M37270MF-XXXSP
M37270EF-XXXSP, M37270EFSP

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER and ON-SCREEN DISPLAY CONTROLLER

DESCRIPTION

The M37270MF-XXXSP is a single-chip microcomputer designed with CMOS silicon gate technology. It is housed in a 64-pin shrink plastic molded DIP.

In addition to their simple instruction sets, the ROM, RAM and I/O addresses are placed on the same memory map to enable easy programming.

The M37270MF-XXXSP has a OSD function and a data slicer function, so it is useful for a channel selection system for TV with a closed caption decoder. The features of the M37270EF-XXXSP and the M37270EFSP are similar to those of the M37270MF-XXXSP except that these chips have a built-in PROM which can be written electrically.

FEATURES

- Number of basic instructions 71
- Memory size
  - ROM ..... 60 K bytes
  - RAM ..... 1024 bytes
  - ROM for OSD ..... 14464 bytes
  - RAM for OSD ..... 1920 bytes
- The minimum instruction execution time
  ..... 0.5µs (at 8 MHz oscillation frequency)
- Power source voltage ..... 5 V ± 10 %
- Subroutine nesting ..... 128 levels (Max.)
- Interrupts ..... 18 types, 16 vectors
- 8-bit timers 6
- Programmable I/O ports (Ports P0, P1, P2, $P3_0$, $P3_1$) 26
- Input ports (Ports $P4_0$–$P4_6$, $P6_3$, $P6_4$) 9
- Output ports (Ports $P3_2$, $P4_7$, P5, $P6_0$–$P6_2$, $P6_5$–$P6_7$) 16
- 12 V withstand ports 11
- LED drive ports 2
- Serial I/O ..... 8-bit × 1 channel
- Multi-master I^2C-BUS interface ..... 1 (2 systems)
- A-D converter (8-bit resolution) ..... 4 channels
- PWM output circuit ..... 8-bit × 8
- Interrupt interval determination circuit 1
- Power dissipation
  - In high-speed mode 165mW
    (at V_{CC} = 5.5V, 8MHz oscillation frequency, CRT on, and Data slicer on)
  - In low-speed mode 0.33mW
    (at V_{CC} = 5.5V, 32kHz oscillation frequency)
- Data slicer
- OSD function
  - Display characters ..... 40 characters × 16 lines
  - Kinds of characters ..... 320 kinds
    (In EXOSD mode, they can be combined with 32 kinds of extra fonts)
  - Dot structure ..... CC mode : 16 × 26 dots
    OSD mode : 16 × 20 dots
    EXOSD mode : 16 × 26 dots
  - Kinds of character sizes ..... CC mode : 2 types
    OSD mode : 14 types
    EXOSD mode : 6 types
  - It can be specified by a character unit (maximum 7 kinds).
    Character font coloring, character background coloring
  - It can be specified by a screen unit (maximum 7 kinds).
    Extra font coloring, raster coloring, border coloring
  - Kinds of character colors ..... CC mode : 7 kinds (R, G, B)
    OSD mode : 15 kinds (R, G, B, I)
    EXOSD mode : 7 kinds (R, G, B, I1, I2)
  - Display position
    - Horizontal ..... 256 levels
    - Vertical ..... 1024 levels
  - Attribute ..... CC mode : smooth italic, underline, flash
    OSD mode : border
    EXOSD mode : border, extra font (32 kinds)
  - Automatic solid space function
  - Window function
  - Dual layer OSD function

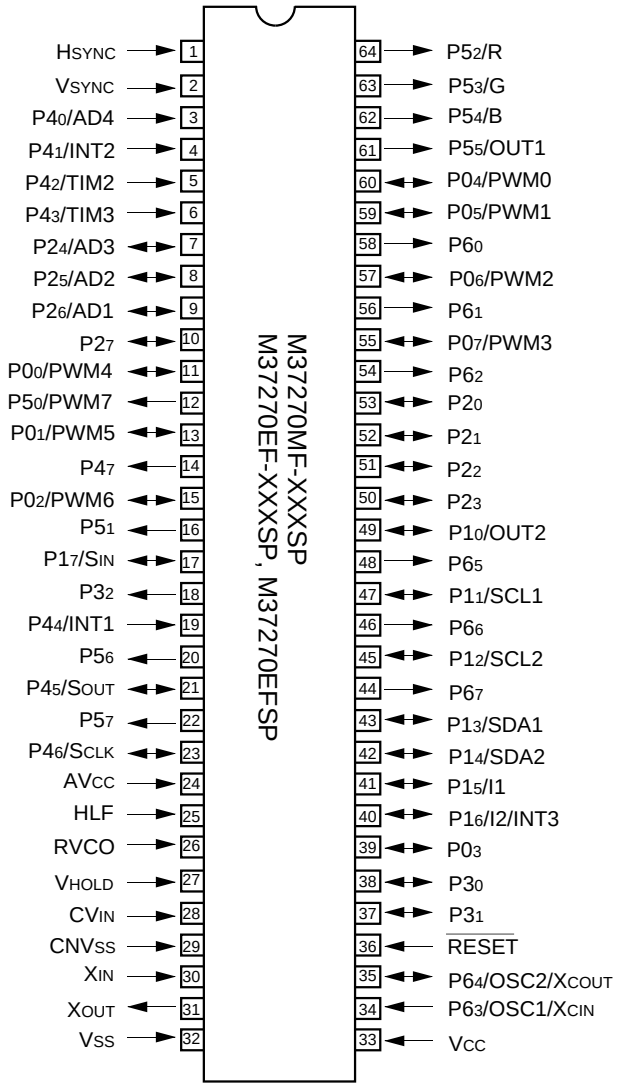
APPLICATION

TV with a closed caption decoder

M37270MF-XXXSP
M37270EF-XXXSP, M37270EFSP

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER
and ON-SCREEN DISPLAY CONTROLLER

PIN CONFIGURATION (TOP VIEW)



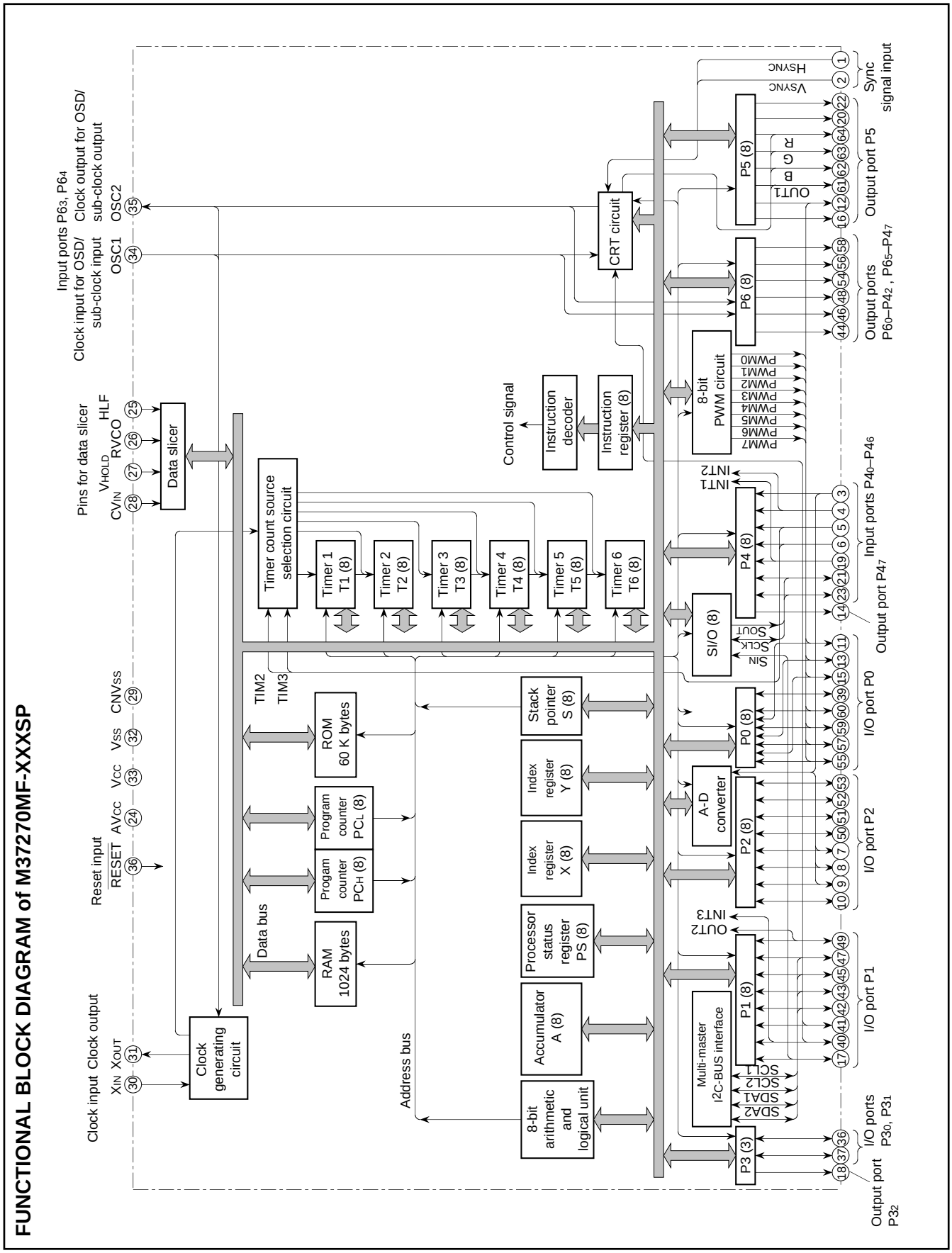
Outline 64P4B

MITSUBISHI ELECTRIC

M37270MF-XXXSP
M37270EF-XXXSP, M37270EFSP

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER and ON-SCREEN DISPLAY CONTROLLER

FUNCTIONAL BLOCK DIAGRAM of M37270MF-XXXSP



M37270MF-XXXSP
M37270EF-XXXSP, M37270EFSP

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER with CLOSED CAPTION DECODER and ON-SCREEN DISPLAY CONTROLLER

FUNCTIONS

Parameter			Functions
Number of basic instructions			71
Instruction execution time			0.5 μs (the minimum instruction execution time, at 8 MHz oscillation frequency)
Clock frequency			8 MHz (maximum)
Memory size	ROM		60 K bytes
	RAM		1024 bytes
	OSD ROM		14464 bytes
	OSD RAM		1920 bytes
Input/Output ports	$P0_0$–$P0_2$, $P0_4$–$P0_7$	I/O	7-bit × 1 (N-channel open-drain output structure, can be used as PWM output pins)
	$P0_3$	I/O	1-bit × 1 (CMOS input/output structure)
	$P1_0$, $P1_5$–$P1_7$	I/O	4-bit × 1 (CMOS input/output structure, can be used as OSD output pin, INT input pin, serial input pin)
	$P1_1$–$P1_4$	I/O	4-bit × 1 (N-channel open-drain output structure, can be used as multi-master I^2C-BUS interface)
	P2	I/O	8-bit × 1 (CMOS input/output structure, can be used as A-D input pins)
	$P3_0$, $P3_1$	I/O	2-bit × 1 (CMOS input/output structure)
	$P3_2$	Output	1-bit × 1 (N-channel open-drain output structure)
	$P4_0$–$P4_4$	Input	5-bit × 1 (can be used as A-D input pins, INT input pins, external clock input pins)
	$P4_5$, $P4_6$	Input	2-bit × 1 (N-channel open-drain output structure when serial I/O is used, can be used as serial I/O pins)
	$P4_7$	Output	1-bit × 1 (N-channel open-drain output structure)
	$P5_0$, $P5_1$, $P5_6$, $P5_7$	Output	4-bit × 1 (N-channel open-drain output structure, can be used as PWM output pins)
	$P5_2$–$P5_5$	Output	4-bit × 1 (CMOS output structure, can be used as OSD output)
	$P6_0$–$P6_2$, $P6_5$–$P6_7$	Output	6-bit × 1 (N-channel open-drain output)
	$P6_3$	Input	1-bit × 1 (can be used as sub-clock input pin, OSD clock input pin)
	$P6_4$	Input	1-bit × 1 (CMOS output structure when LC is oscillating, can be used as sub-clock output pin, OSD clock output pin)
Serial I/O			8-bit × 1
Multi-master I^2C-BUS interface			1
A-D converter			4 channels (8-bit resolution)
PWM output circuit			8-bit × 8
Timers			8-bit timer × 6
Subroutine nesting			128 levels (maximum)
Interrupt interval determination circuit			1
Interrupt			External interrupt × 3, Internal timer interrupt × 6, Serial I/O interrupt × 1, OSD interrupt × 1, Multi-master I^2C-BUS interface interrupt × 1, Data slicer interrupt × 1, f(X_{IN})/4092 interrupt × 1, V_{SYNC} interrupt × 1, A-D conversion interrupt × 1, BRK instruction interrupt × 1
Clock generating circuit			2 built-in circuits (externally connected a ceramic resonator or a quartz-crystal oscillator)
Data slicer			Built in

FUNCTIONS (continued)

OSD function				Number of display characters: 40 characters × 16 lines
				Dot structure: CC mode: 16 × 26 dots (character part : 16 × 20 dots) OSD mode: 16 × 20 dots EXOSD mode: 16 × 26 dots
				Kinds of characters: 320 kinds (In EXOSDmode, they can be combined with 32 kinds of extra fonts)
				Kinds of character sizes: CC mode: 2 kinds OSD mode: 14 kinds EXOSD mode: 6 kinds
				Kinds of character colors: CC mode: 7 kinds (R, G, B) OSD mode: 15 kinds (R, G, B, I1) EXOSD mode: 7 kinds (R, G, B, I1, I2)
				Display position (horizontal, vertical): 256 levels (horizontal) × 1024 levels (vertical)
Power source voltage				5 V ± 10 %
Power dissipation	In high-speed mode	OSD ON	Data slicer ON	165 mW typ. (at oscillation frequency f_{CPU} = 8 MHz, f_{OSD} = 13 MHz)
		OSD OFF	Data slicer OFF	82.5 mW typ. (at oscillation frequency f_{CPU} = 8 MHz)
	In low-speed mode	OSD OFF	Data slicer OFF	0.33mW typ. (at oscillation frequency f_{CLK} = 32 kHz, f(X_{IN}) = stopped)
	In stop mode			0.055 mW (maximum)
Operating temperature range				–10 °C to 70 °C
Device structure				CMOS silicon gate process
Package				64-pin shrink plastic molded DIP

PIN DESCRIPTION

Pin	Name	Input/ Output	Functions
V_{CC}, AV_{CC}, V_{SS}.	Power source		Apply voltage of 5 V ± 10 % (typical) to V_{CC} and AV_{CC}, and 0 V to V_{SS}.
CNVSS	CNVSS		This is connected to V_{SS}.
$\overline{RESET}$	Reset input	Input	To enter the reset state, the reset input pin must be kept at a “L” for 2 μs or more (under normal V_{CC} conditions). If more time is needed for the quartz-crystal oscillator to stabilize, this “L” condition should be maintained for the required time.
X_{IN}	Clock input	Input	This chip has an internal clock generating circuit. To control generating frequency, an external ceramic resonator or a quartz-crystal oscillator is connected between pins X_{IN} and X_{OUT}. If an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$P0_0$/PWM4–$P0_2$/PWM6, $P0_3$, $P0_4$/PWM0–$P0_7$/PWM3	I/O port P0	I/O	Port P0 is an 8-bit I/O port with direction register allowing each I/O bit to be individually programmed as input or output. At reset, this port is set to input mode. The output structure of $P0_3$ is CMOS output, that of $P0_0$–$P0_2$ and $P0_4$–$P0_7$ are N-channel open-drain output. The note out of this Table gives a full of port P0 function.
	PWM output	Output	Pins $P0_0$–$P0_2$ and $P0_4$–$P0_7$ are also used as PWM output pins PWM4–PWM6 and PWM0–PWM3 respectively. The output structure is N-channel open-drain output.
$P1_0$/OUT2, $P1_1$/SCL1, $P1_2$/SCL2, $P1_3$/SDA1, $P1_4$/SDA2, $P1_5$/I1, $P1_6$/I2/INT3, $P1_7$/S_{IN}	I/O port P1	I/O	Port P1 is an 8-bit I/O port and has basically the same functions as port P0. The output structure of $P1_0$ and $P1_5$–$P1_7$ is CMOS output, that of $P1_1$–$P1_4$ is N-channel open-drain output.
	OSD output	Output	Pins $P1_0$, $P1_5$, $P1_6$ are also used as OSD output pins OUT2, I1, I2 respectively. The output structure is CMOS output.
	Multi-master I^2C-BUS interface	Output	Pins $P1_1$–$P1_4$ are used as SCL1, SCL2, SDA1 and SDA2 respectively, when multi-master I^2C-BUS interface is used. The output structure is N-channel open-drain output.
	Serial I/O data input	Input	$P1_7$ pin is also used as serial I/O data input pin S_{IN}.
$P2_0$–$P2_3$ $P2_4$/AD3–$P2_6$/AD1, $P2_7$	I/O port P2	I/O	Port P2 is an 8-bit I/O port and has basically the same functions as port P0. The output structure is CMOS output.
	Analog input	Input	Pins $P2_4$–$P2_6$ are also used as analog input pins AD3–AD1 respectively.
$P3_0$, $P3_1$	I/O port P3	I/O	Ports $P3_0$ and $P3_1$ are a 2-bit I/O port and has basically the same functions as port P0. The output structure is CMOS output.
$P3_2$	Output port P3	Output	Port $P3_2$ is a 1-bit output port. The output structure is N-channel open-drain output.
$P4_0$/AD4, $P4_1$/INT2, $P4_2$/TIM2, $P4_3$/TIM3, $P4_4$/INT1, $P4_5$/S_{OUT}, $P4_6$/S_{CLK},	Input port P4	Input	Ports $P4_0$–$P4_6$ are a 7-bit input port.
	Analog input	Input	$P4_0$ pin is also used as analog input pin AD4.
	External interrupt input	Input	Pins $P4_1$, $P4_4$ are also used as external interrupt input INT2, INT1.
	External clock input	Input	Pins $P4_2$ and $P4_3$ are also used as external clock input pins TIM2, TIM3 respectively.
	Serial I/O data output	Output	$P4_5$ pin is used as serial I/O data output pin S_{OUT}. The output structure is N-channel open-drain output.
	Serial I/O synchronizing clock input/output	I/O	$P4_6$ pin is used as serial I/O synchronizing clock input/output pin S_{CLK}. The output structure is N-channel open-drain output.
$P4_7$	Output port P4	Output	Port $P4_7$ is a 1-bit output port. The output structure is N-channel open-drain output.
$P5_0$/PWN7, $P5_1$, $P5_2$/R, $P5_3$/G, $P5_4$/B, $P5_5$/OUT1, $P5_6$, $P5_7$	Output port P5	Output	Ports $P5_0$–$P5_7$ are an 8-bit output port. The output structure of $P5_0$, $P5_1$, $P5_6$, $P5_7$ are N-channel open-drain output, that of $P5_2$–$P5_5$ is CMOS output.
	PWM output	Output	$P5_0$ pin is also used as PWM output pin PWM7.
	OSD output	Output	Pins $P5_2$–$P5_5$ are also used as OSD output pins R, G, B, OUT1 respectively.

PIN DESCRIPTION (continued)

$P6_0$–$P6_2$, $P6_5$–$P6_7$	Output port	Output	Ports $P6_0$–$P6_2$, $P6_5$–$P6_7$ are a 6–bit output port. The output structure is N-channel open-drain output.
$P6_3$/OSC1/ X_{CIN}, $P6_4$/OSC2/ X_{COUT}	Input port	Input	Ports $P6_3$ and $P6_4$ are 2-bit input port.
	Clock input for OSD	Input	$P6_3$ pin is also used as OSD clock input pin OSC1.
	Clock output for OSD	Output	$P6_4$ pin is also used as OSD clock output pin OSC2. The output structure is CMOS output.
	Sub-clock output	Output	$P6_4$ pin is also used as sub-clock output pin X_{COUT}. The output structure is CMOS output.
	Sub-clock input	Input	$P6_3$ pin is also used as sub-clock input pin X_{CIN}.
CV_{IN}	I/O for data slicer	Input	Input composite video signal through a capacitor.
V_{HOLD}		Input	Connect a capacitor between V_{HOLD} and V_{SS}.
RVCO			Connect a resistor between RVCO and V_{SS}.
HLF			Connect a filter using of a capacitor and a resistor between HLF and V_{SS}.
H_{SYNC}	H_{SYNC} input	Input	This is a horizontal synchronizing signal input for OSD.
V_{SYNC}	V_{SYNC} input	Input	This is a vertical synchronizing signal input for OSD.

Note : As shown in the memory map (Figure 3), port P0 is accessed as a memory at address $00C0_{16}$ of zero page. Port P0 has the port P0 direction register (address $00C1_{16}$ of zero page) which can be used to program each bit as an input (“0”) or an output (“1”). The pins programmed as “1” in the direction register are output pins. When pins are programmed as “0,” they are input pins. When pins are programmed as output pins, the output data are written into the port latch and then output. When data is read from the output pins, the output pin level is not read but the data of the port latch is read. This allows a previously-output value to be read correctly even if the output “L” voltage has risen, for example, because a light emitting diode was directly driven. The input pins are in the floating state, so the values of the pins can be read. When data is written into the input pin, it is written only into the port latch, while the pin remains in the floating state.

FUNCTIONAL DESCRIPTION

Central Processing Unit (CPU)

The M37270MF-XXXSP uses the standard 740 family instruction set. Refer to the table of 740 family addressing modes and machine instructions or the SERIES 740 <Software> User's Manual for details on the instruction set.

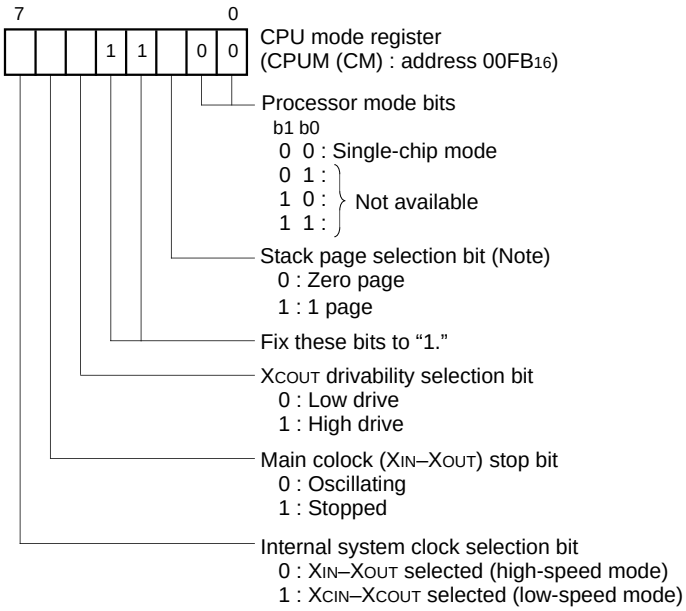
Machine-resident 740 family instructions are as follows:

The FST, SLW instruction cannot be used.

The MUL, DIV, WIT and STP instruction can be used.

CPU Mode Register

The CPU mode register contains the stack page selection bit and internal system clock selection bit. The CPU mode register is allocated at address 00FB16.



Note: Please beware of this bit when programming because it is set to "1" after the reset release.

Fig. 1. Structure of CPU mode register

MEMORY

Special Function Register (SFR) Area

The special function register (SFR) area in the zero page contains control registers such as I/O ports and timers.

RAM

RAM is used for data storage and for stack area of subroutine calls and interrupts.

ROM

ROM is used for storing user programs as well as the interrupt vector area.

RAM for OSD

RAM for display is used for specifying the character codes and colors to display.

ROM for OSD

ROM for display is used for storing character data.

Interrupt Vector Area

The interrupt vector area contains reset and interrupt vectors.

Zero Page

The 256 bytes from addresses 0000_{16} to $00FF_{16}$ are called the zero page area. The internal RAM and the special function registers (SFR) are allocated to this area.

The zero page addressing mode can be used to specify memory and register addresses in the zero page area. Access to this area with only 2 bytes is possible in the zero page addressing mode.

Special Page

The 256 bytes from addresses $FF00_{16}$ to $FFFF_{16}$ are called the special page area. The special page addressing mode can be used to specify memory addresses in the special page area. Access to this area with only 2 bytes is possible in the special page addressing mode.

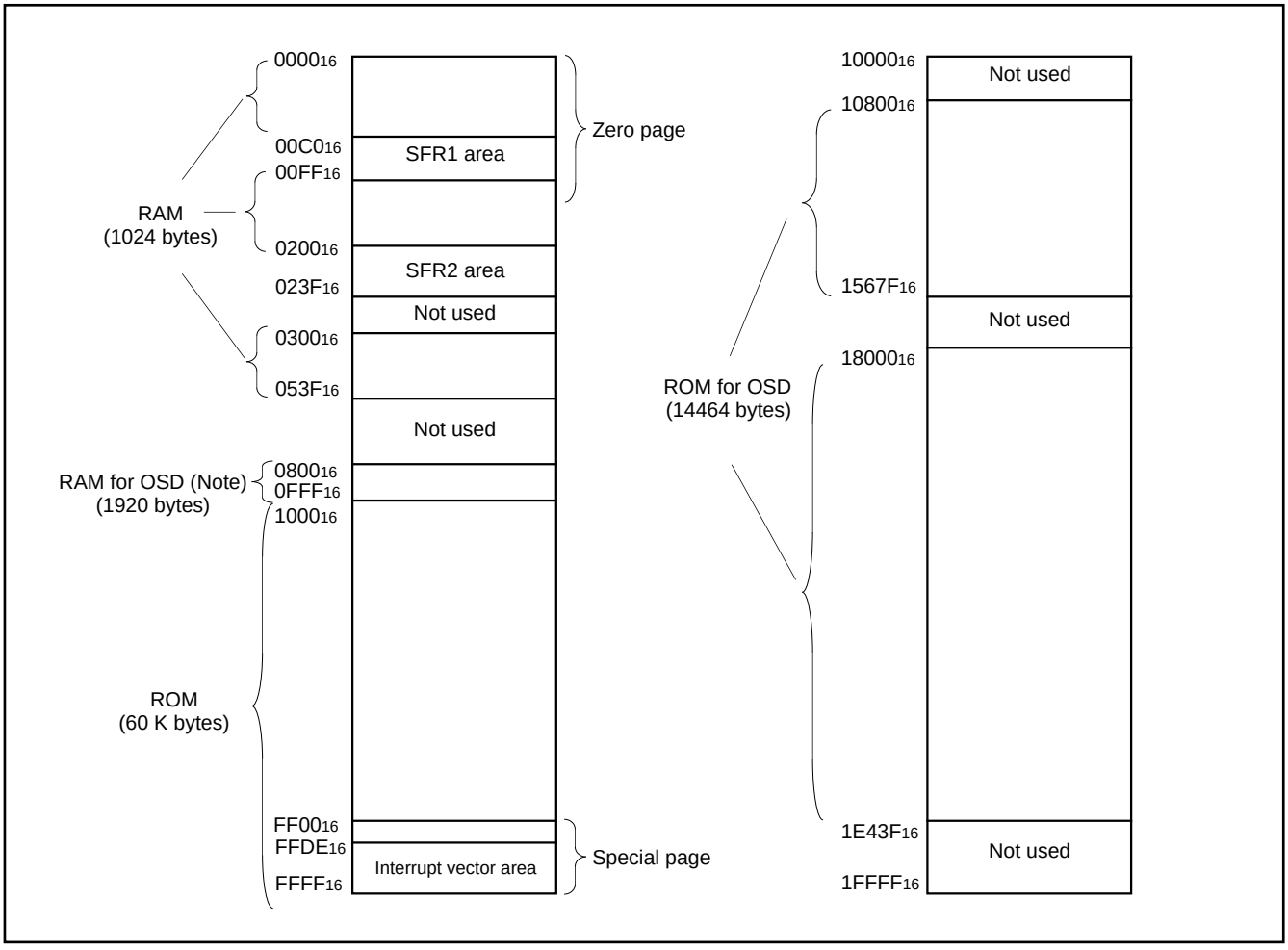


Fig. 2. Memory map

$00C0_{16}$	Port P0	$00E0_{16}$	Caption position register
$00C1_{16}$	Port P0 direction register	$00E1_{16}$	Start bit position register
$00C2_{16}$	Port P1	$00E2_{16}$	Window register
$00C3_{16}$	Port P1 direction register	$00E3_{16}$	Sync slice register
$00C4_{16}$	Port P2	$00E4_{16}$	Data register 1
$00C5_{16}$	Port P2 direction register	$00E5_{16}$	Data register 2
$00C6_{16}$	Port P3	$00E6_{16}$	Clock run-in register 1
$00C7_{16}$	Port P3 direction register	$00E7_{16}$	Clock run-in register 2
$00C8_{16}$	Port P4	$00E8_{16}$	Clock run-in detect register 1
$00C9_{16}$	Port P4 direction register	$00E9_{16}$	Clock run-in detect register 2
$00CA_{16}$	Port P5	$00EA_{16}$	Data slicer control register 1
$00CB_{16}$	OSD port control register	$00EB_{16}$	Data slicer control register 2
$00CC_{16}$	Port P6	$00EC_{16}$	Data register 3
$00CD_{16}$		$00ED_{16}$	Data register 4
$00CE_{16}$	OSD control register	$00EE_{16}$	A-D register
$00CF_{16}$	Horizontal position register	$00EF_{16}$	A-D control register
$00D0_{16}$	Block control register 1	$00F0_{16}$	Timer 1
$00D1_{16}$	Block control register 2	$00F1_{16}$	Timer 2
$00D2_{16}$	Block control register 3	$00F2_{16}$	Timer 3
$00D3_{16}$	Block control register 4	$00F3_{16}$	Timer 4
$00D4_{16}$	Block control register 5	$00F4_{16}$	Timer mode register 1
$00D5_{16}$	Block control register 6	$00F5_{16}$	Timer mode register 2
$00D6_{16}$	Block control register 7	$00F6_{16}$	I^2C data shift register
$00D7_{16}$	Block control register 8	$00F7_{16}$	I^2C address register
$00D8_{16}$	Block control register 9	$00F8_{16}$	I^2C status register
$00D9_{16}$	Block control register 10	$00F9_{16}$	I^2C control register
$00DA_{16}$	Block control register 11	$00FA_{16}$	I^2C clock control register
$00DB_{16}$	Block control register 12	$00FB_{16}$	CPU mode register
$00DC_{16}$	Block control register 13	$00FC_{16}$	Interrupt request register 1
$00DD_{16}$	Block control register 14	$00FD_{16}$	Interrupt request register 2
$00DE_{16}$	Block control register 15	$00FE_{16}$	Interrupt control register 1
$00DF_{16}$	Block control register 16	$00FF_{16}$	Interrupt control register 2

Fig. 3. Memory map of special function register 1 (SFR1)

0200_{16}	PWM0 register	0220_{16}	Vertical register 1_1
0201_{16}	PWM1 register	0221_{16}	Vertical register 1_2
0202_{16}	PWM2 register	0222_{16}	Vertical register 1_3
0203_{16}	PWM3 register	0223_{16}	Vertical register 1_4
0204_{16}	PWM4 register	0224_{16}	Vertical register 1_5
0205_{16}	PWM5 register	0225_{16}	Vertical register 1_6
0206_{16}	PWM6 register	0226_{16}	Vertical register 1_7
0207_{16}	PWM7 register	0227_{16}	Vertical register 1_8
0208_{16}	Clock run-in detect register 3	0228_{16}	Vertical register 1_9
0209_{16}	Clock run-in register 3	0229_{16}	Vertical register 1_{10}
$020A_{16}$	PWM mode register 1	$022A_{16}$	Vertical register 1_{11}
$020B_{16}$	PWM mode register 2	$022B_{16}$	Vertical register 1_{12}
$020C_{16}$	Timer 5	$022C_{16}$	Vertical register 1_{13}
$020D_{16}$	Timer 6	$022D_{16}$	Vertical register 1_{14}
$020E_{16}$		$022E_{16}$	Vertical register 1_{15}
$020F_{16}$	Sync pulse counter register	$022F_{16}$	Vertical register 1_{16}
0210_{16}	Data slicer control register 3	0230_{16}	Vertical register 2_1
0211_{16}	Interrupt interval determination register	0231_{16}	Vertical register 2_2
0212_{16}	Interrupt interval determination control register	0232_{16}	Vertical register 2_3
0213_{16}	Serial I/O mode register	0233_{16}	Vertical register 2_4
0214_{16}	Serial I/O register	0234_{16}	Vertical register 2_5
0215_{16}		0235_{16}	Vertical register 2_6
0216_{16}	Clock source control register	0236_{16}	Vertical register 2_7
0217_{16}	I/O polarity control register	0237_{16}	Vertical register 2_8
0218_{16}	Raster color register	0238_{16}	Vertical register 2_9
0219_{16}	Extra font color register	0239_{16}	Vertical register 2_{10}
$021A_{16}$		$023A_{16}$	Vertical register 2_{11}
$021B_{16}$	Border color register	$023B_{16}$	Vertical register 2_{12}
$021C_{16}$	Window H register 1	$023C_{16}$	Vertical register 2_{13}
$021D_{16}$	Window L register 1	$023D_{16}$	Vertical register 2_{14}
$021E_{16}$	Window H register 2	$023E_{16}$	Vertical register 2_{15}
$021F_{16}$	Window L register 2	$023F_{16}$	Vertical register 2_{16}

Fig. 4. Memory map of special function register 2 (SFR2)

INTERRUPTS

Interrupts can be caused by 18 different sources consisting of 4 external, 12 internal, 1 software, and reset. Interrupts are vectored interrupts with priorities shown in Table 1. Reset is also included in the table because its operation is similar to an interrupt.

When an interrupt is accepted,

(1) The contents of the program counter and processor status register are automatically stored into the stack.
(2) The interrupt disable flag I is set to "1" and the corresponding interrupt request bit is set to "0."
(3) The jump destination address stored in the vector address enters the program counter.

Other interrupts are disabled when the interrupt disable flag is set to "1."

All interrupts except the BRK instruction interrupt have an interrupt request bit and an interrupt enable bit. The interrupt request bits are in interrupt request registers 1 and 2 and the interrupt enable bits are in interrupt control registers 1 and 2. Figure 5 shows the structure of the interrupt-related registers.

Interrupts other than the BRK instruction interrupt and reset are accepted when the interrupt enable bit is "1," interrupt request bit is "1," and the interrupt disable flag is "0." The interrupt request bit can be set to "0" by a program, but not set to "1." The interrupt enable bit can be set to "0" and "1" by a program.

Reset is treated as a non-maskable interrupt with the highest priority. Figure 6 shows interrupt control.

Interrupt Causes

(1) VSYNC and OSD interrupts
The VSYNC interrupt is an interrupt request synchronized with the vertical sync signal.
The OSD interrupt occurs after character block display to the CRT is completed.

(2) INT1, INT2, INT3 interrupts
With an external interrupt input, the system detects that the level of a pin changes from "L" to "H" or from "H" to "L," and generates an interrupt request. The input active edge can be selected by bits 3, 4 and 6 of the interrupt interval determination control register (address 0212_{16}) : when this bit is "0," a change from "L" to "H" is detected; when it is "1," a change from "H" to "L" is detected. Note that all bits are cleared to "0" at reset.

(3) Timer 1, 2, 3 and 4 interrupts
An interrupt is generated by an overflow of timer 1, 2, 3 or 4.

(4) Serial I/O interrupt
This is an interrupt request from the clock synchronous serial I/O function.

(5) f(XIN)/4096 interrupt
This interrupt occurs regularly with a f(XIN)/4096 period. Set bit 0 of the PWM mode register 1 to "0."

(6) Data slicer interrupt
An interrupt occurs when slicing data is completed.

(7) Multi-master I^2C-BUS interface interrupt
This is an interrupt request related to the multi-master I^2C-BUS interface.

(8) A-D conversion interrupt
An interrupt occurs at the completion of A-D conversion. Since A-D conversion interrupt and the INT3 interrupt share the same vector, an interrupt source is selected by bit 7 of the interrupt interval determination control register (address 0212_{16}).

Table 1. Interrupt vector addresses and priority

Interrupt source	Priority	Vector addresses	Remarks
Reset	1	$FFFF_{16}$, $FFFE_{16}$	Non-maskable
OSD interrupt	2	$FFFD_{16}$, $FFFC_{16}$	
INT1 interrupt	3	$FFFB_{16}$, $FFFA_{16}$	Active edge selectable
Data slicer interrupt	4	$FFF9_{16}$, $FFF8_{16}$	
Serial I/O interrupt	5	$FFF7_{16}$, $FFF6_{16}$	
Timer 4 interrupt	6	$FFF5_{16}$, $FFF4_{16}$	
f(XIN)/4096 interrupt	7	$FFF3_{16}$, $FFF2_{16}$	
VSYNC interrupt	8	$FFF1_{16}$, $FFF0_{16}$	Active edge selectable
Timer 3 interrupt	9	$FFEF_{16}$, $FFEE_{16}$	
Timer 2 interrupt	10	$FFED_{16}$, $FFEC_{16}$	
Timer 1 interrupt	11	$FFEB_{16}$, $FFEA_{16}$	
A-D convertion · INT3 interrupt	12	$FFE9_{16}$, $FFE8_{16}$	Active edge selectable
INT2 interrupt	13	$FFE7_{16}$, $FFE6_{16}$	Active edge selectable
Multi-master I^2C-BUS interface interrupt	14	$FFE5_{16}$, $FFE4_{16}$	
Timer 5 · 6 interrupt	15	$FFE3_{16}$, $FFE2_{16}$	
BRK instruction interrupt	16	$FFDF_{16}$, $FFDE_{16}$	Non-maskable (software interrupt)

(9)Timer 5 · 6 interrupt

An interrupt is generated by an overflow of timer 5 or 6. Their priorities are same, and can be switched by software.

(10)BRK instruction interrupt

This software interrupt has the least significant priority. It does not have a corresponding interrupt enable bit, and it is not affected by the interrupt disable flag I (non-maskable).

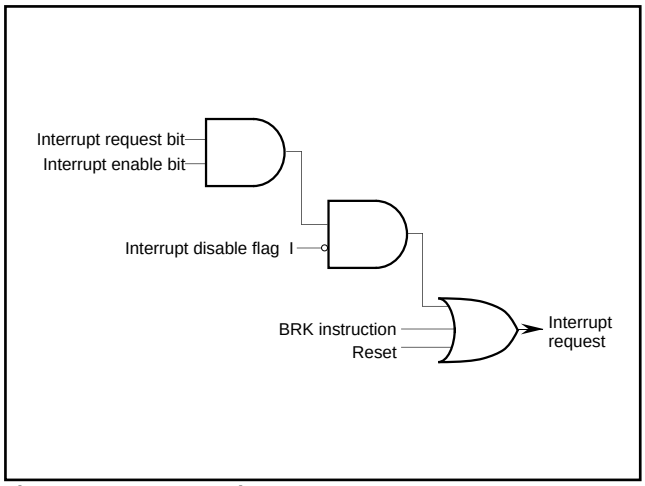


Fig. 6. Interrupt control

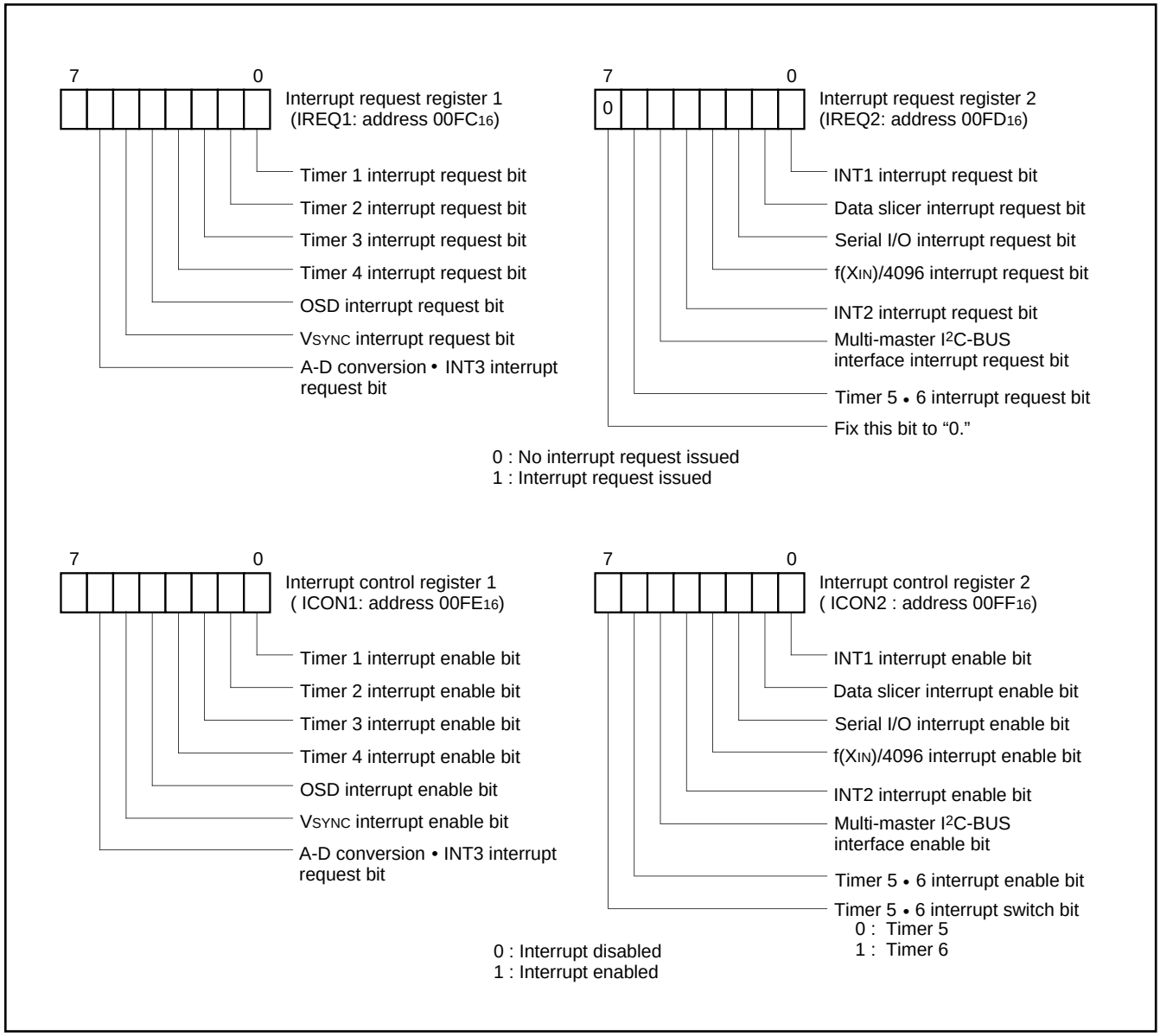


Fig. 5. Structure of interrupt-related registers

TIMERS

The M37270MF-XXXSP has 6 timers: timer 1, timer 2, timer 3, timer 4, timer 5, and timer 6. All timers are 8-bit timers with the 8-bit timer latch. The timer block diagram is shown in Figure 8.

All of the timers count down and their divide ratio is 1/(n+1), where n is the value of timer latch. The value is set to a timer at the same time by writing a count value to the corresponding timer latch (addresses $00F0_{16}$ to $00F3_{16}$: timers 1 to 4, addresses $020C_{16}$ and $020D_{16}$: timers 5 and 6).

The count value is decremented by 1. The timer interrupt request bit is set to "1" by a timer overflow at the next count pulse after the count value reaches "00_{16}".

(1) Timer 1

Timer 1 can select one of the following count sources:

- f(XIN)/16 or f(XCIN)/16
- f(XIN)/4096 or f(XCIN)/4096
- External clock from the P42/TIM2 pin

The count source of timer 1 is selected by setting bits 5 and 0 of the timer mode register 1 (address $00F4_{16}$). Either f(XIN) or f(XCIN) is selected by bit 7 of the CPU mode register.

Timer 1 interrupt request occurs at timer 1 overflow.

(2) Timer 2

Timer 2 can select one of the following count sources:

- f(XIN)/16 or f(XCIN)/16
- Timer 1 overflow signal
- External clock from the P42/TIM2 pin

The count source of timer 2 is selected by setting bits 4 and 1 of the timer mode register 1 (address $00F4_{16}$). Either f(XIN) or f(XCIN) is selected by bit 7 of the CPU mode register. When timer 1 overflow signal is a count source for the timer 2, the timer 1 functions as an 8-bit prescaler.

Timer 2 interrupt request occurs at timer 2 overflow.

(3) Timer 3

Timer 3 can select one of the following count sources:

- f(XIN)/16 or f(XCIN)/16
- f(XCIN)
- External clock from the P43/TIM3 pin

The count source of timer 3 is selected by setting bit 0 of the timer mode register 2 (address $00F5_{16}$) and bit 6 at address $00C7_{16}$. Either f(XIN) or f(XCIN) is selected by bit 7 of the CPU mode register.

Timer 3 interrupt request occurs at timer 3 overflow.

(4) Timer 4

Timer 4 can select one of the following count sources:

- f(XIN)/16 or f(XCIN)/16
- f(XIN)/2 or f(XCIN)/2
- f(XCIN)

The count source of timer 3 is selected by setting bits 4 and 1 of the timer mode register 2 (address $00F5_{16}$). Either f(XIN) or f(XCIN) is selected by bit 7 of the CPU mode register. When timer 3 overflow signal is a count source for the timer 4, the timer 3 functions as an 8-bit prescaler.

Timer 4 interrupt request occurs at timer 4 overflow.

(5) Timer 5

Timer 5 can select one of the following count sources:

- f(XIN)/16 or f(XCIN)/16
- Timer 2 overflow signal
- Timer 4 overflow signal

The count source of timer 3 is selected by setting bit 6 of the timer mode register 1 (address $00F4_{16}$) and bit 7 of the timer mode register 2 (address $00F5_{16}$). Either f(XIN) or f(XCIN) is selected by bit 7 of the CPU mode register.

Timer 5 interrupt request occurs at timer 5 overflow.

(6) Timer 6

Timer 6 can select one of the following count sources:

- f(XIN)/16 or f(XCIN)/16
- Timer 5 overflow signal

The count source of timer 6 is selected by setting bit 7 of the timer mode register 1 (address $00F4_{16}$). Either f(XIN) or f(XCIN) is selected by bit 7 of the CPU mode register. When timer 5 overflow signal is a count source for the timer 6, the timer 5 functions as an 8-bit prescaler.

Timer 6 interrupt request occurs at timer 6 overflow.

At reset, timers 3 and 4 are connected by hardware and "FF_{16}" is automatically set in timer 3; "07_{16}" in timer 4. The f(XIN) * /16 is selected as the timer 3 count source. The internal reset is released by timer 4 overflow at these state, the internal clock is connected.

At execution of the STP instruction, timers 3 and 4 are connected by hardware and "FF_{16}" is automatically set in timer 3; "07_{16}" in timer 4. However, the f(XIN) * /16 is not selected as the timer 3 count source. So set both bit 0 of the timer mode register 2 (address $00F5_{16}$) and bit 6 at address $00C7_{16}$ to "0" before the execution of the STP instruction (f(XIN) * /16 is selected as the timer 3 count source). The internal STP state is released by timer 4 overflow at these state, the internal clock is connected.

Because of this, the program starts with the stable clock.

* : When bit 7 of the CPU mode register (CM7) is "1," f(XIN) becomes f(XCIN).

The structure of timer-related registers is shown in Figure 7.

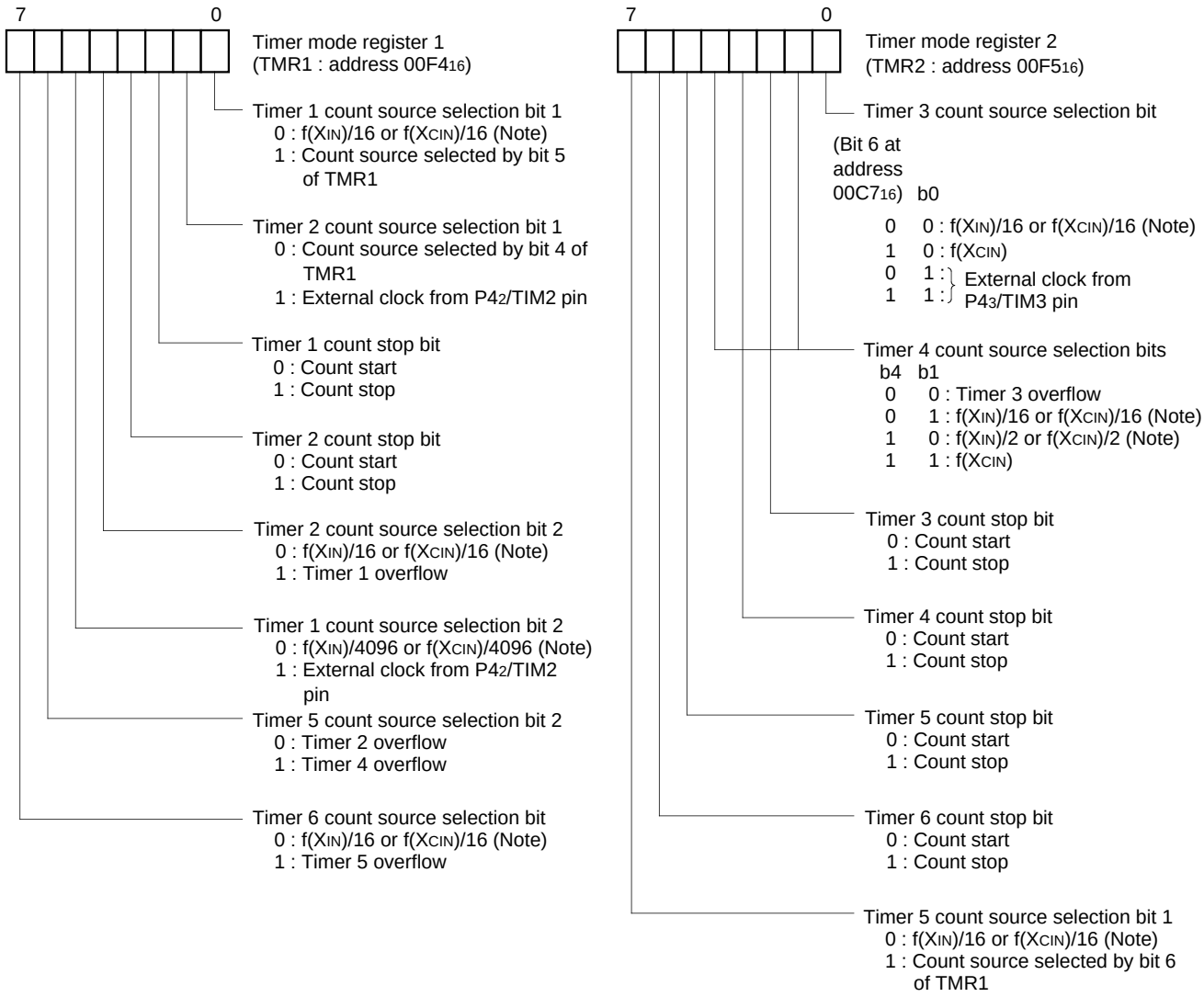
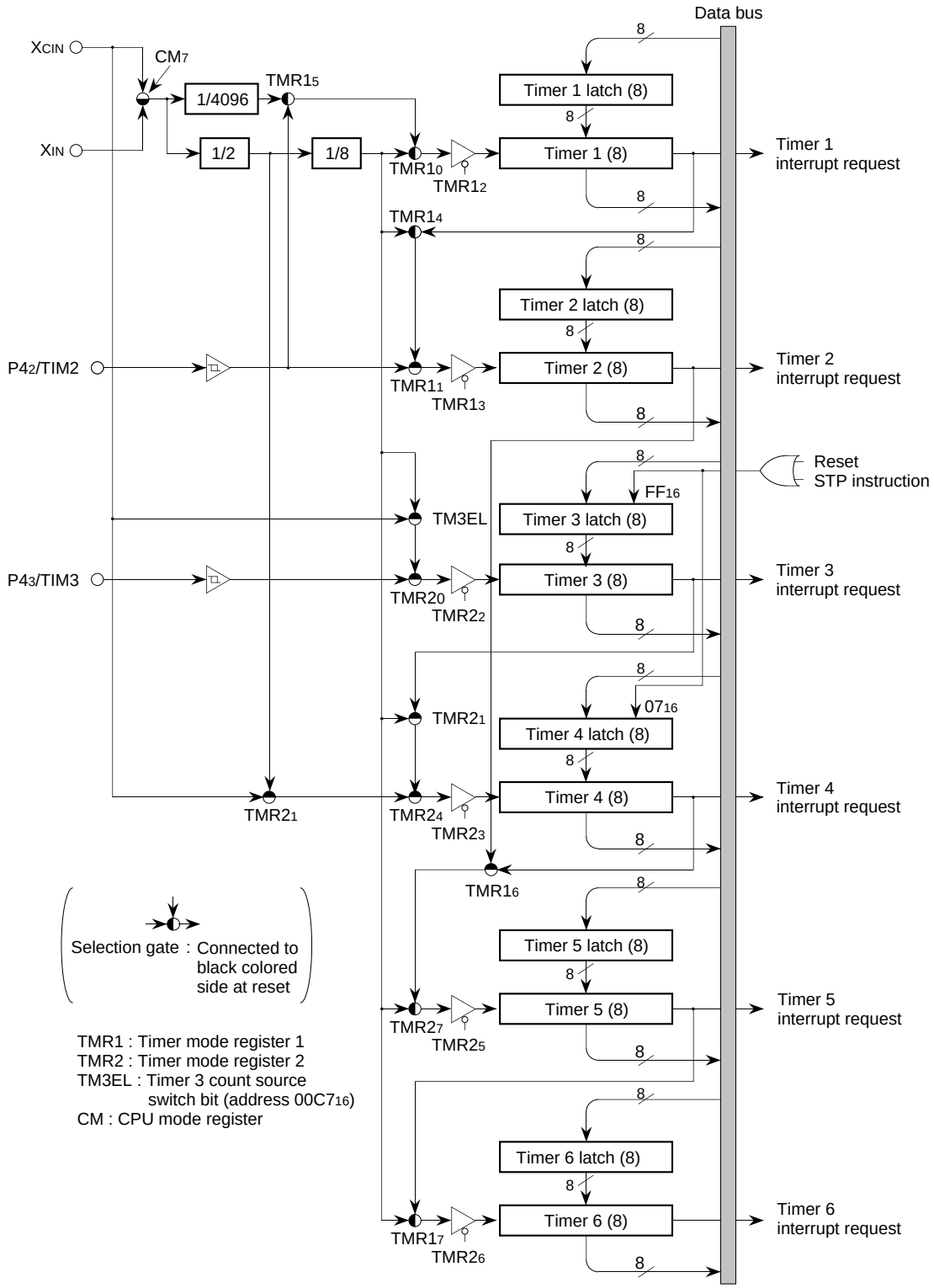


Fig. 7. Structure of timer-related registers



Notes 1: "H" pulse width of external clock inputs TIM2 and TIM3 needs 4 machine cycles or more.

2: When the external clock source is selected, timers 1, 2, and 3 are counted at a rising edge of input signal.

3: In the stop mode or the wait mode, external clock inputs TIM2 and TIM3 cannot be used.

Fig. 8. Timer block diagram

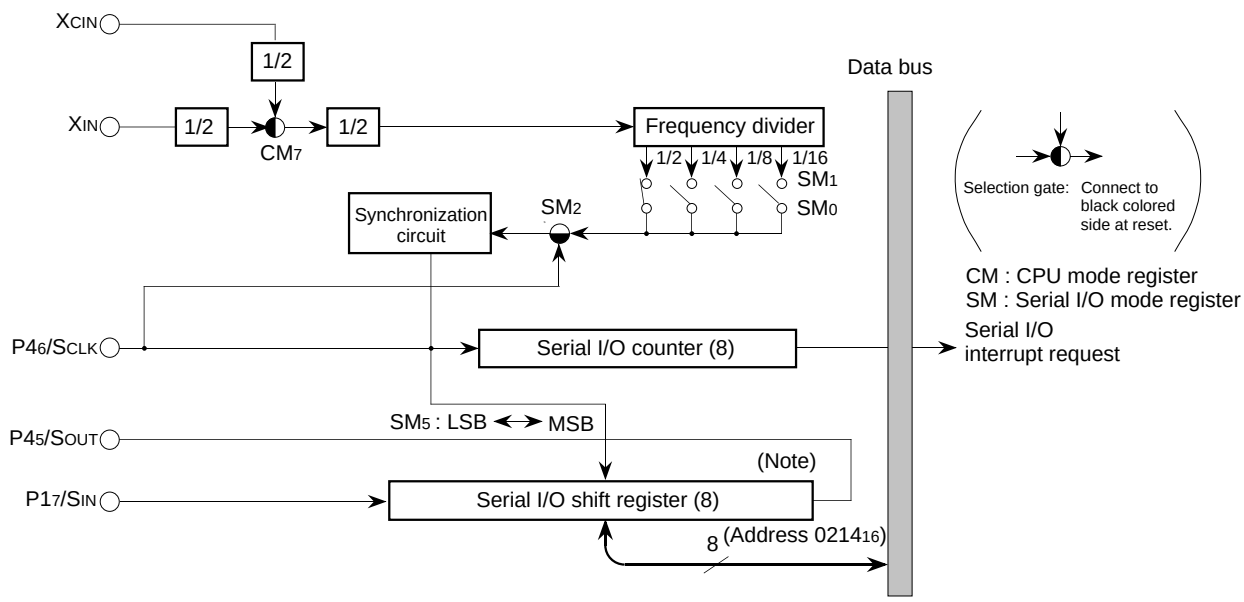
SERIAL I/O

The M37270MF-XXXSP has a built-in serial I/O which can either transmit or receive 8-bit data in serial in the clock synchronous mode.

The serial I/O block diagram is shown in Figure 9. The synchronizing clock I/O pin (SCLK), and data output pin (SOUT) also function as port P4, data input pin (SIN) also functions as port P1.

Bit 2 of the serial I/O mode register (address 0213_{16}) selects whether the synchronizing clock is supplied internally or externally (from the P46/SCLK pin). When an internal clock is selected, bits 1 and 0 select whether f(XIN) or f(XCIN) is divided by 8, 16, 32, or 64. To use P45/SOUT and P46/SCLK pins for serial I/O, set the corresponding bits of the port P4 direction register (address $00C9_{16}$) to "0." To use P17/SIN pin for serial I/O, set the corresponding bit of the port P1 direction register (address $00C3_{16}$) to "0."

The operation of the serial I/O function is described below. The function of the serial I/O differs depending on the clock source; external clock or internal clock.



Note : When the data is set in the serial I/O register (address 0214_{16}), the register functions as the serial I/O shift register.

Fig. 9. Serial I/O block diagram

Internal clock—the serial I/O counter is set to "7" during write cycle into the serial I/O register (address 0214_{16}), and transfer clock goes "H" forcibly. At each falling edge of the transfer clock after the write cycle, serial data is output from the SOUT pin. Transfer direction can be selected by bit 5 of the serial I/O mode register. At each rising edge of the transfer clock, data is input from the SIN pin and data in the serial I/O register is shifted 1 bit.

After the transfer clock has counted 8 times, the serial I/O counter becomes "0" and the transfer clock stops at "H." At this time the interrupt request bit is set to "1."

External clock—when an external clock is selected as the clock source, the interrupt request is set to "1" after the transfer clock has counted 8 times. However, transfer operation does not stop, so control the clock externally. Use the external clock of 500kHz or less with a duty cycle of 50%.

The serial I/O timing is shown in Figure 10. When using an external clock for transfer, the external clock must be held at "H" for initializing the serial I/O counter. When switching between an internal clock and an external clock, do not switch during transfer. Also, be sure to initialize the serial I/O counter after switching.

Notes 1: On programming, note that the serial I/O counter is set by writing to the serial I/O register with the bit managing instructions as SEB and CLB instructions.

2: When an external clock is used as the synchronizing clock, write transmit data to the serial I/O register at "H" of the transfer clock input level.

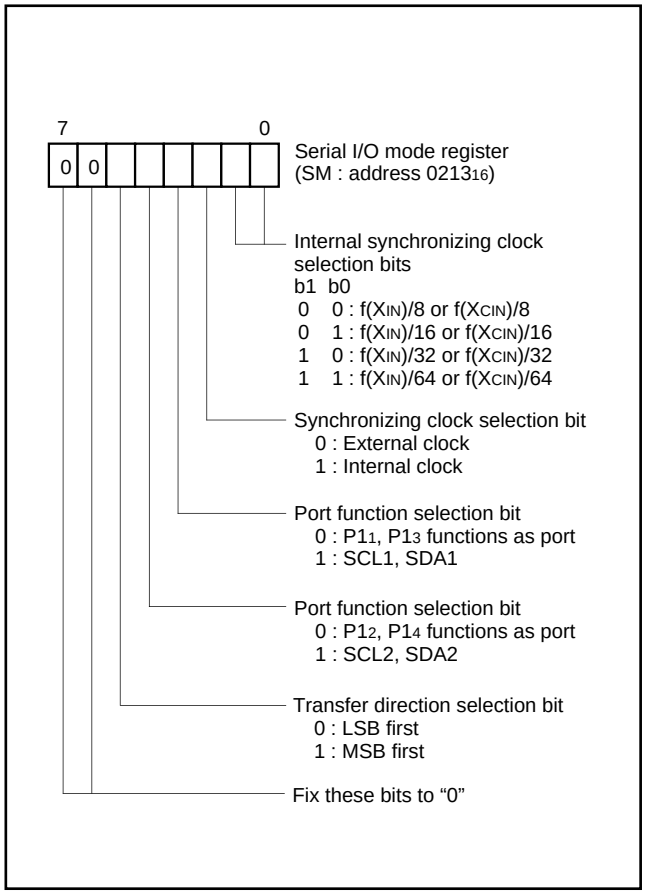
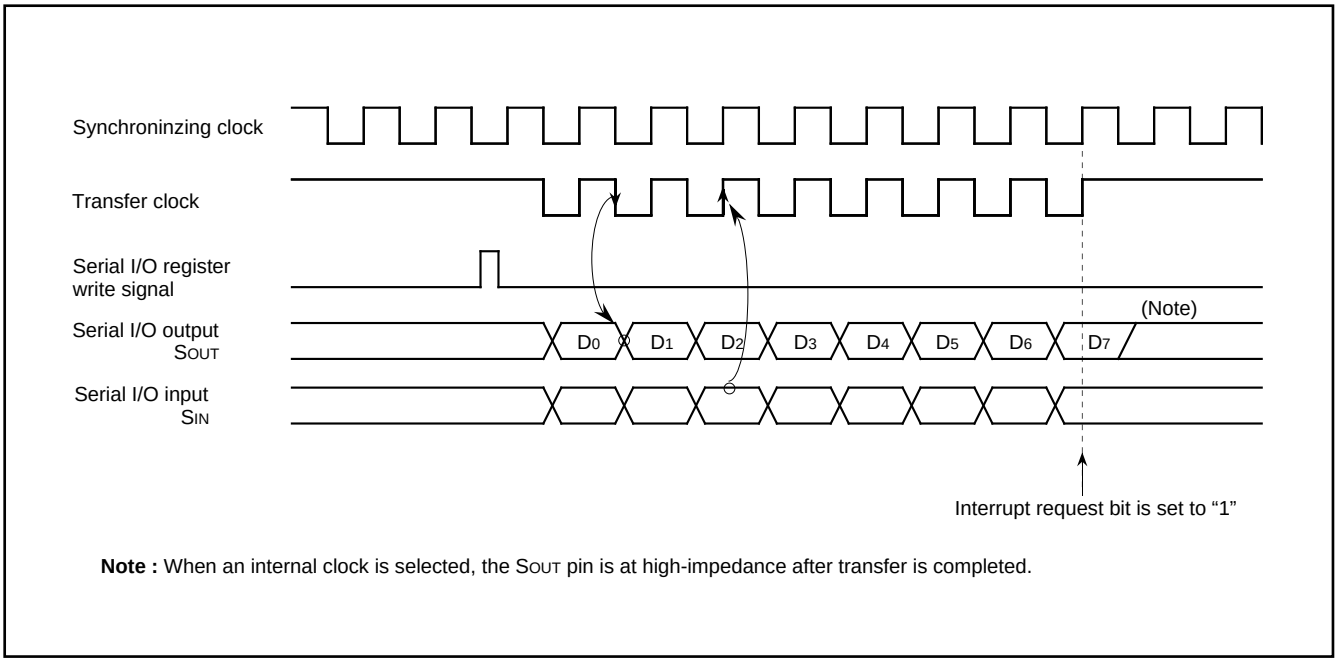


Fig. 11. Structure of serial I/O mode register



Note : When an internal clock is selected, the SOUT pin is at high-impedance after transfer is completed.

Fig. 10. Serial I/O timing (for LSB first)

PWM OUTPUT FUNCTION

The M37270MF-XXXSP is equipped with eight 8-bit PWMs (PWM0–PWM7). PWM0–PWM7 have the same circuit structure and an 8-bit resolution with minimum resolution bit width of 4µs (for f(X_{IN}) = 8 MHz) and repeat period of 1024µs.

Figure 12 shows the PWM block diagram. The PWM timing generating circuit applies individual control signals to PWM0–PWM7 using f(X_{IN}) divided by 2 as a reference signal.

(1) Data Setting

When outputting PWM0–PWM7, set 8-bit output data in the PWMi register (i means 0 to 7; addresses 0200_{16} to 0207_{16}).

(2) Transmitting Data from Register to PWM circuit

Data transfer from the 8-bit PWM register to 8-bit PWM circuit is executed at writing data to the register.

The signal output from the 8-bit PWM output pin corresponds to the contents of this register.

(3) Operating of 8-bit PWM

The following is the explanation about PWM operation.

At first, set the bit 0 of PWM mode register 1 (address $020A_{16}$) to "0" (at reset, bit 0 is already set to "0" automatically), so that the PWM count source is supplied.

PWM0–PWM3 are also used as pins $P0_4$–$P0_7$, PWM4–PWM6 are also used as pins $P0_0$–$P0_2$, PWM7 is also used as pins $P5_0$, respectively. Set the corresponding bits of the port P0 direction register to "1" (output mode). And select each output polarity by bit 3 of the PWM mode register 1 (address $020A_{16}$). Then, set bits 7 to 0 of the PWM output control register 2 to "1" (PWM output).

The PWM waveform is output from the PWM output pins by setting these registers.

Figure 13 shows the 8-bit PWM timing. One cycle (T) is composed of 256 (2^8) segments. The 8 kinds of pulses relative to the weight of each bit (bits 0 to 7) are output inside the circuit during 1 cycle. Refer to Figure 13 (a). The 8-bit PWM outputs waveform which is the logical sum (OR) of pulses corresponding to the contents of bits 0 to 7 of the 8-bit PWM register. Several examples are shown in Figure 13 (b). 256 kinds of output ("H" level area: 0/256 to 255/256) are selected by changing the contents of the PWM register. A length of entirely "H" output cannot be output, i.e. 256/256.

(4) Output after Reset

At reset, the output of ports $P0_0$–$P0_2$ and $P0_4$–$P0_7$ is in the high-impedance state, port $P5_0$ outputs "L," and the contents of the PWM register and the PWM circuit are undefined. Note that after reset, the PWM output is undefined until setting the PWM register.

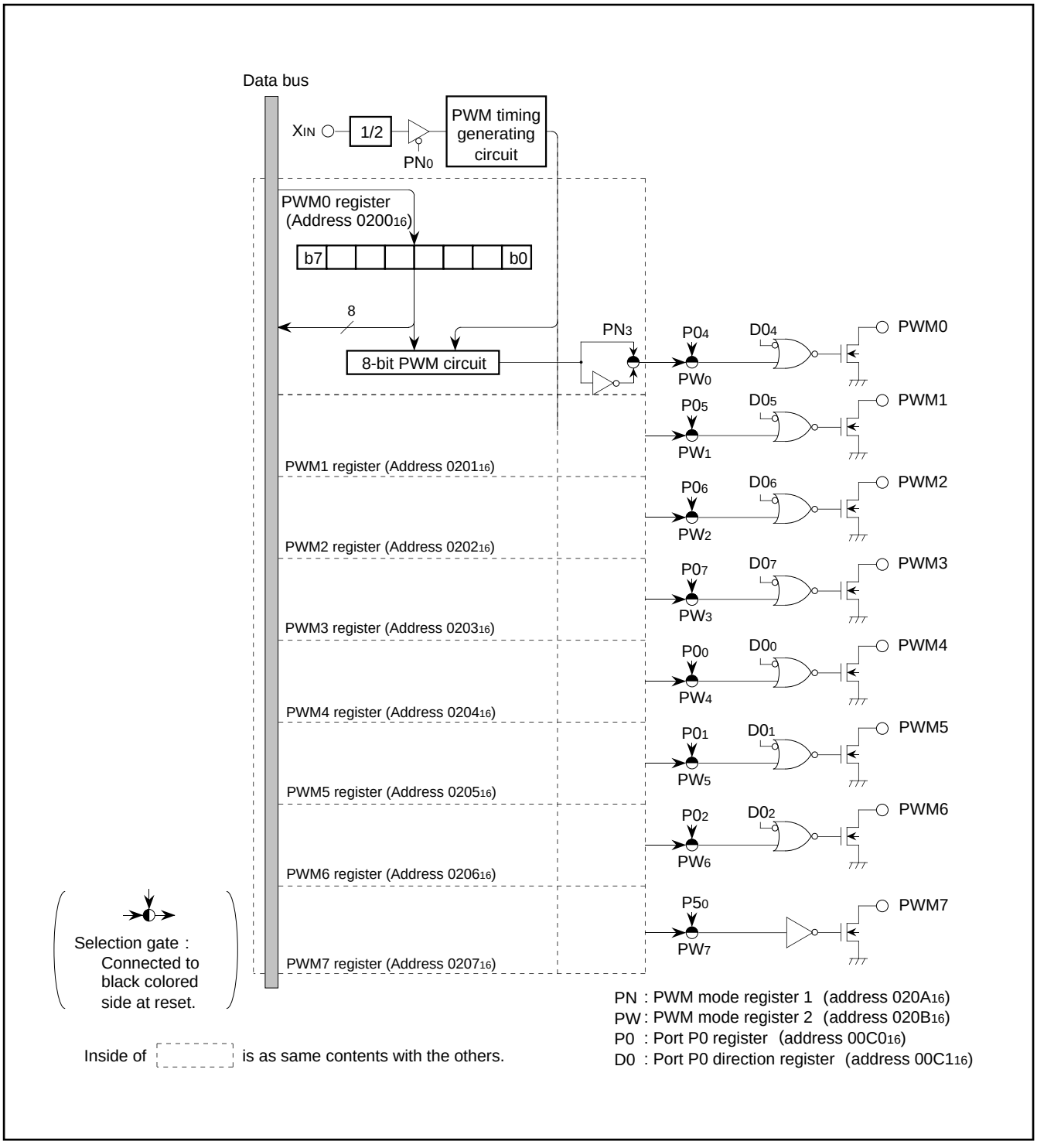


Fig. 12. PWM block diagram

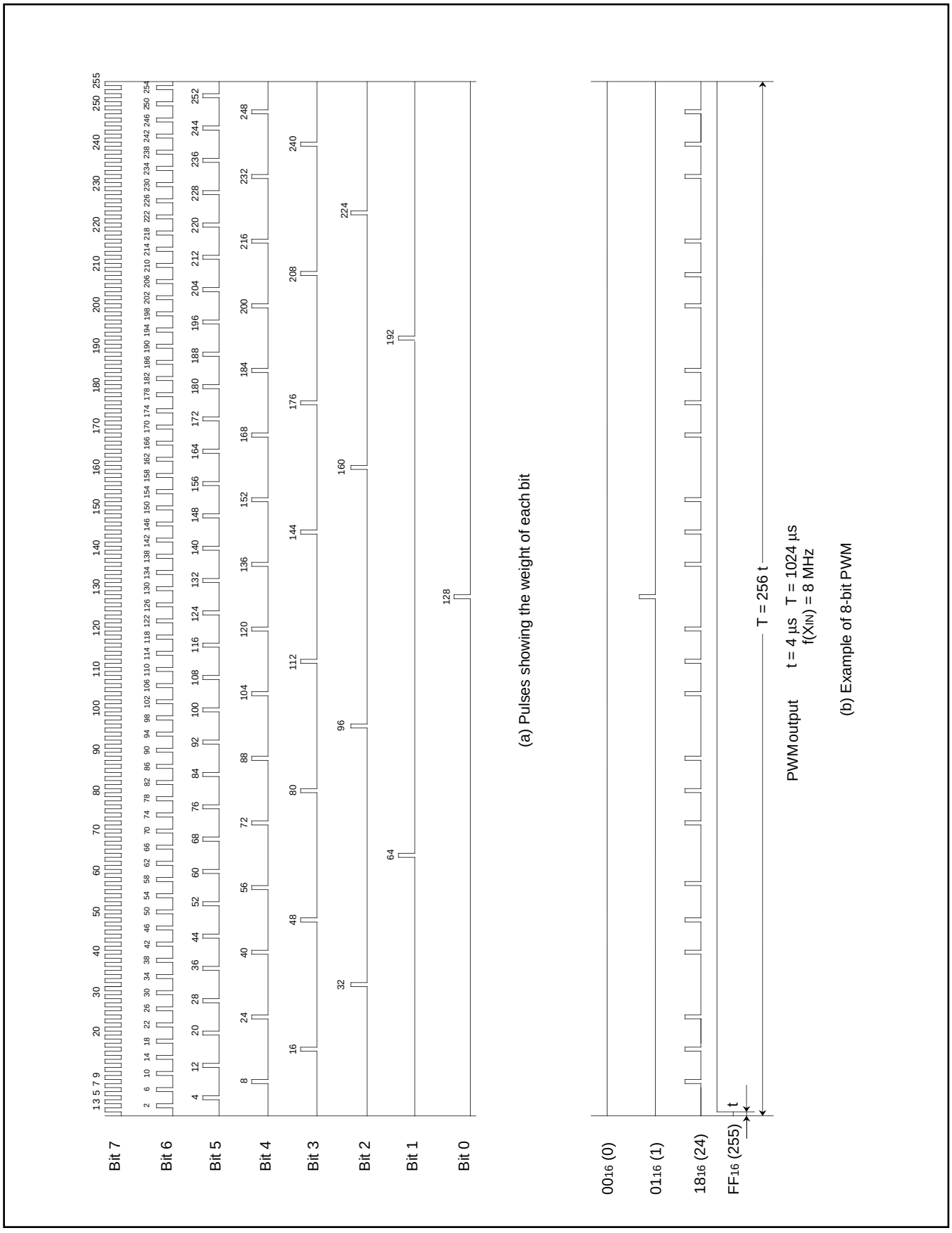


Fig. 13. 8-bit PWM timing

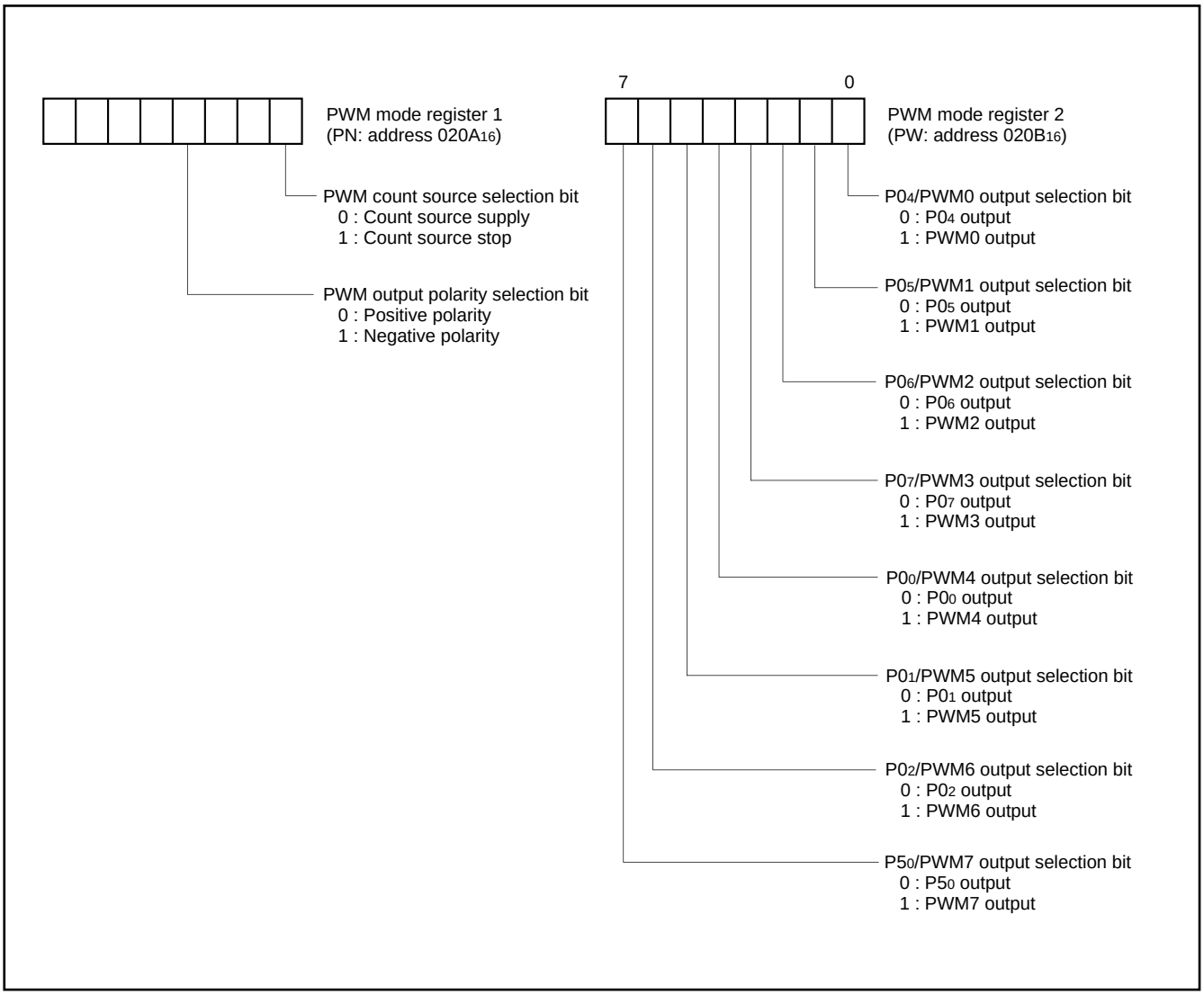


Fig. 14. Structure of PWM-related registers

A-D CONVERTER

(1)A-D Conversion Register (AD)

A-D conversion reigister is a read-only register that stores the result of an A-D conversion. This register should not be read during A-D conversion.

(2)A-D Control Register (ADCON)

The A-D control register controls A-D conversion. Bits 1 and 0 of this register select analog input pins. When these pins are not used as anlog input pins, they are used as ordinary I/O pins. Bit 3 is the A-D conversion completion bit, A-D conversion is started by writing "0" to this bit. The value of this bit remains at "0" during an A-D conversion, then changes to "1" when the A-D conversion is completed.
Bit 4 controls connection between the resistor ladder and V_{CC}. When not using the A-D converter, the resistor ladder can be cut off from the internal V_{CC} by setting this bit to "0." This can realize the low-power dissipation.

(3)Comparison Voltage Generator (Resistor Ladder)

The voltage generator divides the voltage between V_{SS} and V_{CC} by 256, and outputs the divided voltages to the comparator as the reference voltage V_{ref}.

(4)Channel Selector

The channel selector connects an analog input pin selected by bits 1 and 0 of the A-D control register to the comparator.

(5)Comparator and Control Circuit

The conversion result of the analog input voltage and the reference voltage "V_{ref}" is stored in the A-D conversion register. The A-D conversion completion bit and A-D conversion interrupt request bit are set to "1" at the completion of A-D conversion.

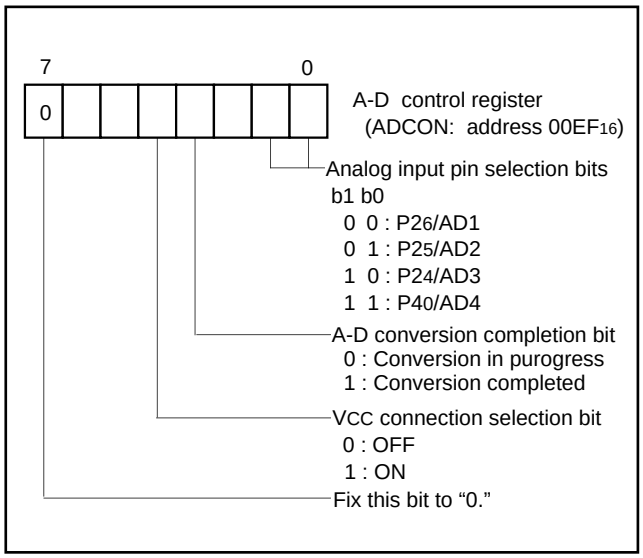


Fig. 15. Structure of A-D control register

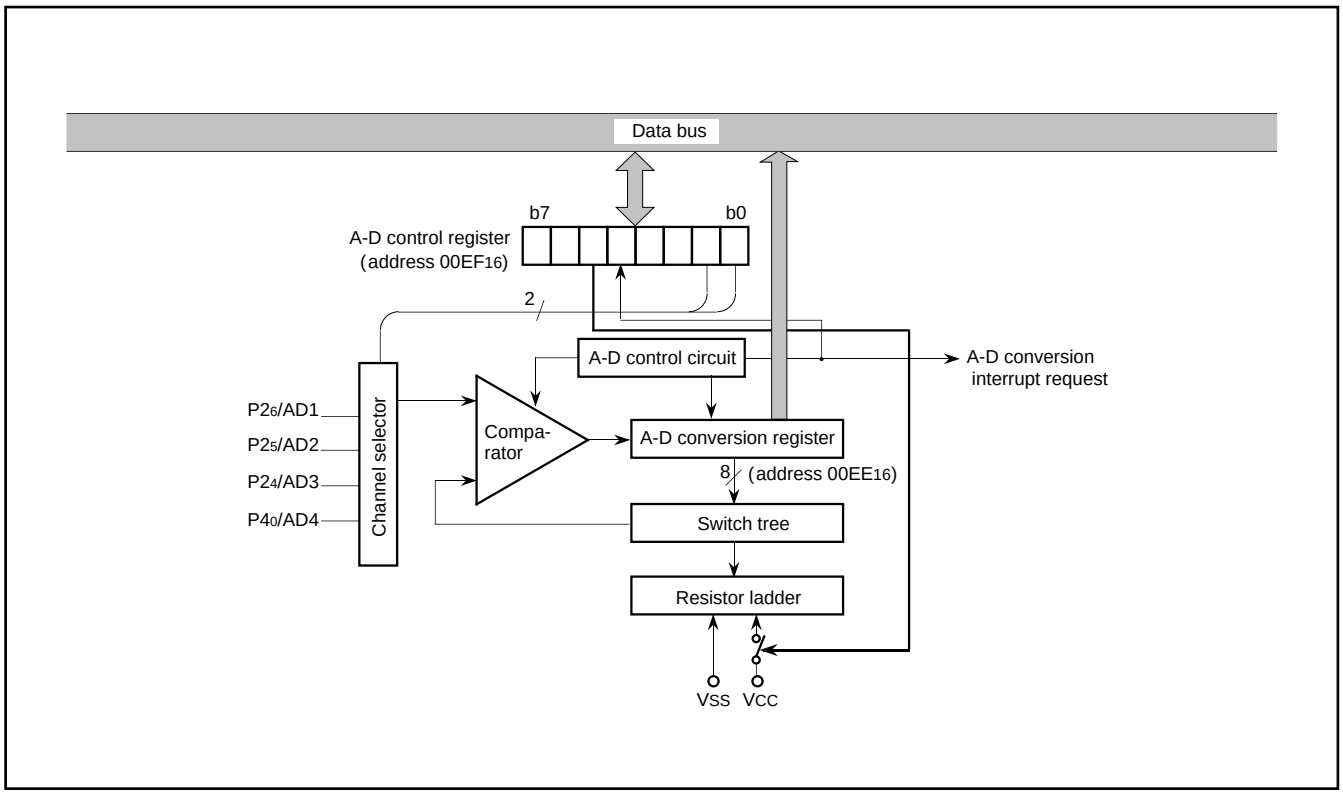


Fig. 16. A-D comparator block diagram

(6) Conversion Method

① Set bit 7 of the interrupt interval determination control register (address 0212_{16}) to "1" to generate an interrupt request at completion of A-D conversion.
② Set the A-D conversion • INT3 interrupt request bit to "0" (even when A-D conversion is started, the A-D conversion • INT3 interrupt bit is not set to "0" automatically).
③ When using A-D conversion interrupt, enable interrupts by setting A-D conversion • INT3 interrupt request bit to "1" and setting the interrupt disable flag to "0."
④ Set the V_{CC} connection selection bit to "1" to connect V_{CC} to the resistor ladder.
⑤ Select analog input pins by setting the analog input selection bit of the A-D control register.
⑥ Set the A-D conversion completion bit to "0." This write operation starts the A-D conversion. Do not read the A-D conversion register during the A-D conversion.
⑦ Verify the completion of the conversion by the state ("1") of the A-D conversion completion bit, that ("1") of A-D conversion • INT3 interrupt bit, or the occurrence of an A-D conversion interrupt.
⑧ Read the A-D conversion register to obtain the conversion results.

Note : When the ladder resistor is disconnect from V_{CC}, set the V_{CC} connection selection bit to "0" between steps ⑦ and ⑧.

(7) Internal Operation

At the time when the A-D conversion starts, the following operations are automatically performed.

① The A-D conversion register is set to "00_{16}."
② The most significant bit of the A-D conversion register becomes "1, " and the comparison voltage "V_{ref}" is input to the comparator. At this point, V_{ref} is compared with the analog input voltage "V_{IN} ."
③ Bit 7 is determined by the comparison result as follows.
When $V_{ref} < V_{IN}$: bit 7 holds "1"
When $V_{ref} > V_{IN}$: bit 7 becomes "0"

With the above operations, the analog value is converted into a digital value. The A-D conversion terminates in a maximum 50 machine cycles (12.5µs at $f(X_{IN})$ = 8 MHz) after it starts, and the conversion result is stored in the A-D conversion register.

An A-D conversion interrupt request occurs at the same time of A-D conversion completion, the A-D conversion • INT3 interrupt request bit becomes "1." The A-D conversion completion bit also becomes "1."

Table 2. Expression for V_{ref} and V_{REF}

A-D conversion register contents "n" (decimal notation)	V_{ref} (V)
0	0
1 to 255	$\frac{V_{REF}}{256} \times (n - 0.5)$

Note: V_{REF} indicates the voltage of internal V_{CC}.

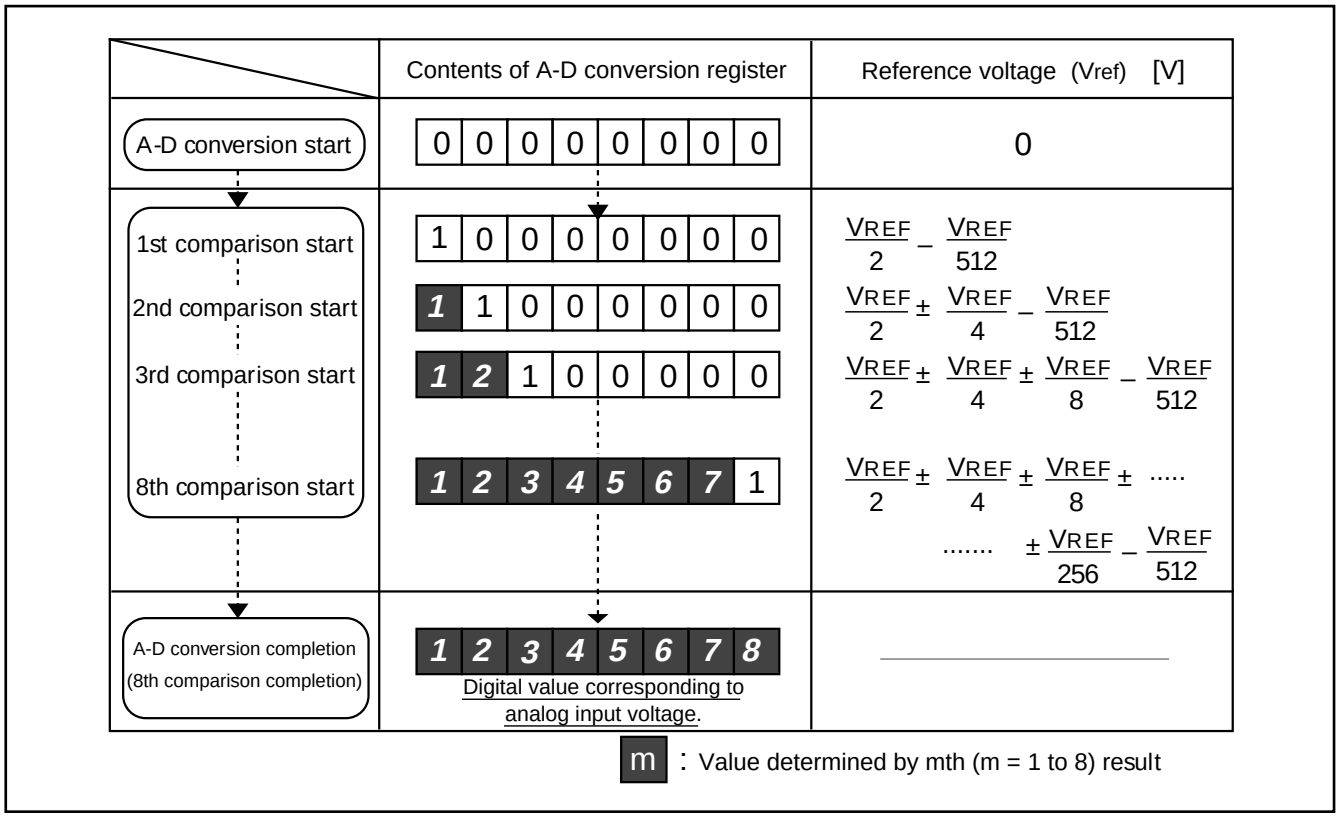


Fig. 17. Changes in A-D conversion register and comparison voltage during A-D conversion

(8) Definition of A-D Conversion Accuracy

The definition of A-D conversion accuracy is described below.

① Relative accuracy

- **Zero transition error (V_{0T})**

The deviation of the input voltage at which A-D conversion output data changes from "0" to "1," from the corresponding ideal A-D conversion characteristics between 0 and V_{REF}.

$$V_{0T} = \frac{(V_0 - 1/2 \times V_{REF}/256)}{1LSB} \quad [LSB]$$

- **Full-scale transition error (V_{FST})**

The deviation of the input voltage at which A-D conversion output data changes from "255" to "254," from the corresponding ideal A-D conversion characteristics between 0 and V_{REF}.

$$V_{FST} = \frac{(V_{REF} - 3/2 \times V_{REF}/256) - V_{254}}{1LSB} \quad [LSB]$$

- **Non-linearity error**

The deviation of the actual A-D conversion characteristics, from the ideal A-D conversion characteristics between V_0 and V_{254}.

$$\text{Non-linearity error} = \frac{V_n - (1LSB \times n + V_0)}{1LSB} \quad [LSB]$$

- **Differential non-linearity error**

The deviation of the input voltage required to change output data by "1," from the corresponding ideal A-D conversion characteristics between 0 and V_{REF}.

$$\text{Differential non-linearity error} = \frac{(V_{n+1} - V_n) - 1LSB}{1LSB} [LSB]$$

② Absolute accuracy

- **Absolute accuracy error**

The deviation of the actual A-D conversion characteristics, from the ideal A-D conversion characteristics between 0 and V_{REF}.

$$\text{Absolute accuracy error} = \frac{V_n - 1LSB_A \times (n + 1/2)}{1LSB_A} \quad [LSB]$$

Note: The analog input voltage "Vn" at which A-D conversion output data changes from "n" to "n + 1" (n ; 0 to 254) is as follows (refer to Figure 18).

$$1LSB \text{ with respect to relative accuracy} = \frac{V_{254} - V_0}{254} \quad [V]$$

$$1LSB_A \text{ with respect to absolute accuracy} = \frac{V_{REF}}{256} \quad [V]$$

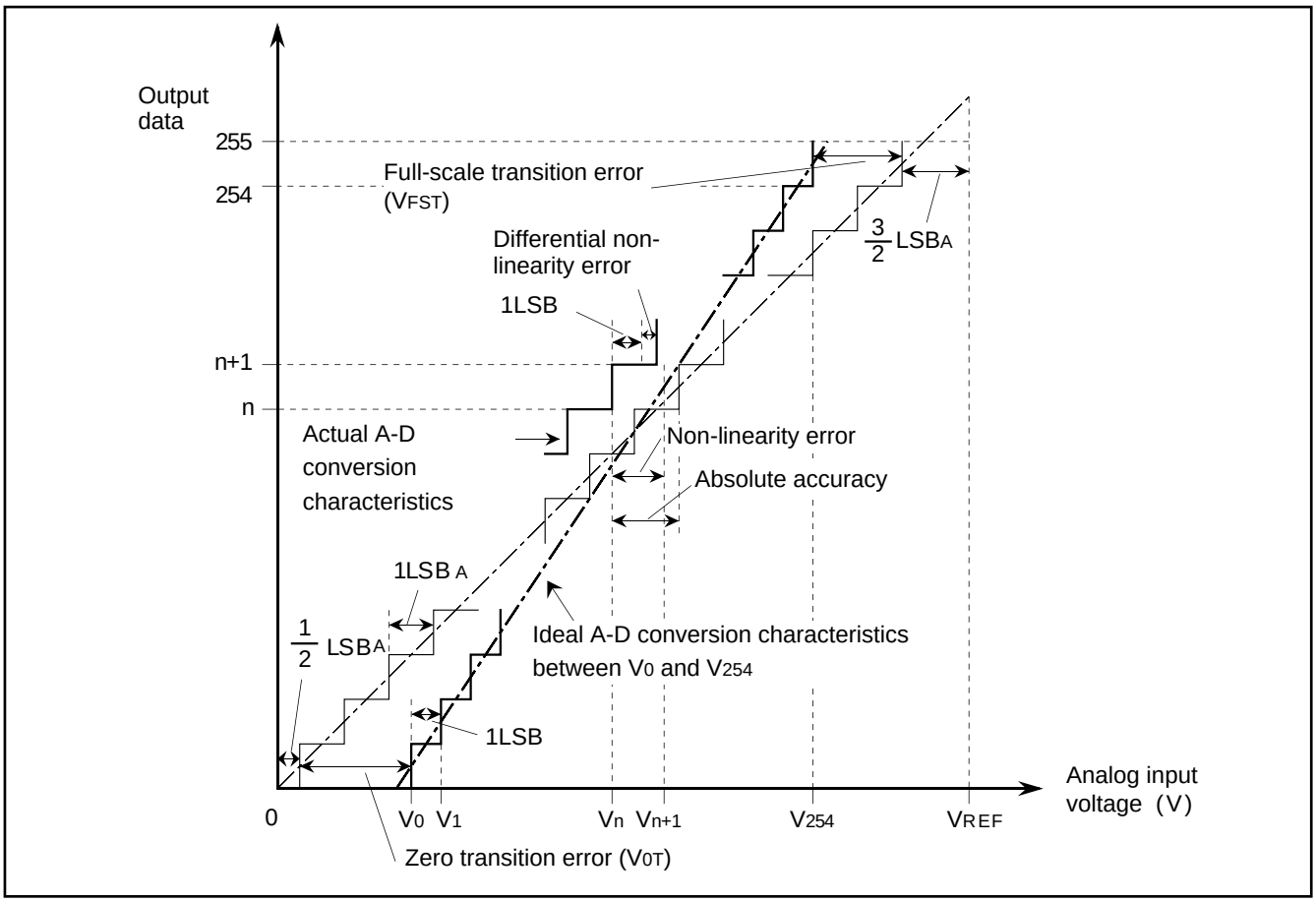


Fig. 18. Definition of A-D conversion precision

DATA SLICER

The M37270MF-XXXSP includes the data slicer function for the closed caption decoder (referred to as the CCD). This function takes out the caption data superimposed in the vertical blanking interval of a composite video signal. A composite video signal which makes the sync chip's polarity negative is input to the CV_{IN} pin.

When the data slicer function is not used, the data slicer circuit can be cut off by setting bit 0 of the data slicer control register 1 (address $00EA_{16}$) to "0." Also, the timing signal generating circuit can be cut off by setting bit 0 of data slicer control register 2 (address $00EB_{16}$) to "0." These settings can realize the low-power dissipation.

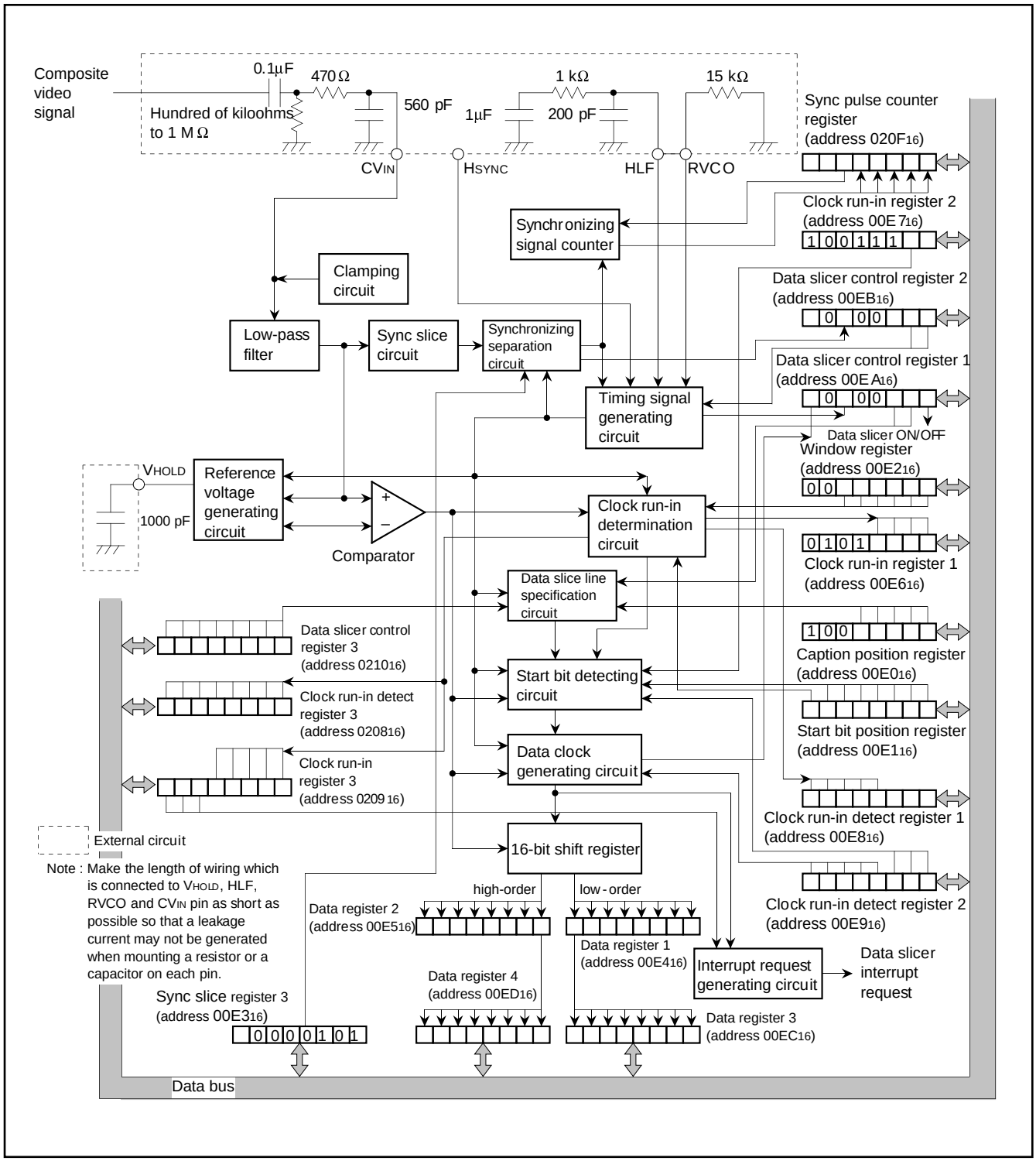


Fig. 19. Data slicer block diagram

Figure 19 shows the structure of the data slicer control registers.

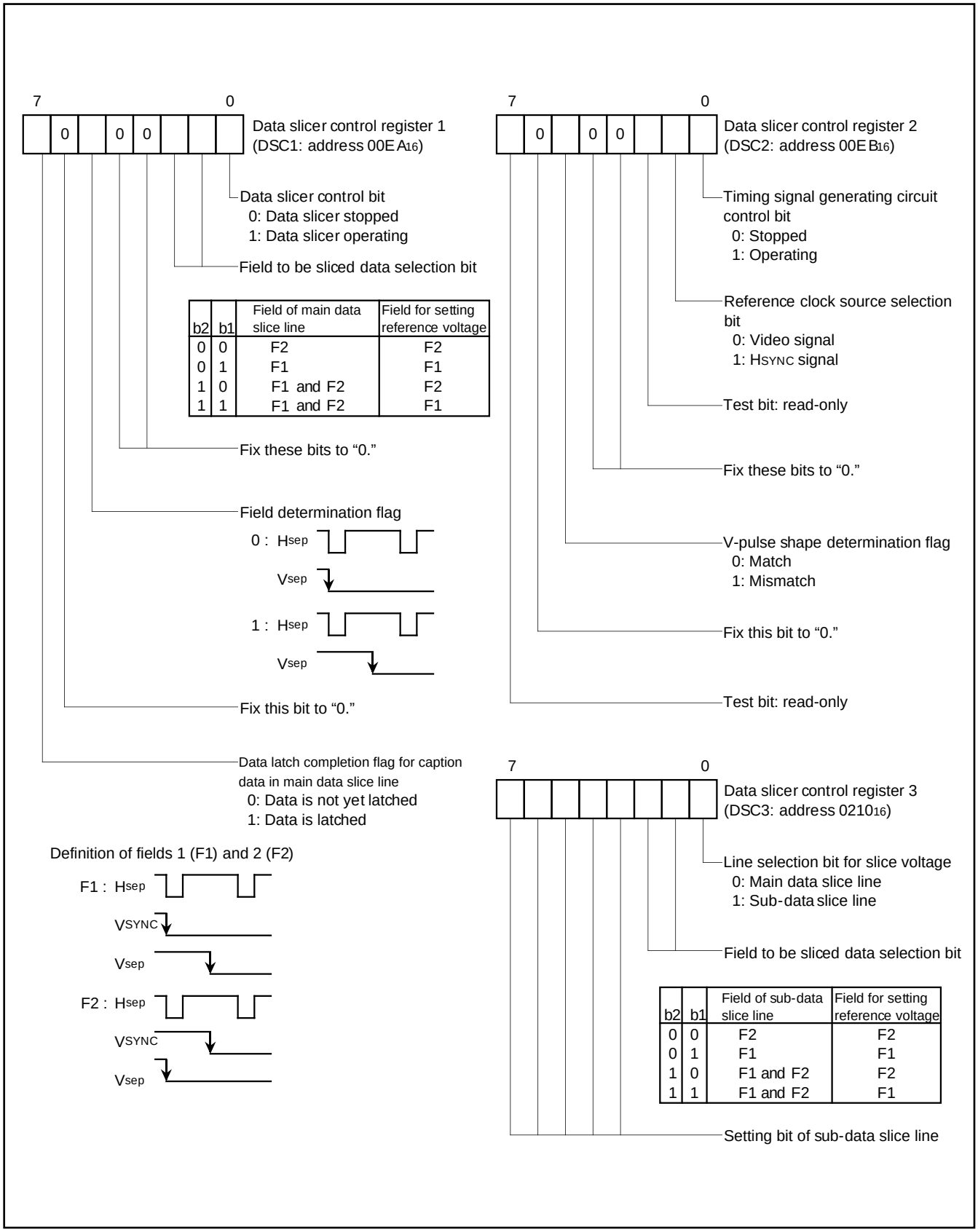


Fig. 20. Structure of data slicer control registers

(1) Clamping Circuit and Low-pass Filter

This filter attenuates the noise of the composite video signal input from the CV_{IN} pin. The CV_{IN} pin to which composite video signal is input requires a capacitor (0.1 μF) coupling outside. Pull down the CV_{IN} pin with a resistor of hundreds of kiloohms to 1 M . In addition, we recommend to install externally a simple low-pass filter using a resistor and a capacitor at the CV_{IN} pin (refer to Figure 19).

(2) Sync Slice Circuit

This circuit takes out a composite sync signal from the output signal of the low-pass filter. Figure 21 shows the structure of the sync slice register.

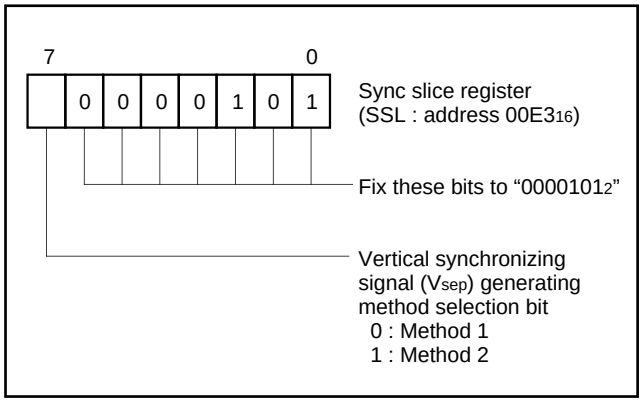


Fig. 21. Structure of sync slice register

(3) Synchronizing Signal Separation Circuit

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

① Horizontal synchronizing signal (H_{sep})

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

② Vertical synchronizing signal (V_{sep})

As a V_{sep} signal generating method, it is possible to select one of the following 2 methods by using bit 7 of the sync slice register (address $00E3_{16}$).

- •Method 1 The "L" level width of the composite sync signal is measured. If this width exceeds a certain time, a V_{sep} signal is generated in synchronization with the rising of the timing signal immediately after this "L" level.
- •Method 2 The "L" level width of the composite sync signal is measured. If this width exceeds a certain time, it is detected whether a falling of the composite sync signal exits or not in the "L" level period of the timing signal immediately after this "L" level. If a falling exists, a V_{sep} signal is generated in synchronization with the rising of the timing signal (refer to Figure 22).

Figure 22 shows a V_{sep} generating timing. The timing signal shown in the figure is generated from the reference clock which the timing generating circuit outputs.

Reading bit 5 of data slicer control register 2 permits determinating the shape of the V-pulse portion of the composite sync signal. As shown in Figure 23, when the A level matches the B level, this bit is "0." In the case of a mismatch, the bit is "1."

For the pins RVCO and the HLF, connect a resistor and a capacitor as shown in Figure 19. Make the length of wiring which is connected to these pins as short as possible so that a leakage current may not be generated.

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

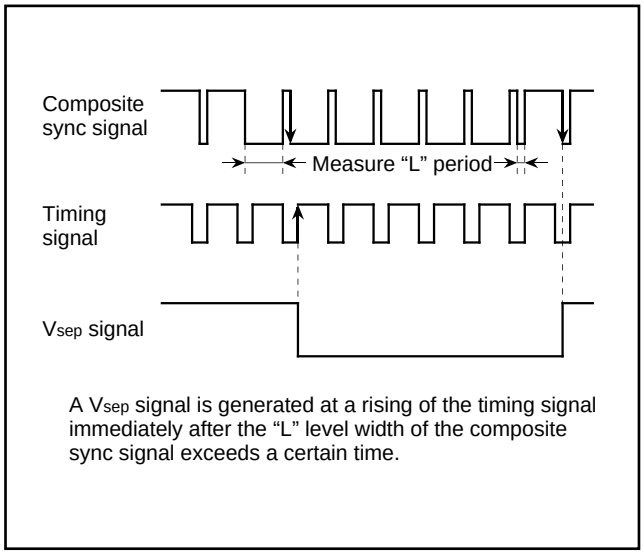


Fig. 22. Vsep generating timing (method 2)

(4) Timing Signal Generating Circuit

This circuit generates a reference clock which is 832 times as large as the horizontal synchronizing signal frequency. It also generates various timing signals on the basis of the reference clock, horizontal synchronizing signal and vertical synchronizing signal. The circuit operates by setting bit 0 of data slicer control register 2 (address $00EB_{16}$) 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 1 of data slicer control register 2 (address $00EB_{16}$).

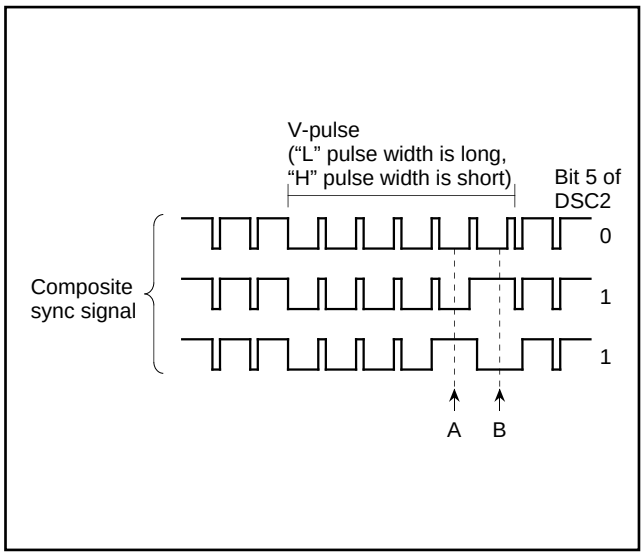


Fig. 23. Determination of V-pulse waveform

(5) Data Slice Line Specification Circuit

① Specification of data slice line

M37270MF-XXXSP has 2 data slice line specification circuits for slicing arbitrary 2 H_{sep} in 1 field. The following 2 data slice lines are specified .

<Main data slice line>

This line is specified by the caption position register (address $00E0_{16}$).

<Sub-data slice line>

This line is specified by the data slicer control register 3 (address $00EB_{16}$).

The counter is reset at the falling edge of V_{sep} and is incremented by 1 every H_{sep} pulse. When the counter value matched the value specified by bits 4 to 0 of the caption position register (in case of the sub-data slice line, by bits 3 to 7 of the data slicer control register 3), this H_{sep} is sliced.

The values of "00_{16}" to "$1F_{16}$" can be set in the caption position register. Bit 7 to bit 5 are used for testing. Set "100_2." Figure 24 shows the signals in the vertical blanking interval. Figure 25 shows the structure of the caption position register.

② Selection of field to be sliced data

In the case of the main data slice line, the field to be sliced data is selected by bits 2 and 1 of the data slicer control register 1 (address $00EA_{16}$). In the case of the sub-data slice line, the field is selected by bits 2 and 1 of the data slicer control register 3. When bit 2 of the data slicer control register 1 is set to "1," it is possible to slice data of both fields (refer to Figure 20).

③ Specification of line to set slice voltage

The reference voltage for slicing (slice voltage) is generated by integrating the amplitude of the clock run-in pulse in the particular line (refer to Table 3).

④ Field determination

The field determination flag can be read out by bit 5 of the data slicer control register 1. This flag charge at the falling edge of V_{sep}.

Table 3. Specifying of field whose sets reference voltage

Bit 0 of DSC3	Field	Line
0	Field specified by bit 1 of DSC1 0: F2 1: F1	Line specified by bits 4 to 0 of CP (Main data slice line)
1	Field specified by bit 1 of DSC3 0: F2 1: F1	Line specified by bits 7 to 3 of DSC3 (Sub-data slice line)

DSC1 : Data slice control register 1

DSC3 : Data slice control register 3

CP : Caption position register

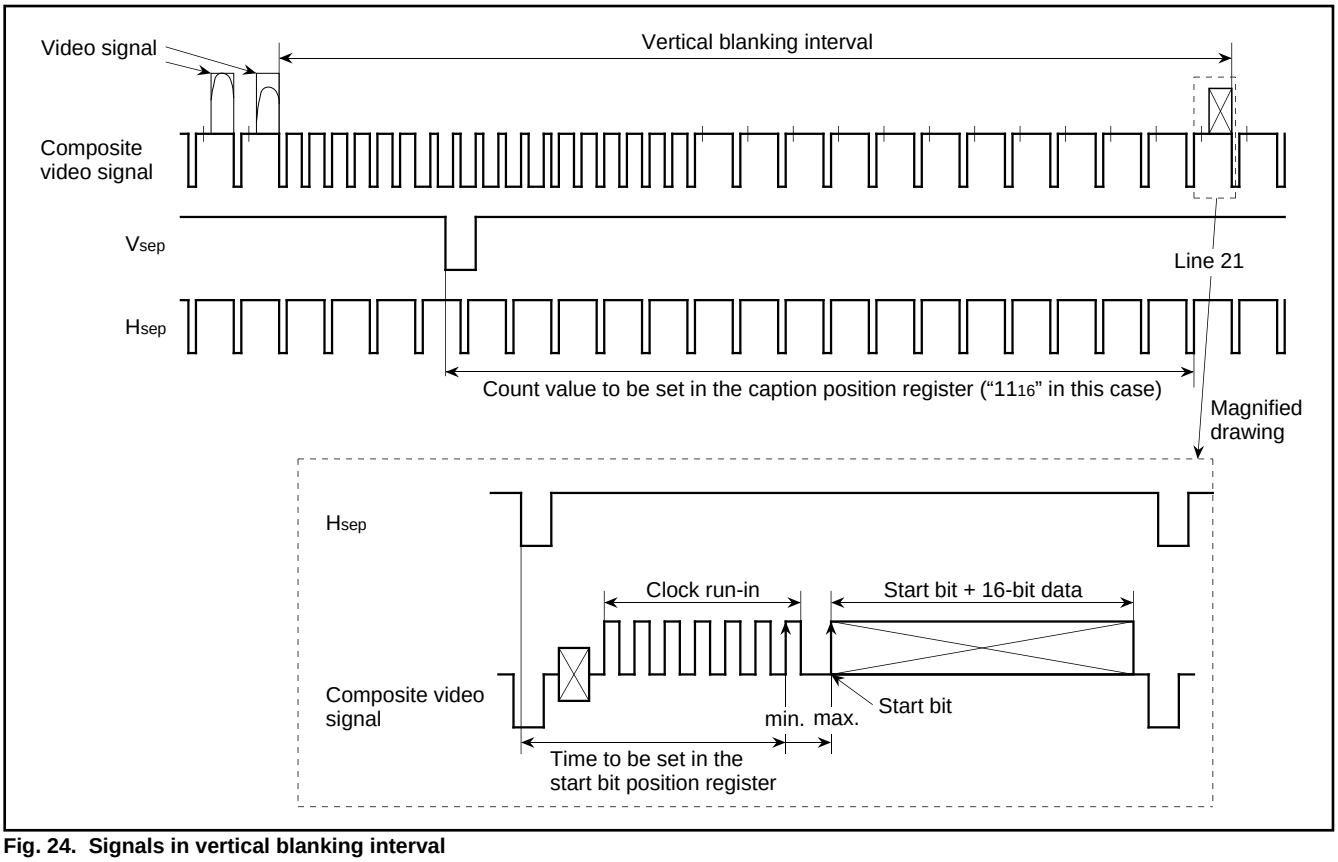


Fig. 24. Signals in vertical blanking interval

(6) 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 V_{HOLD} pin and the V_{SS} 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.

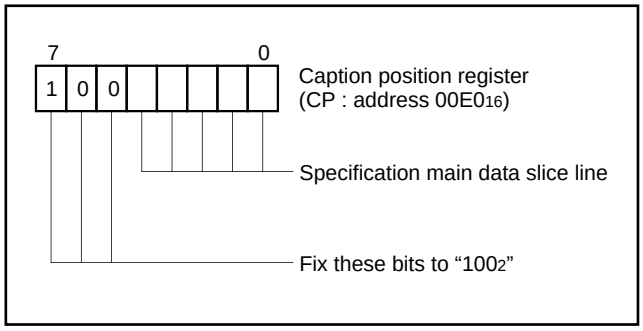


Fig. 25. Structure of caption position register

(7) Start Bit Detecting Circuit

This circuit detects a start bit at line decided in the data slice line specification circuit. For start bit detection, it is possible to select one of the following two types by using bit 1 of the clock run-in register 2 (address 00E7₁₆).

① After the lapse of the time corresponding to the set value of the start bit position register (address 00E1₁₆), the first rising of the composite video signal is detected as a start bit.

The time is set in bits 0 to 6 of the start bit position register (address 00E1₁₆) (refer to Figure 26). Set a value fit for the following conditions.

Figure 26 shows the structure of the start bit position register.

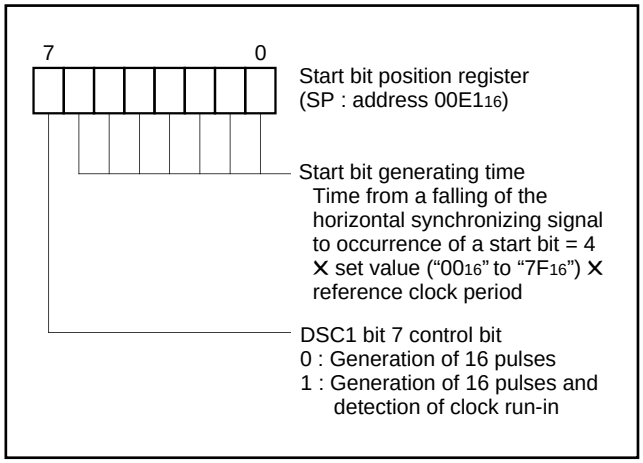


Fig. 26. Structure of start bit position register

[Time from the falling of the horizontal synchronizing signal to the last rising of the clock run-in] < [4 × set value of the start bit position register × reference clock period] < [Time from the falling of the horizontal synchronizing signal to occurrence of the start bit]

② After a falling of the clock run-in pulse set in bits 2 to 0 of clock run-in detect register 2 (address $00E9_{16}$) is detected, a start bit is detected by sampling a comparator output. A sampling clock for sampling is obtained by dividing the reference clock generated in the timing signal generating circuit by 13.
Figure 28 shows the structure of clock run-in detect register 2.
The contents of bits 2 to 0 of clock run-in detect register 2 and bit 1 of clock run-in register 2 are written at a falling of the horizontal synchronizing signal. For this reason, even if an instruction for setting is executed, the contents of the register cannot be rewritten until a falling of the horizontal synchronizing signal.

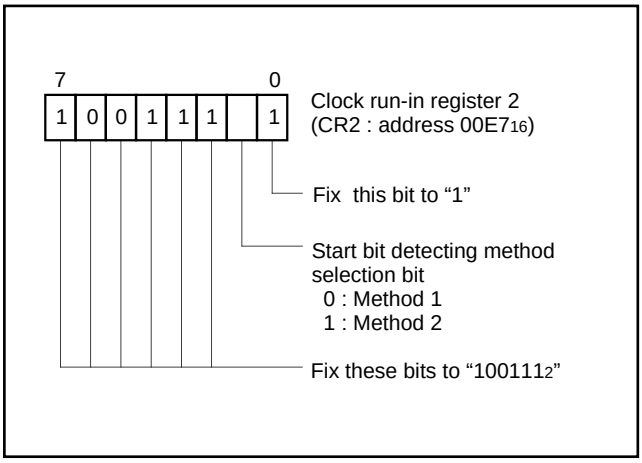


Fig. 27. Structure of clock run-in register 2

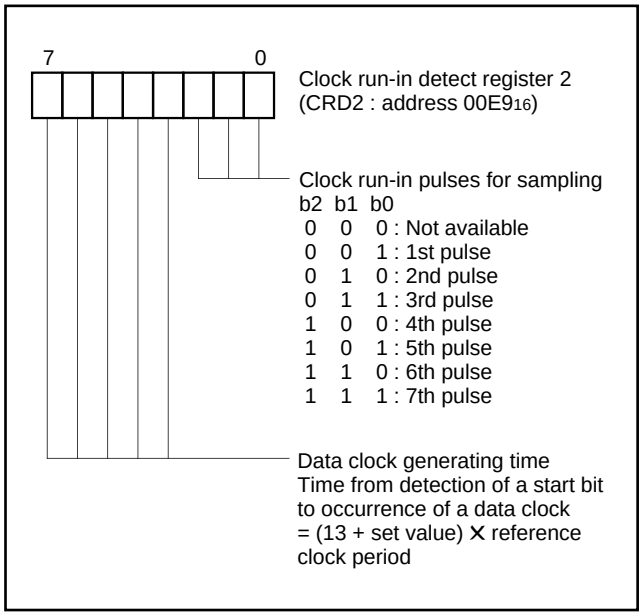


Fig. 28. Structure of clock run-in detect register 2

(8) Clock run-in determination circuit

This circuit sets a window in the clock run-in portion in the composite video signal, and then determinates clock run-in by counting the number of pulses in this window. Set the time from a falling of the horizontal synchronizing signal to a start of the window by bits 0 to 5 of the window register (address $00E2_{16}$; refer to Figure 29). The window ends according to the contents of the setting of the start bit position register (refer to Figure 26).

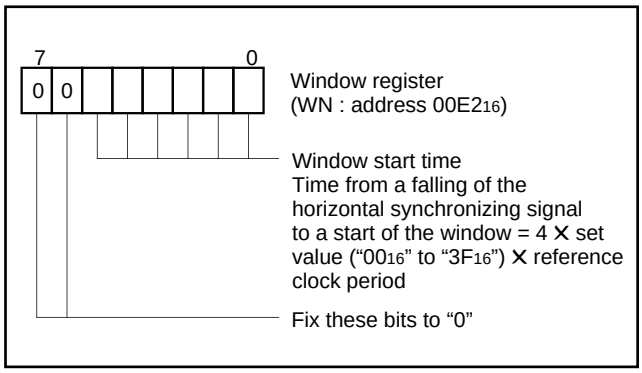


Fig. 29. Structure of window register

For the main data slice line, the count value of pulses in the window is stored in clock run-in register 1 (address $00E6_{16}$; refer to Figure 30). For the sub-data slice line, the count value of pulses in the window is stored in clock run-in register 3 (address 0209_{16}; refer to Figure 29). When this count value is 4 to 6, it is determined as a clock run-in. Accordingly, set the count value so that the window may start after the first pulse of the clock run-in (refer to Figure 32).

The contents to be set in the window register are written at a falling of the horizontal synchronizing signal. For this reason, even if an instruction for setting is executed, the contents of the register cannot be rewritten until a falling of the horizontal synchronizing signal.

For the main data slice line, reference clock is counted in the period from a falling of the clock pulse set in bits 0 to 2 of the clock run-in detect register 2 (address $00E9_{16}$) to the next falling. The count value is stored in bits 3 to 7 of the clock run-in detect register 1 (address $00E8_{16}$) (When the count value exceeds "$1F_{16}$," "$1F_{16}$" is held). For the sub-data slice line, the count value is stored in bits 3 to 7 of the clock run-in detect register 3 (address 0208_{16}). Read out these bits after the occurence of a data slicer interrupt (refer to (11) Interrupt Request Generating Circuit).

Figure 33 shows the structure of clock run-in detect registers 1 and 3.

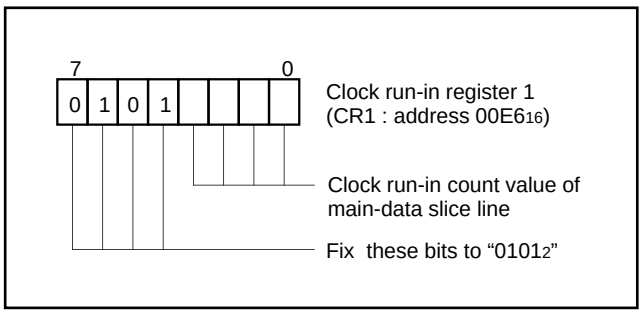


Fig. 30. Structure of clock run-in register 1

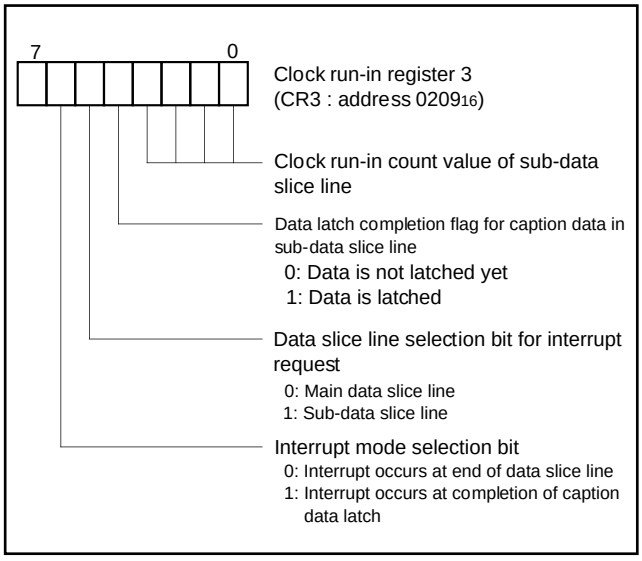


Fig. 31. Structure of clock run-in register 3

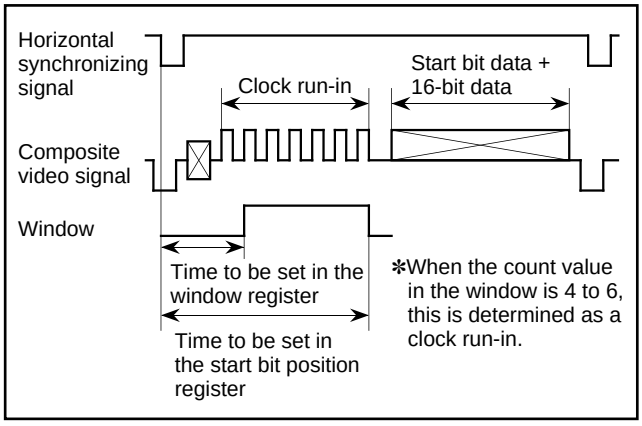


Fig. 32. Window setting

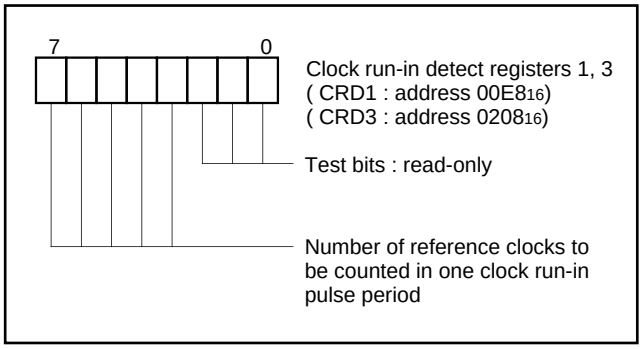


Fig. 33. Structure of clock run-in detect registers 1and 3

(9) Data clock generating circuit

This circuit generates a data clock phase-synchronized with the start bit detected in the start bit detecting circuit.

Set the time from detection of the start bit to occurrence of the data clock in bits 3 to 7 of the clock run-in detect register 2 (address $00E9_{16}$). The time to be set is represented by the following expression:

Time = (13 + set value) × reference clock period

For a data clock, 16 pulses are generated. When just 16 pulses have been generated, bit 7 of the data slicer control register is set to "1" (refer to Figure 20). When method 1 is already selected as a start bit detecting method, this bit becomes a logical product (AND) value with a clock run-in determination result by setting bit 7 of the start bit position register to "1."

When method 2 is already selected as a start bit detecting method and 16 pulses are generated of a data clock regardless of bit 7 of the start bit position register, this bit is set to "1." The contents of this bit are reset at a falling of the vertical synchronizing signal (V_{sep}).

Table 4. Setting conditions for caption data latch completion flag

Bit 7 of SP	Conditions for setting bit 7 of DSC1 to "1"	Conditions for setting bit 4 of DSC3 to "1"
0	Data clock of 16 pulses has occured in main data slaice line	Data clock of 16 pulses has occured in sub-data slaice line
1	Data clock of 16 pulses has occured in main data slaice line AND Clock run-in pulse are detected 4 to 6 times	Data clock of 16 pulses has occured in sub-data slaice line AND Clock run-in pulse are detected 4 to 6 times

(10) 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. For the main data slice line, the contents of the high-order 8 bits of the stored caption data and the contents of the low-order 8 bits of the same data can be obtained by reading out the data register 2 (address $00E5_{16}$) and data register 1 (address $00E4_{16}$), respectively. For the sub-data slice line, the contents of the high-order 8 bits and the contents of the low-order 8 bits can be obtained by reading out the data register 4 (address $00ED_{16}$) and the data register 3 (address $00EC_{16}$), respectively. These registers are reset to "0" at a falling of V_{sep}. Read out data registers 1 and 2 after the occurence of a data slicer interrupt (refer to (11) Interrupt Request Generating Circuit).

(11) Interrupt Request Generating Circuit

The occurence sources of interrupt request are selected by combination of the following bits; bits 5 and 6 of the clock run-in register 3 (address 0209_{16}), bit 1 of the clock run-in register 2 (address $00E7_{16}$) (refer to Table 6). Read out the contents of data registers 1 to 4 and the contents of bits 3 to 7 of the clock run-in detect registers 1 and 3 after the occurence of a data slicer interrupt request.

Table 5. Occurence sources of interrupt request

CR3		CR2	Occurence souces of interrupt request	
b5	b6	b1	Slice line	Sources
0	0	0	Main data slice line	At end of data slice line
		1		
	1	0		Data clock of 16 pulses has occured AND Clock run-in pulse are detected 4 to 6 times
		1		Data clock of 16 pulses has occured
1	0	0	Sub-data slice line	At end of data slice line
		1		
	1	0		Data clock of 16 pulses has occured AND Clock run-in pulse are detected 4 to 6 times
		1		Data clock of 16 pulses has occured

(12) Synchronizing Signal Counter

The synchronizing signal counter counts the composite sync signal taken out from a video signal in the data slicer circuit or the vertical synchronizing signal V_{sep} as a count source.

The count value in a certain time (T time) generated by $f(X_{IN})/2^{13}$ or $f(X_{IN})/2^{13}$ is stored into the 5-bit latch. Accordingly, the latch value changes in the cycle of T time. When the count value exceeds "$1F_{16}$," "$1F_{16}$" is stored into the latch.

The latch value can be obtained by reading out the sync pulse counter register (address $00EA_{16}$). A count source is selected by bit 5 of the sync pulse counter register.

The synchronizing signal counter is used when bit 0 of the PWM mode register 1 (address $02EA_{16}$).

Figure 34 shows the structure of the sync pulse counter and Figure 35 shows the synchronizing signal counter block diagram.

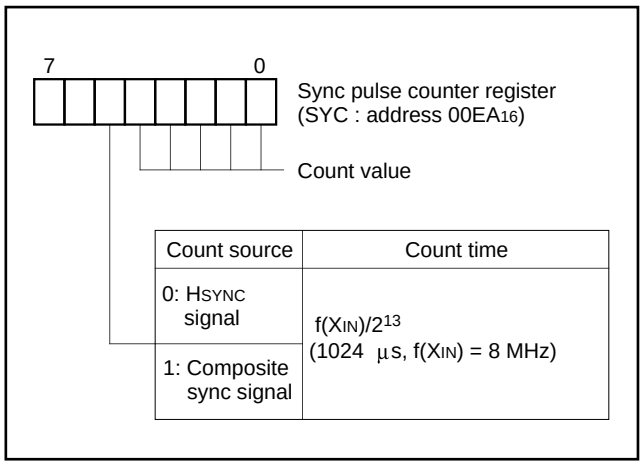


Fig. 34. Sync pulse counter register

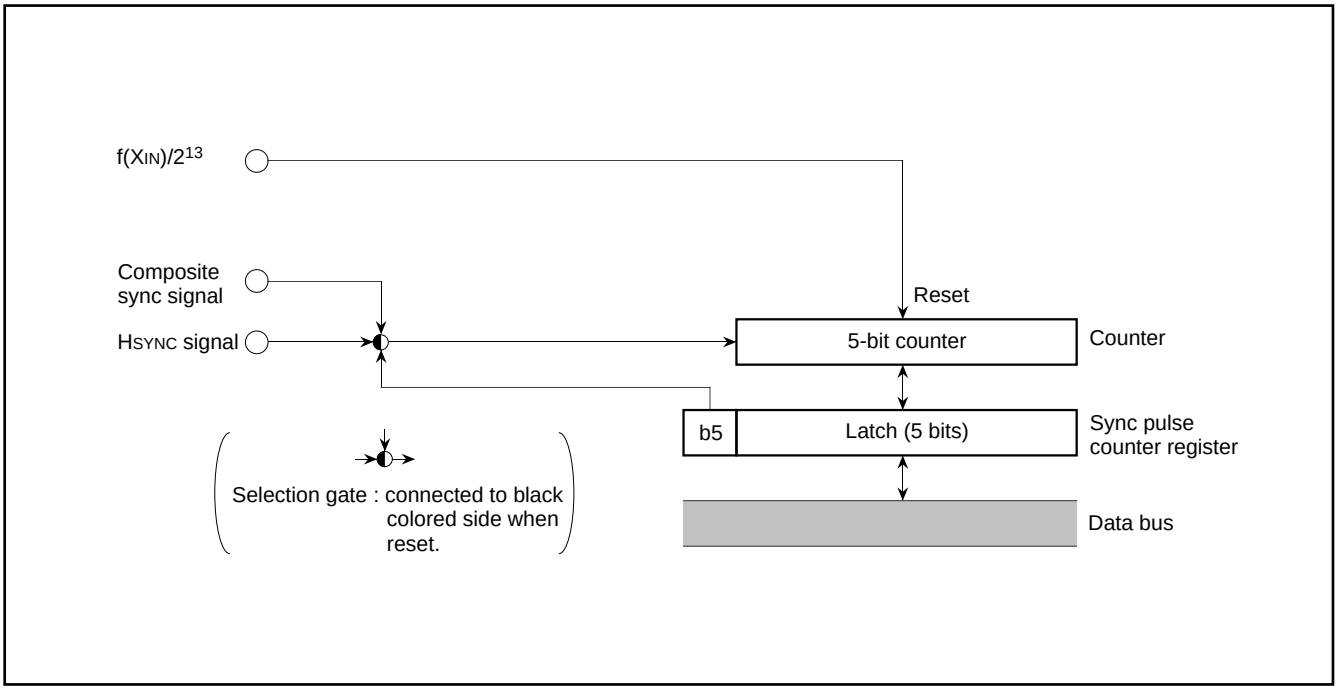


Fig. 35. Synchronizing signal counter block diagram

MULTI-MASTER I²C-BUS INTERFACE

The multi-master I²C-BUS interface is a circuit for serial communications conformed with the Philips I²C-BUS data transfer format. This interface, having an arbitration lost detection function and a synchronous function, is useful for serial communications of the multi-master.

Figure 36 shows a block diagram of the multi-master I²C-BUS interface and Table 6 shows multi-master I²C-BUS interface functions.

This multi-master I²C-BUS interface consists of the I²C address register, the I²C data shift register, the I²C clock control register, the I²C control register, the I²C status register and other control circuits.

Table 6. Multi-master I²C-BUS interface functions

Item	Function
Format	In conformity with Philips I²C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	In conformity with Philips I²C-BUS standard: Master transmission Master reception Slave transmission Slave reception
SCL clock frequency	16.1 kHz to 400 kHz (at ϕ = 4 MHz)

ϕ : System clock = f(X_{IN})/2

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 I²C control register at address 00F9₁₆) for connections between the I²C-BUS interface and ports (SCL1, SCL2, SDA1, SDA2).

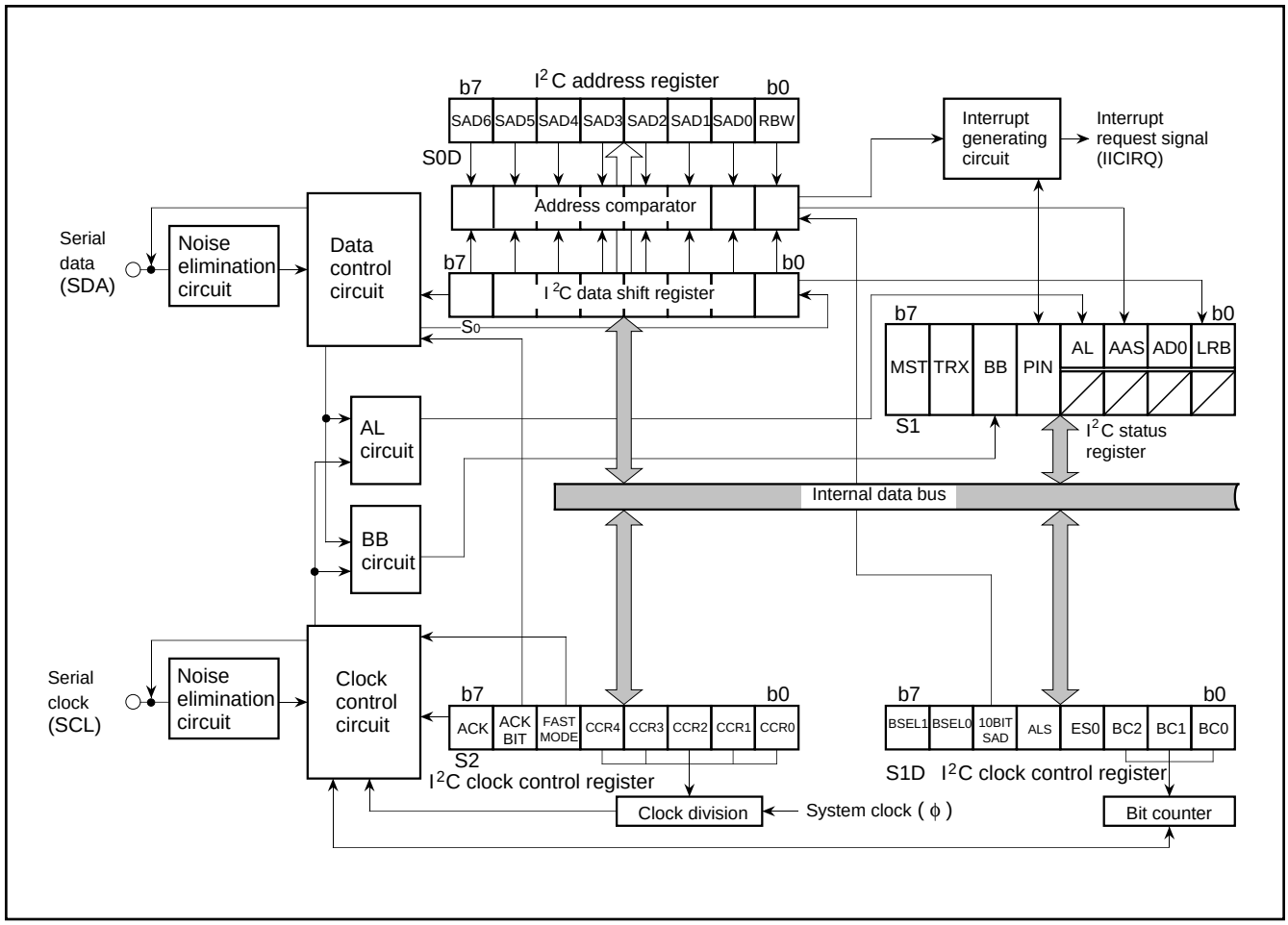


Fig. 36. Block diagram of multi-master I²C-BUS interface

(1) I²C Data Shift Register

The I²C data shift register (S0 : address 00F6₁₆) 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 I²C data shift register is in a write enable status only when the ES0 bit of the I²C control register (address 00F9₁₆) is "1." The bit counter is reset by a write instruction to the I²C data shift register. When both the ES0 bit and the MST bit of the I²C status register (address 00F8₁₆) are "1," the SCL is output by a write instruction to the I²C data shift register. Reading data from the I²C data shift register is always enabled regardless of the ES0 bit value.

Note: To write data into the I²C data shift register after setting the MST bit to "0" (slave mode), keep an interval of 8 machine cycles or more.

(2) I²C Address Register

The I²C address register (address 00F7₁₆) 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 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 I²C 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.

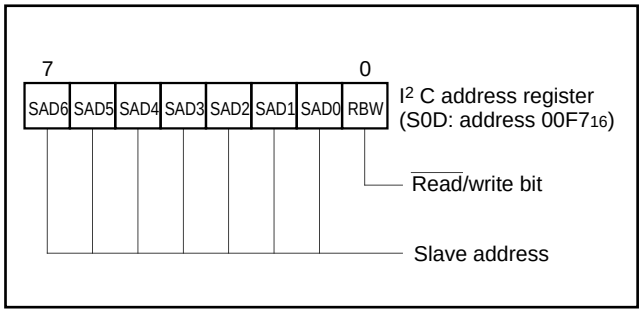


Fig. 37. Structure of I²C address register

(3) I²C Clock Control Register

The I²C clock control register (address 00FA₁₆) 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. Refer to Table 7.

■ 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 make SDA "L" 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 "H" 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 "L" (ACK is returned). If there is a mismatch between the slave address and the address data, the SDA is automatically made "H"(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 "H") and receives the ACK bit generated by the data receiving device.

Note: Do not write data into the I²C clock control register during transmitting. If data is written during transmitting, the I²C clock generator is reset, so that data cannot be transmitted normally.

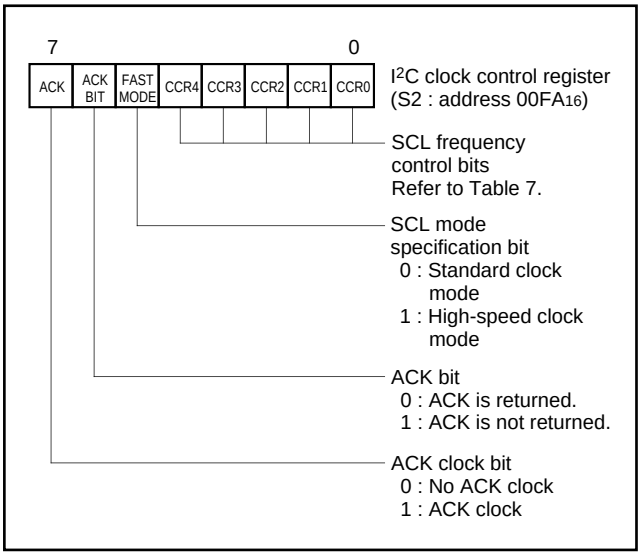


Fig. 38. Structure of I²C clock control register

Table 7. Set values of I²C clock control register and SCL frequency

Setting value of CCR4–CCR0					SCL frequency (at φ = 4MHz, unit : kHz)	
CCR4	CCR3	CCR2	CCR1	CCR0	Standard clock mode	High-speed clock mode
0	0	0	0	0	Setting disabled	Setting disabled
0	0	0	0	1	Setting disabled	Setting disabled
0	0	0	1	0	Setting disabled	Setting disabled
0	0	0	1	1	Setting disabled	333
0	0	1	0	0	Setting disabled	250
0	0	1	0	1	100	400(Note)
0	0	1	1	0	83.3	166
⋮	⋮	⋮	⋮	⋮	500/CCR value	1000/CCR value
1	1	1	0	1	17.2	34.5
1	1	1	1	0	16.6	33.3
1	1	1	1	1	16.1	32.3

Note: At 400 kHz in the high-speed clock mode, the duty is 40%. In the other cases, the duty is 50%.

(4) I²C Control Register

The I²C control register (address 00F9₁₆) controls 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.

■ Bit 3: I²C interface use enable bit (ES0)

This bit enables to use the multimaster I²C BUS interface. 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 ES0 = "0," the following is performed.

- PIN = "1," BB = "0" and AL = "0" are set (they are bits of the I²C status register at address 00F8₁₆).
- Writing data to the I²C data shift register (address 00F6₁₆) is disabled.

■ Bit 4: Data format selection bit (ALS)

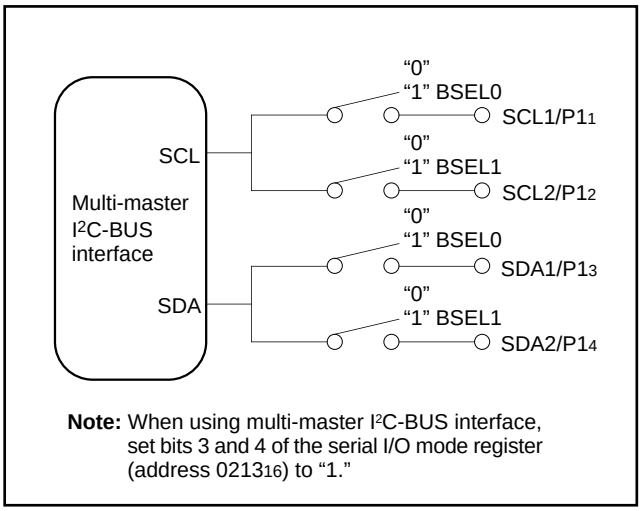
This bit decides whether or not to recognize slave addresses. When this bit is set to "0," the addressing format is selected, so that address data is recognized. When a match is found between a slave address and address data as a result of comparison or when a general call (refer to "(5) I²C Status Register," 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 I²C address register (address 00F7₁₆) are compared with address data. When this bit is set to "1," the 10-bit addressing format is selected, all the bits of the I²C address register are compared with address data.

■ Bits 6 and 7: Connection control bits between I²C-BUS interface and ports (BSEL0, BSEL1)

These bits controls the connection between SCL and ports or SDA and ports (refer to Figure 39).



Note: When using multi-master I²C-BUS interface, set bits 3 and 4 of the serial I/O mode register (address 0213_{16}) to "1."

Fig. 39. Connection port control by BSEL0 and BSEL1

(5) I²C Status Register

The I²C status register (address $00F8_{16}$) controls the I²C-BUS interface status. The low-order 4 bits are read-only bits and the high-order 4 bits 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 I²C data shift register (address $00F6_{16}$).

■ Bit 1: General call detecting flag (AD0)

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

*General call: The master transmits the general call address "00_{16}" to all slaves.

■ Bit 2: Slave address comparison flag (AAS)

This flag indicates a comparison result of address data.

① 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 agrees with the slave address stored in the high-order 7 bits of the I²C address register (address $00F7_{16}$).
- A general call is received.

② 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 I²C address register (8 bits consisted of slave address and RBW), the first bytes agree.

③ The state of this bit is changed from "1" to "0" by executing a write instruction to the I²C data shift register (address $00F6_{16}$).

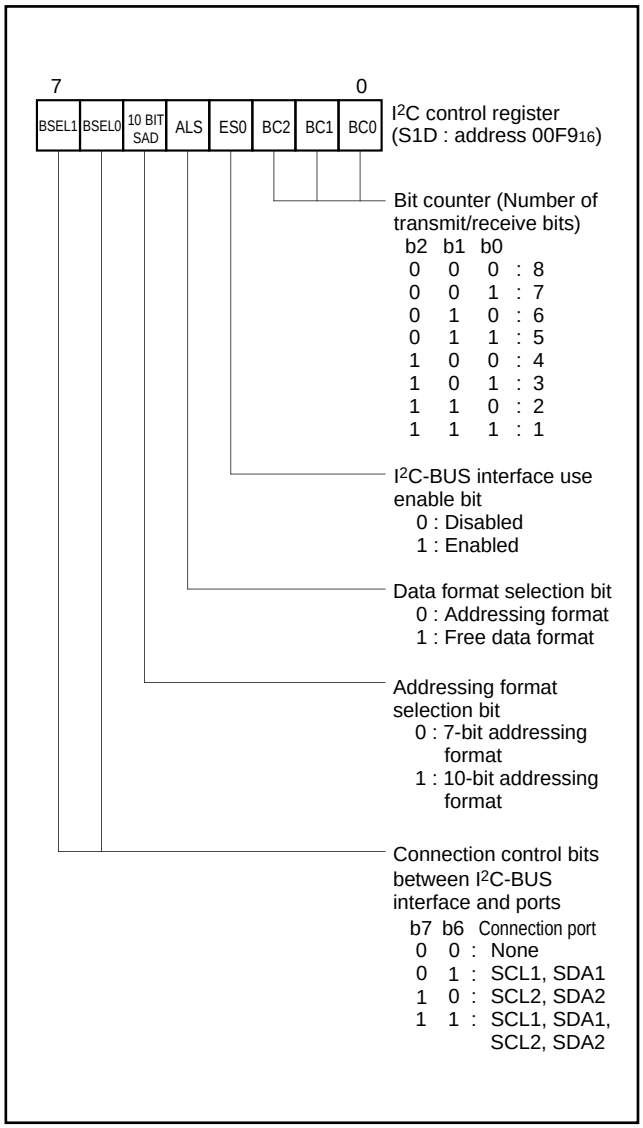


Fig. 40. Structure of I²C control register

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

In the master transmission mode, when the SDA is made "L" by any other device, 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." In the case 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^2C-BUS interface 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 occurs to the CPU. The PIN bit is set to "0" in synchronization with a falling of the last clock (including the ACK clock) of an internal clock and an interrupt request signal occurs in synchronization with a falling of the PIN bit. When the PIN bit is "0," the SCL is kept in the "0" state and clock generation is disabled. Figure 42 shows an interrupt request signal generating timing chart.

The PIN bit is set to "1" in one of the following conditions.

- Executing a write instruction to the I^2C data shift register (address $00F6_{16}$).
- When the ES0 bit is "0"
- At reset

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 (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 ES0 bit of the I^2C control register (address $00F9_{16}$) 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 a 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 onto the SDA in synchronization with the clock generated on the SCL.

When the ALS bit of the I^2C control register (address $00F9_{16}$) is "0" in the slave reception mode is selected, the TRX bit is set to "1" (transmit) if the least significant bit (R/$\overline{W}$ bit) of the address data transmitted by the master is "1." When the ALS bit is "0" and the R/$\overline{W}$ bit is "0," the TRX bit is cleared to "0" (receive).

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

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

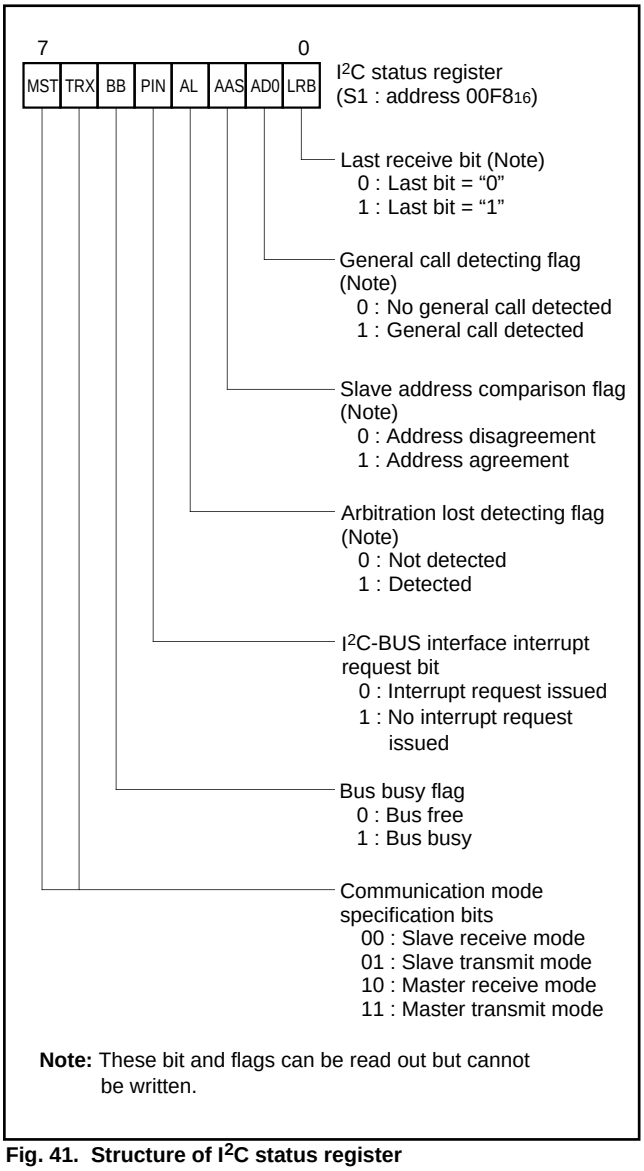
■ Bit 7: Communication mode specification bit (master/slave specification bit: MST)

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

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

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

Note: The START condition duplication prevention function disables the occurence of a START condition, reset of bit counter and SCL output when the following condition is satisfied:
• a START condition is set by another master device.



Note: These bit and flags can be read out but cannot be written.

Fig. 41. Structure of I²C status register

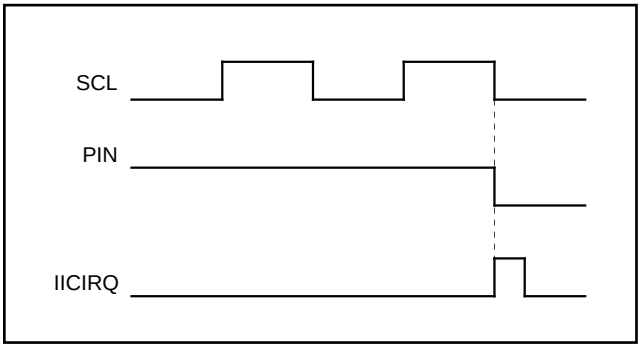


Fig. 42. Interrupt request signal generating timing

(6) START Condition Generating Method

When the ES0 bit of the I²C control register (address 00F9₁₆) is "1," execute a write instruction to the I²C status register (address 00F8₁₆) for setting the MST, TRX and BB bits to "1." Then a START condition occurs. After that, the bit counter becomes "000₂" and an SCL for 1 byte is output. The START condition generating timing and BB bit set timing are different in the standard clock mode and the high-speed clock mode. Refer to Figure 43, the START condition generating timing diagram, and Table 8, the START condition/STOP condition generating timing table.

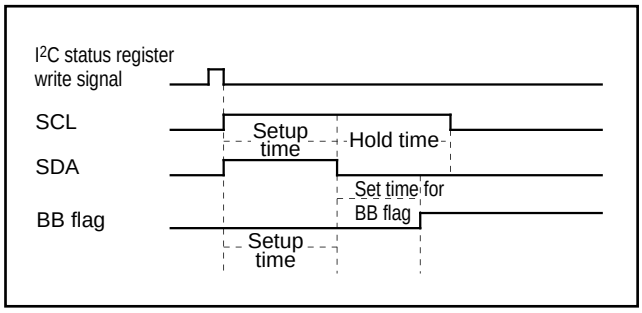


Fig. 43. START condition generating timing diagram

(7) STOP Condition Generating Method

When the ES0 bit of the I²C control register (address 00F9₁₆) is "1," execute a write instruction to the I²C status register (address 00F8₁₆) for setting the MST bit and the TRX bit to "1" and the BB bit to "0". Then a STOP condition occurs. The STOP condition generating timing and the BB flag reset timing are different in the standard clock mode and the high-speed clock mode. Refer to Figure 44, the STOP condition generating timing diagram, and Table 8, the START condition/STOP condition generating timing table.

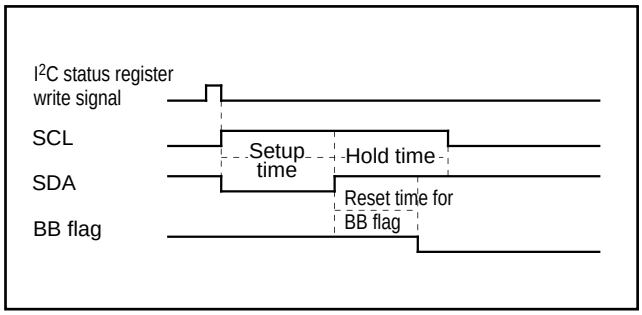


Fig. 44. STOP condition generating timing diagram

Table 8. START condition/STOP condition generating timing table

Item	Standard clock mode	High-speed clock mode
Setup time	5.0 μs (20 cycles)	2.5 μs (10 cycles)
Hold time	5.0 μs (20 cycles)	2.5 μs (10 cycles)
Set/reset time for BB flag	3.0 μs (12 cycles)	1.5 μs (6 cycles)

Note: Absolute time at ϕ = 4 MHz. The value in parentheses denotes the number of ϕ cycles.

(8) START/STOP Condition Detecting Conditions

The START/STOP condition detecting conditions are shown in Figure 45 and Table 9. Only when the 3 conditions of Table 9 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" occurs to the CPU.

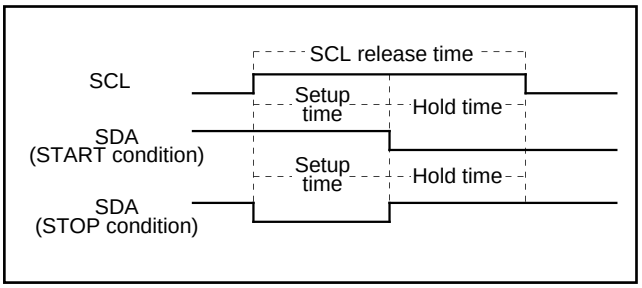


Fig. 45. START condition/STOP condition detecting timing diagram

Table 9. START condition/STOP condition detecting conditions

Standard clock mode	High-speed clock mode
6.5 μs (26 cycles) < SCL release time	1.0 μs (4 cycles) < SCL release time
3.25 μs (13 cycles) < Setup time	0.5 μs (2 cycles) < Setup time
3.25 μs (13 cycles) < Hold time	0.5 μs (2 cycles) < Hold time

Note: Absolute time at ϕ = 4 MHz. The value in parentheses denotes the number of ϕ cycles.

(9) 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.

① 7-bit addressing format

To meet the 7-bit addressing format, set the 10BIT SAD bit of the I^2C control register (address $00F9_{16}$) 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^2C address register (address $00F7_{16}$). At the time of this comparison, address comparison of the RBW bit of the I^2C address register (address $00F7_{16}$) is not made. For the data transmission format when the 7-bit addressing format is selected, refer to Figure 46, (1) and (2).

② 10-bit addressing format

To meet the 10-bit addressing format, set the 10BIT SAD bit of the I^2C control register (address $00F9_{16}$) 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^2C address register (address $00F7_{16}$). At the time of this comparison, an address comparison between the RBW bit of the I^2C address register (address $00F7_{16}$) and the R/$\overline{W}$ bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the R/$\overline{W}$ bit which is the last bit of the address data not only specifies the direction of communication for control data but also is processed as an address data bit.

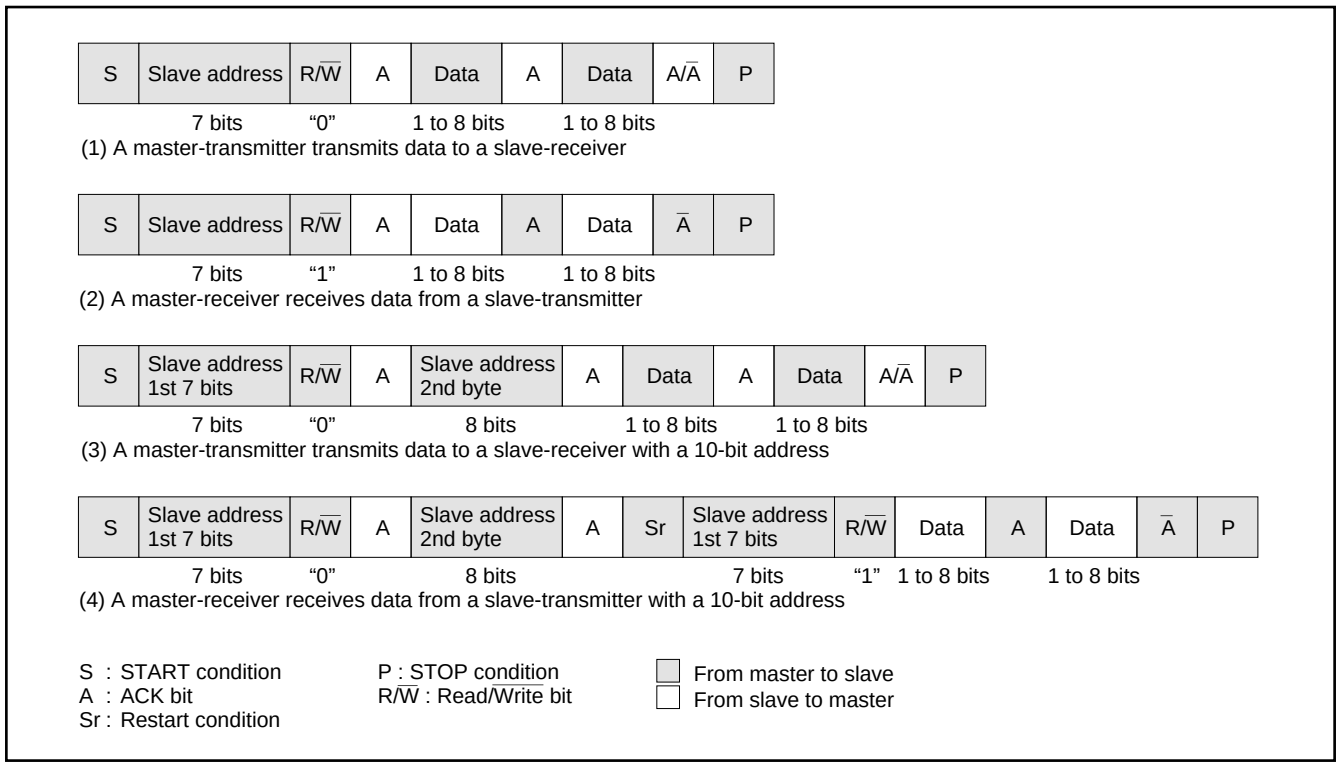


Fig. 46. Address data communication format

When the first-byte address data matches the slave address, the AAS bit of the I^2C status register (address $00F8_{16}$) is set to "1." After the second-byte address data is stored into the I^2C data shift register (address $00F6_{16}$), make an address comparison between the second-byte data and the slave address by software. When the address data of the 2nd byte matches the slave address, set the RBW bit of the I^2C address register (address $00F7_{16}$) to "1" by software. This processing can match the 7-bit slave address and R/$\overline{W}$ data, which are received after a RESTART condition is detected, with the value of the I^2C address register (address $00F7_{16}$). For the data transmission format when the 10-bit addressing format is selected, refer to Figure 46, (3) and (4).

(10) 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^2C address register (address $00F7_{16}$) and "0" in the RBW bit.
② Set the ACK return mode and SCL = 100 kHz by setting "85_{16}" in the I^2C clock control register (address $00FA_{16}$).
③ Set "10_{16}" in the I^2C status register (address $00F8_{16}$) and hold the SCL at the "H" level.
④ Set a communication enable status by setting "48_{16}" in the I^2C control register (address $00F9_{16}$).
⑤ Set the address data of the destination of transmission in the high-order 7 bits of the I^2C data shift register (address $00F6_{16}$) and set "0" in the least significant bit.
⑥ Set "$F0_{16}$" in the I^2C status register (address $00F8_{16}$) 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^2C data shift register (address $00F6_{16}$). At this time, an SCL and an ACK clock automatically occurs.
⑧ When transmitting control data of more than 1 byte, repeat step ⑦.
⑨ Set "$D0_{16}$" in the I^2C status register (address $00F8_{16}$). After this, if ACK is not returned or transmission ends, a STOP condition occurs.

(11) 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 and using the addressing format is shown below.

① Set a slave address in the high-order 7 bits of the I^2C address register (address $00F7_{16}$) and "0" in the RBW bit.
② Set the no ACK clock mode and SCL = 400 kHz by setting "25_{16}" in the I^2C clock control register (address $00FA_{16}$).
③ Set "10_{16}" in the I^2C status register (address $00F8_{16}$) and hold the SCL at the "H" level.
④ Set a communication enable status by setting "48_{16}" in the I^2C control register (address $00F9_{16}$).
⑤ When a START condition is received, an address comparison is made.
⑥ •When all transmitted addresses are "0" (general call)
AD0 of the I^2C status register (address $00F8_{16}$) is set to "1" and an interrupt request signal occurs.
•When the transmitted addresses match the address set in ①
AAS of the I^2C status register (address $00F8_{16}$) is set to "1" and an interrupt request signal occurs.
•In the cases other than the above
AD0 and AAS of the I^2C status register (address $00F8_{16}$) are set to "0" and no interrupt request signal occurs.
⑦ Set dummy data in the I^2C data shift register (address $00F6_{16}$).
⑧ When receiving control data of more than 1 byte, repeat step ⑦.
⑨ When a STOP condition is detected, the communication ends.

OSD FUNCTIONS

Table 10 outlines the OSD functions of the M37270MF-XXXSP.

The M37270MF-XXXSP incorporates an OSD control circuit of 40 characters × 16 lines. OSD is controlled by the OSD control register. There are 3 display modes and they are selected by a block unit. The display modes are selected by the block control register i (i = 1 to 6).

The features of each mode are described below.

Table 10. Features of each display mode

Parameter		Display mode		
		CC mode (Closed caption mode)	OSD mode (On-screen display mode)	EXOSD mode (Extra on-screen display mode)
Number of display characters		40 characters × 16 lines	40 characters × 16 lines	40 characters × 16 lines
Dot structure		16 × 26 dots (Character : 20 × 16 dots)	16 × 20 dots	16 × 26 dots
Kinds of characters		320 kinds (In EXOSD mode, they can be combined with 32 kinds of extra fonts)		
Kinds of character sizes		2 kinds	14 kinds	6 kinds
	Pre-divide ratio (Note)	× 1, × 2	× 1, × 2, × 3	× 1, × 2, × 3
	Dot size	1Tc × 1/2H	1Tc × 1/2H, 1Tc × 1H, 1.5Tc × 1/2H, 1.5Tc × 1H, 2Tc × 2H, 3Tc × 3H	1Tc × 1/2H, 1Tc × 1H
Attribute		Smooth italic, under line, flash	Border	Border, extra font (32 kinds)
Character font coloring		1 screen : 7 kinds, Max. 7 kinds (a character unit)	1 screen : 7 kinds, Max. 15 kinds (a character unit)	1 screen : 7 kinds, Max. 7 kinds (a character unit)
Raster coloring		Possible (a screen unit, 1 screen : 7 kinds, max. 7 kinds)	Possible (a screen unit, 1 screen : 7 kinds, max. 7 kinds)	Possible (a screen unit, 1 screen : 7 kinds, max. 7 kinds)
Character background coloring		Possible (a character unit, 1 screen : 7 kinds, max. 7 kinds)	Possible (a character unit, 1 screen : 7 kinds, max. 7 kinds)	Possible (a character unit, 1 screen : 7 kinds, max. 7 kinds)
Border coloring		———	Possible (a screen unit, 1 screen : 7 kinds, max. 7 kinds)	Possible (a screen unit, 1 screen : 7 kinds, max. 7 kinds)
Extra font coloring		———	———	Possible (a screen unit, 1 screen : 7 kinds, max. 7 kinds)
OSD output		R, G, B, OUT1, OUT2	R, G, B, I1, OUT1, OUT2	R, G, B, I1, I2, OUT1, OUT2
Function		Auto solid space function Window function Dual layer OSD function (layer 1)	Dual layer OSD function (layer 2)	———
Display expansion (multiline display)		Possible	Possible	Possible

Notes 1: The divide ratio of the frequency divider (the pre-divide circuit) is referred as “pre-divide ratio” hereafter.
2: The character size is specified with dot size and pre-divide ratio (refer to (3) Dote size).

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 47 shows the configuration of OSD character. Figure 48 shows the block diagram of the OSD control circuit. Figure 49 shows the structure of the OSD control register. Figure 50 shows the structure of the block control register.

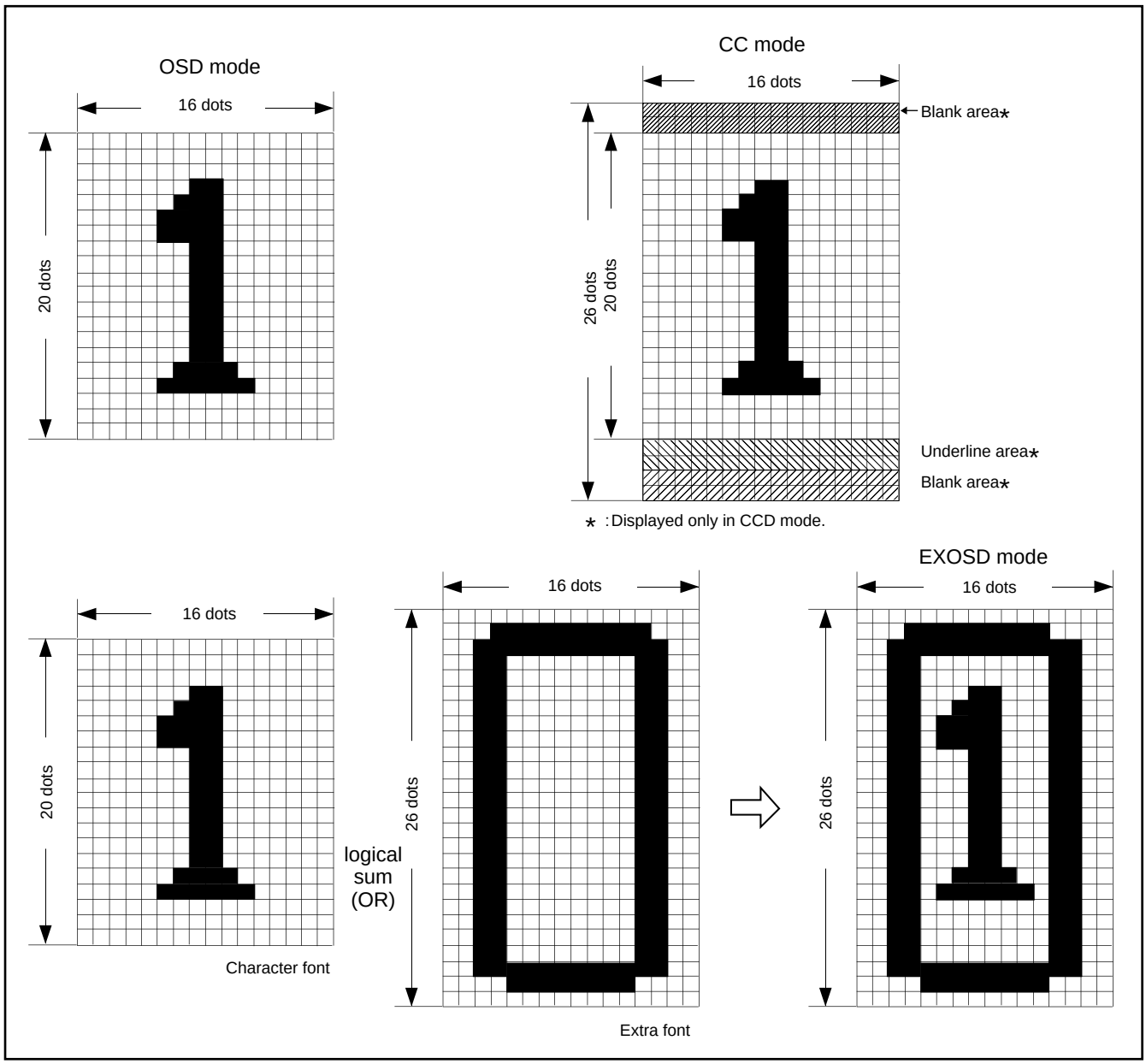


Fig. 47. Configuration of OSD character

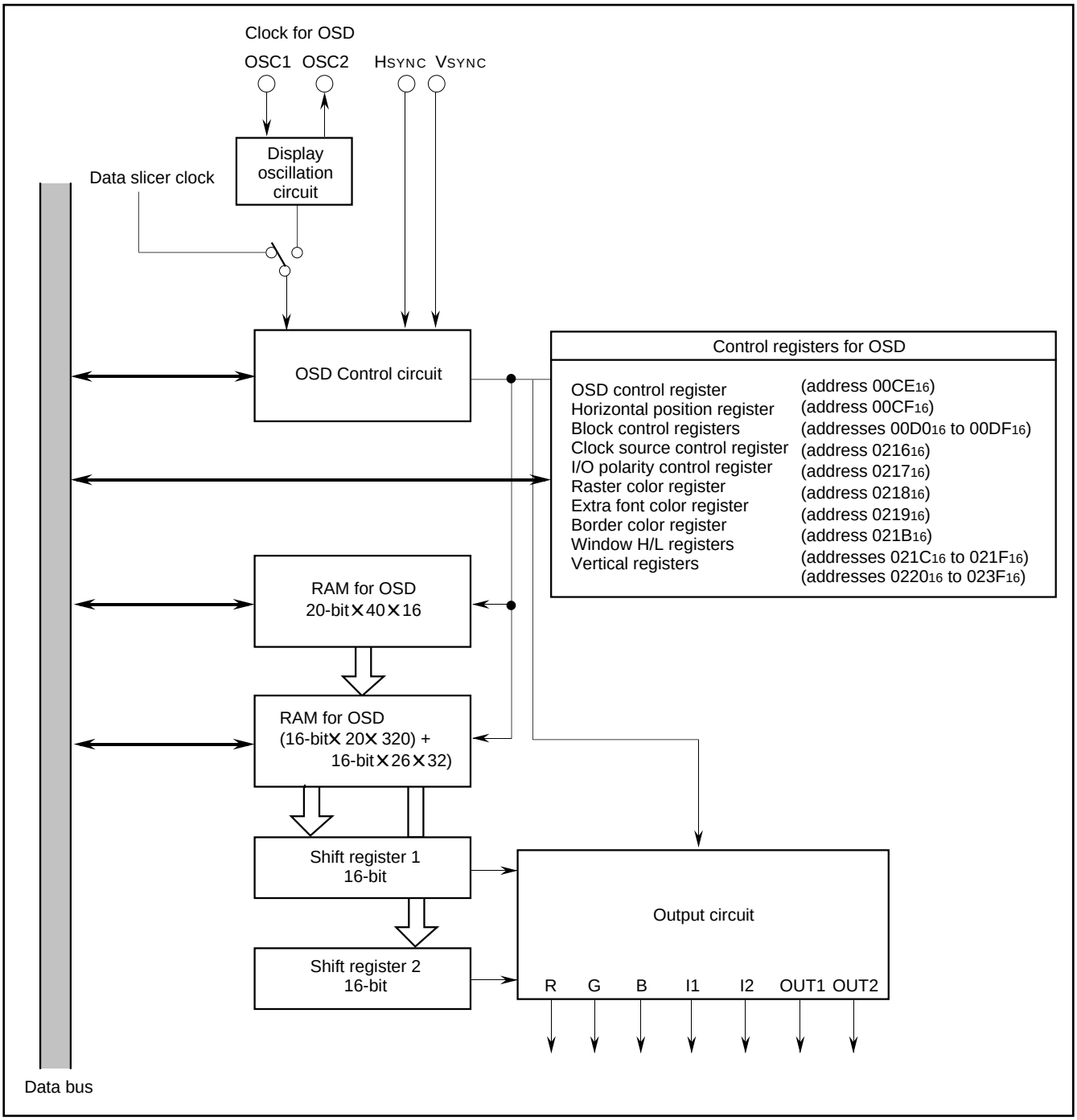


Fig. 48. Block diagram of OSD control circuit

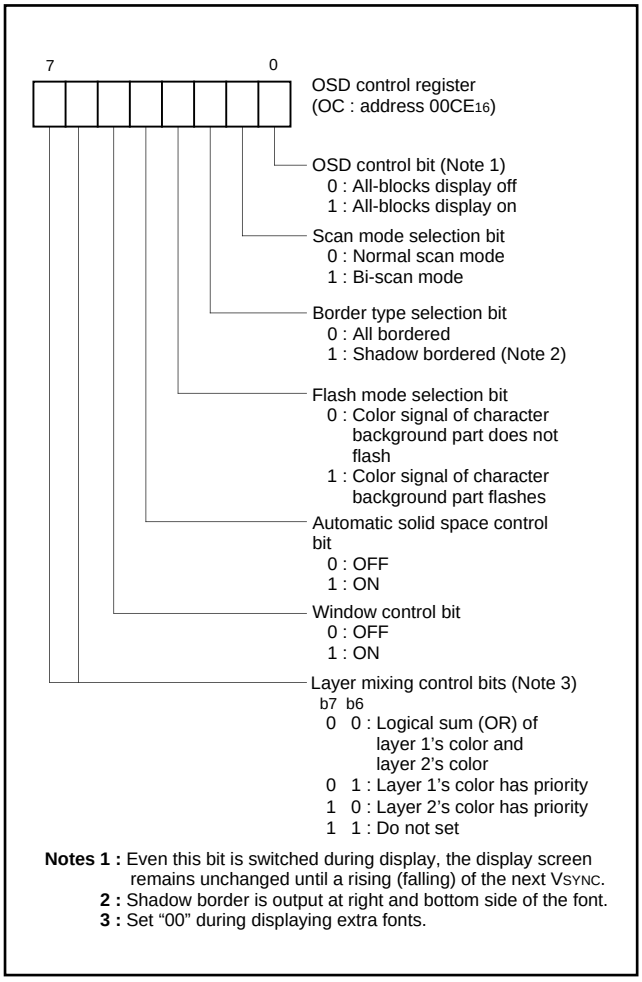


Fig. 49. Structure of OSD control register

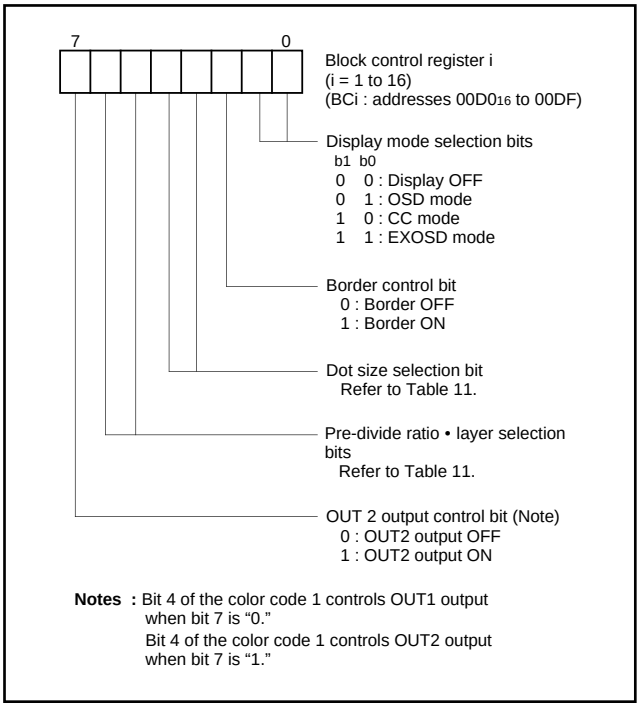


Fig. 50. Structure of block control registers

Table 11. Setting value of block control registers

b6	b5	b4	b3	CS_6	Pre-divide ratio	Dot size	Display layer
0	0	0	0	—	× 1	1Tc × 1/2H	Layer 1
		0	1			1Tc × 1H	
		1	0			2Tc × 2H	
		1	1			3Tc × 3H	
0	1	0	0	—	× 2	1Tc × 1/2H	
		0	1			1Tc × 1H	
		1	0			2Tc × 2H	
		1	1			3Tc × 3H	
1	0	0	0	—	× 3	1Tc × 1/2H	
		0	1			1Tc × 1H	
		1	0			2Tc × 2H	
		1	1			3Tc × 3H	
1	1	—	0	0	× 1	1Tc × 1/2H	Layer 2
		—	1			1Tc × 1H	
1	1	0	0	1	× 2	1Tc × 1/2H	
		0	1			1Tc × 1H	
		1	0			1.5Tc × 1/2H	
		1	1			1.5Tc × 1H	

Notes 1: CS_6 : Bit 6 of clock control register (Address 0216_{16})
2: Tc : OSD clock cycle divided in the pre-divide circuit
3: H : HSYNC

(1) Dual Layer OSD

M37270MF-XXXSP has 2 layers; layer 1 and layer 2. These layers display the OSD for controlling TV and the closed caption display at the same time and overlayed on each other.

Each block can be assigned to either layer by bits 6 and 5 of the block control register (refer to Figure 50). For example, only when both bits 5 and 6 are "1," the block is assigned to layer 2. Other bit combinations assign the block to layer 1.

When a block of layer 1 is overlapped with that of layer 2, a screen is combined (refer to Figure 52) by bits 7 and 6 of the OSD control register (refer to Figure 49).

Note: When using the dual layer OSD, note Table 12.

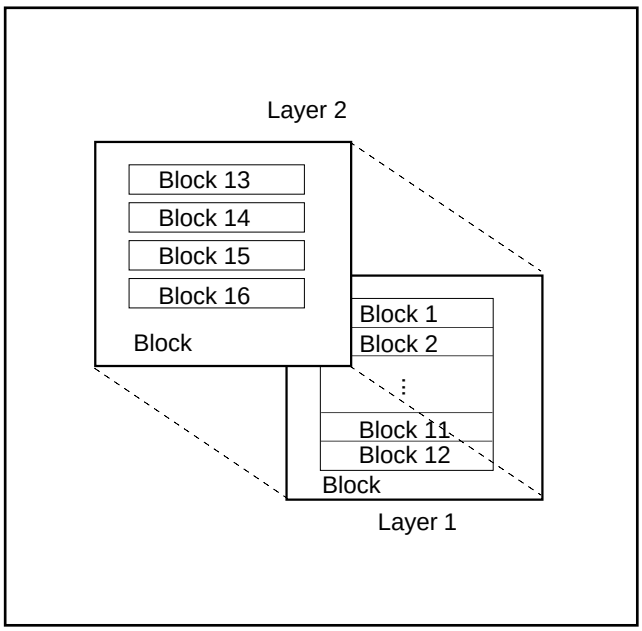


Fig. 51. Image of dual layer OSD

Table 12. Conditions of dual layer

Parameter \ Block	Block in layer 1	Block in layer 2	
Display mode	CC mode	OSD mode	
OSD Clock source	Data slicer clock or OSC1	Same as layer 1	
Pre-divide ratio	× 1 or × 2 (all blocks)	Same as layer 1 (Note)	
Dot size	1Tc × 1/2H	Pre-divide ratio = 1	Pre-divide ratio = 2
		1Tc × 1/2H	1Tc × 1/2H, 1.5Tc × 1/2H
		1Tc × 1H	1Tc × 1H, 1.5Tc × 1H
Horizontal display start position	Arbitrary	Same position as layer 1	

Note: For the pre-divide ratio of the layer 2, select the same as the layer 1's ratio by bit 6 of the clock control register.

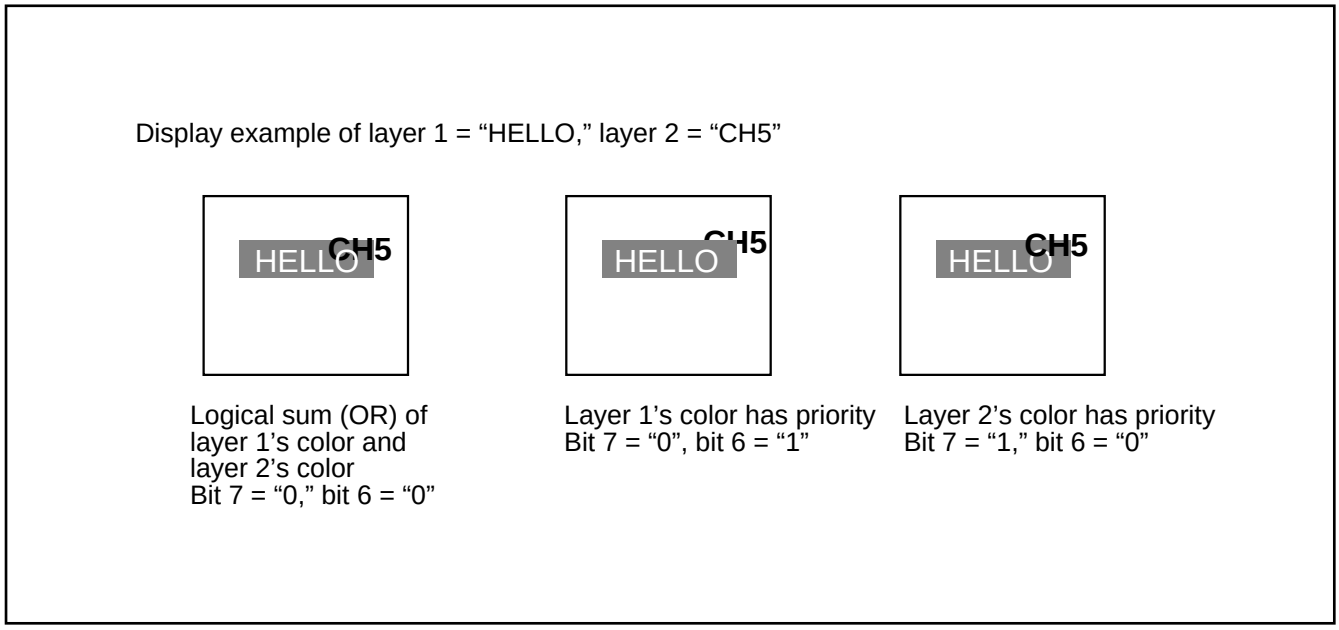


Fig. 52. Display example of dual layer OSD

(2) Display Position

The display positions of characters are specified in units called a "block." There are 16 blocks, blocks 1 to 16. Up to 40 characters can be displayed in each block (refer to (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 T_{OSC} (T_{OSC} = oscillating cycle for OSD).

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

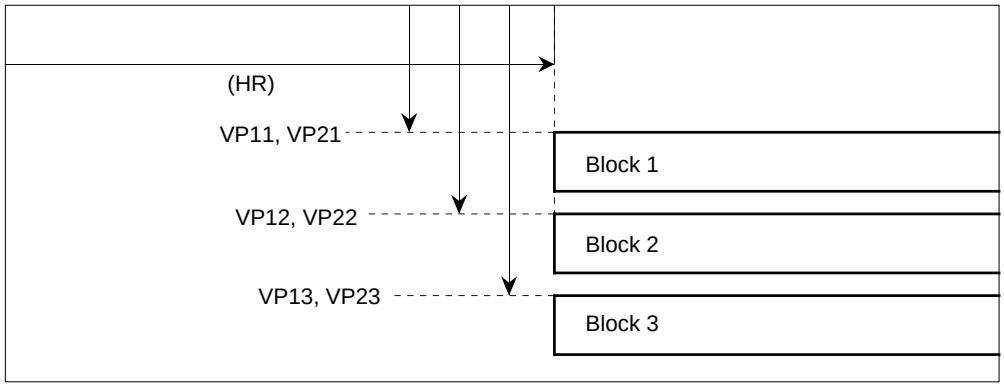
Blocks are displayed in conformance with the following rules:

① When the display position is overlapped with another block (Figure 53, (b)), a lower block number (1 to 16) is displayed on the front.

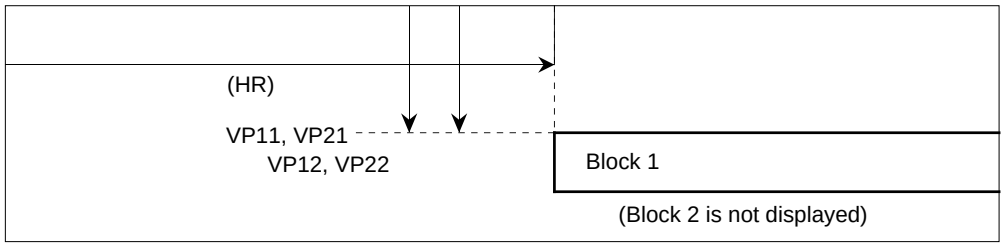
② When another block display position appears while one block is displayed (Figure 53 (c)), the block with a larger set value as the vertical display start position is displayed. However, do not display block with the dot size of $2T_C \times 2H$ or $3T_C \times 3H$ during display period (✻) of another block.

✻ In the case of OSD mode block: 20 dots in vertical from the vertical display start position.

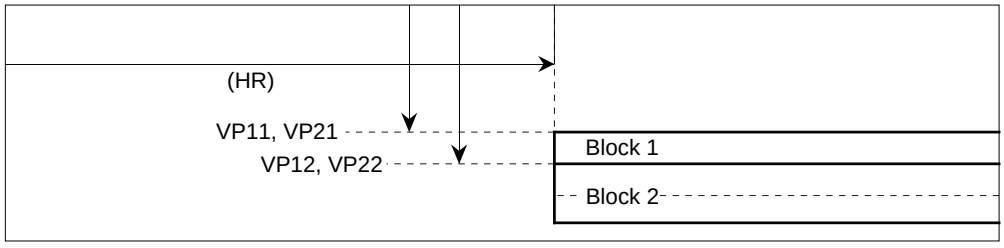
✻ In the case of CC or EXOSD mode block: 26 dots in vertical from the vertical display start position.



(a) Example when each block is separated



(b) Example when block 2 overlaps with block 1



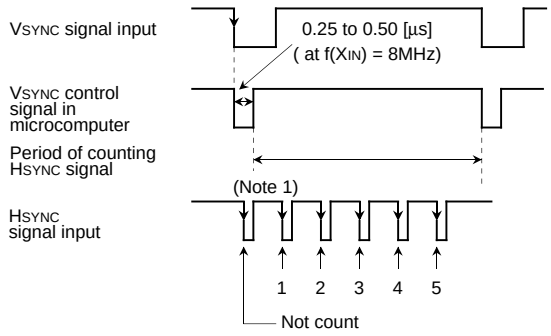
(c) Example when block 2 overlaps in process of block 1

Note: VP1i or VP2i (i : 1 to 16) indicates the contents of vertical position registers 1i or 2i.

Fig. 53. Display position

The display position in the vertical direction is determined by counting the horizontal sync signal (HSYNC). At this time, it starts to count the rising edge (falling edge) of HSYNC signal from after about 1 machine 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 machine cycles or more) for avoiding jitter. The polarity of HSYNC and VSYNC signals can select with the I/O polarity control register (address 0217_{16}). For details, refer to (15) OSD Output Pin Control.

Note: When bits 0 and 1 of the I/O polarity control register (address 0217_{16}) are set to "1" (negative polarity), the vertical position is determined by counting falling edge of HSYNC signal after rising edge of VSYNC control signal in the microcomputer (refer to Figure 54).



When bits 0 and 1 of the I/O polarity control register (address 0217_{16}) are set to "1" (negative polarity)

Notes 1 : Do not generate falling edge of HSYNC signal near rising edge of VSYNC control signal in microcomputer to avoid jitter.
2 : The pulse width of VSYNC and HSYNC needs 8 machine cycles or more.

Fig. 54. Supplement explanation for display position

The vertical position for each block can be set in 1024 steps (where each step is $1T_H$ (T_H: HSYNC cycle)) as values "00_{16}" to "FF_{16}" in vertical position register 1i (i = 1 to 16) (addresses 0220_{16} to $022F_{16}$) and values "00_{16}" to "FF_{16}" in the vertical position register 2i (i = 1 to 16) (addresses 0230_{16} to $023F_{16}$). The structure of the vertical position registers is shown in Figure 55.

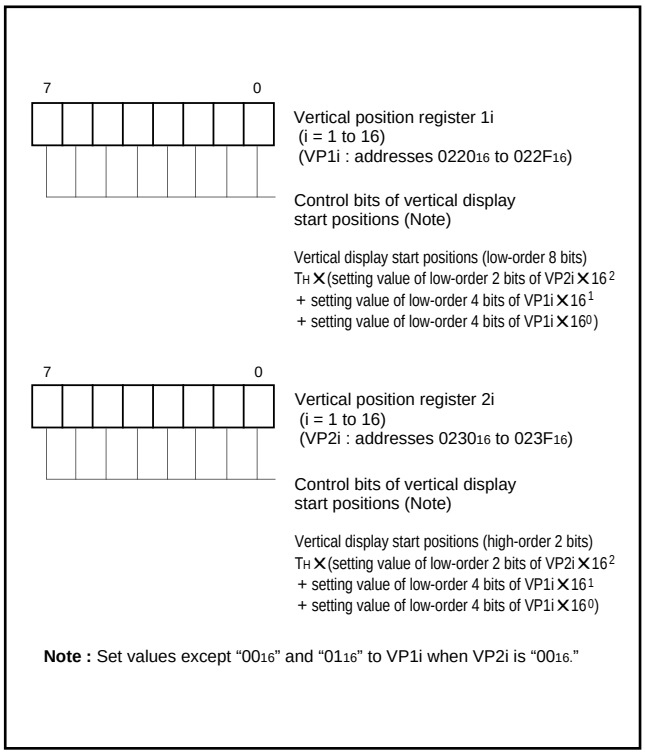


Fig. 55. Structure of vertical position registers

The horizontal position is common to all blocks, and can be set in 256 steps (where 1 step is $4T_{OSC}$, T_{OSC} being the oscillating cycle for display) as values "00_{16}" to "FF_{16}" in bits 0 to 7 of the horizontal position register (address $00CF_{16}$). The structure of the horizontal position register is shown in Figure 56.

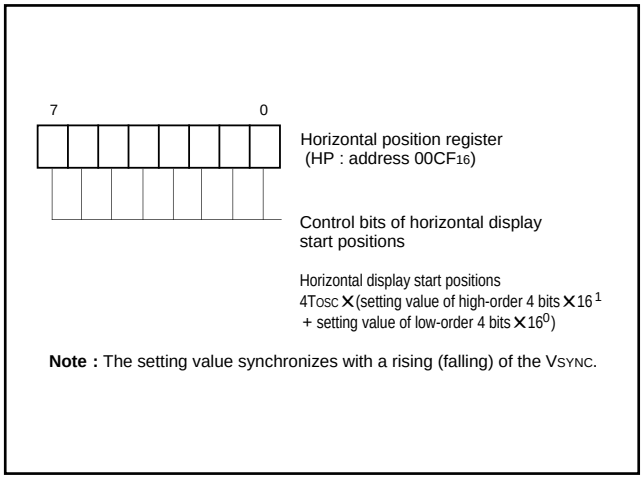


Fig. 56. Structure of horizontal position register

Notes 1 : 1Tc (Tc : OSD clock cycle divided by prescaler) 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.

2 : The horizontal start position is based on the OSD clock source cycle selected for each block. Accordingly, when 2 blocks have different OSD clock source cycles, their horizontal display start position will not match.

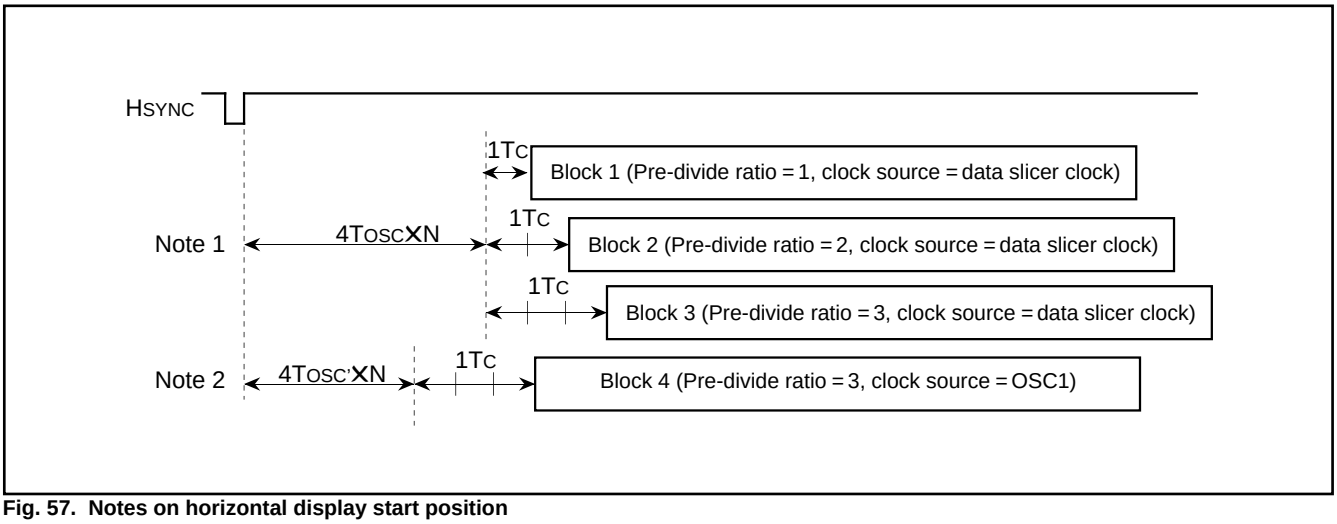


Fig. 57. Notes on horizontal display start position

(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) in the pre-divide circuit. The clock cycle divided in the pre-divide circuit is defined as 1Tc.

The dot size of the layer 1 is specified by bits 6 to 3 of the block control register.

The dot size of the layer 2 is specified by the following bits : bits 3 and 4 of the block control register, bit 6 of the clock source control register. Refer to Figure 50 (the structure of the block control register), refer to Figure 59 (the structure of the clock source control register).

The block diagram of dot size control circuit is shown in Figure 58.

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

2 : The pre-divide ratio of the OSD mode block on the layer 2 must be same as that of the CC mode block on the layer 1 by bit 6 of the clock source control register.

3 : In the bi-scan mode, the dot size in the vertical direction is 2 times as compared with the normal mode. Refer to "(13) Scan Mode" about the scan mode.

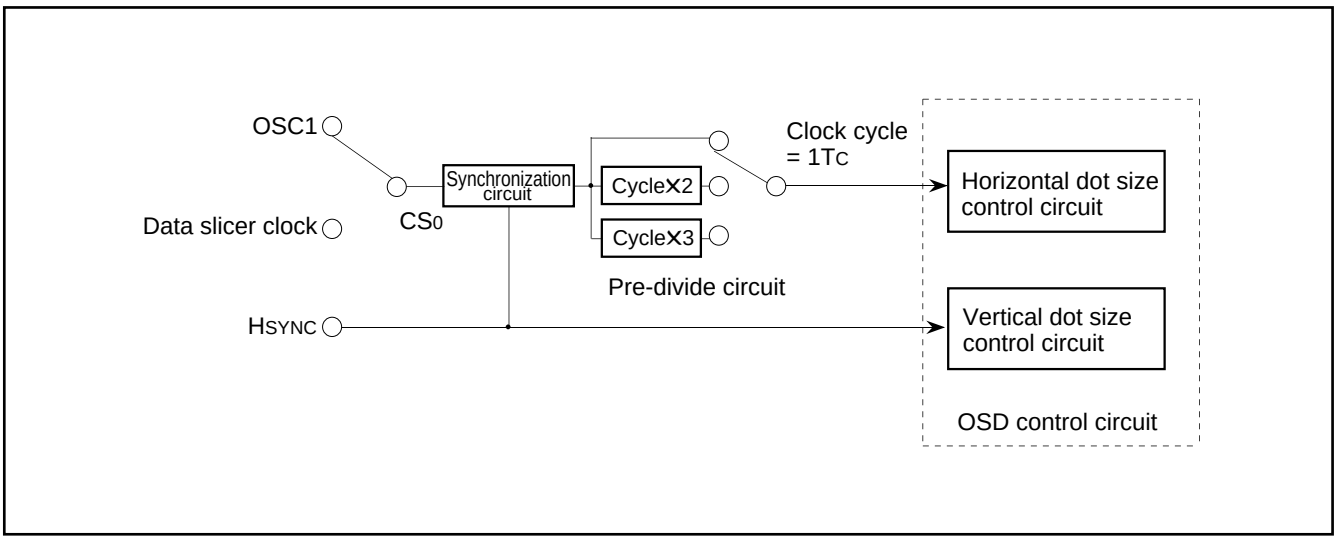


Fig. 58. Block diagram of dot size control circuit

(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

This OSD clock for each block can be selected by the following bits : bit 7 of the port P3 direction register, bits 5 and 4 of the clock source control register (addresses 021616). A variety of character sizes can be obtained by combining dot sizes with OSD clocks. When not using the pins OSC1 and OSC2 for the OSD clock I/O pins, the pins can be used as sub-clock I/O pins or port P6.

Table 13. Setting for $P6_3$/OSC1/X_{CIN}, $P6_4$/OSC2/X_{COUT}

Register \ Function		OSD clock I/O pin			Sub-clock I/O pin	Input port
b7 Port P3 direction register		0			0	1
Clock source control register	b5	0	1	1	0	0
	b4	1	0	1	0	1

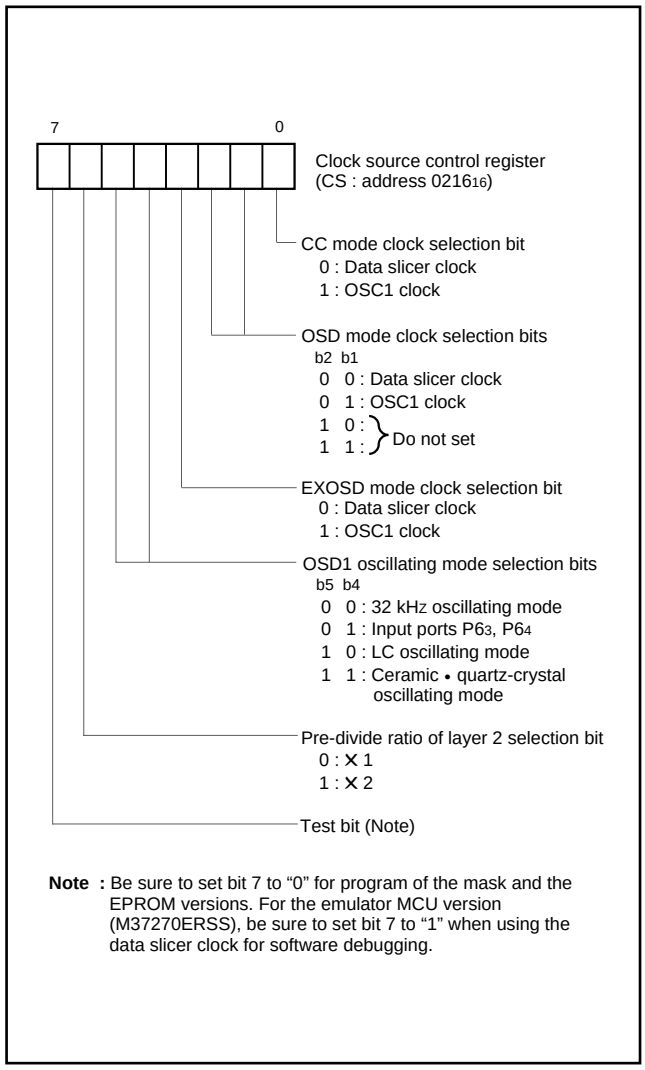


Fig. 59. Structure of clock control register

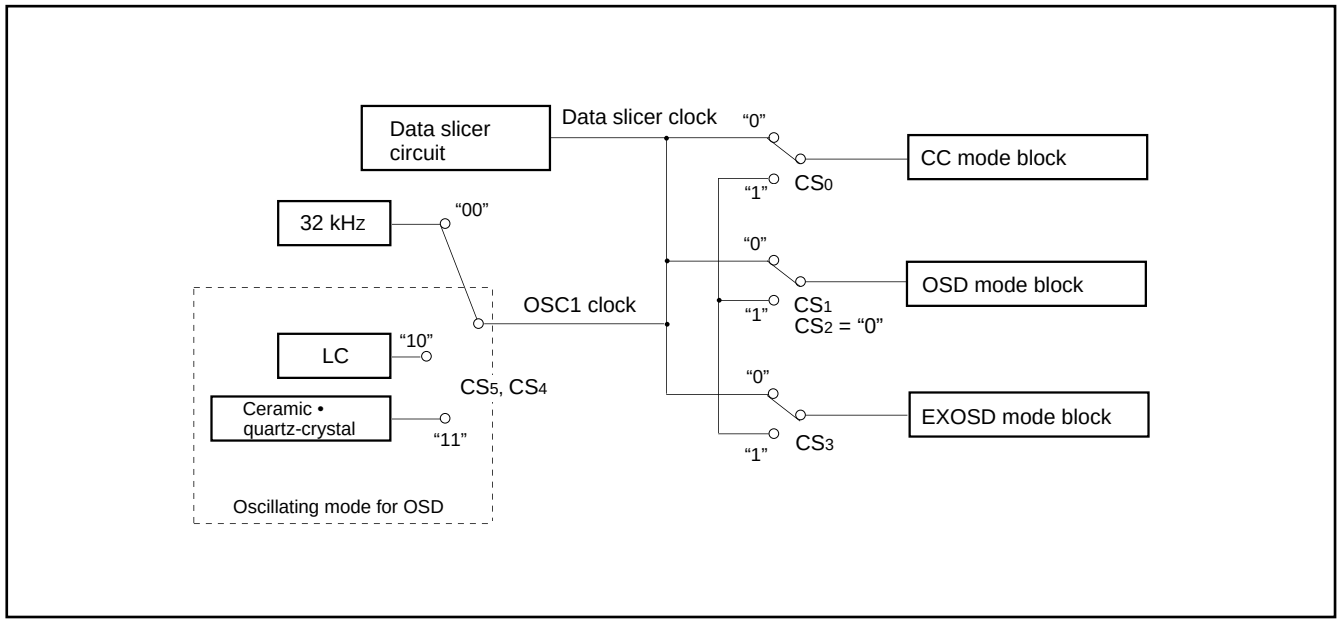


Fig. 60. Block diagram of OSD selection circuit

(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 62) 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 54) 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 contents of this field can be read out by the field determination flag (bit 7 of the I/O polarity control register at address 0217_{16}). A dot line is specified by bit 6 of the I/O polarity control register (refer to Figure 62).

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.

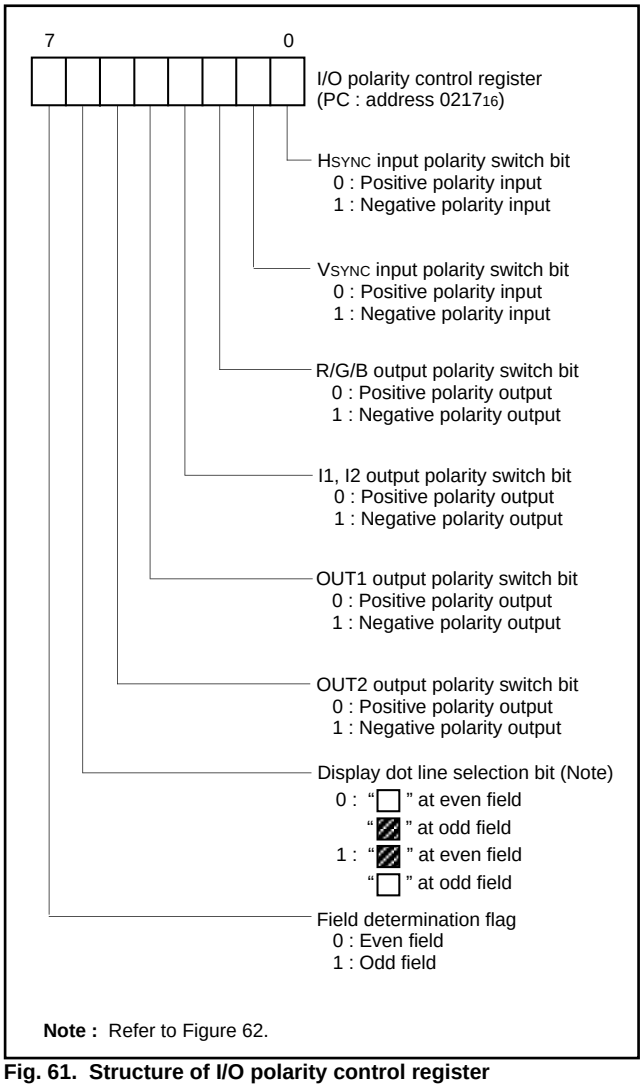
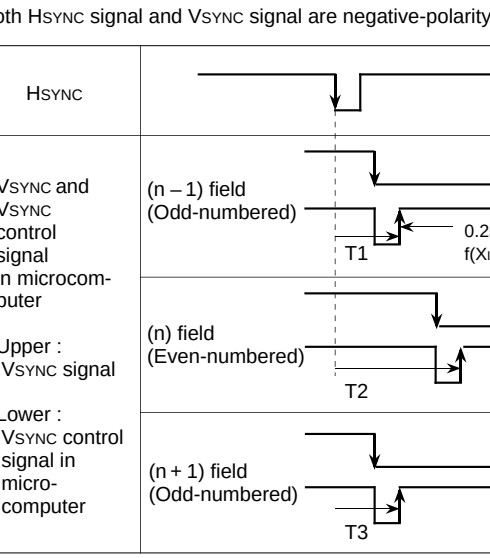


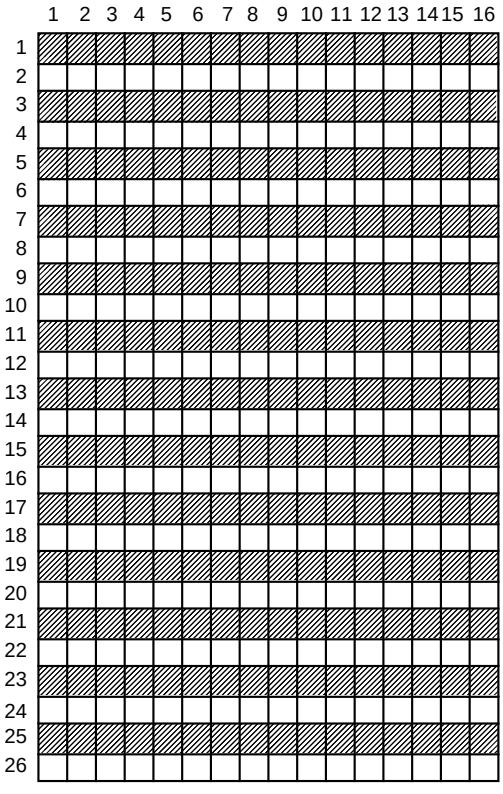
Fig. 61. Structure of I/O polarity control register

Both HSYNC signal and VSYNC signal are negative-polarity input

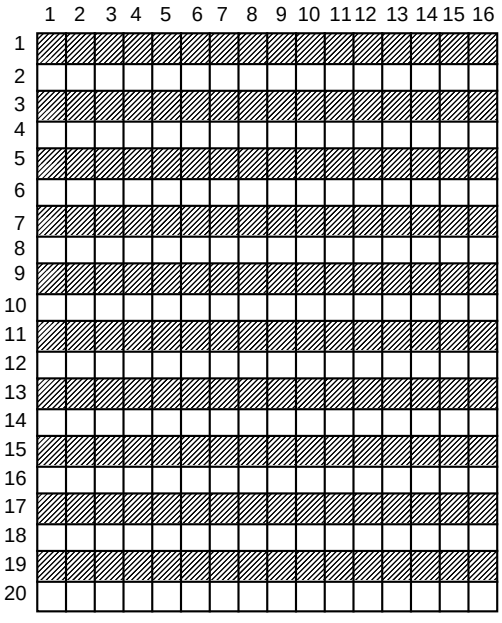
HSYNC		Field	Field determination flag(Note)	Display dot line selection bit	Display dot line
VSYNC and VSYNC control signal in microcomputer Upper : VSYNC signal Lower : VSYNC control signal in microcomputer	(n – 1) field (Odd-numbered)	Odd			
	(n) field (Even-numbered)	Even	0 (T2 > T1)	0	Dot line 1 □
				1	Dot line 0 ▨
	(n + 1) field (Odd-numbered)	Odd	1 (T3 < T2)	0	Dot line 0 ▨
				1	Dot line 1 □



When using the field determination flag, be sure to set bit 0 of the PWM mode register 1 (address 020A16) to “0.”



CC mode • EXOSD mode



OSD mode

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

Character 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.

Fig. 62. Relation between field determination flag and display font

(6) Memory for OSD

There are 2 types of memory for OSD : ROM for OSD (addresses 1080016 to $1567F16$, 1800016 to $1E43F16$) used to store character dot data (masked) and RAM for OSD (addresses 080016 to $0FFF16$) used to specify the characters and colors to be displayed. The following describes each type of memory.

① **ROM for OSD (addresses 1080016 to $1567F16$, 1800016 to $1E43F16$)**

The ROM for OSD contains dot pattern data for characters to be displayed. To actually display the character code and the extra code stored in this ROM, it is necessary to specify them by writing the character code inherent to each character (code determined based on the addresses in the ROM for OSD) into the RAM for OSD.

The OSD ROM of the character font has a capacity of 12800 bytes. Since 40 bytes are required for 1 character data, the ROM can stores up to 320 kinds of characters. The OSD ROM of the extra font has a capacity of 1664 bytes. Since 52 bytes are required for 1 character data, the ROM can stores up to 32 kinds of characters.

Data of the character font and extra font is specified shown in Figure 63.

OSD ROM address of character font data

OSD ROM address bit	AD16	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Line number/character code/font bit	1	0	Line number					Character code									Font bit

Line number = "0216" to "1516"
Character code = "0016" to "$13F16$"
Font bit = 0 : left font 1 : right font

OSD ROM address of extra font data

OSD ROM address bit	AD16	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Line number/extra code /font bit	1	1	Line number					0	0	0	0	Extra code					Font bit

Line number = "0016" to "1916"
Extra code = "0016" to "$1F16$"
Font bit = 0 : left font 1 : right font

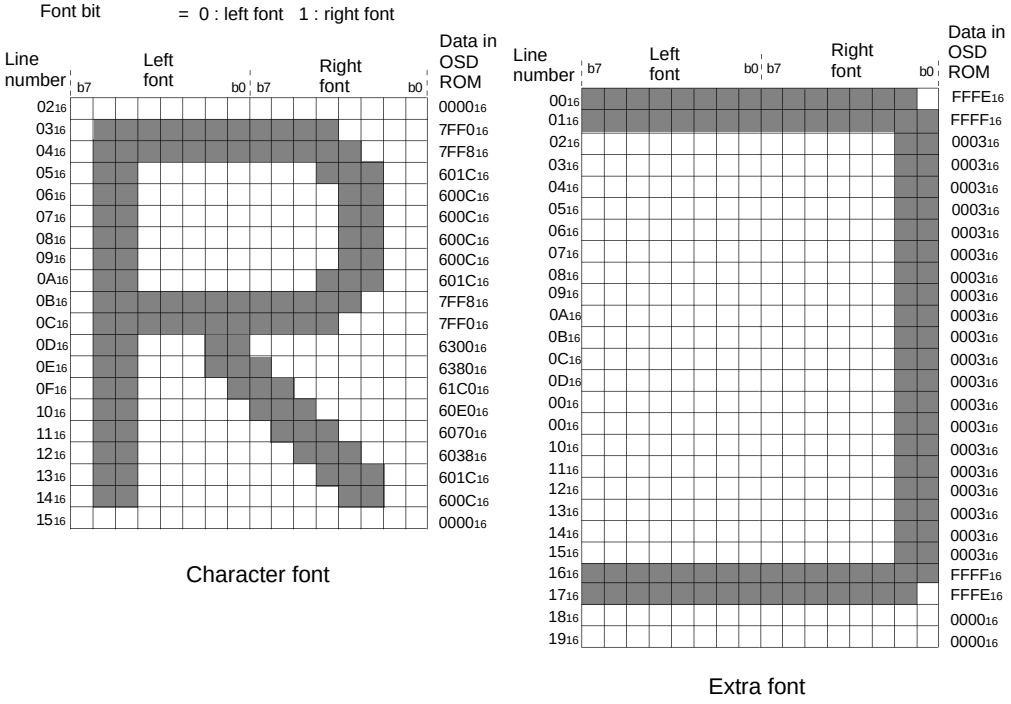


Fig. 63. OSD character data storing form

② **RAM for OSD (addresses 0800_{16} to $0FFF_{16}$)**

The RAM for OSD is allocated at addresses 0800_{16} to $0FFF_{16}$, and is divided into a display character code specification part, color code 1 specification part, and color code 2 specification part for each block. Table 14 shows the contents of the RAM for OSD.

For example, to display 1 character position (the left edge) in block 1, write the character code in address 0800_{16}, write the color code 1 at 0840_{16}, and write the color code 2 at 0828_{16}.

The structure of the RAM for OSD is shown in Figure 65.

Note: For the OSD mode block with dot size of 1.5Tc × 1/2H and 1.5Tc × 1H, the 3nth (n = 1 to 13) character is skipped as compared with ordinary block*. Accordingly, maximum 26 characters are only displayed in 1 block. The RAM data for the 3nth character does not effect the display. Any character data can be stored here.

✻ Blocks with dot size of 1Tc × 1/2H and 1Tc × 1H, or blocks on the layer 1

Table 14. Contents of OSD RAM

Block	Display position (from left)	Character code specification	Color code 1 specification	Color code 2 specification
Block 1	1st character	0800_{16}	0840_{16}	0828_{16}
	2nd character	0801_{16}	0841_{16}	0829_{16}
	:	:	:	:
	24th character	0817_{16}	0857_{16}	$083F_{16}$
	25th character	0818_{16}	0858_{16}	0868_{16}
	:	:	:	:
	39th character	0826_{16}	0866_{16}	0876_{16}
	40th character	0827_{16}	0867_{16}	0877_{16}
Block 2	1st character	0880_{16}	$08C0_{16}$	$08A8_{16}$
	2nd character	0881_{16}	$08C1_{16}$	$08A9_{16}$
	:	:	:	:
	24th character	0897_{16}	$08D7_{16}$	$08BF_{16}$
	25th character	$0E98_{16}$	$08D8_{16}$	$08E8_{16}$
	:	:	:	:
	39th character	$08A6_{16}$	$08E6_{16}$	$08F6_{16}$
	40th character	$08A7_{16}$	$08E7_{16}$	$08F7_{16}$
Block 3	1st character	0900_{16}	0940_{16}	0928_{16}
	2nd character	0901_{16}	0941_{16}	0929_{16}
	:	:	:	:
	24th character	0917_{16}	0957_{16}	$093F_{16}$
	25th character	0918_{16}	0958_{16}	0968_{16}
	:	:	:	:
	39th character	0926_{16}	0966_{16}	0976_{16}
	40th character	0927_{16}	0967_{16}	0977_{16}
Block 4	1st character	0980_{16}	$09C0_{16}$	$09A8_{16}$
	2nd character	0981_{16}	$09C1_{16}$	$09A9_{16}$
	:	:	:	:
	24th character	0997_{16}	$09D7_{16}$	$09BF_{16}$
	25th character	0998_{16}	$08D8_{16}$	$09E8_{16}$
	:	:	:	:
	39th character	$09A6_{16}$	$09E6_{16}$	$09F6_{16}$
	40th character	$09A7_{16}$	$09E7_{16}$	$09F7_{16}$
Block 5	1st character	$0A00_{16}$	$0A40_{16}$	$0A28_{16}$
	2nd character	$0A01_{16}$	$0A41_{16}$	$0A29_{16}$
	:	:	:	:
	24th character	$0A17_{16}$	$0A57_{16}$	$0A3F_{16}$
	25th character	$0A18_{16}$	$0A58_{16}$	$0A68_{16}$
	:	:	:	:
	39th character	$0A26_{16}$	$0A66_{16}$	$0A76_{16}$
	40th character	$0A27_{16}$	$0A67_{16}$	$0A77_{16}$

Table 14. Contents of OSD RAM (continued)

Block	Display position (from left)	Character code specification	Color code 1 specification	Color code 2 specification
Block 6	1st character	$0A80_{16}$	$0AC0_{16}$	$0AA8_{16}$
	2nd character : 24th character	$0A81_{16}$: $0A97_{16}$	$0AC1_{16}$: $0AD7_{16}$	$0AA9_{16}$: $0ABF_{16}$
	25th character : 39th character	$0A98_{16}$: $0AA6_{16}$	$0AD8_{16}$: $0AE6_{16}$	$0AE8_{16}$: $0AF6_{16}$
	40th character	$0AA7_{16}$	$0AE7_{16}$	$0AF7_{16}$
Block 7	1st character	$0B00_{16}$	$0B40_{16}$	$0B28_{16}$
	2nd character : 24th character	$0B01_{16}$: $0B17_{16}$	$0B41_{16}$: $0B57_{16}$	$0B29_{16}$: $0B3F_{16}$
	25th character : 39th character	$0B18_{16}$: $0B26_{16}$	$0B58_{16}$: $0B66_{16}$	$0B68_{16}$: $0B76_{16}$
	40th character	$0B27_{16}$	$0B67_{16}$	$0B77_{16}$
Block 8	1st character	$0B80_{16}$	$0BC0_{16}$	$0BA8_{16}$
	2nd character : 24th character	$0B81_{16}$: $0B97_{16}$	$0BC1_{16}$: $0BD7_{16}$	$0BA9_{16}$: $0BBF_{16}$
	25th character : 39th character	$0B98_{16}$: $0BA6_{16}$	$0BD8_{16}$: $0BE6_{16}$	$0BE8_{16}$: $0BF6_{16}$
	40th character	$0BA7_{16}$	$0BE7_{16}$	$0BF7_{16}$
Block 9	1st character	$0C00_{16}$	$0C40_{16}$	$0C28_{16}$
	2nd character : 24th character	$0C01_{16}$: $0C17_{16}$	$0C41_{16}$: $0C57_{16}$	$0C29_{16}$: $0C3F_{16}$
	25th character : 39th character	$0C18_{16}$: $0C26_{16}$	$0C58_{16}$: $0C66_{16}$	$0C68_{16}$: $0C76_{16}$
	40th character	$0C27_{16}$	$0C67_{16}$	$0C77_{16}$
Block 10	1st character	$0C80_{16}$	$0CC0_{16}$	$0CA8_{16}$
	2nd character : 24th character	$0C81_{16}$: $0C97_{16}$	$0CC1_{16}$: $0CD7_{16}$	$0CA9_{16}$: $0CBF_{16}$
	25th character : 39th character	$0C98_{16}$: $0CA6_{16}$	$0CD8_{16}$: $0CE6_{16}$	$0CE8_{16}$: $0CF6_{16}$
	40th character	$0CA7_{16}$	$0CE7_{16}$	$0CF7_{16}$
Block 11	1st character	$0D00_{16}$	$0D40_{16}$	$0D28_{16}$
	2nd character : 24th character	$0D01_{16}$: $0D17_{16}$	$0D41_{16}$: $0D57_{16}$	$0D29_{16}$: $0D3F_{16}$
	25th character : 39th character	$0D18_{16}$: $0D26_{16}$	$0D58_{16}$: $0D66_{16}$	$0D68_{16}$: $0D76_{16}$
	40th character	$0D27_{16}$	$0D67_{16}$	$0D77_{16}$

Table 14. Contents of OSD RAM (continued)

Block	Display position (from left)	Character code specification	Color code 1 specification	Color code 2 specification
Block 12	1st character	0D80$_{16}$	0DC0$_{16}$	0DA8$_{16}$
	2nd character : 24th character	0D81$_{16}$: 0D97$_{16}$	0DC1$_{16}$: 0DD7$_{16}$	0DA9$_{16}$: 0DBF$_{16}$
	25th character : 39th character	0D98$_{16}$: 0DA6$_{16}$	0DD8$_{16}$: 0DE6$_{16}$	0DE8$_{16}$: 0DF6$_{16}$
	40th character	0DA7$_{16}$	0DE7$_{16}$	0DF7$_{16}$
Block 13	1st character	0E00$_{16}$	0E40$_{16}$	0E28$_{16}$
	2nd character : 24th character	0E01$_{16}$: 0E17$_{16}$	0E41$_{16}$: 0E57$_{16}$	0E29$_{16}$: 0E3F$_{16}$
	25th character : 39th character	0E18$_{16}$: 0E26$_{16}$	0E58$_{16}$: 0E66$_{16}$	0E68$_{16}$: 0E76$_{16}$
	40th character	0E27$_{16}$	0E67$_{16}$	0E77$_{16}$
Block 14	1st character	0E80$_{16}$	0EC0$_{16}$	0EA8$_{16}$
	2nd character : 24th character	0E81$_{16}$: 0E98$_{16}$	0EC1$_{16}$: 0ED7$_{16}$	0EA9$_{16}$: 0EBF$_{16}$
	25th character : 39th character	0E99$_{16}$: 0EA6$_{16}$	0ED8$_{16}$: 0EE6$_{16}$	0EE8$_{16}$: 0EF6$_{16}$
	40th character	0EA7$_{16}$	0EE7$_{16}$	0EF7$_{16}$
Block 15	1st character	0F00$_{16}$	0F40$_{16}$	0F28$_{16}$
	2nd character : 24th character	0F01$_{16}$: 0F17$_{16}$	0F41$_{16}$: 0F57$_{16}$	0F29$_{16}$: 0F3F$_{16}$
	25th character : 39th character	0F18$_{16}$: 0F26$_{16}$	0F58$_{16}$: 0F66$_{16}$	0F68$_{16}$: 0F76$_{16}$
	40th character	0F27$_{16}$	0F67$_{16}$	0F77$_{16}$
Block 16	1st character	0F80$_{16}$	0FC0$_{16}$	0FA8$_{16}$
	2nd character : 24th character	0F81$_{16}$: 0F97$_{16}$	0FC1$_{16}$: 0FD7$_{16}$	0FA9$_{16}$: 0FBF$_{16}$
	25th character : 39th character	0F98$_{16}$: 0FA6$_{16}$	0FD8$_{16}$: 0FE6$_{16}$	0FE8$_{16}$: 0FF6$_{16}$
	40th character	0FA7$_{16}$	0FE7$_{16}$	0FF7$_{16}$

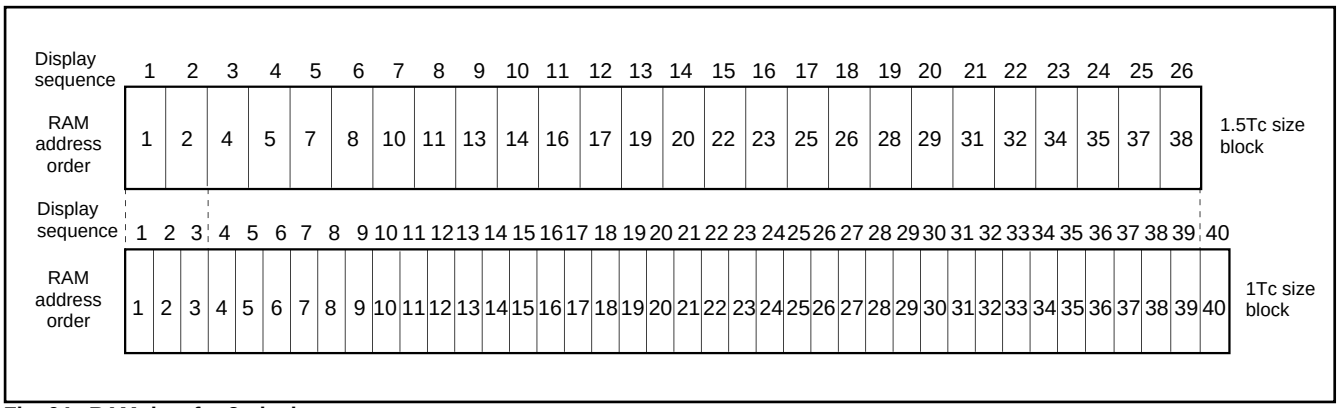


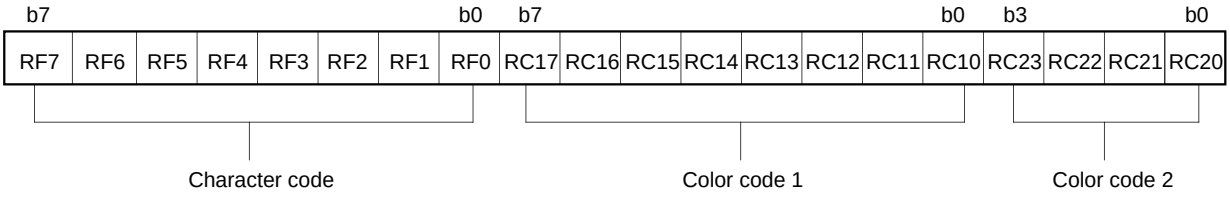
Fig. 64. RAM data for 3nth character

Note: Do not read from and write to addresses in OSD RAM shown in Table 15.

Table 15. List of access disable addresses

0878_{16}	0879_{16}	$087A_{16}$
$08F8_{16}$	$08F9_{16}$	$08FA_{16}$
0978_{16}	0979_{16}	$097A_{16}$
$09F8_{16}$	$09F9_{16}$	$09FA_{16}$
$0A78_{16}$	$0A79_{16}$	$0A7A_{16}$
$0AF8_{16}$	$0AF9_{16}$	$0AFA_{16}$
$0B78_{16}$	$0B79_{16}$	$0B7A_{16}$
$0BF8_{16}$	$0BF9_{16}$	$0BFA_{16}$
$0C78_{16}$	$0C79_{16}$	$0C7A_{16}$
$0CF8_{16}$	$0CF9_{16}$	$0CFA_{16}$
$0D78_{16}$	$0D79_{16}$	$0D7A_{16}$
$0DF8_{16}$	$0DF9_{16}$	$0DFA_{16}$
$0E78_{16}$	$0E79_{16}$	$0E7A_{16}$
$0EF8_{16}$	$0EF9_{16}$	$0EFA_{16}$
$0F78_{16}$	$0F79_{16}$	$0F7A_{16}$
$0FF8_{16}$	$0FF9_{16}$	$0FFA_{16}$

Blocks 1 to16



Bit	CC mode		OSD mode		EXOSD mode	
	Bit name	Function	Bit name	Function	Bit name	Function
RF0 RF1 RF2 RF3 RF4 RF5 RF6 RF7	Character code (Low-order 8 bits)	Specification of character code in OSD ROM	Character code (Low-order 8 bits)	Specification of character code in OSD ROM	Character code (Low-order 8 bits)	Specification of character code in OSD ROM
RC10	Character code (High-order 1 bit)		Character code (High-order 1 bit)		Character code (High-order 1 bit)	
RC11 RC12 RC13	Control of character color R Control of character color G Control of character color B	0: Color signal output OFF 1: Color signal output ON	Control of character color R Control of character color G Control of character color B	0: Color signal output OFF 1: Color signal output ON	Character color code 0 (CC0) Character color code 1 (CC1) Character color code 2 (CC2)	Specification of character color
RC14	OUT1 control	0: Character output 1: Background output	OUT1 control	0: Character output 1: Background output	OUT1 control	0: Character output 1: Background output
RC15	Flash control	0: Flash OFF 1: Flash ON	Control of character color I1	0: Color signal output OFF 1: Color signal output ON	Extra code 0 (EX0)	Specification of extra code in OSD ROM
RC16	Underline control	0: Underline OFF 1: Underline ON	Not used	———	Extra code 1 (EX1)	
RC17	Italic control	0: Italic OFF 1: Italic ON			Extra code 2 (EX2)	
RC20 RC21 RC22	Control of background color R Control of background color G Control of background color B	0: Color signal output OFF 1: Color signal output ON	Control of background color R Control of background color G Control of background color B	0: Color signal output OFF 1: Color signal output ON	Background color code 0 (BCC0) Background color code 1 (BCC1) Background color code 2 (BCC2)	Specification of background color
RC23	Not used	———	Control of background color I1		Extra code 3 (EX3)	Specification of extra code in OSD ROM

Notes 1: Read value of bits 4 to 7 of the color code 2 is undefined.

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

3: The decode value of the extra code is “EX4.”

Fig. 65. Structure of OSD RAM

(7) Character color

The color for each character is displayed by the color code 1. The kinds and specification method of character color are different depending on each mode.

- CC mode ..... 7 kinds
  Specified by bits 1 (R), 2 (G), and 3 (B) of the color code 1
- OSD mode ..... 15 kinds
  Specified by bits 1 (R), 2 (G), 3 (B), and 5 (I1) of the color code 1
- EXOSD mode ..... 7 kinds
  Specified by bits 1 (CC0), 2 (CC1), and 3 (CC2) of the color code 1

The correspondence Table of the color code 1 and color signal output in the EXOSD mode is shown in Table 16.

(8) Character background color

The character background color can be displayed in the character display area. The character background color for each character is specified by the color code 2. The kinds and specification method of character background color are different depending on each mode.

- CC mode ..... 7 kinds
  Specified by bits 0 (R), 1 (G), and 2 (B) of the color code 2
- OSD mode ..... 15 kinds
  Specified by bits 0 (R), 1 (G), 2 (B), and 3 (I1) of the color code 2
- EXOSD mode ..... 7 kinds
  Specified by bits 0 (BCC0), 1 (BCC1), and 2 (BCC2) of the color code 2

The correspondence table of the color code 2 and color signal output in the EXOSD mode is shown in Table 17.

Note : The character background color is displayed in the following part :
(character display area)–(character font)–(border)–(extra font).
Accordingly, the character background color does not mix with these color signal.

Table 16. Correspondence table of color code 1 and color signal output in EXOSD mode

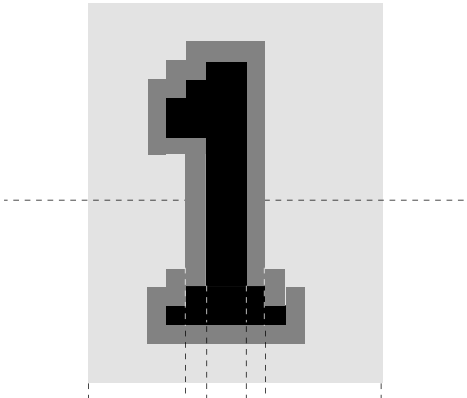
Color code 1			Color signal output				
Bit 3 CC2	Bit 2 CC1	Bit 1 CC0	R	G	B	I1	I2
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	0	0
0	1	1	1	1	0	1	0
1	0	0	1	1	0	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	0	0
1	1	1	1	1	1	0	0

Table 17. Correspondence table of color code 2 and color signal output in EXOSD mode

Color code 2			Color signal output				
Bit 2 BCC2	Bit 1 BCC1	Bit 0 BCC0	R	G	B	I1	I2
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	0	0
0	1	1	1	1	0	1	0
1	0	0	1	1	0	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	0	0
1	1	1	1	1	1	0	0

(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 4 of the color code 1 (refer to Figure 65), bits 2 and 7 of the block control register (refer to Figure 50). The setting values for controlling OUT1, OUT2 and the corresponding output waveform is shown in Figure 66.



Block control register		OUT1 control (b4 of color code 1)	Output waveform
OUT2 output control bit (b7)	Border output control bit (b2)		
0	0	0	OUT1 OUT2
		1	OUT1 OUT2
	1	0	OUT1 OUT2
		1	OUT1 OUT2
1	0	0	OUT1 OUT2
		1	OUT1 OUT2
	1	0	OUT1 OUT2
		1	OUT1 OUT2

Fig. 66. Setting value for controlling OUT1, OUT2 and corresponding output waveform

(10) Attribute

The attributes (flash, underline, italic) are controlled to the character font. The attributes for each character are specified by the color codes 1 and 2 (refer to Figure 65). The attributes to be controlled are different depending on each mode.

CC mode ..... Flash, underline, italic
OSD mode ..... Border (all bordered, shadow bordered can be selected)
EXOSD mode ..... Border (all bordered, shadow bordered can be selected) , extra font (32 kinds)

① **Under line**

The underline is output at the 23th and 24th dots in vertical direction only in the CC mode. The underline is controlled by bit 6 of the color code 1. 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 color signals (R, G, B, OUT1) of the character font and the underline are controlled by bit 5 of the color code 1. All of the color signals for the character font flash. However, the color signal for the character background can be controlled by bit 3 of the OSD control register (refer to Figure 49). The flash cycle bases on the VSYNC count.

- VSYNC cycle × 48] 800 ms (at flash ON)
- VSYNC cycle × 16] 267 ms (at flash OFF)

③ **Italic**

The italic is made by slanting the font stored in OSD ROM only in the CC mode. The italic is controlled by bit 7 of the color code 1.

The display example of the italic and underline is shown in Figure 67. In this case, 16 26 dots are used and “R” is displayed.

Notes 1: When setting both the italic and the flash, the italic character flashes.
2: When the pre-divide ratio = 1, the italic character with slant of 1 dot × 5 steps is displayed (refer to Figure 68 (c)). When the pre-divide ratio = 2, the italic character with slant of 1/2 dot × 10 steps is displayed (refer to Figure 68 (d)).
3: 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 69).
4: 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 69).
5: When displaying the italic character in the block with the pre-divide ratio = 1, set the OSD clock frequency to 11 MHz to 14 MHz.

④ **Extra font**

There are 32 kinds of the extra fonts configured with 16 × 26 dots in OSD ROM. 16 kinds of these fonts can be displayed by ORed with the character font by a character unit (refer to Figure 47). For the others, only the extra font is displayed (refer to Figure 47). In only the EXOSD mode, the extra font is controlled the following : bits 7 to 5 of the color code 1, bit 3 of the color code 2, and decode value (EX4) of the character code. When the character code = “00_{16}” to “$13F_{16}$,” EX4 is “0, ” when the character code = “140_{16},” EX4 is “1.” Since there is no font with the character code = “140_{16},” a blank is displayed.

The extra font color for each screen is specified by the extra color register. When the character font overlaps with the extra font, the color of the area becomes the ORed color of both fonts.

Note : When using the extra font, set bits 7 and 6 of the OSD control register to “0” (refer to Figure 49).

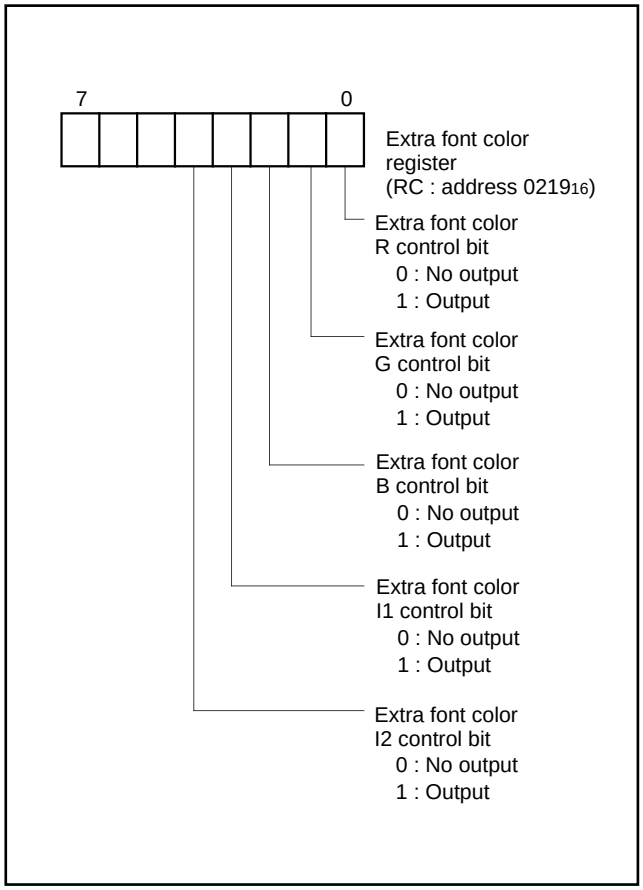


Fig. 67. Structure of extra font color register

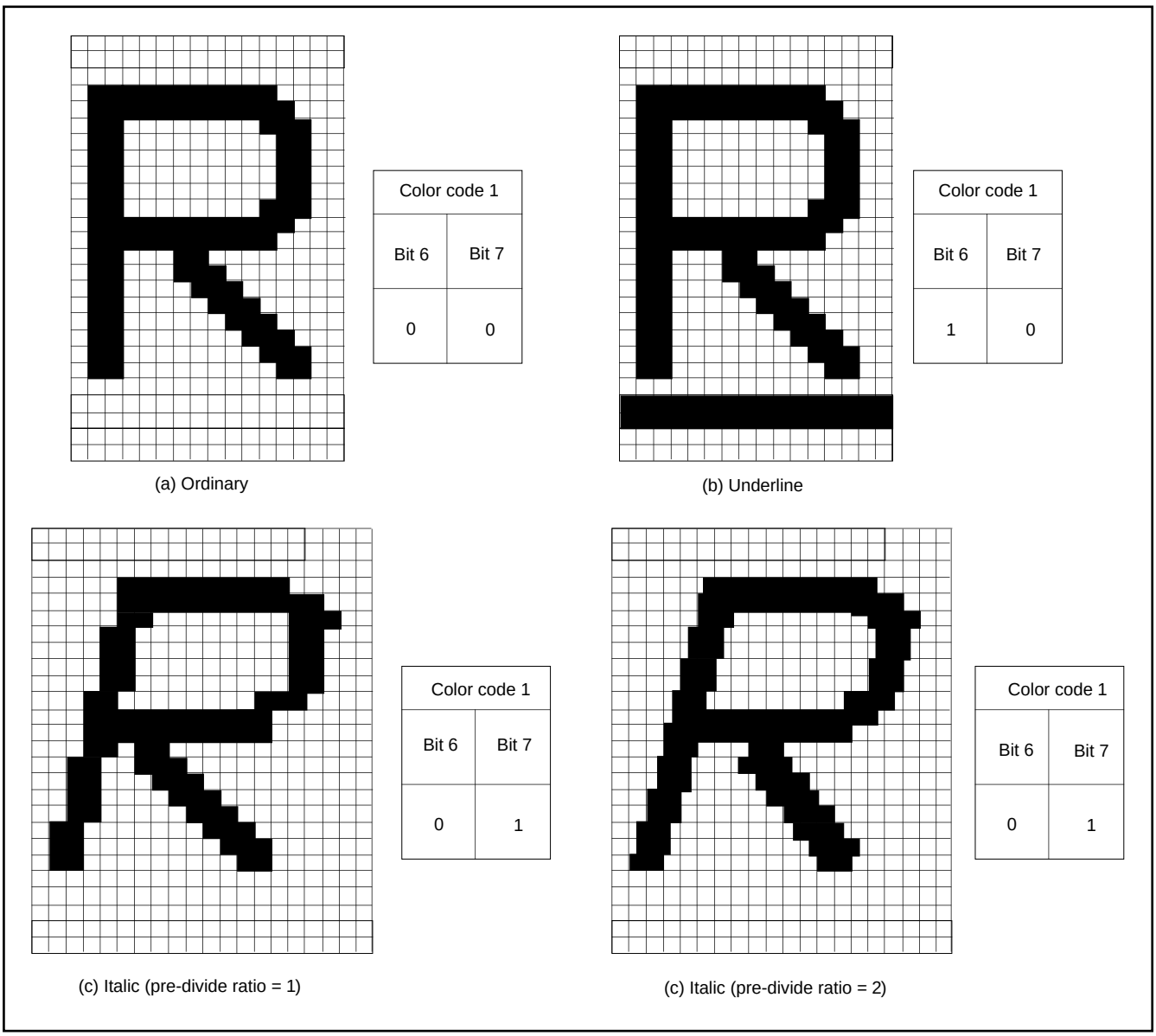


Fig. 68. Example of attribute display (in CC mode)

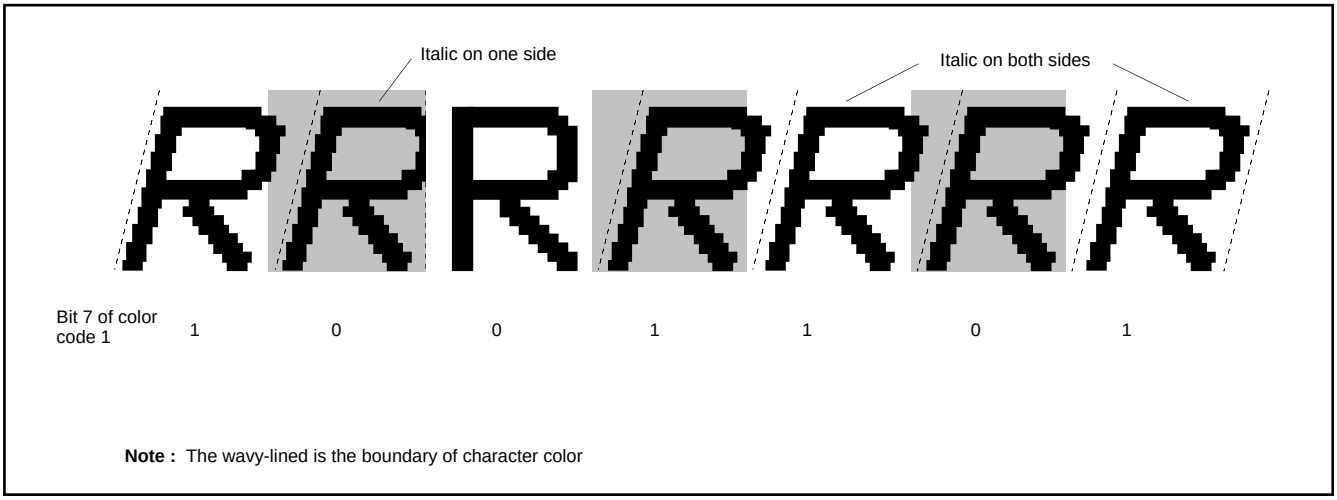


Fig. 69. Example of italic display

⑤ **Border**

The border is output in the OSD mode and the EXOSD 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 70) by bit 2 of the OSD control register (refer to Figure 70). The border ON/OFF is controlled by bit 2 of the block control register (refer to Figure 50).

The OUT1 signal is used for border output. The border color for each screen is specified by the border color register.

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

Notes 1 : There is no border for the extra font.

2 : The border dot area is the shaded area as shown in Figure 72. In the EXOSD mode, top and bottom of character font display area is not bordered.

3 : When the border dot overlaps on the next character font, the character font has priority (refer to Figure 73 A). When the border dot overlaps on the next character back ground, the border has priority (refer to Figure 73 B).

4 : The border is not displayed at right side of the most right dot in the display area of the 40th character (the character located at the most right of the block).

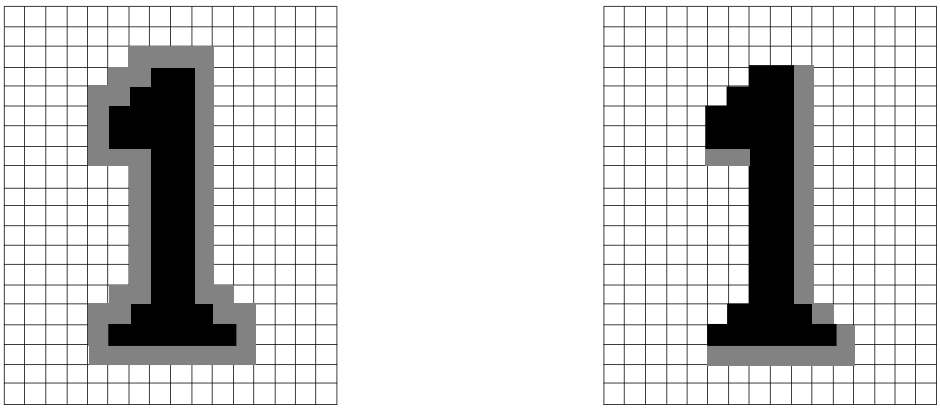
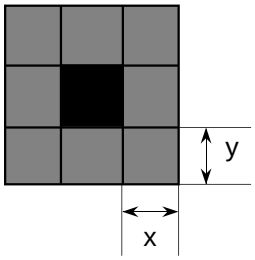


Fig. 70. Example of border display



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

Fig. 71. Horizontal and vertical size of border

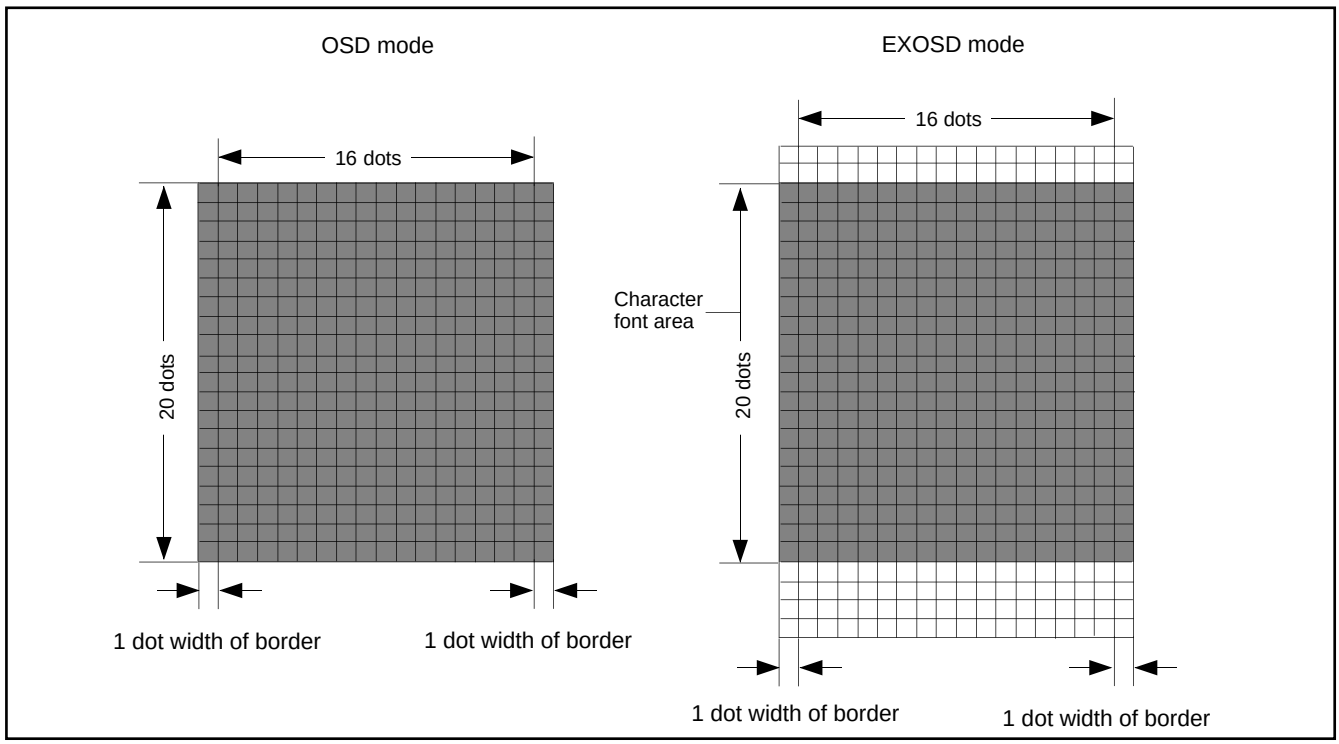


Fig. 72. Border area

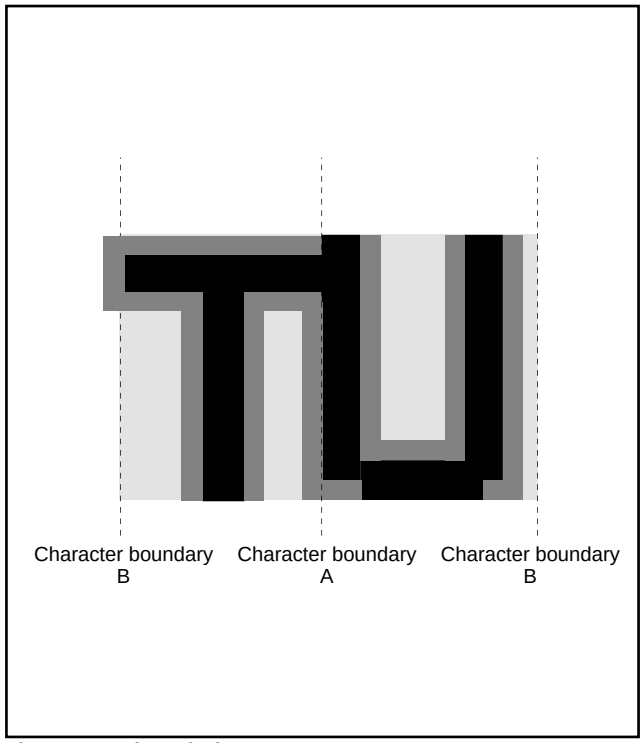


Fig. 73. Border priority

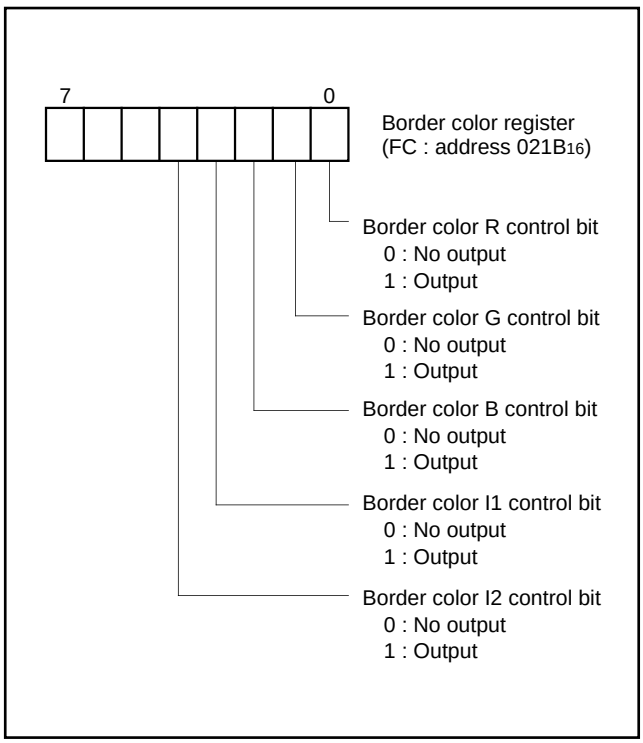


Fig. 74. Structure of border color register

(11) Multiline Display

The M37270MF-XXXSP 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 OSD interrupts.

An OSD 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 OSD interrupt occurs is different depending on the setting of the raster color register (refer to Figure 81).

- When bit 7 of the raster color register is “0”
  An OSD interrupt occurs at the end of block display in the OSD and the EXOSD mode.
- When bit 7 of the raster color register is “1”
  An OSD interrupt occurs at the end of block display in the CC mode.

Notes 1: An OSD 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 (addresses $00D0_{16}$ to $00DF_{16}$), an OSD interrupt request does not occur (refer to Figure 75 (A)).

2: When another block display appeares while one block is displayed, an OSD interrupt request occurs only once at the end of the another block display (refer to Figure 75 (B)).

3: On the screen setting window, an OSD interrupt occurs even at the end of the CC mode block (off display) out of window (refer to Figure 75 (C)).

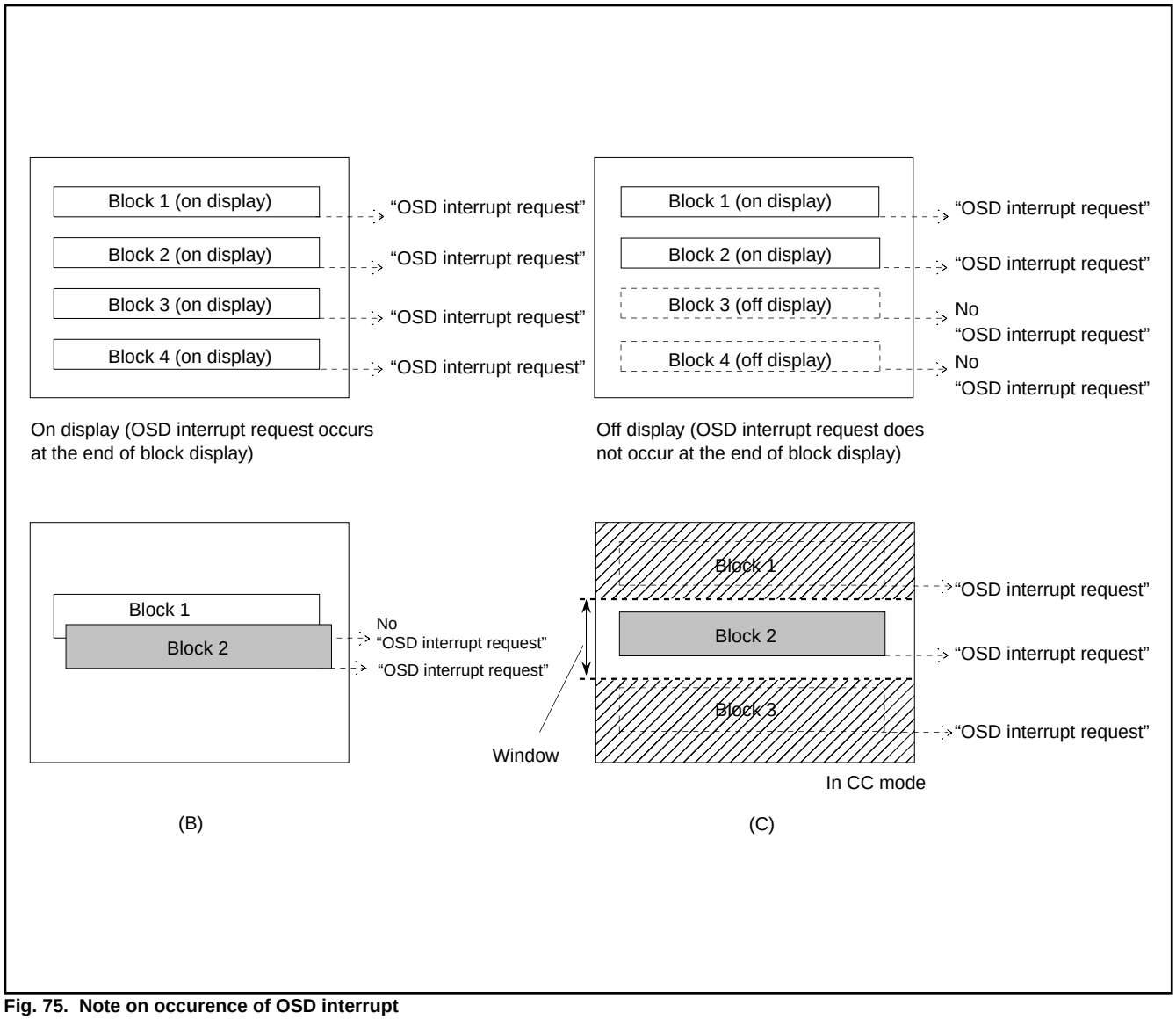


Fig. 75. Note on occurence of OSD interrupt

(12) Automatic Solid Space Function

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

The solid space is output in the following area :

- the character area except character code "009_{16} "
- the character area on the left and right sides of the character area except character code "009_{16} "

This function is turned on and off by bit 4 of the OSD control register (refer to Figure 49).

Notes 1 : Blank is disabled on the left side of the 1st character and on the right side of the 40th character of each block.

2 : When using this function, set "009_{16}" to the 40th character.

Table 18. Setting for automatic solid space

Bit 4 of OSD control register	0				1			
Bit 7 of block control register	0		1		0		1	
Bit 4 of color code 1	0	1	0	1	0	1	0	1
OUT1 output signal	Character font part	Character display area	Character font part		Solid space		Character font part	
OUT2 output signal	OFF		OFF	Character display area	OFF		Solid space	

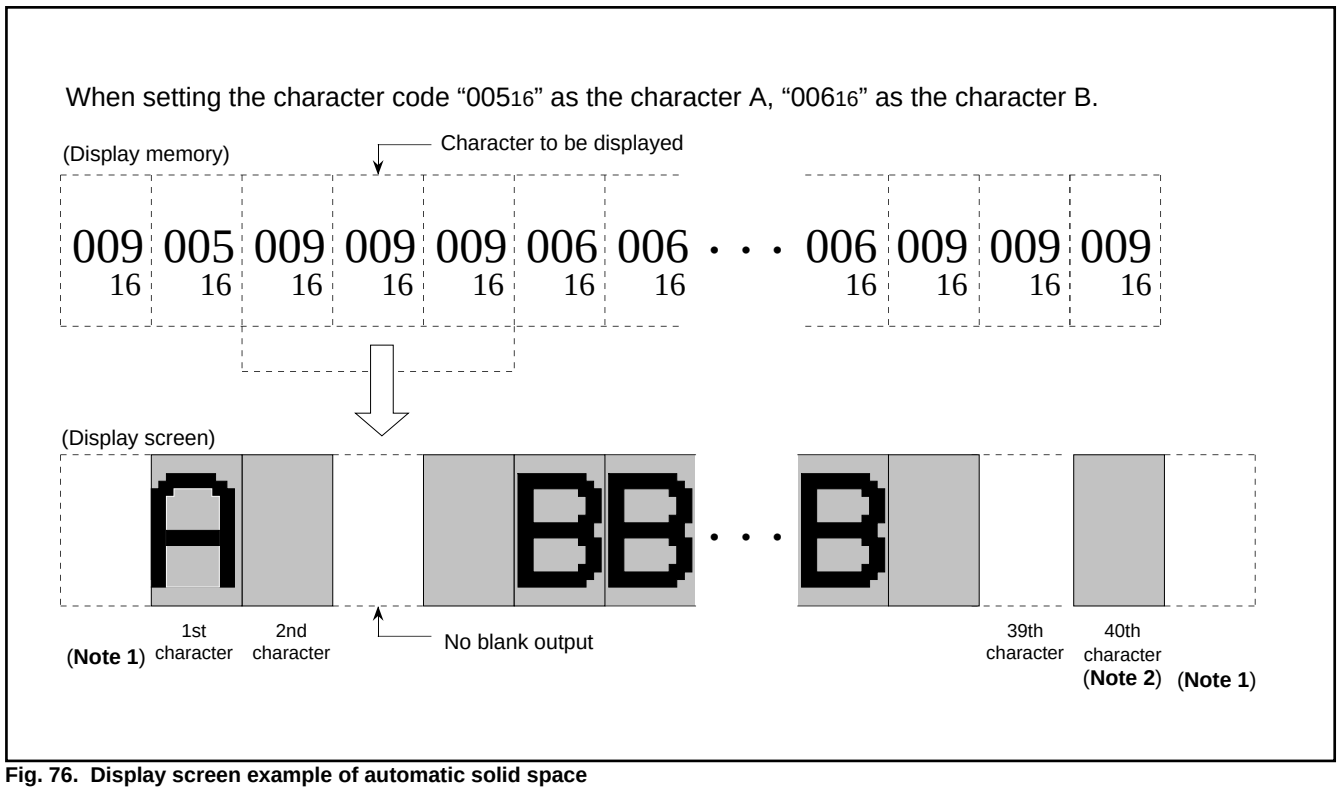


Fig. 76. Display screen example of automatic solid space

(13) Scan Mode

M37270MF-XXXSP 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 (refer to Figure 49).

Table 19. Setting for scan mode

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

(14) Window Function

This function sets the top and bottom boundary of display limit on a screen. The window function is valid only in the CC mode. The top boundary is set by the window H registers 1 and 2. The bottom boundary is set by the window L registers 1 and 2. This function is turned on and off by bit 5 of the OSD control register (refer to Figure 49). The structure of the window H registers 1 and 2 is shown in Figure 78, the structure of the window L registers 1 and 2 is shown in Figure 79.

Notes 1: Set values except "00_{16}" and "01_{16}" to the window H register 1 when the window H register 2 is "00_{16}."
2: Set the register value fit for the following condition :
(WH1 + WH2) < (WL1 + WL2)

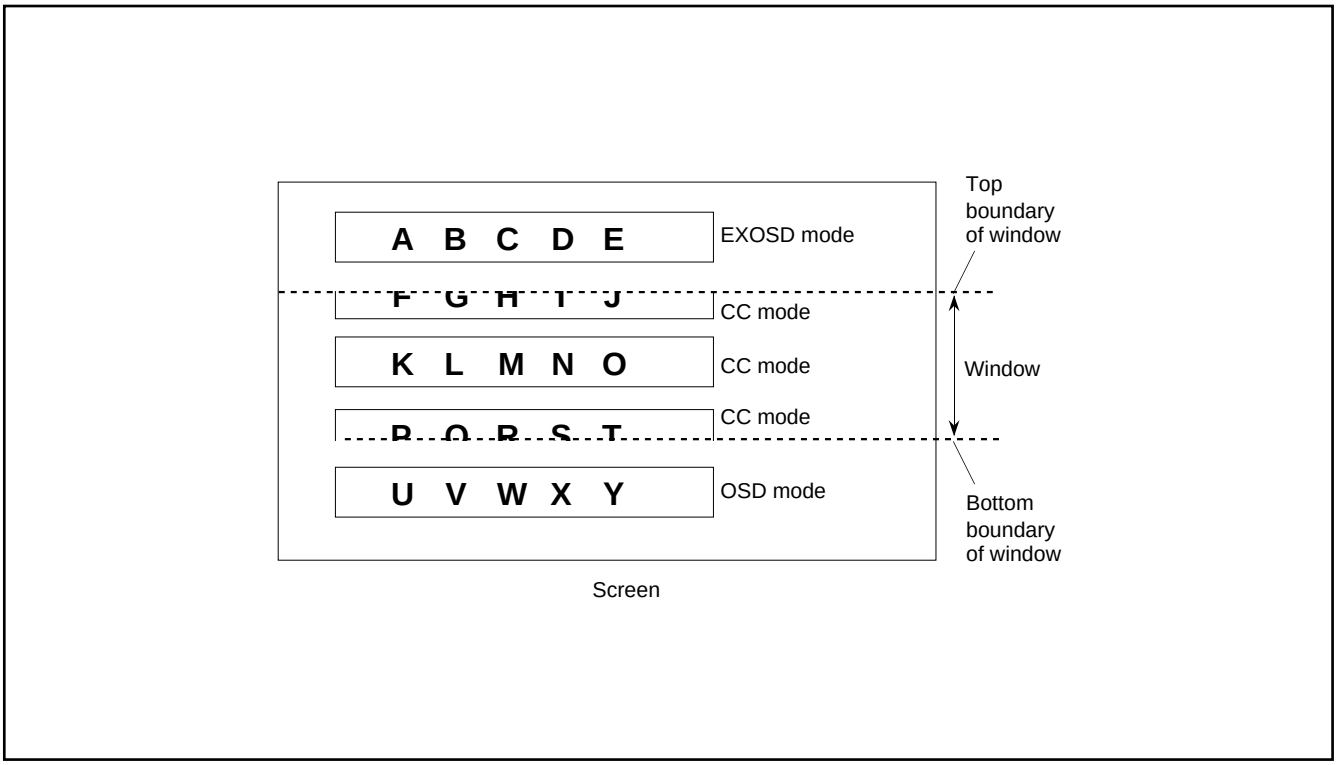
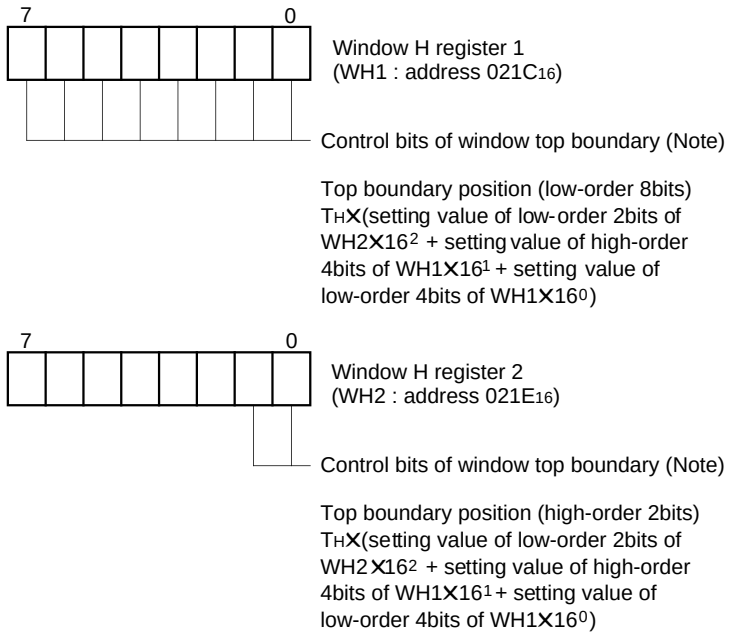
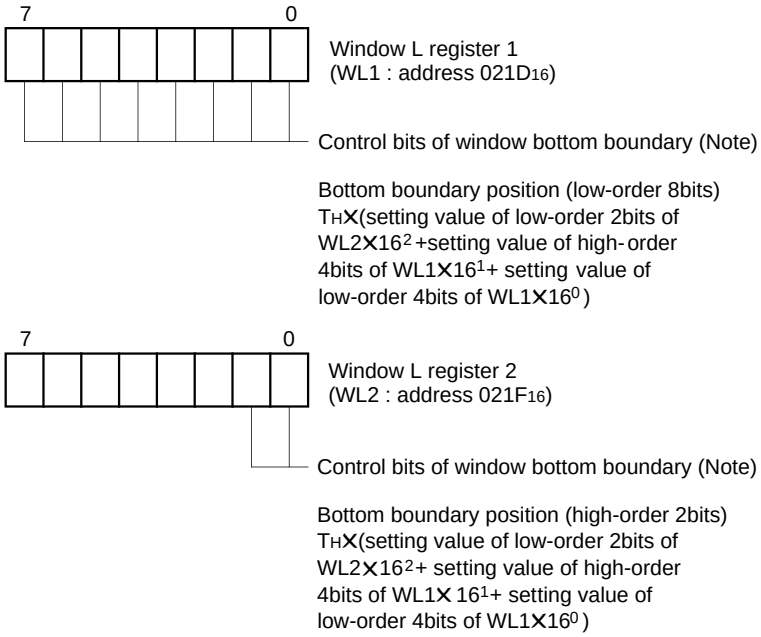


Fig. 77. Example of window function



Note : Set values except "00_{16}" and "01_{16}" to the WH1 when the WH2 is "00_{16}."

Fig. 78. Structure of window H registers



Note : Set values fit for the following condition : (WH1+WH2) < (WL1+WL2).

Fig. 79. Structure of window L registers

(15) OSD Output Pin Control

The OSD output pins R, G, B, and OUT1 can also function as ports $P5_2$, $P5_3$, $P5_4$ and $P5_5$. Set the corresponding bit of the OSD port control register (address $00CB_{16}$) to "0" to specify these pins as OSD output pins, or set it to "1" to specify it as a general-purpose port P5 pins. The OUT2, I1, and I2 can also function as port $P1_0$, $P1_5$, $P1_6$. Set the corresponding bit of the port P1 direction register (address $00C3_{16}$) to "1" (output mode). After that, switch between the OSD output function and the port function by the OSD port control register. Set the corresponding bit to "1" to specify the pin as OSD output pin, or set it to "0" to specify as port P1 pin.

The input polarity of the HSYNC, VSYNC and output polarity of signals R, G, B, I1, I2, OUT1 and OUT2 can be specified with the I/O polarity control register (address 0217_{16}) . Set a bit to "0" to specify positive polarity; set it to "1" to specify negative polarity (refer to Figure 61).

The structure of the OSD port control register is shown in Figure 80.

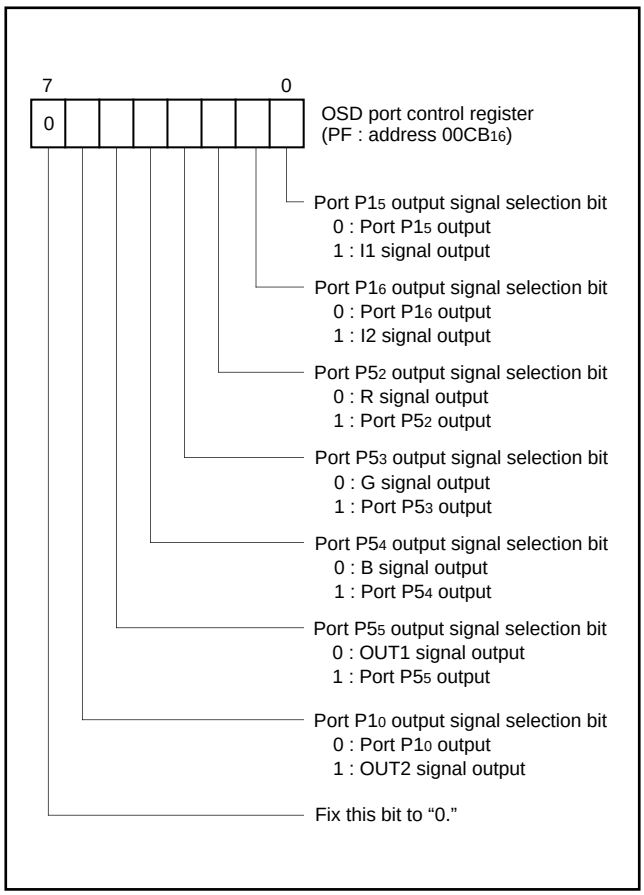


Fig. 80. Structure of OSD port control register

(16) Raster Coloring Function

An entire screen (raster) can be colored by setting the bits 6 to 0 of the raster color register. Since each of the R, G, B, I1, I2, OUT1, and OUT2 pins can be switched to raster coloring output, 7 raster colors can be obtained.

If the OUT1 pin has been set to raster coloring output, a raster coloring signal is always output during 1 horizontal scanning period. This setting is necessary for erasing a background TV image.

If the R, G, B, I1, and I2 pins have been set to output, a raster coloring signal is output in the part except a no-raster colored character (in Figure 82, a character “1”) during 1 horizontal scanning period. This ensures that character colors are not mixed with the raster color.

The structure of the raster color register is shown in Figure 81, the example of raster coloring is shown in Figure 82.

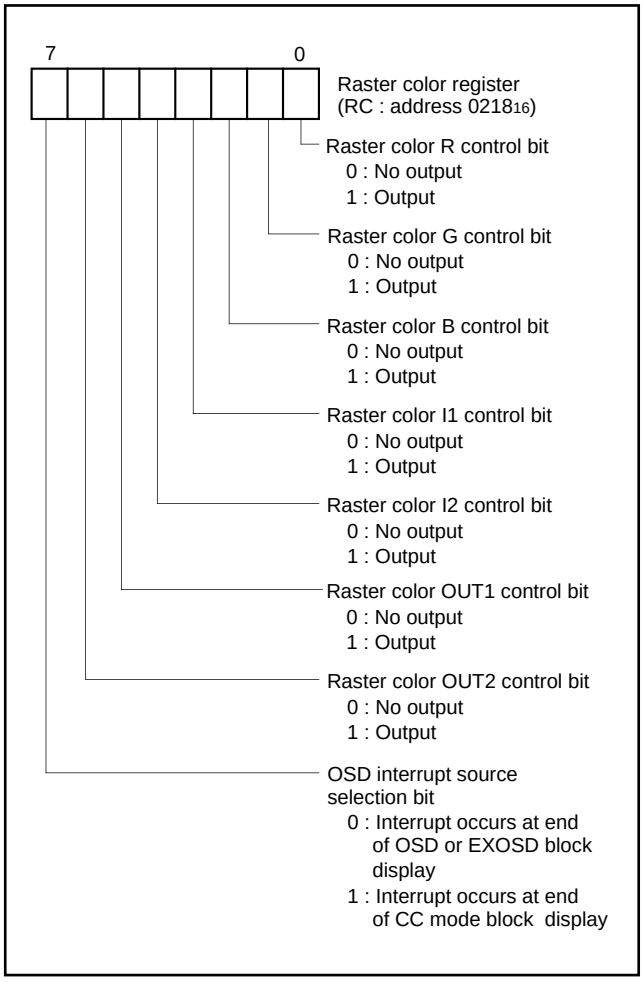


Fig. 81. Structure of raster color register

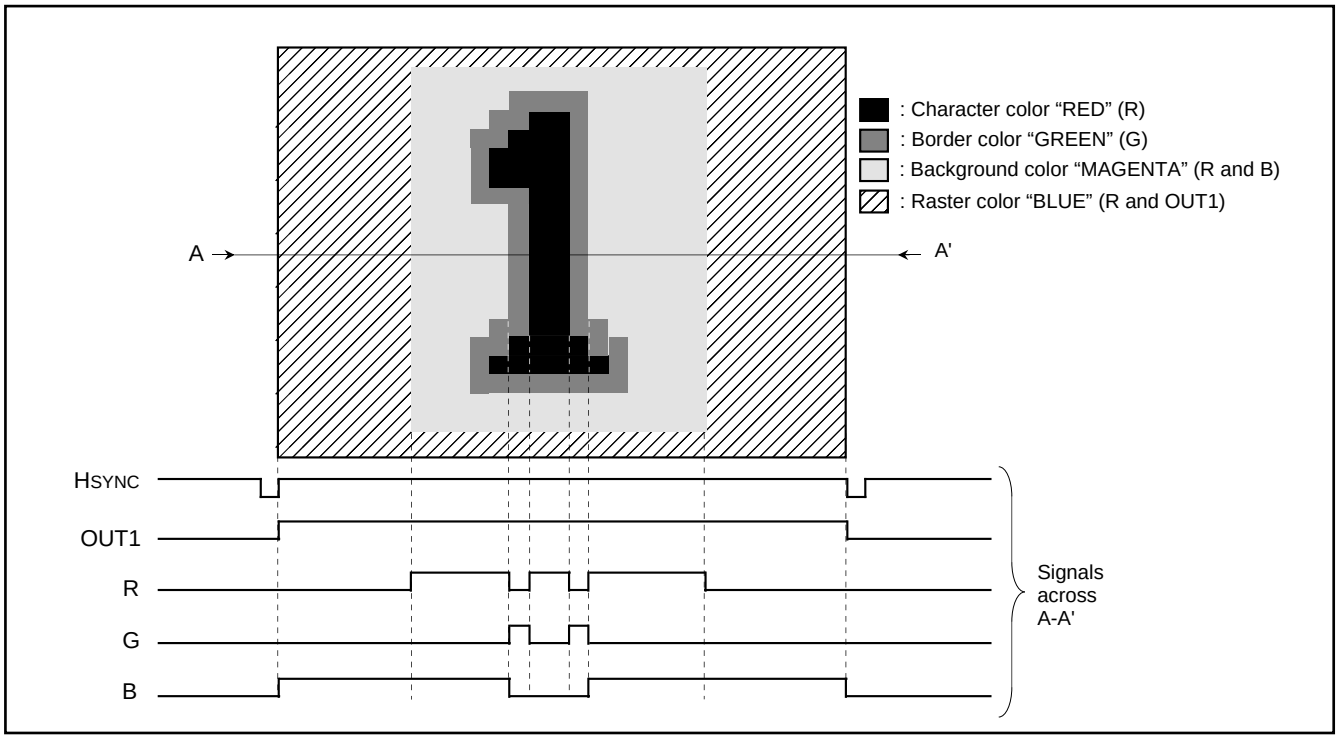


Fig. 82. Example of raster coloring

INTERRUPT INTERVAL DETERMINATION FUNCTION

The M37270MF-XXXSP incorporates an interrupt interval determination circuit. This interrupt interval determination circuit has an 8-bit binary up counter as shown in Figure 83. Using this counter, it determines an interval or a pulse width on the INT1 or INT2 (refer to Figure 85).

The following describes how the interrupt interval is determined.

1. The determination mode is selected by using bit 5 of the interrupt interval determination control register (address 0212_{16}). When this bit is set to "0," the interrupt interval determination mode is selected; when the bit is set to "1," the pulse width determination mode is selected.
2. The interrupt input to be determined (INT1 input or INT2 input) is selected by using bit 2 in the interrupt interval determination control register (address 0212_{16}). When this bit is cleared to "0," the INT1 input is selected ; when the bit is set to "1," the INT2 input is selected.
3. When the INT1 input is to be determined, the polarity is selected by using bit 3 of the interrupt interval determination control register ; when the INT2 input is to be determined, the polarity is selected by using bit 4 of the interrupt interval determination control register.
   When the relevant bit is cleared to "0," determination is made of the interval of a positive polarity (rising transition) ; when the bit is set to "1," determination is made of the interval of a negative polarity (falling transition).
4. The reference clock is selected by using bit 1 of the interrupt interval determination control register. When the bit is cleared to "0," a 32μs clock is selected ; when the bit is set to "1," a 16μs clock is selected (based on an oscillation frequency of 8MHz in either case).
5. Simultaneously when the input pulse of the specified polarity (rising or falling transition) occurs on the INT1 pin (or INT2 pin), the 8-bit binary up counter starts counting up with the selected reference clock (32µs or 16µs).
6. Simultaneously with the next input pulse, the value of the 8-bit binary up counter is loaded into the interrupt interval determination register (address 0211_{16}) and the counter is immediately reset ("00_{16}"). The reference clock is input in succession even after the counter is reset, and the counter restarts counting up from "00_{16}".
7. When count value "FE_{16}" is reached, the 8-bit binary up counter stops counting. Then, simultaneously when the next reference clock is input, the counter sets value "FF_{16}" to the interrupt interval determination register. The reference clock is generated by setting bit 0 of the PWM mode register 1 to "0."

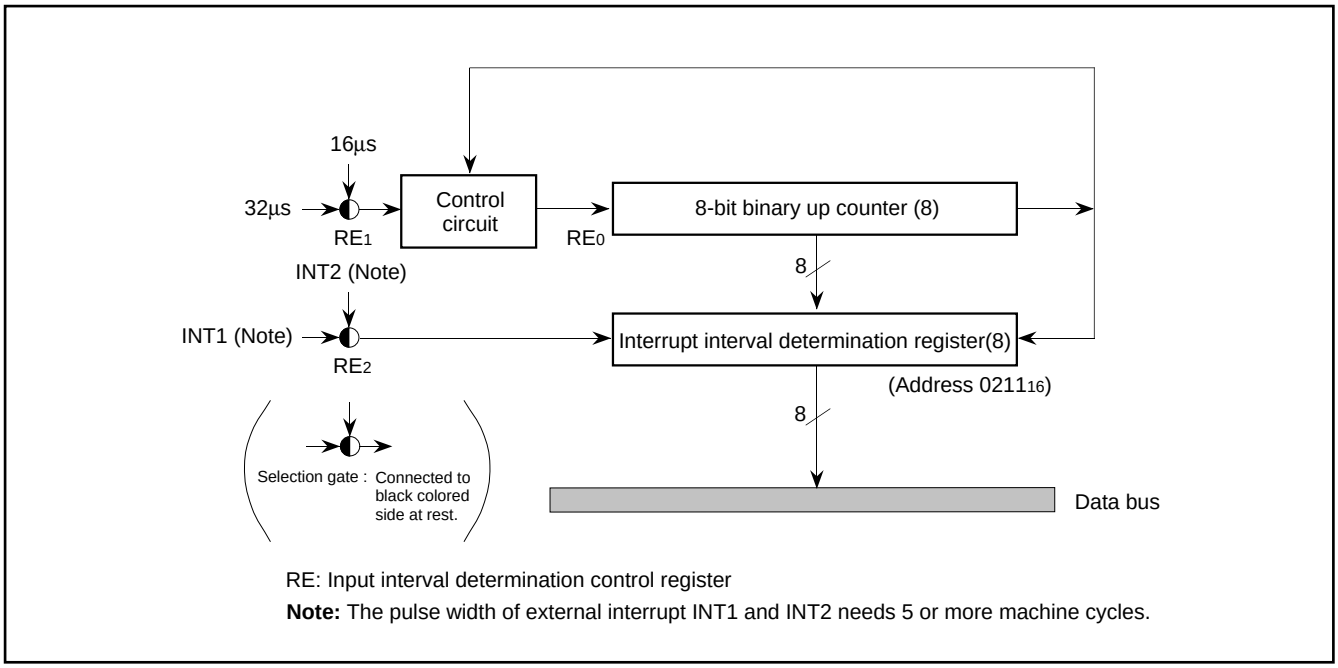


Fig. 83. Block diagram of interrupt interval determination circuit

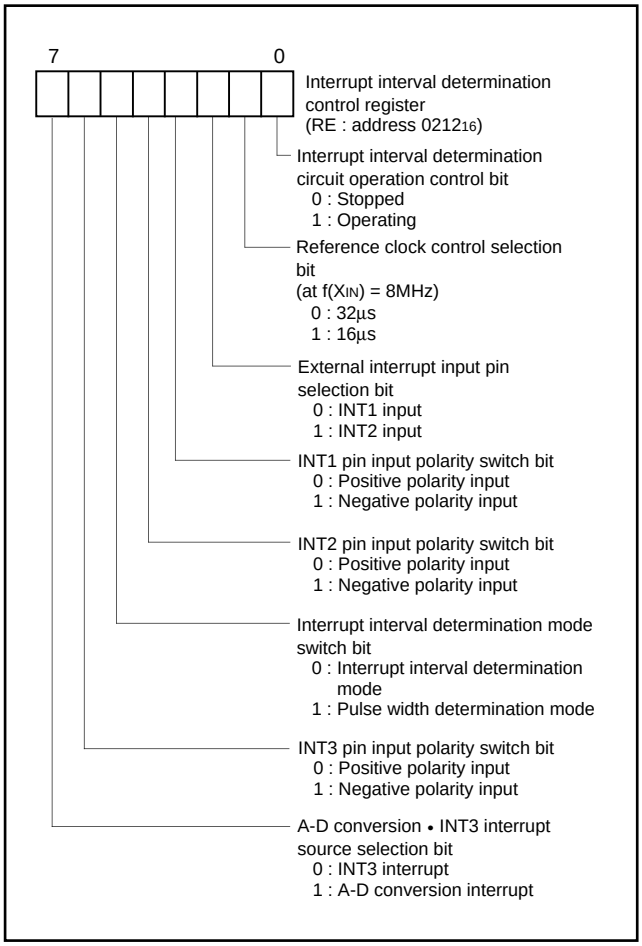


Fig. 84. Structure of interrupt interval determination control register

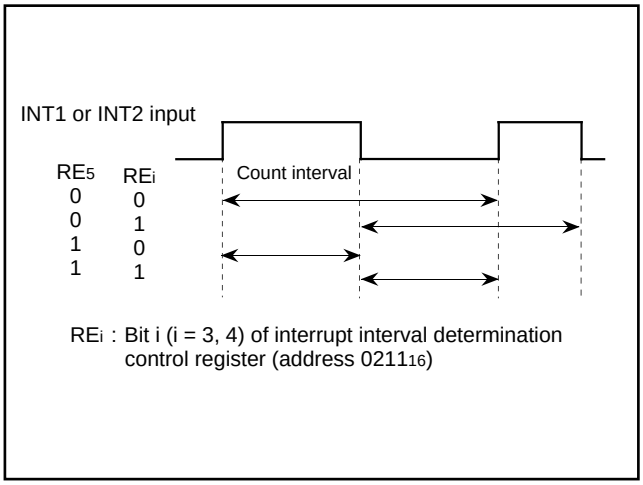


Fig. 85. Setting value of interrpt interval determination control register and measuring interval

RESET CIRCUIT

The M37270MF-XXXSP is reset according to the sequence shown in Figure 87. It starts the program from the address formed by using the content of address FFFF16 as the high-order address and the content of the address FFFE16 as the low-order address, when the $\overline{\text{RESET}}$ pin is held at "L" level for 2 μs or more while the power source voltage is 5 V ± 10 % and the oscillation of a quartz-crystal oscillator or a ceramic resonator is stable and then returned to "H" level. The internal state of microcomputer at reset are shown in Figure 88.

An example of the reset circuit is shown in Figure 86.

The reset input voltage must be kept 0.9 V or less until the power source voltage surpasses 4.5 V.

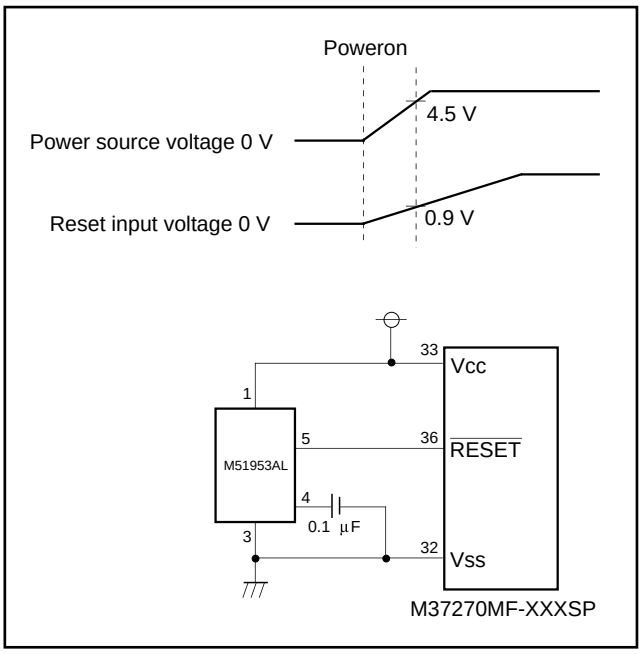
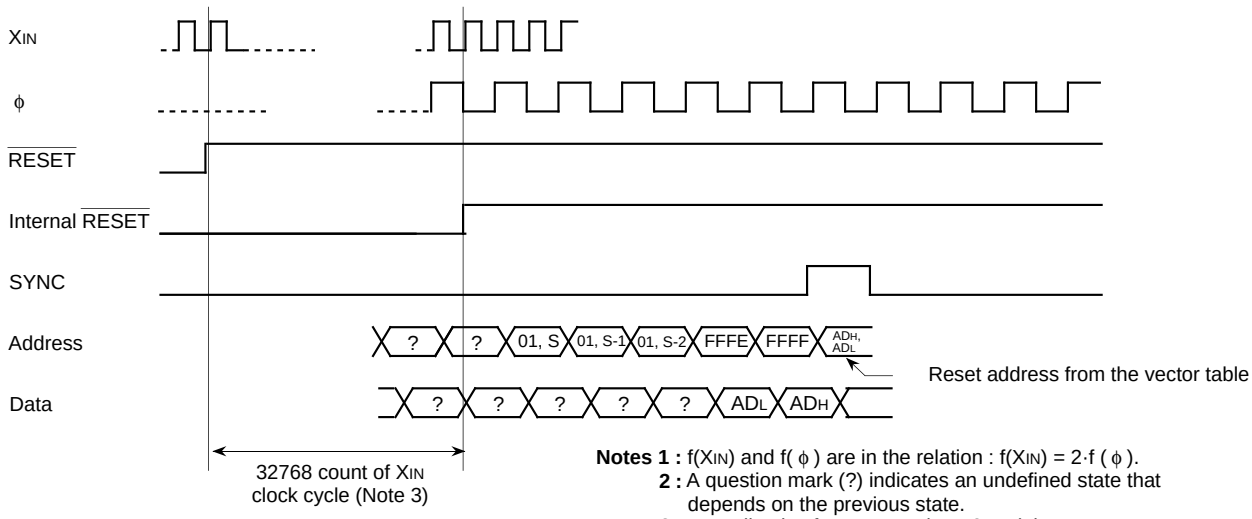


Fig. 86. Example of reset circuit



Notes 1 : f(XIN) and f(ϕ) are in the relation : f(XIN) = 2·f (ϕ).
2 : A question mark (?) indicates an undefined state that depends on the previous state.
3 : Immediately after a reset, timer 3 and timer 4 are connected in hardware. At this time, "FF16" is set in timer 3 and "0716" is set to timer 4. Timer 3 counts down with f(XIN)/16, and reset state is released by the timer 4 overflow signal.

Fig. 87. Reset sequence

	Address	Contents of register
Port P0 direction register	(00C1$_{16}$)	00$_{16}$
Port P1 direction register	(00C3$_{16}$)	00$_{16}$
Port P2 direction register	(00C5$_{16}$)	00$_{16}$
Port P3 direction register	(00C7$_{16}$)	0 0 ⊠ ⊠ ⊠ 0 0
Port P4 direction register	(00C9$_{16}$)	⊠ 0 0 ⊠ ⊠ ⊠ 0
OSD port control register	(00CB$_{16}$)	00$_{16}$
OSD control register	(00CE$_{16}$)	00$_{16}$
Horizontal register	(00CF$_{16}$)	00$_{16}$
Caption position register	(00E0$_{16}$)	00$_{16}$
Start bit position register	(00E1$_{16}$)	00$_{16}$
Window register	(00E2$_{16}$)	00$_{16}$
Sync slice register	(00E3$_{16}$)	00$_{16}$
Data register 1	(00E4$_{16}$)	00$_{16}$
Data register 2	(00E5$_{16}$)	00$_{16}$
Clock run-in register 1	(00E6$_{16}$)	00$_{16}$
Clock run-in register 2	(00E7$_{16}$)	00$_{16}$
Clock run-in detect register 1	(00E8$_{16}$)	00$_{16}$
Clock run-in detect register 2	(00E9$_{16}$)	0 0 0 0 1 0 0 1
Data slicer control register 1	(00EA$_{16}$)	00$_{16}$
Data slicer control register 2	(00EB$_{16}$)	✻ 0 ✻ 0 0 ✻ 0 0
Data register 3	(00EC$_{16}$)	00$_{16}$
Data register 4	(00ED$_{16}$)	00$_{16}$
A-D control register	(00EF$_{16}$)	0 ⊠ ⊠ 0 1 ⊠ 0
Timer 1	(00F0$_{16}$)	FF$_{16}$
Timer 2	(00F1$_{16}$)	07$_{16}$
Timer 3	(00F2$_{16}$)	FF$_{16}$
Timer 4	(00F3$_{16}$)	07$_{16}$
Timer mode register 1	(00F4$_{16}$)	00$_{16}$
Timer mode register 2	(00F5$_{16}$)	00$_{16}$

	Address	Contents of register
I^{2}C address register	(00F7$_{16}$)	00$_{16}$
I^{2}C status register	(00F8$_{16}$)	0 0 0 1 0 0 0 ✻
I^{2}C control register	(00F9$_{16}$)	00$_{16}$
I^{2}C clock control register	(00FA$_{16}$)	00$_{16}$
CPU mode register	(00FB$_{16}$)	0 0 1 1 1 1 0 0
Interrupt request register 1	(00FC$_{16}$)	⊠ 0 0 0 0 0 0 0
Interrupt request register 2	(00FD$_{16}$)	00$_{16}$
Interrupt control register 1	(00FE$_{16}$)	⊠ 0 0 0 0 0 0 0
Interrupt control register 2	(00FF$_{16}$)	00$_{16}$
Clock run-in detect register 3	(0208$_{16}$)	⊠ 0 0 0 0 0 0 0
PWM mode register 1	(020A$_{16}$)	⊠ ⊠ ⊠ ⊠ 0 ⊠ ⊠ 0
PWM mode register 2	(020B$_{16}$)	00$_{16}$
Timer 5	(020C$_{16}$)	07$_{16}$
Timer 6	(020D$_{16}$)	FF$_{16}$
Sync pulse counter register	(020F$_{16}$)	⊠ ⊠ 0 0 0 0 0 0
Data slicer control register 3	(0210$_{16}$)	00$_{16}$
Interrupt interval determination control register	(0212$_{16}$)	00$_{16}$
Serial I/O mode register	(0213$_{16}$)	00$_{16}$
Clock source control register	(0216$_{16}$)	00$_{16}$
I/O polarity control register	(0217$_{16}$)	✻ 0 0 0 0 0 0 0
Raster color register	(0218$_{16}$)	00$_{16}$
Extra font color register	(0219$_{16}$)	⊠ ⊠ ⊠ 0 0 0 0 0
Border color register	(021B$_{16}$)	⊠ ⊠ ⊠ 0 0 0 0 0
Processor status register	(PS)	✻ ✻ ✻ ✻ ✻ ✻ 1 ✻ ✻
Program counter	(PC$_H$)	Contents of address FFFF$_{16}$
	(PC$_L$)	Contents of address FFFE$_{16}$

Note : The contents of all other registers and RAM are undefined at reset, so set their initial values.

- ✻ Undefined
- ⊠ Unused bit

Fig. 88. Internal state of microcomputer at reset

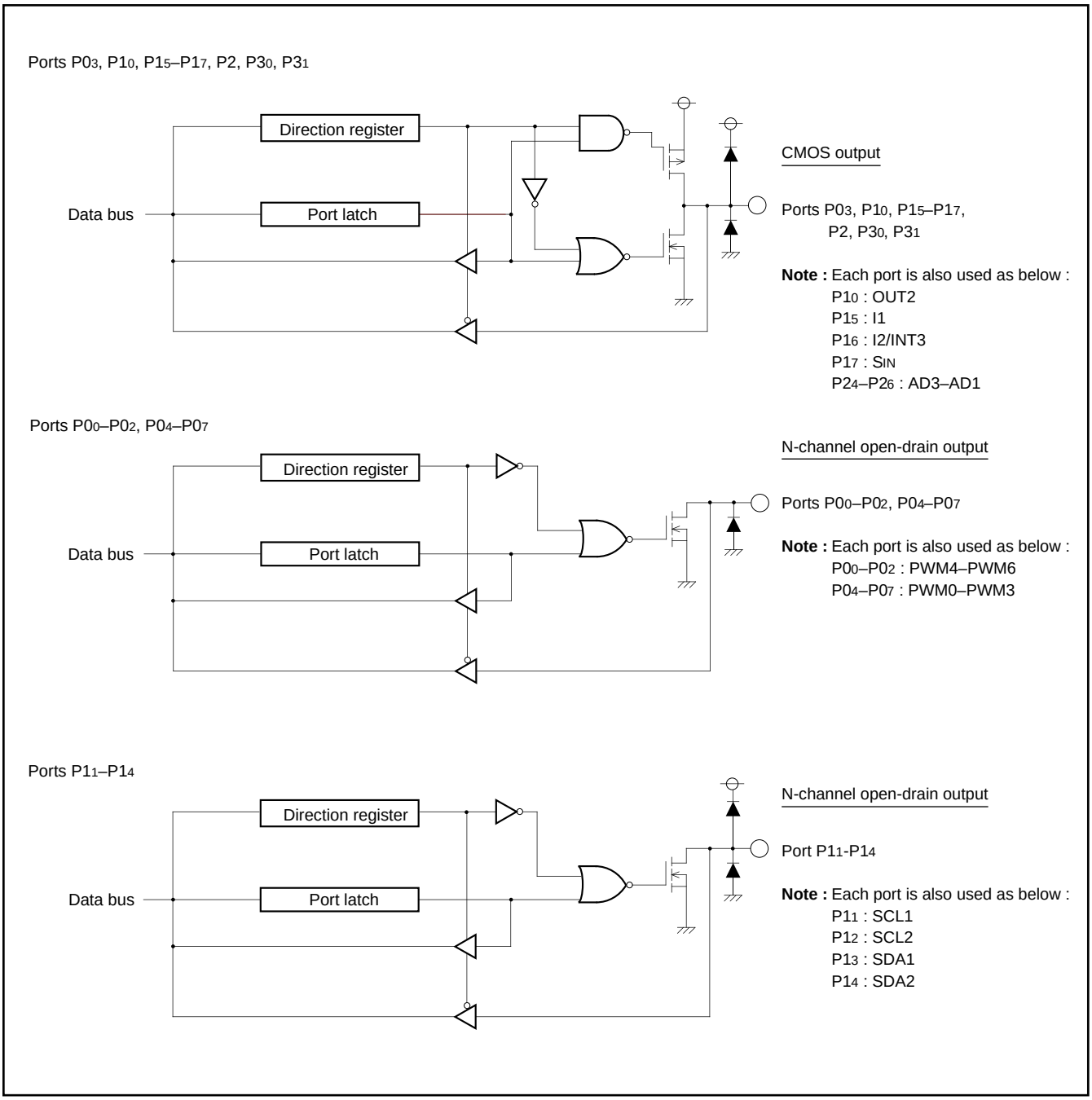


Fig. 89. I/O pin block diagram (1)

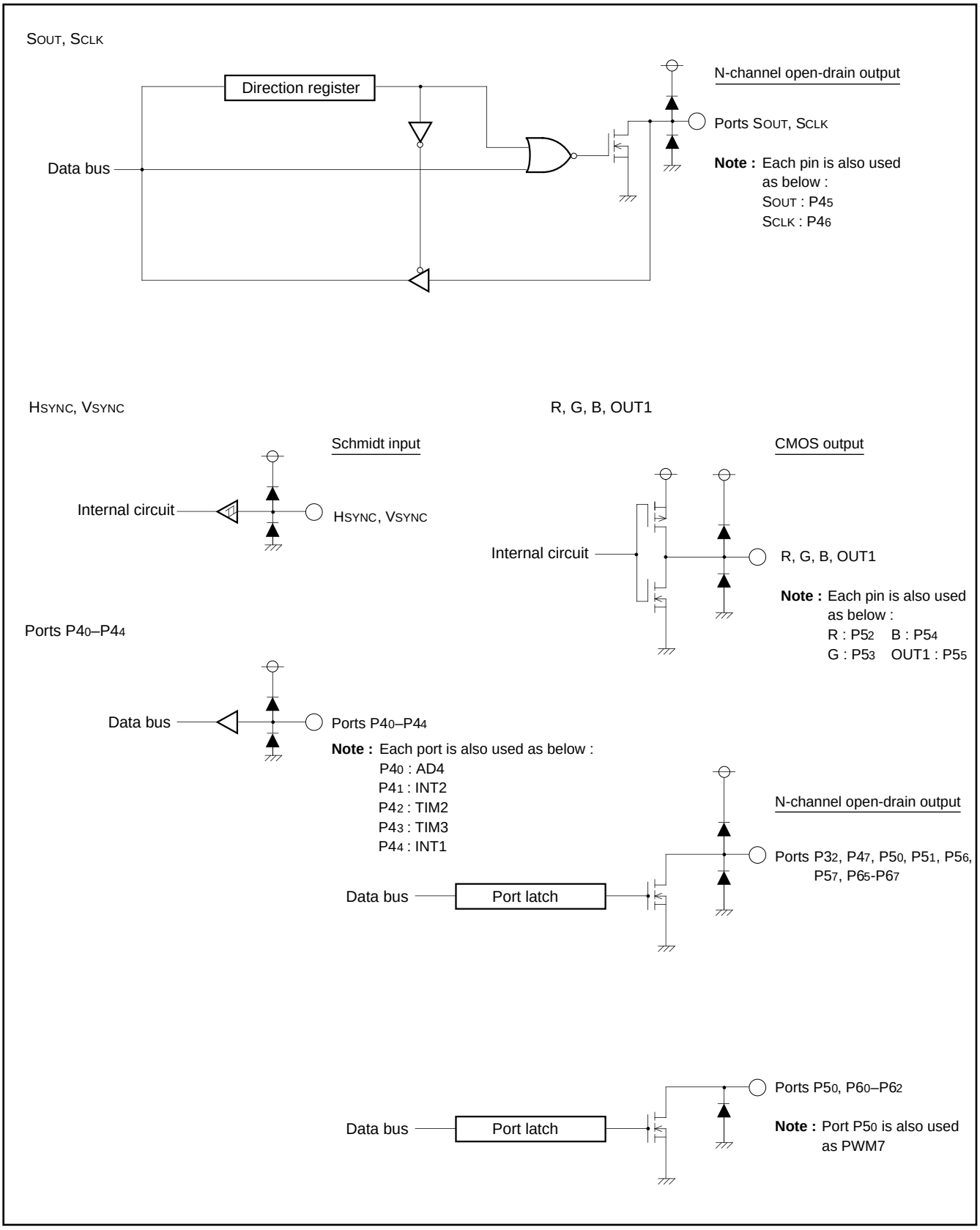


Fig. 90. I/O pin block diagram (2)

CLOCK GENERATING CIRCUIT

The M37270MF-XXXSP has 2 built-in oscillation circuits. An oscillation circuit can be formed by connecting a resonator between XIN and XOUT (XCIN and XCOUT). Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between XIN and XOUT since a feed-back resistor exists on-chip. However, an external feed-back resistor is needed between XCIN and XCOUT. When using XCIN-XCOUT as sub-clock, clear bits 5 and 4 of the clock source control register to "0." To supply a clock signal externally, input it to the XIN (XCIN) pin and make the XOUT (XCOUT) pin open. When not using XCIN clock, connect the XCIN to VSS and make the XCOUT pin open.

After reset has completed, the internal clock ϕ is half the frequency of XIN. Immediately after poweron, both the XIN and XCIN clock start oscillating. To set the internal clock ϕ to low-speed operation mode, set bit 7 of the CPU mode register (address $00FB_{16}$) to "1."

Oscillation Control

(1) Stop mode

The built-in clock generating circuit is shown in Figure 56. When the STP instruction is executed, the internal clock ϕ stops at "H" level. At the same time, timers 3 and 4 are connected in hardware and "FF_{16}" is set in the timer 3, "07_{16}" is set in the timer 4. Select f(XIN)/16 or f(XCIN)/16 as the timer 3 count source (set both bit 0 of the timer mode register 2 and bit 6 at address $00C7_{16}$ to "0" before the execution of the STP instruction). And besides, set the timer 3 and timer 4 interrupt enable bits to disabled ("0") before execution of the STP instruction. The oscillator restarts when external interrupt is accepted, however, the internal clock ϕ keeps its "H" level until timer 4 overflows. Because this allows time for oscillation stabilizing when a ceramic resonator or a quartz-crystal oscillator is used.

(2) Wait mode

When the WIT instruction is executed, the internal clock ϕ stops in the "H" level but the oscillator continues running. This wait state is released at reset or when an interrupt is accepted (Note). Since the oscillator does not stop, the next instruction can be executed at once.

Note: In the wait mode, the following interrupts are invalid.

(1) VSYNC interrupt
(2) OSD interrupt
(3) Timers 1 and 2 interrupts using $P4_2$/TIM2 pin input as count source
(4) Timer 3 interrupt using $P4_3$/TIM3 pin input as count source
(5) Data slicer interrupt
(6) Multi-master I^2C-BUS interface interrupt
(7) f(XIN)/4096 interrupt
(8) All timer interrupts using f(XIN)/2 or f(XCIN)/2 as count source
(9) All timer interrupts using f(XIN)/4096 or f(XCIN)/4096 as count source
(10) A-D conversion interrupt

(3) Low-Speed Mode

If the internal clock is generated from the sub-clock (XCIN), a low power consumption operation can be realized by stopping only the main clock XIN. To stop the main clock, set bit 6 (CM_6) of the CPU mode register ($00FB_{16}$) to "1." When the main clock XIN is restarted, the program must allow enough time to for oscillation to stabilize.

Note that in low-power-consumption mode the XCIN-XCOUT drivability can be reduced, allowing even lower power consumption (60 µA with f (XCIN) = 32kHz). To reduce the XCIN-XCOUT drivability, clear bit 5 (CM_5) of the CPU mode register ($00FB_{16}$) to "0." At reset, this bit is set to "1" and strong drivability is selected to help the oscillation to start. When an STP instruction is executed, set this bit to "1" by software before executing.

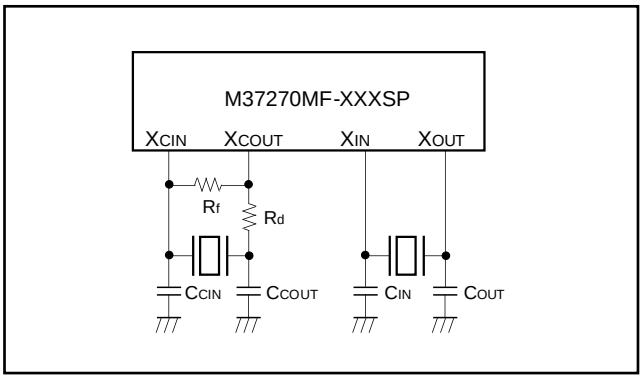


Fig. 91. Ceramic resonator circuit example

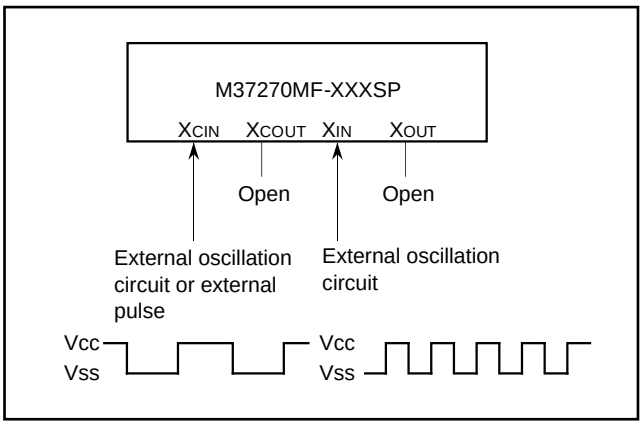
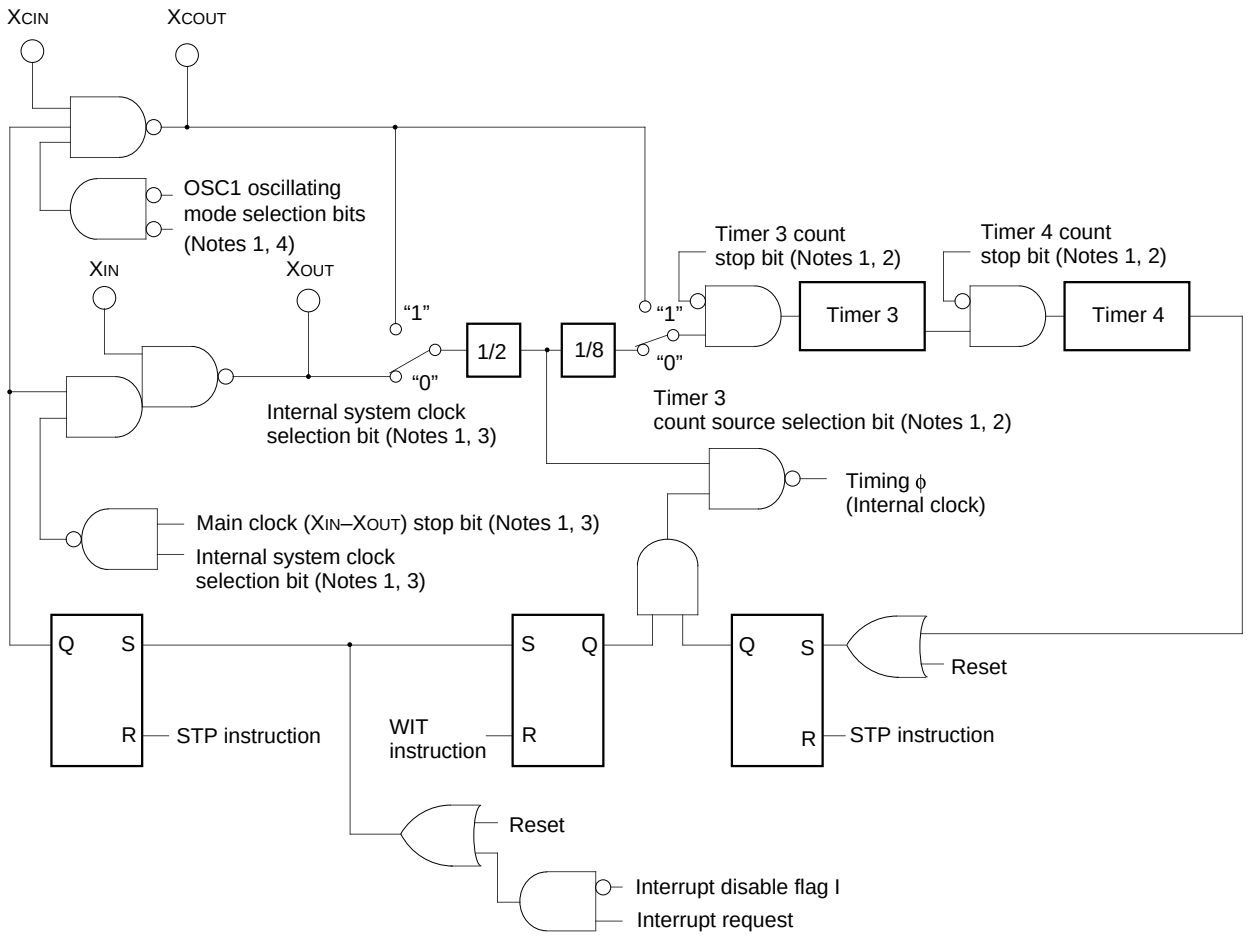
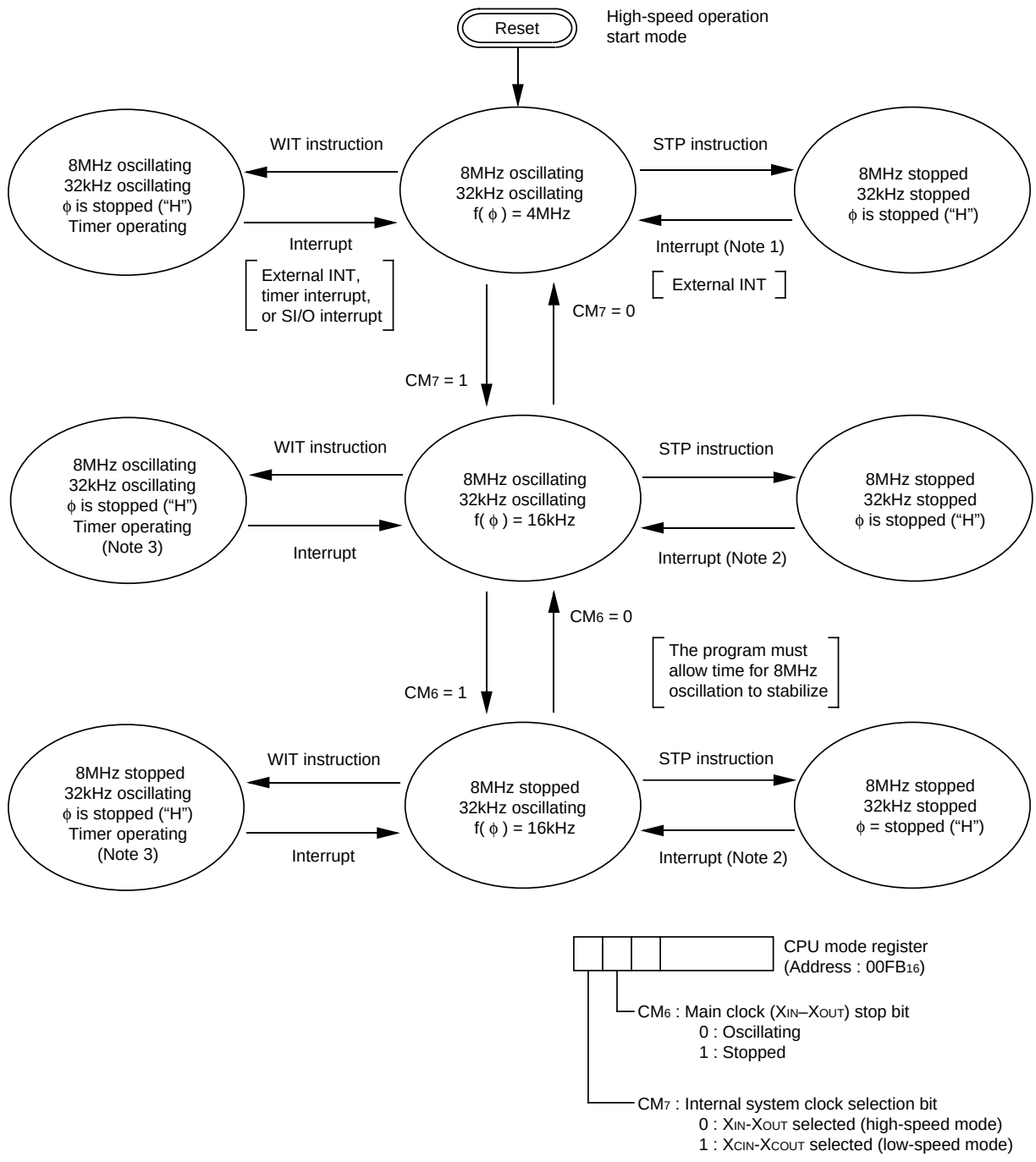


Fig. 92. External clock input circuit example



Notes 1 : The value at reset is "0."
2 : Refer to the structure of timer mode register 2.
3 : Refer to the structure of CPU mode register (next page).
4 : Refer to the structure of clock source control register.

Fig. 93. Clock generating circuit block diagram



The example assumes that 8 MHz is being applied to the X_{IN} pin and 32 kHz to the X_{CIN} pin. The ϕ indicates the internal clock.

Notes 1: When the STP state is ended, a delay of approximately 8ms is automatically generated by timer 3 and timer 4.
2: The delay after the STP state ends is approximately 2s.
3: When the internal clock ϕ divided by 8 is used as the timer count source, the frequency of the count source is 2kHz.

Fig. 94. State transitions of system clock

DISPLAY OSCILLATION CIRCUIT

The OSD oscillation circuit has a built-in clock oscillation circuits, so that a clock for OSD can be obtained simply by connecting an LC, a ceramic resonator, or a quartz-crystal oscillator across the pins OSC1 and OSC2. Which of the sub-clock or the OSD oscillation circuit is selected by setting bits 5 and 4 of the clock source control register (address 0216_{16}).

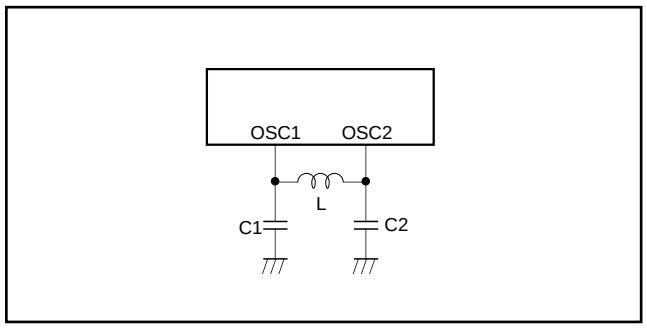
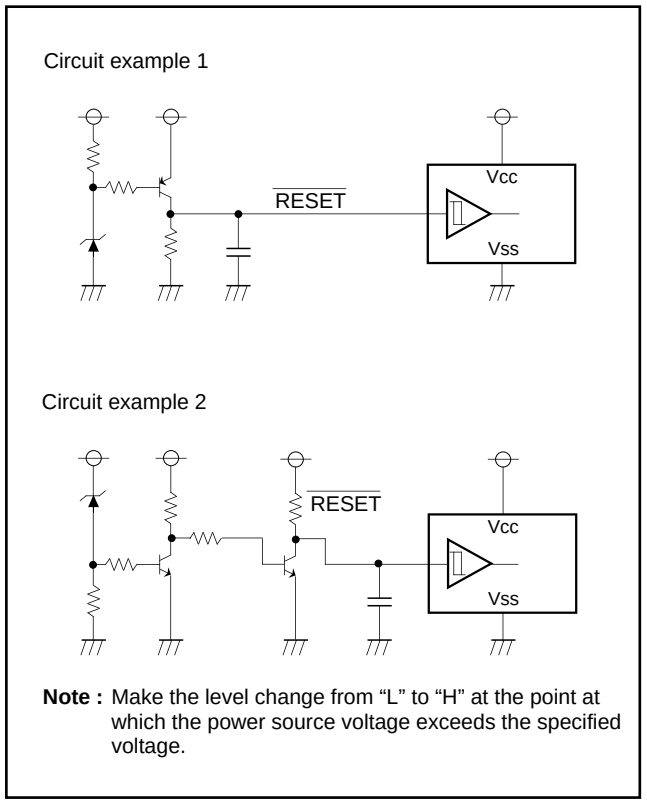


Fig. 95. Display oscillation circuit

AUTO-CLEAR CIRCUIT

When power source is supplied, the auto-clear function can be performed by connecting the following circuit to the $\overline{\text{RESET}}$ pin.



Note : Make the level change from "L" to "H" at the point at which the power source voltage exceeds the specified voltage.

Fig. 96. Auto-clear circuit example

ADDRESSING MODE

The memory access is reinforced with 17 kinds of addressing modes. Refer to the SERIES 740 <Software> User's Manual for details.

MACHINE INSTRUCTIONS

There are 71 machine instructions. Refer to the SERIES 740 <Software> User's Manual for details.

PROGRAMMING NOTES

(1) The divide ratio of the timer is 1/(n+1).

(2) Even though the BBC and BBS instructions are executed immediately after the interrupt request bits are modified (by the program), those instructions are only valid for the contents before the modification. At least one instruction cycle is needed (such as an NOP) between the modification of the interrupt request bits and the execution of the BBC and BBS instructions.

(3) After the ADC and SBC instructions are executed (in decimal mode), one instruction cycle (such as an NOP) is needed before the SEC, CLC, or CLD instruction is executed.

(4) An NOP instruction is needed immediately after the execution of a PLP instruction.

(5) In order to avoid noise and latch-up, connect a bypass capacitor ($\approx 0.1\ \mu F$) directly between the V_{CC} pin–V_{SS} pin, AV_{CC} pin–V_{SS} pin, and the V_{CC} pin–CNV_{SS} pin using a thick wire.

DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

(1) Mask ROM Order Confirmation Form
(2) Mark Specification Form
(3) Data to be written to ROM, in EPROM form (32-pin DIP Type 27C101, three identical copies)

PROM Programming Method

The built-in PROM of the One Time PROM version (blank) and built-in EPROM version can be read or programmed with a general-purpose PROM programmer using a special programming adapter.

Product	Name of Programming Adapter
M37270FSP	PCA7401

The PROM of the One Time PROM version (blank) is not tested or screened in the assembly process and following processes. To ensure proper operation after programming, the procedure shown in Figure 97 is recommended to verify programming.

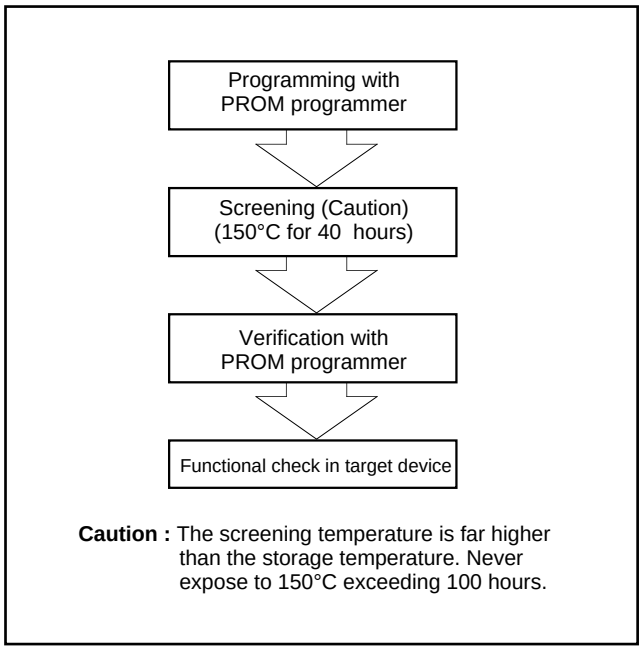


Fig. 97. Programming and testing of One Time PROM version

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter		Conditions	Ratings	Unit
V_{CC}, AV_{CC}	Power source voltage	V_{CC}, AV_{CC}	All voltages are based on V_{SS}. Output transistors are cut off.	–0.3 to 6	V
V_I	Input voltage	CNV_{SS}		–0.3 to 6	V
V_I	Input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P4_0$–$P4_6$, $\overline{P6_4}$, OSC1, X_{IN}, H_{SYNC}, V_{SYNC}, $\overline{RESET}$, CV_{IN}		–0.3 to V_{CC} + 0.3	V
V_O	Output voltage	$P0_3$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_2$, $P4_7$, $P5_1$, $P5_6$, $P5_7$, $P6_0$–$P6_2$, $P6_5$–$P6_7$, R, G, B, OUT1, S_{OUT}, S_{CLK}, X_{OUT}, OSC2		–0.3 to V_{CC} + 0.3	V
V_O	Output voltage	$P0_0$–$P0_2$, $P0_4$–$P0_7$, $P5_0$, $P6_0$–$P6_2$		–0.3 to 13	V
I_{OH}	Circuit current	R, G, B, OUT1, OUT2, $P0_3$, $P1_5$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$		0 to 1 (Note 1)	mA
I_{OL1}	Circuit current	R, G, B, OUT1, OUT2, $P0_3$, $P1_5$–$P1_7$, $P2_0$–$P2_7$, $P5_6$, $P5_7$, $P6_6$, $P6_7$, S_{OUT}, S_{CLK}		0 to 2 (Note 2)	mA
I_{OL2}	Circuit current	$P1_1$–$P1_4$		0 to 6 (Note 2)	mA
I_{OL3}	Circuit current	$P0_0$–$P0_2$, $P0_4$–$P0_7$, $P3_2$, $P4_7$, $P5_0$, $P5_1$, $P6_0$–$P6_2$		0 to 1 (Note 2)	mA
I_{OL4}	Circuit current	$P3_0$, $P3_1$		0 to 10 (Note 3)	mA
P_d	Power dissipation		T_a = 25 °C	550	mW
T_{opr}	Operating temperature			–10 to 70	°C
T_{stg}	Storage temperature			–40 to 125	°C

RECOMMENDED OPERATING CONDITIONS (T_a = –10 °C to 70 °C, V_{CC} = 5 V ± 10 %, unless otherwise noted)

Symbol	Parameter			Limits			Unit
				Min.	Typ.	Max.	
V_{CC}, AV_{CC}	Power source voltage (Note 4), During CPU, OSD, data slicer operation			4.5	5.0	5.5	V
V_{CC}, AV_{CC}	RAM hold voltage (when clock is stopped)			2.0		5.5	V
V_{SS}	Power source voltage			0	0	0	V
V_{IH1}	"H" input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $\overline{P3_0}$, $\overline{P3_1}$, $P4_0$–$P4_6$, $P6_4$, H_{SYNC}, V_{SYNC}, $\overline{RESET}$, X_{IN}, OSC1		0.8V_{CC}		V_{CC}	V
V_{IH2}	"H" input voltage	$P1_1$–$P1_4$ (When using I^2C-BUS)		0.7V_{CC}		V_{CC}	V
V_{IL1}	"L" input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P4_0$–$P4_6$, $P6_3$, $P6_4$		0		0.4 V_{CC}	V
V_{IL2}	"L" input voltage	SCL1, SCL2, SDA1, SDA2, (When using I^2C-BUS)		0		0.3 V_{CC}	V
V_{IL3}	"L" input voltage (Note 6)	$\overline{P4_1}$–$\overline{P4_4}$, $P4_6$, $P1_7$, H_{SYNC}, V_{SYNC}, $\overline{RESET}$, X_{IN}, OSC1		0		0.2 V_{CC}	V
I_{OH}	"H" average output current (Note 1)	R, G, B, OUT1, OUT2, $P0_3$, $P1_5$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$				1	mA
I_{OL1}	"L" average output current (Note 2)	R, G, B, OUT1, OUT2, $P0_3$, $P1_5$–$P1_7$, $P2_0$–$P2_7$, $P4_7$, $P5_1$, $P5_6$, $P5_7$, $P6_5$–$P6_7$, S_{OUT}, S_{CLK}				2	mA
I_{OL2}	"L" average output current (Note 2)	$P1_1$–$P1_4$				6	mA
I_{OL3}	"L" average output current (Note 2)	$P0_0$–$P0_2$, $P0_4$–$P0_7$, $P5_0$, $P6_0$–$P6_2$				1	mA
I_{OL4}	"L" average output current (Note 3)	$P3_0$, $P3_1$				10	mA
f_{CPU}	Oscillation frequency (for CPU operation) (Note 5)		X_{IN}	7.9	8.0	8.1	MHz
f_{CLK}	Oscillation frequency (for sub-clock operation)		X_{CIN}	29	32	35	kHz
f_{OSD}	Oscillation frequency (for OSD)	OSC1	LC oscillating mode	11.0		27.0	MHz
			Ceramic oscillating mode	26.5	27.0	27.5	
f_{hs1}	Input frequency	TIM2, TIM3, INT1, INT2, INT3				100	kHz
f_{hs2}	Input frequency	S_{CLK}				1	MHz
f_{hs3}	Input frequency	SCL1, SCL2				400	kHz
f_{hs4}	Input frequency	Horizontal sync. signal of video signal		15.262	15.734	16.206	kHz
V_I	Input amplitude video signal	CV_{IN}		1.5	2.0	2.5	V

ELECTRIC CHARACTERISTICS (V_{CC} = 5 V ± 10 %, V_{SS} = 0 V, f(X_{IN}) = 8 MHz, T_a = –10 °C to 70 °C, unless otherwise noted)

Symbol	Parameter		Test conditions		Limits			Unit
					Min.	Typ.	Max.	
I_{CC}	Power source current	System operation	V_{CC} = 5.5 V, f(X_{IN}) = 8 MHz	CRT OFF Data slicer OFF		15	30	mA
				CRT ON Data slicer ON		30	45	
			V_{CC} = 5.5 V, f(X_{IN}) = 0, f(X_{CIN}) = 32kHz, OSD OFF, Data slicer OFF, Low-power dissipation mode set (CM_5 = "0", CM_6 = "1")			60	200	µA
		Wait mode	V_{CC} = 5.5 V, f(X_{IN}) = 8 MHz			2	4	mA
			V_{CC} = 5.5 V, f(X_{IN}) = 0, f(X_{CIN}) = 32kHz, Low-power dissipation mode set (CM_5 = "0", CM_6 = "1")			25	100	µA
		Stop mode	V_{CC} = 5.5 V, f(X_{IN}) = 0, f(X_{CIN}) = 0			1	10	
V_{OH}	"H" output voltage	R, G, B, OUT1, OUT2, $P0_3$, $P1_5$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$	V_{CC} = 4.5 V I_{OH} = –0.5 mA		2.4			V
V_{OL}	"L" output voltage	R, G, B, OUT1, OUT2, S_{OUT}, S_{CLK}, $P0_0$–$P0_7$, $P1_5$–$P1_7$, $P2_0$–$P2_7$, $P5_0$, $P3_2$, $P4_7$, $P5_6$, $P5_7$, $P6_0$–$P6_2$, $P6_5$–$P6_7$	V_{CC} = 4.5 V I_{OL} = 0.5 mA				0.4	V
	"L" output voltage	$P3_0$, $P3_1$	V_{CC} = 4.5 V I_{OL} = 10.0 mA				3.0	
	"L" output voltage	$P1_1$–$P1_4$	V_{CC} = 4.5 V	I_{OL} = 3 mA			0.4	
				I_{OL} = 6 mA			0.6	
V_{T+}–V_{T-}	Hysteresis	$\overline{RESET}$	V_{CC} = 5.0 V			0.5	0.7	V
	Hysteresis (Note 6)	H_{SYNC}, V_{SYNC}, $P4_1$–$P4_4$, $P4_6$, $P1_1$–$P1_4$, $P1_7$	V_{CC} = 5.0 V			0.5	1.3	
I_{IZH}	"H" input leak current	$\overline{RESET}$, $P0_3$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P4_0$–$P4_6$, $P6_3$, $P6_4$, H_{SYNC}, V_{SYNC}	V_{CC} = 5.5 V V_I = 5.5 V				5	µA
I_{IZL}	"L" input leak current	$\overline{RESET}$, $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P4_0$–$P4_6$, $P6_3$, $P6_4$, H_{SYNC}, V_{SYNC}	V_{CC} = 5.5 V V_I = 0 V				5	µA
I_{IZH}	"H" input leak current	$P0_0$–$P0_2$, $P0_4$–$P0_7$, $P5_0$, $P6_0$–$P6_2$	V_{CC} = 5.5 V V_I = 12 V				10	µA
R_{BS}	I^2C-BUS·BUS switch connection resistor (between SCL1 and SCL2, SDA1 and SDA2)		V_{CC} = 4.5 V				130	Ω

Notes 1: The total current that flows out of the IC must be 20 or less.

2: The total input current to IC (I_{OL1} + I_{OL2} + I_{OL3}) must be 20 mA or less.

3: The total average input current for ports $P3_0$, $P3_1$ to IC must be 10 mA or less.

4: Connect 0.1 µF or more capacitor externally across the power source pins V_{CC}–V_{SS} and AV_{CC}–V_{SS} so as to reduce power source noise.
Also connect 0.1 µF or more capacitor externally across the pins V_{CC}–CNV_{SS}.

5: Use a quartz-crystal oscillator or a ceramic resonator for the CPU oscillation circuit. When using the data slicer, use 8 MHz.

6: $P1_6$, $P4_1$–$P4_4$ have the hysteresis when these pins are used as interrupt input pins or timer input pins. $P1_1$–$P1_4$ have the hysteresis when these pins are used as multi-master I^2C-BUS interface ports. $P1_7$ and $P4_6$ have the hysteresis when these pins are used as serial I/O pins.

A-D CONVERTER CHARACTERISTICS

(V_{CC} = 5 V ± 10 %, V_{SS} = 0 V, f(X_{IN}) = 8 MHz, T_a = –10 °C to 70 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	bits
—	Non-linearity error		0		±2	LSB
—	Differential non-linearity error		0		±0.9	LSB
V_{OT}	Zero transition error	V_{CC} = 5.12V I_{OL} (SUM) = 0mA	0		2	LSB
V_{FST}	Full-scale transition error	V_{CC} = 5.12V	0		4	LSB
T_{CONV}	Conversion time		12.25		12.5	µs
V_{REF}	Reference voltage				V_{CC}	V
R_{LADDER}	Ladder resistor			25		kΩ
V_{IA}	Analog input current		0		V_{REF}	V

MULTI-MASTER I²C-BUS BUS LINE CHARACTERISTICS

Symbol	Parameter	Standard clock mode		High-speed clock mode		Unit
		Min.	Max.	Min.	Max.	
t_{BUF}	Bus free time	4.7		1.3		µs
$t_{HD:STA}$	Hold time for START condition	4.0		0.6		µs
t_{LOW}	"L" period of SCL clock	4.7		1.3		µs
t_R	Rising time of both SCL and SDA signals		1000	20+0.1C_b	300	ns
$t_{HD:DAT}$	Data hold time	0		0	0.9	µs
t_{HIGH}	"H" period of SCL clock	4.0		0.6		µs
t_F	Falling time of both SCL and SDA signals		300	20+0.1C_b	300	ns
$t_{SU:DAT}$	Data set-up time	250		100		ns
$t_{SU:STA}$	Set-up time for repeated START condition	4.7		0.6		µs
$t_{SU:STO}$	Set-up time for STOP condition	4.0		0.6		µs

Note: C_b = total capacitance of 1 bus line

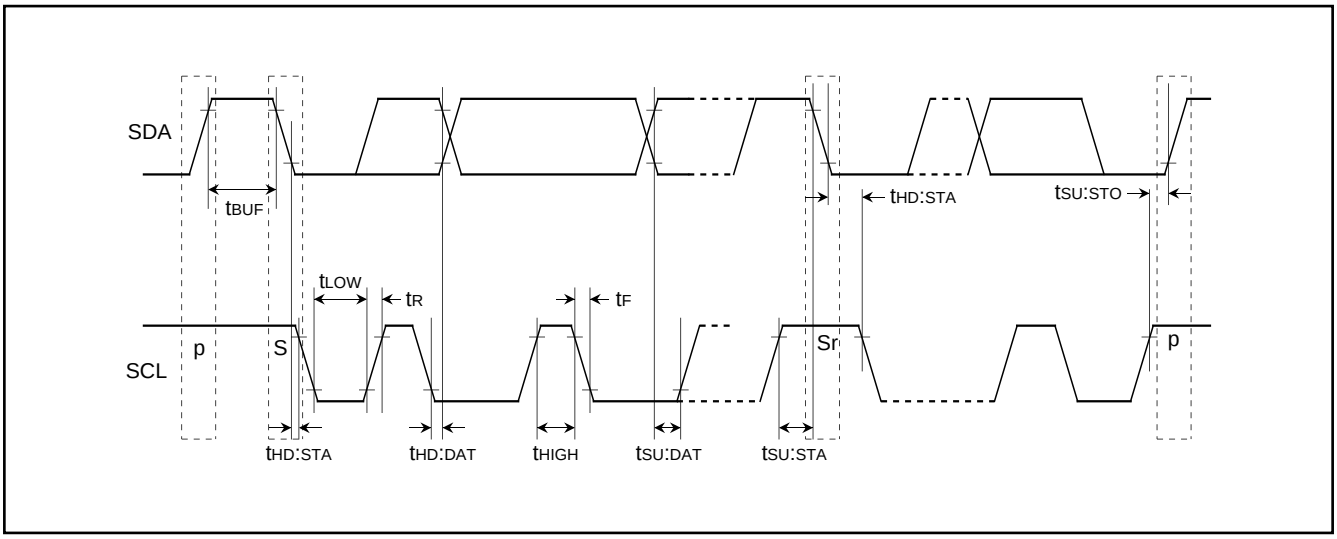


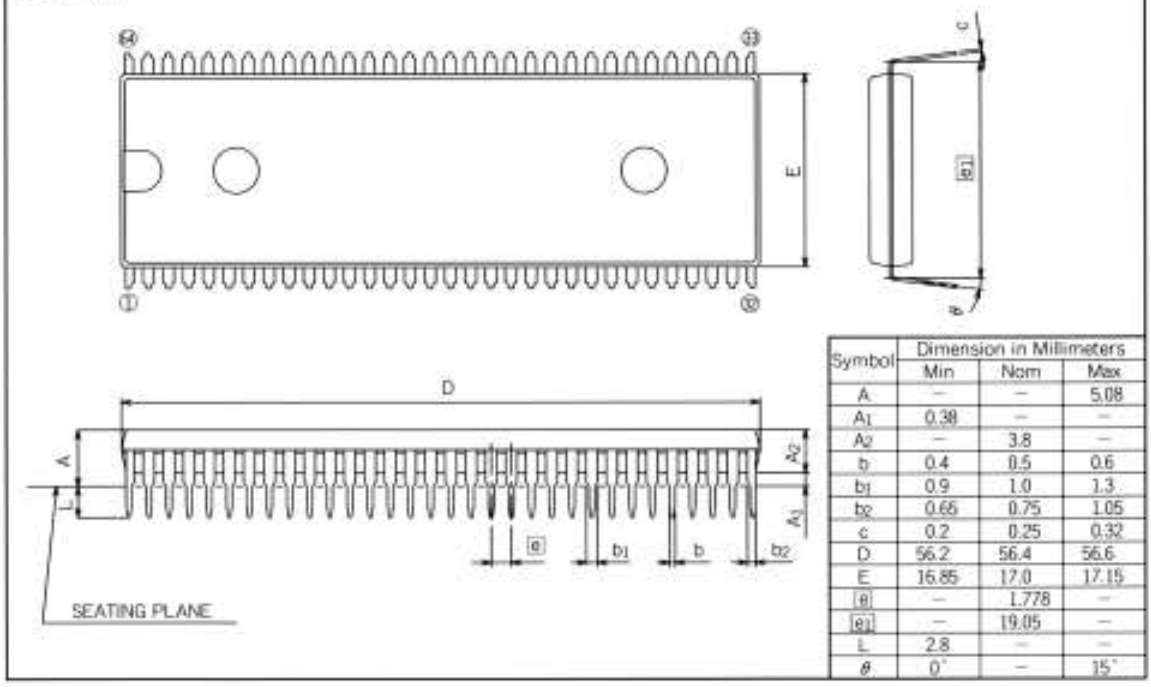
Fig. 98. Definition diagram of timing on multi-master I²C-BUS

PACKAGE OUTLINE

64P4B — Plastic 64pin 750mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP64-P-750-1.78	—	7.9	Alloy 42

Scale :



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	5.08
A_1	0.38	—	—
A_2	—	3.8	—
b	0.4	0.5	0.6
b_1	0.9	1.0	1.3
b_2	0.65	0.75	1.05
c	0.2	0.25	0.32
D	56.2	56.4	56.6
E	16.85	17.0	17.15
e	—	1.778	—
e1	—	19.05	—
L	2.8	—	—
θ	0°	—	15°

GZZ–SH09–39B < 51A0 >

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37270MF-XXXSP
MITSUBISHI ELECTRIC

Note : Please fill in all items marked ✵.

✵ Customer	Company name	TEL ( )	Issuance signature	Submitted by	Supervisor
	Date issued	Date :			

✵1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.

Three EPROMs are required for each pattern.

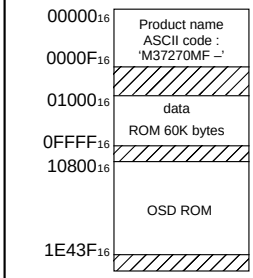
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)

☐ **27C101**

EPROM address



(1) Set "FF$_{16}$" in the shaded area.

(2) Write the ASCII codes that indicates the product name of "M37270MF–" to addresses 0000 $_{16}$ to 000F$_{16}$.

EPROM data check item (Refer the EPROM data and check " ✓ " in the appropriate box)

- Do you set "FF$_{16}$" in the shaded area ? → Yes ☐
- Do you write the ASCII codes that indicates the product name of "M37270MF–" to addresses 0000 $_{16}$ to 000F$_{16}$? → Yes ☐

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (64P4B for M37270MF-XXXSP) and attach to the mask ROM confirmation form.

GZZ–SH09–39B < 51A0 >

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37270MF-XXXSP
MITSUBISHI ELECTRIC

Writing the product name and character ROM data onto EPROMs

Addresses 00000_{16} to $0000F_{16}$ store the product name, and addresses 10800_{16} to $1E43F_{16}$ store the character pattern. If the name of the product contained in the EPROMs does not match the name on the mask ROM confirmation form, the ROM processing is disabled. Write the data correctly.

1. Inputting the name of the product with the ASCII code
   ASCII codes 'M37270MF-' are listed on the right.
   The addresses and data are in hexadecimal notation.

Address		Address	
0000_{16}	'M' = 4 D $_{16}$	0008_{16}	'–' = 2 D $_{16}$
0001_{16}	'3' = 3 3 $_{16}$	0009_{16}	F F $_{16}$
0002_{16}	'7' = 3 7 $_{16}$	$000A_{16}$	F F $_{16}$
0003_{16}	'2' = 3 2 $_{16}$	$000B_{16}$	F F $_{16}$
0004_{16}	'7' = 3 7 $_{16}$	$000C_{16}$	F F $_{16}$
0005_{16}	'0' = 3 0 $_{16}$	$000D_{16}$	F F $_{16}$
0006_{16}	'M' = 4 D $_{16}$	$000E_{16}$	F F $_{16}$
0007_{16}	'F' = 4 6 $_{16}$	$000F_{16}$	F F $_{16}$

2. Inputting the character ROM
   Input the character ROM data to character ROM. For the character ROM data, see the next page and on.

GZZ–SH09–39B < 51A0>

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37270MF-XXXSP
MITSUBISHI ELECTRIC

Font data must be stored in the proper OSD ROM address according to the following table.

(1)OSD ROM address of character font data

OSD ROM address bit	AD16	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Line number / Character code / Font bit	1	0	Line number					Character code									Font bit

Line number = 02_{16} to 15_{16}
Character code = 00_{16} to $13F_{16}$
Font bit = 0 : Left font
1 : Right font

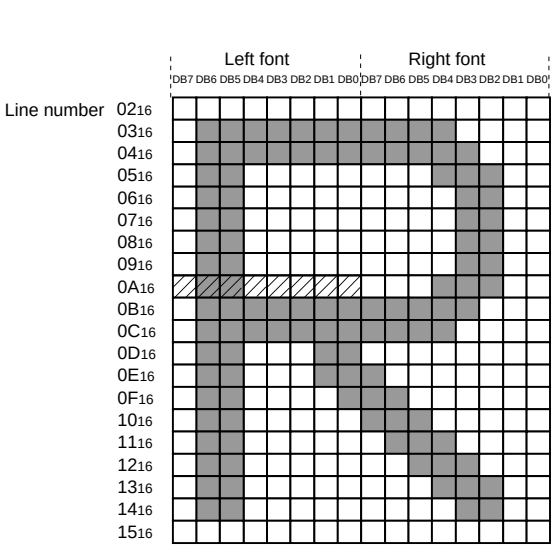
Example) The font data "60" (shaded area ▨) of the character code "AA_{16}" is stored in address
1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 0 $_2$ =12954_{16}.

(2)OSD ROM address of extra font data

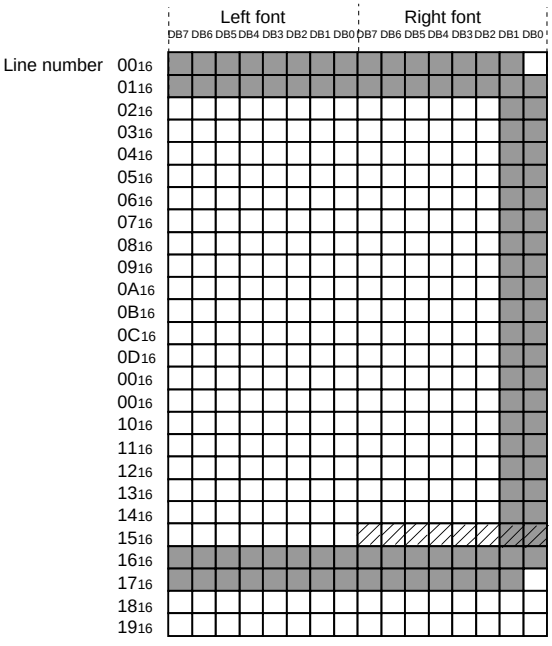
OSD ROM address bit	AD16	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Line number / Extra code / Font bit	1	1	Line number					0	0	0	0	Extra code					Font bit

Line number = 00_{16} to 19_{16}
Extra code = 00_{16} to $1F_{16}$
Font bit = 0 : Left font
1 : Right font

Example) The font data "03" (shaded area ▨) of the extra code "$0A_{16}$" is stored in address
1 1 0 0 1 0 1 0 0 0 0 0 1 0 1 0 1 $_2$ =19415_{16}.



(1) Character code "AA_{16}"



(2) Extra code "$0A_{16}$"

GZZ–SH09–39B < 51A0 >

740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37270MF-XXXSP
MITSUBISHI ELECTRIC

The following OSD ROM addresses must be set "FF." There are no font data in these addresses.

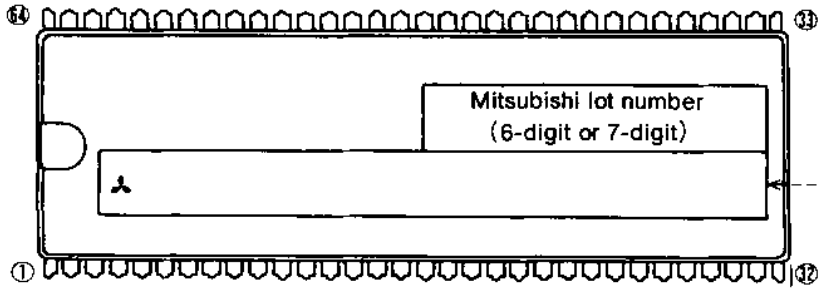
10A80$_{16}$ to 10BFF$_{16}$
10E80$_{16}$ to 10FFF$_{16}$
11280$_{16}$ to 113FF$_{16}$
11680$_{16}$ to 117FF$_{16}$
11A80$_{16}$ to 11BFF$_{16}$
11E80$_{16}$ to 11FFF$_{16}$
12280$_{16}$ to 123FF$_{16}$
12680$_{16}$ to 127FF$_{16}$
12A80$_{16}$ to 12BFF$_{16}$
12E80$_{16}$ to 12FFF$_{16}$
13280$_{16}$ to 133FF$_{16}$
13680$_{16}$ to 137FF$_{16}$
13A80$_{16}$ to 13BFF$_{16}$
13E80$_{16}$ to 13FFF$_{16}$
14280$_{16}$ to 143FF$_{16}$
14680$_{16}$ to 147FF$_{16}$
14A80$_{16}$ to 14BFF$_{16}$
14E80$_{16}$ to 14FFF$_{16}$
15280$_{16}$ to 153FF$_{16}$
15680$_{16}$ to 17FFF$_{16}$
18040$_{16}$ to 183FF$_{16}$
18440$_{16}$ to 187FF$_{16}$
18840$_{16}$ to 18BFF$_{16}$
18C40$_{16}$ to 18FFF$_{16}$
19040$_{16}$ to 193FF$_{16}$
19440$_{16}$ to 197FF$_{16}$
19840$_{16}$ to 19BFF$_{16}$
19C40$_{16}$ to 19FFF$_{16}$
1A040$_{16}$ to 1A3FF$_{16}$
1A440$_{16}$ to 1A7FF$_{16}$
1A840$_{16}$ to 1ABFF$_{16}$
1AC40$_{16}$ to 1AFFF$_{16}$
1B040$_{16}$ to 1B3FF$_{16}$
1B440$_{16}$ to 1B7FF$_{16}$
1B840$_{16}$ to 1BBFF$_{16}$
1BC40$_{16}$ to 1BFFF$_{16}$
1C040$_{16}$ to 1C3FF$_{16}$
1C440$_{16}$ to 1C7FF$_{16}$
1C840$_{16}$ to 1CBFF$_{16}$
1CC40$_{16}$ to 1CFFF$_{16}$
1D040$_{16}$ to 1D3FF$_{16}$
1D440$_{16}$ to 1D7FF$_{16}$
1D840$_{16}$ to 1DBFF$_{16}$
1DC40$_{16}$ to 1DFFF$_{16}$
1E040$_{16}$ to 1E3FF$_{16}$

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

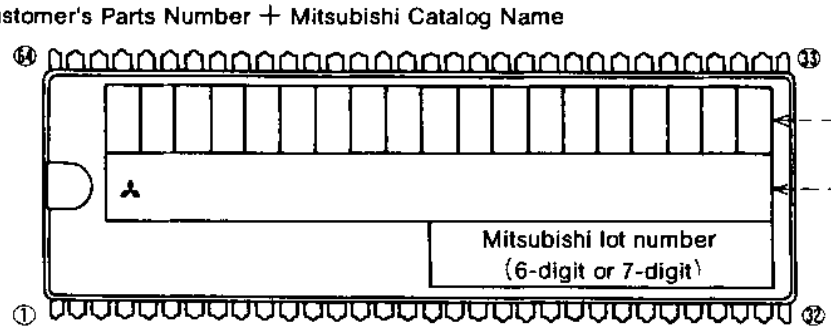
Mitsubishi IC catalog name

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



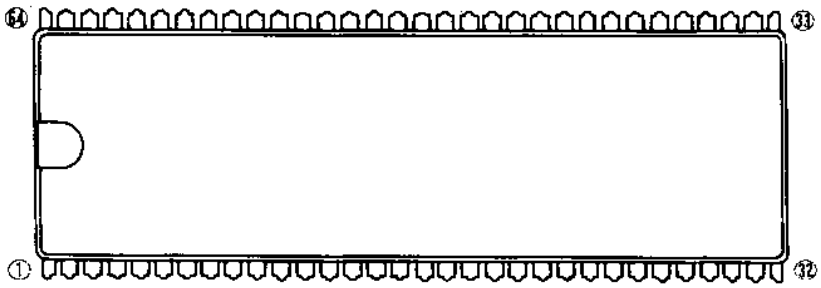
B. Customer's Parts Number + Mitsubishi Catalog Name



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.

Mitsubishi logo is not required ☐

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.

MITSUBISHI DATA BOOK
SINGLE-CHIP 8-BIT MICROCOMPUTERS Vol.3

Sep. First Edition 1996 H-DF319-B

Editioned by
Committee of editing of Mitsubishi Semiconductor Data Book

Published by
Mitsubishi Electric Corp., Semiconductor Division

REVISION DESCRIPTION LIST	M37270MF-XXXSP M37270EF-XXXSP, M37270EFSP DATA SHEET

Rev. No.	Revision Description	Rev. date
1.0	First Edition	9708
2.0	Information about copyright note, revision number, release date added (last page).	971130