CMOS 8-Bit Microcontroller TMP88CS34NG/FG, TMP88CP34NG/FG

The TMP88CS34/CP34 is the high speed and high performance 8-bit single chip microcomputers. This MCU contain CPU core, ROM, RAM, input/output ports, four Multi-function timer/counters, serial bus interface, on-screen display, PWM output, 8-bit AD converter, and remote control signal preprocessor on chip.

Product No.	ROM	RAM	Package	OTP MCU
TMP88CS34NG/FG	64 K × 8-bit	1.5 K × 8-bit	P-SDIP42-600-1.78	TMP88PS34NG/FG
TMP88CP34NG/FG	48 K × 8-bit	1.5 K × 0-bit	P-QFP44-1414-0.80D	TMF 00F 334NG/FG

Features

- ♦ 8-bit single chip microcomputer TLCS-870/X Series
- ♦ Instruction execution time: 0.25 µs (at 16 MHz)
- ♦ 842 basic instructions
 - Multiplication and Division (8 bits × 8 bits, 16 bits × 8 bits, 16 bits/8 bits)
 - Bit manipulations (Set/Clear/Complement/Move/Test/Exclusive or)
 - 16-bit data and 20-bit data operations
 - 1-byte jump/subroutine-call (Short relative jump/Vector call)

RESTRICTIONS ON PRODUCT USE

060116EBP

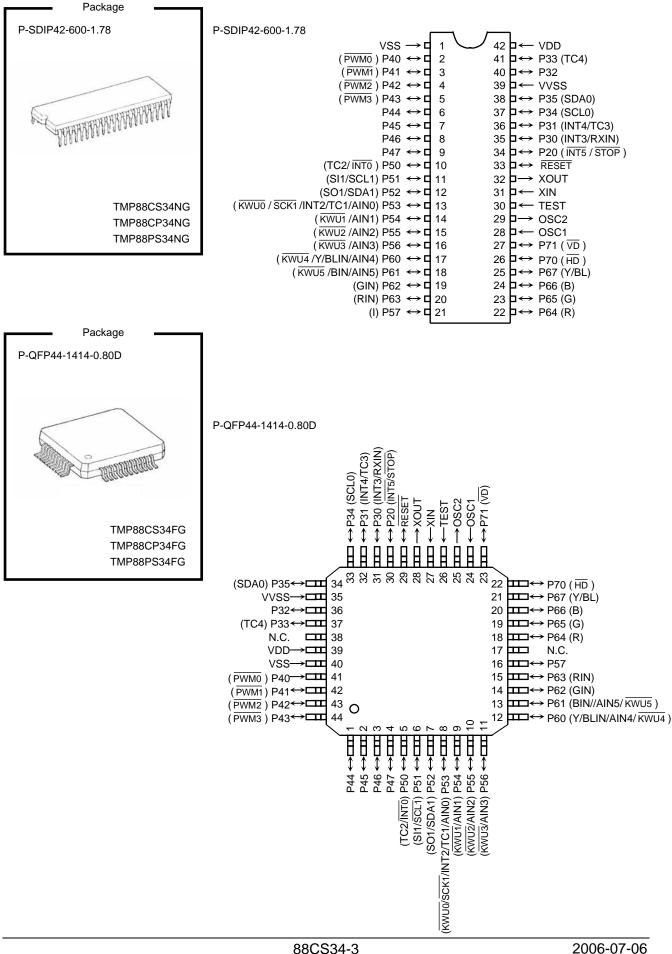
- The information contained herein is subject to change without notice. 021023_D
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor
 devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical
 stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety
 in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such
 TOSHIBA products could cause loss of human life, bodily injury or damage to property.

In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc. 021023_A

- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunctionor failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk. 021023_B
- The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations. 060106_Q
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of TOSHIBA or others. 021023_C
- The products described in this document are subject to the foreign exchange and foreign trade laws. 021023_E
- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions. 030619_s

- ◆ I/O ports: Maximum 33 (High current output: 4)
- ◆ 15 interrupt sources: External 6, Internal 10
 - All sources have independent latches each, and nested interrupt control is available.
 - Edge-selectable external interrupts with noise reject
 - High-speed task switching by register bank changeover
- ◆ ROM corrective function
- ◆ Two 16-bit timer/counters: TC1, TC2
 - Timer, Event-counter, Pulse width measurement, External trigger timer, Window modes
- ◆ Two 8-bit timer/counters: TC3, TC4
 - Timer, Event counter, Capture (Pulse width/duty measurement) mode
- ◆ Time base timer (Interrupt frequency: 0.95 Hz to 31250 Hz)
- ♦ Watchdog timer
 - Interrupt source/reset output
- ♦ Serial bus interface
 - I²C bus, 8-bit SIO mode (Selectable two I/O channels)
- ◆ On-screen display circuit
 - Font ROM characters: Mono font 383 characters, color font 96 characters or mono font 447
 - characters, color font 64 characters
 - Characters display: 32 columns × 12 lines
 - Composition: 16×18 dots
 - Size of character: 4 kinds (line by line)
 - Color of character: 8 or 27 kinds (character by character)
 - Variable display position: Horizontal 256 steps, Vertical 625 steps
 - Fringing, Smoothing, Slant, Underline, Blinking function
- \blacklozenge Jitter elimination
- ◆ DA conversion (Pulse Width Modulation) outputs
 - 14/12-bit resolution (2 channels)
 - 12-bit resolution (2 channels)
- \blacklozenge 8-bit successive approximate type AD converter with sample and hold
- ♦ High current output: 1 pin (typ. 20 mA)
- ◆ Remote control signal preprocessor
- ◆ Two power saving operating modes
 - STOP mode: Oscillation stops. Battery/Capacitor back-up. Port output hold/high-impedance.
 - IDLE mode: CPU stops, and Peripherals operate using high-frequency clock. Release by interrupts.
- ♦ Operating voltage: 4.5 to 5.5 V at 16 MHz
- ♦ Emulation POD: BM88CS34N0A-M15

Pin Assignments



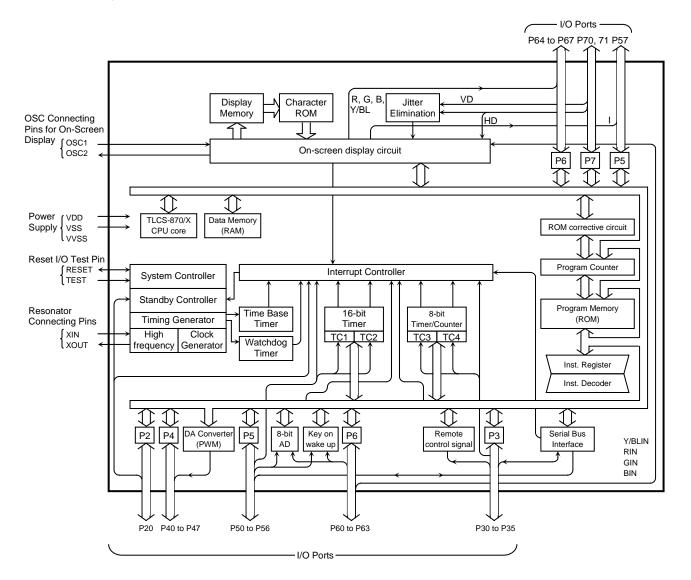
Pin Functions (1/2)

Pin Name	I/O	Function		
P20 (INT5 / STOP)	I/O (Input)	1-bit input/output port with latch. When used as an input port, the latch must be set to "1".	External interrupt input 5 or STOP mode release signal input	
P35 (SDA0)	I/O (Input/Output)	6-bit programmable input/output	I ² C bus serial data input/output 0	
P34 (SCL0)	I/O (Input/Output)	port. Each bit of these ports can be	I ² C bus serial clock input/output 0	
P33 (TC4)	I/O (Input)	individually configured as an input or an output under software control.		
P32	I/O	During reset, all bits are configured as inputs. When used as a serial bus	Video signal input 1 or Composite sync input	
P31 (INT4/TC3)	I/O (Input)	interface input/output, the latch must be set to "1".	External interrupt input 4 or Timer/Counter input 3	
P30 (INT3/RXIN)	I/O (Input)		External interrupt input 3 or Remote control signal preprocessor input	
P47	I/O	8-bit programmable input/output		
P46	I/O	port. Each bit of these ports can be individually configured as an input or		
P45	I/O	an output under software control.		
P44	I/O	During reset, all bits are configured		
P43 (PWM3)	I/O (Output)	as inputs.	12-bit DA conversion (PWM) outputs	
P42 (PWM2)	I/O (Output)			
P41 (PWM1)	I/O (Output)		14/12-bit DA conversion (PWM)	
P40 (PWM0)	I/O (Output)		outputs	
P57 (I)	I/O (Output)	8-bit programmable input/output	Translucent signal output	
P56 (KWU3 /AIN3)	I/O (Input)	port. Each bit of these ports can be individually configured as an input or	Key on wake-up inputs or AD	
P55 (KWU2 /AIN2)	I/O (Input)	an output under software control.	converter analog inputs	
P54 (KWU1 /AIN1)	I/O (Input)	During reset, all bits are configured		
P53 (KWU0 /AIN0/TC1 /INT2/ SCK1)	I/O (Input/Input/Input /Input/Output)	as inputs. When used as a serial bus interface input/output, the latch must be set to "1".	Key on wake-up input or AD converter analog input or Timer/counter input 1 or External interrupt input 2 or SIO serial clock input/output 1	
P52 (SDA1/SO1)	I/O (Input/Output/Output)		I ² C bus serial data Input/Output 1 or SIO serial data output 1	
P51 (SCL1/SI1)	l/O (Input/Output/Input)		I ² C bus serial data Input/Output 1 or SIO serial data input 1	
P50 (TC2/INT0)	l/O (Input/Input)		Timer/Counter input 2 or External interrupt input 0	
P67 (Y/BL)	I/O (Output)	8-bit programmable input/output	Y or BL output	
P66 (B)	I/O (Output)	port. (P67 to 61: Tri-State, P60: High		
P65 (G)	I/O (Output)	current output) Each bit of these ports can be individually configured	R/G/B outputs	
P64 (R)	I/O (Output)	as an input or an output under		
P63 (RIN)	I/O (Input)	software control. During reset, all	R input	
P62 (GIN)	I/O (Input)	bits are configured as inputs. When	G input	
P61 (KWU5 /BIN/AIN5)	I/O (Input)	used P64 to P67 as port, each bit of the P6 port data selection register (bit 7 to 4 in ORP6S) must be set to "1".	Key on wake-up input 5 or B input or AD converter analog input 5	
P60 (KWU4 /YBLIN/AIN4)	I/O (Input)	P63 to P61 output "0" after a reset. When these dual-function pins are used as ports, be sure to set ORP6S2 to "1".	Key on wake-up input 4 or Y/BL input or AD converter analog input 4	

Pin Functions (2/2)

Pin Name	I/O	Function			
P71 (\(\vec{VD}\))	I/O (Input)	2-bit programmable input/output port. Each bit of these ports can be individually configured as an input or			
P70 (HD)	I/O (Input)	an output under software control. During reset, all bits are configured as inputs.	Horizontal synchronous signal input		
XIN, XOUT	Input, Output	Resonator connecting pins. For inputting external clock, XIN is used and XOUT is opened.			
RESET	I/O	Reset signal input or watchdog timer output/address-trap-reset output/system-clock-rest output			
TEST	Input	Test pin for out-going test. Be tied to low.			
OSC1, OSC2	Input, Output	Resonator connecting pins for on-screen display circuitry			
VDD, VSS, VVSS	Power Supply	+5 V, 0 V (GND)	+5 V, 0 V (GND)		

Block Diagram



Operational Description

1. CPU Core Functions

The CPU core consists of a CPU, a system clock controller, and an interrupt controller. This section provides a description of the CPU core, the program memory, the data memory, the external memory interface, and the reset circuit.

1.1 Memory Address Map

The TMP88CS34/CP34 memory consists of four blocks: ROM, RAM, SFR (Special Function Register), and DBR (Data Buffer Register). They are all mapped to a 1-Mbyte address space. Figure 1.1.1 shows the TMP88CS34/CP34 memory address map. There are 16 banks of the general-purpose register. The register banks are also assigned to the RAM address space.

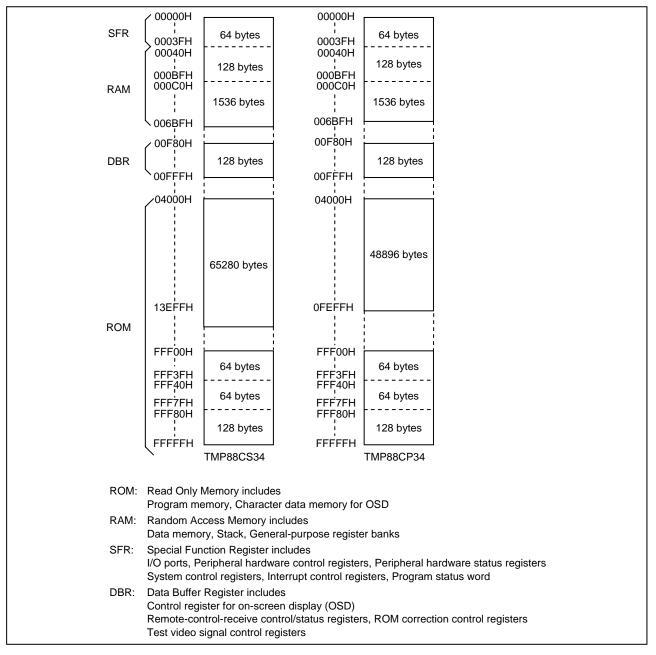


Figure 1.1.1 Memory Address Map

1.2 Program Memory (ROM)

The TMP88CS34 contains a 64-K byte program memory (mask ROM) at addresses from 04000 to 13 EFFH and FFF00 to FFFFFH.

The TMP88CP34 contains a 48-Kbyte program memory (mask ROM) at address from 04000 to 0FEFFH and FFF00 to FFFFFH.

Addresses FFF00 through FFFFFH in the program memory are also used for a particular purpose.

1.3 Data Memory (RAM)

The TMP88CS34/CP34 has a 1.5-Kbyte data memory (Static RAM) address from 0040 to 06BFH.

The first 128 bytes (addresses 00040 through 000BFH) in the built-in RAM are also available as general-purpose register banks.

The general-purpuse registers are mapped in the RAM; therefore, do not clear RAM at the current bank addresses.

Example: Clears RAM to "00H" except the bank 0 (TMP88CS34/CP34):

SRAMCLR:	DEC	HL, 0048H A, H BC, 0677H (HL+), A BC F, SRAMCLR	;	Sets start address to HL register pair Sets initial data (00H) to A register Sets number of byte to BC register pair
----------	-----	--	---	--

Note: The data memory contents become unstable when the power supply is turned on; therefore, the data memory should be initialized by an initialization routine. Note that the general-purpose registers are mapped in the RAM; therefore, do not clear RAM at the current bank addresses.

1.4 System Clock Controller

The system clock controller consists of a clock generator, a timing generator, and a stand-by controller.

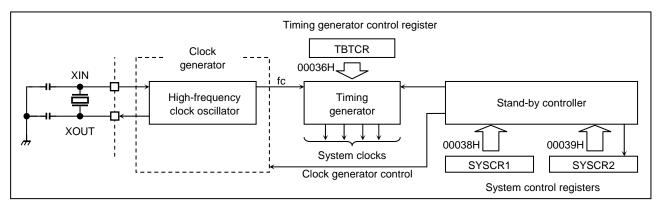


Figure 1.4.1 System Clock Controller

1.4.1 Clock Generator

The clock generator generates the basic clock which provides the system clocks supplied to the CPU core and peripheral hardware. It contains oscillation circuit: one for the high-frequency clock.

The high-frequency (fc) clock can be easily obtained by connecting a resonator between the XIN/XOUT pin, respectively. Clock input from an external oscillator is also possible. In this case, external clock is applied to XIN pin with XOUT pin not connected.

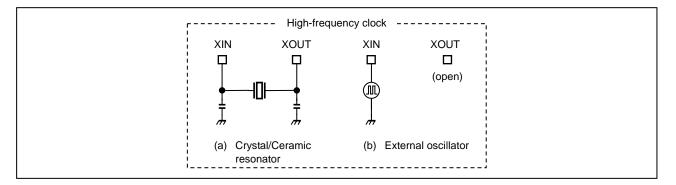


Figure 1.4.2 Examples of Resonator Connection

Note: Accurate adjustment of the oscillation frequency:

Although hardware to externally and directly monitor the basic clock pulse is not provided, the oscillation frequency can be adjusted by making the program to output fixed frequency pulses to the port while disabling all interrupts and monitoring this pulse. With a system requiring adjustment of the oscillation frequency, the adjusting program must be created beforehand.

1.4.2 Timing Generator

The timing generator generates from the basic clock the various system clocks supplied to the CPU core and peripheral hardware. The timing generator provides the following functions:

- 1. Generation of main system clock
- 2. Generation of source clocks for time base timer
- 3. Generation of source clocks for watchdog timer
- 4. Generation of internal source clocks for timer/counters TC1 to TC4
- 5. Generation of warm-up clocks for releasing STOP mode
- 6. Generation of a clock for releasing reset output
- (1) Configuration of Timing Generator

The timing generator consists of a 21-stage divider with a divided-by-3 prescaler, a main system clock generator, and machine cycle counters.

During reset and at releasing STOP mode, the prescaler and the divider are cleared to "0", however, the prescaler is not cleared.

An input clock to the 7th stage of the divider depends on the operating mode.

A divided-by-256 of high-frequency clock $(fc/2^8)$ is input to the 7th stage of the divider.

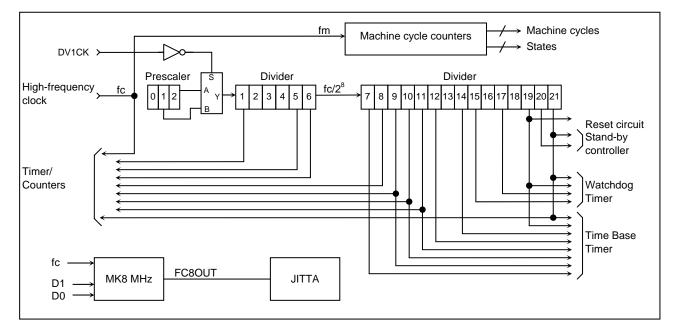
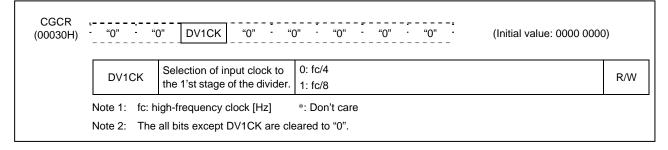
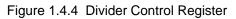


Figure 1.4.3 Configuration of Timing Generator





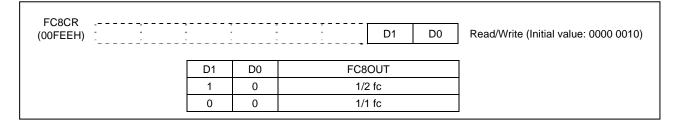


Figure 1.4.5 FC8 Control Register

(2) Machine Cycle

Instruction execution and peripheral hardware operation are synchronized with the main system clock. The minimum instruction execution unit is called a "machine cycle".

There are a total of 15 different types of instructions for the TLCS-870/X Series: ranging from 1-cycle instructions which require one machine cycle for execution to 15-cycle instructions which require 15 machine cycles for execution.

A machine cycle consists of 4 states (S0 to S3), and each state consists of one main system clock.

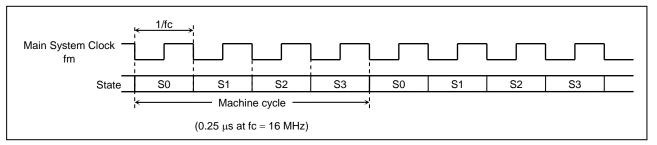


Figure 1.4.6 Machine Cycle

1.4.3 Stand-by Controller

The stand-by controller starts and stops the switches the main system clock. These modes are controlled by the system control registers (SYSCR1, SYSCR2).

Figure 1.4.7 shows the operating mode transition diagram and Figure 1.4.8 shows the system control registers.

Single-clock mode

In the single-clock mode, the machine cycle time is 4/fc [s] (0.25 µs at fc = 16 MHz).

1. NORMAL mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock.

2. IDLE mode

In this mode, the internal oscillation circuit remains active. The CPU and the watchdog timer are halted; however, on-chip peripherals remain active (operate using the high-frequency clock). IDLE mode is started by setting IDLE bit in the system control register 2 (SYSCR2), and IDLE mode is released to NORMAL mode by an interrupt request from on-chip peripherals or external interrupt inputs. When IMF (interrupt master enable flag) is "1" (interrupt enable), the execution will resume upon acceptance of the interrupt, and the operation will return to normal after the interrupt service is completed. When IMF is "0" (interrupt disable), the execution will resume with the instruction which follows IDLE mode start instruction.

3. STOP mode

In this mode, the internal oscillation circuit is turned off, causing all system operations to be halted.

The internal status immediately prior to the halt is held with the lowest power consumption during this mode.

STOP mode is started by setting STOP bit in the system control register 1 (SYSCR1), and STOP mode is released by an input (either level-sensitive or edge-sensitive can be programmably selected) to the $\overline{\text{STOP}}$ pin. After the warming-up period is completed, the execution resumes with the next instruction which follows the STOP mode start instruction.

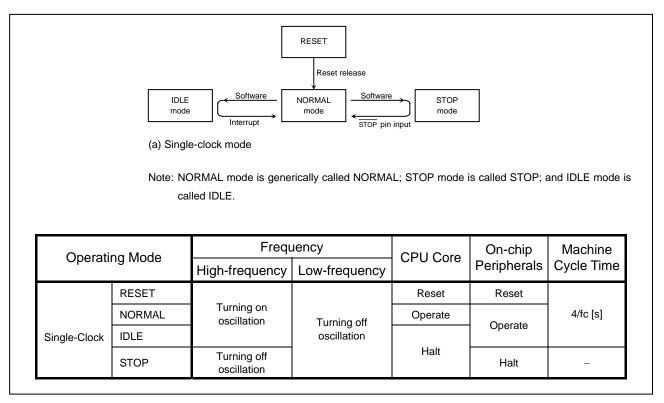


Figure 1.4.7 Operating Mode Transition Diagram

SYSCR1 7 6 5 4 3 2 1 0 (00038H) STOP RELM "0" "1" WUT (Initial value: 0000 00**)
(00038H) STOP RELM "0" "1" WUT (Initial value: 0000 00**)
0: CPU core and peripherals remain active
STOP STOP mode start 1: CPU core and peripherals are halted
(start STOP mode)
RELM Release method for STOP mode 0: Edge-sensitive release (Rising Edge)
1: Level-sensitive release ("H" Level)
Return to NORMAL mode R/W
DV1CK = 0DV1CK = 1Warming-up time at releasing00 3×2^{16} /fc 3×2^{17} /fc
WUTWarming-up time at releasing STOP mode00 $3 \times 2^{16}/fc$ $3 \times 2^{17}/fc$ UUTSTOP mode01 $2^{16}/fc$ $2^{17}/fc$
10 Reserved Reserved
11 Reserved Reserved
Note 1: Always set bit 5 in SYSCR1 to "0".
Note 2: When STOP mode is released with RESET pin input, a return is made to NORMAL mode regardless of the
RETM contents.
Note 3: fc: High-frequency clock [Hz]
*: Don't care
Note 4: Bits 1 and 0 in SYSCR1 are read in as undefined data when a read instruction is executed.
Note 5: Always set bit 4 in SYSCR1 to "1" when STOP mode is started.
System Control Register 2
SYSCR2 7 6 5 4 3 2 1 0
(00039H) <u>"1" "0" "0" IDLE</u> (Initial value: 1000 ****)
· · · · · · · · · · · · · · · · · · ·
IDLE IDLE mode start 0: CPU and watchdog timer remain active R/W
1: CPU and watchdog timer are stopped (start IDLE mode)
Note 1: *: Don't care
Note 2: Always set bit 7, 6 and 5 in SYSCR2 to "100".

Figure 1.4.8 System Control Registers

1.4.4 Operating Mode Control

(1) STOP mode

STOP mode is controlled by the system control register 1 (SYSCR1) and the STOP pin input. The STOP pin is also used both as a port P20 and an INT5 (external interrupt input 5) pin. STOP mode is started by setting STOP (bit 7 in SYSCR1) to "1". During STOP mode, the following status is maintained.

- 1. Oscillations are turned off, and all internal operations are halted.
- 2. The data memory, registers and port output latches are all held in the status in effect before STOP mode was entered.
- 3. The prescaler and the divider of the timing generator are cleared to "0".
- 4. The program counter holds the address of the instruction following the instruction which started the STOP mode.

STOP mode includes a level-sensitive release mode and an edge-sensitive release mode, either of which can be selected with RELM (bit 6 in SYSCR1).

a. Level-sensitive release mode (RELM = 1)

In this mode, STOP mode is released by setting the STOP pin high. This mode is used for capacitor back-up when the main power supply is cut off and long term battery back-up.

When the $\overline{\text{STOP}}$ pin input is high, executing an instruction which starts the STOP mode will not place in STOP mode but instead will immediately start the release sequence (warm-up). Thus, to start STOP mode in the level-sensitive release mode, it is necessary for the program to first confirm that the $\overline{\text{STOP}}$ pin input is low. The following method can be used for confirmation:

Using an external interrupt input $\overline{INT5}$ ($\overline{INT5}$ is a falling edge-sensitive input).

Example: Starting STOP mode with an INT5 interrupt.

 PINT5: TEST (P2).0
 ; To reject noise, the STOP mode does not start if port P20 is at high

 LD
 (SYSCR1), 01010000B
 ; Sets up the level-sensitive release mode.

 SET
 (SYSCR1).7
 ; Starts STOP mode

 LDW
 (IL), 11100111010111B
 ; IL12, 11, 7, 5, 3 ← 0 (Clears interrupt latches)

 SINT5:
 RETI

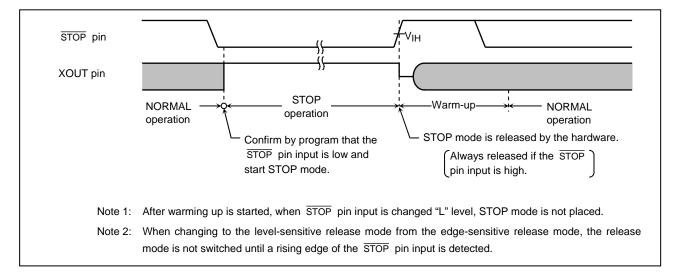


Figure 1.4.9 Level-sensitive Release Mode

b. Edge-sensitive release mode (RELM = 0)

In this mode, STOP mode is released by a rising edge of the $\overline{\text{STOP}}$ pin input. This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is input to the $\overline{\text{STOP}}$ pin.

In the edge-sensitive release mode, STOP mode is started even when the $\overline{\text{STOP}}$ pin input is high.

Example: Starting STOP mode from NORMAL mode

LD (SYSCR1), 10010000B ; Starts after specified to the edge-sensitive mode

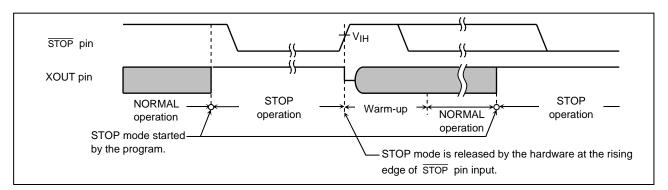


Figure 1.4.10 Edge-sensitive Release Mode

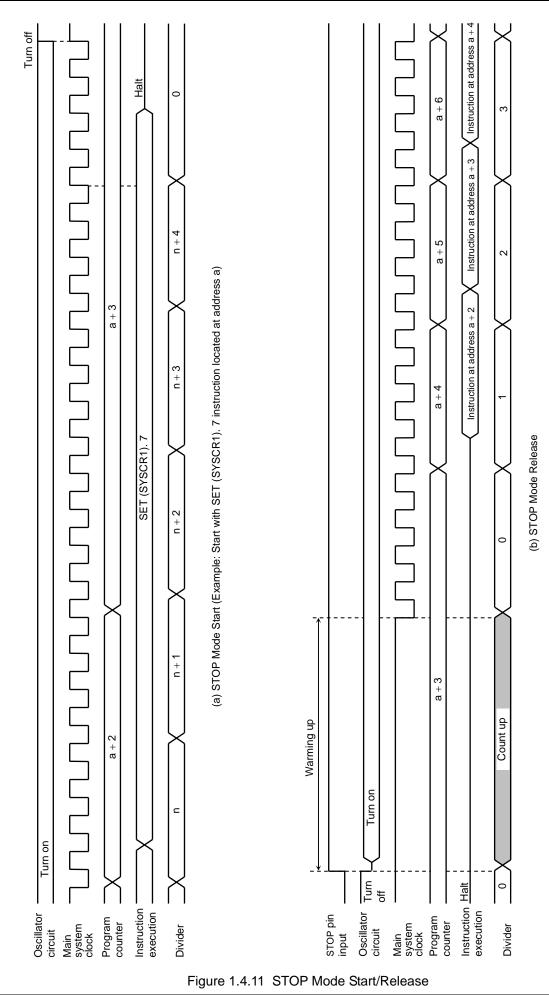
<u>STOP mode is released</u> by the following sequence:

- 1. When returning to NORMAL, clock oscillator is turned on.
- 2. A warming-up period is inserted to allow oscillation time to stabilize. During warm-up, all internal operations remain halted. Two different warming-up times can be selected with WUT (bits 2 and 3 in SYSCR1) as determined by the resonator characteristics.
- 3. When the warming-up time has elapsed, normal operation resumes with the instruction following the STOP mode start instruction (e.g. [SET (SYSCR1). 7]). The start is made after the divider of the timing generator is cleared to "0".

	Warming-up Time [ms]						
WUT	R	Return to NORMAL mode					
	DV1Ck	DV1CK = 0 DV1CK = 1					
00	3×2^{16} /fc	(12.29)	3×2^{17} /fc	(24.58)			
01	2 ¹⁶ /fc	(4.10)	2 ¹⁷ /fc	(8.20)			
10	Reserved	(-)	Reserved	(-)			
11	Reserved	(-)	Reserved	(-)			

 Table 1.4.1
 Warming-up Time Example

Note: The warming-up time is obtained by dividing the basic clock by the divider: therefore, the warming-up time may include a certain amount of error if there is any fluctuation of the oscillation frequency when STOP mode is released. Thus, the warming-up time must be considered an approximate value.



TOSHIBA

TMP88CS34/CP34

STOP mode can also be released by setting the $\overline{\text{RESET}}$ pin low, which immediately performs the normal reset operation.

Note: When STOP mode is released with a low hold voltage, the following cautions must be observed.

The power supply voltage must be at the operating voltage level before releasing STOP mode. The RESET pin input must also be high, rising together with the power supply voltage. In this case, if an external time constant circuit has been connected, the RESET pin input voltage will increase at a slower rate than the power supply voltage. At this time, there is a danger that a reset may occur if input voltage level of the RESET pin drops below the non-inverting high-level input voltage (hysteresis input).

(2) IDLE mode

IDLE mode is controlled by the system control register 2 and maskable interrupts. The following status is maintained during IDLE mode.

- 1. Operation of the CPU and watchdog timer is halted. On-chip peripherals continue to operate.
- 2. The data memory, CPU registers and port output latches are all held in the status in effect before IDLE mode was entered.
- 3. The program counter holds the address of the instruction following the instruction which started IDLE mode.

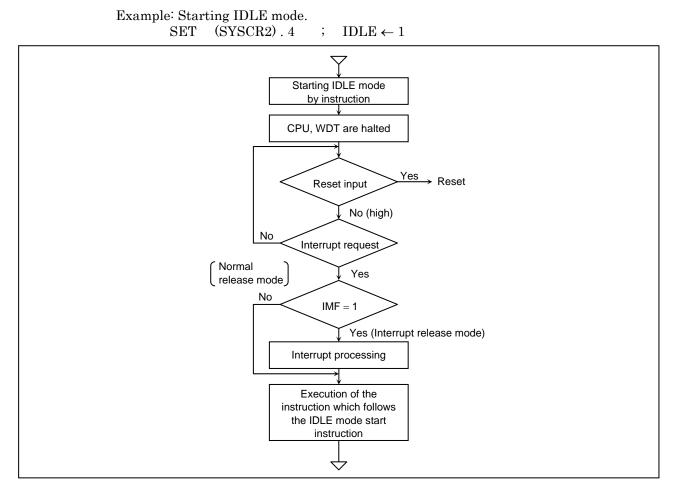


Figure 1.4.12 IDLE Mode

IDLE mode includes a normal release mode and an interrupt release mode. Selection is made with the interrupt master enable flag (IMF). Releasing the IDLE mode returns from IDLE to NORMAL.

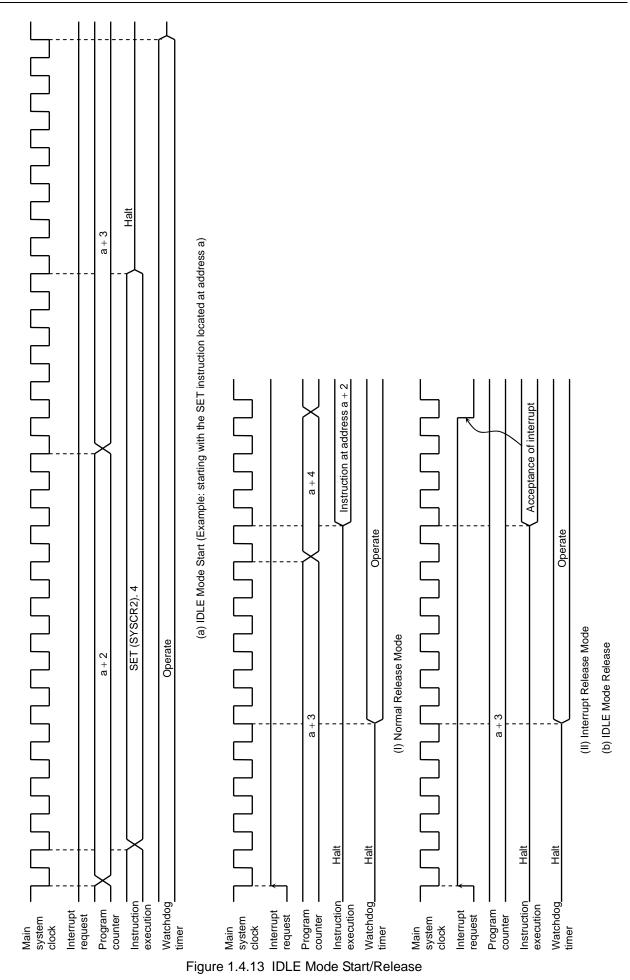
a. Normal release mode (IMF = "0")

IDLE mode is released by any interrupt source enabled by the individual interrupt enable flag (EF) or an external interrupt 0 ($\overline{\text{INT0}}$ pin) request. Execution resumes with the instruction following the IDLE mode start instruction (e.g. [SET (SYSCR2).4]). Normally, IL (Interrupt Latch) of interrupt source to release IDLE mode must be cleared by load instructions.

b. Interrupt release mode (IMF = "1")

IDLE mode is released and interrupt processing is started by any interrupt source enabled with the individual interrupt enable flag (EF) or an external interrupt 0 ($\overline{\text{INT0}}$ pin) request. After the interrupt is processed, the execution resumes from the instruction following the instruction which started IDLE mode.

Note: When a watchdog timer interrupt is generated immediately before the IDLE mode is started, the watchdog timer interrupt will be processed but IDLE mode will not be started.



IDLE mode can also be released by setting the $\overline{\text{RESET}}$ pin low, which immediately performs the reset operation. After reset, the TMP88CS34/CP34 is placed in NORMAL mode.

1.5 Interrupt Controller

The TMP88CS34/CP34 has a total of 16 interrupt sources; 6 externals and 10 internals. Multiple interrupts with priorities are also possible. Two of the internal sources are pseudo non-maskable interrupts; the remainder are all maskable interrupts.

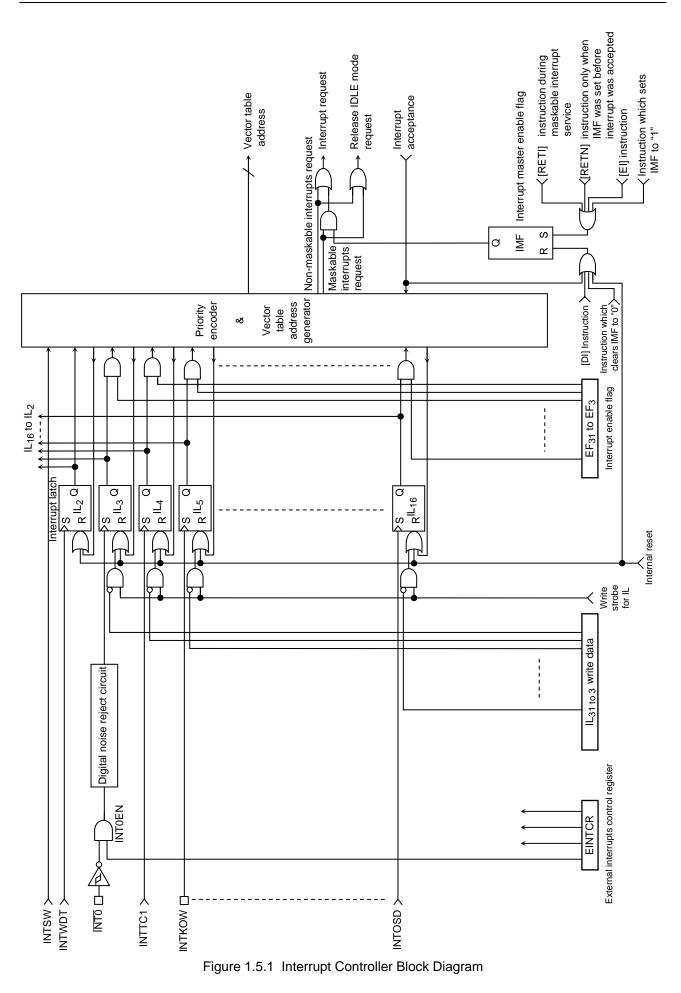
Interrupt sources are provided with interrupt latches (IL), which hold interrupt requests, and independent vectors. The interrupt latch is set to "1" by the generation of its interrupt request which requests the CPU to accept its interrupts. Interrupts are enabled or disabled by software using the interrupt master enable flag (IMF) and interrupt enable flag (EF). If more than one interrupts are generated simulaneously, interrupts are accepted in order which is dominated by hardware. However, there are no prioritized interrupt factors among non-maskable interrupts.

	Inte	rrupt source	Enable condition	Interrupt latch	Vector table address	Priority
Internal/ External	(Reset)		Non-Maskable	_	FFFFCH	High 0
Internal	INTSW	(Software interrupt)	Pseudo non-maskable	_	FFFF8H	1
Internal	INTWDT	(Watchdog timer interrupt)	Pseudo non-maskable	IL ₂	FFFF4H	2
External	INT0	(External interrupt 0)	$IMF \cdot EF_3 = 1$, $INTOEN = 1$	IL ₃	FFFF0H	3
Internal	INTTC1	(16-bit TC1 interrupt)	$IMF \cdot EF_4 = 1$	IL ₄	FFFECH	4
External	INTKWU	(Key-On-Wake-Up)	$IMF \cdot EF_5 = 1$	IL ₅	FFFE8H	5
Internal	INTTBT	(Time base timer interrupt)	$IMF \cdot EF_6 = 1$	IL ₆	FFFE4H	6
External	INT2	(External interrupt 2)	$IMF \cdot EF_7 = 1$	IL ₇	FFFE0H	7
Internal	INTTC3	(8-bit TC3 interrupt)	IMF·EF ₈ = 1	IL ₈	FFFDCH	8
Internal	INTTSBI	(SBI interrupt)	$IMF \cdot EF_9 = 1$	IL ₉	FFFD8H	9
Internal	INTTC4	(8-bit TC4 interrupt)	$IMF \cdot EF_{10} = 1$	IL ₁₀	FFFD4H	10
External	INT3	(External interrupt 3)	$IMF \cdot EF_{11} = 1$	IL ₁₁	FFFD0H	11
External	INT4	(External interrupt 4)	$IMF \cdot EF_{12} = 1$	IL ₁₂	FFFCCH	12
Internal	INTADC	(AD Converter interrupt)	$IMF \cdot EF_{13} = 1$	IL ₁₃	FFFC8H	13
Internal	INTTC2	(16-bit TC2 interrupt)	$IMF \cdot EF_{14} = 1$	IL ₁₄	FFFC4H	14
External	INT5	(External interrupt 5)	$IMF \cdot EF_{15} = 1$	IL ₁₅	FFFC0H	15
Internal	INTOSD	(OSD interrupt)	$IMF \cdot EF_{16} = 1$	IL ₁₆	FFFBCH	16
		Reserved	$IMF \cdot EF_{17} = 1$	IL ₁₇	FFFB8H	17
		Reserved	IMF • EF ₁₈ = 1	IL ₁₈	FFFB4H	18
		Reserved	$IMF \cdot EF_{19} = 1$	IL ₁₉	FFFB0H	19
		Reserved	$IMF \cdot EF_{20} = 1$	IL ₂₀	FFFACH	20
		Reserved	$IMF \cdot EF_{21} = 1$	IL ₂₁	FFFA8H	21
		Reserved	$IMF \cdot EF_{22} = 1$	IL ₂₂	FFFA4H	22
		Reserved	IMF • EF ₂₃ = 1	IL ₂₃	FFFA0H	23
		Reserved	$IMF \cdot EF_{24} = 1$	IL ₂₄	FFF9CH	24
		Reserved	$IMF \cdot EF_{25} = 1$	IL ₂₅	FFF98H	25
		Reserved	IMF • EF ₂₆ = 1	IL ₂₆	FFF94H	26
		Reserved	$IMF \cdot EF_{27} = 1$	IL ₂₇	FFF90H	27
		Reserved	$IMF \cdot EF_{28} = 1$	IL ₂₈	FFF8CH	28
		Reserved	$IMF \cdot EF_{29} = 1$	IL ₂₉	FFF88H	29
		Reserved	$IMF \cdot EF_{30} = 1$	IL ₃₀	FFF84H	30
		Reserved	$IMF \cdot EF_{31} = 1$	IL ₃₁	FFF80H	Low 31

Note : Before you change each enable flag (EF) and/or each interrupt latch (IL), be sure to clear the interrupt master enable flag (IMF) to "0" (to disable interrupts).

- 1. After a DI instruction is executed.
- 2. When an interrupt is accepted, IMF is autamatically cleared to "0".
 - However, to enable nested interrupts change EF and/or IL before setting IMF to "1" (to enable interrupts).

If the individual enable flags (EF) and interrupts (IL) are set under conditions other than the above, proper operation cannot be guararteed.



Interrupt latches (IL) that hold the interrupt requests are provided for interrupt sources. Each interrupt vector is independent.

The interrupt latch is set to "1" when an interrupt request is generated, and requests the CPU to accept the interrupt. The acceptance of maskable interrupts can be selectively enabled and disabled by program using the interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). When two or more interrupts are generated simultaneously, the interrupt is accepted in the highest priority order as determined by the hardware. Figure 1.5.1 shows the interrupt controller.

(1) Interrupt Latches (IL₃₁ to IL₂)

Interrupt latches are provided for each source, except for a software interrupt. The latch is set to "1" when an interrupt request is generated, and requests the CPU to accept the interrupt. The latch is cleared to "0" just after the interrupt is accepted. All interrupt latches are initialized to "0" during reset.

The interrupt latches are assigned to addresses 0003CH, 0003DH, 0002EH and 0002FH in the SFR. Except for IL₂, each latch can be cleared to "0" individually by an instruction; however, the read-modify-write instruction such as bit manipulation or operation instructions cannot be used. When interrupt occurred during order execution, the reason is because interrupt request is cleared. Thus, interrupt requests can be canceled and initialized by the program. Note that request the interrupt latches cannot be set to "1" by an instruction. For example, it may be that each latch is cleared even if an interrupt request is generated during instruction.

The contents of interrupt latches can be read out by an instruction. Therefore, testing interrupt request by software is possible.

Example 1:	Clears interrupt latches		
DI		;	Disable interrupt
LDW	(ILL), 1110100000111111B	;	IL ₁₂ , IL ₁₀ to IL ₆ $\leftarrow 0$
Example 2:	Reads interrupt latches		
LD	WA, (ILL)	;	$W \leftarrow IL_{H}, A \leftarrow IL_{L}$
Example 3:	Tests an interrupt latch		
TEST	(ILL). 7	;	if $IL7 = 1$ then jump
$_{\rm JR}$	F, SSET		

(2) Interrupt Enable Register (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the pseudo non-maskable interrupts (software and watchdog timer interrupts). Pseudo non-maskable interrupts are accepted regardless of the contents of the EIR; however, the pseudo non-maskable interrupt cannot be nested more than once at the same time.

The EIR consists of an interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are assigned to addresses 0003AH, 0003BH, 0002CH and 0002DH in the SFR, and can be read and written by an instruction (including read-modify-write instruction such as bit manipulation instructions).

Note: Do not use the read-modify-write instruction for the EIRL (address 0003AH) during pseudo non-maskable interrupt service task. If the read-modify-write instruction is used, the IMF is not set to "1" after RETN.

1. Interrupt Master enable Flag (IMF)

The interrupt master enable flag (IMF) enables and disables the acceptance of all maskable interrupts. Clearing this flag to "0" disables the acceptance of all maskable interrupts. Setting to "1" enables the acceptance of interrupts.

When an interrupt is accepted, this flag is cleared to "0" to temporarily disable the acceptance of other maskable interrupts. After execution of the interrupt service program, this flag is set to "1" by the maskable interrupt return instruction [RETI] to again enable the acceptance of interrupts. If an interrupt request has already been occurred, interrupt service starts immediately after execution of the [RETI] instruction.

Pseudo non-maskable interrupts are returned by the [RETN] instruction. In this case, the IMF is set to "1" only when pseudo non-maskable interrupt service is started with interrupt acceptance enabled (IMF = 1). Note that the IMF remains "0" when cleared by the interrupt service program.

The IMF is assigned to bit 0 at address 0003AH in the SFR, and can be read and written by an instruction. The IMF is normally set and cleared by the [EI] and [DI] instructions, and the IMF is initialized to "0" during reset.

2. Individual interrupt Enable Flags (EF₁₆ to EF₃)

These flags enable and disable the acceptance of individual maskable interrupts, except for an external interrupt 0. Setting the corresponding bit of an individual interrupt enable flag to "1" enables acceptance of an interrupt, setting the bit to "0" disables acceptance.

Example 1: Sets EF for individual interrupt enable, and sets IMF to "1".

DI		;	Disable interrupt
LD	(EIRE), 00000001B	;	$\text{EF}_{16} \leftarrow 1$
LDW	(EIRL), 1110100010100001B		EF_{15} to EF_{13} , EF_{11} , EF_7 , EF_5 , $IMF \leftarrow 1$

Example 2: Sets an individual interrupt enable flag to "1". SET (EIRH). 4 ; $EF_{12} \leftarrow 1$

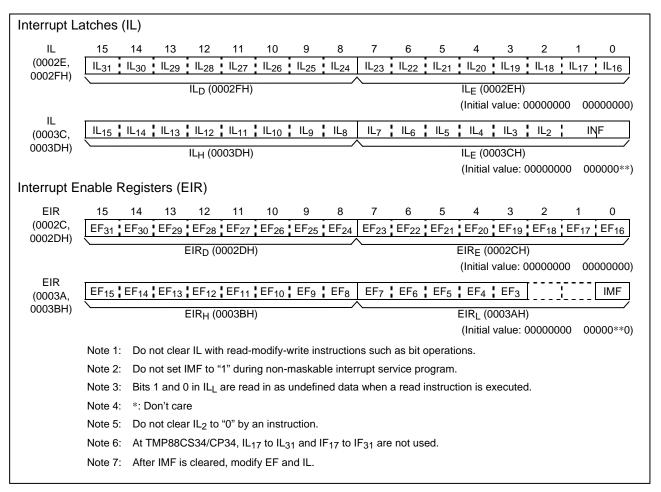


Figure 1.5.2 Interrupt Latches (IL) and Interrupt Enable Registers (EIR)

1.5.1 Interrupt Sequence

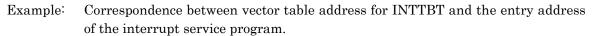
An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to "0" by a reset or an instruction. Interrupt acceptance sequence requires 12 machine cycles (3 μ s at fc = 16 MHz in the NORMAL mode) after the completion of the current instruction execution. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for pseudo non-maskable interrupts). Figure 1.5.3 shows the timing chart of interrupt acceptance processing.

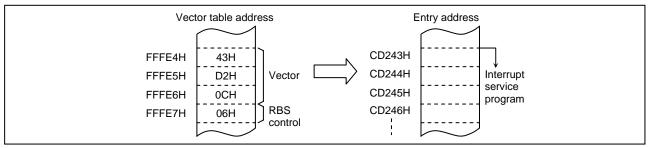
(1) Interrupt acceptance

Interrupt acceptance processing is as follows.

- 1. The interrupt master enable flag (IMF) is cleared to "0" to temporarily disable the acceptance of any following maskable interrupts. When a non-maskable interrupt is accepted, the acceptance of any following interrupts is temporarily disabled.
- 2. The interrupt latch (IL) for the interrupt source accepted is cleared to "0".
- 3. The contents of the program counter (PC) and the program status word (PSW) are saved (pushed) on the stack in sequence of PSW_H, PSW_L, PC_E, PC_H, PC_L. The stack pointer (SP) is decremented five times.
- 4. The entry address of the interrupt service program is read from the vector table, and set to the program counter.

- 5. The RBS control code is read from the vector table. The lower 4-bit of this code is added to the RBS.
- 6. The instruction stored at the entry address of the interrupt service program is executed.





A maskable interrupt is not accepted until the IMF is set to "1" even if the maskable interrupt higher than the level of current servicing interrupt is occurred.

When nested interrupt service is necessary, the IMF is set to "1" in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

Note: Do not use the read-modify-write instruction for the EIRL (address 0003AH) during pseudo non-maskable interrupt service task.

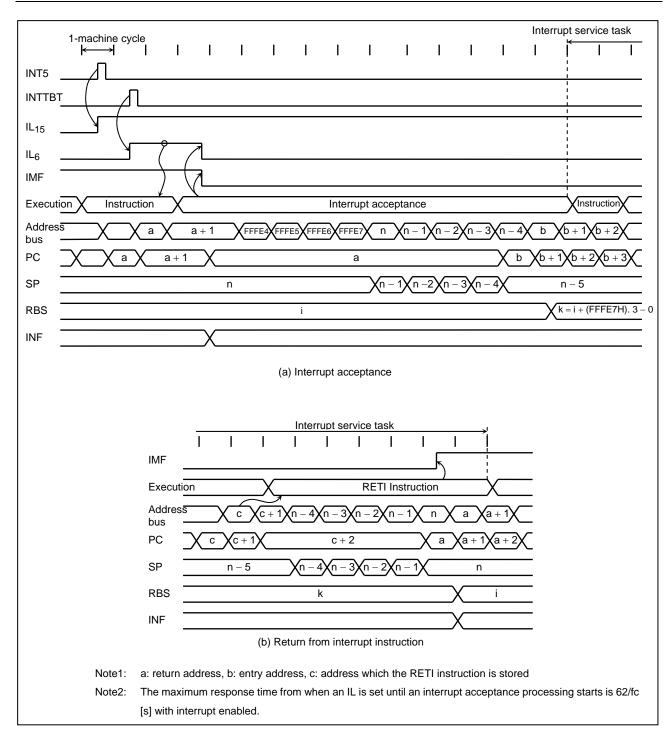


Figure 1.5.3 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

(2) Saving/Restoring general-purpose registers

During interrupt acceptance processing, the program counter (PC) and the program status word (PSW) are automatically saved on the stack, but not the accumulator and other registers. These registers are saved by the program if necessary. Also, when nesting multiple interrupt services, it is necessary to avoid using the same data memory area for saving registers.

The following method is used to save/restore the general-purpose registers.

1. General-purpose register save/restore by automatic register bank changeover

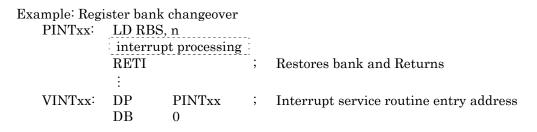
The general-purpose registers can be saved at high-speed by switching to a register bank that is not in use. Normally, the bank 0 is used for the main task and the banks 1 to 15 are assigned to interrupt service tasks. To increase the efficiency of data memory utilization, the same bank is assigned for interrupt sources which are not nested.

The switched bank is automatically restored by executing an interrupt return instruction [RETI] or [RETN]. Therefore, it is not necessary for a program to save the RBS.

Example: Register bank changeover PINTxx: interrupt processing RETI : VINTxx: DP PINTxx DB 1 ; RBS ← RBS + 1

2. General-purpose register save/restore by register bank changeover

The general-purpose registers can be saved at high-speed by switching to a register bank that is not in use. Normally, the bank 0 is used for the main tank and the banks 1 to 15 are assigned to interrupt service tasks.



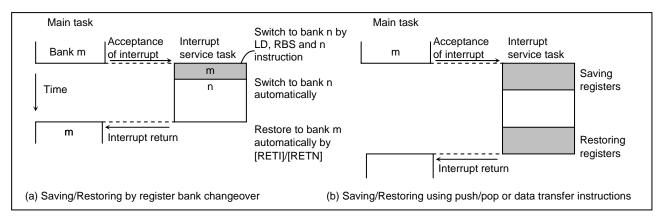
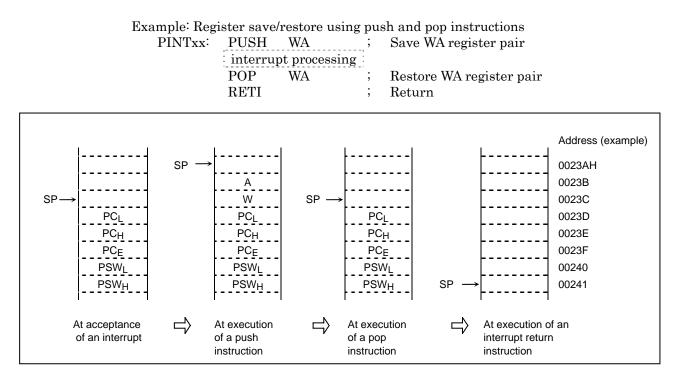


Figure 1.5.4 Saving/Restoring General-purpose Registers

3. General-purpose registers save/restore using push and pop instructions

To save only a specific register, and when the same interrupt source occurs more than once, the general-purpose registers can be saved/restored using the push/pop instructions.



4. General-purpose registers save/restore using data transfer instructions

Data transfer instruction can be used to save only a specific general-purpose register during processing of single interrupt.

Example: Saving/restoring a register using data transfer instructions PINTxx: LD (GSAVA), A ; Save A register

	(0011011),11	'	Davennegibten
interi	rupt processing		
LD	A, (GSAVA)	;	Restore A register
RETI		;	Return

(3) Interrupt return

The interrupt return instructions [RETI]/[RETN] perform the following	operations.
---	-------------

[RETI] Maskable interrupt return	[RETN] Non-maskable interrupt return
 The contents of the program counter and the program status word are restored from the stack. 	1. The contents of the program counter and program status word are restored from the stack.
2. The stack pointer is incremented 5 times.	2. The stack pointer is incremented 5 times.
3. The interrupt master enable flag is set to "1".	 The interrupt master enable flag is set to "1" only when a non-maskable interrupt is accepted in interrupt enable status. However, the interrupt master enable flag remains at "0" when so clear by an interrupt service program.
4. The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.	4. The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.

Interrupt requests are sampled during the final cycle of the instruction being executed. Thus, the next interrupt can be accepted immediately after the interrupt return instruction is executed.

Note: When the interrupt processing time is longer than the interrupt request generation time, the interrupt service task is performed but not the main task.

1.5.2 Software Interrupt (INTSW)

Executing the [SWI] instruction generates a software interrupt and immediately starts interrupt processing (INTSW is highest prioritized interrupt). However, if processing of a non-maskable interrupt is already underway, executing the SWI instruction will not generate a software interrupt but will result in the same operation as the [NOP] instruction.

Use the [SWI] instruction only for detection of the address error or for debugging.

1. Address error detection

FFH is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address. Code FFH is the SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing FFH to unused areas of the program memory. Address-trap reset is generated in case that an instruction is fetched from RAM, SFR or DBR areas.

2. Debugging

Debugging efficiency can be increased by placing the SWI instruction at the software break point setting address.

1.5.3 External Interrupts

The TMP88CS34/CP34 each have five external interrupt inputs ($\overline{INT0}$, INT2, INT3, INT4, and $\overline{INT5}$). Three of these are equipped with digital noise rejection circuits (pulse inputs of less than a certain time are eliminated as noise). Edge selection is also possible with INT2, INT3 and INT4.

The INTO /P50 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

Edge selection, noise rejection control except INT3 pin input and $\overline{\text{INT0}}$ /P50 pin function selection are performed by the external interrupt control register (EINTCR). Edge selecting and noise rejection control for INT3 pin input are preformed by the Remote control signal preprocessor control registers. (refer to the section of the Remote control signal preprocessor.) When INT0EN = 0, the IL3 will not be set even if the falling edge of $\overline{\text{INT0}}$ pin input is detected.

Source	Pin	Secondary function pin Enable conditions Edge		Edge	Digital noise rejection
INTO	ΪΝΤΟ	P50/TC2	IMF = 1, INT0EN = 1, EF ₃ = 1	Falling edge	Any pulse shorter than 2/fc [s] is regarded as noise and removed. Pulses not shorter than 7/fc [s] are definitely regarded as signals.
INT2	INT2	P53/TC1/ SCK1 /AIN0/ KWU0	IMF·EF ₇ = 1	Falling edge or Rising edge	Pulses of less than 7/fc [s] are eliminated as noise. Pulses equal to or more than 25/fc [s] are regarded as signals.
INT3	INT3	P30/RXIN	IMF∙EF ₁₁ = 1	Falling edge, Rising edge or Falling/Rising edge	Refer to the section of the Remote control preprocessor
INT4	INT4	P31/TC3	IMF • EF ₁₂ = 1	Falling edge or Rising edge	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals.
INT5	ĪNT5	P20/ STOP	IMF • EF ₁₅ = 1	Falling edge	Any pulse shorter than 2/fc [s] is regarded as noise and removed. Pulse not shorter than 7/fc [s] are definitely regarded as signals.

Note 1: The noise rejection function is also affected for timer/counter input (TC1 pin).

- Note 2: If a noiseless signal is input to the external interrupt pin in the NORMAL or IDLE mode, the maximum time from the edge of input signal until the IL is set is as follows:
 - (1) INT2, INT4 pin 31/fc [s]

(2) INT3 pin Refer to the section of the Remote control preprocessor.

Note 3: If a dual-function pin is used as an output port, changing data or switching between input and output generates a pseudo interrupt request signal. To ignore this signal, it is necessary to reset the interrupt enable flag.

Note 4: If INT0EN = "0", detecting the falling edge of the $\overline{INT0}$ pin input does not set the interrupt latch IL3.

EINTCR 7 (00037H) 0 		6 5 INTO EN -	4 INT4 ES	3 2 INT2 ES	1 "0"	0 _	(Initial value: 00*0	*00*)
INT	0EN	P50/INTO p	in configuration	0: P50 input/ou 1: INTO pin (P	• •		et to an input mode)	Write
	4ES 2ES	INT4 and IN	2 edge select 0: Rising edge 1: Falling edge		only			
Note	1: fc	: High-freque	ncy clock [Hz], *:	Don't care				
Note	2: E	dge detection	during switching	edge selection is	invalid			
Note	3: D	o not change	EINTCR only whe	en IMF = 1. After	changir	ng EINTCR,	interrupt latches of ex	ternal interrupt
	ir	puts must be	cleared to "0" usir	ng load instruction	n.			
Note	4: Ir	n order to cha	ange of external i	nterrupt input by	/ rewriti	ing the cont	ents of INT2ES and	INT4ES during
	N	IORMAL mode	e, clear interrupt la	atches of externa	l interru	upt inputs (IN	IT2 and INT4) after 8	machine cycles
	fr	om the time o	f rewriting.					
Note	5: Ir	In order to change an edge of timer counter input by rewriting the contents of INT2ES during NORMAL						uring NORMAL
	m	mode, rewrite the contents after timer counter is stopped (TC*s = 0), that is, interrupt disable state.						ble state.
	т	Then, clear a interrupt latch of external interrupt input (INT2) after 8 machine cycles from the time of						
	re	ewriting to cha	inge to interrupt e	nable state. Final	lly, start	t timer count	er.	
Exarr	iple: -	When changin	ng TC1 pin inputs LD (TC1CR), 01 DI LD (EINTCR), 00 NOP to NOP	001000B	; T ; IN	$C1S \leftarrow 00 (MF \leftarrow 0 (dist)$	from rising edge fallin stops TC1) ables interrupt service (change edge selectic	:)
	_	↓ ´	LD (ILL), 011111 El	11B	'	· · ·	rs interrupt latch) ables interrupt service)
			LD (TC1CR), 01	111000B	-	$C1S \leftarrow 11$ (ena)

Figure 1.5.5 External Interrupt Control Register

1.6 Reset Circuit

The TMP88CS34/CP34 has four types of reset generation procedures: an external reset input, an address trap reset output, a watchdog timer reset output and a system clock reset output. Table 1.6.1 shows on-chip hardware initialization by reset action.

The malfunction reset output circuit such as watchdog timer reset, address trap reset and system clock reset is not initialized when power is turned on. The $\overline{\text{RESET}}$ pin can output level "L" at the maximum 24/fc [s] (1.5 µs at 16 MHz) when power is turned on.

On-chip hardware		Initial value	On-chip hardware	Initial value	
Program counter	(PC)	(FFFFEH to FFFFCH)			
Stack pointer	(SP)	not initialized	Prescaler and Divider of timing	0	
General-purpose registers (W, A, B, C, D, E, H, L)		not initialized	generator		
Register bank selector	(RBS)	0	- Watchdog timer	Enable	
Jump status flag	(JF)	1		Ellable	
Zero flag	(ZF)	Not initialized			
Carry flag	(CF)	Not initialized			
Half carry flag	(HF)	Not initialized	Quitaut latabas of I/O parts	Refer to I/O port	
Sign flag	(SF)	Not initialized	Output latches of I/O ports	circuitry	
Overflow flag	(VF)	Not initialized			
Interrupt master enable flag	(IMF)	0			
Interrupt individual enable flags		0		Refer to each of	
	(EF)		Control registers	control register	
Interrupt latches	(IL)	0			
_		_	RAM	Not initialized	

Table 4 C 4	Initializing Internal Otation by Depart Action	
Table 1.6.1	Initializing Internal Status by Reset Action	

1.6.1 External Reset Input

The RESET pin contains a Schmitt trigger (hysteresis) with an internal pull-up resistor. When the RESET pin is held at "L" level for at least 3 machine cycles (12/fc [s]) with the power supply voltage within the operating voltage range and oscillation stable, a reset is applied and the internal state is initialized.

When the $\overline{\text{RESET}}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses FFFFCH to FFFFEH.

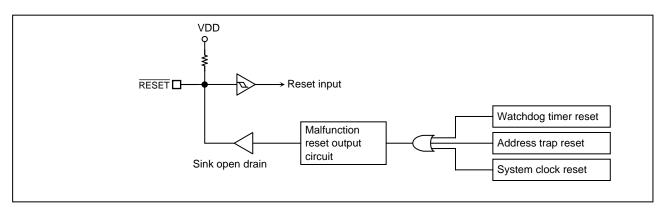


Figure 1.6.1 Reset Circuit

1.6.2 Address-Trap-Reset

If the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM, DBR or the SFR area, address-trap-reset will be generated. Then, the $\overline{\text{RESET}}$ pin output will go low. The reset time is about 8/fc to 24/fc [s] (0.5 to 1.5 µs at 16 MHz).

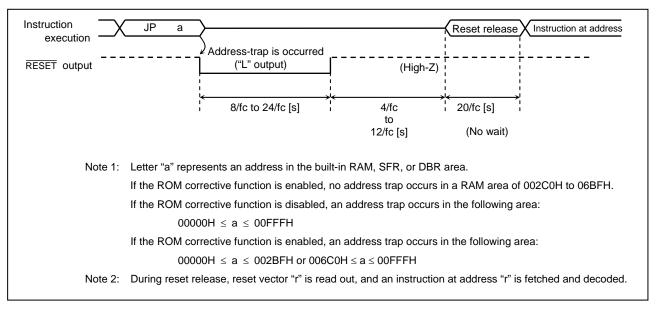


Figure 1.6.2 Address-Trap-Reset

1.6.3 Watchdog Timer Reset

Refer to Section "2.4 Watchdog Timer".

1.6.4 System-Clock-Reset

Clearing bits 7 in SYSCR2 to "0", system clock stops and causes the microcomputer to deadlock. This can be prevented by automatically generating a reset signal whenever bits 7, 6 and 5 in SYSCR2 = 000 is detected to continue the oscillation. The $\overline{\text{RESET}}$ pin output goes low from high-impedance. The reset time is about 8/fc to 24/fc [s] (0.5 to 1.5 µs at 16 MHz).

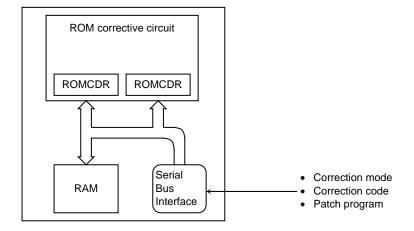
1.7 ROM Corrective Function

The ROM corrective function can patch the part (s) of on-chip ROM with some bugs.

The ROM corrective function have two modes. One is to replaced the instruction on a certain address in the ROM with the jump instruction to branch into the RAM area where the patched codes (Program Jump Mode). The other is to replace a byte or a word (2 or 3 bytes) length data in the ROM with the patched data (Data Replacement Mode). Four independent location can be patched.

- Note 1: When use ROM corrective circuit, it is necessary to contain a program which operates to load patched program and/or replacement data from external memory into an internal data RAM in an initial routine.
- Note 2: The address of an instruction for IDLE mode cannot be specificated as start address of corrective area.
- Note 3: The BM88CS34N0A-M15 does not support the ROM corrective circuit. Use the TMP88PS34 to debug a program of this circuit.

Example:



1.7.1 Configuration

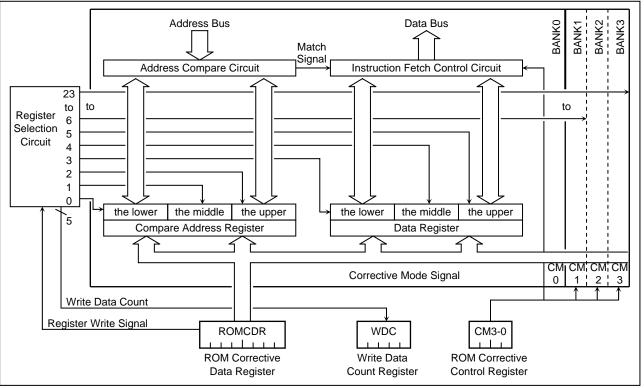


Figure 1.7.1 ROM Corrective Circuit

1.7.2 Control

The ROM corrective function is controlled by ROM corrective control register (ROMCCR) and ROM corrective data register (ROMCDR).

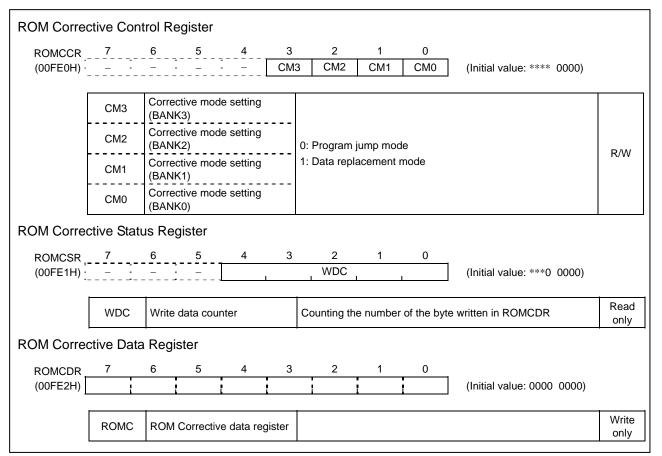


Figure 1.7.2 ROM Corrective Control Register, Status Register and ROM Corrective Data Register

(1) Enable and disable

The ROM corrective function is disabled after releasing reset. It is enabled after setting the data for one bank into ROMCDR. And the address-trap-reset is not generated when fetching an instruction from the RAM area except the address 02C0H to 06BFH.

After the ROM corrective function is enabled, it is neccesary to reset the micro controller in order to disable it.

(2) Data replacement mode

The ROM corrective function has the program jump mode and the data replacement mode.

By setting CMx (x: 0 to 3) in ROMCCR, the data replacement mode is selected.

(3) The ROM corrective data register writing

The ROM corrective data register has four banks corresponding to four independent locations to patch. The write data counter (WDC) points each bank set. (Figure 1.7.2)

BANK 0 The value of WDC after w 00001 The upper start address of the corrective area (8 bits) 00001 00011 The upper start address of the corrective area (8 bits) 00001 00011 The upper start address of the corrective area (4 bits) 00001 000011 The upper start address of the corrective area (4 bits) 00001 000011 The upper start address of the corrective area (4 bits) 00001 000011 The upper 4 bit of the jump address/replacement data 00100 00111 The upper 4 bit of the jump address/replacement data 00100 00111 The upper start address of the corrective area (8 bits) 00100 00111 The upper start address of the corrective area (8 bits) 01001 00111 The upper start address of the corrective area (8 bits) 01001 0111 The upper 4 bit of the jump address/replacement data 01000 0110 The upper start address of the corrective area (8 bits) 01110 0111 The upper start address of the corrective area (8 bits) 01101 0111 The upper start address of the corrective area (8 bits) 01101 0111 The upper 4 bit of the jump address/replacement data 10000 <th>0000 0000)</th>	0000 0000)
BANK 0The middle start address of the corrective area (8 bits)00010The upper start address of the corrective area (4 bits)00011The lower 8 bit of the jump address/replacement data00100The upper 4 bit of the jump address/replacement data00101The lower start address of the corrective area (8 bits)00111The widdle start address of the corrective area (8 bits)00110The upper 4 bit of the jump address/replacement data00100The upper 4 bit of the jump address/replacement data00101The upper 4 bit of the jump address/replacement data01001The upper start address of the corrective area (8 bits)01001The upper 4 bit of the jump address/replacement data01010The upper 4 bit of the jump address/replacement data01010The upper 4 bit of the jump address/replacement data01101The upper 4 bit of the jump address/replacement data01011The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10011The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10011	riting a data to ROMCE (Initial value)
BANK 0The upper start address of the corrective area (4 bits)000111The lower 8 bit of the jump address/replacement data00100The upper 4 bit of the jump address/replacement data00101The upper 4 bit of the jump address/replacement data00111The lower start address of the corrective area (8 bits)00111The upper start address of the corrective area (8 bits)00111The upper start address of the corrective area (8 bits)01001BANK 1The upper start address of the corrective area (4 bits)01001The upper start address of the corrective area (4 bits)01001The upper 4 bit of the jump address/replacement data01101The upper start address of the corrective area (8 bits)01111BANK 2The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data100101The lower start address of the corrective area (8 bits)10111The lower start address of the corrective area (8 bits)10111The upper 4 bit of the jump address/replacement data100101The upper 4 bit of the jump address/replacement data100101The upper start address of the corrective area (8 bits)10111The upper start address of the corrective area (8 bits)101101The upper st	←
BANK 0The lower 8 bit of the jump address/replacement data00100The lower 8 bit of the jump address/replacement data00101The upper 4 bit of the jump address/replacement data00101The upper 4 bit of the jump address/replacement data00110The lower start address of the corrective area (8 bits)00101BANK 1The upper start address of the corrective area (8 bits)01000The lower 8 bit of the jump address/replacement data01001The lower 8 bit of the jump address/replacement data01010The upper 4 bit of the jump address/replacement data01010The upper 4 bit of the jump address/replacement data01101The upper 4 bit of the jump address/replacement data01101The upper 4 bit of the jump address/replacement data01101The upper start address of the corrective area (8 bits)01101BANK 2The upper start address of the corrective area (8 bits)01111Dever 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data100101The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data100101The upper 4 bit of the jump address/replacement data100101The upper start address of the corrective area (8 bits)10101The upper start address of the correcti	
The lower 8 bit of the jump address/replacement data00100The middle 8 bit of the jump address/replacement data00101The upper 4 bit of the jump address/replacement data00101The lower start address of the corrective area (8 bits)00111The upper start address of the corrective area (8 bits)01000BANK 1The upper start address of the corrective area (4 bits)01001The lower 8 bit of the jump address/replacement data01010The upper 4 bit of the jump address/replacement data01010The upper 4 bit of the jump address/replacement data01010The lower start address of the corrective area (8 bits)01101The upper 4 bit of the jump address/replacement data01100The lower start address of the corrective area (8 bits)01101The lower 8 bit of the jump address/replacement data01101The upper 4 bit of the jump address/replacement data01001The upper start address of the corrective area (8 bits)01111The lower 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10010The idle start address of the corrective area (8 bits)10111The upper start address of the corrective area (8 bits)10011The upper 4 bit of the jump address/replacement data10010The upper start address of the corrective area (8 bits)10101The upper start address of the corrective area (8 bits)10101The upper start address of the corrective area (4 bit	
The upper 4 bit of the jump address/replacement data00110The lower start address of the corrective area (8 bits)00111The middle start address of the corrective area (8 bits)01000The upper start address of the corrective area (4 bits)01001The lower 8 bit of the jump address/replacement data01010The upper 4 bit of the jump address/replacement data01010The upper 4 bit of the jump address/replacement data01100The upper 4 bit of the jump address/replacement data01100The upper 4 bit of the jump address/replacement data01100The lower start address of the corrective area (8 bits)01110The inddle start address of the corrective area (8 bits)01111BANK 2The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data10000The inddle 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10010The inddle start address of the corrective area (8 bits)10011The indele start address of the corrective area (8 bits)10011The indele start address of the corrective area (4 bits)10100The upper start address of the corrective area (4 bits)10100The upper start address of the corrective area (4 bits)10101The upper start address of the corrective ar	
The lower start address of the corrective area (8 bits)00111The middle start address of the corrective area (8 bits)01000The upper start address of the corrective area (4 bits)01001The lower 8 bit of the jump address/replacement data01010The upper 4 bit of the jump address/replacement data01010The lower start address of the corrective area (8 bits)01111The upper 4 bit of the jump address/replacement data01010The indele start address of the corrective area (8 bits)01101The upper start address of the corrective area (8 bits)01111The upper start address of the corrective area (4 bits)01111The upper start address of the corrective area (4 bits)01111The upper 4 bit of the jump address/replacement data10000The middle 8 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10011The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10010The upper start address of the corrective area (8 bits)10110The upper start address of the corrective area (4 bits)10101The upper start address of the corrective area (4 bits)10101The upper start address of the corrective area (4 bits)10101The upper start address of the corrective area (4 bits)10101The upper 8 bit of the jump address/replacement data <td></td>	
BANK 1The middle start address of the corrective area (8 bits)01000The upper start address of the corrective area (4 bits)01001The lower 8 bit of the jump address/replacement data010101The upper 4 bit of the jump address/replacement data01011The lower start address of the corrective area (8 bits)01100The upper 4 bit of the jump address/replacement data01100The lower start address of the corrective area (8 bits)01110The upper start address of the corrective area (8 bits)01111The lower 8 bit of the jump address/replacement data010111The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10011The upper 4 bit of the jump address/replacement data10011The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10011The lower start address of the corrective area (8 bits)10111The upper 4 bit of the jump address/replacement data10010The upper start address of the corrective area (4 bits)10110The upper start address of the corrective area (4 bits)10101The upper start address of the corrective area (4 bits)10101The upper 8 bit of the jump address/replacement data10100The upper 8 bit of the jump address/replacement data10101The upper 8 bit of the jump address/replacement da	
BANK 1The upper start address of the corrective area (4 bits)01001The lower 8 bit of the jump address/replacement data01010The middle 8 bit of the jump address/replacement data01011The upper 4 bit of the jump address/replacement data01100The lower start address of the corrective area (8 bits)01101The upper start address of the corrective area (8 bits)01101The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data01101The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10011The lower start address of the corrective area (8 bits)10111The upper start address of the corrective area (8 bits)10111The upper start address of the corrective area (4 bits)10101The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10100The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101The lower 8 bit of the jump address/replacement d	
DAINK 1The lower 8 bit of the jump address/replacement data01010The middle 8 bit of the jump address/replacement data01011The upper 4 bit of the jump address/replacement data01011The upper 4 bit of the jump address/replacement data01100The lower start address of the corrective area (8 bits)01110The upper start address of the corrective area (8 bits)01111BANK 2The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10011The upper 4 bit of the jump address/replacement data10011The upper 4 bit of the jump address/replacement data10011The upper start address of the corrective area (8 bits)10110The lower start address of the corrective area (8 bits)10110The upper start address of the corrective area (8 bits)10101The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101The lower 8 bit of the jump address/replacement data10101The lower 8 bit of the jump address/replacement data10101The lower 8 bit of the jump address/rep	
The lower 8 bit of the jump address/replacement data01010The middle 8 bit of the jump address/replacement data01011The upper 4 bit of the jump address/replacement data01100The lower start address of the corrective area (8 bits)01110The upper start address of the corrective area (8 bits)01110The upper start address of the corrective area (4 bits)01111BANK 2The lower 8 bit of the jump address/replacement data10000The middle 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The lower start address of the corrective area (8 bits)10111The upper 4 bit of the jump address/replacement data10001The lower start address of the corrective area (8 bits)10011The upper 4 bit of the jump address/replacement data10011The lower start address of the corrective area (8 bits)10110The upper start address of the corrective area (8 bits)10101The upper start address of the corrective area (4 bits)10100The lower 8 bit of the jump address/replacement data10011The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10100The upper 8 bit of the jump address/replacement data10101The upper 8 bit of the jump address/replacement data </td <td></td>	
The initial of the jump address/replacement data01100The upper 4 bit of the jump address/replacement data01101The lower start address of the corrective area (8 bits)01111The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10010The lower start address of the corrective area (8 bits)10011The upper 4 bit of the jump address/replacement data10010The upper 4 bit of the jump address/replacement data10010The upper start address of the corrective area (8 bits)10110The upper start address of the corrective area (4 bits)10110The upper start address of the corrective area (4 bits)10100The upper start address of the corrective area (4 bits)10101The upper 8 bit of the jump address/replacement data10100The upper start address of the corrective area (4 bits)10101The upper 8 bit of the jump address/replacement data10100The upper 8 bit of the jump address/replacement data10100	
The upper start address of the corrective area (8 bits)01101The lower start address of the corrective area (8 bits)01110The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The lower start address of the corrective area (8 bits)10111The upper 4 bit of the jump address/replacement data10001The lower start address of the corrective area (8 bits)10011The upper 4 bit of the jump address/replacement data10010The upper start address of the corrective area (8 bits)10110The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101The lower 8 bit of the jump address/replacement data10110	
BANK 2The middle start address of the corrective area (8 bits)01110BANK 2The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10010The lower start address of the corrective area (8 bits)10111The niddle start address of the corrective area (8 bits)10011The upper start address of the corrective area (8 bits)10100The niddle start address of the corrective area (4 bits)10100The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10100	
BANK 2The upper start address of the corrective area (4 bits)01111The lower 8 bit of the jump address/replacement data10000The upper 4 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10010The lower start address of the corrective area (8 bits)10011The middle start address of the corrective area (8 bits)10100The upper start address of the corrective area (4 bits)10100The lower 8 bit of the jump address/replacement data10100The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101	
The lower 8 bit of the jump address/replacement data 10000 The middle 8 bit of the jump address/replacement data 10001 The upper 4 bit of the jump address/replacement data 10010 The lower start address of the corrective area (8 bits) 10011 The upper start address of the corrective area (8 bits) 10100 The upper start address of the corrective area (4 bits) 10101 The lower 8 bit of the jump address/replacement data 10101	
The middle 8 bit of the jump address/replacement data10001The upper 4 bit of the jump address/replacement data10010The lower start address of the corrective area (8 bits)10011The middle start address of the corrective area (8 bits)10100The upper start address of the corrective area (4 bits)10101The lower 8 bit of the jump address/replacement data10101	
The upper 4 bit of the jump address/replacement data 10010 The lower start address of the corrective area (8 bits) 10011 The middle start address of the corrective area (8 bits) 10100 BANK 3 The upper start address of the corrective area (4 bits) 10101 The lower 8 bit of the jump address/replacement data 10101	
BANK 3 The lower start address of the corrective area (8 bits) 10011 The upper start address of the corrective area (8 bits) 10100 The upper start address of the corrective area (4 bits) 10101 The lower 8 bit of the jump address/replacement data 10110	
BANK 3 The middle start address of the corrective area (8 bits) 10100 The upper start address of the corrective area (4 bits) 10101 The lower 8 bit of the jump address/replacement data 10110	
BANK 3 The upper start address of the corrective area (4 bits) 10101 The lower 8 bit of the jump address/replacement data 10110	
The lower 8 bit of the jump address/replacement data	
The lower 8 bit of the jump address/replacement data 10110	
The middle 8 bit of the jump address/replacement data 10111	
The upper 4 bit of the jump address/replacement data 00000	
Note 1: WDC value equals to the number of the byte stored in ROMCDR.	
Note 2: ROMCDR is set in order of the lower (8 bits), the middle (8 bits) and the upper (4 bit	s) start address of the

Figure 1.7.3 Banks and WDC Value of the Program Corrective Data Register

Whenever ROMCDR is written, WDC is incremented to indicate what data is writen via ROMCDR. During reset, WDC is intialized to "0".

- (1) The lower start address of the corrective area (8 bits)
- (2) The middle start address of the corrective area (8 bits)
- (3) The upper start address of the corrective area (4 bits)
- (4) The lower jump address/replacement data (8 bits)
- (5) The middle jump address/replacement data (8 bits)
- (6) The upper jump address (4 bits)/replacement data

Note 1: Corrective addresses must have over five addresses each other.

Note 2: The address of an instruction for IDLE mode cannot be specificated as start address of corrective area.

1.7.3 Functions

The ROM corrective function can correct maximum four ROM areas with their corresponding four banks of ROM corrective registers. Either program jump mode or data replacement mode is selected for each bank by CM0 to CM3 respectively.

(1) Program jump mode

In the program jump mode, the system executes a jump instruction when the program execution reaches the instruction at the corrective ROM address, skips from the instruction which would have been executed, and executes an instruction at a preset jump address.

Clearing ROMCCR CMx (x: 0 to 3) to "0" puts the system in the program jump mode. Use ROMCDR to set the corrective ROM address and jump address.

When the start address of an erroneous program is a corrective ROM address, and that of the patch program is a jump address, the bug in the erroneous program can be fixed. Note that the patch program should end with a jump instruction, which causes a return to the built-in ROM.

Note: For program jump mode, the address to be corrected must be the start address of the instruction.

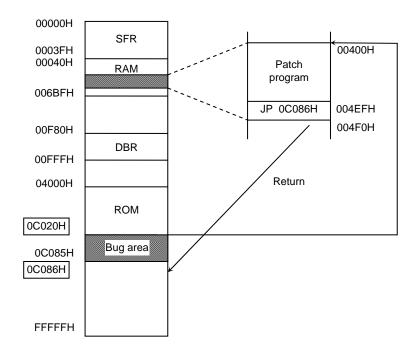
Example 1: Setting the Program Correction Circuit with the Initial Routine

Using the initial routine program, which is executed right after reset, set the program correction circuit's register and stores the patch program into the built-in RAM as follows.

- 1. Read the flag, which indicates whether to use the program correction circuit, from the external memory.
- 2. If that circuit is not used, perform normal initial processing.
- 3. If it is used, clear CMx to 0 to establish the program jump mode.
- 4. Read the corrective ROM address and jump address from the external memory.
- 5. Set the corrective ROM address and jump address, which were read in step 4., in ROMCDR.
- 6. Read the number of bytes for the patch program from the external memory.
- 7. Read the program with a number of bytes, equal to the byte count read in step 6., from the external memory, and store that program into the built-in RAM.
- 8. Repeat steps 4. through 7. as many times as there are required banks.

Example 2: There is bugs on the locations from $0\mathrm{C}020\mathrm{H}$ to $0\mathrm{C}085\mathrm{H}$

The corrective address, the jump vector, the program patch codes and other information to patch the ROM with the bugs must be read out from any of memory storage that holds them during initial program routine. CMn = 0 specifies the program jump mode. Subsequently, the patch program codes are loaded into RAM (00400H to 004EFH). The start address (0C020H) of the ROM necessary to patch is written to the corrective ROM address registers, and the start address (00400H) of the RAM area to patch is loaded onto the jump address registers. When the instruction at 0C020H is fetched, the instruction to jump into 00400H is unconditionally executed instead of the instruction at 0C020H, and the subsequent patch program codes are executed. The jump instruction at the end of the patch program codes returns to the ROM at 0C086H.



- Note: Corrective address must be assigned to 1st byte of instruction codes on the program jump mode.
- (2) Data replacement mode

In the data replacement mode, the system replaces reference data stored in the ROM area with the new instead of correcting the data reference instruction when that reference data is changed.

The program jump mode reduces the complexity of correcting the processing routine. However, when this mode is used, if there is a need to replace only the fixed data in ROM, the instruction to reference this ROM data should be corrected. Thus, a large amount of ROM is required for the patch program. To avoid this, the system has the data replacement mode. With this mode, three consecutive bytes of data can be replaced for each bank. (For an instruction which accesses only one byte, only the first byte can be replaced. For an instruction which accesses only two bytes, the two consecutive bytes can be replaced.) Setting ROMCCR CMx (x: 0 to 3) to "1" puts the system in the data replacement mode. Specify the start address of ROM data to be replaced as the corrective ROM address. Then, specify the new three-byte data as the patch data.

Note: For data replacement mode, the corrective address should be the address of fixed data (including a vector). (The operation code and operand cannot be changed.)

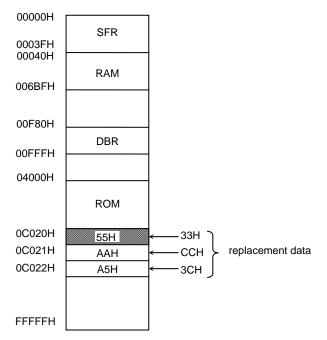
Example 1: Setting the Program Correction Circuit with the Initial Routine

Using the initial routine program, which is executed right after reset, set the program correction circuit's register as follows.

- 1. Read the flag, which indicates whether to use the program correction circuit, from the external memory.
- 2. If that circuit is not used, perform normal initial processing.
- 3. If it is used, set CMx to "1" to establish the data replacement mode.
- 4. Read the address of the data to be replaced and the patch data from the external memory.
- 5. Set the address and patch data, which were read in step 4., in ROMCDR.
- 6. Repeat steps 4. and 5. as many times as there are required banks.

Example 2: Replacing data 55H at 0C020H with 33H

Using the initial routine program, which is executed right after reset, read the start address of the data to be replaced and the patch data from the external memory. Set CMx (x: 0 to 3) to "1" to change the correction mode to the data replacement mode. Specify the start address (0C020H) of the data to be replaced as the corrective ROM address. Then, specify the new three-byte data (33H for 0C020H, CCH for 0C021H, and C3H for 0C022H) as the patch data.



1. At HL = 0C020H, Executing LD A, (HL) loads 33H in A. (Data replacement)

2. At HL = 0C021H, Executing LD A, (HL) loads AAH in A. (No data replacement)

3. At HL = 0C020H, Executing LD WA, (HL) loads CC33H in WA. (Data replacement)

4. At HL = 0C020H, Executing LD IX, (HL) loads CCC33H in IX. (Data replacement)

- Note 1: Corrective address must be assigned to constant data area on the data replacement mode. (Ope-code and Ope-rand cannot be replaced by ROM correction circuit.)
- Note 2: Instructions which includes "(HL +)" or "(– HL) " operation cannot be replaced by ROM corrective circuit on the data replacement mode.

2. On-Chip Peripheral Functions

2.1 Special Function Registers (SFR) and Data Buffer Registers (DBR)

The TLCS-870/X series uses the memory mapped I/O system and all peripheral control and data transfers are performed through the special function registers (SFR) and data buffer registers (DBR).

The SFR are mapped to addresses 00000H to 0003FH, and DBR are mapped to address 00F80H to 00FFFH.

Figure 2.1.1 shows the list of the TMP88CS34/CP34 SFRs and DBRs.

Rese Rese P2 p P3 p P4 p	rved	00020H 00021	SBISRA (SBI status A)	
Р2 р Р3 р Р4 р	Drt	00021		SBICRA (SBI control register A)
			SBIDBR (S	BI Data buffer)
P4 p	ort	00022	-	I ² CAR (I ² C Bus address)
		00023	SBISRB (SBI status B)	SBICRB (SBI control register B)
DF -	ort	00024	-	ORDMAL (OSD control)
P5 p	ort	00025		ORDMAH (OSD control)
Р6 р	ort	00026	RCSR (TC3 status)	RCCR (TC3 control)
Р7 р	ort	00027	PMPXCR	(Port control)
-	P5CR1 (P5 port I/O control1)	00028	-	PWMCR1A (PWM control1A)
-	P7CR (P7 port I/O control)	00029	-	PWMCR1B (PWM control1B)
Rese	rved	0002A	-	PWMDBR1 (PWMDBR1)
Rese	rved	0002B	-	P3CR1 (P3 I/O control)
-	P4CR (P4 port I/O control)	0002C	EIRE	-(Interrupt enable register)
-	P6CR (P6 port I/O control)	0002D	EIRD	(interrupt enable register)
ADCCRA (AD co	nverter control A)	0002E	ILE	_(Interrupt latch)
ADCCRB (AD co	nverter control B)	0002F	ILD	
TC1DRAL (Tir	ner register 1A)	00030	CGCR (D	vider control)
TC1DRAH		00031	ADCDR1 (AD	conversion result)
C1DRBL (Timor register 1B)	-	00032	ADCDR2 (AD	conversion result)
C1DRBH	-	00033	Re	served
TC1CR (T	C1 control)	00034	-	WDTCR1 (Watch-dog timer)
-	TC2CR (TC2 control)	00035	-	WDTCR2 control
-	TC2DRL (Timer register 2)	00036	TBTCR (TI	BT/TG control)
-	TC2DRH	00037	-	EINTCR (External interrupt control)
TC3DRA (Tim	er register 3A)	00038	SYSCR1	-(Svstem control)
C3DRB (Timer register 3B)	-	00039	SYSCR2	
-	TC3CR (TC3 control)	0003A	EIRL	
-	TC4DR (Timer register 4)	0003B	EIRH	-(Interrupt enable register)
-	TC4CR (TC4 control)	0003C	ILL	_(Interrupt latch)
ORDSN (O	SD control)	0003D	ILH	
ORCRAL (C	SD control)	0003E	PSWL	-(Program status word)
ORCRAH (C	OSD control)	0003F	PSWH	
	ADCCRA (AD con ADCCRA (AD con ADCCRB (AD con TC1DRAL TC1DRAL (Timer register 1B) - TC1DRBH TC1CR (T(TC3DRA (Timer C3DRB (Timer register 3B)	- P5CR1 (P5 port I/O control) - P7CR (P7 port I/O control) Reserved Reserved - P4CR (P4 port I/O control) - P4CR (P4 port I/O control) - P4CR (P6 port I/O control) - P6CR (P6 port I/O control) - P6CR (P6 port I/O control) - P6CR (P6 port I/O control) - ADCCRA (AD converter control A) - ADCCRB (AD converter control B) - TC1DRAL - .(Timer register 1A) - TC1DRAH C1DRBL - - TC1CR (TC1 control) - TC2CR (TC2 control) - TC2DRL - TC2DRL - TC3DRA (Timer register 2) - TC3DRA (Timer register 3A) C3DRB (Timer register 3B) - - TC3CR (TC3 control) - TC3CR (TC3 control) - TC4DR (Timer register 4)	- P5CR1 (P5 port I/O control) 00028 - P7CR (P7 port I/O control) 00029 Reserved 0002A - P4CR (P4 port I/O control) 0002B - P4CR (P4 port I/O control) 0002C - P6CR (P6 port I/O control) 0002D - P6CR (P6 port I/O control) 0002E ADCCRA (AD converter control A) 0002F - TC1DRAL 00030 TC1DRAH . 00031 C1DRBH . - 00033 C1DRBH . - 00035 - TC1CR (TC1 control) 00034 00035 - TC2DRL . (Timer register 2) 00036 - TC3DRA (Timer register 3A) 00038 00037 00037 C3DRB (Timer register 3B) - . 00038 . - TC4CR (TC3 control) 00038 . . 00038 C3DRB (Timer register 3B) - <	- P5CR1 (P5 port I/O control1) 00028 - - P7CR (P7 port I/O control) 00029 - Reserved 0002A - 0002B - - P4CR (P4 port I/O control) 0002C EIRE - - P4CR (P4 port I/O control) 0002D EIRE - - P6CR (P6 port I/O control) 0002F ILE - ADCCRA (AD converter control A) 0002F ILD - - ADCCRA (AD converter control B) 0002F ILD - - TC1DRAL .(Timer register 1A) 00031 ADCDR1 (AD O0032 ADCDR2 (AD - - - C1DRBL .(Timer register 1B) 00033 Reserved - C1DRBH TC1CR (TC1 control) 00035 -

Figure 2.1.1 (a) SFR

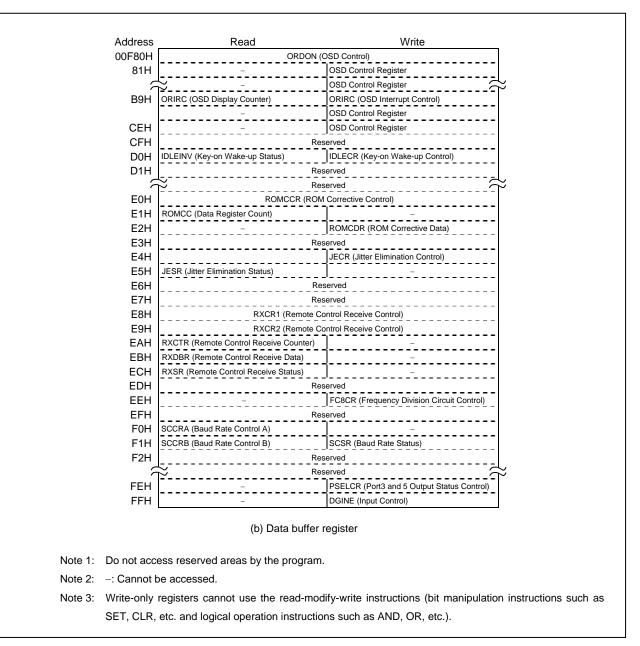


Figure 2.1.1 (b) DBR

2.2 I/O Ports

The TMP88CS34/CP34 has 6 parallel input/output ports (33 pins) as follows:

	Primary Function	Secondary Functions
Port P2	1-bit I/O port	External interrupt input, and STOP mode release signal input
Port P3	6-bit I/O port	External interrupt input, remote control signal input, data slicer analog input, timer/counter input, serial bus interface input/output
Port P4	8-bit I/O port	Pulse width modulation output
Port P5	8-bit I/O port	External interrupt input, timer/counter input, key-on wake-up input, serial bus interface input/output, analog input and I output from OSD circuitry.
Port P6	8-bit I/O port	R, G, B and Y/BL output from OSD circuitry, R.G.B and Y/BL input, analog input, and key-on wake-up input
Port P7	2-bit I/O port	Horizontal synchronous pulse input and vertical synchronous pulse input to OSD circuitry

Each output port contains a latch, which holds the output data. All input ports do not have latches, so the external input data should either be held externally until read or reading should be performed several times before processing. Figure 2.2.1 shows input/output timing examples.

External data is read from an I/O port in the S1 state of the read cycle during execution of the read instruction. This timing can not be recognized from outside, so that transient input such as chattering must be processed by the program. Output data changes in the S2 state of the write cycle during execution of the instruction which writes to an I/O port.

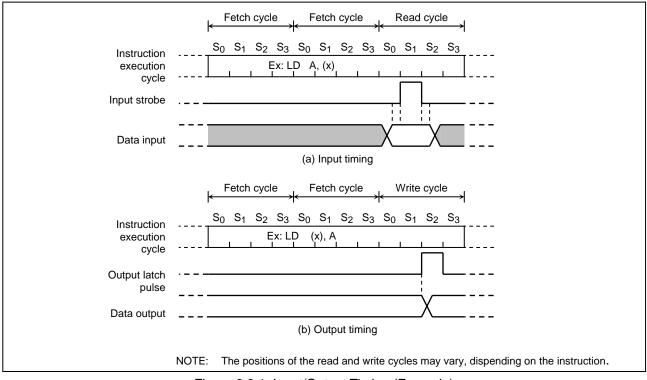


Figure 2.2.1 Input/Output Timing (Example)

When reading an I/O port except programmable I/O ports, whether the pin input data or the output latch contents are read depends on the instructions, as shown below:

- (1) Instructions that read the output latch contents
 - 1. XCH r, (src)
 - 2. SET/CLR/CPL (src).b
 - 3. SET/CLR/CPL (pp).g
 - 4. LD (src).b, CF
 - 5. LD (pp).b, CF
 - 6. ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), n
 - 7. (src) side of ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (HL)
- (2) Instructions that read the pin input data
 - 1. Instructions other than the above (1)
 - 2. (HL) side of ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (HL)

2.2.1 Port P2 (P20)

Port P2 is a 1bit input/output port. It is also used as an external interrupt input, and a STOP mode release signal input. When used as an input port, or a secondary function pin, the output latch should be set to "1". During reset, the output latch is initialized to "1".

It is recommended that pin P20 should be used as an external interrupt input, a STOP mode release signal input, or an input port. If used as an output port, the interrupt latch is set on the falling edge of the P20 output pulse.

When a read instruction for port P2 is executed, bits 7 to 1 in P2 are read in as undefined data.

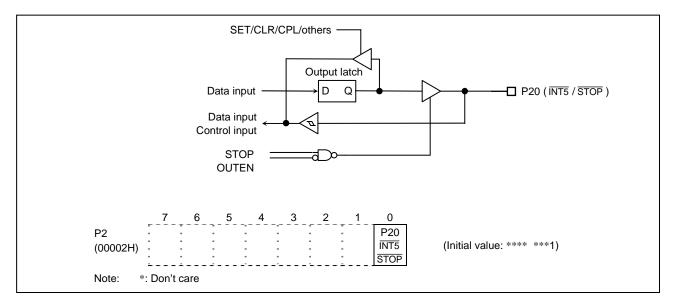


Figure 2.2.2 Port P2

2.2.2 Port P3 (P35 to P30)

Port P3 is an 6-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P3 input/output control register 1 (P3CR1). Port P3 is configured as an input if its corresponding P3CR1 bit is cleared to "0", and as an output if its corresponding P3CR1 bit is set to "1". During reset, P3CR1 is initialized to "0", which configures port P3 as an input. The P3 output latches are also initialized to "1". Data is written into the output latch regardless of the P3CR1 contents. Therefore initial output data should be written into the output latch before setting P3CR1.

Port P3 is also used as an external interrupt input, Remote-control signal input a timer/counter input, and serial bus interface input/output. When used as a secondary function input pin except I²C bus interface input/output, the input pins should be set to the input mode. When used as a secondary function output pin except I²C bus interface input/output, the output pins should be set to the output mode and beforehand the output latch should be set to "1". When P34 and P35 are used as I²C bus interface input/output, P3CR2 bits should be set to the sink open drain mode, the output latches should be set to "1", and the output pins should be set to the output mode.

Note: Input mode port is read the state of input pin. When input/output mode is used mixed, the contents of output latch setting input mode may be changed by executing bit manipulation instructions.

Example 1: Outputs an immediate data 5AH to port P3 LD (P3), 5AH ; P3 \leftarrow 5AH

 $\begin{array}{ccc} \text{Example 2: Inverts the output of the lower 4 bits (P33 to P30) in port P3} \\ \text{XOR} & (P3), 00001111B & ; & P33 to P30 \leftarrow \overline{P33} \text{ to } \overline{P30} \end{array}$

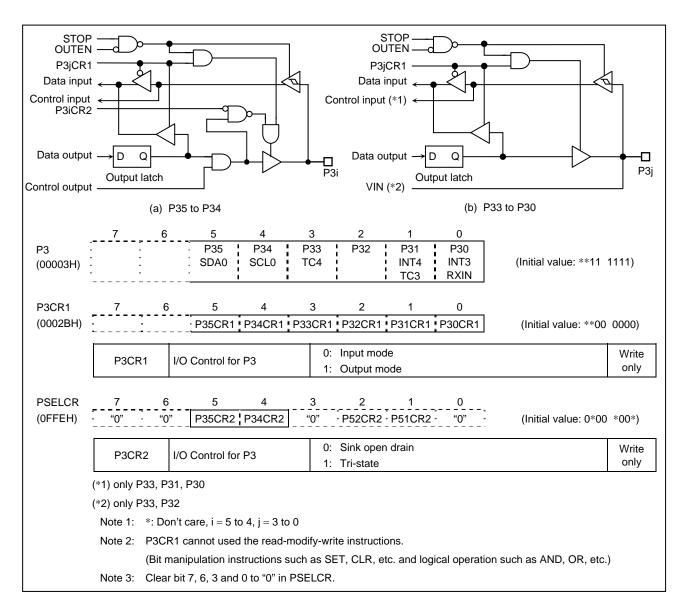


Figure 2.2.3 Port P3 and P3CR

2.2.3 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/Output mode is specified by the corresponding bit in the port P4 input/output control register (P4CR). Port P4 is configured as an input if its corresponding P4CR bit is cleared to "0", and as an output if its corresponding P4CR bit is set to "1". During reset, P4CR is initialized to "0", which configures port P4 as an input. The P4 output latches are also initialized to "1". Data is written into the output latch regardless of the P4CR contents. Therefore initial output data should be written into the output latch before setting P4CR.

Port P4 is also used as a pulse width modulation (PWM) output. When used as a PWM output pin, the output pins should be set to the output mode and beforehand the output latch should be set to "1".

Note: Input mode port is read the state of input pin. When input/output mode is used mixed, the contents of output latch setting input mode may be changed by executing bit manipulation instructions.

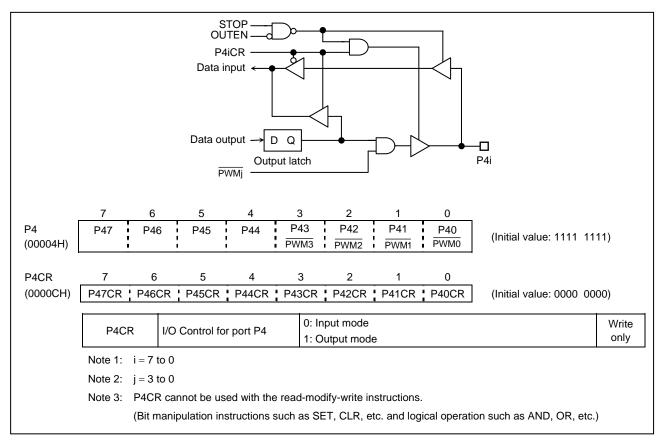


Figure 2.2.4 Ports P4 and P4CR

2.2.4 Port P5 (P57 to P50)

Port P5 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P5 input/output control register 1 (P5CR1). Port P5 is configured as an input if its corresponding P5CR1 bit is cleared to "0", and as an output if its corresponding P5CR1 bit is set to "1". During reset, P5CR1 is initialized to "0", which configures port P5 as an input. The P5 output latches are also initialized to "1". Data is written into the output latch regardless of the P5CR1 contents. Therefore initial output data should be written into the output latch before setting P5CR1.

Port P5 is also used as is also used as AD converter analog input, external interrupt input, timer/counter input, serial bus interface input/output, and an on screen display (OSD) output (I signal). When used as a secondary function input pin except I²C bus interface input/output, the input pins should be set to the input mode. When used as a secondary function output pin except I²C bus interface input/output, the output pin except I²C bus interface input/output, the output pin should be set to the output pins should be set to the output pins should be set to the output mode and beforehand the output latch should be set to "1". When P52 and P51 are used as I²C bus interface input/output, P5CR2 bits should be set to the sink open drain mode, the output latches should be set to "1", and the output pins should be set to the output mode. When P57 is used as an OSD output pin, the output pin should be set to the output mode and beforehand the port 6 data selection register (PIDS) should be clear to "0". When used as port P5, the port 6 data selection register (PIDS) should be set to "1".

Note: Input mode port is read the state of input pin. When input/output mode is used mixed, the contents of output latch setting input mode may be changed by executing bit manipulation instructions.

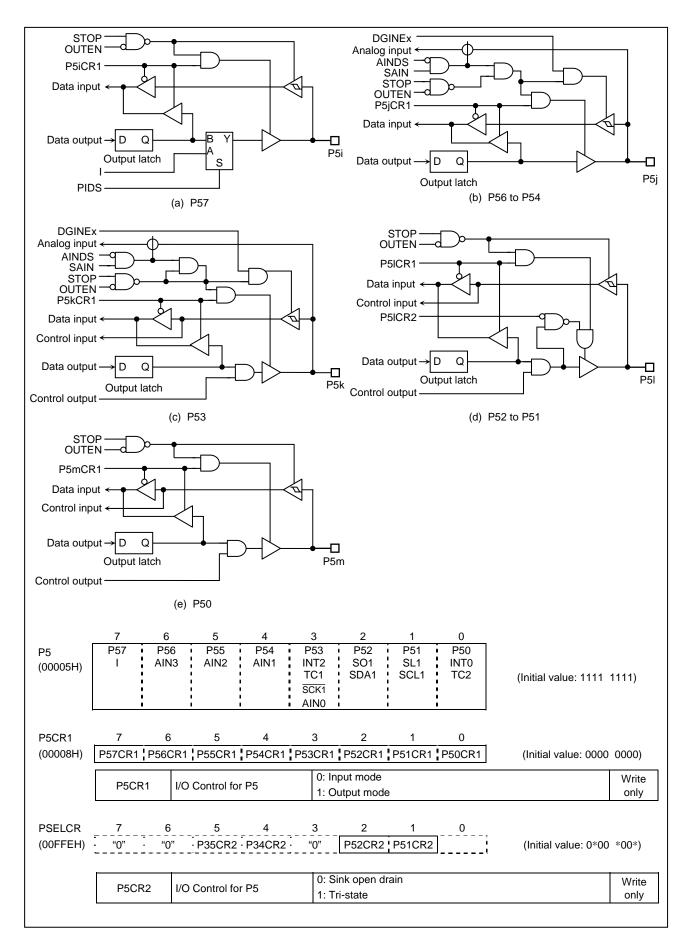


Figure 2.2.5 Ports P5 (1/2)

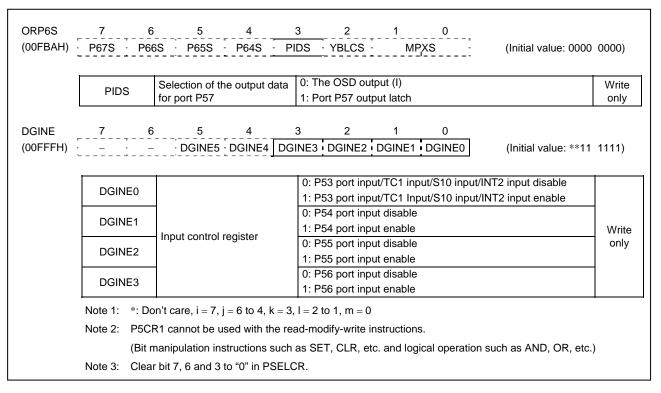


Figure 2.2.6 Ports P5 (2/2)

2.2.5 Port P6 (P67 to P60)

Port P6 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is selected by the corresponding bit in the port P6 input/output control register (P6CR). Port P6 is configured as an input if its corresponding P6CR bit is cleared to "0", and as an output if its corresponding P6CR bit is set to "1" and P6nS bit is set to "1". P63 to P60 are sink open drain ports. During reset, P6CR is initialized to "0", which configures port P6 as an input. The P6 output latches are also initialized to "1".

Data is written into the output latch regardless of the P6CR contents. Therefore initial output data should be written into the output latch before setting P6CR.

Port P6 is used as an on screen display (OSD) output (R, G, B, and Y/BL signal)/input (RIN, GIN BIN, Y/BLIN signal), a test video signal output and AD converter analog input. When used as a secondary function input, the input pins should be set to the input mode. When used as an OSD output pin, the output pins should be set to the output mode and beforehand the port P6 data selection register (P67S to P64S) should be clear to "0". When used as port P6, the signal control register (P67 to P64) should be set to "1".

- Note1: Input mode port is read the state of input pin. When input/output mode is used mixed, the contents of output latch setting input mode may be changed by executing bit manipulation instructions.
- Note2: P63 to P61 output "0" after a reset. When these dual-function pins are used as ports, be sure to set ORP6S2 to "1"
- Example: Sets the lower 4 bits (P63 to P60) in port P6 to the output mode, and the other bit to the input mode.

LD (P6CR), 0FH ; P6CR \leftarrow 00001111B

OUT P6i Data in Data out R, G, B, Y	P6jCR P6jCR Data input P6jCR Data input RIN, GIN (BL Output latch BS Output latch	P6j
P	6iS (a) P67 to P64 (b) P63 to P62	
DGINI Analog inp SAI SAI OUTE P6kC Data inp BIN, Y/BL Data outp		
	(c) P61 to P60	
P6 (00006H)	7 6 5 4 3 2 1 0 P67 P66 P65 P64 P63 P62 P61 P60 Y/BL B G R RIN GIN BIN Y/BLIN (Initial value: 1111 1)	111)
P6CR (0000DH)	7 6 5 4 3 2 1 0 P67CR P66CR P65CR P64CR P63CR P61CR P60CR (Initial value: 0000 (Initial va	0000)
	P6CR I/O Control for port P6 0: Input mode 1: Output mode	Write only
ORP6S (00FBAH) [7 6 5 4 3 2 1 0 P67S P66S P65S P64S PIDS YBLCS MPXS (Initial value: 0000 (Initial value)	0000)
ſ	P67S to P64S Selection of the output data for port P6i 0: The OSD output (R, G, B, Y/BL) 1: Port P6i output latch	Write only
DGINE (00FFFH)	7 6 5 4 3 2 1 0 DGINE5 DGINE4 DGINE3 DGINE2 DGINE1 DGINE0 (Initial value: **11	
	DGINE4 0: P60 port input/YIN/BLIN disable 1: P60 port input/YIN/BLIN enable 0: P61 port input/YIN/BLIN disable 1: P60 port input/YIN/BLIN enable 0: P61 port input/BIN disable 1: P61 port input/BIN disable 1: P61 port input/BIN disable	Write only
ORP6S2 (00FA1H)	7 6 5 4 3 2 1 0 - - - - Fixed at 1 Fixed at 1 Fixed at 1 - (Initial value: **** (Initial value: ****)))	000*)
	Be sure to fix these bits to "1", using the initial routine.	Write only
	 Note 1: *: Don't care, i = 7 to 4, j = 1 to 0 Note 2: P6CR and ORP6S cannot be used with the read-modify-write instructions. (Bit manipulations such as CLR, etc. and logical operation such as AND, OR, etc.) Note3: P63 to P61 output "0" after a reset. When these dual-function pins are used as port, be sure to set OF to "1". 	

Figure 2.2.7 Ports P6, P6CR, and P67S to P64S

2.2.6 Port P7 (P71 to P70)

Port P7 is a 2bit input/output port, and is also used as a vertical synchronous signal ($\overline{\text{VD}}$) input and a horizontal synchronous signal ($\overline{\text{HD}}$) input for the on screen display (OSD) circuitry.

The output latches, are initialized to "1" during reset. When used as an input port or a secondary function pin, the output latch should be set to "1".

When a read instruction for port P7 is executed, bits 7 to 2 in P7 are read in as undefined data.

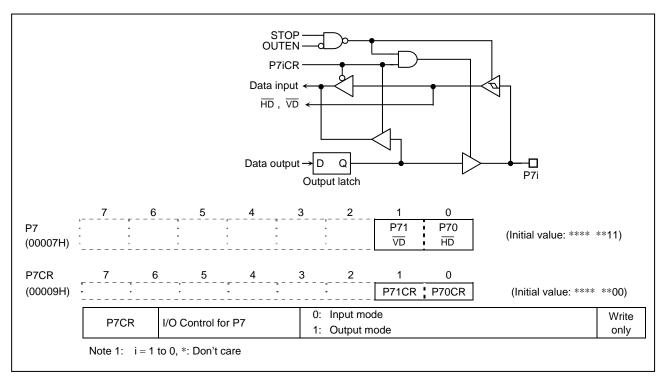


Figure 2.2.8 Ports P7

2.3 Time Base Timer (TBT)

The time base timer generates time base for key scanning, dynamic displaying, etc. It also provides a time base timer interrupt (INTTBT). The time base timer is controlled by a control register (TBTCR) shown in Figure 2.3.1.

An INTTBT is generated on the first rising edge of source clock (the divider output of the timing generator) after the time base timer has been enabled. The divider is not cleared by the program; therefore, only the first interrupt may be generated ahead of the set interrupt period.

The interrupt frequency (TBTCK) must be selected with the time base timer disabled (When the time base timer is changed from enabling to disabling, the interrupt frequency can't be changed.)

Both frequency selection and enabling can be performed simultaneously.

Example: Sets the time base timer frequency to fc/2¹⁶ [Hz] and enables an INTTBT interrupt.

L	D ((TBTCR), 00000010B	;	TB TCK="010"	
L	D ((TBTCR), 00001010B	;	TBTEN="1"	
S	ET (EIRL). 6			

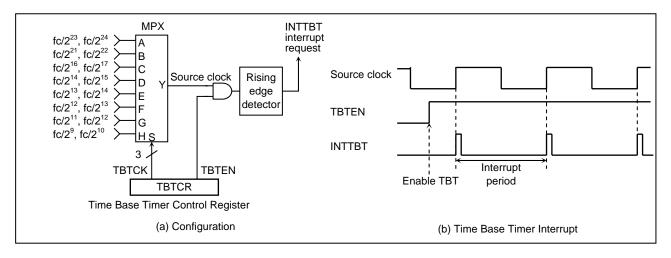


Figure 2.3.1 Time Base Timer

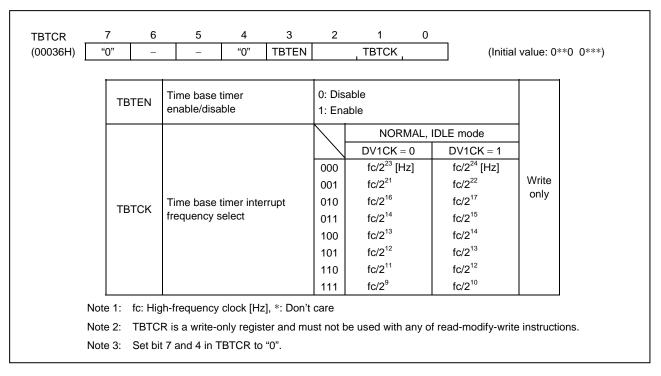


Figure 2.3.2 Time Base Timer and Divider Output Control Register

	Time Base Timer Interrupt Frequency [Hz]							
TBTCK	NORMAL, IDLE mode							
	DV1CK = 0	DV1CK = 1						
000	1.90	0.95						
001	7.62	3.81						
010	244.14	122.07						
011	976.56	488.28						
100	1953.12	976.56						
101	3906.25	1953.12						
110	7812.50	3906.25						
111	31250	15625						

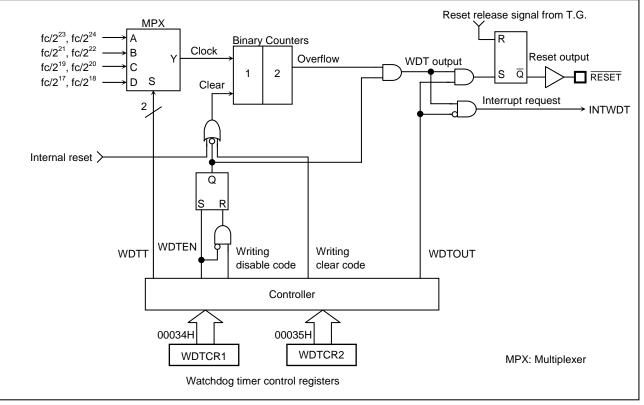
Table 2.3.1 Time Base Timer Interrupt Frequency (Example: at fc = 16MHz)

2.4 Watchdog Timer (WDT)

The watchdog timer is a fail-safe system to rapidly detect the CPU malfunctions such as endless looping caused by noise or the like, or deadlock and resume the CPU to the normal state.

The watchdog timer signal for detecting malfunction can be selected either a reset output or a pseudo non-maskable interrupt request. However, selection is possible only once after reset. At first the reset output is selected.

When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals.



2.4.1 Watchdog Timer Configuration

Figure 2.4.1 Watchdog Timer Configuration

2.4.2 Watchdog Timer Control

Figure 2.4.2 shows the watchdog timer control registers (WDTCR1, WDTCR2). The watchdog timer is automatically enabled after reset.

(1) Malfunction detection methods using the watchdog timer

The CPU malfunction is detected at follows.

- 1. Setting the detection time, selecting output, and clearing the binary counter.
- 2. Repeatedly clearing the binary counter within the setting detection time.
- Note: The watchdog timer consists of an internal divider and two-stage binary counter. Writing the clear code (4EH) clears the binary counter, but not the internal divider. The minimum overflow time for the binary counter might be three quarters of the WDTCR1 (WDTT) time setting depending on when the clear code (4EH) is written into the WDTCR2 register. So, write the clear code on a cycle which is shorter than that minimum overflow time.

If the CPU malfunctions such as endless looping or deadlock occur for any cause, the watchdog timer output will become active at the rising of an overflow from the binary counters unless the binary counters are cleared. At this time, when WDTOUT = 1 a reset is generated, which drivers the $\overline{\text{RESET}}$ pin low to reset the internal hardware and the external circuit. When WDTOUT = 0, a watchdog timer interrupt (INTWDT) is generated.

The watchdog timer temporarily stops counting in STOP mode including warm-up or IDLE mode, and automatically restarts (continues counting) when the STOP/IDLE mode is released.

Examp	ole: Sets t	the watchdog timer detect LD (WD	ion time to 2² DTCR2), 4EH						
					the binary counter				
		_	DTCR1), 0000		\leftarrow 10, WDTOUT \leftarrow				
	LD (WI			; Clears t	the binary counter	rs			
With	nin 3/4 of	WDT		(always	clear immediatel	y before			
detee	ction tim	ie !		and af	ter changing WD'	TT)			
Within 3/4 of WDT detection time			OTCR2), 4EH	; Clears t	the binary counter	rs			
		LD (WD	DTCR2), 4EH	; Clears t	the binary counter	rs			
Watchdog Tim	ner Register	' r 1							
WDTCR1	7	6 5 4	3 2						
(00034H)	·	· · · · · · · · · · · · · · · · · · ·	WDTEN	WPTT WDTOU	T (Initial value: ***	** 1001)			
	WDTEN	Watchdog timer enable/disable	0: Disable (It i WDTCR2) 1: Enable	s necessary to write the	disable code to				
		1		NORMA	L mode	1			
				DV1CK = 0	DV1CK = 1	1 <u> </u>			
	WDTT	Watchdog timer	00	2 ²⁵ /fc	2 ²⁶ /fc	Write only			
	VVDTT	detection time [s]	01	2 ²³ /fc	2 ²⁴ /fc	Unity			
			10	2 ²¹ /fc	2 ²² /fc				
			11	2 ¹⁹ /fc	2 ²⁰ /fc				
	WDTOUT	Watchdog timer output select	0: Interrupt requ 1: Reset output	est					
	Note 1: V	WDTOUT cannot be set to "1" by p		pring WDTOLIT to "0"					
		fc: High-frequency clock [Hz], *: Do							
		WDTCR1 is a write-only register ar		sed with any of read-mod	lifv-write instructions.				
		The watchdog timer must be disat				to the			
		STOP mode. When the counter is							
		the STOP mode.			-	-			
	Note 5:	Just right before disabling the wa	tchdog timer, dis	able the acceptance of	interrupts (DI) and cle	ear the			
	v	watchdog timer.							
	ŀ	If the watchdog timer is disabled u	under conditions	other than the above, th	e proper operation car	not be			
	ç	guaranteed.							
	oor Pogisto	~ 0							
Watchdog Tim	7	6 5 4	3 2	2 1 0					
WDTCR2					(Initial value: **	*** ****)			
(00035H)		<u> </u>	<u> </u>			,			
	WDTCR2	2 Watchdog timer control code write register	•	timer binary counter clea timer disable (disable co	· ,	Write only			
	Note 1: 1	The disable code is invalid unless v	written when WD	 TEN = 0.		,			
		*: Don't care							
		Clears the binary counter does not	-	• •					
		It is recommended that the time to							
	Note 5: 7	The watchdog timer counter must	be disabled by w	riting the disable code (E	31H) to WDRCR2 after	writing			
	V	te 5: The watchdog timer counter must be disabled by writing the disable code (B1H) to WDRCR2 after writing WDTCR2 to. "4EH".							

Figure 2.4.2 Watchdog Timer Control Registers

(2) Watchdog timer enable

The watchdog timer is enabled by setting WDTEN (bit 3 in WDTCR1) to "1". WDTEN is initialized to "1" during reset, so the watchdog timer operates immediately after reset is released.

Example: Disables watchdog timer LDW (WDTCR1), 00001000B ; WDTEN $\leftarrow 1$

(3) Watchdog timer disable

To disable the watchdog timer, clear the interrupt mask enable flag (IMF) to "0" and write the clear code (4EH) into WDTCR2. Then, clear WDTEN (bit 3 in WDTCR1) to "0".

When WDTEN is "0", the watchdog timer is disabled by writing the disable code (B1H) into WDTCR2. If WDTEN is cleared to "0" after the disable code has been written into WDTCR2, the watchdog timer is not disabled. While it is disabled, its binary counter is cleared.

Example:

DI		;	Disables interrupt acceptance.
LD	(WDTCR2), 4EH	;	Clears the watchdog timer.
LDW	(WDTCR1), B101H	;	Disables the watchdog timer.
EI		;	Enables interrupt acceptance.

Table 2.4.1 Watchdog Timer Detection Time (Example: fc = 16 MHz)

	Watchdog timer detection time [s]					
WDTT	NORMAL mode					
	DV1CK = 0	DV1CK = 1				
00	2.097	4.194				
01	524.288 m	1.048				
10	131.072 m	262.1 m				

2.4.3 Watchdog Timer Interrupt (INTWDT)

This is a pseudo non-maskable interrupt which can be accepted regardless of the contents of the EIR. If a watchdog timer interrupt or a software interrupt is already accepted, however, the new watchdog timer interrupt waits until the previous interrupt processing is completed (the end of the [RETN] instruction execution).

The stack pointer (SP) should be initialized before using the watchdog timer output as an interrupt source with WDTOUT.

Example: Wate	hdog timer interrupt setting	up	
LD	SP, 023FH	;	Sets the stack pointer
LD	(WDTCR1), 00001000B	;	WDTOUT $\leftarrow 0$

2.4.4 Watchdog Timer Reset

If the watchdog timer output becomes active, a reset is generated, which drivers the $\overline{\text{RESET}}$ pin (sink open drain input/output with pull-up) low to reset the internal hardware. The reset output time is about 8/fc to 24/fc [s] (0.5 to 1.5 µs at fc = 16.0 MHz).

Note: If there is any fluctuation in the oscillation frequency at the start of clock oscillation, the reset time includes error. Thus, regard the reset time as an approximate value.

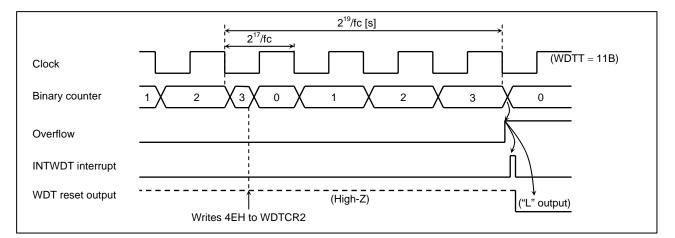
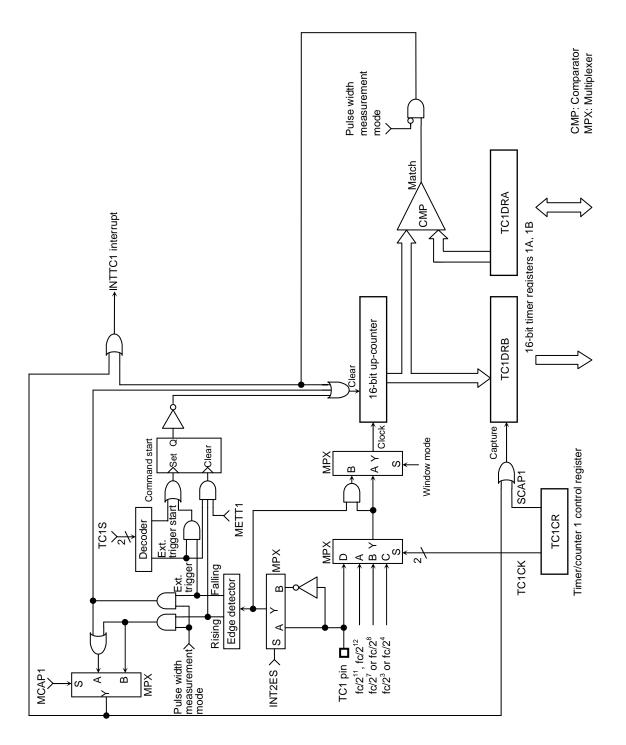


Figure 2.4.3 Watchdog Timer Interrupt/Reset

2.5 16-Bit Timer/Counter1 (TC1A)

2.5.1 Configuration



Note: Be sure to set the function of input/output pins correctly. For details, see the section on I/O port control registers.

Figure 2.5.1 Timer/Counter 1

2.5.2 Control

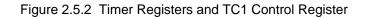
The timer/counter 1 is controlled by a timer/counter 1 control register (TC1CR) and two 16-bit timer registers (TC1DRA and TC1DRB).

rc1dra	1	5 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00010, 000	011H)		TC1	DRAĻ	(00011	1H)	:	1			TC1	ORAL (00010	H)		
FC1DRB	. <u> </u>										Read/W	rite				
00012, 000	013H)		TC1	DRBH	(0001	BH)	. <u> </u>	1			TC1	ORBL (00012	H)		
												Read	l only			
	7	7 6	5	4	3	2	1	0	-							
TC1CF	"(ACPAP1)" MCAP1	TC1	s	TC1	СК	тс	1M		Read/V						
(00014	H)	METT1		-				1		(Initial)	alue: 0	00 00	00)			
							1.1.2									
						xterna v mode		timer/	event co	ounter n	node					
TC1M	TC1 oper	rating mode	e select	-			easure	ment m	node							
				11: R	leserv	ed										
				\backslash						AL, IDL						
				$ \setminus$					DV7CK	= 0, DV	CK = 0C					
TOTOK	TOL				<u> </u>		DV1C					DV1C				
TC1CK	TC1 sour	ce clock se	iect [HZ]	00			fc/2	_					2 ¹²			
				01			fc/: fc/:					fc/				
				10 11			TC/				4	fc/	2			R/W
								EXI	emai cio	ock (TC	ner Exten	1	Window	Pulse	PPG	
				00: St	top an	d coun	iter clea	r		(0	0	0	0	
TC1S	TC1 star	t control		01: C	omma	nd star	rt			(x	×	×	×	×	
									sing ed	-	< 0	0	0	0	0	
ACAP1	Auto con	ture control	1	1		ture dis		at the fa	alling ed	lge : b-captur		0	0	0	0	
-	· ·	dth measure		0. Au	io-cap		sable		T. Auto	-captur	e enable	5				
MCAP1	mode co		ement	0: Do	0: Double edge capture 1: Single edge capture											
METT1	External mode cor	trigger time ntrol	er	0: Trigger start 1: Trigger start and stop												
	Note 1:	fc: High-f	requency	v clock [[Hz]											
	Note 2:	The timer	· register	consist	s of tw	vo shift	registe	rs. A va	alue set	in the ti	mer reg	ister is	put in	effect	at the	rising
		edge of th	-				-				-					-
		lower byt	e must b	e writte	en befo	ore the	upper	byte (i	t is reco	ommeno	ded that	a 16-t	oit acco	ess in	structio	on be
		used in w						•								
		effect.	3/		,				-,				,		- 3.0	
	Note 2.	Set the m	nda sou	irce clo	ck PD/	G cont	rol and	timer -	/F contr	rol wher		one (Ti	219 -	00)		
	NOLE 3.	Sectient	1000, 500				or and		/ Conti	or when		ops (It	-13 =	00).		

- Note 4: Auto-capture can be used in only timer, event counter, and window modes.
- Note 5: Values to be loaded to timer registers must satisfy the following condition.

```
TC1DRA > TC1DRB, TC1DRA > 1
```

- Note 6: Always write "0" to TFF1 except PPG output mode.
- Note 7: On entering STOP mode, the TC1 start control (TC1S) is cleared to "00" automatically. So, the timer stops. Once the STOP mode has been released, to start using the timer counter, set TC1S again.
- Note 8: In the Auto-capture function, when the capture value is read after stop and clear counter or Auto-capture disable is executed by the TC1 start control (TC1S), the correct capture value might not be able to be read.When using Auto-capture function, set capture to enable.



2.5.3 Function

Timer/counter 1 has five operating modes: timer, external trigger timer, event counter, window, pulse width measurement.

(1) Timer mode

In this mode, counting up is performed using the internal clock. The contents of TC1DRA are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to "0". Counting up resumes after the counter is cleared. The current contents of up-counter can be transferred to TC1DRB by setting ACAP1 (bit 6 in TC1CR) to "1" (software capture function). (Auto-capture function)

Table 2.5.1	Source Clock	(internal clock)) for Timer/Counter 1	(Example: at fc = 16.0 MHz)
			,	(

		NORMAL, IDLE mode						
TC1CK	DV1Cł	K = 0	DV1CK = 1					
	Resolution [µs]	Maximum time setting [s]	Resolution [µs]	Maximum time setting [s]				
00	128.0	8.39	256.0	16.78				
01	8.0	0.524	16.0	1.049				
10	0.5	32.77 m	1.0	65.54 m				

Example 1: Sets the timer mode with source clock fc/2¹¹ [Hz] and generates an interrupt 1 later (at fc = 16 MHz)

	LDW	(TC1DRA), 1E84H	;	Sets the timer register $(1 \text{ s} \div 2^{11}/\text{fc} = 1\text{E}84\text{H})$
	DI			
	SET	(EIRL). 4	;	Enable INTTC1
	\mathbf{EI}			
	LD	(TC1CR), 0000000B	;	Selects the source clock and mode
	LD	(TC1CR), 00010000B	;	Starts TC1
Example 2: A	uto-captur	e		
	LD	(TC1CR) 01010000B	:	$ACAP1 \leftarrow 1$ (Canture)

LD	(TC1CR), 01010000B	;	$ACAP1 \leftarrow 1 (Capture)$
LD	WA, (TC1DRB)	;	Reads the capture value

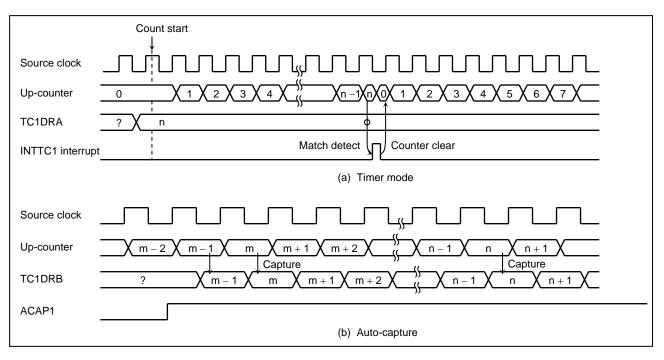


Figure 2.5.3 Timer Mode Timing Chart

(2) External trigger timer mode

LD

In this mode, counting up is started by an external trigger. This trigger is the edge of the TC1 pin input. Either the rising or falling edge can be selected with TC1S. Source clock is an internal clock. The contents of TC1DRA is compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to "0" and halted. The counter is restarted by the selected edge of the TC1 pin input.

When METT1 (bit 6 in TC1CR) is "1", inputting the edge to the reverse direction of the trigger edge to start counting clears the counter, and the counter is stopped. Inputting a constant pulse width can generate interrupts. When METT1 is "0", the reverse directive edge input is ignored. The TC1 pin input edge before a match detection is also ignored.

The TC1 pin input has the noise rejection; therefore, pulses of 7/fc [s] or less are rejected as noise. A pulse width of 13/fc [s] or more is required for edge detection in NORMAL or IDLE mode.

Example 1: Detects rising edge in TC1 pin input and generates an interrupt 100 μ s later.

	at fc = 16.	0 MHz, DV1CK = 1)		
	LDW	(TC1DRA), 0064H	;	$100 \ \mu s \div 2^4/fc = 64H$
	DI			
	SET	(EIRL). 4	;	INTTC1 interrupt enable
	EI			
	LD	(TC1CR), 00001000B	;	Selects the source clock and mode
	LD	(TC1CR), 00101000B	;	TC1 external trigger start, $METT1 = 0$
	a .	• • • • • • • • •	- • • •	
Evennle 2	Concretes			
				evel pulse (pulse width: 4 ms or more) to
		in. (at fc = 16.0 MHz, DV)		
			1CK	= 1)
	the TC1 p	in. (at fc = 16.0 MHz, DV	1CK	= 1)
	the TC1 p LDW	in. (at fc = 16.0 MHz, DV	1CK	= 1)
	the TC1 p LDW DI	in. (at fc = 16.0 MHz, DV) (TC1DRA), 00FAH	1CK	$4 \text{ ms} \div 2^8/\text{fc} = \text{FAH}$

(TC1CR), 01110100B ; TC1 external trigger start, METT1 = 1

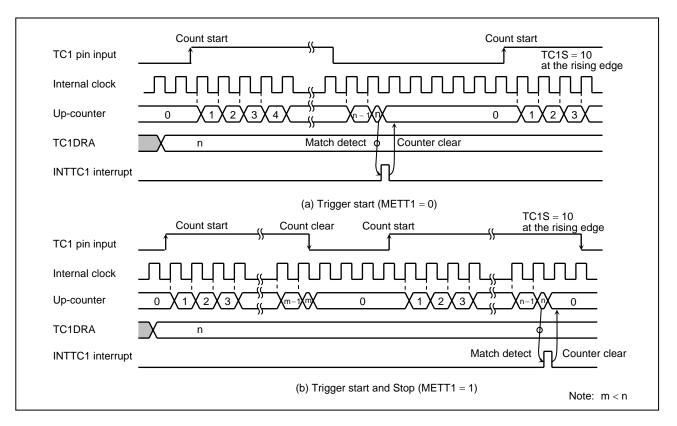


Figure 2.5.4 External Trigger Timer Mode Timing Chart

(3) Event counter mode

In this mode, events are counted at the edge of the TC1 pin input and bit 4 or 5 in TC1CR. Either the rising or falling edge can be selected with the external trigger. The contents of TC1DRA are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared.

Match detect is executed on other edge of count-up. A match can not be detected and INTTC1 is not generated when the pulse is still in same state.

Setting ACAP1 to "1" transfers the current contents of up-counter to TC1DRB (Auto-capture function).

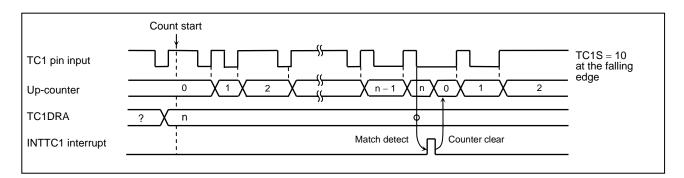


Figure 2.5.5 Event Counter Mode Timing Chart

NORMAL/IDLE	n [s]
"H" width 2 ³ /fc	
"L" width 2 ³ /fc	

Table 2.5.2 Input Pulse Width for Timer/Counter 1

(4) Window mode

Counting up is performed on the rising edge of the pulse that is the logical AND-ed product of the TC1 pin input (window pulse) and an internal clock. The contents of TC1DRA are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Positive or negative logic for the TC1 pin input can be selected with bit4 or 5 in TC1CR.

It is necessary that the maximum applied frequency be such that the counter value can be analyzed by the program. That is; the frequency must be considerably slower than the selected internal clock.

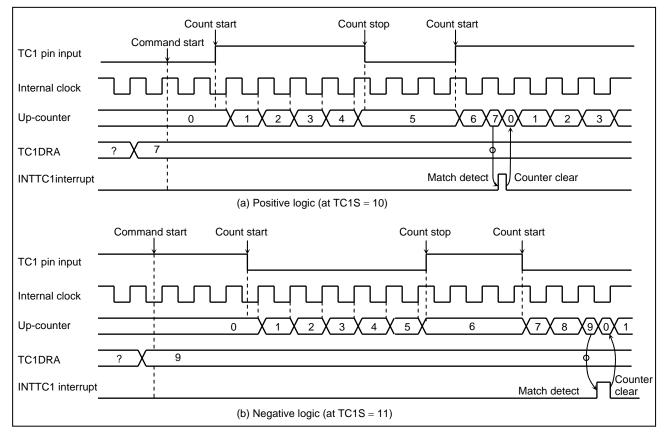


Figure 2.5.6 Window Mode Timing Chart

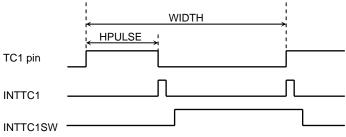
(5) Pulse width measurement mode

In this mode, counting is started by the external trigger (set to external trigger start by TC1CR). The trigger can be selected either the rising or falling edge of the TC1 pin input. The source clock is used an internal clock. On the next falling (rising) edge, the counter contents are transferred to TC1DRB and an INTTC1 interrupt is generated. The counter is cleared when the single edge capture mode is set. When double edge capture is set, the counter continues and, at the next rising (falling) edge, the counter contents are again transferred to TC1DRB. If a falling (rising) edge capture value is required, it is necessary to read out TC1DRB contents until a rising (falling) edge is detected. Falling or rising edge is selected with the external trigger TC1S (bit4 or 5 in TC1CR), and single edge or double edge is selected with MCAP1 (bit 6 in TC1CR).

- Note 1: Be sure to read the captured value from TC1DRB before the next trigger edge is detected. If fail to read it, it becomes undefined. It is recommended that a 16-bit access instruction be used to read from TC1DRB.
- Note 2: If either the falling or rising edge is used in capturing values, the counter stops at "1" after a value has been captured until the next edge is detected. So, the value captured next will become "1" larger than the value captured right after capturing starts.
- Note 3: In the Pulse width measurement mode, the capture value of the first time after the timer starts might not be a correct value. Thus, execute the dummy read once.

Example: Duty measurement (resolution $fc/2^7$ [Hz] DV1CK = 0)

Example. I	July Inc		[112]	D V 10 II = 0
	CLR	(INTTC1SW). 0	;	INTTC1 service switch initial setting: Clears Bit 0 of INTTC1SW. This bit is inverted by CPL instruction before INTTC1 is generated.
	LD DI	(TC1CR), 00000110B	;	Sets the TC1 mode and source clock
	SET EI	(EIRL). 4	;	Enables INTTC1
	LD :	(TC1CR), 00100110B	;	Starts TC1 with an external trigger at MCAP1 = 0
PINTTC1:	ĊPL JRS	(INTTC1SW). 0 F, SINTTC1	;	Complements INTTC1 service switch
	LD	WA, (TC1DRBL)	;	Reads TC1DRB ("H" level pulse width) Lower address in TC1DRBL: TC1DRB
	LD RETI	(HPULSE), WA		
SINTTC1:	LD LD :	WA, (TC1DRBL) (WIDTH), WA	;	Reads TC1DRB (Period)
	RETI :		;	Duty calculation
VINTTC1:	DW	PINTTC1	;	Sets INTTC1



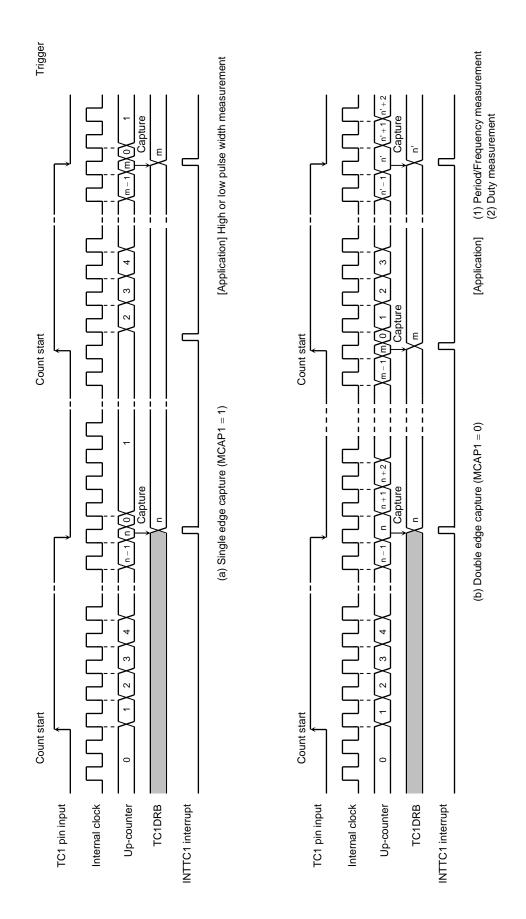


Figure 2.5.7 Pulse Width Measurement Mode Timing Chart

2.6 16-Bit Timer/Counter 2 (TC2A)

2.6.1 Configuration

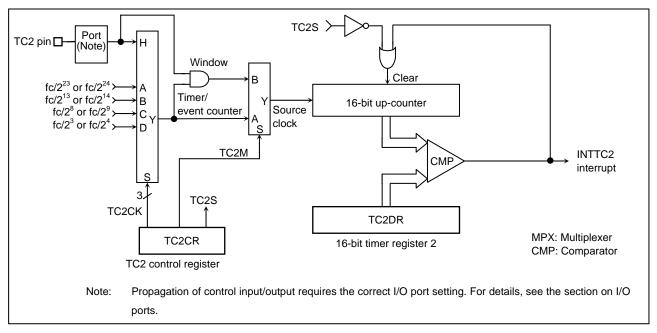


Figure 2.6.1 Timer/Counter 2 (TC2)

2.6.2 Control

The timer/counter 2 is controlled by a timer/counter 2 control register (TC2CR) and a 16-bit timer register 2 (TC2DR). Reset does not affect TC2DR.

1	5 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		_ тс	2DRH	(00017	7H)				1	ТС	2DRL (00016H	1)	1	
										R	ead/Wr	ite			
	6	5 TC2S	4	3 TC2Cł	2	1	0 TC2M]		(Initia	ıl value	: **00	00*0)		
TC2 operating	j mode se	elect	-			unter m	ode								
							NC	RMAL	1/2, IDI	_E1/2 m	ode				
				\searrow							DV1	CK = 1			
			00	0		fc/	2 ²³				fo	c/2 ²⁴			
			00	1											
TC2			01	0			-				-				Writ
source clock select [Hz]															onl
								-							
				-		Rese	erved				Res	served			
				-			-								
TOO								ernal c	lock (10	C2 pin ir	nput)				
	trol			•	ina cour	nter clea	ar								
Note 1:	fc: High-	frequen	cy cloci	k [Hz],	*: Don't	care									
Note 2:	Writing t	to the lo	wer by	te of ti	mer reg	jister 2	(TC2D	RL), th	e comp	arison i	is inhib	ited un	til the u	pper	byte
	(TC2DR	H) is w	ritten.	After w	riting to	the u	pper b	yte, an	y matc	h during	g 1 ma	achine	cycle (instru	ction
					•						-				
Note 3:	Set the r	node an	d sourd	ce cloci	k when t	the TC2	stops	(TC2S	= 0).						
Note 4:	Values te	o be loa [,]	ded to t	timer re	egister n	nust sat	isfy the	follow	ing con	dition.					
	TC2DR :	>1			0				U						
Note 5:	TC2CR	are write	e-only re	egister	s and m	ust not	be use	d with a	any of th	ne read-	-modifv	-write ir	nstructio	ons.	
-			,	5							,				
Note 6:	When S	TOP mo	de is s	tarted.	timer co	ounter is	s stopp	ed and	cleared	d. Set T	C2S to	"1" afte	er STO	P mod	de is
	TC2 operating TC2 source cl TC2 source cl TC2 start cont Note 1: Note 2: Note 3: Note 4:	TC2 source clock select TC2 source clock select TC2 source clock select TC2 start control Note 1: fc: High- Note 2: Writing to (TC2DR execution Note 3: Set the ri Note 4: Values to TC2DR =	7 6 5 TC2 TC2S operating mode select TC2 source clock select [Hz] TC2 source clock select [Hz] TC2 source clock select [Hz] TC2 start control Note 1: fc: High-frequence Note 2: Writing to the lo (TC2DRH) is will execution cycle) Note 3: Set the mode and Note 4: Values to be load TC2DR > 1	7 6 5 4 7 6 5 4 TC2S 0: T TC2 0: T 1: V operating mode select 1: V TC2 01 source clock select [Hz] 01 10 10 11 11 TC2 0: source clock select [Hz] 01 10 10 11 11 TC2 0: start control 1: Note 1: fc: High-frequency clock Note 2: Writing to the lower by (TC2DRH) is written/ execution cycle) is igno Values to be loaded to the component of the lower by Note 3: Set the mode and source Note 4: Values to be loaded to the component of the lower by TC2DR > 1 1	7 6 5 4 3 TC2 TC2S TC2CH TC2 0: Timer/e operating mode select 1: Window 000 001 TC2 010 source clock select [Hz] 010 100 101 110 111 TC2 0: Stop a start control 1: Start Note 1: fc: High-frequency clock [Hz], Note 2: Writing to the lower byte of ti (TC2DRH) is written. After weexecution cycle) is ignored. Note 3: Set the mode and source clock Note 4: Values to be loaded to timer represented by the order of the complexity of	TC2DRH (00017H) 7 6 5 4 3 2 TC2S TC2CK TC2CK TC2CK TC2 0: Timer/event co 0: Timer/event co operating mode select 1: Window mode TC2 0: Timer/event co source clock select [Hz] 010 100 111 100 101 110 111 TC2 0: Stop and cour start control 1: Start Note 1: fc: High-frequency clock [Hz], *: Don't Note 2: Writing to the lower byte of timer reg (TC2DRH) is written. After writing to execution cycle) is ignored. Note 3: Set the mode and source clock when Note 4: Values to be loaded to timer register r TC2DR > 1 1	TC2DRH (00017H) 7 6 5 4 3 2 1 TC2S TC2CK TC2CK TC2CK TC2CK TC2CK TC2 0: Timer/event counter m 1: Window mode DV10 000 fc/ TC2 0: 000 fc/ 001 fc/ 001 fc/ TC2 010 fc/ 011 fc/ 011 fc/ Source clock select [Hz] 011 fc 100 Resettee 110 111 fc TC2 0: Stop and counter cleatee 1: Start 110 fc 110 fc TC2 0: Stop and counter cleatee 1: Start 110 fc 110 fc Note 1: fc: High-frequency clock [Hz], *: Don't caree Note 2: Writing to the lower byte of timer register 2 (TC2DRH) is written. After writing to the uncescution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 Note 4: Values to be loaded to timer register must satt TC2DR > 1	TC2DRH (00017H) TC2 TC2 operating mode select 0: Timer/event counter mode 1: Window mode TC2 operating mode select 0: Timer/event counter mode 1: Window mode 000 fc/2 ²³ 001 fc/2 ²³ 001 fc/2 ³ 010 fc/2 ³ 010 fc/2 ³ 100 Reserved 111 Ext TC2 0: Stop and counter clear 1: Start Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DI (TC2DRH) is written. After writing to the upper by execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops Note 4: Values to be loaded to timer register must satisfy the TC2DR > 1	TC2DRH (00017H) TC2DRH (00017H) TC2 TC2S TC2CK TC2 operating mode select 1: Window mode DV1CK = 0 000 fc/2 ²³ 001 fc/2 ²³ 001 fc/2 ²³ 001 fc/2 ²³ 001 fc/2 ³ 001 fc/2 ³ 010 fc/2 ³ 100 Reserved 101 Reserved 110 111 External c 112 113 114 External c 115 116 117 118 119 110 111 External c 112 113 114 115 116 117 118 119	TC2 TC2DRH, (00017H) TC2 TC2S TC2 0: Timer/event counter mode operating mode select 0: Timer/event counter mode 1: Window mode NORMAL1/2, IDI DV1CK = 0 000 000 fc/2 ²³ 001 fc/2 ³ 001 fc/2 ⁸ 001 fc/2 ⁸ 010 fc/2 ⁸ 100 Reserved 101 Reserved 111 External clock (TC TC2 0: Stop and counter clear 11: Start Note 1: fc: High-frequency clock [Hz], *: Don't care Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DRL), the comp (TC2DRH) is written. After writing to the upper byte, any matc execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops (TC2S = 0). Note 4: Values to be loaded to timer register must satisfy the following con- TC2DR > 1	TÇ2DRH (00017H) TC R TC2 TC2S TC2CK Operating mode select 0: Timer/event counter mode 1: Window mode NORMAL1/2, IDLE1/2 m DV1CK = 0 000 000 fc/2 ²³ 001 fc/2 ¹³ 001 fc/2 ⁸ 001 fc/2 ⁸ 010 fc/2 ⁸ 011 fc/2 ³ 100 Reserved 111 External clock (TC2 pin in TC2 0: Stop and counter clear 111 External clock (TC2 pin in TC2 0: Stop and counter clear 111 External clock (TC2 pin in TC2 0: Stop and counter clear 111 External clock (TC2 pin in TC2 0: Stop and counter clear start control 1: Start Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DRL), the comparison in (TC2DRH) is written. After writing to the upper byte, any match during execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops (TC2S =	TC2DRH (00017H) TC2DRL (Read/Wi TC2 TC2S TC2CK TC2M TC2 0: Timer/event counter mode Initial value TC2 0: Timer/event counter mode 0 operating mode select 1: Window mode NORMAL1/2, IDLE1/2 mode TC2 0: Timer/event counter mode 0/1 000 fc/2²3 ft 001 fc/2²3 ft 001 fc/2³ ft 010 fc/2³ ft 100 Reserved Reserved 111 External clock (TC2 pin input) TC2 0: Stop and counter clear 111 External clock (TC2 pin input) TC2 0: Stop and counter clear 1: Start 0: Stop and counter clear Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DRL), the comparison is inhib (TC2DRH) is written. After writing to the upper byte, any match during 1 ma execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops (TC2S = 0). Note 4: Values to be loaded to timer register must satisfy the following condition. TC2DR > 1 <td>TC2DRH (00017H) TC2DRL (00016H) Read/Write TC2 TC2S TC2S TC2CK TC2 0: Timer/event counter mode 1: Window mode DV1CK = 0 DV1CK = 1 000 fc/2²³ 010 fc/2²³ 011 fc/2³ 100 Reserved 111 External clock (TC2 pin input) TC2 0: Stop and counter clear 1: Start 1: Start Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DRL), the comparison is inhibited uni (TC2DRH) is written. After writing to the upper byte, any match during 1 machine execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops (TC2S = 0). Note 4: Values to be loaded to timer register must satisfy the following condition. TC2DR > 1</td> <td>TÇ2DRH (00017H) TC2DRL (00016H) Read/Write 7 6 5 4 3 2 1 0 TC2 TC2S TC2CK TC2M (Initial value: **00 00*0) TC2 0: Timer/event counter mode 1: Window mode NORMAL1/2, IDLE1/2 mode TC2 0: Timer/event counter mode DV1CK = 0 DV1CK = 1 000 fc/2²³ fc/2²⁴ 01 fc/2¹³ fc/2⁴ 01 fc/2³ fc/2⁴ 01 fc/2³ fc/2⁴ 100 Reserved Reserved 111 External clock (TC2 pin input) TC2 0: Stop and counter clear 1: Start Start control 1: Start 0: Stop and counter clear 1: Start Note 1: fc: High-frequency clock [Hz], *: Don't care Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DRL), the comparison is inhibited until the u (TC2DRH) is written. After writing to the upper byte, any match during 1 machine cycle (i execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops (TC</td> <td>TÇ2DRH (00017H) TC2DRL (00016H) Read/Write TC2 TC2S TC2S TC2CK TC2 0: Timer/event counter mode operating mode select 0: Timer/event counter mode 1: Window mode NORMAL1/2, IDLE1/2 mode DV1CK = 0 DV1CK = 1 000 fc/2²⁴ 001 fc/2³ 1: Window mode fc/2⁴⁴ 000 fc/2³ 1: Window mode fc/2⁴⁴ 000 fc/2⁴⁵ 001 fc/2⁴⁵ 001 fc/2⁴⁵ 100 Reserved 101 Reserved 101 Reserved 111 External clock (TC2 pin input) TC2 0: Stop and counter clear 1: Start 10 Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DRL), the comparison is inhibited until the upper (TC2DRH) is written. After writing to the upper byte, any match during 1 machine cycle (instrue execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops (TC2S = 0). Note 4</td>	TC2DRH (00017H) TC2DRL (00016H) Read/Write TC2 TC2S TC2S TC2CK TC2 0: Timer/event counter mode 1: Window mode DV1CK = 0 DV1CK = 1 000 fc/2 ²³ 010 fc/2 ²³ 011 fc/2 ³ 100 Reserved 111 External clock (TC2 pin input) TC2 0: Stop and counter clear 1: Start 1: Start Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DRL), the comparison is inhibited uni (TC2DRH) is written. After writing to the upper byte, any match during 1 machine execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops (TC2S = 0). Note 4: Values to be loaded to timer register must satisfy the following condition. TC2DR > 1	TÇ2DRH (00017H) TC2DRL (00016H) Read/Write 7 6 5 4 3 2 1 0 TC2 TC2S TC2CK TC2M (Initial value: **00 00*0) TC2 0: Timer/event counter mode 1: Window mode NORMAL1/2, IDLE1/2 mode TC2 0: Timer/event counter mode DV1CK = 0 DV1CK = 1 000 fc/2 ²³ fc/2 ²⁴ 01 fc/2 ¹³ fc/2 ⁴ 01 fc/2 ³ fc/2 ⁴ 01 fc/2 ³ fc/2 ⁴ 100 Reserved Reserved 111 External clock (TC2 pin input) TC2 0: Stop and counter clear 1: Start Start control 1: Start 0: Stop and counter clear 1: Start Note 1: fc: High-frequency clock [Hz], *: Don't care Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DRL), the comparison is inhibited until the u (TC2DRH) is written. After writing to the upper byte, any match during 1 machine cycle (i execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops (TC	TÇ2DRH (00017H) TC2DRL (00016H) Read/Write TC2 TC2S TC2S TC2CK TC2 0: Timer/event counter mode operating mode select 0: Timer/event counter mode 1: Window mode NORMAL1/2, IDLE1/2 mode DV1CK = 0 DV1CK = 1 000 fc/2 ²⁴ 001 fc/2 ³ 1: Window mode fc/2 ⁴⁴ 000 fc/2 ³ 1: Window mode fc/2 ⁴⁴ 000 fc/2 ⁴⁵ 001 fc/2 ⁴⁵ 001 fc/2 ⁴⁵ 100 Reserved 101 Reserved 101 Reserved 111 External clock (TC2 pin input) TC2 0: Stop and counter clear 1: Start 10 Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Writing to the lower byte of timer register 2 (TC2DRL), the comparison is inhibited until the upper (TC2DRH) is written. After writing to the upper byte, any match during 1 machine cycle (instrue execution cycle) is ignored. Note 3: Set the mode and source clock when the TC2 stops (TC2S = 0). Note 4

Figure 2.6.2 Timer Register 2 and TC2 Control Register

2.6.3 Function

The timer/counter 2 has three operating modes: timer, event counter and window modes.

(1) Timer mode

In this mode, the internal clock is used for counting up. The contents of TC2DR are compared with the contents of up-counter. If a match is found, a timer/counter 2 interrupt (INTTC2) is generated, and the counter is cleared. Counting up is resumed after the counter is cleared.

	NORMAL, IDLE mode								
TC2CK	DV1C	CK = 0	DV1CK = 1						
10201	Resolution	Maximum time setting	Resolution	Maximum time setting					
000	524.3 [ms]	9.54 [h]	1.05 [s]	19.1 [h]					
001	512.0 [μs]	33.6 [s]	1.02 [ms]	1.12 [min]					
010	16.0 [μs]	1.05 [s]	32.0 [µs]	2.09 [s]					
011	0.5 [μs]	32.8 [ms]	1.0 [μs]	65.5 [ms]					
100	Reserved	Reserved	Reserved	Reserved					
101	Reserved	Reserved	Reserved	Reserved					

Table 2.6.1 Source Clock (internal clock) for Timer/Counter 2 (at fc = 16.0 MHz)

Example: Sets the source clock $fc/2^4$ [Hz] and generates an interrupt event 25 ms (at fc = 16 MHz, DV1CK = 1)

at fc = 1	16 MHz, DV1CK = 1		
LDW	(TC2DR), 61A8H	;	Sets TC2DR (25 ms \div 2 ⁴ /fc = 61A8H)
DI			
SET	(EIRH).6	;	Enable INTTC2 interrupt
\mathbf{EI}			
LD	(TC2CR), 00001100B	;	Selects TC2 source clock
LD	(TC2CR), 00101100B	;	Starts TC2

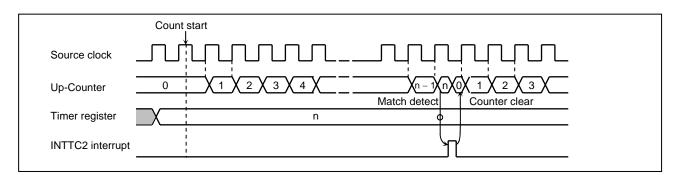


Figure 2.6.3 Timer Mode Timing Chart

(2) Event counter mode

In this mode, events are counted on the rising edge of the TC2 pin input. The contents of TC2DR are compared with the contents of the up-counter. If a match is found, an INTTC2 interrupt is generated, and the counter is cleared. The minimum pulse width to the TC2 pin is shown in Table 2.6.2. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width. Match detect is executed on the falling edge of the TC2 pin. A match can not be detected and INTTC2 is not generated when the pulse is still in a falling state.

Example: Sets the event counter mode and generates an INTTC2 interrupt 640 counts

later.			
LDW	(TC2DR), 640	;	Sets TC2DR
DI			
SET	(EIRH). 6	;	Enables INTTC2 interrupt
EI			
LD	(TC2CR), 00011100B	;	Selects TC2 source clock
LD	(TC2CR), 00111100B	;	Starts TC2

Table 2.6.2 Timer/Counter 2 External Clock Source

\backslash	Minimum pulse width [S]
	NORMAL, IDLE mode
"H" width	2 ³ /fc
"L" width	2 ³ /fc

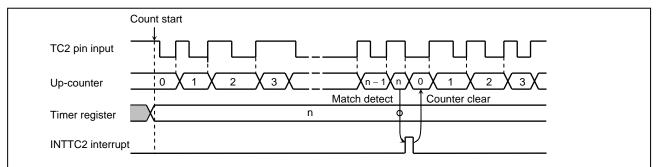


Figure 2.6.4 Event Counter Mode Timing Chart

(3) Window mode

In this mode, counting up performed on the rising edge of an internal clock during TC2 external pin input (window pulse) is "H" level. The contents of TC2DR are compared with the contents of up-counter. If a match found, an INTTC2 interrupt is generated, and the up-counter is cleared.

The maximum applied frequency (TC2 input) must be considerably slower than the selected internal clock.

Example: Generates an interrupt, inputting "H" level pulse width of 120 ms or more.

	(at fc =	16.0 MHz, DV1CK = 1)		
	LDW	(TC2DR), 0075H	;	Sets TC2DR (120 ms $\div 2^{14}$ /fc = 0075H)
	DI			
	SET	(EIRH). 6	;	Enables INTTC2 interrupt
	EI			-
	LD	(TC2CR), 00000101B	;	Selects TC2 source clock
	LD	(TC2CR), 00100101B	;	Starts TC2
TC2 pin input				
Internal clock	וחר			
Up-counter		0 1 2 (1)	3 n -	-2 $\sqrt{n-1}$ \sqrt{n} $\sqrt{1}$ $\sqrt{2}$ $\sqrt{3}$
TC2DR	χ	n		φ
				Match detect
INTTC2 interrupt				

Figure 2.6.5 Window Mode Timing Chart

2.7 8-Bit Timer/Counter3 (TC3B)

2.7.1 Configuration

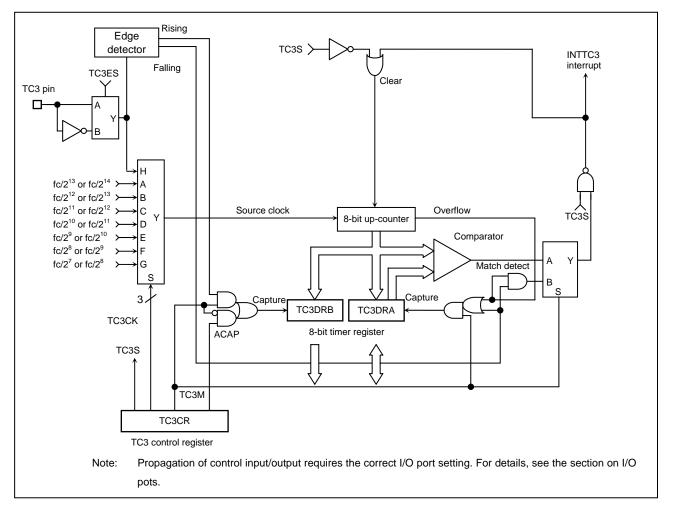


Figure 2.7.1 Timer/Counter 3 (TC3)

2.7.2 Control

The timer/counter 3 is controlled by a timer/counter 3 control register (TC3CR) and two 8-bit timer registers (TC3DRA and TC3DRB) and port multiplex control register (PMPXCR).

TC3DRA (0018H) TC3DRB (0019H)	7 6	5 4 3 2 1	0	·	iitial value: 1111 1111) tial value: 1111 1111)	
TC3CR (001AH)	7 6 ACAF		1 0 TC3M	(Initial value:		
	ТСЗМ	TC3 operating mode set	0: Timer/ever 1: Capture	nt counter		
			000 001	NORMAL, I DV1CK = 0 $fc/2^{13}$ $fc/2^{12}$	DV1CK = 1 fc/2 ¹⁴ fc/2 ¹³	
	тсзск	TC3 source clock select [Hz]	010 011 100 101 110 111	fc/2 ¹¹ fc/2 ¹⁰ fc/2 ⁹ fc/2 ⁸ fc/2 ⁷ External clock (fc/2 ¹² fc/2 ¹¹ fc/2 ¹⁰ fc/2 ⁹ fc/2 ⁸ TC3 pin input)	Write only
	TC3S ACAP	TC3 start control Auto-capture control	0: Stop and clear 1: Start 0: -			
Note 1: Acto capture control 1: Auto-capture enable Note 1: fc: High-frequency clock [Hz], *: Don't care Note 2: Set the mode and the source clock when the TC3 stops (TC3S = 0). Note 3: Values to be loaded to timer register 3A must satisfy the following condition. TC3DRA > 0 (in the timer and event counter mode) Note 4: Auto-capture can be used only in the timer and event counter mode. Note 5: Before setting TC3DRA or switching the operating mode, stop the TC3 (TC3S = 0). Note 6: When STOP mode is started, timer counter is stopped and TC3 start control (TC3S) is cleared to "0" automatically. Set TC3S to "1" after STOP mode is released for restarting timer counter. Note 7: TC3CR, TCESCR is a write-only register and must not be used with any of read-modify-write instructions.						
PMPXCR (0027H)	7 6 "0" CHS	5 4 3 2 1		(Initial value:	00** **00)	
	TC3ES	TC3 input control	0: Normal 1: Invert			Write only
	NOTE 8: AIN	ways write "0" to bit 7 in PMPXCR				

Figure 2.7.2 Timer Register 3 and TC3 Control Register

2.7.3 Function

The timer/counter 3 has three operating modes: timer, event counter, and capture mode. When it is used in the capture mode, the noise rejection time of TC3 pin input can be set by remote control receive control register.

(1) Timer mode

In this mode, the internal clock is used for counting up. The contents of TC3DRA are compared with the contents of up-counter. If a match is found, a timer/counter 3 interrupt (INTTC3) is generated, and the up-counter is cleared. The current contents of up-counter are loaded into TC3DRB by setting ACAP (bit6 in TC3CR) to "1" (Auto-capture function).

The contents of up-counter can be easily confirmed by executing the read instruction (RD instruction) of TC3DRB. Loading the contents of up-counter is not synchronized with counting up. The contents of over flow (FFH) and 00H can not be loaded correctly. It is necessary to consider the count cycle.

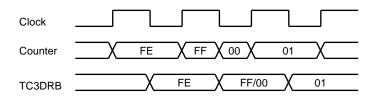


Table 2.7.1	Source Clock (internal clock) for Timer/Counter 3 (Example: at fc = 16.0 MHz)
	NORMAL IDLE mode

		NORMAL,	IDLE mode		
тсзск	DV1C	CK = 0	DV1CK = 1		
	Resolution [µs]	Maximum setting time [ms]	Resolution [µs]	Maximum setting time [ms]	
000	512	130.6	1024	261.1	
001	256	65.3	512	130.6	
010	128	32.6	256	65.3	
011	64	16.3	128	32.6	
100	32	8.2	64	16.3	
101	16	4.1	32	8.2	
110	8	2.0	16	4.1	

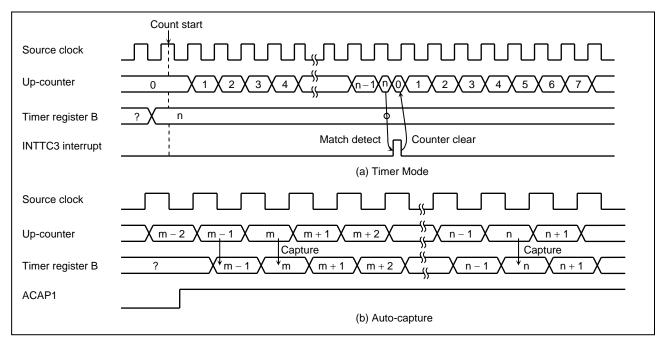


Figure 2.7.3 Timer Mode Timing Chart

(2) Event counter mode

In this mode, the TC3 pin input pulses are used for counting up Either the rising on falling edge can be selected with TC3ES (bit 0 in PMPXCR). The contents of TC3DRA are compared with the contents of the up-counter. If a match is found, an INTTC3 interrupt is generated and the counter is cleared. Match detect is executed on the falling edge of the TC3 pin. A match can not be detected, and INTTC3 is not generated when the pulse is still in a falling state.

The maximum applied frequency is shown in Table 2.7.2. Two or more machine cycles are required for both the high and low levels of the pulse width.

The current contents of up-counter are loaded into TC3DRB by setting ACAP (bit 6 in TC3CR) to "1" (Auto-capture function).

The contents of up-counter can be easily confirmed by executing the read instruction (RD instruction) of TC3DRB. Loading the contents of up-counter is not synchronized with counting up. The contents of over flow (FFH) and 00H can not be loaded correctly. It is necessary to consider the count cycle.

Example: Generates an interrupt every 0.5 s, inputting 50 Hz pulses to the TC3 pin.

LD	(TC3CR), 00001110B	;	Sets TC3 mode and source clock
LD	(TC3DRA), 19H	;	$0.5 \text{ s} \div 1/50 = 25 = 19 \text{H}$
LD	(TC3CR), 00011100B	;	Starts TC3

Table 2.7.2 Source Clock (External Clock) for Timer/Counter

Minimum applied frequency [Hz]						
	NORMAL, IDLE Mode					
"H" width	2 ² /fc					
"L" width	2 ² /fc					

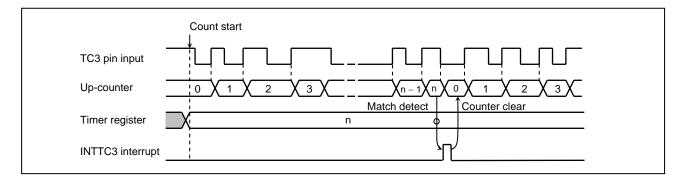


Figure 2.7.4 Event Counter Mode Timing Chart

(3) Capture mode

In this mode, the pulse width, period and duty of the TC3 pin input are measured in this mode, which can be used in decoding the remote control signals or distinguishing AC 50/60 Hz, etc. The TC3 pin input can have its polarity changed between normal and inverse by using the TC3ES Register.

a. If TC3ES = "0" (non-inverting input)

Once command operation has started, the counter free-runs on an internal source clock.

When the falling edge of the TC3 pin input is detected, the counter value is loaded into TC3DRB. When the rising edge is detected, the counter value is loaded into TC3DRA, and the counter is cleared, generating an INTTC3 interrupt.

If the rising edge is detected right after command operation has started, no capture to TC3DRB and an INTTC3 interrupt occurs only on capture to TC3DRA. If a read instruction is executed for TC3DRB, the value that exists at the end of the previous capture (immediately after a reset, "FF") is read.

b. If TC3ES = "1" (inverse input)

Once command operation has started, the counter free-runs on an internal clock.

When the rising edge of the TC3 pin input is detected, the counter value is loaded into TC3DRB. When the falling edge is detected, the counter value is loaded into TC3DRA, and the counter is cleared, generating an INTTC3 interrupt.

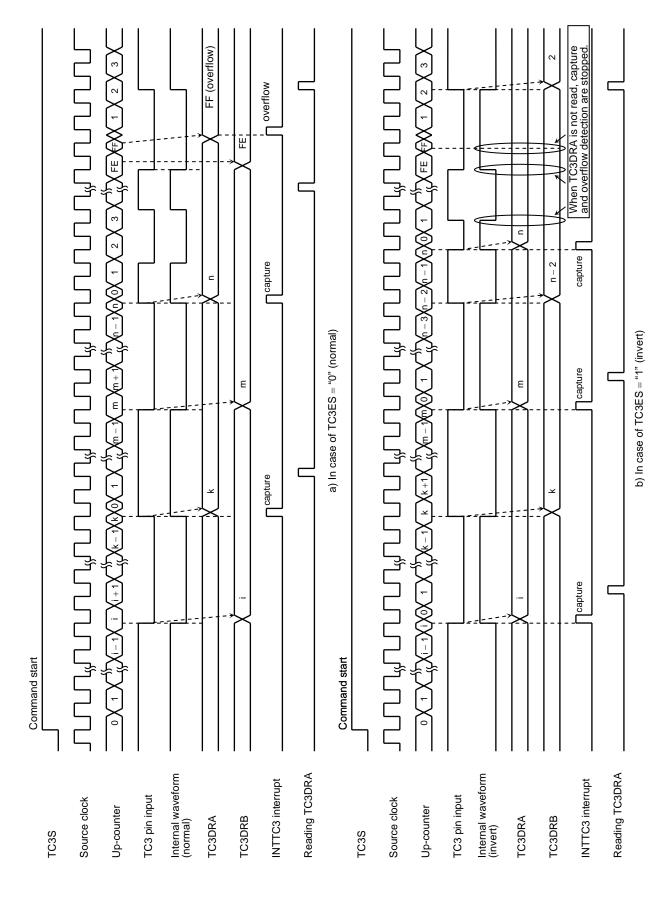
If the falling edge is detected right after command operation has started, the counter value is not captured into TC3DRB and an INTTC3 interrupt occurs only on capture to TC3DRA. If a read instruction is executed for TC3DRB, the value that exists at end of the previous capture (immediately after a reset, "FF") is read.

The minimum acceptable input pulse width is equal to the length of one source clock period selected by TC3CR <TC3CK>.

	1	1 5	
TC3ES	Capture into TC3DRB	Capture into TC3DRA	INTTC3 interrupt
"0" (non-inverting input)	Falling edge	Rising e	dge
"1" (inverting input)	Rising edge	Falling e	dge

Table 2.7.3 TC3INV-Based Capture Input Edges

Note: Capture of the TC3 pin input requires at least 1 cycle of the selected source clock.



The edge of TC3 pin input is detected in the remote control receive circuit with noize rejection. The remote control receive circuit is controlled by the remote control receive control register (RCCR). The romote control receive status register (RCSR) can monitor the porality selection and noize rejection circuit.

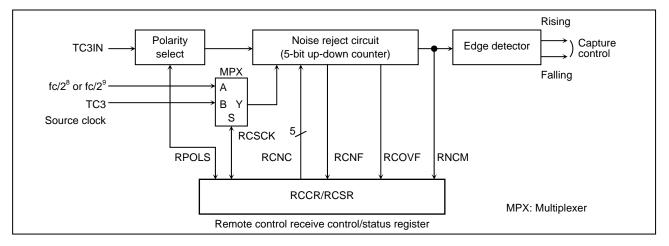


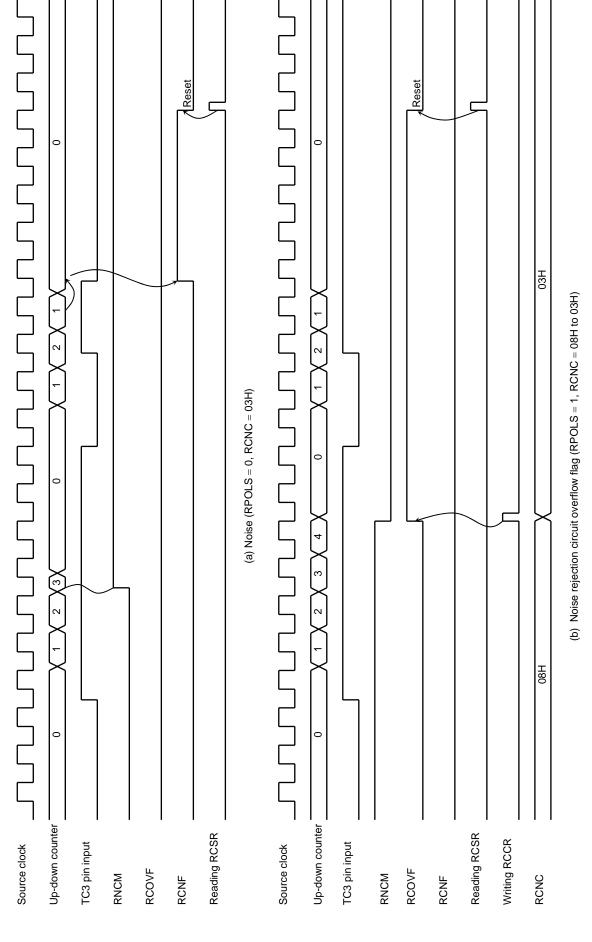
Figure 2.7.6 Remote Control Receiving Circuit

RCCR (00026H)	RCEN RF	POLS RCSCK	RCNC		(Initial value: 0001 1111)			
	RCNC	Noise reject time select $02H \le RCNC \le 1FH$	(Source c	(Source clock) \times (RCNC – 1) [s]					
				NORMAL,	IDLE mode				
		Noise reject circuit		DV1CK = 0	DV1CK = 1				
	RCSCK	Source clock select	0	2 ⁸ /fc	2 ⁹ /fc				
			1	TC3CK	Note 2	R/W			
		Remote control signal polarity	0: Positive	9		-			
	RPOLS	select	1: Negativ	'e					
	DOEN	Remote control receive circuit	0: Disable	1		Write			
	RCEN	operation control	1: Enable			only			
	Note 1: S	et RPOLS and RCSCK when the t	imer/count	er stops (TC3S = 0)					
	Note 2: S	ource clock of timer/counter 3							
	Note 3: fo	: High-frequency clock [Hz], *: Do	n't care						
	Note 4: R	CCR includes a write-only register	r and must	not be used with any of read	d-modify-write instructions				
	Note 5: V	alues to be loaded to RCNC must	satisfy the	following condition. $02 \le RC$	CNC ≤ 1F				
				-					
RCSR									
(00026H)	RCNF RF	POLS RCSCK RCOVF RNCM			(Initial value: 0000 0***)			
		Remote control signal monitor	0: Low le	vel					
	RNCM	after noise rejecter	1: High level						
		· · · · · · · · · · · · · · · · · · ·	0: Signal and definition by overwriting the noise reject time						
	RCOVF	Noise reject circuit Overflow	RCNC						
		flag	1: Overflow						
				NORMAL,	IDLE mode				
	RCSCK	Noise reject circuit		DV1CK = 0	DV1CK = 1				
	RUSUR	Source clock select	0	2 ⁸ /fc	2 ⁹ /fc	R/W			
			1 TC3CK Note 2						
		Remote control signal polarity	0: Positive						
	RPOLS	select	1: Negative						
	DONE	Remote control signal monitor	0: Without noise						
	RCNF	after noise rejecter	1: With noise						
	Note 1: Reading out the register RCSR resets RCNF and RCOVF.								
	Note 2: S	ource clock of timer/counter 3							
	Note 3: W	/hen a 5-bit up-down counter coun	ts down to	"0" after counting up. the R	CNF defines to be noise.				
		: High-frequency clock [Hz], *: Do							
	Note 4. IC. High Frieducity Clock [12], ". Don't care								

Table 2.7.4 Combination between The Polarity and The Edge Selection

RPOLS	TC3 pin input pulse (Interrupt occurrence is shown as allow.)	Measurement
0		
1		

Note: When TC3CK is used in RCSCK, do not select an external clock to the TC3CK.



2.8 8-Bit Timer/Counter 4 (TC4)

2.8.1 Configuration

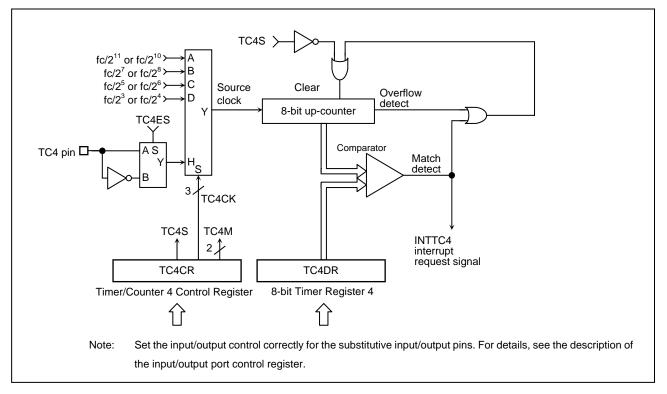


Figure 2.8.1 Timer/Counter 4 (TC4)

2.8.2 Control

The timer/counter 4 is controlled by a timer/counter 4 control register (TC4CR) and an 8-bit timer register 4 (TC4DR).

TC4DR	7	6	5	4	3	2	1	0	-			
(0001BH)		1			1				Write only (Initi	al value: 1111 1	111)	
TC4CR	7	6	5	4	3	2	1	0	-			
(0001CH)		:	TC4S	S TC4CK TC4M Write only (Initial value: **00 000							000)	
					00: T	imer/even	t counter	mode				
TC4S	TC4 start control				01: R	eserved						
1010	10104					eserved						
					11: R	eserved				<u> </u>		
								/AL, IDLE		-		
							CK = 0		$\frac{\text{DV1CK} = 1}{\text{fs/2}^{12}}$	+		
					000		c/2 ¹¹			-		
					001		c/2 ⁷		fs/2 ⁸	-		
TC4CK			select [Hz	z]	010		c/2⁵		fs/2 ⁶	-	R/W	
	(Note 4	•)			011	f	c/2 ³		fs/2 ⁴	-		
					100		served		Reserved	-		
					101		served		Reserved	-	_	
					110	Re	served		Reserved	-		
					111							
			00: Timer/event counter mode									
TC4M TC4 ope		operating mode select				01: Reserved 10: Reserved						
						11: Reserved						
	Note 1:	fo: Lligh f	roquopov									
		-	requency	-	-		atiati (tha	following	condition 1 < TO	4DD < 255		
					•		-	-	condition. $1 \le TC$		240	
	Note 3:								0) or while the TC			
								nese regi	sters are selected	changed during	tnese	
		•	ns, countin	•			•					
	Note 4:					ounter is s	topped a	and cleare	ed. Set TC4S to "1	" after STOP me	ode is	
released for restarting timer counter.												
	Note 5:	Undefine	d values a	re read f	rom bits 6	and 7 of	FC4CR.					
	Note 6:	Do not cl	nange TC4	DR while	e the TC4	is operati	ng.					
PMPXCR	7	6	5	4	3	2	1	0				
(00027H)		CHS	·		•		TC4ES	·) (Initial value: 00	0** **00)		
TC4ES	TC4 e	edge selec	t			tising edg alling edg					Writ onl	
	1				I							
	Note 1:	TC4CR,	TC4DR	and PMI	PXCR ar	e write c	nly regi	ster and	must not be us	sed with any c	of the	

Figure 2.8.2 Timer Register 4 and TC4 Control Register

2.8.3 Function

The timer/counter 4 has two operating modes: timer, event counter mode.

(1) Timer mode

In this mode, the internal clock is used for counting up. The contents of TC4DR are compared with the contents of up-counter. If a match is found, an INTTC4 interrupt is generated and the up-counter is cleared to "0". Counting up resumes after the up-counter is cleared.

Table 2.8.1	Source Clock	(internal clock)) for Timer/Counter 4	(Example: at fc = 16.0 MHz)
10010 2.0.1				$(\mathbf{E} \times \mathbf{u} + \mathbf{b}) = 10 \cdot 0 + 10 \cdot 0$

	NORMAL, IDLE mode					
TC4CK	DV1C	CK = 0	DV1CK = 1			
	Resolution [µs]	Maximum setting time [ms]	Resolution [µs]	Maximum setting time [ms]		
000	128.0	32.6	256.0	65.3		
001	8.0	2.0	16.0	4.1		
010	2.0	0.510	4.0	1.0		
100	0.5	0.128	1.0	0.255		

(2) Event counter mode

In this mode, the TC4 pin input (external clock) pulse is used for counting up. Either the rising or falling edge can be selected with TC4ES (bit 1 PMPXCR). The contents of TC4DR are compared with the contents of the up-counter. If a match is found, an INTTC4 interrupt is generated and the counter is cleared. The maximum applied frequency is shown Table 2.8.2. Two or more machine cycles are required for both the high and low level of the pulse width.

Note: The event counter mode can only be used in NORMAL or IDLE mode.

Minimum input pulse width [s]			
	NORMAL1, IDLE1 mode		
"H" width	2 ³ /fc		
"L" width	2 ³ /fc		

Table 2.8.2 Timer/Counter 4 External Clock Source

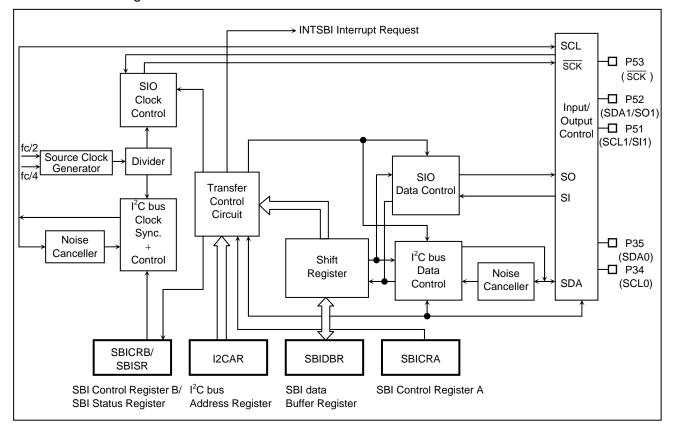
2.9 Serial Bus Interface (SBI-ver. D)

The TMP88CS34/CP34 has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit serial bus interface and an I^2C bus (a bus system by Philips). The serial bus interface pins are selectively used as either channel 0 or channel 1.

The serial interface is connected to external devices through P35 (SDA0)/P52 (SDA1) and P34 (SCL0)/P51 (SCL1) in the I²C bus mode; and through P53 ($\overline{SCK1}$), P52 (SO1) and P51 (SI1) in the clocked-synchronous 8-bit SIO mode.

The serial bus interface pins are also used for the P3/P5 port. When used for serial bus interface pins, set the P3/P5 output latches of these pins to "1". When not used as serial bus interface pins, the P3/P5 port is used as a normal I/O port.

- Note 1: When P3 and P5 is used as serial bus interface pins, P35, P34, P51 and P50 should be set as a sink open drain output by clearing PSELCR to "0".
- Note 2: The I²C of TMP88CS34/CP34 can be used only in the Standard mode of I²C. The Fast mode and the High Speed mode can not be used.



2.9.1 Configuration

Figure 2.9.1 Serial Bus Interface (SBI)

2.9.2 Control

The following registers are used for control the serial bus interface and monitor the operation status.

- Serial bus interface control register A (SBICRA)
- Serial bus interface control register B (SBICRB)
- Serial bus interface data buffer register (SBIDBR)
- I²C bus address register (I2CAR)
- Serial bus interface status register A (SBISRA)
- Serial bus interface status register B (SBISRB)
- Serial clock source control register (SCCRB)
- Serial clock control status register (SCSR)

The above registers differ depending on a mode to be used. Refer to Section " $2.9.7 I^2C$ bus mode control" and "2.9.9 Clocked-synchronous 8-bit SIO mode control".

2.9.3 Serial Clock Source Control

A serial bus interface circuit can reduce the power consumption by stopping a serial clock generater.

Serial Clocl	source Con	trol Register		
SCCRB (00FF1H)	7 6 SCEN	5 4 3	2 1 0 (Initial value: 0*** ****)	
	SCEN	Serial clock source control	0: Do not generate source clock 1: Generate source clock	Write only
	Note: When S	CRQ and SCEN are "1", SCEN can	not be cleared to "0". When SCRQ is "0", SCEN is cleared	d to "0".
Serial Clocl	Control Stat	tus Register		
SCSR (00FF1H)	7 6 SCRQ	5 4 3	2 1 0	
	SCRQ	Serial clock source request	0: No source clock request from serial bus interface 1: Source clock request from serial bus interface	Read only
	SCRQ			
	SCEN)		
	Source clock		Clock generation	
		"1" → SCEN Write c "00" to	lata except Write data "0" → SCEN SBIM "00" to SBIM	

Figure 2.9.2 Serial Clock Source

2.9.4 Channel Select

A serial bus interface circuit can select I/O pin when a serial bus interface is used for $\rm I^2C$ bus mode.

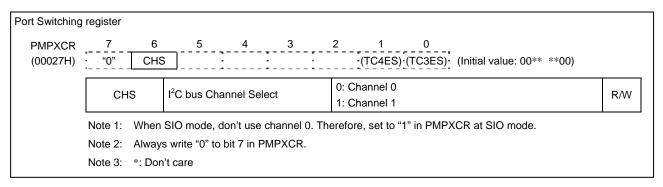


Figure 2.9.3 Channel Select

2.9.5 Software Reset

A serial bus interface circuit has a software reset function, when a serial bus interface circuit is locked by an external noise, etc.

To occur software reset, write "01", "10" into the SWRST (bit 1, 0 in SBICRB). During software reset, the SWRMON (bit 0 in SBISRA) is clear to "0".

2.9.6 The Data Format in The I²C bus Mode

The data format when using the TMP88CS34/CP34 in the $I^2\mathrm{C}$ bus mode are shown in as below.

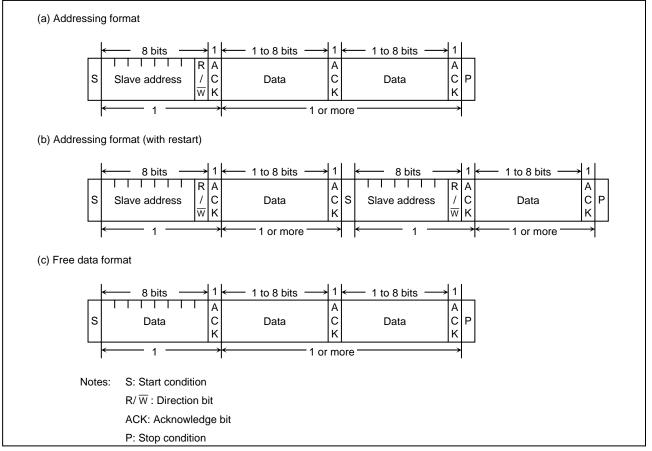


Figure 2.9.4 Data Format in I²C Bus Mode

2.9.7 I²C Bus Mode Control

The following registers are used to control the serial bus interface (SBI) and monitor the operation status in the I²C bus mode.

	nterface C	Control Register A						
SBICRA (00020H)	7	6 5 4 - 3 BC ACK	2	1 SCK	0	(Initial value:	0000 *000)	
(0002011)	,		1	<u>, 30k</u>		(initial value:	0000 *000)	
			вс	ACK	= 0	ACK	= 1	
				Number of Clock	Bits	Number of Clock	Bits	
			000	8	8	9	8	
	BC	Number of transferred bits	001 010	1 2	1 2	2 3	1 2	Write
			010	3	3	4	3	only
			100	4	4	5	4	
			101 110	5 6	5 6	6 7	5 6	
			111	7	7	8	7	
			ACK	Master I		Slave n	node	
			0	Not generat pulse for an		Not count a cl	ock pulse for	
	ACK	Acknowledgement mode	0	acknowledg		an acknowled	gement.	R/W
		specification		Generate a				
			1	pulse for an		Count a clock acknowledge	•	
				acknowledg		-		
			000.	DV1CK = Reserved (N	-	DV1CI 000: Reserve		
				Reserved (N	,	001: Reserve	, ,	
	0.017	Serial clock selection		Reserved (N	Note 3)	010: 58.8 kH		Write
	SCK	(At fc = 16 MHz, Output on SCL pin)		60.6 kHz 30.7 kHz		011: 30.3 kH 100: 15.4 kH		only
			100.			101: 7.7 kH		
			110:	7.8 kHz		110: 3.9 kH		
	Nata di Ca	the DO to "000" before evitables to 0		Reserved		111 : Reserve	d	
		et the BC to "000" before switching to 8 BICRA cannot be used with any of read			ctions such	as hit maninula	tion etc	
		his I2C bus circuit does not support the		-				
		e I2C bus circuit itself allows the settin						
	sp	pecification is not guaranteed in that ca	se.					
Serial Bus Inte	erface Data B	Buffer Register						
SBIDBR	7	6 5 4 3	2	1	0			
(00021H)	<u> </u>			<u> </u>	(Ini	tial value: ****	****) R/W	1
		or writing transmitted data, start from th the data which was written into SBIDBF			nco o writo	data huffor and	l a road buff	
		dependent in SBIDBR. Therefore, SB		,				
		ch as bit manipulation, etc.			,		,	
I ² C bus Addres	ss Register							
	7	6 5 4 3	2	1	0			
I2CAR		Slave address		٨				
(00022H)	SA6	SA5 SA4 SA3 SA2	SA1	<u> </u> SA0	ALS	(Initial value	: 0000 0000)	
	SA	Slave address selection						
	ALS		ave ac	ddress recogn	ition			Write only
				ve address red				-
		CAR is write-only register and canno	t be u	sed with any	of read-mo	odify-write instr	uction such a	as bit
		anipulation, etc. o not set I2CAR to "00H" to avoid the ind	correct	response of a	acknowledg	ment in slave m	ode. If "00H"	is set
		I2CAR as the Slave Address and						
	ac	knowledgement incorrectly.						

Figure 2.9.5 Serial Bus Interface Control Register A, Serial Bus Interface Data Buffer Register and I^2C Bus Address Register In The I^2C Bus Mode

Serial Bus	Interface	Control Register B			
SBICRB	7	6 5 4 3 2 1 0			
(00023H)	MST	TRX BB PIN SBIM SWRST1SWRST0 (Initial value: 0001 000	0)		
	MST	Master/Slave selection 0: Slave 1: Master			
	TRX	Transmitter/Receiver selection 0: Receiver 1: Transmitter			
	BB	Start/Stop generation 0: Generate a stop condition when MST, TRX and PIN are "1" 1: Generate a start condition when MST, TRX and PIN are "1"			
	PIN	Cancel interrupt service request 1: Cancel interrupt service request	Write only		
	SBIM	Serial bus interface operating mode selection 00: Port mode (Serial bus interface output disable) 01: Clocked synchronous 8-bit SIO mode 10: I ² C bus mode 11: Reserved			
	SWRST1 SWRST0	Software reset start bit Software reset starts by first writing "10" and next writing "01".			
	Note 2: S	witch a mode to port after confirming that the bus is free. witch a mode to I ² C bus mode or clock synchronous 8-bit SIO mode after confirming that the igh-level.			
		BICRB has write-only register and must not be used with any of read-modify-write instructions suc nanipulation, etc.	n as dit		
	Note 4: V	When the SWRST (bit 1, 0 in SBICRB) is written to "01", "10", software reset (four machine cycles occurred.			
		his time, control the serial bus interface and monitor the operation status registers except the SBIN	/I (bit 3,		
		in SBICRB) and the CHS (bit 6 in PMPXCR) are reseted.			
		Control the serial bus interface and monitor the operation status registers are SBICRA, SBICRB, SI 2CAR, SBISRA, SBISRB, SCCRA and SCSR.	BIDBR,		
Serial Bus I		Status Register A			
	7	6 5 4 3 2 1 0			
SBISRA (00020H)		ACK SWR MON (Initial value: **** ****	1)		
	SWRMON	N Software reset monitor 0: During software reset 1: - (Initial)	Read only		
	Note 1: *	: Don't care	I		
Serial Bus I	Interface	Status Register B			
SBISRB	7	6 5 4 3 2 1 0			
(00023H)	MST	TRX BB PIN AL AAS AD0 LRB (Initial value: 0001 000)	0)		
	MST	Master/Slave selection status 0: Slave monitor 1: Master			
	TRX	Transmitter/Receiver selection 0: Receiver status monitor 1: Transmitter			
	BB	Bus status monitor 0: Bus free 1: Bus busy			
	PIN	Interrupt service requests 0: Requesting interrupt service	Deed		
	AL	status monitor 1: Releasing interrupt service request Arbitration lost detection 0: - monitor 1: Arbitration lost detected	Read only		
	AAS Slave address match detection 0: Not detect slave address match or "GENERAL CALL"				
	AD0	monitor 1: Detect slave address match or "GENERAL CALL" "GENERAL CALL" detection 0: Not detect "GENERAL CALL"			
		monitor 1: Detect "GENERAL CALL"			
	LRB	Last Received bit monitor 0: Last receive bit is "0" 1: Last receive bit is "1"			

Figure 2.9.6 Serial Bus Interface Control Register B and Serial Bus Interface

Status Register A/B in the I²C Bus Mode

- (1) Acknowledgement mode specification
 - a. Acknowledgement mode (ACK = "1")

To set the device as an acknowledgement mode, the ACK (bit4 in SBICRA) should be set to "1". When a serial bus interface circuit is a master mode, an additional clock pulse is generated for an acknowledge signal. In a slave mode, a clock is counted for the acknowledge signal.

In the master transmitter mode, the SDA pin is released in order to receive an acknowledge signal from the receiver during additional clock pulse cycle. In the master receiver mode, the SDA pin is set to low level generation an acknowledge signal during additional clock pulse cycle.

In a slave mode, when a received slave address matches to a slave address which is set to the I2CAR or when a "GENERAL CALL" is received, the SDA pin is set to low level generating an acknowledge signal. After the matching of slave address or the detection of "GENERAL CALL", in the transmitter the SDA pin is released in order to receive an acknowledge signal from the receiver during additional clock pulse cycle. In a receiver, the SDA pin is set to low level generation an acknowledge signal during additional clock pulse cycle after the matching of slave address or the detection of "GENERAL CALL".

The Table 2.9.1 shows the SCL and SDA pins status in acknowledgement mode.

Mode	Pin Transmitter		Receiver		
	SCL		An additional clock pulse is generated.		
Master		SDA	Released in order to receive and acknowledge signal.	Set to low level generating an acknowledge signal.	
		SCL	A clock is counted for the acknowledge signal.		
Slave	SDA	When slave address matches or a general call is detected	_	Set to low level generating an acknowledge signal.	
		After matching of slave address or general call	Released in order to receive an acknowledge signal.	Set to low level generating an acknowledge signal.	

Table 2.9.1 SCL and SDA Pins Status in Acknowledgement Mode

b. Non-acknowledgement mode (ACK = "0")

To set the device as a non-acknowledgement mode, the ACK should be cleared to "0". In the master mode, a clock pulse for an acknowledge signal is not generated. In the slave mode, a clock for a acknowledge signal is not counted.

(2) Number of transfer bits

The BC (bits 7 to 5 in SBICRA) is used to select a number of bits for next transmitting and receiving data.

Since the BC is cleared to "000" as a start condition, a slave address and direction bit transmissions are always executed in 8 bits. Other than these, the BC retains a specified value.

(3) Serial clock

a. Clock source

The SCK (bits 2 to 0 in SBICRA) is used to select a maximum transfer frequency output from the SCL pin in the master mode. Set a communication baud rate that meets the I2C bus specification, such as the shortest pulse width of tLOW, based on the equations shown below.

Four or more machine cycles are required for both high and low levels of pulse width in the external clock which is input from SCL pin.

Note: Since the I2C of TMP88CS34/CP34 can not be used as the Fast mode and the High Speed mode, do not set SCK as the frequency that is over 100 kHz.

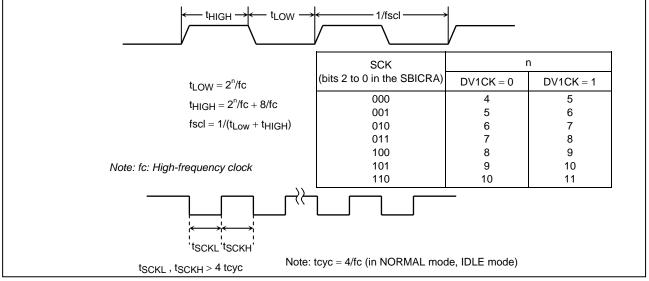


Figure 2.9.7 Clock Source

b. Clock synchronization

In the I²C bus mode, in order to drive a bus with a wired AND, a master device which pulls down a clock pulse to low will, in the first place, invalidate a clock pulse of another master device which generates a high-level clock pulse.

The serial bus interface circuit has a clock synchronization function. This function ensures normal transfer even if there are two or more masters on the same bus.

The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

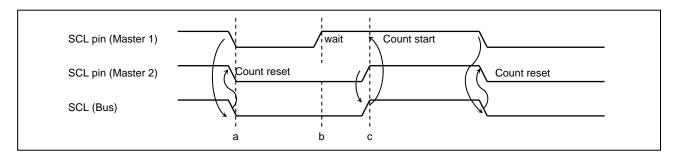


Figure 2.9.8 Clock Synchronization

As Master 1 pulls down the SCL pin to the low level at point "a", the SCL line of the bus becomes the low level. After detecting this situation, Master 2 resets counting a clock pulse in the high level and sets the SCL pin to the low level.

Master 1 finishes counting a clock pulse in the low level at point "b" and sets the SCL

pin to the high level. Since Master 2 holds the SCL line of the bus at the low level, Master 1 waits for counting a clock pulse in the high level. After Master 2 sets a clock pulse to the high level at point "c" and detects the SCL line of the bus at the high level, Master 1 starts counting a clock pulse in the high level. Then, the master, which has finished the counting a clock pulse in the high level, pulls down the SCL pin to the low level.

The clock pulse on the bus is detemined by the master device with the shortest high-level period and the master device with the longest low-level period from among those master devices connected to the bus.

(4) Slave address and address recognition mode specification

When the serial bus interface circuit is used with an addressing format to recognize the slave address, clear the ALS (bit 0 in I2CAR) to "0", and set the SA (bits 7 to 1 in I2CAR) to the slave address.

When the serial bus interfac circuit is used with a free data format not to recognize the slave address, set the ALS to "1". With a free data format, the slave address and the direction bit are not recognized, and they are processed as data from immediately after start condition.

(5) Master/slave selection

To set a master device, the MST (bit 7 in SBICRB) should be set to "1". To set a slave device, the MST should be cleared to "0".

When a stop condition on the bus or an arbitration lost is detected, the MST is cleared to "0" by the hardware.

(6) Transmitter/receiver selection

To set the device as a transmitter, the TRX (bit 6 in SBICRB) should be set to "1". To set the device as a receiver, the TRX should be cleared to "0". When data with an addressing format is transferred in the slave mode, the TRX is set to "1" by a hardware if the direction bit (\mathbb{R}/\overline{W}) sent from the master device is "1", and is cleared to "0" by a hardware if the bit is "0. In the master mode, after an acknowledge signal is returned from the slave device, the TRX is cleared to "0" by a hardware if a transmitted direction bit is "1", and is set to "1" by a hardware if it is "0". When an acknowledge signal is not returned, the current condition is maintained.

When a stop condition on the bus or an arbitration lost is detected, the TRX is cleared to "0" by the hardware. The following table show TRX changing conditions in each mode and TRX value after changing.

Mode	Direction Bit	Conditions	TRX after Changing
Slave	"0"	A received slave address is the	"O"
mode	"1"	same value set to I2CAR	"1"
Master	"0"	ACK signal is returned	"1"
mode	"1"	ACIT Signal IS returned	"O"

When a serial bus interface circuit operates in the free data format, a slave address and a direction bit are not recognized. They are handled as data just after generating a start condition. The TRX is not changed by a hardware.

(7) Start/Stop condition generation

When the BB (bit 5 in SBICRB) is "0", a slave address and a direction bit which are set to the SBIDBR are output on a bus after generating a start condition by writing "1" to the MST, TRX, BB and PIN. It is necessary to set transmitted data to the SBIDBR and set "1" to ACK beforehand.

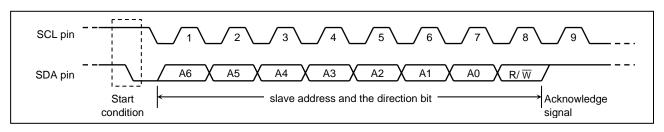
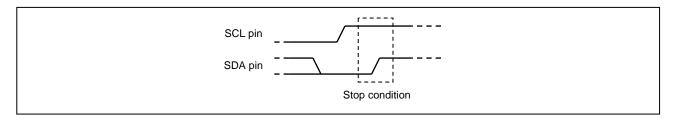


Figure 2.9.9 Start Condition Generation and Slave Address Generation

When the BB is "1", sequence of generating a stop condition is started by writeng "1" to the MST, TRX and PIN, and "0" to the BB. Do not modify the contents of MST, TRX, BB and PIN until a stop condition is generated on a bus.





When a stop condition is generated and the SCL line on a bus is pulled-down to low level by another device, a stop condition is generated after releasing the SCL line.

The bus condition can be indicated by reading the contents of the BB (bit 5 in SBISRB). The BB is set to "1" when a start condition on a bus is detected and is cleared to "0" when a stop condition is detected.

(8) Interrupt service request and cancel

When a serial bus interface circuit is in the master mode and transferring a number of clocks set by the BC and the ACK is complete, a serial bus interface interrupt request (INTSBI) is generated.

In the slave mode, the conditions of generating INTSBI are follows:

- At the end of acknowledge signal when the received slave address matches to the value set by the I2CAR
- At the end of acknowledge signal when a "GENERAL CALL" is received
- At the end of transferring or receiving after matching of slave address or receiving of "GENRAL CALL"

When a serial bus interface interrupt request occurs, the PIN (bit 4 in SBISR) is cleared to "0". During the time that the PIN is "0", the SCL pin is pulled-down to low level.

Either writing data to SBIDBR or reading data from the SBIDBR sets the PIN to "1".

The time from the PIN being set to "1" until the SCL pin is released takes t_{LOW} .

Although the PIN (bit 4 in SBICRB) can be set to "1" by the program, the PIN can not be cleared to "0" by the program.

Note: If the arbitration lost occurs, when the slave address does not match, the PIN is not cleared to "0" even thought INTSBI is generated.

(9) Serial bus interface operating mode selection

The SBIM (bit 3 and 2 in SBICRB) is used to specify a serial bus interface operation mode.

Set the SBIM to "10" in order to change a operation mode to I^2C bus mode. Before changing operation mode, confirm serial bus interface pins in a high level. And switch a mode to port after confirming that a bus is free.

(10) Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus in the I^2C bus mode, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the $\mathrm{I}^{2}\mathrm{C}$ bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master 1 and Master 2 output the same data until point "a". After Master 1 outputs "1" and Master 2, "0", the SDA line of a bus is wired AND and the SDA line is pulled-down to the low level by Master 2. When the SCL line of a bus is pulled-up at point "b", the slave device reads data on the SDA line, that is data in Master 2.

Data transmitted from Master 1 becomes invalid. The state in Master 1 is called "arbitration lost". A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

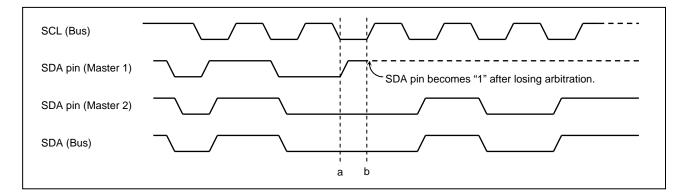


Figure 2.9.11 Arbitration Lost

The serial bus interface circuit compares levels of a SDA line of a bus with its those SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and the AL (bit 3 in SBISRB) is set to "1".

When the AL is set to "1", the MST and TRX are cleared to "0" and the mode is switched to a slave receiver mode.

The AL is cleared to "0" by writing or reading data to or from the SBIDBR or writing data to the SBICRB.

Master	SCL pin				-
A	SDA pin	D7A X D6A X D5A X D4A X D3A X D2A X D1A X D0A			-
Master	SCL pin	$ 1 2 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 7 \cdot 7$;		
В	SDA pin	D7B D6B Releasing SDA pin and SCL pin to high level as losir	ing art	pitration.	
	AL				
	MST	()		<u>\</u>	
	TRX)	
	Accesse SBIDBR	d to		ĺ	
	INTSBI	ľ			

Figure 2.9.12 Example of when a Serial Bus Interface Circuit is a Master B

(11) Slave address match detection monitor

In the slave mode, the AAS (bit 2 in SBISR) is set to "1" when the received data is "GENERAL CALL" or the received data matches the slave address setting by I2CAR with an address recognition mode (ALS = 0).

When a serial bus interface circuit operates in the free data format (ALS = 1), the AAS is set to "1" after receiving the first 1-word of data.

The AAS is cleared to "0" by writing data to the SBIDBR or reading data from the SBIDBR.

(12) GENERAL CALL detection monitor

The AD0 (bit 1 in SBISR) is set to "1" when all 8-bit received data is "0" immediately after a start condition in a slave mode. The AD0 is cleared to "0" when a start or stop condition is detected on a bus.

(13) Last received bit monitor

The SDA value stored at the rising edge of the SCL is set to the LRB (bit0 in SBISRB). In the acknowledge mode, immediately after an INTSBI interrupt request is generated, an acknowledge signal is read by reading the contents of the LRB.

2.9.8 Data Transfer of I²C Bus

(1) Device initialization

For initialization of device, set the ACK in SBICRA to "1" and the BC to "000". Specify the data length to 8 bits to count clocks for an acknowledge signal. Set a transfer frequency to the SCK in SBICRA.

Next, set the slave address to the SA in I2CAR and clear the ALS to "0" to set an addressing format.

After confirming that the serial bus interface pin is high-level, for specifying the default setting to a slave receiver mode, clear "0" to the MST, TRX and BB in SBICRB, set "1" to the PIN, "10" to the SBIM, and "00" to bits SWRST1 and SWRST0.

- Note: The initialization of a serial bus interface circuit must be complete within the time from all devices which are connected to a bus have initialized to and device does not generate a start condition. If not, the data can not be received correctly because the other device starts transferring before an end of the initialization of a serial bus interface circuit.
- (2) Start condition and slave address generation

Confirm a bus free status (when BB = 0).

Set the ACK to "1" and specify a slave address and a direction bit to be transmitted to the SBIDBR.

By writing "1" to the MST, TRX, BB and PIN, the start condition is generated on a bus and then, the slave address and the direction bit which are set to the SBIDBR are output. An INTSBI interrupt request occurs at the 9th falling edge of a SCL clock cycle, and the PIN is cleared to "0". The SCL pin is pulled-down to the low level while the PIN is "0". When an interrupt request occurs the TRX changes by the hardware according to the direction bits only when an acknowledge signal is returned from the slave device.

- Note 1: Do not write a slave address to be output to the SBIDBR while data is transferred. If data is written to the SBIDBR, data to been outputting may be destroyed.
- Note 2: The bus free must be confirmed by software within 98.0 μ s (the shortest transmitting time according to the l²C bus standard) after setting of the slave address to be output. Only when the bus free is confirmed, set "1" to the MST, TRX, BB, and PIN doesn't finish within 98.0 μ s, the other masters may start the transferring and the slave address data written in SBIDBR may be broken.

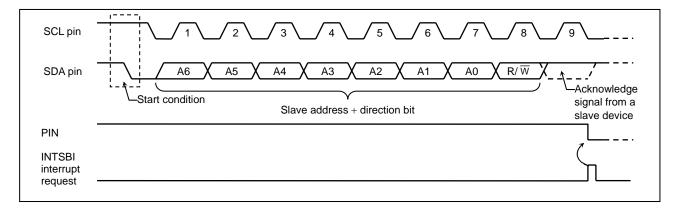


Figure 2.9.13 Start Condition Generation and Slave Address Transfer

(3) 1-word data transfer

Check the MST by the INTSBI interrupt process after an 1-word data transfer is completed, and determine whether the mode is a master or slave.

a. When the MST is "1" (Master mode)

Check the TRX and determine whether the mode is a transmitter or receiver.

1. When the TRX is "1" (Transmitter mode)

Test the LRB. When the LRB is "1", a receiver does not request data. Implement the process to generate a stop condition (described later) and terminate data transfer.

When the LRB is "0", the receiver requests next data. When the next transmitted data is other than 8 bits, set the BC, set the ACK to "1", and write the transmitted data to the SBIDBR. After writing the data, the PIN becomes "1", a serial clock pulse is generated for transferring a next 1-word of data from the SCL pin, and then the 1-word data is transmitted. After the data is transmitted, and an INTSBI interrupt request occurs. The PIN become "0" and the SCL pin is set to low level. If the data to be transferred is more than one word in length, repeat the procedure from the LRB test above.

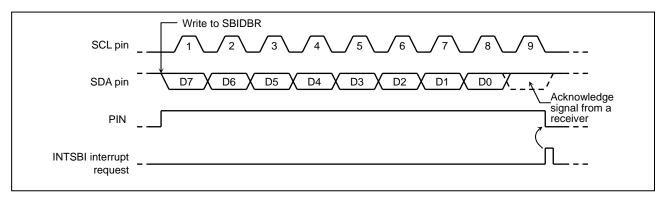


Figure 2.9.14 Example of when BC = "000", ACK = "1"

2. When the TRX is "0" (Receiver mode)

When the next transmitted data is other than of 8 bits, set the BC again. Set the ACK to "1" and read the received data from the SBIDBR (reading data is undefined immediately after a slave address is sent). After the data is read, the PIN becomes "1". A serial bus interface circuit outputs a serial clock pulse to the SCL to transfer next 1-word of data and sets the SDA pin to "0" at the acknowledge signal timing.

An INTSBI interrupt request occurs and the PIN becomes "0". Then a serial bus interface circuit outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBIDBR.

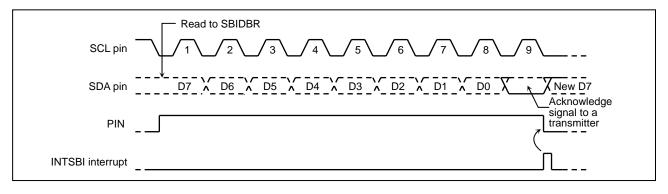
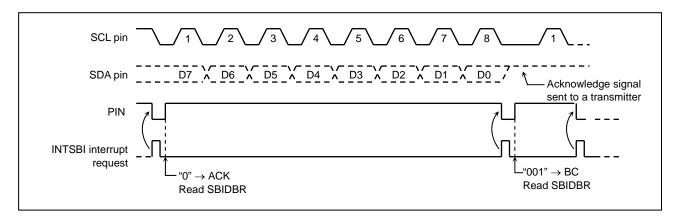
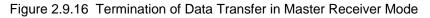


Figure 2.9.15 Example of when BC = "000", ACK = "1"

To make the transmitter terminate transmit, clear the ACK to "0" before reading data which is 1-word before the last data to be received. A serial bus interface circuit does not generate a clock pulse for the acknowledge signal by clearing ACK. In the interrupt routine of end of transmission, when the BC is set to "001" and read the data, PIN is set to "1" and generates a clock pulse for a 1-bit data transfer. In this case, since the master device is a receiver, the SDA line on a bus keeps the high-level. The transmitter receives the high-level signal as an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, generates the stop condition to terminate data transfer.





b. When the MST is "0" (Slave mode)

In the slave mode, a serial bus interface circuit operates either in normal slave mode or in slave mode after losing arbitration.

In the slave mode, the conditions of generating INTSBI are follows:

- When the received slave address matches to the value set by the I2CAR
- When a "GENERAL CALL" is received
- At the end of transferring or receiving after matching of slave address or receiving of "GENERAL CALL"

A serial bus interface circuit changes to a slave mode if arbitration is lost in the master mode. And an INTSBI interrupt request occurs when word data transfer terminates after losing arbitration. The behavior of INTSBI and PIN after losing arbitration are shown in Table 2.9.2.

	When the arbitration occurs during transmission of slave address as a master	When the arbitration occurs during transmission of data as a master transmit mode		
INTSBI	INTSIB is generated at th	e terminatin of word data.		
PIN	When the slave address matches the value set by I2CAR, the PIN is cleared to "0" by generating of INTSBI. When the slave address doesn't match the value set by I2CAR, the PIN keeps "1".	PIN keeps "1".		

Check the AL (bit 3 in the SBISR), the TRX (bit 6 in the SBISR), the AAS (bit 2 in the SBISR), and the AD0 (bit 1 in the SBISR) and implements processes according to conditions listed in Table 2.9.3.

TRX	AL	AAS	AD0	Conditions	Process
1	1	1	0	A serial bus interface circuit loses arbitration when transmitting a slave address. And receives a slave address of which the value of the direction bit sent from another master is "1".	Set the number of bits in 1 word to the BC and write transmitted data to the SBIDBR.
	0	1	0	In the slave receiver mode, a serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "1".	
		0	0	In the slave transmitter mode, 1-word data is transmitted.	Test the LRB. If the LRB is set to "1", set the PIN to "1" since the receiver does not request next data. Then, clear the TRX to "0" release the bus. If the LRB is set to "0", set the number of bits in 1-word to the BC and write transmitted data to the SBIDBR since the receiver requests next data.
0	1	1	1/0	A serial bus interface circuit loses arbitration when transmitting a slave address. And receives a slave address of which the value of the direction bit sent from another master is "0" or receives a "GENERAL CALL".	Read the SBIDBR for setting the PIN to "1" (reading dummy data) or write "1" to the PIN.
		0	0	A serial bus interface circuit loses arbitration when transmitting a slave address or data. And terminates transferring word data.	A serial bus interface circuit is changed to slave mode. To clear AL to "0", read the SBIDBR or write the data to SBIDBR.
	0	1	1/0	In the slave receiver mode, a serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "0" or receives "GENERAL CALL".	Read the SBIDBR for setting the PIN to "1" (reading dummy data) or write "1" to the PIN.
		0	1/0	In the slave receiver mode, a serial bus interface circuit terminates receiving of 1-word data.	Set the number of bits in 1-word to the BC and read received data from the SBIDBR.

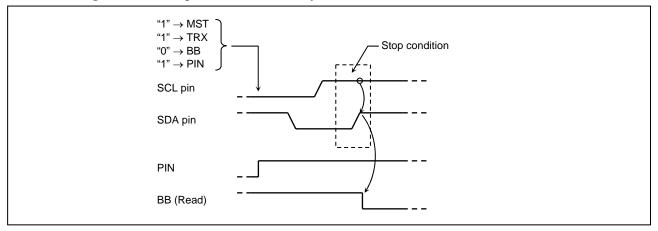
Table 2.9.3	Operation in the Slave Mode
-------------	-----------------------------

Note: In the slave mode, if the slave address set in I2CAR is "0000000B", the TRX changes to "1" by receiving the start byte data "00000001B".

(4) Stop condition generation

When the BB is "1", a sequence of generating a stop condition is started by setting "1" to the MST, TRX, and PIN, and clear "0" to the BB. Do not modify the contents of the MST, TRX, BB, PIN until a stop condition is generated on a bus.

When a SCL line on a bus is pulled-down by other devices, a serial bus interface circuit generates a stop condition after they release a SCL line.





(5) Restart

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart a serial bus interface circuit.

Clear "0" to the MST, TRX and BB and set "1" to the PIN. The SDA pin retains the high-level and the SCL pin is released. Since a stop condition is not generated on a bus, a bus is assumed to be in a busy state from other devices. Test the BB until it becomes "0" to check that the SCL pin a serial bus interface circuit is released. Test the LRB until it becomes "1" to check that the SCL line on a bus is not pulled-down to the low-level by other devices. After confirming that a bus stays in a free state, generate a start condition with procedure (2).

In order to meet setup time when restarting, take at least 4.7 μ s of waiting time by software from the time of restarting to confirm that a bus is free until the time to generate a start condition.

Note: When restarting after receiving in master receiver mode, because the divice doesn't send an acknowledgement as a last data, the level of SCL line can not be conrirmied by reading LRB. Therefore, confirm the status of SCL line by reading P5PRD register.

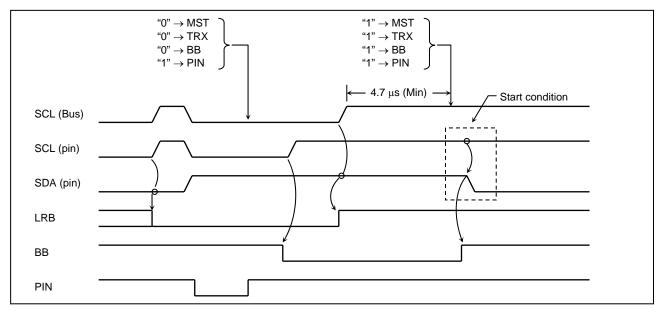


Figure 2.9.18 Timing Diagram when Restarting

2.9.9 Clocked-synchronous 8-Bit SIO Mode Control

The following registers are used to control the serial bus interface (SBI) and monitor the operation in the clocked-synchronous 8-bit SIO mode.

CDICDA	Interface	Control Register A	3 2 1 0		
SBICRA (00020H)	· · · · · ·	6 5 4 SIOINH SIOM "C		(Initial value: 0000 *00	0)
	SIOS	Indicate transfer start/stop	0: Stop 1: Start		
	SIOINH	Continue/abort transfer	0: Continue transfer1: Abort transfer (automatically cleared after abort)		
	SIOM	Transfer mode select	 00: 8-bit transmit mode 01: reserved 10: 8-bit transmit/receive mode 11: 8-bit receive mode 		
	SCK	Serial clock selection (At fc = 16 MHz, Output on \overline{SC} pin)	DV1CK = 0 000: 1000.0 kHz 001: 500.0 kHz 010: 250.0 kHz 011: 125.0 kHz 100: 62.5 kHz 101: 31.2 kHz 110: 15.6 kHz 111: External clock (Input from SCK pin)	DV1CK = 1 000: 500.0 kHz 001: 250.0 kHz 010: 125.0 kHz 011: 62.5 kHz 100: 31.2 kHz 101: 15.6 kHz 110: 7.8 kHz 111: External clock (Input from SCK pin)	Write only
Serial Bus SBIDBR (00021H)	Note 2: C Note 3: S m Interface 1 7 Note1 : T	BICRA is write-only register and nanipulation, etc. Data Register 6 5 4 3 , , , , , , , , , , , , , , , , , , ,	SIOINH to "1" when setting the tra d cannot be used with any of rea	d-modify-write instructions such (Initial value: **** ****) a write buffer and a read but	R/W ffer are
	Note 2: *: [•			
Serial Bus SBICRB (00023H)	Interface	Control Register B 6 5 4 3 "0" - "0" - "1"	3 2 1 0 SBIM SWRST1 SWRST0	(Initial value: **** 000	0)
	SBIM Serial bus interface operation mode selection		 00: Port mode (serial bus interface output disable) 01: SIO mode 10: I²C bus mode 11: reserved 		Write
	SBIM	-	10: I ² C bus mode		
	SBIM SWRST1 SWRST0	-	10: I ² C bus mode 11: reserved	vriting "10" and next writing "01"	

Figure 2.9.19 Control Register/Data Buffer Register/Status Register in SIO Mode (1)

TOSHIBA

Serial Bus I SBISRA		atus Register	A 4 3	2 1 0		
(00020H)				SWR MON (Initial value:	(Initial value: **** ***1)	
	SWRMON Software reset monitor			0: During software reset 1: – (Initial)	Read only	
		atus Register				
SBISRB (00023H)		6 5 '1" "1"	4 3 "1" SIOF	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		
	SIOF	Serial transfer of monitor	perating status	0: Transfer terminated 1: Transfer in process	Read	
	SEF	Shift operating	status monitor	0: Shift operation terminated 1: Shift operation in process	only	
	Note: Set b	it 7 to 4, bit 1 and	bit 0 in SBISR	B to "1".		

Figure 2.9.20 Control Register/Data Buffer Register/Status Register in SIO Mode (2)

- (1) Serial clock
- a. Clock source

The SCK (bits 2 to 0 in SBICRA) is used to select the following functions.

1. Internal clock

In an internal clock mode, any of seven frequencies can be selected. The serial clock is output to the outside on the $\overline{\text{SCK}}$ pin. The $\overline{\text{SCK}}$ pin becomes a high-level when data transfer starts. When writing (in the transmit mode) or reading (in the receive mode) data cannot follow the serial clock rate, an automatic-wait function is executed to stop the serial clock automatically and hold the next shift operation until reading or writing is complete.

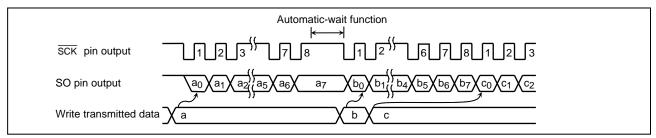
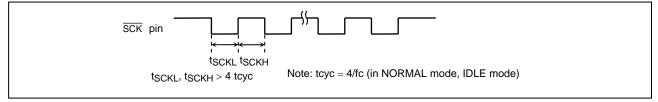
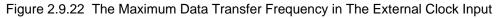


Figure 2.9.21 Automatic Wait Function

2. External (SCK = "111")

An external clock supplied to the $\overline{\text{SCK}}$ pin is used as the serial clock. In order to ensure shift operation, a pulse width of at least 4-machine cycles is required for both high and low levels in the serial clock. The maximum data transfer frequency is 500 KHz (fc = 16.0 MHz).





b. Shift edge

The leading edge is used to transmit data, and the trailing edge is used to receive data.

1. Leading edge

Data is shifted on the leading edge of the serial clock (at a falling edge of the $\overline{\text{SCK}}$ pin input/output).

2. Trailing edge

Data is shifted on the trailing edge of the serial clock (at a rising edge of the \overline{SCK} pin input/output).

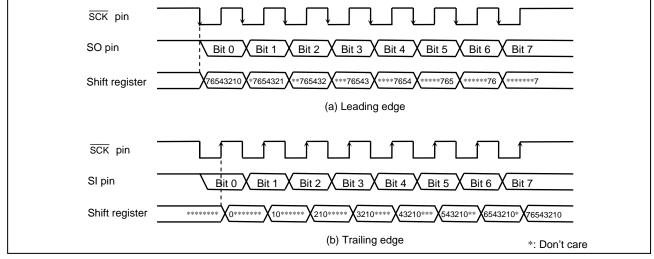


Figure 2.9.23 Shift Edge

(2) Transfer mode

The SIOM (bits 5 and 4 in SBICRA) is used to select a transmit, receive, or transmit/receive mode.

a. 8-bit transmit mode

Set a control register to a transmit mode and write transmit data to the SBIDBR.

After the transmit data is written, set the SIOS to "1" to start data transfer. The transmitted data is transferred from the SBIDBR to the shift register and output to the SO pin in synchronous with the serial clock, starting from the least significant bit (LSB). When the transmit data is transferred to the shift register, the SBIDBR becomes empty. The INTSBI (buffer empty) interrupt request is generated to request new data.

When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated if new data is not loaded to the data buffer register after the specified 8-bit data is transmitted. When transmit new data is written, automatic-wait function is canceled.

When the external clock is used, data should be written to the SBIDBR before new data is shifted.

The SO pin is "1" from the time transmission starts until the first data bit is sent. When SIOF becomes "0", the shift register is cleared. So, output of an undefined value is not prevented at the start of the next transmission.

The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written to the SBIDBR by the interrupt service program. Transmitting data is ended by cleaning the SIOS to "0" by the buffer empty interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, the transmitted mode ends when all data is output. In order to confirm if data is surely transmitted by the program, set the SIOF (bit 3 in the SBISRB) to be sensed. The SIOF is cleared to "0" when transmitting is complete. When the SIOINH is set, transmitting data stops. The SIOF turns "0".

When the external clock is used, it is also necessary to clear the SIOS to "0" before new data is shifted; otherwise, dummy data is transmitted and operation ends.

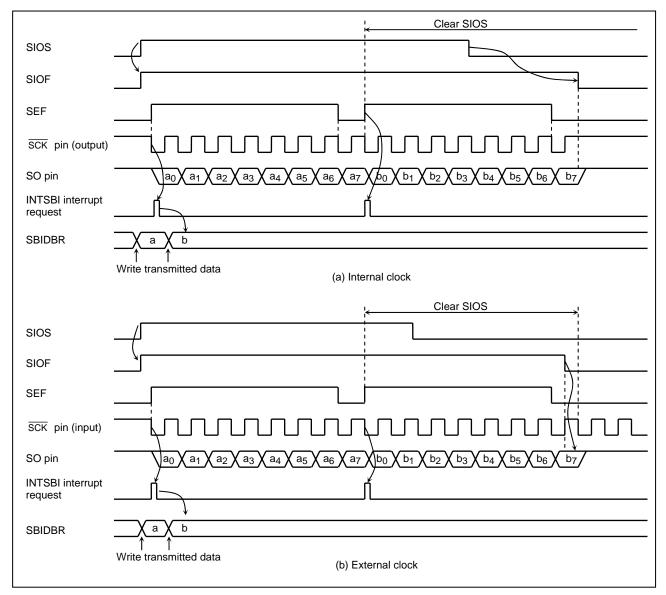


Figure 2.9.24 Transfer Mode

Example: Program to stop transmitting data. (When external clock is used) STEST1: TEST (SBISRB) . SEF ; If SEF = 1 then loop JRS F. STEST1 STEST2: TEST If $\overline{\text{SCK}} = 0$ then loop (P5). 3 JRS T. STEST2 LD (SBICRA), 00000111B ; SIOS $\leftarrow 0$ SCK pin SIOF SO pin Bit 6 Bit 7 tSODH = Min 3.5/fc [s] (In normal mode, IDLE mode)

Figure 2.9.25 Transmitted Data Hold Time at End of Transmit

b. 8-bit receive mode

Set a control register to a receive mode and the SIOS to "1" for switching to a receive mode.

Data is received from the SI pin to the shift register in synchronous with the serial clock, starting from the least significant bit (LSB). When the 8-bit data is received, the data is transferred from the shift register to the SBIDBR. The INTSBI (buffer full) interrupt request is generated to request of reading the received data. The data is read from the SBIDBR by the interrupt service program.

When the external clock is used, since shift operation is synchronized with the clock pulse provided externally, the received data should be read from SBIDBR before next serial clock is input. If the received data is not read, further data to be received is canceled.

When the internal clock is used, the automatic wait function is executed until received data is read from SBIDBR.

The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when received data is read.

Received data disappears if this data is not completely read before reception of the next data terminates. In this case, the next data received is read.

Receiving data is ended by clearing the SIOS to "0" by the buffer full interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, received data is transferred to the SBIDBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm if data is surely received by the program, set the SIOF (bit 3 in SBIDBR) to be sensed. The SIOF is cleared to "0" when receiving is complete. After confirming that receiving has ended, the last data is read. When the SIOINH is set, receiving data stops. The SIOF turns "0" (the received data becomes invalid, therefore no need to read it).

Note: When the transfer mode is switched, the SBIDBR contents are lost. In case that the mode needs to be switched, receiving data is concluded by clearing the SIOS to "0", read the last data, and then switch the mode.

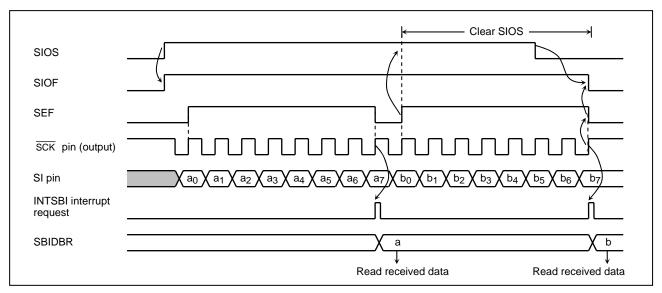


Figure 2.9.26 Receive Mode (Example: Internal clock)

c. 8-bit transmit/receive mode

Set a control register to a transmit/receive mode and write data to the SBIDBR. After the data is written, set the SIOS to "1" to start transmitting/receiving. When transmitting, the data is output from the SO pin on the leading edges in synchronous with the serial clock, starting from the least significant bit (LSB). When receiving, the data is input to the SI pin on the trailing edges of the serial clock. 8-bit data is transferred from the shift register to the SBIDBR, and the INTSBI interrupt request occurs. The interrupt service program reads the received data from the data buffer register and writes data to be transmitted. The SBIDBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

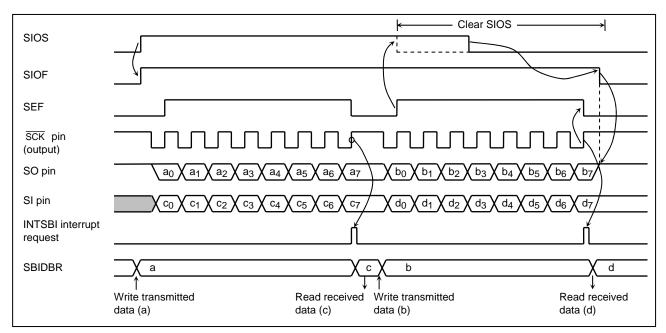
When the internal clock is used, automatic-wait function is initiated until received data is read and next data is written.

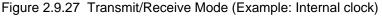
When the external clock is used, since the shift operation is synchronized with the external clock, received data is read and transmitted data is written before new shift operation is executed. The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when received data is read and transmitted data is written.

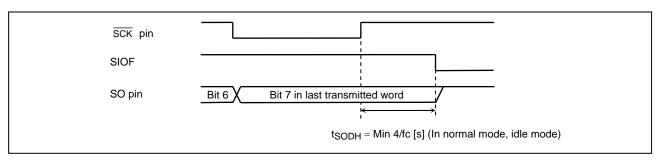
When transmission starts, a value which is the same as the last bit of previously transmitted data is output from the time SIOF is set to "1" until the falling edge of $\overline{\text{SCK}}$ occurs.

Transmitting/receiving data is ended by cleaning the SIOS to "0" by the INTSBI interrupt service program or setting the SIONH to "1". When the SIOS is cleared, received data is transferred to the SBIDBR in complete blocks. The transmit/receive mode ends when the transfer is complete. In order to confirm if data is surely transmitted/received by the program, set the SIOF (bit 3 in SBISRB) to be sensed. The SIOF becomes "0" after transmitting/receiving is complete. When the SIONH is set, transmitting/receiving data stops. The SIOF turns "0".

Note: When the transfer mode is switched, the SBIDBR contents are lost. In case that the mode needs to be switched, conclude transmitting/receiving data by clearing the SIOS to "0", read the last data, and then switch the transfer mode.









2.10 Remote Control Signal Preprocessor/External Interrupt 3 Input Pin

The remote control signal waveform can be determined by inputting the remote control signal waveform from which the carrier wave was eliminated by the receive circuit to P30 (INT3/RXIN) pin. When the remote control signal preprocessor/external interrupt 3 pin is also used as the P30 port, set the P30 port output latch to "1". When it is not used as the remote control signal preprocessor/external interrupt 3 input pin, it can be used for normal port.

2.10.1 Configuration

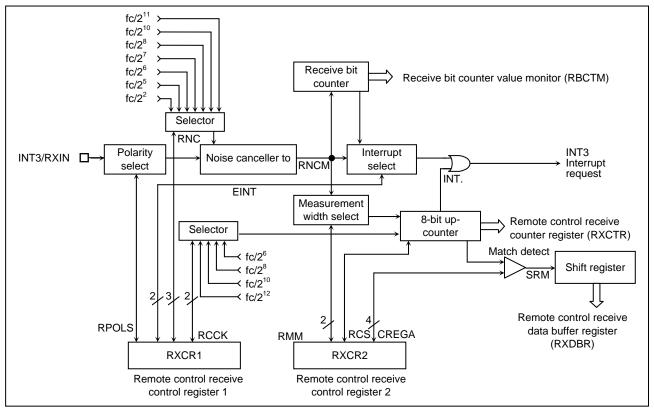


Figure 2.10.1 Remote Control Signal Preprocessor

2.10.2 Remote Control Signal Preprocessor Control

When the remote control signal preprocessor is used, operating states are controlled and monitored by the following registers. Interrupt requests also use the remote control signal preprocessor/external interrupt 3 input pin.

Remote control receive control register 1 (RXCR1)

Remote control receive control register 2 (RXCR2)

Remote control receive counter register (RXCTR)

Remote control receive data buffer register (RXDBR)

Remote control receive status register (RXSR)

When this pin is used for the external interrupt 3 input, set EINT in RXCR1 to other than "11".

Remote cor	ntrol recei	ve control register 1					
RXCR1	7	6 5 4 3	2 1 0				
(00FE8H)	RCCI	K RPOLS EINT	RNC (Initial value: 0000 000	0)			
	RCCK	8-bit up-counter source clock select	00: fc/2 ⁶ (Hz) 01: fc/2 ⁸ 10: fc/2 ¹⁰ 11: fc/2 ¹²				
	RPOLS	Remote control signal polarity select	0: Positive 1: Negative				
	EINT	Interrupt source select	00: Rising edge 01: Falling edge (at RPOLS = 0) 10: Rising/Falling edge 11: 8-bit receive end	R/W			
	RNC	Noise canceler noise eliminating time select	001: $2^2/fc \times 7 - 1/fc$ (s) 010: $2^5/fc \times 7 - 1/fc$ 011: $2^6/fc \times 7 - 1/fc$ 100: $2^7/fc \times 7 - 1/fc$ 101: $2^8/fc \times 7 - 1/fc$ 110: $2^{10}/fc \times 7 - 1/fc$ 111: $2^{11}/fc \times 7 - 1/fc$ 000: Noise canceler disable				
RXCR2	Note 2: A	dge and measurement data, use El ve control register 2 6 5 4 3	2 1 0	nterrupt			
(00FE9H)		CREGA RCS	RMCEN RMM (Initial value: 0000 000	0)			
	CREGA	Setting of detect time for match with 8-bit up-counter upper 4 bits	$ \begin{array}{l} \mbox{Match detect time (Tth) = 16 \times CREGA/RCCK [s]} \\ \mbox{CREGA = 0H to FH} \\ \mbox{Example: } CREGA = 2H, RCCK = fc/2^6 [Hz], at fc = 16 \mbox{ MHz}, \\ \mbox{DV1CK = 0} \\ \mbox{Tth = 128 } [\mu s] \end{array} $				
	RCS	8-bit up-counter start control	0: Stop and counter clear 1: Start				
	RMCEN	Remote control signal preprocesser Enable/Disable	0: Disable 1: Enable				
	RMM	Measurement mode select (invalid when EINT = "10")	00: 01: Refer to Table 2.10.1 10: 11:				
	Note 2: W	: High-frequency clock [Hz] /hen an interrupt source is set f eparately. et CREGA (0H to FH) before EINT	for rising/falling edge, low and high widths are forcibly me sets to 8-bit receive end.	easured			

Figure 2.10.2 Remote Control Receive Control Register 1, 2

TOSHIBA

101100001	ve coun	iter reg	ister						
7	6	5	4	3	2	1	0	Read Only	
								(Initial value: 0000 0000)
			•						
7	6	5	4	3	2	1	0	Read Only	`
	I	I.						(Initial value: 0000 0000)
rol recei	ve statu	is regis	ster						
7	6	5	4	3	2	1	0	Read Only	
	RBÇT	M			OVFF	SRM	RNCM	(Initial value: 0000 *000)
RBCTM	Receive monitor	bit coun	ter value						
OVFF	8-bit up-	counter	overflow	flag	0: No overflow 1: Overflow				
SPM	Data buf	ffer regis	ter input		0: Uppe	er 4 bits o	of 8-bit up-c	counter < CREGA	Read only
monitor					1: Uppe	er 4 bits o	of 8-bit up-c	counter \geq CREGA	,
RNCM		ssing thre	0						
	rol receir 7 rol receir 7 RBCTM OVFF SRM	rol receive data 7 6 rol receive statu 7 6 RBCTM Receive monitor OVFF 8-bit up- SRM Data but monitor Recote Receive	rol receive data buffer 7 6 5 rol receive status regis 7 6 5 rol receive status regis 7 6 5 RBCTM RBCTM RBCTM RECTM Receive bit count monitor OVFF 8-bit up-counter SRM Data buffer regis monitor Remote control s after passing through	rol receive data buffer registe 7 6 5 4 rol receive status register 7 6 5 4 RBCTM RBCTM RBCTM RBCTM RBCTM RBCTM Receive bit counter value monitor OVFF 8-bit up-counter overflow SRM Data buffer register input monitor Remote control signal mo after passing through nois	rol receive data buffer register 7 6 5 4 3 rol receive status register 7 6 5 4 3 rol receive status register 7 6 5 4 3 RBCTM RBCTM RBCTM Receive bit counter value monitor OVFF 8-bit up-counter overflow flag SRM Data buffer register input monitor Remote control signal monitor after passing through noise	rol receive data buffer register 7 6 5 4 3 2 rol receive status register 7 6 5 4 3 2 rol receive status register 7 6 5 4 3 2 OVFF RBCTM OVFF RBCTM Receive bit counter value monitor 0: No o 1: Over SRM Data buffer register input monitor 1: Upper Receive bit counter signal monitor 1: Upper RNCM Remote control signal monitor after passing through noise	rol receive data buffer register 7 6 5 4 3 2 1 rol receive status register 7 6 5 4 3 2 1 rol receive status register 7 6 5 4 3 2 1 OVFF SRM RBCTM Receive bit counter value monitor OVFF 8-bit up-counter overflow flag SRM Data buffer register input monitor 0: Upper 4 bits conter 4 bits	rol receive data buffer register 7 6 5 4 3 2 1 0 rol receive status register 7 6 5 4 3 2 1 0 rol receive status register 7 6 5 4 3 2 1 0 RBCTM OVFF SRM RNCM RBCTM Receive bit counter value monitor 0 OVFF 8-bit up-counter overflow flag 0: No overflow 1: Overflow SRM Data buffer register input 0: Upper 4 bits of 8-bit up-counter Receive bit counter value 0: Upper 4 bits of 8-bit up-counter RNCM Remote control signal monitor RNCM Remote control signal monitor RNCM Remote control signal monitor RNCM Remote control signal monitor	rol rol

Figure 2.10.3 Remote Control Receive Counter Register, Data Buffer Register, Status Register

			i interrupt Source and Mea	
RPOLS	EINT	RMM	Interrupt source	Measurement mode
	00	00 10		
	00	11		
		01		
0	01	10		
		11		
	10	-		
	11	00	Receive end	
		10		$\rightarrow \ \ \leftarrow$
		00		
	00	10		
		11		
		01		
1	01	10		
		11		
	10	-		
	11	00	Receive end	
	11	10		

Table 2.10.1 Combination of Interrupt Source and Measurement Mode

2.10.3 Noise Elimination Time Setting

The remote control receive circuit has a noise canceler. By setting RNC in RXCR1, input signals shorter than the fixed time can be eliminated as noise.

			······j(······)	
RNC	Minimur	n signal pulse width	Maximum noise	e width to be eliminated
000		-		_
001	(2 ⁵ + 5)/fc	(2.31 μs)	$(2^2 \times 7 - 1)/fc$	(1.69 μs)
010	(2 ⁸ + 5)/fc	(16.31 μs)	$(2^{5} \times 7 - 1)/fc$	(13.88 μs)
011	(2 ⁹ + 5)/fc	(32.31 μs)	$(2^6 imes 7 - 1)/fc$	(27.88 μs)
100	(2 ¹⁰ + 5)/fc	(64.31 μs)	$(2^7 \times 7 - 1)/fc$	(55.88 μs)
101	(2 ¹¹ + 5)/fc	(128.3 μs)	$(2^8 imes 7 - 1)/fc$	(111.9 μs)
110	(2 ¹³ + 5)/fc	(512.3 μs)	$(2^{10} \times 7 - 1)/fc$	(447.9 μs)
111	(2 ¹⁴ + 5)/fc	(1.024 ms)	$(2^{11} \times 7 - 1)/fc$	(895.9 μs)

Table 2.10.2 Noise Elimination Time Setting (fc = 16 MHz)

2.10.4 Operation

(1) Interrupts at rising, falling, or rising/falling edge, and measurement modes

First set EINT and RMM. Next, set RCS to "1"; the 8-bit up-counter is counted up by the internal clock. After measurement, the 8-bit up-counter value is saved in RXCTR. Then, the 8-bit up-counter is cleared, an INT3 request is generated, and the 8-bit up-counter resumes counting.

If the 8-bit up-counter overflows (FFH) before measurement is completed, an INT3 request is generated and the overflow flag (OVFF) is set to "1". Then, the 8-bit up-counter is cleared. An overflow can be detected by reading OVFF by the interrupt processing. To restart the 8-bit up-counter, set RCS to "1".

Setting RCS to "1" zero-clears OVFF.

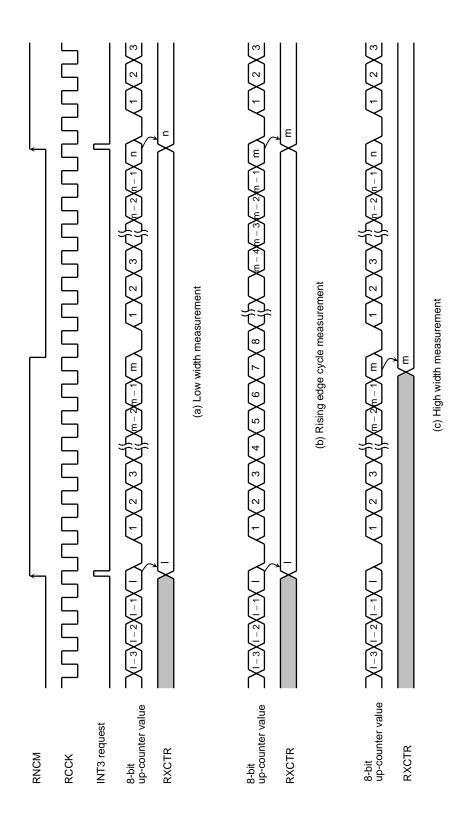


Figure 2.10.4 Rising Edge Interrupt Timing Chart (RPOLS = 0)

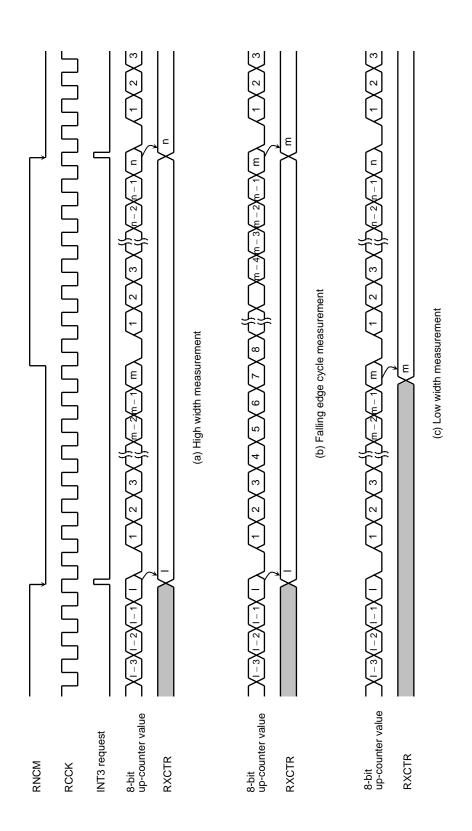


Figure 2.10.5 Falling Edge Interrupt Timing Chart (RPOLS = 0)

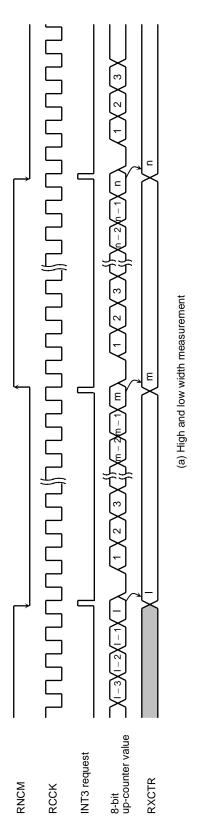


Figure 2.10.6 Rising/Falling Edge Interrupt Timing Chart

(2) 8-bit receive end interrupts and measurement modes

By determining one-cycle remote control signal as one-bit data set to "0" or one-pulse width remote control signal as one-bit data set to "1", an INT3 request is generated after 8-bit data is received. When "0" is determined, this means the upper four bits in the 8-bit up-counter have not reached the CREGA value. When "1" is determined, this means the upper four bits in the 8-bit up-counter have reached or exceeded the CREGA value. The 8-bit up-counter value is saved in RXCTR after one bit is determined. The determined data is saved, bit by bit, in RXDBR at the rising edge of the remote control signal (when RPOLS = 1, falling edge). The number of bits saved in RXDBR is counted by the receive bit counter and saved in RBCTM. RBCTM is set to "0001B" at the rising edge of the input (when RPOLS = 1, falling edge) after the INT3 request is generated.

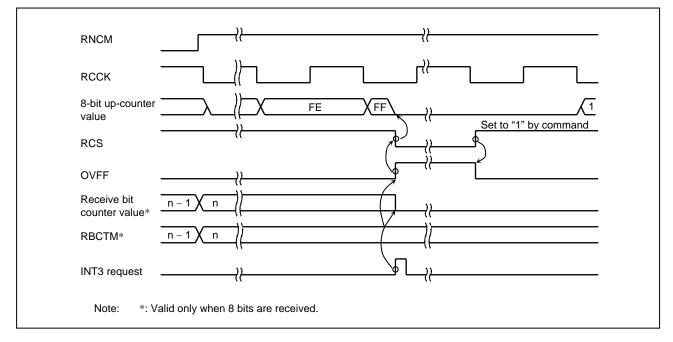


Figure 2.10.7 Overflow Interrupt Timing Chart

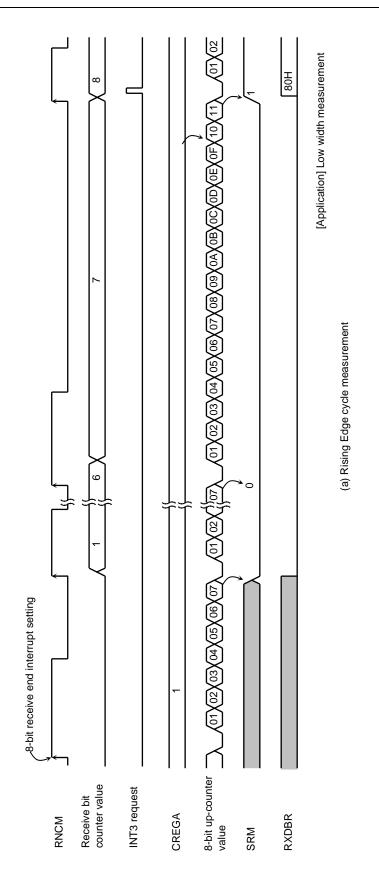


Figure 2.10.8 8-Bit Receive End Interrupt Timing Chart (RPOLS = 0)

Count clock (RCCK)	Resolution [µs]	Maximum setting time [ms]
00	4	1.024
01	16	4.096
10	64	16.38
11	256	65.53

Table 2.10.3 Count Clock for Remote Control Preprocessor Circuit (at fc = 16 MHz)

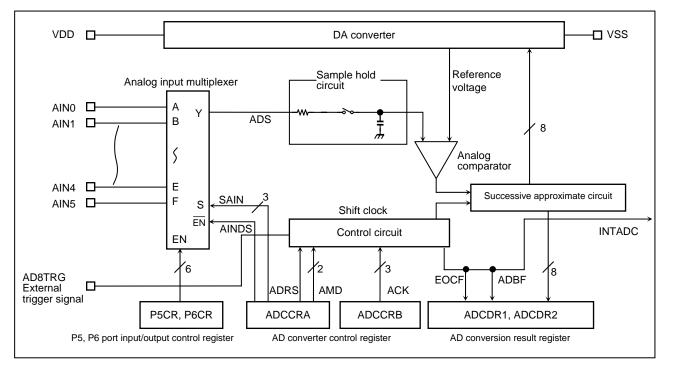
2.11 8-Bit AD Converter (ADC)

The TMP88CS34/CP34 has a 8-bit successive approximation type AD converter.

Figure 2.11.1 shows the circuit configuration of the AD converter.

The AD converter includes control registers ADCCRA and ADCCRB, conversion result registers ADCDR1 and ADCDR2, a DA converter, a sample hold circuit, a comparator, and sequential transducer circuit.

To use P5 and P6 as analog inputs, clear the output latch for P5 and P6 to "0". Also, clear the input/output control registers (P5CR1 and P6CR) to "0". P63 to P61 output "0" after a reset. When these dual-function pins are used as ports, be sure to set ORP6S2 to "1".



2.11.1 Configuration

Figure 2.11.1 AD Converter (ADC)

2.11.2 Control Register

The following register are used foe AD converter.

- AD converter control register 1 (ADCCRA)
- AD converter control register 2 (ADCCRB)
- AD conversion result register (ADCDR1/ADCDR2)
- (1) AD converter control register 1 (ADCCRA)

ADCCRA control AD conversion start, AD operation mode select, analog input control and analog input channel select.

(2) AD converter control register 2 (ADCCRB)

ADCCRB control AD conversion time select.

(3) AD conversion result register (ADCDR1)

AD conversion result is stored after end of conversion.

(4) AD conversion result register (ADCDR2)

For monitoring status of conversion. Figure 2.11.2 and Figure 2.11.3 show AD converter control register.

ADCCRA (0000EH)	7 ADRS	6 5 AMD	4 AINDS	3 "0"	2 1 SA			(Initial value:	0001 0000	0				
(0000211)	ABRO		AINDO	Ľ	, 07				0001 0000	·)				
					The ADRS bit is automatically cleared after starting AD conversion.									
	ADRS	S AD conversion start			During AD conversion, setting ADRS to "1" initializes the ADRS bit and resets conversion.									
				0:	-									
					AD conversion	on restart								
				01.	00: STOP mode 01: Software start mode									
	AMD	AD Operating	mode select	10:	10: Trigger start mode									
					reserved Analog input	enable				R/W				
	AINDS	Analog input c	control	1:	Analog input	disable								
					Selects AIN0 Selects AIN1									
				Selects AIN2										
	SAIN	Analog input c	hannel selec	t l	Selects AIN3									
					Selects AIN4 Selects AIN5									
			110:											
	Nata 1:	Calant an alam inn		111:										
		Select analog inp When the analog			•	S should be	set to "1"							
		During conversio	•		•			for all of the	pins.					
		-							P					
		And port near to analog input, do not input intense signaling of change. 4: The ADRS is automatically cleared to "0" after starting conversion.												
	Note 5:	Always set bit 3 in	n ADCCRA te	o "0".										
	Note 6:	Do not set ADRS	(bit 7 in ADC	CRA) to "	1" during AD	conversion.	Re-set it af	ter confirmin	g with EOCF	f (bit 5				
		in ADCDR2) that	t the convers	Note 6: Do not set ADRS (bit 7 in ADCCRA) to "1" during AD conversion. Re-set it after confirming with EOCF (bit in ADCDR2) that the conversion is completed or after generating an interrupt signal (INTADC) (by the										
	interrupt processing routine or the like).													
			-	the like).										
	Note 7	In the trigger mod	le, the system	[.] the like). n does not	accept the se	econd and s	ubsequent	triggers after	accepting th	ne first				
	Note 7	In the trigger mod trigger for starting	le, the system g AD conversion	the like). does not ion. To res	accept the se start AD conve	econd and s ersion by a	ubsequent trigger, set <i>i</i>	triggers after AMD (bits 6 a	accepting th	ne first				
	Note 7	In the trigger mod trigger for starting to "00" and then p	le, the system g AD conversion put the system	the like). does not ion. To res m in trigge	accept the se start AD conve r start mode	econd and s ersion by a again (with	ubsequent trigger, set / AMD = "10	triggers after AMD (bits 6 a ").	accepting th and 5 in ADC	ne first				
	Note 7 Note 8:	In the trigger mod trigger for starting	le, the system g AD conversion put the system n enters STO	the like). n does not ion. To res m in trigge P mode, <i>A</i>	accept the se start AD conve start mode AD converter	econd and s ersion by a again (with control regi	ubsequent trigger, set / AMD = "10	triggers after AMD (bits 6 a ").	accepting th and 5 in ADC	ne first				
D Converter	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist	le, the system g AD conversion put the system n enters STO	the like). n does not ion. To res m in trigge P mode, <i>F</i> ystem ree	accept the se start AD conve start mode AD converter nters NORM	econd and s ersion by a again (with control regi AL mode.	ubsequent trigger, set / AMD = "10	triggers after AMD (bits 6 a ").	accepting th and 5 in ADC	ne first				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5	de, the system g AD conversion put the system n enters STO er after the s 4	the like). n does not ion. To res m in trigge P mode, <i>A</i>	accept the se start AD conve or start mode AD converter nters NORM/ 2 1	econd and s ersion by a again (with control regi AL mode.	trigger, set a AMD = "10 ster 1 (ADC	triggers after AMD (bits 6 a "). CRA) is initia	accepting th and 5 in ADC alized.	ne first CRA)				
	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist	le, the system g AD conversion put the system n enters STO	the like). n does not ion. To res m in trigge P mode, <i>F</i> ystem ree	accept the se start AD conve start mode AD converter nters NORM	econd and s ersion by a again (with control regi AL mode. 0 "0"	trigger, set AMD = "10 ster 1 (ADC	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value:	accepting th and 5 in ADC alized. **0* 000*	ne first CRA)				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5	de, the system g AD conversion put the system n enters STO er after the s 4	the like). n does not ion. To res m in trigge P mode, <i>F</i> ystem ree	accept the sestart AD converter start mode AD converter Inters NORM/ 2 1 ACK _	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0	trigger, set AMD = "10 ster 1 (ADC	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value:	accepting th and 5 in ADC alized. **0* 000*	ne first CRA)				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5	de, the system g AD conversion put the system n enters STO er after the s 4	the like). In does not ion. To res m in trigge P mode, <i>A</i> ystem ree 3 	accept the sestart AD converter start mode AD converter Inters NORM/ 2 1 ACK _	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0	trigger, set AMD = "10 ster 1 (ADC	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value:	accepting th and 5 in ADC alized. **0* 000*	ne first CRA)				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5	de, the system g AD conversion put the system n enters STO er after the s 4	the like). In does not ion. To res m in trigge P mode, <i>A</i> ystem ree	accept the sestart AD converter start mode AD converter Inters NORM/ 2 1 ACK _	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0	trigger, set AMD = "10 ster 1 (ADC	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value:	accepting th and 5 in ADC alized. **0* 000*	ne first CRA)				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5	de, the system g AD conversion put the system enters STO er after the s 4	the like). n does not ion. To res m in trigge P mode, <i>A</i> ystem ree 3 ACK 000	accept the sestart AD converter start mode AD converter Inters NORM/ 2 1 ACK _	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0	trigger, set a AMD = "10 ster 1 (ADC 	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value:	accepting th and 5 in ADC alized. **0* 000*	ne first CRA)				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5	de, the system g AD conversion put the system n enters STO er after the s 4 "1"	the like). n does not ion. To res m in trigge P mode, <i>A</i> ystem ree 3 ACK 000 001	accept the sestart AD converter start mode AD converter Inters NORM/ 2 1 ACK _	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0	trigger, set a AMD = "10 ster 1 (ADC 	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value:	accepting th and 5 in ADC alized. **0* 000*	ne first CRA)				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 "0"	de, the system g AD conversion put the system n enters STO er after the s 4 "1"	the like). n does not ion. To res m in trigge P mode, A ystem ree 3 ACK 000 001 010	accept the se start AD converter of start mode AD converter nters NORM/ 2 1 ACK Conversion time	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0	trigger, set a AMD = "10 ster 1 (ADC CK = 0 fc = 8 MHz Reserved	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value:	accepting thand 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz	ne first CRA)				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 "0"	de, the system g AD conversion put the system n enters STO er after the s 4 "1"	the like). n does not ion. To res m in trigge P mode, <i>A</i> ystem ree 3 ACK 000 001 010 011	accept the se start AD conver er start mode AD converter inters NORM/ 2 1 ACK Conversion time	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	AMD = "10 AMD = "10 ster 1 (ADC CK = 0 fc = 8 MHz Reserved	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value: DV10 z fc = 16 MHz –	accepting th and 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz 39	ne first CRA)				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 "0"	de, the system g AD conversion put the system n enters STO er after the s 4 "1"	the like). a does not ion. To res m in trigge P mode, <i>F</i> ystem ree 3 ACK 000 001 010 011 100	accept the se start AD conver- er start mode AD converter inters NORM/ 2 1 ACK Conversion time 156/fc [s] 312/fc [s]	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	trigger, set J AMD = "10" ster 1 (ADC CK = 0 CK = 0 CK = 8 MHz Reserved 19.5 39.0	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value: VIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	accepting thand 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz 39 78	ne first CRA)				
ADCCRB	Note 7 Note 8:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 "0"	de, the system g AD conversion put the system n enters STO er after the s 4 "1"	the like). n does not ion. To res m in trigge P mode, <i>A</i> ystem ree 3 ACK 000 001 010 011 100 101	accept the se start AD converter or start mode AD converter nters NORM/ 2 1 ACK Conversion time 156/fc [s] 312/fc [s] 624/fc [s]	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	trigger, set J AMD = "10" ster 1 (ADC CK = 0 CK = 0 CK = 8 MHz Reserved 19.5 39.0	triggers after AMD (bits 6 a "). CRA) is initia (Initial value: DV10 z fc = 16 MHz 78 156	accepting th and 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz 39 78 156	ne first CRA)				
ADCCRB	Note 7 Note 8: Control Re 7 ACK	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 	de, the system g AD conversion out the system e enters STO er after the s 4 	the like). n does not ion. To res m in trigge P mode, A ystem ree 3 ACK 000 001 010 011 100 101 110 111 above list	accept the sestart AD conver er start mode AD converter inters NORM/ 2 1 ACK Conversion time 156/fc [s] 312/fc [s] 624/fc [s] 1248/fc [s]	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Subsequent trigger, set AMD = "10" ster 1 (ADC	triggers after AMD (bits 6 a "). CRA) is initia (Initial value: DV10 z fc = 16 MHz 78 156	accepting th and 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz 39 78 156	ne first CRA)				
ADCCRB	Note 7 Note 8: Control Re 7 ACK	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 "0" AD conversion Do not use settin Set conversion tim	de, the system g AD conversion put the system n enters STO er after the side 4 "1" for time select g except the me by analog	the like). a does not ion. To res m in trigge P mode, A ystem ree 3 ACK 000 001 010 011 100 101 110 111 above list g reference	accept the sestant AD converter or start mode AD converter inters NORM/ 2 1 ACK Conversion time 156/fc [s] 312/fc [s] 624/fc [s] 1248/fc [s]	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Subsequent trigger, set AMD = "10" ster 1 (ADC	triggers after AMD (bits 6 a "). CRA) is initia (Initial value: DV10 z fc = 16 MHz 78 156	accepting th and 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz 39 78 156	ne first CRA)				
ADCCRB	Note 7 Note 8: Control Re 7 ACK ACK	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 "0" AD conversion Do not use settin Set conversion tin VDD =	de, the system g AD conversion out the system n enters STO er after the s 4 "1" g except the me by analog 4.5 to 5.5 V (the like). a does not ion. To res m in trigge P mode, <i>A</i> ystem ree 3 ACK 000 001 010 011 100 101 100 101 110 111 above list p reference 15.6 μ or	accept the sestart AD converter or start mode AD converter inters NORM/ 2 1 ACK Conversion time 156/fc [s] 312/fc [s] 624/fc [s] 1248/fc [s]	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	aubsequent trigger, set AMD = "10" ster 1 (ADC	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value: DV10 z fc = 16 MHz 78 156	accepting th and 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz 39 78 156	ne first CRA)				
ADCCRB	Note 7 Note 8: Control Re 7 ACK ACK Note 1: Note 1: Note 2: Note 3:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 	de, the system g AD conversion out the system e enters STO er after the signal 4 "1" g except the me by analog 4.5 to 5.5 V (and bit 5 in Al	the like). a does not ion. To res m in trigge P mode, <i>A</i> ystem ree 3 ACK 000 011 100 011 100 101 101 110 111 above list g reference 15.6 μ or DCCRB to	accept the sestart AD converter or start mode AD converter inters NORM/ 2 1 ACK Conversion time 156/fc [s] 312/fc [s] 624/fc [s] 1248/fc [s] e voltage (VD more) 0 "0" and set f	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Subsequent trigger, set / AMD = "10" ster 1 (ADC	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value: V10 z fc = 16 MHz 78 156	accepting th and 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz 39 78 156	ne first CRA)				
ADCCRB	Note 7 Note 8: Control Re 7 ACK ACK Note 1: Note 1: Note 2: Note 3: Note 4:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 "0" AD conversion bo not use settin Set conversion tin VDD = Always set bit 0 a When a read inst	de, the system g AD conversion out the system n enters STO er after the s 4 	the like). a does not ion. To res m in trigge P mode, <i>A</i> ystem ree 3 ACK 000 011 100 011 100 101 101 110 111 above list g reference 15.6 μ or DCCRB to	accept the sestart AD converter or start mode AD converter inters NORM/ 2 1 ACK Conversion time 156/fc [s] 312/fc [s] 624/fc [s] 1248/fc [s] e voltage (VD more) 0 "0" and set f	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Subsequent trigger, set / AMD = "10" ster 1 (ADC	triggers after AMD (bits 6 a "). CCRA) is initia (Initial value: V10 z fc = 16 MHz 78 156	accepting th and 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz 39 78 156	ne first CRA)				
ADCCRB	Note 7 Note 8: Control Re 7 ACK ACK Note 1: Note 1: Note 2: Note 3: Note 4: Note 5:	In the trigger mod trigger for starting to "00" and then p When the system Re-set this regist egister 2 6 5 	de, the system g AD conversion put the system n enters STO er after the side d	the like). a does not ion. To res m in trigge P mode, A ystem ree 3 ACK 000 001 010 011 100 101 110 111 above list g reference 15.6 μ or DCCRB to DCCRB to	accept the sestant AD converter inters NORM/ 2 1 ACK _ Conversion time 156/fc [s] 312/fc [s] 624/fc [s] 1248/fc [s] 1248/fc [s]	econd and s ersion by a again (with control regi AL mode. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	AMD = "10 AMD = "10 ster 1 (ADC CK = 0 fc = 8 MHz Reserved 19.5 39.0 78.0 - Reserved //s. CRB to "1". d in as under	triggers after AMD (bits 6 a "). CRA) is initia (Initial value: DV10 z fc = 16 MHz 78 156 - - 39 78 156	accepting th and 5 in ADC alized. **0* 000* CK = 1 fc = 8 MHz 39 78 156 _	ne first CRA)				

Figure 2.11.2 AD Converter Control Register

AD Conversion	n Result Re	egister								
ADCDR1	7	6	5	4	3	2	1	0		
(00031H)	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00	(Initial value: 0000 0000))
ADCDR2	7	6	5	4	3	2	1	0		
(00032H)			EOCF	ADBF					(Initial value: **00 ****)	
	EOCF	AD co	nversion e	end flag	0: 1:		conversion conversior		conversion	Read
	ADBF	AD co	nversion t	ousy flag	0: 1:	During stop of AD conversion onl During AD conversion				
	Note 1:	The EOC	F is clear	ed to "0" v	when read	ling the A	DCDR1.			
	٦	Therefore	e, the AD	conversio	n result sh	nould be	ead to AD	CDR1 mc	ore first than ADCDR2.	
	Note 2:	ADBF is	set to "1" I	by starting	AD conv	ersion ar	d cleared	to "0" by e	end of AD conversion. Additionally	/,
	/	ADBF is	DBF is cleared to "0" by setting AMD = "00" in ADCCR2 or entering to the STOP mode.							
	Note 3: I	f the pin	is used a	as an ana	og input	pin, reset	the DGIN	IE registe	r to "0" to disable all inputs other	r than
	â	analog in	puts.							

Figure 2.11.3 AD Converter Result Register

2.11.3 AD Converter Operation

The high side of an analog reference voltage is applied to VDD, and the low side is applied to VSS pin. Dividing a reference voltage between VDD and VSS to the voltage corresponding to a bit by a rudder resistance and comparing it with the analog input voltage converts the AD.

Mode	Function
AD converter disable mode	AD converter stop mode. This mode is always used to change modes.
Software start mode	Single AD conversion of 1 channel which specifies input.
Trigger start mode	Single AD conversion of 1 channel which specifies input (AD8TRG) from Key-On-Wake-Up circuit as a trigger.

Table 2.11.1 AD Converter Operation mode

2.11.4 Interrupt

Interrupt request signal occur at the timing when the EOCF bit is set to "1".

2.11.5 AD Converter Operation Modes

When the MCU places in the STOP mode during the AD conversion, the conversion is stopped and the ADCDR2 content becomes indefinite. After returning from the STOP mode, the EOCF and INTADC does not occur. Therefore, the AD conversion must be restarted after returning from the STOP mode.

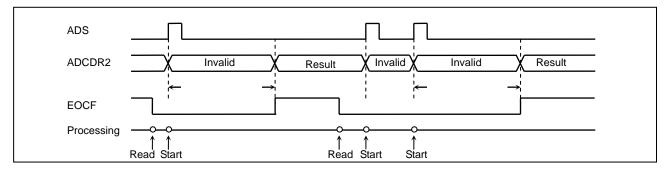


Figure 2.11.4 AD Conversion Timing Chart

(1) AD conversion in STOP mode

When the AD converter stop mode is specified during AD conversion, the AD conversion is stopped immediately. The AD conversion is not implemented, so the undefined value is not written to the AD conversion result register. The AD conversion start commands which occur is the AD converter stop mode are ignored.

This mode is automatically selected by reset.

This mode is used to change the AD converter operation mode.

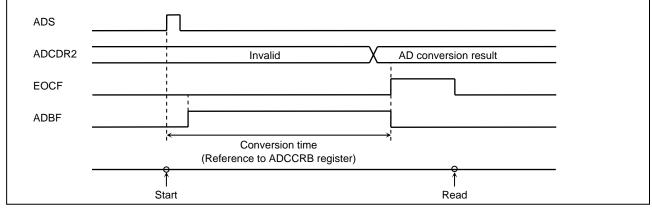
(2) Single mode

When the AMD (bit 6, 5 to in ADCCRA) set to "01", the AD conversion signal mode.

This mode does AD conversion of single channel, and conversion result is stored in ADCDR1. The EOCF (bit 5 in ADCDR2) is set to "1" at end of one conversion, and an interrupt request signal occurs. The EOCF is cleared to "0" by reading the AD conversion registers.

But when the AD conversion is restarted before the ADCDR is read, the EOCF is cleared to "0" and the last AD conversion result is maintained till next conversion end.

Do not set ADRS (bit 7 in ADCCRA) during AD conversion. Again set it after confirming with EOCF (bit 5 in ADCDR2) that the conversion is completed or after generating an interrupt signal (INTADC) (by the interrupt processing routine or the like).





Example: The AD conversion starts after $19.5 \,\mu s$ (at fc = 16 MHz) and AIN4 pin are selected as the conversion time and the analog input channel. Confirming the EOCF, the converted value is read out, and the 8 bits data is stored to address 009EH in RAM. The operation mode is a signal mode. : AIN SELECT

, AIIN	SELEC.	L
LD	(P5), (0000000B

- LD (P5CR1), 0000000B
- (P6), 0000000B LD
- (P6CR), 0000000B LDSelects AIN4, Selects the software start LD (ADCCRA), 00100100B mode LD (ADCCRB), 00011000B ; Selects the conversion time and the operation mode. ; AD CONVERT START SET (ADCCRA).7 ADRS = 1SLOOP: TEST EOCF = 1? (ADCCR2).5 JRS T. SLOOP ; RESULT DATA READ (9EH), (ADCDR1)
- (3) Trigger start mode

LD

The AD conversion of a specified single channel is executed when input (AD8TRG) from Key-On-Wake-Up circuit is set as trigger, the conversion result is stored in the ADCDR1.

The EOCF (bit 5 in ADCDR2) is set to "1" at end of one conversion, and an interrupt request signal occurs.

It needs to be set the STOP mode by bit 5 to 6 in ADCCRA before the AD conversion is executed again.

2.11.6 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 8-bit digital value converted by the AD as shown in Figure 2.11.6.

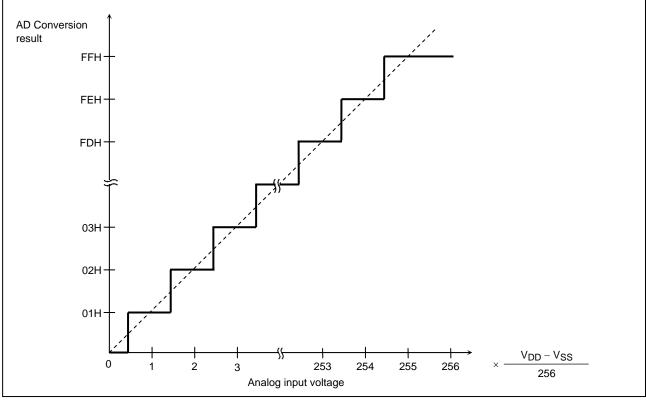


Figure 2.11.6 Analog Input Voltage and AD Conversion Result (typ.)

2.11.7 STOP Modes during AD Conversion

When standby mode (STOP mode) is entered forcibly during AD conversion, the AD convert operation is suspended and the AD converter is initialized. (ADCCRA and ADCCRB are initialized to initial value.) Also, the conversion result is indeterminate. (Conversion results up to the previous operation are cleared, so be sure to read the conversion results before entering standby mode.) When restored from standby mode, AD conversion is not automatically restarted, so it is necessary to restart AD conversion after setting ADCCRA and ADCCRB. Note that since the analog reference voltage is automatically disconnected, there is no possibility of current flowing into the analog reference voltage.

2.11.8 Notice of AD converter

(1) Analog input voltage range

Voltage range of analog input (AIN0 to AIN5) must be forced from V_{SS} to V_{DD} . If input voltage of which out of range is forced to analog input pin, AD conversion result to unknown. Also, this cause other analog input pin unstable.

(2) I/O port with analog input

Analog input pins (AIN0 to AIN5) are also I/O port. During AD conversion using any analog input pin, don't operate other I/O port with analog input. Because, AD accuracy would be worse. Also, other electrically swinging port without analog input may cause noise to near analog input pin.

(3) Reduce to noise

Figure 2.11.7 is shown as internal equivalent circuit of analog input pin.

Increasing output impedance of analog input supply, cause noise or other non-good condition.

Therefore, output impedance of analog input supply must be less than $5k\Omega$.

And we recommend to connect capacitance to analog input pin.

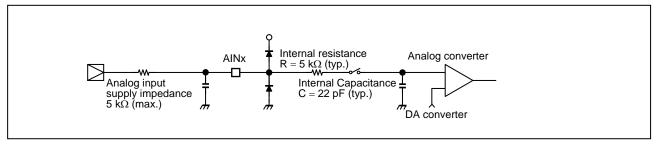


Figure 2.11.7 Analog Input Equivalent Circuit and Analog Input Pin

2.12 Key-On-Wake-Up

In this MCU the IDLE mode is also released by Low active port inputs. The low input voltage is regulated higher than the other normal ports. Therefore the ports can be enabled by analog input level.

2.12.1 Configuration

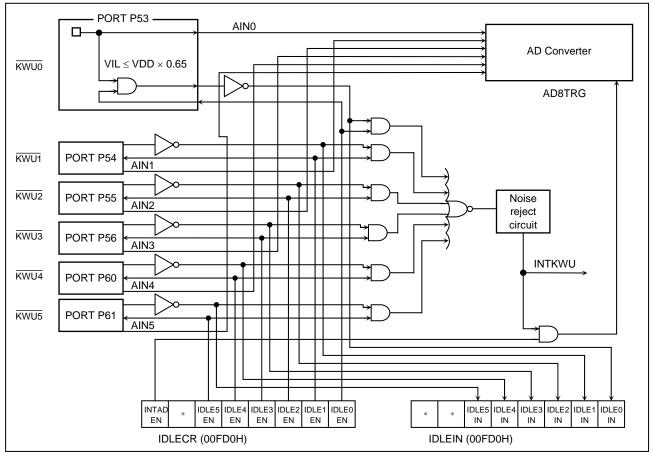


Figure 2.12.1 Key-On-Wake-Up Control Circuit

2.12.2 Control

P53 to P56 and P60, P61 ports can be controlled by IDLE control register (IDLECR).

It can be configured as enable/disable in one-bit unit. When those pins are used by IDLE mode release, those pins must be set input mode (P5CR1, P5, P6CR, P6, ADCCRA).

IDLE mode is controlled by system control register 2 (SYSCR2) and maskable interrupts. After the individual enable flag (EF5) is set to "1", the IDLE mode must starts. When enabled port input generates INTKWU interrupt, the IDLE mode is released. Low level input voltage in those ports is regulated to less than VDD \times 0.65 (V).

IDLE port monitorring register (IDLEIN) can be used to check state of ports.

INTADEN can enable to generate AD8TRG, which is used as trigger of AD converter trigger start mode.

Noise reject circuit eliminate noise, which is less than $24 \ \mu s$ period.

IDLECR	7	6		5	4	3	2	1	0		
(00FD0H)	INTAD EN	*		IDLE5 EN	IDLE4 EN	IDLE3 EN	IDLE2 EN	IDLE1 EN	IDLE0 EN	(Initial value: 0*00 0000))
	INTADE	N (Gener	ation of	AD8TRG		0: 1:	Disable Enable			
	IDLE5E	N F	Releas	se IDLE	mode by	KWU5	0: 1:	Disable Enable			
	IDLE4E	N F	Releas	se IDLE	mode by	KWU4	0: 1:	Disable Enable			
	IDLE3E	N F	Releas	se IDLE	mode by	KWU3	0: 1:	Disable Enable			Write only
	IDLE2E	N F	Releas	se IDLE	mode by	KWU2	0: 1:	Disable Enable			
	IDLE1E	IN F	Releas	se IDLE	mode by	KWU1	0: 1:	Disable Enable			
	IDLE0E	N F	Releas	se IDLE	mode by	KWU0	0:	Disable Enable			
							1.	Enable			
	Note :	*: Do	n't cai	re			1.	ETIADIe			
DLE port I		ng re	giste	ər	4	3	2		0		
IDLEIN	monitorrir		giste		4 IDLE4 IN	3 IDLE3 IN	1	1 IDLE1 IN	0 IDLE0 IN	(Initial value: **00 0000))
IDLEIN	monitorrir 7	ng re 6 *	giste	er 5 IDLE5	IDLE4 IN	IDLE3	2 IDLE2 IN 0:	1 IDLE1 IN "0" detect	IDLE0 IN	(Initial value: **00 0000))
•	monitorrir 7 *	ng re 6 *	nput I	er 5 IDLE5 IN	IDLE4 IN KWU5	IDLE3	2 IDLE2 IN	1 IDLE1 IN	IDLE0 IN	(Initial value: **00 0000))
IDLEIN	monitorrir 7 * IDLE5II	ng re 6 * N I	nput I	er 5 IDLE5 IN evel of	IDLE4 IN KWU5 KWU4	IDLE3	2 IDLE2 IN 0: 1: 0:	1 IDLE1 IN "0" detect "1" detect "0" detect	IDLE0 IN	(Initial value: **00 0000)) Read
IDLEIN	monitorrir 7 * IDLE5II IDLE4II	ng re 6 × N I N I	nput l	er 5 IDLE5 IN evel of	IDLE4 IN KWU5 KWU4 KWU3	IDLE3	2 IDLE2 IN 0: 1: 0: 1: 0: 0:	1 IDLE1 IN "0" detect "1" detect "1" detect "0" detect "0" detect	IDLE0 IN	(Initial value: **00 0000)	
IDLEIN	monitorrir 7 * IDLE5II IDLE4II IDLE3II	ng re 6 × N I N I N I	nput I nput I	Er 5 IDLE5 IN level of level of	IDLE4 IN KWU5 KWU4 KWU3 KWU2	IDLE3	2 IDLE2 IN 0: 1: 0: 1: 0: 1: 0: 1: 0: 0:	1 IDLE1 IN "0" detect "1" detect "1" detect "0" detect "1" detect "1" detect	IDLE0 IN	(Initial value: **00 0000)	Read

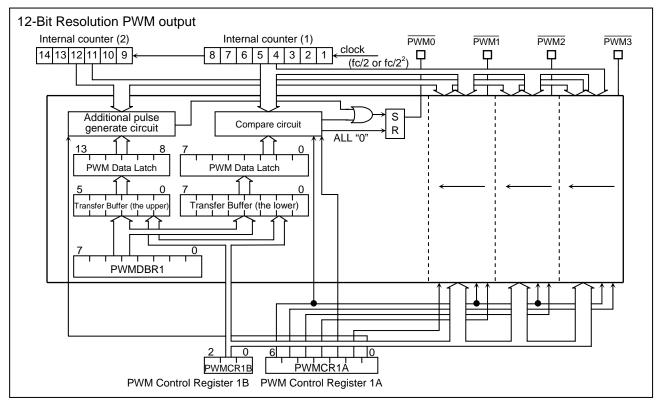
Figure 2.12.2 Key-On-Wake-Up Control Register

2.13 Pulse Width Modulation Circuit Output

The TMP88CS34/CP34 has four 12-bit resolution PWM output channels including two 14-bit resolution selectable.

DA converter output can easily be obtained by connecting an external low-pass filter. PWM outputs are multiplexed with general purpose I/O ports as; P40 ($\overline{PWM0}$) to P43 ($\overline{PWM3}$). PWM output is negative logic. When these ports are used PWM outputs, the corresponding bits of P4, P5 output latches and input/output control latches should be set to "1".

In STOP mode, PWM output pin keeps high-level. When operation mode is changed from STOP mode to NORMAL mode, PWM control register (PWMCR1A, PWMCR1B) are initialized.



2.13.1 Configuration

Figure 2.13.1 PWM Output Circuit

2.13.2 PWM Output Wave Form

(1) $\overline{PWM0}$ to $\overline{PWM1}$ Outputs

PWM0 and PWM1 output can be selected 12-bit or 14-bit resolution PWM outputs.

1. 12-bit Resolution PWM Output

When these are used as 12-bit PWM output, one period is $T_M = 2^{13}/\text{fc}$ [s] (When DV1CK = 0) and $T_M = 2^{14}/\text{fc}$ [s] (When DV1CK = 1) and sub-period is $T_S = T_M/16$.

The lower 8-bit of the PWM data latch controls the low level pulse width with a cycle of T_S. The lower 8-bit of the PWM data latch is n (n = 1 to 255), the low level pulse width with a cycle becomes n x t₀ [s] (t₀ = 2/fc [s] when DV1CK = 0, t₀ = 4/fc [s] when DV1CK = 1).

The upper 4-bit of the PWM data latch controls a position to output the additional pulses. When the upper 4-bit of the PWM data latch is m, the additional pulses are generated in each of m periods out of 16 periods contained in a T_M period.

The relationship between the 4-bit data and the position of TS period where the additional pulses are generated is shown in Table 2.13.1.

	Bit positi	on of the lowe	er 4 bits of PV	VMDRxH	Relative position of T_{S} in T_{M} period where the additional		
	Bit 11	Bit 10	Bit 9	Bit 8	pulse is generated. (Number of $T_{S(I)}$ is listed)		
a)	0	0	0	0	No additional pulse		
b)	0	0	0	1	8		
c)	0	0	1	0	4, 12		
d)	0	1	0	0	2, 6, 10, 14		
e)	1	0	0	0	1, 3, 5, 7, 9, 11, 13, 15		

Table 2.13.1 The addition pulse (12 bit mode)

Note 1: The bit positions of a) to e) can be combined.

Note 2: If the low order eight bits for the PWM data latch are set to "FFH", be sure to set the high order four bits for this latch to "00H".

2. 14-bit Resolution PWM Output

When these are used as 14-bit PWM output, one period is $T_M = 2^{15}/\text{fc}$ [s] (When DV1CK = 0) and $T_M = 2^{16}/\text{fc}$ [s] (When DV1CK = 1) and sub-period is $T_S = T_M/64$.

The lower 8-bit of the PWM data latch controls the low level pulse width with a cycle of T_S. The lower 8-bit of the PWM data latch is n (n = 1 to 255), the low level pulse width with a cycle becomes n x t₀ [s] (t₀ = 2/fc [s] when DV1CK = 0, t₀ = 4/fc [s] when DV1CK = 1).

The upper 6-bit of the PWM data latch controls a position to output the additional pulses. When the upper 6-bit of the PWM data latch is m, the additional pulses are generated in each of m periods out of 64 periods contained in a T_M period.

The relationship between the 6-bit data and the position of T_S period where the additional pulses are generated is shown in Table 2.13.2.

-							
	Bit p	osition of	the lowe	er 6 bits o	of PWMD	RxH	Relative position of T_S in T_M period where the additional
	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	pulse is generated. (Number of $T_{S(I)}$ is listed)
a)	0	0	0	0	0	0	No additional pulse
b)	0	0	0	0	0	1	32
c)	0	0	0	0	1	0	16, 48
d)	0	0	0	1	0	0	8, 24, 40, 56
e)	0	0	1	0	0	0	4, 12, 20, 28, 36, 44, 52, 60
f)	0	1	0	0	0	0	2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62
g)	1	0	0	0	0	0	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,
							35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63

Table 2.13.2 The addition pulse (14 bit mode)

Note 1: The bit positions of a) to g) can be combined.

Note 2: If the low order eight bits for the PWM data latch are set to "FFH", be sure to set the high order six bits for this latch to "00H".

(2) $\overline{PWM2}$ to $\overline{PWM3}$ Outputs

 $\overline{PWM2}$ and $\overline{PWM3}$ output are 12-bit resolution PWM outputs.

One period is T_M = 2^{13} /fc [s] (When DV1CK = 0) and T_M = 2^{14} /fc [s] (When DV1CK = 1) and sub-period is T_S = T_M /16.

The lower 8-bit of the PWM data latch controls the low level pulse width with a cycle of Ts. The lower 8-bit of the PWM data latch is n (n = 1 to 255), the low level pulse width with a cycle becomes n x t₀ [s] (t₀ = 2/fc [s] when DV1CK = 0, t₀ = 4/fc [s] when DV1CK = 1).

The upper 4-bit of the PWM data latch controls a position to output the additional pulses. When the upper 4-bit of the PWM data latch is m, the additional pulses are generated in each of m periods out of 16 periods contained in a T_M period.

The relationship between the 4-bit data and the position of $T_{\rm S}$ period where the additional pulses are generated is shown in Table 2.13.1.

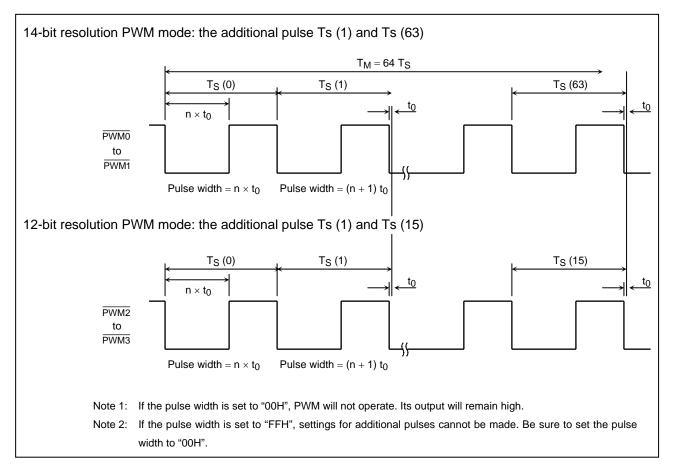


Figure 2.13.2 PWM Output Wave Form

2.13.3 Control

PWM output is controlled by PWM Control Register (PWMCR1A, PWMCR1B) and PWM Data Buffer Register (PWMDBR1).

PWM Contr	ol Register 1			
PWMCR1A	7 6	5 4 3	2 1 0 RESOLUTION (Initial value: *000 0000)	1
(00028H)	– ABOR	T1 START3 START2 START1	STARTO 1 , 0	
	ABORT1	Abort PWM operation of channel 3 to 0	0: Operation1: PWM Abort (PWM outputs are fixed to a high-level.)	
	START3	Start channel 3	0: Stop PWM3 1: Start PWM3	
	START2	Start channel 2	0: Stop PWM2 1: Start PWM2	
	START1	Start channel 1	0: Stop PWM1 1: Start PWM1	Write only
	START0	Start channel 0	0: Stop PWM0 1: Start PWM0	
	RESOLUTION	1 Select channel 1 resolution	0: 14-bit resolution 1: 12-bit resolution	
	RESOLUTION	2 Select channel 0 resolution	0: 14-bit resolution 1: 12-bit resolution	
	Note 3: PWM0	set the ABORT1 to "1", the ABOR	RT1 is cleared to "0" automatically. annot be used with any of the read-modify-write instructions	such
PWM Contr	ol Register 1			
PWMCR1B (00029H)	7 6	5 4 3	2 1 0 PWMCHS1 PWMHL (Initial value: **** *000)	
	PWMCHS1	Select the PWM data latch of 12-bit PWM channels	00: Channel 0 01: Channel 1 10: Channel 2 11: Channel 3	Write only
	PWMHL	Select upper or lower data transfer buffer (PWMDBR1)	0: Lower 8-bit 1: Upper 4-bit or 6-bit	
			annot be used with any of the read-modify-write instructions	such
PWM Data	Buffer Regist	er 1		
PWMDBR1 (0002AH)	7 6	5 4 3	2 1 0 Write only (Initial value: 0000 0000))
	as SE	Γ, CLR, etc. operation mode is changed from	annot be used with any of the read-modify-write instructions	

Figure 2.13.3 PWM Control Register 1A/1B and PWM Data Buffer Register 1

Binary Cou	nter Co	ntrol Re	egister										
	7	6	5	4	3		2		1	 0		(Initial value: 0000 0000)	
CGCR (00030H)	"0"	"0"	DV1CK	"0"	"0"		"0"		"0"	 "0"			
	DV	1CK	Select of i 1st divider	•	ck to			fc/4 fc/8					R/W
		*: Don't The all b	care bits except [OV1CK	are clea	ared to	o "0".						

Figure 2.13.4 DIVIDER Control Register

(1) Internal Counter

The internal counter of PWM outputs is a free running counter. The all bits of counter are set to "1" and are not counted up at one of the following conditions.

- 1. During reset
- 2. The operation mode is changed to STOP mode.
- 3. Setting ABORT1 to "1".
- 4. The START3 to 0 are "0" in 12-bit PWM outputs.
- 5. The lower 8-bit of PWM data latch in 12-bit PWM outputs is "00H". The PWM data latch in 7-bit PWM outputs is "00H".
- (2) Outputs control and Programming of PWM data

The PWM outputs are fixed to a high-level immediately when the ABORT1 is set to "1". The PWM outputs starts the operation when the STARTx (x: 0 to 3) is set to "1".

The data from the transfer buffer to a PWM data latch is transferred when the all bits of internal counter are set to "1". Therefore, the data is transferred to a PWM data latch immediately when the internal counter is initialized. And the data is transferred to a PWM data latch at the beginning of the next cycle when all bits of the internal counter are not set to "1".

The sequence of writing the output data to PWM data latches is shown as follows;

- 1. $\overline{PWM0}$ to $\overline{PWM1}$
 - a. Write the channel number of PWM data latch to PWMCHS1 (bit 2 and 1 in PWMCR1B) and clear PWMHL (bit 0 in PWMCR1B) to "0".
 - b. Write the lower 8-bit PWM output data to PWMDBR1.
 - c. Write the channel number of PWM data latch to PWMCHS1 and set PWMHL to "1".
 - d. Write the upper 4-bit or 6-bit PWM output data to PWMDBR1.
 - e. Select the resolution of PWM output to RESOLUTIONx (x: 0, 1) (bit 0 and 1 in PWMCR1A) and set STARTx (x: 0, 1) (bit 2 and 3 in PWMCR1B) to "1".
 - Note: PWM output data must be write to PWMDBR1 in the order of the lower 8-bit PWM output data, the upper 4-bit (or 6-bit) PWM output data. If the upper 4-bit (or 6-bit) PWM output data is write to PWMDBR1, the lower 8-bit PWM output data is not changed (Except when lower 8-bit PWM output data is "00H".).

- 2. $\overline{\text{PWM2}}$ to $\overline{\text{PWM3}}$
 - a. Write the channel number of PWM data latch to PWMCHS1 and clear PWMHL to "0".
 - b. Write the lower 8-bit PWM output data to PWMDBR1.
 - c. Write the channel number of PWM data latch to PWMCHS1 and set PWMHL to "1".
 - d. Write the upper 4-bit PWM output data to PWMDBR1.
 - e. Set STARTx (x: 2, 3) to "1".

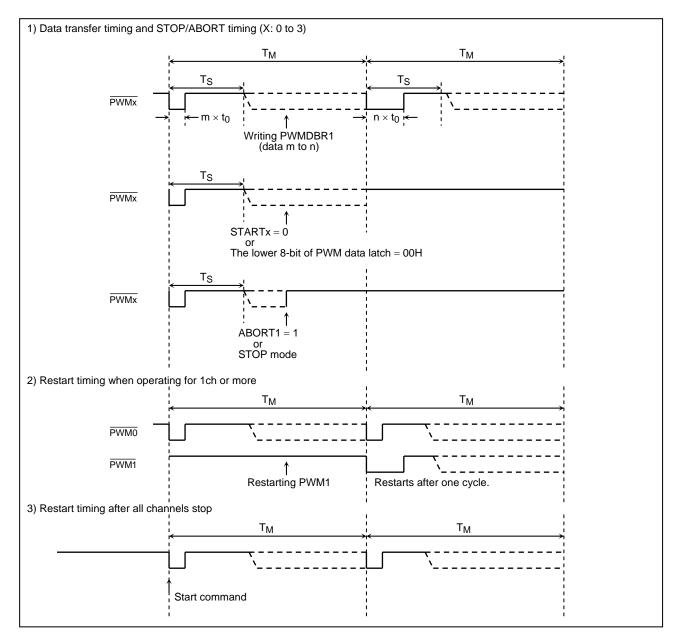


Figure 2.13.5 Wave form of PWM0 to PWM3

Note: PWM output data must be write to PWMDBR1 in the order of the lower 8-bit PWM output data, the upper 4-bit (or 6-bit) PWM output data. If the upper 4-bit (or 6-bit) PWM output data is write to PWMDBR1, the lower 8-bit PWM output data is not changed (Except when lower 8-bit PWM output data is "00H".).

Example: At fc = 16 MHz, DV1CK = 0

 $\overline{PWM0}\,$ pin outputs a 14-bit resolution PWM wave form with a low-level of 32 μs width and no additional pulse.

 $\overline{PWM1}\,$ pin outputs a 12-bit resolution PWM wave form with a low-level of 16 μs width and no additional pulse.

LD	(CGCR), 00H	;	DV1CK = 0
			Select the lower 8-bit of $\overline{\text{PWM0}}$ output data latch
	(PWMDBR1),80H		
LD	(PWMCR1B),01H	;	Select the upper 6-bit of PWM0 output data latch
			No additional pulse = 00H
			Select the lower 8-bit of $\overline{\text{PWM0}}$ output data latch
	(PWMDBR1),40H		
			Select the upper 4-bit of PWM0 output data latch
			Additional pulse (Ts (8)) = 01H
LD	(PWMCR1A),0DH	;	Start $\overline{\text{PWM0}}$ and $\overline{\text{PWM1}}$,
			$\overline{\text{PWM0}}$: 14-bit resolution, $\overline{\text{PWM1}}$: 12-bit resolution

2.14 On-Screen Display (OSD) Circuit

The TMP88CS34/CP34 features a built-in on-screen display circuit used to display characters and symbols on the TV screen. There are 383 characters of mono font and 96 characters of color font (447 characters of mono font and 64 characters of color font) and any characters can be displayed in an area of 32 columns \times 12 lines (include 2 columns for solid space). With an OSD interrupt, additional lines can be displayed.

OSD circuit functions are as follows :

JOD (ficult functions are as follows.		
(1)	Number of character fonts	:	mono font 383 and color font 96 mono font 447 and color font 64
(2)	Number of display characters	:	384 (32 columns × 12 lines).
(3)	Composition of character	:	horizontal $16 \times$ vertical 18 dots
(4)	Character sizes	:	3 kinds for large, middle and small characters (Selectable line by line)
(5)	Character ornamentation func	tion	
	Fringing function	:	mono font
	Smoothing function	:	mono font
	Slant function (Italics)	:	mono font
	Blinking function		
	Underline		
(6)	Solid space		
(7)	Area plane function	:	2 planes
(8)	Full-raster blanking function		
(9)	Display colors		
	Character colors : 8 or	27 c	colors (selectable character by character)
	8		colors (selectable page by page)
	_		colors (selectable page by page)
	-		colors (selectable each of 2 planes)
	Raster color : 8 or	27 c	colors (selectable page by page)
(10)	1 0 1		orizontal steps and 625 vertical steps for code plane orizontal steps and 625 vertical steps for Area plane
(11)	Window function : 62	$25 \mathrm{v}$	ertical steps
(12)	Half transparency output func	tion	
(13)	27 colors display function		
(14)	Color palette		
	PAL100/NTSC120 display		
,	· · · · · · · · · · · · · · · · · · ·		

Note: The function of the OSD circuit don't meet the requirements of on-screen display functions of closed caption decoders based on FCC standards.

The TMP88CS34/CP34 outputs OSD through 3 planes; code, area, and raster. 3 planes function independently. In addition, they are displayed simultaneously. There is the priority among these 3 planes, so they are displayed on a screen according to the priority.

These 3 planes have the priority such as

Code > Area > Raster.

1. Code plane

OSD character is displayed on the code plane.

The code plane consists of 32 characters \times 1 row and a total of 12 planes. The 12 planes have the priority such as code 1 > code 2 >...> code 11 > code 12.

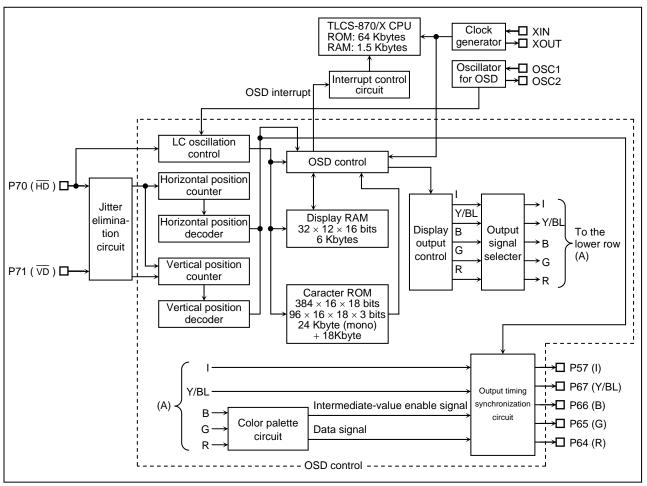
On the code plane, characters of 16×18 dots is displayed. These fonts are called characters, and read from character ROM and display memory through the character code on the display memory.

2. Area plane

The area on a screen is displayed on the area plane.

The area plane can display 2 square areas of any size by specifying coordinates. The 2 planes have the priority such as area plane 1 > area plane 2.

2.14.1 OSD Configuration



Shown below is the block diagram of the OSD circuit.

Figure 2.14.1 OSC Block Diamgram

2.14.2 Monochrome and Color Fonts

The TMP88CS34 can display both monochrome and color fonts.

The monochrome font is intended for monochromic display. Each character in the font consists of 18 vertical \times 16 horizontal dots. For the color font, each display dot in each character can be specified separately for R (red), G (green), and B (blue). Each character consists of 18 vertical \times 16 horizontal dots.

The monochrome and color fonts can be mixed on one display row.

2.14.3 Character ROM and Display Memory

(1) Character ROM

The character ROM incorporates 383 different monochrome font character data items and 96 different color font character data items (447 different monochrome font character data items). Users can define font data.

Each monochrome character ROM data item consists of 16×18 dots. Each monochrome font dot corresponds to one character ROM bit. A value of "1" represents a display state, and a value of "0" represents a non-display state.

Each color font character ROM data item consists of 16×18 dots for red, 16×18 dots for green, and 16×18 dots for blue. Each color font dot corresponds to three character ROM bits (with each bit corresponding to red, green, or blue).

The character ROM start address for each character code is calculated as listed in Table 2.14.1.

Number of usable	character patterns	Usable chara	Register for switching number of		
Monochrome font	Color font	Monochrome font	Color font	fonts, ROMACH (bit 4 in ORDON)	
383	96	1 to 17FH	180H to 1DFH	0	
447	64	1 to 17FH, 1C0H to 1FFH	180H to 1BFH	1	

 Table 2.14.1
 Number of Character Patterns and Character Codes

Table 2 14 2	Monochrome/Color Font Character ROM Start Address

ROMACH	Character ROM start address
	Monochrome font (CRA = 1 to 17FH)
	Character ROM start address = CRA \times 40H + 20000H
0	Color font (CRA = 180H to 1DFH)
0	Character ROM start address for red = CRA \times 40H + 26000H
	Character ROM start address for green = $CRA \times 40H + 27800H$
	Character ROM start address for blue = $CRA \times 40H + 29000H$
	Monochrome font
	Character ROM start address = CRA × 40H + 20000H (CRA = 1 to 17FH)
	Character ROM start address = CRA × 40H + 27000H (CRA = 1C0H to 1DFH)
	Character ROM start address = CRA × 40H + 28C00H (CRA = 1E0H to 1EFH)
1	Character ROM start address = CRA × 40H + 2A400H (CRA = 1F0H to 1FFH)
	Color font (CRA=180H to 1BFH)
	Character ROM start address for red = CRA \times 40H + 26000H
	Character ROM start address for green = CRA \times 40H + 27800H
	Character ROM start address for blue = CRA \times 40H + 29000H

Figure 2.14.2 (a) shows an example of configuring a character font (character code 001H) as well as monochrome font ROM addresses and the related data. Figure 2.14.2 (b) shows a character ROM dump list for this character font (character code 001H).

Figure 2.14.3 (a) shows an example of configuring a character font (character code 180H) as well as color font ROM addresses and the related data. Figure 2.14.4 (b) shows a character ROM dump list for this character font.

- Note 1: A data cannot be read from character ROM by software.
- Note 2: When ordering a mask, load the data to character ROM at addresses 20000H to 2A7FFH. And the data in unused are of character ROM are must be specified to FFH.
- Note 3: Do not use character code 000H

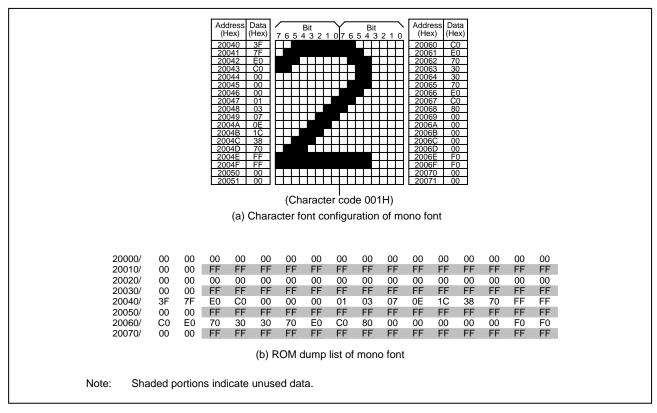


Figure 2.14.2 Character Font Configuration and ROM Dump List

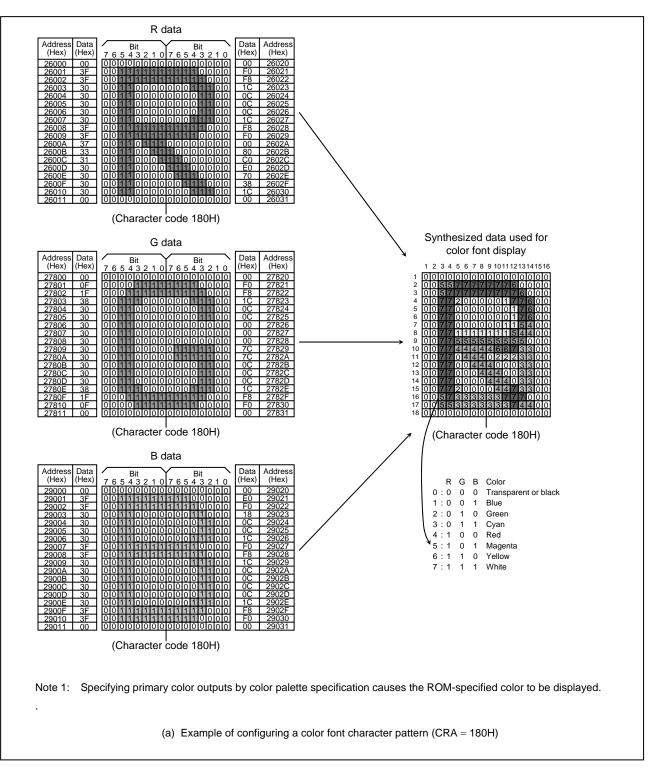


Figure 2.14.3 (1/2)

26000/ 26010/ 26020/	00 30 00	3F 00 F0	3F FF F8	30 FF 1C	30 FF 0C	30 FF 0C	30 FF 0C	30 FF 1C	3F FF F8	3F FF F0	37 FF 00	33 FF 80	31 FF C0	30 FF E0	30 FF 70	30 FF 38
26030/	1C	00	FF													
27800/ 27810/	00 0F	0F 00	1F FF	38 FF	30 FF	38 FF	1F FF									
27820/ 27830/	00 F0	F0 00	F8 FF	1C FF	0C FF	0C FF	00 FF	00 FF	00 FF	7C FF	7C FF	0C FF	0C FF	0C FF	1C FF	F8 FF
29000/	00	3F	3F	30	30	30	30	3F	3F	30	30	30	30	30	30	3F
29010/ 29020/ 20020/	3F 00 F0	00 E0	FF F0 FF	FF 18 FF	FF 0C FF	FF 0C FF	FF 1C FF	FF F0 FF	FF F8 FF	FF 1C FF	FF 0C FF	FF 0C	FF 0C FF	FF 0C	FF 1C	FF F8
29030/	FU	00	FF					mp lis				FF	FF	FF	FF	FF
Note:	Shadi	ing ind	licates	data i	n unu:	sed ar	eas.									

Figure 2.14.4 (2/2)

(2) Display memory

Each character of the 384 characters displayed in 32 columns \times 12 lines consists of 16 bits in the display memory. Five data items are written to the display memory: character code, color data, blinking specification, underline enable, and slant enable.

There are two modes for writing display data to the display memory. One mode is used for writing all display data (character code, color data, blinking specification, underline enable, and slant enable) simultaneously. The other mode is used for changing either character codes or the remaining data items (color data, blinking specification, underline enable, and slant enable). How to write display data to the display memory is described in section 2.14.6.7 (1).

Note: The display memory is in an unknown state at reset.

Display memory configuration

- Character code specification register (9 bits) CRA8 to CRA0
- Color data specification register (4 bits) IDT/RDT/GDT/BDT
- Blinking specification register (1 bit)......BLF
- Underline enable register (1 bit) EUL
- Flag (1 bit) for specifying whether to turn on or off the character-specific background ECBKD

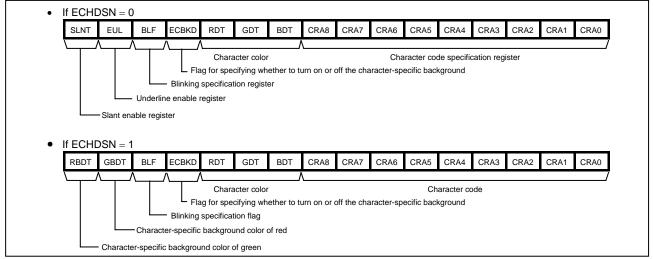
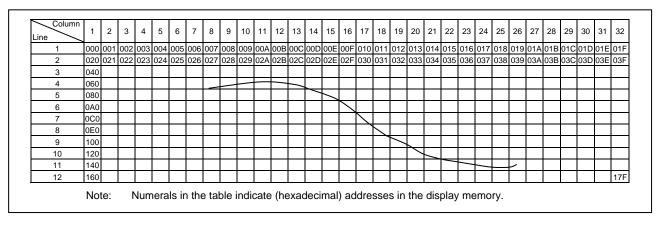


Figure 2.14.5 Display Memory Bit Configuration





(3) Color palette

The color palette can contain eight colors out of 27 colors and the display colors are specified by the color palette registers (ORCPT0-7). The color palette registers (ORCPT0-7) are assigned by the RGB setting register for each display mode (character, background, fringe, area, raster).

• RGB setting register values and their corresponding color palette registers

$\text{RGB} = 000\text{b} \rightarrow \text{ORCPT0}$	$\text{RGB} = 001\text{b} \rightarrow \text{ORCPT1}$
$\text{RGB} = 010\text{b} \rightarrow \text{ORCPT2}$	$\text{RGB} = 011\text{b} \rightarrow \text{ORCPT3}$
$\text{RGB} = 100\text{b} \rightarrow \text{ORCPT4}$	$\text{RGB} = 101\text{b} \rightarrow \text{ORCPT5}$
$\text{RGB} = 110\text{b} \rightarrow \text{ORCPT6}$	$\text{RGB} = 111\text{b} \rightarrow \text{ORCPT7}$

• Configuration of the color palette registers

Register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Name	Audress			ŀ	२	(G	I	В	
ORCPT0	00FC6	CPT1MD1	0 (fixed)	CPT0R1	CPT0R0	CPT0G1	CPT0G0	CPT0B1	CPT0B0	Color palette composition register 0 CPT1MD1: OSD color select register (x = 1, 2) CPT1MD1 = 0: 8–color mode CPT1MD1 = 1: 27–color mode
ORCPT1	00FC7	-	-	CPT1R1	CPT1R0	CPT1G1	CPT1G0	CPT1B1	CPT1B0	Color palette composition register 1
ORCPT2	00FC8	-	-	CPT2R1	CPT2R0	CPT2G1	CPT2G0	CPT2B1	CPT2B0	Color palette composition register 2
ORCPT3	00FC9	-	-	CPT3R1	CPT3R0	CPT3G1	CPT3G0	CPT3B1	CPT3B0	Color palette composition register 3
ORCPT4	00FCA	-	-	CPT4R1	CPT4R0	CPT4G1	CPT4G0	CPT4B1	CPT4B0	Color palette composition register 4
ORCPT5	00FCB	-	-	CPT5R1	CPT5R0	CPT5G1	CPT5G0	CPT5B1	CPT5B0	Color palette composition register 5
ORCPT6	00FCC	-	-	CPT6R1	CPT6R0	CPT6G1	CPT6G0	CPT6B1	CPT6B0	Color palette composition register 6
ORCPT7	00FCD	-	-	CPT7R1	CPT7R0	CPT7G1	CPT7G0	CPT7B1	CPT7B0	Color palette composition register 7

• Color palette setting and output colors

<u>27-color mode (CPT1MD1 = 1) 3-value output</u>

	-		
n = 0 to 7	$\mathbf{x} = \mathbf{R}$	$\mathbf{x} = \mathbf{G}$	$\mathbf{x} = \mathbf{B}$
CPTnx1/CPTnx0 = 1/1	Bright red	Bright green	Bright blue
CPTnx1/CPTnx0 = 1/0 or 0/1	Dark red	Dark green	Dark blue
CPTnx1/CPTnx0 = 0/0	No output	No output	No output
<u>8-color mode (CPT1MD1 = 0) 2-value</u>	output		
n = 0 to 7	$\mathbf{x} = \mathbf{R}$	$\mathbf{x} = \mathbf{G}$	$\mathbf{x} = \mathbf{B}$
CPTnx1/CPTnx0 = 1/1	Bright red	Bright green	Bright blue
CPTnx1/CPTnx0 = 1/0 or 0/1	Bright red	Bright green	Bright blue
CPTnx1/CPTnx0 = 0/0	No output	No output	No output

• Setting the display colors

The color palette registers are assigned by setting RGB data for each display mode. The display colors are then specified in the color palette registers.

Setting the character color to bright red and the background color to dark blue for the code plane.

Setting character c	color: After setting the character code, set ORDSN (RDT = 0, GDT = 1, BDT = 0). (Assign a color palette register.)
	RGB-010b corresponds to color palette register ORCPT2.
	RGB-010b corresponds to color palette register ORCF12.
	To set the character color to bright red, set ORCPT2=00110000b. (Set the display color
	in color palette register.)
 Setting background 	d color: Set background setting register ORBK (0FA5h) (RBDT = 0, GBDT = 0, BBDT = 1).
	(Assign a color palette register.)
	RGB = 001b corresponds to color palette register ORCPT1.
	To set the background color to dark blue, set ORCPT1 = 00000001b. (<u>Set the display</u> <u>color in color palette register.</u>)

(4) Color font

For the color font, the display color (R, G, B) can be specified on a dot-by-dot basis. The size of the color font is 18 dots long by 16 dots wide, which is the same as the size of the normal font (mono font). A dot of the color font is comprised of three bits. Font data is combination of three bits (R, G, B) and they are arranged in the order of R (upper), G (middle), B (lower). The color palette registers are assigned by combining these three bits of data.

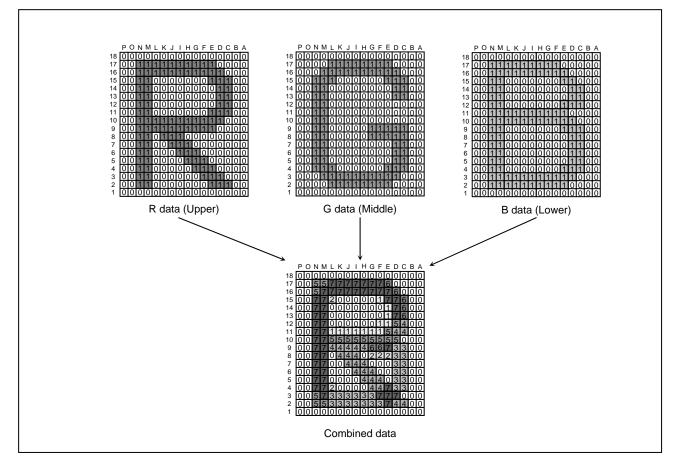


Figure 2.14.7

• Assignment of the color palette registers for the color font

RGB data Color palette register

- RGB = 000b 0 ORCPT0
- RGB = 001b 1 ORCPT1
- RGB = 010b 2 ORCPT2
- RGB = 011b 3 ORCPT3
- RGB = 100b 4 ORCPT4
- RGB = 101b 5 ORCPT5
- $RGB = 110b 6 \quad ORCPT6$
- $RGB = 111b 7 \quad ORCPT7$

ONML D C Р Κ н G F Е В А Output Color Color Palette Setting 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ORCPT0 = 0000000b Black 0 0 ORCPT1 = 00000011b Blue 7 7 0 0 0 5 ORCPT2 = 00001100b Green 0 0 6 0 ORCPT3 = 00110000b Red 0 0 6 0 ORCPT4 = 00001111b Cyan 6 0 ORCPT5 = 00111100b Yellow 0 7 0 0 4 0 ORCPT6 = 00110011b Magenta 0 7 1 1 4 0 ORCPT7 = 00111111b White 0 0 0 7 4 0 3 0 When background = OFF, dots in which is written are 4 4 0 0 3 0 displayed in black. 3 0 When background = ON, dots in which is written 0 4 0 7 4 0 3 0 show no OSD display. 0 7 2 0 0 0 0 4 7 3 0 However, if area-plane data and raste data exist in 3 3 3 3 3 3 7 7 0 0 5 7 0 0 the background of the display these colors are 3 7 4 4 0 0 5 3 3 3 3 3 displayed. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 OSD output waveform when display line 9 R _ G В * Background = ON * Background = OFF Y

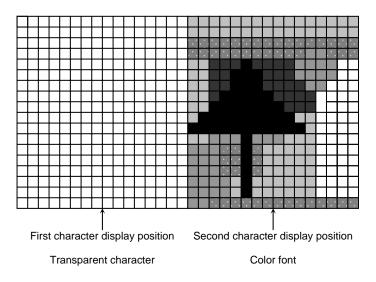
The following shows how the color font shown on the preceding page is displayed by setting the color palette registers.

Figure 2.14.8

Note: Do not use the color font in the first character display position. The color font can be used in the second and subsequent character display positions. If you want to use a color font character in the first character display position as counted from the left side of the TV screen, display a transparent character in the first character display position, and use the color font in the second character display position. Prepare a monochrome font character with no dot as a transparent character. It is recommended that character code CRA = 0x20H be prepared as a transparent character.

Example of display

First character display position: Transparent character. Second character display position: Color font



(5) Dark color setting function

The dark color setting function is intended to control OSD intermediate-value outputs, using High, High-Z, and Low outputs. Setting CPT1MD1 (bit 7 in ORCP1) to "1" enables this function.

Producing 3-value outputs requires installing an external circuit.

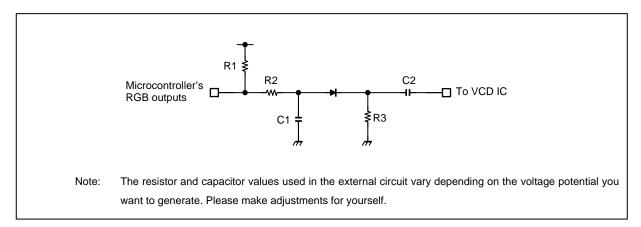


Figure 2.14.9 Example of an External Circuit for Creating Colors between Primary Colors

(6) Switching the OSD ROM area

When the TMP88CS34 is initialized, it is configured for 383 characters of mono font and 96 characters of color font. By setting ROMACH (bit 5 of ORDON) to 1, this configuration can be changed to 447 characters of mono font and 64 characters of color font, as shown below.

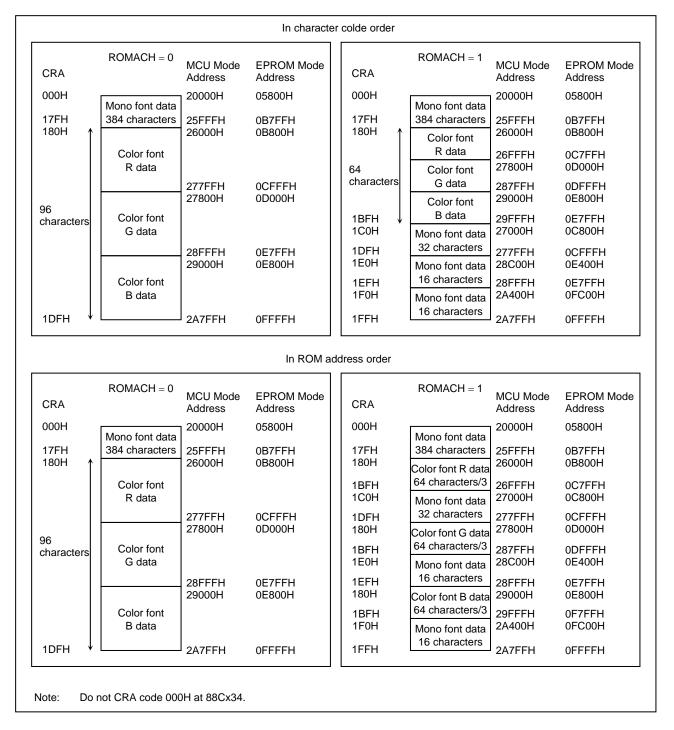


Figure 2.14.10

2.14.4 OSD Circuit Control

The OSD circuit performs control functions using the OSD control registers which reside in addresses 0001DH to 0001FH and 00024H to 00025H in the special function registers (SFR), and in addresses 00F80H to 00FCEH in the data buffer register (DBR). Section 2.14.6.8 shows the OSD control registers. The OSD control registers are used to set display start position, display character designs (that is, fringing, smoothing, color data, character size, and etc.), display memory addresses, and character codes.

Setting the display on-off control bit, DON, (bit 0 in ORDON) to "1" enables display (starts display). Setting DON to "0" disables display (halts display).

2.14.5 OSD Control Register Write

There is a list of the OSD control registers on pages 199 to 201.

When data is written into a shaded register, the data is transferred to the OSD circuit, and then the data becomes valid. After data is written into an unshaded register, the data is transferred to the OSD circuit, and then the data becomes valid.

To transfer the contents of a control register to the OSD circuit, use data transfer request register RGWR (bit 2 in ORDON).

Setting "1" in the RGWR register outputs the transfer request signal to the OSD circuit. Three instruction cycles later, transfer of the written data to the OSD circuit starts. While the data is being transferred, data transfer status monitoring flag RGWR (bit 2 in ORDON) is "1". When this transfer is completed, the flag is cleared to "0".

Written data transfer register (1 bit) … RGWR (Bit 2 in ORDON) "0" … Initialized state "1" Transfers written data to OSD circuit. (After transfer, RGWR is reset to 0.)

Note: Don't write "0" to RGWR.

(1) RGWR system

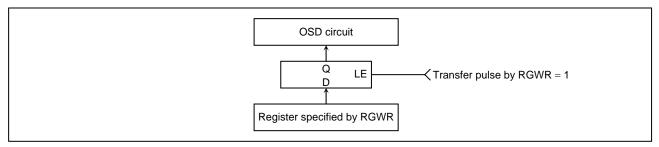


Figure 2.14.11 RGWR System

- (2) Transfer timing
 - 1. No display area

When having set RGWR to "1" during no display area, the timing OSD register can be transferred is at the falling edge of $\overline{\text{HD}}$ signal.

HD		
RGWR Register Data Transfer Pulse	Set RGWR Register to "1"	Clear RGWR Transfer the contents of OSD registers into OSD circuit

Figure 2.14.12 Data Transfer Timing in No Display Area

2. Display area (including any lines specified as display off by character size)

When having set RGWR to "1" during display area, the timing OSD register can be transferred is at the falling edge of $\overline{\text{HD}}$ signal when the display line has been finished.

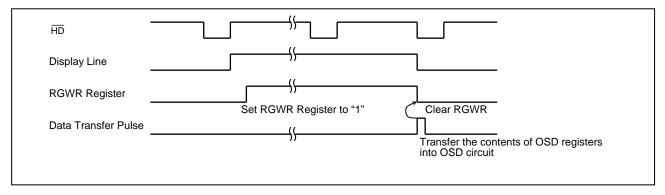


Figure 2.14.13 Data Transfer Timing in Display Area

2.14.6 OSD Function

- 2.14.6.1 Signal Control (Port I/O)
 - (1) P6 port output select function

This function is used to select whether the contents of port P57, P67 to P64 will be output or I, R, G, B, Y/BL signals of the OSD circuit will be output on pins P57, P67 to P64.

P57 port output select registers (1 bits): PIDS (bit 3 in ORP6S)

	PIDS = 0	PIDS = 1	
P57	I	Port	

P67 to P64 port output select registers (4 bits): P67S, P66S, P65S, P64S, (bit 7 to 4 in ORP6S)

	P6nS = 0	P6nS = 1
P64	R	
P65	G	Port
P66	В	TOR
P67	Y/BL	

Note: Be sure to write "0EH" to the ORP6S2 register (0x0FA1H).

(2) OSD pin output polarity control function

This function is used to select the polarity of the OSD outputs for RGB, I and Y/BL.

Output polarity control register (4 bits) \cdots BLIV, YIV, RGBIV, IIV (bit 3 to 0 in ORIV)

"0" ··· Active high "1" ··· Active low

(3) OSD pin input polarity control

Input polarity control

Input polarity control register of RIN/GIN/BIN/Y/BLIN (2 bits)

For Y/BLIN... YBLII (Bit 5 in ORIV)For RIN, GIN, and BIN... RGBII (Bit 4 in ORIV)

Input polarity control

RGBII "0" ··· Active high "1" ··· Active low

Input polarity control register of $\overline{\text{HD}}/\overline{\text{VD}}$ (2 bits)

- For $\overline{\text{VD}}$... VDPOL (Bit 7 in ORIV)
- For $\overline{\text{HD}}$ … HDPOL (Bit 6 in ORIV)

Input polarity control

VDPOL, HDPOL

"0" ... Not invert input signal

"1" ... Invert input signal

Note: To direct P64 (R), P65 (G), and P66 (B) to produce three-value outputs (High, High-Z, and Low), be sure to write "0" to the output polarity control register (4 bits).

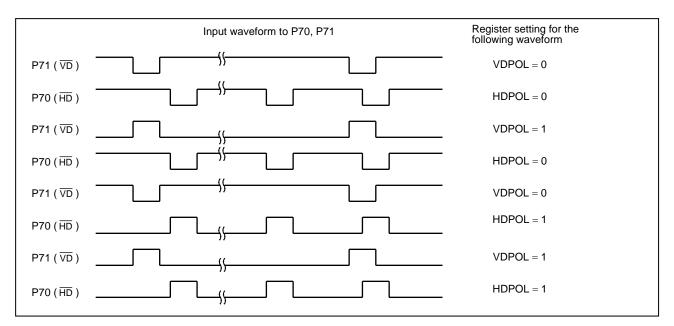


Figure 2.14.14 VD / HD input and VDPOL/HDPOL

(4) Y/BL signal select function

This function is used to select either Y or BL signal output from the Y/BL pin.

Y/BL signal select register (1 bit) ··· YBLCS (bit 7 in ORP6S)

- "0" … Y signal output
 "1" … BL signal output
 Y signal … Output in all OSD areas (Logical OR for R, G, B, Character data, Fringing data, area data, etc.)
 BL signal … When EXBL is "0":
 Output in all display character areas
 When EXBL is "1":
 Output in the whole page
- (5) I signal function select

When PIDS (bit 3 in ORP6S) is set to "0", Port 57 (I pin) can be used as Half Transparency/Half Tone through an extra circuit.

The I-pin output is made high only for the area planes. If you want to make the I-pin output high for area plane 1, set PISEL1 (bit 3 in the ORACL register) to "1". If you want to make the I-pin output high for area plane 2, set PISEL2 (bit 7 in the ORACL register) to "1".

(6) R, G, B, Y/BL Internal/external signal select.

Selects either R, G, B, and Y/BL signals from the internal OSD circuit, or RIN, GIN, BIN, and Y/BLIN signals from external input.

R, G, B, Y/BL signal select registers (2 bits) ··· MPXS1/MPXS0

(Bits 1 and 0 in ORP6S)

"00"	 Simultaneous output (Signal from the OSD circuit has higher priority.)
"01"	 Output of signal from internal OSD circuit
"10"	 Output of signal from external input
"11"	 Simultaneous output (External input signal has higher priority.)

2.14.6.2 OSD data output format control

(1) Scan mode

start position 1 dot height

The double scan mode is used to handle non-interlaced scanning TV. When double scan mode is enabled, the vertical display counter increases every 2 scan lines and a vertical size of a dot is double. This function is enabled by setting VDSMD (bit 7 in ORETC) in the OSD control register to "1".

Scan mode select register (1 bit) \cdots VDSMD (bit 7 in ORETC)

"0" ··· Normal mode

- "1" \cdots Double scan mode
- Note 1: The data written to those control register is transferred to the OSD circuit and become valid when the data is written.
- Note 2: When OSD circuit is used on an interlace scanning TV, a jitter elimination circuit must be enabled and set AFLD to "1" in JECR.

Normal mode height × 2

Table 2.14.3 The Difference of 2 types of Scan Mode					
	Normal mode Double scan mode				
Specification Unit of vertical display	One scanning line	Two scanning lines			

Normal mode	Double scan mode	Normal mode	Double scan mode
Interlac	Interlace scanning		ace scanning



- 2.14.6.3 Display Position Control
 - (1) Code display position setting
 - 1. Horizontal display start position

The horizontal display start position can be set in 256 steps by writing to OSD control registers HS17 to HS10 (bit 7 to 0 in ORHS1). The value is in common with all lines.

Specification unit: 2 T_{OSC} Specification steps: 256 Specification horizontal display start position: Line 1 to 12: HS17 to HS10 (ORHS1)

 $HS1 = (HS17 \text{ to } HS10) \text{ H} \times 2T_{OSC} + 22T_{OSC} \text{ (Line1 to } 12)$

Note 1: T_{OSC}; One cycle of OSD oscillation.

Note 2: The data written to these control registers is transmitted to OSD circuit by setting RGWR (bit 2 in ORDON) to "1".

2. Vertical display start position

The vertical display start position can be specified for each display line using 625 steps by writing to VSn9 to VSn0 (in ORVSn (n; 1 to 12)).

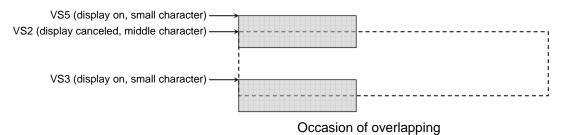
Specification unit: 1 scan line Specification steps: 512 Specification vertical display start position: Line1: VS19 to VS10 (ORVS 1) Line2: VS29 to VS20 (ORVS 2) ... Line12: VS129 to VS120 (ORVS 12) Line n: VSn = (VSn9 to VSn0) H × 1THD (n; 1 to 12)

Note 1: T_{HD} ; One cycle of \overline{HD} signal.

- Note 2: The data written to these control registers is transmitted to OSD circuit by setting RGWR (bit 2 in ORDON) to "1".
- Note 3: If display lines are overlapped each other, previous display line is enabled and next line is disabled. If vertical display start positions of two or more lines are set on same value, high priority line is enabled. Lines of OSD (VS1 to VS12) are fixed priority levels as follows:

$VS1 > VS2 > VS3 > \cdots > VS12$

Set the vertical display start position not to overlap display lines.



- Note 4: The line which is displayed off is managed as a small size character line.
- Note 5: Transfer the contents of vertical display start position registers into OSD circuit before the position of the scanning line coincides with their own vertical display start position.

(2) Area display position setting

The planes have the priority such as Code plane > Area plane 1 > Area plane 2 > Raster plane.

1. Horizontal display start position

The horizontal display start position can be set in 512 steps by writing to OSD control registers AHSn8 to AHSn0 (bit 8 to 0 in ORAHSn). And also display stop position is correspond to AHEn8 to AHEn0 (bit 8 to 0 in ORAHEn). (n; 1 to 2)

Horizontal display start position

 $AHSn = (AHSn8 \text{ to } AHSn0)H \times 2T_{OSC}$

Horizontal display end position

 $AHEn = (AHEn8 \text{ to } AHEn0)H \times 2T_{OSC}$

Note 1: T_{OSC}: One cycle of OSD oscillation.

- Note 2: If the horizontal display start position for characters is the same as that for areas, the two positions are not displayed at the same time. The horizontal display start position for characters is displayed 16 T_{OSC} (corresponding to a register value of 8) later than that for areas.
- 2. Vertical display start position

The vertical display start position can be set in 625 steps by writing to OSD control registers AVSn9to AVSn0 (bit 9 to 0 ORAVSn). And also display stop position is correspond to AVEn9 to AVEn0 (bit 9 to 0 in ORAVEn). (n; 1 to 2)

Vertical display start position

 $AVSn = (AVSn9 \text{ to } AVSn0)H \times T_{HD}$

Vertical display end position

AVEn = $(AVEn9 \text{ to } AVEn0)H \times T_{HD}$

Note: T_{HD} : One cycle of \overline{HD} signal.

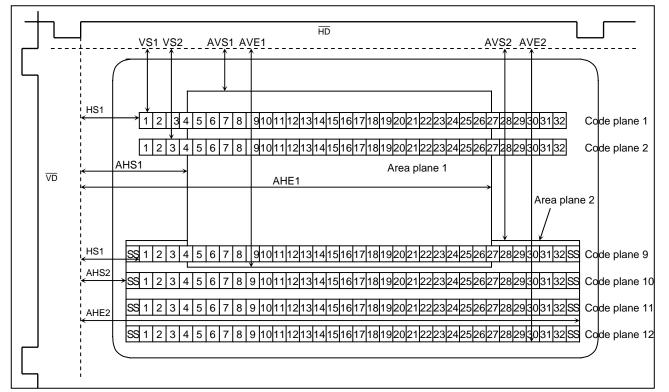


Figure 2.14.16 TV Scan Image

2.14.6.4 Character Ornamentation Control

(1) Character sizes

Character size can be selected line by line from 4 sizes. And display on/off also can be set line by line. Small, middle, large and double height character size and display on/off can be set with OSD control registers CSn (n = 1 to 12, ORCS4, ORCS8, ORCS12) in the OSD control registers.

Character sizes: 4 sizes (Small, middle, large and double height)

Character size and display on/off specification unit: Line

Character size select/display on/off register (2 bits \times 12)

Line 1: CS1 Line 2: CS2 : :

Line 12: CS12

CSn (high-order bit)	CSn (low-order bit)	Character size	DCSCn (double-height specification)	Display on/off
1	1	Small-size character	0	On
1	0	Medium-size character	0	On
0	1	Large-size character	0	On
1	0	Double-height character	1	On
0	0	_	0	Off

- Note 1: To display a double-height character, write "10" and "1", respectively, to CSn (medium-size character specification) and DCSCm (double-height display specification). If DCSCm and CSn are, respectively, "0" and "10", medium-size characters are displayed.
- Note 2: If the character size specification (CSn) is "11" or "01", no double-height character can be displayed.
- Note 3: Do not specify to modify double-height characters (such as fringing, smoothing, and slanting) because such specifications hamper normal display.
- Note 4: The display off line operates like the width of small character size line thought the character is not displayed.
- Note 5: The data written to these control registers is transmitted to OSD circuit by setting RGWR (bit 2 in ORDON) to "1".
- Note 6: When OSD circuit is used on an interlace scanning TV, a jitter elimination circuit must be enabled and set AFLD to "1" in JECR.
- Note 7: When VDSMD and AFLD are "0", only character of even display dot is displayed. (refer to 2.16 a jitter elimination circuit)

	VDSMI (normal				1D = 1 can mode)
		Dot size Character size		Dot size	Character size
EULAn = 0	Small-size character	1T _{OSC} ×0.5T _{HD}	16T _{OSC} ×9T _{HD}	$1T_{OSC} \times 1T_{HD}$	16T _{OSC} × 18T _{HD}
(underline off)	Medium-size character	$2T_{OSC} \times 1T_{HD}$	$32T_{OSC} \times 18T_{HD}$	2T _{OSC} ×2T _{HD}	$32T_{OSC} \times 36T_{HD}$
	Large-size character	$4T_{OSC} \times 2T_{HD}$	$64T_{OSC} \times 36T_{HD}$	$4T_{OSC} \times 4T_{HD}$	$64T_{OSC} \times 72T_{HD}$
	Double-height character	$1T_{OSC} \times 1T_{HD}$	$16T_{OSC} \times 18T_{HD}$	$1T_{OSC} \times 2T_{HD}$	$16T_{OSC} \times 36T_{HD}$
EULAn = 1	Small-size character	$1T_{OSC} \times 0.5T_{HD}$	$16T_{OSC} \times 12T_{HD}$	$1T_{OSC} \times 1T_{HD}$	$16T_{OSC} \times 24T_{HD}$
(underline on)	Medium-size character	$2T_{OSC} \times 1T_{HD}$	$32T_{OSC} \times 24T_{HD}$	$2T_{OSC} \times 2T_{HD}$	$32T_{OSC} \times 48T_{HD}$
	Large-size character	$4T_{OSC} \times 2T_{HD}$	$64T_{OSC} \times 48T_{HD}$	$4T_{OSC} \times 4T_{HD}$	$64T_{OSC} \times 72T_{HD}$
	Double-height character	$1T_{OSC} \times 1T_{HD}$	$16T_{OSC} \times 24T_{HD}$	$1T_{OSC} \times 2T_{HD}$	$16T_{OSC} \times 48T_{HD}$

Table 2.14.5 Dot Size and Character Size

Note: T_{OSC} = one OSD oscillation cycle. T_{HD} = one \overline{HD} signal cycle.

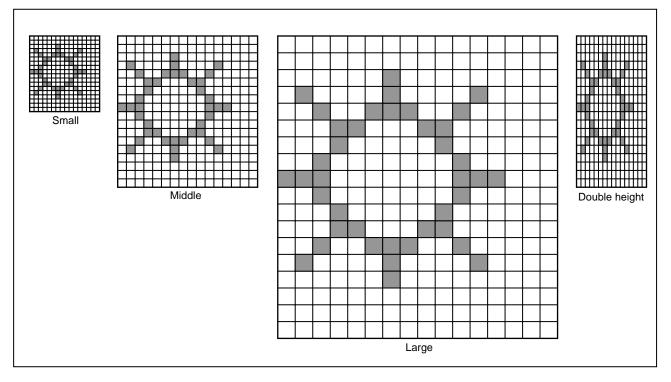


Figure 2.14.17 Character Size

(2) Smoothing function

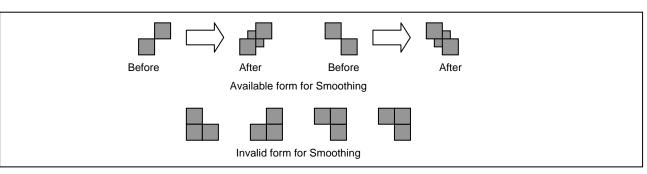
The smoothing function is used to make characters look smooth. Enabling smoothing displays 1/4 dot between two dots connecting corner to corner within a character. Small size character and color font can not be enabled smoothing. Smoothing is enabled by setting ESMZ (bit 4 in ORETC) in the OSD control register to "1".

Smoothing specification unit: Display page

Smoothing specification register (1 bit) \cdots ESMZ (bit 4 in ORETC)

"0" ... Disable smoothing "1" ... Enable smoothing

Note 1: Data of the register is transferred to the OSD circuit and become valid when the data is written.



Note 2: The smoothing function is invalid for the color font.

Figure 2.14.18 Available Form and Invalid Form for Smoothing

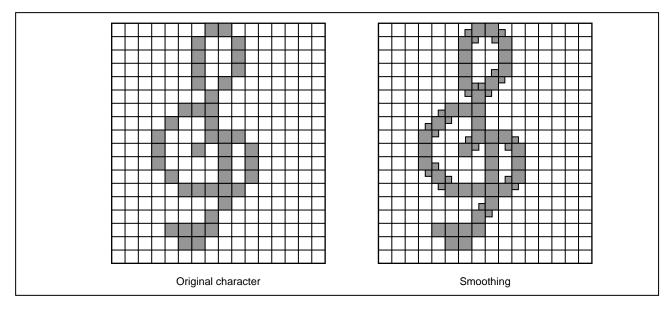


Figure 2.14.19 Smoothing Example

(3) Fringing function

The fringing function is used to display a character with a fringe width is 1 dot in a different color from that of the character. When a character is displayed with the maximum of 18 vertical dots and 16 horizontal dots, the fringe exceeds right and left of the character display area. No vertical fringing is displayed out of the character display area. If there is an adjacent character that outer dot is active, then this dot will overrule the fringe in the horizontal direction. Underlines are not fringed.

Fringing is enabled for each line by setting EFR1 to EFR8 (OREFR8) and EFR9 to EFR12 (OREFR12) in the OSD control register to "1".

A color for fringe is specified common to all lines using OSD control registers, RFDT, GFDT, and BFDT (bit 2 to 0 in ORBK).

Fringing specification unit: Line

Fringing enable register (1 bit \times 12) \cdots EFRn (n; 1 to 8) (OREFR8), EFRn (n; 9 to 12) (OREFR12)

"0" ... Disable fringing "1" ... Enable fringing

Fringe colors: 8 or 27

Fringe color specification unit: Display page

Fringe color register (3 bits) ··· RFDT, GFDT, BFDT (bit 2 to 0 in ORBK)

- Note 1: The fringe of 1st column character does not exceed left, and the fringe of 32th character does not exceed right.
- Note 2: Do not specify fringing for the color font.
- Note 3: Do not specify fringing for characters for which double-height display is specified.

RFDT	GFDT	BFDT	Figure color
0	0	0	Setting color of ORCPT0
0	0	1	Setting color of ORCPT1
0	1	0	Setting color of ORCPT2
0	1	1	Setting color of ORCPT3
1	0	0	Setting color of ORCPT4
1	0	1	Setting color of ORCPT5
1	1	0	Setting color of ORCPT6
1	1	1	Setting color of ORCPT7

Table 2.14.6 Fringe Color

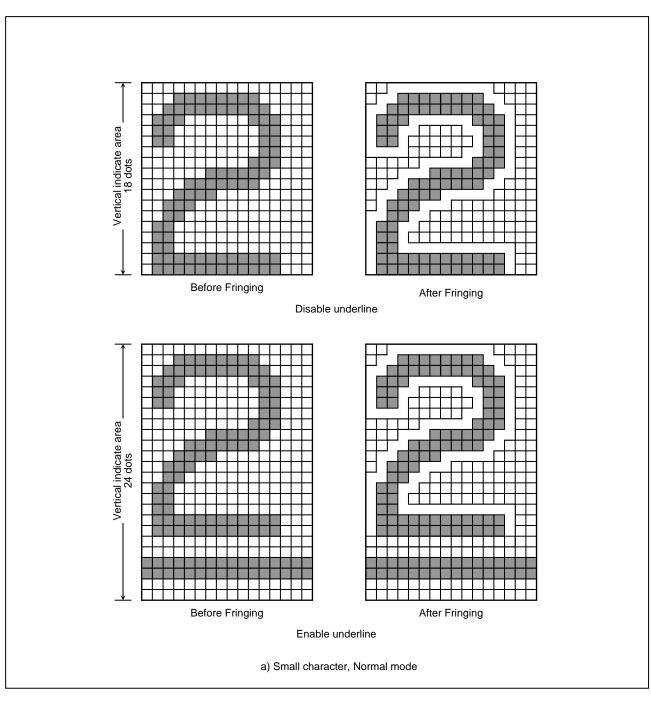


Figure 2.14.20 (a) Fringing Example

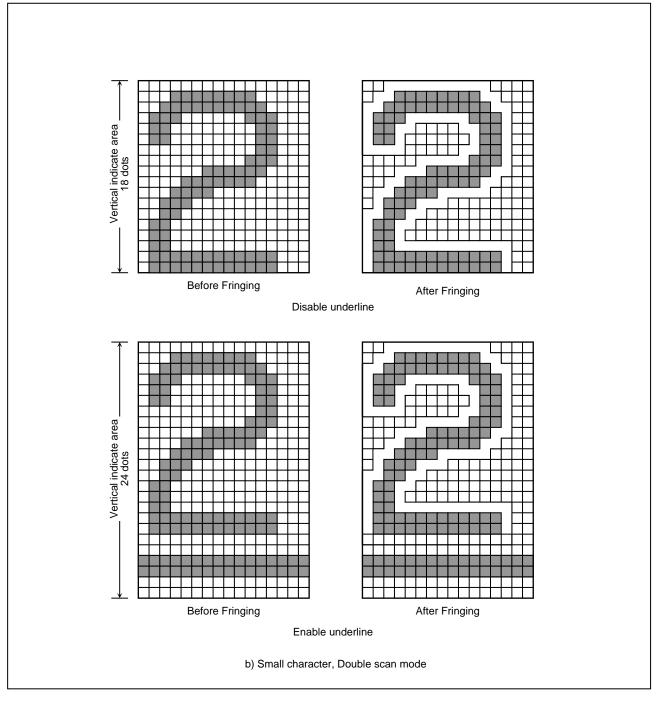


Figure 2.14.21 (b) Fringing Example

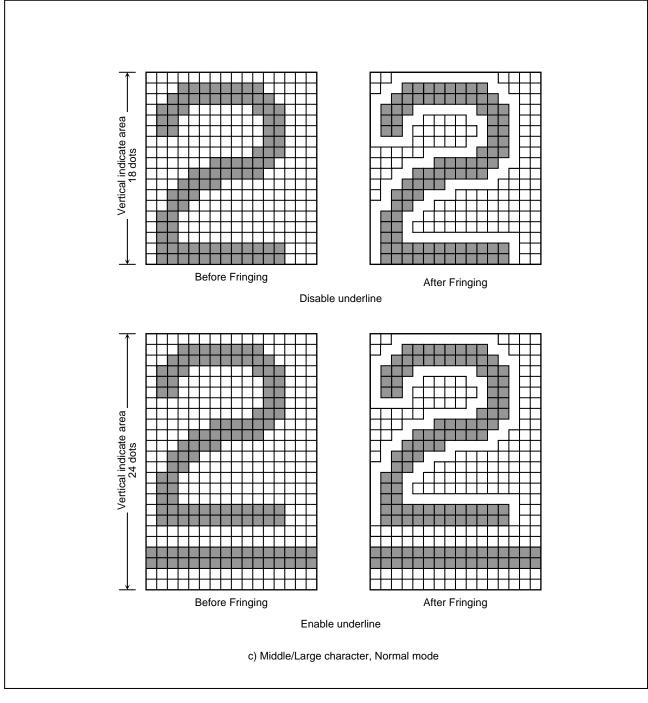


Figure 2.14.22 (c) Fringing Example

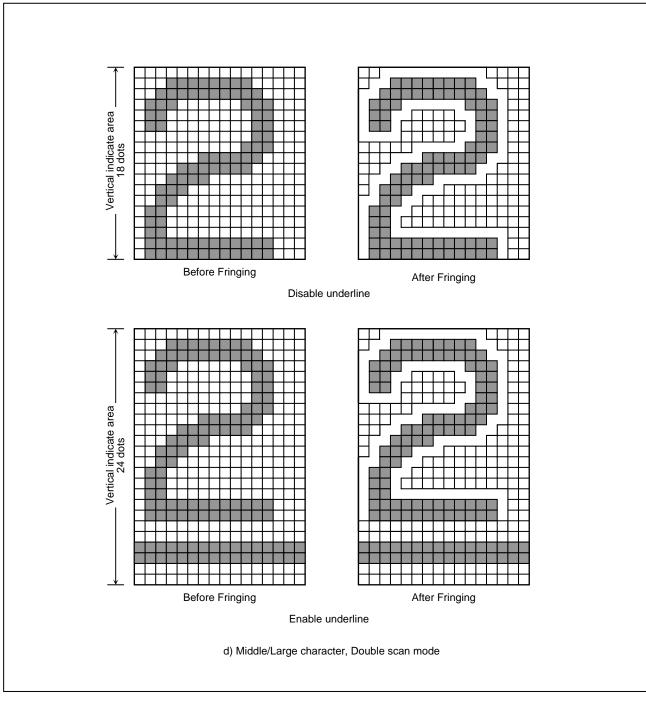


Figure 2.14.23 (d) Fringing Example

(4) Double-height display function

It is possible to display a character having the same horizontal size as for the small-size character and the same vertical size as for the medium-size character. This function can be realized by specifying medium-size character display for the character size and setting up the double-height display setting register (ORDCSC). Its specification unit is the row.

Double-height display enable unit: Row

Double-height display enable register (1 bit \times 12): DCSCn (n = 1 to 12) (ORDCSC register)

Character size specification: "10" is set in CSn (n = 1 to 12; ORCS4, ORCS8, and ORCS12).

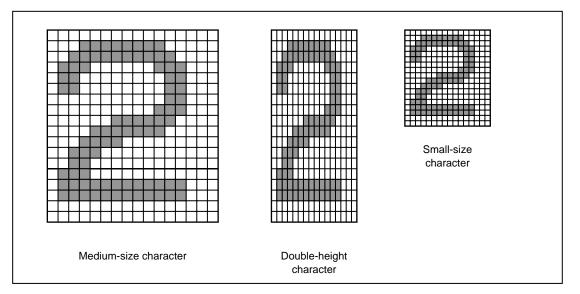


Figure 2.14.24 Double-Height Character Display

Note: Do not specify the fringing, smoothing, or slanting character modification function for a row where double-height display is specified.

(5) Displaying a Small-Size Character Consisting of 26 Vertical and 18 Horizontal Dots

It is possible to display small-size characters at vertical intervals of 26 scanning lines. This function is realized by specifying small-size character display and setting up the 26-dot vertical display setting register ORCCD. This specification can be made in line units.

26-dot vertical display enable unit: Row

26-dot vertical display enable register (1 bit \times 12): CCDn (n = 1 to 12) (ORCCD register)

Character size specification: "11" is set in CSn (n = 1 to 12; ORCS4, ORCS8, and ORCS12).

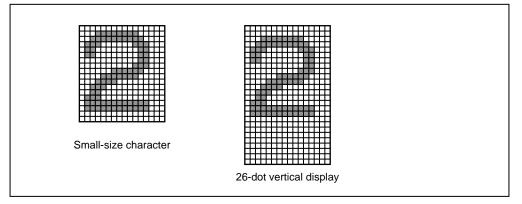


Figure 2.14.25 26-dot Vertical Display

(6) Background function

The background color is the color of all backgrounds including the background of the character area (see Table 2.14.5). The background function is specified in screen units by setting the EBKGD OSD control register (bit 7 in the ORRCL register) to "1". Using the ECBKD OSD control register (bit 3 in the ORDSN register) can enable/disable the character-specific background color.

The background color is specified, using the RBDT, GBDT, and BBDT OSD control registers (bits 6 to 4 in the ORBK register). Setting the ECHDSN OSD control register (bit 3 in the ORDON register) to "1" specifies SLNT (bit 6 in the ORDSN register) and EUL (bit 5 in the ORDSN register), respectively, as RBDT and GBDT. A background color different from that of the screen can be set up as a character-specific background.

Background color enable units: Screen and character

Background enable register (2 bits)

Screen unit: EBKGD (bit 7 in the ORRCL register) Character unit: ECBKD (bit 3 in the ORDSN register)

Background color specification units: Screen and character

Background color specification register

If ECHDSN = 0: RBDT, GBDT, and BBDT (bits 6 to 4 in the ORBK register) If ECHDSN = 1: RBDT, GBDT (bits 6 to 4 in the ORBK register),

SLNT (corresponding to RBDT), and EUL (corresponding to GBDT)

OSD control register		Display status	
EBKGD	ECBKD	Display status	
0	0	No background is displayed.	
0	1	No background is displayed.	
1	0	No background is displayed.	
1	1	A background is displayed.	

Table 2.14.7 Background Color Control

Table 2.14.8 Character-Specific Background Color Setting Function

	Register name	Function	Character-specific background color setting (ECHDSN)	
	name		0	1
	SLNT	Slanting	←	RBDT (background color of red)
Character modification specification	EUL	Underlining	←	GBDT (background color of green)
register	BLF	Blanking	\leftarrow	\leftarrow
C .	ECBKE	Character-specific background enable	\leftarrow	←

- Note1: When the ECHDSN is set to "1", the background color is specified by RBDT (red) and GBDT (green) bits.In this case, ORCPT0,ORCPT2,ORCPT4 and ORCPT6 are available for color pallet.
- Note 2: OSD output isn't done, and a video signal is indicated in the background area in case of EBKGD=0, ECBKD=0 and EBKGD=1, ECBKD=0.A background area becomes transparent in case of EBKGD=0 and ECBKD=1. That color is indicated when it is piled up and indicated with the area plane. The background color specified in case of EBKGD=1 and ECBKD=1 is indicated.

			-
RBDT	GBDT	BBDT	Background color
0	0	0	Setting color of ORCPT0
0	0	1	Setting color of ORCPT1
0	1	0	Setting color of ORCPT2
0	1	1	Setting color of ORCPT3
1	0	0	Setting color of ORCPT4
1	0	1	Setting color of ORCPT5
1	1	0	Setting color of ORCPT6
1	1	1	Setting color of ORCPT7

Table 2.14.9 Background Color

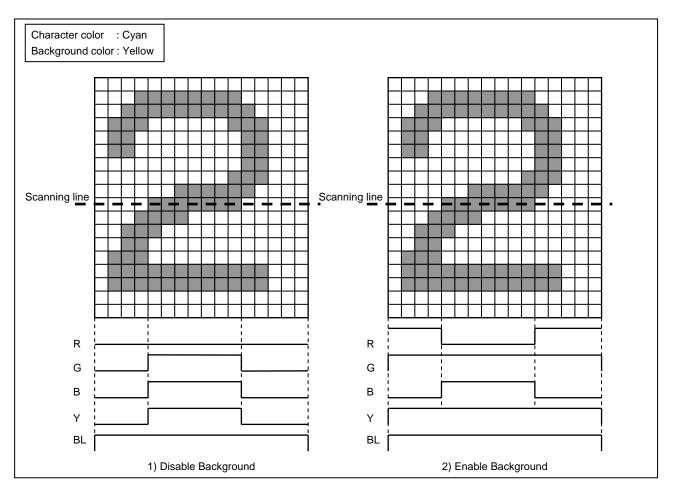


Figure 2.14.26 Background Function

Note: When the background function is enabled, the line enable the fringing function should not start with a blank character. If it starts with a blank character, a fringe is displayed to the left of the blank character.

2.14.6.5 OSD Display Screen Control

(1) Display on/off

This function is used to display characters specified for on/off display.

Display on/off specification unit: Display page

Display on/off specification register (1 bit) ... DON (bit 0 in ORDON)

"0"	 Disable display
"1"	 Enable display

Note: Do not start STOP mode during display is enable.

(2) Window function

This function is used to set upper and lower limit of display page. Window upper limit is specified by WVSH (ORWVSH). Window lower limit is specified by WVSL (ORWVSL). This function is enabled by setting EWDW (bit 1 in ORDON) in the OSD control register to 1.

Window specification unit: Display page

Window	function	enable	specification	register	(1	bit)	•••	EWDW	(bit	1	in
								ORDON	1)		
"0"		Disable	window functi	ion							
"1"		Enable v	window functi	on							

Window upper limit specification register (10 bits) … WVSH9 to 0 (ORWVSH) Window lower limit specification register (10 bits) … WVSL9 to 0 (ORWVSL)

Window upper and lower limit position …

When VDSMD is "0" (Normal mode):

WVSH = (WVSH9 to WVSH0) $H \times T_{HD}$

 $WVSL = (WVSL9 \text{ to } WVSL0) \text{ H} \times \text{T}_{HD}$

When VDSMD is "1" (Double scan mode):

WVSH = (WVSH9 to WVSH0) $H \times 2T_{HD}$

WVSL = (WVSL9 to WVSL0) $H \times 2T_{HD}$

Note 1: T_{HD}; One cycle of HD signal

Note 2: $WVSL > WVSH \ge "1"$

- Note 3: Modify the value of window upper and lower limit register and the value of EWDW during \overline{VD} signal is low.
- Note 4: It is recommendable that the window function is always enabled (EWDW = "1") and set WVSH to "01H", WVSL to "1FEH".

Note 5: Characters and symbols at scanning line specified by WVSL are not displayed.

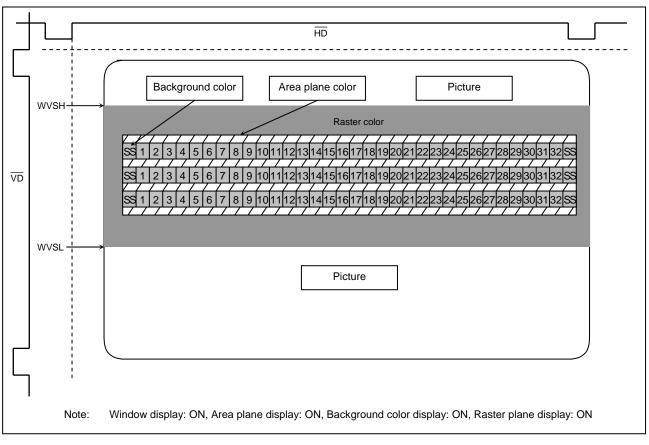


Figure 2.14.27 Display Example

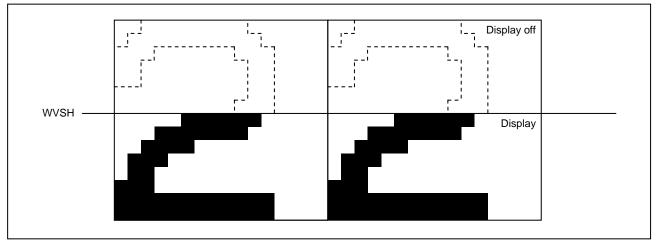


Figure 2.14.28 If WVSH is on a Code Plane

(3) Full-raster blanking function

Full-raster blanking function is used to color the entire background for the display area (TV screen). When using the full-raster blanking function, set YBLCS (bit 2 in ORP6S) to "1", output BL signal from Y/BL pin, because Y signal cannot delete whole display page from video signal.

This function is specified for each display page by setting EXBL (bit 6 in ORRCL) in the OSD register to "1".

Full-raster blanking specification unit: Display page

Full-raster blanking enable register (1 bit) ... EXBL (bit 6 in ORRCL)

- "0" ... Disable full-raster blanking
- "1" ... Enable full-raster blanking

Full-raster blanking color specification...RCLR, RCLG, RCLBregisters (3 bits)(bit 2 to 0 in ORRCL)

RCLR	RCLG	RCLB	Raster plane color
0	0	0	Setting color or ORCPT0
0	0	1	Setting color or ORCPT1
0	1	0	Setting color or ORCPT2
0	1	1	Setting color or ORCPT3
1	0	0	Setting color or ORCPT4
1	0	1	Setting color or ORCPT5
1	1	0	Setting color or ORCPT6
1	1	1	Setting color or ORCPT7

Table 2.14.10 Raster Plane Color

(4) Area plane function

Area plane function is used to display square area to two points on a screen.

Two planes operate independently. They are displayed according to the priority (area plane 1 > area plane 2).

See area plane display position setting in section 2.14.6.3 (2) how to set display positions for each area.

Each area plane is set to ON or OFF by AON2 and AON1 (bit 5 and bit 4 in ORRCL).

Area plane colors are set by ACLRx, ACLGx, ACLBx (bit 6 to bit 4 and bit 2 to bit 0 in ORACL, x = 1, 2).

Area plane colors: 8 or 27

Area plane specification unit: plane

Area plane color specification register (6 bit)

Area plane 1: ACLR1/ACLG1/ACLB1 (bit 2 to 0 in ORACL)

Area plane 2: ACLR2/ACLG2/ACLB2 (bit 6 to 4 in ORACL)

ACLRx	ACLGx	ACLBx	Area plane color
0	0	0	Setting color of ORCPT0
0	0	1	Setting color of ORCPT1
0	1	0	Setting color of ORCPT2
0	1	1	Setting color of ORCPT3
1	0	0	Setting color of ORCPT4
1	0	1	Setting color of ORCPT5
1	1	0	Setting color of ORCPT6
1	1	1	Setting color of ORCPT7
1	1	1	Ŭ

Table 2.14.11 Area Plane Color

(5) I-pin function

The I-pin output becomes valid only for area planes. Resetting the PIDS OSD control register (bit 3 in the ORP6S register) to "0" causes P57 to work for I-pin output. If you want to produce an I-pin output for area plane 1, set the PISEL1 OSD control register (bit 3 in the ORACL register) to "1". If you want to produce an I-pin output for area plane 2, set the PISEL2 OSD control register (bit 7 in the ORACL register) to "1". The I-pin output depends on the display priority of the area planes.

(6) Examples of OSD outputs

- PISEL1 = 1 PISEL2 = 1 Example 3
 <EXBL = 0, YBLCS = 0, 1> Area 1 Halftone Area 2 Halftone ∢ ഷധവ ≻ _ В ≻ В ≻ ≻ В Ы ∢ മ υ Δ PISEL1 = 1PISEL2 = 0 Example 2
 <EXBL = 0, YBLCS = 0, 1> Area 1 Halftone < ++ Area where any of font colors R, G, and B is high Area 2 ∢ ഷധവ ≻ В -≻ В _ ≻ Ч ≻ В Δ ∢ മ c $\begin{array}{l} \mathsf{PISEL1}=0\\ \mathsf{PISEL2}=1\\ \end{array}$ → Character display area Example 1 <EXBL = 0, YBLCS = 0, 1> Area 2 Halftone Area 1 ||||∢ ഷധവ • _ ВГ _ В ≻ ≻ _ ≻ _ ≻ ВГ Ы
 - Figure 2.14.29 OSD Output Examples (a)

∢

മ

C

Δ

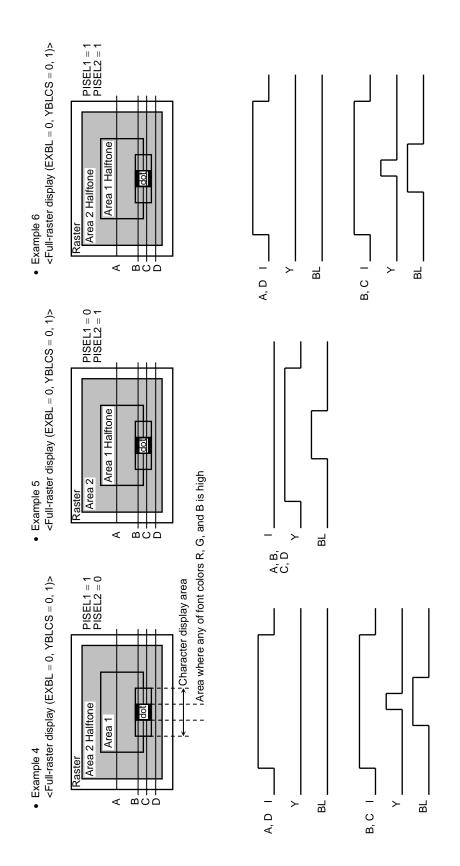
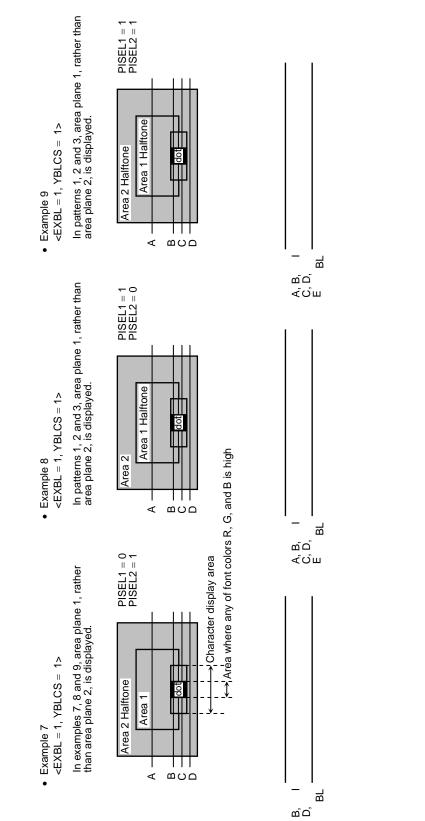


Figure 2.14.30 OSD Output Examples (b)



₹ÚШ

Figure 2.14.31 OSD Output Examples (c)

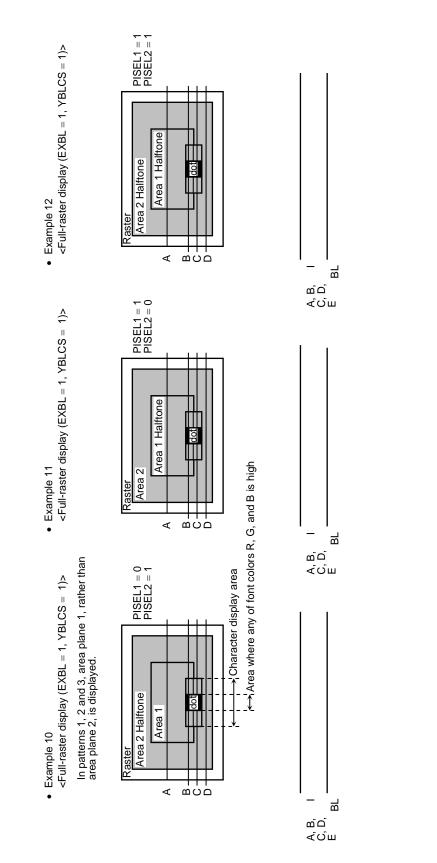


Figure 2.14.32 OSD Output Examples (d)

2.14.6.6 Interrupt Control

(1) Display line counter

to

The display line counter indicates number of display line (s) by OSD circuit on the TV screen. The display line counter is a 4-bit counter which is initialized to "0" by the falling edge of the $\overline{\text{VD}}$ signal and which increments when last scanning of each display line is completed (falling edge of the $\overline{\text{HD}}$ signal). It is necessary to be read out display line counter several times, because it does not synchronize CPU clock.

Display line counter register (4 bits) ... DCTR (bit 3 to 0 in ORIRC)

"0000" ... No display line is completed.

"0001" ... 1st display line is completed.

"0010" ... 2nd display line is completed.

to

"1111" \cdots 15th display line is completed.

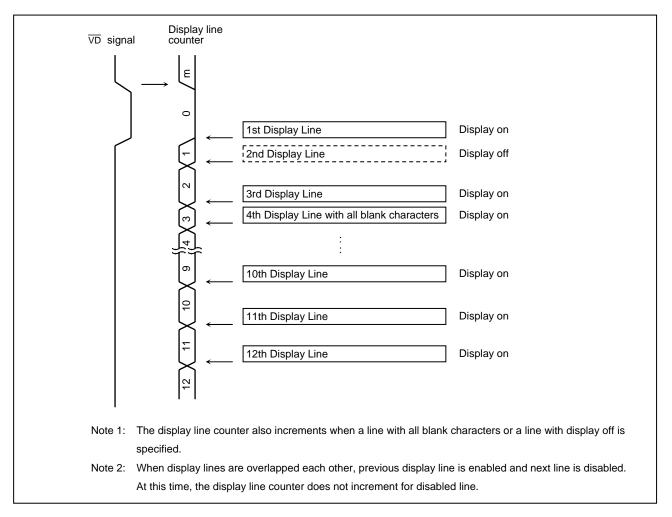


Figure 2.14.33 Display Line Counter

(2) Interrupt generator circuit

An interrupt request is generated when a falling edge of $\overline{\text{VD}}$ signal or when line counter (DCTR) is counted to the certain value specified by ISDC.

Interrupt source select register (1 bit): SVD (bit 4 in ORIRC)

- "0" ... Interrupt request generated when the display line counter (DCTR) is counted to the certain value which is specified by ISDC.
- "1" \cdots Interrupt request is generated when a falling edge of $\overline{\text{VD}}$ signal.

Interrupt generation line specification register (4 bits) ··· ISDC (bit 3 to 0 in ORIRC)

"0000"	•••	Interrupt request generated when the display line counter is
		cleared.
"0001"		Interrupt request generated at end points of the last scanning
		line of the first display line
"0010"		Interrupt request generated at end points of the last scanning
		line of the 2'nd display line
to		
"1111"		Interrupt request generated at end points of the last scanning
		line of the 15'th display line

2.14.6.7 Display Memory Access

(1) Display memory

The display memory is accessed for two purposes, one for writing data to the display memory, and one for reading data from the display memory.

Display memory address specification registers $\ \cdots$ DMA8 to MDA0 (ORDMA) (9 bits)

Display memory data write registers

Character code write register (9 bits)	\cdots CRA8 to CRA0 (ORCRA)
Character ornamentation data write registers (6 bits)	···· SLNT, EUL, BLF, RDT, GDT, and BDT (ORDSN)
Character-specific background on/off specification register (1 bit)	··· ECBKD (ORDSN register)

Display memory bank select register MBK (bit 1 in ORETC)

- "0" ... When writing either character code or character ornamentation data
- "1" ... When writing both character code and character ornamentation data
- Note 1: These control registers have a characteristic that immediately when a value is written to the register, the content of the register is transferred as valid data to the OSD circuit/display memory.
- Note 2: The data written to the display memory takes effect at the same time it is written. When character code or character ornamentation data is written to the display memory while it is displaying some character, the character may not be displayed correctly. When writing data to the display memory, make sure no character is being displayed in the memory location where you are going to write data.
- Note 3: When writing data to or reading data from the display memory, do not use two-byte transfer instructions such as "LDW(HL),mn LD rr, (pp)." Otherwise, erroneous data may be written to the display memory or data may be written to an incorrect address.
- Note 4: Allow for at least two instruction cycles between a display memory address write instruction and a data write or read instruction. Also, when continuous writing data to or reading data from the display memory, allow for at least two instruction cycles between one write or read instruction and the next. Otherwise, erroneous data may be written to the display memory or data may be written to an incorrect address.
- Note 5: When setting display memory addresses, always be sure to write all of 9 address bits sequentially in order of DMA8 and DMA7 to DMA0.

1. Normal mode

In normal mode, the display memory addresses are automatically incremented each time data is read from or written to the memory. Because addresses are automatically incremented, this mode may be used for reading from or writing data to multiple continuous addresses simultaneously.

<Display memory write sequence in normal mode>

- (a) When writing either character code or character ornamentation data
 - (1) Set MFYWR, MBK, and RDWRV all to 0.
 - (2) Write the most significant address bit of the display memory to DMA8. Go on and write the 8 low-order address bits of the display memory to DMA7 to DMA0.
 - (3) Writing character code or character ornamentation data
 - Writing character code

Write the most significant bit of character code to CRA8. Go on and write the 8 low-order bits of character code to CRA7 through CRA0. At this point in time, the 9 bits of character code written are transferred to the display memory, and DMA8 to DMA0 are automatically incremented.

• Writing character ornamentation data

Write character ornamentation data to SLNT, EUL, BLF, ECBKD, RDT, GDT, and BDT. At this point in time, the character ornamentation data written are transferred to the display memory, and DMA8 to DMA0 are automatically incremented.

- (4) To write data (character code or character ornamentation data) to continuous addresses, repeat step (3).
- (b) When writing character code and character ornamentation data at a time
 - (1) Set MFYWR to 0, MBK to 1, and RDWRV to 0.
 - (2) Write the most significant address bit of the display memory to DMA8. Go on and write the 8 low-order address bits of the display memory to DMA7 to DMA0.
 - (3) Write character ornamentation data to SLNT, EUL, BLF, ECBKD, RDT, GDT, and BDT. At this point in time, the character ornamentation written are transferred to the display memory.
 - (4) Write the most significant bit of character code to CRA8. Go on and write the 8 low-order bits of character code to CRA7 to CRA0. At this point in time, the 9 bits of character code written and the character ornamentation data written in step (3) are transferred to the display memory, and DMA8 to DMA0 are automatically incremented.
 - (5) To write data to continuous addresses, repeat steps (3) and (4).

<Display memory read sequence in normal mode>

- (a) When reading either character code or character ornamentation data
 - (1) Set MFYWR to 0, MBK to 0, and RDWRV to 1.
 - (2) Write the most significant address bit of the display memory to DMA8. Go on and write the 8 low-order address bits of the display memory to DMA7 to DMA0.
 - (3) Reading character code or character ornamentation data
 - Reading character code

Read the most significant bit of character code to CRA8. Go on and read the 8 low-order bits of character code to CRA7 to CRA0. At this point in time, DMA8 to DMA0 are automatically incremented.

• Reading character ornamentation data

Read character ornamentation data SLNT, EUL, BLF, ECBKD, RDT, GDT, and BDT. At this point in time, DMA8 through DMA0 are automatically incremented.

- (4) To read data (character code or character ornamentation data) from continuous addresses, repeat step (3).
- (b) When reading character code and character ornamentation data at a time
 - (1) Set MFYWR to 0, MBK to 1, and RDWRV to 1.
 - (2) Write the most significant address bit of the display memory to DMA8. Go on and write the 8 low-order address bits of the display memory to DMA7 to DMA0.
 - (3) Read character ornamentation data SLNT, EUL, BLF, ECBKD, RDT, GDT, and BDT.
 - (4) Read the most significant bit of character code to CRA8. Read the 8 low-order bits of character code to CRA7 to CRA0. At this point in time, DMA8 to DMA0 are automatically incremented.
 - (5) To read data from continuous addresses, repeat steps (3) and (4).
- 2. Read-modify-write mode

When writing data in read-modify-write mode, the display memory addresses are automatically incremented as in normal mode, but when reading data in this mode, the memory addresses are not automatically incremented.

Therefore, immediately after executing a read from some display memory address, you can execute a write to the same display memory address. After executing a write, the display memory addresses are automatically incremented.

- (a) Reading/writing either character code or character ornamentation data in read-modify-write mode
 - (1) Set MFYWR to 1 and MBK to 0, and RDWRV to 1.
 - (2) Write the most significant address bit of the display memory to DMA8. Go on and write the 8 low-order address bits of the display memory to DMA7 to DMA0.
 - (3) Reading character code or character ornamentation data
 - Reading character code

Read the most significant bit of character code to CRA8. Read the 8 low-order bits of character code to CRA7 to CRA0. DMA8 to DMA0 are not incremented.

• Reading character ornamentation data

Read character ornamentation data SLNT, EUL, BLF, ECBKD, RDT, GDT, and BDT. DMA8 to DMA0 are not incremented.

- (4) Writing character code or character ornamentation data
 - Set RDWRV to "0".
 - Writing character code

Write the most significant bit of character code to CRA8. Go on and write the 8 low-order bits of character code to CRA7 to CRA0. At this point in time, the 9 bits of character code written are transferred to the display memory, and DMA8 to DMA0 are automatically incremented.

• Writing character ornamentation data

Write character ornamentation data to SLNT, EUL, BLF, ECBKD, RDT, GDT, and BDT. At this point in time, the character ornamentation data written are transferred to the display memory, and DMA8 to DMA0 are automatically incremented.

- (5) To continue executing read-modify-write operations, repeat steps (1) to (4). To read/write data (character code or character ornamentation data). To continue executing read modify-write mode from continuous addresses, repeat steps (3) and (4).
- (b) Reading/writing both character code and character ornamentation data in read-modify-write mode
 - (1) Set MFYWR to 1, MBK to 1 and RDWRV to 1.
 - (2) Write the most significant address bit of the display memory to DMA8. Go on and write the 8 low-order address bits of the display memory to DMA7 to DMA0.
 - (3) Read character ornamentation data SLNT, EUL, BLF, ECBKD, RDT, GDT, and BDT. At this point in time, DMA8 to DMA0 are not incremented.
 - (4) Read the most significant bit of character code to CRA8. Read the 8 low-order bits of character code to CRA7 to CRA0. At this point in time, DMA8 to DMA0 are not incremented.
 - (5) Set RDWRV to "0".
 - (6) Write character ornamentation data to SLNT, EUL, BLF, ECBKD, RDT, GDT, and BDT. At this point in time, the character ornamentation data written is transferred to the display memory.
 - (7) Write the most significant bit of character code to CRA8. Go on and write the 8 low-order bits of character code to CRA7 to CRA0. At this point in time, the 9 bits of character code written and the character ornamentation data written in step (6) are transferred to the display memory, and DMA8 to DMA0 are automatically incremented.
 - (8) To continue executing read-modify-write operations, repeat steps (1) to (7). (To read/write data to and from continuous addresses in read-modify-write mode, repeat steps (3) to (7).)

		RD(RDV	VRV =1)	WR(RDWRV=0)		
		Character ornamentation	Character code	Character ornamentation	Character code	
MFYWR = 0	MBK = 0	INC	INC	INC	INC	
	MBK = 1	-	INC	-	INC	
MFYWR = 1	MBK = 0	-	-	INC	INC	
	MBK = 1	_	-	=	INC	

Table 2.14.12 Address Increment

INC: Automatic address increment at read or write.

-: No address change at data read or write.

Example: Setting a character code (020H) to the display memory (Address: 120H) and setting a character ornamentation (001H) for character code 020H and display memory address 120H.

1.	MBK = 0				
	; Set	display n	nemory addres	s	
		LD	(0x25),	0x01	; ORDMA <dma8></dma8>
		LD	(0x24),	0x20	; ORDMA <dma7:0></dma7:0>
	; Set	character	r code		
		LD	(0x1F),	0x00	; ORCRA <cra8></cra8>
		LD	(0x1E),	0x20	; ORCRA <cra7:0></cra7:0>
	; Set	display n	nemory addres	s again	
		LD	(0x25),	0x01	
		LD	(0x24),	0x20	
	; Set	character	r ornamentatio	n	
		LD	(0x1D),	0X01	; ORDSN <slnt, bdt=""></slnt,>
2.	MBK = 1				
	; Set	display n	nemory addres	s	
		LD	(0x25),	0x01	
		LD	(0x24),	0x20	
	; Set	character	r ornamentatic	n	
		LD	(0x1D),	0X01	
	; Set	character	r code		
		LD	(0x1F),	0x00	
		LD	(0x1E),	0x20	

- Note 1: To write character data into the display memory, first write into register CRA8 and then write into registers CRA7 to CRA0. When data is written into registers CRA7 to CRA0, DMA is incremented. It is impossible to write into the display memory for CRA7 to CRA0 alone. If no data is written into register CRA8 while data is written into registers CRA7 to CRA0, the value previously written into register CRA8 is written into the associated display memory.
- Note 2: To read data from the display memory, first read from register CRA8, and then read from registers CRA7 to CRA0. When data is read from registers CRA7 to CRA0, DMA8 to DMA0 is incremented.
- Note 3: There should be a time interval of at least two machine cycles between a DMA set instruction and a data write/read instruction. There should be a time interval of at least two machine cycles between a data write instruction and a data read instruction.

- (2) Characters
 - If ROMACH (bit 5 in ORDON) = 0

Characters: 383 monochrome font characters and 96 color font characters

Character specification register (9 bits): CRA8 to CRA0 (bits 8 to 0 in the ORCRA register)

Character codes: User-programmable in character ROM

Monochrome font codes "001H" to "17FH"

Color font codes "180H" to "1DFH"

• If ROMACH (bit 5 in ORDON) = 1

Characters: 447 monochrome font characters and 64 color font characters

Character specification register (9 bits): CRA8 to CRA0 (bits 8 to 0 in the ORCRA register)

Character codes: User-programmable in character ROM

Monochrome font codes "001H" to "17FH", "1C0H" to "1DFH", "1F0H" to "1FFH" Color font codes "180H" to "1BFH"

(3) Character color

Character colors: 8 or 27

Character color specification unit: Character

Character color specification register (3 bits): RDT/GDT/BDT (bit2 to 0 in ORDSN)

RDT	GDT	BDT	Character color
0	0	0	Setting color of ORCPT0
0	0	1	Setting color of ORCPT1
0	1	0	Setting color of ORCPT2
0	1	1	Setting color of ORCPT3
1	0	0	Setting color of ORCPT4
1	0	1	Setting color of ORCPT5
1	1	0	Setting color of ORCPT6
1	1	1	Setting color of ORCPT7

Table 2.14.13 Character Color

(4) Blinking function

Blinking function is used to blink display characters.

When BKMF is "1", characters specified for blinking by BLF are not displayed. (If the background color function is used, the background color is not disappeared.)

Blinking specification unit: Character

Blinking specification register (1 bit) ... BLF (bit 4 in ORDSN)

"0" ··· No blinking

"1" ··· Blinking

Blinking master specification register (1 bit) ... BKMF (bit 5 in ORETC)

- "0" ... Disable blinking
- "1" ... Enable blinking (Characters whose BLF are set to "1" are not displayed.)
- Note: Regarding the extra dot of the left and/or right character by fringing function, it is not enabled as blink.

(5) Underline function

Underline function is used to add a line under a display character. The underline is same color as that of character.

Underline specification unit: Character/Line

Underline enable register (Character unit) (1 bit) … EUL (Bit 5 in ORDSN)

"0" ··· No underline "1" ··· Underline

Underline enable register (Line unit) (1 bit × 12) …EULAn (n: 1 to 8) (OREULA8), EULAn (n: 9 to 12) (OREULA12)

Underline colors: 8 or 27

Underline color specification registers (3 bits) … RDT, GDT, BDT (Bit 2 to 0 in ORDSN) (refer to Table 2.15.10)

- Note 1: To use the underline function, set both the underline enable register for underlining text in characters and that for underlining text in lines. If the former register (EUL) only is set, an underline is not displayed.
- Note 2: A color font underline can be displayed in colors set up using RDT, GDT, and RDT.

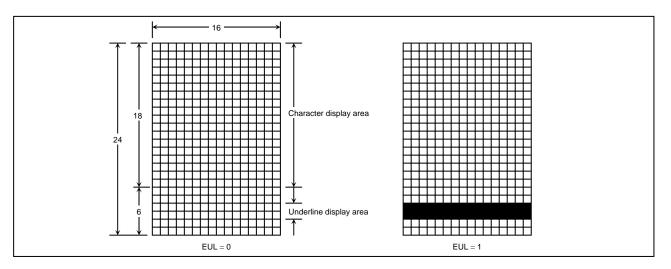


Figure 2.14.34 Underline

(6) Solid space control

Solid space control is used to display one column of solid space to the left and right of 32 columns.

Solid space control is used to delete the Video signal in the areas where solid spaces are located in the original display page, then add color (raster color) to them.

Solid space specification unit: line

Solid space specification register (24 bits)

For line 1	SOL11 and SOL10 (Bits 1 and 0 in ORSOL4)
For line 2	SOL21 and SOL20 (Bits 3 and 2 in ORSOL4)
÷	
For line 12	SOL121 and SOL120 (Bits 7 and 6 in ORSOL12)

Solid space specification

The solid space control functions as follows:

SOLx1/SOLx0 (x = 1 to 12)

4

"00"	•••	No solid space display
"01"	•••	Solid space display left for 32 columns
"10"	•••	Solid space display right for 32 columns
"11"	•••	Solid space display left and right for 32 columns

Solid space color specification registers (3 bits)

··· RBDT, GBDT, BBDT (Bits 2 to 0 in ORBK)

(Same color as that of background)

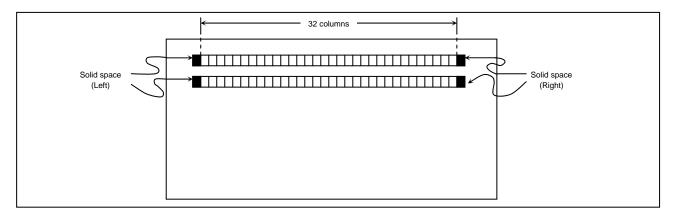


Figure 2.14.35 Solid Space

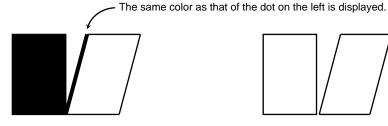
(7) Slant function

Slant function is used to slant characters for italics.

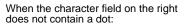
Slant specification unit: Character

Slant enable register (1 bit) … SLNT (Bit 6 in ORDSN) "0" No slant ... "1" . . . Slant

- Note 1: SLANT function is enabled each characters, and therefore, in case of using background function, this color of the Background is enable as slant. Regarding the extra dots of the left and/or right character by fringing function, it is not enabled as slant.
- Note 2: When a character is slanted in an area, which overlaps with the character field, the overlap is also slanted.
- Note 3: If slanting a character causes part of the character to get into the character field to the immediate right of the character, then this part is not displayed.
- Note 4: R, G, B, and Y are all slanted. Thus, if the Y signal is selected, a video signal is displayed above and to the left of the slant character. If the specified background color is black, setting YBLCS to "1" prevents the upper-left video signal for a slant character from being displayed.
- Note 5: When a character is slanted, the dot data to the immediate left of the character is also slanted.
- Note 6: Do not specify slanting for the color font.



When an entire character field (including its background) contains dots:



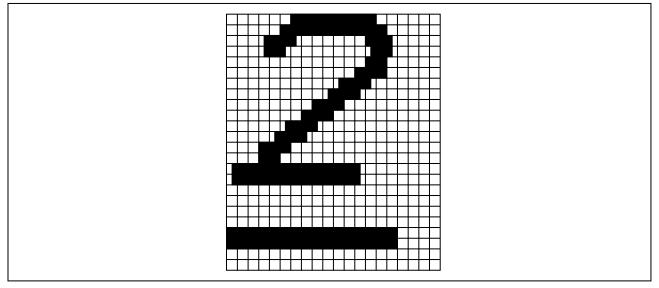


Figure 2.14.36 Slant

(8) Functions supporting PAL100/NTSC120

This LSI package supports the PAL (Phase Alternating Lines) 100 and NTSC (National Television System Community) 120 broadcasting systems. Figure 2.14.35 schematically shows the supported screen scanning method.

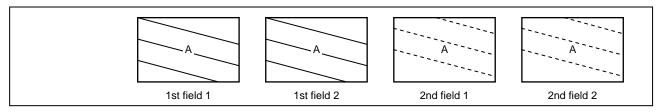


Figure 2.14.37 PAL100/NTSC120 Image Scanning Lines (Schematic Diagram)

PAL100 support enable unit: Screen

PAL100 support enable register (1 bit): EPAL100 (bit 5 in the ORDON register)

PAL100 screen display start enable register (1 bit): PALTRG (bit 0 in the ORSTRG register)

To support PAL100/NTSC120, follow this procedure.

- (a) To use PAL100/NTSC120, set the EPAL100 OSD control register (bit 5 in the ORDON register) to "1".
- (b) Read the phase detection results, PDF0 to PDF2, of the horizontal sync signal (HD) and the vertical sync signal (VD) (bits 6, 5, and 0 in the JESR jitter elimination status register) each time a VD interrupt occurs.
- (c) By reading the phase detection results PDF0 to PDF2, the phase of screen scanning is determined according to the detected field (1st or 2nd field).
- (d) Write PALTRG (bit 0 in the ORSTRG register) during the second cycle of the 2nd field (2nd field 2).

Once PALTRG has been written, it becomes possible to support PAL100/NTSC120 for OSD display in the next field (1st field).

Note 1: Use software to determine the write timing for PALTRG.

Note 2: It is impossible to normally display the screen on the field of which PALTRG is written.

Note 3: To read the phase detection results PDF0 to PDF2, write "1" to the JEEN jitter elimination control register (bit 2 in the JECR register) to enable the jitter elimination circuit.

2.14.6.8 OSD Control Registers

Can not access all OSD control registers in any of read-modify-write instructions such as bit operation, etc.

0RHS1	7	6	5	4	3	2	1	0		
(00F81H)	HS17	HS16	HS15	HS14	HS13	HS12	HS11	HS10	(Initial value: 0000 0000)	
	t									1
	Horizont	al display	start posi	tion speci	fication					Write only
ORVS1	7	6	5	4	3	2	1	0		
(00F82H)	VS17	VS16	VS15	VS14	VS13	VS12	VS11	VS10	(Initial value: 0000 000)0)
(00F83H)	-	-	: –	. –	. –	-	VS19	VS18	(Initial value: **** **0	0)
ORVS2									•	
(00F84H)	VS27	VS26	VS25	VS24	VS23	VS22	VS21	VS20	(Initial value: 0000 000	00)
(00F85H)		_	-			_	VS29	VS28	(Initial value: **** **0	0)
ORVS3										
(00F86H)	VS37	VS36	VS35	VS34	VS33	VS32	VS31	VS30	(Initial value: 0000 000	00)
(00F87H)	-	-	-	-	-	-	VS39	VS38	(Initial value: **** **0	0)
ORVS4									•	
(00F88H)	VS47	VS46	VS45	VS44	VS43	VS42	VS41	VS40	(Initial value: 0000 000	00)
(00F89H)	-	-	-	-	-	-	VS49	VS48	(Initial value: **** **0	0)
ORVS5										
(00F8AH)	VS57	VS56	VS55	VS54	VS53	VS52	VS51	VS50	(Initial value: 0000 000	00)
(00F8BH)	-	-	-	-		-	VS59	VS58	(Initial value: **** **0	0)
ORVS6										
(00F8CH)	VS67	VS66	VS65	VS64	VS63	VS62	VS61	VS60	(Initial value: 0000 000)0)
(00F8DH)	-	_		+ : –	 ; _		VS69	VS68	(Initial value: **** **0	,
ORVS7									<u>1</u>	
(00F8EH)	VS77	VS76	VS75	VS74	VS73	VS72	VS71	VS70	(Initial value: 0000 000	00)
(00F8FH)	: _		: _	: _	: _	-	VS79	VS78	(Initial value: **** **0	0)
ORVS8									•	
(00F90H)	VS87	VS86	VS85	VS84	VS83	VS82	VS81	VS80	(Initial value: 0000 000	00)
(00F91H)					:		VS89	VS88	(Initial value: **** **0	0)
ORVS9										
(00F92H)	VS97	VS96	VS95	VS94	VS93	VS92	VS91	VS90	(Initial value: 0000 000	00)
(00F93H)						: –	VS99	VS98	(Initial value: **** **0	0)
ORVS10										
(00F94H)	VS107	VS106	VS105	VS104	VS103	VS102	VS101	VS100	(Initial value: 0000 000	00)
(00F95H)				:	:		VS109	VS108	(Initial value: **** **0	0)
ORVS11										
(00F96H)	VS117	VS116	VS115	VS114	VS113	VS112	VS111	VS110	(Initial value: 0000 000)0)
(00F97H)	-	_	-	-	-	_	VS119	VS118	(Initial value: **** **0	,
									. ·	
ORVS12 (00F98H)	VS127	VS126	VS125	VS124	VS123	VS122	VS121	VS120	(Initial value: 0000 000)0)
(00F99H)	-		-			-	VS129	VS128	(Initial value: **** **0	,
· · · ·								1	1 [`]	,
	VSn8 to	0 Ver	tical displa	ay start po	osition for	line n				Write only
	VSn8 to 0 Vertical display start position for line n									

(n: 1 to 12)

Note 1: If display lines are overlapped each other, previous display line is enabled and next line is disabled. Set the vertical display start position not to overlap display lines.

Note 2: Transfer the contents of vertical display start position registers into OSD circuit before a position of the scanning line coincides with their own vertical display start position.

TOSHIBA

ORCS4	7	6	5	4	3	2	1	0			
(00F9AH)	Cې	4	CS	53	C	Ş2	CS	51	(Initial value: 0000	0000)
ORCS8 (00F9BH)	CŞ	3	CS	§7	C	Ş6	CS	65	(Initial value: 0000	0000)
ORCS12					n		r				
(00F9CH)	CS1	2	CS	11	CS	10	CS	S9	(Initial value: 0000	0000)
						00: Di	splay off				
	00-	Chara	cter size a	and displa	y		irge size				Write
	CSn	on/off	for line n		-		iddle size				only
						11: Sr	nall size			(n: 1	to 12)
										(11. 1	10 12)
OREULA8					·	·			1		
(00F9DH) OREULA12	EULA8	EULA7	EULA6	EULA5	EULA4	EULA3	EULA2	EULA1	(Initial value: 0000		
(00F9EH)					EULA12	EULA11	EULA10	EULA9	(Initial value: ****	0000)	
	EULAn	Under	line for dis	splay line	for	0: Di	splay off				
	LOLAII	line n				1: Di	splay on				
										(n: 1	to 12)
OREFR8	7	6	5	4	3	2	1	0			
(00F9FH)	EFR8	EFR7	EFR6	EFR5	EFR4	EFR3	EFR2	EFR1	(Initial value: 0000	0000)
OREFR12 (00FA0H)	· _ ·				EFR12	EFR11	EFR10	EFR9	(Initial value: ****	0000)	
		Fringir	ng enable	specifica	tion	0: Di	sable fring	aina			Write
	EFRn		er for line i				hable fring				only
										(n: 1	to 12)
ORSLO4											
(00FA2H)	SLO	4	SL	O3	SL	02	SL	01	(Initial value: 0000	0000)
ORSLO8					_	-	_		(,
(00FA3H)	SLO	SLO8 SLO7 S				06	SL	O5	(Initial value: 0000	0000)
ORSLO12					1		1				
(00FA4H)	SLO	SLO12 SLO11 S					SL	09	(Initial value: 0000	0000)
						00: No	o solid spa	ce displa	у		
	SLOn	Solid s	space for I	ine n		01: Solid space display left 10: Solid space display right					Write
			•						ght ft and right		only

(n: 0 to 12)

ORBK	7	6 5	4	3	2	1	0			
(00FA5H)	<u> </u>	RBDT GBDT	BBDT		RFDT	GFDT	BFDT	(Initial value:	0000 0000)
	RBDT/ GBDT/ BBDT	Background col	or select		000: Setting color of ORCPT0 001: Setting color of ORCPT1 010: Setting color of ORCPT2 011: Setting color of ORCPT3 000: Setting color of ORCPT4 101: Setting color of ORCPT5 110: Setting color of ORCPT6 111: Setting color of ORCPT7					
	RFDT/ GFDT/ BFDT	Fringing color s	elect		000: Sett 001: Sett 010: Sett 011: Sett 000: Sett 101: Sett 110: Sett	ing color ing color ing color ing color ing color ing color ing color	of ORCPT of ORCPT of ORCPT of ORCPT of ORCPT of ORCPT of ORCPT of ORCPT	-0 -1 -2 -3 -4 -5 -6		Write only

ORACL	7	6	5	4	3	2	1	0				
(00FA6H)	PISEL2	ACLR2	ACLG2	ACLB2	PISEL1	ACLR1	ACLG1	ACLB1	(Initial value: 0000 0000))		
	ACLR2/ ACLG2/ ACLB2		2 plane co	lor select		000: Setting color of ORCPT0 001: Setting color of ORCPT1 010: Setting color of ORCPT2 011: Setting color of ORCPT3 000: Setting color of ORCPT4 101: Setting color of ORCPT5 110: Setting color of ORCPT6						
	ACLR1/ ACLG1/ ACLB1	1/ Area 1 plane color select				000: Set 001: Set 010: Set 011: Set 000: Set 101: Set 110: Set	ting color (ting color (ting color (ting color (ting color (ting color (ting color (of ORCPT7 of ORCPT1 of ORCPT2 of ORCPT2 of ORCPT2 of ORCPT2 of ORCPT2 of ORCPT2) 1 2 3 4 5 5	Write only		
	PISEL2					0: Not assign half transparency for area 2 plane 1: Assign half transparency for area 2 plane						
	PISEL1					0: Not assign half transparency for area 1 plane 1: Assign half transparency for area 1 plane						

(00FBBH) VDPOL HDPOL YBLII RGBII YIV BLIV RGBIV IIV (Initial value: 0000 0000) VDPOL VD input polarity select 0: Non-invert input signal 1: Invert input signal 1 HDPOL HD input polarity select 0: Non-invert input signal 1 Invert input signal YBLII Y/BLIN input polarity select 0: Active high 1 Invert input signal 1 YBLII Y/BLIN input polarity select 0: Active high 1 Active low 1 RGBII RIN, GIN, BIN input polarity select 0: Active high 1 Active low 0 YIV Y output polarity select 0: Active high 1 Active low 0 BLIV BL output polarity select 0: Active high 1 Active low 0 IIV I output polarity select 0: Active high 1 Active low 0 IVV I output polarity select 0: Active high 1 Active low 0 Active low 0 IVV I output polarity select 0: Active high 1 Active low <td< th=""><th>ORIV</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th><th></th></td<>	ORIV	7	6	5	4	3	2	1	0			
VDPOL VD PoL VD Polarity select 1: Invert input signal HDPOL HD input polarity select 0: Non-invert input signal YBLII Y/BLIN input polarity select 0: Active high YBLII Y/BLIN input polarity select 0: Active high YV Y output polarity select 0: Active high YIV Y output polarity select 0: Active high YACTOR RGBIV R, G, B output polarity select 0: Active high YACTOR Y 6 5 4 3 2 0 ORDMA T 6 5 4 3 2 1 0 (00024H) DMA7 DMA6 DMA5 DMA4 DMA3 DMA1 DMA8 (Initial value: **** ***0)	(00FBBH)	VDPOL	HDPOL	YBLII	RGBII	YIV	BLIV	RGBIV	IIV	(Initial value: 0000 0000)		
HD POL HD input polarity select 1: Invert input signal YBLII Y/BLIN input polarity select 0: Active high 1: Active low 0: Active high RGBII RIN, GIN, BIN input polarity select 0: Active high YIV Y output polarity select 0: Active high YIV Y output polarity select 0: Active high BLIV BL output polarity select 0: Active high 1: Active low 0: Active high RGBIV R, G, B output polarity select 0: Active high 1: Active low 0: Active high IIV I output polarity select 0: Active high 1: Active low 0: Active high IIV I output polarity select 0: Active high 1: Active low 0: Active high IIV I output polarity select 0: Active high 1: Active low 0: Active high (00024H) DMA7 DMA6 DMA3 DMA2 DMA1 DMA0 (00025H) - - - - DMA8 (Initial value: **** ***0)		VDPOL	\overline{VD} in	put polari	ty select		1 5					
YBLIN Y/BLIN input polarity select 1: Active low Write RGBII RIN, GIN, BIN input polarity select 0: Active high Write YIV Y output polarity select 0: Active high Only YIV Y output polarity select 0: Active high Only BLIV BL output polarity select 0: Active high Only BLIV BL output polarity select 0: Active high Only RGBIV R, G, B output polarity select 0: Active high Only IIV I output polarity select 0: Active high Only IIV I output polarity select 0: Active high Only IIV I output polarity select 0: Active high Only IIV I output polarity select 0: Active high Only IIV I output polarity select 0: Active high Only IIV I output polarity select 0: Active high Only IIV I output polarity select 0: Active high Only (00024H) DMA7 DMA6 DMA5 DMA4 DMA3 DMA1 DMA8 (Initial		HDPOL	HD in	put polari	ty select							
RGBII RIN, GIN, BIN input polarity select 1: Active low Write YIV Y output polarity select 0: Active high 0: Active high 1: Active low 0: Active high 0: Active high 0: Active high BLIV BL output polarity select 0: Active high 0: Active high 1: Active low 0: Active high 1: Active low RGBIV R, G, B output polarity select 0: Active high 1: Active low 0: Active high 1: Active low IIV I output polarity select 0: Active high 1: Active low 0: Active high 1: Active low ORDMA 7 6 5 4 3 2 1 (00024H) DMA7 DMA6 DMA4 DMA3 DMA2 DMA1 DMA0 (Initial value: 0000 0000) (Initial value: **** ***0)		YBLII	Y/BLI	N input po	larity sele	ct	-					
YIV Y output polarity select 1: Active low BLIV BL output polarity select 0: Active high 1: Active low 0: Active high RGBIV R, G, B output polarity select 0: Active high 1: Active low 0: Active high 0: Active high 1: Active low (00024H) DMA5 DMA4 DMA2 DMA1 DMA8 (Initial value: **** ***0) (Initial value: **** ****0) Write		RGBII	RIN, G	GIN, BIN ir	nput polar	ity select		0			Write	
BLIV BL output polarity select 1: Active low RGBIV R, G, B output polarity select 0: Active high 1: Active low 1: Active low IIV I output polarity select 0: Active high 0: Active high 1: Active low IIV I output polarity select 0: Active high 0: Active low 0: Active low IIV I output polarity select 0: Active low ORDMA 7 6 5 4 3 2 1 0 (00024H) DMA7 DMA6 DMA5 DMA4 DMA3 DMA2 DMA1 DMA0 (Initial value: 0000 0000) (00025H) DMA8 (Initial value: **** ***0) Write		YIV	Y outp	out polarity	/ select		5					
RGBIV R, G, B output polarity select 1: Active low IIV I output polarity select 0: Active high 1: Active low ORDMA 7 6 5 4 3 2 1 0 (00024H) DMA7 DMA6 DMA5 DMA4 DMA3 DMA2 DMA1 DMA0 (Initial value: 0000 0000) (00025H) DMA8 (Initial value: **** ***0)		BLIV	BL ou	tput polari	ty select			0				
IV I output polarity select 1: Active low ORDMA 7 6 5 4 3 2 1 0 (00024H) DMA7 DMA6 DMA5 DMA4 DMA3 DMA2 DMA1 DMA0 (Initial value: 0000 0000) (00025H) - - - - - DMA8 (Initial value: **** ***0)		RGBIV	R, G,	B output p	olarity se	lect		0				
(00024H) DMA7 DMA6 DMA5 DMA4 DMA3 DMA2 DMA1 DMA0 (Initial value: 0000 0000) (00025H)		IIV	l outpu	ut polarity	select		-					
(00025H)	ORDMA	7	6	5	4	3	2	1	0			
Write	(00024H)	DMA7	DMA6	DMA5	DMA4	DMA3	DMA2	DMA1	DMA0	(Initial value: 0000 0000))	
DMAn Display memory address Write	(00025H)								DMA8	(Initial value: **** ***0))	
(n: 0 to 8)		DMAr	n Disj	olay mem	ory addre	SS				(n:	only	

Note: It is necessary to write all bits of display memory address, writing DMA7 to DMA0 after DMA8, when writing display address.

ORDSN	7	6	5	4	3	2	1	0				
(0001DH)		SLNT	EUL	BLF	ECBKD	RDT	GDT	BDT	(Initial value: **** ****)			
	SLNT	Slant o	enable sp er	ecificatior	۱	0: Disabl 1: Enable						
	EUL	Under registe	line enabl er	e specific	ation	0: Disabl 1: Enable						
	BLF	Blinkir registe	ng enable er	specificat	tion		e blinking e blinking					
	ECBKD		cter-spec specificat	•	round	0: Disable backgournd color display 1: Enable backgournd color display						
	RDT/ GDT/ BDT	Chara	cter color	select		1: Enable backgournd color display 000: Setting color of ORCPT0 001: Setting color of ORCPT1 010: Setting color of ORCPT2 011: Setting color of ORCPT3 000: Setting color of ORCPT4 101: Setting color of ORCPT5 110: Setting color of ORCPT6 111: Setting color of ORCPT7						

Note: To display a background color, write "1" to EBKGD (bit 7 in the ORRCL register) to enable the background function enable register for the entire screen.

ORCRA (0001EH) (0001FH)	7	6 5 4 3 CRA6 CRA5 CRA4 CRA3	2 1 0 CRA2 CRA1 CRA0 (Initial value: **** **** CRA8 (Initial value: **** ****	
	CRAn	Character code		Read/ Write
			· ·	0 to 8)
	Note: Wr	ite or read CRA7 to CRA0 af	ter write or read CRA8.	
ORWVSH	7	6 5 4 3	2 1 0	
(00FBCH)	WVSH7 V	VVSH6 WVSH5 WVSH4 WVSH3		,
(00FBDH)	:;		· – WVSH9 WVSH8 (Initial value: **** **00)
	WVSLn	Window upper limit position		Write
			(n:	only 0 to 9)
			(0 10 0)
ORWVSL	7	6 5 4 3	2 1 0	
(00FBEH)	WVSL7 V	VVSL6 WVSL5 WVSL4 WVSL3		,
(00FBFH)	· _ ·	_ · _ · _ · _ · _	WVSL9 WVSL8 (Initial value: **** **00))
	WVSLn	Window lower limit position		Write
				only
			(n:	0 to 9)
ORDON	7	6 5 4 3	2 1 0	
(00F80H)	<u> </u>	– EPAL100 ROMACH ECHDSN	RGWR EWDW DON (Initial value: **00 0000))
	EPAL100	PAL100 mode specification	0: PAL100 mode disable	
	EFALIOU	register	1: PAL100 mode enable	
			0: 383 monochrome font characters 96 color font characters	
	ROMACH	Monochrome/color font area switching register	1: 447 monochrome font characters	
			64 color font characters	
		Character-specific background	0: Character-specific background color setting off	
	ECHDSN	color setting on/off specification register	1: Character-specific background color setting on	Read/ Write
			0: (Initial setting)	
	RGWR	Data transfer control OSD register	 Written data is transferred to the OSD circuit (cleared to "0" after the transfer). 	
	EWDW	Window enable specification	0: Window specification off	
		register	1: Window specification on	
	DON	Display on/off specification	0: Display off	
		register	1: Display on	

Note 1: *: Don't care

Note 2: All OSD control registers cannot use the read-modify-write instructions. (Bit manipulation instructions such as SET, CLR, etc. and logical operation such as AND, OR, etc.)

ORRCL	7	6	5	4	3	2	1	0		
(00FA7H)	EBKGD	EXBL	AON2	AON1	_	RCLR	RCLG	RCLB	(Initial value: 0000 *000)	
	EBKGD	Backg	round fun	ction enab	le	0: No ba	ckground	function		
	LDIGD	specif	ication reg	gister		1: Backg	round fun	ction enable	e	
	EXBL	Full-ra	ster blank	king enable	9	0: No Fu	ll-raster bl	anking		
	LADL	specif	ication reg	gister		1: Full-ra	ister blank	ing		
	AON2	Area 2	2 plane dis	splay enab	le	0: No are	ea 2 plane	display		
	AONZ	specif	ication reg	gister		1: Area 2	2 plane dis	play enable	9	
	AON1	Area 1	I plane dis	splay enab	le	0: No are	ea 1 plane	display		
	AONT	specif	ication reg	gister		1: Area 1	plane dis	play enable	9	Write
						000: Set	ting color	of ORCPT0		only
						001: Set	ting color	of ORCPT1		
						010: Set	ting color	of ORCPT2		
	RCLR/ RCLG/	Pasto	r nlana co	lor select		011: Set	ting color	of ORCPT3		
	RCLB	Nasie		IOI SEIECI		000: Set	ting color	of ORCPT4		
						101: Set	ting color	of ORCPT5		
						110: Set	ting color	of ORCPT6		
						111: Sett	ting color	of ORCPT7		

TOSHIBA

AVE2n

Vertical end point for area 2 plane

ORAHS1	7	6	5	4	3	2	1	0			
(00FA8H)	AHS17	AHS16	AHS15	AHS14	AHS13	AHS12	AHS11	AHS10	(Initial value: 0000	0000)
(00FA9H)						· _ ·		AHS18	(Initial value: ****	***0))
ORAHE1											
(00FAAH)	AHE17	AHE16	AHE15	AHE14	AHE13	AHE12	AHE11	AHE10	(Initial value: 0000	0000)
(00FABH)	· _					· _		AHE18	(Initial value: ****		,
()	·							_	(1
	AHS1		rizontal sta	· ·							Write
	AHE1	n Hor	rizontal en	d point fo	r area 1 p	lane					only
										(n:	0 to 8)
ORAVS1 (00FACH)	AVS17	AVS16	AVS15	AVS14	AVS13	AVS12	AVS11	AVS10	(Initial value: 0000	0000	`
(00FACH) (00FADH)		AV310	AV313	AV314	AV313	AV312	AVS11 AVS19	AVS10 AVS18	(Initial value: 0000 (Initial value: ****		,
							AV019	A0010	(initial value.		
ORAVE1	<u> </u>										
(00FAEH)	AVE17	AVE16	AVE15	AVE14	AVE13	AVE12	AVE11	AVE10	(Initial value: 0000		
(00FAFH)	·						AVE19	AVE18	(Initial value: ****	**00)	
	AVS1	n Ver	tical start	point for a	irea 1 plai	ne					Write
	AVE1	n Ver	tical end p	point for a	rea 1 plan	е					only
										(n:	0 to 9)
ORAHS2		i	i	i		ı — — —	ı — — —	·1			
(00FB0H)	AHS27	AHS26	AHS25	AHS24	AHS23	AHS22	AHS21	AHS20			
(00FB1H)			:	:		: _		AHS28			
ORAHE2											
(00FB2H)	AHE27	AHE26	AHE25	AHE24	AHE23	AHE22	AHE21	AHE20	(Initial value: 0000	0000)
(00FB3H)								AHE28	(Initial value: ****	***0))
	AHS2		rizontal sta	ort point fo	r aroa 2 r						\\/rito
	AHE2		izontal en								Write only
					alea z p					(n·	0 to 8)
										(0 10 0)
ORAVS2											
(00FB4H)	AVS27	AVS26	AVS25	AVS24	AVS23	AVS22	AVS21	AVS20	(Initial value: 0000	0000)
(00FB5H)				· _	_	· _	AVS29	AVS28	(Initial value: ****	**00))
							•				
ORAVE2 (00FB6H)	AVE27	AVE26	AVE25	AVE24	AVE23	AVE22	AVE21	AVE20	(Initial value: 0000	0000)
(00FB7H)					AVL23		AVE21	AVE28	(Initial value: ****		
							////223		(initial value.		,
	AVS2	n Ver	tical start	point for a	irea 2 plai	ne					Write

only

ORP6S	7	6	5	4	3	2	1	0		
(00FBAH)	P67S	P66S	P65S	P64S	PIDS	YBLCS	М	PXS	(Initial value: 0000 0000)
	P67S to P64S	P6 por	rt output s	elect		0: R, G, B 1: Port co	-	signal outpo output	ut	
	PIDS	l pin o	utput sele	ct		0: I signal 1: Port cor		output		
	YBLCS	Y/BL s	signal sele	ect		0: Y signa 1: BL sign	•			Write
	MPXS	R, G, I	B, Y/BL si	gnal selec	t	higher 01: Output 10: Output	priority. of sign of sign aneous) al from inte al from ext	gnal from the OSD circuit has ernal OSD circuit ernally input kternally input signal has higher	only
ORETC (00FB8H)	7 VDSMD	6 "0"	5 BKMF	4 ESMZ	3 "0"	2 MFYWR	1 MBK	0 RDWRV	(Initial value: 0000 0000)
	VDSMD	Scan r	mode sele	ect		0: Normal 1: Double		iode		
	BKMF	Blinkir	ng master			0: Double 1: Enable				
	ESMZ	Smoot registe	0	ole specifi	cation	0: Disable 1: Enable				
	MFYWR	Displa select		read mod	de	0: Normal 1: Read-m		vrite-mode		Write only
	MBK	Displa switch	y memory ing	v bank		options			r code or character display de and character display	
	RDWRV		write mod I mode	e select a	t				ay memory ay memory	

Note: Clear "0" to bit 6 and 3 in ORETC.

ORIRC	7	6 5 4 3	2 1 0	
(00FB9H)		SDV	ISDC (Initial value: ***0 0000))
	SVD	Interrupt source select	 0: Interrupt request by ISDC value 1: Interrupt request at falling edge of VD signal 	
	ISDC	Interrupt generation line select	 When the line display of the ISDC value ends (with the falling edge of HD signal) while SVD = 0, interrupt request is generated. 0000: Request interrupt when display of low-order 4 bits "0000" of DCTR ends. 0001: Low-order 4 bits "0010" of DCTR 0010: Low-order 4 bits "0010" of DCTR 0011: Low-order 4 bits "0011" of DCTR 0102: Low-order 4 bits "0100" of DCTR 0103: Low-order 4 bits "0101" of DCTR 0104: Low-order 4 bits "0101" of DCTR 0105: Low-order 4 bits "0101" of DCTR 0106: Low-order 4 bits "0111" of DCTR 0107: Low-order 4 bits "0111" of DCTR 0108: Low-order 4 bits "1000" of DCTR 0109: Low-order 4 bits "1001" of DCTR 01001: Low-order 4 bits "1001" of DCTR 01011: Low-order 4 bits "1001" of DCTR 10011: Low-order 4 bits "1011" of DCTR 10111: Low-order 4 bits "1011" of DCTR 10112: Low-order 4 bits "1111" of DCTR 11111: Low-order 4 bits "1111" of DCTR 11111: Low-order 4 bits "1111" of DCTR 	Write only
ORIRC (00FB9H)	[]	- 1 1 [, DCTR , (Initial value: **** 0000))
	DCTR	Display line counter	 0000: No line display or when the display of the 16th line ends. 0001: 1st line display ends. 0010: 2nd line display ends. 0011: 3rd line display ends. 0100: 4th line display ends. 0111: 5th line display ends. 0111: 7th line display ends. 1000: 8th line display ends. 1001: 9th line display ends. 1010: 10th line display ends. 1011: 11th line display ends. 1101: 12th line display ends. 1101: 13th line display ends. 1101: 13th line display ends. 1111: 15th line display ends. 	Read only

Note: The display line counter also increments when a line with all blank data or a line with display off is specified.

If display lines are overlapped each other, previous display line is enabled and next line is disabled. At this time, the display line counter also increments.

ORDCSC	7	6 5	4 3	2	1	0				
(00FC4H)	DCSC8 D	CSC7 DCSC6	DCSC5 DCSC4	DCSC3	DCSC2	DCSC1	(Initial valu	e: 0000 (0000)	
(00FC5H)			– DCSC12	CDSC11	CDSC10	CDSC9	(Initial valu	e:**** C	0000)	
	DCSCn	n: Double-height row n n: 1 to 12	specification for	characte 1: Display	er display i a double	is specified	haracter v			Write only
	Note: To	o display double-h	eight characters, v	write "10" to	o CSn (n =	= 1 to 12) ii	n the ORC	Sm (m = -	4, 8, 12) re	gister,
	sp	ecify the medium	character size, an	d write "1"	to DCSCr	n (n = 1 to ′	12).			
	7	6 5	4 3	2	1	0				
ORCPT0 (00FC6H)	CPT0MD1	Fixed at 0 CPT0R1	CPTOR0 CPT0G1	CPT0G0	CPT0B1	CPT0B0	(Initi	al value: (0000 0000)
				·i						
ORCPT1 (00FC7H)	· · · · · ·	6 5 - CPT1R1	4 3 CPT1R0 CPT1G1	2 CPT1G0	1 CPT1B1 (0 CPT1B0	(Initi	al value:	**00 0000)
	·	• • • • • · · · · · · · · · · · · · · ·		11			(1111		00 0000	/
ORCPT2 (00FC8H)	7	6 5 - CPT2R1	4 3 CPT2R0 CPT2G1	2 CPT2G0	1 CPT2B1	0 CPT2B0	(Initi	al value:	**00 0000)
(0010811)		- GFT2RT		CF 1200		GF 12B0	(initi	ai vaiue.	00 0000)
ORCPT3	7	6 5	4 3	2	1	0				`
(00FC9H)	!!	- CPT3R1	CPT3R0 CPT3G1	CP13G0	CPT3B1	CP13B0	(Initi	al value:	**00 0000)
ORCPT4	7	6 5	4 3	2	1	0				
(00FCAH)		– CPT4R1	CPT4R0 CPT4G1	CPT4G0	CPT4B1	CPT4B0	(Initi	al value:	**00 0000)
ORCPT5 (00FCBH)	7	6 5 – CPT5R1	4 3 CPT5R0 CPT5G1	2 CPT5G0	1 CPT5B1	0 CPT5B0	(Initi	al value:	**00 0000)
()	•									/
				•						
ORCPT6 (00FCCH)	7	6 5 - CPT6R1	4 3 CPT6R0 CPT6G1	2 CPT6G0	1 CPT6B1 (0 CPT6B0	(Initi	al value:	**00 0000)
	7 						(Initi	al value:	**00 0000)
(00FCCH)	7	- CPT6R1	CPT6R0 CPT6G1	CPT6G0 2	CPT6B1	CPT6B0			**00 0000	
(00FCCH) ORCPT7	7 	- CPT6R1	CPT6R0 CPT6G1 4 3 CPT7R0 CPT7G1	CPT6G0 2	CPT6B1	СРТ6В0 0				
(00FCCH) ORCPT7	7 	CPT6R1 CPT7R1 CPT7R1 Z7-color mode specification	CPT6R0 CPT6G1 4 3 CPT7R0 CPT7G1 0: 8-color mode	CPT6G0 2 CPT7G0	CPT6B1	СРТ6В0 0) Write
(00FCCH) ORCPT7	7 	- CPT6R1 - - CPT7R1 27-color mode	CPT6R0 CPT6G1 4 3 CPT7R0 CPT7G1	CPT6G0 2 CPT7G0	CPT6B1	СРТ6В0 0)
(00FCCH) ORCPT7	7 	CPT6R1 CPT7R1 CPT7R1 Z7-color mode specification	CPT6R0 CPT6G1 4 3 CPT7R0 CPT7G1 0: 8-color mode 1: 27-color mode	CPT6G0 2 CPT7G0	CPT6B1	СРТ6В0 0		al value: *) Write
(00FCCH) ORCPT7		CPT6R1 CPT7R1 CPT7R1 Z7-color mode specification	CPT6R0 CPT6G1 4 3 CPT7R0 CPT7G1 0: 8-color mode 1: 27-color mode 1: 27-color mode CPTC CRTxR1 = 0, CR CRT6	CPT6G0 2 CPT7G0 DMD1 = 0 TxR0 = 0:	CPT6B1	0 CPT7B0	(Initi CPTOM	al value:	**00 0000) Write
(00FCCH) ORCPT7	7 7 CPT0MD1 CPTxR0 CPTxR1	CPT6R1 CPT6R1 CPT7R1 CPT7R1 27-color mode specification register R luminance specification	CPT6R0 CPT6G1 4 3 CPT7R0 CPT7G1 0: 8-color mode 1: 27-color mode CPTC	CPT6G0 2 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1:	CPT6B1	0 CPT7B0 CPT7B0	(Initi	Al value: <u> 101 = 1</u> (R0 = 0: N (R0 = 1: [**00 0000) Write
(00FCCH) ORCPT7	CPTxR0	CPT6R1 CPT7R1 CPT7R1 CPT7R1 CPT7R1 CPT7R1 CPT7R1 R luminance	CPT6R0 CPT6G1 4 3 CPT7R0 CPT7G1 0: 8-color mode 1: 27-color mode 1: 27-color mode CPTC CRTxR1 = 0, CR CRTxR1 = 0, CR CRTxR1 = 1, CR CRTxR1 = 1, CR	CPT6G0 2 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1: TxR0 = 0: TxR0 = 1:	CPT6B1	0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxR1	(Initi CPTON = 0, CRT) = 0, CRT) = 1, CRT) = 1, CRT)	Al value: AD1 = 1 (R0 = 0: N (R0 = 1: I (R0 = 0: I (R0 = 1: L	**00 0000 No output Dark red Dark red Light red) Write
(00FCCH) ORCPT7	CPTxR0	CPT6R1 CPT6R1 CPT7R1 CPT7R1	CPT6R0 CPT6G1 4 3 CPT7R0 CPT7G1 0: 8-color mode 1: 27-color mode 1: 27-color mode CPTC CRTxR1 = 0, CR CRTxR1 = 0, CR CRTxR1 = 0, CR CRTxR1 = 1, CR	2 CPT7G0 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1: TxR0 = 0: TxR0 = 1: TxR0 = 1: TxR0 = 0:	CPT6B1	0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxR1	(Initi CPTON = 0, CRT> = 0, CRT> = 0, CRT> = 1, CRT>	Al value: Al value:	**00 0000 No output Dark red Dark red Light red No output) Write
(00FCCH) ORCPT7	CPTxR0 CPTxR1	CPT6R1 CPT6R1 CPT7R1 CPT7R1 27-color mode specification register R luminance specification register	$\begin{array}{c c} CPT6R0 & CPT6G1 \\ \hline 4 & 3 \\ \hline CPT7R0 & CPT7G1 \\ \hline 0: 8-color mode \\ \hline 1: 27-color mode \\ \hline 1: 27-color mode \\ \hline CPTC \\ CRTxR1 = 0, CR \\ CRTxR1 = 0, CR \\ CRTxR1 = 1, CR \\ CRTxR1 = 1, CR \\ CRTxG1 = 0, CR \\ CRTxG1 = 0, CR \\ CRTxG1 = 0, CR \\ CRTxG1 = 1, CR \\ \hline \end{array}$	2 CPT7G0 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1: TxR0 = 1: TxG0 = 0: TxG0 = 1: L TxG0 = 0: L	CPT6B1	0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR2 CRTxG2 CRTxG2 CRTxG2 CRTxG2	(Initi CPTON = 0, CRT> = 0, CRT> = 1, CRT> = 1, CRT> = 0, CRT> = 0, CRT> = 0, CRT> = 0, CRT>	al value: <u>AD1 = 1</u> <u>(R0 = 0: N</u> <u>(R0 = 1: L</u> <u>(R0 = 1: L</u> <u>(G0 = 0: I</u> <u>(G0 = 0: C</u>)	**00 0000 No output Dark red Dark red Light red No output Dark green Dark green) Write only
(00FCCH) ORCPT7	CPTxR0 CPTxR1 CPTxG0	CPT6R1 CPT6R1 CPT7R1 CPT7R1 Z7-color mode specification register R luminance specification register G luminance specification register	CPT6R0 CPT6G1 4 3 CPT7R0 CPT7G1 0: 8-color mode 1: 27-color mode 1: 27-color mode CPTC CRTxR1 = 0, CR CRTxR1 = 0, CR CRTxR1 = 1, CR CRTxR1 = 1, CR CRTxR1 = 1, CR CRTxR1 = 1, CR CRTxR1 = 0, CR CRTxR1 = 0, CR	2 CPT6G0 2 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1: TxR0 = 1: TxG0 = 0: TxG0 = 1: TxG0 = 0: I TxG0 = 1: I	CPT6B1 (1 CPT7B1 (CPT7B1 (Light red Light red Light red Light red Light red Light greer Light greer Light greer	CPT6B0 0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR2 CRTxG2 CRTxG2 CRTxG2 CRTxG2 CRTxG2 CRTxG2	(Initi CPTON = 0, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 0, CRT) I = 0, CRT)	$\frac{AD1 = 1}{AD1 = 1}$ $\frac{AD1 = 1}{AB0 = 0: 1}$ $\frac{AB0 = 0: 1}{AB0 = 0: 1}$	**00 0000 No output Dark red Dark red Light red No output Dark green Dark green ight green) Write Only Write
(00FCCH) ORCPT7	CPTxR0 CPTxR1 CPTxG0 CPTxG1 CPTxB0	CPT6R1 G LOPT7R1 CPT7R1 CPT7 CPT7R1 CPT7R1 CPT7R1 CPT7R1 CPT7R1 CPT7R	$\begin{array}{c c} CPT6R0 & CPT6G1 \\\hline 4 & 3 \\\hline CPT7R0 & CPT7G1 \\\hline 0: 8-color mode \\\hline 1: 27-color mode \\\hline 1: 27-color mode \\\hline CRTxR1 = 0, CR \\\hline CRTxR1 = 0, CR \\\hline CRTxR1 = 1, CR \\\hline CRTxR1 = 1, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 1, CR \\\hline CRTxG1 = 1, CR \\\hline CRTxG1 = 1, CR \\\hline CRTxB1 = 0, CR \\\hline \end{array}$	2 CPT7G0 2 CPT7G0 CPT	CPT6B1 (1 CPT7B1 (CPT7B1 (Light red Light red Light red Light red Light greer Light greer Light greer Light greer Light blue	CPT6B0 0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxR2 CRTxG2 CRTxG2 CRTxG2 CRTxG2 CRTxG2 CRTxG3 CRTxG3 CRTxG3 CRTxG4 CRTxB1 CRTxB1	(Initi CPTOM = 0, CRT> = 0, CRT> = 1, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 1, CRT> = 1, CRT> = 1, CRT> = 0, CRT> = 0, CRT> = 0, CRT>	al value: $\frac{MD1 = 1}{(R0 = 0: N)}$ $\frac{R0 = 1: L}{(R0 = 0: C)}$ $\frac{R0 = 1: L}{(R0 = 0: C)}$ $\frac{R0 = 1: L}{(R0 = 0: N)}$ $\frac{R0 = 0: N}{(R0 = 1: C)}$	**00 0000 No output Dark red Dark red Light red No output Dark green Dark green ight green ight green Jo output Dark blue) Write Only Write
(00FCCH) ORCPT7	CPTxR0 CPTxR1 CPTxG0 CPTxG1	 CPT6R1 5 CPT7R1 27-color mode specification register R luminance specification register G luminance specification register B luminance 	$\begin{array}{c c} CPT6R0 & CPT6G1 \\ \hline 4 & 3 \\ \hline CPT7R0 & CPT7G1 \\ \hline 0: 8-color mode \\ 1: 27-color mode \\ \hline 1: 27-color mode \\ \hline CRTxR1 = 0, CR \\ CRTxR1 = 0, CR \\ CRTxR1 = 1, CR \\ CRTxR1 = 1, CR \\ CRTxG1 = 0, CR \\ CRTxG1 = 0, CR \\ CRTxG1 = 0, CR \\ CRTxG1 = 1, CR \\ CRTxB1 = 0, CR \\ \hline \end{array}$	2 CPT6G0 2 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1: TxR0 = 0: TxR0 = 1: TxG0 = 0: TxG0 = 0: TxB0 = 0: T	CPT6B1 (1 CPT7B1 (CPT7B1 (CPT7B1 (CPT7B1 (CPT7B1 (CPT7B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT7B1 (CPT	CPT6B0 0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxG2 CRTxG2 CRTxG2 CRTxG2 CRTxG3 CRTxG1 CRTxG1 CRTxG1 CRTxG1 CRTxB1 CRTxB1 CRTxB1	(Initi CPTOM = 0, CRT> = 0, CRT> = 1, CRT> = 1, CRT> = 0, CRT> = 0, CRT> = 1, CRT> = 1, CRT> = 1, CRT> = 1, CRT>	al value: $\frac{MD1 = 1}{(R0 = 0: N)}$ $\frac{R0 = 1: L}{(R0 = 0: C)}$ $\frac{R0 = 1: L}{(R0 = 0: C)}$ $\frac{R0 = 1: L}{(R0 = 0: C)}$ $\frac{R0 = 0: C}{(R0 = 0: C)}$ $\frac{R0 = 0: C}{(R0 = 0: C)}$	**00 0000 No output Dark red Dark red Light red Dark green Dark green ight green ight green Jo output Dark blue Dark blue) Write Only Write
(00FCCH) ORCPT7	CPTxR0 CPTxR1 CPTxG0 CPTxG1 CPTxB0	CPT6R1 CPT6R1 CPT7R1 CPT6R1 CPT7R1 CPT7R1	$\begin{array}{c c} CPT6R0 & CPT6G1 \\\hline 4 & 3 \\\hline CPT7R0 & CPT7G1 \\\hline 0: 8-color mode \\\hline 1: 27-color mode \\\hline 1: 27-color mode \\\hline CRTxR1 = 0, CR \\\hline CRTxR1 = 0, CR \\\hline CRTxR1 = 1, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 1, CR \\\hline CRTxG1 = 1, CR \\\hline CRTxB1 = 0, CR \\\hline CRTxB1 = 1, CR \\\hline \end{array}$	2 CPT6G0 2 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1: TxR0 = 0: TxR0 = 1: TxG0 = 0: TxG0 = 0: TxB0 = 0: T	CPT6B1 (1 CPT7B1 (CPT7B1 (CPT7B1 (CPT7B1 (CPT7B1 (CPT7B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT6B1 (CPT7B1 (CPT	CPT6B0 0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxG2 CRTxG2 CRTxG2 CRTxG2 CRTxG3 CRTxG1 CRTxG1 CRTxG1 CRTxG1 CRTxB1 CRTxB1 CRTxB1	(Initi CPTOM = 0, CRT> = 0, CRT> = 1, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 1, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 0, CRT> = 1, CRT> = 0, CR	al value: $\frac{MD1 = 1}{(R0 = 0: N)}$ $\frac{R0 = 1: L}{(R0 = 0: C)}$ $\frac{R0 = 1: L}{(R0 = 0: C)}$ $\frac{R0 = 1: L}{(R0 = 0: C)}$ $\frac{R0 = 0: C}{(R0 = 0: C)}$ $\frac{R0 = 0: C}{(R0 = 0: C)}$	**00 0000 No output Dark red Dark red Light red Dark green Dark green ight green ight green Jo output Dark blue Dark blue) Write Only Write
(00FCCH) ORCPT7 (00FCDH)	CPTxR0 CPTxR1 CPTxG0 CPTxG1 CPTxB0 CPTxB1	 CPT6R1 5 CPT7R1 27-color mode specification register R luminance specification register G luminance specification register B luminance specification register 	$\begin{array}{c c} CPT6R0 & CPT6G1 \\\hline 4 & 3 \\\hline CPT7R0 & CPT7G1 \\\hline 0: 8-color mode \\\hline 1: 27-color mode \\\hline 1: 27-color mode \\\hline CRTxR1 = 0, CR \\\hline CRTxR1 = 0, CR \\\hline CRTxR1 = 1, CR \\\hline CRTxR1 = 1, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 1, CR \\\hline CRTxB1 = 1, CR \\\hline CRTxB1 = 0, CR \\\hline CRTxB1 = 1, CR \\\hline CRTTxB1 = 1, CR \\\hline CRTxB1 = 1, CR \\\hline CRTT$	2 CPT6G0 2 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1: TxR0 = 0: TxR0 = 1: TxG0 = 0: I TxG0 = 0: I TxB0 =	CPT6B1 0 1 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT7B1 0 CPT7B1	CPT6B0 0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxR2 CRTxG2 CRTxG2 CRTxG2 CRTxG1 CRTxG1 CRTxB1 CRTxB1 CRTxB1 CRTxB1	(Initi CPTOM = 0, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 1, CRT)	al value: MD1 = 1 (R0 = 0: N (R0 = 1: L) (R0 = 1: L) (G0 = 1: L) (G0 = 1: L) (G0 = 1: L) (B0 = 0: N) (B0 = 1: L) (B0 = 1: L) (B0 = 1: L)	**00 0000 No output Dark red Dark red Light red Dark green Dark green ight green ight green Jo output Dark blue Dark blue) Write Only Write
(00FCCH) ORCPT7 (00FCDH)	CPTxR0 CPTxR1 CPTxG0 CPTxG1 CPTxB0 CPTxB1	CPT6R1 G S CPT7R1 CPT6R1 content of the specification register Specification register Specification register Secondary for the specification register	$\begin{array}{c c} CPT6R0 & CPT6G1 \\ \hline 4 & 3 \\ \hline CPT7R0 & CPT7G1 \\ \hline 0: 8-color mode \\ 1: 27-color mode \\ \hline 1: 27-color mode \\ \hline CRTxR1 = 0, CR \\ CRTxR1 = 0, CR \\ CRTxR1 = 1, CR \\ CRTxR1 = 1, CR \\ CRTxG1 = 0, CR \\ CRTxG1 = 1, CR \\ CRTxB1 = 0, CR \\ CRTxB1 = 1, CR \\ CRTxB1 = 0, CR \\ CRTTxB1 = 0, CR \\ CRTTxB1 = 0, CR \\ CRTTxB1 = 0, CR \\ CRTTTRB1 \\ CRTTTR$	2 CPT6G0 2 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1: TxR0 = 0: TxR0 = 1: TxG0 = 0: TxG0 = 0: TxG0 = 0: TxG0 = 1: TxG0 = 0: TxG0 = 0: TxB0 = 0: T	CPT6B1 0 1 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT7B1 0 CPT7B1	0 CPT7B0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxG2 CRTxG2 CRTxG2 CRTxG3 CRTxG3 CRTxG1 CRTxB1 CRTXB1	(Initi CPTOM = 0, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 1, CRT)	al value: MD1 = 1 (R0 = 0: N (R0 = 1: L) (R0 = 1: L) (G0 = 1: L) (G0 = 1: L) (G0 = 1: L) (B0 = 0: N) (B0 = 1: L) (B0 = 1: L) (B0 = 1: L)	**00 0000 No output Dark red Dark red Light red Dark green Dark green Dark green light green No output Dark blue Dark blue Dark blue) Write only Write only
(00FCCH) ORCPT7 (00FCDH)	CPTxR0 CPTxR1 CPTxG0 CPTxG1 CPTxB0 CPTxB1	 CPT6R1 5 CPT7R1 27-color mode specification register R luminance specification register G luminance specification register B luminance specification register 	$\begin{array}{c c} CPT6R0 & CPT6G1 \\\hline 4 & 3 \\\hline CPT7R0 & CPT7G1 \\\hline 0: 8-color mode \\\hline 1: 27-color mode \\\hline 1: 27-color mode \\\hline CRTxR1 = 0, CR \\\hline CRTxR1 = 0, CR \\\hline CRTxR1 = 1, CR \\\hline CRTxR1 = 1, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 0, CR \\\hline CRTxG1 = 1, CR \\\hline CRTxB1 = 1, CR \\\hline CRTxB1 = 0, CR \\\hline CRTxB1 = 1, CR \\\hline CRTTxB1 = 1, CR \\\hline CRTxB1 = 1, CR \\\hline CRTT$	2 CPT6G0 2 CPT7G0 DMD1 = 0 TxR0 = 0: TxR0 = 1: TxR0 = 0: TxR0 = 1: TxG0 = 0: I TxG0 = 0: TxG0 = 1: TxG0 = 0: TxG0 = 0: TxB0 = 0:	CPT6B1 0 1 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT7B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT6B1 0 CPT7B1 0 CPT7B1	0 CPT7B0 CPT7B0 CPT7B0 CRTxR1 CRTxR1 CRTxR1 CRTxR1 CRTxG2 CRTxG2 CRTxG2 CRTxG3 CRTxG3 CRTxG1 CRTxB1 CRTXB1	(Initi CPTOM = 0, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 0, CRT) = 1, CRT) = 1, CRT) = 1, CRT)	al value: MD1 = 1 (R0 = 0: N (R0 = 1: L) (R0 = 1: L) (G0 = 1: L) (G0 = 1: L) (G0 = 1: L) (B0 = 0: N) (B0 = 1: L) (B0 = 1: L) (B0 = 1: L)	**00 0000 No output Dark red Dark red Light red Dark green Dark green Dark green light green No output Dark blue Dark blue Dark blue) Write Only Write

Register	Register			R	egister bit co	onfiguration				-	
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit contents	R/W
0001D	ORDSN	-	SLNT	EUL	BLF	ECBKD	RDT	GDT	BDT	When ECHDSN = 0 SLNT = 1: Slant enable, 0: Slant disable EUL = 1: Underline display on, 0: Underline display off When ECHDSN = 1 SLNT: Background color red EUL: Background color green BLF = 1: Blinking enable, 0: Blinking disable ECBKD = 1:Character background color display enable, Character background color display disable	R/W
0001E	ORCRA	CRA7	CRA6	CRA5	CRA4	CRA3	CRA2	CRA1	CRA0	CRAx: Character code (x: 0 to 8)	R/W
0001F		-	-	-	-	-	-	-	CRA8		
00024	ORDMA	DMA7	DMA6	DMA5	DMA4	DMA3	DMA2	DMA1	DMA0	DMAx: Display memory address setting (x: 0 to 8)	w
00025		-	-	-	-	-	-	-	DMA8		
00F80	ORDON	-	-	EPAL100	ROMACH	ECHDSN	RGWR	EWDW	DON	EPAL100 = 1: PAL100/NTSC120 select, 0: Other ROMACH: Select font number (mono font/color font) 1: 447 mono font character/64 color font character, 0: 383 mono font character/96 color font character ECHDSN = 1: Bit 6 and 5 in ORDSN is changed to character background color, 0: Bit 6 and 5 in ORDSN is character ornamentation RGWR: Writing data transfer control bit EWDW = 1: Window function enable, 0: Window function disable DON = 1: OSD display ON, 0: OSD display OFF	R/W
00F81	ORHS1	HS17	HS16	HS15	HS14	HS13	HS12	HS11	HS10	HS17 to HS10: Code horizontal display base position setting	w
00F82	ORVS1	VS17	VS16	VS15	VS14	VS13	VS12	VS11	VS10	VS19 to VS10: Code vertical display potision setting	w
00F83		-	-	-	-	-	-	VS19	VS18		
00F84	ORVS2	VS27	VS26	VS25	VS24	VS23	VS22	VS21	VS20	VS29 to VS20: Code vertical display potision setting	w
00F85		-	-	-	-	-	-	VS29	VS28		
00F86	ORVS3	VS37	VS36	VS35	VS34	VS33	VS32	VS31	V\$30	VS39 to VS30: Code vertical display potision setting	w
00F87		-	-	-	-	-	-	VS39	VS38		
00F88 00F89	ORVS4	VS47	VS46	VS45	VS44	VS43	VS42	VS41 VS49	VS40 VS48	VS49 to VS40: Code vertical display potision setting	w
00F89	ORVS5	- VS57	_ VS56	_ VS55			- VS52	VS49 VS51	VS40		
00F8B	011/03	-	-	-	-	-	-	VS59	VS58	VS59 to VS50: Code vertical display potision setting	w
00F8C	ORVS6	VS67	VS66	VS65	VS64	VS63	VS62	VS61	VS60		
00F8D		-	-	_	-	-	_	VS69	VS68	VS69 to VS60: Code vertical display potision setting	w
00F8E	ORVS7	VS77	VS76	VS75	VS74	VS73	VS72	VS71	VS70		
00F8F		-	-	-	-	-	-	VS79	VS78	VS79 to VS70: Code vertical display potision setting	W
00F90	ORVS8	VS87	VS86	VS58	VS84	VS83	VS82	VS81	VS80	VS89 to VS80: Code vertical display potision setting	w
00F91		-	-	-	-	-	-	VS89	VS88		
00F92	ORVS9	VS97	VS96	VS95	VS94	VS93	VS92	VS91	VS90	VS99 to VS90: Code vertical display potision setting	w
00F93		-	-	-	-	-	-	VS99	VS98		
00F94	ORVS10	VS107	VS106	VS105	VS104	VS103	VS102	VS101	VS100	VS100 to VS109:Code vertical display potision setting	w
00F95	00/011	-	-	-	-	-	-	VS109	VS108		
00F96 00F97	ORVS11	VS117	VS116	VS115	VS114	VS113	VS112	VS111 VS119	VS100 VS118	VS110 to VS119:Code vertical display potision setting	w
00F97 00F98	ORVS12	- VS127	- VS126	- VS125		- VS123	- VS122	VS119 VS121	VS118 VS120		
00F99	0111012	-	-	-	-	-	-	VS121	VS120	VS120 to VS129:Code vertical display potision setting	W
00F9A	ORCS4		S4		S3	CS			S1	CSn: Character size (n: 1 to 12)	
00F9B	ORCS8		S8		S7	CS			S5	00: Display off 10: Middle size	w
00F9C	ORCS12	CS	512	CS	S11	CS	10	C	S9	01: Large size 11: Small size	
00F9D	OREULA8	EULA8	EULA7	EULA6	EULA5	EULA4	EULA3	EULA2	EULA1	EULAn: Underline display setting for line n (n: 0 to 12)	w
00F9E	OREULA12	-	-	-	1	EULA12	EULA11	EULA10	EULA9		**
00F9F	OREFR8	EFR8	EFR7	EFR6	EFR5	EFR4	EFR3	EFR2	EFR1	EFRn: Fringing setting for line n (n: 0 to 12)	w
00FA0	OREFR12	-	-	-	-	EFR12	EFR11	EFR10	EFR9		

OSD Control Register List (2/3)	

D	D			-			Ū		、 ,		
Register Address	Register Name	Bit 7	Bit 6	Bit 5	Register bit c Bit 4	-	Bit 2	DH 4	Dit O	Bit contents	R/W
00FA2	ORSOL4	SC			DL3	Bit 3 SC	-	Bit 1	Bit 0 DL1	SOLn: Solid space display setting for line n	
										(n; 0 to 12)	
00FA3	ORSOL8		DL8		DL7	SC			DL5	00: No solid space 10: Right	W
00FA4	ORSOL12	SO			0L11	SO	i		DL9	01: Left 11: Left and right	
00FA5	ORBK	-	RBDT	GBDT	BBDT	-	RFDT	GFDT	BFDT	RBDT, GBDT, BBDT: Background color setting	W
										ACLR2/ACLG2/ACLB2: Area 2 plane color ACLR1/ACLG1/ACLB1: Area 1 plane color	
00FA6	ORACL	PISEL2	ACLR2	ACLG2	ACLB2	PISEL1	ACLR1	ACLG1	ACLB1	PISEL2: Set half transparency for area 2 plane	W
										PISEL1: Set half transparency for area 1 plane	
										EBKGD: Background function	
										EXBL: Full-rasterblanking	
00FA7	ORRCL	EBKGD	EXBL	AON2	AON1	-	RCLR	RCLG	RCLB	AON2: Area 2 plane display	W
										AON1: Area 1 plane display RCLR/RCLG/RCLB:Raster plane color	
00FA8		AHS17	AHS16	AHS15	AHS14	AHS13	AHS12	AHS11	AHS10	AHS1x: Area 1 plane horizonatal start position	
00FA9	ORAHS1	-	-	-	-	-	-	-	AHS18	(x: 0 to 8)	W
00FAA	0.0.4.1.5.4	AHE17	AHE16	AHE15	AHE14	AHE13	AHE12	AHE11	AHE10	AHE1x: Area 1 plane horizonatal end position	
00FAB	ORAHE1	I	I	-	-	-	-	-	AHE18	(x: 0 to 8)	W
00FAC	ORAVS1	AVS17	AVS16	AVS15	AVS14	AVS13	AVS12	AVS11	AVS10	AVS1x: Area 1 plane vertical start position (x: 0 to 8)	w
00FAD	ORAVOI	-	-	-	-	-	-	AVS19	AVS18		**
00FAE	ORAVE1	AVE17	AVE16	AVE15	AVE14	AVE13	AVE12	AVE11	AVE10	AVE1x: Area 1 plane vertical end position (x: 0 to 8)	w
00FAF		-	-	-	-	-	-	AVE19	AVE18		
00FB0	ORAHS2	AHS27	AHS26	AHS25	AHS24	AHS23	AHS22	AHS21	AHS20	AHS2x: Area 2 plane horizonatal start position	w
00FB1 00FB2		AHE27	AHE26	AHE25	– AHE24	AHE23	– AHE22	AHE21	AHS28 AHE20	(x: 0 to 8)	
00FB2	ORAHE2	-	- -	- -	- ANE24	- -	- -	-	AHE20	AHE2x: Area 2 plane horizonatal end position (x: 0 to 8)	w
00FB4		AVS27	AVS26	AVS25	AVS24	AVS23	AVS22	AVS21	AVS20	AVS2x: Area 2 plane vertical start position	
00FB5	ORAVS2	-	-	_	-	-	_	AVS29	AVS28	(x: 0 to 8)	W
00FB6	0041/50	AVE27	AVE26	AVE25	AVE24	AVE23	AVE22	AVE21	AVE20	AVE2x: Area 2 plane vertical end position	
00FB7	ORAVE2	-	-	-	-	-	-	AVE29	AVE28	(x: 0 to 8)	W
										VDSMD: Scan mode select	
										BKMF: Blinking master	
00FB8	ORETC	VDSMD	"0"	BKMF	ESMZ	"0"	MFYWR	MBK	RDWRV	ESMZ: Smoothing MFYWR: Display memory read mode select	W
										MBK: Display memory bank switching select	
										RDWRV: Read/write mode select normal mode	
00FB9	ORIRC	-	-	-	SVD		ISD			SVD: Interrupt source select	w
001 89	ONING	_	_	_	370		151			ISDC: Interrupt generation line select	vv
00FB9	ORIRC	-	-	-	-		DC ⁻	ſR		DCTR:Display line counter	R
										P6xS: P6 port output select (x:4 to 7)	
00FBA	ORP6S	P67S	P66S	P65S	P64S	PIDS	YBLCS	MF	PXS	PIDS: I pin output select YBLCS: Y/BL signal select	W
										MPXS: R, G, B, Y/BL signal select	
										HDPOL: VD input polarity select	
										HDPOL: HD input polarity select	
										YBLII: Y/BLIN input polarity select	
00FBB	ORIV	VDPOL	HDPOL	YBLII	RGBII	YIV	BLIV	RGBIV	IIV	RGBII: RIN, GIN, BIN input polarity select	w
										Y/V: Y Output polarity select BLIV: BL output polarity select	
										RGBIV: R, G, B output polarity select	
										IIV: I pin output polarity select	
00FBC	ORWVSH	WVSH7	WVSH6	WVSH5	WVSH4	WVSH3	WVSH2	WVSH1	WVSH0	WVSHx: Window upper limit position (x: 0 to 9)	w
00FBD	0110000	-	-	-	-	-	-	WVSH9	WVSH8		vv
00FBE	ORWVSL	WVSL7	WVSL6	WVSL5	WVSL4	WVSL3	WVSL2	WVSL1	WVSL0	WVSLx: Window lower limit position (x: 0 to 9)	w
00FBF		-	-	-	-	-	-	WVSL9	WVSL8	, ,	
00FC2	ORCCD	CCD8	CCD7	CCD6	CCD5	CCD4	CCD3	CCD2	CCD1	CCDx: Horizontal 16 dot and vertical 26 dot display at small size character (x: 0 to 12)	w
00FC3		-	-	-	-	CCD12	CCD11	CCD10	CCD9	a Sman Size Glatacler (A. U (U 12)	
00FC4	ORDCSC	DCSC8	DCSC7	DCSC6	DCSC5	DCSC4	DCSC3	DCSC2	DCSC1	DCSCx: Double height display (x: 0 to 12)	w
00FC5		-	-	-	-	DCSC12	DCSC11	DCSC10	DCSC9		

Register	Register			R	egister bit co	onfiguration				Bit contents	R/W
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Dictoments	
00FC6	ORCPT0	CPT0MD1	"0"	CPT0R1	CPT0R0	CPT0G1	CPT0G0	CPT0B1	CPT0B0	Color palette composition register 0 CPT1MD1: OSD color select register (x: 1, 2) CPT1MD1 = 0: 27-color select mode CPT1MD1 = 1: 8-color select mode	w
00FC7	ORCPT1	-	-	CPT1R1	CPT1R0	CPT1G1	CPT1G0	CPT1B1	CPT1B0	Color palette composition register 1	W
00FC8	ORCPT2	-	Ι	CPT2R1	CPT2R0	CPT2G1	CPT2G0	CPT2B1	CPT2B0	Color palette composition register 2	W
00FC9	ORCPT3	-	-	CPT3R1	CPT3R0	CPT3G1	CPT3G0	CPT3B1	CPT3B0	Color palette composition register 3	W
00FCA	ORCPT4	-	-	CPT4R1	CPT4R0	CPT4G1	CPT4G0	CPT4B1	CPT4B0	Color palette composition register 4	W
00FCB	ORCPT5	-	I	CPT5R1	CPT5R0	CPT5G1	CPT5G0	CPT5B1	CPT5B0	Color palette composition register 5	W
00FCC	ORCPT6	-	-	CPT6R1	CPT6R0	CPT6G1	CPT6G0	CPT6B1	CPT6B0	Color palette composition register 6	W
00FCD	ORCPT7	-	-	CPT7R1	CPT7R0	CPT7G1	CPT7G0	CPT7B1	CPT7B0	Color palette composition register 7	W
00FCE	ORSTRG	-	-	-	-	-	-	-	PALTRG	PAL100/NTSC120 start trigger	W

OSD Control Register List (3/3)

Note 1: Except the meshed registers are changed by RGWR.

Note 2: Only lower 2 bits of the register in address 00F80H are changed by RGWR (The register in address 00F80H must not be used with any of the read-modify-write instructions as SET, CLR, etc.).

2.15 Jitter Elimination Circuit

The TMP88CS34/CP34 has a built-in jitter elimination circuit which maintains the vertical stability of the OSD even when input of the vertical signal fluctuates.

And the field decision information for the OSD circuit is detected by using jitter elimination circuit.

2.15.1 Configuration

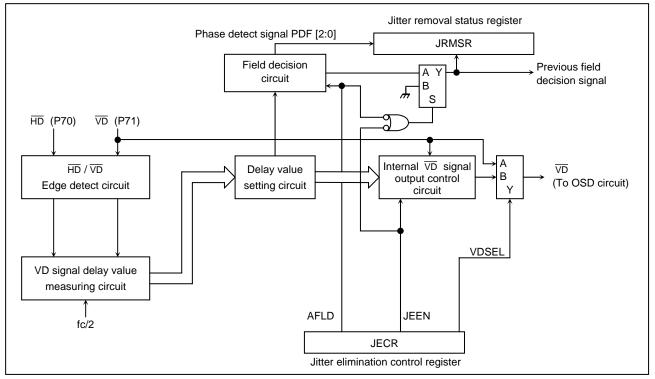


Figure 2.15.1 Jitter Elimination Circuit

2.15.2 Control

Jitter elimination circuit is controlled by the jitter elimination control register (JECR).

littor oliminati	ion control ro	aiotor		
Jitter eliminati		•	1 0	
JECR	7			
(00FE4H)		VDSEL AFLD JEE	N "0" : "0" : (Initial value: ***0 0000)	
	VDSEL	VD select	0: VD from P71	
	VDSEL	VD Select	1: VD from jitter elimination circuit	
	AFLD	Automatic field decision	0: Automatic field decision disabled	Write
			Automatic field decision enabled Jitter elimination disabled	only
	JEEN	Jitter elimination enable specification	1: Jitter elimination disabled	
	Note 1: C	lear the AFLD to "0" to disable jitter elimina	ation circuit.	
	Note 2: A	lways clear "0" to bit 1 and 0 of JECR.		
	Note 3: C	lear "0" to AFLD and VDSEL if there is no	phase shift in the vertical and horizontal sync. signals	everv
		ther time, such as with non-interlaced TV.		,
		: Don't care		
		etting JEEN to "0", OSD display is only 2nd	1 field	
		etting AFLD to "0", OSD display is only 2nd		
	Note 0. 5	etting AFED to 0, OSD display is only zhd		
Jitter eliminati	ion status reg	gister		
JESR	7	6 5 4 3 2	1 0	
(00FE5H)	FDSF	PDF1 PDF0	PDF2 (Initial value: 0*** ****)	
			11	
			0: A position of a scanning line exists in the field	
			which has a second display dot of character on an interlace TV screen.	
	FDSF	Field detect status flag	1: A position of a scanning line exists in the field	
			which has a first display dot of character on an	
			interlace TV screen.	Dood
			000: Phase 0 001: Phase 1	Read only
			010: Phase 2	eniy
		Phase detect flag between $\overline{\text{HD}}$ and $\overline{\text{VD}}$	011: Phase 3	
	FDF2, 1, 0		100: Phase 4	
			101: Phase 5	
			110: Phase 6 111: Phase 7	
			eld. It is a unique field decided for OSD display.	
		: Don't care		
	Note 3: HE	5		
	VE			
		Phase 7 Phase 0 Phase 1 Phase 2 Pha	ase 3 Phase 4 Phase 5 Phase 6 Phase 7 Phase 0	

Figure 2.15.2 Jitter Elimination Control Register and Jitter Elimination Status Register

2.15.3 Jitter Elimination Mode

The jitter elimination circuit is to identify the phase of the falling edges of the external $\overline{\text{VD}}$ signal and $\overline{\text{HD}}$ signal. When $\overline{\text{VD}}$ signal is falling within $\overline{\text{HD}}$ signal falling +/-1/4HD, the jitter is automatically eliminated and internal $\overline{\text{VD}}$ signal is set to the stable location.

This function is enabled by setting JEEN (bit2 in JECR) in the jitter elimination control register to "1".

2.15.4 Auto Field Line Decision

The internal vertical and horizontal sync. signals corrected by the jitter elimination circuit generate the field line decision signals used in the OSD.

The OSD display in normal mode

- Type A) When the OSD circuit is used on the TV system which has a phase shift in the vertical and horizontal sync. Signals every other filed such as the interlace TV, enable jitter elimination circuit and set "1" to AFLD and VDSEL. At this time, the field lines which have first and second display dot of character are displayed.
- Type B) When the OSD circuit is used on the TV system which has no phase shift in the vertical and horizontal sync. Signals every other filed such as the non-interlace TV, enable jitter elimination circuit and clear "0" to AFLD and VDSEL. At this time, the field line which has a second display dot of character is only displayed.

The OSD display in double scan mode

Type C) Disable jitter elimination circuit and clear "0" to AFLD and VDSEL. At this time, the field lines which have first and second display dot of character are displayed.

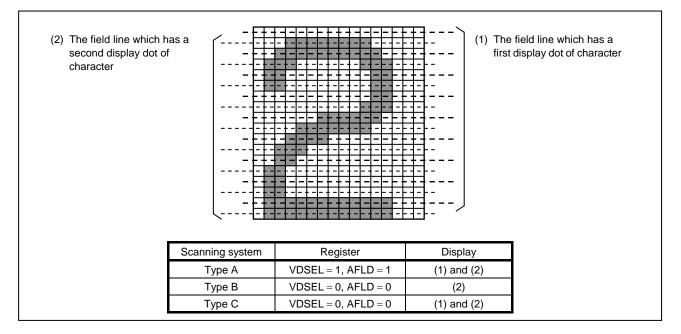


Figure 2.15.3 Relation with Field Line and VDSEL, AFLD

Input/Output Circuit

(1) Control pins

The input/output circuitries of the $\rm TMP88CS34/CP34$ control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remarks
XIN XOUT	I/O	Osc. enable	Resonator connecting pins (high-frequency) $R_{f} = 1.2 \text{ M}\Omega \text{ (typ.)}$ $R_{O} = 0.5 \text{ k}\Omega \text{ (typ.)}$
RESET	I/O	Address-trap-reset Watchdog-timer-reset System-clock-reset	Sink open drain output Hysteresis input Pull-up register $R_{IN} = 220 \text{ M}\Omega \text{ (typ.)}$ $R = 1 \text{ k}\Omega \text{ (typ.)}$
STOP / INT5 (P20)	Input		Hysteresis input R = 1 kΩ (typ.)
TEST	Input		Pull-down register R _{IN} = 70 kΩ (typ.) R = 1 kΩ (typ.)
OSC1 OSC2	I/O	Osc. enable	Pin for connecting a resonator for on-screen display $R_f = 1.2 \ M\Omega \ (typ.)$ $R_O = 0.5 \ k\Omega \ (typ.)$

(2) Input/Output ports

Port	I/O	Input/Output Circuitry	Remarks
		Initial "High-Z"	Sink open drain output Hysteresis input
P20	I/O		R = 1 kΩ (typ.)
P30 to		Initial "High-Z"	Tri-state I/O Hysteresis input
P33 P50, P57 P70, P71	I/O		R = 1 kΩ (typ.)
P34,		Open drain Initial "High-Z" VDD o	Tri-state I/O or Open drain output programmable Hysteresis input
P35, P51, P52	I/O		R = 1 kΩ (typ.)
			Tri-state I/O
P40 to P47	I/O		R = 1 kΩ (typ.)
			Tri-state I/O Hysteresis input Key-on wake-up input
P53			$(V_{IL4} = 0.65 \times V_{DD})$
to P56	I/O		R = 1 kΩ (typ.) R _A = 5 kΩ (typ.)
		C _A Key-on Wake-up	C _A = 22 pF (typ.)

Port	I/O	Input/Output Circuitry	Remarks
P60	I/O	Initial "High-Z"	$ Sink open drain input/output \\ High-current output \\ I_{OL} = 20 mA (typ.) \\ $
P61	I/O	Initial "High-Z" Disable	Tri-state input/output $R = 1 k\Omega (typ.)$ $R_A = 5 k\Omega (typ.)$ $C_A = 22 pF (typ.)$ Key-on wake-up input (V _{IL4} = 0.65 × V _{DD})
P62 to P67	I/O	Initial "High-Z"	Tri-state input/output R = 1 kΩ (typ.)

Electrical Characteristics

Absolute maximum rating	S	(V _{SS} = 0 V)		
Parameter	Symbol	Pins	Ratings	Unit
Supply Voltage	V _{DD}	-	-0.3 to 6.5	
Input Voltage	V _{IN}	_	-0.3 to V _{DD} + 0.3	V
Output Voltage	VOUT1	-	-0.3 to V _{DD} + 0.3	
Output Current (Per 1 pin)	I _{OUT1}	Ports P2, P3, P4, P5, P61 to P67, P7	3.2	
	I _{OUT2}	Ports P60	30	mA
Output Current (Total)	Σ I _{OUT1}	Ports P2, P3, P4, P5, P64 to P67, P7	30	IIIA
Output Outrent (Total)	Σ I _{OUT2}	Ports P60	30	
Power Dissipation [Topr = 70 °C]	PD	_	400	mW
Soldering Temperature (time)	Tsld	-	260 (10 s)	
Storage Temperature	Tstg	-	-55 to 125	°C
Operating Temperature	Topr	-	-30 to 70	

Note: The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

Parameter	Symbol	Pins	Conditions		Min	Max	Unit	
			fc = 16 MHz	NOR	MAL mode			
Supply Voltage	V _{DD}		fc = 16 MHz IDLE mode		mode	4.5	5.5	
				STO	P mode			
	V _{IH1}	Except hysteresis input	V _{DD} = 4.5 to 5.5V		$V_{DD} \times 0.70$			
Input High Voltage	V _{IH2}	Hysteresis input			$V_{DD} \times 0.75$	V _{DD}	V	
	V _{IH3}	Key-on Wake-up input				$V_{\text{DD}} \times 0.90$		
	V _{IL1}	Except hysteresis input	V _{DD} = 4.5 to 5.5V			$V_{DD} \times 0.30$		
Input Low Voltage	V _{IL2}	Hysteresis input			0	$V_{DD} \times 0.25$		
	V _{IL3}	Key-on Wake-up input	V _{DD} = 4.5 to 5.5V			$V_{DD} \times 0.65$		
	fc	XIN, XOUT	V _{DD} = 4.5 to 5	.5V		8.0	16.0	
Clock Frequency	face	Internal clock		fc = 8 MHz	8.0	12.0	MHz	
	fosc	Internal CIUCK	V _{DD} = 4.5 to 5.5V		fc = 16 MHz	16.0	24.0	

Recommended operating conditions (V_{SS} = 0 V, Topr = -30 to 70 °C)

- Note 1: The recommended operating conditions for a device are operating conditions under which it can be guaranteed that the device will operate as specified. If the device is used under operating conditions other than the recommended operating conditions (supply voltage, operating temperature range, specified AC/DC values etc.), malfunction may occur. Thus, when designing products which include this device, ensure that the recommended operating conditions for the device are always adhered to.
- Note 2: Clock frequency fc: Supply voltage range is specified in NORMAL mode and IDLE mode.

Note 3: Smaller value is alternatively specified as the maximum value.

DC Char	acteristics	s (V _{SS} =	0 V, Topr = -30 to 70 °C)				
Parameter	Symbol	Pins	Conditions	Min	Тур.	Max	Unit
Hysteresis voltage	V _{HS}	Hysteresis inputs		-	0.9	-	V
	I _{IN1}	TEST	$V_{DD} = 5.5 \text{ V}, \text{ V}_{IN} = 5.5 \text{ V/0 V}$	-	-	± 2	
Input ourropt	I _{IN2}	Open drain ports	$V_{DD} = 5.5 \text{ V}, \text{ V}_{IN} = 5.5 \text{ V/0 V}$	-	-	± 2	
Input current	I _{IN3}	Tri-state ports	$V_{DD} = 5.5 \text{ V}, \text{ V}_{IN} = 5.5 \text{ V/0 V}$	-	-	± 2	μA
	I _{IN4}	RESET, STOP	$V_{DD} = 5.5 \text{ V}, \text{ V}_{IN} = 5.5 \text{ V/0 V}$	-	-	± 2	
Input resistance	R _{IN2}	RESET	$V_{DD} = 5.5 \text{ V}, \text{ V}_{IN} = 0 \text{ V}$	100	220	450	kΩ
Output leakage	I _{LO1}	Sink open drain ports	$V_{DD} = 5.5 \text{ V}, V_{OUT} = 5.5 \text{ V}$	-	-	2	۵
current	I _{LO2}	Tri-state ports	$V_{DD} = 5.5 \text{ V}, V_{OUT} = 5.5 \text{ V/0 V}$	-	-	± 2	μA
Output high voltage	V _{OH2}	Tri-state ports	$V_{DD} = 4.5 \text{ V}, I_{OH} = -0.7 \text{ mA}$	4.1	-	-	
Output low voltage	put low voltage V _{OL} Except XOUT and ports P60 V _{DD} = 4.		$V_{DD} = 4.5 \text{ V}, \text{ I}_{OL} = 1.6 \text{ mA}$	-	-	0.4	V
Output low current	I _{OL3}	Port P60	$V_{DD} = 4.5 \text{ V}, I_{OL} = 1.0 \text{ V}$	-	20	-	
Supply current in NORMAL mode	AL mode		V _{DD} = 5.5 V fc = 16 MHz (Note3)	_	25	30	mA
Supply current in IDLE mode			fc = 16 MHz (Note3) $V_{IN} = 5.3 V/0.2 V$	_	20	25	
Supply current in STOP mode			$V_{DD} = 5.5 V$ $V_{IN} = 5.3 V/0.2 V$	_	0.5	10	μΑ

Note 1: Typical values show those at Topr = 25 °C, V_{DD} = 5 V.

Note 2: Input Current I_{IN3} ; The current through resistor is not included.

Note 3: Supply Current I_{DD}; The current (Typ. 0.5 mA) through ladder resistors of ADC is included in NORMAL mode and IDLE mode.

AD Conversion Characteristics		(V_{SS} = 0 V, V_{DD} = 4.5 V to 5.5 V, Topr = -30 to 70 °C)					
Parameter Symbol		Conditions	Min	Тур.	Max	Unit	
Analog reference voltage	VAREF	supplied from V _{DD} pin.	-	V _{DD}	-	V	
Allalog Telefence Voltage	V _{ASS}	supplied from V_{SS} pin.	-	0	-		
Analog reference voltage range	ΔV_{AREF}	$= V_{DD} - V_{SS}$	-	V _{DD}	-	v	
Analog input voltage	V _{AIN}		V _{SS}		V _{DD}		
Nonlinearity error		- V _{DD} = 5.0 V	-	Ι	±1		
Zero point error			-	-	±2	LSB	
Full scale error			-	_	±2	LOD	
Total error			-	_	±3		

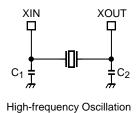
Note: The total error means all error except quanting error.

AC characteristics		(V _{SS} = 0 V, V _{DD} = 4.5 V to 5.5 V, Topr = -30 to 70 °C)					
Parameter	Symbol	Conditions	Min	Тур.	Max	Unit	
Machine cycle time	t _{cy} -	In NORMAL mode	0.5	_	1.0	μS	
		In IDLE mode	0.0			μσ	
High level clock pulse width	tWCH	For external clock operation	31.25			ns	
Low level clock pulse width	tWCL	(XIN input), fc = 16 MHz	51.25		_	115	

Recommended oscillating conditions

 $(V_{SS} = 0 \text{ V}, V_{DD} = 4.5 \text{ V} \text{ to } 5.5 \text{ V}, \text{ Topr} = -30 \text{ to } 70 \text{ }^{\circ}\text{C})$

Parameter	Oscillator	Oscillation	Recommended Oscillator		Recommend	led Constant
T arameter	Oscillator	Frequency			C ₁	C ₂
High-frequency oscillation	Ceramic resonator	8 MHz	Murata	CSA 8.00MTZ	30 pF	30 pF
		16 MHz	Murata	CSA 16.00MXZ040	5 pF	5 pF



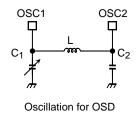
- Note 1: To keep reliable operation, shield the device electrically with the metal plate on its package mold surface against the high electric field, for example, by CRT (Cathode Ray Tube).
- Note 2: The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL;

http://www.murata.co.jp/search/index.html

Recommended oscillating conditions

 $(V_{SS} = 0 \text{ V}, \text{ V}_{DD} = 4.5 \text{ V} \text{ to } 5.5 \text{ V}, \text{ Topr} = -30 \text{ to } 70 \text{ }^{\circ}\text{C})$

ltem	Resonator	Oscillation	Recom	mmended parameter value			
T COM	Resolution	Frequency	L (μΗ)	C ₁ (pF)	C ₂ (pF)		
Oscillation for OSD	LC resonator	8 MHz	33	5 to 30	10		
		12 MHz	15	5 to 30	10		
		16 MHz	10	5 to 30	10		
		20 MHz	6.8	5 to 25	10		
		24 MHz	4.7	5 to 25	10		



The frequency generated in LC oscillation can be obtained using the following equations.

$$f = \frac{1}{2\pi\sqrt{LC}}, C = \frac{C_1 \cdot C_2}{C_1 + C_2}$$

 C_1 is not fixed at a constant value. It can be changed to tune into the desired frequency.

Note 1: Toshiba's OSD circuit determines a horizontal display start position by counting clock pulses generated in LC oscillation. For this reason, the OSD circuit may fail to detect clock pulses normally, resulting in the horizontal start position becoming unstable, at the beginning of oscillation, if the oscillation amplitude is low.

Changing L and C_2 from the values recommended for a specific frequency may hamper a stable OSD display.

If the LC oscillation frequency is the same as a high-frequency clock value, the oscillation of the high-frequency oscillator may cause the LC oscillation frequency to fluctuate, thus making OSD displays flicker.

When determining these parameters, please check the oscillation frequency and the stability of oscillation on your TV sets.

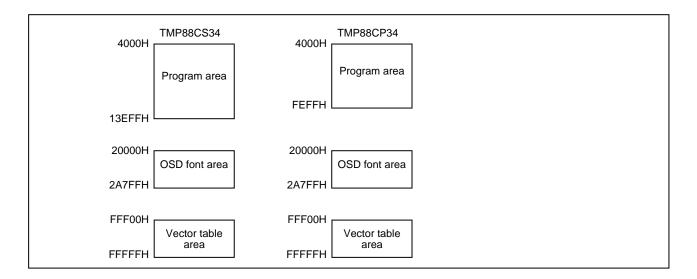
Also check the determined parameters on your final products, because the optimum parameter values may vary from one product to another.

Note 2: When using the LSI package in a strong electric field, such as near a CRT, electrically shield the package so that its normal operation can be maintained.

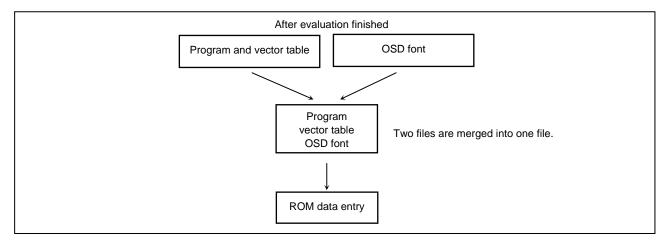
Notice of ROM Entry

When you make a ROM data entry for TMP88CS34/CP34, Please transfer one file including program area, vector table area and OSD font area.

The ROM area must be transferred is as follows.



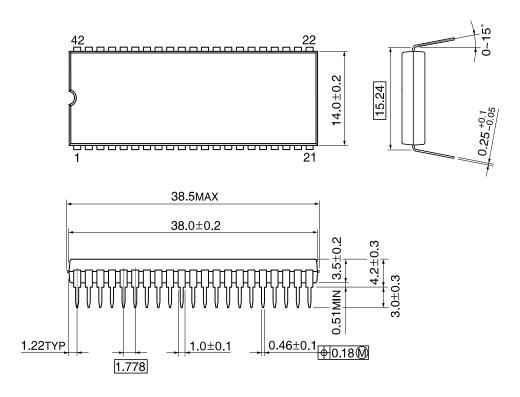
Flow of ROM data entry



Package

P-SDIP42-600-1.78

Unit: mm



P-QFP44-1414-0.80D

Unit: mm

