30	ns
t _h (BCLK-WR)	WR signal output hold time		0		ns
t _d (BCLK-DB)	Data output delay time (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 1)		ns
t _h (WR-DB)	Data output hold time (refers to WR)(Note 3)		(Note 2)		ns
t _d (BCLK-HLDA)	HLDA output delay time			40	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:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \quad [\text{ns}]$$

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.

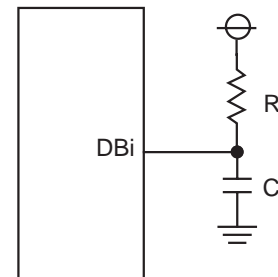
Hold time of data bus is expressed in

$$t = -CR \times \ln(1 - V_{OL} / V_{CC2})$$

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

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2V_{CC2} / V_{CC2})$$



Switching Characteristics(V_{CC1} = V_{CC2} = V_{CC3} = 3.3V, V_{SS} = 0V, at Topr = – 20 to 70°C, CM15 = “1” unless otherwise specified)**Table 18.25. 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	Fig.19.11		50	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			50	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			40	ns
t _h (BCLK-RD)	RD signal output hold time		0		ns
t _d (BCLK-WR)	WR signal output delay time			40	ns
t _h (BCLK-WR)	WR signal output hold time		0		ns
t _d (BCLK-DB)	Data output delay time (refers to BCLK)			50	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-HLDA)	HLDA output delay time			40	ns
t _d (BCLK-ALE)	ALE signal output delay time (refers to BCLK)			40	
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)		(Note 4)		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:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 10 \quad [\text{ns}]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(\text{BCLK})} - 50 \quad [\text{ns}] \quad n \text{ is "2" for 2-wait setting, "3" for 3-wait setting.}$$

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 40 \quad [\text{ns}]$$

Note 4: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(\text{BCLK})} - 15 \quad [\text{ns}]$$

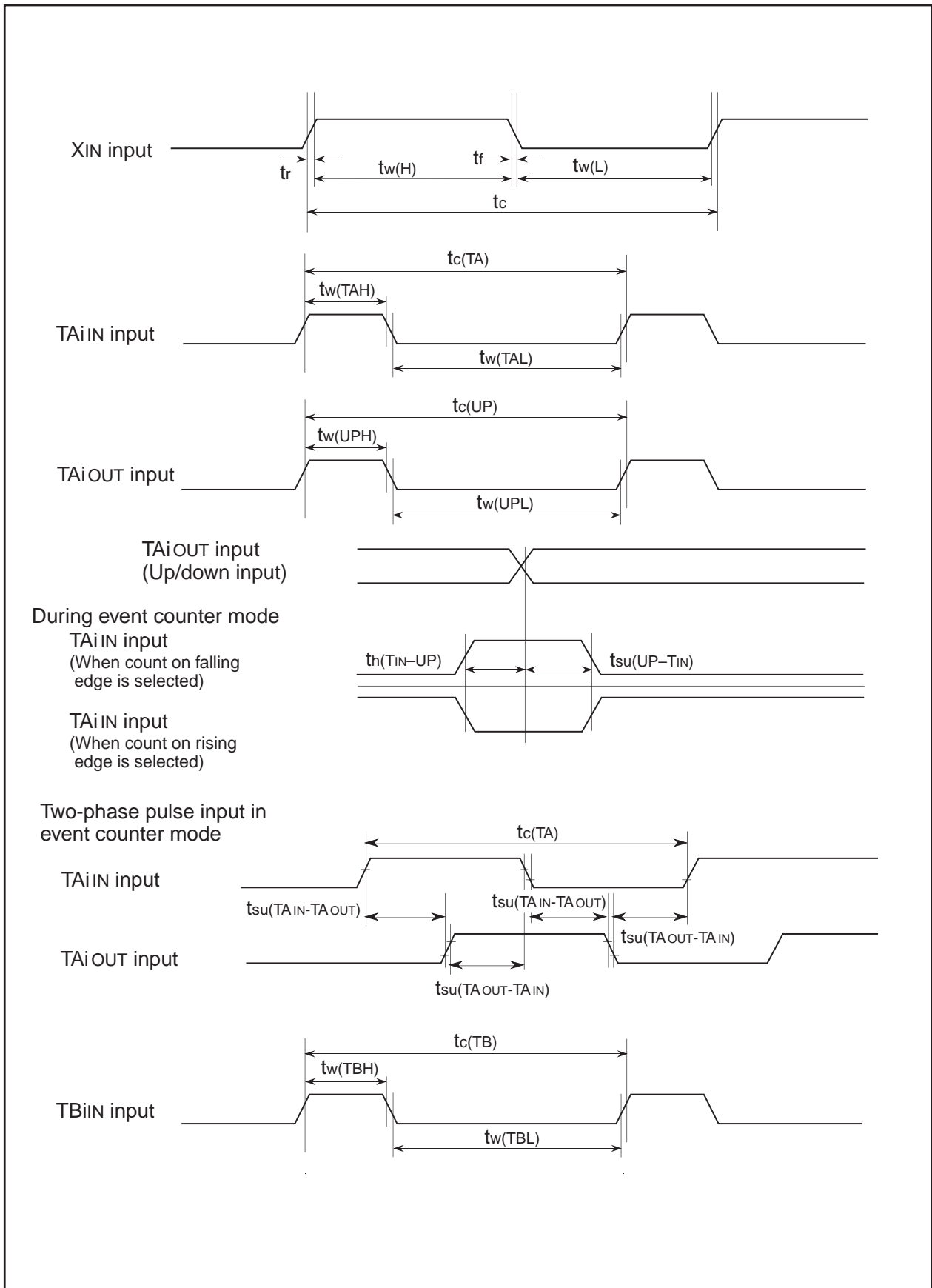


Figure 18.3. Timing Diagram (1)

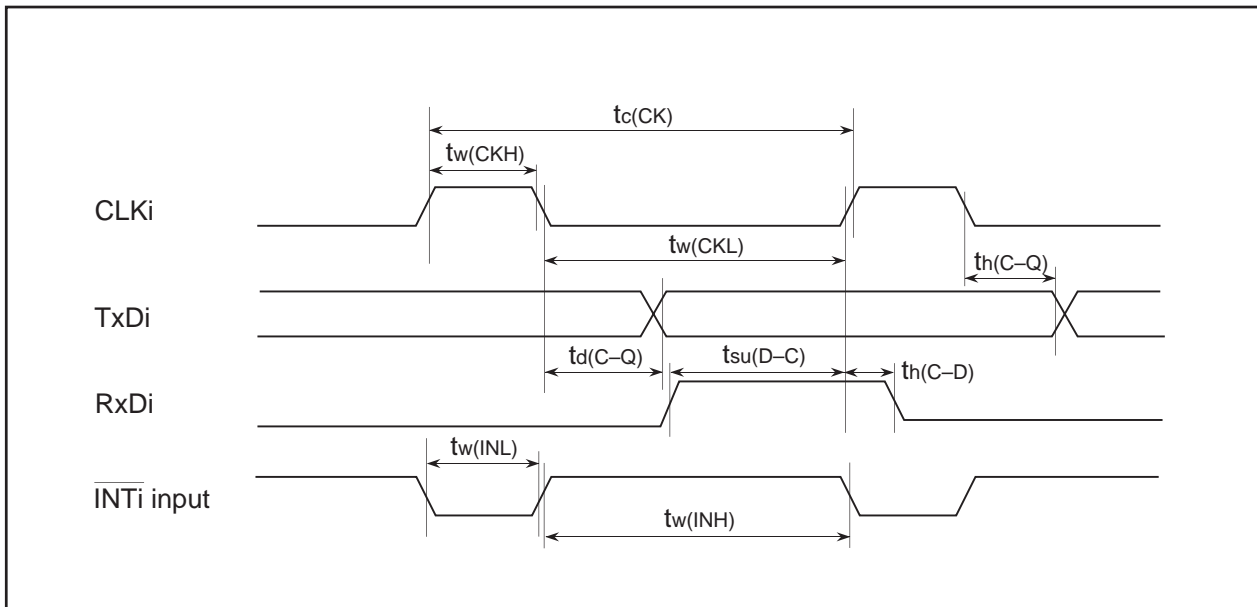
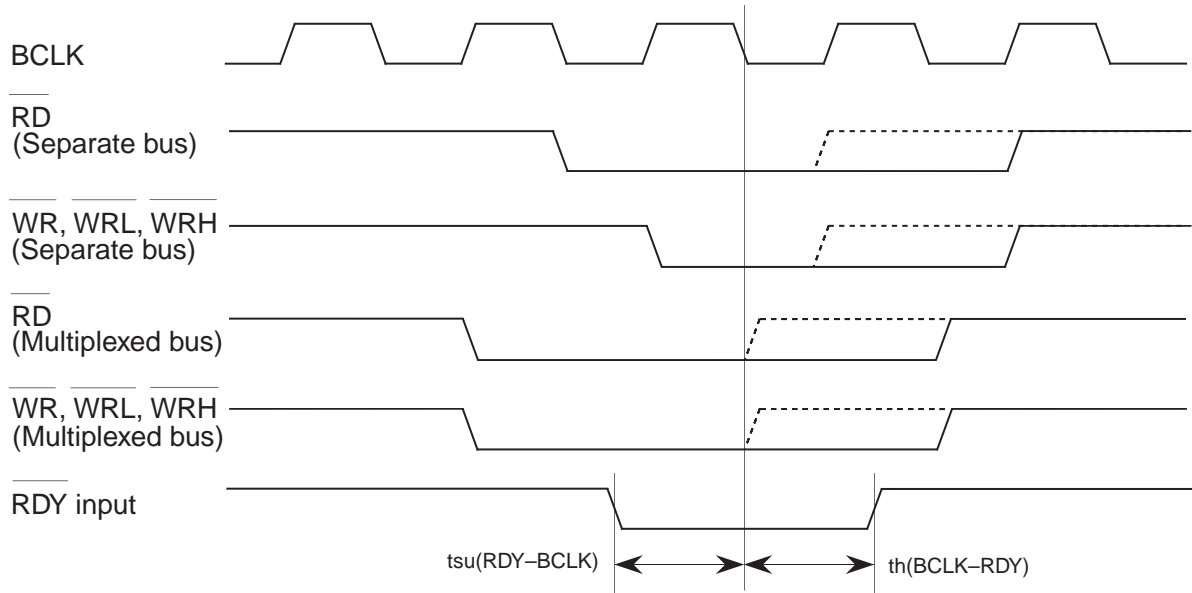


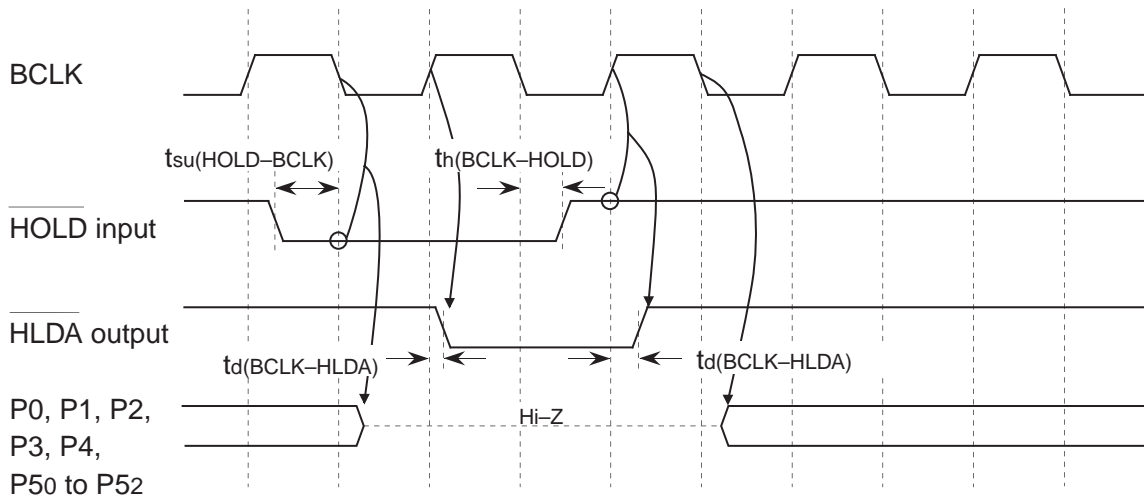
Figure 18.4. Timing Diagram (2)

Memory Expansion Mode, Microprocessor Mode

(Effective for setting with wait)



(Common to setting with wait and setting without wait)



Note: The above pins are set to high-impedance regardless of the input level of the BYTE pin, PM06 bit in PM0 register.

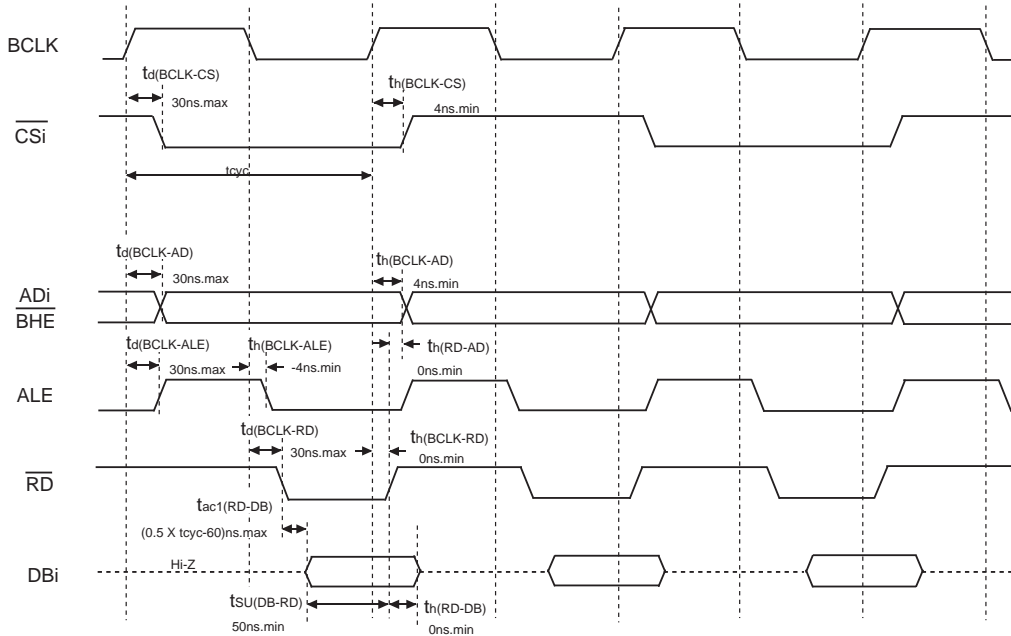
Measuring conditions :

- $V_{CC1}=V_{CC2}=V_{CC3}=3.3V$
- Input timing voltage : Determined with $V_{IL}=0.6V$, $V_{IH}=2.4V$
- Output timing voltage : Determined with $V_{OL}=1.5V$, $V_{OH}=1.5V$

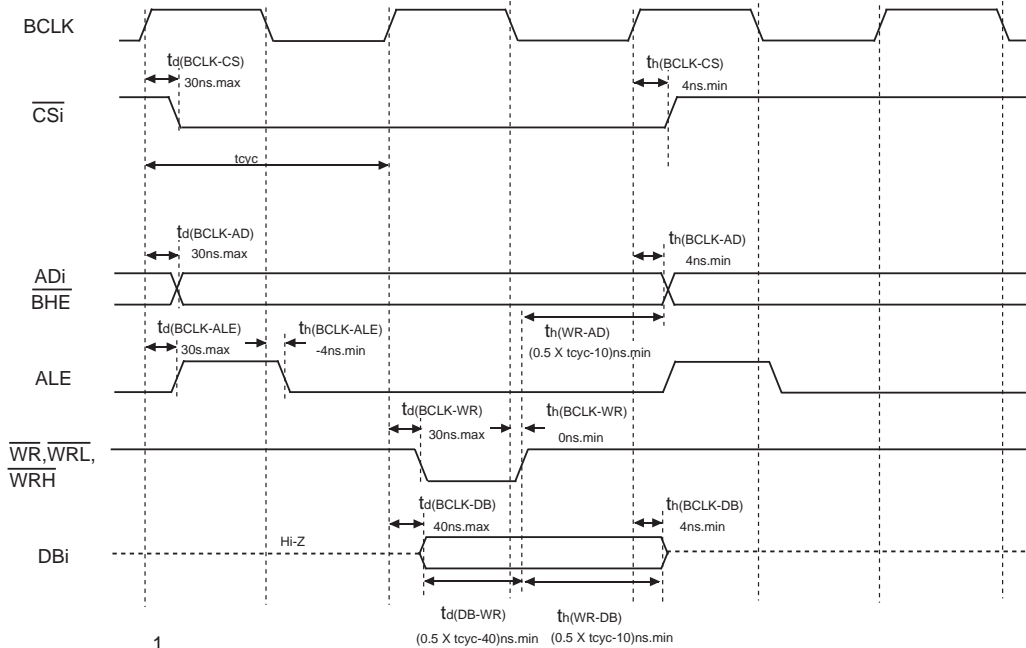
Figure 18.5. Timing Diagram (3)

Memory Expansion Mode, Microprocessor Mode (For setting with no wait)

Read timing



Write timing



$$t_{\text{CYC}} = \frac{1}{f(\text{BCLK})}$$

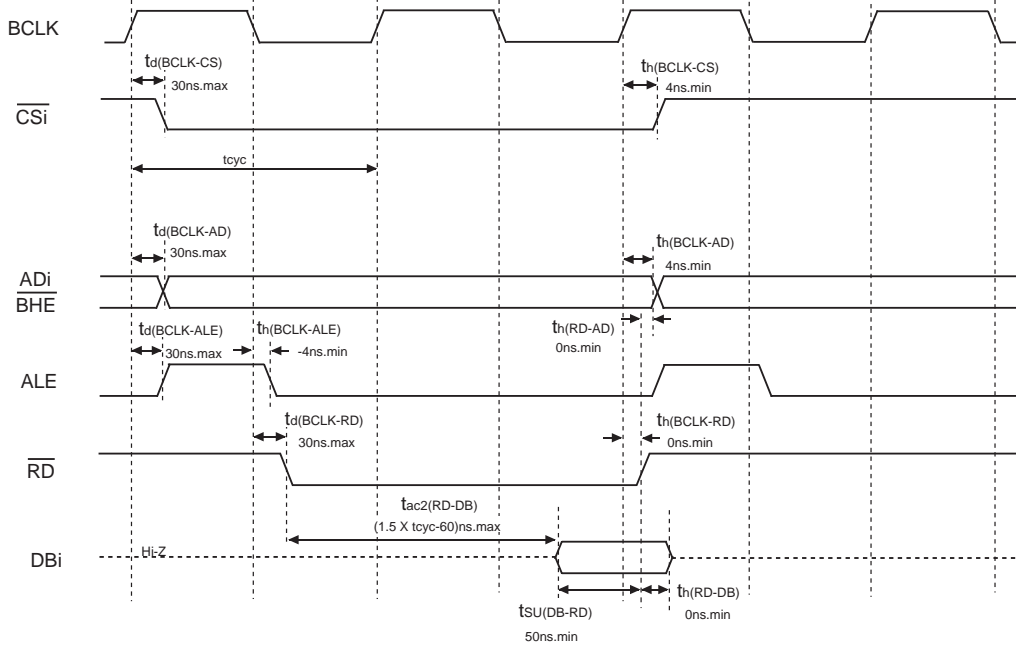
Measuring conditions

- VCC1=VCC2=VCC3=3.3V
- Input timing voltage : VIL=0.6V, VIH=2.4V
- Output timing voltage : VOL=1.5V, VOH=1.5V

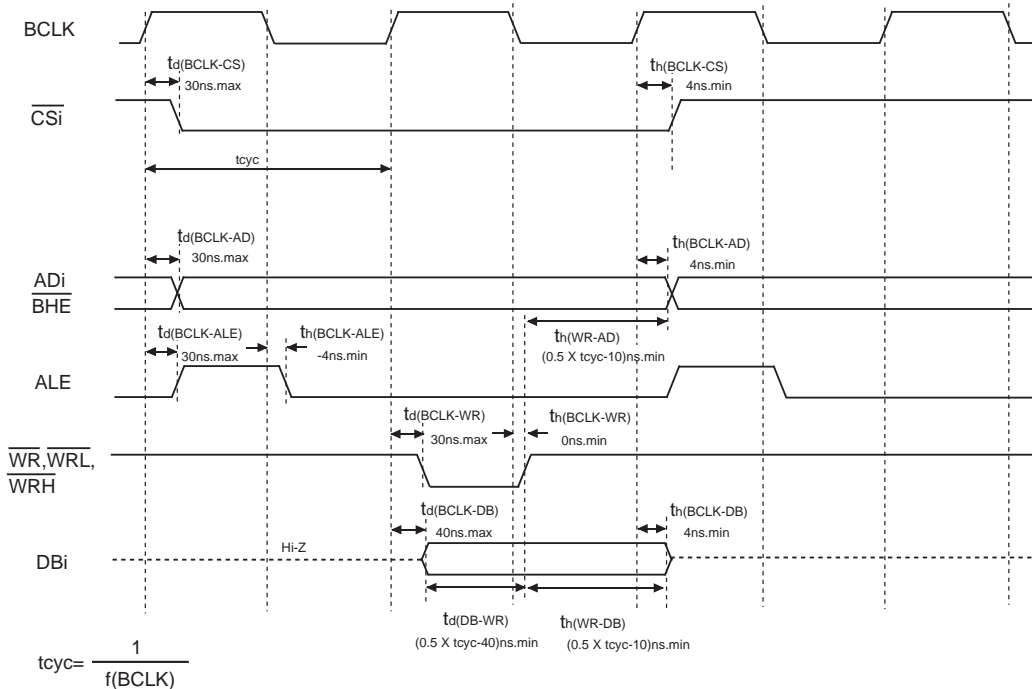
Figure 18.6. Timing Diagram (4)

Memory Expansion Mode, Microprocessor Mode
 (for 1-wait setting and external area access)

Read timing



Write timing



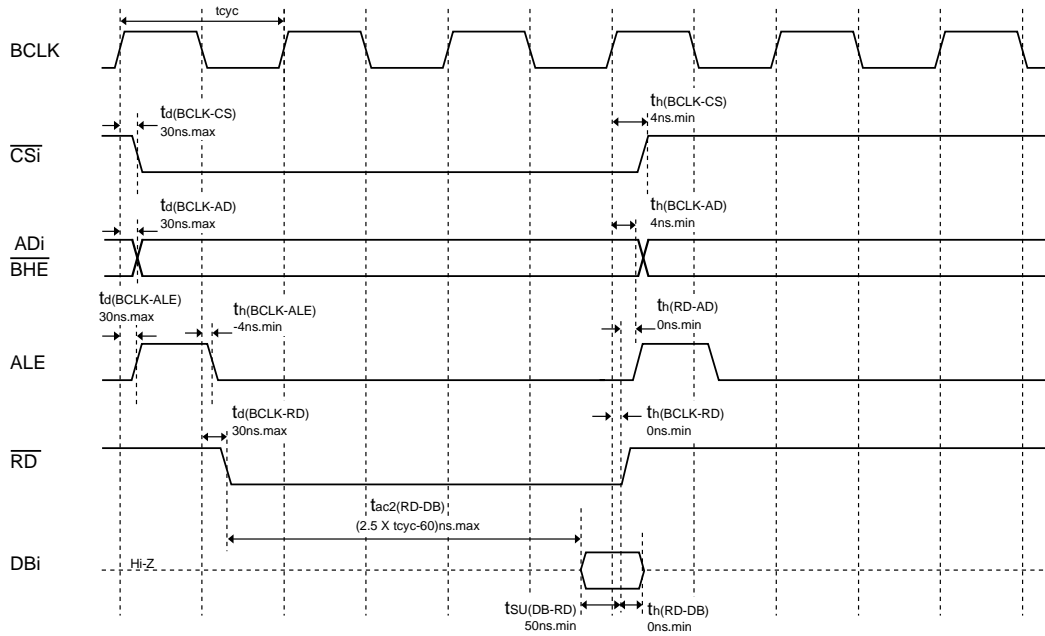
- Measuring conditions
- $V_{CC1}=V_{CC2}=V_{CC3}=3V$
 - Input timing voltage : $V_{IL}=0.6V, V_{IH}=2.4V$
 - Output timing voltage : $V_{OL}=1.5V, V_{OH}=1.5V$

Figure 18.7. Timing Diagram (5)

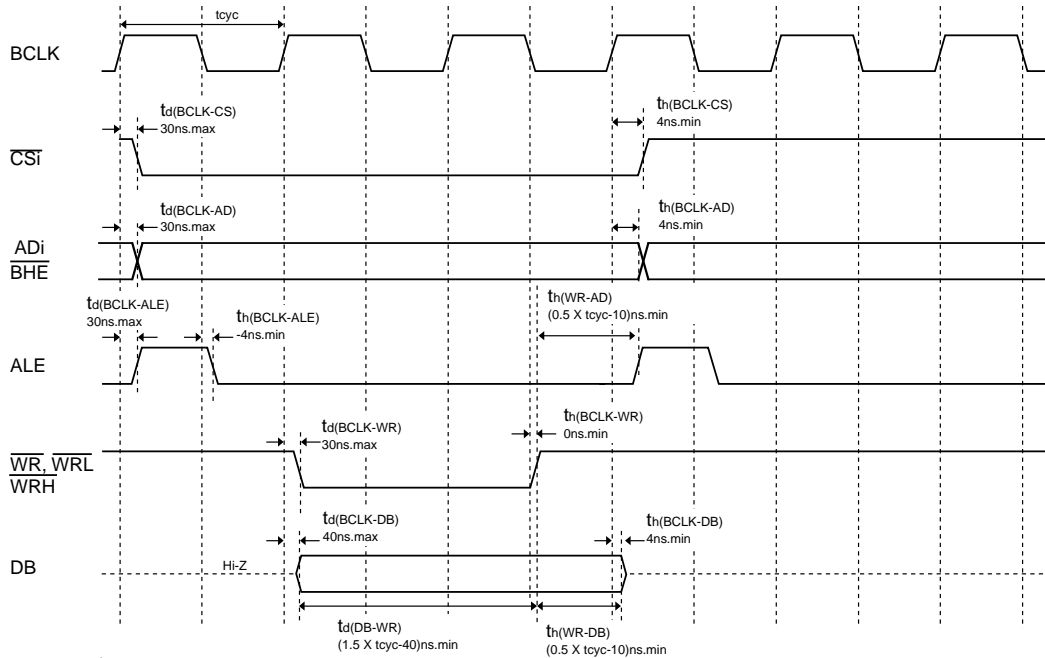
Memory Expansion Mode, Microprocessor Mode

(for 2-wait setting and external area access)

Read timing



Write timing



$$t_{cy} = \frac{1}{f(\text{BCLK})}$$

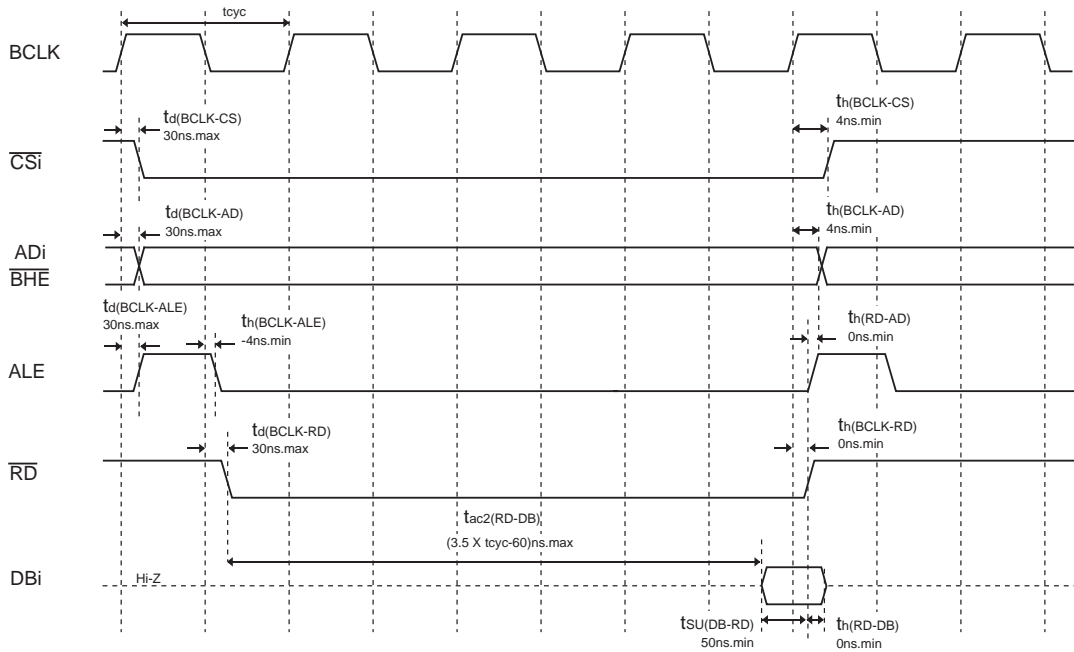
Measuring conditions

- $V_{CC1}=V_{CC2}=V_{CC3}=3.3\text{V}$
- Input timing voltage : $V_{IL}=0.6\text{V}$, $V_{IH}=2.4\text{V}$
- Output timing voltage : $V_{OL}=1.5\text{V}$, $V_{OH}=1.5\text{V}$

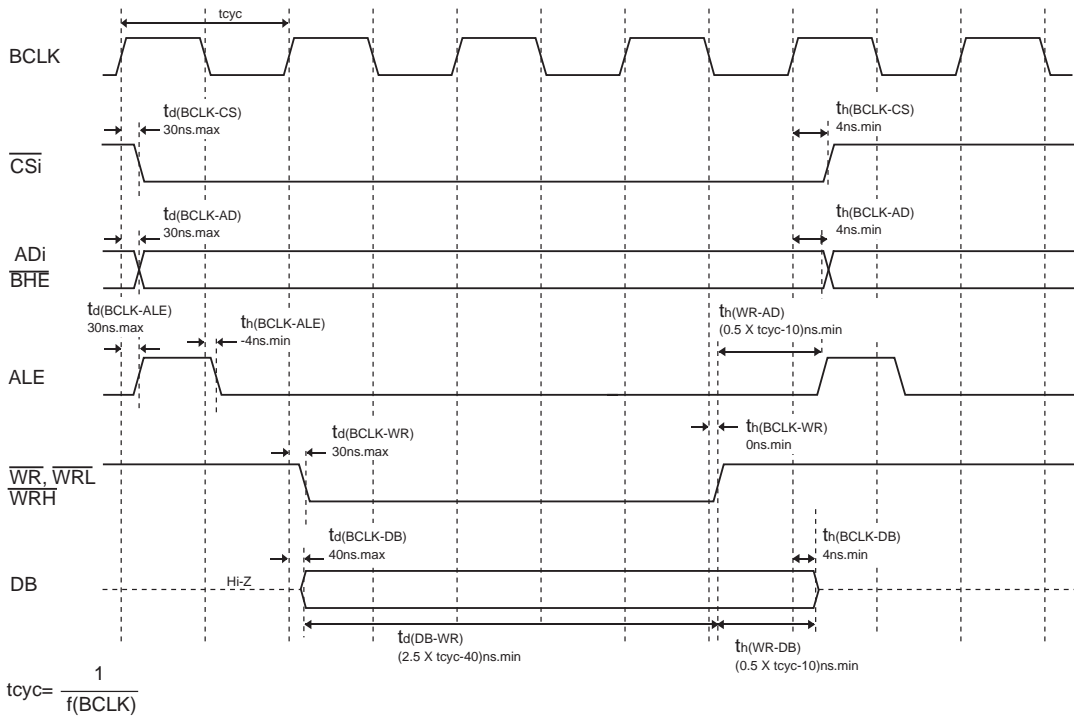
Figure 18.8. Timing Diagram (6)

Memory Expansion Mode, Microprocessor Mode (for 3-wait setting and external area access)

Read timing



Write timing

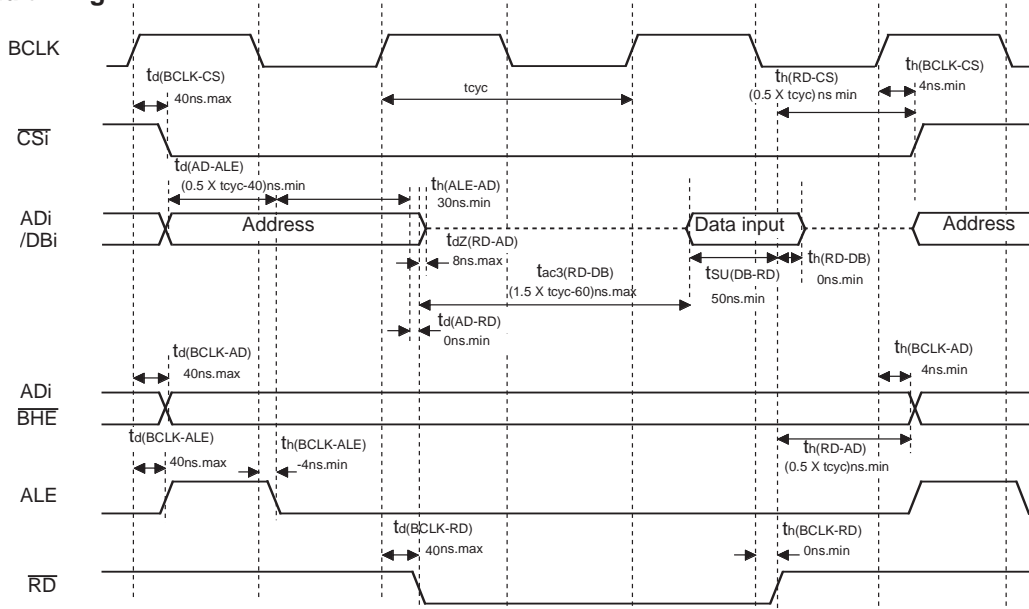


- Measuring conditions
- $V_{CC1}=V_{CC2}=V_{CC3}=3.3V$
 - Input timing voltage : $V_{IL}=0.6V, V_{IH}=2.4V$
 - Output timing voltage : $V_{OL}=1.5V, V_{OH}=1.5V$

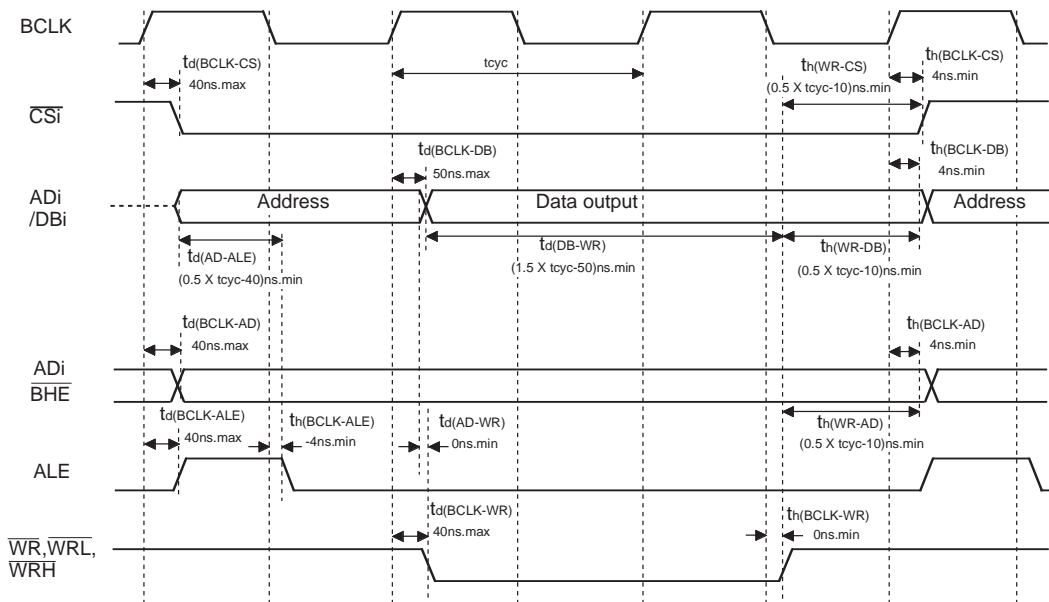
Figure 18.9. Timing Diagram (7)

Memory Expansion Mode, Microprocessor Mode (For 2-wait setting, external area access and multiplex bus selection)

Read timing



Write timing



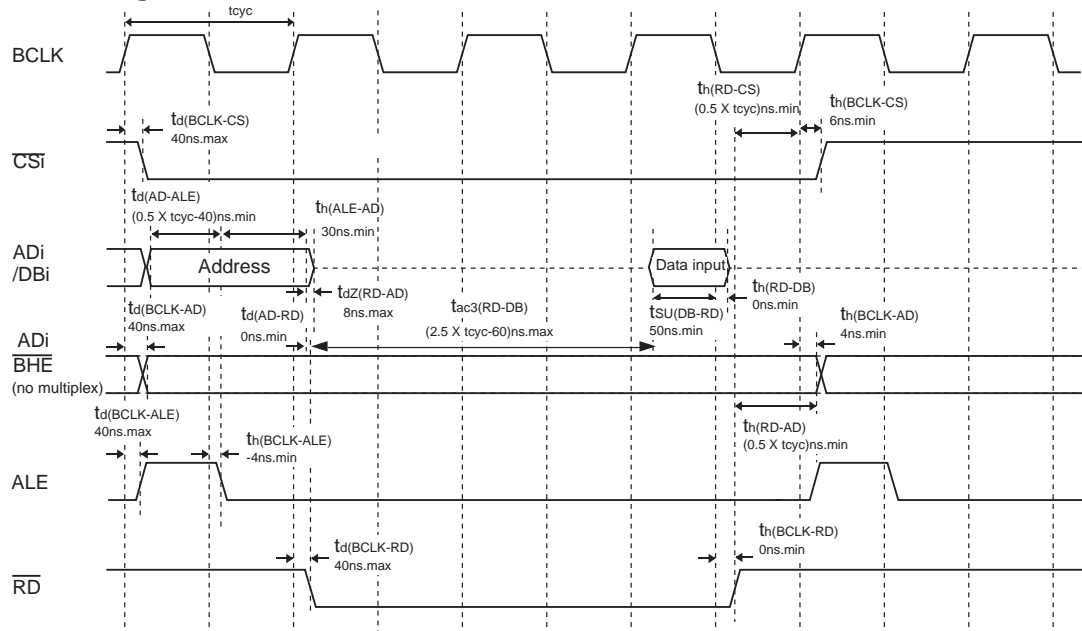
$$t_{cyc} = \frac{1}{f(BCLK)}$$

- Measuring conditions
- $V_{CC1}=V_{CC2}=V_{CC3}=3.3V$
 - Input timing voltage : $V_{IL}=0.6V, V_{IH}=2.4V$
 - Output timing voltage : $V_{OL}=1.5V, V_{OH}=1.5V$

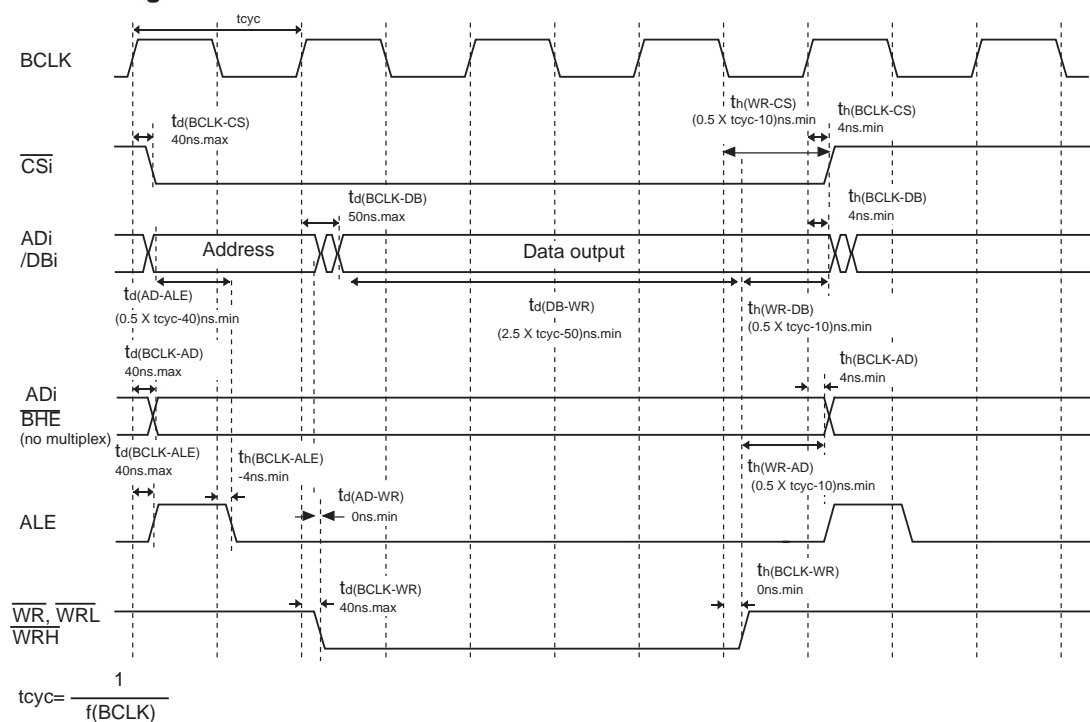
Figure 18.10. Timing Diagram (8)

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=3.3V
- Input timing voltage : VIL=0.6V, VIH=2.4V
- Output timing voltage : VOL=1.5V, VOH=1.5V

Figure 18.11. 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 19.1 shows the outline performance of flash memory version (refer to “Table 1.1. Performance outline of M306V8FJFP” for the items not listed in Table 19.1.).

Table 19.1. 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	Refer to “Figure 19.1. Flash Memory Block Diagram”
	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
ROM code protection		Parallel I/O and standard serial I/O modes are supported.

Notes 1: The boot ROM area contains a standard serial I/O mode rewrite control program which is stored in it when shipped from the factory. This area can only be rewritten in parallel input/output mode

2: Can be programmed in byte units in only parallel input/output mode.

Table 19.2. Flash Memory Rewrite Modes Overview

Flash memory rewrite mode	CPU rewrite mode (Note 1)	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 (Note 2) 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

Note 1: The PM13 bit remains set to "1" while the FMR0 register FMR01 bit = 1 (CPU rewrite mode enabled). The PM13 bit is reverted to its original value by clearing the FMR01 bit to "0" (CPU rewrite mode disabled). However, if the PM13 bit is changed during CPU rewrite mode, its changed value is not reflected until after the FMR01 bit is cleared to "0".

Note 2: When in CPU rewrite mode, the PM10 and PM13 bits in the PM1 register are set to "1". The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1. When the PM13 bit = 0 and the flash memory is used in 4M-byte mode, the extended accessible area (50000₁₆ to BFFFF₁₆) cannot be used.

Memory Map

The ROM in the flash memory version is separated between a user ROM area and a boot ROM area, OSD ROM area. Figure 19.1 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 10000₁₆ to 26FFF₁₆).

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 CNVss1 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 CNVss1 pin, the program in the user ROM area is executed (but the boot ROM area cannot be read). OSD ROM area that stores character font data. It is rewritten in CPU rewriting mode, standard serial I/O mode, or parallel I/O mode.

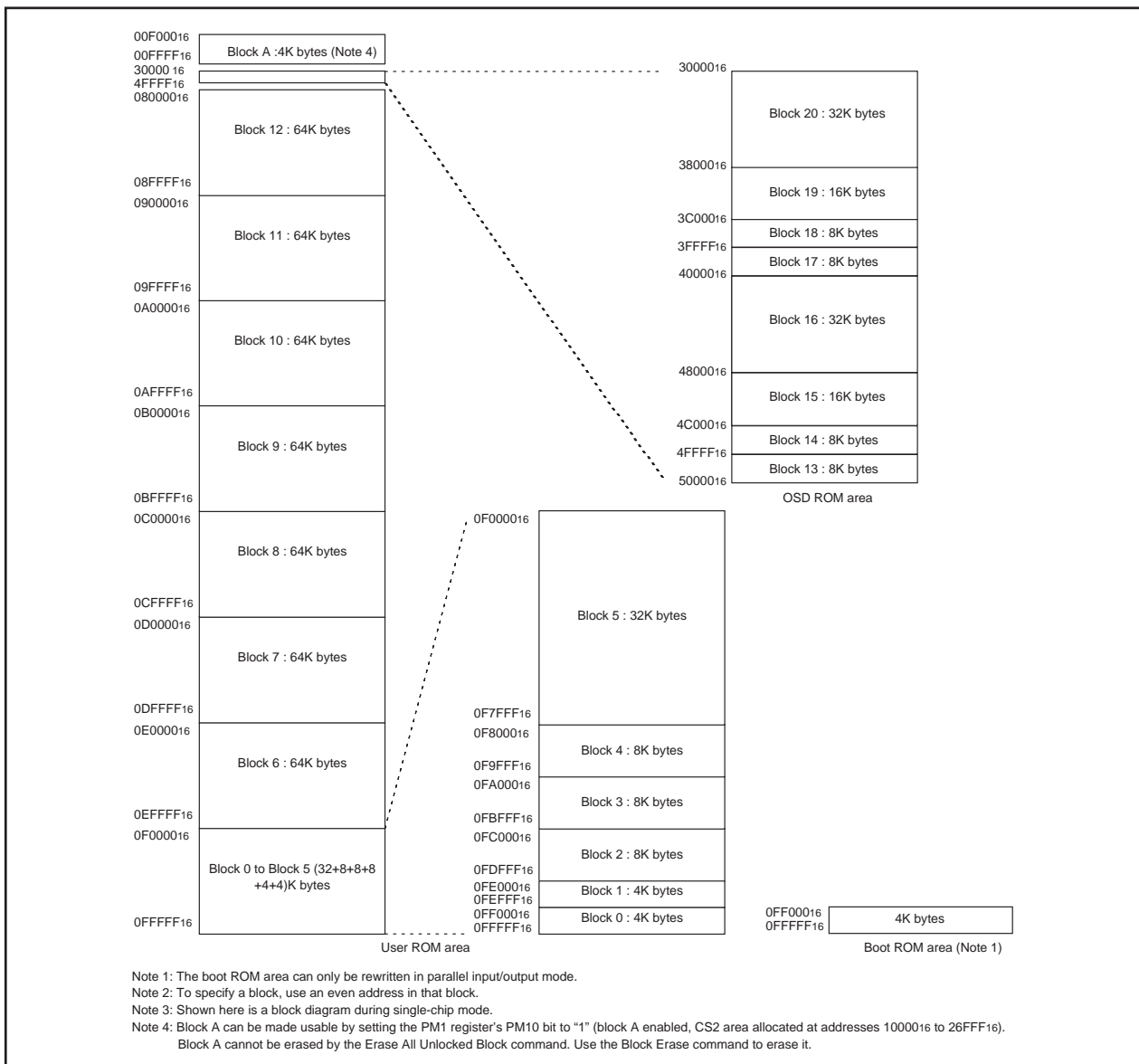


Figure 19.1. Flash Memory Block Diagram

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 CNVSS1 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 19.2 shows the ROMCP register.

The ROMCP register is located in the user ROM area. The ROM code protect function is enabled when the ROMCR bits are set to other than "11b". In this case, set the bit 5 to bit 0 to "111111b".

When exiting ROM code protect, erase the block including the ROMCP1 register by the CPU rewrite mode or the standard serial I/O mode.

- **ID Code Check Function**

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

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

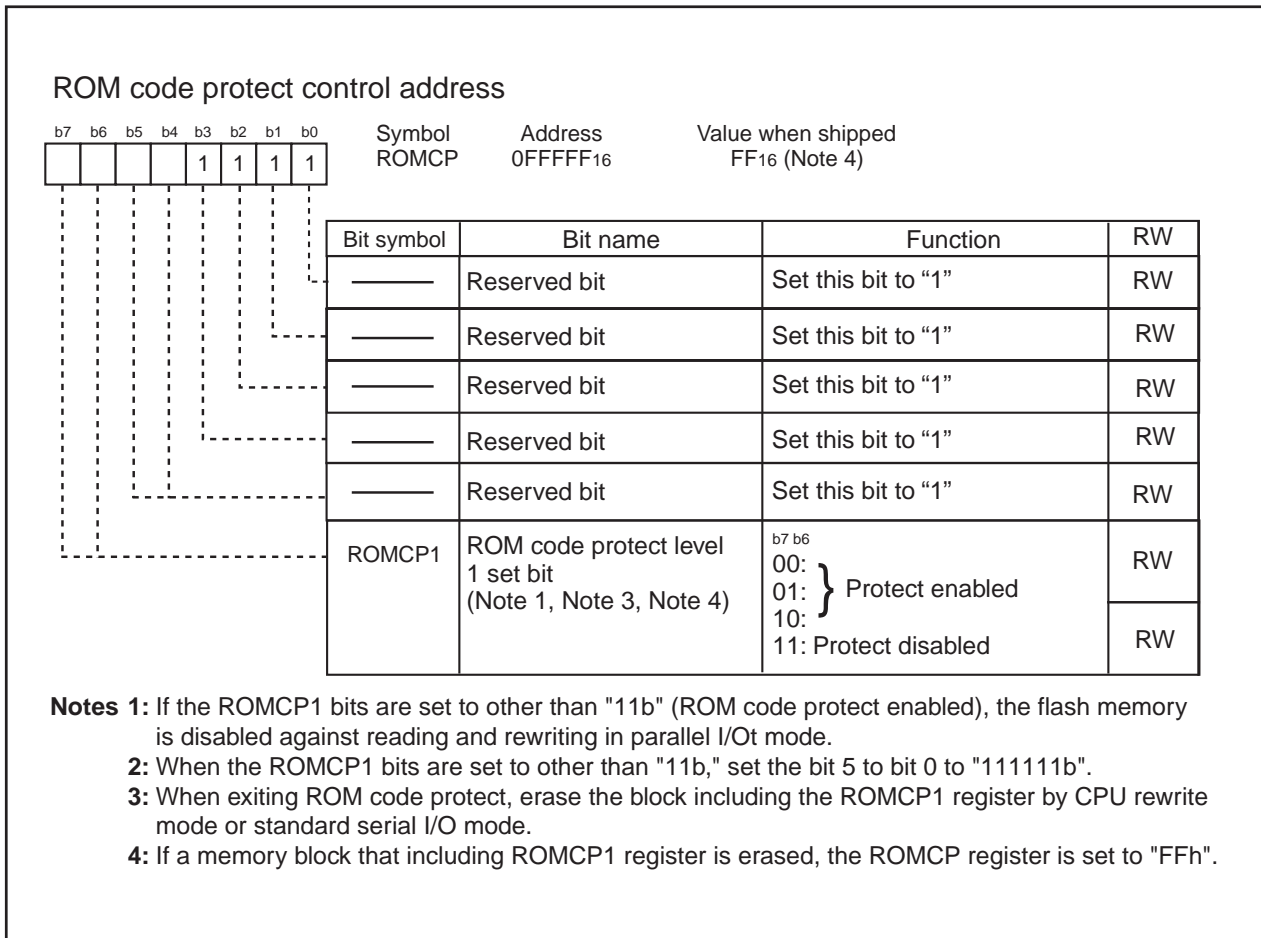


Figure 19.2. ROMCP Register

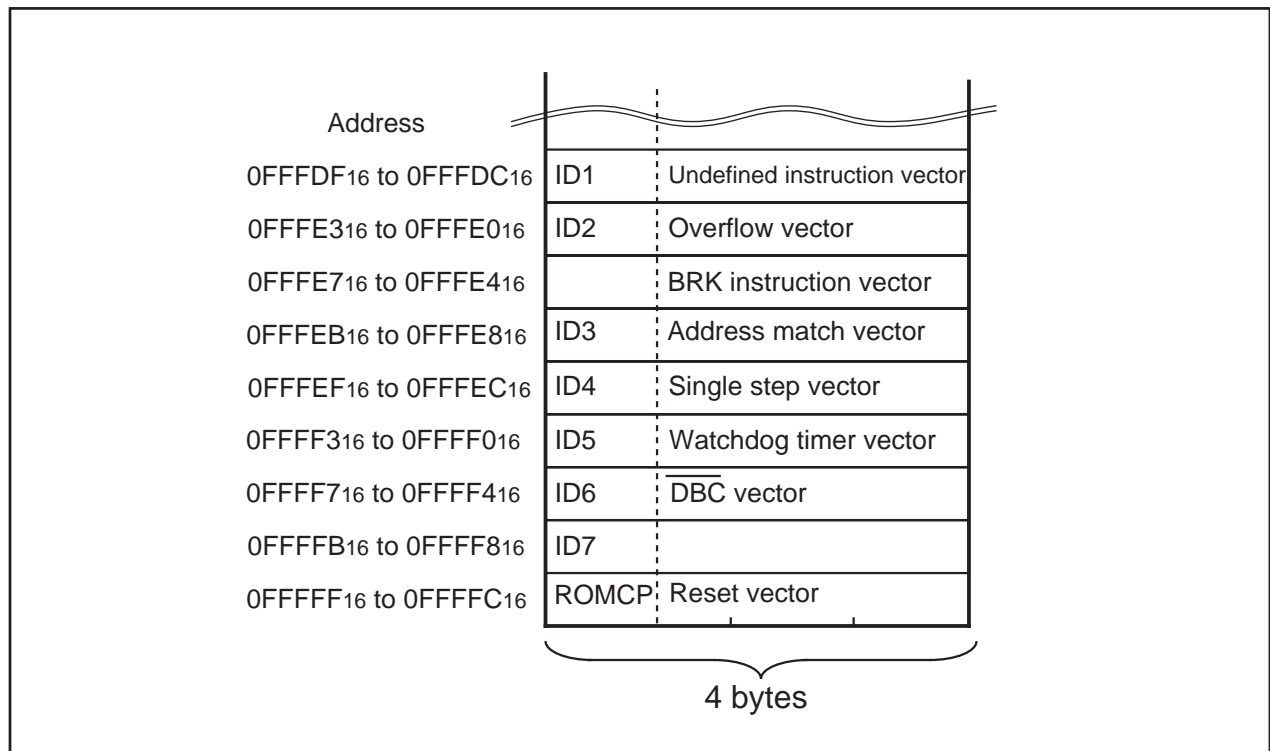


Figure 19.3. Address for ID Code Stored

CPU Rewrite Mode

In CPU rewrite mode, the user ROM area or OSD ROM area can be rewritten by executing software commands from the CPU. Therefore, the user ROM area or OSD 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 19.1 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 19.3 lists the differences between Erase Write 0 (EW0) and Erase Write 1 (EW1) modes.

Table 19.3. EW0 Mode and EW1 Mode

Item	EW0 mode	EW1 mode
Operation mode	<ul style="list-style-type: none"> • Single chip mode • Memory expansion mode • Boot mode 	Single chip mode
Areas in which a rewrite control program can be located	<ul style="list-style-type: none"> • User ROM area • Boot ROM area 	User ROM area
Areas in which a rewrite control program can be executed	Must be transferred to any area other than the flash memory (e.g., RAM) before being executed (Note 2)	Can be executed directly in the user ROM area
Areas which can be rewritten	User ROM area OSD ROM area	User ROM area OSD ROM area However, this does not include the area in which a rewrite control program exists
Software command limitations (Note 3)	None	<ul style="list-style-type: none"> • Program, Block Erase command Cannot be executed on any block in which a rewrite control program exists • 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) • Read Status Register command Cannot be executed
Modes after Program or Erase	Read Status Register mode	Read Array mode
CPU status during Auto Write and Auto Erase	Operating	Hold state (I/O ports retain the state in which they were before the command was executed) (Note 1)
Flash memory status detection (Note 3)	<ul style="list-style-type: none"> • Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program • Execute the Read Status Register command to read the status register's SR7, SR5, and SR4 flags. 	Read the FMR0 register's FMR00, FMR06, and FMR07 bits in a program

Note 1: Make sure no interrupts (except watchdog timer interrupts) and DMA transfers will occur.

Note 2: When in CPU rewrite mode, the PM10 and PM13 bits in the PM1 register are set to "1". The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1. When the PM13 bit = 0 and the flash memory is used in 4M-byte mode, the extended accessible area (50000₁₆ to BFFFF₁₆) cannot be used.

Note 3: The register name in explanatory note and a bit name are the cases of rewriting of user ROM area.

- **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 Control Register (FIDR, FMR0 and FMR1 registers)

Figure 19.4 shows the FIDR, FMR0 and FMR1 registers.

FMR00 Bit

This bit indicates the flash memory operating state. It is set to "0" while the program, block erase, erase all unlocked block, lock bit program, or read lock bit status command is being executed; otherwise, it is set to "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 is disabled by setting the FMR02 bit to "1" (lock bit disabled). (Refer to 22.3.6 Data Protect Function.) The lock bit is enabled by setting the FMR02 bit to "0" (lock bit enabled).

The FMR02 bit does not change the lock bit status but disables the lock bit function. If the block erase or erase all unlocked block command is executed when the FMR02 bit is set to "1", the lock bit status changes "0" (locked) to "1" (unlocked) after command execution is completed.

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. Setting the FMSTP bit to "1" makes the internal flash memory inaccessible. Set the FMSTP bit by program in a space 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 19.8 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 19.6 show the setting and resetting of EWO mode and 19.7 show the setting and resetting of EW1 mode, respectively.

FMR11 Bit

When the FMR11 bit is set to "0" (EW0 mode), the MCU (microcomputer) enters EW0 mode.

When the FMR11 bit is set to "1" (EW1 mode), the MCU (microcomputer) enters EW1 mode.

FMR16 Bit

This is a read-only bit indicating the execution result of the Read Lock Bit Status command.

While the block is locked, the FMR16 bit is set to "0". While the block is not locked, this bit is set to "1".

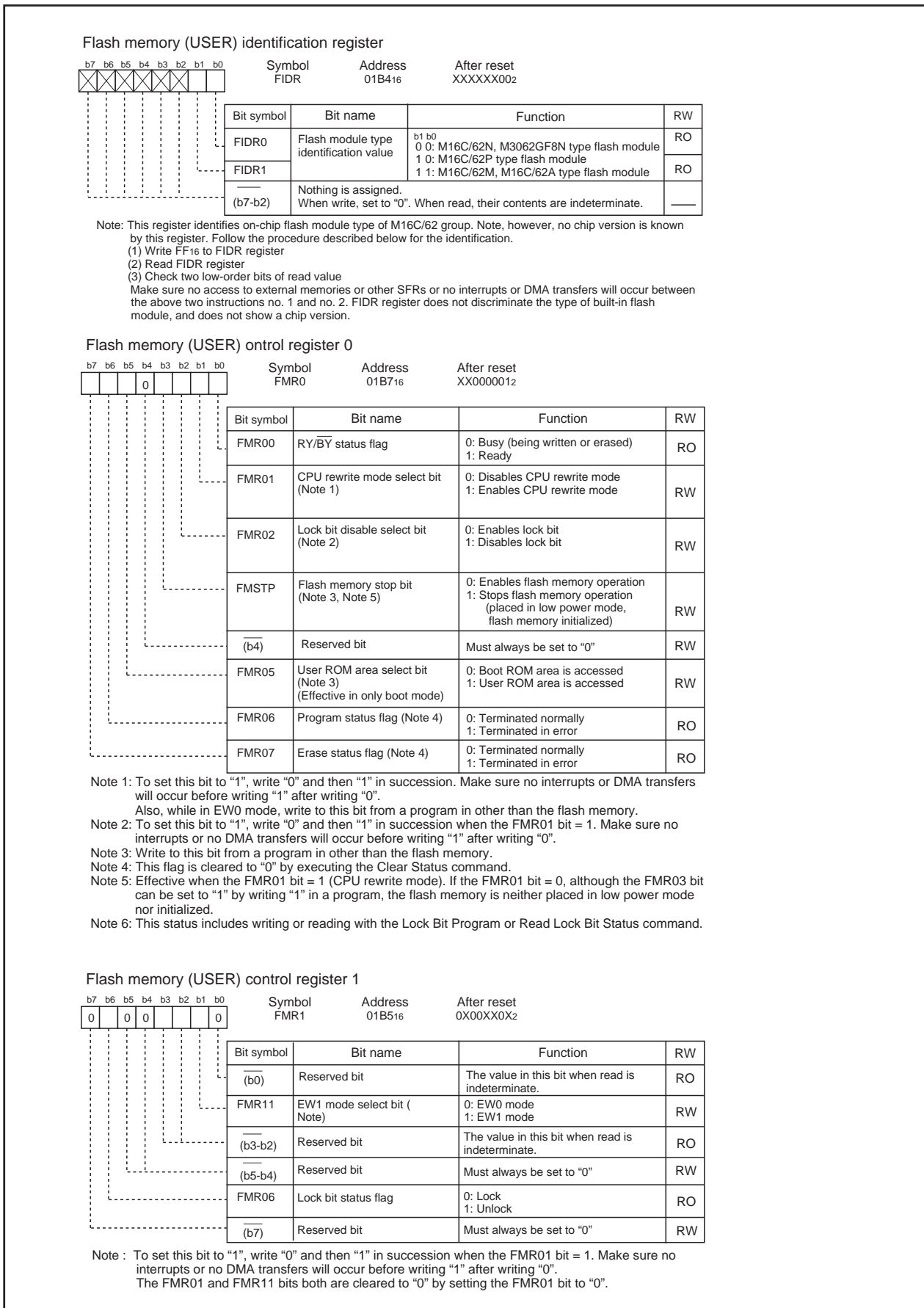


Figure 19.4. FIDR Register and FMR0 and FMR1 Registers

Figure 19.5 shows the register FMOSi0, figure 19.5-2 shows registers FMOSi1 and FMOSi4 (i=A, B).

FMOSi00 Bit

This bit indicates the operating status of the flash memory. The bit is "0" when the Program, Erase, or erase suspend mode is running; otherwise, the bit is "1".

FMOSi01 Bit

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

FMOSi02 Bit

When FMR02 bit is "0" (rewriting is forbidden,) block 0 and block 1 do not receive the command of a program and block erase.

FMOSiSTP 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. Setting the FMSTP bit to "1" makes the internal flash memory inaccessible. Therefore, FMSTP bit should program of domains other than a 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

Figure 19.8 shows a flow chart to be followed before and after entering low power mode.

When CPU rewriting mode shifts at stop mode or wait mode at the time of invalid, 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.

FMOSi06 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.

FMOSi07 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.

FMOSi11 Bit

Setting this bit to "1" places the microcomputer in EW1 mode.

FMOSi40 Bit

When FMR40 bit is set to "1" (permission), an erase suspension function will be permitted.

FMOSi41 Bit

In the EW0 mode, when FMR41 bit is set to "1" by the program, it will shift to erase suspension mode. In the EW1 mode, if the interruption demand of permitted interruption occurs, FMR41 bit will be automatically set to "1" (suspension request), and they will shift to erase suspension mode.

When resume automatic elimination operation, set FMR41 bit to "0" (erase restart.)

FMOSi46 Bit

FMR46 is set to "0" during automatic elimination execution. It is set to "1" by the inside of erase suspension mode. Between "0", access to a flash memory is prohibition.

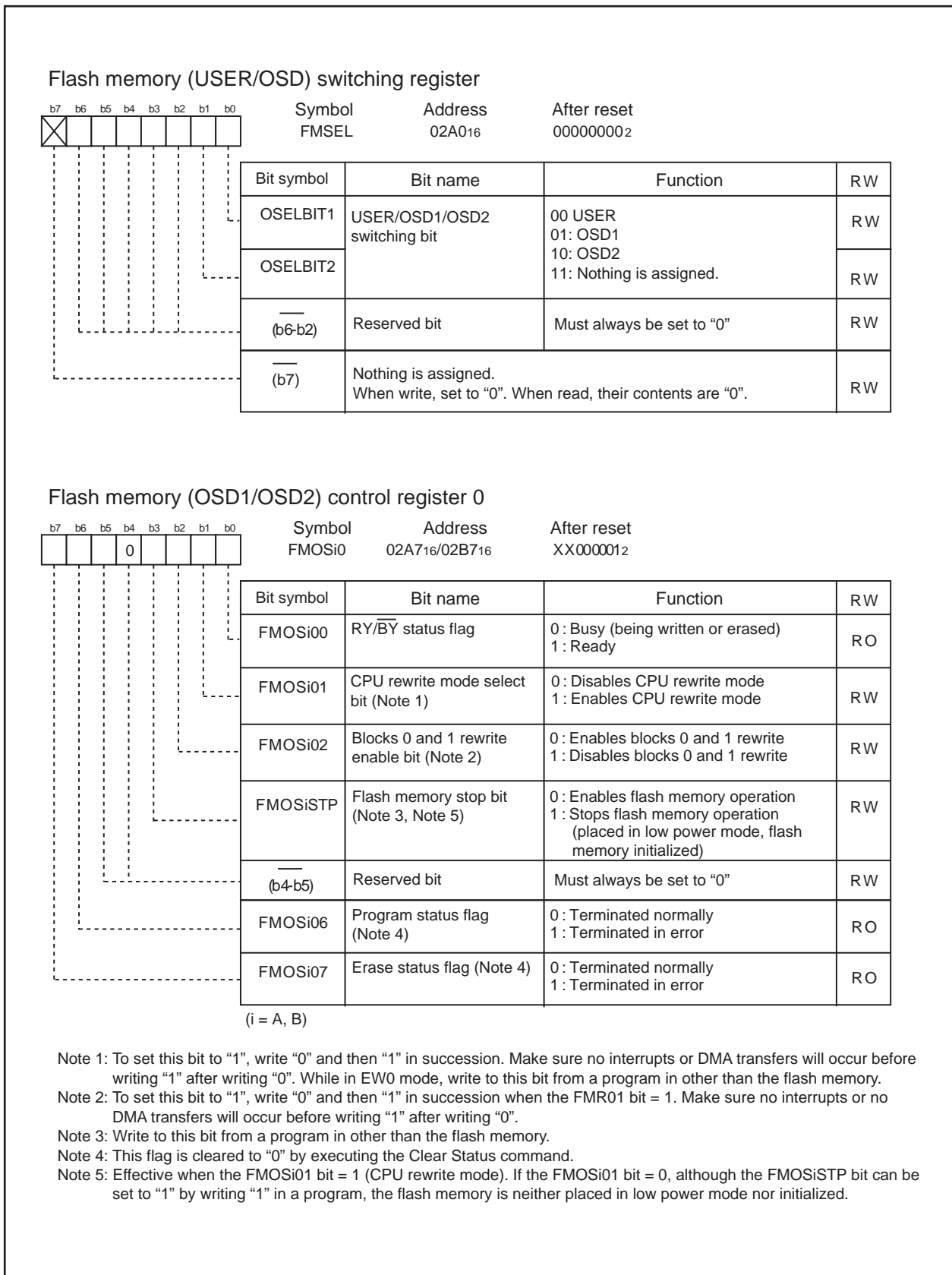


Figure 19.5-1. Register FMOSA0/FMOSB0/FMSEL

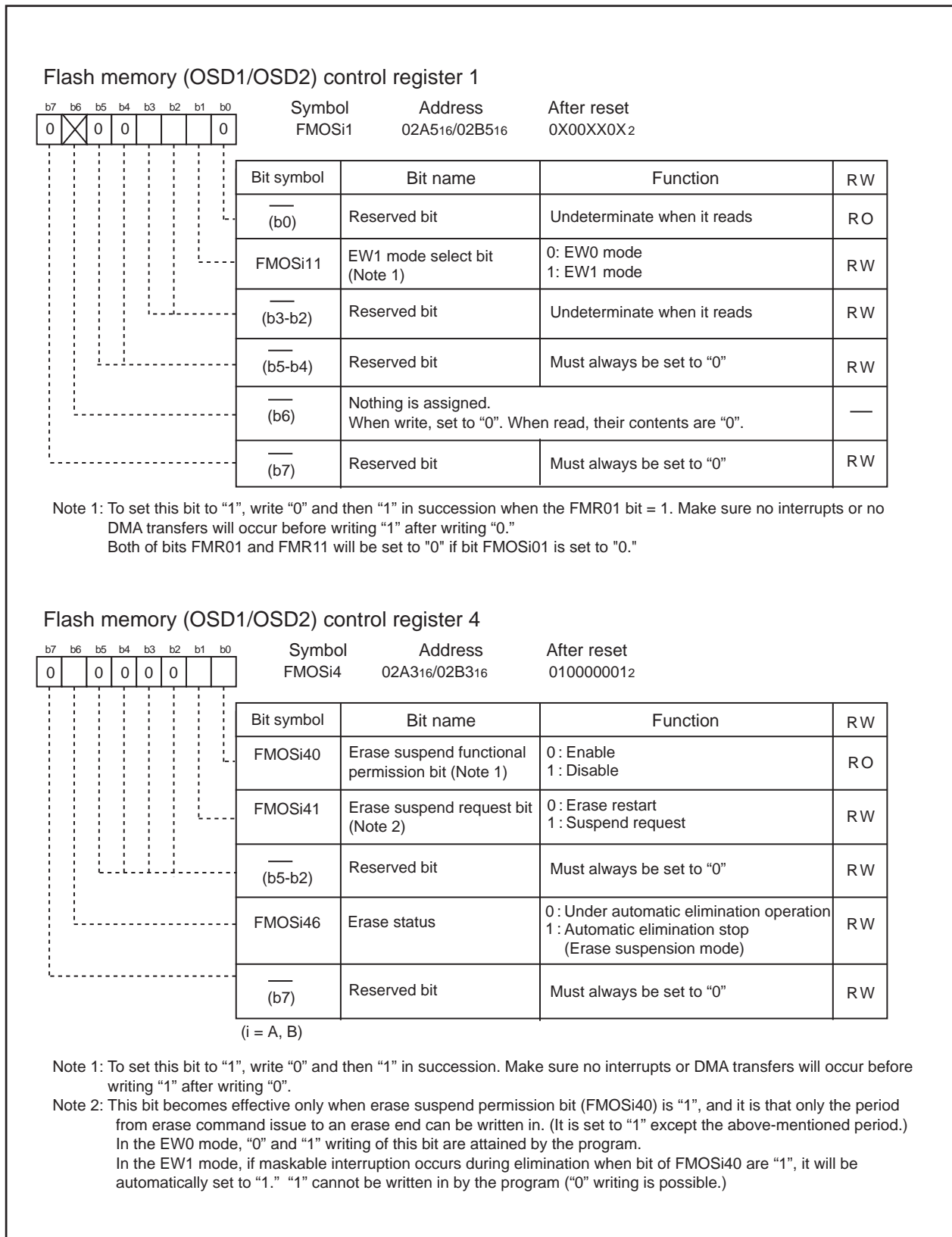


Figure 19.5-2. Registers FMOSA1, FMOSB1, FMOSA4 and FMOSB4

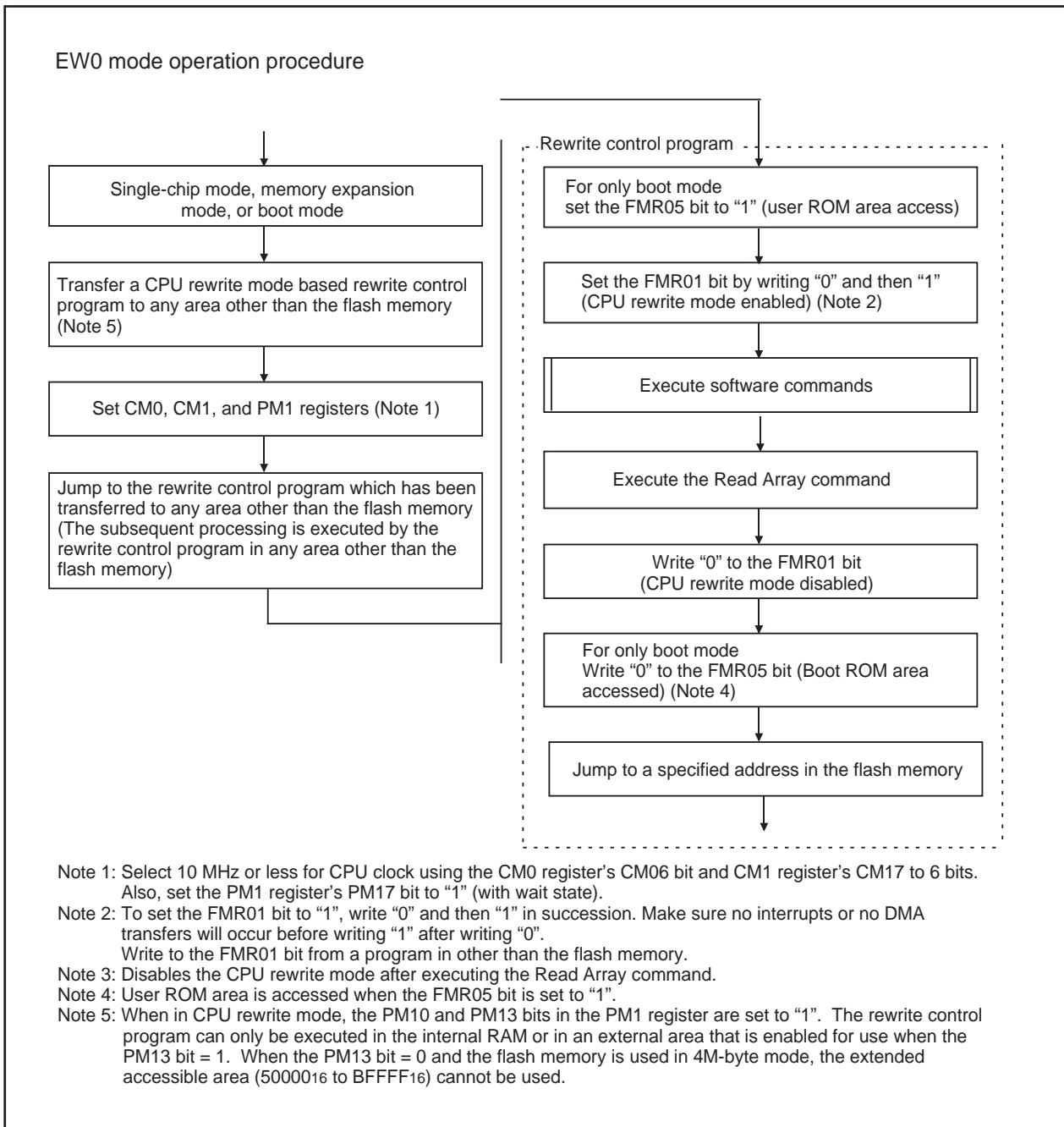


Figure 19.6. Setting and Resetting of EW0 Mode

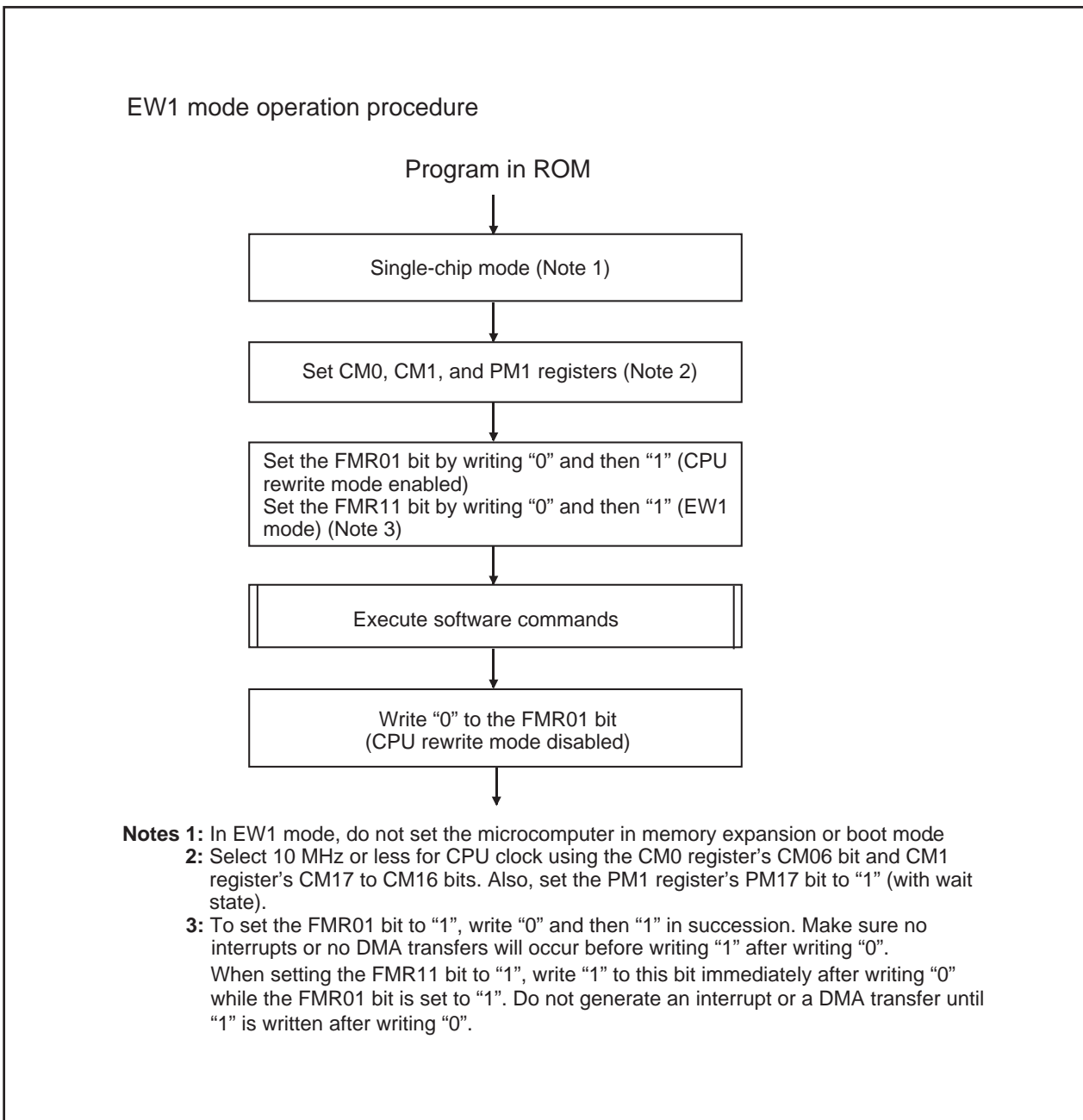


Figure 19.7. Setting and Resetting of EW1 Mode

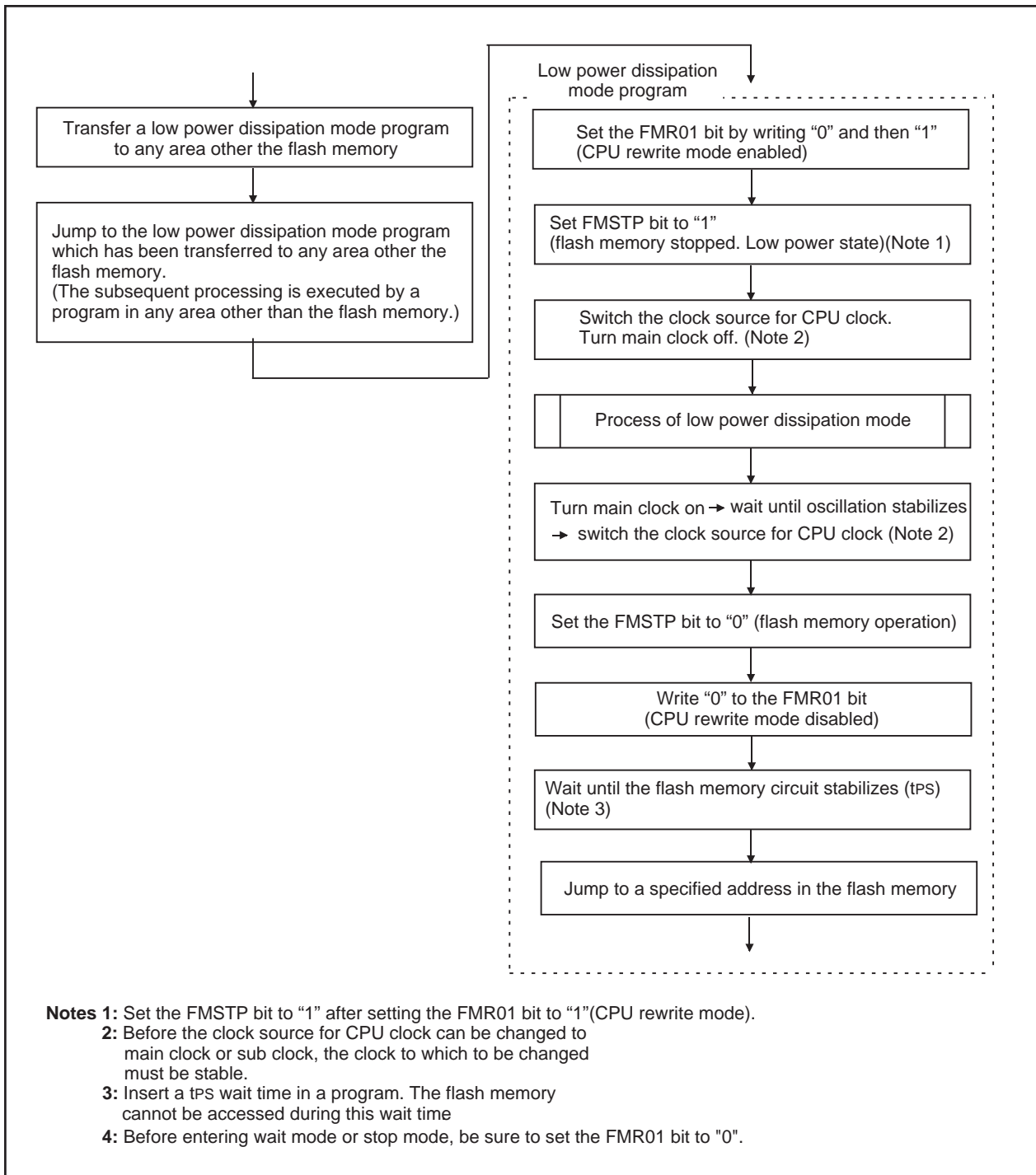


Figure 19.8. Processing Before and After Low Power Dissipation Mode

Precautions on CPU Rewrite Mode

(1) Operation Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), select 10 MHz or less for CPU clock 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) Interrupt (EW0 Mode)

- Any interrupt which has a vector in the variable vector table can be used providing that its vector is transferred into the RAM area.
- The watchdog timer 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.
When a monitor timer interrupt is generated, the rewriting operation ends. Execute the rewriting program again after an interrupt routine ends.
- The address match interrupt cannot be used because the flash memory's internal data is referenced.

(4) Interrupt (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.

(5) 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".

(6) Rewrite 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.

(7) Rewrite user ROM area (EW1 Mode)

- Avoid rewriting any block in which the rewrite control program is stored.

(8) 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).

(9) Writing Command and Data

Write the command code and data at even addresses.

(10) Wait Mode

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

(11) 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

(12) 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 command
- Read lock bit status

Software Commands

Software commands are described below. The command code and data must be read and written in 16-bit units, to and from even addresses in the user ROM area. When writing command code, the 8 high-order bits (D15–D8) are ignored.

Table 19.4. Software Commands

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read array	Write	X	xxFF ₁₆			
Read status register	Write	X	xx70 ₁₆	Read	X	SRD
Clear status register	Write	X	xx50 ₁₆			
Program	Write	WA	xx40 ₁₆	Write	WA	WD
Block erase	Write	X	xx20 ₁₆	Write	BA	xxD0 ₁₆
Erase all unlocked block ^(Note 1, 2)	Write	X	xxA7 ₁₆	Write	X	xxD0 ₁₆
Lock bit program (Note 2)	Write	BA	xx77 ₁₆	Write	BA	xxD0 ₁₆
Read lock bit status (Note 2)	Write	X	xx71 ₁₆	Write	BA	xxD0 ₁₆

Note 1: 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.

Note 2: It can perform only to USER area. The execution to OSD area serves as error.

SRD: Status register data (D7 to D0)

WA: Write address (Make sure the address value specified in the the first bus cycle is the same 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

xx: High-order 8 bits of command code (ignored)

Read Array Command (FF₁₆)

This command reads the flash memory.

Writing 'xxFF₁₆' in the first bus cycle places the microcomputer in read array mode. Enter the read address in the next or subsequent bus cycles, and the content of the specified address can be read in 16-bit units.

Because the microcomputer remains in read array mode until another command is written, the contents of multiple addresses can be read in succession.

Read Status Register Command (70₁₆)

This command reads the status register.

Write 'xx70₁₆' in the first bus cycle, and the status register can be read in the second bus cycle. (Refer to "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.

Clear Status Register Command (5016)

The clear status register command clears the status register. By writing "xx50h" in the first bus cycle, the FMR07 to FMR06 bits in the FMR0 register are set to "00b" and the SR5 to SR4 bits in the status register are set to "00b".

Program Command (4016)

This command writes data to the flash memory in 1 word (2 byte) units.

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. 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.")

Additional writing to the programmed address cannot be performed. Figure 19.9 shows the program flow chart. Also, each block can disable a program by the lock bit.

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.

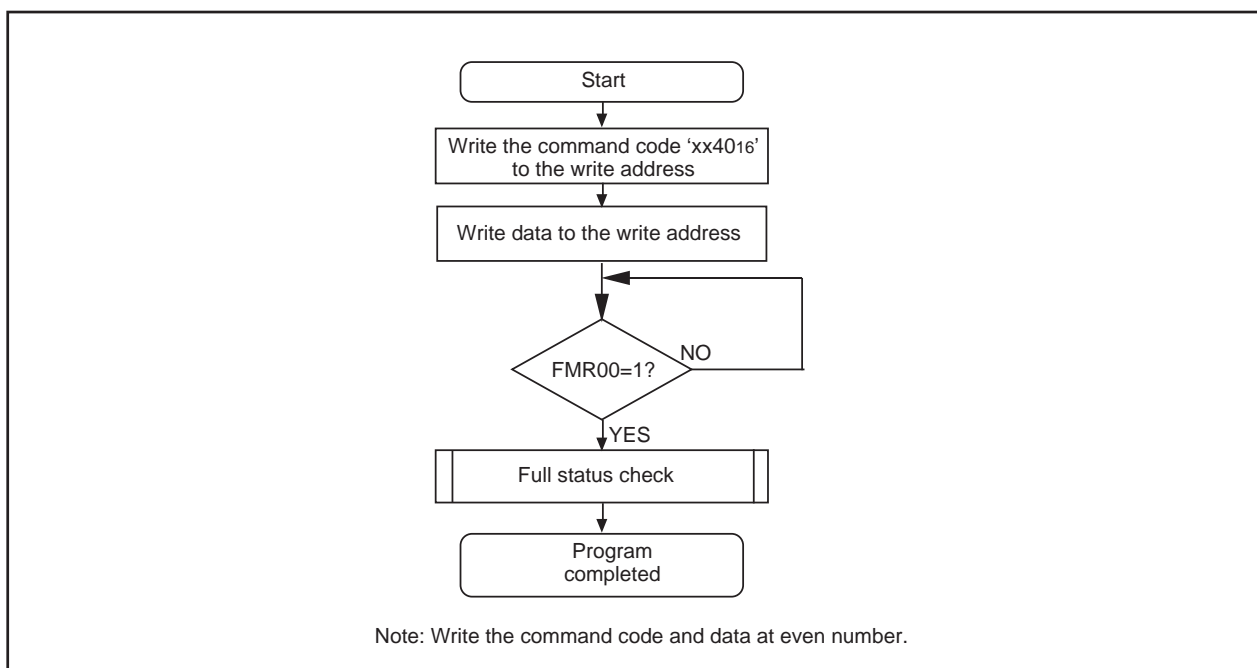


Figure 19.9. Program Command

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" (busy) during auto erasing and set to "1" (ready) 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 19.10 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.

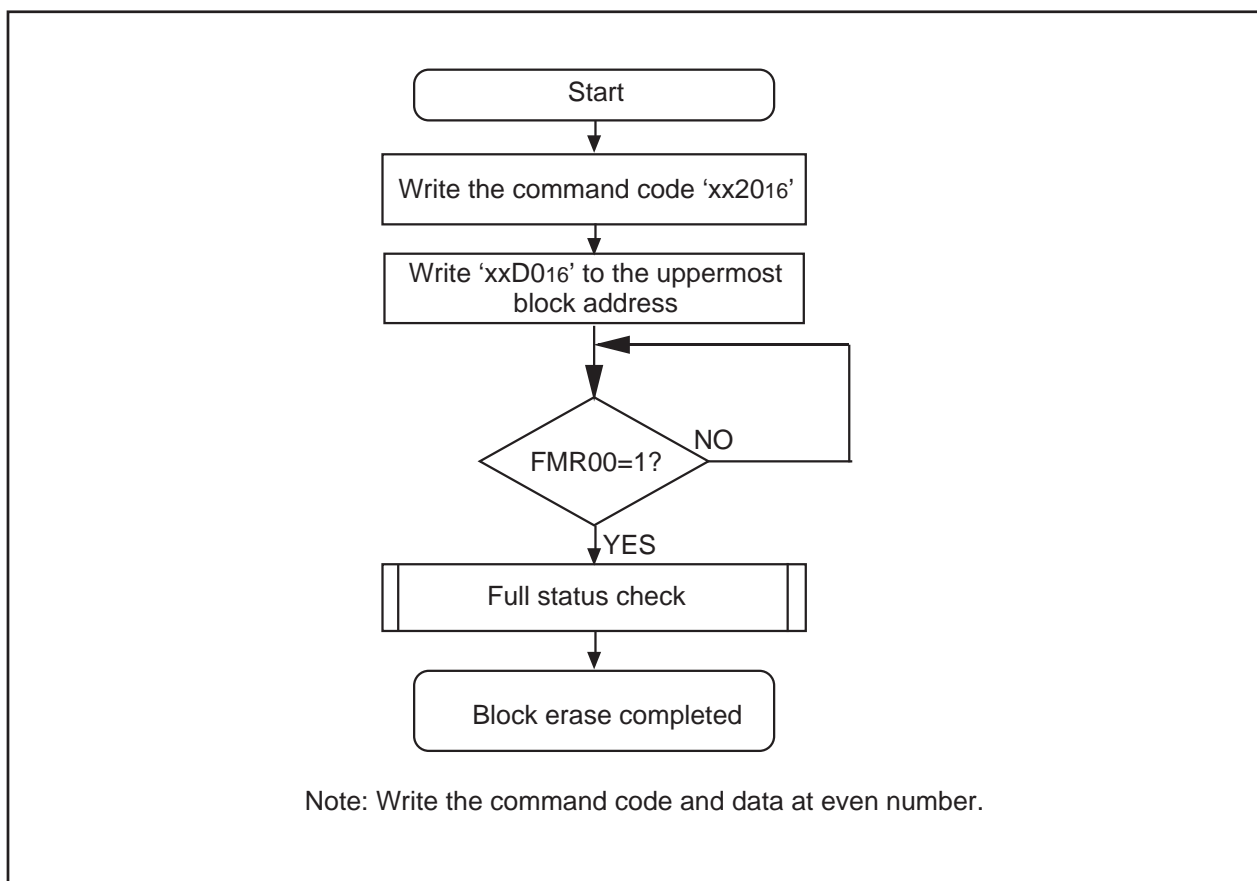


Figure 19.10. Block Erase Command

Erase All Unlocked Block (User area only)

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" (busy) at the same time auto erasing starts, and set back to "1" (ready) 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 (User area only)

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 2.11 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.

Refer to "Data protect function" for the lock bit function and how to set the lock bit to "1" (unlocked status).

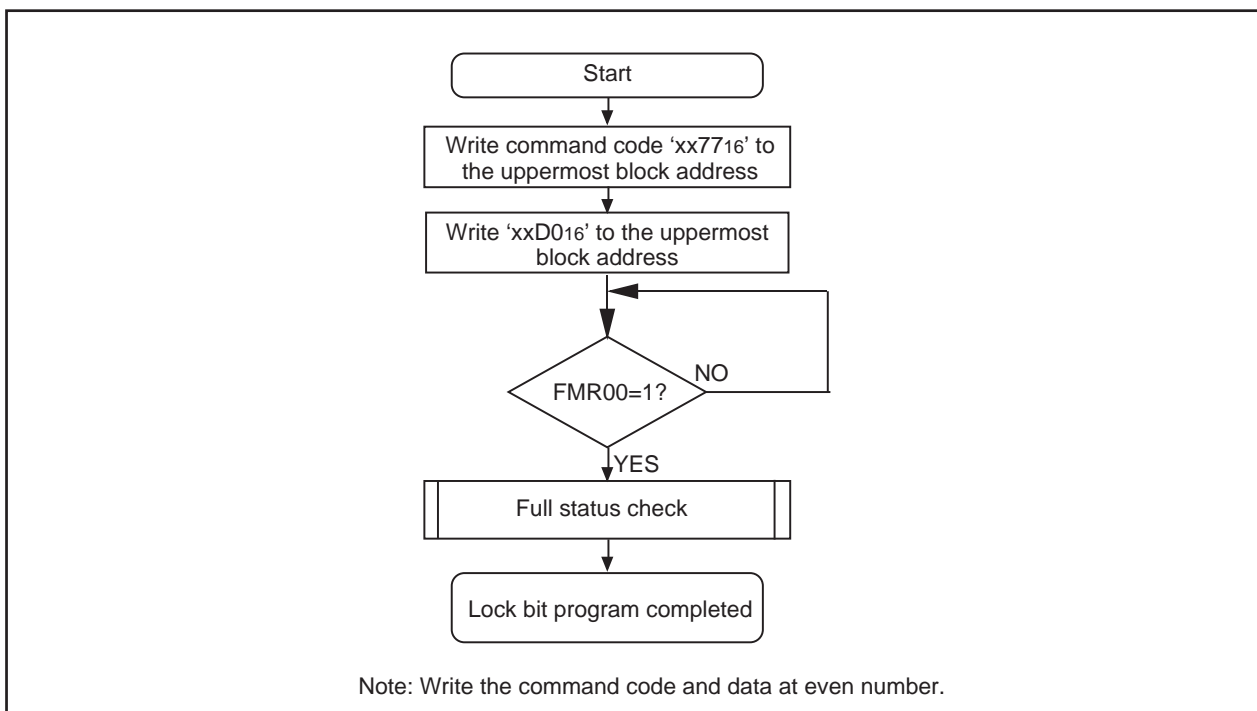


Figure 19.11. Lock Bit Program Command

Read Lock Bit Status Command (User area only)

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 19.12 shows an example of a read lock bit status flowchart.

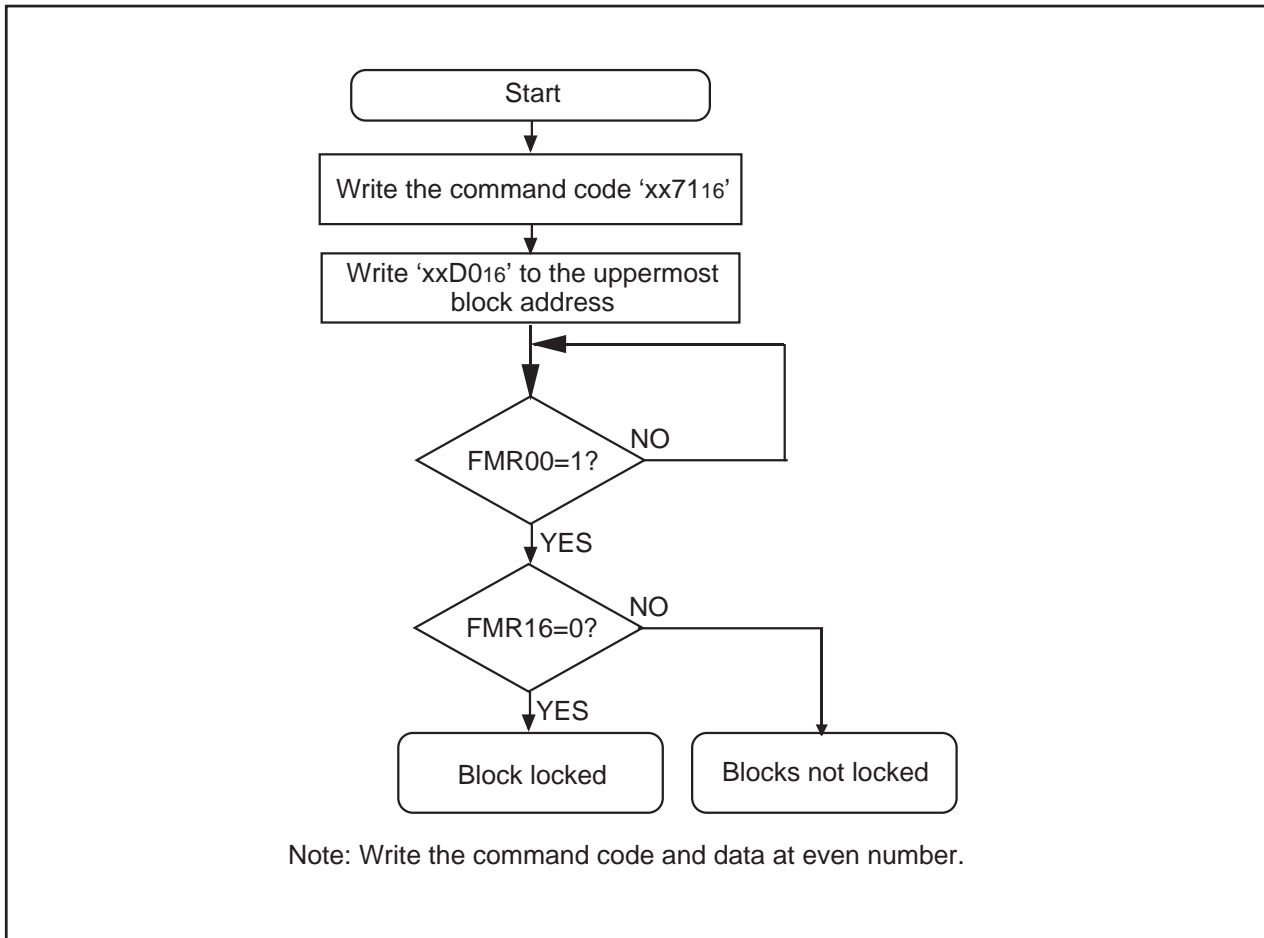


Figure 19.12. Read Lock Bit Status Command

Data Protect Function (User area only)

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 bit 0, bit 6 and bit 7 of the FMR0, FMOSA0 and FMOSB0 registers.

A status register exists in each of three areas of USER/OSD1/OSD2. In order to read the right result, it is necessary to choose an object area by FMSEL0 and FMSEL1 before command execution.

Table 19.5 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/FMOSA00/FMOSB00 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/FMOSA07/FMOSB07 Bits)

Refer to “Full Status Check.”

Program Status (SR4 and FMR06/FMOSA06/FMOSB06 Bits)

Refer to “Full Status Check.”

Table 19.5. Status Register

Status register bit	Flash memory control register0	Status name	Contents		Value after reset
			"0"	"1"	
SR7 (D7)	bit0	Sequencer status	Busy	Ready	1
SR6 (D6)	—	Reserved bit	-	-	—
SR5 (D5)	bit7	Erase status	Terminated normally	Terminated in error	0
SR4 (D4)	bit6	Program status	Terminated normally	Terminated in error	0
SR3 (D3)	—	Reserved bit	-	-	—
SR2 (D2)	—	Reserved bit	-	-	—
SR1 (D1)	—	Reserved bit	-	-	—
SR0 (D0)	—	Reserved bit	-	-	—

- D0 to D7: Indicates the data bus which is read out when the Read Status Register command is executed.
- SR5 and SR4 are cleared to "0" by executing the Clear Status Register command.
- When SR5 or SR4 = 1, the Program, Block Erase, Erase All Unlocked Block, and Lock Bit Program commands are not accepted.
- Flash memory control register exists independently to each area of USER, OSD1, and OSD2, and serves as FMR0, FMOSA0, and FMOSB0, respectively.

Full Status Check

When an error occurs, the bit 6 to 7 of the Flash memory control register are set to “1”, indicating occurrence of each specific error. Therefore, execution results can be verified by checking these status bits (full status check). Table 19.6 lists errors and FMR0, FMOSA0, and FMOSB0 register status. Figure 19.13 shows a full status check flowchart and the action to be taken when each error occurs.

Table 19.6. Errors and the Flash Memory Control Register Status

Flash memory control register (status register) status		Error	Error occurrence condition
(SR5)	(SR4)		
1	1	Command sequence error	<ul style="list-style-type: none"> • When any command is not written correctly • When invalid data was written other than those that can be written in the second bus cycle of the Lock Bit Program, Block Erase, or Erase All Unlocked Block command (i.e., other than 'xxD016' or 'xxFF16') (Note 1)
1	0	Erase error	<ul style="list-style-type: none"> • When the Block Erase command was executed on locked blocks (Notes 2, 3) • When the Block Erase or Erase All Unlocked Block command was executed on unlocked blocks but the blocks were not automatically erased correctly
0	1	Program error	<ul style="list-style-type: none"> • When the Block Erase command was executed on locked blocks (Notes 2, 3) • When the Program command was executed on unlocked blocks but the blocks were not automatically programmed correctly. • When the Lock Bit Program command was executed but not programmed correctly (Note 3)

Note 1: Writing 'xxFF16' in the second bus cycle of these commands places the microcomputer in read array mode, and the command code written in the first bus cycle is nullified.

Note 2: When the O2 bit = 1 (lock bit disabled), no error will occur under this condition.

Note 3: It does not correspond, when it performs to OSD1 or OSD2 area, and error is not generated.

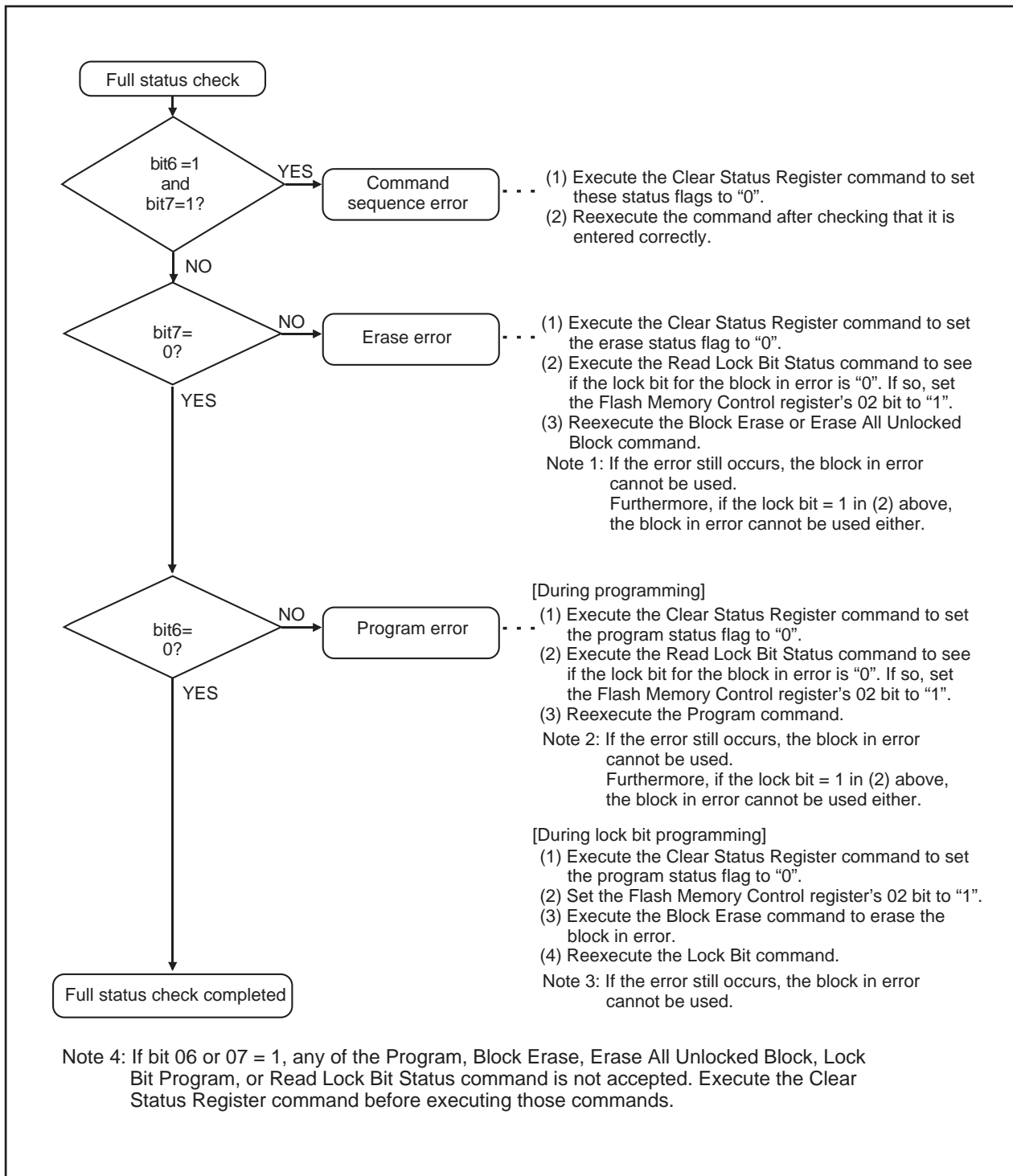


Figure 19.13. Full Status Check and Handling Procedure for Each Error

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/6V8 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 19.7 lists pin functions (flash memory standard serial input/output mode). Figures 19.14 to 19.16 show pin connections for standard 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 19.7. Pin Functions (Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Description
VCC1, VCC2, VCC3, VSS	Power input		Please input the guarantee voltage of program erase into pins VCC1, VCC2, and VCC3. Please input 0V into VSS.
CNVSS1, CNVSS2	CNVSS	I	Please connect CNVSS1 to Vcc and connect CNVSS2 to Vss.
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 input	I	Connect this pin to Vss.
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/RTS1	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitors the boot program operation check signal output pin.
P65/CLK1	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input "L".
P66/RxD1	RxD input	I	Serial data input pin.
P67/TxD1	TxD output	O	Serial data output pin. (Note 1)
P70 to P77	Input port P7	I	Input "H" or "L" level signal or open.
P82 to P83, P86, P87	Input port P8	I	Input "H" or "L" level signal or open.
P90 to P91	Input port P9	I	Input "H" or "L" level signal or open.
P103 to P107	Input port P10	I	Input "H" or "L" level signal or open.
Other input pins			Input "H" or "L" level signal.
Other output pins			Please open.

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 VCC1 via a resistor. Because this pin is directed for data output after reset, adjust the pull-up resistance value in the system so that data transfers will not be affected.

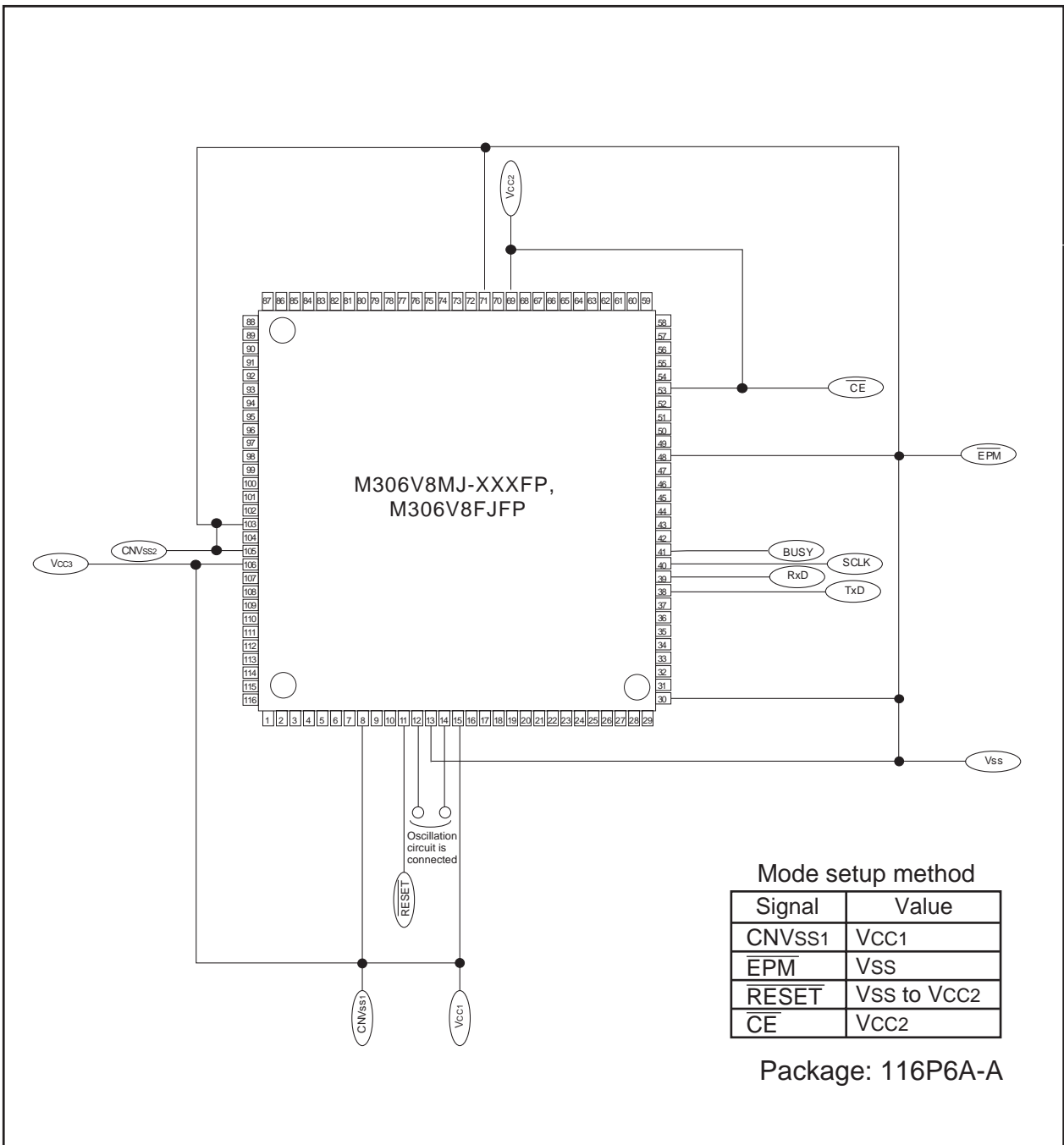


Figure 19.14. Pin circuit at the time of standard serial I/O mode

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

Figure 19.15 and 19.16 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.

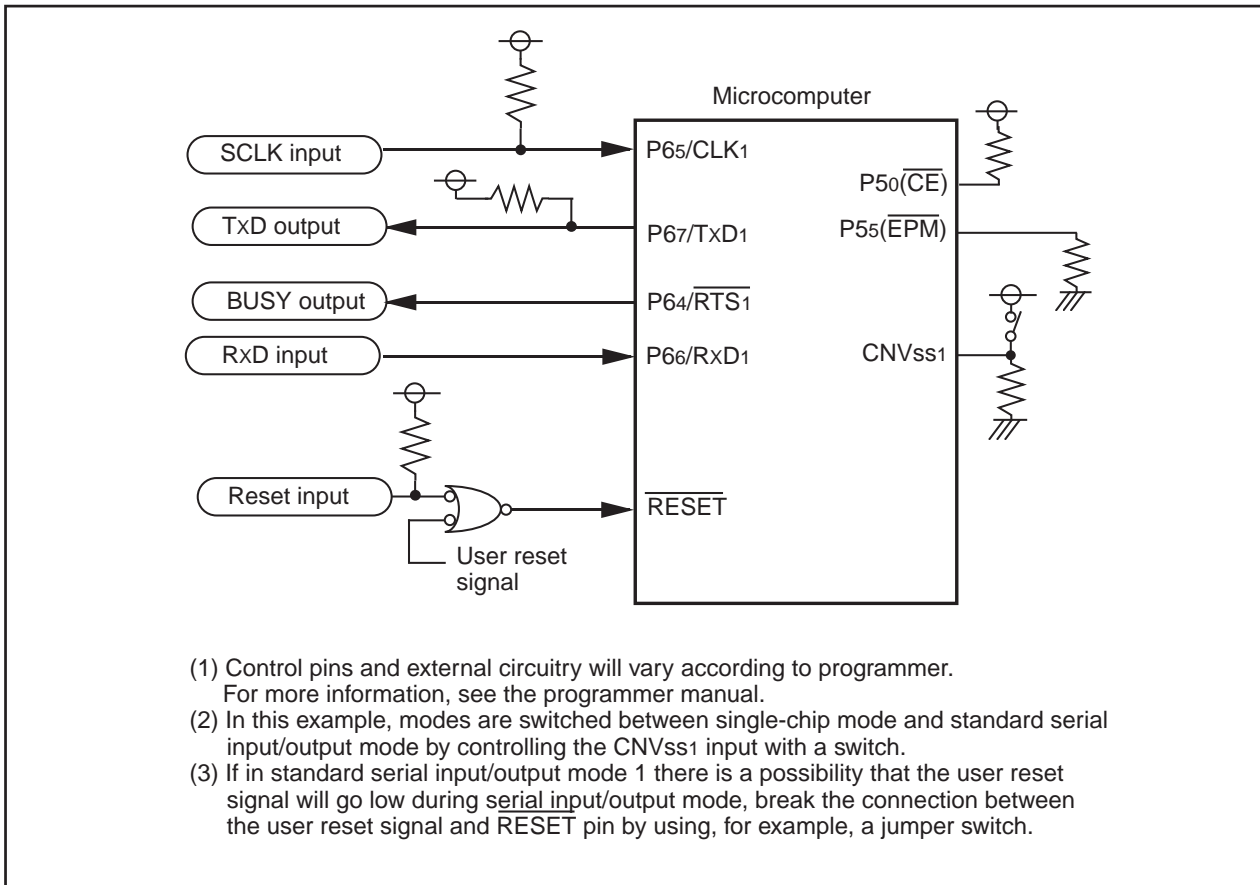


Figure 19.15. Circuit Application in Standard Serial I/O Mode 1

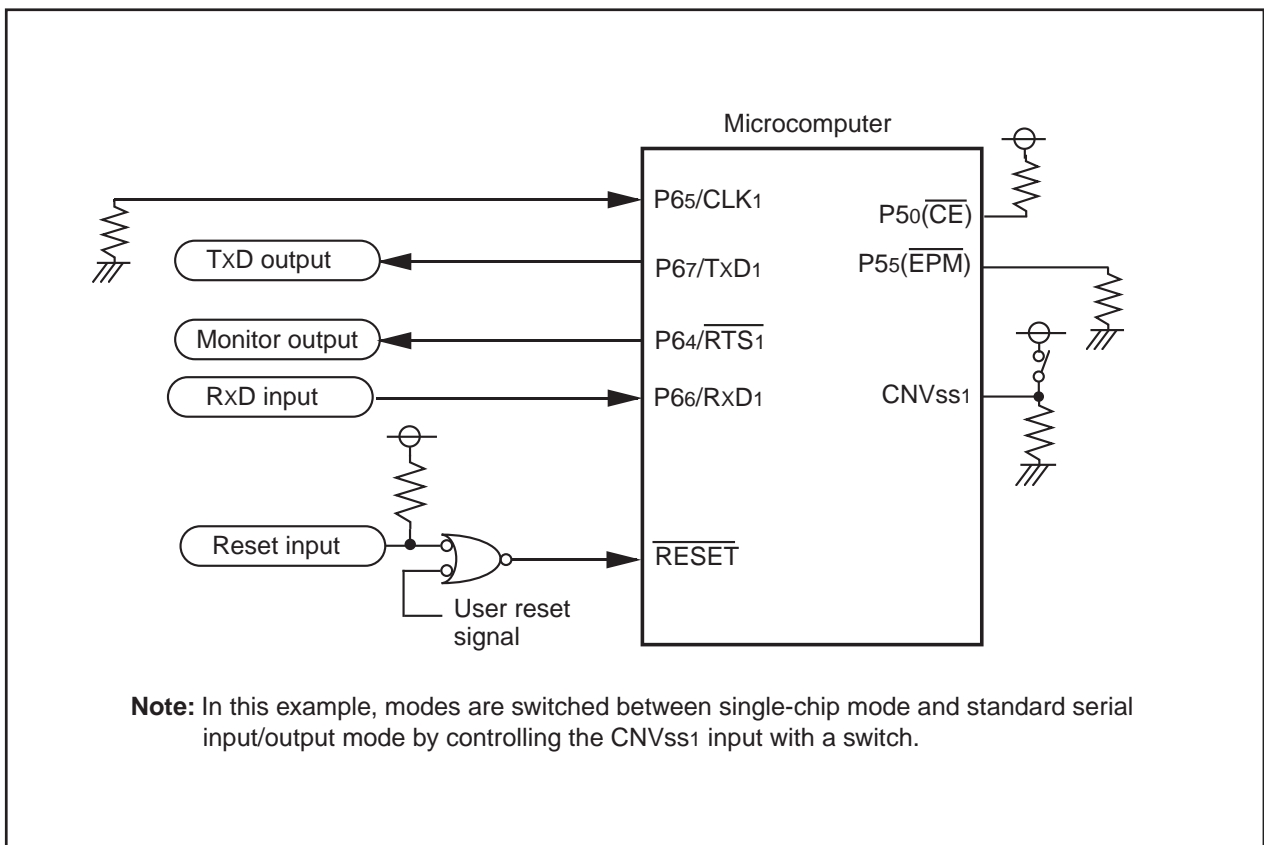


Figure 19.16. Circuit Application in Standard Serial I/o Mode 2

Parallel I/O Mode

In parallel input/output mode, the user ROM, OSD ROM, and boot ROM areas can be rewritten by using a parallel programmer suitable for the M16C/6V8 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.

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 0FF000₁₆ to 0FFFFFF₁₆. When rewriting the boot ROM area, make sure that only this address range is rewritten. (Do not access other than the addresses 0FF000₁₆ to 0FFFFFF₁₆.)

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.)

Usage Precaution

Reset

When supplying power to the microcomputer, the power supply voltage applied to the VCC1 pin must meet the conditions of SVCC.

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
SVcc	Power supply rising gradient (Vcc1)	0.05			V/ms

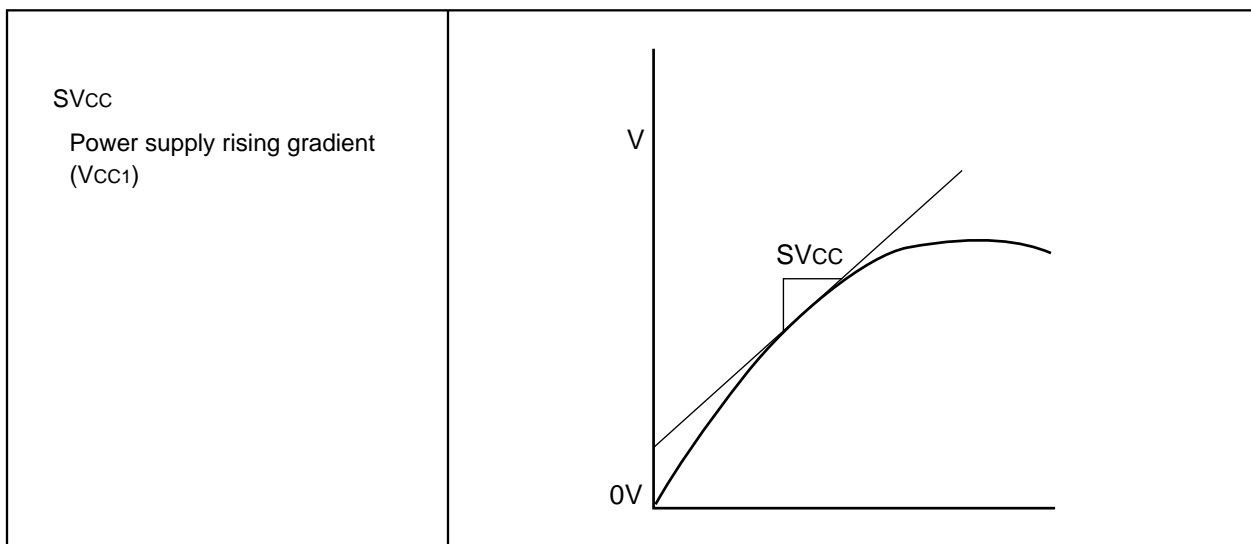


Figure 20.1 Timing of SVcc

Bus

- The ROMless version can operate only in the microprocessor mode, connect the CNVss1 pin to VCC1.
- When resetting CNVss1 pin with "H" input, contents of internal ROM cannot be read out.

Power Control

- When exiting stop mode by hardware reset, set $\overline{\text{RESET}}$ pin to “L” until a main clock oscillation is stabilized.
- Set the MR0 bit in the TAI_{MR} register (i=0 to 4) to “0” (pulse is not output) to use the timer A to exit stop mode.
- 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.
- Wait the main clock oscillation stabilizes, before switching the clock source for CPU clock to the main clock.
Similarly, wait until the sub clock oscillates stably before switching the clock source for CPU clock to the sub clock.
- Suggestions to reduce power consumption

Ports

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A pass current flows in input ports that high-impedance state. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

A/D converter

When A/D conversion is not performed, set the VCUT bit of ADI_{CON1} register to “0” (no VREF connection). When A/D conversion is performed, start the A/D conversion at least 1 μ s or longer after setting the VCUT bit to “1” (VREF connection).

Stopping peripheral functions

Use the CM0 register CM02 bit to stop the unnecessary peripheral functions during wait mode. However, because the peripheral function clock (f_{C32}) generated from the sub-clock does not stop, this measure is not conducive to reducing the power consumption of the chip. If low speed mode or low power dissipation mode is to be changed to wait mode, set the CM02 bit to “0” (do not peripheral function clock stopped when in wait mode), before changing wait mode.

Switching the oscillation-driving capacity

Set the driving capacity to “LOW” when oscillation is stable.

Changing the Interrupt Generate Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to "1" (interrupt requested). If you changed the interrupt generate factor for an interrupt that needs to be used, be sure to clear the IR bit for that interrupt to "0" (interrupt not requested).

Changing the interrupt generate factor referred to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the generate factor, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to "0" (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.

Figure 20.2 shows the procedure for changing the interrupt generate factor.

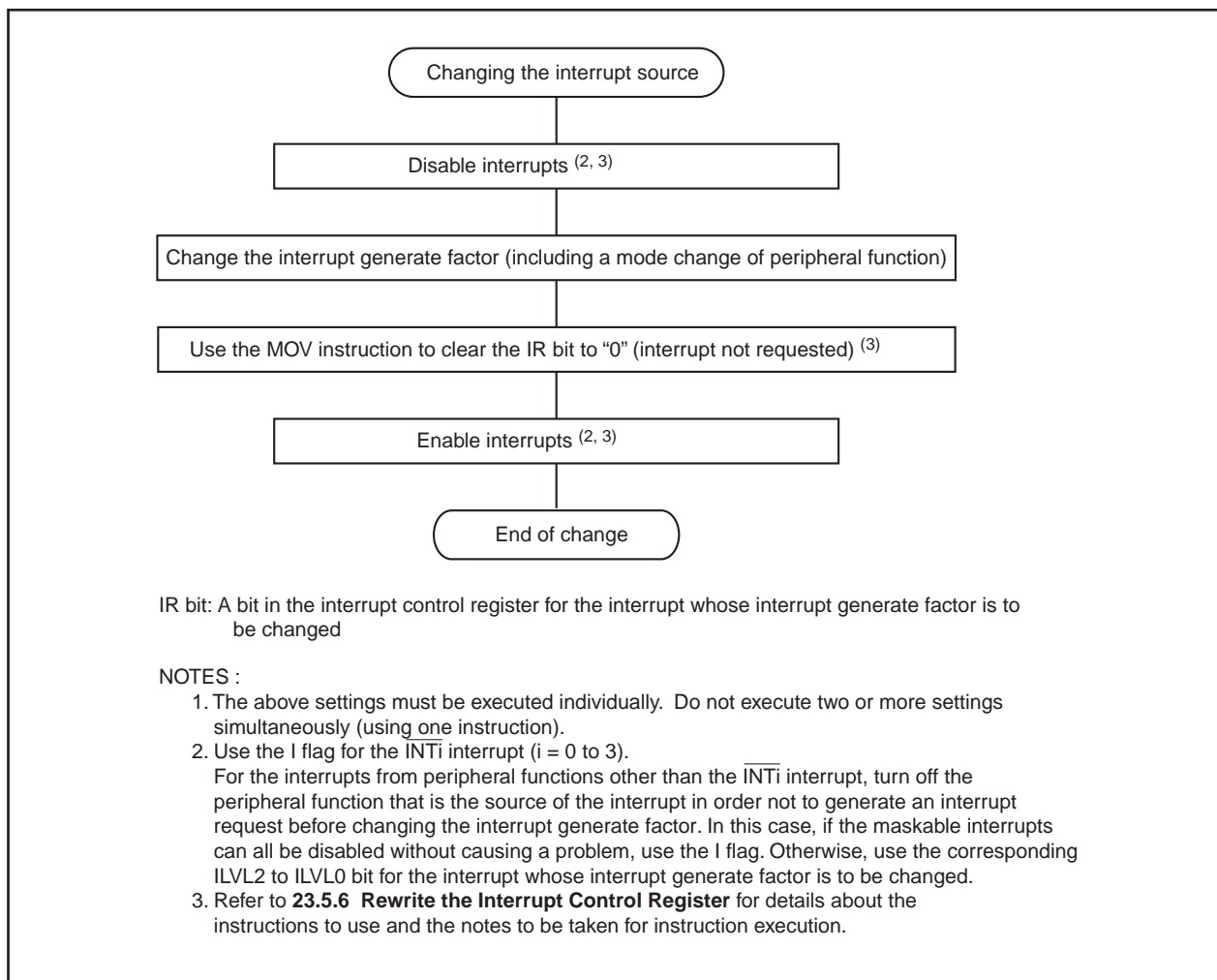


Figure 20.2 Procedure for Changing the Interrupt Generate Factor

DMAC

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 the DMiCON register simultaneously⁽¹⁾.

Step 2: Make sure that the DMAi is in an initial state⁽²⁾ in a program.

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

NOTES :

1. The DMAS bit remains unchanged even if "1" is written. However, if "0" is written to this bit, it is set to "0" (DMA not requested). In order to prevent the DMAS bit from being modified to "0," "1" should be written to the DMAS bit when "1" is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.
Similarly, when writing to the DMAE bit with a read-modify-write instruction, "1" should be written to the DMAS bit in order to maintain a DMA request which is generated during execution.
2. Read the TCRi register to verify whether the DMAi is in an initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in an initial state. (If a DMA request occurs after writing to the DMAE bit, the value written to the TCRi register is "1".) If the read value is a value in the middle of transfer, the DMAi is not in an initial state.

Timers

Timer A

(1) Timer A (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_{MR} (i = 0 to 4) register and the TAI register before setting the TAI_S bit in the TABSR register to "1" (count starts).

Always make sure the TAI_{MR} register is modified while the TAI_S bit remains "0" (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, if the counter is read at the same time it is reloaded, the value "FFFFh" is read. Also, if the counter is read before it starts counting after a value is set in the TAI register while not counting, the set value is read.

(2) Timer A (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_{MR} (i = 0 to 4) register, the TAI register, the UDF register, the ONSF register TAZIE, TA0TGL and TA0TGH bits and the TRGSR register before setting the TAI_S bit in the TABSR register to "1" (count starts).

Always make sure the TAI_{MR} register, the UDF register, the TAZIE, TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register are modified while the TAI_S bit remains "0" (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAI register. However, "FFFFh" can be read in underflow, while reloading, and "0000h" in overflow. When setting TAI register to a value during a counter stop, the setting value can be read before a counter starts counting. Also, if the counter is read before it starts counting after a value is set in the TAI register while not counting, the set value is read.

(3) Timer A (One-shot Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAI_iMR (i = 0 to 4) register, the TAI_i register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAI_iS bit in the TABSR register to "1" (count starts).

Always make sure the TAI_iMR register, the TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAI_iS bit remains "0" (count stops) regardless whether after reset or not.

When setting TAI_iS bit to "0" (count stop), the followings occur:

- A counter stops counting and a content of reload register is reloaded.
- TAI_iOUT pin outputs "L".
- After one cycle of the CPU clock, the IR bit in the TAI_iC register is set to "1" (interrupt request).

Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger has been selected, one-cycle delay of a count source as maximum occurs between a trigger input to TAI_iIN pin and output in one-shot timer mode.

The IR bit is set to "1" when timer operation mode is set with any of the following procedures:

- Select one-shot timer mode after reset.
- Change an operation mode from timer mode to one-shot timer mode.
- Change an operation mode from event counter mode to one-shot timer mode.

To use the Timer A_i interrupt (the IR bit), set the IR bit to "0" after the changes listed above have been made.

When a trigger occurs, while counting, a counter reloads the reload register to continue counting after generating a re-trigger and counting down once. To generate a trigger while counting, generate a second trigger between occurring the previous trigger and operating longer than one cycle of a timer count source.

(4) Timer A (Pulse Width Modulation Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAIMR ($i = 0$ to 4) register, the TAI register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAI S bit in the TABSR register to "1" (count starts).

Always make sure the TAIMR register, TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAI S bit remains "0" (count stops) regardless whether after reset or not.

The IR bit is set to "1" when setting a timer operation mode with any of the following procedures:

- Select the PWM mode after reset.
- Change an operation mode from timer mode to PWM mode.
- Change an operation mode from event counter mode to PWM mode.

To use the Timer Ai interrupt (interrupt request bit), set the IR bit to "0" by program after the above listed changes have been made.

When setting TAI S register to "0" (count stop) during PWM pulse output, the following action occurs:

- Stop counting.
- When TAIOUT pin is output "H," output level is set to "L" and the IR bit is set to "1".
- When TAIOUT pin is output "L," both output level and the IR bit remains unchanged.

Timer B

(1) Timer B (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 5) register and TBi register before setting the TBiS bit in the TABSR or the TBSR register to "1" (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains "0" (count stops) regardless whether after reset or not.

A value of a counter, while counting, can be read in TBi register at any time. "FFFFh" is read while reloading. Setting value is read between setting values in TBi register at count stop and starting a counter.

(2) Timer B (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 5) register and TBi register before setting the TBiS bit in the TABSR or the TBSR register to "1" (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains "0" (count stops) regardless whether after reset or not.

The counter value can be read out on-the-fly at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always "FFFFh." If the TBi register is read after setting a value in it while not counting but before the counter starts counting, the read value is the one that has been set in the register.

(3) Timer B (Pulse Period/pulse Width Measurement Mode)

The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR (i = 0 to 5) register before setting the TBiS bit in the TABSR or the TBSR register to "1" (count starts).

Always make sure the TBiMR register is modified while the TBiS bit remains "0" (count stops) regardless whether after reset or not. To clear the MR3 bit to "0" by writing to the TBiMR register while the TBiS bit = 1 (count starts), be sure to write the same value as previously written to the TM0D0, TM0D1, MR0, MR1, TCK0 and TCK1 bits and a 0 to the MR2 bit.

The IR bit in the TBiIC register (i=0 to 5) goes to "1" (interrupt request), when an effective edge of a measurement pulse is input or Timer Bi is overflowed. The factor of interrupt request can be determined by use of the MR3 bit in the TBiMR register within the interrupt routine.

If the source of interrupt cannot be identified by the MR3 bit such as when the measurement pulse input and a timer overflow occur at the same time, use another timer to count the number of times Timer B has overflowed.

To set the MR3 bit to "0" (no overflow), set TBiMR register with setting the TBiS bit to "1" and counting the next count source after setting the MR3 bit to "1" (overflow).

Use the IR bit to detect only overflows. Use the MR3 bit only to determine the interrupt factor within the interrupt routine.

When a count is started and the first effective edge is input, an indeterminate value is transferred to the reload register. At this time, Timer Bi interrupt request is not generated.

A value of the counter is indeterminate at the beginning of a count. MR3 may be set to "1" and Timer Bi interrupt request may be generated between a count start and an effective edge input.

For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an "H" level width or an "L" level width.

Serial I/O

Clock Synchronous Serial I/O

(1) Transmission/reception

With an external clock selected, and choosing the $\overline{\text{RTSi}}$ function, the output level of the $\overline{\text{RTSi}}$ pin goes to “L” when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{\text{RTSi}}$ pin goes to “H” when reception starts. So if the $\overline{\text{RTSi}}$ pin is connected to the $\overline{\text{CTSi}}$ pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the $\overline{\text{RTS}}$ function has no effect.

(2) Transmission

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

- The TE bit in the UiC1 register= 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in UiTB register)
- If CTS function is selected, input on the $\overline{\text{CTSi}}$ pin = L

(3) Reception

In operating the clock-synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TXDi pin when receiving data.

When an internal clock is selected, set the TE bit in the UiC1 register ($i = 0$ to 2) to 1 (transmission enabled) and write dummy data to the UiTB register, and the shift clock will thereby be generated. When an external clock is selected, set the TE bit to 1 and write dummy data to the UiTB register, and the shift clock will be generated when the external clock is fed to the CLKi input pin.

When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the RE bit in the UiC1 register ($i = 0$ to 2) = 1 (data present in the UiRB register), an overrun error occurs and the OER bit in the UiRB register is set to “1” (overrun error occurred). In this case, because the content of the UiRB register is indeterminate, a corrective measure must be taken by programs on the transmit and receive sides so that the valid data before the overrun error occurred will be retransmitted. Note that when an overrun error occurred, the IR bit in the SiRIC register does not change state.

To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.

When an external clock is selected, the conditions must be met while if the CKPOL bit = 0, the external clock is in the high state; if the CKPOL bit = 1, the external clock is in the low state.

- The RE bit in the UiC1 register= 1 (reception enabled)
- The TE bit in the UiC1 register= 1 (transmission enabled)
- The TI bit in the UiC1 register= 0 (data present in the UiTB register)

A/D Converter

Set ADCON0 (except bit 6), ADCON1 and ADCON2 registers when A/D conversion is stopped (before a trigger occurs).

When the VCUT bit in the ADCON1 register is changed from "0" (Vref not connected) to "1" (Vref connected), start A/D conversion after passing 1 μ s or longer.

When changing an A/D operation mode, select analog input pin again in the CH2 to CH0 bits in the ADCON0 register and the SCAN1 to SCAN0 bits in the ADCON1 register.

When setting the ADST bit in the ADCON0 register to "0" in single-sweep mode during A/D conversion and aborting A/D conversion, disable the interrupt before setting the ADST bit to "0".

Programmable I/O Ports

The input threshold voltage of pins differs between programmable input/output ports and peripheral functions.

Therefore, if any pin is shared by a programmable input/output port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions V_{IH} and V_{IL} (neither “high” nor “low”), the input level may be determined differently depending on which side—the programmable input/output port or the peripheral function—is currently selected.

Flash Memory Version

(1) Functions to Inhibit Rewriting Flash Memory Rewrite

ID codes are stored in addresses 0FFFFDFh, 0FFFFE3h, 0FFFFEBh, 0FFFFEFh, 0FFFFF3h, 0FFFFF7h, and 0FFFFFBh. 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 0FFFFFFh. 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.

(2) Program Command

Write "xx40h" in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

(3) Lock Bit Program Command

Write "xx77h" in the first bus cycle and write "xxD0h" to the uppermost address of a block (even address, however) in the second bus cycle, and the lock bit for the specified block is cleared to "0". Make sure the address value specified in the first bus cycle is the same uppermost block address that is specified in the second bus cycle.

Noise

Use thick and shortest possible wiring to connect bypass capacitors (0.1 μ F) between the VCC1 pin and VSS pin, VCC2 pin and VSS pin, and VCC3 pin and VSS pin.

Figure 20.3 shows the bypass capacitor connection.

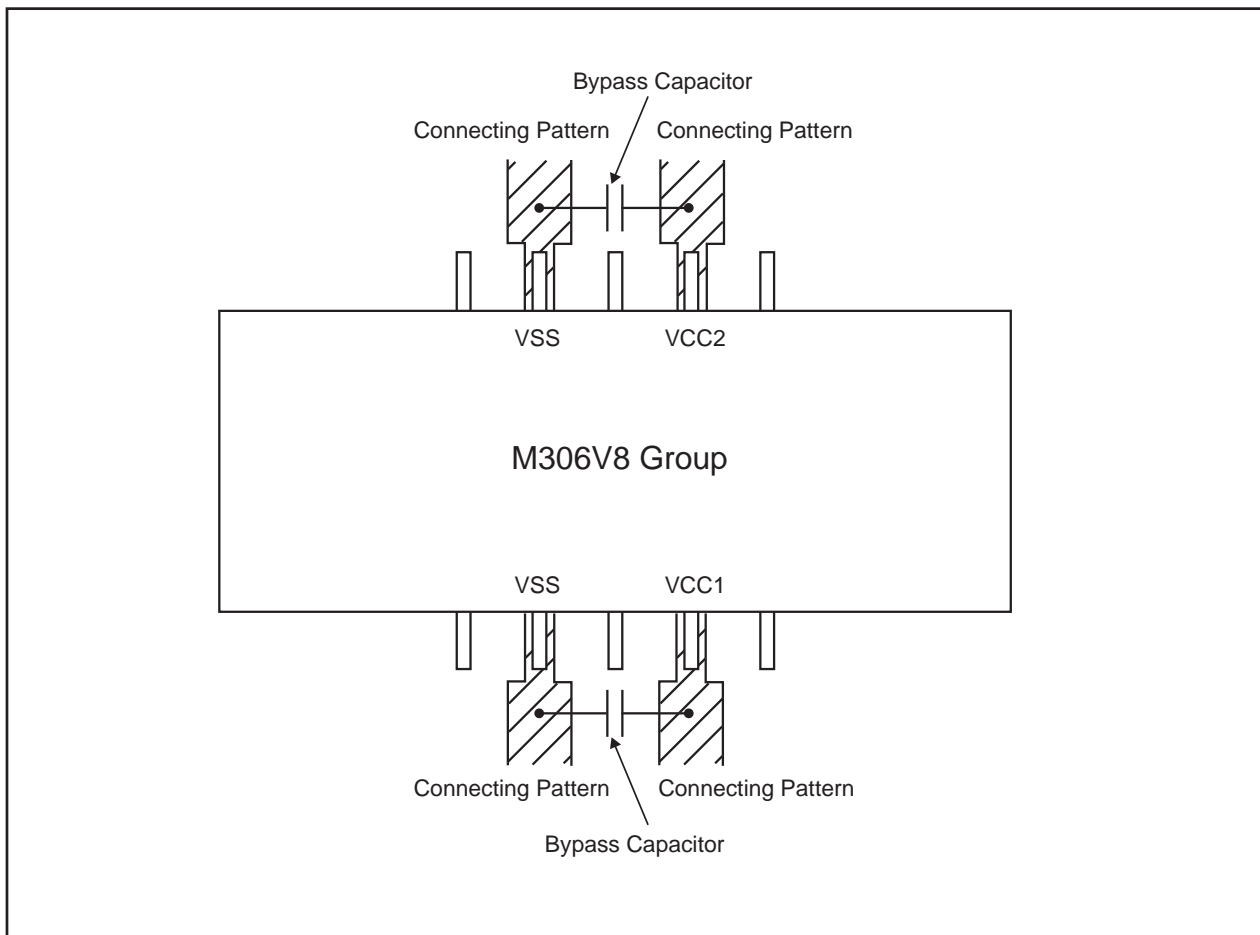


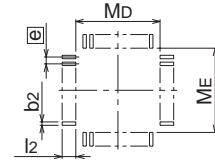
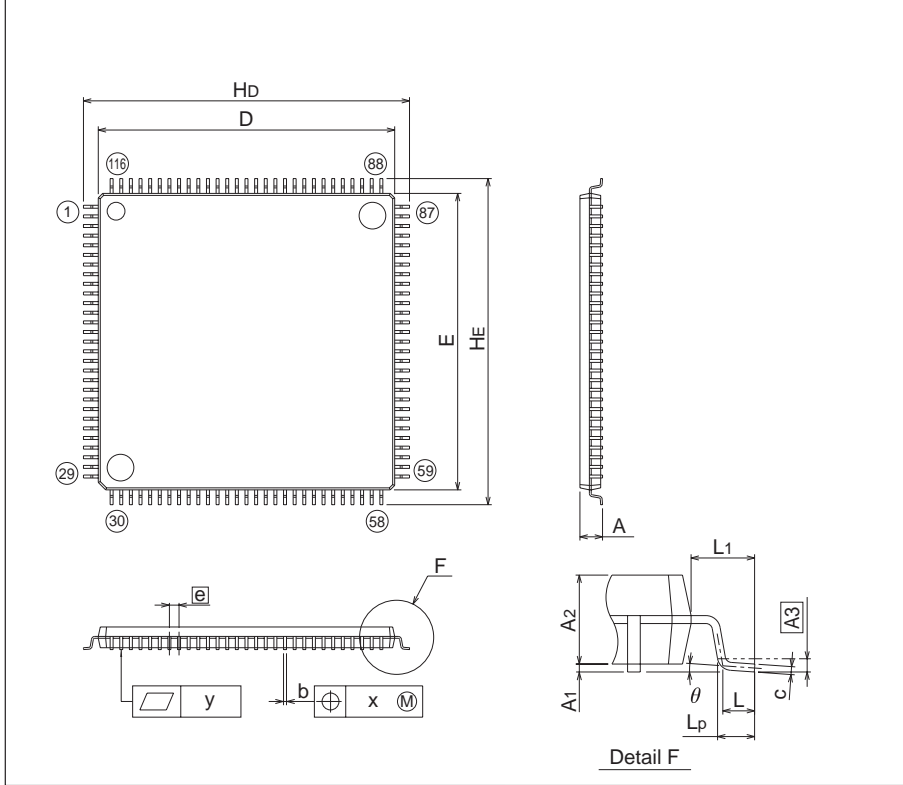
Figure 20.3 Bypass Capacitor Connection

Package Outline

116P6A-A MMP

Plastic 116pin 20X20mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP116-P-2020-0.65	-		Cu Alloy



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	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	19.9	20.0	20.1
E	19.9	20.0	20.1
e	-	0.65	-
Hd	21.8	22.0	22.2
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	-	-	0.13
y	-	-	0.1
theta	0°	-	8°
b2	-	0.225	-
l2	0.95	-	-
MD	-	20.4	-
ME	-	20.4	-

REVISION HISTORY

M306V8FJFP

Rev.	Date	Description	
		Page	Summary
1.30	Mar 15, 2005	–	First edition issued.
1.31	Apr 18, 2005	2	Table 1.1 revised.
		25	Table 3.1 partly deleted.
		265	Figure 17.5 partly deleted.
		295	Table 18.1 partly deleted.
		298	Table 18.7 revised.
		299	Table 18.8 partly deleted.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

Keep safety first in your circuit designs!

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

Notes regarding these materials

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



RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

Renesas Technology America, Inc.

450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology Hong Kong Ltd.

7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.

10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.

Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001