To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

M306V8FJFP

REJ03B0082-0131
Rev.1.31
Apr 18, 2005

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

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

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^2Cbus 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 • Main clock • Sub-clock } (These circuits contain a built-in feedback resistor and external ceramic/quartz oscillator) • OSD clock }
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. I^2C 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.

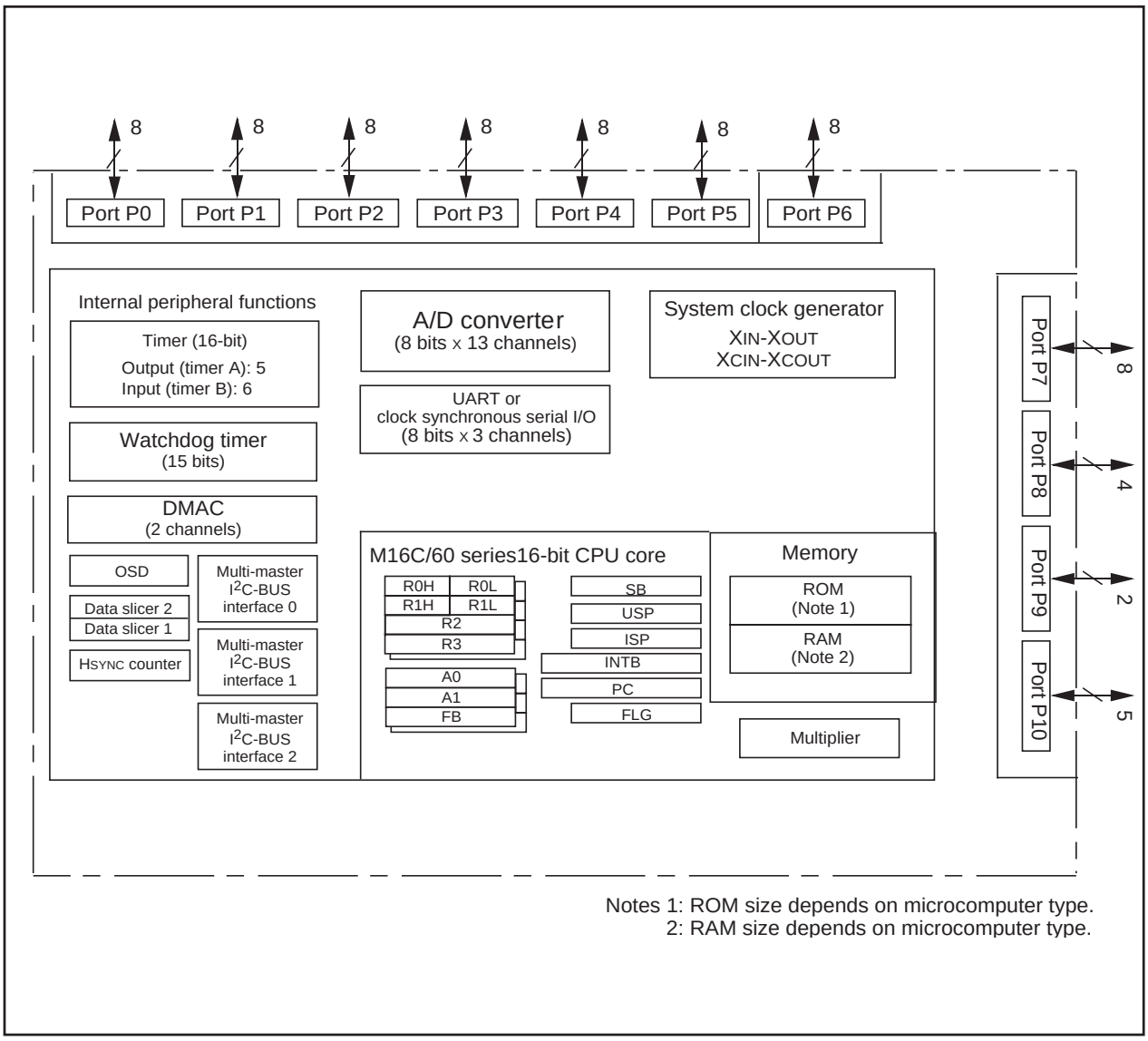


Figure 1.1. Block Diagram

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

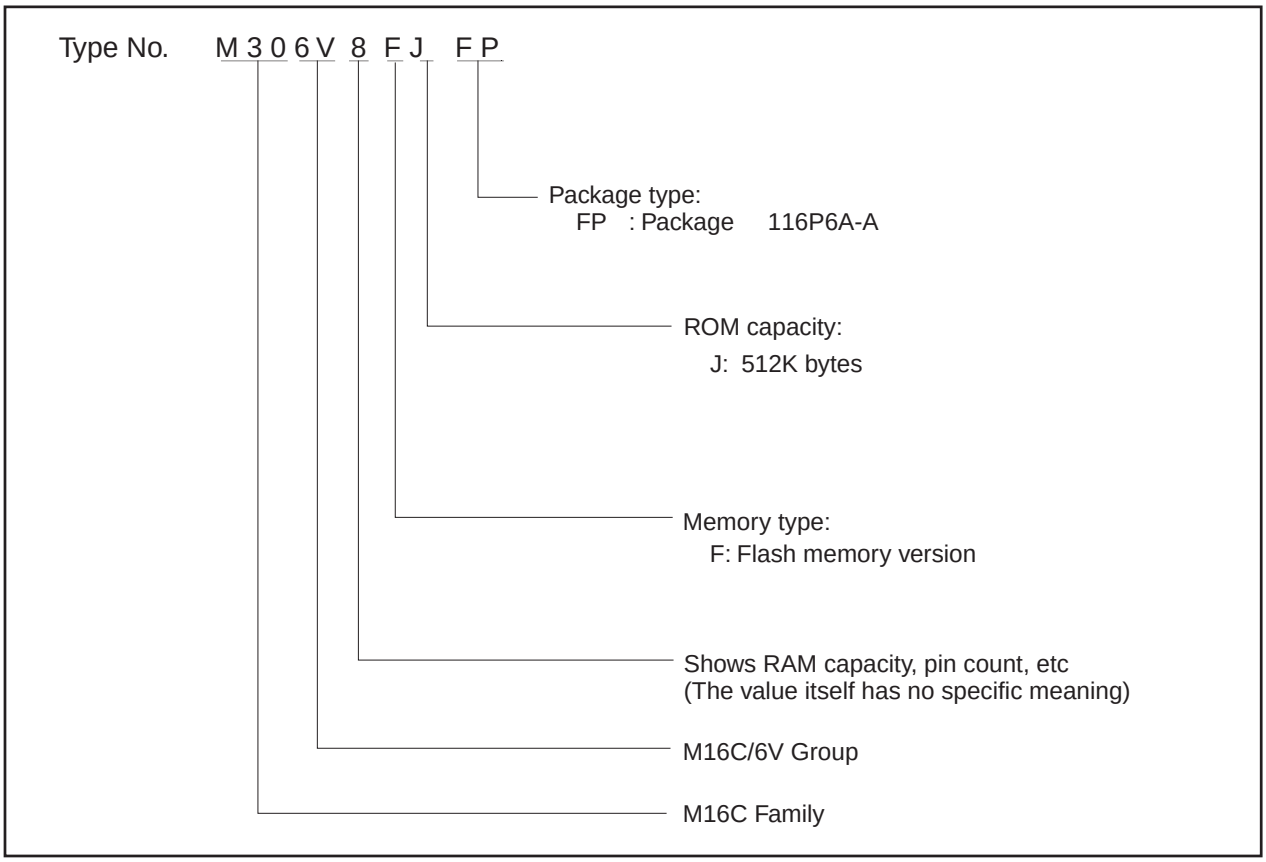
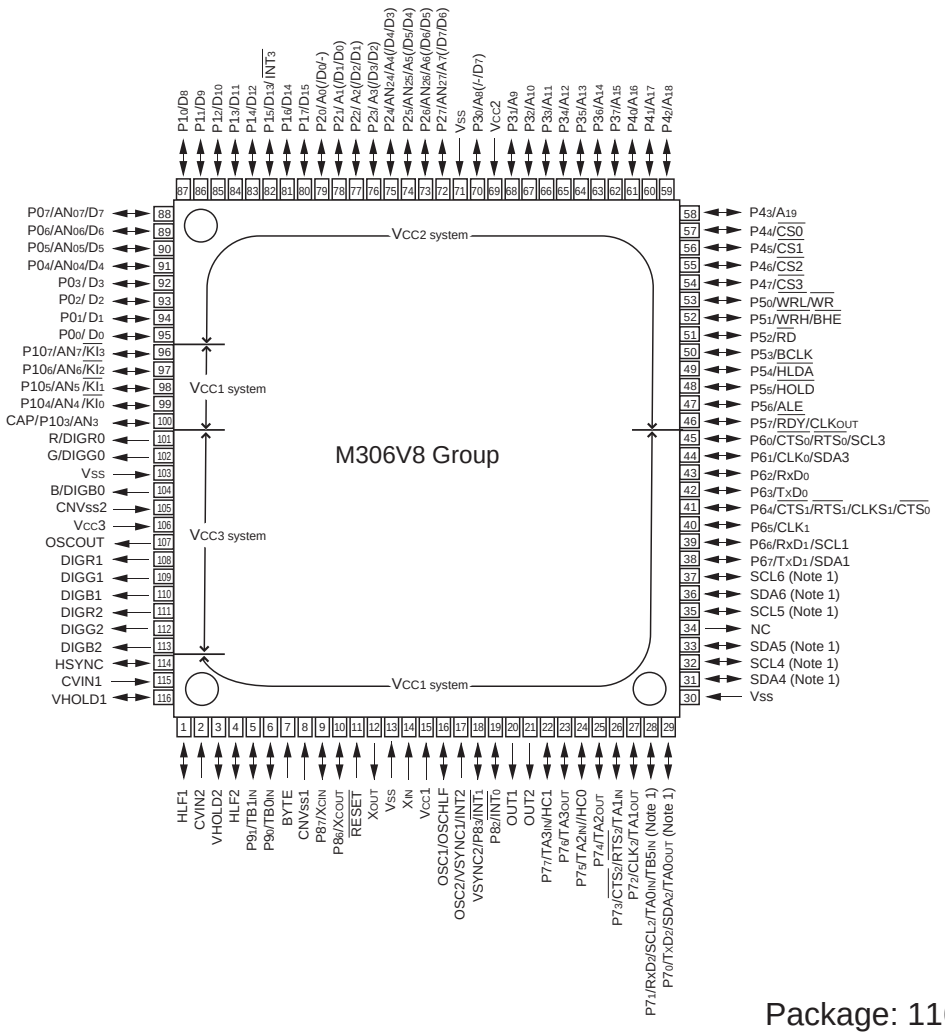


Figure 1.2. Type No., Memory Size, and Package

Pin Configuration

Figures 1.3 show the pin configuration.

PIN CONFIGURATION (top view)



Package: 116P6A-A

Notes 1: N channel open-drain output pins.
2: Use this MCU (microcomputer) with $V_{CC1} = V_{CC2} = V_{CC3} = 3.3V$.

Figure 1.3. Pin Configuration (Top View)

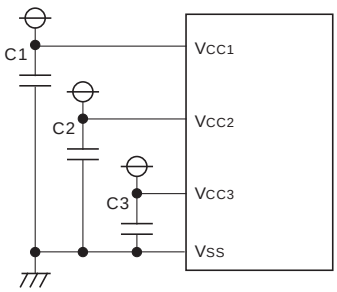
Pin Description

Table 1.3. Pin Description (1)

Pin name	Signal name	I/O type	Power supply	Function
V_{CC1}, V_{CC2}, V_{CC3}, V_{SS}	Power supply input		—	Apply 3.3 V to the V_{CC1}, V_{CC2} and V_{CC3} pins and 0 V to the V_{SS}pin. (Note 1) Insert a bypass capacitor between power supply and GND. (Note 2)
CNV_{SS1}, CNV_{SS2}	CNV_{SS1}/ CNV_{SS2}	Input	V_{CC}	CNV_{SS}1 pin switches between processor modes. Connect this pin to V_{SS} pin when after a reset you want to start operation in single-chip mode (memory expansion mode) or the V_{CC1} pin when starting operation in microprocessor mode. Always connect CNV_{SS}2 pin to V_{SS}.
$\overline{RESET}$	Reset input	Input	V_{CC}	"L" on this input resets the microcomputer.
X_{IN} X_{OUT}	Clock input Clock output	Input Output	V_{CC}	These pins are provided for the main clock generating circuit input/ output. Connect a ceramic resonator or crystal between the X_{IN} and the X_{OUT} pins. To use an externally derived clock, input it to the X_{IN} pin and leave the X_{OUT} pin open.
BYTE	External data bus width select input	Input	V_{CC}	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 V_{SS} pin when operating in single-chip mode.
$P0_0$ to $P0_7$	I/O port P0	I/O	V_{CC}	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.
D_0 to D_7		I/O		When set as a separate bus, these pins input and output data (D_0 –D_7).
$P1_0$ to $P1_7$	I/O port P1	I/O	V_{CC}	This is an 8-bit I/O port equivalent to P0. $P1_5$ also function as $\overline{INT}$ interrupt input pins as selected by a program.
D_8 to D_{15}		I/O		When set as a separate bus, these pins input and output data (D_8 –D_{15}).
$P2_0$ to $P2_7$	I/O port P2	I/O	V_{CC}	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.
A_0 to A_7		Output		These pins output 8 low-order address bits (A_0 to A_7).
A_0/D_0 to A_7/D_7		I/O		If the external bus is set as an 8-bit wide multiplexed bus, these pins input and output data (D_0 to D_7) and output 8 low-order address bits (A_0 to A_7) separated in time by multiplexing.
A_0 A_1/D_0 to A_7/D_6		Output I/O		If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D_0 to D_6) and output address (A_1 to A_7) separated in time by multiplexing. They also output address (A_0).
$P3_0$ to $P3_7$	I/O port P3	I/O	V_{CC}	This is an 8-bit I/O port equivalent to P0.
A_8 to A_{15}		Output		These pins output 8 middle-order address bits (A_8 to A_{15}).
A_8/D_7, A_9 to A_{15}		I/O Output		If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D_7) and output address (A_8) separated in time by multiplexing. They also output address (A_9 to A_{15}).
$P4_0$ to $P4_7$	I/O port P4	I/O	V_{CC}	This is an 8-bit I/O port equivalent to P0.
$\overline{A_{16}}$ to $\overline{A_{19}}$, $\overline{CS_0}$ to $\overline{CS_3}$		Output Output		These pins output A_{16} to A_{19} and $\overline{CS_0}$ to $\overline{CS_3}$ signals. A_{16} to A_{19} are 4 high- order address bits. $\overline{CS_0}$ to $\overline{CS_3}$ are chip select signals used to specify an access space.

Notes 1: In this manual, hereafter, V_{CC} refers to V_{CC1} 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.



C1 ≧0.1∝F, C2 ≧0.1∝F, C3 ≧0.1∝F (reference value)

Table 1.4. Pin Description (2)

Pin name	Signal name	I/O type	Power supply	Function
P5$_0$ to P5$_7$	I/O port P5	I/O	V_{CC}	This is an 8-bit I/O port equivalent to P0. In single-chip mode, P5$_7$ in this port outputs a divide-by-8 or divide-by-32 clock of X_{IN} or a clock of the same frequency as X_{CIN} as selected by program.
$\overline{WRL}$ / $\overline{WR}$, $\overline{WRH}$ / $\overline{BHE}$, $\overline{RD}$, BCLK, $\overline{HLDA}$, $\overline{HOLD}$, ALE, $\overline{RDY}$		Output Output Output Output Output Input Output Input		Output $\overline{WRL}$/$\overline{WRH}$, ($\overline{BHE}$/$\overline{WR}$), $\overline{RD}$, BCLK, $\overline{HLDA}$, 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 $\overline{BHE}$. ■ $\overline{WRL}$, $\overline{WRH}$, and $\overline{RD}$ 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 $\overline{RD}$ signal is low. ■ $\overline{WR}$, $\overline{BHE}$, and $\overline{RD}$ selected Data are written when the $\overline{WR}$ signal is low, or read out when the $\overline{RD}$ 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, $\overline{HLDA}$ outputs a low level. ALE is used to latch the address. While the input level of the $\overline{RDY}$ pin is low, the bus of the microcomputer goes to a wait state.
P6$_0$ to P6$_7$	I/O port P6	I/O	V_{CC}	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as UART0, UART1 and multi-master I^2C bus I/O pins as selected by program.
P7$_0$ to P7$_7$	I/O port P7	I/O	V_{CC}	This is an 8-bit I/O port equivalent to P0. (However, P7$_0$ and P7$_1$ 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^2C bus I/O pin, P7$_5$ and P7$_7$ can also function as input pin for Hsync conter.
P8$_2$, P8$_4$, P8$_6$, P8$_7$,	I/O port P8	I/O	V_{CC}	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 $\overline{INT}$ interrupt, Vsync input pins and the sub clock oscillator circuit. In that case, connect a crystal resonator between P8$_6$ (X_{COUT} pin) and P8$_7$ (X_{CIN} pin).
P9$_0$, P9$_1$	I/O port P9	I/O	V_{CC}	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.
P10$_3$ to P10$_7$	I/O port P10	I/O	V_{CC}	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^2C-bus interface	I/O		These are exclusive pins for multi-master I^2C-bus interface (N-channel open drain output.)
Hsync	Hsync input	Input	V_{CC}	OSD function Hsync input pins.
R/DIGR0, B/DIGB0, G/DIGG0, OUT1, OUT2, DIGR1, DIGB1, DIGG1, DIGR2, DIGG2, DIGB2, OSCOUT	OSD function output pin	Output	V_{CC}	These are exclusive pins for OSD functions.
CVin1, VHOLD1, HLF1, CVin2, VHOLD2, HLF2	Data slicer function I/O pin	I/O	V_{CC}	These are exclusive pins for data slicer function.
OSC1/ OSDHLF, OSC2	Oscillation pin for OSD function	I/O	V_{CC}	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 00000_{16} to $FFFFF_{16}$.

The internal ROM is allocated in a lower address direction beginning with address $FFFFF_{16}$. For example, a 64 Kbytes internal ROM is allocated to the addresses from $F0000_{16}$ to $FFFFF_{16}$.

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

The internal RAM is allocated in an upper address direction beginning with address 00400_{16}. For example, a 10 Kbytes internal RAM is allocated to the addresses from 00400_{16} to $02BFF_{16}$. In addition to storing data, the internal RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SRF is allocated to the addresses from 00000_{16} to $003FF_{16}$. Peripheral function control registers are located here. Of the SFR, any area which has no functions allocated is reserved for future use and cannot be used by users.

The special page vector table is allocated to the addresses from $FFE00_{16}$ to $FFFDB_{16}$. 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.

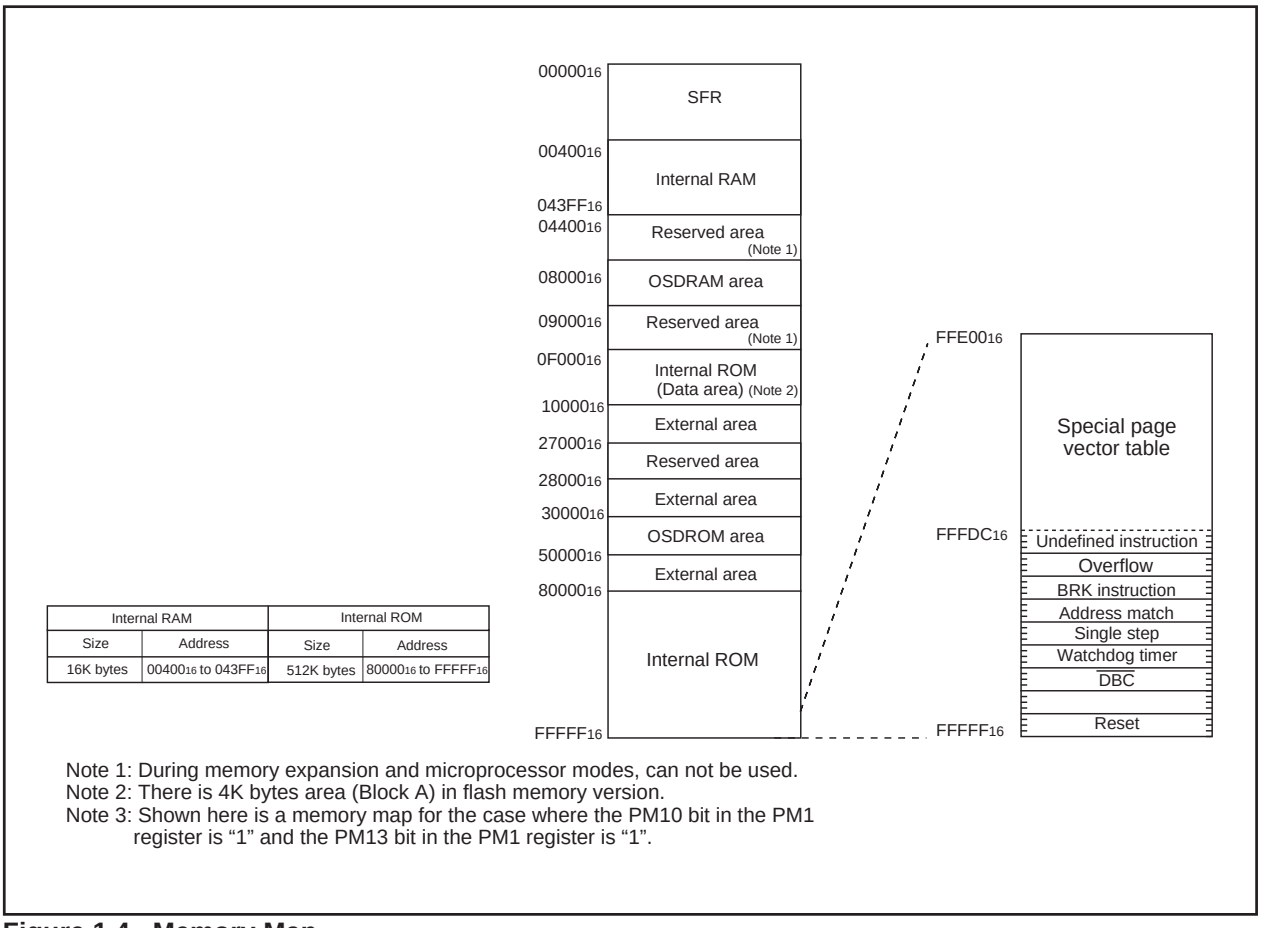
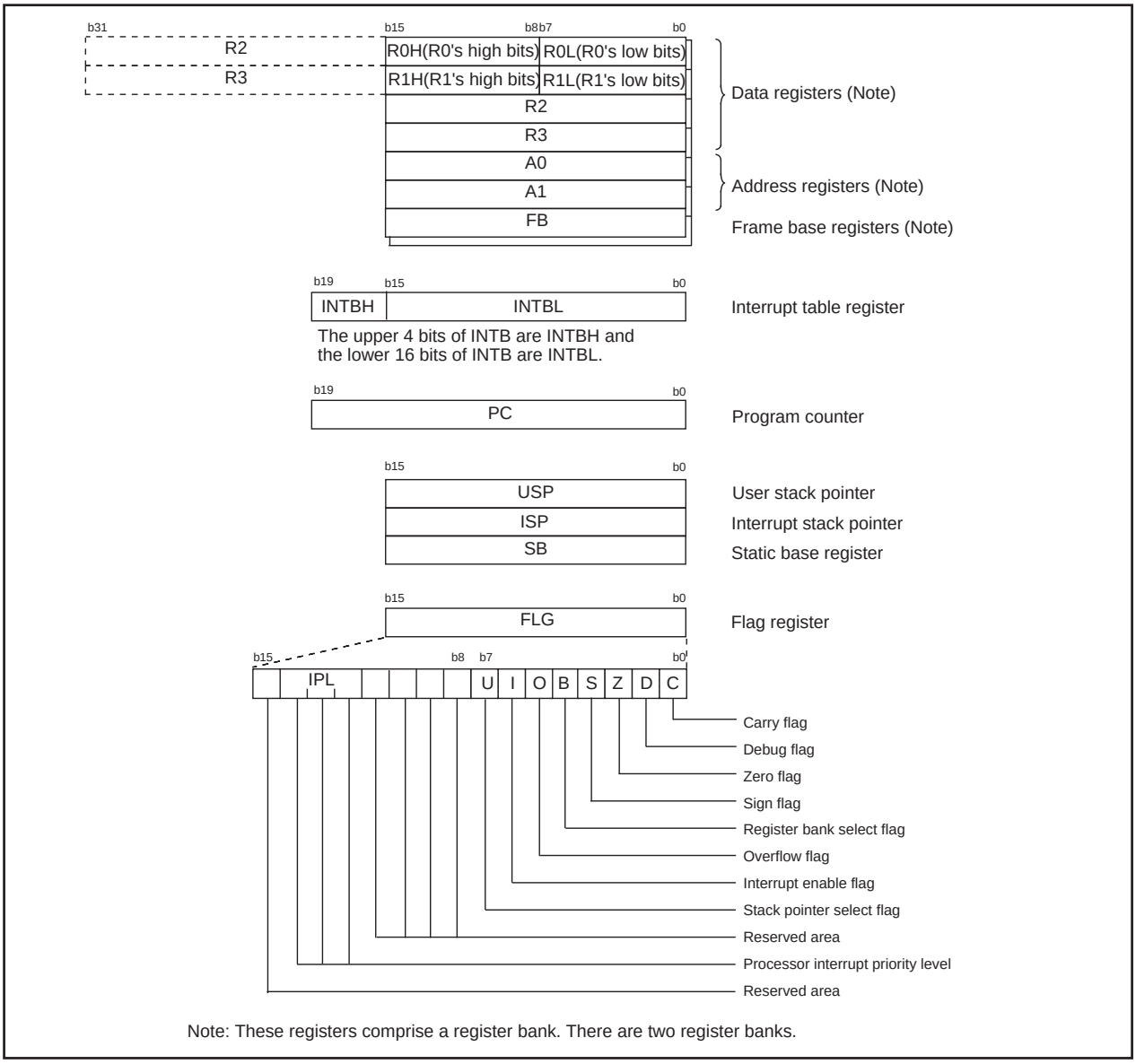


Figure 1.4. Memory Map

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.



Note: These registers 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
0000₁₆			
0001₁₆			
0002₁₆			
0003₁₆			
0004₁₆	Processor mode register 0 (Note 2)	PM0	00000000₂(CNVSS1 pin is "L") 00000011₂(CNVSS1 pin is "H")
0005₁₆	Processor mode register 1	PM1	00001000₂
0006₁₆	System clock control register 0	CM0	01001000₂
0007₁₆	System clock control register 1	CM1	00100000₂
0008₁₆	Chip select control register	CSR	00000001₂
0009₁₆	Address match interrupt enable register	AIER	XXXXXX00₂
000A₁₆	Protect register	PRCR	XX000000₂
000B₁₆	Data bank register	DBR	00₁₆
000C₁₆	System clock control register 2	CM2	0000X000₂
000D₁₆			
000E₁₆	Watchdog timer start register	WDTS	??₁₆
000F₁₆	Watchdog timer control register	WDC	00??????₂
0010₁₆	Address match interrupt register 0	RMAD0	00₁₆
0011₁₆			00₁₆
0012₁₆			X0₁₆
0013₁₆			
0014₁₆	Address match interrupt register 1	RMAD1	00₁₆
0015₁₆			00₁₆
0016₁₆			X0₁₆
0017₁₆			
0018₁₆			
0019₁₆	Reserved register	RSVREG0019	00001000₂
001A₁₆	Reserved register	RSVREG001A	00₁₆
001B₁₆	Chip select expansion control register 2	CSE	00₁₆
001C₁₆	Reserved register	RSVREG001C	0001X010₂
001D₁₆			
001E₁₆	Reserved register	RSVREG001E	XXX00000₂
001F₁₆	Reserved register	RSVREG001F	00₁₆
0020₁₆	DMA0 source pointer	SAR0	??₁₆
0021₁₆			??₁₆
0022₁₆			X?₁₆
0023₁₆			
0024₁₆	DMA0 destination pointer	DAR0	??₁₆
0025₁₆			??₁₆
0026₁₆			X?₁₆
0027₁₆			
0028₁₆	DMA0 transfer counter	TCR0	??₁₆
0029₁₆			??₁₆
002A₁₆			
002B₁₆			
002C₁₆	DMA0 control register	DM0CON	00000?00₂
002D₁₆			
002E₁₆			
002F₁₆			
0030₁₆	DMA1 source pointer	SAR1	??₁₆
0031₁₆			??₁₆
0032₁₆			X?₁₆
0033₁₆			
0034₁₆	DMA1 destination pointer	DAR1	??₁₆
0035₁₆			??₁₆
0036₁₆			X?₁₆
0037₁₆			
0038₁₆	DMA1 transfer counter	TCR1	??₁₆
0039₁₆			??₁₆
003A₁₆			
003B₁₆			
003C₁₆	DMA1 control register	DM1CON	00000?00₂
003D₁₆			
003E₁₆			
003F₁₆			

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
0040₁₆			
0041₁₆			
0042₁₆			
0043₁₆			
0044₁₆	INT3 interrupt control register	INT3IC	XX00?000₂
0045₁₆	Timer B5 interrupt control register	TB5IC	XXXX?000₂
0046₁₆	Timer B4 interrupt control register	TB4IC	XXXX?000₂
0047₁₆	Timer B3 interrupt control register	TB3IC	XXXX?000₂
0048₁₆	Slicer 1 interrupt control register	DSC1IC	XX00?000₂
0049₁₆	Slicer 2 interrupt control register	DSC2IC	XX00?000₂
004A₁₆	Bus collision detection interrupt control register	BCNIC	XXXX?000₂
004B₁₆	DMA0 interrupt control register	DM0IC	XXXX?000₂
004C₁₆	DMA1 interrupt control register	DM1IC	XXXX?000₂
004D₁₆	Key input interrupt control register	KUPIC	XXXX?000₂
004E₁₆	A/D conversion interrupt control register	ADIC	XXXX?000₂
004F₁₆	UART2 transmit interrupt control register	S2TIC	XXXX?000₂
0050₁₆	UART2 receive interrupt control register	S2RIC	XXXX?000₂
0051₁₆	UART0 transmit interrupt control register	S0TIC	XXXX?000₂
0052₁₆	UART0 receive interrupt control register	S0RIC	XXXX?000₂
0053₁₆	UART1 transmit interrupt control register	S1TIC	XXXX?000₂
0054₁₆	UART1 receive interrupt control register	S1RIC	XXXX?000₂
0055₁₆	Timer A0 interrupt control register	TA0IC	XXXX?000₂
0056₁₆	Timer A1 interrupt control register	TA1IC	XXXX?000₂
0057₁₆	Timer A2 interrupt control register	TA2IC	XXXX?000₂
0058₁₆	Timer A3 interrupt control register	TA3IC	XXXX?000₂
0059₁₆	Timer A4 interrupt control register	TA4IC	XXXX?000₂
005A₁₆	Timer B0 interrupt control register	TB0IC	XXXX?000₂
005B₁₆	Timer B1 interrupt control register	TB1IC	XXXX?000₂
005C₁₆	Timer B2 interrupt control register	TB2IC	XXXX?000₂
005D₁₆	INT0 interrupt control register	INT0IC	XX00?000₂
005E₁₆	INT1 interrupt control register	INT1IC	XX00?000₂
005F₁₆	INT2 interrupt control register	INT2IC	XX00?000₂
0060₁₆			
0061₁₆			
0062₁₆			
0063₁₆			
0064₁₆			
0065₁₆			
0066₁₆			
0067₁₆			
0068₁₆			
0069₁₆			
006A₁₆			
006B₁₆			
006C₁₆			
006D₁₆			
006E₁₆			
006F₁₆			
0070₁₆			
0071₁₆			
0072₁₆			
0073₁₆			
0074₁₆			
0075₁₆			
0076₁₆			
0077₁₆			
0078₁₆			
0079₁₆			
007A₁₆			
007B₁₆			
007C₁₆			
007D₁₆			
007E₁₆			
007F₁₆			

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
0080₁₆			
0081₁₆			
0082₁₆			
0083₁₆			
0084₁₆			
0085₁₆			
0086₁₆			
≈			≈
01B0₁₆			
01B1₁₆			
01B2₁₆			
01B3₁₆			
01B4₁₆	Flash identification register (Note 2)	FIDR	XXXXXX00₂
01B5₁₆	Flash memory control register 1 (Note 2)	FMR1	0?00??0?₂
01B6₁₆			
01B7₁₆	Flash memory control register 0 (Note 2)	FMR0	??000001₂
01B8₁₆ 01B9₁₆ 01BA₁₆	Address match interrupt register 2	RMAD2	00₁₆ 00₁₆ X0₁₆
01BB₁₆	Address match interrupt enable register 2	AIER2	XXXXXX00₂
01BC₁₆ 01BD₁₆ 01BE₁₆	Address match interrupt register 3	RMAD3	00₁₆ 00₁₆ X0₁₆
01BF₁₆			
01C0₁₆			
01C1₁₆			
01C2₁₆			
≈			≈
01E0₁₆			
01E1₁₆			
01E2₁₆			
01E3₁₆			
01E4₁₆			
01E5₁₆			
01E6₁₆			
01E7₁₆			
01E8₁₆			
01E9₁₆			
01EA₁₆			
01EB₁₆			
01EC₁₆			
01ED₁₆			
01EE₁₆			
01EF₁₆			
01F0₁₆			
01F1₁₆			
01F2₁₆			
01F3₁₆			
01F4₁₆			
01F5₁₆			
01F6₁₆			
01F7₁₆			
01F8₁₆			
01F9₁₆			
01FA₁₆			
01FB₁₆			
01FC₁₆			
01FD₁₆			
01FE₁₆			
01FF₁₆			

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
0200₁₆			
0201₁₆	Sprite OSD control register	SC	XXX00000₂
0202₁₆	OSD control register 1	OC1	00₁₆
0203₁₆	OSD control register 2	OC2	00₁₆
0204₁₆	Horizontal position register	HP	00₁₆
0205₁₆	Clock control register 1	CS	00₁₆
0206₁₆	I/O polarity control register	PC	10000000₂
0207₁₆	OSD control register 3	OC3	00₁₆
0208₁₆ 0209₁₆	Raster color register	RSC	00₁₆ 00₁₆
020A₁₆	OSD reserved register 5	OR5	00₁₆
020B₁₆	Clock control register 2	CG	00₁₆
020C₁₆ 020D₁₆	Top border control register	TBR	??₁₆
020E₁₆ 020F₁₆	Bottom border control register	BBR	??₁₆
0210₁₆	Block control register 1	BC1	??₁₆
0211₁₆	Block control register 2	BC2	??₁₆
0212₁₆	Block control register 3	BC3	??₁₆
0213₁₆	Block control register 4	BC4	??₁₆
0214₁₆	Block control register 5	BC5	??₁₆
0215₁₆	Block control register 6	BC6	??₁₆
0216₁₆	Block control register 7	BC7	??₁₆
0217₁₆	Block control register 8	BC8	??₁₆
0218₁₆	Block control register 9	BC9	??₁₆
0219₁₆	Block control register 10	BC10	??₁₆
021A₁₆	Block control register 11	BC11	??₁₆
021B₁₆	Block control register 12	BC12	??₁₆
021C₁₆	Block control register 13	BC13	??₁₆
021D₁₆	Block control register 14	BC14	??₁₆
021E₁₆	Block control register 15	BC15	??₁₆
021F₁₆	Block control register 16	BC16	??₁₆
0220₁₆ 0221₁₆	Vertical position register 1	VP1	??₁₆ ??₁₆
0222₁₆ 0223₁₆	Vertical position register 2	VP2	??₁₆ ??₁₆
0224₁₆ 0225₁₆	Vertical position register 3	VP3	??₁₆ ??₁₆
0226₁₆ 0227₁₆	Vertical position register 4	VP4	??₁₆ ??₁₆
0228₁₆ 0229₁₆	Vertical position register 5	VP5	??₁₆ ??₁₆
022A₁₆ 022B₁₆	Vertical position register 6	VP6	??₁₆ ??₁₆
022C₁₆ 022D₁₆	Vertical position register 7	VP7	??₁₆ ??₁₆
022E₁₆ 022F₁₆	Vertical position register 8	VP8	??₁₆ ??₁₆
0230₁₆ 0231₁₆	Vertical position register 9	VP9	??₁₆ ??₁₆
0232₁₆ 0233₁₆	Vertical position register 10	VP10	??₁₆ ??₁₆
0234₁₆ 0235₁₆	Vertical position register 11	VP11	??₁₆ ??₁₆
0236₁₆ 0237₁₆	Vertical position register 12	VP12	??₁₆ ??₁₆
0238₁₆ 0239₁₆	Vertical position register 13	VP13	??₁₆ ??₁₆
023A₁₆ 023B₁₆	Vertical position register 14	VP14	??₁₆ ??₁₆
023C₁₆ 023D₁₆	Vertical position register 15	VP15	??₁₆ ??₁₆
023E₁₆ 023F₁₆	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
0240₁₆ 0241₁₆	Color palette register 1	CR1	??16 ??16
0242₁₆ 0243₁₆	Color palette register 2	CR2	??16 ??16
0244₁₆ 0245₁₆	Color palette register 3	CR3	??16 ??16
0246₁₆ 0247₁₆	Color palette register 4	CR4	??16 ??16
0248₁₆ 0249₁₆	Color palette register 5	CR5	??16 ??16
024A₁₆ 024B₁₆	Color palette register 6	CR6	??16 ??16
024C₁₆ 024D₁₆	Color palette register 7	CR7	??16 ??16
024E₁₆ 024F₁₆	Color palette register 9	CR9	??16 ??16
0250₁₆ 0251₁₆	Color palette register 10	CR10	??16 ??16
0252₁₆ 0253₁₆	Color palette register 11	CR11	??16 ??16
0254₁₆ 0255₁₆	Color palette register 12	CR12	??16 ??16
0256₁₆ 0257₁₆	Color palette register 13	CR13	??16 ??16
0258₁₆ 0259₁₆	Color palette register 14	CR14	??16 ??16
025A₁₆ 025B₁₆	Color palette register 15	CR15	??16 ??16
025C₁₆			
025D₁₆	OSD reserved register 1	OR1	0016
025E₁₆	Peripheral clock selection register	PCLKR	0316
025F₁₆	OSD control register 4	OC4	XXXXXX002
0260₁₆	Data slicer 0 control register 1	DSC01	0016
0261₁₆	Data slicer 0 control register 2	DSC02	?0?0??0?2
0262₁₆ 0263₁₆	Caption data register 01	CD01	????????2 ????????2
0264₁₆ 0265₁₆	Caption data register 02	CD02	????????2 ????????2
0266₁₆	Caption position register 0	CPS0	00?000002
0267₁₆	Slice standard voltage selection register	SBV0	0016
0268₁₆	Data slicer 0 reserved register 1	DR01	0016
0269₁₆	Clock run-in detection register 0	CRD0	0016
026A₁₆	Data clock position register 0	DPS0	X00000002
026B₁₆	ID1 control register 0	IDC0	0016
026C₁₆	Standard clock detection register 0	BCD0	XX??????2
026D₁₆	CRCC data register 0	CRC0	XX0000002
026E₁₆	Test reservation register 0	IDT0	0016
026F₁₆	Reserved register	RSVREG026F	XXXXXXX02
0270₁₆ 0271₁₆	Left border control register	LBR	XXXXX0002 0016
0272₁₆ 0273₁₆	Right border control register	RBR	0016 XXXXX0002
0274₁₆ 0275₁₆	Sprite vertical position register 1	VS1	??16 ??16
0276₁₆ 0277₁₆	Sprite vertical position register 2	VS2	??16 ??16
0278₁₆ 0279₁₆	Sprite horizontal position register	HS	??16 XXXXX0002
027A₁₆	OSD reserved register 4	OR4	X00000002
027B₁₆	OSD reserved register 3	OR3	0016
027C₁₆	OSD reserved register 2	OR2	0016
027D₁₆	Peripheral mode register	PM	000XXXXX2
027E₁₆	HSYNC counter register	HC	XXX00X002
027F₁₆	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
0280₁₆	Internal oscillation control register 1	DIV0	00₁₆
0281₁₆	Internal oscillation control register 2	DIV1	00₁₆
0282₁₆	Internal oscillation control register 3	VCO	00₁₆
0283₁₆			
0284₁₆			
0285₁₆			
0286₁₆			
0287₁₆			
0288₁₆			
0289₁₆			
028A₁₆			
028B₁₆			
028C₁₆			
028D₁₆			
028E₁₆			
028F₁₆			
0290₁₆			
0291₁₆			
0292₁₆			
0293₁₆			
0294₁₆			
0295₁₆			
0296₁₆			
0297₁₆			
0298₁₆			
0299₁₆			
029A₁₆			
029B₁₆			
029C₁₆			
029D₁₆			
029E₁₆			
029F₁₆			
02A0₁₆	Flash memory (USER/OSD) change register	FMSEL	00₁₆
02A1₁₆			
02A2₁₆			
02A3₁₆	Flash memory OSD1 control register 4	FMOSA4	X0XXXX00₂
02A4₁₆			
02A5₁₆	Flash memory OSD1 control register 1	FMOSA1	XXXXXX0X₂
02A6₁₆			
02A7₁₆	Flash memory OSD1 control register 0	FMOSA0	XX000001₂
02A8₁₆			
02A9₁₆			
02AA₁₆			
02AB₁₆			
02AC₁₆			
02AD₁₆			
02AE₁₆			
02AF₁₆			
02B0₁₆			
02B1₁₆			
02B2₁₆			
02B3₁₆	Flash memory OSD2 control register 4	FMOSB4	X0XXXX00₂
02B4₁₆			
02B5₁₆	Flash memory OSD2 control register 1	FMOSB1	XXXXXX0X₂
02B6₁₆			
02B7₁₆	Flash memory OSD2 control register 0	FMOSB0	XX000001₂
02B8₁₆			
02B9₁₆			
02BA₁₆			
02BB₁₆			
02BC₁₆			
02BD₁₆			
02BE₁₆			
02BF₁₆			

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
02C0₁₆	Extended register 00	EXTREG02C0	00₁₆
02C1₁₆	Extended register 01	EXTREG02C1	00₁₆
02C2₁₆	Extended register 02	EXTREG02C2	00₁₆
02C3₁₆	Extended register 03	EXTREG02C3	00₁₆
02C4₁₆	Extended register 04	EXTREG02C4	00₁₆
02C5₁₆	Extended register 05	EXTREG02C5	00₁₆
02C6₁₆	Extended register 06	EXTREG02C6	00₁₆
02C7₁₆	Extended register 07	EXTREG02C7	00₁₆
02C8₁₆	Extended register 08	EXTREG02C8	00₁₆
02C9₁₆	Extended register 09	EXTREG02C9	00₁₆
02CA₁₆	Extended register 0A	EXTREG02CA	00₁₆
02CB₁₆	Extended register 0B	EXTREG02CB	00₁₆
02CC₁₆	Extended register 0C	EXTREG02CC	00₁₆
02CD₁₆	Extended register 0D	EXTREG02CD	00₁₆
02CE₁₆	Extended register 0E	EXTREG02CE	00₁₆
02CF₁₆	Extended register 0F	EXTREG02CF	00₁₆
02D0₁₆	Extended register 10	EXTREG02D0	00₁₆
02D1₁₆	Extended register 11	EXTREG02D1	00₁₆
02D2₁₆	Extended register 12	EXTREG02D2	00₁₆
02D3₁₆	Extended register 13	EXTREG02D3	00₁₆
02D4₁₆	Extended register 14	EXTREG02D4	00₁₆
02D5₁₆	Extended register 15	EXTREG02D5	00₁₆
02D6₁₆	Extended register 16	EXTREG02D6	00₁₆
02D7₁₆	Extended register 17	EXTREG02D7	00₁₆
02D8₁₆	Extended register 18	EXTREG02D8	00₁₆
02D9₁₆	Extended register 19	EXTREG02D9	00₁₆
02DA₁₆	Extended register 1A	EXTREG02DA	00₁₆
02DB₁₆	Extended register 1B	EXTREG02DB	00₁₆
02DC₁₆	Extended register 1C	EXTREG02DC	00₁₆
02DD₁₆	Extended register 1D	EXTREG02DD	00₁₆
02DE₁₆	Extended register 1E	EXTREG02DE	00₁₆
02DF₁₆	Extended register 1F	EXTREG02DF	00₁₆
02E0₁₆	I²C0 data shift register	IIC0S0	??₁₆
02E1₁₆	I²C0 address register	IIC0S0D	00₁₆
02E2₁₆	I²C0 status register	IIC0S1	0001000?₂
02E3₁₆	I²C0 control register	IIC0S1D	00₁₆
02E4₁₆	I²C0 clock control register	IIC0S2	00₁₆
02E5₁₆	Reserved register	RSVREG02E5	00?00000₂
02E6₁₆	I²C0 transmitting buffer register	IIC0S0S	??₁₆
02E7₁₆			
02E8₁₆	I²C1 data shift register	IIC1S0	??₁₆
02E9₁₆	I²C1 address register	IIC1S0D	00₁₆
02EA₁₆	I²C1 status register	IIC1S1	0001000?₂
02EB₁₆	I²C1 control register	IIC1S1D	00₁₆
02EC₁₆	I²C1 clock control register	IIC1S2	00₁₆
02ED₁₆	Reserved register	RSVREG02ED	00?00000₂
02EE₁₆	I²C1 transmitting buffer register	IIC1S0S	??₁₆
02EF₁₆			
02F0₁₆	I²C2 data shift register	IIC2S0	??₁₆
02F1₁₆	I²C2 address register	IIC2S0D	00₁₆
02F2₁₆	I²C2 status register	IIC2S1	0001000?₂
02F3₁₆	I²C2 control register	IIC2S1D	00₁₆
02F4₁₆	I²C2 clock control register	IIC2S2	00₁₆
02F5₁₆	Reserved register	RSVREG02F5	00?00000₂
02F6₁₆	I²C2 transmitting buffer register	IIC2S0S	??₁₆
02F7₁₆			
02F8₁₆			
02F9₁₆			
02FA₁₆			
02FB₁₆			
02FC₁₆			
02FD₁₆			
02FE₁₆			
02FF₁₆			

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
0300₁₆	Data slicer 1 control register 1	DSC11	00₁₆
0301₁₆	Data slicer 1 control register 2	DSC12	?0?0??0?₂
0302₁₆ 0303₁₆	Caption data register 11	CD11	????????₂ ????????₂
0304₁₆ 0305₁₆	Caption data register 12	CD12	????????₂ ????????₂
0306₁₆	Caption position register 1	CPS1	00?00000₂
0307₁₆	Slice standard voltage selection register	SBV1	00₁₆
0308₁₆	Data slicer 1 reserved register 1	DR11	00₁₆
0309₁₆	Clock run-in detection register 1	CRD1	00₁₆
030A₁₆	Data clock position register 1	DPS1	X0000000₂
030B₁₆	ID1 control register 1	IDC1	00₁₆
030C₁₆	Standard clock detection register 1	BCD1	XX??????₂
030D₁₆	CRCC data register 1	CRC1	XX000000₂
030E₁₆	Test reservation register 1	IDT1	00₁₆
030F₁₆	Reserved register	RSVREG030F	XXXXXXX0₂
0310₁₆			
0311₁₆			
0312₁₆			
0313₁₆			
0314₁₆			
0315₁₆			
0316₁₆			
0317₁₆			
0318₁₆			
0319₁₆			
031A₁₆			
031B₁₆			
031C₁₆	ID1 reserved register 0	IRSV0	00??????₂
031D₁₆	ID1 reserved register 1	IRSV1	00??????₂
031E₁₆			
031F₁₆			
0320₁₆			
0321₁₆			
0322₁₆			
0323₁₆			
0324₁₆			
0325₁₆			
0326₁₆			
0327₁₆			
0328₁₆			
0329₁₆			
032A₁₆			
032B₁₆			
032C₁₆			
032D₁₆			
032E₁₆			
032F₁₆			
0330₁₆			
0331₁₆			
0332₁₆			
0333₁₆			
0334₁₆			
0335₁₆			
0336₁₆			
0337₁₆			
0338₁₆			
0339₁₆			
033A₁₆			
033B₁₆			
033C₁₆			
033D₁₆			
033E₁₆			
033F₁₆			

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
0340₁₆	Timer B3, 4, 5 count start flag	TBSR	000XXXXX₂
0341₁₆			
0342₁₆	Reserved register	RSVREG0342	??₁₆
0343₁₆	Reserved register	RSVREG0343	??₁₆
0344₁₆	Reserved register	RSVREG0344	??₁₆
0345₁₆	Reserved register	RSVREG0345	??₁₆
0346₁₆	Reserved register	RSVREG0346	??₁₆
0347₁₆	Reserved register	RSVREG0347	??₁₆
0348₁₆	Reserved register	RSVREG0348	00₁₆
0349₁₆	Reserved register	RSVREG0349	00₁₆
034A₁₆	Reserved register	RSVREG034A	00₁₆
034B₁₆	Reserved register	RSVREG034B	00₁₆
034C₁₆	Reserved register	RSVREG034C	??₁₆
034D₁₆	Reserved register	RSVREG034D	??₁₆
034E₁₆			
034F₁₆			
0350₁₆ 0351₁₆	Timer B3 register	TB3	??₁₆ ??₁₆
0352₁₆ 0353₁₆	Timer B4 register	TB4	??₁₆ ??₁₆
0354₁₆ 0355₁₆	Timer B5 register	TB5	??₁₆ ??₁₆
0356₁₆			
0357₁₆			
0358₁₆			
0359₁₆			
035A₁₆			
035B₁₆	Timer B3 mode register	TB3MR	00??0000₂
035C₁₆	Timer B4 mode register	TB4MR	00?X0000₂
035D₁₆	Timer B5 mode register	TB5MR	00?X0000₂
035E₁₆	Interrupt cause select register 2	IFSR2A	00XXXXXX₂
035F₁₆	Interrupt cause select register	IFSR	00₁₆
0360₁₆	Reserved register	RSVREG0360	??₁₆
0361₁₆			
0362₁₆	Reserved register	RSVREG0362	01000000₂
0363₁₆	Reserved register	RSVREG0363	??₁₆
0364₁₆	Reserved register	RSVREG0364	??₁₆
0365₁₆			
0366₁₆	Reserved register	RSVREG0366	01000000₂
0367₁₆	Reserved register	RSVREG0367	??₁₆
0368₁₆			
0369₁₆			
036A₁₆			
036B₁₆			
036C₁₆	UART0 special mode register 4	U0SMR4	00₁₆
036D₁₆	UART0 special mode register 3	U0SMR3	000X0X0X₂
036E₁₆	UART0 special mode register 2	U0SMR2	X0000000₂
036F₁₆	UART0 special mode register	U0SMR	X0000000₂
0370₁₆	UART1 special mode register 4	U1SMR4	00₁₆
0371₁₆	UART1 special mode register 3	U1SMR3	000X0X0X₂
0372₁₆	UART1 special mode register 2	U1SMR2	X0000000₂
0373₁₆	UART1 special mode register	U1SMR	X0000000₂
0374₁₆	UART2 special mode register 4	U2SMR4	00₁₆
0375₁₆	UART2 special mode register 3	U2SMR3	000X0X0X₂
0376₁₆	UART2 special mode register 2	U2SMR2	X0000000₂
0377₁₆	UART2 special mode register	U2SMR	X0000000₂
0378₁₆	UART2 transmit/receive mode register	U2MR	00₁₆
0379₁₆	UART2 bit rate generator	U2BRG	??₁₆
037A₁₆ 037B₁₆	UART2 transmit buffer register	U2TB	????????₂ XXXXXXX?₂
037C₁₆	UART2 transmit/receive control register 0	U2C0	00001000₂
037D₁₆	UART2 transmit/receive control register 1	U2C1	00000010₂
037E₁₆ 037F₁₆	UART2 receive buffer register	U2RB	????????₂ ?????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
0380₁₆	Count start flag	TABSR	00₁₆
0381₁₆	Clock prescaler reset flag	CPSRF	0XXXXXXX₂
0382₁₆	One-shot start flag	ONSF	00₁₆
0383₁₆	Trigger select register	TRGSR	00₁₆
0384₁₆	Up-down flag	UDF	00₁₆
0385₁₆			
0386₁₆ 0387₁₆	Timer A0 register	TA0	??₁₆ ??₁₆
0388₁₆ 0389₁₆	Timer A1 register	TA1	??₁₆ ??₁₆
038A₁₆ 038B₁₆	Timer A2 register	TA2	??₁₆ ??₁₆
038C₁₆ 038D₁₆	Timer A3 register	TA3	??₁₆ ??₁₆
038E₁₆ 038F₁₆	Timer A4 register	TA4	??₁₆ ??₁₆
0390₁₆ 0391₁₆	Timer B0 register	TB0	??₁₆ ??₁₆
0392₁₆ 0393₁₆	Timer B1 register	TB1	??₁₆ ??₁₆
0394₁₆ 0395₁₆	Timer B2 register	TB2	??₁₆ ??₁₆
0396₁₆	Timer A0 mode register	TA0MR	00₁₆
0397₁₆	Timer A1 mode register	TA1MR	00₁₆
0398₁₆	Timer A2 mode register	TA2MR	00₁₆
0399₁₆	Timer A3 mode register	TA3MR	00₁₆
039A₁₆	Timer A4 mode register	TA4MR	00₁₆
039B₁₆	Timer B0 mode register	TB0MR	00??0000₂
039C₁₆	Timer B1 mode register	TB1MR	00?X0000₂
039D₁₆	Timer B2 mode register	TB2MR	00?X0000₂
039E₁₆	Reserved register	RSVREG039E	XXXXXX00₂
039F₁₆			
03A0₁₆	UART0 transmit/receive mode register	U0MR	00₁₆
03A1₁₆	UART0 bit rate generator	U0BRG	??₁₆
03A2₁₆ 03A3₁₆	UART0 transmit buffer register	U0TB	????????₂ XXXXXXX?₂
03A4₁₆	UART0 transmit/receive control register 0	U0C0	00001000₂
03A5₁₆	UART0 transmit/receive control register 1	U0C1	00000010₂
03A6₁₆ 03A7₁₆	UART0 receive buffer register	U0RB	????????₂ ?????XX?₂
03A8₁₆	UART1 transmit/receive mode register	U1MR	00₁₆
03A9₁₆	UART1 bit rate generator	U1BRG	??₁₆
03AA₁₆ 03AB₁₆	UART1 transmit buffer register	U1TB	????????₂ XXXXXXX?₂
03AC₁₆	UART1 transmit/receive control register 0	U1C0	00001000₂
03AD₁₆	UART1 transmit/receive control register 1	U1C1	00000010₂
03AE₁₆ 03AF₁₆	UART1 receive buffer register	U1RB	????????₂ ?????XX?₂
03B0₁₆	UART transmit/receive control register 2	UCON	X0000000₂
03B1₁₆			
03B2₁₆			
03B3₁₆			
03B4₁₆			
03B5₁₆			
03B6₁₆			
03B7₁₆			
03B8₁₆	DMA0 request cause select register	DM0SL	00₁₆
03B9₁₆			
03BA₁₆	DMA1 request cause select register	DM1SL	00₁₆
03BB₁₆			
03BC₁₆	Reserved register	RSVREG03BC	??₁₆
03BD₁₆	Reserved register	RSVREG03BD	??₁₆
03BE₁₆	Reserved register	RSVREG03BE	??₁₆
03BF₁₆			

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
03C0₁₆	Reserved register	RSVREG03C0	????????₂
03C1₁₆	Reserved register	RSVREG03C1	XXXXXX??₂
03C2₁₆	Reserved register	RSVREG03C2	????????₂
03C3₁₆	Reserved register	RSVREG03C3	XXXXXX??₂
03C4₁₆	Reserved register	RSVREG03C4	????????₂
03C5₁₆	Reserved register	RSVREG03C5	XXXXXX??₂
03C6₁₆	A/D register 3	AD3	????????₂
03C7₁₆			XXXXXX??₂
03C8₁₆	A/D register 4	AD4	????????₂
03C9₁₆			XXXXXX??₂
03CA₁₆	A/D register 5	AD5	????????₂
03CB₁₆			XXXXXX??₂
03CC₁₆	A/D register 6	AD6	????????₂
03CD₁₆			XXXXXX??₂
03CE₁₆	A/D register 7	AD7	????????₂
03CF₁₆			XXXXXX??₂
03D0₁₆			
03D1₁₆			
03D2₁₆			
03D3₁₆			
03D4₁₆	A/D control register 2	ADCON2	00₁₆
03D5₁₆			
03D6₁₆	A/D control register 0	ADCON0	00000???₂
03D7₁₆	A/D control register 1	ADCON1	00₁₆
03D8₁₆	Reserved register	RSVREG03D8	??₁₆
03D9₁₆			
03DA₁₆	Reserved register	RSVREG03DA	??₁₆
03DB₁₆			
03DC₁₆	Reserved register	RSVREG03DC	00₁₆
03DD₁₆			
03DE₁₆	Reserved register	RSVREG03DE	XX00XXXX₂
03DF₁₆	Reserved register	RSVREG03DF	00₁₆
03E0₁₆	Port P0 register	P0	??₁₆
03E1₁₆	Port P1 register	P1	??₁₆
03E2₁₆	Port P0 direction register	PD0	00₁₆
03E3₁₆	Port P1 direction register	PD1	00₁₆
03E4₁₆	Port P2 register	P2	??₁₆
03E5₁₆	Port P3 register	P3	??₁₆
03E6₁₆	Port P2 direction register	PD2	00₁₆
03E7₁₆	Port P3 direction register	PD3	00₁₆
03E8₁₆	Port P4 register	P4	??₁₆
03E9₁₆	Port P5 register	P5	??₁₆
03EA₁₆	Port P4 direction register	PD4	00₁₆
03EB₁₆	Port P5 direction register	PD5	00₁₆
03EC₁₆	Port P6 register	P6	??₁₆
03ED₁₆	Port P7 register	P7	??₁₆
03EE₁₆	Port P6 direction register	PD6	00₁₆
03EF₁₆	Port P7 direction register	PD7	00₁₆
03F0₁₆	Port P8 register	P8	??₁₆
03F1₁₆	Port P9 register	P9	??₁₆
03F2₁₆	Port P8 direction register	PD8	00X000002
03F3₁₆	Port P9 direction register	PD9	00₁₆
03F4₁₆	Port P10 register	P10	??₁₆
03F5₁₆	Rserved register	RSVREG03F5	??₁₆
03F6₁₆	Port P10 direction register	PD10	00₁₆
03F7₁₆	Reserved register	RSVREG03F7	00₁₆
03F8₁₆	Reserved register	RSVREG03F8	??₁₆
03F9₁₆	Reserved register	RSVREG03F9	??₁₆
03FA₁₆	Reserved register	RSVREG03FA	00₁₆
03FB₁₆	Reserved register	RSVREG03FB	00₁₆
03FC₁₆	Pull-up control register 0	PUR0	00₁₆
03FD₁₆	Pull-up control register 1	PUR1	00000000₂ 00000010₂ (Note 2)
03FE₁₆	Pull-up control register 2	PUR2	00₁₆
03FF₁₆	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 XIN pin.
(3) Apply an “H” signal to the RESET pin.

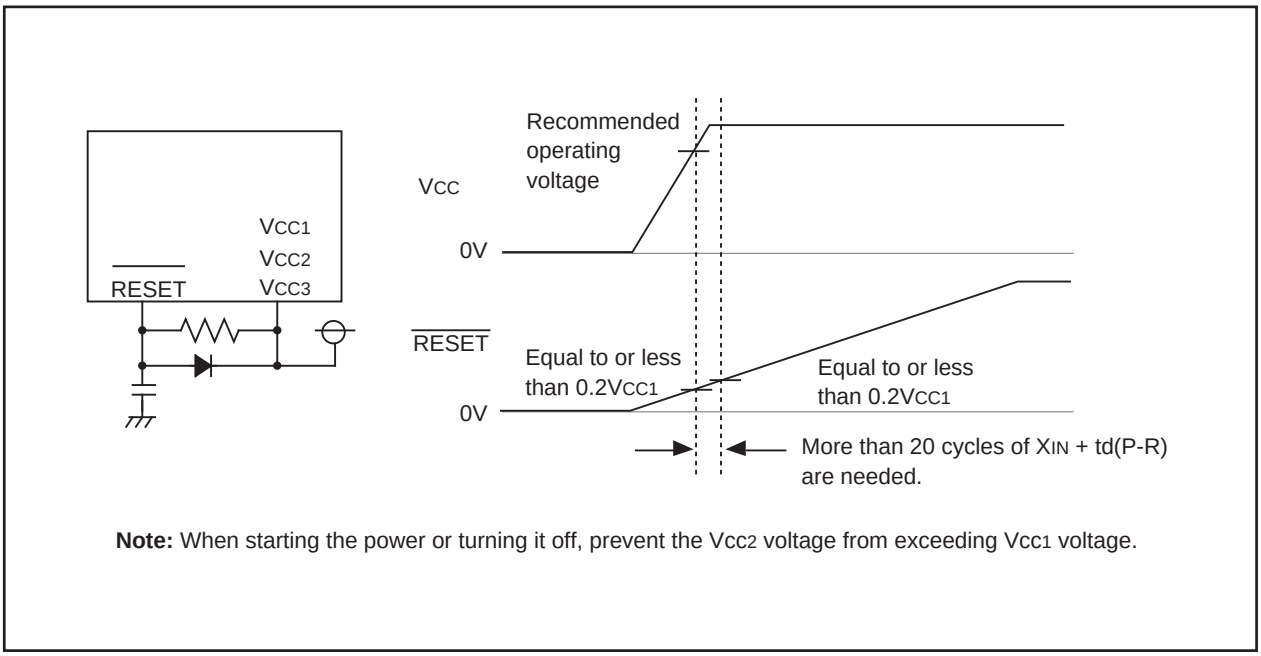
2. Power on
(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin.
(2) Let the power supply voltage increase until it meets the recommended operating condition.
(3) Wait td(P-R) or more until the internal power supply stabilizes.
(4) Supply a clock for 20 cycles or more to the XIN pin.
(5) Apply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

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.



Note: When starting the power or turning it off, prevent the V_{CC2} voltage from exceeding V_{CC1} voltage.

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.

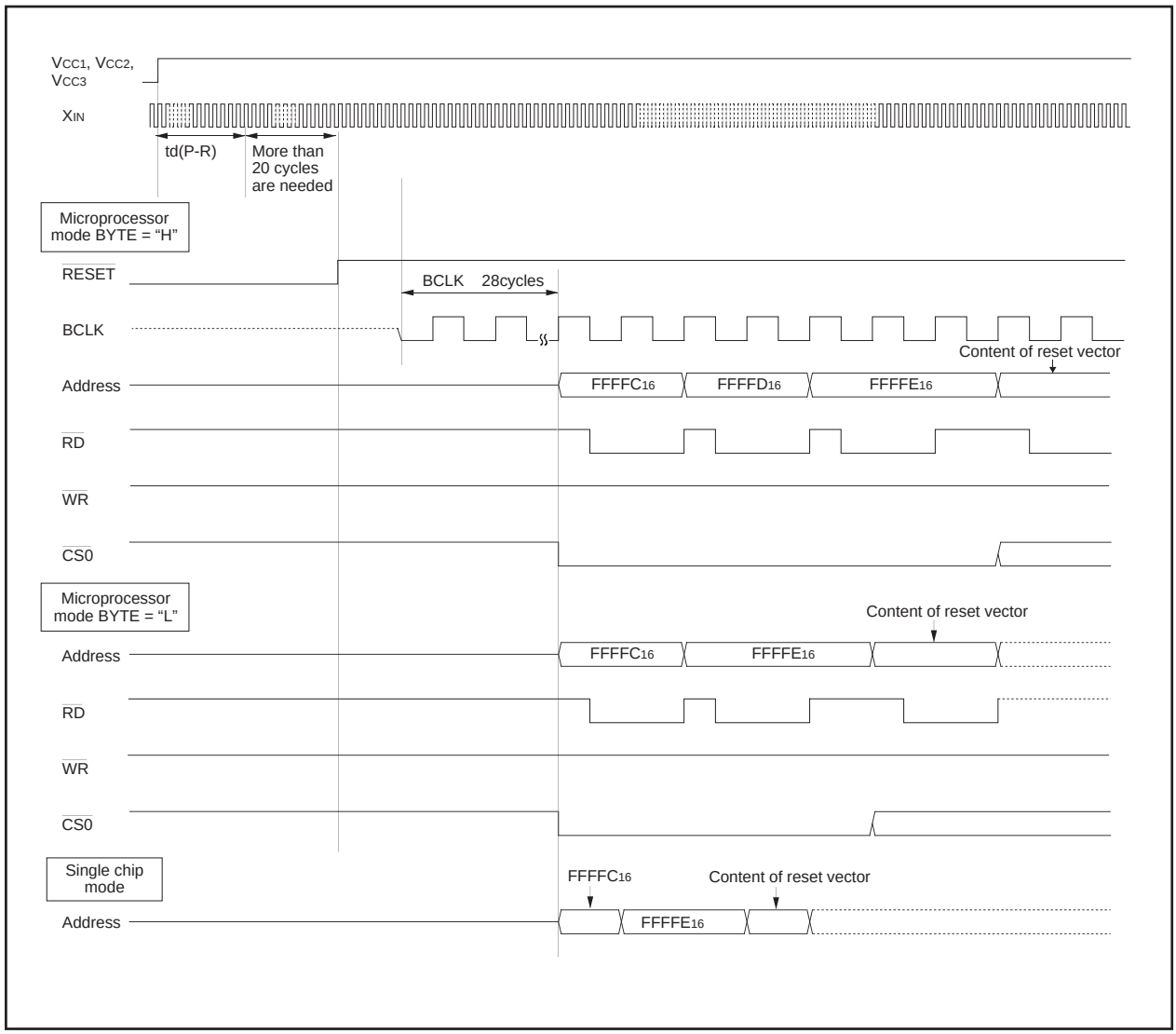


Figure 3.2. Reset sequence

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 $\overline{\text{HOLD}}$ pin)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)
P55	Input port	$\overline{\text{HOLD}}$ input	$\overline{\text{HOLD}}$ 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, OSCOUT, 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		

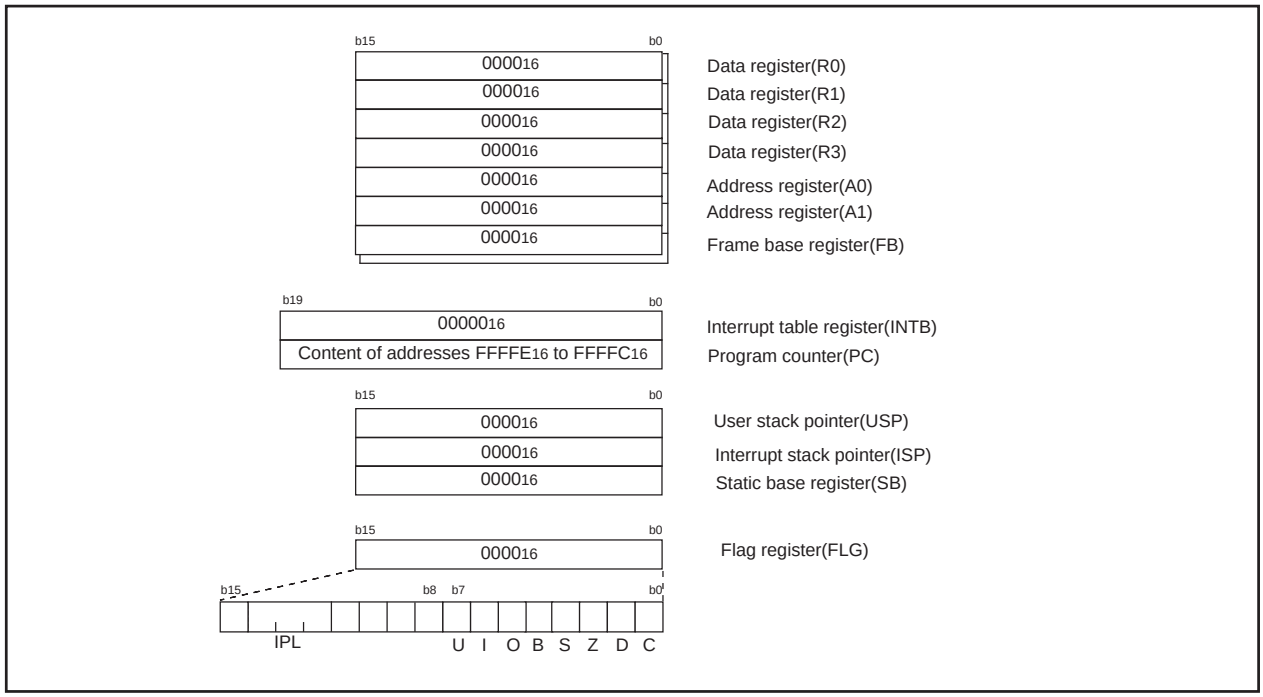


Figure 3.3. CPU Register Status After Reset

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	After reset
WDC	000F16	00XXXXXX2

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

Figure 3.4. WDC Register

Reserved register

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

Symbol: RSVREG0019 | Address: 0019$_{16}$ | After reset: 00001000$_{2}$

Bit symbol	Bit name	Function	RW
(b2-b0)	Reserved bit	Must set to “0”	RW
(b3)	Reserved bit	—	RO
(b7-b4)	Reserved bit	Must set to “0”	RW

Reserved register

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

Symbol: RSVREG001A | Address: 001A$_{16}$ | After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
(b7-b0)	Reserved bit	Must set to “0”	RW

Reserved register

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

Symbol: RSVREG001F | Address: 001F$_{16}$ | After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
(b5-b0)	Reserved bit	Must set to “0”	RW
(b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

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

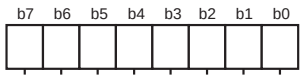
PM01 to PM00 bits	Processor modes
00_2	Single-chip mode
01_2	Memory expansion mode
10_2	Must not be set
11_2	Microprocessor mode

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 “01_2” (memory expansion mode) or “11_2” (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.

Processor mode register 0 (Note 1)



Symbol	Address	After reset (Note 4)
PM0	0004_{16}	00000000_2 (CNVSS1 pin = "L") 00000011_2 (CNVSS1 pin = "H")

Bit symbol	Bit name	Function	RW
PM00	Processor mode bit (Note 4)	b1 b0 0 0: Single-chip mode 0 1: Memory expansion mode 1 0: Must not be set 1 1: Microprocessor mode	RW
PM01			RW
PM02	R/W mode select bit (Note 2)	0 : $\overline{RD}$,$\overline{BHE}$,$\overline{WR}$ 1 : RD,WRH,WRL	RW
PM03	Software reset bit	Setting this bit to "1" resets the microcomputer. When read, its content is "0".	RW
PM04	Multiplexed bus space select bit (Note 2)	b5 b4 0 0 : Multiplexed bus is unused (Separate bus in the entire $\overline{CS}$ space) 0 1 : Allocated to $\overline{CS2}$ space 1 0 : Allocated to $\overline{CS1}$ space 1 1 : Allocated to the entire $\overline{CS}$ space (Note 3)	RW
PM05			RW
PM06	Port $P4_0$ to $P4_3$ function select bit (Note 2)	0 : Address output 1 : Port function (Address is not output)	RW
PM07	BCLK output disable bit (Note 2)	0 : BCLK is output 1 : BCLK is not output (Pin is left high-impedance)	RW

Note 1: Write to this register after setting the PRC1 bit in the PRCR register to "1" (write enable).
Note 2: Effective when the PM01 to PM00 bits are set to "01_2" (memory expansion mode) or "11_2" (microprocessor mode).
Note 3: To set the PM01 to PM00 bits are "01_2" and the PM05 to PM04 bits are "11_2" (multiplexed bus assigned to the entire $\overline{CS}$ space), apply an "H" signal to the BYTE pin (external data bus is 8 bits wide). While the CNVSS1 pin is held "H" (= V_{CC1}), do not rewrite the PM05 to PM04 bits to "11_2" after reset.
If the PM05 to PM04 bits are set to "11_2" during memory expansion mode, $P3_1$ to $P3_7$ and $P4_0$ to $P4_3$ become I/O ports, in which case the accessible area for each $\overline{CS}$ is 256 bytes.
Note 4: The PM01 to PM00 bits do not change at software reset and watchdog timer reset.

Figure 4.1. PM0 Register

Processor mode register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0						

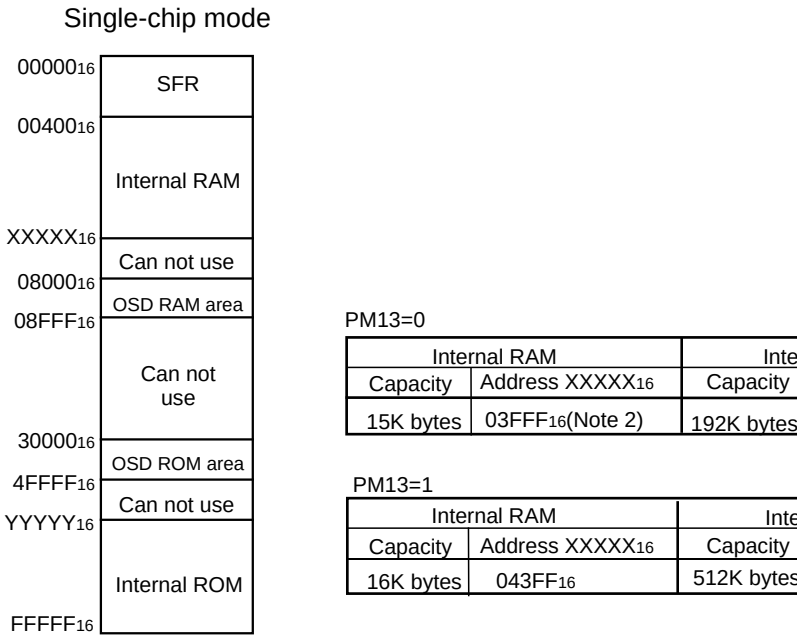
Symbol	Address	After reset
PM1	0005_{16}	$0X001000_2$

Bit symbol	Bit name	Function	RW
PM10	CS2 area switch bit (data block enable bit) (Note 2)	0: 09000_{16} to $26FFF_{16}$ (block A disable) 1: 10000_{16} to $26FFF_{16}$ (block A enable)	RW
PM11	Port $P3_7$ to $P3_4$ function select bit (Note 3)	0 : Address output 1 : Port function	RW
PM12	Watchdog timer function select bit	0 : Watchdog timer interrupt 1 : Watchdog timer reset (Note 4)	RW
PM13	Internal reserved area expansion bit	See Note 6	RW
PM14	Memory area expansion bit (Note 3)	b5 b4 0 0 : 1 Mbyte mode (Do not expand) 0 1 : Must not be set 1 0 : Must not be set 1 1 : 4 Mbyte mode	RW
PM15			RW
$\overline{\text{(b6)}}$	Reserved bit	Should be set to “0”.	RW
PM17	Wait bit (Note 5)	0 : No wait state 1 : With wait state (1 wait)	RW

Note 1: Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable).
Note 2: For the mask ROM version, this bit must be set to “0” . For the flash memory version, the PM10 bit also controls block A by enabling or disabling it. However, the PM10 bit is automatically set to “1” when the FMR0 1 bit in the FMR0 register is “1” (CPU rewrite mode).
Note 3: Effective when the PM01 to PM00 bits are set to “01_2” (memory expansion mode) or “11_2” (microprocessor mode).
Note 4: PM12 bit is set to “1” by writing a “1” in a program. (Writing a “0” has no effect.)
Note 5: When PM17 bit is set to “1” (with wait state), one wait state is inserted when accessing the internal RAM, internal ROM, or an external area. If the CSiW bit (i = 0 to 3) in the CSR register is “0” (with wait state), the CSi area is always accessed with one or more wait states regardless of whether the PM17 bit is set or not. Where the $\overline{\text{RDY}}$ signal is used or multiplex bus is used, set the CSiW bit to “0” (with wait state).
Note 6: The PM13 bit is automatically set to “1” when the FMR01 bit in the FMR0 register is “1” (CPU rewrite mode).
Note 7: The access area is changed by the PM13 bit as listed in the table below.

Access area		PM13=0	PM13=1
Internal	RAM	Up to addresses 00400_{16} to $03FFF_{16}$ (15 Kbytes)	The entire area is usable
	ROM	Up to addresses $D0000_{16}$ to $FFFFF_{16}$ (192 Kbytes)	The entire area is usable
External		Addresses 04000_{16} to $07FFF_{16}$ are usable Addresses 80000_{16} to $CFFFF_{16}$ are usable	Addresses 04000_{16} to $07FFF_{16}$ are reserved Addresses 80000_{16} to $CFFFF_{16}$ are reserved

Figure 4.2. PM1 Register



PM13=0

Internal RAM		Internal ROM	
Capacity	Address XXXXX$_{16}$	Capacity	Address YYYYY16
15K bytes	03FFF$_{16}$(Note 2)	192K bytes	D0000$_{16}$(Note 2)

PM13=1

Internal RAM		Internal ROM	
Capacity	Address XXXXX$_{16}$	Capacity	Address YYYYY$_{16}$
16K bytes	043FF$_{16}$	512K bytes	80000$_{16}$

Note 1: For the mask ROM version, set the PM10 bit to “0” (08000$_{16}$ to 26FFF$_{16}$ for $\overline{CS2}$ area).
Note 2: If PM13 bit is set to “0”, 15 Kbytes of the internal RAM and 192 Kbytes of the internal ROM can be used.

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 A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}/\overline{WR}$, $\overline{WRH}/\overline{BHE}$, 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

D_0 to D_7 and A_1 to A_8 are multiplexed. D_8 to D_{15} are not multiplexed. Do not use D_8 to D_{15}.

External buses connecting to a multiplexed bus are allocated to only the even addresses of the microcomputer. 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, A_0 to A_{19}. 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	$P3_4$ to $P3_7$ $P4_0$ to $P4_3$	12 bits
PM11=0 PM06=1	A_{12} to A_{15} $P4_0$ to $P4_3$	16 bits
PM11=0 PM06=0	A_{12} to A_{15} A_{16} to A_{19}	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 D_0 to D_{15} 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 $\overline{CSi}$ (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 $\overline{CSi}$ signal which is output from the $\overline{CSi}$ pin. During 4 Mbyte mode, $\overline{CSi}$ signal or bank number is output from the $\overline{CSi}$ pin. Refer to "Memory space expansion function". Figure 4.5 shows the example of address bus and $\overline{CSi}$ signal output in 1 Mbyte mode.

Chip select control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CSR | Address: 0008_{16} | After reset: 01_{16}

Bit symbol	Bit name	Function	RW
CS0	$\overline{CS0}$ output enable bit	0 : Chip select output disabled (functions as I/O port) 1 : Chip select output enabled	RW
CS1	$\overline{CS1}$ output enable bit		RW
CS2	$\overline{CS2}$ output enable bit		RW
CS3	$\overline{CS3}$ output enable bit		RW
CS0W	$\overline{CS0}$ wait bit	0 : With wait state 1 : Without wait state (Note 1, Note 2, Note 3)	RW
CS1W	$\overline{CS1}$ wait bit		RW
CS2W	$\overline{CS2}$ wait bit		RW
CS3W	$\overline{CS3}$ wait bit		RW

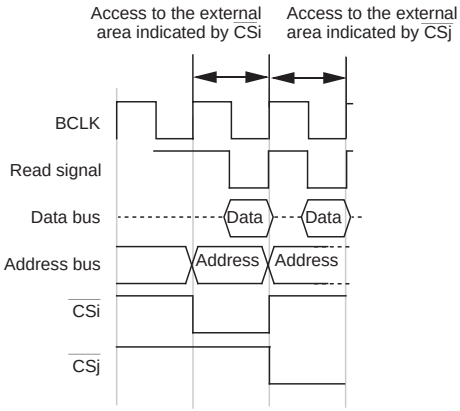
Notes 1: Where the $\overline{RDY}$ signal is used in the area indicated by $\overline{CSi}$ (i = 0 to 3) or the multiplex bus is used, set the CSiW bit to "0" (Wait state).
2: When the PM17 bit in the PM1 register is set to "1" (wait), set the CSiW bit to "1" (wait).
3: When the CSiW bit = "0" (with wait state), the number of wait states (interms of clock cycles) can be selected using the CSEi1W to CSEi0W bits in the CSE register.

Figure 4.4. CSR Register

Example 1

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

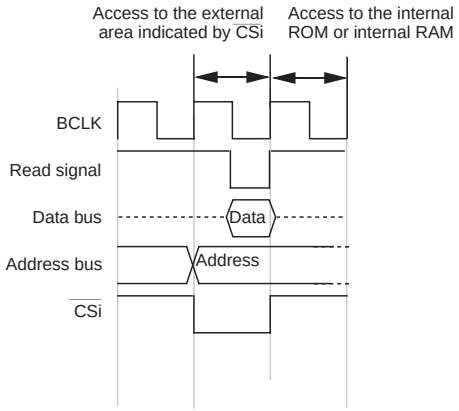
The address bus and the chip select signal both change state between these two cycles.



Example 2

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

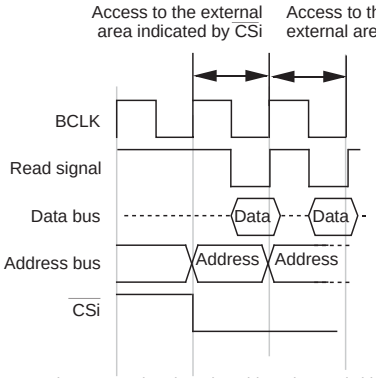
The chip select signal changes state but the address bus does not change state



Example 3

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

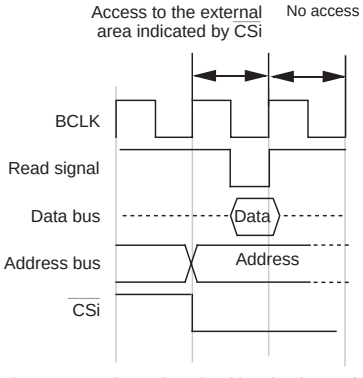
The address bus changes state but the chip select signal does not change state



Example 4

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

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



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

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

Figure 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.

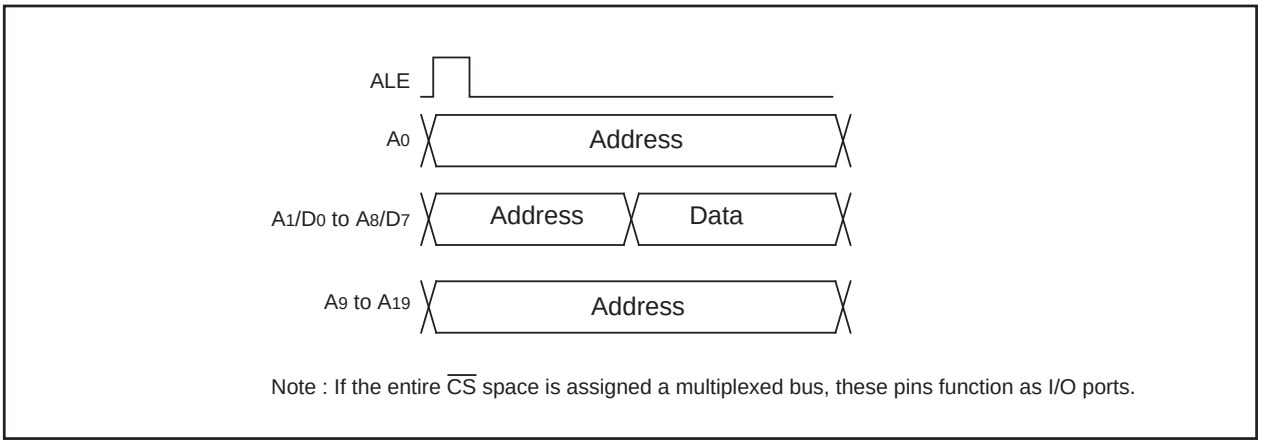


Figure 4.6. ALE Signal, Address Bus, Data Bus

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

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

A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{WR}$, $\overline{BHE}$, ALE, $\overline{HLDA}$

Then, when the input on the $\overline{RDY}$ 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 $\overline{RDY}$ signal. To use the $\overline{RDY}$ signal, set the corresponding bit (CS3W to CS0W bits) in the CSR register to "0" (with wait state). When not using the $\overline{RDY}$ signal, process the $\overline{RDY}$ pin as an unused pin.

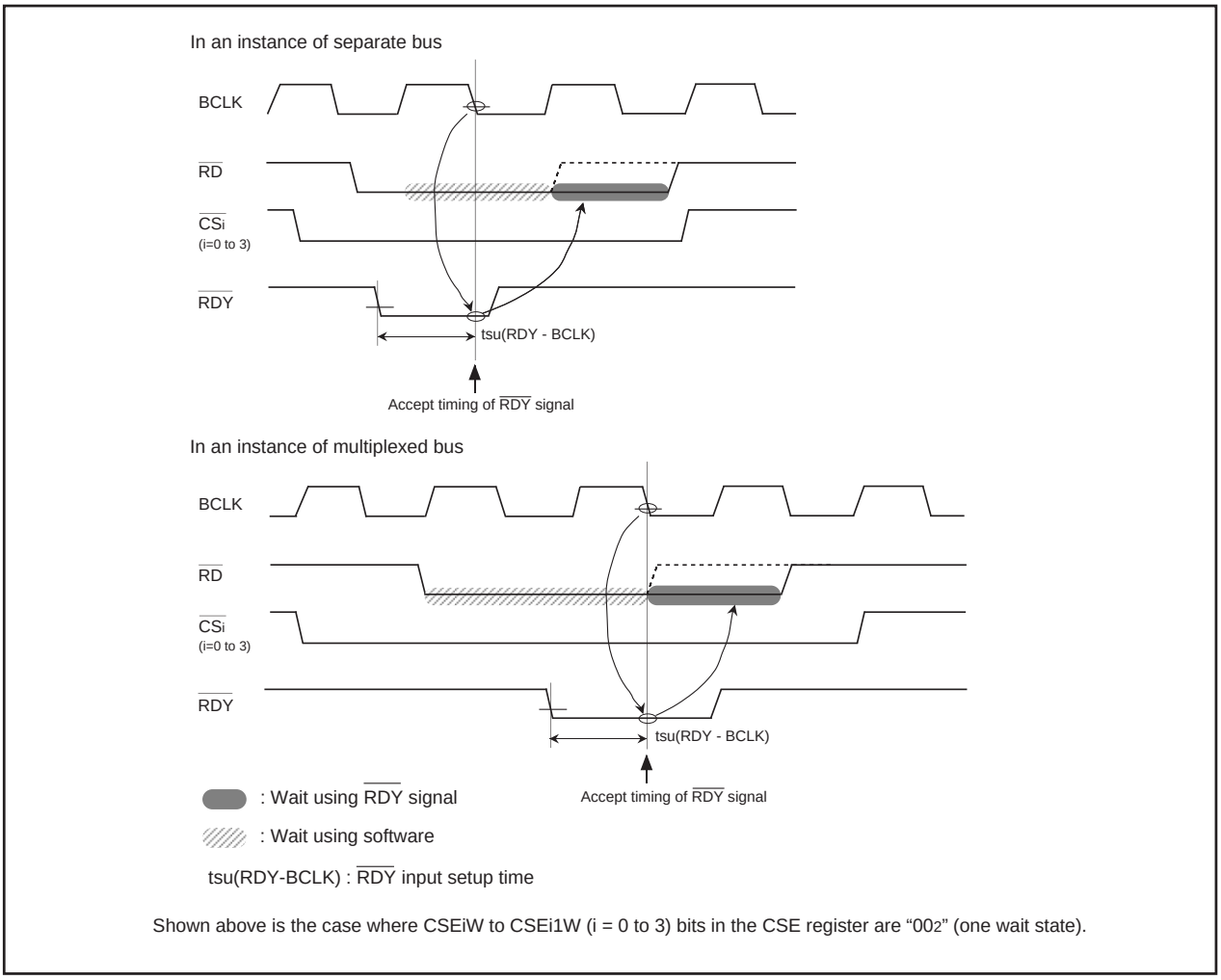


Figure 4.7. Example in which Wait State was Inserted into Read Cycle by $\overline{RDY}$ Signal

(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.

$\overline{\text{HOLD}}$ > DMAC > 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 $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$		High-impedance
I/O ports	P0, P1, P3, P4(Note 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		00_2(separate bus)	01_2($\overline{CS2}$ is for multiplexed bus and others are for separate bus) 10_2($\overline{CS1}$ is for multiplexed bus and others are for separate bus)
Data bus width BYTE pin		16 bits "L"	16 bits "L"
$P0_0$ to $P0_7$		D_0 to D_7	D_0 to D_7(Note)
$P1_0$ to $P1_7$		D_8 to D_{15}	D_8 to D_{15}(Note)
$P2_0$		A_0	A_0
$P2_1$ to $P2_7$		A_1 to A_7	A_1 to A_7/D_0 to D_6 (Note 2)
$P3_0$		A_8	A_8/D_7(Note 2)
$P3_1$ to $P3_3$		A_9 to A_{11}	
$P3_4$ to $P3_7$	PM11=0	A_{12} to A_{15}	
	PM11=1	I/O ports	
$P4_0$ to $P4_3$	PM06=0	A_{16} to A_{19}	
	PM06=1	I/O ports	
$P4_4$	CS0=0	I/O ports	
	CS0=1	$\overline{CS0}$	
$P4_5$	CS1=0	I/O ports	
	CS1=1	$\overline{CS1}$	
$P4_6$	CS2=0	I/O ports	
	CS2=1	$\overline{CS2}$	
$P4_7$	CS3=0	I/O ports	
	CS3=1	$\overline{CS3}$	
$P5_0$	PM02=0	$\overline{WR}$	
	PM02=1	$\overline{WRL}$	$\overline{WRL}$
$P5_1$	PM02=0	$\overline{BHE}$	
	PM02=1	$\overline{WRH}$	$\overline{WRH}$
$P5_2$		$\overline{RD}$	
$P5_3$		BCLK	
$P5_4$		$\overline{HLDA}$	
$P5_5$		$\overline{HOLD}$	
$P5_6$		ALE	
$P5_7$		$\overline{RDY}$	

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

Note 1: When V_{CC1} is inputted into $CNV_{SS}1$ pin, do not set bits PM05 and PM04 to "11_2" after reset.
Since $P3_1$ to $P3_7$, and $P4_0$ to $P4_3$ become I/O ports when bits PM05 snd PM04 are set to "11_2" in memory extension mode, the area which can be accessed is 256 bytes per $\overline{CS}$.

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
A_0 to A_{19}		Address output	Maintain status before accessed address of external area or SFR
D_0 to D_{15}	When read	High-impedance	High-impedance
	When write	Output data	Undefined
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$ output	Output “H”
$\overline{BHE}$		$\overline{BHE}$ output	Maintain status before accessed status of external area or SFR
$\overline{CS0}$ to $\overline{CS3}$		Output “H”	Output “H”
ALE		Output “L”	Output “L”

(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 $\overline{RDY}$ 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.

Chip select expansion control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CSE — Address: $001B_{16}$ — After reset: 00_{16}

Bit symbol	Bit name	Function	RW
CSE00W	$\overline{CS0}$ wait expansion bit (Note)	b1 b0 0 0: 1 wait 0 1: 2 waits 1 0: 3 waits 1 1: Must not be set	RW
CSE01W			RW
CSE10W	$\overline{CS1}$ wait expansion bit (Note)	b3 b2 0 0: 1 wait 0 1: 2 waits 1 0: 3 waits 1 1: Must not be set	RW
CSE11W			RW
CSE20W	$\overline{CS2}$ wait expansion bit (Note)	b5 b4 0 0: 1 wait 0 1: 2 waits 1 0: 3 waits 1 1: Must not be set	RW
CSE21W			RW
CSE30W	$\overline{CS3}$ wait expansion bit (Note)	b7 b6 0 0: 1 wait 0 1: 2 waits 1 0: 3 waits 1 1: Must not be set	RW
CSE31W			RW

Note: Set the CSiW bit (i = 0 to 3) in the CSR register to “0” (with wait state) before writing to the CSEi1W to CSEi0W bits. If the CSiW bit needs to be set to “1” (without wait state), set the CSEi1W to CSEi0W bits to “00_2” before setting it.

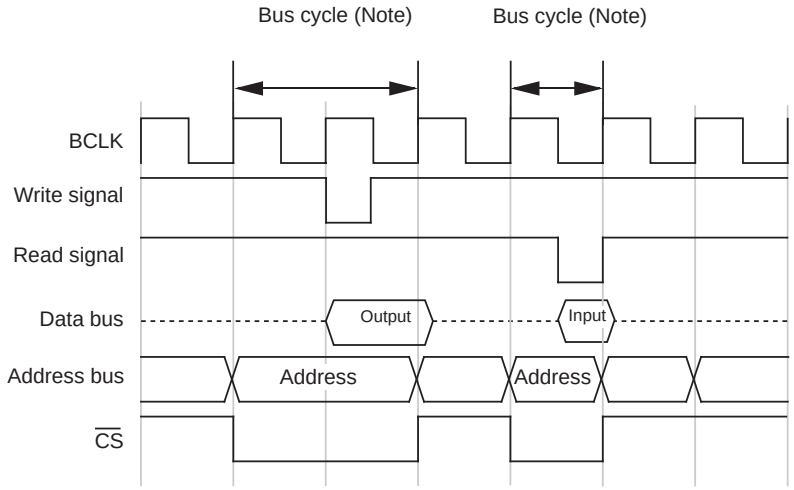
Figure 4.9. CSE Register

Table 4.10. Bit and Bus Cycle Related to Software Wait

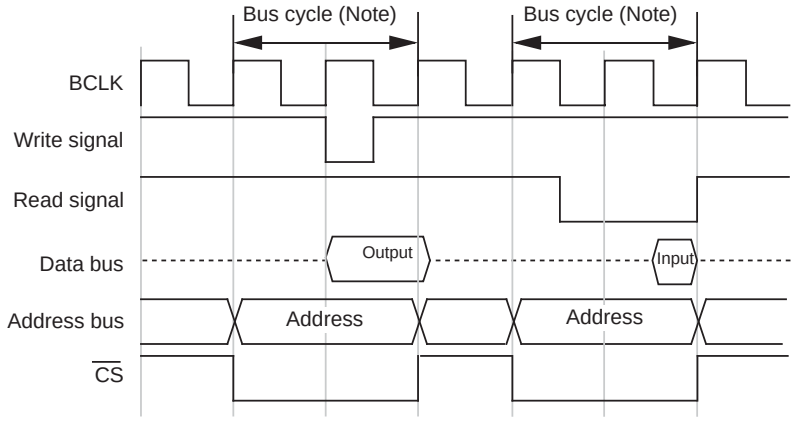
Area	Bus mode	PM1 register PM17 bit	CSR register CS3W bit (Note 1) CS2W bit (Note 1) CS1W bit (Note 1) CS0W bit (Note 1)	CSE register CSE31W to CSE30W bit CSE21W to CSE20W bit CSE11W to CSE10W bit CSE01W to CSE00W bit	Software wait	Bus cycle
SFR	—	—	—	—	—	2 BCLK cycle
Internal RAM, ROM	—	0	—	—	No wait	1 BCLK cycle (Note 3)
	—	1	—	—	1 wait	2 BCLK cycles
External area	Separate bus	0	1	00_2	No wait	1 BCLK cycle (read)
						2 BCLK cycles (write)
		—	0	00_2	1 wait	2 BCLK cycles (Note 3)
		—	0	01_2	2 waits	3 BCLK cycles
		—	0	10_2	3 waits	4 BCLK cycles
		1	1	00_2	1 wait	2 BCLK cycles
	Multiplexed bus (Note 2)	—	0	00_2	1 wait	3 BCLK cycles
		—	0	01_2	2 waits	3 BCLK cycles
		—	0	10_2	3 waits	4 BCLK cycles
		1	0	00_2	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 "00_{16}" (one wait state for $\overline{CS0}$ to $\overline{CS3}$). 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).

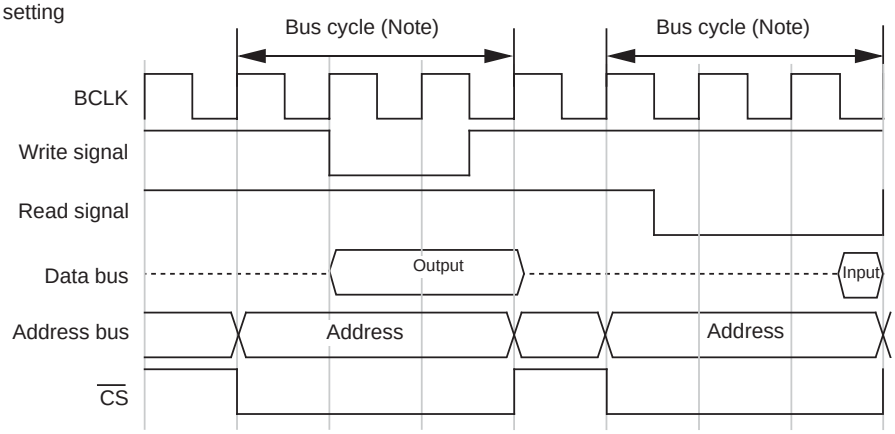
(1) Separate bus, No wait setting



(2) Separate bus, 1-wait setting



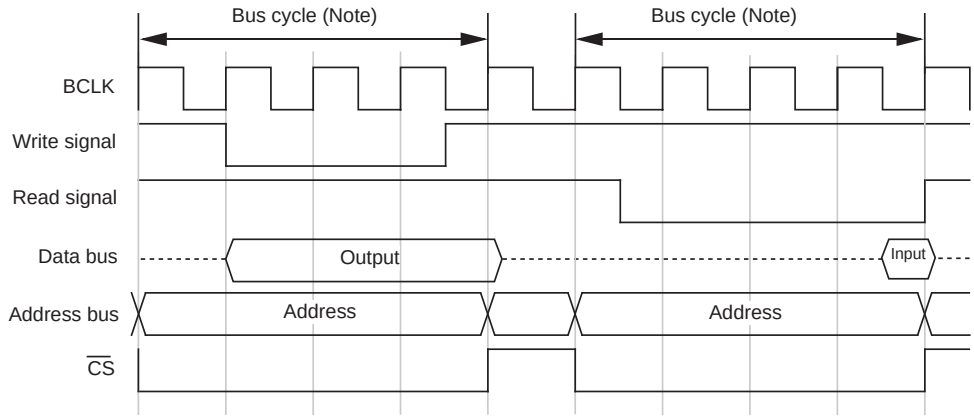
(3) Separate bus, 2-wait setting



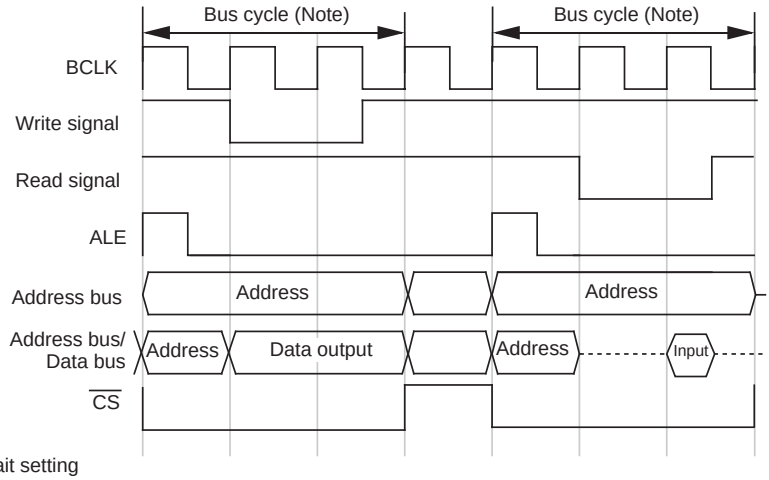
Note : These example timing charts indicate bus cycle length. After this bus cycle sometimes come read and write cycles in succession.

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

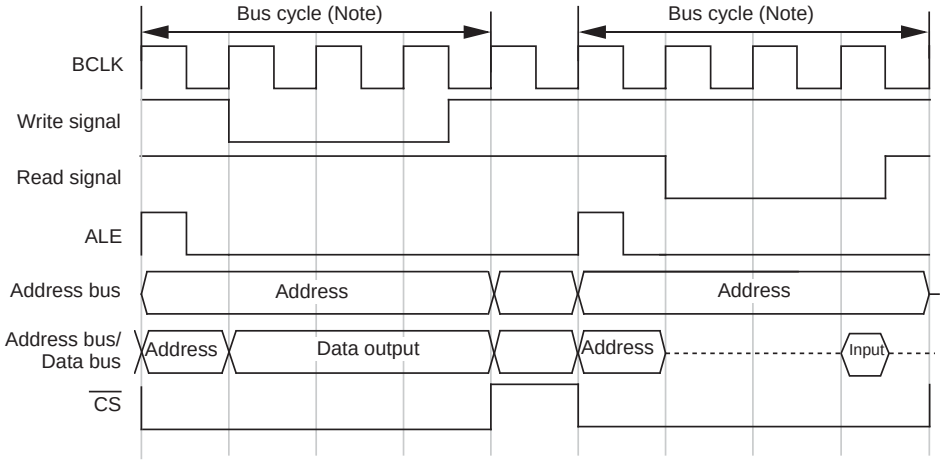
(1) Separate bus, 3-wait setting



(2)Multiplexed bus, 1- or 2-wait setting



(3)Multiplexed bus, 3-wait setting



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

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 $\overline{CSi}$ (i = 0 to 3) signals (hereafter referred to as the $\overline{CSi}$ area). Figures 4.13 to 4.14 show the memory mapping and $\overline{CS}$ area in 1 Mbyte mode.

(2) 4 Mbyte Mode

In this mode, the memory space is 4 Mbytes. Figure 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 40000_{16}.

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

Addresses 0400_{16} to $3FFFF_{16}$, $C000_{16}$ to $FFFFF_{16}$

- The $\overline{CSi}$ signal is output from the $\overline{CSi}$ pin (same operation as 1 Mbyte mode. However the last address of $\overline{CS1}$ area is $3FFFF_{16}$)

Addresses 40000_{16} to $BFFFF_{16}$

- The $\overline{CS0}$ pin outputs “L”
- The $\overline{CS1}$ to $\overline{CS3}$ 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 $\overline{CS}$ area in 4 Mbyte mode. Note that banks 0 to 6 are data-only areas. Locate the program in bank 7 or the $\overline{CSi}$ area.

Data bank register (Note)

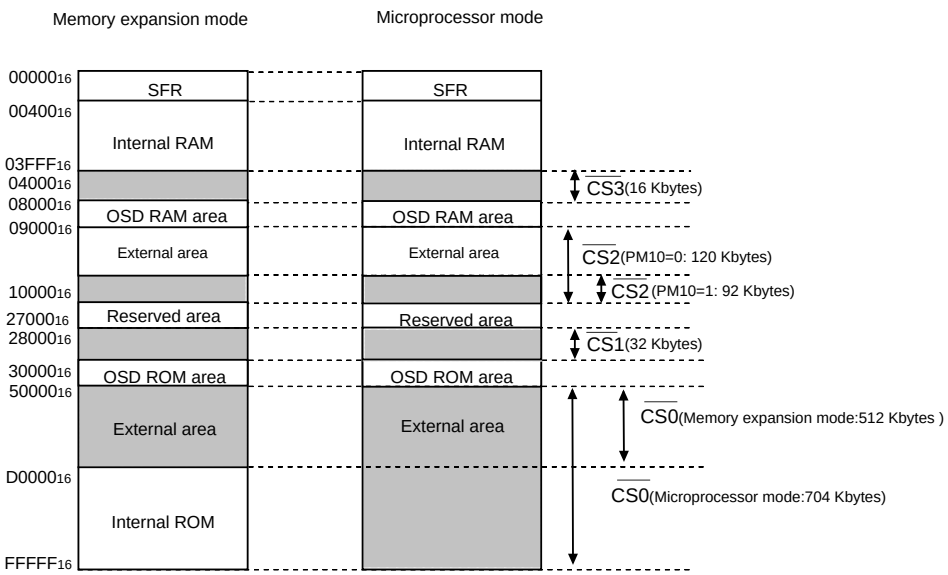
b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After reset
DBR	$000B_{16}$	00_{16}

Bit symbol	Bit name	Description	RW
— (b1-b0)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—
OFS	Offset bit	0: Not offset 1: Offset	RW
BSR0	Bank selection bits	b5 b4 b3 0 0 0: Bank 0 0 0 1: Bank 1 0 1 0: Bank 2 0 1 1: Bank 3 b5 b4 b3 1 0 0: Bank 4 1 0 1: Bank 5 1 1 0: Bank 6 1 1 1: Bank 7	RW
BSR1			RW
BSR2			RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

Note : Effective when the PM01 to PM00 bits are set to “01_2” (memory expansion mode) or “11_2” (microprocessor mode).

Figure 4.12. DBR Register

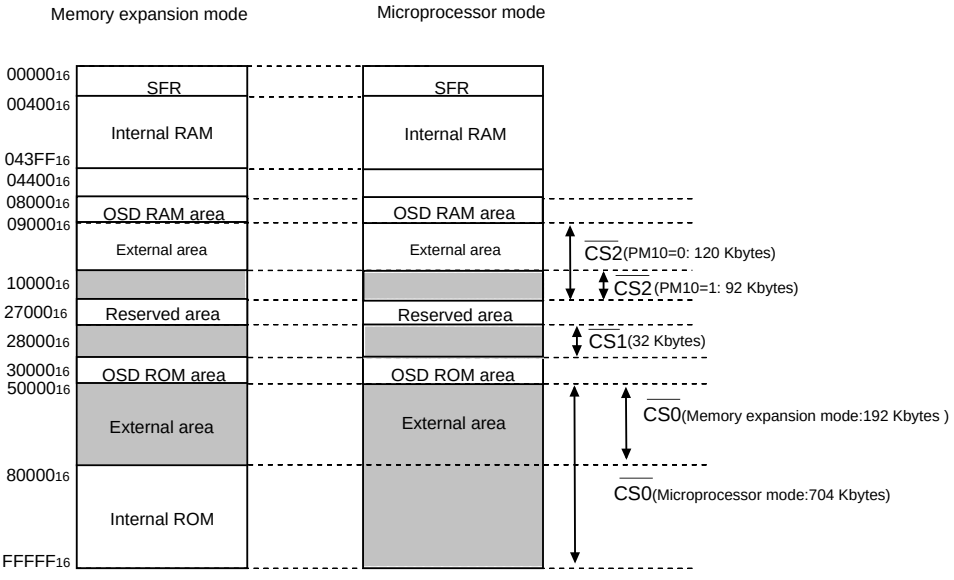


PM13=0

Internal RAM (Note 1)		Internal ROM (Note 1)		External area			
Capacity	Address	Capacity	Address	$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$	$\overline{CS3}$
15 Kbytes	00400₁₆ to 03FFF₁₆	192 Kbytes	D0000₁₆ to FFFFF₁₆	Memory expansion mode 50000₁₆–CFFFF₁₆ Microprocessor mode 50000₁₆–FFFFF₁₆	28000₁₆–2FFFF₁₆	When PM10=0 09000₁₆–26FFF₁₆ When PM10=1 10000₁₆–26FFF₁₆	04000₁₆–07FFF₁₆

Note : If PM13 bit is set to “0”, 15 Kbytes of the internal RAM and 192 Kbytes of the internal ROM can be used.

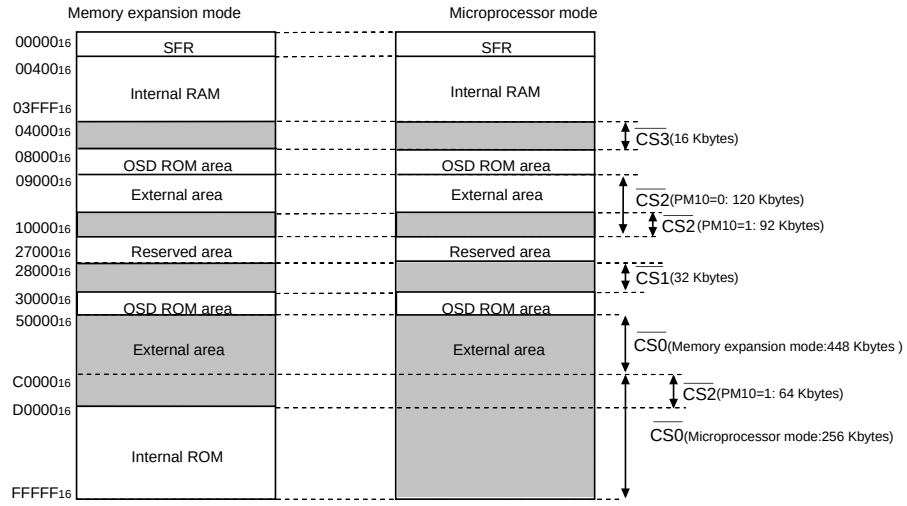
Figure 4.13. Memory Mapping and $\overline{CS}$ Area in 1 Mbyte Mode (PM13=0)



PM13=1

Internal RAM		Internal ROM		External area			
Capacity	Address	Capacity	Address	$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$	$\overline{CS3}$
16 Kbytes	00400₁₆ to 043FF₁₆	512 Kbytes	80000₁₆ to FFFFF₁₆	Memory expansion mode 50000₁₆–7FFFF₁₆ Microprocessor mode 50000₁₆–FFFFF₁₆	28000₁₆–2FFFF₁₆	When PM10=0 09000₁₆–26FFF₁₆ When PM10=1 10000₁₆–26FFF₁₆	No area

Figure 4.14. Memory Mapping and $\overline{CS}$ Area in 1 Mbyte Mode (PM13=1)

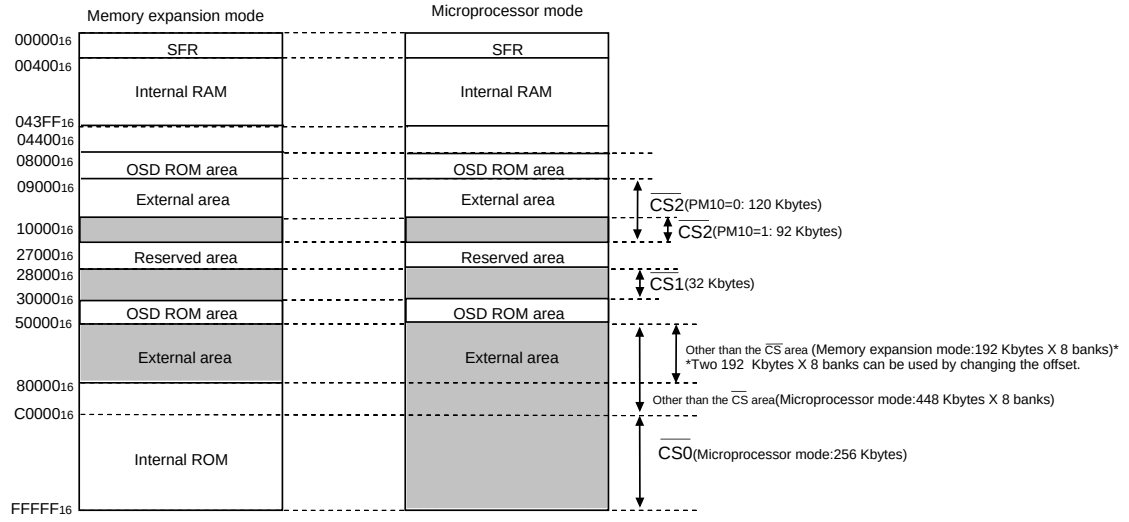


PM13=0

Internal RAM (Note 2)		Internal ROM (Note 2)		External area				
Capacity	Address	Capacity	Address	CS0	CS1	CS2	CS3	Other than the CS area (Note 1)
15 Kbytes	00400₁₆ to 03FFF₁₆	192 Kbytes	D0000₁₆ to FFFFF₁₆	Memory expansion mode C0000₁₆–CFFFF₁₆ Microprocessor mode C0000₁₆–FFFFF₁₆	28000₁₆–2FFFF₁₆	When PM10=0 09000₁₆–26FFF₁₆ When PM10=1 10000₁₆–26FFF₁₆	04000₁₆–07FFF₁₆	50000₁₆–BFFFF₁₆

Note 1: The CS0 pin outputs a low signal, and the CS1 - CS3 pins output a bank number.
Note 2: If PM13 bit is set to "0", 15 Kbytes of the internal RAM and 192 Kbytes of the internal ROM can be used.

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



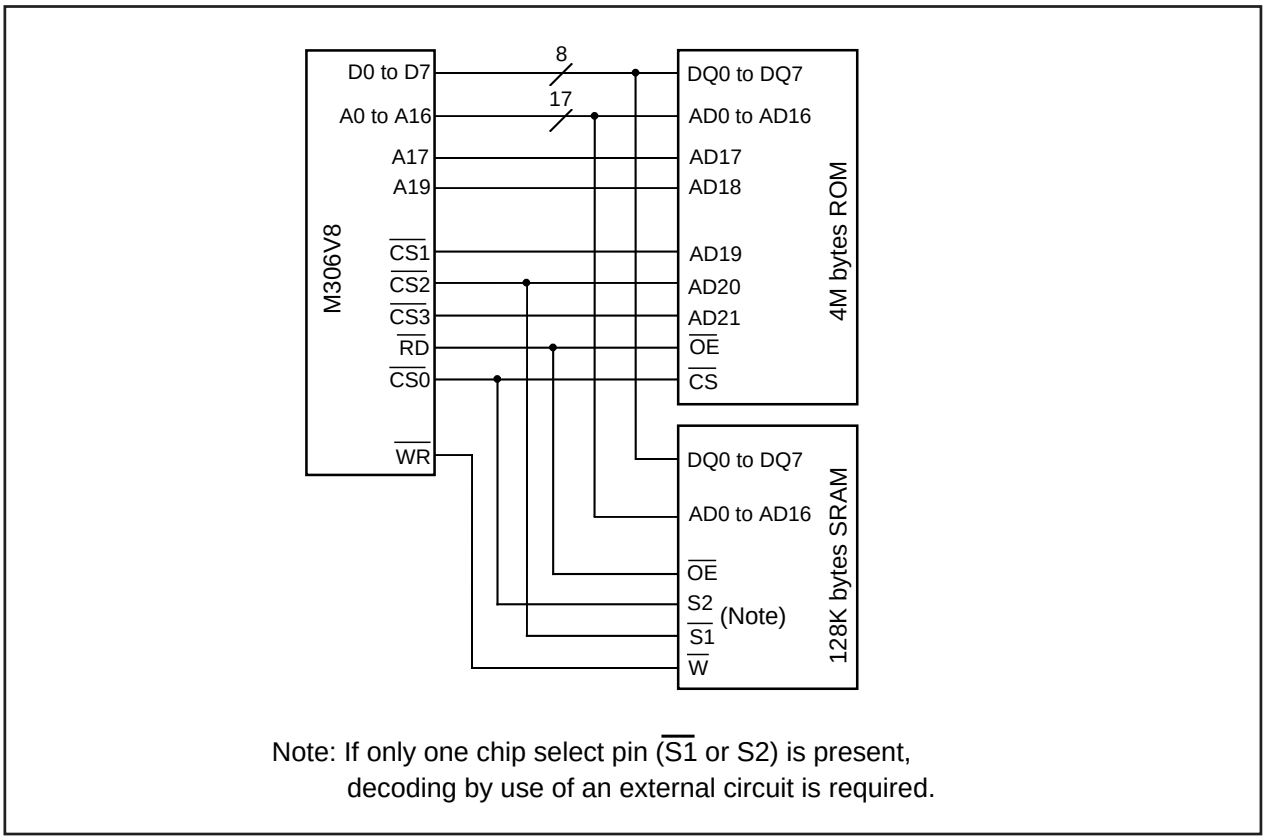
PM13=1

Internal RAM		Internal ROM		External area				
Capacity	Address	Capacity	Address	CS0	CS1	CS2	CS3	Other than the CS area (Note 1)
16 Kbytes	00400₁₆ to 043FF₁₆	512 Kbytes	80000₁₆ to FFFFF₁₆	Microprocessor mode C0000₁₆–FFFFF₁₆	28000₁₆–2FFFF₁₆	When PM10=0 09000₁₆–26FFF₁₆ When PM10=1 10000₁₆–26FFF₁₆	No area	Memory expansion mode 50000₁₆–7FFFF₁₆ Microprocessor mode 50000₁₆–BFFFF₁₆

Note 1: The CS0 pin outputs a low signal, and the CS1 - CS3 pins output a bank number.

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.



Note: If only one chip select pin ($\overline{S1}$ or S2) is present,
decoding by use of an external circuit is required.

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

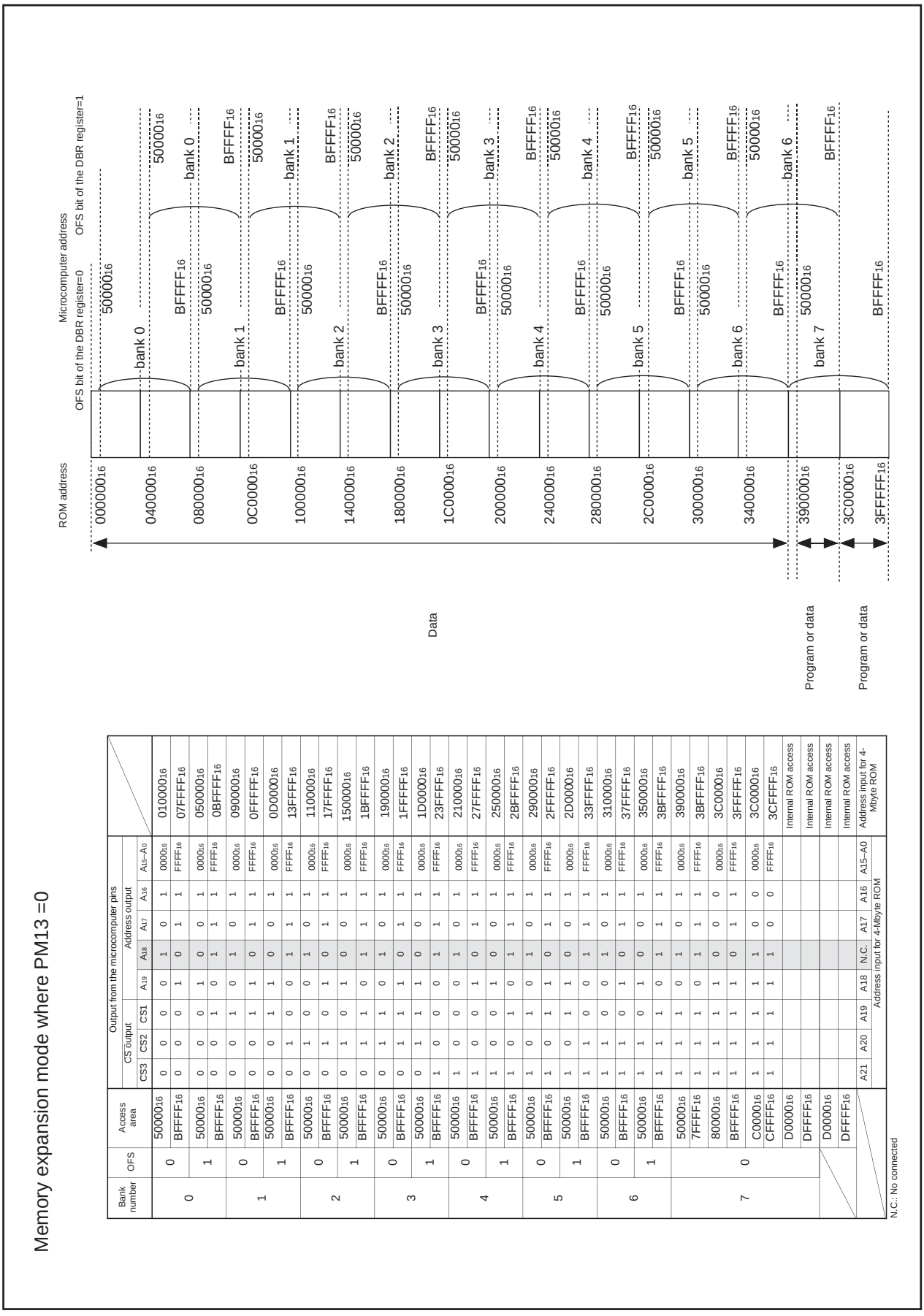


Figure 4.18. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (1)

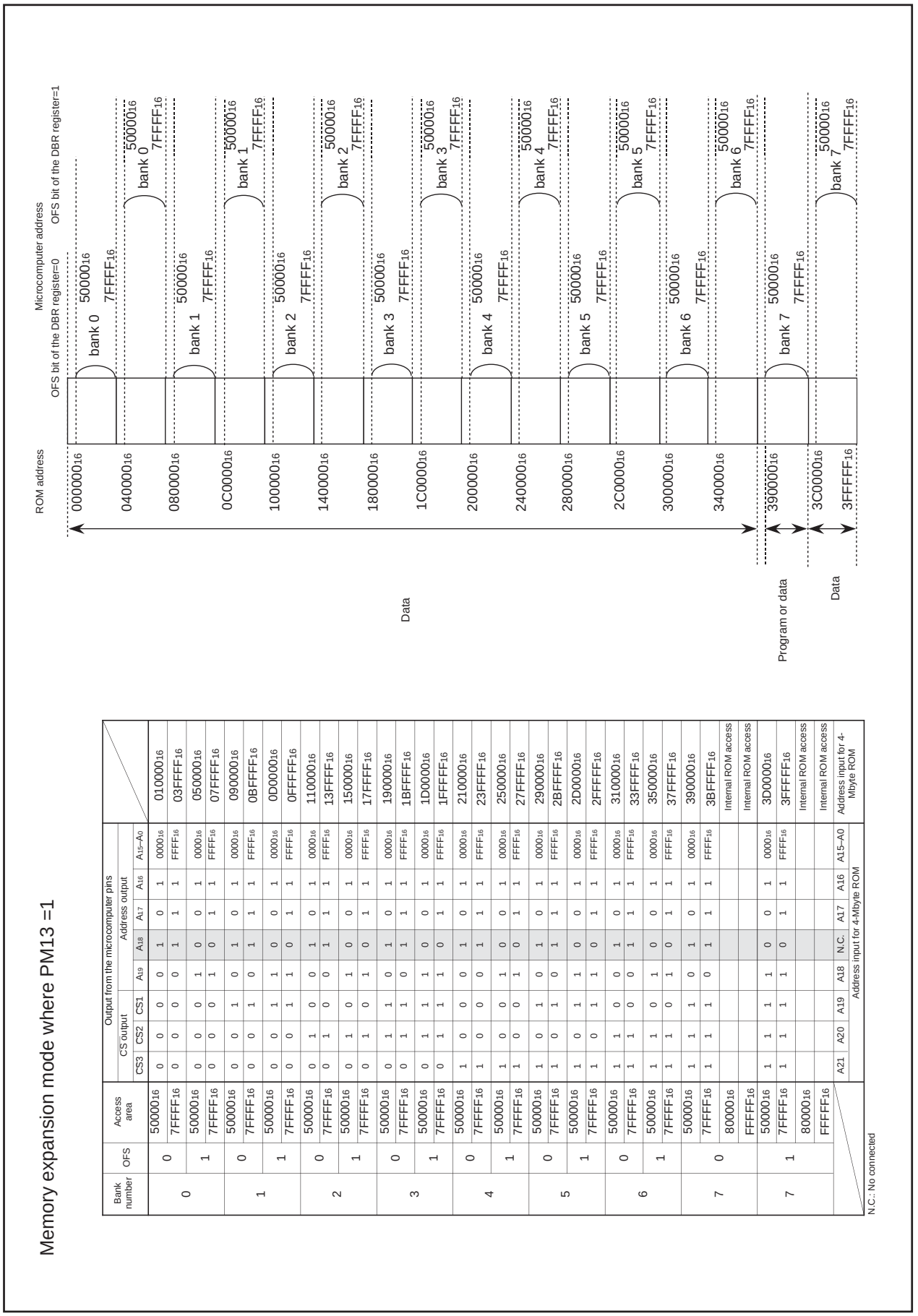
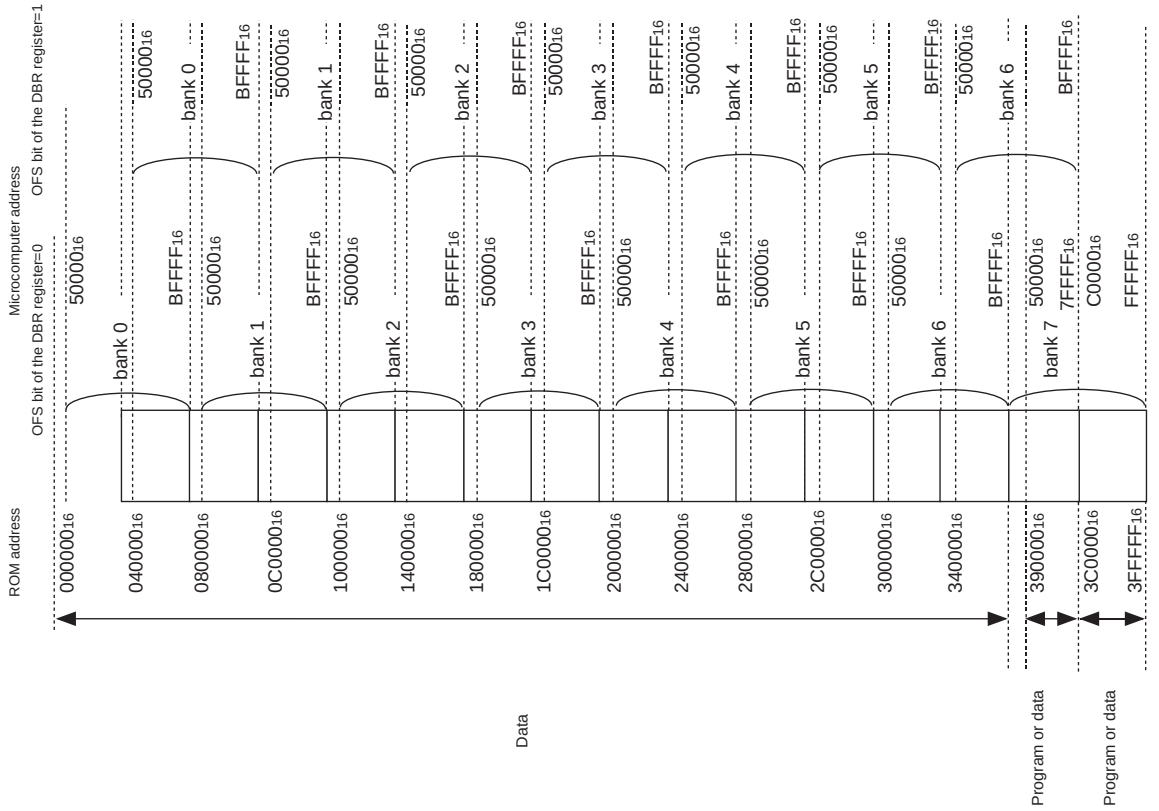


Figure 4.19. Relationship Between Addresses on 4-M Byte ROM and Those on Microcomputer (2)

Microprocessor mode



Bank number	OFS	Access area	Output from the microcomputer pins: CS output CS3	CS2	CS1	Address output A19	A18	A17	A16	A15–A0	
0	0	500000₁₆	0	0	0	0	1	0	1	0000₁₆	010000₁₆
		BFFFF₁₆	0	0	0	1	0	1	1	FFFF₁₆	07FFFF₁₆
	1	500000₁₆	0	0	0	1	0	0	1	0000₁₆	050000₁₆
		BFFFF₁₆	0	0	1	0	1	1	1	FFFF₁₆	0BFFFF₁₆
1	0	500000₁₆	0	0	1	0	1	0	1	0000₁₆	090000₁₆
		BFFFF₁₆	0	0	1	1	0	1	1	FFFF₁₆	0FFFFF₁₆
	1	500000₁₆	0	0	1	1	0	0	1	0000₁₆	0D0000₁₆
		BFFFF₁₆	0	1	0	0	1	1	1	FFFF₁₆	13FFFF₁₆
2	0	500000₁₆	0	1	0	0	1	0	1	0000₁₆	110000₁₆
		BFFFF₁₆	0	1	0	1	0	1	1	FFFF₁₆	17FFFF₁₆
	1	500000₁₆	0	1	0	1	0	0	1	0000₁₆	150000₁₆
		BFFFF₁₆	0	1	1	0	1	1	1	FFFF₁₆	1BFFFF₁₆
3	0	500000₁₆	0	1	1	0	1	0	1	0000₁₆	190000₁₆
		BFFFF₁₆	0	1	1	1	0	1	1	FFFF₁₆	1FFFFF₁₆
	1	500000₁₆	0	1	1	1	0	0	1	0000₁₆	1D0000₁₆
		BFFFF₁₆	1	0	0	0	1	1	1	FFFF₁₆	23FFFF₁₆
4	0	500000₁₆	1	0	0	0	1	0	1	0000₁₆	210000₁₆
		BFFFF₁₆	1	0	0	1	0	1	1	FFFF₁₆	27FFFF₁₆
	1	500000₁₆	1	0	0	1	0	0	1	0000₁₆	250000₁₆
		BFFFF₁₆	1	0	1	0	1	1	1	FFFF₁₆	2BFFFF₁₆
5	0	500000₁₆	1	0	1	0	1	0	1	0000₁₆	290000₁₆
		BFFFF₁₆	1	0	1	1	0	1	1	FFFF₁₆	2FFFFF₁₆
	1	500000₁₆	1	0	1	1	0	0	1	0000₁₆	2D0000₁₆
		BFFFF₁₆	1	1	0	0	1	1	1	FFFF₁₆	33FFFF₁₆
6	0	500000₁₆	1	1	0	0	1	0	1	0000₁₆	310000₁₆
		BFFFF₁₆	1	1	0	1	0	1	1	FFFF₁₆	37FFFF₁₆
	1	500000₁₆	1	1	0	1	0	0	1	0000₁₆	350000₁₆
		BFFFF₁₆	1	1	1	0	1	1	1	FFFF₁₆	3BFFFF₁₆
7	0	500000₁₆	1	1	1	0	1	0	1	0000₁₆	390000₁₆
		7FFFF₁₆	1	1	1	1	0	1	1	FFFF₁₆	3BFFFF₁₆
		800000₁₆	1	1	1	1	0	0	0	0000₁₆	3C0000₁₆
		BFFFF₁₆	1	1	1	1	0	1	1	FFFF₁₆	3FFFFF₁₆
		C0000₁₆	1	1	1	1	1	0	0	0000₁₆	3C0000₁₆
		FFFFF₁₆	1	1	1	1	1	1	1	FFFF₁₆	3FFFFF₁₆
			A21	A20	A19	A18	N.C.	A17	A16	A15–A0	Address input for 4-Mbyte ROM
			Address input for 4-Mbyte ROM								Address input for 4-Mbyte ROM

N.C.: No connected

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	• CPU clock source • Peripheral function clock source	•CPU clock source • Timer A, B's clock source
Clock frequency	16 MHz	32.768 kHz
Usable oscillator	• Ceramic oscillator • Crystal oscillator	• Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT
Oscillation stop, restart function	Presence	Presence
Oscillator status after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

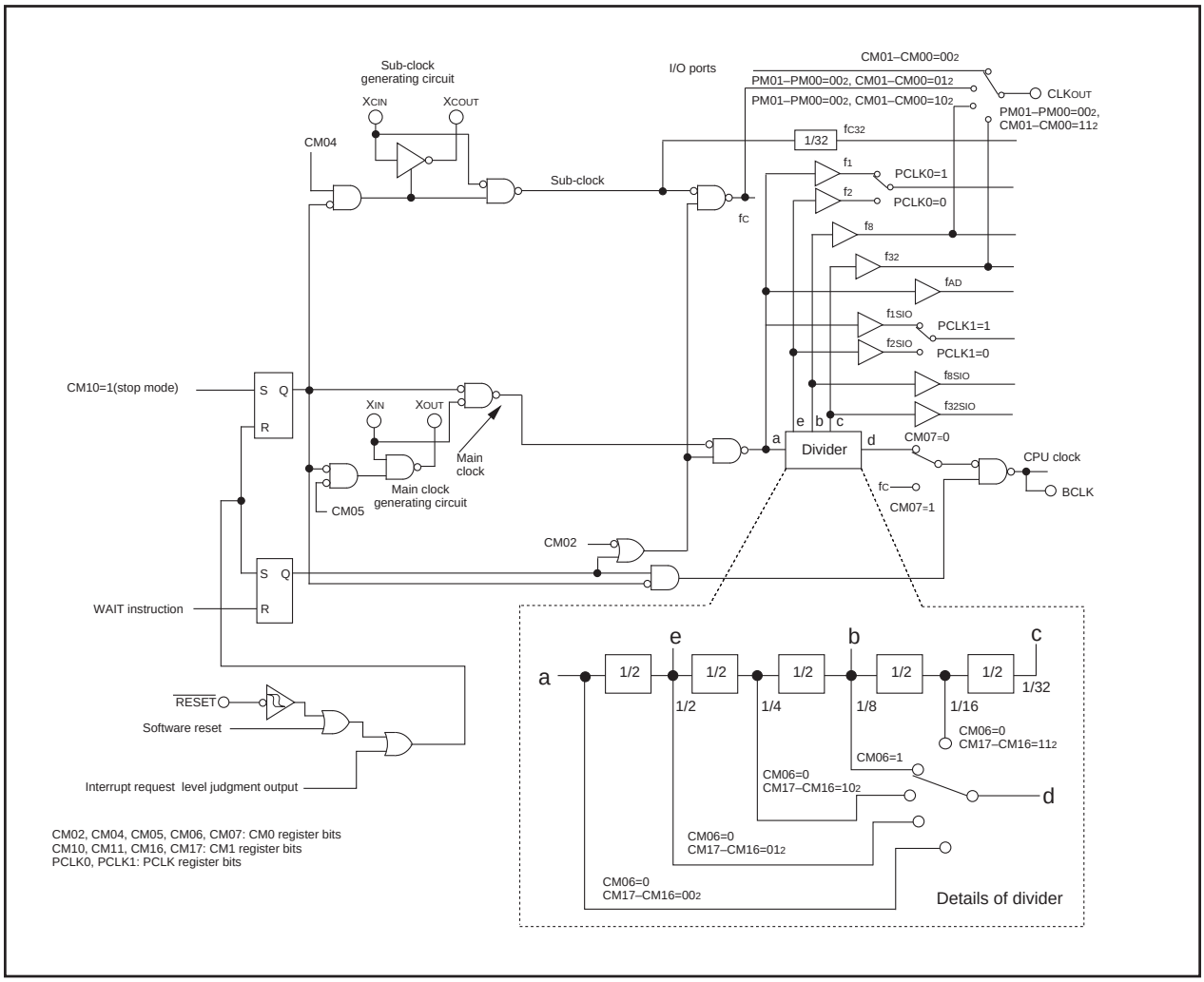
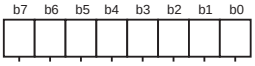


Figure 5.1. Clock Generation Circuit

System clock control register 0 (Note 1)



Symbol	Address	After reset
CM0	0006_{16}	01001000_2

Bit symbol	Bit name	Function	RW
CM00	Clock output function select bit (Valid only in single-chip mode)	b1 b0 0 0 : I/O port $P5_7$ 0 1 : f_C output 1 0 : f_8 output 1 1 : f_{32} output	RW
CM01			RW
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (Note 8)	RW
CM03	XCIN-XCOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	RW
CM04	Port XC select bit (Note 2)	0 : I/O port $P8_6$, $P8_7$ 1 : XCIN-XCOUT generation function(Note 9)	RW
CM05	Main clock stop bit (Notes 3, 11, 12)	0 : On 1 : Off (Note 4, Note5)	RW
CM06	Main clock division select bit 0 (Notes 7, 11)	0 : CM16 and CM17 valid 1 : Division by 8 mode	RW
CM07	System clock select bit (Notes 6, 10)	0 : Main clock 1 : Sub-clock	RW

Note 1: Write to this register after setting the PRC0 bit of PRCR register to “1” (write enable).
Note 2: The CM03 bit is set to “1” (high) when the CM04 bit is set to “0” (I/O port) or the microcomputer goes to a stop mode.
Note 3: This bit is provided to stop the main clock when the low power dissipation mode is selected.
This bit cannot be used for detection as to whether the main clock stopped or not. To stop the main clock, the following setting is required:
(1) Set the CM07 bit to “1” (Sub-clock select) with the sub-clock stably oscillating.
(2) Set the CM05 bit to “1” (Stop).
Note 4: During external clock input, only the clock oscillation buffer is turned off and clock input is accepted if the sub clock is not chosen as a CPU clock.
Note 5: When CM05 bit is set to “1, the XOUT pin goes “H”. Furthermore, because the internal feedback resistor remains connected, the XIN pin is pulled “H” to the same level as XOUT via the feedback resistor.
Note 6: After setting the CM04 bit to “1” (XCIN-XCOUT oscillator function), wait until the sub-clock oscillates stably before switching the CM07 bit from “0” to “1” (sub-clock).
Note 7: When entering stop mode from high or middle speed mode, the CM06 bit is set to “1” (divide-by-8 mode).
Note 8: The f_{C32} clock does not stop. During low speed or low power dissipation mode, do not set this bit to “1” (peripheral clock turned off when in wait mode).
Note 9: To use a sub-clock, set this bit to “1”. Also make sure ports $P8_6$ and $P8_7$ are directed for input, with no pull-ups.
Note 10: To use the main clock as the clock source for the CPU clock, follow the procedure below.
(1) Set the CM05 bit to “0” (oscillate).
(2) Wait until td(M-L) elapses or the main clock oscillation stabilizes, whichever is longer.
(3) Set the CM07 bit to “0”.
Note 11: When the CM05 bit = 1 (main clock turned off), the CM06 bit is fixed to “1” (divide-by-8 mode) and the CM15 bit is fixed to “1” (drive capability High). Please do not change bits CM06, CM17 to CM16 in low power consumption mode.

Figure 5.2. CM0 Register

System clock control register 1 (Note 1)

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

Symbol	Address	After reset
CM1	0007_{16}	00100000_2

Bit symbol	Bit	Function	RW
CM10	All clock stop control bit (Notes 4, 5)	0 : Clock on 1 : All clocks off (stop mode)	RW
(b4-b1)	Reserved bit	Must set to “0”	RW
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	RW
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	RW
CM17			RW

Note 1: Write to this register after setting the PRC0 bit of PRCR register to “1” (write enable).
Note 2: When entering stop mode from high or middle speed mode, or when the CM05 bit is set to “1” (main clock turned off) in low speed mode, the CM15 bit is set to “1” (drive capability high).
Note 3: Effective when the CM06 bit is “0” (CM16 and CM17 bits enable).
Note 4: If the CM10 bit is “1” (stop mode), XOUT goes “H” and the internal feedback resistor is disconnected. The XCIN and XCOUT pins are placed in the high-impedance state.
Note 5: Effective when CM07 bit is “0” .

Figure 5.3. CM1 Register

System clock control register 2 (Note 1)

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

Symbol	Address	After reset
CM2	$000C_{16}$	$0X000000_2$

Bit symbol	Bit name	Function	RW
(b2-b0)	Reserved bit	Must set to “1”	RW
CM23	XIN moniter flag	0: Main clock on 1: Main clock turned off	RO
(b5-b4)	Reserved bit	Must set to “1”	RW
(b6)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
(b7)	Reserved bit	Must set to “1”	RW

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

Figure 5.4. CM2 Register

Peripheral clock select register (Note 1)

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

Symbol: PCLKR
Address: $025E_{16}$
When reset: 00000011_2

Bit symbol	Bit name	Function	RW
PCLK0	Timers A, B clock select bit (Clock source for the timers A, B, and the dead timer)	0 : f_2 1 : f_1	RW
PCLK1	SI/O clock select bit (Clock source for UART0 to UART2)	0 : f_{2SIO} 1 : f_{1SIO}	RW
—— (b7-b2)	Reserved bit	Must set to “0”	RW

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

Reserved register (Note 1)

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

Symbol: RSVREG001E
Address: $001E_{16}$
When reset: $XXX00000_2$

Bit symbol	Bit name	Function	RW
—— (b4-b3)	Reserved bit	Must set to “0”	RW
—— (b7-b5)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—

Note 1: Write to this register after setting the PRC1 bit of PRCR register to “1” (write enable).

Figure 5.5. PCLKR Register and PM2 Register

Reserved register

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

Symbol	Address	After reset
RSVREG001C	$001C_{16}$	$0001X010_2$

Bit symbol	Bit name	Function	RW
(b0)	Reserved bit	Must set to “0”	RW
(b1)	Reserved bit	Must set to “1”	RW
(b2)	Reserved bit	Must set to “0”	RW
(b3)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
(b4)	Reserved bit	Must set to “1”	RW
(b7-b5)	Reserved bits	Must set to “0”	RW

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”.

Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XIN and XOUT following the instruction.

Figure 5.7. Examples of Main Clock Connection Circuit

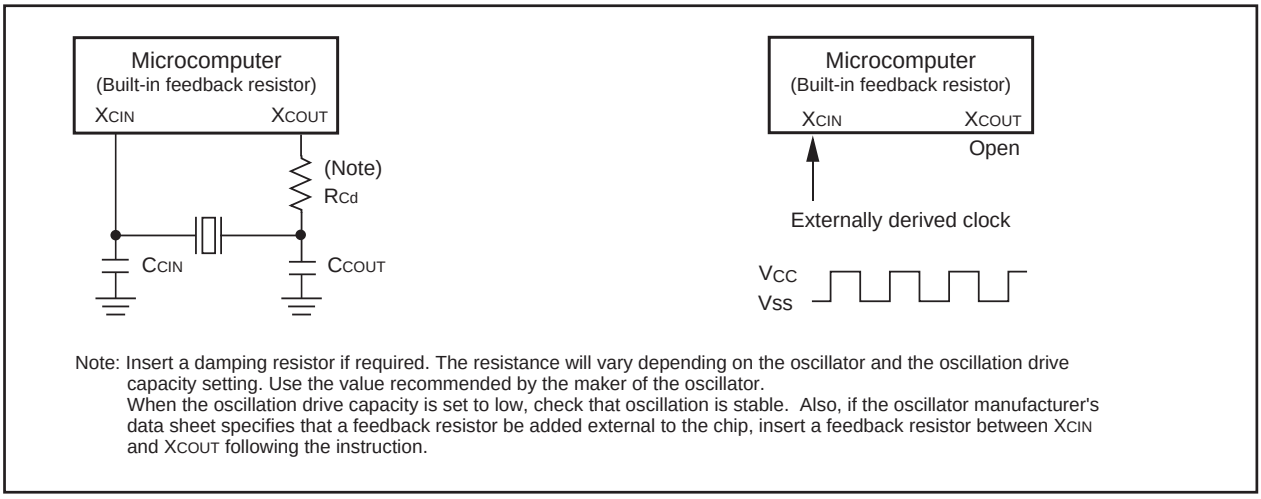
(2) Sub Clock

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



Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XCIN and XCOUT following the instruction.

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 0205_{16}) and bit 1 of the extended register 1C (address $02DC_{16}$).

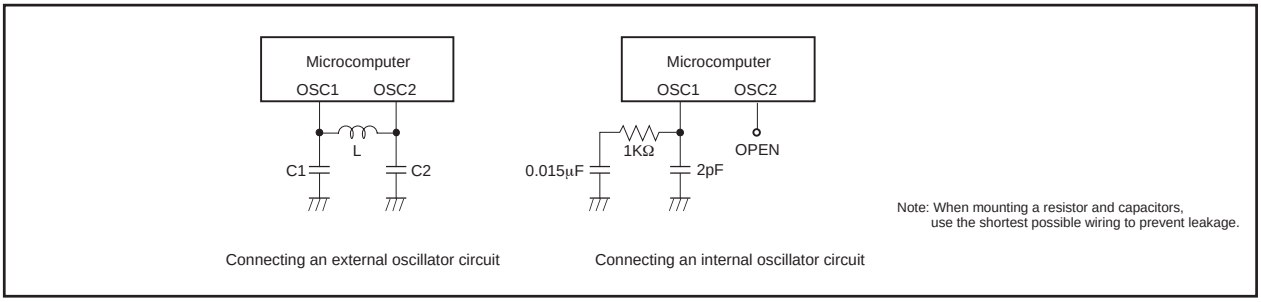


Figure 5.9. OSD clock connection example

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(f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}, f_{AD}, f_{C32})

These are operating clocks for the peripheral functions.

Of these, fi (i = 1, 2, 8, 32) and f_{iSIO} are derived from the main clock. The clock fi is used for timers A and B, and f_{iSIO} is used for serial I/O. The f_8 and f_{32} clocks can be output from the CLK_{OUT} pin.

The f_{AD} 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, f_{iSIO} and f_{AD} clocks are turned off.

The f_{C32} clock is produced from the sub clock, and is used for timers A and B. This clock can be used when the sub clock is on.

Clock Output Function

During single-chip mode, the f_8, f_{32} or f_C clock can be output from the CLK_{OUT} 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, f_{C32} 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, f_{C32} can be used as the count source for timers A and B.

- **Low-speed Mode**

  The sub clock provides the CPU clock.

  The f_{C32} 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 f_{C32} 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		00_2	0	0	0	—
Medium-speed mode	divided by 2	01_2	0	0	0	—
	divided by 4	10_2	0	0	0	—
	divided by 8	—	0	1	0	—
	divided by 16	11_2	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 f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{8SIO}, f_{32SIO} and f_{AD} clocks are turned off when in wait mode, with the power consumption reduced that much. However, f_{C32} 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 “000_2” (interrupts disabled) before executing the WAIT instruction.

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

Table 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 $\overline{CS3}$, $\overline{BHE}$		Retains status before wait mode	
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		"H"	
$\overline{HLDA}$,BCLK		"H"	
ALE		"L"	
I/O ports		Retains status before wait mode	Retains status before wait mode
CLK_{OUT}	When f_C selected		Does not stop
	When f_8, f_{32} 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 f_{C32}
$\overline{INT}$ 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 "000_2" (interrupt disable).
2. Set the I flag to "1".
3. Enable the peripheral function whose interrupt is to be used to exit wait mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt 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 “000_2” (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 “000_2”.
2. Set the I flag to “1”.
3. Enable the peripheral function whose interrupt is to be used to exit stop mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt 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:

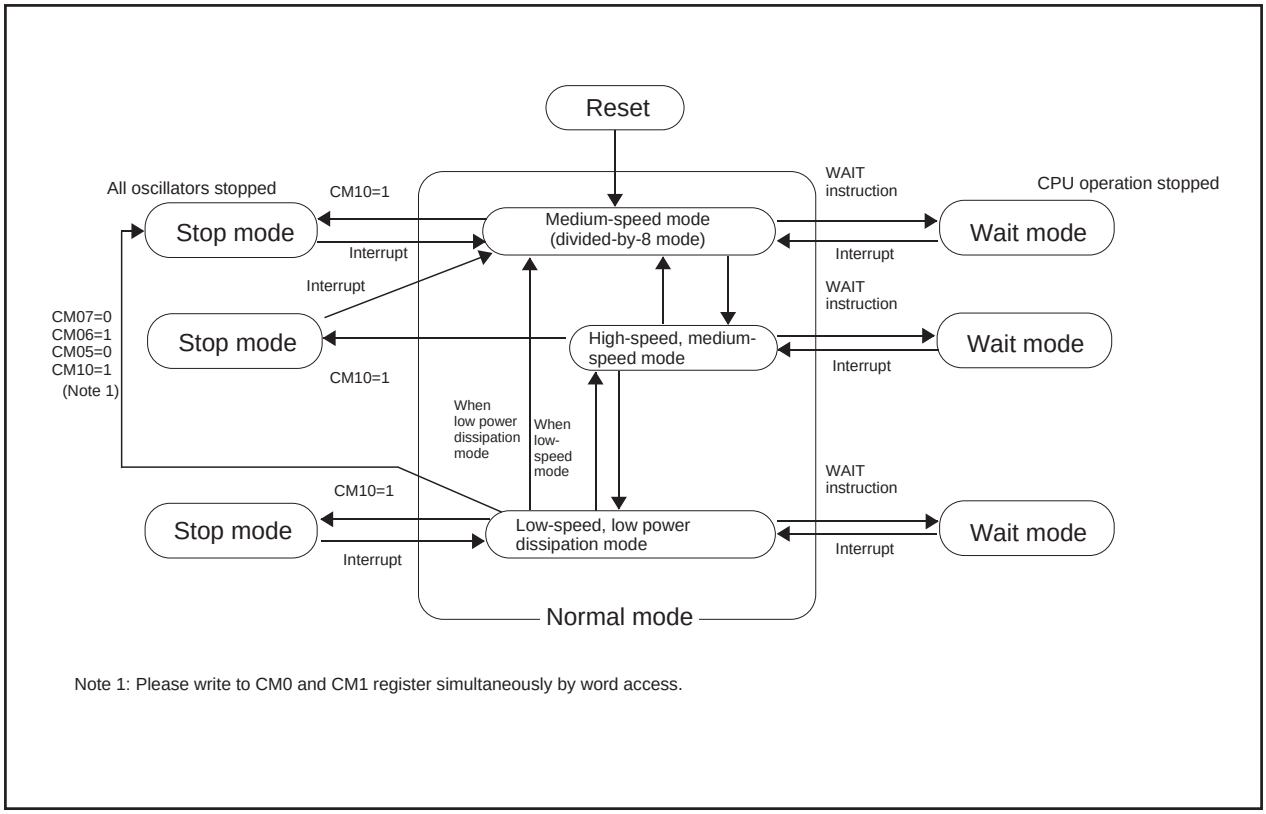
If the CPU clock before entering stop mode was derived from the sub clock: sub clock

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

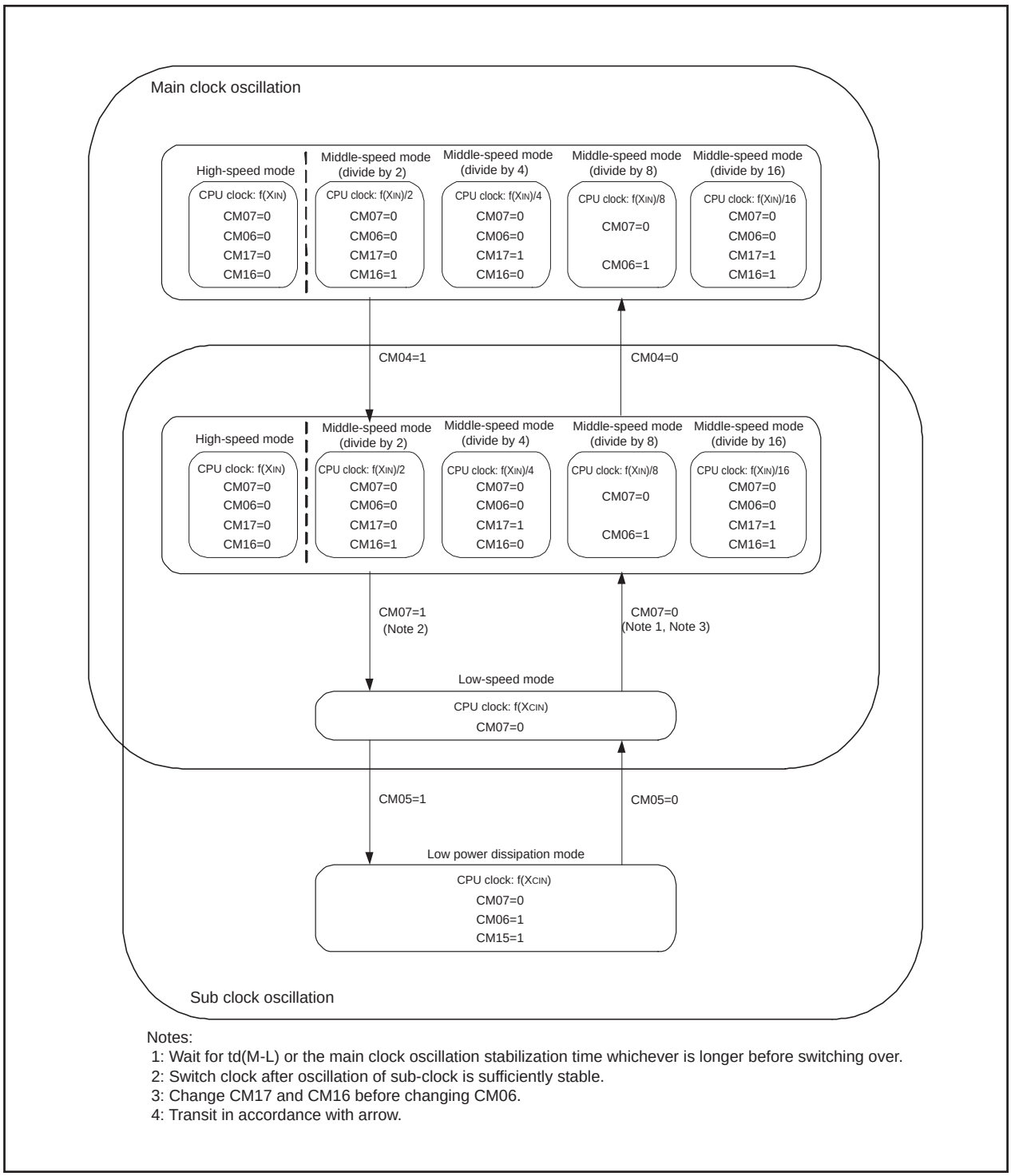
Pin		Memory expansion mode Microprocessor mode	Single-chip mode
A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$		Retains status before stop mode	
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		"H"	
$\overline{HLDA}$, BCLK		"H"	
ALE		undefined	
I/O ports		Retains status before stop mode	Retains status before stop mode
CLK_{OUT}	When fc selected		"H"
	When f_8, f_{32} 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.



Note 1: Please write to CM0 and CM1 register simultaneously by word access.

Figure 5.10. State Transition to Stop Mode and Wait Mode



Notes:
1: Wait for td(M-L) or the main clock oscillation stabilization time whichever is longer before switching over.
2: Switch clock after oscillation of sub-clock is sufficiently stable.
3: Change CM17 and CM16 before changing CM06.
4: Transit in accordance with arrow.

Figure 5.11. State Transition in Normal Mode

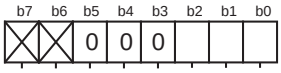
Protection

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 $001C_{16}$.
- Registers protected by PRC1 bit: PM0 and PM1 registers, reserved registers address $001E_{16}$, $039E_{16}$, 0348_{16}, and 0349_{16}
- Registers protected by PRC2 bit: PD9 register, reserved registers address 0362_{16} and 0366_{16}

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.

Protect register



Symbol: PRCR — Address: $000A_{16}$ — After reset: $XX000000_2$

Bit symbol	Bit name	Function	RW
PRC0	Protect bit 0	Enable write to CM0, CM1, CM2, PCLKR and register $001C_{16}$ 0 : Write protected 1 : Write enabled	RW
PRC1	Protect bit 1	Enable write to PM0, PM1 registers $001E_{16}$, $039E_{16}$, 0348_{16} and 0349_{16} 0 : Write protected 1 : Write enabled	RW
PRC2	Protect bit 2	Enable write to PD9, registers 0362_{16} and 0366_{16} 0 : Write protected 1 : Write enabled (Note)	RW
(b5-b3)	Reserved bits	Must set to “0”	RW
(b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—

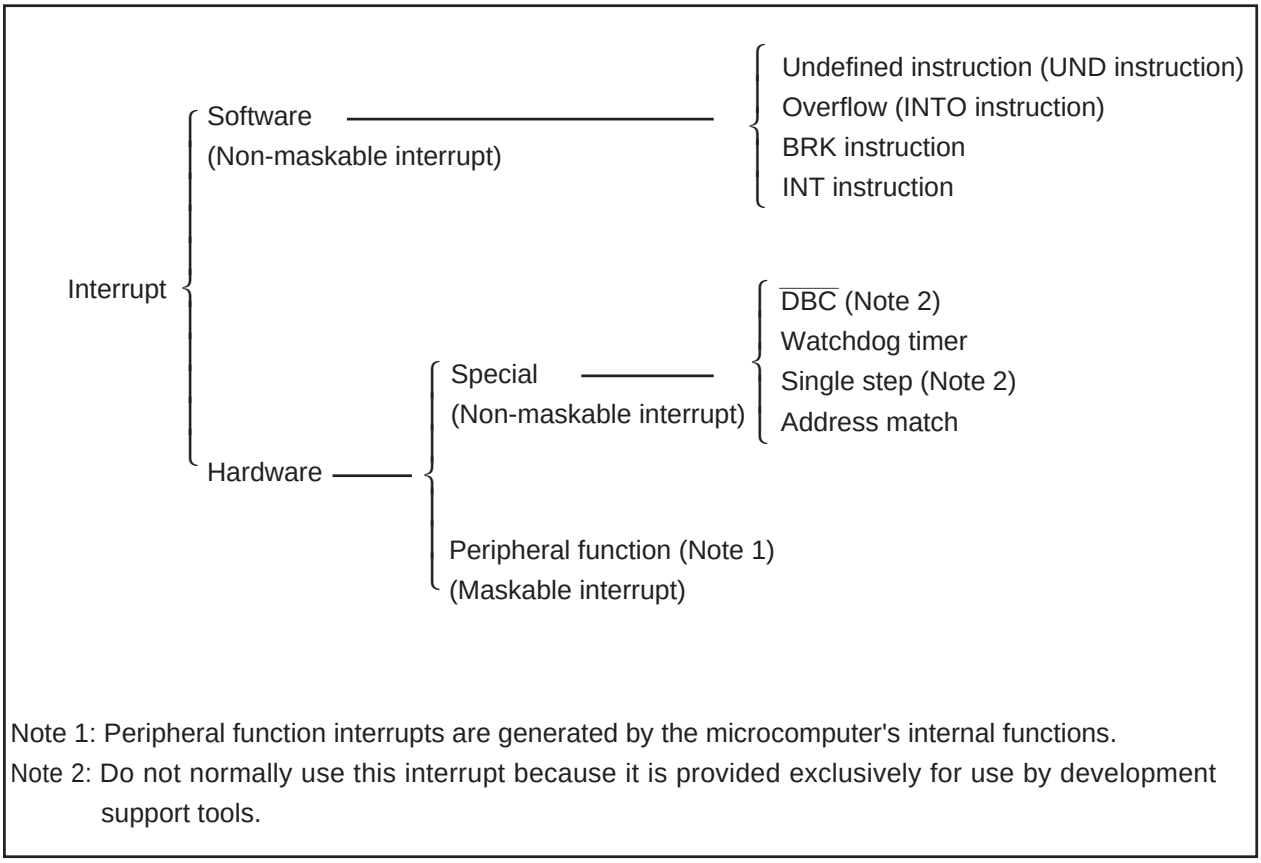
Note: The PRC2 bit is set to “0” by writing to any address after setting it to “1”. Other bits are not set to “0” by writing to any address, and must therefore be set in a program.

Figure 6.1. PRCR Register

Interrupts

Type of Interrupts

Figure 7.1 shows types of interrupts.



Note 1: Peripheral function interrupts are generated by the microcomputer's internal functions.
Note 2: Do not normally use this interrupt because it is provided exclusively for use by development support tools.

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.

- **$\overline{\text{DBC}}$ 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.

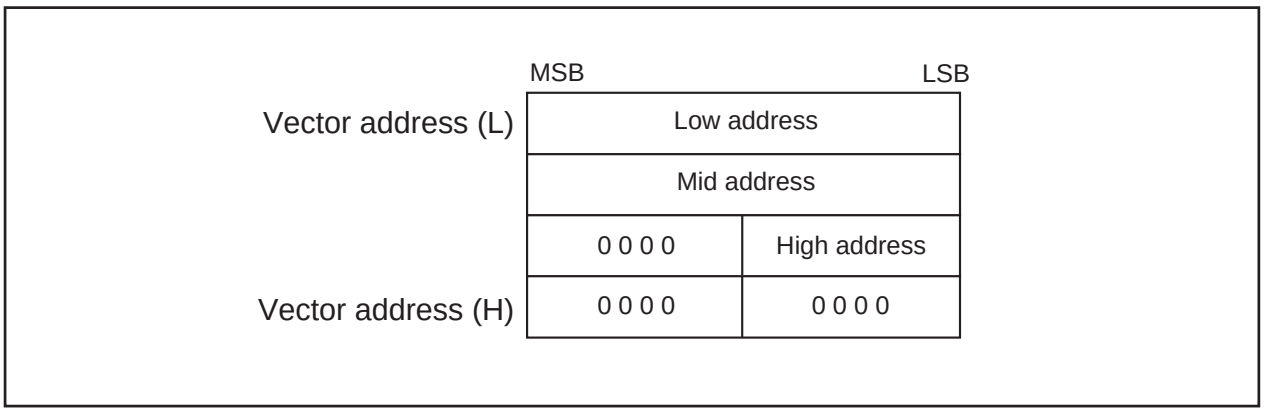


Figure 7.2. Interrupt Vector

- **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	FFFDC16 to FFFDF16	Interrupt on UND instruction	M16C/60, M16C/20 serise software maual
Overflow	FFFE016 to FFFE316	Interrupt on INTO instruction	
BRK instruction	FFFE416 to FFFE716	If the contents of address FFFE716 is FF16, program execution starts from the address shown by the vector in the relocatable vector table.	
Address match	FFFE816 to FFFEB16		Address match interrupt
Single step (Note)	FFFEC16 to FFFEF16		
Watchdog timer	FFFF016 to FFFF316		Watchdog timer
$\overline{\text{DBC}}$ (Note)	FFFF416 to FFFF716		
Reset	FFFFC16 to FFFFF16		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 reloacatable 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^2C-bus0	+40 to +43 (0028_{16} to $002B_{16}$)	10	Serial I/O/I^2C-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^2C-bus1NACK	+56 to +59 (0038_{16} to $003B_{16}$)	14	A/D convertor/I^2C-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^2C-bus0	+84 to +87 (0054_{16} to 0057_{16})	21	Timer/I^2C-bus
Timer A1/I^2C-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^2C-bus0NACK	+100 to +103 (0064_{16} to 0067_{16})	25	
Timer B0/I^2C-bus1NACK	+104 to +107 (0068_{16} to $006B_{16}$)	26	
Timer B1/I^2C-bus2NACK	+108 to +111 ($006C_{16}$ to $006F_{16}$)	27	
Timer B2/I^2C-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

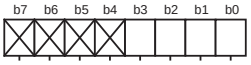
The following describes how to enable/disable the maskable interrupts, and how to set the priority in which order they are accepted. What is explained here does not apply to nonmaskable interrupts.

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.

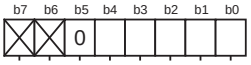
Interrupt control register (Note 2)

Symbol	Address	After reset
TB5IC	0045_{16}	$XXXXX000_2$
TB4IC	0046_{16}	$XXXXX000_2$
TB3IC	0047_{16}	$XXXXX000_2$
BCNIC	$004A_{16}$	$XXXXX000_2$
DM0IC, DM1IC	$004B_{16}$, $004C_{16}$	$XXXXX000_2$
KUPIC	$004D_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
S0TIC to S2TIC	0051_{16}, 0053_{16}, $004F_{16}$	$XXXXX000_2$
S0RIC to S2RIC	0052_{16}, 0054_{16}, 0050_{16}	$XXXXX000_2$
TA0IC to TA4IC	0055_{16} to 0059_{16}	$XXXXX000_2$
TB0IC to TB2IC	$005A_{16}$ to $005C_{16}$	$XXXXX000_2$



Bit symbol	Bit	Functio	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	RW (Note 1)
— (b7-b4)	No functions are assigned. When writing to these bits, write “0”. The values in these bits when read are indeterminate.		—

Note 1: IR bit write “0” only (Do not write “1”).
Note 2: To rewrite the interrupt control registers, do so at a point that does not generate the interrupt request for that register. For details, see the “Notes of interruption” of the Usage Notes Reference Book.



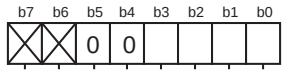
Symbol	Address	After reset
INT0IC to INT2IC	$005D_{16}$ to $005F_{16}$	$XX00X000_2$

Bit	Bit	Functio	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	RW (Note 1)
POL	Polarity select bit	0 : Selects falling edge (Notes 3, 5) 1 : Selects rising edge	RW
— (b5)	Reserved bit	Must always be set to “0”	RW
— (b7-b6)	No functions are assigned. When writing to these bits, write “0”. The values in these bits when read are indeterminate.		—

Note 1: This bit can only be reset by writing “0” (Do not write “1”).
Note 2: To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. For details, see the “Notes of interruption” of the Usage Notes Reference Book.
Note 3: If the IFSR register’s IFSRi bit (i = 0 to 2) is “1” (both edges), set the INTiIC register’s POL bit to “0” (falling edge).

Figure 7.3. Interrupt Control Registers

Interrupt control register



Symbol	Address	After reset
INT3IC	0044_{16}	$XX00X000_2$
DSC1IC	0048_{16}	$XX00X000_2$
DSC2IC	0049_{16}	$XX00X000_2$

Bit	Bit	Functio	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	RW (Note 1)
—	Reserved bit	Must always be set to “0”	RW
—	Reserved bit	Must always be set to “0”	RW
— (b7-b6)	No functions are assigned. When writing to these bits, write “0”. The values in these bits when read are indeterminate.		—

Note 1: This bit can only be reset by writing “0” (Do not write “1”).
Note 2: To rewrite the interrupt control registers, do so at a point that does not generate the interrupt request for that register. For details, see the “precautions for interrupts” of the Usage Notes Reference Book.

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
000_2	Level 0 (interrupt disabled)	———
001_2	Level 1	Low ↓
010_2	Level 2	
011_2	Level 3	
100_2	Level 4	
101_2	Level 5	
110_2	Level 6	
111_2	Level 7	High

Table 7.4. Interrupt Priority Levels Enabled by IPL

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

Interrupt Sequence

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

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

The CPU behavior during the interrupt sequence is described below. Figure 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 00000_{16}. Then it clears the IR bit for the corresponding interrupt to “0” (interrupt not requested).

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

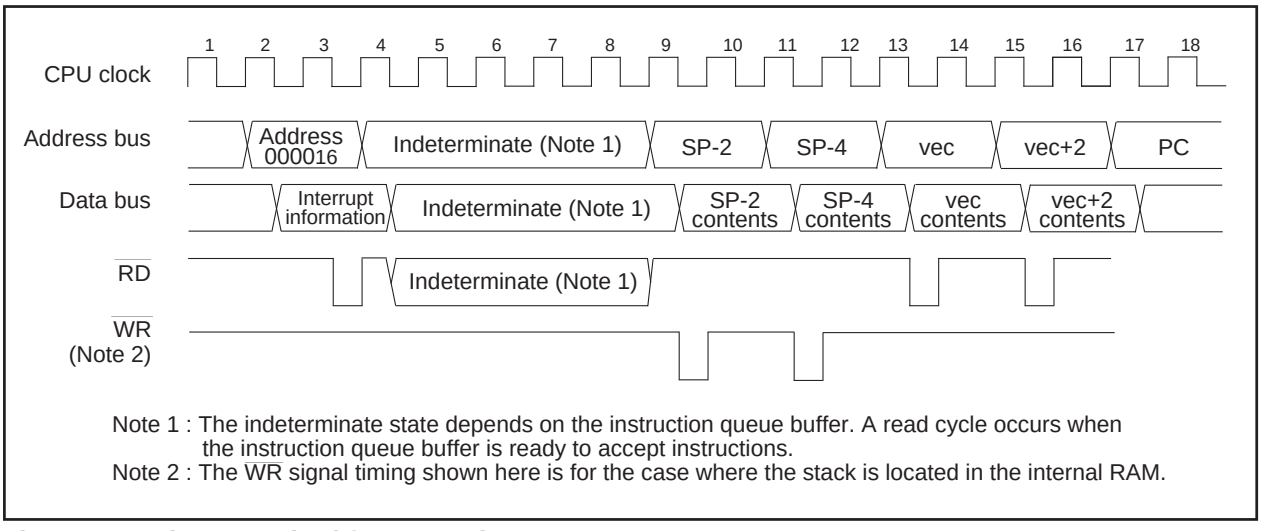
(5) The PC is saved to the stack.

(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.

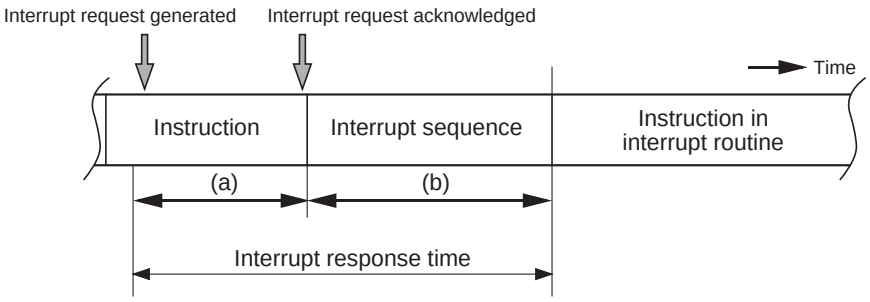


Note 1 : The indeterminate state depends on the instruction queue buffer. A read cycle occurs when the instruction queue buffer is ready to accept instructions.
Note 2 : The WR signal timing shown here is for the case where the stack is located in the internal RAM.

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).



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

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

Interrupt vector address	SP value	16-Bit bus, without wait	8-Bit bus, without wait
Even	Even	18 cycles	20 cycles
Even	Odd	19 cycles	20 cycles
Odd	Even	19 cycles	20 cycles
Odd	Odd	20 cycles	20 cycles

Figure 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, $\overline{\text{DBC}}$, 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.

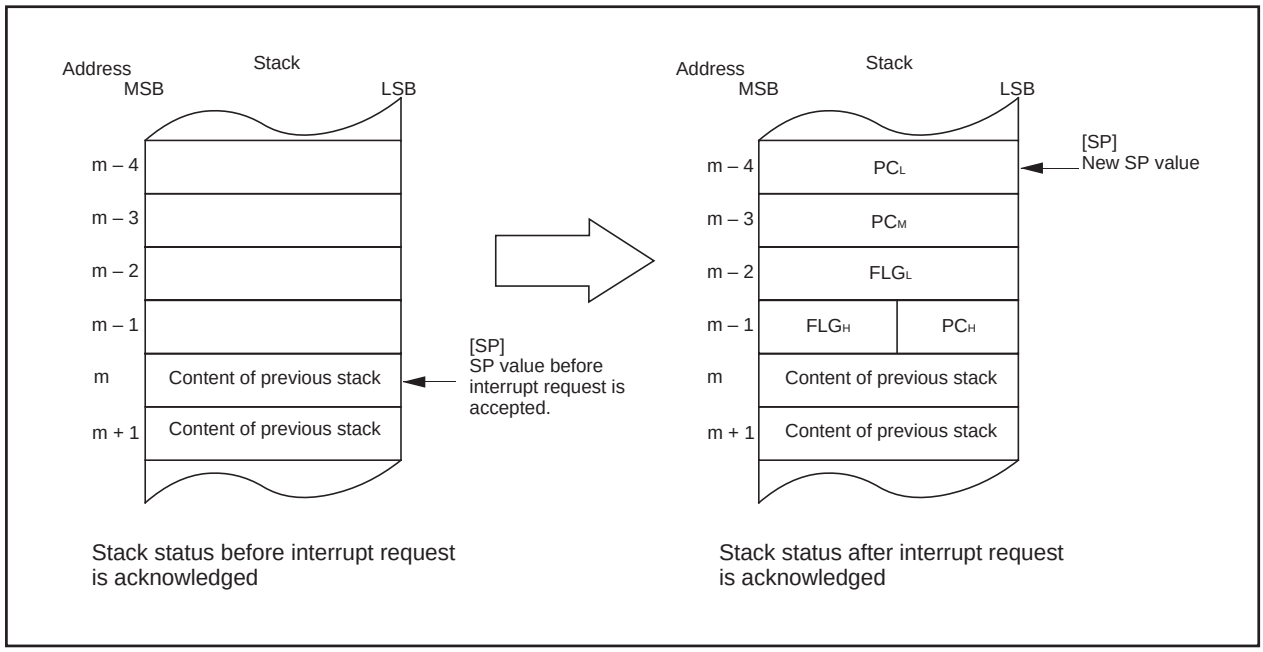


Figure 7.7. Stack Status Before and After Acceptance of Interrupt Request

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.

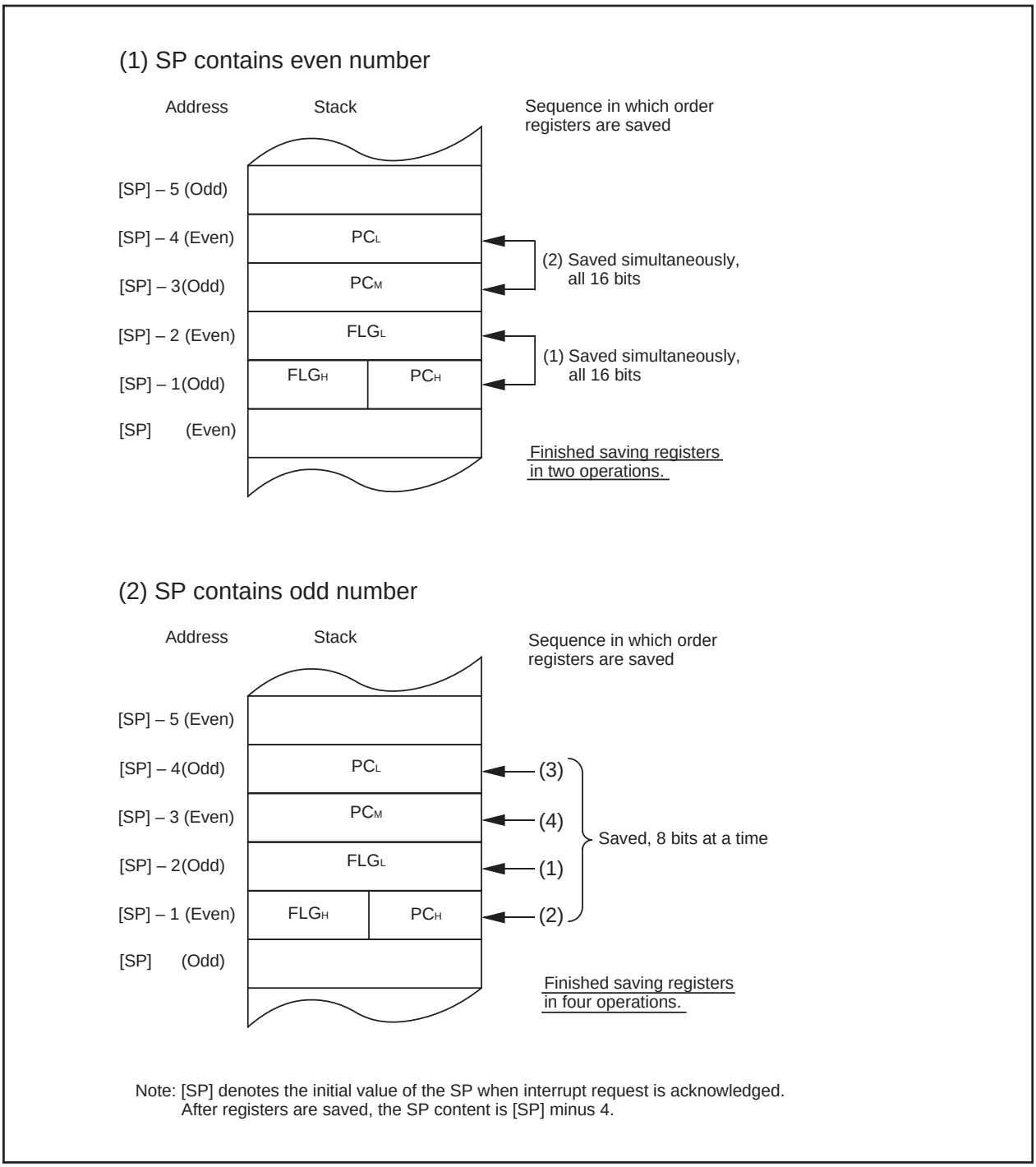


Figure 7.8. Operation of Saving Register

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.

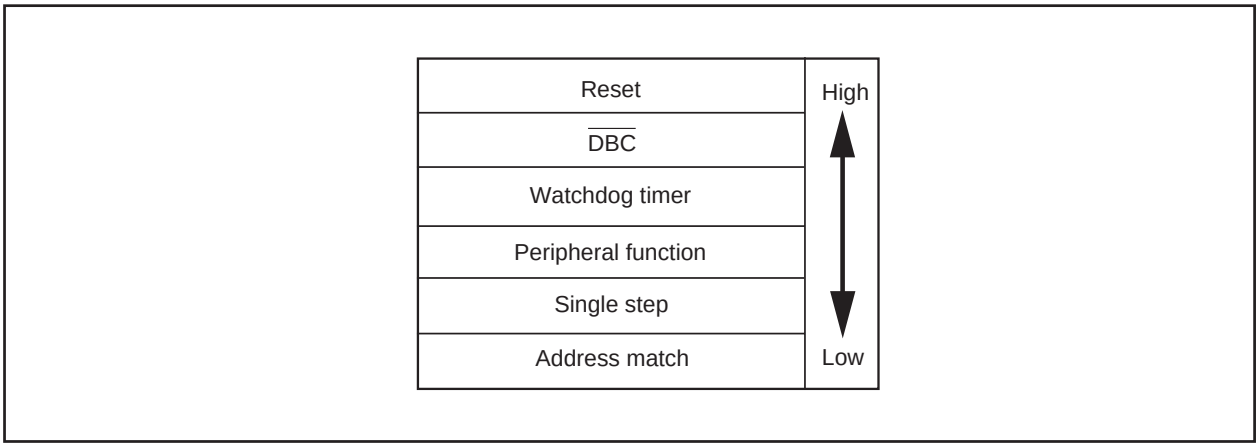


Figure 7.9. Hardware Interrupt Priority

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.

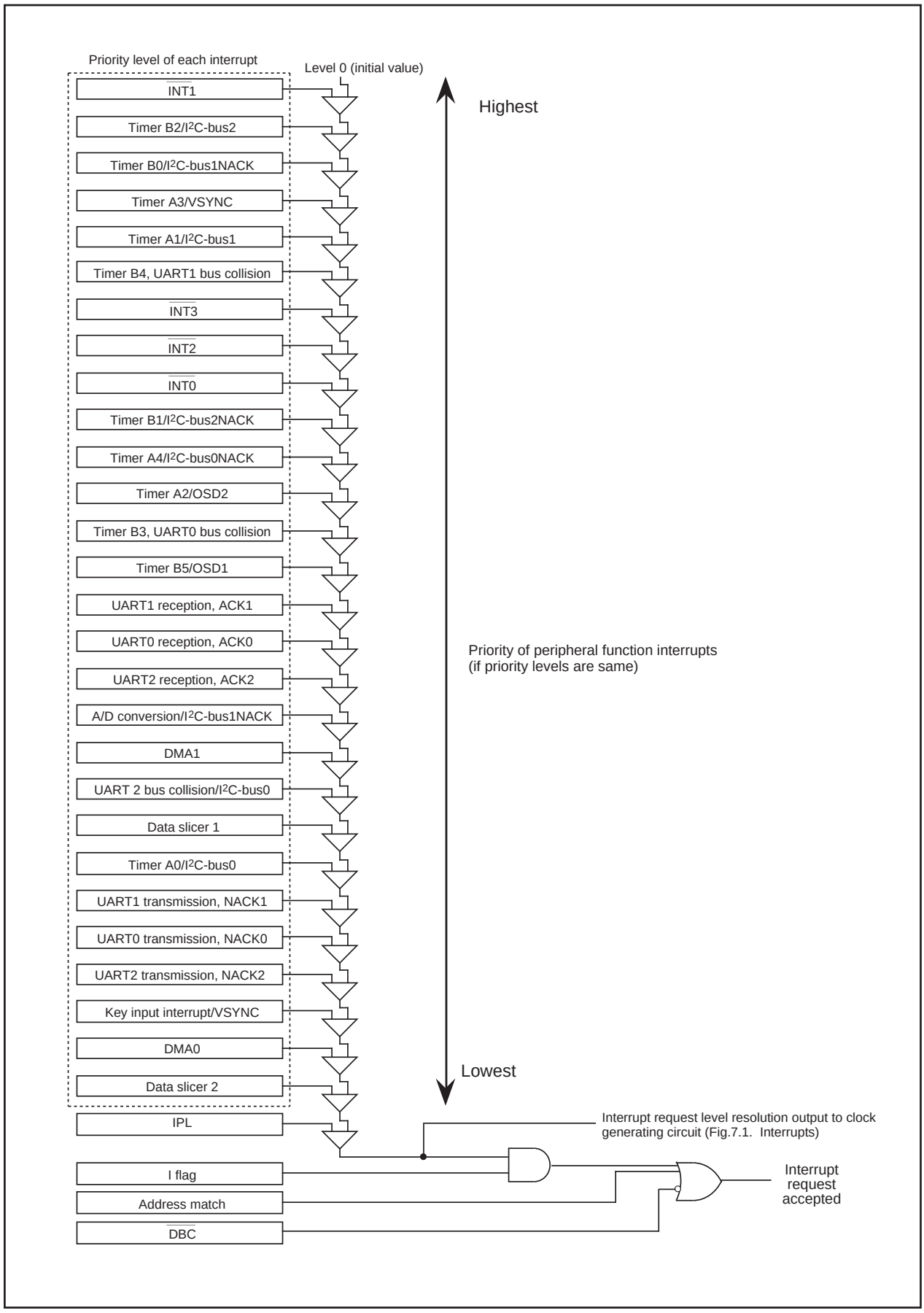


Figure 7.10. Interrupts Priority Select Circuit

INT Interrupt

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

Figure 7.11 shows the IFSR and IFSR2A registers.

Interrupt request cause select register

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

Symbol: IFSR | Address: $035F_{16}$ | After reset: 00_{16}

Bit symbol	Bit name	Function	RW
IFSR0	INT0 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
IFSR1	INT1 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
IFSR2	INT2 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
IFSR3	INT3 interrupt polarity switching bit	0 : One edge 1 : Both edges (Note 1)	RW
—	Reserved bit	Must always be set to "0"	RW

Note 1: When setting this bit to "1" (= both edges), make sure the INT0IC to INT2IC register's POL bit is set to "0" (= falling edge).

Interrupt request cause select register 2

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: IFSR2A | Address: $035E_{16}$ | After reset: $00XXXXXX_2$

Bit symbol	Bit name	Function	RW
— (b5-b0)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—
IFSR26	Interrupt request cause select bit (Note 1)	0 : Timer B3 1 : UART0 bus collision detection	RW
IFSR27	Interrupt request cause select bit (Note 2)	0 : Timer B4 1 : UART1 bus collision detection	RW

Note 1: Timer B3 and UART0 bus collision detection share the vector and interrupt control register. When using the timer B3 interrupt, clear the IFSR26 bit to "0" (timer B3). When using UART0 bus collision detection, set the IFSR26 bit to "1".

Note 2: Timer B4 and UART1 bus collision detection share the vector and interrupt control register. When using the timer B4 interrupt, clear the IFSR27 bit to "0" (timer B4). When using UART1 bus collision detection, set the IFSR27 bit to "1".

Figure 7.11. IFSR Register and IFSR2A Register

Key Input Interrupt

Of P10$_4$ to P10$_7$, a key input interrupt is generated when input on any of the P10$_4$ to P10$_7$ pins which has had the PD10 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 P10$_4$ to P10$_7$ 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.

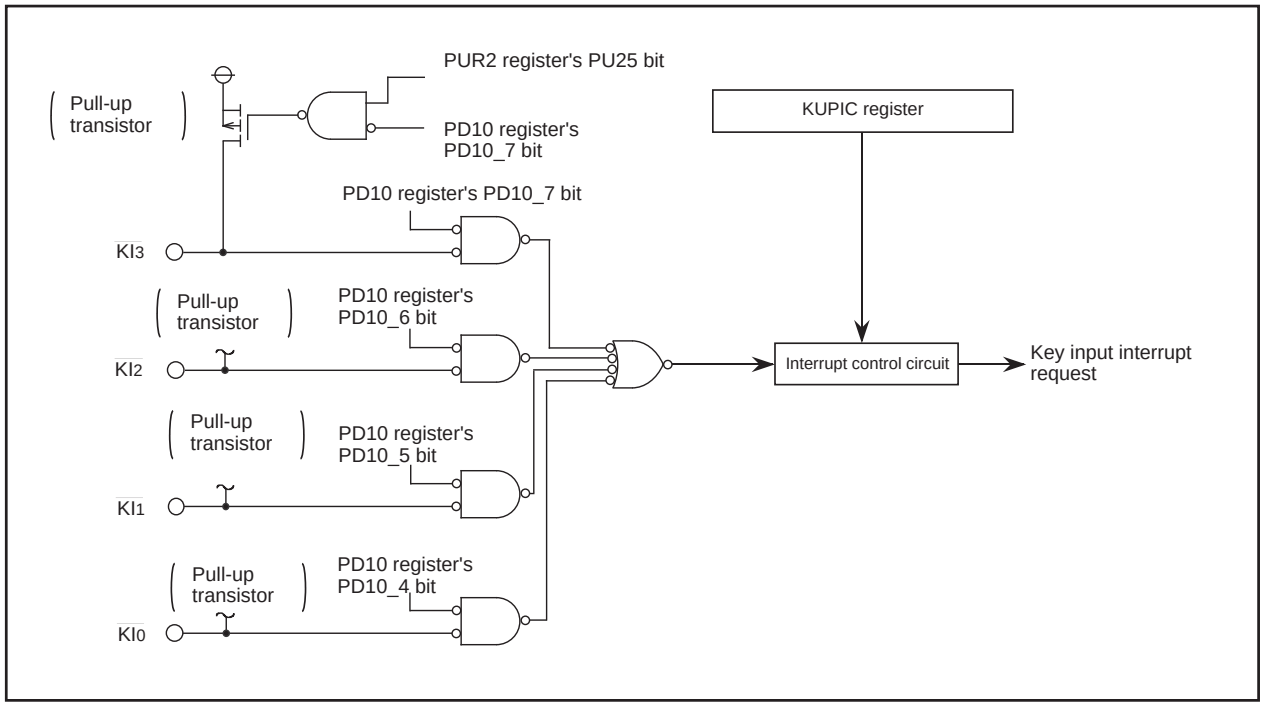


Figure 7.12. Key Input Interrupt

Address Match Interrupt

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

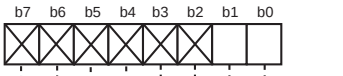
Instruction at the address indicated by the RMADi register	Value of the PC that is saved to the stack area
• 16-bit op-code instruction • Instruction shown below among 8-bit operation code instructions ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ.B:S #IMM8,dest STNZ.B:S #IMM8,dest STZX.B:S #IMM81,#IMM82,dest CMP.B:S #IMM8,dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM,dest (However, dest=A0 or A1)	The address indicated by the RMADi register +2
Instructions other than the above	The address indicated by the RMADi register +1

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

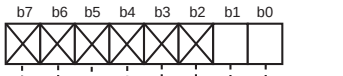
Address match interrupt enable register



Symbol	Address	After reset
AIER	0009_{16}	$XXXXXX00_2$

Bit symbol	Bit name	Function	RW
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
—— (b7-b2)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

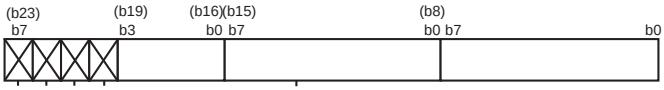
Address match interrupt enable register 2



Symbol	Address	After reset
AIER2	$01BB_{16}$	$XXXXXX00_2$

Bit symbol	Bit name	Function	RW
AIER20	Address match interrupt 2 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER21	Address match interrupt 3 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
—— (b7-b2)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

Address match interrupt register i (i = 0 to 3)



Symbol	Address	After reset
RMAD0	0012_{16} to 0010_{16}	$X0000000_{16}$
RMAD1	0016_{16} to 0014_{16}	$X0000000_{16}$
RMAD2	$01BA_{16}$ to $01B8_{16}$	$X0000000_{16}$
RMAD3	$01BE_{16}$ to $01BC_{16}$	$X0000000_{16}$

Function	Setting range	RW
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	RW
Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—

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

Notes of interruption

(1) Address 00000$_{16}$ read-out

● Please do not read address 00000$_{16}$ 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 00000$_{16}$.IR bit of received interruption is set to "0" at this time.If address 00000$_{16}$ 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 "0000$_{16}$." 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.

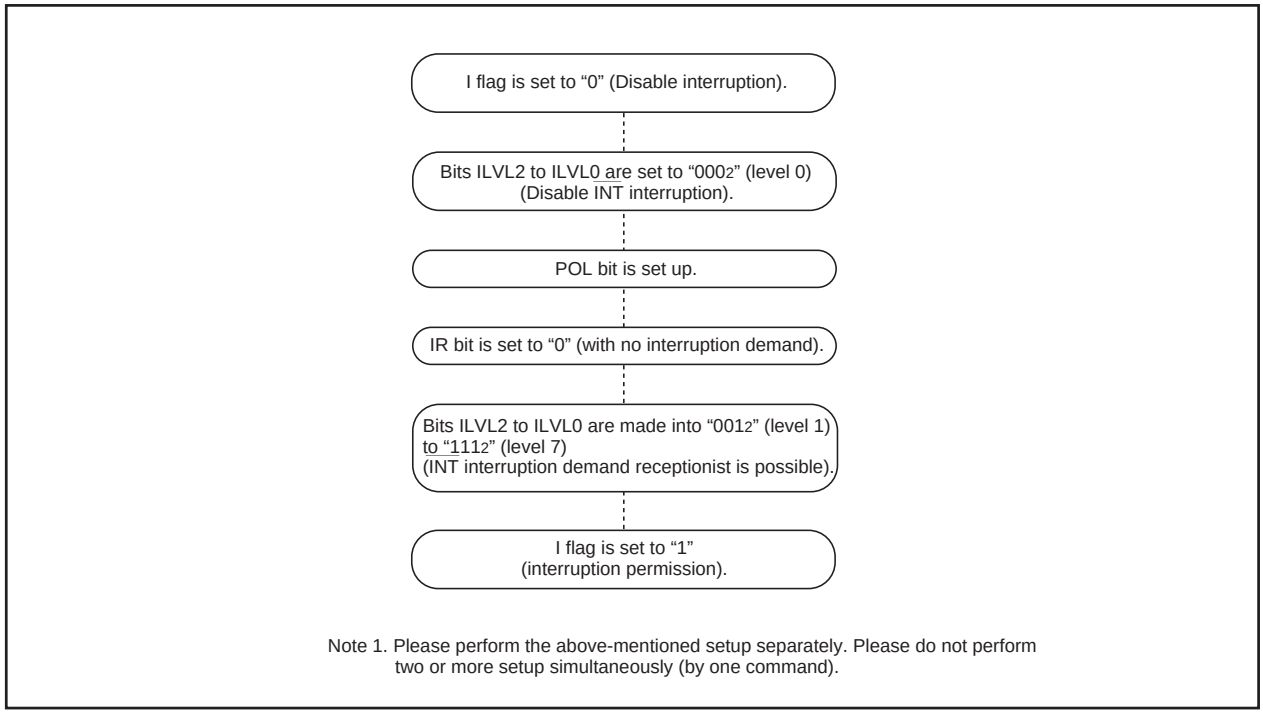


Figure 7.14. The example of change procedure of INT interruption generating factor

(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.

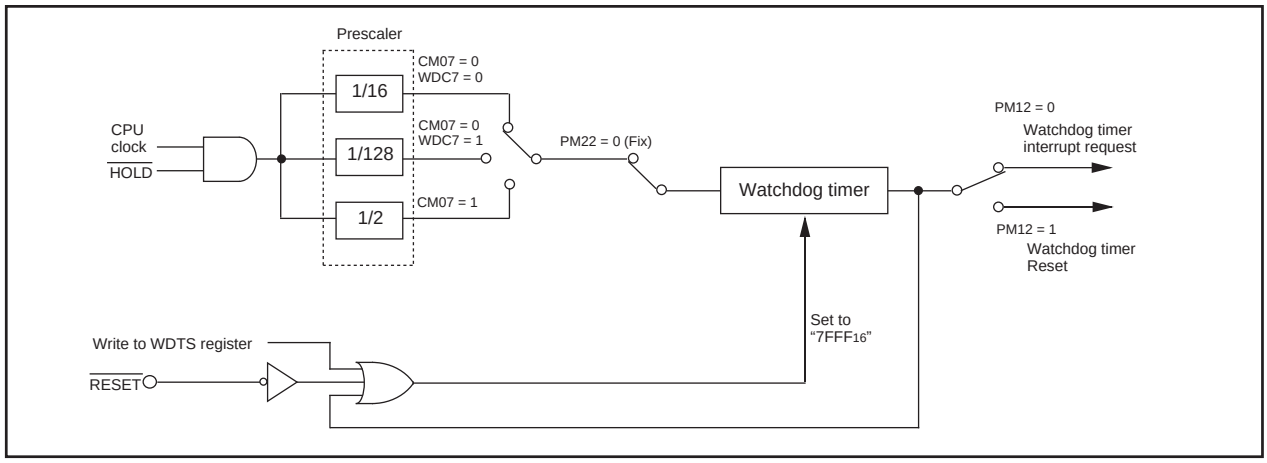


Figure 8.1. Watchdog Timer Block Diagram

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	After reset
WDC	$000F_{16}$	$00XXXXXX_2$

Bit symbol	Bit name	Function	RW
$\overline{(b4\text{-}b0)}$	High-order bit of watchdog timer		RO
$\overline{(b5, b6)}$	Reserved bit	Must set to "0"	RW
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	RW

Watchdog timer start register (Note)

b7	b0

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

Function	RW
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to "7FFF16" regardless of whatever value is written.	WO

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

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.

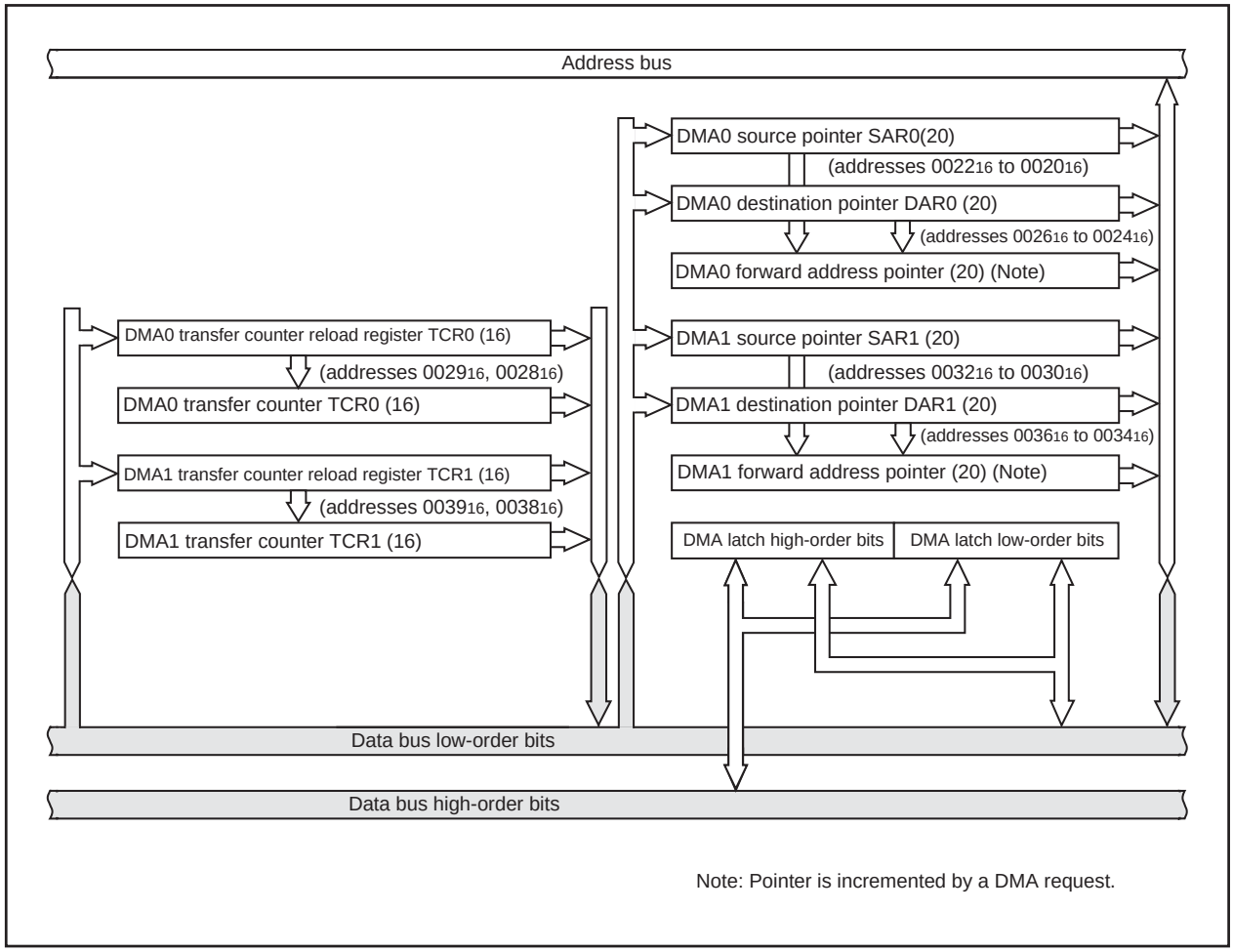


Figure 9.1. DMAC Block Diagram

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".

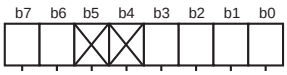
Table 9.1. DMAC Specifications

Item		Specification
No. of channels		2 (cycle steal method)
Transfer memory space		• 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^2C-bus interface 0, 1, 2 interrupt I^2C-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 DMAi transfer counter (i = 0–1) underflows after reaching the terminal count.
	•Repeat transfer	When the DMAi transfer counter underflows, it is reloaded with the value of the DMAi transfer counter reload register and a DMA transfer is con tinued with it.
DMA interrupt request generation timing		When the DMAi transfer counter underflowed
DMA startup		Data transfer is initiated each time a DMA request is generated when the DMAiCON register's DMAE bit = "1" (enabled).
DMA shutdown	•Single transfer	• When the DMAE bit is set to "0" (disabled)
	•Repeat transfer	• After the DMAi 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" (en abled), the forward address pointer is reloaded with the value of the SARi or the DARi pointer whichever is specified to be in the forward direction and the DMAi transfer counter is reloaded with the value of the DMAi transfer counter reload register.

Notes:

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

DMA0 request cause select register



Symbol	Address	After reset
DM0SL	03B8$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
DSEL0	DMA request cause select bit	Refer to note	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
— (b5-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—
DMS	DMA request cause expansion select bit	0: Basic cause of request 1: Extended cause of request	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to “1” when the DMS bit is “0” (basic cause) and the DSEL3 to DSEL0 bits are “0001$_2$” (software trigger). The value of this bit when read is “0” .	RW

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

DSEL3 to DSEL0	DMS=0(basic cause of request)	DMS=1(extended cause of request)
0 0 0 0$_2$	Falling edge of $\overline{\text{INT0}}$ pin	–
0 0 0 1$_2$	Software trigger	–
0 0 1 0$_2$	Timer A0/I^{2}C-bus0	–
0 0 1 1$_2$	Timer A1/I^{2}C-bus1	–
0 1 0 0$_2$	Timer A2/OSD2	–
0 1 0 1$_2$	Timer A3/VSYNC	–
0 1 1 0$_2$	Timer A4/I^{2}C-bus0NACK	Two edges of $\overline{\text{INT0}}$ pin
0 1 1 1$_2$	Timer B0/I^{2}C-bus1NACK	Timer B3
1 0 0 0$_2$	Timer B1/I^{2}C-bus2NACK	Timer B4
1 0 0 1$_2$	Timer B2/I^{2}C-bus2	Timer B5/OSD1
1 0 1 0$_2$	UART0 transmit –	
1 0 1 1$_2$	UART0 receive	–
1 1 0 0$_2$	UART2 transmit –	
1 1 0 1$_2$	UART2 receive	–
1 1 1 0$_2$	A/D conversion	–
1 1 1 1$_2$	UART1 transmit	–

Figure 9.2. DM0SL Register

DMA1 request cause select register

b7	b6	b5	b4	b3	b2	b1	b0
		X	X				

Symbol	Address	After reset
DM1SL	$03BA_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
DSEL0	DMA request cause select bit	Refer to note	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
— (b5-b4)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—
DMS	DMA request cause expansion select bit	0: Basic cause of request 1: Extended cause of request	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to “1” when the DMS bit is “0” (basic cause) and the DSEL3 to DSEL0 bits are “0001_2” (software trigger). The value of this bit when read is “0” .	RW

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

DSEL3 to DSEL0	DMS=0 (basic cause of request)	DMS=1 (extended cause of request)
$0\,0\,0\,0_2$	Falling edge of INT1 pin	–
$0\,0\,0\,1_2$	Software trigger	–
$0\,0\,1\,0_2$	Timer A0/I^2C-bus0	–
$0\,0\,1\,1_2$	Timer A1/I^2C-bus1	–
$0\,1\,0\,0_2$	Timer A2/OSD2	–
$0\,1\,0\,1_2$	Timer A3/VSYNC	–
$0\,1\,1\,0_2$	Timer A4/I^2C-bus0NACK	–
$0\,1\,1\,1_2$	Timer B0/I^2C-bus1NACK	Two edges of INT1 pin
$1\,0\,0\,0_2$	Timer B1/I^2C-bus2NACK	–
$1\,0\,0\,1_2$	Timer B2 /I^2C-bus2	–
$1\,0\,1\,0_2$	UART0 transmit	–
$1\,0\,1\,1_2$	UART0 receive/ACK0	–
$1\,1\,0\,0_2$	UART2 transmit	–
$1\,1\,0\,1_2$	UART2 receive/ACK2	–
$1\,1\,1\,0_2$	A/D conversion	–
$1\,1\,1\,1_2$	UART1 receive/ACK1	–

DMAi control register(i=0,1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X						

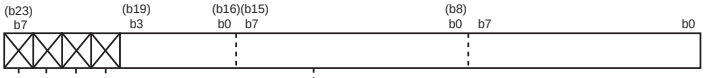
Symbol	Address	After reset
DM0CON	$002C_{16}$	$00000X00_2$
DM1CON	$003C_{16}$	$00000X00_2$

Bit symbol	Bit name	F unction	RW
DMBIT	Transfer unit bit select bit	0 : 16 bits 1 : 8 bits	RW
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	RW
DMAS	DMA request bit	0 : DMA not requested 1 : DMA requested	RW (Note 1)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	RW
DSD	Source address direction select bit (Note 2)	0 : Fixed 1 : Forward	RW
DAD	Destination address direction select bit (Note 2)	0 : Fixed 1 : Forward	RW
— (b7-b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—

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

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

DMAi source pointer (i = 0, 1) (Note)

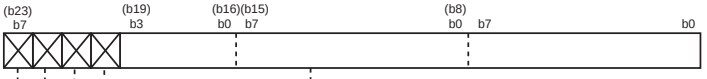


Symbol	Address	After reset
SAR0	0022_{16} to 0020_{16}	Indeterminate
SAR1	0032_{16} to 0030_{16}	Indeterminate

Function	Setting range	RW
Set the source address of transfer	00000_{16} to FFFFF$_{16}$	RW
Nothing is assigned. When write, set "0". When read, these contents are "0".		—

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

DMAi destination pointer (i = 0, 1)(Note)



Symbol	Address	After reset
DAR0	0026_{16} to 0024_{16}	Indeterminate
DAR1	0036_{16} to 0034_{16}	Indeterminate

Function	Setting range	RW
Set the destination address of transfer	00000_{16} to FFFFF$_{16}$	RW
Nothing is assigned. When write, set "0". When read, these contents are "0".		—

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

DMAi transfer counter (i = 0, 1)

Symbol	Address	After reset
TCR0	0029_{16}, 0028_{16}	Indeterminate
TCR1	0039_{16}, 0038_{16}	Indeterminate

Function	Setting range	RW
Set the transfer count minus 1. The written value is stored in the DMAi transfer counter reload register, and when the DMAE bit of DMiCON register is set to "1" (DMA enabled) or the DMAi transfer counter underflows when the DMASL bit of DMiCON register is "1" (repeat transfer), the value of the DMAi transfer counter reload register is transferred to the DMAi transfer counter. When read, the DMAi transfer counter is read.	0000_{16} to FFFF$_{16}$	RW

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

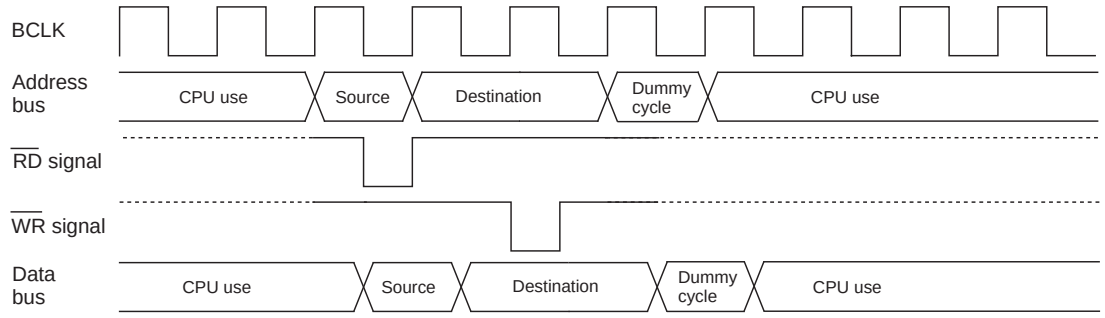
For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

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

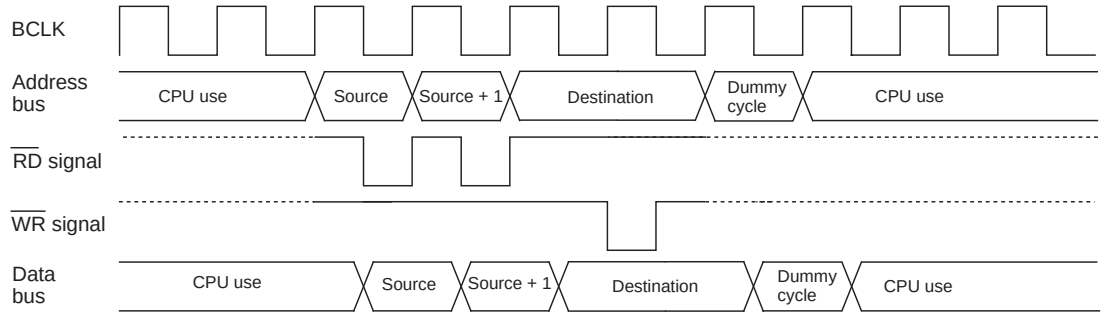
During memory extension and microprocessor modes, DMA transfers to and from an external area are affected by the $\overline{\text{RDY}}$ signal. Refer to “$\overline{\text{RDY}}$ signal”.

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.

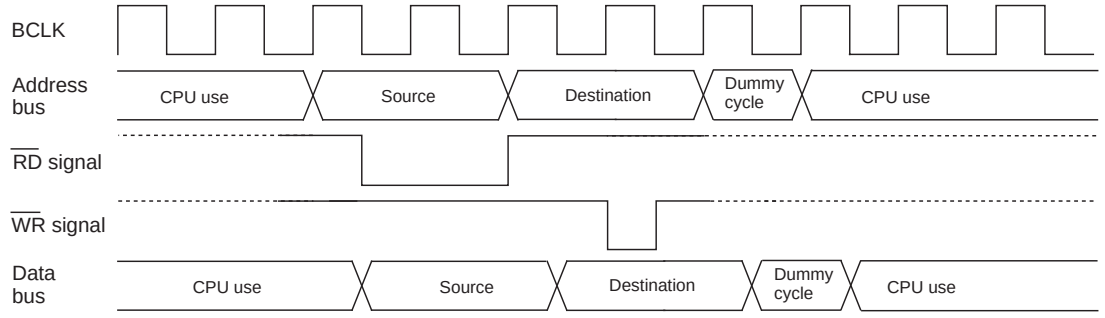
(1) When the transfer unit is 8 or 16 bits and the source of transfer is an even address



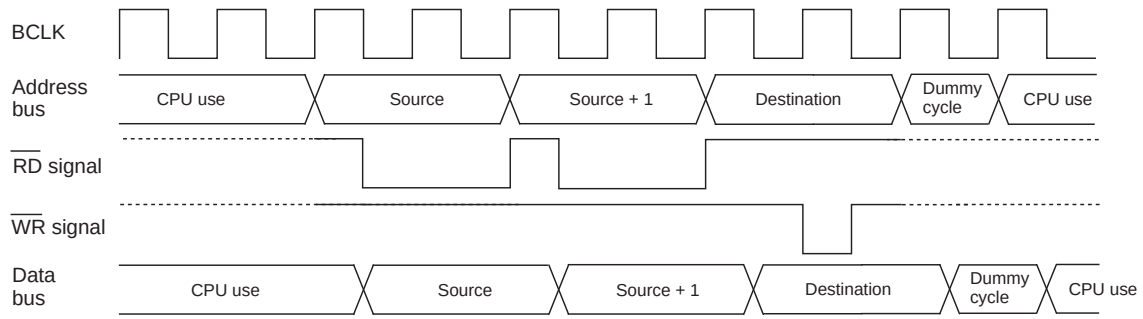
(2) When the transfer unit is 16 bits and the source address of transfer is an odd address, or when the transfer unit is 16 bits and an 8-bit bus is used



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



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



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

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:

No. of transfer cycles per transfer unit = No. of read cycles x j + No. of write cycles x 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”	• 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) $\overline{\text{HOLD}}$ Signal in Bus Control” for details about bus arbitration between the CPU and DMA.

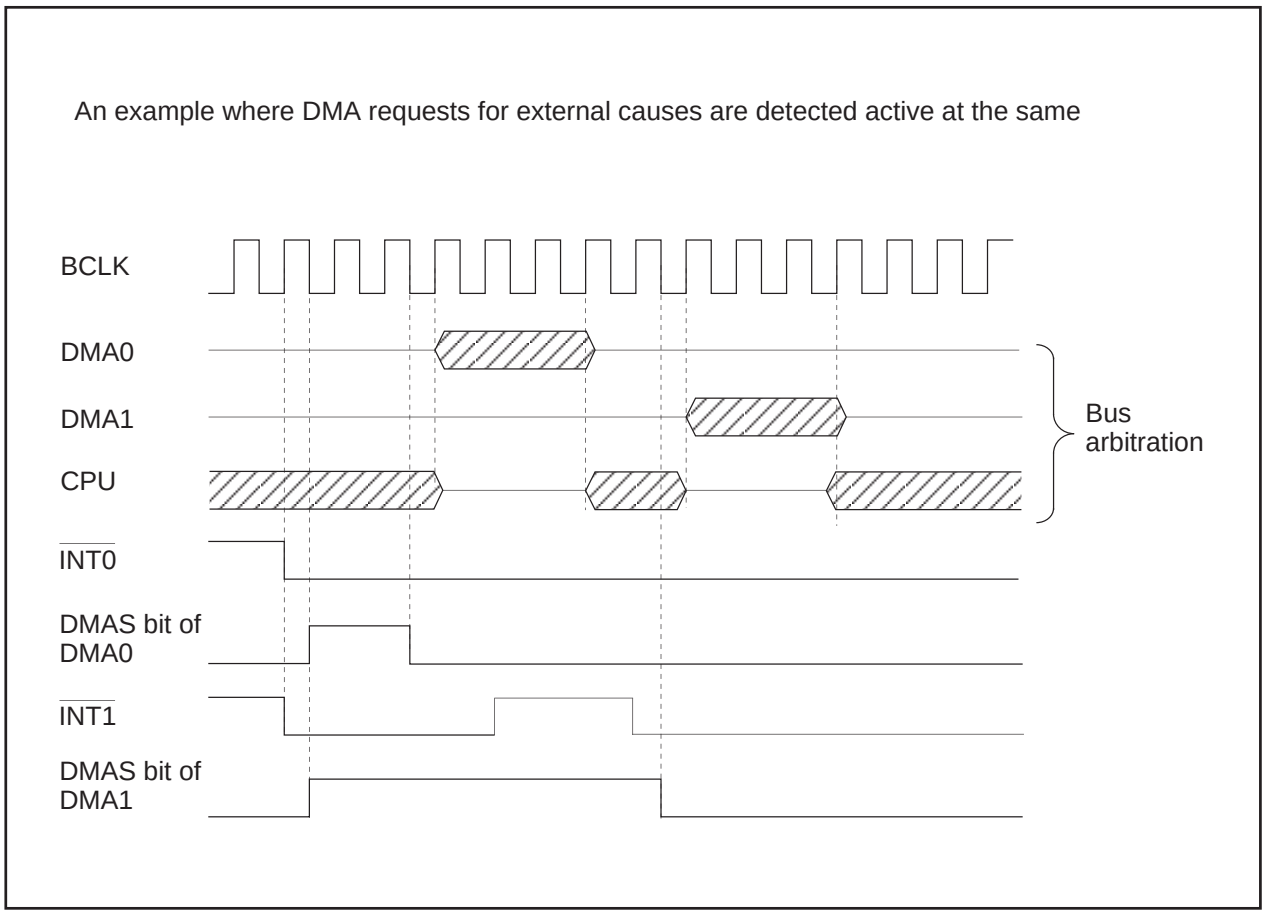
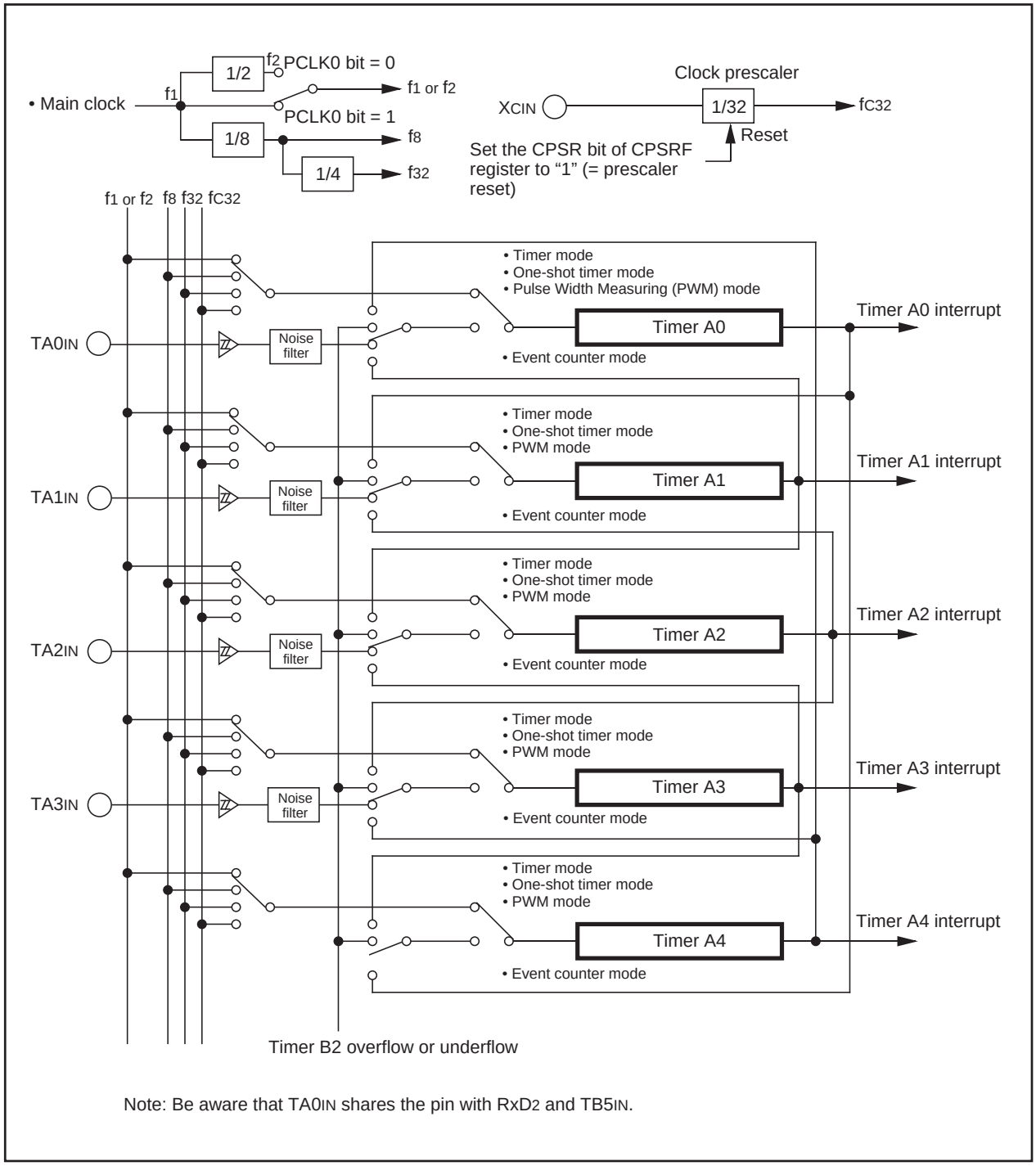


Figure 9.6. DMA Transfer by External Factors

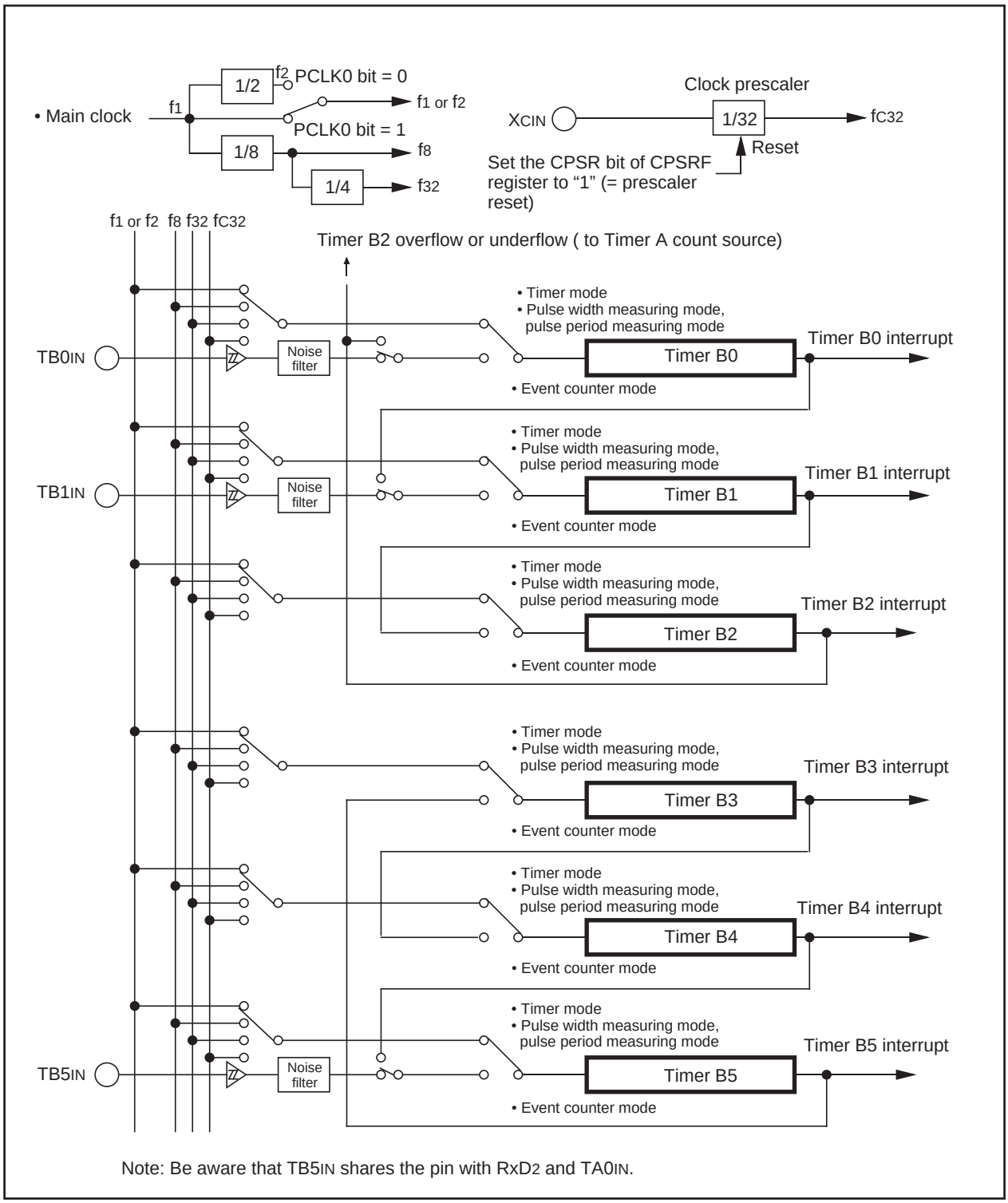
Timers

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.



Note: Be aware that TA0IN shares the pin with RxD2 and TB5IN.

Figure 10.1. Timer A Configuration



Note: Be aware that TB5IN shares the pin with RxD2 and TA0IN.

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 "0000_{16}."
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.

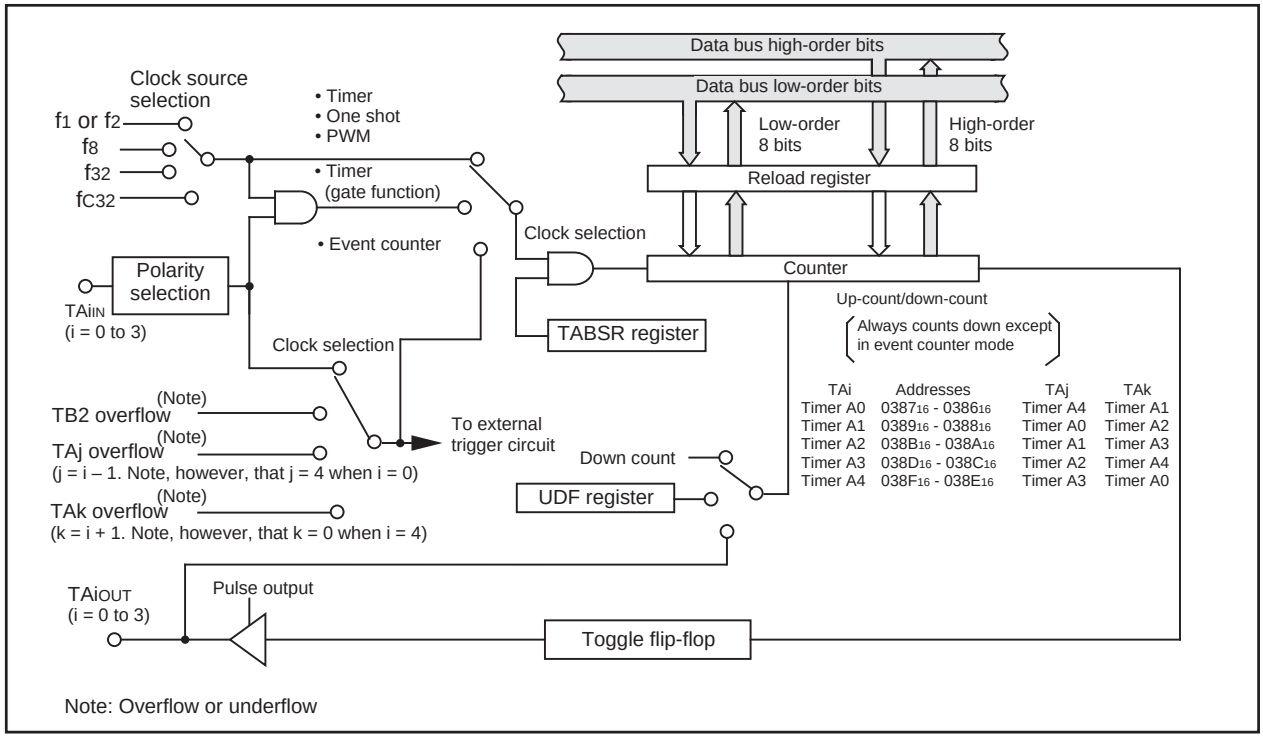


Figure 10.3. Timer A Block Diagram

Timer Ai mode register (i=0 to 4)

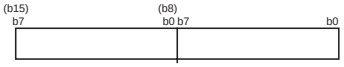
Symbol	Address	After reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

b7	b6	b5	b4	b3	b2	b1	b0

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	RW
TMOD1			RW
MR0	Function varies with each operation mode		RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count source selection bit (Function varies with each operation mode)		RW
TCK1			RW

Figure 10.4. TA0MR to TA4MR Registers

Timer Ai register (i= 0 to 4) (Note 1)



Symbol	Address	After reset
TA0	0387₁₆, 0386₁₆	Indeterminate
TA1	0389₁₆, 0388₁₆	Indeterminate
TA2	038B₁₆, 038A₁₆	Indeterminate
TA3	038D₁₆, 038C₁₆	Indeterminate
TA4	038F₁₆, 038E₁₆	Indeterminate

Mode	Function	Setting range	RW
Timer mode	Divide the count source by n + 1 where n = set value	0000₁₆ to FFFF₁₆	RW
Event counter mode	Divide the count source by FFFF₁₆ – n + 1 where n = set value when counting up or by n + 1 when counting down (Note 5)	0000₁₆ to FFFF₁₆	RW
One-shot timer mode	Divide the count source by n where n = set value and cause the timer to stop	0000₁₆ to FFFF₁₆ (Notes 2, 4)	WO
Pulse width modulation mode (16-bit PWM)	Modify the pulse width as follows: PWM period: (2^{16} – 1) / fj High level PWM pulse width: n / fj where n = set value, fj = count source frequency	0000₁₆ to FFFE₁₆ (Note 3, 4)	WO
Pulse width modulation mode (8-bit PWM)	Modify the pulse width as follows: PWM period: (2^8 – 1) x (m + 1)/ fj High level PWM pulse width: (m + 1)n / fj where n = high-order address set value, m = low-order address set value, fj = count source frequency	00₁₆ to FE₁₆ (High-order address) 00₁₆ to FF₁₆ (Low-order address) (Note 3, 4)	WO

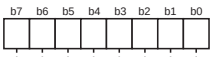
Note 1: The register must be accessed in 16-bit units.
Note 2: If the TAi register is set to '0000₁₆,' the counter does not work and timer Ai interrupt requests are not generated either. Furthermore, if "pulse output" is selected, no pulses are output from the TAiOUT pin.
Note 3: If the TAi register is set to '0000₁₆,' the pulse width modulator does not work, the output level on the TAiOUT pin remains low, and timer Ai interrupt requests are not generated either. The same applies when the 8 high-order bits of the timer TAi register are set to '00₁₆' while operating as an 8-bit pulse width modulator.
Note 4: Use the MOV instruction to write to the TAi register.
Note 5: The timer counts pulses from an external device or overflows or underflows in other timers.

Count start flag

Symbol	Address	After reset
TABSR	0380₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

Up/down flag (Note 1)



Symbol	Address	After reset
UDF	0384₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count Enabled by setting the TAiMR register's MR2 bit to "0" (= switching source in UDF register) during event counter mode.	RW
TA1UD	Timer A1 up/down flag		RW
TA2UD	Timer A2 up/down flag		RW
TA3UD	Timer A3 up/down flag		RW
TA4UD	Timer A4 up/down flag		RW
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled (Notes 2, 3)	WO
TA3P	Timer A3 two-phase pulse signal processing select bit		WO
TA4P	Timer A4 two-phase pulse signal processing select bit		WO

Note 1: Use MOV instruction to write to this register.
Note 2: Make sure the port direction bits for the TA2IN to TA3IN and TA2OUT to TA3OUT pins are set to "0" (input mode).
Note 3: When not using the two-phase pulse signal processing function, set corresponding bits to "0".

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

One-shot start flag

Symbol	Address	After reset
ONSF	038216	0016

Bit symbol	Bit name	Function	RW
TA0OS	Timer A0 one-shot start flag	The timer starts counting by setting this bit to "1" while the TMOD1 to TMOD0 bits of TAiMR register (i = 0 to 4) = '102' (= one-shot timer mode) and the MR2 bit of TAiMR register = "0" (=TAiOS bit enabled). When read, its content is "0".	RW
TA1OS	Timer A1 one-shot start flag		RW
TA2OS	Timer A2 one-shot start flag		RW
TA3OS	Timer A3 one-shot start flag		RW
TA4OS	Timer A4 one-shot start flag		RW
TA4OS	Z-phase input enable bit	0 : Z-phase input disabled 1 : Z-phase input enabled	RW
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on TA0IN is selected (Note 1) 0 1 : TB2 overflow is selected (Note 2) 1 0 : TA4 overflow is selected (Note 2) 1 1 : TA1 overflow is selected (Note 2)	RW
TA0TGH			RW

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

Trigger select register

Symbol	Address	After reset
TRGSR	038316	0016

Bit symbol	Bit name	Function	RW
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on TA1IN is selected (Note) 0 1 : TB2 is selected 1 0 : TA0 is selected 1 1 : TA2 is selected	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on TA2IN is selected (Note) 0 1 : TB2 is selected 1 0 : TA1 is selected 1 1 : TA3 is selected	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3IN is selected (Note) 0 1 : TB2 is selected 1 0 : TA2 is selected 1 1 : TA4 is selected	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Must not to be set 0 1 : TB2 is selected 1 0 : TA3 is selected 1 1 : TA0 is selected	RW
TA4TGH			RW

Note : Make sure the port direction bits for the TA1IN to TA3IN pins are set to "0" (= input mode).

Clock prescaler reset flag

Symbol	Address	After reset
CPSRF	038116	0XXXXXXX2

Bit symbol	Bit name	Function	RW
— (b6-b0)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—
CPSR	Clock prescaler reset flag	Setting this bit to "1" initializes the prescaler for the timekeeping clock. (When read, its content is "0".)	RW

Figure 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 TAiMR register in timer mode.

Table 10.1. Specifications in Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TAiMR register (i= 0 to 4) 0000_{16} to $FFFF_{16}$
Count start condition	Set TAiS bit of TABSR register to "1" (= start counting)
Count stop condition	Set TAiS bit to "0" (= stop counting)
Interrupt request generation timing	At underflow
TAi_{IN} pin function	I/O port or gate input
TAi_{OUT} pin function	I/O port or pulse output
Read from timer	Count value can be read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Gate function Counting can be started and stopped by an input signal to TAi_{IN} pin • Pulse output function Whenever the timer underflows, the output polarity of TAi_{OUT} pin is inverted. While the TAiS bit is set to "0", the pin outputs an "L" level signal during the count stop.

Timer Ai mode register (i=0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol: TA0MR to TA4MR — Address: 0396_{16} to $039A_{16}$ — After reset: 00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0 (Note 3)	Pulse output function select bit	0 : Pulse is not output (TAi_{OUT} pin is a normal port pin) 1 : Pulse is output (Note 1) (TAi_{OUT} pin is a pulse output pin)	RW
MR1 (Note 3)	Gate function select bit	b4 b3 0 0 : 0 1 : } Gate function not available (TAi_{IN} pin functions as I/O port) 1 0 : Counts while input on the TAi_{IN} pin is low (Note 2) 1 1 : Counts while input on the TAi_{IN} pin is high (Note 2)	RW
MR2 (Note 3)			RW
MR3	Must be set to "0" in timer mode		RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

Note 1: $TA0_{OUT}$ pin is N-channel open drain output.
Note 2: The port direction bit for the TAi_{IN} pin must be set to "0" (= input mode).
Note 3: Bits MR0, MR1 and MR2 of TA4MR are set to "0".

Figure 10.7. TAiMR 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 TAiMR 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	• External signals input to TAiIN pin (i=0 to 3) (effective edge can be selected in program) • Timer B2 overflows or underflows, timer Aj (j=i-1, except j=4 if i=0) overflows or underflows, timer Ak (k=i+1, except k=0 if i=4) overflows or underflows
Count operation	• Up-count or down-count can be selected by external signal or program • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divided ratio	1/ ($FFFF_{16}$ - 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 TAiS bit of TABSR register to “1” (= start counting)
Count stop condition	Set TAiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	I/O port or count source input
TAiOUT pin function	I/O port, pulse output, or up/down-count select input
Read from timer	Count value can be read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Whenever the timer underflows or underflows, the output polarity of TAiOUT pin is inverted . When not counting, the pin outputs a low.

Timer Ai mode register (i=0 to 4)
(When not using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol	Address	After reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

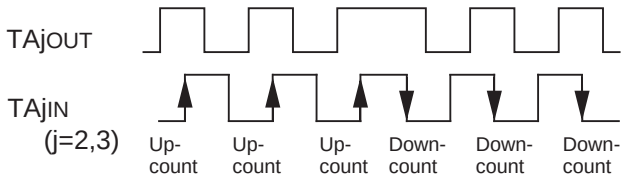
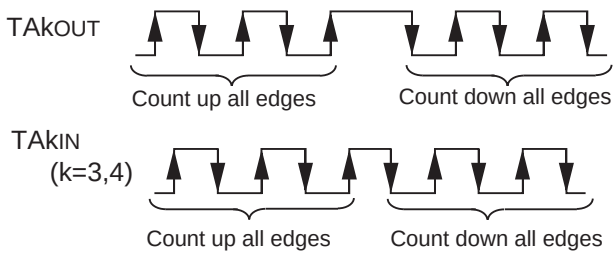
Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode (Note 1)	RW
TMOD1			RW
MR0 (Note 5)	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (Note 2) (TAiOUT pin functions as pulse output pin)	RW
MR1 (Note 5)	Count polarity select bit (Note 3)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	RW
MR2 (Note 5)	Up/down switching cause select bit	0 : UDF register 1 : Input signal to TAiOUT pin (Note 4)	RW
MR3	Must be set to “0” in event counter mode		RW
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	RW
TCK1	Can be “0” or “1” when not using two-phase pulse signal processing		RW

Note 1: During event counter mode, the count source can be selected using the ONSF and TRGSR registers.
Note 2: TA0OUT pin is N-channel open drain output.
Note 3: Effective when the TAiTGH and TAiTGL bits of ONSF or TRGSR register are ‘00_2’ (TAiIN pin input).
Note 4: Count down when input on TAiOUT pin is low or count up when input on that pin is high. The port direction bit for TAiOUT pin must be set to “0” (= input mode).
Note 5: Bits MR0, MR1 and MR2 of TA4MR are set to “0”.

Figure 10.8. TAiMR 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/ (FFFF16 - n + 1) for up-count 1/ (n + 1) for down-count n : set value of TAi register 000016 to FFFF16
Count start condition	Set TAiS bit of TABSR register to “1” (= start counting)
Count stop condition	Set TAiS bit to “0” (= stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	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)	• 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”.  • 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.  • 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.

Timer Ai mode register (i=2 to 3)
(When using two-phase pulse signal processing)

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

Symbol: TA2MR to TA3MR
Address: 0398$_{16}$ to 0399$_{16}$
After reset: 00$_{16}$

	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	To use two-phase pulse signal processing, set this bit to “0”.		RW
MR1	To use two-phase pulse signal processing, set this bit to “0”.		RW
MR2	To use two-phase pulse signal processing, set this bit to “1”.		RW
MR3	To use two-phase pulse signal processing, set this bit to “0”.		RW
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	RW
TCK1	Two-phase pulse signal processing operation select bit (Notes 1 and 2)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	RW

Note 1: Timer A3 can be chosew. Timer A2 is usually fixation irrespective of this bit at processing operation.
Note 2: If two-phase pulse signal processing is desired, following register settings are required:
- Set the UDF register’s TAiP bit to “1” (two-phase pulse signal processing function enabled).
- Set the TRGSR register’s TAiGH and TAiGL bits to ‘00$_2$’ (TAi$_{IN}$ pin input).
- Set the port direction bits for TAi$_{IN}$ and TAi$_{OUT}$ to “0” (input mode).

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 $\overline{INT2}$ pin.

Counter initialization by Z-phase input is enabled by writing “0000_{16}” 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 $\overline{INT2}$ 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.

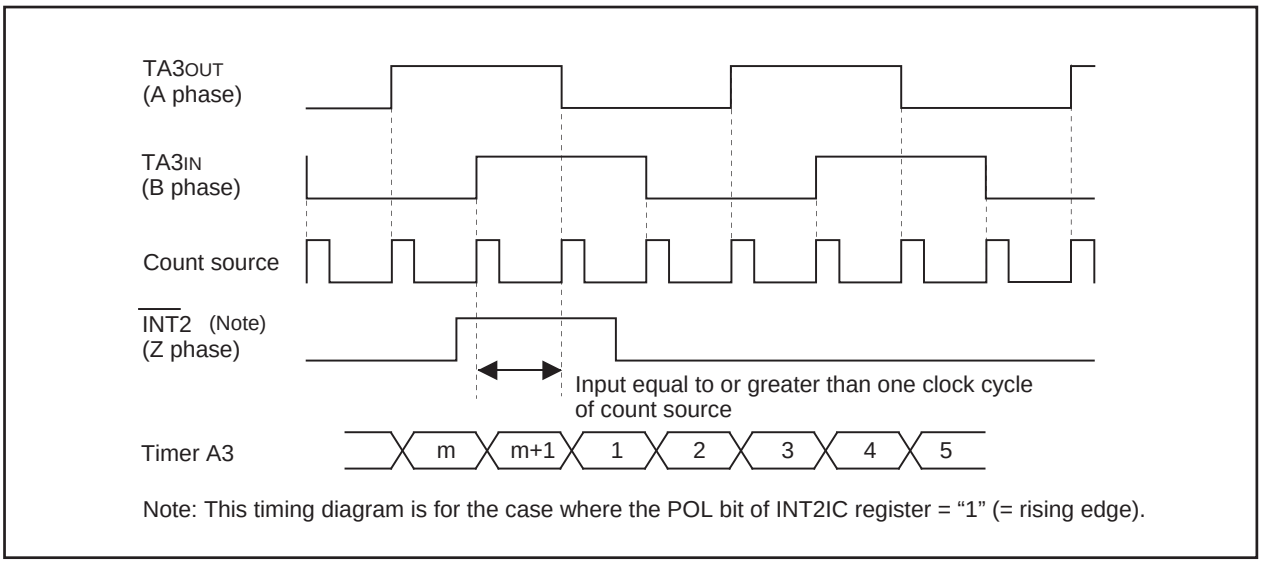


Figure 10.10. Two-phase Pulse (A phase and B phase) and the Z Phase

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	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count • When the counter reaches 0000_{16}, 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	TAiS bit of TABSR register = "1" (start counting) and one of the following triggers occurs. • External trigger input from the TAi_{IN} pin • Timer B2 overflow or underflow, timer Aj (j=i-1, except j=4 if i=0) overflow or underflow, timer Ak (k=i+1, except k=0 if i=4) overflow or underflow • The TAiOS bit of ONSF register is set to "1" (= timer starts)
Count stop condition	• When the counter is reloaded after reaching "0000_{16}" • TAiS 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	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Pulse output function The timer outputs a low when not counting and a high when counting.

Timer Ai mode register (i=0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol	Address	After reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	RW
TMOD1			RW
MR0 (Note 4)	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (Note 1) (TAiOUT pin functions as a pulse output pin)	RW
MR1 (Note 4)	External trigger select bit (Note 2)	0 : Falling edge of input signal to TAiIN pin (Note 3) 1 : Rising edge of input signal to TAiIN pin (Note 3)	RW
MR2	Trigger select bit	0 : TAiOS bit is enabled 1 : Selected by TAiTGH to TAiTGL bits	RW
MR3	Must be set to “0” in one-shot timer mode		RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : fC_{32}	RW
TCK1			RW

Note 1: TA0OUT pin is N-channel open drain output.
Note 2: Effective when the TAiTGH and TAiTGL bits of ONSF or TRGSR register are ‘00_2’ (TAiIN pin input).
Note 3: The port direction bit for the TAiIN pin must be set to “0” (= input mode).
Note 4: Bits MR0 and MR1 of TA4MR are set to “0”.

Figure 10.11. TAiMR Register in One-shot Timer Mode

4. Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see Table 10.5). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 10.12 shows TAiMR 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	• Down-count (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new value at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs during counting
16-bit PWM	• High level width n / fj n : set value of TAi register (i=0 to 3) • Cycle time $(2^{16}-1)$ / fj fixed fj: count source frequency (f_1, f_2, f_8, f_{32}, f_{C32})
8-bit PWM	• High level width n x (m+1) / fj n : set value of TAiMR register high-order address • Cycle time (2^8-1) x (m+1) / fj m : set value of TAiMR register low-order address
Count start condition	• TAiS bit of TABSR register is set to “1” (= start counting) • The TAiS bit = 1 and external trigger input from the TAi_{IN} pin • The TAiS bit = 1 and one of the following external triggers occurs • Timer B2 overflow or underflow, timer Aj (j=i-1, except j=4 if i=0) overflow or underflow, timer Ak (k=i+1) overflow or underflow
Count stop condition	TAiS 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	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)

Timer Ai mode register (i= 0 to 3)

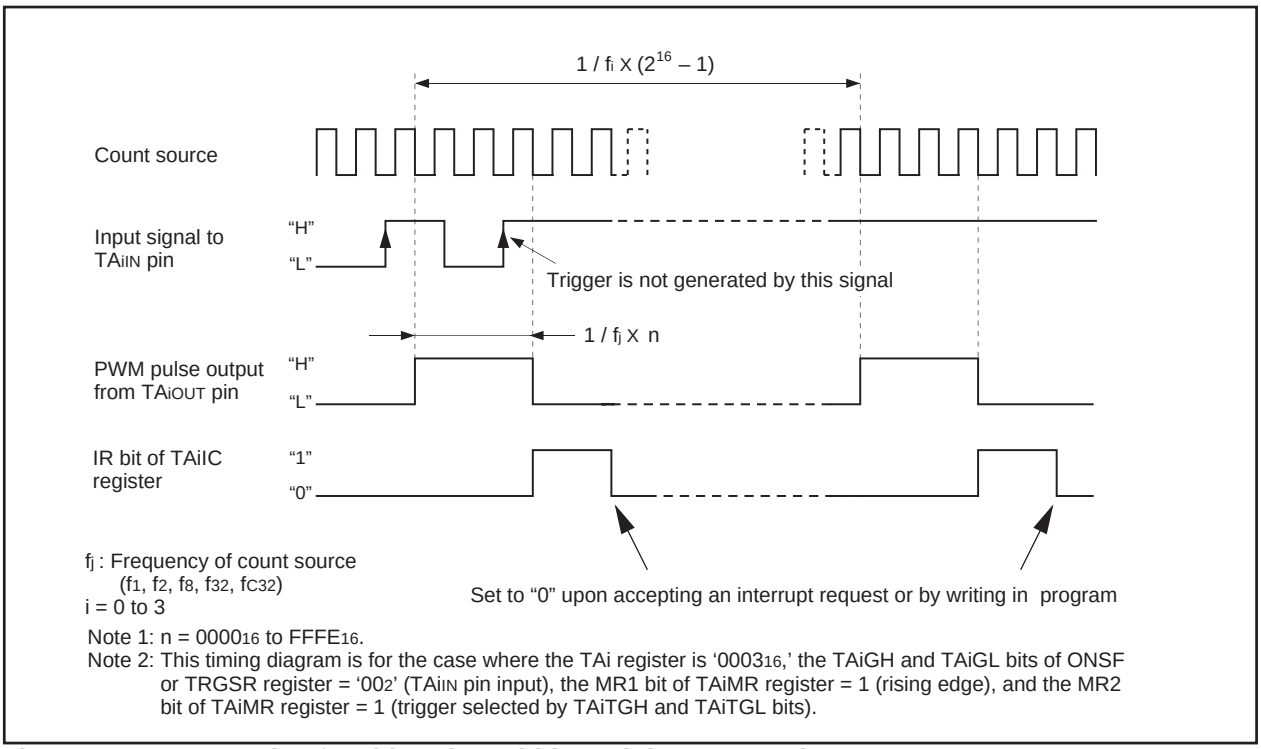
b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Symbol	Address	After reset
TA0MR to TA3MR	0396_{16} to 0399_{16}	00_{16}

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode (Note 1)	RW
TMOD1			RW
MR0	Must be set to “1” in PWM mode		RW
MR1	External trigger select bit (Note 2)	0: Falling edge of input signal to TAiIN pin(Note 3) 1: Rising edge of input signal to TAiIN pin(Note 3)	RW
MR2	Trigger select bit	0 : Write “1” to TAiS bit in the TASF register 1 : Selected by TAiTGH to TAiTGL bits	RW
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	RW
TCK1			RW

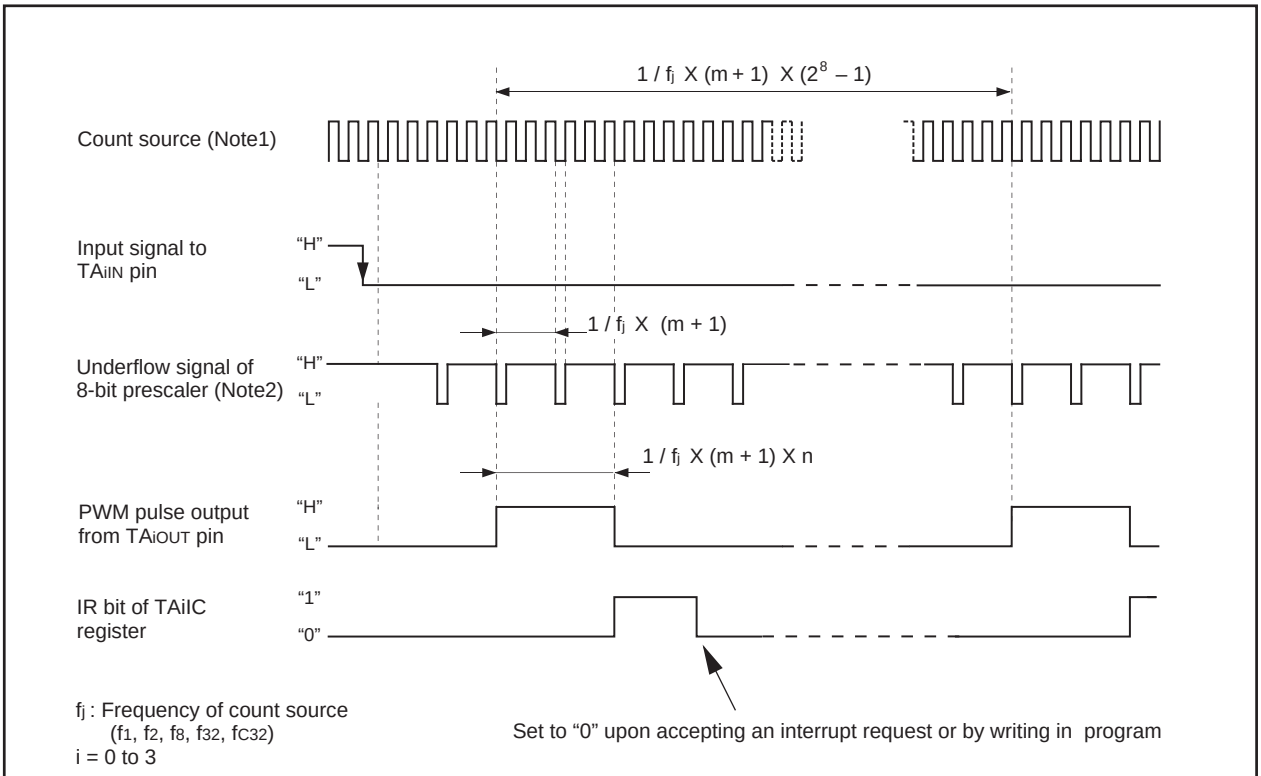
Note 1: TA0OUT pin is N-channel open drain output.
Note 2: Effective when the TAiTGH and TAiTGL bits of ONSF or TRGSR register are ‘00_2’ (TAiIN pin input).
Note 3: The port direction bit for the TAiIN pin must be set to “0” (= input mode).

Figure 10.12. TAiMR Register in PWM Mode



Note 1: n = 0000$_{16}$ to FFFE$_{16}$.
Note 2: This timing diagram is for the case where the TAi register is '0003$_{16}$,' the TAiGH and TAiGL bits of ONSF or TRGSR register = '00$_2$' (TAi$_{IN}$ pin input), the MR1 bit of TAiMR register = 1 (rising edge), and the MR2 bit of TAiMR register = 1 (trigger selected by TAiTGH and TAiTGL bits).

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



Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00$_{16}$ to FF$_{16}$; n = 00$_{16}$ to FE$_{16}$.
Note 4: This timing diagram is for the case where the TAi register is '0202$_{16}$,' the TAiGH and TAiGL bits of ONSF or TRGSR register = '00$_2$' (TAi$_{IN}$ pin input), the MR1 bit of TAiMR register = 0 (falling edge), and the MR2 bit of TAiMR register = 1 (trigger selected by TAiTGH and TAiTGL bits).

Figure 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.

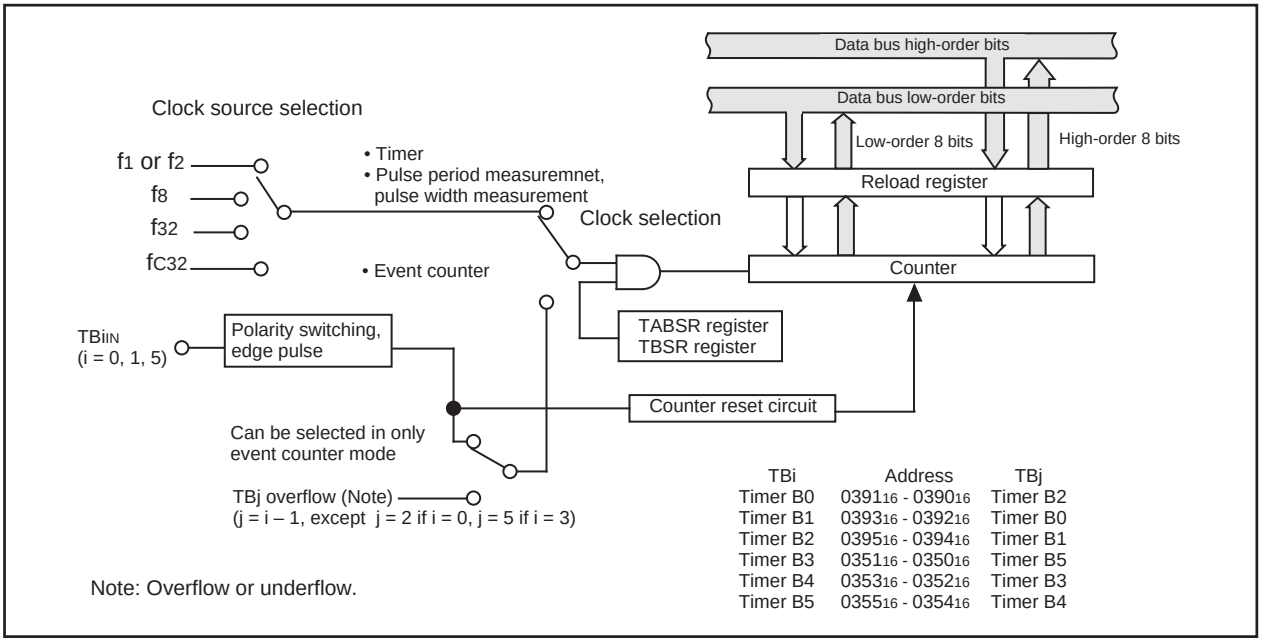


Figure 10.15. Timer B Block Diagram

Timer Bi mode register (i=0 to 5)

b7	b6	b5	b4	b3	b2	b1	b0

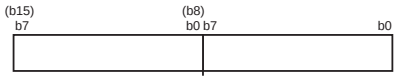
Symbol	Address	After reset
TB0MR to TB2MR	039B16 to 039D16	00XX00002
TB3MR to TB5MR	035B16 to 035D16	00XX00002

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period measurement mode, pulse width measurement mode 1 1 : Must not be set	RW
TMOD1			RW
MR0	Function varies with each operation mode		RW
MR1			RW
MR2			RW (Note 1) — (Note 2)
MR3			RO
TCK0	Count source select bit (Function varies with each operation mode)		RW
TCK1			RW

Note 1: Timer B0, timer B3.
Note 2: Timer B1, timer B2, timer B4, timer B5.

Figure 10.16. TB0MR to TB5MR Registers

Timer Bi register (i=0 to 5)(Note 1)

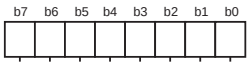


Symbol	Address	After reset
TB0	039116, 039016	Indeterminate
TB1	039316, 039216	Indeterminate
TB2	039516, 039416	Indeterminate
TB3	035116, 035016	Indeterminate
TB4	035316, 035216	Indeterminate
TB5	035516, 035416	Indeterminate

Mode	Function	Setting range	RW
Timer mode	Divide the count source by n + 1 where n = set value	000016 to FFFF16	RW
Event counter mode	Divide the count source by n + 1 where n = set value (Note 2)	000016 to FFFF16	RW
Pulse period modulation mode, Pulse width modulation mode	Measures a pulse period or width	——	RO

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

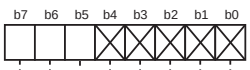
Count start flag



Symbol	Address	After reset
TABSR	038016	0016

Bit symbol	Bit name	Function	RW
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

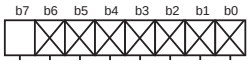
Timer B3, B4, B5 count start flag



Symbol	Address	After reset
TBSR	034016	000XXXXX2

Bit symbol	Bit name	Function	RW
—— (b4-b0)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—
TB3S	Timer B3 count start flag	0 : Stops counting 1 : Starts counting	RW
TB4S	Timer B4 count start flag		RW
TB5S	Timer B5 count start flag		RW

Clock prescaler reset flag



Symbol	Address	After reset
CPSRF	038116	0XXXXXXX2

Bit symbol	Bit name	Function	RW
—— (b6-b0)	Nothing is assigned. When write, set to “0”. When read, their contents are indeterminate.		—
CPSR	Clock prescaler reset flag	Setting this bit to “1” initializes the prescaler for the timekeeping clock. (When read, the value of this bit is “0”.)	RW

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	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of 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
TBi_{IN} pin function	I/O port
Read from timer	Count value can be read by reading TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

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.

Timer Bi mode register (i= 0 to 5)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol	Address	After reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$
TB3MR to TB5MR	$035B_{16}$ to $035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Has no effect in timer mode Can be set to "0" or "1"		RW
MR1			RW
MR2	TB0MR, TB3MR registers Must be set to "0" in timer mode		RW
	TB1MR, TB2MR, TB4MR, TB5MR registers Nothing is assigned. When write, set to "0". When read, its content is indeterminate		—
MR3	When write in timer mode, set to "0". When read in timer mode, its content is indeterminate.		RO
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

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	• 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	• 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	• 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.

Timer Bi mode register (i=0 to 5)

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol	Address	After reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$
TB3MR to TB5MR	$035B_{16}$ to $035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	Count polarity select bit (Note 1)	b3 b2 0 0 : Counts external signal's falling edges 0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Must not be set	RW
MR1			RW
MR2	TB0MR, TB3MR registers Must be set to “0” in timer mode		RW
	TB1MR, TB2MR, TB4MR, TB5MR registers Nothing is assigned. When write, set to “0”. When read, its content is indeterminate.		—
MR3	When write in event counter mode, set to “0”. When read in event counter mode, its content is indeterminate.		RO
TCK0	Has no effect in event counter mode. Can be set to “0” or “1”.		RW
TCK1 (Note 3)	Event clock select	0 : Input from TBiIN pin (Note 2) 1 : TBj overflow or underflow (j = i – 1, except j = 2 if i = 0, j = 5 if i = 3)	RW

Note 1: Effective when the TCK1 bit = “0” (input from TBiIN pin). If the TCK1 bit = “1” (TBj overflow or underflow), these bits can be set to “0” or “1.”
Note 2: The port direction bit for the TBiIN pin must be set to “0” (= input mode).
Note 3: Bit TCK1 of TB2MR, TB3MR and TB4MR set to “1.”

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	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Up-count • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to "0000_{16}" 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	• 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).
TBi_{IN} 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.

Timer Bi mode register (i=0, 1, 5)

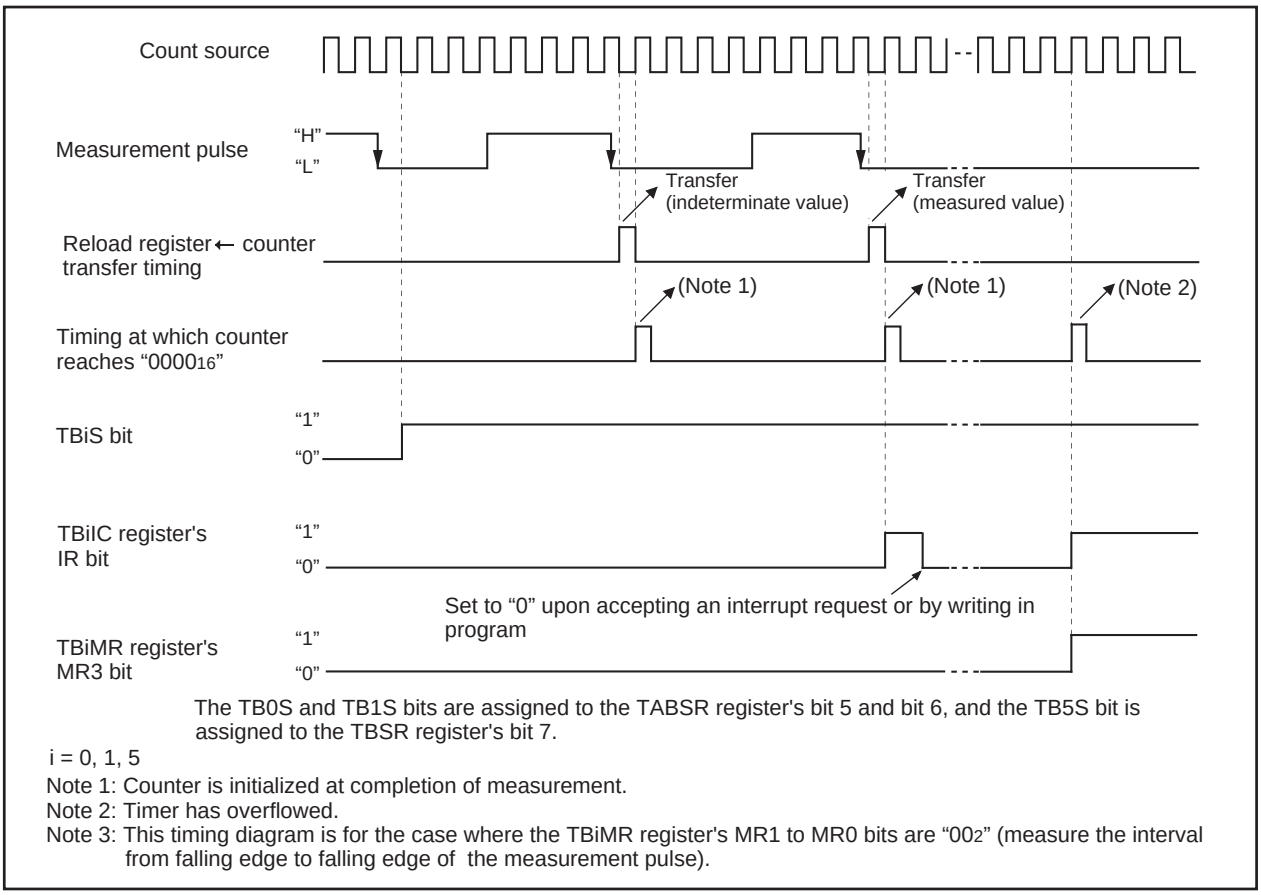
b7	b6	b5	b4	b3	b2	b1	b0
						1	0

Symbol	Address	After reset
TB0MR to TB1MR	$039B_{16}$ to $039C_{16}$	$00XX0000_2$
TB5MR	$035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	RW
TMOD1			RW
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Measurement between a falling edge and the next falling edge of measured pulse) 0 1 : Pulse period measurement (Measurement between a rising edge and the next rising edge of measured pulse) 1 0 : Pulse width measurement (Measurement between a falling edge and the next rising edge of measured pulse and between a rising edge and the next falling edge) 1 1 : Must not be set.	RW
MR1			RW
MR2	TB0MR register Must be set to "0" in pulse period and pulse width measurement mode		RW
	TB1MR, TB5MR registers Nothing is assigned. When write, set to "0". When read, its content turns out to be indeterminate.		—
MR3	Timer Bi overflow flag (Note)	0 : Timer did not overflow 1 : Timer has overflowed	RO
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

Note: This flag is indeterminate after reset. When the TBiS bit = 1 (start counting), the MR3 bit is cleared to "0" (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to "1" (overflowed). The MR3 bit cannot be set to "1" in a program. The TB0S to TB1S bits are assigned to the TABSR register's bit 5 to bit 6, and the TB5S bit is assigned to the TBSR register's bit 7.

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

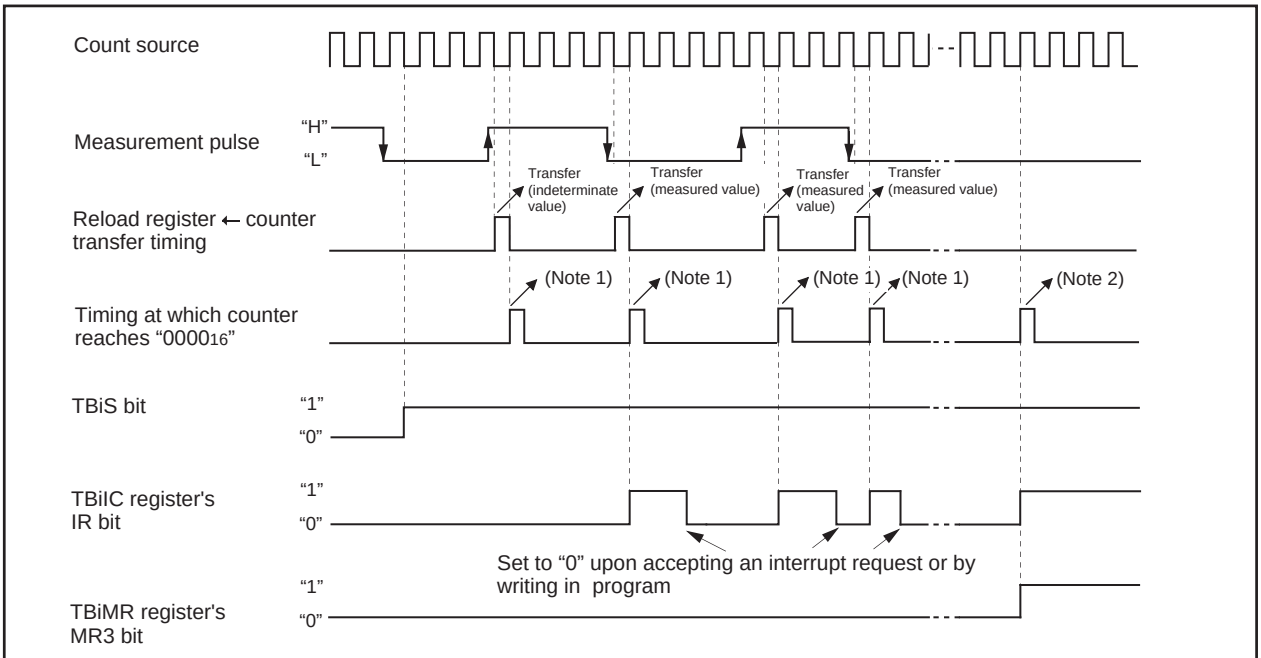


The TB0S and TB1S bits are assigned to the TABSR register's bit 5 and bit 6, and the TB5S bit is assigned to the TBSR register's bit 7.

i = 0, 1, 5

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.
Note 3: This timing diagram is for the case where the TBiMR register's MR1 to MR0 bits are "00₂" (measure the interval from falling edge to falling edge of the measurement pulse).

Figure 10.21. Operation timing when measuring a pulse period



The TB0S and TB1S bits are assigned to the TABSR register's bit 5 and bit 6, and the TB5S bit is assigned to the TBSR register's bit 7.

i = 0, 1, 5

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.
Note 3: This timing diagram is for the case where the TBiMR register's MR1 to MR0 bits are "10₂" (measure the interval from a falling edge to the next rising edge and the interval from a rising edge to the next falling edge of the measurement pulse).

Figure 10.22. Operation timing when measuring a pulse width

Serial I/O

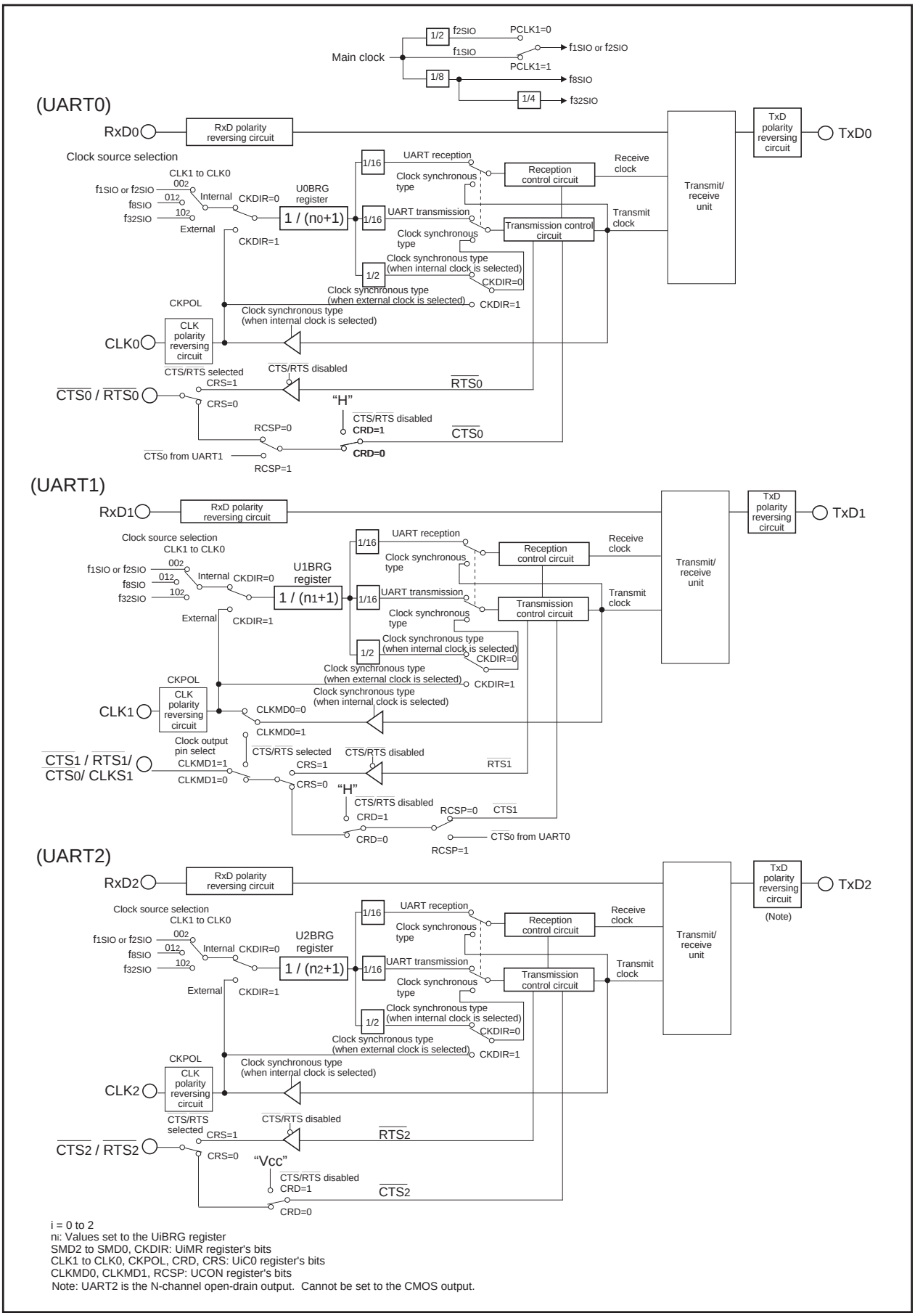
Serial I/O is configured with 3 channels: UART0 to UART2.
Each is explained below.

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. Figures 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.



i = 0 to 2
ni: Values set to the UiBRG register
SMD2 to SMD0, CKDIR: UiMR register's bits
CLK1 to CLK0, CKPOL, CRD, CRS: UiC0 register's bits
CLKMD0, CLKMD1, RCSP: UCON register's bits
Note: UART2 is the N-channel open-drain output. Cannot be set to the CMOS output.

Figure 11.1. UARTi Block Diagram

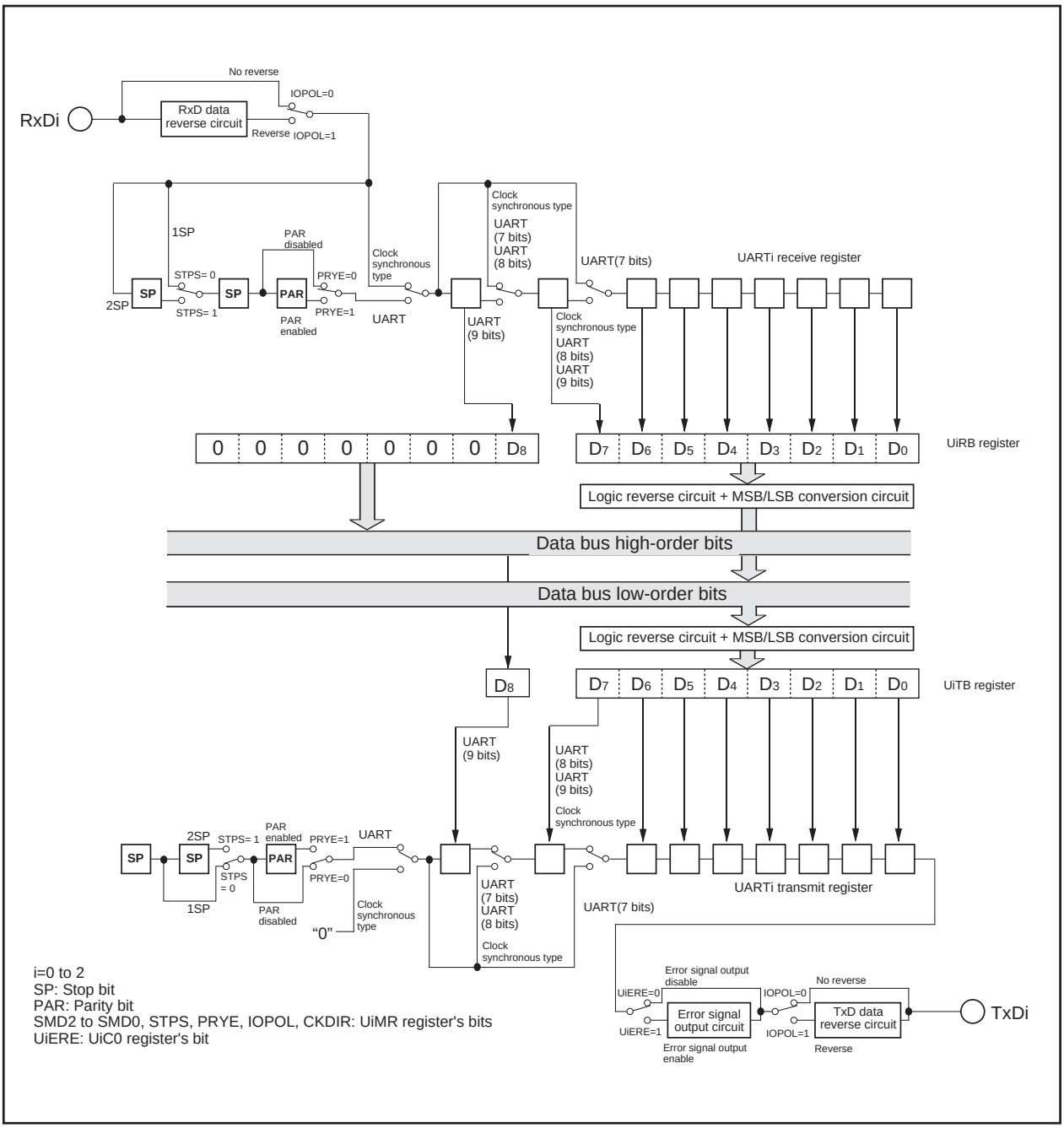
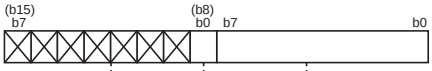


Figure 11.2. UARTi Transmit/Receive Unit

UARTi transmit buffer register (i=0 to 2)(Note)

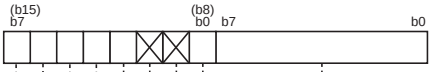


Symbol	Address	After reset
U0TB	$03A3_{16}$-$03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$-$03AA_{16}$	Indeterminate
U2TB	$037B_{16}$-$037A_{16}$	Indeterminate

Function	RW
Transmit data	WO
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.	—

Note: Use MOV instruction to write to this register.

UARTi receive buffer register (i=0 to 2)

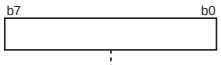


Symbol	Address	After reset
U0RB	$03A7_{16}$-$03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$-$03AE_{16}$	Indeterminate
U2RB	$037F_{16}$-$037E_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
— (b7-b0)	———	Receive data (D_7 to D_0)	RO
— (b8)	———	Receive data (D_8)	RO
— (b10-b9)	Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.		—
ABT	Arbitration lost detecting flag (Note 2)	0 : Not detected 1 : Detected	RW
OER	Overrun error flag (Note 1)	0 : No overrun error 1 : Overrun error found	RO
FER	Framing error flag (Note 1)	0 : No framing error 1 : Framing error found	RO
PER	Parity error flag (Note 1)	0 : No parity error 1 : Parity error found	RO
SUM	Error sum flag (Note 1)	0 : No error 1 : Error found	RO

Note 1: When the UiMR register’s SMD2 to SMD0 bits = “000_2” (serial I/O disabled) or the UiC1 register’s RE bit = “0” (reception disabled), all of the SUM, PER, FER and OER bits are set to “0” (no error). The SUM bit is set to “0” (no error) when all of the PER, FER and OER bits = “0” (no error). Also, the PER and FER bits are set to “0” by reading the lower byte of the UiRB register.
Note 2: The ABT bit is set to “0” by writing “0” in a program. (Writing “1” has no effect.)

UARTi baud rate generation register (i=0 to 2)(Notes 1, 2)



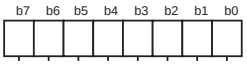
Symbol	Address	After reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate
U2BRG	0379_{16}	Indeterminate

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

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

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)

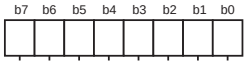


Symbol	Address	After reset
U0MR to U2MR	$03A0_{16}$, $03A8_{16}$, 0378_{16}	00_{16}

Bit symbol	Bit name	Function	RW
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)



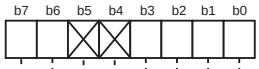
Symbol	Address	After reset
U0C0 to U2C0	$03A4_{16}$, $03AC_{16}$, $037C_{16}$	00001000_2

Bit symbol	Bit name	Function	RW
CLK0	BRG count source select bit	b1 b0 0 0 : $f1_{SIO}$ or $f2_{SIO}$ is selected 0 1 : $f8_{SIO}$ is selected 1 0 : $f32_{SIO}$ is selected 1 1 : Must not be set	RW
CLK1			RW
CRS	$\overline{CTS}/\overline{RTS}$ function select bit (Note 4)	Effective when CRD = 0 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected	RW
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	RO
CRD	$\overline{CTS}/\overline{RTS}$ disable bit	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : $\overline{CTS}/\overline{RTS}$ 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 $\overline{CTSi}$ pin to “0” (input mode).
Note 2: TxD_2 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: $\overline{CTS_1}/\overline{RTS_1}$ can be used when the UCON register’s CLKMD1 bit = “0” (only CLK_1 output) and the UCON register’s RCSP bit = “0” ($\overline{CTS_0}/\overline{RTS_0}$ not separated).

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

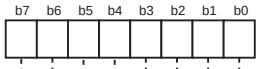
UARTi transmit/receive control register 1 (i=0, 1)



Symbol: U0C1, U1C1 — Address: 03A5$_{16}$,03AD$_{16}$ — After reset: 00XX0010$_2$

Bit symbol	Bit name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in UiTB register 1 : No data present in UiTB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data present in UiRB register 1 : Data present in UiRB register	RO
— (b5-b4)	Nothing is assigned. When write, set “0”. When read, these contents are “0”.		—
UiLCH	Data logic select bit	0 : No reverse 1 : Reverse	RW
UiERE	Error signal output enable bit	0 : Output disabled 1 : Output enabled	RW

UART2 transmit/receive control register 1



Symbol: U2C1 — Address: 037D$_{16}$ — After reset: 00000010$_2$

Bit symbol	Bit name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in U2TB register 1 : No data present in U2TB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data present in U2RB register 1 : Data present in U2RB register	RO
U2IRS	UART2 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	RW
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
U2LCH	Data logic select bit	0 : No reverse 1 : Reverse	RW
U2ERE	Error signal output enable bit	0 : Output disabled 1 : Output enabled	RW

Figure 11.5. U0C1 to U2C1 Registers

UART transmit/receive control register 2

b7	b6	b5	b4	b3	b2	b1	b0
☒							

Symbol	Address	After reset
UCON	$03B0_{16}$	$X0000000_2$

Bit symbol	Bit name	Function	RW
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U1IRS	UART1 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	RW
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
CLKMD0	UART1 CLK/CLKS select bit 0	Effective when CLKMD1 = "1" 0 : Clock output from CLK1 1 : Clock output from CLKS1	RW
CLKMD1	UART1 CLK/CLKS select bit 1 (Note)	0 : CLK output is only CLK1 1 : Transfer clock output from multiple pins function selected	RW
RCSP	Separate UART0 CTS/RTS bit	0 : $\overline{CTS}/\overline{RTS}$ shared pin 1 : $\overline{CTS}/\overline{RTS}$ separated ($\overline{CTS}_0$ supplied from the $P6_4$ pin)	RW
— (b7)	Nothing is assigned. When write, set "0". When read, its content is indeterminate.		—

Note: When using multiple transfer clock output pins, make sure the following conditions are met:
U1MR register's CKDIR bit = "0" (internal clock)

UARTi special mode register (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
☒				0	0	0	0

Symbol	Address	After reset
U0SMR to U2SMR	$036F_{16}$, 0373_{16}, 0377_{16}	$X0000000_2$

Bit symbol	Bit name	Function	RW
Reserved bits		Set to "0"	RW
ABSCS	Bus collision detect sampling clock select bit	0 : Rising edge of transfer clock 1 : Underflow signal of timer Aj (Note 1)	RW
ACSE	Auto clear function select bit of transmit enable bit	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	RW
SSS	Transmit start condition select bit	0 : Not synchronized to RxDi 1 : Synchronized to RxDi (Note 2)	RW
— (b7)	Nothing is assigned. When write, set "0". When read, its content is indeterminate.		—

Note 1: Underflow signal of timer A3 in UART0, underflow signal of timer A4 in UART1, underflow signal of timer A0 in UART2.
Note 2: SSS bit will be set to "0" (unrelated to RxDi) if transmission starts.

Figure 11.6. UCON Register and U0SMR to U2SMR Registers

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

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

Symbol	Address	After reset
U0SMR2 to U2SMR2	$036E_{16}$, 0372_{16}, 0376_{16}	$X0000000_2$

Bit symbol	Bit name	Function	RW
Reserved bits		Set to “0”	RW
— (b7)		Nothing is assigned. When write, set “0”. When read, its content is indeterminate.	—

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

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

Symbol	Address	After reset
U0SMR3 to U2SMR3	$036D_{16}$, 0371_{16}, 0375_{16}	$000X0X0X_2$

Bit symbol	Bit name	Function	RW
— (b0)		Nothing is assigned. When write, set “0”. When read, its content is indeterminate.	—
CKPH	Clock phase set bit	0 : Without clock delay 1 : With clock delay	RW
— (b2)		Nothing is assigned. When write, set “0”. When read, its content is indeterminate.	—
NODC	Clock output select bit	0 : CLKi is CMOS output 1 : CLKi is N-channel open drain output	RW
— (b4)		Nothing is assigned. When write, set “0”. When read, its content is indeterminate.	—
Reserved bits		Set to “0”	RW

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

UARTi special mode register 4 (i=0 to 2)

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

Symbol	Address	After reset
U0SMR4 to U2SMR4	$036C_{16}$, 0370_{16}, 0374_{16}	00_{16}

Bit symbol	Bit name	Function	RW
Reserved bits		Set to “0”	RW

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	• Transfer data length: 8 bits
Transfer clock	• UiMR(i=0 to 2) register's CKDIR bit = "0" (internal clock) : fj/ 2(n+1) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} • CKDIR bit = "1" (external clock) : Input from CLKi pin
Transmission, reception control	• Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disable
Transmission start condition	• Before transmission can start, the following requirements must be met (Note 1) • The TE bit of UiC1 register= 1 (transmission enabled) • The TI bit of UiC1 register = 0 (data present in UiTB register) • If $\overline{CTS}$ function is selected, input on the $\overline{CTS}$i pin = "L"
Reception start condition	• Before reception can start, the following requirements must be met (Note 1) • The RE bit of UiC1 register= 1 (reception enabled) • The TE bit of UiC1 register= 1 (transmission enabled) • The TI bit of UiC1 register= 0 (data present in the UiTB register)
Interrupt request generation timing	• For transmission, one of the following conditions can be selected • 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	• 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	• CLK polarity selection Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock • LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Continuous receive mode selection Reception is enabled immediately by reading the UiRB register • Switching serial data logic This function reverses the logic value of the transmit/receive data • Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set • Separate $\overline{CTS}/\overline{RTS}$ pins (UART0) $\overline{CTS_0}$ and $\overline{RTS_0}$ 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 “001_2”
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to “0”
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select $\overline{CTS}$ or $\overline{RTS}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ 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 $\overline{CTS_0}$ signal from the $P6_4$ 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 $P6_4$ pin functions during clock synchronous serial I/O mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an "H". (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

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

Pin name	Function	Method of selection
TxDi (i = 0 to 2) ($P6_3$, $P6_7$, $P7_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi ($P6_2$, $P6_6$, $P7_1$)	Serial data input	PD6 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 ($P6_1$, $P6_5$, $P7_2$)	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
$\overline{CTS}i/\overline{RTS}i$ ($P6_0$, $P6_4$, $P7_3$)	$\overline{CTS}$ 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
	$\overline{RTS}$ output	UiC0 register's CRD bit=0 UiC0 register's CRS bit=1
	I/O port	UiC0 register's CRD bit=1

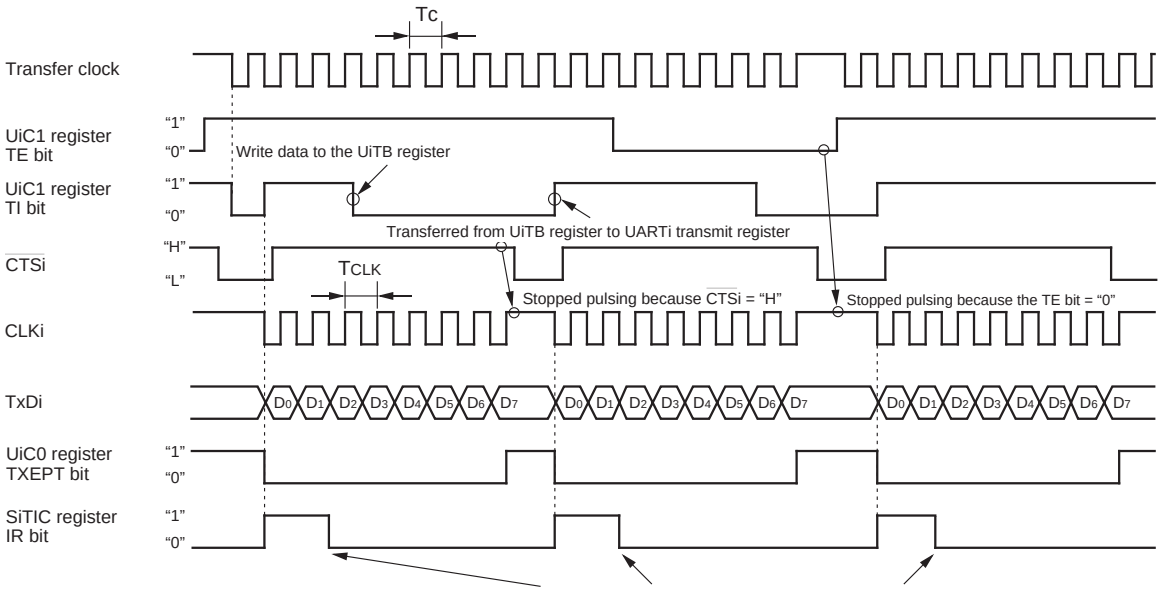
Table 11.4. $P6_4$ Pin Functions

Pin function	Bit set value					
	U1C0 register		UCON register			PD6 register
	CRD	CRS	RCSP	CLKMD1	CLKMD0	PD6_4
$P6_4$	1	—	0	0	—	Input: 0, Output: 1
$\overline{CTS}_1$	0	0	0	0	—	0
$\overline{RTS}_1$	0	1	0	0	—	—
$\overline{CTS}_0$(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" ($\overline{CTS}_0/\overline{RTS}_0$ enabled) and the U0C0 register's CRS bit to "1" ($\overline{RTS}_0$ 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

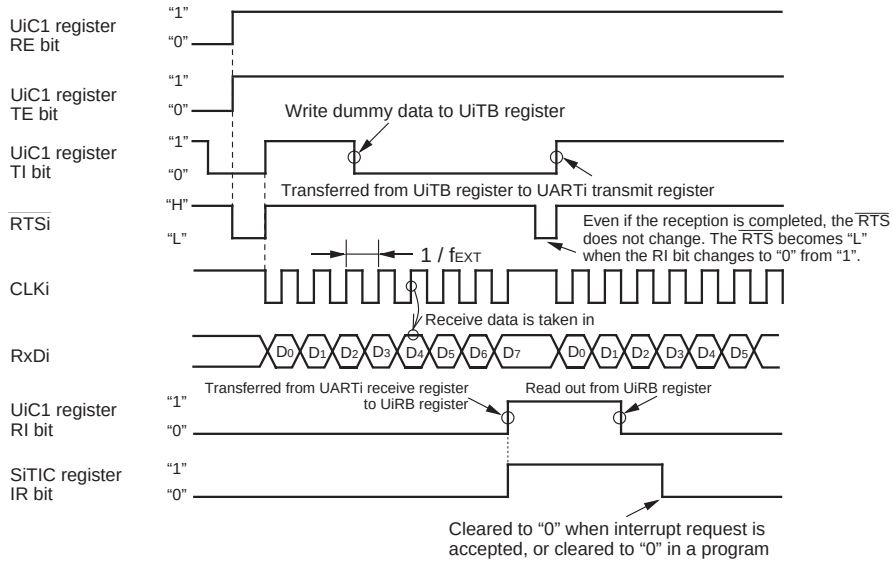
(1) Example of transmit timing (when internal clock is selected)



Tc = T_{CLK} = 2(n + 1) / fj
fj: frequency of UiBRG count source (f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO})
n: value set to UiBRG register
i: 0 to 2

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)

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

f_{EXT}: 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.

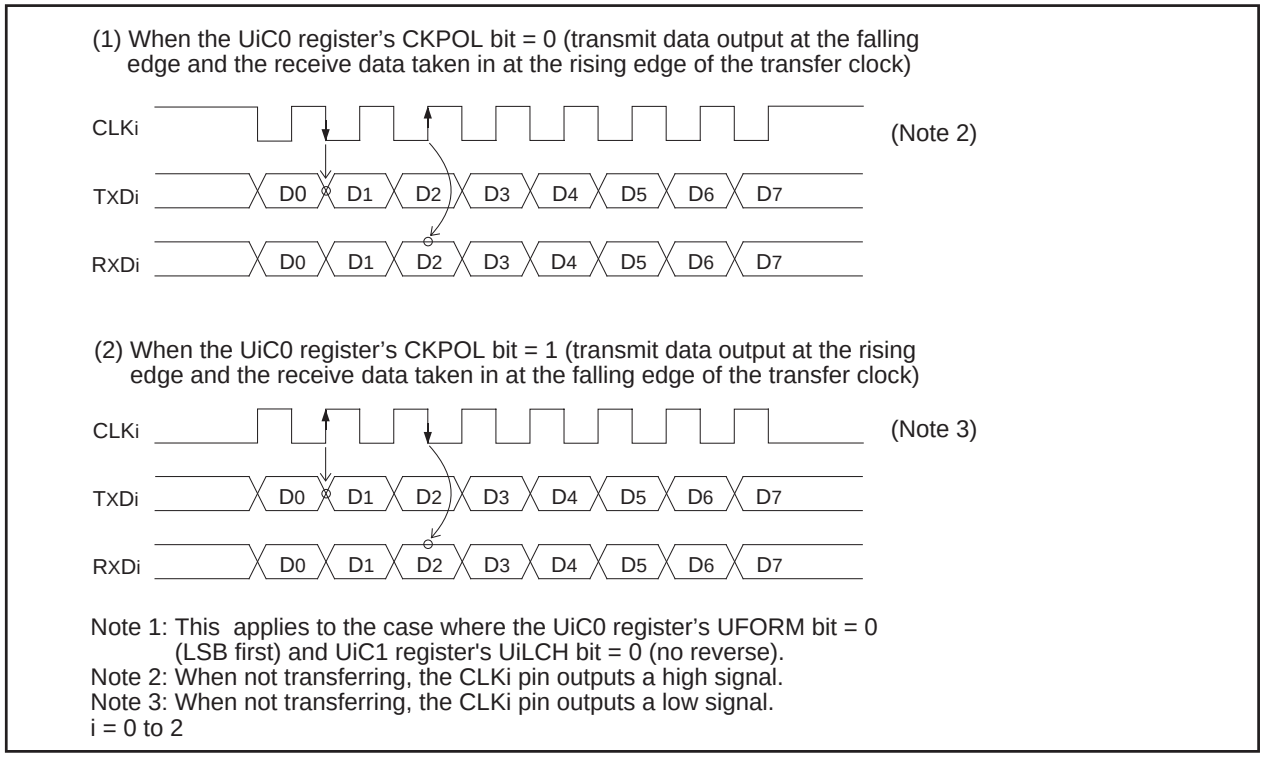


Figure 11.10. Transfer Clock Polarity

(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.

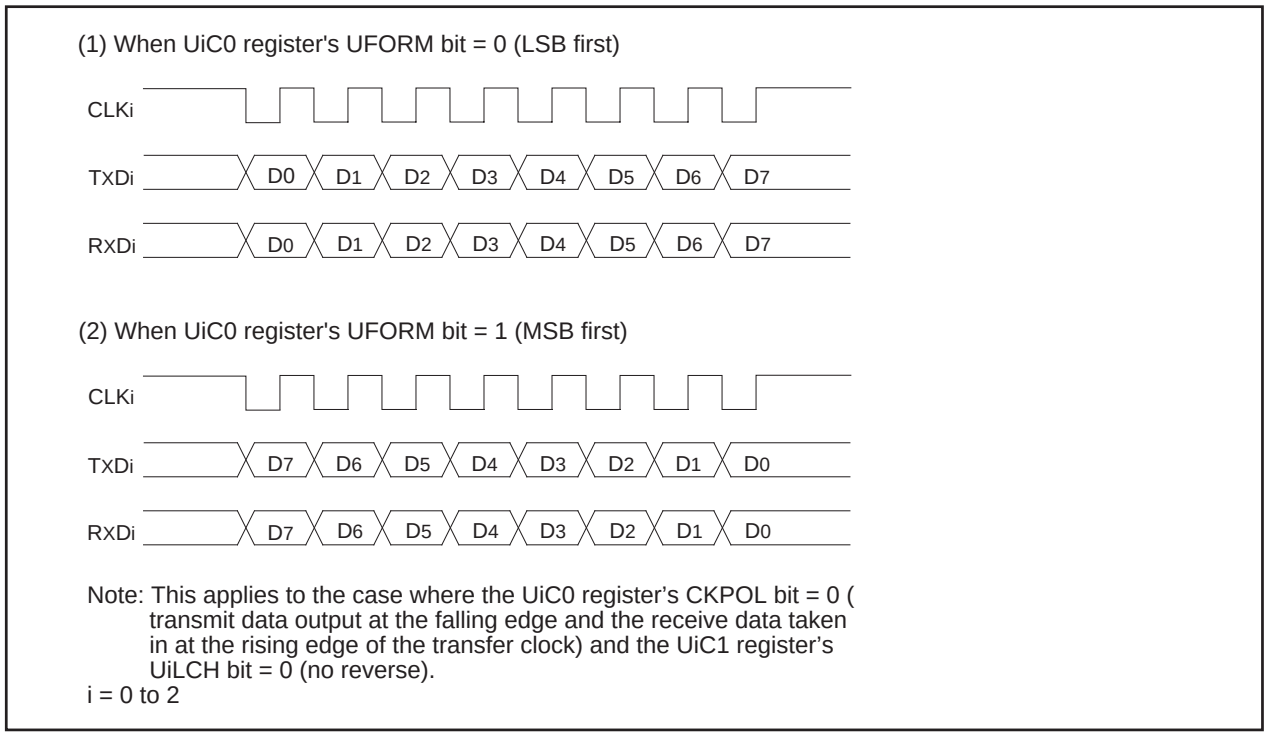


Figure 11.11. Transfer Format

(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 UiRRM bit (i = 0 to 2) = 1 (continuous receive mode), the UiC1 register's TI bit is set to "0" (data present in the UiTB register) by reading the UiRB register. In this case, i.e., UiRRM bit = 1, do not write dummy data to the UiTB register in a program. The U0RRM and U1RRM bits are the UCON register bit 2 and bit 3, respectively, and the U2RRM bit is the U2C1 register bit 4.

(d) Serial Data Logic Switching Function

When the UiC1 register (i = 0 to 2)'s UiLCH bit = 1 (reverse), the data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 11.12 shows serial data logic.

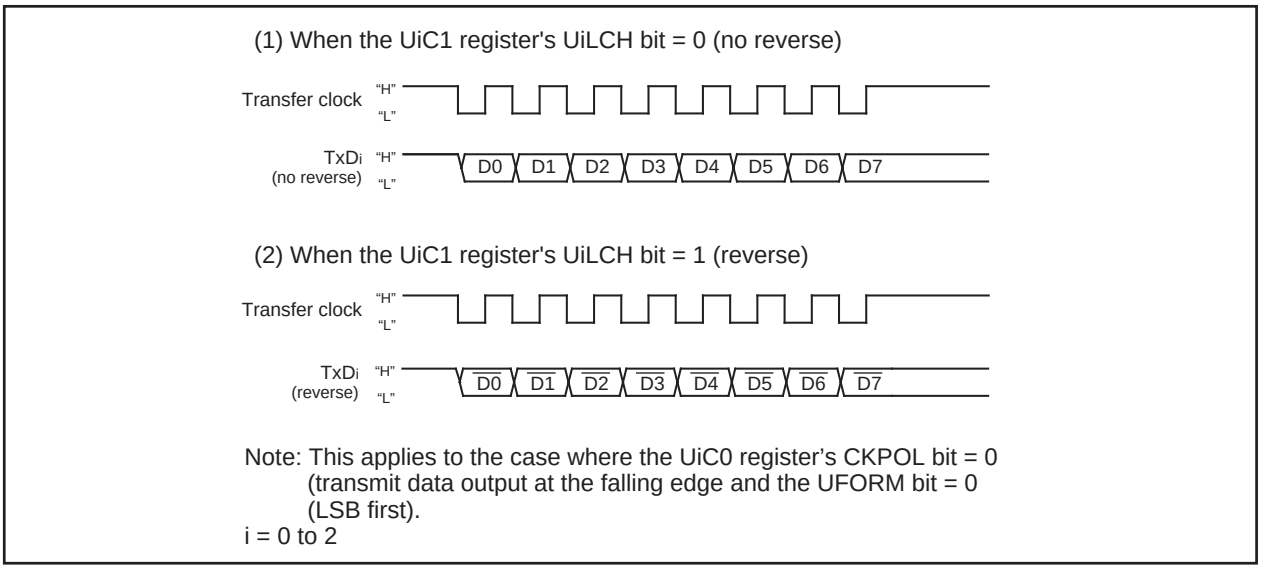


Figure 11.12. Serial Data Logic Switching

(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.

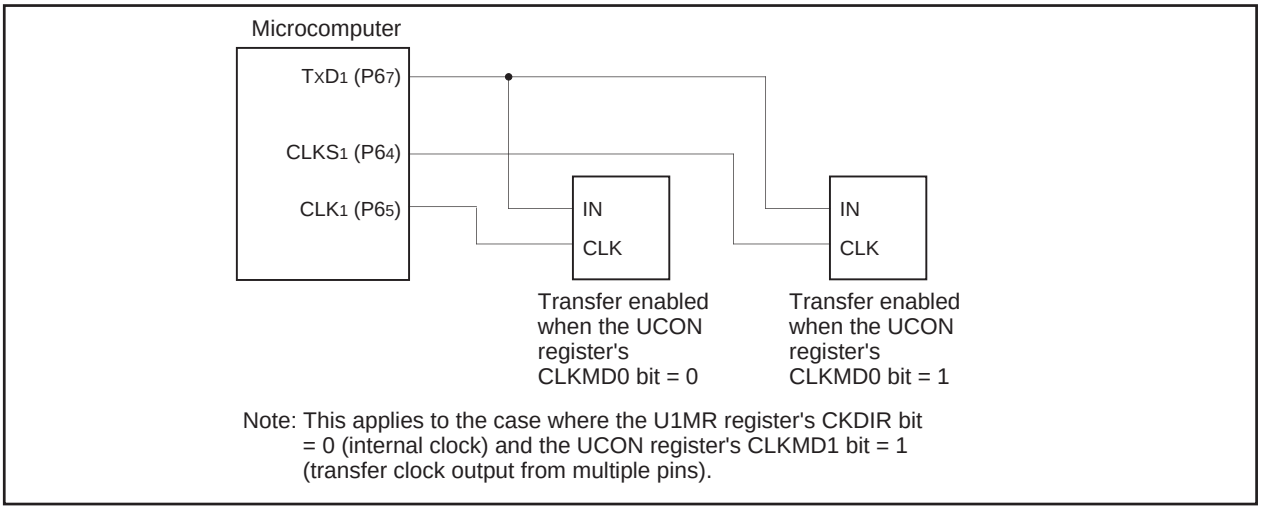


Figure 11.13. Transfer Clock Output From Multiple Pins

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

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

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

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

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

This function separates $\overline{CTS}_0/\overline{RTS}_0$, outputs $\overline{RTS}_0$ from the $P6_0$ pin, and accepts as input the $\overline{CTS}_0$ from the $P6_4$ pin. To use this function, set the register bits as shown below.

- U0C0 register's CRD bit = 0 (enables UART0 $\overline{CTS}/\overline{RTS}$)
- U0C0 register's CRS bit = 1 (outputs UART0 $\overline{RTS}$)
- U1C0 register's CRD bit = 0 (enables UART1 $\overline{CTS}/\overline{RTS}$)
- U1C0 register's CRS bit = 0 (inputs UART1 $\overline{CTS}$)
- UCON register's RCSP bit = 1 (inputs $\overline{CTS}_0$ from the $P6_4$ pin)
- UCON register's CLKMD1 bit = 0 ($CLKS_1$ not used)

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

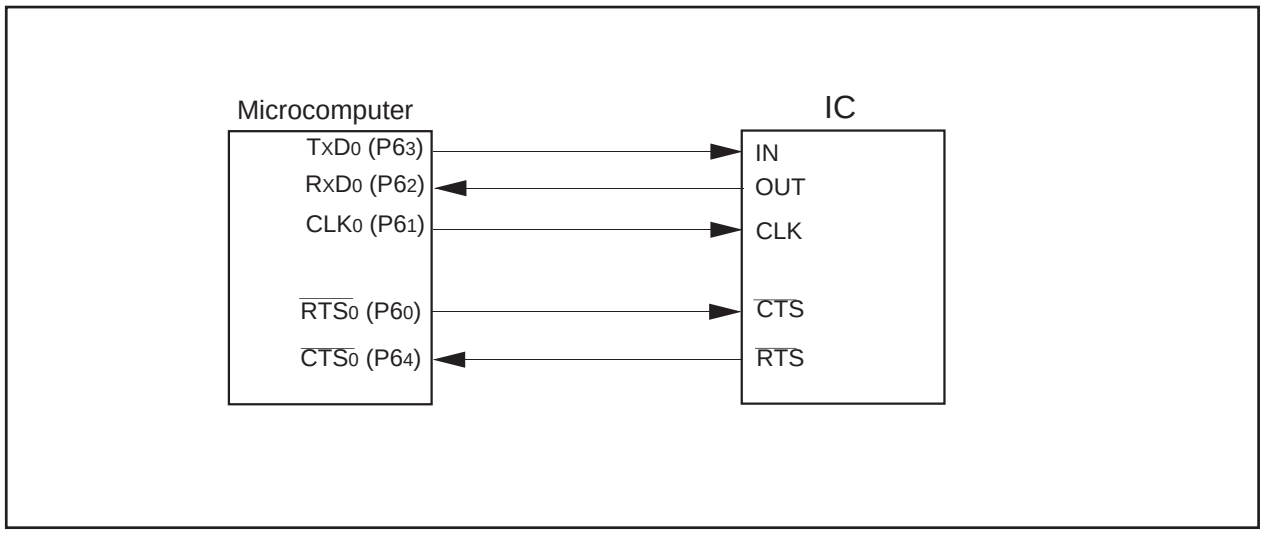


Figure 11.14. $\overline{CTS}/\overline{RTS}$ Separate Function

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	● Character bit (transfer data): Selectable from 7, 8 or 9 bits ● Start bit: 1 bit ● Parity bit: Selectable from odd, even, or none ● Stop bit: Selectable from 1 or 2 bits
Transfer clock	● UiMR(i=0 to 2) register's CKDIR bit = 0 (internal clock) : fj/ 16(n+1) fj = f1SIO, f2SIO, f8SIO, f32SIO. n: Setting value of UiBRG register 00_{16} to FF_{16} ● CKDIR bit = "1" (external clock) : fEXT/16(n+1) fEXT: Input from CLKi pin. n :Setting value of UiBRG register 00_{16} to FF_{16}
Transmission, reception control	● Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disable
Transmission start condition	● Before transmission can start, the following requirements must be met • The TE bit of UiC1 register= 1 (transmission enabled) • The TI bit of UiC1 register = 0 (data present in UiTB register) • If $\overline{CTS}$ function is selected, input on the $\overline{CTS}$i pin = "L"
Reception start condition	● Before reception can start, the following requirements must be met • The RE bit of UiC1 register= 1 (reception enabled) • Start bit detection
Interrupt request generation timing	●For transmission, one of the following conditions can be selected • The UiIRS bit (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 When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	● 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	● LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected ● Serial data logic switch This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. ● TxD, RxD I/O polarity switch This function reverses the polarities of hte TxD pin output and RxD pin input. The logic levels of all I/O data is reversed. ● Separate $\overline{CTS}/\overline{RTS}$ pins (UART0) $\overline{CTS_0}$ and $\overline{RTS_0}$ 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 '100$_2$' when transfer data is 7 bits long Set these bits to '101$_2$' when transfer data is 8 bits long Set these bits to '110$_2$' when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	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 $\overline{CTS}$ or $\overline{RTS}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ 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 $\overline{CTS_0}$ signal from the P6$_4$ 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 $P6_4$ 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

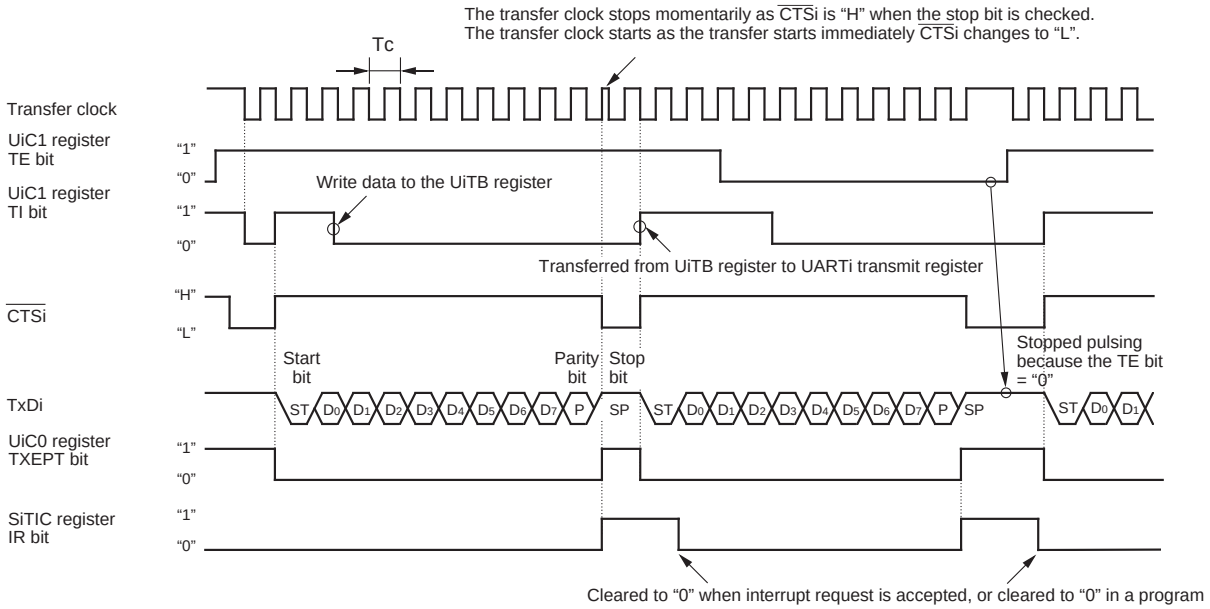
Pin name	Function	Method of selection
TxDi (i = 0 to 2) ($P6_3$, $P6_7$, $P7_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi ($P6_2$, $P6_6$, $P7_1$)	Serial data input	PD6 register's PD6_2 bit=0, PD6_6 bit=0, PD7 register's PD7_1 bit=0 (Can be used as an input port when performing transmission only)
CLKi ($P6_1$, $P6_5$, $P7_2$)	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
$\overline{CTS}$i/$\overline{RTS}$i ($P6_0$, $P6_4$, $P7_3$)	$\overline{CTS}$ 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
	$\overline{RTS}$ 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. $P6_4$ Pin Functions

Pin function	Bit set value				
	U1C0 register		UCON register		PD6 register
	CRD	CRS	RCSP	CLKMD1	PD6_4
$P6_4$	1	—	0	0	Input: 0, Output: 1
$\overline{CTS}_1$	0	0	0	0	0
$\overline{RTS}_1$	0	1	0	0	—
$\overline{CTS}_0$ (Note)	0	0	1	0	0

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

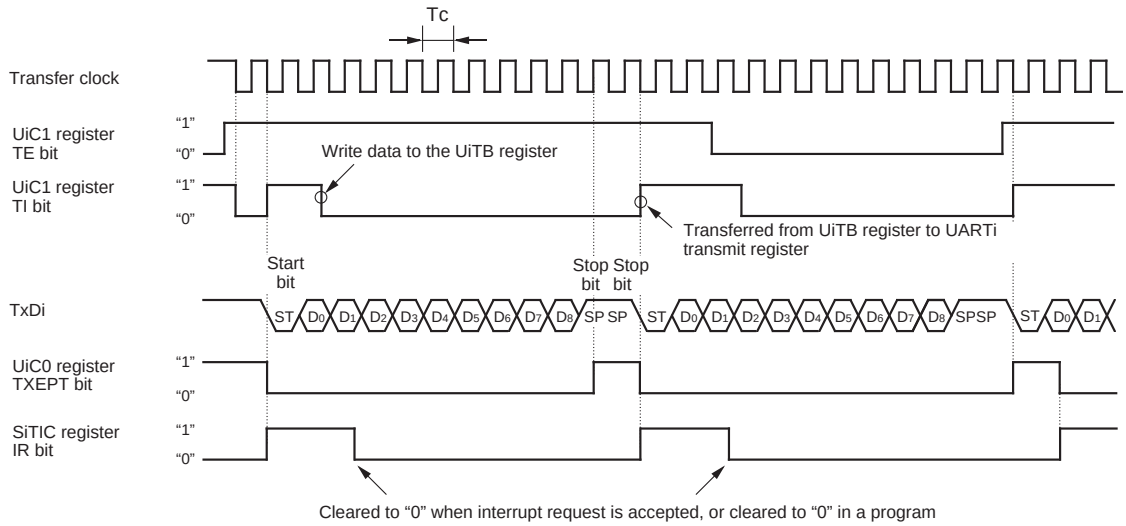
(1) Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)



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

Tc = 16 (n + 1) / fj or 16 (n + 1) / f_{EXT}
fj : frequency of UiBRG count source (f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO})
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)

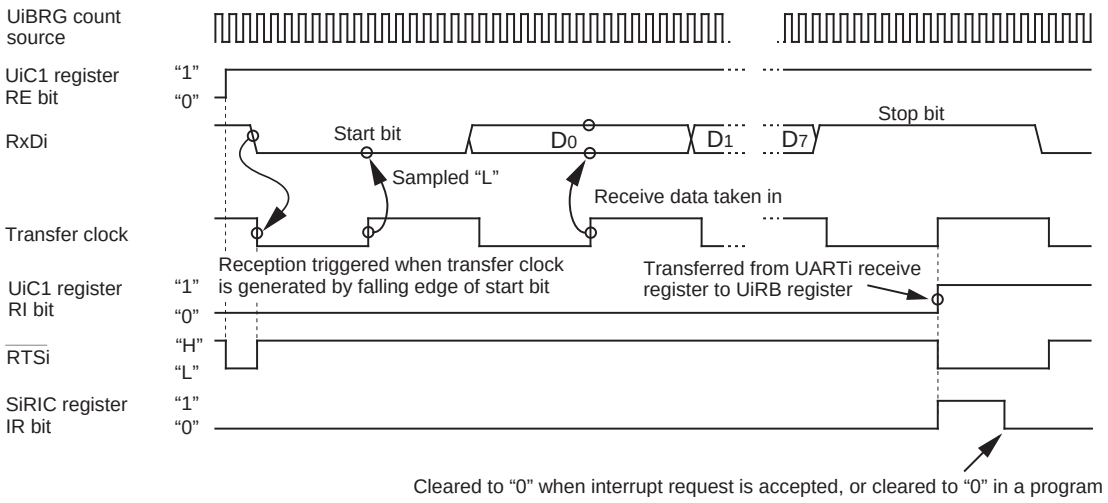


The above timing diagram applies to the case where the register bits are set as follows:
- UiMR register PRYE bit = 0 (parity disabled)
- UiMR register STPS bit = 1 (2 stop bits)
- UiC0 register CRD bit = 1 (CTS/RTS disabled)
- UiIRS bit = 0 (an interrupt request occurs when transmit buffer becomes empty):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

Tc = 16 (n + 1) / fj or 16 (n + 1) / f_{EXT}
fj : frequency of UiBRG count source (f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO})
f_{EXT} : frequency of UiBRG count source (external clock)
n : value set to UiBRG
i: 0 to 2

Figure 11.15. Transmit Operation

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



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

i = 0 to 2

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)

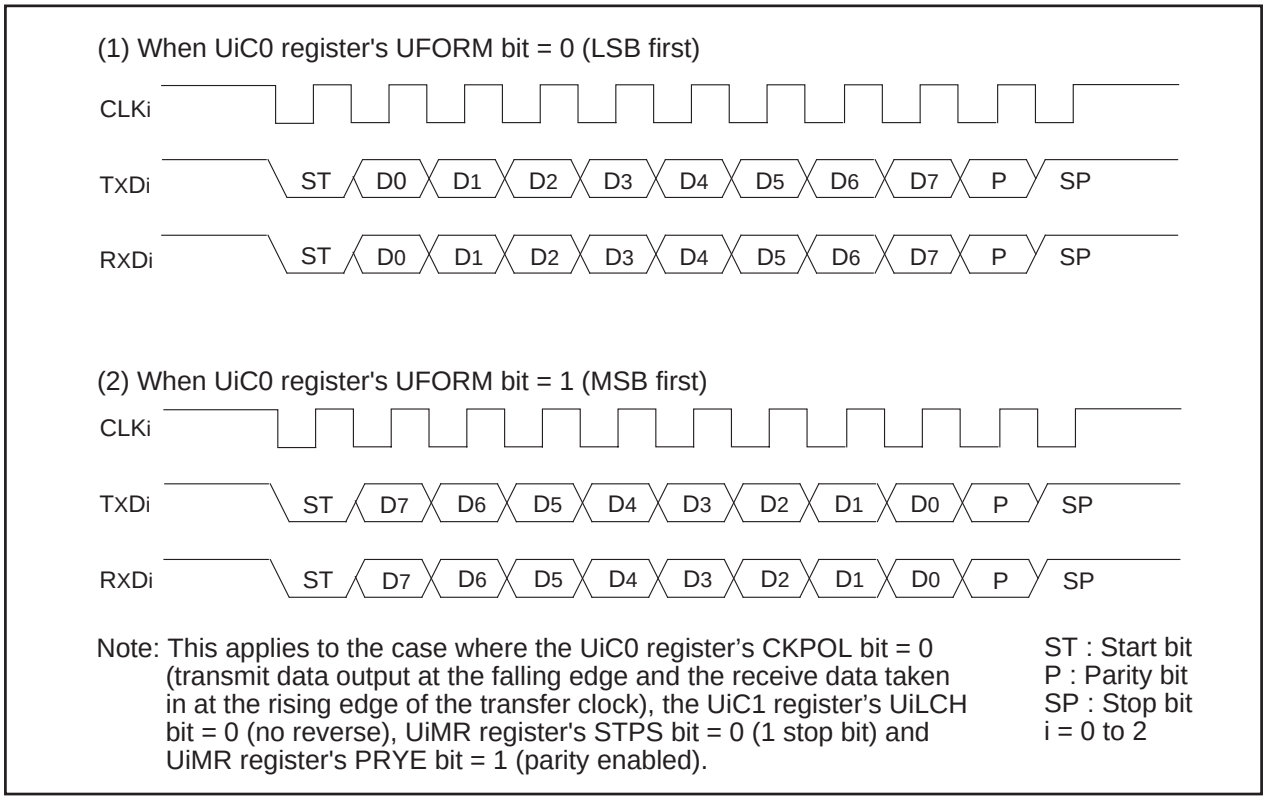
(1) Set the SMD2 to SMD0 bits in the UiMR register “000b” (Serial I/O disabled)

(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.



Note: This applies to the case where 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 UiC1 register’s UiLCH bit = 0 (no reverse), UiMR register's STPS bit = 0 (1 stop bit) and UiMR register's PRYE bit = 1 (parity enabled).

ST : Start bit
P : Parity bit
SP : Stop bit
i = 0 to 2

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.

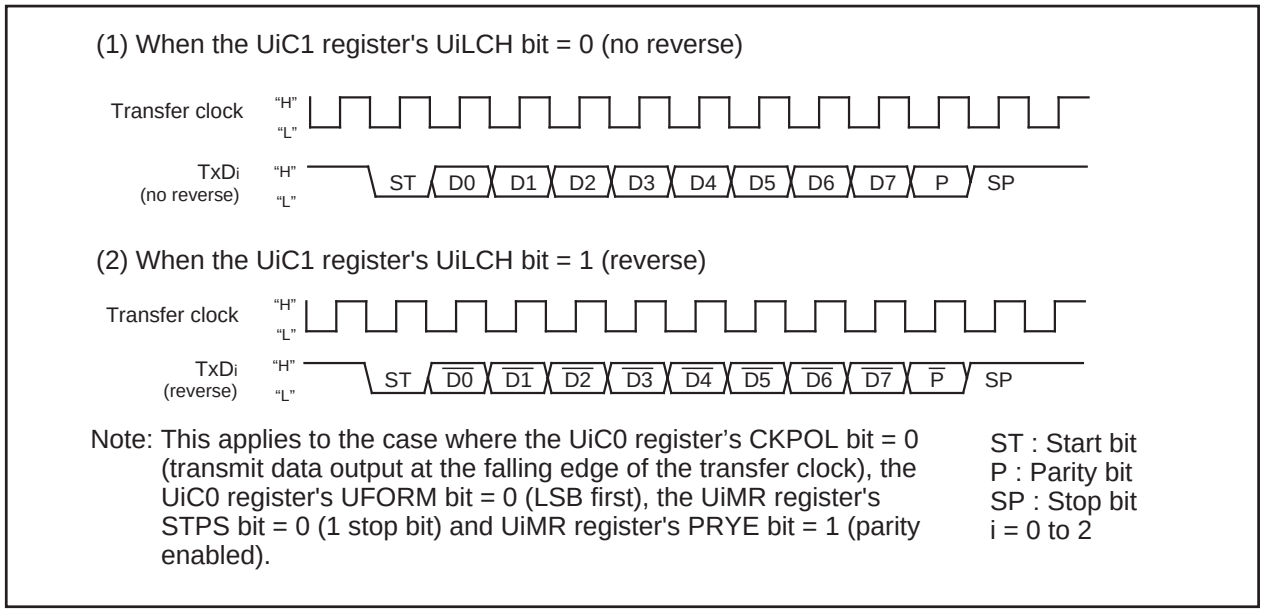


Figure 11.18. Serial Data Logic Switching

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

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

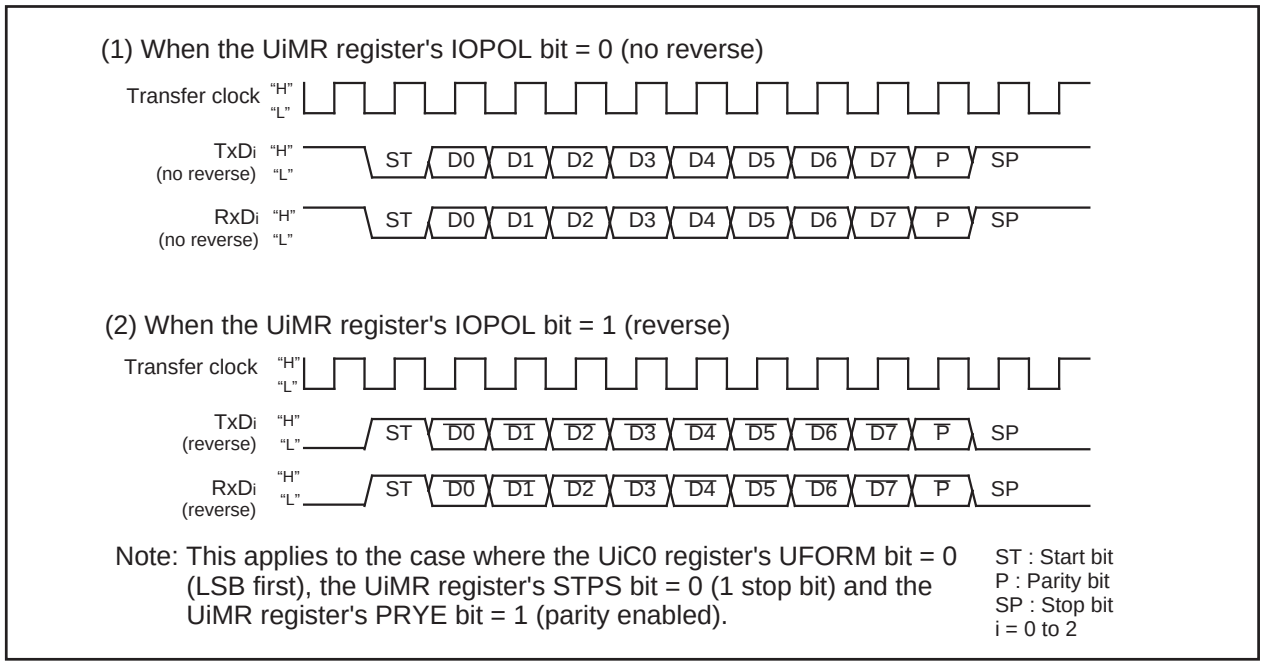


Figure 11.19. TxD and RxD I/O Polarity Inverse

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

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

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

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

(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 $P6_0$ pin, and accepts as input the $\overline{\text{CTS}}_0$ from the $P6_4$ 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 $P6_4$ pin)
- UCON register's CLKMD1 bit = 0 ($CLKS_1$ 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.

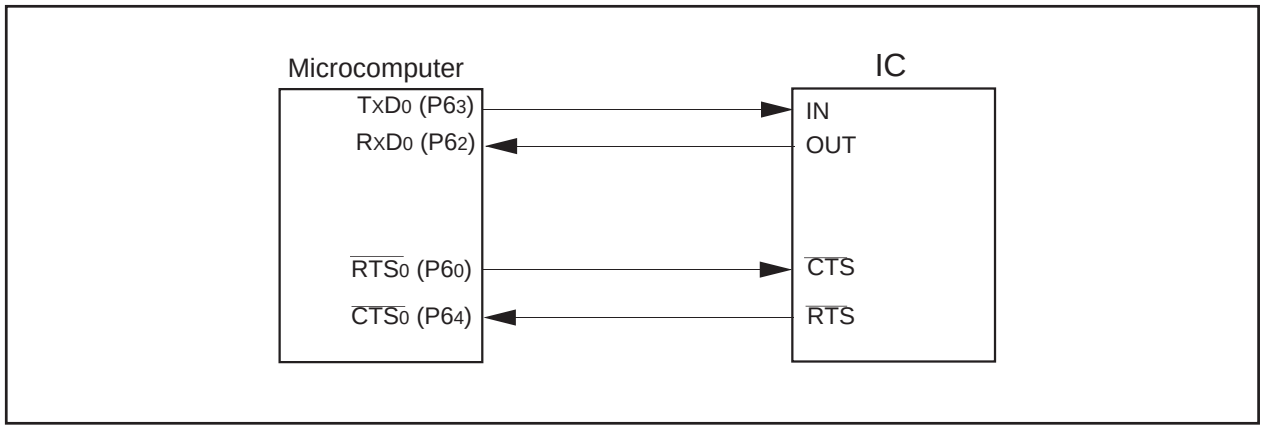


Figure 11.20. $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

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	● Transfer data length: 8 bits
Transfer clock	● Master mode UiMR(i=0 to 2) register's CKDIR bit = "0" (internal clock) : fj/ 2(n+1) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} ● Slave mode CKDIR bit = "1" (external clock selected) : Input from CLKi pin
Transmit/receive control	Controlled by input/output ports
Transmission start condition	● Before transmission can start, the following requirements must be met (Note 1) • The TE bit of UiC1 register= 1 (transmission enabled) • The TI bit of UiC1 register = 0 (data present in UiTB register)
Reception start condition	● Before reception can start, the following requirements must be met (Note 1) • The RE bit of UiC1 register= 1 (reception enabled) • The TE bit of UiC1 register= 1 (transmission enabled) • The TI bit of UiC1 register= 0 (data present in the UiTB register)
Interrupt request generation timing	● For transmission, one of the following conditions can be selected • 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	● 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	● 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.

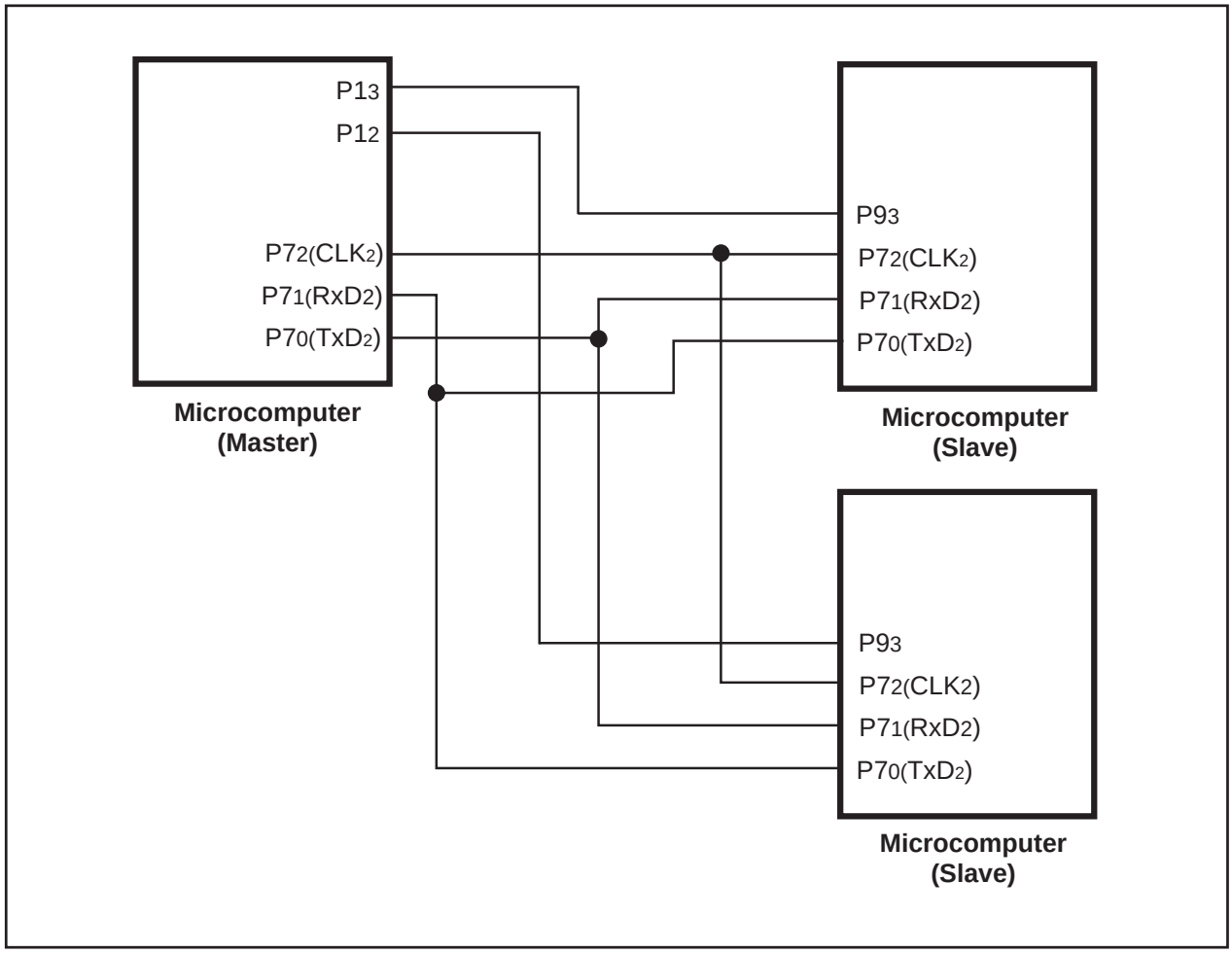


Figure 11.21. Serial Bus Communication Control Example (UART2)

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	Overrun error flag
UiBRG	0 to 7	Set a transfer rate
UiMR(Note3)	SMD2 to SMD0	Set to '001$_2$'
	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 salves 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).

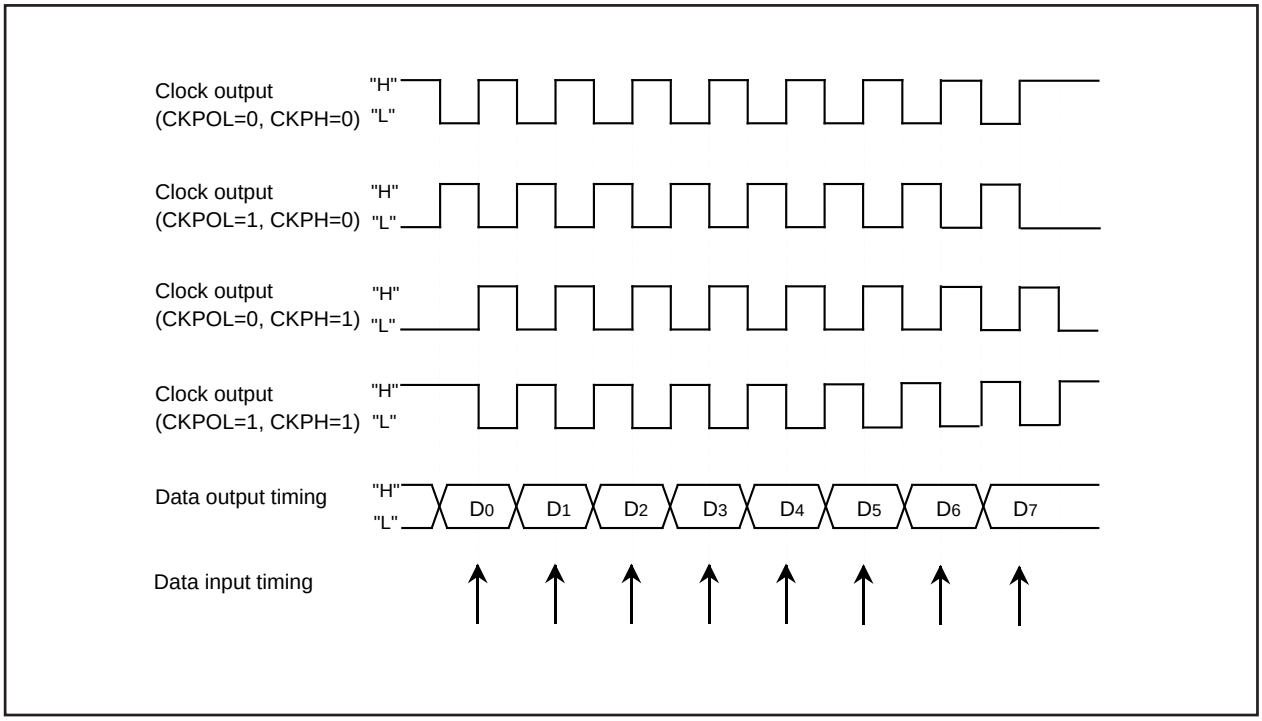


Figure 11.22. Transmission and Reception Timing in Master Mode (Internal Clock)

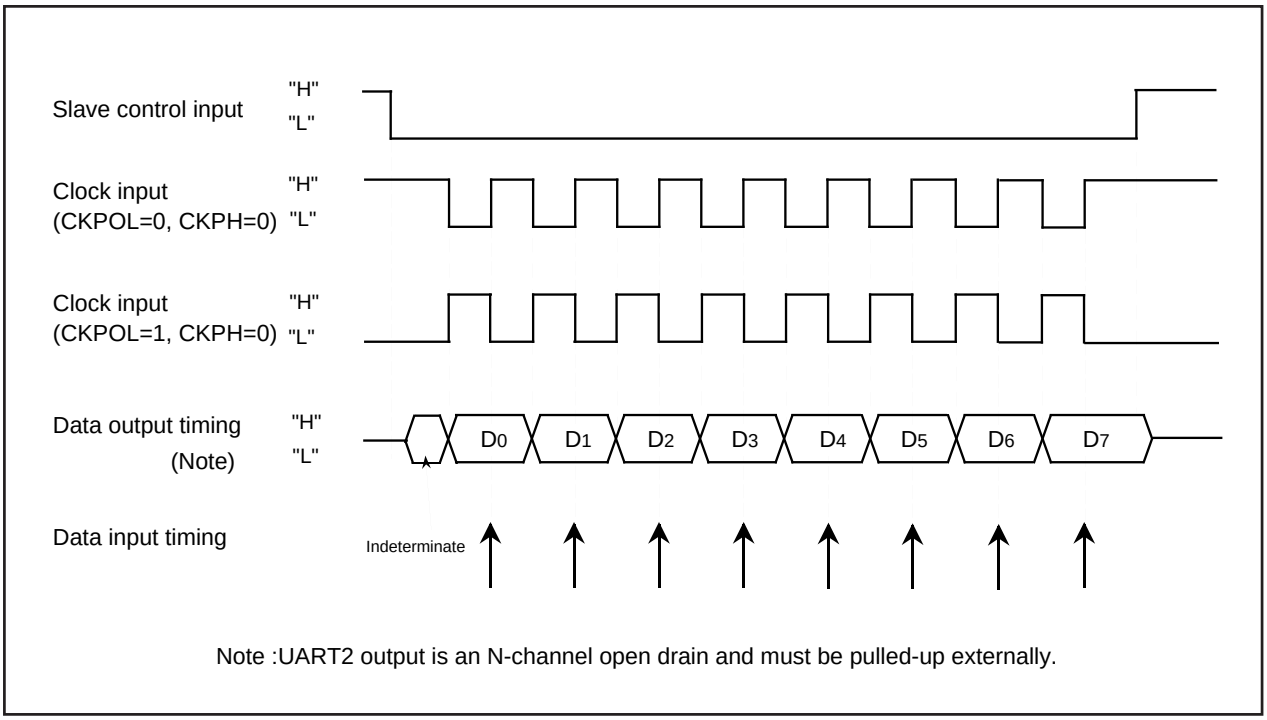


Figure 11.23. Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)

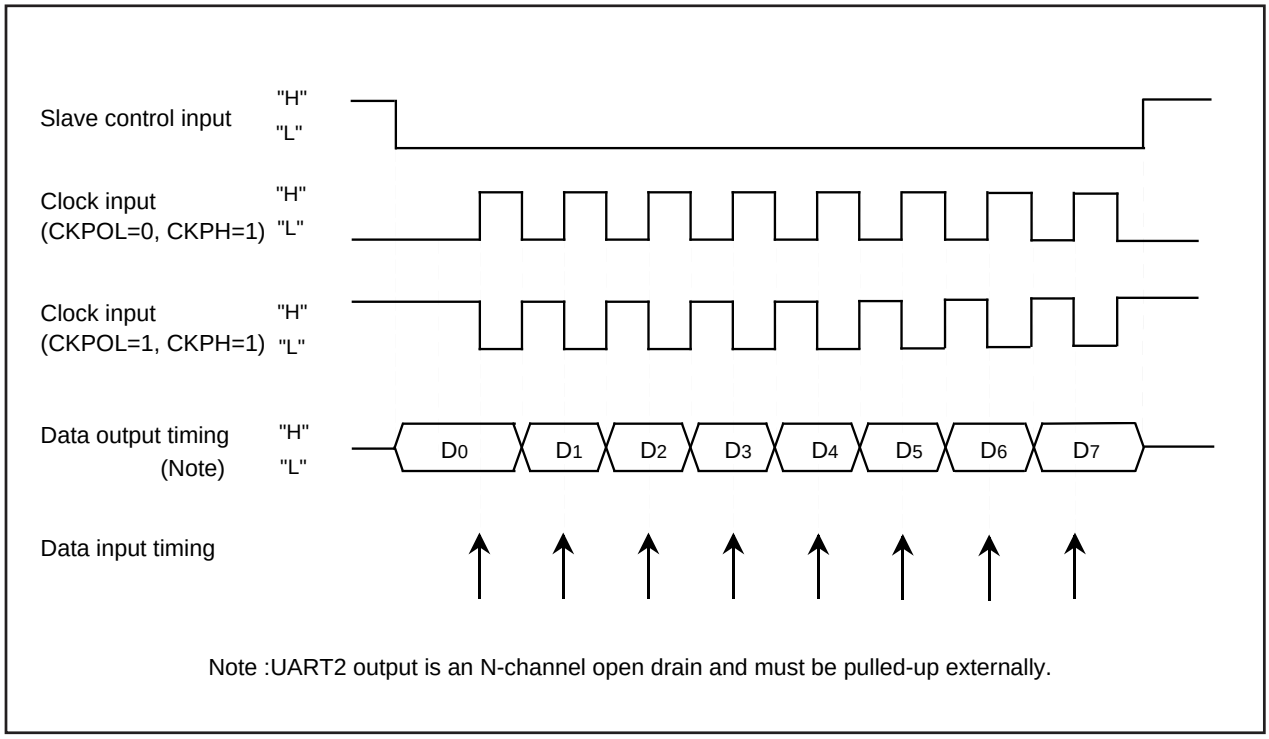


Figure 11.24. Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)

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 UARTi bus collision detect interrupt request is generated.

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

Table 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 '110$_2$'
	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"

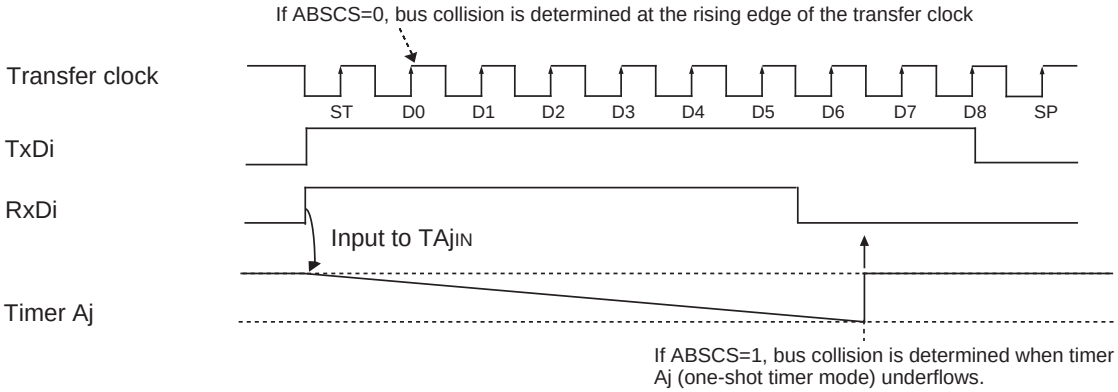
Notes 1: Set the U0C0 and U1C1 registers bit 4 and bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.

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.

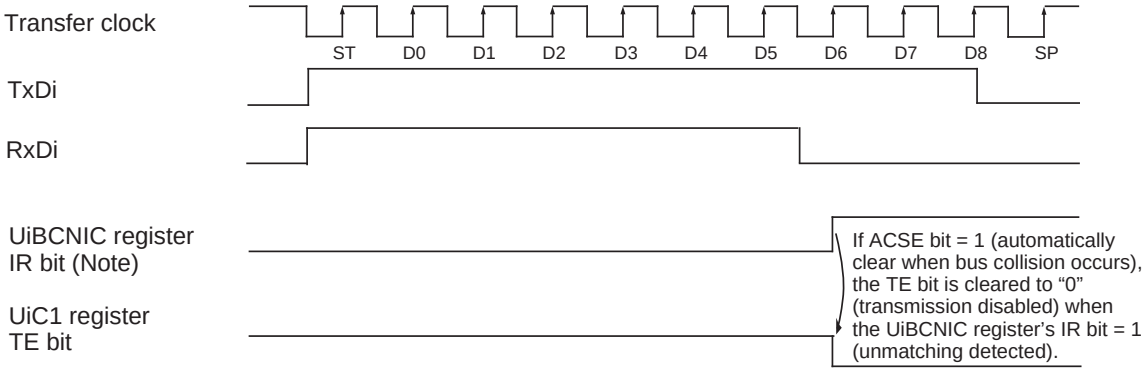
i= 0 to 2

(1) UiSMR register ABSCS bit (bus collision detect sampling clock select) (i=0 to 2)



Timer Aj: timer A3 when UART0; timer A4 when UART1; timer A0 when UART2

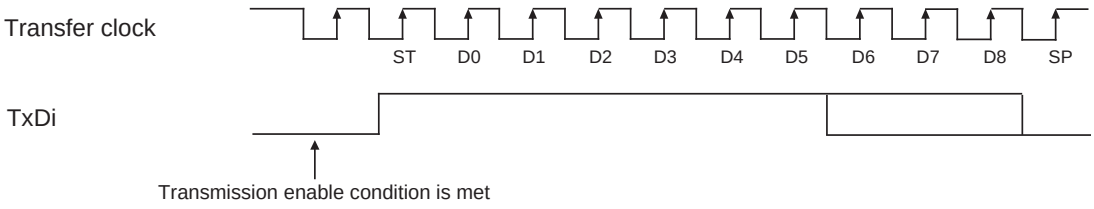
(2) UiSMR register ACSE bit (auto clear of transmit enable bit)



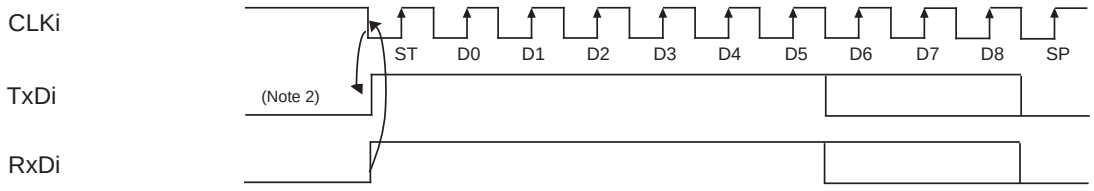
Note: BCNIC register when UART2.

(3) UiSMR register SSS bit (Transmit start condition select)

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



If SSS bit = 1, the serial I/O starts sending data at the rising edge (Note 1) of RxDi



Note 1: The falling edge of RxDi when IOPOL=0; the rising edge of RxDi when IOPOL =1.
Note 2: The transmit condition must be met before the falling edge (Note 1) of RxD.

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

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 P10$_3$ to P10$_7$, P0$_4$ to P0$_7$, and P2$_4$ to P2$_7$.

When not using the A/D converter, set the VCUT bit to “0” (= Vref unconnected), so that no current will flow from the V_{REF} 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).

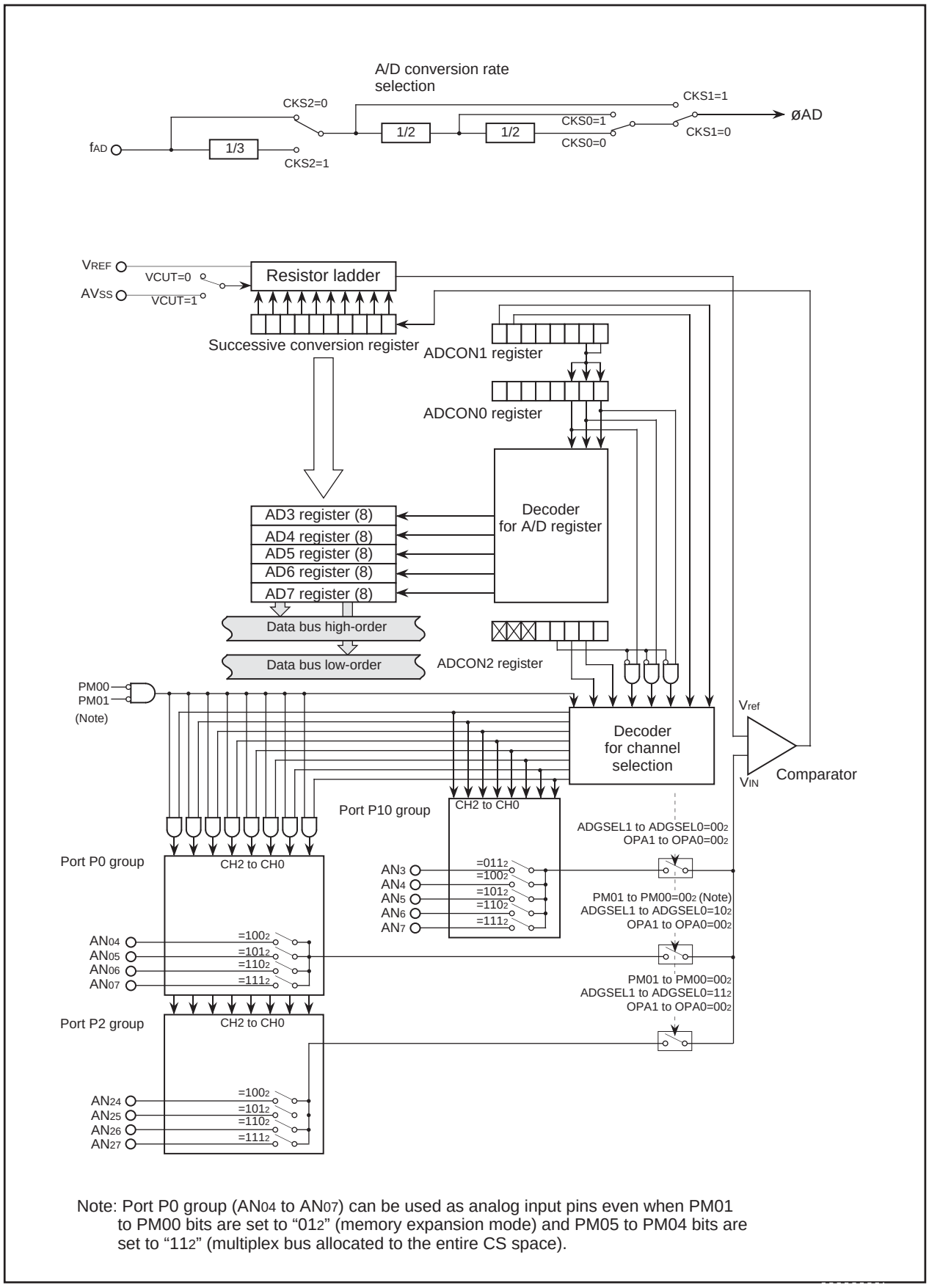
Table 12.1 shows the performance of the A/D converter. Figure 12.1 shows the block diagram of the A/D converter, and Figures 12.2 and 12.3 show the A/D converter-related registers.

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 V_{CC1}
Operating clock ϕ_{AD} (Note 2)	f_{AD}/divide-by-2 of f_{AD}/divide-by-3 of f_{AD}/divide-by-4 of f_{AD}/divide-by-6 of f_{AD}/divide-by-12 of f_{AD}
Resolution	8-bit
Integral nonlinearity error	±5LSB
Operating modes	One-shot mode, repeat mode, single sweep mode and repeat sweep mode 0
Analog input pins	5 pins (AN$_3$ to AN$_7$) + 4 pins (AN$_{04}$ to AN$_{07}$) + 4 pins (AN$_{24}$ to AN$_{27}$)
A/D conversion start condition	• Software trigger The ADCON0 register's ADST bit is set to “1” (A/D conversion starts)
Conversion speed per pin	• 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.



Note: Port P0 group (AN04 to AN07) can be used as analog input pins even when PM01 to PM00 bits are set to “012” (memory expansion mode) and PM05 to PM04 bits are set to “112” (multiplex bus allocated to the entire CS space).

Figure 12.1. A/D Converter Block Diagram

A/D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol: ADCON0 — Address: 03D6$_{16}$ — After reset: 00000XXX$_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	Function varies with each operation mode	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0	RW
MD1			RW
Reserved bit		Must always be set to “0”	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	See Note 2 for the ADCON2 register	RW

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

A/D control register 1 (Note 1)

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

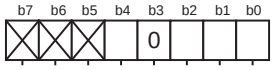
Symbol: ADCON1 — Address: 03D7$_{16}$ — After reset: 00$_{16}$

Bit symbol	Bit name	Function	RW
SCAN0	A/D sweep pin select bit	Function varies with each operation mode	RW
SCAN1			RW
Reserved bit		Must always be set to “0”	RW
Reserved bit		Must always be set to “0”	RW
CKS1	Frequency select bit 1	See Note 2 for the ADCON2 register	RW
VCUT	Vref connect bit (Note 2)	0 : Vref not connected 1 : Vref connected	RW
Reserved bit		Must always be set to “0”	RW

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

Figure 12.2. ADCON0 to ADCON1 Registers

A/D control register 2 (Note 1)



Symbol	Address	After reset
ADCON2	03D4₁₆	00₁₆

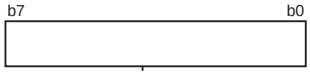
Bit symbol	Bit name	Function	RW
SMP	A/D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0 : Port P10 group is selected 0 1 : Must not be set 1 0 : Port P0 group is selected 1 1 : Port P2 group is selected	RW
ADGSEL1			RW
(b3)	Reserved bit	Must always be set to “0”	RW
CKS2	Frequency select bit 2 (Note 2)	0: Selects f_{AD}, f_{AD} divided by 2, or f_{AD} divided by 4. 1: Selects f_{AD} divided by 3, f_{AD} divided by 6, or f_{AD} divided by 12.	RW
(b7-b5)	Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.		—

Notes 1: If the ADCON2 register is rewritten during A/D conversion, the conversion result will be indeterminate
2: The $Ø_{AD}$ frequency must be 10 MHz or less. The selected $Ø_{AD}$ 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	$Ø_{AD}$
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	03C6₁₆	Indeterminate
AD4	03C8₁₆	Indeterminate
AD5	03CA₁₆	Indeterminate
AD6	03CC₁₆	Indeterminate
AD7	03CE₁₆	Indeterminate



Function	RW
A/D conversion result	RO

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	• End of A/D conversion • Writing “0” to ADST bit
Interrupt request generation timing	End of A/D conversion
Input pin	One of AN_3 to AN_7, AN_{04} to AN_{07}, AN_{24} to AN_{27}, as selected
Reading of result of A/D converter	Read AD3 to AD7 registers corresponding to selected pin

A/D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0			

Symbol: ADCON0 — Address: $03D6_{16}$ — When reset: $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : Do not set 0 0 1 : Do not set 0 1 0 : Do not set 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected (Note 2) 1 1 1 : AN_7 is selected (Note 3)	○	○
CH1			○	○
CH2			○	○
MD0	A/D operation mode select bit 0	b4 b3 0 0 : One-shot mode (Note 3)	○	○
MD1			○	○
Reserved bit		Must always be set to “0”	○	○
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	○	○
CKS0	Frequency select bit 0	1 : Refer to note 3 of ADCON2 register	○	○

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

2: AN_{04} to AN_{07} and AN_{24} to AN_{27} can be used like AN_4 to AN_7. Please choose by bits ADGSEL1 to ADGSEL0 of ADCON2 register.

3: Please re-set up bits CH2 to CH0 by another command after rewriting bits MD1 to MD0.

A/D control register 1 (Note 1)

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

Symbol: ADCON1 — Address: $03D7_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A/D sweep pin select bit	Invalid in one-shot mode	○	○
SCAN1			○	○
MD2	A/D operation mode select bit 1	Set to “0” in one-shot mode	○	○
Reserved bit		Must always be set to “0”	○	○
CKS1	Frequency select bit1	1 : Refer to note 3 of ADCON2 register	○	○
VCUT	Vref connect bit (Note 2)	1 : Vref connected	○	○
Reserved bits		Must always be set to “0”	○	○

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

2: When VCUT bit is set to “1” (connection) from “0” (un-connecting), after 1 microsecond or more passes, please start A/D conversion.

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 AN_3 to AN_7 and AN_{04} to AN_{07} and AN_{24} to AN_{27}.
Read-out of A/D conversion value	Read out of registers AD3 to AD7 corresponding to the selected pin.

A/D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	1			

Symbol ADCON0 — Address 03D6$_{16}$ — When reset 00000XXX$_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : Do not set 0 0 1 : Do not set 0 1 0 : Do not set 0 1 1 : AN$_3$ is selected 1 0 0 : AN$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected (Note 2) 1 1 1 : AN$_7$ is selected (Note 3)	○	○
CH1			○	○
CH2			○	○
MD0	A/D operation mode select bit 0	b4 b3 0 1 : Repeat mode (Note 3)	○	○
MD1			○	○
Reserved bit		Must always be set to “0”	○	○
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	○	○
CKS0	Frequency select bit 0	1 : Refer to note 3 of ADCON2 register	○	○

Notes 1: If the A/D control register is rewritten during A/D conversion, the conversion result is indeterminate.
2: AN$_{04}$ to AN$_{07}$, and AN$_{24}$ to AN$_{27}$ can be used like AN$_4$ to AN$_7$.
Please choose by bits ADGSEL1 and ADGSEL2 of ADCON2 register.
3: Please re-set up CH2 to CH0 bits by another command after rewriting MD1 to MD0 bits.

A/D control register 1 (Note 1)

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

Symbol ADCON1 — Address 03D7$_{16}$ — When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A/D sweep pin select bit	Invalid in repeat mode	○	○
SCAN1			○	○
MD2	A/D operation mode select bit 1	Set to “0” in one-shot mode	○	○
Reserved bit		Must always be set to “0”	○	○
CKS1	Frequency select bit 1	1 : Refer to note 3 of ADCON2 register	○	○
VCUT	Vref connect bit (Note 2)	1 : Vref connected	○	○
Reserved bits		Must always be set to “0”	○	○

Notes 1: If the A/D control register is rewritten during A/D conversion, the conversion result is indeterminate.
2: When VCUT bit is set to “1” (connection) from “0” (un-connecting), after 1 microsecond or more passes, please start A/D conversion.

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 ADCON1register 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	● 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 ANi_4 to ANi_5 (two pins), and ANi_4 to ANi_7 (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: AN_4 to AN_7 can be used like AN_{04} to AN_{07}, and AN_{24} to AN_{27}. In this case, it becomes selection from AN_4 to AN_5 (two pins), and AN_4 to AN_7 (four pins).

A/D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0			

Symbol	Address	After reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	Invalid in single sweep mode	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 1 0 : Single sweep mode	RW
MD1			RW
Reserved bit		Must always be set to "0"	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	See Note 3 for the ADCON2 register	RW

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

A/D control register 1 (Note 1)

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

Symbol	Address	After reset
ADCON1	$03D7_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
SCAN0	A/D sweep pin select bit	When single sweep mode is selected b1 b0 0 0 : Must not be set 0 1 : Must not be set 1 0 : ANi_4 to ANi_5 (2 pins) (i=0, 2) 1 1 : ANi_4 to ANi_7 (4 pins) (Note 2)	RW
SCAN1			RW
Reserved bit		Must always be set to "0"	RW
Reserved bit		Must always be set to "0"	RW
CKS1	Frequency select bit 1	See Note 3 for the ADCON2 register	RW
VCUT	Vref connect bit (Note 3)	1 : Vref connected	RW
Reserved bits		Must always be set to "0"	RW

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

2: AN_4 to AN_7 can be used like AN_{04} to AN_{07}, and AN_{24} to AN_{27}. In this case, it becomes selection from AN_4 to AN_5 (two pins), and AN_4 to AN_7 (four pins).

0 0 : Must not be set
0 1 : Must not be set
1 0 : AN_4 to AN_5 (2 pins)
1 1 : AN_4 to AN_7 (4 pins)

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

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).

A/D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	1			

Symbol: ADCON0 — Address: $03D6_{16}$ — After reset: $00000XXX_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
Reserved bit		Must always be set to "0"	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	See Note 3 for the ADCON2 register	RW

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

A/D control register 1 (Note 1)

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

Symbol: ADCON1 — Address: $03D7_{16}$ — After reset: 00_{16}

Bit symbol	Bit name	Function	RW
SCAN0	A/D sweep pin select bit	When repeat sweep mode 0 is selected b1 b0	RW
SCAN1		0 0 : Must not be set 0 1 : Must not be set 1 0 : ANi_4 to ANi_5 (2 pins) (i=0, 2) 1 1 : ANi_4 to ANi_7 (4 pins) (Note 2)	RW
Reserved bit		Must always be set to "0"	RW
Reserved bit		Must always be set to "0"	RW
CKS1	Frequency select bit 1	See Note 3 for the ADCON2 register	RW
VCUT	Vref connect bit (Note 3)	1 : Vref connected	RW
Reserved bits		Must always be set to "0"	RW

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

2: AN_4 to AN_7 can be used like AN_{04} to AN_{07}, and AN_{24} to AN_{27}. In this case, it becomes selection from AN_4 to AN_5 (two pins), and AN_4 to AN_7 (four pins).

0 0 : Must not be set
0 1 : Must not be set
1 0 : AN_4 to AN_5 (2 pins)
1 1 : AN_4 to AN_7 (4 pins)

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

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 ϕAD 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 (V_{REF}) can be separated using the ADCON1 register’s VCUT bit. When separated, no current will flow from the V_{REF} 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” (V_{REF} 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” (V_{REF} 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 AN_4 to AN_7 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 V_{CC1} pin, V_{CC2} pin, an analog input pin (ANi (i=3 to 7), AN0i, AN2i), 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 ADi register when CPU reads an ADi register to the timing which stores the result in an ADi 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.

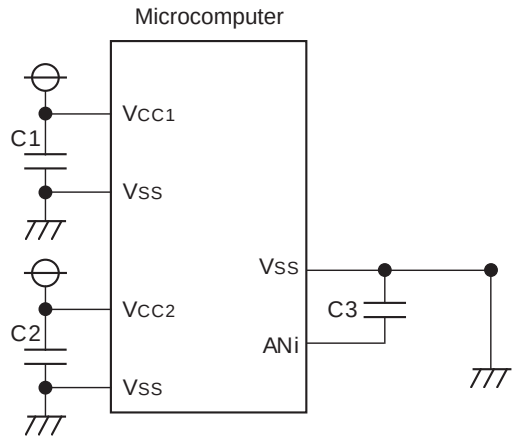
● **When using it in one-shot mode or single sweep mode**

Please read the target ADi 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 ADi 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 ADi registers.



ANi: ANi (i=3 to 7), AN0i , AN2i

Note 1: C1≥0.1μF, C2≥0.1μF, C3≥100pF (reference value).
Note 2: Please connect a capacitor in the shortest distance using thick wiring.

Figure 12.8. Example of noise measure processing of each pin

Multi-master I^2C-BUS Interface 0 to 2

The multi-master I^2C-BUS interface i (i=0 to 2) have each dedicated circuit and operate independently. The multi-master I^2C-BUS interface i is a serial communications circuit, conforming to the Philips I^2C-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 I^2C-BUS interface i and Table 13.1 shows multi-master I^2C-BUS interface i functions.

This multi-master I^2C-BUS interface i consists of the I^2Ci address register, the I^2Ci data shift register, the I^2Ci clock control register, the I^2Ci control register, the I^2Ci status register, the I^2Ci port selection register and other control circuits.

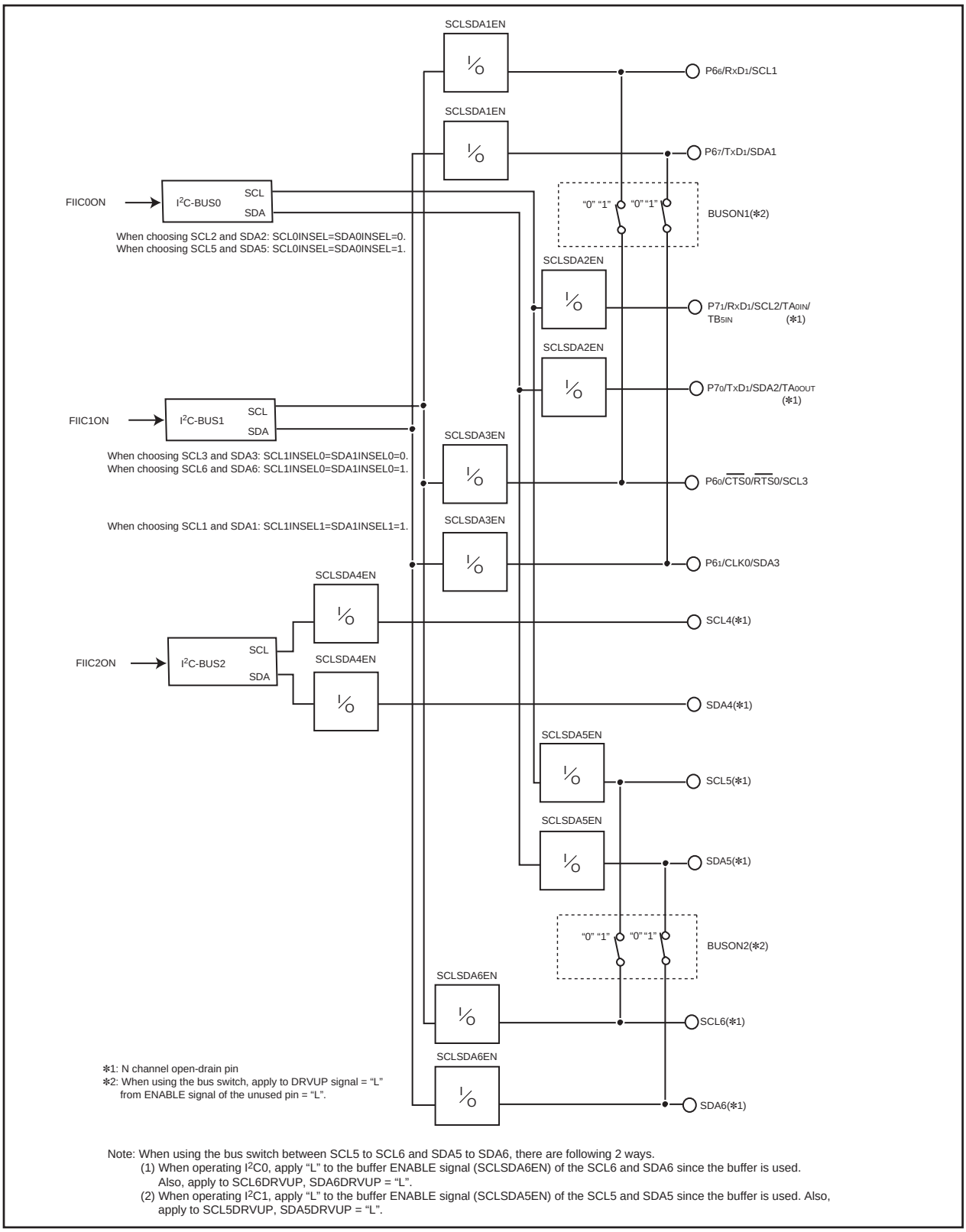
Table 13.1 Multi-master I^2C-BUS Interface Functions

Item	Function
Format	In conformity with Philips I^2C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	In conformity with Philips I^2C-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 I^2C control register at address $02D9_{16}$) for connections between the I^2C-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.



Note: When using the bus switch between SCL5 to SCL6 and SDA5 to SDA6, there are following 2 ways.
(1) When operating I²C0, apply "L" to the buffer ENABLE signal (SCLSDA6EN) of the SCL6 and SDA6 since the buffer is used. Also, apply to SCL6DRVUP, SDA6DRVUP = "L".
(2) When operating I²C1, apply "L" to the buffer ENABLE signal (SCLSDA5EN) of the SCL5 and SDA5 since the buffer is used. Also, apply to SCL5DRVUP, SDA5DRVUP = "L".

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

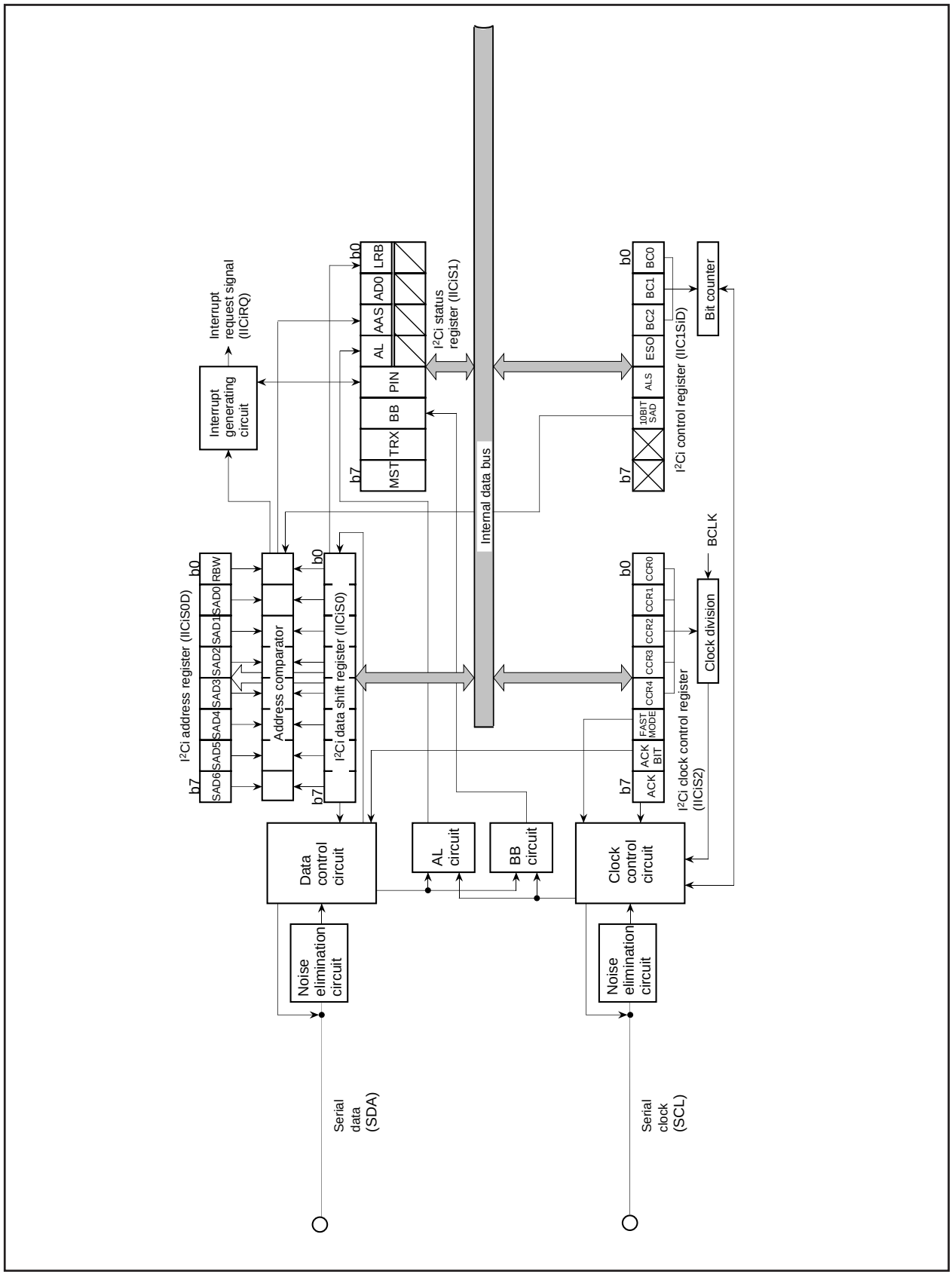


Fig. 13.2 Block Diagram of Multi-master I²C-BUS Interface i (i = 0 to 2)

(1) Reserved register

Reserved register

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

Symbol	Address	When reset
RSVREG02E5 RSVREG02ED RSVREG02F5	$02E5_{16}$, $02ED_{16}$, $02F5_{16}$	$00?00000_2$

Bit Symbol	Bit name	Function	RW
Reserved bits		Must always be set to “0”	WO
RSVREG02E52 RSVREG02ED2 RSVREG02F52	Multi-master I2C-BUS i (i=0 to 2) interface enable bit $02E5_{16}$ = i = 0 $02ED_{16}$ = i = 1 $02F5_{16}$ = i =2	0 = Non active 1 = Active	RW
Reserved bits		Must always be set to “0”	WO

Fig. 13.3 Reserved register

(2) I^2Ci data shift register, I^2Ci transmit buffer register (i = 0 to 2)

The I^2Ci 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.

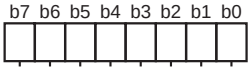
The I^2Ci data shift register is in a write enable status only when the ESO bit of the I^2Ci control register is "1." The bit counter is reset by a write instruction to the I^2Ci data shift register. When both the ESO bit and the MST bit of the I^2Ci status register are "1," the SCL is output by a write instruction to the I^2Ci data shift register. Reading data from the I^2Ci data shift register is always enabled regardless of the ESO bit value.

The I^2Ci transmit buffer register is a register to store transmit data (slave address) to the I^2Ci data shift register before RESTART condition generation. That is, in master, transmit data written to the I^2Ci transmit buffer register is written to the I^2Ci data shift register simultaneously. However, the SCL is not output. The I^2Ci transmit buffer register can be written only when the ESO bit is "1," reading data from the I^2Ci transmit buffer register is disabled regardless of the ESO bit value.

Notes 1: To write data into the I^2Ci data shift register or the I^2Ci 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 I^2Ci data shift register or the I^2Ci transmit buffer register is written, keep an interval of 4 BCLK or more.

I²Ci data shift register (i = 0 to 2)



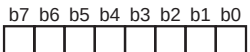
Symbol	Address	When reset
IIC0S0	02E0₁₆	Indeterminate
IIC1S0	02E8₁₆	Indeterminate
IIC2S0	02F0₁₆	Indeterminate

Bit Symbol	Bit name	Function	RW
D0	Data shift register	This is an 8-bit shift register to store receive data and write transmit data.	RW
D1			
D2			
D3			
D4			
D5			
D6			
D7			

Note: To write data into the I²Ci data shift register after setting the MST bit to "0" (slave mode), keep an interval of 8 machine cycles or more.

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

I²Ci transmit buffer register (i = 0 to 2)



Symbol	Address	When reset
IIC0S0S	02E6₁₆	Indeterminate
IIC1S0S	02EE₁₆	Indeterminate
IIC2S0S	02F6₁₆	Indeterminate

Bit Symbol	Bit name	Function	RW
S0S0	Transmit buffer register	This is an 8-bit register to write transmit data to I²Ci data shift register.	WO
S0S1			
S0S2			
S0S3			
S0S4			
S0S5			
S0S6			
S0S7			

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)**

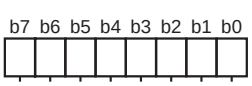
Not used when comparing addresses, in the 7-bit addressing mode. In the 10-bit addressing mode, the first address data to be received is compared with the contents (SAD6 to SAD0 + RBW) of the I²Ci address register.

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.

I²Ci address register (i = 0 to 2)



Symbol	Address	When reset
IIC0S0D	$02E1_{16}$	00_{16}
IIC1S0D	$02E9_{16}$	00_{16}
IIC2S0D	$02F1_{16}$	Indeterminate

Bit Symbol	Bit name	Function	RW
RBW	$\overline{\text{Read}}$/write bit	<Only in 10-bit addressing (in slave) mode> The last significant bit of address data is compared. 0 : Wait the first byte of slave address after START condition (read state) 1 : Wait the first byte of slave address after RESTART condition (write state)	RW
SAD0	Slave address	<In both modes> The address data is compared.	RW
SAD1			
SAD2			
SAD3			
SAD4			
SAD5			
SAD6			

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

(4) I^2Ci clock control register (i = 0 to 2)

The I^2Ci 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 I^2Ci clock control register during transmission. If data is written during transmission, the I^2Ci clock generator is reset, so that data cannot be transmitted normally.

I^2Ci clock control register (i = 0 to 2)

Symbol	Address	When reset
IIC0S2	$02E4_{16}$	00_{16}
IIC1S2	$02EC_{16}$	00_{16}
IIC2S2	$02F4_{16}$	00_{16}

b7 b6 b5 b4 b3 b2 b1 b0

Bit Symbol	Bit name	Function			RW
CCR0	SCL frequency control bits	Setup value of CCR4–CCR0	Standard clock mode	High speed clock mode	RW
CCR1		00 to 02	Setup disabled	Setup disabled	
CCR2		03	Setup disabled	333	
CCR3		04	Setup disabled	250	
CCR4		05	100	400 (See note)	
		06	83.3	166	
		:	500/CCR value	1000/CCR value	
		1D	17.2	34.5	
		1E	16.6	33.3	
		1F	16.1	32.3	
		(at BCLK = 10 MHz, unit : kHz)			
FAST MODE	SCL mode specification bit	0 : Standard clock mode 1 : High-speed clock mode			RW
ACK BIT	ACK bit	0 : ACK is returned. 1 : ACK is not returned.			RW
ACK	ACK clock bit	0 : No ACK clock 1 : ACK clock			RW

Note: At 400 kHz in the high-speed clock mode, the duty is as below.
"0" period : "1" period = 3 : 2
In the other cases, the duty is as below.
"0" period : "1" period = 1 : 1

Fig. 13.7 I^2Ci clock control register (i = 0 to 2)

(5) I^2Ci control register (i = 0 to 2)

The I^2Ci 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 "000_2" and the address data is always transmitted and received in 8 bits.

Note: When the bit counter value = "111_2," a STOP condition and START condition cannot be waited.

■ **Bit 3: I^2C-BUS interface i use enable bit (ESO)**

This bit enables usage of the multimaster I^2C-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 I^2Ci status register).

• Writing data to the I^2Ci data shift register and the I^2Ci 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) I^2Ci 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 I^2Ci 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 I^2Ci address register are compared with address data.

I^2Ci control register (i = 0 to 2)

Symbol	Address	When reset
IIC0S1D	$02E3_{16}$	00_{16}
IIC1S1D	$02EB_{16}$	00_{16}
IIC2S1D	$02F3_{16}$	00_{16}

b7 b6 b5 b4 b3 b2 b1 b0

Bit Symbol	Bit name	Function	RW
BC0 BC1 BC2	Bit counter (Number of transmit/receive bits)	b2 b1 b0 0 0 0 : 8 0 0 1 : 7 0 1 0 : 6 0 1 1 : 5 1 0 0 : 4 1 0 1 : 3 1 1 0 : 2 1 1 1 : 1	RW
ESO	I^2C-BUS interface i use enable bit	0 : Disabled 1 : Enabled	RW
ALS	Data format selection bit	0 : Addressing format 1 : Free data format	RW
10BIT SAD	Address format selection bit	0 : 7-bit addressing format 1 : 10-bit addressing format	RW
Nothing is assigned. In an attempt to write to these bits, write “0.” The value, if read, turns out to be “0.”			—

Fig. 13.8 I^2Ci 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 "00$_{16}$" 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)**

n 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/$\overline{W}$ bit) of the address data transmitted by the master is "1." When the ALS bit is "0" and the R/$\overline{W}$ 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 occurence 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 occurence 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 transmittion/reception (occurrence of transmission/reception interrupt request) <IICIRQ>.

I^2Ci status register (i = 0 to 2)

Symbol	Address	When reset
IIC0S1	$02E2_{16}$	0001000?$_2$
IIC1S1	$02EA_{16}$	0001000?$_2$
IIC2S1	$02F2_{16}$	0001000?$_2$

b7	b6	b5	b4	b3	b2	b1	b0

Bit Symbol	Bit name	Function	RW
LRB	Last receive bit	0 : Last bit = "0" 1 : Last bit = "1" (See note 1)	RO
AD0	General call detecting flag	0 : No general call detected 1 : General call detected (See note 1)	RO
AAS	Slave address comparison flag	0 : Address mismatch 1 : Address match (See note 1)	RO
AL	Arbitration lost detecting flag	0 : Not detected 1 : Detected (See note 1)	RO
PIN	I^2C-BUS interface i interrupt request bit	0 : Interrupt request issued 1 : No interrupt request issued (See note 2)	RO
BB	Bus busy flag	0 : Bus free 1 : Bus busy (See note 1)	RO
TRX	Communication mode specification bits	b7b6 0 0 : Slave receive mode 0 1 : Slave transmit mode 1 0 : Master receive mode 1 1 : Master transmit mode	RW
MST			

Notes 1: These bits and flags can be read out, but cannot be written.
2: This bit can be written only "1."

Fig. 13.9 I^2Ci status register (i = 0 to 2)

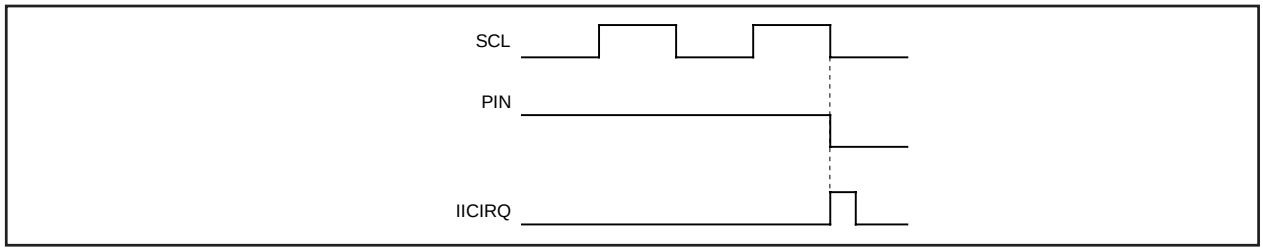


Fig. 13.10 Interrupt request signal generation timing

(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.

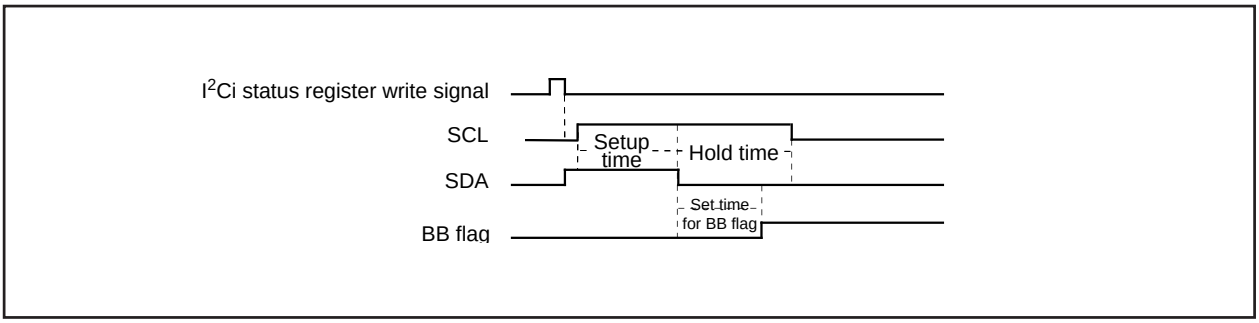


Fig. 13.11 START condition generation timing diagram

(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.

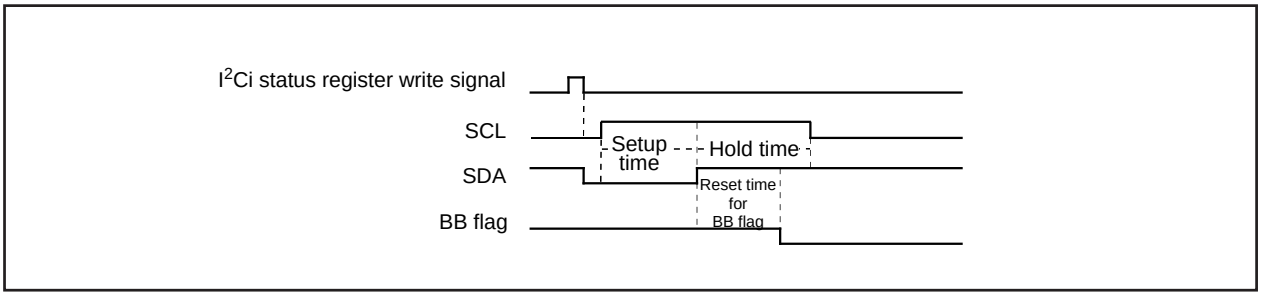


Fig. 13.12 STOP condition generation timing diagram

Table 13.2 START condition/STOP condition generation timing table

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

(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.

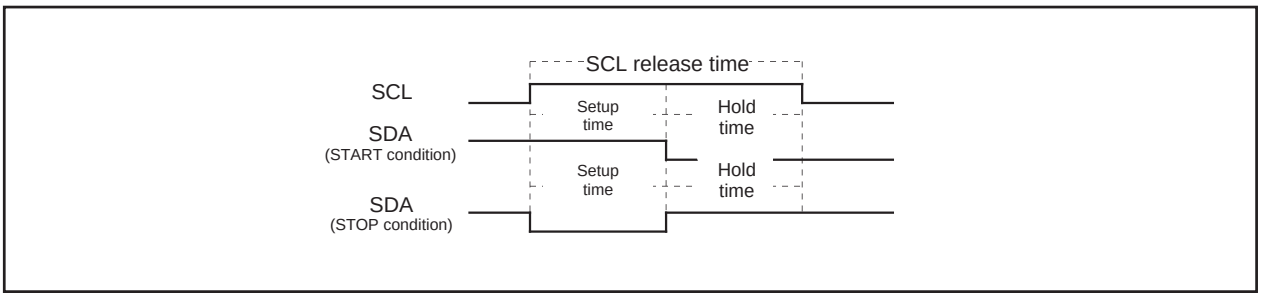


Fig. 13.13 START condition/STOP condition detect timing diagram

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 I^2Ci 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 I^2Ci address register. At the time of this comparison, address comparison of the RBW bit of the I^2Ci 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 I^2Ci 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 I^2Ci address register. At the time of this comparison, an address comparison between the RBW bit of the I^2Ci address register and the R/$\overline{W}$ bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the R/$\overline{W}$ 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 I^2Ci status register is set to "1." After the second-byte address data is stored into the I^2Ci 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 I^2Ci address register to "1" by software. This processing can match the 7-bit slave address and R/$\overline{W}$ data, which are received after a RESTART condition is detected, with the value of the I^2Ci 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 I^2Ci address register and "0" in the RBW bit.
② Set the ACK return mode and SCL = 100 kHz by setting "85$_{16}$" in the I^2Ci clock control register.
③ Set "10$_{16}$" in the I^2Ci status register and hold the SCL at the HIGH.
④ Set a communication enable status by setting "08$_{16}$" in the I^2Ci control register.
⑤ Set the address data of the destination of transmission in the high-order 7 bits of the I^2Ci data shift register and set "0" in the least significant bit.
⑥ Set "F0$_{16}$" in the I^2Ci 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 I^2Ci 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 "D0$_{16}$" in the I^2Ci 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 I^2Ci address register and "0" in the RBW bit.
② Set the no ACK clock mode and SCL = 400 kHz by setting "25$_{16}$" in the I^2Ci clock control register.
③ Set "10$_{16}$" in the I^2Ci status register and hold the SCL at the HIGH.
④ Set a communication enable status by setting "08$_{16}$" in the I^2Ci 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 I^2Ci 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 I^2Ci status register is set to "1" and an interrupt request signal occurs.
•In the cases other than the above:
AD0 and AAS of the I^2Ci status register are set to "0" and no interrupt request signal occurs.

⑦ Set dummy data in the I^2Ci 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.

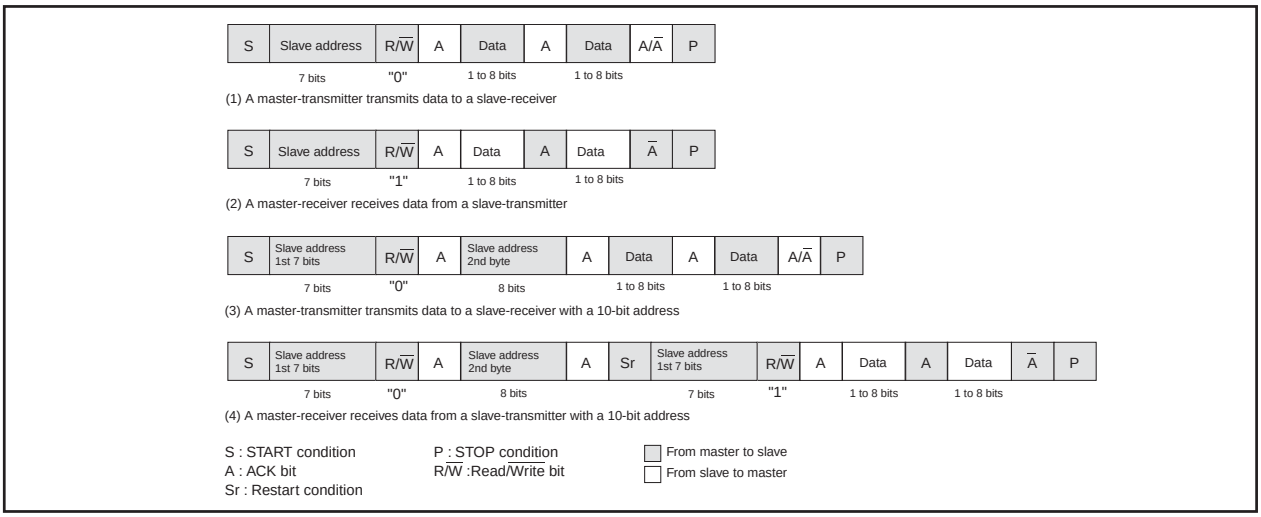


Fig. 13.14 Address data communication format

(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 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 × 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>)
        NOP
        NOP
        MOV.B   #F0H, IICiS1        (Trigger of RESTART condition generating)
                :
```

① Use the I^2Ci transmit buffer register to write the slave address value to the I^2Ci data shift register. And also, be sure to add NOP instruction × 4.

■ **Writing to I^2Ci 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 I^2Ci data shift register (IICiS0) and the I^2Ci 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 0260$_{16}$/0300$_{16}$) to "0." These settings can realize the low-power dissipation.

Notes 1. 525i (480i)/525p (480p):ID1 data slice can be performed. No CC data slice at 525p (480p).

2. When there is no specification, it becomes the publication about 525i (480i) below.

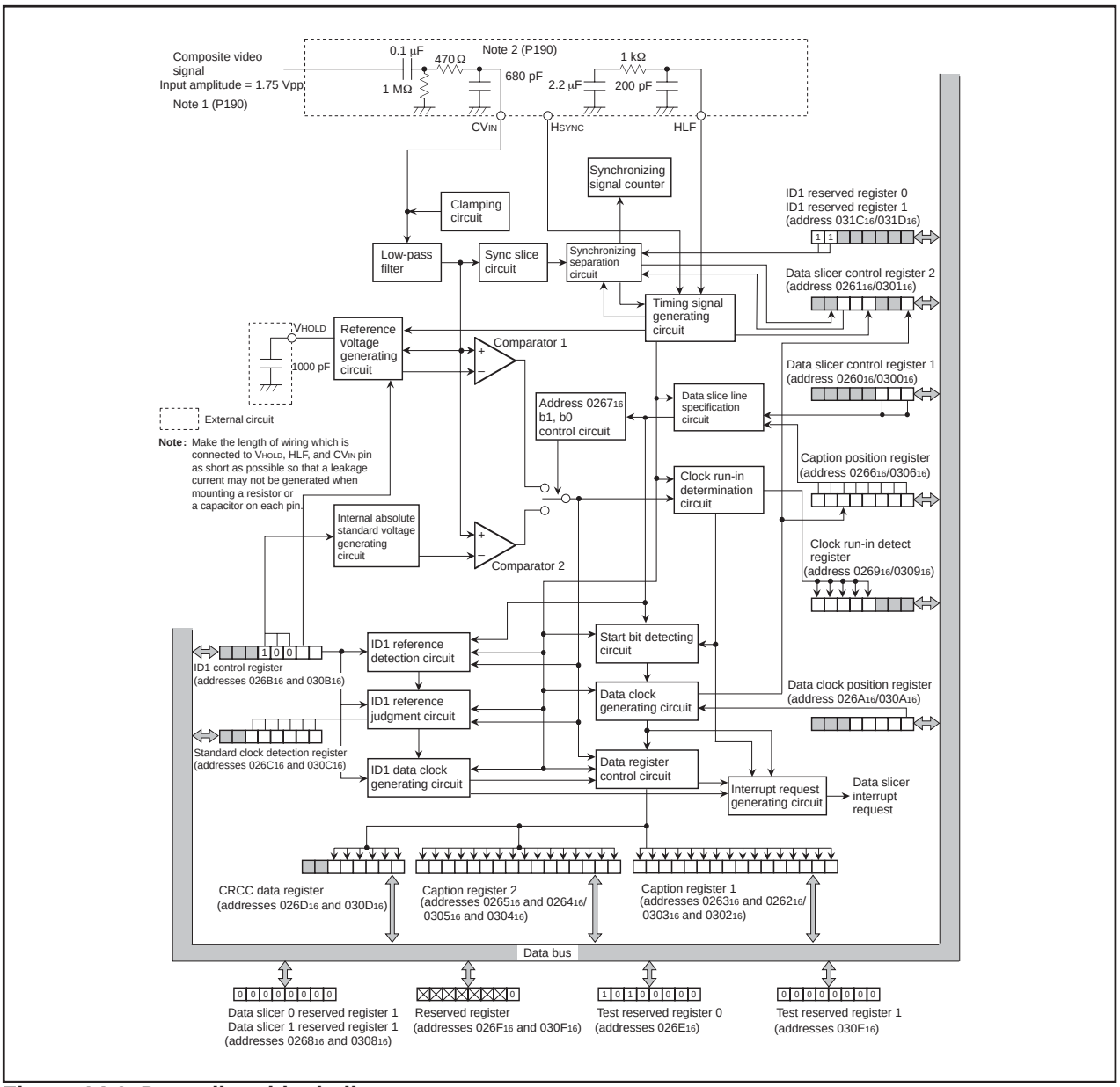


Figure 14.1 Data slicer block diagram

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 < VCCi + 0.3 V.

Vcci shows Vcci power supply pin voltage.

Sink tip clamp pin serves as (43/120) x VCCi .

Example) In the case of VCCi = 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.

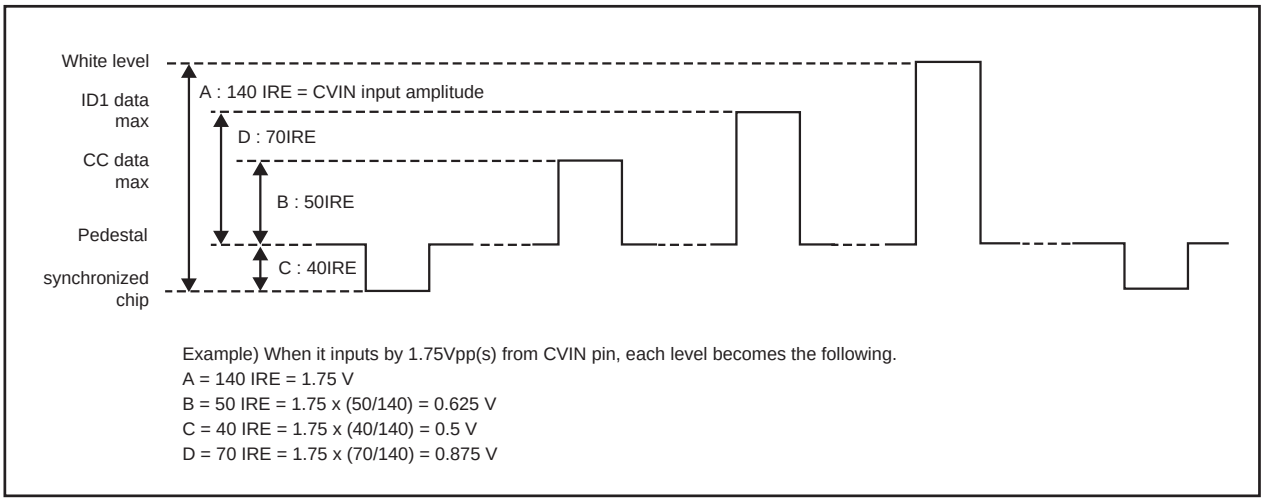


Figure 14.2 Each signal level to input amplitude of CVIN pin

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 0260_{16}/0300_{16}) is “0,” terminate the pins as shown in Figure 14.3

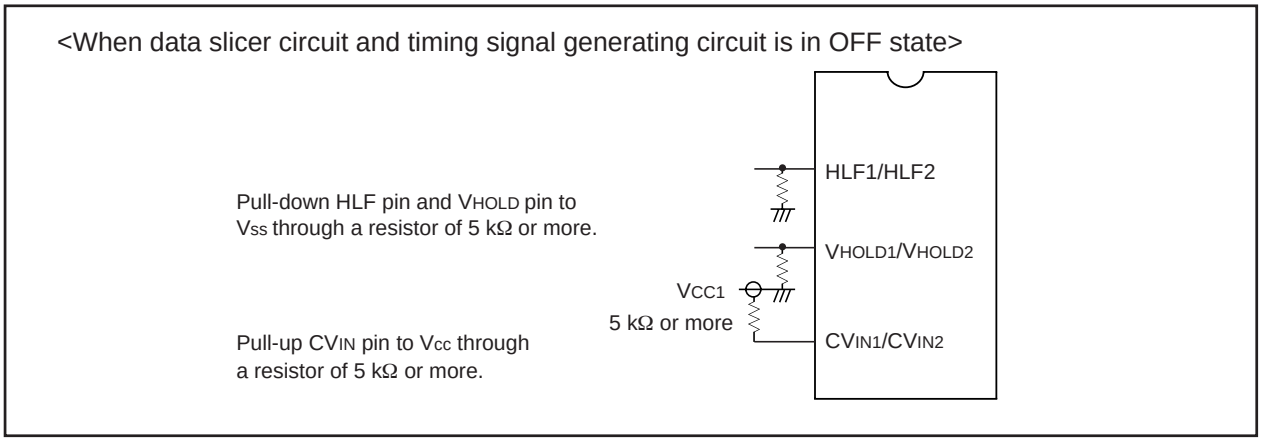


Figure 14.3 Termination of data slicer input/output pins when data slicer circuit and timing generating circuit is in OFF state

Figures 14.4 and 14.5 the data slicer control registers.

Data slicer control register 1

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

Symbol	Address	When reset
DSC01	026016	0016
DSC11	030016	0016

Bit symbol	Bit name	Function	RW
DSC010/ DSC110	Data slicer and timing signal generating circuit control bit	0: Stopped (Set at slicer unused) 1: Operating (Set at slicer used)	RW
DSC011/ DSC111	Selection bit of data slice reference voltage generating field	0: F2 1: F1 At two lines: CC21, and CCX or ID1 are sliced (notes 1 and 2) 1: Select F1, normally At only ID1 is sliced 0/1: Select either (note 3) 525p (480p):When ID1 data slice X:This bit setting is invalid.	RW
DSC012/ DSC112	Reference clock source selection bit	0: Video signal (Set “0”, normally) 1: HSYNC signal	RW
Reserved bits		Must always be set to “0”	RW

Notes 1. Selected by addresses 0266_{16}, $026B_{16}$, 0306_{16} and $030B_{16}$ register setting.
2. When ID1 slice is set, addresses $026B_{16}$ and $030B_{16}$ are need to be set.
3. It is required to superimpose F1 and F2 on the same data.
4. CC21: line 21data of CC format.
CCX: the line data which can be selected by addresses 0266_{16} and 0306_{16} of CC format.
ID1: ID1 format data.

Definition of fields 1 (F1) and 2 (F2)

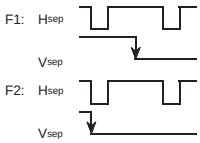


Figure 14.4 Data slicer control register 1

Data slicer control register 2

b7 b6 b5 b4 b3 b2 b1 b0
0 0

Symbol	Address	When reset
DSC02	026116	$?0?0??0?_2$
DSC12	030116	$?0?0??0?_2$

Bit symbol	Bit name	Function	RW
DSC020/ DSC120	Caption data latch completion flag 1	When two lines of CC21 and CCX are sliced, 0: Incompletion of CC21 caption data latch, or no clock run-in. 1: Completion of CC21 caption data latch, and clock run-in. When only CCX is sliced, 0: Incompletion of CCX caption data latch, or no clock run-in. 1: Completion of CCX caption data latch, and clock run-in. When only ID1 is sliced, 0: Incompletion of ID1 caption data latch. 1: Completion of ID1 caption data latch. note: A flag is reset by 0 in falling of vertical synchronized signal.	RO
Reserved bit		Must always be set to “0”	RW
Test bit		Read-only	RO
DSC023/ DSC123	Field determination flag	0: F2 1: F1 (*)This flag is invalid at $026B_{16}$ and $030B_{16}$ at the time of 525p(480p) selection.	RO
DSC024/ DSC124	Vertical synchronous signal (V_{sep}) generating method selection bit	0: Method (1) 1: Method (2)	RW
DSC025/ DSC125	V-pulse shape determination flag	0: Match 1: Mismatch	RO
Reserved bit		Must always be set to “0”	RW
Test bit		Read-only	RO

Definition of fields 1 (F1) and 2 (F2)

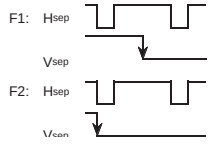


Figure 14.5 Data slicer control register 2

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 kiloohms 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 $031C_{16}$ and $031D_{16}$) 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 (H_{sep})

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

(2) Vertical synchronous signal (V_{sep})

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 0261_{16}/0301_{16}).

- •Method 1 The "L" level width of the composite sync signal is measured. If this width exceeds a certain time, a V_{sep} 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 V_{sep} signal is generated in synchronization with the rising of the timing signal (refer to Figure 14.6).

Figure 14.6 shows a V_{sep} 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 determinating 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."

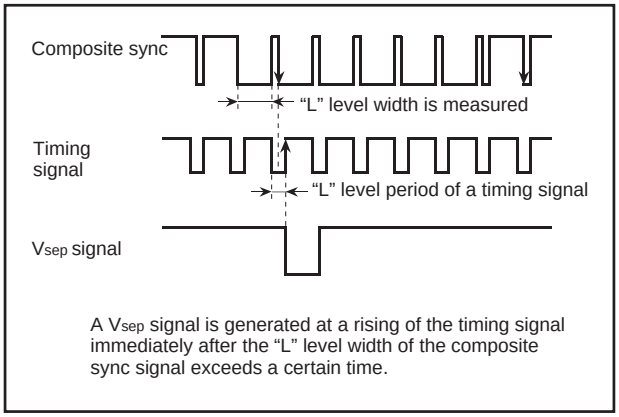


Figure 14.6 V_{sep} generating timing (method 2)

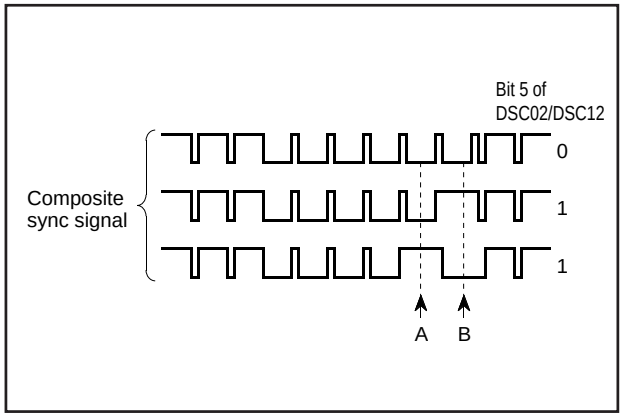


Figure 14.7 Determination of v-pulse waveform

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 0260_{16}/0300_{16}) to "1."

The reference clock is the H_{SYNC} signal can be used as a count source instead of the composite sync signal. However, when the H_{SYNC} 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 0260_{16}/0300_{16}).

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 0266_{16}/0306_{16}) 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 Hsep 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 "00_{16}" to "$1F_{16}$" 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 0266_{16} and 0306_{16} = 10000b.

525p (480p):When ID1 data slice, set up addresses 0266_{16}/0306_{16} bit 4-0 = 00001b and the data clock position register (addresses $026A_{16}$ and $030A_{16}$) 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 0266_{16} and 0306_{16} = 00b or 01b.

525p (480p):When ID1 data slice, set up the addresses 0266_{16} and 0306_{16} 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.

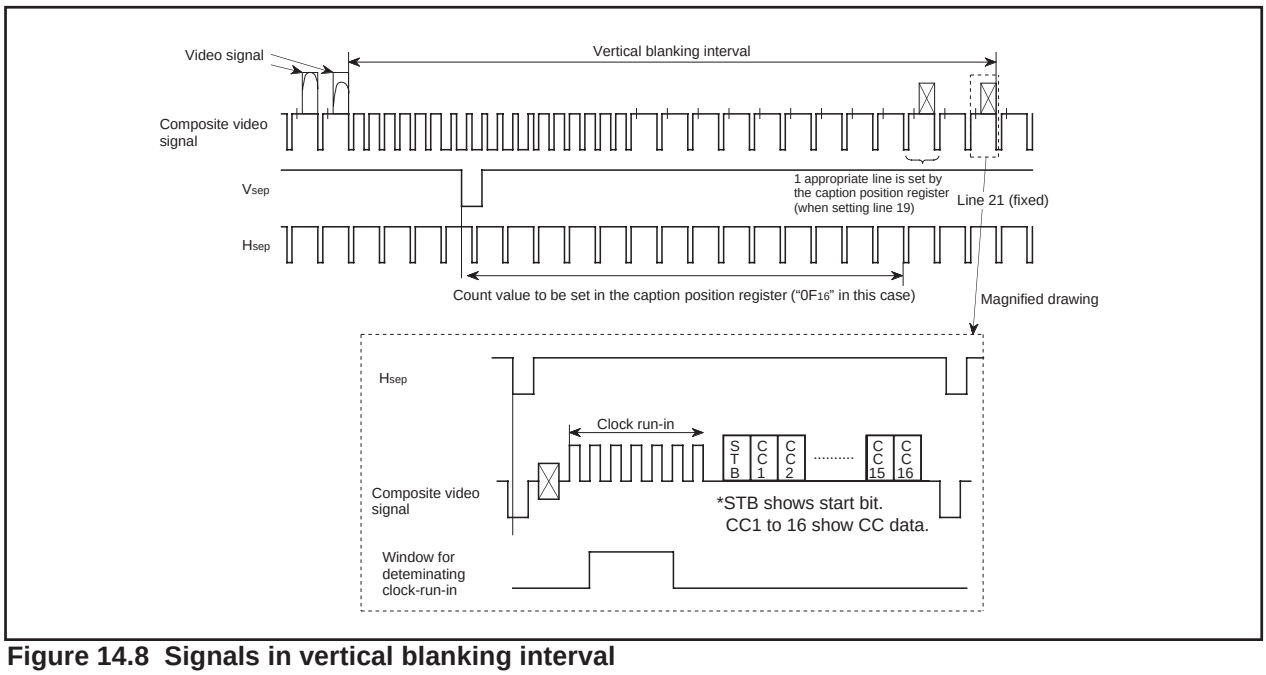


Figure 14.8 Signals in vertical blanking interval

Caption position register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
CPS0	0266_{16}	00?00000$_2$
CPS1	0306_{16}	00?00000$_2$

Bit symbol	Bit name	Function	RW
CPS00/CPS10	Caption position bits	Set caption position (CCX or ID1). For CCX, refer to Table 15.1. For ID1 slice, set bits 4 to 0 = 10000b (line 20 selection) When 525p (480p) ID1 data slice, set up bit 4-0 = 00001b (line 41 selection). (*) addresses $026A_{16}$ and $030A_{16}$ bit 6 and 5 = 01b need to be set up.	RW
CPS01/CPS11			RW
CPS02/CPS12			RW
CPS03/CPS13			RW
CPS04/CPS14			RW
CPS05/CPS15	Caption data latch completion flag 2	When two lines of CC21 and CCX are sliced, 0: Incompletion of CCX caption data latch, or no clock run-in. 1: Completion of CCX caption data latch, and clock run-in. When two lines of CC21 and ID1 are sliced, 0: Incompletion of ID1 caption data latch. 1: Completion of ID1 caption data latch. This bit is invalid when slice only any one line of CC21, CCX and ID1. note: A flag is reset by 0 in rising of vertical synchronized signal.	RO
CPS06/CPS16	Slice line mode specification bits	Refer to Table 15.1 at slice CC21 or CCX. Set bits 6 and 7 = 00b or 01b when ID1 slice. When 525p (480p) ID1 data slice, set bit 7 and 6 = 01b.	RW
CPS07/CPS17			RW

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	• 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)	• Field specified by bit 1 of DSC01/DSC11 • Line 21 (total 1 line)
0	1	• Both fields of F1 and F2 • A line specified by bits 4 to 0 of CPS0/CPS1 (total 1 line) (See note 3)	• 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	• Both fields of F1 and F2 • Line 21 (total 1 line)	• Field specified by bit 1 of DSC01/DSC11 • Line 21 (total 1 line)
1	1	• 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)	• 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 "00_{16}" – "10_{16}" to bits 4 to 0 of CPS0/CPS1.
3: Set the value of "00_{16}" – "$1F_{16}$" to bits 4 to 0 of CPS0/CPS1.

Slice standard voltage selection register

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

Symbol	Address	When reset
SBV0	267_{16}	00_{16}
SBV1	307_{16}	00_{16}

Bit symbol	Bit name	Function	RW
SVB00/SVB10	Slice standard voltage selection bit	b1, b0 0 0 Standard voltage selection by standard voltage generating circuit. 0 1 Internal absolute standard voltage selection. 1 0 CC21 is the voltage by the standard voltage generating circuit. CCX or ID1 is internal absolute standard voltage selection. 1 1 Do not set	RW
SVB01/SVB11			
Reseved bits		Must be set to "0."	RW

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 0269_{16}/0309_{16}).

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 0269_{16}/the $030C_{16}$). 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.

Clock run-in detection register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
CRD0	0269_{16}	$00000???_2$
CRD1	0309_{16}	$00000???_2$

Bit symbol	Bit name	Function	RW
Test bits		Read-only	RO
CRD03/CRD13	Clock run-in detection bits	Number of reference clocks to be counted in one clock run-in pulse period.	RO
CRD04/CRD14			RO
CRD05/CRD15			RO
CRD06/CRD16			RO
CRD07/CRD17			RO

Figure 14.11 Clock run-in detection register

Standard clock detection register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
BCD0	$026C_{16}$	$??_{16}$
BCD1	$030C_{16}$	$??_{16}$

Bit symbol	Bit name	Function	RW
BCD00/BCD10	ID1 REF width detection bit	The number of standard clocks counted in a fixed period of ID1 REF. It is effective, only when "1" is set as the $026C_{16}$ and $030C_{16}$th bits 0 and ID1 slice function is operating.	RO
BCD01/BCD11			
BCD02/BCD12			
BCD03/BCD13			
BCD04/BCD14			
BCD05/BCD15			
Nothing is assigned. If an attempt to write to these bits, write “0.” The read turns out to be “0.”			–

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 $026A_{16}$/$030A_{16}$). 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 $026D_{16}$/$030D_{16}$). If 20-bit data is stored in each register, the completion flag of a caption data latch will be set.

Data clock position register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
DPS0	026A16	X0000000 2
DPS1	030A16	X0000000 2

Bit symbol	Bit name	Function	RW
DPS00/DPS10	Data clock position set bits	Only when CC21 or CCX slice is effective.	RW
DPS01/DPS11			
DPS02/DPS12			
DPS03/DPS13			
DPS04/DPS14			
DPS05/DPS15	Reserved bit Caption position bit 2	It is effective only at the time of 525p (480p) ID1 slice. b6 b5 0 1	RW
DPS06/DPS16			
Nothing is assigned. If an attempt to write to this bit, write “0.” The read turns out to be “0.”			—

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.

Caption register 1L

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
C1L0	026216	??16
C1L1	030216	??16

Bit symbol	Bit name	Function	RW
C1L00/C1L10	Caption data 1L	The following data is stored in 026616, and the 030616 street of bits 7 and 6. In the case of CC caption b7, b6 0 0 Data 16-9 of CC21 is stored in bit 7-0. 0 1 Data 16-9 of CCX is stored in bit 7-0. 1 0 Data 16-9 of CC21 is stored in bit 7-0. 1 1 Data 16-9 of CC21 is stored in bit 7-0. In the case of ID1 caption b7, b6 0 0 Data 16-9 of CC21 is stored in bit 7-0. 0 1 Data 14-7 of ID1 is stored in bit 7-0(*). 1 0 Data 16-9 of CC21 is stored in bit 7-0. 1 1 Do not set. (*)When 525p (480p) ID1 data slice, set it as b7 and b6 = 01b. The caption data stored also becomes the same.	RO
C1L01/C1L11			
C1L02/C1L12			
C1L03/C1L13			
C1L04/C1L14			
C1L05/C1L15			
C1L06/C1L16			
C1L07/C1L17			

Figure 14.14 Caption register 1L

Caption register 1H

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
C1H0	0263_{16}	$??_{16}$
C1H1	0303_{16}	$??_{16}$

Bit symbol	Bit name	Function	RW
C1H00/C1H10	Caption data 1H	The following data is stored in 0266_{16}, and the 0306_{16} street of bits 7 and 6. In the case of CC caption b7, b6 0 0 Data 8-1 of CC21 is stored in bit 7-0. 0 1 Data 8-1 of CCX is stored in bit 7-0. 1 0 Data 8-1 of CC21 is stored in bit 7-0. 1 1 Data 8-1 of CC21 is stored in bit 7-0. In the case of ID1 caption b7, b6 0 0 Data 8-1 of CC21 is stored in bit 7-0. 0 1 Data 6-1 of ID1 is stored in bit 7-2 (note). 1 0 Data 8-1 of CC21 is stored in bit 7-0. 1 1 Do not set. Note: The reading value of bits 1 and 0 is unfixed. 525p (480p):When ID1 data slice, set it as b7 and b6 = 01b. The caption data stored also becomes the same.	RO
C1H01/C1H11			
C1H02/C1H12			
C1H03/C1H13			
C1H04/C1H14			
C1H05/C1H15			
C1H06/C1H16			
C1H07/C1H17			

Figure 14.15 Caption register 1H

Caption register 2L

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
C2L0	0264_{16}	$??_{16}$
C2L1	0304_{16}	$??_{16}$

Bit symbol	Bit name	Function	RW
C2L00/C2L10	Caption data 2L	The following data is stored in 0266_{16}, and the 0306_{16} street of bits 7 and 6. In the case of CC caption b7, b6 0 0 Data 16-9 of CCX is stored in bit 7-0. 0 1 Data is invalid. 1 0 Data is invalid. 1 1 Data 16-9 of CCX is stored in bit 7-0. In the case of ID1 caption b7, b6 0 0 Data 14-7 of ID1 is stored in bit 7-0. 0 1 Data is invalid. 1 0 Data is invalid. 1 1 Do not set.	RO
C2L01/C2L11			
C2L02/C2L12			
C2L03/C2L13			
C2L04/C2L14			
C2L05/C2L15			
C2L06/C2L16			
C2L07/C2L17			

Figure 14.16 Caption register 2L

Caption register 2H

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
C2H0	0265_{16}	$??_{16}$
C2H1	0305_{16}	$??_{16}$

Bit symbol	Bit name	Function	RW
C2H00/C2H10	Caption data 2H	The following data is stored in 0266_{16}, and the 0306_{16} street of bits 7 and 6. In the case of CC caption b7, b6 0 0 Data 8-1 of CCX is stored in bit 7-0. 0 1 Data is invalid. 1 0 Data is invalid. 1 1 Data 8-1 of CCX is stored in bit 7-0. In the case of ID1 caption b7, b6 0 0 Data 6-1 of ID1 is stored in bit 7-2 (note). 0 1 Data is invalid. 1 0 Data is invalid. 1 1 Do not set. Note : The reading value of bits 1 and 0 is unfixed.	RO
C2H01/C2H11			
C2H02/C2H12			
C2H03/C2H13			
C2H04/C2H14			
C2H05/C2H15			
C2H06/C2H16			
C2H07/C2H17			

Figure 14.17 Caption register 2H

CRCC data register

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕						

Symbol	Address	When reset
CRC0	$026D_{16}$	00?????? $_2$
CRC1	$030D_{16}$	00?????? $_2$

Bit symbol	Bit name	Function	RW
CRC00/CRC10	CRCC data register	Data 20-15 of ID1 is stored in bit 5-0. It is effective, only when "1" is set as the bit 0 of addresses $026B_{16}$ and $030B_{16}$, and ID1 slice function is operating.	RO
CRC01/CRC11			
CRC02/CRC12			
CRC03/CRC13			
CRC04/CRC14			
CRC05/CRC15			
Nothing is assigned. If an attempt to write to these bits, write “0.” The read turns out to be “0.”			–

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 0266_{16}/0306_{16}). 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.

Data slicer 0 reserved register 1
Data slicer 1 reserved register 1

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

Symbol	Address	When reset
DR01	0268_{16}	00_{16}
DR11	0308_{16}	00_{16}

Bit symbol	Bit name	Description	RW
Reserved bits		Must always be set to “0”	RW

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.

ID1 control register

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

Symbol	Address	When reset
IDC0	$026B_{16}$	00_{16}
IDC1	$030B_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
IDC00/IDC10	ID1 selection bit	0: ID1 slice unused 1: ID1 slice operate * Must always be set to "0" at ID1 slice unused. When set to "1", be sure to set bits 4 to 0 = 10000b of addresses 0266_{16}/0306_{16}. 525p (480p):When ID1 data slice, Set up bit 4-0 = 00001b at addresses 0266_{16}/0306_{16}. Set bit 6 and 5 = 01b of addresses $026A_{16}$/$030A_{16}$.	RW
IDC01/IDC11	ID1 limitation slice select bit	0: Set this bit when slice simultaneously with CC21. 1: Set this bit when slice only ID1.	
IDC02/IDC12 IDC03/IDC13 IDC04/IDC14	Internal absolute standard voltage setting bit	b4 b3 b2 1 0 0	
Reserved bit		Must always be set to "0."	
IDC0b/IDC1b 525i (480i)/525p (480p) Selection bit		0:525i (480i) set up at CC21, CCX, and ID1 slice 0:525p (480p) set up at ID1 slice	
Reserved bit		Must always be set to "0."	

Figure 14.20 ID1 control register

ID1 reserved register 0 and 1

b7	b6	b5	b4	b3	b2	b1	b0
1	1						

Symbol	Address	When reset
IRSV0	$031C_{16}$	00_{16}
IRSV1	$031D_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
The register only for read-out		The contents are unfixed when it reads.	RW
IRSV06/IRSV16	Sync slice setting bit	Must always be set to "1."	
IRSV07/IRSV17			

Figure 14.21 ID1 reserved register

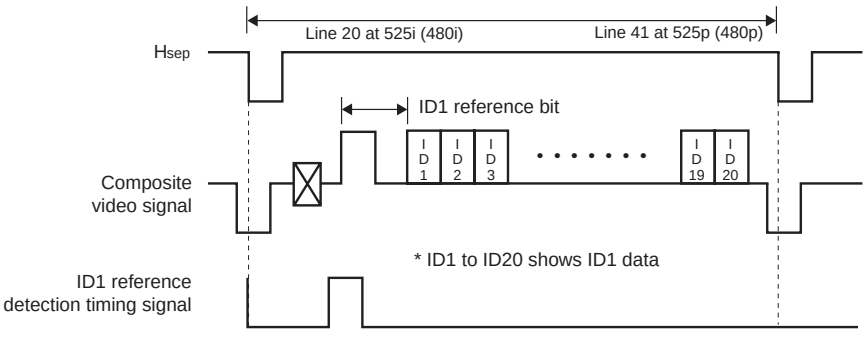


Figure 14.22 ID1 signal in vertial blanking interval

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 µs, 2048 µs, 4096 µs and 8192 µs) 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.

HSYNC counter register

b7

Symbol HC | Address 027E16 | When reset XXX00X0016

Bit symbol	Bit name	Function	RW
HCC0	Count source switch bit	0 : HC0 pin input 1 : HC1 pin input	RW
HCC1	Input polarity switch bit	0 : (Falling edge count) 1 : (Rising edge count)	RW
Nothing is assigned. In an attempt to write to this bit, write "0." The value, if read, turns out to be "0."			—
HCC3	Count freguency selection bits	b4 b3 <Count freguency> 0 0 : 1024 µs 0 1 : 2048 µs 1 0 : 4096 µs 1 1 : 8192 µs	RW
HCC4			RW
Nothing is assigned. In an attempt to write to these bits, write "0." The value, if read, turns out to be "0."			WO

Note: When HC0 and HC1 input are positive polarity (negetive polarity), HIGH width (LOW width) needs 3 main clock cycles or more of system clock.

Figure 15.1 HSYNC counter register

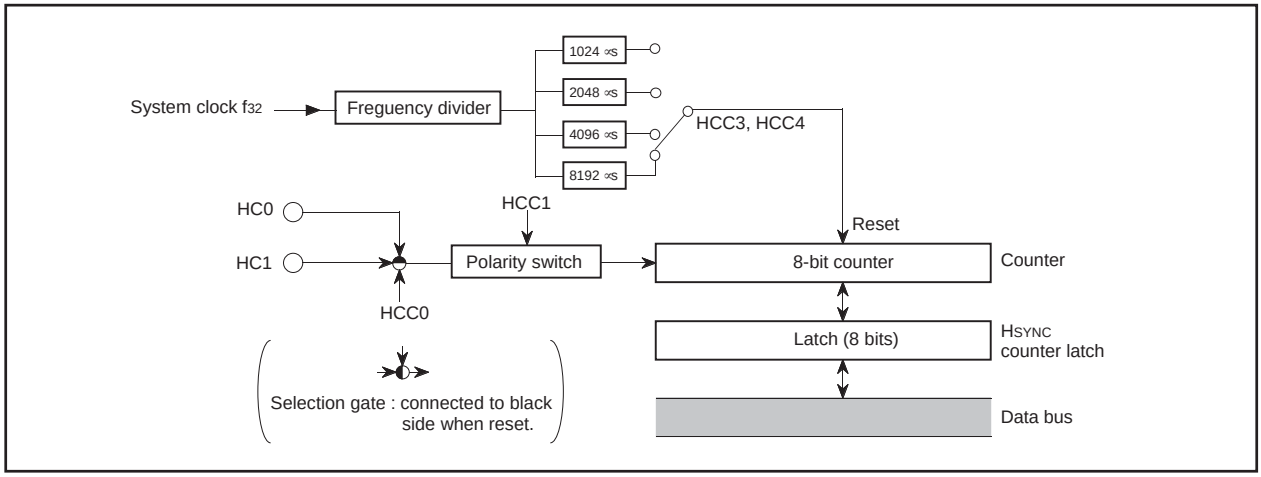


Figure 15.2 HSYNC counter block diagram

OSD Functions

Table 16.1 outlines the OSD functions of this microcomputer. This OSD function can display the following: the block display (32 characters × 16 lines or 42 characters × 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

Parameter \ Display style		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 × 16 lines/42 characters × 16 lines					1 character × 2 lines
Dot structure		16 × 20 dots (Character display area: 16 × 26 dots)		16 × 20 dots 12 × 20 dots 8 × 20 dots 4 × 20 dots	24 × 32 dots	16 × 26 dots	32 × 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)	× 1, × 2	× 1, × 2, × 3				× 1, × 2
	Dot size	1Tc × 1/2H, 1Tc × 1H	1Tc × 1/2H, 1Tc × 1H, 1.5Tc × 1/2H, 1.5Tc × 1H, 2Tc × 2H, 3Tc × 3H	1Tc × 1/2H, 1Tc × 1H, 2Tc × 2H, 3Tc × 3H		1Tc × 1/2H, 1Tc × 1H, 1.5Tc × 1/2H, 1.5Tc × 1H, 2Tc × 2H, 3Tc × 3H	1Tc × 1/2H, 1Tc × 1H, 2Tc × 1H, 2Tc × 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.

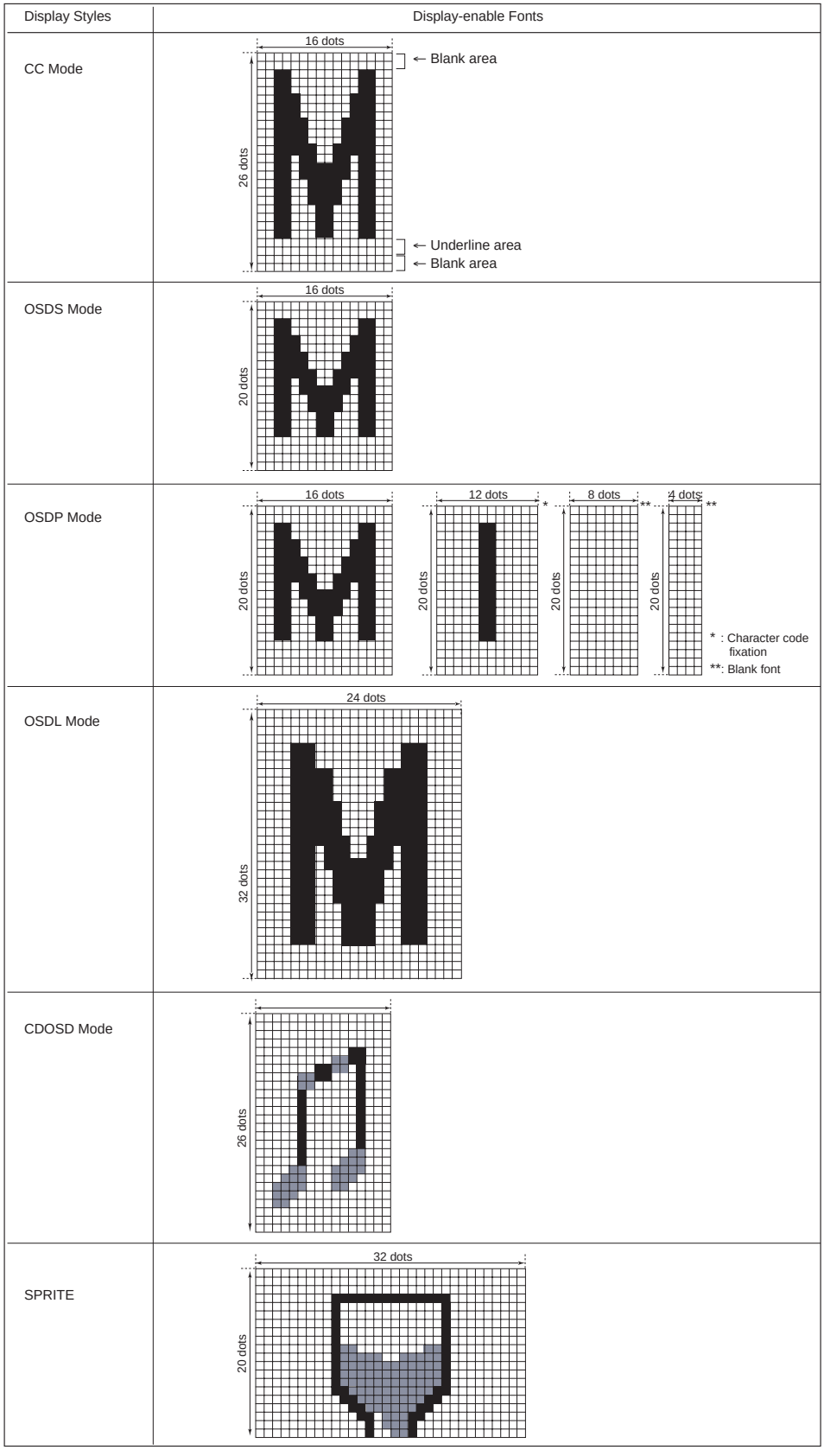


Figure 16.1 Display-enable fonts for each display style

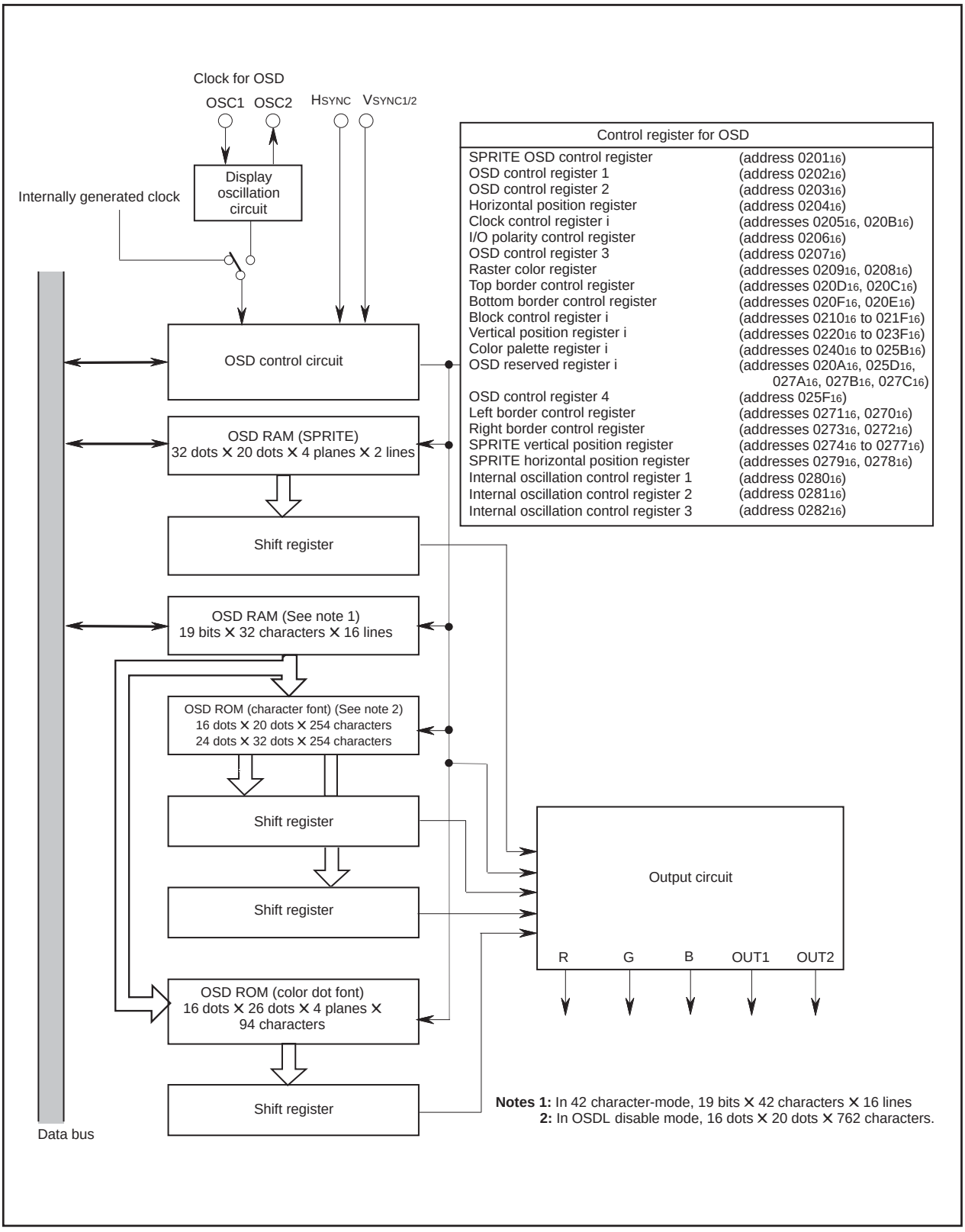


Figure 16.2 Block diagram of OSD circuit

OSD control register 1

b7

Symbol	Address	When reset
OC1	0202_{16}	00_{16}

OC16 .	Bit name	Function	RW
OC10	OSD control bit (See note 1)	0 : All-blocks and SPRITE display OFF 1 : All-blocks and SPRITE display ON	RW
OC11	Scan mode selection bit	0 : Normal scan mode 1 : Bi-scan mode	RW
OC12	Border type selection bit	0 : All bordered 1 : Shadow bordered (See note 2)	RW
OC13	Flash mode selection bit	0 : Color signal of character background part does not flash 1 : Color signal of character background part flashes	RW
OC14	Automatic solid space control bit	0 : OFF 1 : ON	RW
OC15	Vertical window/blank control bit	0 : OFF 1 : ON	RW
OC16	Layer mixing control bits (See note 3)	b7 b6 0 0: Logic sum (OR) of layer 1's color and layer 2's color 0 1: Layer 1's color has priority 1 0: Layer 2's color has priority 1 1: Do not set.	RW
OC17			RW

Notes 1 : When this bit is switched "1" from "0", the display screen remains unchanged until a rising (falling) of the next V_{SYNC}.
2 : Shadow border is output at right and bottom side of the font.
3 : OUT2 is always ORed, regardless of values of these bits.

Figure 16.3 OSD control register 1

Block control register i

b7	b6	b5	b4	b3	b2	b1	b0
X							

Symbol: BCi (i = 1 to 16)

Address: 0210₁₆ to 021F₁₆

When reset: Indeterminate

Bit symbol	Bit name	Function	RW
BCi_0	Display mode selection bits	b2 b1 b0 — Functions: 0 0 0 Display OFF 0 0 1 OSDS mode (No bordered) 0 1 0 CC mode 0 1 1 CDOSD mode 1 0 0 OSDP mode (No bordered) 1 0 1 OSDS mode (Bordered) 1 1 0 OSDP mode (Bordered) 1 1 1 OSDL mode	RW
BCi_1			RW
BCi_2			RW
BCi_3	Dot size selection bits	b6 b5 b4 b3 — Pre-divide ratio — Dot size: 0 0 0 0 × 1 1Tc × 1/2H 0 0 0 1 × 1 1Tc × 1H 0 0 1 0 × 1 2Tc × 2H 0 0 1 1 × 1 3Tc × 3H 0 1 0 0 × 2 1Tc × 1/2H 0 1 0 1 × 2 1Tc × 1H 0 1 1 0 × 2 2Tc × 2H 0 1 1 1 × 2 3Tc × 3H 1 1 0 0 1.5Tc × 1/2H (See notes 3, 4) 1 1 0 1 1.5Tc × 1H (See notes 3, 4) 1 1 0 0 × 3 1Tc × 1/2H 1 1 0 1 × 3 1Tc × 1H 1 1 1 0 × 3 2Tc × 2H 1 1 1 1 × 3 3Tc × 3H	RW
BCi_4			RW
BCi_5	Pre-divide ratio selection bits		RW
BCi_6			RW
Nothing is assigned. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—

Notes 1: Tc is OSD clock cycle divided in pre-divide circuit
2: H is HSYNC
3: This character size is available only in Layer 2. At this time, set layer 1’s pre-divide ratio = × 2, layer 1’s horizontal dot size = 1Tc.
4: In OSDL and OSDP modes, 1.5Tc size cannot be used.

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

<table>
<tr><th>Parameter \ Block</th><th>Block in Layer 1</th><th colspan="2">Block in Layer 2</th></tr>
<tr><td>Display mode</td><td>CC, OSDS/L, CDOSD mode</td><td colspan="2">OSDS/L, CDOSD mode</td></tr>
<tr><td>Pre-divide ratio</td><td>× 1, × 2 (CC mode)
× 1 to × 3 (OSD, CDOSD mode)</td><td colspan="2">Same as layer 1 (See note)</td></tr>
<tr><td rowspan="2">Dot size</td><td rowspan="2">1Tc × 1/2H, 1Tc × 1H
(CC mode)</td><td>Pre-divide ratio = × 1</td><td>Pre-divide ratio = × 2</td></tr>
<tr><td>1Tc × 1/2H
1Tc × 1H</td><td>1Tc×1/2H, 1.5Tc×1/2H
1Tc×1H, 1.5Tc×1H (See note)</td></tr>
<tr><td>1Tc × 1H, 1Tc × 1/2H, 2Tc × 2H,
3Tc × 3H
(OSDS/L, CDOSD mode)</td><td colspan="2">• Same size as layer 1
•1.5Tc can be selected only when: layer 1's pre-divide ratio = × 2 AND layer 1's horizontal dot size = 1Tc.
As this time, vertical dot size is the same as layer 1.</td></tr>
<tr><td>Horizontal display start position</td><td>Arbitrary</td><td colspan="2">Same position as layer 1</td></tr>
<tr><td>Vertical display start position</td><td colspan="3">Arbitrary
However, when dot size is 2Tc × 2H or 3Tc × 3H, set difference between vertical display position of layer 1 and that of layer 2 as follows.
•2Tc × 2H: 2H units
•3Tc × 3H: 3H units</td></tr>
</table>

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

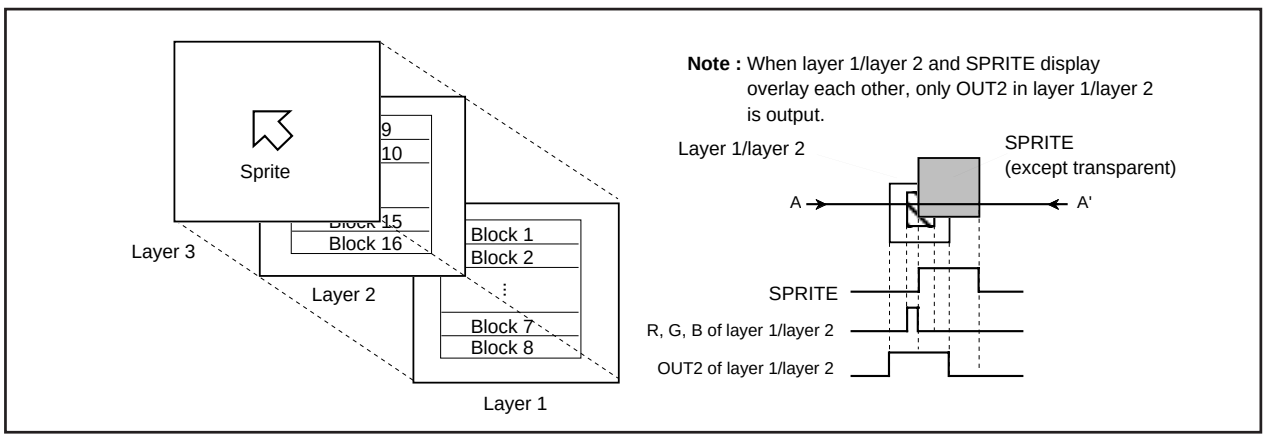
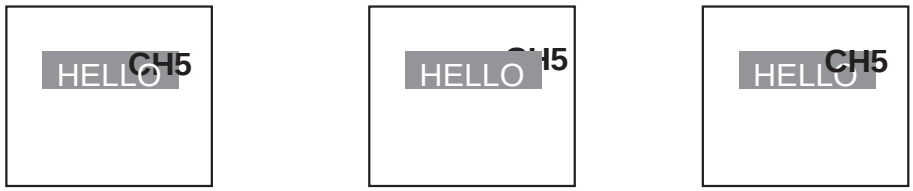


Figure 16.5 Triple layer OSD

Display example of layer 1 = “HELLO,” layer 2 = “CH5”



Logical sum (OR) of layer 1’s color and layer 2’s color (See note) OC17 = “0,” OC16 = “0”

Layer 1’s color has priority OC17 = “0”, OC16 = “1”

Layer 2’s color has priority OC17 = “1,” OC16 = “0”

Note: The logical sum (OR) of layer mixing is not OR of the color palette registers’ contents (color), but that of color pallet registers’ numbers (i).
Example) When the logical sum (OR) is performed on the color palettes 1 and 4; the number 1 (0001_2) and number 4 (0100_2) are ORed and it results in the number 5 (0101_2). That is, the contents (color) of color palette register 5 is output. The color of color palette register 5 is output in the ORed part, regardless of colors of color palettes registers 1 and 4.

Figure 16.6 Display example of layer mixing OSD

OSD control register 2

b7 b6

Symbol: OC2 | Address: 0203_{16} | When reset: 00_{16}

Bit symbol	Bit name	Function				RW
OC20	Display layer selection bits	b1	b0	Layer 1	Layer 2	RW
		0	0	CC, OSDS/L/P, CDOSD	—	
		0	1	CC, OSDS/L/P	CDOSD	
OC21		1	0	CC, OSDP, CDOSD	OSDS/L	RW
		1	1	CC, OSDP	CDOSD OSDS/L	
OC22	R, G, B signal output selection bit	0: Digital output 1: Analog output (8 gradations)				RW
OC23	Solid space output bit	1: OUT2 output				RW
OC24	Horizontal window/blank control bit	0: OFF 1: ON				RW
OC25	Window/blank selection bit 1 (horizontal)	0: Horizontal blank function 1: Horizontal window function				RW
OC26	Window/blank selection bit 2 (vertical)	0: Vertical blank function 1: Vertical window function				RW
OC27	OSD interrupt request selection bit	0: At completion of layer 1 block display 1: At completion of layer 2 block display				RW

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).

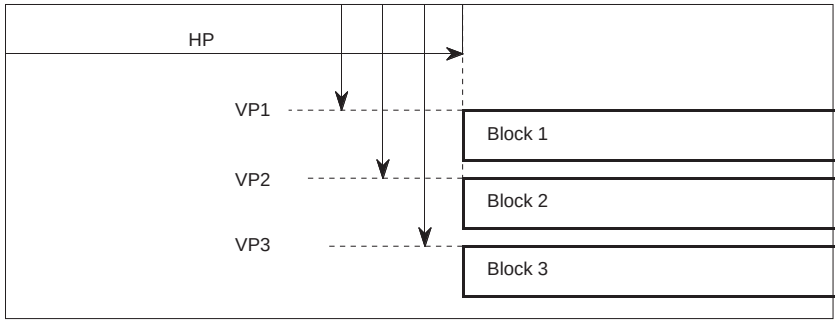
The display position of each block can be set in both horizontal and vertical directions by software.

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

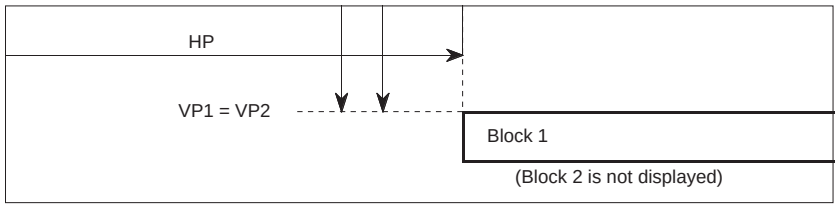
The display position in the vertical direction for each block can be selected from 1024-step display positions in units of 1 T_H (T_H = H_{SYNC} cycle).

Blocks are displayed in conformance with the following rules:

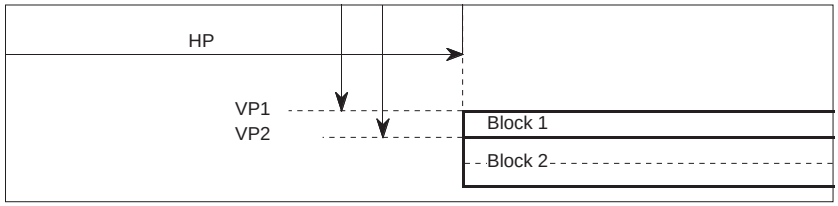
- When the display position is overlapped with another block in the same layer (Figure 16.8 (b)), a low block number (1 to 16) is displayed on the front.
- When another block display position appears while one block is displayed in the same layer (Figure 16.8 (c)), the block with a larger set value as the vertical display start position is displayed. However, do not display block with the dot size of 2T_C × 2H or 3T_C × 3H during display period (✻) of another block.
  - ✻ In the case of OSDS/P 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.



(a) Example when each block is separated



(b) Example when block 2 overlaps with block 1

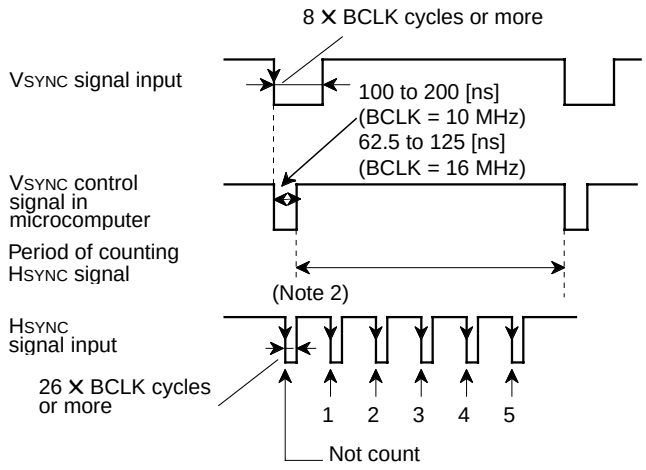


(c) Example when block 2 overlaps in process of block 1

Note: VPi (i = 1 to 16) indicates the vertical display start position of display block i.

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 × BCLK cycles or more) for avoiding jitter. The polarity of HSYNC and VSYNC signals can select with the I/O polarity control register (address 0206_{16}).



When bits 0 and 1 of the I/O polarity control register (address 0206_{16}) are set to "1" (negative polarity)

Notes 1 : The vertical position is determined by counting falling edge of HSYNC signal after rising edge of VSYNC control signal in the microcomputer.
2 : Do not generate falling edge of HSYNC signal near rising edge of VSYNC control signal in microcomputer to avoid jitter.
3 : The pulse width of HSYNC needs 26 × BCLK cycles or more.

Figure 16.9 Supplement explanation for display position

The vertical position for each block can be set in 1024 steps (where each step is 1TH (TH: HSYNC cycle)) as values “002$_{16}$” to “3FF$_{16}$” in vertical position register i (i = 1 to 16) (addresses 0220$_{16}$ to 023F$_{16}$). The vertical position register i is shown in Figure 16.10.

Vertical position register i

(b15) (b8)
b7 b6 b5 b4 b3 b2 b1 b0 b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
VPi (i = 1 to 16)	Even addresses within addresses 0220$_{16}$ to 023F$_{16}$, Odd addresses within addresses 0220$_{16}$ to 023F$_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
VPi_9 to VPi_0	Vertical display start position control bits of SPRITE font	Vertical display start position = TH × n (n: setting value, TH: HSYNC cycle)	RW
Nothing is assined. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—

Note : Do not set VPi ≤ “001$_{16}$,” VPi ≥ “400$_{16}$.”

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 4TOSC, TOSC being OSD oscillation cycle) as values “00$_{16}$” to “FF$_{16}$” in bits 0 to 7 of the horizontal position register (address 0204$_{16}$). The horizontal position register is shown in Figure 16.11.

Horizontal position register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
HP	0204$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
HP_7 to HP_0	Horizontal display start position control bits	Horizontal display start position = 4TOSC × n (n: setting value, TOSC: OSD oscillation cycle)	RW

Note : The setting value synchronizes with the VSYNC.

Figure 16.11 Horizontal position register

Note : 1Tc (Tc : 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 1Tc regardless of character sizes, however, the gap is 1.5Tc only when the character size is 1.5Tc.

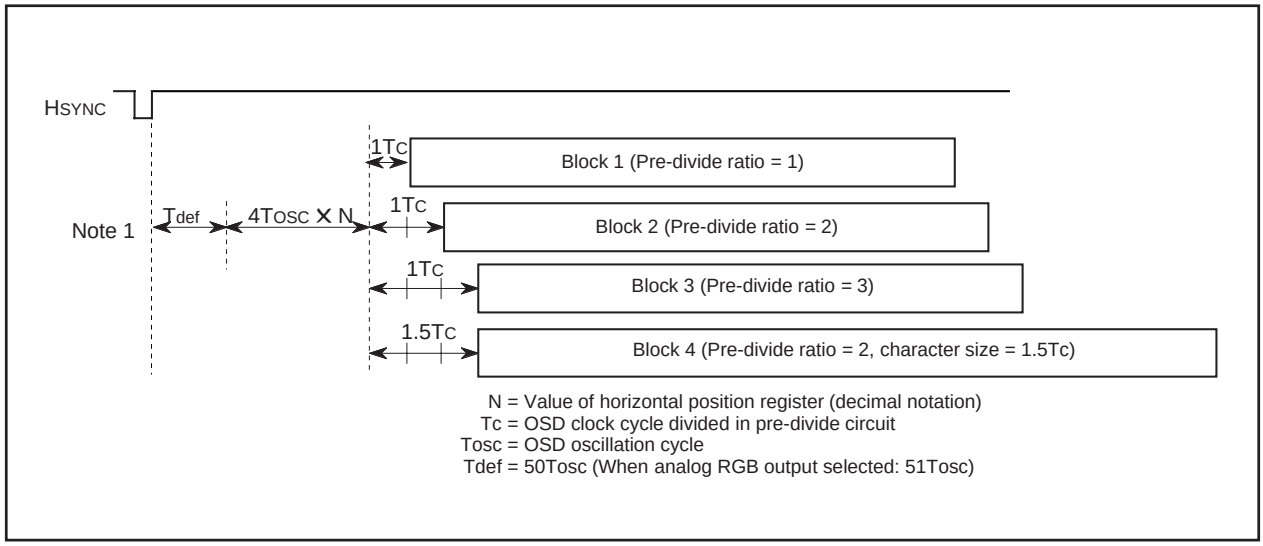


Figure 16.12 Notes on horizontal display start position

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.

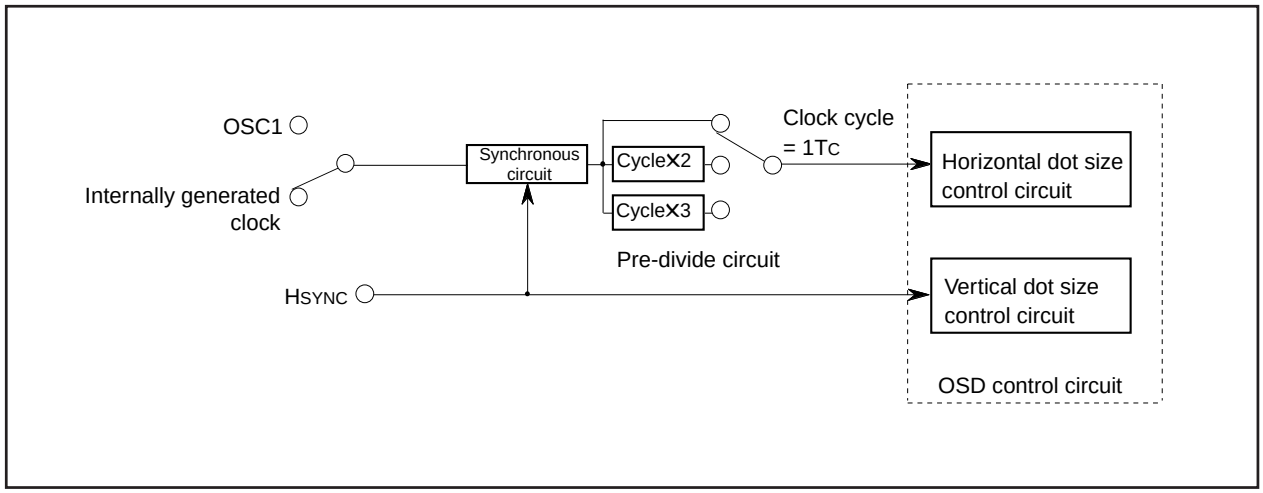


Figure 16.13 Block diagram of dot size control circuit

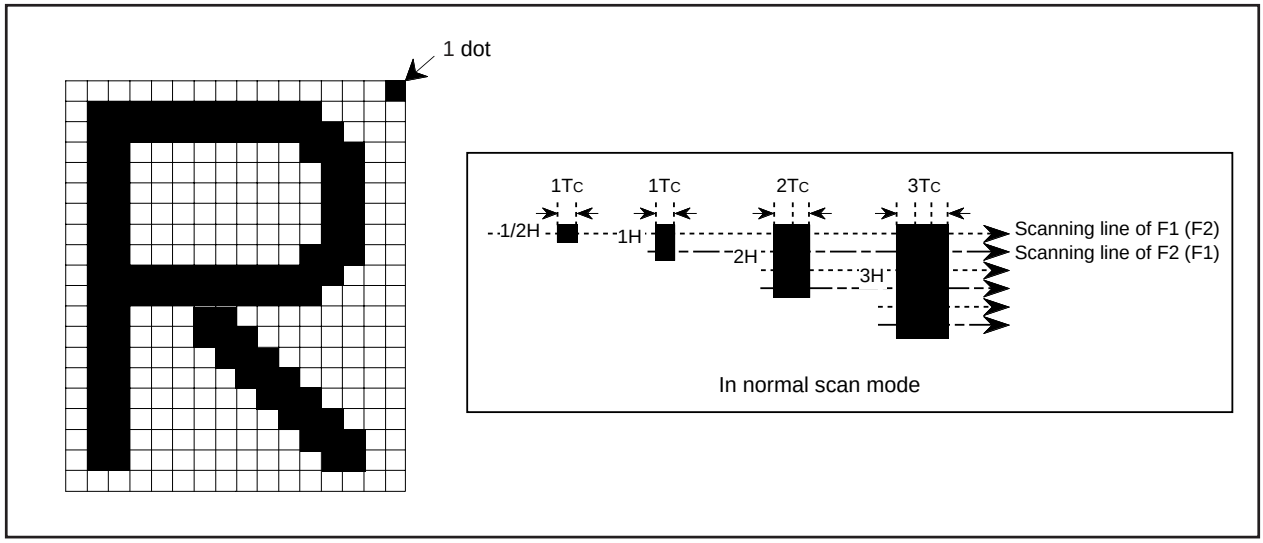


Figure 16.14 Definition of dot sizes

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.

Clock control register 1

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

Symbol: CS1 | Address: 020516 | When reset: 0016

Bit symbol	Bit name	Function	RW
CS10	Clock selection bit	0: Internally generated clock 1: OSC1 clock	RW
CS11	OSC1 oscillating mode selection bits	b2 b1 0 0: Stopped 0 1: Do not set. 1 0: LC oscillating mode 1 1: Ceramic • quartz-crystal oscillating mode	RW
CS12			RW
Reserved bit		Must always be set to “0”	RW
CS14	R, G, B output pin control bit	0: Outputs 8 or 2 gray levels from the R, G and B pins. 1: Converts the 8 gray levels output for R, G and B each into 3-bit digital quantities which are output from the respective ports.	RW
Reserved bit		Must always be set to “1”	RW
Reserved bits		Must always be set to “0”	RW

Figure 16.15 Clock control register 1

Clock control register 2

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

Symbol: CG | Address: $020B16$ | When reset: 0016

Bit symbol	Bit name	Function	RW
Reserved bit		Must always be set to “0”.	RW
CS21	Clock adjustment bit	Set the value which is same as CS27.	RW
Reserved bits		Must always be set to “0”.	RW
CS27	Clock devided bit (Note)	0: Devided in 2 1: No-division	RW

Note: At the time of internal oscillation use for clock display, set this bit to “0” and set this bit the “1” at the time of ceramic resonator use. This bit does not function at the time of LC oscillator use.

Figure 16.16 Clock control register 2

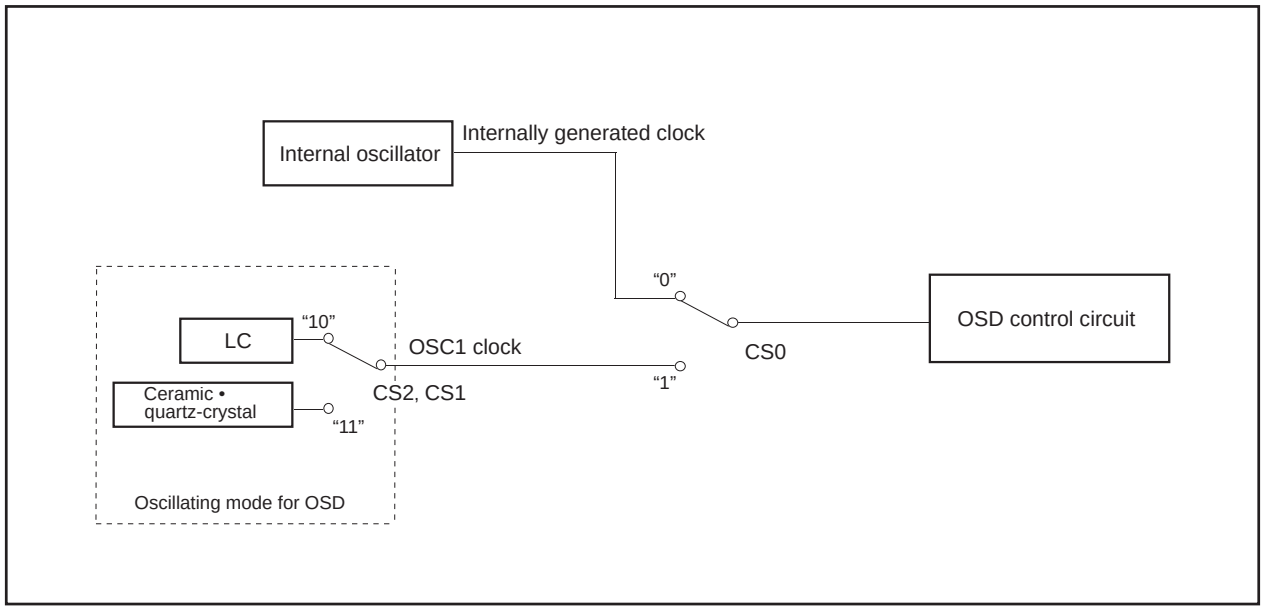


Figure 16.17 Block Diagram of OSD selection circuit

Internal oscillation control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Symbol: DIV0 | Address: 0280 | When reset: 00_{16}

BIt symbol	BIt name	Function	RW
DIV00	Reference clock divide bit	Set "001111_2" or "001102"	RW
DIV01			
DIV02			
DIV03			
DIV04			
DIV05		Set "1" at the time of internal oscillation circuit use, and set "0" at non-uisng.	
Reserved bits		Must always be set to "0"	

Internal oscillation control register 2 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol: DIV1 | Address: 0281 | When reset: 00_{16}

BIt symbol	BIt name	Function	RW
DIV10	Internal oscillation frequency select bit	Internal oscillation frequency =n • 4(N+1)/(m+1)(MHz) N : Values that can be set by DIV16 to DIV10 m : Values that can be set by DIV04 to DIV00 n : Values set by VCO05/04 Set N, m, and n as shown below: N="$3B_{16}$"–"$7F_{16}$", m=06_{16} , n=2 when VCO01="0" N="27_{16}"–"$3B_{16}$", m=07_{16} , n=1 when VCO01="1"	RW
DIV11			
DIV12			
DIV13			
DIV14			
DIV15			
DIV16			
Reserved bit		Must always be set to "0"	

Internal oscillation control register 3 (Note)

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

Symbol: VCO | Address: 0282 | When reset: 00_{16}

BIt symbol	BIt name	Function	RW
VCO00	Internal oscillator operating bit	0 : OFF 1 : ON	RW
VCO01	Internal oscillator operating bit	0 : Selects oscillator for 30 to 65 MHz 1 : Selects oscillator for 20 to 30 MHz	
VCO02	Oscillation characteristic switch bits	Must be fixed to (b3, b2)=(0, 0)	
VCO03			
VCO05/04	Must be fixed to (b5, b4)=(1, 0), When VCO01 is set to "0" Must be fixed to (b5, b4)=(0, 0), When VCO01 is set to "1"		
Reserved bits		Must always be set to "0"	

Note: Since there is a possibility that jitter may occer, do not access these registers during display.

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 0206_{16}). 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.

I/O polarity control register

b7	b6	b5	b4	b3	b2	b1	b0
				0			

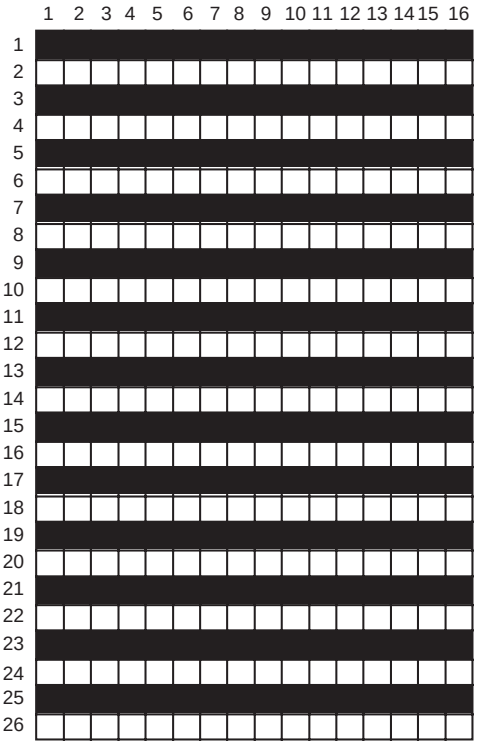
Symbol: POLC
Address: 0206_{16}
When reset: 80_{16}

Bit symbol	Bit name	Function	RW
POLC0	HSYNC input polarity switch bit	0 : Positive polarity input 1 : Negative polarity input	RW
POLC1	VSYNC input polarity switch bit	0 : Positive polarity input 1 : Negative polarity input	RW
POLC2	R, G, B output polarity switch bit	0 : Positive polarity output 1 : Negative polarity output	RW
Reserved bit		Must always be set to “0.”	RW
POLC4	OUT1 output polarity switch bit	0 : Positive polarity output 1 : Negative polarity output	RW
POLC5	OUT2 output polarity switch bit	0 : Positive polarity output 1 : Negative polarity output	RW
POLC6	Display dot line selection bit	0 : “□” at even field “■” at odd field 1 : “■” at even field “□” at odd field	RW
POLC7	Field determination flag	0 : Even field 1 : Odd field	RO

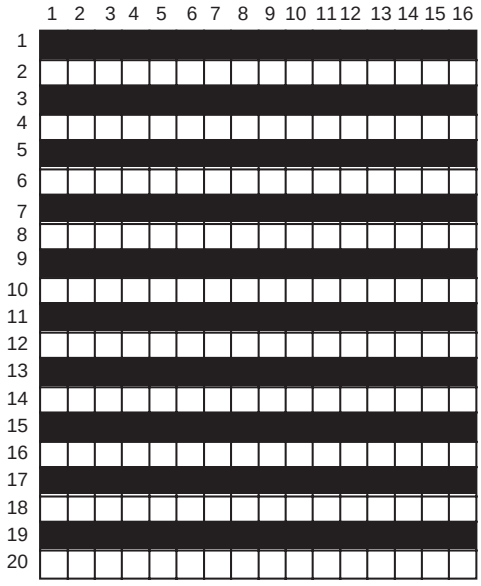
Figure 16.19 I/O polarity control register

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 microcomputer Upper : VSYNC signal Lower : VSYNC control signal in microcomputer	(n - 1) field (Odd-numbered) T1 0.5 to 0.1 [µs] at f(BCLK) = 10 MHz	Odd			
	(n) field (Even-numbered) T2	Even	0 (T2 > T1)	0	Dot line 1 □
				1	Dot line 0 ■
	(n + 1) field (Odd-numbered) T3	Odd	1 (T3 < T2)	0	Dot line 0 ■
				1	Dot line 1 □



CC mode · CDOSD mode



OSDS mode

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 × 20-dot font and 24 × 32-dot font
- OSDL disable mode
  16 × 20-dot font

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

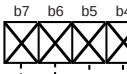
The conditions for each OSDL enable/disable mode are shown in Figure 16.22.

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 × 20-dot fonts, whereas the character codes 100_{16} through $1FF_{16}$ are turned to 24 × 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 × 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 × 26-dot fonts. Of these, however, CD codes $3F_{16}$ and 40_{16} cannot be used.

OSD control register 4



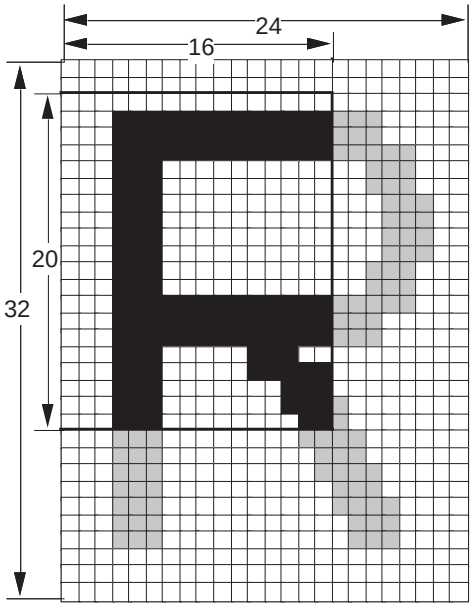
Symbol: OC4 | Address: $025F_{16}$ | When reset: $XXXXX00_2$

Bit symbol	Bit name	Function	R	W
OC40	OSDL mode selection bit	0 : OSDL enable mode 1 : OSDL disable mode	O	O
OC41	Number of horizontal display characters selection bit	0 : 32 characters for each block (32-character mode) 1 : 42 characters for each block (42-character mode)	O	O
Nothing is assigned. In an attempt to write to these bits, write “0.” The value, if read, turns out to be “0.”			—	—

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.

Display mode & character code / 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")		
Display mode		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)	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 × 32 font is displayed.

2: In OSDL disable mode, character codes "280_{16}" to "$2FF_{16}$" are used in OSDS/P mode (no border).

3: As setting this make output of font data indeterminate, do not use. However, "$3FE_{16}$" and "$3FF_{16}$" 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 8400_{16} to $8EFF_{16}$)

The OSD RAM for character is allocated at addresses 8400_{16} to $8EFF_{16}$, 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 8400_{16}, write color code 1 at 8401_{16}, and write color code 2 at 8480_{16}. The structure of the OSD RAM is shown in Figure 16.23.

Note : For blocks of the following dot sizes, the 3nth (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 × 1/2H or 1.5Tc × 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 3nth 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 back ground is displayed in the 22nd's character area. When not displaying this back-ground, 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).

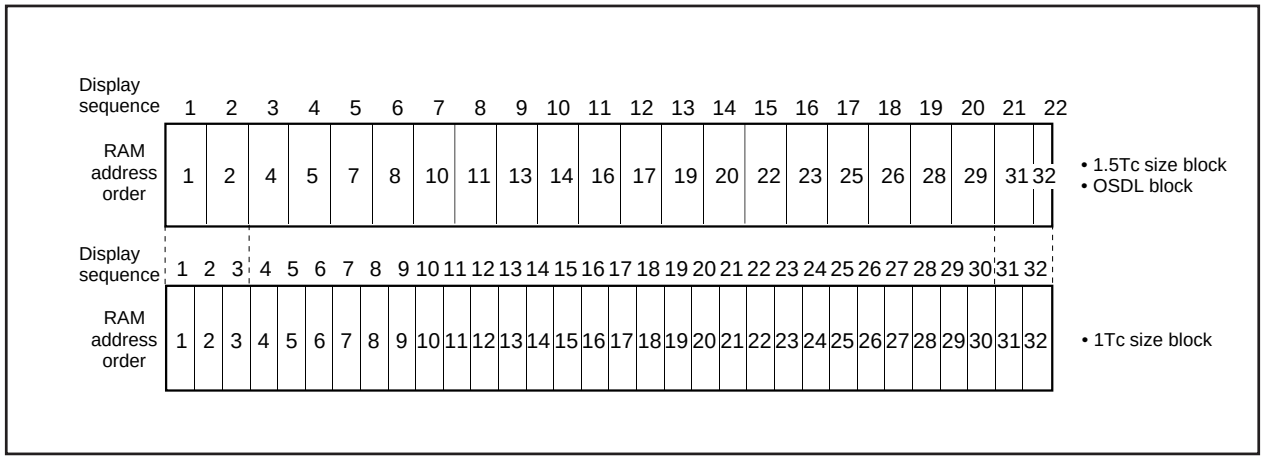


Figure 16.23 RAM data for 3rd character (in 32-character mode)

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 : 31st character	8402_{16} : $843C_{16}$	8403_{16} : $843D_{16}$	8482_{16} : $84BC_{16}$
	32nd character	$843E_{16}$	$843F_{16}$	$84BE_{16}$
Block 2	1st character	8440_{16}	8441_{16}	$84C0_{16}$
	2nd character : 31st character	8442_{16} : $847C_{16}$	8443_{16} : $847D_{16}$	$84C2_{16}$: $84FC_{16}$
	32nd character	$847E_{16}$	$847F_{16}$	$84FE_{16}$
Block 3	1st character	8500_{16}	8501_{16}	8580_{16}
	2nd character : 31st character	8502_{16} : $853C_{16}$	8503_{16} : $853D_{16}$	8582_{16} : $85BC_{16}$
	32nd character	$853E_{16}$	$853F_{16}$	$85BE_{16}$
Block 4	1st character	8540_{16}	8541_{16}	$85C0_{16}$
	2nd character : 31st character	8542_{16} : $857C_{16}$	8543_{16} : $857D_{16}$	$85C2_{16}$: $85FC_{16}$
	32nd character	$857E_{16}$	$857F_{16}$	$85FE_{16}$
Block 5	1st character	8600_{16}	8601_{16}	8680_{16}
	2nd character : 31st character	8602_{16} : $863C_{16}$	8603_{16} : $863D_{16}$	8682_{16} : $86BC_{16}$
	32nd character	$863E_{16}$	$863F_{16}$	$86BE_{16}$
Block 6	1st character	8640_{16}	8641_{16}	$86C0_{16}$
	2nd character : 31st character	8642_{16} : $867C_{16}$	8643_{16} : $867D_{16}$	$86C2_{16}$: $86FC_{16}$
	32nd character	$867E_{16}$	$867F_{16}$	$86FE_{16}$
Block 7	1st character	8700_{16}	8701_{16}	8780_{16}
	2nd character : 31st character	8702_{16} : $873C_{16}$	8703_{16} : $873D_{16}$	8782_{16} : $87BC_{16}$
	32nd character	$873E_{16}$	$873F_{16}$	$87BE_{16}$
Block 8	1st character	8740_{16}	8741_{16}	$87C0_{16}$
	2nd character : 31st character	8742_{16} : $877C_{16}$	8743_{16} : $877D_{16}$	$87C2_{16}$: $87FC_{16}$
	32nd character	$877E_{16}$	$877F_{16}$	$87FE_{16}$
Block 9	1st character	8800_{16}	8801_{16}	8880_{16}
	2nd character : 31st character	8802_{16} : $883C_{16}$	8803_{16} : $883D_{16}$	8882_{16} : $88BC_{16}$
	32nd character	$883E_{16}$	$883F_{16}$	$88BE_{16}$
Block 10	1st character	8840_{16}	8841_{16}	$88C0_{16}$
	2nd character : 31st character	8842_{16} : $887C_{16}$	8843_{16} : $887D_{16}$	$88C2_{16}$: $88FC_{16}$
	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 : 31st character	8902_{16} : $893C_{16}$	8903_{16} : $893D_{16}$	8982_{16} : $89BC_{16}$
	32nd character	$893E_{16}$	$893F_{16}$	$89BE_{16}$
Block 12	1st character	8940_{16}	8941_{16}	$89C0_{16}$
	2nd character : 31st character	8942_{16} : $897C_{16}$	8943_{16} : $897D_{16}$	$89C2_{16}$: $89FC_{16}$
	32nd character	$897E_{16}$	$897F_{16}$	$89FE_{16}$
Block 13	1st character	$8A00_{16}$	$8A01_{16}$	$8A80_{16}$
	2nd character : 31st character	$8A02_{16}$: $8A3C_{16}$	$8A03_{16}$: $8A3D_{16}$	$8A82_{16}$: $8ABC_{16}$
	32nd character	$8A3E_{16}$	$8A3F_{16}$	$8ABE_{16}$
Block 14	1st character	$8A40_{16}$	$8A41_{16}$	$8AC0_{16}$
	2nd character : 31st character	$8A42_{16}$: $8A7C_{16}$	$8A43_{16}$: $8A7D_{16}$	$8AC2_{16}$: $8AFC_{16}$
	32nd character	$8A7E_{16}$	$8A7F_{16}$	$8AFE_{16}$
Block 15	1st character	$8B00_{16}$	$8B01_{16}$	$8B80_{16}$
	2nd character : 31st character	$8B02_{16}$: $8B3C_{16}$	$8B03_{16}$: $8B3D_{16}$	$8B82_{16}$: $8BBC_{16}$
	32nd character	$8B3E_{16}$	$8B3F_{16}$	$8BBE_{16}$
Block 16	1st character	$8B40_{16}$	$8B41_{16}$	$8BC0_{16}$
	2nd character : 31st character	$8B42_{16}$: $8B7C_{16}$	$8B43_{16}$: $8B7D_{16}$	$8BC2_{16}$: $8BF0_{16}$
	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 : 39th character	$8C02_{16}$: $8C0C_{16}$	$8C03_{16}$: $8C0D_{16}$	$8C82_{16}$: $8C8C_{16}$
	40th character	$8C0E_{16}$	$8C0F_{16}$	$8C8E_{16}$
	41st character	$8E00_{16}$	$8E01_{16}$	$8E80_{16}$
	42nd character	$8E02_{16}$	$8E03_{16}$	$8E82_{16}$
Block 2	33rd character	$8C10_{16}$	$8C11_{16}$	$8C90_{16}$
	34th character : 39th character	$8C12_{16}$: $8C1C_{16}$	$8C13_{16}$: $8C1D_{16}$	$8C92_{16}$: $8C9C_{16}$
	40th character	$8C1E_{16}$	$8C1F_{16}$	$8C9E_{16}$
	41st character	$8E08_{16}$	$8E09_{16}$	$8E88_{16}$
	42nd character	$8E0A_{16}$	$8E0B_{16}$	$8E8A_{16}$
Block 3	33rd character	$8C20_{16}$	$8C21_{16}$	$8CA0_{16}$
	34th character : 39th character	$8C22_{16}$: $8C2C_{16}$	$8C23_{16}$: $8C2D_{16}$	$8CA2_{16}$: $8CAC_{16}$
	40th character	$8C2E_{16}$	$8C2F_{16}$	$8CAE_{16}$
	41st character	$8E10_{16}$	$8E11_{16}$	$8E90_{16}$
	42nd character	$8E12_{16}$	$8E13_{16}$	$8E92_{16}$
Block 4	33rd character	$8C30_{16}$	$8C31_{16}$	$8CB0_{16}$
	34th character : 39th character	$8C32_{16}$: $8C3C_{16}$	$8C33_{16}$: $8C3D_{16}$	$8CB2_{16}$: $8CBC_{16}$
	40th character	$8C3E_{16}$	$8C3F_{16}$	$8CBE_{16}$
	41st character	$8E18_{16}$	$8E19_{16}$	$8E98_{16}$
	42nd character	$8E1A_{16}$	$8E1B_{16}$	$8E9A_{16}$
Block 5	33rd character	$8C40_{16}$	$8C41_{16}$	$8CC0_{16}$
	34th character : 39th character	$8C42_{16}$: $8C4C_{16}$	$8C43_{16}$: $8C4D_{16}$	$8CC2_{16}$: $8CCC_{16}$
	40th character	$8C4E_{16}$	$8C4F_{16}$	$8CCE_{16}$
	41st character	$8E20_{16}$	$8E21_{16}$	$8EA0_{16}$
	42nd character	$8E22_{16}$	$8E23_{16}$	$8EA2_{16}$
Block 6	33rd character	$8C50_{16}$	$8C51_{16}$	$8CD0_{16}$
	34th character : 39th character	$8C52_{16}$: $8C5C_{16}$	$8C53_{16}$: $8C5D_{16}$	$8CD2_{16}$: $8CDC_{16}$
	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 : 39th character	$8C62_{16}$: $8C6C_{16}$	$8C63_{16}$: $8C6D_{16}$	$8CE2_{16}$: $8CEC_{16}$
	40th character	$8C6E_{16}$	$8C6F_{16}$	$8CEE_{16}$
	41st character	$8E30_{16}$	$8E31_{16}$	$8EB0_{16}$
	42nd character	$8E32_{16}$	$8E33_{16}$	$8EB2_{16}$

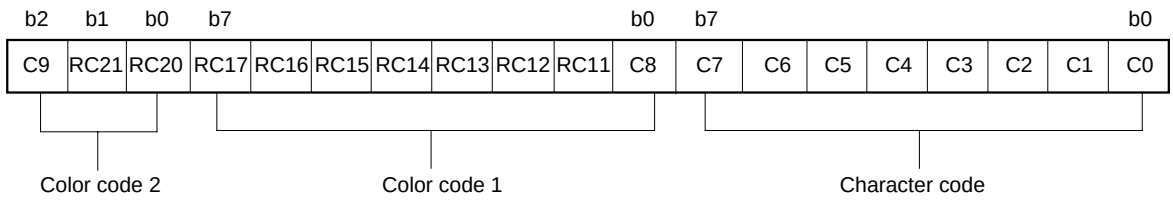
Table 16.6 Contents of OSD RAM (33rd to 42nd character) (continued)

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 : 39th character	8C72$_{16}$: 8C7C$_{16}$	8C73$_{16}$: 8C7D$_{16}$	8CF2$_{16}$: 8CFC$_{16}$
	40th character	8C7E$_{16}$	8C7F$_{16}$	8CFE$_{16}$
	41st character	8E38$_{16}$	8E39$_{16}$	8EB8$_{16}$
	42nd character	8E3A$_{16}$	8E3B$_{16}$	8EBA$_{16}$
Block 9	33rd character	8D00$_{16}$	8D01$_{16}$	8D80$_{16}$
	34th character : 39th character	8D02$_{16}$: 8D0C$_{16}$	8D03$_{16}$: 8D0D$_{16}$	8D82$_{16}$: 8D8C$_{16}$
	40th character	8D0E$_{16}$	8D0F$_{16}$	8D8E$_{16}$
	41st character	8E40$_{16}$	8E41$_{16}$	8EC0$_{16}$
	42nd character	8E42$_{16}$	8E43$_{16}$	8EC2$_{16}$
Block 10	33rd character	8D10$_{16}$	8D11$_{16}$	8D90$_{16}$
	34th character : 39th character	8D12$_{16}$: 8D1C$_{16}$	8D13$_{16}$: 8D1D$_{16}$	8D92$_{16}$: 8D9C$_{16}$
	40th character	8D1E$_{16}$	8D1F$_{16}$	8D9E$_{16}$
	41st character	8E48$_{16}$	8E49$_{16}$	8EC8$_{16}$
	42nd character	8E4A$_{16}$	8E4B$_{16}$	8ECA$_{16}$
Block 11	33rd character	8D20$_{16}$	8D21$_{16}$	8DA0$_{16}$
	34th character : 39th character	8D22$_{16}$: 8D2C$_{16}$	8D23$_{16}$: 8D2D$_{16}$	8DA2$_{16}$: 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$_{16}$
Block 12	33rd character	8D30$_{16}$	8D31$_{16}$	8DB0$_{16}$
	34th character : 39th character	8D32$_{16}$: 8D3C$_{16}$	8D33$_{16}$: 8D3D$_{16}$	8DB2$_{16}$: 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$_{16}$
Block 13	33rd character	8D40$_{16}$	8D41$_{16}$	8DC0$_{16}$
	34th character : 39th character	8D42$_{16}$: 8D4C$_{16}$	8D43$_{16}$: 8D4D$_{16}$	8DC2$_{16}$: 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 : 39th character	8D52$_{16}$: 8D5C$_{16}$	8D53$_{16}$: 8D5D$_{16}$	8DD2$_{16}$: 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$_{16}$
	42nd character	8E72$_{16}$	8E73$_{16}$	8EF2$_{16}$
Block 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$_{16}$
	41st character	8E78$_{16}$	8E79$_{16}$	8EF8$_{16}$
	42nd character	8E7A$_{16}$	8E7B$_{16}$	8EFA$_{16}$

Blocks 1 to 16



Bit	CC mode		OSDS/L/P mode		CDOSD mode	
	Bit name	Function	Bit name	Function	Bit name	Function
C0 C1 C2 C3 C4 C5 C6 C7	Character code (Low-order 9 bits)	Specify character code in OSD ROM	Character code (Low-order 9 bits)	Specify character code in OSD ROM	CD code (7 bits)	Specify character code in OSD ROM (color dot)
C8					Not used	———
RC11	Character: Color palette selection bit 0	Specify color palette for character (See note 3)	Character: Color palette selection bit 0	Specify color palette for character (See note 3)	Color palette selection bit 4	0: Color palette set 0 1: Color palette set 1
RC12	Character: Color palette selection bit 1		Character: Color palette selection bit 1			
RC13	Character: Color palette selection bit 2		Character: Color palette selection bit 2		Dot color: Color palette selection bit 0	Specify a dot which selects color palette 0 by OSD ROM (See note 4)
RC14	Italic control	0: Italic OFF 1: Italic ON	Character: Color palette selection bit 3		Dot color: Color palette selection bit 1	
RC15	Flash control	0: Flash OFF 1: Flash ON	Character background: Color palette selection bit 0	Specify color palette for character (See note 3)	Dot color: Color palette selection bit 2	
RC16	Underline control	0: Underline OFF 1: Underline ON	Character background: Color palette selection bit 1		Dot color: Color palette selection bit 3	
RC17	OUT2 output control	0: OUT2 output OFF 1: OUT2 output ON	OUT2 output control	0: OUT2 output OFF 1: OUT2 output ON	OUT2 output control	0: OUT2 output OFF 1: OUT2 output ON
RC20	Character background: Color palette selection bit 0	Specify color palette for background (See note 3)	Character background: Color palette selection bit 2	Specify color palette for background (See note 3)	Not used	———
RC21	Character background: Color palette selection bit 1		Character background: Color palette selection bit 3			
C9	Character code (High-order 1 bit)	Specify character code in OSD ROM	Character code (High-order 1 bit)	Specify character code in OSD ROM	Not used	———

Notes 1: Read value of bits 3 to 7 of the color code 2 is undefined.

2: For “not used” bits, the write value is read.

3: Refer to Figure 16.25.

4: Only in CDOSD mode, a dot which selects color palette 0 is colored to the color palette set and color palette by RC13 to RC16 of OSD RAM in character units. When the character size is 1.5Tc × 1H or 1.5Tc × 1/2H, however, set RCI3 to RC16 and RC17 of all characters (including the 3nth character) within the same block to the same value.

Figure 16.24 Structure of OSD RAM

(3) OSD RAM (OSD RAM for SPRITE, addresses 8000_{16} to $83E7_{16}$)

The OSD RAM for SPRITE fonts 1 and 2, consisting of 4 planes for each font, is assigned to addresses 8000_{16} to $83E7_{16}$. 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 paleltte selection bit 3)				Plane 2 (Color paleltte selection bit 2)				Plane 1 (Color paleltte selection bit 1)				Plane 0 (Color paleltte selection bit 0)			
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 paleltte selection bit 3)				Plane 2 (Color paleltte selection bit 2)				Plane 1 (Color paleltte selection bit 1)				Plane 0 (Color paleltte selection bit 0)			
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	$82E6_{16}$	$82E7_{16}$	$83E6_{16}$	$83E7_{16}$	$82A6_{16}$	$82A7_{16}$	$83A6_{16}$	$883A7_{16}$	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 0207_{16}). 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.

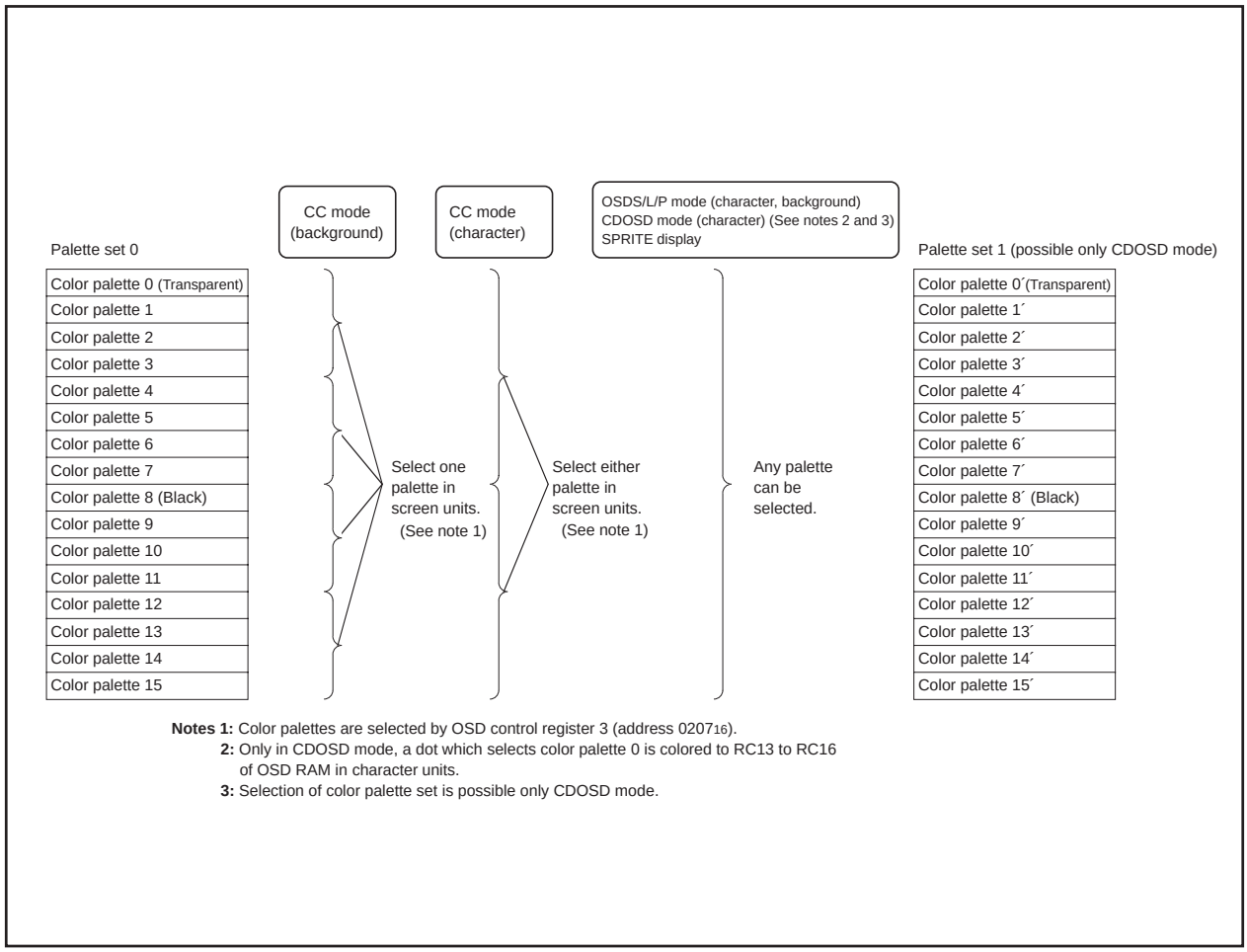
2: Color palette 0 (transparent) and the transparent setting of other color palettes will differ. When there are multiple layers overlapping (on top of each other, piled up), and the priority layer is color palette 0 (transparent), the bottom layer is displayed, but if the priority layer is the transparent 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 0207_{16}). 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.



Notes 1: Color palettes are selected by OSD control register 3 (address 0207_{16}).
2: Only in CDOSD mode, a dot which selects color palette 0 is colored to RC13 to RC16 of OSD RAM in character units.
3: Selection of color palette set is possible only CDOSD mode.

Figure 16.25 Color palette selection

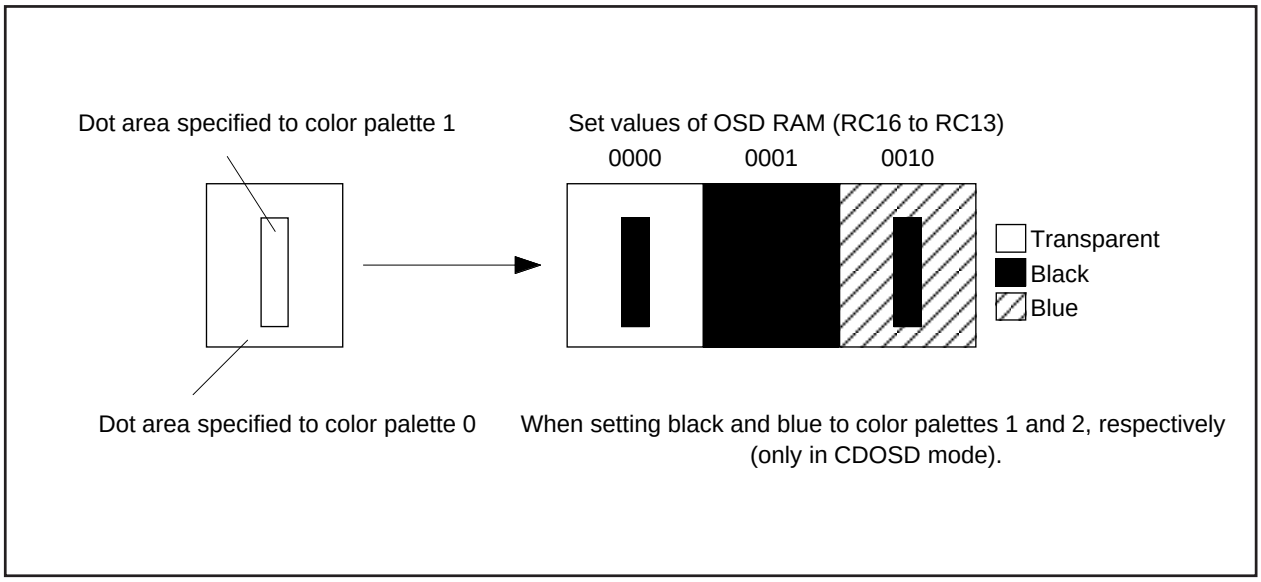


Figure 16.26 Set of color palette 0 or 8 in CDOSD mode

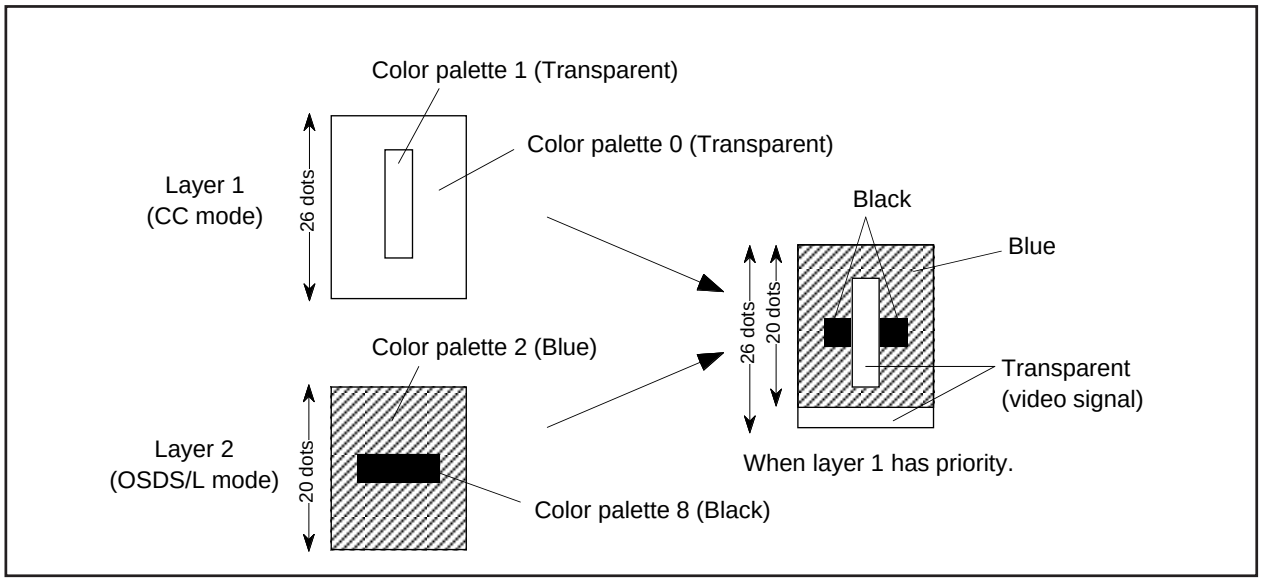


Figure 16.27 Difference between color palette 0 (transparent) and transparent setting of other color palettes

OSD control register 3

b7	b6	b5	b4	b3	b2	b1	b0
				0			

Symbol	Address	When reset
OC3	0207_{16}	00_{16}

Bit symbol			RW
OC30	CC mode character color selection bit	0: Color palettes 0 to 7 1: Color palettes 8 to 15	RW
OC31	CC mode character background color selection bits (See note)	b2 0 0: Color palettes 0 to 3 0 1: Color palettes 4 to 7 1 0: Color palettes 8 to 11 1 1: Color palettes 12 to 15	RW
OC32			RW
Reserved bit		Must always be set to “0”	RW
OC34	Flash cycle selection bit	0: 1 cycle = VSYNC cycle × 32 1: 1 cycle = VSYNC cycle × 64	RW
OC35	OSD mode window control bit	0: Window OFF 1: Window ON	RW
OC36	CC mode window control bit	0: Window OFF 1: Window ON	RW
OC37	CDOSD mode window control bit	0: Window OFF 1: Window ON	RW

Note: Color palette 8 is always selected for solid space (when OUT1 output is selected), regardless of value of this register.

Figure 16.28 OSD control register 3

Color palette register i

Symbol	Addresses	When reset
CRi (i = 1 to 7)	Even addresses within addresses 0240$_{16}$ to 024D$_{16}$, Odd addresses within addresses 0240$_{16}$ to 024D$_{16}$	Indeterminate
CRi (i = 9 to 15)	Even addresses within addresses 024E$_{16}$ to 025B$_{16}$, Odd addresses within addresses 024E$_{16}$ to 025B$_{16}$	Indeterminate

(b15) (b8)

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

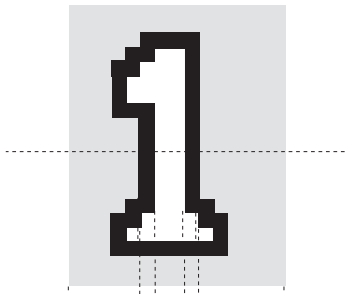
Bit symbol	Bit name	Function	RW
CRi_2 to CRi_0	R signal output control bits	b2 b1 b0 0 0 0 : V_{SS} 0 0 1 : 1/7V 0 1 0 : 2/7V 0 1 1 : 3/7V 1 0 0 : 4/7V 1 0 1 : 5/7V 1 1 0 : 6/7V 1 1 1 : 7/7V	RW
Nothing is assined. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—
CRi_6 to CRi_4	G signal output control bits	b6 b5 b4 0 0 0 : V_{SS} 0 0 1 : 1/7V 0 1 0 : 2/7V 0 1 1 : 3/7V 1 0 0 : 4/7V 1 0 1 : 5/7V 1 1 0 : 6/7V 1 1 1 : 7/7V	RW
Nothing is assined. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—
CRi_10 to CRi_8	B signal output control bits	b2 b1 b0 0 0 0 : V_{SS} 0 0 1 : 1/7V 0 1 0 : 2/7V 0 1 1 : 3/7V 1 0 0 : 4/7V 1 0 1 : 5/7V 1 1 0 : 6/7V 1 1 1 : 7/7V	RW
Nothing is assined. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—
CRi_12	OUT1 signal output control bit	0: No output 1: Output	RW
Nothing is assined. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—

Note 1: When set up color palette of the palette set 1, set up the color palette register i after setting the bit 6 of an extended register (address 02D5$_{16}$) as 1.
The color palette register of the palette set 1 cannot be read out.

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.



	Conditions				
	OUT2 output control (RC 17 of OSD RAM)	Border output (See note 1)	OUT1 signal output control bit (See note 2) bit12(CRi12) of color pallet register i		Output waveform
			Background	Character	
OUT1 signal	×	No output	0	0	H L
				1	H L
			1	0	H L
				1	H L
		Output (See note 1)	0	0	H L
				1	H L
			1	0	H L
				1	H L
OUT2 signal	0	×	×	×	H L
	1	×	×	×	H L

Notes 1: This control is only valid in the OSDS/P mode. It is invalid in CC/CDOSD/OSDL mode .
2: In the CDOSD mode, coloring is performed for each dot. Accordingly, OUT1 outputs to dots which bit 12 (CRi12) of the color pallet register i is set to "0."
3: OUT2 cannot be output in sprite OSD.
4: × is an arbitrary value.

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 × 24 ≈ 400 ms (at flash ON)
- VSYNC cycle × 8 ≈ 133 ms (at flash OFF)

■ When bit 4 = “1”
- VSYNC cycle × 48 ≈ 800 ms (at flash ON)
- VSYNC cycle × 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 × 5 steps is displayed ; when the pre-divide ratio = 2, the italic character with slant of 1/2 dot × 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.

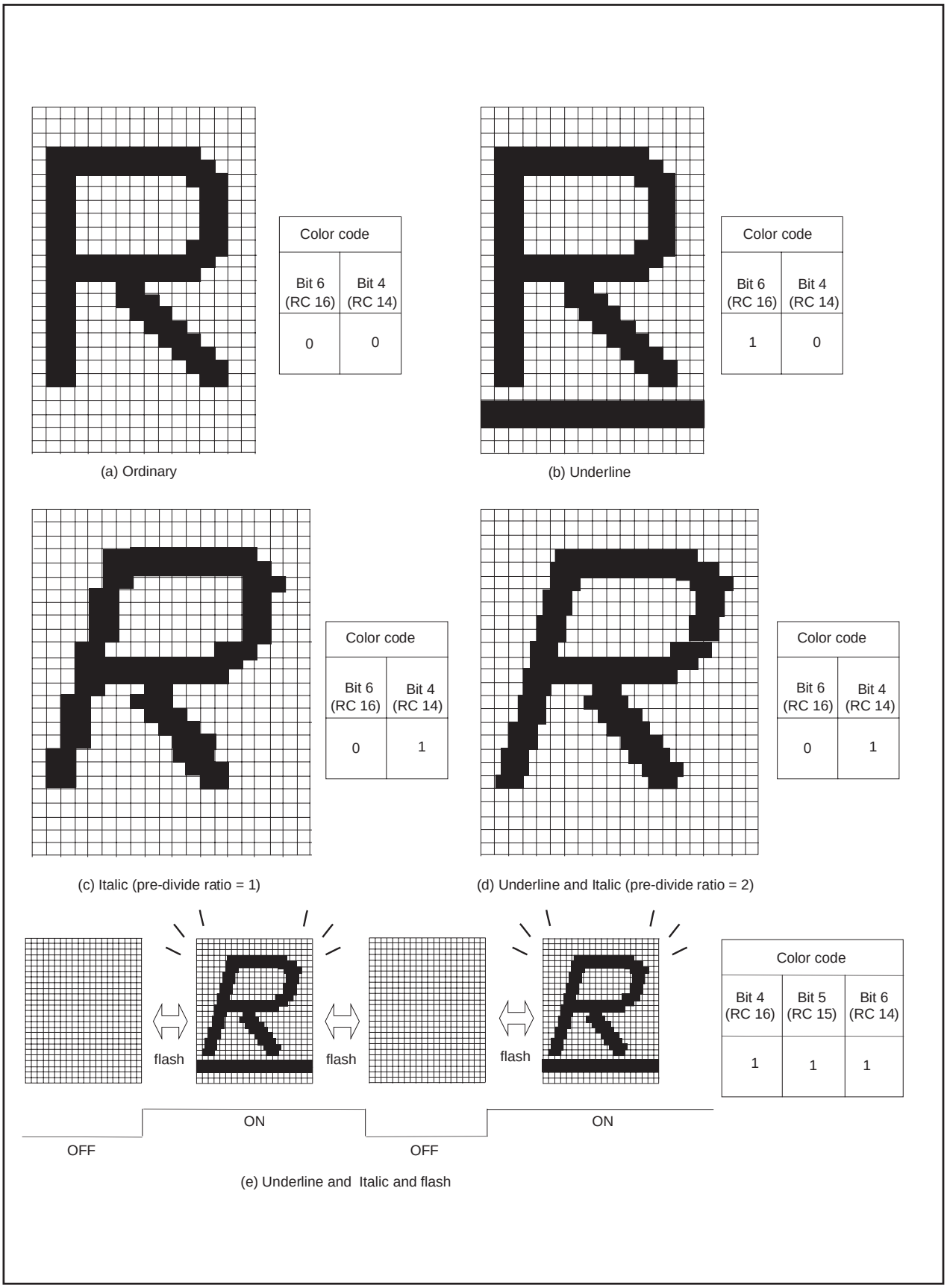


Figure 16.31 Example of attribute display (in CC mode)

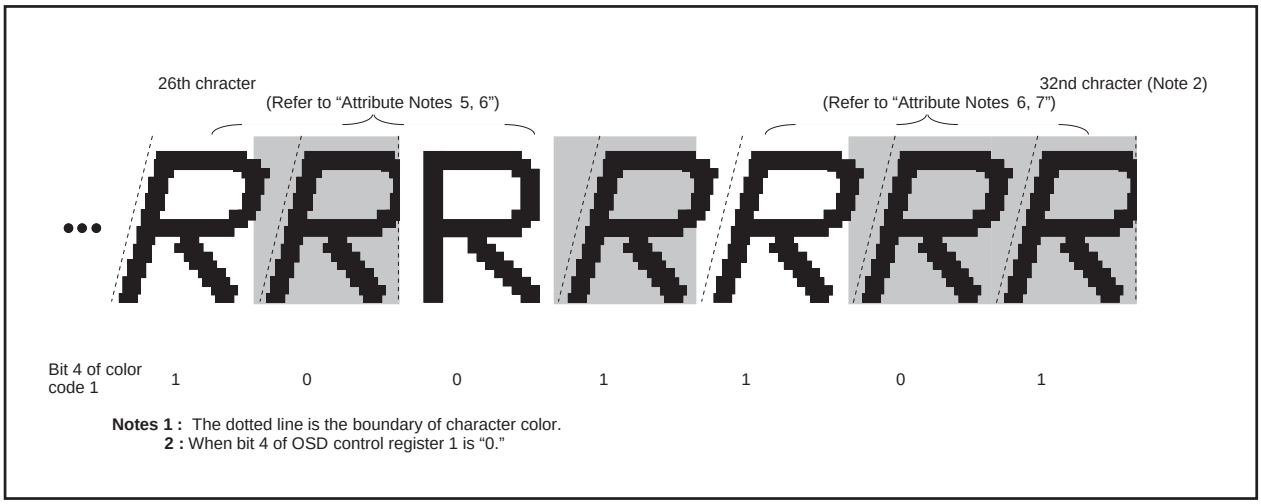


Figure 16.32 Example of italic display

(4) Border

The border is output in the OSDS/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 (block). 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 back ground, 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).

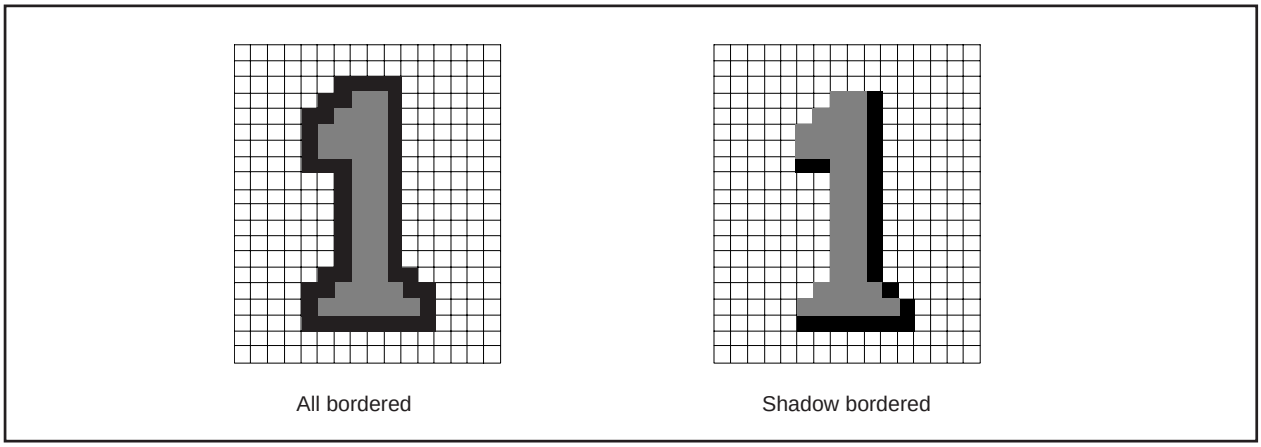
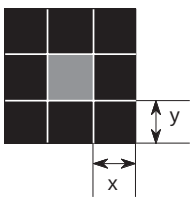


Figure 16.33 Example of border display



Scan mode / Vertical dot size of character font / Border dot size	Normal scan mode		Bi-scan mode
	1/2H	1H, 2H, 3H	1/2H, 1H, 2H, 3H
Horizontal size (x)	1Tc (OSD clock cycle divided in pre-divide circuit) 1.5Tc when selecting 1.5Tc for character size.		
Vertical size (y)	1/2H	1H	1H

Figure 16.34 Horizontal and vertical size of border

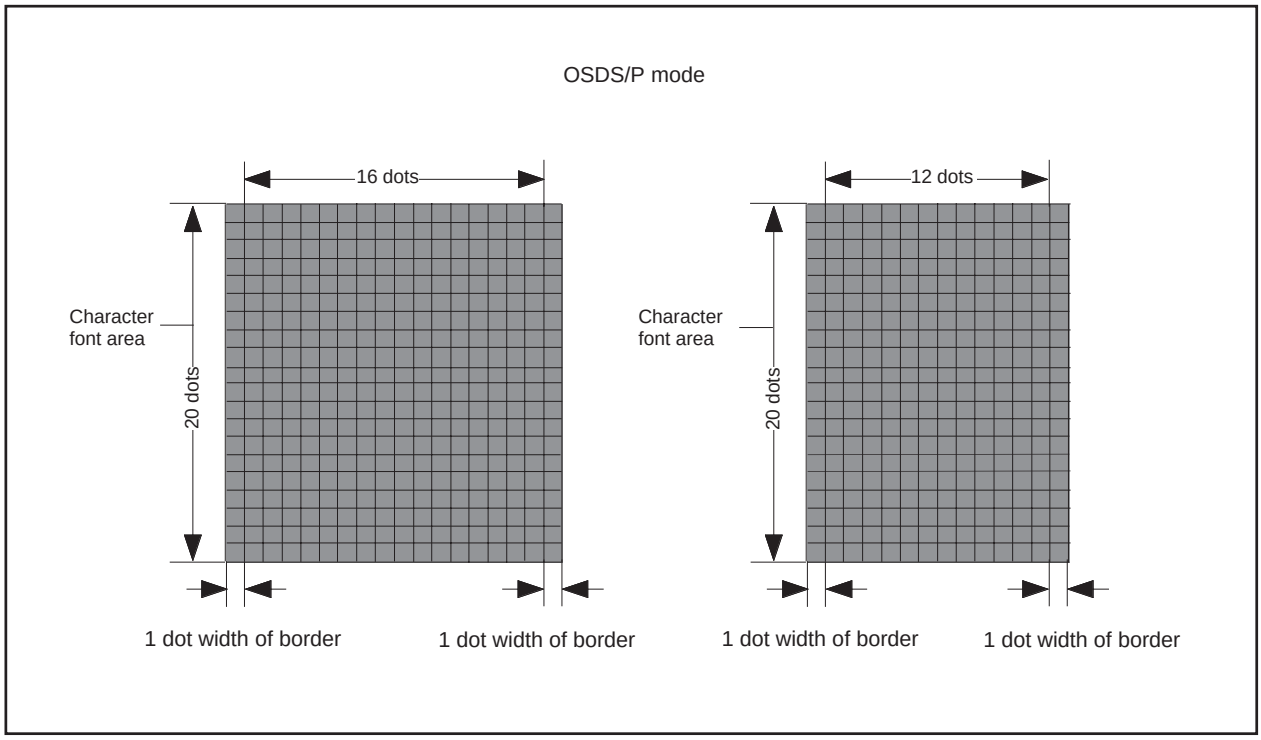


Figure 16.35 Border area

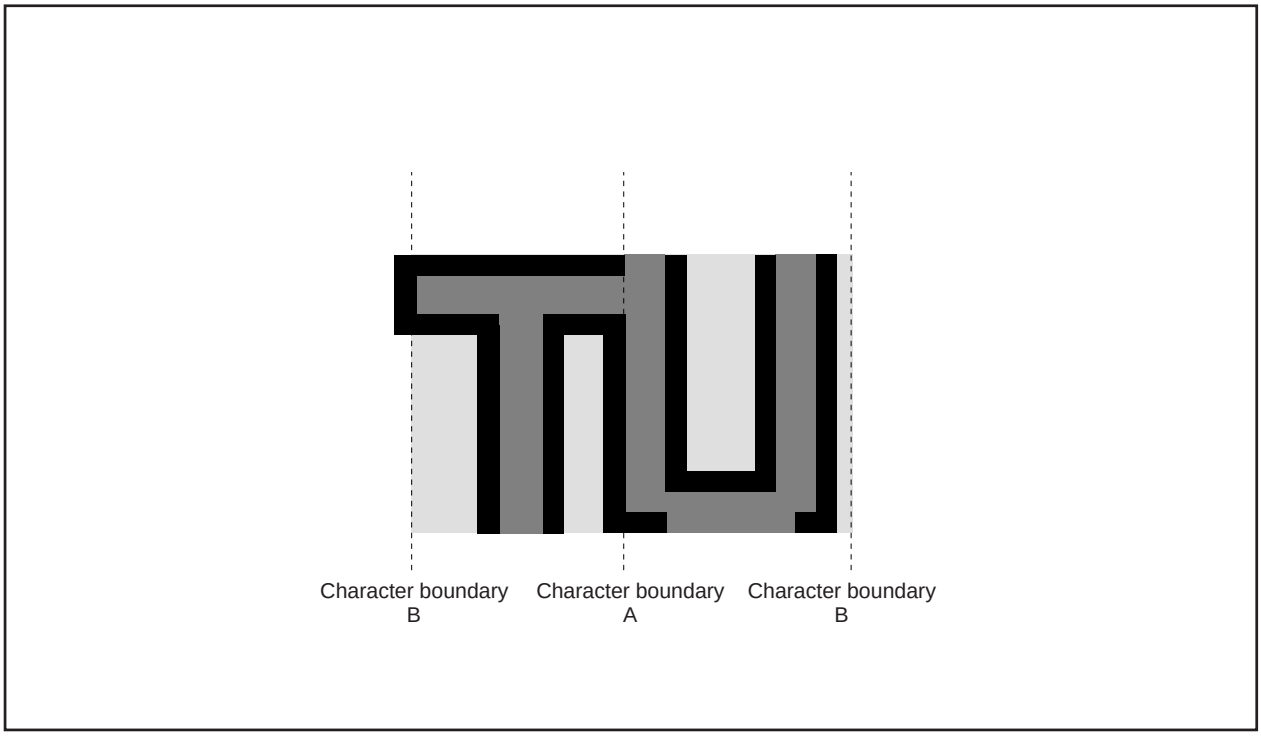


Figure 16.36 Border priority

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 "009_{16} "

•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 "009_{16}," 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

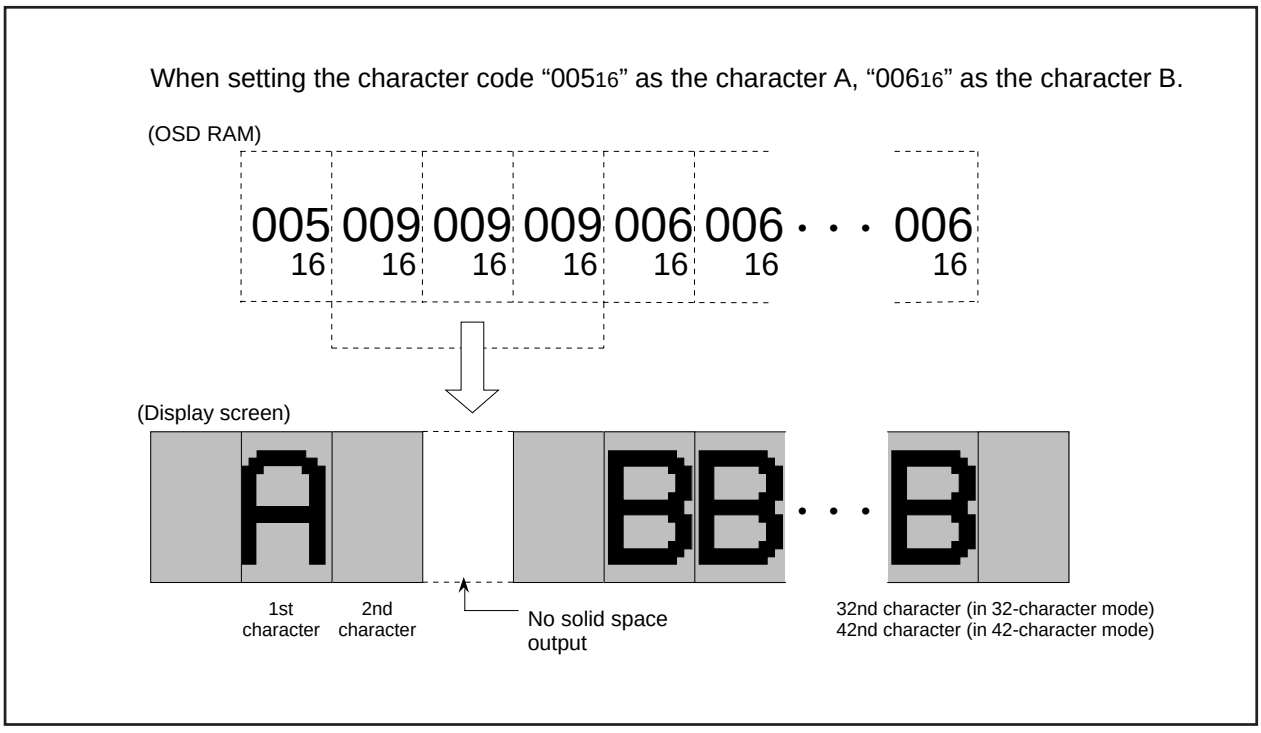


Figure 16.37 Display screen example of automatic solid space

Particular OSD Mode Block

This function can display with mixing the fonts below within the OSDP 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.

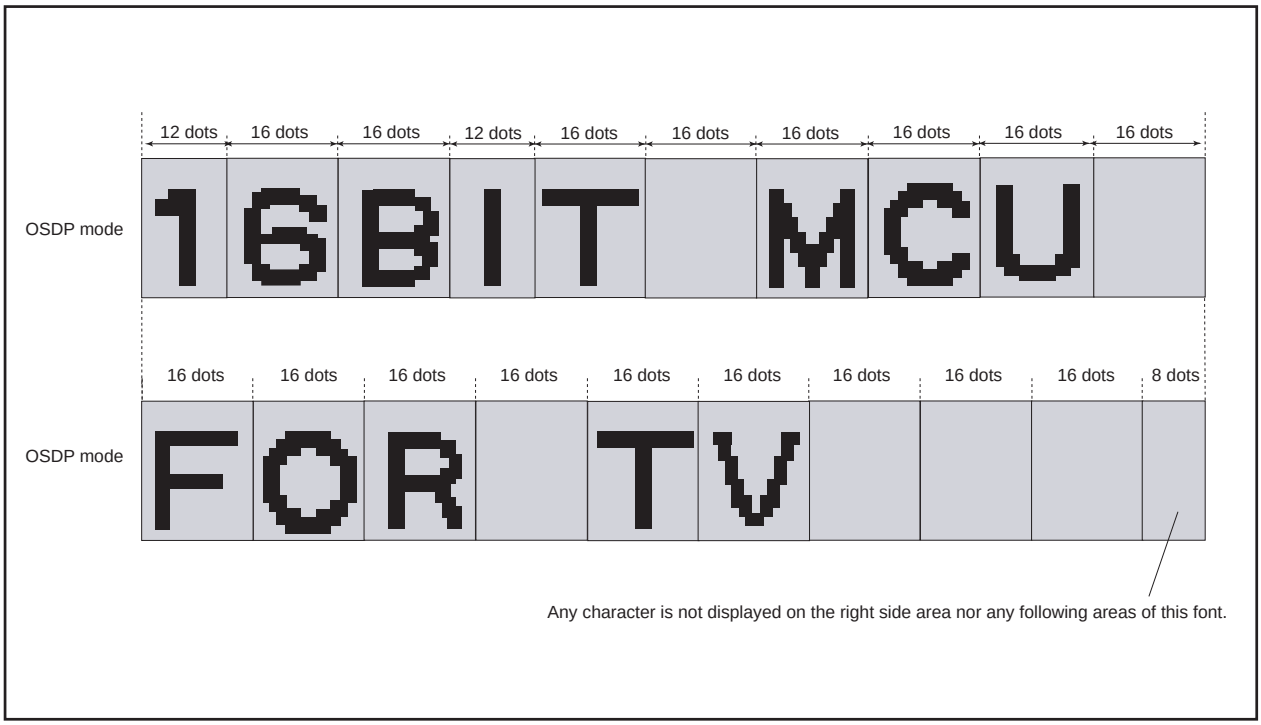
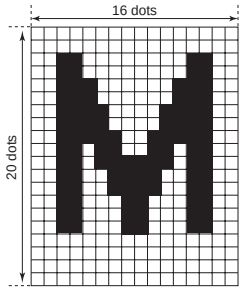
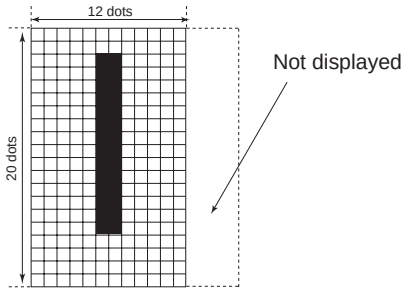
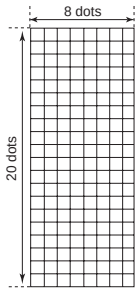
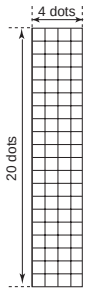


Figure 16.38 Display example of OSD mode block

Table 16.11 Corresponding between character codes and display fonts

Character code	Display fonts	Notes
0001_{16} to $0EF_{16}$, 100_{16} to $2FF_{16}$ (except 100_{16}, 180_{16}, 200_{16}, 280_{16})		
$0F0_{16}$ to $0FD_{16}$		• The left 12-dot part (16 × 12 dots) of set font is displayed. • In CC and OSDS modes, entire part (16 × 20 dots) of set font is displayed.
$3FE_{16}$		• 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.
$3FF_{16}$		• 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 0210_{16} to $021F_{16}$), 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)).

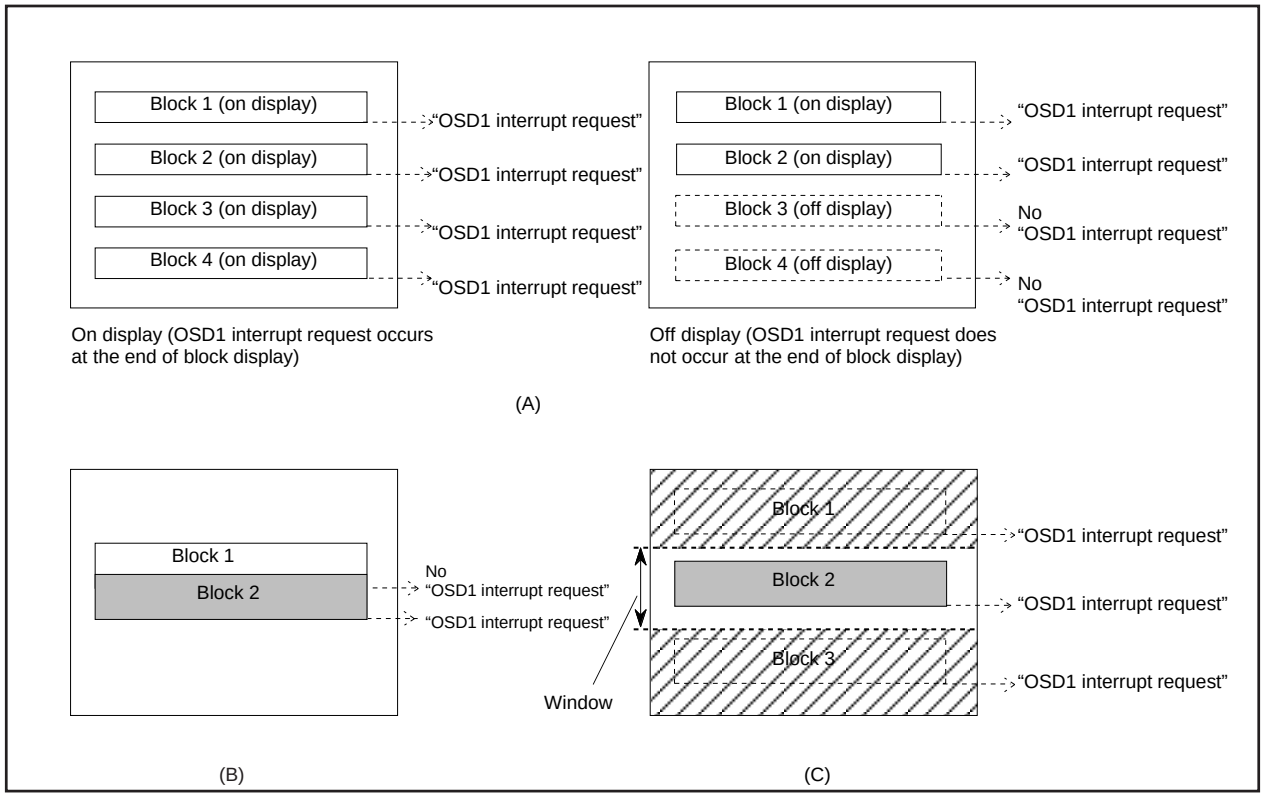


Figure 16.39 Note on occurrence of OSD1 interrupt

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 × 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 $2T_{OSC}$ units, and the vertical position is set in 1024 steps in $1T_H$ 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 ≤ "003_{16}", HS ≥ "800_{16}."

3: When using SPRITE OSD, do not set VSi = "000_{16}," VSi ≥ "400_{16}."

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.

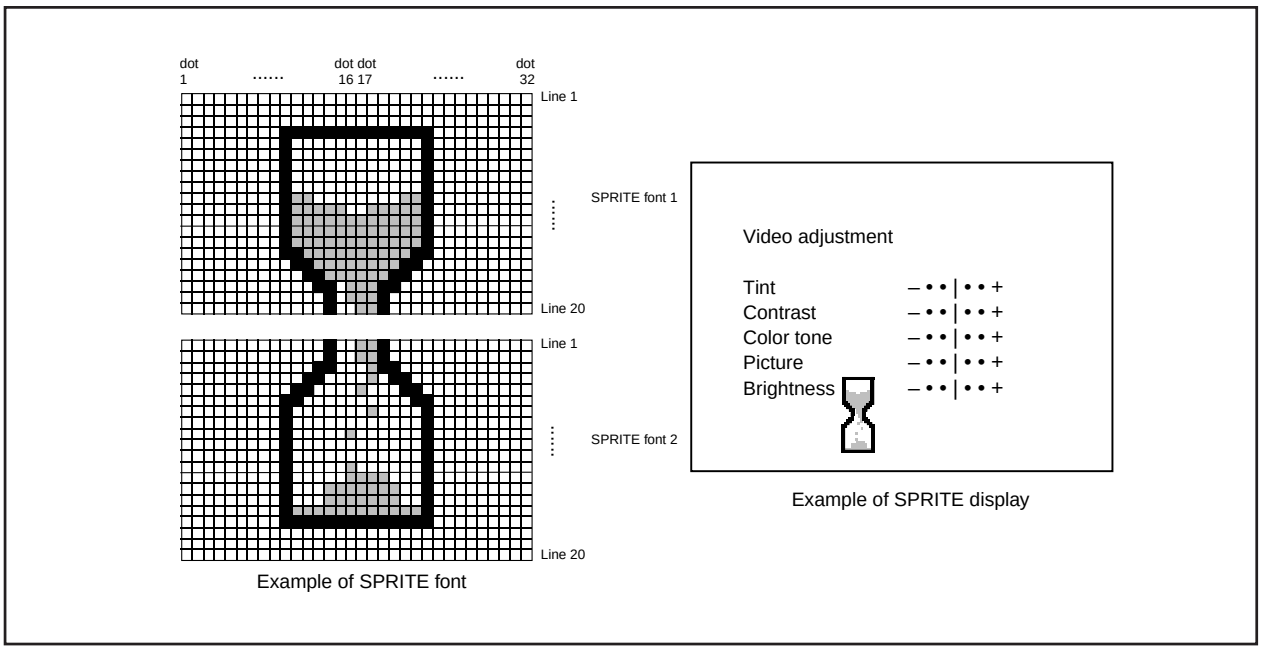
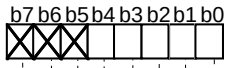


Figure 16.40 SPRITE OSD display example

SPRITE OSD control register



Symbol	Address	When reset
SC	0201_{16}	$XXX00000_2$

Bit symbol	Bit name	Function	RW
SC0	SPRITE font 1 control bit	0: Do not display 1: Display	RW
SC1	Pre-divide ratio selection bit	0: Pre-divide ratio 1 1: Pre-divide ratio 2	
SC2	Dot size selection bits	b3 b2 0 0: 1Tc × 1/2H 0 1: 1Tc × 1H 1 0: 2Tc × 1H 1 1: 2Tc × 2H	
SC3			
SC4	SPRITE font 2 control bit	0: Do not display 1: Display	
Nothing is assigned. In an attempt to write to these bits, write “0.” The value, if read, turns out to be “0.”			–

Notes 1: Tc is OSD clock cycle divided in pre-divide circuit.
2: H is HSYNC

Figure 16.41 SPRITE OSD control register

SPRITE horizontal position register

(b15) (b8)
b7 b6 b5 b4 b3 b2 b1 b0 b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
HS	0279_{16}, 0278_{16}	Indeterminate

Bit symbol	Bit name	Function	RW
HS10 to HS0	Horizontal display start position control bits of SPRITE font	Horizontal display start position = $2T_{OSC} \times n$ (n: setting value, T_{OSC}: OSD oscillation cycle)	RW
Nothing is assined. In an attempt to write to this bit, write "0." The value, if read, turns out to be indeterminate.			—

Note : Do not set HS ≤ "003_{16}," HS ≥ "800_{16}."

Figure 16.42 SPRITE horizontal position register

SPRITE vertical position register i

(b15) (b8)
b7 b6 b5 b4 b3 b2 b1 b0 b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
VS1	0275_{16}, 0274_{16}	Indeterminate
VS2	0277_{16}, 0276_{16}	Indeterminate

Bit symbol	Bit name	Function	RW
VSi9 to VSi0	Vertical display start position control bits of SPRITE font i (i = 1, 2)	Vertical display start position = $T_H \times n$ (n: setting value, T_H: H_{SYNC} cycle)	RW
Nothing is assined. In an attempt to write to this bit, write "0." The value, if read, turns out to be indeterminate.			—

Note : Do not set VSi = "000_{16}," VSi ≥ "400_{16}" (i = 1, 2).

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.

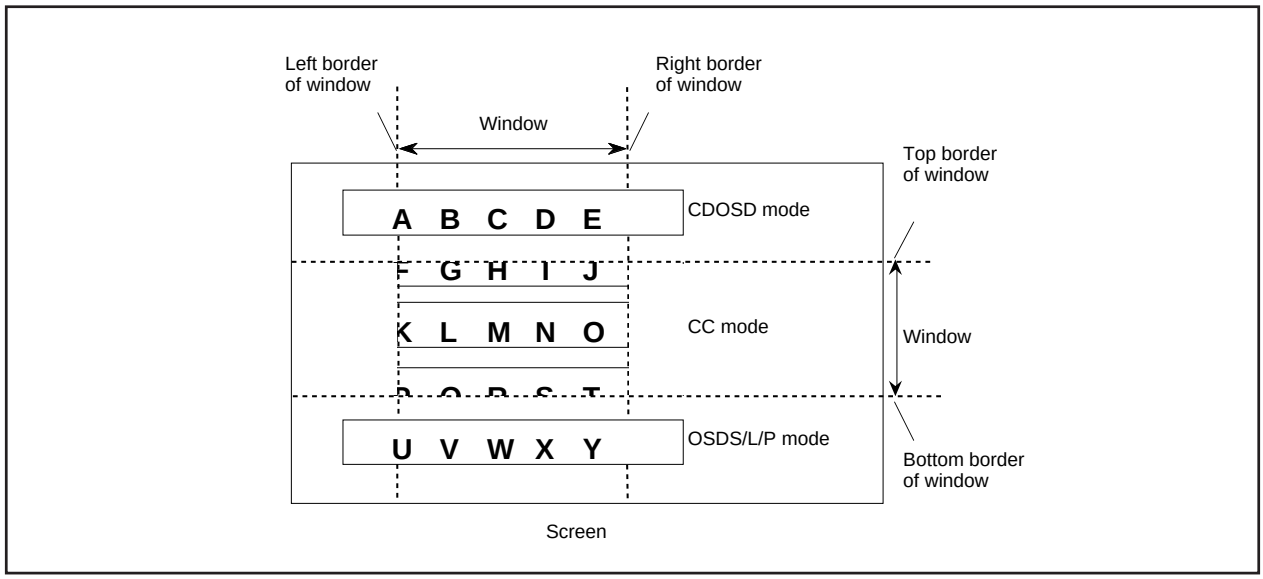


Figure 16.44 Example of window function (When CC mode is valid)

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.

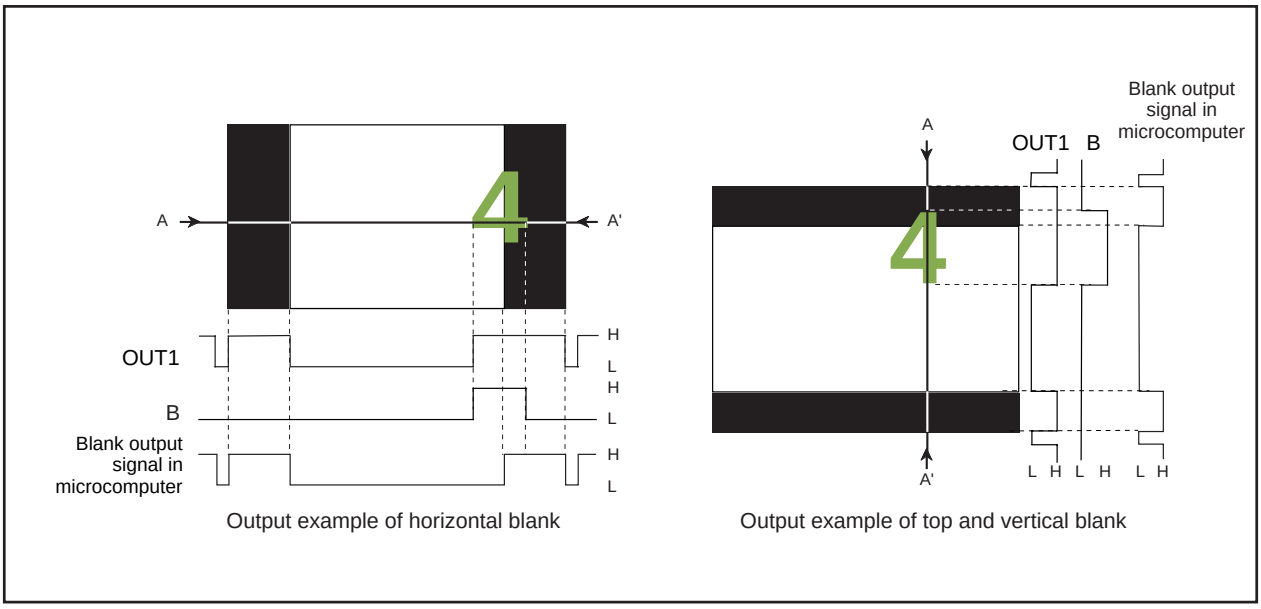


Figure 16.45 Blank output example (when OSD output is B + OUT1)

Top border control register

(b15) (b8)
b7 b6 b5 b4 b3 b2 b1 b0 b7 b6 b5 b4 b3 b2 b1 b0

Symbol: TBR
Address: 020D16, 020C16
When reset: Indeterminate

Bit symbol	Bit name	Function	RW
TBR_9 to TBR_0	Top border control bits	Top border position = TH × n (n: setting value, TH: HSYNC cycle)	RW
Nothing is assined. In an attempt to write to this bit, write "0." The value, if read, turns out to be indeterminate.			—

Notes 1 : Do not set TBR ≤ "00116," TBR ≥ "40016."
2 : Set as TBR < BBR.

Figure 16.46 Top border control register

Bottom border control register

(b15) (b8)
b7 b6 b5 b4 b3 b2 b1 b0 b7 b6 b5 b4 b3 b2 b1 b0

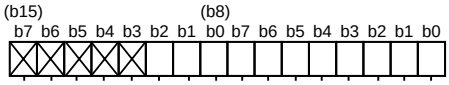
Symbol: BBR
Address: 020F16, 020E16
When reset: Indeterminate

Bit symbol	Bit name	Function	RW
BBR_9 to BBR_0	Bottom border control bits	Bottom border position = TH × n (n: setting value, TH: HSYNC cycle)	RW
Nothing is assined. In an attempt to write to this bit, write "0." The value, if read, turns out to be indeterminate.			—

Notes 1 : Do not set BBR ≥ "40016."
2 : Set as TBR < BBR.

Figure 16.47 Bottom border control register

Left border control register



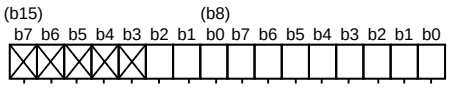
Symbol	Address	When reset
LBR	027116, 027016	XXXXX00000000001$_2$

Bit symbol	Bit name	Function	RW
LBR_10 to LBR_0	Left border control bits	Left border position = 4Tosc × n (n: setting value, Tosc: OSD oscillation cycle)	RW
Nothing is assined. In an attempt to write to this bit, write "0." The value, if read, turns out to be indeterminate.			—

Notes 1 : Do not set LBR ≤ "003_{16}," LBR ≥ "800_{16}."
2 : Set as LBR < RBR.

Figure 16.48 Left border control register

Right border control register



Symbol	Address	When reset
RBR	027316, 027216	XXXXX00000000000$_2$

Bit symbol	Bit name	Function	RW
RBR_10 to RBR_0	Right border control bits	Left border position = 4Tosc × n (n: setting value, Tosc: OSD oscillation cycle)	RW
Nothing is assined. In an attempt to write to this bit, write "0." The value, if read, turns out to be indeterminate.			—

Notes 1 : Do not set RBR ≥ "800_{16}."
2 : Set as LBR < RBR.

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.

Raster color register

(b15) (b8)
b7 b6 b5 b4 b3 b2 b1 b0 b7 b6 b5 b4 b3 b2 b1 b0

Symbol: RC
Address: 0209_{16}, 0208_{16}
When reset: 0000_{16}

Bit symbol	Bit name	Function	RW
RC2 to RC0	R singnal output control bits	b2 b1 b0 0 0 0 : Vss 0 0 1 : 1/7V 0 1 0 : 2/7V 0 1 1 : 3/7V 1 0 0 : 4/7V 1 0 1 : 5/7V 1 1 0 : 6/7V 1 1 1 : 7/7V	RW
Nothing is assined. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—
RC6 to RC4	G singnal output control bits	b6 b5 b4 0 0 0 : Vss 0 0 1 : 1/7V 0 1 0 : 2/7V 0 1 1 : 3/7V 1 0 0 : 4/7V 1 0 1 : 5/7V 1 1 0 : 6/7V 1 1 1 : 7/7V	RW
Nothing is assined. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—
RC10 to RC8	B singnal output control bits	b2 b1 b0 0 0 0 : Vss 0 0 1 : 1/7V 0 1 0 : 2/7V 0 1 1 : 3/7V 1 0 0 : 4/7V 1 0 1 : 5/7V 1 1 0 : 6/7V 1 1 1 : 7/7V	RW
Nothing is assined. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—
RC12	OUT1 singnal output control bit	0: No output 1: Output	RW
RC13	OUT2 singnal output control bit	0: No output 1: Output	RW
Nothing is assined. In an attempt to write to this bit, write “0.” The value, if read, turns out to be indeterminate.			—

Figure 16.50 Raster color register

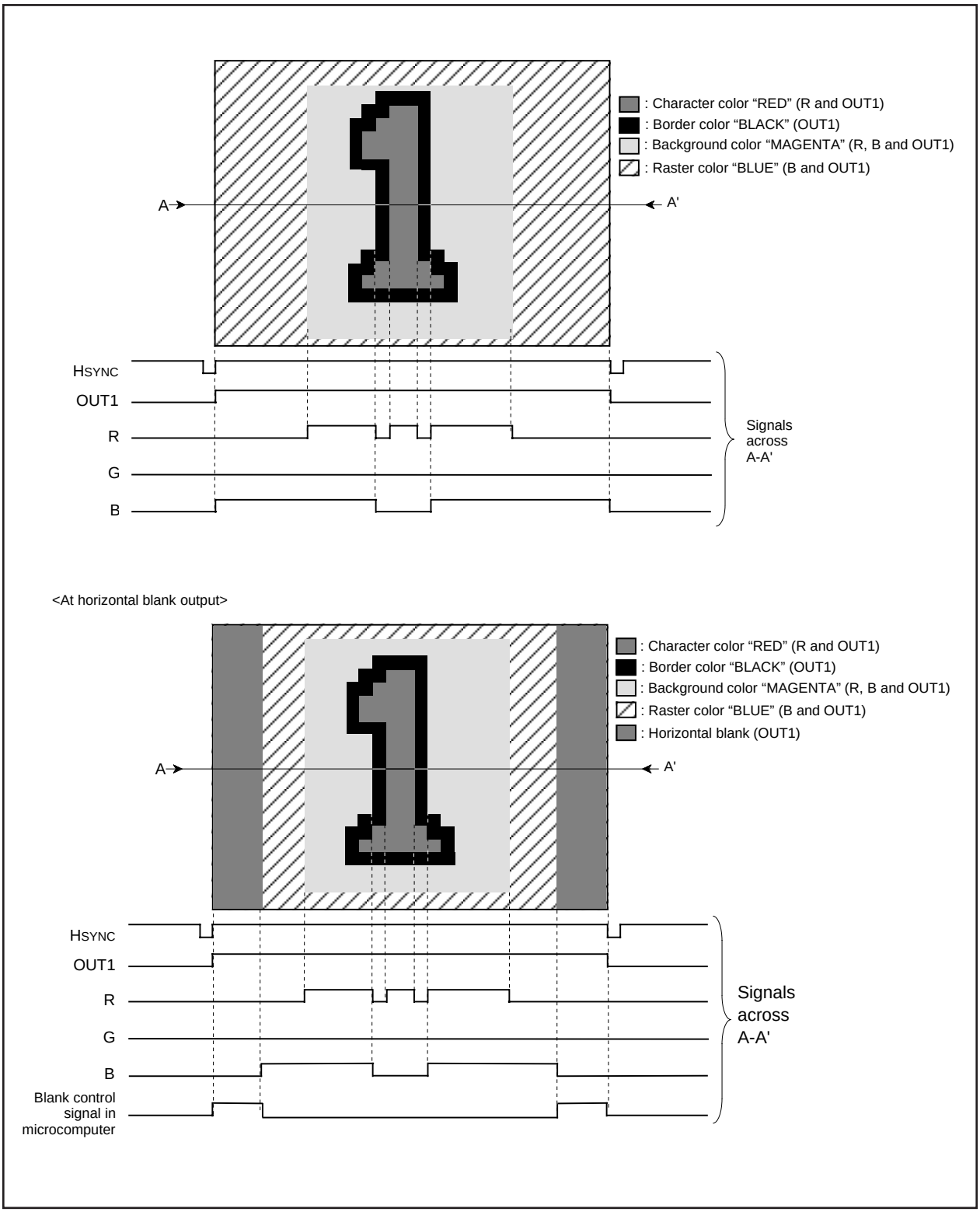


Figure 16.51 Example of raster coloring

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 × 1H	Value of vertical position register × 2H
Vertical dot size	1Tc × 1/2H 1Tc × 1H 2Tc × 2H 3Tc × 3H	1Tc × 1H 1Tc × 2H 2Tc × 4H 3Tc × 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 $02DE_{16}$, bit 4) and the RGBRON bit (address $025D_{16}$, bit 6) to "1."Also, set the ANARGBCAPON bit (address 2DE, bit3) and attach the capacitor for analog RGB internal operation stability (0.47 μF (reference value)) to the CAP pin.

2: To use the OUT1 and OUT2 pin, set the OUT1EN bit (address $02DB_{16}$, bit 4) and the OUT2EN bit (address $02DB_{16}$, bit 5) to "1."

OSD Reserved Register

OSD reserved register 1

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

Symbol	Address	When reset
OR1	025D16	0016

Bit symbol	Bit name	Description	RW
Reserved bits		Must always be set to “0”	RW
Reserved bit		Must always be set to “0”	
RGBRON	Analog RGB output control bit	0: OFF 1: ON	
Reserved bit		Must always be set to “0”	

Figure 16.52 OSD reserved register 1

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

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

Symbol	Address	When reset
OR2	027C16	0016
OR3	027B16	0016
OR5	020A16	0016

Bit symbol	Bit name	Description	RW
Reserved bits		Must always be set to 0	RW

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

OSD reserved register 4

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

Symbol	Address	When reset
OR4	027A16	X0000000 2

Bit symbol	Bit name	Description	RW
Reserved bits		Must always be set to “0”	RW

Figure 16.54 OSD reserved register 4

TEST reserved register 0

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

Symbol: IDT0 | Address: 026E16 | When reset: 0016

Bit symbol	Bit name	Function	RW
Reserved bits		Must always be set to “0”	RW
Reserved bit		Must always be set to “1”	RW
Reserved bit		Must always be set to “0”	RW

Figure 16.55 TEST reserved register 0

TEST reserved register 1

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

Symbol: IDT1 | Address: 030E16 | When reset: 0016

Bit symbol	Bit name	Description	RW
Reserved bits		Must always be set to “0”	RW

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.

Peripheral mode register

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

Symbpl	Address	After reset
PM	$027D_{16}$	$000XXXXX_2$

Bit symbol	Bit name	Function	RW
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be Indeterminate.			—
Reserved bit		Must be set to "0".	RW
Reserved bit		Must be set to "0".	RW
Reserved bit		Must be set to "1".	RW

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 (A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL/WR}$, $\overline{WRH/BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$, and BCLK) cannot be modified.

No direction register bit for $P8_5$ 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 (A_0 to A_{19}, D_0 to D_{15}, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL/WR}$, $\overline{WRH/BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$, 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, $P4_0$ to $P4_3$, and P5 during memory extension and microprocessor modes. Although the register contents can be modified, no pull-up resistors are connected.

(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.

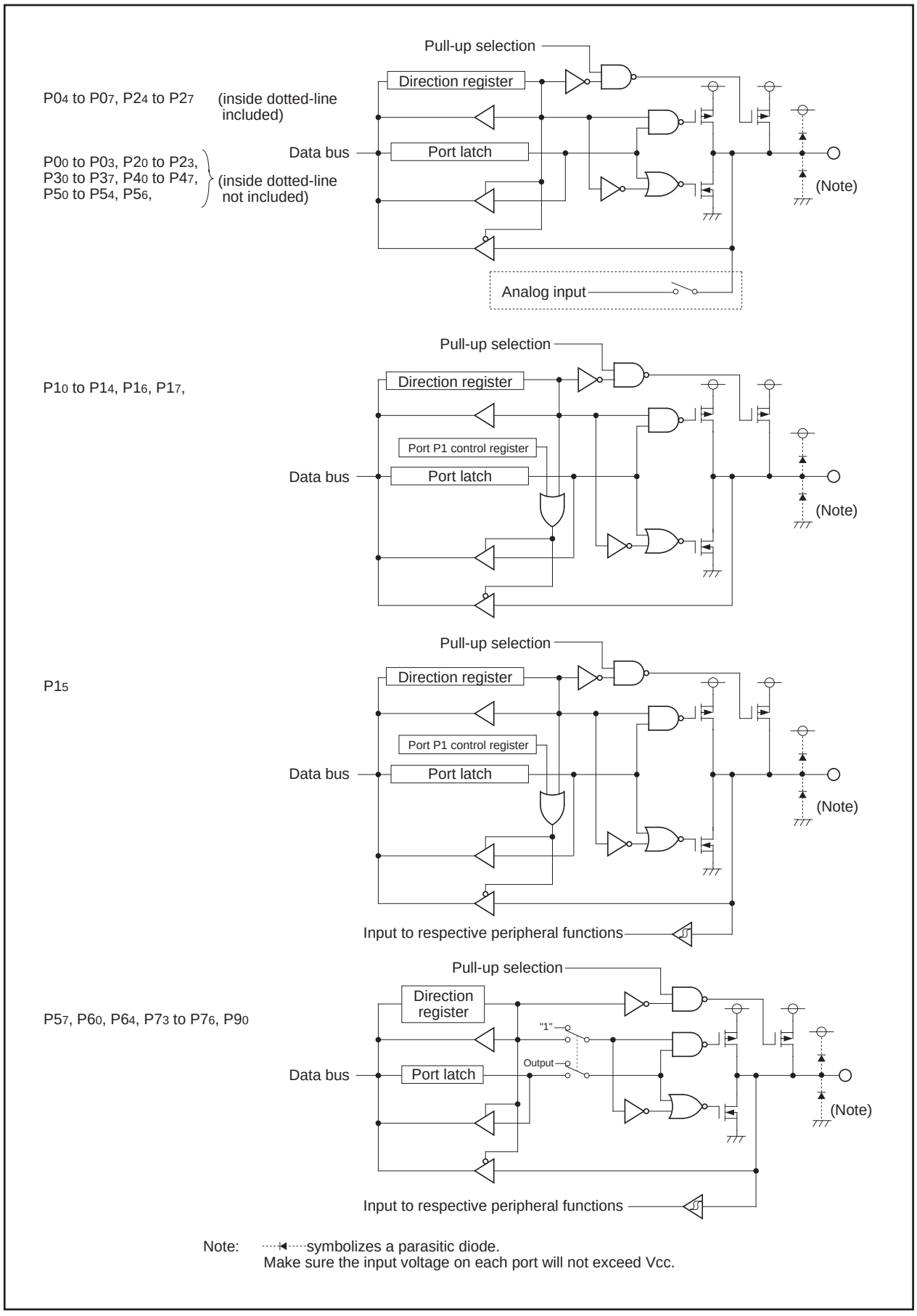
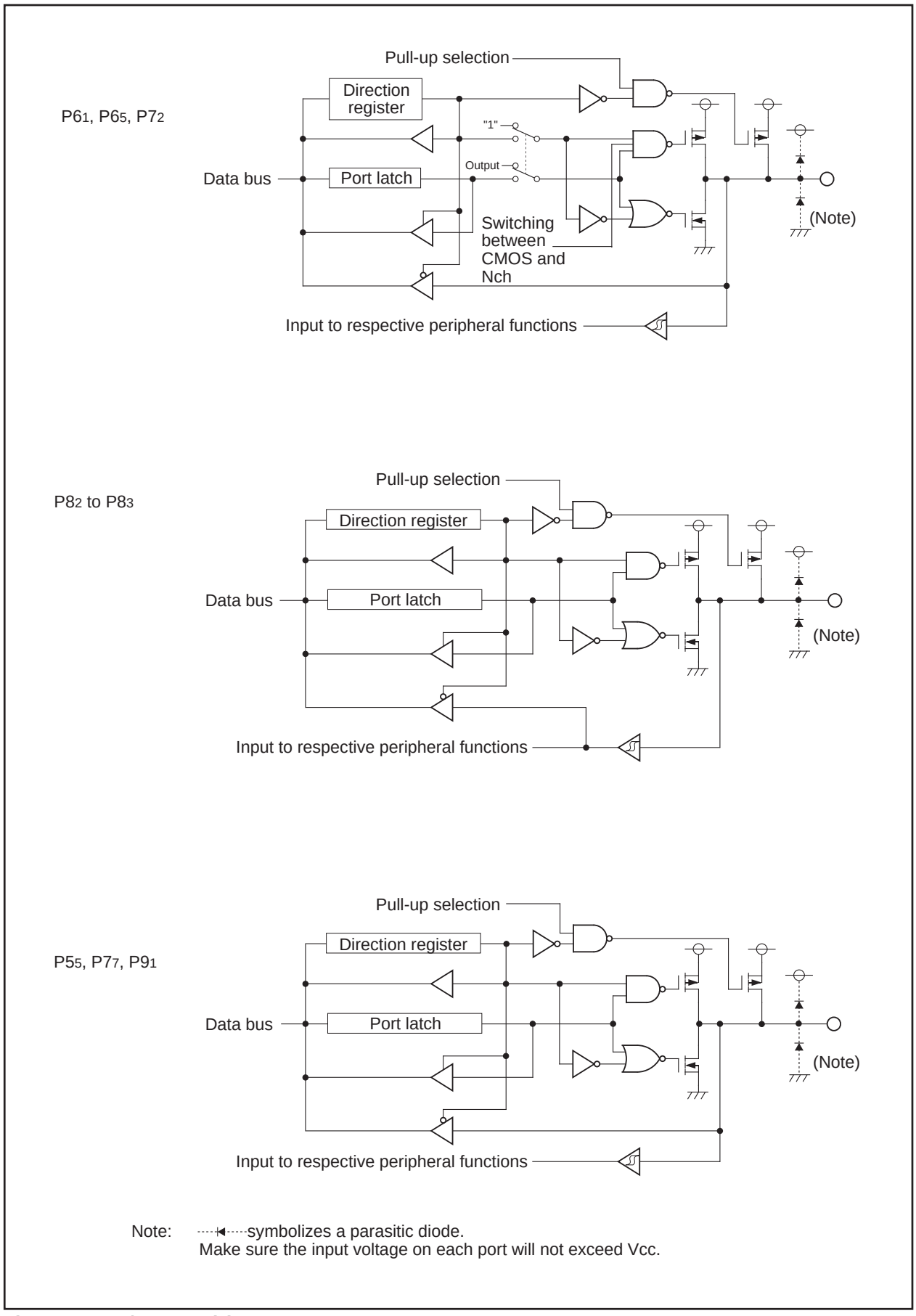
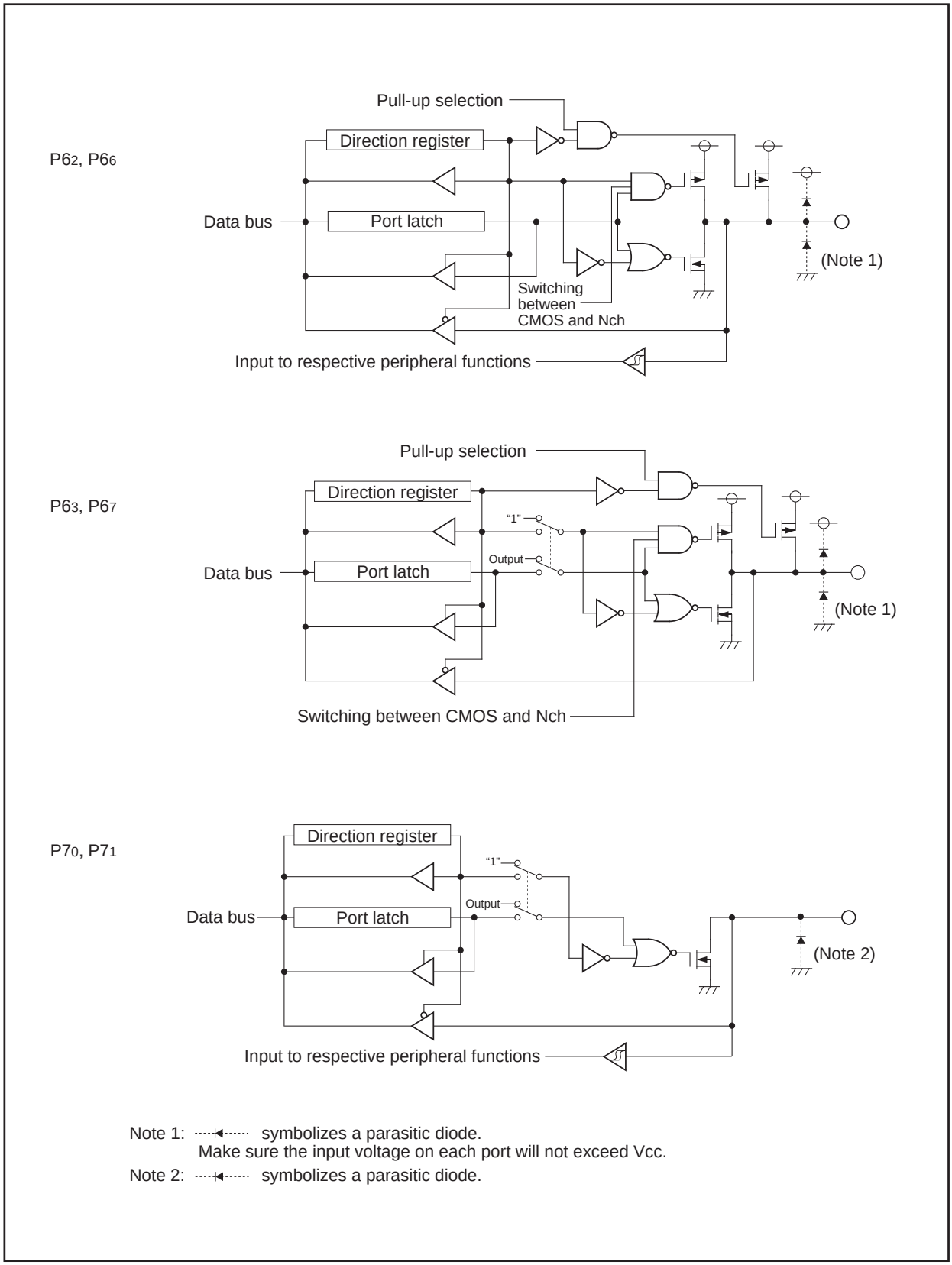


Figure 17.1. I/O Ports (1)



Note: ⋯◂⋯ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 17.2. I/O Ports (2)



Note 1: ⊣ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.
Note 2: ⊣ symbolizes a parasitic diode.

Figure 17.3. I/O Ports (3)

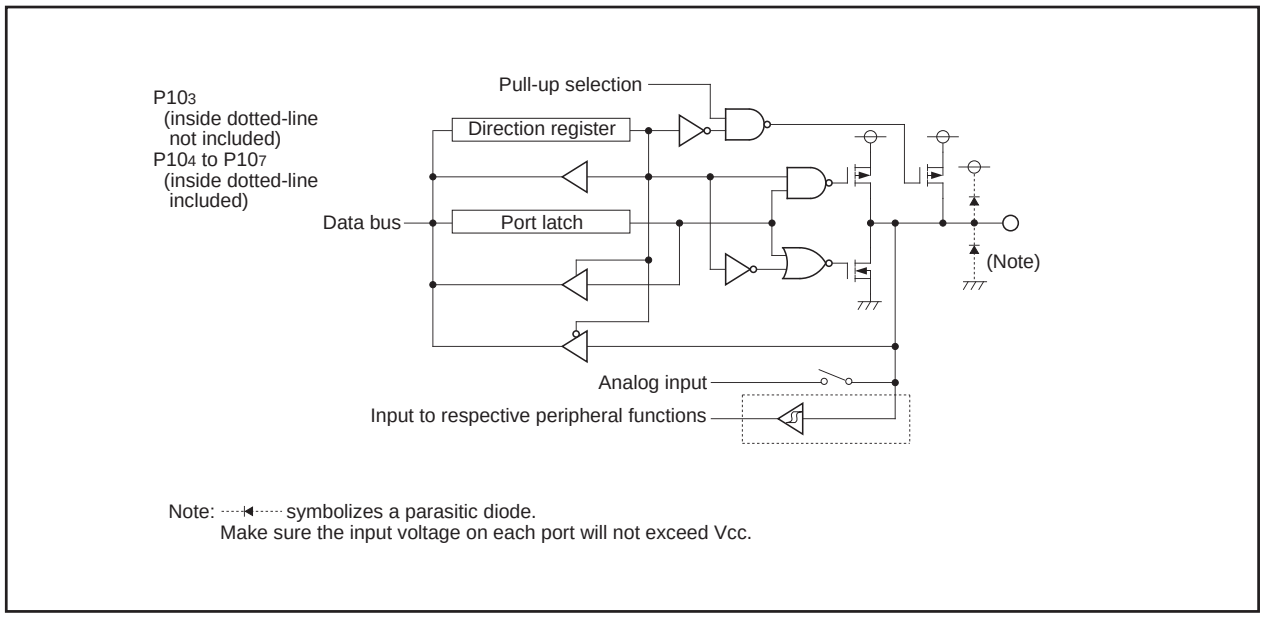


Figure 17.4. I/O Ports (4)

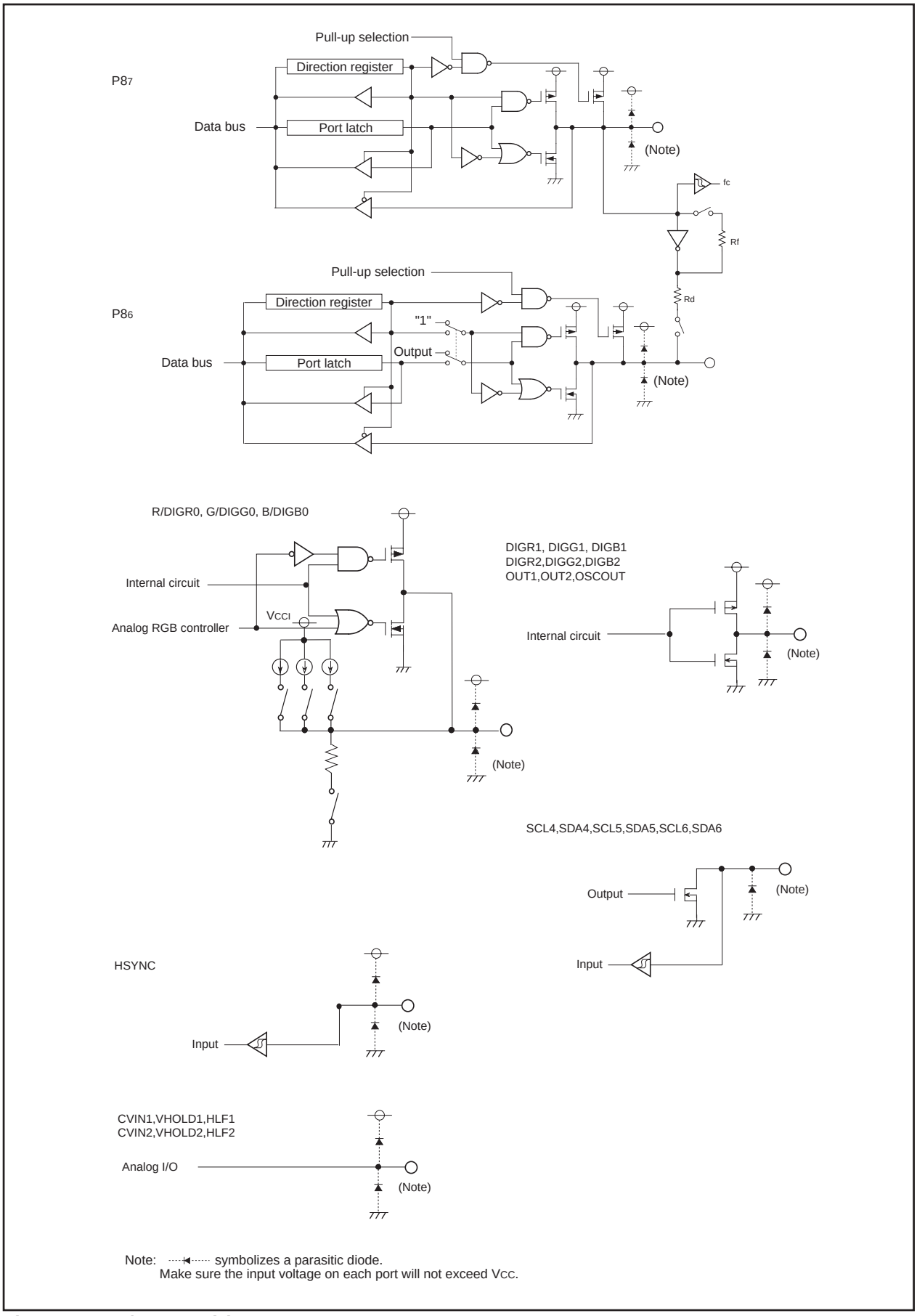
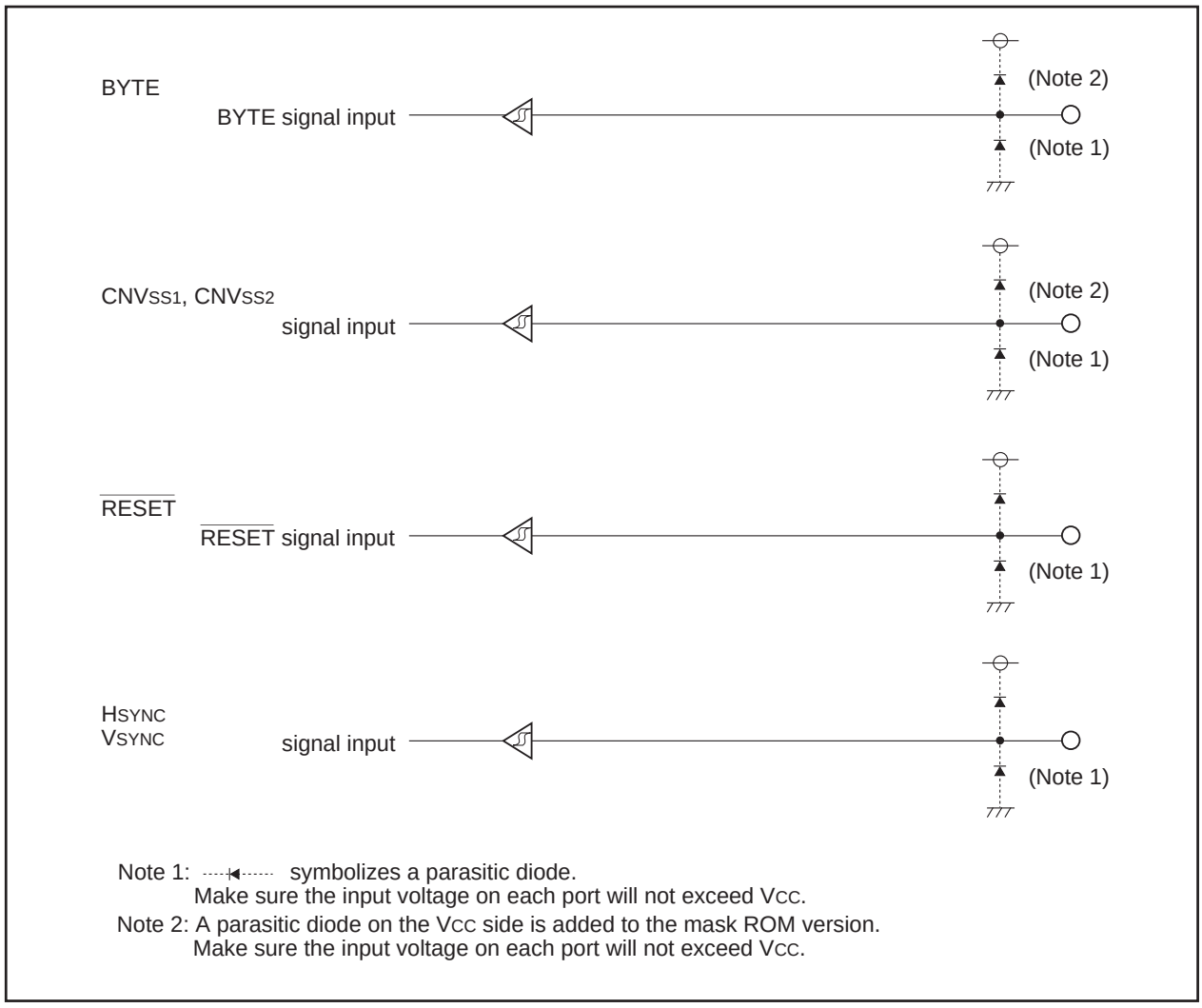


Figure 17.5. I/O Ports (5)



Note 1: ----|◄---- symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed VCC.
Note 2: A parasitic diode on the VCC side is added to the mask ROM version.
Make sure the input voltage on each port will not exceed VCC.

Figure 17.6. I/O Pins

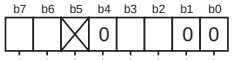
Port Pi direction register (i=0 to 7) (Note 1)

Symbol	Address	After reset
PD0 to PD3	03E2$_{16}$, 03E3$_{16}$, 03E6$_{16}$, 03E7$_{16}$	00$_{16}$
PD4 to PD7	03EA$_{16}$, 03EB$_{16}$, 03EE$_{16}$, 03EF$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
PDi_0	Port Pi$_0$ direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 7)	RW
PDi_1	Port Pi$_1$ direction bit		RW
PDi_2	Port Pi$_2$ direction bit		RW
PDi_3	Port Pi$_3$ direction bit		RW
PDi_4	Port Pi$_4$ direction bit		RW
PDi_5	Port Pi$_5$ direction bit		RW
PDi_6	Port Pi$_6$ direction bit		RW
PDi_7	Port Pi$_7$ direction bit		RW

Note 1: During memory extension and microprocessor modes, the PD register for the pins functioning as bus control pins (A$_0$ to A$_{19}$, D$_0$ to D$_{15}$, $\overline{CS_0}$ to $\overline{CS_3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK) cannot be modified.

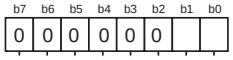
Port P8 direction register



Symbol	Address	After reset
PD8	03F2$_{16}$	00X00000$_2$

Bit symbol	Bit name	Function	RW
($\overline{b1}$,b0)	Reserved bit	Must be set to "0".	RW
PD8_2	Port P8$_2$ direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_3	Port P8$_3$ direction bit		RW
($\overline{b4}$)	Reserved bit	Must be set to "0".	RW
—— (b5)	Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		—
PD8_6	Port P8$_6$ direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_7	Port P8$_7$ direction bit		RW

Port P9 direction register

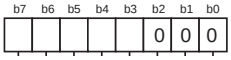


Symbol	Address	After reset
PD9	03F3$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
PD9_0	Port P9$_0$ direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD9_1	Port P9$_1$ direction bit		RW
($\overline{b7}$-$\overline{b2}$)	Reserved bit	Must be set to "0".	RW

Note: PD9 register should write PRC2 bit of PRCR register by the next command set to "1" (write-in permitted.)

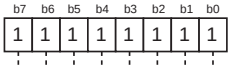
Port P10 direction register



Symbol	Address	After reset
PD10	03F6$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
($\overline{b2}$-$\overline{b0}$)	Reserved bit	Must be set to "0".	RW
PD10_3	Port P10$_3$ direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD10_4	Port P10$_4$ direction bit		RW
PD10_5	Port P10$_5$ direction bit		RW
PD10_6	Port P10$_6$ direction bit		RW
PD10_7	Port P10$_7$ direction bit		RW

Reserved register

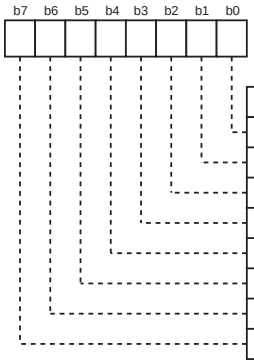


Symbol	Address	After reset
RSVREG03F7 RSVREG03FA RSVREG03FB	03F7$_{16}$, 03FA$_{16}$, 03FB$_{16}$	00$_{16}$

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

Figure 17.7. PD0 to PD10 Registers

Port Pi register (i=0 to 7) (Note 2)

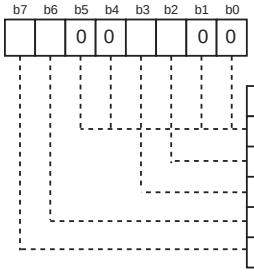


Symbol	Address	After reset
P0 to P3	03E0$_{16}$, 03E1$_{16}$, 03E4$_{16}$, 03E5$_{16}$	Indeterminate
P4 to P7	03E8$_{16}$, 03E9$_{16}$, 03EC$_{16}$, 03ED$_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
Pi_0	Port Pi$_0$ bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level (Note 1) (i = 0 to 7)	RW
Pi_1	Port Pi$_1$ bit		RW
Pi_2	Port Pi$_2$ bit		RW
Pi_3	Port Pi$_3$ bit		RW
Pi_4	Port Pi$_4$ bit		RW
Pi_5	Port Pi$_5$ bit		RW
Pi_6	Port Pi$_6$ bit		RW
Pi_7	Port Pi$_7$ bit		RW

Note 1: Since P7$_0$ and P7$_1$ are N-channel open drain ports, the data is high-impedance.
Note 2: During memory extension and microprocessor modes, the Pi register for the pins functioning as bus control pins (A$_0$ to A$_{19}$, D$_0$ to D$_{15}$, $\overline{CS_0}$ to $\overline{CS_3}$, $\overline{RD}$, WRL/$\overline{WR}$, WRH/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK) cannot be modified.

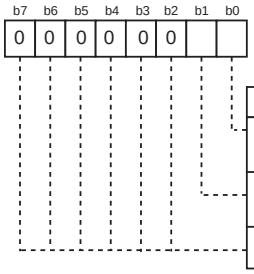
Port P8 register



Symbol	Address	After reset
P8	03F0$_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
($\overline{b5}$-$\overline{b4}$, b1-b0)	Reserved bit	Must be set to "0".	RW
P8_2	Port P8$_2$ bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level	RW
P8_3	Port P8$_3$ bit		RW
P8_6	Port P8$_6$ bit		RW
P8_7	Port P8$_7$ bit		RW

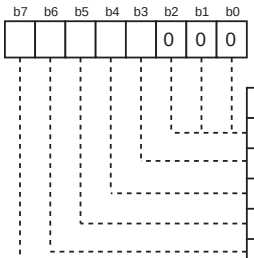
Port P9 register



Symbol	Address	After reset
P9	03F1$_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
P9_0	Port P9$_0$ bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level	RW
P9_1	Port P9$_1$ bit		RW
($\overline{b7}$-$\overline{b2}$)	Reserved bit	Must be set to "0".	RW

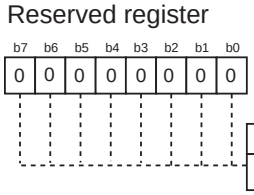
Port P10 register



Symbol	Address	After reset
P10	03F4$_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
($\overline{b7}$-$\overline{b2}$)	Reserved bit	Must be set to "0".	RW
P10_3	Port Pi$_1$ bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level	RW
P10_4	Port Pi$_2$ bit		RW
P10_5	Port Pi$_3$ bit		RW
P10_6	Port Pi$_4$ bit		RW
P10_7	Port Pi$_5$ bit		RW

Reserved register

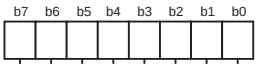


Symbol	Address	After reset
RSVREG03F5 RSVREG03F8 RSVREG03F9	03F5$_{16}$, 03F8$_{16}$, 03F9$_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
($\overline{b7}$-$\overline{b2}$)	Reserved bit	Must be set to "0".	RW

Figure 17.8. P0 to P10 Registers

Pull-up control register 0 (Note 1)

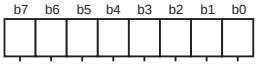


Symbol	Address	After reset
PUR0	03FC16	0016

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

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

Pull-up control register 1

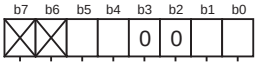


Symbol	Address	After reset(Note 5)
PUR1	03FD16	000000002 000000102

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

Note 1: The P70 and P71 pins do not have pull-ups.
Note 2: During memory extension and microprocessor modes, the pins are not pulled high although the contents of this bit can be modified.
Note 3: The pin for which this bit is “1” (pulled high) and the direction bit is “0” (input mode) is pulled high.
Note 4: If the PM01 to PM00 bits are set to “012” (memory expansion mode) or “112” (microprocessor mode) in a program during single-chip mode, the PU11 bit becomes “1”.
Note 5: The values after hardware reset 1 and 2 are as follows:
• 000000002 when input on CNVss1 pin is “L“
• 000000102 when input on CNVss1 pin is “H“
The values after software reset, watchdog timer reset and oscillation stop detection reset are as follows:
• 000000002 when PM 01 to PM00 bits of PM0 register are “002“ (single-chip mode)
• 000000102 when PM 01 to PM00 bits of PM0 register are “012“ (memory expansion mode) or “112“ (microprocessor mode)

Pull-up control register 2



Symbol	Address	After reset
PUR2	03FE16	0016

Bit symbol	Bit name	Function	RW
PU20	P80 to P83 pull-up	0 : Not pulled high 1 : Pulled high (Note 1)	RW
PU21	P84 to P87 pull-up (Note 2)		RW
(b3-b2)	Reserved bit	Must be set to "0".	RW
PU24	P103 pull-up	0 : Not pulled high 1 : Pulled high (Note 1)	RW
PU25	P104 to P107 pull-up		RW
(b7-b6)	Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.		—

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

Figure 17.9. PUR0 to PUR2 Registers

Port control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbpl: PCR
Address: $03FF_{16}$
After reset: 00_{16}

Bit symbol	Bit name	Function	RW
PCR0	Port P1 control bit	Operation performed when the P1 register is read 0: When the port is set for input, the input levels of $P1_0$ to $P1_7$ pins are read. When set for output, the port latch is read. 1: The port latch is read regardless of whether the port is set for input or output.	RW
— (b7-b1)	Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.		—

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 V_{SS} via a resistor(pull-down); or after setting for output mode, leave these pins open. (Note 1)
X_{OUT} (Note 2)	Open
BYTE	Connect to V_{SS}

Note 1: When the ports $P7_0$ and $P7_1$ are set for output mode, make sure a low-level signal is output from the pins. The ports $P7_0$ and $P7_1$ are N-channel open-drain outputs.
Note 2: With external clock input to X_{IN} 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 V_{SS} via a resistor (pull-down); or after setting for output mode, leave these pins open. (Note 1, 2)
$P4_5$ / $\overline{CS1}$ to $P4_7$ / $\overline{CS3}$	Connect to V_{CC} via a resistor (pulled high) by setting the PD4 register's corresponding direction bit for $\overline{CSi}$ (i=1 to 3) to "0" (input mode) and the CSR register's CSi bit to "0" (chip select disabled).
$\overline{BHE}$, ALE, $\overline{HLDA}$, X_{OUT}(Note 3), BCLK (Note 4)	Open
$\overline{HOLD}$, $\overline{RDY}$	Connect via resistor to V_{CC2} (pull-up)

Note 1: If the CNV_{SS1} pin has the V_{SS} 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 $P7_0$ and $P7_1$ are set for output mode, make sure a low-level signal is output from the pins. The ports $P7_0$ and $P7_1$ are N-channel open-drain outputs.
Note 3: With external clock input to X_{IN} 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 V_{CC} via a resistor (pulled high).

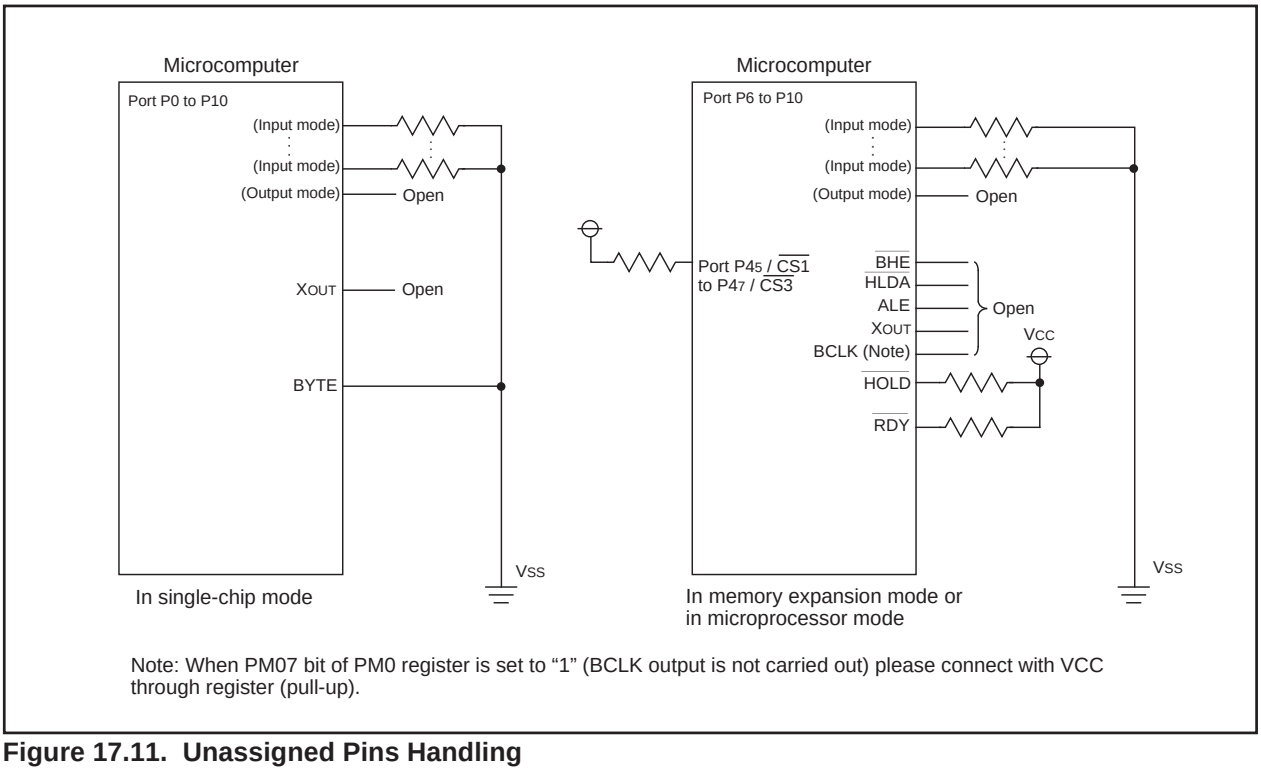
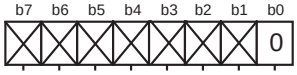


Figure 17.11. Unassigned Pins Handling

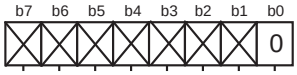
Reserved register



Symbpl RSVREG026F | Address 026F16 | After reset XXXXXXX02

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

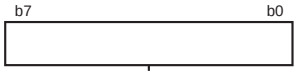
Reserved register



Symbpl RSVREG030F | Address 030F16 | After reset XXXXXXX02

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

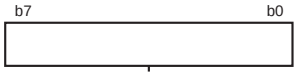
Reserved register



Symbpl	Address	After reset
RSVREG0342	0342_{16}	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b7-b0)}}$	Reserved bit	Setup is arbitrary.	WO

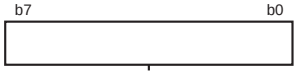
Reserved register



Symbpl	Address	After reset
RSVREG0343	0343_{16}	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b7-b0)}}$	Reserved bit	Setup is arbitrary.	WO

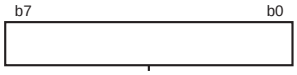
Reserved register



Symbpl	Address	After reset
RSVREG0344	0344_{16}	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b7-b0)}}$	Reserved bit	Setup is arbitrary.	WO

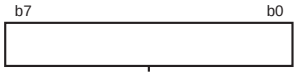
Reserved register



Symbpl	Address	After reset
RSVREG0345	0345_{16}	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b7-b0)}}$	Reserved bit	Setup is arbitrary.	WO

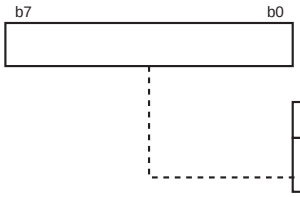
Reserved register



Symbpl	Address	After reset
RSVREG0346	0346_{16}	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b7-b0)}}$	Reserved bit	Setup is arbitrary.	WO

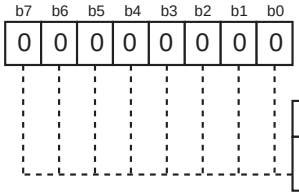
Reserved register



Symbpl	Address	After reset
RSVREG0347	034716	Indeterminate

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

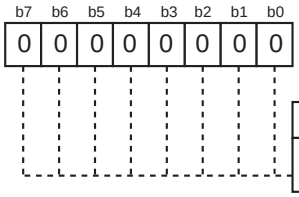
Reserved register



Symbpl	Address	After reset
RSVREG0348	034816	0016

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

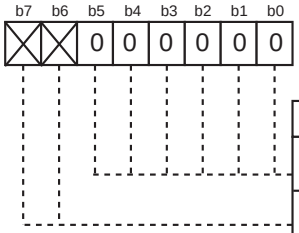
Reserved register



Symbpl	Address	After reset
RSVREG0349	034916	0016

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

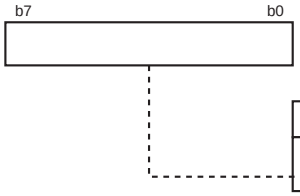
Reserved register



Symbpl	Address	After reset
RSVREG034A	$034A16$	0016
RSVREG034B	$034B16$	0016

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



Symbpl	Address	After reset
RSVREG034C	$034C16$	Indeterminate

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

Reserved register

b7 b0

Symbpl	Address	After reset
RSVREG03BC	03BC16	Indeterminate

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

Reserved register

b7 b0

Symbpl	Address	After reset
RSVREG03BD	03BD16	Indeterminate

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

Reserved register

b7 b0

Symbpl	Address	After reset
RSVREG03BE	03BE16	Indeterminate

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

Reserved register

b7 b0

Symbpl	Address	After reset
RSVREG03C0	03C016	Indeterminate

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

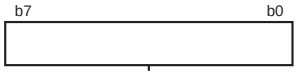
Reserved register

b7 b0

Symbpl	Address	After reset
RSVREG03C1	03C116	Indeterminate

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

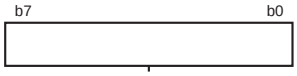
Reserved register



Symbpl	Address	After reset
RSVREG03C2	$03C2_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b7-b0)}}$	Reserved bit	Setup is arbitrary.	RO

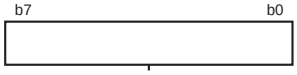
Reserved register



Symbpl	Address	After reset
RSVREG03C3	$03C3_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b7-b0)}}$	Reserved bit	Setup is arbitrary.	RO

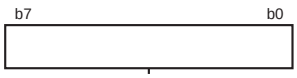
Reserved register



Symbpl	Address	After reset
RSVREG03C4	$03C4_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b7-b0)}}$	Reserved bit	Setup is arbitrary.	RO

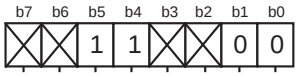
Reserved register



Symbpl	Address	After reset
RSVREG03C5	$03C5_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
$\overline{\text{(b7-b0)}}$	Reserved bit	Setup is arbitrary.	RO

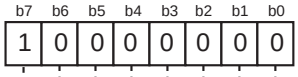
Reserved register



Symbpl: RSVREG03DE
Address: $03DE_{16}$
After reset: $XX00XXXX_2$

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

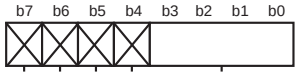
Reserved register



Symbpl: RSVREG03DF
Address: $03DF_{16}$
After reset: 00_{16}

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

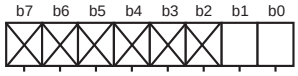
Reserved register



Symbpl	Address	After reset
RSVREG034D	034D16	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		—

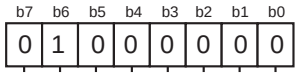
Reserved register



Symbpl	Address	After reset
RSVREG039E	039E16	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”.		—

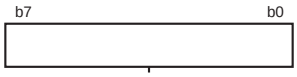
Reserved register



Symbpl	Address	After reset
RSVREG0362	036216	010000002
RSVREG0366	036616	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

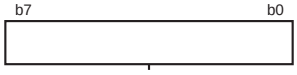
Reserved register



Symbpl	Address	After reset
RSVREG0363	036316	Indeterminate
RSVREG0367	036716	Indeterminate

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

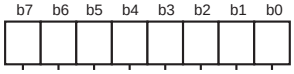
Reserved register



Symbpl	Address	After reset
RSVREG0360	036016	Indeterminate
RSVREG0364	036416	Indeterminate

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

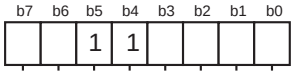
Extended register



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	○ ○

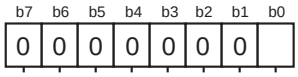
Extended register



Symbpl (EXTREG02C1) Address (02C1$_{16}$) After reset (00$_{16}$)

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	○ ○

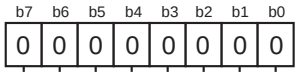
Extended register



Symbpl (EXTREG02C2) Address (02C2$_{16}$) After reset (00$_{16}$)

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 (02C3$_{16}$) After reset (00$_{16}$)

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

Extended register

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

Symbpl (EXTREG02C4) Address ($02C4_{16}$) After reset (00_{16})

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

Extended register

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

Symbpl (EXTREG02C5) Address ($02C5_{16}$) After reset (00_{16})

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

Extended register

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

Symbpl (EXTREG02C6) Address ($02C6_{16}$) After reset (00_{16})

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

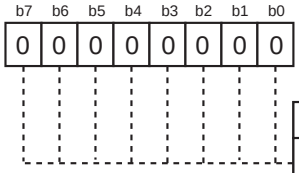
Extended register

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

Symbpl (EXTREG02C7) Address ($02C7_{16}$) After reset (00_{16})

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

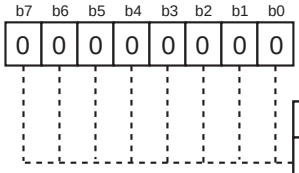
Extended register



Symbpl (EXTREG02C8) Address ($02C8_{16}$) After reset (00_{16})

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

Extended register



Symbpl (EXTREG02C9) Address ($02C9_{16}$) After reset (00_{16})

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

Extended register

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

Symbpl (EXTREG02CA) Address (02CA16) 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
0	0	0	0	0	0	0	0

Symbpl (EXTREG02CB) Address (02CB16) 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
0	0	0	0	0	0	0	0

Symbpl (EXTREG02CC) Address (02CC16) 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
0	0	0	0	0	0	0	0

Symbpl (EXTREG02CD) Address (02CD16) 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
0	0	0	0	0	0	0	0

Symbpl (EXTREG02CE) Address ($02CE_{16}$) After reset (00_{16})

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

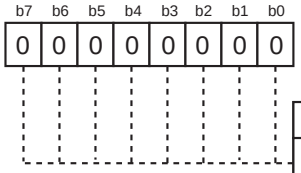
Extended register

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

Symbpl (EXTREG02CF) Address ($02CF_{16}$) After reset (00_{16})

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

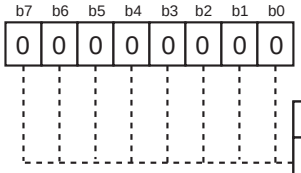
Extended register



Symbpl (EXTREG02D0) Address (02D0$_{16}$) After reset (00$_{16}$)

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

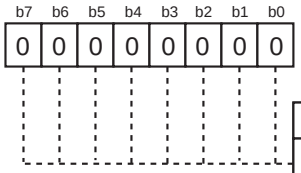
Extended register



Symbpl (EXTREG02D1) Address (02D1$_{16}$) After reset (00$_{16}$)

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

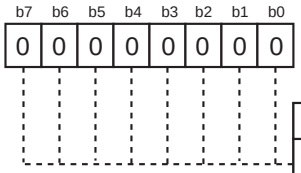
Extended register



Symbpl (EXTREG02D2) Address (02D2$_{16}$) After reset (00$_{16}$)

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

Extended register



Symbpl (EXTREG02D3) Address (02D3$_{16}$) After reset (00$_{16}$)

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

Extended register

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

Symbpl (EXTREG02D4) Address ($02D4_{16}$) After reset (00_{16})

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
1		0	0	0	0	0	

Symbpl (EXTREG02D5) Address ($02D5_{16}$) After reset (00_{16})

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

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Symbpl (EXTREG02D6) Address (02D6$_{16}$) After reset (00$_{16}$)

Bit symbol	Bit name	Function	R	W
SCL3DRVUP	SCL3 output buffer size adjustment	0 : weak 1 : strong	○	○
SDA3DRVUP	SDA3 output buffer size adjustment	0 : weak 1 : strong	○	○
SCL5DRVUP	SCL5 output buffer size adjustment	0 : weak 1 : strong	○	○
SDA5DRVUP	SDA5 output buffer size adjustment	0 : weak 1 : strong	○	○
SCL6DRVUP	SCL6 output buffer size adjustment	0 : weak 1 : strong	○	○
SDA6DRVUP	SDA6 output buffer size adjustment	0 : weak 1 : strong	○	○
Reserved bit	Must be set to "0".		○	○
Reserved bit	Must be set to "0".		○	○

Extended register

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

Symbpl (EXTREG02D7) Address (02D7$_{16}$) After reset (00$_{16}$)

Bit symbol	Bit name	Function	R	W
Reserved bit	Must be set to "0".		○	○
Reserved bit	Must be set to "1".		○	○

Extended register

b7	b6	b5	b4	b3	b2	b1	b0
1	1						

Symbpl (EXTREG02D8) Address ($02D8_{16}$) After reset (00_{16})

Bit symbol	Bit name	Function	R	W
SCL0INSEL	I^2C-bus0 SCL input pin selection	0 : Pin "SCL2" 1 : Pin "SCL5"	○	○
SDA0INSEL	I^2C-bus0 SDA input pin selection	0 : Pin "SDA2" 1 : Pin "SDA5"	○	○
SCL1INSEL0	I^2C-bus1 SCL input pin selection	0 : Pin "SCL3" 1 : Pin "SCL6"	○	○
SDA1INSEL0	I^2C-bus1 SDA input pin selection	0 : Pin "SDA3" 1 : Pin "SDA6"	○	○
SCL1INSEL1	I^2C-bus1 SCL input pin selection	0 : SCL3 or SCL6 (SCL1INSEL0 available) 1 : Pin "SCL1"	○	○
SDA1INSEL1	I^2C-bus1 SDA input pin selection	0 : SDA3 or SDA6 (SDA1INSEL0 available) 1 : Pin "SDA1"	○	○
$\overline{\text{(b7-b6)}}$	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
0	0	0	0	0	0	0	0

Symbpl (EXTREG02DA) Address (02DA16) After reset (0016)

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

Extended register

b7	b6	b5	b4	b3	b2	b1	b0

Symbpl (EXTREG02DB) Address (02DB16) After reset (0016)

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
0	0	0	0	0	0		0

Symbpl (EXTREG02DC) Address (02DC16) After reset (0016)

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
0	0	0				0	0

Symbpl (EXTREG02DD) Address (02DD16) After reset (0016)

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∝s (The removable maximum bus width=1∝s) 0 1 : 8∝s (The removable maximum bus width=32∝s) 1 0 : 16∝s (The removable maximum bus width=64∝s) 1 1 : 32∝s (The removable maximum bus width=128∝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
0	0	0			0	0	0

Symbpl (EXTREG02DE) Address (02DE16) After reset (0016)

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
0	0	0	0	0	0	0	0

Symbpl (EXTREG02DF) Address (02DF16) After reset (0016)

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
V_I	Input voltage	$\overline{RESET}$, $CNV_{SS}1$, BYTE, $CNV_{SS}2$, $P0_0$ 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, $P8_7$ $P9_0$ to $P9_1$, $P10_3$ to $P10_7$, VSYNC1, OSC1, HLF2, VHOLD2, CVIN2, HLF1,VHOLD1, CVIN1, HSYNC, X_{IN}, SCL5, SDA5,SCL6, SDA6		-0.3 to V_{CC1}+0.3	V
		$P7_0$, $P7_1$, SCL4, SDA4		-0.3 to 6.5	V
V_O	Output voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_2$, $P8_3$, $P8_6$, $P8_7$ $P9_0$ to $P9_1$, $P10_3$ to $P10_7$, OUT1, OUT2, OSC2, OSCHLF, HLF2, VHOLD2, CVIN2, HLF1, VHOLD1, CVIN1, X_{OUT}, DIGR1, DIGR2, DIGG1, DIGG2, DIGB1, DIGB2, OSCOUT, R, G, B, SCL5, SDA5, SCL6,SDA6		-0.3 to V_{CC1}+0.3	V
		$P7_0$, $P7_1$, SCL4, SDA4		-0.3 to 6.5	V
P_d	Power dissipation		Topr=25°C	500	mW
T_{opr}	Operating ambient temperature	At microcomputer operate		-20 to 70	°C
		At flash memory erase		0 to 60	°C
T_{stg}	Storage temperature			-40 to 125	°C

Table 18.2. Recommended Operating Conditions (Note 1)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V_{CC1}, V_{CC2}, V_{CC3}	Supply voltage(V_{CC1}=V_{CC2}=V_{CC3})		3.15	3.3	3.45	V
Vss	Supply voltage			0		V
V_{IH}	HIGH input voltage	$P3_1$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$	0.8V_{CC1}		V_{CC1}	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (during single-chip mode)	0.8V_{CC1}		V_{CC1}	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (data input during memory expansion and microprocessor modes)	0.5V_{CC1}		V_{CC1}	V
		$P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_2$, $P8_3$, $P8_6$, 8_7, $P9_0$ to $P9_1$, $P10_3$ to $P10_7$, X_{IN}, RESET, CNV_{SS1}, BYTE, VSYNC1,OSC1,HSYNC SCL5,SDA5,SCL6,SDA6	0.8V_{CC1}		V_{CC1}	V
		$P7_0$, $P7_1$, SCL4,SDA4	0.8V_{CC1}		6.5	V
V_{IL}	LOW input voltage	$P3_1$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$	0		0.2V_{CC1}	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (during single-chip mode)	0		0.2V_{CC1}	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (data input during memory expansion and microprocessor modes)	0		0.16V_{CC1}	V
		$P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_2$, $P8_3$, $P8_6$, $P8_7$, $P9_0$ to $P9_1$, $P10_3$ to $P10_7$, X_{IN}, RESET, CNV_{SS1}, BYTE, VSYNC1,OSC1,HSYNC, SCL4,SDA4,SCL5,SDA5,SCL6,SDA6	0		0.2V_{CC1}	V
$I_{OH (peak)}$	HIGH peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$,$P7_2$ to $P7_7$, $P8_2$ to $P8_3$,$P8_6$,$P8_7$,$P9_0$,$P9_1$,$P10_3$ to $P10_7$, R,G,B,OUT1,OUT2,OSCOUT,DIGR1,DIGR2, DIGG1,DIGG2,DIGB1,DIGB2			-10.0	mA
$I_{OH (avg)}$	HIGH average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$,$P7_2$ to $P7_7$, $P8_2$,$P8_3$,$P8_6$,$P8_7$,$P9_0$,$P9_1$,$P10_3$ to $P10_7$, R,G,B,OUT1,OUT2,OSCOUT,DIGR1,DIGR2, DIGG1,DIGG2,DIGB1,DIGB2			-5.0	mA
$I_{OL (peak)}$	LOW peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$,$P7_0$ to $P7_7$, $P8_2$,$P8_3$,$P8_6$,$P8_7$,$P9_0$,$P9_1$,$P10_3$ to $P10_7$, R,G,B,OUT1,OUT2,OSCOUT,DIGR1,DIGR2,DIGG1,DIGG2, DIGB1,DIGB2, SCL4, SDA4, SCL5, SDA5, SCL6,SDA6			10.0	mA
$I_{OL (avg)}$	LOW average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$,$P7_0$ to $P7_7$, $P8_2$,$P8_3$,$P8_6$,$P8_7$,$P9_0$,$P9_1$,$P10_3$ to $P10_7$, R,G,B,OUT1,OUT2,OSCOUT,DIGR1,DIGR2,DIGG1,DIGG2, DIGB1,DIGB2, SCL4, SDA4, SCL5, SDA5, SCL6,SDA6			5.0	mA
f (X_{IN})	Main clock input oscillation frequency			16		MHz
f (X_{CIN})	Sub-clock oscillation frequency			32.768		kHz
f_{OSC}	Oscillation frequency (for OSD)	OSC1 LC oscillation mode	8.0		30	MHz
		OSC1 Ceramic oscillation mode	20.0		30	
		OSC1 Internal oscillation mode (X_{IN}=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	–	
V_I	Input amplitude	Video signal CV_{IN1}, CV_{IN2}	1.5	1.75	2.00	V

Note 1: Referenced to Vcc = V_{CC1} = V_{CC2} = V_{CC3} = 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 I_{OL} (peak) for ports P0, P1, P2, $P8_6$, $P8_7$ and P9 must be 80mA max. The total I_{OL} (peak) for ports P3, P4, P5, P6, P7 and $P8_0$ to $P8_4$ must be 80mA max. The total I_{OH} (peak) for ports P0, P1, and P2 must be -40mA max. The total I_{OH} (peak) for ports P3, P4 and P5 must be -40mA max. The total I_{OH} (peak) for ports P6, P7, and $P8_0$ to $P8_4$ must be -40mA

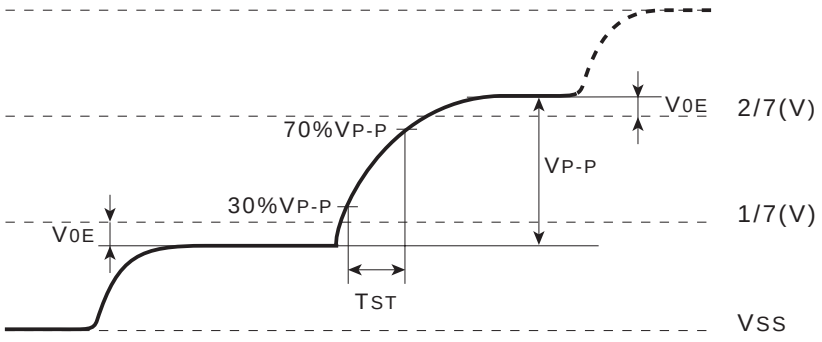
Table 18.3. A/D Conversion Characteristics (Note 1)

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
t_{CONV}	Conversion time, Sample & hold function available	$Ø_{AD}$=10 MHz	2.8			µs
t_{SAMP}	Sampling time		0.3			µs
V_{REF}	Reference voltage			V_{CC1}		V
V_{IA}	Analog input voltage		0		V_{CC1}	V

Note 1: Referenced to V_{CC1}=3.3V, V_{SS}=0V at Topr = -20 to 70°C unless otherwise specified.
Note 2: AD operation clock frequency ($Ø_{AD}$ frequency) must be 10 MHz or less. And divide the f_{AD} and make $Ø_{AD}$ frequency equal to or lower than f_{AD}/2.
Note 3: A case without sample & hold function turn $Ø_{AD}$ frequency into 250 kHz or more in addition to a limit of Note 2.
A case with sample & hold function turn $Ø_{AD}$ 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, Ta=25°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	α × 0.8	$\alpha = \frac{100}{140} \fallingdotseq 0.71$	α × 1.2	V
Voe	Output deviation				±20	%
I_O	Maximum output current	RGB each output control bit=111b setup	2.2	4.0	5.8	mA
R_O	Output register		190		400	ΩA
T_{st}	Set ring time	30 to 70% or 70 to 30%			33	nS



* This figure is the publication about 0/7=Vss, 1/7, and 2/7 in maximum output amplitude Vppm=7/7(V).

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 V_{CC1}=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)

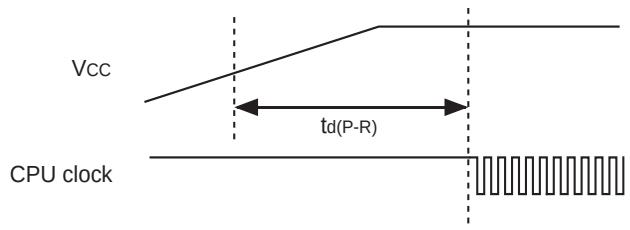
Flash program, erase voltage	Flash read operation voltage
V_{CC1} = 3.3 V ± 0.15V	V_{CC1}=3.3 to 0.15 V

Table 18.7. Power Supply Circuit Timing Characteristics

Symbol	Parameter	Measuring Condition	Standard			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Time for Internal Power Supply Stabilization During Powering-On	V_{CC1}=2.7 to 3.45V			2	ms
$t_{d(R-S)}$	STOP Release Time				150	µs
$t_{d(W-S)}$	Low Power Dissipation Mode Wait Mode Release Time				150	µs

Note: When V_{CC1} = 5V.

$t_{d(P-R)}$
Time for Internal Power Supply Stabilization During Powering-On



$t_{d(R-S)}$
STOP Release Time

$t_{d(W-S)}$
Low Power Dissipation Mode Wait Mode Release Time

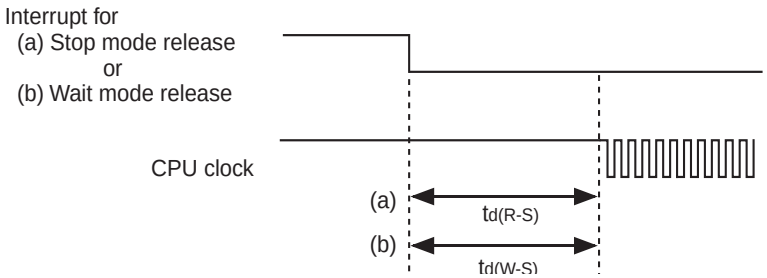


Table 18.8. Electrical Characteristics (Note 1)

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 $P7_7$,$P8_2$,$P8_3$,$P8_6$,$P8_7$, $P9_0$,$P9_1$,$P10_3$ 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_{COUT}	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 $P7_7$,$P8_2$,$P8_3$,$P8_6$,$P8_7$, $P9_0$,$P9_1$,$P10_3$ 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_{COUT}	HIGHPOWER	With no load applied		0		V
			LOWPOWER	With no load applied		0		
V_{T+}-V_{T-}	Hysteresis	$\overline{HOLD}$,$\overline{RDY}$,$TA0_{IN}$ to $TA3_{IN}$,$TB0_{IN}$,$TB1_{IN}$,$\overline{INT_0}$ to $\overline{INT_3}$, $\overline{CTS_0}$ to $\overline{CTS_2}$,SCL0 to SCL6,SDA0 to SDA6, CLK_0 to CLK_2,$TA0_{OUT}$ to $TA3_{OUT}$,KI_1 to KI_3,RxD_0 to RxD_2 VSYNC1,VSYNC2,HC0,HC1,HSYNC2			0.2		0.8	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$			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 $P7_7$,$P8_2$,$P8_3$,$P8_6$,$P8_7$, $P9_0$,$P9_1$,$P10_3$ to $P10_7$,X_{IN},$\overline{RESET}$,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 $P7_7$,$P8_2$,$P8_3$,$P8_6$,$P8_7$, $P9_0$,$P9_1$,$P10_3$ to $P10_7$,X_{IN},$\overline{RESET}$,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 $P7_7$,$P8_2$,$P8_3$,$P8_6$,$P8_7$, $P9_0$,$P9_1$,$P10_3$ to $P10_7$		V_I = 0V	66	160	500	kΩ
R_{fXIN}	Feedback resistance	X_{IN}				3.0		MΩ
R_{fXCIN}	Feedback resistance	X_{CIN}				25		MΩ
R_{BS}	I^2C-bus, Bus switch						130	Ω

Note 1: Referenced to V_{CC} = V_{CC1} = V_{CC2} = V_{CC3} = 3.3V, V_{SS} = 0V, Topr = -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		OSD ON Data slicer ON		120	170	mA
					OSD OFF Data slicer OFF		15		mA
			Mask ROM	f(X_{CIN})=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 V_{CC} = V_{CC1} = V_{CC2} = V_{CC3} = 3.3V, V_{SS} = 0V, Topr = -20 to 70 °C unless otherwise specified.
Note 2: With one timer operated using f_{C32}.
Note 3: This indicates the memory in which the program to be executed exists.

Timing Requirements

(V_{CC1} = V_{CC2} = V_{CC3} = 3.3V, V_{SS} = 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
$t_{su(RDY-BCLK)}$	$\overline{RDY}$ input setup time	40		ns
$t_{su(HOLD-BCLK)}$	$\overline{HOLD}$ input setup time	50		ns
$t_{h(RD-DB)}$	Data input hold time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{RDY}$ input hold time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{HOLD}$ input hold time	0		ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 60 \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 60 \quad [ns]$$

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:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 60 \quad [ns]$$

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
$t_{su(TAIN-TAOUT)}$	TAi_{OUT} input setup time	500		ns
$t_{su(TAOUT-TAIN)}$	TAi_{IN} input setup time	500		ns

Timing Requirements

(V_{CC1} = V_{CC2} = V_{CC3} = 3.3V, V_{SS} = 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.	
$t_{c(TB)}$	TBi_{IN} input cycle time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width (counted on one edge)	60		ns
$t_{c(TB)}$	TBi_{IN} input cycle time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width (counted on both edges)	160		ns
$t_{w(TBL)}$	TBi_{IN} 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.	
$t_{c(TB)}$	TBi_{IN} input cycle time	600		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width	300		ns

Table 18.20. Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	600		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width	300		ns

Table 18.21. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	100		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 18.22. External Interrupt $\overline{INTi}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	380		ns

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 18.23. 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
$t_{d(BCLK-HLDA)}$	$\overline{HLDA}$ output delay time			40	ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 40 \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

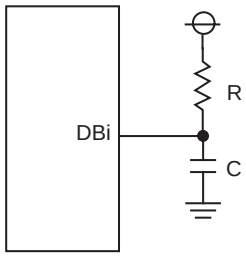
$$\frac{0.5 \times 10^9}{f(BCLK)} - 10 \quad [ns]$$

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 = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30pF \times 1k\Omega \times \ln (1 - 0.2V_{CC2} / V_{CC2})$$
$$= 6.7ns.$$



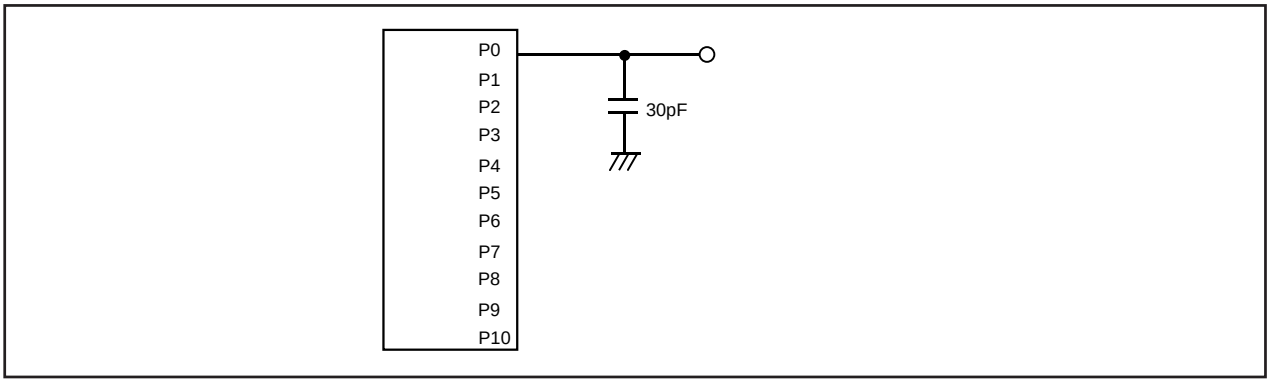


Figure 18.2. Ports P0 to P10 Measurement Circuit

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 18.24. 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
$t_{d(BCLK-HLDA)}$	$\overline{HLDA}$ output delay time			40	ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 40 \quad [ns]$$

n is "1" for 1-wait setting, "2" for 2-wait setting and "3" for 3-wait setting.

Note 2: Calculated according to the BCLK frequency as follows:

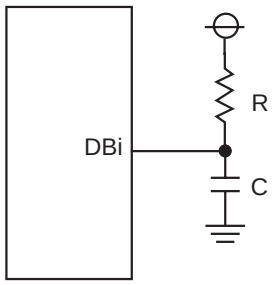
$$\frac{0.5 \times 10^9}{f(BCLK)} - 10 \quad [ns]$$

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 = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30pF \times 1k\Omega \times \ln (1 - 0.2V_{CC2} / V_{CC2})$$



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
$t_{d(BCLK-HLDA)}$	$\overline{HLDA}$ 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 Adderss)		(Note 4)		ns
$t_{d(AD-RD)}$	RD signal output delay from the end of Adress		0		ns
$t_{d(AD-WR)}$	WR signal output delay from the end of Adress		0		ns
$t_{dZ(RD-AD)}$	Address output floating start time			8	ns

Note 1: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 10 \quad [ns]$$

Note 2: Calculated according to the BCLK frequency as follows:

$$\frac{(n-0.5) \times 10^9}{f(BCLK)} - 50 \quad [ns]$$

n is "2" for 2-wait setting, "3" for 3-wait setting.

Note 3: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 40 \quad [ns]$$

Note 4: Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 15 \quad [ns]$$

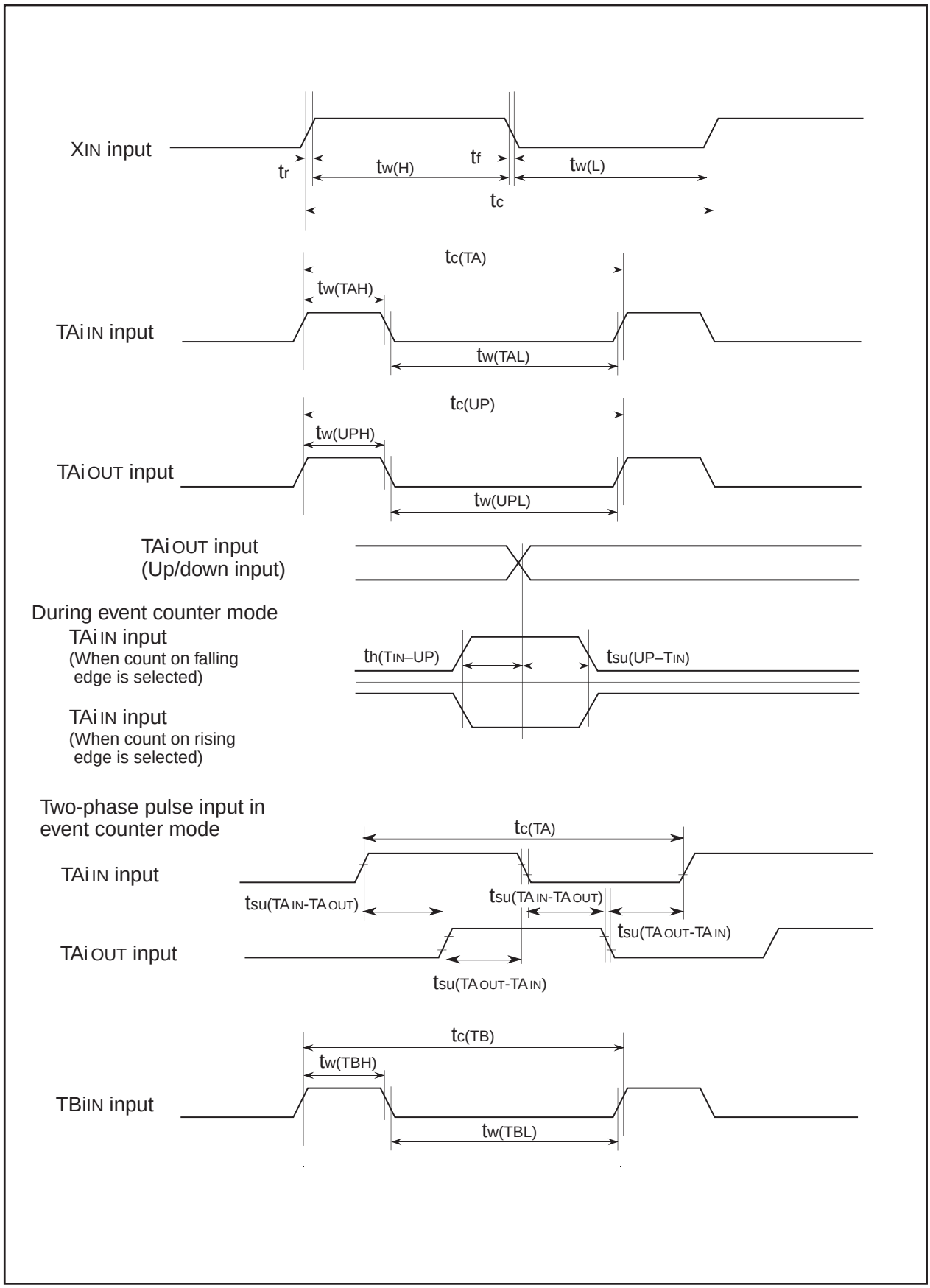


Figure 18.3. Timing Diagram (1)

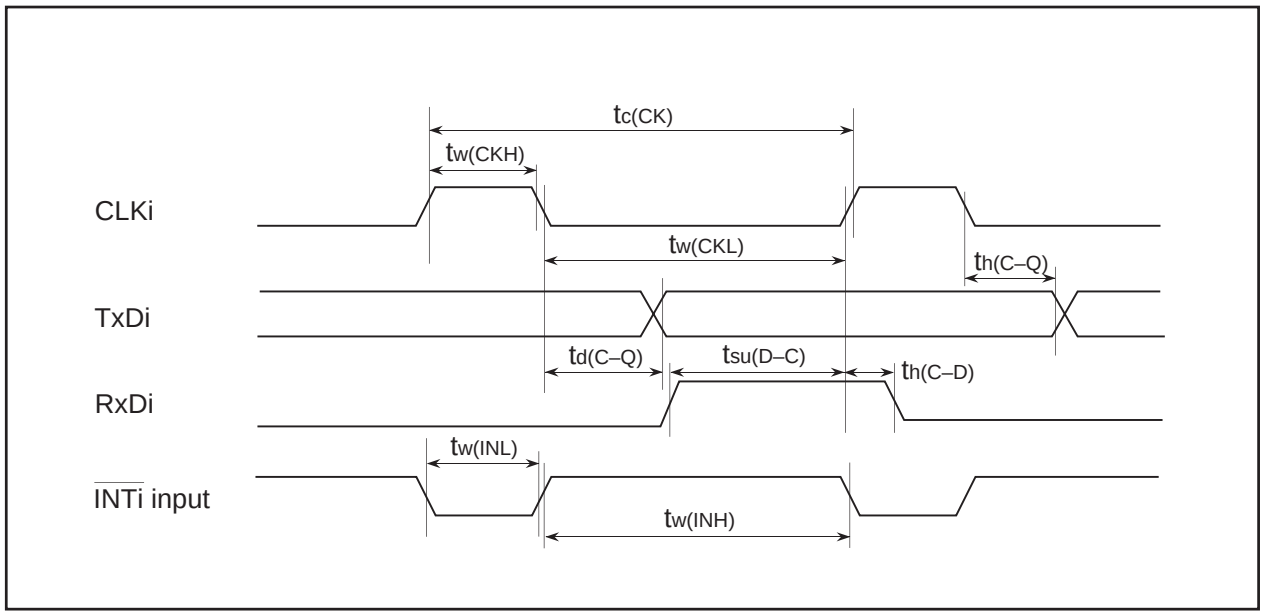
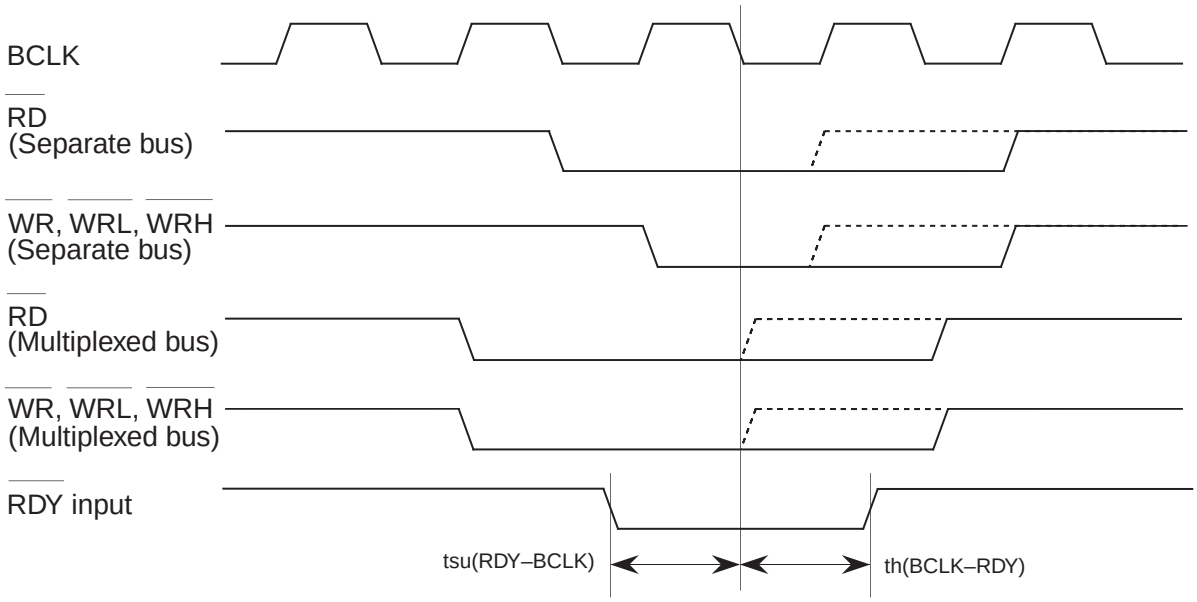


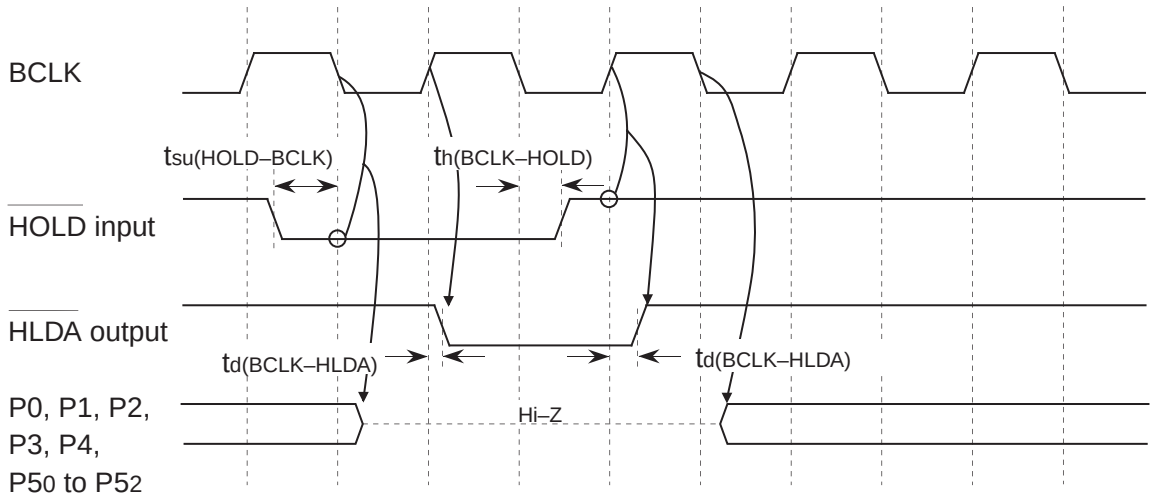
Figure 18.4. Timing Diagram (2)

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.

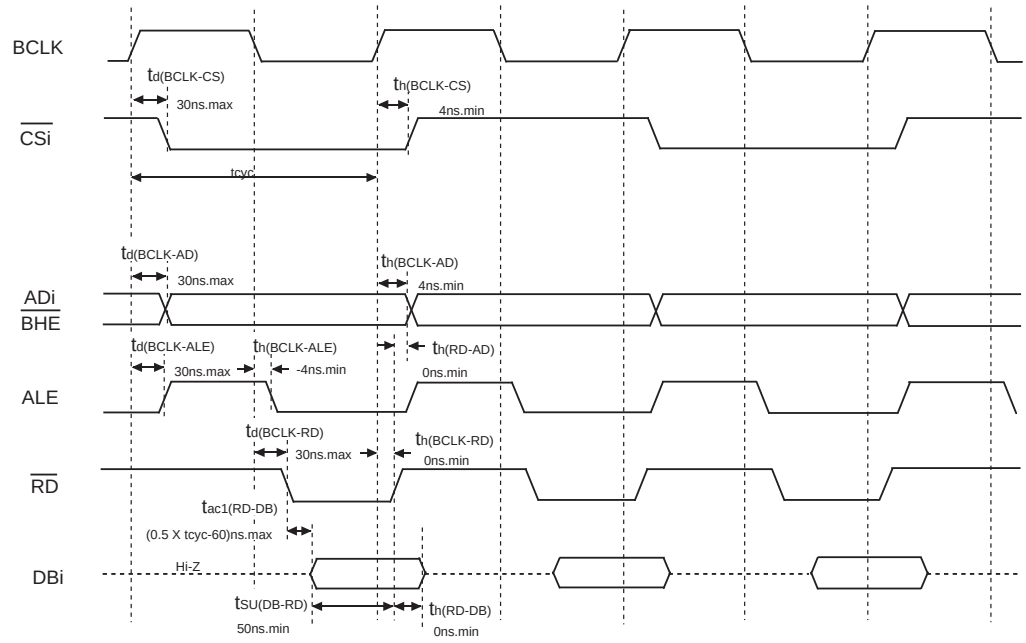
Measuring conditions :

- $V_{CC1}=V_{CC2}=V_{CC3}=3.3V$
- Input timing voltage : Determined with $V_{IL}=0.6V$, $V_{IH}=2.4V$
- Output timing voltage : Determined with $V_{OL}=1.5V$, $V_{OH}=1.5V$

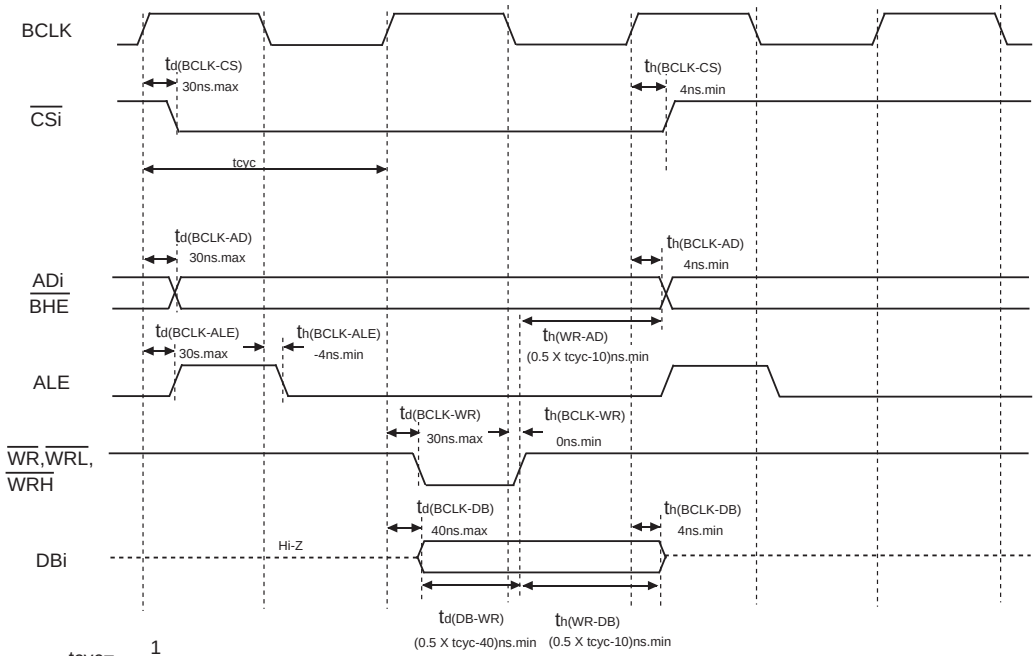
Figure 18.5. Timing Diagram (3)

Memory Expansion Mode, Microprocessor Mode
(For setting with no wait)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

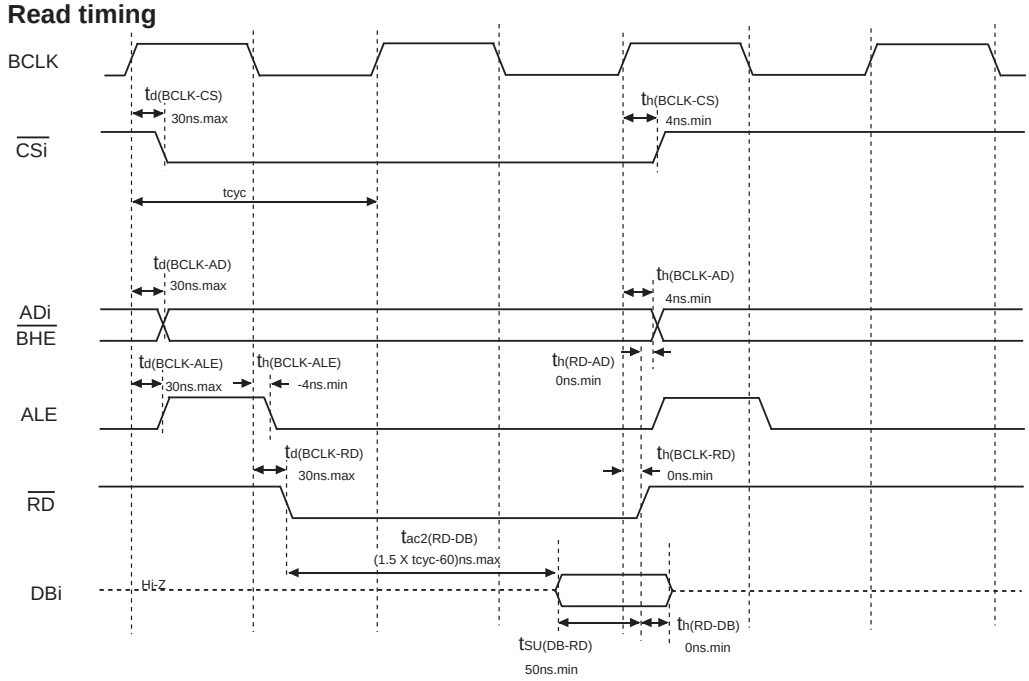
Measuring conditions
- $V_{CC1}=V_{CC2}=V_{CC3}=3.3V$
- Input timing voltage : $V_{IL}=0.6V$, $V_{IH}=2.4V$
- Output timing voltage : $V_{OL}=1.5V$, $V_{OH}=1.5V$

Figure 18.6. Timing Diagram (4)

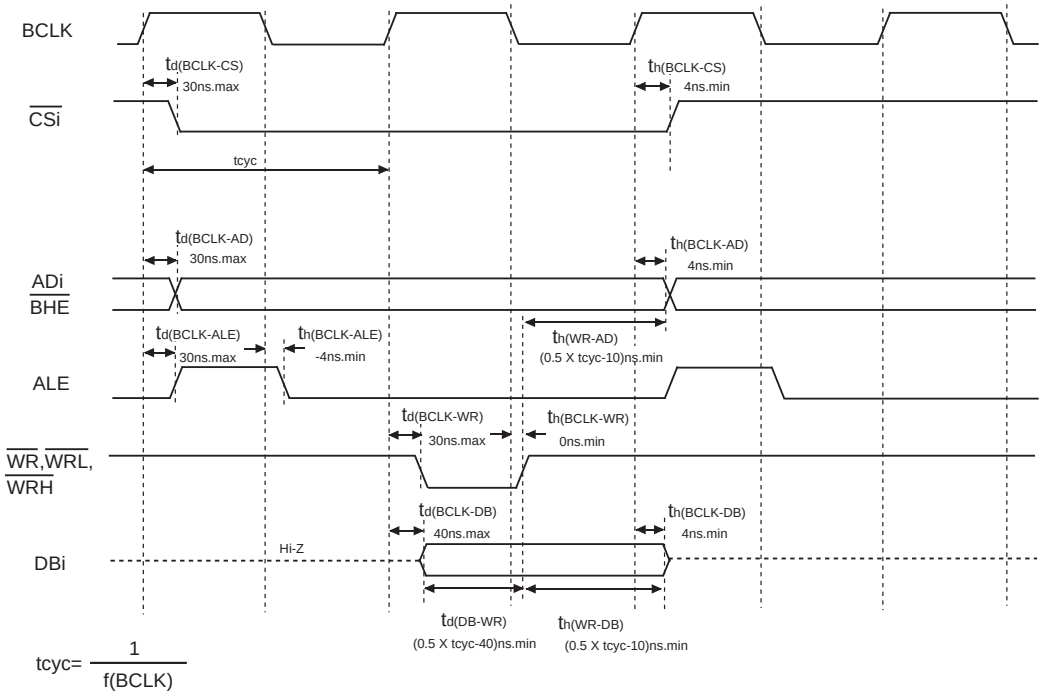
Memory Expansion Mode, Microprocessor Mode

(for 1-wait setting and external area access)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions

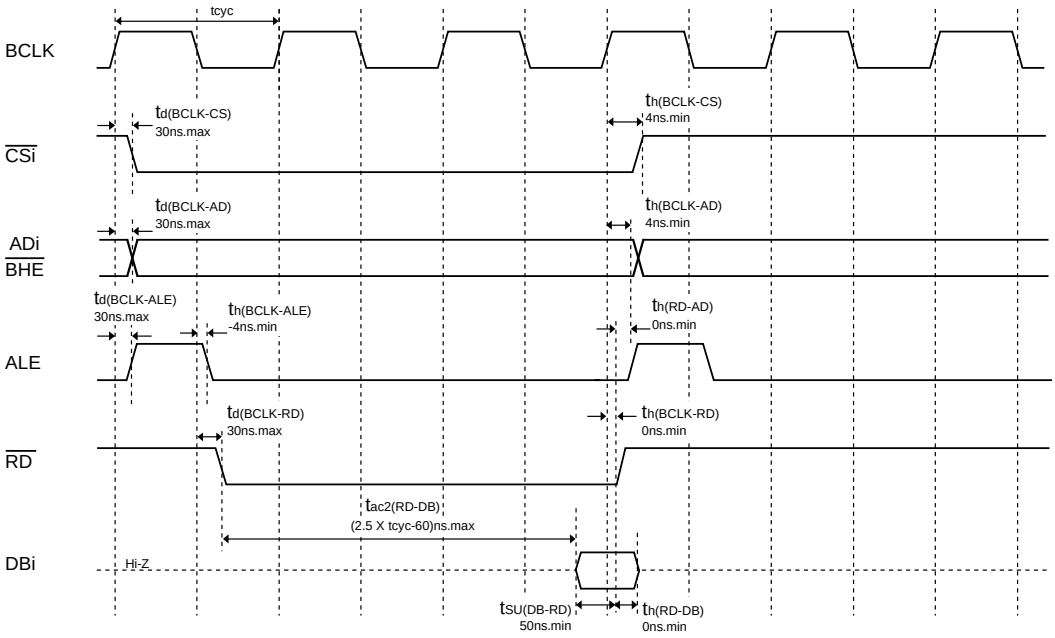
- $V_{CC1}=V_{CC2}=V_{CC3}=3V$
- Input timing voltage : $V_{IL}=0.6V$, $V_{IH}=2.4V$
- Output timing voltage : $V_{OL}=1.5V$, $V_{OH}=1.5V$

Figure 18.7. Timing Diagram (5)

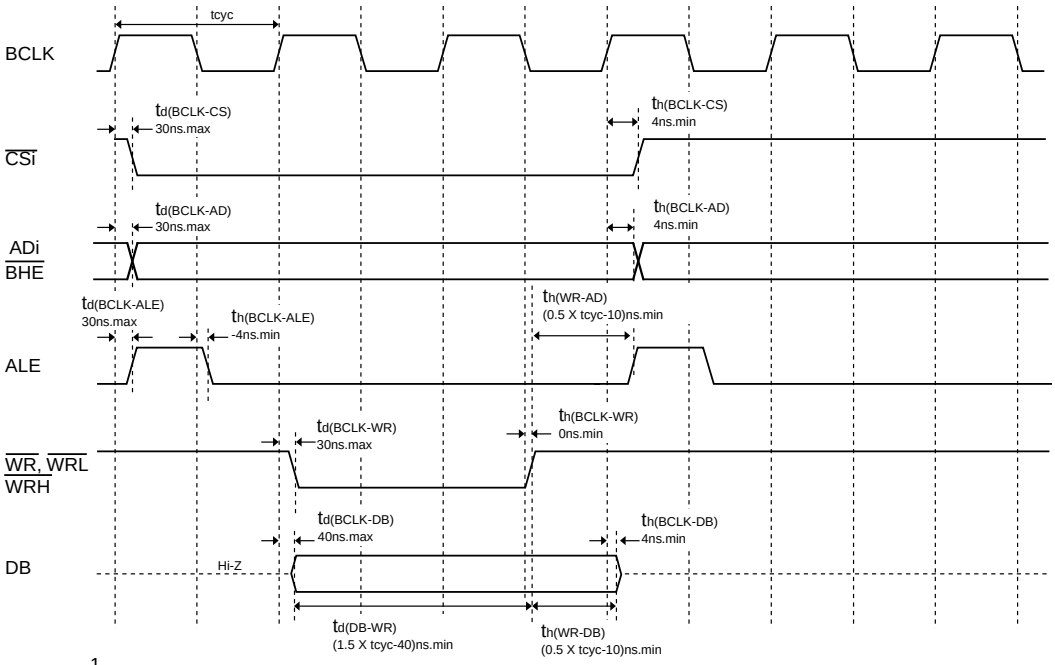
Memory Expansion Mode, Microprocessor Mode

(for 2-wait setting and external area access)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions

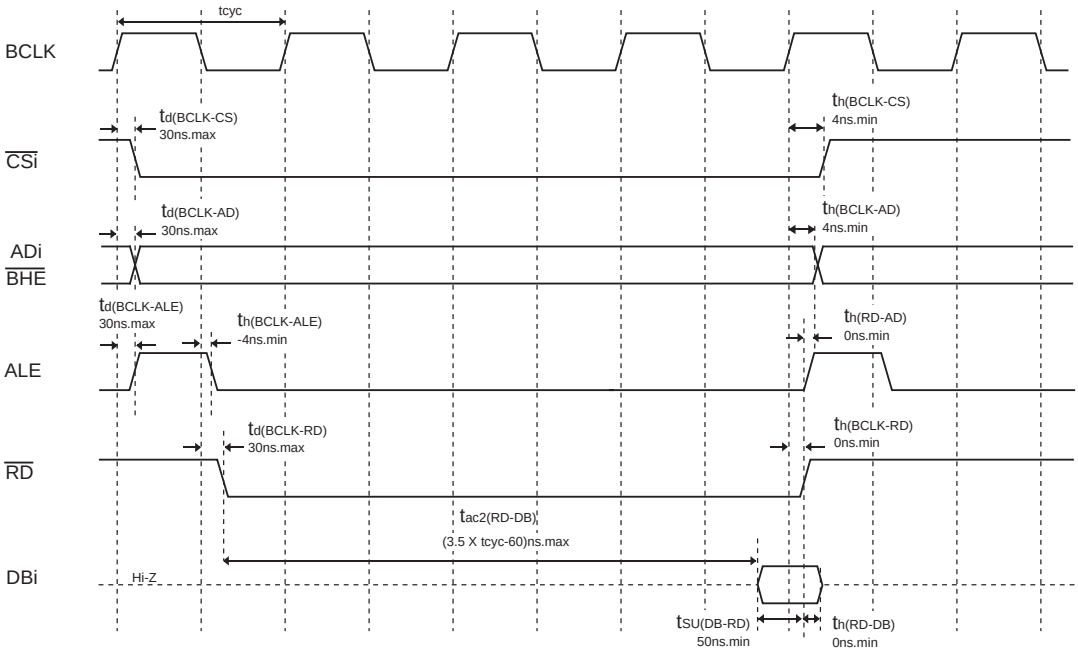
- $V_{CC1}=V_{CC2}=V_{CC3}=3.3V$
- Input timing voltage : $V_{IL}=0.6V$, $V_{IH}=2.4V$
- Output timing voltage : $V_{OL}=1.5V$, $V_{OH}=1.5V$

Figure 18.8. Timing Diagram (6)

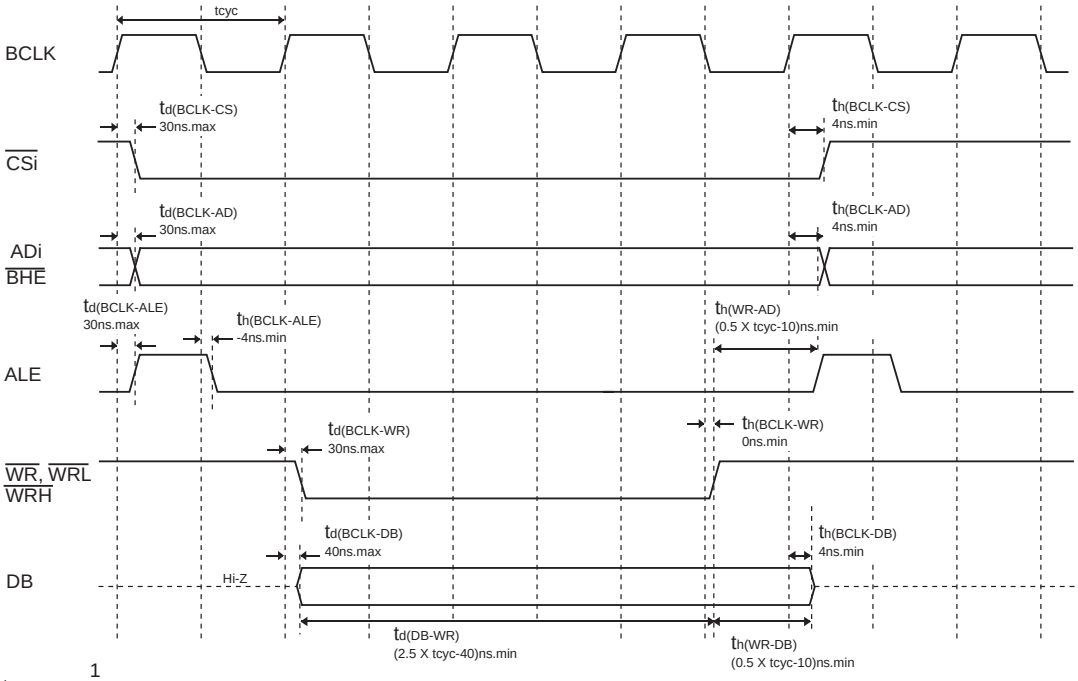
Memory Expansion Mode, Microprocessor Mode

(for 3-wait setting and external area access)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions

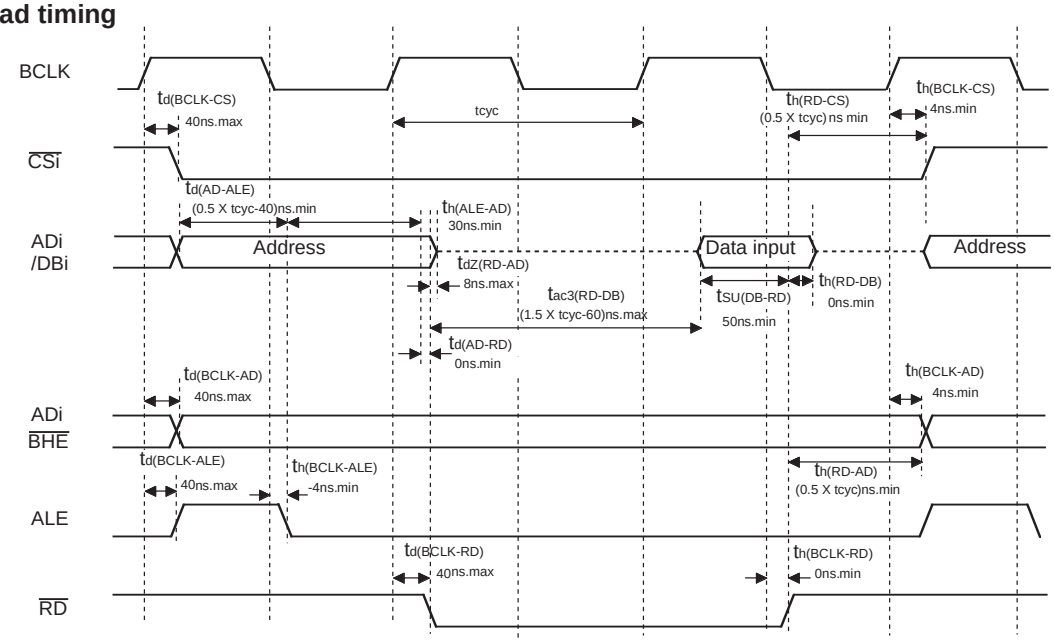
- $V_{CC1}=V_{CC2}=V_{CC3}=3.3V$
- Input timing voltage : $V_{IL}=0.6V$, $V_{IH}=2.4V$
- Output timing voltage : $V_{OL}=1.5V$, $V_{OH}=1.5V$

Figure 18.9. Timing Diagram (7)

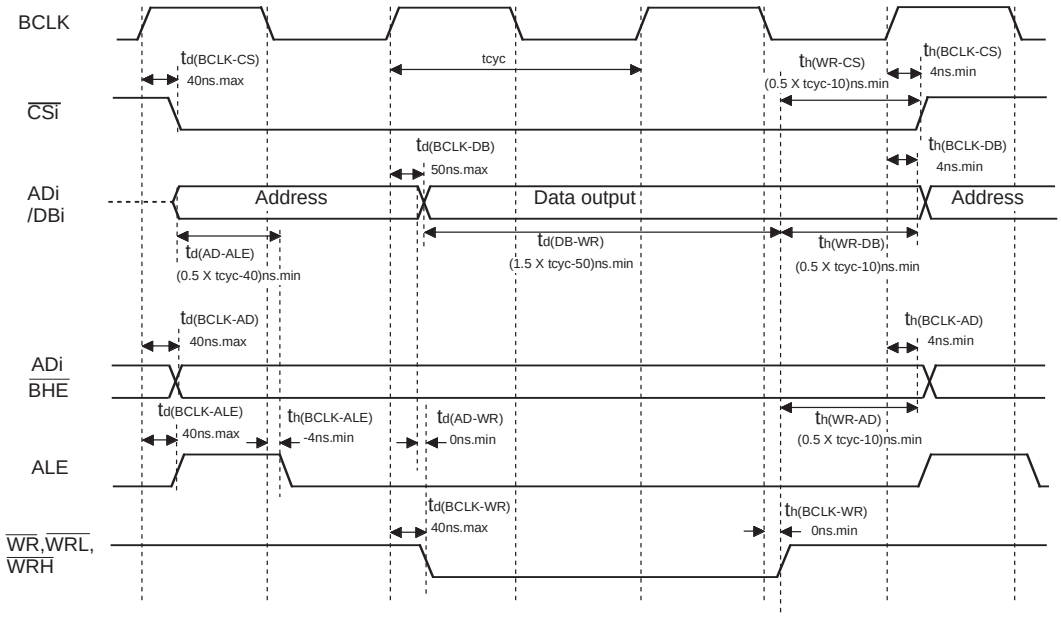
Memory Expansion Mode, Microprocessor Mode

(For 2-wait setting, external area access and multiplex bus selection)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

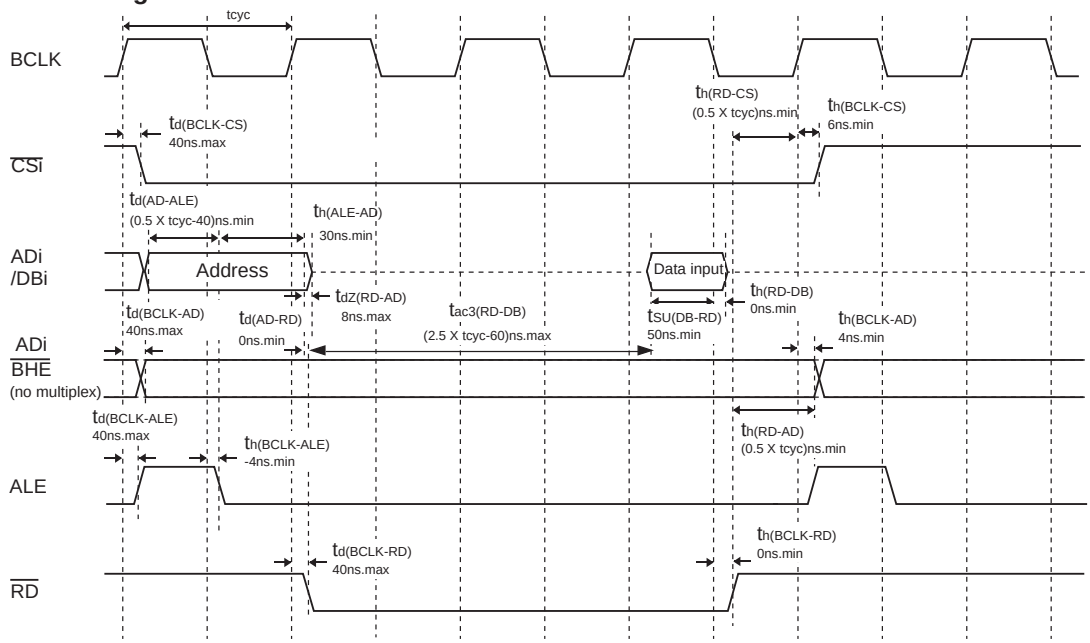
Measuring conditions
- $V_{CC1}=V_{CC2}=V_{CC3}=3.3V$
- Input timing voltage : $V_{IL}=0.6V$, $V_{IH}=2.4V$
- Output timing voltage : $V_{OL}=1.5V$, $V_{OH}=1.5V$

Figure 18.10. Timing Diagram (8)

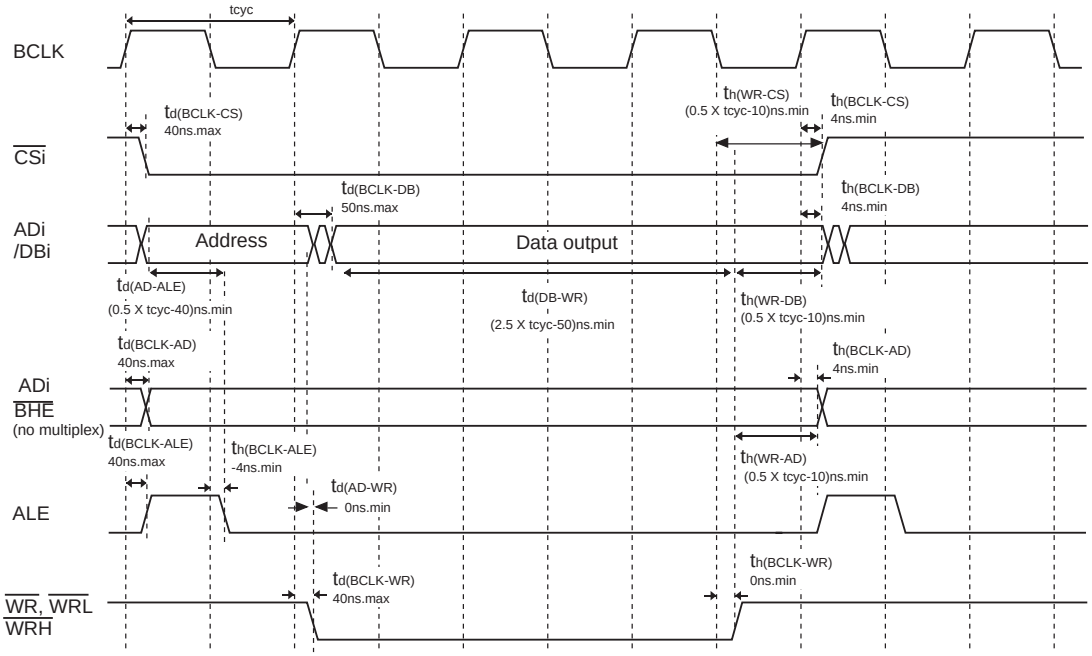
Memory Expansion Mode, Microprocessor Mode

(For 3-wait setting, external area access and multiplex bus selection)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions
- V_{CC1}=V_{CC2}=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 18.11. 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”.

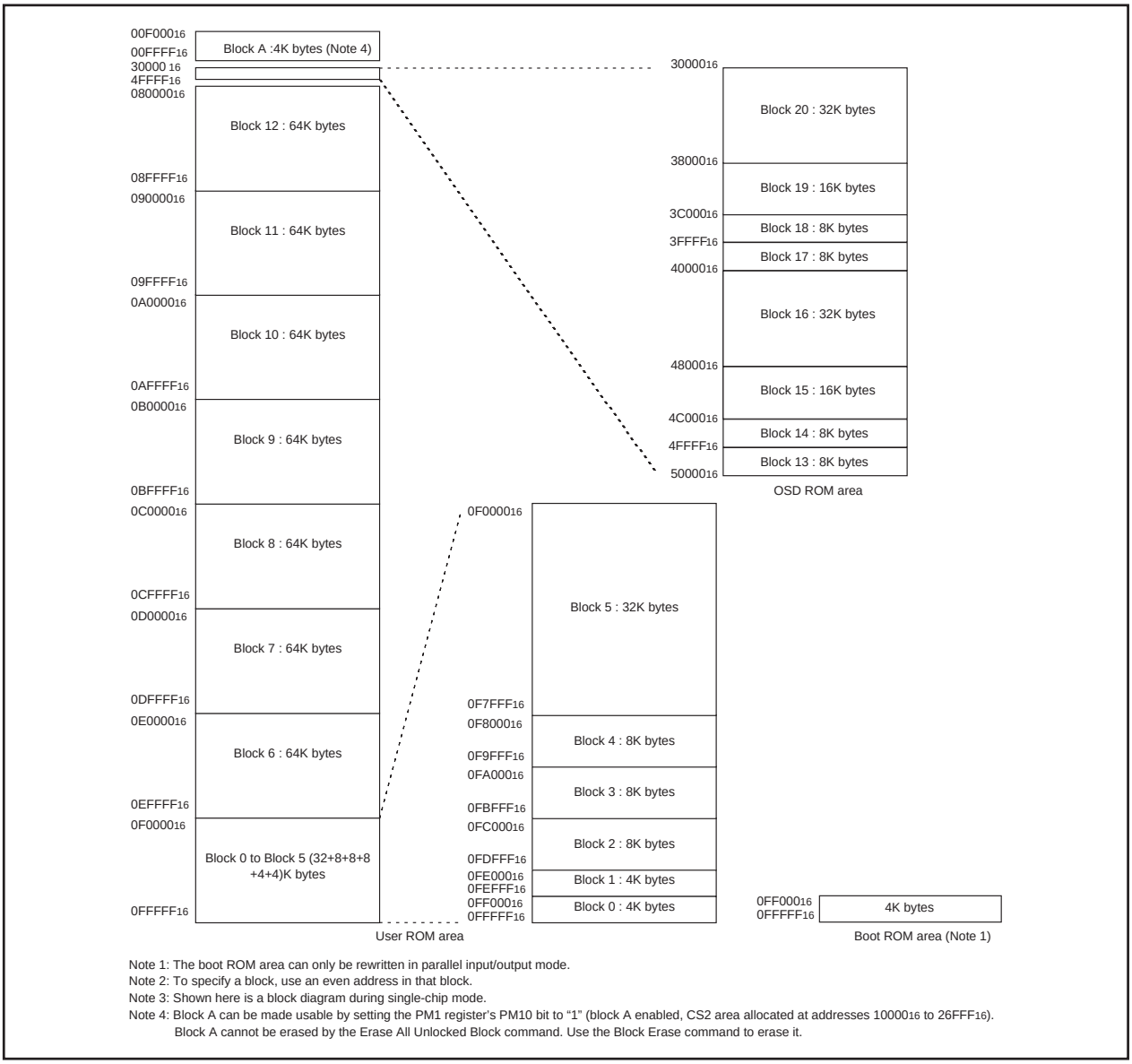
Note 2: When in CPU rewrite mode, the PM10 and PM13 bits in the PM1 register are set to “1”. The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1. When the PM13 bit = 0 and the flash memory is used in 4M-byte mode, the extended accessible area (50000_{16} to $BFFFF_{16}$) 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 momoery. The user ROM area has a 4K-byte block A, in addition to the area that stores a program for microcomputer operation during singe-chip or memory expansion mode.

The user ROM area is divided into several blocks, each of which can individually be protected (locked) against programming or erasure. The user ROM area can be rewritten in all of CPU rewrite, standard serial input/output, and parallel input/output modes. Block A is enabled for use by setting the PM1 register's PM10 bit to "1" (block A enabled, CS2 area at addresses 10000_{16} to $26FFF_{16}$).

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 CNV_{SS1} and $P5_0$ pins and a low-level signal to the $P5_5$ 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 CNV_{SS1} 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.



Note 1: The boot ROM area can only be rewritten in parallel input/output mode.
Note 2: To specify a block, use an even address in that block.
Note 3: Shown here is a block diagram during single-chip mode.
Note 4: Block A can be made usable by setting the PM1 register's PM10 bit to "1" (block A enabled, CS2 area allocated at addresses 10000_{16} to $26FFF_{16}$).
Block A cannot be erased by the Erase All Unlocked Block command. Use the Block Erase command to erase it.

Figure 19.1. Flash Memory Block Diagram

Boot Mode

After a hardware reset which is performed by applying a low-level signal to the P5$_5$ pin and a high-level signal to the CNV$_{SS1}$ and P5$_0$ 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 0FFFDFh, 0FFFE3h, 0FFFEBh, 0FFFEFh, 0FFFF3h, 0FFFF7h, and 0FFFFBh. The flash memory must have a program with the ID codes set in these addresses.

ROM code protect control address

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

Symbol: ROMCP
Address: 0FFFFF16
Value when shipped: FF16 (Note 4)

Bit symbol	Bit name	Function	RW
———	Reserved bit	Set this bit to “1”	RW
———	Reserved bit	Set this bit to “1”	RW
———	Reserved bit	Set this bit to “1”	RW
———	Reserved bit	Set this bit to “1”	RW
———	Reserved bit	Set this bit to “1”	RW
ROMCP1	ROM code protect level 1 set bit (Note 1, Note 3, Note 4)	b7 b6 00: 01: 10: } Protect enabled 11: Protect disabled	RW RW

Notes 1: If the ROMCP1 bits are set to other than "11b" (ROM code protect enabled), the flash memory is disabled against reading and rewriting in parallel I/Ot mode.
2: When the ROMCP1 bits are set to other than "11b," set the bit 5 to bit 0 to "111111b".
3: When exiting ROM code protect, erase the block including the ROMCP1 register by CPU rewrite mode or standard serial I/O mode.
4: If a memory block that including ROMCP1 register is erased, the ROMCP register is set to "FFh".

Figure 19.2. ROMCP Register

Address		
0FFFDF16 to 0FFFDC16	ID1	Undefined instruction vector
0FFFE316 to 0FFFE016	ID2	Overflow vector
0FFFE716 to 0FFFE416		BRK instruction vector
0FFFEB16 to 0FFFE816	ID3	Address match vector
0FFFEF16 to 0FFFEC16	ID4	Single step vector
0FFFF316 to 0FFFF016	ID5	Watchdog timer vector
0FFFF716 to 0FFFF416	ID6	$\overline{\text{DBC}}$ vector
0FFFFB16 to 0FFFF816	ID7	
0FFFFF16 to 0FFFFC16	ROMCP	Reset vector

4 bytes

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	• Single chip mode • Memory expansion mode • Boot mode	Single chip mode
Areas in which a rewrite control program can be located	• 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	• 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)	• 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 (50000_{16} to $BFFFF_{16}$) 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.

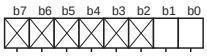
When the FMR11 bit is set to “1” (EW1 mode), the MCU (microcomputer) enters EW1 mode.

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”.

Flash memory (USER) identification register



Symbol	Address	After reset
FIDR	$01B4_{16}$	$XXXXXX00_2$

Bit symbol	Bit name	Function	RW
FIDR0	Flash module type identification value	b1 b0 0 0: M16C/62N, M3062GF8N type flash module 1 0: M16C/62P type flash module 1 1: M16C/62M, M16C/62A type flash module	RO
FIDR1			RO
(b7-b2)	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.		—

Note: This register identifies on-chip flash module type of M16C/62 group. Note, however, no chip version is known by this register. Follow the procedure described below for the identification.
(1) Write FF_{16} to FIDR register
(2) Read FIDR register
(3) Check two low-order bits of read value
Make sure no access to external memories or other SFRs or no interrupts or DMA transfers will occur between the above two instructions no. 1 and no. 2. FIDR register does not discriminate the type of built-in flash module, and does not show a chip version.

Flash memory (USER) ontrol register 0

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

Bit symbol	Bit name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0: Busy (being written or erased) 1: Ready	RO
FMR01	CPU rewrite mode select bit (Note 1)	0: Disables CPU rewrite mode 1: Enables CPU rewrite mode	RW
FMR02	Lock bit disable select bit (Note 2)	0: Enables lock bit 1: Disables lock bit	RW
FMSTP	Flash memory stop bit (Note 3, Note 5)	0: Enables flash memory operation 1: Stops flash memory operation (placed in low power mode, flash memory initialized)	RW
(b4)	Reserved bit	Must always be set to "0"	RW
FMR05	User ROM area select bit (Note 3) (Effective in only boot mode)	0: Boot ROM area is accessed 1: User ROM area is accessed	RW
FMR06	Program status flag (Note 4)	0: Terminated normally 1: Terminated in error	RO
FMR07	Erase status flag (Note 4)	0: Terminated normally 1: Terminated in error	RO

Note 1: To set this bit to "1", write "0" and then "1" in succession. Make sure no interrupts or DMA transfers will occur before writing "1" after writing "0".
Also, while in EW0 mode, write to this bit from a program in other than the flash memory.
Note 2: To set this bit to "1", write "0" and then "1" in succession when the FMR01 bit = 1. Make sure no interrupts or no DMA transfers will occur before writing "1" after writing "0".
Note 3: Write to this bit from a program in other than the flash memory.
Note 4: This flag is cleared to "0" by executing the Clear Status command.
Note 5: Effective when the FMR01 bit = 1 (CPU rewrite mode). If the FMR01 bit = 0, although the FMR03 bit can be set to "1" by writing "1" in a program, the flash memory is neither placed in low power mode nor initialized.
Note 6: This status includes writing or reading with the Lock Bit Program or Read Lock Bit Status command.

Flash memory (USER) control register 1

Symbol	Address	After reset
FMR1	$01B5_{16}$	$0X00XX0X_2$

Bit symbol	Bit name	Function	RW
(b0)	Reserved bit	The value in this bit when read is indeterminate.	RO
FMR11	EW1 mode select bit (Note)	0: EW0 mode 1: EW1 mode	RW
(b3-b2)	Reserved bit	The value in this bit when read is indeterminate.	RO
(b5-b4)	Reserved bit	Must always be set to "0"	RW
FMR06	Lock bit status flag	0: Lock 1: Unlock	RO
(b7)	Reserved bit	Must always be set to "0"	RW

Note : To set this bit to "1", write "0" and then "1" in succession when the FMR01 bit = 1. Make sure no interrupts or no DMA transfers will occur before writing "1" after writing "0".
The FMR01 and FMR11 bits both are cleared to "0" by setting the FMR01 bit to "0".

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 resome 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.

Flash memory (USER/OSD) switching register

b7	b6	b5	b4	b3	b2	b1	b0
☒							

Symbol	Address	After reset
FMSEL	$02A0_{16}$	00000000_2

Bit symbol	Bit name	Function	RW
OSELBIT1	USER/OSD1/OSD2 switching bit	00 USER 01: OSD1 10: OSD2 11: Nothing is assigned.	RW
OSELBIT2			RW
$\overline{\text{(b6-b2)}}$	Reserved bit	Must always be set to “0”	RW
$\overline{\text{(b7)}}$	Nothing is assigned. When write, set to “0”. When read, their contents are “0”.		RW

Flash memory (OSD1/OSD2) control register 0

b7	b6	b5	b4	b3	b2	b1	b0
			0				

Symbol	Address	After reset
FMOSi0	$02A7_{16}$/$02B7_{16}$	$XX000001_2$

Bit symbol	Bit name	Function	RW
FMOSi00	RY/$\overline{\text{BY}}$ status flag	0 : Busy (being written or erased) 1 : Ready	RO
FMOSi01	CPU rewrite mode select bit (Note 1)	0 : Disables CPU rewrite mode 1 : Enables CPU rewrite mode	RW
FMOSi02	Blocks 0 and 1 rewrite enable bit (Note 2)	0 : Enables blocks 0 and 1 rewrite 1 : Disables blocks 0 and 1 rewrite	RW
FMOSiSTP	Flash memory stop bit (Note 3, Note 5)	0 : Enables flash memory operation 1 : Stops flash memory operation (placed in low power mode, flash memory initialized)	RW
$\overline{\text{(b4-b5)}}$	Reserved bit	Must always be set to “0”	RW
FMOSi06	Program status flag (Note 4)	0 : Terminated normally 1 : Terminated in error	RO
FMOSi07	Erase status flag (Note 4)	0 : Terminated normally 1 : Terminated in error	RO

(i = A, B)

Note 1: To set this bit to “1”, write “0” and then “1” in succession. Make sure no interrupts or DMA transfers will occur before writing “1” after writing “0”. While in EW0 mode, write to this bit from a program in other than the flash memory.
Note 2: To set this bit to “1”, write “0” and then “1” in succession when the FMR01 bit = 1. Make sure no interrupts or no DMA transfers will occur before writing “1” after writing “0”.
Note 3: Write to this bit from a program in other than the flash memory.
Note 4: This flag is cleared to “0” by executing the Clear Status command.
Note 5: Effective when the FMOSi01 bit = 1 (CPU rewrite mode). If the FMOSi01 bit = 0, although the FMOSiSTP bit can be set to “1” by writing “1” in a program, the flash memory is neither placed in low power mode nor initialized.

Figure 19.5-1. Register FMOSA0/FMOSB0/FMSEL

Flash memory (OSD1/OSD2) control register 1

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

Symbol	Address	After reset
FMOSi1	$02A5_{16}/02B5_{16}$	$0X00XX0X_2$

Bit symbol	Bit name	Function	RW
— (b0)	Reserved bit	Undeterminate when it reads	RO
FMOSi11	EW1 mode select bit (Note 1)	0: EW0 mode 1: EW1 mode	RW
— (b3-b2)	Reserved bit	Undeterminate when it reads	RW
— (b5-b4)	Reserved bit	Must always be set to “0”	RW
— (b6)	Nothing is assigned. When write, set to “0”. When read, their contents are “0”.		—
— (b7)	Reserved bit	Must always be set to “0”	RW

Note 1: To set this bit to “1”, write “0” and then “1” in succession when the FMR01 bit = 1. Make sure no interrupts or no DMA transfers will occur before writing “1” after writing “0.”
Both of bits FMR01 and FMR11 will be set to "0" if bit FMOSi01 is set to "0."

Flash memory (OSD1/OSD2) control register 4

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

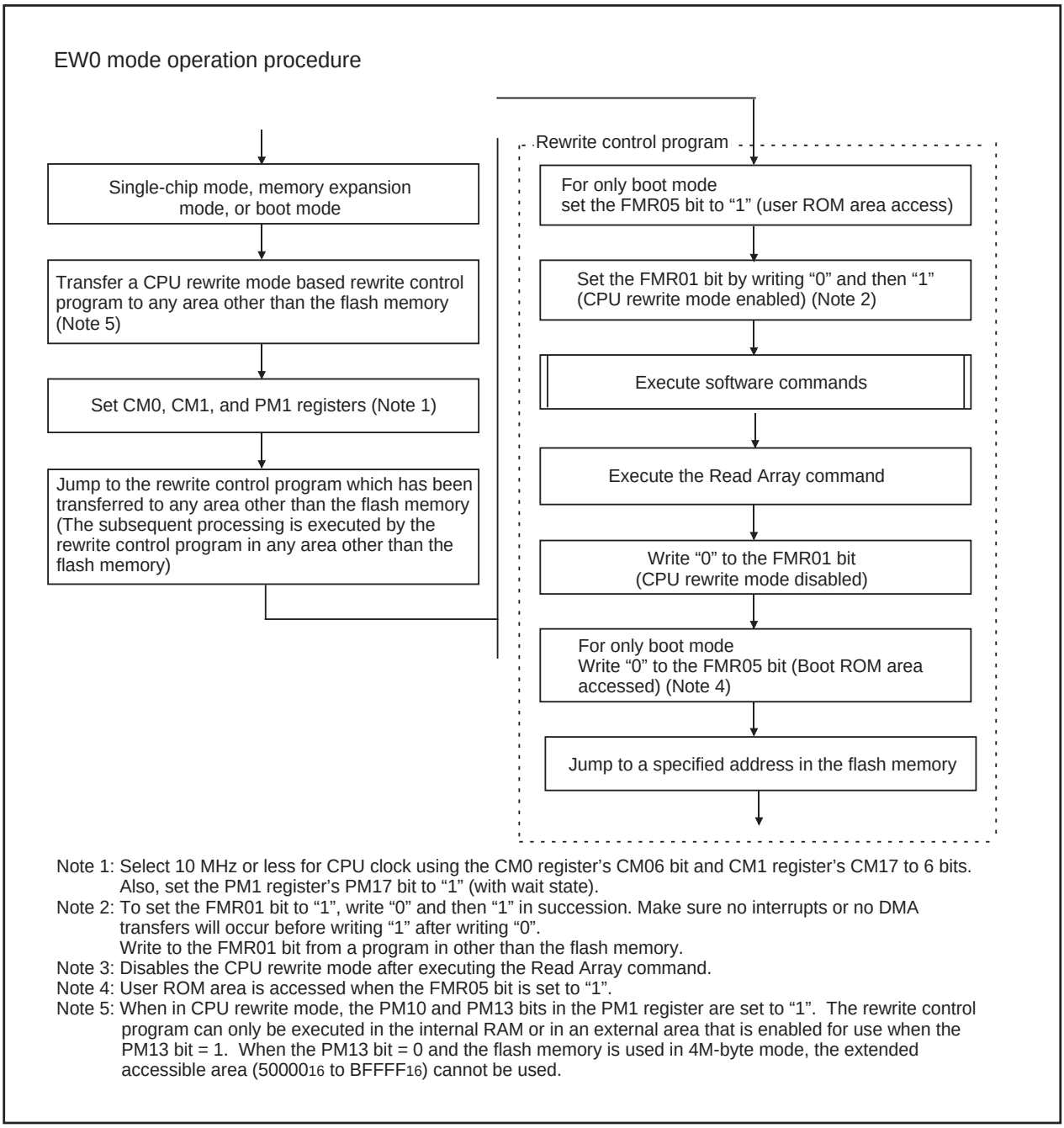
Symbol	Address	After reset
FMOSi4	$02A3_{16}/02B3_{16}$	01000000_2

Bit symbol	Bit name	Function	RW
FMOSi40	Erase suspend functional permission bit (Note 1)	0 : Enable 1 : Disable	RO
FMOSi41	Erase suspend request bit (Note 2)	0 : Erase restart 1 : Suspend request	RW
— (b5-b2)	Reserved bit	Must always be set to “0”	RW
FMOSi46	Erase status	0 : Under automatic elimination operation 1 : Automatic elimination stop (Erase suspension mode)	RW
— (b7)	Reserved bit	Must always be set to “0”	RW

(i = A, B)

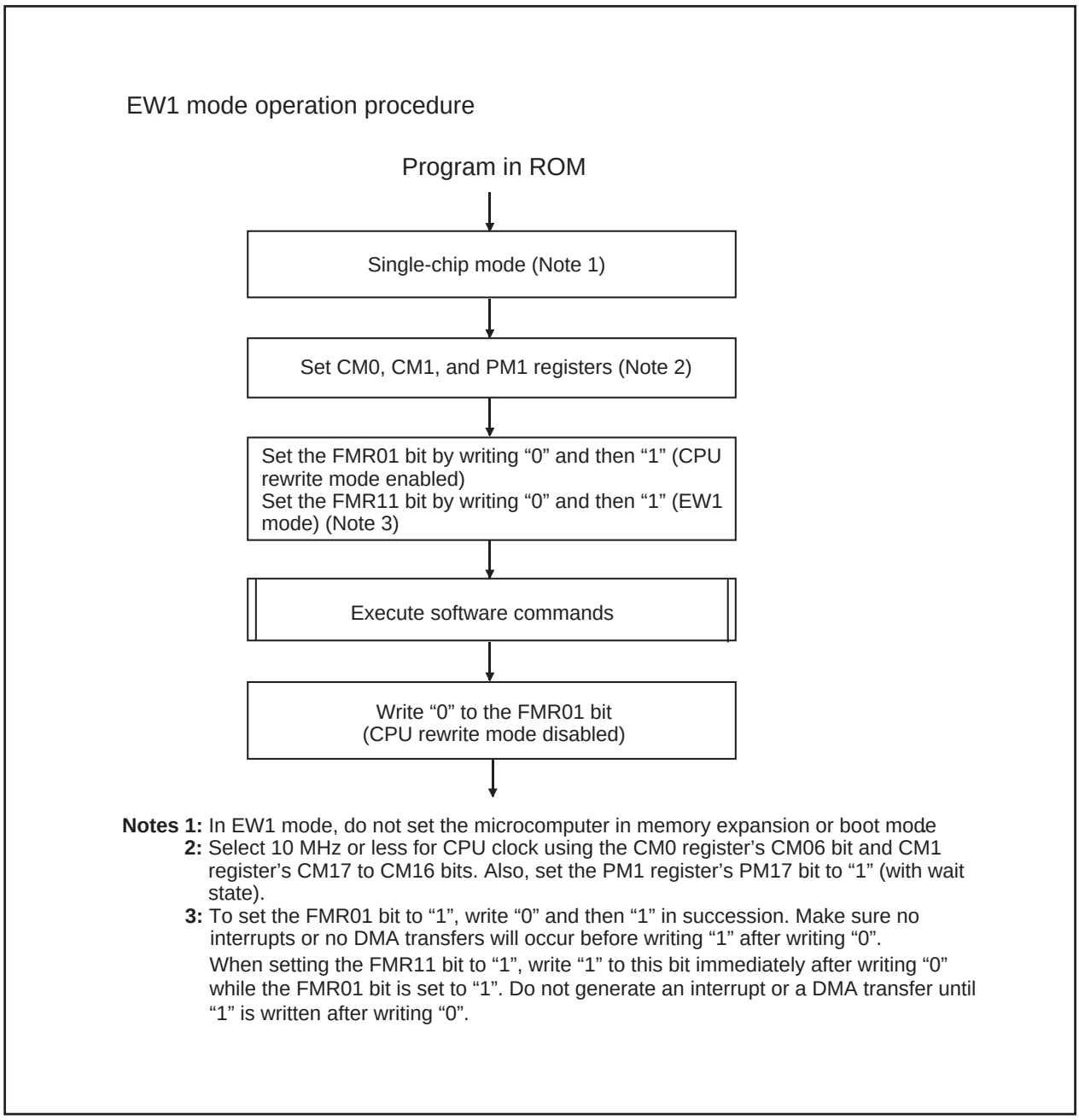
Note 1: To set this bit to “1”, write “0” and then “1” in succession. Make sure no interrupts or DMA transfers will occur before writing “1” after writing “0”.
Note 2: This bit becomes effective only when erase suspend permission bit (FMOSi40) is “1”, and it is that only the period from erase command issue to an erase end can be written in. (It is set to “1” except the above-mentioned period.)
In the EW0 mode, “0” and “1” writing of this bit are attained by the program.
In the EW1 mode, if maskable interruption occurs during elimination when bit of FMOSi40 are “1”, it will be automatically set to “1.” “1” cannot be written in by the program (“0” writing is possible.)

Figure 19.5-2. Registers FMOSA1, FMOSB1, FMOSA4 and FMOSB4



Note 1: Select 10 MHz or less for CPU clock using the CM0 register’s CM06 bit and CM1 register’s CM17 to 6 bits. Also, set the PM1 register’s PM17 bit to “1” (with wait state).
Note 2: To set the FMR01 bit to “1”, write “0” and then “1” in succession. Make sure no interrupts or no DMA transfers will occur before writing “1” after writing “0”.
Write to the FMR01 bit from a program in other than the flash memory.
Note 3: Disables the CPU rewrite mode after executing the Read Array command.
Note 4: User ROM area is accessed when the FMR05 bit is set to “1”.
Note 5: 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 (50000_{16} to $BFFFF_{16}$) cannot be used.

Figure 19.6. Setting and Resetting of EW0 Mode



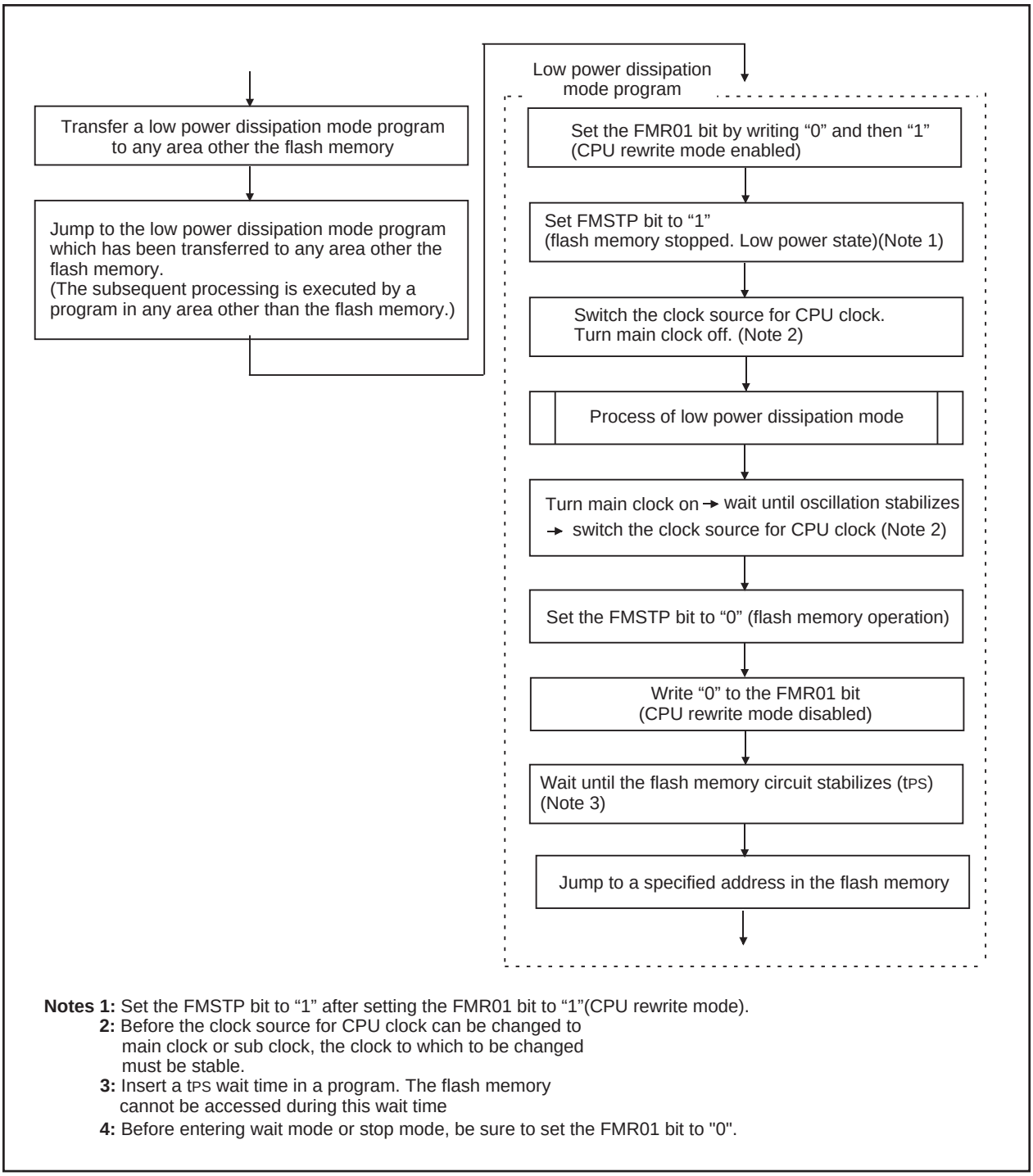
Notes 1: In EW1 mode, do not set the microcomputer in memory expansion or boot mode

2: Select 10 MHz or less for CPU clock using the CM0 register's CM06 bit and CM1 register's CM17 to CM16 bits. Also, set the PM1 register's PM17 bit to “1” (with wait state).

3: To set the FMR01 bit to “1”, write “0” and then “1” in succession. Make sure no interrupts or no DMA transfers will occur before writing “1” after writing “0”.

When setting the FMR11 bit to “1”, write “1” to this bit immediately after writing “0” while the FMR01 bit is set to “1”. Do not generate an interrupt or a DMA transfer until “1” is written after writing “0”.

Figure 19.7. Setting and Resetting of EW1 Mode



Notes 1: Set the FMSTP bit to “1” after setting the FMR01 bit to “1”(CPU rewrite mode).
2: Before the clock source for CPU clock can be changed to main clock or sub clock, the clock to which to be changed must be stable.
3: Insert a tPS wait time in a program. The flash memory cannot be accessed during this wait time
4: Before entering wait mode or stop mode, be sure to set the FMR01 bit to "0".

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 (D_{15}–D_8) are ignored.

Table 19.4. Software Commands

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_{15} to D_0)	Mode	Address	Data (D_{15} to D_0)
Read array	Write	X	xxFF16			
Read status register	Write	X	xx7016	Read	X	SRD
Clear status register	Write	X	xx5016			
Program	Write	WA	xx4016	Write	WA	WD
Block erase	Write	X	xx2016	Write	BA	xxD016
Erase all unlocked block (Note 1, 2)	Write	X	xxA716	Write	X	xxD016
Lock bit program (Note 2)	Write	BA	xx7716	Write	BA	xxD016
Read lock bit status (Note 2)	Write	X	xx7116	Write	BA	xxD016

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 (D_7 to D_0)
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.

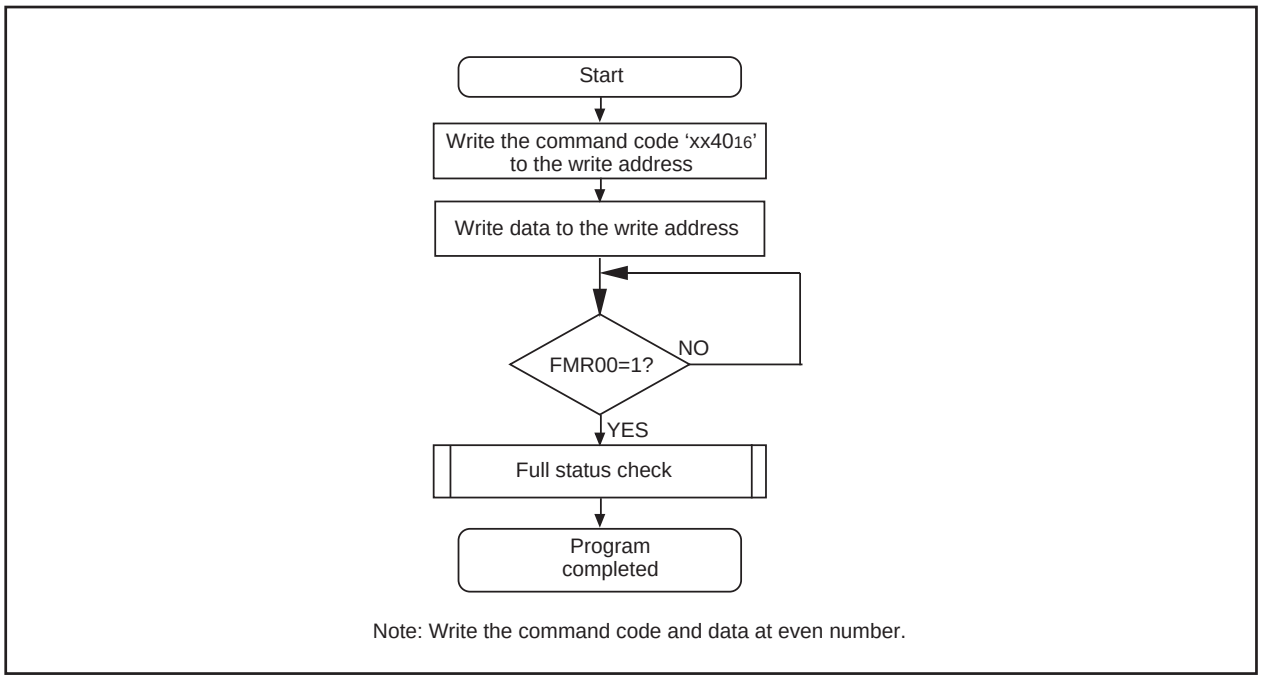
Check the FMR00 bit in the FMR0 register to see if auto programming has finished. The FMR00 bit is "0" during auto programming and set to "1" when auto programming is completed.

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.

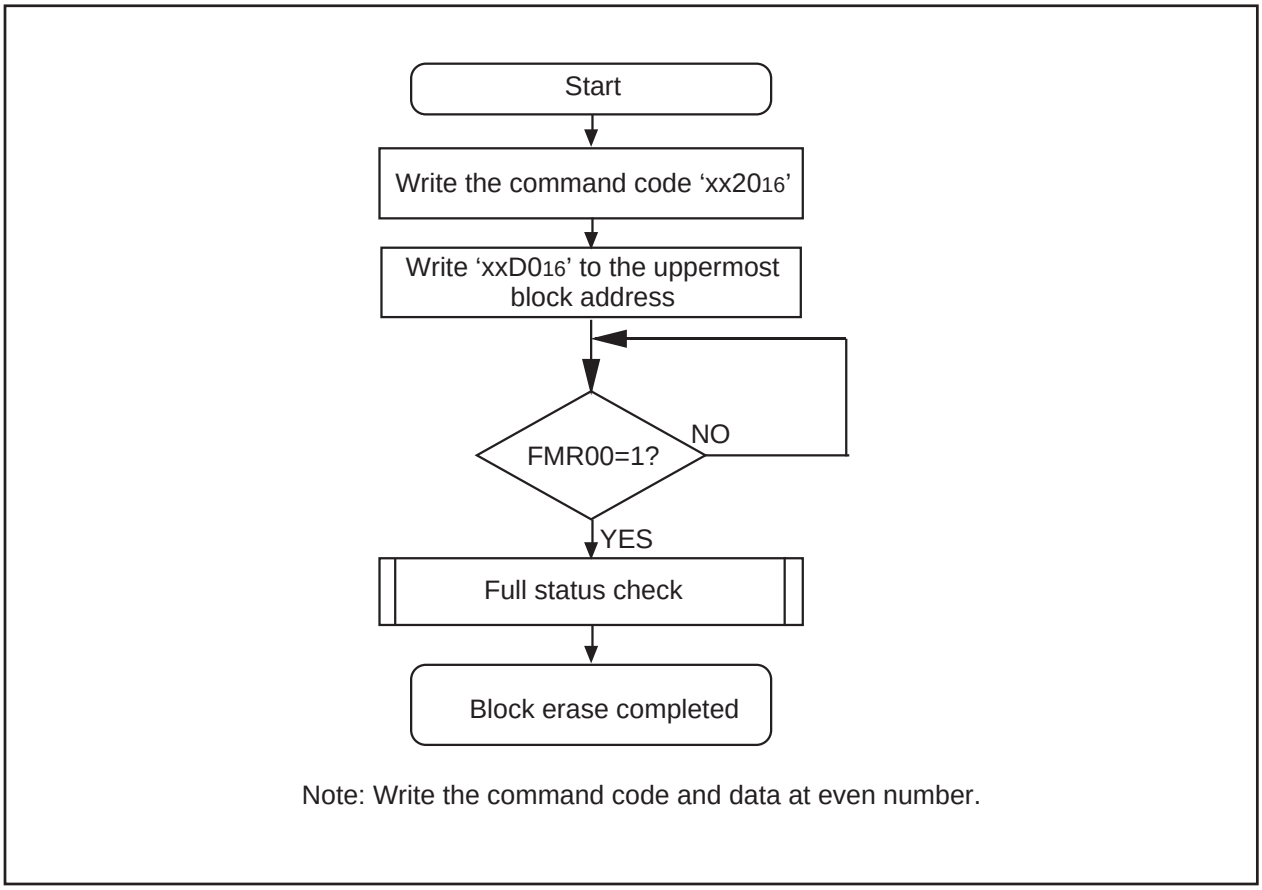


Note: Write the command code and data at even number.

Figure 19.9. Program Command

Block Erase

Write ‘xx20$_{16}$’ in the first bus cycle and write ‘xxD0$_{16}$’ 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.



Note: Write the command code and data at even number.

Figure 19.10. Block Erase Command

Erase All Unlocked Block (User area only)

Write 'xxA7$_{16}$' in the first bus cycle and write 'xxD0$_{16}$' 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 'xx77$_{16}$' in the first bus cycle and write 'xxD0$_{16}$' 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).

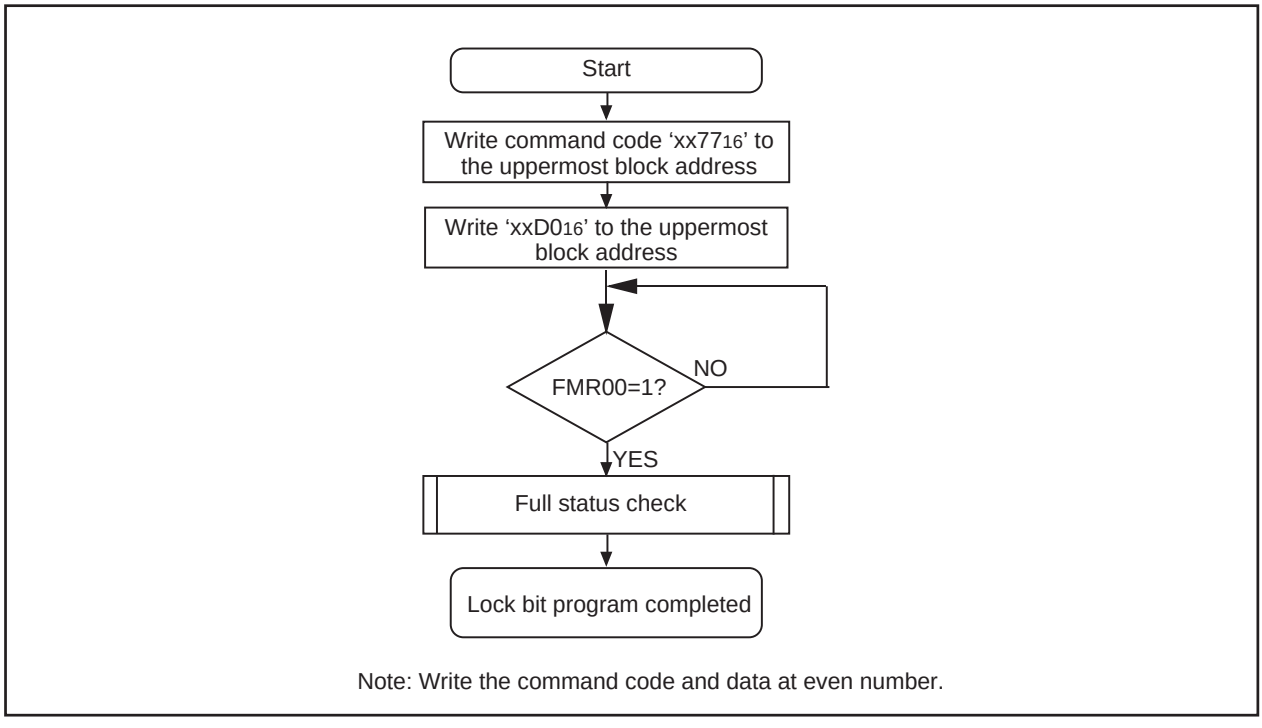


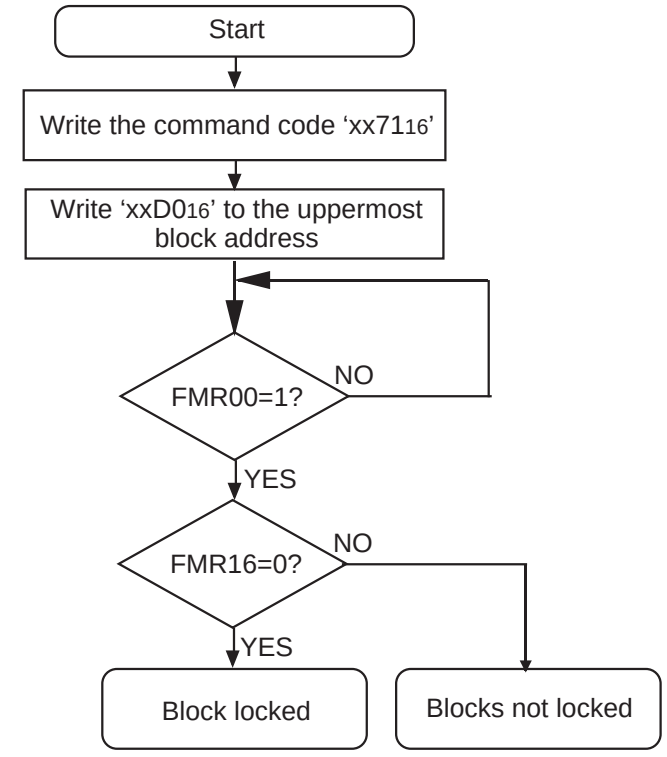
Figure 19.11. Lock Bit Program Command

Read Lock Bit Status Command (User area only)

This command reads the lock bit status of a specified block.

Write ‘xx71$_{16}$’ in the first bus cycle and write ‘xxD0$_{16}$’ 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.



Note: Write the command code and data at even number.

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 to three area 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 (D_7)	bit0	Sequencer status	Busy	Ready	1
SR6 (D_6)	———	Reserved bit	-	-	———
SR5 (D_5)	bit7	Erase status	Terminated normally	Terminated in error	0
SR4 (D_4)	bit6	Program status	Terminated normally	Terminated in error	0
SR3 (D_3)	———	Reserved bit	-	-	———
SR2 (D_2)	———	Reserved bit	-	-	———
SR1 (D_1)	———	Reserved bit	-	-	———
SR0 (D_0)	———	Reserved bit	-	-	———

• D_0 to D_7: 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	• 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 ‘xxD0$_{16}$’ or ‘xxFF$_{16}$’) (Note 1)
1	0	Erase error	• 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	• 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 ‘xxFF$_{16}$’ in the second bus cycle of these commands places the microcomputer in read array mode, and the command code written in the first bus cycle is nullified.

Note 2: When the 02 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.

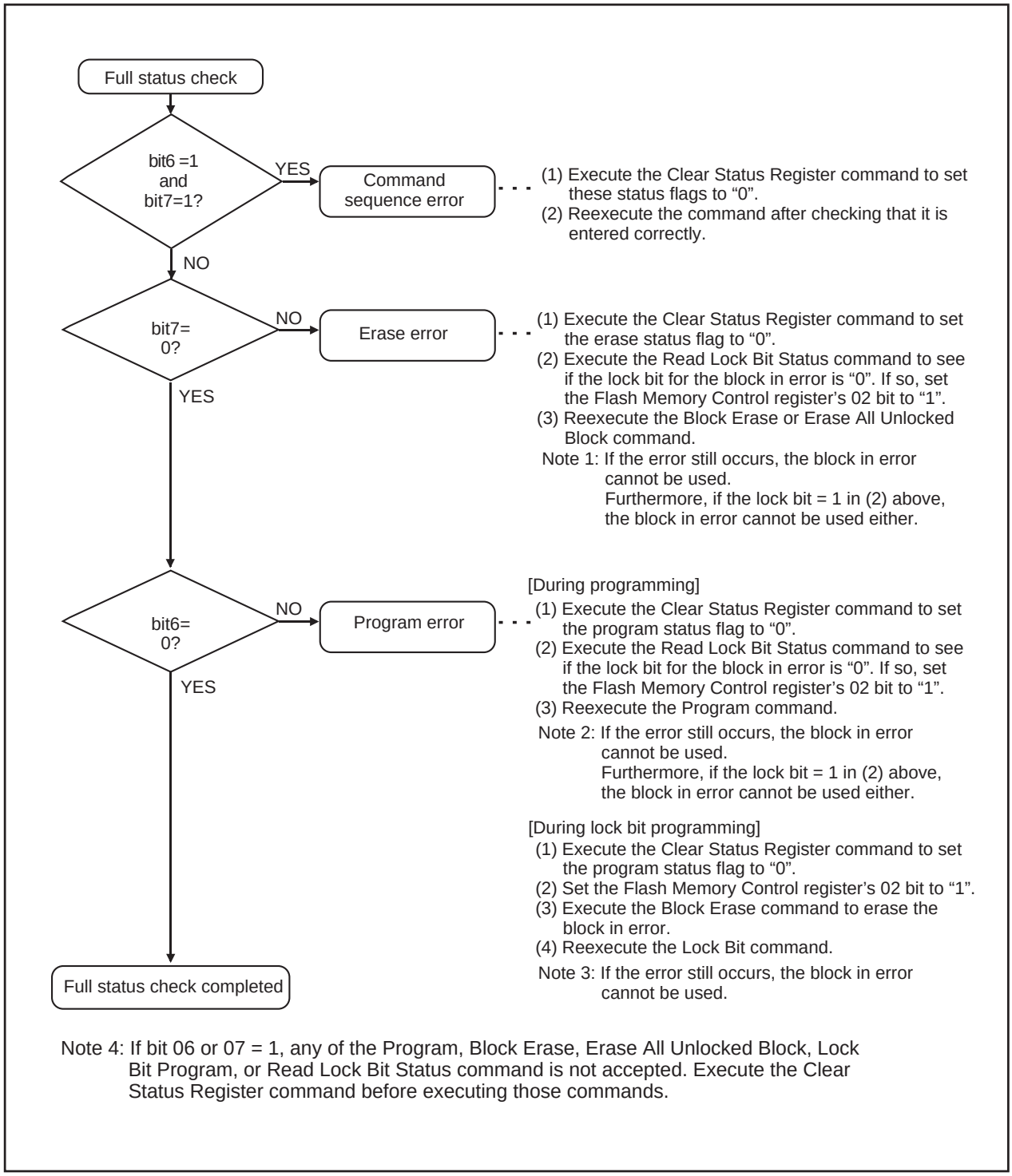


Figure 19.13. Full Status Check and Handling Procedure for Each Error

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
V_{CC1}, V_{CC2}, V_{CC3}, V_{SS}	Power input		Please input the guarantee voltage of program erase into pins V_{CC1}, V_{CC2}, and V_{CC3}. Please input 0V into V_{SS}.
CNV_{SS1}, CNV_{SS2}	CNV_{SS}	I	Please connect CNV_{SS1} to V_{CC} and connect CNV_{SS2} to V_{SS}.
$\overline{RESET}$	Reset input	I	Reset input pin. While $\overline{RESET}$ pin is "L" level, input a 20 cycle or longer clock to X_{IN} pin.
X_{IN}	Clock input	I	Connect a ceramic resonator or crystal oscillator between X_{IN} and X_{OUT} pins. To input an externally generated clock, input it to X_{IN} pin and open X_{OUT} pin.
X_{OUT}	Clock output	O	
BYTE	BYTE input	I	Connect this pin to Vss.
$P0_0$ to $P0_7$	Input port P0	I	Input "H" or "L" level signal or open.
$P1_0$ to $P1_7$	Input port P1	I	Input "H" or "L" level signal or open.
$P2_0$ to $P2_7$	Input port P2	I	Input "H" or "L" level signal or open.
$P3_0$ to $P3_7$	Input port P3	I	Input "H" or "L" level signal or open.
$P4_0$ to $P4_7$	Input port P4	I	Input "H" or "L" level signal or open.
$P5_1$ to $P5_4$, $P5_6$, $P5_7$	Input port P5	I	Input "H" or "L" level signal or open.
$P5_0$	$\overline{CE}$ input	I	Input "H" level signal.
$P5_5$	$\overline{EPM}$ input	I	Input "L" level signal.
$P6_0$ to $P6_3$	Input port P6	I	Input "H" or "L" level signal or open.
$P6_4/\overline{RTS_1}$	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitors the boot program operation check signal output pin.
$P6_5/CLK_1$	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input "L".
$P6_6/RxD_1$	RxD input	I	Serial data input pin.
$P6_7/TxD_1$	TxD output	O	Serial data output pin. (Note 1)
$P7_0$ to $P7_7$	Input port P7	I	Input "H" or "L" level signal or open.
$P8_2$ to $P8_3$, $P8_6$, $P8_7$	Input port P8	I	Input "H" or "L" level signal or open.
$P9_0$ to $P9_1$	Input port P9	I	Input "H" or "L" level signal or open.
$P10_3$ to $P10_7$	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 $\overline{RESET}$ pin is pulled low. Therefore, connect this pin to V_{CC1} 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.

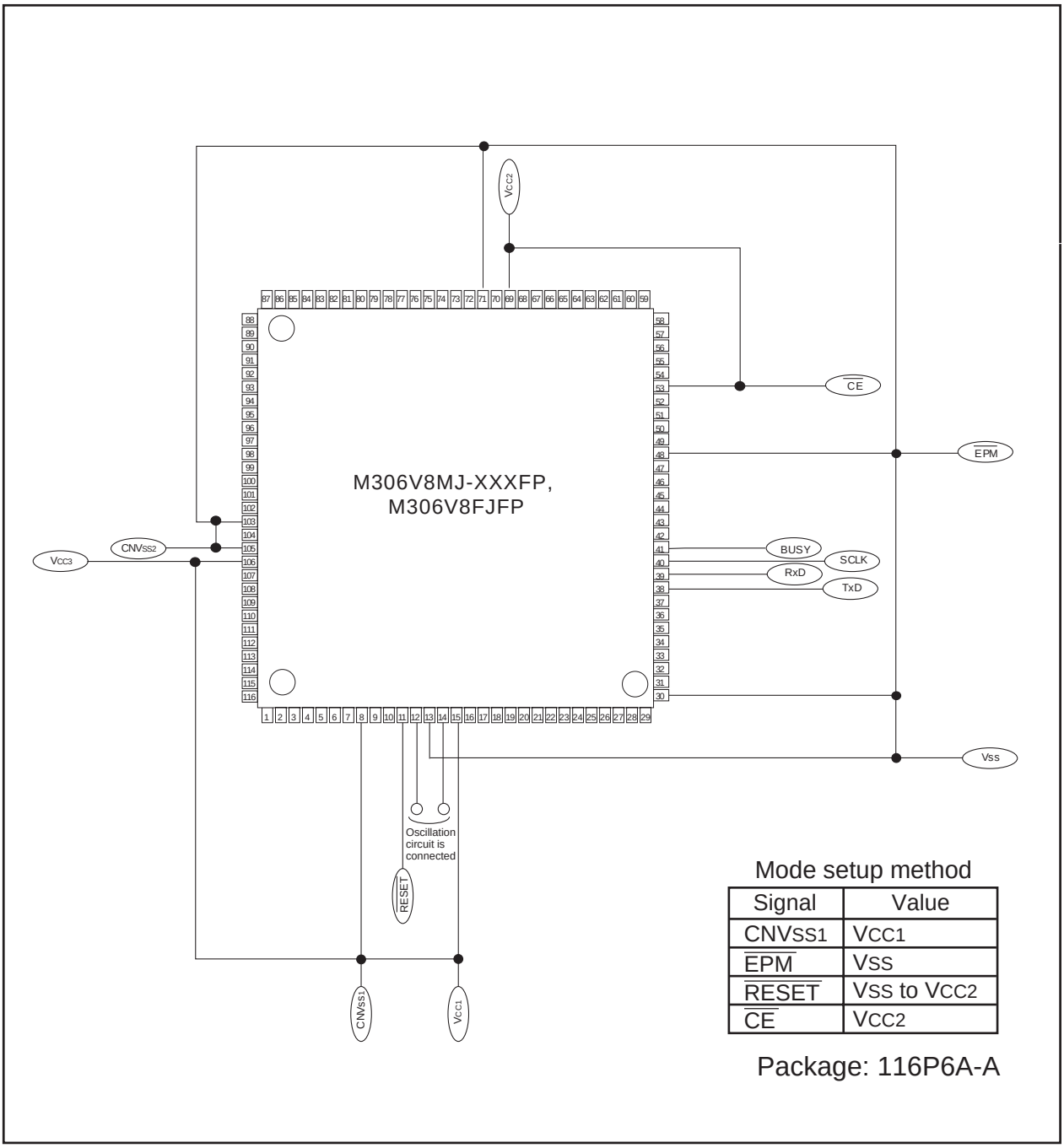
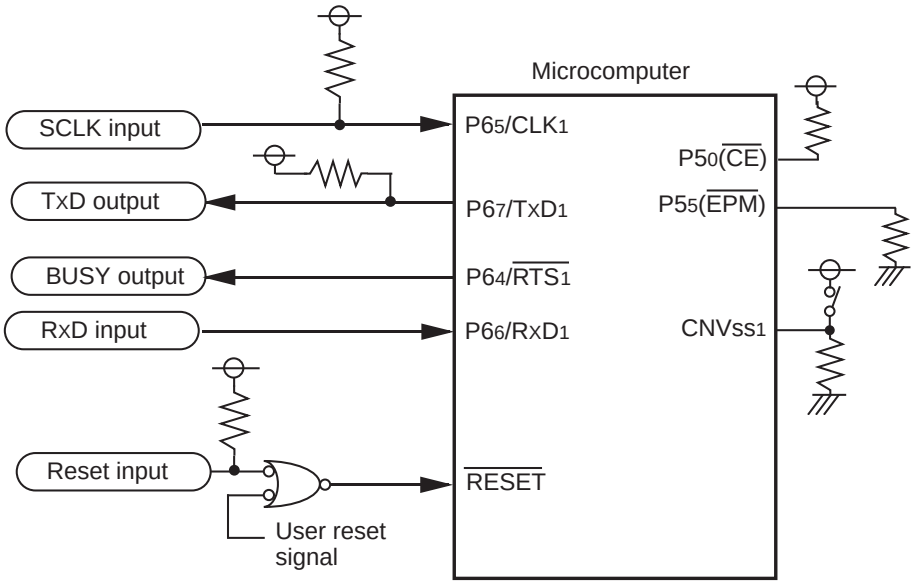


Figure 19.14. Pin circuit at the time of standard serial I/O mode

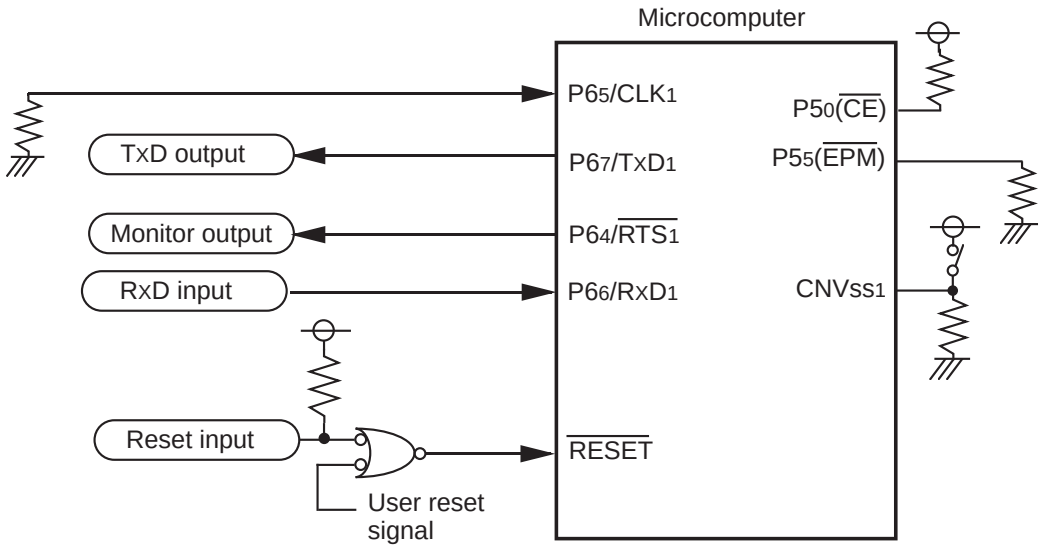
Example of Circuit Application in the Standard Serial I/O Mode

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.



(1) Control pins and external circuitry will vary according to programmer. For more information, see the programmer manual.
(2) In this example, modes are switched between single-chip mode and standard serial input/output mode by controlling the CNVss1 input with a switch.
(3) If in standard serial input/output mode 1 there is a possibility that the user reset signal will go low during serial input/output mode, break the connection between the user reset signal and RESET pin by using, for example, a jumper switch.

Figure 19.15. Circuit Application in Standard Serial I/O Mode 1



Note: In this example, modes are switched between single-chip mode and standard serial input/output mode by controlling the CNVss1 input with a switch.

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 $0FF000_{16}$ to $0FFFFF_{16}$. When rewriting the boot ROM area, make sure that only this address range is rewritten. (Do not access other than the addresses $0FF000_{16}$ to $0FFFFF_{16}$.)

ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten. (Refer to the description of the functions to inhibit rewriting flash memory version.)

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.	
SV_{CC}	Power supply rising gradient (V_{CC1})	0.05			V/ms

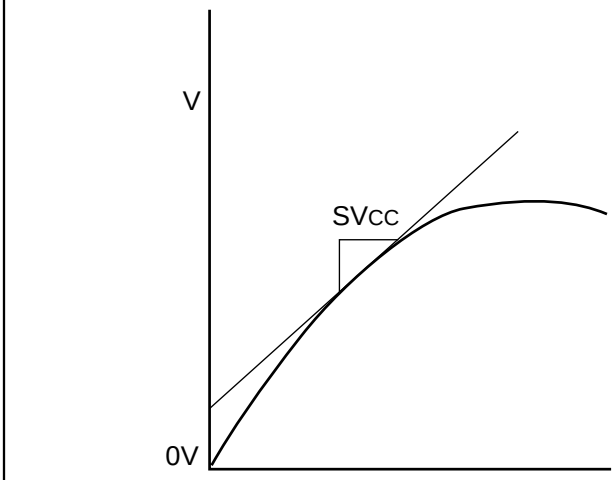
SV_{CC} Power supply rising gradient (V_{CC1})	

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 TAiMR 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 ADiCON1 register to “0” (no VREF connection). When A/D conversion is performed, start the A/D conversion at least 1 µs or longer after setting the VCUT bit to “1” (VREF connection).

Stopping peripheral functions

Use the CM0 register CM02 bit to stop the unnecessary peripheral functions during wait mode.
However, because the peripheral function clock (fC32) generated from the sub-clock does not stop, this measure is not conducive to reducing the power consumption of the chip. If low speed mode or low power dissipation mode is to be changed to wait mode, set the CM02 bit to “0” (do not peripheral function clock stopped when in wait mode), before changing wait mode.

Switching the oscillation-driving capacity

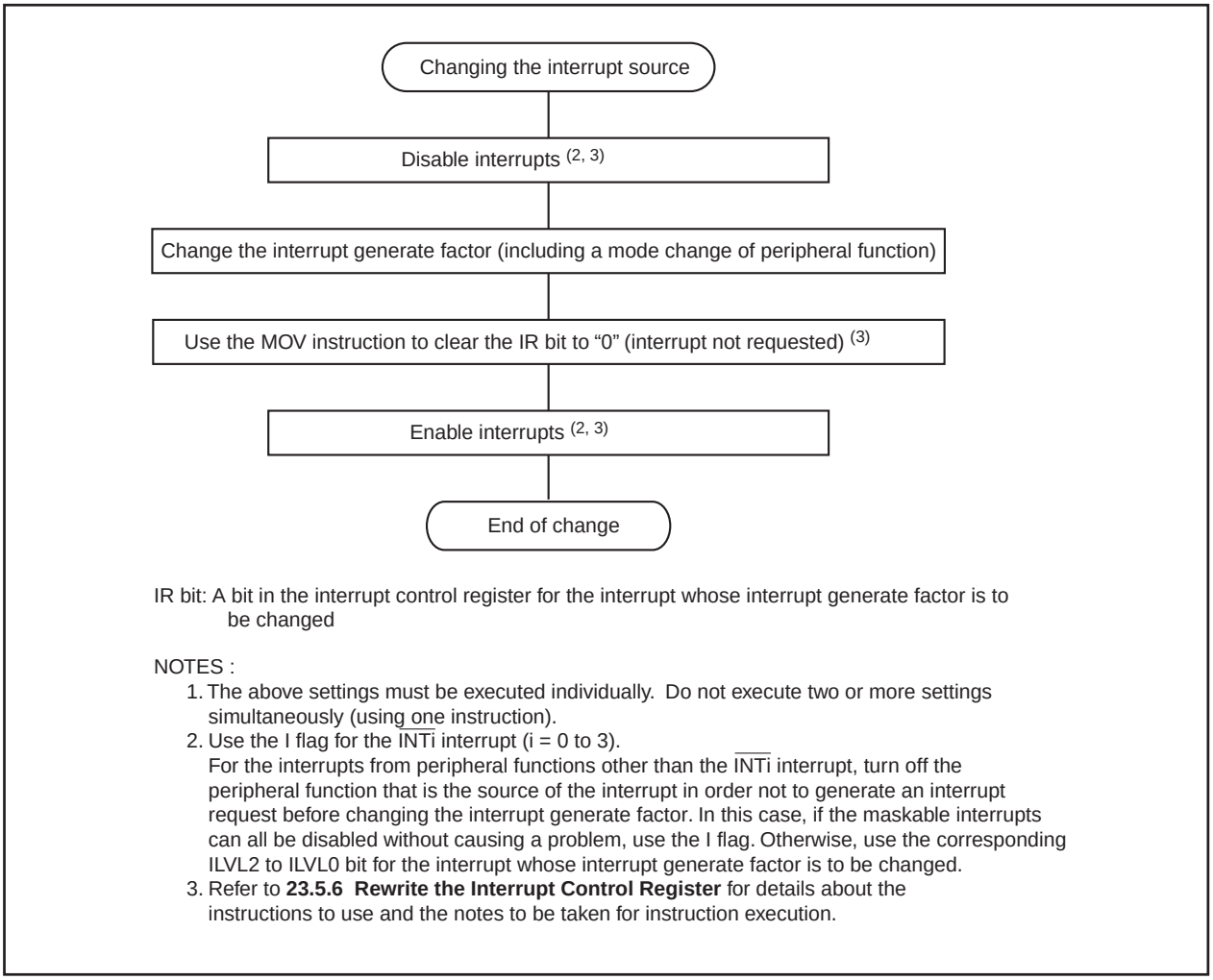
Set the driving capacity to “LOW” when oscillation is stable.

Changing the Interrupt Generate Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to “1” (interrupt requested). If you changed the interrupt generate factor for an interrupt that needs to be used, be sure to clear the IR bit for that interrupt to “0” (interrupt not requested).

Changing the interrupt generate factor refered to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the generate factor, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to “0” (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.

Figure 20.2 shows the procedure for changing the interrupt generate factor.



IR bit: A bit in the interrupt control register for the interrupt whose interrupt generate factor is to be changed

NOTES :
1. The above settings must be executed individually. Do not execute two or more settings simultaneously (using one instruction).
2. Use the I flag for the $\overline{\text{INT}}$i interrupt (i = 0 to 3).
   For the interrupts from peripheral functions other than the $\overline{\text{INT}}$i interrupt, turn off the peripheral function that is the source of the interrupt in order not to generate an interrupt request before changing the interrupt generate factor. In this case, if the maskable interrupts can all be disabled without causing a problem, use the I flag. Otherwise, use the corresponding ILVL2 to ILVL0 bit for the interrupt whose interrupt generate factor is to be changed.
3. Refer to **23.5.6 Rewrite the Interrupt Control Register** for details about the instructions to use and the notes to be taken for instruction execution.

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 TAiMR (i = 0 to 4) register and the TAi register before setting the TAiS bit in the TABSR register to “1” (count starts).

Always make sure the TAiMR register is modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the counter is read at the same time it is reloaded, the value “FFFFh” is read. Also, if the counter is read before it starts counting after a value is set in the TAi register while not counting, the set value is read.

(2) Timer A (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the UDF register, the ONSF register TAZIE, TA0TGL and TA0TGH bits and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).

Always make sure the TAiMR register, the UDF register, the TAZIE, TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, “FFFFh” can be read in underflow, while reloading, and “0000h” in overflow. When setting TAi register to a value during a counter stop, the setting value can be read before a counter starts counting. Also, if the counter is read before it starts counting after a value is set in the TAi register while not counting, the set value is read.

(3) Timer A (One-shot Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the TA0TGL and TA0TGH bits in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to “1” (count starts).
Always make sure the TAiMR register, the TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

When setting TAiS bit to “0” (count stop), the followings occur:

- A counter stops counting and a content of reload register is reloaded.
- TAiOUT pin outputs “L”.
- After one cycle of the CPU clock, the IR bit in the TAiIC register is set to “1” (interrupt request).

Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger has been selected, one-cycle delay of a count source as maximum occurs between a trigger input to TAiIN pin and output in one-shot timer mode.

The IR bit is set to “1” when timer operation mode is set with any of the following procedures:

- Select one-shot timer mode after reset.
- Change an operation mode from timer mode to one-shot timer mode.
- Change an operation mode from event counter mode to one-shot timer mode.

To use the Timer Ai interrupt (the IR bit), set the IR bit to “0” after the changes listed above have been made.

When a trigger occurs, while counting, a counter reloads the reload register to continue counting after generating a re-trigger and counting down once. To generate a trigger while counting, generate a second trigger between occurring the previous trigger and operating longer than one cycle of a timer count source.

(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 TAiS bit in the TABSR register to “1” (count starts).

Always make sure the TAiMR register, TA0TGL and TA0TGH bits and the TRGSR register are modified while the TAiS bit remains “0” (count stops) regardless whether after reset or not.

The IR bit is set to “1” when setting a timer operation mode with any of the following procedures:

- Select the PWM mode after reset.
- Change an operation mode from timer mode to PWM mode.
- Change an operation mode from event counter mode to PWM mode.

To use the Timer Ai interrupt (interrupt request bit), set the IR bit to “0” by program after the above listed changes have been made.

When setting TAiS register to “0” (count stop) during PWM pulse output, the following action occurs:

- Stop counting.
- When TAiOUT pin is output “H,” output level is set to “L” and the IR bit is set to “1”.
- When TAiOUT pin is output “L,” both output level and the IR bit remains unchanged.

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 $\overline{\text{RTS}}$ function, the output level of the $\overline{\text{RTSi}}$ pin goes to “L” when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{\text{RTSi}}$ pin goes to “H” when reception starts. So if the $\overline{\text{RTSi}}$ pin is connected to the $\overline{\text{CTSi}}$ pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the $\overline{\text{RTS}}$ function has no effect.

(2) Transmission

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

- The TE bit in the UiC1 register= 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in UiTB register)
- If $\overline{\text{CTS}}$ function is selected, input on the $\overline{\text{CTSi}}$ pin = L

(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 µs 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 VIH and VIL (neither “high” nor “low”), the input level may be determined differently depending on which side—the programmable input/output port or the peripheral function—is currently selected.

Flash Memory Version

(1) Functions to Inhibit Rewriting Flash Memory Rewrite

ID codes are stored in addresses 0FFFDFh, 0FFFE3h, 0FFFEBh, 0FFFEFh, 0FFFF3h, 0FFFF7h, and 0FFFFBh. If wrong data are written to theses addresses, the flash memory cannot be read or written in standard serial I/O mode.

The ROMCP register is mapped in address 0FFFFFh. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.

In the flash memory version of 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 (0.1µF) between the V_{CC1} pin and V_{SS} pin, V_{CC2} pin and V_{SS} pin, and V_{CC3} pin and V_{SS} pin.

Figure 20.3 shows the bypass capacitor connection.

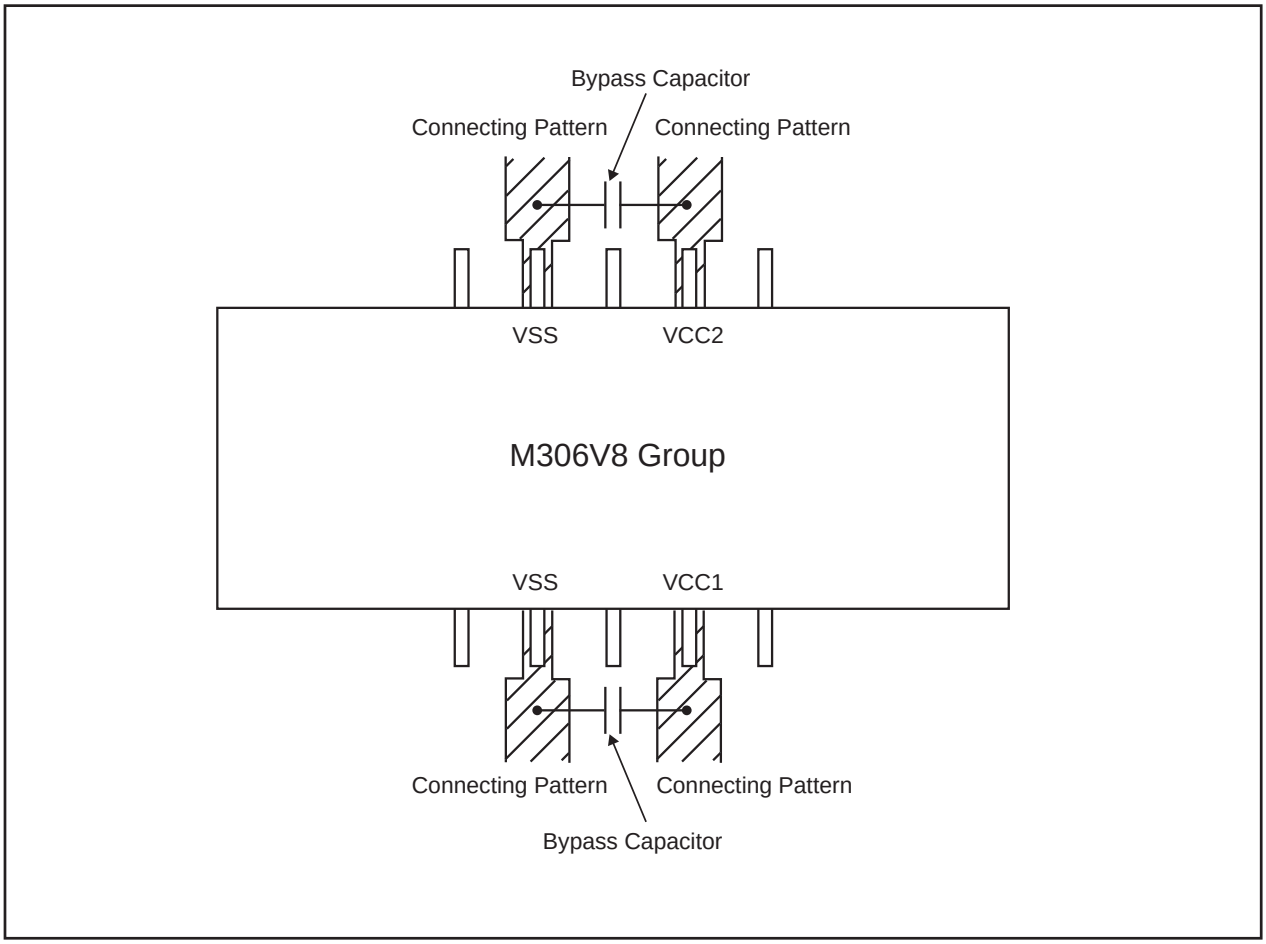


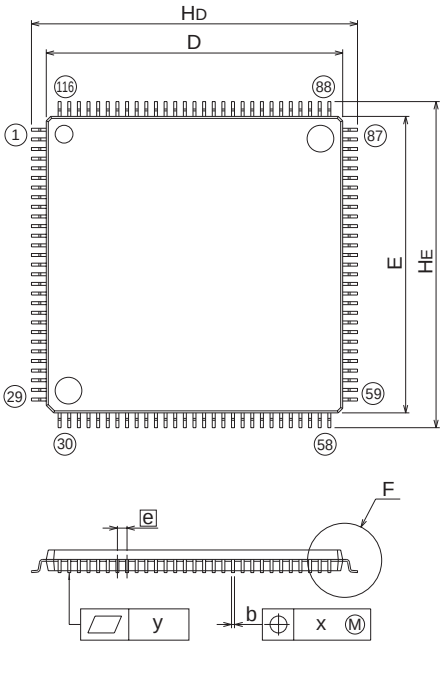
Figure 20.3 Bypass Capacitor Connection

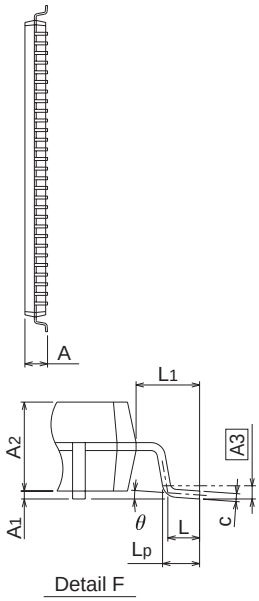
Package Outline

116P6A-A (MMP)

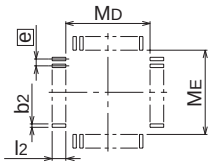
Plastic 116pin 20×20mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP116-P-2020-0.65	–		Cu Alloy





Detail F



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
θ	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

© 2005. Renesas Technology Corp., All rights reserved. Printed in Japan.

Colophon 2.0