



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

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

1. DESCRIPTION .....	1	2.16.18 R, G, B Signal Output Control .....	210
1.1 Features .....	1	2.16.19 OSD Reserved Register .....	210
1.2 Applications .....	1	2.17 Programmable I/O Ports .....	212
1.3 Pin Configuration .....	3		
1.4 Block Diagram .....	4	3. USAGE PRECAUTION .....	225
1.5 Performance Outline .....	5	3.1 Timer A (timer mode) .....	225
		3.2 Timer A (event counter mode) .....	225
2. OPERATION OF FUNCTIONAL BLOCKS ....	9	3.3 Timer A (one-shot timer mode) .....	225
2.1 Memory .....	9	3.4 Timer A (pulse width modulation mode) ....	225
2.2 Central Processing Unit (CPU) .....	15	3.5 Timer B (timer mode, event counter mode) .....	226
2.3 Reset .....	18	3.6 Timer B (pulse period, pulse width	
2.4 Processor Mode .....	23	measurement mode) .....	226
2.5 Clock Generating Circuit .....	36	3.7 A-D Converter .....	226
2.6 Protection .....	46	3.8 Stop Mode and Wait Mode .....	226
2.7 Interrupts .....	47	3.9 Interrupts .....	227
2.8 Watchdog Timer .....	67	3.10 Built-in PROM version .....	228
2.9 DMAC .....	69		
2.10 Timer .....	79	4. ITEMS TO BE SUBMITTED WHEN	
2.11 Serial I/O .....	99	ORDERING MASKED ROM VERSION .....	229
2.12 A-D Converter .....	133		
2.13 D-A Converter .....	148	5. ELECTRICAL CHARACTERISTICS .....	230
2.14 Data Slicer .....	150	5.1 Absolute Maximum Ratings .....	230
2.15 Hsync Counter .....	160	5.2 Recommended Operating Conditions ..	231
2.16 OSD Functions .....	161	5.3 Electrical Characteristics .....	232
2.16.1 Triple Layer OSD .....	166	5.4 A-D Conversion Characteristics .....	233
2.16.2 Display Position .....	168	5.5 D-A Conversion Characteristics .....	233
2.16.3 Dot size .....	172	5.6 Analog R, G, B Output Characteristics ....	233
2.16.4 Clock for OSD .....	173	5.7 Timing Requirements .....	234
2.16.5 Field Determination Display ....	174	5.8 Switching Characteristics .....	236
2.16.6 Memory for OSD .....	176	5.9 Measurement Circuit .....	239
2.16.7 Character Color .....	189	5.10 Timing Diagram .....	240
2.16.8 Character Background color .....	189		
2.16.9 OUT1, OUT2 Signals .....	193	6. MASK CONFIRMATION FORM .....	245
2.16.10 Attribute .....	194		
2.16.11 Automatic Solid Space Function ..	199	7. MARK SPECIFICATION FORM .....	249
2.16.12 Multiline Display .....	200		
2.16.13 SPRITE OSD Function .....	201	8. ONE TIME PROM VERSION	
2.16.14 Window Function .....	204	M306V0EEFP MARKING .....	250
2.16.15 Blank Function .....	205		
2.16.16 Raster Coloring Function .....	208	9. PACKAGE OUTLINE .....	251
2.16.17 Scan Mode .....	210		

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**1.3 Pin Configuration**

Figure 1.3.1 shows the pin configuration (top view).

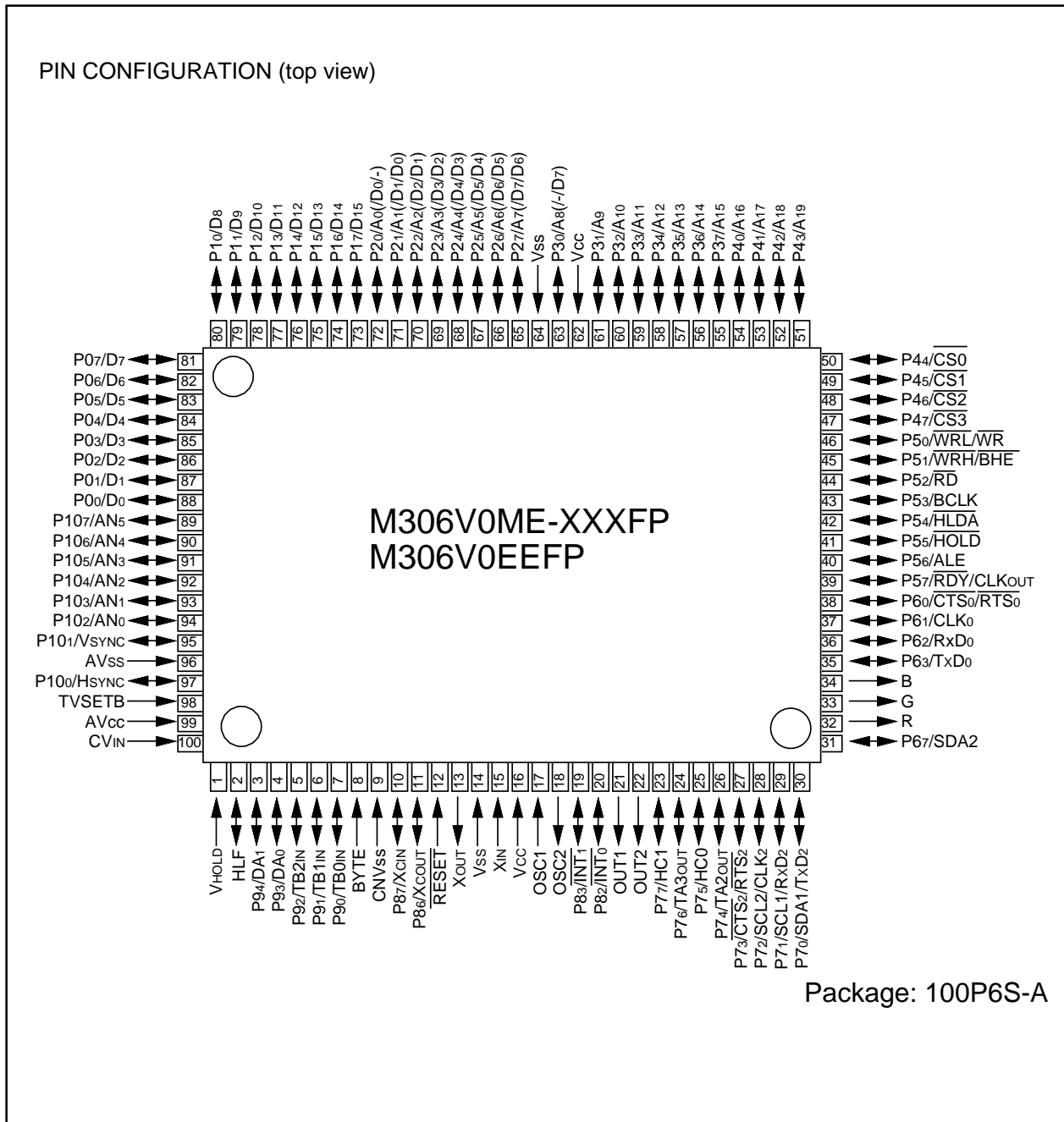


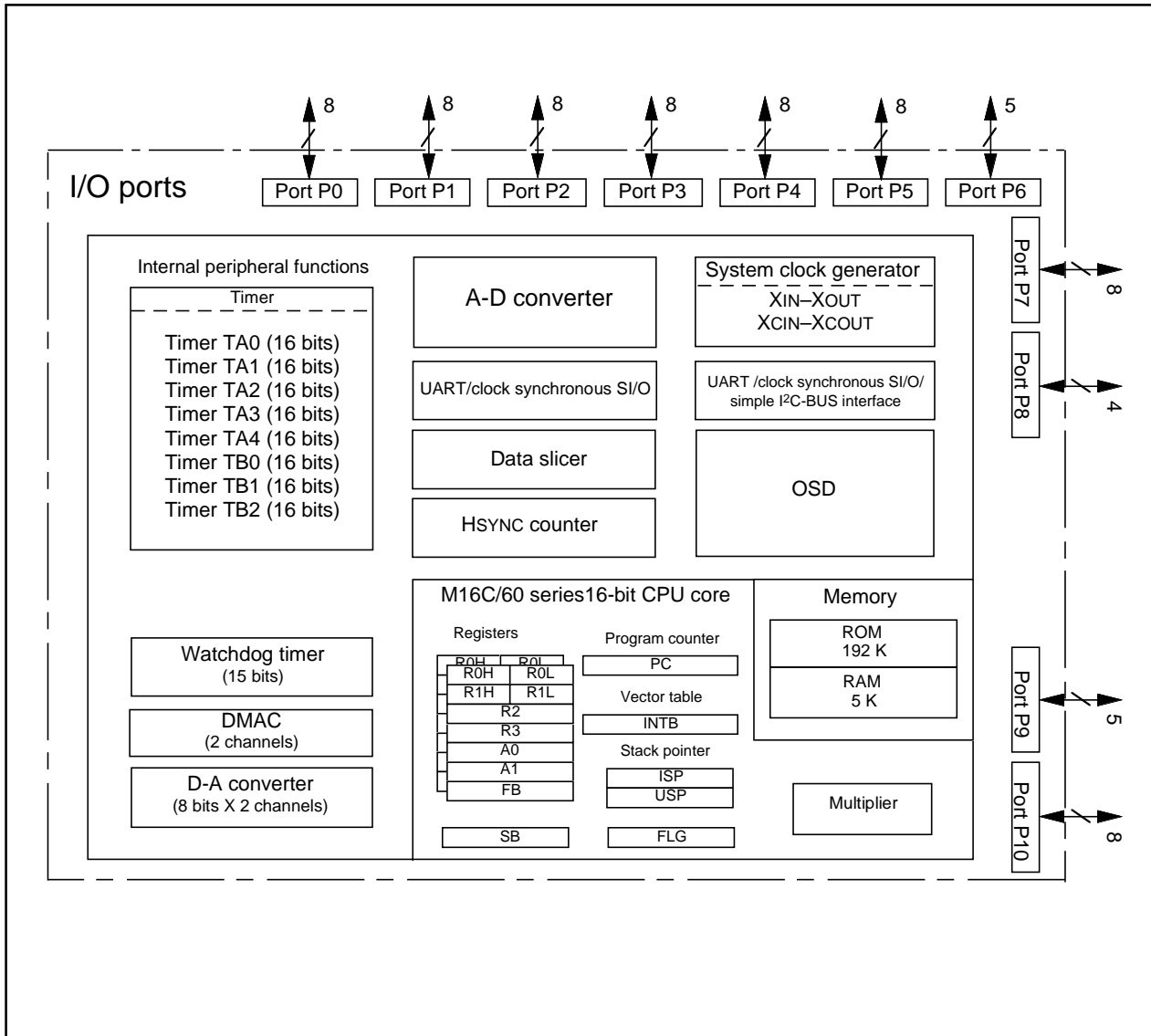
Figure 1.3.1 Pin configuration (top view)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**1.4 Block Diagram**

Figure 1.4.1 is a block diagram.



**Figure 1.4.1 Block diagram**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**1.5 Performance Outline**

Table 1.5.1 is a performance outline.

**Table 1.5.1 Performance outline**

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		100 ns(f(XIN)=10 MHz)
Memory size	ROM	192K bytes
	RAM	5K bytes
	OSD ROM	44K bytes
	OSD RAM	1.7K bytes
I/O port	P0 to P10	8 bits X 8, 5 bits X 2, 4 bits X 1
Multifunction timer	TA0, TA1, TA2, TA3, TA4	16 bits X 5
	TB0, TB1, TB2	16 bits X 3
Serial I/O	UART0	1 unit: UART or clock synchronous
	UART2	1 unit: UART, clock synchronous or simple I <sup>2</sup> C-BUS interface
A-D converter		8 bits X 6
D-A converter		8 bits X 2
DMAC		2 channels (trigger: 21 sources)
OSD function		Triple layer, 825 kinds of fonts, 42 character X 16 lines
Data slicer		32-bit buffer
HSYNC counter		8bits X 2 channel
Watchdog timer		15 bits X 1 (with prescaler)
Interrupt		20 internal and 3 external sources, 4 software sources, 7 levels
Clock generating circuit		3 built-in clock generation circuits
Power source voltage		4.5 V to 5.5V (f(XIN) = 10 MHz)
Power consumption		250 mW
I/O characteristics	I/O withstand voltage	5 V
	Output current	5 mA
Memory expansion		Available
Operating ambient temperature		-10 °C to 70 °C
Device configuration		CMOS high performance silicon gate
Package		100-pin plastic molded QFP

**M306V0ME-XXXFP**  
**M306V0EEFP**

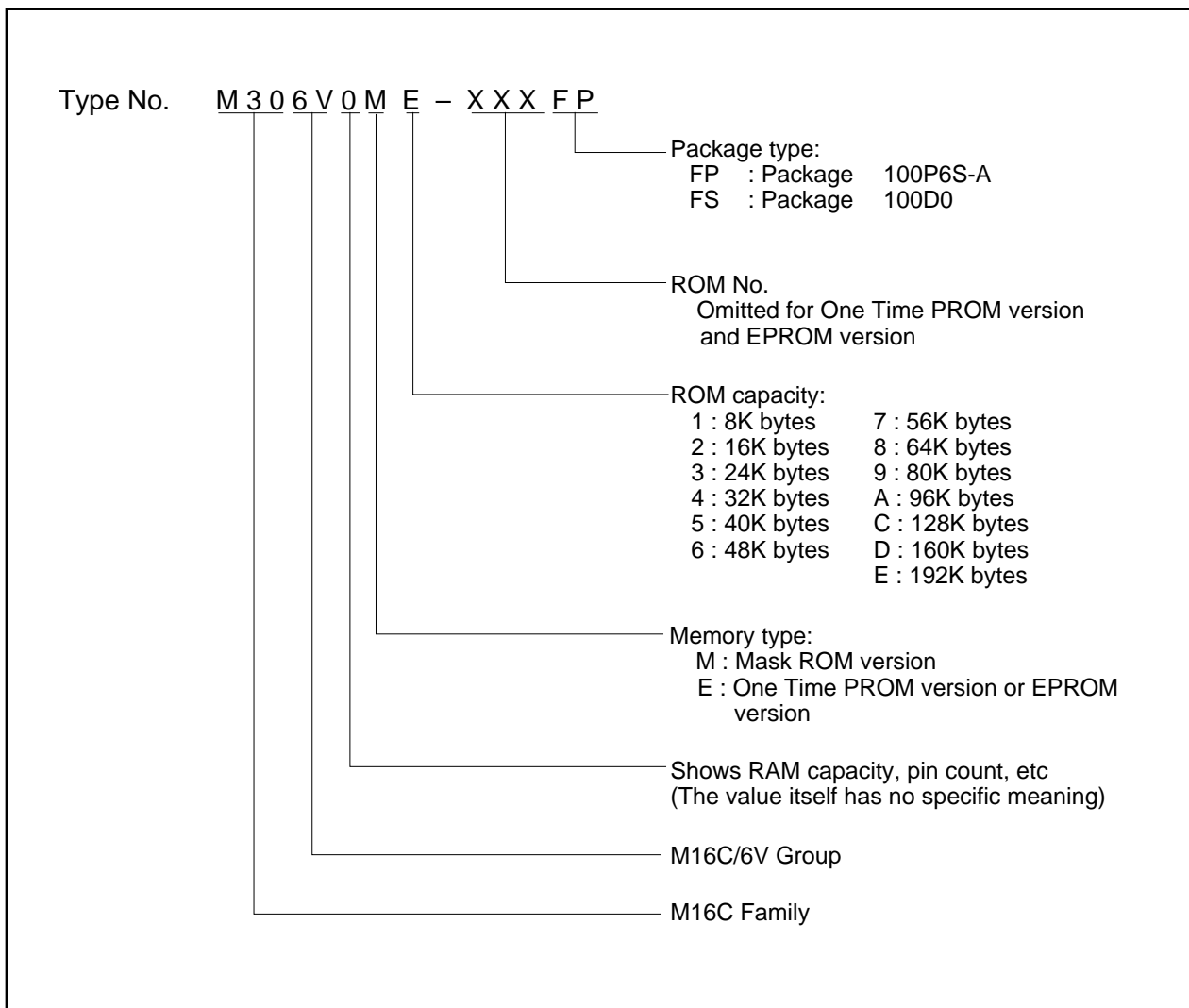
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

Currently supported products are listed below.

**Table 1.5.2 List of supported products**

Type No	ROM capacity	RAM capacity	Package type	Remarks
M306V0ME-XXXFP	192K bytes	5K bytes	100P6S-A	Mask ROM version
M306V0EEFP	192K bytes	5K bytes	100P6S-A	One Time PROM version
M306V0EEFS	192K bytes	5K bytes	100D0	EPROM version

**Note:** Since EPROM version is for development support tool (for evaluation), do not use for mass production.



**Figure 1.5.1 Type No., memory size, and package**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 1.5.3 Pin description (1)**

Pin name	Signal name	I/O type	Function
Vcc, Vss	Power supply input		Supply 4.5 V to 5.5 V to the Vcc pin. Supply 0 V to the Vss pin.
CNVss	CNVss	Input	This pin switches between processor modes. Connect it to the Vss pin when operating in single-chip or memory expansion mode. Connect it to the Vcc pin when in microprocessor mode.
$\overline{\text{RESET}}$	Reset input	Input	A "L" on this input resets the microcomputer.
XIN XOUT	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the XIN and the XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
BYTE	External data bus width select input	Input	This pin selects the width of an external data bus. A 16-bit width is selected when this input is "L"; an 8-bit width is selected when this input is "H". This input must be fixed to either "H" or "L". When operating in single-chip mode, connect this pin to Vss.
AVcc	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to Vcc.
AVss	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to Vss.
P0 <sub>0</sub> to P0 <sub>7</sub>	I/O port P0	Input/output	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When set for input in single-chip mode, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistor. In memory expansion and microprocessor modes, the user cannot specify that.
D <sub>0</sub> to D <sub>7</sub>		Input/output	When set as a separate bus, these pins input and output data (D <sub>0</sub> –D <sub>7</sub> ).
P1 <sub>0</sub> to P1 <sub>7</sub>	I/O port P1	Input/output	This is an 8-bit I/O port equivalent to P0.
D <sub>8</sub> to D <sub>15</sub>		Input/output	When set as a separate bus, these pins input and output data (D <sub>8</sub> –D <sub>15</sub> ).
P2 <sub>0</sub> to P2 <sub>7</sub>	I/O port P2	Input/output	This is an 8-bit I/O port equivalent to P0.
A <sub>0</sub> to A <sub>7</sub>		Output	These pins output 8 low-order address bits (A <sub>0</sub> –A <sub>7</sub> ).
A <sub>0</sub> /D <sub>0</sub> to A <sub>7</sub> /D <sub>7</sub>		Input/output	If the external bus is set as an 8-bit wide multiplexed bus, these pins input and output data (D <sub>0</sub> –D <sub>7</sub> ) and output 8 low-order address bits (A <sub>0</sub> –A <sub>7</sub> ) separated in time by multiplexing.
A <sub>0</sub> A <sub>1</sub> /D <sub>0</sub> to A <sub>7</sub> /D <sub>6</sub>		Output Input/output	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D <sub>0</sub> –D <sub>6</sub> ) and output address (A <sub>1</sub> –A <sub>7</sub> ) separated in time by multiplexing. They also output address (A <sub>0</sub> ).
P3 <sub>0</sub> to P3 <sub>7</sub>	I/O port P3	Input/output	This is an 8-bit I/O port equivalent to P0.
A <sub>8</sub> to A <sub>15</sub>		Output	These pins output 8 middle-order address bits (A <sub>8</sub> –A <sub>15</sub> ).
A <sub>8</sub> /D <sub>7</sub> , A <sub>9</sub> to A <sub>15</sub>		Input/output Output	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D <sub>7</sub> ) and output address (A <sub>8</sub> ) separated in time by multiplexing. They also output address (A <sub>9</sub> –A <sub>15</sub> ).
P4 <sub>0</sub> to P4 <sub>7</sub>	I/O port P4	Input/output	This is an 8-bit I/O port equivalent to P0.
$\overline{\text{CS}}_0$ to $\overline{\text{CS}}_3$ , A <sub>16</sub> to A <sub>19</sub>		Output Output	These pins output $\overline{\text{CS}}_0$ – $\overline{\text{CS}}_3$ signals and A <sub>16</sub> –A <sub>19</sub> . $\overline{\text{CS}}_0$ – $\overline{\text{CS}}_3$ are chip select signals used to specify an access space. A <sub>16</sub> –A <sub>19</sub> are 4 high-order address bits.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 1.5.4 Pin description (continued) (2)**

Pin name	Signal name	I/O type	Function
P50 to P57	I/O port P5	Input/output	This is an 8-bit I/O port equivalent to P0. In single-chip mode, P57 in this port outputs a divide-by-8 or divide-by-32 clock of XIN or a clock of the same frequency as XCIN as selected by software.
$\overline{WRL}$ / $\overline{WR}$ , $\overline{WRH}$ / $\overline{BHE}$ , $\overline{RD}$ , $\overline{BCLK}$ , $\overline{HLDA}$ , $\overline{HOLD}$ ,  $\overline{ALE}$ , $\overline{RDY}$		Output Output Output Output Input  Output Input	Output $\overline{WRL}$ , $\overline{WRH}$ ( $\overline{WR}$ and $\overline{BHE}$ ), $\overline{RD}$ , $\overline{BCLK}$ , $\overline{HLDA}$ , and $\overline{ALE}$ signals. $\overline{WRL}$ and $\overline{WRH}$ , and $\overline{BHE}$ and $\overline{WR}$ can be switched using software control. <ul style="list-style-type: none"> <li>■ <math>\overline{WRL}</math>, <math>\overline{WRH}</math>, and <math>\overline{RD}</math> selected With a 16-bit external data bus, data is written to even addresses when the <math>\overline{WRL}</math> signal is "L" and to the odd addresses when the <math>\overline{WRH}</math> signal is "L". Data is read when <math>\overline{RD}</math> is "L".</li> <li>■ <math>\overline{WR}</math>, <math>\overline{BHE}</math>, and <math>\overline{RD}</math> selected Data is written when <math>\overline{WR}</math> is "L". Data is read when <math>\overline{RD}</math> is "L". Odd addresses are accessed when <math>\overline{BHE}</math> is "L". Use this mode when using an 8-bit external data bus.</li> </ul> While the input level at the $\overline{HOLD}$ pin is "L", the microcomputer is placed in the hold state. While in the hold state, $\overline{HLDA}$ outputs a "L" level. $\overline{ALE}$ is used to latch the address. While the input level of the $\overline{RDY}$ pin is "L", the microcomputer is in the ready state.
P60 to P63, P67	I/O port P6	Input/output	This is an 5-bit I/O port equivalent to P0. When set for input in single-chip, microprocessor and memory expansion modes, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistor. Pins in this port also function as UART0 and UART2 pins as selected by software.
P70 to P77	I/O port P7	Input/output	This is an 8-bit I/O port equivalent to P6 (P70 and P71 are N-channel open-drain output). Pins in this port also function as timers A2 and A3, UART2, or HSYNC counter I/O pins as selected by software.
P82, P83, P86, P87	I/O port P8	Input/output	P82, P83, P86 and P87 are I/O ports with the same functions as P6. Using software, P82 and P83 can be made to function as the I/O pins for the input pins for external interrupts. P86 and P87 can be set using software to function as the I/O pins for a sub clock generation circuit. In this case, connect a quartz oscillator between P86 (XCOUT pin) and P87 (XCIN pin).
P90 to P94	I/O port P9	Input/output	This is an 5-bit I/O port equivalent to P6. Pins in this port also function as Timer B0 to B2 input pins, or D-A converter output pins.
P100 to P107	I/O port P10	Input/output	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as A-D converter input pins. Furthermore, P100 and P101 also function as input pins for OSD function.
R, G, B	OSD output	Output	These are OSD output pins (analog output).
OUT1, OUT2	OSD output	Output	These are OSD output pins (digital output).
OSC1	Clock input for OSD	Input	This is an OSD clock input pin.
OSC2	Clock output for OSD	Output	This is an OSD clock output pin.
CVIN	I/O for data slicer	Input	Input composite video signal through a capacitor.
VHOLD		Input	Connect a capacitor between VHOLD and Vss.
HLF		Input/output	Connect a filter using of a capacitor and a resistor between HLF and Vss.
TVSETB	Test input	Input	This is a test input pin. Fix it to "L."



## 2. OPERATION OF FUNCTIONAL BLOCKS

This microcomputer accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also included are peripheral units such as timers, serial I/O, D-A converter, DMAC, OSD circuit, data slicer, A-D converter, and I/O ports.

The following explains each unit.

### 2.1 Memory

Figure 2.1.1 is a memory map. The address space extends the 1M bytes from address 00000<sub>16</sub> to FFFFF<sub>16</sub>. From FFFFF<sub>16</sub> down is ROM. There is 192K bytes of internal ROM from D0000<sub>16</sub> to FFFFF<sub>16</sub>. The vector table for fixed interrupts such as the reset mapped to FFFDC<sub>16</sub> to FFFFF<sub>16</sub>. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

5K bytes of internal RAM is mapped to the space from 02C00<sub>16</sub> to 03FFF<sub>16</sub>. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SFR area is mapped to 00000<sub>16</sub> to 003FF<sub>16</sub>. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Figures 2.1.2 to 2.1.5 are location of peripheral unit control registers. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.

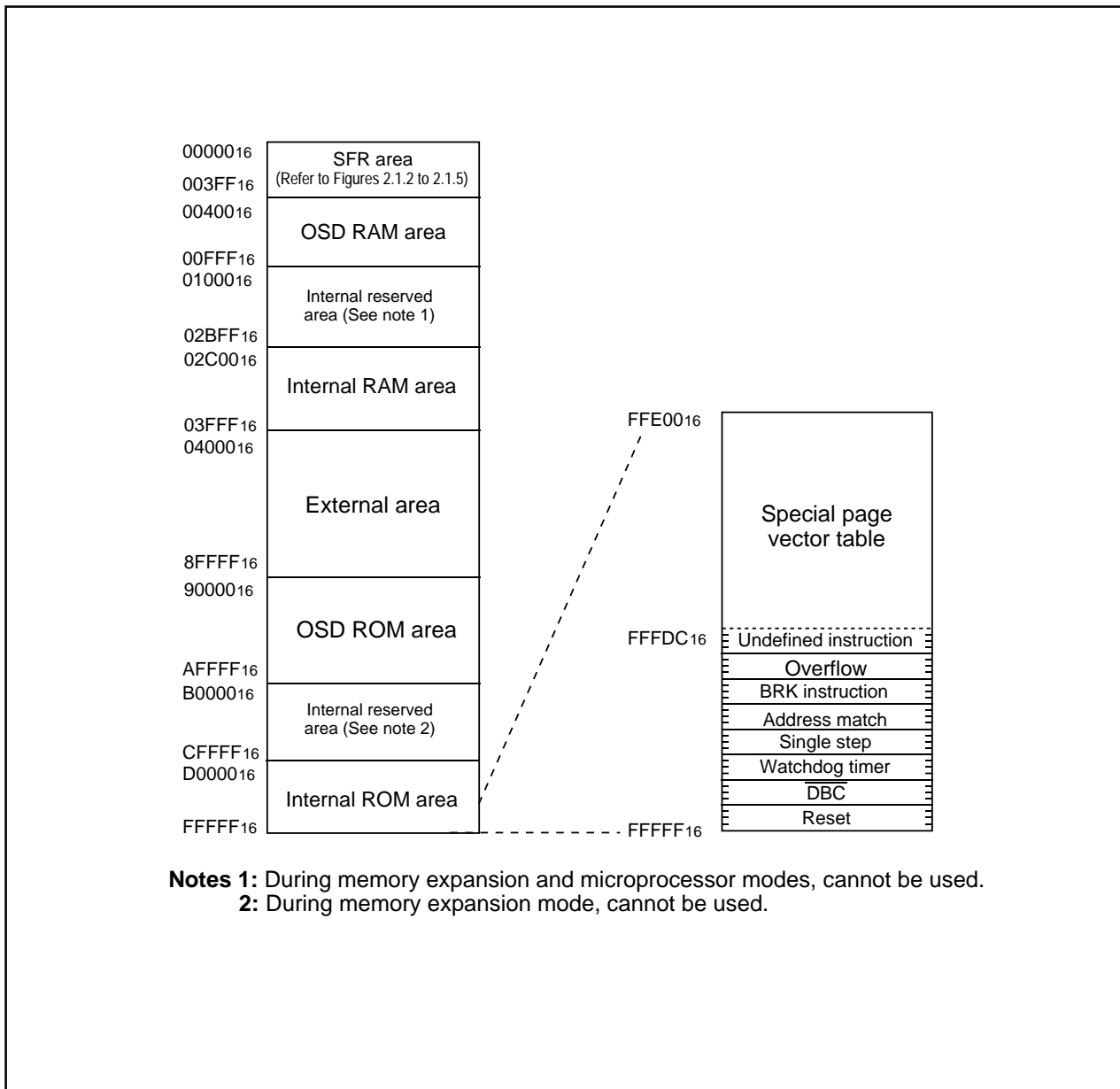
The special page vector table is mapped to FFE00<sub>16</sub> to FFFDB<sub>16</sub>. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.

In memory expansion mode and microprocessor mode, a part of the spaces are reserved and cannot be used. The following spaces cannot be used.

- The space between 01000<sub>16</sub> and 02BFF<sub>16</sub> (in memory expansion and microprocessor modes)
- The space between B0000<sub>16</sub> and CFFFF<sub>16</sub> (in memory expansion mode)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Notes 1:** During memory expansion and microprocessor modes, cannot be used.  
**2:** During memory expansion mode, cannot be used.

Figure 2.1.1 Memory map

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

0000 <sub>16</sub>		0040 <sub>16</sub>	
0001 <sub>16</sub>		0041 <sub>16</sub>	
0002 <sub>16</sub>		0042 <sub>16</sub>	
0003 <sub>16</sub>		0043 <sub>16</sub>	
0004 <sub>16</sub>	Processor mode register 0 (PM0)	0044 <sub>16</sub>	OSD1 interrupt control register (OSD1IC)
0005 <sub>16</sub>	Processor mode register 1 (PM1)	0045 <sub>16</sub>	Interrupt control reserved register 0 (RE0IC)
0006 <sub>16</sub>	System clock control register 0 (CM0)	0046 <sub>16</sub>	Interrupt control reserved register 1 (RE1IC)
0007 <sub>16</sub>	System clock control register 1 (CM1)	0047 <sub>16</sub>	Interrupt control reserved register 2 (RE2IC)
0008 <sub>16</sub>	Chip select control register (CSR)	0048 <sub>16</sub>	OSD2 interrupt control register (OSD2IC)
0009 <sub>16</sub>	Address match interrupt enable register (AIER)	0049 <sub>16</sub>	Interrupt control reserved register 3 (RE3IC)
000A <sub>16</sub>	Protect register (PRCR)	004A <sub>16</sub>	Bus collision detection interrupt control register (BCNIC)
000B <sub>16</sub>		004B <sub>16</sub>	DMA0 interrupt control register (DM0IC)
000C <sub>16</sub>		004C <sub>16</sub>	DMA1 interrupt control register (DM1IC)
000D <sub>16</sub>		004D <sub>16</sub>	Interrupt control reserved register 5 (RE5IC)
000E <sub>16</sub>	Watchdog timer start register (WDTS)	004E <sub>16</sub>	A-D conversion interrupt control register (ADIC)
000F <sub>16</sub>	Watchdog timer control register (WDC)	004F <sub>16</sub>	UART2 transmit interrupt control register (S2TIC)
0010 <sub>16</sub>		0050 <sub>16</sub>	UART2 receive interrupt control register (S2RIC)
0011 <sub>16</sub>	Address match interrupt register 0 (RMAD0)	0051 <sub>16</sub>	UART0 transmit interrupt control register (S0TIC)
0012 <sub>16</sub>		0052 <sub>16</sub>	UART0 receive interrupt control register (S0RIC)
0013 <sub>16</sub>		0053 <sub>16</sub>	Data slicer interrupt control register (DSIC)
0014 <sub>16</sub>		0054 <sub>16</sub>	VSYNC interrupt control register (VSYNCIC)
0015 <sub>16</sub>	Address match interrupt register 1 (RMAD1)	0055 <sub>16</sub>	Timer A0 interrupt control register (TA0IC)
0016 <sub>16</sub>		0056 <sub>16</sub>	Timer A1 interrupt control register (TA1IC)
0017 <sub>16</sub>		0057 <sub>16</sub>	Timer A2 interrupt control register (TA2IC)
0018 <sub>16</sub>		0058 <sub>16</sub>	Timer A3 interrupt control register (TA3IC)
0019 <sub>16</sub>		0059 <sub>16</sub>	Timer A4 interrupt control register (TA4IC)
001A <sub>16</sub>		005A <sub>16</sub>	Timer B0 interrupt control register (TB0IC)
001B <sub>16</sub>		005B <sub>16</sub>	Timer B1 interrupt control register (TB1IC)
001C <sub>16</sub>		005C <sub>16</sub>	Timer B2 interrupt control register (TB2IC)
001D <sub>16</sub>		005D <sub>16</sub>	INT0 interrupt control register (INT0IC)
001E <sub>16</sub>		005E <sub>16</sub>	INT1 interrupt control register (INT1IC)
001F <sub>16</sub>		005F <sub>16</sub>	Interrupt control reserved register 4 (RE4IC)
0020 <sub>16</sub>		0060 <sub>16</sub>	
0021 <sub>16</sub>	DMA0 source pointer (SAR0)		
0022 <sub>16</sub>			
0023 <sub>16</sub>			
0024 <sub>16</sub>			
0025 <sub>16</sub>	DMA0 destination pointer (DAR0)		
0026 <sub>16</sub>			
0027 <sub>16</sub>			
0028 <sub>16</sub>	DMA0 transfer counter (TCR0)		
0029 <sub>16</sub>			
002A <sub>16</sub>			
002B <sub>16</sub>			
002C <sub>16</sub>	DMA0 control register (DM0CON)		
002D <sub>16</sub>			
002E <sub>16</sub>			
002F <sub>16</sub>			
0030 <sub>16</sub>			
0031 <sub>16</sub>	DMA1 source pointer (SAR1)		
0032 <sub>16</sub>			
0033 <sub>16</sub>			
0034 <sub>16</sub>			
0035 <sub>16</sub>	DMA1 destination pointer (DAR1)		
0036 <sub>16</sub>			
0037 <sub>16</sub>			
0038 <sub>16</sub>	DMA1 transfer counter (TCR1)		
0039 <sub>16</sub>			
003A <sub>16</sub>			
003B <sub>16</sub>			
003C <sub>16</sub>	DMA1 control register (DM1CON)		
003D <sub>16</sub>			
003E <sub>16</sub>			
003F <sub>16</sub>			

Figure 2.1.2 Location of peripheral unit control registers (1)



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

0340 <sub>16</sub>	Reserved register 1 (INVC1)	0380 <sub>16</sub>	Count start flag (TABSR)
0341 <sub>16</sub>		0381 <sub>16</sub>	Clock prescaler reset flag (CPSRF)
0342 <sub>16</sub>		0382 <sub>16</sub>	One-shot start flag (ONSF)
0343 <sub>16</sub>		0383 <sub>16</sub>	Trigger select register (TRGSR)
0344 <sub>16</sub>		0384 <sub>16</sub>	Up-down flag (UDF)
0345 <sub>16</sub>		0385 <sub>16</sub>	
0346 <sub>16</sub>		0386 <sub>16</sub>	Timer A0 register (TA0)
0347 <sub>16</sub>		0387 <sub>16</sub>	
0348 <sub>16</sub>	Reserved register 0 (INVC0)	0388 <sub>16</sub>	Timer A1 register (TA1)
0349 <sub>16</sub>		0389 <sub>16</sub>	
034A <sub>16</sub>		038A <sub>16</sub>	Timer A2 register (TA2)
034B <sub>16</sub>		038B <sub>16</sub>	
034C <sub>16</sub>		038C <sub>16</sub>	Timer A3 register (TA3)
034D <sub>16</sub>		038D <sub>16</sub>	
034E <sub>16</sub>		038E <sub>16</sub>	Timer A4 register (TA4)
034F <sub>16</sub>		038F <sub>16</sub>	
0350 <sub>16</sub>		0390 <sub>16</sub>	Timer B0 register (TB0)
0351 <sub>16</sub>		0391 <sub>16</sub>	
0352 <sub>16</sub>		0392 <sub>16</sub>	Timer B1 register (TB1)
0353 <sub>16</sub>		0393 <sub>16</sub>	
0354 <sub>16</sub>		0394 <sub>16</sub>	Timer B2 register (TB2)
0355 <sub>16</sub>		0395 <sub>16</sub>	
0356 <sub>16</sub>		0396 <sub>16</sub>	Timer A0 mode register (TA0MR)
0357 <sub>16</sub>		0397 <sub>16</sub>	Timer A1 mode register (TA1MR)
0358 <sub>16</sub>		0398 <sub>16</sub>	Timer A2 mode register (TA2MR)
0359 <sub>16</sub>		0399 <sub>16</sub>	Timer A3 mode register (TA3MR)
035A <sub>16</sub>		039A <sub>16</sub>	Timer A4 mode register (TA4MR)
035B <sub>16</sub>		039B <sub>16</sub>	Timer B0 mode register (TB0MR)
035C <sub>16</sub>		039C <sub>16</sub>	Timer B1 mode register (TB1MR)
035D <sub>16</sub>		039D <sub>16</sub>	Timer B2 mode register (TB2MR)
035E <sub>16</sub>		039E <sub>16</sub>	
035F <sub>16</sub>	Interrupt request cause select register (IFSR)	039F <sub>16</sub>	
0360 <sub>16</sub>		03A0 <sub>16</sub>	UART0 transmit/receive mode register (U0MR)
0361 <sub>16</sub>		03A1 <sub>16</sub>	UART0 bit rate generator (U0BRG)
0362 <sub>16</sub>	Reserved register 3 (INVC3)	03A2 <sub>16</sub>	
0363 <sub>16</sub>		03A3 <sub>16</sub>	UART0 transmit buffer register (U0TB)
0364 <sub>16</sub>		03A4 <sub>16</sub>	UART0 transmit/receive control register 0 (U0C0)
0365 <sub>16</sub>		03A5 <sub>16</sub>	UART0 transmit/receive control register 1 (U0C1)
0366 <sub>16</sub>	Reserved register 4 (INVC4)	03A6 <sub>16</sub>	
0367 <sub>16</sub>		03A7 <sub>16</sub>	UART0 receive buffer register (U0RB)
0368 <sub>16</sub>		03A8 <sub>16</sub>	Reserved register 2 (INVC2)
0369 <sub>16</sub>		03A9 <sub>16</sub>	
036A <sub>16</sub>		03AA <sub>16</sub>	
036B <sub>16</sub>		03AB <sub>16</sub>	
036C <sub>16</sub>		03AC <sub>16</sub>	
036D <sub>16</sub>		03AD <sub>16</sub>	
036E <sub>16</sub>		03AE <sub>16</sub>	
036F <sub>16</sub>		03AF <sub>16</sub>	
0370 <sub>16</sub>		03B0 <sub>16</sub>	UART transmit/receive control register 2 (UCON)
0371 <sub>16</sub>		03B1 <sub>16</sub>	
0372 <sub>16</sub>		03B2 <sub>16</sub>	
0373 <sub>16</sub>		03B3 <sub>16</sub>	
0374 <sub>16</sub>		03B4 <sub>16</sub>	
0375 <sub>16</sub>		03B5 <sub>16</sub>	
0376 <sub>16</sub>	Reserved register 5 (INVC5)	03B6 <sub>16</sub>	
0377 <sub>16</sub>	UART2 special mode register (U2SMR)	03B7 <sub>16</sub>	
0378 <sub>16</sub>	UART2 transmit/receive mode register (U2MR)	03B8 <sub>16</sub>	DMA0 request cause select register (DM0SL)
0379 <sub>16</sub>	UART2 bit rate generator (U2BRG)	03B9 <sub>16</sub>	
037A <sub>16</sub>		03BA <sub>16</sub>	DMA1 request cause select register (DM1SL)
037B <sub>16</sub>	UART2 transmit buffer register (U2TB)	03BB <sub>16</sub>	
037C <sub>16</sub>	UART2 transmit/receive control register 0 (U2C0)	03BC <sub>16</sub>	
037D <sub>16</sub>	UART2 transmit/receive control register 1 (U2C1)	03BD <sub>16</sub>	
037E <sub>16</sub>		03BE <sub>16</sub>	
037F <sub>16</sub>	UART2 receive buffer register (U2RB)	03BF <sub>16</sub>	

Figure 2.1.4 Location of peripheral unit control registers (3)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

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

Figure 2.1.5 Location of peripheral unit control registers (4)

## 2.2 Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure 2.2.1. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.

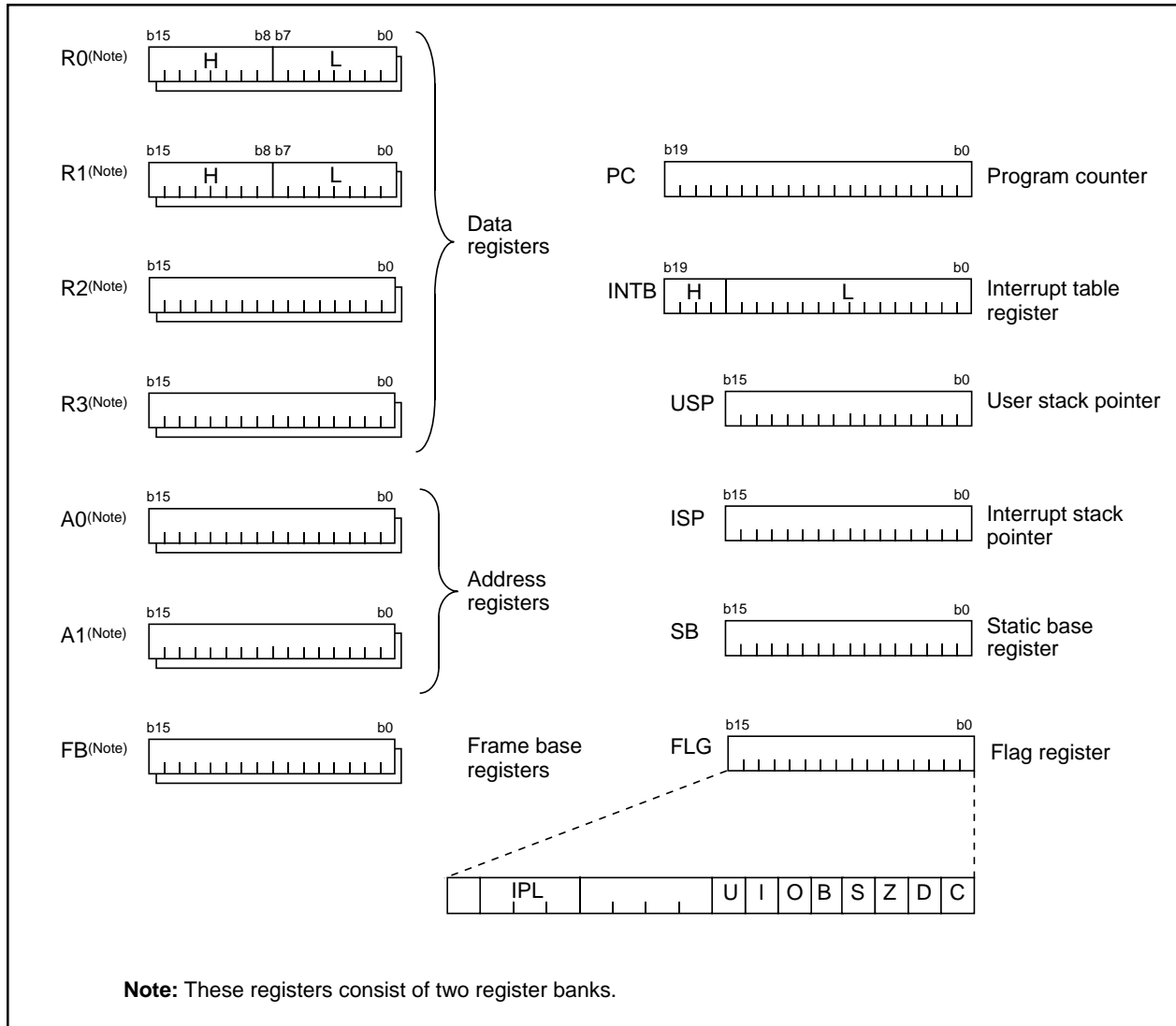


Figure 2.2.1 Central processing unit register

### 2.2.1 Data registers (R0, R0H, R0L, R1, R1H, R1L, R2, and R3)

Data registers (R0, R1, R2, and R3) are configured with 16 bits, and are used primarily for transfer and arithmetic/logic operations.

Registers R0 and R1 each can be used as separate 8-bit data registers, high-order bits as (R0H/R1H), and low-order bits as (R0L/R1L). In some instructions, registers R2 and R0, as well as R3 and R1 can use as 32-bit data registers (R2R0/R3R1).

### 2.2.2 Address registers (A0 and A1)

Address registers (A0 and A1) are configured with 16 bits, and have functions equivalent to those of data registers. These registers can also be used for address register indirect addressing and address register relative addressing.

In some instructions, registers A1 and A0 can be combined for use as a 32-bit address register (A1A0).

### 2.2.3 Frame base register (FB)

Frame base register (FB) is configured with 16 bits, and is used for FB relative addressing.

### 2.2.4 Program counter (PC)

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

### 2.2.5 Interrupt table register (INTB)

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

### 2.2.6 Stack pointer (USP/ISP)

Stack pointer comes in two types: user stack pointer (USP) and interrupt stack pointer (ISP), each configured with 16 bits.

Your desired type of stack pointer (USP or ISP) can be selected by a stack pointer select flag (U flag).

This flag is located at the position of bit 7 in the flag register (FLG).

### 2.2.7 Static base register (SB)

Static base register (SB) is configured with 16 bits, and is used for SB relative addressing.

### 2.2.8 Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure 2.2.2 shows the flag register (FLG). The following explains the function of each flag:

- **Bit 0: Carry flag (C flag)**

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

- **Bit 1: Debug flag (D flag)**

This flag enables a single-step interrupt.

When this flag is "1", a single-step interrupt is generated after instruction execution. This flag is cleared to "0" when the interrupt is acknowledged.

- **Bit 2: Zero flag (Z flag)**

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

- **Bit 3: Sign flag (S flag)**

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

- **Bit 4: Register bank select flag (B flag)**

This flag chooses a register bank. Register bank 0 is selected when this flag is "0"; register bank 1 is selected when this flag is "1".



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

• **Bit 5: Overflow flag (O flag)**

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

• **Bit 6: Interrupt enable flag (I flag)**

This flag enables a maskable interrupt.

An interrupt is disabled when this flag is “0”, and is enabled when this flag is “1”. This flag is cleared to “0” when the interrupt is acknowledged.

• **Bit 7: Stack pointer select flag (U flag)**

Interrupt stack pointer (ISP) is selected when this flag is “0” ; user stack pointer (USP) is selected when this flag is “1”.

This flag is cleared to “0” when a hardware interrupt is acknowledged or an INT instruction of software interrupt Nos. 0 to 31 is executed.

• **Bits 8 to 11: Reserved area**

• **Bits 12 to 14: Processor interrupt priority level (IPL)**

Processor interrupt priority level (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 the processor interrupt priority level (IPL), the interrupt is enabled.

• **Bit 15: Reserved area**

The C, Z, S, and O flags are changed when instructions are executed. See the software manual for details.

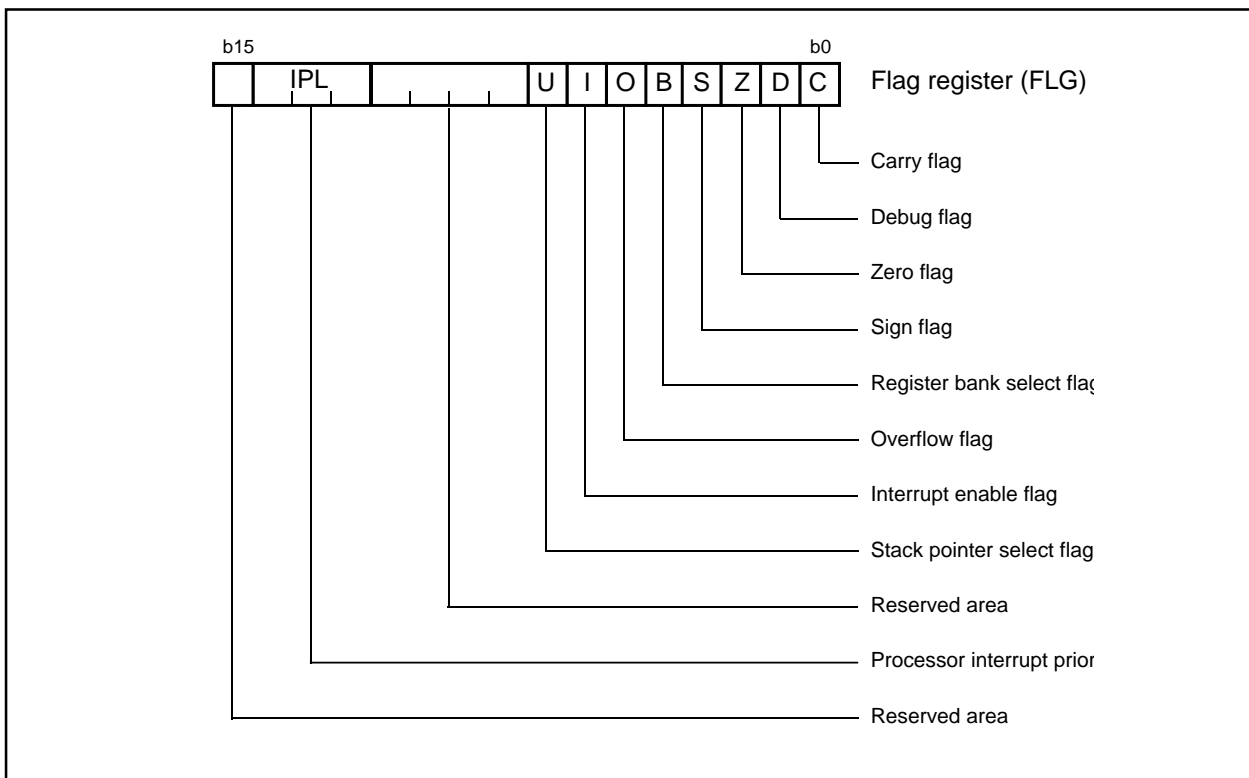


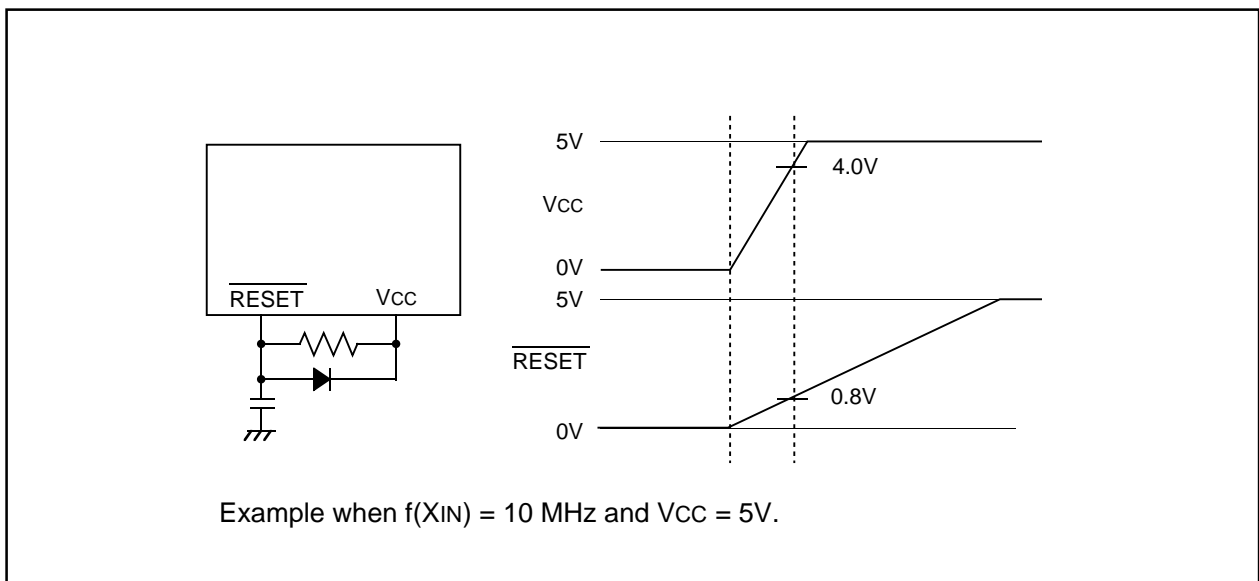
Figure 2.2.2 Flag register (FLG)

### 2.3 Reset

There are two kinds of resets; hardware and software. In both cases, operation is the same after the reset. (See "Software Reset" for details of software resets.) This section explains on hardware resets.

When the supply voltage is in the range where operation is guaranteed, a reset is effected by holding the reset pin level "L" (0.2V<sub>CC</sub> max.) for at least 20 cycles. When the reset pin level is then returned to the "H" level while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Figure 2.3.1 shows the example reset circuit. Figure 2.3.2 shows the reset sequence.



**Figure 2.3.1 Example reset circuit**

#### 2.3.1 Software Reset

Writing "1" to bit 3 of the processor mode register 0 (address 000416) applies a (software) reset to the microcomputer. A software reset has almost the same effect as a hardware reset. The contents of internal RAM are preserved.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

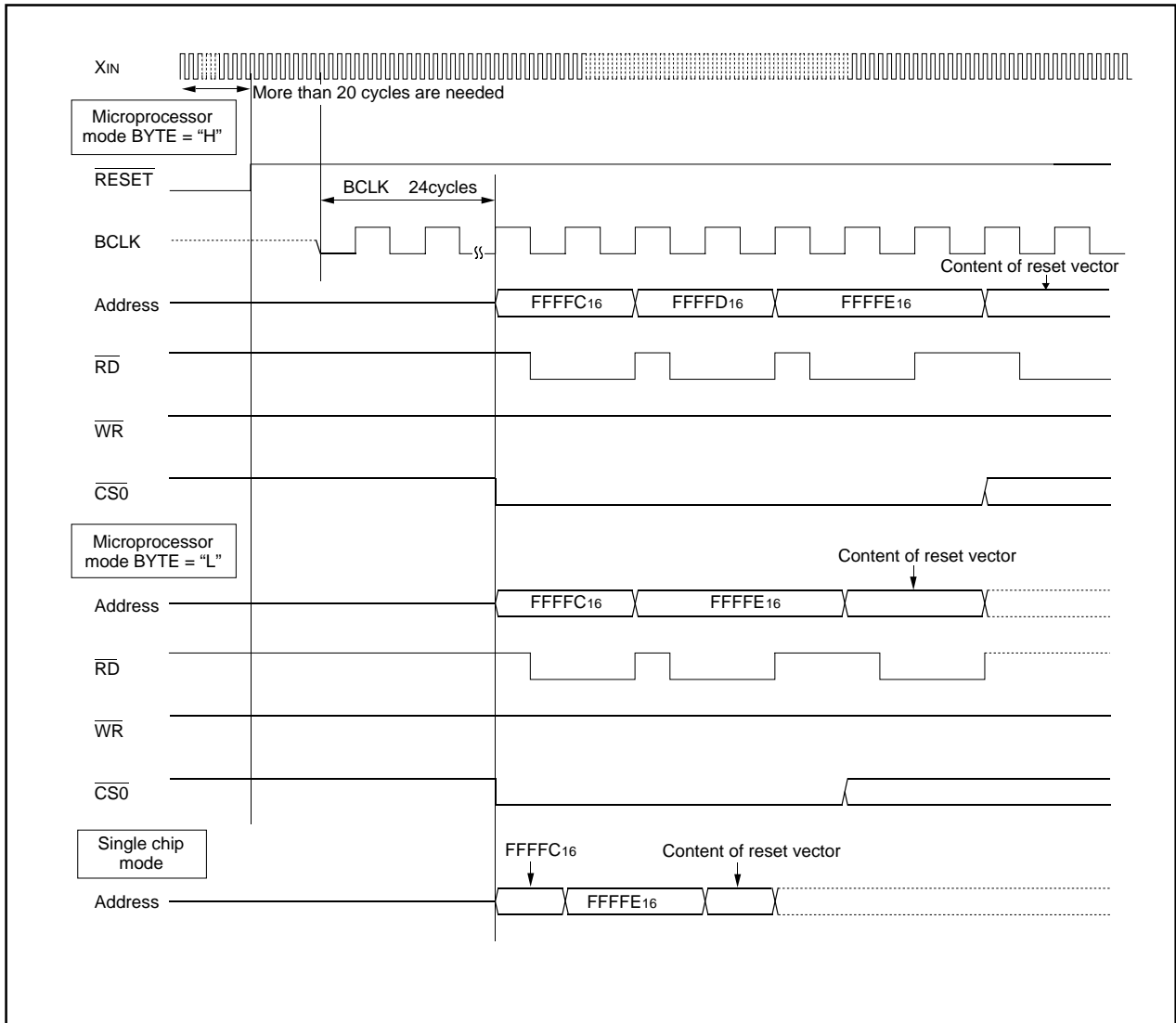


Figure 2.3.2 Reset sequence

**2.3.2 Pin Status When  $\overline{\text{RESET}}$  Pin Level is “L”**

Table 2.3.1 shows the statuses of the other pins while the  $\overline{\text{RESET}}$  pin level is “L”. Figures 2.3.3 and 2.3.4 show the internal status of the microcomputer immediately after the reset is cancelled.

**Table 2.3.1 Pin status when  $\overline{\text{RESET}}$  pin level is “L”**

Pin name	Status		
	CNVss = Vss	CNVss = Vcc	
		BYTE = Vss	BYTE = Vcc
P0	Input port (floating)	Data input (floating)	Data input (floating)
P1	Input port (floating)	Data input (floating)	Input port (floating)
P2, P3, P40 to P43	Input port (floating)	Address output (undefined)	Address output (undefined)
P44	Input port (floating)	$\overline{\text{CS0}}$ output (“H” level is output)	$\overline{\text{CS0}}$ output (“H” level is output)
P45 to P47	Input port (floating)	Input port (floating) (pull-up resistor is on)	Input port (floating) (pull-up resistor is on)
P50	Input port (floating)	$\overline{\text{WR}}$ output (“H” level is output)	$\overline{\text{WR}}$ output (“H” level is output)
P51	Input port (floating)	$\overline{\text{BHE}}$ output (undefined)	$\overline{\text{BHE}}$ output (undefined)
P52	Input port (floating)	$\overline{\text{RD}}$ output (“H” level is output)	$\overline{\text{RD}}$ output (“H” level is output)
P53	Input port (floating)	BCLK output	BCLK output
P54	Input port (floating)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the HOLD pin)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the HOLD pin)
P55	Input port (floating)	$\overline{\text{HOLD}}$ input (floating)	$\overline{\text{HOLD}}$ input (floating)
P56	Input port (floating)	ALE output (“L” level is output)	ALE output (“L” level is output)
P57	Input port (floating)	$\overline{\text{RDY}}$ input (floating)	$\overline{\text{RDY}}$ input (floating)
P60 to P63, P67, P7, P82, P83, P86, P87, P9, P10	Input port (floating)	Input port (floating)	Input port (floating)
R, G, B, OUT1, OUT2	Output port		
CVIN, VHOLD, HLF	Input/output port		
OSC1	Input port		
OSC2	Output port		

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

Processor mode register 0 (Note)	(0004 <sub>16</sub> )...	00 <sub>16</sub>	OSD control register 1	(0202 <sub>16</sub> )...	00 <sub>16</sub>
Processor mode register 1	(0005 <sub>16</sub> )...	0 0 0 0 0 X X 0	OSD control register 2	(0203 <sub>16</sub> )...	00 <sub>16</sub>
System clock control register 0	(0006 <sub>16</sub> )...	48 <sub>16</sub>	Horizontal position register	(0204 <sub>16</sub> )...	00 <sub>16</sub>
System clock control register 1	(0007 <sub>16</sub> )...	20 <sub>16</sub>	Clock control register	(0205 <sub>16</sub> )...	00 <sub>16</sub>
Chip select control register	(0008 <sub>16</sub> )...	01 <sub>16</sub>	I/O polarity control register	(0206 <sub>16</sub> )...	1 0 0 0 X 0 0 0
Address match interrupt enable register	(0009 <sub>16</sub> )...	X X X X X X 0 0	OSD control register 3	(0207 <sub>16</sub> )...	00 <sub>16</sub>
Protect register	(000A <sub>16</sub> )...	X X X X X 0 0 0	Raster color register	(0208 <sub>16</sub> )...	00 <sub>16</sub>
Watchdog timer control register	(000F <sub>16</sub> )...	0 0 0 ? ? ? ? ?	Left border control register	(0250 <sub>16</sub> )...	01 <sub>16</sub>
Address match interrupt register 0	(0010 <sub>16</sub> )...	00 <sub>16</sub>		(0251 <sub>16</sub> )...	X X X X X 0 0 0
	(0011 <sub>16</sub> )...	00 <sub>16</sub>	Right border control register	(0252 <sub>16</sub> )...	00 <sub>16</sub>
	(0012 <sub>16</sub> )...	X X X X 0 0 0 0		(0253 <sub>16</sub> )...	X X X X X 0 0 0
Address match interrupt register 1	(0014 <sub>16</sub> )...	00 <sub>16</sub>	SPRITE horizontal position register 2	(0257 <sub>16</sub> )...	X X X X X 0 0 0
	(0015 <sub>16</sub> )...	00 <sub>16</sub>	SPRITE OSD control register	(0258 <sub>16</sub> )...	X X X 0 0 0 0 0
	(0016 <sub>16</sub> )...	X X X X 0 0 0 0	OSD control register 4	(025F <sub>16</sub> )...	X X X X X X 0 0
DMA0 control register	(002C <sub>16</sub> )...	0 0 0 0 0 ? 0 0	OSD reserved register 1	(025D <sub>16</sub> )...	00 <sub>16</sub>
DMA1 control register	(003C <sub>16</sub> )...	0 0 0 0 0 ? 0 0	Data slicer control register 1	(0260 <sub>16</sub> )...	00 <sub>16</sub>
OSD1 interrupt control register	(0044 <sub>16</sub> )...	X X X X ? 0 0 0	Data slicer control register 2	(0261 <sub>16</sub> )...	? 0 ? 0 ? ? 0 ?
OSD2 interrupt control register	(0048 <sub>16</sub> )...	X X X X ? 0 0 0	Caption position register	(0266 <sub>16</sub> )...	0 0 ? 0 0 0 0 0
Bus collision detection interrupt control register	(004A <sub>16</sub> )...	X X X X ? 0 0 0	Data slicer reserved register 2	(0267 <sub>16</sub> )...	00 <sub>16</sub>
DMA0 interrupt control register	(004B <sub>16</sub> )...	X X X X ? 0 0 0	Data slicer reserved register 1	(0268 <sub>16</sub> )...	00 <sub>16</sub>
DMA1 interrupt control register	(004C <sub>16</sub> )...	X X X X ? 0 0 0	Clock run-in detect register	(0269 <sub>16</sub> )...	00 <sub>16</sub>
A-D conversion interrupt control register	(004E <sub>16</sub> )...	X X X X ? 0 0 0	Data clock position register	(026A <sub>16</sub> )...	X X X 0 0 0 0 1
UART2 transmit interrupt control register	(004F <sub>16</sub> )...	X X X X ? 0 0 0	OSD reserved register 3	(027B <sub>16</sub> )...	00 <sub>16</sub>
UART2 receive interrupt control register	(0050 <sub>16</sub> )...	X X X X ? 0 0 0	OSD reserved register 2	(027C <sub>16</sub> )...	00 <sub>16</sub>
UART0 transmit interrupt control register	(0051 <sub>16</sub> )...	X X X X ? 0 0 0	Peripheral mode register	(027D <sub>16</sub> )...	0 X X 0 0 0 0 0
UART0 receive interrupt control register	(0052 <sub>16</sub> )...	X X X X ? 0 0 0	Hsync counter register	(027E <sub>16</sub> )...	X X X 0 0 X 0 0
Data slicer interrupt control register	(0053 <sub>16</sub> )...	X X X X ? 0 0 0			
Vsync interrupt control register	(0054 <sub>16</sub> )...	X X X X ? 0 0 0			
Timer A0 interrupt control register	(0055 <sub>16</sub> )...	X X X X ? 0 0 0			
Timer A1 interrupt control register	(0056 <sub>16</sub> )...	X X X X ? 0 0 0			
Timer A2 interrupt control register	(0057 <sub>16</sub> )...	X X X X ? 0 0 0			
Timer A3 interrupt control register	(0058 <sub>16</sub> )...	X X X X ? 0 0 0			
Timer A4 interrupt control register	(0059 <sub>16</sub> )...	X X X X ? 0 0 0			
Timer B0 interrupt control register	(005A <sub>16</sub> )...	X X X X ? 0 0 0			
Timer B1 interrupt control register	(005B <sub>16</sub> )...	X X X X ? 0 0 0			
Timer B2 interrupt control register	(005C <sub>16</sub> )...	X X X X ? 0 0 0			
INT0 interrupt control register	(005D <sub>16</sub> )...	X X 0 0 ? 0 0 0			
INT1 interrupt control register	(005E <sub>16</sub> )...	X X 0 0 ? 0 0 0			

X : Nothing is mapped to this bit  
? : Undefined

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

**Note:** When the VCC level is applied to the CNVSS pin, it is 0316 at a reset.

Figure 2.3.3 Device's internal status after a reset is cleared (1)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

Reserved register 1	(0340 <sub>16</sub> )...	0 0 0 ? ? ? ? ?	A-D control register 0	(03D6 <sub>16</sub> )...	0 0 0 0 0 ? ? ? ?
Reserved register 0	(0348 <sub>16</sub> )...	00 <sub>16</sub>	A-D control register 1	(03D7 <sub>16</sub> )...	00 <sub>16</sub>
Interrupt request cause select register	(035F <sub>16</sub> )...	00 <sub>16</sub>	D-A control register	(03DC <sub>16</sub> )...	00 <sub>16</sub>
Reserved register 3	(0362 <sub>16</sub> )...	40 <sub>16</sub>	Port P0 direction register	(03E2 <sub>16</sub> )...	00 <sub>16</sub>
Reserved register 4	(0366 <sub>16</sub> )...	40 <sub>16</sub>	Port P1 direction register	(03E3 <sub>16</sub> )...	00 <sub>16</sub>
Reserved register 5	(0376 <sub>16</sub> )...	00 <sub>16</sub>	Port P2 direction register	(03E6 <sub>16</sub> )...	00 <sub>16</sub>
UART2 special mode register	(0377 <sub>16</sub> )...	00 <sub>16</sub>	Port P3 direction register	(03E7 <sub>16</sub> )...	00 <sub>16</sub>
UART2 transmit/receive mode register	(0378 <sub>16</sub> )...	00 <sub>16</sub>	Port P4 direction register	(03EA <sub>16</sub> )...	00 <sub>16</sub>
UART2 transmit/receive control register 0	(037C <sub>16</sub> )...	08 <sub>16</sub>	Port P5 direction register	(03EB <sub>16</sub> )...	00 <sub>16</sub>
UART2 transmit/receive control register 1	(037D <sub>16</sub> )...	02 <sub>16</sub>	Port P6 direction register	(03EE <sub>16</sub> )...	00 <sub>16</sub>
Count start flag	(0380 <sub>16</sub> )...	00 <sub>16</sub>	Port P7 direction register	(03EF <sub>16</sub> )...	00 <sub>16</sub>
Clock prescaler reset flag	(0381 <sub>16</sub> )...	0 x x x x x x x	Port P8 direction register	(03F2 <sub>16</sub> )...	0 0 x 0 0 0 0 0 0
One-shot start flag	(0382 <sub>16</sub> )...	0 0 x 0 0 0 0 0 0	Port P9 direction register	(03F3 <sub>16</sub> )...	00 <sub>16</sub>
Trigger select register	(0383 <sub>16</sub> )...	00 <sub>16</sub>	Port P10 direction register	(03F6 <sub>16</sub> )...	00 <sub>16</sub>
Up-down flag	(0384 <sub>16</sub> )...	00 <sub>16</sub>	Pull-up control register 0	(03FC <sub>16</sub> )...	00 <sub>16</sub>
Timer A0 mode register	(0396 <sub>16</sub> )...	00 <sub>16</sub>	Pull-up control register 1(Note)	(03FD <sub>16</sub> )...	00 <sub>16</sub>
Timer A1 mode register	(0397 <sub>16</sub> )...	00 <sub>16</sub>	Pull-up control register 2	(03FE <sub>16</sub> )...	00 <sub>16</sub>
Timer A2 mode register	(0398 <sub>16</sub> )...	00 <sub>16</sub>	Port control register	(03FF <sub>16</sub> )...	00 <sub>16</sub>
Timer A3 mode register	(0399 <sub>16</sub> )...	00 <sub>16</sub>	Data registers (R0/R1/R2/R3)		0000 <sub>16</sub>
Timer A4 mode register	(039A <sub>16</sub> )...	00 <sub>16</sub>	Address registers (A0/A1)		0000 <sub>16</sub>
Timer B0 mode register	(039B <sub>16</sub> )...	0 0 ? x 0 0 0 0 0	Frame base register (FB)		0000 <sub>16</sub>
Timer B1 mode register	(039C <sub>16</sub> )...	0 0 ? x 0 0 0 0 0	Interrupt table register (INTB)		00000 <sub>16</sub>
Timer B2 mode register	(039D <sub>16</sub> )...	0 0 ? x 0 0 0 0 0	User stack pointer (USP)		0000 <sub>16</sub>
UART0 transmit/receive mode register	(03A0 <sub>16</sub> )...	00 <sub>16</sub>	Interrupt stack pointer (ISP)		0000 <sub>16</sub>
UART0 transmit/receive control register 0	(03A4 <sub>16</sub> )...	08 <sub>16</sub>	Static base register (SB)		0000 <sub>16</sub>
UART0 transmit/receive control register 1	(03A5 <sub>16</sub> )...	02 <sub>16</sub>	Flag register (FLG)		0000 <sub>16</sub>
Reserved register 2	(03A8 <sub>16</sub> )...	00 <sub>16</sub>			
UART transmit/receive control register 2	(03B0 <sub>16</sub> )...	x 0 0 0 0 0 0 0 0			
DMA0 request cause select register	(03B8 <sub>16</sub> )...	00 <sub>16</sub>			
DMA1 request cause select register	(03BA <sub>16</sub> )...	00 <sub>16</sub>			
A-D control register 2	(03D4 <sub>16</sub> )...	0 0 0 0 ? ? ? 0			

x : Nothing is mapped to this bit  
? : Undefined

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

**Note:** When the VCC level is applied to the CNVSS pin, it is 02<sub>16</sub> at a reset.

Figure 2.3.4 Device's internal status after a reset is cleared (2)

## 2.4 Processor Mode

### 2.4.1 Types of Processor Mode

One of three processor modes can be selected: single-chip mode, memory expansion mode, and microprocessor mode. The functions of some pins, the memory map, and the access space differ according to the selected processor mode.

#### (1) Single-chip mode

In single-chip mode, only internal memory space (SFR, OSD RAM, internal RAM, and internal ROM) can be accessed. Ports P0 to P10 can be used as programmable I/O ports or as I/O ports for the internal peripheral functions.

#### (2) Memory expansion mode

In memory expansion mode, external memory can be accessed in addition to the internal memory space (SFR, OSD RAM, internal RAM, and internal ROM).

In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus and register settings. (See "2.4.3 Bus Settings" for details.)

#### (3) Microprocessor mode

In microprocessor mode, the SFR, OSD RAM, internal RAM, and external memory space can be accessed. The internal ROM area cannot be accessed.

In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus and register settings. (See "2.4.3 Bus Settings" for details.)

### 2.4.2 Setting Processor Modes

The processor mode is set using the CNVSS pin and the processor mode bits (bits 1 and 0 at address 000416). Do not set the processor mode bits to "102".

Regardless of the level of the CNVSS pin, changing the processor mode bits selects the mode. Therefore, never change the processor mode bits when changing the contents of other bits. Also do not attempt to shift to or from the microprocessor mode within the program stored in the internal ROM area.

#### (1) Applying VSS to CNVSS pin

The microcomputer begins operation in single-chip mode after being reset. Memory expansion mode is selected by writing "012" to the processor mode bits.

#### (2) Applying VCC to CNVSS pin

The microcomputer starts to operate in microprocessor mode after being reset.

Figures 2.4.1 and 2.4.2 show the processor mode register 0 and 1.

Figure 2.4.3 shows the memory maps applicable for each of the modes.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

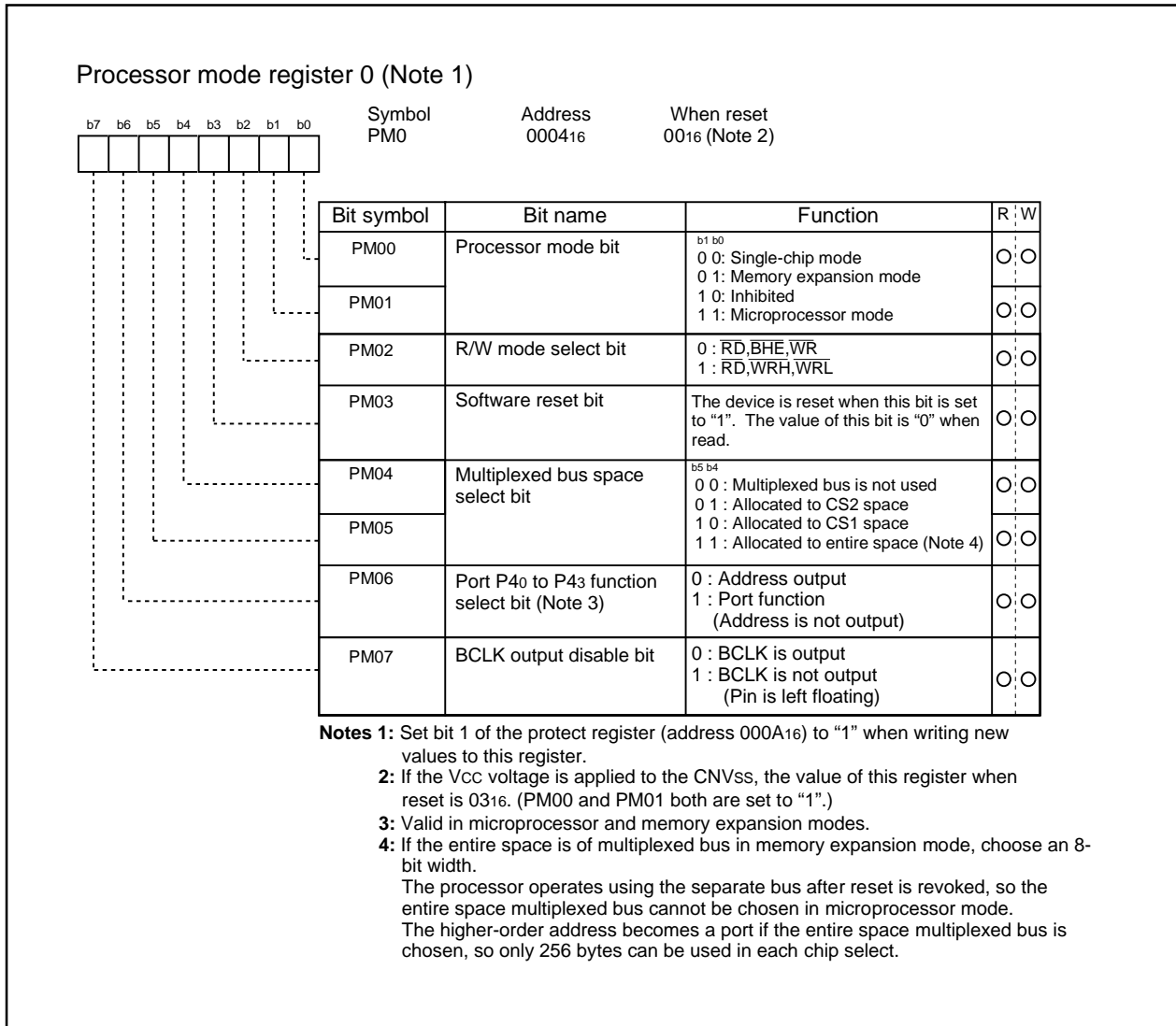


Figure 2.4.1 Processor mode register 0



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

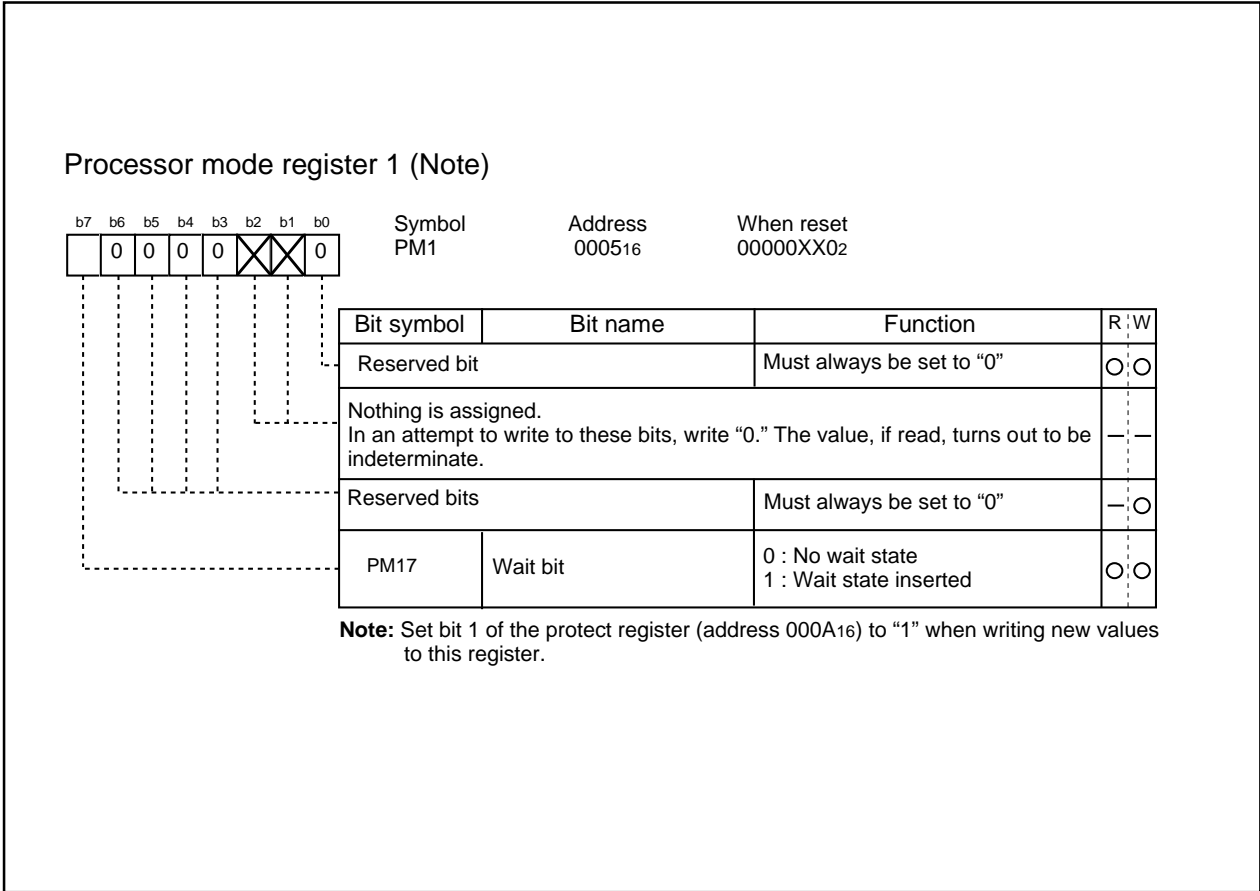


Figure 2.4.2 Processor mode register 1

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

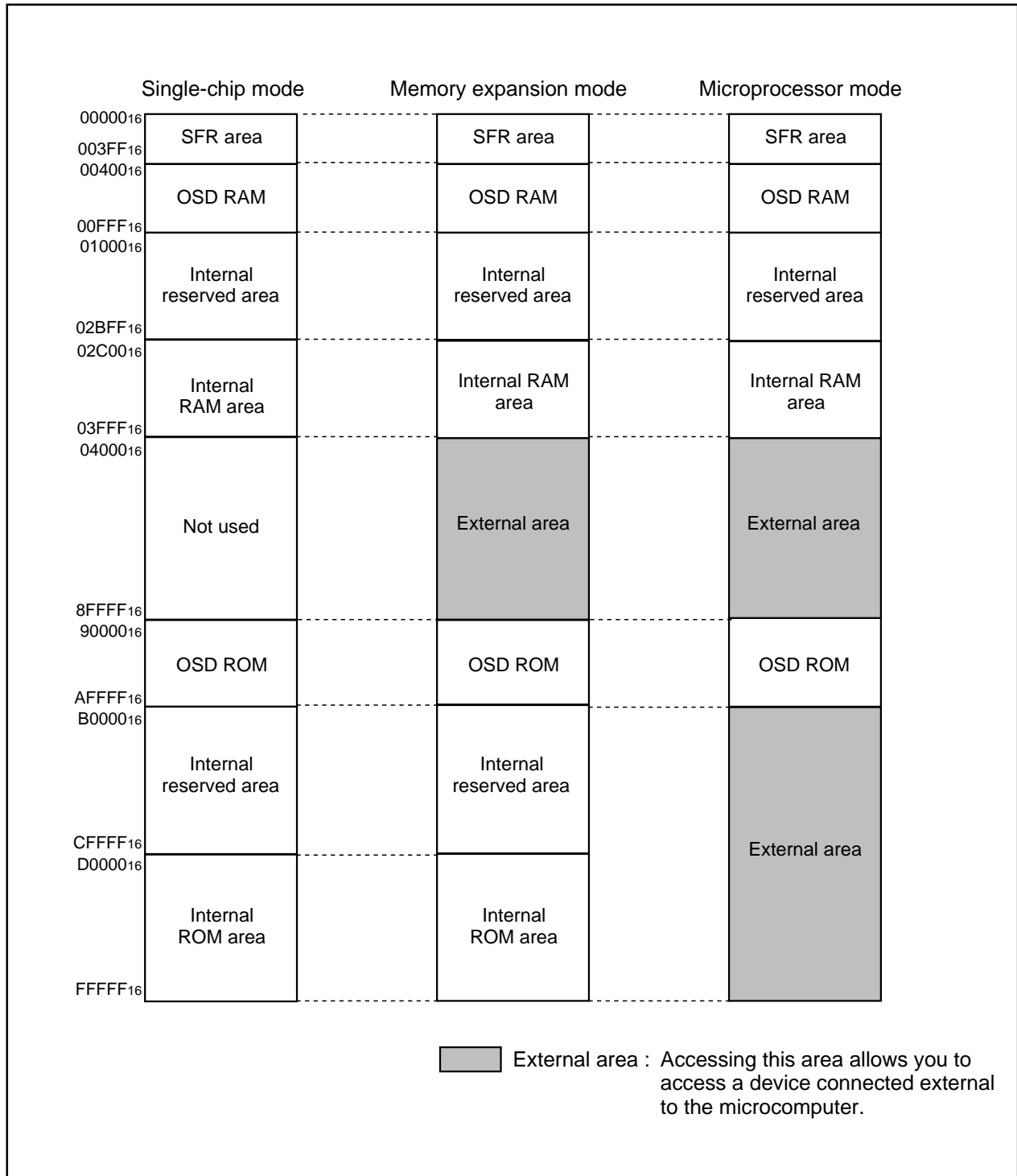


Figure 2.4.3 Memory maps in each processor mode

### 2.4.3 Bus Settings

The BYTE pin and bits 4 to 6 of the processor mode register 0 (address 000416) are used to change the bus settings.

Table 2.4.1 shows the factors used to change the bus settings.

**Table 2.4.1 Factors for switching bus settings**

Bus setting	Switching factor
Switching external address bus width	Bit 6 of processor mode register 0
Switching external data bus width	BYTE pin
Switching between separate and multiplex bus	Bits 4 and 5 of processor mode register 0

#### (1) Selecting external address bus width

The address bus width for external output in the 1M bytes of address space can be set to 16 bits (64K bytes address space) or 20 bits (1M bytes address space). When bit 6 of the processor mode register 0 is set to "1", the external address bus width is set to 16 bits, and P2 and P3 become part of the address bus. P40 to P43 can be used as programmable I/O ports. When bit 6 of processor mode register 0 is set to "0", the external address bus width is set to 20 bits, and P2, P3, and P40 to P43 become part of the address bus.

#### (2) Selecting external data bus width

The external data bus width can be set to 8 or 16 bits. (Note, however, that only the separate bus can be set.) When the BYTE pin is "L", the bus width is set to 16 bits; when "H", it is set to 8 bits. (The internal bus width is permanently set to 16 bits.)

While operating, fix the BYTE pin either to "H" or to "L."

#### (3) Selecting separate/multiplex bus

The bus format can be set to multiplex or separate bus using bits 4 and 5 of the processor mode register 0.

##### • Separate bus

In this mode, the data and address are input and output separately. The data bus can be set using the BYTE pin to be 8 or 16 bits. When the BYTE pin is "H", the data bus is set to 8 bits and P0 functions as the data bus and P1 as a programmable I/O port. When the BYTE pin is "L", the data bus is set to 16 bits and P0 and P1 are both used for the data bus.

When the separate bus is used for access, a software wait can be selected.

##### • Multiplex bus

In this mode, data and address I/O are time multiplexed. With an 8-bit data bus selected (BYTE pin = "H"), the 8 bits from D0 to D7 are multiplexed with A0 to A7.

With a 16-bit data bus selected (BYTE pin = "L"), the 8 bits from D0 to D7 are multiplexed with A1 to A8. D8 to D15 are not multiplexed. In this case, the external devices connected to the multiplexed bus are mapped to the microcomputer's even addresses (every 2nd address). To access these external devices, access the even addresses as bytes.

The ALE signal latches the address. It is output from P56.

Before using the multiplex bus for access, be sure to insert a software wait.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

In memory expansion mode, select a 8-bit multiplex bus.

The processor operates using the separate bus after reset is revoked, so the entire space multiplexed bus cannot be chosen in microprocessor mode.

The higher-order address becomes a port if the entire space multiplexed bus is chosen, so only 256 bytes can be used in each chip select.

**Table 2.4.2 Pin functions for each processor mode**

Processor mode	Single-chip mode	Memory expansion mode/microprocessor modes				Memory expansion mode
		"01", "10" [ Either CS1 or CS2 is for multiplexed bus and others are for separate bus ]		"00" (separate bus)		
Multiplexed bus space select bit						"11" (Note 1) [ Multiplexed bus for the entire space ]
Data bus width BYTE pin level		8 bits = "H"	16 bits = "L"	8 bits = "H"	16 bits = "L"	8 bits = "H"
P00 to P07	I/O port	Data bus	Data bus	Data bus	Data bus	I/O port
P10 to P17	I/O port	I/O port	Data bus	I/O port	Data bus	I/O port
P20	I/O port	Address bus /data bus(Note 3)	Address bus	Address bus	Address bus	Address bus /data bus
P21 to P27	I/O port	Address bus	Address bus /data bus(Note 3)	Address bus /data bus(Note 3)	Address bus	Address bus /data bus
P30	I/O port	Address bus /data bus(Note 3)	Address bus	Address bus	Address bus	A8/D7
P31 to P37	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port
P40 to P43 Port P40 to P43 function select bit = 1	I/O port	I/O port	I/O port	I/O port	I/O port	I/O port
P40 to P43 Port P40 to P43 function select bit = 0	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port
P44 to P47	I/O port	$\overline{CS}$ (chip select) or programmable I/O port (For details, refer to "2.4.4 Bus control")				
P50 to P53	I/O port	Outputs $\overline{RD}$ , $\overline{WRL}$ , $\overline{WRH}$ , and $\overline{BCLK}$ or $\overline{RD}$ , $\overline{BHE}$ , $\overline{WR}$ , and $\overline{BCLK}$ (For details, refer to "2.4.4 Bus control")				
P54	I/O port	$\overline{HLDA}$	$\overline{HLDA}$	$\overline{HLDA}$	$\overline{HLDA}$	$\overline{HLDA}$
P55	I/O port	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$
P56	I/O port	ALE	ALE	ALE	ALE	ALE
P57	I/O port	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$

**Notes 1:** In memory expansion mode, select a 8-bit multiplex bus.

The processor operates using the separate bus after reset is revoked, so the entire space multiplexed bus cannot be chosen in microprocessor mode.

The higher-order address becomes a port if the entire space multiplexed bus is chosen, so only 256 bytes can be used in each chip select.

**2:** Address bus when in separate bus mode.

### 2.4.4 Bus Control

The following explains the signals required for accessing external devices and software waits. The signals required for accessing the external devices are valid when the processor mode is set to memory expansion mode and microprocessor mode. The software waits are valid in all processor modes.

#### (1) Address bus/data bus

The address bus consists of the 20 pins A0 to A19 for accessing the 1M bytes of address space.

The data bus consists of the pins for data I/O. When the BYTE pin is "H", the 8 ports D0 to D7 function as the data bus. When BYTE is "L", the 16 ports D0 to D15 function as the data bus.

When a change is made from single-chip mode to memory expansion mode, the value of the address bus is undefined until external memory is accessed.

#### (2) Chip select signal

The chip select signal is output using the same pins as P44 to P47. Bits 0 to 3 of the chip select control register (address 000816) set each pin to function as a port or to output the chip select signal. The chip select control register is valid in memory expansion mode and microprocessor mode. In single-chip mode, P44 to P47 function as programmable I/O ports regardless of the value in the chip select control register.

In microprocessor mode, only  $\overline{CS0}$  outputs the chip select signal after the reset state has been cancelled.  $\overline{CS1}$  to  $\overline{CS3}$  function as input ports. Figure 2.4.4 shows the chip select control register.

The chip select signal can be used to split the external area into as many as four blocks. Table 2.4.4 shows the external memory areas specified using the chip select signal.

**Table 2.4.3 External areas specified by the chip select signals**

Chip select	Specified address range	
	Memory expansion mode	Microprocessor mode
$\overline{CS0}$	30000 <sub>16</sub> to 8FFFF <sub>16</sub> (384K)	30000 <sub>16</sub> to 8FFFF <sub>16</sub> (384K), B0000 <sub>16</sub> to FFFFF <sub>16</sub> (320K)
$\overline{CS1}$	28000 <sub>16</sub> to 2FFFF <sub>16</sub> (32K)	28000 <sub>16</sub> to 2FFFF <sub>16</sub> (32K)
$\overline{CS2}$	08000 <sub>16</sub> to 27FFF <sub>16</sub> (128K)	08000 <sub>16</sub> to 27FFF <sub>16</sub> (128K)
$\overline{CS3}$	04000 <sub>16</sub> to 07FFF <sub>16</sub> (16K)	04000 <sub>16</sub> to 07FFF <sub>16</sub> (16K)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

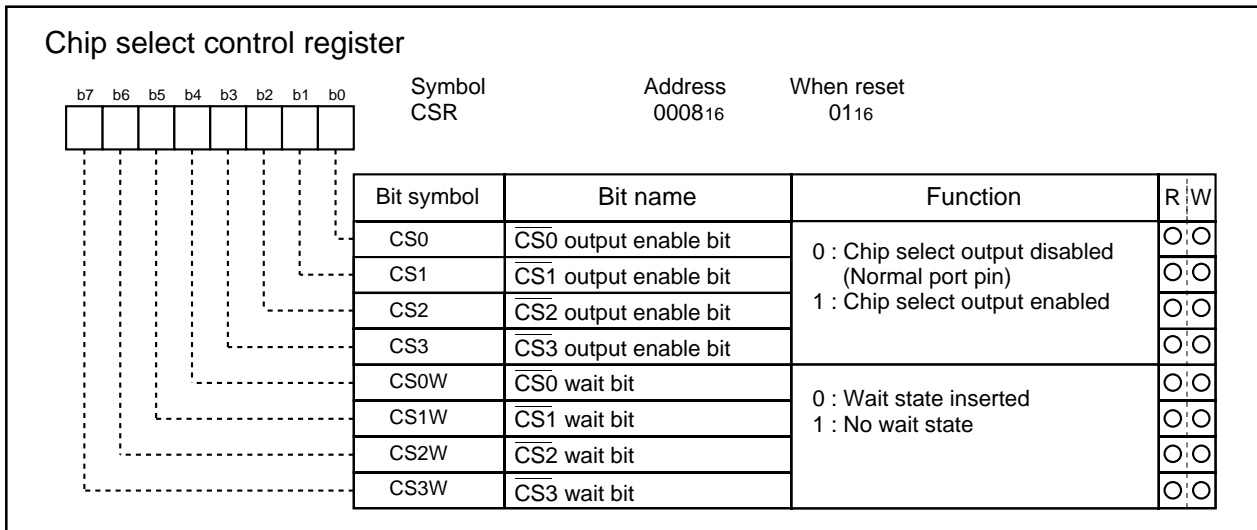


Figure 2.4.4 Chip select control register

**(3) Read/write signals**

With a 16-bit data bus (BYTE pin = "L"), bit 2 of the processor mode register 0 (address 0004<sub>16</sub>) select the combinations of  $\overline{RD}$ ,  $\overline{BHE}$ , and  $\overline{WR}$  signals or  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  signals. With an 8-bit data bus (BYTE pin = "H"), use the combination of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$  signals. (Set bit 2 of the processor mode register 0 (address 0004<sub>16</sub>) to "0".) Tables 2.4.4 and 2.4.5 show the operation of these signals. After a reset has been cancelled, the combination of  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{BHE}$  signals is automatically selected.

When switching to the  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  combination, do not write to external memory until bit 2 of the processor mode register 0 (address 0004<sub>16</sub>) has been set (Note).

**Note:** Before attempting to change the contents of the processor mode register 0, set bit 1 of the protect register (address 000A<sub>16</sub>) to "1".

Table 2.4.4 Operation of  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  signals

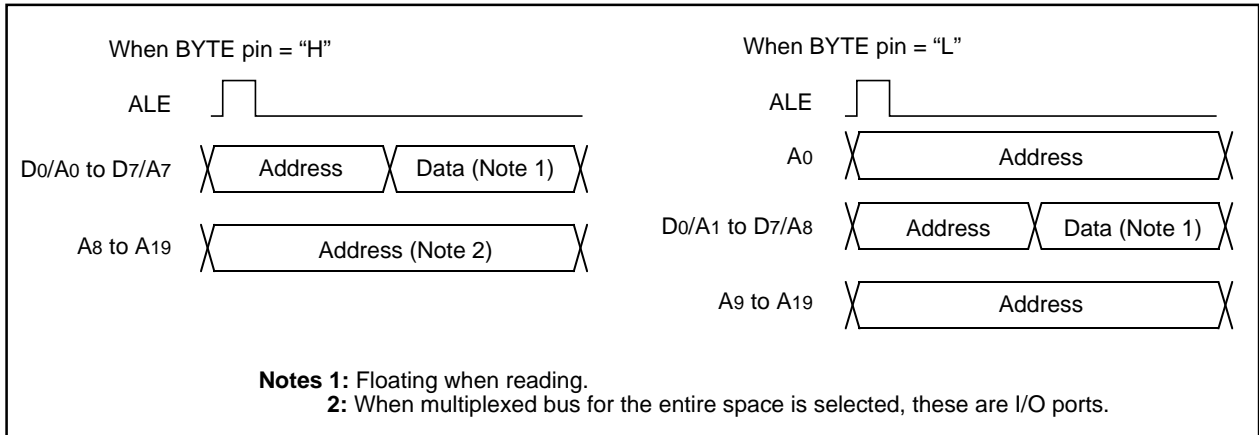
Data bus width	$\overline{RD}$	$\overline{WRL}$	$\overline{WRH}$	Status of external data bus
16-bit (BYTE = "L")	L	H	H	Read data
	H	L	H	Write 1 byte of data to even address
	H	H	L	Write 1 byte of data to odd address
	H	L	L	Write data to both even and odd addresses

Table 2.4.5 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 = "L")	H	L	L	H	Write 1 byte of data to odd address
	L	H	L	H	Read 1 byte of data from odd address
	H	L	H	L	Write 1 byte of data to even address
	L	H	H	L	Read 1 byte of data from even address
	H	L	L	L	Write data to both even and odd addresses
	L	H	L	L	Read data from both even and odd addresses
8-bit (BYTE = "H")	H	L	Not used	H / L	Write 1 byte of data
	L	H	Not used	H / L	Read 1 byte of data

**(4) ALE signal**

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



**Figure 2.4.5 ALE signal and address/data bus**

(5) RDY signal

RDY signal facilitates access of external devices that require a long time for access. As shown in Figure 2.4.6, if an "L" is being input to the RDY pin at the BCLK falling edge, the bus turns to the wait state. If an "H" is being input to the RDY pin at the BCLK falling edge, the bus cancels the wait state. Table 2.4.6 shows the microcomputer state in the wait state. Figure 2.4.6 shows the example of the RD signal being extended using the RDY signal.

The RDY signal is valid when accessing the external area during the bus cycle in which bits 4 to 7 of the chip select control register (address 000816) are set to "0." The RDY signal is invalid when setting "1" to all bits 4 to 7 of the chip select control register (address 000816), but the RDY pin should be treated as properly as in non-using.

Table 2.4.6 Microcomputer status in ready state (Note)

Item	Status
Oscillation	On
R/W signal, address bus, data bus, CS	Maintain status when RDY signal received
ALE signal, HLDA, programmable I/O ports	
Internal peripheral circuits	On

Note: The RDY signal cannot be received immediately prior to a software wait.

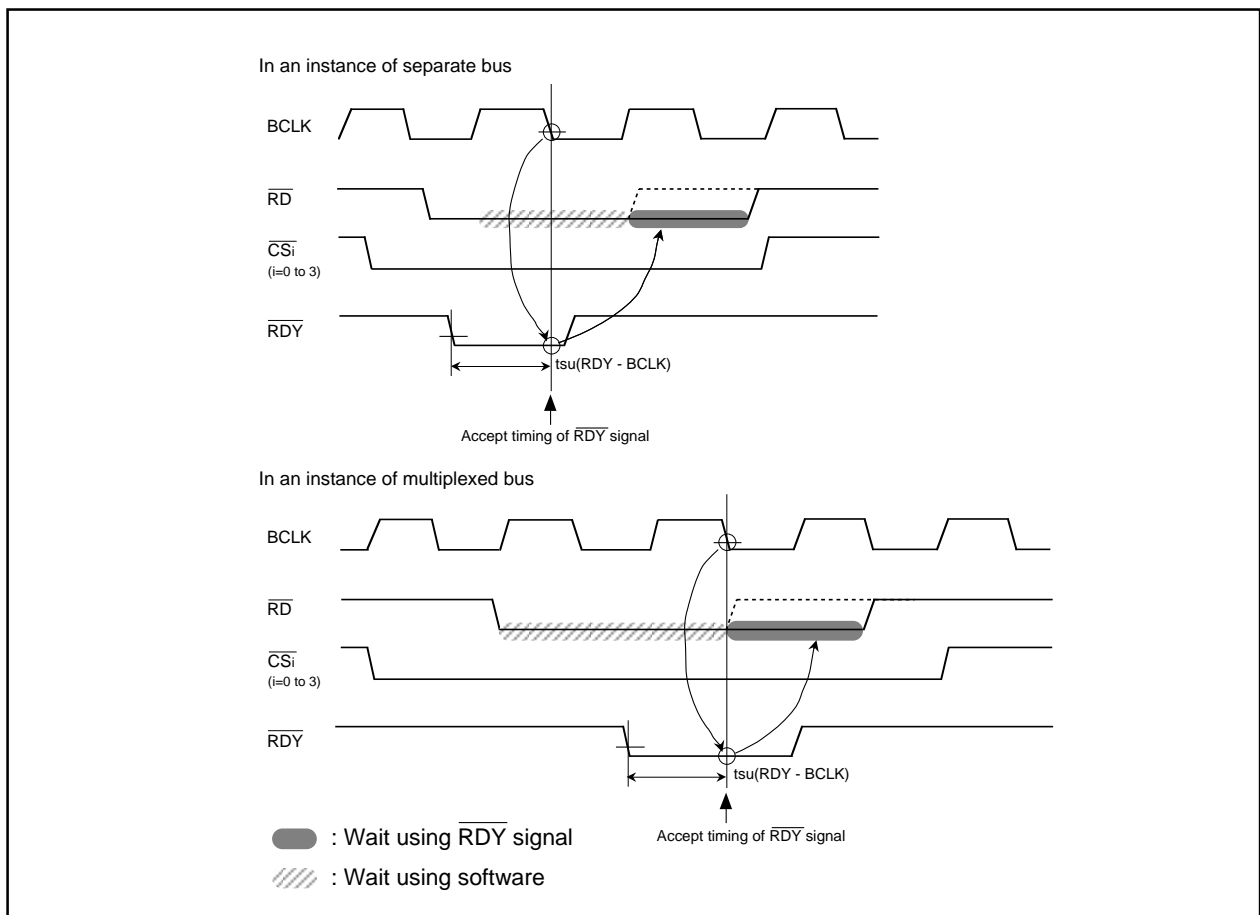


Figure 2.4.6 Example of RD signal extended by RDY signal



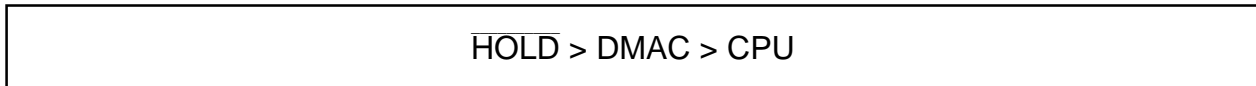
**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**(6) Hold signal**

The hold signal is used to transfer the bus privileges from the CPU to the external circuits. Inputting “L” to the  $\overline{\text{HOLD}}$  pin places the microcomputer in the hold state at the end of the current bus access. This status is maintained and “L” is output from the  $\overline{\text{HLDA}}$  pin as long as “L” is input to the  $\overline{\text{HOLD}}$  pin. Table 2.4.7 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.



**Figure 2.4.7 Bus-using priorities**

**Table 2.4.7 Microcomputer status in hold state**

Item	Status
Oscillation	ON
R/W signal, address bus, data bus, $\overline{\text{CS}}$ , $\overline{\text{BHE}}$	Floating
Programmable I/O ports	P0, P1, P2, P3, P4, P5 P6, P7, P8, P9, P10
	Floating Maintains status when hold signal is received
$\overline{\text{HLDA}}$	Output “L”
Internal peripheral circuits	ON (but watchdog timer stops)
ALE signal	Undefined

**(7) External bus status when internal area is accessed**

Table 2.4.8 shows the external bus status when the internal area is accessed.

**Table 2.4.8 External bus status when the internal area is accessed**

Item	SFR accessed	Internal ROM/RAM accessed
Address bus	Address output	Maintain status before accessed address of external area
Data bus	When read	Floating
	When write	Output data
$\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{WRL}}$ , $\overline{\text{WRH}}$	$\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{WRL}}$ , $\overline{\text{WRH}}$ output	Output “H”
$\overline{\text{BHE}}$	$\overline{\text{BHE}}$ output	Maintain status before accessed status of external area
$\overline{\text{CS}}$	Output “H”	Output “H”
ALE	Output “L”	Output “L”

**(8) BCLK output**

The output of the internal clock  $\phi$  can be selected using bit 7 of the processor mode register 0 (address 000416) (Note). The output is floating when bit 7 is set to “1”.

**Note:** Before attempting to change the contents of the processor mode register 0, set bit 1 of the protect register (address 000A16) to “1”.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**(9) Software wait**

A software wait can be inserted by setting the wait bit (bit 7) of the processor mode register 1 (address 000516) (Note 1) and bits 4 to 7 of the chip select control register (address 000816).

A software wait is inserted in the internal ROM/RAM area, in the OSD RAM area (Note 2), and in the external memory area by setting the wait bit of the processor mode register 1. When set to "0", each bus cycle is executed in one BCLK cycle. When set to "1", each bus cycle is executed in two or three BCLK cycles. After the microcomputer has been reset, this bit defaults to "0". When set to "1", a wait is applied to all memory areas (two or three BCLK cycles), regardless of the contents of bits 4 to 7 of the chip select control register. Set this bit after referring to the recommended operating conditions (main clock input oscillation frequency) of the electric characteristics. However, when the user is using the RDY signal, the relevant bit in the chip select control register's bits 4 to 7 must be set to "0."

When the wait bit of the processor mode register 1 is "0", software waits can be set independently for each of the 4 areas selected using the chip select signal. Bits 4 to 7 of the chip select control register correspond to chip selects  $\overline{CS0}$  to  $\overline{CS3}$ . When one of these bits is set to "1", the bus cycle is executed in one BCLK cycle. When set to "0", the bus cycle is executed in two or three BCLK cycles. These bits default to "0" after the microcomputer has been reset. These bits default to "0" after the microcomputer has been reset.

The SFR area is always accessed in two BCLK cycles regardless of the setting of these control bits. Also, the corresponding bits of the chip select control register must be set to "0" if using the multiplex bus to access the external memory area.

Table 2.4.9 shows the software wait and bus cycles. Figure 2.4.8 shows example bus timing when using software waits.

**Notes 1:** Before attempting to change the contents of the processor mode register 1, set bit 1 of the protect register (address 000A16) to "1".

**2:** Be sure to set a software wait to access to OSD RAM.

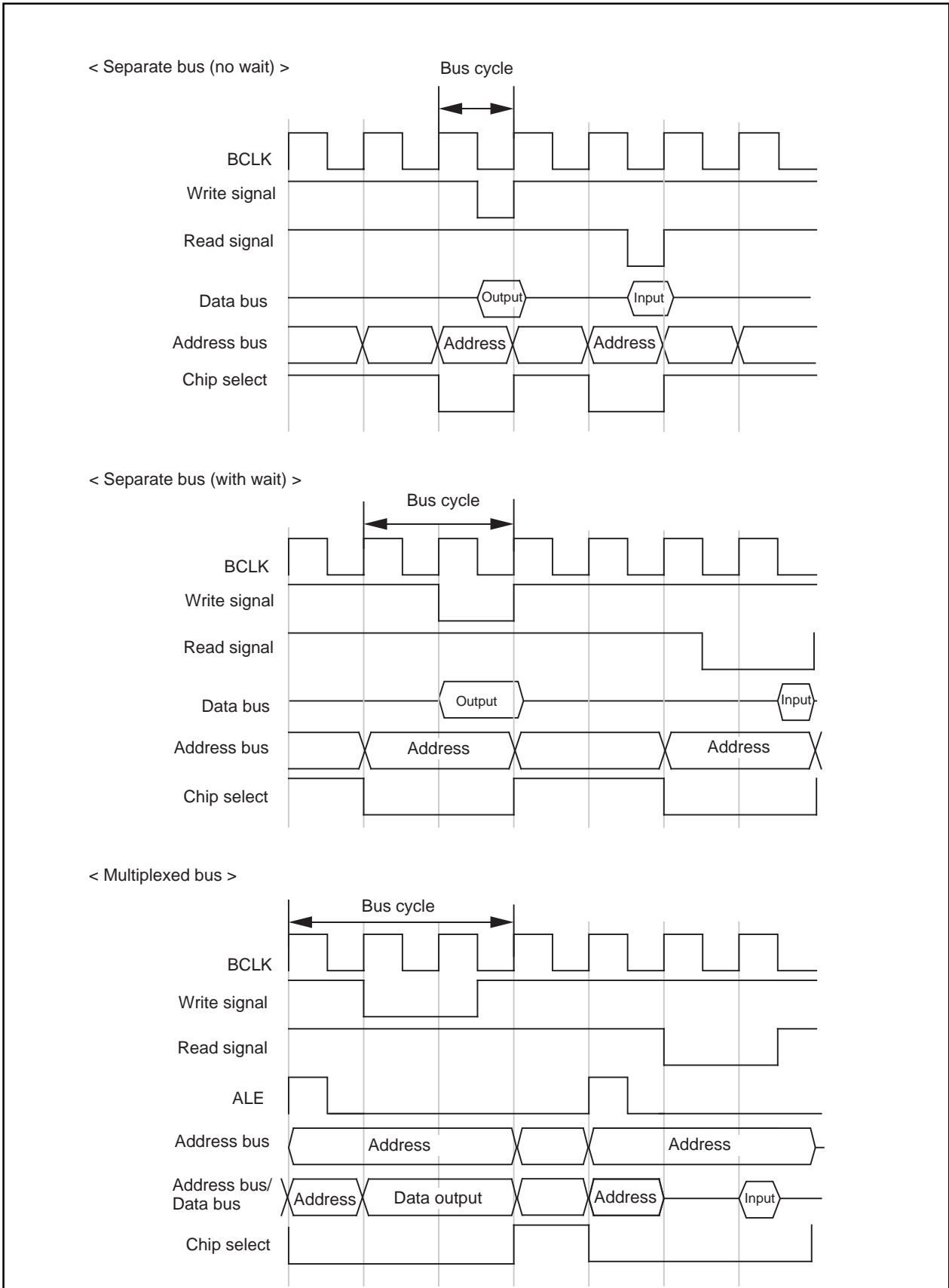
**Table 2.4.9 Software waits and bus cycles**

Area	Bus status	Wait bit	Bits 4 to 7 of chip select control register	Bus cycle
SFR	———	Invalid	Invalid	2 BCLK cycles
Internal ROM/RAM • OSD RAM	———	0	Invalid	1 BCLK cycle
	———	1	Invalid	2 BCLK cycles
External memory area	Separate bus	0	1	1 BCLK cycle
	Separate bus	0	0	2 BCLK cycles
	Separate bus	1	0 (Note)	2 BCLK cycles
	Multiplex bus	0	0	3 BCLK cycles
	Multiplex bus	1	0 (Note)	3 BCLK cycles

**Note:** When using the RDY signal, always set to "0."

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.4.8 Typical bus timings using software wait**

## 2.5 Clock Generating Circuit

The clock generating circuit contains 2 oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units and 1 oscillator circuit that supplies the operating clock source to OSD.

**Table 2.5.1. Clock generating circuits**

	Main clock oscillation circuit	Sub-clock oscillation circuit	OSD oscillation circuit
Use of clock	<ul style="list-style-type: none"> <li>• CPU's operating clock source</li> <li>• Internal peripheral units' operating clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU's operating clock source</li> <li>• Timer A/B's count clock source</li> </ul>	<ul style="list-style-type: none"> <li>• OSD's operating clock source</li> </ul>
Usable oscillator	<ul style="list-style-type: none"> <li>• Ceramic resonator (or quartz-crystal oscillator)</li> </ul>	<ul style="list-style-type: none"> <li>• Quartz-crystal oscillator</li> </ul>	<ul style="list-style-type: none"> <li>• Ceramic resonator (or quartz-crystal oscillator)</li> <li>• LC oscillator</li> </ul>
Pins to connect oscillator	X <sub>IN</sub> , X <sub>OUT</sub>	X <sub>CIN</sub> , X <sub>COU</sub> T	OSC1, OSC2
Oscillation stop/restart function	Available	Available	/
Oscillator status immediately after reset	Oscillating	Stopped	
Other	Externally derived clock can be input		

### 2.5.1 Example of Oscillator Circuit

Figure 2.5.1 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Figure 2.5.2 shows some examples of sub-clock circuits, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Figures 2.5.1 and 2.5.2 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.

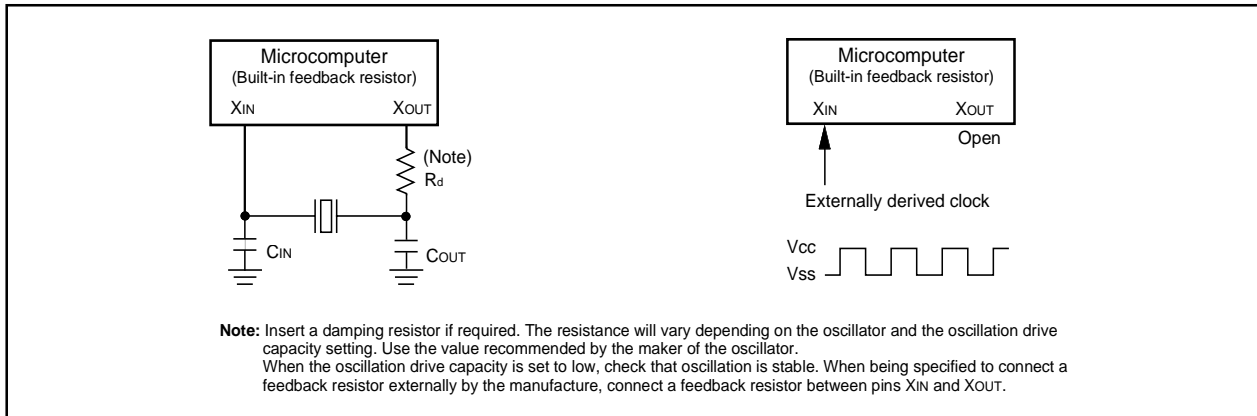


Figure 2.5.1 Examples of main clock

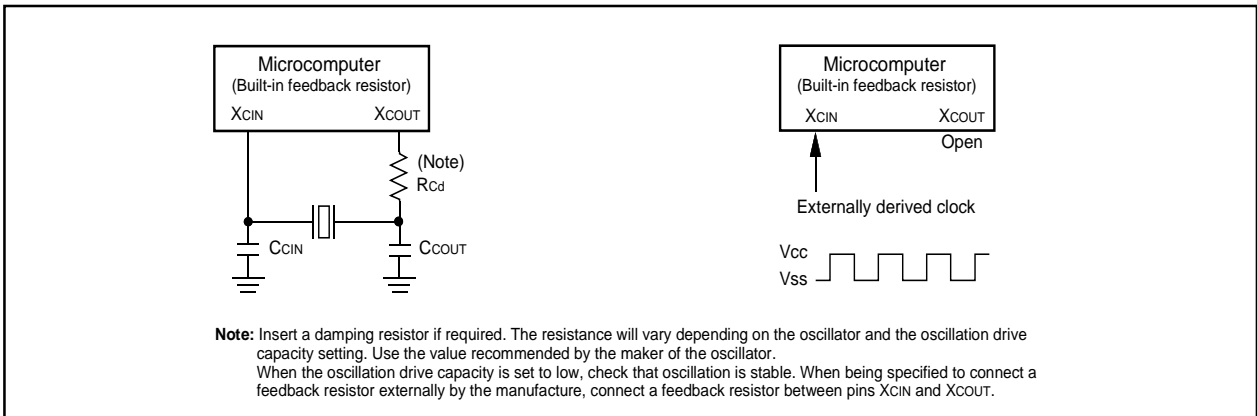


Figure 2.5.2 Examples of sub-clock

### 2.5.2 OSD Oscillation Circuit

The OSD clock oscillation circuit can obtain simply a clock for OSD by connecting an LC oscillator or a ceramic resonator (or a quartz-crystal oscillator) across the pins OSC1 and OSC2. Which of LC oscillator or a ceramic resonator (or a quartz-crystal oscillator) is selected by setting bits 1 and 2 of the clock control register (address 020516).

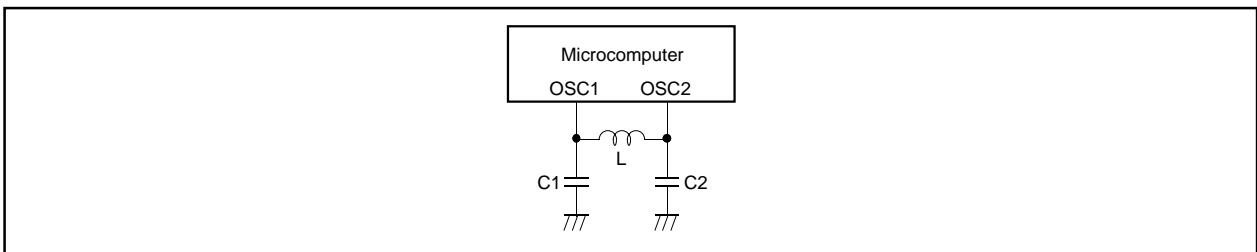
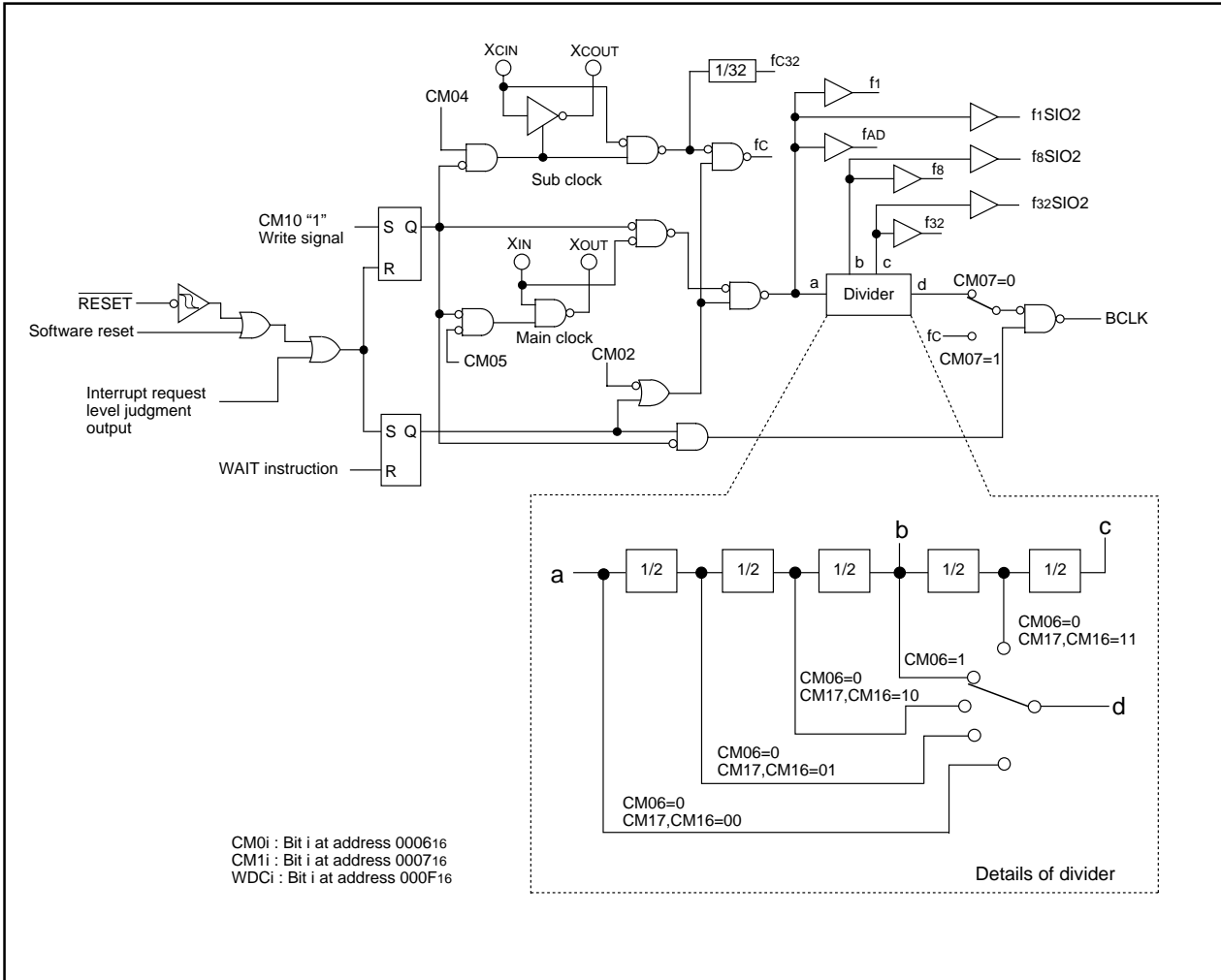


Figure 2.5.3 OSD clock connection example

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



The following paragraphs describes the clocks generated by the clock generating circuit.

**(1) Main clock**

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to the BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006<sub>16</sub>). Stopping the clock, after switching the operating clock source of CPU to the sub-clock, reduces the power dissipation.

After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the X<sub>IN</sub>-X<sub>OUT</sub> drive capacity select bit (bit 5 at address 0007<sub>16</sub>). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit changes to "1" when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

**(2) Sub-clock**

The sub-clock is generated by the sub clock oscillation circuit. No sub clock is generated after a reset. After oscillation is started using the port X<sub>c</sub> select bit (bit 4 at address 0006<sub>16</sub>), the sub-clock can be selected as the BCLK by using the system clock select bit (bit 7 at address 0006<sub>16</sub>). However, be sure that the sub-clock oscillation has fully stabilized before switching.

After the oscillation of the sub-clock oscillation circuit has stabilized, the drive capacity of the sub-clock oscillation circuit can be reduced using the X<sub>CIN</sub>-X<sub>COU</sub>T drive capacity select bit (bit 3 at address 0006<sub>16</sub>). Reducing the drive capacity of the sub-clock oscillation circuit reduces the power dissipation. This bit changes to "1" when shifting to stop mode and at a reset.

**(3) BCLK**

The internal clock  $\phi$  is the clock that drives the CPU, and is  $f_c$  or the clock derived by dividing the main clock by 1, 2, 4, 8, or 16. The BCLK is derived by dividing the main clock by 8 after a reset. The BCLK signal can be output from pin BCLK by the BCLK output disable bit (bit 7 at address 0004<sub>16</sub>) in the memory expansion and the microprocessor modes.

The main clock division select bit 0 (bit 6 at address 0006<sub>16</sub>) changes to "1" when shifting from high-speed/medium-speed to stop mode and at reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

**(4) Peripheral function clock (f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub>, f<sub>1SIO2</sub>, f<sub>8SIO2</sub>, f<sub>32SIO2</sub>, f<sub>AD</sub>)**

The clock for the peripheral devices is derived by dividing the main clock by 1, 8 or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006<sub>16</sub>) to "1" and then executing a WAIT instruction.

**(5) f<sub>c32</sub>**

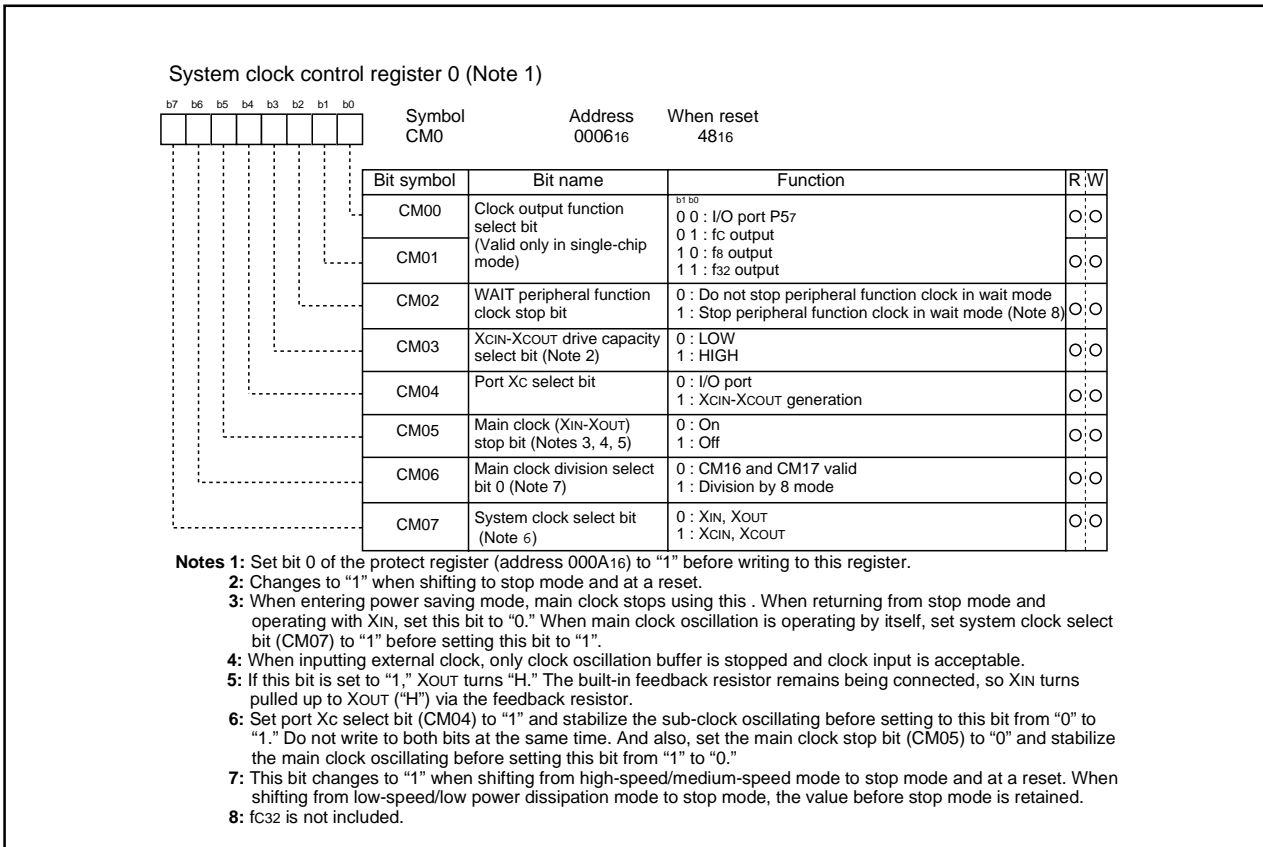
This clock is derived by dividing the sub-clock by 32. It is used for the timer A and timer B counts.

**(6) f<sub>c</sub>**

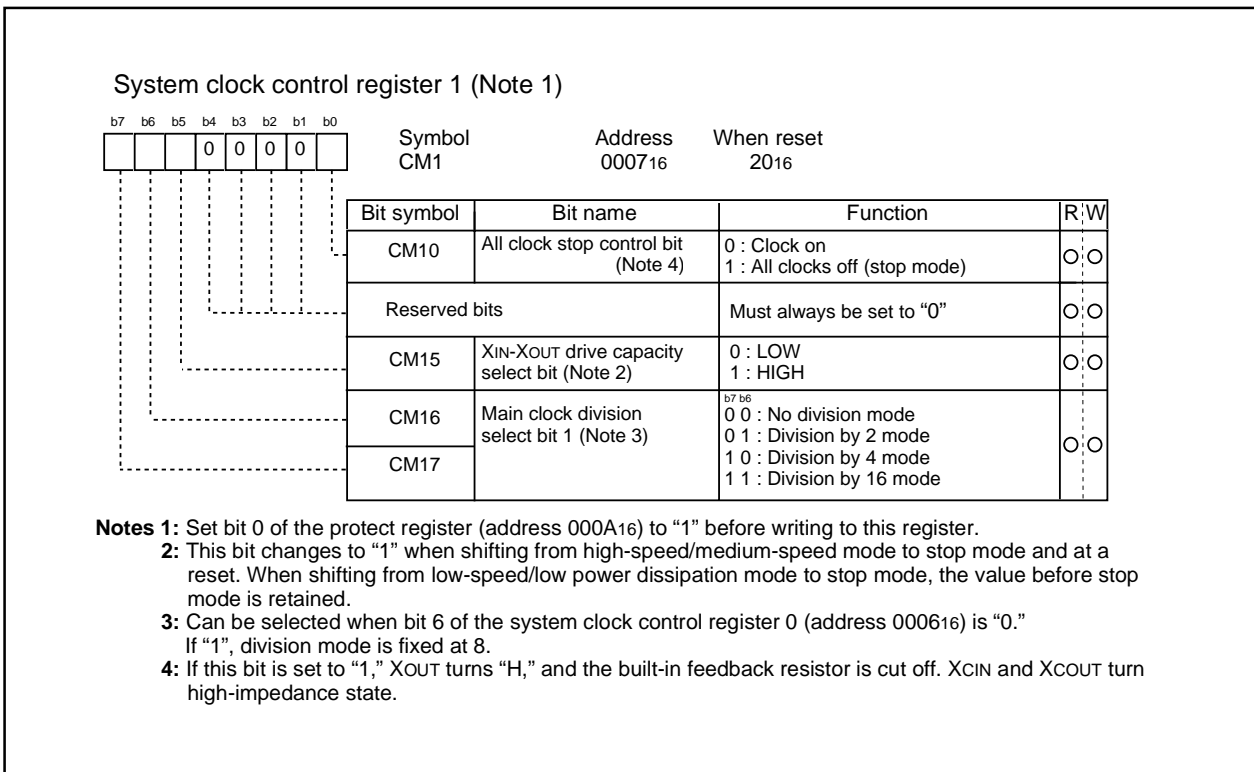
This clock has the same frequency as the sub-clock. It is used for the BCLK and for the watchdog timer.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



Figures 2.5.5 System clock control register 0





**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**2.5.4 Clock Output**

In single-chip mode, the clock output function select bits (bits 0 and 1 at address 0006<sub>16</sub>) enable f<sub>8</sub>, f<sub>32</sub>, or f<sub>c</sub> to be output from the P57/CLKOUT pin. When the WAIT peripheral function clock stop bit (bit 2 at address 0006<sub>16</sub>) is set to "1", the output of f<sub>8</sub> and f<sub>32</sub> stops when a WAIT instruction is executed.

**2.5.5 Stop Mode**

Writing "1" to the all-clock stop control bit (bit 0 at address 0007<sub>16</sub>) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that V<sub>CC</sub> remains above 4.5V.

Because the oscillation, BCLK, f<sub>1</sub> to f<sub>32</sub>, f<sub>1</sub>SI0<sub>2</sub> to f<sub>32</sub>SI0<sub>2</sub>, f<sub>c</sub>, f<sub>c</sub>3<sub>2</sub>, and f<sub>AD</sub> stops in stop mode, peripheral functions such as the A-D converter and watchdog timer do not function. However, timer B operates provided that the event counter mode is set to an external pulse, and UART<sub>i</sub> (i = 0, 2) functions provided an external clock is selected. Table 2.5.2 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or an interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled. If returning by an interrupt, that interrupt routine is executed.

When shifting from high-speed/medium-speed mode to stop mode and at a reset, the main clock division select bit 0 (bit 6 at address 0006<sub>16</sub>) is set to "1." When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

**Table 2.5.2 Port status during stop mode**

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$		Retains status before stop mode	/
RD, WR, BHE, WRL, WRH		"H"	
HLDA, BCLK		"H"	
ALE		"H"	
Port		Retains status before stop mode	Retains status before stop mode
CLKOUT	When f <sub>c</sub> selected	Valid only in single-chip mode	"H"
	When f <sub>8</sub> , f <sub>32</sub> selected	Valid only in single-chip mode	Retains status before stop mode

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**2.5.6 Wait Mode**

When a WAIT instruction is executed, the BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but the BCLK and watchdog timer stop. Writing "1" to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. Table 2.5.3 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or an interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts from the interrupt routine using as BCLK, the clock that had been selected when the WAIT instruction was executed.

**Table 2.5.3 Port status during wait mode**

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$		Retains status before wait mode	/
RD, WR, BHE, $\overline{WRL}$ , WRH		"H"	
HLDA, BCLK		"H"	
ALE		"H"	
Port		Retains status before wait mode	Retains status before wait mode
CLKOUT	When f <sub>c</sub> selected	Valid only in single-chip mode	Does not stop
	When f <sub>8</sub> , f <sub>32</sub> selected	Valid only in single-chip mode	Does not stop when the WAIT peripheral function clock stop bit is "0". When the WAIT peripheral function clock stop bit is "1", the status immediately prior to entering wait mode is maintained.

### 2.5.7 Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for internal clock  $\phi$ . Table 2.5.4 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

After a reset, operation defaults to division by 8 mode. When shifting to stop mode, the main clock division select bit 0 (bit 6 at address 0006<sub>16</sub>) is set to "1". The following shows the operational modes of internal clock  $\phi$ .

**(1) Division by 2 mode**

The main clock is divided by 2 to obtain the BCLK.

**(2) Division by 4 mode**

The main clock is divided by 4 to obtain the BCLK.

**(3) Division by 8 mode**

The main clock is divided by 8 to obtain the BCLK. Note that oscillation of the main clock must have stabilized before transferring from this mode to another mode.

**(4) Division by 16 mode**

The main clock is divided by 16 to obtain the BCLK.

**(5) No-division mode**

The main clock is used as the BCLK.

**(6) Low-speed mode**

fc is used as the BCLK. Note that oscillation of both the main and sub clocks must have stabilized before transferring from this mode to another or vice versa. At least 2 to 3 seconds are required after the sub clock starts. Therefore, the program must be written to wait until this clock has stabilized immediately after powering up and after stop mode is cancelled.

**(7) Low power dissipation mode**

fc is the BCLK and the main clock is stopped.

**Note:** When switching the count source for BCLK between  $X_{IN}$  and  $X_{CIN}$  it needs that the oscillation of the switched count source is sufficiently stable. Shift after taking the oscillation stabilizing time by software.

**Table 2.5.4 Operating modes dictated by settings of system clock control registers 0 and 1**

CM17	CM16	CM07	CM06	CM05	CM04	Operating mode of BCLK
0	1	0	0	0	Invalid	Division by 2 mode
1	0	0	0	0	Invalid	Division by 4 mode
Invalid	Invalid	0	1	0	Invalid	Division by 8 mode
1	1	0	0	0	Invalid	Division by 16 mode
0	0	0	0	0	Invalid	No-division mode
Invalid	Invalid	1	Invalid	0	1	Low-speed mode
Invalid	Invalid	1	Invalid	1	1	Low power dissipation mode

## 2.5.8 Power Control

The following is a description of the three available power control modes:

### Modes

Power control is available in three modes.

#### (1) Normal operation mode

##### ■ High-speed mode

Divide-by-1 frequency of the main clock becomes the BCLK. The CPU operates with the internal clock selected. Each peripheral function operates according to its assigned clock.

##### ■ Medium-speed mode

Divide-by-2, divide-by-4, divide-by-8, or divide-by-16 frequency of the main clock becomes the BCLK. The CPU operates according to the internal clock selected. Each peripheral function operates according to its assigned clock.

##### ■ Low-speed mode

fc becomes the BCLK. The CPU operates according to the fc clock. The fc clock is supplied by the secondary clock. Each peripheral function operates according to its assigned clock.

##### ■ Low power consumption mode

The main clock operating in low-speed mode is stopped. The CPU operates according to the fc clock. The fc clock is supplied by the secondary clock. The only peripheral functions that operate are those with the sub-clock selected as the count source.

#### (2) Wait mode

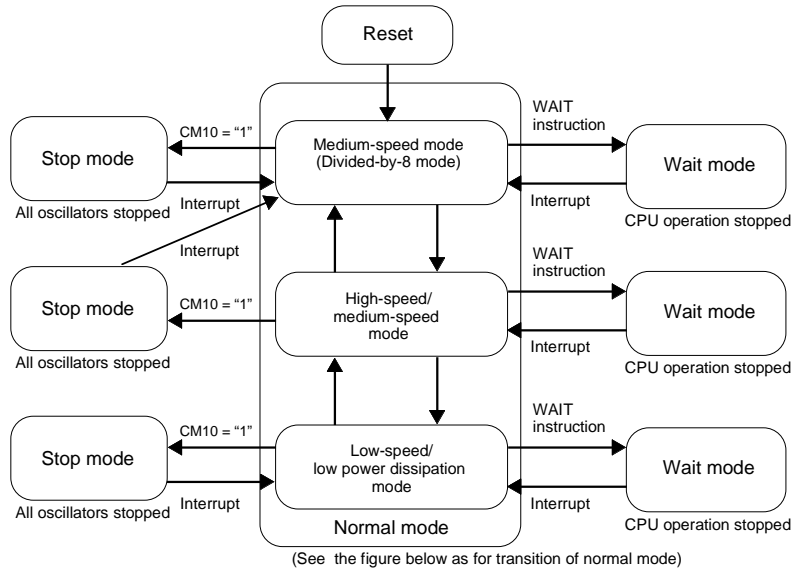
The CPU operation is stopped. The oscillators do not stop.

#### (3) Stop mode

All oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in decreasing power consumption.

Figure 2.5.7 is the state transition diagram of the above modes.

Transition of stop mode, wait mode



Transition of normal mode

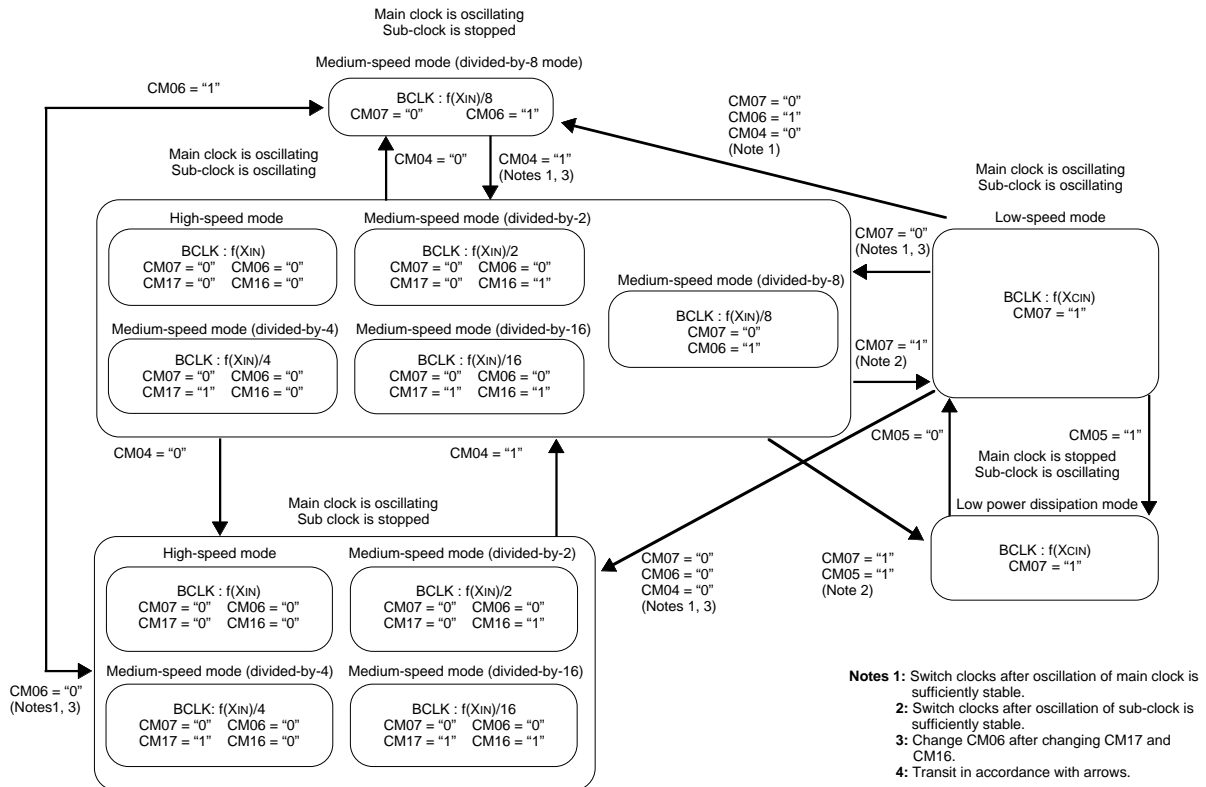


Figure 2.5.7 State transition diagram of Power control mode

### 2.6 Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Figure 2.6.1 shows the protect register. The values in the processor mode register 0 (address 0004<sub>16</sub>), processor mode register 1 (address 0005<sub>16</sub>), system clock control register 0 (address 0006<sub>16</sub>), system clock control register 1 (address 0007<sub>16</sub>) and port P9 direction register (address 03F3<sub>16</sub>) can only be changed when the respective bit in the protect register is set to "1". Therefore, important outputs can be allocated to port P9.

If, after "1" (write-enabled) has been written to the port P9 direction register write-enable bit (bit 2 at address 000A<sub>16</sub>), a value is written to any address, the bit automatically reverts to "0" (write-inhibited). However, the system clock control registers 0 and 1 write-enable bit (bit 0 at 000A<sub>16</sub>) and processor mode register 0 and 1 write-enable bit (bit 1 at 000A<sub>16</sub>) do not automatically return to "0" after a value has been written to an address. The program must therefore be written to return these bits to "0".

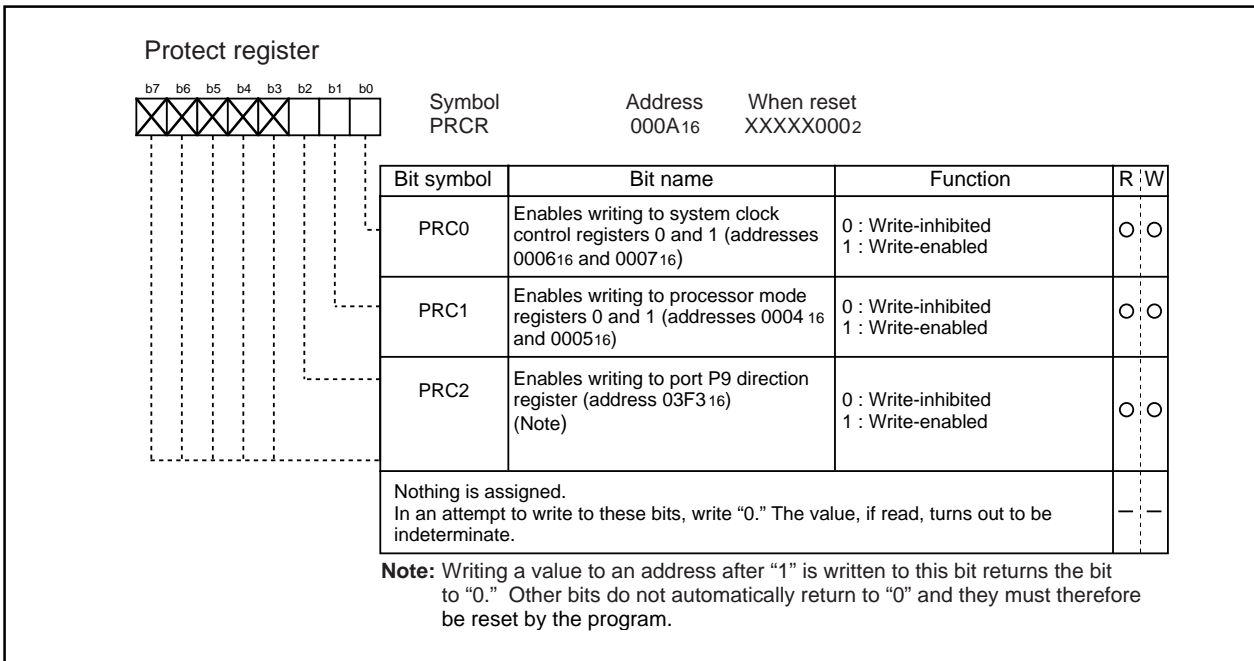
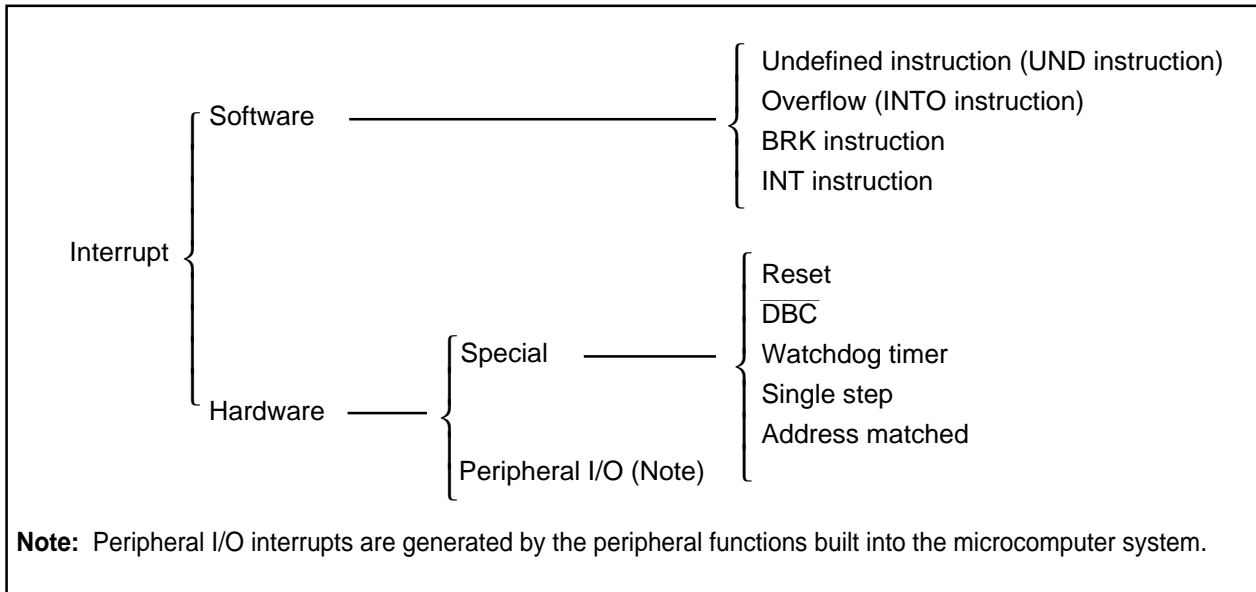


Figure 2.6.1 Protect register

## 2.7 Interrupts

### 2.7.1 Type of Interrupts

Figure 2.7.1 lists the types of interrupts.



**Figure 2.7.1 Classification of interrupts**

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

## 2.7.2 Software Interrupts

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

- **Undefined instruction interrupt**

An undefined instruction interrupt occurs when executing the UND instruction.

- **Overflow interrupt**

An overflow interrupt occurs when executing the INTO instruction with the overflow flag (O flag) set to "1". 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 interrupt**

An INT interrupt occurs when assigning one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 31 are assigned to peripheral I/O interrupts, so executing the INT instruction allows executing the same interrupt routine that a peripheral I/O interrupt does.

The stack pointer (SP) used for the INT interrupt is dependent on which software interrupt number is involved.

So far as software interrupt numbers 0 through 31 are concerned, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. It changes the U flag to "0" and selects the interrupt stack pointer (ISP), and then executes an interrupt sequence. When returning from the interrupt routine, the U flag is returned to the state it was before the acceptance of interrupt request. So far as software numbers 32 through 63 are concerned, the stack pointer does not make a shift.



### 2.7.3 Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral I/O interrupts.

#### (1) Special interrupts

Special interrupts are non-maskable interrupts.

- **Reset**

Reset occurs if an “L” is input to the  $\overline{\text{RESET}}$  pin.

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

This interrupt is exclusively for the debugger, do not use it in other circumstances.

- **Watchdog timer interrupt**

Generated by the watchdog timer.

- **Single-step interrupt**

This interrupt is exclusively for the debugger, do not use it in other circumstances. With the debug flag (D flag) set to “1,” a single-step interrupt occurs after one instruction is executed.

- **Address match interrupt**

An address match interrupt occurs immediately before the instruction held in the address indicated by the address match interrupt register is executed with the address match interrupt enable bit set to “1.” If an address other than the first address of the instruction in the address match interrupt register is set, no address match interrupt occurs. For address match interrupt, see 2.11 Address match Interrupt.

#### (2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. Built-in peripheral functions are dependent on classes of products, so the interrupt factors too are dependent on classes of products. The interrupt vector table is the same as the one for software interrupt numbers 0 through 31 the INI instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **Bus collision detection interrupt**

This is an interrupt that the serial I/O bus collision detection generates.

- **DMA0 interrupt, DMA1 interrupt**

These are interrupts DMA generates.

- **V<sub>SYNC</sub> interrupt**

V<sub>SYNC</sub> interrupt occurs if a V<sub>SYNC</sub> edge is input.

- **A-D conversion interrupt**

This is an interrupt that the A-D converter generates.

- **UART0 transmission, UART2 transmission/NACK interrupts**

These are interrupts that the serial I/O transmission generates.

- **UART0 reception, UART2 reception/ACK interrupts**

These are interrupts that the serial I/O reception generates.

- **Timer A0 interrupt through timer A4 interrupt**

These are interrupts that timer A generates

- **Timer B0 interrupt through timer B2 interrupt**

These are interrupts that timer B generates.

- **$\overline{\text{INT0}}$  interrupt and  $\overline{\text{INT1}}$  interrupt**

An  $\overline{\text{INT}}$  interrupt occurs if either a rising edge or a falling edge or a both edge is input to the  $\overline{\text{INT}}$  pin.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

---

- **OSD1 interrupt and OSD2 interrupt**

These are interrupts that OSD display is completed.

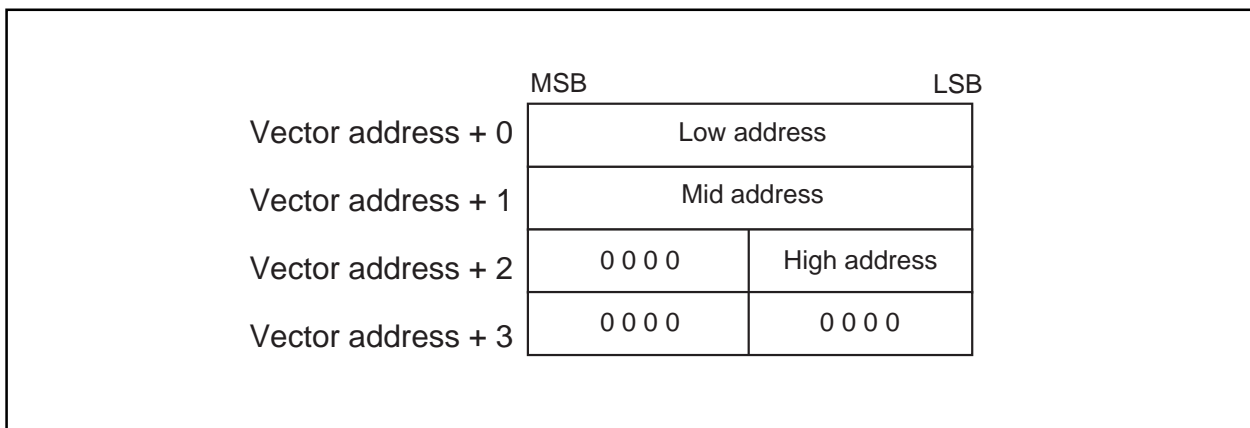
- **Data slicer interrupt**

This is an interrupt that data slicer circuit requests.

### 2.7.4 Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Figure 2.7.2 shows the format for specifying the address.

Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.



**Figure 2.7.2 Format for specifying interrupt vector addresses**

#### (1) Fixed vector tables

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFDC<sub>16</sub> to FFFFF<sub>16</sub>. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 2.7.1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

**Table 2.7.1 Interrupts assigned to the fixed vector tables and addresses of vector tables**

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFD <sub>16</sub> to FFFD <sub>16</sub>	Interrupt on UND instruction
Overflow	FFFE0 <sub>16</sub> to FFFE3 <sub>16</sub>	Interrupt on INTO instruction
BRK instruction	FFFE4 <sub>16</sub> to FFFE7 <sub>16</sub>	If the vector is filled with FF <sub>16</sub> , program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE8 <sub>16</sub> to FFE <sub>16</sub>	There is an address-matching interrupt enable bit
Single step (Note)	FFFE <sub>16</sub> to FFE <sub>16</sub>	Do not use
Watchdog timer	FFFF0 <sub>16</sub> to FFFF3 <sub>16</sub>	
DBC (Note)	FFFF4 <sub>16</sub> to FFFF7 <sub>16</sub>	Do not use
Reserved source	FFFE8 <sub>16</sub> to FFE <sub>16</sub>	Do not use
Reset	FFFF <sub>16</sub> to FFFF <sub>16</sub>	

**Note:** Interrupts used for debugging purposes only.

**(2) Variable vector tables**

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFDC<sub>16</sub> to FFFFF<sub>16</sub>. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 2.7.2 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

**Table 2.7.2 Interrupts assigned to the variable vector tables and addresses of vector tables**

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note 1)	BRK instruction	Cannot be masked I flag
Software interrupt number 4	+16 to +19 (Note 1)	OSD1	
Software interrupt number 5	+20 to +23 (Note 1)	Reserved source	
Software interrupt number 6	+24 to +27 (Note 1)	Reserved source	
Software interrupt number 7	+28 to +31 (Note 1)	Reserved source	
Software interrupt number 8	+32 to +35 (Note 1)	OSD2	
Software interrupt number 9	+36 to +39 (Note 1)	Reserved source	
Software interrupt number 10	+40 to +43 (Note 1)	Bus collision detection	
Software interrupt number 11	+44 to +47 (Note 1)	DMA0	
Software interrupt number 12	+48 to +51 (Note 1)	DMA1	
Software interrupt number 13	+52 to +55 (Note 1)	Reserved source	
Software interrupt number 14	+56 to +59 (Note 1)	A-D conversion	
Software interrupt number 15	+60 to +63 (Note 1)	UART2 transmit/NACK (Note 2)	
Software interrupt number 16	+64 to +67 (Note 1)	UART2 receive/ACK (Note 2)	
Software interrupt number 17	+68 to +71 (Note 1)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note 1)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note 1)	Data slicer	
Software interrupt number 20	+80 to +83 (Note 1)	VSYNC	
Software interrupt number 21	+84 to +87 (Note 1)	Timer A0	
Software interrupt number 22	+88 to +91 (Note 1)	Timer A1	
Software interrupt number 23	+92 to +95 (Note 1)	Timer A2	
Software interrupt number 24	+96 to +99 (Note 1)	Timer A3	
Software interrupt number 25	+100 to +103 (Note 1)	Timer A4	
Software interrupt number 26	+104 to +107 (Note 1)	Timer B0	
Software interrupt number 27	+108 to +111 (Note 1)	Timer B1	
Software interrupt number 28	+112 to +115 (Note 1)	Timer B2	
Software interrupt number 29	+116 to +119 (Note 1)	INT0	
Software interrupt number 30	+120 to +123 (Note 1)	INT1	
Software interrupt number 31	+124 to +127 (Note 1)	Reserved source	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note 1) to +252 to +255 (Note 1)	Software interrupt	Cannot be masked I flag

**Notes 1:** Address relative to address in interrupt table register (INTB).

**2:** When I<sup>2</sup>C mode is selected, NACK and ACK interrupts are selected.

### 2.7.5 Interrupt Control

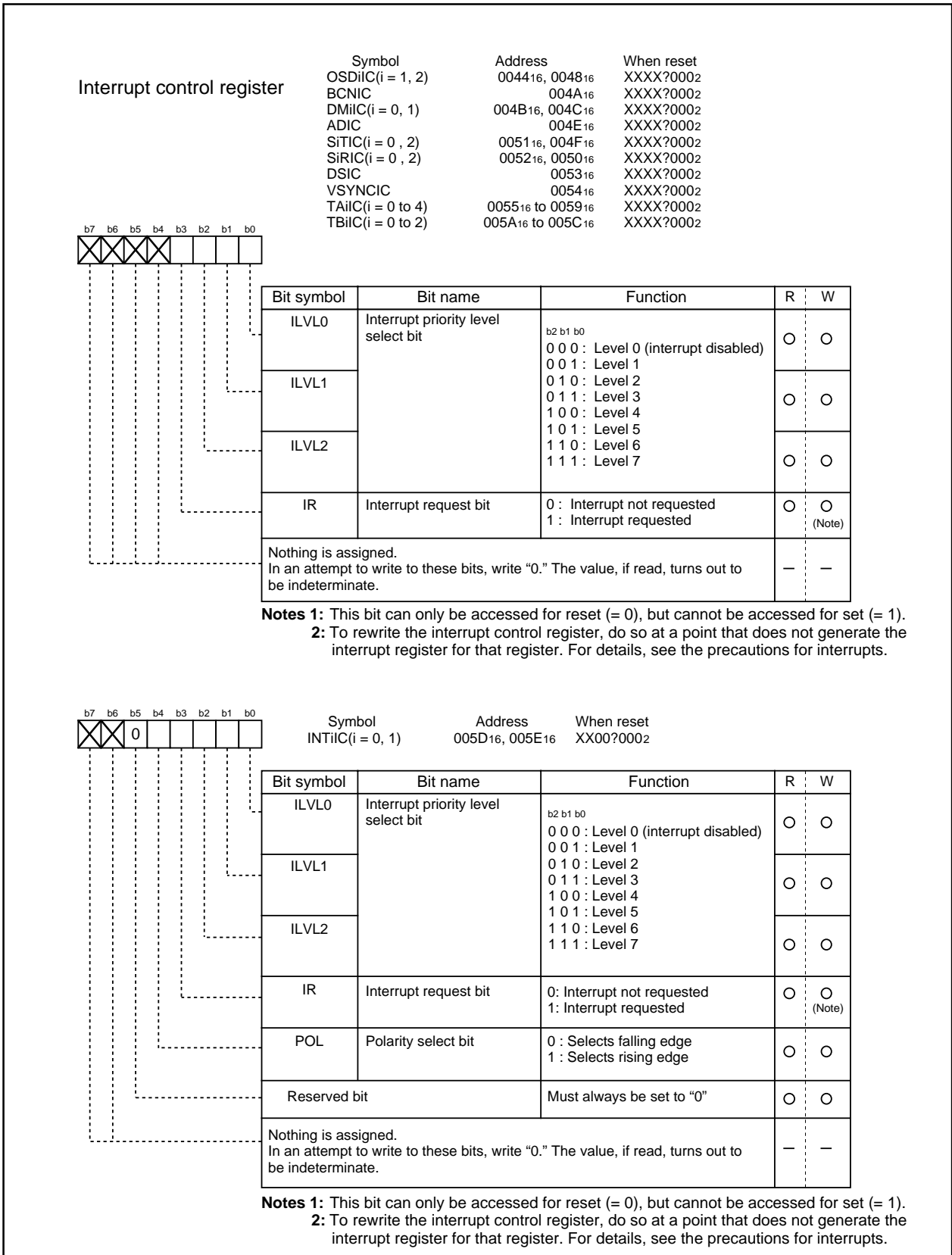
Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

Enable or disable a non-maskable interrupt using the interrupt enable flag (I flag), interrupt priority level selection bit, or processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt priority level selection bit are located in the interrupt control register of each interrupt. Also, the interrupt enable flag (I flag) and the IPL are located in the flag register (FLG).

Figure 2.7.3 shows the interrupt control registers.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.7.3 Interrupt control registers**

### 2.7.6 Interrupt Enable Flag (I flag)

The interrupt enable flag (I flag) controls the enabling and disabling of maskable interrupts. Setting this flag to "1" enables all maskable interrupts; setting it to "0" disables all maskable interrupts. This flag is set to "0" after reset.

### 2.7.7 Interrupt Request Bit

The interrupt request bit is set to "1" by hardware when an interrupt is requested. After the interrupt is accepted and jumps to the corresponding interrupt vector, the request bit is set to "0" by hardware. The interrupt request bit can also be set to "0" by software. (Do not set this bit to "1").

### 2.7.8 Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to "0" disables the interrupt.


Table 2.7.3 shows the settings of interrupt priority levels and Table 2.7.4 shows the interrupt levels enabled, according to the consist of the IPL.

The following are conditions under which an interrupt is accepted:

- interrupt enable flag (I flag) = 1
- interrupt request bit = 1
- interrupt priority level > IPL

The interrupt enable flag (I flag), the interrupt request bit, the interrupt priority select bit, and the IPL are independent, and they are not affected by one another.

**Table 2.7.3 Settings of interrupt priority levels**

Interrupt priority level select bit	Interrupt priority level	Priority order
b2 b1 b0 0 0 0	Level 0 (interrupt disabled)	_____
0 0 1	Level 1	Low  High
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	

**Table 2.7.4 Interrupt levels enabled according to the contents of the IPL**

IPL	Enabled interrupt priority levels
IPL <sub>2</sub> IPL <sub>1</sub> IPL <sub>0</sub> 0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

## 2.7.9 Rewrite Interrupt Control Register

To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

### Example 1:

```
INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  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 ; Clear TA0IC int. priority level and int. request bit.
  MOV.W MEM, R0     ; Dummy read.
  FSET  I           ; Enable interrupts.
```

### Example 3:

```
INT_SWITCH3:
  PUSHC FLG        ; Push Flag register onto stack
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  POPC  FLG        ; Enable interrupts.
```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET



### 2.7.10 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.

In the interrupt sequence, the processor carries out the following in sequence given:

- (1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000<sub>16</sub>.
- (2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
- (3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to "0" (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)
- (4) Saves the content of the temporary register (Note 1) within the CPU in the stack area.
- (5) Saves the content of the program counter (PC) in the stack area.
- (6) Sets the interrupt priority level of the accepted instruction in the IPL.

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

**Note:** This register cannot be utilized by the user.

### 2.7.11 Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure 2.7.4 shows the interrupt response time.

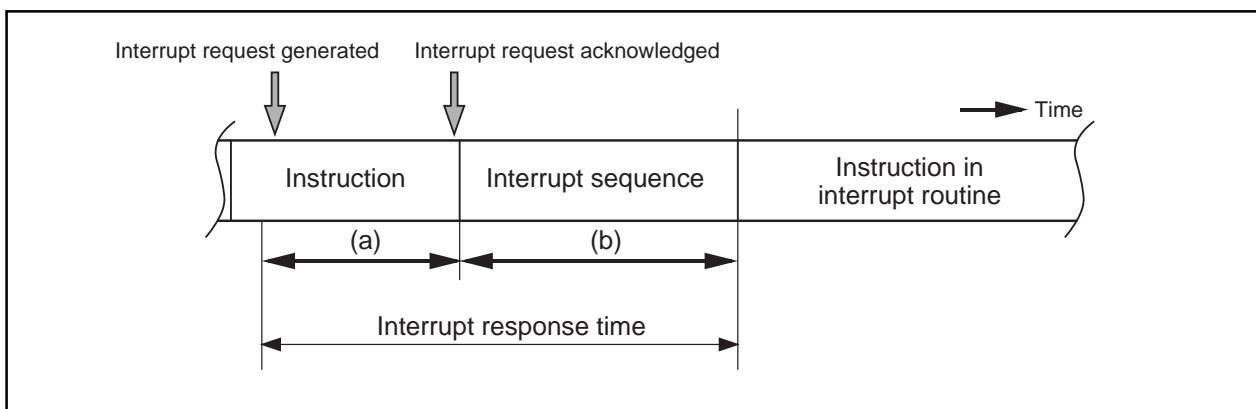


Figure 2.7.4 Interrupt response time

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

Time (a) is dependent on the instruction under execution. Thirty cycles is the maximum required for the DIVX instruction (without wait).

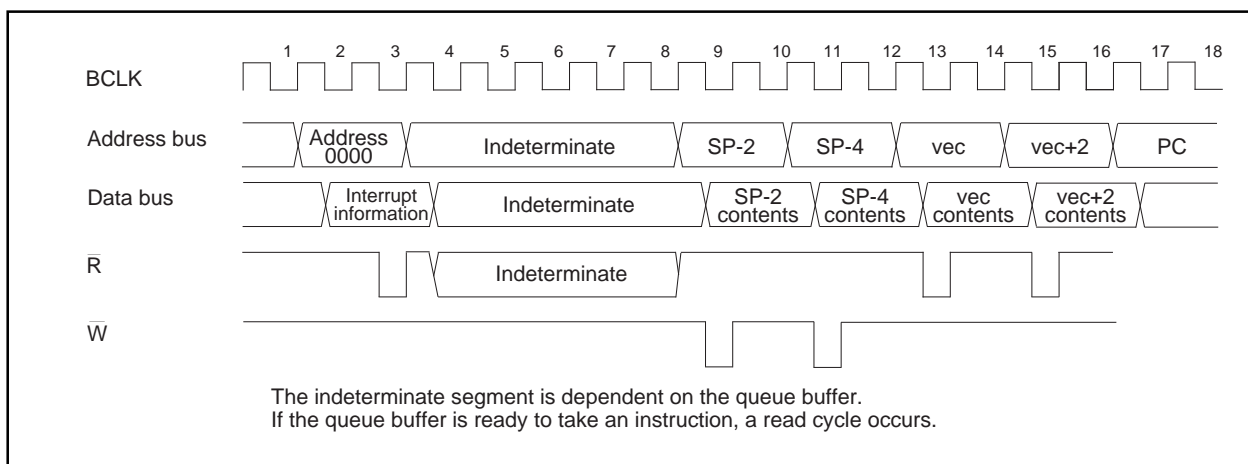
Time (b) is as shown in Table 2.7.5.

**Table 2.7.5 Time required for executing the interrupt sequence**

Interrupt vector address	Stack pointer (SP) value	16-Bit bus, without wait	8-Bit bus, without wait
Even	Even	18 cycles (Note 1)	20 cycles (Note 1)
Even	Odd	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Even	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Odd	20 cycles (Note 1)	20 cycles (Note 1)

**Notes 1:** Add 2 cycles in the case of a DBC interrupt; add 1 cycle in the case either of an address coincidence interrupt or of a single-step interrupt.

**2:** Locate an interrupt vector address in an even address, if possible.



**Figure 2.7.5 Time required for executing the interrupt sequence**

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

If an interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.

If an interrupt request, that does not have an interrupt priority level, is accepted, one of the values shown in Table 2.7.6 is set in the IPL.

**Table 2.7.6 Relationship between interrupts without interrupt priority levels and IPL**

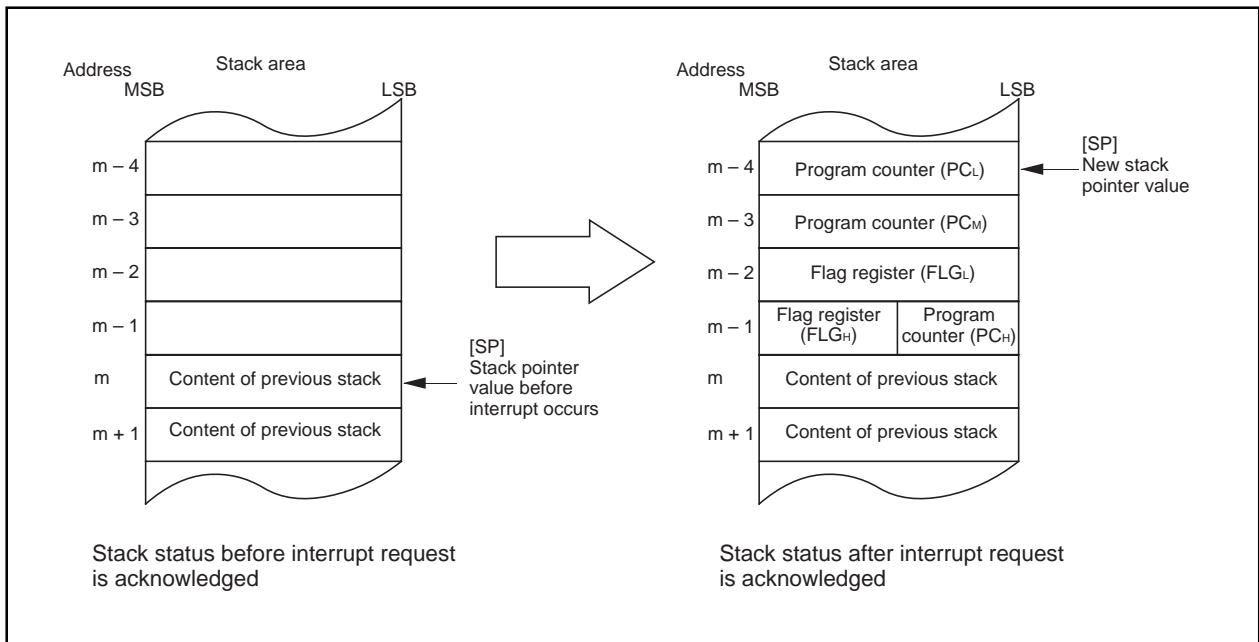
Interrupt sources without priority levels	Value set in the IPL
Watchdog timer	7
Reset	0
Other	Not changed

### 2.7.13 Saving Registers

In the interrupt sequence, only the contents of the flag register (FLG) and that of the program counter (PC) are saved in the stack area.

First, the processor saves the four higher-order bits of the program counter, and 4 upper-order bits and 8 lower-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 lower-order bits of the program counter. Figure 2.7.6 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

Save other necessary registers at the beginning of the interrupt routine using software. Using the PUSHM instruction alone can save all the registers except the stack pointer (SP).



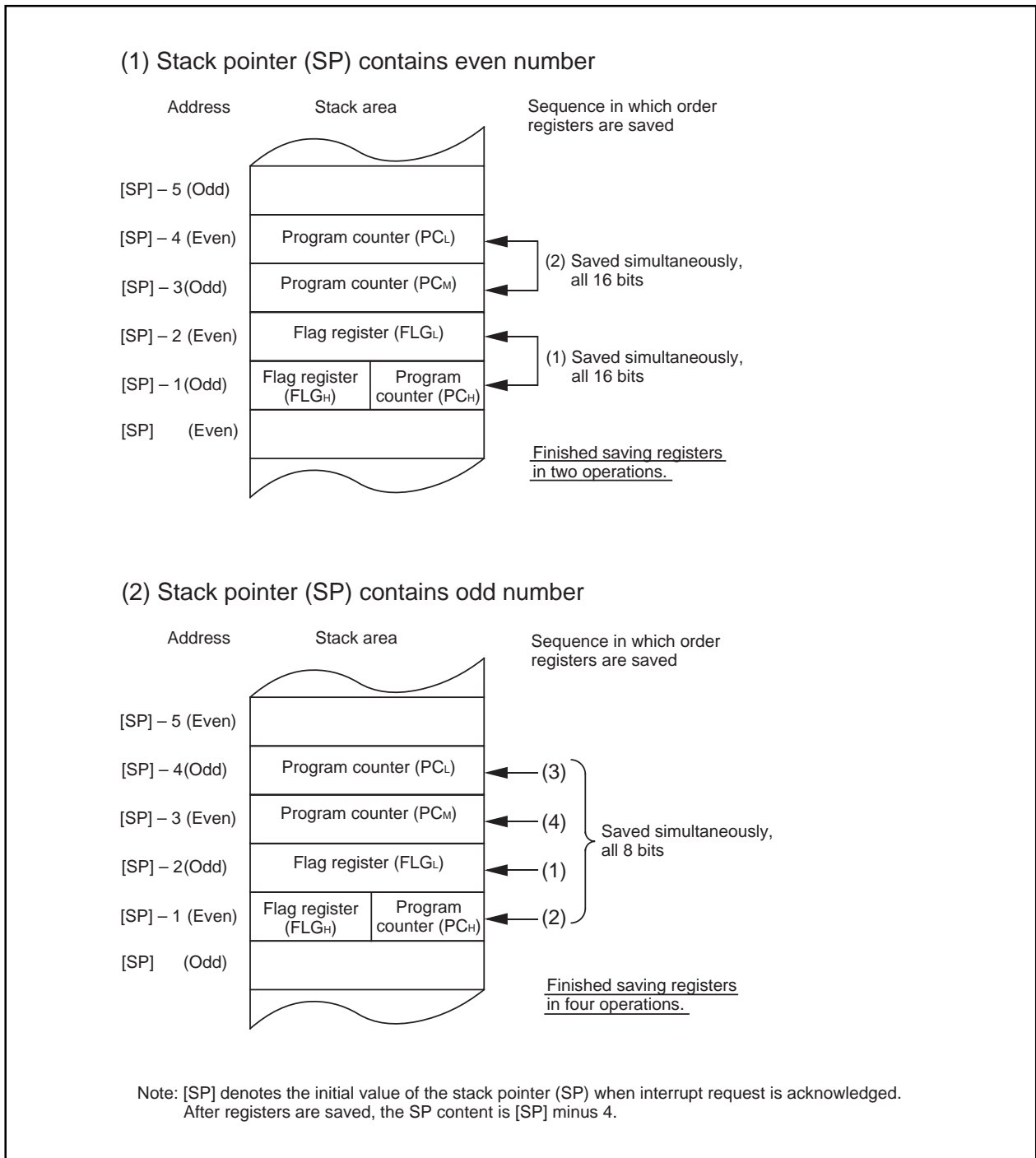
**Figure 2.7.6 State of stack before and after acceptance of interrupt request**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer, at the time of acceptance of an interrupt request, is even or odd. If the content of the stack pointer (Note) is even, the content of the flag register (FLG) and the content of the program counter (PC) are saved, 16 bits at a time. If odd, their contents are saved in two steps, 8 bits at a time. Figure 2.7.7 shows the operation of the saving registers.

**Note:** Stack pointer indicated by U flag.



**Figure 2.7.7 Operation of saving registers**

### 2.7.14 Returning from an Interrupt Routine

Executing the REIT instruction at the end of an interrupt routine returns the contents of the flag register (FLG) as it was immediately before the start of interrupt sequence and the contents of the program counter (PC), both of which have been saved in the stack area. Then control returns to the program that was being executed before the acceptance of the interrupt request, so that the suspended process resumes.

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

### 2.7.15 Interrupt Priority

If there are two or more interrupt requests occurring at a point in time within a single sampling (checking whether interrupt requests are made), the interrupt assigned a higher priority is accepted.

Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bit. If the same interrupt priority level is assigned, however, the interrupt assigned a higher hardware priority is accepted.

Priorities of the special interrupts, such as Reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are regulated by hardware.

Figure 2.7.8 shows the priorities of hardware interrupts.

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

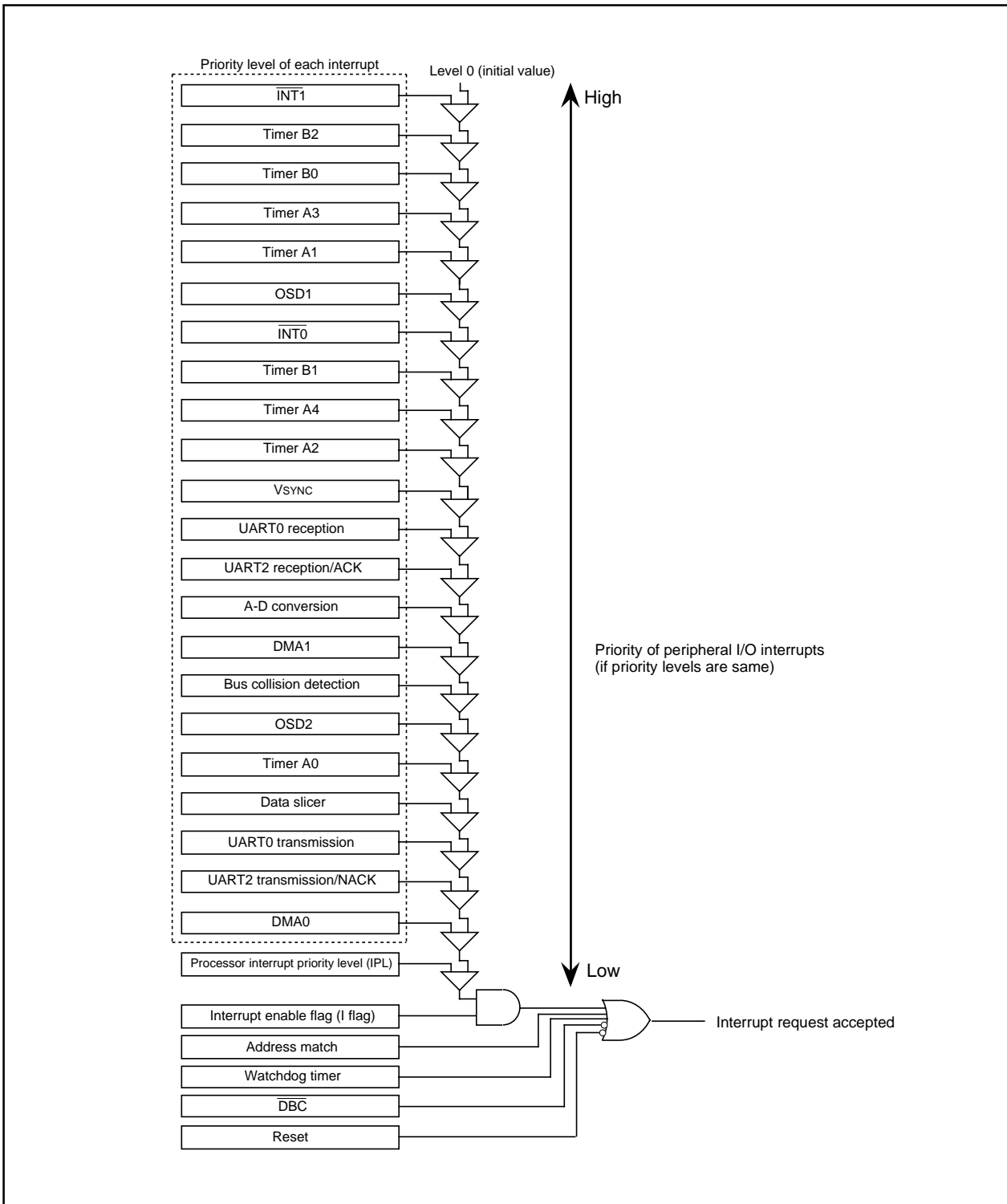
### 2.7.16 Interrupt priority level resolution circuit

When two or more interrupts are generated simultaneously, this circuit selects the interrupt with the highest priority level.

Figure 2.7.9 shows the circuit that judges the interrupt priority level.

Reset >  $\overline{DBC}$  > Watchdog timer > Peripheral I/O > Single step > Address match

**Figure 2.7.8 Hardware interrupts priorities**



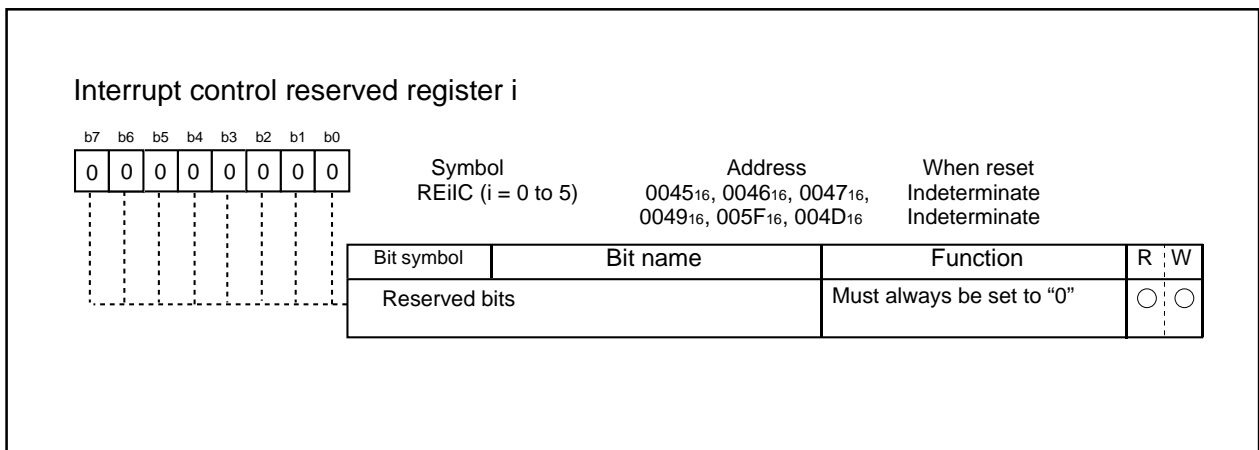
**Figure 2.7.9 Maskable interrupts priorities (peripheral I/O interrupts)**

**2.7.17 INT Interrupt**

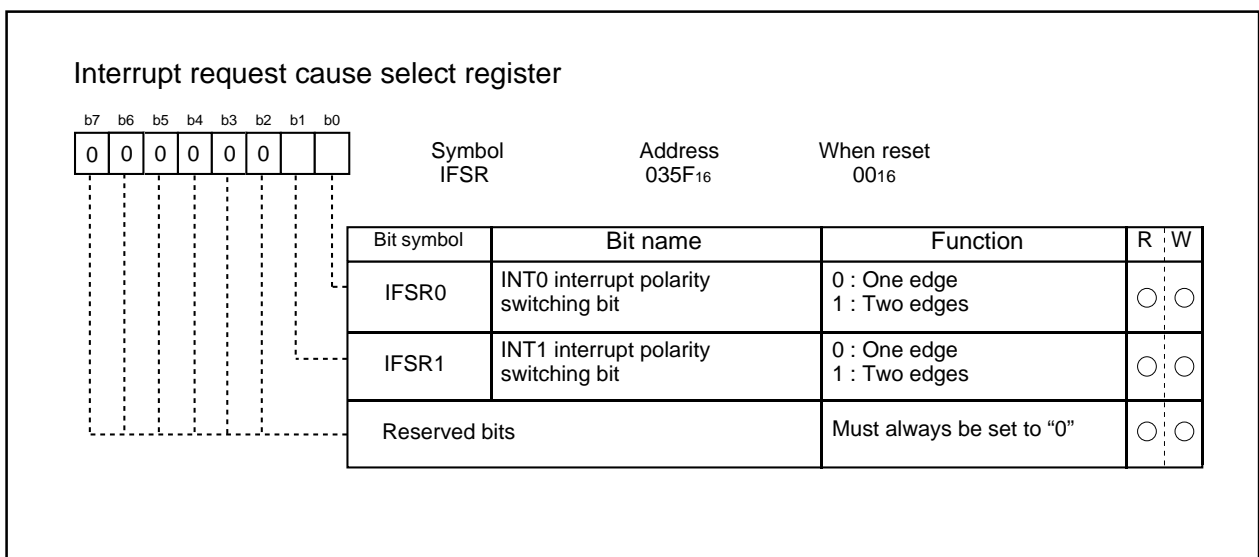
INT0 and INT1 are triggered by the edges of external inputs. The edge polarity is selected using the polarity select bit.

As for external interrupt input, an interrupt can be generated both at the rising edge and at the falling edge by setting "1" in the INTi interrupt polarity switching bit of the interrupt request cause select register (035F<sub>16</sub>). To select both edges, set the polarity switching bit of the corresponding interrupt control register to 'falling edge' ("0").

Figure 2.7.10 shows the Interrupt control reserved register, Figure 2.7.11 shows the Interrupt request cause select register.



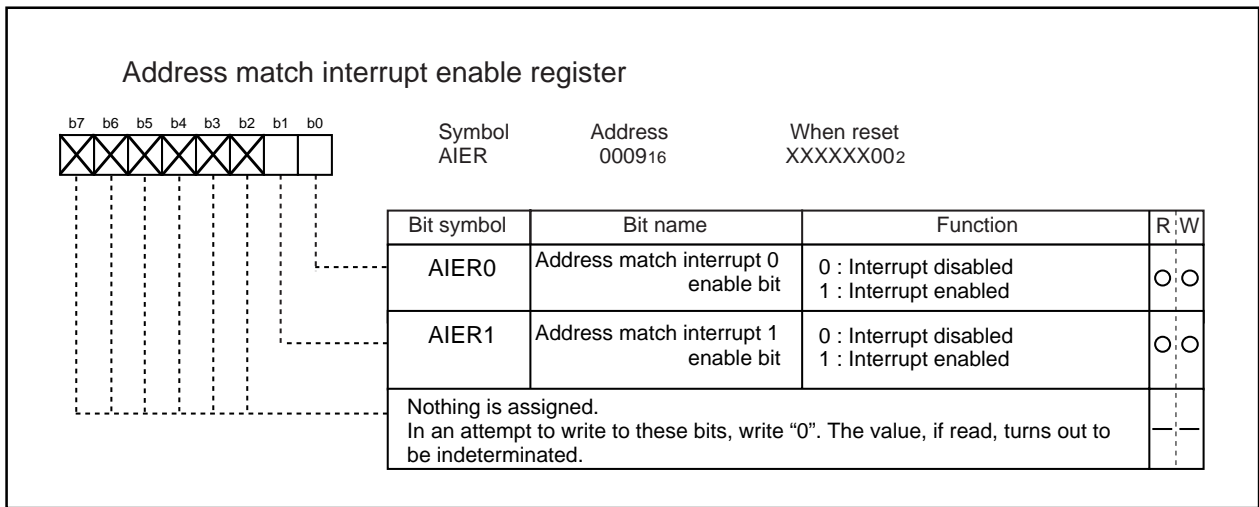
**Figure 2.7.10** Interrupt control reserved register i



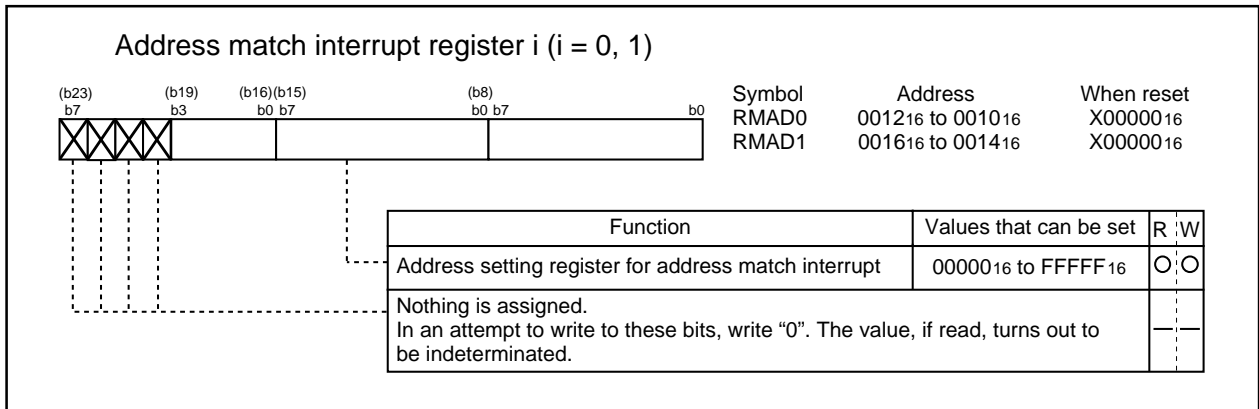
**Figure 2.7.11** Interrupt request cause select register

**2.7.18 Address Match Interrupt**

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Two address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL). The value of the program counter (PC) for an address match interrupt varies depending on the instruction being executed. Figures 2.7.12 and 2.7.13 show the address match interrupt-related registers.



**Figure 2.7.12 Address match interrupt enable register**



**Figure 2.7.13 Address match interrupt register i (i = 0, 1)**



## 2.7.19 Precautions for Interrupts

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

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.

The interrupt request bit of the certain interrupt written in address 00000<sub>16</sub> will then be set to "0".

Reading address 00000<sub>16</sub> by software sets enabled highest priority interrupt source request bit to "0".

Though the interrupt is generated, the interrupt routine may not be executed.

Do not read address 00000<sub>16</sub> by software.

### (2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000<sub>16</sub>. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt.

### (3) External interrupt

- Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to pins  $\overline{\text{INT}}_0$  and  $\overline{\text{INT}}_1$  regardless of the CPU operation clock.
- When the polarity of the  $\overline{\text{INT}}_0$  and  $\overline{\text{INT}}_1$  pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure 2.7.14 shows the procedure for changing the  $\overline{\text{INT}}$  interrupt generate factor.

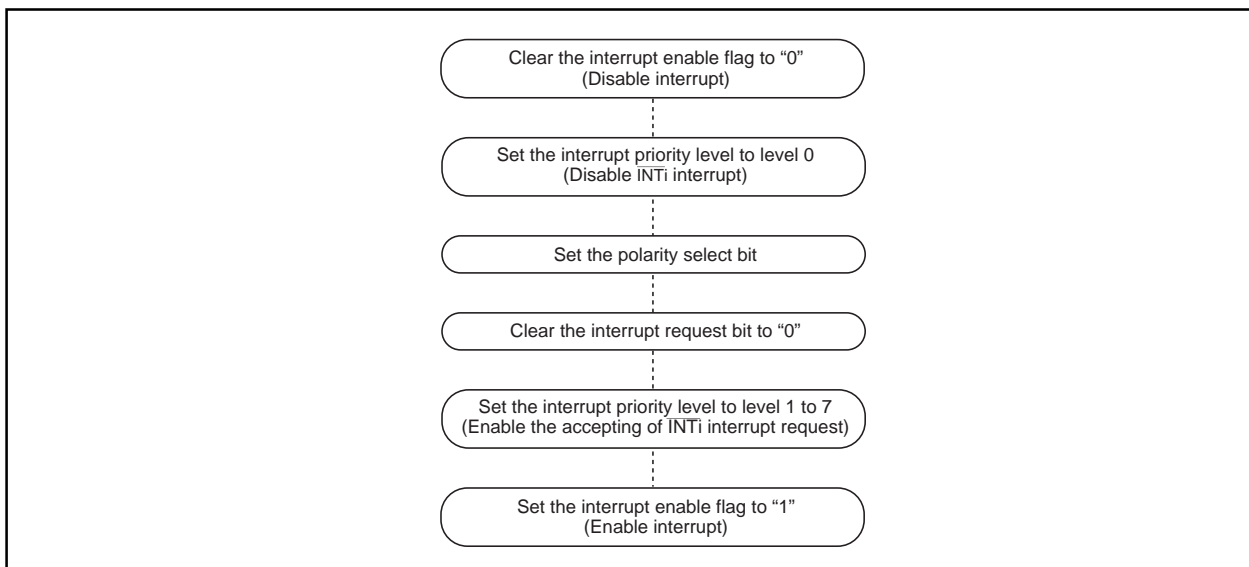


Figure 2.7.14 Switching condition of INT interrupt request

**(5) Rewrite interrupt control register**

- To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

**Example 1:**

```

INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  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 ; Clear TA0IC int. priority level and int. request bit.
  MOV.W MEM, R0    ; Dummy read.
  FSET  I           ; Enable interrupts.

```

**Example 3:**

```

INT_SWITCH3:
  PUSHC FLG       ; Push Flag register onto stack
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  POPC  FLG       ; Enable interrupts.

```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

- When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

## 2.8 Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. A watchdog timer interrupt is generated when an underflow occurs in the watchdog timer. When X<sub>IN</sub> is selected for the BCLK, bit 7 of the watchdog timer control register (address 000F<sub>16</sub>) selects the prescaler division ratio (by 16 or by 128). When X<sub>CIN</sub> is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address 000F<sub>16</sub>). Thus the watchdog timer's period can be calculated as given below. The watchdog timer's period is, however, subject to an error due to the pre-scaler.

### With X<sub>IN</sub> chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{pre-scaler dividing ratio (16 or 128)} \times \text{watchdog timer count (32768)}}{\text{BCLK}}$$

### With X<sub>CIN</sub> chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{pre-scaler dividing ratio (2)} \times \text{watchdog timer count (32768)}}{\text{BCLK}}$$

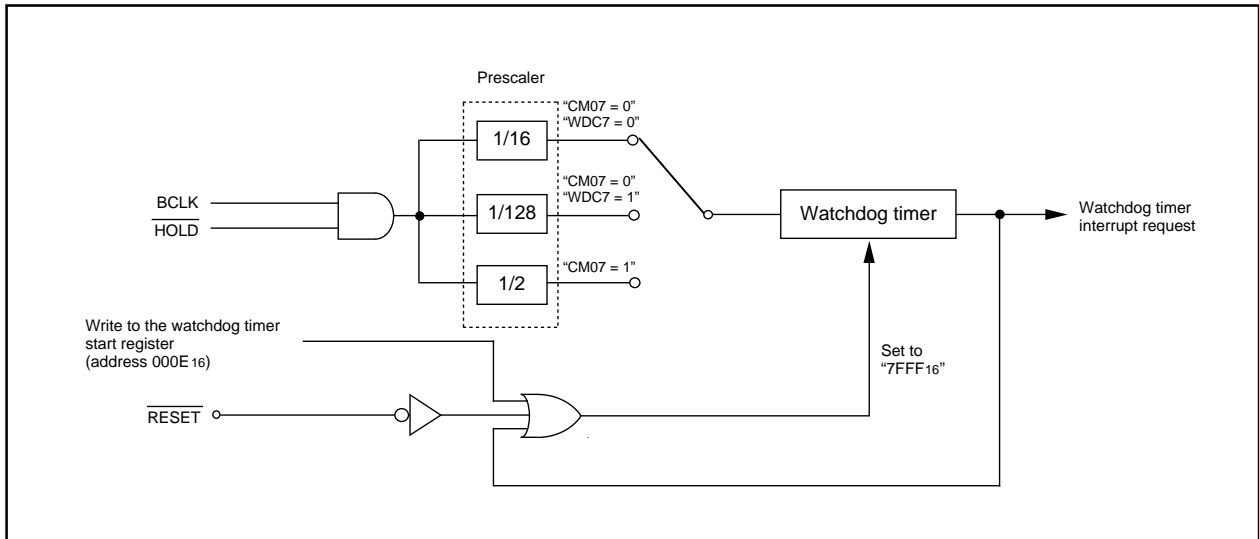
For example suppose that BCLK runs at 10 MHz and that 16 has been chosen for the dividing ratio of the pre-scaler, then the watchdog timer's period becomes approximately 52.4 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address 000E<sub>16</sub>) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address 000E<sub>16</sub>).

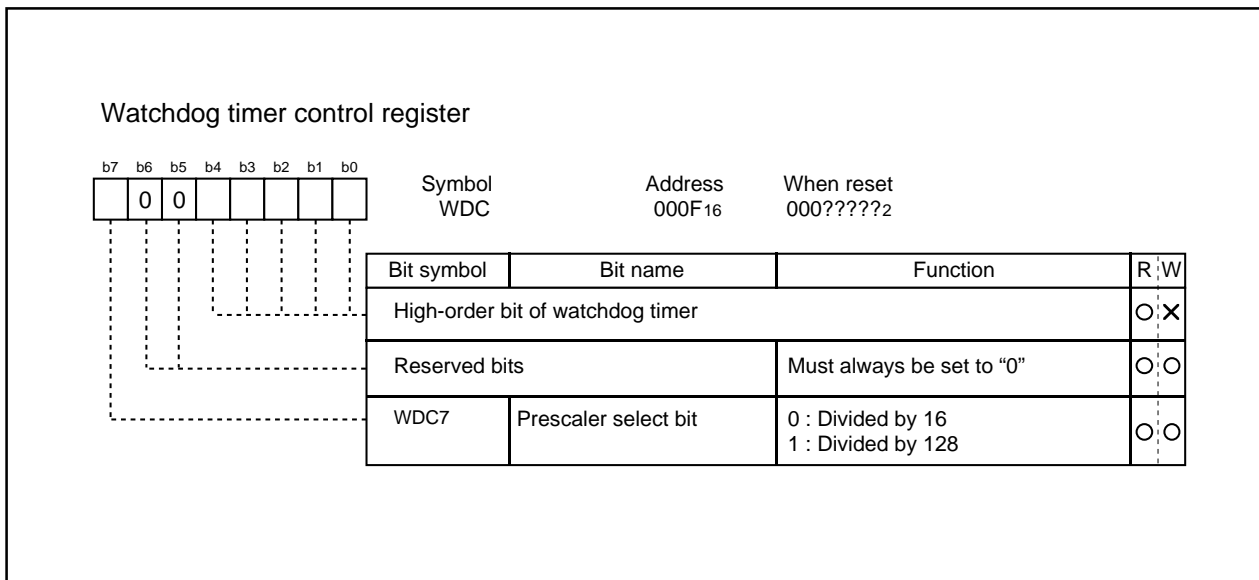
Figure 2.8.1 shows the block diagram of the watchdog timer. Figure 2.8.2 shows the watchdog timer control register and Figure 2.8.3 shows the watchdog timer start register.

**M306V0ME-XXXFP**  
**M306V0EEFP**

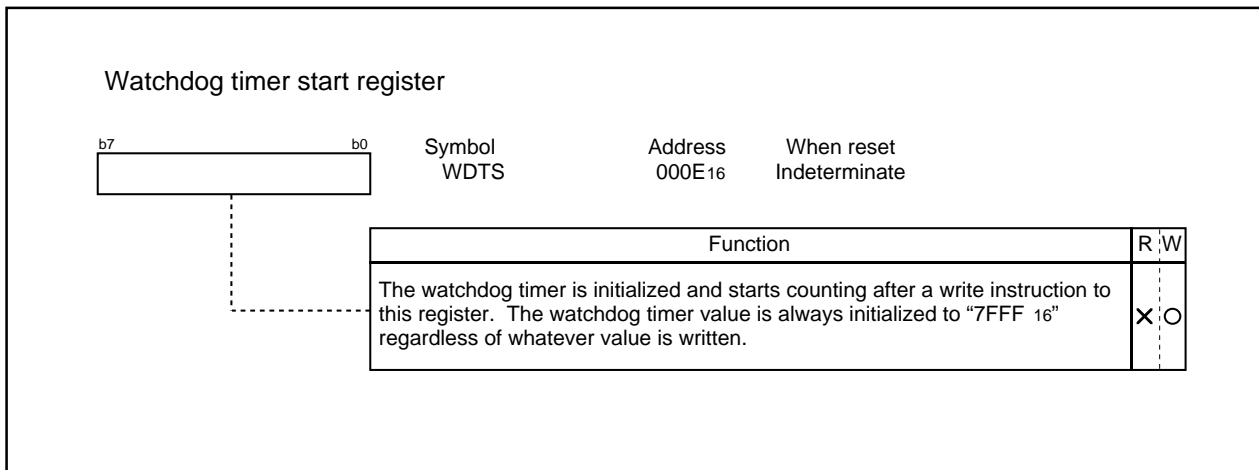
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.8.1 Block diagram of watchdog timer**



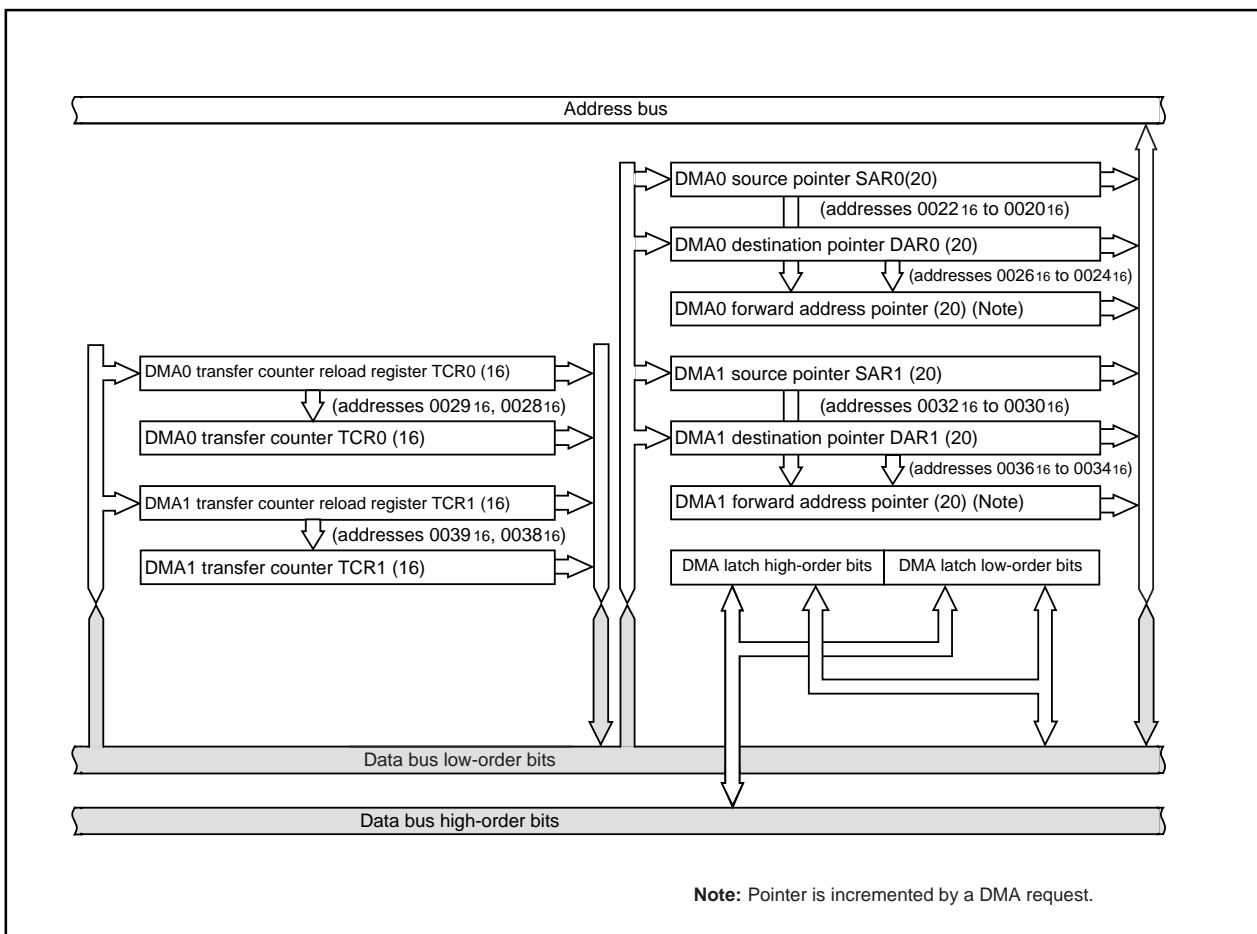
**Figure 2.8.2 Watchdog timer control register**



**Figure 2.8.3 Watchdog timer start register**

## 2.9 DMAC

This microcomputer has two DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC shares the same data bus with the CPU. The DMAC is given a higher right of using the bus than the CPU, which leads to working the cycle stealing method. On this account, the operation from the occurrence of DMA transfer request signal to the completion of 1-word (16-bit) or 1-byte (8-bit) data transfer can be performed at high speed. Figure 2.9.1 shows the block diagram of the DMAC. Table 2.9.1 shows the DMAC specifications. Figures 2.9.2 to 2.9.7 show the registers used by the DMAC.



**Figure 2.9.1 Block diagram of DMAC**

Either a write signal to the software DMA request bit or an interrupt request signal is used as a DMA transfer request signal. But the DMA transfer is affected neither by the interrupt enable flag (I flag) nor by the interrupt priority level. The DMA transfer doesn't affect any interrupts either.

If the DMAC is active (the DMA enable bit is set to 1), data transfer starts every time a DMA transfer request signal occurs. If the cycle of the occurrences of DMA transfer request signals is higher than the DMA transfer cycle, there can be instances in which the number of transfer requests doesn't agree with the number of transfers. For details, see the description of the DMA request bit.

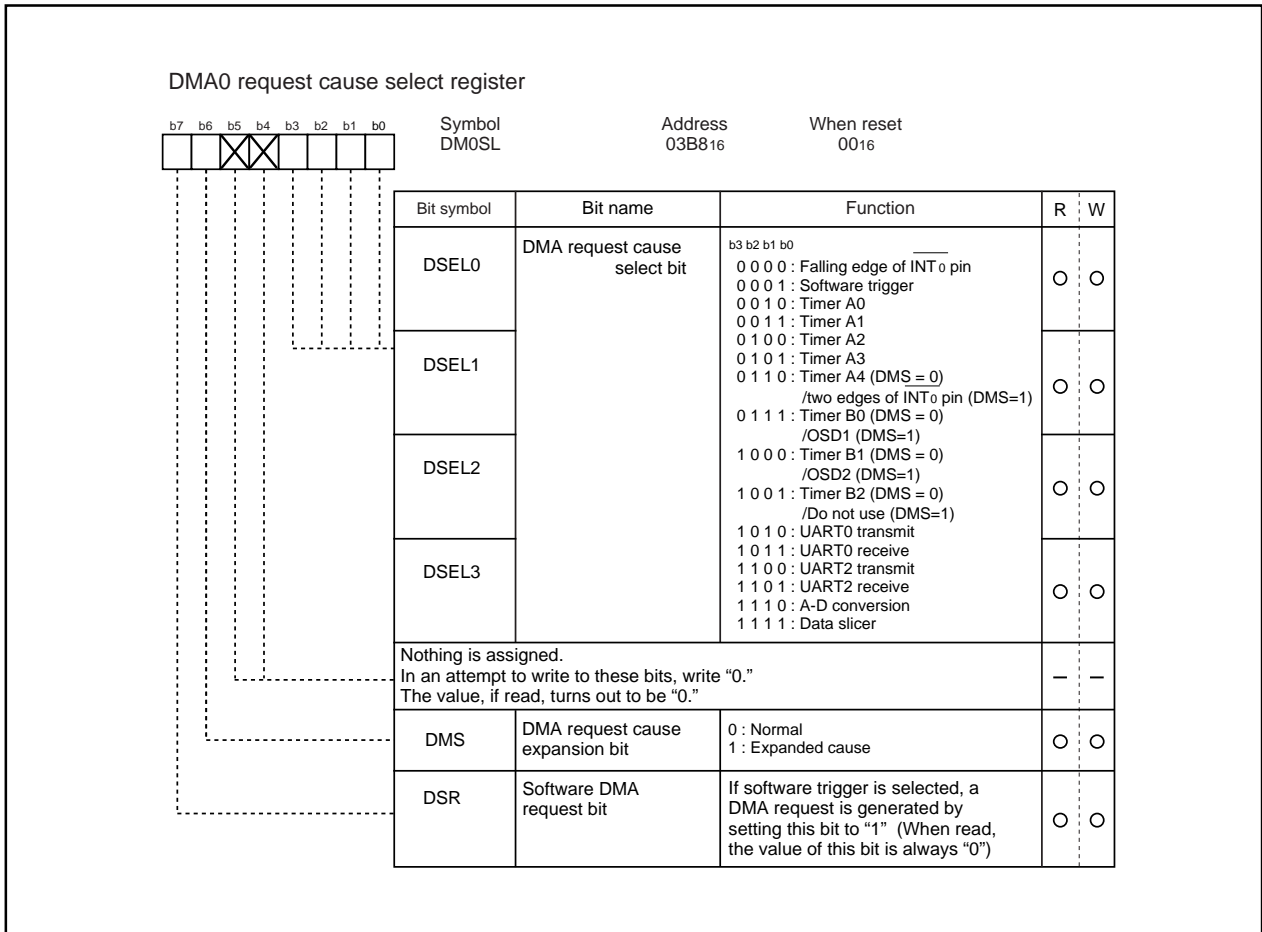
Table 2.9.1 DMAC specifications

Item	Specification
No. of channels	2 (cycle steal method)
Transfer memory space	<ul style="list-style-type: none"> <li>• From any address in the 1M bytes space to a fixed address</li> <li>• From a fixed address to any address in the 1M bytes space</li> <li>• From a fixed address to a fixed address</li> </ul> (Note that DMA-related registers [0020 <sub>16</sub> to 003F <sub>16</sub> ] cannot be accessed)
Maximum No. of bytes transferred	128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (Note)	Falling edge or both edge of pin $\overline{INT_0}$ Falling edge of pin $\overline{INT_1}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transmission and reception interrupt requests UART2 transmission and reception interrupt requests A-D conversion interrupt requests OSD1 and OSD2 interrupt requests Data slicer interrupt request V <sub>SYNC</sub> interrupt request Software triggers
Channel priority	DMA0 takes precedence if DMA0 and DMA1 requests are generated simultaneously
Transfer unit	8 bits or 16 bits
Transfer address direction	forward/fixed (forward direction cannot be specified for both source and destination simultaneously)
Transfer mode	<ul style="list-style-type: none"> <li>• Single transfer mode After the transfer counter underflows, the DMA enable bit turns to "0", and the DMAC turns inactive</li> <li>• Repeat transfer mode After the transfer counter underflows, the value of the transfer counter reload register is reloaded to the transfer counter. The DMAC remains active unless a "0" is written to the DMA enable bit.</li> </ul>
DMA interrupt request generation timing	When an underflow occurs in the transfer counter
Active	When the DMA enable bit is set to "1", the DMAC is active. When the DMAC is active, data transfer starts every time a DMA transfer request signal occurs.
Inactive	<ul style="list-style-type: none"> <li>• When the DMA enable bit is set to "0", the DMAC is inactive.</li> <li>• After the transfer counter underflows in single transfer mode</li> </ul>
Forward address pointer and reload timing for transfer counter	At the time of starting data transfer immediately after turning the DMAC active, the value of one of source pointer and destination pointer - the one specified for the forward direction - is reloaded to the forward direction address pointer, and the value of the transfer counter reload register is reloaded to the transfer counter.
Writing to register	Registers specified for forward direction transfer are always write enabled. Registers specified for fixed address transfer are write-enabled when the DMA enable bit is "0".
Reading the register	Can be read at any time. However, when the DMA enable bit is "1", reading the register set up as the forward register is the same as reading the value of the forward address pointer.

**Note:** DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the interrupt enable flag (I flag) nor by the interrupt priority level.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.9.2 DMA0 request cause select register**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

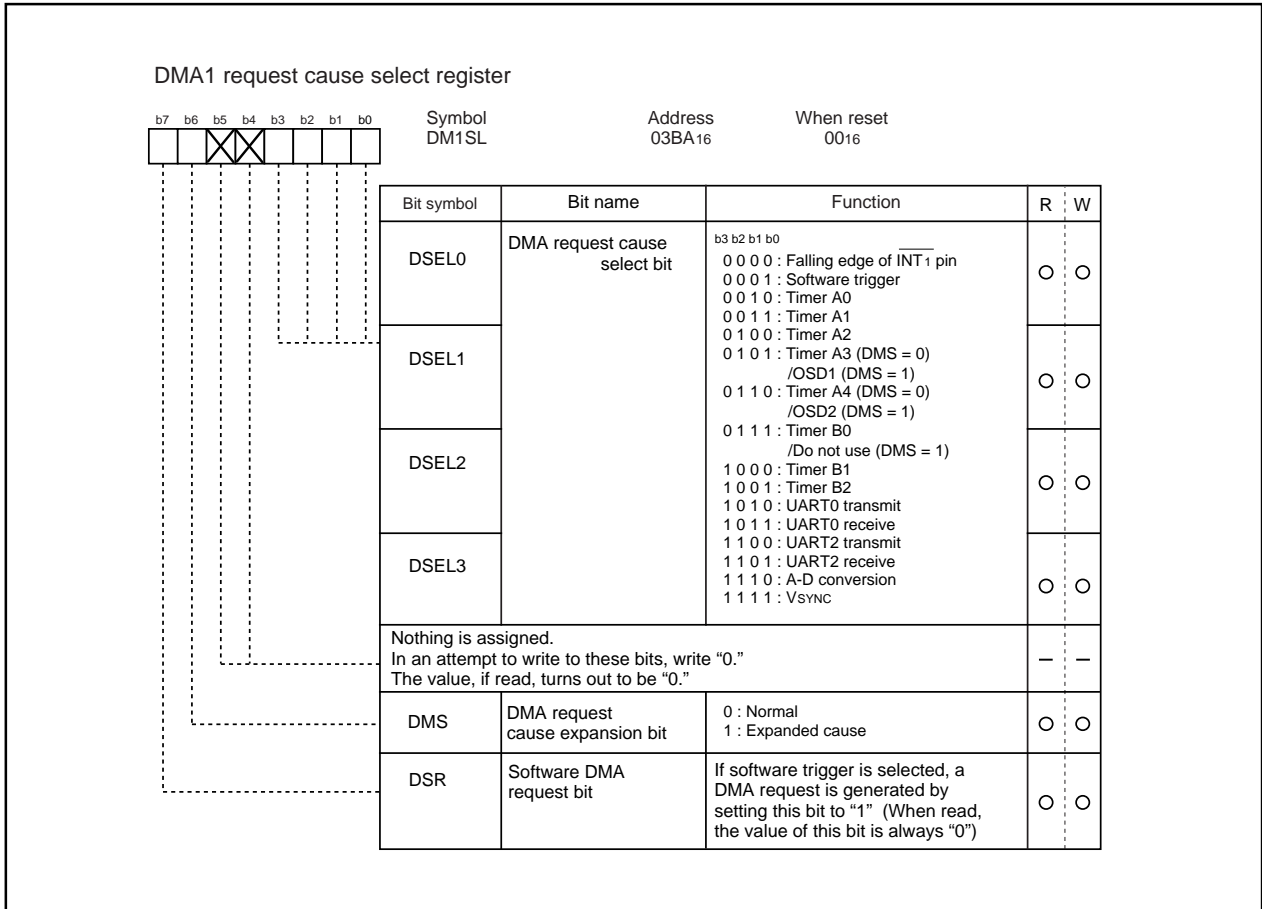


Figure 2.9.3 DMA1 request cause select register

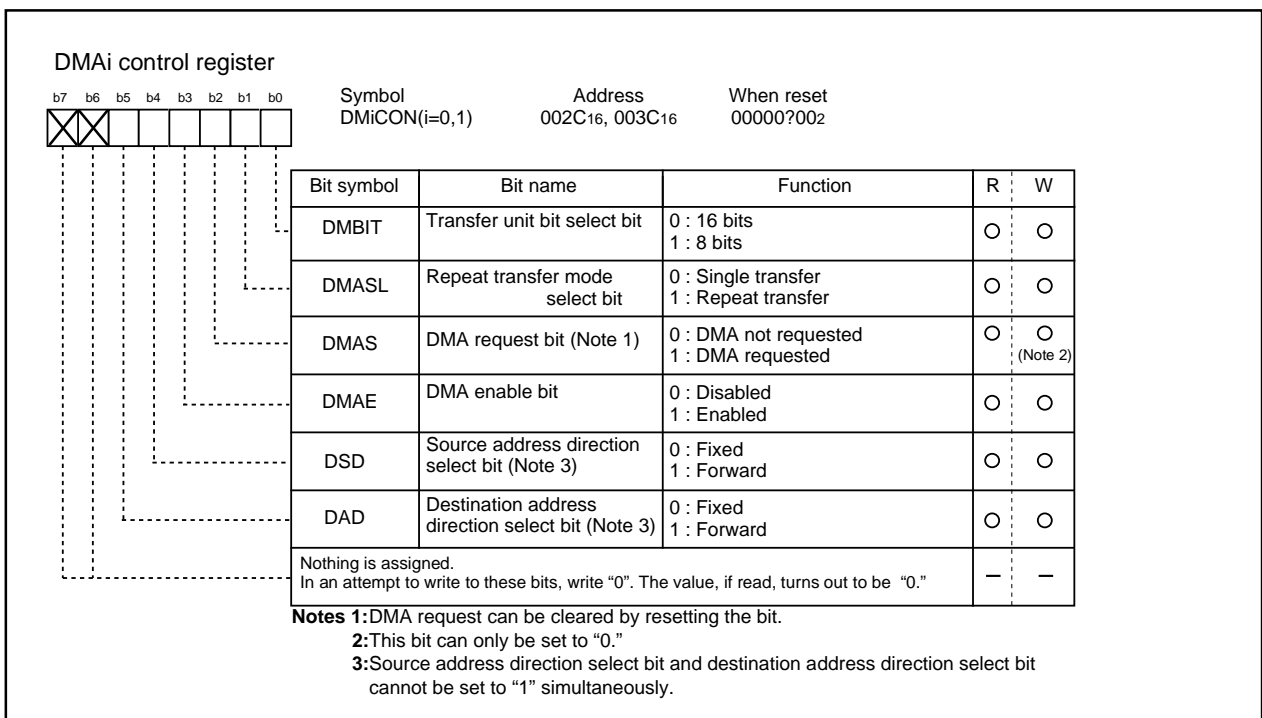
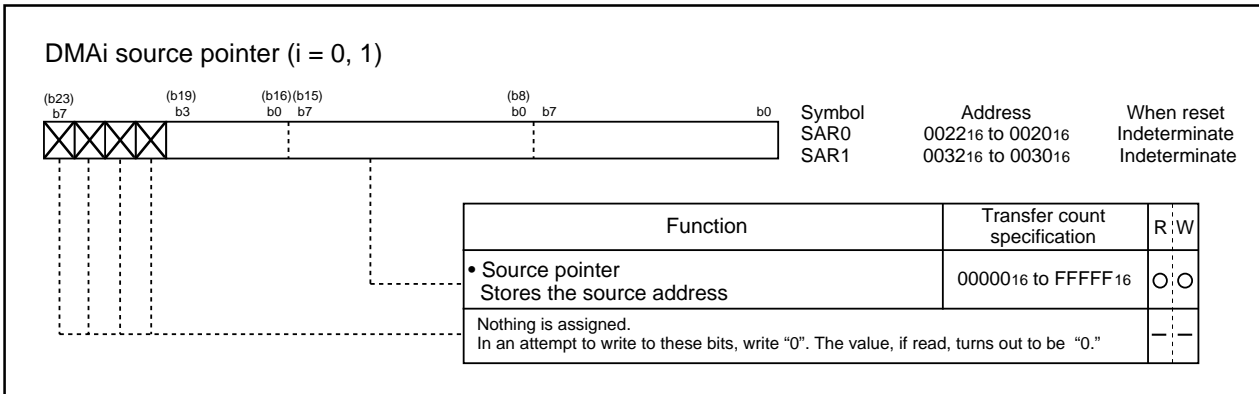


Figure 2.9.4 DMAi control register (i = 0, 1)

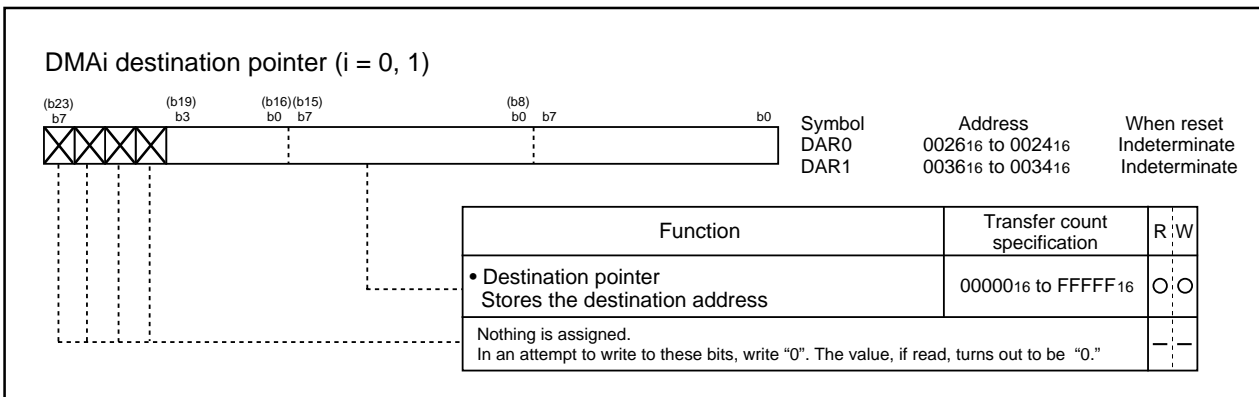


**M306V0ME-XXXFP**  
**M306V0EEFP**

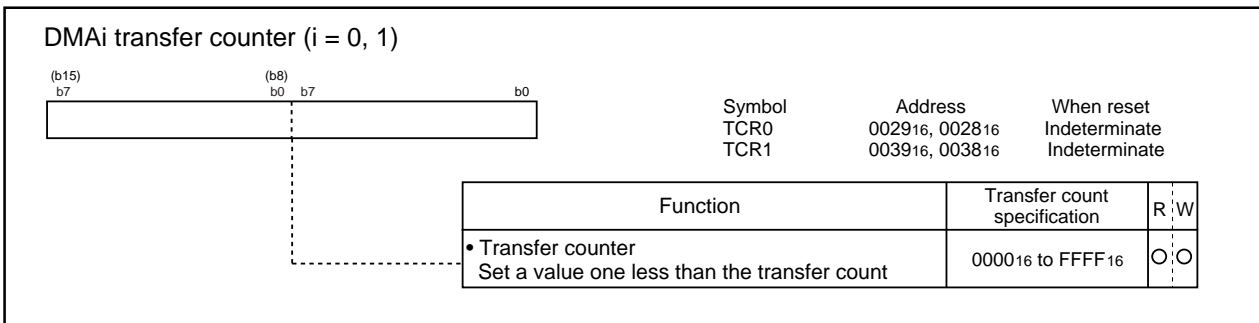
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.9.5 DMA<sub>i</sub> source pointer (i = 0, 1)**



**Figure 2.9.6 DMA<sub>i</sub> destination pointer (i = 0, 1)**



**Figure 2.9.7 DMA<sub>i</sub> transfer counter (i = 0, 1)**

### 2.9.1 Transfer cycle

The transfer cycle consists of the bus cycle in which data is read from memory or from the SFR area (source read) and the bus cycle in which the data is written to memory or to the SFR area (destination write). The number of read and write bus cycles depends on the source and destination addresses. In memory expansion mode and microprocessor mode, the number of read and write bus cycles also depends on the level of the BYTE pin. Also, the bus cycle itself is longer when software waits are inserted.

#### (1) Effect of source and destination addresses

When 16-bit data is transferred on a 16-bit data bus, and the source and destination both start at odd addresses, there are one more source read cycle and destination write cycle than when the source and destination both start at even addresses.

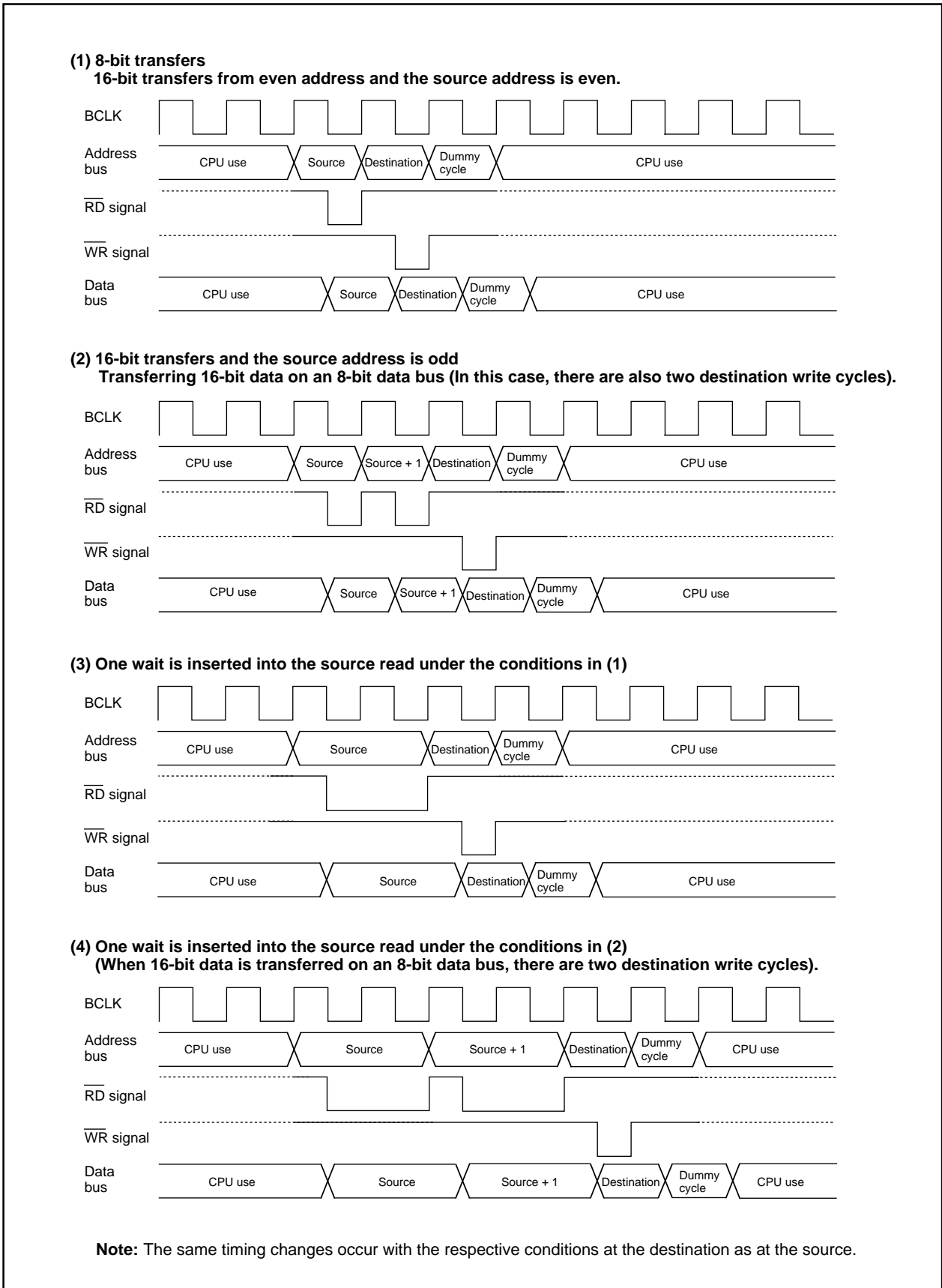
#### (2) Effect of BYTE pin level

When transferring 16-bit data over an 8-bit data bus (BYTE pin = "H") in memory expansion mode and microprocessor mode, the 16 bits of data are sent in two 8-bit blocks. Therefore, two bus cycles are required for reading the data and two are required for writing the data. Also, in contrast to when the CPU accesses internal memory, when the DMAC accesses internal memory (internal ROM, internal RAM, and SFR), these areas are accessed using the data size selected by the BYTE pin.

#### (3) Effect of software wait

When the SFR area or a memory area with a software wait is accessed, the number of cycles is increased for the wait by 1 bus cycle. The length of the cycle is determined by BCLK.

Figure 2.9.8 shows the example of the transfer 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 the transfer cycle, remember to apply the respective conditions to both the destination write cycle and the source read cycle. For example (2) in Figure 47, if data is being transferred in 16-bit units on an 8-bit bus, two bus cycles are required for both the source read cycle and the destination write cycle.



**Figure 2.9.8 Example of the transfer cycles for a source read**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**2.9.2 DMAC transfer cycles**

Any combination of even or odd transfer read and write addresses is possible. Table 2.9.2 shows the number of DMAC transfer cycles.

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

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

**Table 2.9.2 No. of DMAC 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

**Coefficient j, k**

Internal memory			External memory		
Internal ROM/RAM	Internal ROM/RAM • OSD RAM	SFR area	Separate bus	Separate bus	Multiplex bus
No wait	With wait		No wait	With wait	
1	2	2	1	2	3

### 2.9.3 DMA enable bit

Setting the DMA enable bit to 1 makes the DMAC active. The DMAC carries out the following operations at the time data transfer starts immediately after DMAC is turned active.

- (1) Reloads the value of one of the source pointer and the destination pointer - the one specified for the forward direction - to the forward direction address pointer.
- (2) Reloads the value of the transfer counter reload register to the transfer counter.

Thus overwriting 1 to the DMA enable bit with the DMAC being active carries out the operations given above, so the DMAC operates again from the initial state at the instant 1 is overwritten to the DMA enable bit.

### 2.9.4 DMA request bit

The DMAC can generate a DMA transfer request signal triggered by a factor chosen in advance out of DMA request factors for each channel.

DMA request factors include the following.

\* Factors effected by using the interrupt request signals from the built-in peripheral functions and software DMA factors (internal factors) effected by a program.

\* External factors effected by utilizing the input from external interrupt signals.

For the selection of DMA request factors, see the descriptions of the DMA<sub>i</sub> factor selection register.

The DMA request bit turns to 1 if the DMA transfer request signal occurs regardless of the DMAC's state (regardless of whether the DMA enable bit is set 1 or to 0). It turns to 0 immediately before data transfer starts.

In addition, it can be set to 0 by use of a program, but cannot be set to 1.

There can be instances in which a change in DMA request factor selection bit causes the DMA request bit to turn to 1. So be sure to set the DMA request bit to 0 after the DMA request factor selection bit is changed.

The DMA request bit turns to 1 if a DMA transfer request signal occurs, and turns to 0 immediately before data transfer starts. If the DMAC is active, data transfer starts immediately, so the value of the DMA request bit, if read by use of a program, turns out to be 0 in most cases. To examine whether the DMAC is active, read the DMA enable bit.

Here follows the timing of changes in the DMA request bit.

#### (1) Internal factors

Except the DMA request factors triggered by software, the timing for the DMA request bit to turn to 1 due to an internal factor is the same as the timing for the interrupt request bit of the interrupt control register to turn to 1 due to several factors.

Turning the DMA request bit to 1 due to an internal factor is timed to be effected immediately before the transfer starts.

#### (2) External factors

An external factor is a factor caused to occur by the leading edge of input from the  $\overline{\text{INT}}_i$  pin (i depends on which DMAC channel is used).

Selecting the  $\overline{\text{INT}}_i$  pins as external factors using the DMA request factor selection bit causes input from these pins to become the DMA transfer request signals.

The timing for the DMA request bit to turn to 1 when an external factor is selected synchronizes with

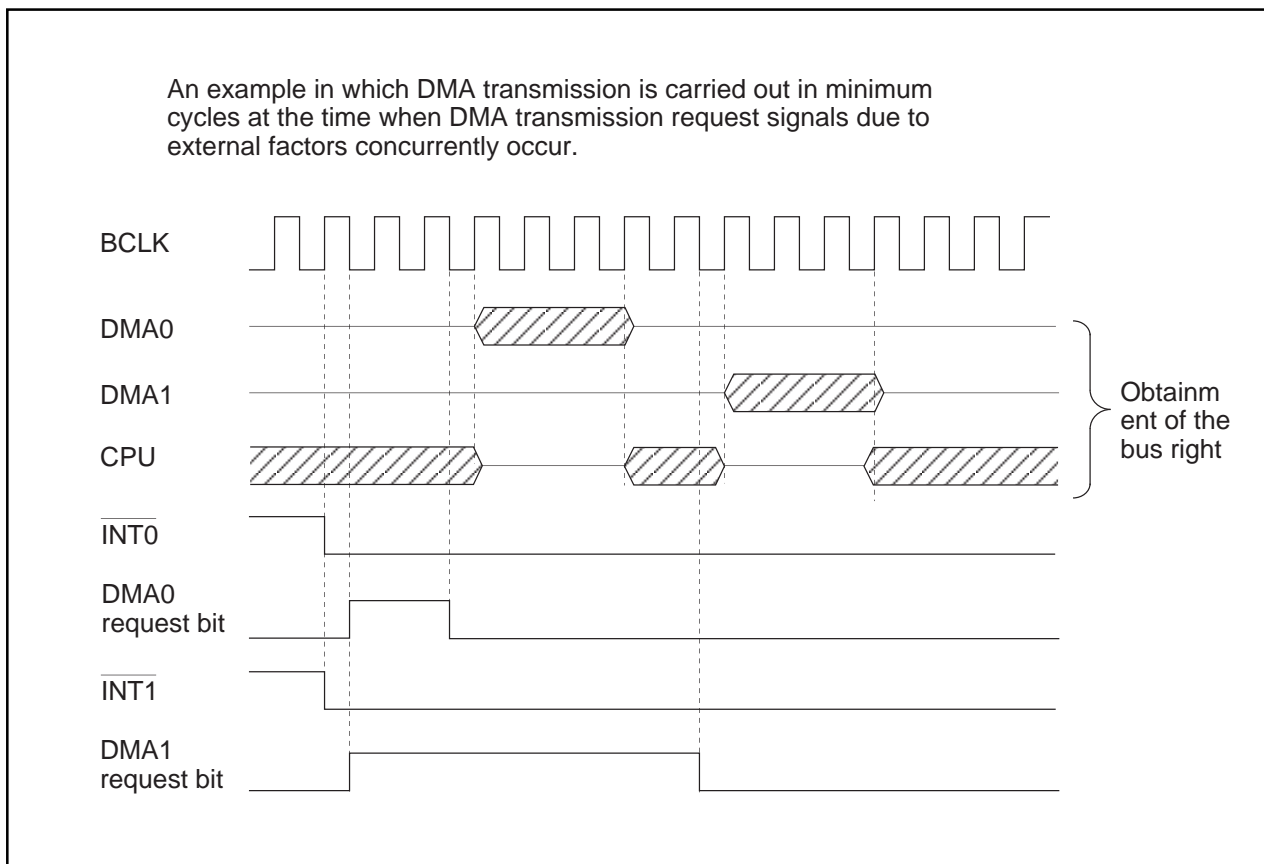
the signal's edge applicable to the function specified by the DMA request factor selection bit (synchronizes with the trailing edge of the input signal to each  $\overline{INTi}$  pin, for example).

With an external factor selected, the DMA request bit is timed to turn to 0 immediately before data transfer starts similarly to the state in which an internal factor is selected.

**(3) The priorities of channels and DMA transfer timing**

If a DMA transfer request signal falls on a single sampling cycle (a sampling cycle means one period from the leading edge to the trailing edge of BCLK), the DMA request bits of applicable channels concurrently turn to 1. If the channels are active at that moment, DMA0 is given a high priority to start data transfer. When DMA0 finishes data transfer, it gives the bus right to the CPU. When the CPU finishes single bus access, then DMA1 starts data transfer and gives the bus right to the CPU. Figure 2.9.9 illustrates these operations.

An example in which DMA transfer is carried out in minimum cycles at the time when DMA transfer request signals due to external factors concurrently occur.



**Figure 2.9.9 An example of DMA transfer effected by external factors**

### 2.10 Timer

There are eight 16-bit timers. These timers can be classified by function into timers A (five) and timers B (three). All these timers function independently. Figures 2.10.1 and 2.10.2 show the block diagram of timers.

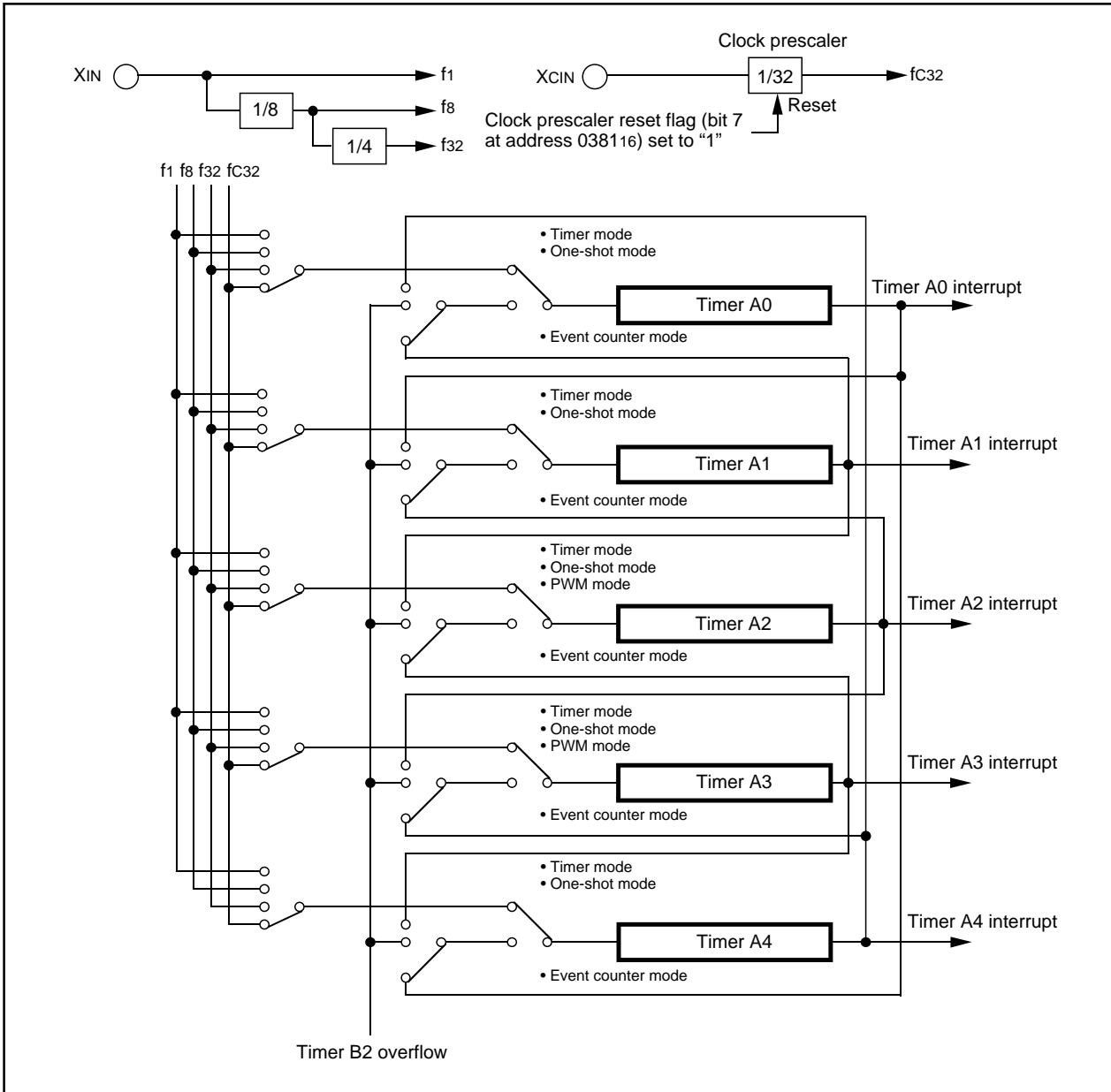
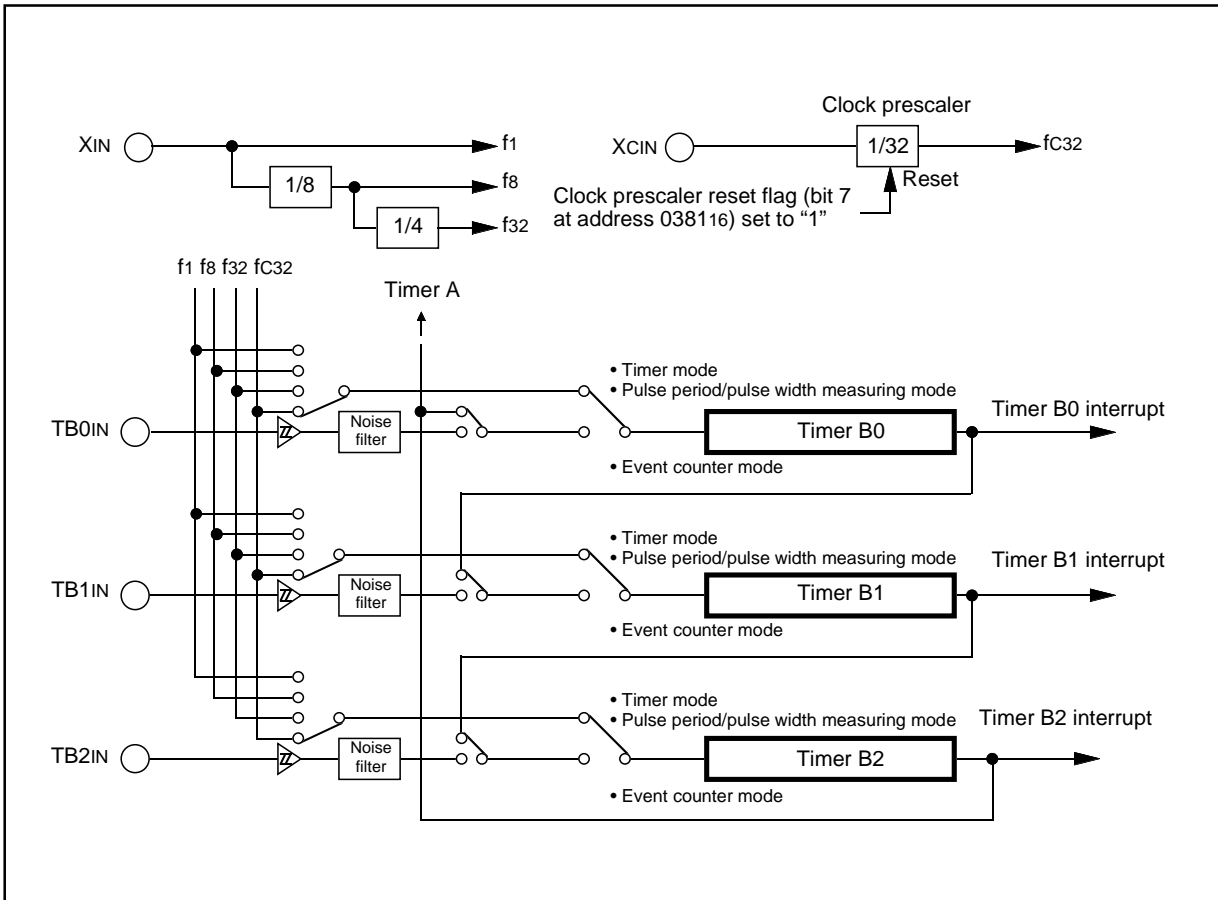


Figure 2.10.1 Timer A block diagram

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.10.2 Timer B block diagram**



2.10.1 Timer A

Figure 2.10.3 shows the block diagram of timer A. Figures 2.10.4 to 2.10.10 show the timer A-related registers.

Except the pulse output function, timers A0 through A4 all have the same function. Use the timer Ai mode register (i = 0 to 4) bits 0 and 1 to choose the desired mode.

Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts a timer over flow.
- One-shot timer mode: The timer stops counting when the count reaches "0000<sub>16</sub>".
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

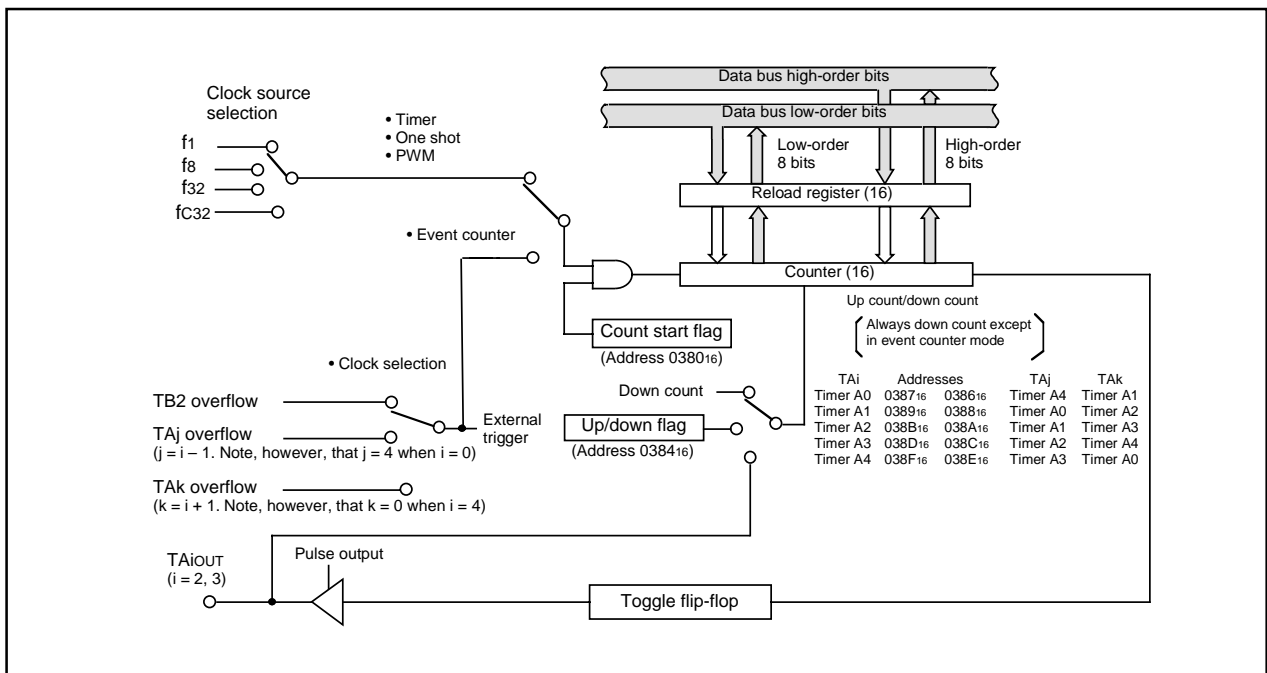


Figure 2.10.3 Block diagram of timer A

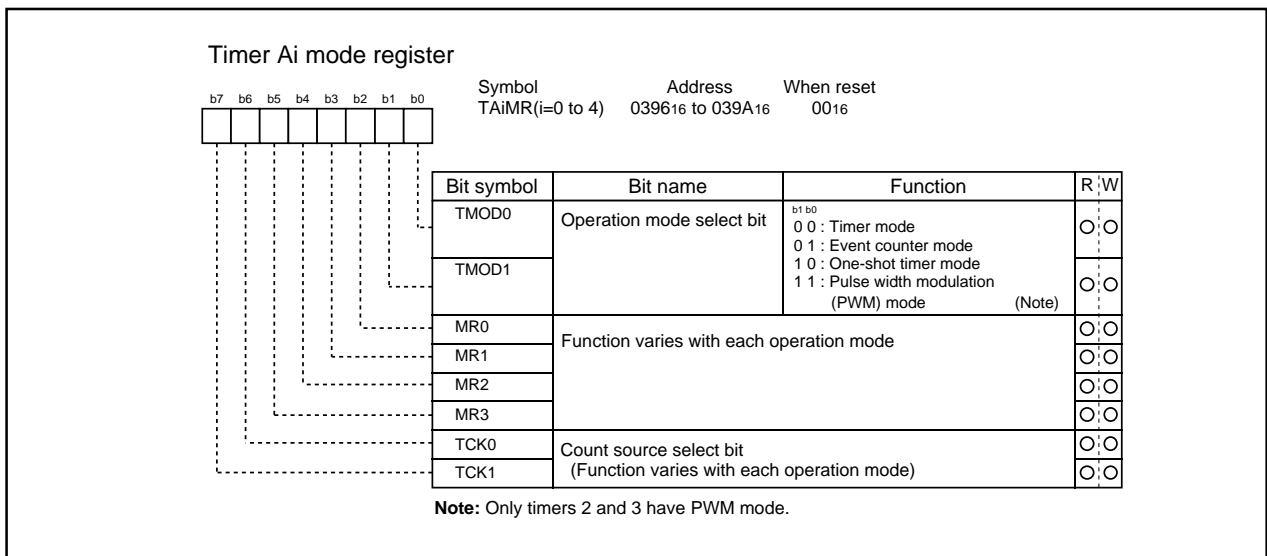


Figure 2.10.4 Timer Ai mode register (i = 0 to 4)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

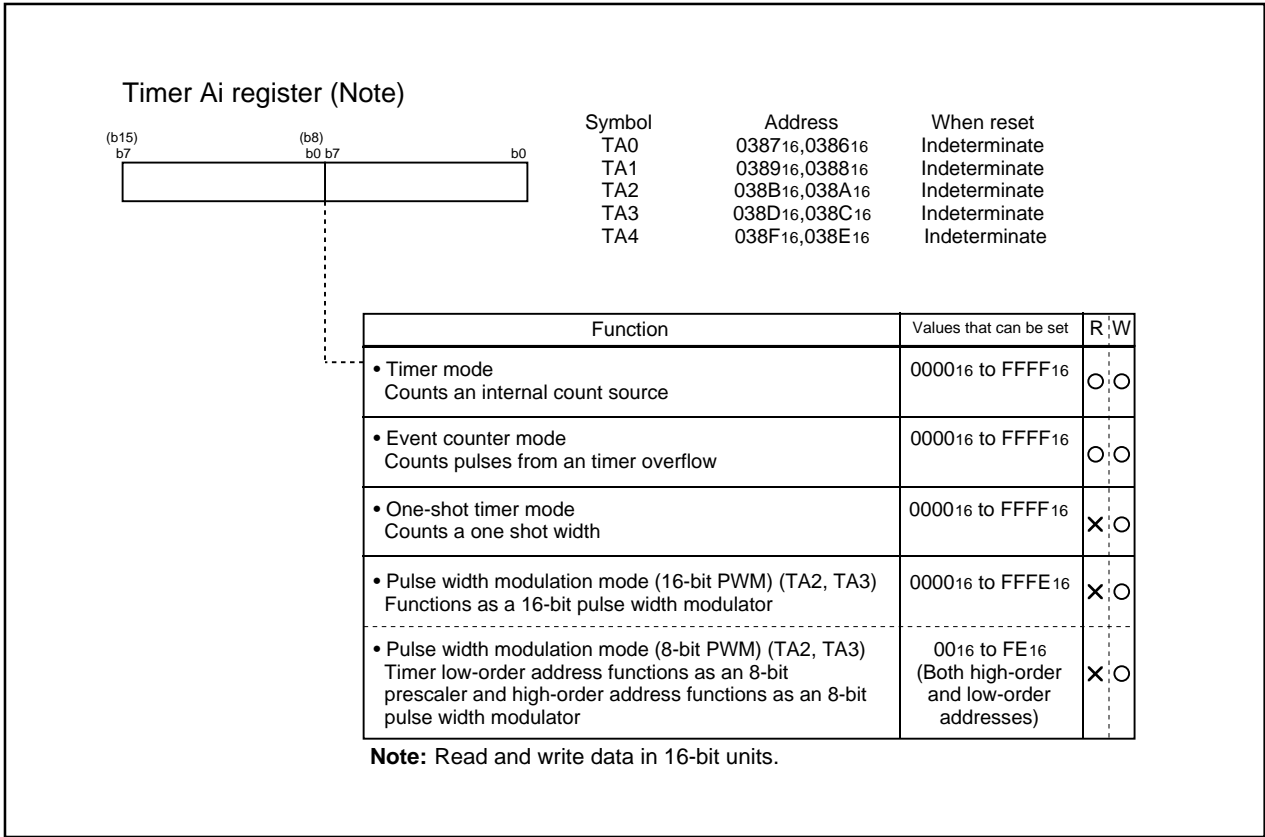


Figure 2.10.5 Timer Ai register (i = 0 to 4)

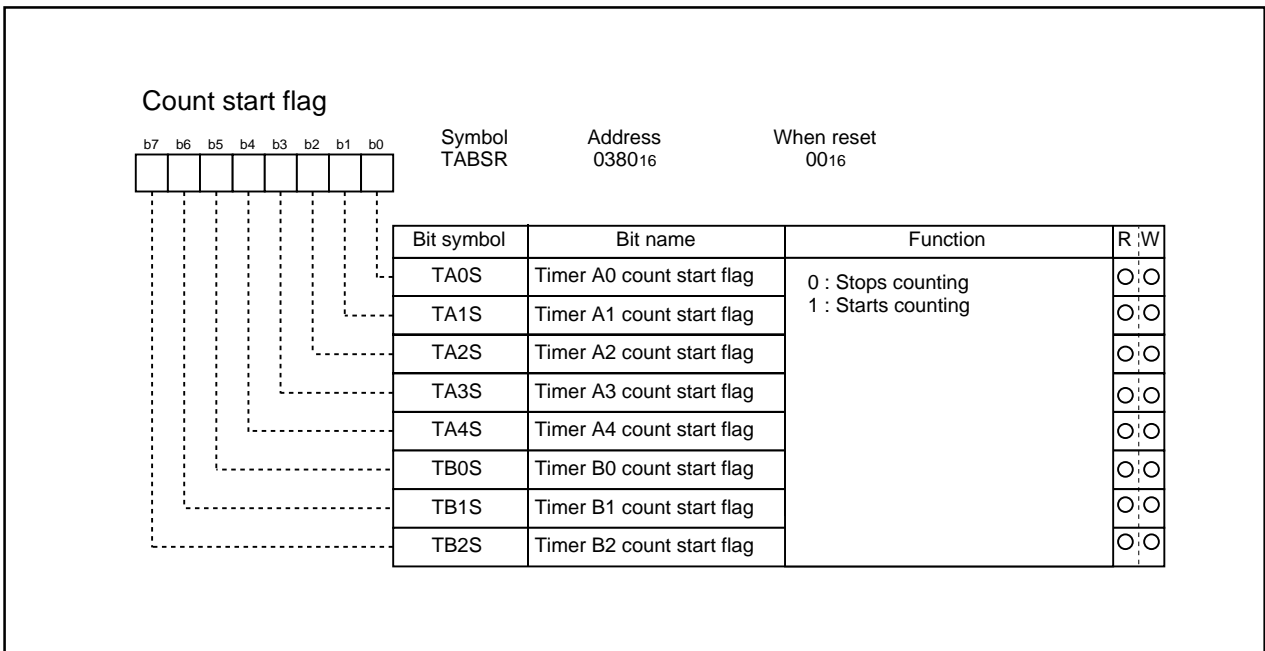


Figure 2.10.6 Count start flag

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

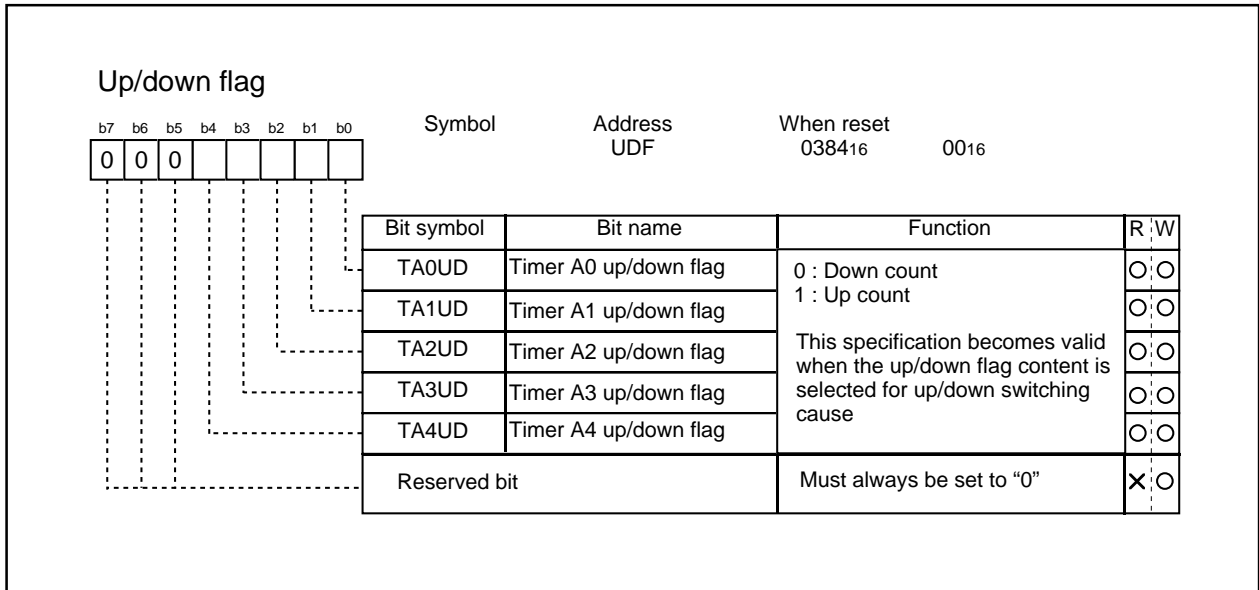


Figure 2.10.7 Up/down flag

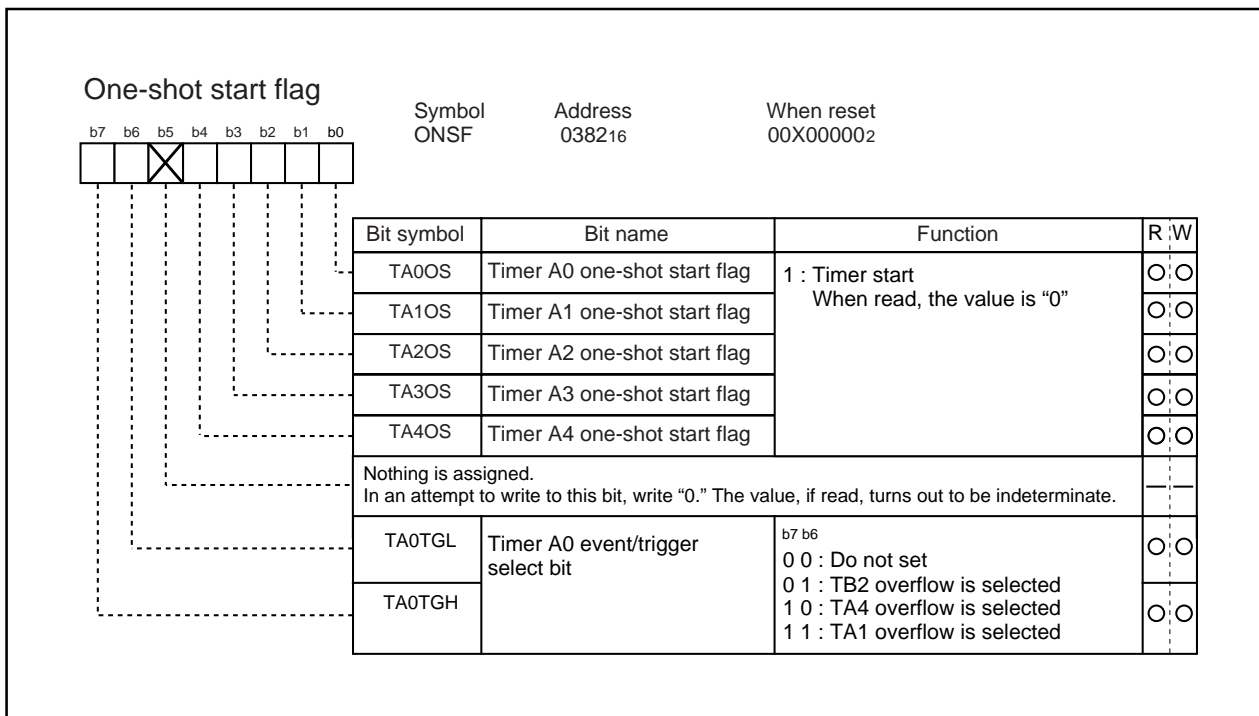


Figure 2.10.8 One-shot start flag

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

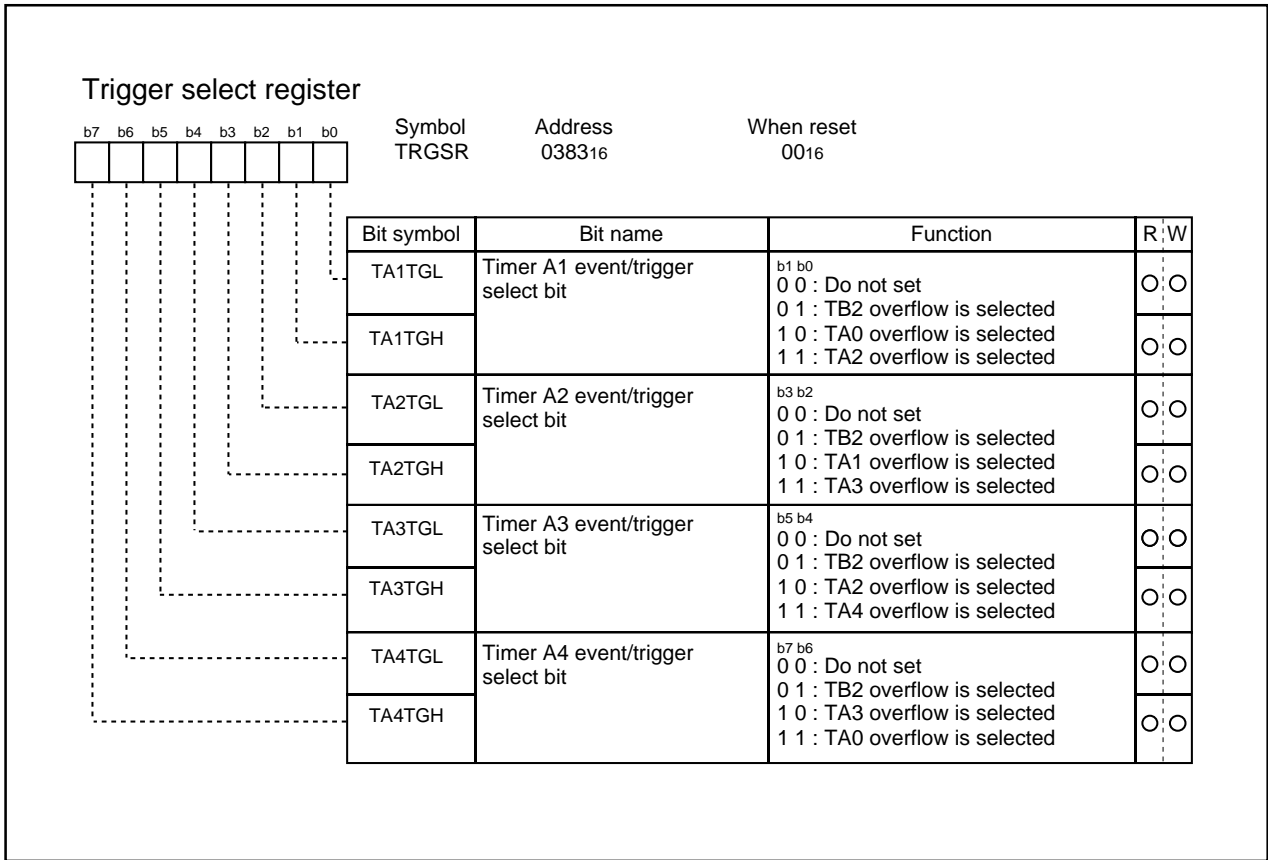


Figure 2.10.9 Trigger select register

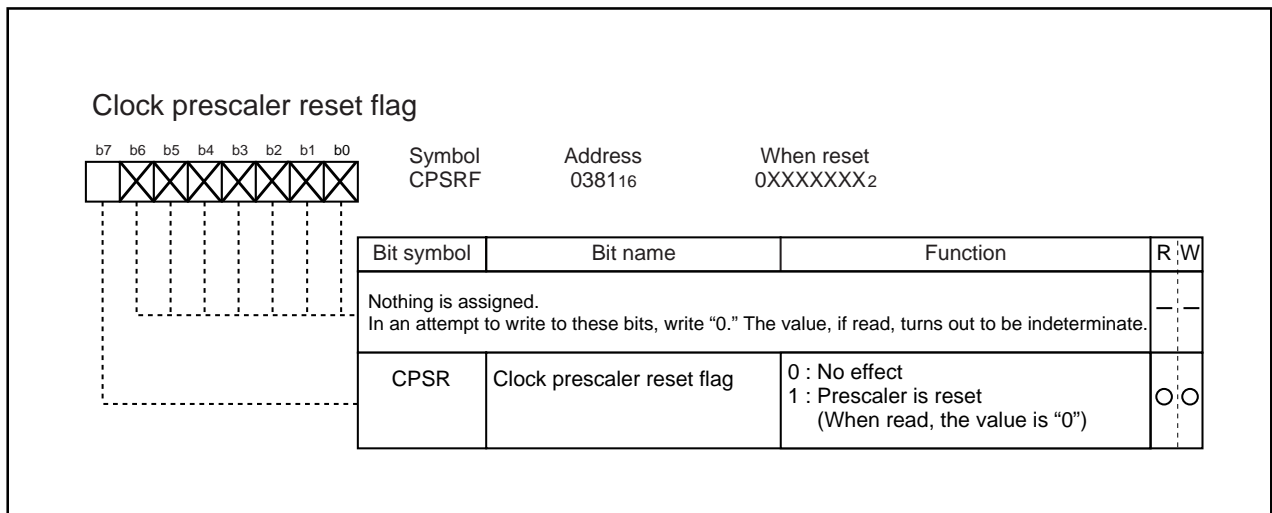


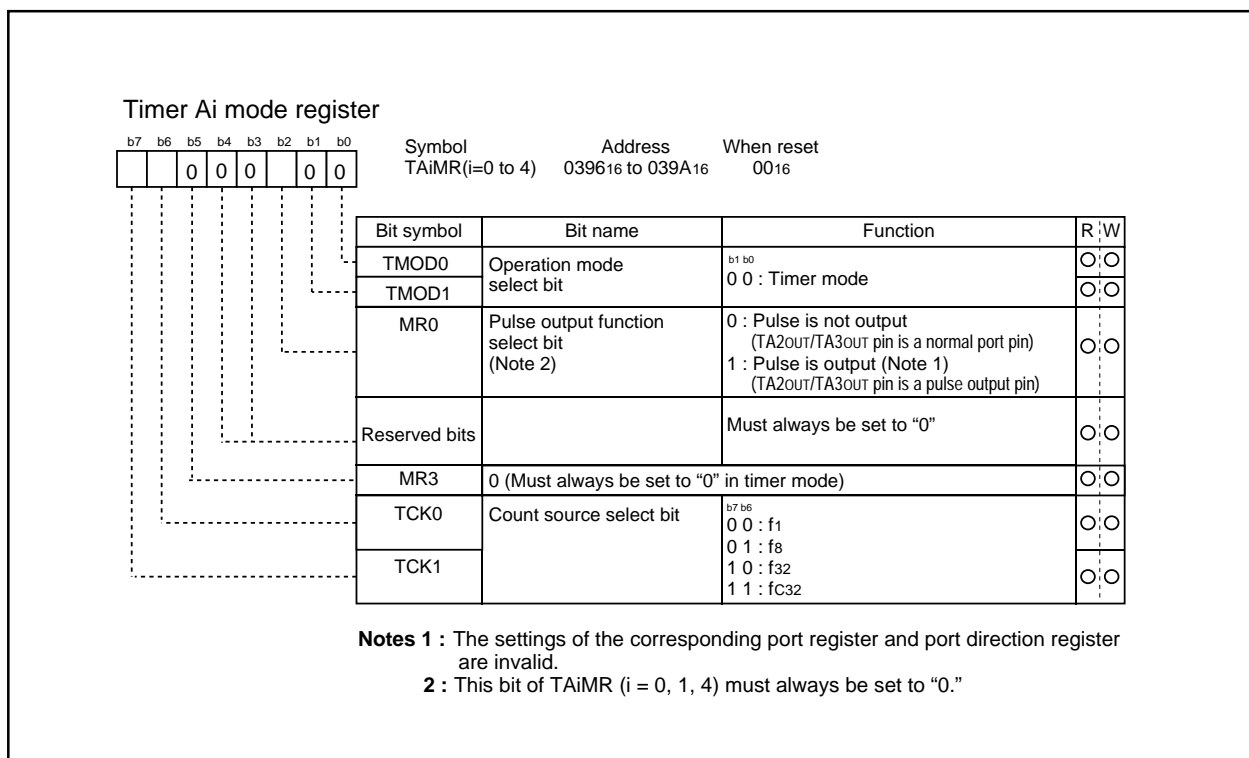
Figure 2.10.10 Clock prescaler reset flag

**(1) Timer mode**

In this mode, the timer counts an internally generated count source. (See Table 2.10.1.) Figure 2.10.11 shows the timer Ai mode register in timer mode.

**Table 2.10.1 Specifications of timer mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>8</sub> , f <sub>32</sub> , f <sub>c32</sub>
Count operation	<ul style="list-style-type: none"> <li>Down count</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TA2 <sub>OUT</sub> /TA3 <sub>OUT</sub> pin function	Programmable I/O port or pulse output
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Pulse output function Each time the timer underflows, the TA<sub>i</sub>OUT pin's polarity is reversed</li> </ul>



**Figure 2.10.11 Timer Ai mode register in timer mode (i = 0 to 4)**

**(2) Event counter mode**

In this mode, the timer counts an internal timer's overflow.

**Table 2.10.2 Timer specifications in event counter mode**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>• TB2 overflow, TAj overflow, TAk overflow</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>• Up count or down count can be selected by external signal or software</li> <li>• When the timer overflows or underflows, it reloads the reload register contents before continuing counting (Note)</li> </ul>
Divide ratio	$1 / (FFFF_{16} - n + 1)$ for up count $1 / (n + 1)$ for down count      n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TA2 <sub>OUT</sub> /TA3 <sub>OUT</sub> pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> <li>• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>• When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it</li> <li>• Pulse output function Each time the timer overflows or underflows, the TAi<sub>OUT</sub> pin's polarity is reversed</li> </ul>

**Note:** This does not apply when the free-run function is selected.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

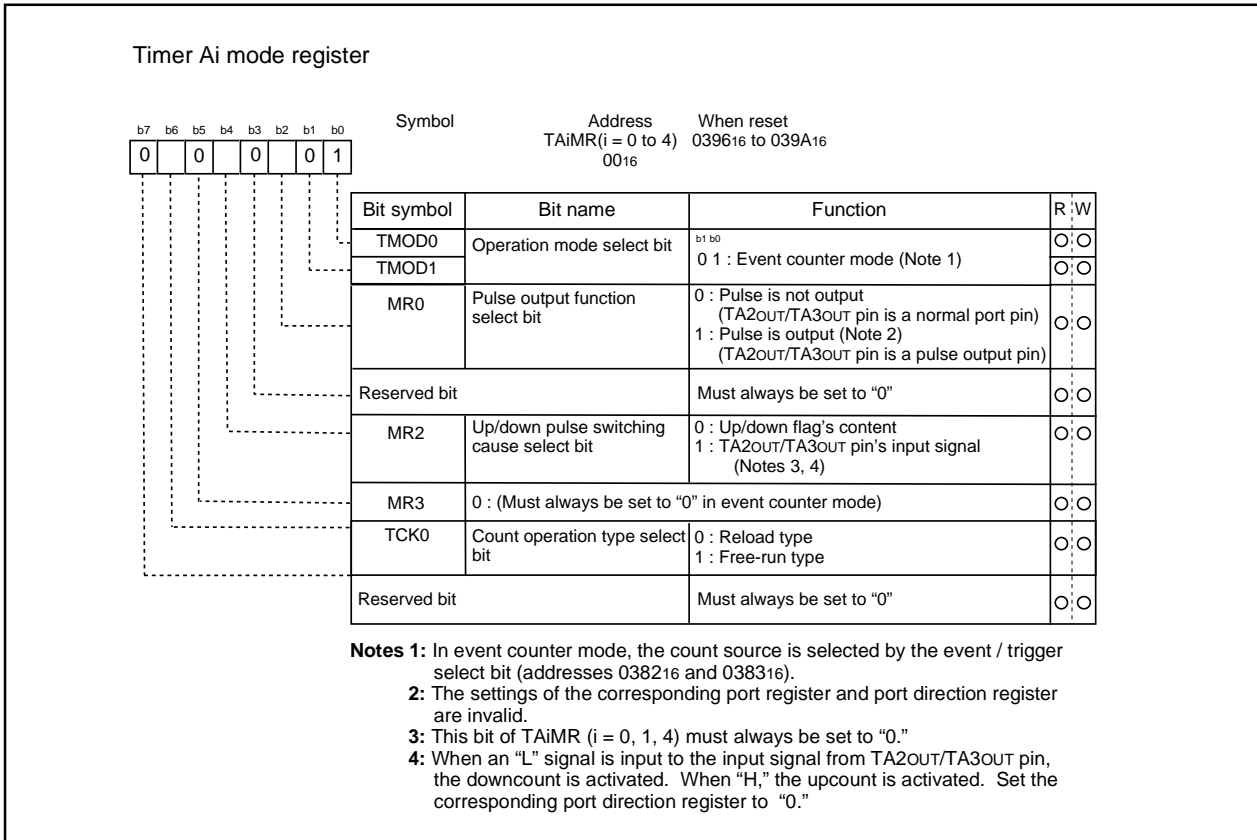


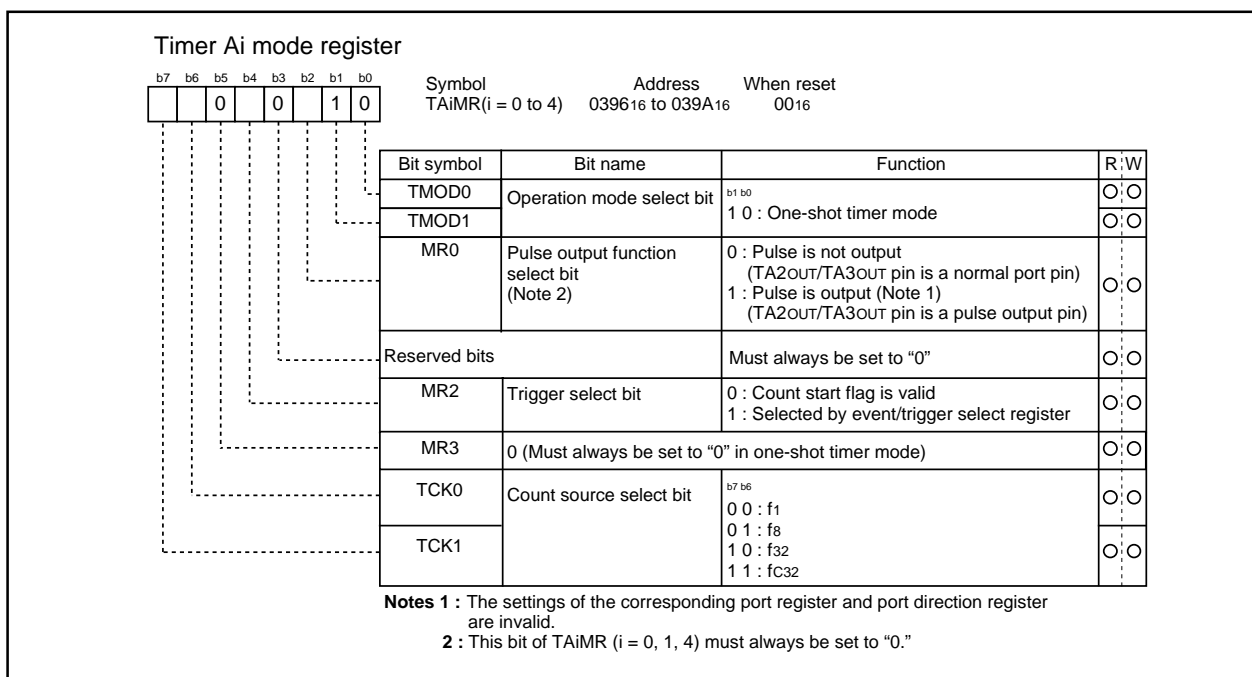
Figure 2.10.12 Timer Ai mode register in event counter mode (i = 0 to 4)

**(3) One-shot timer mode**

In this mode, the timer operates only once. (See Table 2.10.3.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 2.10.13 shows the timer Ai mode register in one-shot timer mode.

**Table 2.10.3 Timer specifications in one-shot timer mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>The timer counts down</li> <li>When the count reaches 0000<sub>16</sub>, the timer stops counting after reloading a new count</li> <li>If a trigger occurs when counting, the timer reloads a new count and restarts counting</li> </ul>
Divide ratio	1/n n : Set value
Count start condition	<ul style="list-style-type: none"> <li>The timer overflows</li> <li>The one-shot start flag is set (= 1)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>A new count is reloaded after the count has reached 0000<sub>16</sub></li> <li>The count start flag is reset (= 0)</li> </ul>
Interrupt request generation timing	The count reaches 0000 <sub>16</sub>
TA2 <sub>OUT</sub> /TA3 <sub>OUT</sub> pin function	Programmable I/O port or pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>



**Figure 2.10.13 Timer Ai mode register in one-shot timer mode (i = 0 to 4)**



**M306V0ME-XXXFP**  
**M306V0EEFP**

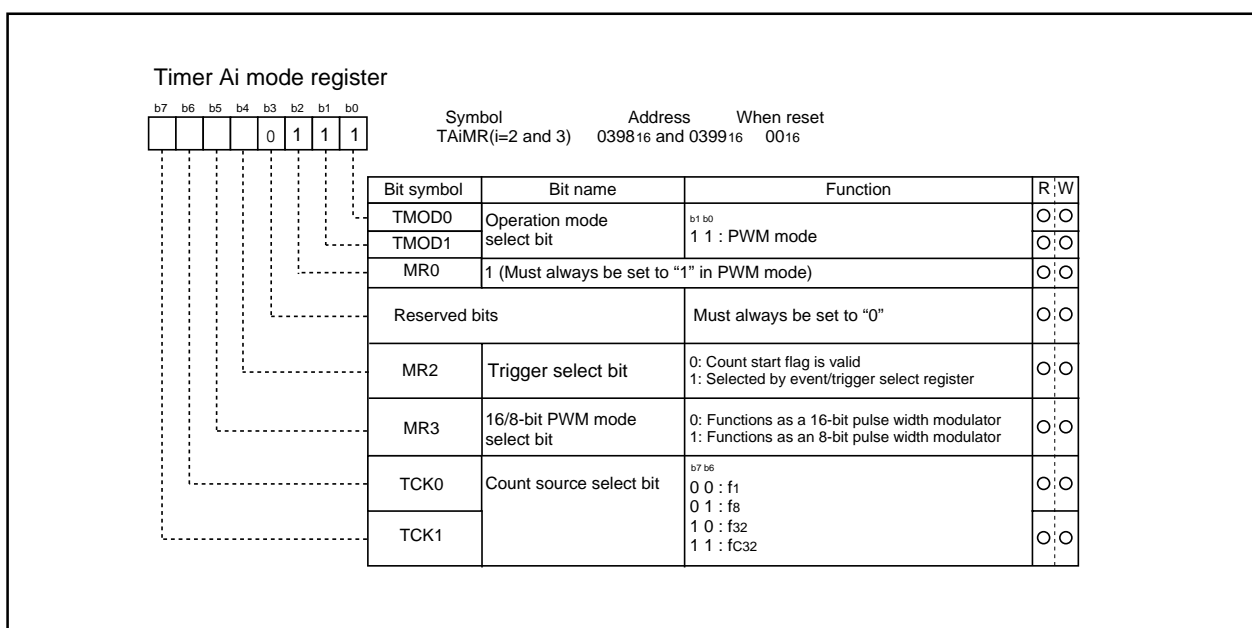
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**(4) Pulse width modulation (PWM) mode**

In this mode, the timer outputs pulses of a given width in succession. (See Table 2.10.4.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 2.10.14 shows the timer Ai mode register in pulse width modulation mode. Figure 2.10.15 shows the example of how an 8-bit pulse width modulator operates.

**Table 2.10.4 Timer specifications in pulse width modulation mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>8</sub> , f <sub>32</sub> , f <sub>c32</sub>
Count operation	<ul style="list-style-type: none"> <li>The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator)</li> <li>The timer reloads a new count at a rising edge of PWM pulse and continues counting</li> <li>The timer is not affected by a trigger that occurs when counting</li> </ul>
16-bit PWM •	High level width $n / f_i$ n : Set value Cycle time $(2^{16}-1) / f_i$ fixed
8-bit PWM	High level width $n \times (m+1) / f_i$ n : values set to timer Ai register's high-order address Cycle time $(2^8-1) \times (m+1) / f_i$ m : values set to timer Ai register's low-order address
Count start condition	<ul style="list-style-type: none"> <li>The timer overflows</li> <li>The count start flag is set (= 1)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>The count start flag is reset (= 0)</li> </ul>
Interrupt request generation timing	PWM pulse goes "L"
TA2 <sub>OUT</sub> /TA3 <sub>OUT</sub> pin function	Pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>



**Figure 2.10.14 Timer Ai mode register in pulse width modulation mode (i = 2 and 3)**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

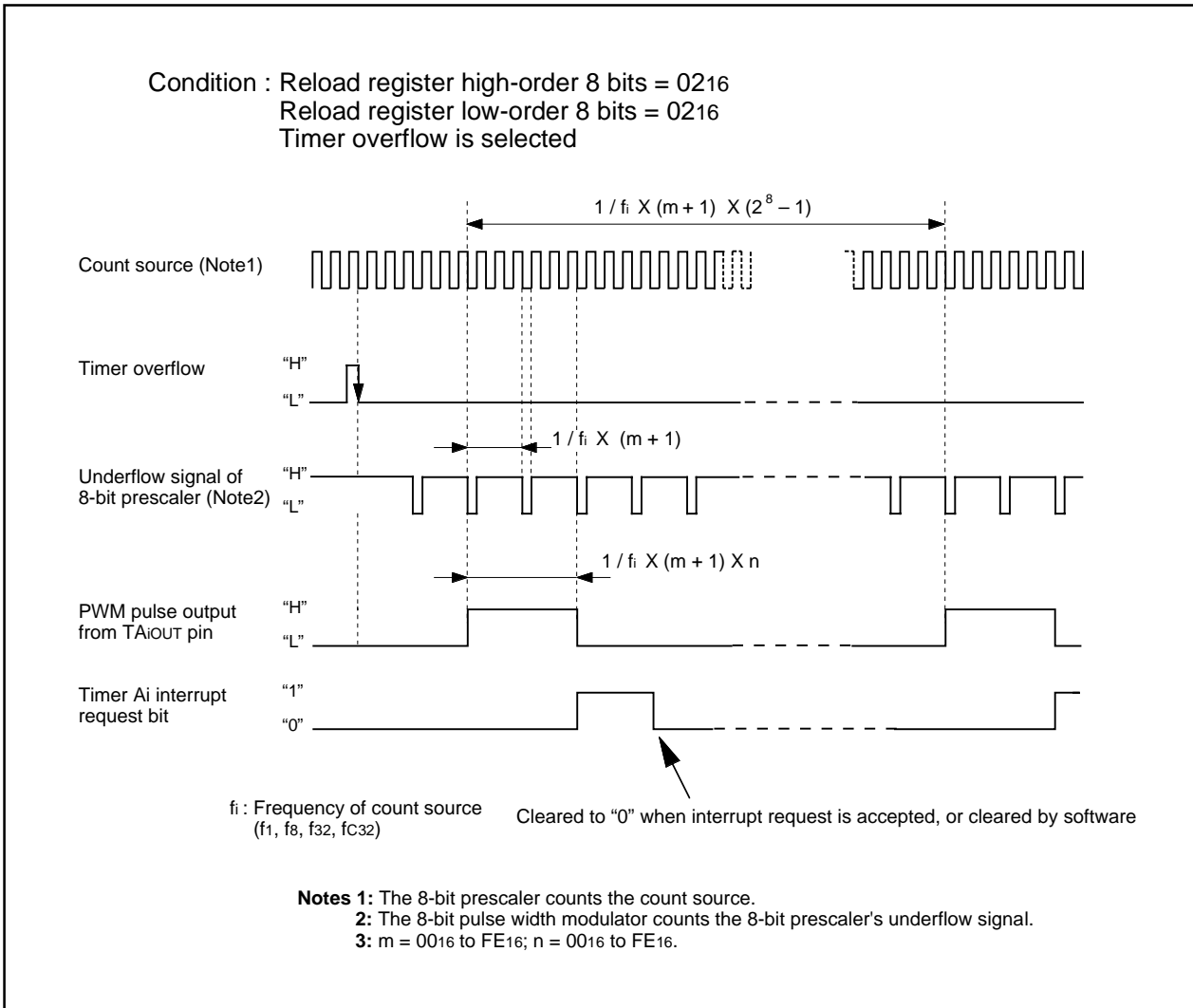


Figure 2.10.15 Example of how an 8-bit pulse width modulator operates

2.10.2 Timer B

Figure 2.10.17 shows the block diagram of timer B. Figures 2.10.17 and 2.10.20 show the timer B-related registers.

Use the timer Bi mode register (i = 0 to 2) bits 0 and 1 to choose the desired mode.

Timer B has three operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer overflow.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.

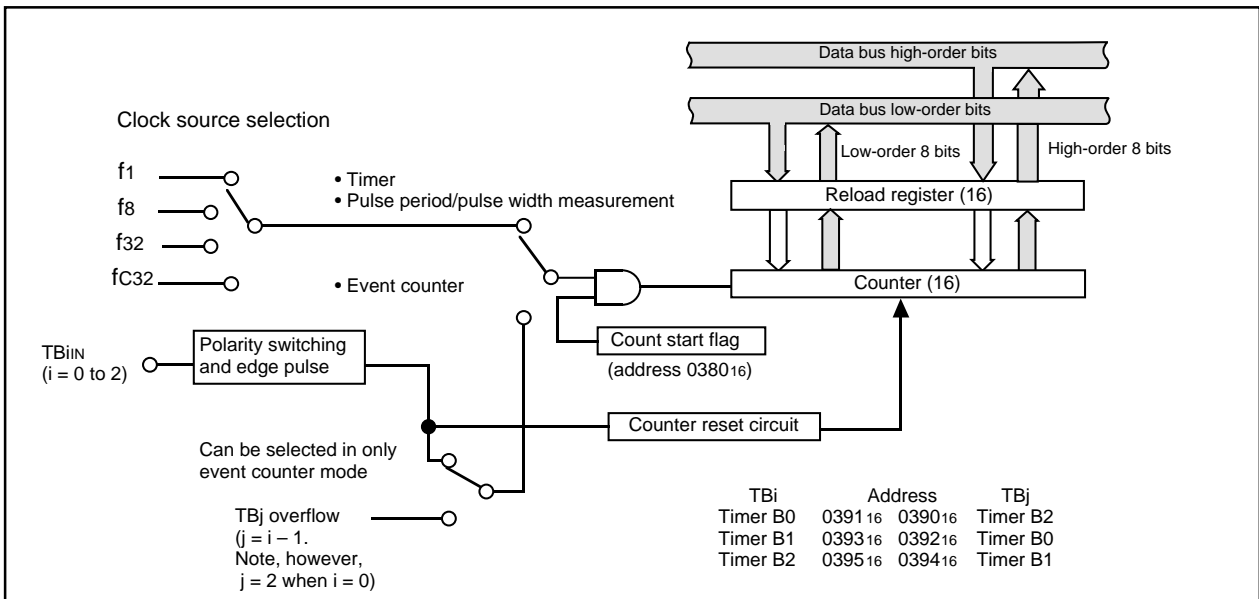


Figure 2.10.16 Block diagram of timer B

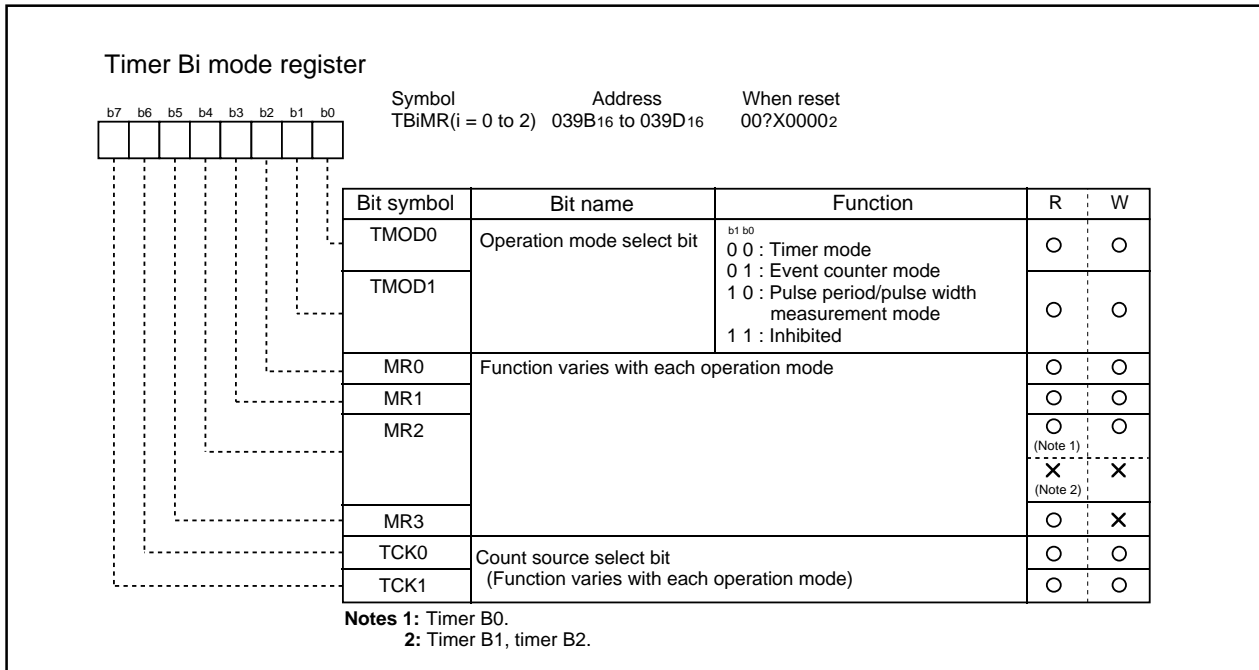
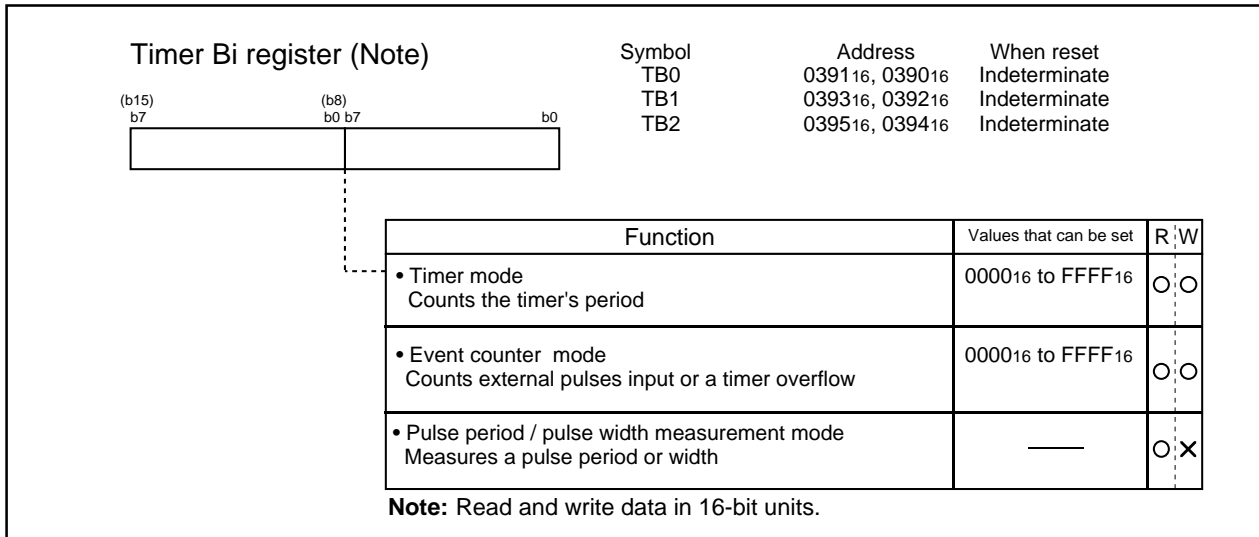


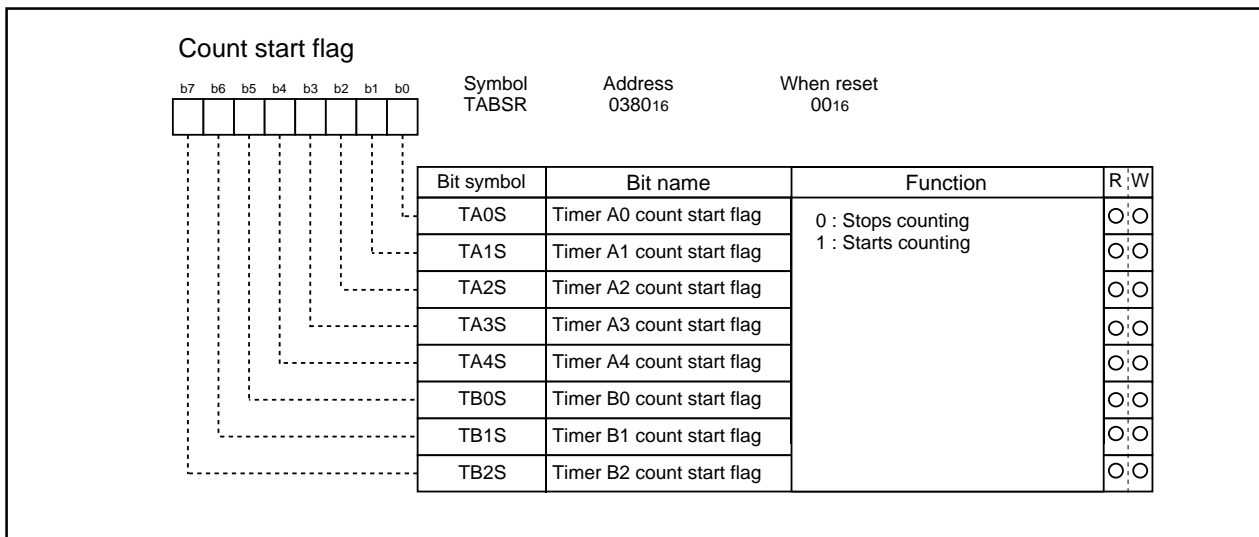
Figure 2.10.17 Timer Bi mode register (i = 0 to 2)

**M306V0ME-XXXFP**  
**M306V0EEFP**

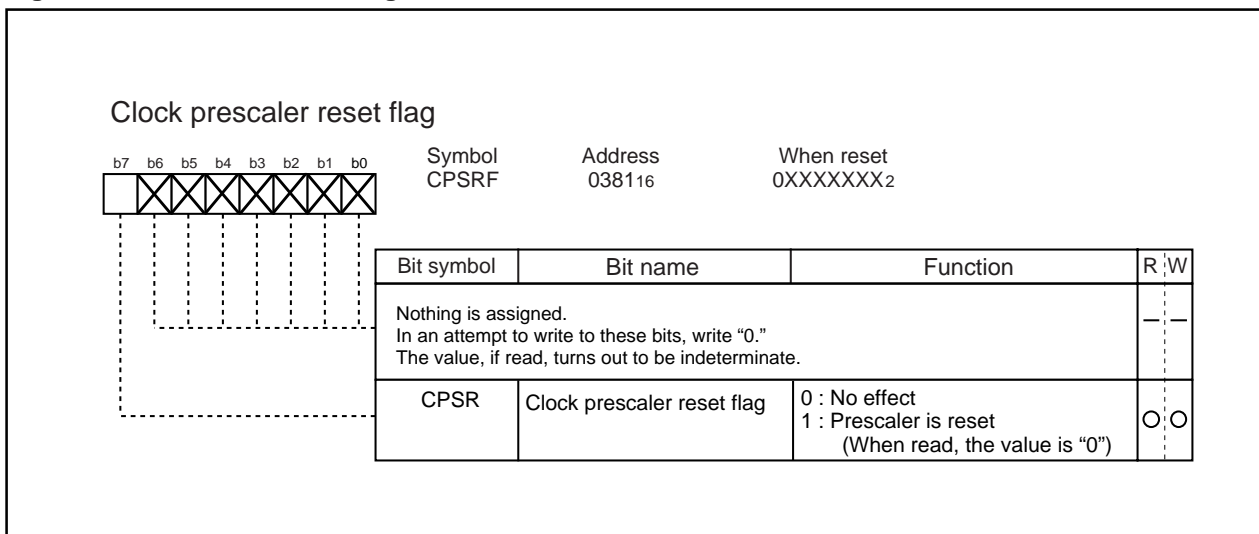
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.10.18** Timer Bi register (i = 0 to 2)



**Figure 2.10.19** Count start flag



**Figure 2.10.20** Clock prescaler reset flag

**M306V0ME-XXXFP**  
**M306V0EEFP**

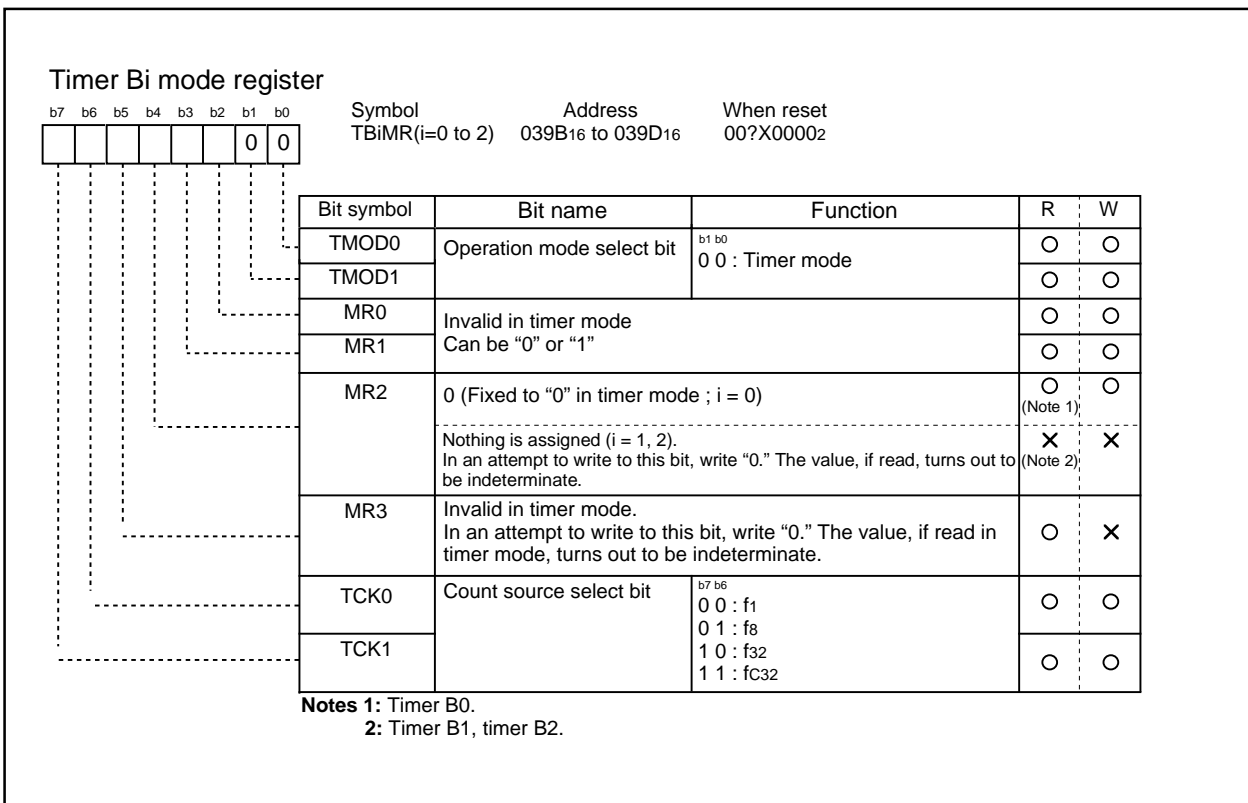
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**(1) Timer mode**

In this mode, the timer counts an internally generated count source. (See Table 2.10.5) Figure 2.10.21 shows the timer Bi mode register in timer mode.

**Table 2.10.5 Timer specifications in timer mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>Counts down</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>



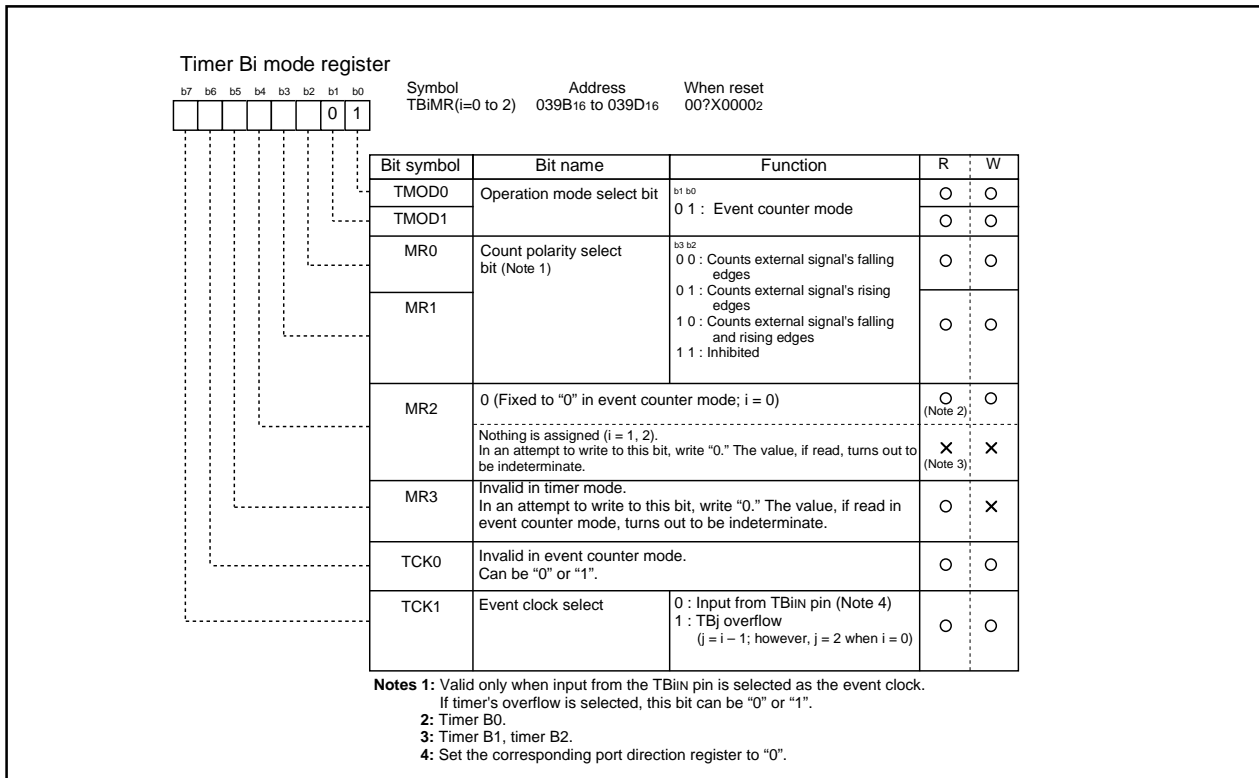
**Figure 2.10.21 Timer Bi mode register in timer mode (i = 0 to 2)**

**(2) Event counter mode**

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 2.10.6) Figure 2.10.22 shows the timer Bi mode register in event counter mode.

**Table 2.10.6 Timer specifications in event counter mode**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signals input to TBiIN pin</li> <li>Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Counts down</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
Divide ratio	1/(n+1)      n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Count source input
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter</li> <li>When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)</li> </ul>



**Figure 2.10.22 Timer Bi mode register in event counter mode (i = 0 to 2)**

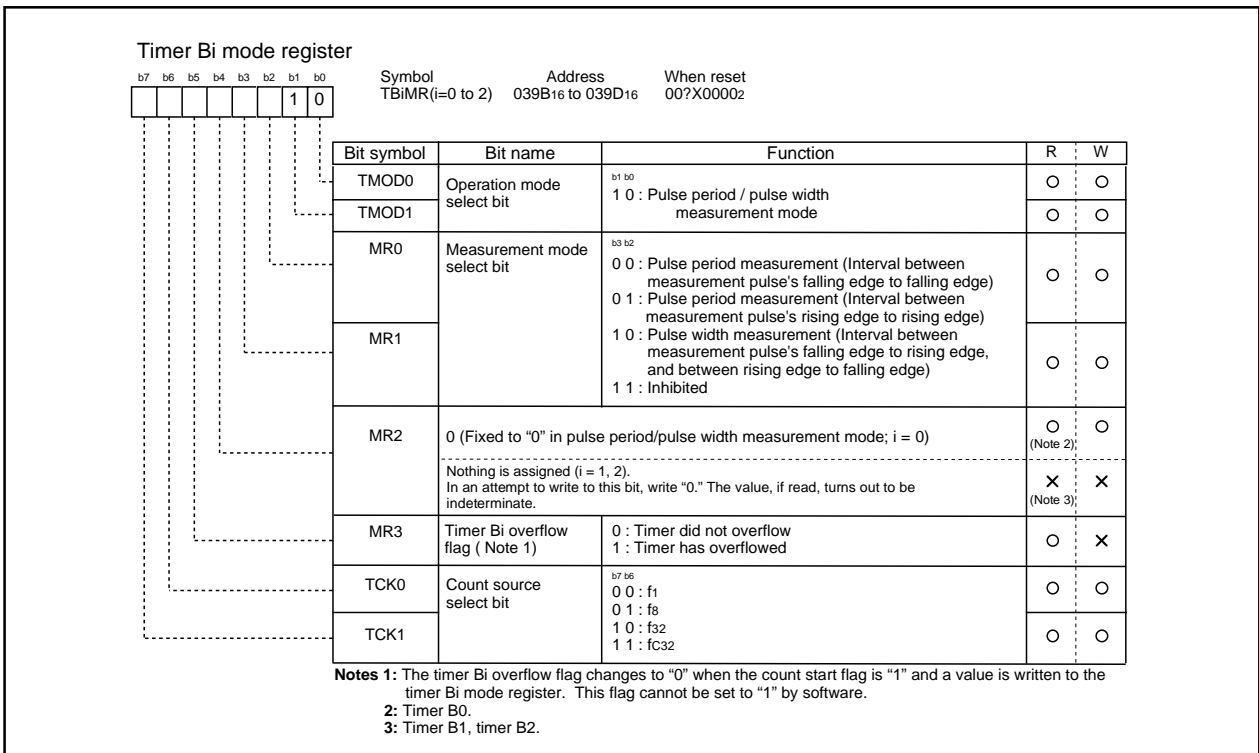
**(3) Pulse period/pulse width measurement mode**

In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 2.10.7) Figure 2.10.23 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 2.10.24 shows the operation timing when measuring a pulse period. Figure 2.10.25 shows the operation timing when measuring a pulse width.

**Table 2.10.7 Timer specifications in pulse period/pulse width measurement mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>• Up count</li> <li>• Counter value "0000<sub>16</sub>" is transferred to reload register at measurement pulse's effective edge and the timer continues counting</li> </ul>
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When measurement pulse's effective edge is input (Note 1)</li> <li>• When an overflow occurs. (Simultaneously, the timer Bi overflow flag changes to "1". The timer Bi overflow flag changes to "0" when the count start flag is "1" and a value is written to the timer Bi mode register.)</li> </ul>
TBiIN pin function	Measurement pulse input
Read from timer	When timer Bi register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

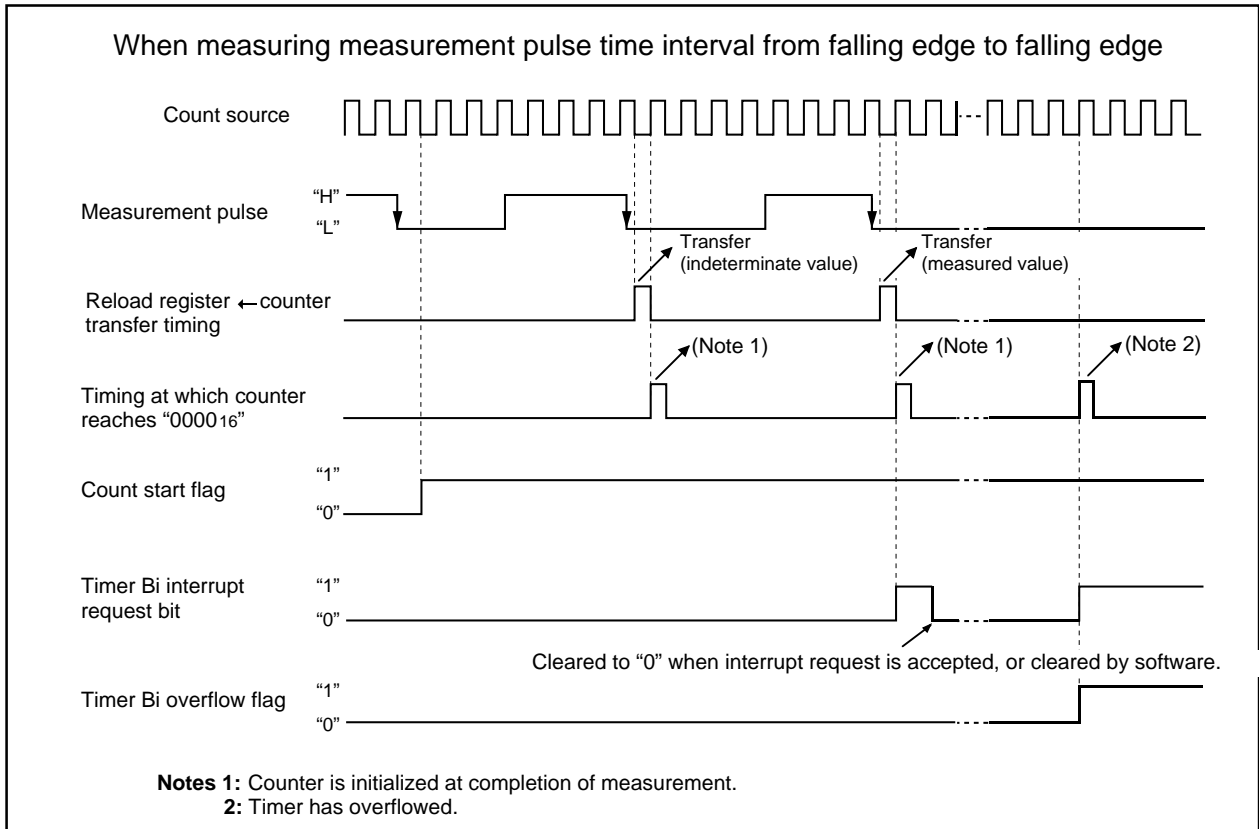
**Notes 1:** An interrupt request is not generated when the first effective edge is input after the timer has started counting.  
**2:** The value read out from the timer Bi register is indeterminate until the second effective edge is input after the timer.



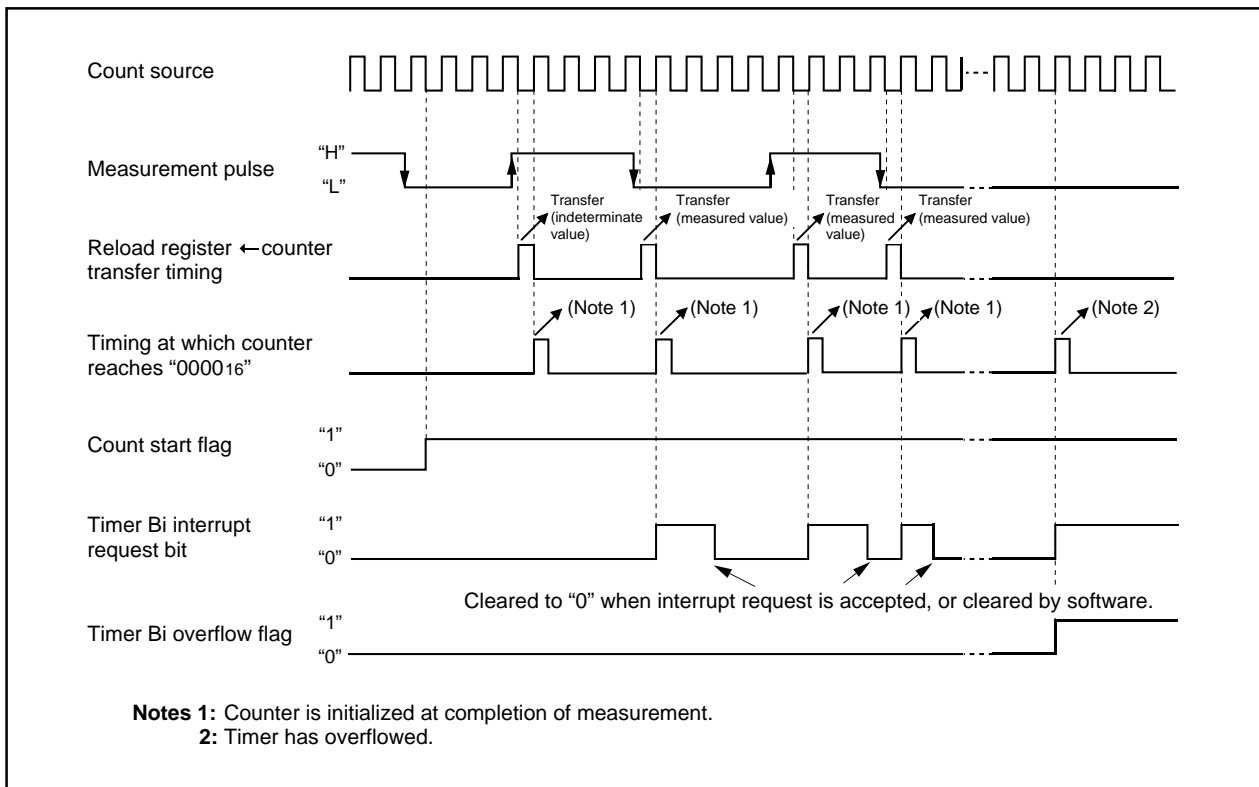
**Figure 2.10.23 Timer Bi mode register in pulse period/pulse width measurement mode (i = 0 to 2)**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.10.24** Operation timing when measuring a pulse period



**Figure 2.10.25** Operation timing when measuring a pulse width



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

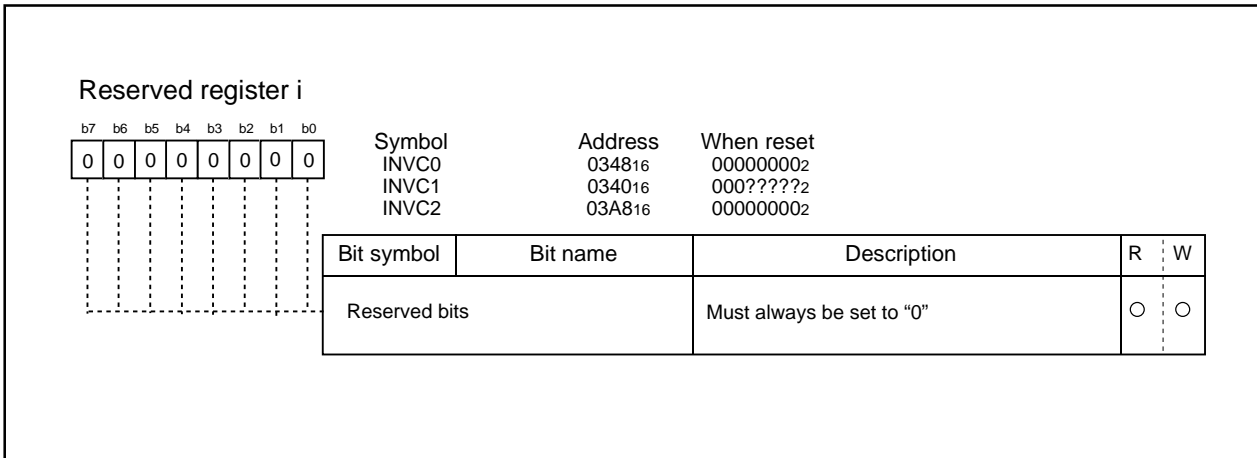


Figure 2.10.26 Reserved register i (i = 0 to 2)

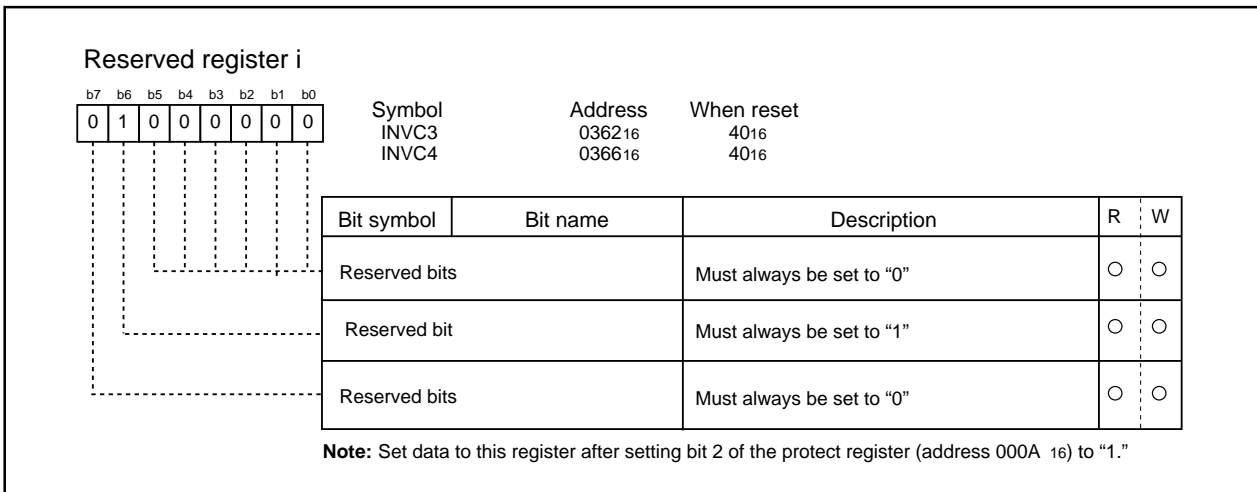


Figure 2.10.27 Reserved register i (i = 3 and 4)

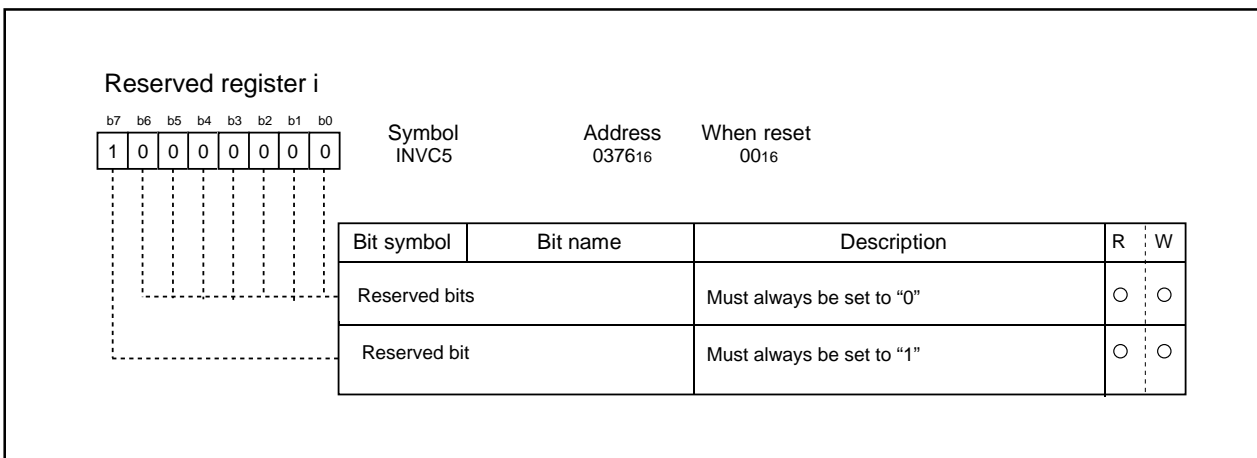


Figure 2.10.28 Reserved register i (i = 5)

**(4) TB0IN noise filter**

The input signal of pin TB0IN has the noise filter. The ON/OFF of noise filter and selection of filter clock are set by bits 2 to 4 of the peripheral mode register.

**Note:** When using the noise filter, set bit 7 of the peripheral mode register according to the main clock frequency.

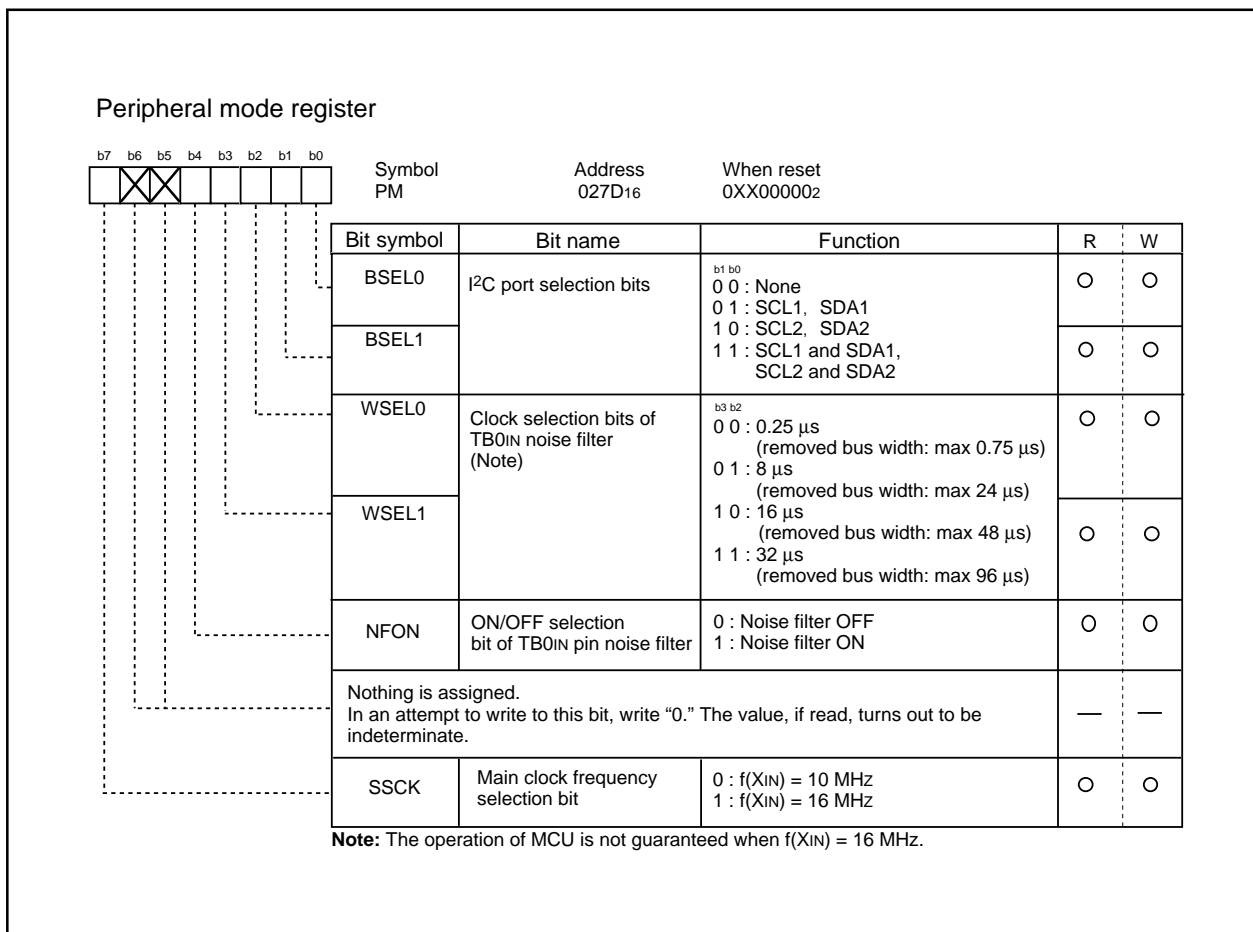


Figure 2.10.29 Peripheral mode register

## 2.11 Serial I/O

Serial I/O is configured as two units: UART0 and UART2.

### 2.11.1 UART0 and UART2

UART0 and UART2 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 2.11.1 shows the block diagram of UART0 and UART2. Figures 2.11.2 and 2.11.3 show the block diagram of the transmit/receive unit.

UART<sub>i</sub> (i = 0 and 2) has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode select bits (bits 0 to 2 at addresses 03A0<sub>16</sub> and 0378<sub>16</sub>) determine whether UART<sub>i</sub> is used as a clock synchronous serial I/O or as a UART. Although a few functions are different, UART0 and UART2 have almost the same functions.

UART0 and UART2 are almost equal in their functions with minor exceptions. UART2, in particular, is compliant with the SIM interface and I<sup>2</sup>C-BUS interface with some extra settings added in clock-synchronous serial I/O mode (Note). It also has the bus collision detection function that generates an interrupt request if the TxD pin and the RxD pin are different in level.

Table 2.11.1 shows the comparison of functions of UART0 and UART2, and Figures 2.11.4 to 2.11.14 show the registers related to UART<sub>i</sub>.

**Table 2.11.1 Comparison of functions of UART0 and UART2**

Function	UART0	UART2
CLK polarity selection	Possible (Note 1)	Possible (Note 1)
LSB first / MSB first selection	Possible (Note 1)	Possible (Note 2)
Continuous receive mode selection	Possible (Note 1)	Possible (Note 1)
Transfer clock output from multiple pins selection	Impossible	Impossible
Serial data logic switch	Impossible	Possible (Note 4)
Sleep mode selection	Possible (Note 3)	Impossible
TxD, RxD I/O polarity switch	Impossible	Possible
TxD, RxD port output format	CMOS output	N-channel open-drain output
Parity error signal output	Impossible	Possible (Note 4)
Bus collision detection	Impossible	Possible

**Notes 1:** Only when clock synchronous serial I/O mode.

**2:** Only when clock synchronous serial I/O mode and 8-bit UART mode.

**3:** Only when UART mode.

**4:** Using for SIM interface.

M306V0ME-XXXFP  
M306V0EEFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

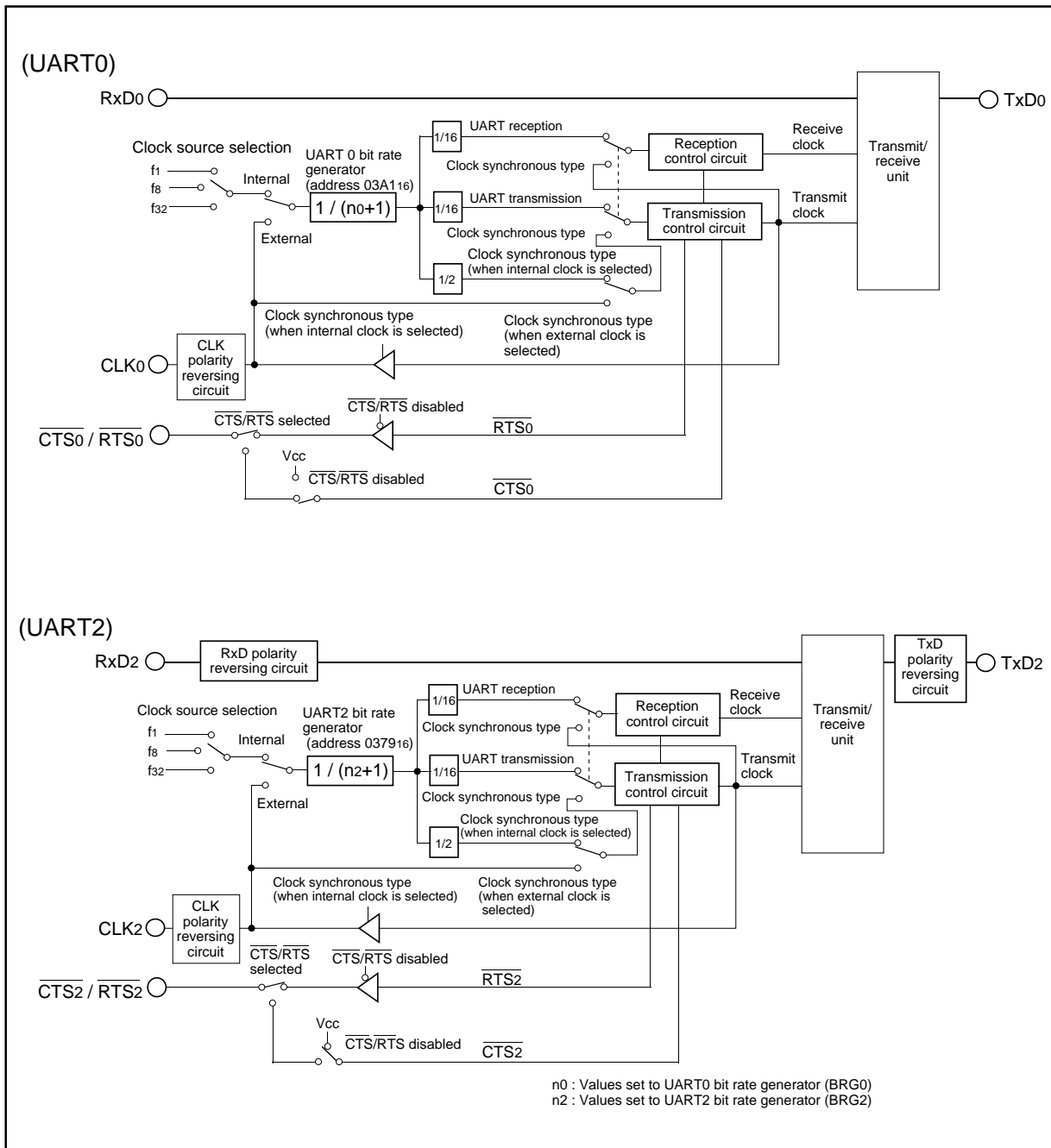


Figure 2.11.1 Block diagram of UART<sub>i</sub> (i = 0 and 2)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

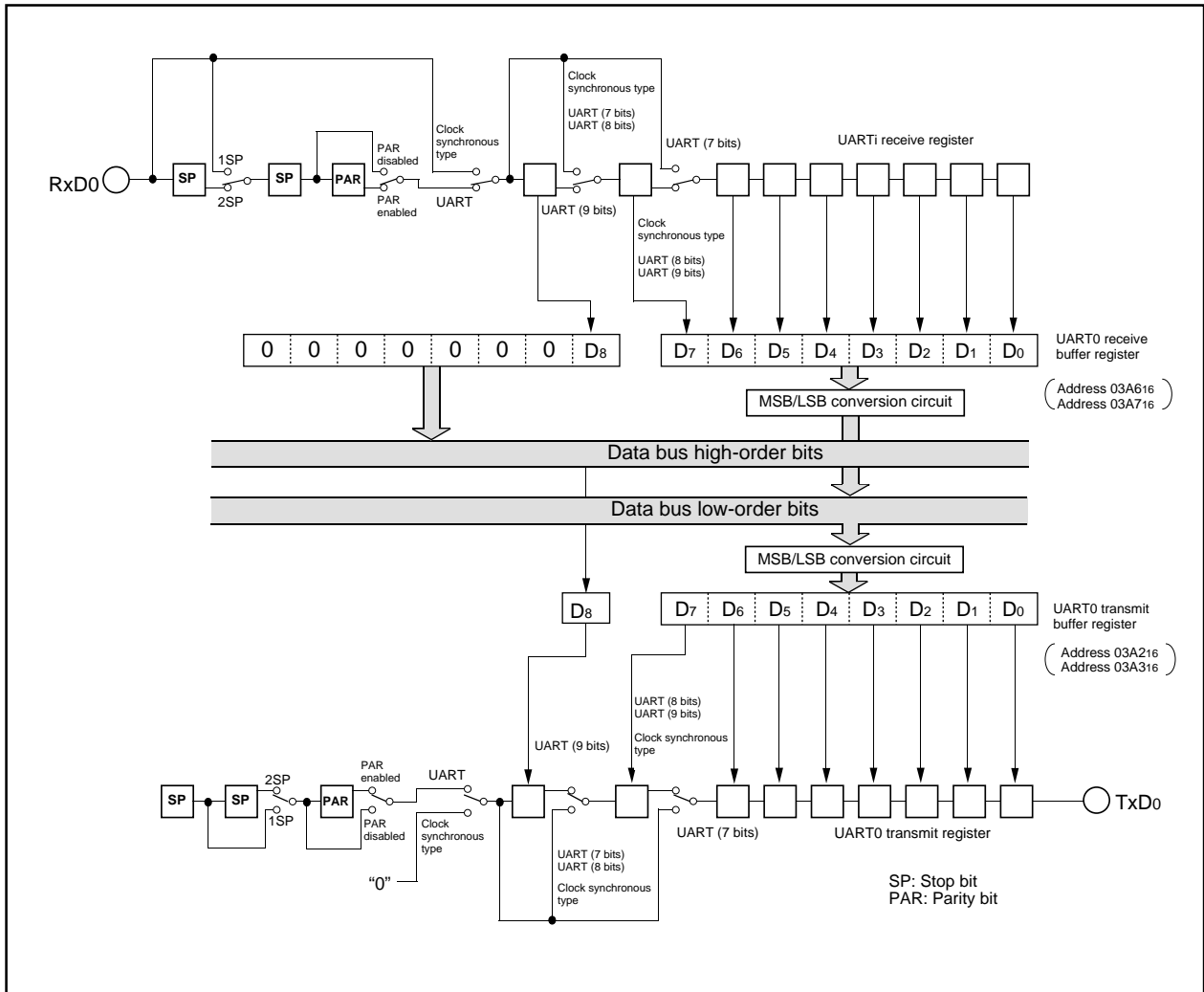


Figure 2.11.2 Block diagram of UART0 transmit/receive unit

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

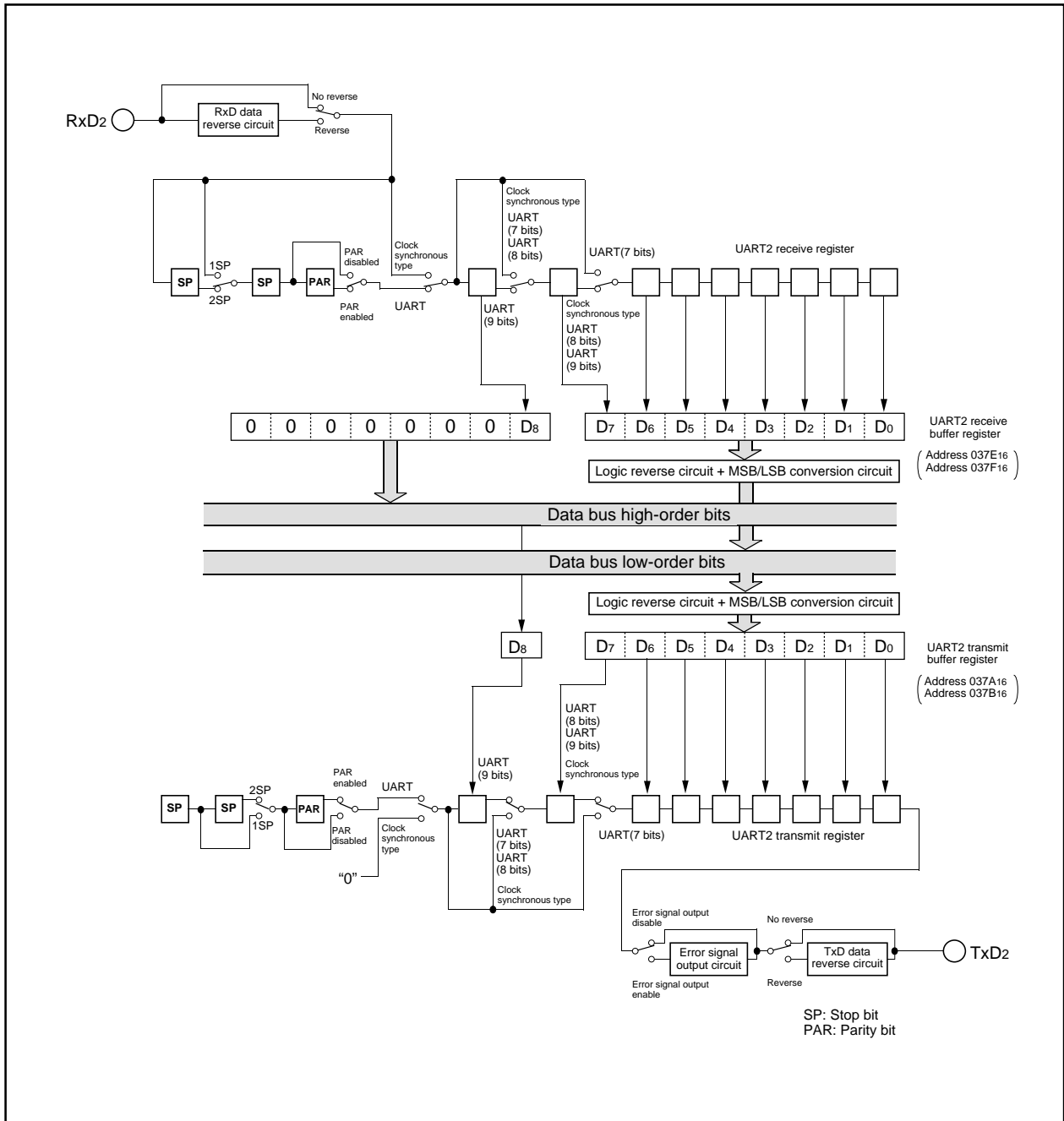
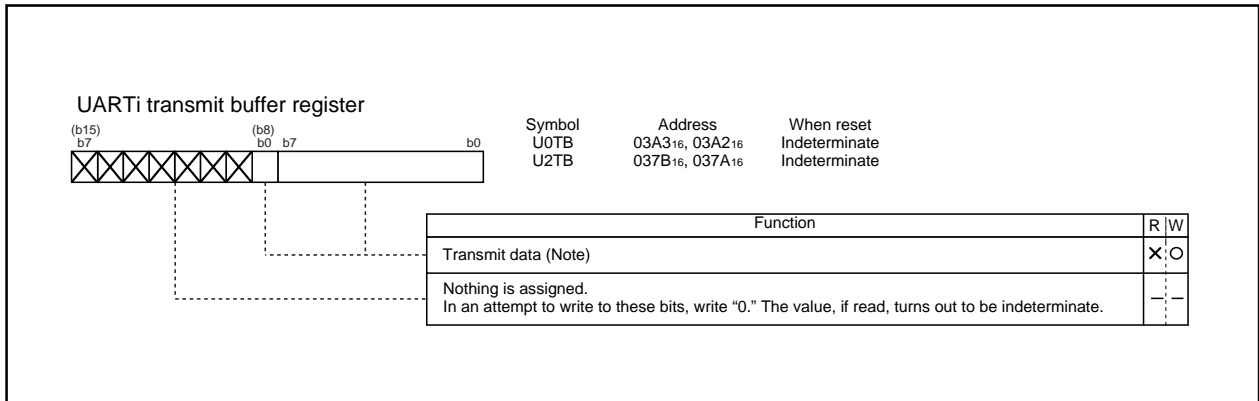


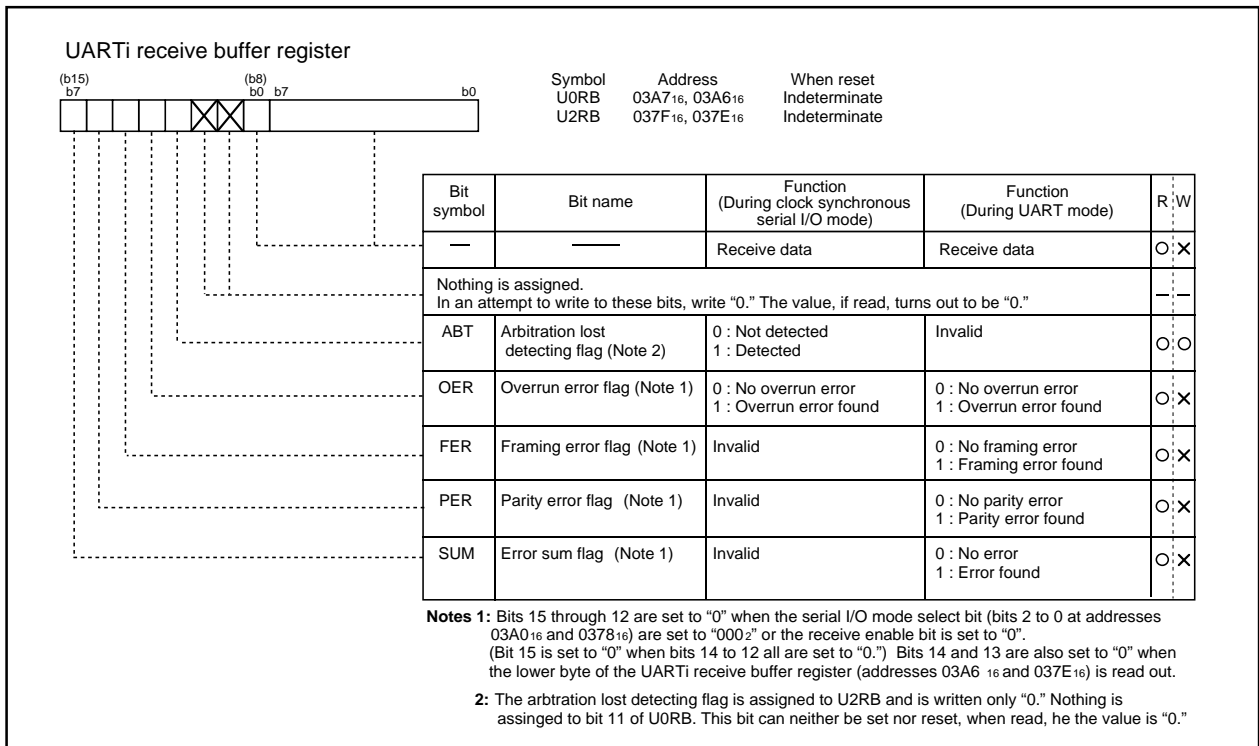
Figure 2.11.3 Block diagram of UART2 transmit/receive unit

**M306V0ME-XXXFP**  
**M306V0EEFP**

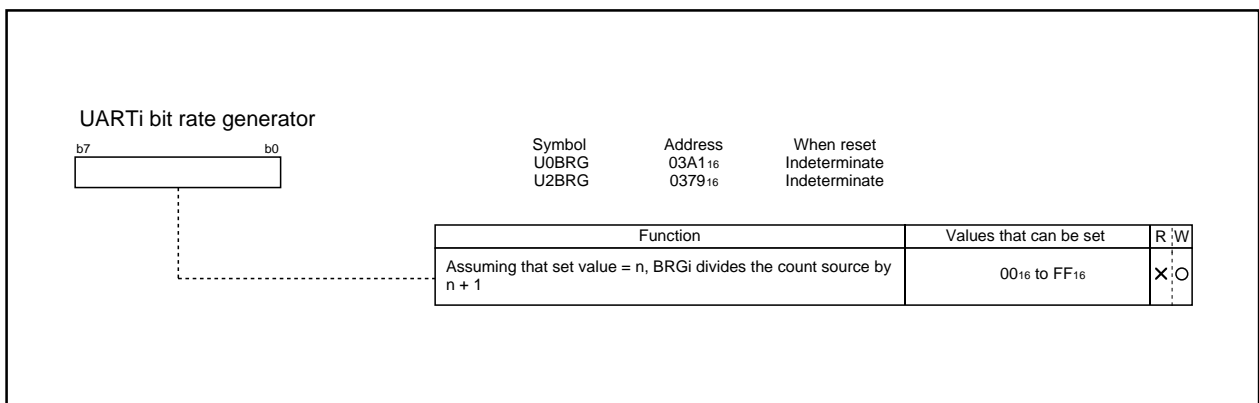
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.11.4 UARTi transmit buffer register (i = 0 and 2)**



**Figure 2.11.5 UARTi receive buffer register (i = 0 and 2)**



**Figure 2.11.6 UARTi bit rate generator (i = 0 and 2)**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

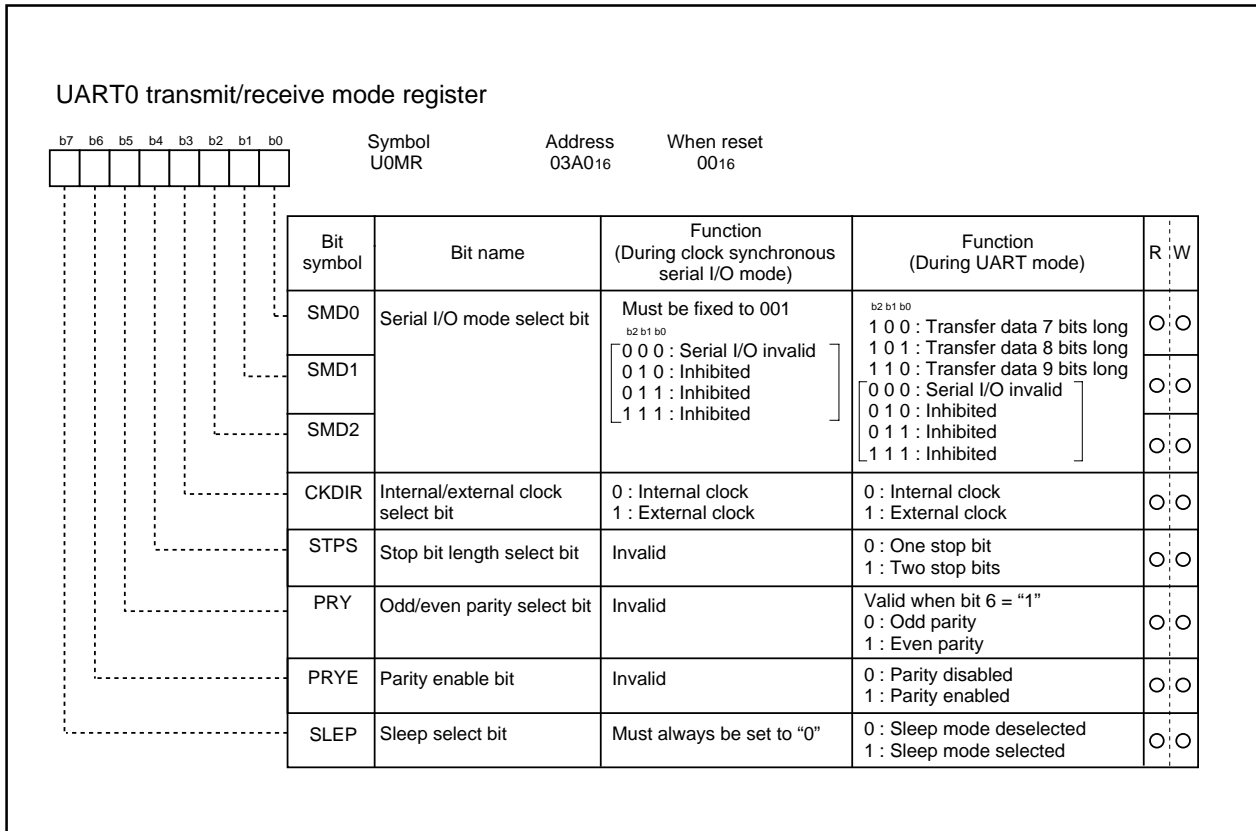


Figure 2.11.7 UART0 transmit/receive mode register

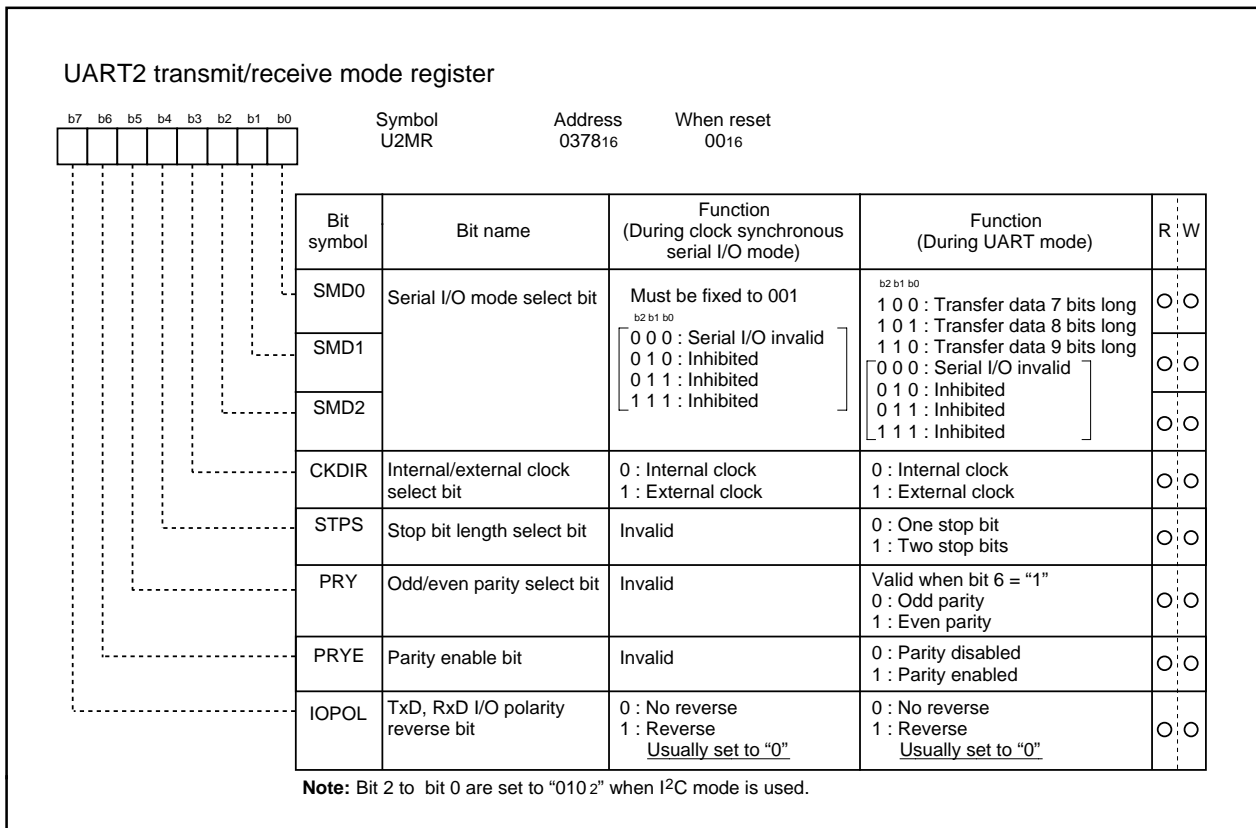


Figure 2.11.8 UART2 transmit/receive mode register



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

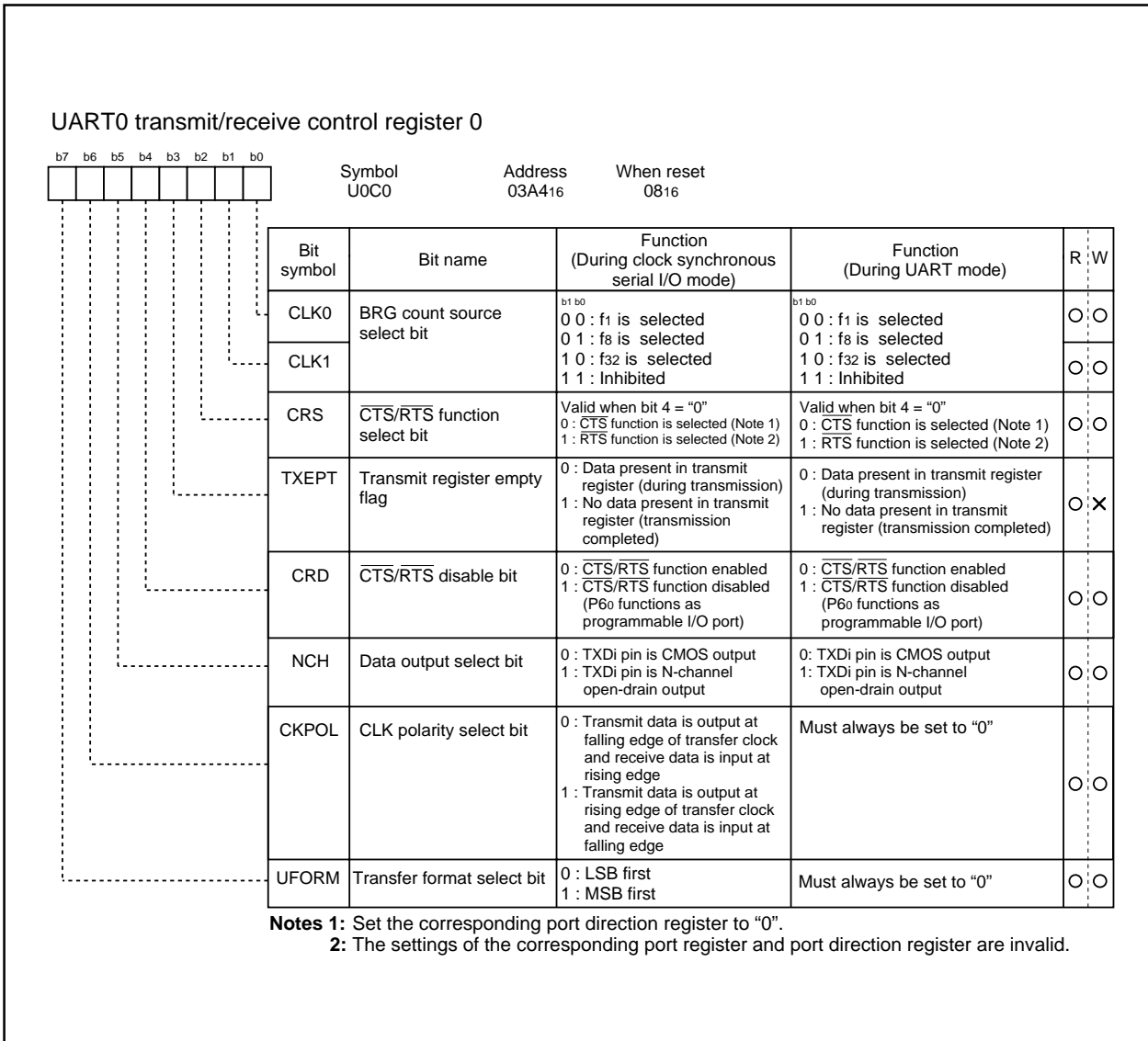


Figure 2.11.9 UART0 transmit/receive control register 0

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

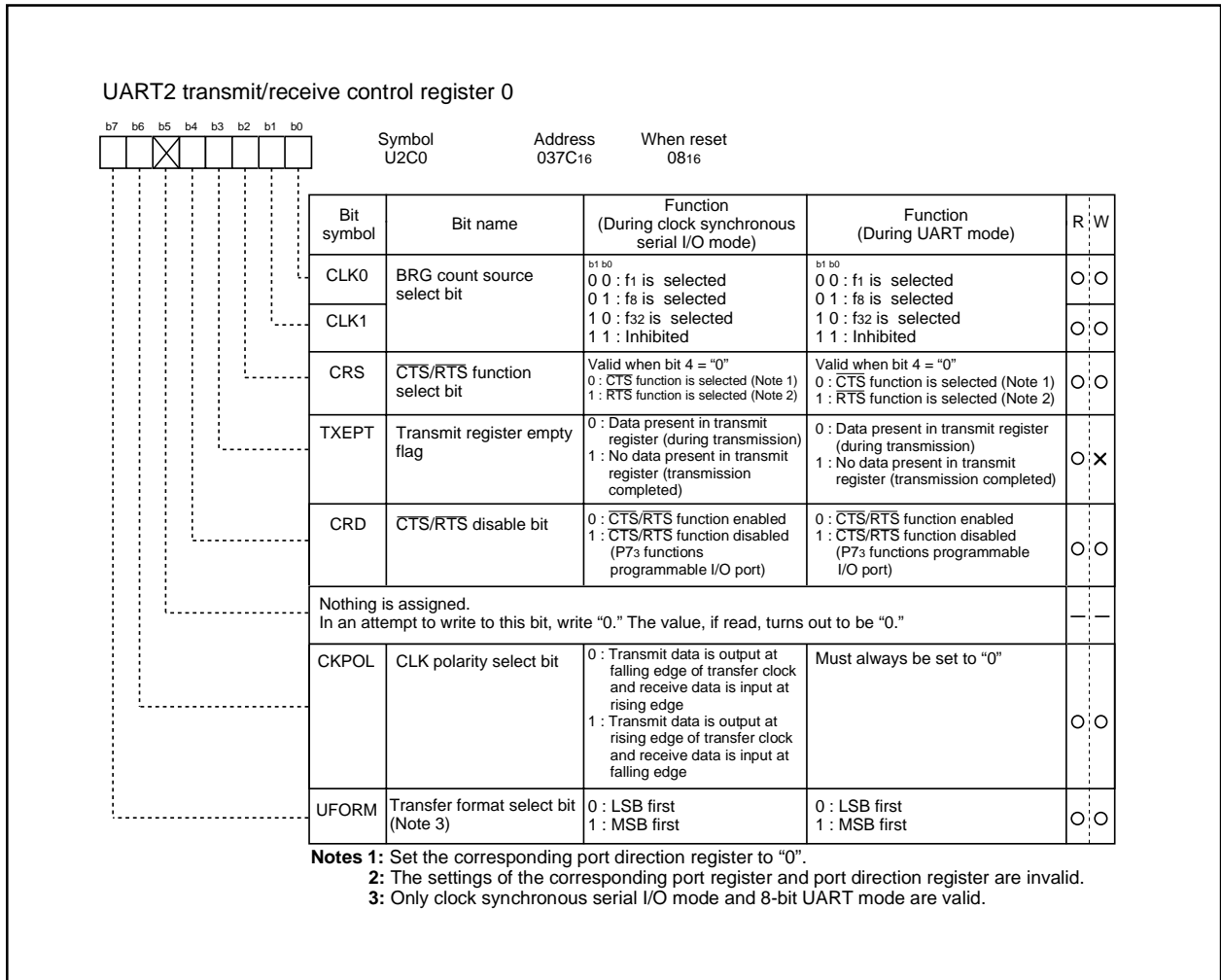


Figure 2.11.10 UART2 transmit/receive control register 0

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

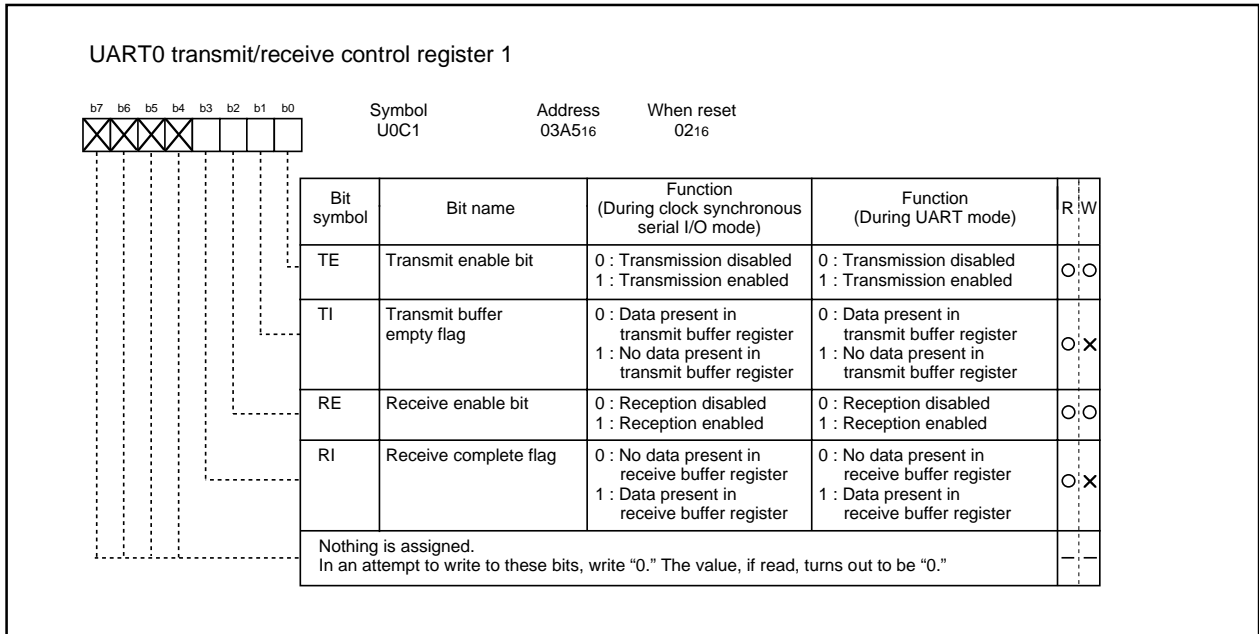


Figure 2.11.11 UART0 transmit/receive control register 1

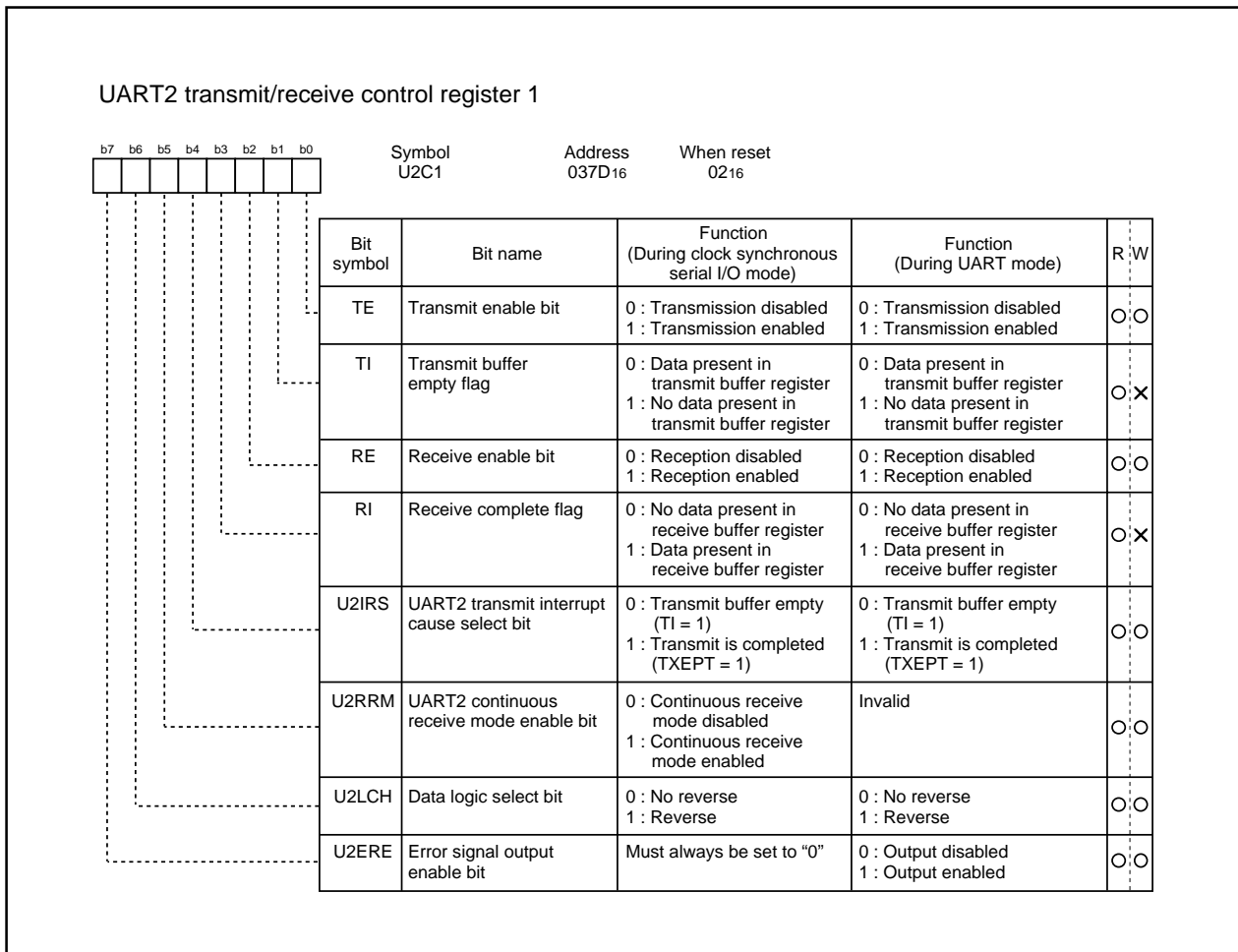


Figure 2.11.12 UART2 transmit/receive control register 1

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

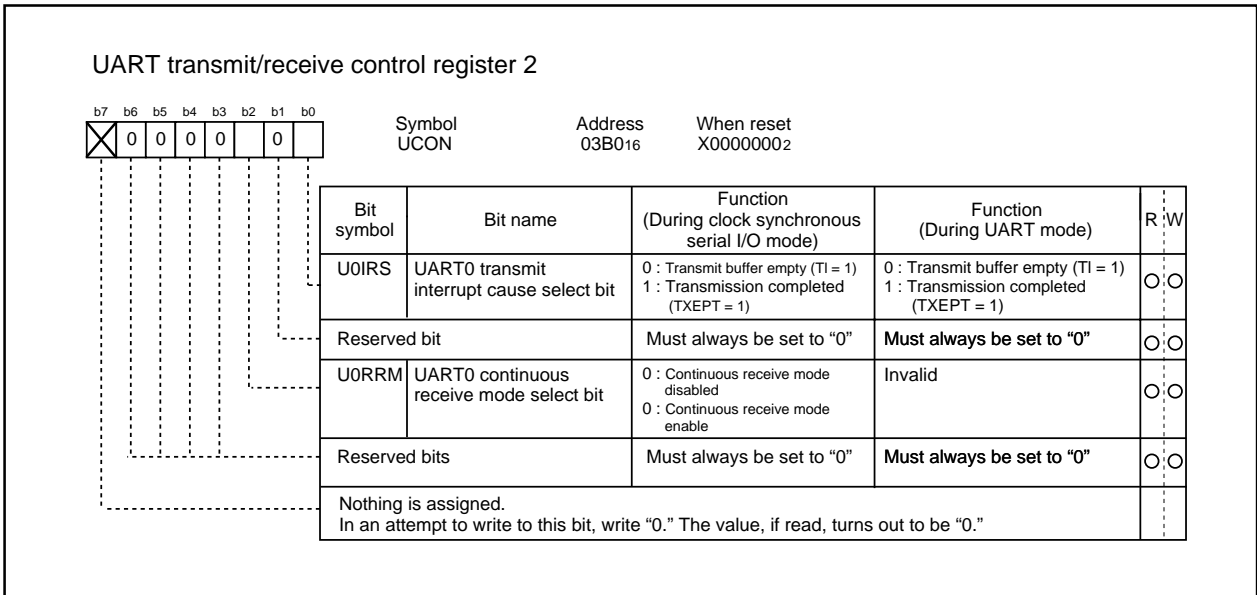


Figure 2.11.13 UART transmit/receive control register 2

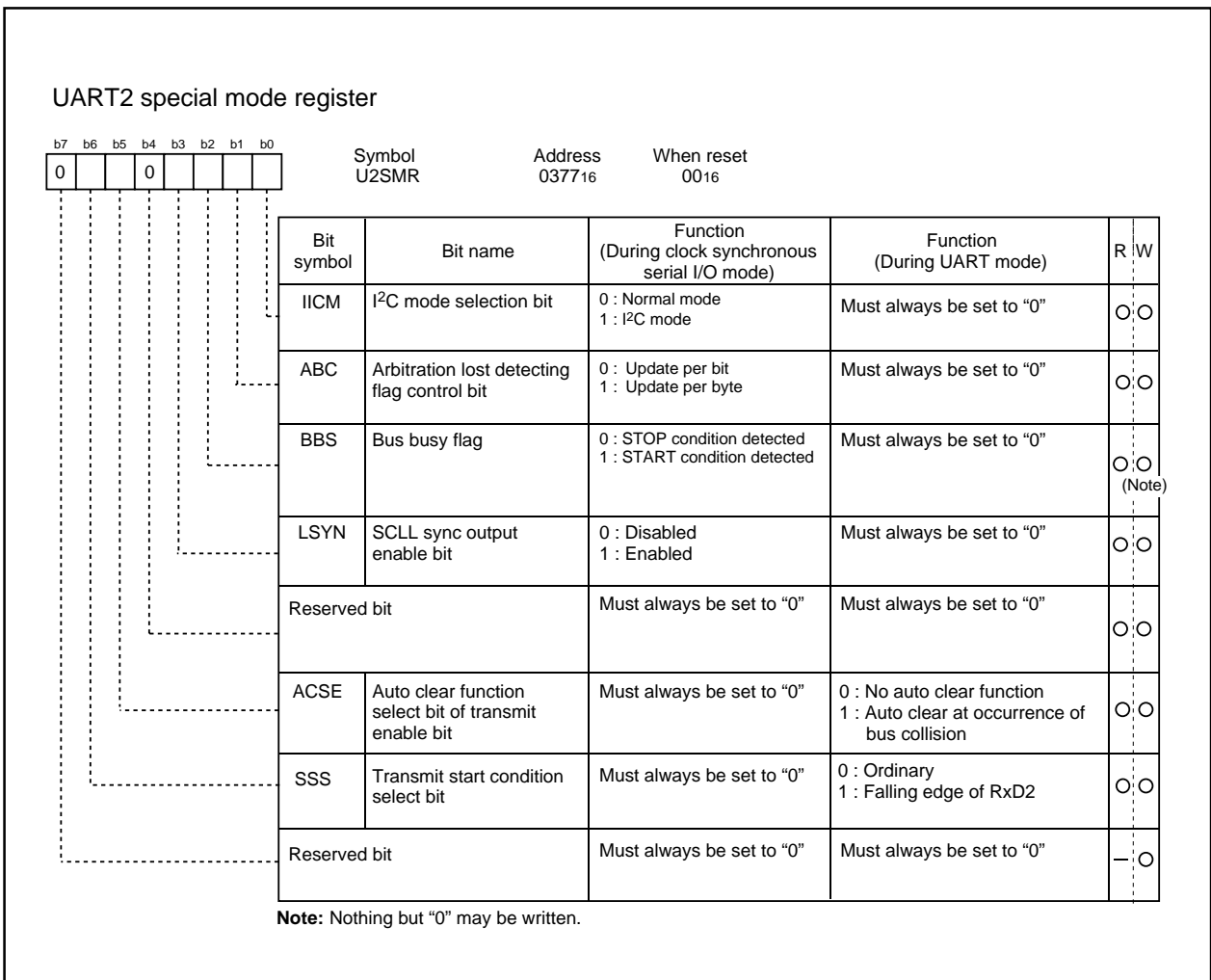


Figure 2.11.14 UART2 special mode register

### 2.11.2 Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables 2.11.2 and 2.11.3 list the specifications of the clock synchronous serial I/O mode. Figures 2.11.15 and 2.11.16 show the UARTi transmit/receive mode register in clock synchronous serial I/O mode.

**Table 2.11.2 Specifications of clock synchronous serial I/O mode (1)**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>When internal clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 0378<sub>16</sub> = "0") :  <math>f_i / 2(n+1)</math> (Note 1) <math>f_i = f_1, f_8, f_{32}</math></li> <li>When external clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 0378<sub>16</sub> = "1") :                      Input from CLKi pin</li> </ul>
Transmission/reception control	<ul style="list-style-type: none"> <li>CTS function/RTS function/<math>\overline{\text{CTS}}</math>, <math>\overline{\text{RTS}}</math> function chosen to be invalid</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>To start transmission, the following requirements must be met:                             <ul style="list-style-type: none"> <li>Transmit enable bit (bit 0 at addresses 03A5<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>Transmit buffer empty flag (bit 1 at addresses 03A5<sub>16</sub>, 037D<sub>16</sub>) = "0"</li> <li>When <math>\overline{\text{CTS}}</math> function selected, CTS input level = "L"</li> </ul> </li> <li>Furthermore, if external clock is selected, the following requirements must also be met:                             <ul style="list-style-type: none"> <li>CLKi polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 037C<sub>16</sub>) = "0":                                      CLKi input level = "H"</li> <li>CLKi polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 037C<sub>16</sub>) = "1":                                      CLKi input level = "L"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>To start reception, the following requirements must be met:                             <ul style="list-style-type: none"> <li>Receive enable bit (bit 2 at addresses 03A5<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>Transmit enable bit (bit 0 at addresses 03A5<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>Transmit buffer empty flag (bit 1 at addresses 03A5<sub>16</sub>, 037D<sub>16</sub>) = "0"</li> </ul> </li> <li>Furthermore, if external clock is selected, the following requirements must also be met:                             <ul style="list-style-type: none"> <li>CLKi polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 037C<sub>16</sub>) = "0":                                      CLKi input level = "H"</li> <li>CLKi polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 037C<sub>16</sub>) = "1":                                      CLKi input level = "L"</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When transmitting                             <ul style="list-style-type: none"> <li>Transmit interrupt cause select bit (bit 0 at address 03B0<sub>16</sub>, bit 4 at address 037D<sub>16</sub>) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed</li> <li>Transmit interrupt cause select bit (bit 0 at address 03B0<sub>16</sub>, bit 4 at address 037D<sub>16</sub>) = "1": Interrupts requested when data transmission from UARTi transfer register is completed</li> </ul> </li> <li>When receiving                             <ul style="list-style-type: none"> <li>Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error (Note 2)                      This error occurs when the next data is ready before contents of UARTi receive buffer register are read out</li> </ul>

**Table 2.11.3 Specifications of clock synchronous serial I/O mode (2)**

Item	Specification
Select function	<ul style="list-style-type: none"> <li>• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected</li> <li>• LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected</li> <li>• Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register</li> <li>• Switching serial data logic (UART2) Whether to reverse data in writing to the transmission buffer register or reading the reception buffer register can be selected.</li> <li>• TxD, RxD I/O polarity reverse (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.</li> </ul>

**Notes 1:** “n” denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UART bit rate generator.

**2:** If an overrun error occurs, the UART<sub>i</sub> receive buffer will have the next data written in. Note also that the UART<sub>i</sub> receive interrupt request bit is not set to “1”.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

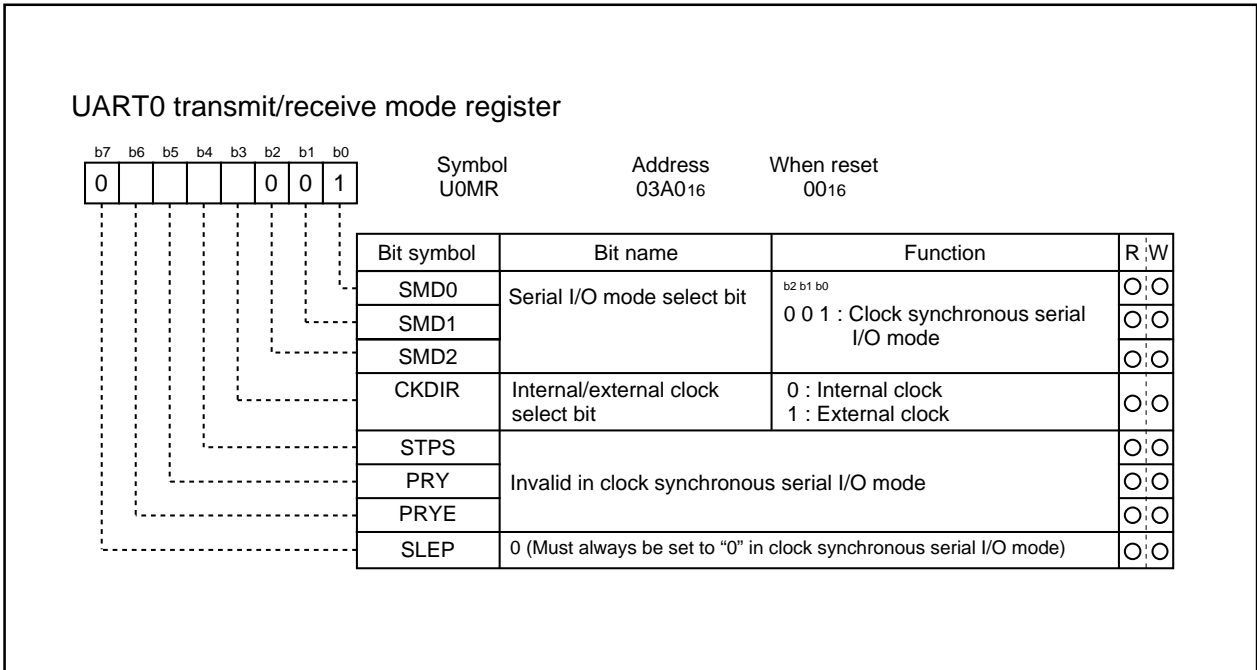


Figure 2.11.15 UART0 transmit/receive mode registers in clock synchronous serial I/O mode

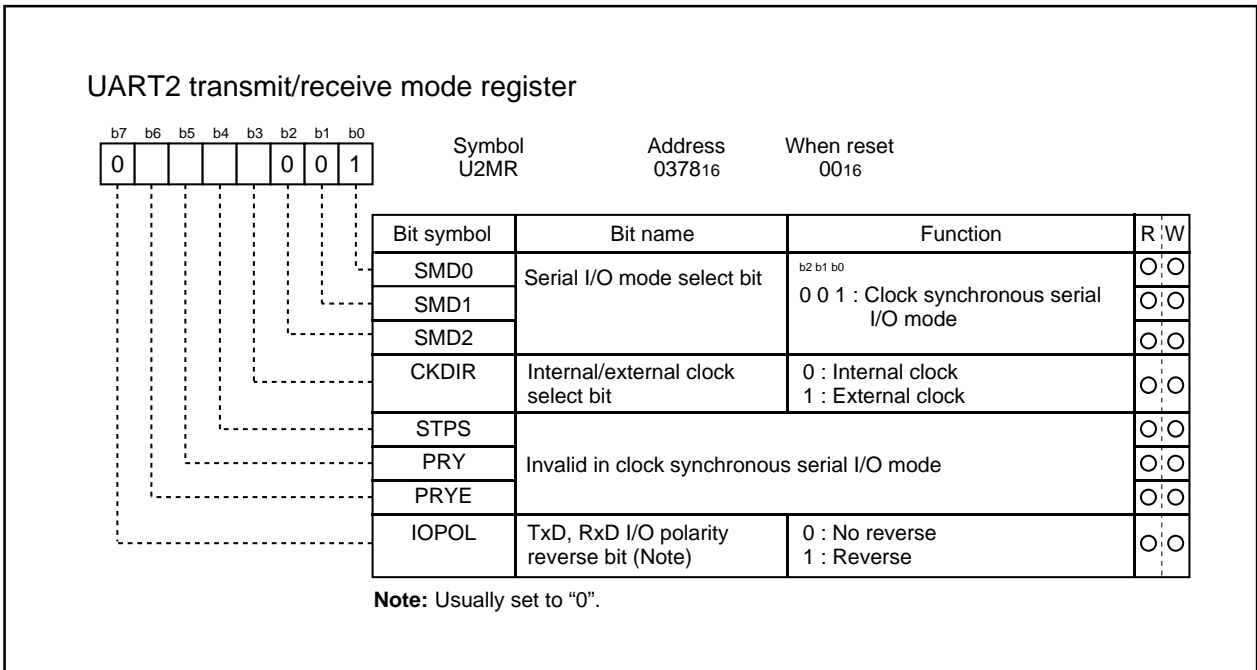


Figure 2.11.16 UART2 transmit/receive mode register in clock synchronous serial I/O mode

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

Table 2.11.4 lists the functions of the input/output pins 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 a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

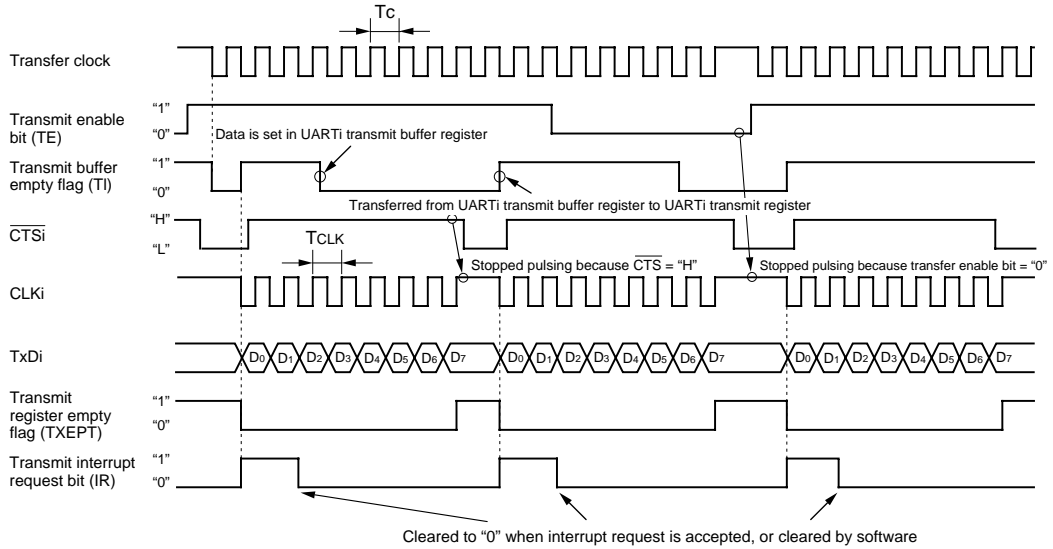
**Table 2.11.4 Input/output pin functions in clock synchronous serial I/O mode**

Pin name	Function	Method of selection
TxDi (P63, P70)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P62, P71)	Serial data input	Port P62 and P71 direction register (bits 2 at address 03EE 16, bit 1 at address 03EF 16) = "0" (Can be used as an input port when performing transmission only)
CLKi (P61, P72)	Transfer clock output	Internal/external clock select bit (bit 3 at address 03A0 16, 0378 16) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0 16, 0378 16) = "1" Port P61 and P72 direction register (bits 1 at address 03EE 16, bit 2 at address 03EF 16) = "0"
$\overline{\text{CTS}}/\overline{\text{RTS}}_i$ (P60, P73)	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4 16, 037C 16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit (bit 2 at addresses 03A4 16, 037C 16) = "0" Port P60 and P73 direction register (bits 0 at address 03EE 16, bit 3 at address 03EF 16) = "0"
	$\overline{\text{RTS}}$ output	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4 16, 037C 16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4 16, 037C 16) = "1"
	Programmable I/O port	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4 16, 037C 16) = "1"

(when transfer clock output from multiple pins and separate  $\overline{\text{CTS}}/\overline{\text{RTS}}$  pins functions are not selected)



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



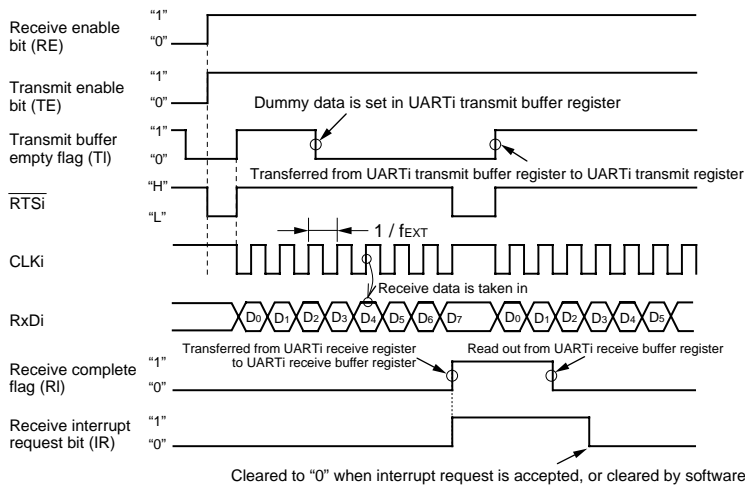
The above timing applies to the following settings:

- Internal clock is selected.
- CTS function is selected.
- CLK polarity select bit = "0".
- Transmit interrupt cause select bit = "0".

$$T_c = \frac{T_{CLK}}{2(n+1)} \cdot f_i$$

f<sub>i</sub>: frequency of BRGi count source (f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub>)  
n: value set to BRGi

• Example of receive timing (when external clock is selected)



The above timing applies to the following settings:

- External clock is selected.
- RTS function is selected.
- CLK polarity select bit = "0".

fEXT: frequency of external clock

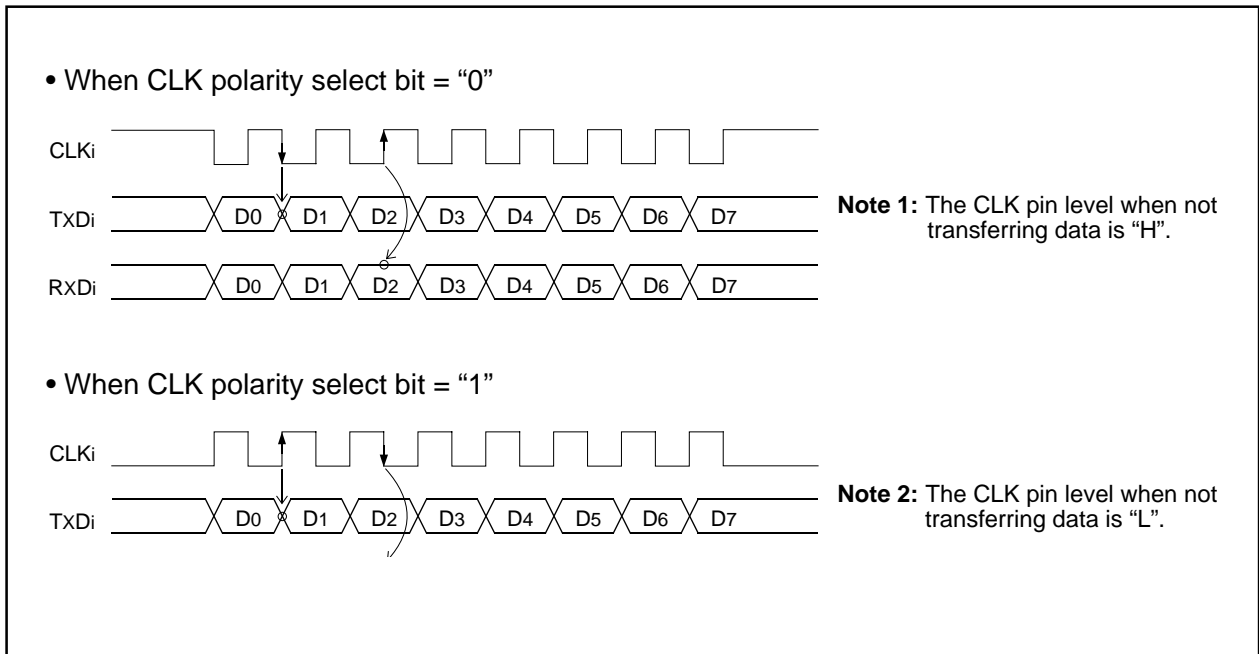
Meet the following conditions are met when the CLK input before data reception = "H"

- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UARTi transmit buffer register

Figure 2.11.17 Typical transmit/receive timings in clock synchronous serial I/O mode

**(1) Polarity select function**

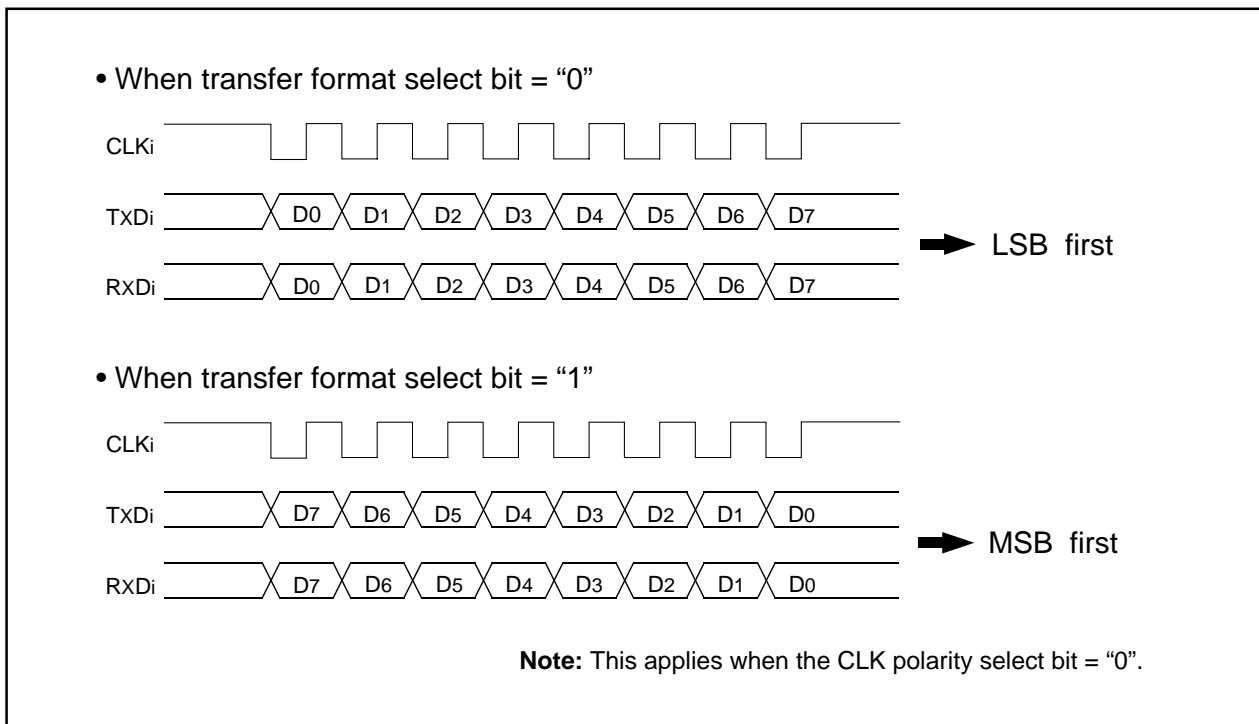
As shown in Figure 2.11.18, the CLK polarity select bit (bit 6 at addresses 03A416, 037C16) allows selection of the polarity of the transfer clock.



**Figure 2.11.18 Polarity of transfer clock**

**(2) LSB first/MSB first select function**

As shown in Figure 2.11.19, when the transfer format select bit (bit 7 at addresses 03A416, 037C16) = "0", the transfer format is "LSB first"; when the bit = "1", the transfer format is "MSB first".



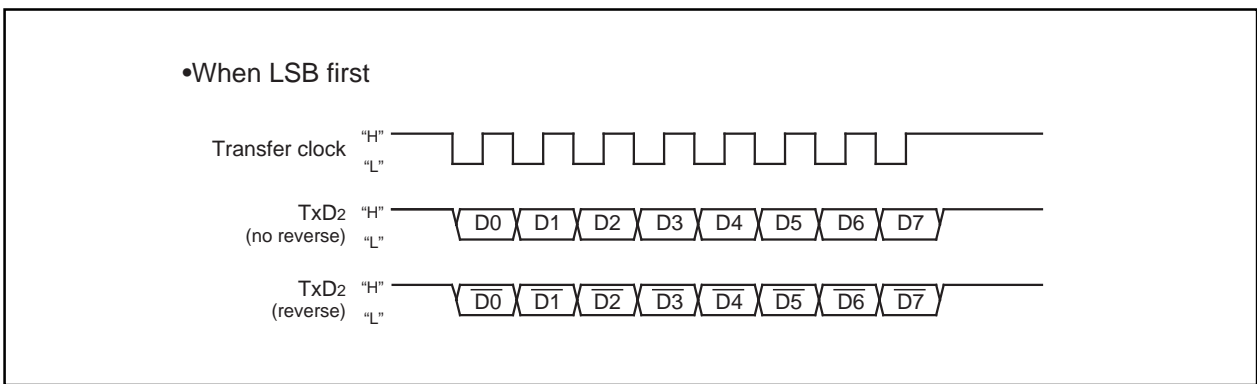
**Figure 2.11.19 Transfer format**

**(3) Continuous receive mode**

If the continuous receive mode enable bit (bits 2 at address 03B0<sub>16</sub>, bit 5 at address 037D<sub>16</sub>) is set to "1", the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

**(4) Serial data logic switch function (UART2)**

When the data logic select bit (bit6 at address 037D<sub>16</sub>) = "1", and writing to transmit buffer register or reading from receive buffer register, data is reversed. Figure 2.11.20 shows the example of serial data logic switch timing.



**Figure 2.11.20 Serial data logic switch timing**

### 2.11.3 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 2.11.5 and 2.11.6 list the specifications of the UART mode. Figure 2.11.21 and 2.11.22 show the UARTi transmit/receive mode register in UART mode.

**Table 2.11.5 Specifications of UART Mode (1)**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected</li> <li>• Start bit: 1 bit</li> <li>• Parity bit: Odd, even, or nothing as selected</li> <li>• Stop bit: 1 bit or 2 bits as selected</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• When internal clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 0378<sub>16</sub> = "0") : f<sub>i</sub>/16(n+1) (Note 1) f<sub>i</sub> = f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub></li> <li>• When external clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 0378<sub>16</sub> = "1") : f<sub>EXT</sub>/16(n+1)(Note 1) (Note 2)</li> </ul>
Transmission/reception control	<ul style="list-style-type: none"> <li>• CTS function/RTS function/CTS, RTS function chosen to be invalid</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>• To start transmission, the following requirements must be met:                             <ul style="list-style-type: none"> <li>- Transmit enable bit (bit 0 at addresses 03A5<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>- Transmit buffer empty flag (bit 1 at addresses 03A5<sub>16</sub>, 037D<sub>16</sub>) = "0"</li> <li>- When CTS function selected, <math>\overline{\text{CTS}}</math> input level = "L"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>• To start reception, the following requirements must be met:                             <ul style="list-style-type: none"> <li>- Receive enable bit (bit 2 at addresses 03A5<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>- Start bit detection</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When transmitting                             <ul style="list-style-type: none"> <li>- Transmit interrupt cause select bits (bits 0 at address 03B0<sub>16</sub>, bit4 at address 037D<sub>16</sub>) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed</li> <li>- Transmit interrupt cause select bits (bits 0 at address 03B0<sub>16</sub>, bit4 at address 037D<sub>16</sub>) = "1": Interrupts requested when data transmission from UARTi transfer register is completed</li> </ul> </li> <li>• When receiving                             <ul style="list-style-type: none"> <li>- Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error (Note 3) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out</li> <li>• Framing error This error occurs when the number of stop bits set is not detected</li> <li>• Parity error This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set</li> <li>• Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered</li> </ul>

**Table 2.11.6 Specifications of UART Mode (2)**

Item	Specification
Select function	<ul style="list-style-type: none"> <li data-bbox="507 405 895 434">• Sleep mode selection (UART0) This mode is used to transfer data to and from one of multiple slave micro-computers</li> <li data-bbox="507 524 911 553">• Serial data logic switch (UART2) This function is reversing logic value of transferring data. Start bit, parity bit and stop bit are not reversed.</li> <li data-bbox="507 642 863 672">• TxD, RxD I/O polarity switch This function is reversing TxD port output and RxD port input. All I/O data level is reversed.</li> </ul>

**Notes 1:** 'n' denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UART<sub>i</sub> bit rate generator.

**2:** f<sub>EXT</sub> is input from the CLK<sub>i</sub> pin.

**3:** If an overrun error occurs, the UART<sub>i</sub> receive buffer will have the next data written in. Note also that the UART<sub>i</sub> receive interrupt request bit is not set to "1".

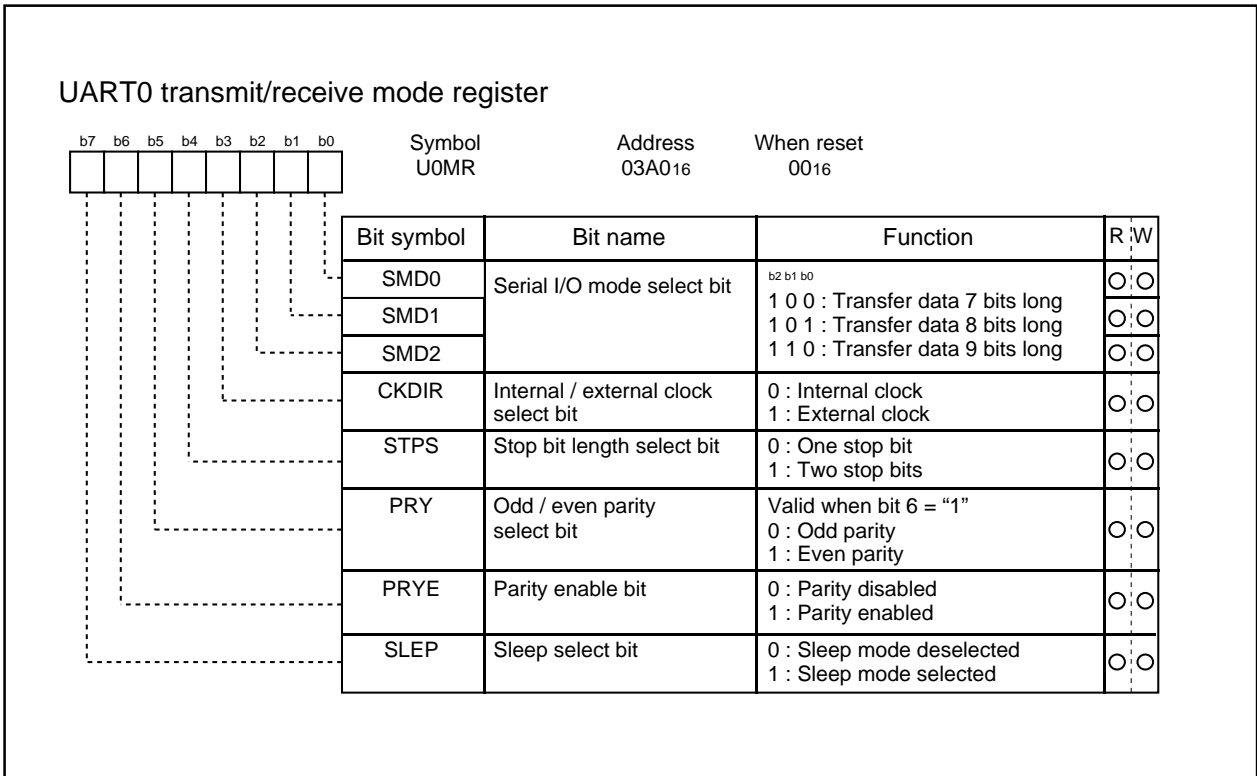


Figure 2.11.21 UART0 transmit/receive mode register in UART mode

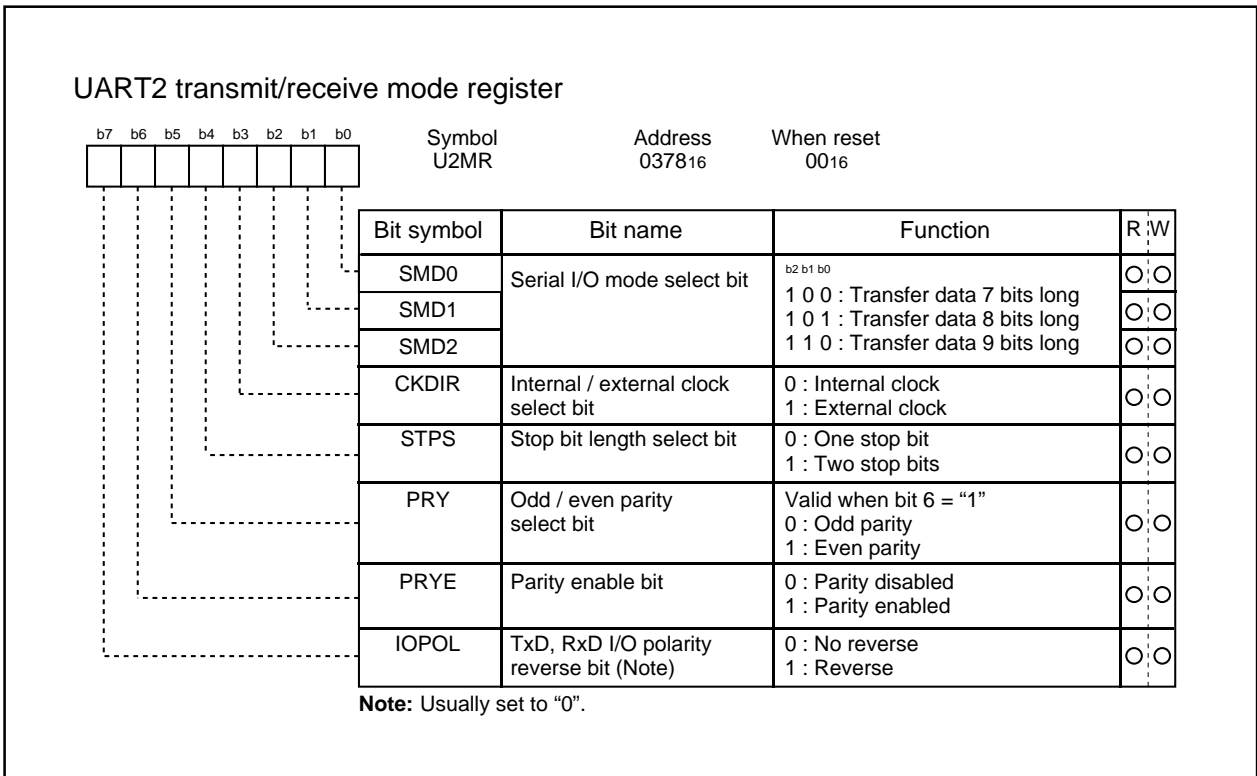


Figure 2.11.22 UART2 transmit/receive mode register in UART mode

# M306V0ME-XXXFP

# M306V0EEFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

Table 2.11.7 lists the functions of the input/output pins during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

**Table 2.11.7 Input/output pin functions in UART mode**

Pin name	Function	Method of selection
TxDi (P63, P70)	Serial data output	
RxDi (P62, P71)	Serial data input	Port P62 and P71 direction register (bit 2 at address 03EE 16, bit 1 at address 03EF 16) = "0" (Can be used as an input port when performing transmission only)
CLKi (P61, P72)	Programmable I/O port	Internal/external clock select bit (bit 3 at address 03A0 16, 0378 16) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0 16, 0378 16) = "1" Port P61 and P72 direction register (bit 1 at address 03EE 16, bit 2 at address 03EF 16) = "0"
$\overline{\text{CTS}}/\overline{\text{RTS}}$ (P60, P73)	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4 16, 037C 16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4 16, 037C 16) = "0" Port P60 and P73 direction register (bit 0 at address 03EE 16, bit 3 at address 03EF 16) = "0"
	$\overline{\text{RTS}}$ output	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4 16, 037C 16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4 16, 037C 16) = "1"
	Programmable I/O port	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4 16, 037C 16) = "1"

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

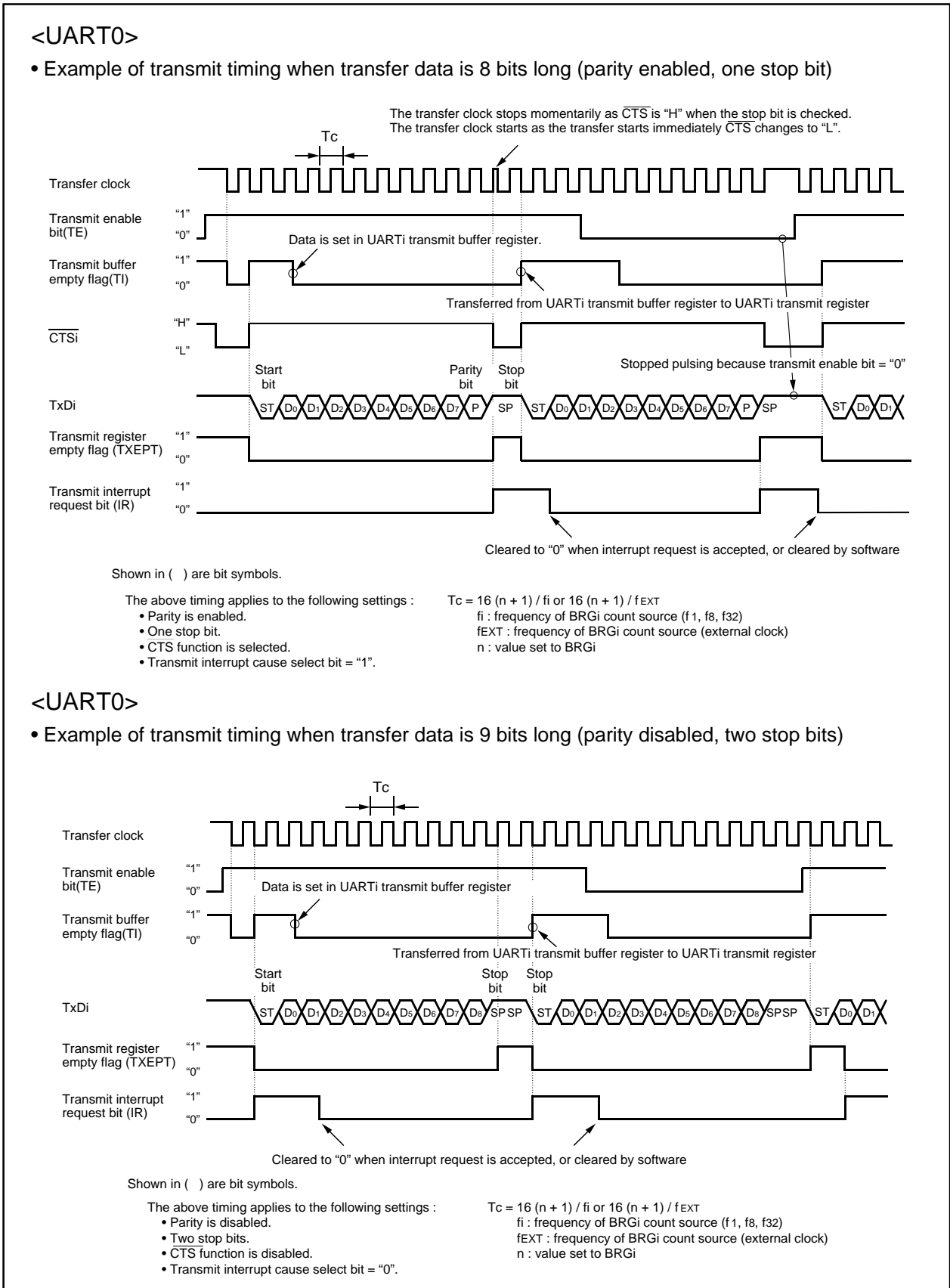
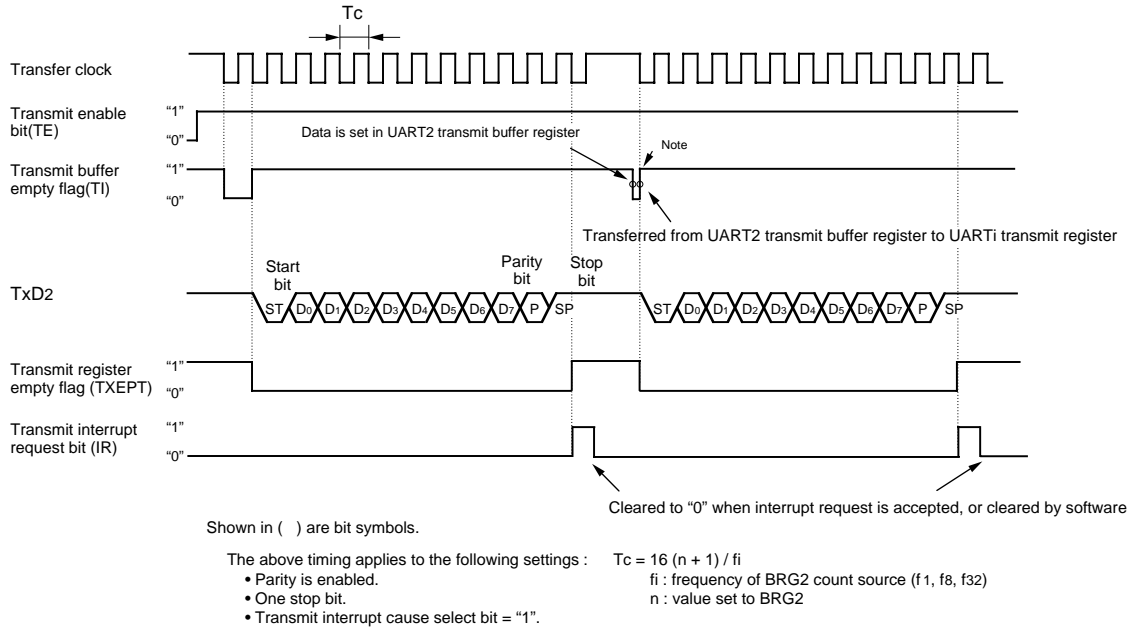


Figure 2.11.23 Typical transmit/receive timings in UART mode



<UART2>

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



**Note:** The transmit is started with overflow timing of BRG after having written in a value at the transmit buffer in the above timing.

<UART2, UART0>

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

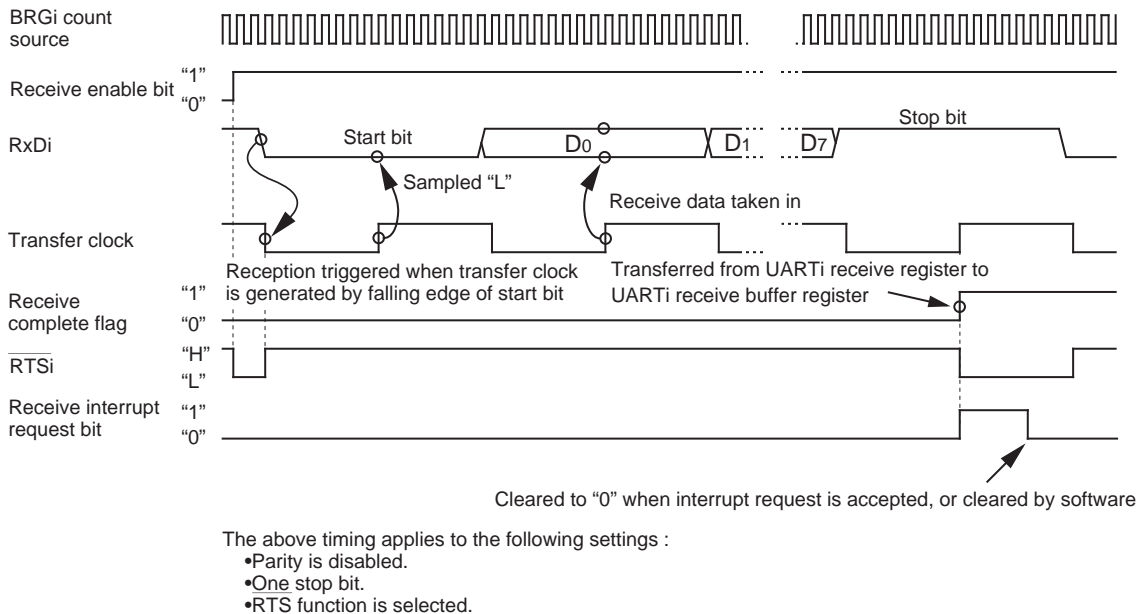


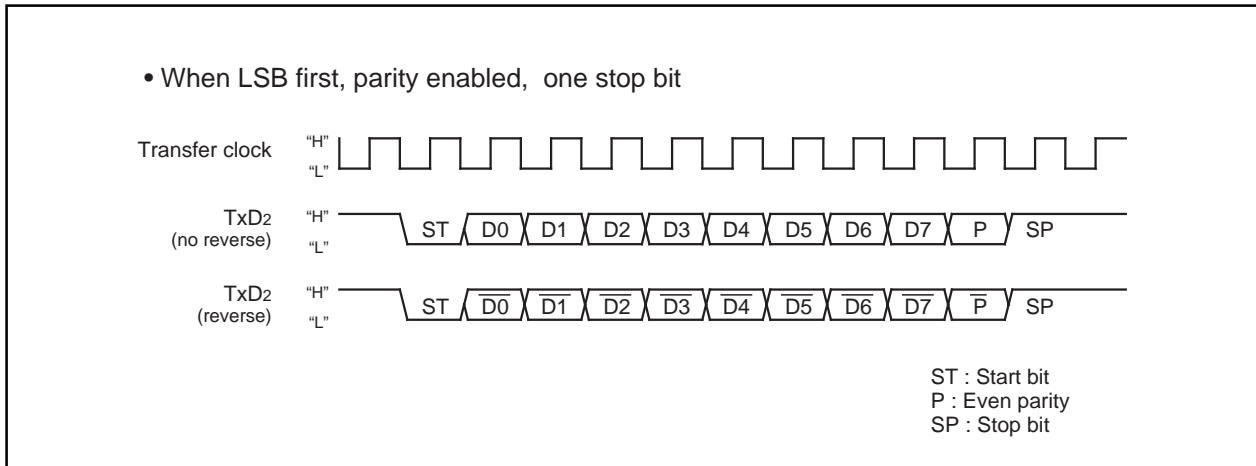
Figure 2.11.23 Typical transmit/receive timings in UART mode

**(1) Sleep mode (UART0)**

This mode is used to transfer data between specific microcomputers among multiple microcomputers connected using UART0. The sleep mode is selected when the sleep select bit (bit 7 at address 03A0<sub>16</sub>) is set to "1" during reception. In this mode, the unit performs receive operation when the MSB of the received data = "1" and does not perform receive operation when the MSB = "0".

**(2) Function for switching serial data logic (UART2)**

When the data logic select bit (bit 6 of address 037D<sub>16</sub>) is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register. Figure 2.11.24 shows the example of timing for switching serial data logic.



**Figure 2.11.24 Timing for switching serial data logic**

**(3) TxD, RxD I/O polarity reverse function (UART2)**

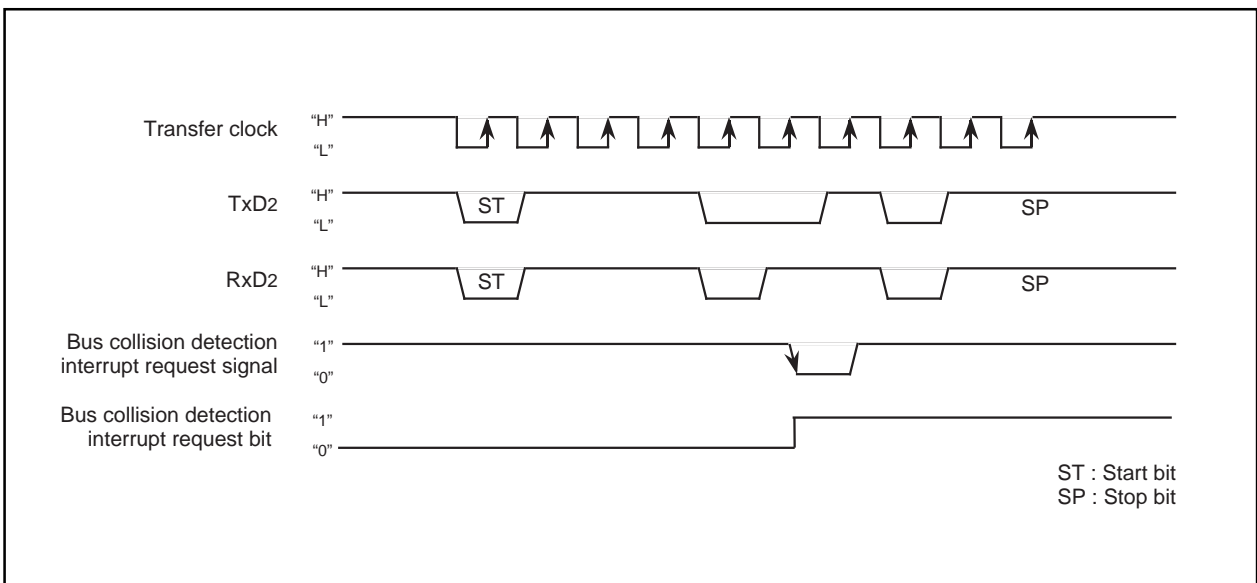
This function is to reverse TxD pin output and RxD pin input. The level of any data to be input or output (including the start bit, stop bit(s), and parity bit) is reversed. Set this function to "0" (not to reverse) for usual use.

**(4) Bus collision detection function and other functions (UART2)**

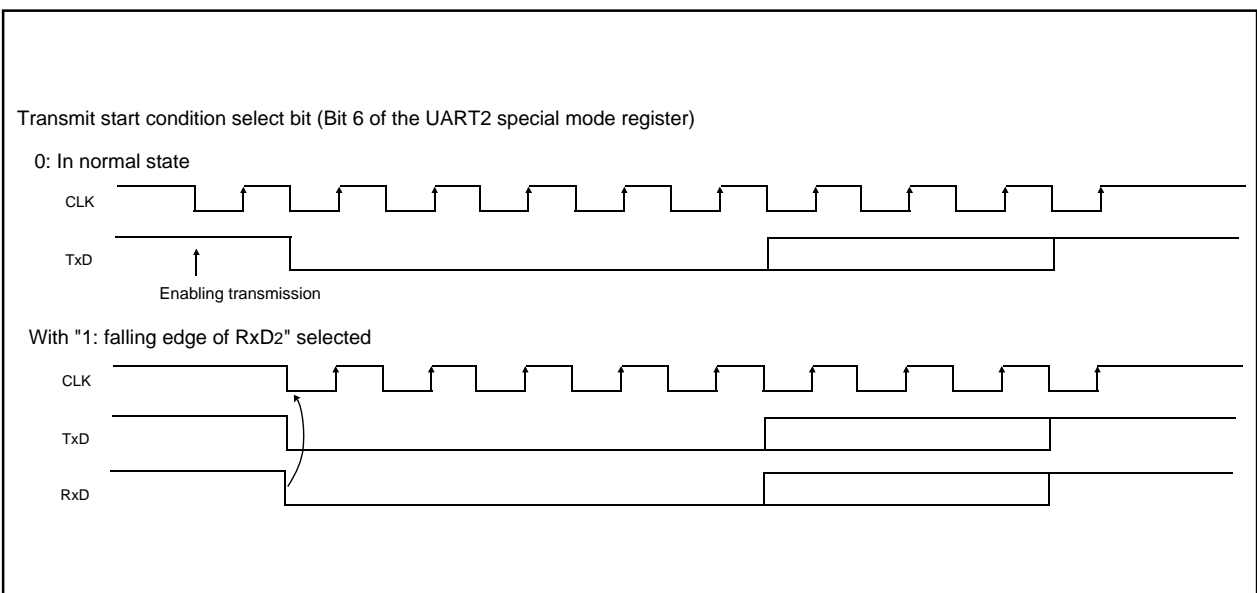
This function is to sample the output level of the TxD pin and the input level of the RxD pin at the rising edge of the transfer clock; if their values are different, then an interrupt request occurs. Figure 2.11.26 shows the example of detection timing of a buss collision (in UART mode).

And also, bit 5 of the special UART2 mode register is used as the selection bit for auto clear function select bit of enable bit. Setting this bit to "1" automatically resets the transmit enable bit to "0" when "1" is set in the bus collision detection interrupt request bit (nonconformity) (refer to Figure 2.11.25).

Bit 6 of the special UART2 mode register is used as the transmit start condition select bit. Setting this bit to "1" starts the TxD transmission in synchronization with the falling edge of the RxD terminal (refer to Figure 2.11.26).



**Figure 2.11.25 Detection timing of a bus collision (in UART mode)**



**Figure 2.11.26 Some other functions**

**2.11.4 Clock-asynchronous serial I/O mode (compliant with the SIM interface)**

The SIM interface is used for connecting the microcomputer with a memory card I/C or the like; adding some extra settings in UART2 clock-asynchronous serial I/O mode allows the user to effect this function. Tables 2.11.8 and 2.11.9 show the specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface).

**Table 2.11.8 Specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface) (1)**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Transfer data 8-bit UART mode (bit 2 through bit 0 of address 0378<sub>16</sub> = "1012")</li> <li>• One stop bit (bit 4 of address 0378<sub>16</sub> = "0")</li> <li>• With the direct format chosen               <ul style="list-style-type: none"> <li>Set parity to "even" (bit 5 and bit 6 of address 0378<sub>16</sub> = "1" and "1" respectively)</li> <li>Set data logic to "direct" (bit 6 of address 037D<sub>16</sub> = "0").</li> <li>Set transfer format to LSB (bit 7 of address 037C<sub>16</sub> = "0").</li> </ul> </li> <li>• With the inverse format chosen               <ul style="list-style-type: none"> <li>Set parity to "odd" (bit 5 and bit 6 of address 0378<sub>16</sub> = "0" and "1" respectively)</li> <li>Set data logic to "inverse" (bit 6 of address 037D<sub>16</sub> = "1")</li> <li>Set transfer format to MSB (bit 7 of address 037C<sub>16</sub> = "1")</li> </ul> </li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• With the internal clock chosen (bit 3 of address 0378<sub>16</sub> = "0") : <math>f_i / 16 (n + 1)</math> (Note 1) : <math>f_i = f_1, f_8, f_{32}</math></li> <li>• With an external clock chosen (bit 3 of address 0378<sub>16</sub> = "1") : <math>f_{EXT} / 16 (n+1)</math> (Note 1) (Note 2)</li> </ul>
Transmission / reception control	<ul style="list-style-type: none"> <li>• Disable the <math>\overline{CTS}</math> and <math>\overline{RTS}</math> function (bit 4 of address 037C<sub>16</sub> = "1")</li> </ul>
Other settings	<ul style="list-style-type: none"> <li>• The sleep mode select function is not available for UART2</li> <li>• Set transmission interrupt factor to "transmission completed" (bit 4 of address 037D<sub>16</sub> = "1")</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>• To start transmission, the following requirements must be met:               <ul style="list-style-type: none"> <li>- Transmit enable bit (bit 0 of address 037D<sub>16</sub>) = "1"</li> <li>- Transmit buffer empty flag (bit 1 of address 037D<sub>16</sub>) = "0"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>• To start reception, the following requirements must be met:               <ul style="list-style-type: none"> <li>- Reception enable bit (bit 2 of address 037D<sub>16</sub>) = "1"</li> <li>- Detection of a start bit</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When transmitting               <ul style="list-style-type: none"> <li>When data transmission from the UART2 transfer register is completed (bit 4 of address 037D<sub>16</sub> = "1")</li> </ul> </li> <li>• When receiving               <ul style="list-style-type: none"> <li>When data transfer from the UART2 receive register to the UART2 receive buffer register is completed</li> </ul> </li> </ul>

**Table 2.11.9 Specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface) (2)**

Item	Specification
Error detection	<ul style="list-style-type: none"> <li>• Overrun error (see the specifications of clock-asynchronous serial I/O) (Note 3)</li> <li>• Framing error (see the specifications of clock-asynchronous serial I/O)</li> <li>• Parity error (see the specifications of clock-asynchronous serial I/O)               <ul style="list-style-type: none"> <li>- On the reception side, an "L" level is output from the TxD2 pin by use of the parity error signal output function (bit 7 of address 037D<sub>16</sub> = "1") when a parity error is detected</li> <li>- On the transmission side, a parity error is detected by the level of input to the RxD2 pin when a transmission interrupt occurs</li> </ul> </li> <li>• The error sum flag (see the specifications of clock-asynchronous serial I/O)</li> </ul>

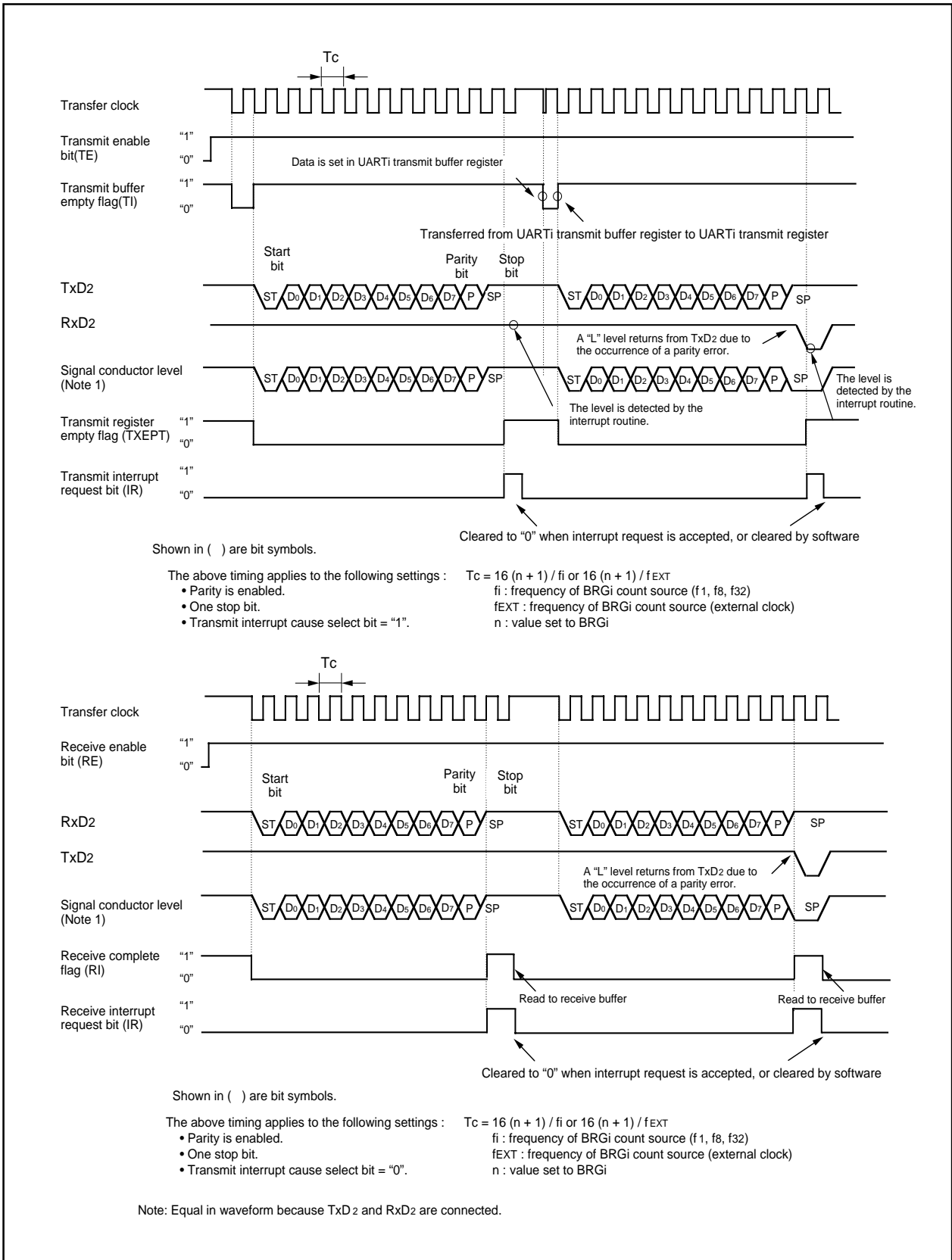
**Notes 1:** 'n' denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UARTi bit rate generator.

**2:** f<sub>EXT</sub> is input from the CLK2 pin.

**3:** If an overrun error occurs, the UART2 receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit is not set to "1".

**M306V0ME-XXXFP**  
**M306V0EEFP**

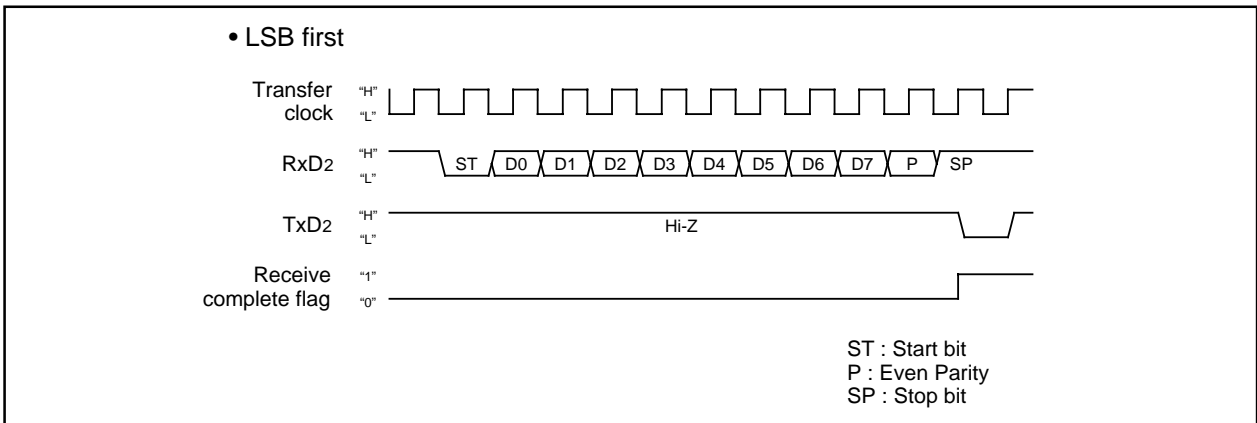
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.11.27 Typical transmit/receive timing in UART mode (compliant with the SIM interface)**

**(1) Function for outputting a parity error signal**

With the error signal output enable bit (bit 7 of address 037D16) assigned “1”, you can output an “L” level from the TxD2 pin when a parity error is detected. In step with this function, the generation timing of a transmission completion interrupt changes to the detection timing of a parity error signal. Figure 2.11.28 shows the output timing of the parity error signal.

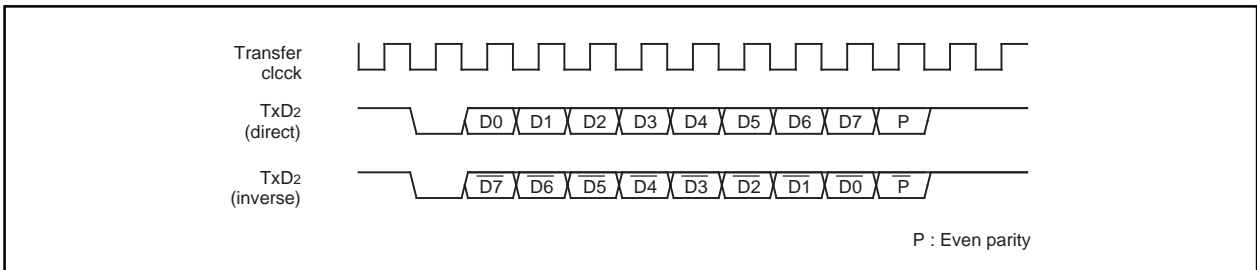


**Figure 2.11.28 Output timing of the parity error signal**

**(2) Direct format/inverse format**

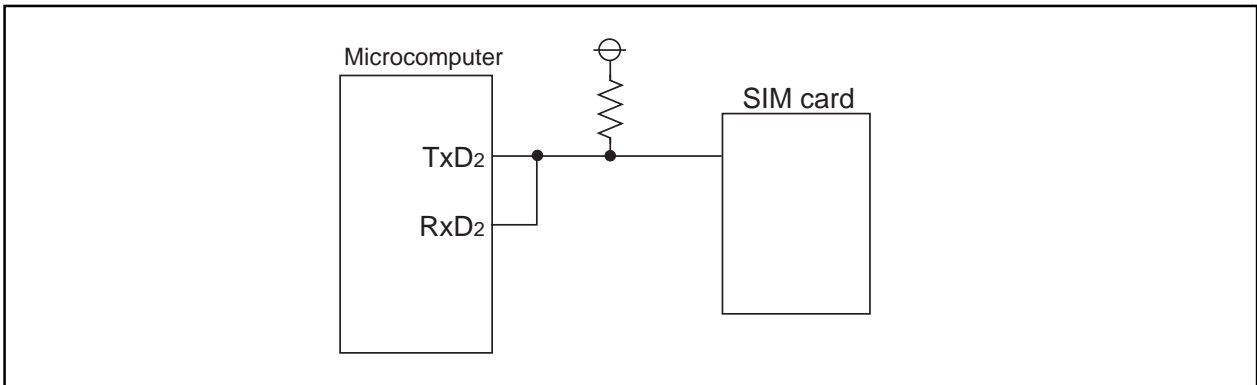
Connecting the SIM card allows you to switch between direct format and inverse format. If you choose the direct format, D0 data is output from TxD2. If you choose the inverse format, D7 data is inverted and output from TxD2.

Figure 2.11.29 shows the SIM interface format.



**Figure 2.11.29 SIM interface format**

Figure 2.11.30 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.



**Figure 2.11.30 Connecting the SIM interface**

### 2.11.5 UART2 Special Mode Register

The UART2 special mode register (address 0377<sub>16</sub>) is used to control UART2 in various ways. Figure 2.11.31 shows the UART2 special mode register.

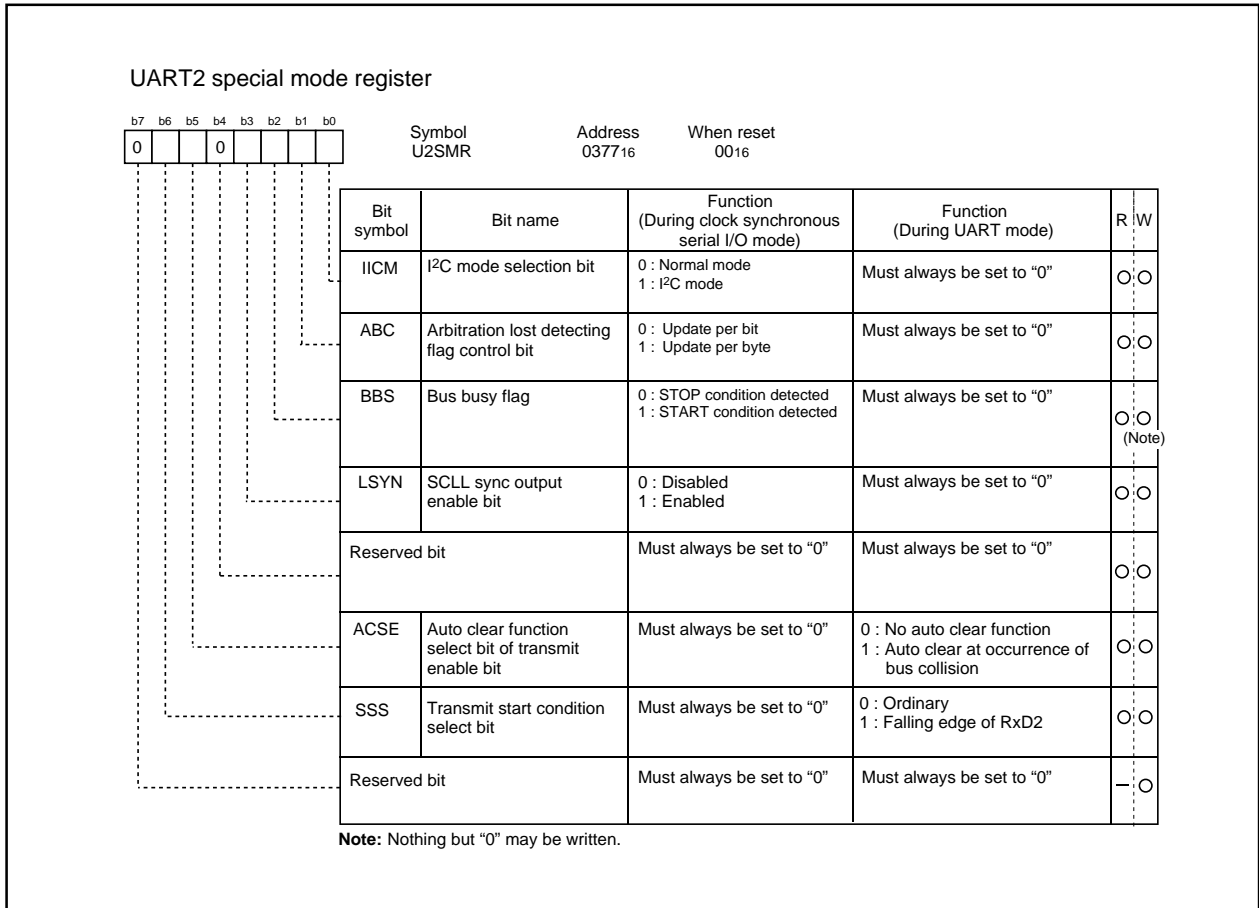


Figure 2.11.31 UART2 special mode register



**Table 2.11.10 Features in I<sup>2</sup>C mode**

	Function	Normal mode	I <sup>2</sup> C mode
1	Factor of interrupt number 10	Bus collision detection	Start condition detection or stop condition detection (Note)
2	Factor of interrupt number 15	UART2 transmission	No acknowledgment detection (NACK) (Note)
3	Factor of interrupt number 16	UART2 reception	Acknowledgment detection (ACK) (Note)
4	UART2 transmission output delay	Not delayed	Delayed
5	P7 <sub>0</sub> at the time when UART2 is in use	TxD <sub>2</sub> (output)	SDA1 (input/output) (Note 3)
6	P7 <sub>1</sub> at the time when UART2 is in use	RxD <sub>2</sub> (input)	SCL1 (input/output) (Note 3)
7	P7 <sub>2</sub> at the time when UART2 is in use	CLK <sub>2</sub> (CMOS)	SCL2 (N-channel open-drain) (Note 3)
8	DMA1 factor at the time when 1 1 0 1 is assigned to the DMA request factor selection bits	UART2 reception	Acknowledgment detection (ACK)
9	Noise filter width	15 ns	50 ns
10	Reading P7 <sub>1</sub> /P7 <sub>2</sub>	Reading the terminal when 0 is assigned to the direction register	Reading the terminal regardless of the value of the direction register (Note 3)
11	Initial value of UART2 output	H level (when 0 is assigned to the CLK polarity select bit)	The value set in latch P7 <sub>0</sub> /P6 <sub>7</sub> when the port is selected (Note 3)
12	P6 <sub>7</sub> input/output	P6 <sub>7</sub> (CMOS)	SDA2 (N-channel open-drain) (Note 3)

- Notes 1:** Make the settings given below when I<sup>2</sup>C mode is in use.
- Set 0 1 0 in bits 2, 1, 0 of the UART2 transmission-reception mode register.
  - Disable the RTS/CTS function.
  - Choose the LSB First function. Follow the steps given below to switch from a factor to another.
- 2:** Follow the steps given below to switch from a factor to another.
1. Disable the interrupt of the corresponding number.
  2. Switch from a factor to another.
  3. Reset the interrupt request flag of the corresponding number.
  4. Set an interrupt level of the corresponding number.
- 3:** In I<sup>2</sup>C mode and when setting as I<sup>2</sup>C-BUS interface ports.

In the first place, the control bits related to the I<sup>2</sup>C-BUS interface are explained.

Setting 1 in the I<sup>2</sup>C mode selection bit (bit 0) goes the circuit to achieve the I<sup>2</sup>C-BUS (simplified I<sup>2</sup>C-BUS) interface effective. Table 2.11.10 shows the relation between the I<sup>2</sup>C mode selection bit and respective control workings.

Since this function uses clock-synchronous serial I/O mode, be sure to set this bit to “0” in UART mode. Figure 2.11.32 shows the functional block diagram for I<sup>2</sup>C mode. Setting “1” in the I<sup>2</sup>C mode selection bit (IICM) causes ports P7<sub>0</sub>, P6<sub>7</sub>, P7<sub>1</sub>, and P7<sub>2</sub> selected by bits 0 and 1 of the peripheral mode register (address 027D<sub>16</sub>) to work as data transmission-reception terminals; SDA1, SDA2, clock input-output terminals; SCL1, SCL2 respectively. A delay circuit is added to the SDA transmission output, so the SDA output changes after SCL fully goes to L. An attempt to read Port P7<sub>1</sub> (SCL1), P7<sub>2</sub> (SCL2) results in getting the terminal’s level regardless of the content of the port direction register. The initial value of SDA transmission output in this mode goes to the value set in port (P7<sub>0</sub> when using SDA1, P6<sub>7</sub> when using SDA2, P7<sub>0</sub> when using both SDA1 and SDA2). The interrupt factors of the bus collision detection interrupt, UART<sub>2</sub> transmission interrupt, and of UART2 reception interrupt turn to the start/stop condition detection interrupt, acknowledgment non-detection (NACK) interrupt, and acknowledgment detection (ACK) interrupt respectively.

The start condition detection interrupt refers to the interrupt that occurs when the falling edge of the SDA terminal is detected with the SCL terminal staying “H.” The stop condition detection interrupt refers to the interrupt that occurs when the rising edge of the SDA terminal is detected with the SCL terminal staying “H”. The bus busy flag (bit 2 of the special UART2 mode register) is set to “1” by the start condition detection, and set to “0” by the stop condition detection. The acknowledgment non-detection interrupt refers to the interrupt that occurs when the SDA terminal’s level is detected still staying “H” at the rising edge of the 9th transmission clock. The acknowledgment detection interrupt refers to the interrupt that occurs when SDA terminal’s level is detected already went to “L” at the 9th transmission clock. Also, assigning 1 1 0 1 (UART2 reception) to the DMA1 request factor selection bits provides the means to start up the DMA transfer by the effect of acknowledgment detection.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

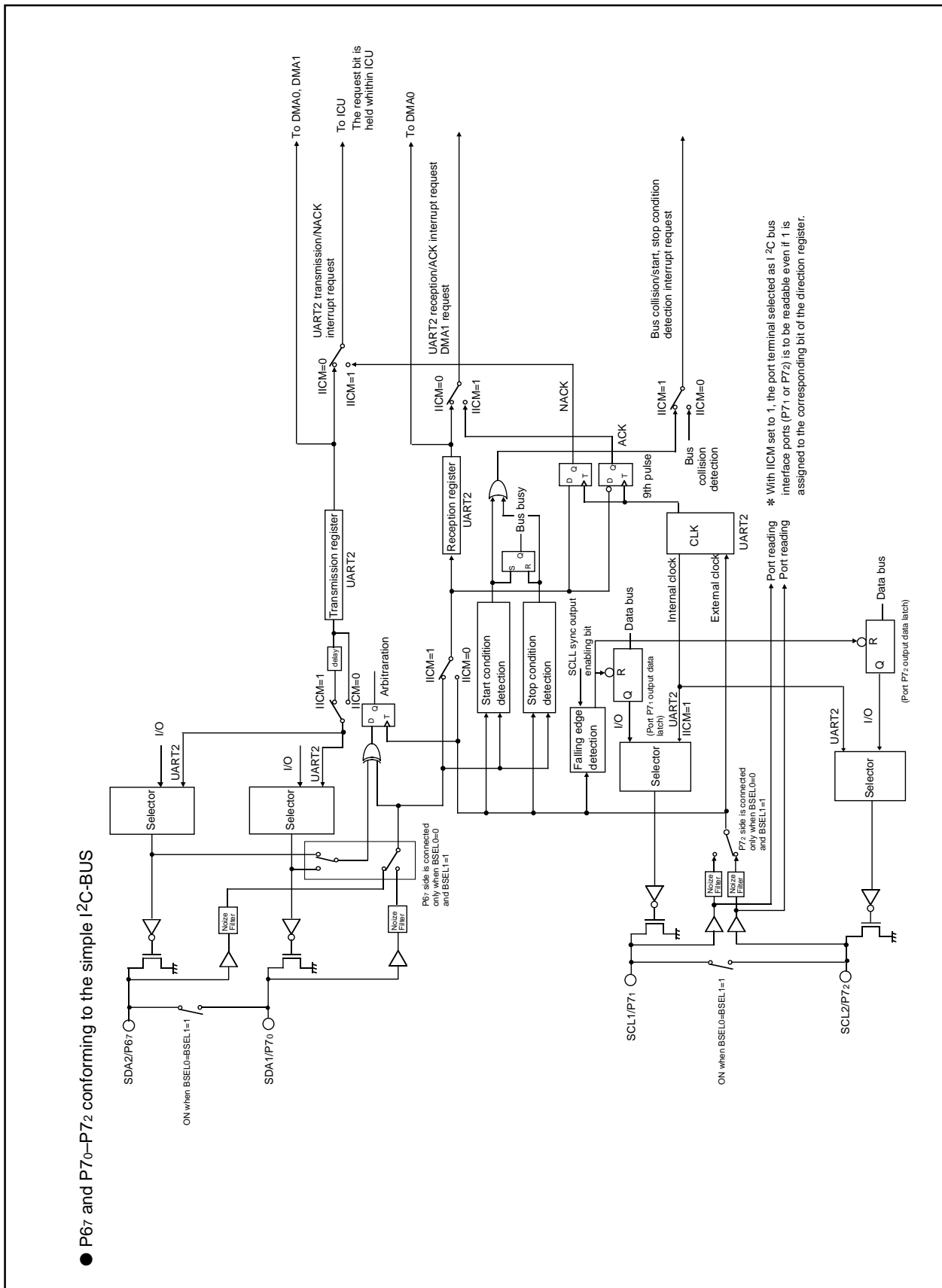


Figure 2.11.32 Functional block diagram for I<sup>2</sup>C mode

**M306V0ME-XXXFP**  
**M306V0EEFP**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

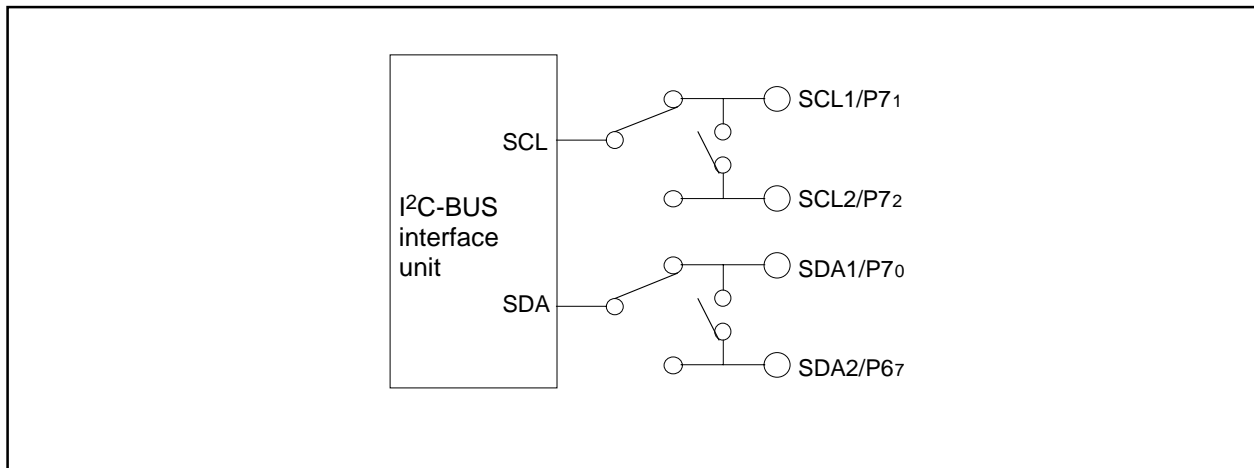
---

Bit 1 of the special UART2 mode register (0377<sub>16</sub>) is used as the arbitration lost detecting flag control bit. Arbitration means the act of detecting the nonconformity between transmission data and SDA terminal data at the timing of the SCL rising edge. This detection flag is located at bit 3 of the UART2 reception buffer register (037F<sub>16</sub>), and "1" is set in this flag when nonconformity is detected. Use the arbitration lost detecting flag control bit to choose which way to use to update the flag, bit by bit or byte by byte. Setting this bit to "1" sets the flag for each byte. When detecting nonconformity, the arbitration lost detecting flag is set to "1" at the falling edge of the ninth clock. To update the flag, be sure to determinate the flag and set "0" to it until the next 1-byte transfer starts after the detection of first-byte acknowledge.

Bit 3 of the UART2 special mode register is used as SCL L sync output enable bit. Setting this bit to "1" resets the data register of port (P7<sub>1</sub> when using SCL1, P7<sub>2</sub> when using SCL2, P7<sub>1</sub> when using both SCL1 and SCL2) to "0" in synchronization with the SCL terminal's level going to "L."

### 2.11.6 Simple I<sup>2</sup>C-BUS interface

The I/O ports (P6<sub>7</sub>, P7<sub>0</sub> to P7<sub>2</sub>) function as I<sup>2</sup>C-BUS interface ports. These ports are set by bits 0 and 1 of the peripheral mode register (see Figure 2.10.33).



**Figure 2.11.33 I<sup>2</sup>C-BUS interface port control**

## 2.12 A-D Converter

The A-D converter consists of one 8-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins P102 to P107 also function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address 03D716) can be used to isolate the resistance ladder of the A-D converter from the reference voltage (VREF) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from VREF, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of 03D716 to connect VREF.

The result of A-D conversion is stored in the A-D registers of the selected pins.

Table 2.12.1 shows the performance of the A-D converter. Figure 2.12.1 shows the block diagram of the A-D converter, and Figures 2.12.2 to 2.12.5 show the A-D converter-related registers.

**Table 2.12.1 Performance of A-D converter**

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AVCC (VCC)
Operating clock $\phi_{AD}$ (Note 2)	$f_{AD}/\text{divide-by-2}$ of $f_{AD}/\text{divide-by-4}$ of $f_{AD}$ , $f_{AD}=f(XIN)$
Resolution	8-bit
Absolute precision	VCC = 5V <ul style="list-style-type: none"> <li>• Without sample and hold function: <math>\pm 5</math> LSB</li> <li>• With sample and hold function: <math>\pm 5</math> LSB</li> </ul>
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	6 pins (AN0 to AN5)
A-D conversion start condition	<ul style="list-style-type: none"> <li>• Software trigger</li> </ul> A-D conversion starts when the A-D conversion start flag changes to "1"
Conversion speed per pin	<ul style="list-style-type: none"> <li>• Without sample and hold function 49 <math>\phi_{AD}</math> cycles</li> <li>• With sample and hold function 28 <math>\phi_{AD}</math> cycles</li> </ul>

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

**2:** Divide the frequency if  $f(XIN)$  exceeds 10 MHz, and make  $\phi_{AD}$  frequency equal to 10 MHz. Without sample and hold function, set the  $\phi_{AD}$  frequency to 250kHz min.

With the sample and hold function, set the  $\phi_{AD}$  frequency to 1MHz min.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

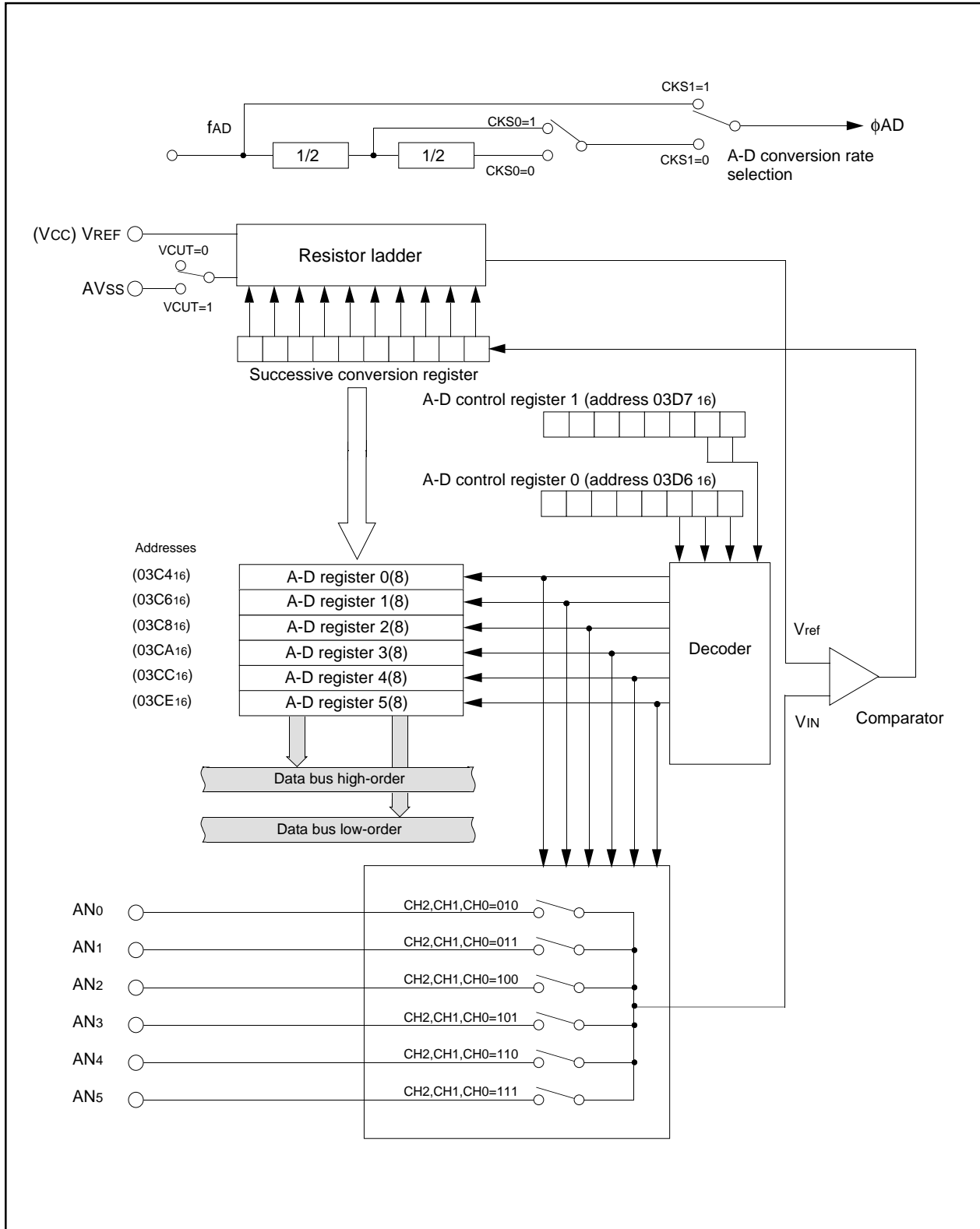


Figure 2.12.1 Block diagram of A-D converter

M306V0ME-XXXFP  
M306V0EEFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

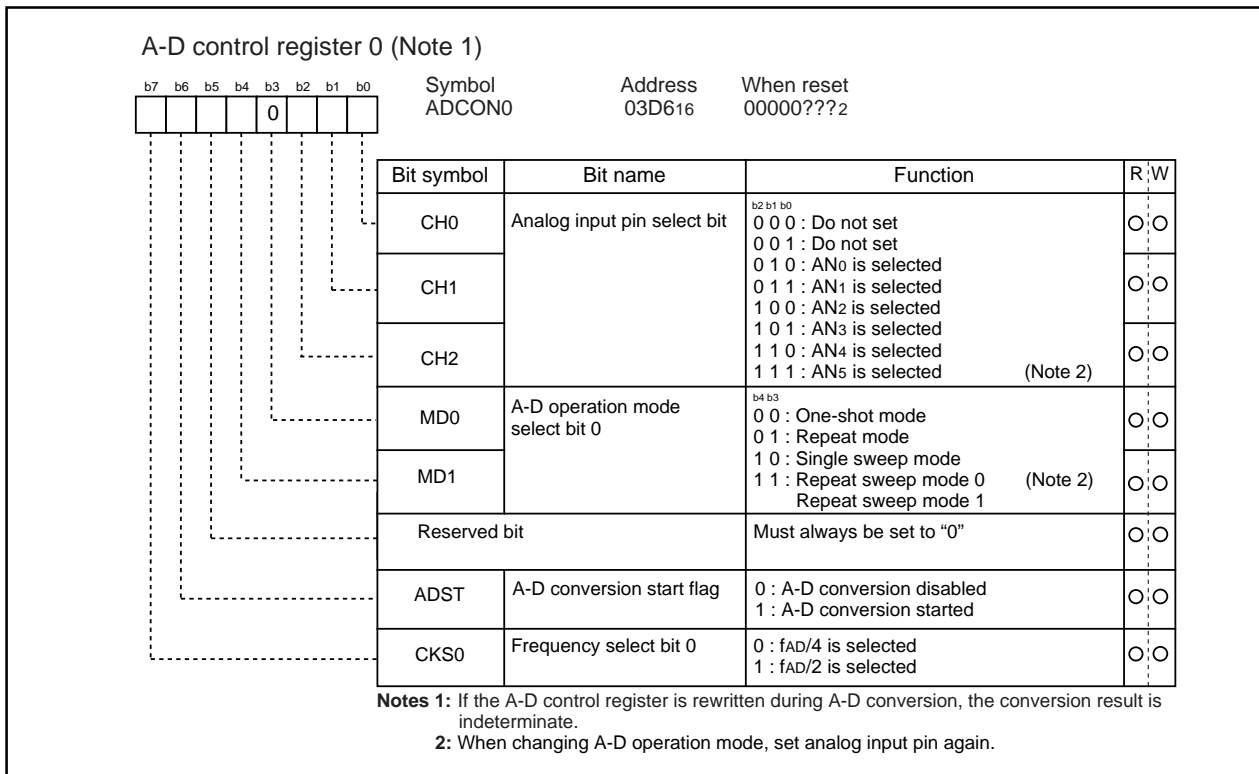


Figure 2.12.2 A-D control register 0

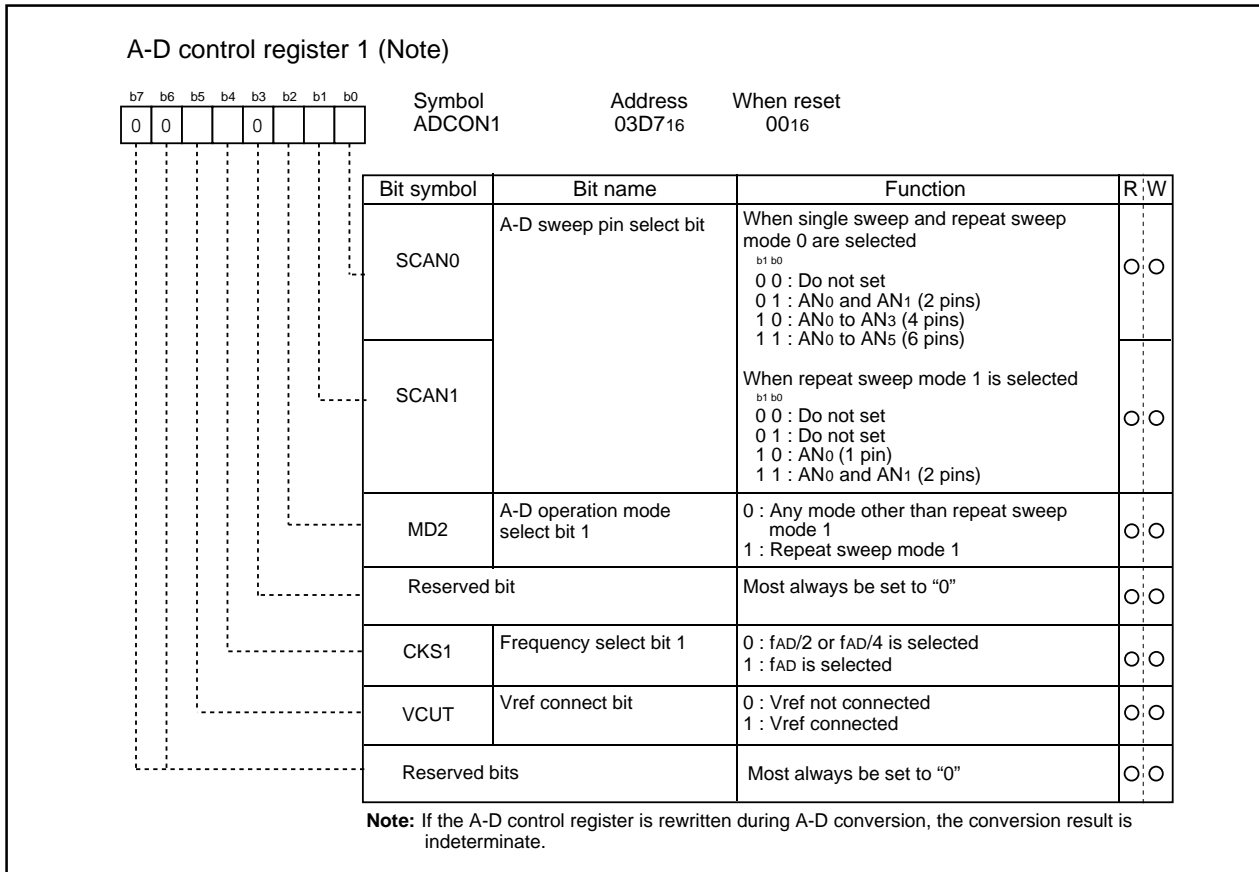


Figure 2.12.3 A-D control register 1

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

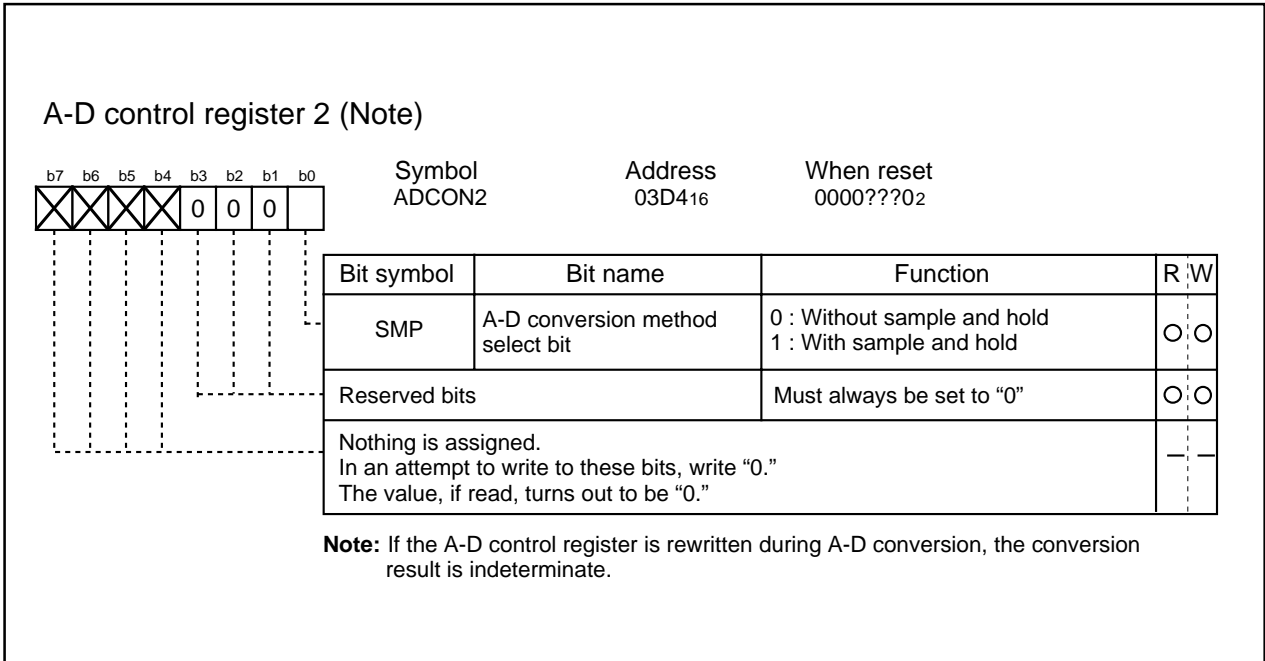


Figure 2.12.4 A-D control register 2

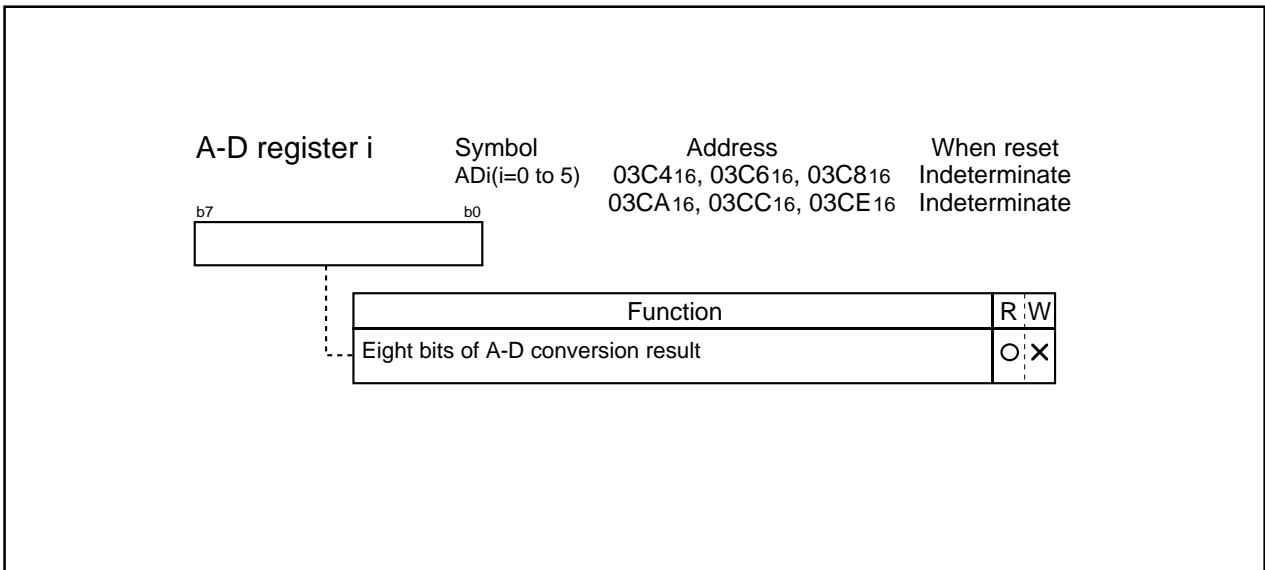


Figure 2.12.5 A-D register i (i = 0 to 5)



### 2.12.1 One-shot mode

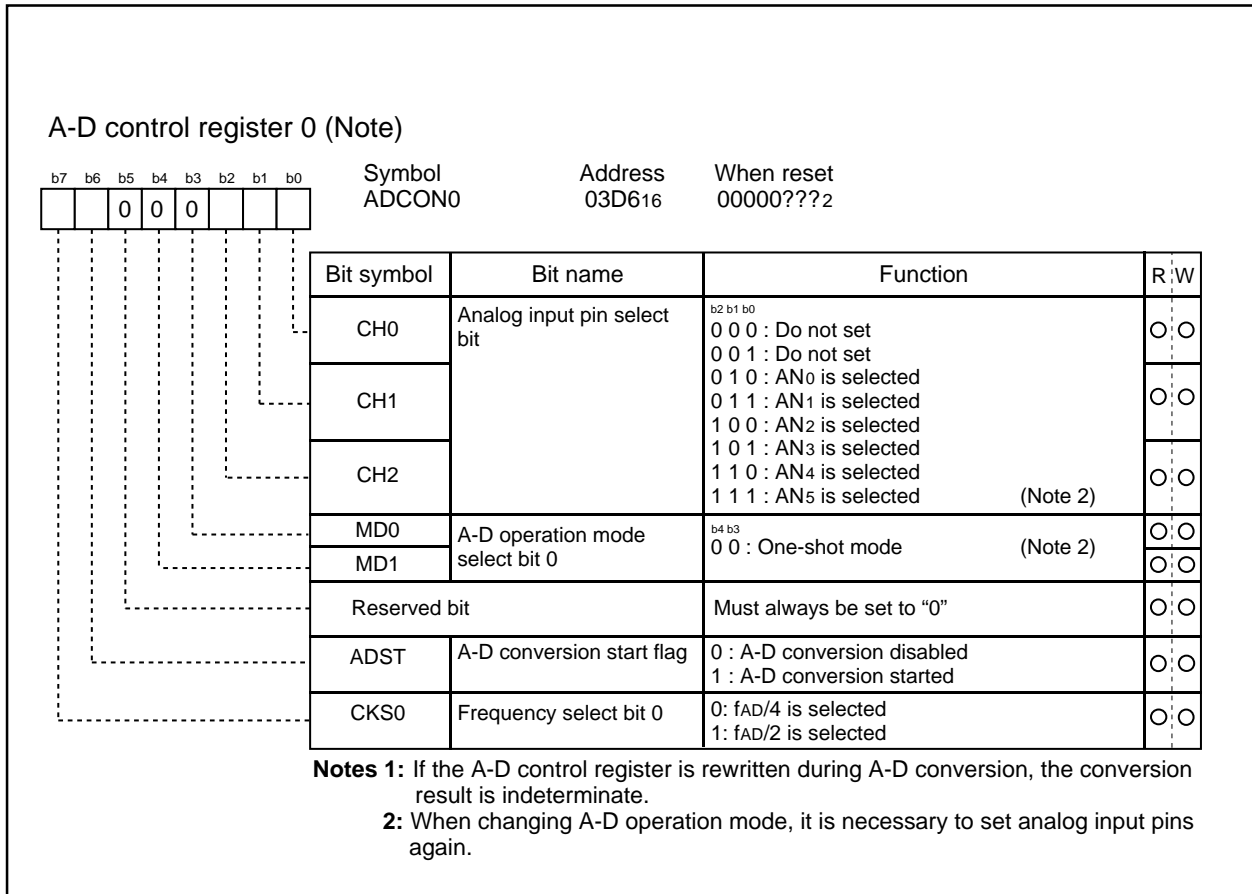
In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion. Table 2.12.2 shows the specifications of one-shot mode. Figures 2.12.6 and 2.12.7 show the A-D control register in one-shot mode.

**Table 2.12.2 One-shot mode specifications**

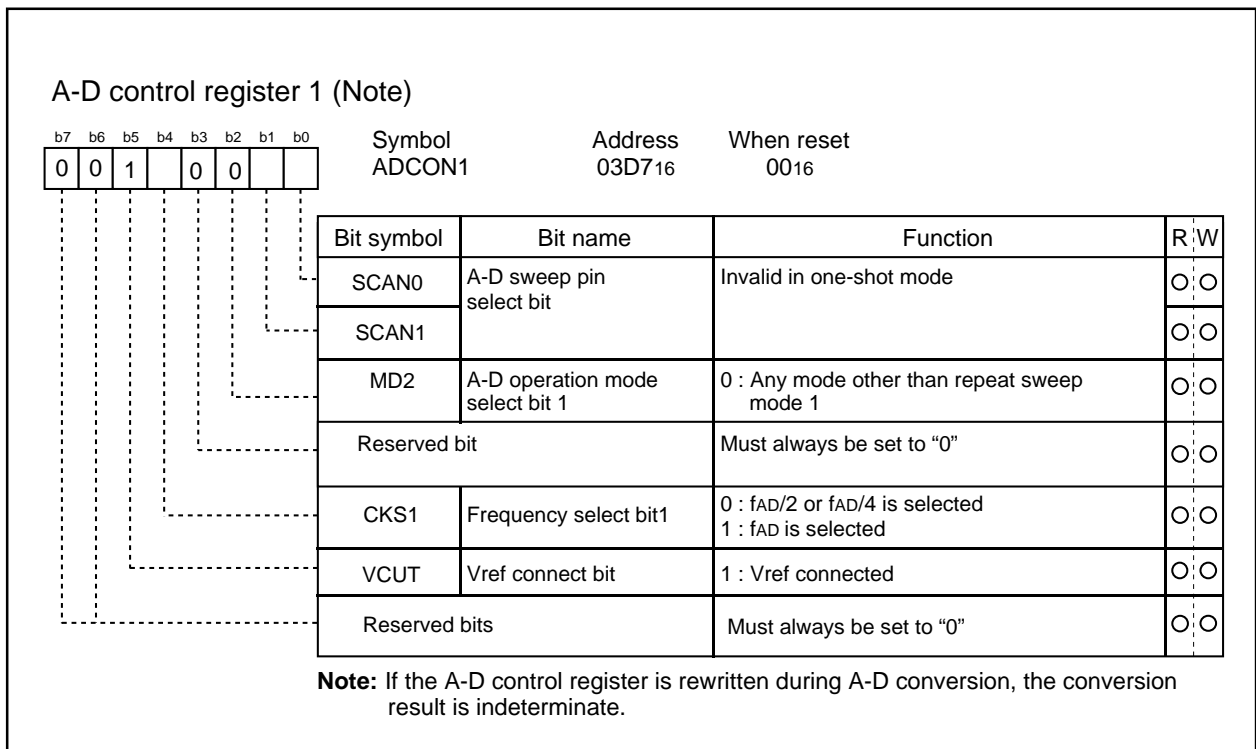
Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	<ul style="list-style-type: none"> <li>• End of A-D conversion</li> <li>• Writing "0" to A-D conversion start flag</li> </ul>
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN <sub>0</sub> to AN <sub>5</sub> , as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.12.6 A-D control register 0 in one-shot mode**



**Figure 2.12.7 A-D control register 1 in one-shot mode**

### 2.12.2 Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion.

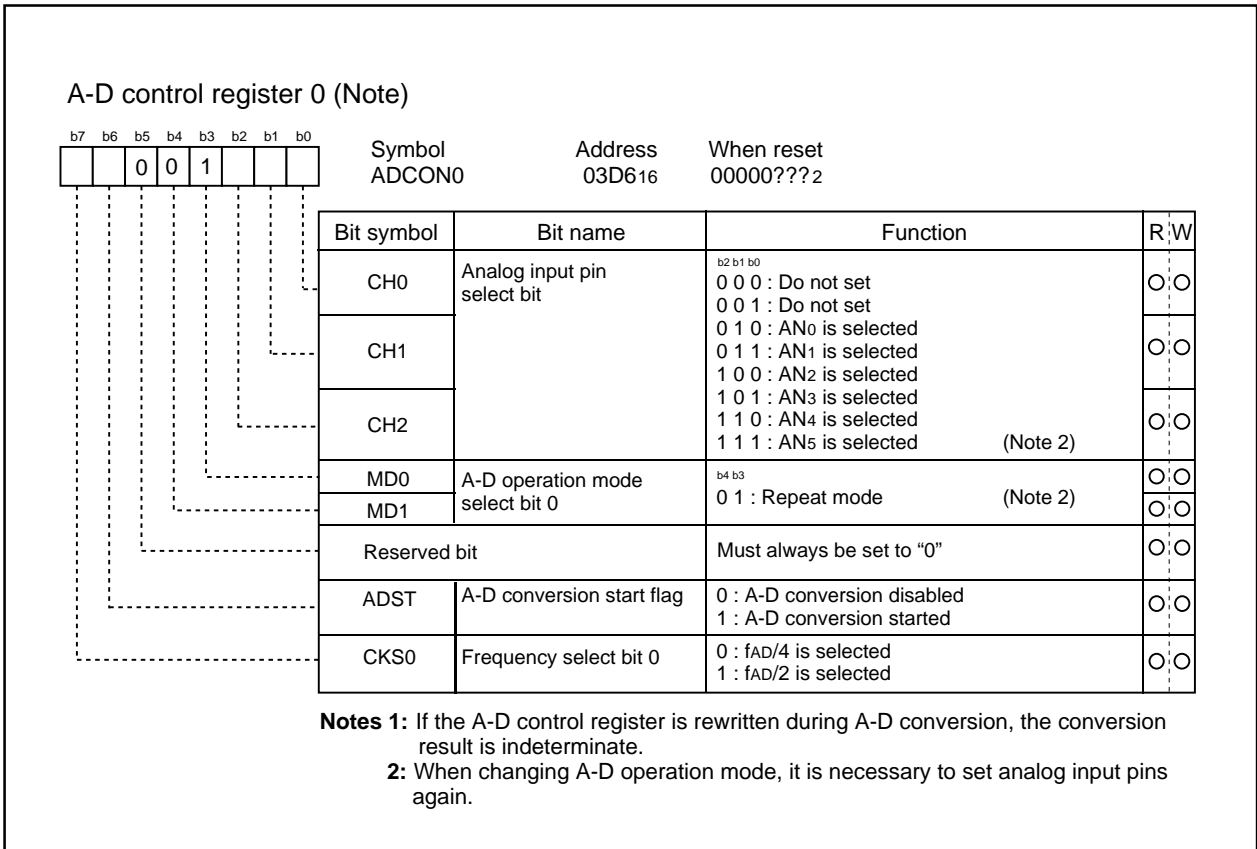
Table 2.12.3 shows the specifications of repeat mode. Figures 2.12.8 and 2.12.9 show the A-D control register in repeat mode.

**Table 2.12.3 Repeat mode specifications**

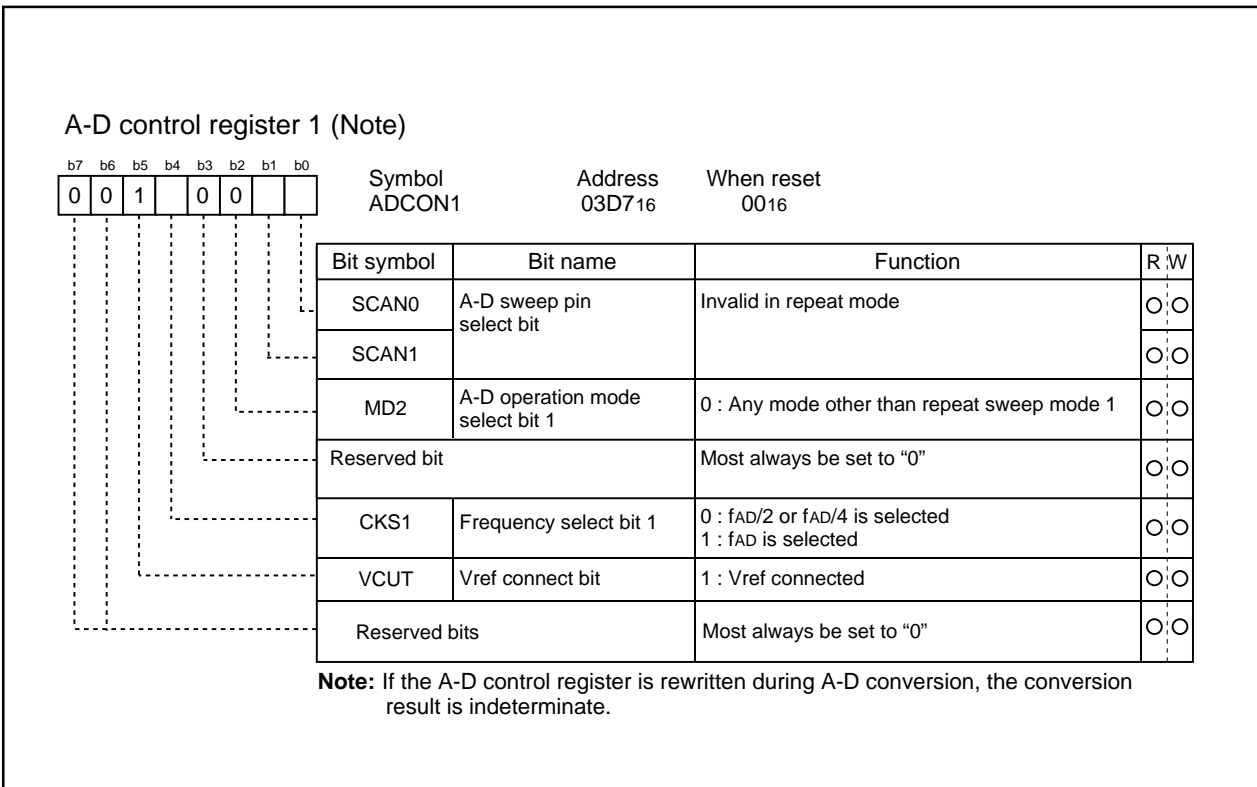
Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D conversion
Star condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN <sub>0</sub> to AN <sub>5</sub> , as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.12.8 A-D conversion register 0 in repeat mode**



**Figure 2.12.9 A-D conversion register 1 in repeat mode**

### 2.12.3 Single sweep mode

In single sweep mode, the pins selected using the A-D sweep pin select bit are used for one-by-one A-D conversion. Table 2.12.4 shows the specifications of single sweep mode. Figures 2.12.10 and 2.12.11 show the A-D control register in single sweep mode.

**Table 2.12.4 Single sweep mode specifications**

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for one-by-one A-D conversion
Start condition	Writing "1" to A-D converter start flag
Stop condition	<ul style="list-style-type: none"> <li>• End of A-D conversion</li> <li>• Writing "0" to A-D conversion start flag</li> </ul>
Interrupt request generation timing	End of A-D conversion
Input pin	AN <sub>0</sub> and AN <sub>1</sub> (2 pins), AN <sub>0</sub> to AN <sub>3</sub> (4 pins), AN <sub>0</sub> to AN <sub>5</sub> (6 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

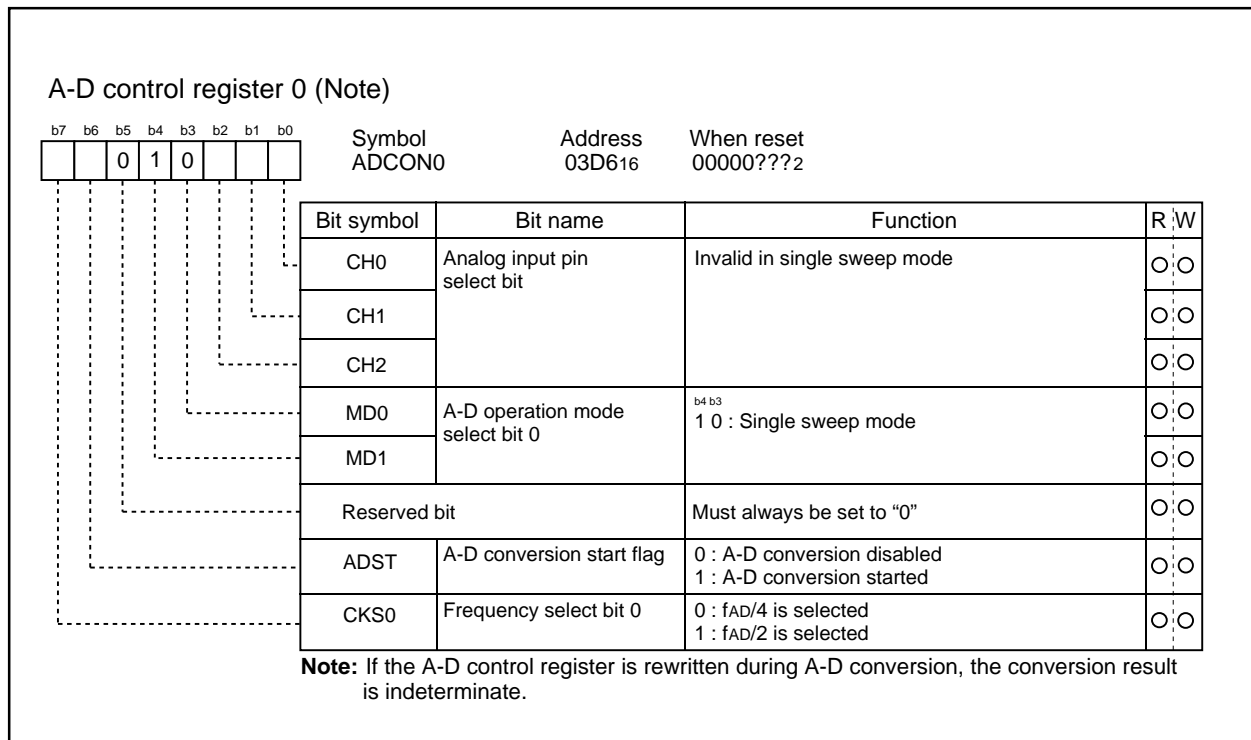


Figure 2.12.10 A-D control register 0 in single sweep mode

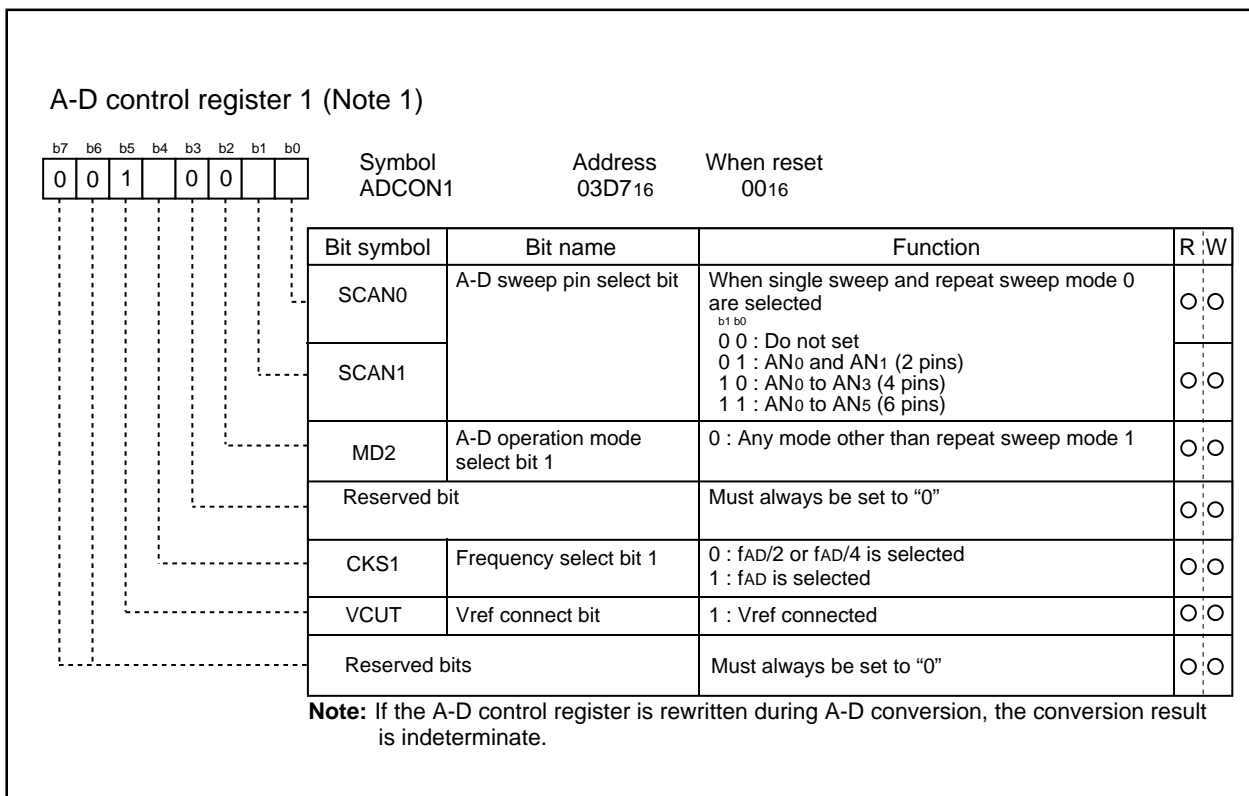


Figure 2.12.11 A-D control register 1 in single sweep mode

### 2.12.4 Repeat sweep mode 0

In repeat sweep mode 0, the pins selected using the A-D sweep pin select bit are used for repeat sweep A-D conversion. Table 2.12.5 shows the specifications of repeat sweep mode 0. Figures 2.12.12 and 2.12.13 show the A-D control register in repeat sweep mode 0.

**Table 2.12.5 Repeat sweep mode 0 specifications**

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for repeat sweep A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN <sub>0</sub> and AN <sub>1</sub> (2 pins), AN <sub>0</sub> to AN <sub>3</sub> (4 pins), AN <sub>0</sub> to AN <sub>5</sub> (6 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

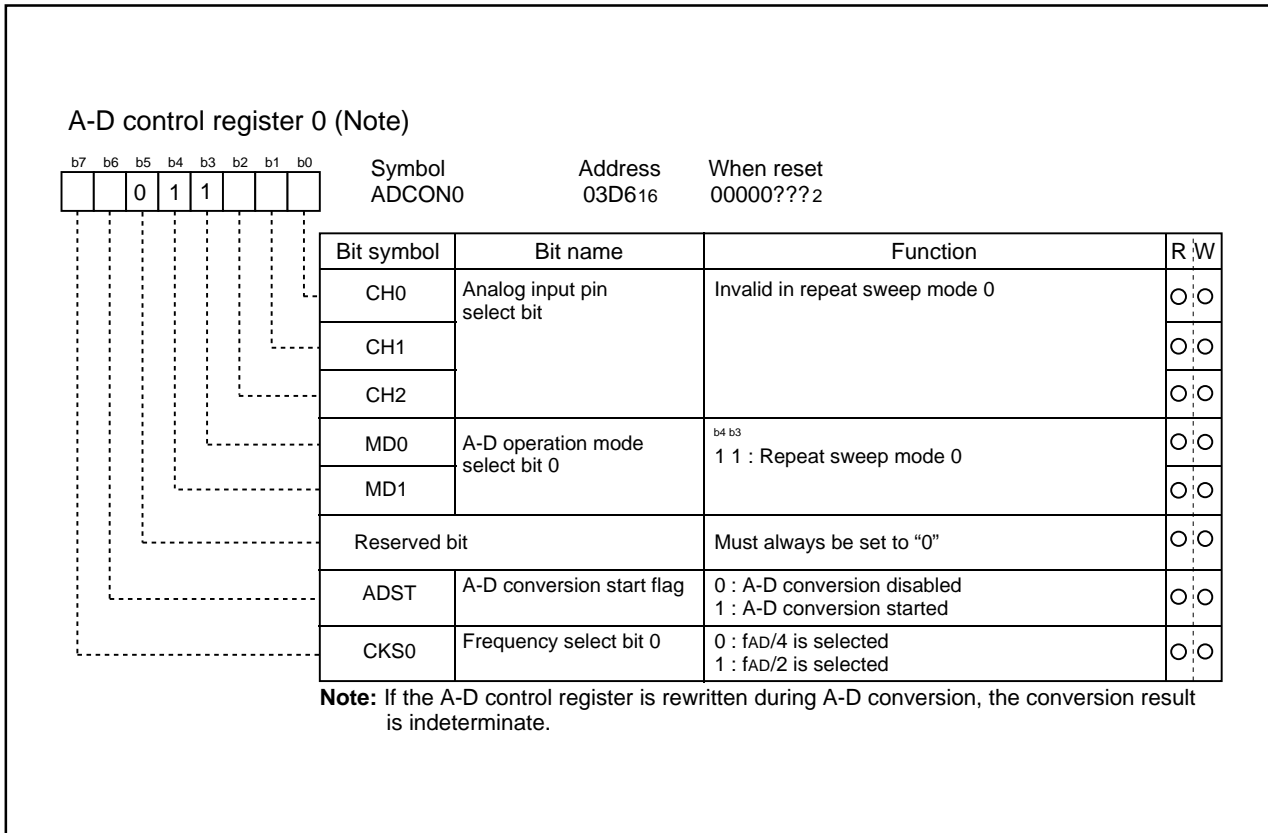


Figure 2.12.12 A-D control register 0 in repeat sweep mode 0

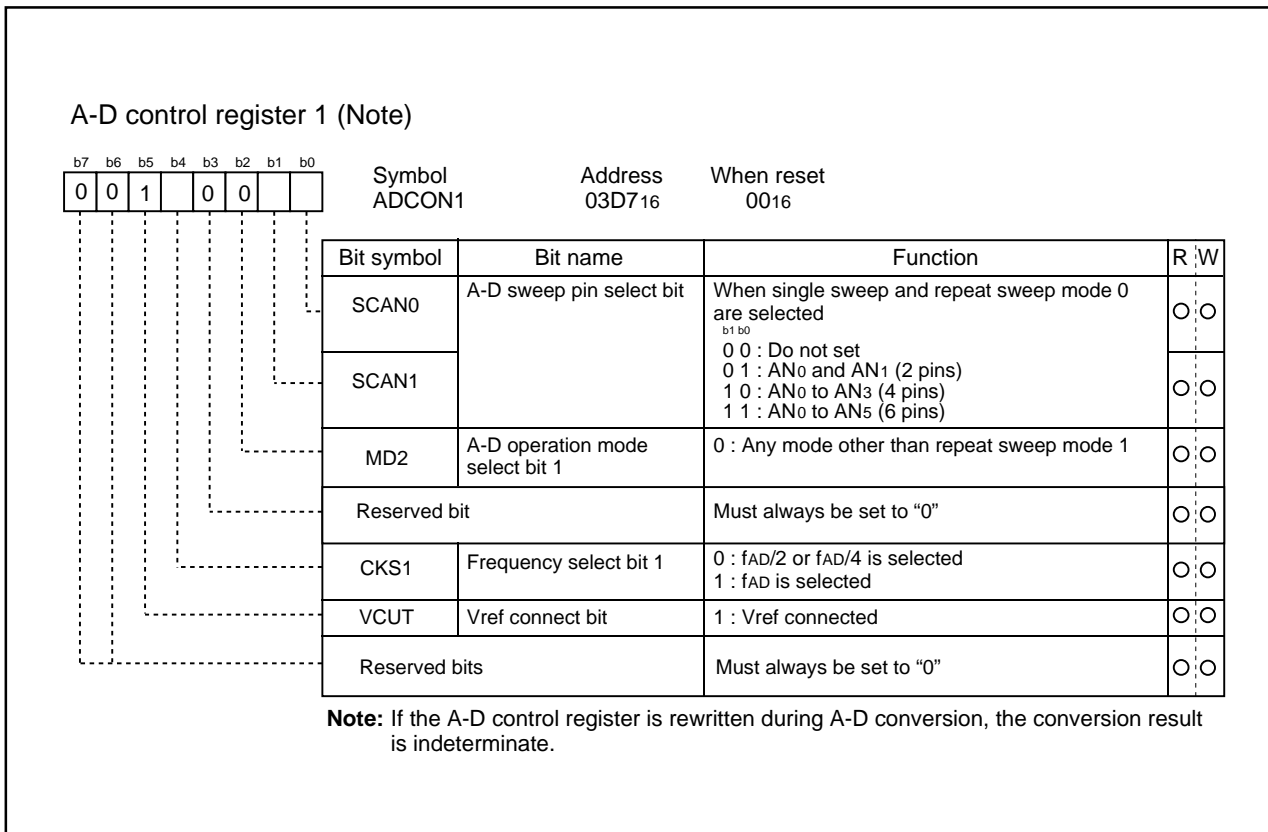


Figure 2.12.13 A-D control register 1 in repeat sweep mode 0



**M306V0ME-XXXFP**  
**M306V0EEFP**

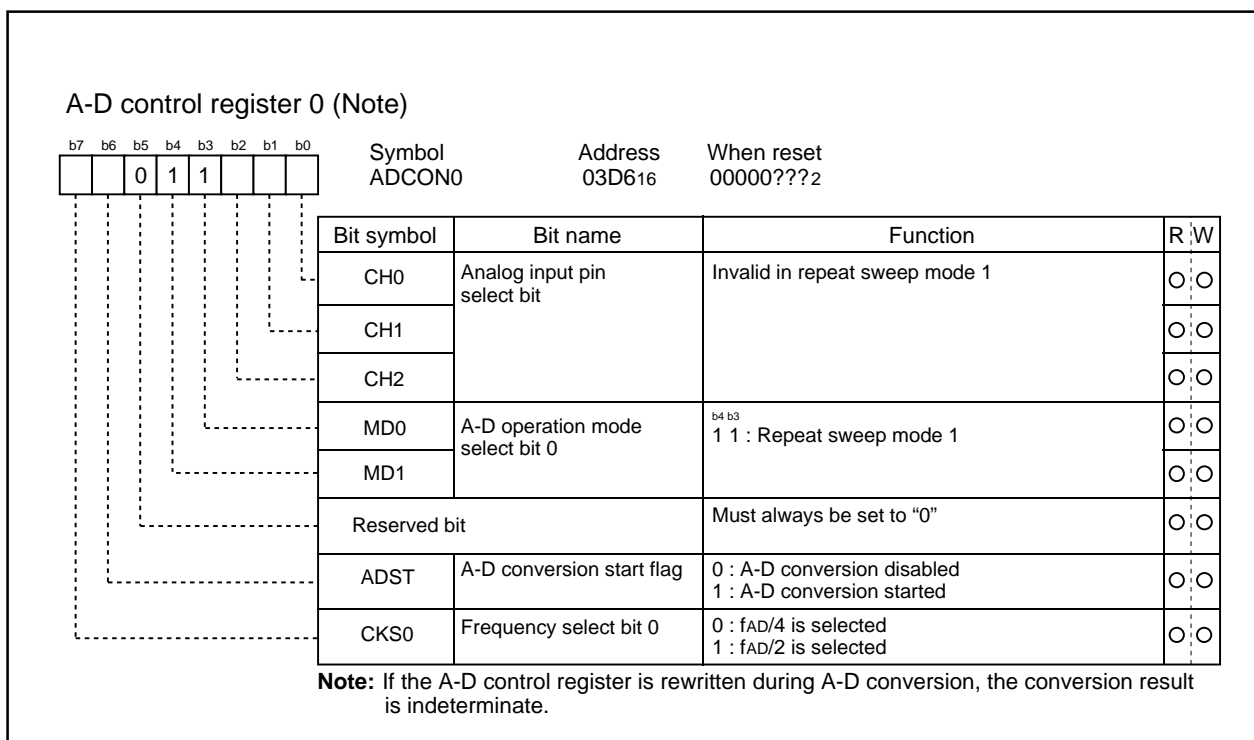
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**2.12.5 Repeat sweep mode 1**

In repeat sweep mode 1, all pins are used for A-D conversion with emphasis on the pin or pins selected using the A-D sweep pin select bit. Table 2.12.6 shows the specifications of repeat sweep mode 1. Figures 2.12.14 and 2.12.15 show the A-D control register in repeat sweep mode 1.

**Table 2.12.6 Repeat sweep mode 1 specifications**

Item	Specification
Function	All pins perform repeat sweep A-D conversion, with emphasis on the pin or pins selected by the A-D sweep pin select bit Example : AN <sub>0</sub> selected AN <sub>0</sub> → AN <sub>1</sub> → AN <sub>0</sub> → AN <sub>2</sub> → AN <sub>0</sub> → AN <sub>3</sub> , etc
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN <sub>0</sub> (1 pin), AN <sub>0</sub> and AN <sub>1</sub> (2 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)



**Figure 2.12.14 A-D control register 0 in repeat sweep mode 1**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

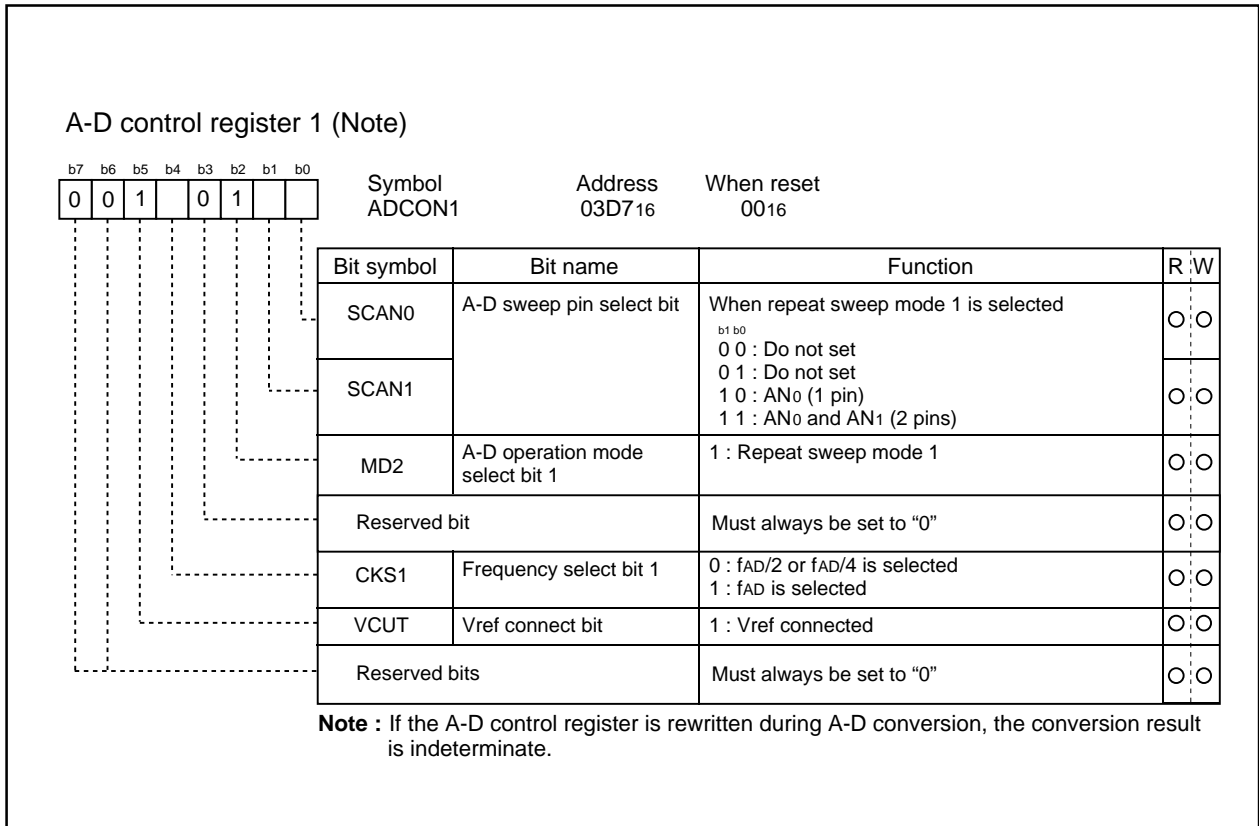


Figure 2.12.15 A-D control register 1 in repeat sweep mode 1

### 2.12.6 Sample and hold

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 03D4<sub>16</sub>) to "1". When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28  $\phi_{AD}$  cycle is achieved. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A-D conversion whether sample and hold is to be used.

### 2.13 D-A Converter

This is an 8-bit, R-2R type D-A converter. The microcomputer contains two independent D-A converters of this type.

D-A conversion is performed when a value is written to the corresponding D-A register. Bits 0 and 1 (D-A output enable bits) of the D-A control register decide if the result of conversion is to be output. Do not set the target port to output mode if D-A conversion is to be performed.

Output analog voltage (V) is determined by a set value (n : decimal) in the D-A register.

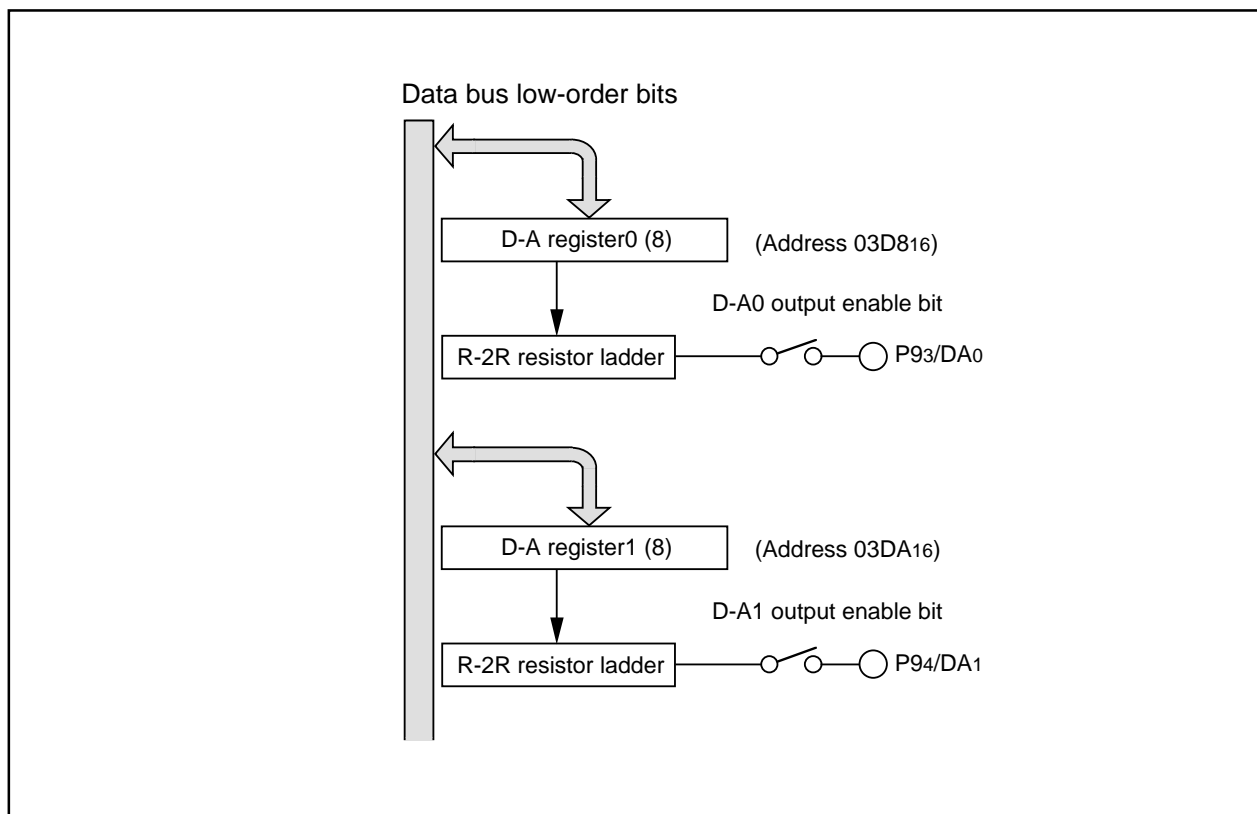
$$V = V_{REF} \times n / 256 \quad (n = 0 \text{ to } 255)$$

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

Table 2.13.1 lists the performance of the D-A converter. Figure 2.13.1 shows the block diagram of the D-A converter. Figure 2.13.2 shows the A-D control register, Figure 2.13.3 shows the D-A register and Figure 2.13.4 shows the D-A converter equivalent circuit.

**Table 2.13.1 Performance of D-A converter**

Item	Performance
Conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 channels



**Figure 2.13.1 Block diagram of D-A converter**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

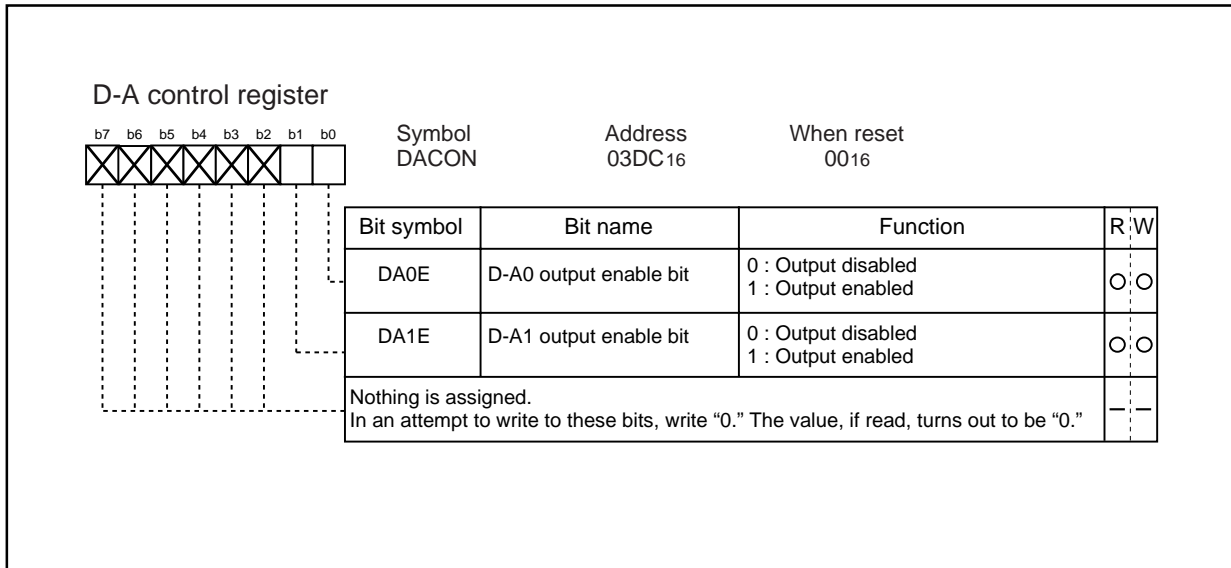


Figure 2.13.2 D-A control register

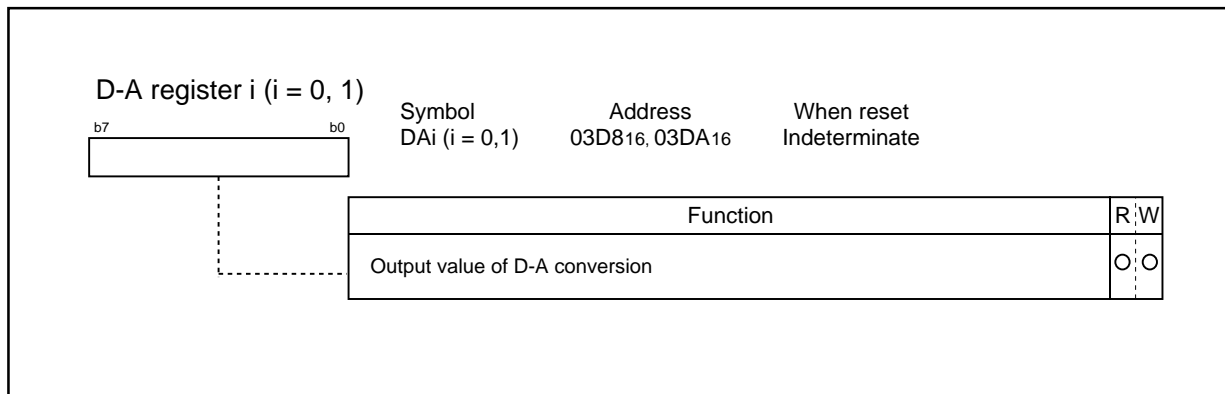


Figure 2.13.3 D-A register i (i = 0 and 1)

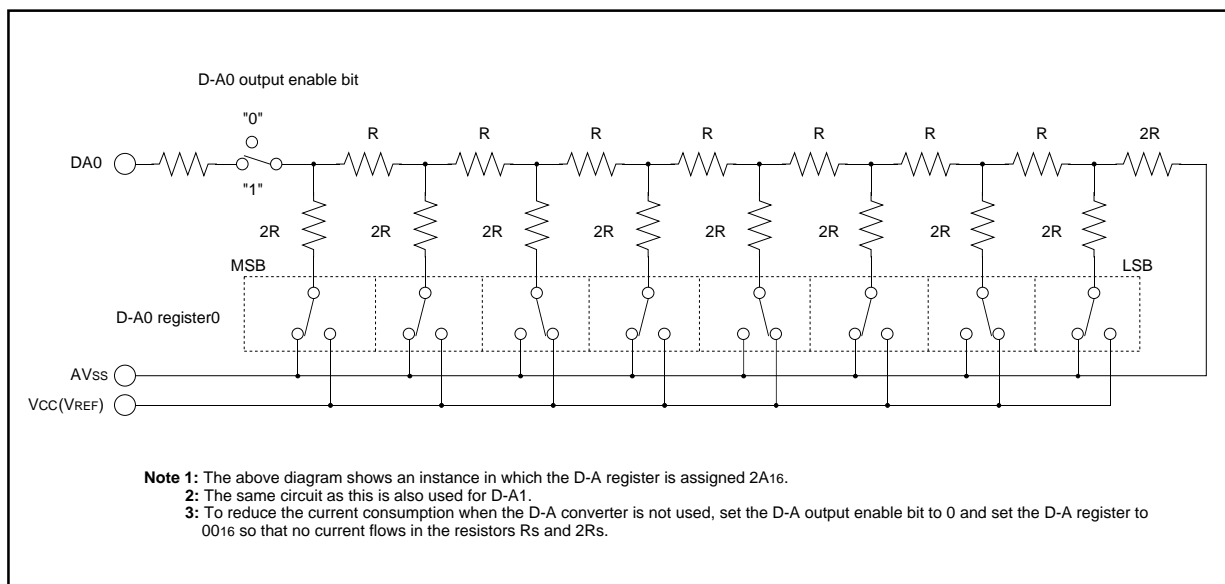


Figure 2.13.4 D-A converter equivalent circuit

### 2.14 Data Slicer

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

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

**Note:** When using the data slicer, set bit 7 of the peripheral mode register (address 027D16) according to the main clock frequency.

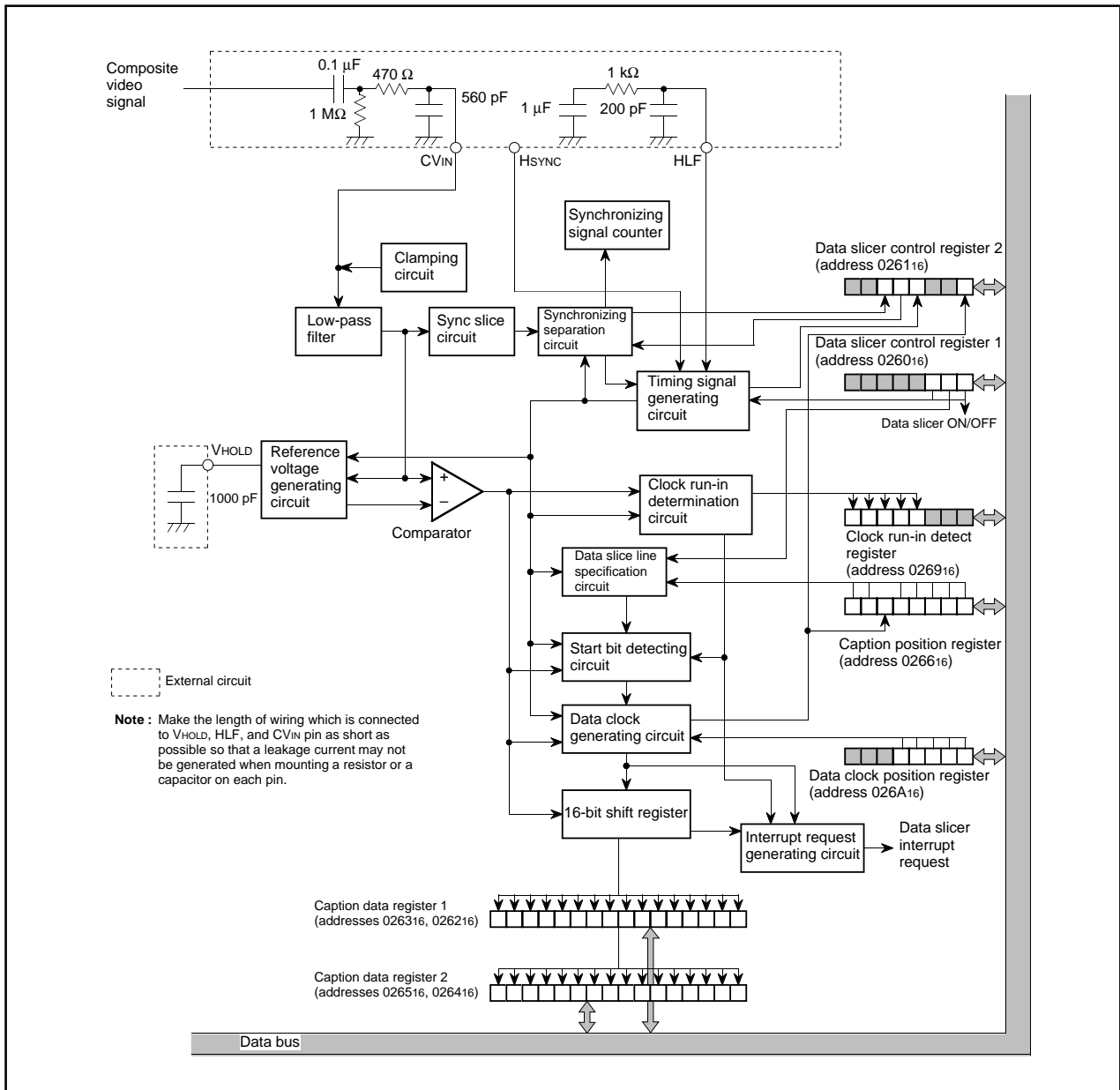
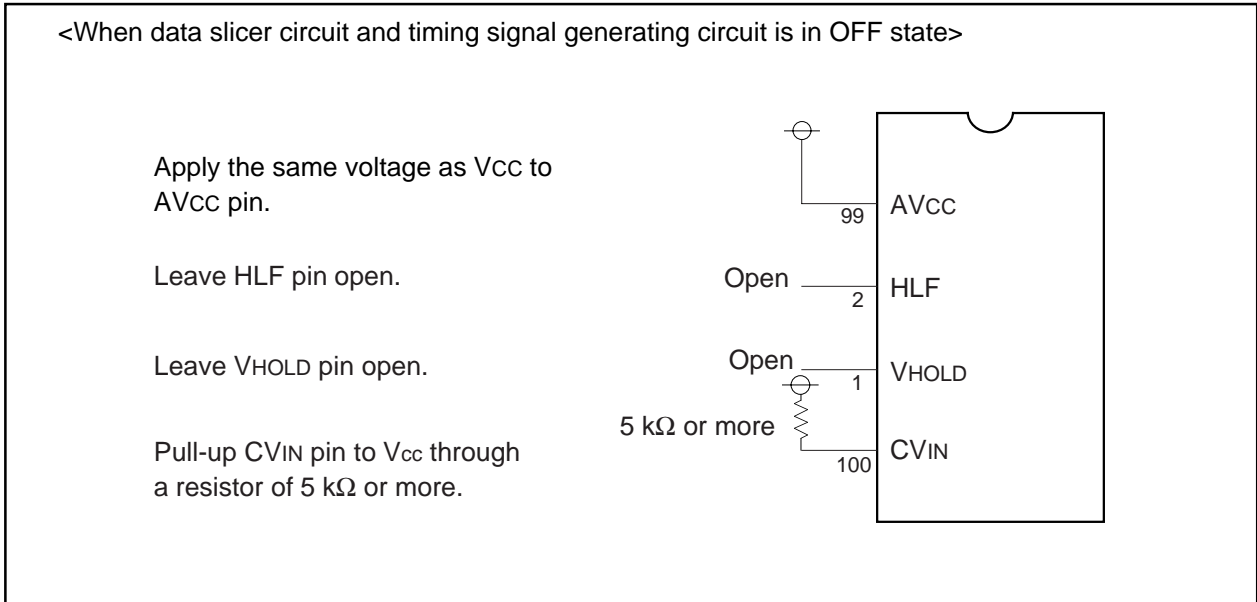


Figure 2.14.1 Data slicer block diagram

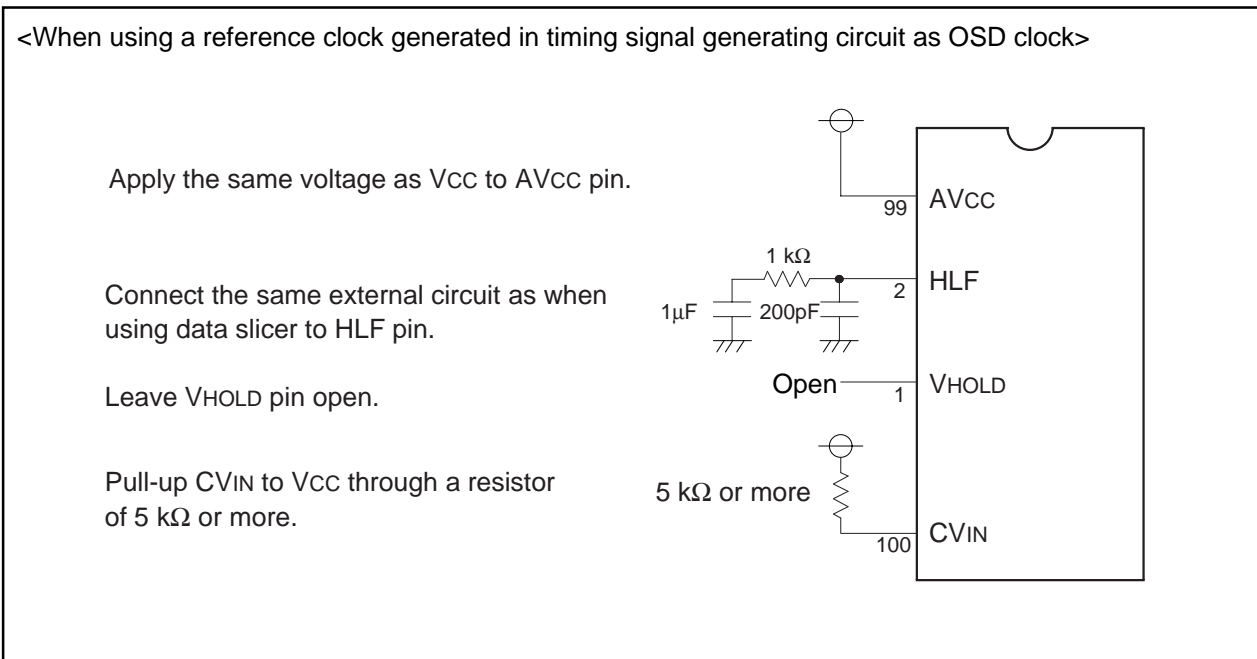
**2.14.1 Notes when not Using Data Slicer**

When bit 0 of data slicer control register 1 (address 026016) is "0," terminate the pins as shown in Figure 2.14.2



**Figure 2.14.2 Termination of data slicer input/output pins when data slicer circuit and timing generating circuit is in OFF state**

When both bits 0 and 2 of data slicer control register 1 (address 026016) are "1," terminate the pins as shown in Figure 2.14.3.



**Figure 2.14.3 Termination of data slicer input/output pins when timing signal generating circuit is in ON state**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

Figures 2.14.4 and 2.14.5 the data slicer control registers.

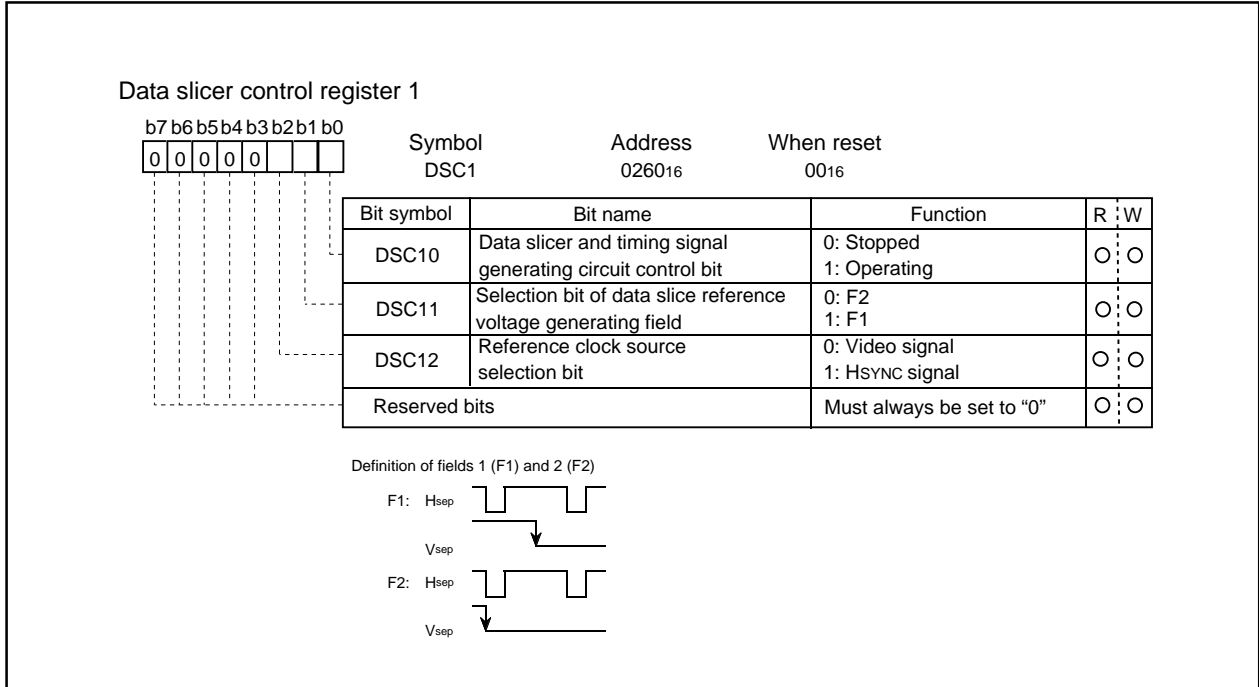


Figure 2.14.4 Data slicer control register 1

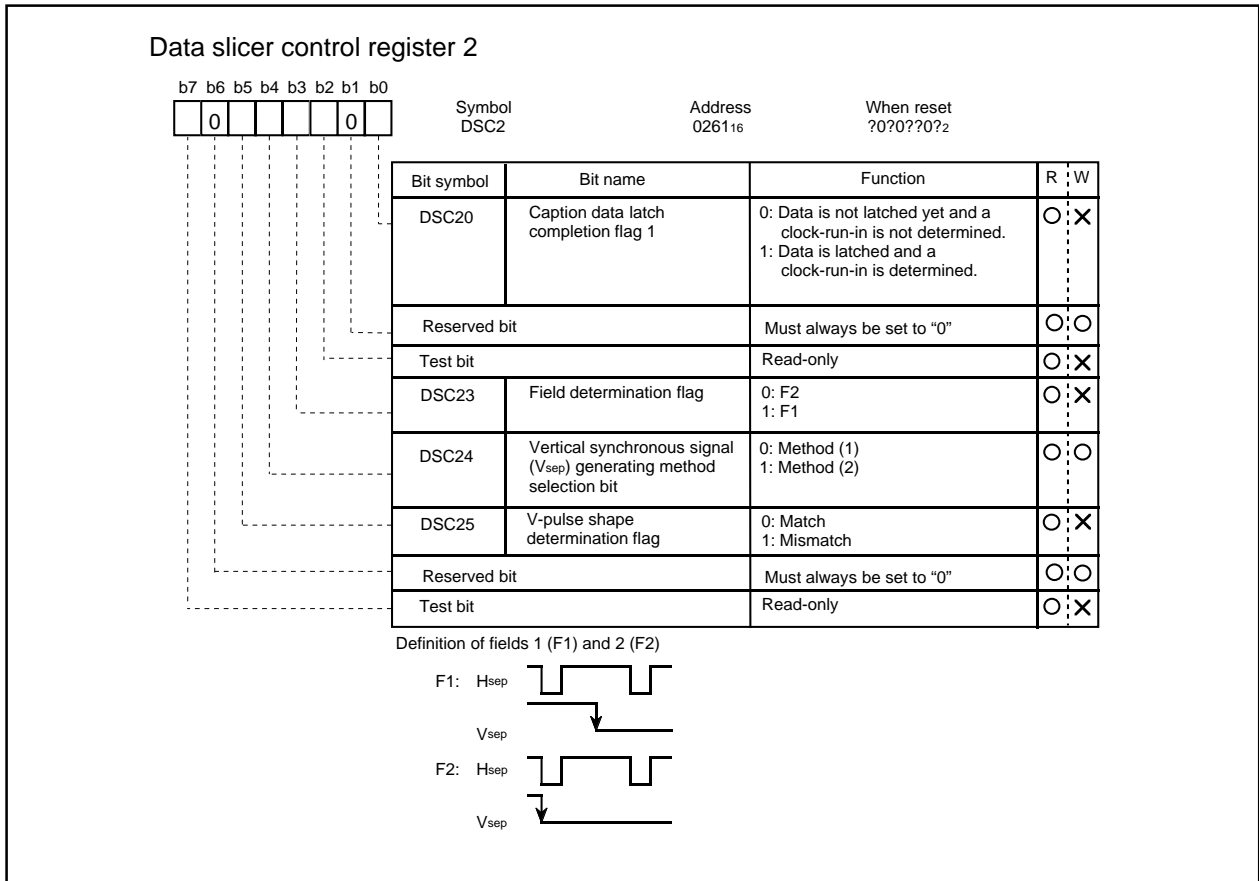


Figure 2.14.5 Data slicer control register 2



### 2.14.2 Clamping Circuit and Low-pass Filter

The clamp circuit clamps the sync chip 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 mF) coupling outside. Pull down the CVIN pin with a resistor of hundreds of kilohms to 1 M $\Omega$ . 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 2.14.1).

### 2.14.3 Sync Slice Circuit

This circuit takes out a composite sync signal from the output signal of the low-pass filter.

### 2.14.4 Synchronous Signal Separation Circuit

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

#### (1) Horizontal synchronous signal (Hsep)

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

#### (2) Vertical synchronous signal (Vsep)

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

- Method 1 The "L" level width of the composite sync signal is measured. If this width exceeds a certain time, a Vsep signal is generated in synchronization with the rising of the timing signal immediately after this "L" level.
- Method 2 The "L" level width of the composite sync signal is measured. If this width exceeds a certain time, it is detected whether a falling of the composite sync signal exits or not in the "L" level period of the timing signal immediately after this "L" level. If a falling exists, a Vsep signal is generated in synchronization with the rising of the timing signal (refer to Figure 2.14.6).

Figure 2.14.6 shows a Vsep generating timing. The timing signal shown in the figure is generated from the reference clock which the timing generating circuit outputs.

Reading bit 5 of data slicer control register 2 permits determining the shape of the V-pulse portion of the composite sync signal. As shown in Figure 2.14.7, when the A level matches the B level, this bit is "0." In the case of a mismatch, the bit is "1."

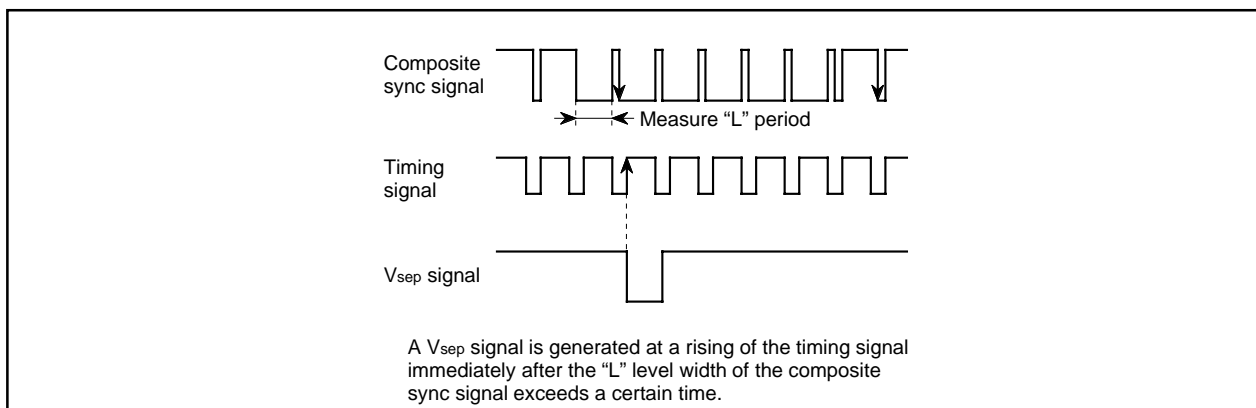


Figure 2.14.6 Vsep generating timing (method 2)

### 2.14.5 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<sub>16</sub>) to "1."

The reference clock can be used as a display clock for OSD function in addition to the data slicer. The HSYNC signal can be used as a count source instead of the composite sync signal. However, when the HSYNC signal is selected, the data slicer cannot be used. A count source of the reference clock can be selected by bit 2 of data slicer control register 1 (address 0260<sub>16</sub>).

For the pins HLF, connect a resistor and a capacitor as shown in Figure 2.14.1 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, Hsep signals and Vsep signals become unstable. For this reason, take stabilization time into consideration when programming.

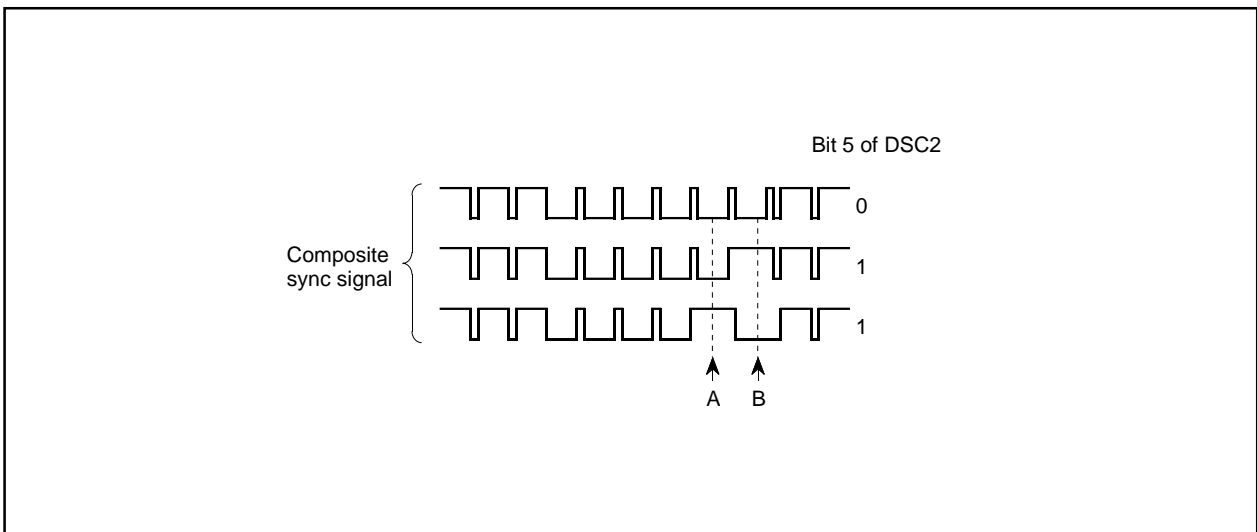


Figure 2.14.7 Determination of v-pulse waveform

## 2.14.6 Data Slice Line Specification Circuit

### (1) Specification of data slice line

This circuit decides a line on which caption data is superimposed. The line 21 (fixed), 1 appropriate line for a period of 1 field (total 2 line for a period of 1 field), and both fields (F1 and F2) are sliced their data. The caption position register (address 026616) is used for each setting (refer to Table 2.14.1). The counter is reset at the falling edge of  $V_{sep}$  and is incremented by 1 every  $H_{sep}$  pulse. When the counter value matched the value specified by bits 4 to 0 of the caption position register, this  $H_{sep}$  is sliced.

The values of "0016" to "1F16" can be set in the caption position register (at setting only 1 appropriate line). Figure 2.14.8 shows the signals in the vertical blanking interval. Figure 2.14.9 shows the caption position register.

### (2) Specification of line to set slice voltage

The reference voltage for slicing (slice voltage) is generated for the clock run-in pulse in the particular line (refer to Table 2.14.1). 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 2.14.1).

### (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  $V_{sep}$ .

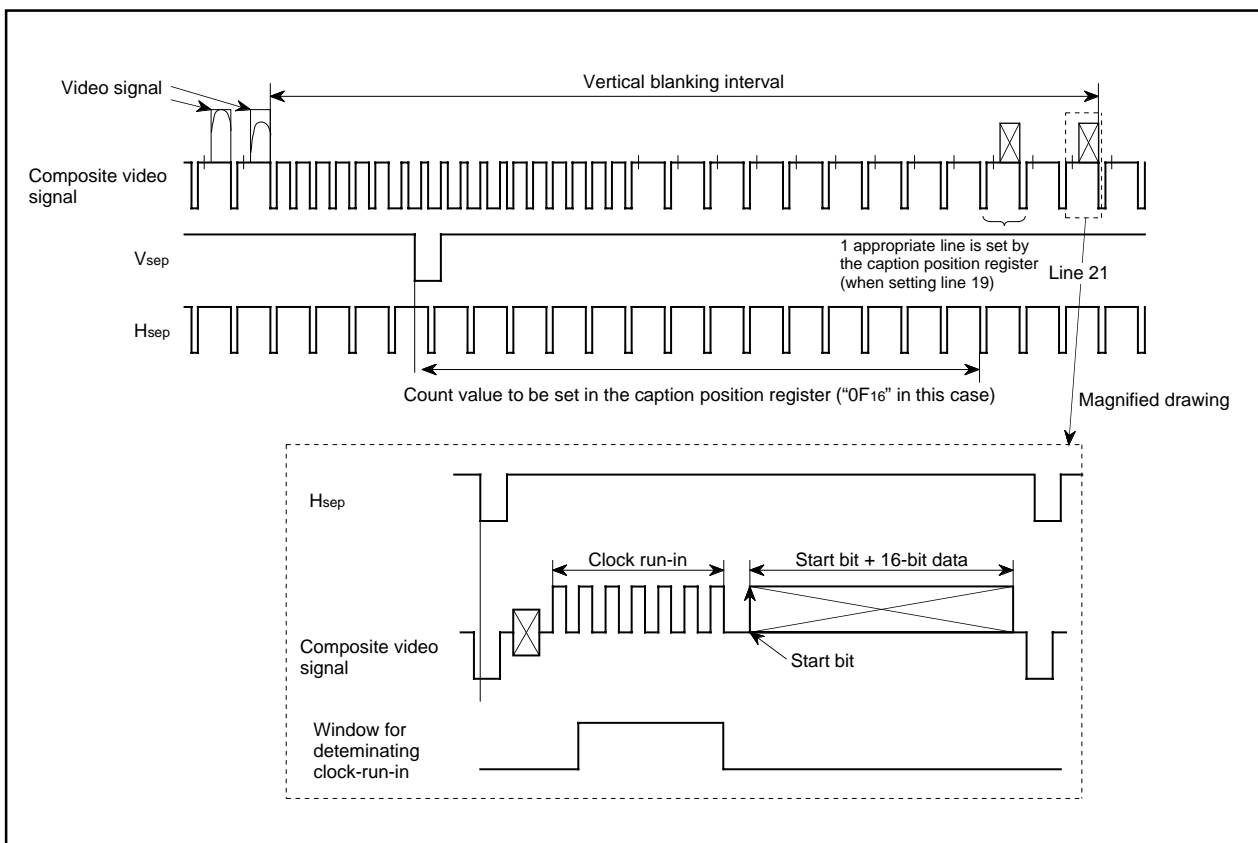
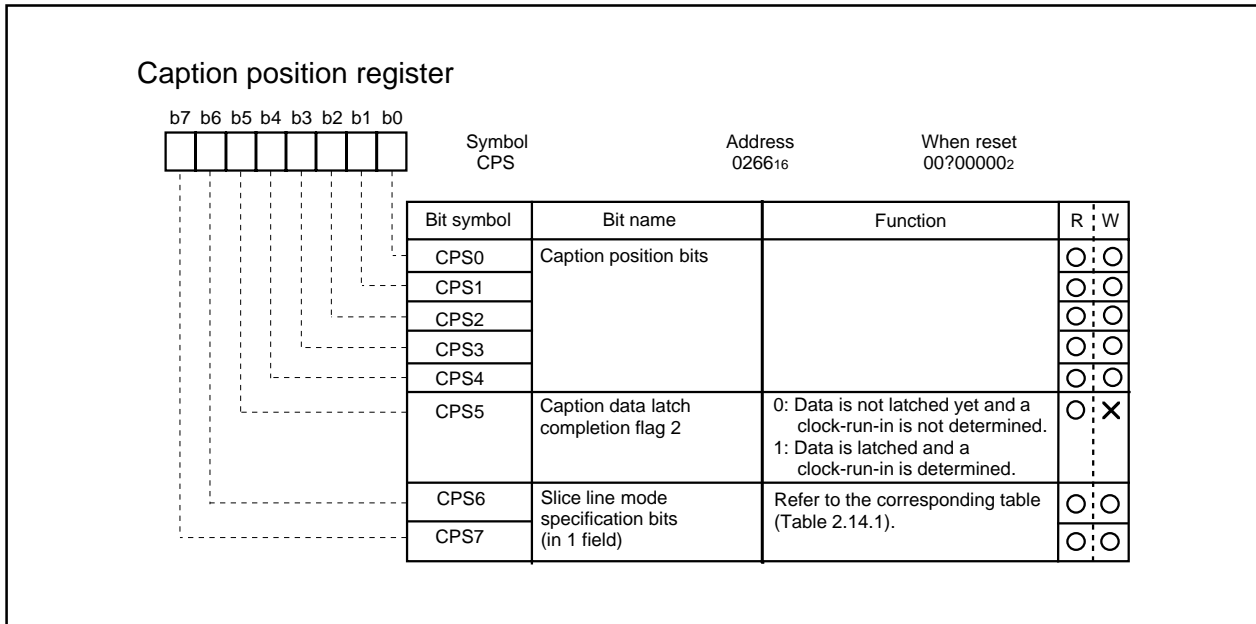


Figure 2.14.8 Signals in vertical blanking interval

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.14.9 Caption position register**

**Table 2.14.1 Specification of data slice line**

CPS		Field and Line to Be Sliced Data	Field and Line to Generate Slice Voltage
b7	b6		
0	0	<ul style="list-style-type: none"> <li>Both fields of F1 and F2</li> <li>Line 21 and a line specified by bits 4 to 0 of CPS (total 2 lines) (See note 2)</li> </ul>	<ul style="list-style-type: none"> <li>Field specified by bit 1 of DSC1</li> <li>Line 21 (total 1 line)</li> </ul>
0	1	<ul style="list-style-type: none"> <li>Both fields of F1 and F2</li> <li>A line specified by bits 4 to 0 of CPS (total 1 line) (See note 3)</li> </ul>	<ul style="list-style-type: none"> <li>Field specified by bit 1 of DSC1</li> <li>A line specified by bits 4 to 0 of CPS (total 1 line) (See note 3)</li> </ul>
1	0	<ul style="list-style-type: none"> <li>Both fields of F1 and F2</li> <li>Line 21 (total 1 line)</li> </ul>	<ul style="list-style-type: none"> <li>Field specified by bit 1 of DSC1</li> <li>Line 21 (total 1 line)</li> </ul>
1	1	<ul style="list-style-type: none"> <li>Both fields of F1 and F2</li> <li>Line 21 and a line specified by bits 4 to 0 of CPS (total 2 lines) (See note 2)</li> </ul>	<ul style="list-style-type: none"> <li>Field specified by bit 1 of DSC1</li> <li>Line 21 and a line specified by bits 4 to 0 of CPS (total 2 lines) (See note 2)</li> </ul>

**Notes 1:** DSC is data slicer control register 1.

CPS is caption position register.

**2:** Set "00<sub>16</sub>" to "10<sub>16</sub>" to bits 4 to 0 of CPS.

**3:** Set "00<sub>16</sub>" to "1F<sub>16</sub>" to bits 4 to 0 of CPS.

### 2.14.7 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) 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.

#### (2) Comparator

The comparator 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.

### 2.14.8 Start Bit Detecting Circuit

This circuit detects a start bit at line decided in the data slice line specification circuit.

The detection of a start bit is described below.

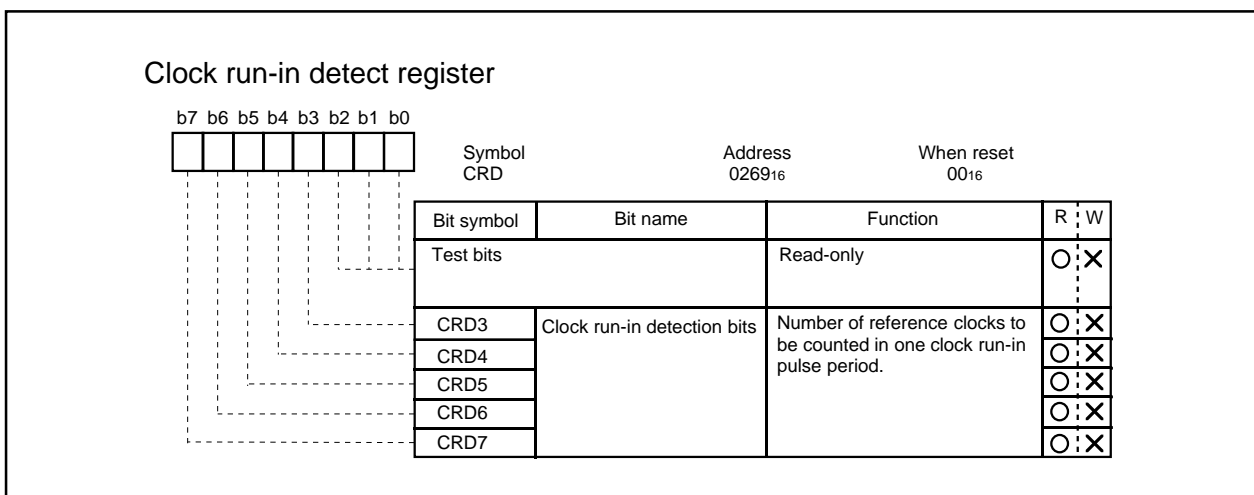
- ① A sampling clock is generated by dividing the reference clock output by the timing signal.
- ② A clock run-in pulse is detected by the sampling clock.
- ③ After detection of the pulse, a start bit pattern is detected from the comparator output.

### 2.14.9 Clock Run-in Determination Circuit

This circuit determinates clock run-in by counting the number of pulses in a window of the composite video signal.

The reference clock count value in one pulse cycle is stored in bits 3 to 7 of the clock run-in detect register (address 0269<sub>16</sub>). Read out these bits after the occurrence of a data slicer interrupt (refer to 2.14.12 Interrupt request generating circuit).

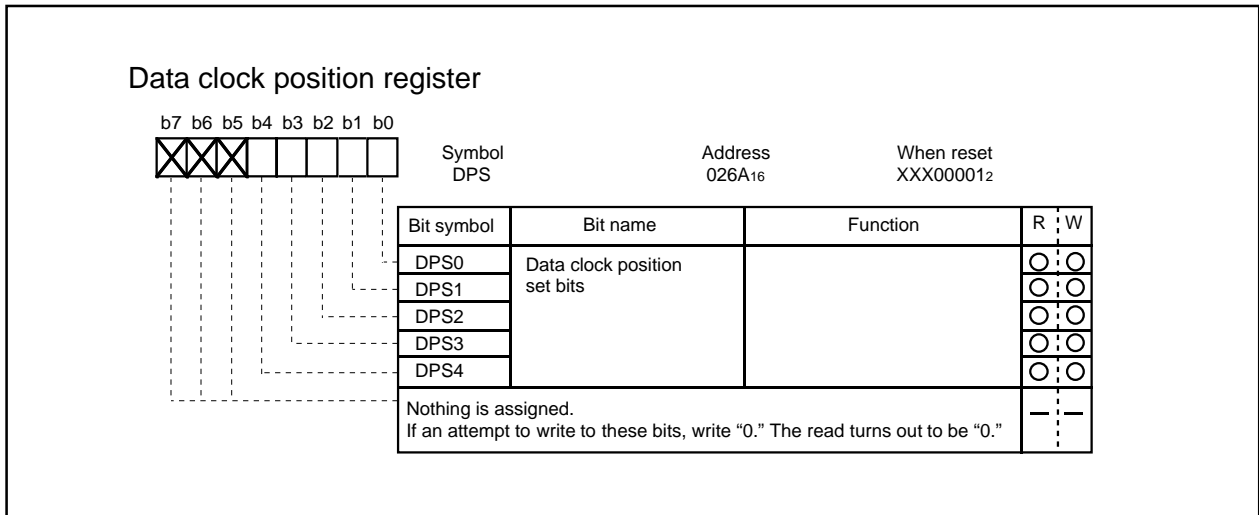
Figure 2.14.10 shows the structure of clock run-in detect register.



**Figure 2.14.10 Clock run-in detect register**

**2.14.10 Data Clock Generating Circuit**

This circuit generates a data clock synchronized with the start bit detected in the start bit detecting circuit. The data clock stores caption data to the 16-bit shift register. When the 16-bit data has been stored and the clock run-in determination circuit determines clock run-in, the caption data latch completion flag is set. This flag is reset at a falling of the vertical synchronous signal ( $V_{sep}$ ).



**Figure 2.14.11 Data clock position register**

**2.14.11 16-bit Shift Register**

The caption data converted into a digital value by the comparator is stored into the 16-bit shift register in synchronization with the data clock. The contents of the stored caption data can be obtained by reading out the caption data register 1 (addresses 0263<sub>16</sub>, 0262<sub>16</sub>) and caption data register 2 (addresses 0265<sub>16</sub>, 0264<sub>16</sub>). These registers are reset to "0" at a falling of  $V_{sep}$ . Read out data registers 1 and 2 after the occurrence of a data slicer interrupt (refer to "2.14.12 Interrupt request generating circuit").

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**2.14.12 Interrupt Request Generating Circuit**

The interrupt requests as shown in Table 2.14.3 are generated by combination of the following bits; bits 6 and 7 of the caption position register (address 0266<sub>16</sub>). Read out the contents of data registers 1, 2 and the contents of bits 3 to 7 of the clock run-in detect register after the occurrence of a data slicer interrupt request.

**Table 2.14.2 Contents of caption data latch completion flag and 16-bit shift register**

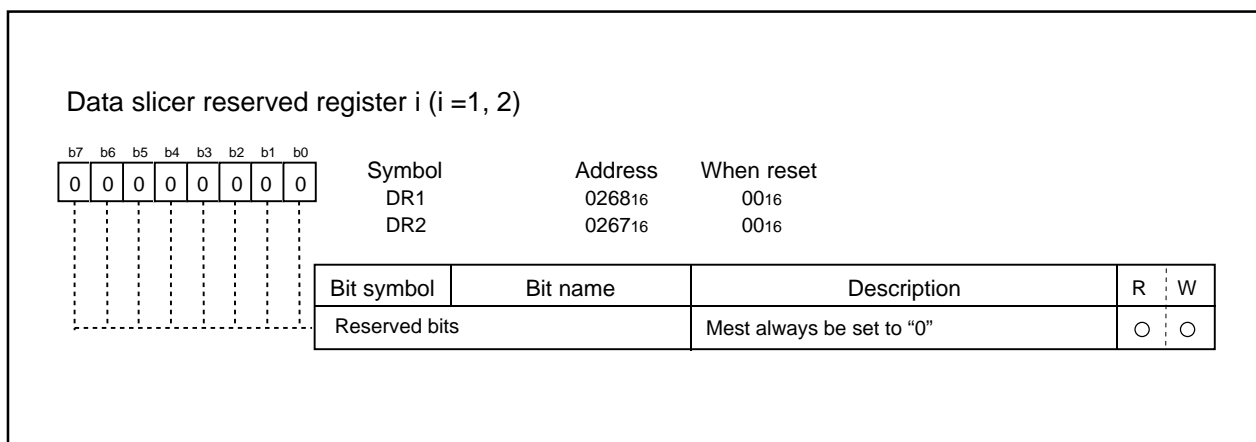
Slice Line Specification Mode		Contents of Caption Data Latch Completion Flag		Contents of 16-bit Shift Register	
CPS		Completion Flag 1 (bit 0 of DSC2)	Completion Flag 2 (bit 5 of CPS)	Caption Data Register 1	Caption Data Register 2
bit 7	bit 6				
0	0	Line 21	A line specified by bits 4 to 0 of CPS	16-bit data of line 21	16-bit data of a line specified by bits 4 to 0 of CPS
0	1	A line specified by bits 4 to 0 of CPS	Invalid	16-bit data of a line specified by bits 4 to 0 of CPS	Invalid
1	0	Line 21	Invalid	16-bit data of line 21	Invalid
1	1	Line 21	A line specified by bits 4 to 0 of CPS	16-bit data of line 21	16-bit data of a line specified by bits 4 to 0 of CPS

CPS: Caption position register  
DSC2: Data slicer control register 2

**Table 2.14.3 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
1	0	After slicing line 21
	1	After slicing line 21

CPS: Caption position register



**Figure 2.14.12 Data slicer reserved register i (i = 1, 2)**

2.15 HSYNC Counter

The synchronous signal counter counts HSYNC from HSYNC count input pins (HC0/P7<sub>5</sub>, HC1/P7<sub>7</sub>) 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 f<sub>32</sub> is stored into the 8-bit latch.

Accordingly, the latch value changes in the cycle of T time. When the count value exceeds “FF<sub>16</sub>,” “FF<sub>16</sub>” is stored into the latch.

The latch value can be obtained by reading out the HSYNC counter latch (address 027F<sub>16</sub>). A count source and count update cycle (T time) are selected by bits 0, 3 and 4 of the HSYNC counter register.

Figure 2.15.1 shows the HSYNC counter and Figure 2.15.2 shows the synchronous signal counter block diagram.

**Note:** When using the HSYNC counter, set bit 7 of the peripheral mode register (address 027D<sub>16</sub>) according to the main clock frequency.

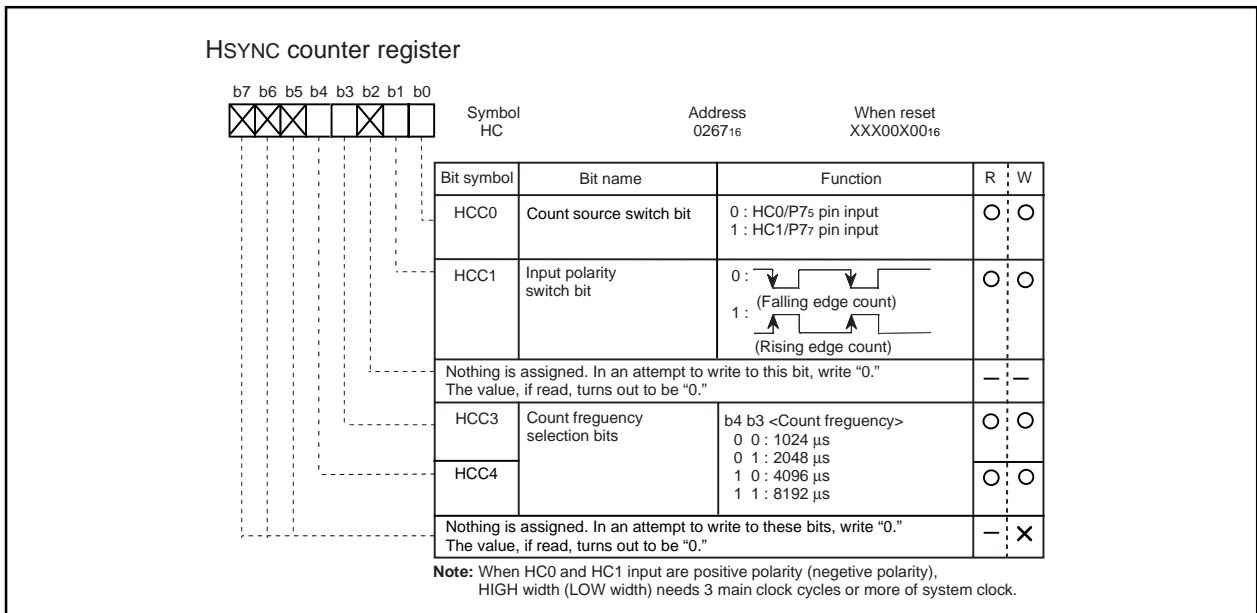


Figure 2.15.1 HSYNC counter register

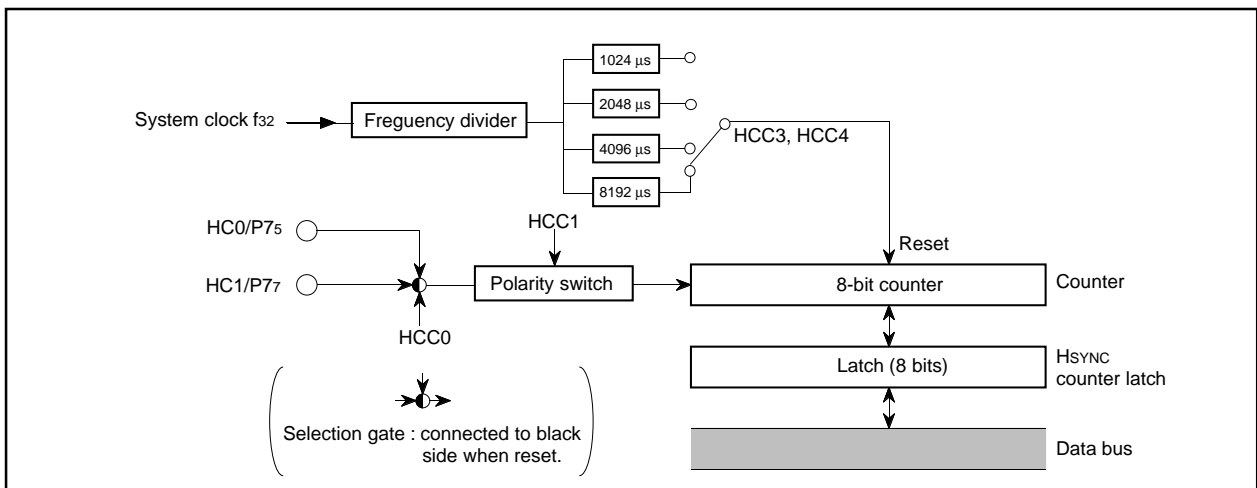


Figure 2.15.2 HSYNC counter block diagram



**2.16 OSD Functions**

Table 2.16.1 outlines the OSD functions of this microcomputer. This OSD function can display the following: the block display (32 characters X 16 lines or 42 characters X 16 lines) and the SPRITE display, and can display the both display at the same time. There are 3 display modes and they are selected by a block unit. The display modes are selected by block control register i (i = 1 to 16). The features of each display are described below.

**Note:** When using OSD function, select “No-division mode” as BCLK operating mode and set the main clock frequency to  $f(XIN) = 10 \text{ MHz}$ .

**Table 2.16.1 Features of each display style**

Display style Parameter		Block display			SPRITE display
		CC mode (Closed caption mode)	OSD mode (On-screen display mode)		
			OSDS mode	OSDL mode	
Number of display characters		32 characters X 16 lines/42 characters X 16 lines			1 character
Dot structure		16 X 20 dots (Character display area: 16 X 26 dots)	24 X 32 dots	16 X 26 dots	16 X 20 dots
Kinds of character ROM	OSDL enable mode	254 kinds		254 kinds	62 kinds
	OSDL disable mode	508 kinds	254 kinds	62 kinds	1 kind
Font memory		ROM			RAM
Kinds of character sizes (See note 1)		4 kinds	14 kinds	12 kinds	14 kinds
Pre-divide ratio (Note)		X 1, X 2	X 1, X 2, X 3		
Dot size		1Tc X 1/2H, 1Tc X 1H	1TC X 1/2H, 1TC X 1H, 1.5TC X 1/2H, 1.5TC X 1H, 2TC X 2H, 3TC X 3H	1TC X 1/2H, 1TC X 1H, 2TC X 2H, 3TC X 3H	1TC X 1/2H, 1TC X 1H, 1.5TC X 1/2H, 1.5TC X 1H, 2TC X 2H, 3TC X 3H
Attribute		Smooth italic, under line, flash	Border		
Character font coloring		1 screen: 8 kinds (a character unit) Max. 64 kinds	1 screen: 16 kinds (a character unit) Max. 64 kinds	1 screen: 8 kinds (a dot unit) (only specified dots are colored by a character unit) Max. 64 kinds	1 screen: 8 kinds (a dot unit) Max. 64 kinds
Character background coloring		Possible (a character unit, 1 screen: 4 kinds, Max. 64 kinds)	Possible (a character unit, 1 screen: 16 kinds, Max. 64 kinds)		
Display layer		Layer 1	Layer 1 and layer 2		Layer 3 (with highest priority)
OSD output (See note 2)		Analog R, G, B output (each 4 adjustment levels: 64 colors), Digital OUT1, OUT2 output			
Raster coloring		Possible (a screen unit, max 64 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 divide ratio of the frequency divider (the pre-divide circuit) is referred as “pre-divide ratio” hereafter.  
**2:** The character size is specified with dot size and pre-divide ratio (refer to “2.16.3 Dot Size”).  
**3:** As for SPRITE display, the window function does not operate.  
**4:** The divide ratio of the frequency divider (the pre-divide circuit) is referred as “pre-divide ratio” hereafter.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

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 2.16.1 shows the configuration of OSD character display area. Figure 2.16.2 shows the block diagram of the OSD circuit. Figure 2.16.3 shows the OSD control register 1. Figure 2.16.4 shows the block control register i.

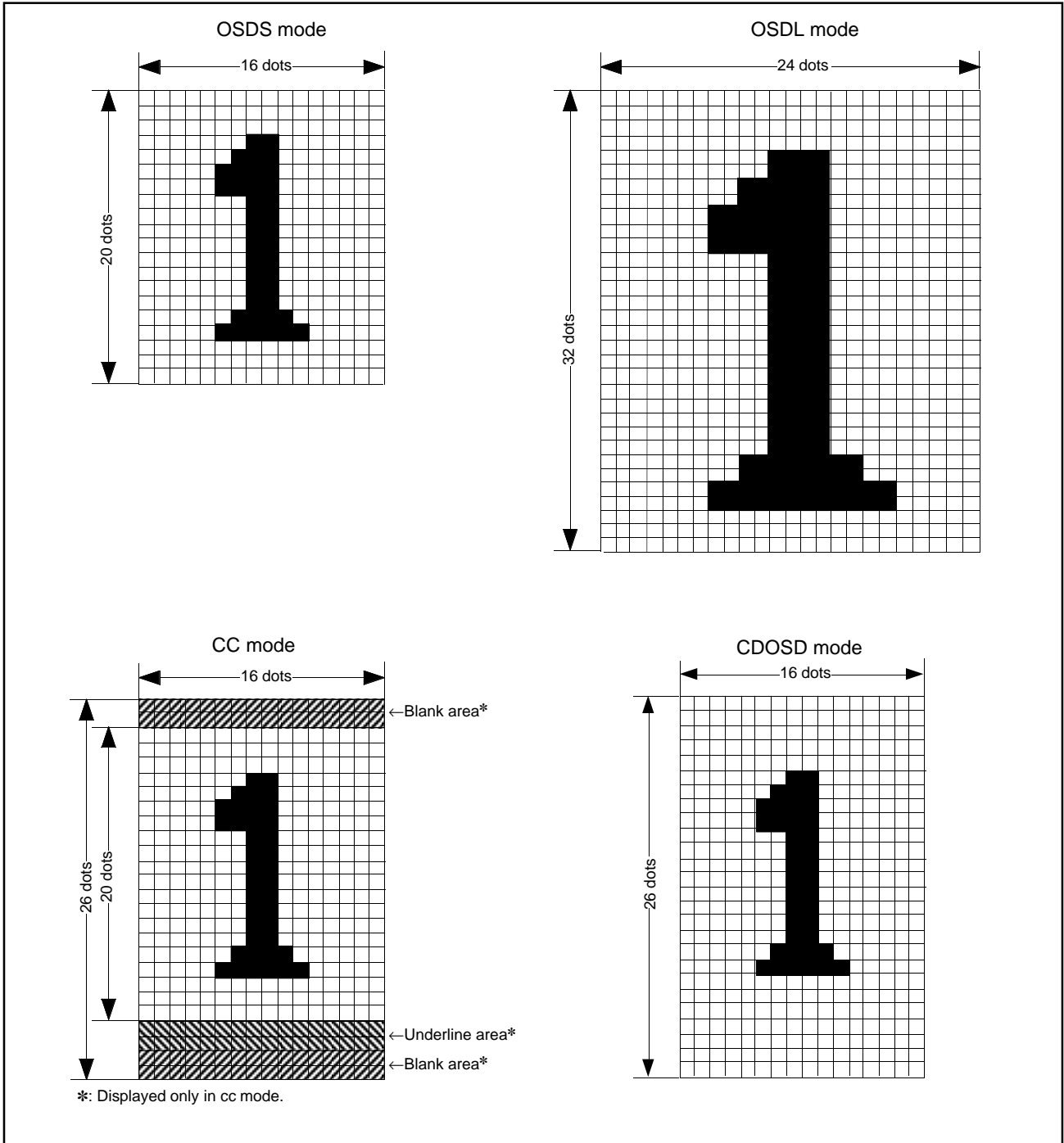


Figure 2.16.1 Configuration of OSD character display area

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

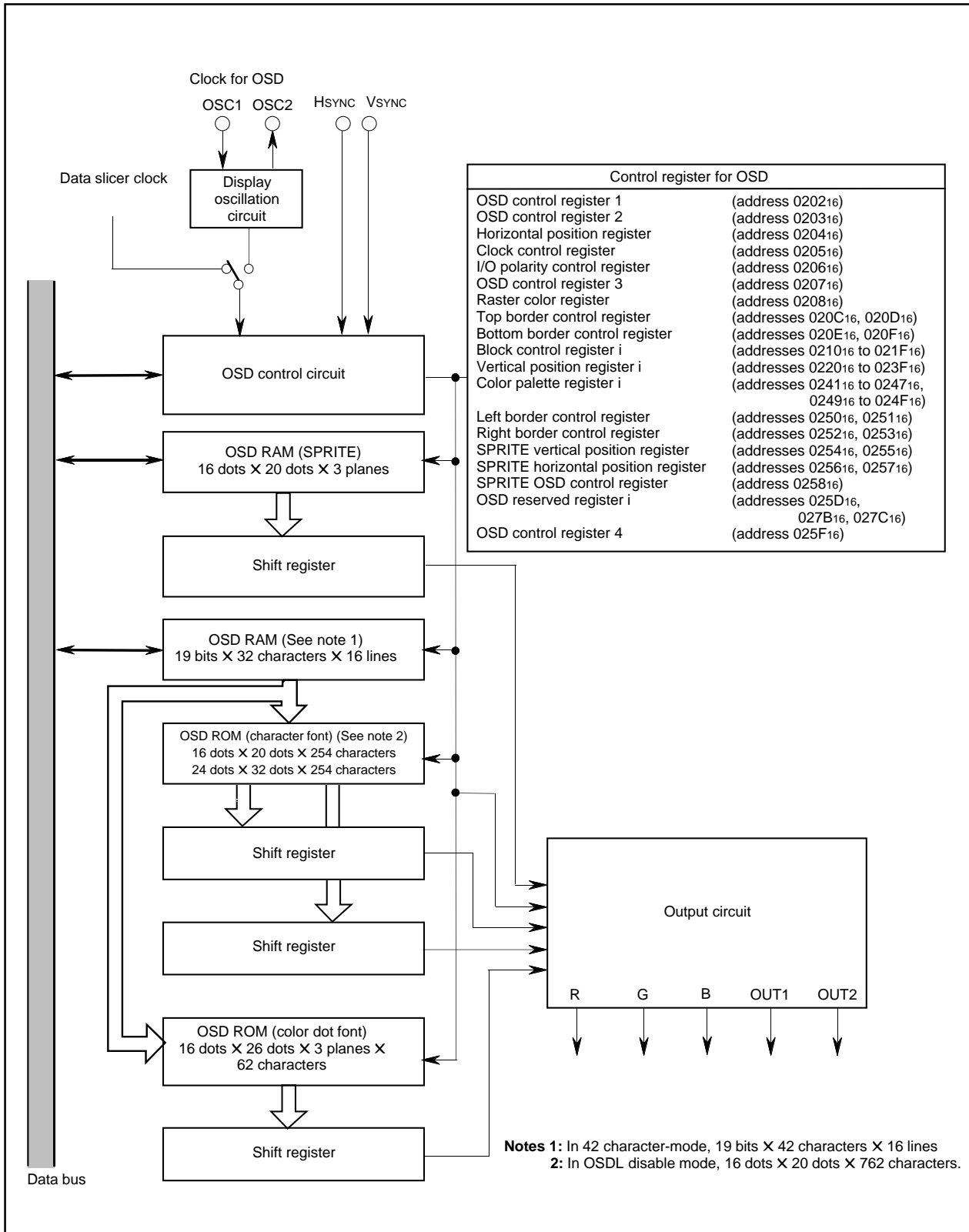


Figure 2.16.2 Block diagram of OSD circuit

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

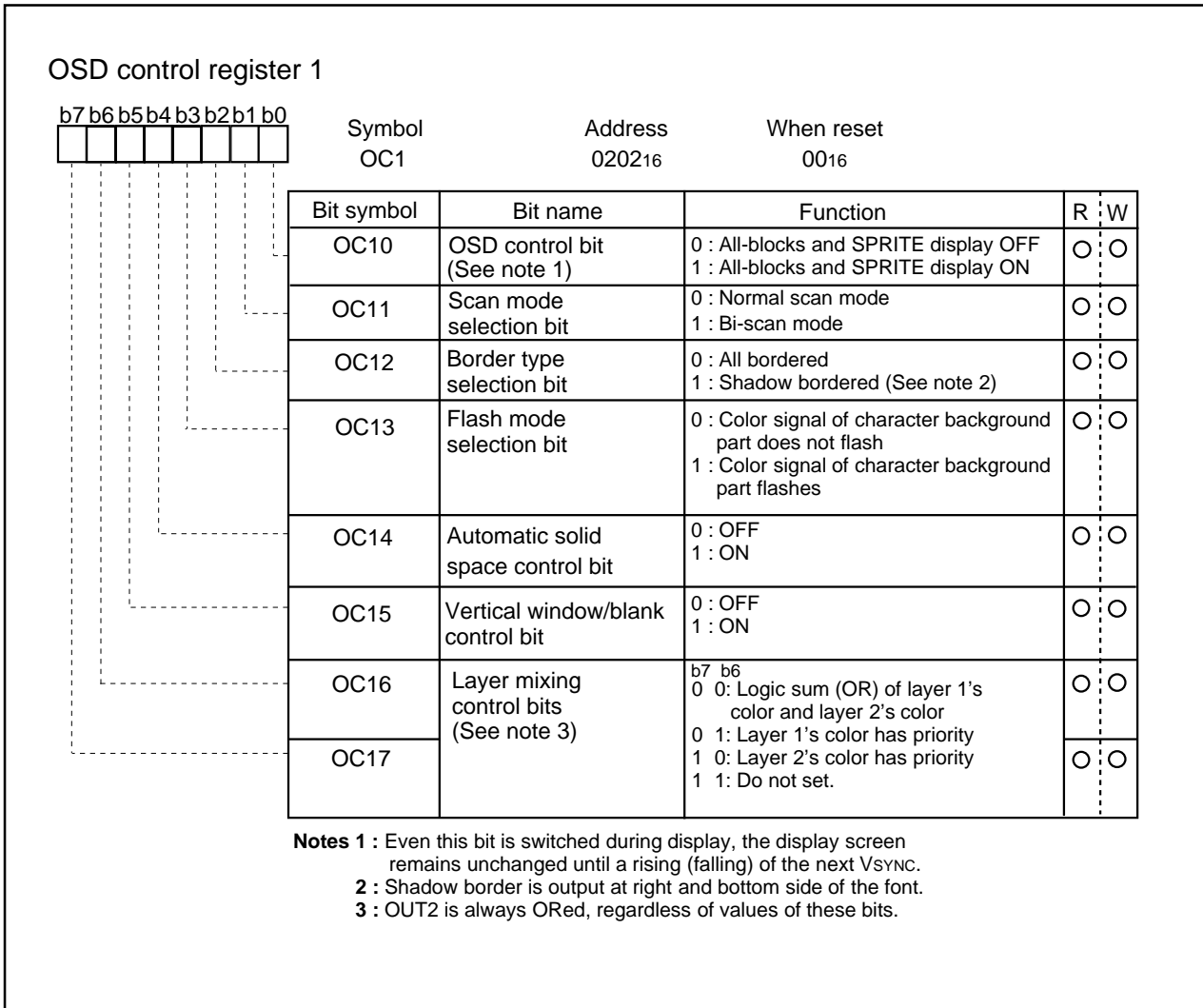


Figure 2.16.3 OSD control register 1

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

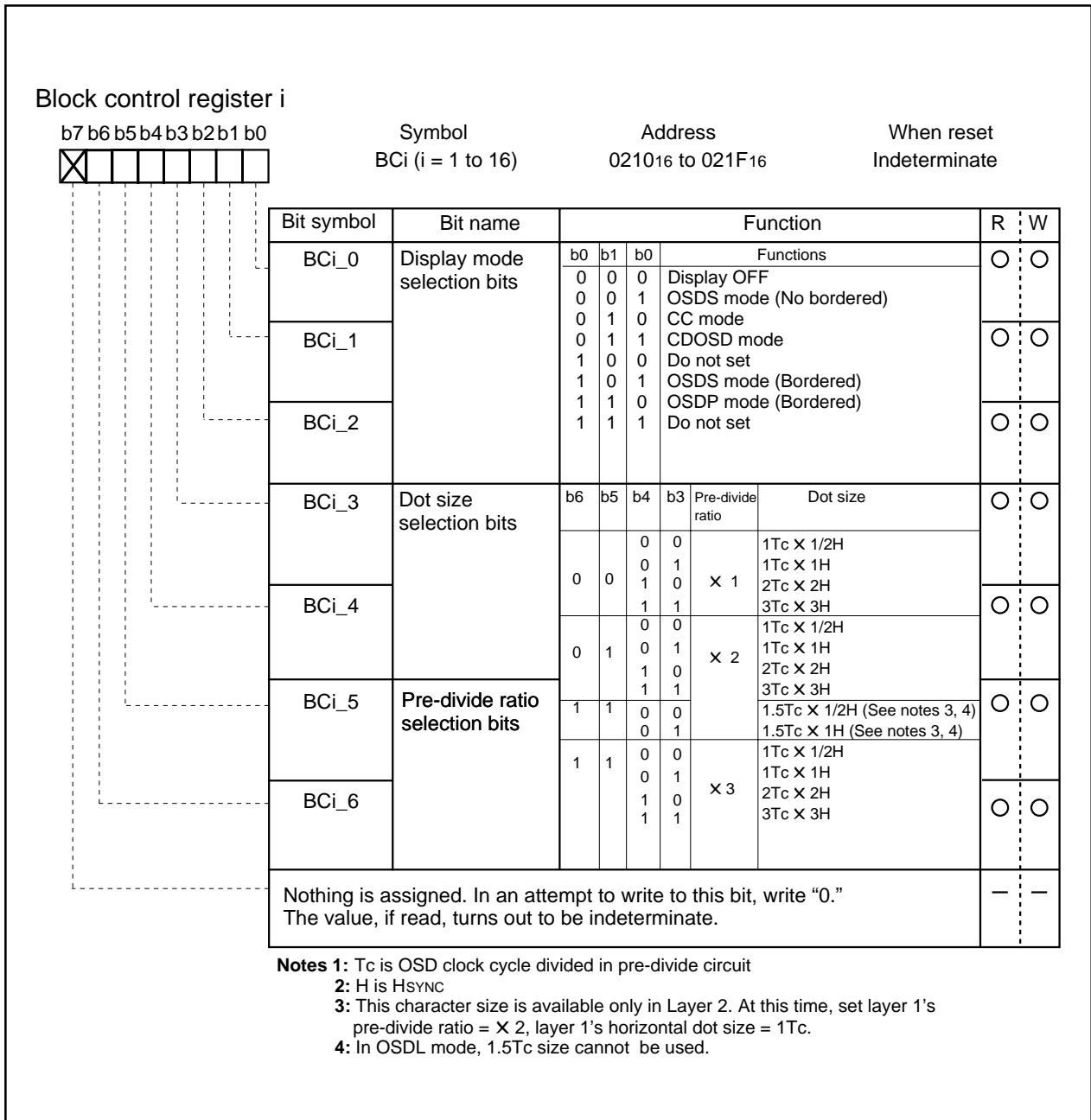


Figure 2.16.4 Block control register i (i = 0 to 16)

### 2.16.1 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 2.16.7). 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 2.16.5. Layer 3 always takes display priority of layers 1 and 2.

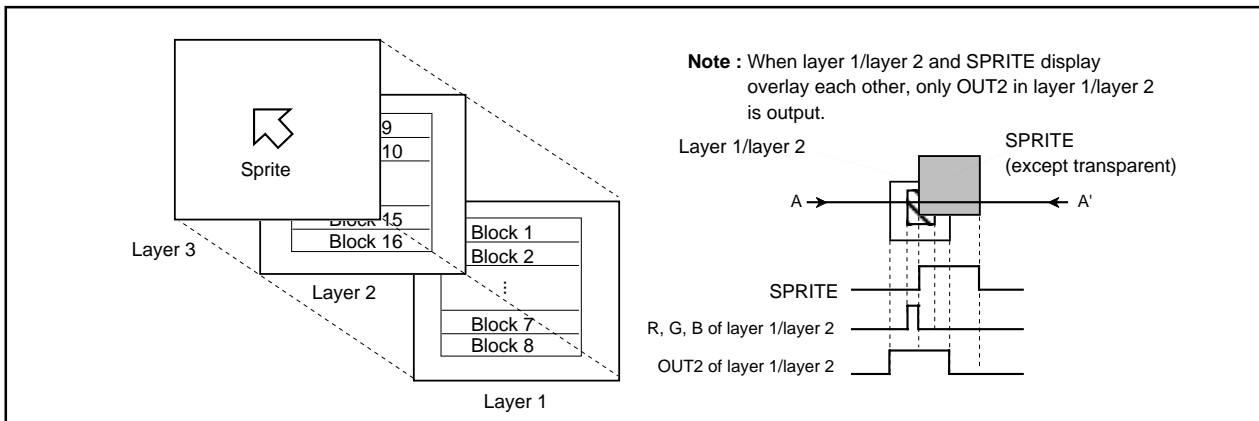
**Notes 1:** When mixing layer 1 and layer 2, note Table 2.16.2.

**2:** 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 2.16.2** Mixing layer 1 and layer 2

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

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



**Fig 2.16.5** Triple layer OSD

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

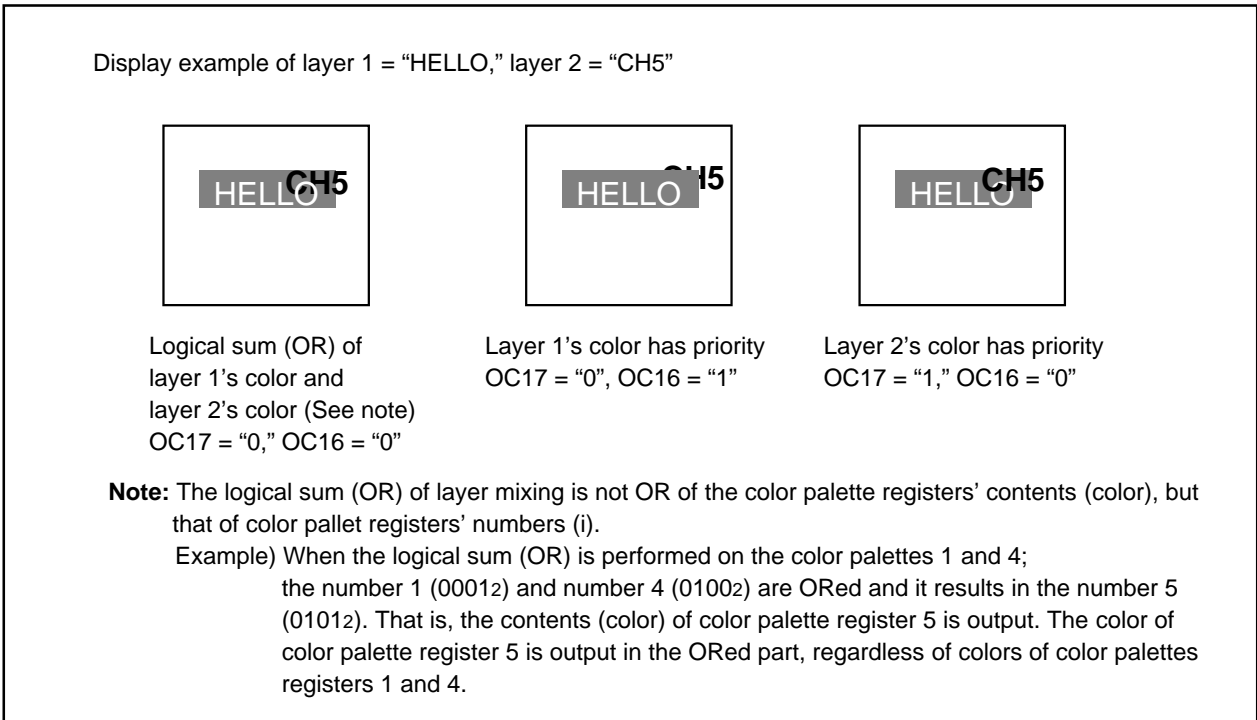


Figure 2.16.6 Display example of triple layer OSD

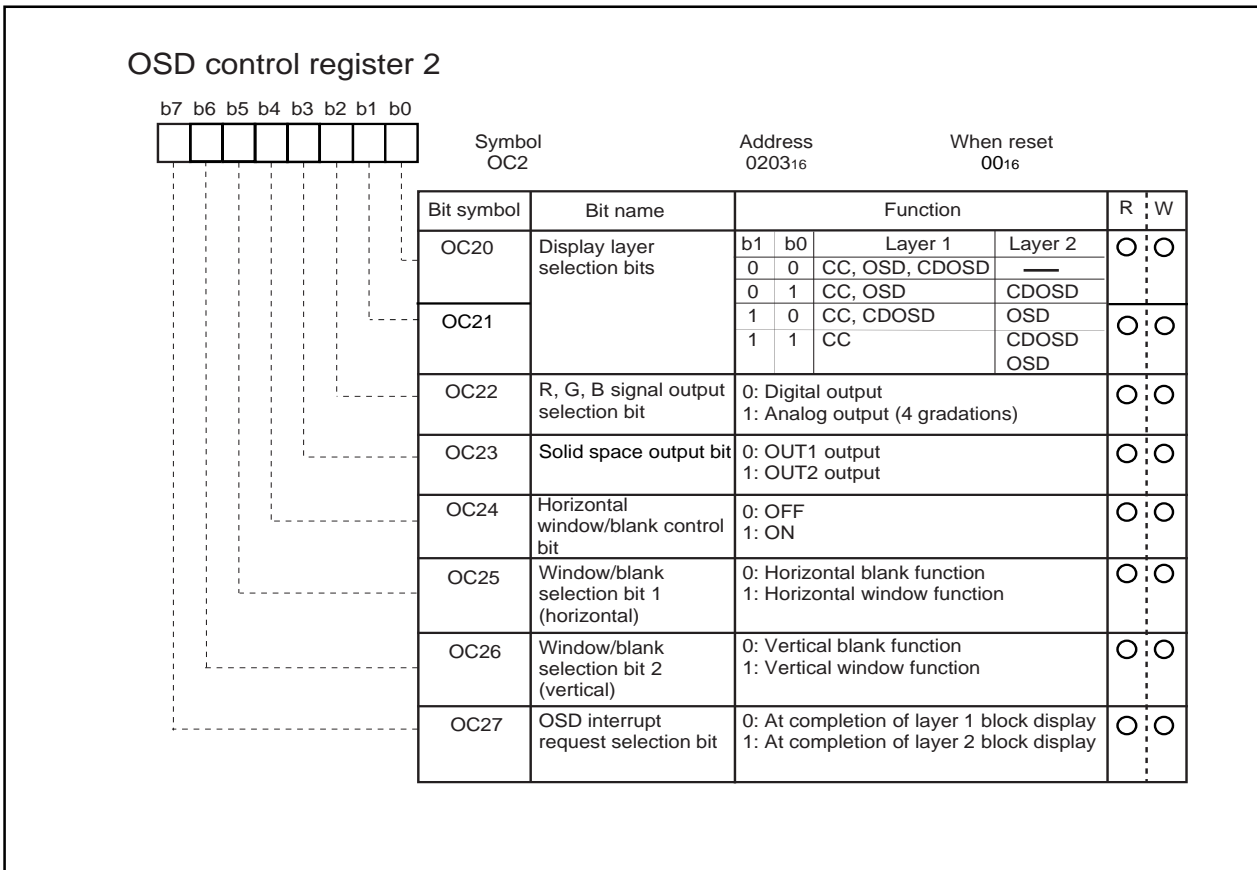


Figure 2.16.7 OSD control register 2

### 2.16.2 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 2.16.6 Memory for OSD).

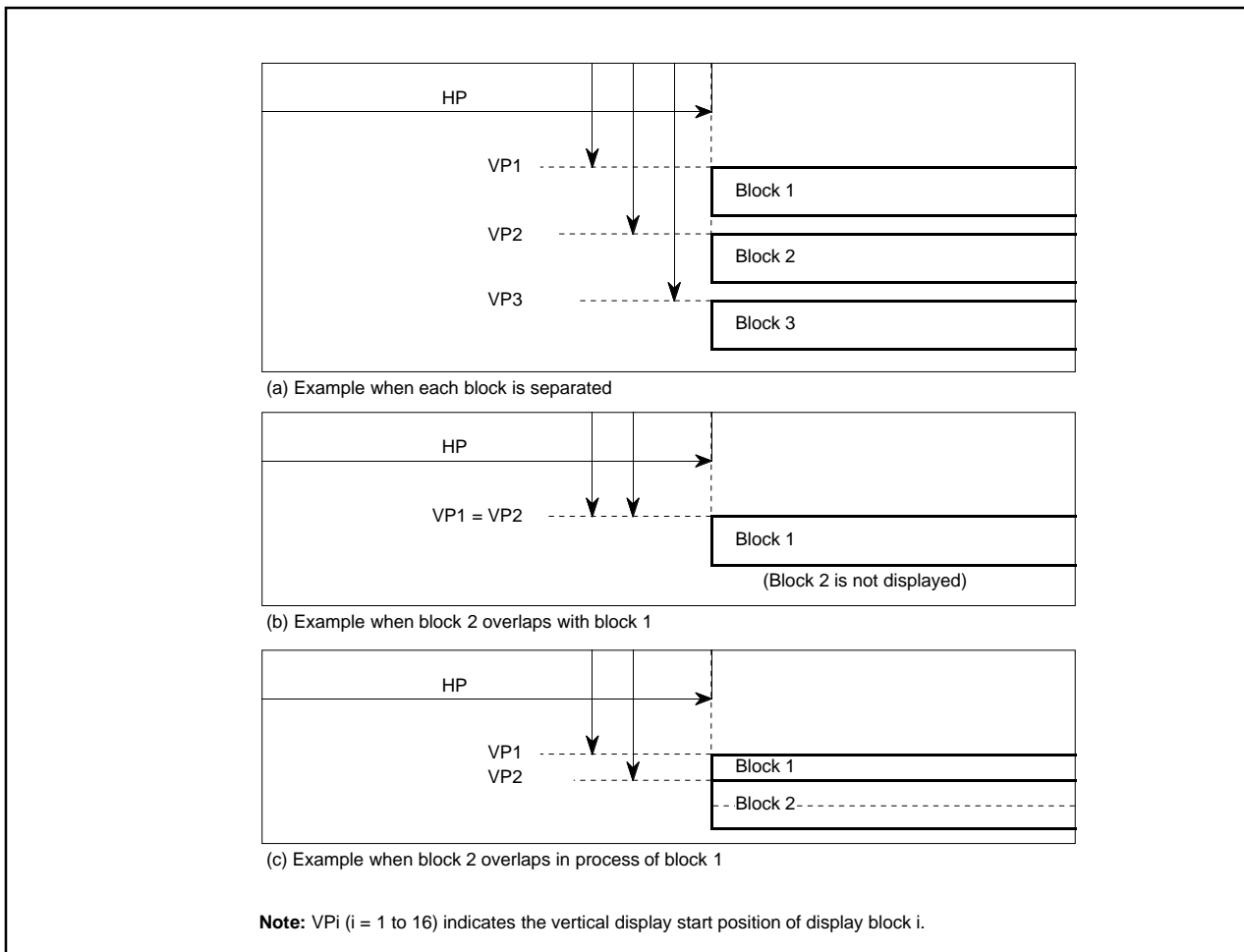
The display position of each block can be set in both horizontal and vertical directions by software.

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

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

Blocks are displayed in conformance with the following rules:

- When the display position is overlapped with another block (Figure 2.16.8 (b)), a lower block number (1 to 16) is displayed on the front.
- When another block display position appears while one block is displayed (Figure 2.16.8 (c)), the block with a larger set value as the vertical display start position is displayed. However, do not display block with the dot size of 2Tc X 2H or 3Tc X 3H during display period (\*) of another block.
  - \* In the case of OSD mode block: 20 dots in vertical from the vertical display start position.
  - \* In the case of OSDL mode block: 32 dots in vertical from the vertical display start position.
  - \* In the case of CC or CDOSD mode block: 26 dots in vertical from the vertical display start position.



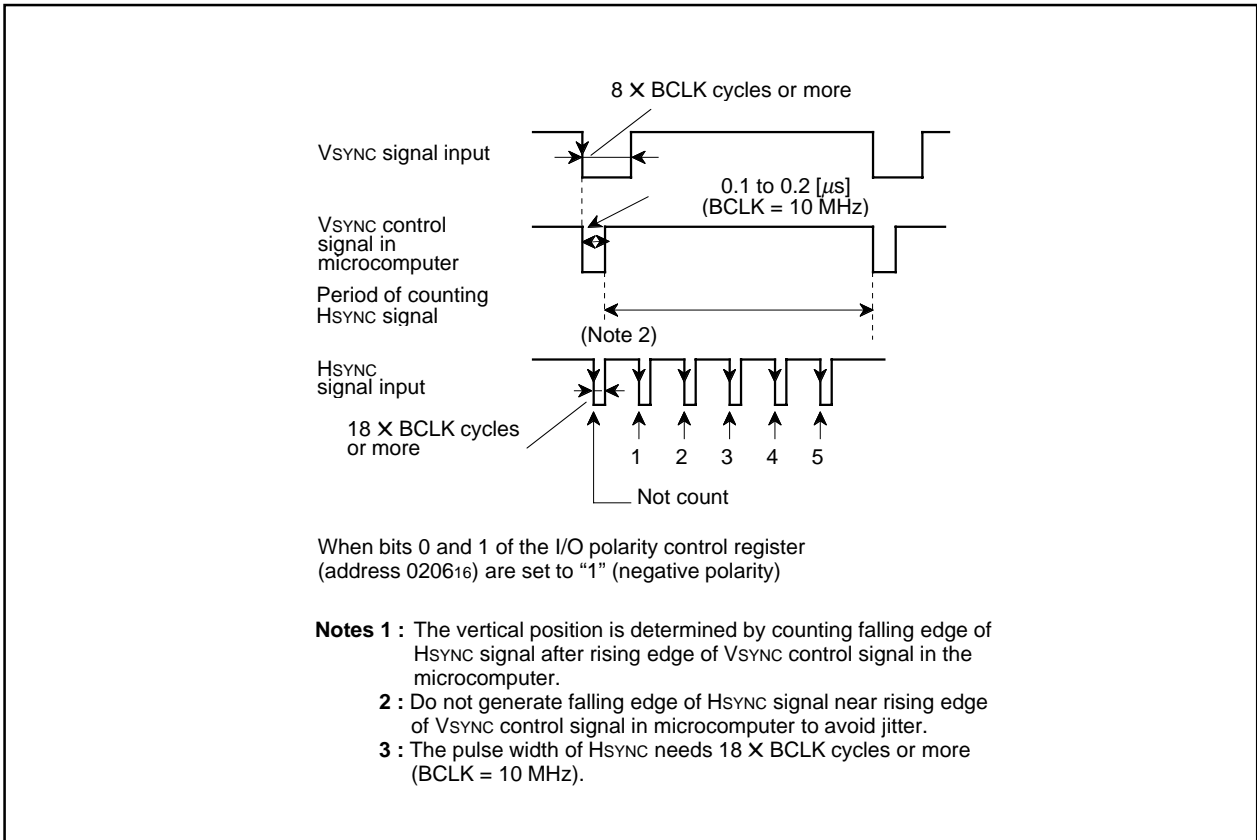
**Figure 2.16.8 Display position**



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

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 X BCLK cycles or more) for avoiding jitter. The polarity of HSYNC and VSYNC signals can select with the I/O polarity control register (address 020616).

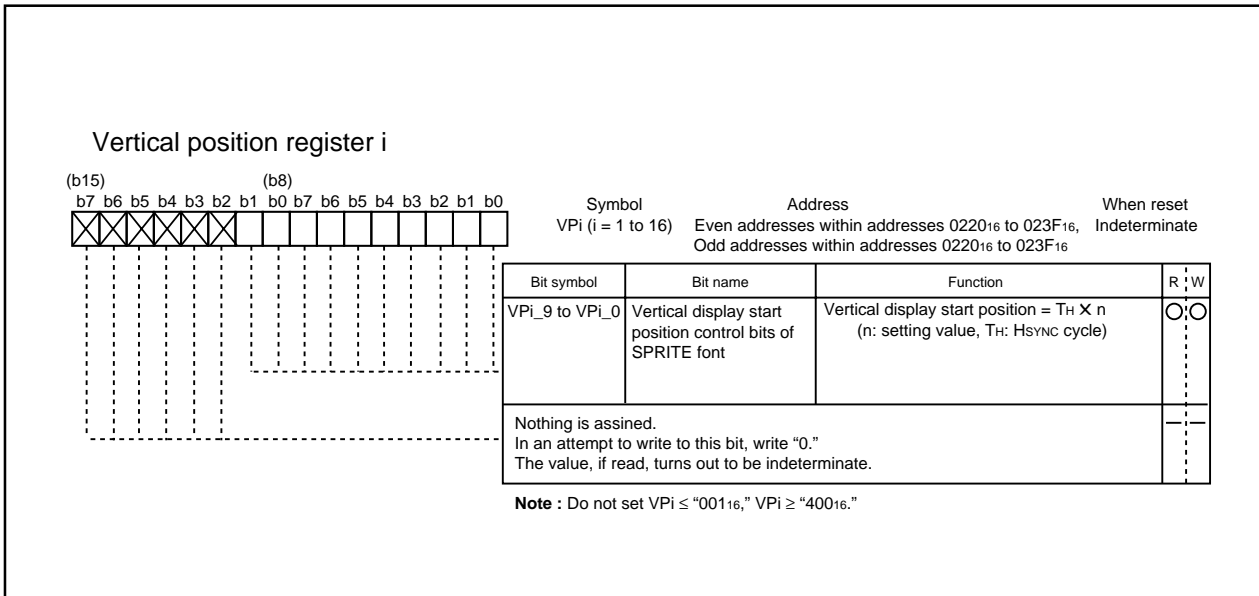


**Figure 2.16.9 Supplement explanation for display position**

**M306V0ME-XXXFP**  
**M306V0EEFP**

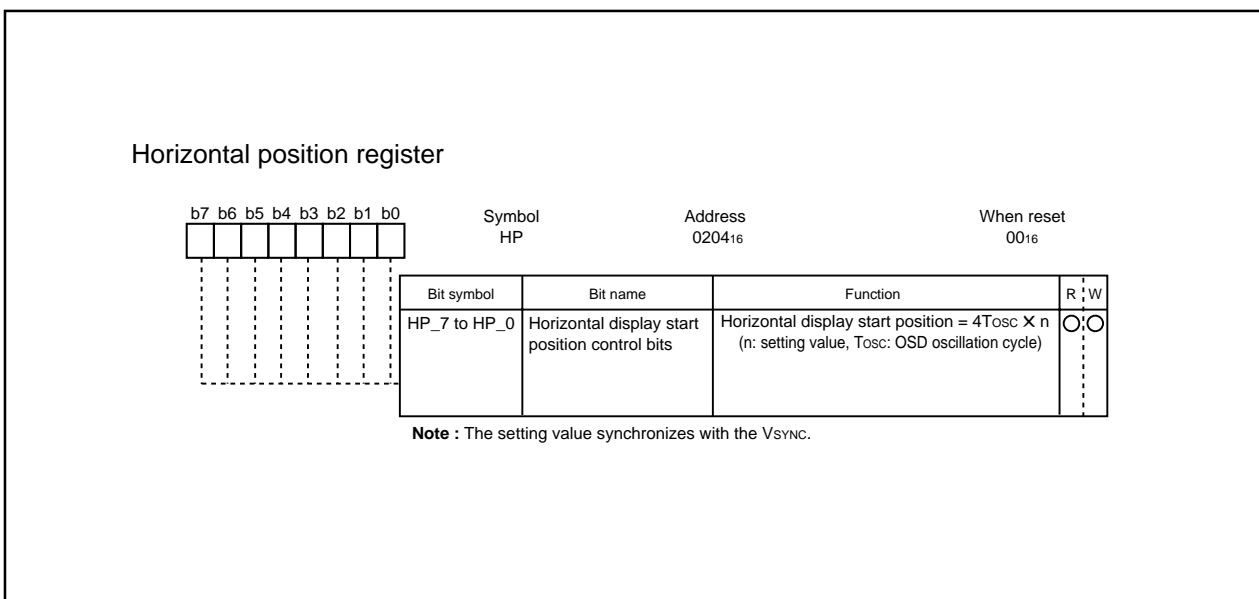
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

The vertical position for each block can be set in 1024 steps (where each step is 1<sub>TH</sub> (TH: Hsync cycle)) as values "002<sub>16</sub>" to "3FF<sub>16</sub>" in vertical position register i (i = 1 to 16) (addresses 0220<sub>16</sub> to 023F<sub>16</sub>). The vertical position register i is shown in Figure 2.16.10.



**Figure 2.16.10 Vertical position register i (i = 1 to 16)**

The horizontal position is common to all blocks, and can be set in 256 steps (where 1 step is 4<sub>Tosc</sub>, T<sub>osc</sub> being OSD oscillation cycle) as values "00<sub>16</sub>" to "FF<sub>16</sub>" in bits 0 to 7 of the horizontal position register (address 0204<sub>16</sub>). The horizontal position register is shown in Figure 2.16.11.



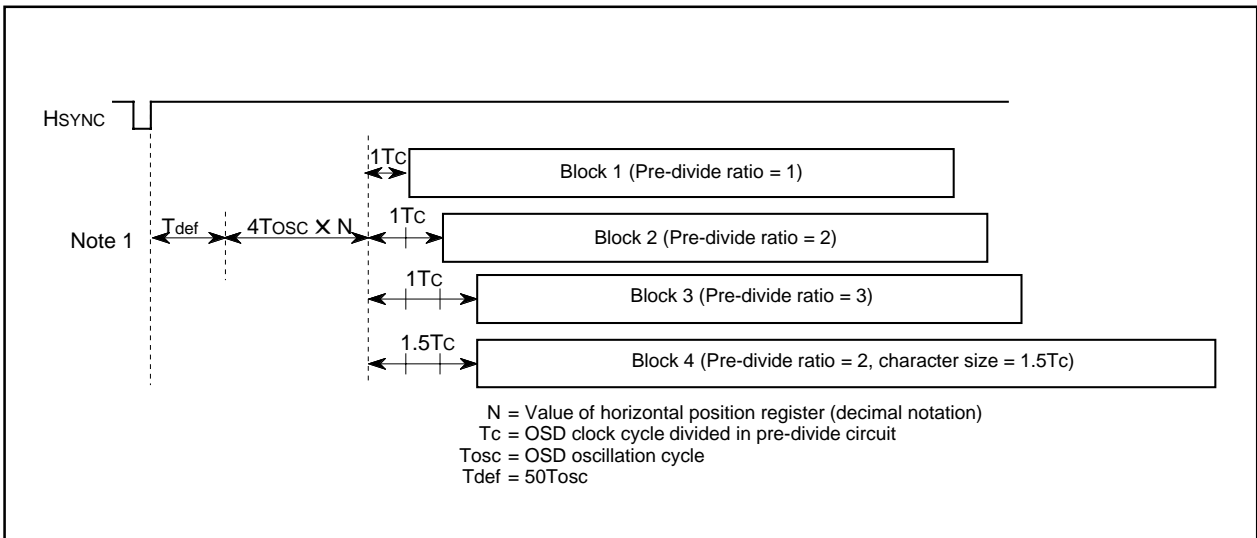
**Figure 2.16.11 Horizontal position register**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

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

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



**Figure 2.16.12 Notes on horizontal display start position**

### 2.16.3 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 (data slicer 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 2.16.4 (the block control register i), refer to Figure 2.16.15 (the clock control register).

The block diagram of dot size control circuit is shown in Figure 2.16.13.

**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 “2.16.17 Scan Mode” about the scan mode.

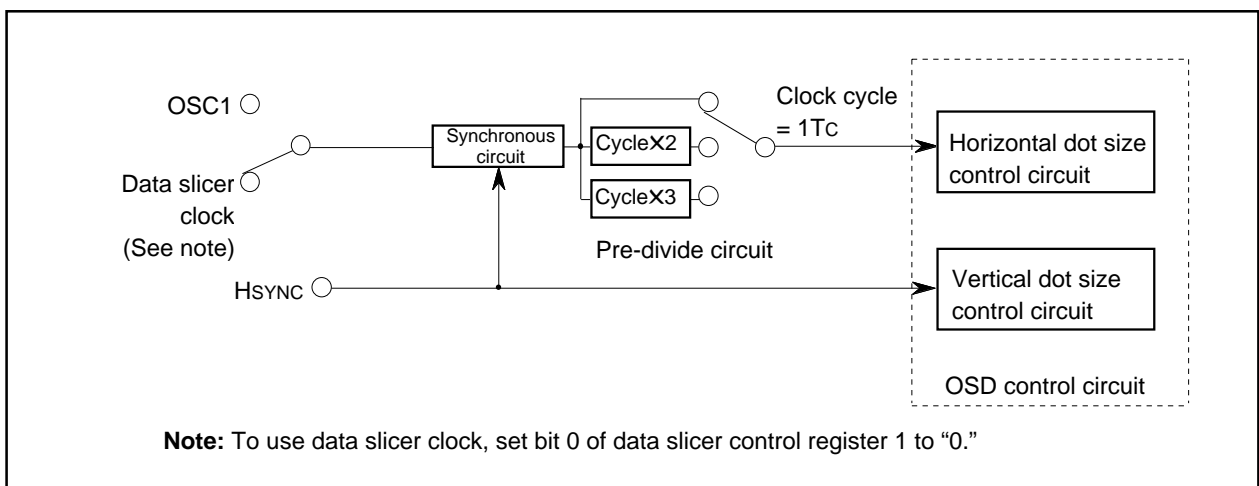


Figure 2.16.13 Block diagram of dot size control circuit

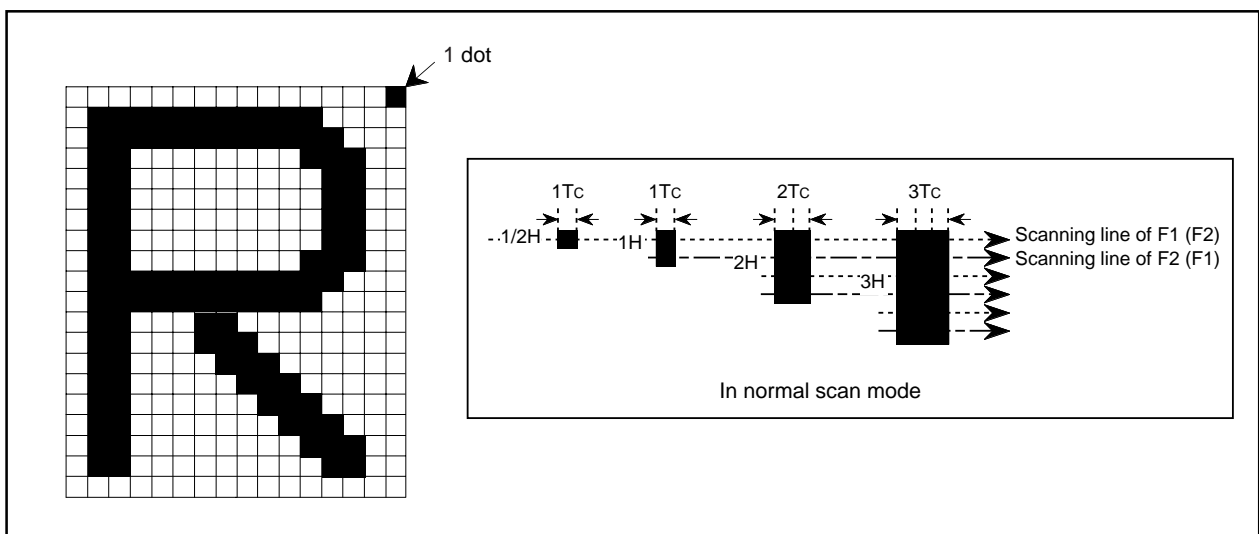


Figure 2.16.14 Definition of dot sizes

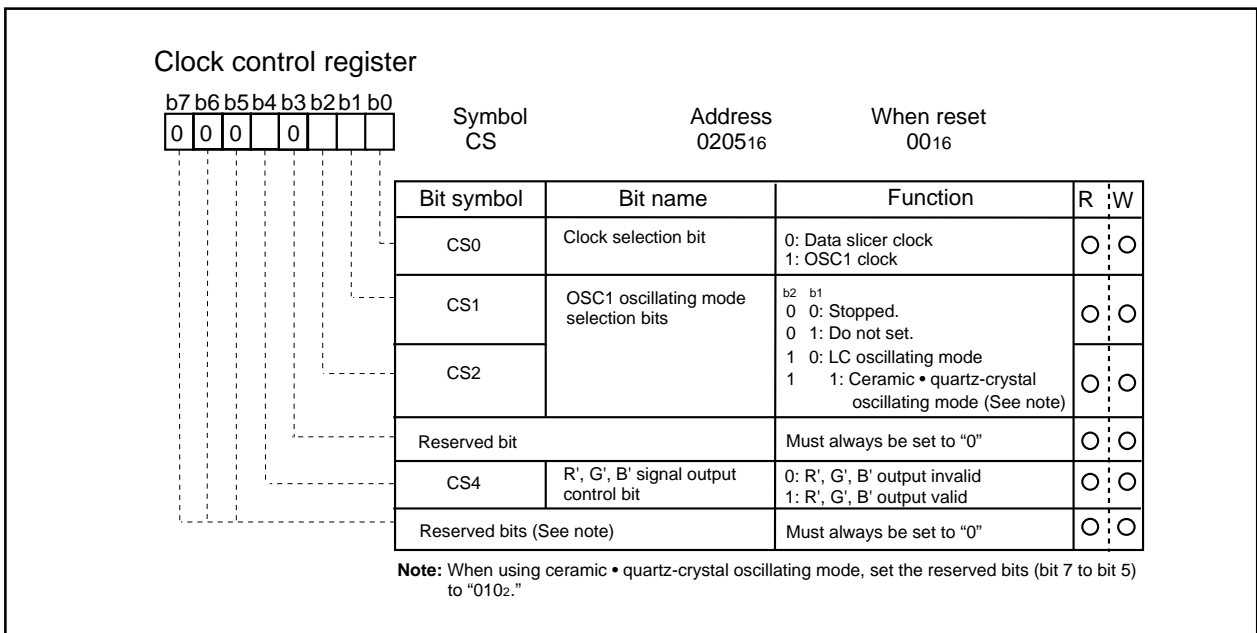
**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

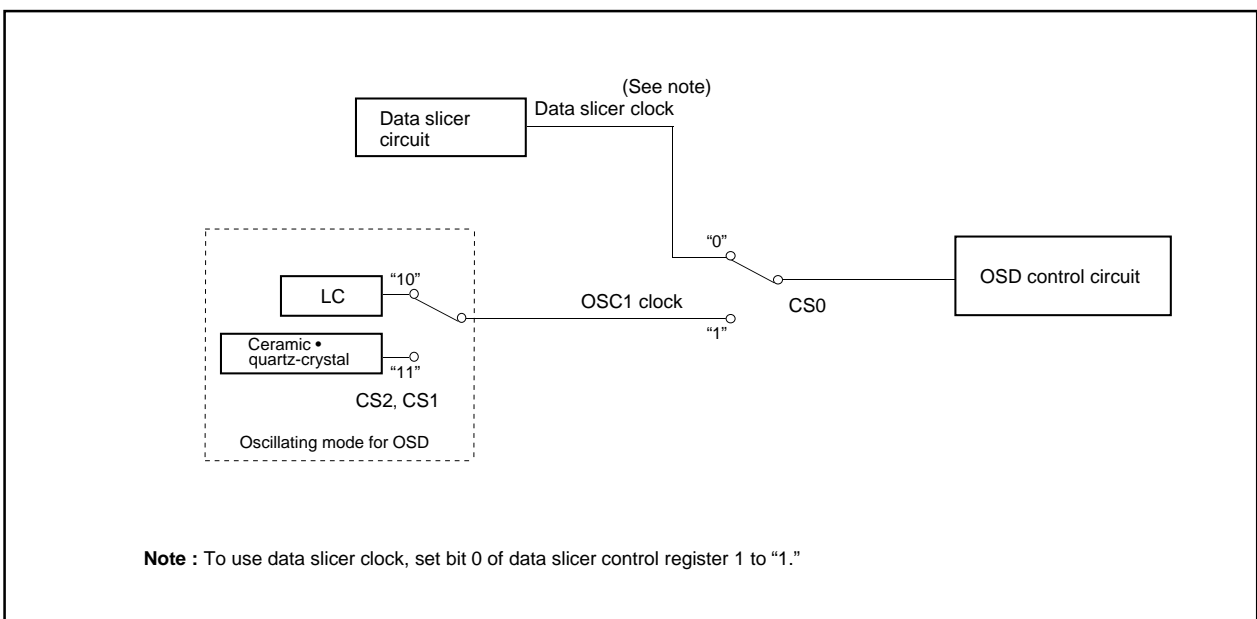
**2.16.4 Clock for OSD**

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

- Data slicer clock output from the data slicer (approximately 26 MHz)
- Clock from the LC oscillator supplied from the pins OSC1 and OSC2
- Clock from the ceramic resonator (or the quartz-crystal oscillator) from the pins OSC1 and OSC2



**Figure 2.16.15 Clock control register**



**Figure 2.16.16 Block Diagram of OSD selection circuit**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**2.16.5 Field Determination Display**

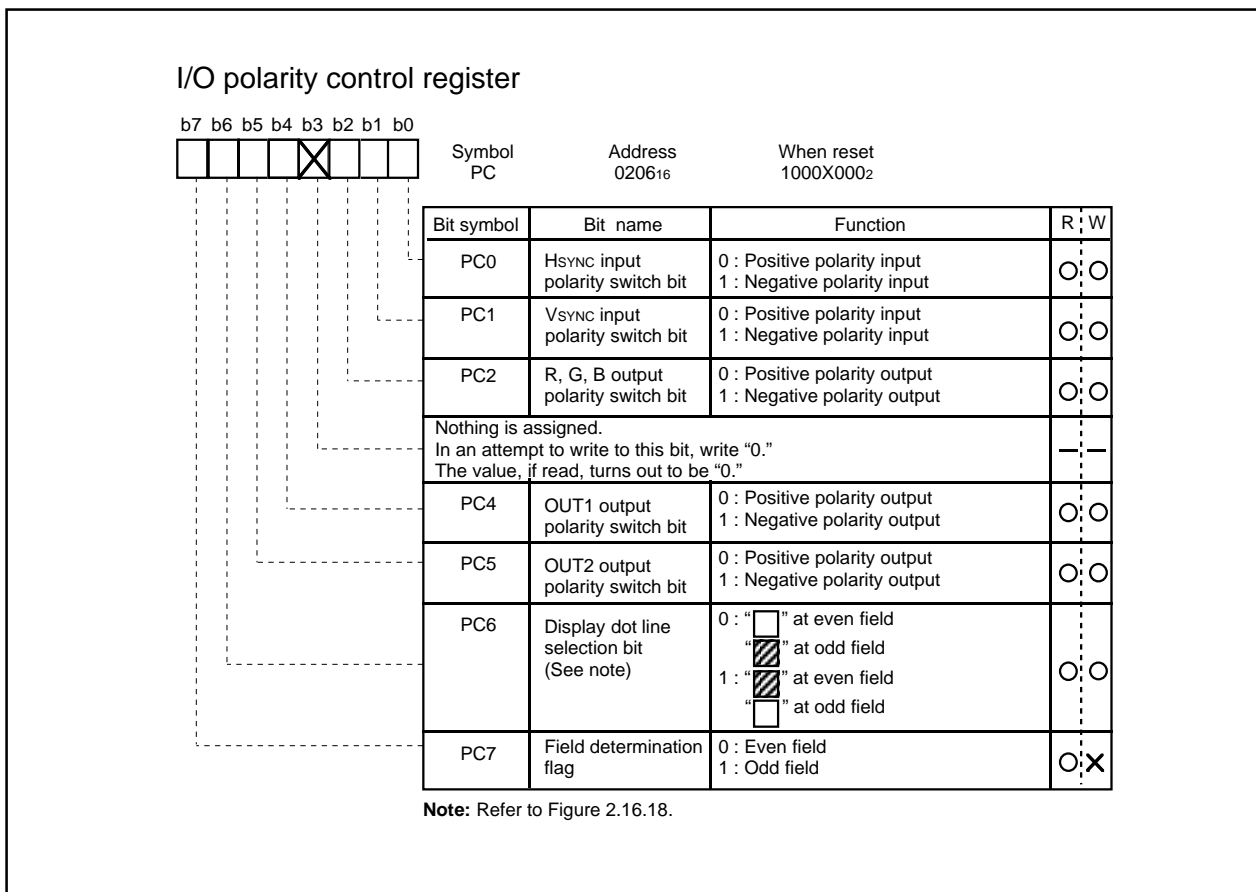
To display the block with vertical dot size of 1/2H, whether an even field or an odd field is determined through differences in a synchronizing signal waveform of interlacing system. The dot line 0 or 1 (refer to Figure 2.16.18) corresponding to the field is displayed alternately.

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 2.16.9) 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<sub>16</sub>). A dot line is specified by bit 6 of the I/O polarity control register (refer to Figure 2.16.18).

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



**Figure 2.16.17 I/O polarity control register**

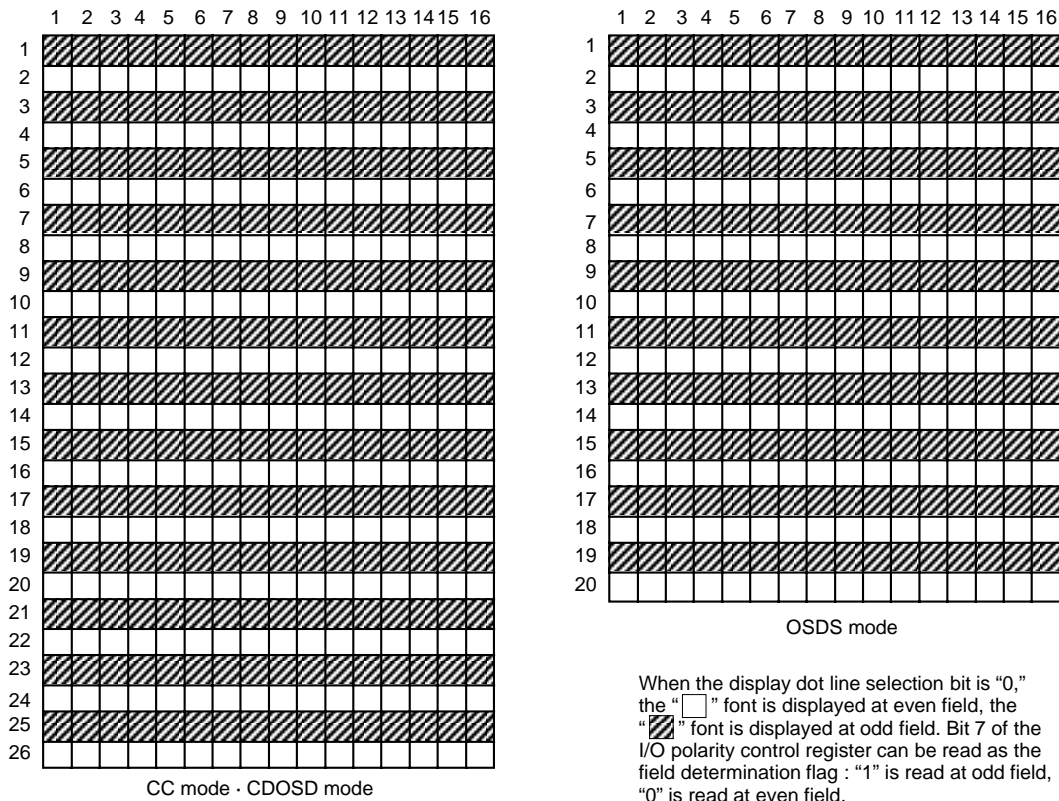
M306V0ME-XXXFP  
M306V0EEFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

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	(n - 1) field (Odd-numbered)	Odd	/	/	/
	Upper : Vsync signal	Even	0 (T2 > T1)	0	Dot line 1 <input type="checkbox"/>
	Lower : Vsync control signal in microcomputer	Odd	1 (T3 < T2)	1	Dot line 0 <input checked="" type="checkbox"/>

When using the field determination flag, set bit 7 of the peripheral mode register (address 027D16) according to the main clock frequency.



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 2.16.18 Relation between field determination flag and display font

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**2.16.6 Memory for OSD**

There are 2 types of memory for OSD : OSD ROM (addresses 90000<sub>16</sub> to AFFFF<sub>16</sub>) used to store character dot data and OSD RAM (addresses 0400<sub>16</sub> to 0FFF<sub>16</sub>) 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 90000<sub>16</sub> to AFFFF<sub>16</sub>)**

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

For character font, there are the following 2 mode.

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

The modes are selected by bit 3 of the OSD control register 3 for each screen.

The character font data storing address for OSDL enable/OSDL disable mode are shown in Figures 2.16.20 and 2.16.21. The conditions for each OSDL enable/disable mode are shown in Figure 2.16.22. The CD font data storing address is shown in Figure 2.16.23.

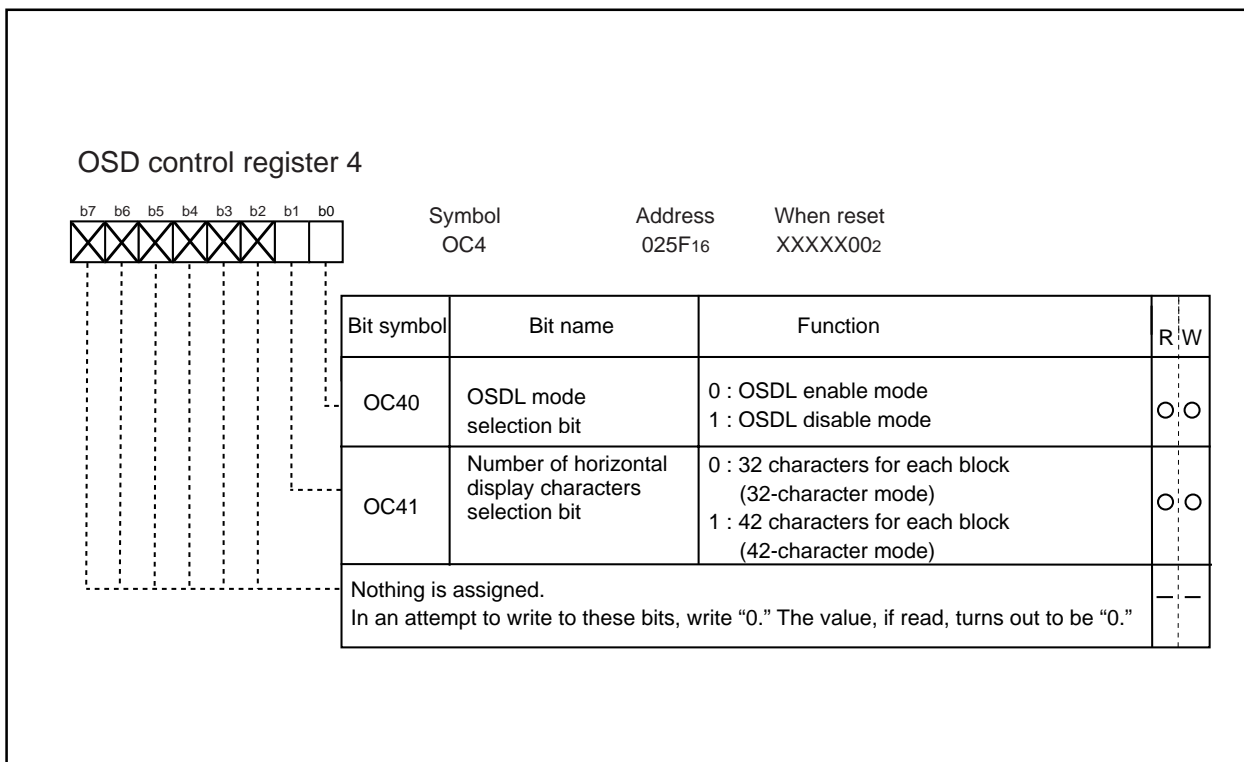


Figure 2.16.19 OSD control register 4



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

OSD ROM address of character font data (OSDL enable mode)

OSD ROM address bit		AD16	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Kinds of font		Structure of address pointer																
	Font (1) Character codes 000 <sub>16</sub> to 0FF <sub>16</sub>	0	Line number (1) (MSB to LSB)					Character code (C8)=0	Character code (C7 to C0)							0	Area bit	
Font (2) Character codes 100 <sub>16</sub> to 1FF <sub>16</sub>	Area 0, 1	0	Line number (2) (MSB to LSB)					Character code (C8)=1	Character code (C7 to C0)							0	Area bit	
	Area 2	1	Line number (2) (MSB to LSB)					0	0	Character code (C6 to C0)							0	Character code (C7)

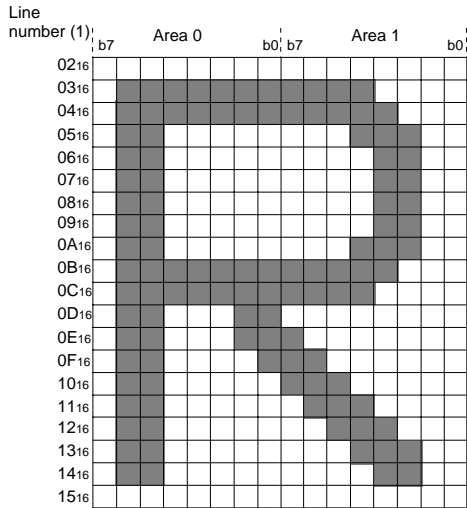
Line number (1) = "02<sub>16</sub>" to "15<sub>16</sub>"

Line number (2) = "00<sub>16</sub>" to "1F<sub>16</sub>"

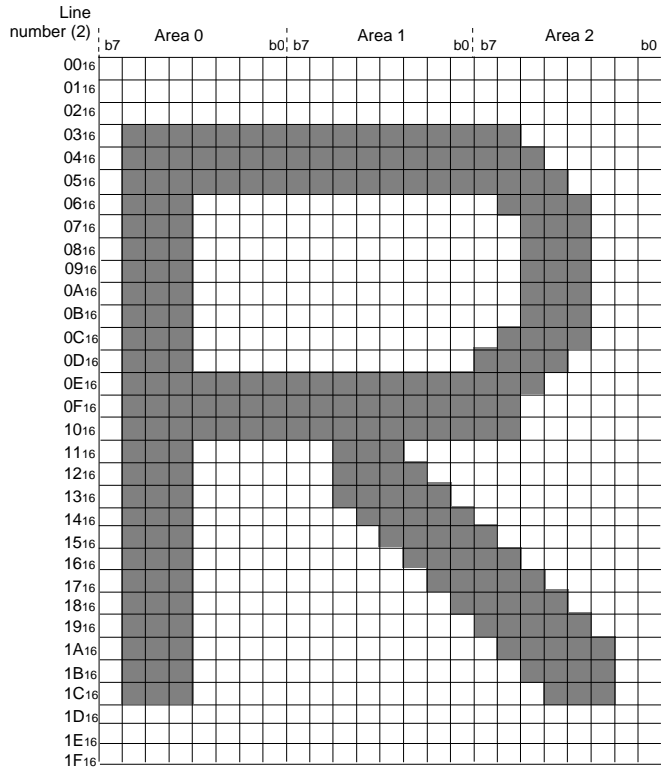
Character code = "000<sub>16</sub>" to "1FF<sub>16</sub>" ("0FE<sub>16</sub>," "0FF<sub>16</sub>," "100<sub>16</sub>" and "180<sub>16</sub>" cannot be used. Write "FF<sub>16</sub>" to corresponding addresses.)

Area bit = 0: Area 0

1: Area 1



Font (1)  
(Character codes 000<sub>16</sub> to 0FF<sub>16</sub>)



Font (2)  
(Character codes 100<sub>16</sub> to 1FF<sub>16</sub>)

Figure 2.16.20 Character font data storing address (OSDL enable mode)

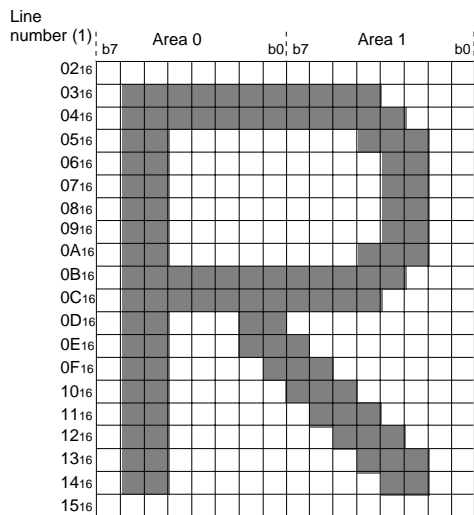
**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

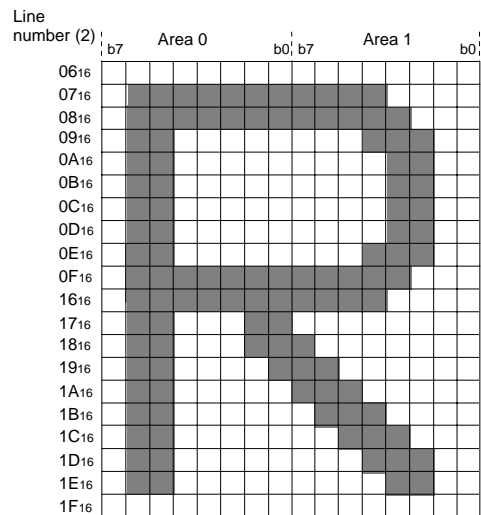
OSD ROM address of character font data (OSDL disable mode)

OSD ROM address bit		AD16	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Structure of address pointer																
Kinds of font	Font (1) Character codes 000 <sub>16</sub> to 1FF <sub>16</sub>	Character code (C9)=0	Line number (1) (MSB to LSB)						Character code (C8 to C0)						0	Area bit		
	Font (2) Character codes 200 <sub>16</sub> to 27F <sub>16</sub>	Character code (C9)=1	Line number (1) (MSB to LSB)						Character code (C8 to C0)						0	Area bit		
	Font (3) Character codes 280 <sub>16</sub> to 2FF <sub>16</sub>	0	1	Line number (3) (NL3 to NL0)				1	Line number (3) (NL4)	Character code (C6 to C0)						0	Area bit	

Line number (1) = "02<sub>16</sub>" to "15<sub>16</sub>"  
 Line number (3) = "06<sub>16</sub>" to "0F<sub>16</sub>" and "16<sub>16</sub>" to "1F<sub>16</sub>"  
 Character code = "000<sub>16</sub>" to "2FF<sub>16</sub>" ("0FE<sub>16</sub>," "0FF<sub>16</sub>," "100<sub>16</sub>," "180<sub>16</sub>," "200<sub>16</sub>" and "280<sub>16</sub>" cannot be used. Write "FF<sub>16</sub>" to corresponding addresses.)  
 Area bit = 0: Area 0  
 1: Area 1



Font (1)  
Font (2)  
(Character codes 000<sub>16</sub> to 27F<sub>16</sub>)



Font (3)  
(Character codes 280<sub>16</sub> to 2FF<sub>16</sub>)

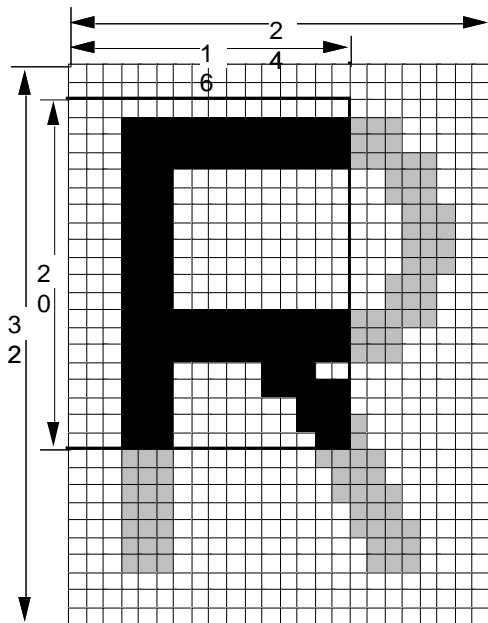
Figure 2.16.21 Character font data storing address (OSDL disable mode)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

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

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



- Notes 1:** Part of 24 X 32 font is displayed.  
**2:** In OSDL disable mode, character codes "280<sub>16</sub>" to "2FF<sub>16</sub>" are used in OSDS mode (no border).  
**3:** As setting this make output of font data indeterminate, do not use.

Figure 2.16.22 Conditions for each OSDL enable/disable mode

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

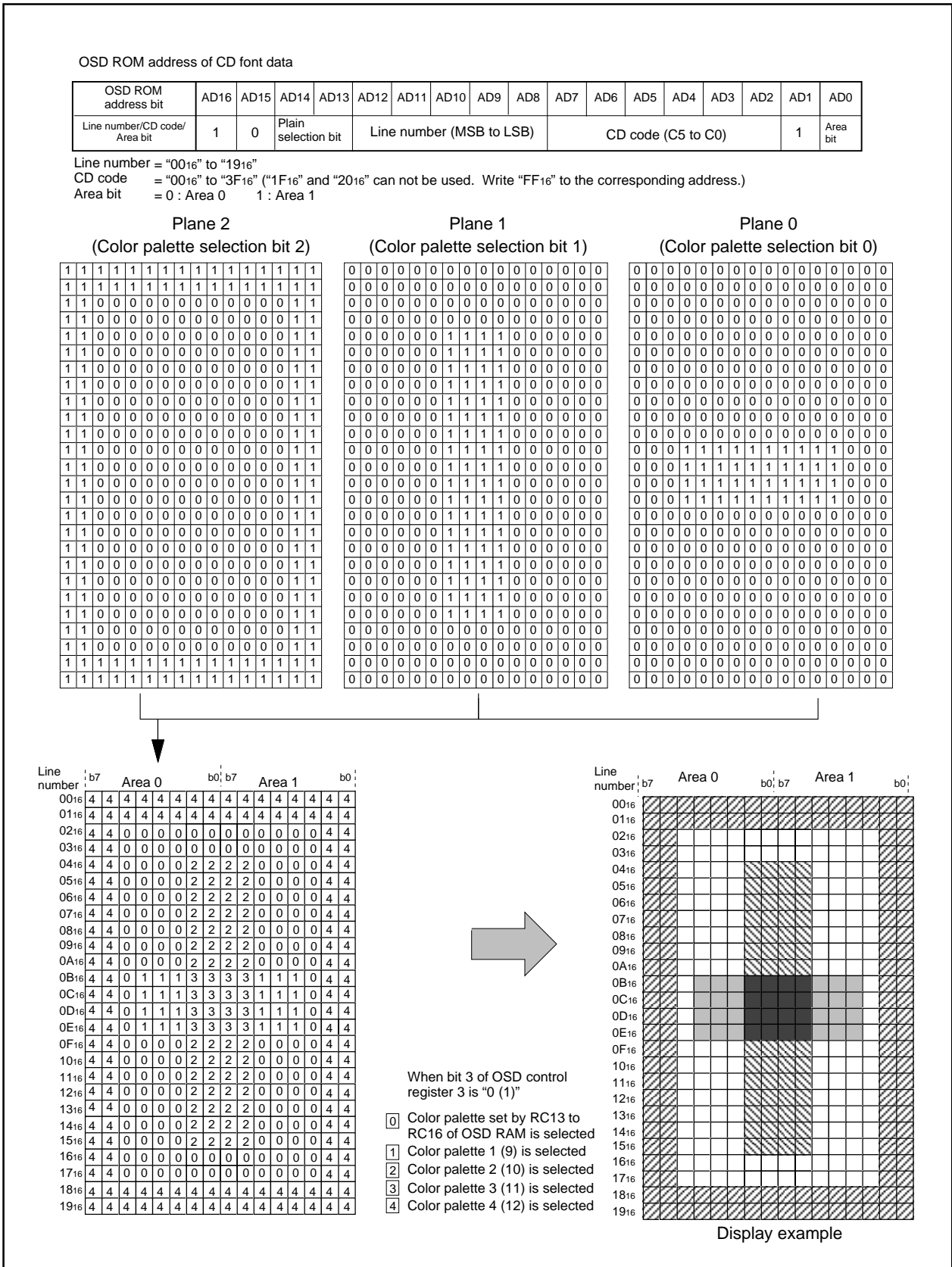


Figure 2.16.23 Color dot font data storing address

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**(2) OSD RAM (OSD RAM for character, addresses 0400<sub>16</sub> to 0EFF<sub>16</sub>)**

The OSD RAM for character is allocated at addresses 0400<sub>16</sub> to 0EFF<sub>16</sub>, 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 2.16.3 to 2.16.7 show the address map.

For example, to display 1 character position (the left edge) in block 1, write the character code in address 0400<sub>16</sub>, write color code 1 at 0401<sub>16</sub>, and write color code 2 at 0480<sub>16</sub>. The structure of the OSD RAM is shown in Figure 2.16.25.

**Notes 1:** Be sure to set a software wait to access to OSD RAM.

**2:** For blocks of the following dot sizes, the 3<sup>rd</sup> (n = 1 to 14) character is skipped as compared with ordinary block.

■ In OSDL mode: all dot size.

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

Accordingly, maximum 22 characters (32-character mode)/28 characters (42-character mode) are only displayed in 1 block. Blocks with dot size of 1Tc X 1/2H and 1Tc X 1H, or blocks on the layer 1. The RAM data for the 3<sup>rd</sup> 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 28<sup>th</sup>'s character area in 42-character mode, set ordinarily.

• In OSDS mode

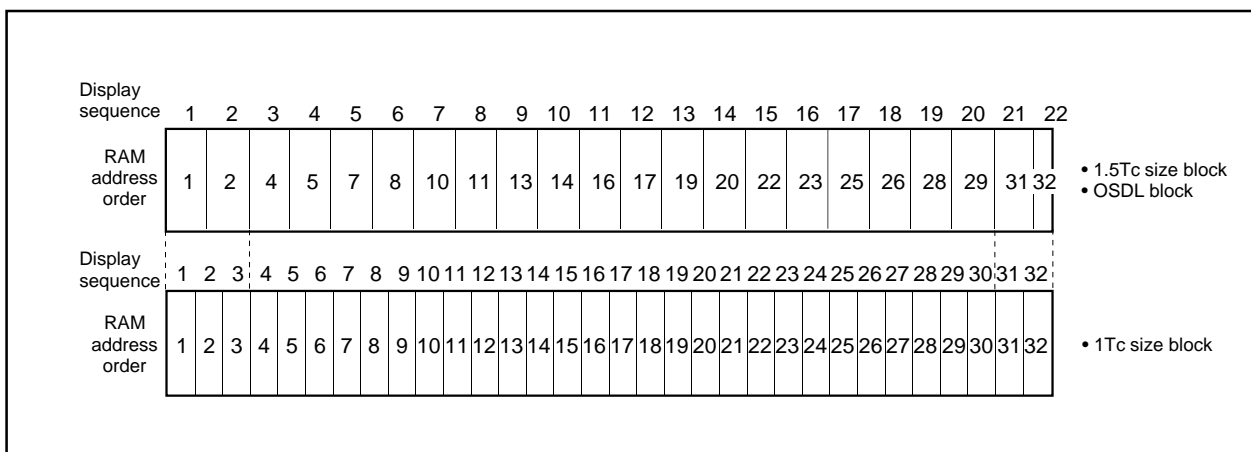
The character is not displayed, and only the left 1/3 part of the 22<sup>nd</sup> character background is displayed in the 22<sup>nd</sup>'s character area. When not displaying this background, set transparent for character background color.

• In OSDL mode

Set a blank character or a character of transparent color to the 22<sup>nd</sup> 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 22<sup>nd</sup>'s character area (left 1/3 part).



**Figure 2.16.24 RAM data for 3<sup>rd</sup> character (in 32-character mode)**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 2.16.3 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	0400 <sub>16</sub>	0401 <sub>16</sub>	0480 <sub>16</sub>
	2nd character	0402 <sub>16</sub>	0403 <sub>16</sub>	0482 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	043E <sub>16</sub>	043F <sub>16</sub>	04BE <sub>16</sub>
Block 2	1st character	0440 <sub>16</sub>	0441 <sub>16</sub>	04C0 <sub>16</sub>
	2nd character	0442 <sub>16</sub>	0443 <sub>16</sub>	04C2 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	047E <sub>16</sub>	047F <sub>16</sub>	04FE <sub>16</sub>
Block 3	1st character	0500 <sub>16</sub>	0501 <sub>16</sub>	0580 <sub>16</sub>
	2nd character	0502 <sub>16</sub>	0503 <sub>16</sub>	0582 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	053E <sub>16</sub>	053F <sub>16</sub>	05BE <sub>16</sub>
Block 4	1st character	0540 <sub>16</sub>	0541 <sub>16</sub>	05C0 <sub>16</sub>
	2nd character	0542 <sub>16</sub>	0543 <sub>16</sub>	05C2 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	057E <sub>16</sub>	057F <sub>16</sub>	05FE <sub>16</sub>
Block 5	1st character	0600 <sub>16</sub>	0601 <sub>16</sub>	0680 <sub>16</sub>
	2nd character	0602 <sub>16</sub>	0603 <sub>16</sub>	0682 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	063E <sub>16</sub>	063F <sub>16</sub>	06BE <sub>16</sub>
Block 6	1st character	0640 <sub>16</sub>	0641 <sub>16</sub>	06C0 <sub>16</sub>
	2nd character	0642 <sub>16</sub>	0643 <sub>16</sub>	06C2 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	067E <sub>16</sub>	067F <sub>16</sub>	06FE <sub>16</sub>
Block 7	1st character	0700 <sub>16</sub>	0701 <sub>16</sub>	0780 <sub>16</sub>
	2nd character	0702 <sub>16</sub>	0703 <sub>16</sub>	0782 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	073E <sub>16</sub>	073F <sub>16</sub>	07BE <sub>16</sub>
Block 8	1st character	0740 <sub>16</sub>	0741 <sub>16</sub>	07C0 <sub>16</sub>
	2nd character	0742 <sub>16</sub>	0743 <sub>16</sub>	07C2 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	077E <sub>16</sub>	077F <sub>16</sub>	07FE <sub>16</sub>
Block 9	1st character	0800 <sub>16</sub>	0801 <sub>16</sub>	0880 <sub>16</sub>
	2nd character	0802 <sub>16</sub>	0803 <sub>16</sub>	0882 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	083E <sub>16</sub>	083F <sub>16</sub>	08BE <sub>16</sub>
Block 10	1st character	0840 <sub>16</sub>	0841 <sub>16</sub>	08C0 <sub>16</sub>
	2nd character	0842 <sub>16</sub>	0843 <sub>16</sub>	08C2 <sub>16</sub>
	⋮	⋮	⋮	⋮
	32nd character	087E <sub>16</sub>	087F <sub>16</sub>	08FE <sub>16</sub>

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 2.16.4 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	0900 <sub>16</sub>	0901 <sub>16</sub>	0980 <sub>16</sub>
	2nd character	0902 <sub>16</sub>	0903 <sub>16</sub>	0982 <sub>16</sub>
	:	:	:	:
	31st character	093C <sub>16</sub>	093D <sub>16</sub>	09BC <sub>16</sub>
	32nd character	093E <sub>16</sub>	093F <sub>16</sub>	09BE <sub>16</sub>
Block 12	1st character	0940 <sub>16</sub>	0941 <sub>16</sub>	09C0 <sub>16</sub>
	2nd character	0942 <sub>16</sub>	0943 <sub>16</sub>	09C2 <sub>16</sub>
	:	:	:	:
	31st character	097C <sub>16</sub>	097D <sub>16</sub>	09FC <sub>16</sub>
	32nd character	097E <sub>16</sub>	097F <sub>16</sub>	09FE <sub>16</sub>
Block 13	1st character	0A00 <sub>16</sub>	0A01 <sub>16</sub>	0A80 <sub>16</sub>
	2nd character	0A02 <sub>16</sub>	0A03 <sub>16</sub>	0A82 <sub>16</sub>
	:	:	:	:
	31st character	0A3C <sub>16</sub>	0A3D <sub>16</sub>	0ABC <sub>16</sub>
	32nd character	0A3E <sub>16</sub>	0A3F <sub>16</sub>	0ABE <sub>16</sub>
Block 14	1st character	0A40 <sub>16</sub>	0A41 <sub>16</sub>	0AC0 <sub>16</sub>
	2nd character	0A42 <sub>16</sub>	0A43 <sub>16</sub>	0AC2 <sub>16</sub>
	:	:	:	:
	31st character	0A7C <sub>16</sub>	0A7D <sub>16</sub>	0AFC <sub>16</sub>
	32nd character	0A7E <sub>16</sub>	0A7F <sub>16</sub>	0AFE <sub>16</sub>
Block 15	1st character	0B00 <sub>16</sub>	0B01 <sub>16</sub>	0B80 <sub>16</sub>
	2nd character	0B02 <sub>16</sub>	0B03 <sub>16</sub>	0B82 <sub>16</sub>
	:	:	:	:
	31st character	0B3C <sub>16</sub>	0B3D <sub>16</sub>	0BBC <sub>16</sub>
	32nd character	0B3E <sub>16</sub>	0B3F <sub>16</sub>	0BBE <sub>16</sub>
Block 16	1st character	0B40 <sub>16</sub>	0B41 <sub>16</sub>	0BC0 <sub>16</sub>
	2nd character	0B42 <sub>16</sub>	0B43 <sub>16</sub>	0BC2 <sub>16</sub>
	:	:	:	:
	31st character	0B7C <sub>16</sub>	0B7D <sub>16</sub>	0BF0 <sub>16</sub>
	32nd character	0B7E <sub>16</sub>	0B7F <sub>16</sub>	0BFE <sub>16</sub>

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 2.16.5 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	0C00 <sub>16</sub>	0C01 <sub>16</sub>	0C80 <sub>16</sub>
	34th character	0C02 <sub>16</sub>	0C03 <sub>16</sub>	0C82 <sub>16</sub>
	:	:	:	:
	39th character	0C0C <sub>16</sub>	0C0D <sub>16</sub>	0C8C <sub>16</sub>
	40th character	0C0E <sub>16</sub>	0C0F <sub>16</sub>	0C8E <sub>16</sub>
	41st character	0E00 <sub>16</sub>	0E01 <sub>16</sub>	0E80 <sub>16</sub>
Block 2	42nd character	0E02 <sub>16</sub>	0E03 <sub>16</sub>	0E82 <sub>16</sub>
	33rd character	0C10 <sub>16</sub>	0C11 <sub>16</sub>	0C90 <sub>16</sub>
	34th character	0C12 <sub>16</sub>	0C13 <sub>16</sub>	0C92 <sub>16</sub>
	:	:	:	:
	39th character	0C1C <sub>16</sub>	0C1D <sub>16</sub>	0C9C <sub>16</sub>
	40th character	0C1E <sub>16</sub>	0C1F <sub>16</sub>	0C9E <sub>16</sub>
Block 3	41st character	0E08 <sub>16</sub>	0E09 <sub>16</sub>	0E88 <sub>16</sub>
	42nd character	0E0A <sub>16</sub>	0E0B <sub>16</sub>	0E8A <sub>16</sub>
	33rd character	0C20 <sub>16</sub>	0C21 <sub>16</sub>	0CA0 <sub>16</sub>
	34th character	0C22 <sub>16</sub>	0C23 <sub>16</sub>	0CA2 <sub>16</sub>
	:	:	:	:
	39th character	0C2C <sub>16</sub>	0C2D <sub>16</sub>	0CAC <sub>16</sub>
Block 4	40th character	0C2E <sub>16</sub>	0C2F <sub>16</sub>	0CAE <sub>16</sub>
	41st character	0E10 <sub>16</sub>	0E11 <sub>16</sub>	0E90 <sub>16</sub>
	42nd character	0E12 <sub>16</sub>	0E13 <sub>16</sub>	0E92 <sub>16</sub>
	33rd character	0C30 <sub>16</sub>	0C31 <sub>16</sub>	0CB0 <sub>16</sub>
	34th character	0C32 <sub>16</sub>	0C33 <sub>16</sub>	0CB2 <sub>16</sub>
	:	:	:	:
Block 5	39th character	0C3C <sub>16</sub>	0C3D <sub>16</sub>	0CBC <sub>16</sub>
	40th character	0C3E <sub>16</sub>	0C3F <sub>16</sub>	0CBE <sub>16</sub>
	41st character	0E18 <sub>16</sub>	0E19 <sub>16</sub>	0E98 <sub>16</sub>
	42nd character	0E1A <sub>16</sub>	0E1B <sub>16</sub>	0E9A <sub>16</sub>
	33rd character	0C40 <sub>16</sub>	0C41 <sub>16</sub>	0CC0 <sub>16</sub>
	34th character	0C42 <sub>16</sub>	0C43 <sub>16</sub>	0CC2 <sub>16</sub>
Block 6	:	:	:	:
	39th character	0C4C <sub>16</sub>	0C4D <sub>16</sub>	0CCC <sub>16</sub>
	40th character	0C4E <sub>16</sub>	0C4F <sub>16</sub>	0CCE <sub>16</sub>
	41st character	0E20 <sub>16</sub>	0E21 <sub>16</sub>	0EA0 <sub>16</sub>
	42nd character	0E22 <sub>16</sub>	0E23 <sub>16</sub>	0EA2 <sub>16</sub>
	33rd character	0C50 <sub>16</sub>	0C51 <sub>16</sub>	0CD0 <sub>16</sub>
Block 7	34th character	0C52 <sub>16</sub>	0C53 <sub>16</sub>	0CD2 <sub>16</sub>
	:	:	:	:
	39th character	0C5C <sub>16</sub>	0C5D <sub>16</sub>	0CDC <sub>16</sub>
	40th character	0C5E <sub>16</sub>	0C5F <sub>16</sub>	0CDE <sub>16</sub>
	41st character	0E28 <sub>16</sub>	0E29 <sub>16</sub>	0EA8 <sub>16</sub>
	42nd character	0E2A <sub>16</sub>	0E2B <sub>16</sub>	0EAA <sub>16</sub>
Block 8	33rd character	0C60 <sub>16</sub>	0C61 <sub>16</sub>	0CE0 <sub>16</sub>
	34th character	0C62 <sub>16</sub>	0C63 <sub>16</sub>	0CE2 <sub>16</sub>
	:	:	:	:
	39th character	0C6C <sub>16</sub>	0C6D <sub>16</sub>	0CEC <sub>16</sub>
	40th character	0C6E <sub>16</sub>	0C6F <sub>16</sub>	0CEE <sub>16</sub>
	41st character	0E30 <sub>16</sub>	0E31 <sub>16</sub>	0EB0 <sub>16</sub>
Block 9	42nd character	0E32 <sub>16</sub>	0E33 <sub>16</sub>	0EB2 <sub>16</sub>



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 2.16.6 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	0C70 <sub>16</sub>	0C71 <sub>16</sub>	0CF0 <sub>16</sub>
	34th character	0C72 <sub>16</sub>	0C73 <sub>16</sub>	0CF2 <sub>16</sub>
	:	:	:	:
	39th character	0C7C <sub>16</sub>	0C7D <sub>16</sub>	0CFC <sub>16</sub>
	40th character	0C7E <sub>16</sub>	0C7F <sub>16</sub>	0CFE <sub>16</sub>
	41st character	0E38 <sub>16</sub>	0E39 <sub>16</sub>	0EB8 <sub>16</sub>
Block 9	42nd character	0E3A <sub>16</sub>	0E3B <sub>16</sub>	0EBA <sub>16</sub>
	33rd character	0D00 <sub>16</sub>	0D01 <sub>16</sub>	0D80 <sub>16</sub>
	34th character	0D02 <sub>16</sub>	0D03 <sub>16</sub>	0D82 <sub>16</sub>
	:	:	:	:
	39th character	0D0C <sub>16</sub>	0D0D <sub>16</sub>	0D8C <sub>16</sub>
	40th character	0D0E <sub>16</sub>	0D0F <sub>16</sub>	0D8E <sub>16</sub>
Block 10	41st character	0E40 <sub>16</sub>	0E41 <sub>16</sub>	0EC0 <sub>16</sub>
	42nd character	0E42 <sub>16</sub>	0E43 <sub>16</sub>	0EC2 <sub>16</sub>
	33rd character	0D10 <sub>16</sub>	0D11 <sub>16</sub>	0D90 <sub>16</sub>
	34th character	0D12 <sub>16</sub>	0D13 <sub>16</sub>	0D92 <sub>16</sub>
	:	:	:	:
	39th character	0D1C <sub>16</sub>	0D1D <sub>16</sub>	0D9C <sub>16</sub>
Block 11	40th character	0D1E <sub>16</sub>	0D1F <sub>16</sub>	0D9E <sub>16</sub>
	41st character	0E48 <sub>16</sub>	0E49 <sub>16</sub>	0EC8 <sub>16</sub>
	42nd character	0E4A <sub>16</sub>	0E4B <sub>16</sub>	0ECA <sub>16</sub>
	33rd character	0D20 <sub>16</sub>	0D21 <sub>16</sub>	0DA0 <sub>16</sub>
	34th character	0D22 <sub>16</sub>	0D23 <sub>16</sub>	0DA2 <sub>16</sub>
	:	:	:	:
Block 12	39th character	0D2C <sub>16</sub>	0D2D <sub>16</sub>	0DAC <sub>16</sub>
	40th character	0D2E <sub>16</sub>	0D2F <sub>16</sub>	0DAE <sub>16</sub>
	41st character	0E50 <sub>16</sub>	0E51 <sub>16</sub>	0ED0 <sub>16</sub>
	42nd character	0E52 <sub>16</sub>	0E53 <sub>16</sub>	0ED2 <sub>16</sub>
	33rd character	0D30 <sub>16</sub>	0D31 <sub>16</sub>	0DB0 <sub>16</sub>
	34th character	0D32 <sub>16</sub>	0D33 <sub>16</sub>	0DB2 <sub>16</sub>
Block 13	:	:	:	:
	39th character	0D3C <sub>16</sub>	0D3D <sub>16</sub>	0DBC <sub>16</sub>
	40th character	0D3E <sub>16</sub>	0D3F <sub>16</sub>	0DBE <sub>16</sub>
	41st character	0E58 <sub>16</sub>	0E59 <sub>16</sub>	0ED8 <sub>16</sub>
	42nd character	0E5A <sub>16</sub>	0E5B <sub>16</sub>	0EDA <sub>16</sub>
	33rd character	0D40 <sub>16</sub>	0D41 <sub>16</sub>	0DC0 <sub>16</sub>
Block 14	34th character	0D42 <sub>16</sub>	0D43 <sub>16</sub>	0DC2 <sub>16</sub>
	:	:	:	:
	39th character	0D4C <sub>16</sub>	0D4D <sub>16</sub>	0DCC <sub>16</sub>
	40th character	0D4E <sub>16</sub>	0D4F <sub>16</sub>	0DCE <sub>16</sub>
	41st character	0E60 <sub>16</sub>	0E61 <sub>16</sub>	0EE0 <sub>16</sub>
	42nd character	0E62 <sub>16</sub>	0E63 <sub>16</sub>	0EE2 <sub>16</sub>
Block 14	33rd character	0D50 <sub>16</sub>	0D51 <sub>16</sub>	0DD0 <sub>16</sub>
	34th character	0D52 <sub>16</sub>	0D53 <sub>16</sub>	0DD2 <sub>16</sub>
	:	:	:	:
	39th character	0D5C <sub>16</sub>	0D5D <sub>16</sub>	0DDC <sub>16</sub>
	40th character	0D5E <sub>16</sub>	0D5F <sub>16</sub>	0DDE <sub>16</sub>
	41st character	0E68 <sub>16</sub>	0E69 <sub>16</sub>	0EE8 <sub>16</sub>
42nd character	0E6A <sub>16</sub>	0E6B <sub>16</sub>	0EEA <sub>16</sub>	

**M306V0ME-XXXFP**  
**M306V0EEFP**

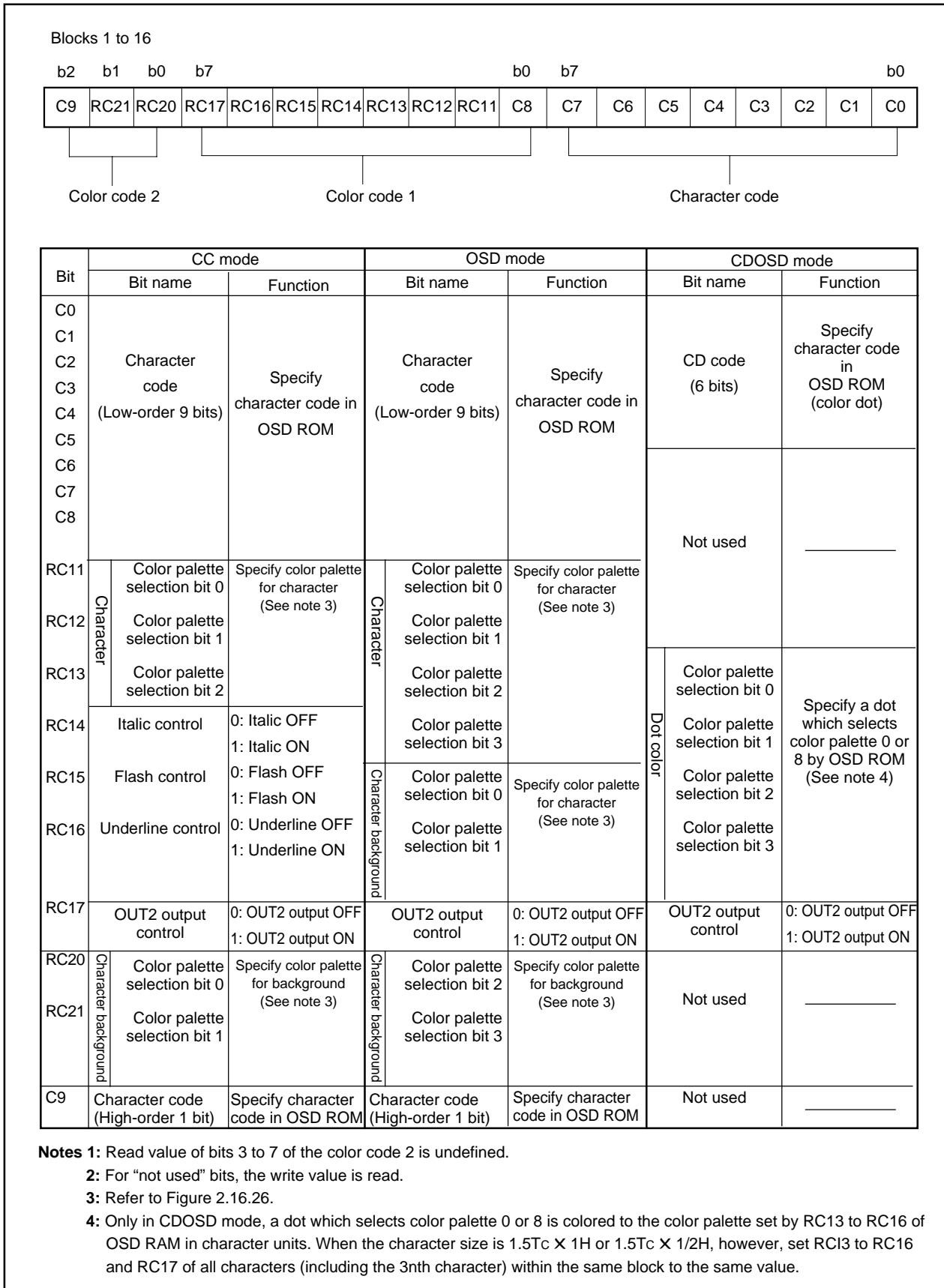
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 2.16.7 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	0D60 <sub>16</sub>	0D61 <sub>16</sub>	0DE0 <sub>16</sub>
	34th character	0D62 <sub>16</sub>	0D63 <sub>16</sub>	0DE2 <sub>16</sub>
	⋮	⋮	⋮	⋮
	39th character	0D6C <sub>16</sub>	0D6D <sub>16</sub>	0DEC <sub>16</sub>
	40th character	0D6E <sub>16</sub>	0D6F <sub>16</sub>	0DEE <sub>16</sub>
	41st character	0E70 <sub>16</sub>	0E71 <sub>16</sub>	0EF0 <sub>16</sub>
Block 16	42nd character	0E72 <sub>16</sub>	0E73 <sub>16</sub>	0EF2 <sub>16</sub>
	33rd character	0D70 <sub>16</sub>	0D71 <sub>16</sub>	0DF0 <sub>16</sub>
	34th character	0D72 <sub>16</sub>	0D73 <sub>16</sub>	0DF2 <sub>16</sub>
	⋮	⋮	⋮	⋮
	39th character	0D7C <sub>16</sub>	0D7D <sub>16</sub>	0DFC <sub>16</sub>
	40th character	0D7E <sub>16</sub>	0D7F <sub>16</sub>	0DFE <sub>16</sub>
	41st character	0E78 <sub>16</sub>	0E79 <sub>16</sub>	0EF8 <sub>16</sub>
	42nd character	0E7A <sub>16</sub>	0E7B <sub>16</sub>	0EFA <sub>16</sub>

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.16.25 Structure of OSD RAM**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

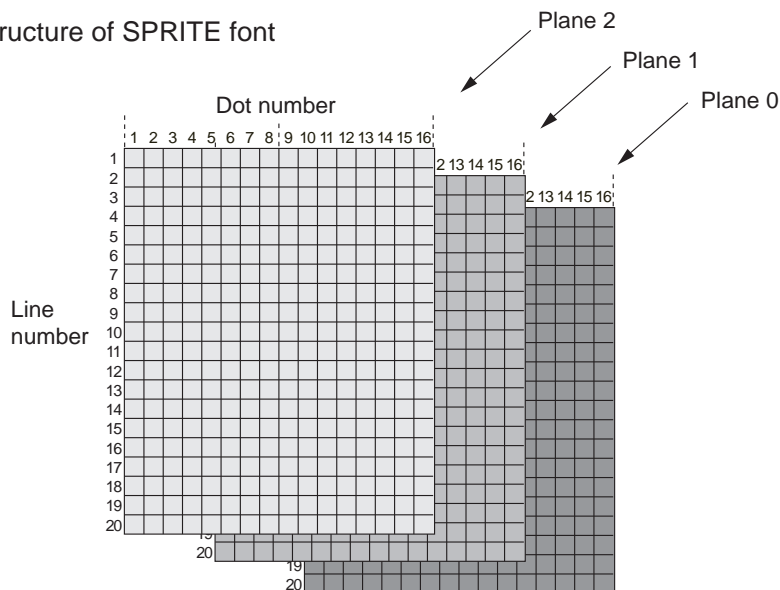
**(3) OSD RAM (OSD RAM for SPRITE, addresses 0F00<sub>16</sub> to 0FA7<sub>16</sub>)**

The OSD RAM for SPRITE fonts 1 and 2, consisting of 3 planes for each font, is assigned to addresses 0F00<sub>16</sub> to 0FA7<sub>16</sub>. Each plane corresponds to each color palette selection bit and the color palette of each dot is determined from among 8 kinds.

**Table 2.16.8 OSD RAM address (SPRITE)**

Planes	Plane 2		Plane 1		Plane 0	
	(Color palette selection bit 2)		(Color palette selection bit 1)		(Color palette selection bit 0)	
Dots	1 to 8	9 to 16	1 to 8	9 to 16	1 to 8	9 to 16
Bits	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0
Line 1	0F80 <sub>16</sub>	0F81 <sub>16</sub>	0F40 <sub>16</sub>	0F41 <sub>16</sub>	0F00 <sub>16</sub>	0F01 <sub>16</sub>
Line 2	0F82 <sub>16</sub>	0F83 <sub>16</sub>	0F42 <sub>16</sub>	0F43 <sub>16</sub>	0F02 <sub>16</sub>	0F03 <sub>16</sub>
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Line 19	0FA4 <sub>16</sub>	0FA5 <sub>16</sub>	0F64 <sub>16</sub>	0F65 <sub>16</sub>	0F24 <sub>16</sub>	0F25 <sub>16</sub>
Line 20	0FA6 <sub>16</sub>	0FA7 <sub>16</sub>	0F66 <sub>16</sub>	0F67 <sub>16</sub>	0F26 <sub>16</sub>	0F27 <sub>16</sub>

Dot structure of SPRITE font



### 2.16.7 Character Color

As shown in Figure 2.16.26, 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 64 colors available. The setting procedure for character colors is as follows:

- CC mode ..... 8 kinds  
Color palette selection range (color palettes 0 to 7 or 8 to 15) can be selected by bit 0 of the OSD control register 3 (address 020716). Color palettes are set by bits RC11 to RC13 of the OSD RAM from among the selection range.
- OSD mode ..... 16 kinds  
Color palettes are set by bits RC11 to RC14 of the OSD RAM.
- CDOSD mode ..... 8 kinds  
Color palette selection range (color palettes 0 to 7 or 8 to 15) can be selected by bit 3 of the OSD control register 3 (address 020716). Color palettes are set in dot units according to CD font data (the OSD RAM <color dot font> contents) from among the selection range.  
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 2.16.28).
- SPRITE display ..... 8 kinds  
Color palette selection range (color palettes 0 to 7 or 8 to 15) can be selected by bit 4 of the OSD control register 3 (address 020716). Color palettes are set in dot units according to the CD font data (the OSD RAM <color dot font> contents) from among the selection range.

**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 2.16.28).

### 2.16.8 Character Background Color

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

- CC mode ..... 4 kinds  
Color palette selection range (color codes 0 to 3, 4 to 7, 8 to 11, or 12 to 15) can be selected by bits 1 and 2 of the OSD control register 3 (address 020716). Color palettes are set by bits RC20 and RC21 of the OSD RAM from among the selection range.
- OSDS mode ..... 16 kinds  
Color palettes are set by bits RC15, RC16, RC20, and RC21 of the OSD RAM.

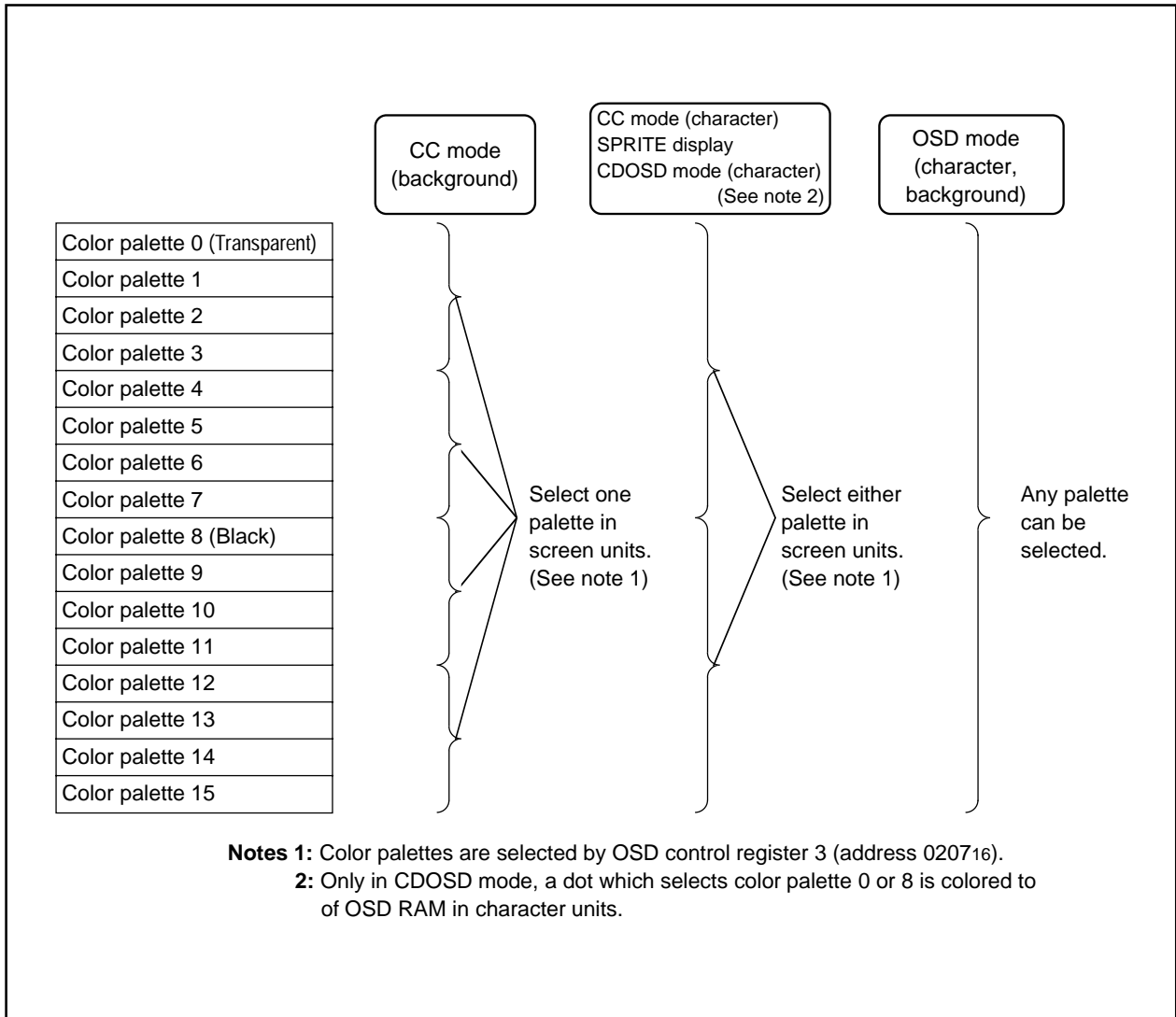
**Note:** The character background is displayed in the following part:

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

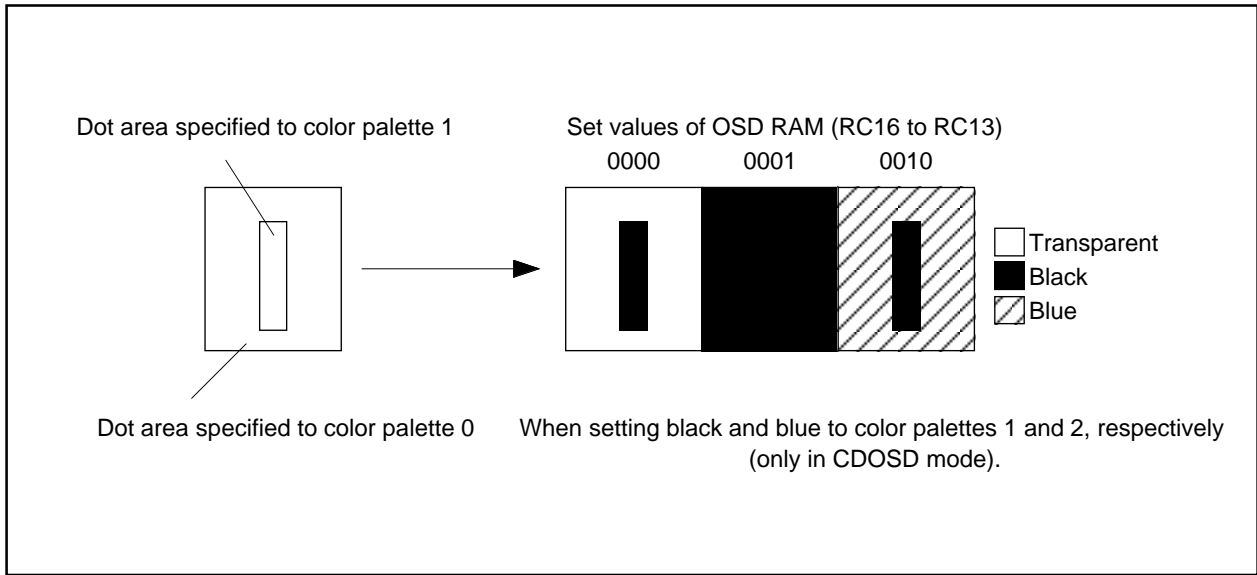
Accordingly, the character background color and the color signal for these two sections cannot be mixed.

**M306V0ME-XXXFP**  
**M306V0EEFP**

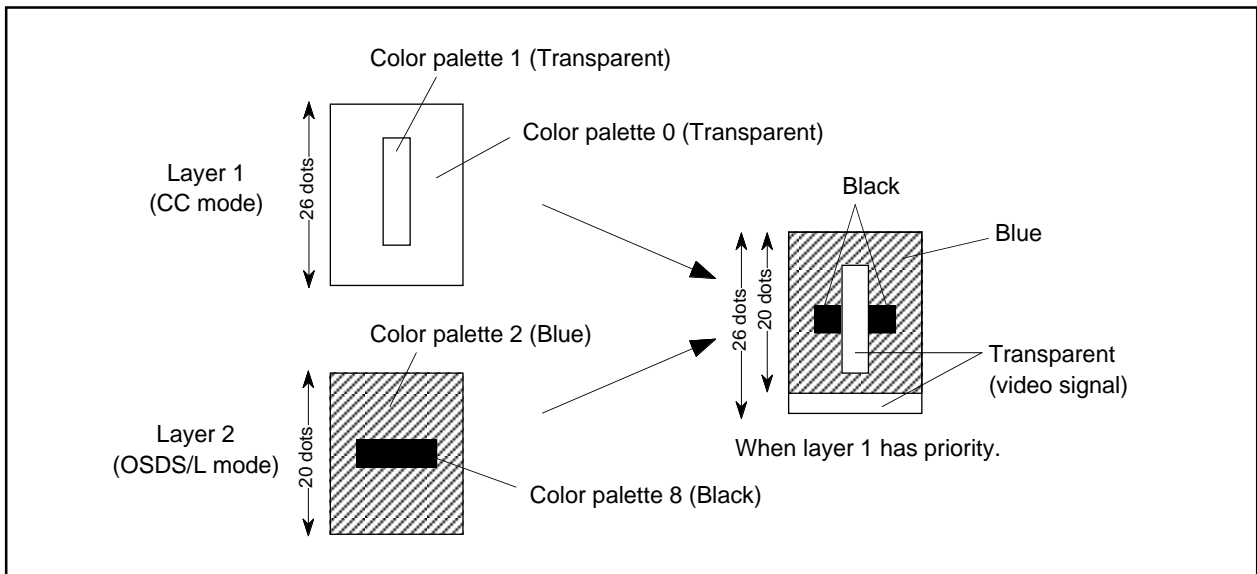
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.16.26 Color palette selection**



**Figure 2.16.27 Set of color palette 0 or 8 in CDOSD mode**



**Figure 2.16.28 Difference between color palette 0 (transparent) and transparent setting of other color palettes**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

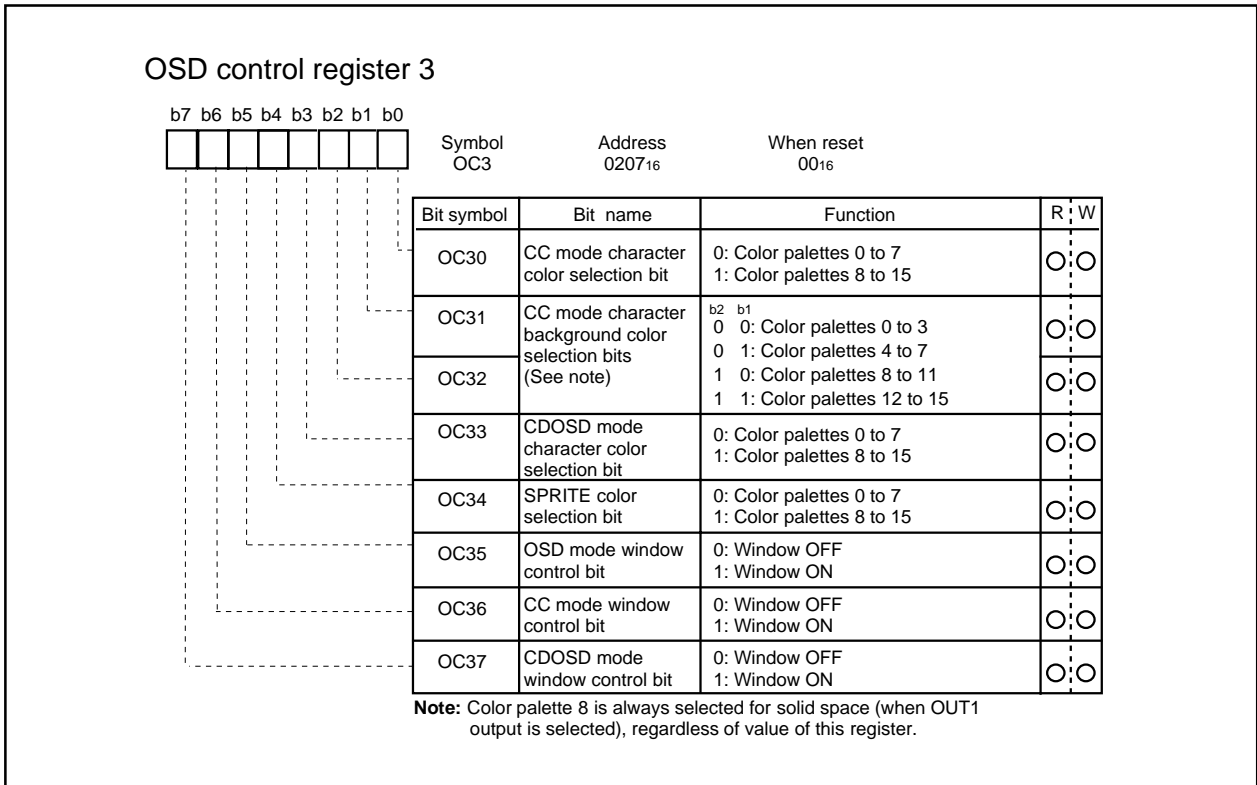


Figure 2.16.29 OSD control register 3

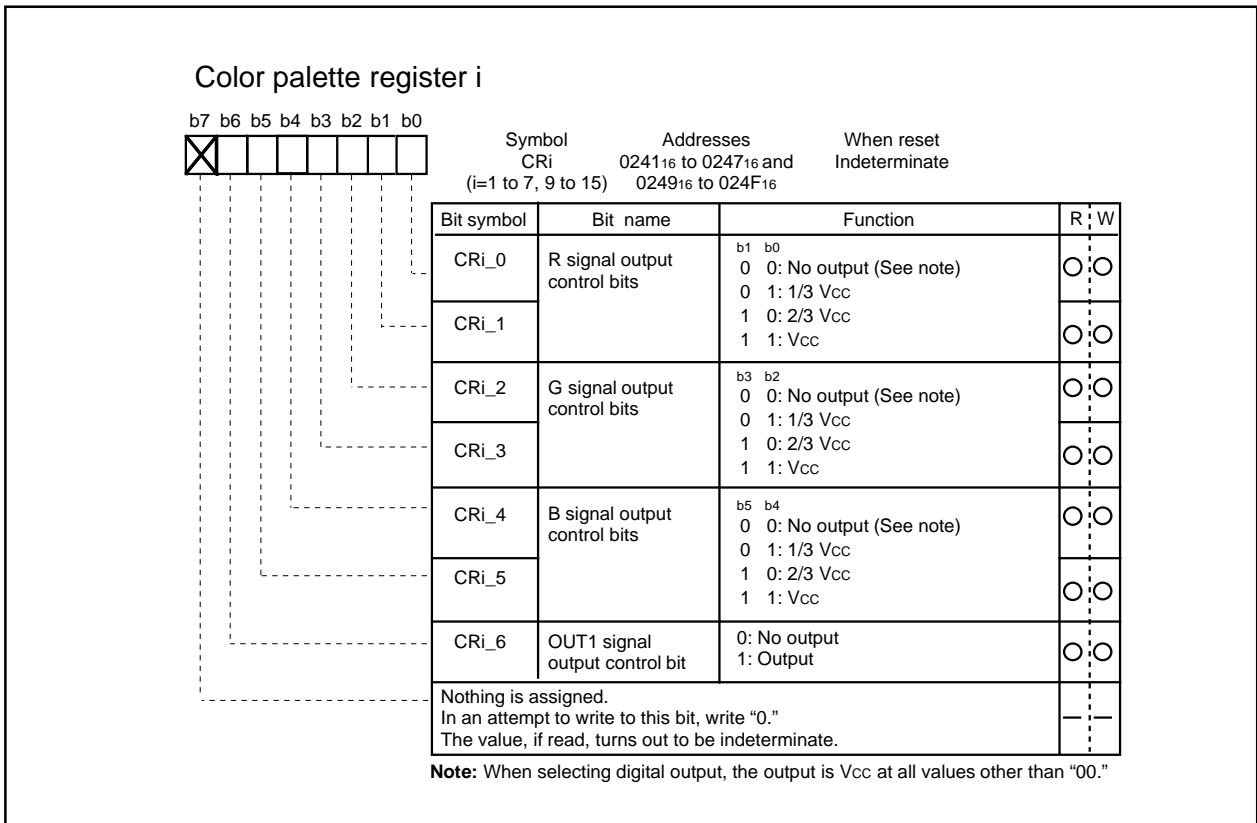


Figure 2.16.30 Color palette register i (i = 1 to 7, 9 to 15)



### 2.16.9 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 2.16.30), bits 0 to 2 of the block control register *i* (refer to Figure 2.16.4) and RC17 of OSD RAM. The setting values for controlling OUT1, OUT2 and the corresponding output waveform is shown in Figure 2.16.31.

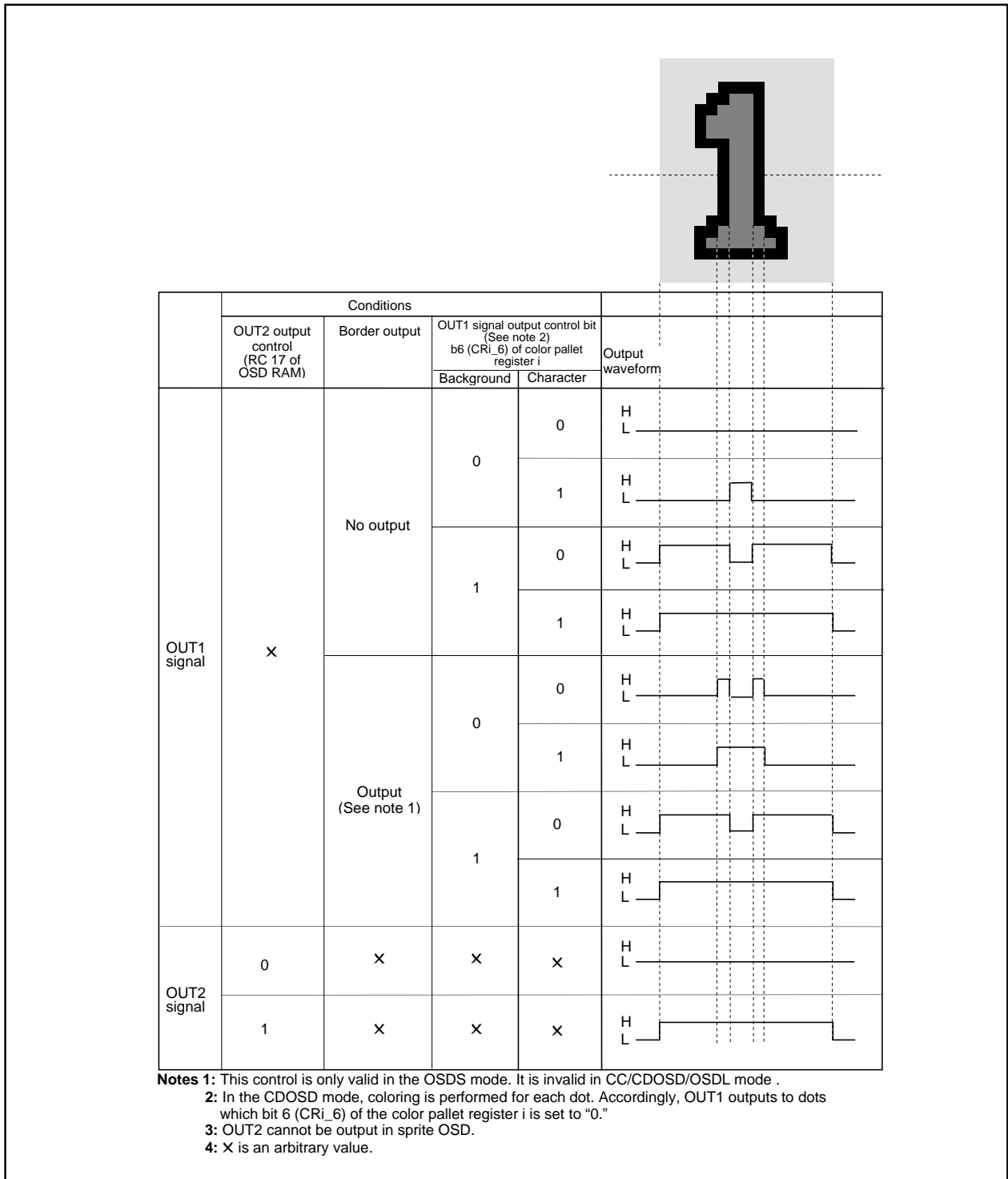


Figure 2.16.31 Setting value for controlling OUT1, OUT2 and corresponding output waveform

## 2.16.10 Attribute

The attributes (flash, underline, italic fonts) are controlled to the character font. The attributes for each character are specified by RC14 to RC16 of OSD RAM (refer to Figure 2.16.26). The attributes to be controlled are different depending on each mode.

CC mode ..... Flash, underline, italic for each character

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

### (1) Under line

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 2.16.3). 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.

<NTSC method> · VSYNC cycle × 24 ≈ 400 ms (at flash ON)

· VSYNC cycle × 8 ≈ 133 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 2.16.33. 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 2.16.32 (c), (d)). However, when displaying the italic character with the pre-divide ratio = 1, set the OSD clock frequency to 11 MHz to 14 MHz.

**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 2.16.33).

**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 2.16.33).

**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 2.16.33).

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

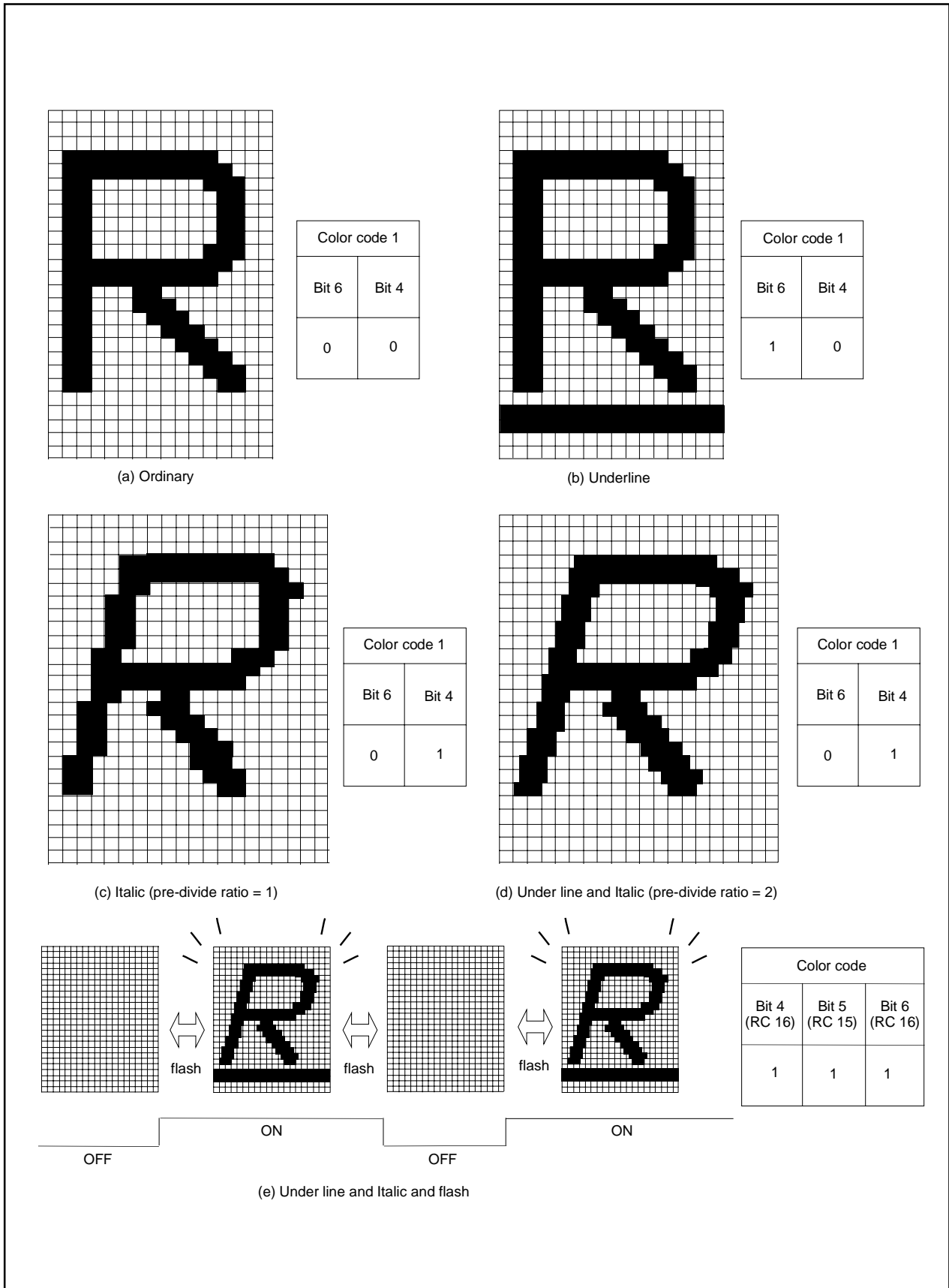
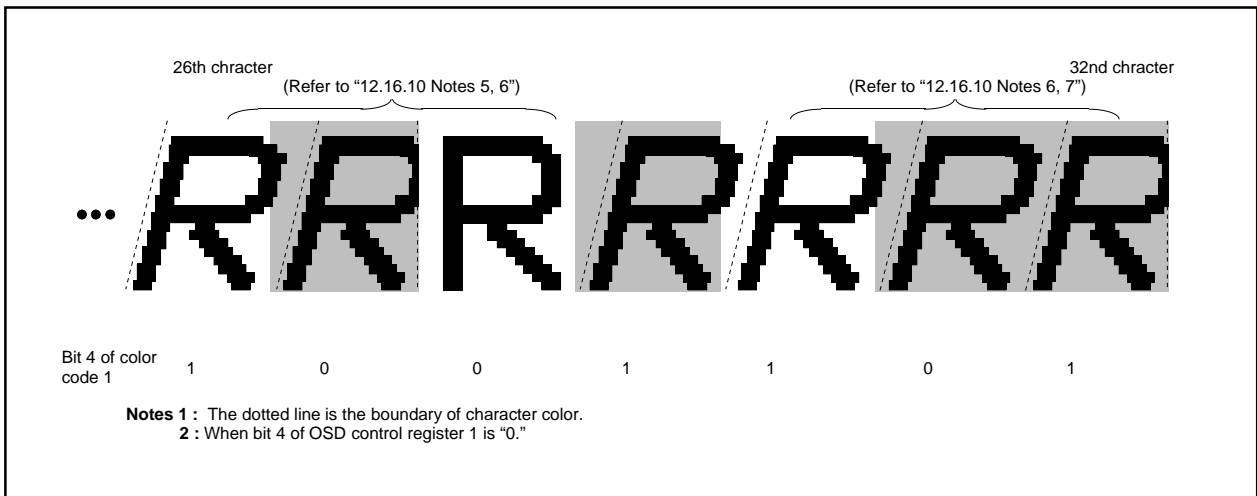


Figure 2.16.32 Example of attribute display (in CC mode)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
 and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.16.33 Example of italic display**

**(4) Border**

The border is output in only the OSDs 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 2.16.34) by bit 2 of the OSD control register 1 (refer to Figure 2.16.3). 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 2.16.4).

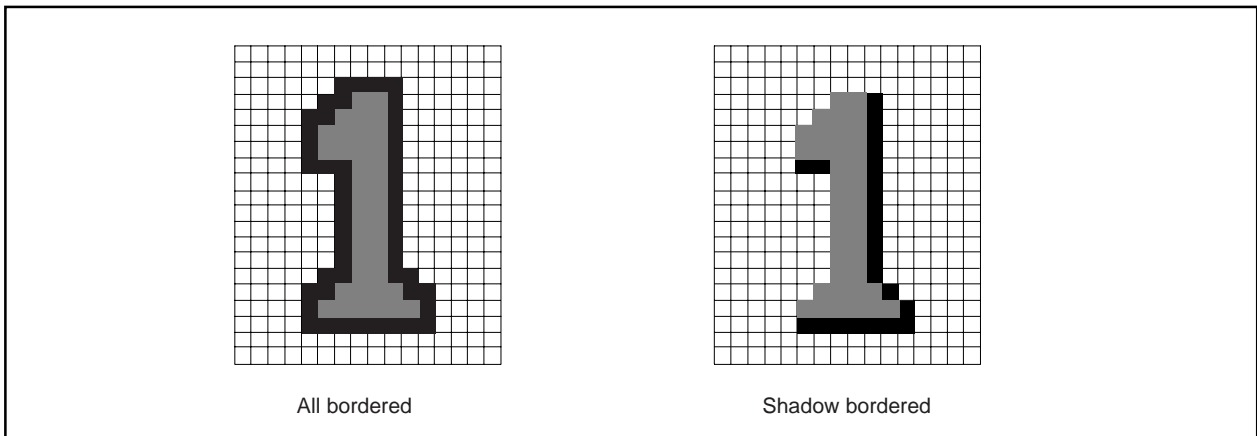
The OUT1 signal is used for border output. The border color is fixed at color palette 8 (black). The border color for each screen is specified by the border color register i.

The horizontal size (x) of border is  $1T_c$  (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 =  $1.5T_c$ , the horizontal size is  $1.5T_c$ . The vertical size (y) different depending on the screen scan mode and the vertical dot size of character font.

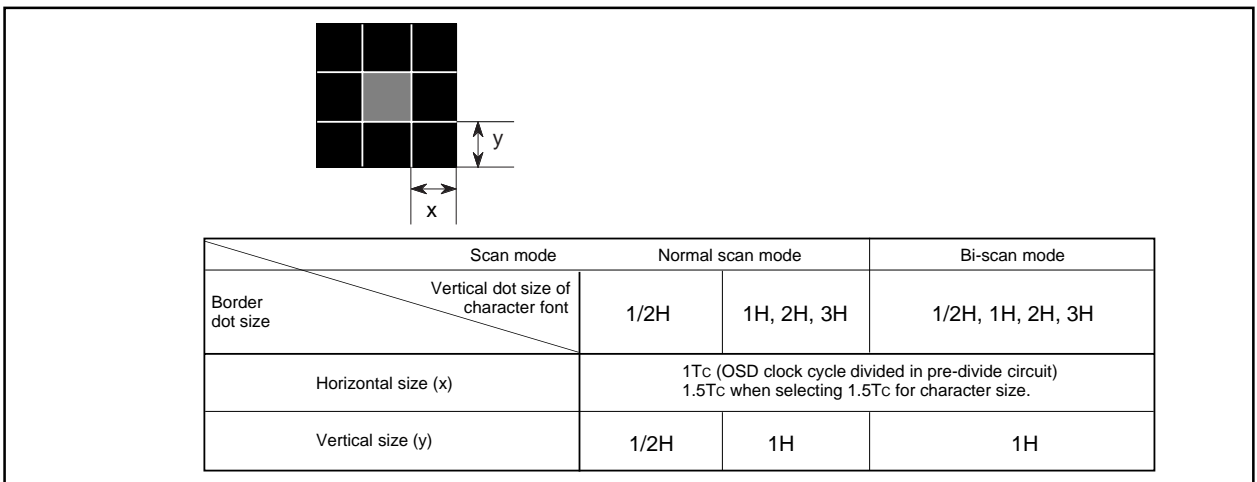
**Notes 1 :** The border dot area is the shaded area as shown in Figure 2.16.36.

- 2 :** When the border dot overlaps on the next character font, the character font has priority (refer to Figure 2.16.37 A). When the border dot overlaps on the next character background, the border has priority (refer to Figure 2.16.37 B).

**3 :** The border in vertical out of character area is not displayed (refer to Figure 2.16.38).



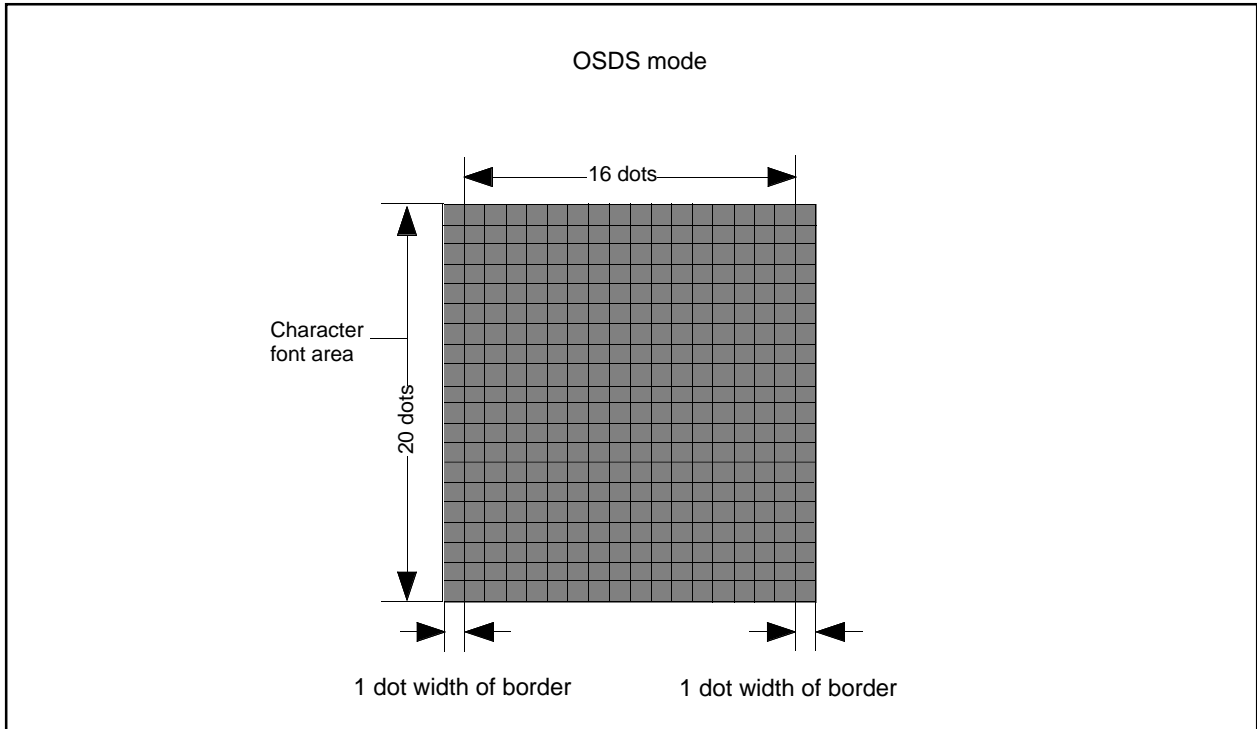
**Figure 2.16.34 Example of border display**



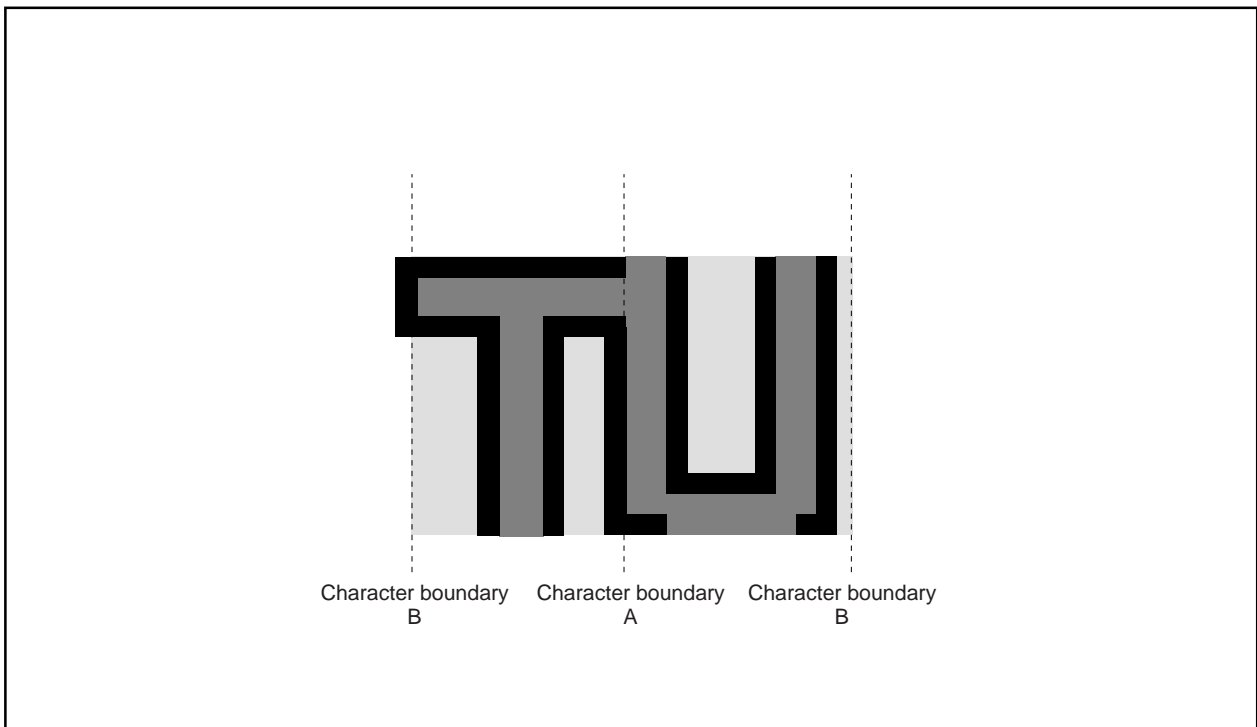
**Figure 2.16.35 Horizontal and vertical size of border**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
 and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.16.36 Border area**



**Figure 2.16.37 Border priority**

**2.16.11 Automatic Solid Space Function**

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

The solid space is output in the following area :

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

This function is turned on and off by bit 4 of the OSD control register 1 (refer to Figure 2.16.3).

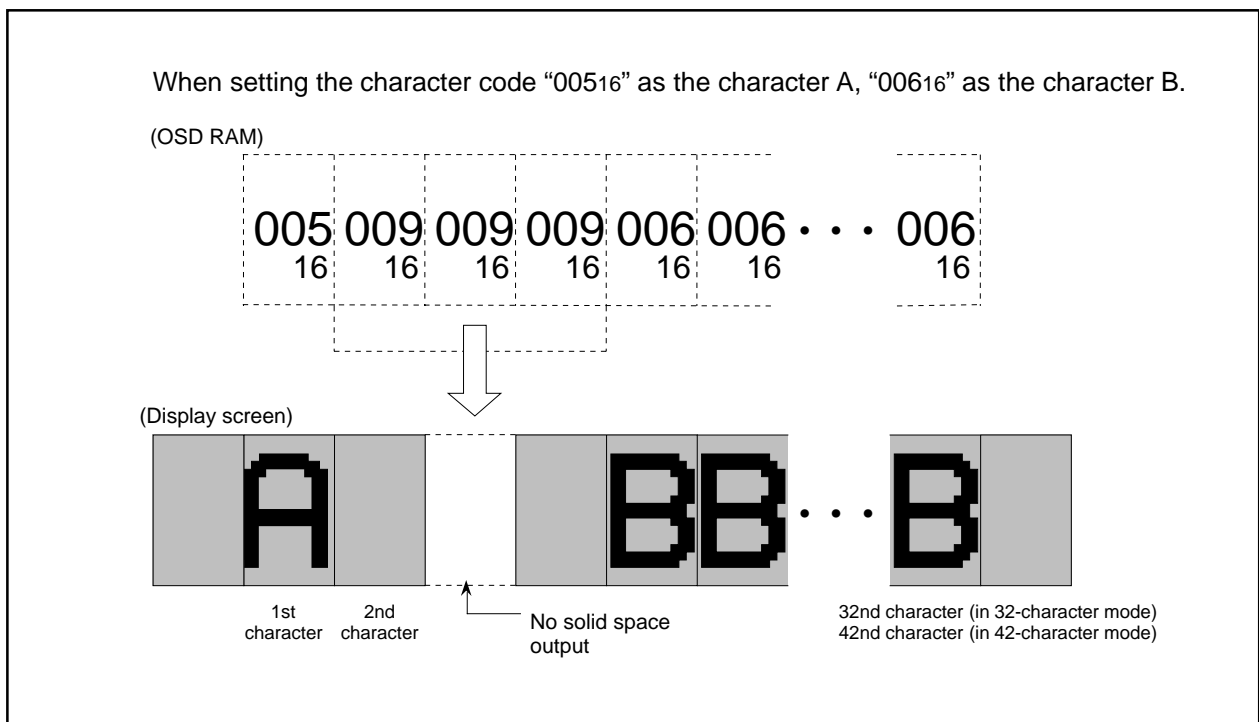
OUT1 or OUT2 output is selected by bit 3 of the OSD control register 2.

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

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

**Table 2.16.10 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 2.16.38 Display screen example of automatic solid space**

**2.16.12 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.

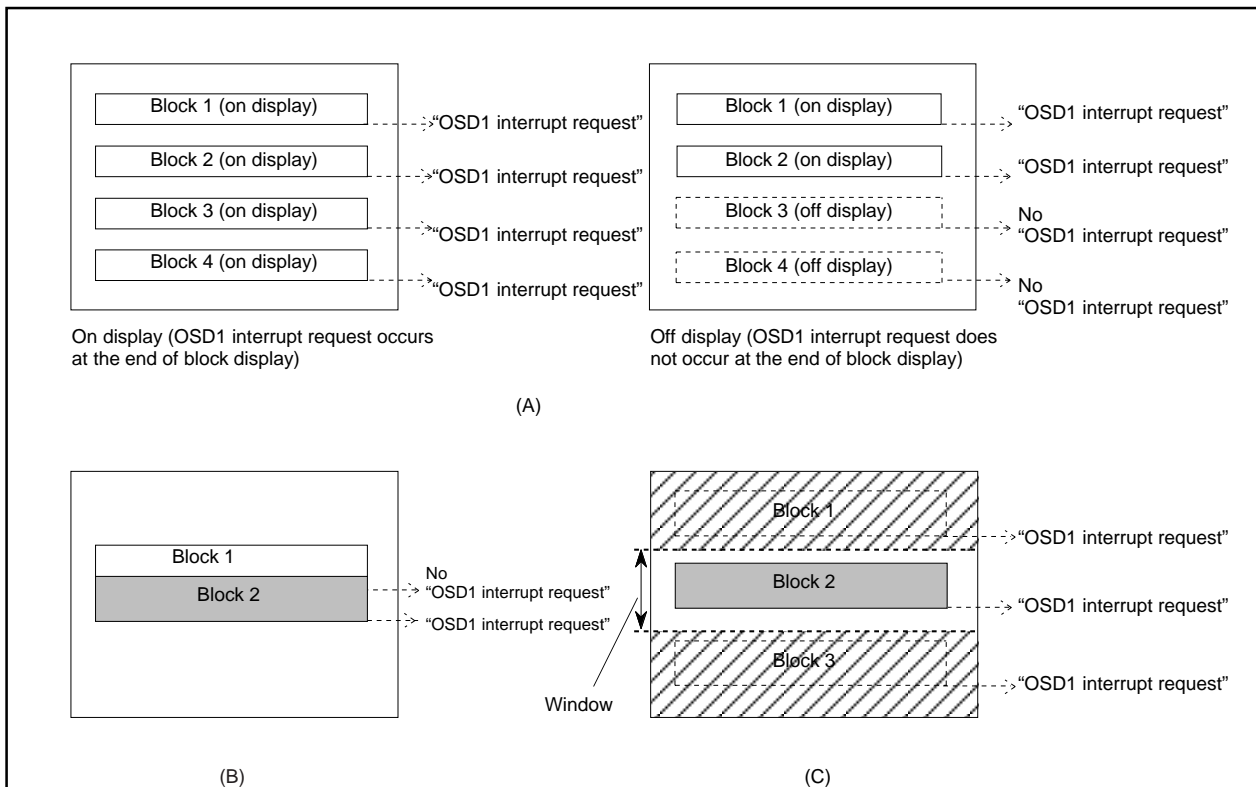
An OSD1 interrupt request occurs at the point at which display of each block has been completed. In other words, when a scanning line reaches the point of the display position (specified by the vertical position registers) of a certain block, the character display of that block starts, and an interrupt 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 2.16.7).

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

**Notes 1:** An OSD1 interrupt does not occur at the end of display when the block is not displayed. In other words, if a block is set to off display by the display control bit of the block control register i (addresses 021016 to 021F16), an OSD1 interrupt request does not occur (refer to Figure 2.16.41 (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 2.16.40 (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 2.16.40 (C)).



**Figure 2.16.40 Note on occurrence of OSD1 interrupt**

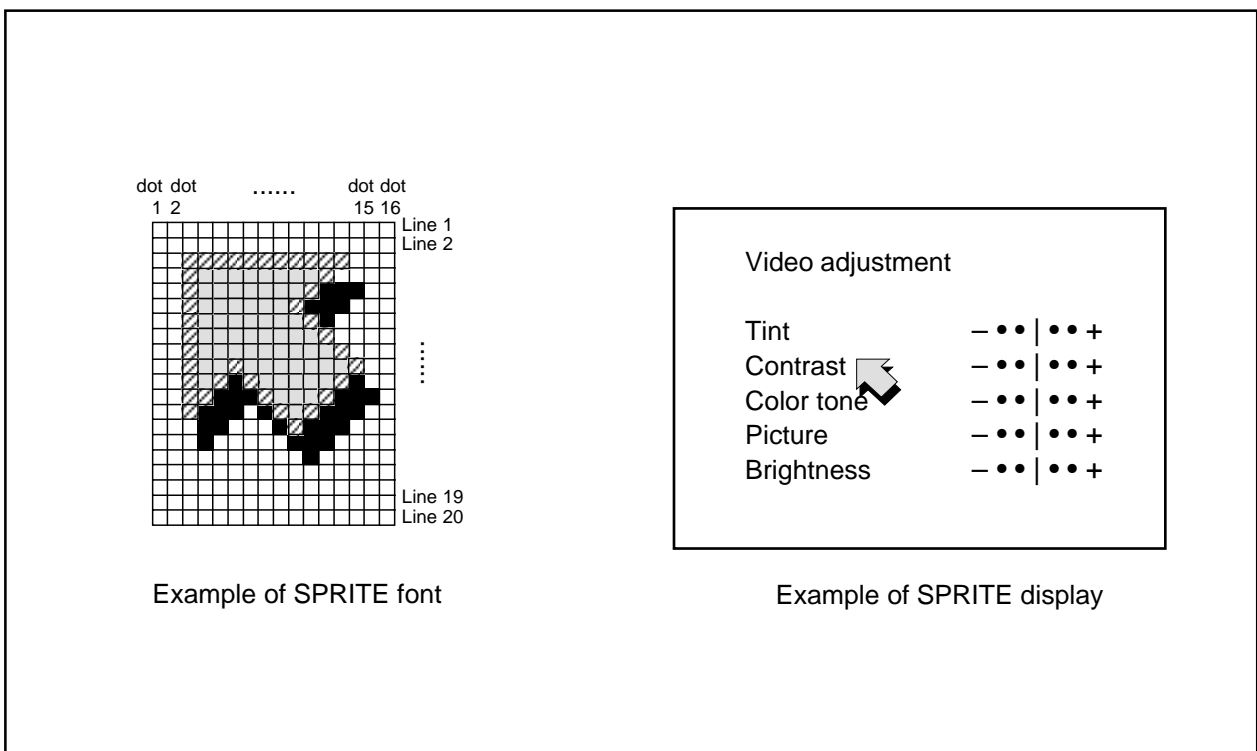


**2.16.13 SPRITE OSD Function**

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. The sprite font is a RAM font consisting of 16 horizontal dots X 20 vertical dots, three planes, and three bits of data per dot. Each plane has corresponding color palette selection bit, and 8 kinds of color palettes can be selected by the plane bit combination (three bits) for each dot. In addition, the selection range (color palettes 0 to 7 and 8 to 15) can be set, per screen, by bit 4 of the OSD control register 3. 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, dot size, interrupt position, and interrupt generation factors for the SPRITE OSD. 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. At this time, the horizontal position is set in 2048 steps in 2TOSC units, and the vertical position is set in 1024 steps in 1TH units. When SPRITE display overlaps with other OSD displays, SPRITE display is always given priority. However, the SPRITE display overlaps with the display which includes OUT2 output, OUT2 in the OSD is output without masking.

- Notes 1:** The SPRITE OSD function cannot output OUT2.  
**2:** When using SPRITE OSD, do not set HS ≤ "00316", HS ≥ "80016."  
**3:** When using SPRITE OSD, do not set VS = "00016," VS ≥ "40016."



**Figure 2.16.41 SPRITE OSD display example**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

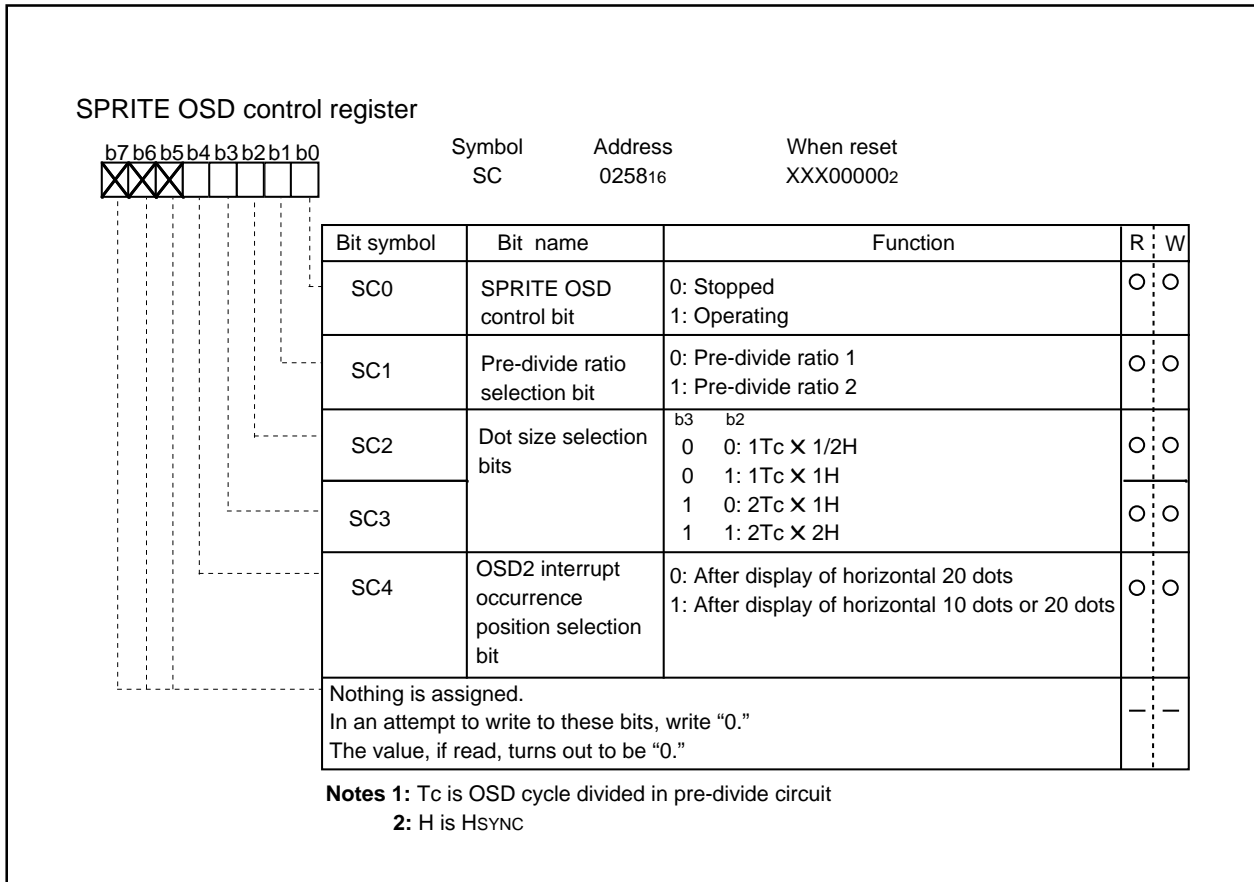
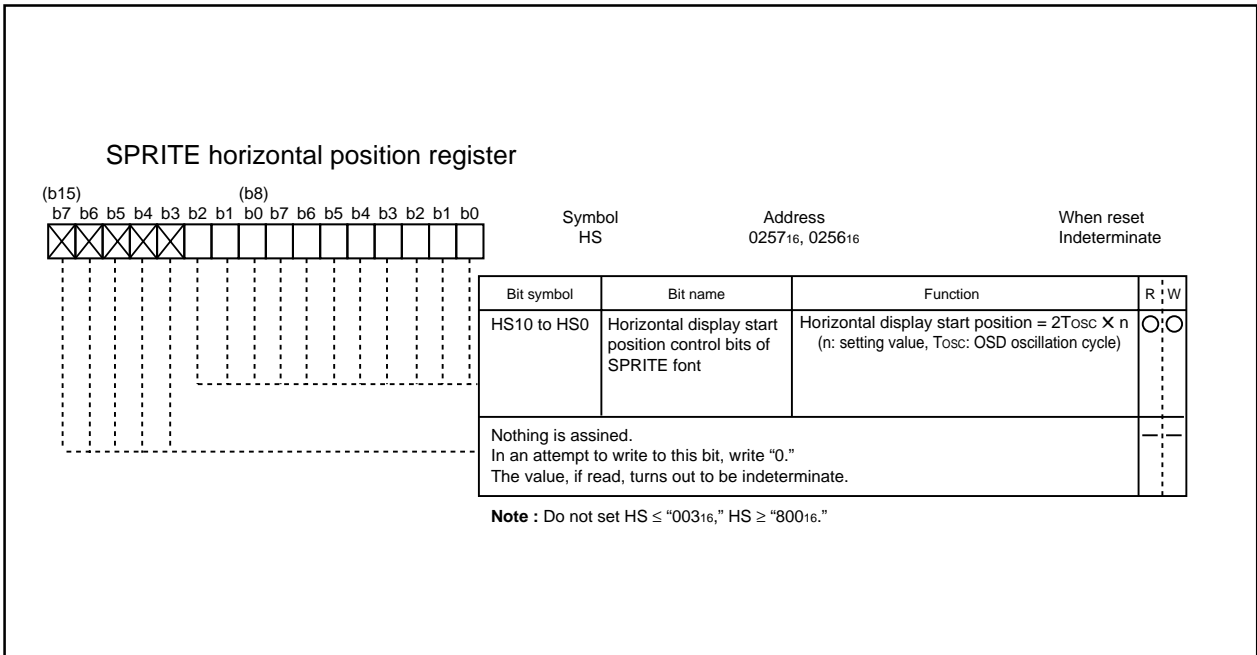


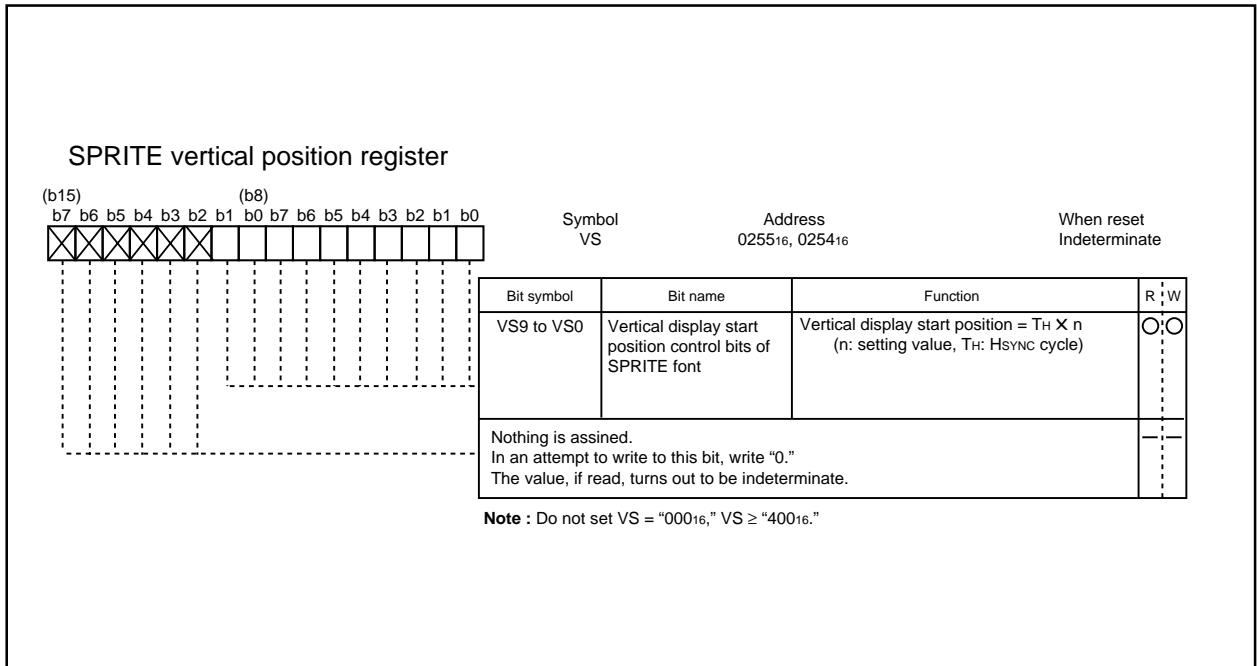
Figure 2.16.42 SPRITE OSD control register

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.16.43** SPRITE horizontal position register



**Figure 2.16.44** SPRITE vertical position register i (i = 1, 2)

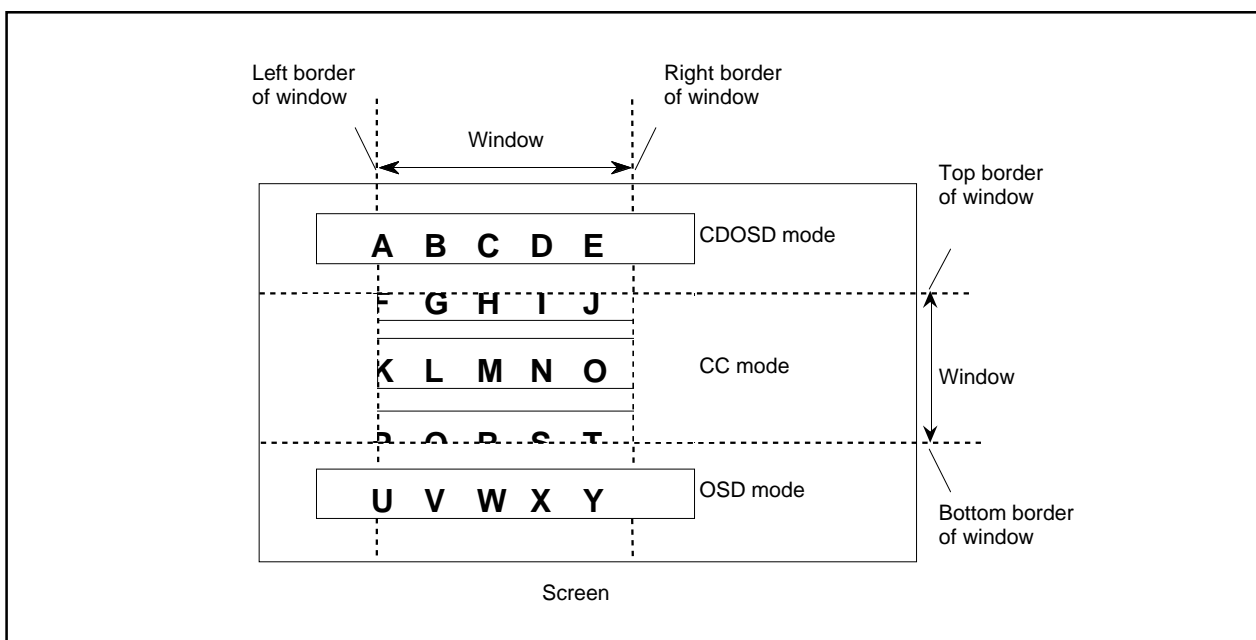
**2.16.14 Window Function**

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 (TB) and the bottom border is set by the bottom border control register (BB).

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 (LB), and the right border is set by the right border control register (RB).

- 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 OSD and CDOSD modes.
- 3:** As for SPRITE display, the window function does not operate.



**Figure 2.16.45 Example of window function (When CC mode is valid)**

**2.16.15 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 (TB), and the bottom border is set by the bottom border control register (BB), in 1H units.

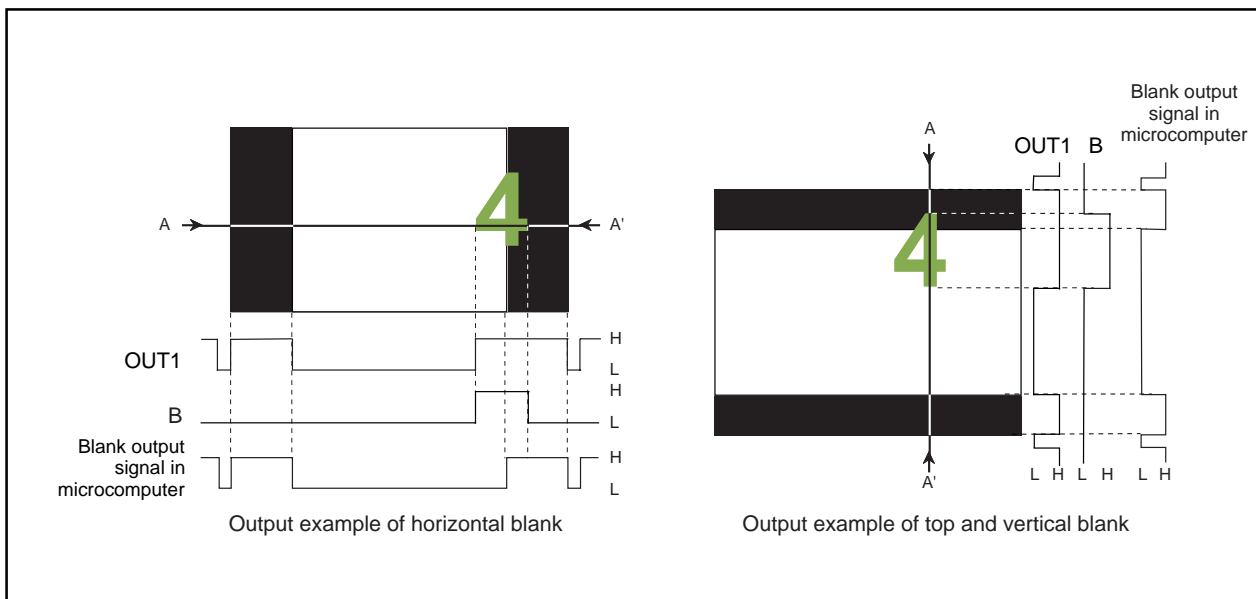
The ON/OFF for horizontal blank function is performed by bit 4 of the OSD control register 2 and is used interchangeably for the horizontal window function with bit 5 of the OSD control register 2. Accordingly, the horizontal blank function cannot be used simultaneously with the horizontal window function.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

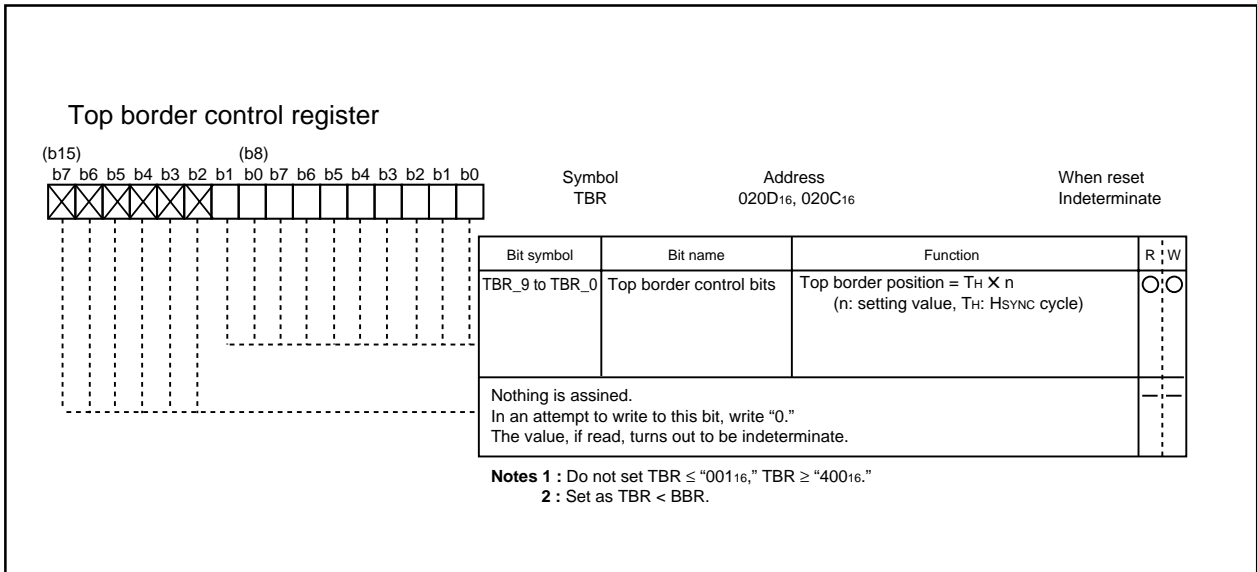
**2:** When using the window function, be sure to set "1" to bit 0 of OSD control register 1.



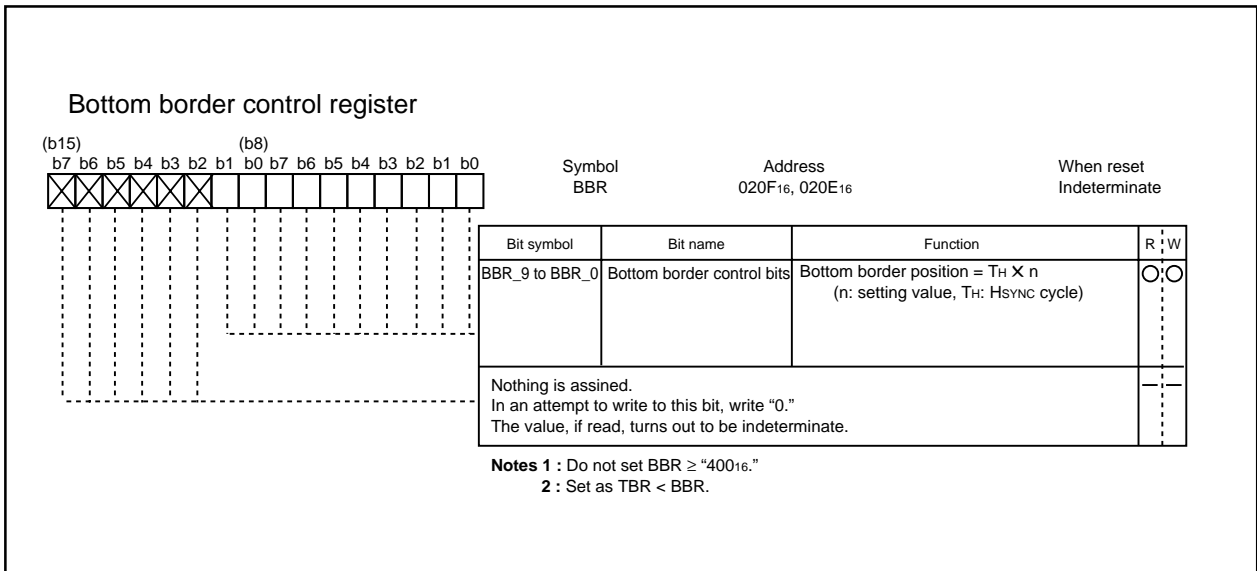
**Figure 2.16.46 Blank output example (when OSD output is B + OUT1)**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



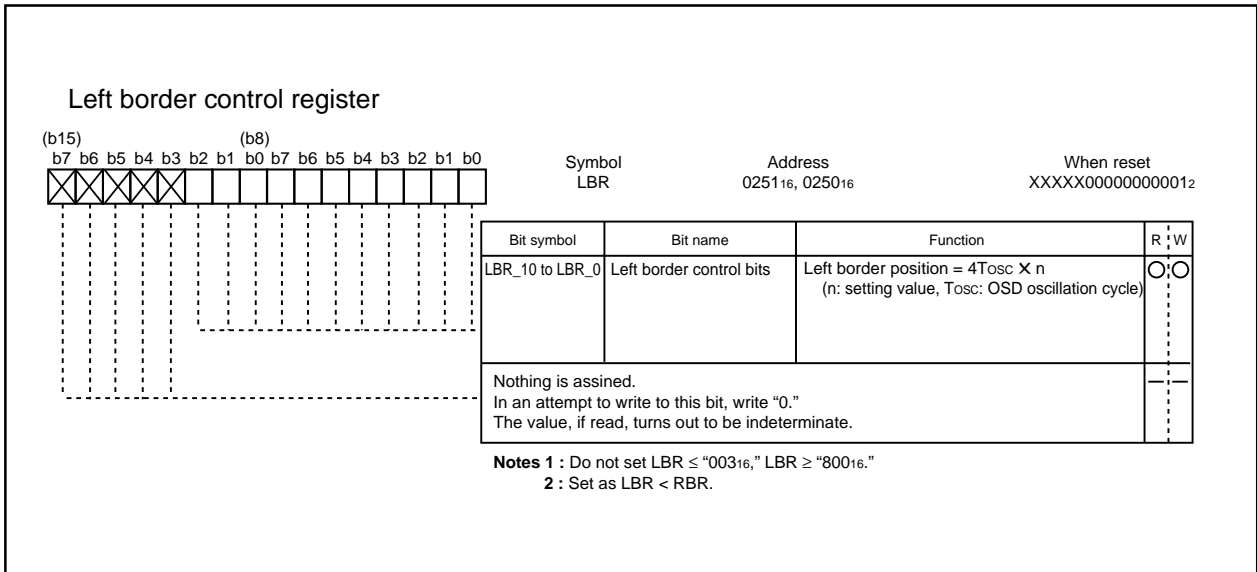
**Figure 2.16.47 Top border control register**



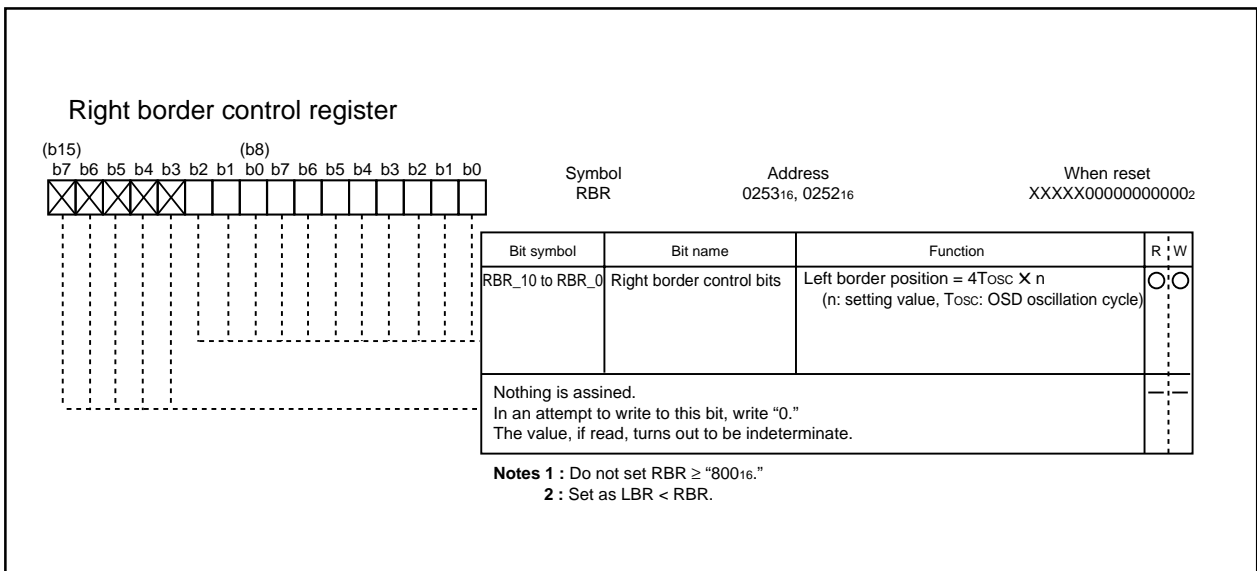
**Figure 2.16.48 Bottom border control register**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.16.49 Left border control register**



**Figure 2.16.50 Bottom border control register**

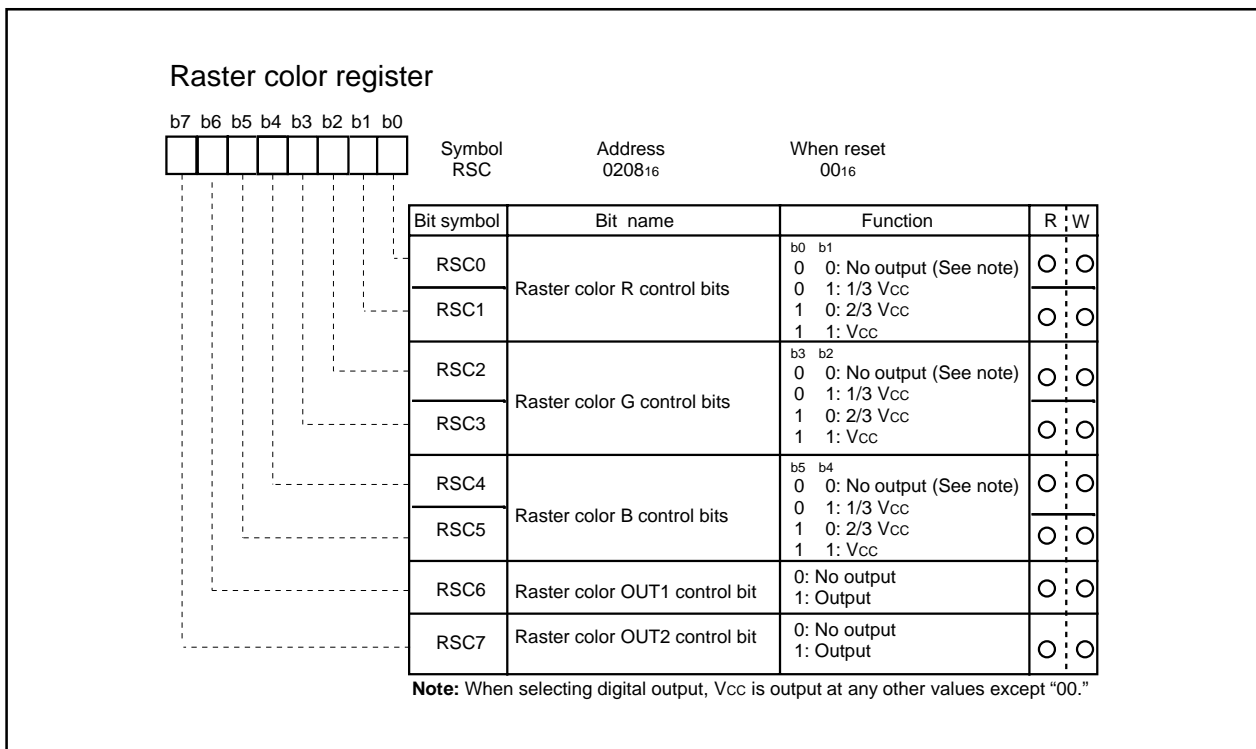
**2.16.16 Raster Coloring Function**

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 2.16.51, the example of raster coloring is shown in Figure 2.16.52.

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

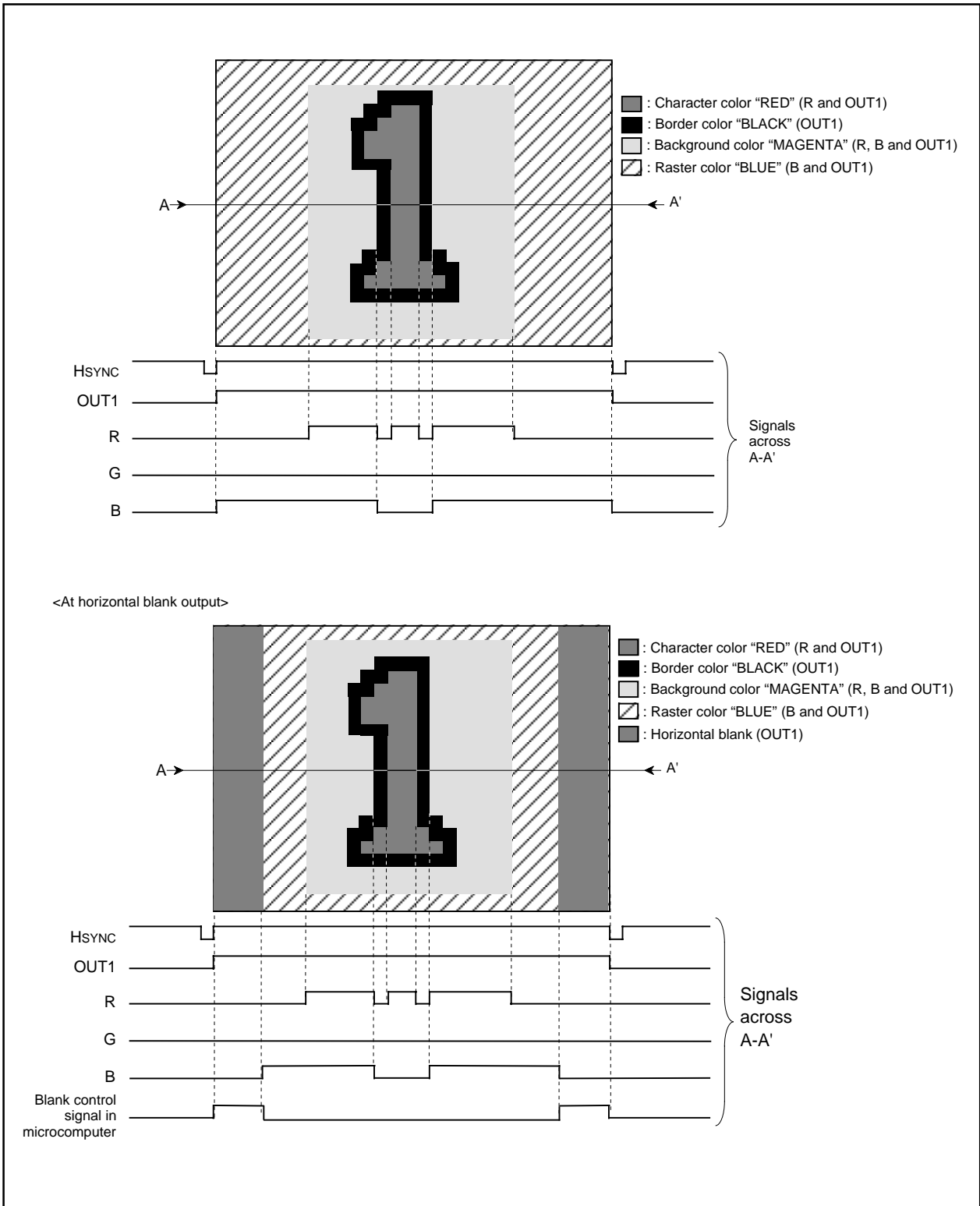


**Figure 2.16.51 Raster color register**



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



### 2.16.17 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 2.16.3).

**Table 2.16.12 Setting for scan mode**

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

### 2.16.18 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 2.16.13 R, G, B signal output control**

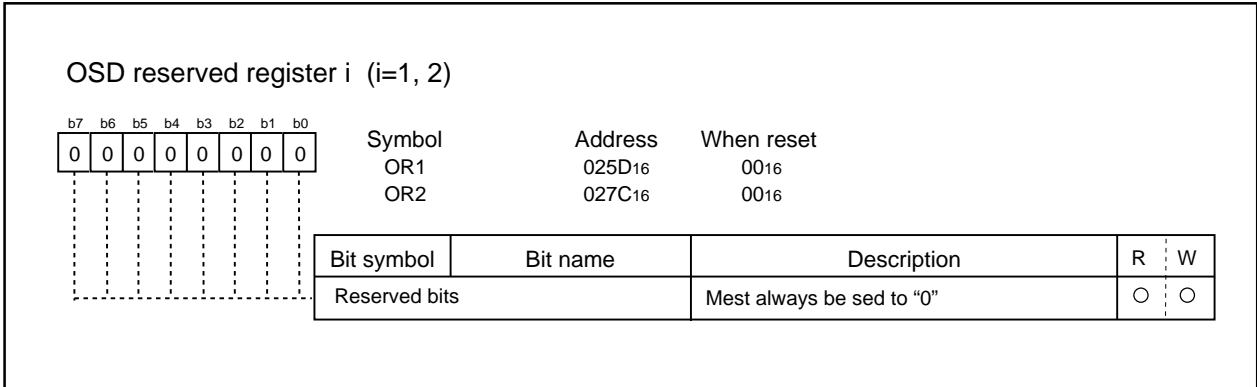
Bit 4 of clock control register	Bit 2 of OSD control register 2	Form of R, G, B signal output
0	0	Each R, G, B pin outputs 2 values (digital output).
	1	Each R, G, B pin outputs 4 values (analog output).
1		Each R, R' (P7 <sub>3</sub> ), G, G' (P6 <sub>0</sub> ), B, B' (P5 <sub>7</sub> ) pin outputs 2 values. (Corresponding to each signal output control bits of color palette register i) R, G, B: CRi_1, CRi_3, CRi_5, respectively R', G', B': CRi_0, CRi_2, CRi_4, respectively

**Note:** When bit 4 of the clock control register is "1," ports P5<sub>7</sub>, P6<sub>0</sub>, P7<sub>3</sub> function as OSD function pins R', G', B', respectively. When emulating, however, set bit 4 to "0."

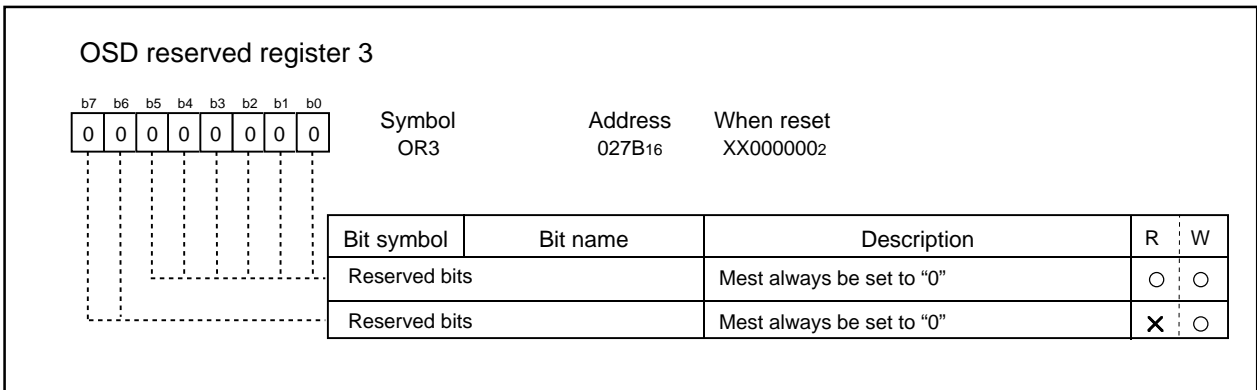
**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**2.16.19 OSD Reserved Register**



**Figure 2.16.53 OSD reserved register i (i=1, 2)**



**Figure 2.16.54 OSD reserved register 3**

## 2.17 Programmable I/O Ports

There are 78 programmable I/O ports: P0–P5, P60–P63, P67, P7, P82, P83, P86, P87, P90–P94 and P10. Each port can be set independently for input or output using the direction register. A pull-up resistance for each block of 4 ports can be set.

Figures 2.17.1 to 2.17.4 show the programmable I/O ports.

Each pin functions as a programmable I/O port and as the I/O for the built-in peripheral devices.

To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices (other than the D-A converter), they function as outputs regardless of the contents of the direction registers. When pins are to be used as the outputs for the D-A converter, do not set the direction registers to output mode. See the descriptions of the respective functions for how to set up the built-in peripheral devices.

### 2.17.1 Direction Registers

Figures 2.17.6 to 2.17.9 show the direction registers.

These registers are used to choose the direction of the programmable I/O ports. Each bit in these registers corresponds one for one to each I/O pin.

#### (1) Effect of the protection register

Data written to the direction register of P9 is affected by the protection register. The direction register of P9 cannot be easily written.

### 2.17.2 Port Registers

Figures 2.17.10 to 2.17.13 show the port registers.

These registers are used to write and read data for input and output to and from an external device. A port register consists of a port latch to hold output data and a circuit to read the status of a pin. Each bit in port registers corresponds one for one to each I/O pin.

#### (1) Reading a port register

With the direction register set to output, reading a port register takes out the content of the port register, not the content of the pin. With the direction register set to input, reading the port register takes out the content of the pin.

#### (2) Writing to a port register

With the direction register set to output, the level of the written values from each relevant pin is output by writing to a port register. Writing to the port register, with the direction register set to input, inputs a value to the port register, but nothing is output to the relevant pins. The output level remains floating.

### 2.17.3 Pull-up Control Registers

Figures 2.17.15 to 2.17.17 show the pull-up control registers.

The pull-up control register can be set to apply a pull-up resistance to each block of 4 ports. When ports are set to have a pull-up resistance, the pull-up resistance is connected only when the direction register is set for input.

However, in memory expansion mode and microprocessor mode, pull-up control register of P0 to P5 is invalid.

### 2.17.4 Port Control Register

Figure 2.17.14 shows the port control register.

The bit 0 of port control register is used to read port P1 as follows:

0: When port P1 is input port, port input level is read.

When port P1 is output port, the contents of port P1 register is read.

1: The contents of port P1 register is read through port P1 is input/output port.

This register is valid in the following :

- External bus width is 8 bits in microprocessor mode or memory expansion mode.
- Port P1 can be used as a port in multiplexed bus for the entire space.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

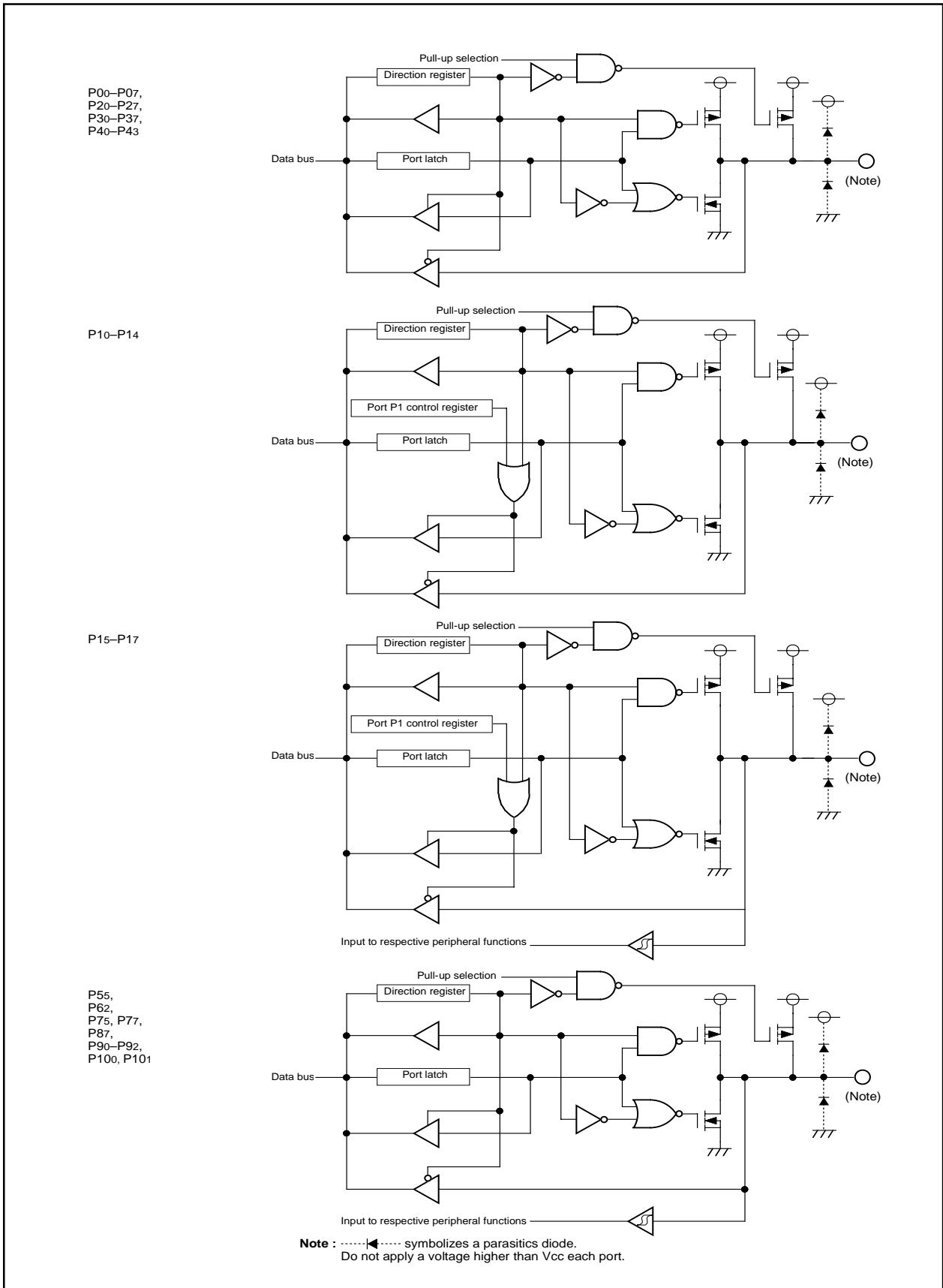
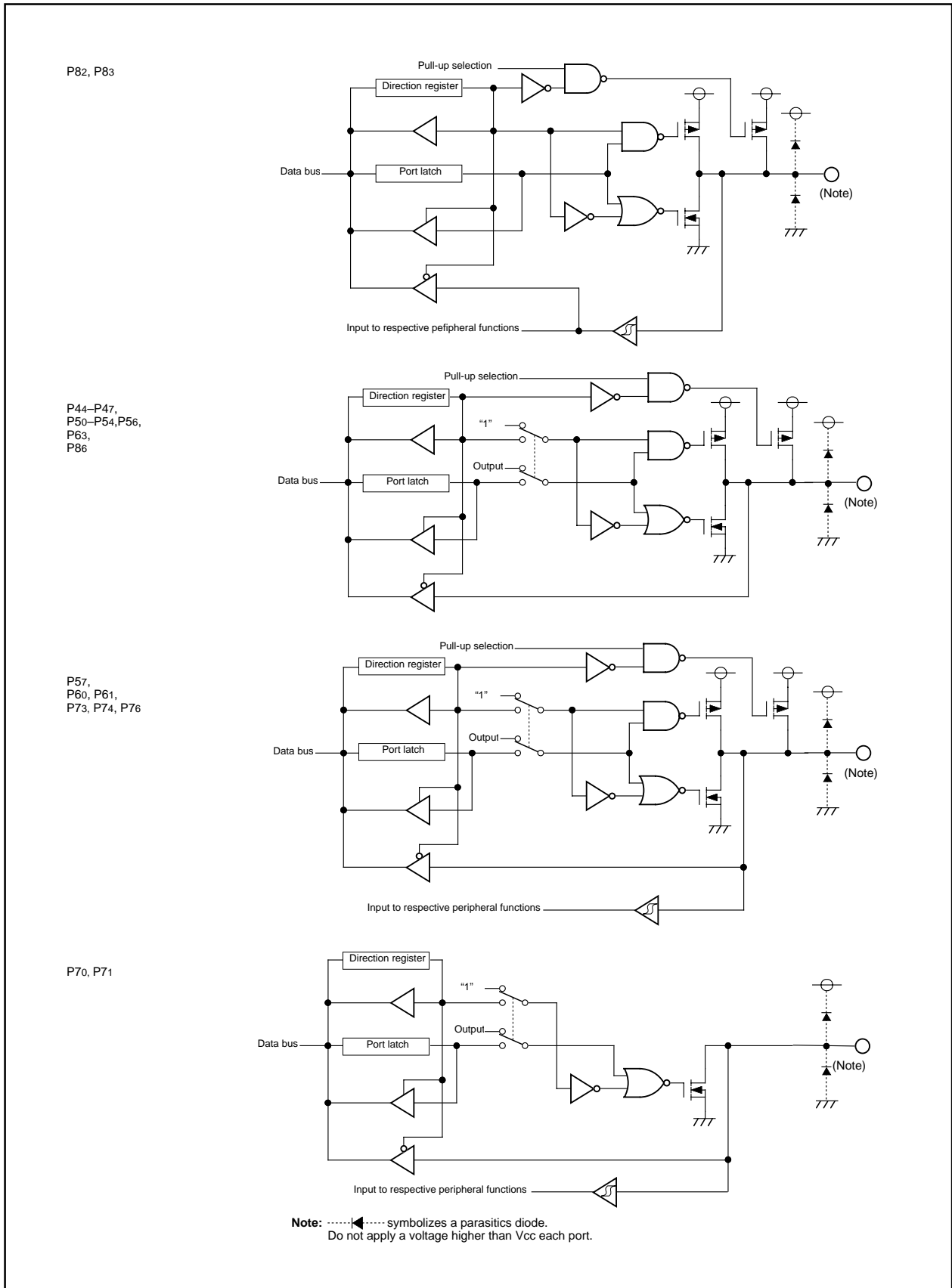


Figure 2.17.1 Programmable I/O ports (1)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.17.2 Programmable I/O ports (3)**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

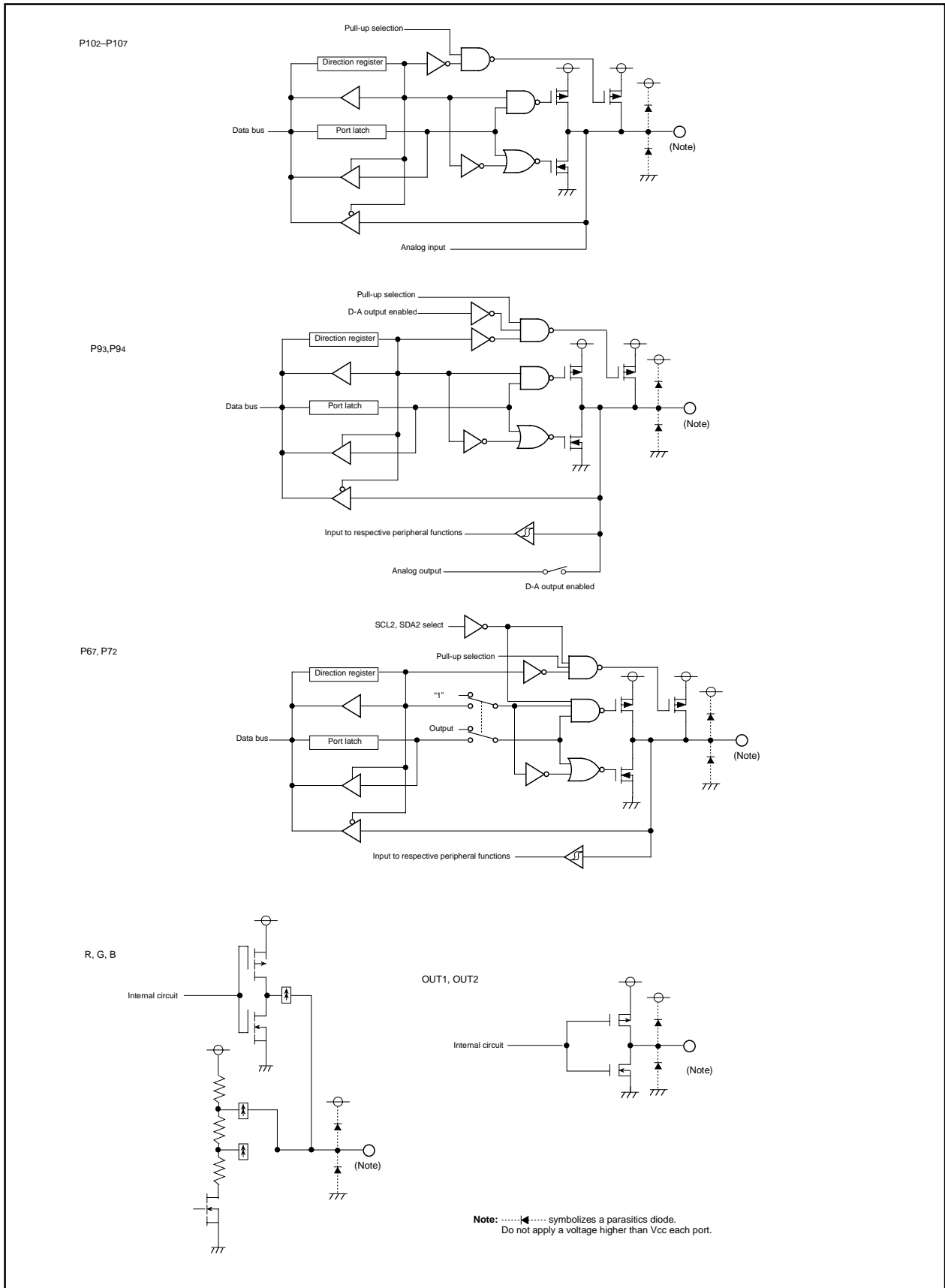
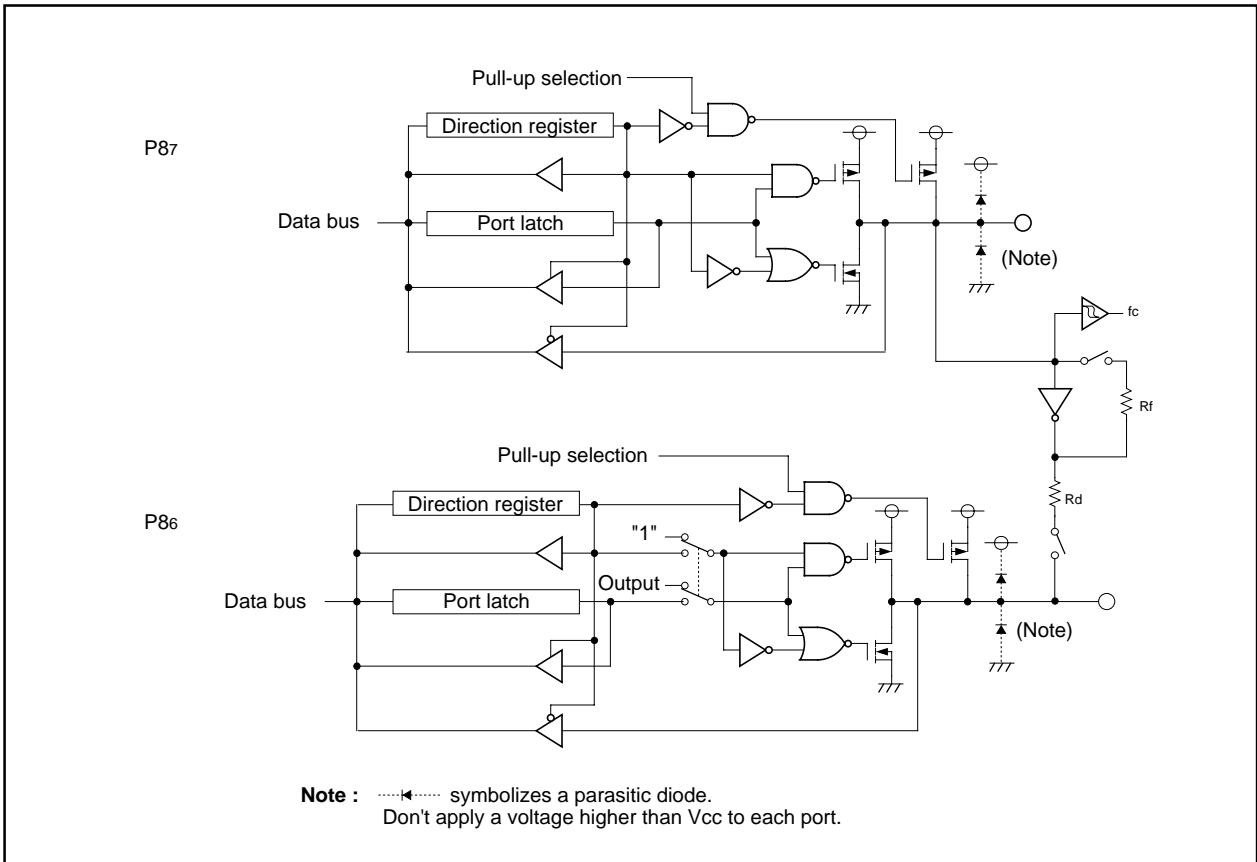


Figure 2.17.3 Programmable I/O ports (2)

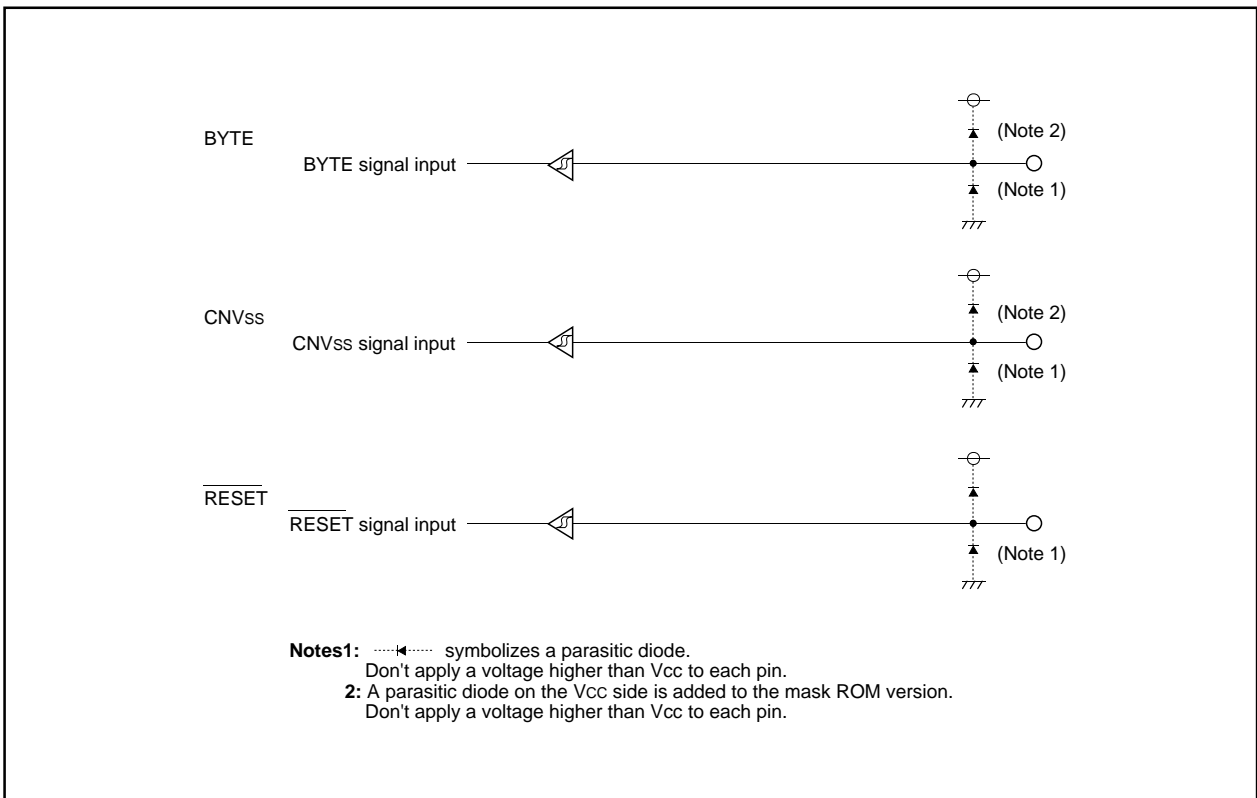


**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.17.4 Programmable I/O ports (4)**



**Figure 2.17.5 I/O pins**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

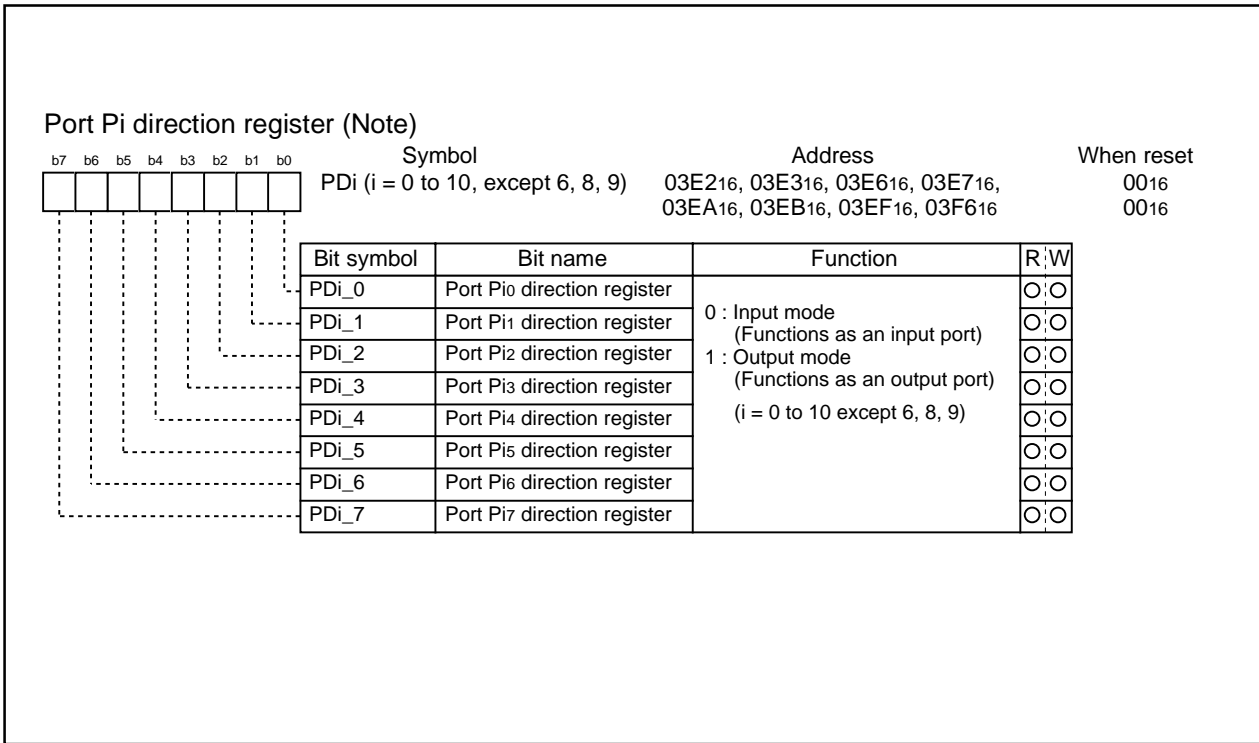


Figure 2.17.6 Port Pi direction register (i = 0 to 10, except 6, 8, 9)

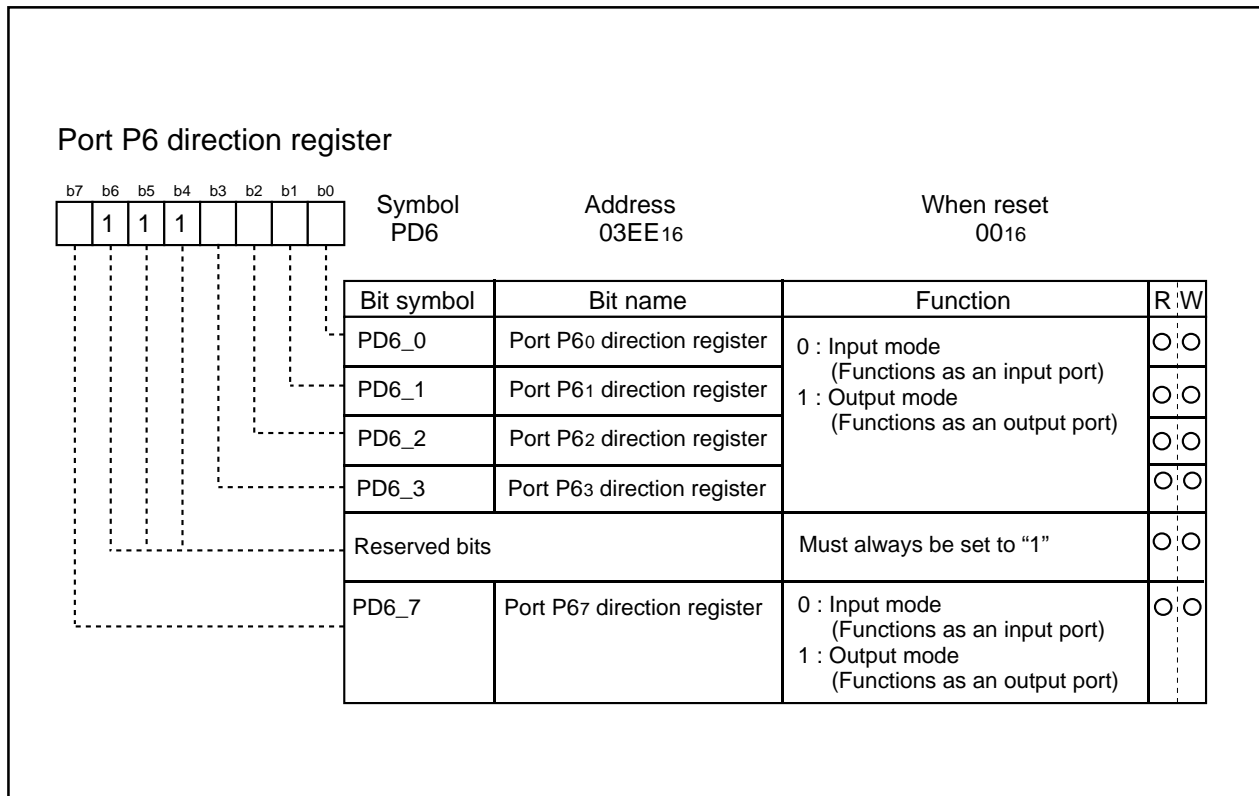
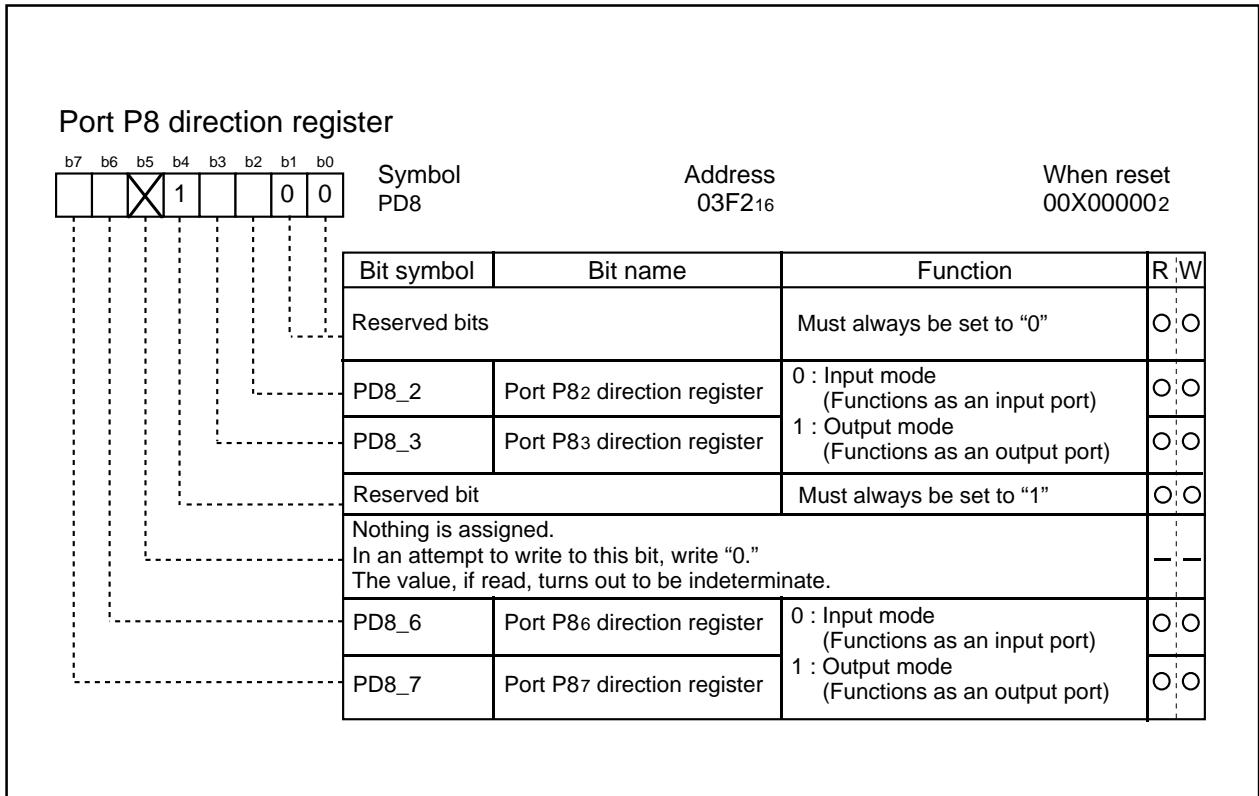


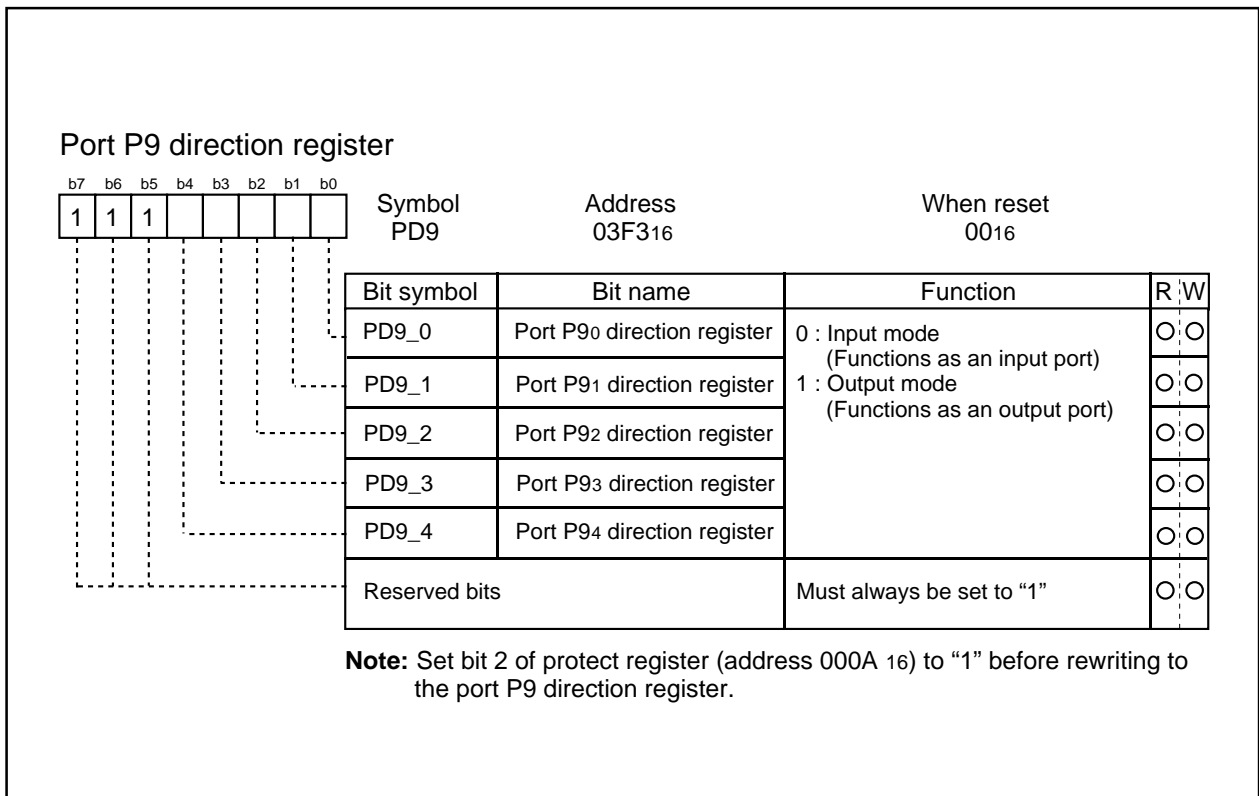
Figure 2.17.7 Port P6 direction register

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.17.8 Port P8 direction register**



**Figure 2.17.9 Port P9 direction register**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

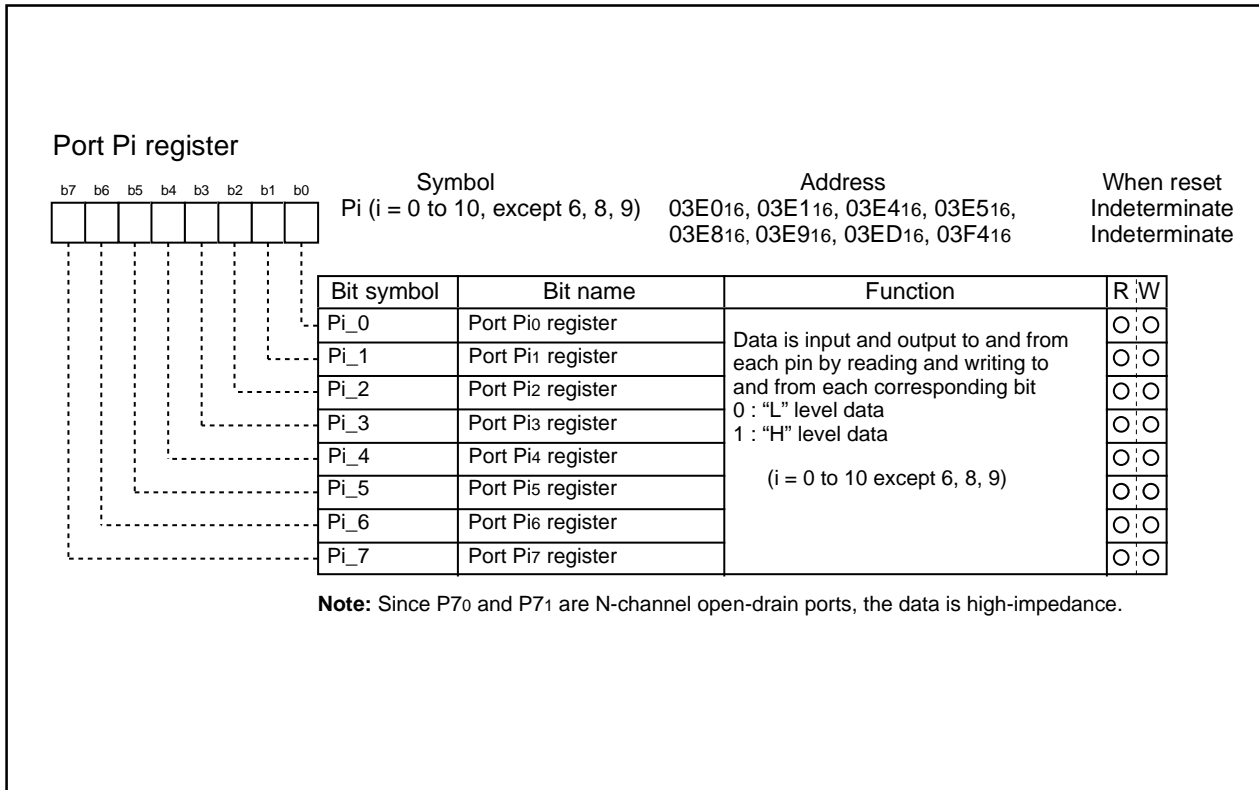


Figure 2.17.10 Port Pi register (i = 0 to 10, except 6, 7, 8)

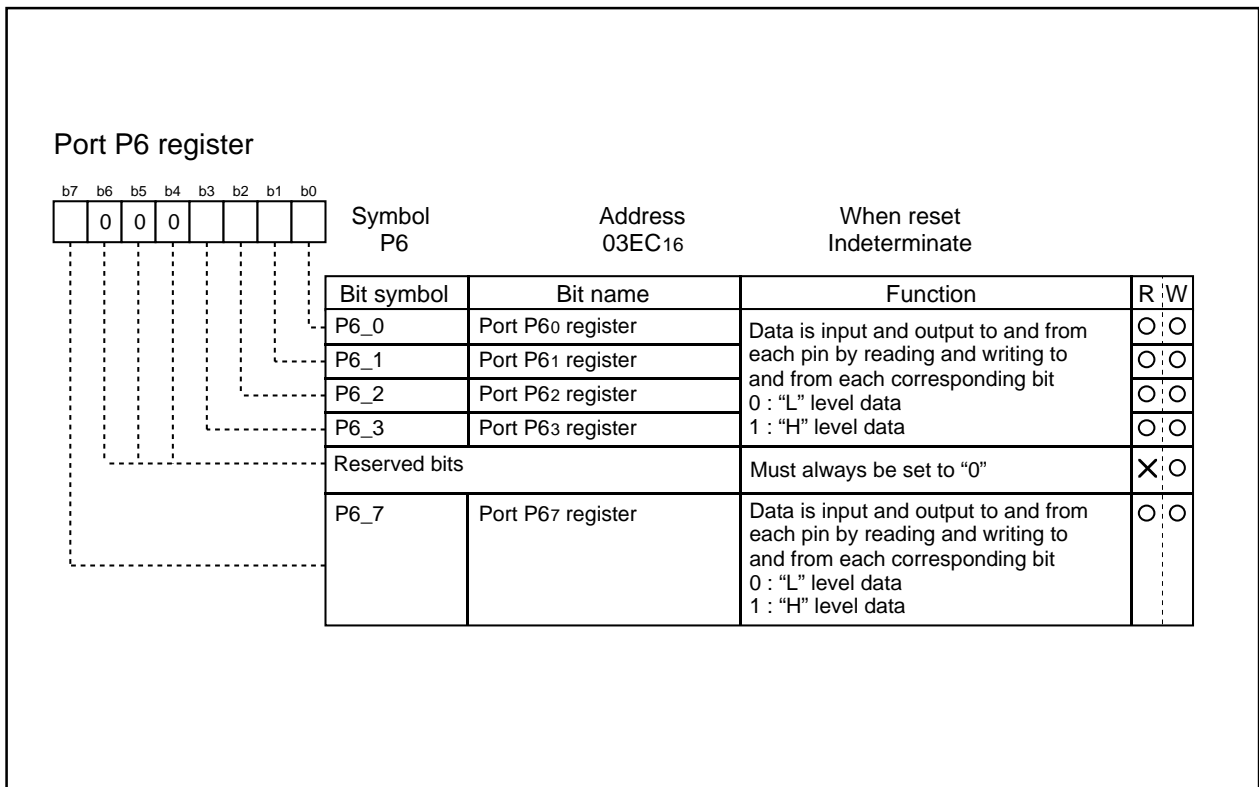
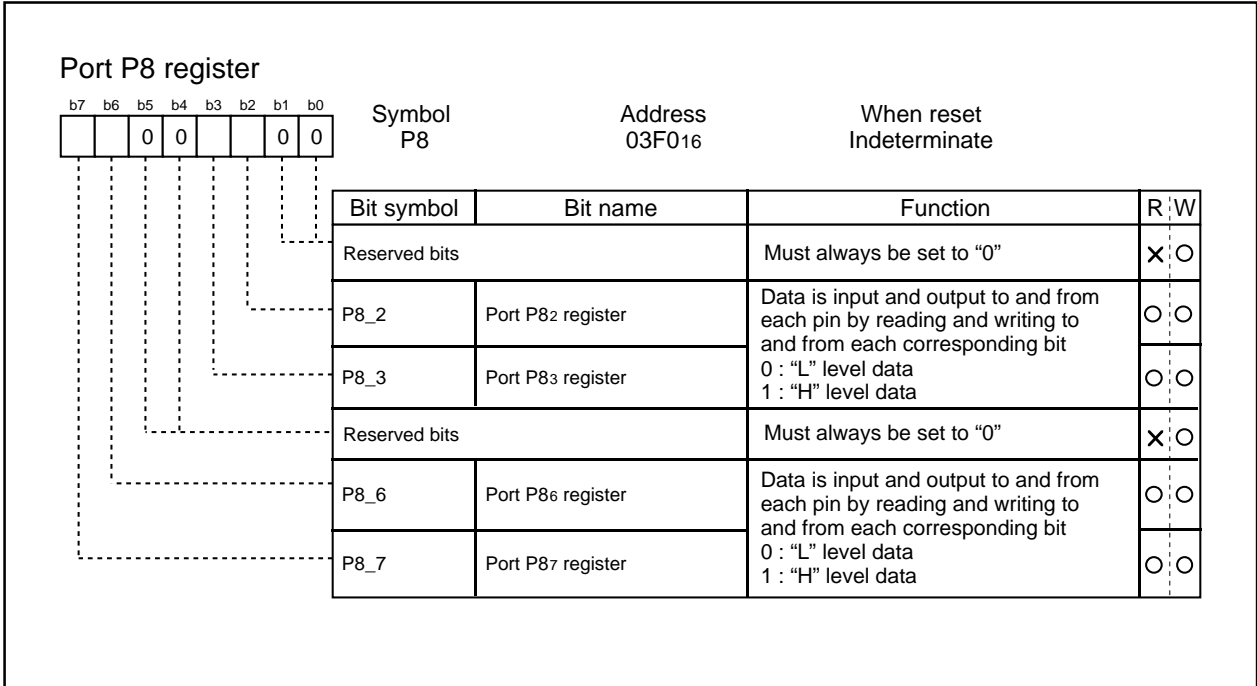


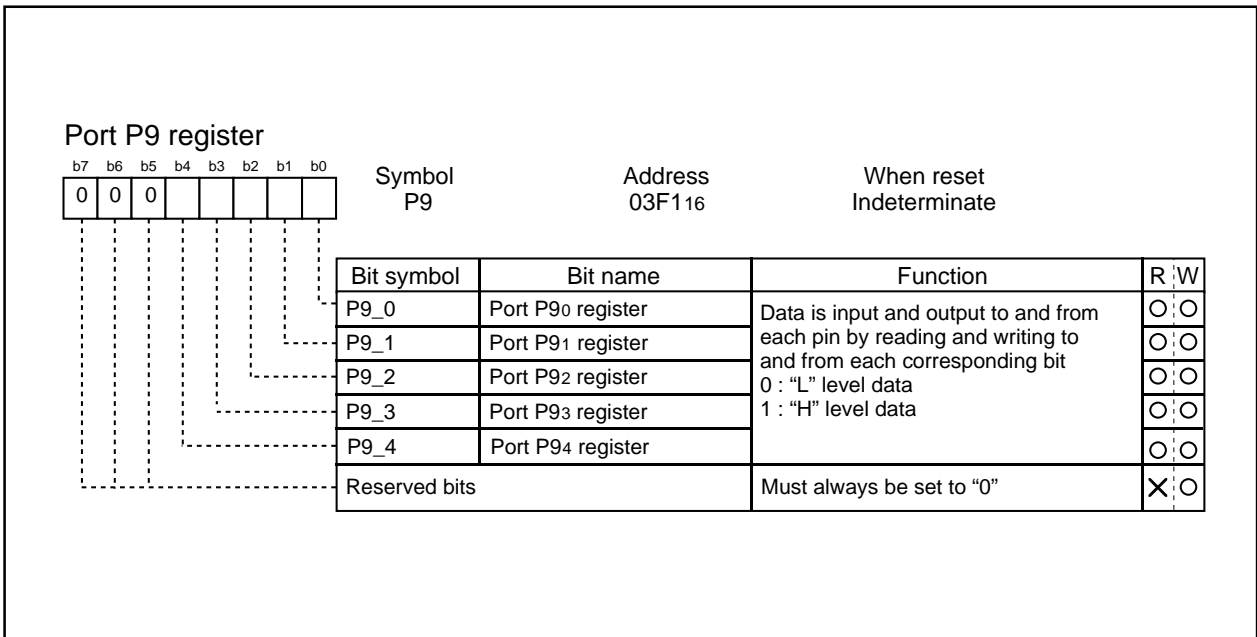
Figure 2.17.11 Port P6 register

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.17.12 Port P8 register**



**Figure 2.17.13 Port P9 register 0**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

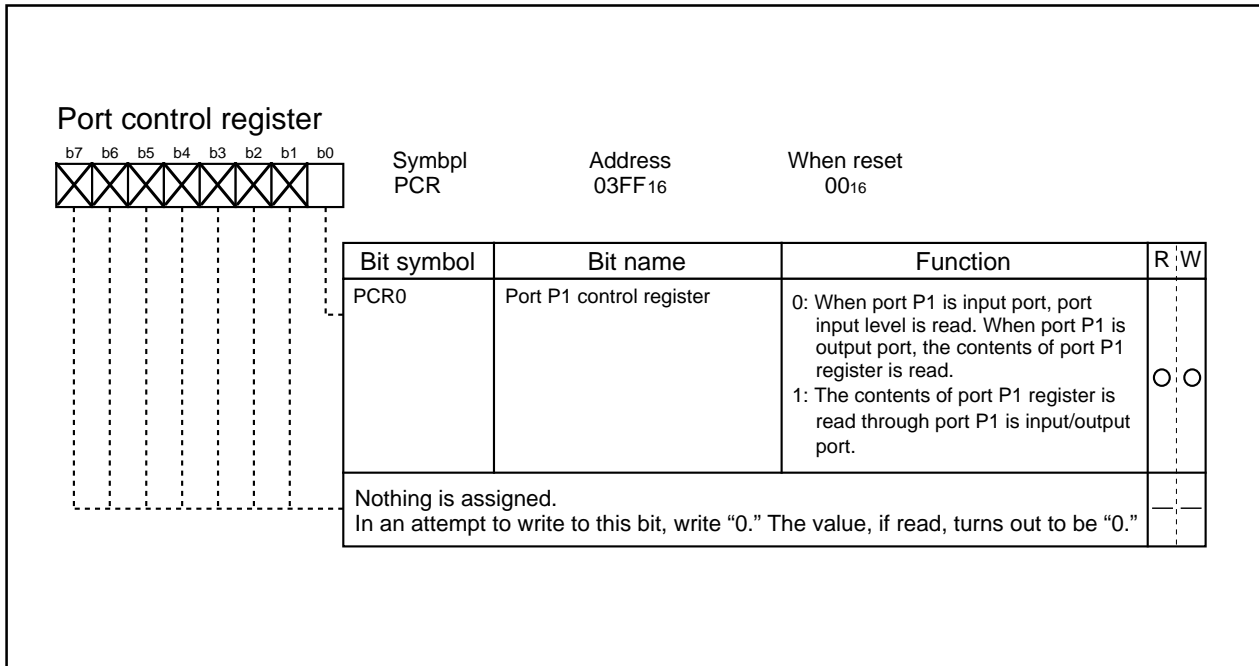


Figure 2.17.14 Port control register

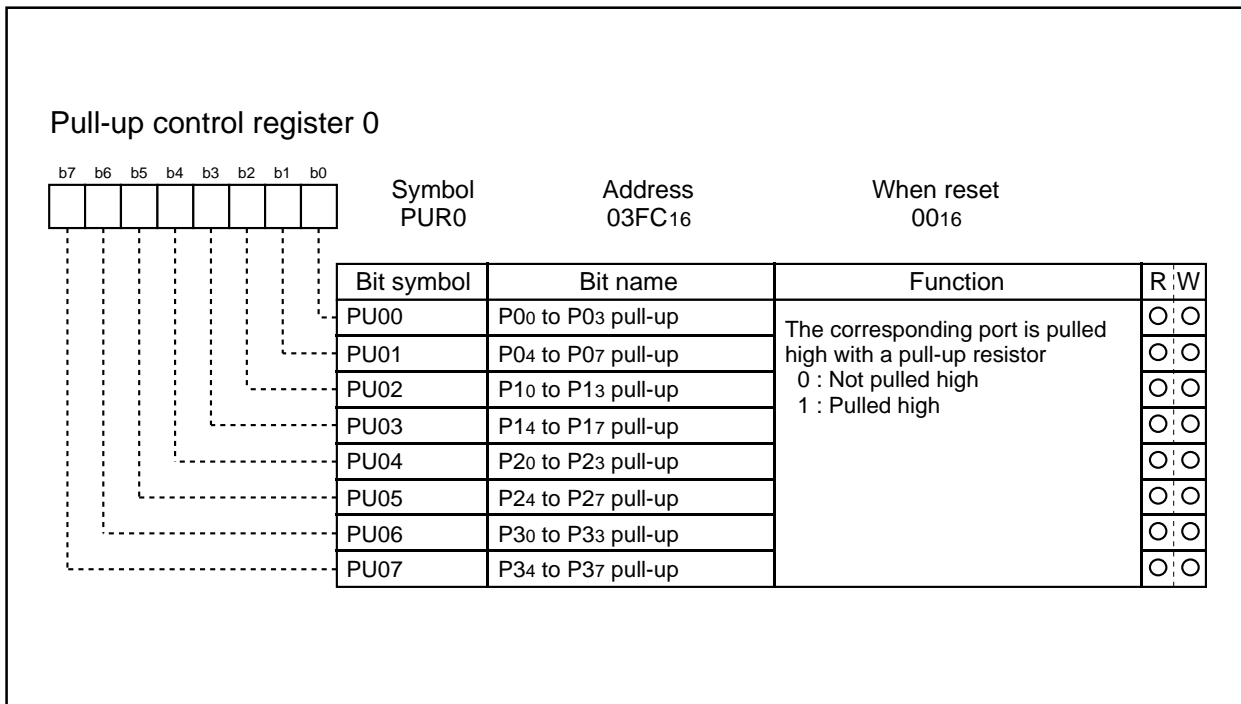


Figure 2.17.15 Pull-up control register 0

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

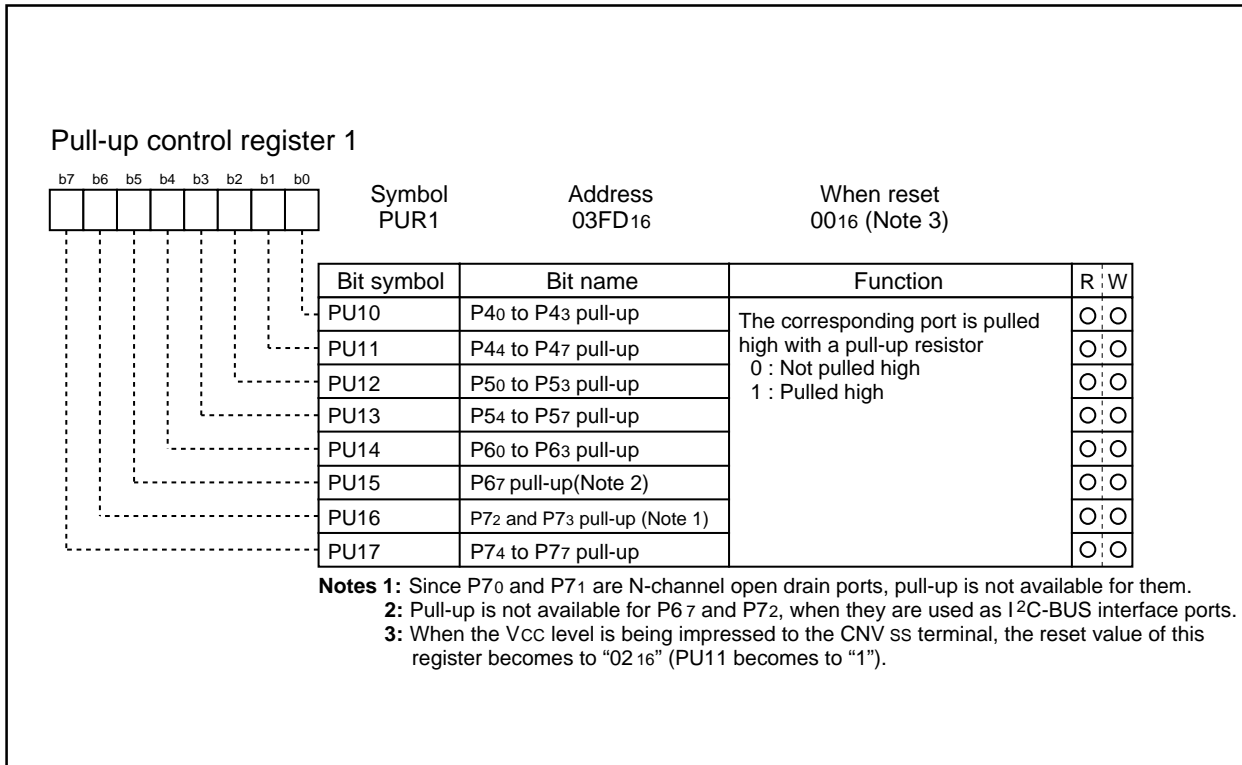


Figure 2.17.16 Pull-up control register 1

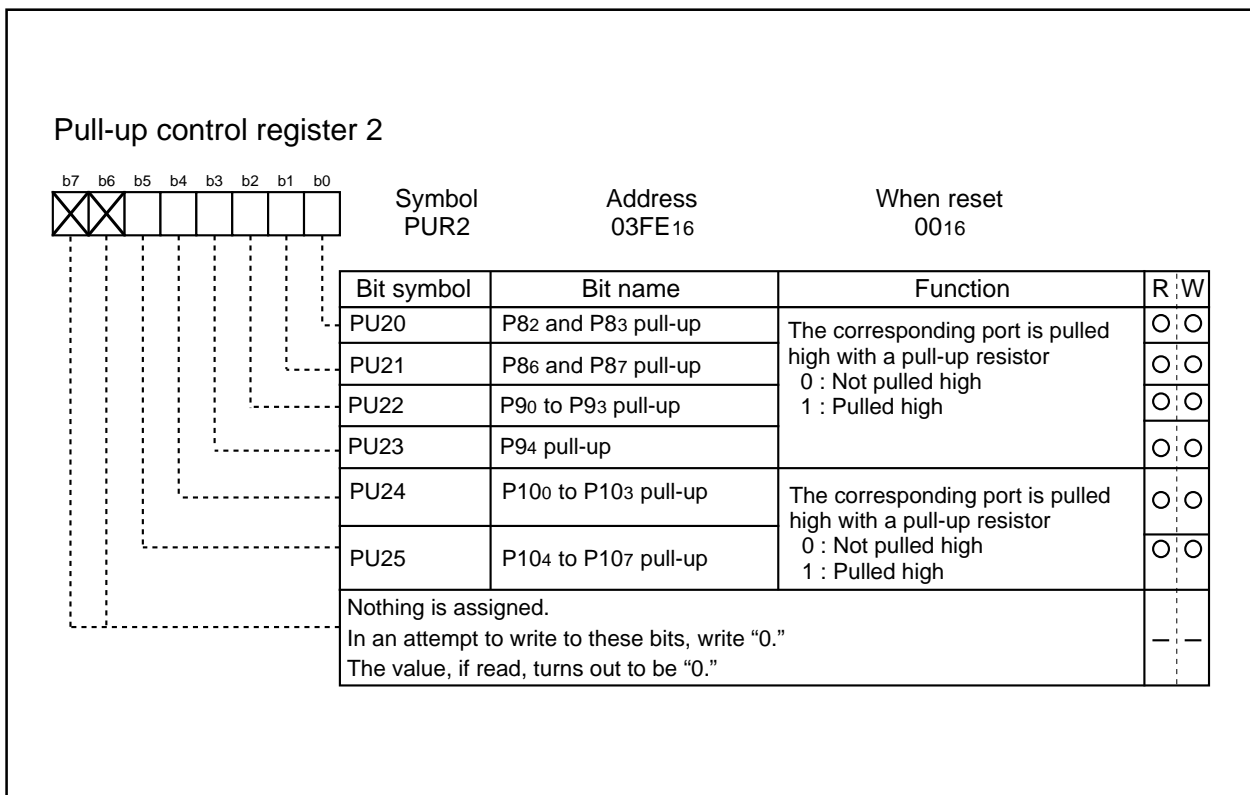


Figure 2.17.17 Pull-up control register 2

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 2.17.1 Example connection of unused pins in single-chip mode**

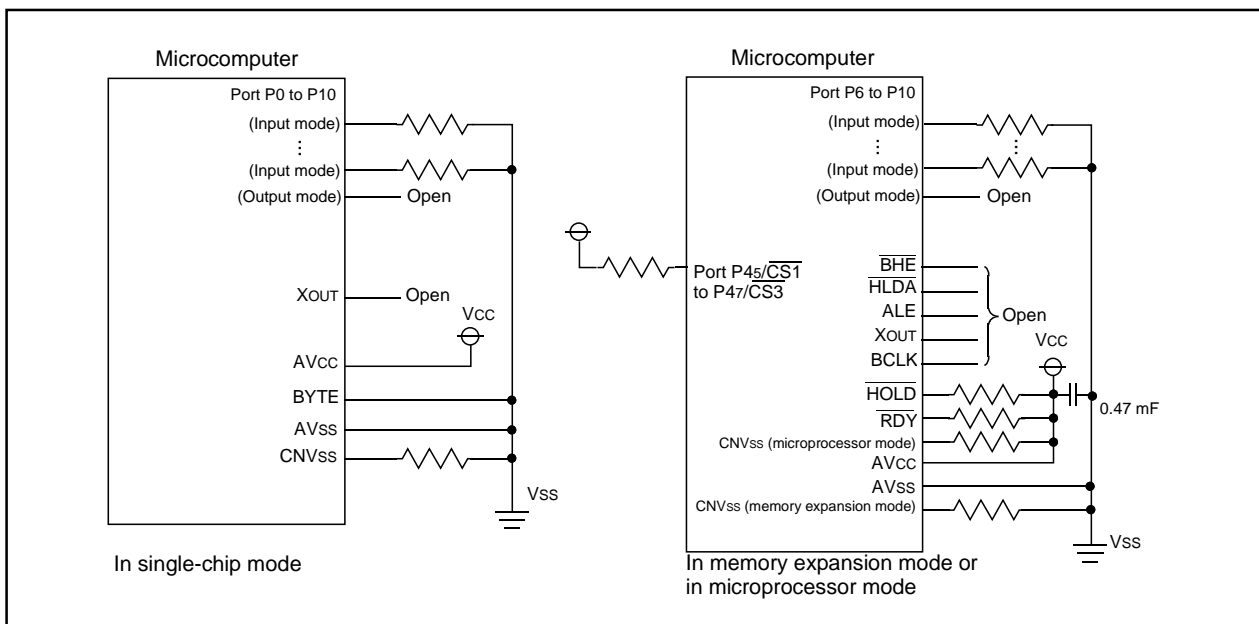
Pin name	Connection
Ports P0 to P10	After setting for input mode, connect every pin to V <sub>SS</sub> or V <sub>CC</sub> via a resistor; or after setting for output mode, leave these pins open.
XOUT (Note)	Open
AV <sub>CC</sub>	Connect to V <sub>CC</sub>
AV <sub>SS</sub> , BYTE	Connect to V <sub>SS</sub>
CNV <sub>SS</sub>	Connect via resistor to V <sub>SS</sub> (Pull-down)

**Note:** With external clock input to X IN pin.

**Table 2.17.2 Example connection of unused pins in memory expansion mode and microprocessor mode**

Pin name	Connection
Ports P6 to P10	After setting for input mode, connect every pin to V <sub>SS</sub> or V <sub>CC</sub> via a resistor; or after setting for output mode, leave these pins open.
P4 <sub>5</sub> /CS1 to P4 <sub>7</sub> /CS3	Sets ports to input mode, sets bits CS1 through CS3 to "0," and connects to V <sub>CC</sub> via resistors (pull-up).
BHE, ALE, HLDA, XOUT(Note), BCLK	Open
HOLD, RDY	Connect via resistor to V <sub>CC</sub> (pull-up)
AV <sub>CC</sub>	Connect to V <sub>CC</sub>
AV <sub>SS</sub>	Connect to V <sub>SS</sub>
CNV <sub>SS</sub>	Connect via resistor to V <sub>SS</sub> (pull-down) in the memory expansion mode. Connect via resistor to V <sub>CC</sub> (pull-up) in the microprocessor mode.

**Note:** With external clock input to X IN pin.



**Figure 2.17.18 Example connection of unused pins**



### 3. USAGE PRECAUTION

#### 3.1 Timer A (timer mode)

- (1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets "FFFF<sub>16</sub>". Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.

#### 3.2 Timer A (event counter mode)

- (1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets "FFFF<sub>16</sub>" by underflow or "0000<sub>16</sub>" by overflow. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.
- (2) When stop counting in free run type, set timer again.

#### 3.3 Timer A (one-shot timer mode)

- (1) Setting the count start flag to "0" while a count is in progress causes as follows:
  - The counter stops counting and a content of reload register is reloaded.
  - The TAIOUT pin outputs "L" level.
  - The interrupt request generated and the timer Ai interrupt request bit goes to "1".
- (2) The timer Ai interrupt request bit goes to "1" if the timer's operation mode is set using any of the following procedures:
  - Selecting one-shot timer mode after reset.
  - Changing operation mode from timer mode to one-shot timer mode.
  - Changing operation mode from event counter mode to one-shot timer mode.Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.

#### 3.4 Timer A (pulse width modulation mode)

- (1) The timer Ai interrupt request bit becomes "1" if setting operation mode of the timer in compliance with any of the following procedures:
  - Selecting PWM mode after reset.
  - Changing operation mode from timer mode to PWM mode.
  - Changing operation mode from event counter mode to PWM mode.Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.
- (2) Setting the count start flag to "0" while PWM pulses are being output causes the counter to stop counting. If the TAIOUT pin is outputting an "H" level in this instance, the output level goes to "L", and the timer Ai interrupt request bit goes to "1". If the TAIOUT pin is outputting an "L" level in this instance, the level does not change, and the timer Ai interrupt request bit does not becomes "1".

### 3.5 Timer B (timer mode, event counter mode)

- (1) Reading the timer Bi register while a count is in progress allows reading , with arbitrary timing, the value of the counter. Reading the timer Bi register with the reload timing gets "FFFF16". Reading the timer Bi register after setting a value in the timer Bi register with a count halted but before the counter starts counting gets a proper value.

### 3.6 Timer B (pulse period, pulse width measurement mode)

- (1) If changing the measurement mode select bit is set after a count is started, the timer Bi interrupt request bit goes to "1".
- (2) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

### 3.7 A-D Converter

- (1) Write to each bit (except bit 6) of A-D control register 0, to each bit of A-D control register 1, and to bit 0 of A-D control register 2 when A-D conversion is stopped (before a trigger occurs).  
In particular, when the Vref connection bit is changed from "0" to "1", start A-D conversion after an elapse of 1  $\mu$ s or longer.
- (2) When changing A-D operation mode, select analog input pin again.
- (3) When using A-D converter in the one-shot mode and in the single sweep mode  
After confirming the completion of A-D conversion, read the A-D register (the completion of A-D conversion is determined by A-D interrupt request bit).
- (4) When using A-D converter in the repeat mode and in the repeat sweep mode  
Use the main clock without dividing as the internal clock of CPU.
- (5) The A-D conversion in the sweep mode needs the time as follows; (number of sweep pins + 2 pins)  $\times$  repeat times  $\times$  A-D conversion time for 1 pin.
- (6) When operating OSD or operating data slicer using the HSYNC and VSYNC input, do not use the A-D sweep mode (single sweep mode, repeat sweep mode 0, and repeat sweep mode 1).

### 3.8 Stop Mode and Wait Mode

- (1) When returning from stop mode by hardware reset,  $\overline{\text{RESET}}$  pin must be set to "L" level until main clock oscillation is stabilized.
- (2) When switching to either wait mode or stop mode, instructions occupying four bytes either from the WAIT instruction or from the instruction that sets the every-clock stop bit to "1" within the instruction queue are perfected and then the program stops. So put at least four NOPs in succession either to the WAIT instruction or to the instruction that sets the every-clock stop bit to "1."

### 3.9 Interrupts

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

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.

The interrupt request bit of the certain interrupt written in address 00000<sub>16</sub> will then be set to "0". Reading address 00000<sub>16</sub> by software sets enabled highest priority interrupt source request bit to "0".

Though the interrupt is generated, the interrupt routine may not be executed.

Do not read address 00000<sub>16</sub> by software.

#### (2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000<sub>16</sub>. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt.

#### (3) External interrupt

- When the polarity of the  $\overline{\text{INT}}_0$  and  $\overline{\text{INT}}_1$  pins is changed, the interrupt request bit is sometimes set to "1." After changing the polarity, set the interrupt request bit to "0."

#### (4) Rewrite the interrupt control register

- To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

#### Example 1:

```
INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  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 ; Clear TA0IC int. priority level and int. request bit.
  MOV.W MEM, R0     ; Dummy read.
  FSET  I           ; Enable interrupts.
```

#### Example 3:

```
INT_SWITCH3:
  PUSHC FLG         ; Push Flag register onto stack
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  POPC  FLG         ; Enable interrupts.
```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

- When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

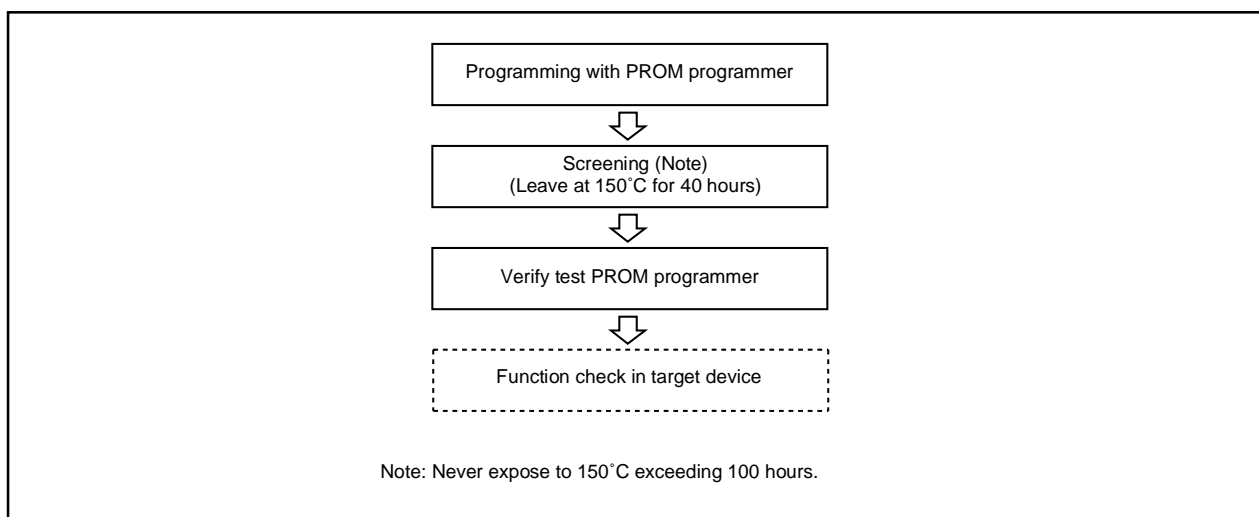
### 3.10 Built-in PROM version

#### 3.10.1 All built-in PROM versions

High voltage is required to program to the built-in PROM. Be careful not to apply excessive voltage. Be especially careful during power-on.

#### 3.10.2 One Time PROM version

One Time PROM versions shipped in blank, of which built-in PROMs are programmed by users, are also provided. For these microcomputers, a programming test and screening are not performed in the assembly process and the following processes. To improve their reliability after programming, we recommend to program and test as flow shown in Figure 3.10.1 before use.



**Figure 3.10.1 Programming and test flow for One Time PROM version**

#### **4. ITEMS TO BE SUBMITTED WHEN ORDERING MASKED ROM VERSION**

Please submit the following when ordering masked ROM products.

- (1) Mask ROM confirmation form
- (2) Mask specification sheet
- (3) ROM data : EPROMs (3 sets)

\*: In the case of EPROMs, three sets of EPROMs are required per pattern.

\*: In the case of floppy disks, 3.5-inch double-sided high-density disk (IBM format) is required per pattern.

## 5. ELECTRICAL CHARACTERISTICS

### 5.1. Absolute Maximum Ratings

**Table 5.1.1 Absolute maximum ratings**

Symbol	Parameter	Condition	Rated value	Unit
V <sub>cc</sub>	Supply voltage		-0.3 to 6.5	V
AV <sub>cc</sub>	Analog supply voltage		-0.3 to 6.5	V
V <sub>i</sub>	Input voltage P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>3</sub> , P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>2</sub> , P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>4</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>IN</sub> , OSC1, RESET		-0.3 to V <sub>cc</sub> +0.3	V
V <sub>i</sub>	Input voltage CNV <sub>ss</sub> , BYTE		-0.3 to 6.5 (Note)	V
V <sub>o</sub>	Output voltage P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>3</sub> , P6 <sub>7</sub> , P7 <sub>2</sub> to P7 <sub>7</sub> , P8 <sub>2</sub> , P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>4</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , R, G, B, OUT1, OUT2, OSC2, XOUT		-0.3 to V <sub>cc</sub> +0.3	V
V <sub>o</sub>	Output voltage P7 <sub>0</sub> , P7 <sub>1</sub>		-0.3 to 6.5	V
P <sub>d</sub>	Power dissipation	T <sub>a</sub> =25 °C	500	mW
T <sub>opr</sub>	Operating ambient temperature		-10 to 70	°C
T <sub>stg</sub>	Storage temperature		-40 to 125	°C

**Note:** When writing to EPROM, only CNVSS is -0.3 to 13(V).

# M306V0ME-XXXFP

# M306V0EEFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

## 5.2 Recommended Operating Conditions

**Table 5.2.1 Recommended operating conditions (referenced to  $V_{CC} = 4.5\text{ V}$  to  $5.5\text{ V}$  at  $T_a = -10\text{ }^\circ\text{C}$  to  $70\text{ }^\circ\text{C}$  unless otherwise specified)**

Symbol	Parameter		Standard			Unit
			Min	Typ.	Max.	
$V_{CC}$	Supply voltage (Note 3)		4.5	5.0	5.5	V
$AV_{CC}$	Analog supply voltage (Note 3)			$V_{CC}$		V
$V_{SS}$	Supply voltage			0		V
$AV_{SS}$	Analog supply voltage			0		V
$V_{IH}$	HIGH input voltage	P31 to P37, P40 to P47, P50 to P57, P60 to P63, P67, P70 to P77, P82, P83, P86, P87, P90 to P94, P100 to P107, HLF, $V_{HOLD}$ , $CV_{IN}$ , TVSETB, $X_{IN}$ , OSC1, RESET, $CNV_{SS}$ , BYTE	0.8 $V_{CC}$		$V_{CC}$	V
$V_{IH}$	HIGH input voltage	P00 to P07, P10 to P17, P20 to P27, P30 (during single-chip mode)	0.8 $V_{CC}$		$V_{CC}$	V
$V_{IH}$	HIGH input voltage	P00 to P07, P10 to P17, P20 to P27, P30 (data input function during memory expansion and microprocessor modes)	0.5 $V_{CC}$		$V_{CC}$	V
$V_{IL}$	LOW input voltage	P31 to P37, P40 to P47, P50 to P57, P60 to P63, P67, P70 to P77, P82, P83, P86, P87, P90 to P94, P100 to P107, $X_{IN}$ , OSC1, RESET, $CNV_{SS}$ , BYTE	0		0.2 $V_{CC}$	V
$V_{IL}$	LOW input voltage	P00 to P07, P10 to P17, P20 to P27, P30 (during single-chip mode)	0		0.2 $V_{CC}$	V
$V_{IL}$	LOW input voltage	P00 to P07, P10 to P17, P20 to P27, P30 (data input function during memory expansion and microprocessor modes)	0		0.16 $V_{CC}$	V
$I_{OH(peak)}$	HIGH peak output current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P63, P72 to P77, P82, P83, P86, P87, P90 to P94, P100 to P107, R, G, B, OUT1, OUT2			-10.0	mA
$I_{OH(avg)}$	HIGH average output current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P63, P67, P72 to P77, P82, P83, P86, P87, P90 to P94, P100 to P107, R, G, B, OUT1, OUT2			-5.0	mA
$I_{OL(peak)}$	LOW peak output current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P63, P67, P72 to P77, P82, P83, P86, P87, P90 to P94, P100 to P107, R, G, B, OUT1, OUT2			10.0	mA
$I_{OL(avg)}$	LOW average output current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P63, P67, P70 to P77, P82, P83, P86, P87, P90 to P94, P100 to P107, R, G, B, OUT1, OUT2			5.0	mA
$f(X_{IN})$	Main clock input oscillation frequency				10	MHz
$f(X_{CIN})$	Sub-clock oscillation frequency			32.768	50	kHz
$f_{OSC}$	Oscillation frequency (for OSD) OSC1	LC oscillating mode	11.0		27.0	MHz
		Ceramic oscillating mode	24.0		25.0	
$f_{CVIN}$	Input frequency	Horizontal sync. signal of video signal	15.262	15.743	16.206	kHz
$V_I$	Input amplitude video signal	$CV_{IN}$	1.5	2.0	2.5	V

**Notes 1:** The mean output current is the mean value within 100 ms.

**2:** The total  $I_{OL(peak)}$  for ports P0, P1, P2, P86, P87, P9, and P10 must be 80 mA max. The total  $I_{OH(peak)}$  for ports P0, P1, P2, P86, P87, P9, and P10 must be 80 mA max. The total  $I_{OL(peak)}$  for ports P3, P4, P5, P6, P7, P82 and P83 must be 80 mA max. The total  $I_{OH(peak)}$  for ports P3, P4, P5, P6, P72 to P77, P82 and P83 must be 80 mA max.

**3:** Connect 0.1  $\mu\text{F}$  or more capacitor externally between the power source pins  $V_{CC}-V_{SS}$  and  $AV_{CC}-AV_{SS}$  so as to reduce power source noise. Also connect 0.1  $\mu\text{F}$  or more capacitor externally between the power source pins  $V_{CC}-C_{NVSS}$ .

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**5.3 Electrical Characteristics**

**Table 5.3.1 Electrical characteristics (referenced to V<sub>CC</sub> = 5 V, V<sub>SS</sub> = 0 V at Ta = 25 °C, f(X<sub>IN</sub>) = 10 MHz unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit	
				Min.	Typ.	Max.		
V <sub>OH</sub>	HIGH output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>3</sub> , P6 <sub>7</sub> , P7 <sub>2</sub> to P7 <sub>7</sub> , P8 <sub>2</sub> , P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>4</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , R, G, B, OUT1, OUT2	I <sub>OH</sub> = -5 mA	3.0			V	
V <sub>OH</sub>	HIGH output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub>	I <sub>OH</sub> = -200 μA	4.7			V	
V <sub>OH</sub>	HIGH output voltage	X <sub>OUT</sub>	HIGH POWER	I <sub>OH</sub> = -1 mA	3.0		V	
			LOW POWER	I <sub>OH</sub> = -0.5 mA	3.0			
V <sub>OL</sub>	LOW output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>3</sub> , P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>2</sub> , P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>4</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , R, G, B, OUT1, OUT2	I <sub>OL</sub> = 5 mA			2.0	V	
V <sub>OL</sub>	LOW output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>3</sub>	I <sub>OL</sub> = 200 μA			0.45	V	
V <sub>OL</sub>	LOW output voltage	X <sub>OUT</sub>	HIGH POWER	I <sub>OL</sub> = 1 mA		2.0	V	
			LOW POWER	I <sub>OL</sub> = 0.5 mA		2.0		
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	$\overline{HOLD}$ , $\overline{RDY}$ , TB0 <sub>IN</sub> to TB2 <sub>IN</sub> , INT <sub>0</sub> , INT <sub>1</sub> , CTS <sub>0</sub> , CTS <sub>2</sub> , CLK <sub>0</sub> , CLK <sub>2</sub> , SCL1, SCL2, SDA1, SDA2, H <sub>SYNC</sub> , V <sub>SYNC</sub> , HC0, HC1, RxD <sub>0</sub>		0.2		0.8	V	
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	$\overline{RESET}$		0.2		1.8	V	
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	X <sub>IN</sub>		0.2		0.8	V	
I <sub>IH</sub>	HIGH input current	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>3</sub> , P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>2</sub> , P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>4</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>IN</sub> , $\overline{RESET}$ , CNV <sub>SS</sub> , BYTE, OSC1	V <sub>I</sub> = 5 V			5.0	μA	
I <sub>IL</sub>	LOW input current	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>3</sub> , P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>2</sub> , P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>4</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>IN</sub> , $\overline{RESET}$ , CNV <sub>SS</sub> , BYTE, OSC1	V <sub>I</sub> = 0 V			±5.0	μA	
I <sub>CC</sub>	Power supply current	In single-chip mode, the output pins are open and other pins are V <sub>SS</sub>	f(X <sub>IN</sub> ) = 10 MHz Square wave, no division	OSD ON, Data slicer ON	50	80	mA	
				OSD OFF, Data slicer OFF	30	50		
			f(X <sub>CIN</sub> ) = 32kHz Square wave, division by 8	OSD OFF, Data slicer OFF		10		μA
				When a WAIT instruction is executed		10		
				Ta = 25 °C when clock is stopped			10	
Ta = 70 °C when clock is stopped			200					
R <sub>BS</sub>	I <sup>2</sup> C-BUS • BUS switch connection resistor (between SCL1 and SCL2, SDA1 and SDA2)		V <sub>CC</sub> = 4.5 V			130	Ω	
R <sub>IXIN</sub>	Feedback resistor X <sub>IN</sub>				1.0		MΩ	
R <sub>IXCIN</sub>	Feedback resistor X <sub>CIN</sub>				6.0		MΩ	



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**5.4 A-D Conversion Characteristics**

**Table 5.4.1 A-D conversion characteristics (referenced to  $V_{CC} = AV_{CC} = 5V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_a = 25\text{ }^\circ\text{C}$ ,  $f(X_{IN}) = 10\text{ MHz}$  unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
—	Resolution		$V_{REF} = V_{CC}$			8	Bits
—	Absolute accuracy	Sample & hold function not available	$V_{REF} = V_{CC} = 5V$			$\pm 5$	LSB
		Sample & hold function available (8 bit)	$V_{REF} = V_{CC} = 5V$			$\pm 5$	LSB
$R_{LADDER}$	Ladder resistance		$V_{REF} = V_{CC}$	10		40	$k\Omega$
$t_{CONV}$	Conversion time			2.8			$\mu s$
$t_{SAMP}$	Sampling time			0.3			$\mu s$
$V_{REF}$	Reference voltage				$V_{CC}$		V
$V_{IA}$	Analog input voltage			0		$V_{CC}$	V

**5.5 D-A Conversion Characteristics**

**Table 5.5.1 D-A conversion characteristics (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_a = 25\text{ }^\circ\text{C}$ ,  $f(X_{IN}) = 10\text{ MHz}$  unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
—	Resolution					8	Bits
—	Absolute accuracy					10	%
$t_{su}$	Setup time					3	$\mu s$
$R_O$	Output resistance			4	10	20	$k\Omega$
$I_{VREF}$	Reference power supply input current		(Note)			1.5	mA

**Note:** This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "0016."  
The A-D converter's ladder resistance is not included.  
Also, when the  $V_{ref}$  is unconnected at the A-D control register,  $I_{VREF}$  is sent.

**5.6 Analog R, G, B Output Characteristics**

**Table 5.6.1 Analog R, G, B output characteristics ( $V_{CC} = 5V \pm 10\%$ ,  $V_{SS} = 0V$ ,  $f(X_{IN}) = 10\text{ MHz}$ ,  $T_a = -10\text{ }^\circ\text{C}$  to  $70\text{ }^\circ\text{C}$ , unless otherwise specified)**

Symbol	Parameter		Test conditions	Standard		Unit
				Min.	Max.	
$R_O$	Output impedance		$V_{CC} = 4.5V$		2	$k\Omega$
$V_{OE}$	Output deviation		$V_{CC} = 5.5V$		$\pm 0.5$	V
$T_{ST}$	Settling time		$V_{CC} = 4.5V$ , load capacity of 10 pF, load resistance of 20 $k\Omega$ , 70 % DC level		50	ns

## 5.7 Timing Requirements

**Table 5.7.1 External clock input (referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C unless otherwise specified)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External clock input cycle time	100		ns
tw(H)	External clock input HIGH pulse width	40		ns
tw(L)	External clock input LOW pulse width	40		ns
tr	External clock rise time		15	ns
tf	External clock fall time		15	ns

**Table 5.7.2 Memory expansion and microprocessor modes (referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C unless otherwise specified)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tac1(RD-DB)	Data input access time (no wait)		(Note)	ns
tac2(RD-DB)	Data input access time (with wait)		(Note)	ns
tac3(RD-DB)	Data input access time (when accessing multiplex bus area)		(Note)	ns
tsu(DB-RD)	Data input setup time	40		ns
tsu(RDY-BCLK)	$\overline{\text{RDY}}$ input setup time	30		ns
tsu(HOLD-BCLK)	$\overline{\text{HOLD}}$ input setup time	40		ns
th(RD-DB)	Data input hold time	0		ns
th(BCLK-RDY)	$\overline{\text{RDY}}$ input hold time	0		ns
th(BCLK-HOLD)	$\overline{\text{HOLD}}$ input hold time	0		ns
td(BCLK-HLDA)	$\overline{\text{HLDA}}$ output delay time		40	ns

**Note:** Calculated according to the BCLK frequency as follows:

$$t_{ac1}(\text{RD} - \text{DB}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 45 \quad [\text{ns}]$$

$$t_{ac2}(\text{RD} - \text{DB}) = \frac{3 \times 10^9}{f(\text{BCLK}) \times 2} - 45 \quad [\text{ns}]$$

$$t_{ac3}(\text{RD} - \text{DB}) = \frac{3 \times 10^9}{f(\text{BCLK}) \times 2} - 45 \quad [\text{ns}]$$

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 5.7.3 Timer B input (counter input in event counter mode)**

(referenced to  $V_{CC} = 5\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_a = 25\text{ }^\circ\text{C}$  unless otherwise specified)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIn input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIn input HIGH pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIn input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIn input HIGH pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width (counted on both edges)	80		ns

**Table 5.7.4 Timer B input (pulse period measurement mode)**

(referenced to  $V_{CC} = 5\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_a = 25\text{ }^\circ\text{C}$  unless otherwise specified)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIn input cycle time	400		ns
$t_{w(TBH)}$	TBiIn input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width	200		ns

**Table 5.7.5 Timer B input (pulse width measurement mode)**

(referenced to  $V_{CC} = 5\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_a = 25\text{ }^\circ\text{C}$  unless otherwise specified)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIn input cycle time	400		ns
$t_{w(TBH)}$	TBiIn input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIn input LOW pulse width	200		ns

**Table 5.7.6 Serial I/O (referenced to  $V_{CC} = 5\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_a = 25\text{ }^\circ\text{C}$  unless otherwise specified)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

**Table 5.7.7 External interrupt  $\overline{INT}_i$  inputs (referenced to  $V_{CC} = 5\text{ V}$ ,  $V_{SS} = 0\text{ V}$  at  $T_a = 25\text{ }^\circ\text{C}$  unless otherwise specified)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT}_i$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{INT}_i$ input LOW pulse width	250		ns

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**5.8 Switching Characteristics**

**Table 5.8.1 Memory expansion mode and microprocessor mode (no wait) (referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C, CM15 = "1" unless otherwise specified)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 5.9.1		25	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		4		ns
th(RD-AD)	Address output hold time (RD standard)		0		ns
th(WR-AD)	Address output hold time (WR standard)		0		ns
td(BCLK-CS)	Chip select output delay time			25	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		4		ns
td(BCLK-ALE)	ALE signal output delay time			25	ns
th(BCLK-ALE)	ALE signal output hold time		-4		ns
td(BCLK-RD)	RD signal output delay time			25	ns
th(BCLK-RD)	RD signal output hold time		0		ns
td(BCLK-WR)	WR signal output delay time			25	ns
th(BCLK-WR)	WR signal output hold time		0		ns
td(BCLK-DB)	Data output delay time (BCLK standard)			40	ns
th(BCLK-DB)	Data output hold time (BCLK standard)		4		ns
td(DB-WR)	Data output delay time (WR standard)		(Note 1)		ns
th(WR-DB)	Data output hold time (WR standard)(Note 2)		0		ns

**Note 1:** Calculated according to the BCLK frequency as follows:

$$td(DB - WR) = \frac{10^9}{f(BCLK)} - 40 \quad [ns]$$

**2:** This is standard value shows the timing when the output is off, and does not show hold time of data bus. Hold time of data bus is different by capacitor volume and pull-up (pull-down) resistance value.

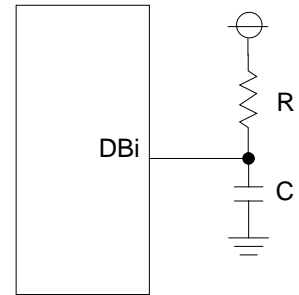
Hold time of data bus is expressed in

$$t = -CR \times \ln(1 - VOL / VCC)$$

by a circuit of the right figure.

For example, when VOL = 0.2VCC, C = 30 pF, R = 1 kΩ, hold time of output "L" level is

$$t = -30 \text{ pF} \times 1\text{k}\Omega \times \ln(1 - 0.2VCC / VCC) = 6.7 \text{ ns.}$$



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 5.8.2 Memory expansion mode and microprocessor mode**

(with wait, accessing external memory)

(referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C, CM15 = "1" unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 5.9.1		25	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		4		ns
th(RD-AD)	Address output hold time (RD standard)		0		ns
th(WR-AD)	Address output hold time (WR standard)		0		ns
td(BCLK-CS)	Chip select output delay time			25	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		4		ns
td(BCLK-ALE)	ALE signal output delay time			25	ns
th(BCLK-ALE)	ALE signal output hold time		- 4		ns
td(BCLK-RD)	RD signal output delay time			25	ns
th(BCLK-RD)	RD signal output hold time		0		ns
td(BCLK-WR)	WR signal output delay time			25	ns
th(BCLK-WR)	WR signal output hold time		0		ns
td(BCLK-DB)	Data output delay time (BCLK standard)			40	ns
th(BCLK-DB)	Data output hold time (BCLK standard)		4		ns
td(DB-WR)	Data output delay time (WR standard)		(Note 1)		ns
th(WR-DB)	Data output hold time (WR standard)(Note 2)		0		ns

**Note 1:** Calculated according to the BCLK frequency as follows:

$$td(DB - WR) = \frac{10^9}{f(BCLK)} - 40 \quad [ns]$$

**2:** This is standard value shows the timing when the output is off, and does not show hold time of data bus. Hold time of data bus is different by capacitor volume and pull-up (pull-down) resistance value.

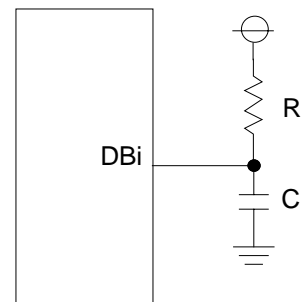
Hold time of data bus is expressed in

$$t = -CR \times \ln(1 - VOL / VCC)$$

by a circuit of the right figure.

For example, when VOL = 0.2Vcc, C = 30 pF, R = 1 kΩ, hold time of output "L" level is

$$t = -30 \text{ pF} \times 1\text{k}\Omega \times \ln(1 - 0.2Vcc / Vcc) = 6.7 \text{ ns.}$$



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Table 5.8.3 Memory expansion mode and microprocessor mode**  
(with wait, accessing external memory, multiplex bus area selected)  
(referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C, CM15 = “1” unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 5.9.1		25	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		4		ns
th(RD-AD)	Address output hold time (RD standard)		(Note)		ns
th(WR-AD)	Address output hold time (WR standard)		(Note)		ns
td(BCLK-CS)	Chip select output delay time			25	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		4		ns
th(RD-CS)	Chip select output hold time (RD standard)		(Note)		ns
th(WR-CS)	Chip select output hold time (WR standard)		(Note)		ns
td(BCLK-RD)	RD signal output delay time			25	ns
th(BCLK-RD)	RD signal output hold time		0		ns
td(BCLK-WR)	WR signal output delay time			25	ns
th(BCLK-WR)	WR signal output hold time		0		ns
td(BCLK-DB)	Data output delay time (BCLK standard)			40	ns
th(BCLK-DB)	Data output hold time (BCLK standard)		4		ns
td(DB-WR)	Data output delay time (WR standard)		(Note)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note)		ns
td(BCLK-ALE)	ALE signal output delay time (BCLK standard)			25	ns
th(BCLK-ALE)	ALE signal output hold time (BCLK standard)		- 4		ns
td(AD-ALE)	ALE signal output delay time (Address standard)		(Note)		ns
th(ALE-AD)	ALE signal output hold time (Address standard)		30		ns
td(AD-RD)	Post-address RD signal output delay time	0		ns	
td(AD-WR)	Post-address WR signal output delay time	0		ns	
tdZ(RD-AD)	Address output floating start time		8	ns	

**Note:** Calculated according to the BCLK frequency as follows:

$$th(RD - AD) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$th(RD - CS) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$td(DB - WR) = \frac{10^9 \times 3}{f(BCLK) \times 2} - 40 \quad [ns]$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$td(AD - ALE) = \frac{10^9}{f(BCLK) \times 2} - 25 \quad [ns]$$

### 5.9 Measurement Circuit

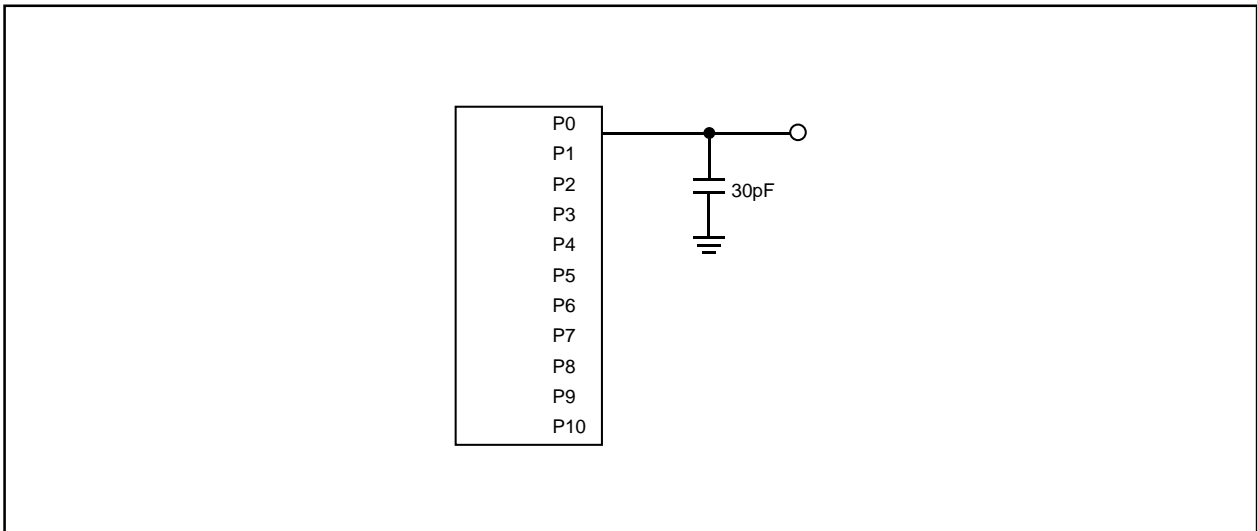


Figure 5.9.1 Port P0 to P10 measurement circuit

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**5.10 Timing Diagram**

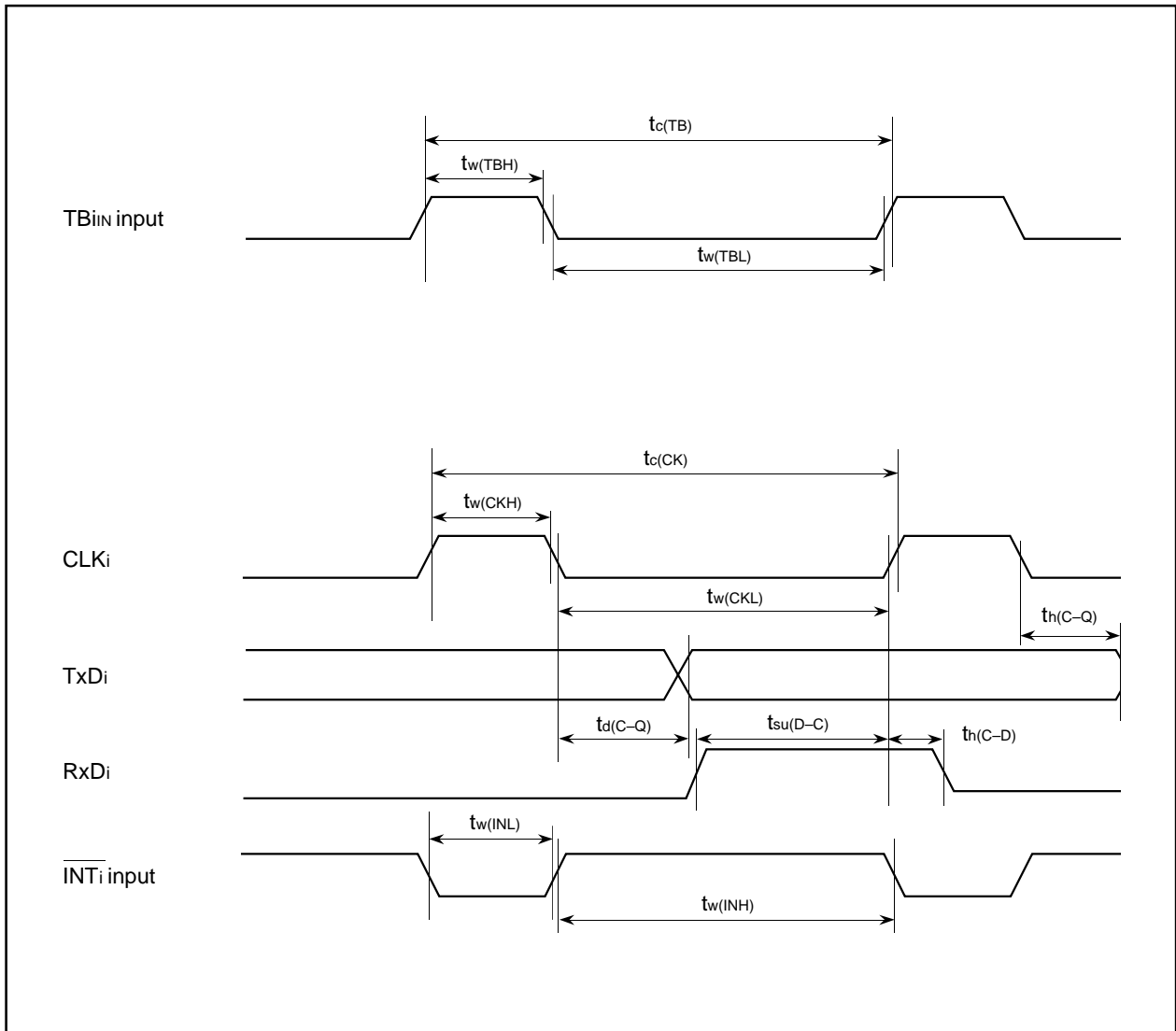


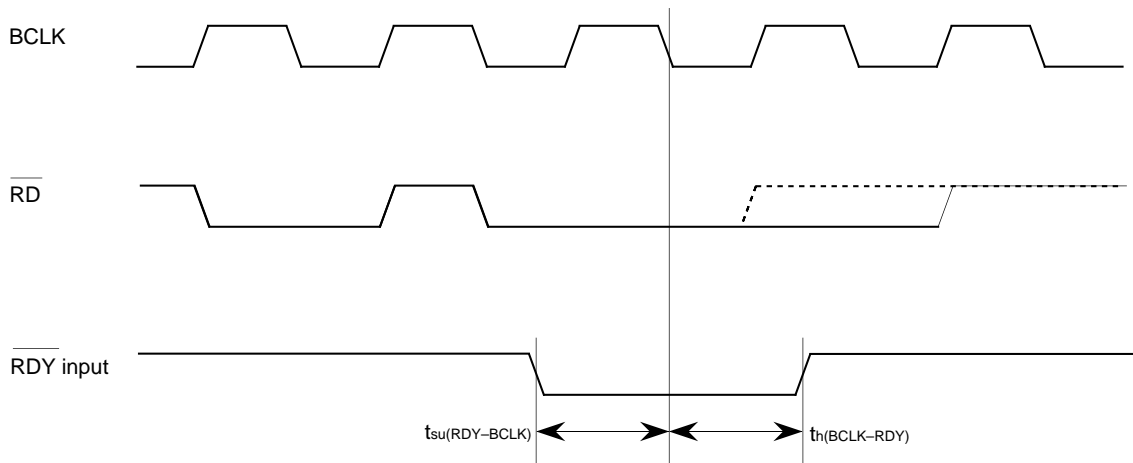
Figure 5.10.1 Timing diagram



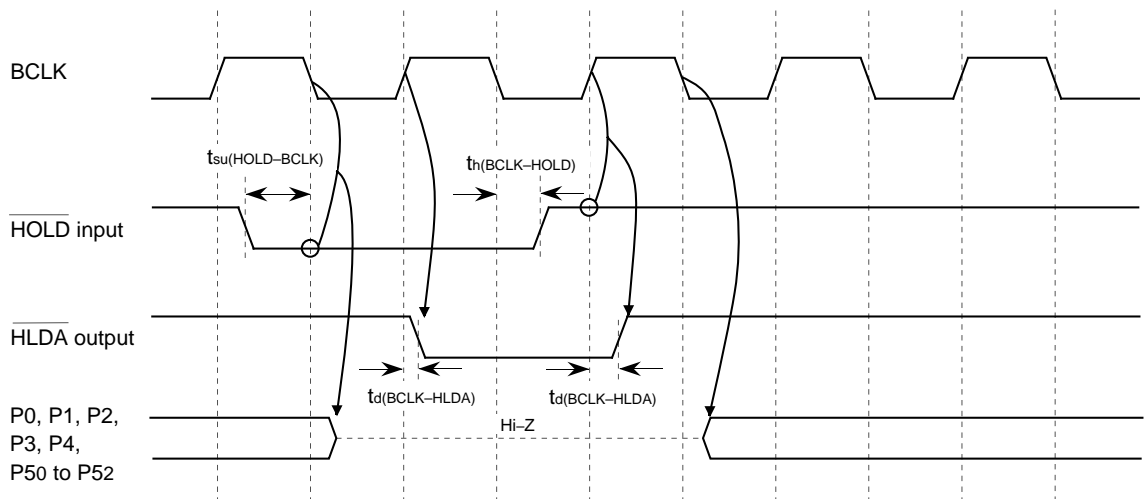
**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Memory Expansion Mode and Microprocessor Mode**  
(Valid only with wait)



(Valid with or without wait)



Note: The above pins are set to high-impedance regardless of the input level of the BYTE pin and bit (PM06) of processor mode register 0 selects the function of ports P40 to P43.

Measuring conditions

- $V_{CC}=5V$
- Input timing voltage : Determined with  $V_{IL}=1.0V, V_{IH}=4.0V$
- Output timing voltage : Determined with  $V_{OL}=2.5V, V_{OH}=2.5V$

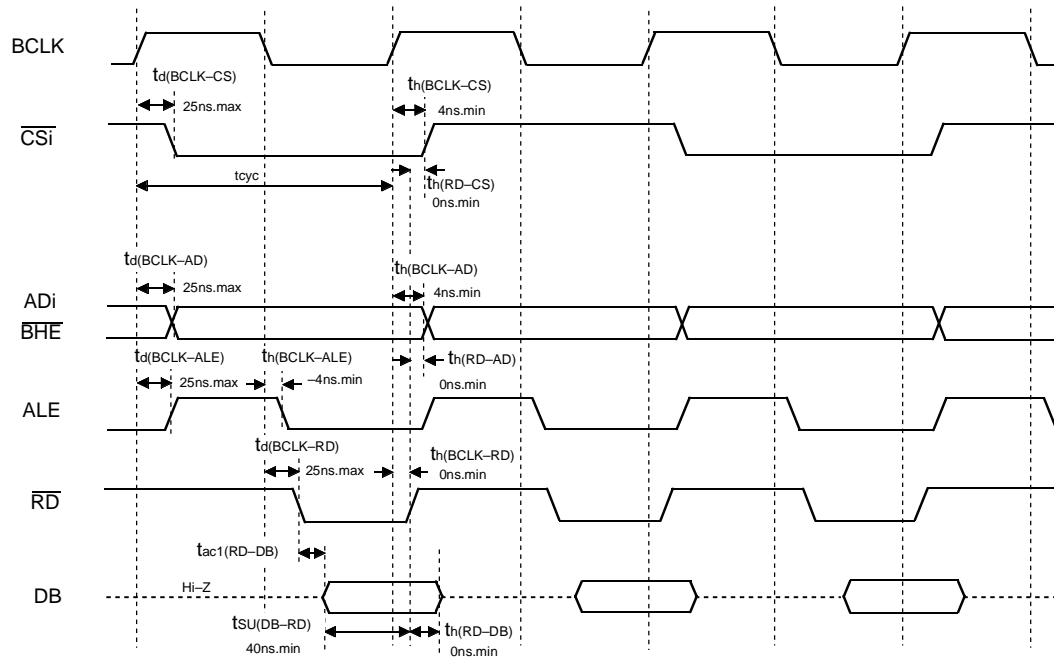
**Figure 5.10.2 Timing diagram in memory expansion mode and microprocessor mode (1)**

**M306V0ME-XXXFP**  
**M306V0EEFP**

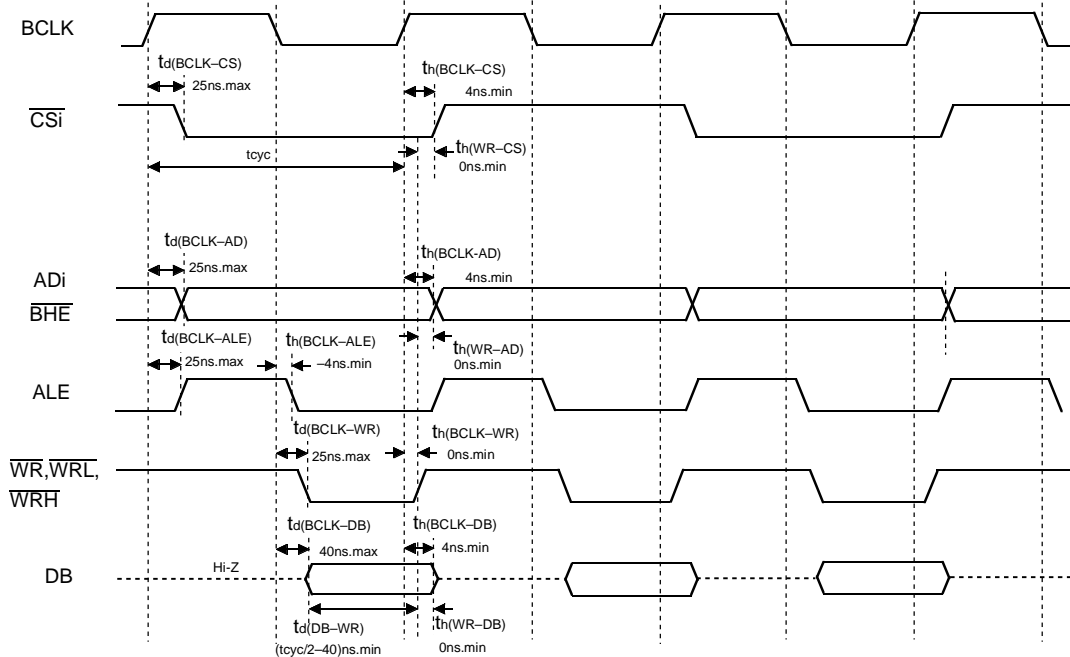
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Memory Expansion Mode and Microprocessor Mode**  
(With no wait)

Read timing



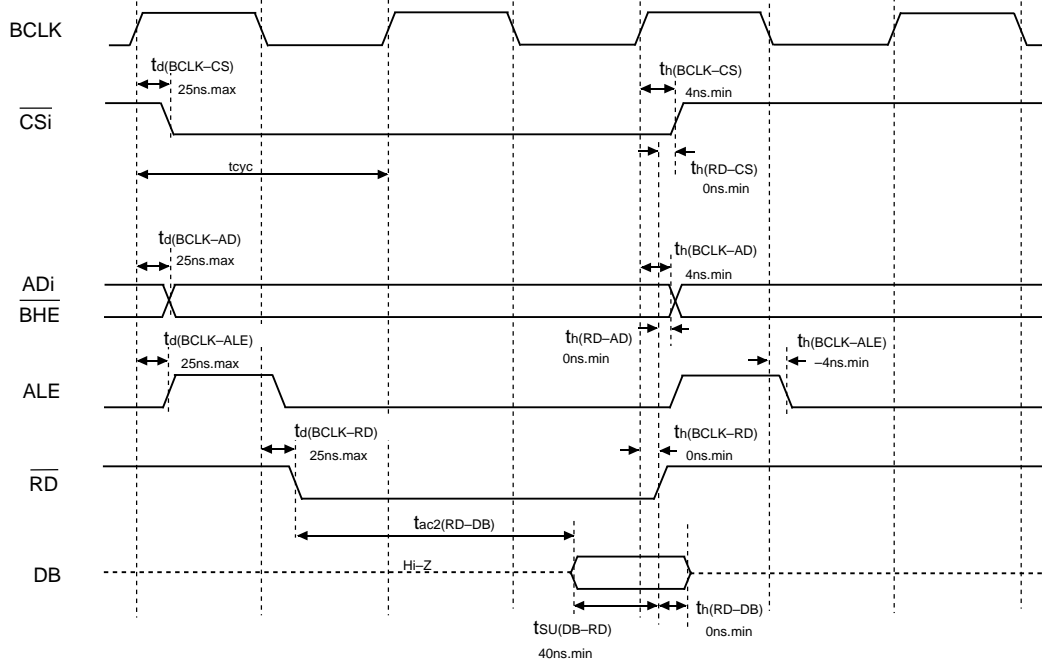
Write timing



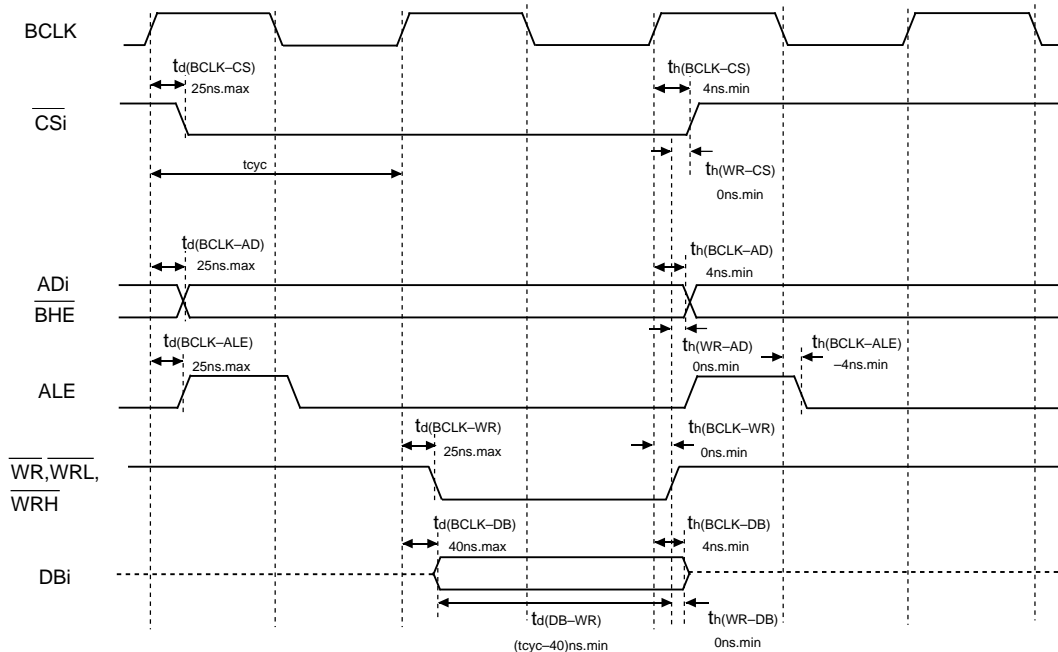
**Figure 5.10.3 Timing diagram in memory expansion mode and microprocessor mode (2)**

**Memory Expansion Mode and Microprocessor Mode**  
(When accessing external memory area with wait)

**Read timing**



**Write timing**



**Measuring conditions**

- VCC=5V
- Input timing voltage : Determined with:  $V_{IL}=0.8\text{V}$ ,  $V_{IH}=2.5\text{V}$
- Output timing voltage : Determined with:  $V_{OL}=0.8\text{V}$ ,  $V_{OH}=2.0\text{V}$

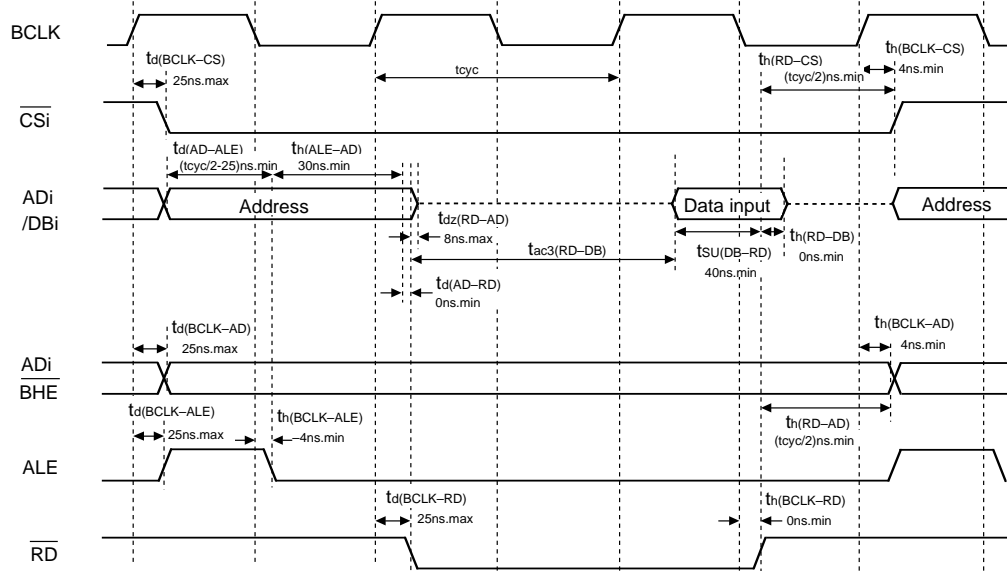
**Figure 5.10.4 Timing diagram in memory expansion mode and microprocessor mode (3)**

**M306V0ME-XXXFP**  
**M306V0EEFP**

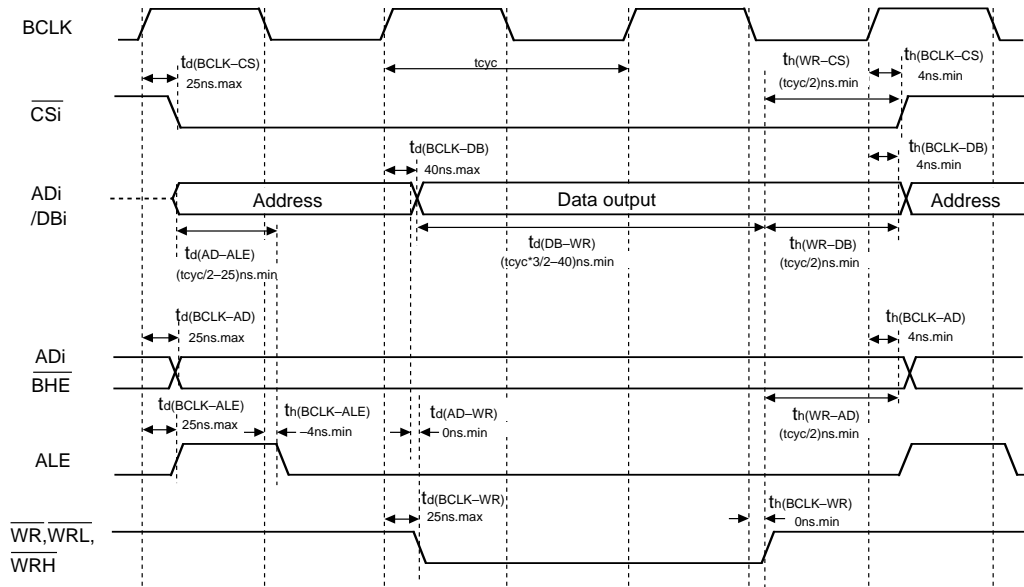
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Memory Expansion Mode and Microprocessor Mode**  
(When accessing external memory area with wait, and select multiplexed bus)

**Read timing**



**Write timing**



**Measuring conditions**

- $V_{CC}=5V$
- Input timing voltage : Determined with  $V_{IL}=0.8V$ ,  $V_{IH}=2.5V$
- Output timing voltage : Determined with  $V_{OL}=0.8V$ ,  $V_{OH}=2.0V$

**Figure 5.10.5 Timing diagram in memory expansion mode and microprocessor mode (4)**

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**6. MASK CONFIRMATION FORM**

GZZ—SH52—57B <82A0>

**MITSUBISHI ELECTRIC SINGLE-CHIP 16-BIT  
MICROCOMPUTER M306V0ME-XXXFP  
MASK ROM CONFIRMATION FORM**

Mask ROM number	
-----------------	--

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please complete all items marked ※.

※ Customer	Company name	TEL ( )	Issuance signature	Submitted by	Supervisor
	Date issued	Date :			

※ 1. Check sheet

Name the product you order, and choose which to give in, EPROMs or floppy disks.  
If you order by means of EPROMs, three sets of EPROMs are required per pattern. If you order by means of floppy disks, one floppy disk is required per pattern.

In the case of EPROMs

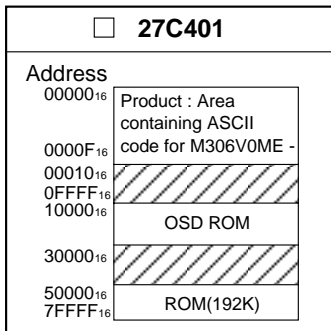
Mitsubishi will create the mask using the data on the EPROMs supplied, providing the data is the same on at least two of those sets. Mitsubishi will, therefore, only accept liability if there is any discrepancy between the data on the EPROM sets and the ROM data written to the product.  
Please carefully check the data on the EPROMs being submitted to Mitsubishi.

Checksum code for total EPROM area : 

--	--	--	--

 (hex)

EPROM type :



- (1) The area from 00000<sub>16</sub> to 0000F<sub>16</sub> is for storing data on the product type name.  
The ASCII code for 'M306V0ME-' is shown at right.  
The data in this table must be written to address 00000<sub>16</sub> to 0000F<sub>16</sub>.  
Both address and data are shown in hex.
- (2) Write "FF<sub>16</sub>" to the lined area.

<table style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">Address</td></tr> <tr><td style="padding: 2px;">00000<sub>16</sub> 'M' = 4D<sub>16</sub></td></tr> <tr><td style="padding: 2px;">00001<sub>16</sub> '3' = 33<sub>16</sub></td></tr> <tr><td style="padding: 2px;">00002<sub>16</sub> '0' = 30<sub>16</sub></td></tr> <tr><td style="padding: 2px;">00003<sub>16</sub> '6' = 36<sub>16</sub></td></tr> <tr><td style="padding: 2px;">00004<sub>16</sub> 'V' = 56<sub>16</sub></td></tr> <tr><td style="padding: 2px;">00005<sub>16</sub> '0' = 30<sub>16</sub></td></tr> <tr><td style="padding: 2px;">00006<sub>16</sub> 'M' = 4D<sub>16</sub></td></tr> <tr><td style="padding: 2px;">00007<sub>16</sub> 'E' = 45<sub>16</sub></td></tr> </table>	Address	00000 <sub>16</sub> 'M' = 4D <sub>16</sub>	00001 <sub>16</sub> '3' = 33 <sub>16</sub>	00002 <sub>16</sub> '0' = 30 <sub>16</sub>	00003 <sub>16</sub> '6' = 36 <sub>16</sub>	00004 <sub>16</sub> 'V' = 56 <sub>16</sub>	00005 <sub>16</sub> '0' = 30 <sub>16</sub>	00006 <sub>16</sub> 'M' = 4D <sub>16</sub>	00007 <sub>16</sub> 'E' = 45 <sub>16</sub>	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">Address</td></tr> <tr><td style="padding: 2px;">00008<sub>16</sub> '—' = 2D<sub>16</sub></td></tr> <tr><td style="padding: 2px;">00009<sub>16</sub> FF<sub>16</sub></td></tr> <tr><td style="padding: 2px;">0000A<sub>16</sub> FF<sub>16</sub></td></tr> <tr><td style="padding: 2px;">0000B<sub>16</sub> FF<sub>16</sub></td></tr> <tr><td style="padding: 2px;">0000C<sub>16</sub> FF<sub>16</sub></td></tr> <tr><td style="padding: 2px;">0000D<sub>16</sub> FF<sub>16</sub></td></tr> <tr><td style="padding: 2px;">0000E<sub>16</sub> FF<sub>16</sub></td></tr> <tr><td style="padding: 2px;">0000F<sub>16</sub> FF<sub>16</sub></td></tr> </table>	Address	00008 <sub>16</sub> '—' = 2D <sub>16</sub>	00009 <sub>16</sub> FF <sub>16</sub>	0000A <sub>16</sub> FF <sub>16</sub>	0000B <sub>16</sub> FF <sub>16</sub>	0000C <sub>16</sub> FF <sub>16</sub>	0000D <sub>16</sub> FF <sub>16</sub>	0000E <sub>16</sub> FF <sub>16</sub>	0000F <sub>16</sub> FF <sub>16</sub>
Address																			
00000 <sub>16</sub> 'M' = 4D <sub>16</sub>																			
00001 <sub>16</sub> '3' = 33 <sub>16</sub>																			
00002 <sub>16</sub> '0' = 30 <sub>16</sub>																			
00003 <sub>16</sub> '6' = 36 <sub>16</sub>																			
00004 <sub>16</sub> 'V' = 56 <sub>16</sub>																			
00005 <sub>16</sub> '0' = 30 <sub>16</sub>																			
00006 <sub>16</sub> 'M' = 4D <sub>16</sub>																			
00007 <sub>16</sub> 'E' = 45 <sub>16</sub>																			
Address																			
00008 <sub>16</sub> '—' = 2D <sub>16</sub>																			
00009 <sub>16</sub> FF <sub>16</sub>																			
0000A <sub>16</sub> FF <sub>16</sub>																			
0000B <sub>16</sub> FF <sub>16</sub>																			
0000C <sub>16</sub> FF <sub>16</sub>																			
0000D <sub>16</sub> FF <sub>16</sub>																			
0000E <sub>16</sub> FF <sub>16</sub>																			
0000F <sub>16</sub> FF <sub>16</sub>																			

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

GZZ—SH52—57B <82A0>

**MITSUBISHI ELECTRIC SINGLE-CHIP 16-BIT  
MICROCOMPUTER M306V0ME-XXXFP  
MASK ROM CONFIRMATION FORM**

Mask ROM number	
-----------------	--

(3) Be sure to store "FF 16" in the following test font addresses in OSD ROM.

When producing OSD ROM data with the OSD font editor program of Mitsubishi,  
"FF16" is set automatically to these test font addresses.

(All addresses below are shown in hex.)

1007E	1007F	1207E	1207F	1407E	1407F	20400	20401	20600	20601	213F8	213F9	213FC	213FD
1017E	1017F	1217E	1217F	1417E	1417F	21400	21401	21600	21601	223F8	223F9	223FC	223FD
1027E	1027F	1227E	1227F	1427E	1427F	22400	22401	22600	22601	233F8	233F9	233FC	233FD
1037E	1037F	1237E	1237F	1437E	1437F	23400	23401	23600	23601	243F8	243F9	243FC	243FD
1047E	1047F	1247E	1247F	1447E	1447F	24400	24401	24600	24601	253F8	253F9	253FC	253FD
1057E	1057F	1257E	1257F	1457E	1457F	25400	25401	25600	25601	263F8	263F9	263FC	263FD
1067E	1067F	1267E	1267F	1467E	1467F	26400	26401	26600	26601	273F8	273F9	273FC	273FD
1077E	1077F	1277E	1277F	1477E	1477F	27400	27401	27600	27601	283F8	283F9	283FC	283FD
1087E	1087F	1287E	1287F	1487E	1487F	28400	28401	28600	28601	293F8	293F9	293FC	293FD
1097E	1097F	1297E	1297F	1497E	1497F	29400	29401	29600	29601	2A3F8	2A3F9	2A3FC	2A3FD
10A7E	10A7F	12A7E	12A7F	14A7E	14A7F	2A400	2A401	2A600	2A601	21BF8	21BF9	21BFC	21BFD
10B7E	10B7F	12B7E	12B7F	14B7E	14B7F	2B400	2B401	2B600	2B601	22BF8	22BF9	22BFC	22BFD
10C7E	10C7F	12C7E	12C7F	14C7E	14C7F	2C400	2C401	2C600	2C601	23BF8	23BF9	23BFC	23BFD
10D7E	10D7F	12D7E	12D7F	14D7E	14D7F	2D400	2D401	2D600	2D601	24BF8	24BF9	24BFC	24BFD
10E7E	10E7F	12E7E	12E7F	14E7E	14E7F	2E400	2E401	2E600	2E601	25BF8	25BF9	25BFC	25BFD
10F7E	10F7F	12F7E	12F7F	14F7E	14F7F	2F400	2F401	2F600	2F601	26BF8	26BF9	26BFC	26BFD
1107E	1107F	1307E	1307F	1507E	1507F	20C00	20C01	20E00	20E01	27BF8	27BF9	27BFC	27BFD
1117E	1117F	1317E	1317F	1517E	1517F	21C00	21C01	21E00	21E01	28BF8	28BF9	28BFC	28BFD
1127E	1127F	1327E	1327F	1527E	1527F	22C00	22C01	22E00	22E01	29BF8	29BF9	29BFC	29BFD
1137E	1137F	1337E	1337F	1537E	1537F	23C00	23C01	23E00	23E01	2ABF8	2ABF9	2ABFC	2ABFD
1147E	1147F	1347E	1347F	1547E	1547F	24C00	24C01	24E00	24E01				
1157E	1157F	1357E	1357F	1557E	1557F	25C00	25C01	25E00	25E01				
1167E	1167F	1367E	1367F	1567E	1567F	26C00	26C01	26E00	26E01				
1177E	1177F	1377E	1377F	1577E	1577F	27C00	27C01	27E00	27E01				
1187E	1187F	1387E	1387F	1587E	1587F	28C00	28C01	28E00	28E01				
1197E	1197F	1397E	1397F	1597E	1597F	29C00	29C01	29E00	29E01				
						2AC00	2AC01	2AE00	2AE01				
10082	10083	12082	12083	14082	14083	2BC00	2BC01	2BE00	2BE01				
10182	10183	12182	12183	14182	14183	2CC00	2CC01	2CE00	2CE01				
10282	10283	12282	12283	14282	14283	2DC00	2DC01	2DE00	2DE01				
10382	10383	12382	12383	14382	14383	2EC00	2EC01	2EE00	2EE01				
10482	10483	12482	12483	14482	14483	2FC00	2FC01	2FE00	2FE01				
10582	10583	12582	12583	14582	14583	10000	10800	10001	10801				
10682	10683	12682	12683	14682	14683	11000	11800	11001	11801				
10782	10783	12782	12783	14782	14783	12000	12800	12001	12801				
10882	10883	12882	12883	14882	14883	13000	13800	13001	13801				
10982	10983	12982	12983	14982	14983	14000	14800	14001	14801				
10A82	10A83	12A82	12A83	14A82	14A83	15000	15800	15001	15801				
10B82	10B83	12B82	12B83	14B82	14B83	16000	16800	16001	16801				
10C82	10C83	12C82	12C83	14C82	14C83	17000	17800	17001	17801				
10D82	10D83	12D82	12D83	14D82	14D83	18000	18800	18001	18801				
10E82	10E83	12E82	12E83	14E82	14E83	19000	19800	19001	19801				
10F82	10F83	12F82	12F83	14F82	14F83	1A000	1A800	1A001	1A801				
11082	11083	13082	13083	15082	15083	1B000	1B800	1B001	1B801				
11182	11183	13182	13183	15182	15183	1C000	1C800	1C001	1C801				
11282	11283	13282	13283	15282	15283	1D000	1D800	1D001	1D801				
11382	11383	13382	13383	15382	15383	1E000	1E800	1E001	1E801				
11482	11483	13482	13483	15482	15483	1F000	1F800	1F001	1F801				
11582	11583	13582	13583	15582	15583								
11682	11683	13682	13683	15682	15683								
11782	11783	13782	13783	15782	15783								
11882	11883	13882	13883	15882	15883								
11982	11983	13982	13983	15982	15983								

(2/4)

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

GZZ—SH52—57B <82A0>

**MITSUBISHI ELECTRIC SINGLE-CHIP 16-BIT  
MICROCOMPUTER M306V0ME-XXXFP  
MASK ROM CONFIRMATION FORM**

Mask ROM number	
-----------------	--

The ASCII code for the type No. can be written to EPROM addresses 0000<sub>16</sub> to 0000F<sub>16</sub> by specifying the pseudo-instructions shown in the following table at the beginning of the assembler source program.

EPROM type	27C401
Code entered in source program	△.SECTION △ASCII CODE, ROM DATA △.ORG △080000H △.BYTE △' M306V0ME- '

Note: The ROM cannot be processed if the type No. written to the EPROM does not match the type No. in the check sheet.

In the case of floppy disks

Mitsubishi processes the mask files generated by the mask file generation utilities out of those held on the floppy disks you give in to us, and forms them into masks. Hence, we assume liability provided that there is any discrepancy between the contents of these mask files and the ROM data to be burned into products we produce. Check thoroughly the contents of the mask files you give in.  
Prepare 3.5 inches 2HD(IBM format) floppy disks. And store only one mask file in a floppy disk.

File code : 

--	--	--	--	--	--	--	--

 (hex)

Mask file name : 

--	--	--	--	--	--	--	--

 .MSK (alpha-numeric 8-digit)

Note: When using the floppy disks, do not store the type No. to addresses 0000<sub>16</sub> to 0000F<sub>16</sub>.

※2. Mark specification

The mark specification differs according to the type of package. After entering the mark specification on the separate mark specification sheet (for each package), attach that sheet to this masking check sheet for submission to Mitsubishi.

For the M306V0ME-XXXFP, submit the 100P6S mark specification sheet.

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

GZZ—SH52—57B <82A0>

**MITSUBISHI ELECTRIC SINGLE-CHIP 16-BIT  
MICROCOMPUTER M306V0ME-XXXFP  
MASK ROM CONFIRMATION FORM**

Mask ROM number	
-----------------	--

※ 3. Usage Conditions

For our reference when of testing our products, please reply to the following questions about the usage of the products you ordered.

(1) Which kind of X<sub>IN</sub>-X<sub>OUT</sub> oscillation circuit is used?

- Ceramic resonator       Quartz-crystal oscillator  
 External clock input       Other (                    )

What frequency do you use?

f(X<sub>IN</sub>) =  MHz

(2) Which kind of X<sub>CIN</sub>-X<sub>COU</sub>T oscillation circuit is used?

- Ceramic resonator       Quartz-crystal oscillator  
 External clock input       Other (                    )

What frequency do you use?

f(X<sub>CIN</sub>) =  kHz

(3) Which operation mode do you use?

- Single-chip mode                      Memory expansion mode  
Microprocessor mode

Thank you cooperation.

※ 4. Special item (Indicate none if there is no specified item)

(4/4)



**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

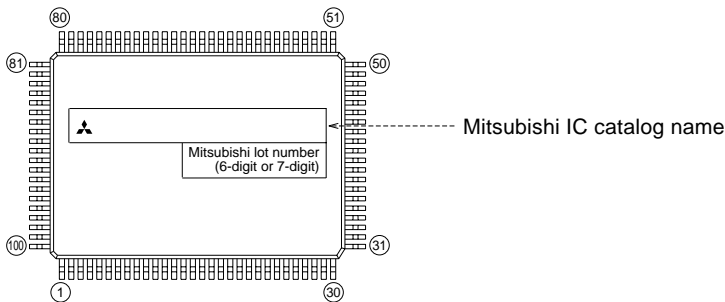
**7. MARK SPECIFICATION FORM**

**100P6S (100-PIN QFP) MARK SPECIFICATION FORM**

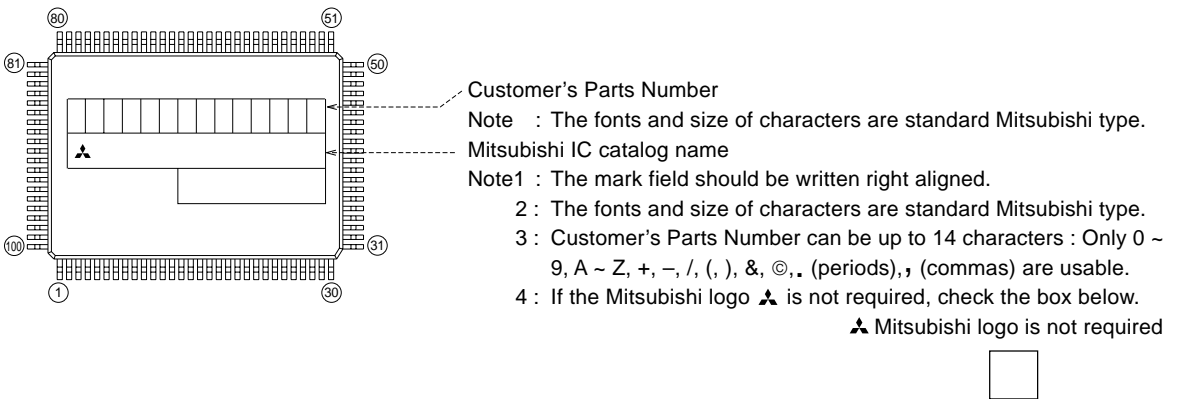
Mitsubishi IC catalog name

Please choose one of the marking types below (A, B, C), and enter the Mitsubishi catalog name and the special mark (if needed).

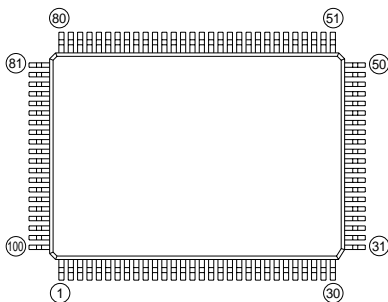
**A. Standard Mitsubishi Mark**



**B. Customer's Parts Number + Mitsubishi catalog name**



**C. Special Mark Required**



Note1 : If the Special Mark is to be Printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible.  
Mitsubishi lot number (6-digit or 7-digit) and Mask ROM number (3-digit) are always marked.

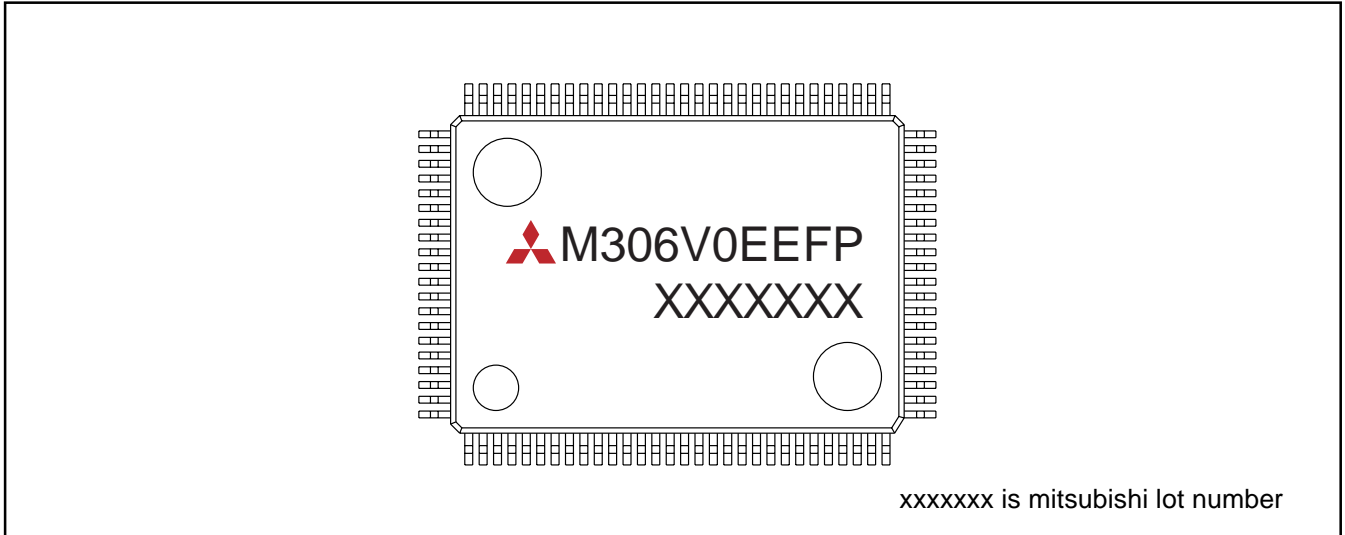
2 : If the customer's trade mark logo must be used in the Special Mark, check the box below.  
Please submit a clean original of the logo.  
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

Special logo required

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**8. ONE TIME PROM VERSION M306V0EEFP MARKING**



**M306V0ME-XXXFP**  
**M306V0EEFP**

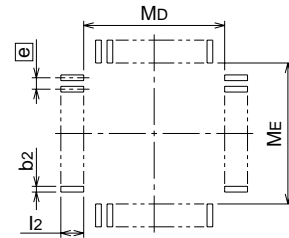
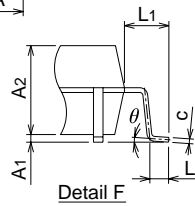
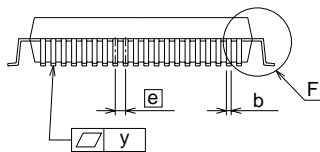
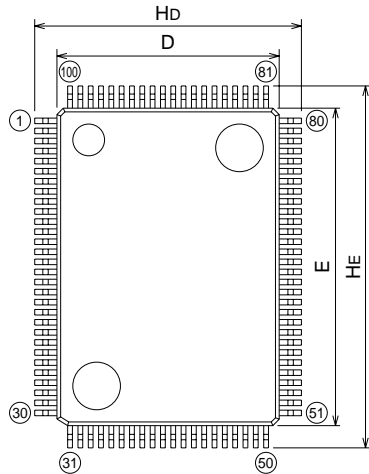
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**9. PACKAGE OUTLINE**

**100P6S-A**

Plastic 100pin 14X20mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP100-P-1420-0.65	-	1.58	Alloy 42



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	3.05
A1	0	0.1	0.2
A2	-	2.8	-
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	-	0.65	-
HD	16.5	16.8	17.1
HE	22.5	22.8	23.1
L	0.4	0.6	0.8
L1	-	1.4	-
y	-	-	0.1
$\theta$	0°	-	10°
b2	-	0.35	-
l2	1.3	-	-
MD	-	14.6	-
ME	-	20.6	-

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**Structure of Register**

Refer to the figure below as for each register.

<Example>

Processor mode register 1 (Note)

**Values immediately after reset release** (Note 1)

When reset: 0000XX0z

**Bit attributes** (Note 2)

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

Symbol: PM1  
Address: 0005<sub>16</sub>

Bit symbol	Bit name	Function	R	W
	Reserved bit	Must always be set to "0"	○	○
	Nothing is assigned. In an attempt to write to these bits, write "0." The value, if read, turns out to be indeterminate.		—	—
	Reserved bits	Must always be set to "0"	—	○
PM17	Wait bit	0 : No wait state 1 : Wait state inserted	○	○

**Note:** Set bit 1 of the protect register (address 000A<sub>16</sub>) to "1" when writing new values to this register.

X : Bit in which nothing is assigned

**Notes 1:** Values immediately after reset release

0 ..... "0" after reset release

1 ..... "1" after reset release

? ..... Indeterminate after reset release

X ..... Bit in which nothing is assigned

**2:** Bit attributes ..... The attributes of control register bits are classified into 3 types : read-only, write-only and read and write. In the figure, these attributes are represented as follows :

**R** ..... Read

○ ..... Read enabled

X ..... Read disabled

— ..... Bit in which nothing is assigned (The value, if read, turns out to be indeterminate.)

**W** ..... Write

○ ..... Write enabled

X ..... Write disabled

— ..... Bit in which nothing is assigned

## -----Register Index-----

<b>[A]</b>	
A-D conversion interrupt control register (ADIC) .....	54
Address match interrupt enable register (AIER) .....	64
Address match interrupt register i (RMADi) ..	64
A-D register i (ADi) .....	136
A-D control register 2 (ADCON2) .....	136
A-D control register 1 (ADCON1) .....	135, 138, 140, 142, 144, 146
A-D control register 0 (ADCON0) .....	135, 138, 140, 142, 144, 145
<b>[B]</b>	
Block control register i (BCi) .....	165
Bottom border control register (BBR) .....	206
Bus collision detection interrupt control register (BCNIC) .....	54
<b>[C]</b>	
Caption position register (CPS) .....	156
Chip select control register (CSR) .....	30
Clock control register (CS) .....	173
Clock prescaler reset flag (CPSRF) .....	92
Clock run-in detect register (CRD) .....	157
Color palette register i (CRi) .....	192
Count start flag (TABSR) .....	92
<b>[D]</b>	
D-A control register (DACON) .....	149
D-A register i (DAi) .....	149
Data clock position register (DPS) .....	158
Data slicer control register 1 (DSC1) .....	152
Data slicer control register 2 (DSC2) .....	152
Data slicer interrupt control register (DSIC) ..	54
Data slicer reserved register i (DRi) .....	159
DMA0 request cause select register (DM0SL) .....	71
DMA1 request cause select register (DM1SL) ..	72
DMAi control register (DMiCON) .....	72
DMAi interrupt control register (DMiIC) .....	54
DMAi destination pointer (DARi) .....	73
DMAi transfer counter (TCRi) .....	73
DMAi source pointer (SARi) .....	73
<b>[H]</b>	
Horizontal position register .....	170
H <sub>SYNC</sub> counter register (HC) .....	160
<b>[I]</b>	
I/O polarity control register (PC) .....	174
Interrupt control reserved register i (REiIC) .....	63
INTi interrupt control register (INTiIC) .....	54
<b>[L]</b>	
Left border control register (LBR) .....	207
<b>[O]</b>	
One-shot start flag (ONSF) .....	83
OSD control register 1 (OC1) .....	164
OSD control register 2 (OC2) .....	167
OSD control register 3 (OC3) .....	192
OSD control register 4 (OR4) .....	176
OSD reserved register i (ORi) .....	211
OSDi interrupt control register (OSDiIC) .....	54
<b>[P]</b>	
Peripheral mode register (PM) .....	98
Port control register (PCR) .....	222
Port P0 to P5, P7, P10 register (P0 to P5, P7, P10) .....	220
Port P0 to P5, P7, P10 direction register (PD0 to PD5, PD7, PD10) .....	218
Port P6 register (P6) .....	220
Port P6 direction register (PD6) .....	218
Port P8 register (P8) .....	221
Port P9 register (P9) .....	221
Port P8 direction register (PD8) .....	219
Port P9 direction register (PD9) .....	219
Processor mode register 0 (PM0) .....	24
Processor mode register 1 (PM1) .....	25

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

Protect register (PRCR) .....	46	.....	106
Pull-up control register 0 (PUR0) .....	222	UART2 transmit/receive control register 1 (U2C1)	
Pull-up control register 1 (PUR1) .....	223	.....	107
Pull-up control register 2 (PUR2) .....	223	UARTi bit rate generator (UiBRG) .....	103

**[R]**

Raster color register (RSC) .....	208	UARTi receive buffer register (UiRB) .....	103
Reserved register i (INVCi) .....	97	UARTi receive interrupt control register (SiRIC)	
Right border control register (RBR) .....	207	.....	54
		UARTi transmit buffer register (UiTB) .....	103
		UARTi transmit interrupt control register (SiTIC)	
		.....	54
		Up/down flag (UDF) .....	83

**[S]**

SPRITE horizontal position register (HS) ....	203
SPRITE OSD control register (SC) .....	202
SPRITE vertical position register i (VSi) ....	203
System clock control register 0 (CM0) .....	40
System clock control register 1 (CM1) .....	40

**[V]**

V <sub>SYNC</sub> interrupt control register (VSYNCIC) .....	54
Vertical position register i (VPi) .....	170

**[W]**

Watchdog timer control register (WDC) .....	68
Watchdog timer start register (WDTS) .....	68

**[T]**

Timer Ai interrupt control register (TAiIC)	
.....	54
Timer Bi interrupt control register (TBiIC)	
.....	54
Timer Ai register (TAi) .....	82
Timer Bi register (TBi) .....	92
Timer Ai mode register (TAiMR)	
.....	81, 85, 87 to 89
Timer Bi mode register (TBiMR)	
.....	91, 93 to 95
Top border control register (TBR) .....	206
Trigger select register (TRGSR) .....	84

**[U]**

UART transmit/receive control register 2 (U2CON)	
.....	108
UART0 transmit/receive control register 0 (U0C0)	
.....	105
UART0 transmit/receive control register 1 (U0C1)	
.....	107
UART0 transmit/receive mode register (U0MR)	
.....	104, 111, 118
UART2 special mode register (U2SMR) ...	108, 128
UART2 transmit/receive mode register (U2MR)	
.....	104, 111, 118
UART2 transmit/receive control register 0 (U2C0)	

**M306V0ME-XXXFP**  
**M306V0EEFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Keep safety first in your circuit designs!**

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

**Notes regarding these materials**

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
- The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors. Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (<http://www.mitsubishichips.com>).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
- Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.



REVISION HISTORY

M306V0ME-XXXFP, M306V0EEFP (REV1.1) DATA SHEET

Rev. No.	Revision Description	Rev. date
1.0	First Edition of PDF File	9910
1.1	<ul style="list-style-type: none"> <li>• Explanations for OSDP mode are deleted (Figure 2.16.4)</li> <li>• Add address (Figure 2.16.4)</li> <li>• Correct the last sentence; •• overlaps with the <u>display</u> which includes••• (P201, "2.16.13")</li> <li>• Correct note 2; "HS1" and "HS2" to "HS" (P201)</li> </ul>	0004