

To all our customers

---

## **Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.**

---

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.  
Customer Support Dept.  
April 1, 2003



**M306V5ME-XXXSP**  
**M306V5EESP**

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

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

1. DESCRIPTION .....	1	2.16.18 Scan Mode .....	219
1.1 Features .....	1	2.16.19 R, G, B Signal Output Control ...	219
1.2 Applications .....	1	2.16.20 OSD Reserved Register .....	220
1.3 Pin Configuration .....	3	2.17 Programmable I/O Ports .....	221
1.4 Block Diagram .....	4		
1.5 Performance Outline .....	5	3. USAGE PRECAUTION .....	239
2. OPERATION OF FUNCTIONAL BLOKS .....	10	3.1 Timer A (timer mode) .....	239
2.1 Memory .....	10	3.2 Timer A (event counter mode) .....	239
2.2 Central Processing Unit (CPU) .....	16	3.3 Timer A (one-shot timer mode) .....	239
2.3 Reset .....	19	3.4 Timer A (pulse width modulation mode)	239
2.4 Single-chip Mode .....	23	3.5 Timer B (timer mode, event counter mode)	240
2.5 Clock Generating Circuit .....	27	3.6 Timer B (pulse period, pulse width	
2.6 Protection .....	35	measurement mode) .....	240
2.7 Interrupts .....	36	3.7 A-D Converter .....	240
2.8 Watchdog Timer .....	56	3.8 Stop Mode and Wait Mode .....	240
2.9 DMAC .....	58	3.9 Interrupts .....	241
2.10 Timer .....	68	3.10 Built-in PROM Version .....	242
2.11 Serial I/O .....	88	4. ITEMS TO BE SUBMITTED WHEN ORDERING	
2.12 A-D Converter .....	138	MASKED ROM VERSION .....	243
2.13 D-A Converter .....	153	5. ELECTRICAL CHARACTERISTICS .....	244
2.14 Data Slicer .....	155	5.1. Absolute Maximum Ratings .....	244
2.15 HSYNC Counter .....	165	5.2 Recommended Operating Conditions ..	245
2.16 OSD Functions .....	166	5.3 Electrical Characteristics .....	246
2.16.1 Triple Layer OSD .....	172	5.4 A-D Conversion Characteristics .....	247
2.16.2 Display Position .....	174	5.5 D-A Conversion Characteristics .....	247
2.16.3 Dot Size .....	178	5.6 Analog R, G, B Output Characteristics	247
2.16.4 Clock for OSD .....	179	5.7 Timing Requirements .....	248
2.16.5 Field Determination Display .....	180	5.8 Timing Diagram .....	250
2.16.6 Memory for OSD .....	182		
2.16.7 Character Color .....	195	6. MASK CONFIRMATION FORM .....	251
2.16.8 Character Background Color .....	195		
2.16.9 OUT1, OUT2 Signals .....	200	7. MARK SPECIFICATION FORM .....	255
2.16.10 Attribute .....	201		
2.16.11 Automatic Solid Space Function .....	206	8. ONE TIME PROM VERSION M306V5EESP	
2.16.12 Particular OSD Mode Block .....	207	MARKING .....	256
2.16.13 Multiline Display .....	209		
2.16.14 SPRITE OSD Function .....	210	9. PACKAGE OUTLINE .....	257
2.16.15 Window Function .....	213		
2.16.16 Blank Function .....	214		
2.16.17 Raster Coloring Function .....	217		

**M306V5ME-XXXSP**  
**M306V5EESP**

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).

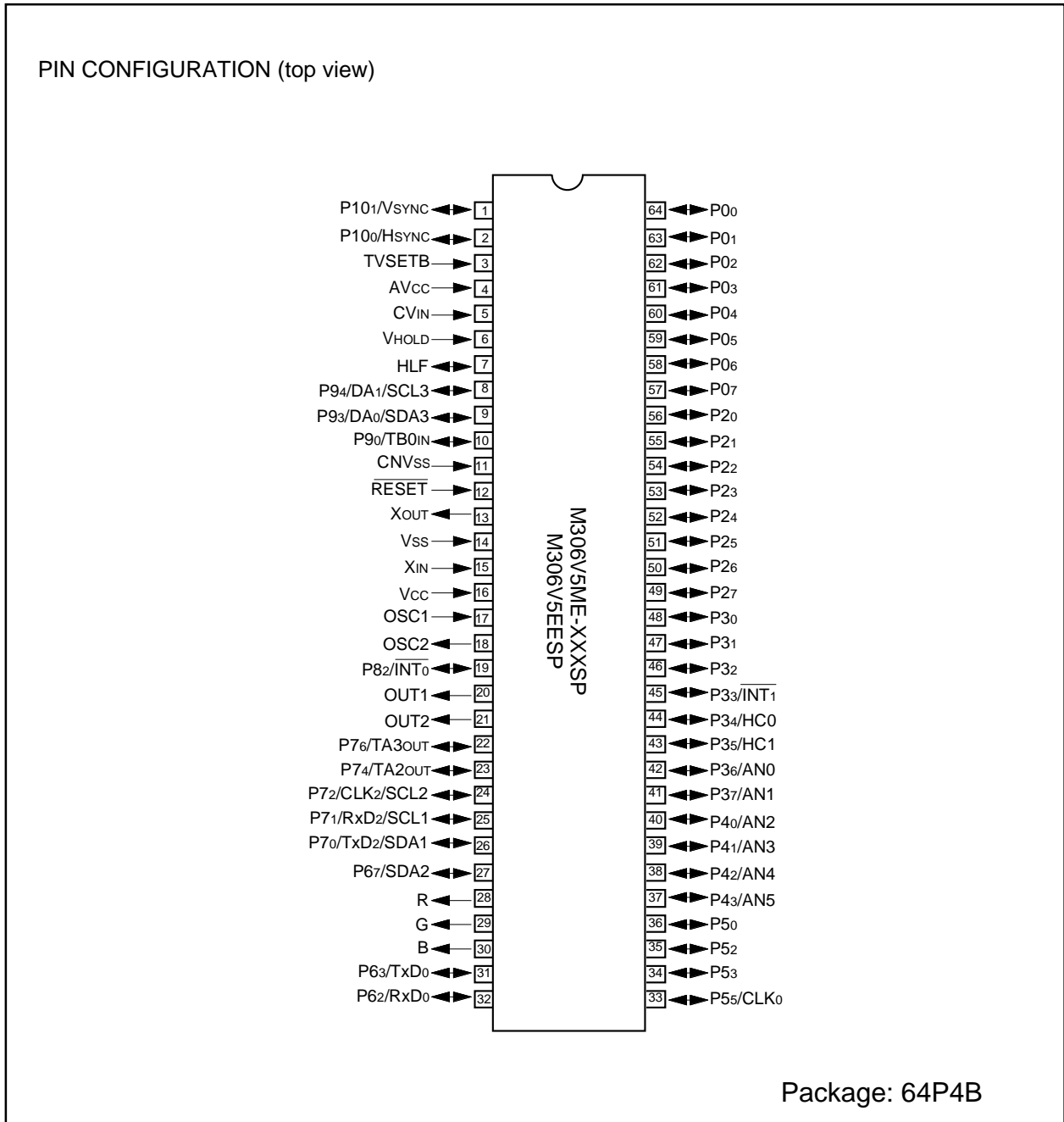


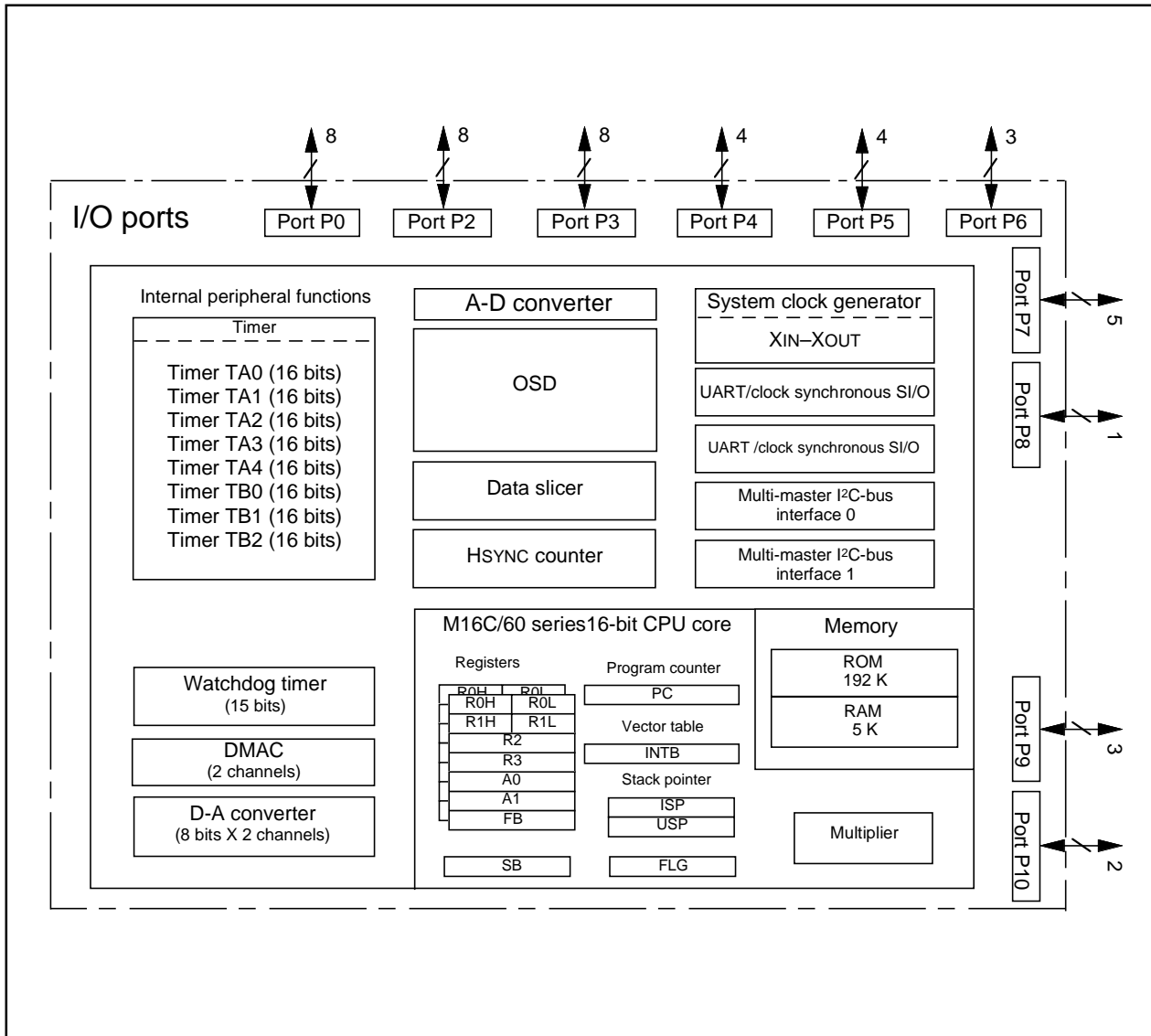
Figure 1.3.1 Pin configuration (top view)

**M306V5ME-XXXSP**  
**M306V5EESP**

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**

**M306V5ME-XXXSP**  
**M306V5EESP**

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	61K bytes
	OSD RAM	2.2K bytes
I/O port	P0, P2 to P10	8 bits X 3, 5 bits X 1, 4 bits X 2, 3 bits X 2, 2 bits X 1, 1 bit 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 or clock synchronous
	Multi-master I <sup>2</sup> C-BUS interface 0	1 unit (2 channels)
	Multi-master I <sup>2</sup> C-BUS interface 1	1 unit (1 channels)
A-D converter		8 bits X 6 channels
D-A converter		8 bits X 2 channels
DMAC		2 channels (trigger: 23 sources)
OSD function		Triple layer, 890 kinds of fonts, 42 character X 16 lines
Data slicer		32-bit buffer
HSYNC counter		8 bits X 2 channels
Watchdog timer		15 bits X 1 (with prescaler)
Interrupt		21 internal and 3 external sources, 4 software sources, 7 levels
Clock generating circuit		2 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
Operating ambient temperature		-10 °C to 70 °C
Device configuration		CMOS high performance silicon gate
Package		64-pin plastic molded SDIP

**M306V5ME-XXXSP**  
**M306V5EESP**

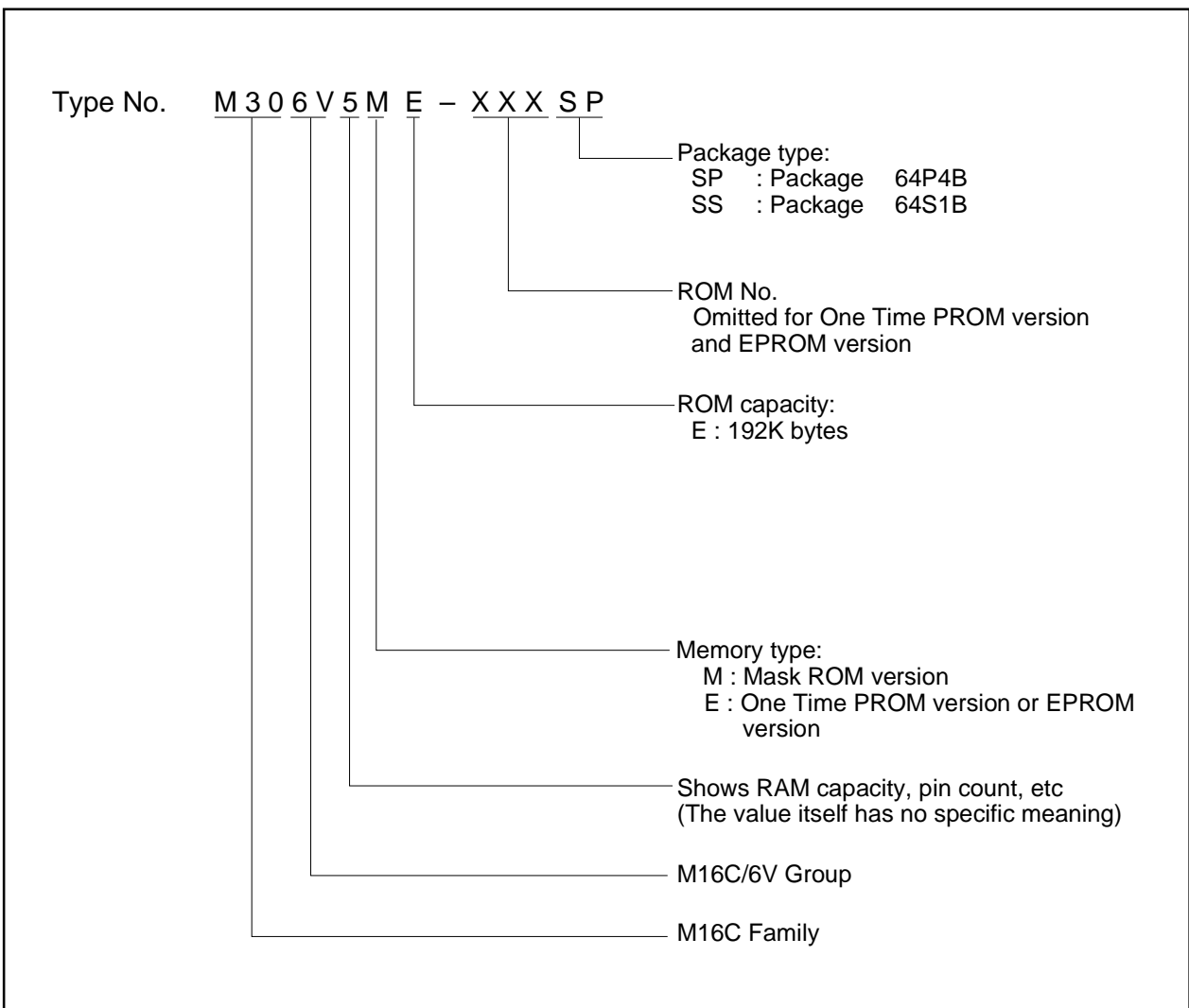
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

Currently supported products are listed below.

**Table 1.5.2 List of supported products**

Type No	ROM capacity	RAM capacity	Package type	Remarks
M306V5ME-XXXSP	192K bytes	5K bytes	64P4B	Mask ROM version
M306V5EESP	192K bytes	5K bytes	64P4B	One Time PROM version
M306V5EESS	192K bytes	5K bytes	64S1B	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**

### 1.5.1 As For M16C/6V (64-Pin Version) Group

M16C/6V (64-pin version) group is packaged in a 64-pin plastic molded SDIP. Note that the number of pins is reduced when it is compared with a 100-pin package product.

- (1) M16C/6V (64-pin version) group supports only the shingle-chip mode. It does not support the memory expansion and the microprocessor modes.
- (2) Be sure to initialize in the sequence below immediately after reset release.
  - ① Set OSD reserved register  $i$  ( $i = 1$  to  $4$ ) to the specified values.
  - ② Set each reserved bit of the port  $P_i$  direction register, the port  $P_i$  register, and pull-up control register  $i$  to the specified values.
  - ③ Set port reserved register  $i$  ( $i = 1$  to  $3$ ) to the specified values.
  - ④ Set other reserved registers and each reserved bit of other registers to the specified values.



**M306V5ME-XXXSP**  
**M306V5EESP**

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	Connect this pin to the Vss pin.
$\overline{\text{RESET}}$	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.
AVcc	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to Vcc.
P00 to P07	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, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistor.
P20 to P27	I/O port P2	Input/output	This is an 8-bit I/O port equivalent to P0.
P30 to P37	I/O port P3	Input/output	This is an 8-bit I/O port equivalent to P0. Pins in this port function as external interrupt pin, HSYNC counter I/O pins, and A-D converter input pins as selected by software.
P40 to P43	I/O port P4	Input/output	This is an 8-bit I/O port equivalent to P0. Pins in this port function as A-D converter input pins as selected by software.
P50, P52, P53, P55	I/O port P5	Input/output	This is a 4-bit I/O port equivalent to P0. P57 in this port functions as UART0 I/O pin as selected by software.
P62, P63, P67	I/O port P6	Input/output	This is a 3-bit I/O port equivalent to P0. Pins in this port also function as UART0 and multi-master I <sup>2</sup> C-BUS interface 0 I/O pins as selected by software.
P70 to P72, P74, P76	I/O port P7	Input/output	This is a 5-bit I/O port equivalent to P0 (P70 and P71 are N-channel open-drain output). Pins in this port also function as timers A2 and A3, UART2, multi-master I <sup>2</sup> C-BUS interface 0 I/O pins as selected by software.
P82	I/O port P8	Input/output	P82 is I/O port with the same functions as P0. P82 can be made to function as the I/O pin for the input pins for external interrupts as selected by software.
P90, P93, P94	I/O port P9	Input/output	This is an 3-bit I/O port equivalent to P0. Pins in this port also function as Timer B0 input pin, D-A converter output pins, and multi-master I <sup>2</sup> C-BUS interface 1 I/O pins as selected by software.

**M306V5ME-XXXSP**  
**M306V5EESP**

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
P10 <sub>0</sub> , P10 <sub>1</sub>	I/O port P10	Input/output	This is a 2-bit I/O port equivalent to P0. Pins in this port also function as a input pins for OSD function as selected by software.
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 BLOKS

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.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

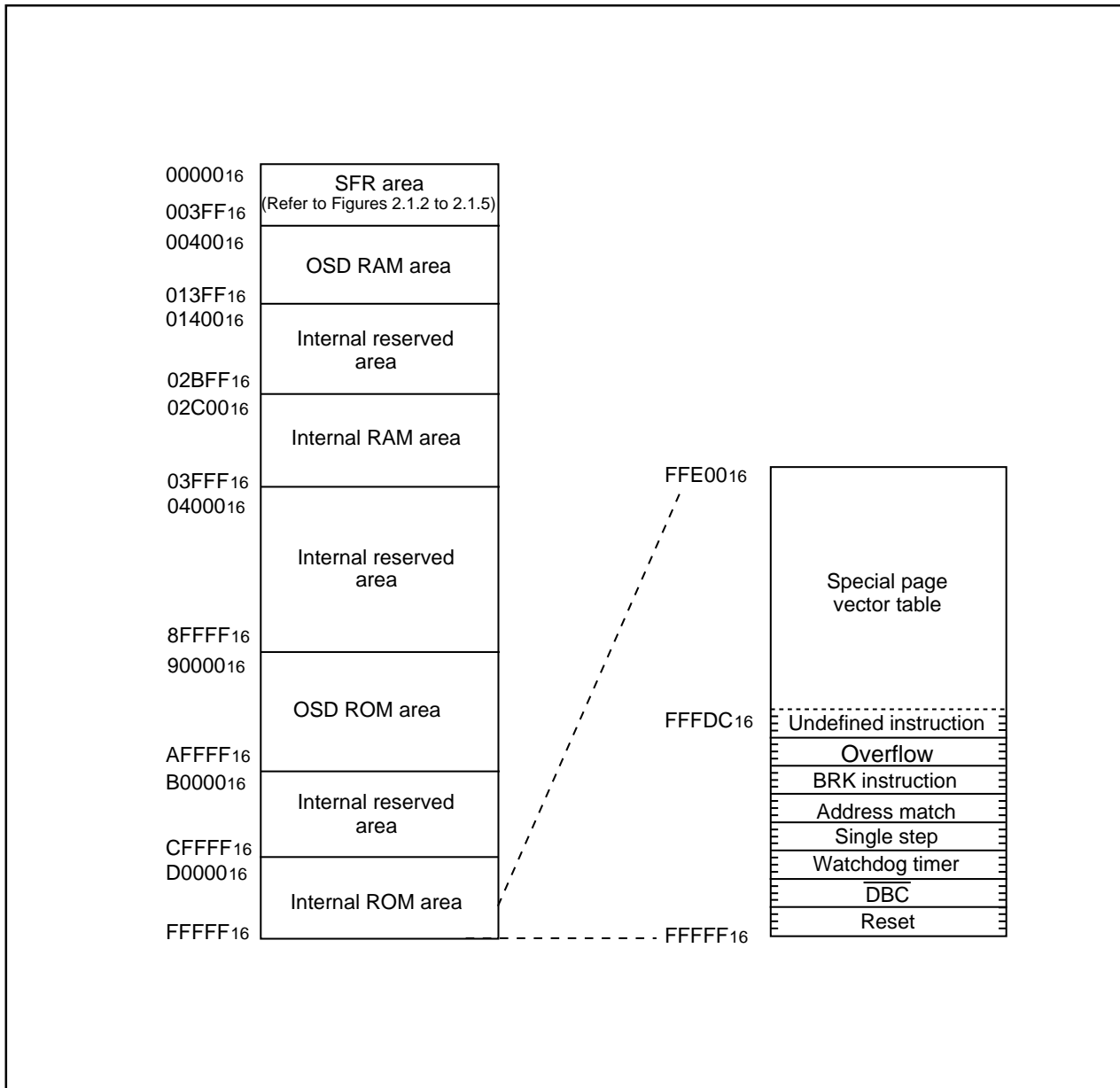


Figure 2.1.1 Memory map

**M306V5ME-XXXSP**  
**M306V5EESP**

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>		0048 <sub>16</sub>	OSD2 interrupt control register (OSD2IC)
0009 <sub>16</sub>	Address match interrupt enable register (AIER)	0049 <sub>16</sub>	Multi-master I <sup>2</sup> C-BUS interface 1 interrupt control register (IIC1IC)
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>	Multi-master I <sup>2</sup> C-BUS interface 0 interrupt control register (IIC0IC)
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 3 (RE3IC)
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)

**M306V5ME-XXXSP**  
**M306V5EESP**

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

0200 <sub>16</sub>		0240 <sub>16</sub>	Color palette register 1 (CR1)
0201 <sub>16</sub>	SPRITE OSD control register (SC)	0241 <sub>16</sub>	
0202 <sub>16</sub>	OSD control register 1 (OC1)	0242 <sub>16</sub>	Color palette register 2 (CR2)
0203 <sub>16</sub>	OSD control register 2 (OC2)	0243 <sub>16</sub>	
0204 <sub>16</sub>	Horizontal position register (HP)	0244 <sub>16</sub>	Color palette register 3 (CR3)
0205 <sub>16</sub>	Clock control register (CS)	0245 <sub>16</sub>	
0206 <sub>16</sub>	I/O polarity control register (PC)	0246 <sub>16</sub>	Color palette register 4 (CR4)
0207 <sub>16</sub>	OSD control register 3 (OC3)	0247 <sub>16</sub>	
0208 <sub>16</sub>		0248 <sub>16</sub>	Color palette register 5 (CR5)
0209 <sub>16</sub>	Raster color register (RSC)	0249 <sub>16</sub>	
020A <sub>16</sub>		024A <sub>16</sub>	Color palette register 6 (CR6)
020B <sub>16</sub>		024B <sub>16</sub>	
020C <sub>16</sub>		024C <sub>16</sub>	Color palette register 7 (CR7)
020D <sub>16</sub>	Top border control register (TBR)	024D <sub>16</sub>	
020E <sub>16</sub>		024E <sub>16</sub>	Color palette register 9 (CR9)
020F <sub>16</sub>	Bottom border control register (BBR)	024F <sub>16</sub>	
0210 <sub>16</sub>	Block control register 1 (BC1)	0250 <sub>16</sub>	Color palette register 10 (CR10)
0211 <sub>16</sub>	Block control register 2 (BC2)	0251 <sub>16</sub>	
0212 <sub>16</sub>	Block control register 3 (BC3)	0252 <sub>16</sub>	Color palette register 11 (CR11)
0213 <sub>16</sub>	Block control register 4 (BC4)	0253 <sub>16</sub>	
0214 <sub>16</sub>	Block control register 5 (BC5)	0254 <sub>16</sub>	Color palette register 12 (CR12)
0215 <sub>16</sub>	Block control register 6 (BC6)	0255 <sub>16</sub>	
0216 <sub>16</sub>	Block control register 7 (BC7)	0256 <sub>16</sub>	Color palette register 13 (CR13)
0217 <sub>16</sub>	Block control register 8 (BC8)	0257 <sub>16</sub>	
0218 <sub>16</sub>	Block control register 9 (BC9)	0258 <sub>16</sub>	Color palette register 14 (CR14)
0219 <sub>16</sub>	Block control register 10 (BC10)	0259 <sub>16</sub>	
021A <sub>16</sub>	Block control register 11 (BC11)	025A <sub>16</sub>	Color palette register 15 (CR15)
021B <sub>16</sub>	Block control register 12 (BC12)	025B <sub>16</sub>	
021C <sub>16</sub>	Block control register 13 (BC13)	025C <sub>16</sub>	
021D <sub>16</sub>	Block control register 14 (BC14)	025D <sub>16</sub>	OSD reserved register 1 (OR1)
021E <sub>16</sub>	Block control register 15 (BC15)	025E <sub>16</sub>	
021F <sub>16</sub>	Block control register 16 (BC16)	025F <sub>16</sub>	OSD control register 4 (OC4)
0220 <sub>16</sub>		0260 <sub>16</sub>	Data slicer control register 1 (DSC1)
0221 <sub>16</sub>	Vertical position register 1 (VP1)	0261 <sub>16</sub>	Data slicer control register 2 (DSC2)
0222 <sub>16</sub>		0262 <sub>16</sub>	Caption data register 1 (CD1)
0223 <sub>16</sub>	Vertical position register 2 (VP2)	0263 <sub>16</sub>	
0224 <sub>16</sub>		0264 <sub>16</sub>	Caption data register 2 (CD2)
0225 <sub>16</sub>	Vertical position register 3 (VP3)	0265 <sub>16</sub>	
0226 <sub>16</sub>		0266 <sub>16</sub>	Caption position register (CPS)
0227 <sub>16</sub>	Vertical position register 4 (VP4)	0267 <sub>16</sub>	Data slicer reserved register 2 (DR2)
0228 <sub>16</sub>		0268 <sub>16</sub>	Data slicer reserved register 1 (DR1)
0229 <sub>16</sub>	Vertical position register 5 (VP5)	0269 <sub>16</sub>	Clock run-in detect register (CRD)
022A <sub>16</sub>		026A <sub>16</sub>	Data clock position register (DPS)
022B <sub>16</sub>	Vertical position register 6 (VP6)	026B <sub>16</sub>	
022C <sub>16</sub>		≈	≈
022D <sub>16</sub>	Vertical position register 7 (VP7)	026F <sub>16</sub>	
022E <sub>16</sub>		0270 <sub>16</sub>	Left border control register (LBR)
022F <sub>16</sub>	Vertical position register 8 (VP8)	0271 <sub>16</sub>	
0230 <sub>16</sub>		0272 <sub>16</sub>	Right border control register (RBR)
0231 <sub>16</sub>	Vertical position register 9 (VP9)	0273 <sub>16</sub>	
0232 <sub>16</sub>		0274 <sub>16</sub>	SPRITE vertical position register 1 (VS1)
0233 <sub>16</sub>	Vertical position register 10 (VP10)	0275 <sub>16</sub>	
0234 <sub>16</sub>		0276 <sub>16</sub>	SPRITE vertical position register 2 (VS2)
0235 <sub>16</sub>	Vertical position register 11 (VP11)	0277 <sub>16</sub>	
0236 <sub>16</sub>		0278 <sub>16</sub>	SPRITE horizontal position register (HS)
0237 <sub>16</sub>	Vertical position register 12 (VP12)	0279 <sub>16</sub>	
0238 <sub>16</sub>		027A <sub>16</sub>	OSD reserved register 4 (OR4)
0239 <sub>16</sub>	Vertical position register 13 (VP13)	027B <sub>16</sub>	OSD reserved register 3 (OR3)
023A <sub>16</sub>		027C <sub>16</sub>	OSD reserved register 2 (OR2)
023B <sub>16</sub>	Vertical position register 14 (VP14)	027D <sub>16</sub>	Peripheral mode register (PM)
023C <sub>16</sub>		027E <sub>16</sub>	HSYNC counter register (HC)
023D <sub>16</sub>	Vertical position register 15 (VP15)	027F <sub>16</sub>	HSYNC counter latch
023E <sub>16</sub>		0280 <sub>16</sub>	
023F <sub>16</sub>	Vertical position register 16 (VP16)	≈	≈
		02DF <sub>16</sub>	

Figure 2.1.3 Location of peripheral unit control registers (2)

**M306V5ME-XXXSP**  
**M306V5EESP**

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

02E0 <sub>16</sub>	I <sup>2</sup> C0 data shift register (IIC0S0)	0380 <sub>16</sub>	Count start flag (TABSR)
02E1 <sub>16</sub>	I <sup>2</sup> C0 address register (IIC0S0D)	0381 <sub>16</sub>	Reserved register 6 (INVC6)
02E2 <sub>16</sub>	I <sup>2</sup> C0 status register (IIC0S1)	0382 <sub>16</sub>	One-shot start flag (ONSF)
02E3 <sub>16</sub>	I <sup>2</sup> C0 control register (IIC0S1D)	0383 <sub>16</sub>	Trigger select register (TRGSR)
02E4 <sub>16</sub>	I <sup>2</sup> C0 clock control register (IIC0S2)	0384 <sub>16</sub>	Up-down flag (UDF)
02E5 <sub>16</sub>	I <sup>2</sup> C0 port selection register (IIC0S2D)	0385 <sub>16</sub>	
02E6 <sub>16</sub>	I <sup>2</sup> C0 transmit buffer register (IIC0S0S)	0386 <sub>16</sub>	Timer A0 register (TA0)
02E7 <sub>16</sub>		0387 <sub>16</sub>	
02E8 <sub>16</sub>	I <sup>2</sup> C1 data shift register (IIC1S0)	0388 <sub>16</sub>	Timer A1 register (TA1)
02E9 <sub>16</sub>	I <sup>2</sup> C1 address register (IIC1S0D)	0389 <sub>16</sub>	
02EA <sub>16</sub>	I <sup>2</sup> C1 status register (IIC1S1)	038A <sub>16</sub>	Timer A2 register (TA2)
02EB <sub>16</sub>	I <sup>2</sup> C1 control register (IIC1S1D)	038B <sub>16</sub>	
02EC <sub>16</sub>	I <sup>2</sup> C1 clock control register (IIC1S2)	038C <sub>16</sub>	Timer A3 register (TA3)
02ED <sub>16</sub>	I <sup>2</sup> C1 port selection register (IIC1S2D)	038D <sub>16</sub>	
02EE <sub>16</sub>	I <sup>2</sup> C1 transmit buffer register (IIC1S0S)	038E <sub>16</sub>	Timer A4 register (TA4)
02EF <sub>16</sub>		038F <sub>16</sub>	
	≈	0390 <sub>16</sub>	Timer B0 register (TB0)
		0391 <sub>16</sub>	
0339 <sub>16</sub>		0392 <sub>16</sub>	Timer B1 register (TB1)
0340 <sub>16</sub>	Reserved register 1 (INVC1)	0393 <sub>16</sub>	
0341 <sub>16</sub>		0394 <sub>16</sub>	Timer B2 register (TB2)
0342 <sub>16</sub>		0395 <sub>16</sub>	
0343 <sub>16</sub>		0396 <sub>16</sub>	Timer A0 mode register (TA0MR)
0344 <sub>16</sub>		0397 <sub>16</sub>	Timer A1 mode register (TA1MR)
0345 <sub>16</sub>		0398 <sub>16</sub>	Timer A2 mode register (TA2MR)
0346 <sub>16</sub>		0399 <sub>16</sub>	Timer A3 mode register (TA3MR)
0347 <sub>16</sub>		039A <sub>16</sub>	Timer A4 mode register (TA4MR)
0348 <sub>16</sub>	Reserved register 0 (INVC0)	039B <sub>16</sub>	Timer B0 mode register (TB0MR)
0349 <sub>16</sub>		039C <sub>16</sub>	Timer B1 mode register (TB1MR)
	≈	039D <sub>16</sub>	Timer B2 mode register (TB2MR)
		039E <sub>16</sub>	
035E <sub>16</sub>		039F <sub>16</sub>	
035F <sub>16</sub>	Interrupt request cause select register (IFSR)	03A0 <sub>16</sub>	UART0 transmit/receive mode register (U0MR)
0360 <sub>16</sub>		03A1 <sub>16</sub>	UART0 bit rate generator (U0BRG)
0361 <sub>16</sub>		03A2 <sub>16</sub>	
0362 <sub>16</sub>	Reserved register 3 (INVC3)	03A3 <sub>16</sub>	UART0 transmit buffer register (U0TB)
0363 <sub>16</sub>		03A4 <sub>16</sub>	UART0 transmit/receive control register 0 (U0C0)
0364 <sub>16</sub>		03A5 <sub>16</sub>	UART0 transmit/receive control register 1 (U0C1)
0365 <sub>16</sub>		03A6 <sub>16</sub>	
0366 <sub>16</sub>	Reserved register 4 (INVC4)	03A7 <sub>16</sub>	UART0 receive buffer register (U0RB)
0367 <sub>16</sub>		03A8 <sub>16</sub>	Reserved register 2 (INVC2)
0368 <sub>16</sub>		03A9 <sub>16</sub>	
0369 <sub>16</sub>		03AA <sub>16</sub>	
036A <sub>16</sub>		03AB <sub>16</sub>	
036B <sub>16</sub>		03AC <sub>16</sub>	
036C <sub>16</sub>		03AD <sub>16</sub>	
036D <sub>16</sub>		03AE <sub>16</sub>	
036E <sub>16</sub>		03AF <sub>16</sub>	
036F <sub>16</sub>		03B0 <sub>16</sub>	UART transmit/receive control register 2 (U0CON)
0370 <sub>16</sub>		03B1 <sub>16</sub>	
0371 <sub>16</sub>		03B2 <sub>16</sub>	
0372 <sub>16</sub>		03B3 <sub>16</sub>	
0373 <sub>16</sub>		03B4 <sub>16</sub>	
0374 <sub>16</sub>		03B5 <sub>16</sub>	
0375 <sub>16</sub>		03B6 <sub>16</sub>	
0376 <sub>16</sub>	Reserved register 5 (INVC5)	03B7 <sub>16</sub>	
0377 <sub>16</sub>	UART2 special mode register (U2SMR)	03B8 <sub>16</sub>	DMA0 request cause select register (DM0SL)
0378 <sub>16</sub>	UART2 transmit/receive mode register (U2MR)	03B9 <sub>16</sub>	
0379 <sub>16</sub>	UART2 bit rate generator (U2BRG)	03BA <sub>16</sub>	DMA1 request cause select register (DM1SL)
037A <sub>16</sub>		03BB <sub>16</sub>	
037B <sub>16</sub>	UART2 transmit buffer register (U2TB)	03BC <sub>16</sub>	
037C <sub>16</sub>	UART2 transmit/receive control register 0 (U2C0)	03BD <sub>16</sub>	
037D <sub>16</sub>	UART2 transmit/receive control register 1 (U2C1)	03BE <sub>16</sub>	
037E <sub>16</sub>		03BF <sub>16</sub>	
037F <sub>16</sub>	UART2 receive buffer register (U2RB)		

Figure 2.1.4 Location of peripheral unit control registers (3)

**M306V5ME-XXXSP**  
**M306V5EESP**

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

03C0 <sub>16</sub>	
03C1 <sub>16</sub>	
03C2 <sub>16</sub>	
03C3 <sub>16</sub>	
03C4 <sub>16</sub>	A-D register 0 (AD0)
03C5 <sub>16</sub>	
03C6 <sub>16</sub>	A-D register 1 (AD1)
03C7 <sub>16</sub>	
03C8 <sub>16</sub>	A-D register 2 (AD2)
03C9 <sub>16</sub>	
03CA <sub>16</sub>	A-D register 3 (AD3)
03CB <sub>16</sub>	
03CC <sub>16</sub>	A-D register 4 (AD4)
03CD <sub>16</sub>	
03CE <sub>16</sub>	A-D register 5 (AD5)
03CF <sub>16</sub>	
03D0 <sub>16</sub>	
03D1 <sub>16</sub>	
03D2 <sub>16</sub>	
03D3 <sub>16</sub>	
03D4 <sub>16</sub>	A-D control register 2 (ADCON2)
03D5 <sub>16</sub>	
03D6 <sub>16</sub>	A-D control register 0 (ADCON0)
03D7 <sub>16</sub>	A-D control register 1 (ADCON1)
03D8 <sub>16</sub>	D-A register 0 (DA0)
03D9 <sub>16</sub>	
03DA <sub>16</sub>	D-A register 1 (DA1)
03DB <sub>16</sub>	
03DC <sub>16</sub>	D-A control register (DACON)
03DD <sub>16</sub>	
03DE <sub>16</sub>	
03DF <sub>16</sub>	
03E0 <sub>16</sub>	Port P0 register (P0)
03E1 <sub>16</sub>	Port reserved register 1 (PR1)
03E2 <sub>16</sub>	Port P0 direction register (PD0)
03E3 <sub>16</sub>	Port reserved register 2 (PR2)
03E4 <sub>16</sub>	Port P2 register (P2)
03E5 <sub>16</sub>	Port P3 register (P3)
03E6 <sub>16</sub>	Port P2 direction register (PD2)
03E7 <sub>16</sub>	Port P3 direction register (PD3)
03E8 <sub>16</sub>	Port P4 register (P4)
03E9 <sub>16</sub>	Port P5 register (P5)
03EA <sub>16</sub>	Port P4 direction register (PD4)
03EB <sub>16</sub>	Port P5 direction register (PD5)
03EC <sub>16</sub>	Port P6 register (P6)
03ED <sub>16</sub>	Port P7 register (P7)
03EE <sub>16</sub>	Port P6 direction register (PD6)
03EF <sub>16</sub>	Port P7 direction register (PD7)
03F0 <sub>16</sub>	Port P8 register (P8)
03F1 <sub>16</sub>	Port P9 register (P9)
03F2 <sub>16</sub>	Port P8 direction register (PD8)
03F3 <sub>16</sub>	Port P9 direction register (PD9)
03F4 <sub>16</sub>	Port P10 register (P10)
03F5 <sub>16</sub>	
03F6 <sub>16</sub>	Port P10 direction register (PD10)
03F7 <sub>16</sub>	
03F8 <sub>16</sub>	
03F9 <sub>16</sub>	
03FA <sub>16</sub>	
03FB <sub>16</sub>	
03FC <sub>16</sub>	Pull-up control register 0 (PUR0)
03FD <sub>16</sub>	Pull-up control register 1 (PUR1)
03FE <sub>16</sub>	Pull-up control register 2 (PUR2)
03FF <sub>16</sub>	Port reserved register 3 (PR3)

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.

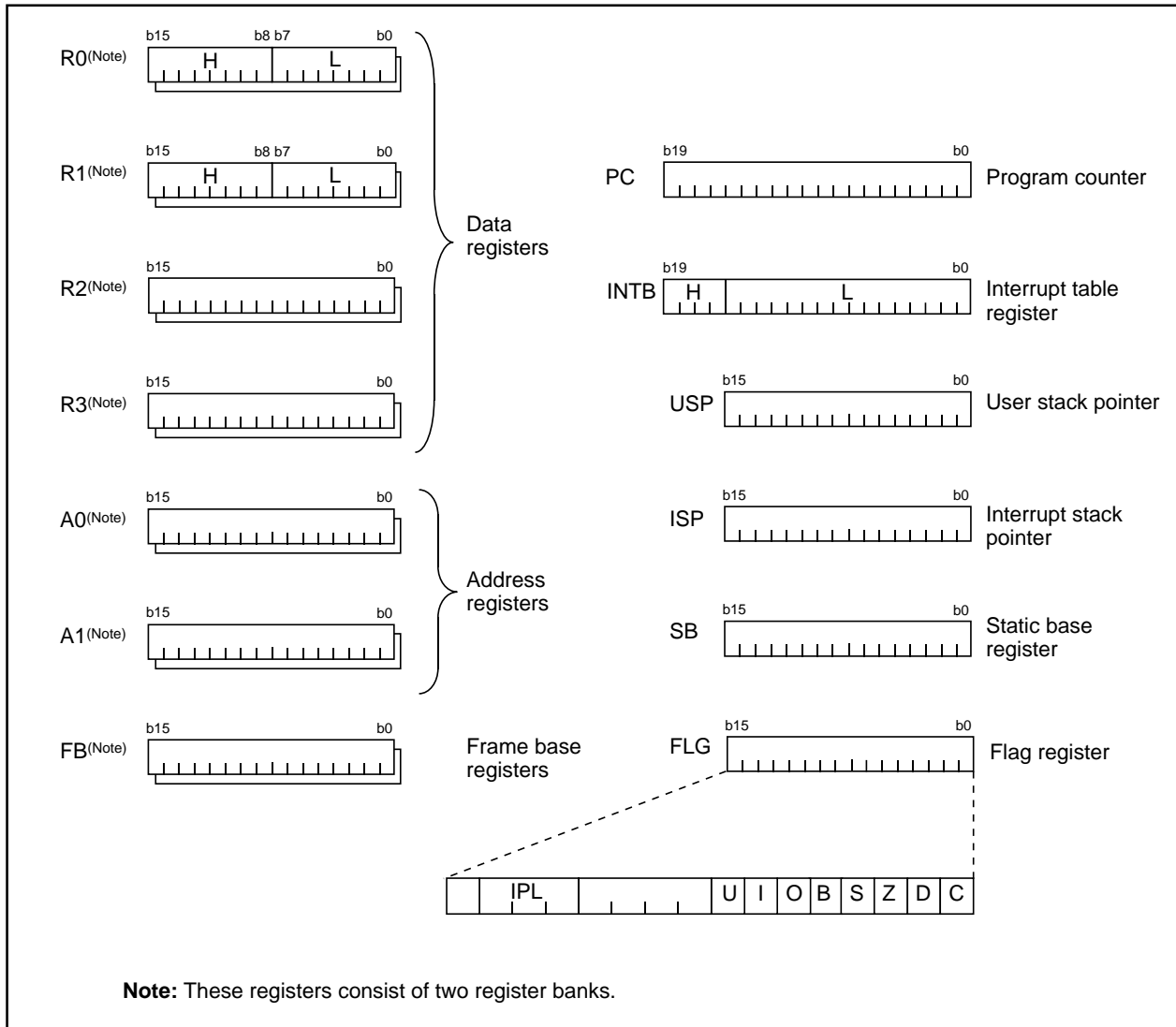


Figure 2.2.1 Central processing unit register

### 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".

**M306V5ME-XXXSP**  
**M306V5EESP**

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.

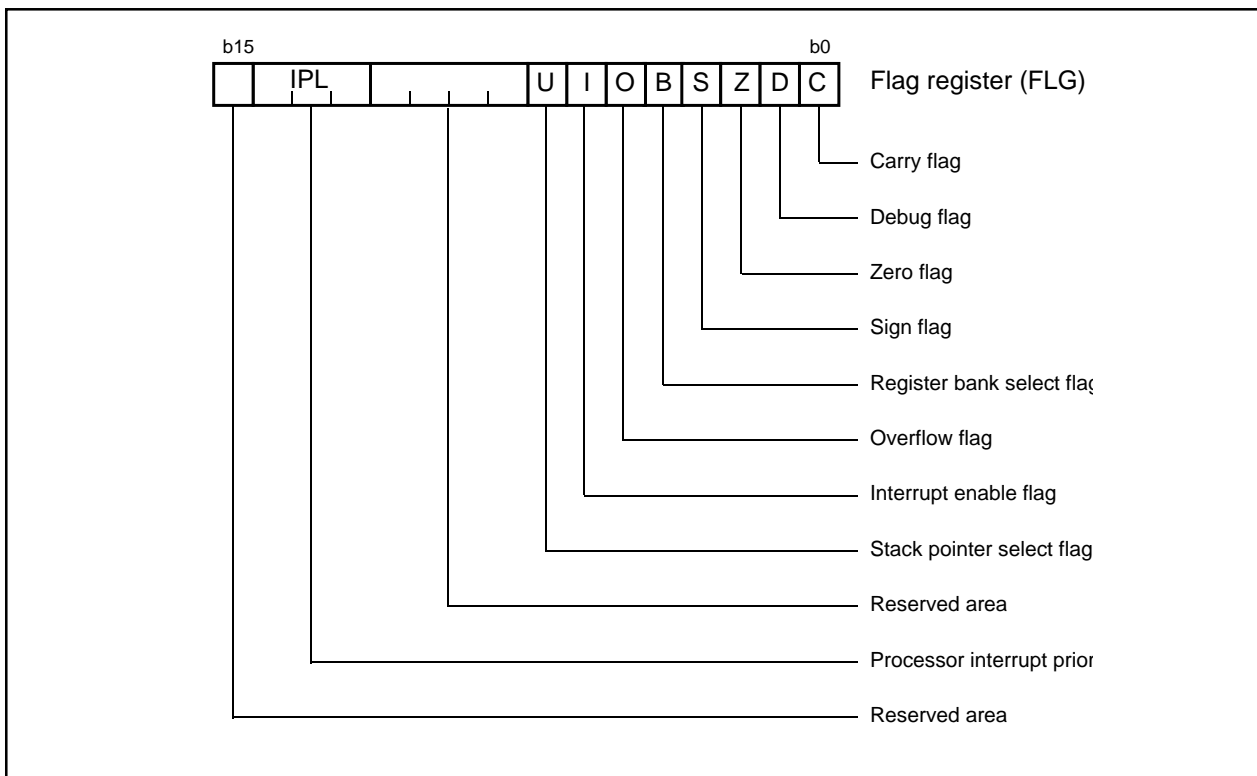


Figure 2.2.2 Flag register (FLG)

**M306V5ME-XXXSP**  
**M306V5EESP**

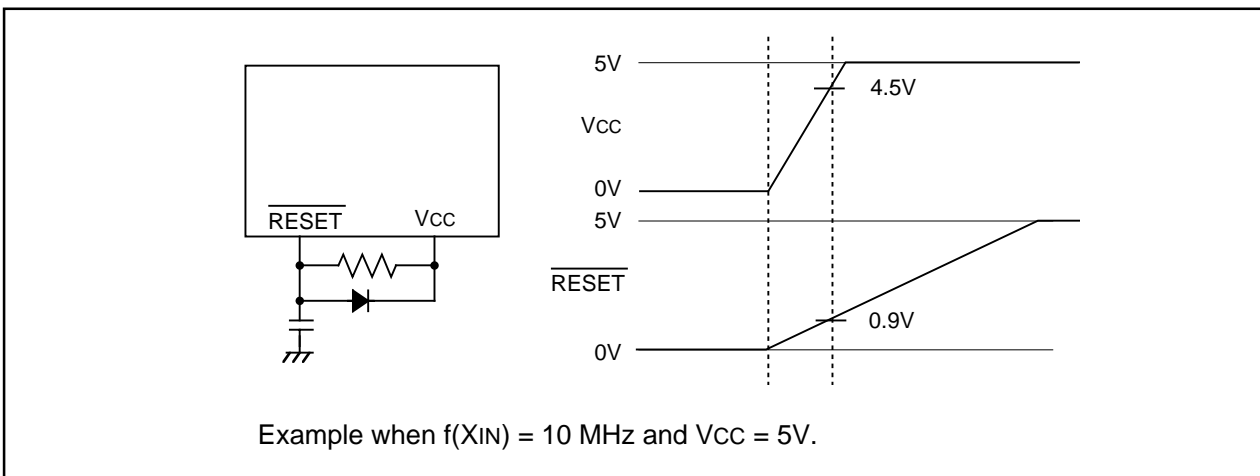
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**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.

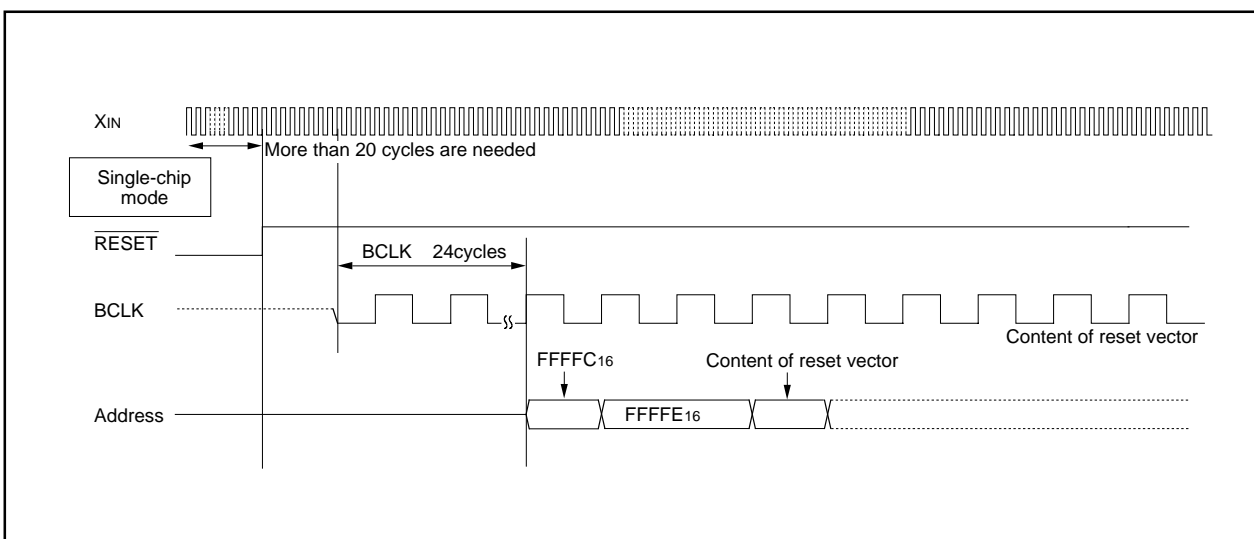
Figure 2.3.1 shows the example reset circuit. Figure 2.3.2 shows the reset sequence.



**Figure 2.3.1 Example reset circuit**

**2.3.1 Software Reset**

Writing "1" to bit 3 of the processor mode register 0 (address 0004<sub>16</sub>) 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.



**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
P0, P2, P3, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P70 to P72, P74, P76, P82, P90, P93, P94, P100, P101	Input port (floating)
R, G, B, OUT1,OUT2	Output port
CVIN, VHOLD, HLF	Input/output port
OSC1	Input port
OSC2	Output port

**M306V5ME-XXXSP**  
**M306V5EESP**

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>	Timer B0 interrupt control register	(005A <sub>16</sub> )...	XXXX?000
Processor mode register 1	(0005 <sub>16</sub> )...	0000X0	Timer B1 interrupt control register	(005B <sub>16</sub> )...	XXXX?000
System clock control register 0	(0006 <sub>16</sub> )...	48 <sub>16</sub>	Timer B2 interrupt control register	(005C <sub>16</sub> )...	XXXX?000
System clock control register 1	(0007 <sub>16</sub> )...	20 <sub>16</sub>	INT0 interrupt control register	(005D <sub>16</sub> )...	XX00?000
Address match interrupt enable register	(0009 <sub>16</sub> )...	XXXXXXXX00	INT1 interrupt control register	(005E <sub>16</sub> )...	XX00?000
Protect register	(000A <sub>16</sub> )...	XXXXXXXX00	SPRITE OSD control register	(0201 <sub>16</sub> )...	XXXX0000
Watchdog timer control register	(000F <sub>16</sub> )...	000? ??	OSD control register 1	(0202 <sub>16</sub> )...	00 <sub>16</sub>
Address match interrupt register 0	(0010 <sub>16</sub> )...	00 <sub>16</sub>	OSD control register 2	(0203 <sub>16</sub> )...	00 <sub>16</sub>
	(0011 <sub>16</sub> )...	00 <sub>16</sub>	Horizontal position register	(0204 <sub>16</sub> )...	00 <sub>16</sub>
	(0012 <sub>16</sub> )...	XXXXX000	Clock control register	(0205 <sub>16</sub> )...	00 <sub>16</sub>
Address match interrupt register 1	(0014 <sub>16</sub> )...	00 <sub>16</sub>	I/O polarity control register	(0206 <sub>16</sub> )...	10000000
	(0015 <sub>16</sub> )...	00 <sub>16</sub>	OSD control register 3	(0207 <sub>16</sub> )...	00 <sub>16</sub>
	(0016 <sub>16</sub> )...	XXXXX000	Raster color register	(0208 <sub>16</sub> )...	00 <sub>16</sub>
DMA0 control register	(002C <sub>16</sub> )...	00000?00		(0209 <sub>16</sub> )...	00 <sub>16</sub>
DMA1 control register	(003C <sub>16</sub> )...	00000?00	OSD reserved register 1	(025D <sub>16</sub> )...	00 <sub>16</sub>
OSD1 interrupt control register	(0044 <sub>16</sub> )...	XXXX?000	OSD control register 4	(025F <sub>16</sub> )...	XXXXXXXX00
OSD2 interrupt control register	(0048 <sub>16</sub> )...	XXXX?000	Data slicer control register 1	(0260 <sub>16</sub> )...	00 <sub>16</sub>
Multi-master I <sup>2</sup> C-BUS interface 1 interrupt control register	(0049 <sub>16</sub> )...	XX00?000	Data slicer control register 2	(0261 <sub>16</sub> )...	?0?0?0?
Bus collision detection interrupt control register	(004A <sub>16</sub> )...	XXXX?000	Caption position register	(0266 <sub>16</sub> )...	00000000
DMA0 interrupt control register	(004B <sub>16</sub> )...	XXXX?000	Data slicer reserved register 2	(0267 <sub>16</sub> )...	00 <sub>16</sub>
DMA1 interrupt control register	(004C <sub>16</sub> )...	XXXX?000	Data slicer reserved register 1	(0268 <sub>16</sub> )...	00 <sub>16</sub>
Multi-master I <sup>2</sup> C-BUS interface 0 interrupt control register	(004D <sub>16</sub> )...	XXXX?000	Clock run-in detect register	(0269 <sub>16</sub> )...	00 <sub>16</sub>
A-D conversion interrupt control register	(004E <sub>16</sub> )...	XXXX?000	Data clock position register	(026A <sub>16</sub> )...	XXXX00001
UART2 transmit interrupt control register	(004F <sub>16</sub> )...	XXXX?000	Left border control register	(0270 <sub>16</sub> )...	01 <sub>16</sub>
UART2 receive interrupt control register	(0050 <sub>16</sub> )...	XXXX?000		(0271 <sub>16</sub> )...	XXXXXXXX000
UART0 transmit interrupt control register	(0051 <sub>16</sub> )...	XXXX?000	Right border control register	(0272 <sub>16</sub> )...	00 <sub>16</sub>
UART0 receive interrupt control register	(0052 <sub>16</sub> )...	XXXX?000		(0273 <sub>16</sub> )...	XXXXX000
Data slicer interrupt control register	(0053 <sub>16</sub> )...	XXXX?000	SPRITE horizontal position register (high-order)	(0279 <sub>16</sub> )...	XXXXXXXX000
Vsync interrupt control register	(0054 <sub>16</sub> )...	XXXX?000	OSD reserved register 4	(027A <sub>16</sub> )...	00000000
Timer A0 interrupt control register	(0055 <sub>16</sub> )...	XXXX?000	OSD reserved register 3	(027B <sub>16</sub> )...	00 <sub>16</sub>
Timer A1 interrupt control register	(0056 <sub>16</sub> )...	XXXX?000	OSD reserved register 2	(027C <sub>16</sub> )...	00 <sub>16</sub>
Timer A2 interrupt control register	(0057 <sub>16</sub> )...	XXXX?000	Peripheral mode register	(027D <sub>16</sub> )...	0XX00000
Timer A3 interrupt control register	(0058 <sub>16</sub> )...	XXXX?000	Hsync counter register	(027E <sub>16</sub> )...	XXXX00XX
Timer A4 interrupt control register	(0059 <sub>16</sub> )...	XXXX?000			

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.

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

**M306V5ME-XXXSP**  
**M306V5EESP**

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

I <sup>2</sup> C0 address register	(02E1 <sub>16</sub> )...	00 <sub>16</sub>	UART transmit/receive control register 2	(03B0 <sub>16</sub> )...	00000000
I <sup>2</sup> C0 status register	(02E2 <sub>16</sub> )...	0001000?	DMA0 request cause select register	(03B8 <sub>16</sub> )...	00 <sub>16</sub>
I <sup>2</sup> C0 control register	(02E3 <sub>16</sub> )...	00 <sub>16</sub>	DMA1 request cause select register	(03BA <sub>16</sub> )...	00 <sub>16</sub>
I <sup>2</sup> C0 clock control register	(02E4 <sub>16</sub> )...	00 <sub>16</sub>	A-D control register 2	(03D4 <sub>16</sub> )...	0000???
I <sup>2</sup> C0 port selection register	(02E5 <sub>16</sub> )...	00?00000	A-D control register 0	(03D6 <sub>16</sub> )...	0000???
I <sup>2</sup> C1 address register	(02E9 <sub>16</sub> )...	00 <sub>16</sub>	A-D control register 1	(03D7 <sub>16</sub> )...	00 <sub>16</sub>
I <sup>2</sup> C1 status register	(02EA <sub>16</sub> )...	0001000?	D-A control register	(03DC <sub>16</sub> )...	00 <sub>16</sub>
I <sup>2</sup> C1 control register	(02EB <sub>16</sub> )...	00 <sub>16</sub>	Port P0 direction register	(03E2 <sub>16</sub> )...	00 <sub>16</sub>
I <sup>2</sup> C1 clock control register	(02EC <sub>16</sub> )...	00 <sub>16</sub>	Port reserved register 2	(03E3 <sub>16</sub> )...	00 <sub>16</sub>
I <sup>2</sup> C1 port selection register	(02ED <sub>16</sub> )...	00?00000	Port P2 direction register	(03E6 <sub>16</sub> )...	00 <sub>16</sub>
Reserved register 1	(0340 <sub>16</sub> )...	000???	Port P3 direction register	(03E7 <sub>16</sub> )...	00 <sub>16</sub>
Reserved register 0	(0348 <sub>16</sub> )...	00 <sub>16</sub>	Port P4 direction register	(03EA <sub>16</sub> )...	00 <sub>16</sub>
Interrupt request cause select register	(035F <sub>16</sub> )...	00 <sub>16</sub>	Port P5 direction register	(03EB <sub>16</sub> )...	00 <sub>16</sub>
Reserved register 3	(0362 <sub>16</sub> )...	40 <sub>16</sub>	Port P6 direction register	(03EE <sub>16</sub> )...	00 <sub>16</sub>
Reserved register 4	(0366 <sub>16</sub> )...	40 <sub>16</sub>	Port P7 direction register	(03EF <sub>16</sub> )...	00 <sub>16</sub>
Reserved register 5	(0376 <sub>16</sub> )...	00 <sub>16</sub>	Port P8 direction register	(03F2 <sub>16</sub> )...	00x00000
UART2 special mode register	(0377 <sub>16</sub> )...	00 <sub>16</sub>	Port P9 direction register	(03F3 <sub>16</sub> )...	00 <sub>16</sub>
UART2 transmit/receive mode register	(0378 <sub>16</sub> )...	00 <sub>16</sub>	Port P10 direction register	(03F6 <sub>16</sub> )...	00 <sub>16</sub>
UART2 transmit/receive control register 0	(037C <sub>16</sub> )...	08 <sub>16</sub>	Pull-up control register 0	(03FC <sub>16</sub> )...	00 <sub>16</sub>
UART2 transmit/receive control register 1	(037D <sub>16</sub> )...	02 <sub>16</sub>	Pull-up control register 1 (Note)	(03FD <sub>16</sub> )...	00 <sub>16</sub>
Count start flag	(0380 <sub>16</sub> )...	00 <sub>16</sub>	Pull-up control register 2	(03FE <sub>16</sub> )...	00 <sub>16</sub>
Reserved register 6	(0381 <sub>16</sub> )...	0xxxxxxx	Port reserved register 3	(03FF <sub>16</sub> )...	00 <sub>16</sub>
One-shot start flag	(0382 <sub>16</sub> )...	00x00000	Data registers (R0/R1/R2/R3)		0000 <sub>16</sub>
Trigger select register	(0383 <sub>16</sub> )...	00 <sub>16</sub>	Address registers (A0/A1)		0000 <sub>16</sub>
Up-down flag	(0384 <sub>16</sub> )...	00 <sub>16</sub>	Frame base register (FB)		0000 <sub>16</sub>
Timer A0 mode register	(0396 <sub>16</sub> )...	00 <sub>16</sub>	Interrupt table register (INTB)		00000 <sub>16</sub>
Timer A1 mode register	(0397 <sub>16</sub> )...	00 <sub>16</sub>	User stack pointer (USP)		0000 <sub>16</sub>
Timer A2 mode register	(0398 <sub>16</sub> )...	00 <sub>16</sub>	Interrupt stack pointer (ISP)		0000 <sub>16</sub>
Timer A3 mode register	(0399 <sub>16</sub> )...	00 <sub>16</sub>	Static base register (SB)		0000 <sub>16</sub>
Timer A4 mode register	(039A <sub>16</sub> )...	00 <sub>16</sub>	Flag register (FLG)		0000 <sub>16</sub>
Timer B0 mode register	(039B <sub>16</sub> )...	00?x0000			
Timer B1 mode register	(039C <sub>16</sub> )...	00?x0000			
Timer B2 mode register	(039D <sub>16</sub> )...	00?x0000			
UART0 transmit/receive mode register	(03A0 <sub>16</sub> )...	00 <sub>16</sub>			
UART0 transmit/receive control register 0	(03A4 <sub>16</sub> )...	08 <sub>16</sub>			
UART0 transmit/receive control register 1	(03A5 <sub>16</sub> )...	02 <sub>16</sub>			
Reserved register 2	(03A8 <sub>16</sub> )...	00 <sub>16</sub>			

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.

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

**M306V5ME-XXXSP**  
**M306V5EESP**

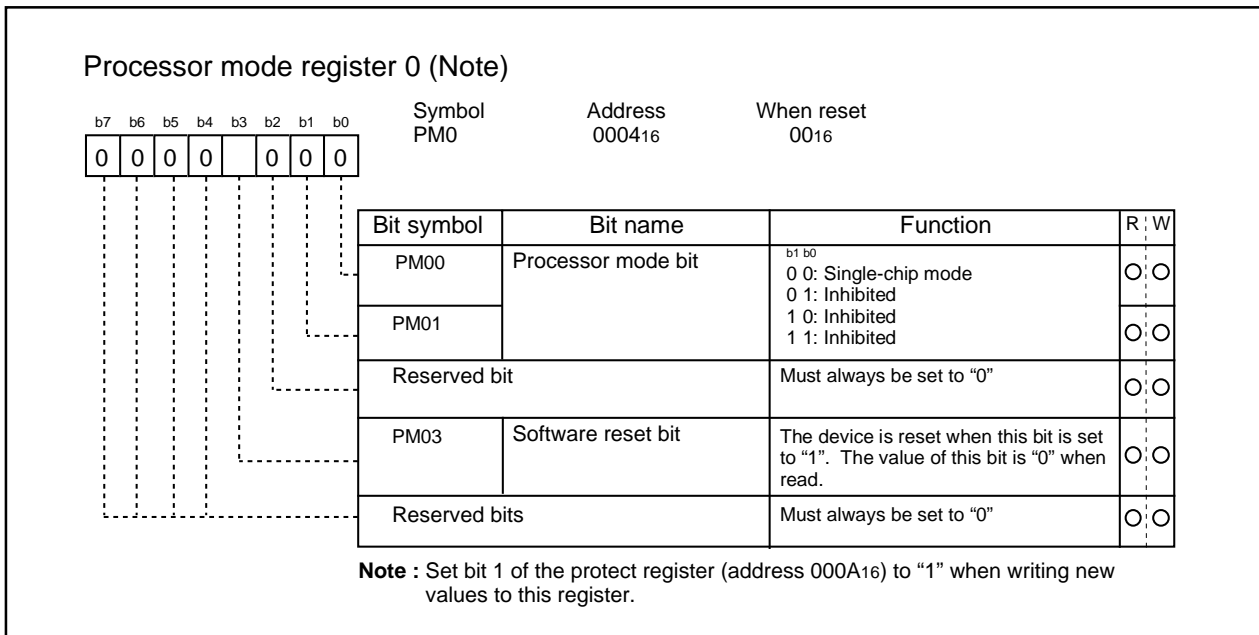
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**2.4 Single-chip Mode**

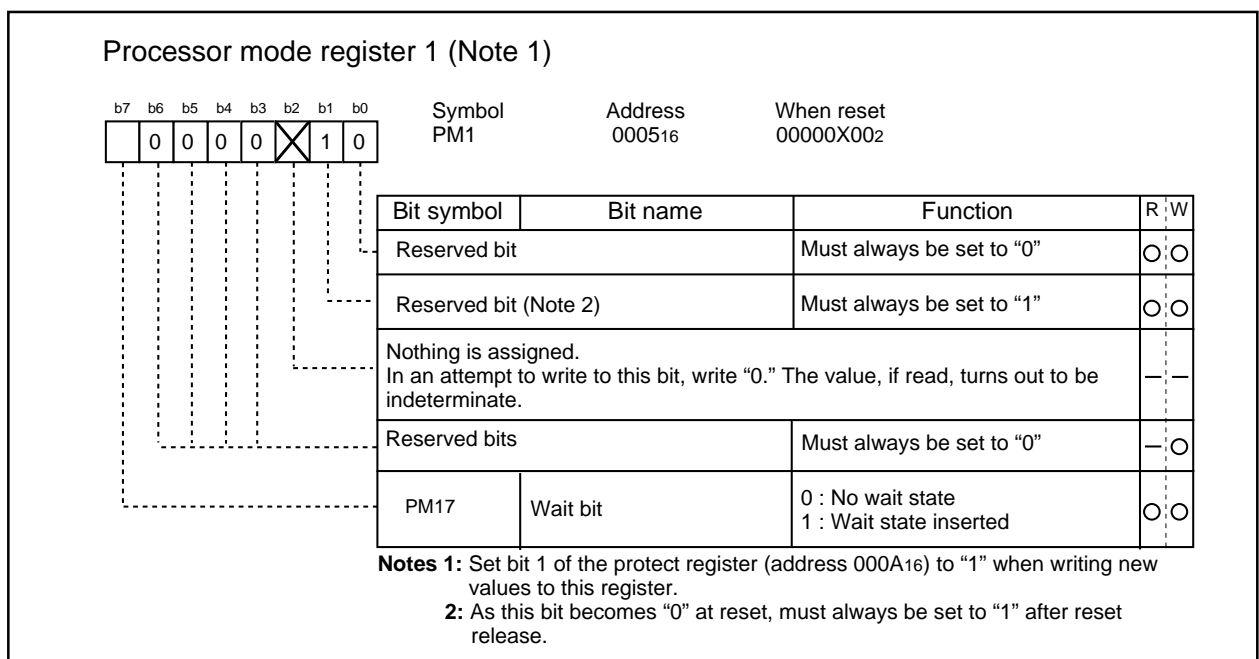
This microcomputer supports single-chip mode only.

In single-chip mode, only internal memory space (SFR, OSD RAM, internal RAM, and internal ROM) can be accessed. Ports P0, P2 to P10 can be used as programmable I/O ports or as I/O ports for the internal peripheral functions.

Figure 2.4.1 shows the processor mode register 0 and Figure 2.4.2 shows the processor mode register 1. Figure 2.4.3 shows the memory map.



**Figure 2.4.1 Processor mode register 0**



**Figure 2.4.2 Processor mode register 1**



**M306V5ME-XXXSP**  
**M306V5EESP**

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

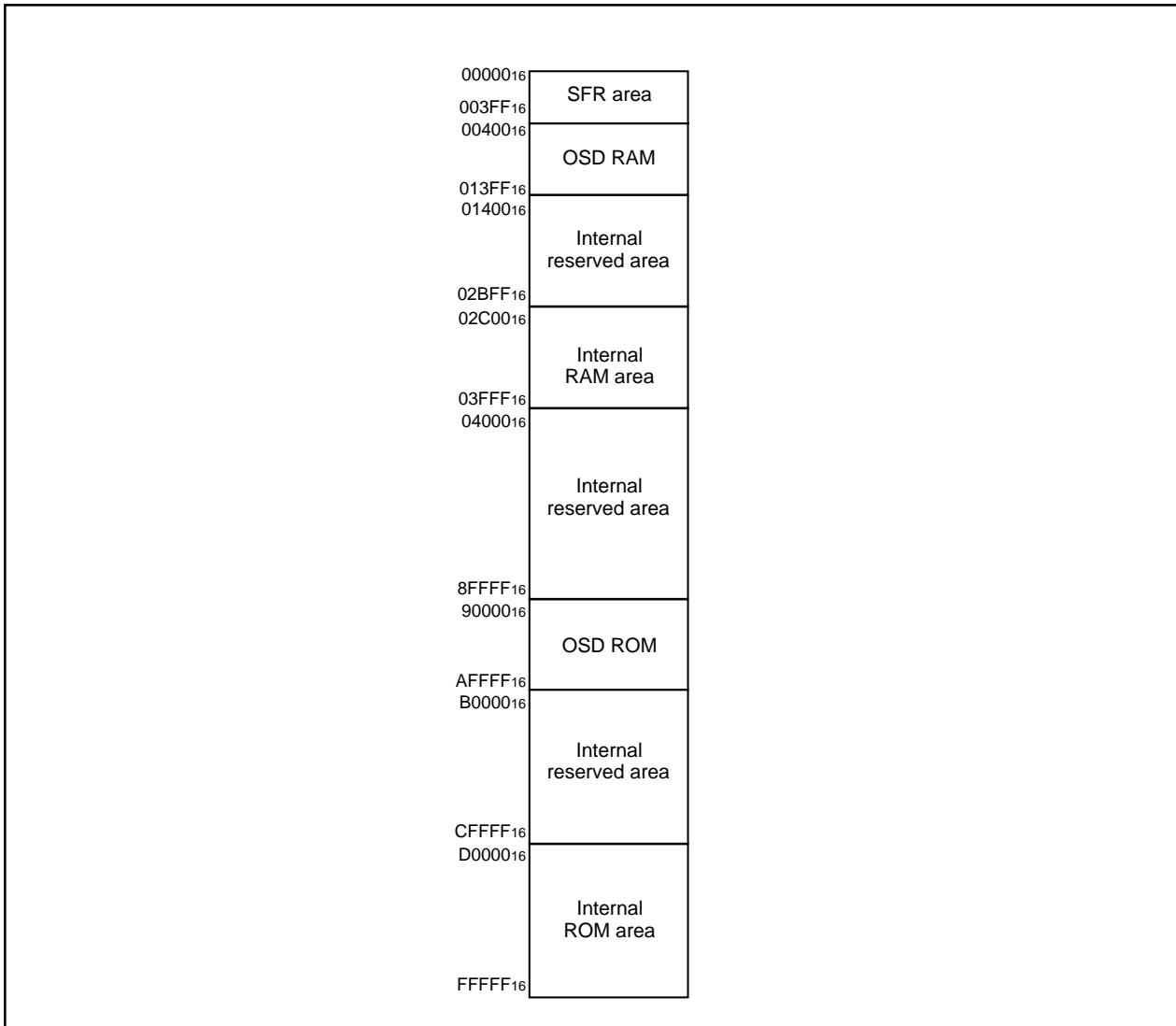


Figure 2.4.3 Memory map in single-chip mode

**2.4.1 Software Wait**

A software wait can be inserted by setting the wait bit (bit 7) of processor mode register 1 (address 0005<sub>16</sub>).

A software wait is inserted in the internal ROM/RAM 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 BCLK cycles. After the microcomputer has been reset, this bit defaults to “0”.

The SFR area and the OSD RAM area is always accessed in two BCLK cycles regardless of the setting of these control bits.

Table 2.4.1 shows the software wait and bus cycles. Figure 2.4.4 shows example bus timing when using software waits.

**Table 2.4.1 Software waits and bus cycles**

Area	Wait bit	Bus cycle
SFR/ OSD RAM	Invalid	2 BCLK cycles
Internal ROM/RAM	0	1 BCLK cycle
	1	2 BCLK cycles

**M306V5ME-XXXSP**  
**M306V5EESP**

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

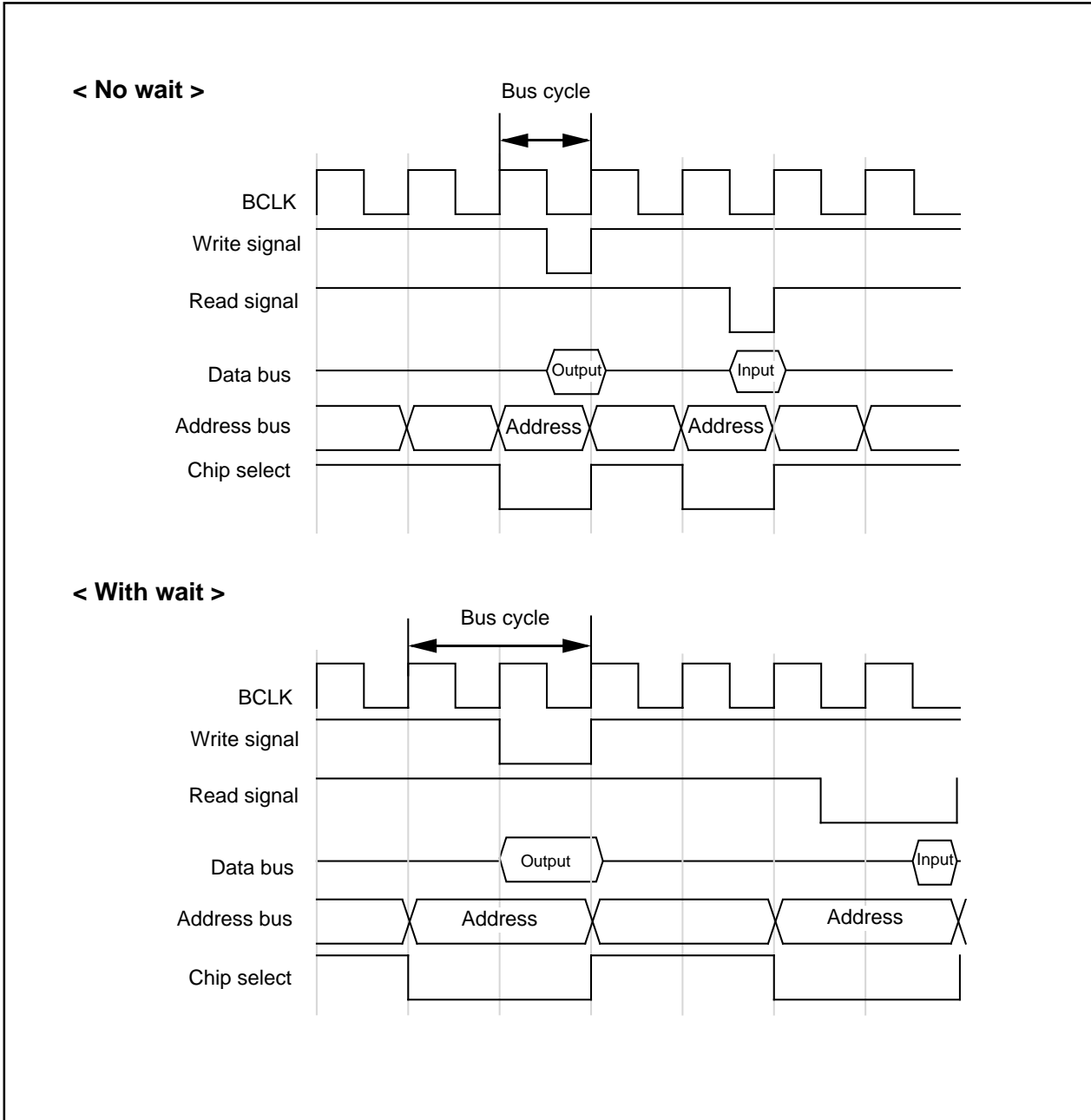


Figure 2.4.4 Typical bus timings using software wait

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**2.5 Clock Generating Circuit**

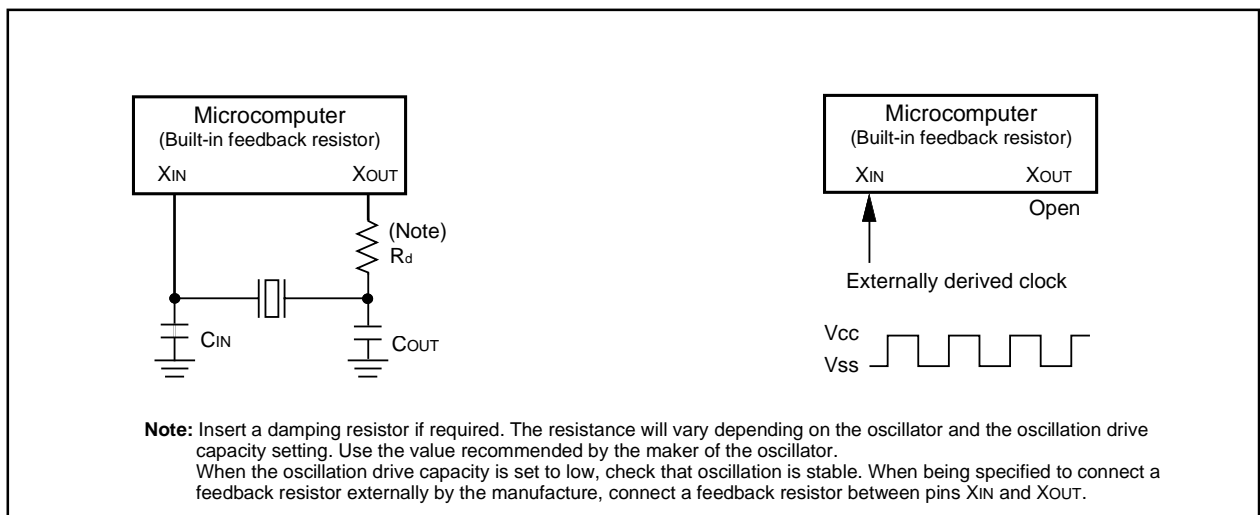
The clock generating circuit contains each oscillator circuit that supplies the operating clock sources to the CPU and internal peripheral units and that supplies the operating clock source to OSD.

**Table 2.5.1. Clock oscillation circuits**

	Main 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>• 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>• Ceramic resonator (or quartz-crystal oscillator)</li> <li>• LC oscillator</li> </ul>
Pins to connect oscillator	XIN, XOUT	OSC1, OSC2
Oscillation stop/restart function	Available	/
Oscillator status immediately after reset	Oscillating	
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. Circuit constants in Figure 2.5.1 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.



**Figure 2.5.1 Examples of main clock**

# M306V5ME-XXXSP M306V5EESP

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

## 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).

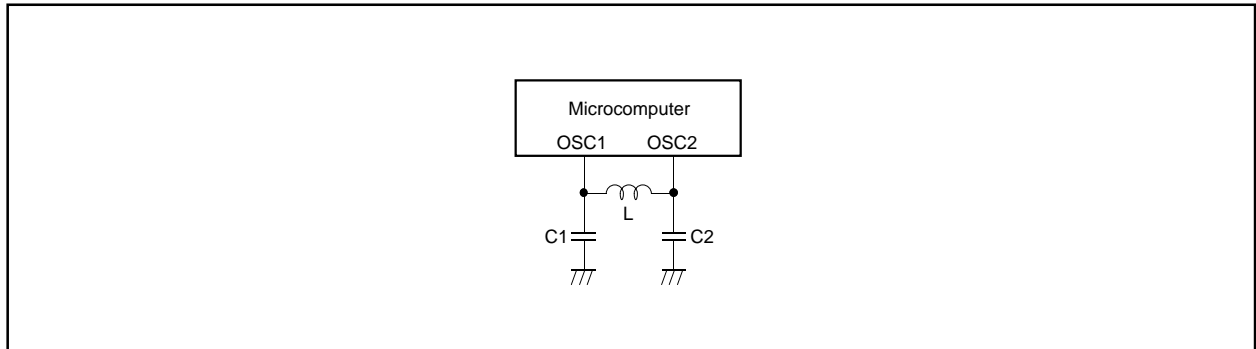


Figure 2.5.2 OSD clock connection example

## 2.5.3 Clock Control

Figure 2.5.3 shows the block diagram of the main clock generating circuit.

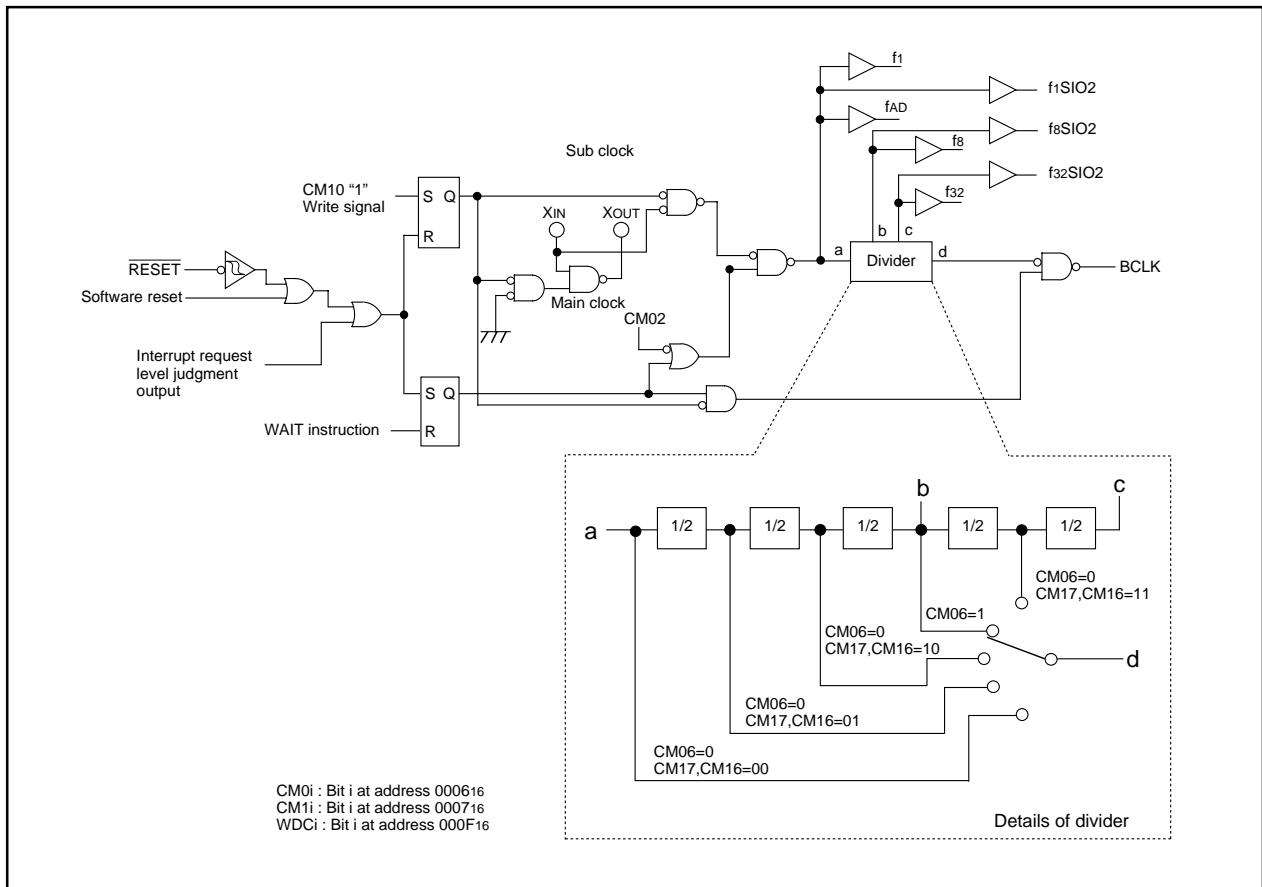


Figure 2.5.3 Clock generating circuit

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 000616).

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 XIN-XOUT drive capacity select bit (bit 5 at address 000716). 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.

**(2) BCLK**

The internal clock  $\phi$  is the clock that drives the CPU, and is 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 main clock division select bit 0 (bit 6 at address 000616) changes to "1" when shifting from high-speed/medium-speed to stop mode and at reset.

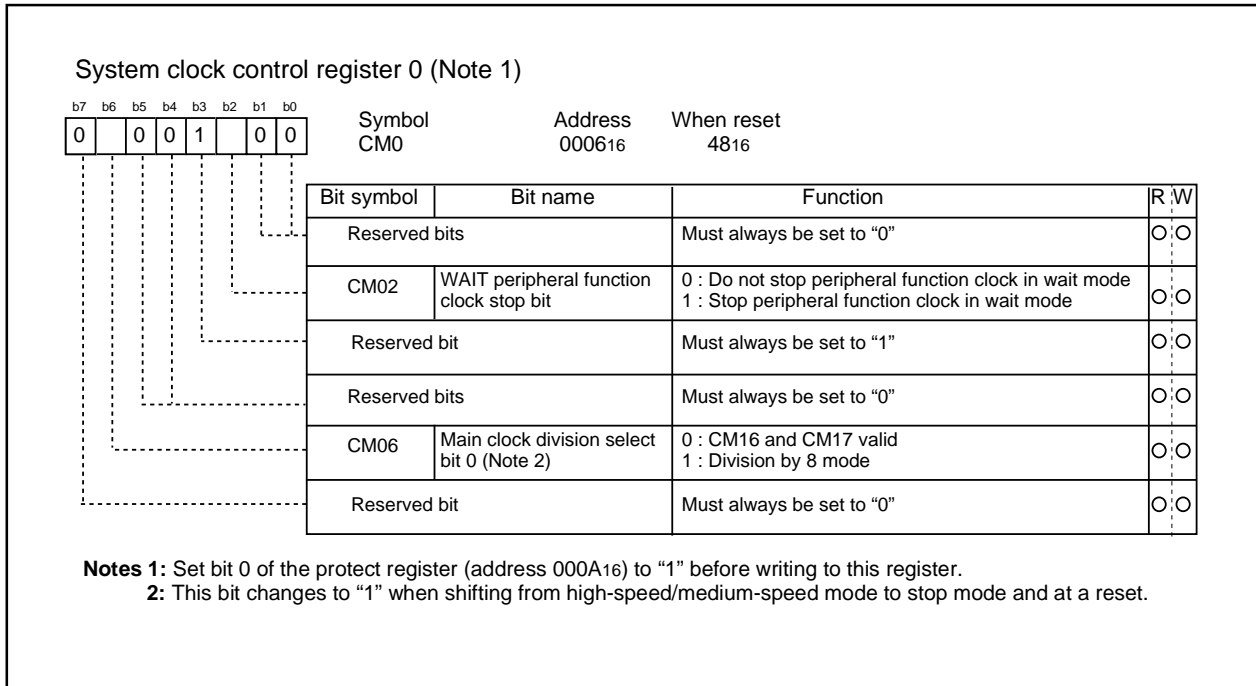
**(3) Peripheral function clock (f1, f8, f32, f1SIO2, f8SIO2, f32SIO2, fAD)**

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 000616) to "1" and then executing a WAIT instruction.

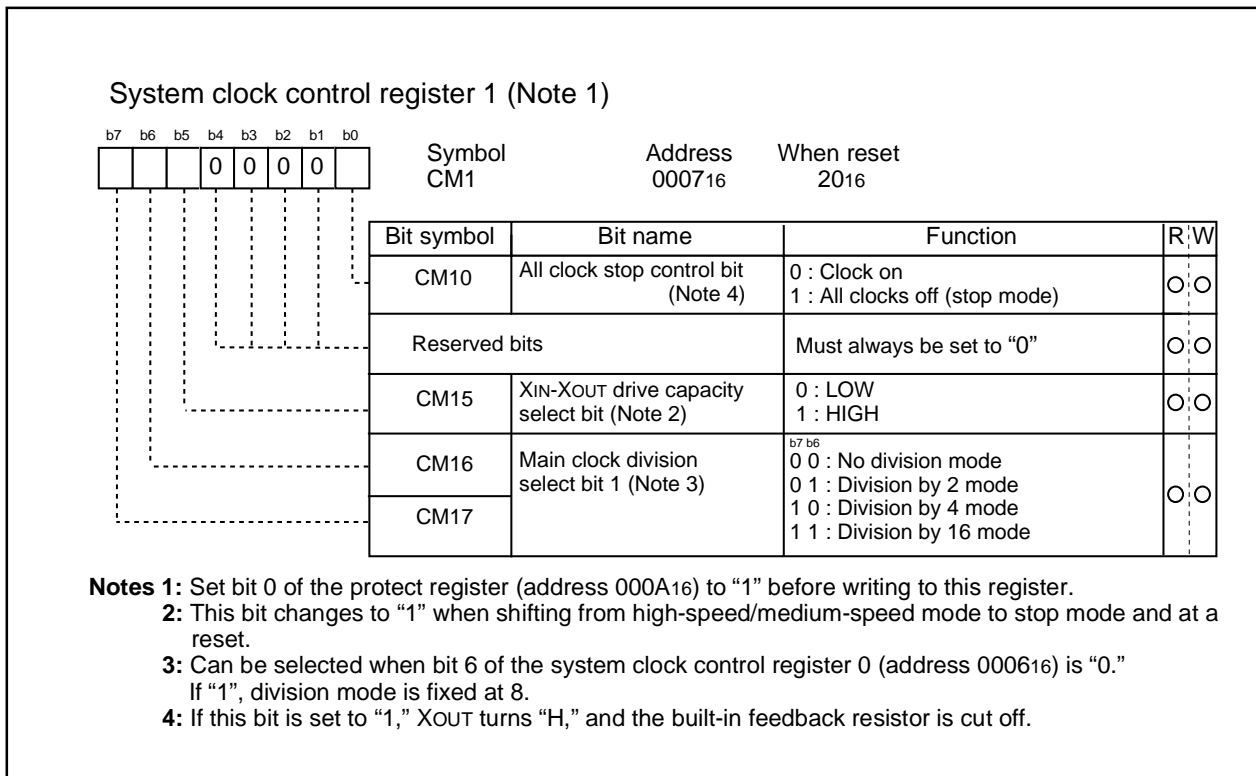
**M306V5ME-XXXSP**  
**M306V5EESP**

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

Figures 2.5.4 and 2.5.5 shows the system clock control registers 0 and 1.



**Figures 2.5.4 System clock control register 0**



**Figure 2.5.5 System clock control register 1**

### 2.5.4 Stop Mode

Writing "1" to the all-clock stop control bit (bit 0 at address 000716) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that VCC remains above 4.5V.

Because the oscillation, BCLK, f1 to f32, f1SIO2 to f32SIO2, and fAD 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 UARTi (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 000616) 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	State
Port	Retains status before stop mode

### 2.5.5 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	State
Port	Retains status before wait mode



### 2.5.6 Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. 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.

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

CM17	CM16	CM06	CM04	Operating mode of BCLK
0	1	0	Invalid	Division by 2 mode
1	0	0	Invalid	Division by 4 mode
Invalid	Invalid	1	Invalid	Division by 8 mode
1	1	0	Invalid	Division by 16 mode
0	0	0	Invalid	No-division mode

## 2.5.7 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.

#### (2) Wait mode

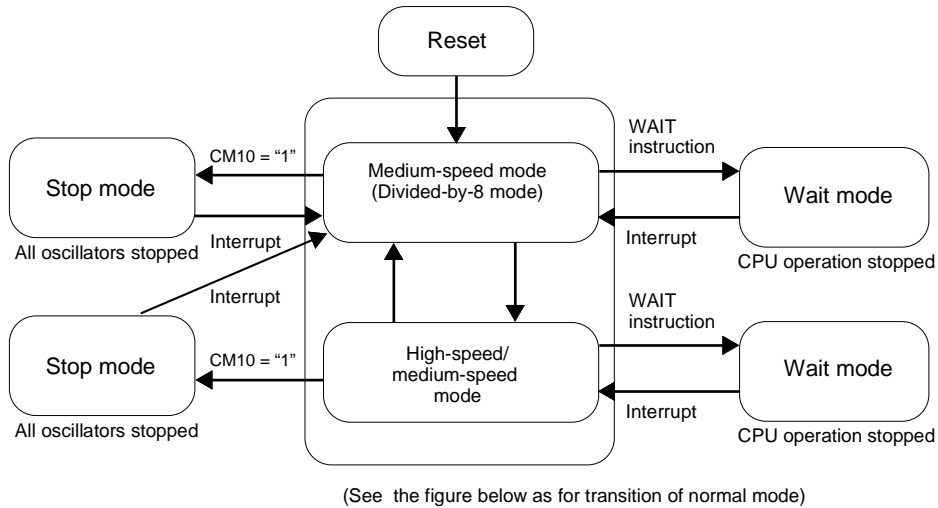
The CPU operation is stopped. The oscillators do not stop.

#### (3) Stop mode

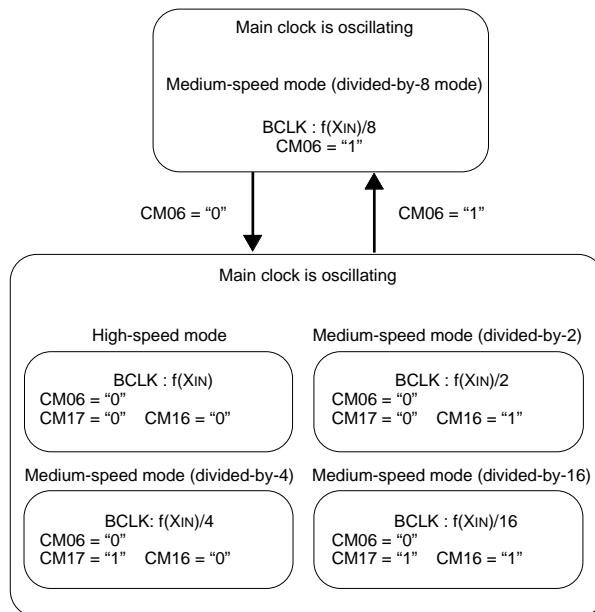
All oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in decreasing power consumption.

Figure 2.5.6 is the state transition diagram of the above modes.

**Transition of stop mode, wait mode**



**Transition of normal mode**



- Notes 1:** Switch clocks after oscillation of main clock is sufficiently stable.  
**2:** Change CM06 after changing CM17 and CM16.  
**3:** Transit in accordance with arrows.

**Figure 2.5.6 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".

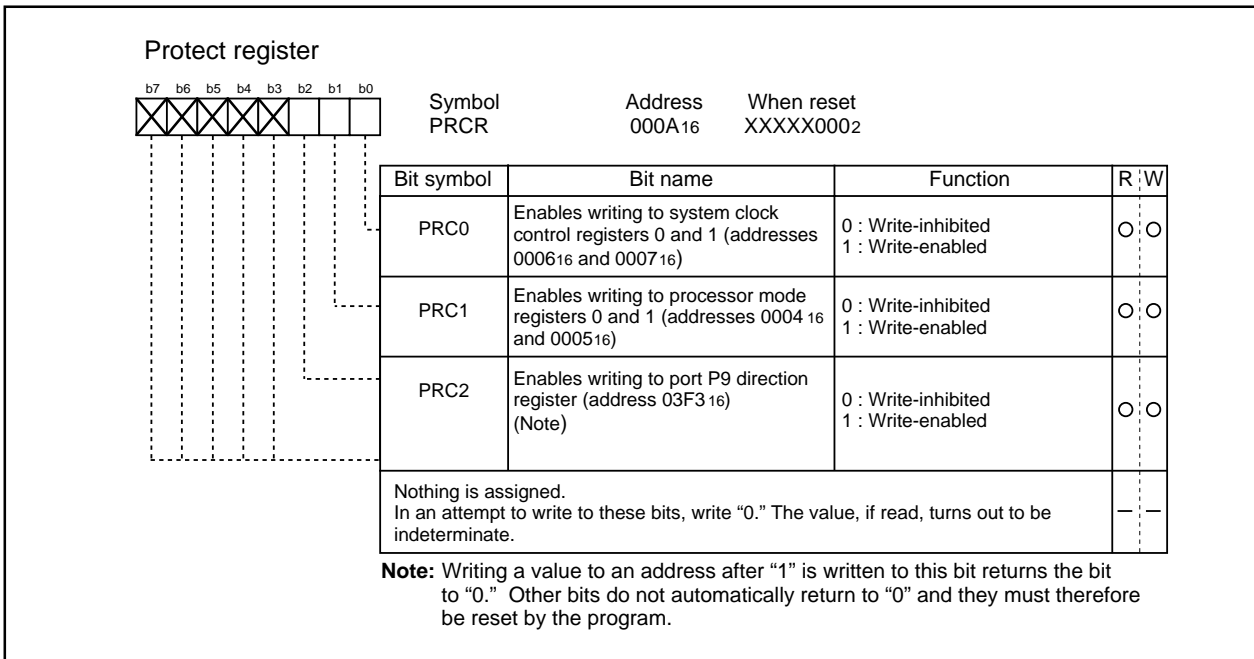
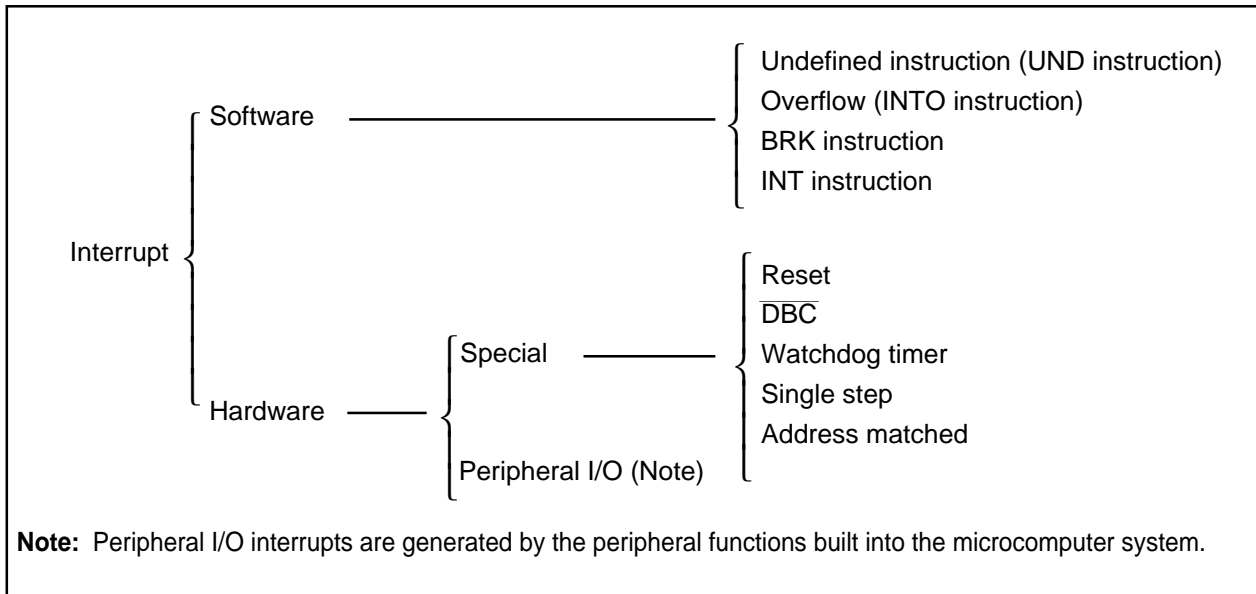


Figure 2.6.1 Protect register

## 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. If change the U flag to "0" and select the interrupt stack pointer (ISP), and then execute 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.

- **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 interrupts**

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

- **UART0 reception, UART2 reception interrupts**

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

- **Multi-master I<sup>2</sup>C-BUS interface 0 and multi-master I<sup>2</sup>C-BUS interface 1 interrupts**

This is an interrupt that the serial I/O transmission/reception is completed, or a STOP condition is detected.

- **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.

- **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.

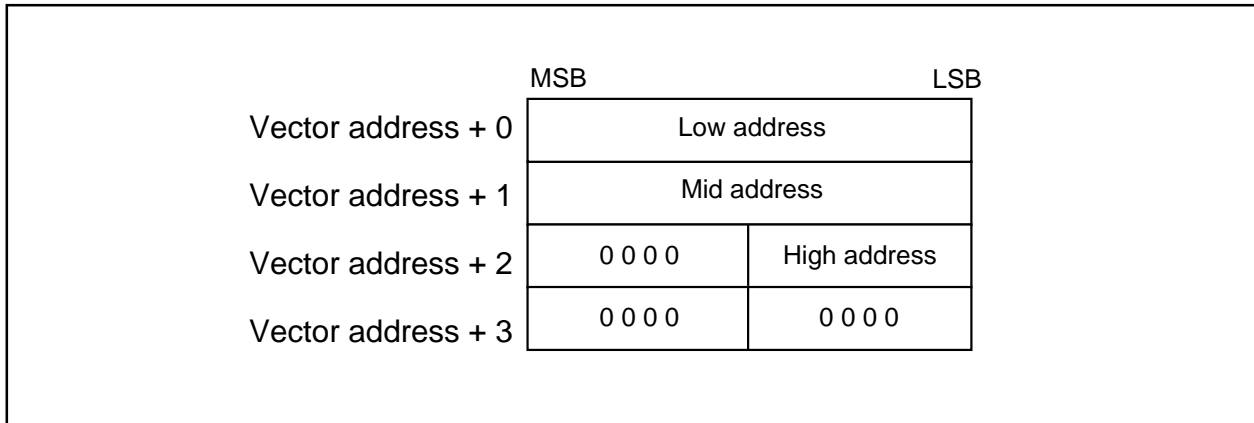


Figure 2.7.2 Format for specifying interrupt vector addresses

### (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> F	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 FFFE <sub>16</sub> B	There is an address-matching interrupt enable bit
Single step (Note)	FFFE <sub>16</sub> C to FFFE <sub>16</sub> F	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 FFFE <sub>16</sub> B	Do not use
Reset	FFFF <sub>16</sub> C to FFFF <sub>16</sub> F	

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

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**(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)	BRK instruction	Cannot be masked I flag
Software interrupt number 4	+16 to +19 (Note)	OSD1	
Software interrupt number 5	+20 to +23 (Note)	Reserved source	
Software interrupt number 6	+24 to +27 (Note)	Reserved source	
Software interrupt number 7	+28 to +31 (Note)	Reserved source	
Software interrupt number 8	+32 to +35 (Note)	OSD2	
Software interrupt number 9	+36 to +39 (Note)	Multi-master I <sup>2</sup> C-BUS interface 1	
Software interrupt number 10	+40 to +43 (Note)	Bus collision detection	
Software interrupt number 11	+44 to +47 (Note)	DMA0	
Software interrupt number 12	+48 to +51 (Note)	DMA1	
Software interrupt number 13	+52 to +55 (Note)	Multi-master I <sup>2</sup> C-BUS interface 0	
Software interrupt number 14	+56 to +59 (Note)	A-D conversion	
Software interrupt number 15	+60 to +63 (Note)	UART2 transmit	
Software interrupt number 16	+64 to +67 (Note)	UART2 receive	
Software interrupt number 17	+68 to +71 (Note)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note)	Data slicer	
Software interrupt number 20	+80 to +83 (Note)	VSYNC	
Software interrupt number 21	+84 to +87 (Note)	Timer A0	
Software interrupt number 22	+88 to +91 (Note)	Timer A1	
Software interrupt number 23	+92 to +95 (Note)	Timer A2	
Software interrupt number 24	+96 to +99 (Note)	Timer A3	
Software interrupt number 25	+100 to +103 (Note)	Timer A4	
Software interrupt number 26	+104 to +107 (Note)	Timer B0	
Software interrupt number 27	+108 to +111 (Note)	Timer B1	
Software interrupt number 28	+112 to +115 (Note)	Timer B2	
Software interrupt number 29	+116 to +119 (Note)	$\overline{\text{INT}}_0$	
Software interrupt number 30	+120 to +123 (Note)	$\overline{\text{INT}}_1$	
Software interrupt number 31	+124 to +127 (Note)	Reserved source	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note) to +252 to +255 (Note)	Software interrupt	Cannot be masked I flag

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

### 2.7.5 Interrupt Control

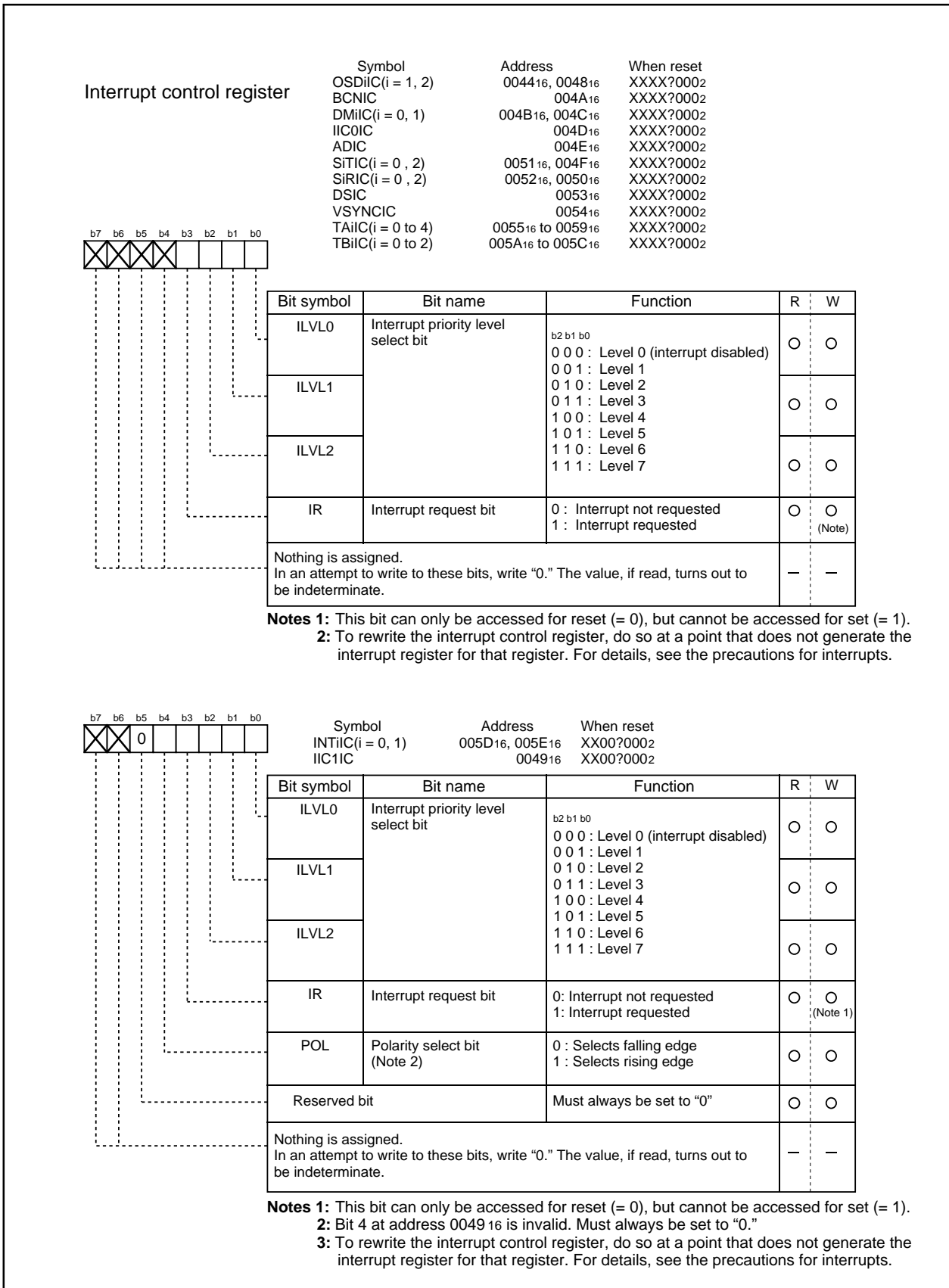
Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

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.

**M306V5ME-XXXSP**  
**M306V5EESP**

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.

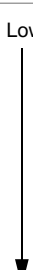
Table 2.7.3 shows the settings of interrupt priority levels and Table 2.7.4 shows the interrupt levels enabled, according to the consist of the IPL.

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
  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 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:

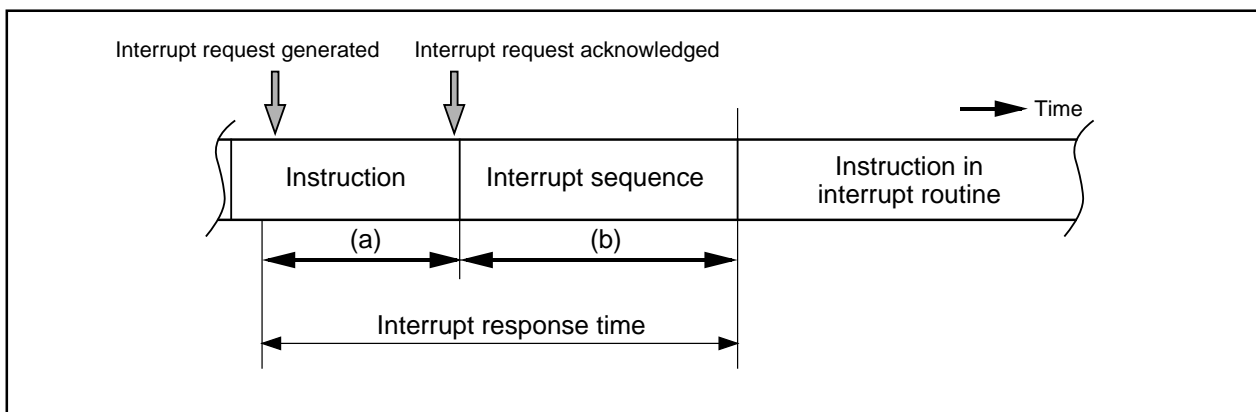
- (1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000<sub>16</sub>.
- (2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
- (3) Sets 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

**M306V5ME-XXXSP**  
**M306V5EESP**

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).

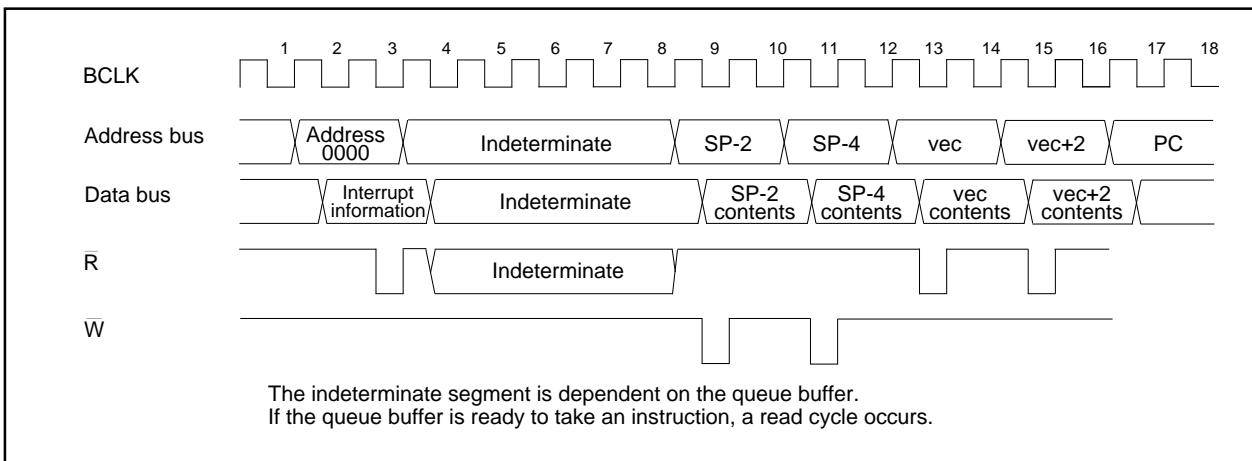
Time (b) is as shown in Table 2.7.5.

**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

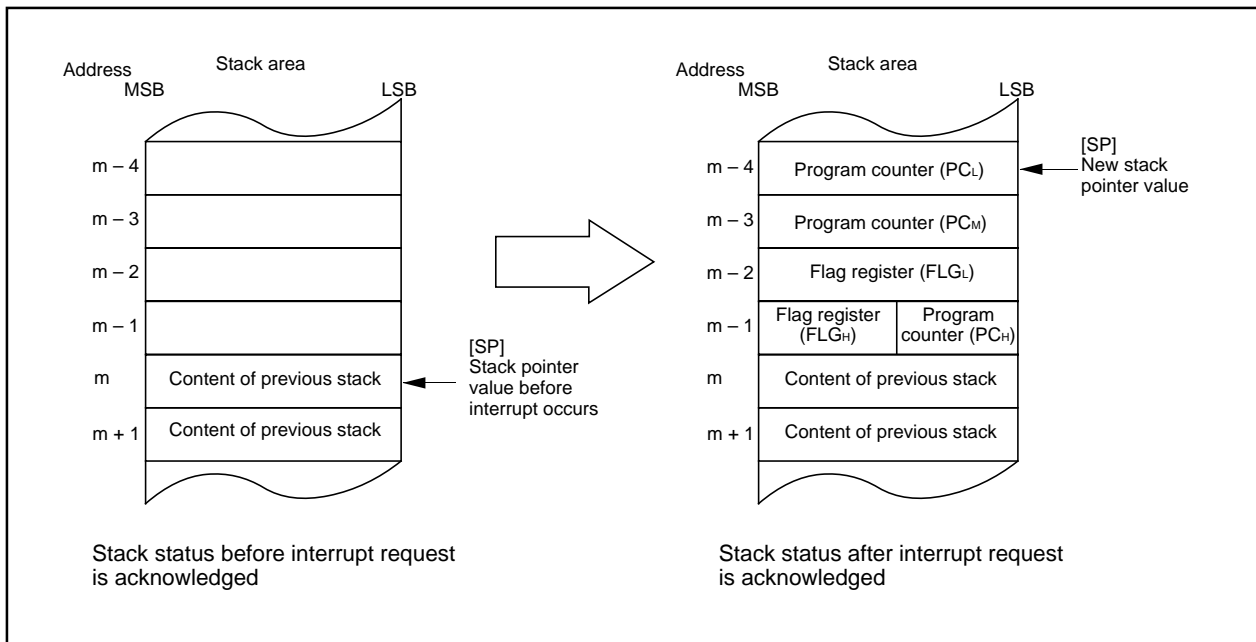


### 2.7.13 Saving Registers

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).



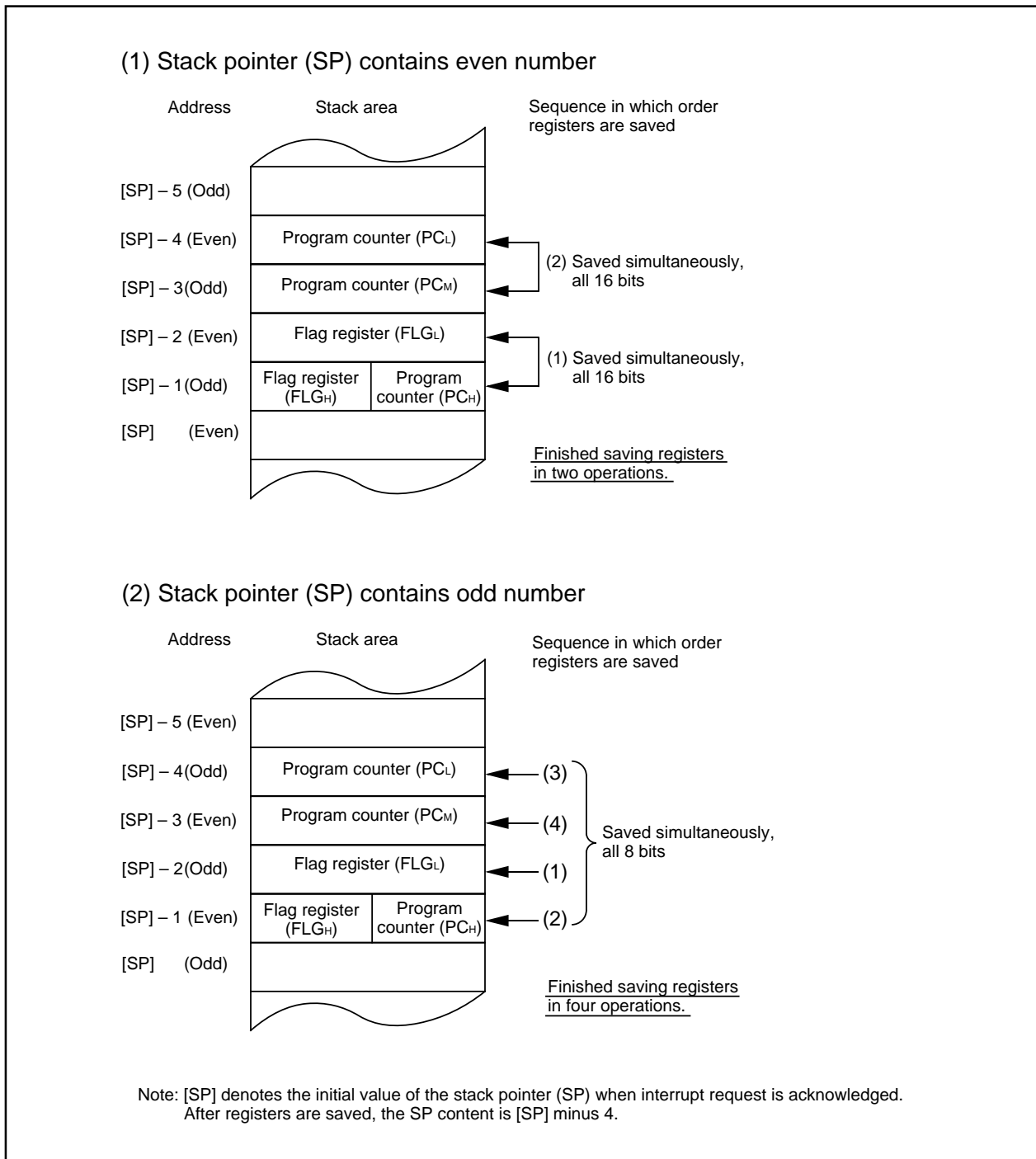
**Figure 2.7.6 State of stack before and after acceptance of interrupt request**

**M306V5ME-XXXSP**  
**M306V5EESP**

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.

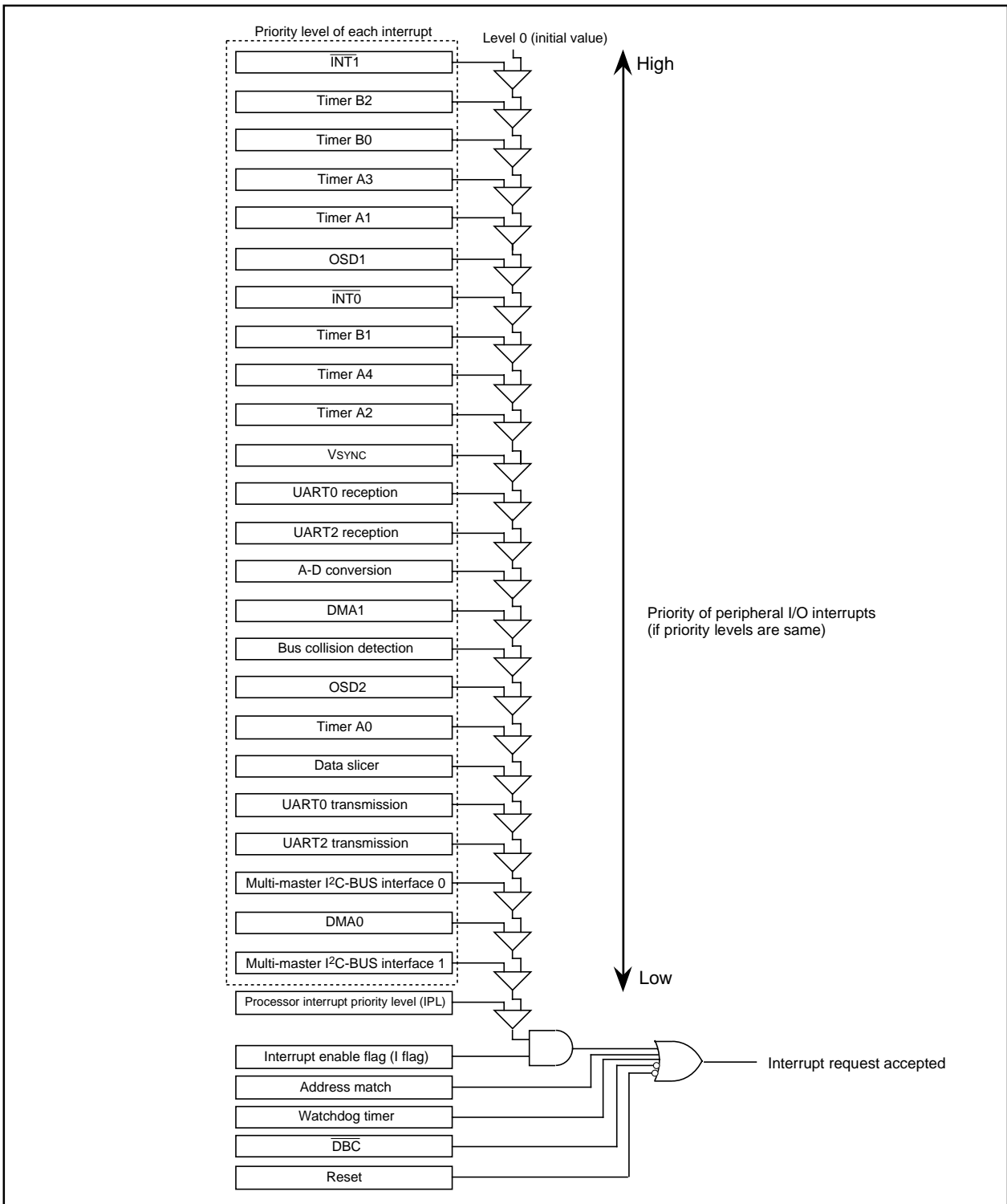
### 2.7.16 Interrupt priority level resolution circuit

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**



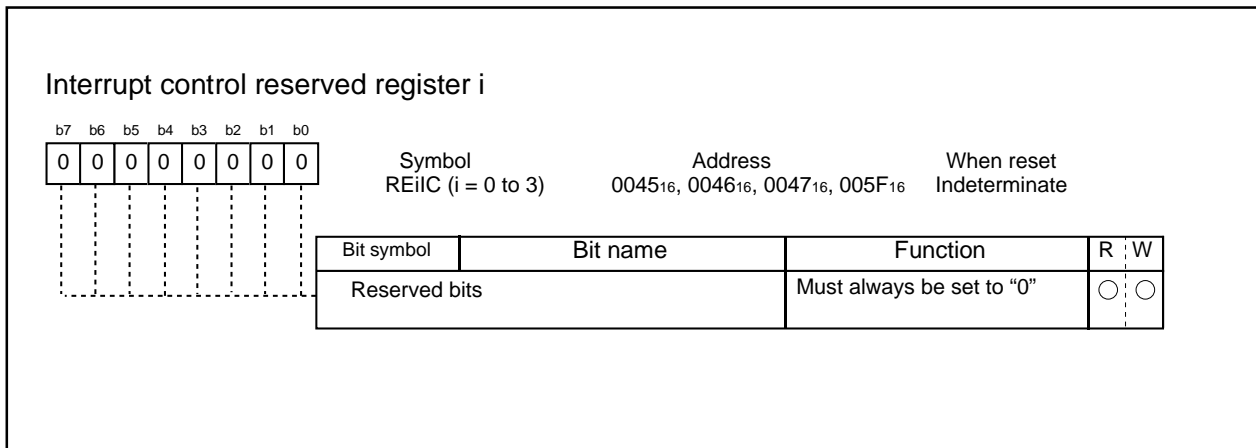
**Figure 2.7.9 Maskable interrupts priorities (peripheral I/O interrupts)**

**2.7.17 INT Interrupt**

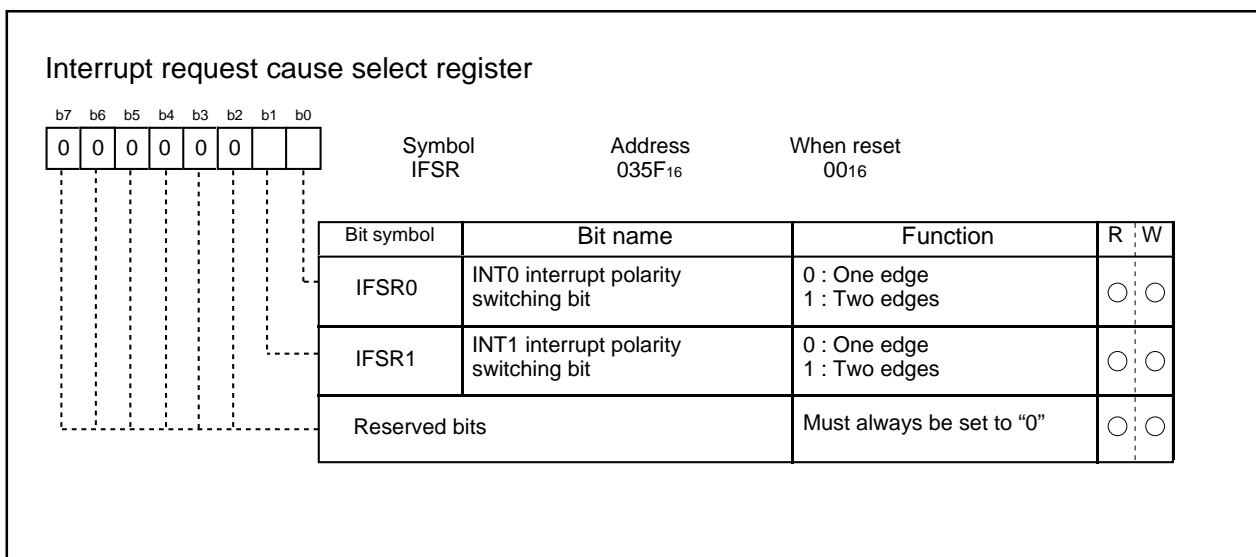
INT0 and INT1 are triggered by the edges of external inputs. The edge polarity is selected using the polarity select bit.

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 (035F16). 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.



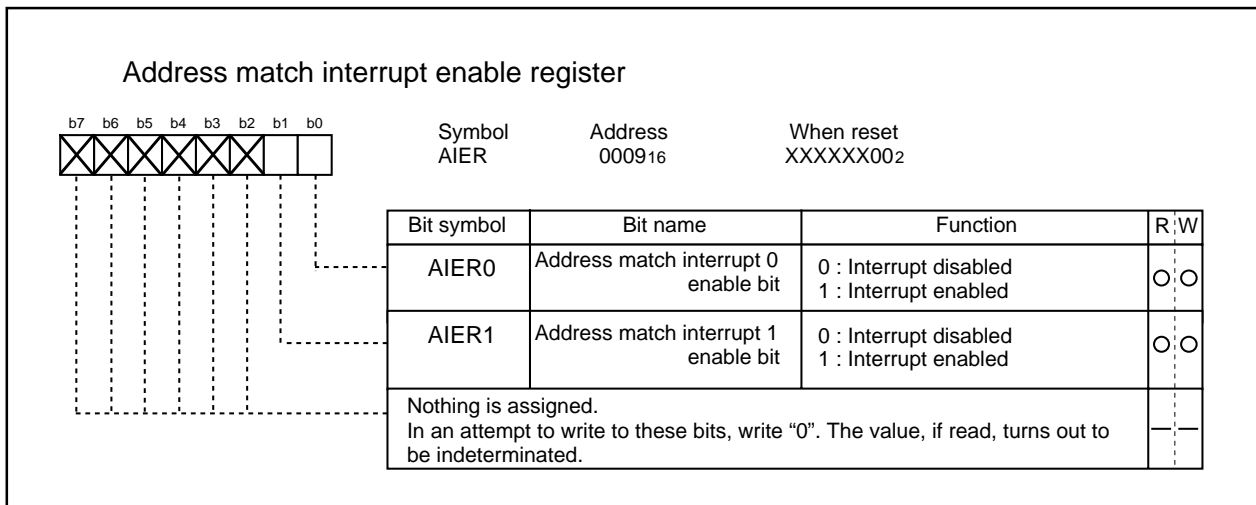
**Figure 2.7.10** Interrupt control reserved register i (i = 0 to 3)



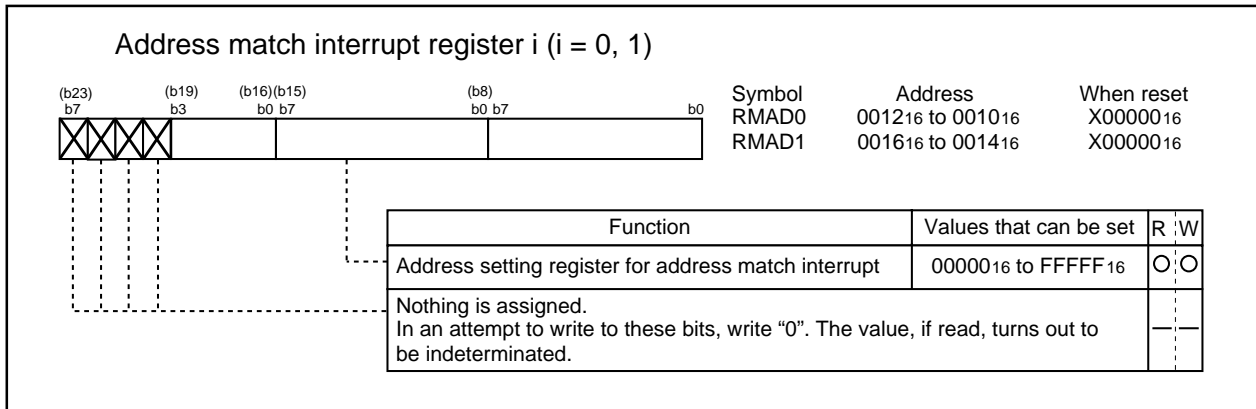
**Figure 2.7.11** Interrupt request cause select register

**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.

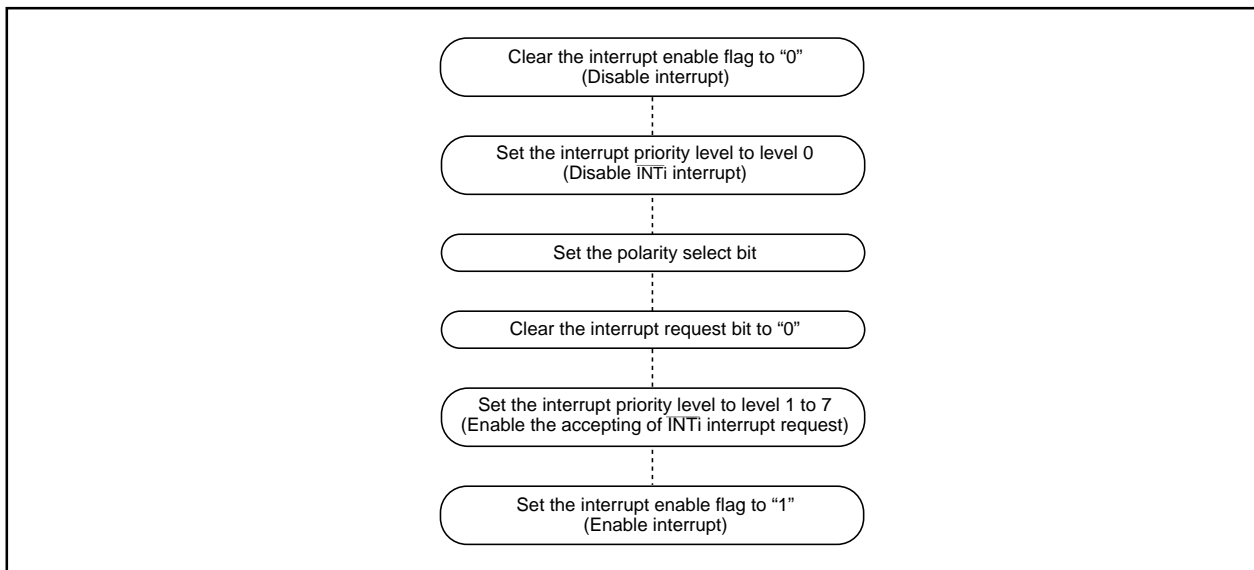


Figure 2.7.14 Switching condition of INT interrupt request

**(4) 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
  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 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. Bit 7 of the watchdog timer control register (address 000F16) selects the prescaler division ratio (by 16 or by 128). 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.

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

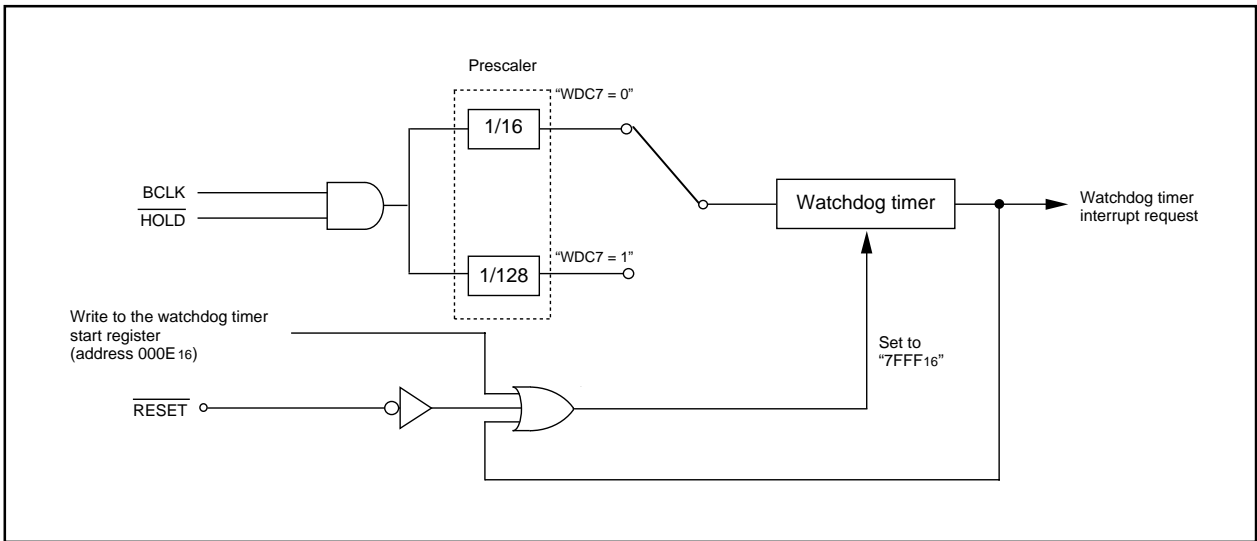
For example suppose that BCLK runs at 10 MHz and that 16 has been chosen for the dividing ratio of the pre-scaler, then the watchdog timer's period becomes approximately 52.4 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address 000E16) 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 000E16).

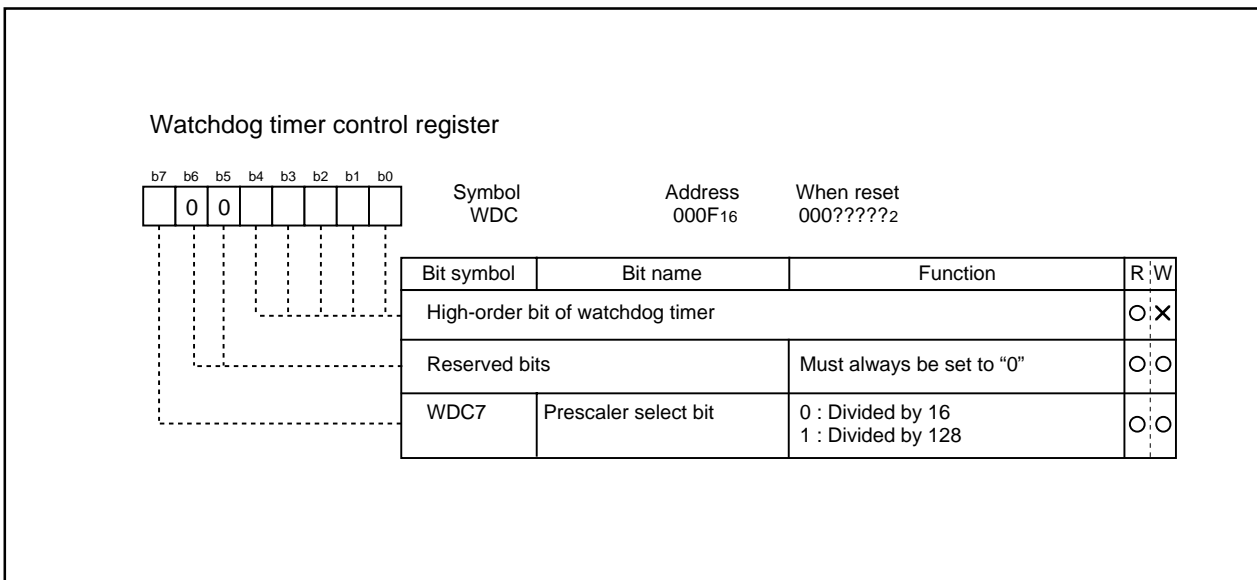
Figure 2.8.1 shows the block diagram of the watchdog timer. Figure 2.8.2 shows the watchdog timer control register and Figure 2.8.3 shows the watchdog timer start register.

**M306V5ME-XXXSP**  
**M306V5EESP**

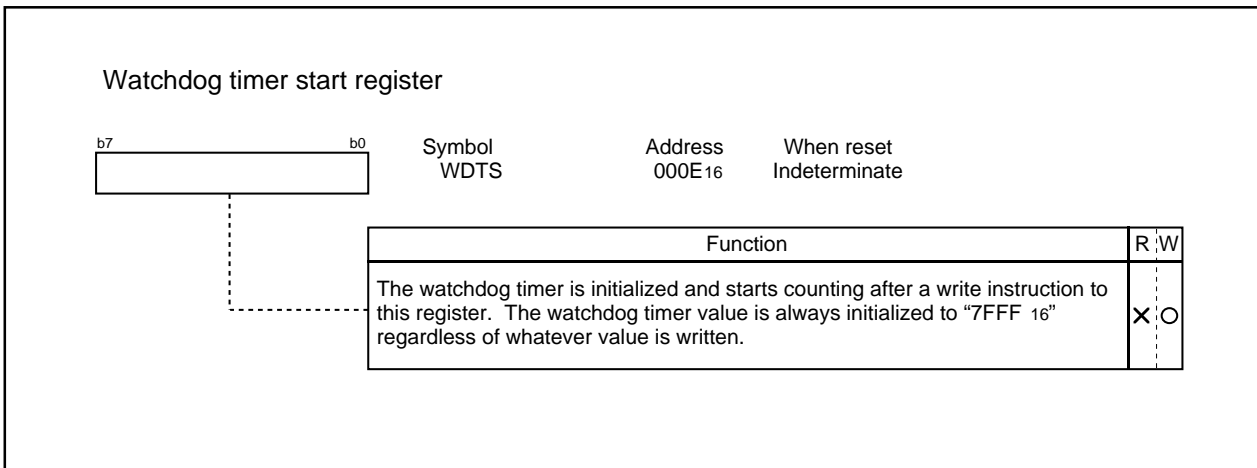
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.8.1 Block diagram of watchdog timer**



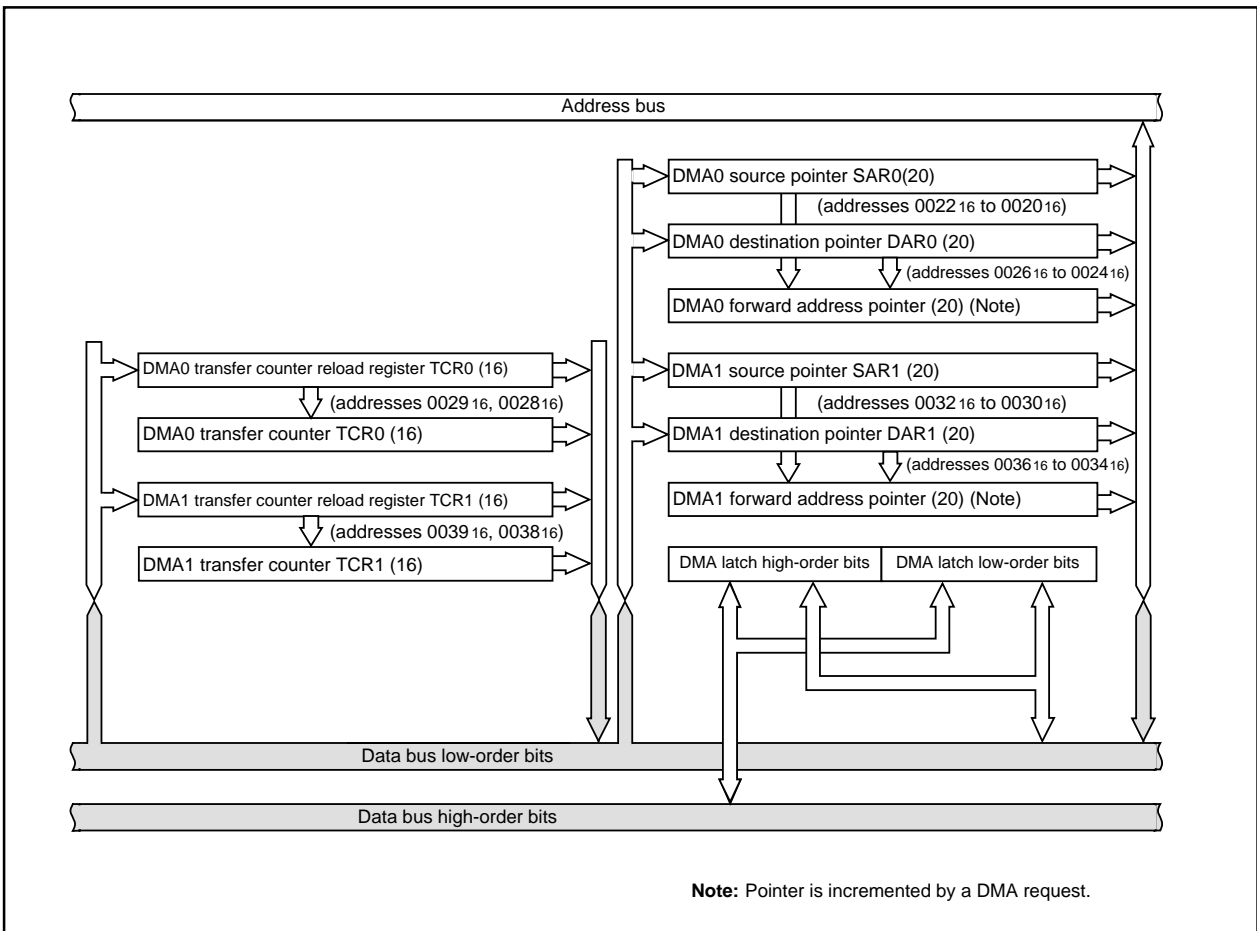
**Figure 2.8.2 Watchdog timer control register**



**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.

# M306V5ME-XXXSP

## M306V5EESP

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

**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 Multi-master I <sup>2</sup> C-BUS interface 0 interrupt request Multi-master I <sup>2</sup> C-BUS interface 1 interrupt request A-D conversion interrupt request 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.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

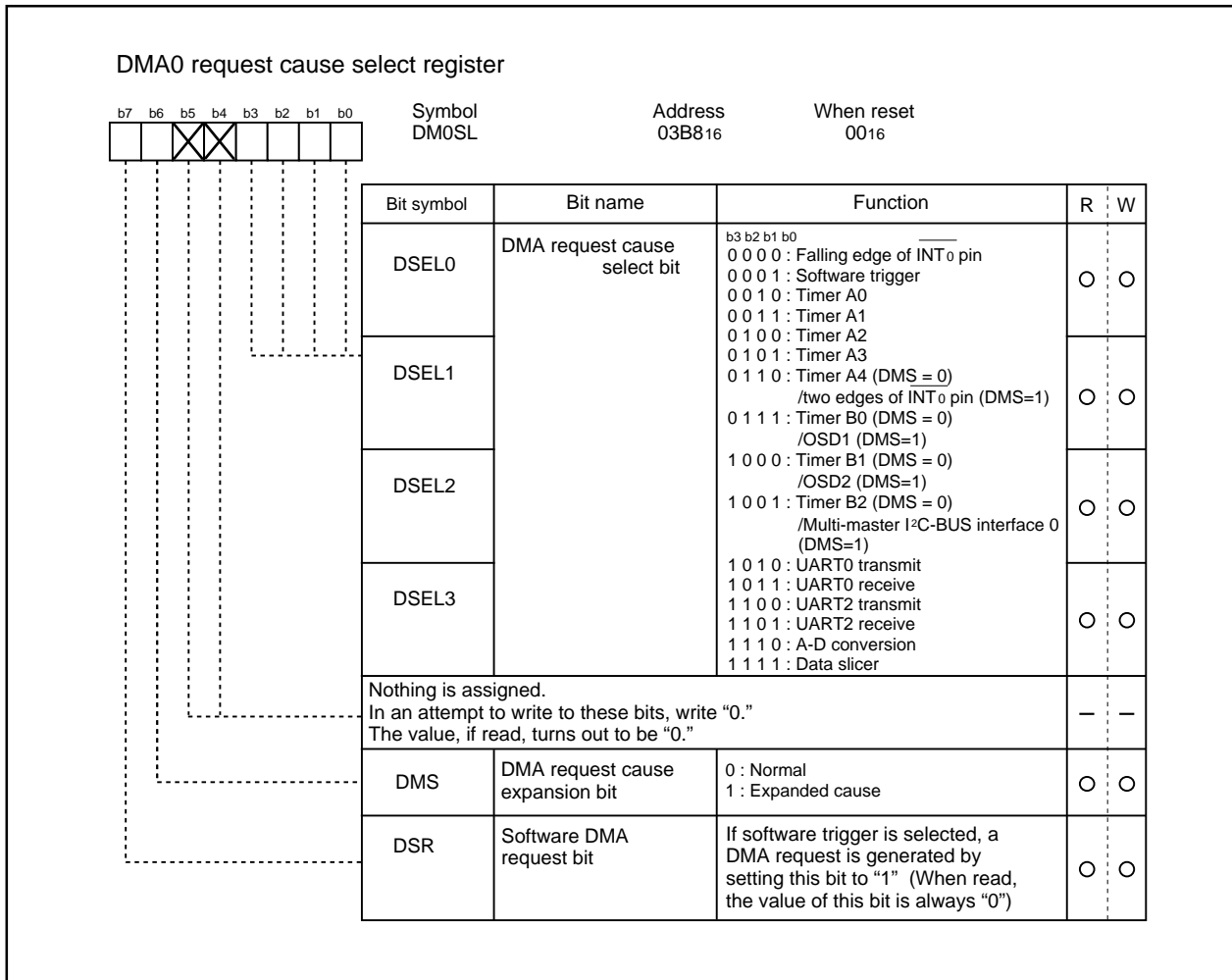
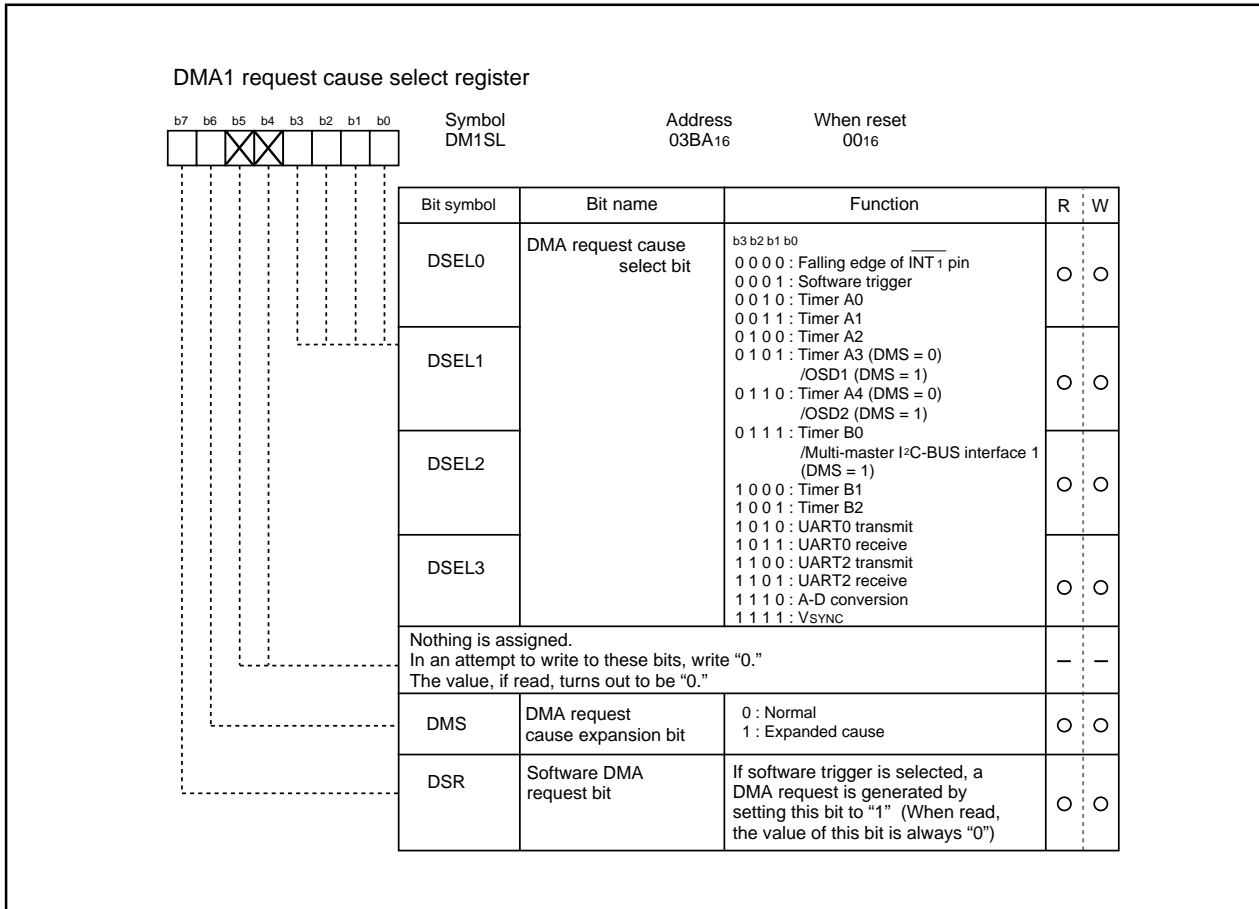


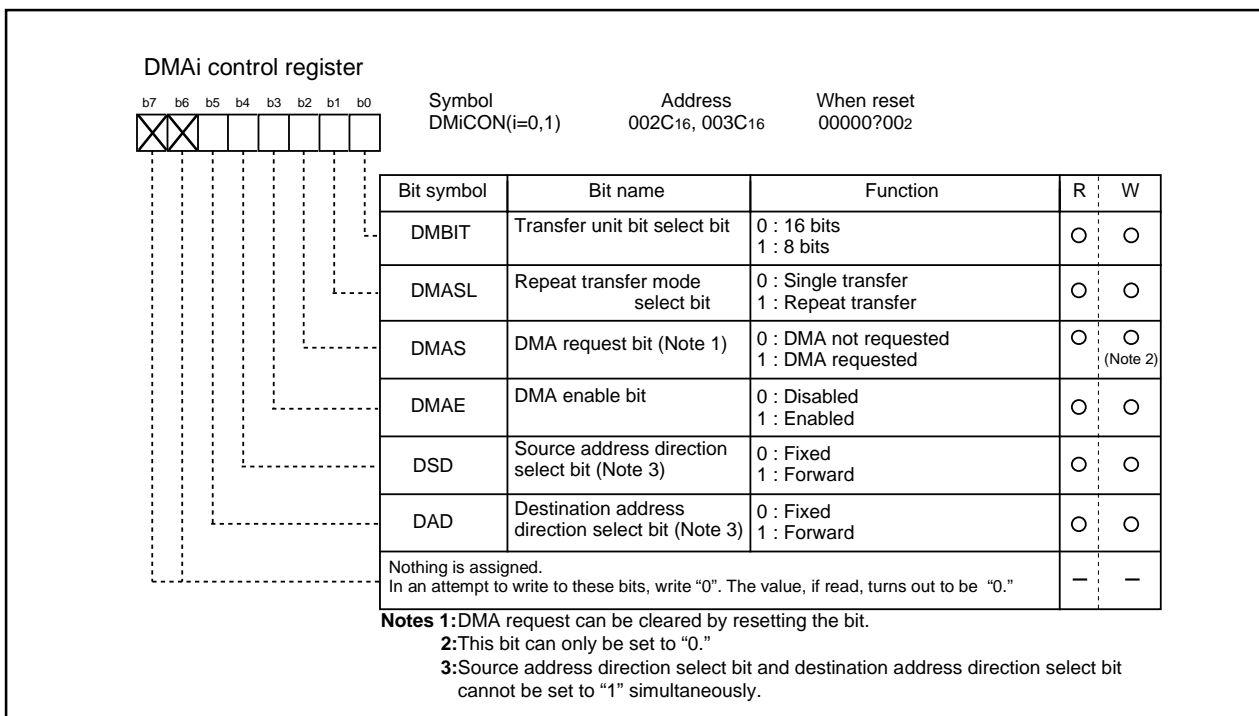
Figure 2.9.2 DMA0 request cause select register

**M306V5ME-XXXSP**  
**M306V5EESP**

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



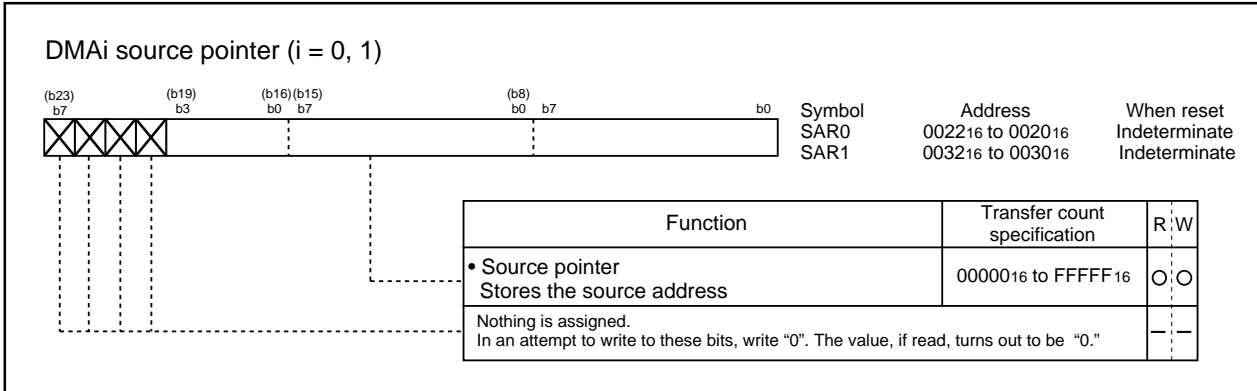
**Figure 2.9.3 DMA1 request cause select register**



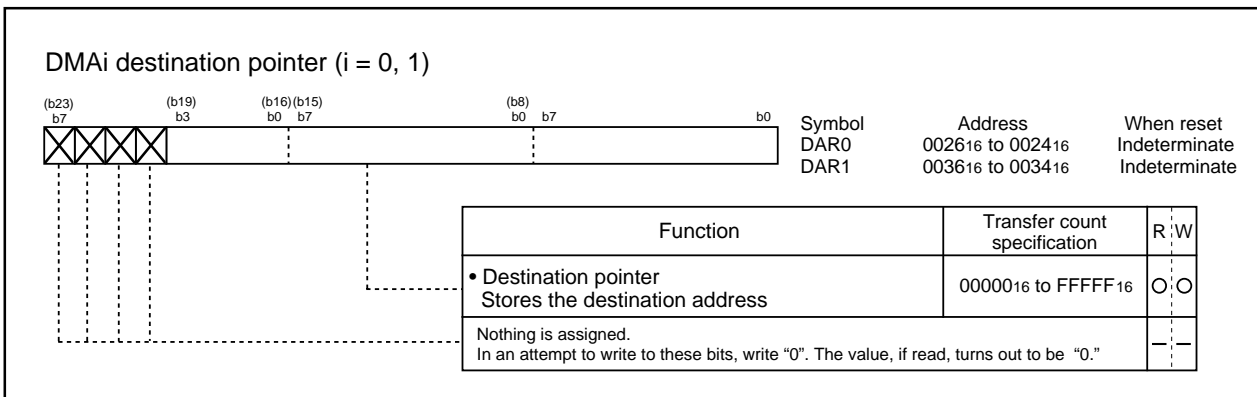
**Figure 2.9.4 DMA<sub>i</sub> control register (i = 0, 1)**

**M306V5ME-XXXSP**  
**M306V5EESP**

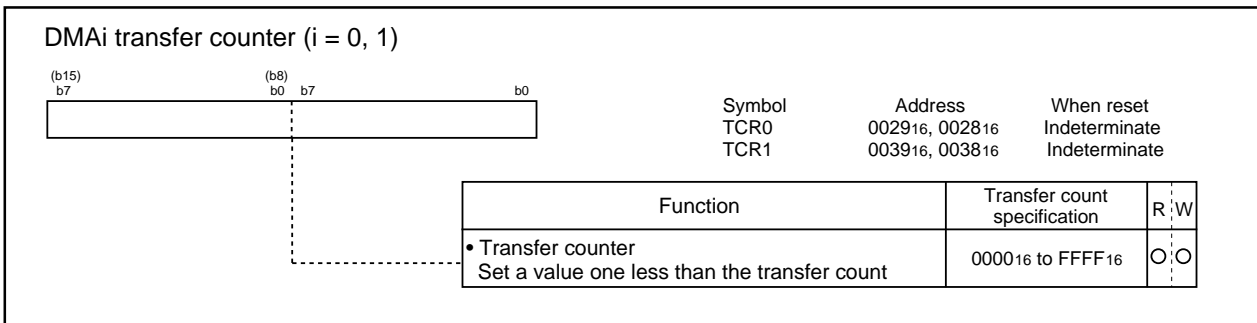
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**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. 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 software wait

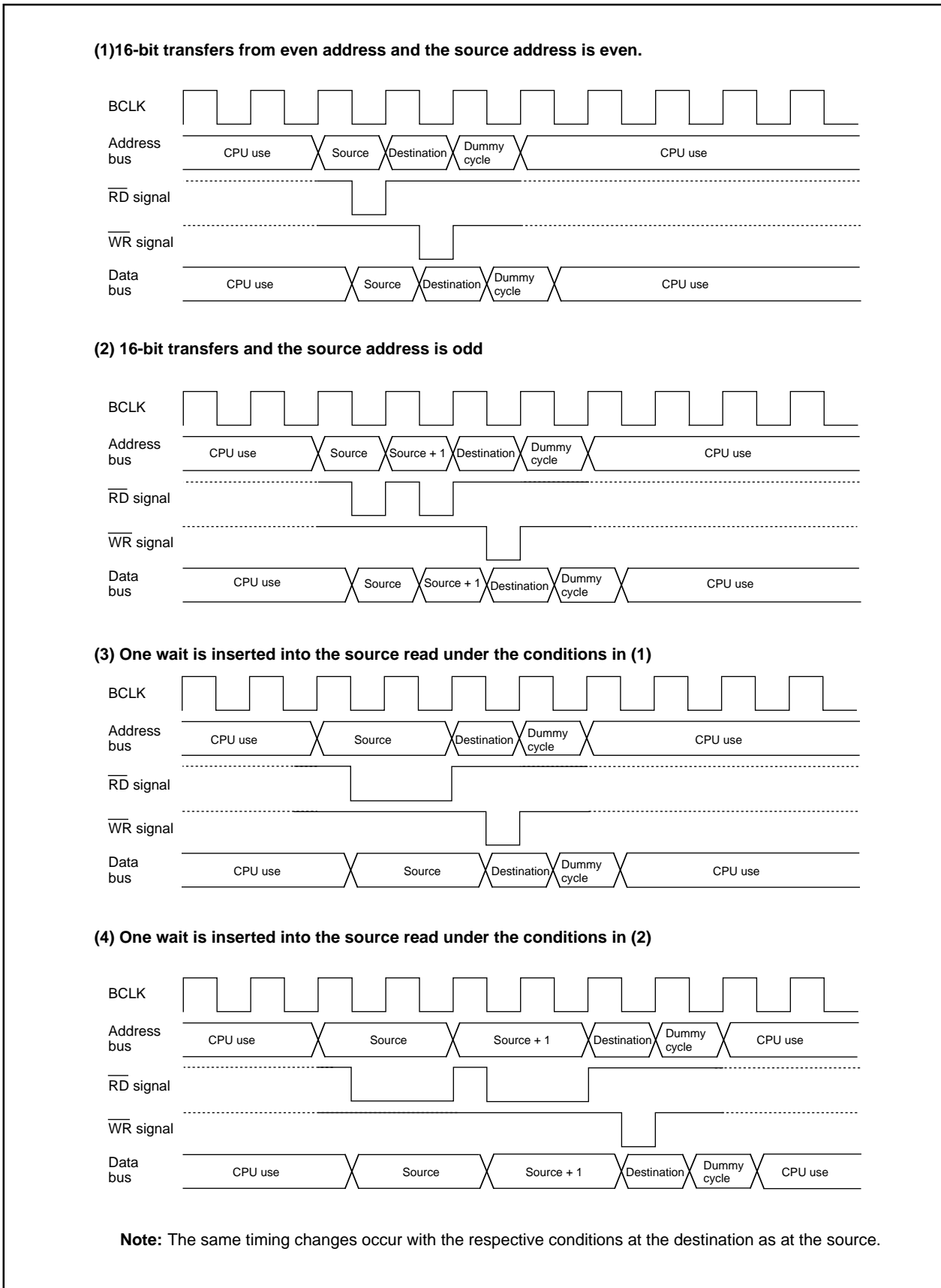
When the SFR area or a memory area with a software wait is accessed, the number of cycles is increased for the wait by 1 bus cycle. The length of the cycle is determined by BCLK.

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.



**M306V5ME-XXXSP**  
**M306V5EESP**

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



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

**M306V5ME-XXXSP**  
**M306V5EESP**

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	
			No. of read cycles	No. of write cycles
8-bit transfers (DMBIT= "1")	16-bit	Even	1	1
		Odd	1	1
16-bit transfers (DMBIT= "0")	16-bit	Even	1	1
		Odd	2	2

**Coefficient j, k**

Internal memory		
Internal ROM/RAM	Internal ROM/RAM	SFR area /OSD RAM
No wait	With wait	No wait
1	2	2

### 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.

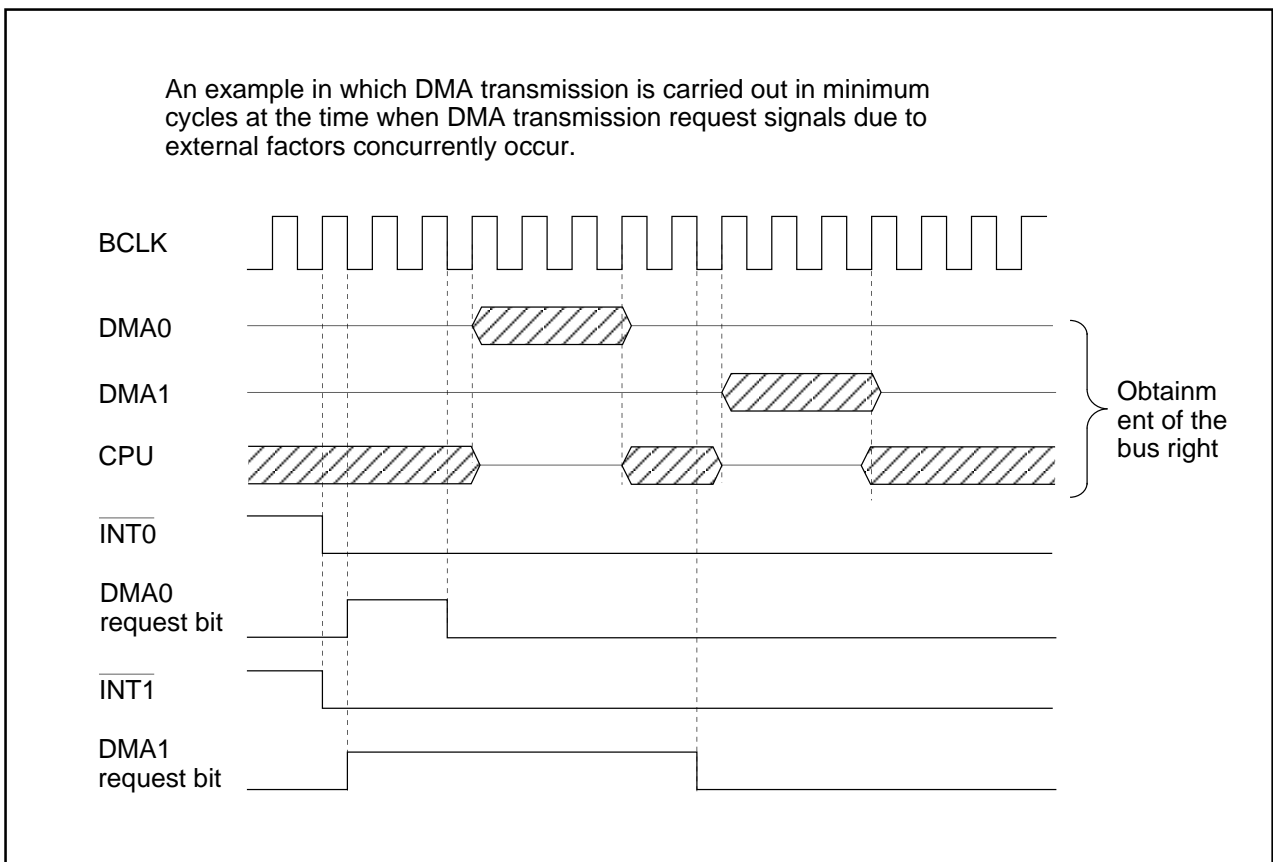
The timing for the DMA request bit to turn to 1 when an external factor is selected synchronizes with the signal's edge applicable to the function specified by the DMA request factor selection bit (synchronizes with the trailing edge of the input signal to each  $\overline{INTi}$  pin, for example).

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.

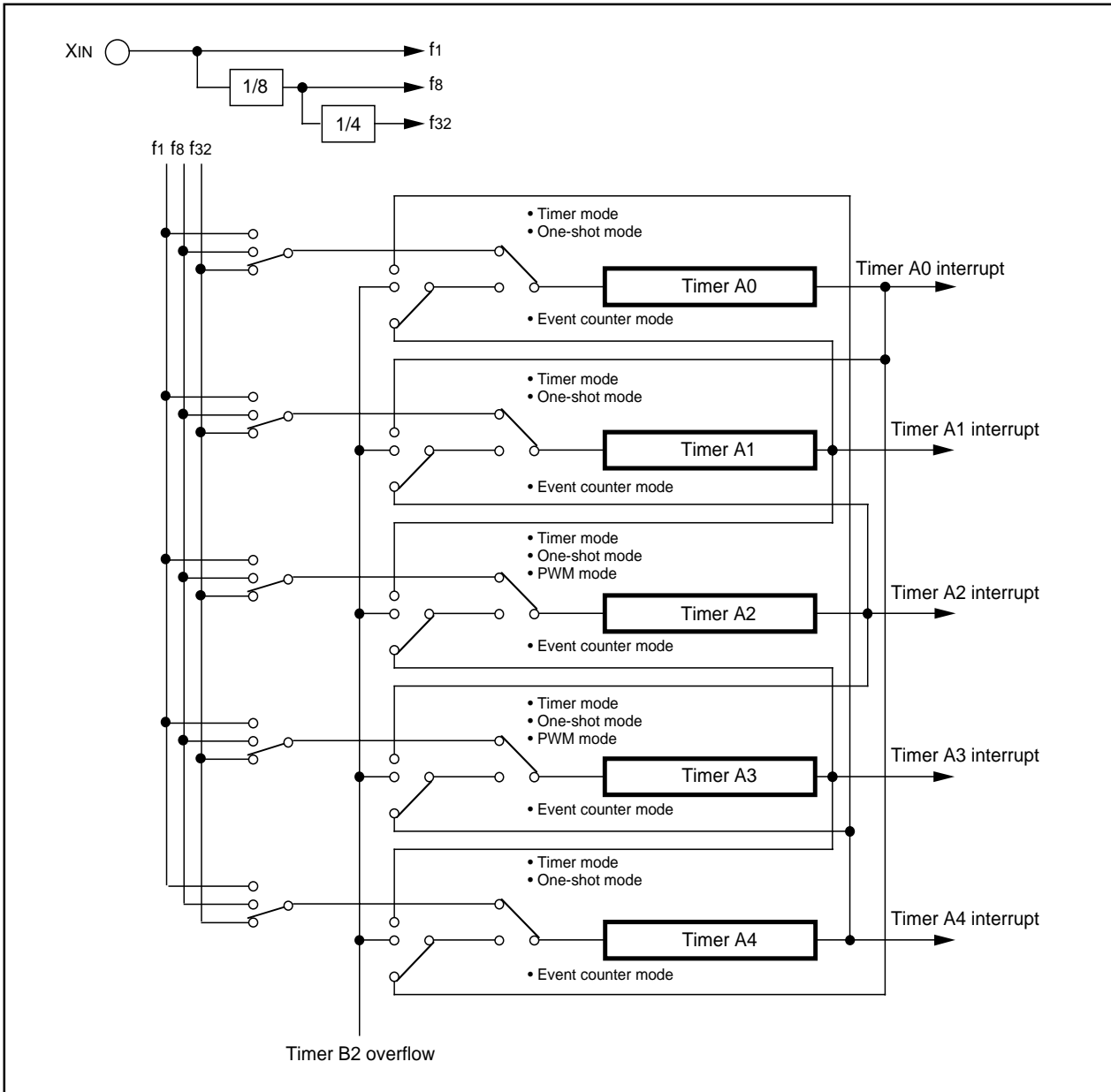


Figure 2.10.1 Timer A block diagram

**M306V5ME-XXXSP**  
**M306V5EESP**

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

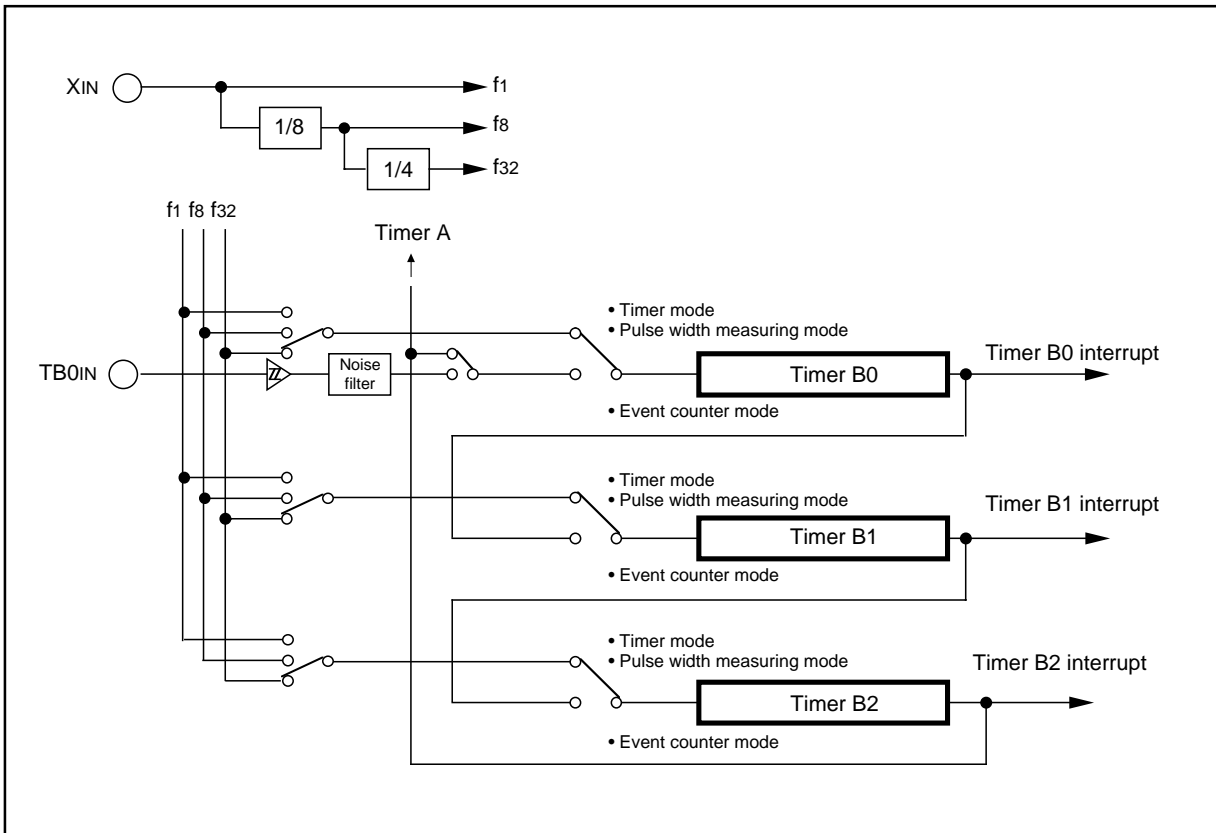


Figure 2.10.2 Timer B block diagram

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.

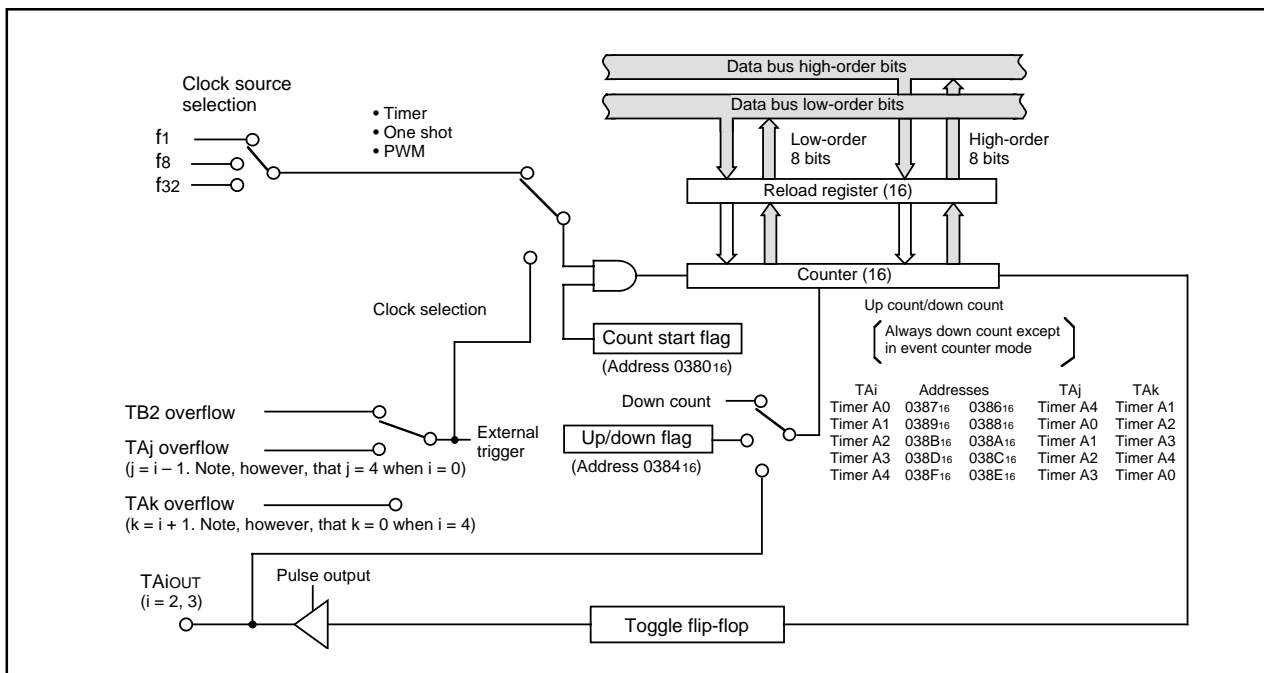


Figure 2.10.3 Block diagram of timer A

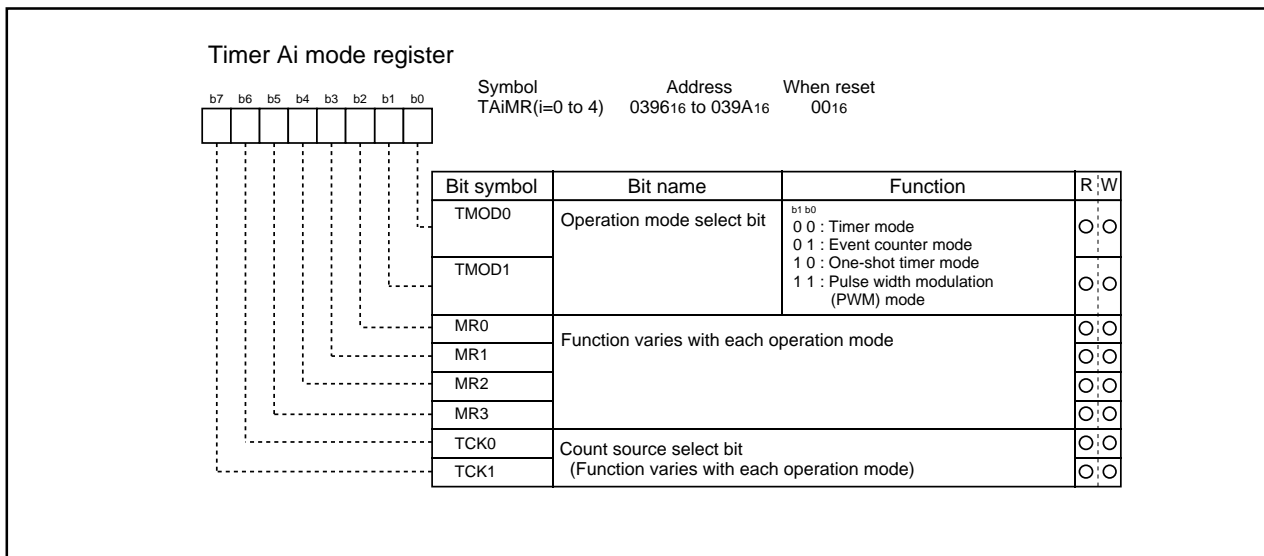


Figure 2.10.4 Timer Ai mode register (i = 0 to 4)

**M306V5ME-XXXSP**  
**M306V5EESP**

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

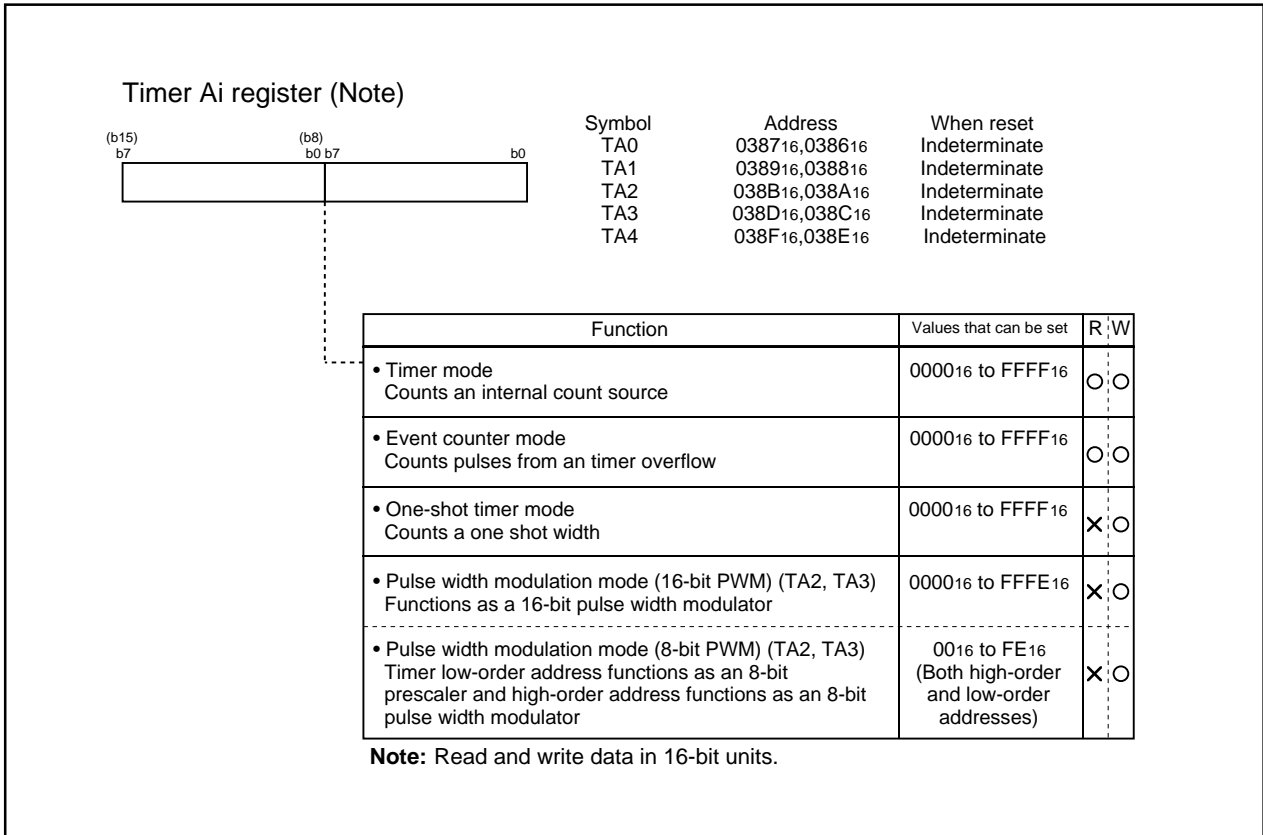


Figure 2.10.5 Timer Ai register (i = 0 to 4)

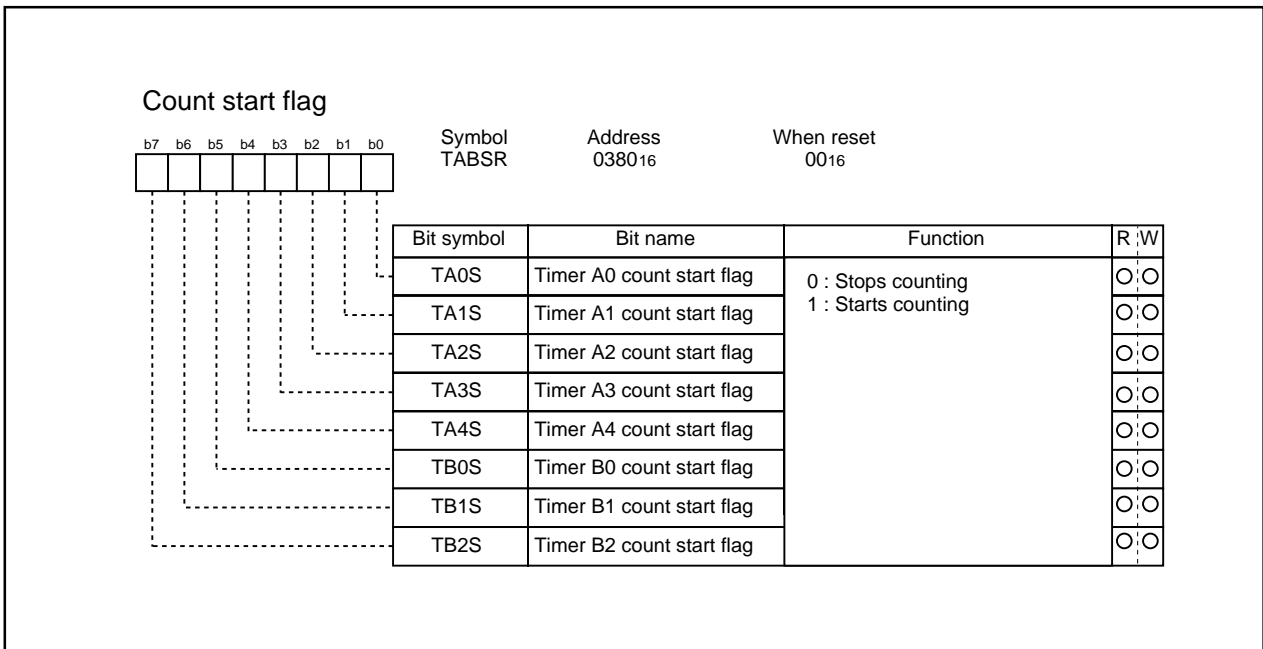


Figure 2.10.6 Count start flag



**M306V5ME-XXXSP**  
**M306V5EESP**

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

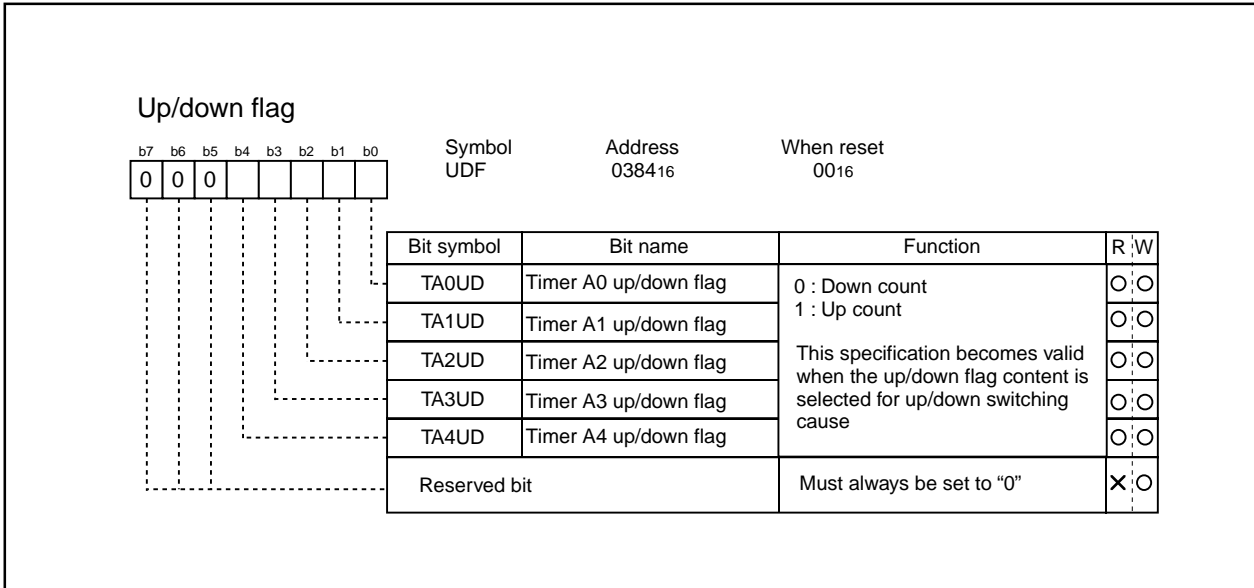


Figure 2.10.7 Up/down flag

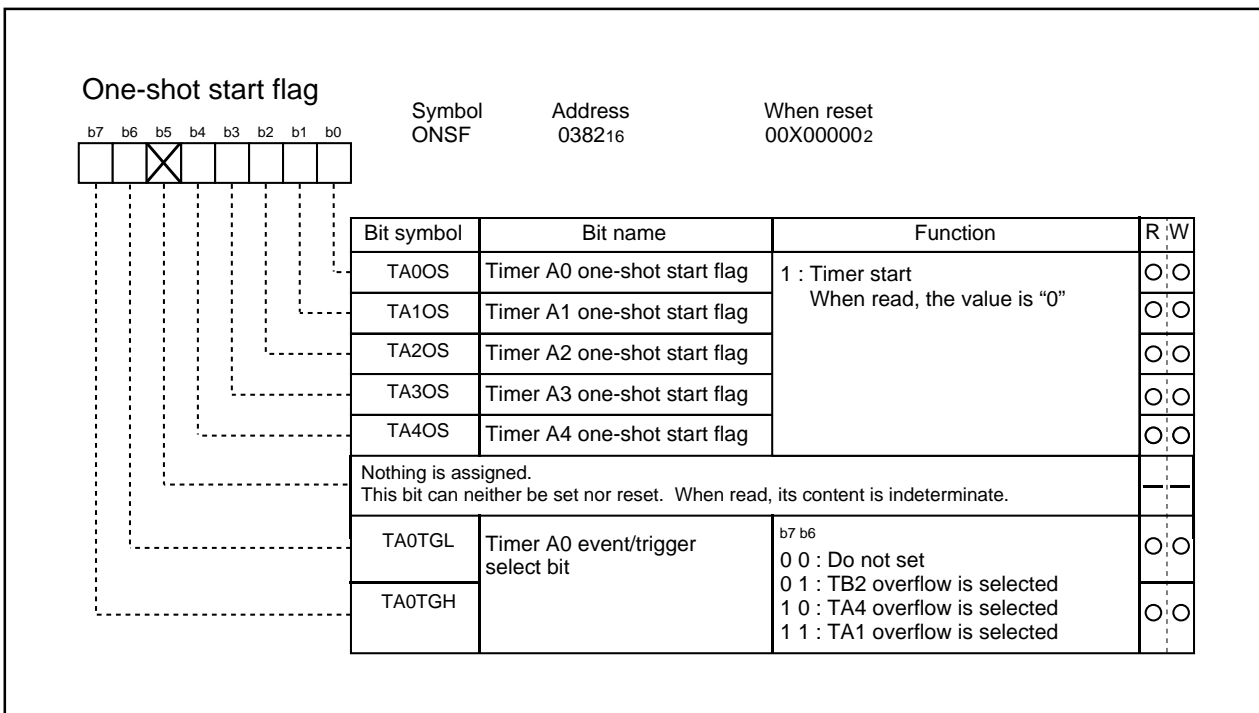


Figure 2.10.8 One-shot start flag

**M306V5ME-XXXSP**  
**M306V5EESP**

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

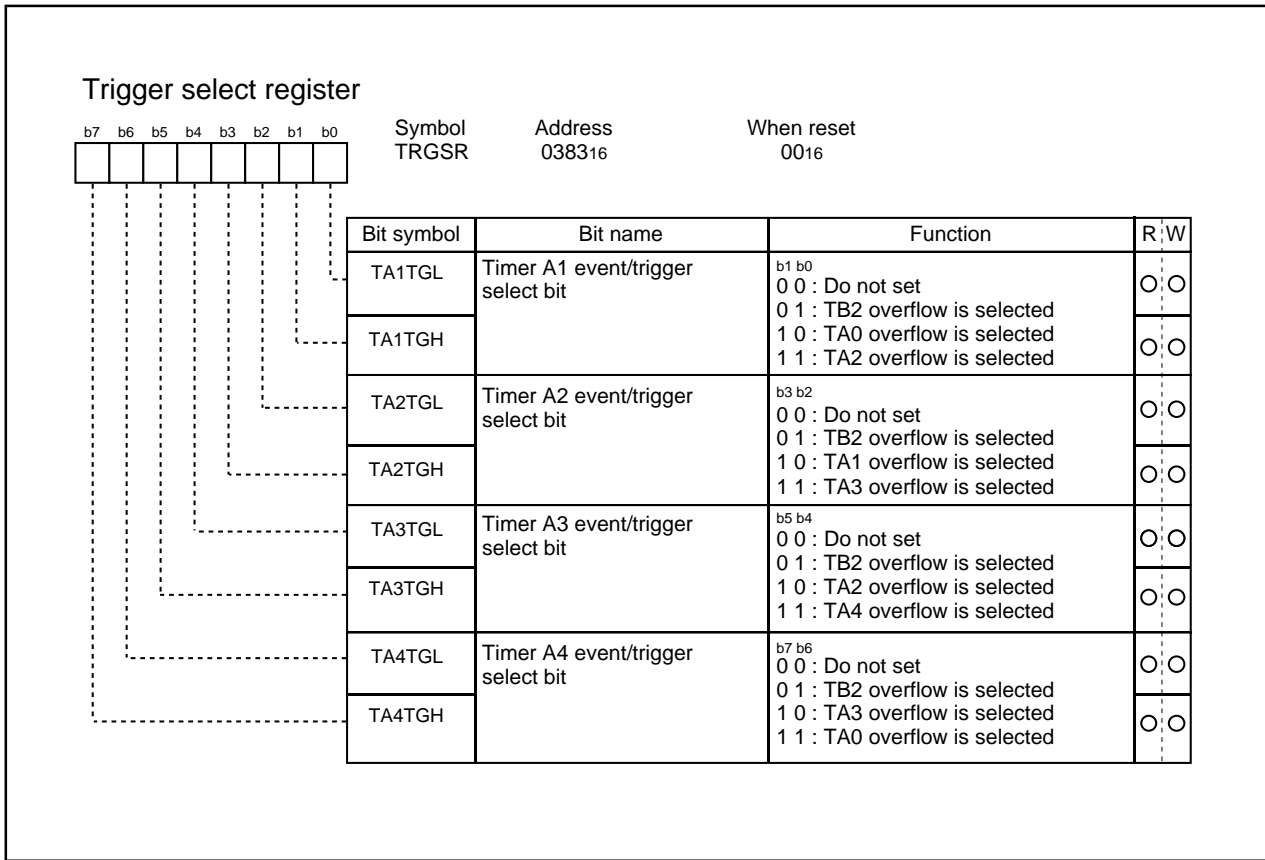


Figure 2.10.9 Trigger select register

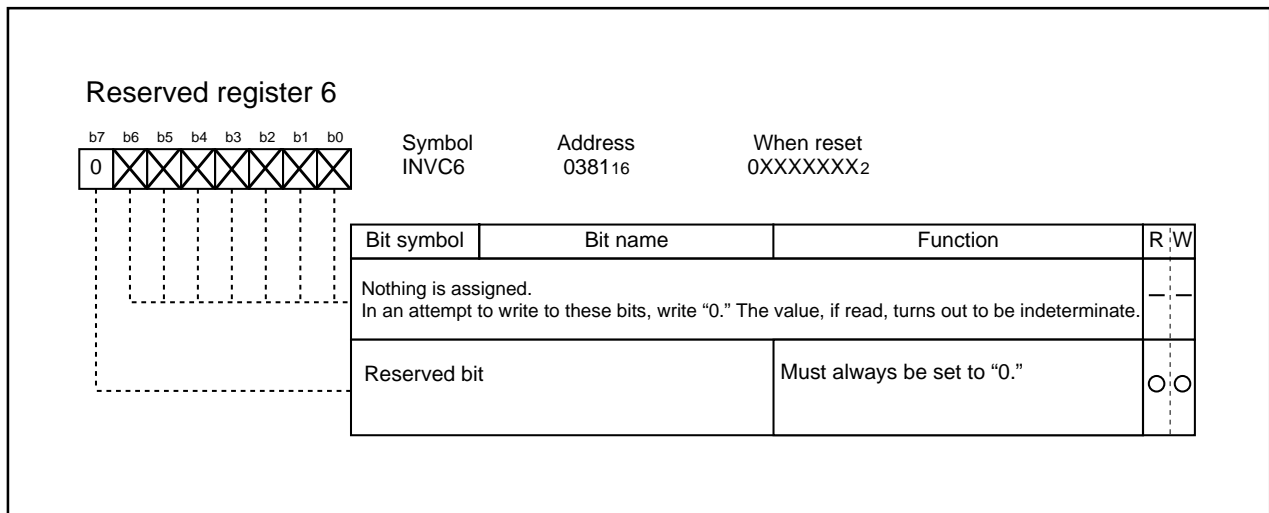


Figure 2.10.10 Reserved register 6

**M306V5ME-XXXSP**  
**M306V5EESP**

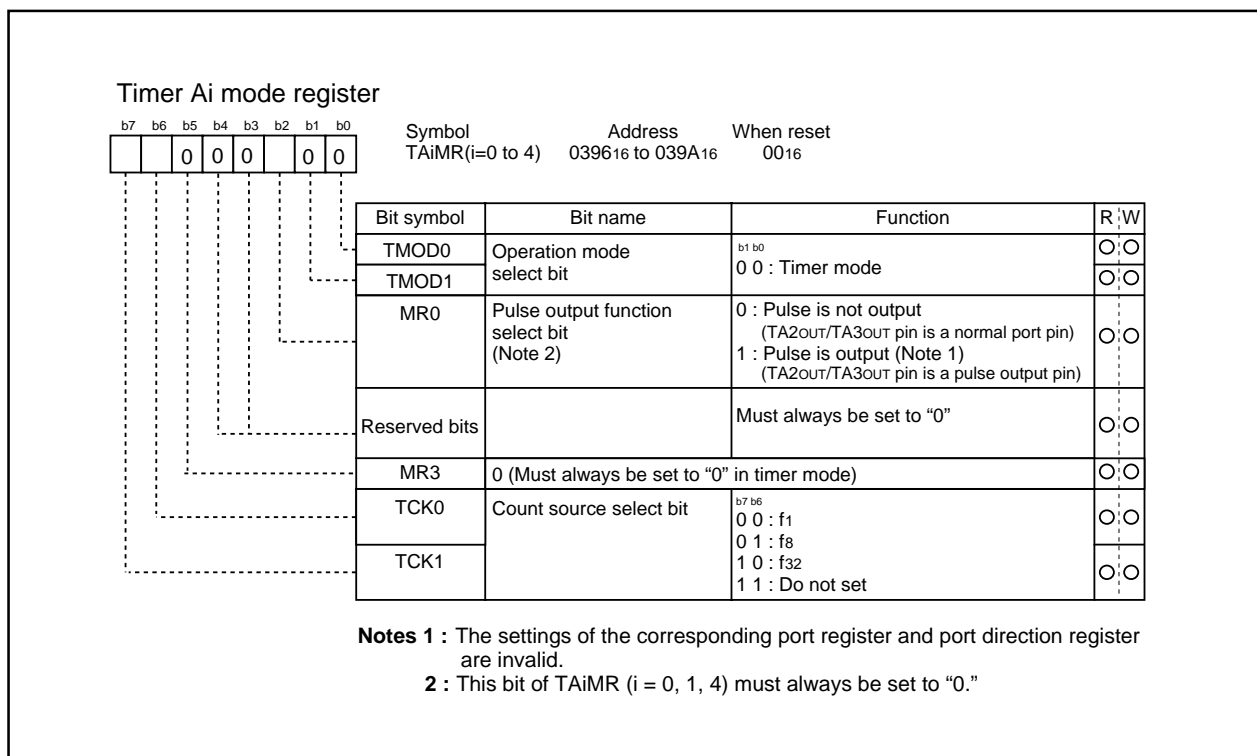
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**(1) Timer mode**

In this mode, the timer counts an internally generated count source. (See Table 2.10.1.) 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>
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	• TB2 overflow, TAj overflow, TAk overflow
Count operation	• Up count or down count can be selected by external signal or software • When the timer overflows or underflows, it reloads the reload register contents before continuing counting (Note)
Divide ratio	1/ (FFFF <sub>16</sub> - 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
TA2OUT/TA3OUT 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	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • 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)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Each time the timer overflows or underflows, the TAIOUT pin's polarity is reversed

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

**M306V5ME-XXXSP**  
**M306V5EESP**

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

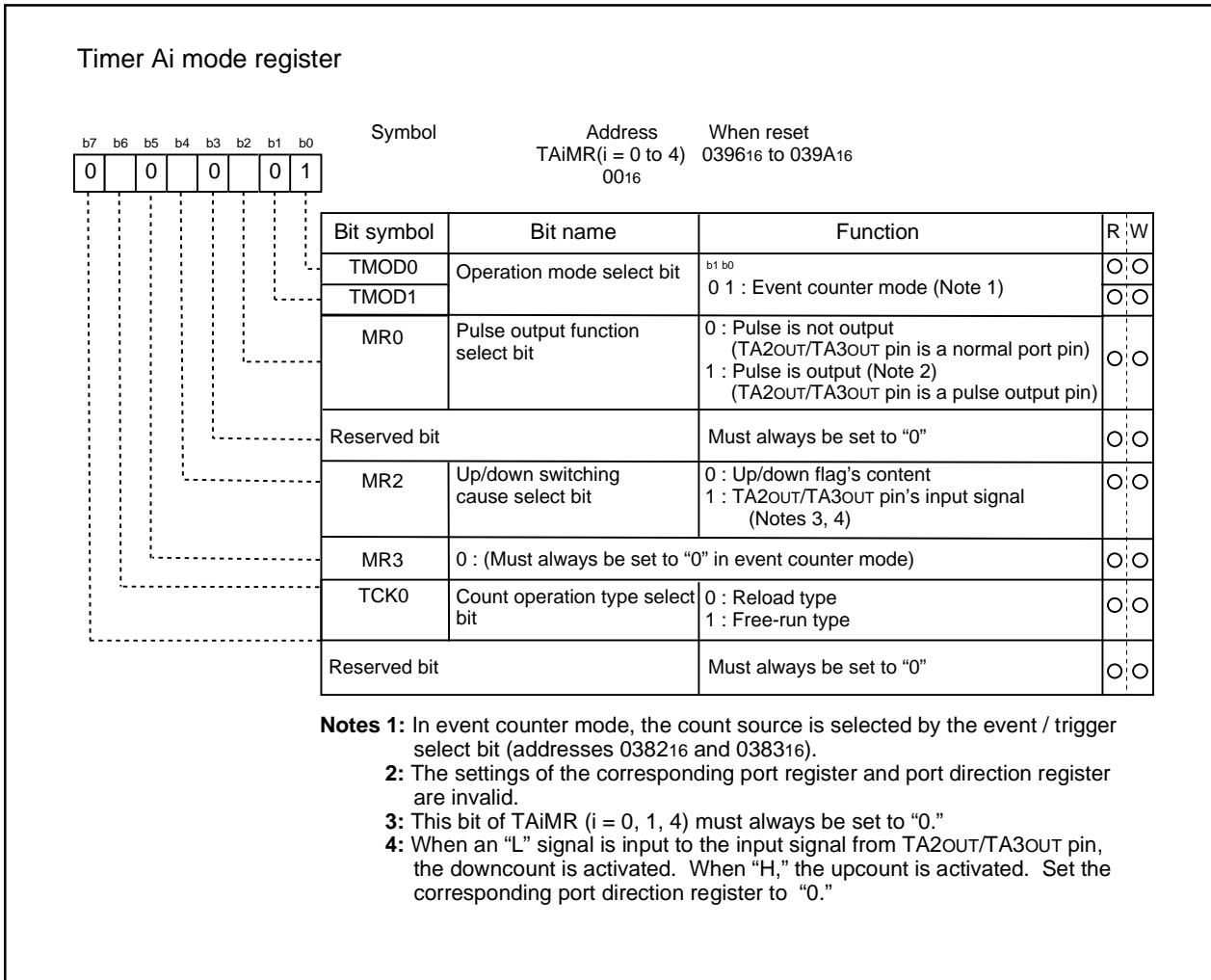


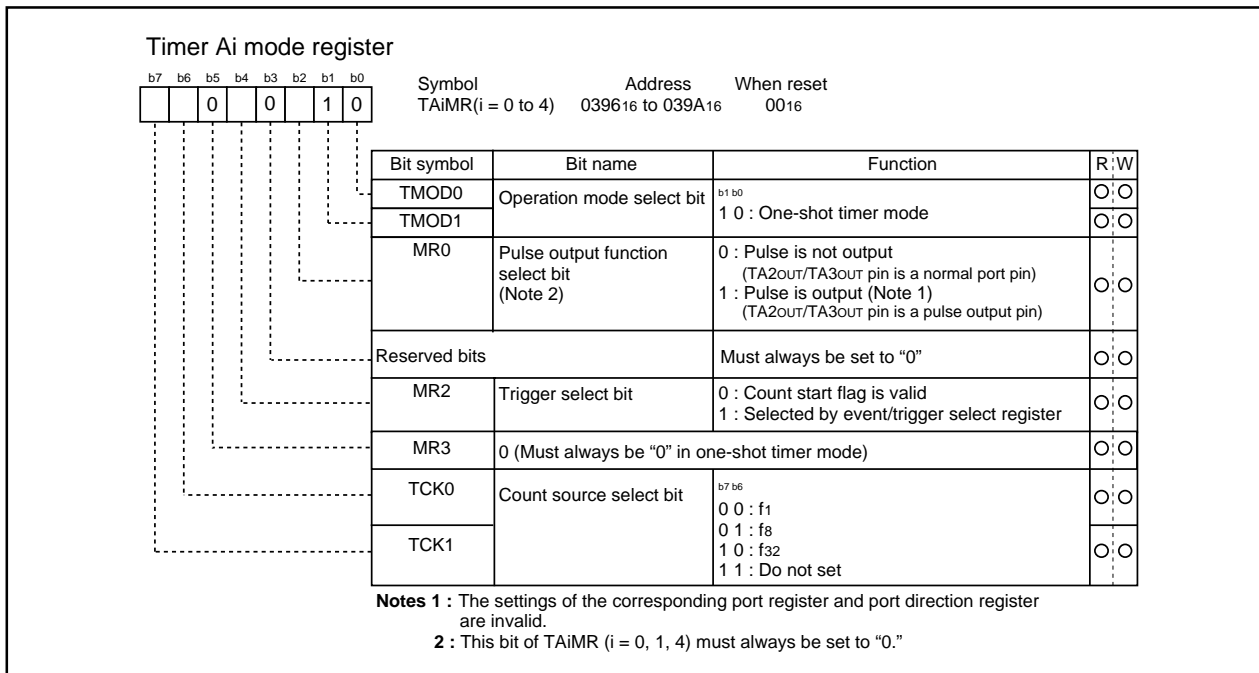
Figure 2.10.12 Timer Ai mode register in event counter mode (i = 0 to 4)

**(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
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>
TA2OUT/TA3OUT 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)**

**M306V5ME-XXXSP**  
**M306V5EESP**

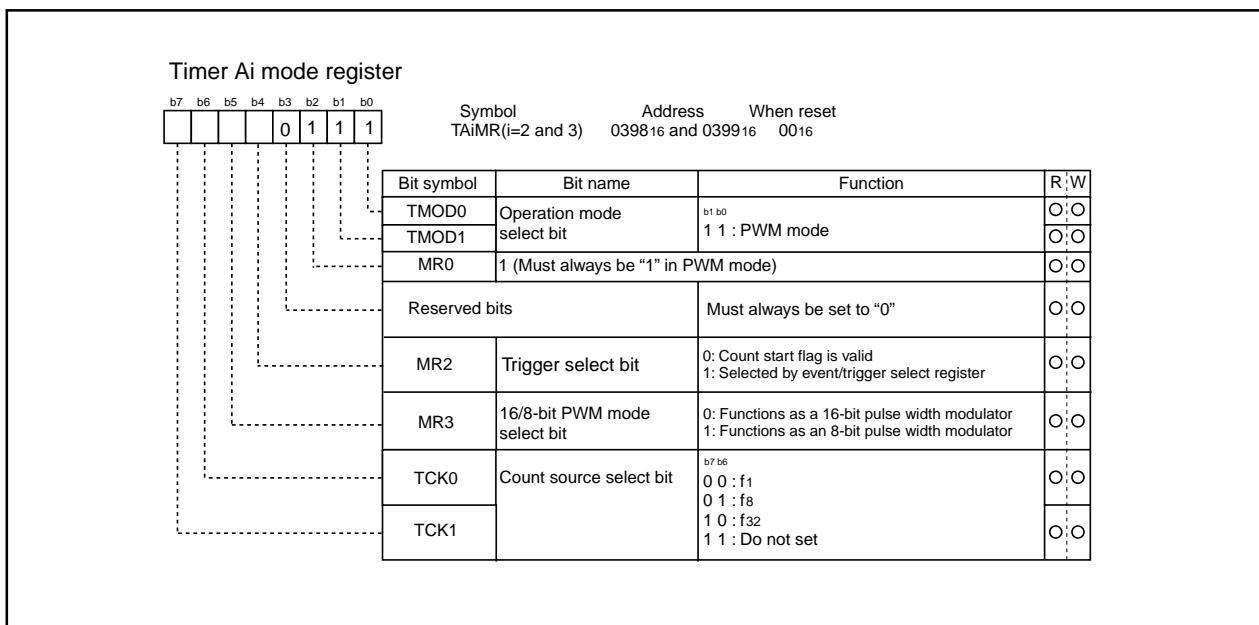
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**(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	f1, f8, f32
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	<ul style="list-style-type: none"> <li>High level width <math>n / f_i</math> n : Set value</li> <li>Cycle time <math>(2^{16}-1) / f_i</math> fixed</li> </ul>
8-bit PWM	<ul style="list-style-type: none"> <li>High level width <math>n \times (m+1) / f_i</math> n : values set to timer Ai register's high-order address</li> <li>Cycle time <math>(2^8-1) \times (m+1) / f_i</math> m : values set to timer Ai register's low-order address</li> </ul>
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"
TA2OUT/TA3OUT 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)**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

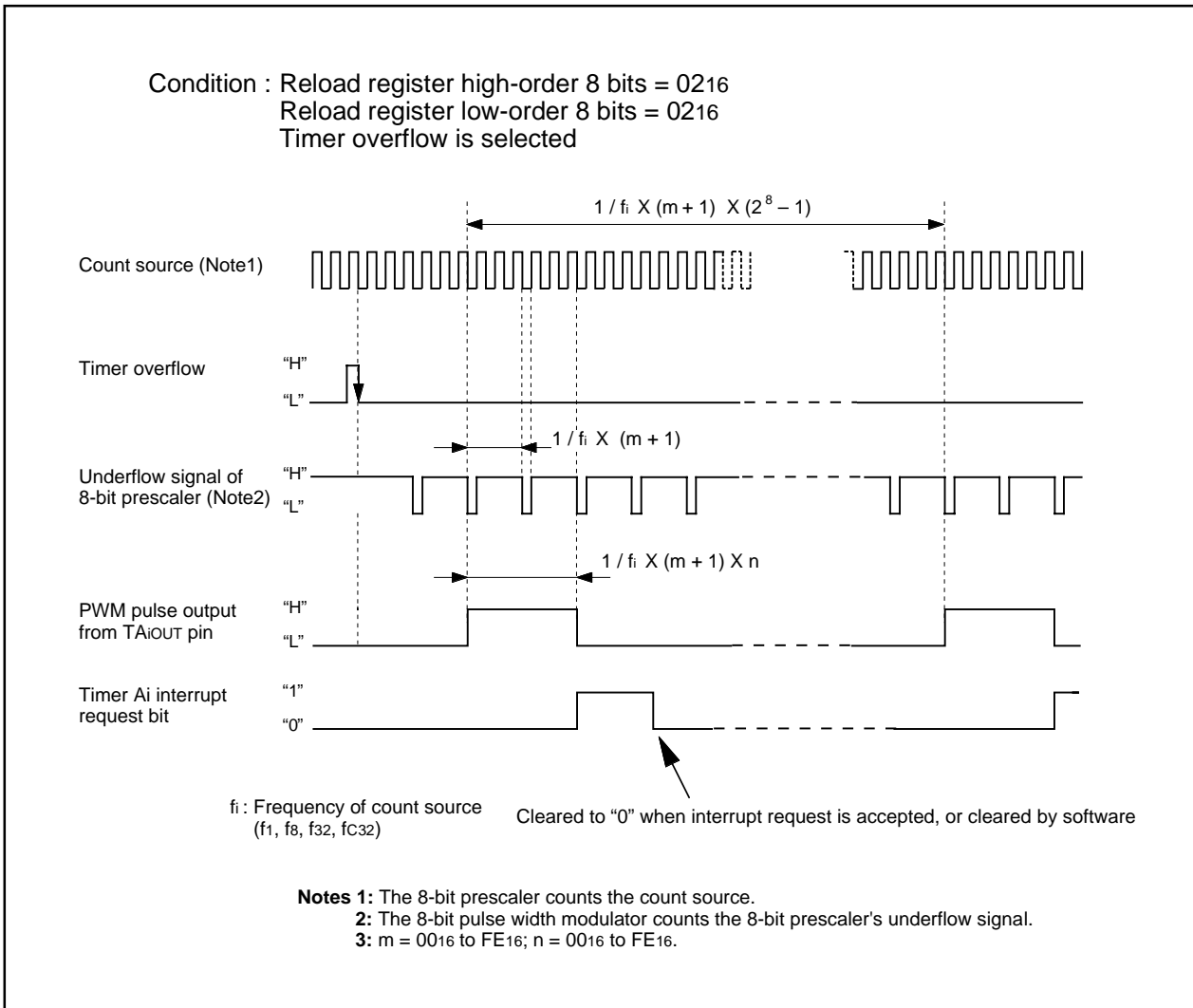


Figure 2.10.15 Example of how an 8-bit pulse width modulator operates



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.

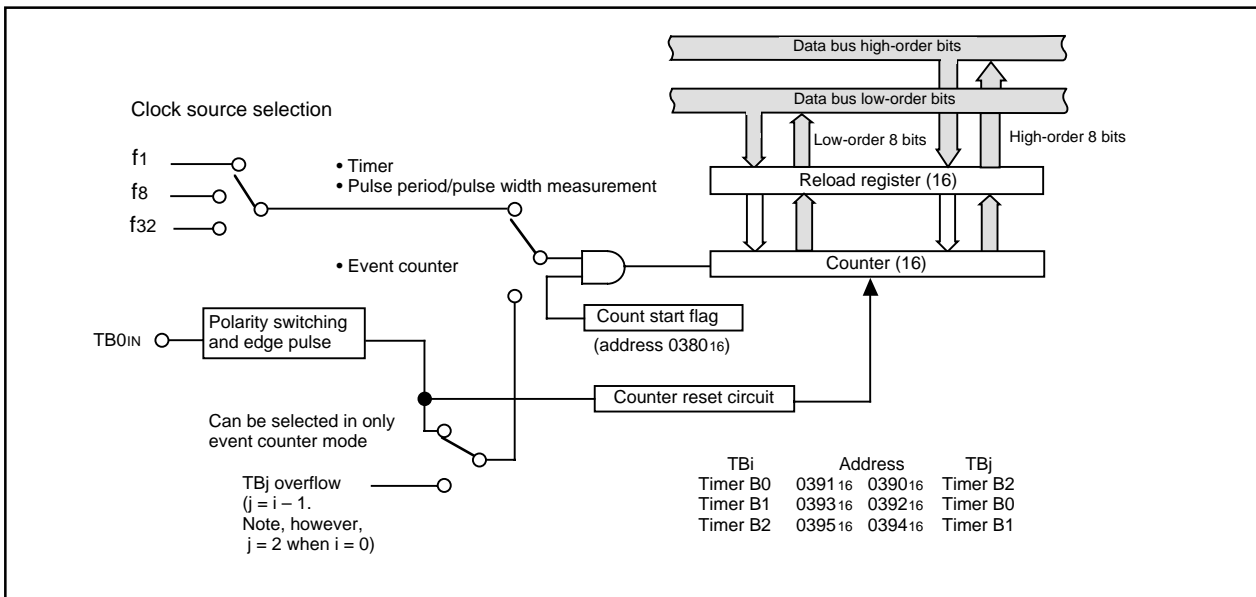


Figure 2.10.16 Block diagram of timer B

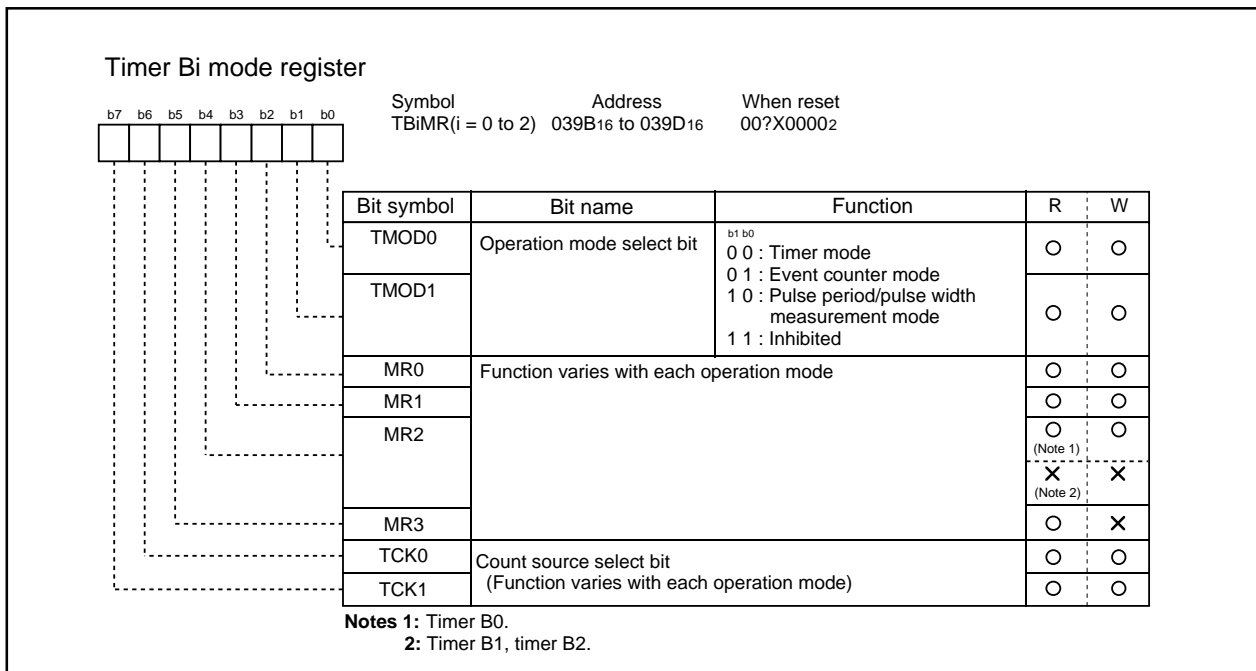
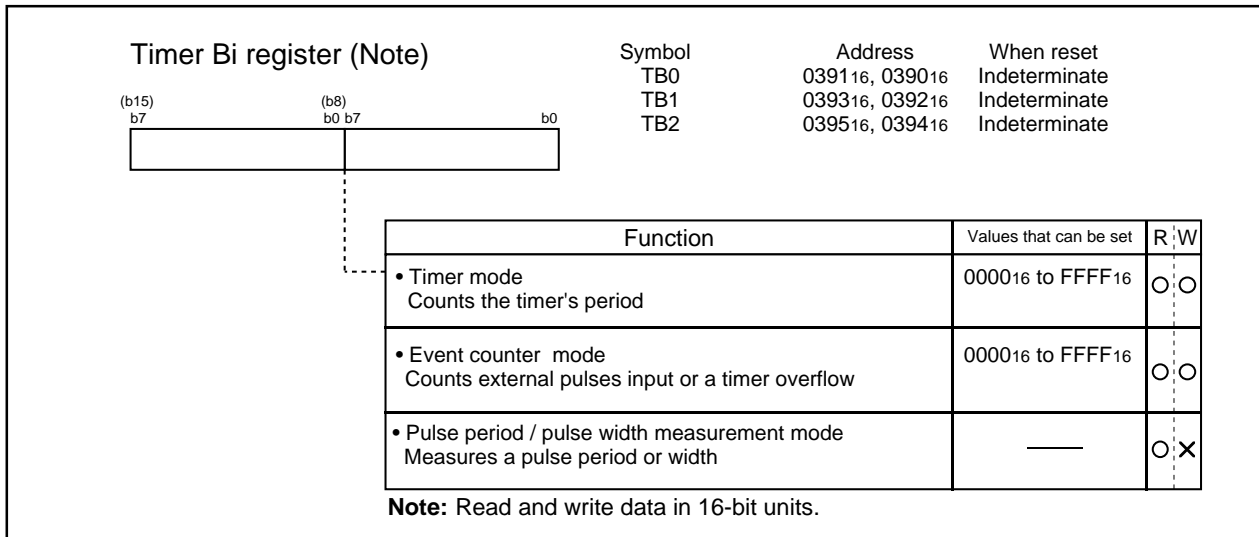


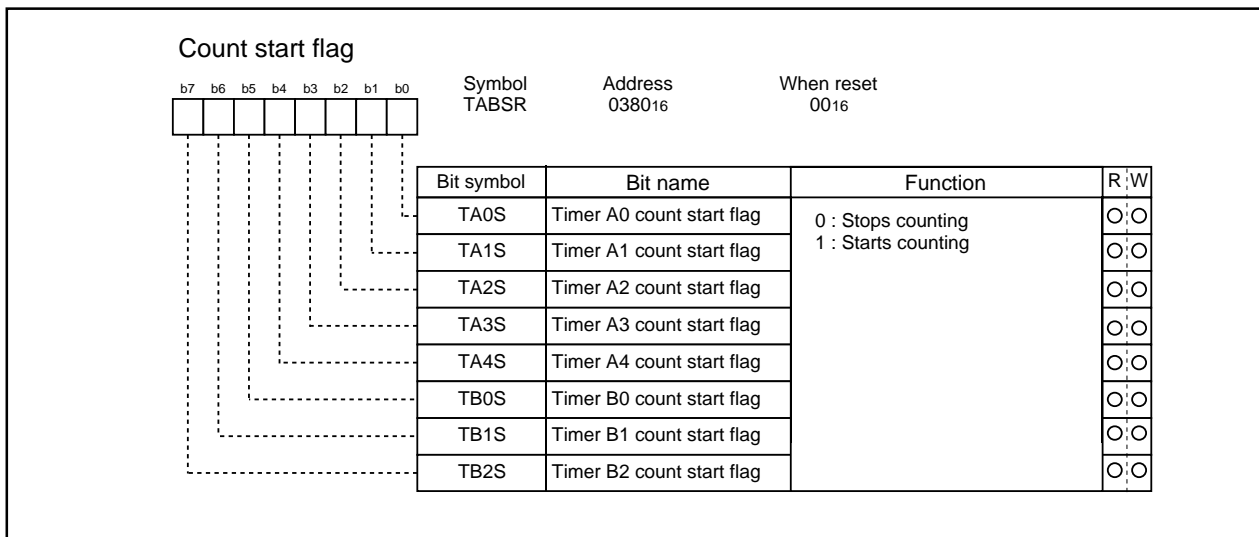
Figure 2.10.17 Timer Bi mode register (i = 0 to 2)

**M306V5ME-XXXSP**  
**M306V5EESP**

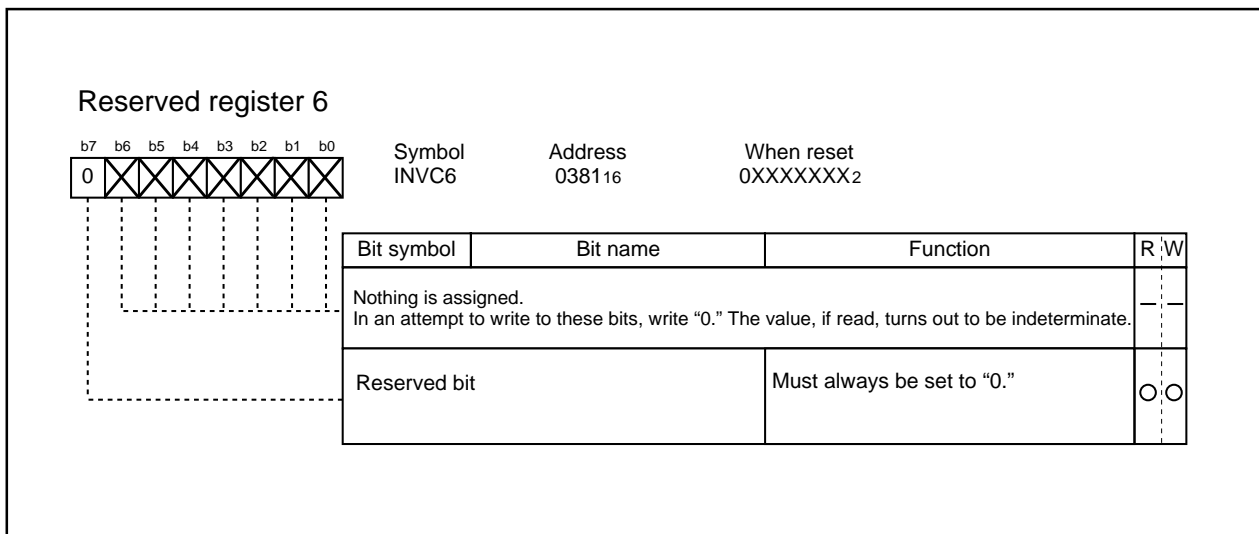
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



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



**Figure 2.10.19** Count start flag



**Figure 2.10.20** Reserved register

**M306V5ME-XXXSP**  
**M306V5EESP**

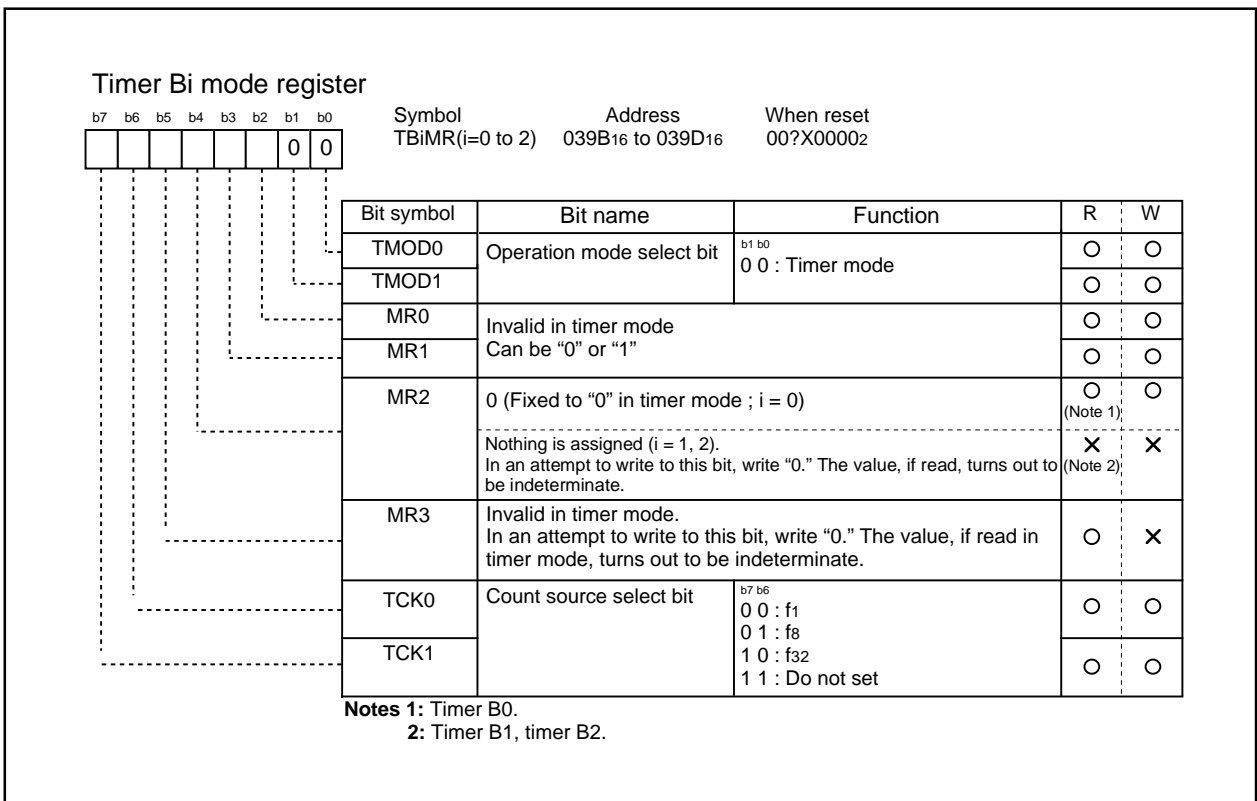
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**(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
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
TBOIN 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)**

**M306V5ME-XXXSP**  
**M306V5EESP**

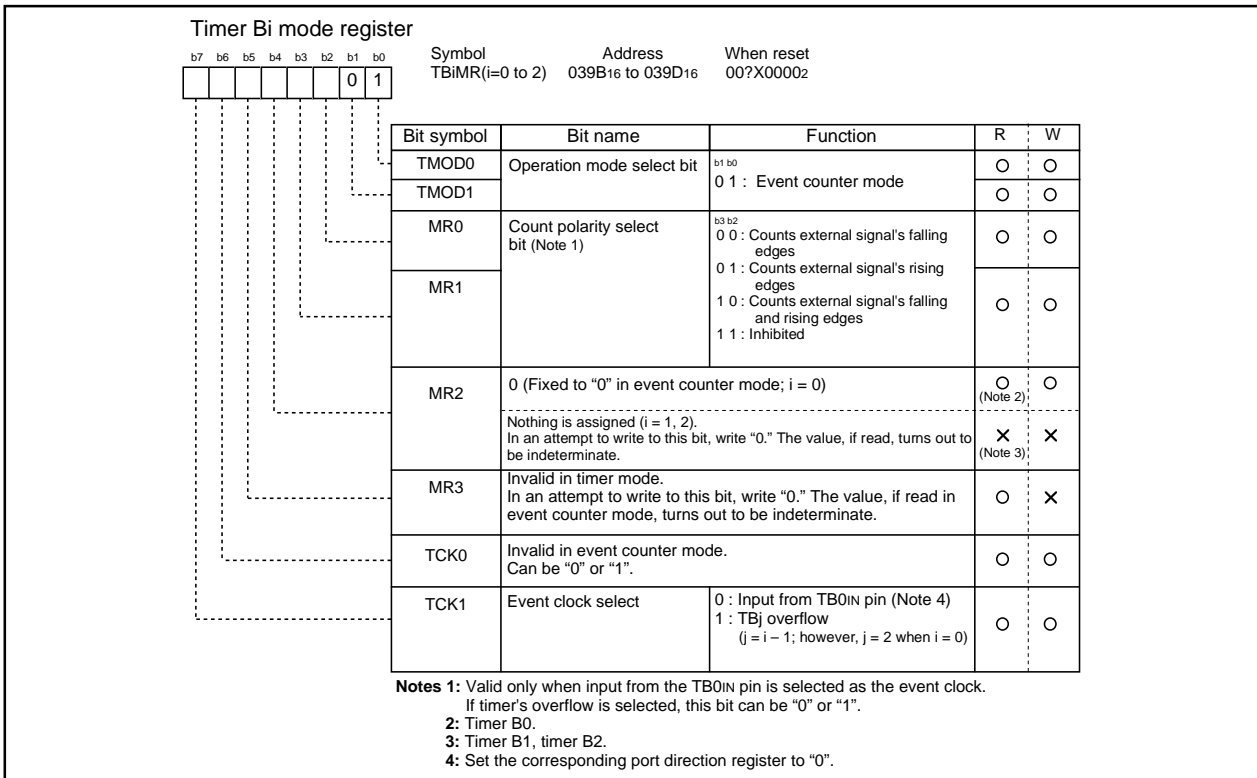
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**(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 TB0IN 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
TB0IN 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)**

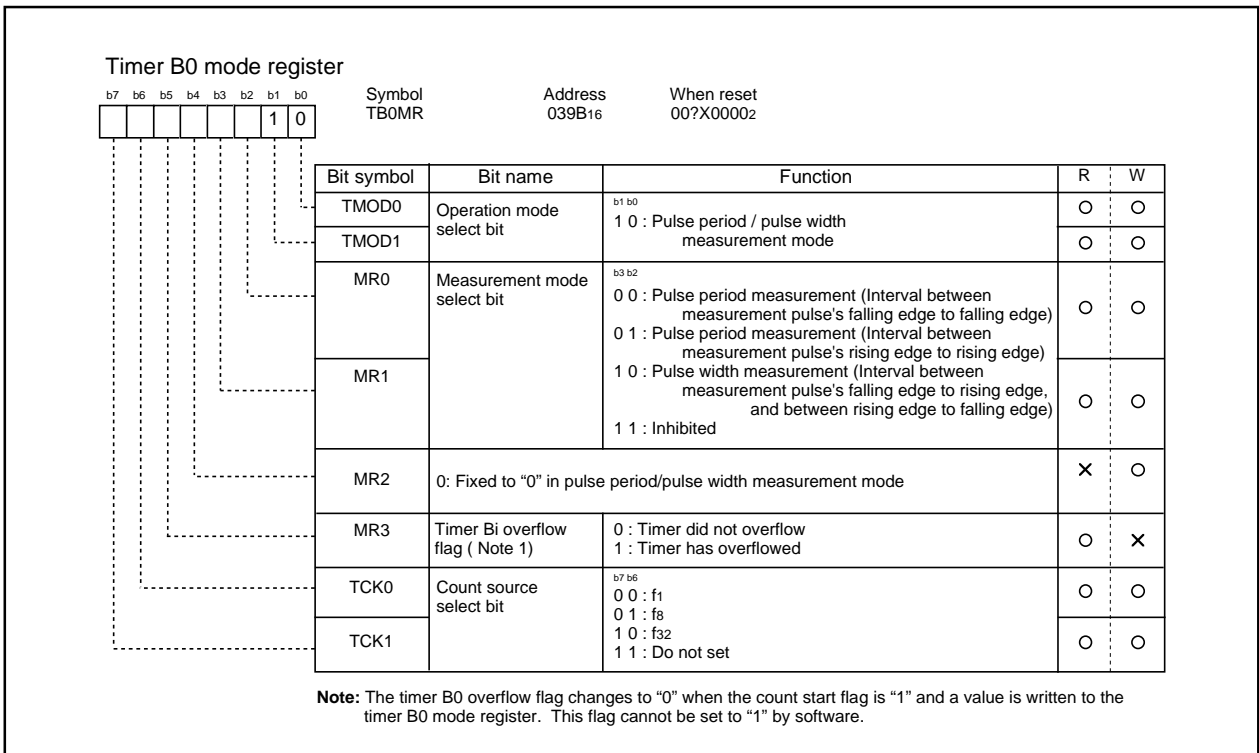
**(3) Pulse period/pulse width measurement mode**

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 B0 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
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 B0 overflow flag changes to "0" when the count start flag is "1" and a value is written to the timer B0 mode register.)</li> </ul>
TB0IN pin function	Measurement pulse input
Read from timer	When timer B0 register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

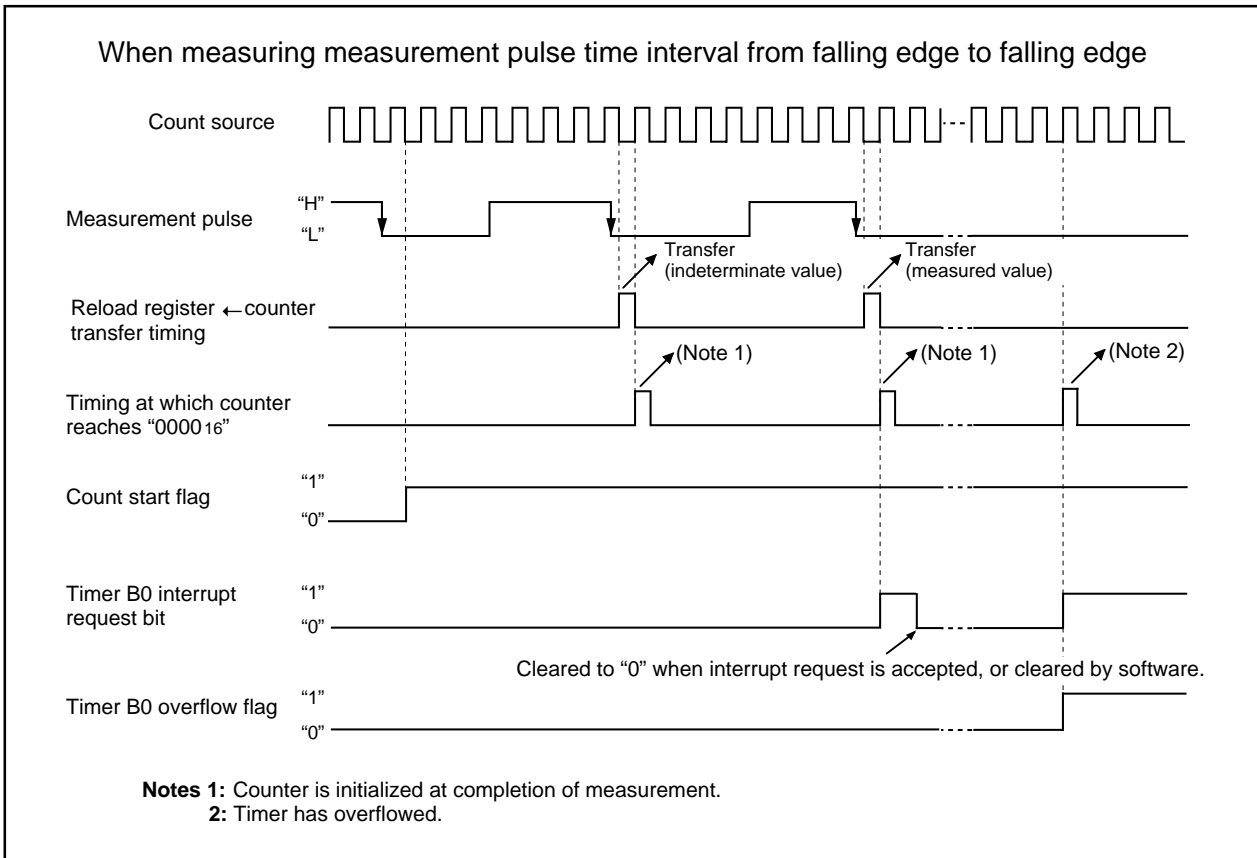
**Notes 1:** An interrupt request is not generated when the first effective edge is input after the timer has started counting.  
**2:** The value read out from the timer B0 register is indeterminate until the second effective edge is input after the timer.



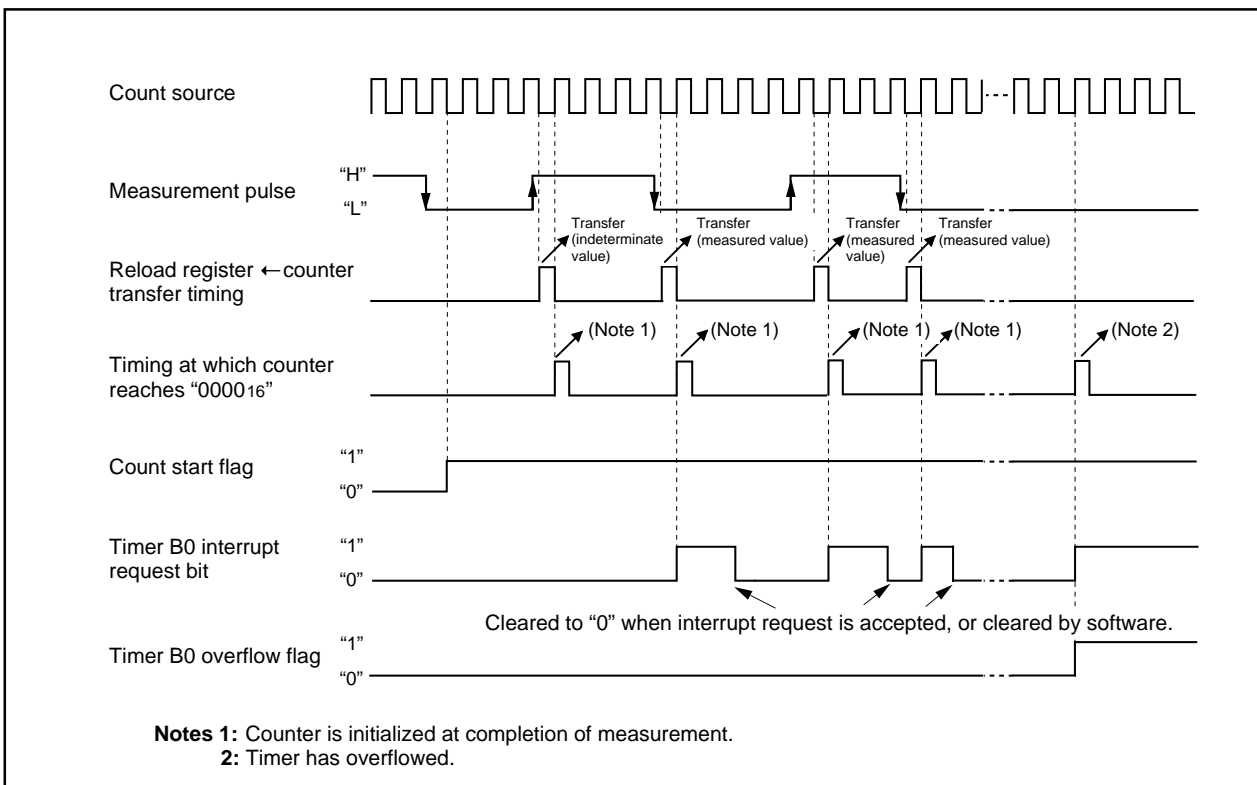
**Figure 2.10.23 Timer B0 mode register in pulse period/pulse width measurement mode**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**M306V5ME-XXXSP**  
**M306V5EESP**

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

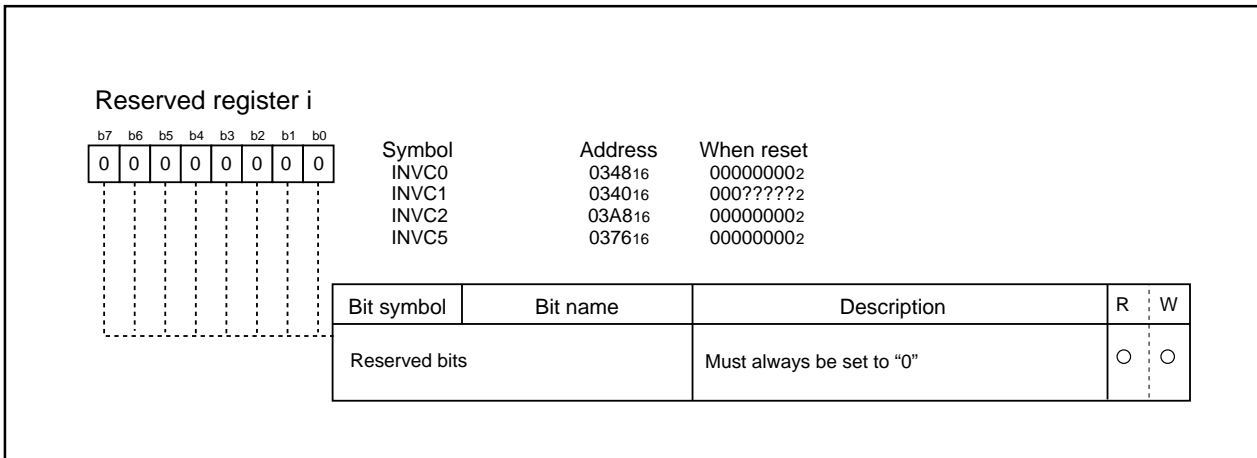


Figure 2.10.26 Reserved register i (i = 0 to 2, 5)

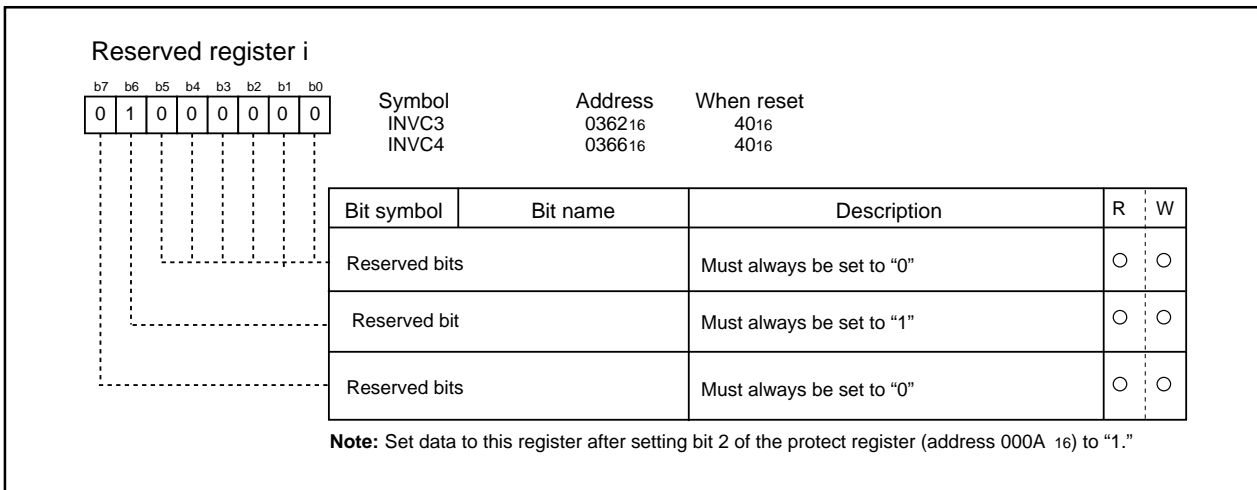
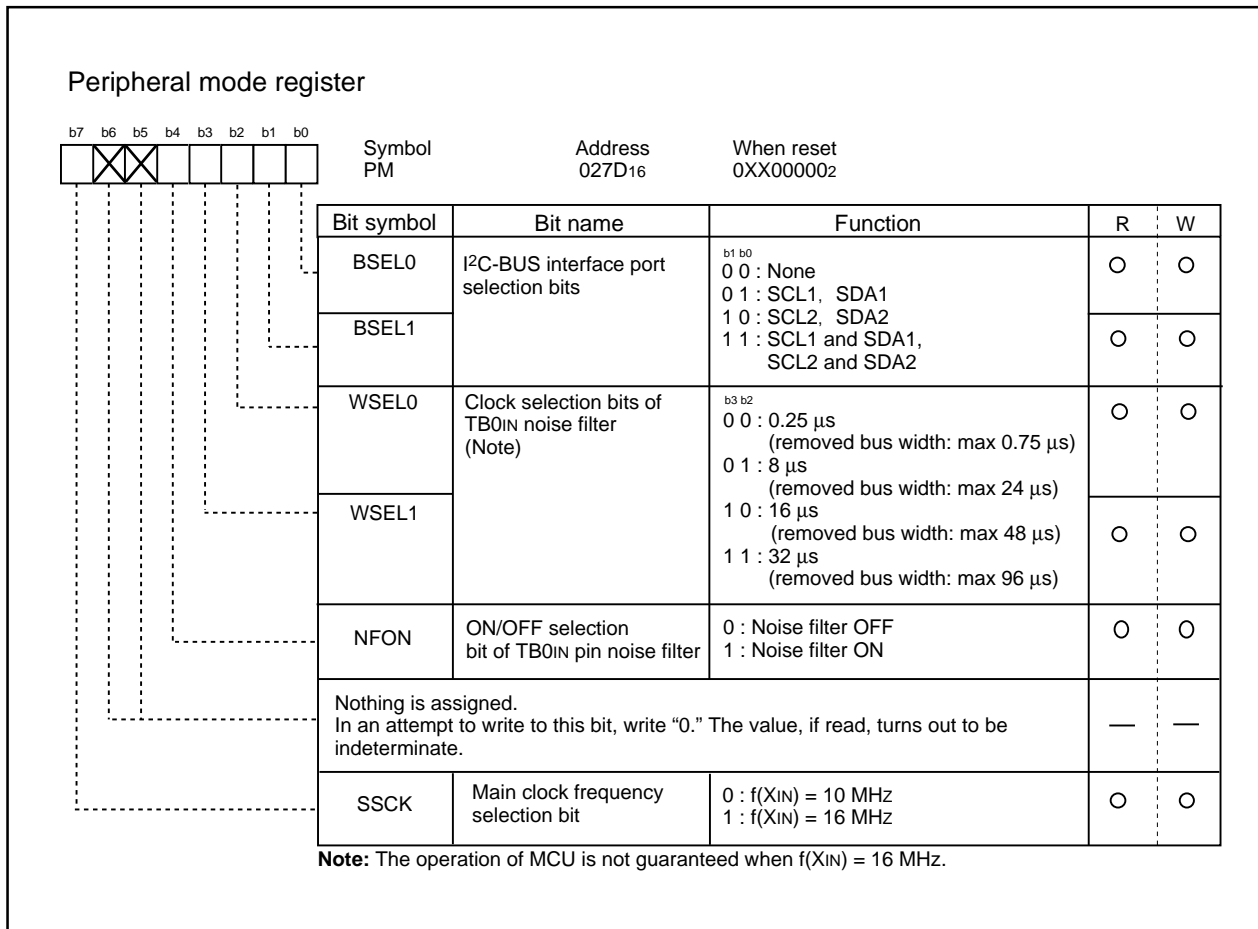


Figure 2.10.27 Reserved register i (i = 3 and 4)

**(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 2.10.28 Peripheral mode register**



## 2.11 Serial I/O

Serial I/O is configured as 4 unites: UART0, UART2, multi-master I<sup>2</sup>C-BUS interface 0, and multi-master I<sup>2</sup>C-BUS interface 1.

### 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.

UARTi (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 UARTi 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. 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 UARTi.

**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.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

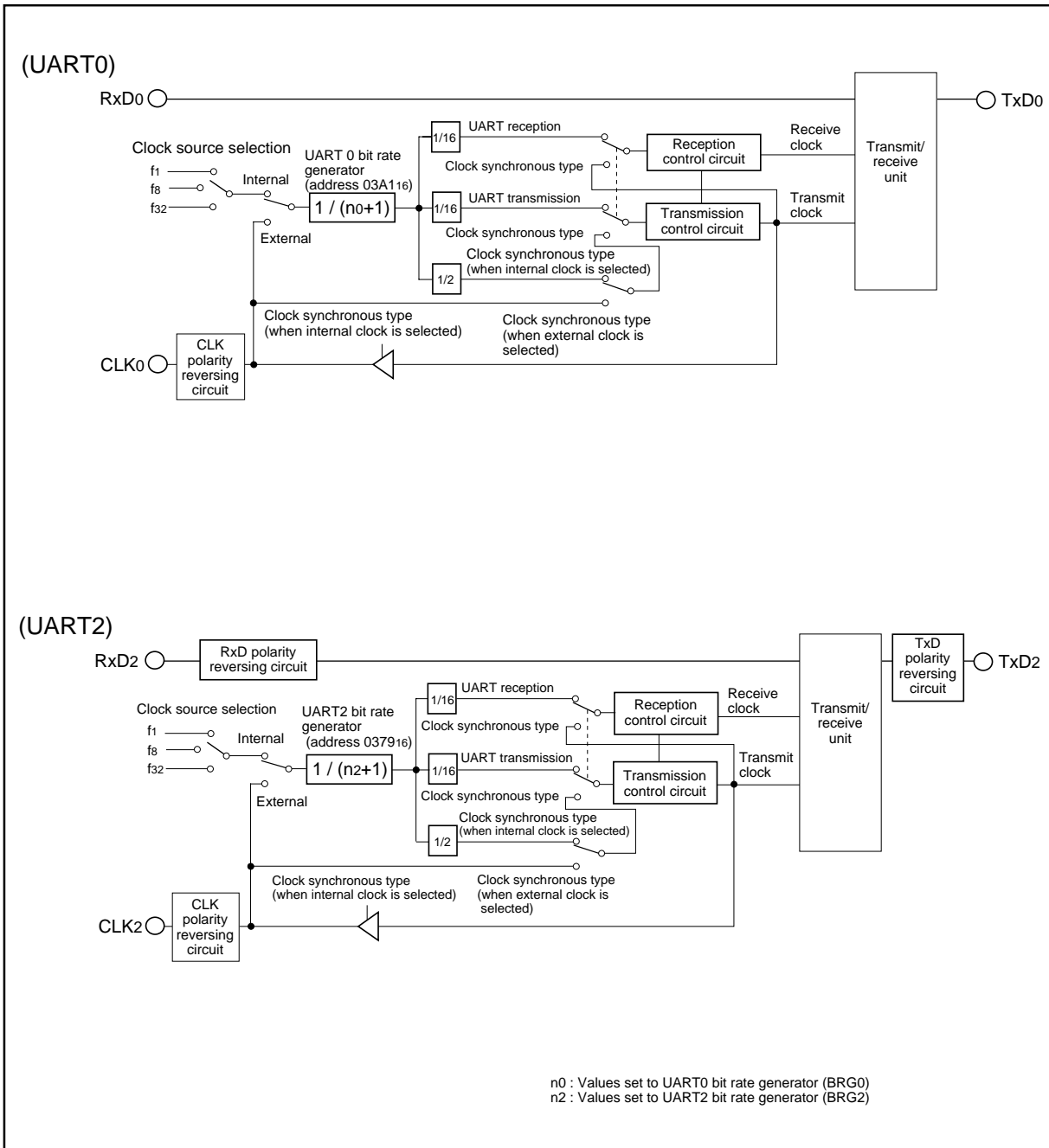


Figure 2.11.1 Block diagram of UARTi (i = 0 and 2)

**M306V5ME-XXXSP**  
**M306V5EESP**

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

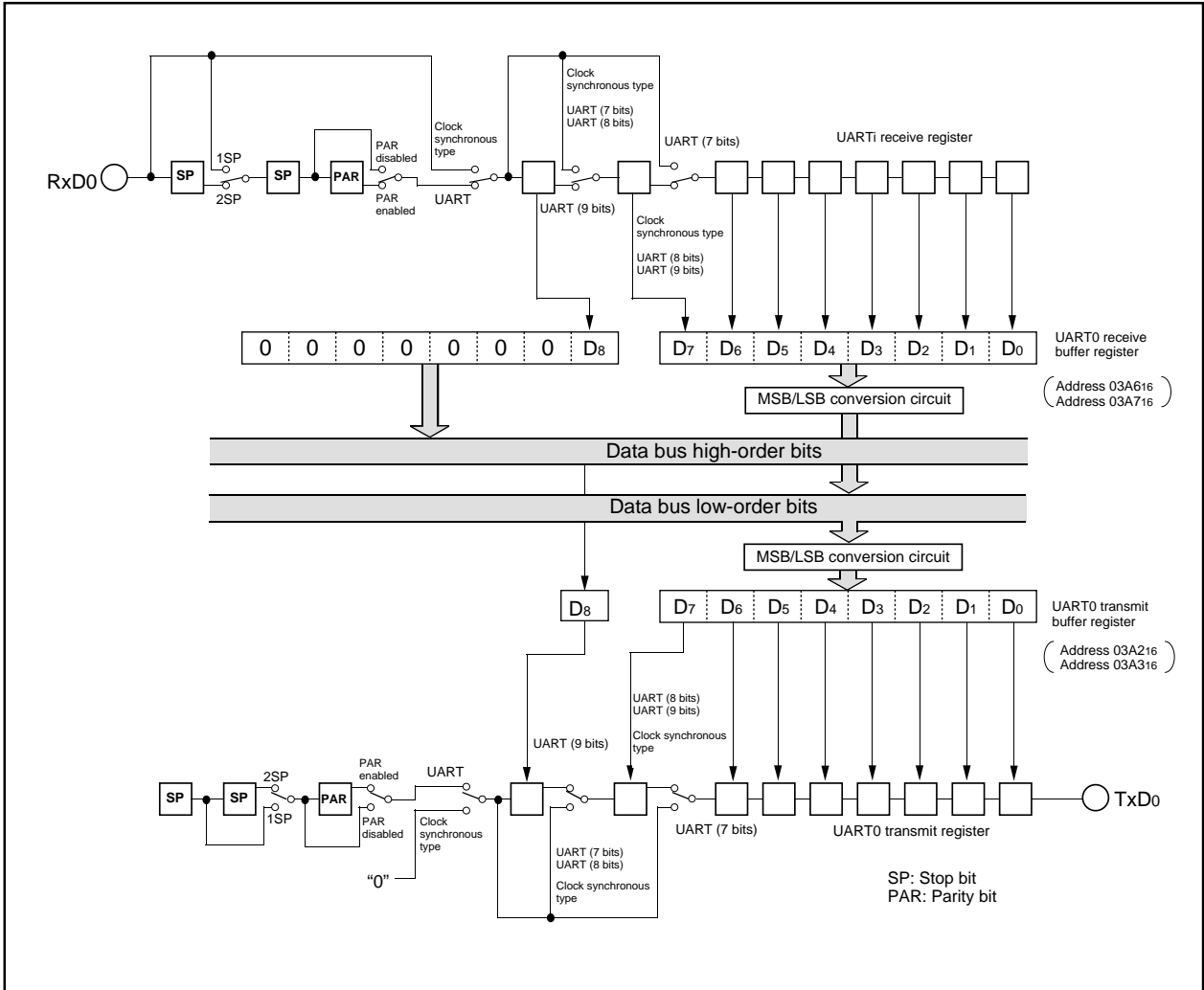


Figure 2.11.2 Block diagram of UART0 transmit/receive unit

**M306V5ME-XXXSP**  
**M306V5EESP**

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

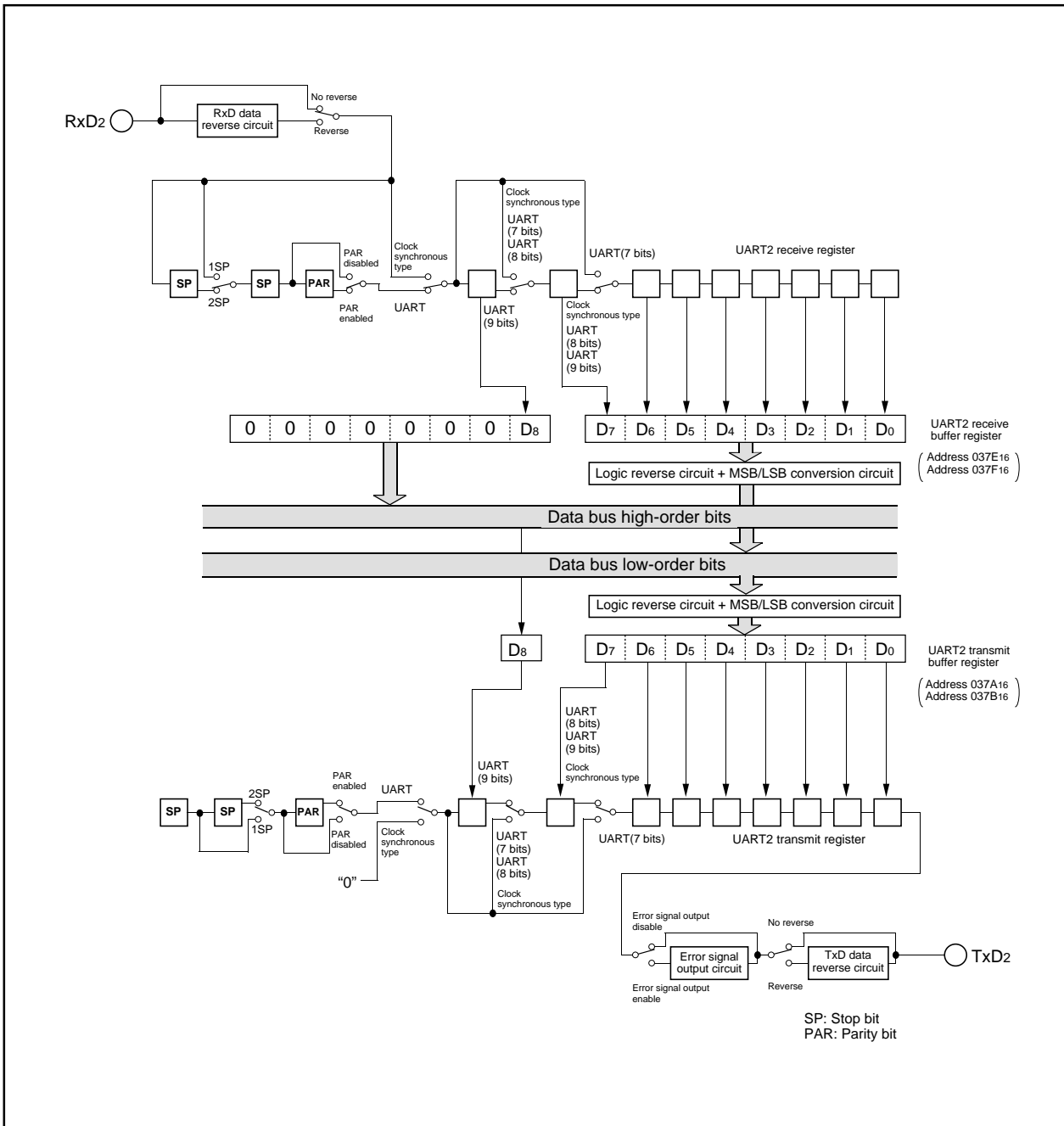
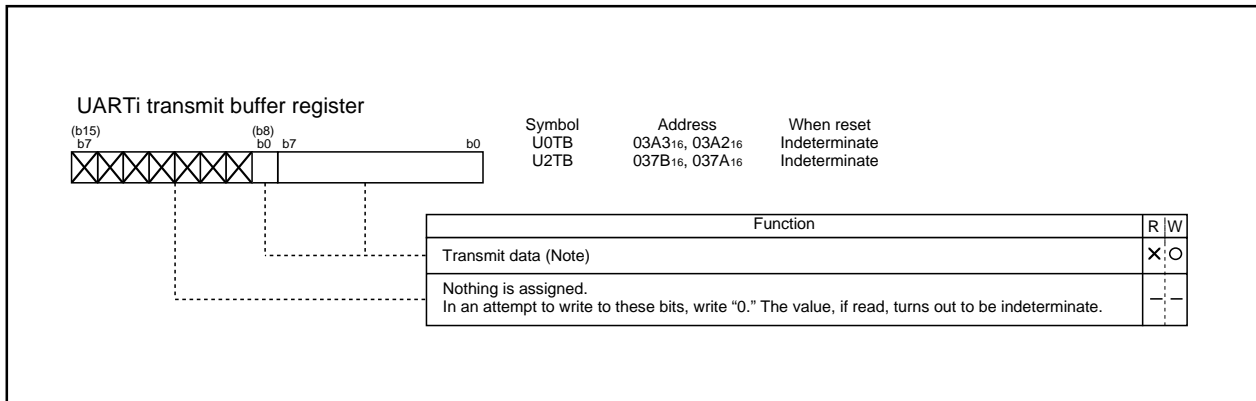


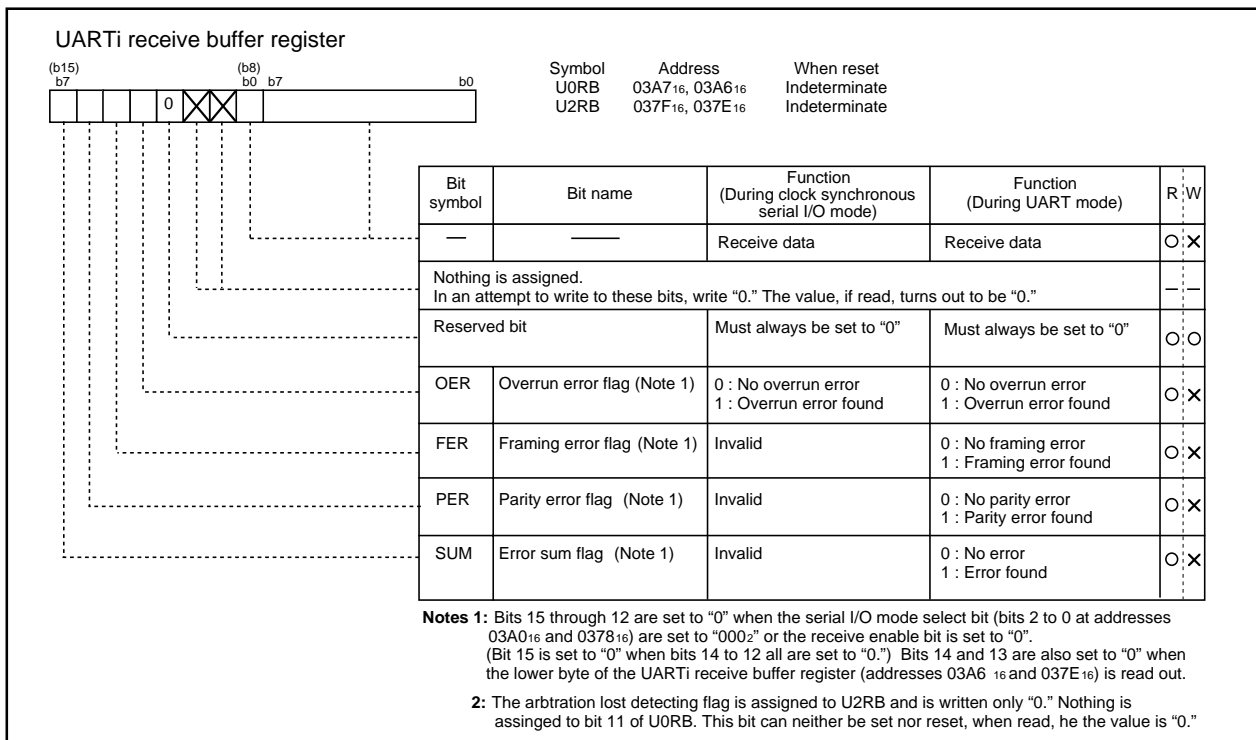
Figure 2.11.3 Block diagram of UART2 transmit/receive unit

**M306V5ME-XXXSP**  
**M306V5EESP**

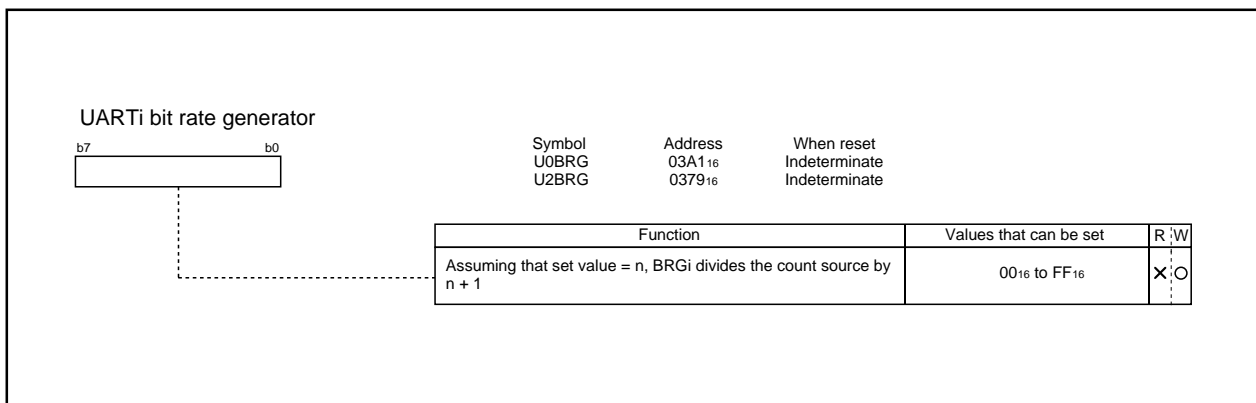
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**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)**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

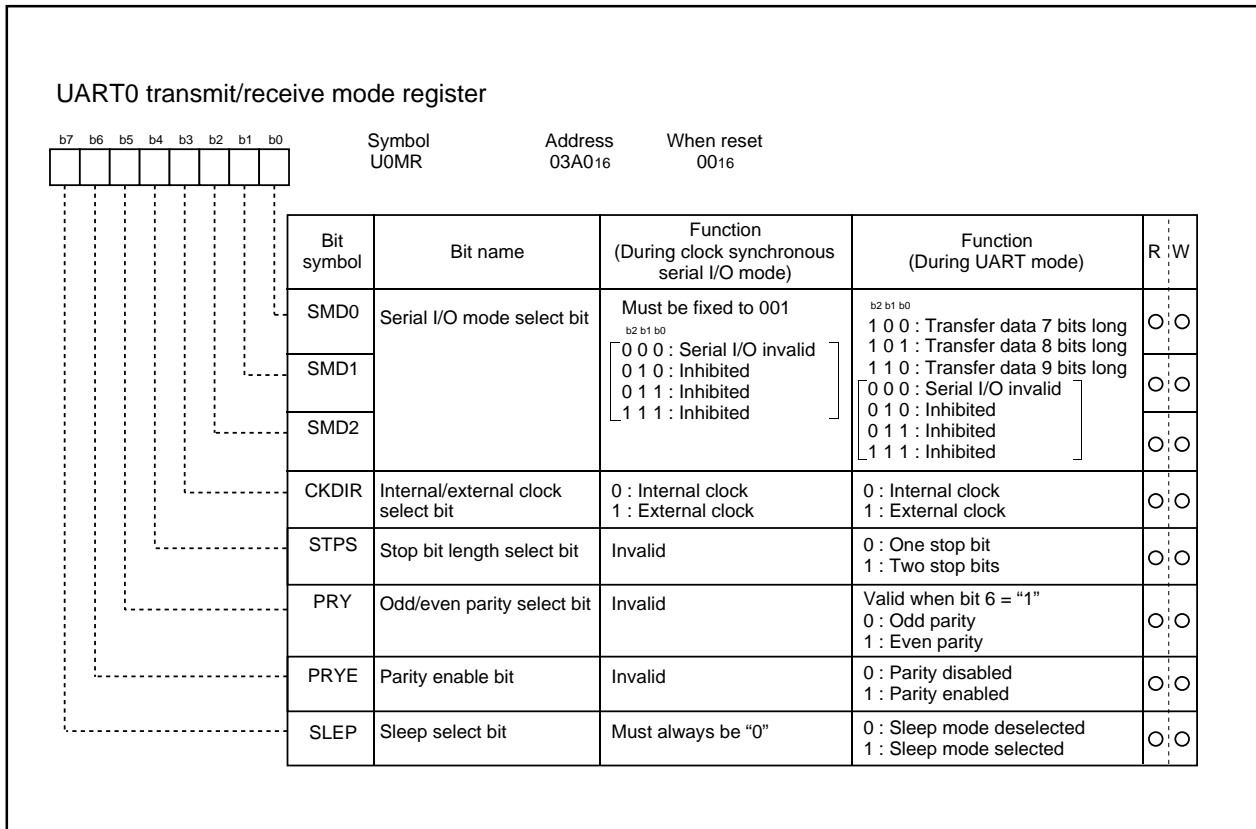


Figure 2.11.7 UART0 transmit/receive mode register

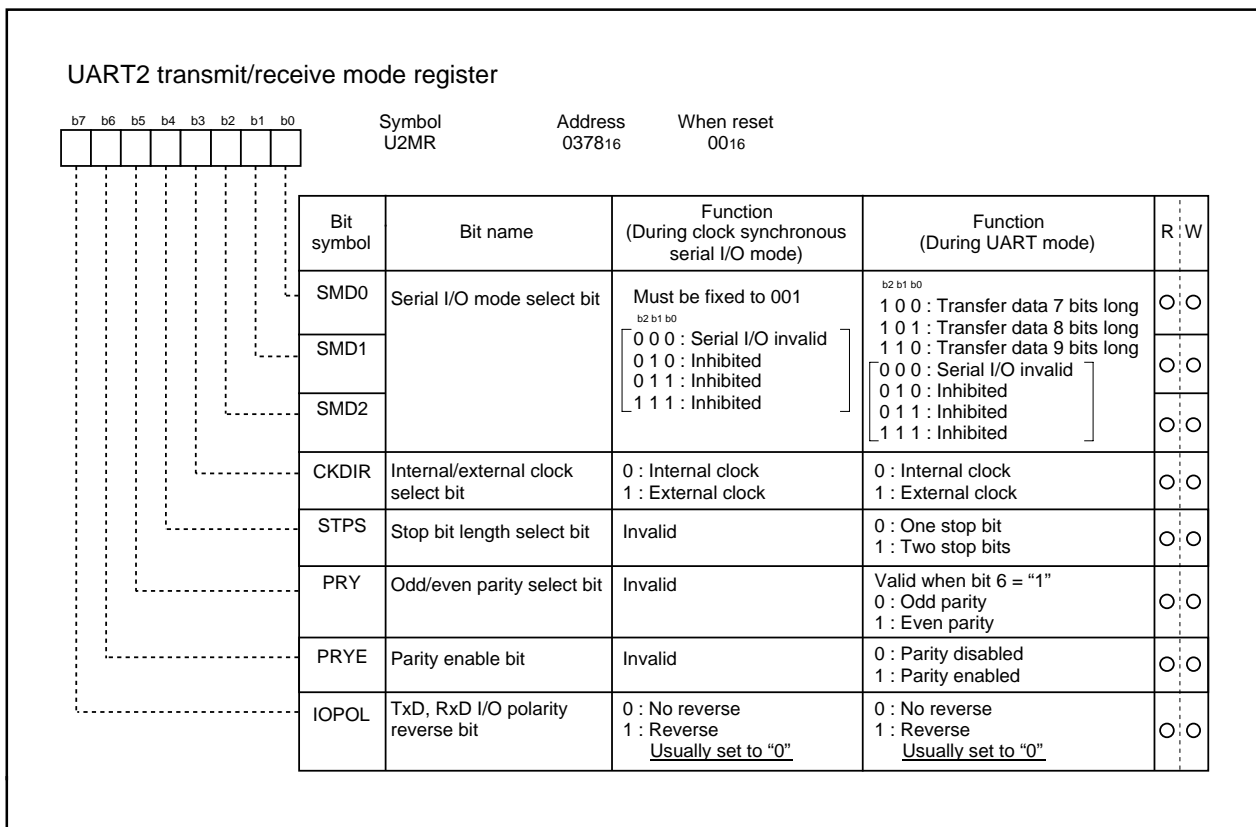


Figure 2.11.8 UART2 transmit/receive mode register

**M306V5ME-XXXSP**  
**M306V5EESP**

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

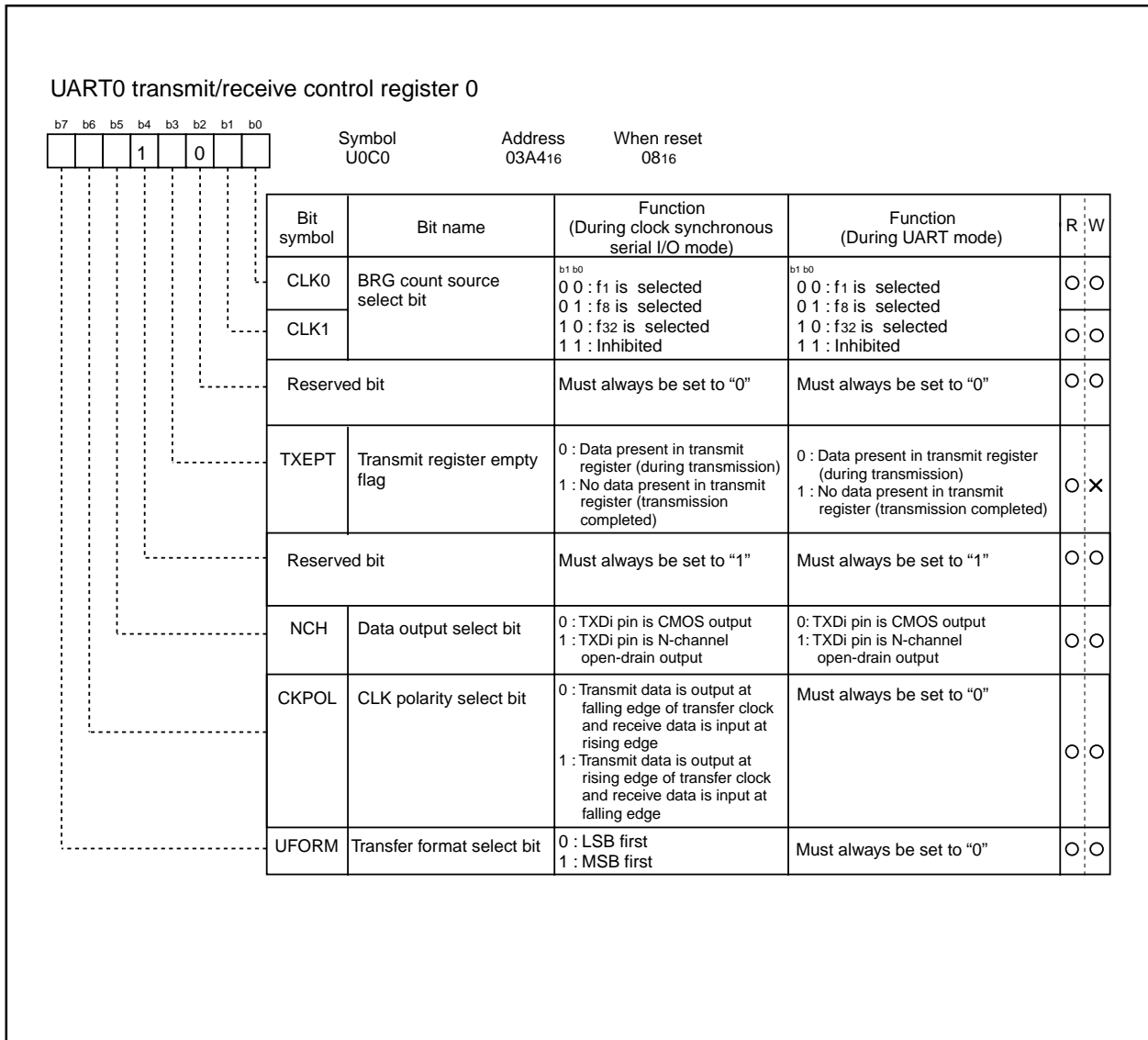


Figure 2.11.9 UART0 transmit/receive control register 0

**M306V5ME-XXXSP**  
**M306V5EESP**

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

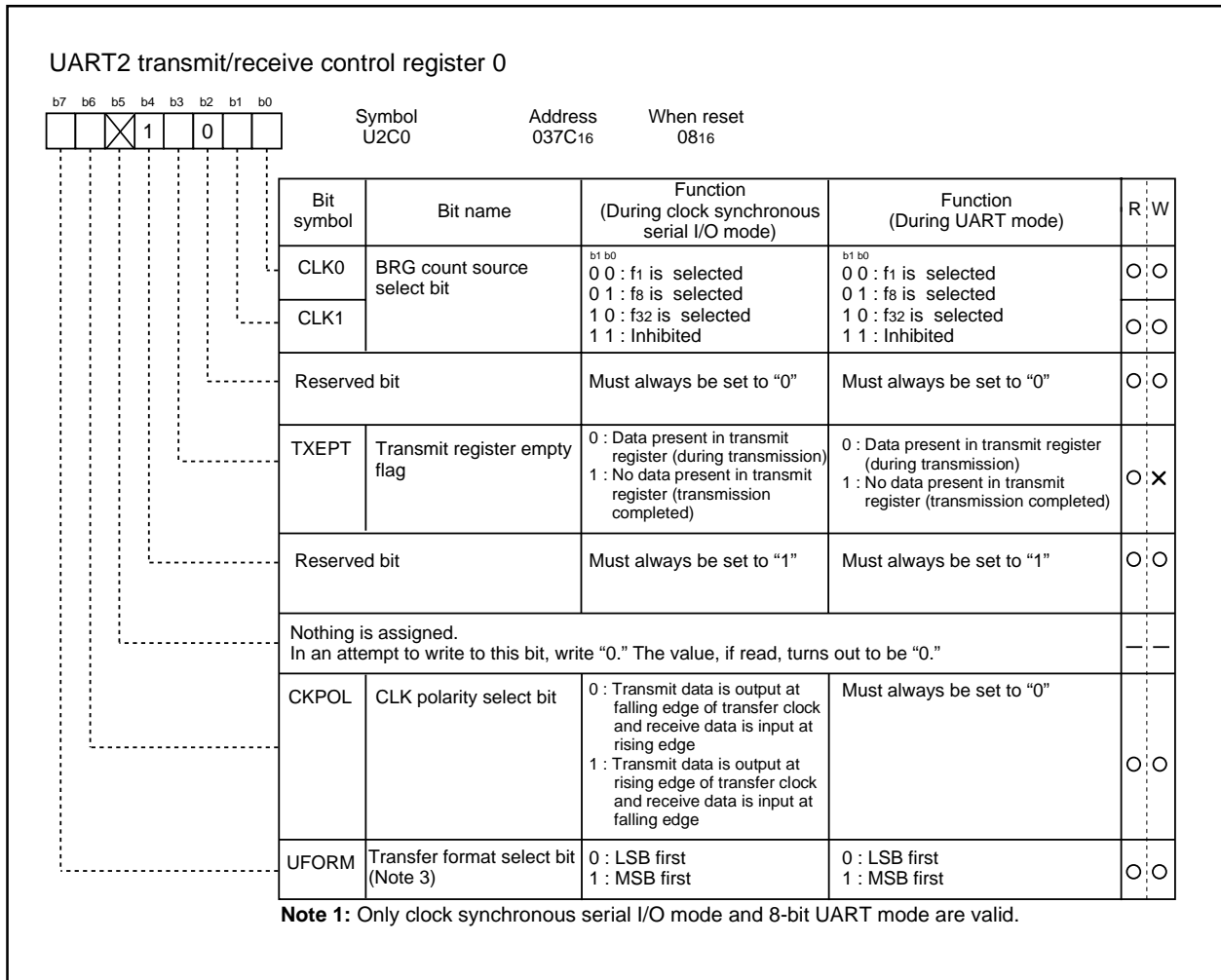


Figure 2.11.10 UART2 transmit/receive control register 0



**M306V5ME-XXXSP**  
**M306V5EESP**

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

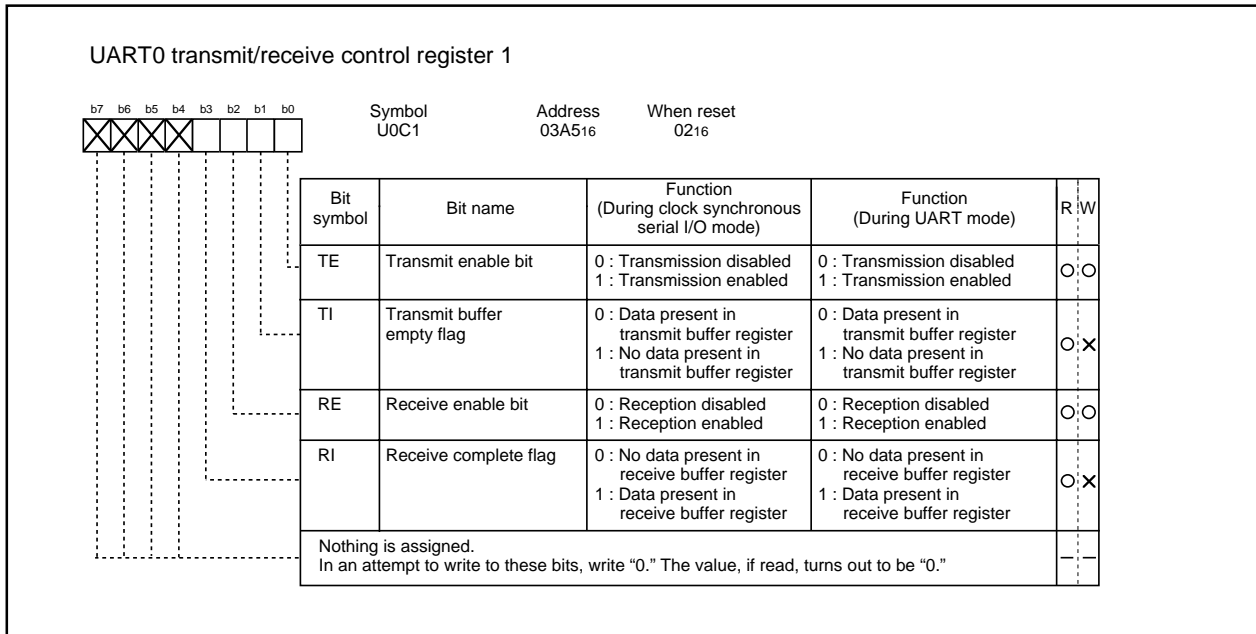


Figure 2.11.11 UART0 transmit/receive control register 1

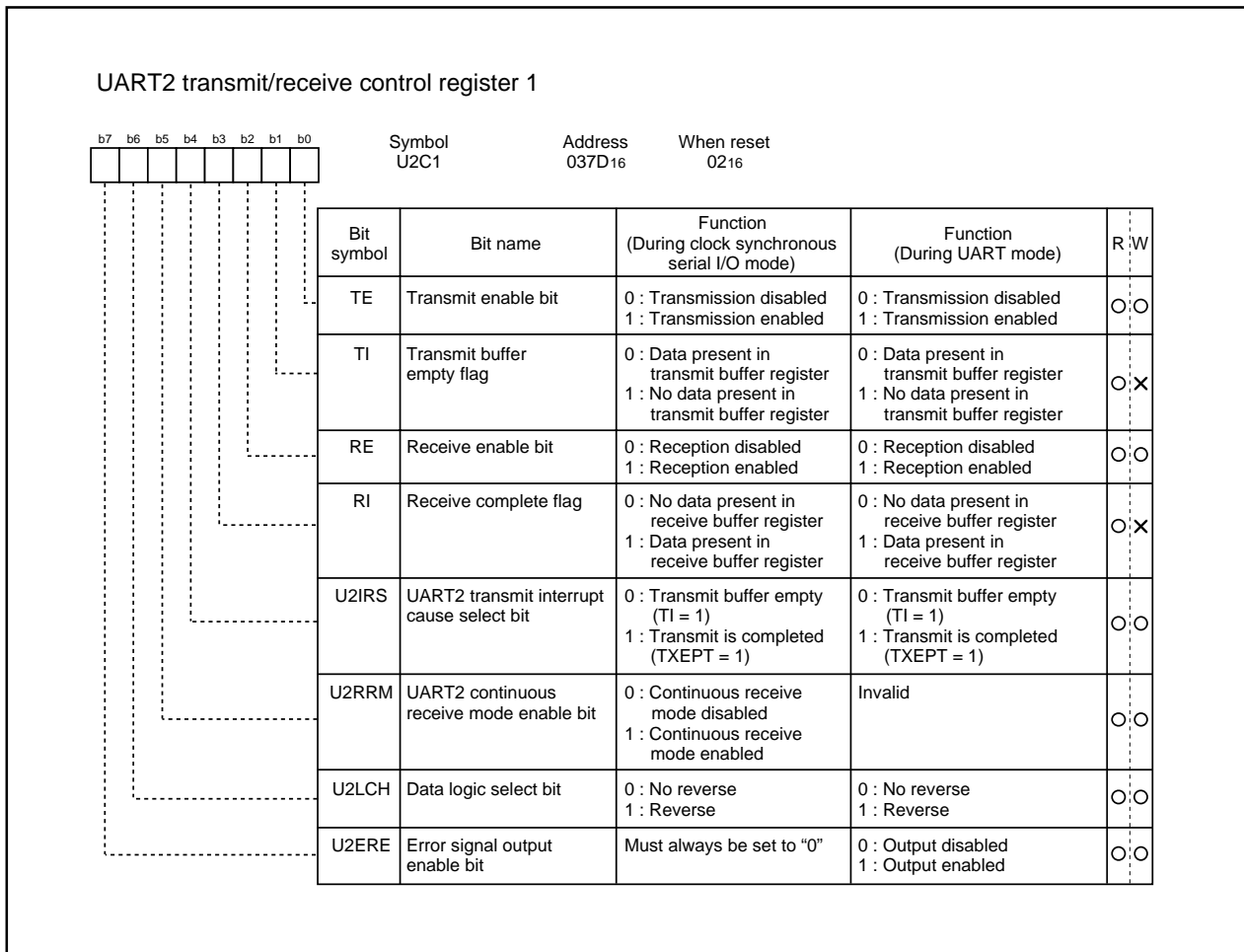
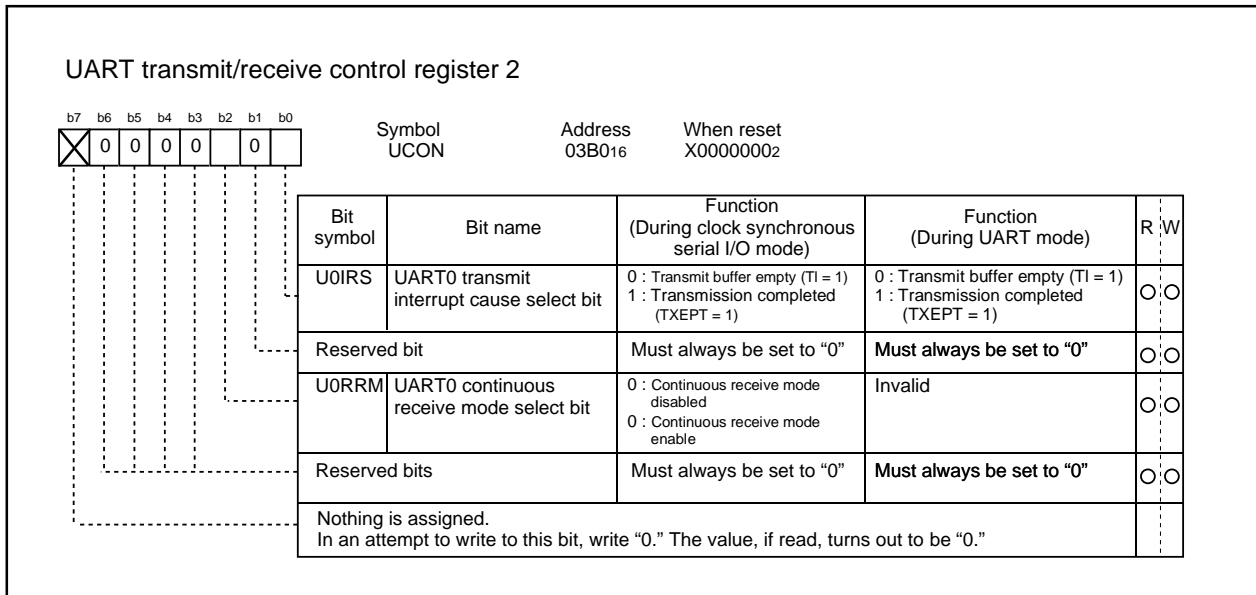


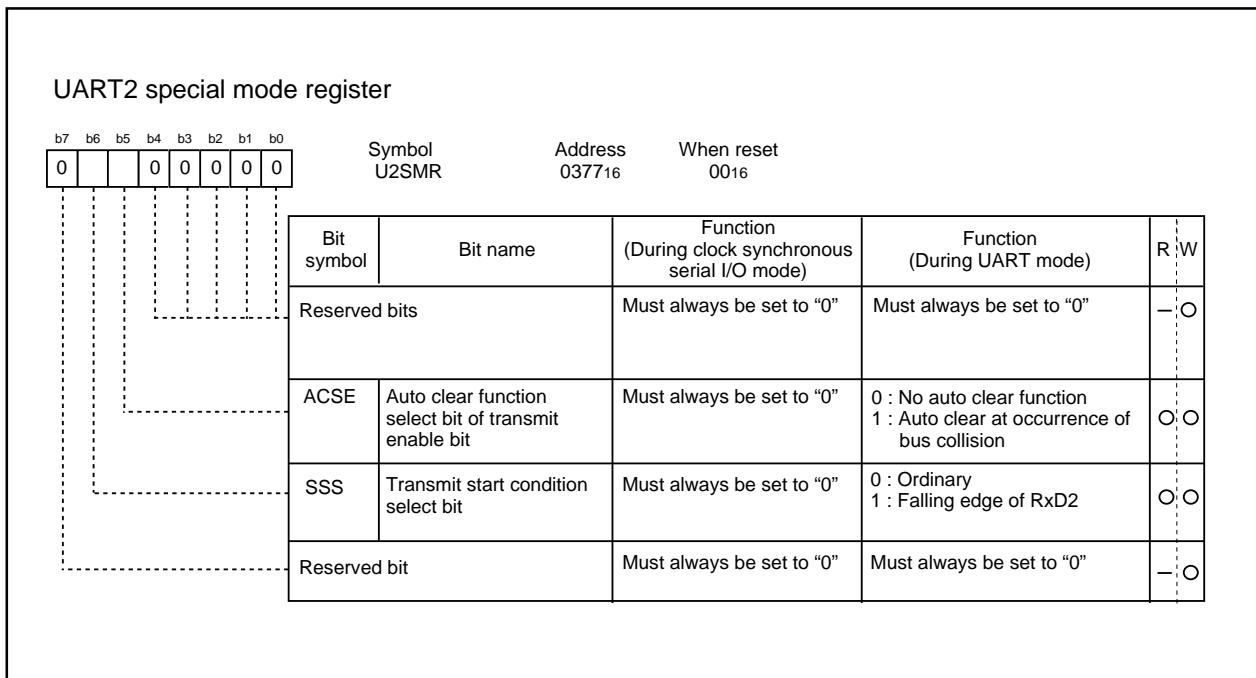
Figure 2.11.12 UART2 transmit/receive control register 1

**M306V5ME-XXXSP**  
**M306V5EESP**

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 Transfer clock	<ul style="list-style-type: none"> <li>• Transfer data length: 8 bits</li> <li>• When internal clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 0378<sub>16</sub> = "0") : fi/ 2(n+1) (Note 1) fi = f1, f8, f32</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 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> </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”.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

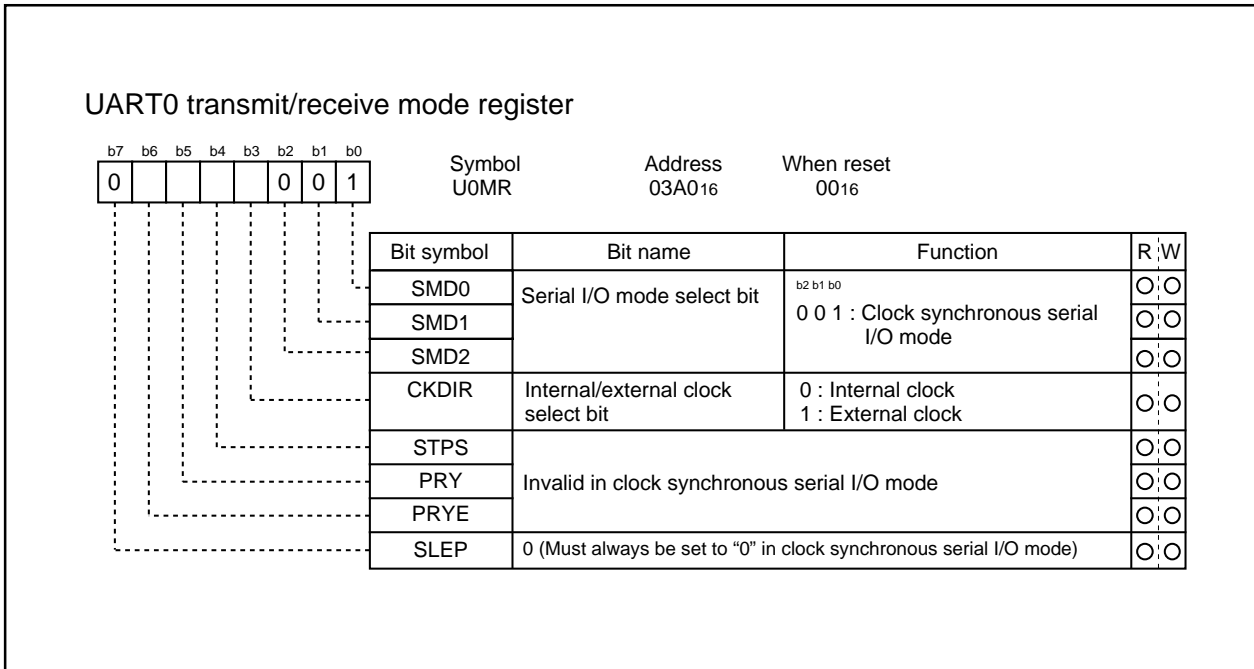


Figure 2.11.15 UART0 transmit/receive mode registers in clock synchronous serial I/O mode

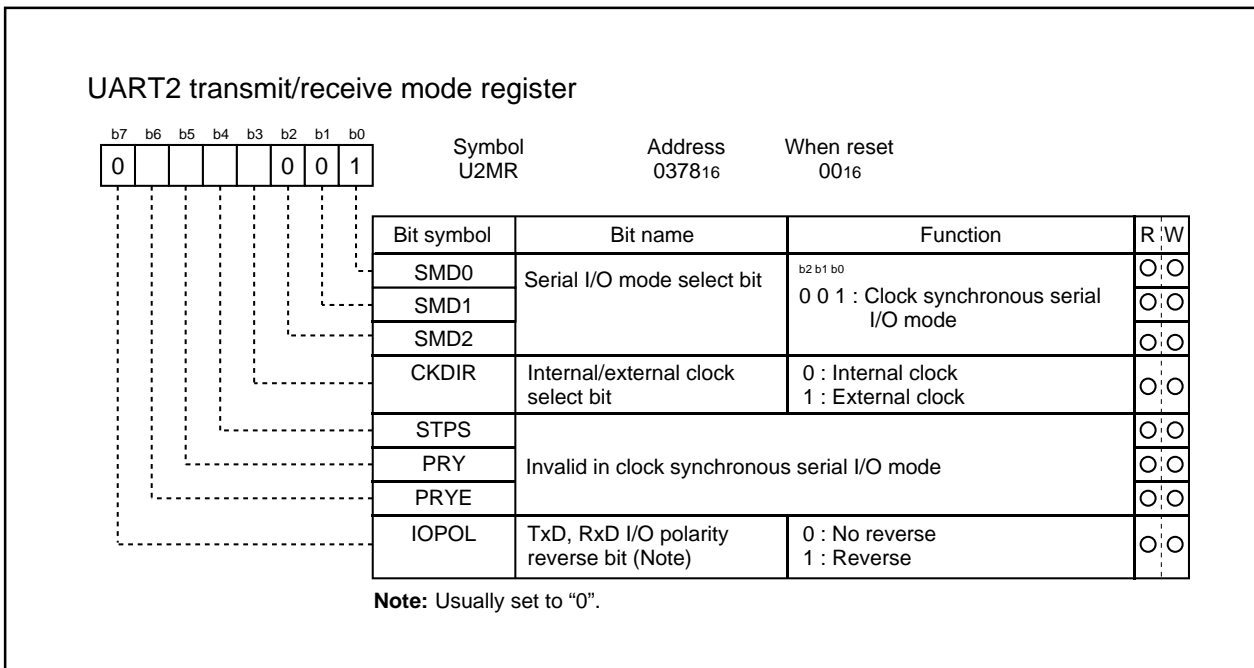


Figure 2.11.16 UART2 transmit/receive mode register in clock synchronous serial I/O mode

# M306V5ME-XXXSP

## M306V5EESP

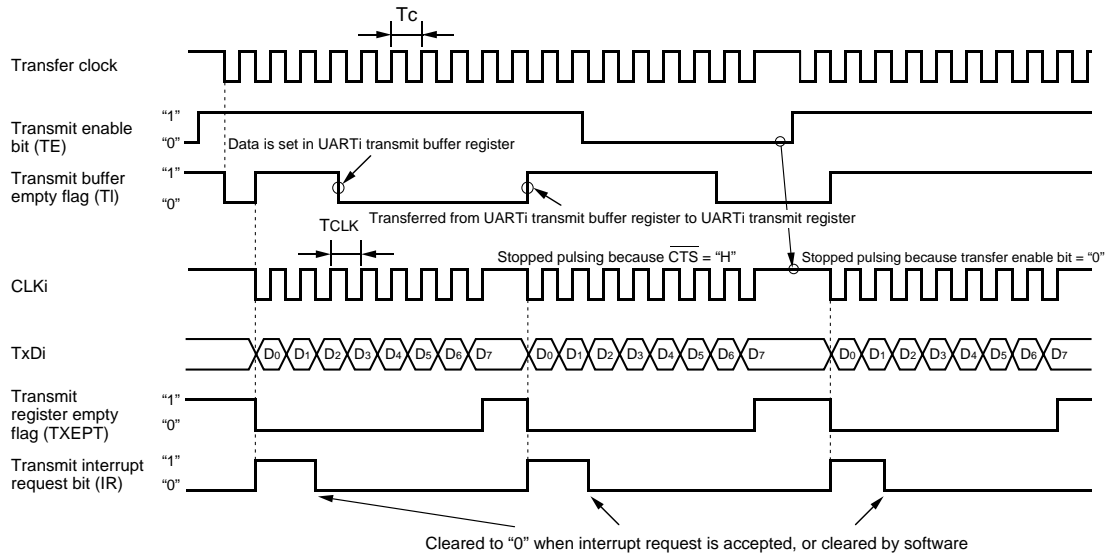
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

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 (P55, P72)	Transfer clock output	Internal/external clock select bit (bit 3 at address 03A0 16, 0378 16) = "0" Port P55 and P72 direction register (bit 5 at address 03EB 16, bit 2 at address 03EF 16) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0 16, 0378 16) = "1" Port P55 and P72 direction register (bit 5 at address 03EB 16, bit 2 at address 03EF 16) = "0"

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



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

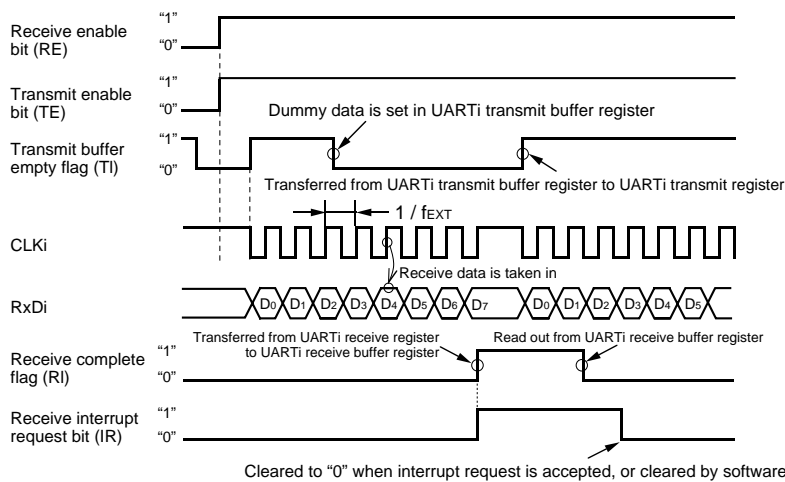
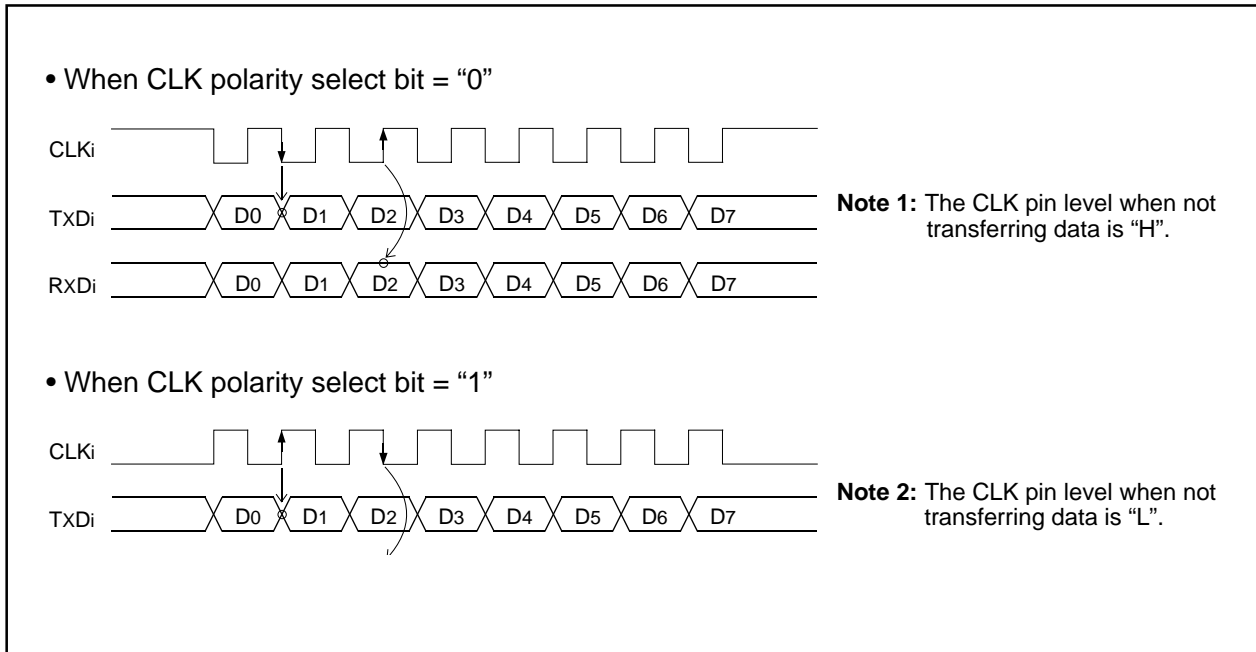


Figure 2.11.17 Typical transmit/receive timings in clock synchronous serial I/O mode

**(1) Polarity select function**

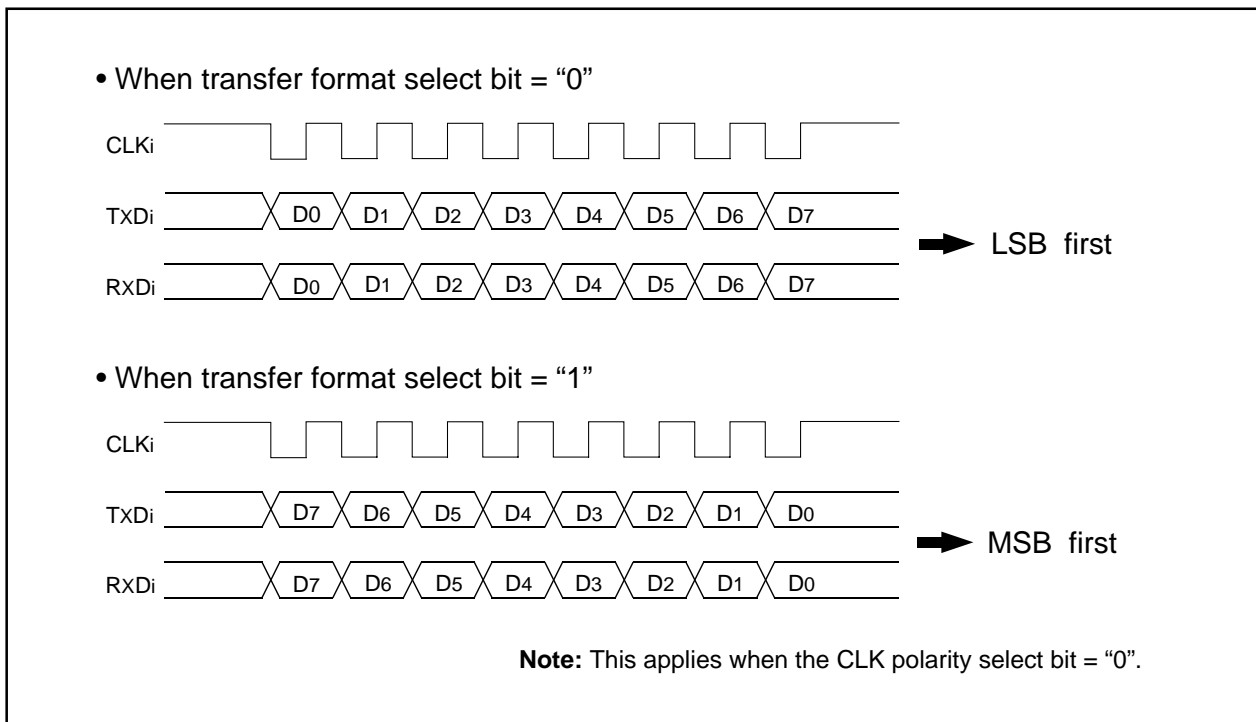
As shown in Figure 2.11.18, the CLK polarity select bit (bit 6 at addresses 03A416, 037C16) allows selection of the polarity of the transfer clock.



**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**

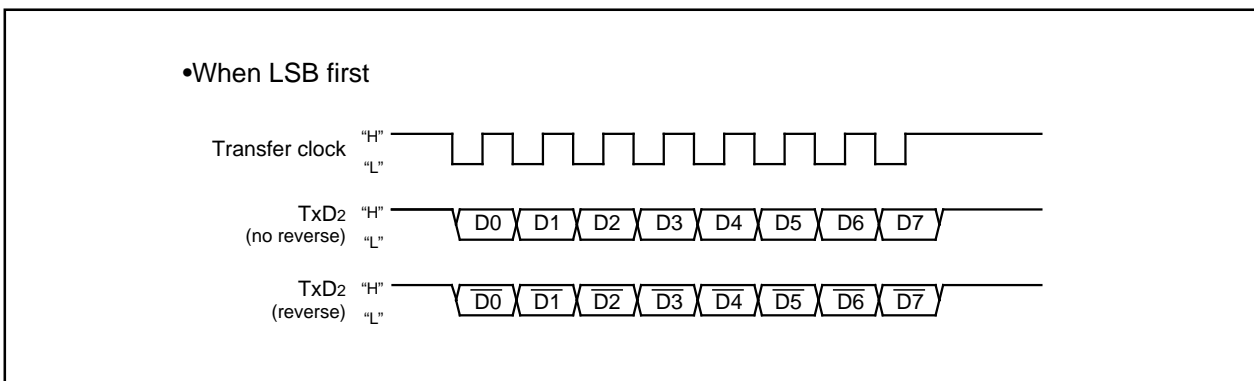


**(3) Continuous receive mode**

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") : fi/16(n+1) (Note 1) fi = f1, f8, f32</li> <li>• When external clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 0378<sub>16</sub> = "1") : fEXT/16(n+1)(Note 1) (Note 2)</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> </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>• Sleep mode selection (UART0) This mode is used to transfer data to and from one of multiple slave micro-computers</li> <li>• 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>• 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".

**M306V5ME-XXXSP**  
**M306V5EESP**

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

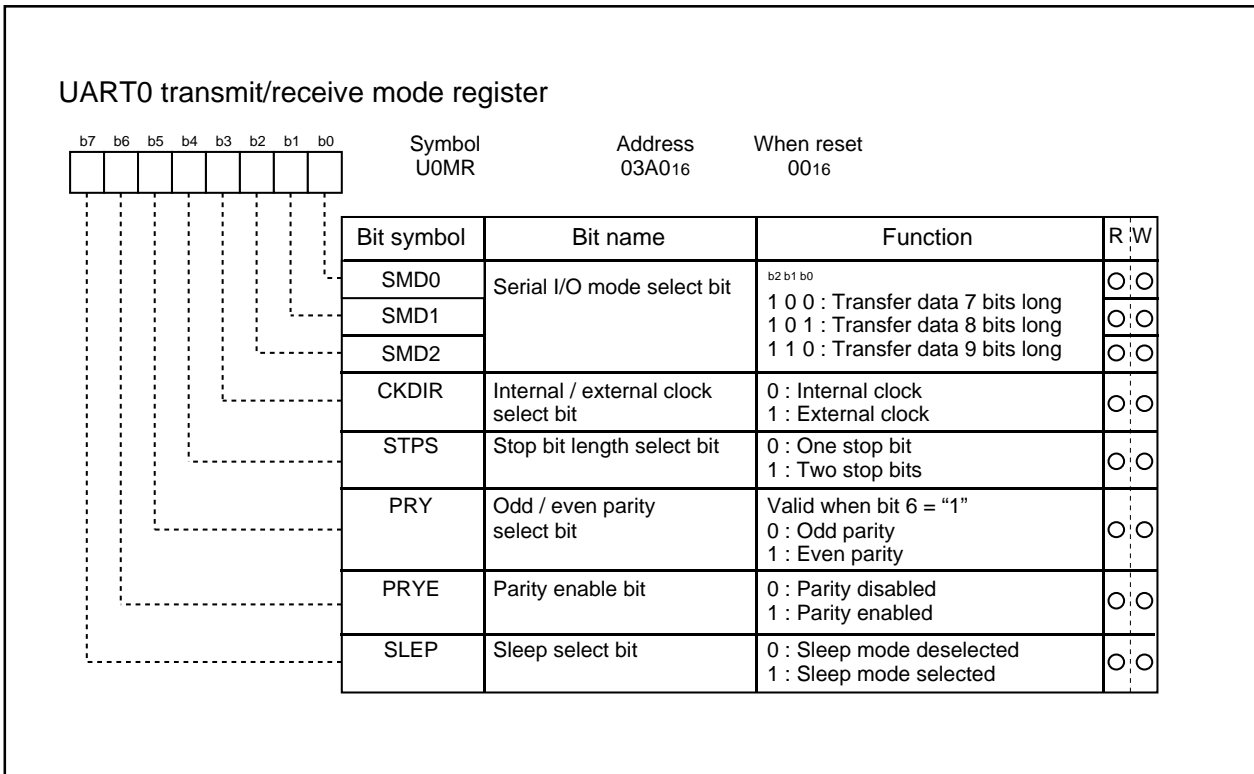


Figure 2.11.21 UART0 transmit/receive mode register in UART mode

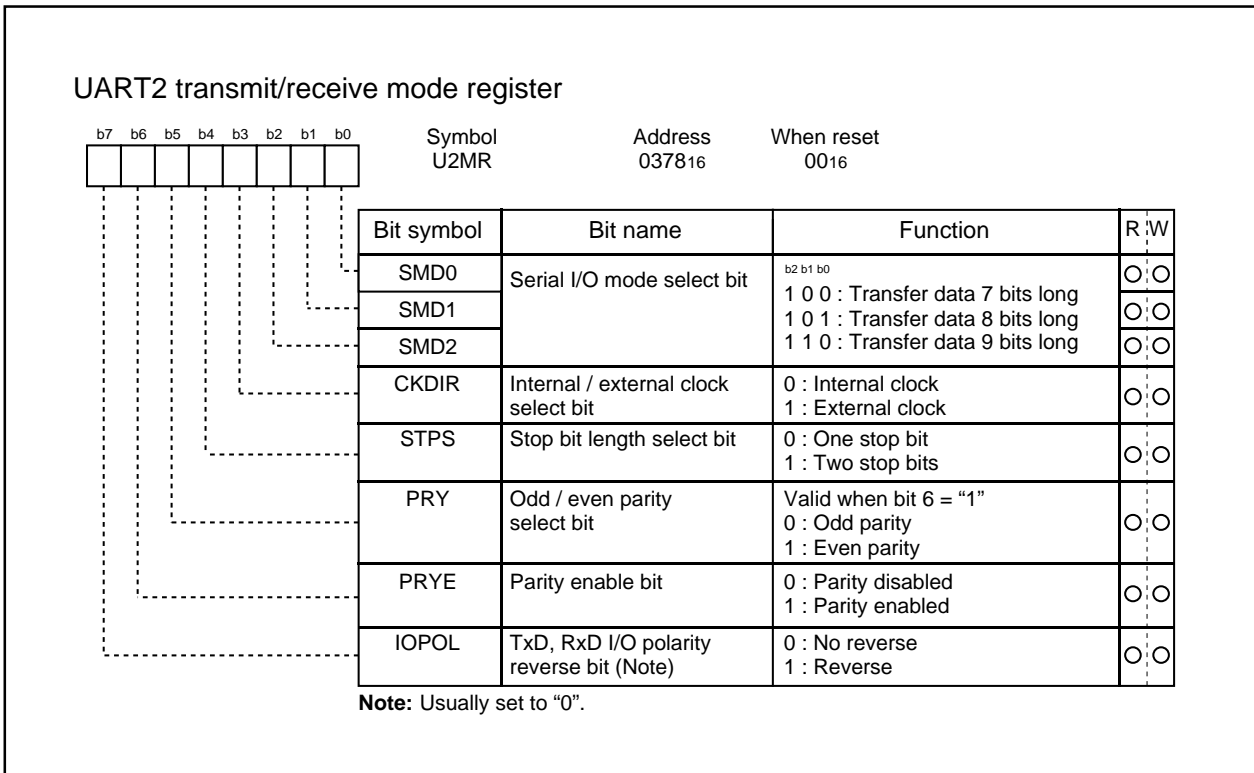


Figure 2.11.22 UART2 transmit/receive mode register in UART mode

**M306V5ME-XXXSP**  
**M306V5EESP**

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**

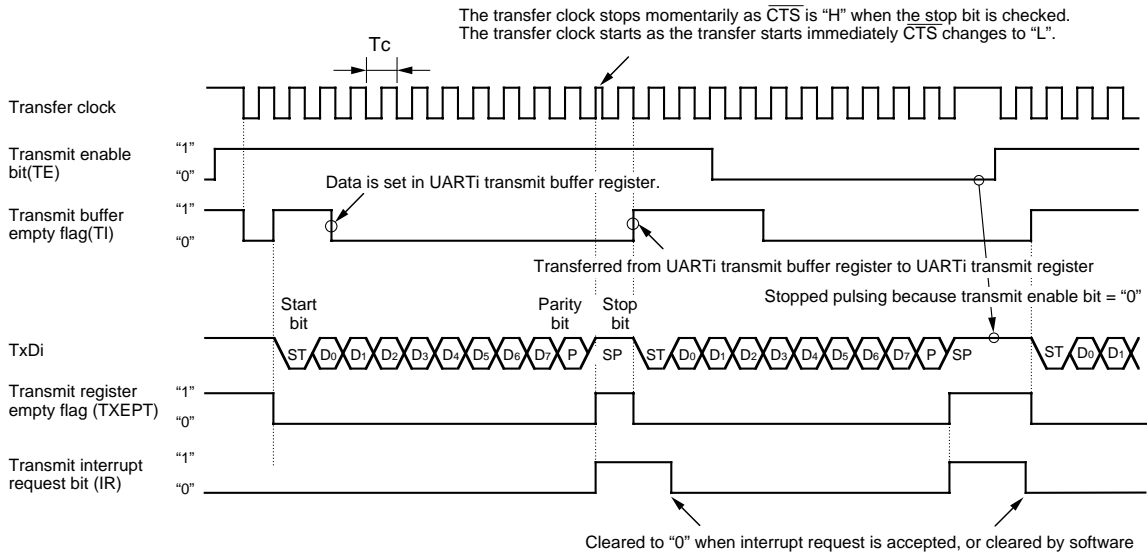
Pin name	Function	Method of selection
TxDi (P63, P70)	Serial data output	
RxDi (P62, P71)	Serial data input	Port P62 and P71 direction register (bit 2 at address 03EE <sub>16</sub> , bit 1 at address 03EF <sub>16</sub> ) = "0" (Can be used as an input port when performing transmission only)
CLKi (P55, P72)	Programmable input port	Internal/external clock select bit (bit 3 at address 03A0 <sub>16</sub> , 0378 <sub>16</sub> ) = "0" Port P55 and P72 direction register (bit 5 at address 03EB <sub>16</sub> , bit 2 at address 03EF <sub>16</sub> ) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0 <sub>16</sub> , 0378 <sub>16</sub> ) = "1" Port P55 and P72 direction register (bit 5 at address 03EB <sub>16</sub> , bit 2 at address 03EF <sub>16</sub> ) = "0"

**M306V5ME-XXXSP**  
**M306V5EESP**

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

<UART0>

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



Shown in ( ) are bit symbols.

The above timing applies to the following settings :

- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

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

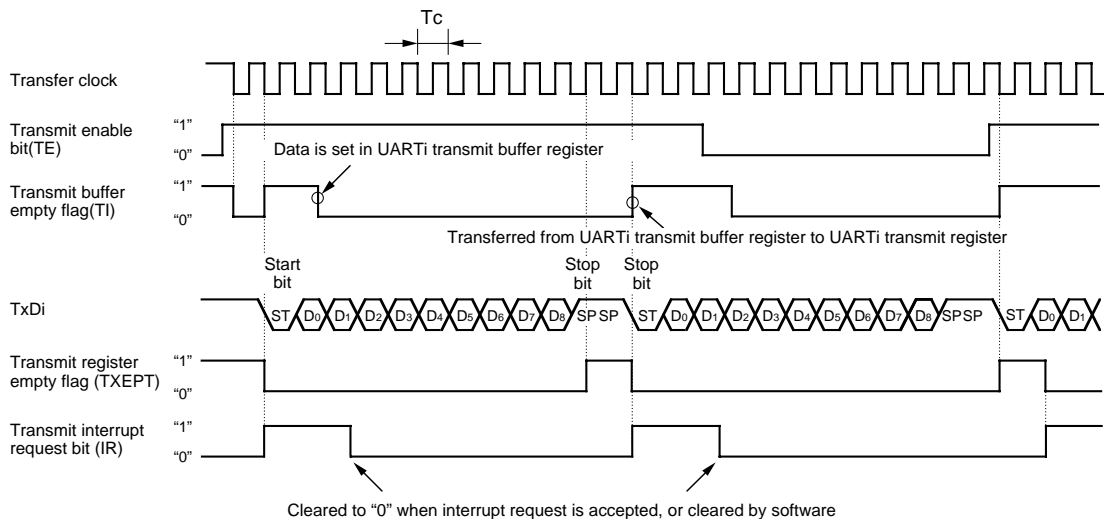
$f_i$  : frequency of BRGi count source ( $f_1, f_8, f_{32}$ )

$f_{EXT}$  : frequency of BRGi count source (external clock)

$n$  : value set to BRGi

<UART0>

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



Shown in ( ) are bit symbols.

The above timing applies to the following settings :

- Parity is disabled.
- Two stop bits.
- Transmit interrupt cause select bit = "0".

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

$f_i$  : frequency of BRGi count source ( $f_1, f_8, f_{32}$ )

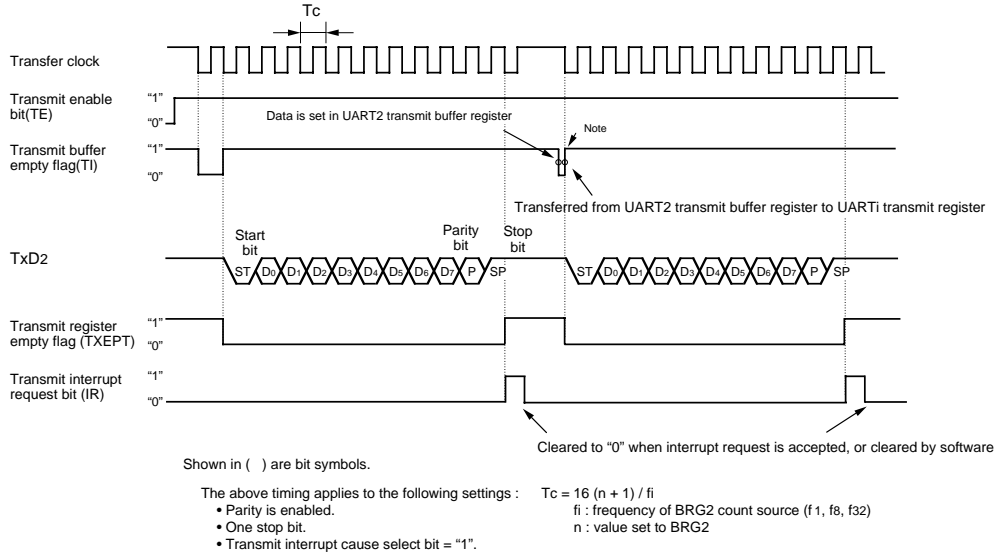
$f_{EXT}$  : frequency of BRGi count source (external clock)

$n$  : value set to BRGi

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 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)

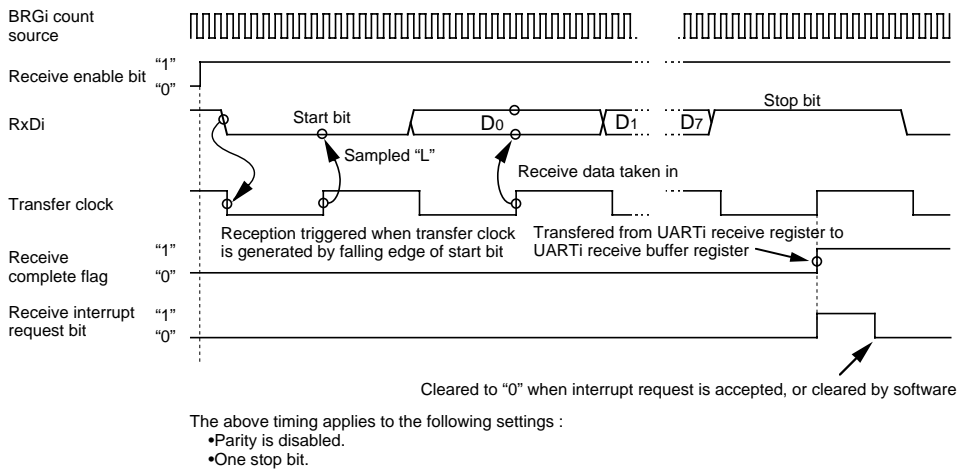


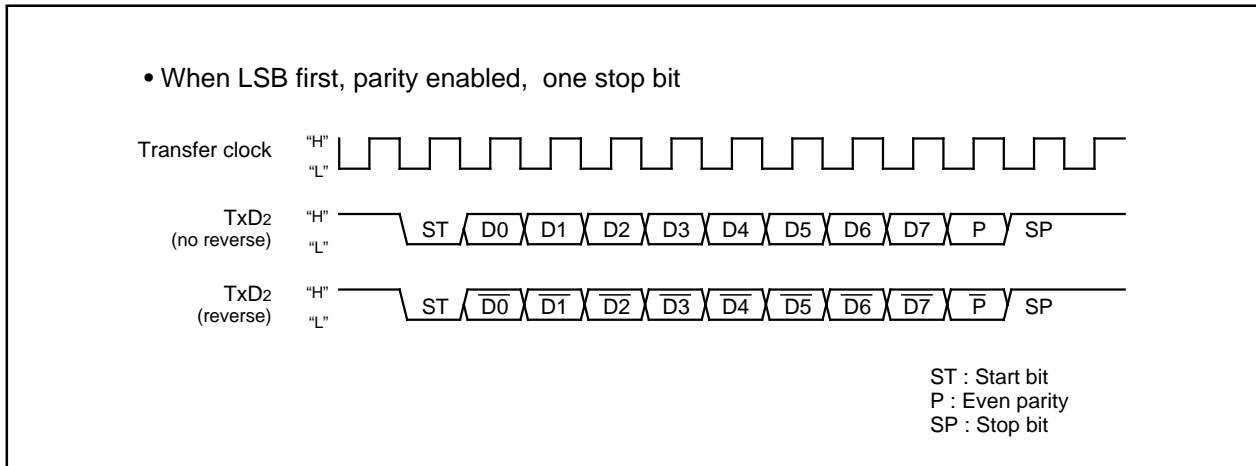
Figure 2.11.23 Typical transmit/receive timings in UART mode

**(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 03A016) 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 037D16) 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.

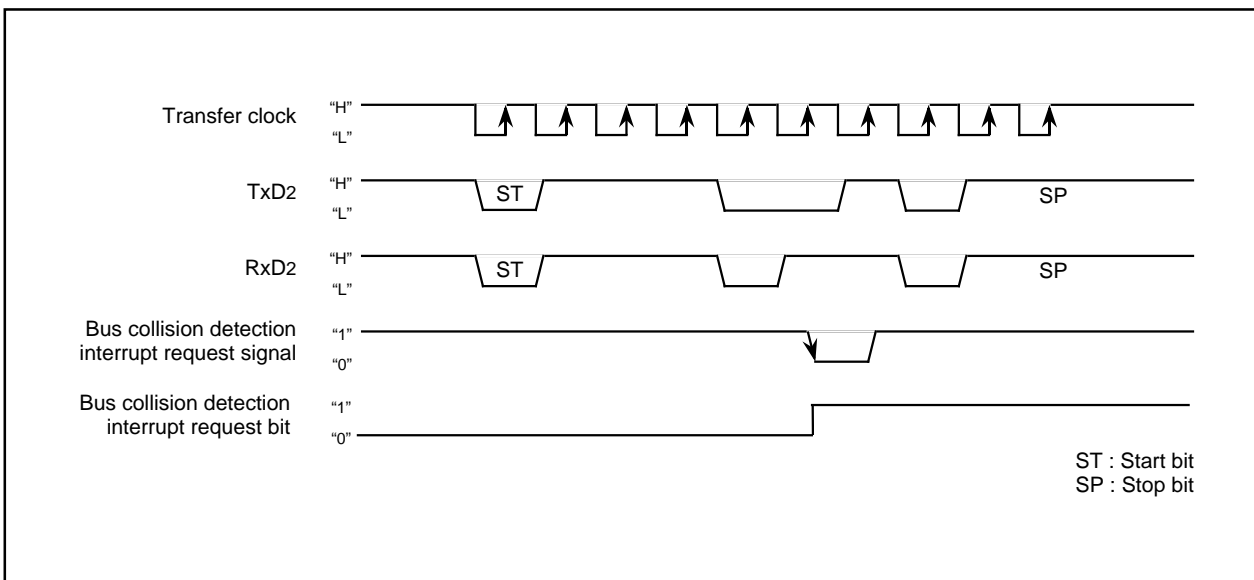


**(4) Bus collision detection function and other functions (UART2)**

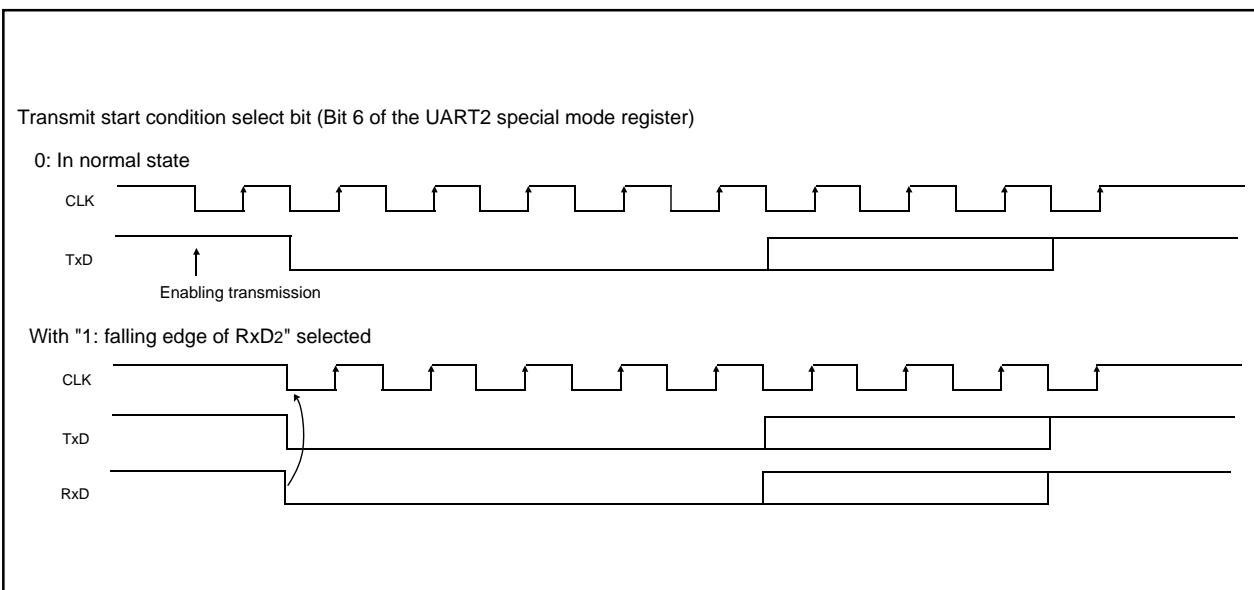
This function is to sample the output level of the TxD pin and the input level of the RxD pin at the rising edge of the transfer clock; if their values are different, then an interrupt request occurs. Figure 2.11.26 shows the example of detection timing of a buss collision (in UART mode).

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") : 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>• With an external clock chosen (bit 3 of address 0378<sub>16</sub> = "1") : f<sub>EXT</sub> / 16 (n+1) (Note 1) (Note 2)</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>

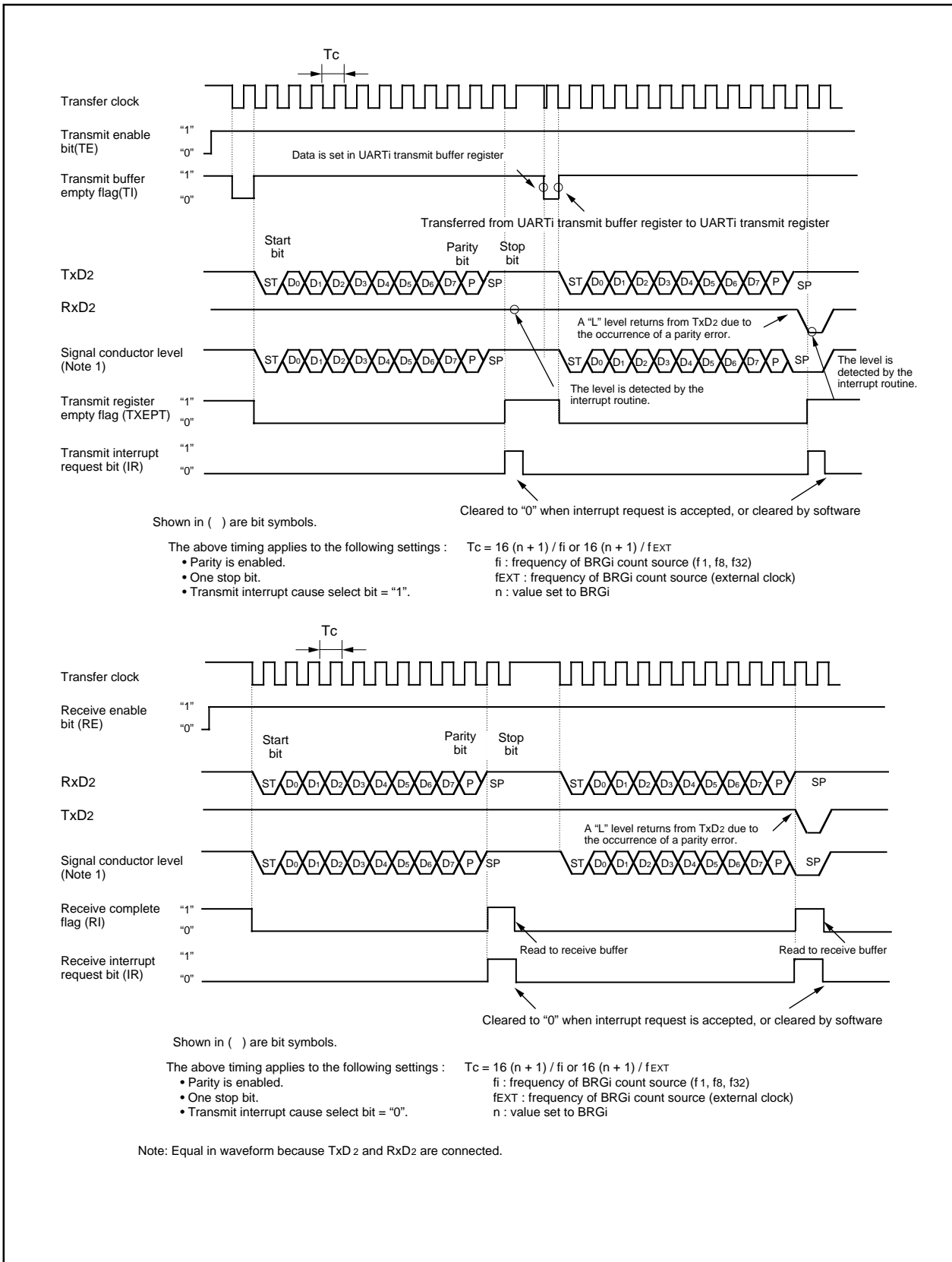
**Notes 1:** 'n' denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UARTi bit rate generator.

**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".

**M306V5ME-XXXSP**  
**M306V5EESP**

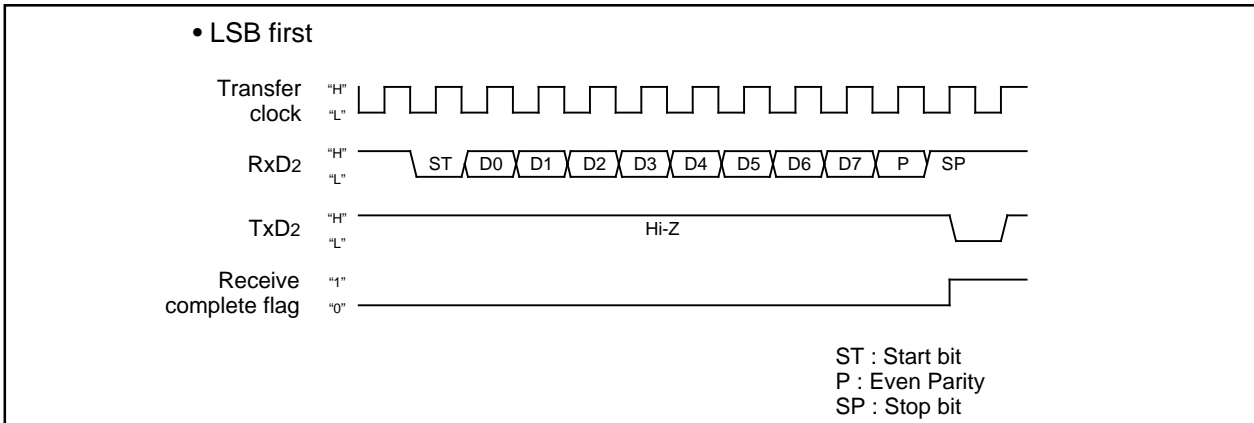
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**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.

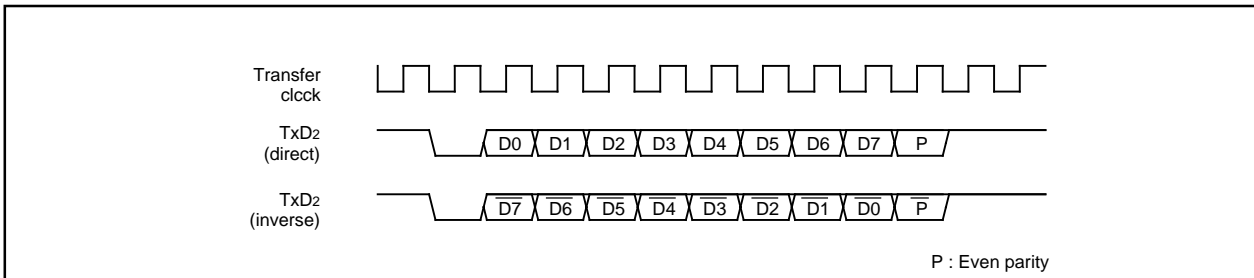


**Figure 2.11.28 Output timing of the parity error signal**

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

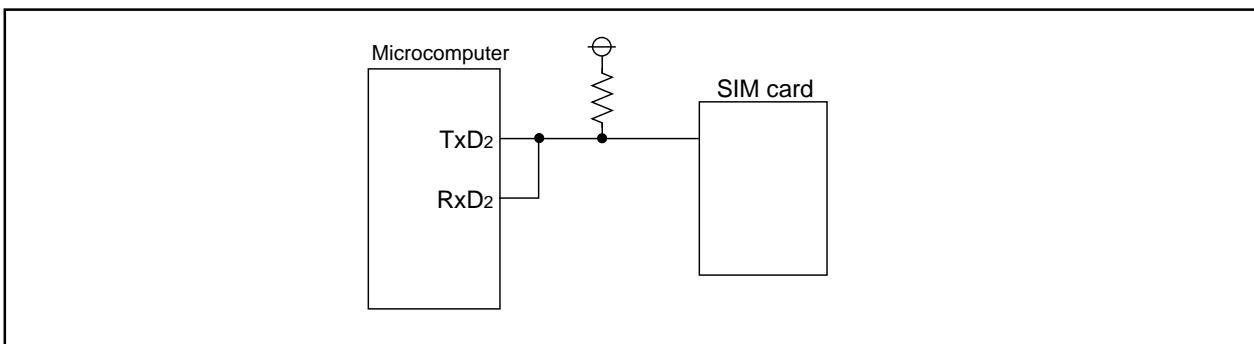
Connecting the SIM card allows you to switch between direct format and inverse format. If you choose the direct format, D0 data is output from TxD2. If you choose the inverse format, D7 data is inverted and output from TxD2.

Figure 2.11.29 shows the SIM interface format.



**Figure 2.11.29 SIM interface format**

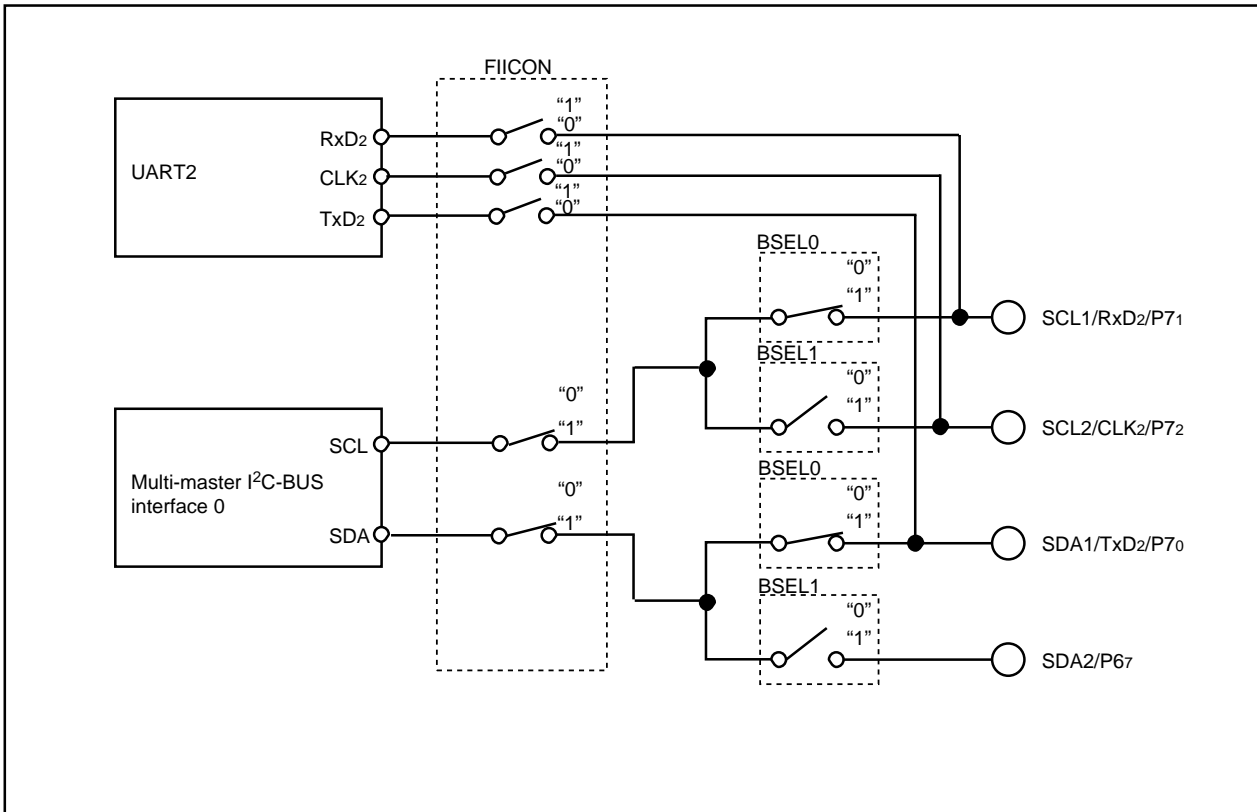
Figure 2.11.30 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.



**Figure 2.11.30 Connecting the SIM interface**

**2.11.5 Serial Interface Ports**

The I/O ports (P67, P70 to P72) function as I/O ports of UART2 and multi-master I<sup>2</sup>C-BUS interface 0 (refer to “2.11.6 Multi-master I<sup>2</sup>C-BUS interface i”) . Set the connection between both serial interfaces and each port by bits 0 and 1 (BSEL0 and BSEL1) of the peripheral mode register (address 027D16) and bit 2 (FIICON) of the I<sup>2</sup>C0 port selection register (address 02E516).



**Figure 2.11.31 Serial interface port control**

**2.11.6 Multi-master I<sup>2</sup>C-BUS Interface 0 and Multi-master I<sup>2</sup>C-BUS Interface 1**

The multi-master I<sup>2</sup>C-BUS interface 0 and 1 have each dedicated circuit and operate independently.

The multi-master I<sup>2</sup>C-BUS interface i is a serial communications circuit, conforming to the Philips I<sup>2</sup>C-BUS data transfer format. This interface i, offering both arbitration lost detection and a synchronous functions, is useful for the multi-master serial communications.

Figures 2.11.32 and Figure 2.11.33 show a block diagram of the multi-master I<sup>2</sup>C-BUS interface i and Table 2.11.13 shows multi-master I<sup>2</sup>C-BUS interface i functions.

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

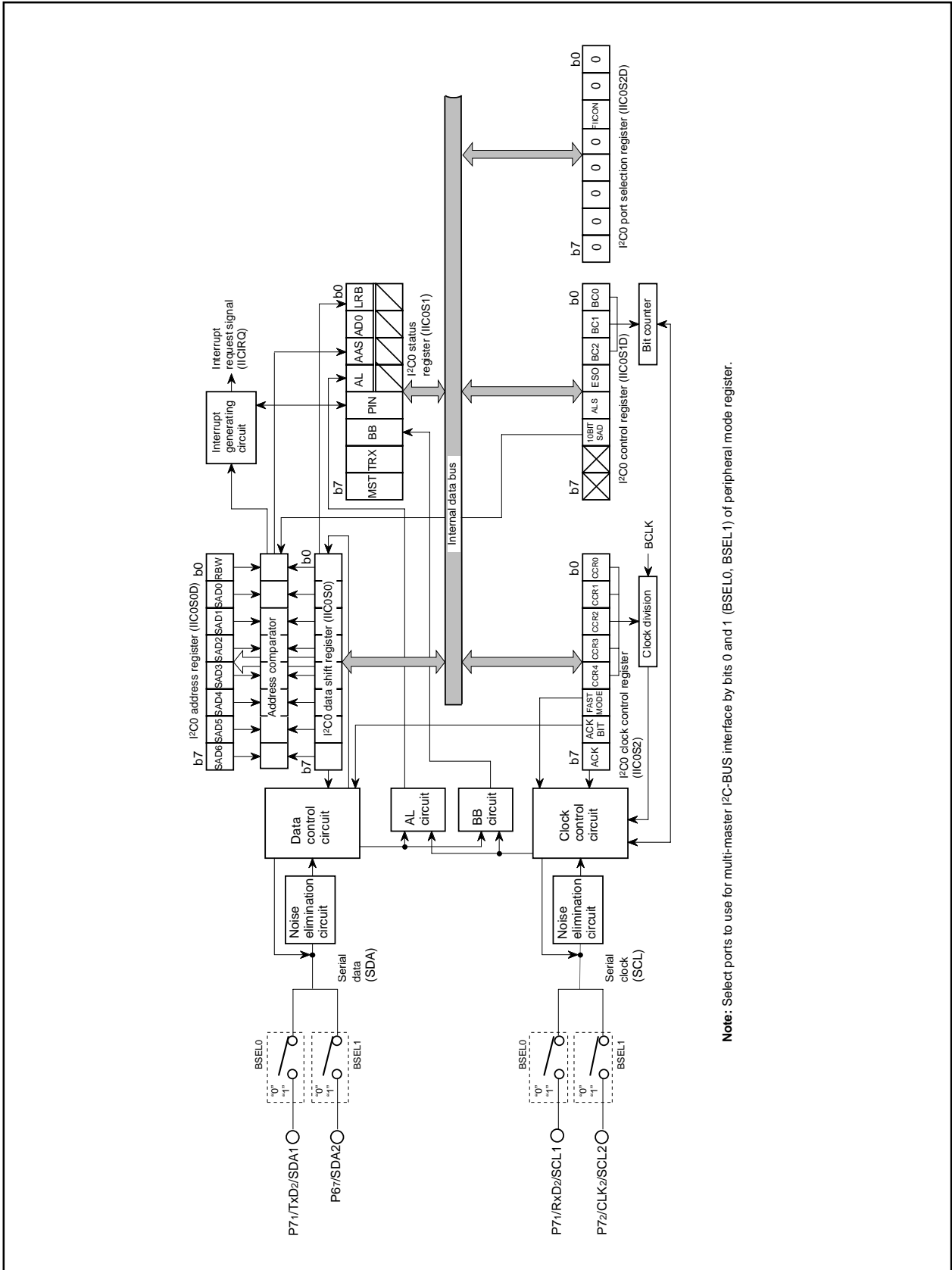
**Table 2.11.13 Multi-master I<sup>2</sup>C-BUS Interface Functions**

Item	Function
Format	In conformity with Philips I <sup>2</sup> C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	In conformity with Philips I <sup>2</sup> C-BUS standard: Master transmission Master reception Slave transmission Slave reception
SCL clock frequencyn	16.1 kHz to 400 kHz (at BCLK = 10 MHz)

**Note :** We are not responsible for any third party's infringement of patent rights or other rights attributable to the use of the control function (bits 6 and 7 of the I<sup>2</sup>C control register at address 027D16) for connections between the I<sup>2</sup>C-BUS interface 0 and ports (SCL1, SCL2, SDA1, SDA2).

**M306V5ME-XXXSP**  
**M306V5EESP**

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



**Note:** Select ports to use for multi-master I<sup>2</sup>C-BUS interface by bits 0 and 1 (BSEL0, BSEL1) of peripheral mode register.

**Fig. 2.11.32 Block Diagram of Multi-master I<sup>2</sup>C-BUS Interface 0**



**M306V5ME-XXXSP**  
**M306V5EESP**

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

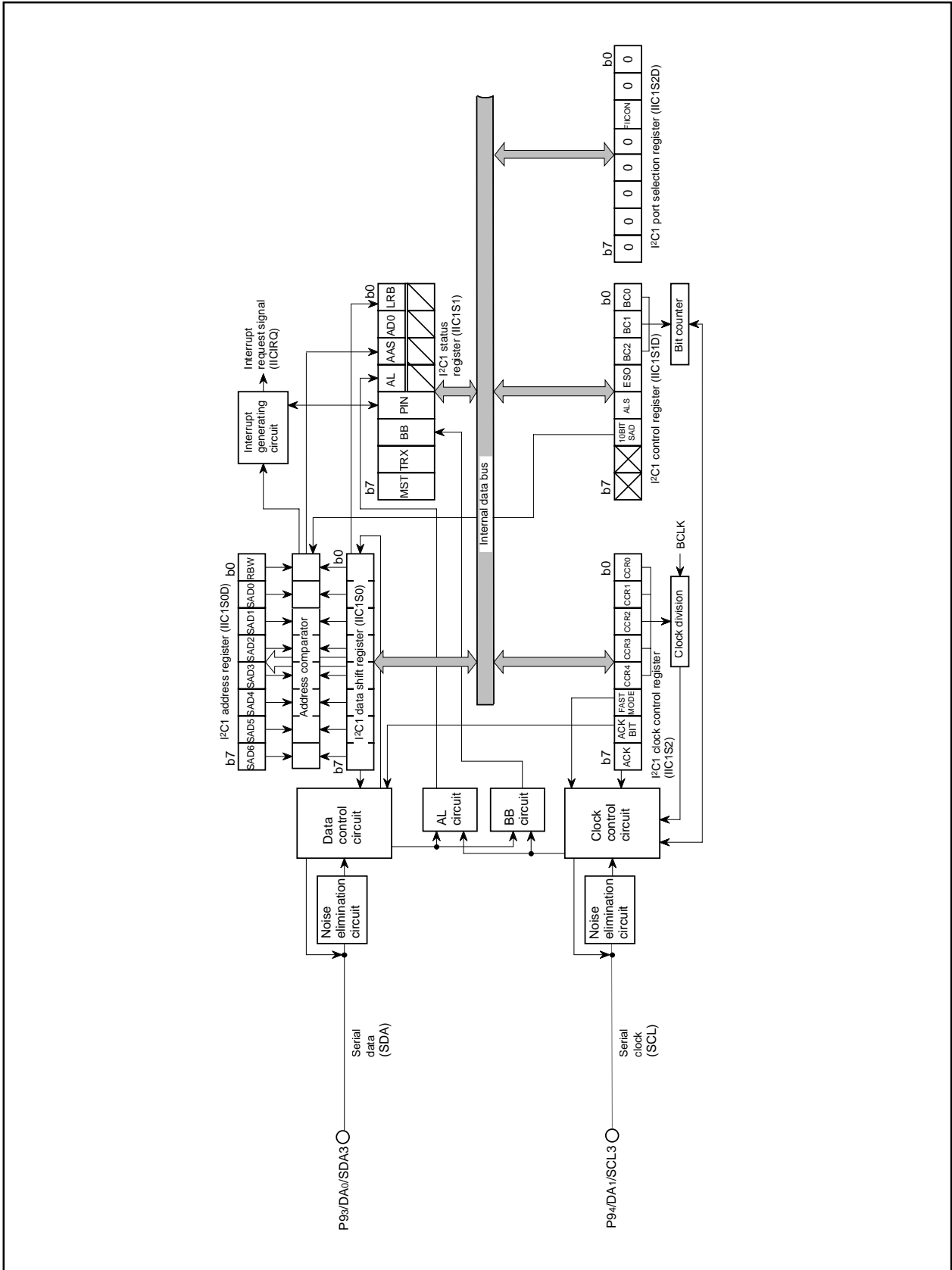


Fig. 2.11.33 Block Diagram of Multi-master I<sup>2</sup>C-BUS Interface 1

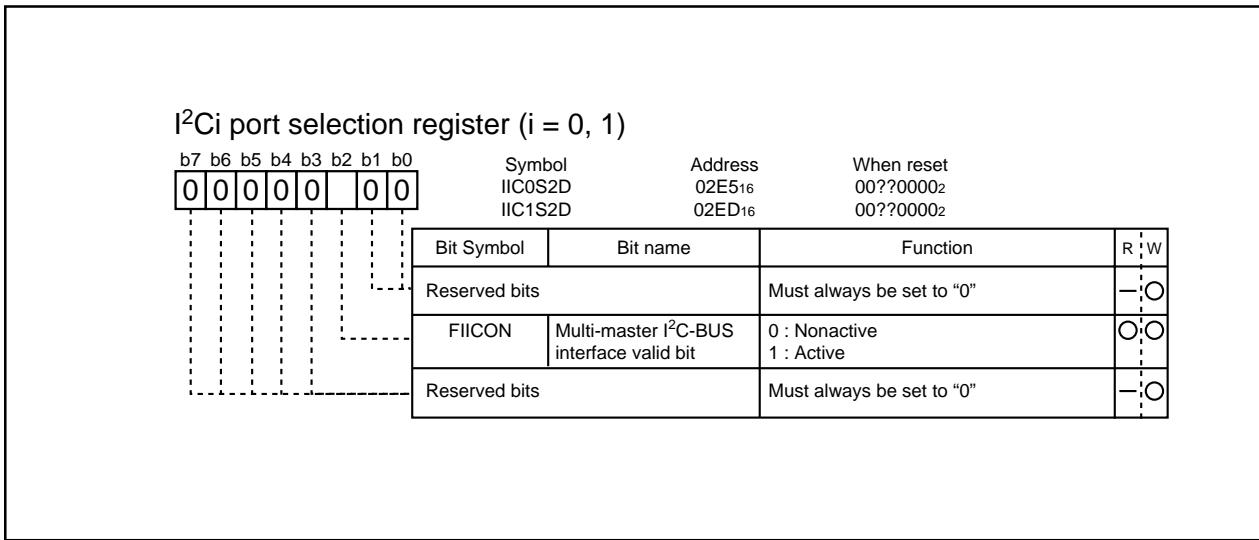
**(1) I<sup>2</sup>Ci port selection register (i = 0, 1)**

The I<sup>2</sup>Ci port selection register consists of bit to validate the multi-master I<sup>2</sup>C-BUS interface i function.

■ **Bit 2: Multi-master I<sup>2</sup>C-BUS interface valid bit (FIICON)**

When this bit is “0,” the multi-master I<sup>2</sup>C-BUS interface i is nonactive; when “1,” it is active. When selecting active, multi-master I<sup>2</sup>C-BUS interface 0 is connected with the ports selected by bits 0 and 1 of the peripheral mode register (address 027D<sub>16</sub>) and multi-master I<sup>2</sup>C-BUS interface 1 is connected with the ports P9<sub>3</sub> and P9<sub>4</sub>.

**Note:** It needs 10-BCLK cycles from setting this bit to “1” to being active of multi-master I<sup>2</sup>C-BUS interface i. Accordingly, do not access multi-master I<sup>2</sup>C-BUS interface i-related registers in this period.



**Fig. 2.11.34 I<sup>2</sup>Ci port selection register (i = 0, 1)**

**(2) I<sup>2</sup>Ci data shift register, I<sup>2</sup>Ci transmit buffer register (i = 0, 1)**

The I<sup>2</sup>Ci data shift register is an 8-bit shift register to store receive data and write transmit data.

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

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

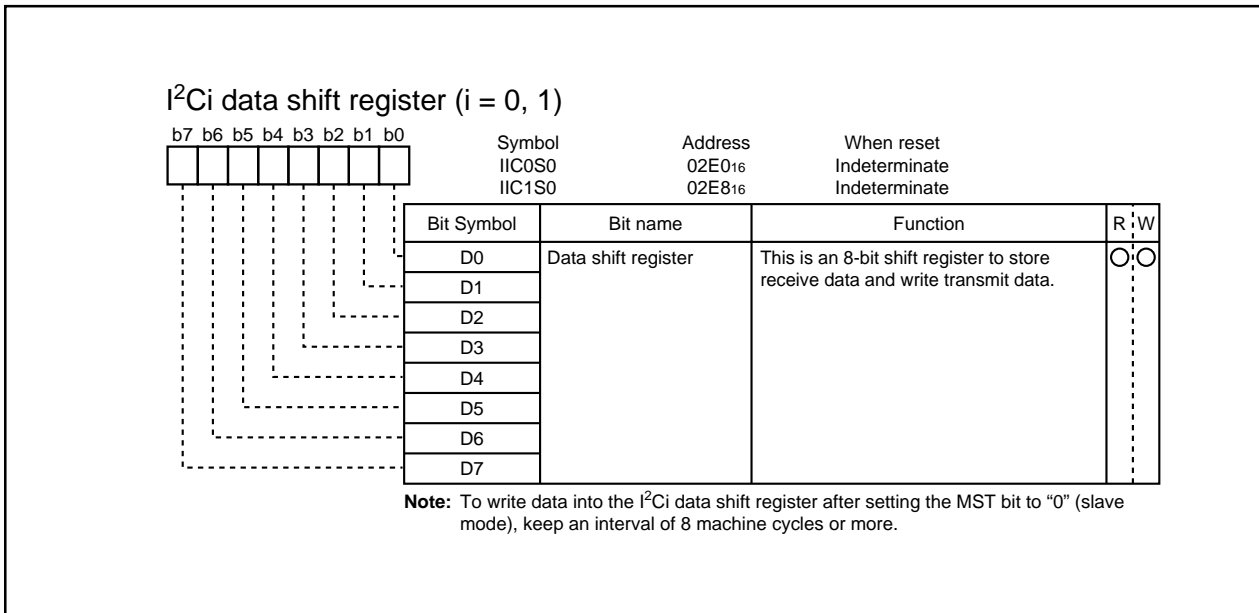
The I<sup>2</sup>Ci transmit buffer register is a register to store transmit data (slave address) to the I<sup>2</sup>Ci data shift register before RESTART condition generation. That is, in master, transmit data written to the I<sup>2</sup>Ci transmit buffer register is written to the I<sup>2</sup>Ci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

**Notes 1:** To write data into the I<sup>2</sup>Ci data shift register or the I<sup>2</sup>Ci transmit buffer register after the MST bit value changes from "1" to "0" (slave mode), keep an interval of 20 BCLK or more.

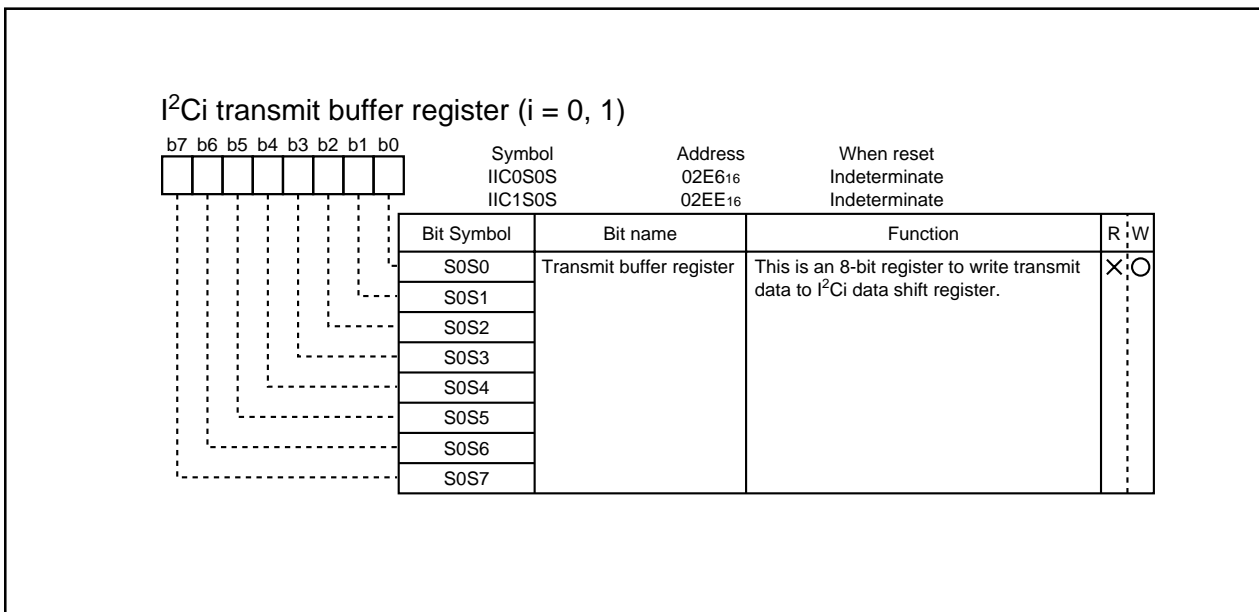
**2:** To generate START/RESTART condition after the I<sup>2</sup>Ci data shift register or the I<sup>2</sup>Ci transmit buffer register is written, keep an interval of 2 BCLK or more.

**M306V5ME-XXXSP**  
**M306V5EESP**

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



**Fig. 2.11.35 I<sup>2</sup>Ci data shift register (i = 0, 1)**



**Fig. 2.11.36 I<sup>2</sup>Ci transmit buffer register (i = 0, 1)**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**(3) I<sup>2</sup>Ci address register (i = 0, 1)**

The I<sup>2</sup>Ci address register consists of a 7-bit slave address and a  $\overline{\text{read/write}}$  bit. In the addressing mode, the slave address written in this register is compared with the address data to be received immediately after the START condition are detected.

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

Not used when comparing addresses, in the 7-bit addressing mode. In the 10-bit addressing mode, the first address data to be received is compared with the contents (SAD6 to SAD0 + RBW) of the I<sup>2</sup>Ci address register.

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

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

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

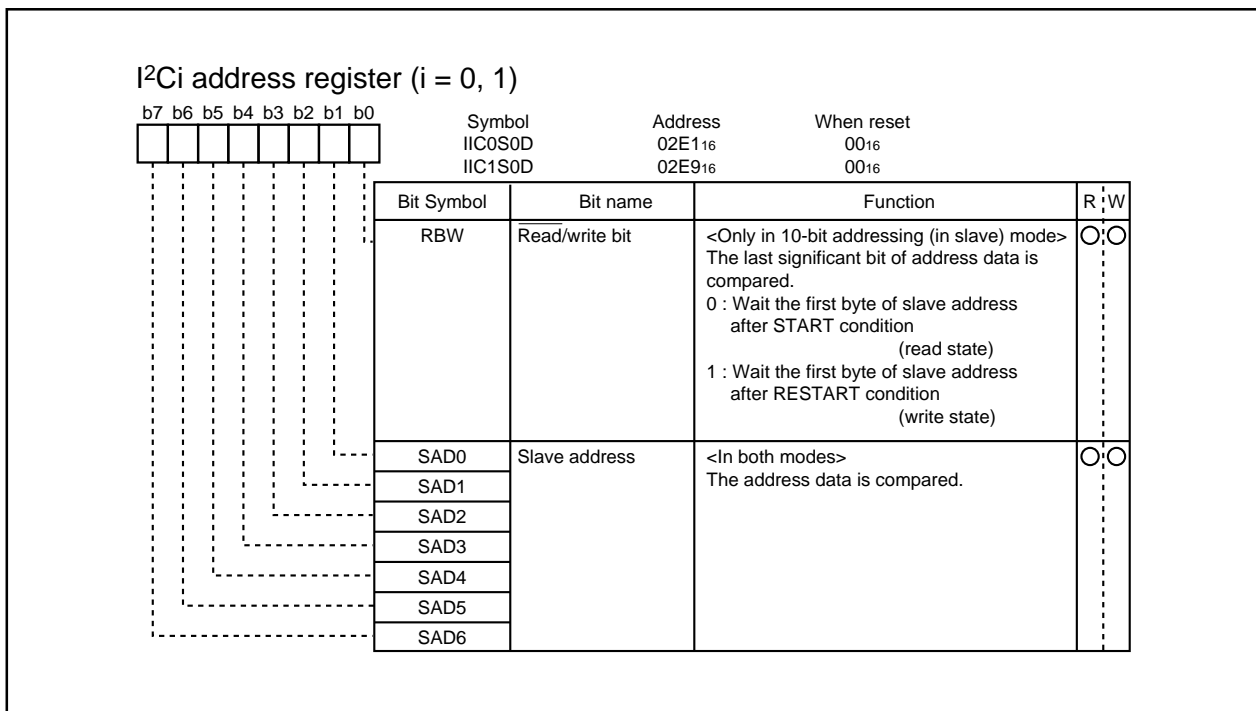


Fig. 2.11.37 I<sup>2</sup>Ci address register (i = 0, 1)

**(4) I<sup>2</sup>Ci clock control register (i = 0, 1)**

The I<sup>2</sup>Ci clock control register is used to set ACK control, SCL mode and SCL frequency.

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

These bits control the SCL frequency.

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

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

■ **Bit 6: ACK bit (ACK BIT)**

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

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

\*ACK clock: Clock for acknowledgement

■ **Bit 7: ACK clock bit (ACK)**

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

**Note:** Do not write data into the I<sup>2</sup>Ci clock control register during transmission. If data is written during transmission, the I<sup>2</sup>Ci clock generator is reset, so that data cannot be transmitted normally.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

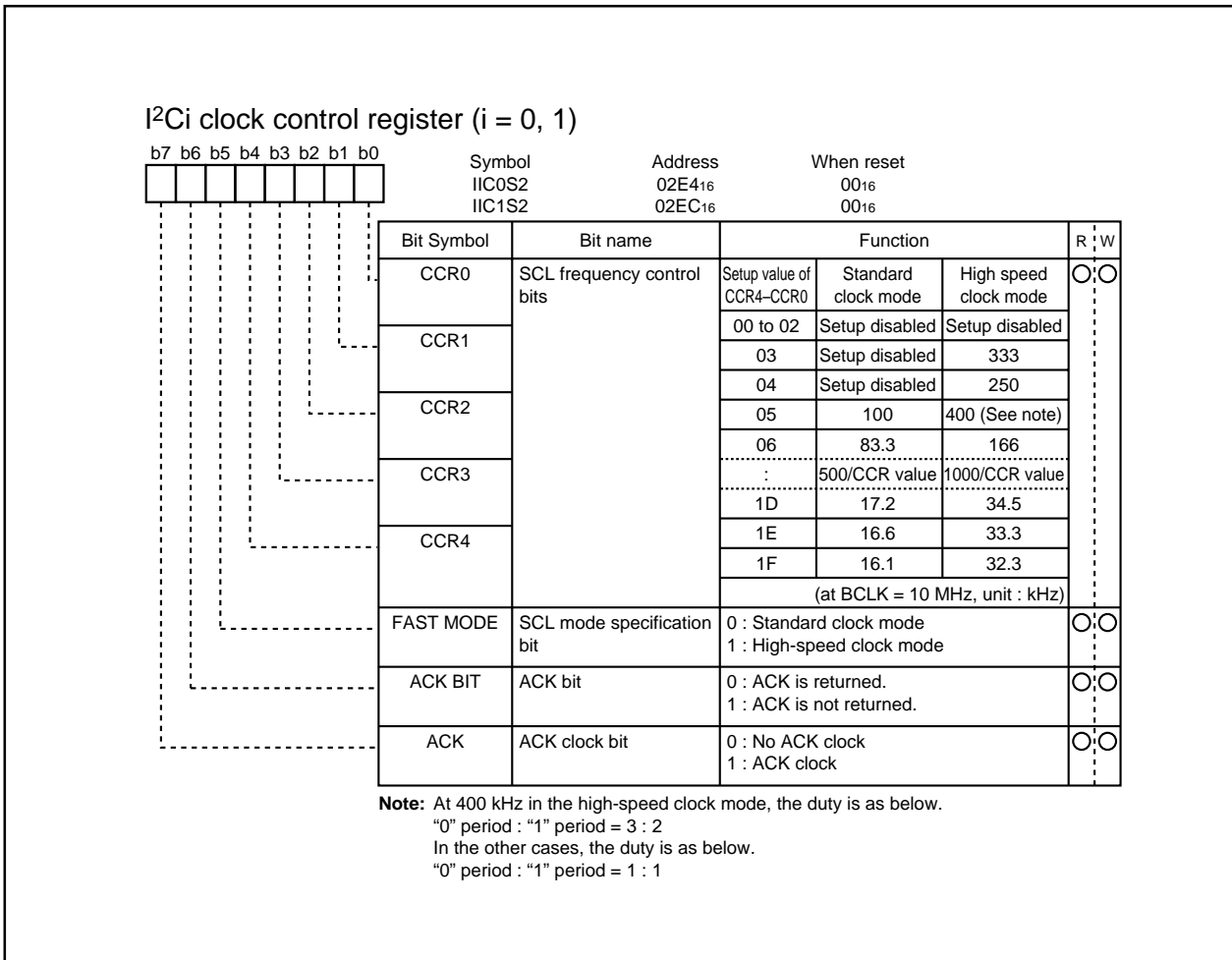


Fig. 2.11.38 I<sup>2</sup>Ci clock control register (i = 0, 1)

**(5) I<sup>2</sup>C<sub>i</sub> control register (i = 0, 1)**

The I<sup>2</sup>C<sub>i</sub> control register controls the data communication format.

■ **Bits 0 to 2: bit counter (BC0–BC2)**

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

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

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

■ **Bit 3: I<sup>2</sup>C-BUS interface i use enable bit (ESO)**

This bit enables usage of the multimaster I<sup>2</sup>C-BUS interface i. When this bit is set to “0,” the use disable status is provided, so the SDA and the SCL become high-impedance. When the bit is set to “1,” use of the interface is enabled.

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

- PIN = “1,” BB = “0” and AL = “0” are set (they are bits of the I<sup>2</sup>C<sub>i</sub> status register).
- Writing data to the I<sup>2</sup>C<sub>i</sub> data shift register and the I<sup>2</sup>C<sub>i</sub> transmit buffer register is disabled.

■ **Bit 4: data format selection bit (ALS)**

This bit decides whether or not to recognize slave addresses. When this bit is set to “0,” the addressing format is selected, so that address data is recognized. When a match is found between a slave address and address data as a result of comparison or when a general call (refer to “(6) I<sup>2</sup>C<sub>i</sub> status register,” bit 1) is received, transmission processing can be performed. When this bit is set to “1,” the free data format is selected, so that slave addresses are not recognized.

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

This bit selects a slave address specification format. When this bit is set to “0,” the 7-bit addressing format is selected. In this case, only the high-order 7 bits (slave address) of the I<sup>2</sup>C<sub>i</sub> address register are compared with address data. When this bit is set to “1,” the 10-bit addressing format is selected, all the bits of the I<sup>2</sup>C<sub>i</sub> address register are compared with address data.



**M306V5ME-XXXSP**  
**M306V5EESP**

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

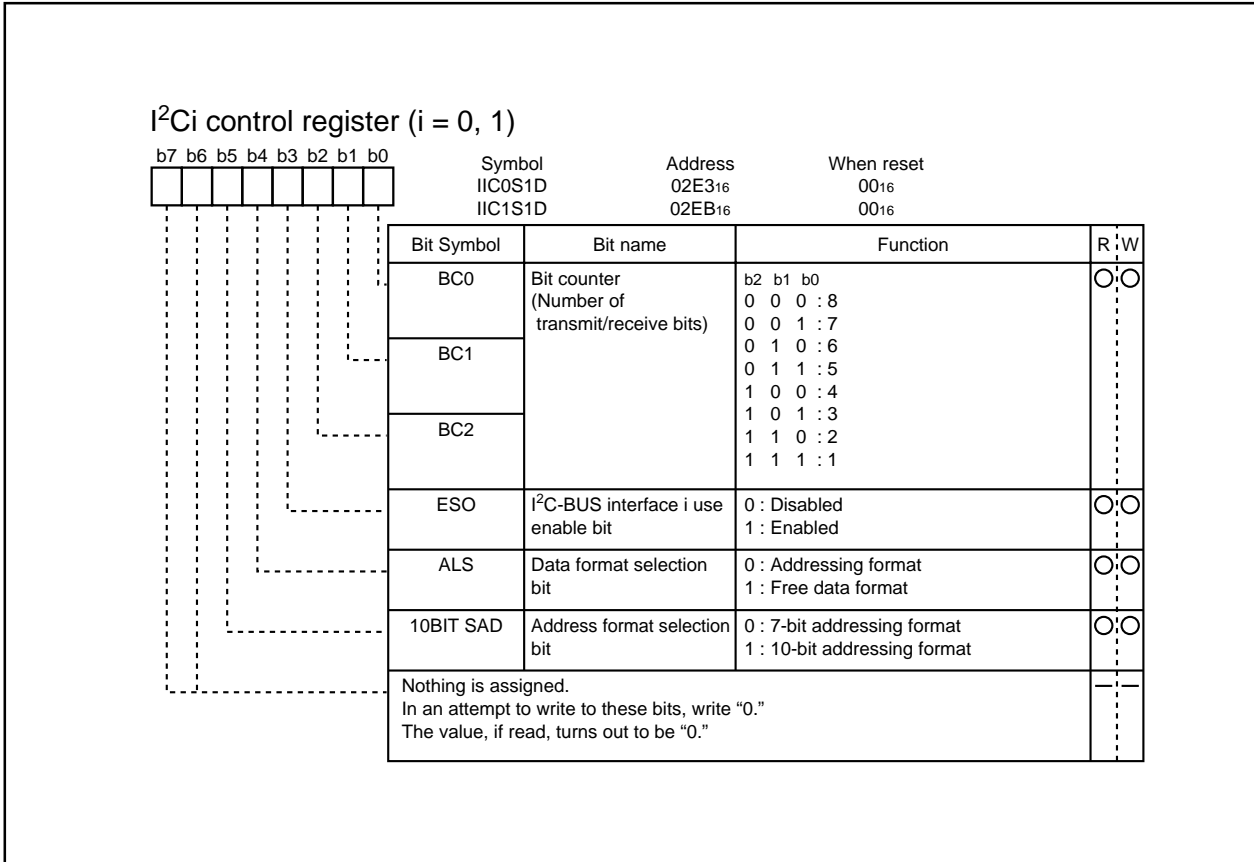


Fig. 2.11.39 I<sup>2</sup>Ci control register (i = 0, 1)

**(6) I<sup>2</sup>Ci status register (i = 0, 1)**

The I<sup>2</sup>Ci status register controls the I<sup>2</sup>C-BUS interface i status. Bits 0 to 3, 5 are read-only bits and bits 4, 6, 7 can be read out and written to.

**■ Bit 0: last receive bit (LRB)**

This bit stores the last bit value of received data and can also be used for ACK receive confirmation. If ACK is returned when an ACK clock occurs, the LRB bit is set to "0." If ACK is not returned, this bit is set to "1." Except in the ACK mode, the last bit value of received data is input. The state of this bit is changed from "1" to "0" by executing a write instruction to the I<sup>2</sup>Ci data shift register or the I<sup>2</sup>Ci transmit buffer register.

**■ Bit 1: general call detecting flag (AD0)**

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

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

**■ Bit 2: slave address comparison flag (AAS)**

This flag indicates a comparison result of address data.

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

- The address data immediately after occurrence of a START condition matches the slave address stored in the high-order 7 bits of the I<sup>2</sup>Ci address register.
- A general call is received.

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

- When the address data is compared with the I<sup>2</sup>Ci address register (8 bits consists of slave address and RBW), the first bytes match.

<<The state of this bit is changed from "1" to "0" by executing a write instruction to the I<sup>2</sup>Ci data shift register or the I<sup>2</sup>Ci transmit buffer register.>>

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

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

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

**■ Bit 4: I<sup>2</sup>C-BUS interface i interrupt request bit (PIN)**

This bit generates an interrupt request signal. Each time 1-byte data is transmitted, the state of the PIN bit changes from “1” to “0.” At the same time, an interrupt request signal is sent to the CPU. The PIN bit is set to “0” in synchronization with a falling edge of the last clock (including the ACK clock) of an internal clock and an interrupt request signal occurs in synchronization with a falling edge of the PIN bit. When detecting the STOP condition in slave, the multi-master I<sup>2</sup>C-BUS interface interrupt request bit (IR) is set to “1” (interrupt requested) regardless of falling of PIN bit. When the PIN bit is “0,” the SCL is kept in the “0” state and clock generation is disabled. Figure 2.11.41 shows an interrupt request signal generating timing chart.

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

- Writing “1” to the PIN bit
- Executing a write instruction to the I<sup>2</sup>Ci data shift register or the I<sup>2</sup>Ci transmit buffer register (See note).
- When the ESO bit is “0”
- At reset

**Note :** It takes 8 BCLK cycles or more until PIN bit becomes “1” after write instructions are executed to these registers.

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

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

**■ Bit 5: bus busy flag (BB)**

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

This flag can be written by software only in the master transmission mode. In the other modes, this bit is set to “1” by detecting a START condition and set to “0” by detecting a STOP condition. When the ESO bit of the I<sup>2</sup>Ci control register is “0” and at reset, the BB flag is kept in the “0” state.

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

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

When the ALS bit of the I<sup>2</sup>Ci control register is “0” in the slave reception mode is selected, the TRX bit is set to “1” (transmit) if the least significant bit (R/ $\bar{W}$  bit) of the address data transmitted by the master is “1.” When the ALS bit is “0” and the R/ $\bar{W}$  bit is “0,” the TRX bit is cleared to “0” (receive).

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

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

**M306V5ME-XXXSP**  
**M306V5EESP**

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

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

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

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

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

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

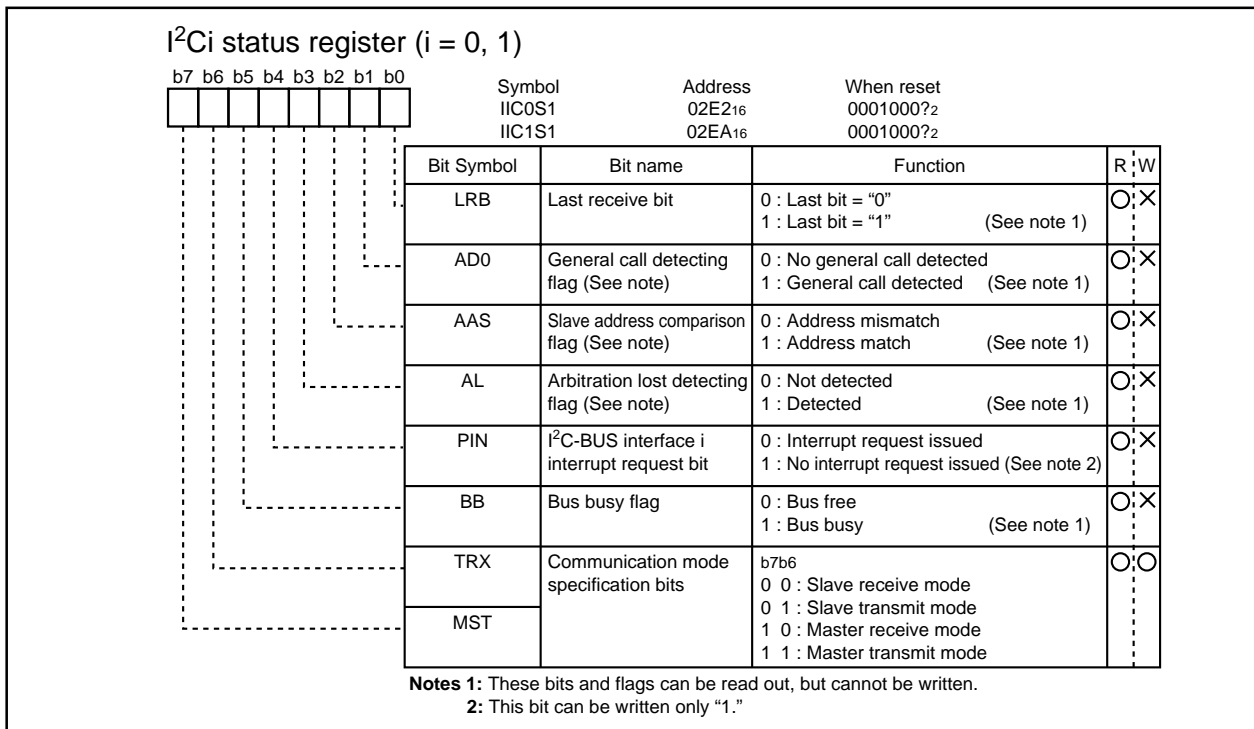


Fig. 2.11.40 I<sup>2</sup>Ci status register (i = 0, 1)

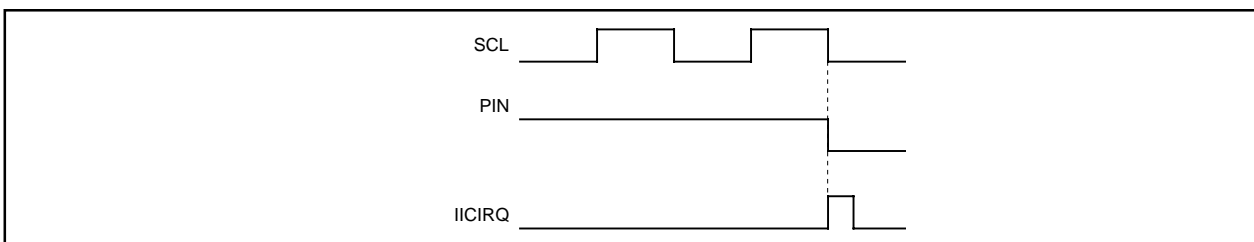
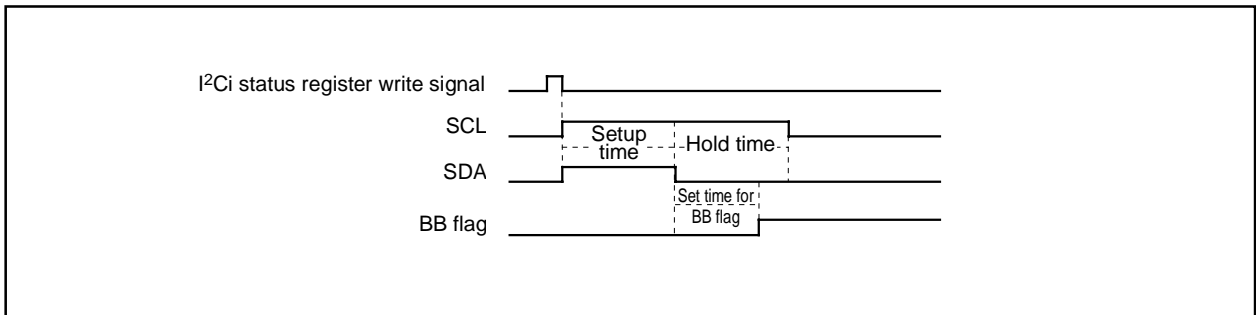


Fig. 2.11.41 Interrupt request signal generation timing

**(7) START condition generation method**

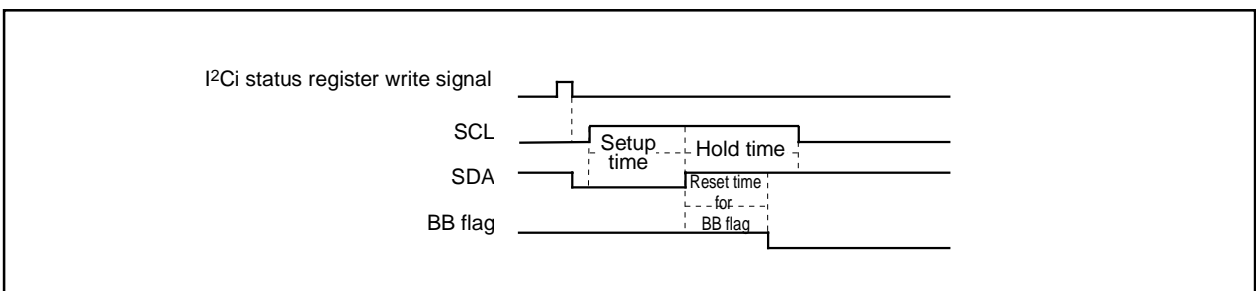
When the ESO bit of the I<sup>2</sup>Ci control register is “1,” execute a write instruction to the I<sup>2</sup>Ci status register to set the MST, TRX and BB bits to “1.” A START condition will then be generated. After that, the bit counter becomes “0002” and an SCL for 1 byte is output. The START condition generation timing and BB bit set timing are different in the standard clock mode and the high-speed clock mode. Refer to Figure 2.11.42 for the START condition generation timing diagram, and Table 2.11.13 for the START condition/STOP condition generation timing table.



**Fig. 2.11.42 START condition generation timing diagram**

**(8) STOP condition generation method**

When the ESO bit of the I<sup>2</sup>Ci control register is “1,” execute a write instruction to the I<sup>2</sup>Ci status register for setting the MST bit and the TRX bit to “1” and the BB bit to “0”. A STOP condition will then be generated. The STOP condition generation timing and the BB flag reset timing are different in the standard clock mode and the high-speed clock mode. Refer to Figure 2.11.43 for the STOP condition generation timing diagram, and Table 2.11.13 for the START condition/STOP condition generation timing table.



**Fig. 2.11.43 STOP condition generation timing diagram**

**Table 2.11.13 START condition/STOP condition generation timing table**

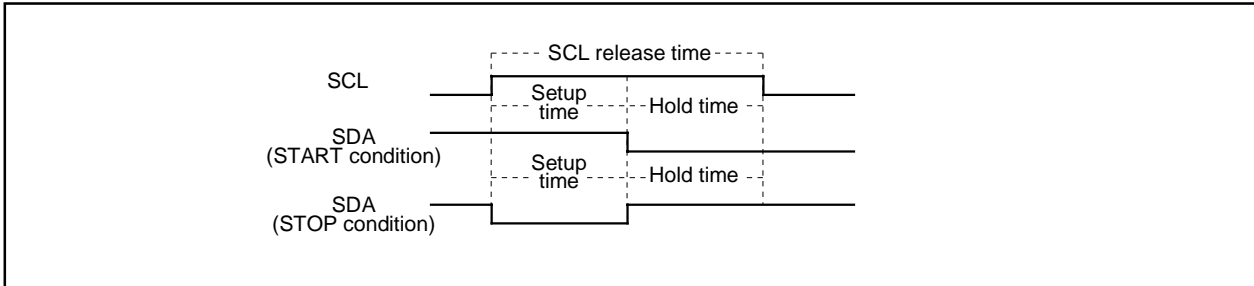
Item	Standard Clock Mode	High-speed Clock Mode
Setup time	5.35 μs (53.5 cycles)	1.85 μs (18.5 cycles)
Hold time	4.9 μs (49 cycles)	2.4 μs (24 cycles)
Set/reset time for BB flag	3.75 μs (37.5 cycles)	0.85 μs (8.5 cycles)

**Note:** Absolute time at BCLK = 10 MHz. The value in parentheses denotes the number of BCLK cycles.

**(9) START/STOP condition detect conditions**

The START/STOP condition detect conditions are shown in Figure 2.11.44 and Table 2.11.14. Only when the 3 conditions of Table 2.11.14 are satisfied, a START/STOP condition can be detected.

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



**Fig. 2.11.44 START condition/STOP condition detect timing diagram**

**Table 2.11.14 START condition/STOP condition detect conditions**

Standard Clock Mode	High-speed Clock Mode
6.5 μs (65 cycles) < SCL release time	1.0 μs (10 cycles) < SCL release time
3.25 μs (32.5 cycles) < Setup time	0.5 μs (5 cycles) < Setup time
3.25 μs (32.5 cycles) < Hold time	0.5 μs (5 cycles) < Hold time

**Note:** Absolute time at BCLK = 10 MHz. The value in parentheses denotes the number of BCLK cycles.

**(10) Address data communication**

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

**■ 7-bit addressing format**

To meet the 7-bit addressing format, set the 10BIT SAD bit of the I<sup>2</sup>Ci control register to "0." The first 7-bit address data transmitted from the master is compared with the high-order 7-bit slave address stored in the I<sup>2</sup>Ci address register. At the time of this comparison, address comparison of the RBW bit of the I<sup>2</sup>Ci address register is not made. For the data transmission format when the 7-bit addressing format is selected, refer to Figure 2.11.45, (1) and (2).

**■ 10-bit addressing format**

To meet the 10-bit addressing format, set the 10BIT SAD bit of the I<sup>2</sup>Ci control register to "1." An address comparison is made between the first-byte address data transmitted from the master and the 7-bit slave address stored in the I<sup>2</sup>Ci address register. At the time of this comparison, an address comparison between the RBW bit of the I<sup>2</sup>Ci address register and the R/ $\bar{W}$  bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the R/ $\bar{W}$  bit which is the last bit of the address data not only specifies the direction of communication for control data but also is processed as an address data bit.

When the first-byte address data matches the slave address, the AAS bit of the I<sup>2</sup>Ci status register is set to "1." After the second-byte address data is stored into the I<sup>2</sup>Ci data shift register, make an address comparison between the second-byte data and the slave address by software. When the address data of the 2nd bytes matches the slave address, set the RBW bit of the I<sup>2</sup>Ci address register to "1" by software. This processing can match the 7-bit slave address and R/ $\bar{W}$  data, which are received after a RESTART condition is detected, with the value of the I<sup>2</sup>Ci address register. For the data transmission format when the 10-bit addressing format is selected, refer to Figure 2.11.45, (3) and (4).

**(11) Example of Master Transmission**

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

- ① Set a slave address in the high-order 7 bits of the I<sup>2</sup>Ci address register and "0" in the RBW bit.
- ② Set the ACK return mode and SCL = 100 kHz by setting "85<sub>16</sub>" in the I<sup>2</sup>Ci clock control register.
- ③ Set "10<sub>16</sub>" in the I<sup>2</sup>Ci status register and hold the SCL at the HIGH.
- ④ Set a communication enable status by setting "08<sub>16</sub>" in the I<sup>2</sup>Ci control register.
- ⑤ Set the address data of the destination of transmission in the high-order 7 bits of the I<sup>2</sup>Ci data shift register and set "0" in the least significant bit.
- ⑥ Set "F0<sub>16</sub>" in the I<sup>2</sup>Ci status register to generate a START condition. At this time, an SCL for 1 byte and an ACK clock automatically occurs.
- ⑦ Set transmit data in the I<sup>2</sup>Ci data shift register. At this time, an SCL and an ACK clock automatically occurs.
- ⑧ When transmitting control data of more than 1 byte, repeat step ⑦.
- ⑨ Set "D0<sub>16</sub>" in the I<sup>2</sup>Ci status register. After this, if ACK is not returned or transmission ends, a STOP condition will be generated.

**(12) Example of Slave Reception**

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

- ① Set a slave address in the high-order 7 bits of the I<sup>2</sup>Ci address register and "0" in the RBW bit.
- ② Set the no ACK clock mode and SCL = 400 kHz by setting "25<sub>16</sub>" in the I<sup>2</sup>Ci clock control register.
- ③ Set "10<sub>16</sub>" in the I<sup>2</sup>Ci status register and hold the SCL at the HIGH.
- ④ Set a communication enable status by setting "08<sub>16</sub>" in the I<sup>2</sup>Ci control register.
- ⑤ When a START condition is received, an address comparison is made.
- ⑥
  - When all transmitted address are "0" (general call):  
AD0 of the I<sup>2</sup>Ci status register is set to "1" and an interrupt request signal occurs.
  - When the transmitted addresses match the address set in ①:  
ASS of the I<sup>2</sup>Ci status register is set to "1" and an interrupt request signal occurs.
  - In the cases other than the above:  
AD0 and AAS of the I<sup>2</sup>Ci status register are set to "0" and no interrupt request signal occurs.
- ⑦ Set dummy data in the I<sup>2</sup>Ci data shift register.
- ⑧ When receiving control data of more than 1 byte, repeat step ⑦.
- ⑨ When a STOP condition is detected, the communication ends.



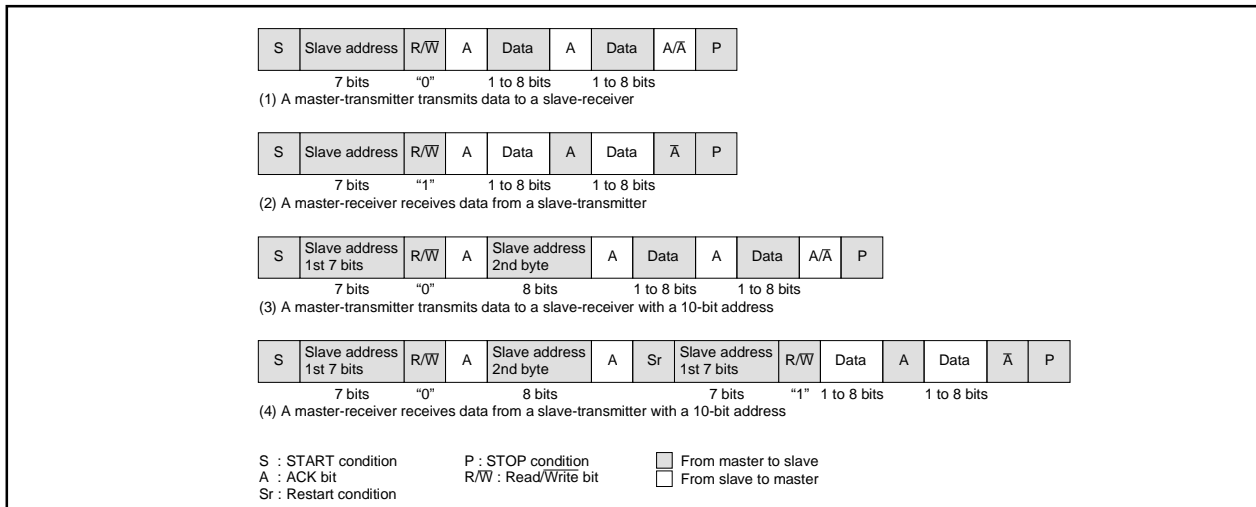


Fig. 2.11.45 Address data communication format

(13) Precautions when using multi-master I<sup>2</sup>C-BUS interface i

■ BCLK operation mode

Select the no-division mode and set the main clock frequency to  $f(XIN) = 10 \text{ MHz}$ .

■ Used instructions

Specify byte (.B) as data size to access multi-master I<sup>2</sup>C-BUS interface i-related registers.

■ Read-modify-write instruction

The precautions when the read-modify-write instruction such as BSET, BCLR etc. is executed for each register of the multi-master I<sup>2</sup>C-BUS interface are described below.

• I<sup>2</sup>Ci data shift register (IICiS0)

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

• I<sup>2</sup>Ci address register (IICiS0D)

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

• I<sup>2</sup>Ci status register (IICiS1)

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

• I<sup>2</sup>Ci control register (IICiS1D)

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

• I<sup>2</sup>Ci clock control register (IICiS2)

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

• I<sup>2</sup>Ci port selection register (IICiS2D)

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

• I<sup>2</sup>Ci transmit buffer register (IICiS0S)

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

**M306V5ME-XXXSP**  
**M306V5EESP**

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

■ **START condition generating procedure using multi-master**

```

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

```

① Be sure to add NOP instruction X 2 between writing the slave address value and setting trigger of START condition generating shown the above procedure example.

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

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

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

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

■ **RESTART condition generating procedure**

```

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

```

① Use the I<sup>2</sup>Ci transmit buffer register to write the slave address value to the I<sup>2</sup>Ci data shift register. And also, be sure to add NOP instruction X 2.

■ **Writing to I<sup>2</sup>Ci status register**

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

■ **Process of after STOP condition generating**

Do not write data in the I<sup>2</sup>Ci data shift register (IICiS0) and the I<sup>2</sup>Ci status register (IICiS1) until the bus busy flag BB becomes “0” after generating the STOP condition in the master mode. It is because the STOP condition waveform might not be normally generated. Reading to the above registers do not have the problem.

## 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 P36, P37, P40–P43 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.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

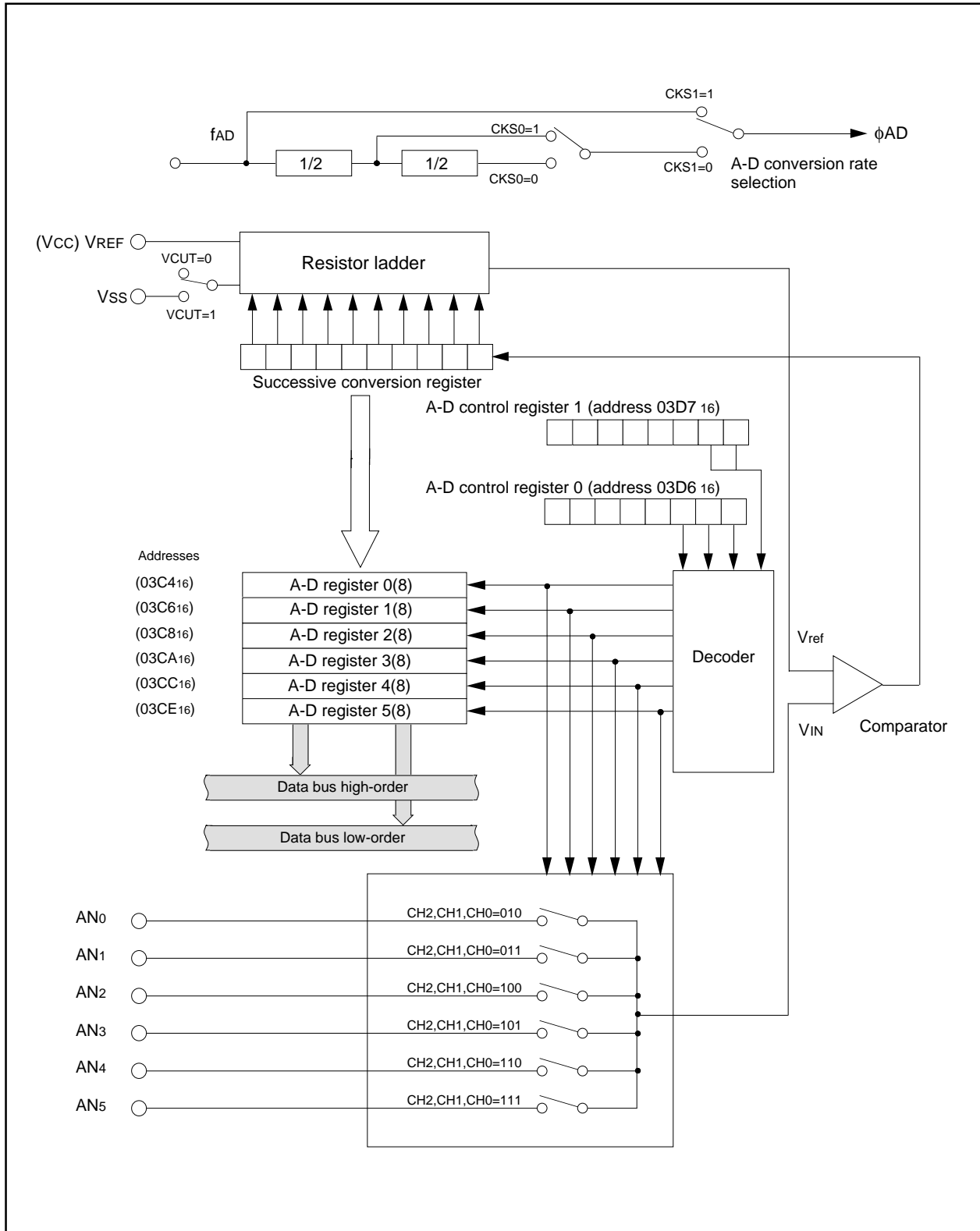


Figure 2.12.1 Block diagram of A-D converter

**M306V5ME-XXXSP**  
**M306V5EESP**

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

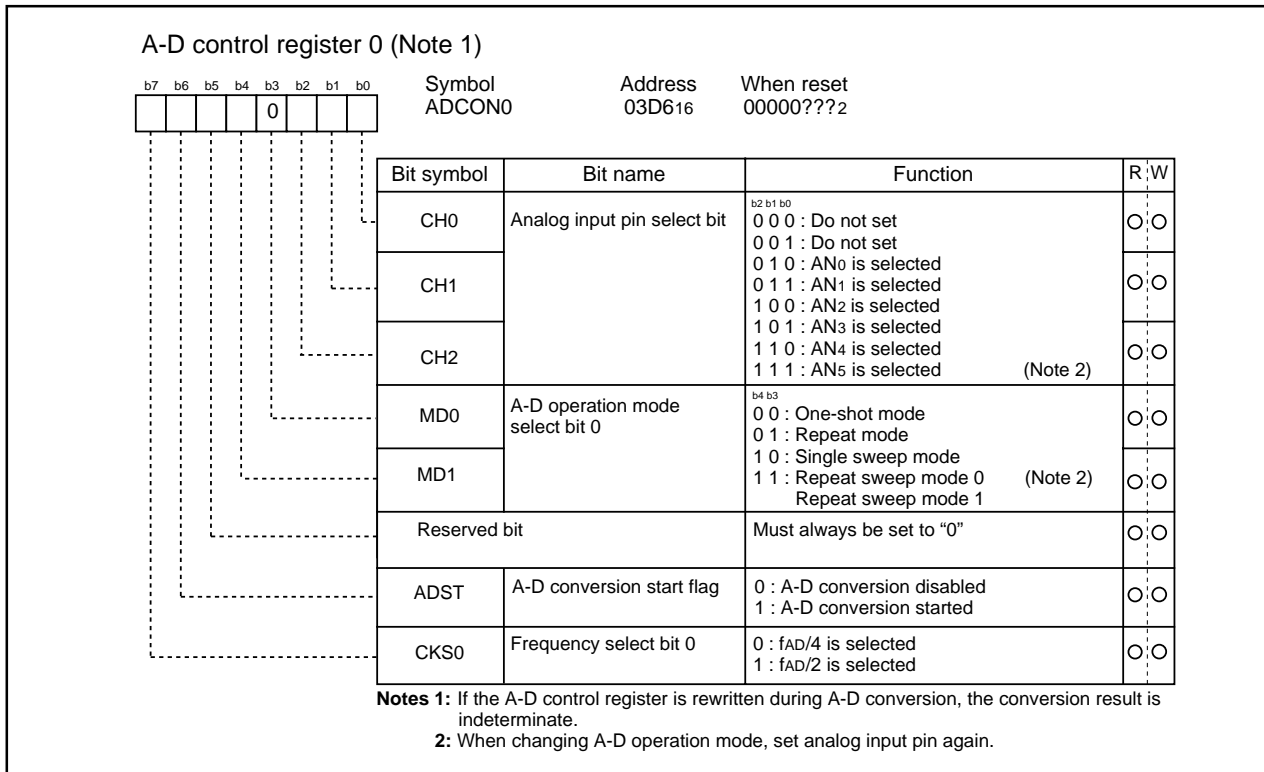


Figure 2.12.2 A-D control register 0

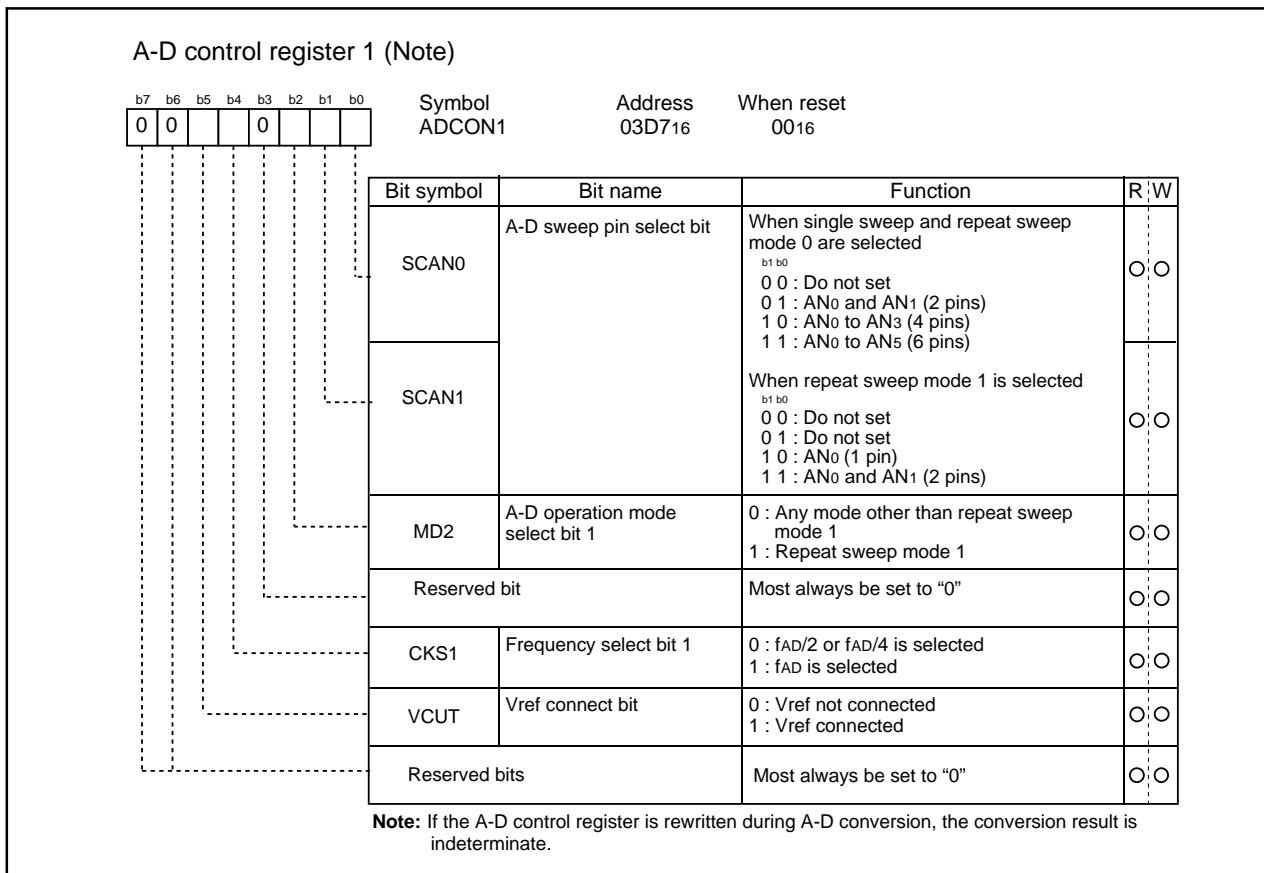


Figure 2.12.3 A-D control register 1

**M306V5ME-XXXSP**  
**M306V5EESP**

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

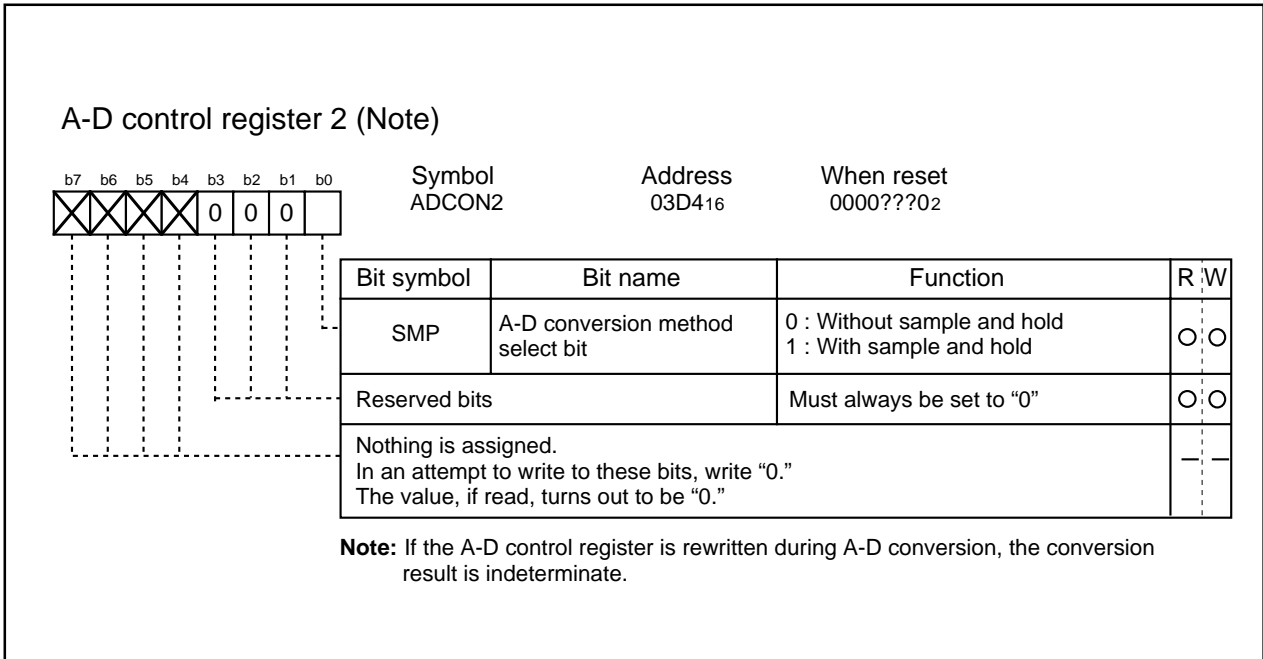


Figure 2.12.4 A-D control register 2

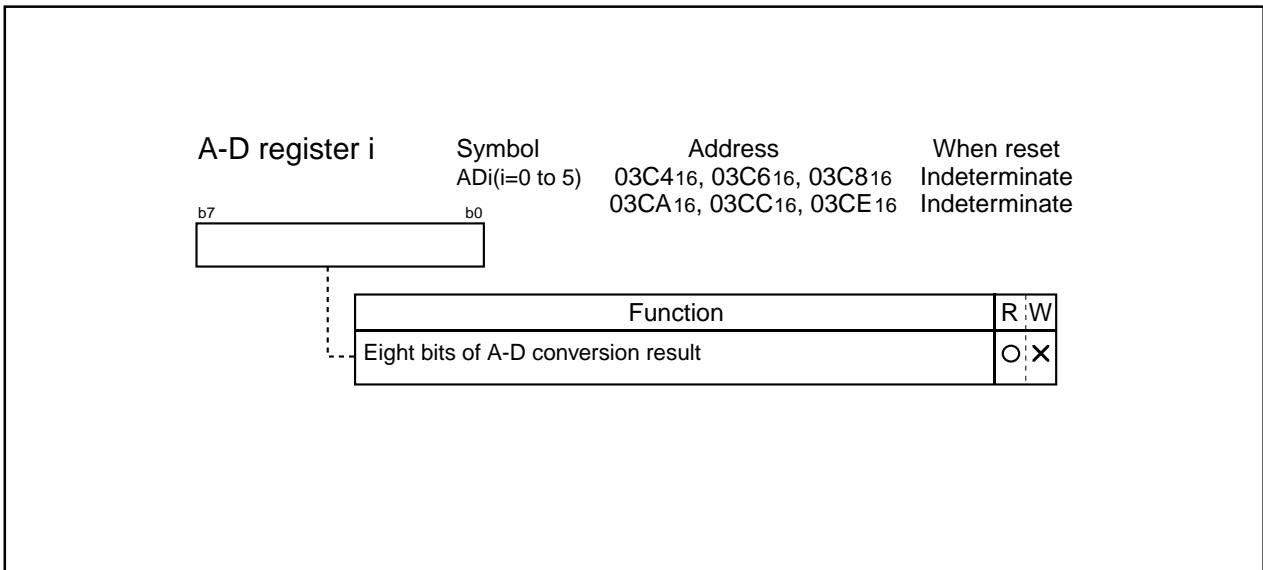


Figure 2.12.5 A-D register i (i = 0 to 5)

### 2.12.1 One-shot Mode

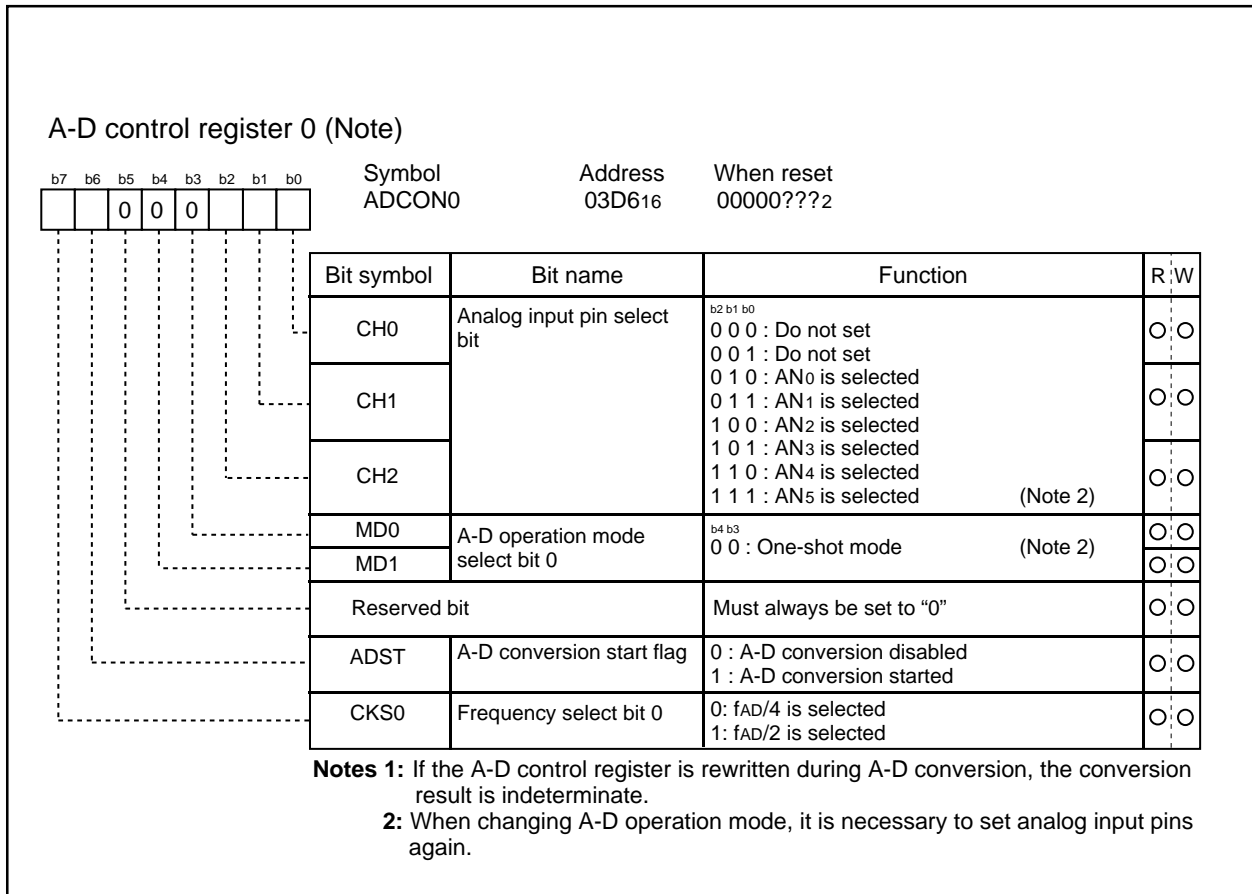
In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion. Table 2.12.2 shows the specifications of one-shot mode. Figures 2.12.6 and 2.12.7 show the A-D control register in one-shot mode.

**Table 2.12.2 One-shot mode specifications**

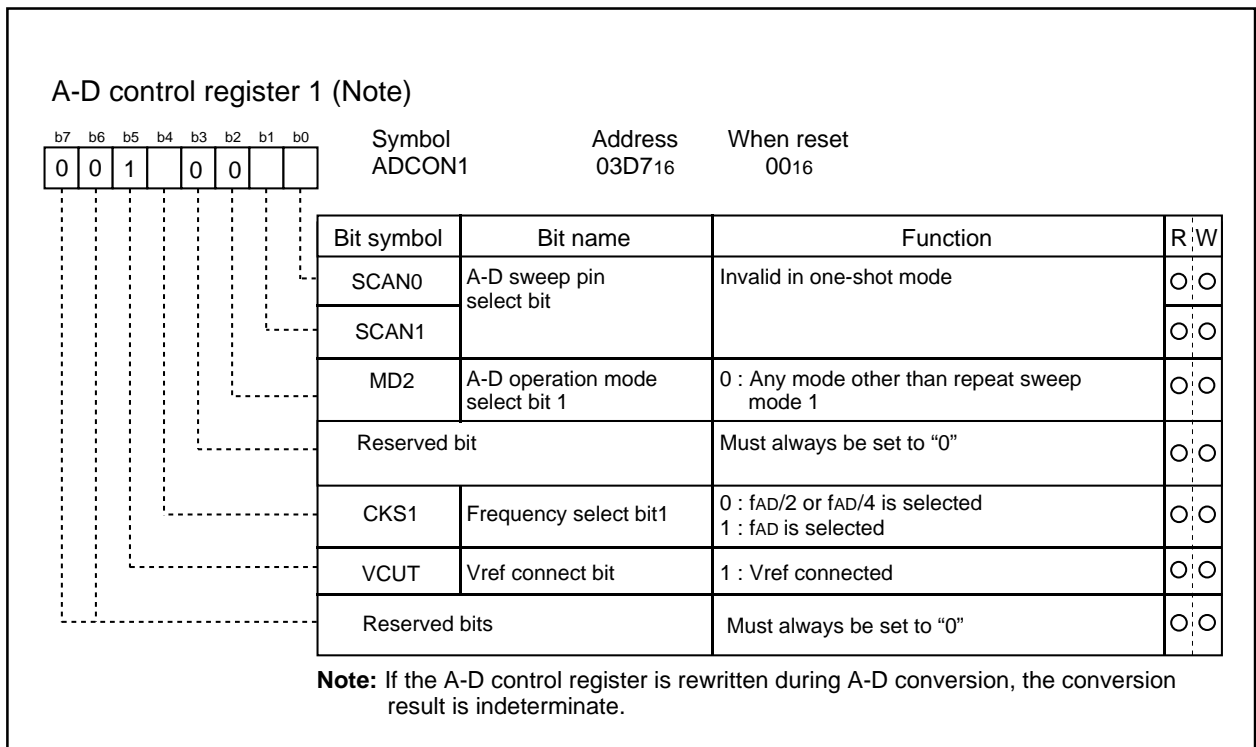
Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	<ul style="list-style-type: none"> <li>• End of A-D conversion</li> <li>• Writing "0" to A-D conversion start flag</li> </ul>
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN0 to AN5, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

**M306V5ME-XXXSP**  
**M306V5EESP**

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.

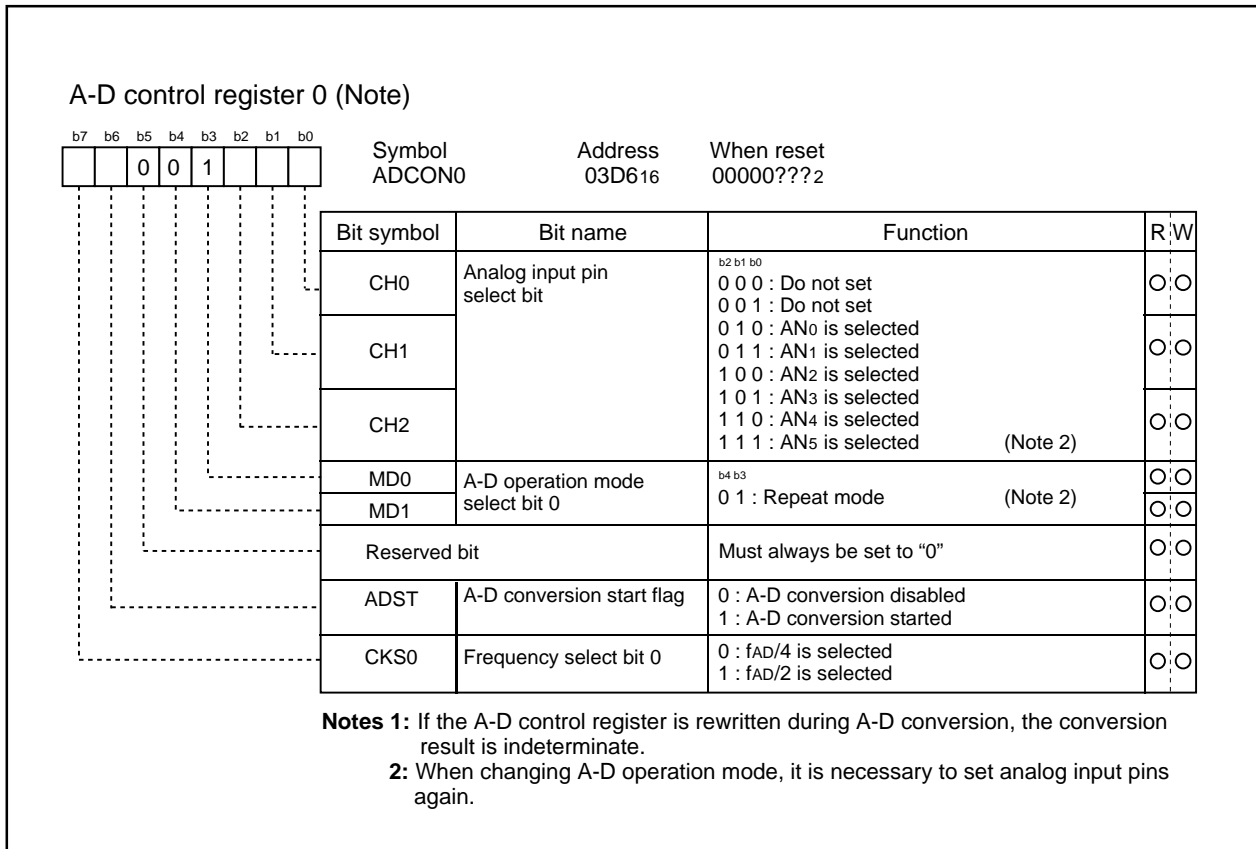
Table 2.12.3 shows the specifications of repeat mode. Figures 2.12.8 and 2.12.9 show the A-D control register in repeat mode.

**Table 2.12.3 Repeat mode specifications**

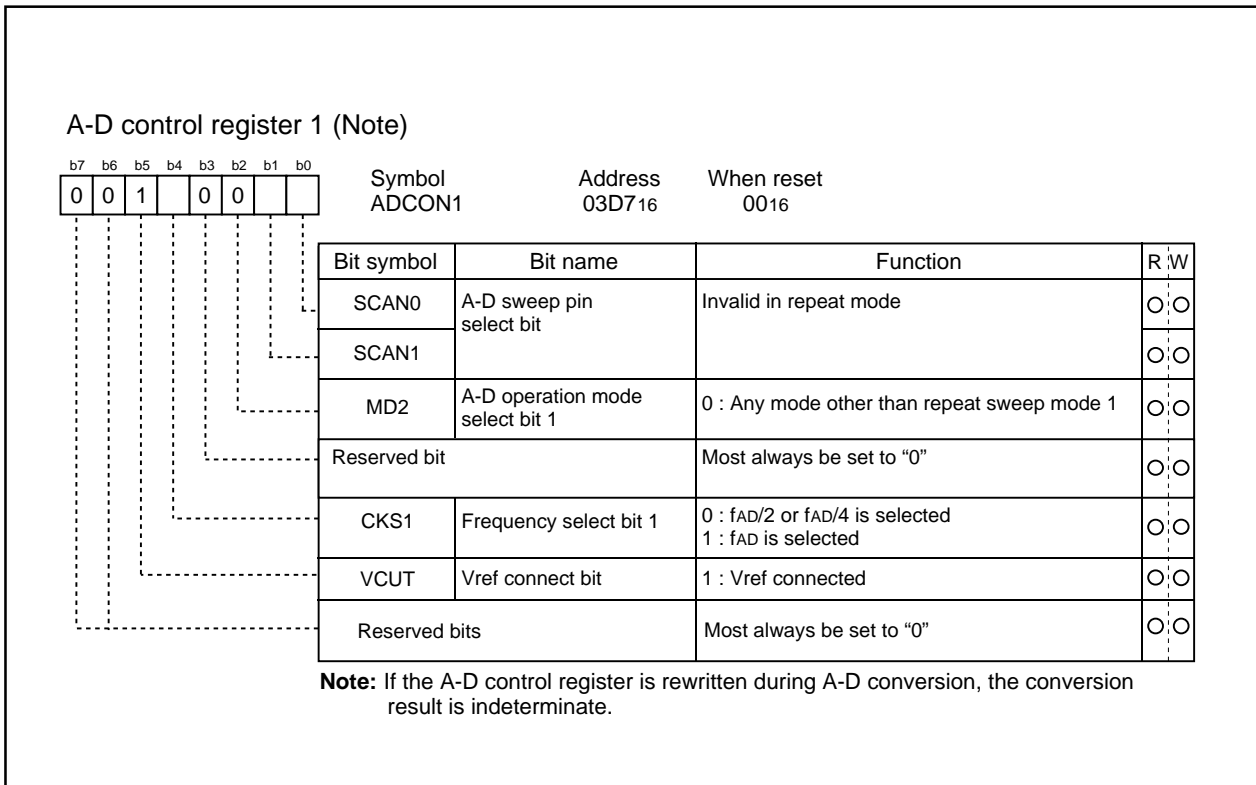
Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D conversion
Star condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN0 to AN5, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**M306V5ME-XXXSP**  
**M306V5EESP**

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

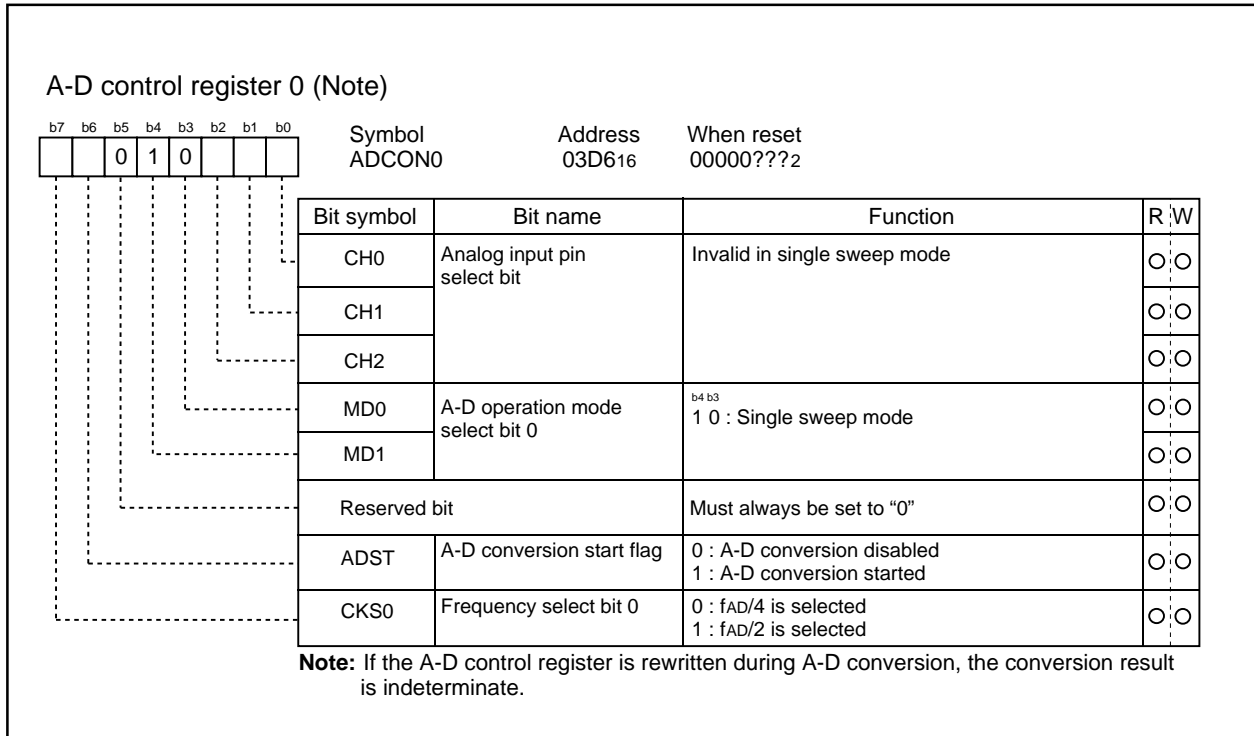


Figure 2.12.10 A-D control register 0 in single sweep mode

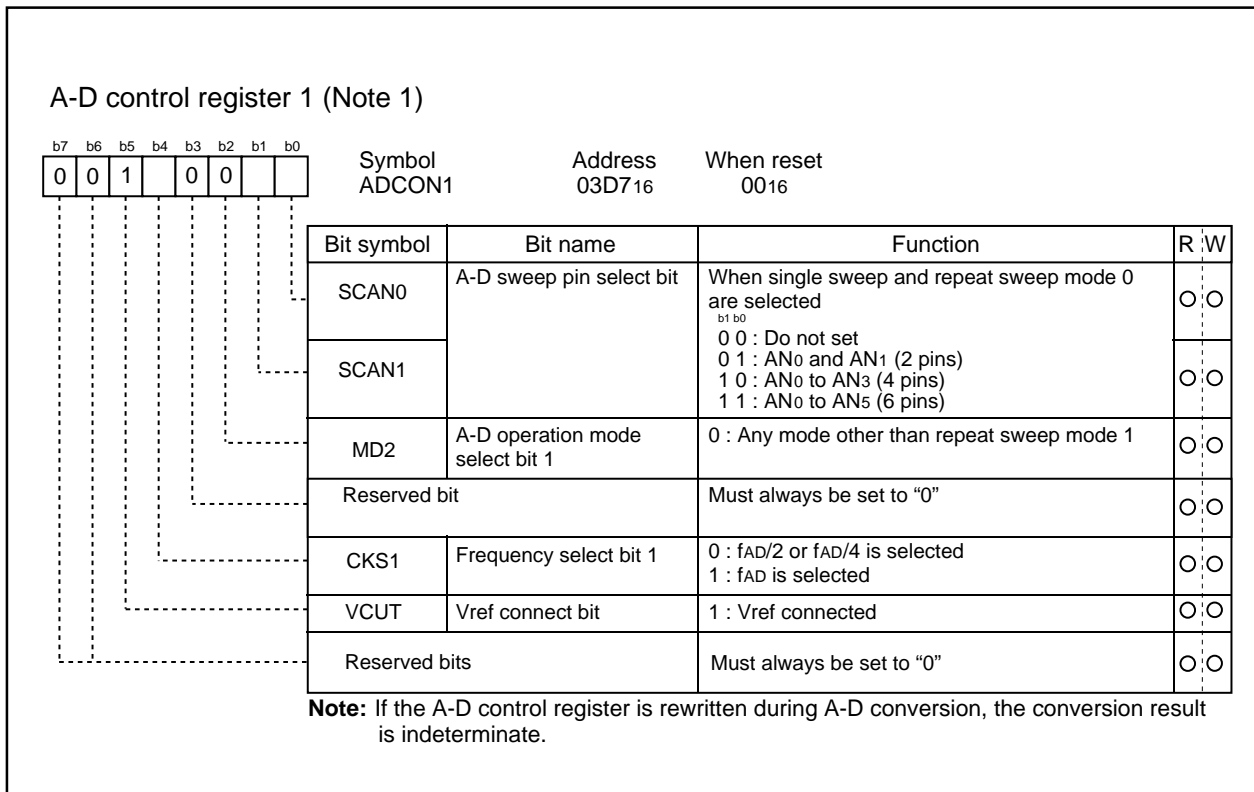


Figure 2.12.11 A-D control register 1 in single sweep mode

### 2.12.4 Repeat Sweep Mode 0

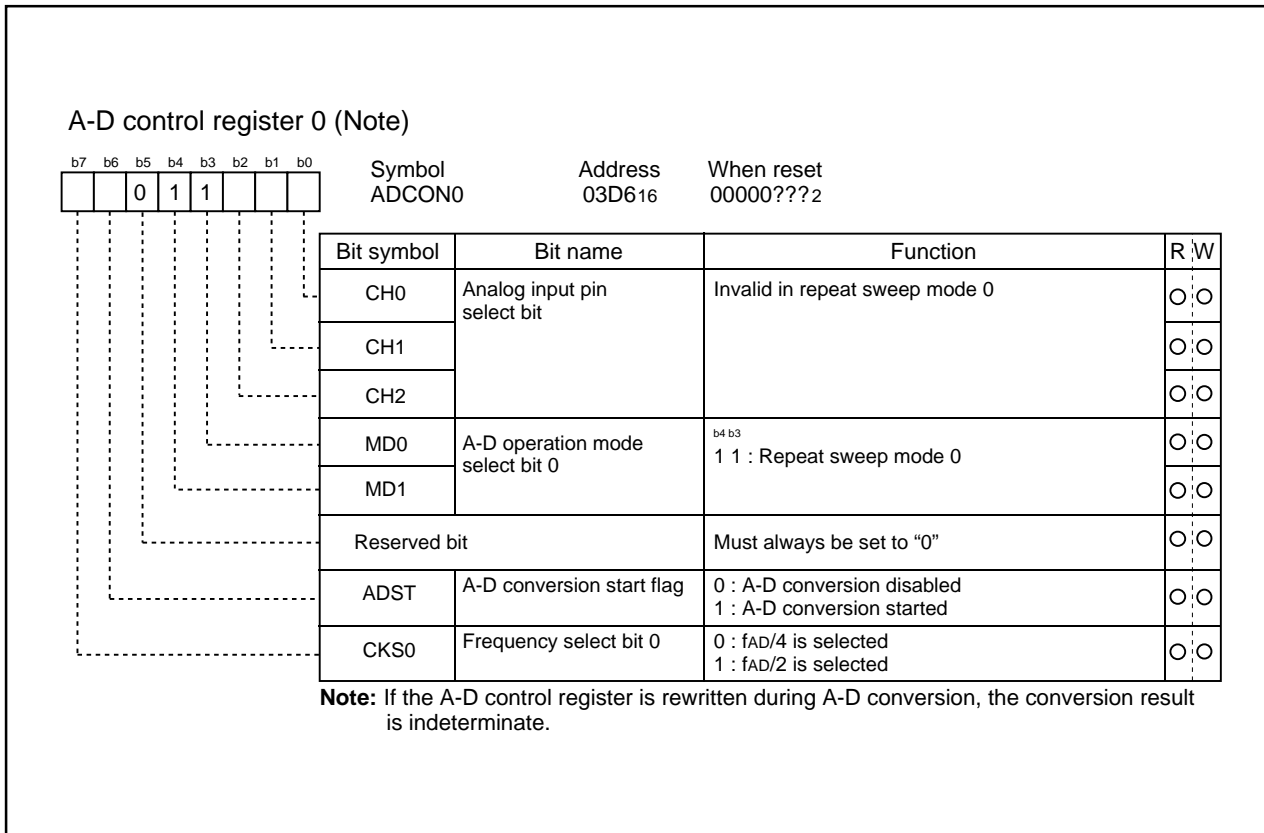
In repeat sweep mode 0, the pins selected using the A-D sweep pin select bit are used for repeat sweep A-D conversion. Table 2.12.5 shows the specifications of repeat sweep mode 0. Figures 2.12.12 and 2.12.13 show the A-D control register in repeat sweep mode 0.

**Table 2.12.5 Repeat sweep mode 0 specifications**

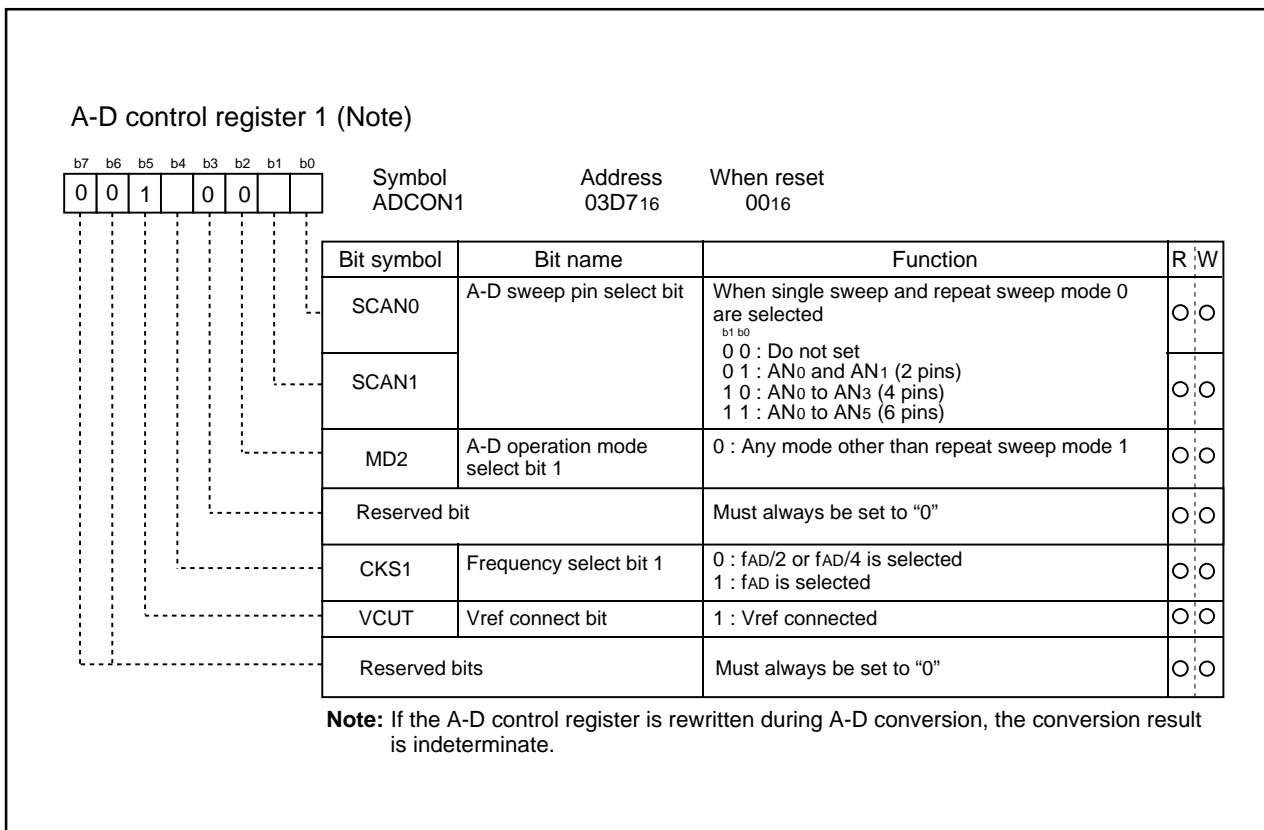
Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for repeat sweep A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

**M306V5ME-XXXSP**  
**M306V5EESP**

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**

**M306V5ME-XXXSP**  
**M306V5EESP**

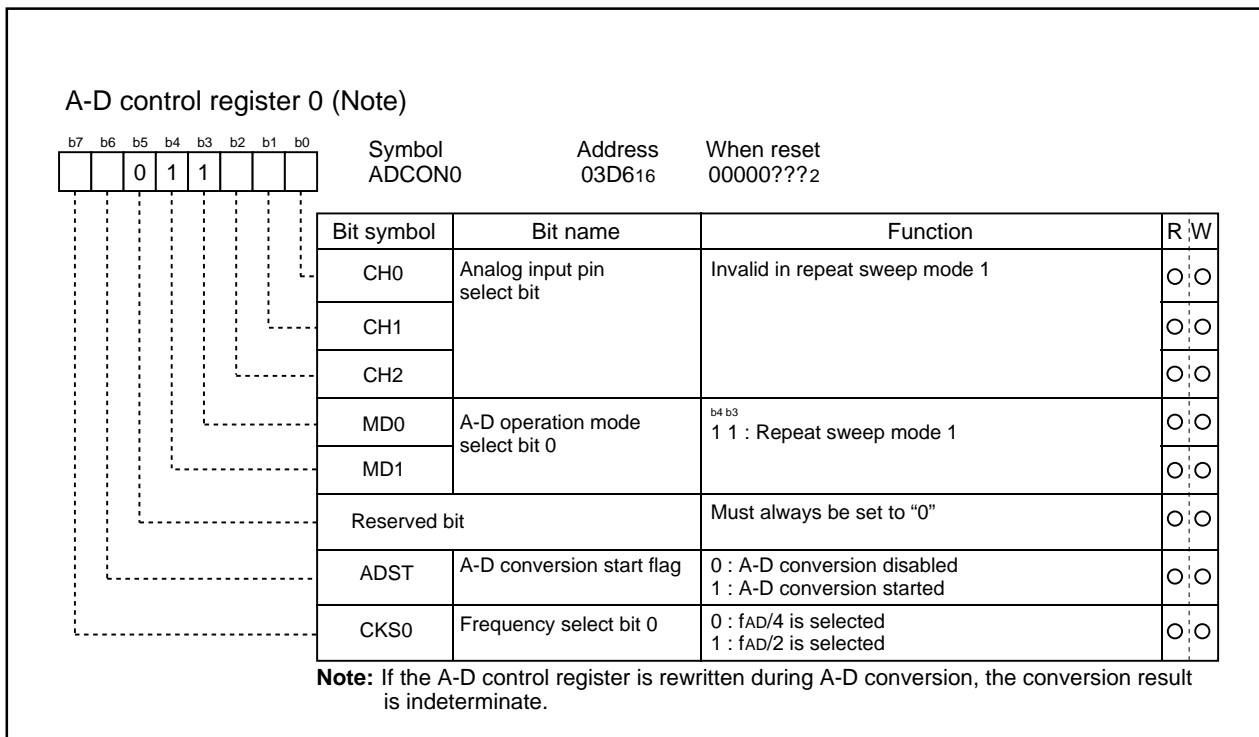
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**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**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

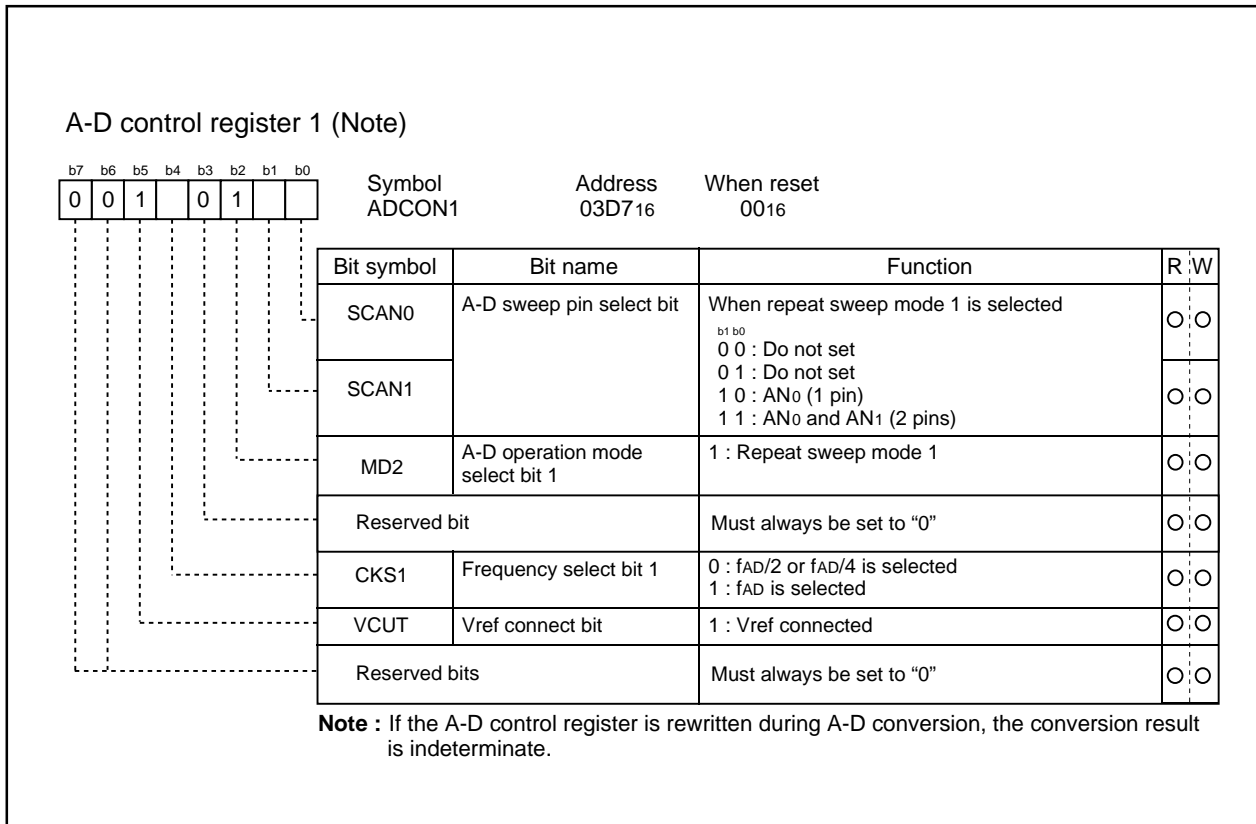


Figure 2.12.15 A-D control register 1 in repeat sweep mode 1



### 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.

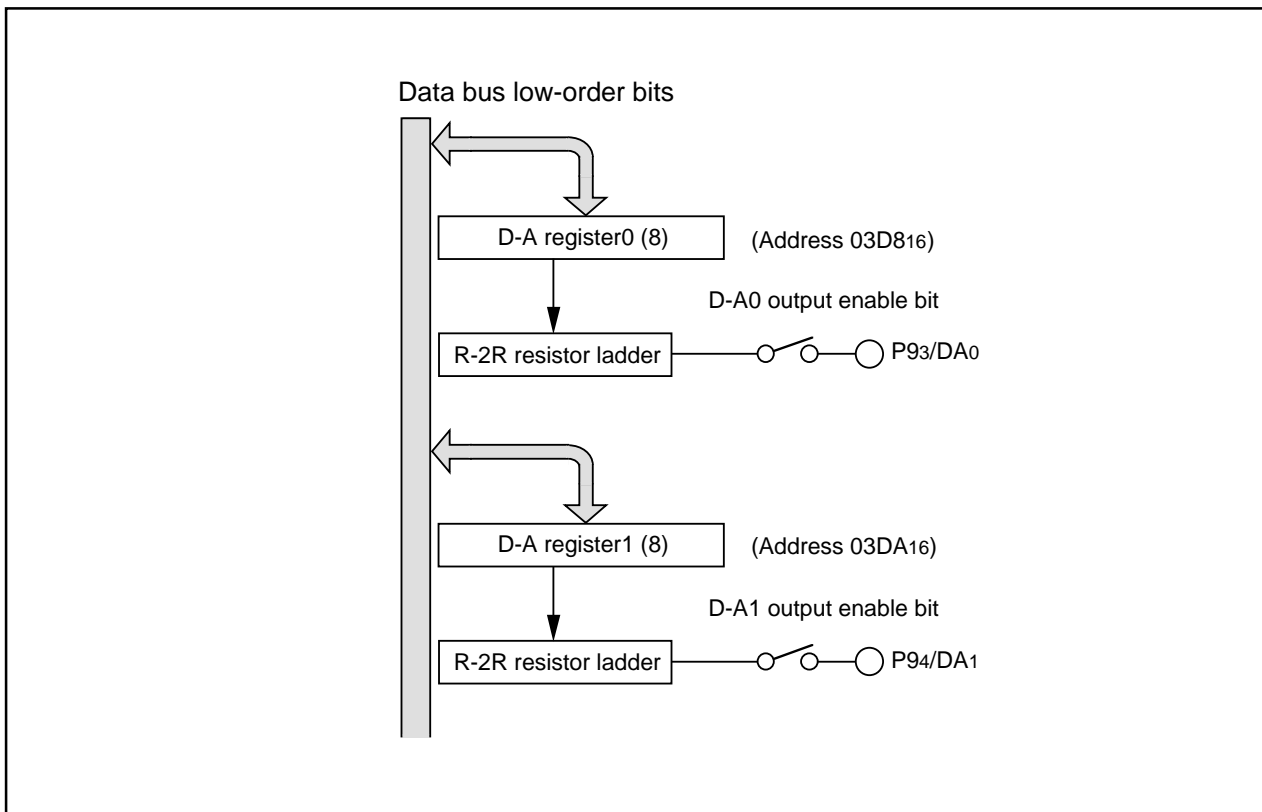
$$V = V_{REF} \times n / 256 \quad (n = 0 \text{ to } 255)$$

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**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

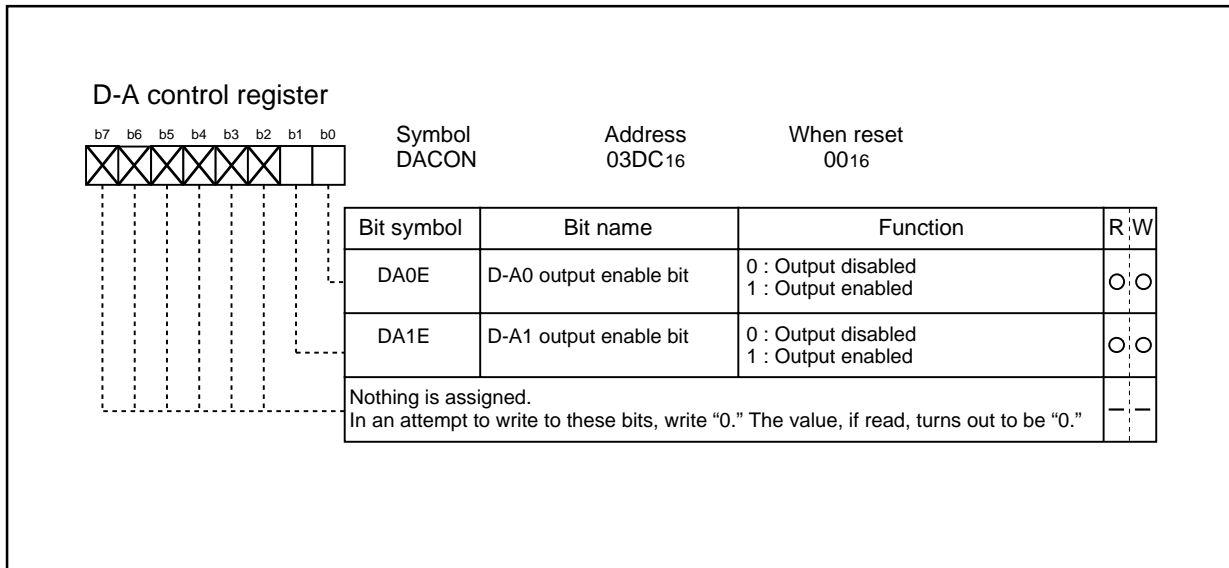


Figure 2.13.2 D-A control register

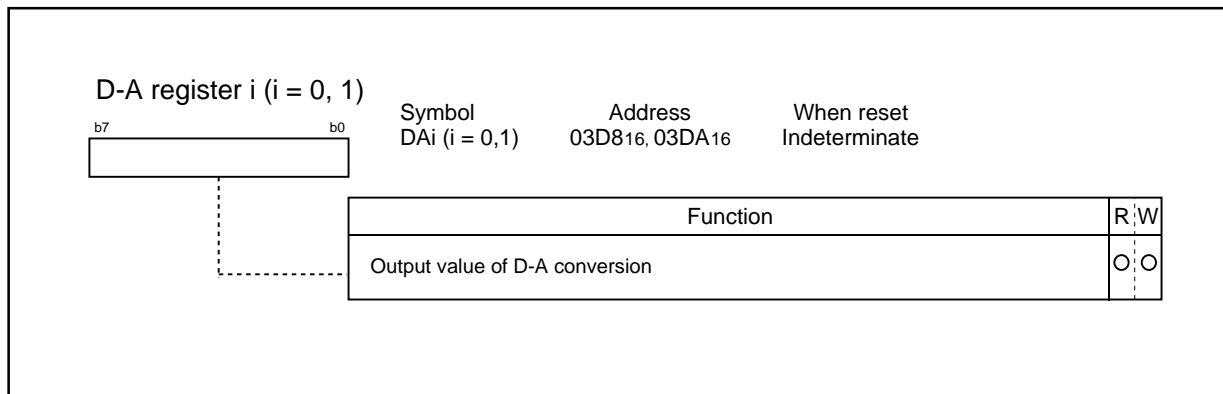


Figure 2.13.3 D-A register i (i = 0 and 1)

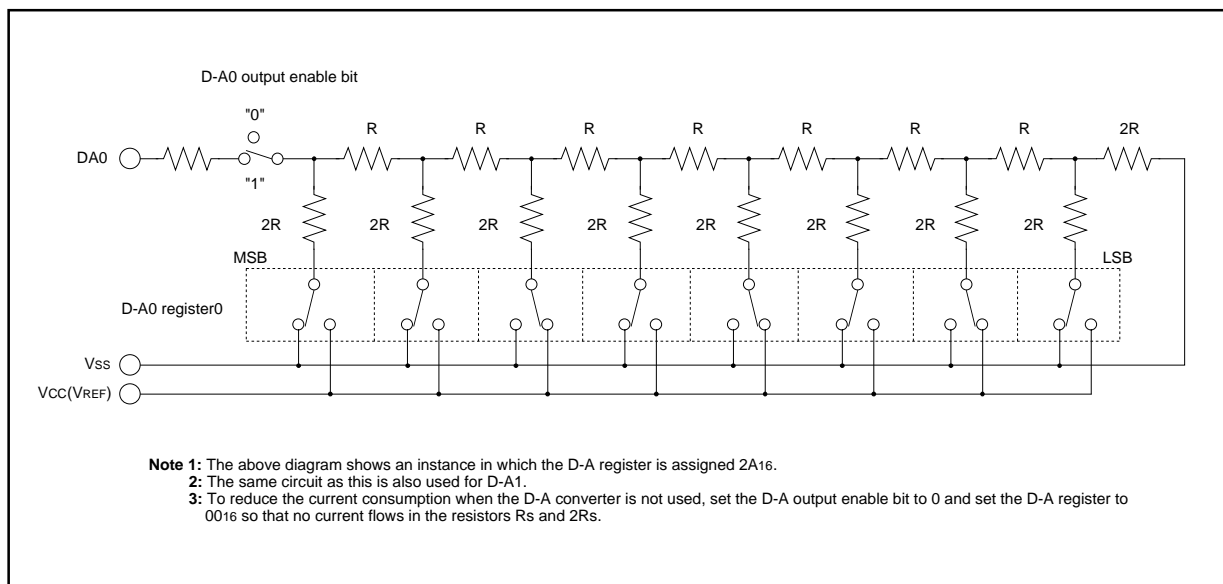


Figure 2.13.4 D-A converter equivalent circuit

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.tip's polarity negative is input to the CVIN pin. When the data slicer function is not used, the data slicer circuit and the timing signal generating circuit can be cut off by setting bit 0 of the data slicer control register 1 (address 0260<sub>16</sub>) 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 027D<sub>16</sub>) according to the main clock frequency.

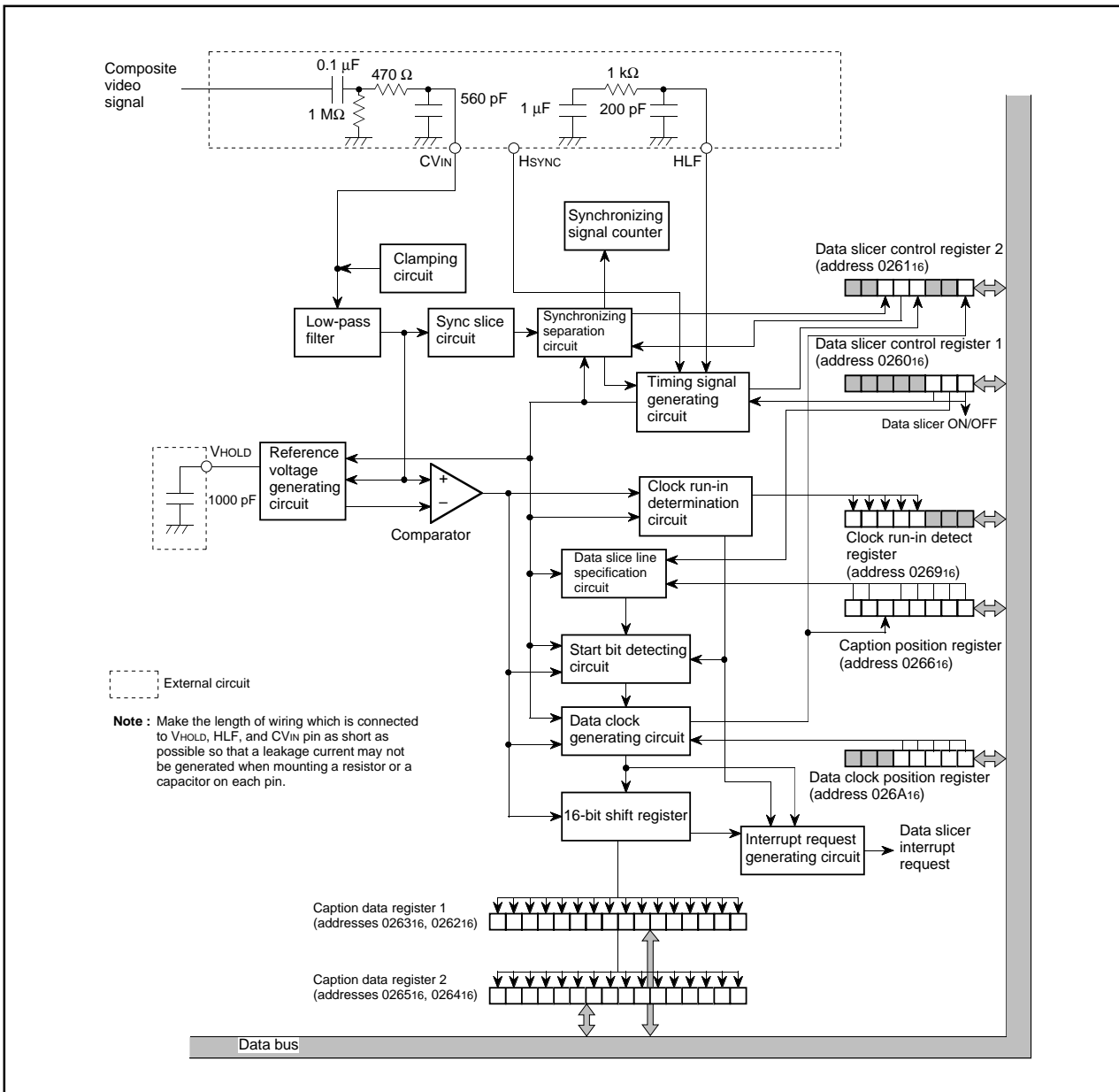


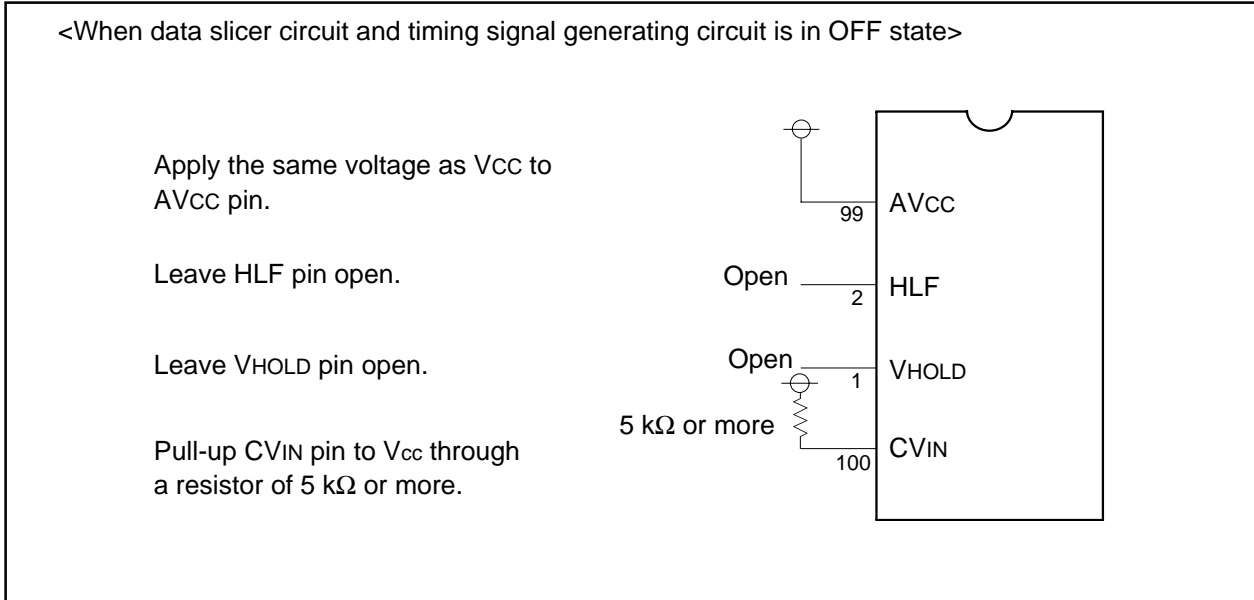
Figure 2.14.1 Data slicer block diagram

**M306V5ME-XXXSP**  
**M306V5EESP**

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

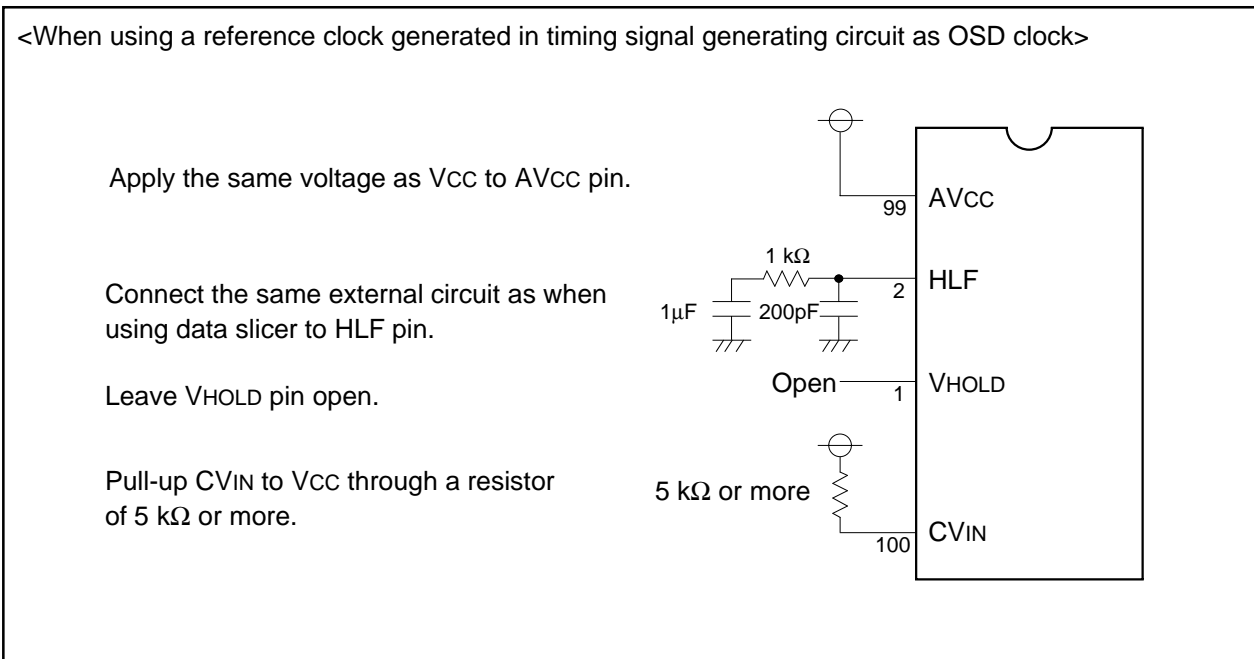
**2.14.1 Notes when not Using Data Slicer**

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**

**M306V5ME-XXXSP**  
**M306V5EESP**

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.

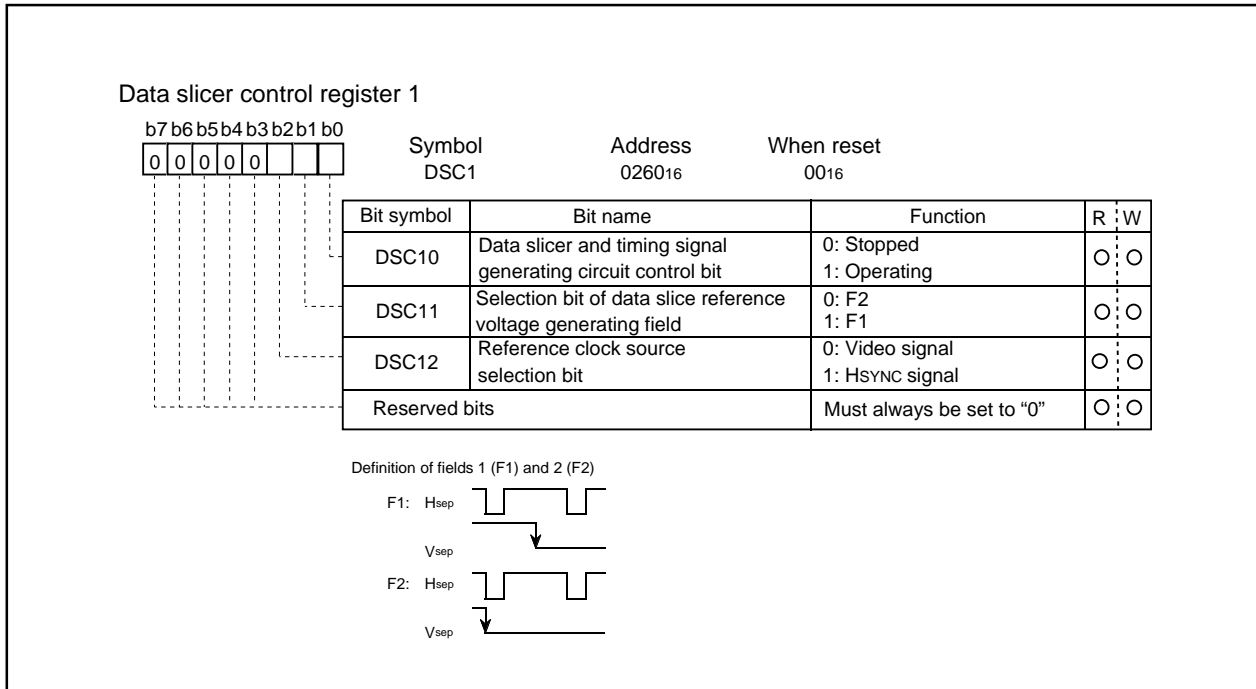


Figure 2.14.4 Data slicer control register 1

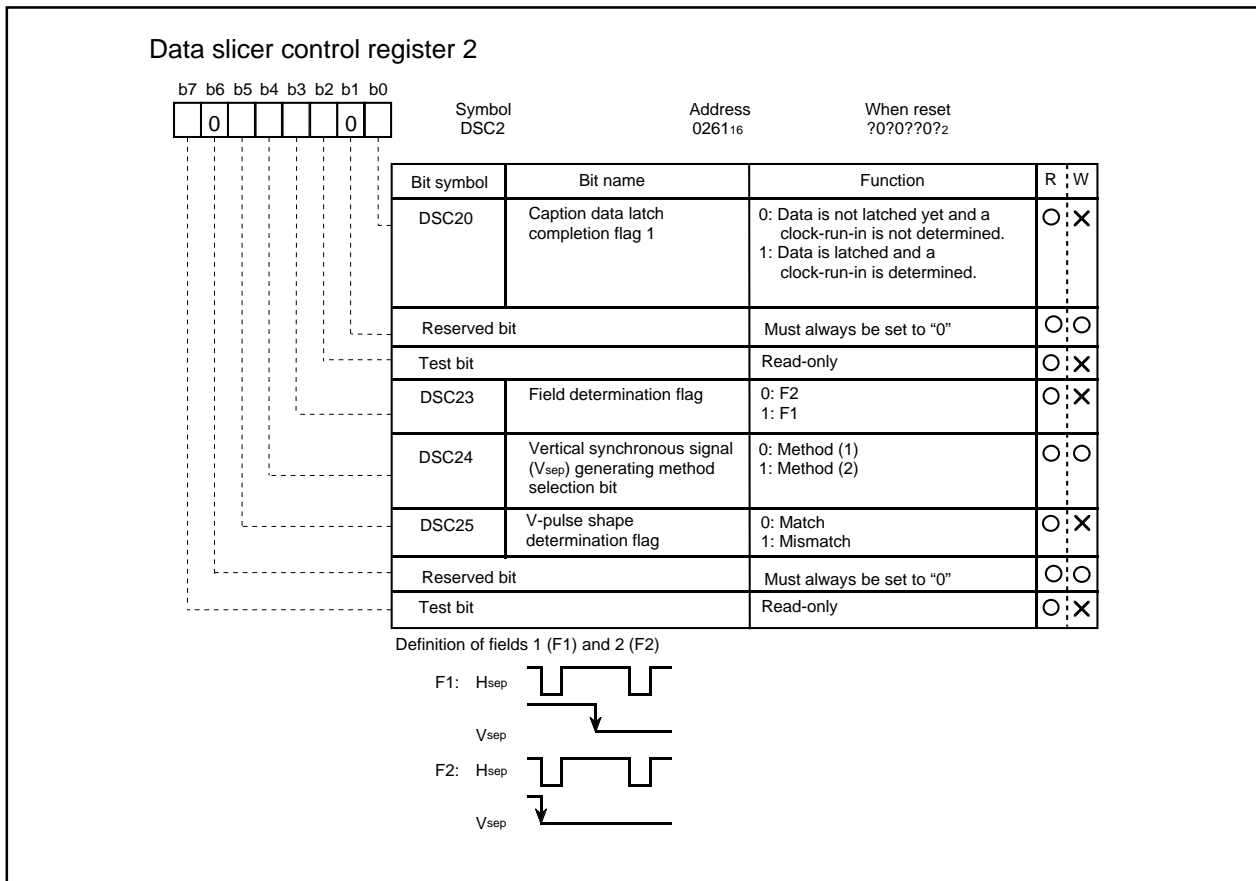


Figure 2.14.5 Data slicer control register 2

### 2.14.2 Clamping Circuit and Low-pass Filter

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

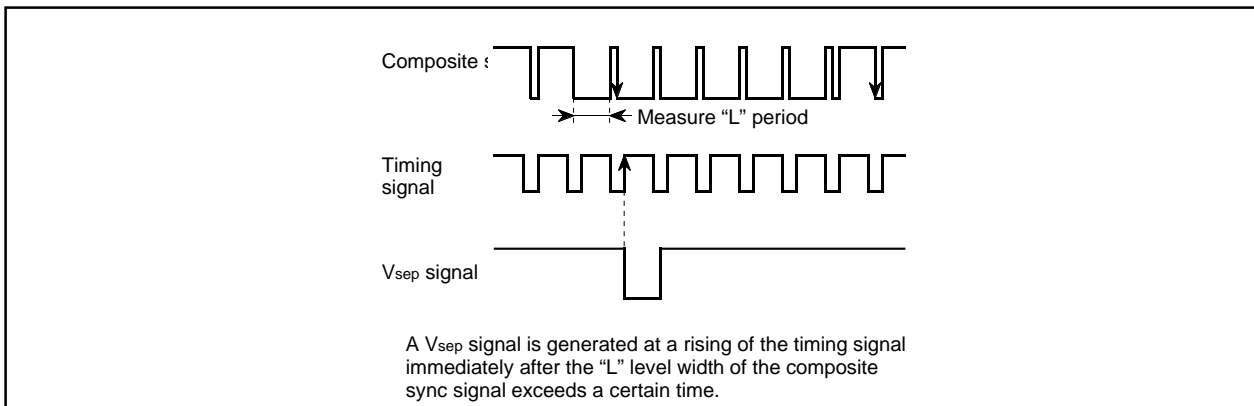
#### (2) Vertical synchronous signal (Vsep)

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)**

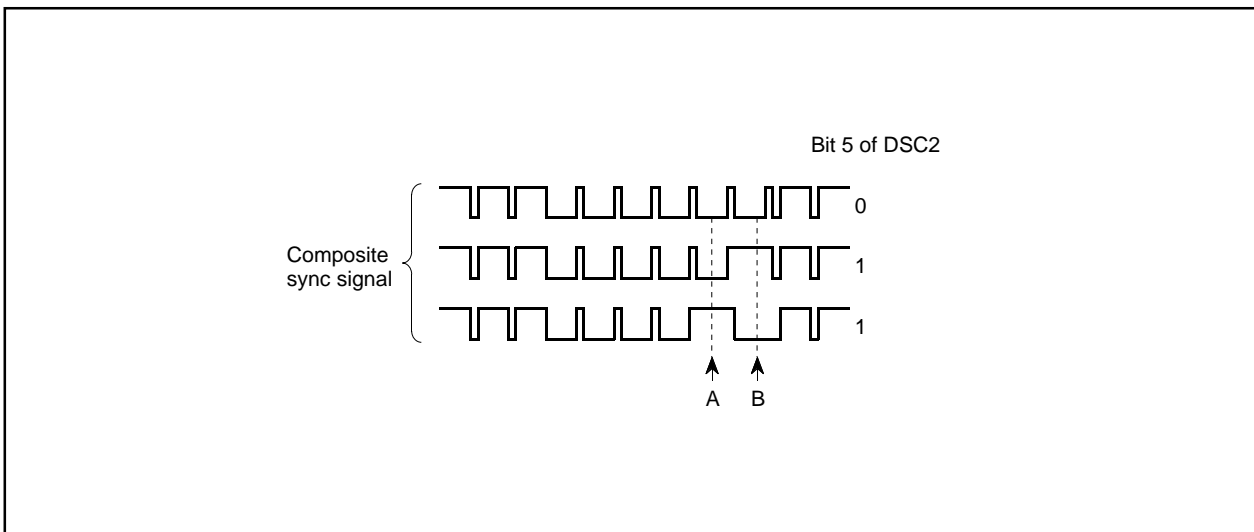
**2.14.5 Timing Signal Generating Circuit**

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 0266<sub>16</sub>) is used for each setting (refer to Table 2.14.1). The counter is reset at the falling edge of V<sub>sep</sub> and is incremented by 1 every H<sub>sep</sub> pulse. When the counter value matched the value specified by bits 4 to 0 of the caption position register, this H<sub>sep</sub> is sliced.

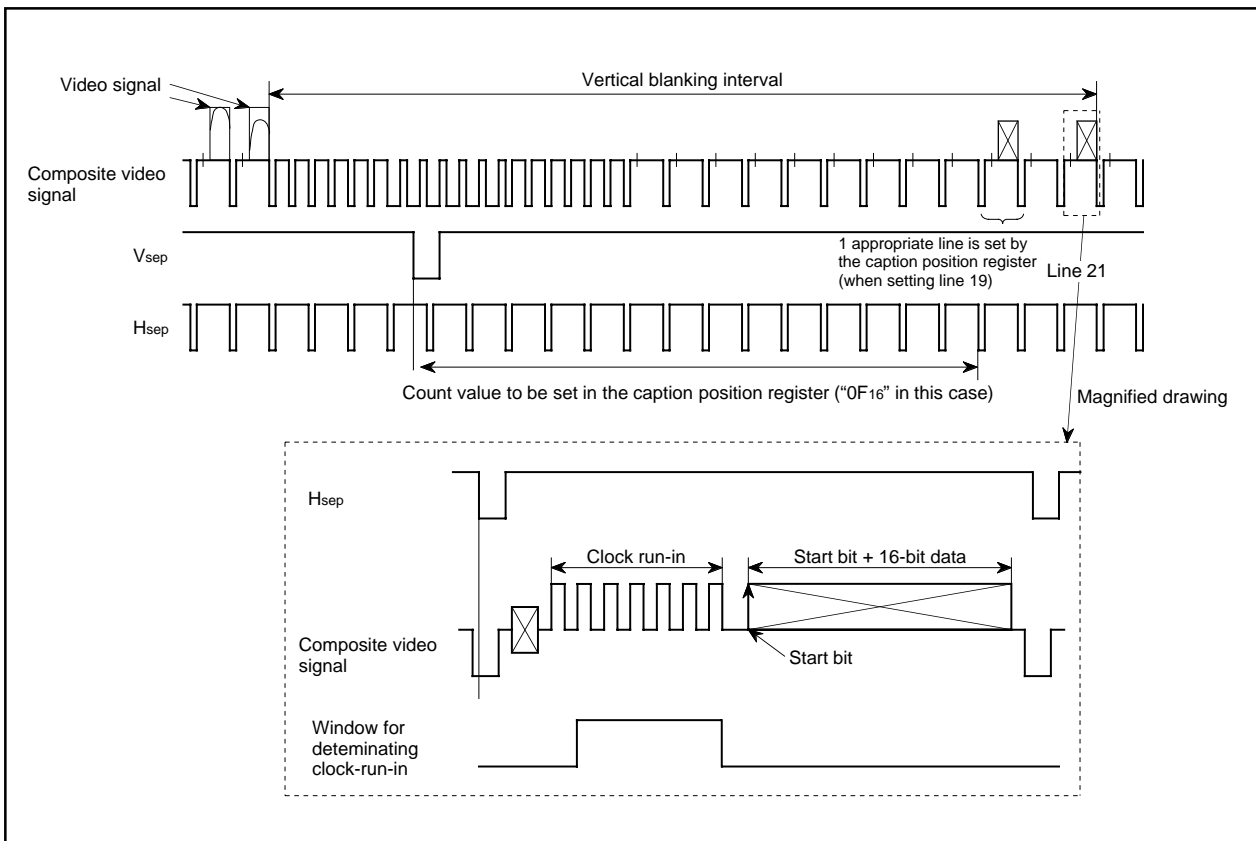
The values of "00<sub>16</sub>" to "1F<sub>16</sub>" can be set in the caption position register (at setting only 1 appropriate line). Figure 2.14.8 shows the signals in the vertical blanking interval. Figure 2.14.9 shows the caption position register.

**(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**

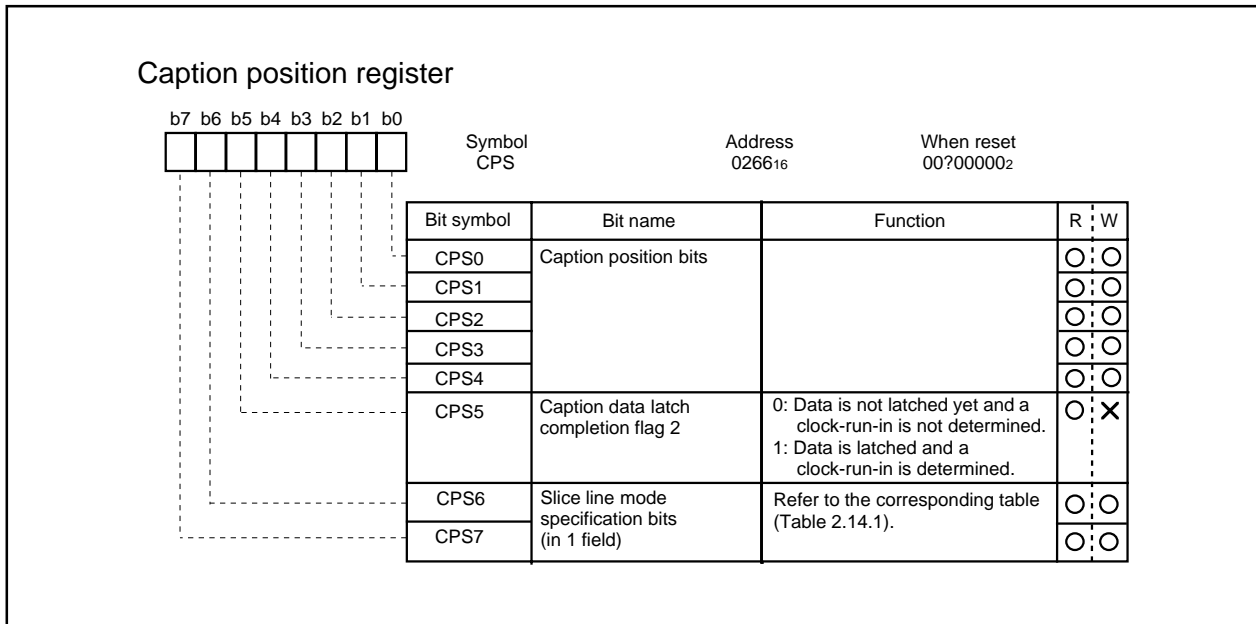
The field determination flag can be read out by bit 3 of data slicer control register 2. This flag change at the falling edge of V<sub>sep</sub>.



**Figure 2.14.8 Signals in vertical blanking interval**

**M306V5ME-XXXSP**  
**M306V5EESP**

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.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**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.

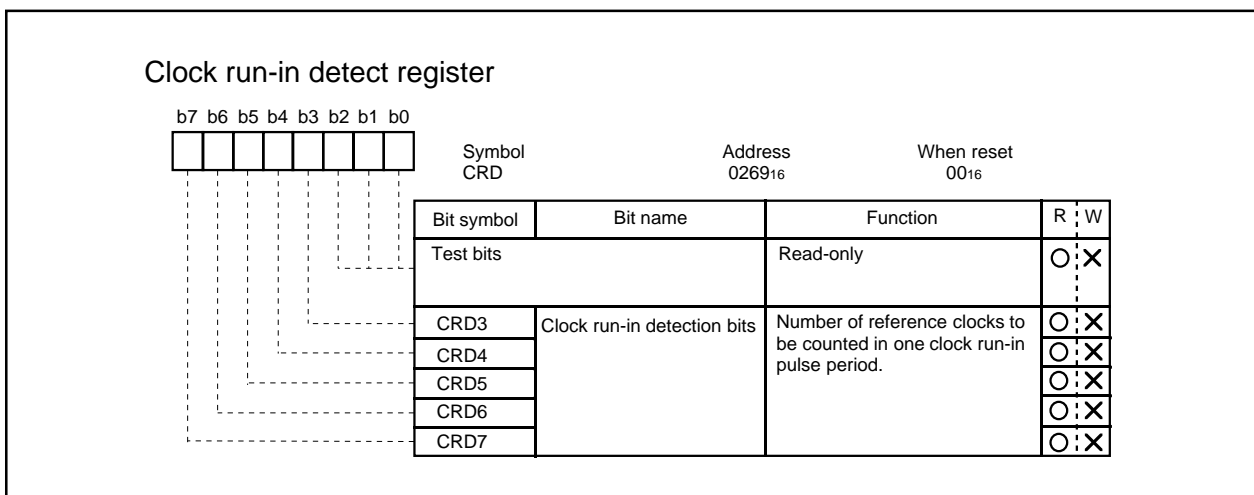
- ① A sampling clock is generated by dividing the reference clock output by the timing signal.
- ② A clock run-in pulse is detected by the sampling clock.
- ③ After detection of the pulse, a start bit pattern is detected from the comparator output.

**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.



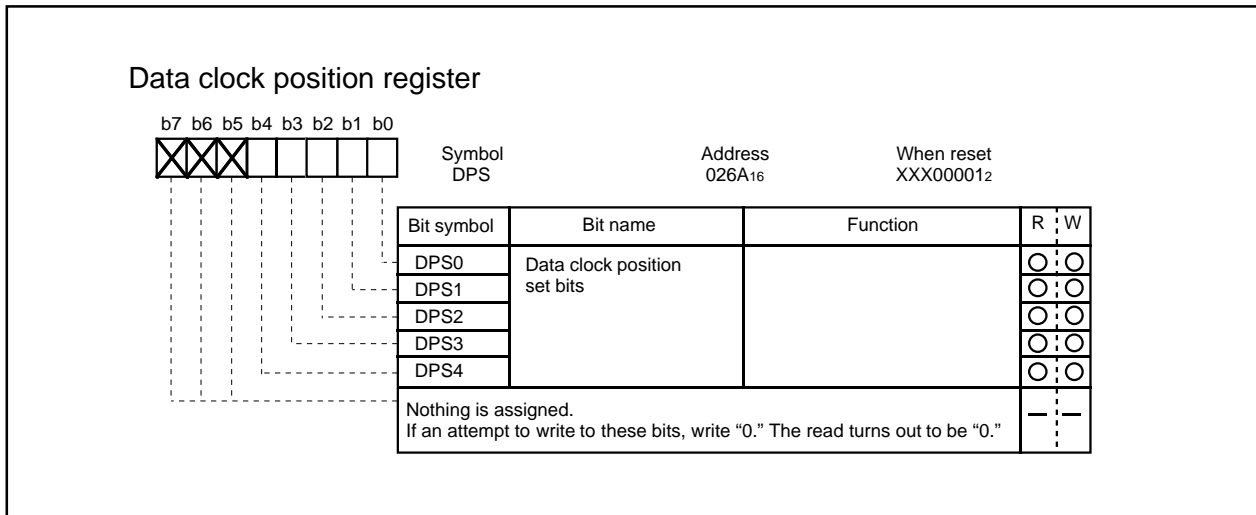
**Figure 2.14.10 Clock run-in detect register**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**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").

**M306V5ME-XXXSP**  
**M306V5EESP**

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**

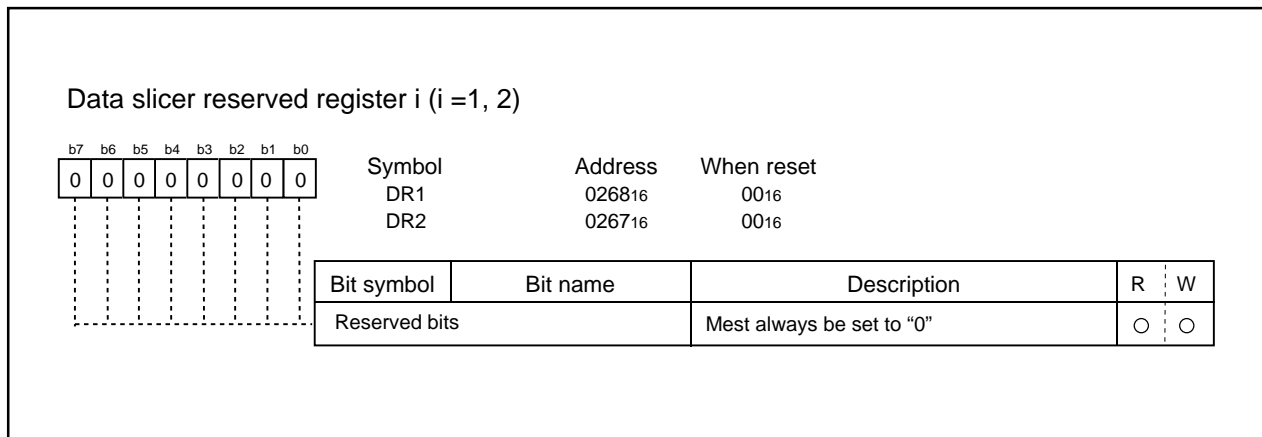
Slice Line Specification Mode		Contents of Caption Data Latch Completion Flag		Contents of 16-bit Shift Register	
CPS		Completion Flag 1 (bit 0 of DSC2)	Completion Flag 2 (bit 5 of CPS)	Caption Data Register 1	Caption Data Register 2
bit 7	bit 6				
0	0	Line 21	A line specified by bits 4 to 0 of CPS	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/P75, HC1/P77) as a count source.

The count value in a certain time (T time; 1024  $\mu$ s, 2048  $\mu$ s, 4096  $\mu$ s and 8192  $\mu$ s) divided system clock f32 is stored into the 8-bit latch.

Accordingly, the latch value changes in the cycle of T time. When the count value exceeds "FF16," "FF16" is stored into the latch.

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

Figure 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 the port direction register corresponding to the HSYNC count input pins for input.

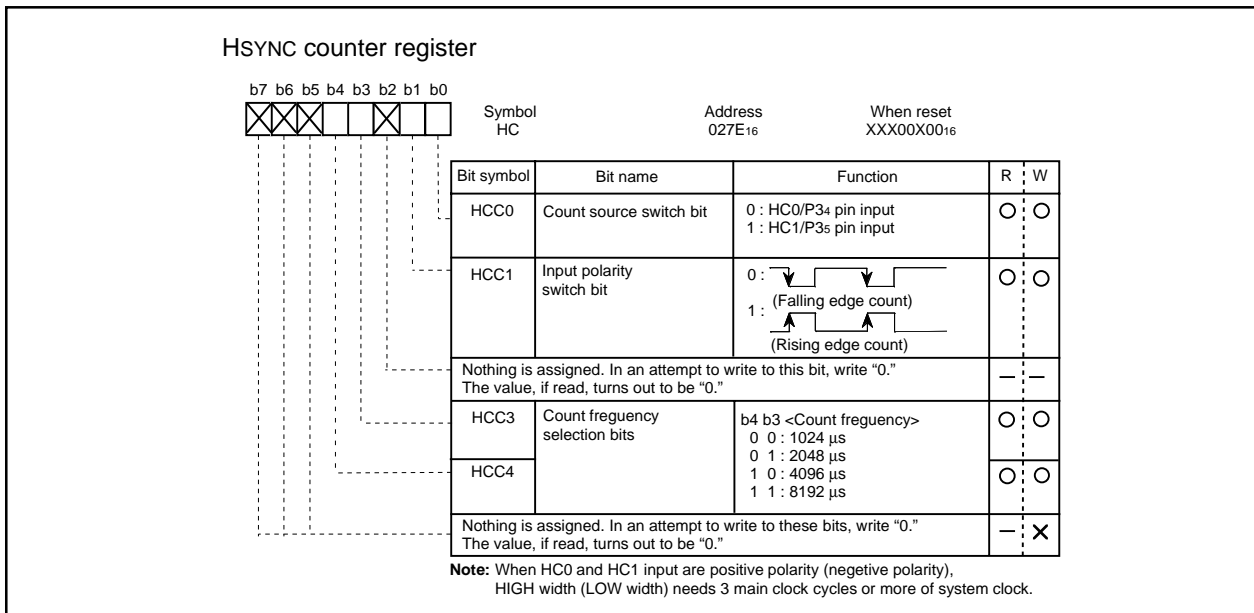


Figure 2.15.1 HSYNC counter register

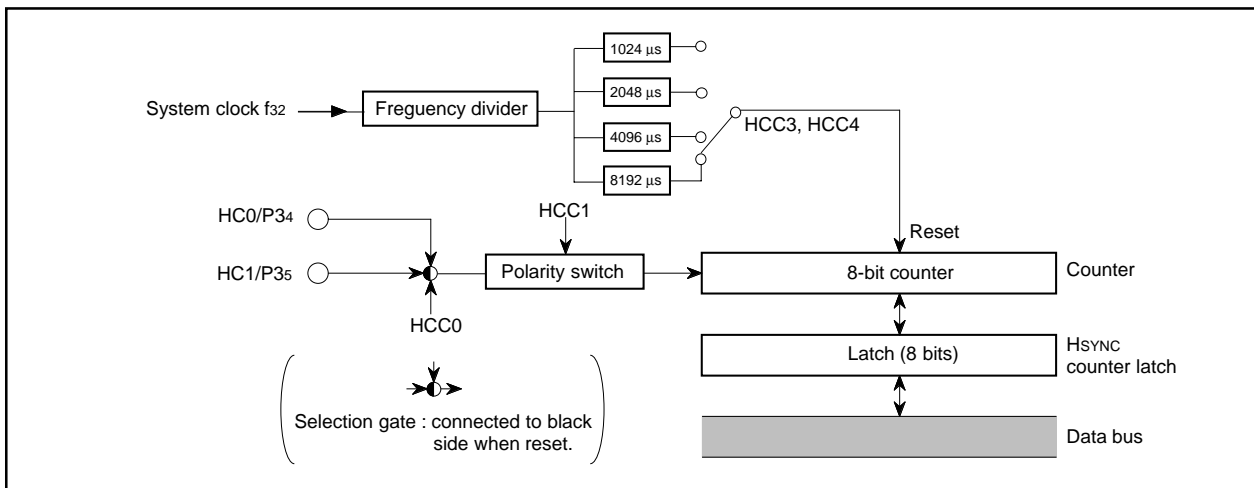


Figure 2.15.2 HSYNC counter block diagram

**M306V5ME-XXXSP**  
**M306V5EESP**

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

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

- Notes** 1: The character size is specified with dot size and pre-divide ratio (refer to “2.16.3 Dot Size”).  
 2: As for SPRITE display, OUT2 is not output.  
 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.

**M306V5ME-XXXSP**  
**M306V5EESP**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 display-enable fonts for each display style. 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.



**M306V5ME-XXXSP**  
**M306V5EESP**

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

Display Styles	Display-enable Fonts
CC Mode	
OSDS Mode	
OSDP Mode	<p>* : Only character codes ** : Blank font</p>
OSDL Mode	
CDOSD Mode	
SPRITE	

Figure 2.16.1 Display-enable fonts for each display style

**M306V5ME-XXXSP**  
**M306V5EESP**

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

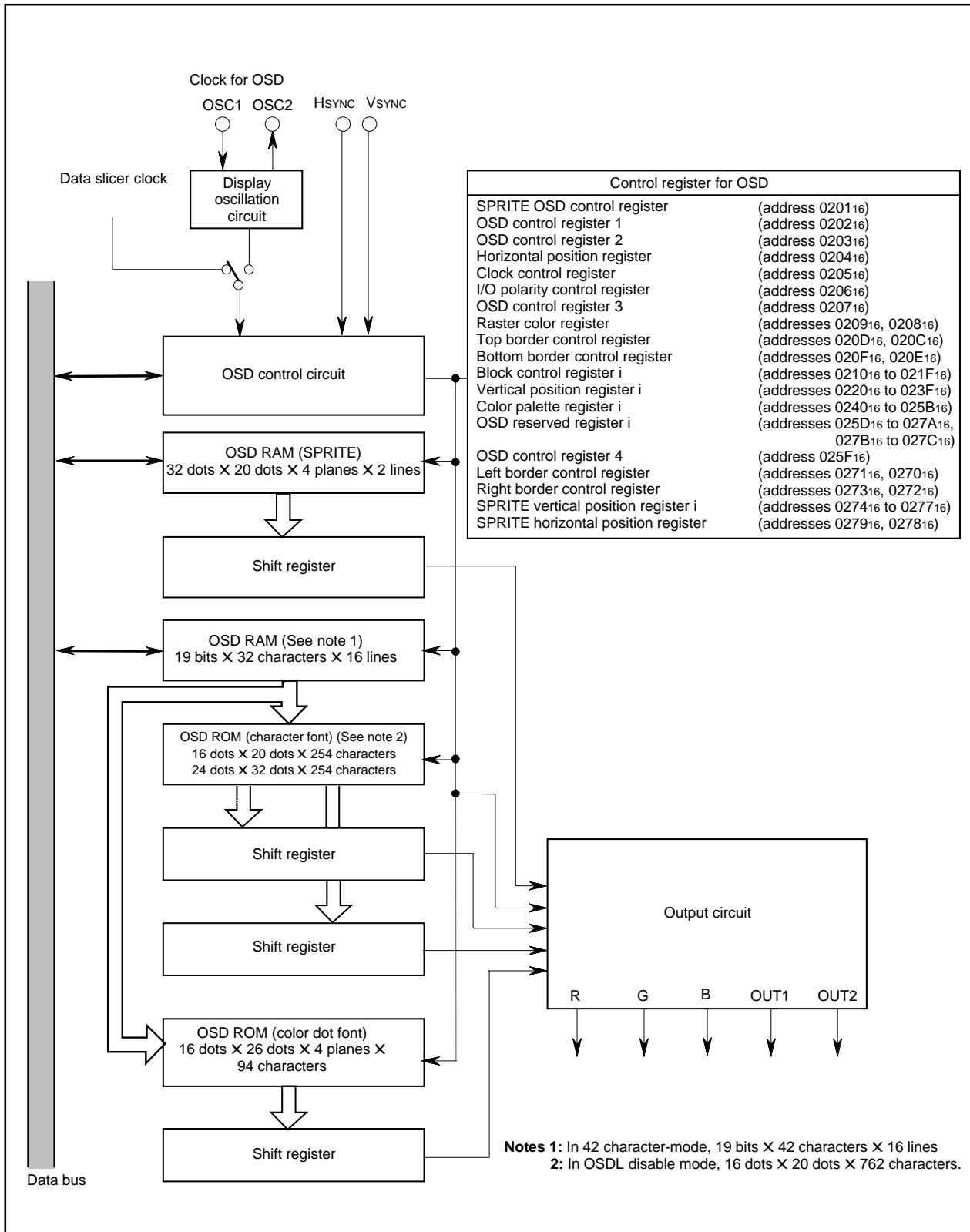


Figure 2.16.2 Block diagram of OSD circuit

**M306V5ME-XXXSP**  
**M306V5EESP**

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

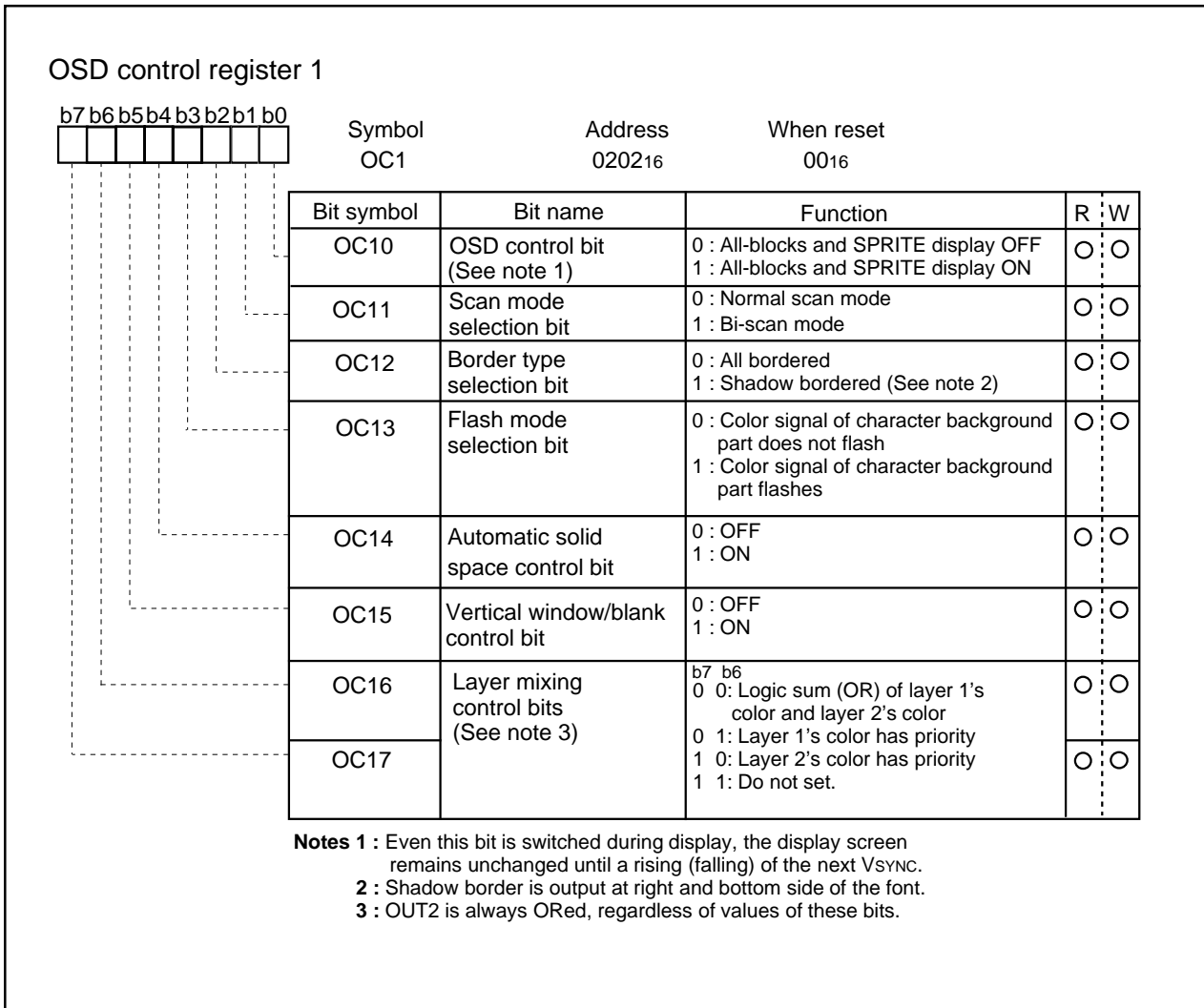


Figure 2.16.3 OSD control register 1

**M306V5ME-XXXSP**  
**M306V5EESP**

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

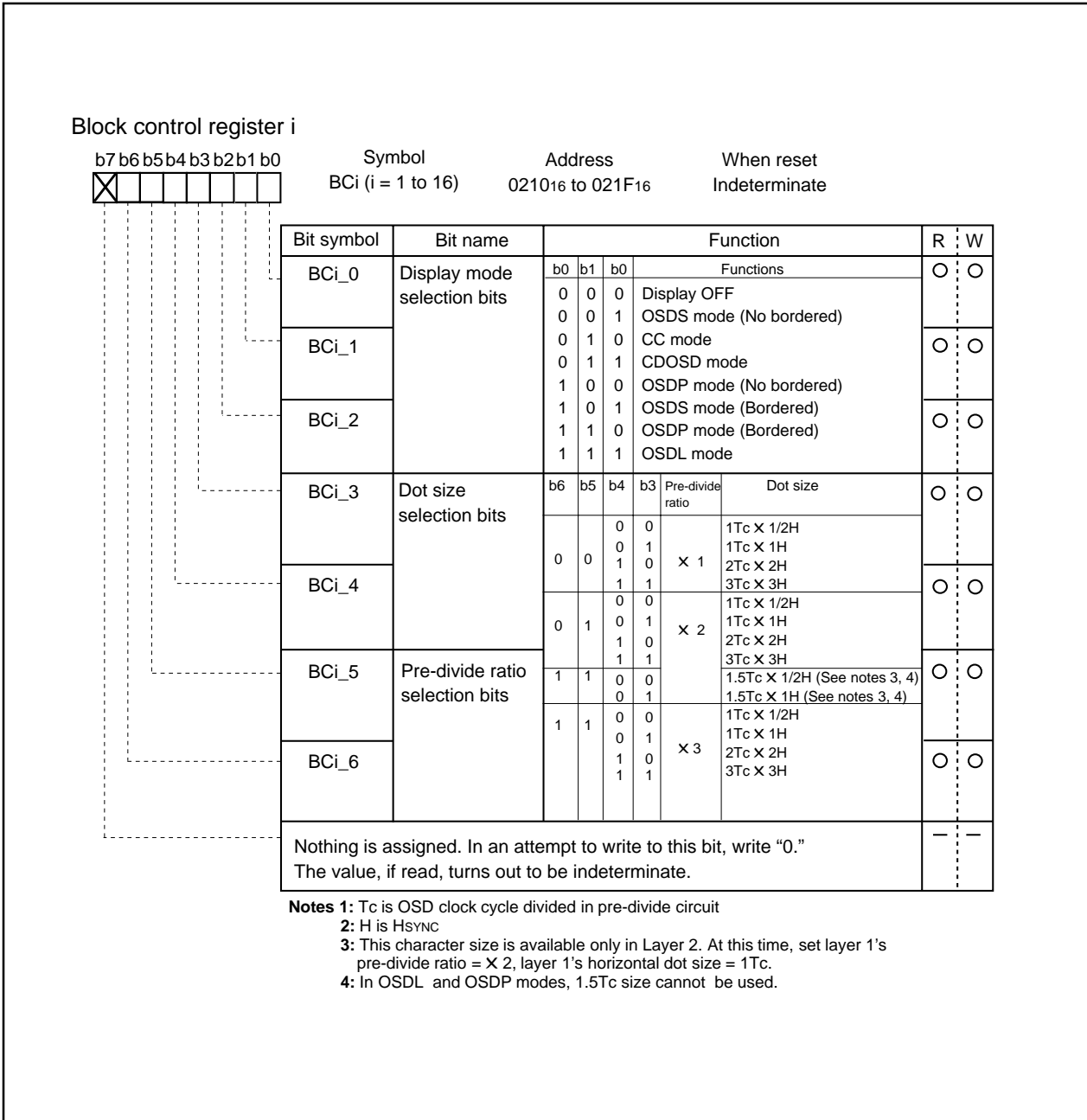


Figure 2.16.4 Block control register i (i = 0 to 16)

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**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.

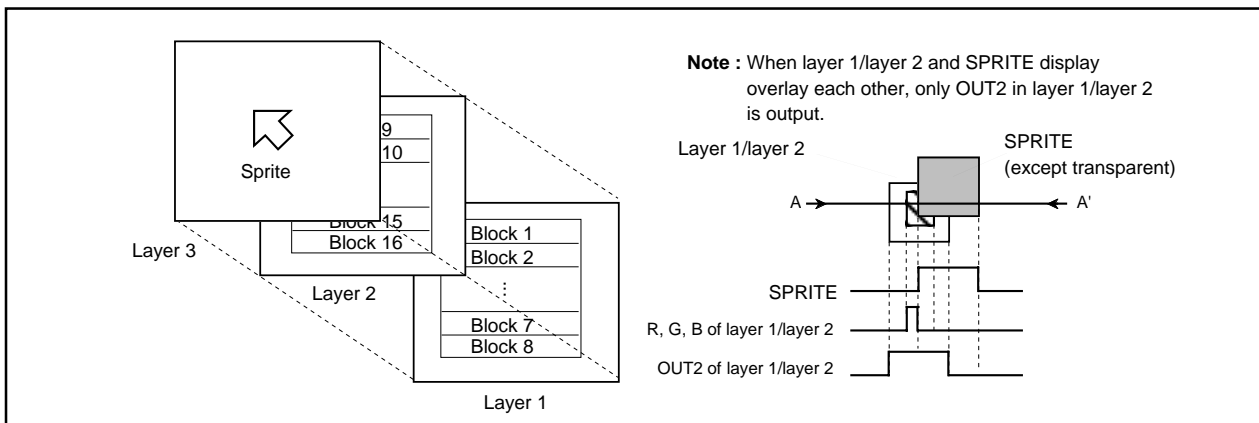
**2:** OSDP mode is always displayed on layer 1. And also, it cannot be overlapped with layer 2's block.

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

**Table 2.16.2 Mixing layer 1 and layer 2**

Parameter \ Block	Block in Layer 1	Block in Layer 2	
Display mode	CC, OSDS/L, CDOSD mode	OSDS/L, CDOSD mode	
Pre-divide ratio	X 1, X 2 (CC mode) X 1 to X 3 (OSD, CDOSD mode)	Same as layer 1 (See note)	
Dot size	1Tc X 1/2H, 1Tc X 1H (CC mode)	Pre-divide ratio = X 1	Pre-divide ratio = X 2
		1Tc X 1/2H 1Tc X 1H	1TcX1/2H, 1.5TcX1/2H 1TcX1H, 1.5TcX1H(Seenote)
	1Tc X 1H, 1Tc X 1/2H, 2Tc X 2H, 3Tc X 3H (OSDS/L, CDOSD mode)	<ul style="list-style-type: none"> <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 However, 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>		

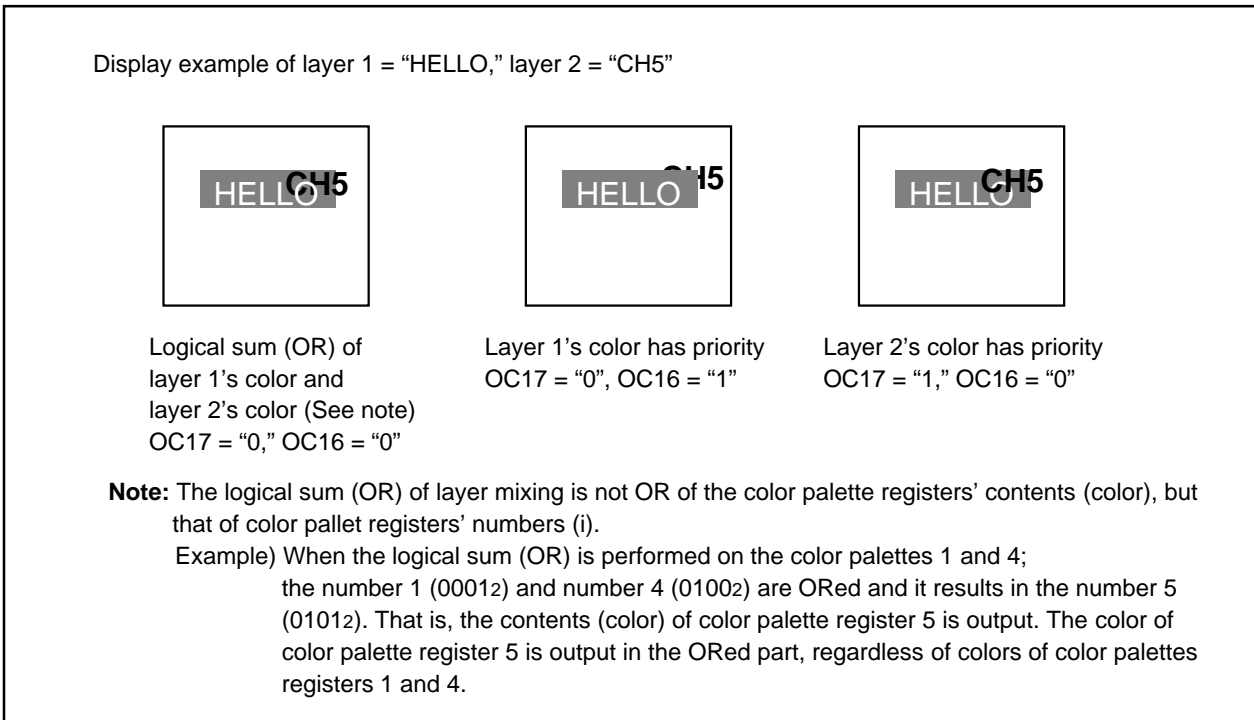
**Note:** In the OSDL mode, 1.5Tc size cannot be used.



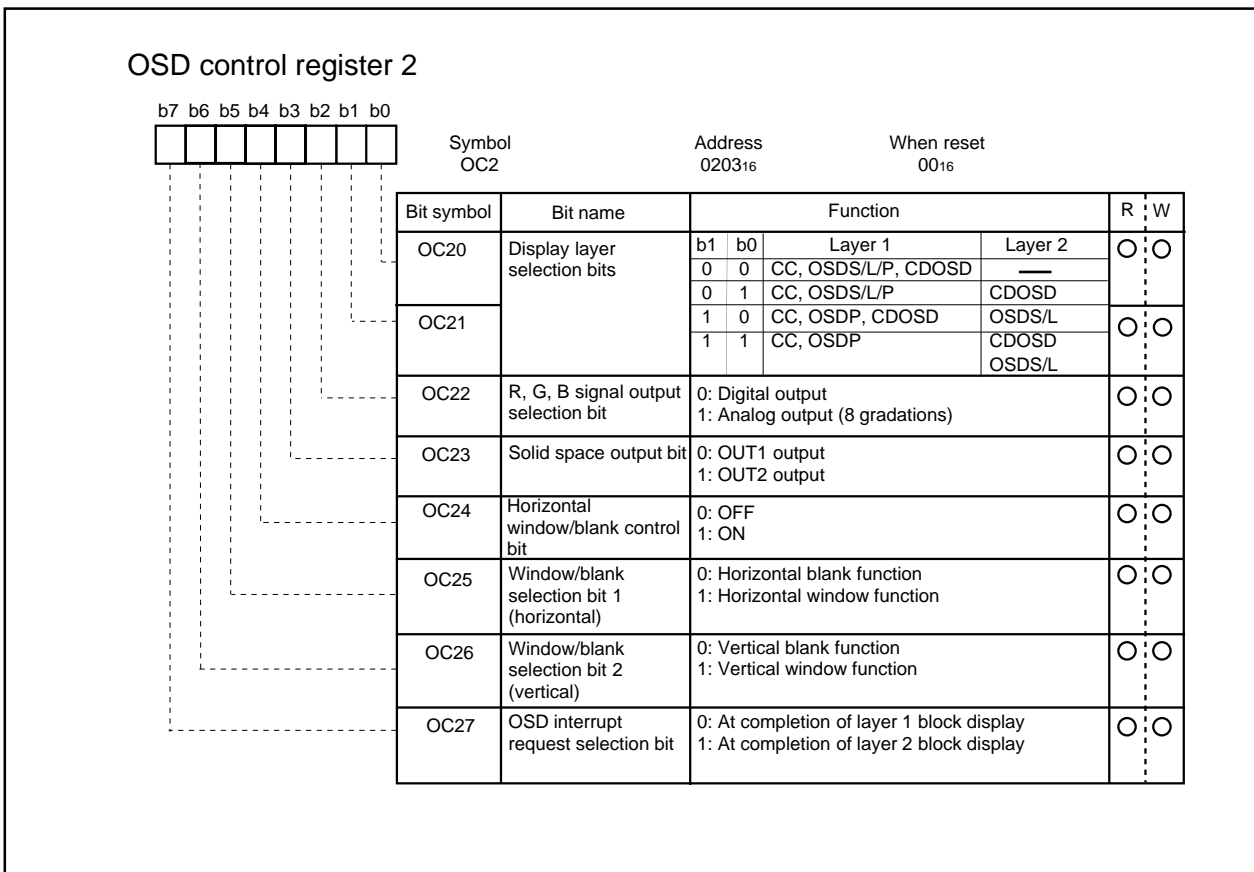
**Fig 2.16.5 Triple layer OSD**

**M306V5ME-XXXSP**  
**M306V5EESP**

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).

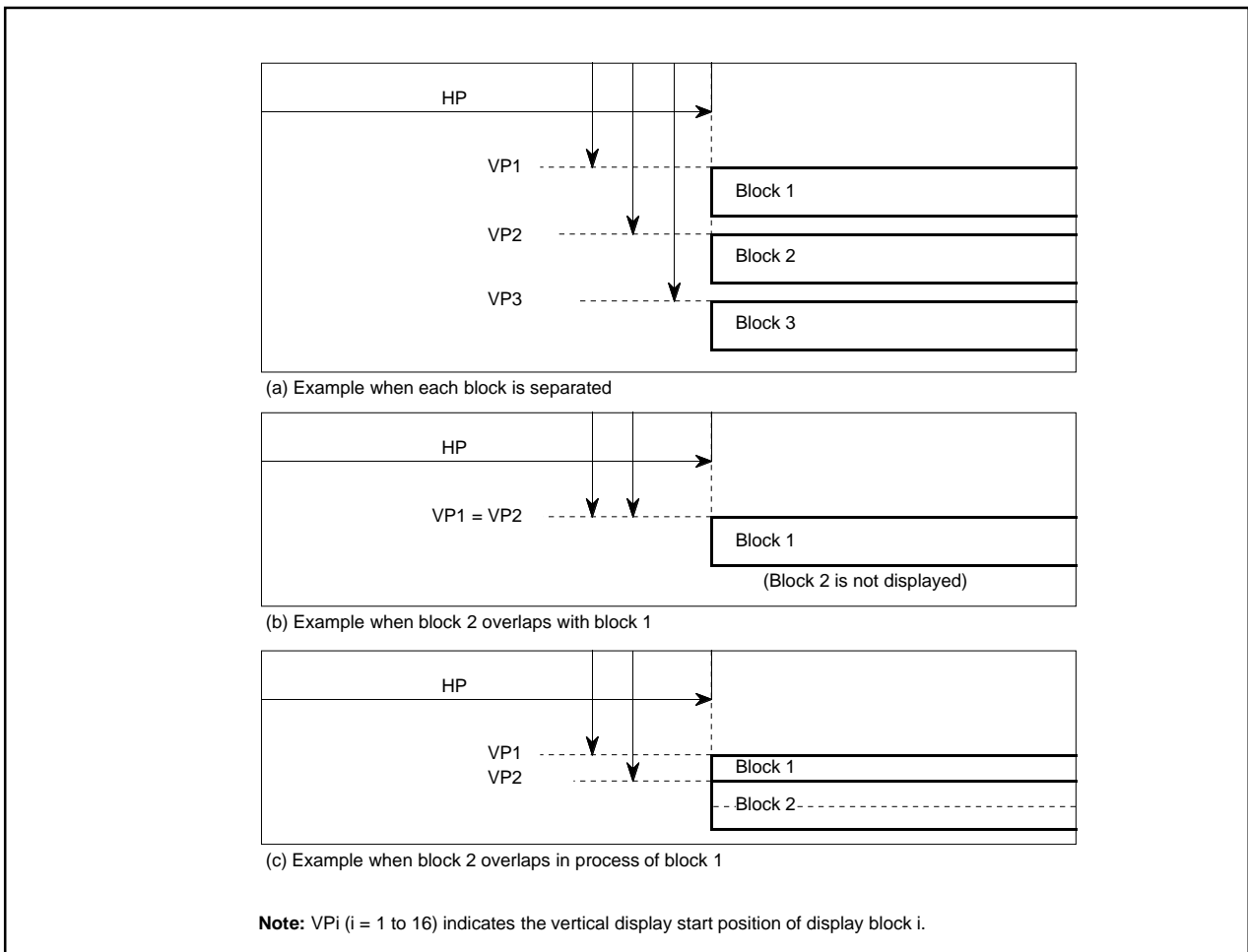
The display position of each block can be set in both horizontal and vertical directions by software.

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 in the same layer (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 in the same layer (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 OSDSP mode block: 20 dots in vertical from the vertical display start position.
  - \* In the case of OSDL mode block: 32 dots in vertical from the vertical display start position.
  - \* In the case of CC or CDOSD mode block: 26 dots in vertical from the vertical display start position.

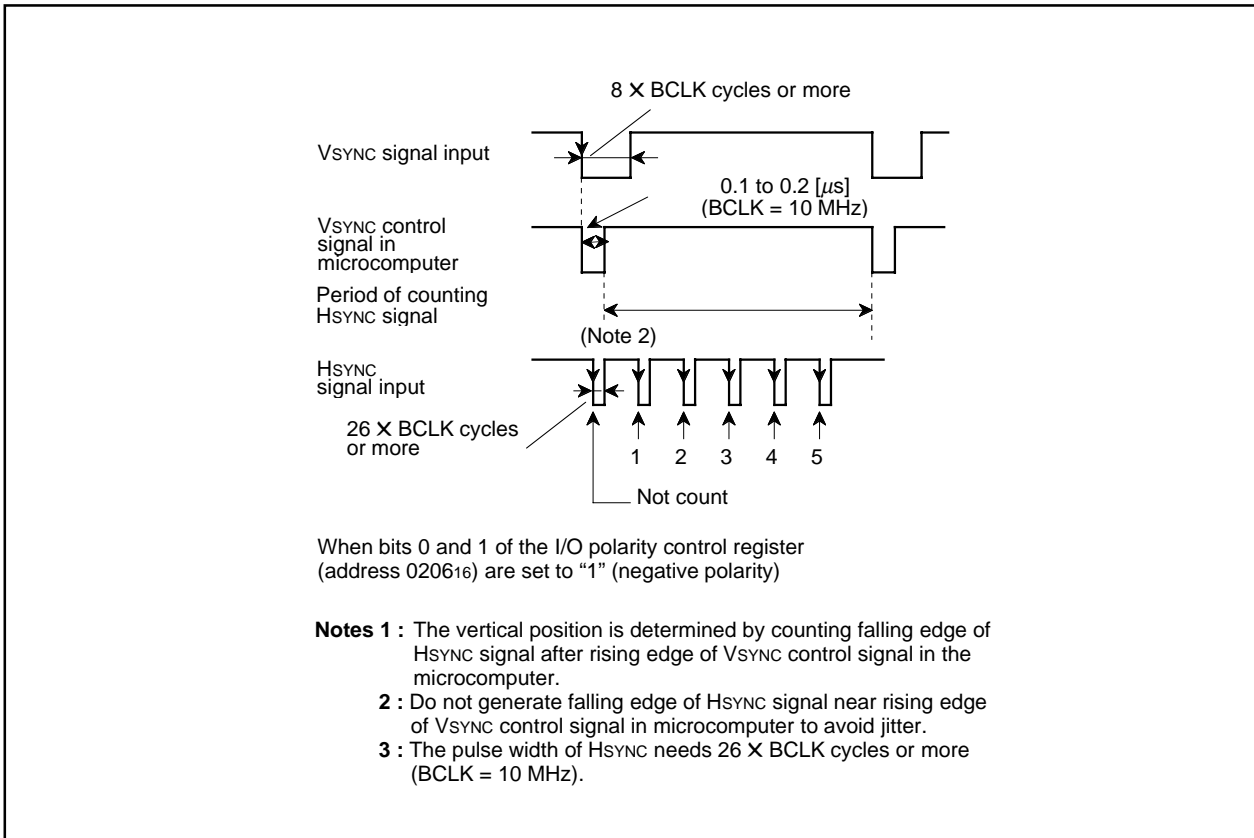


**Figure 2.16.8 Display position**

**M306V5ME-XXXSP**  
**M306V5EESP**

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**



**M306V5ME-XXXSP**  
**M306V5EESP**

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  $1T_H$  ( $T_H$ : 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.

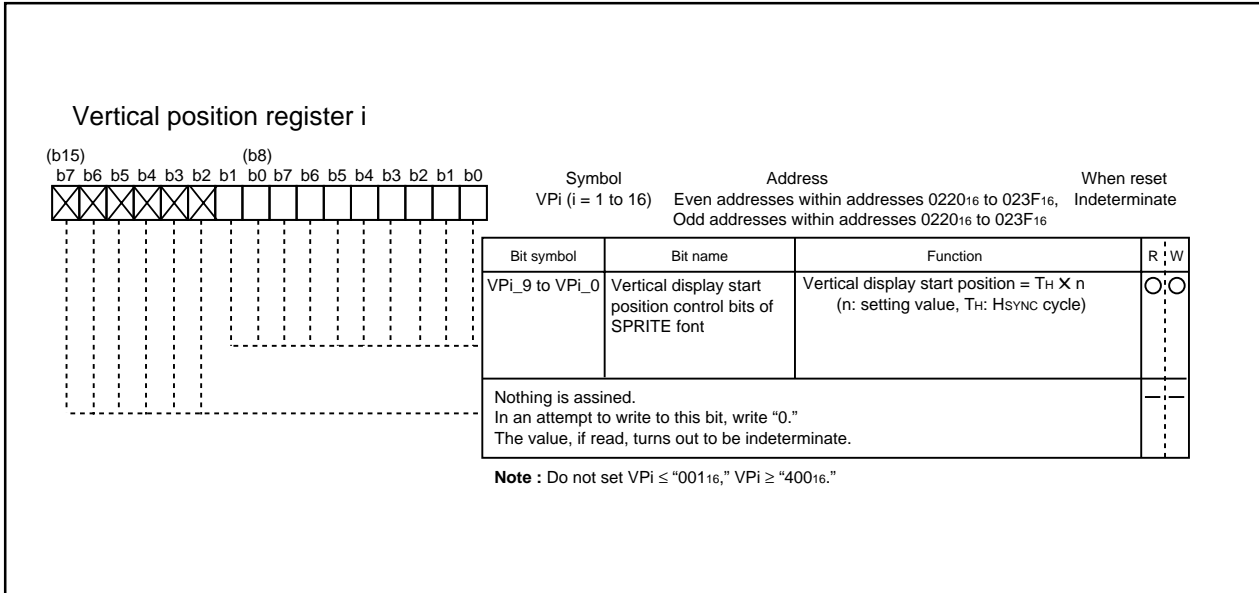


Figure 2.16.10 Vertical position register  $i$  ( $i = 1$  to 16)

The horizontal position is common to all blocks, and can be set in 256 steps (where 1 step is  $4T_{OSC}$ ,  $T_{OSC}$  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.

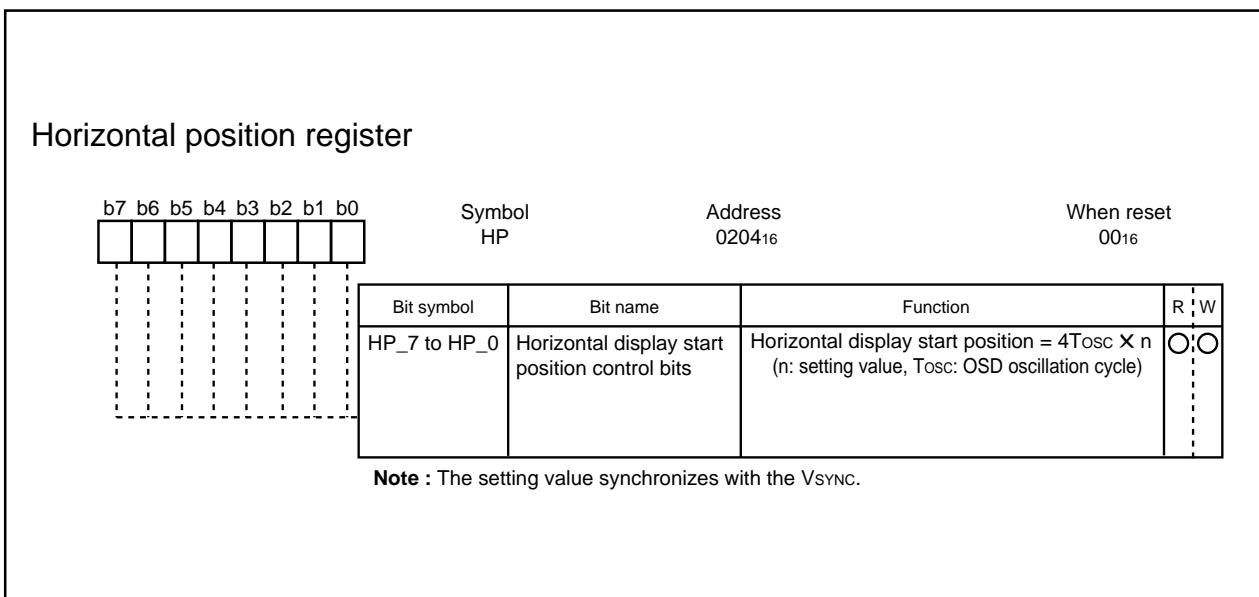


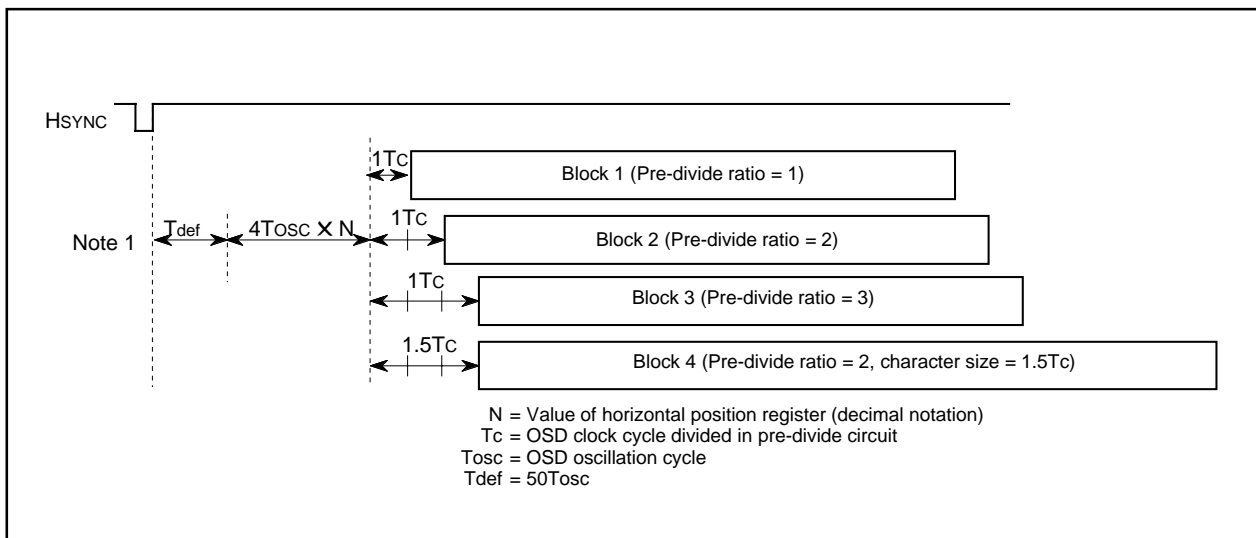
Figure 2.16.11 Horizontal position register

**M306V5ME-XXXSP**  
**M306V5EESP**

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.

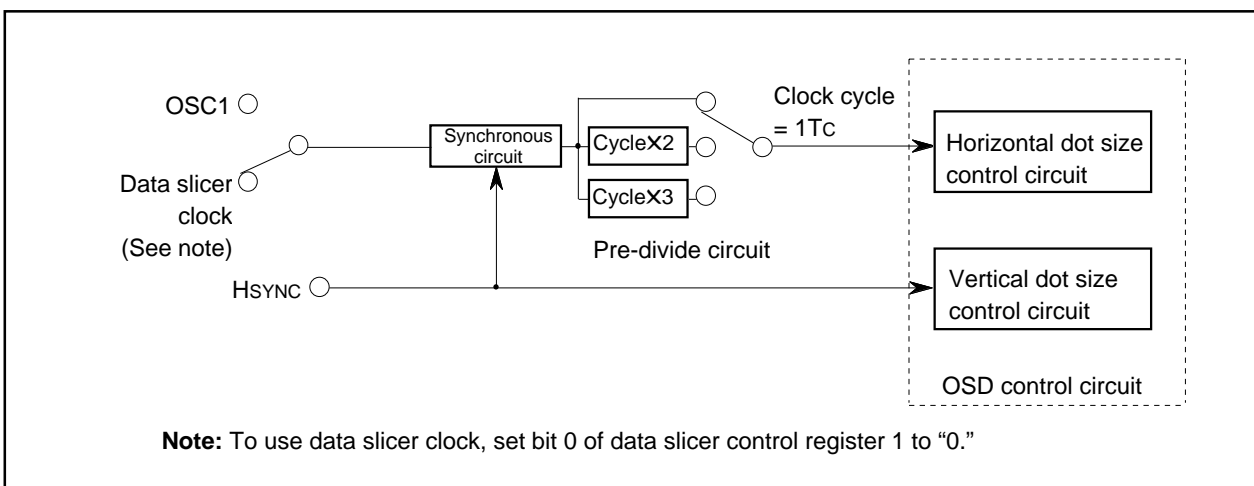
Refer to Figure 2.16.4 (the block control register i), refer to Figure 2.16.15 (the clock control register).

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

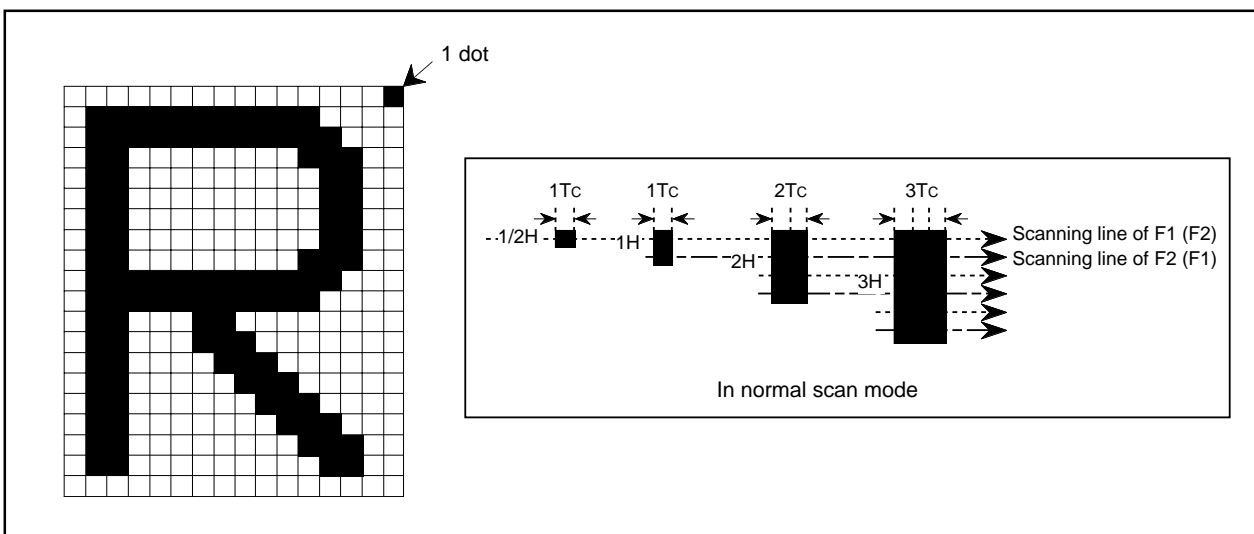
**Notes 1 :** The pre-divide ratio = 3 cannot be used in the CC mode.

**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 “2.16.18 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**

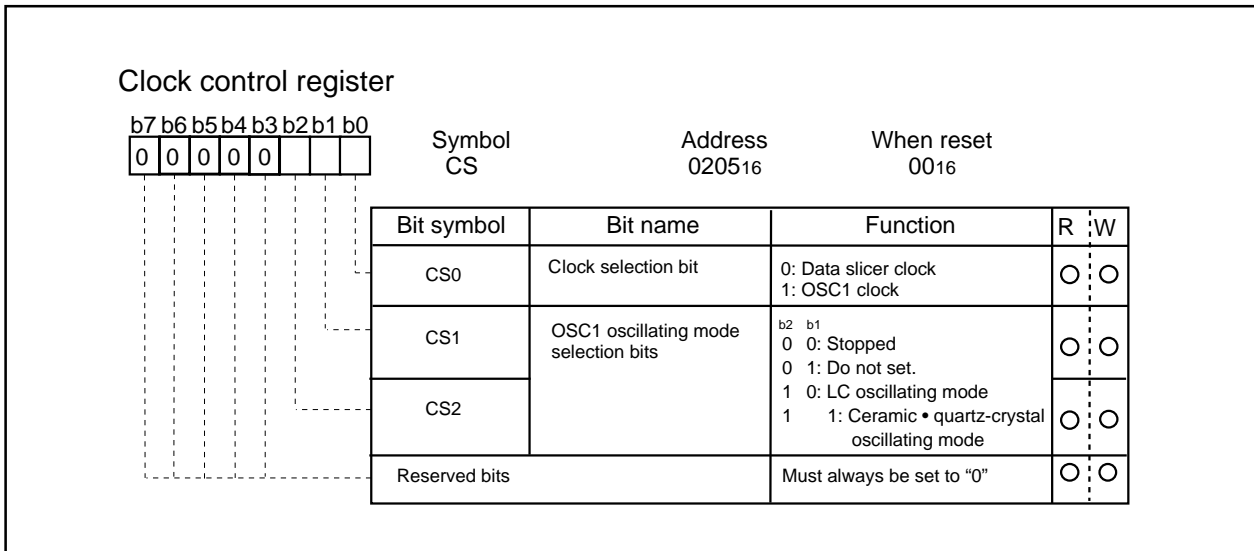
**M306V5ME-XXXSP**  
**M306V5EESP**

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

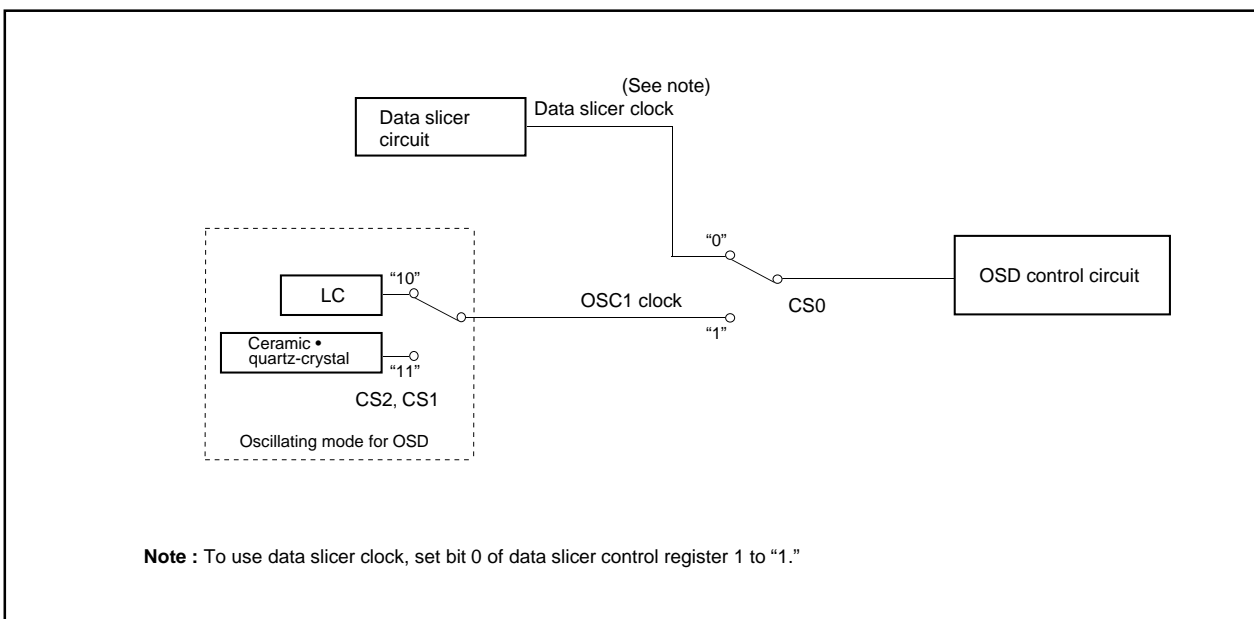
**2.16.4 Clock for OSD**

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**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**2.16.5 Field Determination Display**

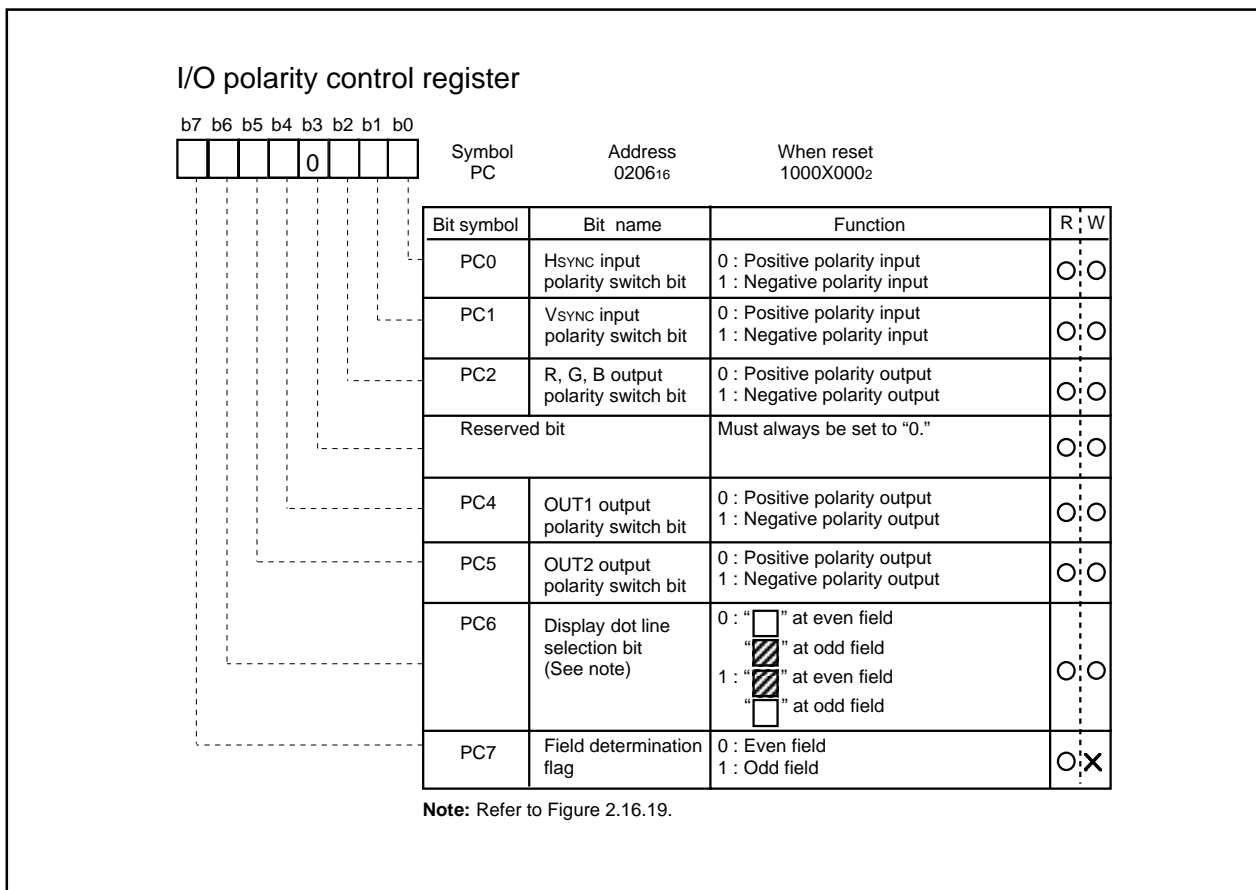
To display the block with vertical dot size of 1/2H, whether an even field or an odd field is determined through differences in a synchronizing signal waveform of interlacing system. The dot line 0 or 1 (refer to Figure 2.16.18) corresponding to the field is displayed alternately.

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**

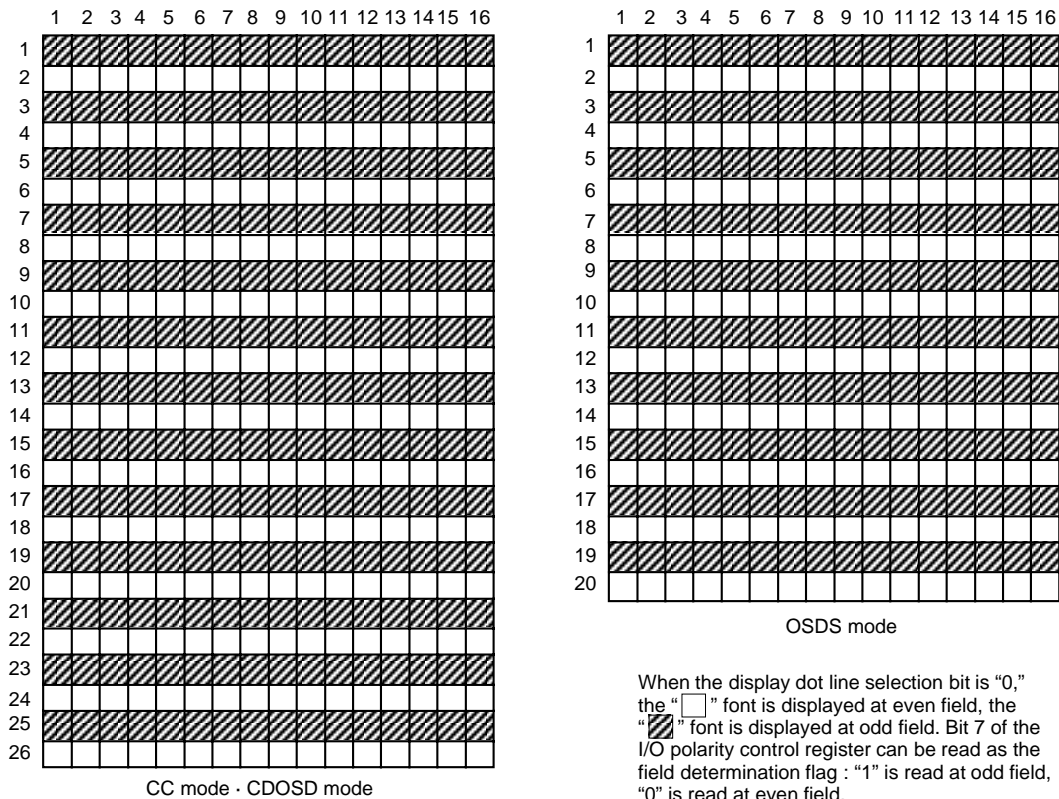
**M306V5ME-XXXSP**  
**M306V5EESP**

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

**M306V5ME-XXXSP**  
**M306V5EESP**

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 13FF<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>)**

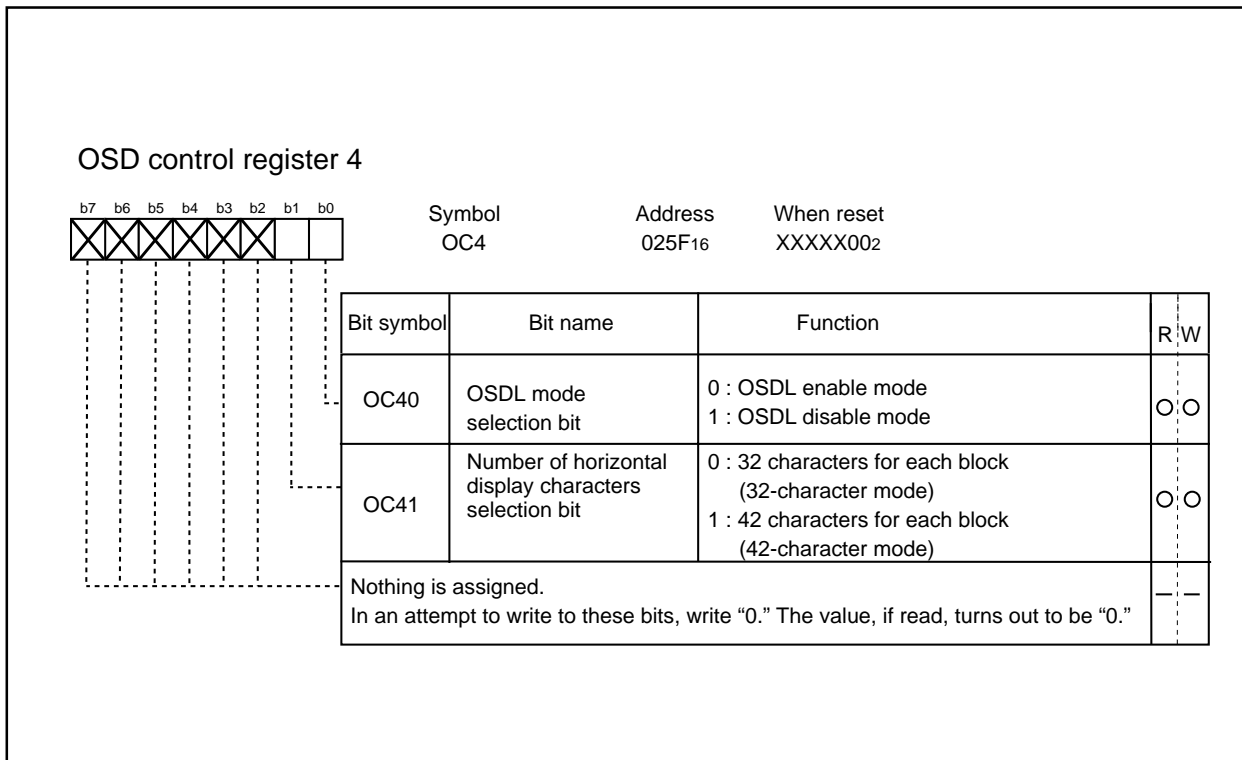
The dot pattern data for OSD characters is stored in the character font area in the OSD ROM and the CD font data for OSD characters is stored in the color dot font area in the OSD ROM. To specify the kinds of the character font and the CD font, it is necessary to write the character code into the OSD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

**M306V5ME-XXXSP**  
**M306V5EESP**

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

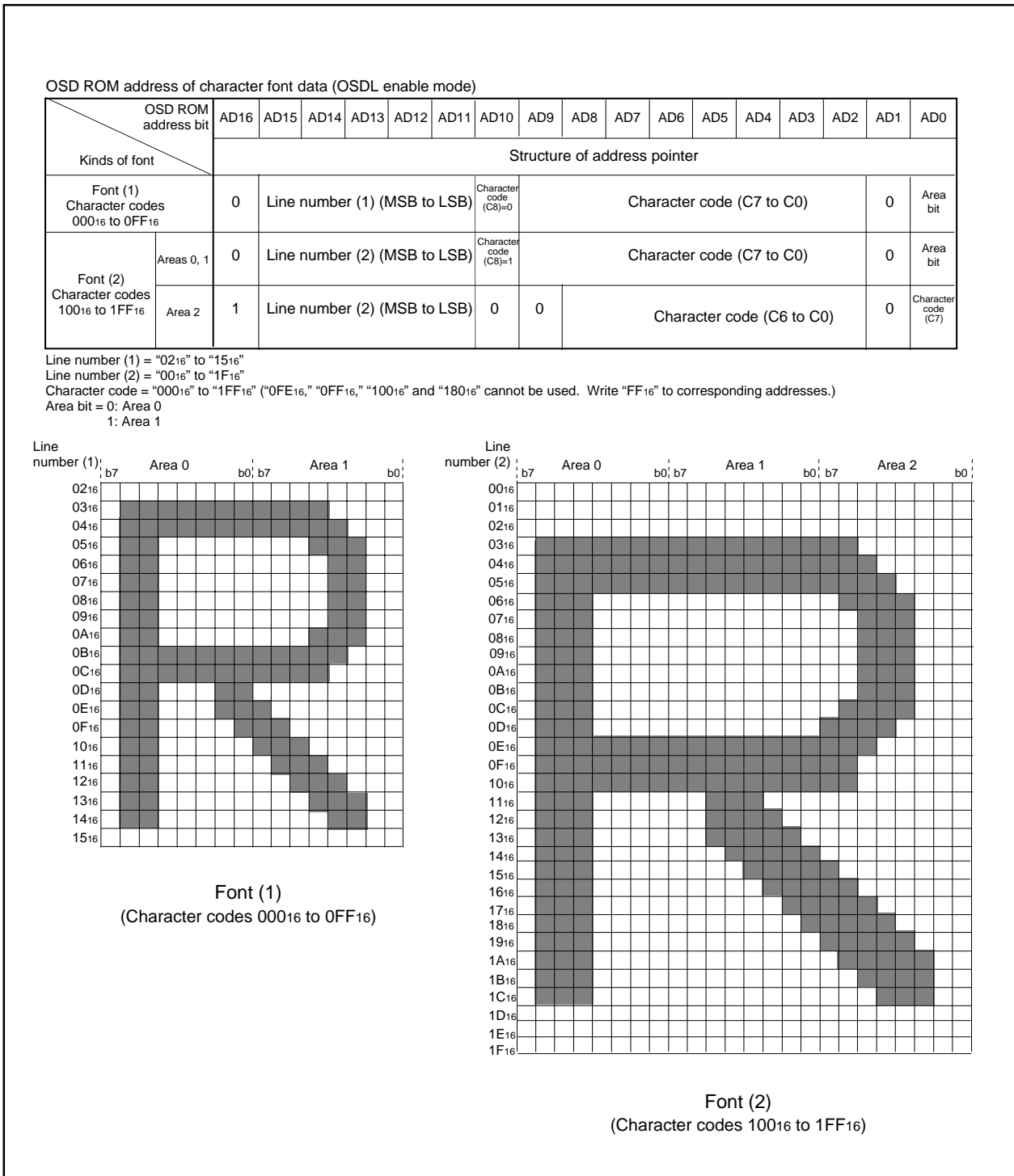


Figure 2.16.20 Character font data storing address (OSDL enable mode)



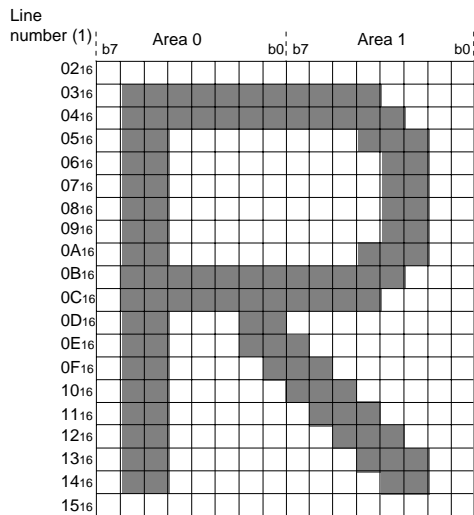
**M306V5ME-XXXSP**  
**M306V5EESP**

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

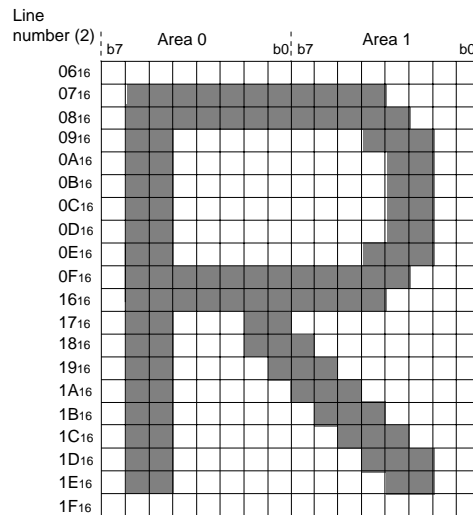
OSD ROM address of character font data (OSDL disable mode)

OSD ROM address bit		AD16	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0		
Kinds of font		Structure of address pointer																		
	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>)

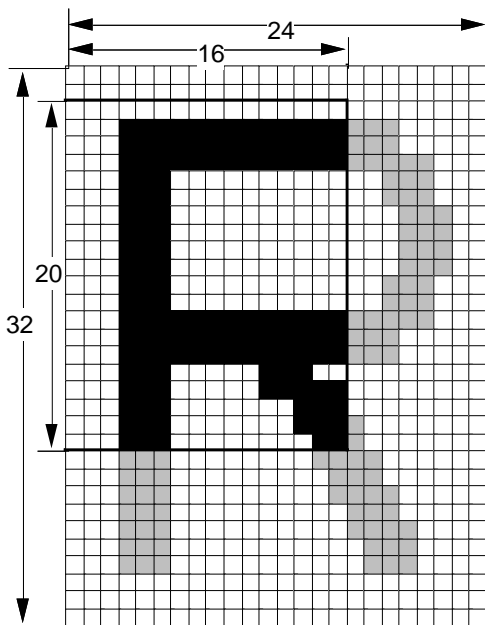
Figure 2.16.21 Character font data storing address (OSDL disable mode)

**M306V5ME-XXXSP**  
**M306V5EESP**

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/P	OSDL	Character size	CC	OSDS/P	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	S	Used	Used	Display OFF
	200 <sub>16</sub> to 27F <sub>16</sub>		Not used (See note 3)			S		Used	Display OFF
	280 <sub>16</sub> to 2FF <sub>16</sub>					Not used (See note 3)	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/P mode (no border).
- 3:** As setting this make output of font data indeterminate, do not use. However, "3FE<sub>16</sub>" and "3FF<sub>16</sub>" can be used as character codes of blank font output in OSDP mode.

**Figure 2.16.22 Conditions for each OSDL enable/disable mode**

# M306V5ME-XXXSP M306V5EESP

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

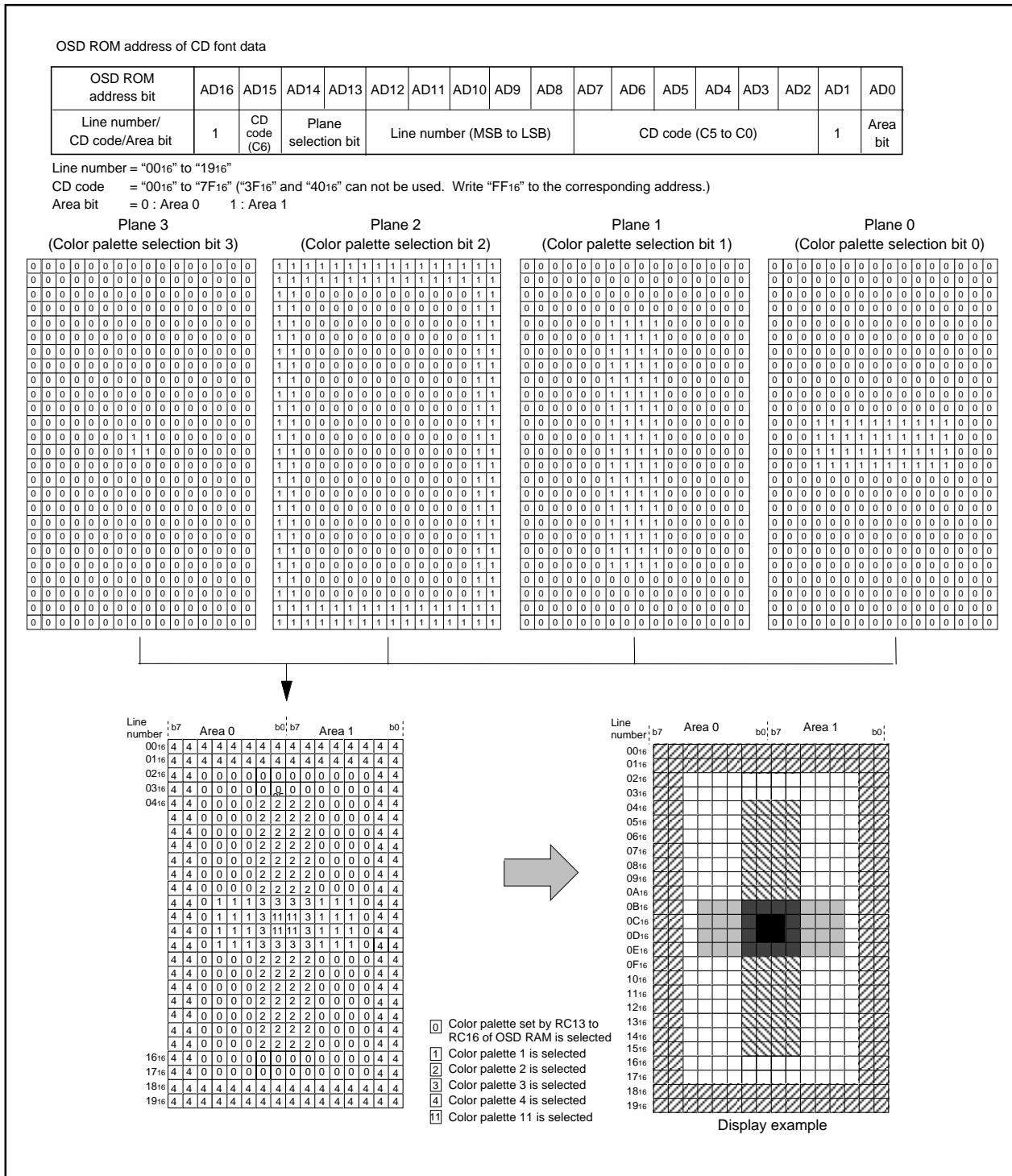


Figure 2.16.23 Color dot font data storing address

**M306V5ME-XXXSP**  
**M306V5EESP**

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.

**Note :** 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

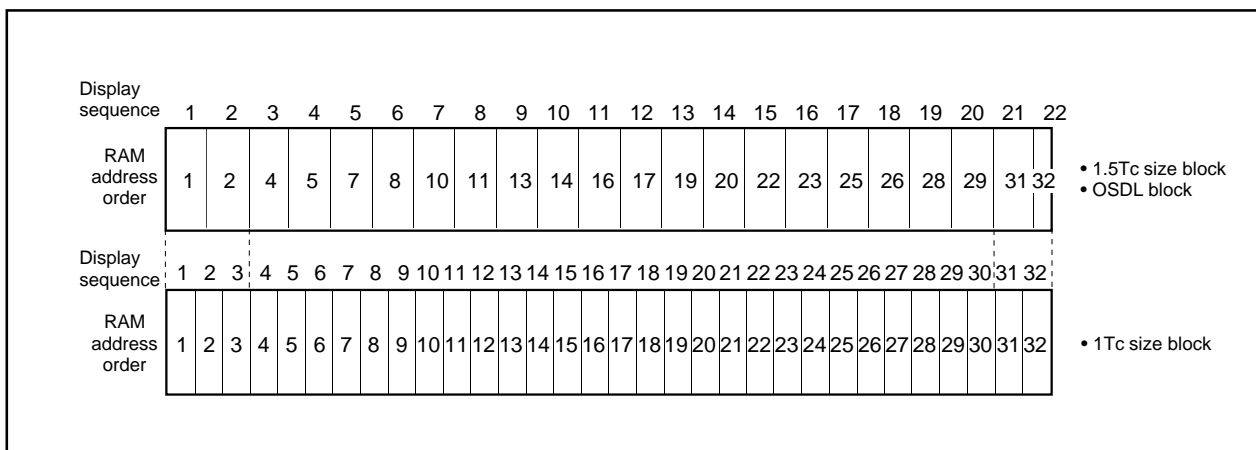
The character is not displayed, and only the left 1/3 part of the 22<sup>nd</sup> character background is displayed in the 22<sup>nd</sup>'s character area. When not displaying this background, set transparent for character background color.

- 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)**

# M306V5ME-XXXSP

## M306V5EESP

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>
	⋮	⋮	⋮	⋮
	31st character	043C <sub>16</sub>	043D <sub>16</sub>	04BC <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>
	⋮	⋮	⋮	⋮
	31st character	047C <sub>16</sub>	047D <sub>16</sub>	04FC <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>
	⋮	⋮	⋮	⋮
	31st character	053C <sub>16</sub>	053D <sub>16</sub>	05BC <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>
	⋮	⋮	⋮	⋮
	31st character	057C <sub>16</sub>	057D <sub>16</sub>	05FC <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>
	⋮	⋮	⋮	⋮
	31st character	063C <sub>16</sub>	063D <sub>16</sub>	06BC <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>
	⋮	⋮	⋮	⋮
	31st character	067C <sub>16</sub>	067D <sub>16</sub>	06FC <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>
	⋮	⋮	⋮	⋮
	31st character	073C <sub>16</sub>	073D <sub>16</sub>	07BC <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>
	⋮	⋮	⋮	⋮
	31st character	077C <sub>16</sub>	077D <sub>16</sub>	07FC <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>
	⋮	⋮	⋮	⋮
	31st character	083C <sub>16</sub>	083D <sub>16</sub>	08BC <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>
	⋮	⋮	⋮	⋮
	31st character	087C <sub>16</sub>	087D <sub>16</sub>	08FC <sub>16</sub>
	32nd character	087E <sub>16</sub>	087F <sub>16</sub>	08FE <sub>16</sub>

**M306V5ME-XXXSP**  
**M306V5EESP**

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>

**M306V5ME-XXXSP**  
**M306V5EESP**

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 7	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>
42nd character	0E32 <sub>16</sub>	0E33 <sub>16</sub>	0EB2 <sub>16</sub>	

**M306V5ME-XXXSP**  
**M306V5EESP**

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>	



**M306V5ME-XXXSP**  
**M306V5EESP**

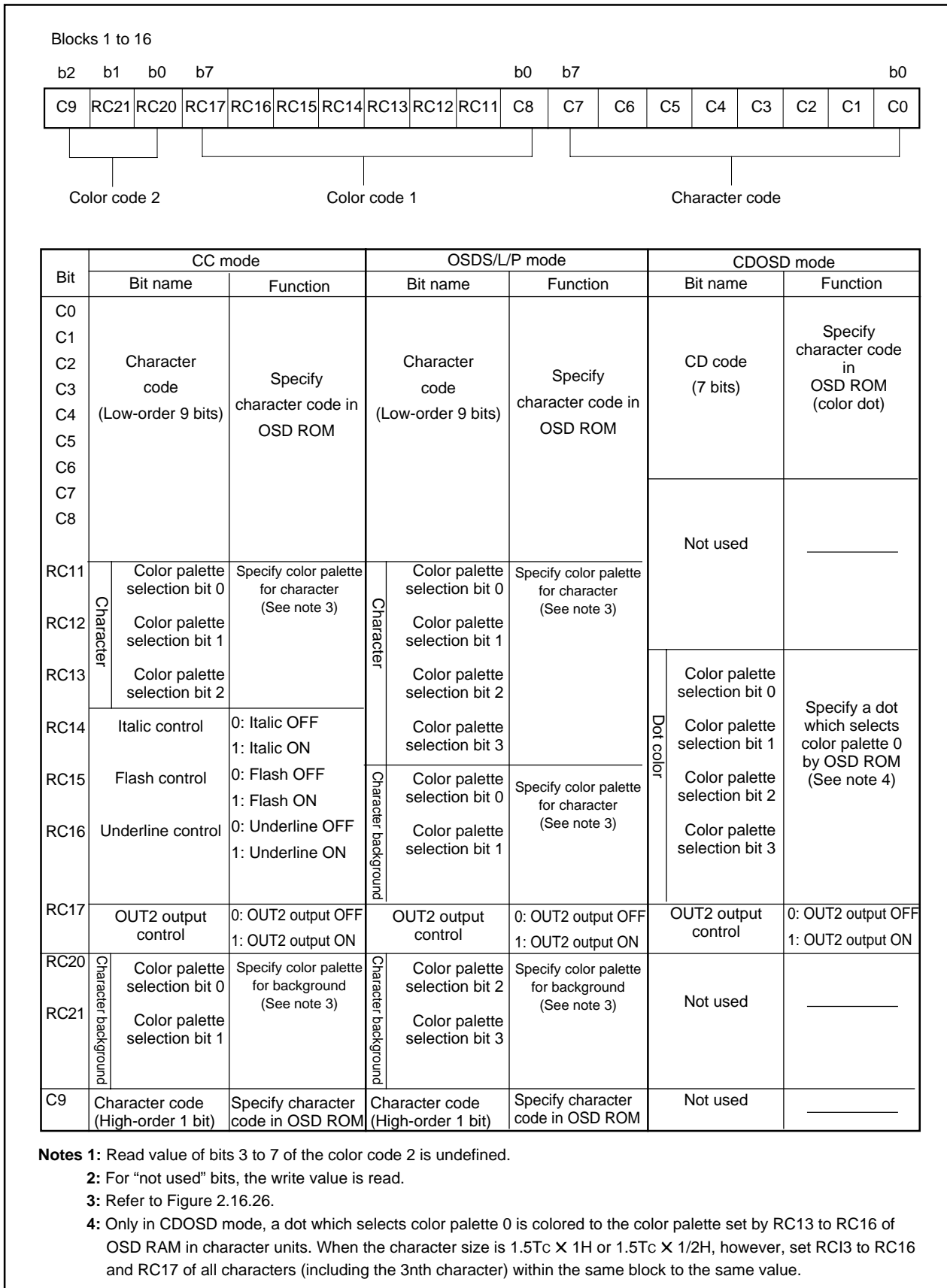
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**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>

**M306V5ME-XXXSP**  
**M306V5EESP**

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



**Figure 2.16.25 Structure of OSD RAM**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**(3) OSD RAM (OSD RAM for SPRITE, addresses 1000<sub>16</sub> to 13E7<sub>16</sub>)**

The OSD RAM for SPRITE fonts 1 and 2, consisting of 4 planes for each font, is assigned to addresses 1000<sub>16</sub> to 13E7<sub>16</sub>. Each plane corresponds to each color palette selection bit and the color palette of each dot is determined from among 16 kinds.

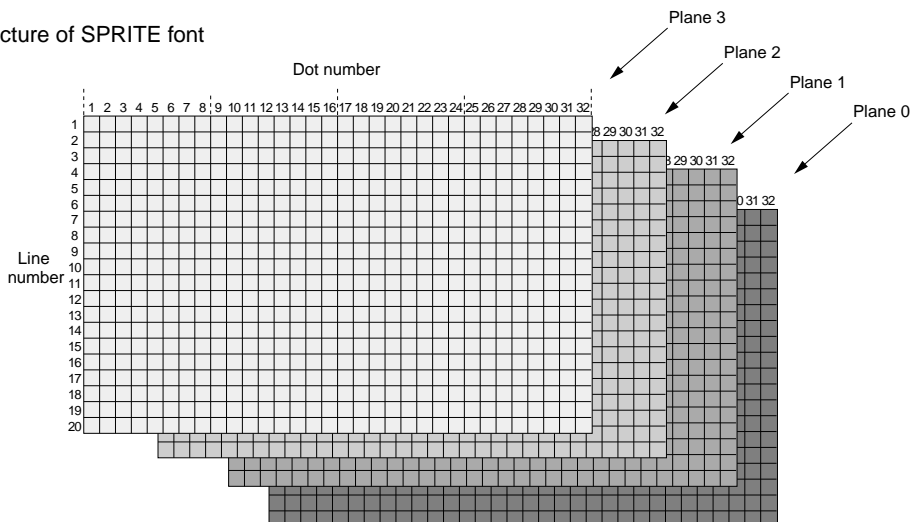
**Table 2.16.8 OSD RAM address (SPRITE font 1)**

Planes	Plane 3 (Color palette selection bit 3)				Plane 2 (Color palette selection bit 2)				Plane 1 (Color palette selection bit 1)				Plane 0 (Color palette selection bit 0)			
	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32
Dots	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32
Bits	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0
Line 1	10C0 <sub>16</sub>	10C1 <sub>16</sub>	11C0 <sub>16</sub>	11C1 <sub>16</sub>	1080 <sub>16</sub>	1081 <sub>16</sub>	1180 <sub>16</sub>	1181 <sub>16</sub>	1040 <sub>16</sub>	1041 <sub>16</sub>	1140 <sub>16</sub>	1141 <sub>16</sub>	1000 <sub>16</sub>	1001 <sub>16</sub>	1100 <sub>16</sub>	1101 <sub>16</sub>
Line 2	10C2 <sub>16</sub>	10C3 <sub>16</sub>	11C2 <sub>16</sub>	11C3 <sub>16</sub>	1082 <sub>16</sub>	1083 <sub>16</sub>	1182 <sub>16</sub>	1183 <sub>16</sub>	1042 <sub>16</sub>	1043 <sub>16</sub>	1142 <sub>16</sub>	1143 <sub>16</sub>	1002 <sub>16</sub>	1003 <sub>16</sub>	1102 <sub>16</sub>	1103 <sub>16</sub>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Line 19	10E4 <sub>16</sub>	10E5 <sub>16</sub>	11E4 <sub>16</sub>	11E5 <sub>16</sub>	10A4 <sub>16</sub>	10A5 <sub>16</sub>	11A4 <sub>16</sub>	11A5 <sub>16</sub>	1064 <sub>16</sub>	1065 <sub>16</sub>	1164 <sub>16</sub>	1165 <sub>16</sub>	1024 <sub>16</sub>	1025 <sub>16</sub>	1124 <sub>16</sub>	1125 <sub>16</sub>
Line 20	10E6 <sub>16</sub>	10E7 <sub>16</sub>	11E6 <sub>16</sub>	11E7 <sub>16</sub>	10A6 <sub>16</sub>	10A7 <sub>16</sub>	11A6 <sub>16</sub>	11A7 <sub>16</sub>	1066 <sub>16</sub>	1067 <sub>16</sub>	1166 <sub>16</sub>	1167 <sub>16</sub>	1026 <sub>16</sub>	1027 <sub>16</sub>	1126 <sub>16</sub>	1127 <sub>16</sub>

**Table 2.16.9 OSD RAM address (SPRITE font 2)**

Planes	Plane 3 (Color palette selection bit 3)				Plane 2 (Color palette selection bit 2)				Plane 1 (Color palette selection bit 1)				Plane 0 (Color palette selection bit 0)			
	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32
Dots	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32	1 to 8	9 to 16	17 to 24	25 to 32
Bits	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0	b7 to b0
Line 1	12C0 <sub>16</sub>	12C1 <sub>16</sub>	13C0 <sub>16</sub>	13C1 <sub>16</sub>	1280 <sub>16</sub>	1281 <sub>16</sub>	1380 <sub>16</sub>	1381 <sub>16</sub>	1240 <sub>16</sub>	1241 <sub>16</sub>	1340 <sub>16</sub>	1341 <sub>16</sub>	1200 <sub>16</sub>	1201 <sub>16</sub>	1300 <sub>16</sub>	1301 <sub>16</sub>
Line 2	12C2 <sub>16</sub>	12C3 <sub>16</sub>	13C2 <sub>16</sub>	13C3 <sub>16</sub>	1282 <sub>16</sub>	1283 <sub>16</sub>	1382 <sub>16</sub>	1383 <sub>16</sub>	1242 <sub>16</sub>	1243 <sub>16</sub>	1342 <sub>16</sub>	1343 <sub>16</sub>	1202 <sub>16</sub>	1203 <sub>16</sub>	1302 <sub>16</sub>	1303 <sub>16</sub>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Line 19	12E4 <sub>16</sub>	12E5 <sub>16</sub>	13E4 <sub>16</sub>	13E5 <sub>16</sub>	12A4 <sub>16</sub>	12A5 <sub>16</sub>	13A4 <sub>16</sub>	13A5 <sub>16</sub>	1264 <sub>16</sub>	1265 <sub>16</sub>	1364 <sub>16</sub>	1365 <sub>16</sub>	1224 <sub>16</sub>	1225 <sub>16</sub>	1324 <sub>16</sub>	1325 <sub>16</sub>
Line 20	12E6 <sub>16</sub>	12E7 <sub>16</sub>	13E6 <sub>16</sub>	13E7 <sub>16</sub>	12A6 <sub>16</sub>	12A7 <sub>16</sub>	13A6 <sub>16</sub>	13A7 <sub>16</sub>	1266 <sub>16</sub>	1267 <sub>16</sub>	1366 <sub>16</sub>	1367 <sub>16</sub>	1226 <sub>16</sub>	1227 <sub>16</sub>	1326 <sub>16</sub>	1327 <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 512 colors available. The setting procedure for character colors is as follows:

- CC mode ..... 8 kinds  
Color palette selection range (color palettes 0 to 7 or 8 to 15) can be selected by bit 0 of the OSD control register 3 (address 020716). Color palettes are set by bits RC11 to RC13 of the OSD RAM from among the selection range.
- OSDS/L/P mode ..... 16 kinds  
Color palettes are set by bits RC11 to RC14 of the OSD RAM.
- CDOSD mode ..... 16 kinds  
Color palettes are set in dot units according to CD font data.  
Only in CDOSD mode, a dot which selects color palette 0 or 8 is colored to the color palette set by RC13 to RC16 of OSD RAM in character units (refer to Figure 2.16.28).
- SPRITE display ..... 16 kinds  
Color palettes are set in dot units according to the CD font data.

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

**2:** Color palette 0 (transparent) and the transparent setting of other color palettes will differ. When there are multiple layers overlapping (on top of each other, piled up), and the priority layer is color palette 0 (transparent), the bottom layer is displayed, but if the priority layer is the transparent setting of any other color palette, the background is displayed without displaying the bottom layer (refer to Figure 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/L/P mode ..... 16 kinds  
Color palettes are set by bits RC15, RC16, RC20, and RC21 of the OSD RAM.

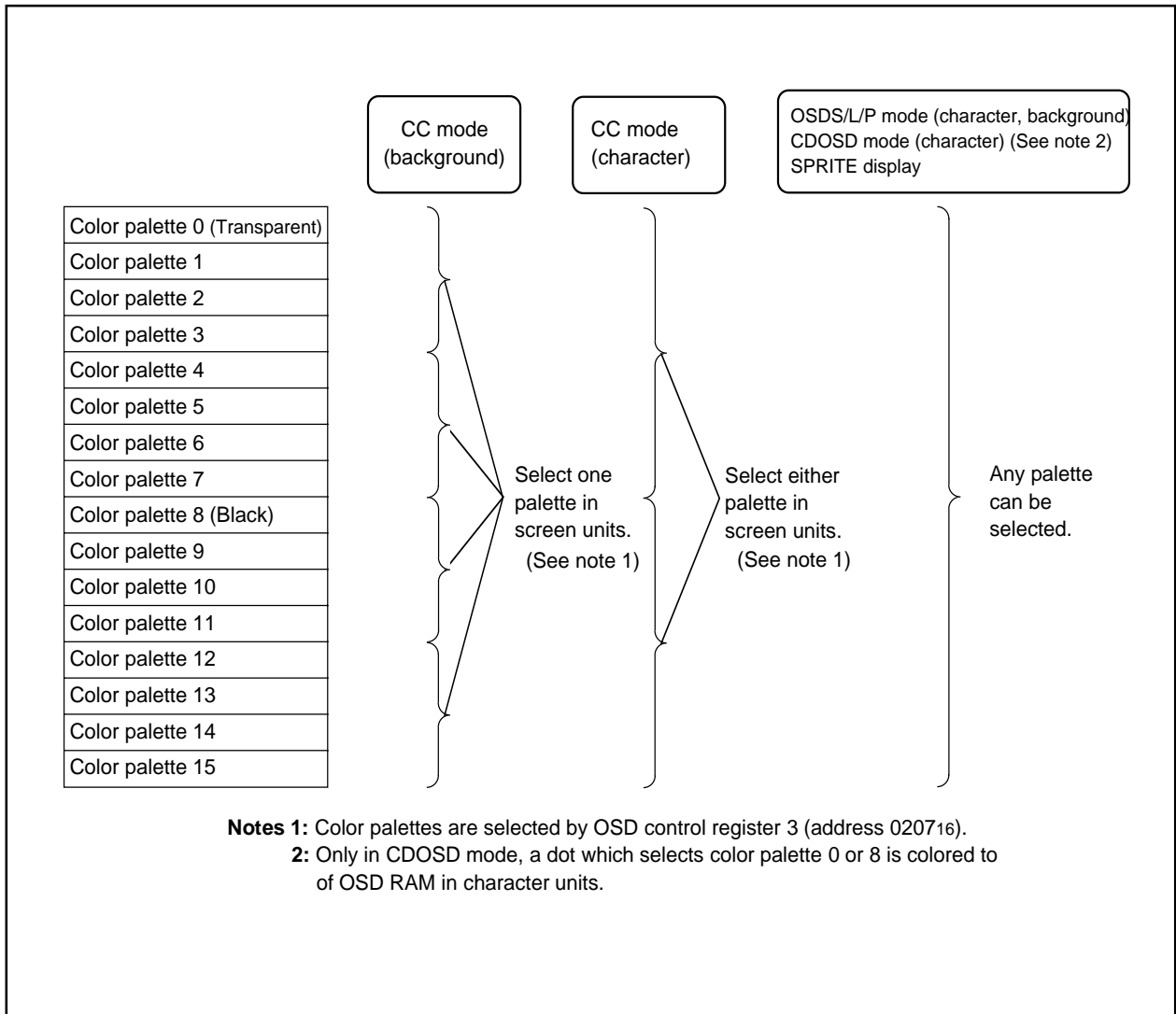
**Note:** The character background is displayed in the following part:

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

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

**M306V5ME-XXXSP**  
**M306V5EESP**

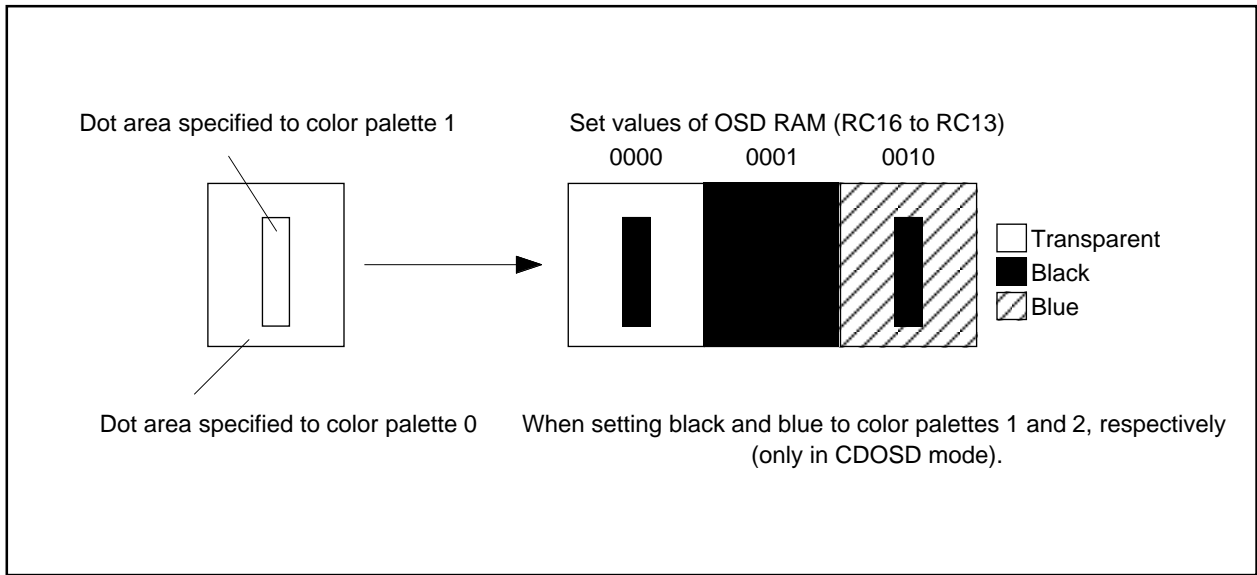
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



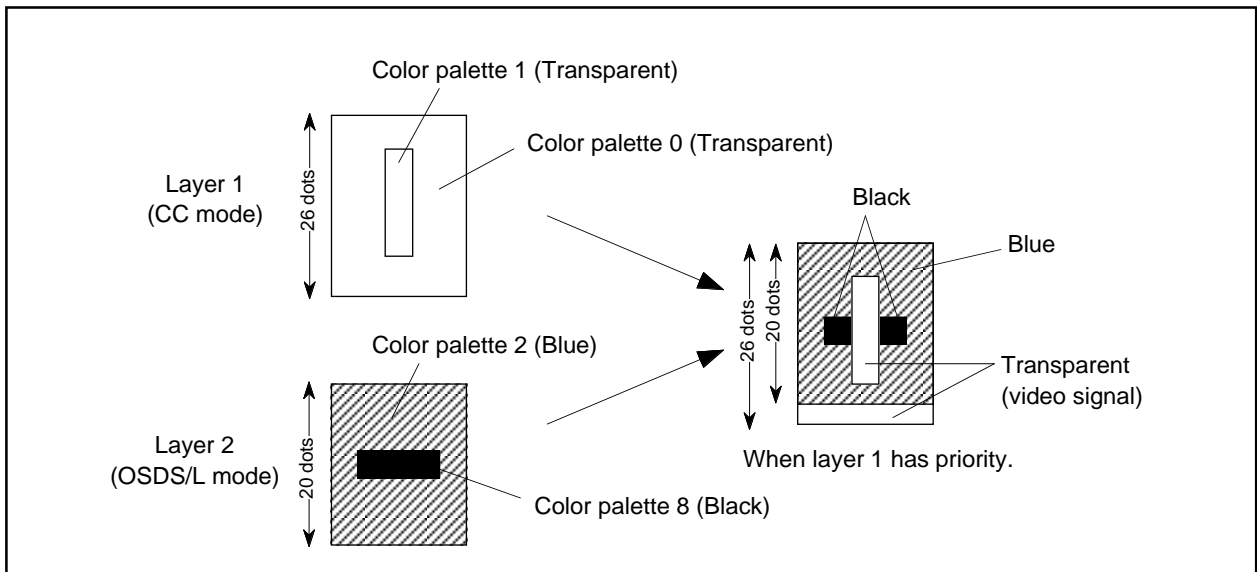
**Figure 2.16.26 Color palette selection**

**M306V5ME-XXXSP**  
**M306V5EESP**

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



**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**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

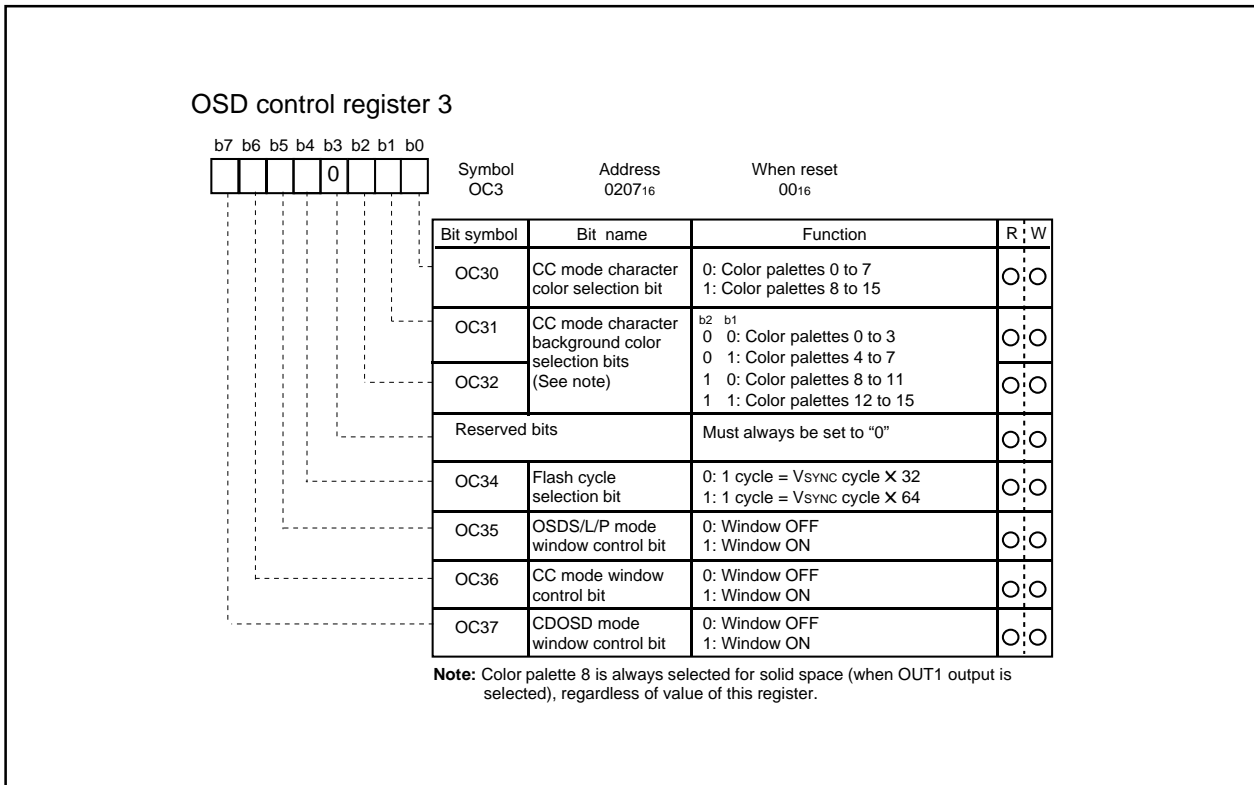
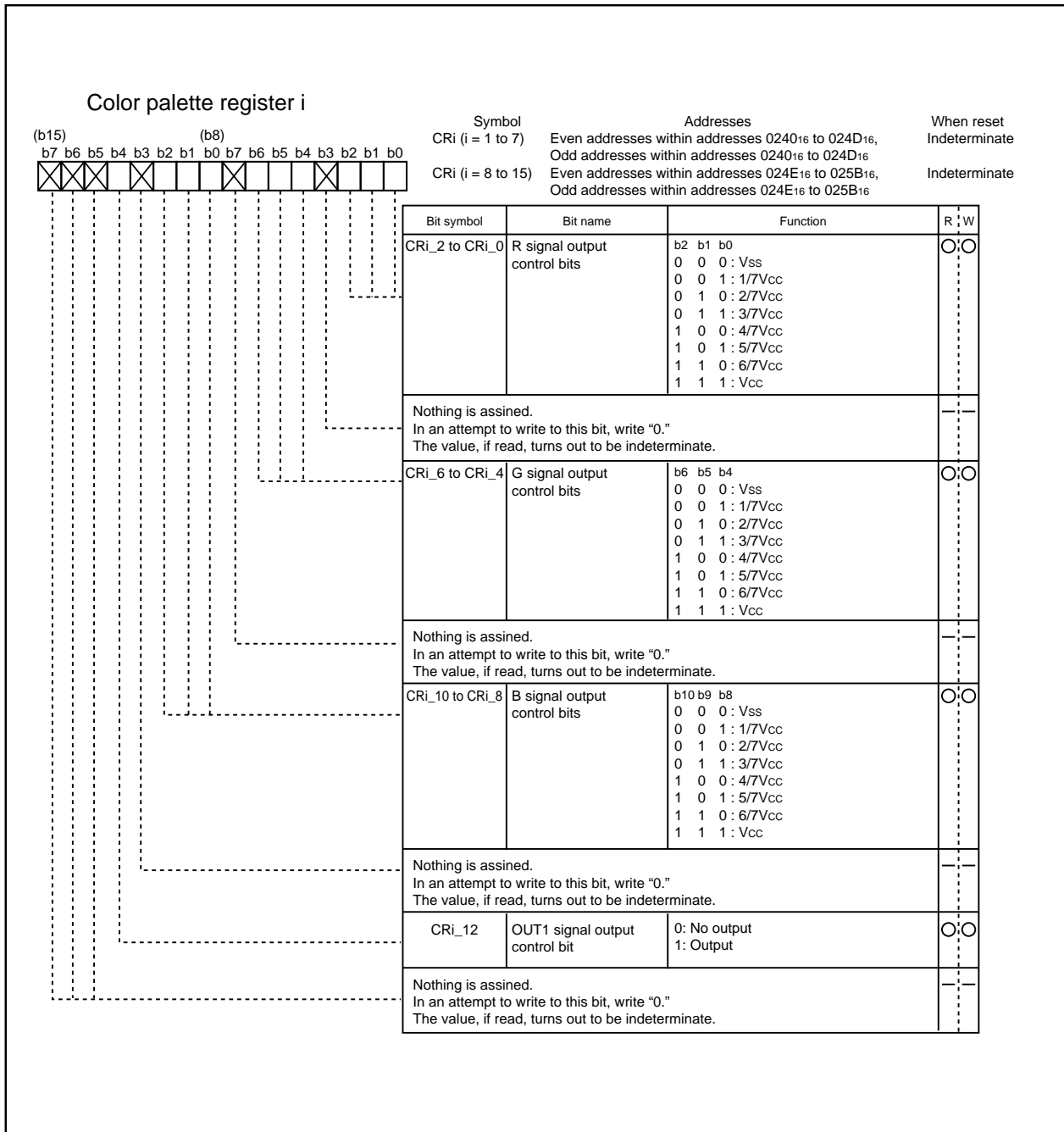


Figure 2.16.29 OSD control register 3

**M306V5ME-XXXSP**  
**M306V5EESP**

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



**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.

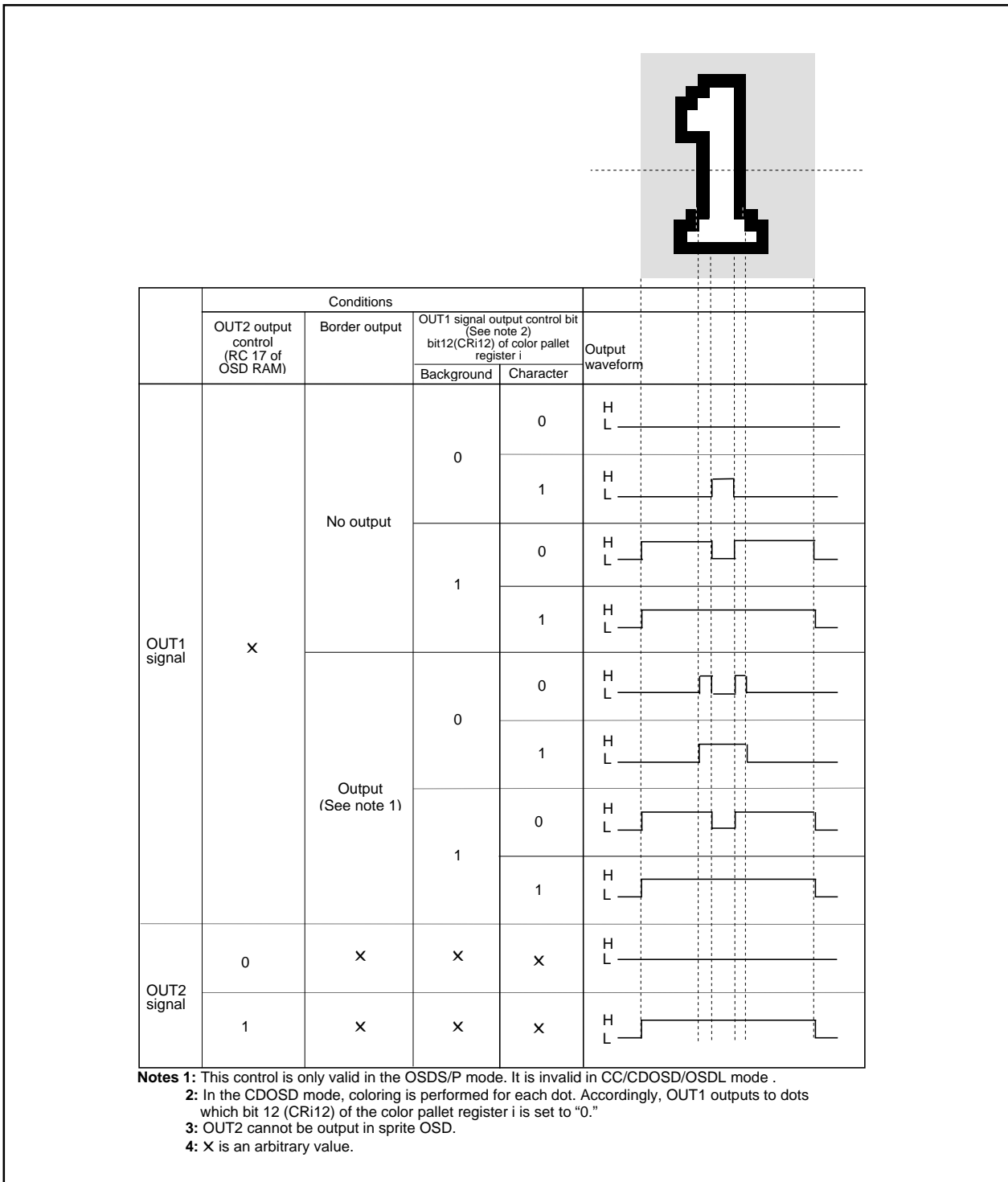


Figure 2.16.31 Setting value for controlling OUT1, OUT2 and corresponding output waveform

## 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/P 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>

- When bit 4 = "0"
  - VSYNC cycle X 24 ≈ 400 ms (at flash ON)
  - VSYNC cycle X 8 ≈ 133 ms (at flash OFF)
- When bit 4 = "1"
  - VSYNC cycle X 48 ≈ 800 ms (at flash ON)
  - VSYNC cycle X 8 ≈ 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 X 5 steps is displayed ; when the pre-divide ratio = 2, the italic character with slant of 1/2 dot X 10 steps is displayed (refer to Figure 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).

**M306V5ME-XXXSP**  
**M306V5EESP**

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

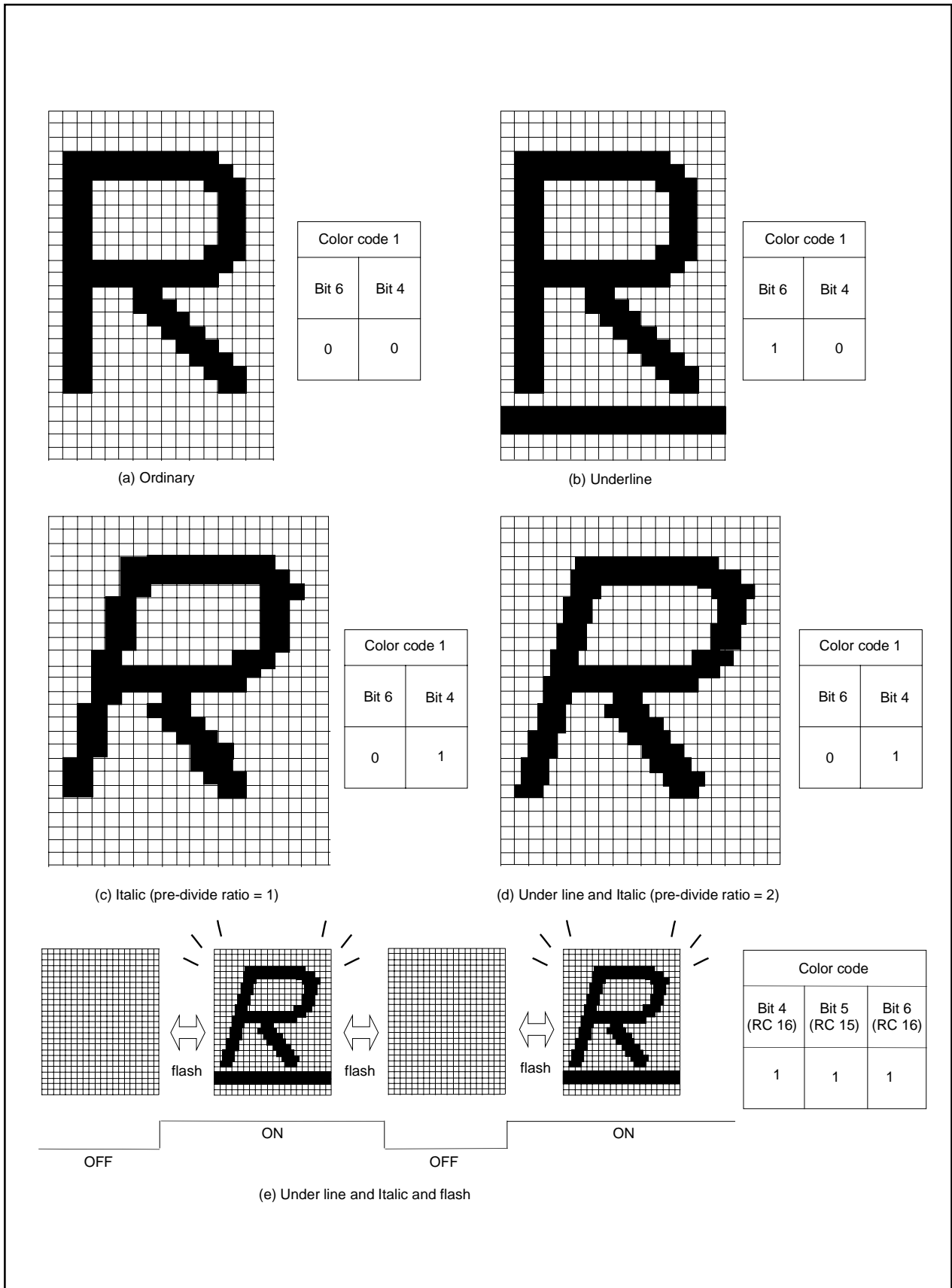
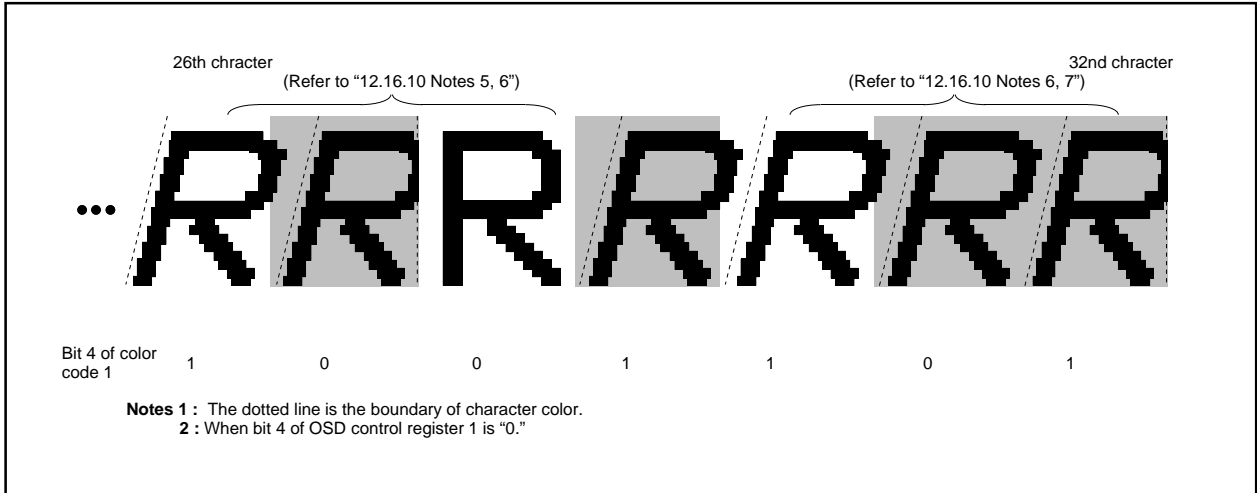


Figure 2.16.32 Example of attribute display (in CC mode)

**M306V5ME-XXXSP**  
**M306V5EESP**

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 the OSDSP 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).

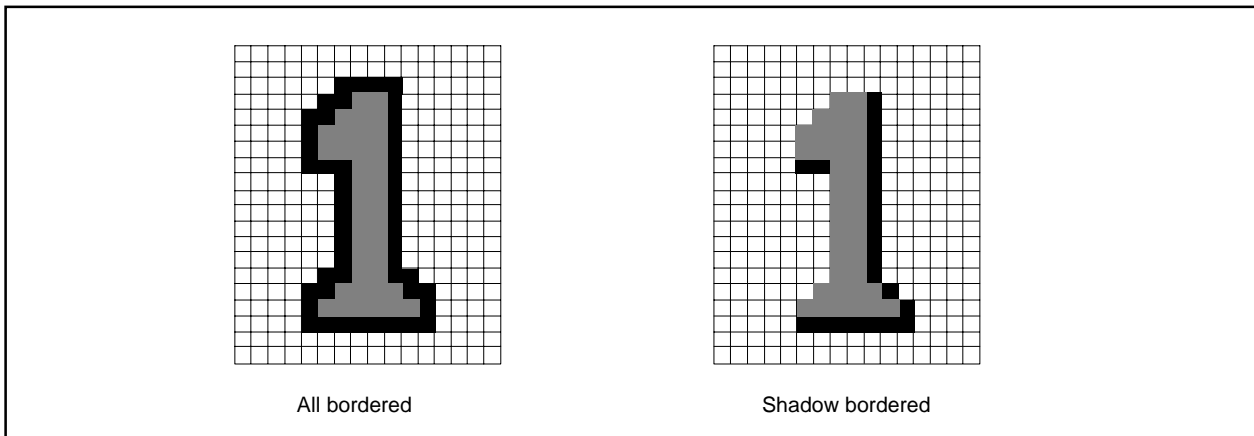
The OUT1 signal is used for border output. The border color is fixed at color palette 8 (black). The border color for each screen is specified by the border color register i.

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

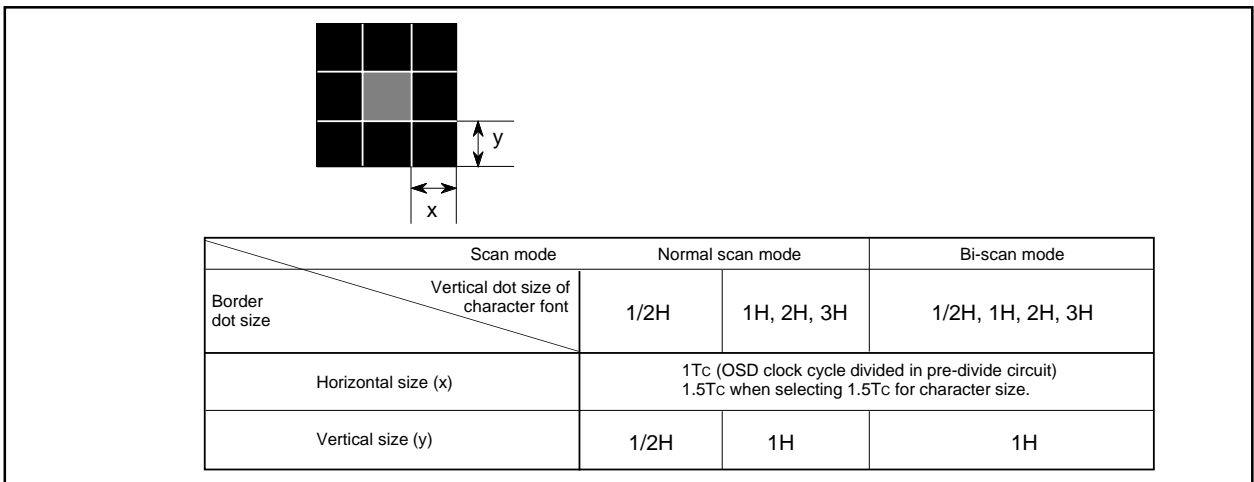
**Notes 1 :** The border dot area is the shaded area as shown in Figure 2.16.36.

**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 back ground, 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).



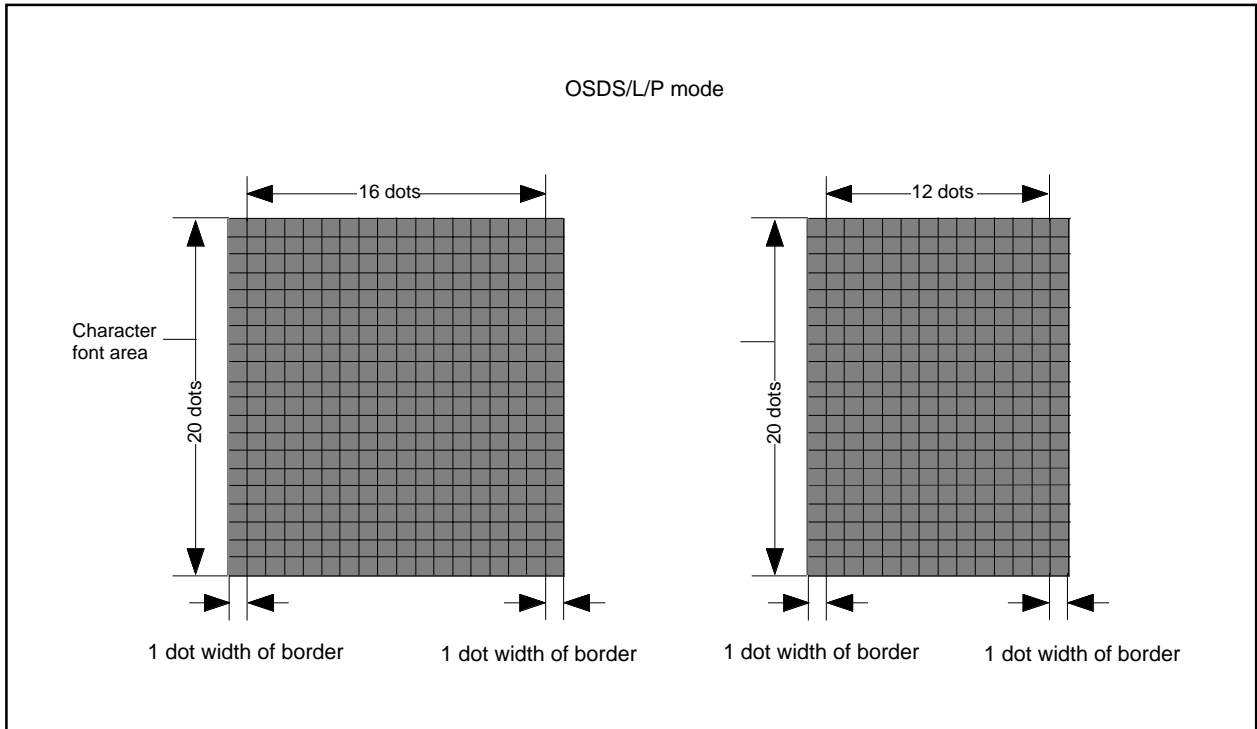
**Figure 2.16.34 Example of border display**



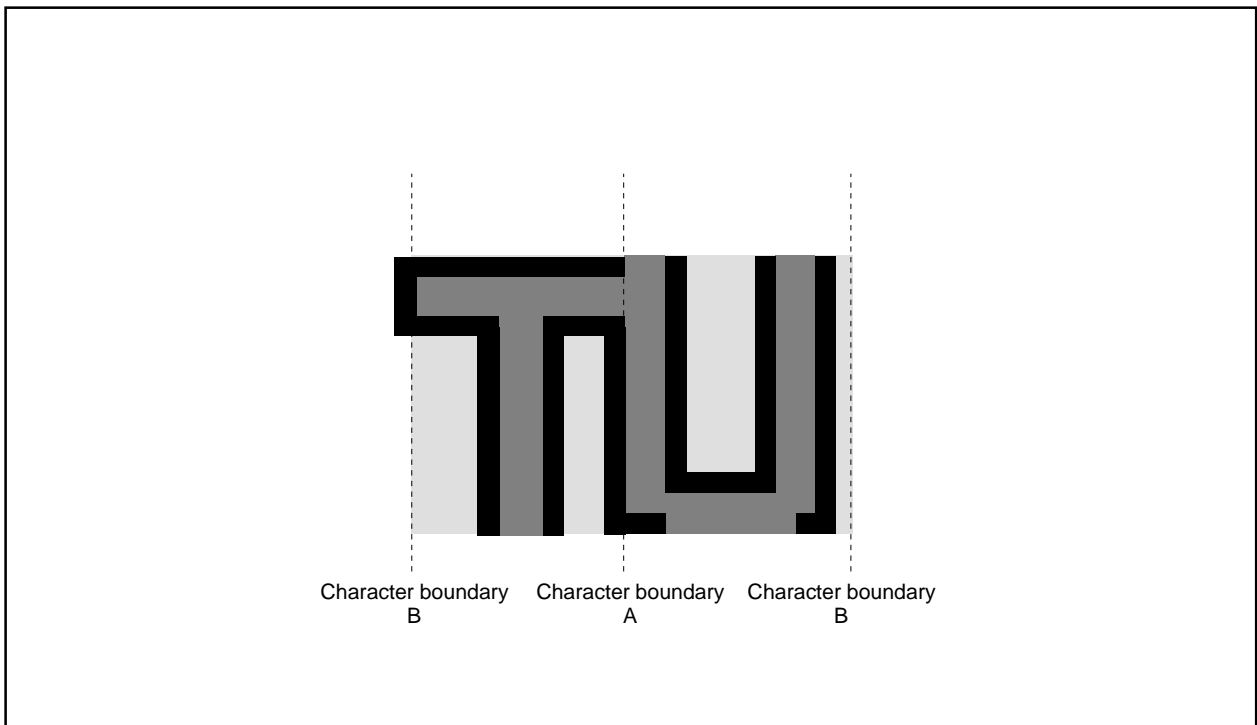
**Figure 2.16.35 Horizontal and vertical size of border**

**M306V5ME-XXXSP**  
**M306V5EESP**

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 “009<sub>16</sub>”
- 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).

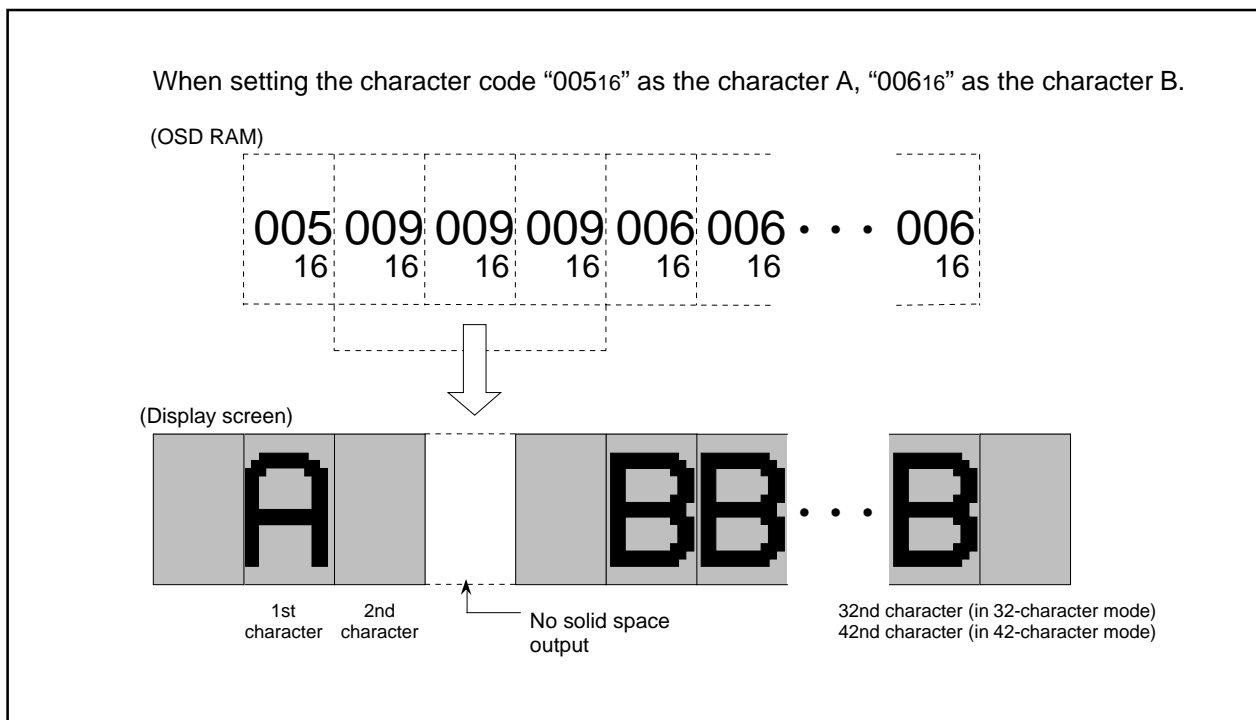
OUT1 or OUT2 output is selected by bit 3 of the OSD control register 2.

**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 “009<sub>16</sub>,” 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 Particular OSD Mode Block

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

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

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

Each font is selected by a character code. Figure 2.16.39 shows the display example of particular OSD mode block and Table 2.16.11 shows the corresponding between character codes and display fonts.

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



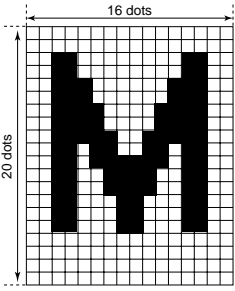
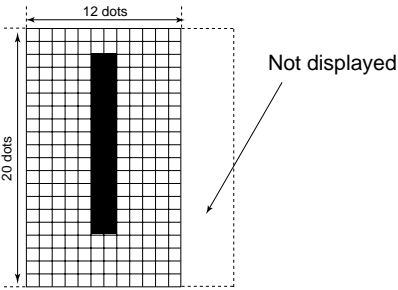
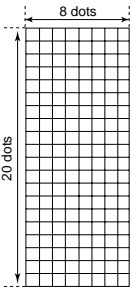
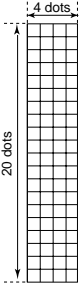
Figure 2.16.39 Display example of OSD mode block



**M306V5ME-XXXSP**  
**M306V5EESP**

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

**Table 2.16.11 Corresponding between character codes and display fonts**

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

**2.16.13 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.

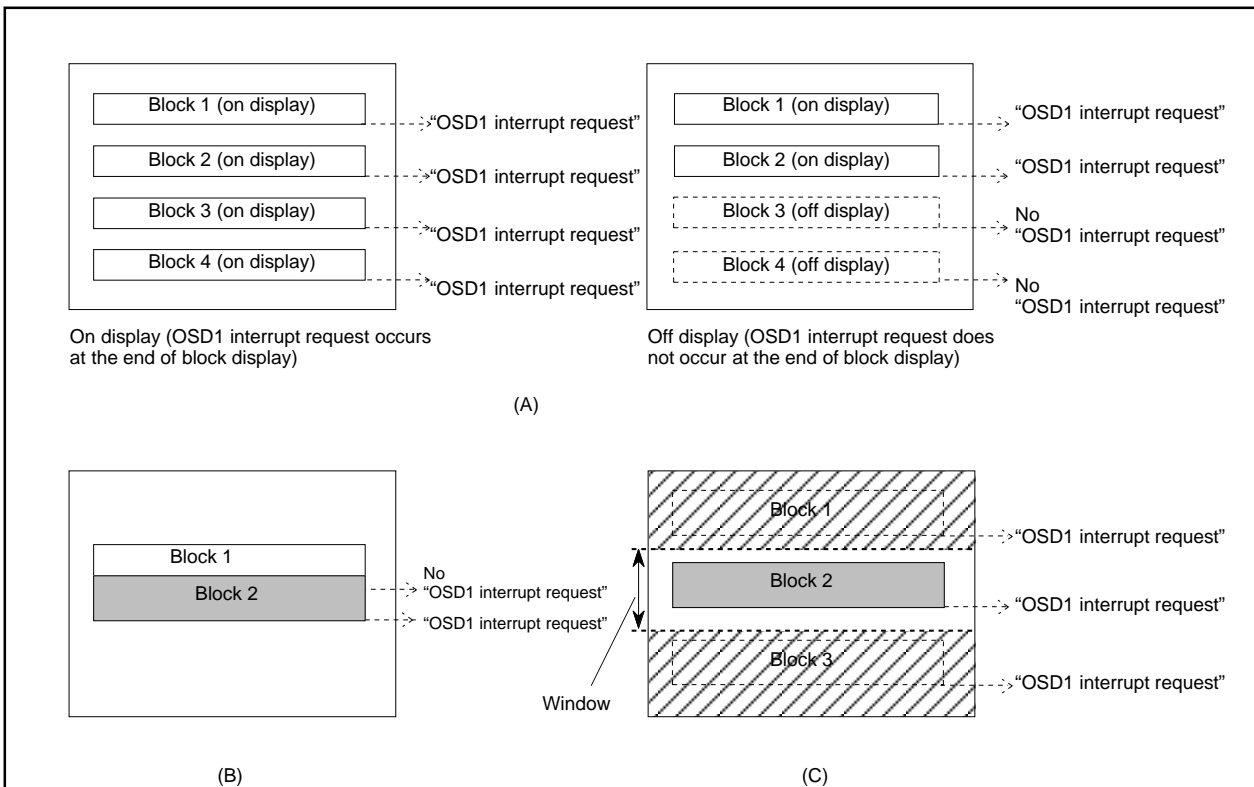
An OSD1 interrupt request occurs at the point at which display of each block has been completed. In other words, when a scanning line reaches the point of the display position (specified by the vertical position registers) of a certain block, the character display of that block starts, and an interrupt 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**

**2.16.14 SPRITE OSD Function**

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

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

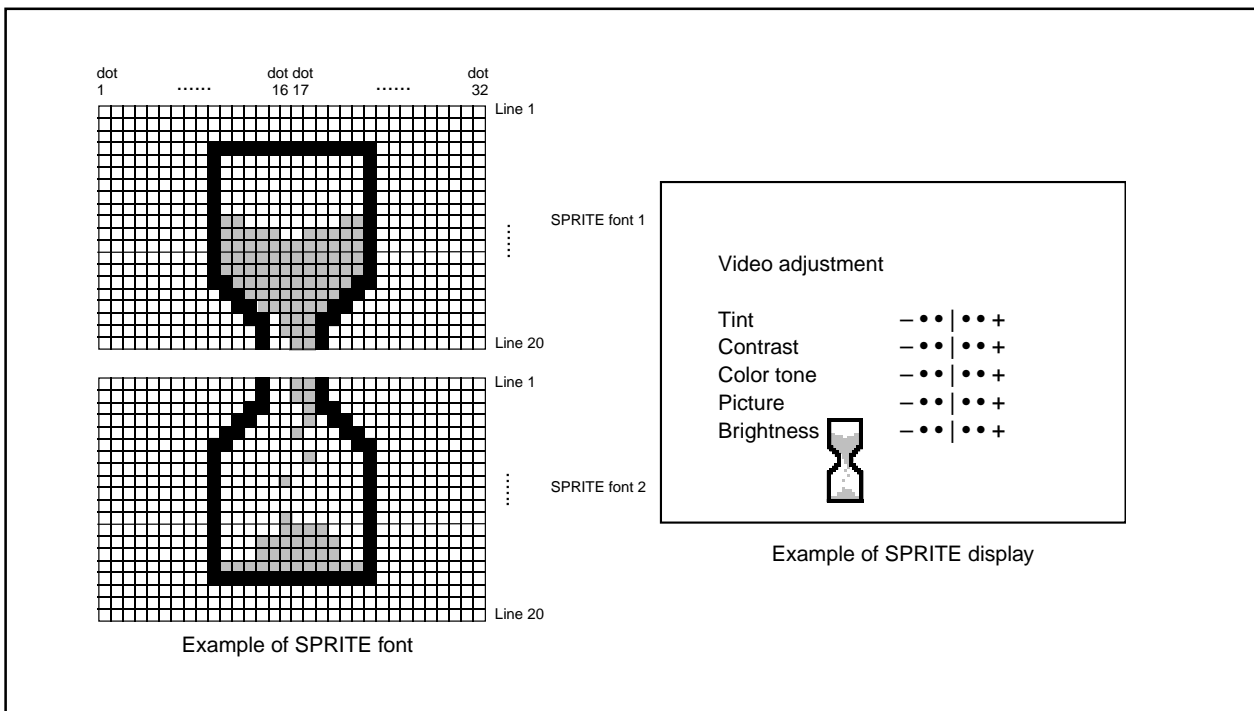
When SPRITE display overlaps with other OSD displays, SPRITE display is always given priority. However, the SPRITE display overlaps with the display which includes OUT2 output, OUT2 in the OSD is output without masking.

**Notes 1:** The SPRITE OSD function cannot output OUT2.

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

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

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



**Figure 2.16.41 SPRITE OSD display example**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

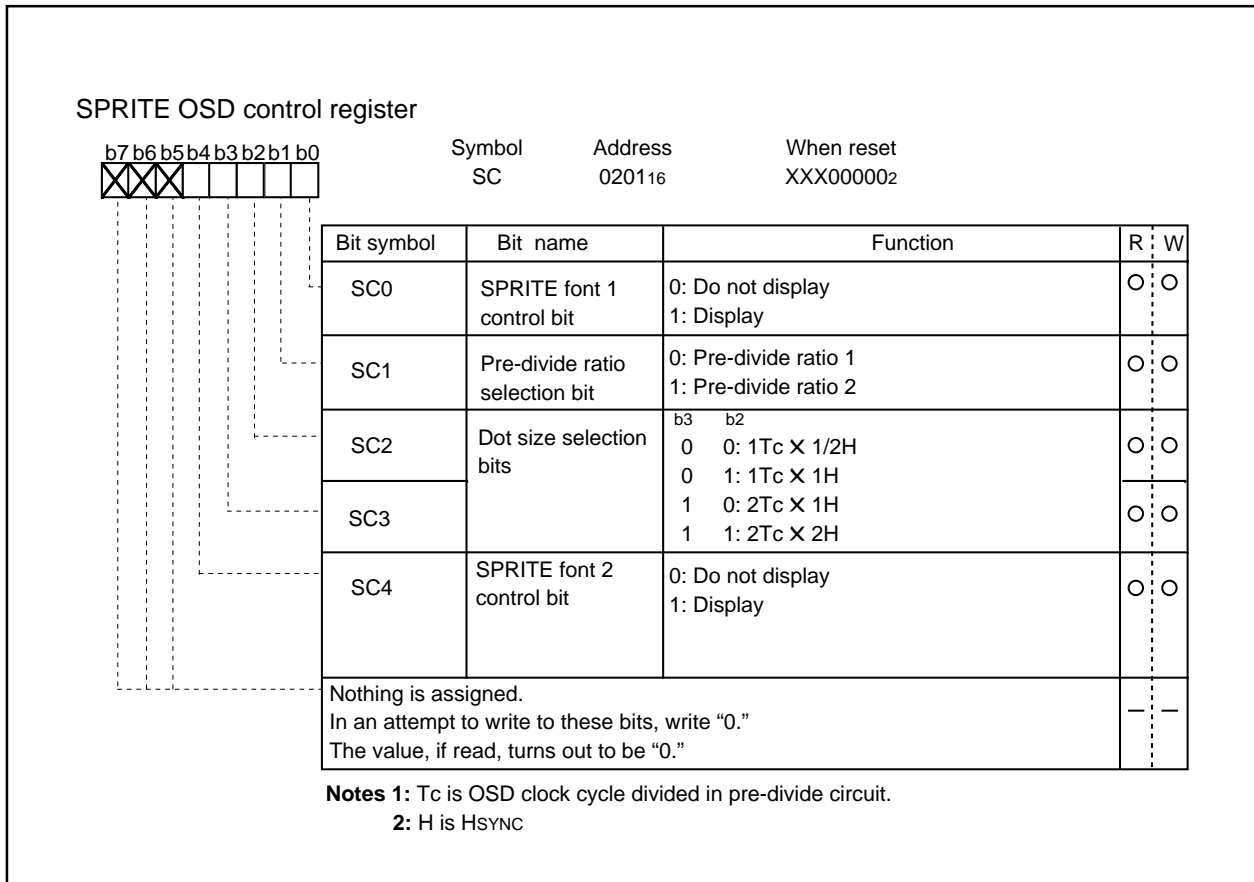
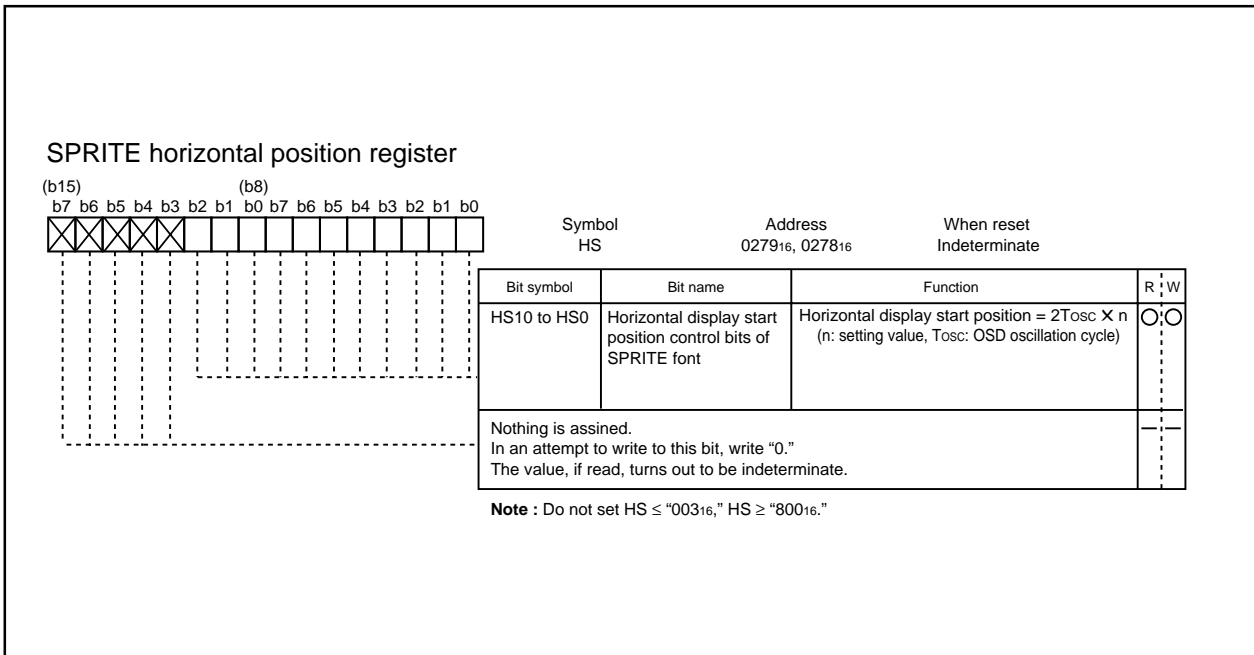


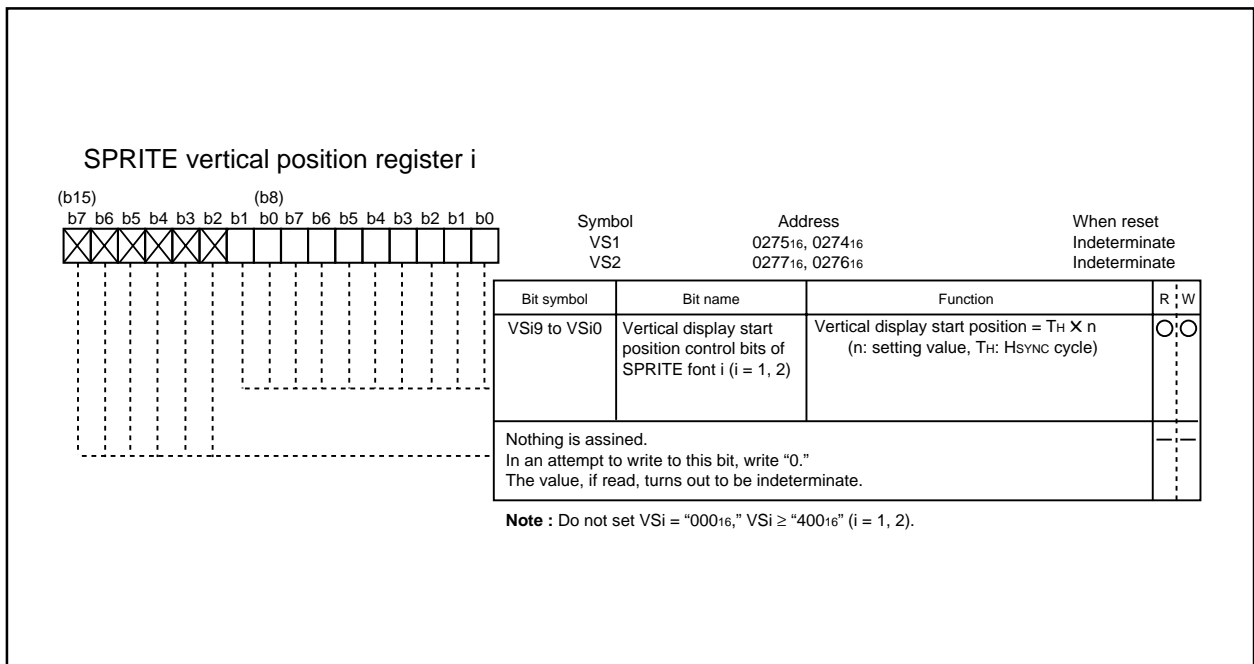
Figure 2.16.42 SPRITE OSD control register

**M306V5ME-XXXSP**  
**M306V5EESP**

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)**

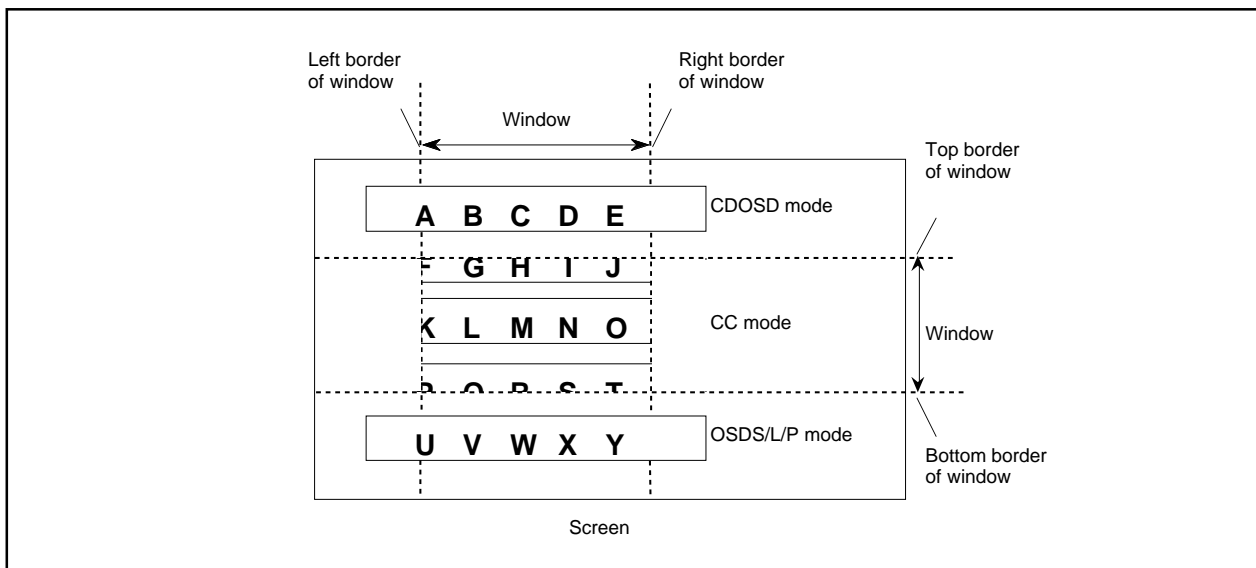
**2.16.15 Window Function**

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

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

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

- Notes 1:** Horizontal blank and horizontal window, as well as vertical blank and vertical window can not be used simultaneously.
- 2:** When the window function is ON by OSD control registers 1 and 2, the window function of OUT2 is valid in all display mode regardless of setting value of the OSD control register 3 (bits 5 to 7). For example, even when make the window function valid in only CC mode, the function of OUT2 is valid in OSDS/L/P and CDOSD modes.
- 3:** As for SPRITE display, the window function does not operate.



**Figure 2.16.45 Example of window function (When CC mode is valid)**

**2.16.16 Blank Function**

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

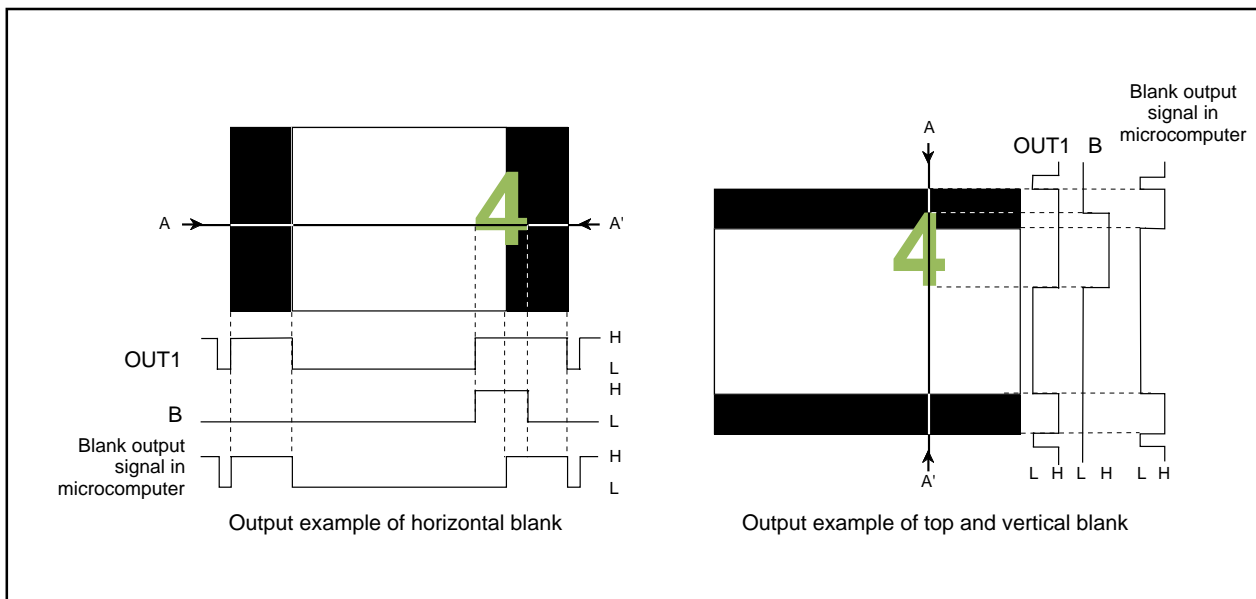
The ON/OFF for vertical blank function is performed by bit 5 of the OSD control register 1 and is used to select vertical window function or vertical blank function by bit 6 of the OSD control register 2. Accordingly, the vertical blank function cannot be used simultaneously with the vertical window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

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

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

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

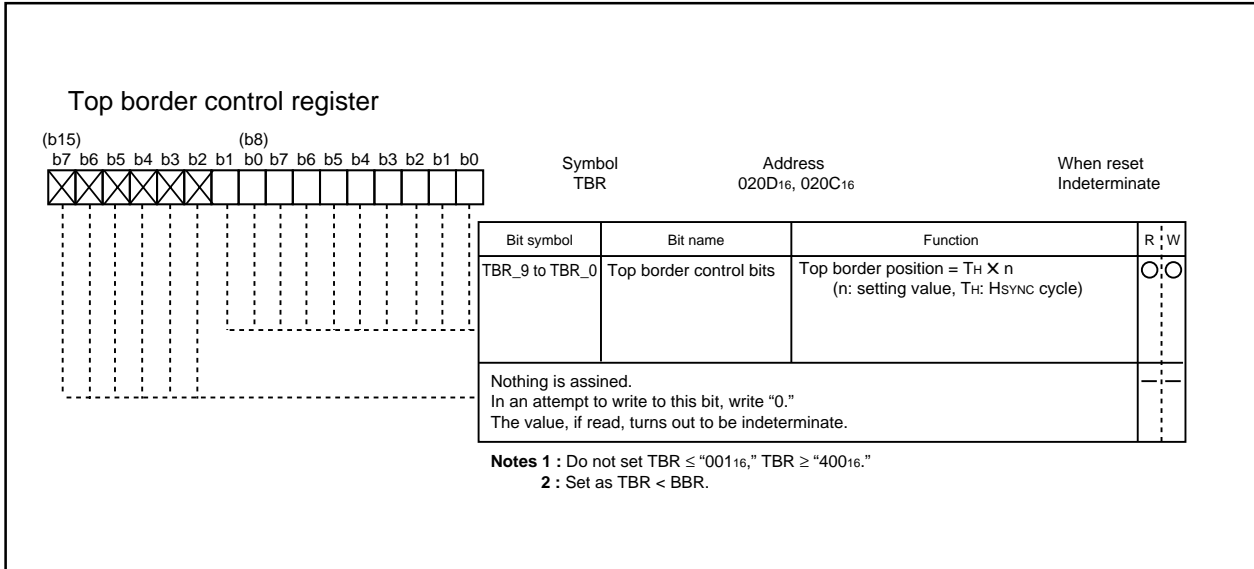
- Notes 1.** Horizontal blank and horizontal window, as well as vertical blank and vertical window can not be used simultaneously.
- 2.** When using the window function, be sure to set “1” to bit 0 of OSD control register 1.



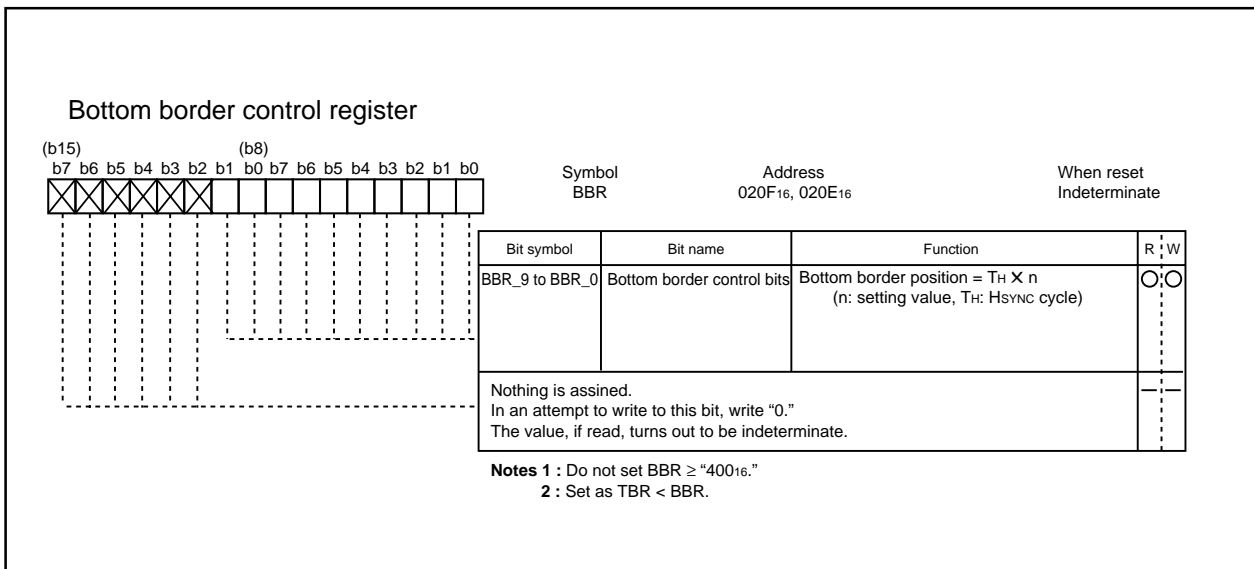
**Figure 2.16.46 Blank output example (when OSD output is B + OUT1)**

**M306V5ME-XXXSP**  
**M306V5EESP**

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**



**M306V5ME-XXXSP**  
**M306V5EESP**

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

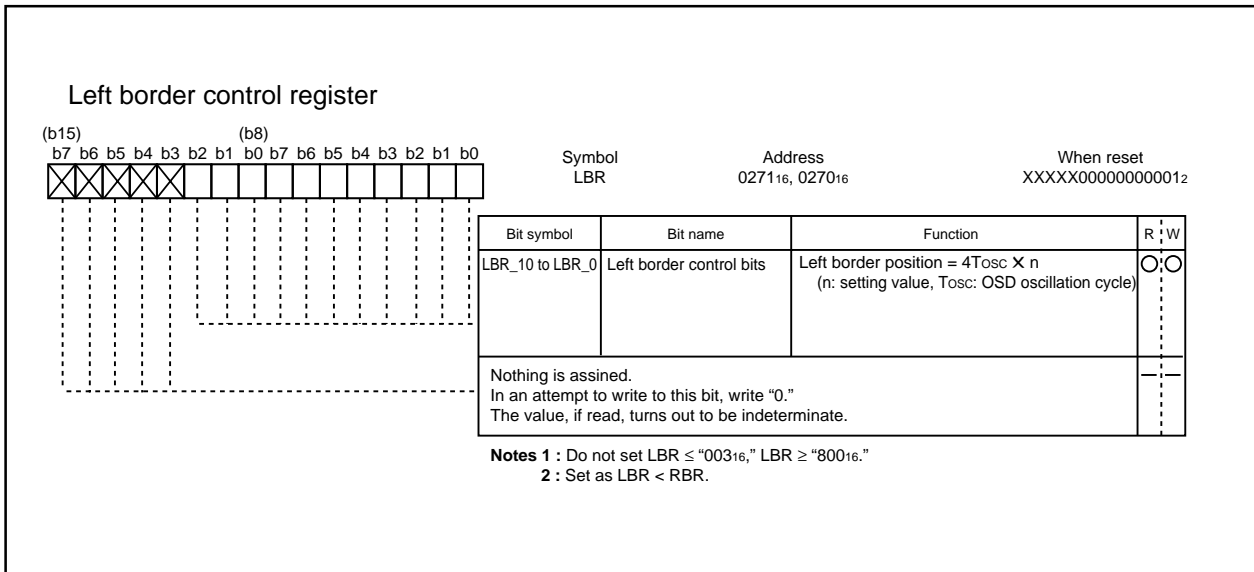


Figure 2.16.49 Left border control register

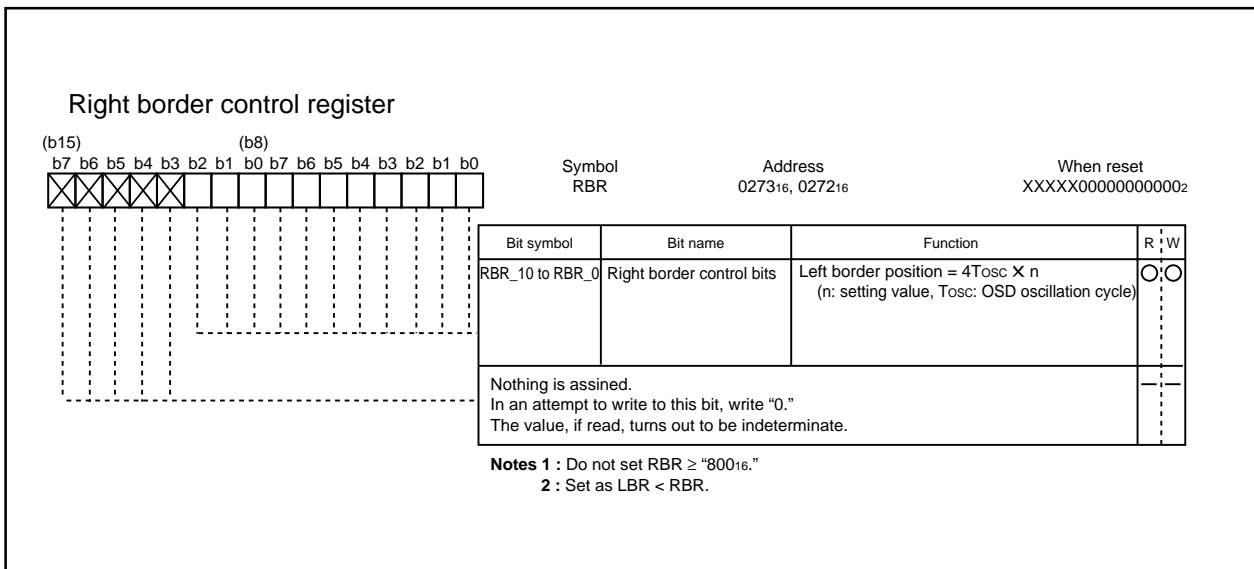


Figure 2.16.50 Bottom border control register

**M306V5ME-XXXSP**  
**M306V5EESP**

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

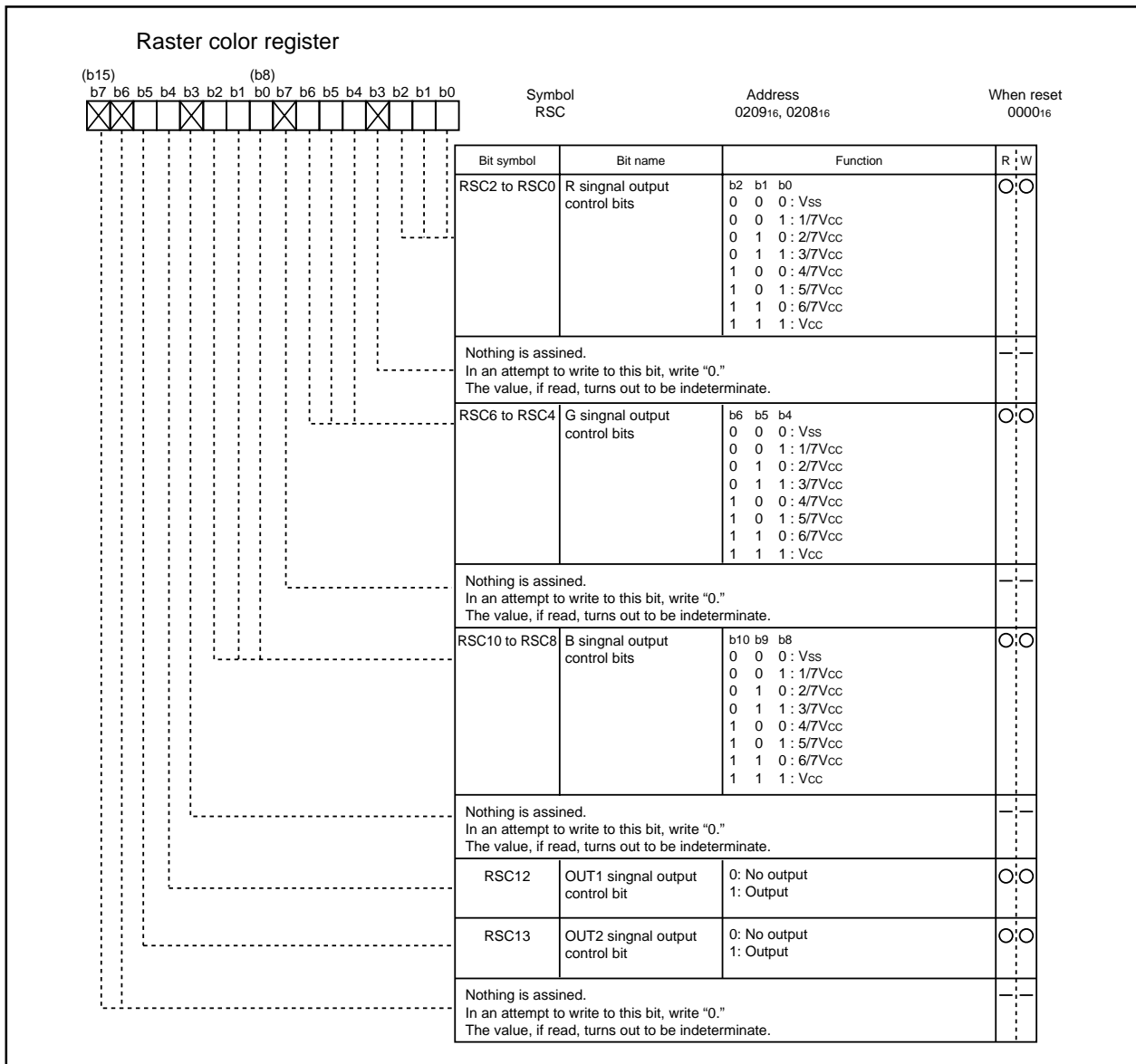
**2.16.17 Raster Coloring Function**

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**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

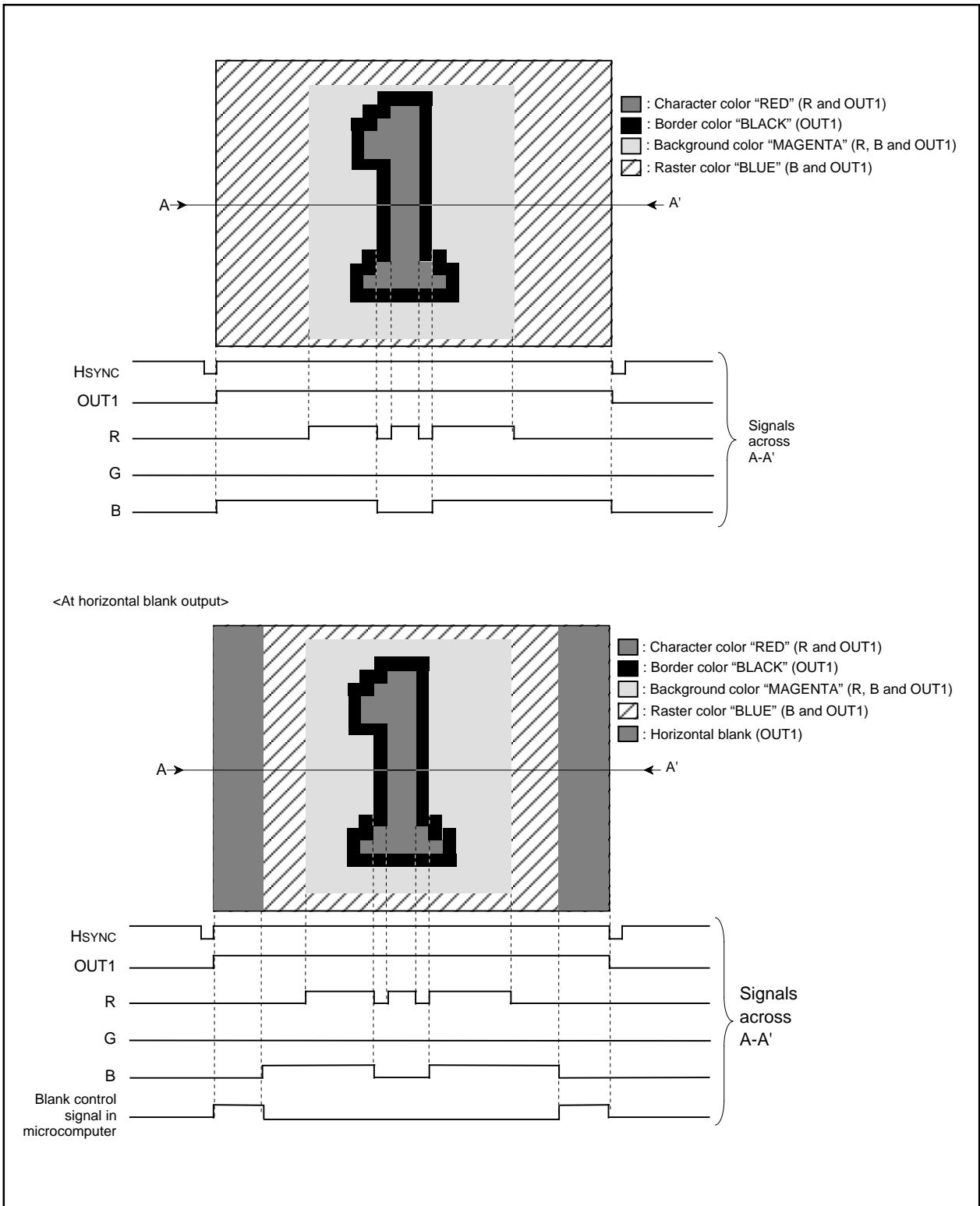


Figure 2.16.52 Example of raster coloring

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**2.16.18 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

**2.16.19 R, G, B Signal Output Control**

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

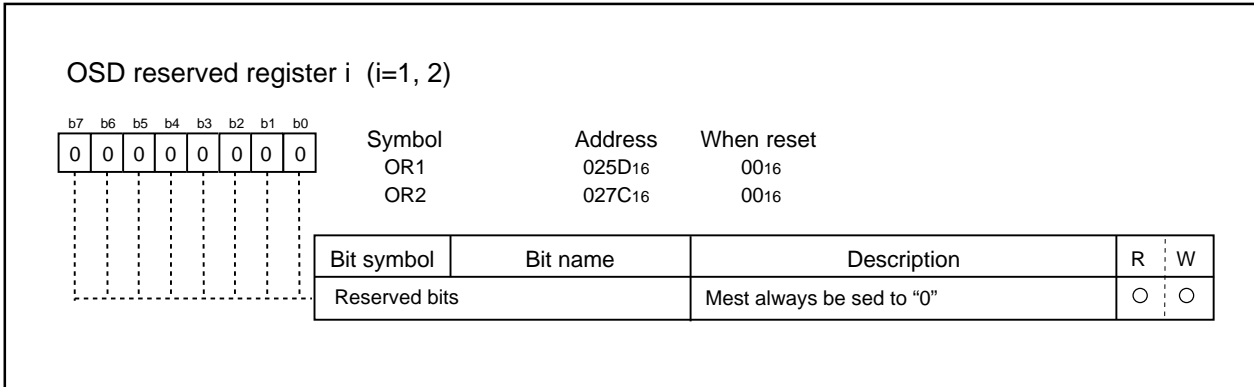
**Table 2.16.13 R, G, B signal output control**

Bit 2 of OSD control register 2	Form of R, G, B signal output
0	Each R, G, B pin outputs 2 values (digital output).
1	Each R, G, B pin outputs 8 values (analog output).

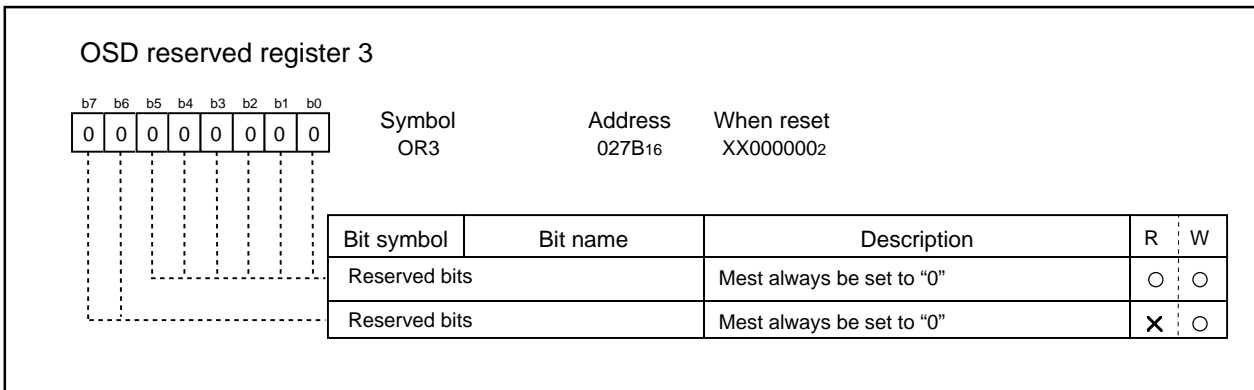
**M306V5ME-XXXSP**  
**M306V5EESP**

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

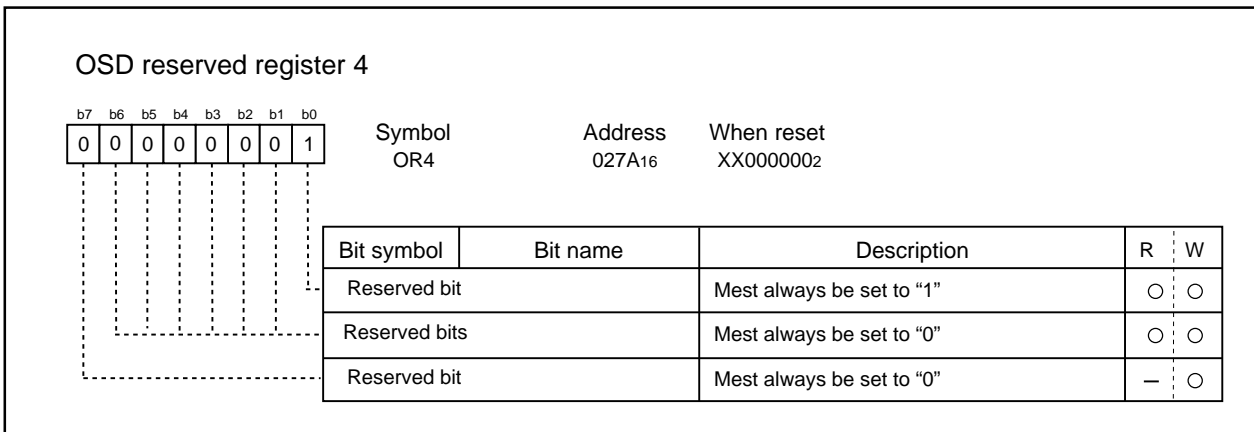
**2.16.20 OSD Reserved Register**



**Figure 2.16.53 OSD reserved register i (i=1, 2)**



**Figure 2.16.54 OSD reserved register 3**



**Figure 2.16.55 OSD reserved register 4**

## 2.17 Programmable I/O Ports

There are 46 programmable I/O ports: P00–P07, P20–P27, P30–P37, P40–P43, P50, P52, P53, P62, P63, P67, P70–P72, P74, P76, P82, P90, P93, P94, P100 and P101. 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 2.17.3 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 2.17.5 to 2.17.12 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 2.17.13 to 2.17.20 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.

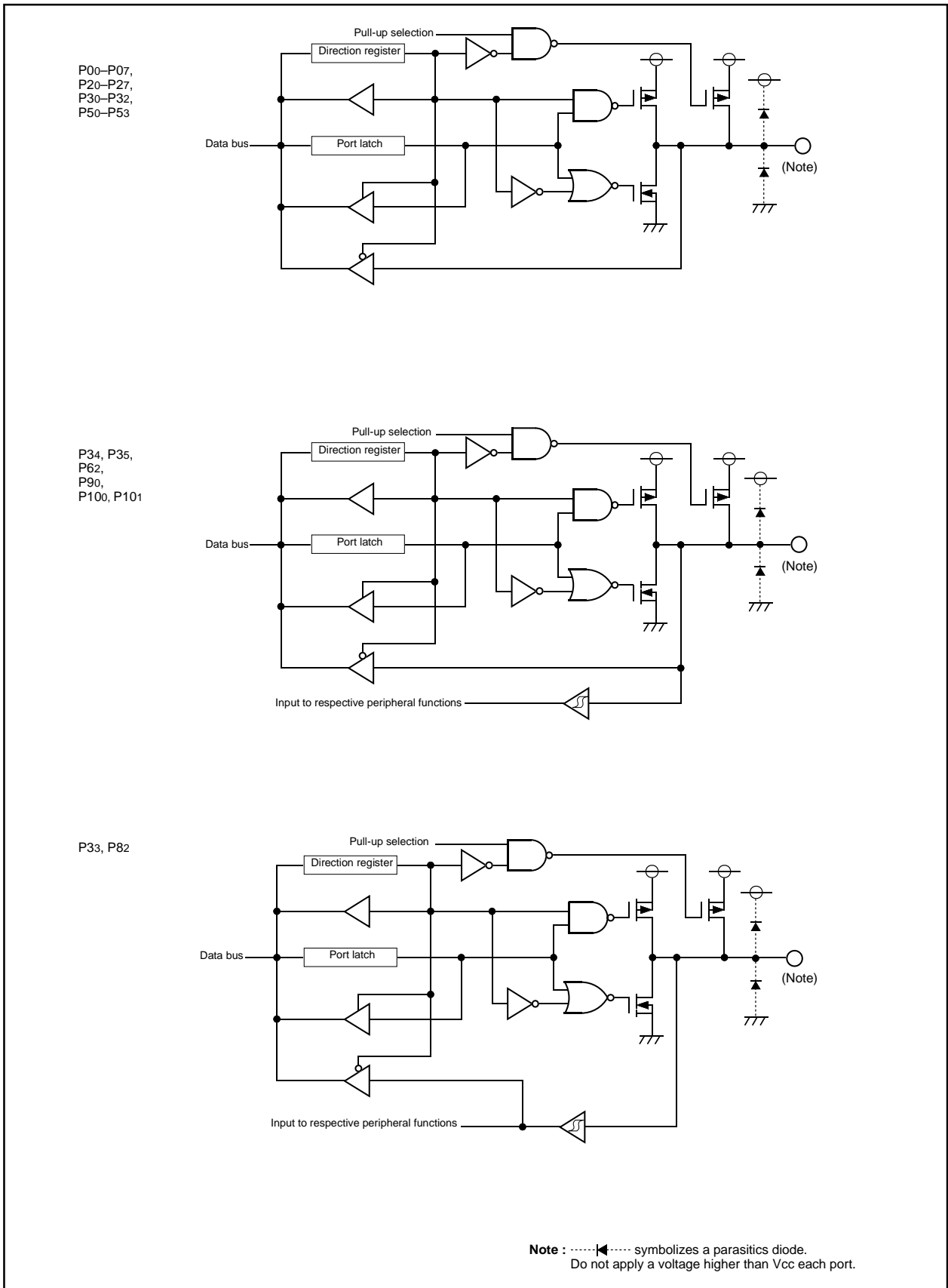
### 2.17.3 Pull-up Control Registers

Figures 2.17.24 to 2.17.26 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.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

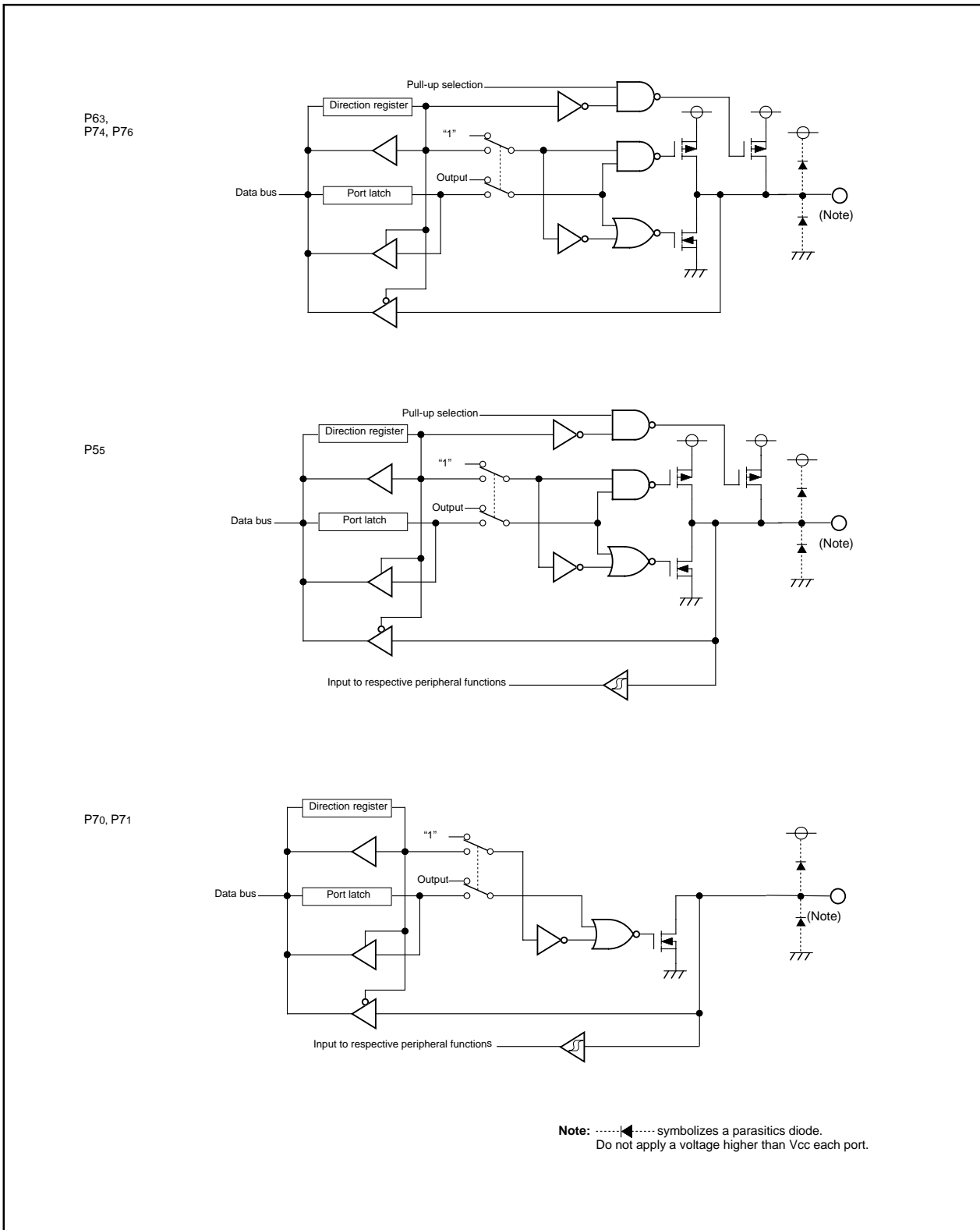


**Figure 2.17.1 Programmable I/O ports (1)**



**M306V5ME-XXXSP**  
**M306V5EESP**

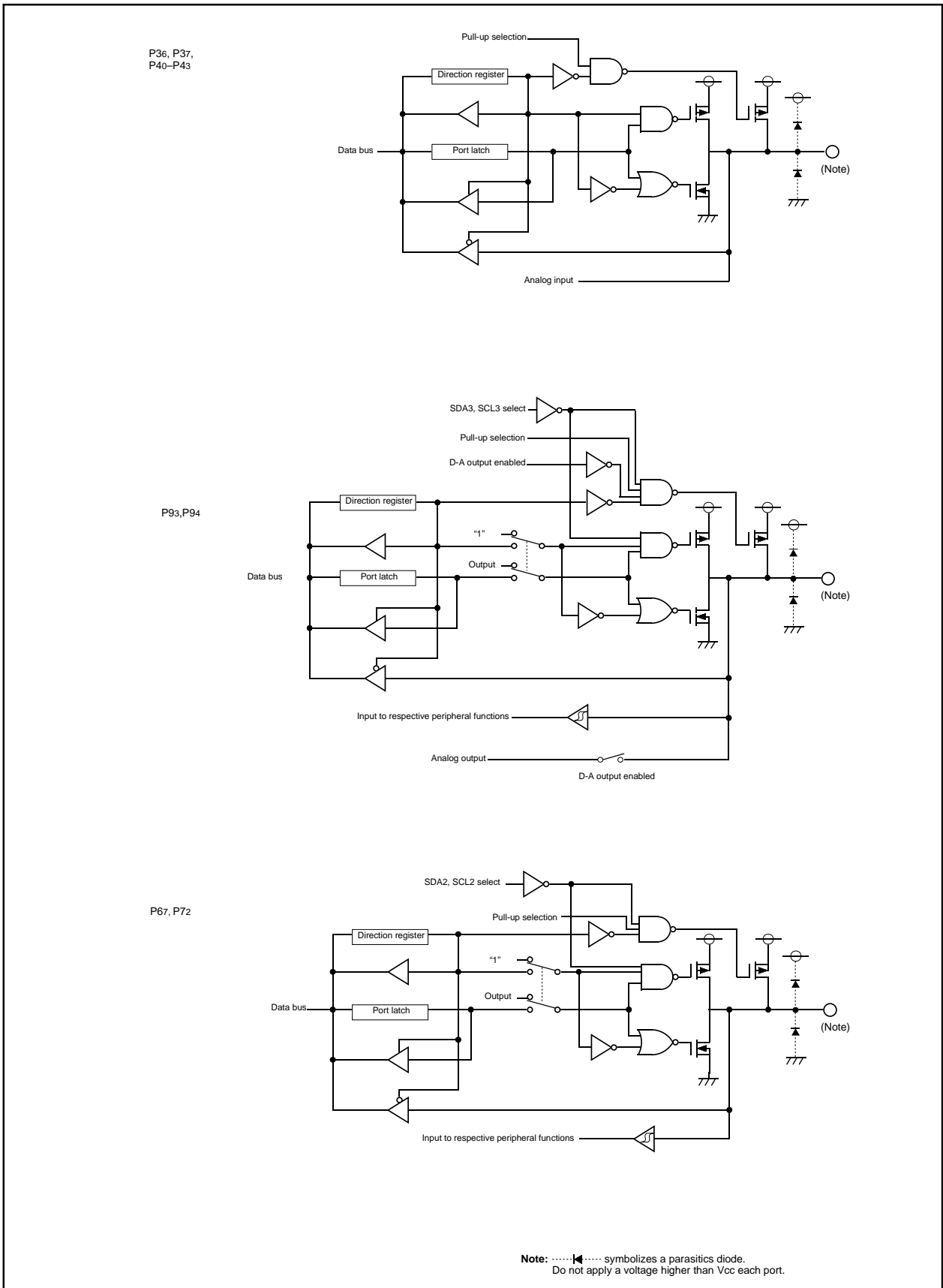
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER



**Figure 2.17.2 Programmable I/O ports (2)**

**M306V5ME-XXXSP**  
**M306V5EESP**

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



**Figure 2.17.3 Programmable I/O ports (3)**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

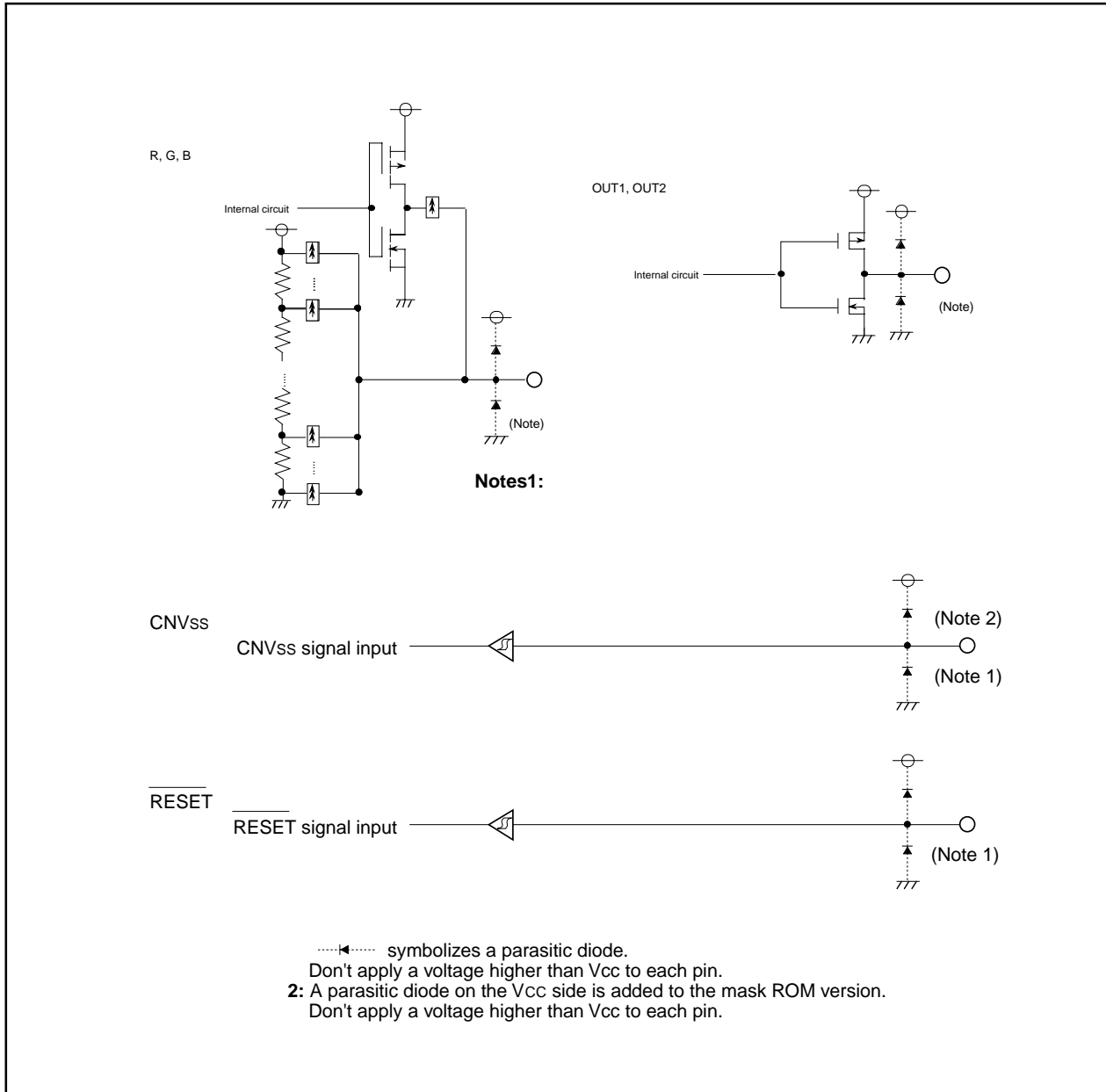
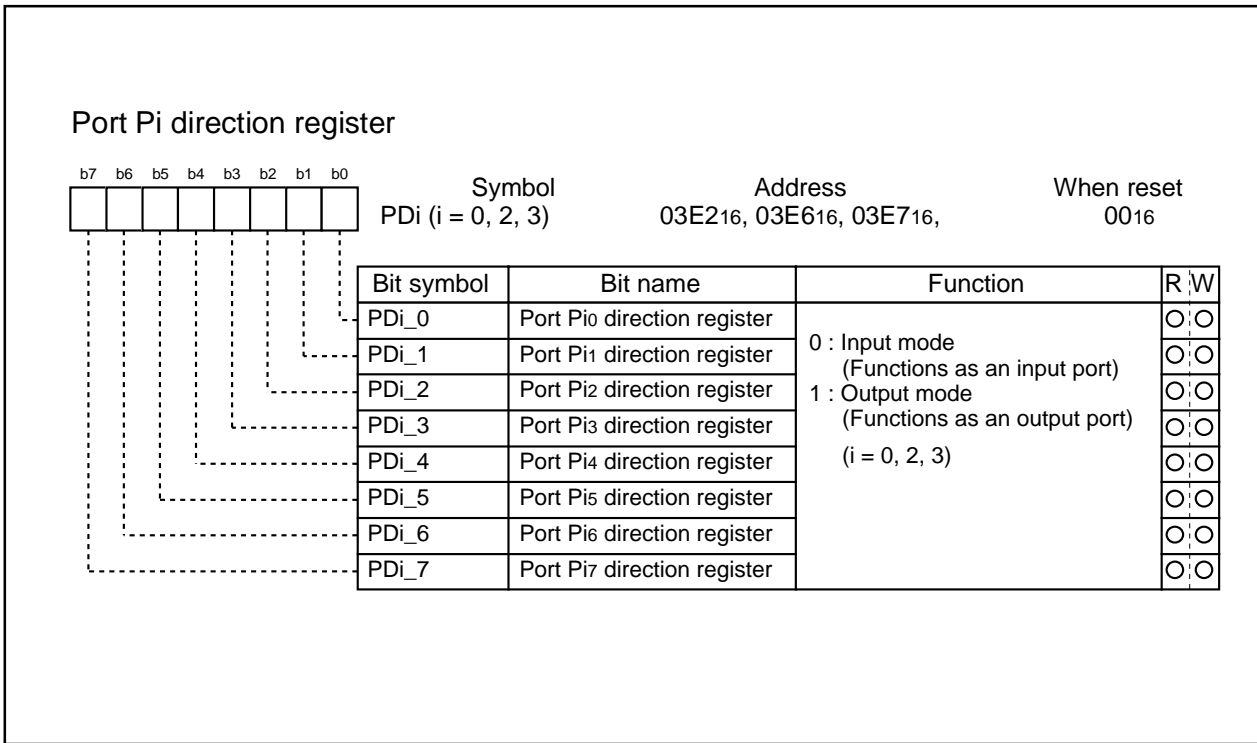


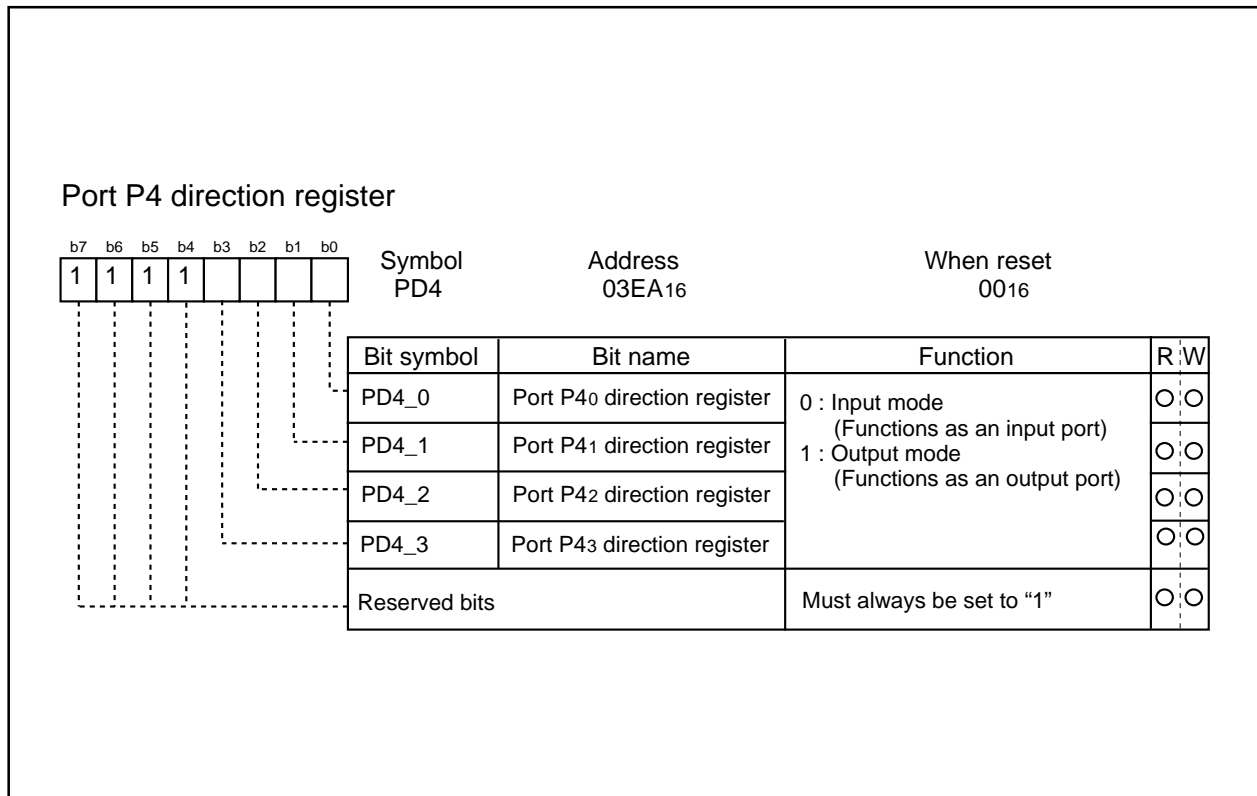
Figure 2.17.4 I/O pins

**M306V5ME-XXXSP**  
**M306V5EESP**

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



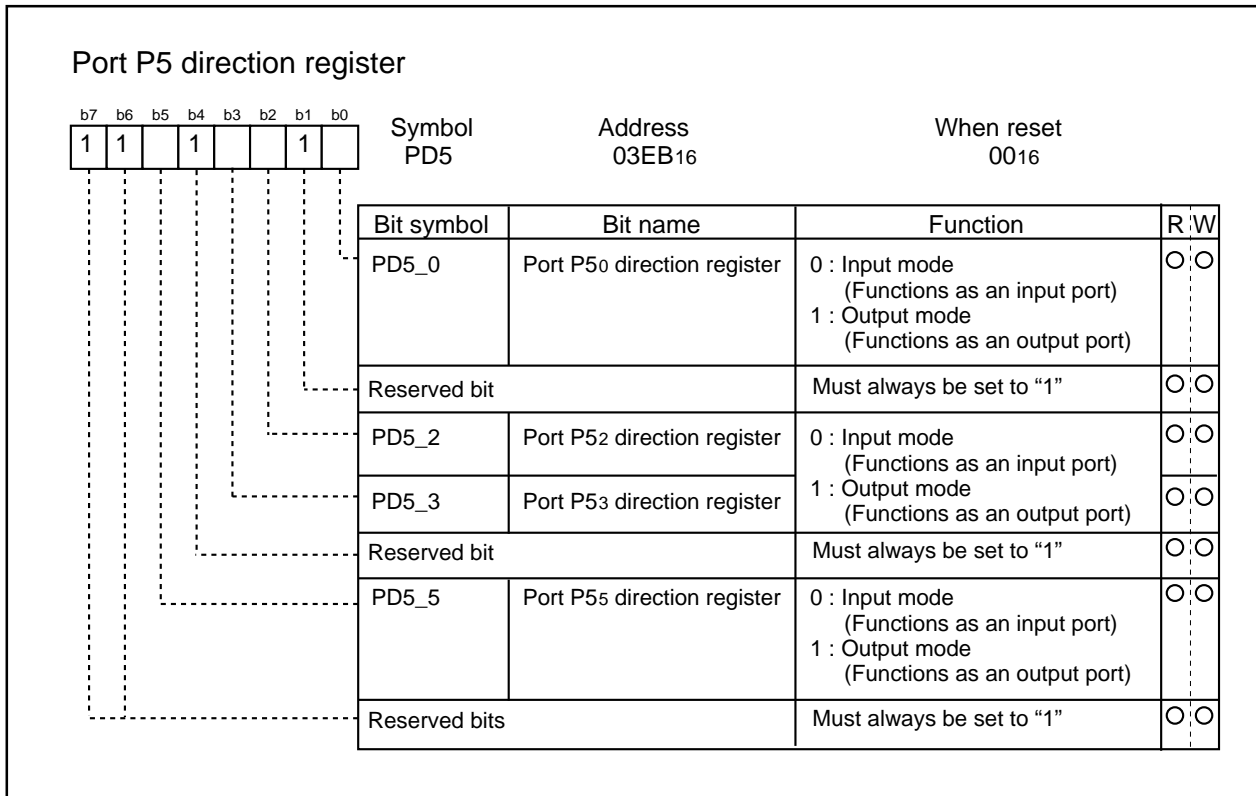
**Figure 2.17.5 Port Pi direction register (i = 0, 2, 3)**



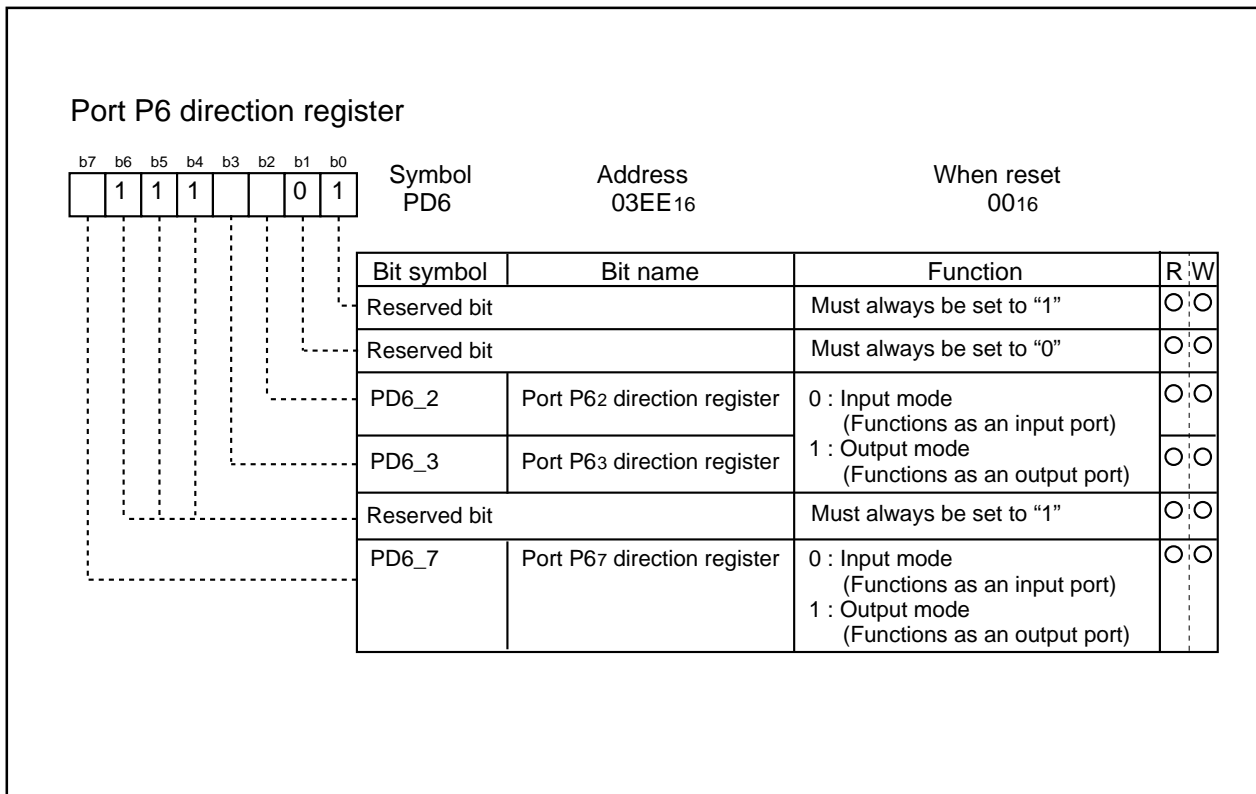
**Figure 2.17.6 Port P4 direction register**

**M306V5ME-XXXSP**  
**M306V5EESP**

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



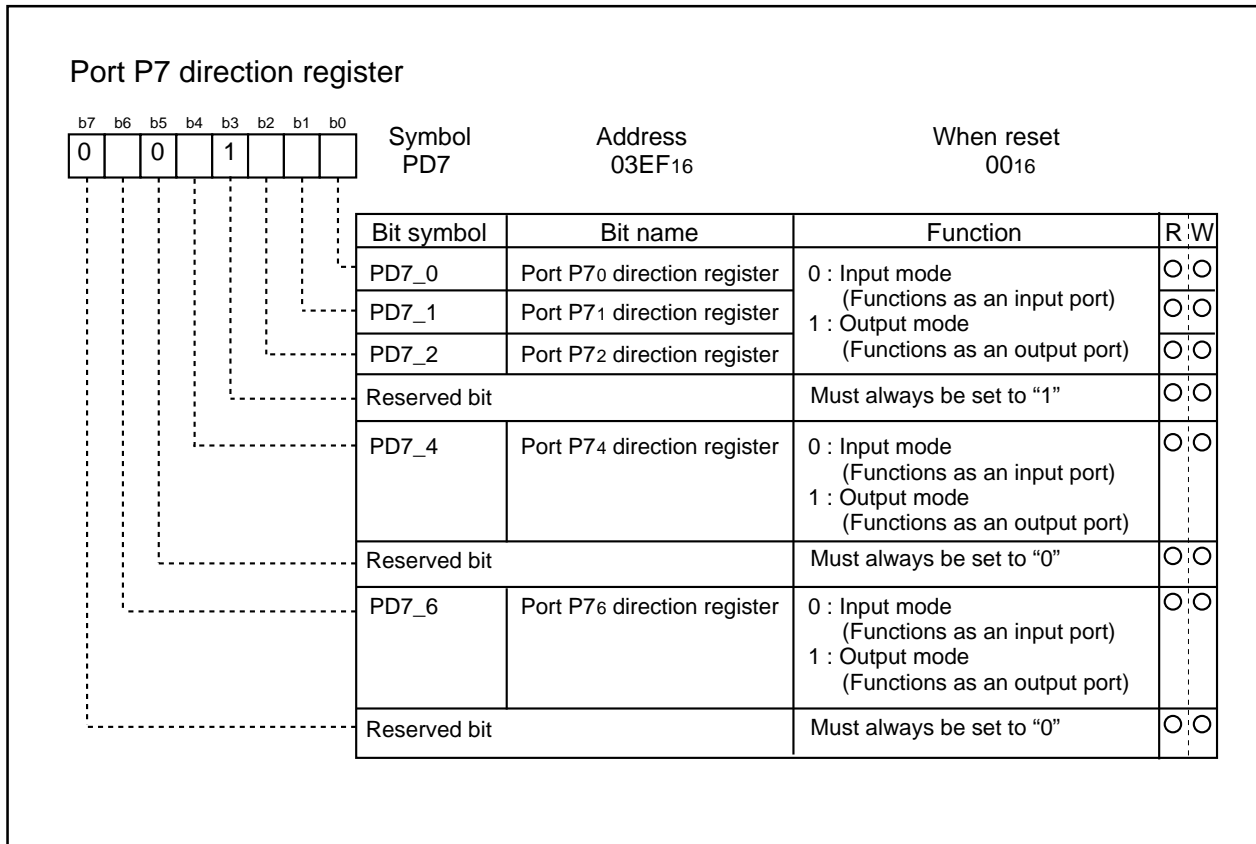
**Figure 2.17.7 Port P5 direction register**



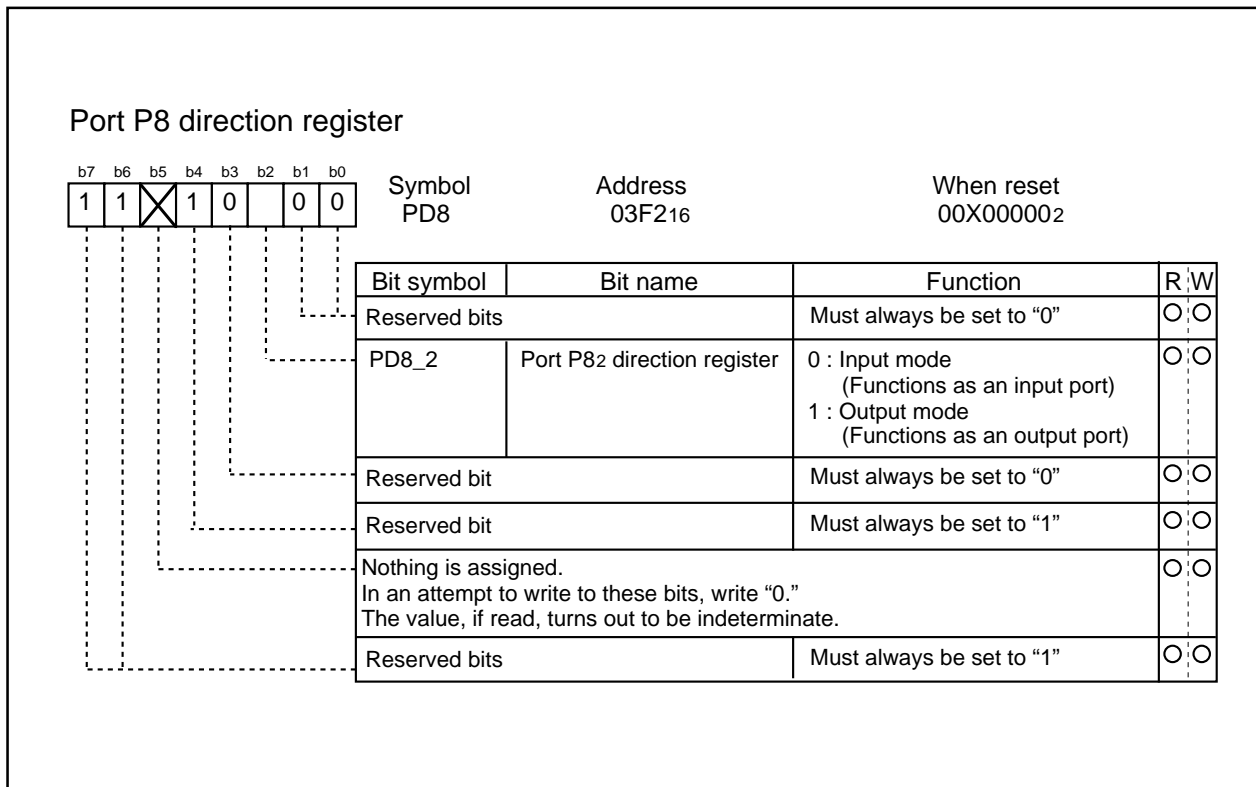
**Figure 2.17.8 Port P6 direction register**

**M306V5ME-XXXSP**  
**M306V5EESP**

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



**Figure 2.17.9 Port P7 direction register**



**Figure 2.17.10 Port P8 direction register**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

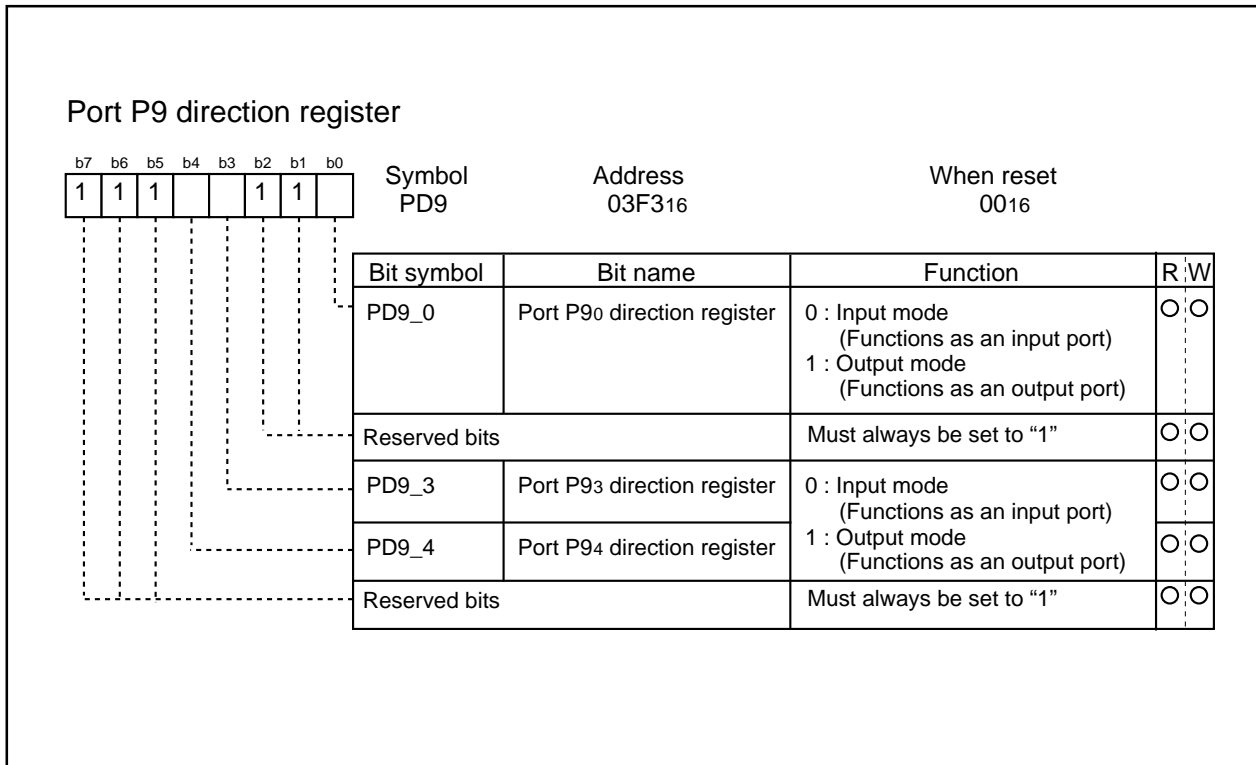


Figure 2.17.11 Port P9 direction register

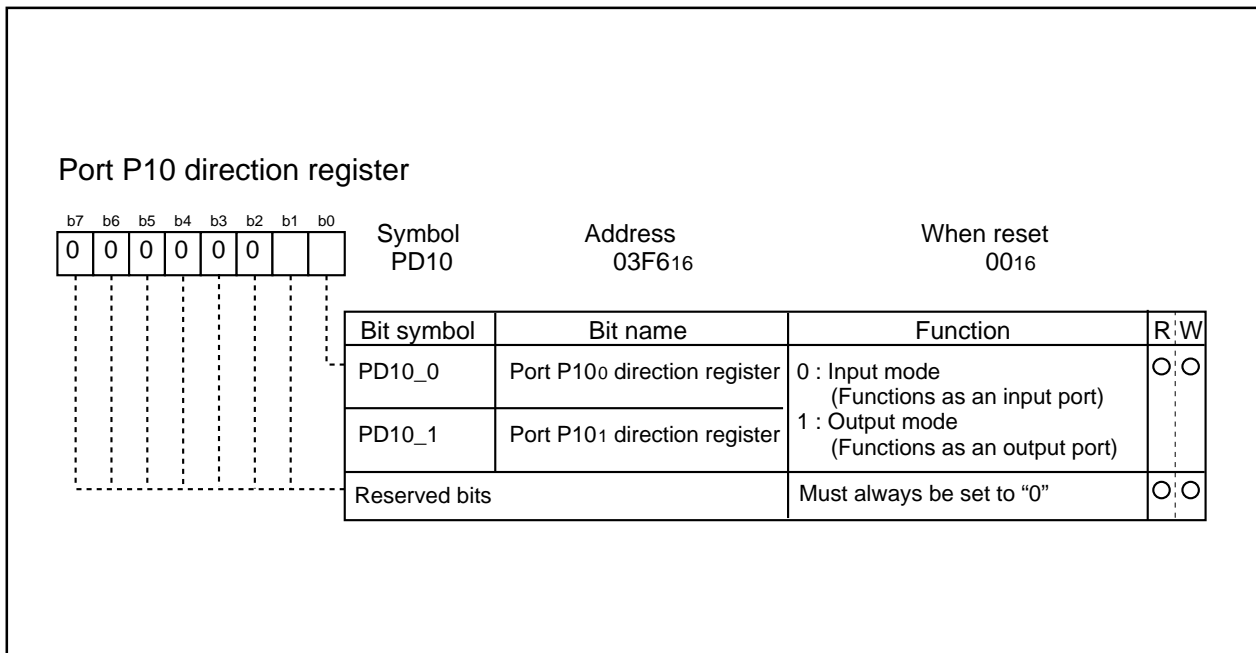


Figure 2.17.12 Port P10 direction register

**M306V5ME-XXXSP**  
**M306V5EESP**

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

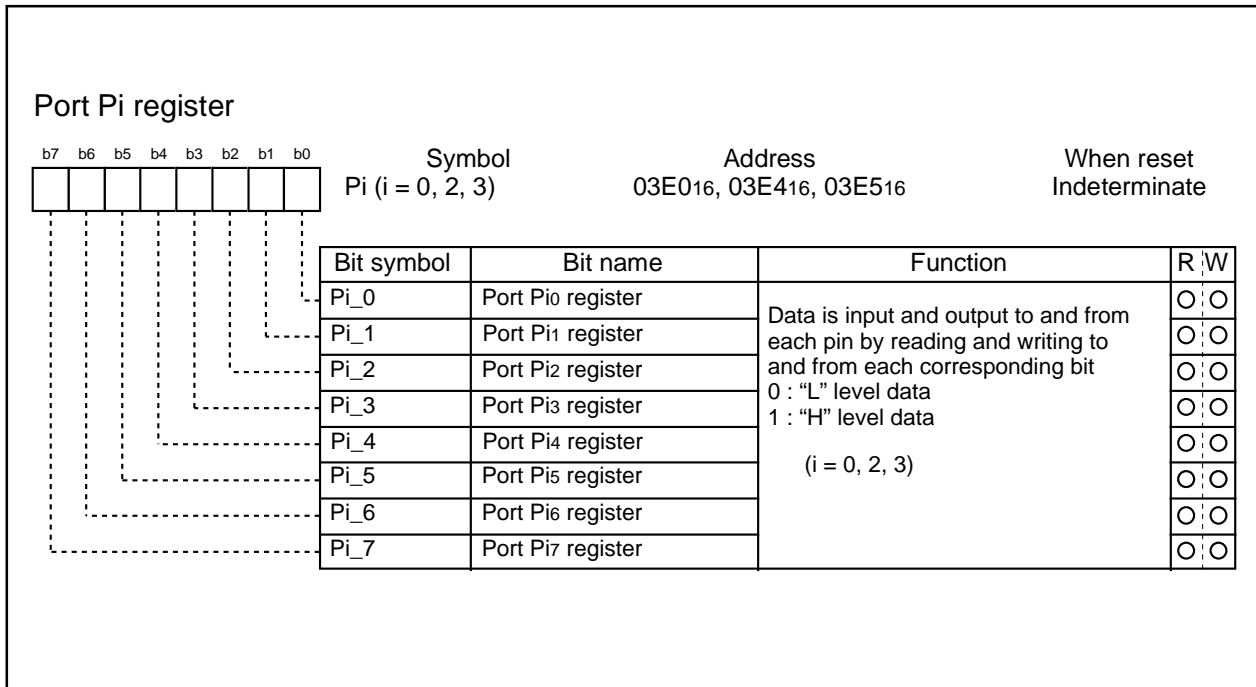


Figure 2.17.13 Port Pi register (i = 0, 2, 3)

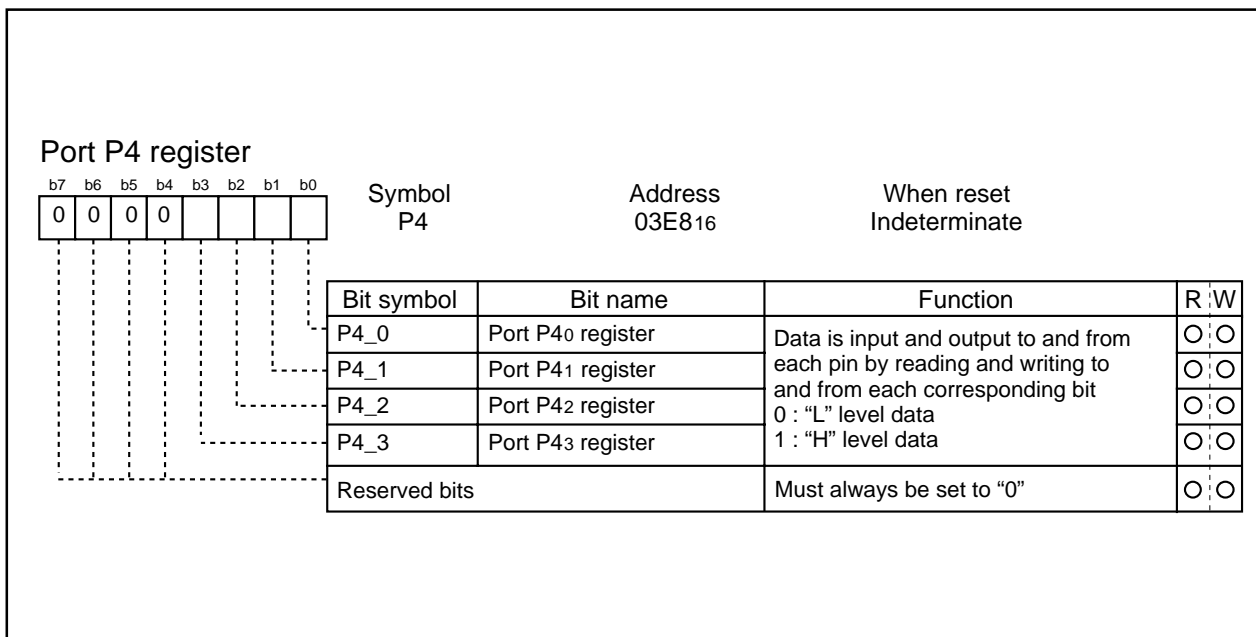
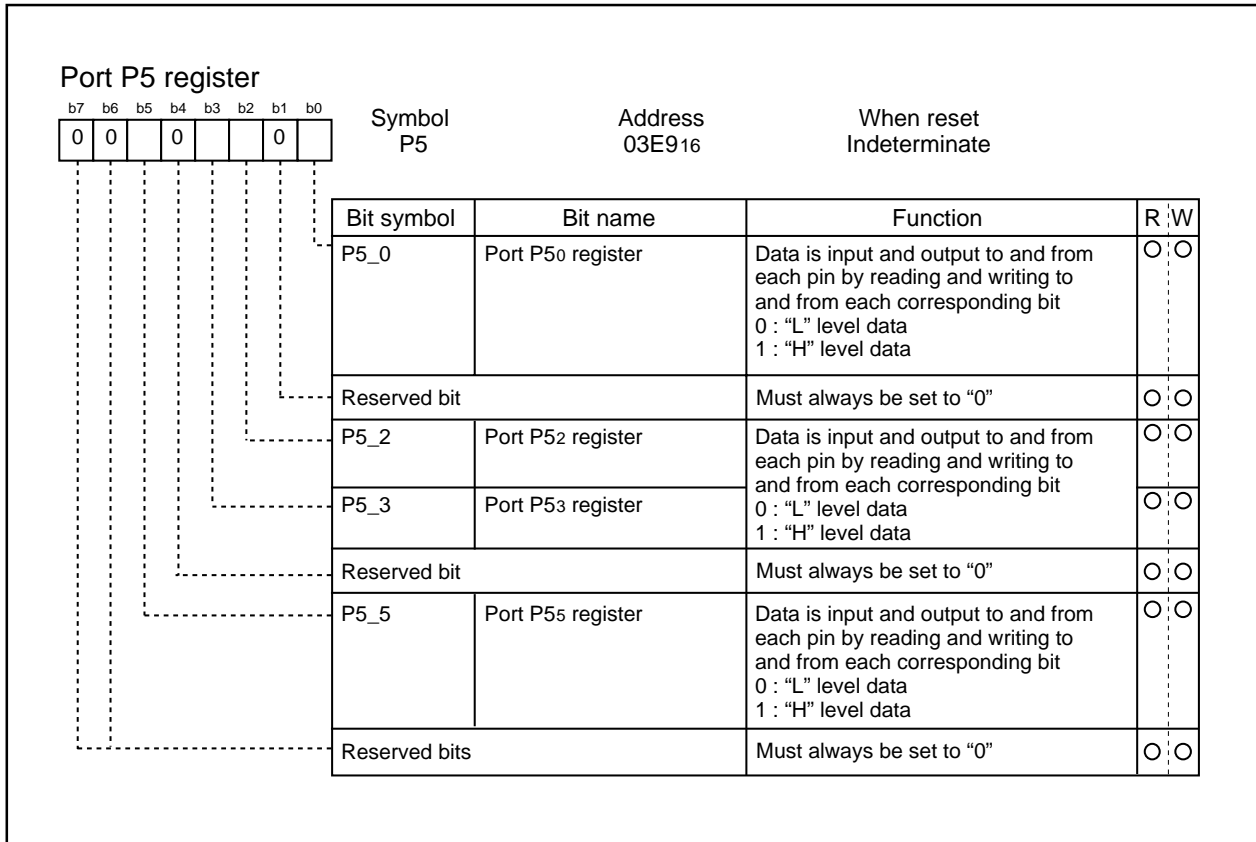


Figure 2.17.14 Port P4 register

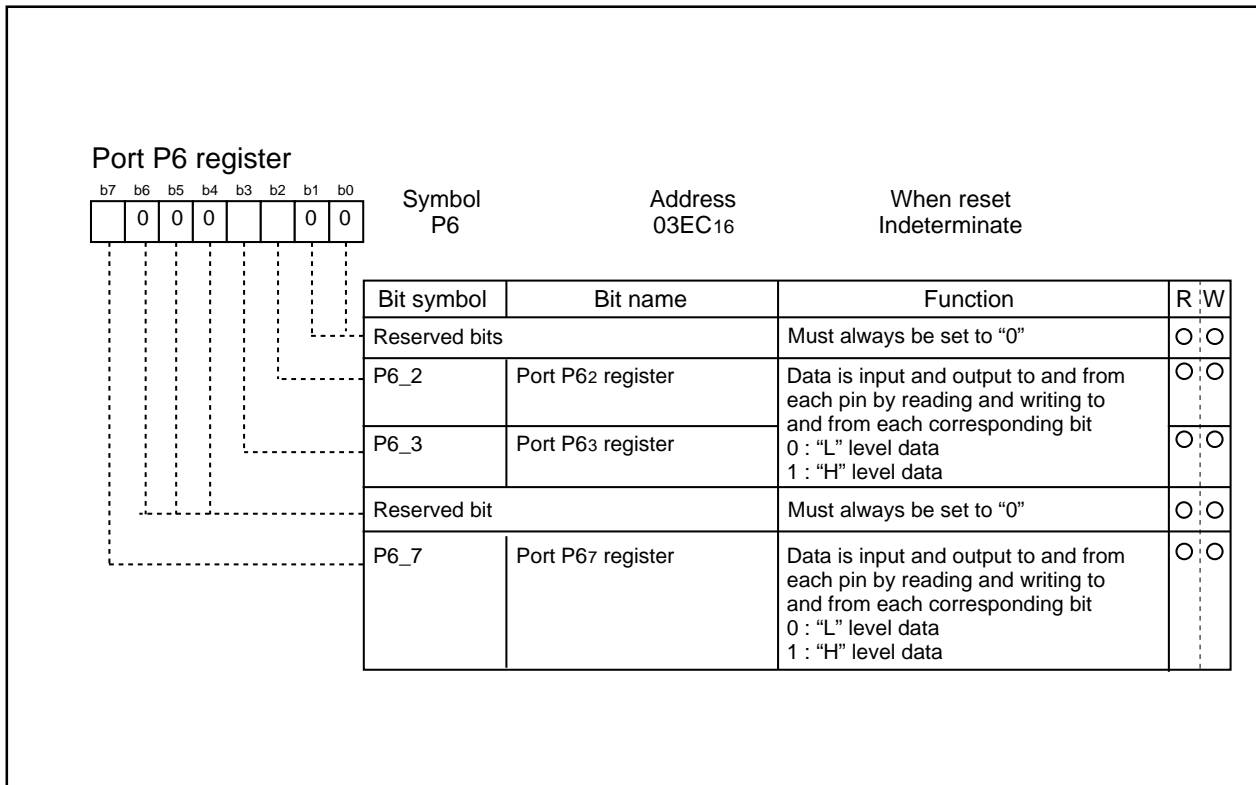


**M306V5ME-XXXSP**  
**M306V5EESP**

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



**Figure 2.17.15 Port P5 register**



**Figure 2.17.16 Port P6 register**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

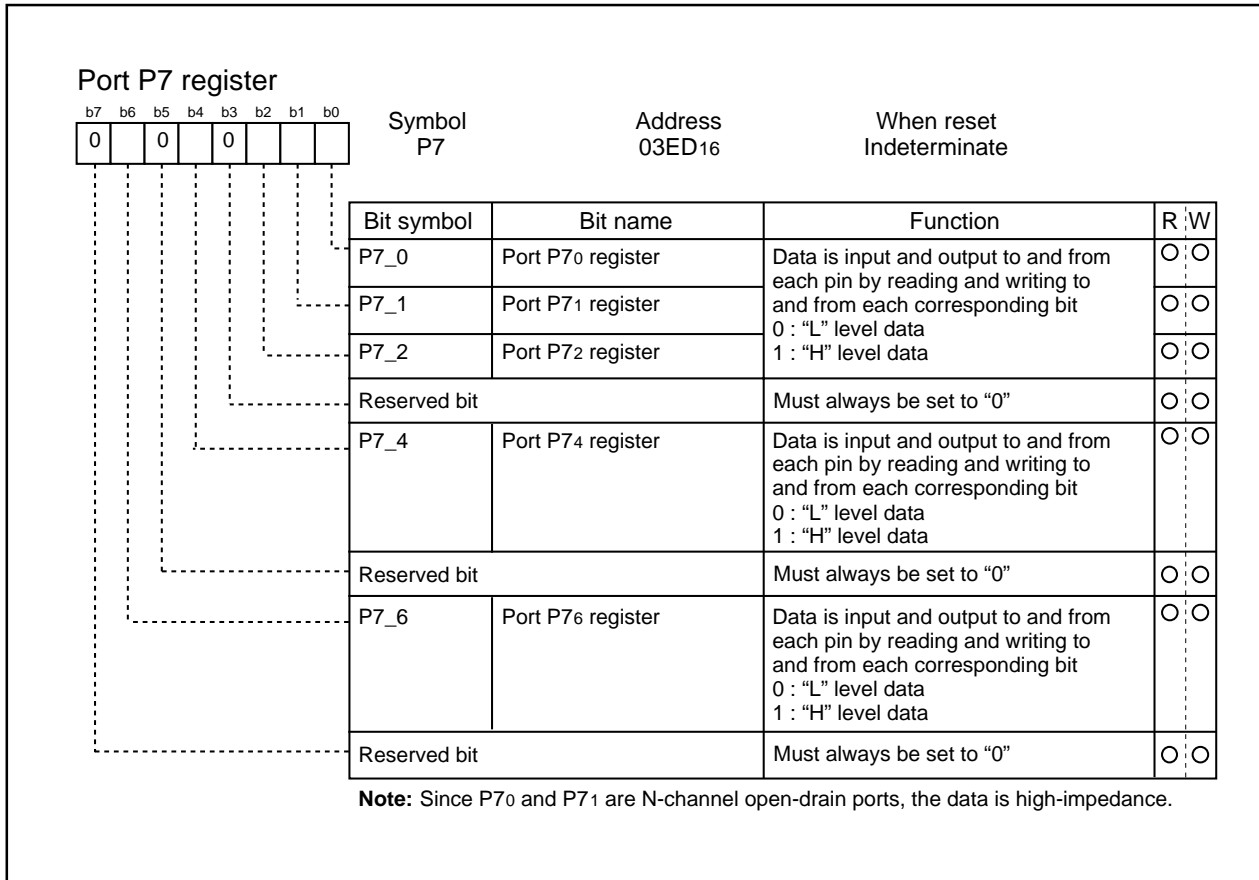


Figure 2.17.17 Port P7 register

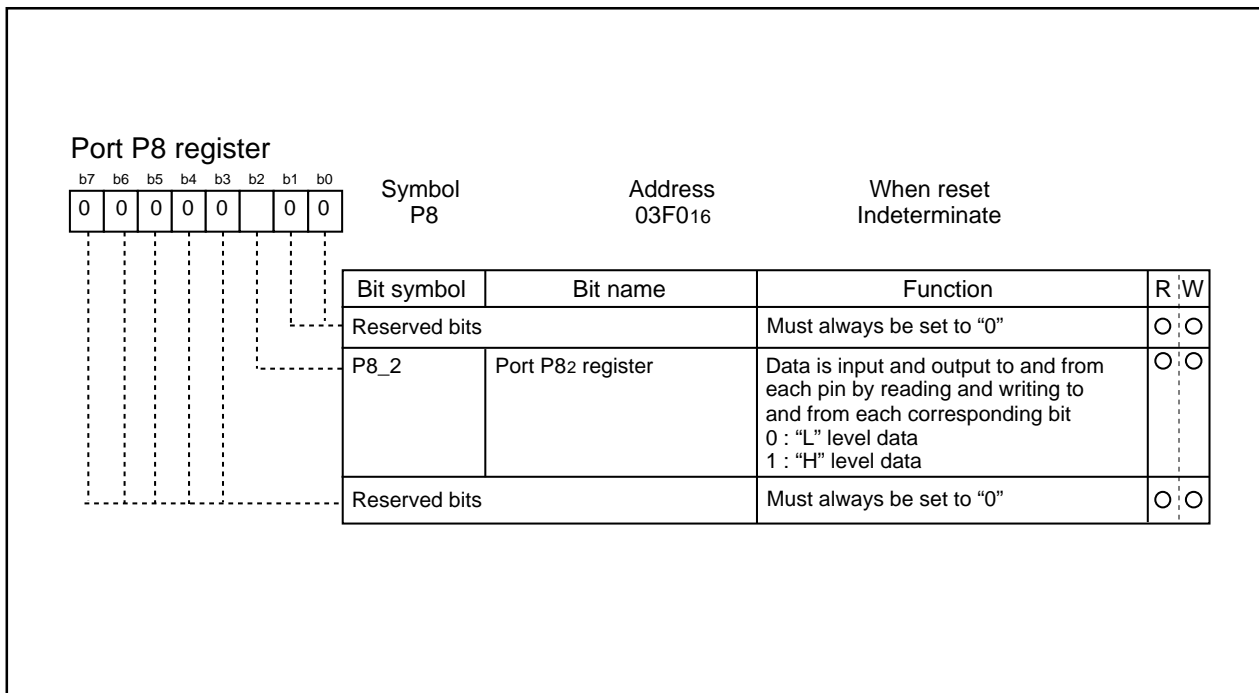


Figure 2.17.18 Port P8 register

**M306V5ME-XXXSP**  
**M306V5EESP**

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

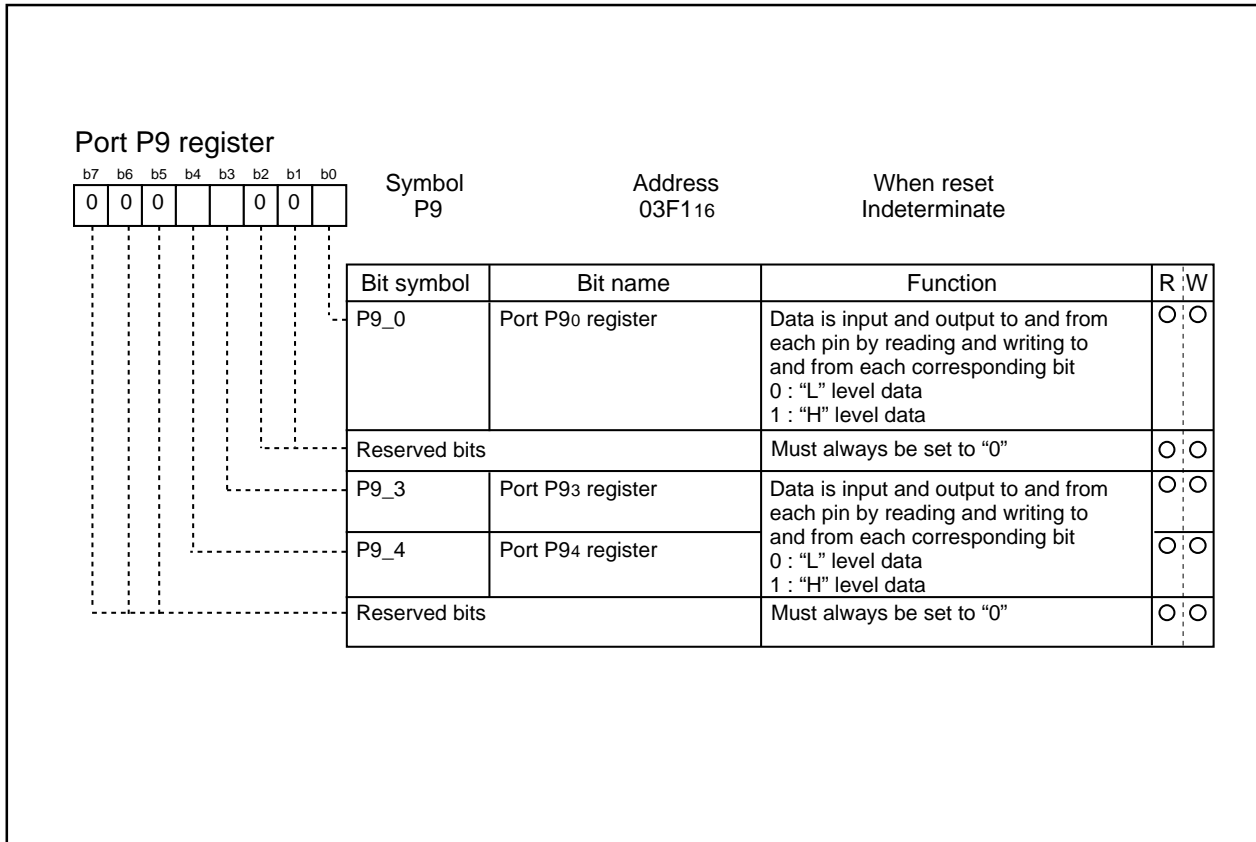


Figure 2.17.19 Port P9 register

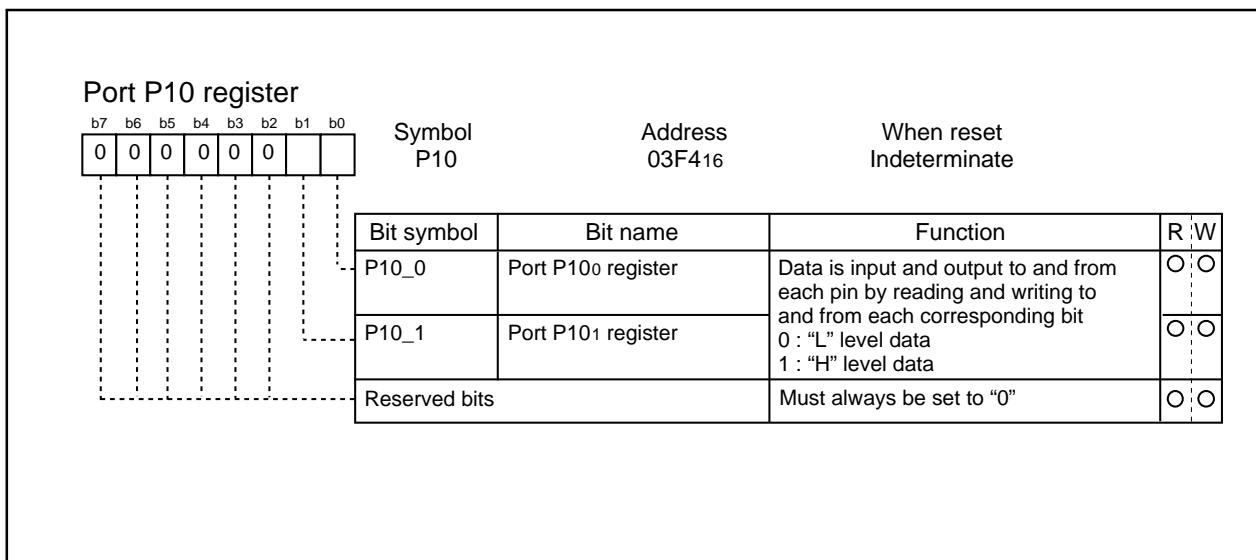


Figure 2.17.20 Port P10 register

**M306V5ME-XXXSP**  
**M306V5EESP**

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

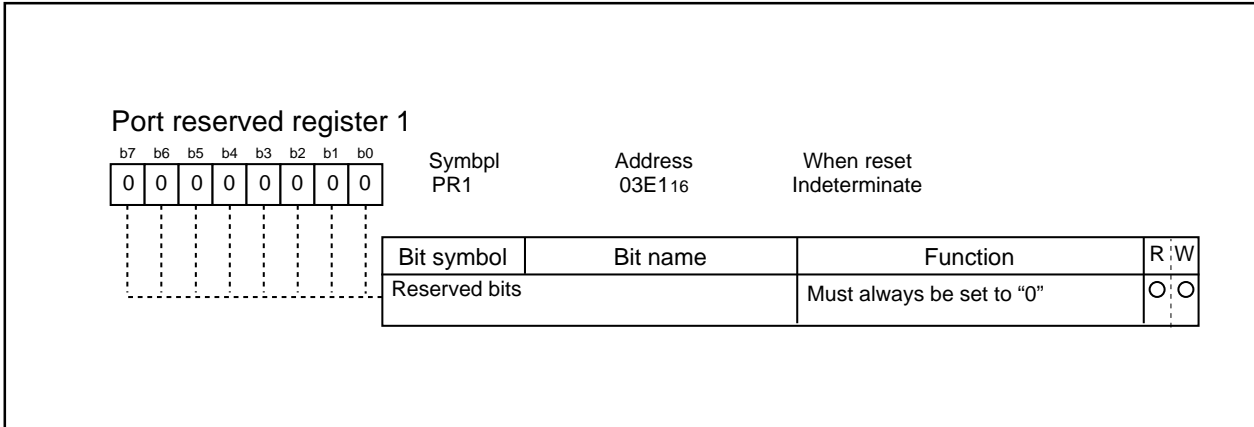


Figure 2.17.21 Port reserved register 1

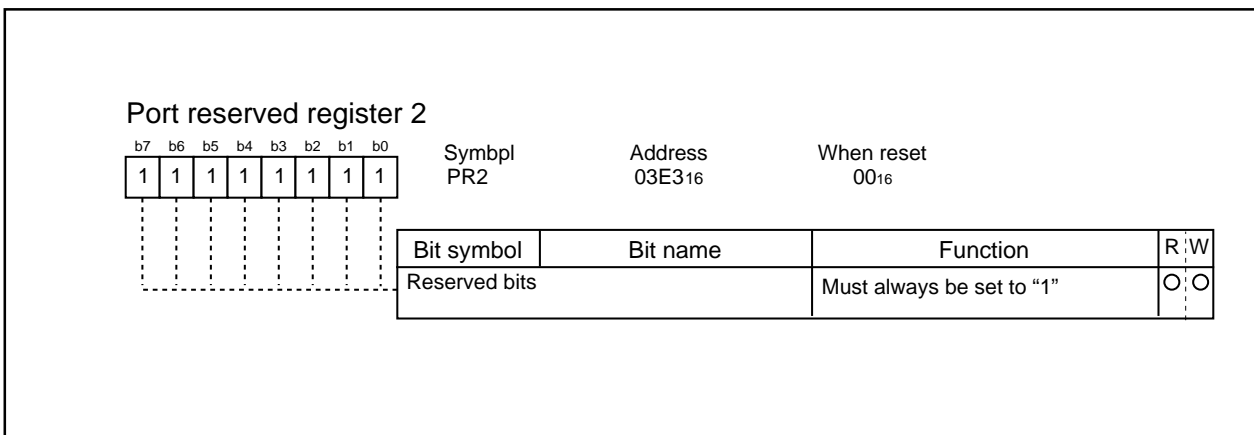


Figure 2.17.22 Port reserved register 2

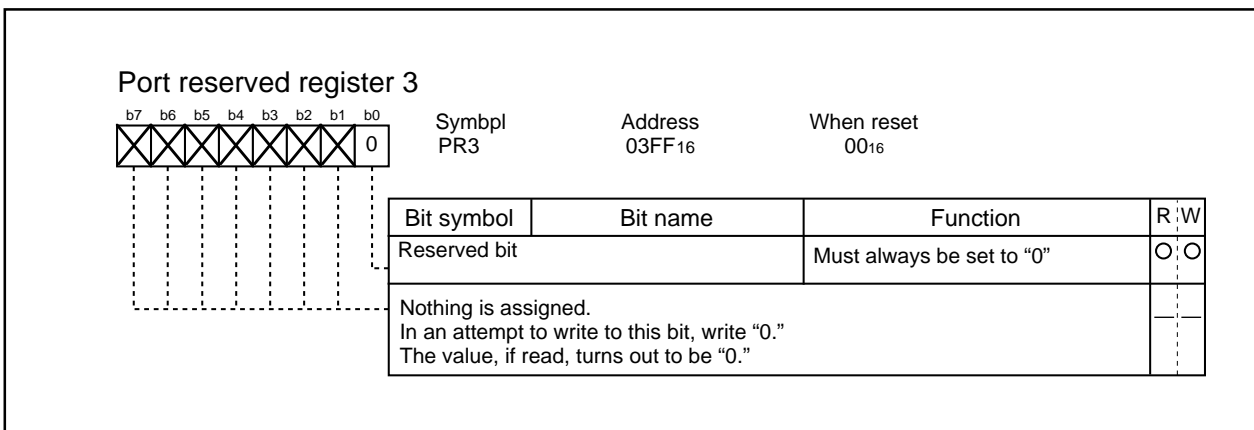
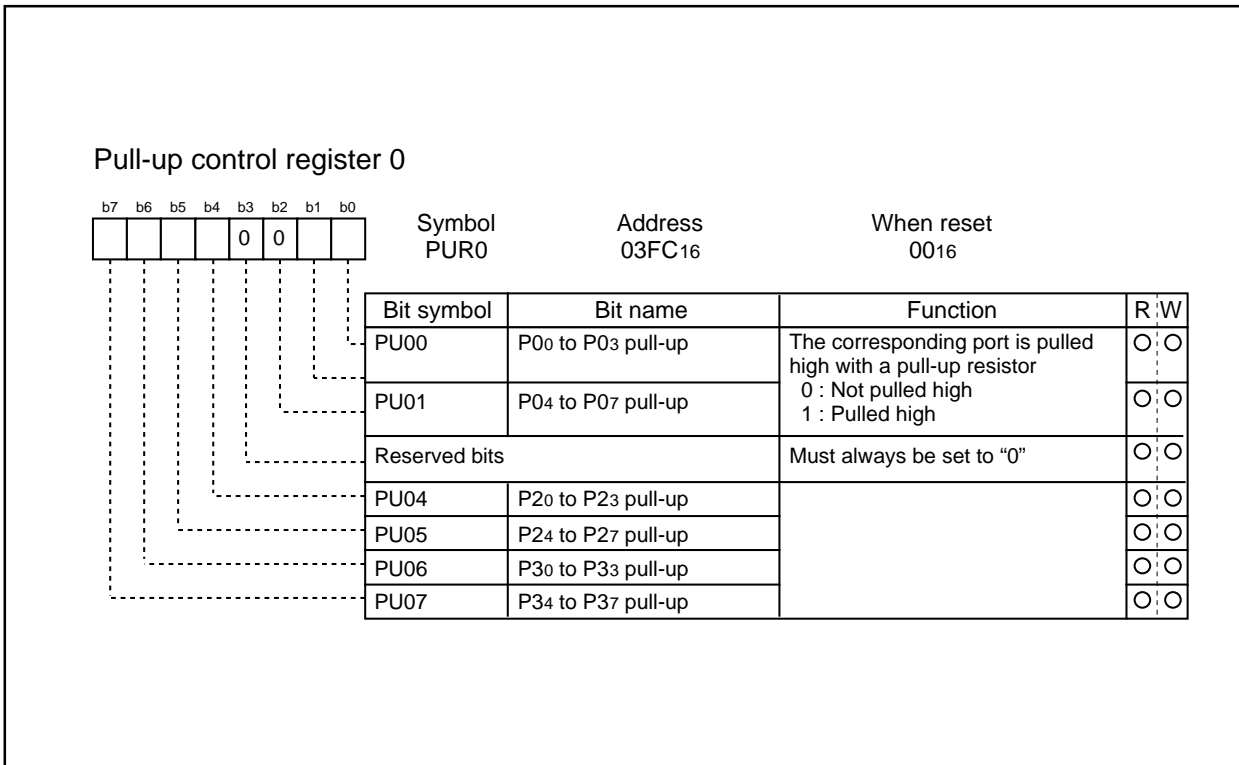


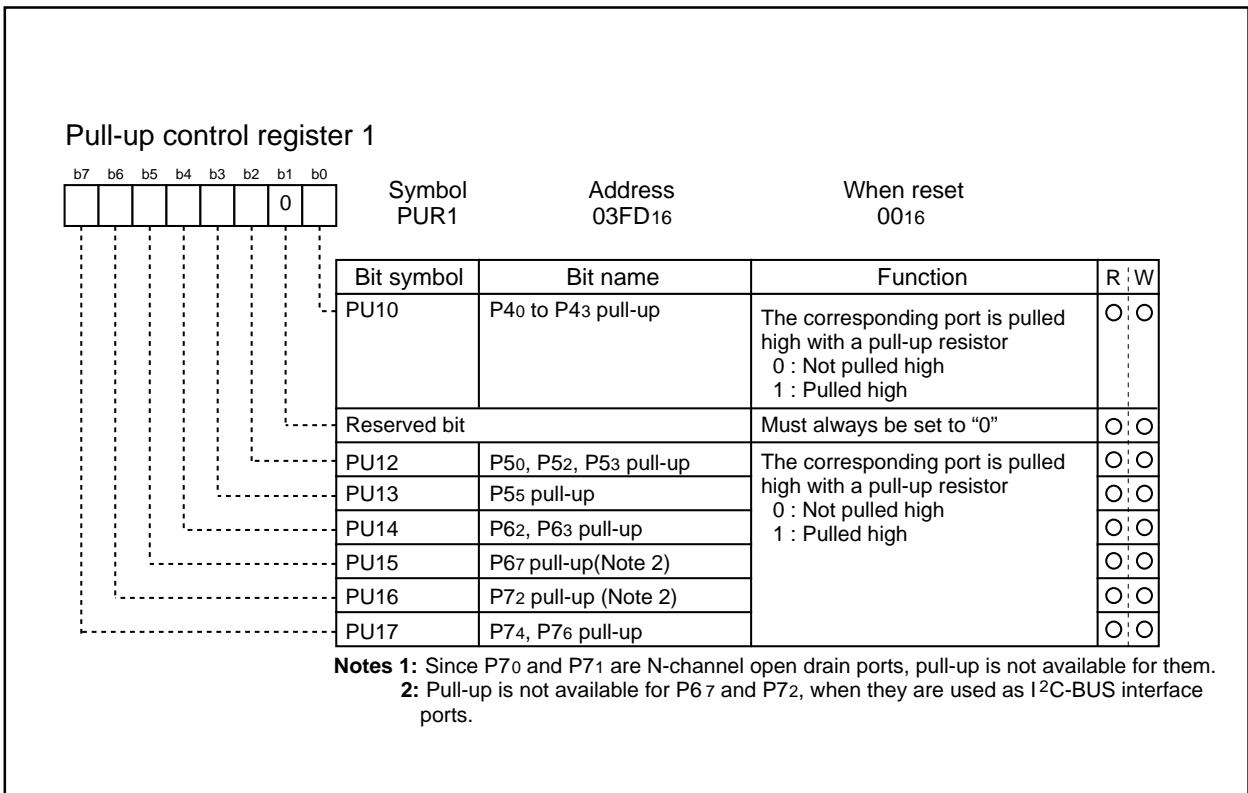
Figure 2.17.23 Port reserved register 3

**M306V5ME-XXXSP**  
**M306V5EESP**

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



**Figure 2.17.24 Pull-up control register 0**



**Figure 2.17.25 Pull-up control register 1**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

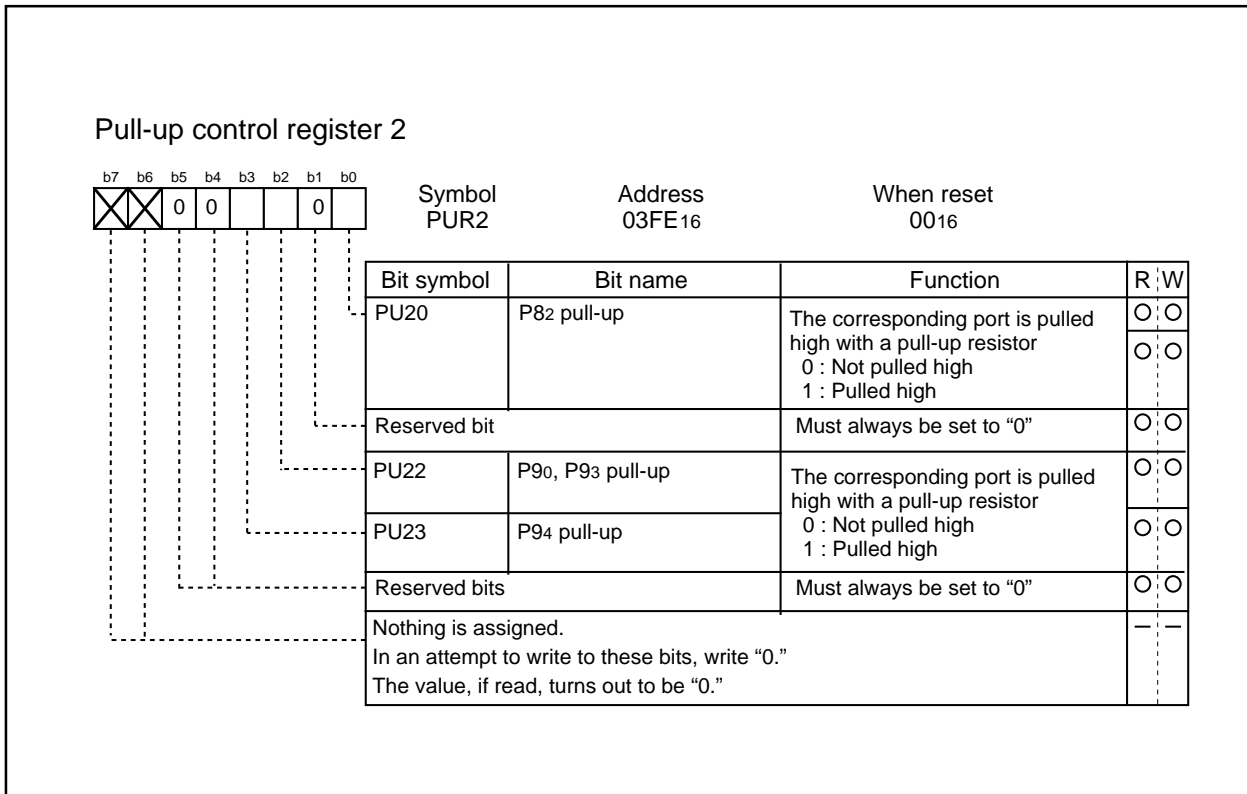


Figure 2.17.26 Pull-up control register 2

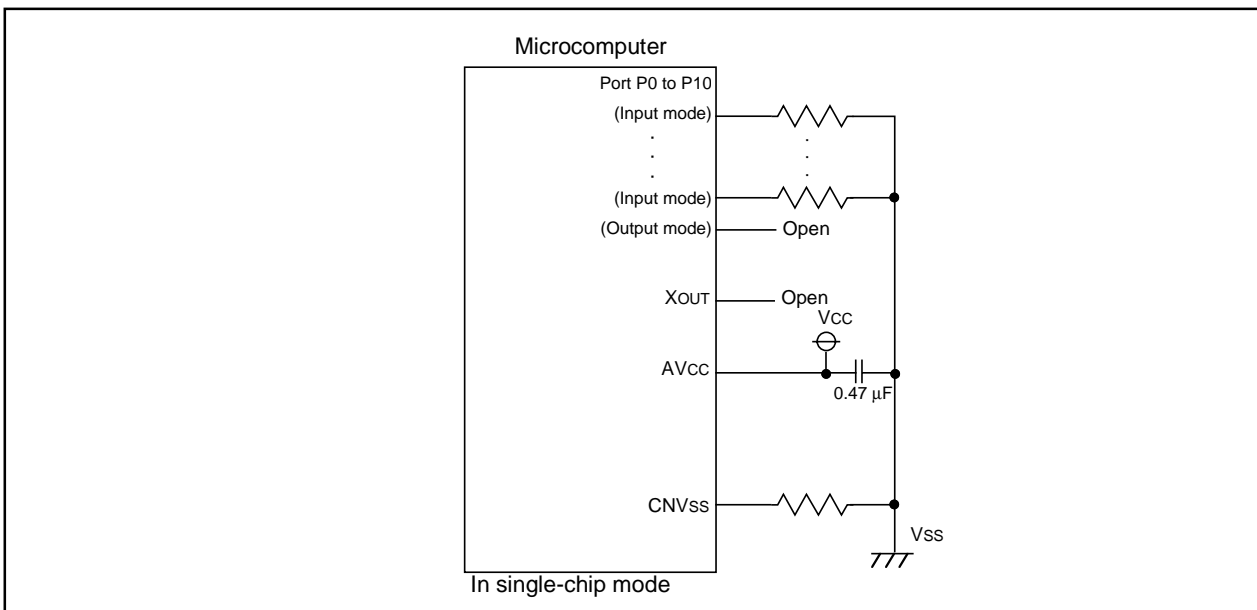
**M306V5ME-XXXSP**  
**M306V5EESP**

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, P2 to P10	After setting for input mode, connect every pin to Vss or Vcc via a resistor; or after setting for output mode, leave these pins open.
XOUT (Note)	Open
AVcc	Connect to VCC
CNVss	Connect via resistor to Vss (pull-down)

**Note:** With external clock input to XIN pin.



**Figure 2.17.27 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 "FFFF<sub>16</sub>". 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
  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 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

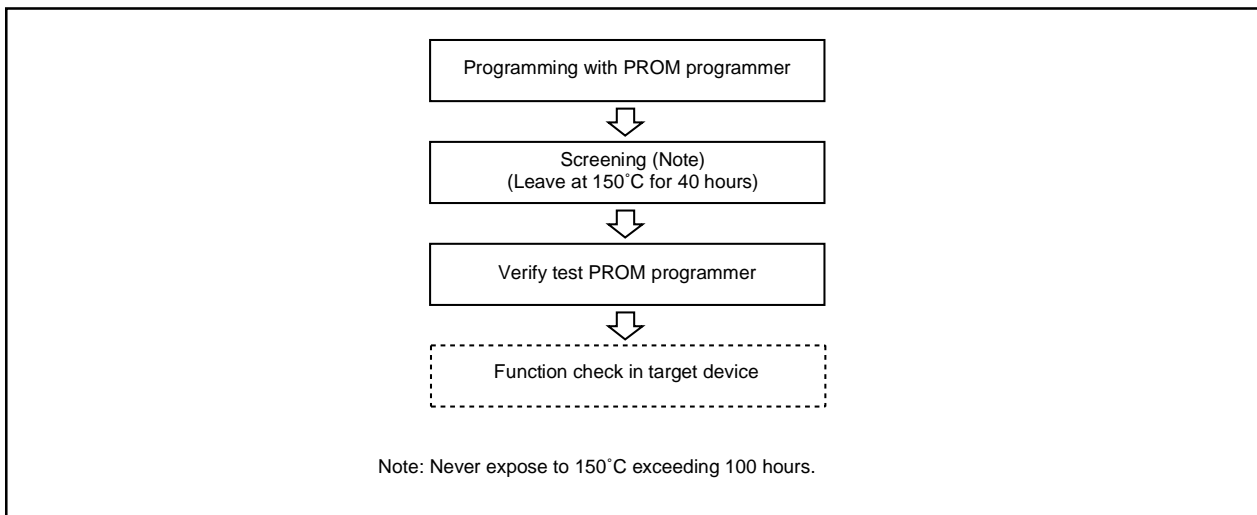
### 3.10 Built-in PROM Version

#### 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.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**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 6.0	V
AV <sub>cc</sub>	Analog supply voltage		-0.3 to 6.0	V
V <sub>i</sub>	Input voltage P0 <sub>0</sub> to P0 <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>3</sub> , P5 <sub>0</sub> , P5 <sub>2</sub> , P5 <sub>3</sub> , P5 <sub>5</sub> , P6 <sub>2</sub> , P6 <sub>3</sub> , P6 <sub>7</sub> , P7 <sub>0</sub> , P7 <sub>1</sub> , P7 <sub>2</sub> , P7 <sub>4</sub> , P7 <sub>6</sub> , P8 <sub>2</sub> , P9 <sub>0</sub> , P9 <sub>3</sub> , P9 <sub>4</sub> , P10 <sub>0</sub> , P10 <sub>1</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>		-0.3 to 6.0 (Note)	V
V <sub>o</sub>	Output voltage P0 <sub>0</sub> to P0 <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>3</sub> , P5 <sub>0</sub> , P5 <sub>2</sub> , P5 <sub>3</sub> , P5 <sub>5</sub> , P6 <sub>2</sub> , P6 <sub>3</sub> , P6 <sub>7</sub> , P7 <sub>0</sub> , P7 <sub>1</sub> , P7 <sub>2</sub> , P7 <sub>4</sub> , P7 <sub>6</sub> , P8 <sub>2</sub> , P9 <sub>0</sub> , P9 <sub>3</sub> , P9 <sub>4</sub> , P10 <sub>0</sub> , P10 <sub>1</sub> , R, G, B, OUT1, OUT2, OSC2, X <sub>OUT</sub>		-0.3 to V <sub>cc</sub> +0.3	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 CNV<sub>ss</sub> is -0.3 to 13(V).

**M306V5ME-XXXSP**  
**M306V5EESP**

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 Vcc = 4.5 V to 5.5 V at Ta = – 10 °C to 70 °C unless otherwise specified)**

Symbol	Parameter		Standard			Unit
			Min	Typ.	Max.	
Vcc	Supply voltage (Note 3)		4.5	5.0	5.5	V
AVcc	Analog supply voltage (Note 3)			Vcc		V
Vss	Supply voltage			0		V
V <sub>IH</sub>	HIGH input voltage	P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P70, P71, P72, P74, P76, P82, P90, P93, P94, P100, P101, X <sub>IN</sub> , OSC1, RESET, CNVss	0.8Vcc		Vcc	V
V <sub>IL</sub>	LOW input voltage	P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P70, P71, P72, P74, P76, P82, P90, P93, P94, P100, P101, X <sub>IN</sub> , OSC1, RESET, CNVss	0		0.2Vcc	V
I <sub>OH</sub> (peak)	HIGH peak output current	P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P72, P74, P76, P82, P90, P93, P94, P100, P101, R, G, B, OUT1, OUT2			–10.0	mA
I <sub>OH</sub> (avg)	HIGH average output current	P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P72, P74, P76, P82, P90, P93, P94, P100, P101, R, G, B, OUT1, OUT2			–5.0	mA
I <sub>OL</sub> (peak)	LOW peak output current	P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P70, P71, P72, P74, P76, P82, P90, P93, P94, P100, P101, R, G, B, OUT1, OUT2			10.0	mA
I <sub>OL</sub> (avg)	LOW average output current	P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P74, P76, P82, P90, P100, P101, R, G, B, OUT1, OUT2			5.0	mA
I <sub>OL</sub> (avg)	LOW average output current	P67, P70 to P72, P93, P94			6.0	mA
f (X <sub>IN</sub> )	Main clock input oscillation frequency				10	MHz
f <sub>osc</sub>	Oscillation frequency (for OSD)	OSC1	LC oscillating mode	11.0	27.0	MHz
			Ceramic oscillating mode	15.0	27.0	
f <sub>CVIN</sub>	Input frequency	Horizontal sync. signal of video signal	15.262	15.743	16.206	kHz
V <sub>I</sub>	Input amplitude video signal	CV <sub>IN</sub>	1.5	2.0	2.5	V

**Notes 1:** The mean output current is the mean value within 100 ms.

**2:** The total I<sub>OL</sub> (peak) for ports P0, P2, P9, and P10 must be 80 mA max. The total I<sub>OH</sub> (peak) for ports P0, P2, P9, and P10 must be 80 mA max. The total I<sub>OL</sub> (peak) for ports P3, P4, P5, P6, P7 and P82 must be 80 mA max. The total I<sub>OH</sub> (peak) for ports P3, P4, P5, P6, P72, P74, P76, and P82 must be 80 mA max.

**3:** Connect 0.1 μF or more capacitor externally between the power source pins Vcc–Vss and AVcc–Vss so as to reduce power source noise. Also connect 0.1 μF or more capacitor externally between the power source pins Vcc–CNVss.

**M306V5ME-XXXSP**  
**M306V5EESP**

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 Vcc = 5 V, Vss = 0 V at Ta = 25 °C, f(XIN) = 10 MHz unless otherwise specified)**

Symbol	Parameter	Measuring condition	Standard			Unit	
			Min.	Typ.	Max.		
V <sub>OH</sub>	HIGH output voltage P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P72, P74, P76, P82, P90, P93, P94, P100, P101, R, G, B, OUT1, OUT2	I <sub>OH</sub> = -5 mA	3.0			V	
V <sub>OH</sub>	HIGH output voltage P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55	I <sub>OH</sub> = -200 μA	4.7			V	
V <sub>OH</sub>	HIGH output voltage XOUT	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 P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P74, P76, P82, P90, P100, P101, R, G, B, OUT1, OUT2	I <sub>OL</sub> = 5 mA			2.0	V	
V <sub>OL</sub>	LOW output voltage P67, P70 to P72, P93, P94	I <sub>OL</sub> = 6.0 mA			0.6	V	
V <sub>OL</sub>	LOW output voltage P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55	I <sub>OL</sub> = 200 μA			0.45	V	
V <sub>OL</sub>	LOW output voltage XOUT	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 TB0IN, INT0, INT1, CLK0, CLK2, SCL1, SCL2, SCL3, SDA1, SDA2, SDA3, Hsync, Vsync, HC0, HC1, RxD0, RxD2		0.2		0.8	V	
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis RESET		0.2		1.8	V	
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis XIN		0.2		0.8	V	
I <sub>IH</sub>	HIGH input current P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P70, P71, P72, P74, P76, P82, P90, P93, P94, P100, P101 XIN, RESET, CNVss, OSC1	V <sub>i</sub> = 5 V			5.0	μA	
I <sub>IL</sub>	LOW input current P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P70, P71, P72, P74, P76, P82, P90, P93, P94, P100, P101 XIN, RESET, CNVss, OSC1	V <sub>i</sub> = 0 V			-5.0	μA	
PPULLUP	Pull-up resistor P00 to P07, P20 to P27, P30 to P37, P40 to P43, P50, P52, P53, P55, P62, P63, P67, P72, P74, P76, P82, P90, P93, P94	V <sub>i</sub> = 0 V	30.0	50.0	167.0	kΩ	
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(XIN) = 10 MHz Square wave, no division	OSD ON, Data slicer ON	70	90	mA
				OSD OFF, Data slicer OFF	30	50	
			f(XIN) = 10 MHz Square wave, division by 8	OSD OFF, Data slicer OFF	10		mA
				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>XIN</sub>	Feedback resistor XIN			1.0		MΩ	
R <sub>XCIN</sub>	Feedback resistor XCIN			6.0		MΩ	

**M306V5ME-XXXSP**  
**M306V5EESP**

 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			$\mu\text{s}$
$t_{SAMP}$	Sampling time			0.3			$\mu\text{s}$
$V_{REF}$	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	$\mu\text{s}$
$R_o$	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 ( $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_a = 25\text{ }^\circ\text{C}$ ,  $f(X_{IN}) = 10\text{ MHz}$  unless otherwise specified)**

Symbol	Parameter		Test conditions	Standard		Unit
				Min.	Max.	
$R_o$	Output impedance				2	$k\Omega$
$V_{OE}$	Output deviation				$\pm 0.5$	V
$T_{ST}$	Settling time		load capacity of 10 pF, load resistance of 20 $k\Omega$ , 70 % DC level		50	ns



**M306V5ME-XXXSP**  
**M306V5EESP**

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

**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.	
t <sub>c</sub>	External clock input cycle time	100		ns
t <sub>w(H)</sub>	External clock input HIGH pulse width	40		ns
t <sub>w(L)</sub>	External clock input LOW pulse width	40		ns
t <sub>r</sub>	External clock rise time		15	ns
t <sub>f</sub>	External clock fall time		15	ns

**Table 5.7.2 Timer B input (counter input in event counter mode)**

(referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C unless otherwise specified)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TB)</sub>	TB0IN input cycle time (counted on one edge)	100		ns
t <sub>w(TBH)</sub>	TB0IN input HIGH pulse width (counted on one edge)	40		ns
t <sub>w(TBL)</sub>	TB0IN input LOW pulse width (counted on one edge)	40		ns
t <sub>c(TB)</sub>	TB0IN input cycle time (counted on both edges)	200		ns
t <sub>w(TBH)</sub>	TB0IN input HIGH pulse width (counted on both edges)	80		ns
t <sub>w(TBL)</sub>	TB0IN input LOW pulse width (counted on both edges)	80		ns

**Table 5.7.3 Timer B input (pulse period measurement mode)**

(referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C unless otherwise specified)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TB)</sub>	TB0IN input cycle time	400		ns
t <sub>w(TBH)</sub>	TB0IN input HIGH pulse width	200		ns
t <sub>w(TBL)</sub>	TB0IN input LOW pulse width	200		ns

**Table 5.7.4 Timer B input (pulse width measurement mode)**

(referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C unless otherwise specified)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c(TB)</sub>	TB0IN input cycle time	400		ns
t <sub>w(TBH)</sub>	TB0IN input HIGH pulse width	200		ns
t <sub>w(TBL)</sub>	TB0IN input LOW pulse width	200		ns

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**Table 5.7.5 Serial I/O (referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C unless otherwise specified)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>c</sub> (CK)	CLKi input cycle time	200		ns
t <sub>w</sub> (CKH)	CLKi input HIGH pulse width	100		ns
t <sub>w</sub> (CKL)	CLKi input LOW pulse width	100		ns
t <sub>d</sub> (C-Q)	TxDi output delay time		80	ns
t <sub>h</sub> (C-Q)	TxDi hold time	0		ns
t <sub>su</sub> (D-C)	RxDi input setup time	30		ns
t <sub>h</sub> (C-D)	RxDi input hold time	90		ns

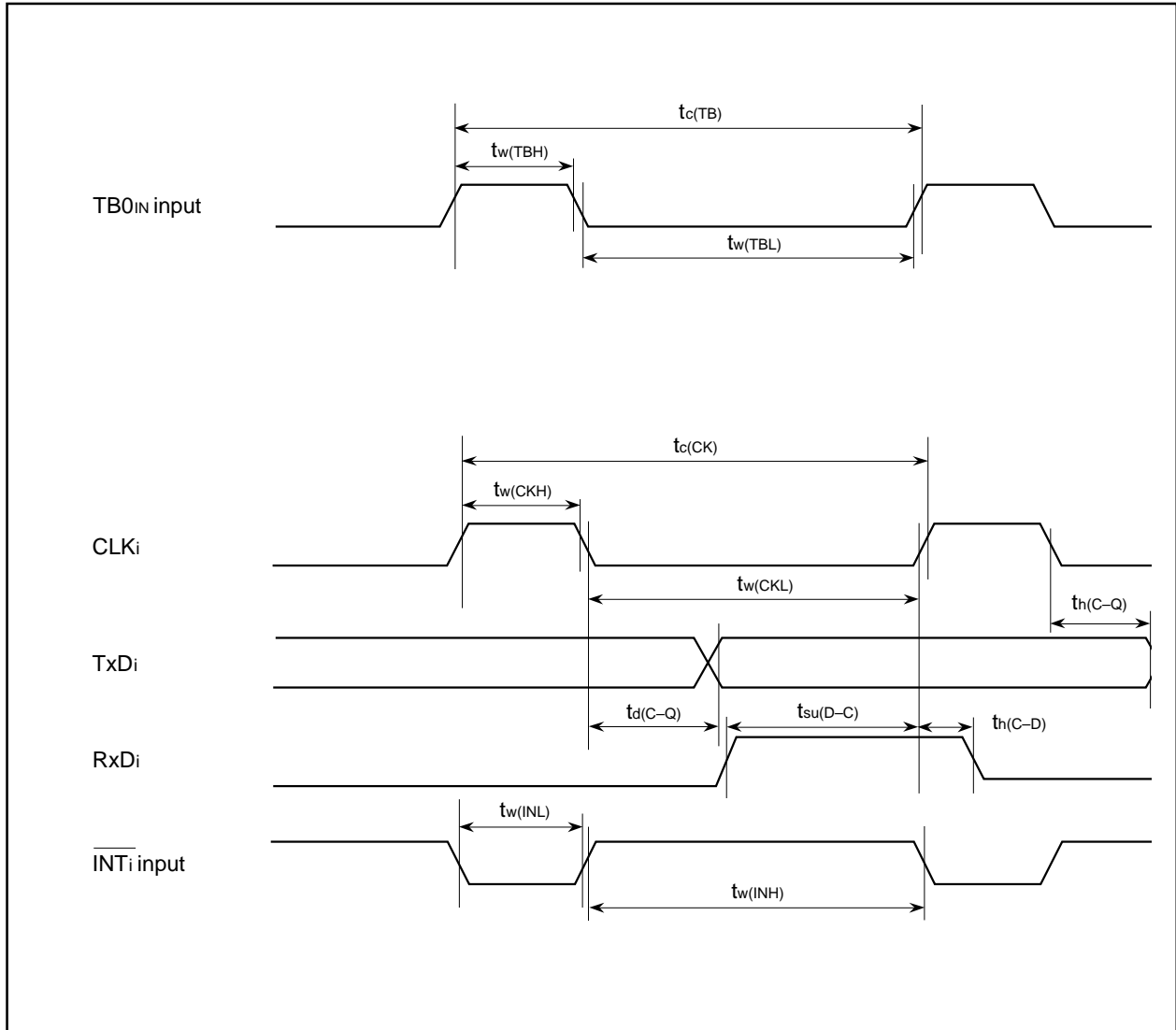
**Table 5.7.6 External interrupt  $\overline{\text{INT}}_i$  inputs (referenced to Vcc = 5 V, Vss = 0 V at Ta = 25 °C unless otherwise specified)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t <sub>w</sub> (INH)	$\overline{\text{INT}}_i$ input HIGH pulse width	250		ns
t <sub>w</sub> (INL)	$\overline{\text{INT}}_i$ input LOW pulse width	250		ns

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**5.8 Timing Diagram**



**Figure 5.8.1 Timing diagram**

**M306V5ME-XXXSP**  
**M306V5EESP**

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

**6. MASK CONFIRMATION FORM**

GZZ—SH56—45B <93A0>

**MITSUBISHI ELECTRIC SINGLE-CHIP 16-BIT  
MICROCOMPUTER M306V5ME-XXXSP  
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

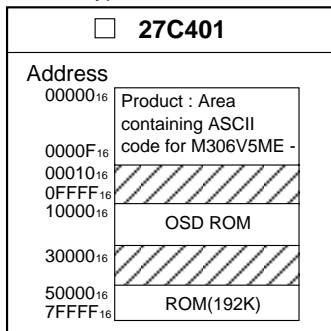
Mitsubishi will create the mask using the data on the EPROMs supplied, providing the data is the same on at least two of those sets. Mitsubishi will, therefore, only accept liability if there is any discrepancy between the data on the EPROM sets and the ROM data written to the product.  
Please carefully 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 'M306V5ME-' 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> '5' = 35<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> '5' = 35 <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> '5' = 35 <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>																				

# M306V5ME-XXXSP M306V5EESP

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

GZZ—SH56—45B <93A0>

## MITSUBISHI ELECTRIC SINGLE-CHIP 16-BIT MICROCOMPUTER M306V5ME-XXXSP 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.)

100FE	120FE	140FE	160FE	18002	1A002	1C002	1E002	20400	20401	20600	20601	213F8	213F9	213FC	213FD
100FF	120FF	140FF	160FF	18003	1A003	1C003	1E003	21400	21401	21600	21601	223F8	223F9	223FC	223FD
101FE	121FE	141FE	161FE	18102	1A102	1C102	1E102	22400	22401	22600	22601	233F8	233F9	233FC	233FD
101FF	121FF	141FF	161FF	18103	1A103	1C103	1E103	23400	23401	23600	23601	243F8	243F9	243FC	243FD
102FE	122FE	142FE	162FE	18202	1A202	1C202	1E202	24400	24401	24600	24601	253F8	253F9	253FC	253FD
102FF	122FF	142FF	162FF	18203	1A203	1C203	1E203	25400	25401	25600	25601	263F8	263F9	263FC	263FD
103FE	123FE	143FE	163FE	18302	1A302	1C302	1E302	26400	26401	26600	26601	273F8	273F9	273FC	273FD
103FF	123FF	143FF	163FF	18303	1A303	1C303	1E303	27400	27401	27600	27601	283F8	283F9	283FC	283FD
104FE	124FE	144FE	164FE	18402	1A402	1C402	1E402	28400	28401	28600	28601	293F8	293F9	293FC	293FD
104FF	124FF	144FF	164FF	18403	1A403	1C403	1E403	29400	29401	29600	29601	2A3F8	2A3F9	2A3FC	2A3FD
105FE	125FE	145FE	165FE	18502	1A502	1C502	1E502	2A400	2A401	2A600	2A601	21BF8	21BF9	21BFC	21BFD
105FF	125FF	145FF	165FF	18503	1A503	1C503	1E503	2B400	2B401	2B600	2B601	22BF8	22BF9	22BFC	22BFD
106FE	126FE	146FE	166FE	18602	1A602	1C602	1E602	2A400	2A401	2A600	2A601	21BF8	21BF9	21BFC	21BFD
106FF	126FF	146FF	166FF	18603	1A603	1C603	1E603	2B400	2B401	2B600	2B601	22BF8	22BF9	22BFC	22BFD
107FE	127FE	147FE	167FE	18702	1A702	1C702	1E702	2C400	2C401	2C600	2C601	23BF8	23BF9	23BFC	23BFD
107FF	127FF	147FF	167FF	18703	1A703	1C703	1E703	2D400	2D401	2D600	2D601	24BF8	24BF9	24BFC	24BFD
108FE	128FE	148FE	168FE	18802	1A802	1C802	1E802	2E400	2E401	2E600	2E601	25BF8	25BF9	25BFC	25BFD
108FF	128FF	148FF	168FF	18803	1A803	1C803	1E803	2F400	2F401	2F600	2F601	26BF8	26BF9	26BFC	26BFD
109FE	129FE	149FE	169FE	18902	1A902	1C902	1E902	20C00	20C01	20E00	20E01	27BF8	27BF9	27BFC	27BFD
109FF	129FF	149FF	169FF	18903	1A903	1C903	1E903	21C00	21C01	21E00	21E01	28BF8	28BF9	28BFC	28BFD
10AFE	12AFE	14AFE	16AFE	18A02	1AA02	1CA02	1EA02	22C00	22C01	22E00	22E01	29BF8	29BF9	29BFC	29BFD
10AFF	12AFF	14AFF	16AFF	18A03	1AA03	1CA03	1EA03	23C00	23C01	23E00	23E01	2ABF8	2ABF9	2ABFC	2ABFD
10BFE	12BFE	14BFE	16BFE	18B02	1AB02	1CB02	1EB02	24C00	24C01	24E00	24E01				
10BFF	12BFF	14BFF	16BFF	18B03	1AB03	1CB03	1EB03	25C00	25C01	25E00	25E01				
10CFE	12CFE	14CFE	16CFE	18C02	1AC02	1CC02	1EC02	26C00	26C01	26E00	26E01				
10CFF	12CFF	14CFF	16CFF	18C03	1AC03	1CC03	1EC03	27C00	27C01	27E00	27E01				
10DFE	12DFE	14DFE	16DFE	18D02	1AD02	1CD02	1ED02	28C00	28C01	28E00	28E01				
10DFF	12DFF	14DFF	16DFF	18D03	1AD03	1CD03	1ED03	29C00	29C01	29E00	29E01				
10EFE	12EFE	14EFE	16EFE	18E02	1AE02	1CE02	1EE02	10000	10800	10001	10801				
10EFF	12EFF	14EFF	16EFF	18E03	1AE03	1CE03	1EE03	11000	11800	11001	11801				
10FFE	12FFE	14FFE	16FFE	18F02	1AF02	1CF02	1EF02	12000	12800	12001	12801				
10FFF	12FFF	14FFF	16FFF	18F03	1AF03	1CF03	1EF03	13000	13800	13001	13801				
110FE	130FE	150FE	170FE	19002	1B002	1D002	1F002	14000	14800	14001	14801				
110FF	130FF	150FF	170FF	19003	1B003	1D003	1F003	15000	15800	15001	15801				
111FE	131FE	151FE	171FE	19102	1B102	1D102	1F102	16000	16800	16001	16801				
111FF	131FF	151FF	171FF	19103	1B103	1D103	1F103	17000	17800	17001	17801				
112FE	132FE	152FE	172FE	19202	1B202	1D202	1F202	18000	18800	18001	18801				
112FF	132FF	152FF	172FF	19203	1B203	1D203	1F203	19000	19800	19001	19801				
113FE	133FE	153FE	173FE	19302	1B302	1D302	1F302	1A000	1A800	1A001	1A801				
113FF	133FF	153FF	173FF	19303	1B303	1D303	1F303	1B000	1B800	1B001	1B801				
114FE	134FE	154FE	174FE	19402	1B402	1D402	1F402	1C000	1C800	1C001	1C801				
114FF	134FF	154FF	174FF	19403	1B403	1D403	1F403	1D000	1D800	1D001	1D801				
115FE	135FE	155FE	175FE	19502	1B502	1D502	1F502	1E000	1E800	1E001	1E801				
115FF	135FF	155FF	175FF	19503	1B503	1D503	1F503	1F000	1F800	1F001	1F801				
116FE	136FE	156FE	176FE	19602	1B602	1D602	1F602								
116FF	136FF	156FF	176FF	19603	1B603	1D603	1F603								
117FE	137FE	157FE	177FE	19702	1B702	1D702	1F702								
117FF	137FF	157FF	177FF	19703	1B703	1D703	1F703								
118FE	138FE	158FE	178FE	19802	1B802	1D802	1F802								
118FF	138FF	158FF	178FF	19803	1B803	1D803	1F803								
119FE	139FE	159FE	179FE	19902	1B902	1D902	1F902								
119FF	139FF	159FF	179FF	19903	1B903	1D903	1F903								

(2/4)

**M306V5ME-XXXSP**  
**M306V5EESP**

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

GZZ—SH56—45B <93A0>

**MITSUBISHI ELECTRIC SINGLE-CHIP 16-BIT  
MICROCOMPUTER M306V5ME-XXXSP  
MASK ROM CONFIRMATION FORM**

Mask ROM number	
-----------------	--

The ASCII code for the type No. can be written to EPROM addresses 00000<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 △ ' M306V5ME- '

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 M306V5ME-XXXSP, submit the 64P4B mark specification sheet.

**M306V5ME-XXXSP**  
**M306V5EESP**

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

GZZ—SH56—45B <93A0>

**MITSUBISHI ELECTRIC SINGLE-CHIP 16-BIT  
 MICROCOMPUTER M306V5ME-XXXSP  
 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 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)

**M306V5ME-XXXSP**  
**M306V5EESP**

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

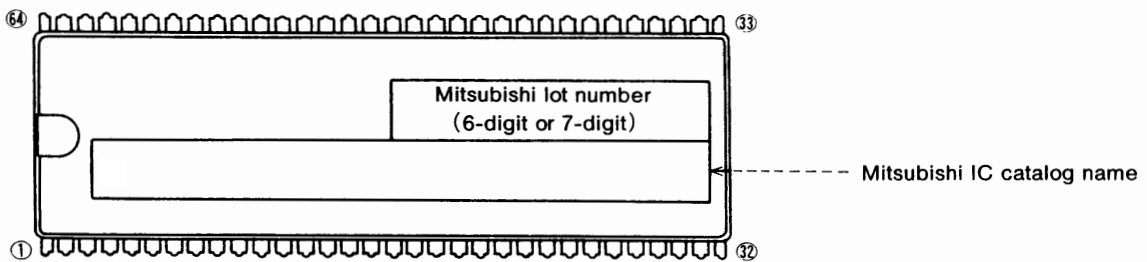
**7. MARK SPECIFICATION FORM**

**64P4B (64-PIN SHRINK DIP) MARK SPECIFICATION FORM**

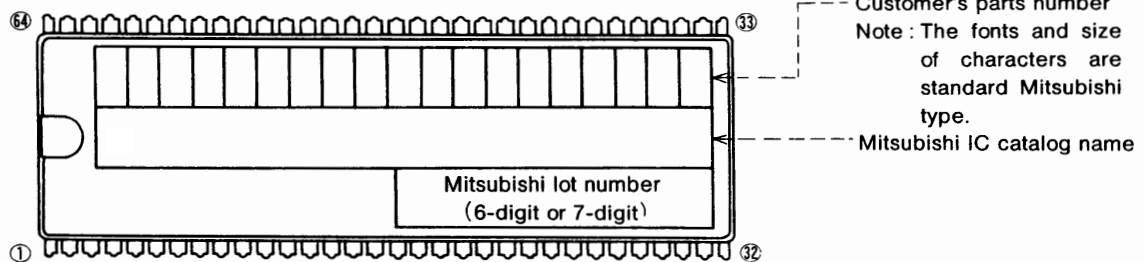
Mitsubishi IC catalog name

Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

**A. Standard Mitsubishi Mark**

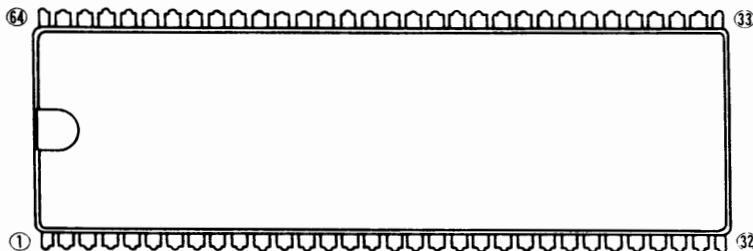


**B. Customer's Parts Number + Mitsubishi Catalog Name**



- Note1 :** The mark field should be written right aligned.  
**2 :** The fonts and size of characters are standard Mitsubishi type.  
**3 :** Customer's parts number can be up to 19 characters :  
 Only 0~9, A~Z, +, -, /, (, ), &, ©, . (period), and , (comma) are usable.  
**4 :** If the Mitsubishi logo ♪ is not required, check the box on the right.

**C. Special Mark Required**



- Note1 :** If the special mark is to be printed, indicate the desired layout of the mark in the upper 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

The standard Mitsubishi font is used for all characters except for a logo.

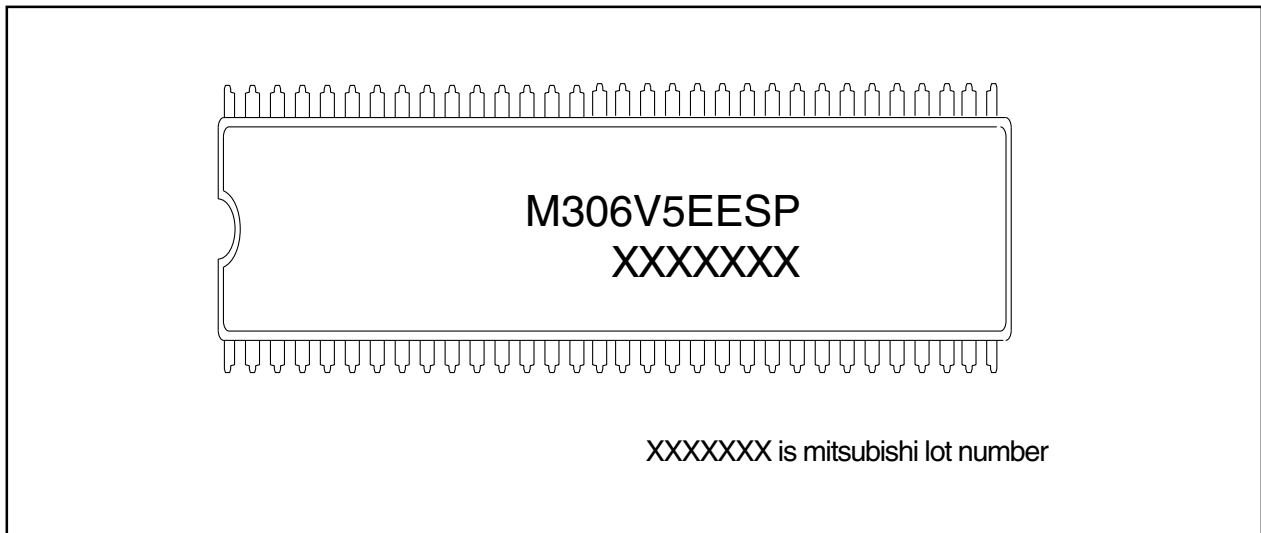


**M306V5ME-XXXSP**  
**M306V5EESP**

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

---

**8. ONE TIME PROM VERSION M306V5EESP MARKING**



**M306V5ME-XXXSP**  
**M306V5EESP**

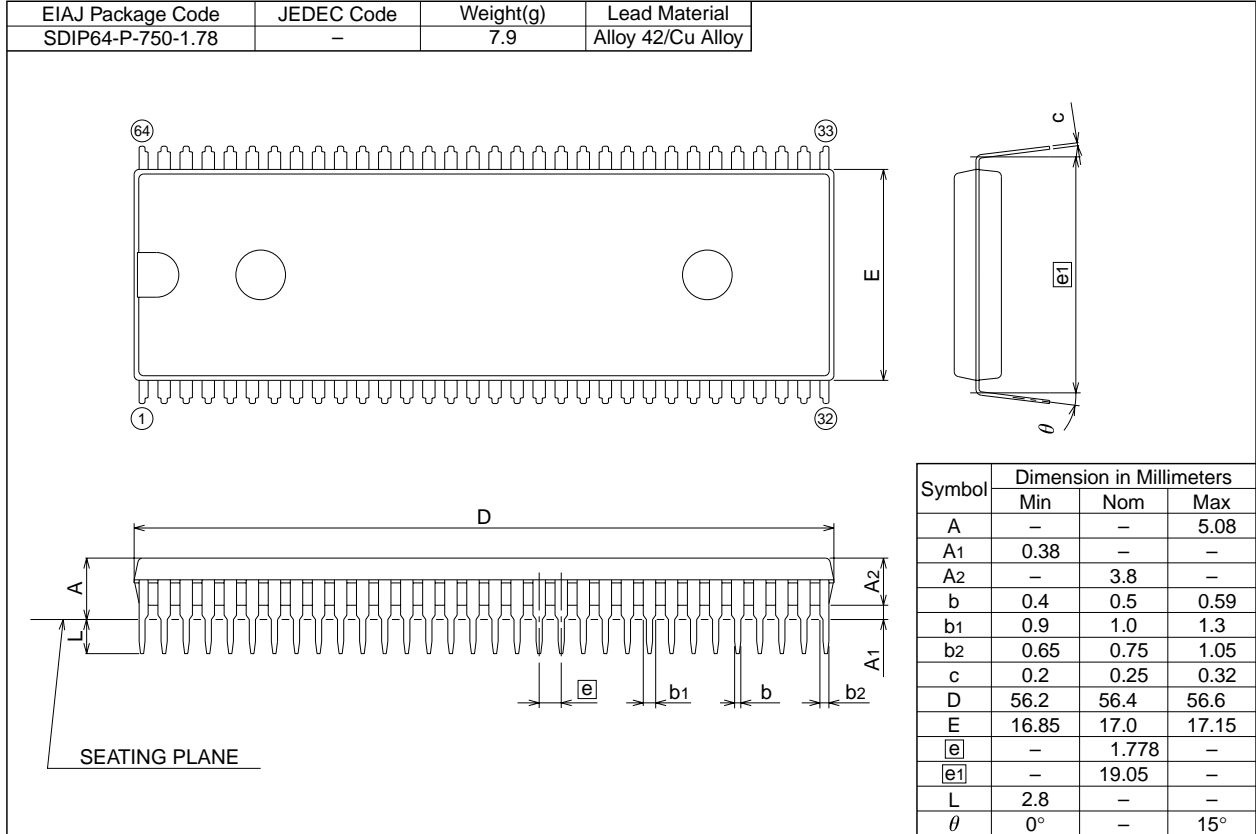
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER  
and ON-SCREEN DISPLAY CONTROLLER

**9. PACKAGE OUTLINE**

**64P4B**

(MMP)

Plastic 64pin 750mil SDIP



**M306V5ME-XXXSP**  
**M306V5EESP**

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)

b7 b6 b5 b4 b3 b2 b1 b0

0	0	0	0	0	X	1	0
---	---	---	---	---	---	---	---

Symbol  
PM1

Address  
000516

Values immediately after reset release (Note 1)  
When reset  
0000X002

Bit attributes (Note 2)

Bit symbol	Bit name	Function	R:W
	Reserved bit	Must always be set to "0"	○:○
	Reserved bit	Must always be set to "1"	○:○
	Nothing is assigned. In an attempt to write to this bit, 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:** As bit 1 of this register becomes "0" at reset, must always be set to "1" after reset release. Set bit 1 of the protect register (address 000A16) 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 read value is indeterminate unless otherwise mentioned.)

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) .....	43
Address match interrupt enable register (AIER) .....	53
Address match interrupt register i (RMADi) ..	53
A-D register i (ADi) .....	141
A-D control register 2 (ADCON2) .....	141
A-D control register 1 (ADCON1) .....	140, 143, 145, 147, 149, 150
A-D control register 0 (ADCON0) .....	140, 143, 145, 147, 149, 151
<b>[B]</b>	
Block control register i (BCi) .....	171
Bottom border control register (BBR) .....	215
Bus collision detection interrupt control register (BCNIC) .....	43
<b>[C]</b>	
Caption data register i (CDi) .....	13
Caption position register (CPS) .....	161
Clock control register (CS) .....	179
Clock run-in detect register (CRD) .....	162
Color palette register i (CRi) .....	119
Count start flag (TABSR) .....	71, 81
<b>[D]</b>	
D-A control register (DACON) .....	154
D-A register i (DAi) .....	154
Data clock position register (DPS) .....	163
Data slicer control register 1 (DSC1) .....	157
Data slicer control register 2 (DSC2) .....	157
Data slicer interrupt control register (DSIC) ..	43
Data slicer reserved register i (DRi) .....	164
DMA0 request cause select register (DM0SL) .....	60
DMA1 request cause select register (DM1SL) ..	61
DMAi control register (DMiCON) .....	61
DMAi interrupt control register (DMiIC) .....	43
DMAi destination pointer (DARi) .....	62
DMAi transfer counter (TCRi) .....	62
DMAi source pointer (SARi) .....	62
<b>[H]</b>	
Horizontal position register (HP) .....	176
HSYNC counter register (HC) .....	165
HSYNC counter latch .....	13
<b>[I]</b>	
I <sup>2</sup> Ci data shift register (IICiS0) .....	123
I <sup>2</sup> Ci address register (IICiS0D) .....	124
I <sup>2</sup> Ci status register (IICiS1) .....	131
I <sup>2</sup> Ci control register (IICiS1D) .....	128
I <sup>2</sup> Ci clock control register (IICiS2) .....	126
I <sup>2</sup> Ci port selection register (IICiS2D) .....	121
I <sup>2</sup> Ci transmit buffer register (IICiS0S) .....	123
I/O polarity control register (PC) .....	180
Interrupt control reserved register i (REiIC) .....	52
Interrupt request cause select register (IFSR) .....	52
INTi interrupt control register (INTiIC) .....	43
<b>[L]</b>	
Left border control register (LBR) .....	216
<b>[M]</b>	
Multi-master I <sup>2</sup> C-BUS interface i interrupt control register (IICiIC) .....	43
<b>[O]</b>	
One-shot start flag (ONSF) .....	72
OSD control register 1 (OC1) .....	170
OSD control register 2 (OC2) .....	173
OSD control register 3 (OC3) .....	198
OSD control register 4 (OC4) .....	182
OSD reserved register i (ORi) .....	220
OSDi interrupt control register (OSDiIC) .....	43
<b>[P]</b>	
Peripheral mode register (PM) .....	87

**M306V5ME-XXXSP**  
**M306V5EESP**

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

Port P0, P2, P3 register  
(P0 to P0, P2, P3) ..... 231

Port P0, P2, P3 direction register  
(PD0, PD2, PD3) ..... 227

Port P4 register (P4) ..... 231

Port P4 direction register (PD4) ..... 227

Port P5 register (P5) ..... 232

Port P5 direction register (PD5) ..... 228

Port P6 register (P6) ..... 232

Port P6 direction register (PD6) ..... 228

Port P7 register (P7) ..... 233

Port P7 direction register (PD7) ..... 229

Port P8 register (P8) ..... 233

Port P8 direction register (PD8) ..... 229

Port P9 register (P9) ..... 234

Port P9 direction register (PD9) ..... 230

Port P10 register (P10) ..... 234

Port P10 direction register (PD10) ..... 230

Port reserved register i (PRi) ..... 235

Processor mode register 0 (PM0) ..... 23

Processor mode register 1 (PM1) ..... 23

Protect register (PRCR) ..... 35

Pull-up control register 0 (PUR0) ..... 236

Pull-up control register 1 (PUR1) ..... 236

Pull-up control register 2 (PUR2) ..... 237

**[R]**

Raster color register (RSC) ..... 217

Reserved register i (INVCi) ..... 73, 81, 86

Right border control register (RBR) ..... 216

**[S]**

SPRITE horizontal position register (HS) .... 212

SPRITE OSD control register (SC) ..... 211

SPRITE vertical position register i (VSi) .... 212

System clock control register 0 (CM0) ..... 30

System clock control register 1 (CM1) ..... 30

**[T]**

Timer Ai interrupt control register (TAiIC)  
..... 43

Timer Bi interrupt control register (TBiIC)  
..... 43

Timer Ai register (TAi) ..... 71

Timer Bi register (TBi) ..... 81

Timer Ai mode register (TAiMR)  
..... 70, 74, 76, 77, 78

Timer Bi mode register (TBiMR)  
..... 80, 82, 83, 84

Top border control register (TBR) ..... 215

Trigger select register (TRGSR) ..... 73

**[U]**

UART transmit/receive control register 2 (UCON)  
..... 97

UART0 transmit/receive control register 0 (U0C0)  
..... 94

UART0 transmit/receive control register 1 (U0C1)  
..... 96

UART0 transmit/receive mode register (U0MR)  
..... 93, 100, 107

UART2 special mode register (U2SMR) ..... 97

UART2 transmit/receive mode register (U2MR)  
..... 93, 100, 107

UART2 transmit/receive control register 0 (U2C0)  
..... 95

UART2 transmit/receive control register 1 (U2C1)  
..... 96

UARTi bit rate generator (UiBRG) ..... 92

UARTi receive buffer register (UiRB) ..... 92

UARTi receive interrupt control register (SiRIC)  
..... 43

UARTi transmit buffer register (UiTB) ..... 92

UARTi transmit interrupt control register (SiTIC)  
..... 43

Up/down flag (UDF) ..... 72

**[V]**

VSYNC interrupt control register (VSYNCIC) ..... 43

Vertical position register i (VPi) ..... 176

**[W]**

Watchdog timer control register (WDC) ..... 57

Watchdog timer start register (WDTS) ..... 57

**M306V5ME-XXXSP**  
**M306V5EESP**

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

## Renesas Technology Corp.

Nippon Bldg., 6-2, Otemachi 2-chome, Chiyoda-ku, Tokyo, 100-0004 Japan

### 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 underwater 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

M306V5ME-XXXSP, M306V5EESP (REV.1.1) DATA SHEET

Rev. No.	Revision Description	Rev. date																								
1.0	First Edition of PDF File	0006																								
1.1	<p>P199 Fugure 2.16.30</p> <table border="0" data-bbox="284 551 1347 667"> <tr> <td></td> <td style="text-align: center;"><u>BEFORE</u></td> <td></td> <td style="text-align: center;"><u>AFTER</u></td> </tr> <tr> <td>G signal output control bit</td> <td style="text-align: center;">b2 b1 b0</td> <td style="text-align: center;">→</td> <td style="text-align: center;">b6 b5 b4</td> </tr> <tr> <td>B signal output control bit</td> <td style="text-align: center;">b2 b1 b0</td> <td style="text-align: center;">→</td> <td style="text-align: center;">b10 b9 b8</td> </tr> </table> <p>P245 Table 5.2.1</p> <table border="0" data-bbox="284 725 1347 842"> <tr> <td>fosc Oscillation frequency (for OSD) OSC1</td> <td style="text-align: center;"><u>BEFORE</u></td> <td></td> <td style="text-align: center;"><u>AFTER</u></td> </tr> <tr> <td>Ceramic oscillating mode</td> <td style="text-align: center;">Min. 24.0 MHz</td> <td style="text-align: center;">→</td> <td style="text-align: center;">15.0MHz</td> </tr> <tr> <td></td> <td style="text-align: center;">Max. 25.0 MHz</td> <td style="text-align: center;">→</td> <td style="text-align: center;">27.0MHz</td> </tr> </table>		<u>BEFORE</u>		<u>AFTER</u>	G signal output control bit	b2 b1 b0	→	b6 b5 b4	B signal output control bit	b2 b1 b0	→	b10 b9 b8	fosc Oscillation frequency (for OSD) OSC1	<u>BEFORE</u>		<u>AFTER</u>	Ceramic oscillating mode	Min. 24.0 MHz	→	15.0MHz		Max. 25.0 MHz	→	27.0MHz	0108
	<u>BEFORE</u>		<u>AFTER</u>																							
G signal output control bit	b2 b1 b0	→	b6 b5 b4																							
B signal output control bit	b2 b1 b0	→	b10 b9 b8																							
fosc Oscillation frequency (for OSD) OSC1	<u>BEFORE</u>		<u>AFTER</u>																							
Ceramic oscillating mode	Min. 24.0 MHz	→	15.0MHz																							
	Max. 25.0 MHz	→	27.0MHz																							