CMOS 16-Bit Microcontroller
TMP93C071F

1. Outline and Feature

TMP93C071F is a high-speed advanced 16-bit microcontroller developed for application with VCR system control, software servo motor control and timer control.

In addition to basics such as I/O ports, the TMP93C071F has high-speed/high-precision signal measuring circuit, PWM (Pulse-Width-Modulator) and high-precision real timing pulse generator.

The device characteristics are as follows:

(1) Original 16-bit CPU (900L_CPU)
- TLCS-90 instruction mnemonic upward compatible
- 16 Mbyte linear address space
- General-purpose registers and register bank system
- 16-bit multiplication/division and bit transfer/arithmetic instructions
- High-speed micro DMA: 4 channels (1.6 μs / 2 byte at 20 MHz)

(2) Minimum instruction execution time: 200 ns at 20 MHz

(3) Internal ROM: ROMless

(4) Internal RAM: 8 Kbyte

(5) External memory expansion
- Can be expanded up to 16 Mbyte (for both programs and data)
- AM8/16 pin (select the external data bus width)
- Can be mixed 8 and 16bit external data buses.
  ...Dynamic data bus sizing.

(6) 20-bit time-base-counter (TBC)
- free running counter
- accuracy: 100 ns (at 20 MHz)
- overflow: 105 ms (at 20 MHz)

(7) 8-bit timer (TC0): 1 channel
- for CTL linear time counter

(8) 16-bit timer (TC1-5): 5 channels
- C-sync count, capstan FG count, general: (3 channels)

(9) Timing pulse generator (TPG): 2 channels
- (16-bit timing data + 6-bit-output data) with 8-stages FIFO: 1 channel
- (16-bit timing data + 4-bit-output data): 1 channel
- accuracy: 400 ns (at 20 MHz)

(10) Pulse width modulation outputs (PWM)
- 14-bit PWM: 3 channels (for controlling capstan, drum and tuner)
- 8-bit PWM: 9 channels (for controlling volume)
- carrier frequency: 39.1 kHz (at 20 MHz)

(11) 24-bit time base counter capture circuit (Capture 0)
- (18-bit timing data + 6-bit trigger data) with 8-stages FIFO: 1 channel
- capture input sources: Remote-control-input (RMTIN), V-sync, CTL, Drum-PG, general (1 channel)
- accuracy: 400 ns (at 20 MHz)

(12) 17-bit time base counter capture circuit (Capture 1/2)
- (16-bit timing data + 1-bit trigger data): 2 channel
- capture input sources: Drum-FG, Capstan-FG
- accuracy: 100 ns (at 20 MHz)

(13) VISS/VASS detection circuit (VISS/VASS)
- CTL duty detection
- VASS data 16-bit latch

(14) Composite-sync-signal (C-sync) input (C-sync In)
- Vertical-sync-signal (V-sync) separation (V-sepa)

(15) Head Amp switch/Color Rotary control (HA/CR)

(16) Pseudo-V/H generator (PV/PH)

(17) 8-bit A/D converter (ADC): 16 channels
- Conversion speed: 95states (9.5 μs at 20 MHz)

(18) Serial bus I/F
- 8-bit synchronous (SIO0, 1): 2 channels
- UART: 1 channel
- I^2CBUS: 1 channel/2 ports
  - • • • • Multi - Master function/Master transfer with micro DMA.

(19) Watch dog timer (WDT)

(20) Interrupt controller (INTC)
- CPU: 2 sources • • • SWI instruction, and illegal instruction
- Internal: 20 sources ┐ 7-level priority can be set.
- External: 5 sources ┘

(21) I/O ports
- 57 I/O ports (multiplexed functional pins)
- 8 Input ports (P40/AIN3-P47/AIN10: These pins are used as analog input for A/D converter.)
- 4 Output ports (P24/A20-P27/A23: These pins are also used as address bus outputs.)

(22) Standby function: 4 halt modes (RUN, IDLE2, IDLE1, STOP)

(23) System clock function
- Dual clock operation 20 MHz (High-speed: normal)/32 kHz(Low-speed: slow)
  - • • • • 17-bit Real Time Counter built in

(24) Operating Voltage
- Vcc = 2.7 to 5.5 V (at 32 kHz)
- Vcc = 4.5 to 5.5 V (at 20 MHz)

(25) Package
- 120 pin QFP 28 mm × 28 mm (Pin pitch: 0.8 mm)
- Type name QFP120-P-2828-0.80A

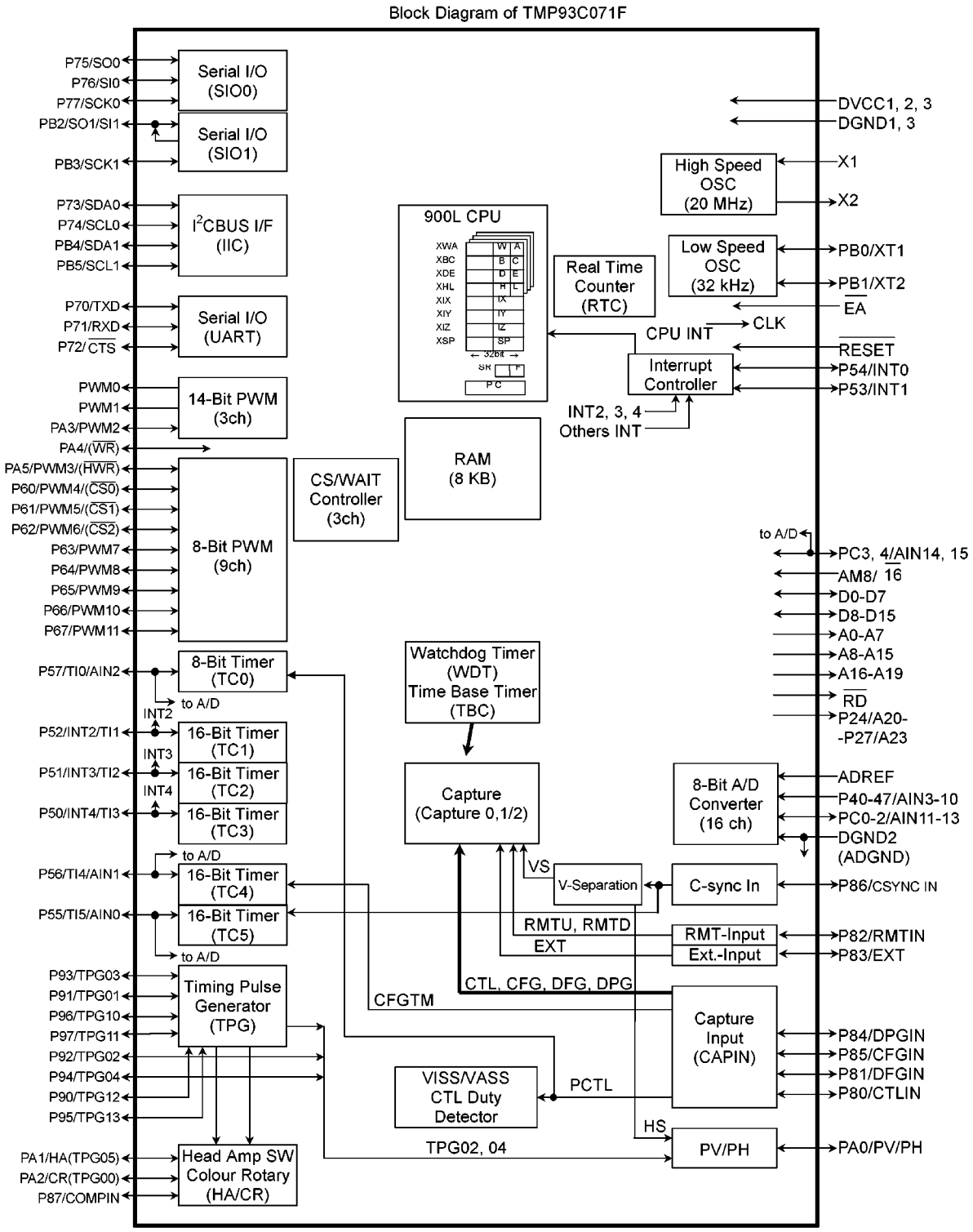


Figure 1 TMP93C071 Block Diagram

2. Pin Assignment And Functions

The assignment of input and output pins for the TMP93C071, their names and functions are described below.

2.1 Pin Assignment

Figure 2.1.1 shows pin assignment of the TMP93C071.

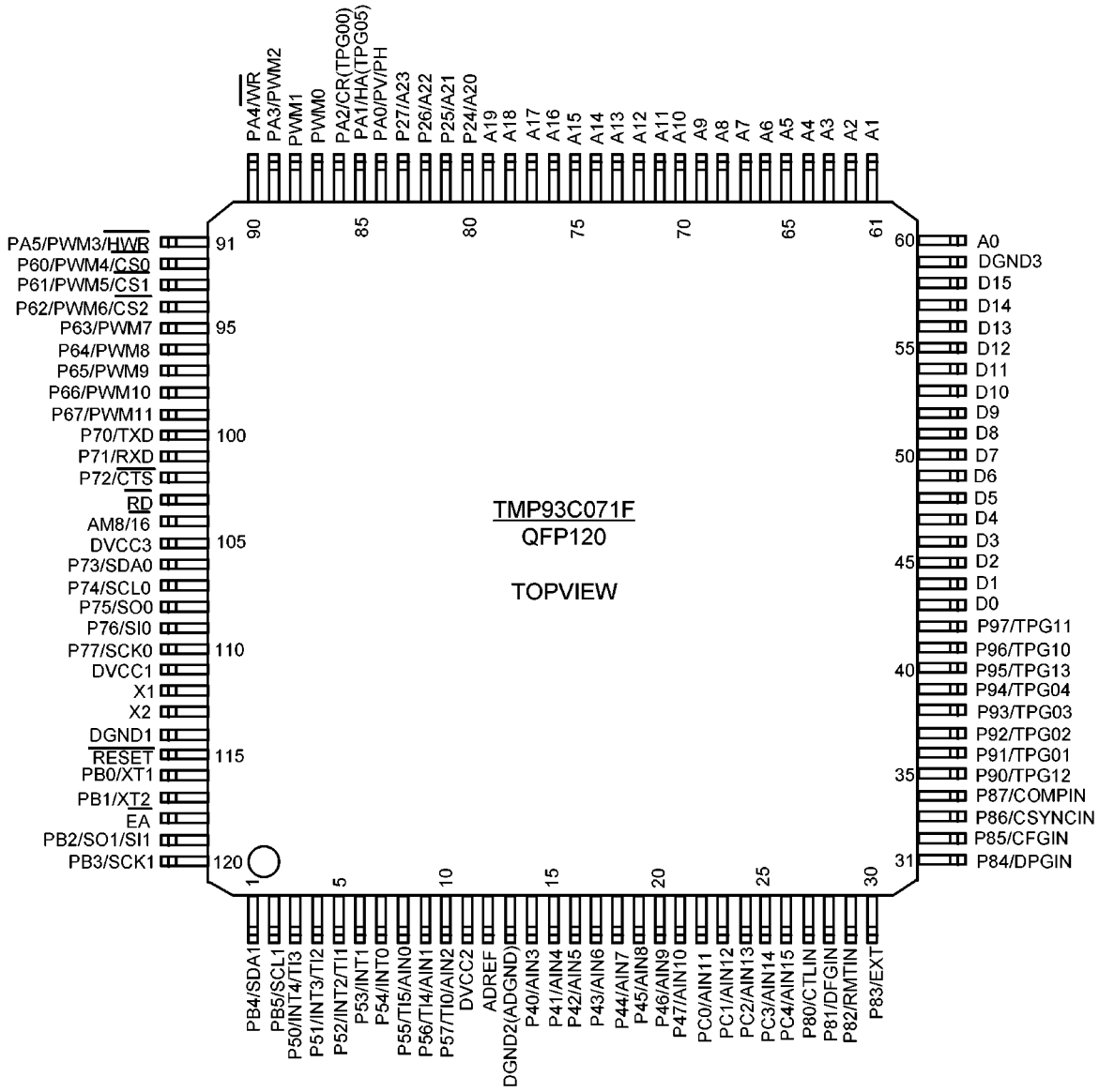


Figure 2.1.1 Pin Assignment (120-pin QFP)

2.2 Pin Names and Functions

The names of input/output pins and their functions are described below.

Table 2.2.1 Pin Names and Function (1/5)

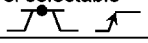
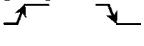
Pin name	Number of pins	I/O	Functions
D0 to D15	16	I/O (3-state)	data: 0 to 15 for data bus
A0 to A19	20	Output	Address: 0 to 19 for address bus
A20 to A23/ P24 to P27	4	Output Output	Address: 20 to 23 for address bus Port 2: Output port
$\overline{RD}$	1	Output	Read: strobe signal for reading external memory
AM8/16	1	Input	data bus width select input (only 8 bit or 8 bit/16 bit)
PC3, 4/ $\overline{16}$ AIN14, 15	2	I/O Input	Port C3, 4: I/O port that allows selection of I/O on a bit basis. Analog Input: Analog input signal for A/D converter
$\overline{EA}$	1	Input	External access: Always set to _0
$\overline{RESET}$	1	Input	Reset: Initializes LSI.(with pull-up R)
X1/X2	2	I/O	High Frequency Oscillator connecting pins (20 MHz)
PB0/ XT1	1	I/O Input	Port B0: I/O port (Open Drain Output) Low Frequency Oscillator connecting pin (32 kHz)
PB1/ XT2	1	I/O Output	Port B1: I/O port (Open drain Output) Low Frequency Oscillator connecting pin
ADREF	1	Input	A/D reference Voltage input
P40 to P47/ AIN3 to AIN10	8	Input Input	Port 4: Input ports Analog input: Analog input signal for A/D converter
PC0 to PC2/ AIN11 to AIN13	3	I/O Input	Port C: PC0 to PC2 I/O port that allows selection of I/O on a bit basis. Analog input: Analog input signal for A/D converter
P57/ TI0/ AIN2	1	I/O Input (schmitt) Input	Port 57: I/O port 8-bit timer0 (TC0) Input 0 Analog input: Analog input signal for A/D converter
P56/ TI4/ AIN1	1	I/O Input (schmitt) Input	Port 56: I/O port 16-bit timer4 (TC4) Input 4 Analog input: Analog input signal for A/D converter
P55/ TI5/ AIN0	1	I/O Input (schmitt) Input	Port 55: I/O port 16-bit timer5 (TC5) Input 5 Analog input: Analog input signal for A/D converter
P54/ INT0	1	I/O Input (schmitt)	Port 54: I/O port External Interrupt request input 0: Rising edge/ Level selectable 
P53/ INT1	1	I/O Input (schmitt)	Port 53: I/O port External Interrupt request input 1: Rising edge/ Level selectable 
P52/ INT2/ TI1	1	I/O Input Input (schmitt)	Port 52: I/O port External Interrupt request input 2 Rising edge/Falling edge selectable  16-bit timer1(TC1) Input 1

Table 2.2.1 Pin Names and Function (2/5)

Pin name	Number of pins	I/O	Functions
P51/ INT3/ TI2	1	I/O Input Input (schmitt)	Port 51: I/O port External Interrupt request input 3 Rising edge/Falling edge selectable 16-bit timer2 (TC2) Input 2
P50/ INT4/ TI3	1	I/O Input Input (schmitt)	Port 50: I/O port External Interrupt request input 4 Rising edge/Falling edge selectable 16-bit timer3 (TC3) Input 3
PWM0	1	Output 3-state Open Drain	PWM (14 bit) output 0: PWM0 output push/pull or open drain output selectable
PWM1	1	Output 3-state Open Drain	PWM (14 bit) output 1: PWM1 output push/pull or open drain output selectable
PA3/ PWM2	1	I/O 3-state Open Drain	Port A3: I/O port PWM (14 bit) output 2: PWM2 output push/pull or open drain output selectable
PA4/ $\overline{WR}$	1	I/O 3-state Open Drain Output	Port A4: I/O port push/pull or open drain output selectable Write: Strobe signal for writing data on pins D0 to D7
PA5/ PWM3/ $\overline{HWR}$	1	I/O Output 3-state Open Drain Output	Port A5: I/O port 8-bit PWM output 3: PWM3 output push/pull or open drain output selectable High write: Strobe signal for writing data on pins D8 to D15
P60/ PWM4/ $\overline{CS0}$	1	I/O Output 3-state Open Drain Output	Port 60: I/O port 8-bit PWM output 4: PWM4 output push/pull or open drain output selectable Chip select0: Output _0_ when address is within specified address area.
P61/ PWM5/ $\overline{CS1}$	1	I/O Output 3-state Open Drain Output	Port 61: I/O port 8-bit PWM output 5: PWM5 output push/pull or open drain output selectable Chip select1: Output _0_ when address is within specified address area.
P62/ PWM6/ $\overline{CS2}$	1	I/O Output 3-state Open Drain Output	Port 62: I/O port 8-bit PWM output 6: PWM6 output push/pull or open drain output selectable Chip select2: Output _0_ when address is within specified address area.

Table 2.2.1 Pin Names and Function (3/5)

Pin name	Number of pins	I/O	Functions
P63/ PWM7	1	I/O Output 3-state Open Drain	Port 63: I/O port 8-bit PWM output7: PWM7 output push/pull or open drain output selectable
P64/ PWM8	1	I/O Output 3-state Open Drain	Port 64: I/O port 8-bit PWM output8: PWM8 output push/pull or open drain output selectable
P65/ PWM9	1	I/O Output 3-state Open Drain	Port 65: I/O port 8-bit PWM output9: PWM9 output push/pull or open drain output selectable
P66/ PWM10	1	I/O Output 3-state Open Drain	Port 66: I/O port 8-bit PWM output 10: PWM10 output push/pull or open drain output selectable
P67/ PWM11	1	I/O Output 3-state Open Drain	Port 67: I/O port 8-bit PWM output 11: PWM11 output push/pull or open drain output selectable
P73/ SDA0	1	I/O I/O (schmitt) Open Drain	Port 73: I/O port I^2CBUS SDA line 0 push/pull or open drain output selectable
P74/ SCL0	1	I/O I/O (schmitt) Open Drain	Port 74: I/O port I^2CBUS SCL line 0 push/pull or open drain output selectable
P75/ SO0	1	I/O Output (schmitt) Open Drain	Port 75: I/O port SIO0 send data 0 push/pull or open drain output selectable
P76/ SI0	1	I/O Input (schmitt)	Port 76: I/O port SIO0 receive data 0
P77/ SCK0	1	I/O I/O (schmitt) Open Drain	Port 77: I/O port SIO0 transfer clock input/output 0 push/pull or open drain output selectable
P70/ TXD	1	I/O Output (schmitt) Open Drain	Port 70: I/O port UART send data push/pull or open drain output selectable
P71/ RXD	1	I/O Input (schmitt)	Port 71: I/O port UART receive data
P72/ $\overline{CTS}$	1	I/O Input (schmitt)	Port 72: I/O port UART clear to send
P80/ CTLIN	1	I/O Input (schmitt)	Port 80: I/O port Capture input for Control signal (CTL)

Table 2.2.1 Pin Names and Function (4/5)

Pin Name	Number of pins	I/O	Functions
P81/ DFGIN	1	I/O Input (schmitt)	Port 81: I/O port Capture input for Drum-FG signal (DFG)
P82/ RMTIN	1	I/O Input (schmitt)	Port 82: I/O port Capture input for Remote Control Input signal
P83/ EXT	1	I/O Input (schmitt)	Port 83: I/O port External Capture input (Rising edge only)
P84/ DPGIN	1	I/O Input (schmitt)	Port 84: I/O port Capture input for Drum-PG signal (DPG)
P85/ CFGIN	1	I/O Input (schmitt)	Port 85: I/O port Capture input for Capstan-FG signal (CFG)
P86/ CSYNC IN	1	I/O Input (schmitt)	Port 86: I/O port Capture input for C-sync
P87/ COMPIN	1	I/O Input (schmitt)	Port 87: I/O port Envelope Comparator Input (to HA/CR)
P90/ TPG12	1	I/O Output Open Drain	Port 90: I/O port TPG12: TPG output 12 push/pull or open drain output selectable
P91/ TPG01	1	I/O Output Open Drain	Port 91: I/O port TPG01: TPG output 01 push/pull or open drain output selectable
P92/ TPG02	1	I/O Output Open Drain	Port 92: I/O port TPG02: TPG output 02 (Internally connected to PV/PH Logic) push/pull or open drain output selectable
P93/ TPG03	1	I/O Output Open Drain	Port 93: I/O port TPG03: TPG output 03 push/pull or open drain output selectable
P94/ TPG04	1	I/O Output Open Drain	Port 93: I/O port TPG04: TPG output 04 (Internally connected to PV/PH Logic) push/pull or open drain output selectable
P95/ TPG13	1	I/O Output Open Drain	Port 95: I/O port TPG13: TPG output 13 push/pull or open drain output selectable
P96/ TPG10	1	I/O Output Open Drain	Port 96: I/O port TPG10: TPG output 10 push/pull or open drain output selectable
P97/ TPG11	1	I/O Output Open Drain	Port 97: I/O port TPG11: TPG output 11 push/pull or open drain output selectable
PA0/ PV-PH	1	I/O Output 3-state	Port PA0: I/O Port Pseudo-Vsync/Pseudo-Hsync (PV/PH) output (controlled by TPG02/04.)
PA1/ HA (TPG05)	1	I/O Output	Port PA1: I/O Port HA: Head amp switch output (are also used as TPG05 output.)
PA2/ CR (TPG00)	1	I/O Output	Port PA2: I/O Port CR: Colour Rotary output (are also used as TPG00 output.)

Table 2.2.1 Pin Names and Function (5/5)

Pin name	Number of pins	I/O	Functions
PB2/ SO1/SI1	1	I/O I/O (schmitt) Open Drain	Port PB2: I/O Port SIO1 send data 1 and receive data 1 (Internally connected) push/pull or open drain output selectable
PB3/ SCK1	1	I/O I/O (schmitt) Open Drain	Port PB3: I/O Port SIO1 transfer clock input/output 1 push/pull or open drain output selectable
PB4/ SDA1	1	I/O I/O (schmitt) Open Drain	Port PB4: I/O Port I^2CBUS SDA line 1 push/pull or open drain output selectable
PB5/ SCL1	1	I/O I/O (schmitt) Open Drain	Port PB5: I/O Port I^2CBUS SCL line 1 push/pull or open drain output selectable
DVCC1, 2, 3	3		Power supply pins All of these pins should be connected to power source.
DGND1, DGND2 (ADGND), DGND3	3		GND pins (0 V) All of these pins should be connected to GND (0 V) line. DGND2 are also used as ADGND for A/D converter.

3. Operation

This section describes the functions and basic operational blocks of TMP93C071 devices.

See the "7. Points of Concern and Restrictions" for the using notice and restrictions for each block.

3.1 CPU

TMP93C071 devices have a built-in high-performance 16-bit CPU (900 / L CPU). (For CPU operation, see TLCS-900 / L CPU in the previous section).

This section describes CPU functions unique to the TMP93C071 that are not described in the previous section.

3.1.1 Reset

To reset the TMP93C071, the RESET input must be kept at 0 for at least 10 system clocks. (1 μs at 20 MHz) within the operating voltage range and with a stable oscillation.

When reset is accepted, the CPU sets as follows:

- Program Counter (PC) according to Reset Vector that is stored FFFF00H to FFFF02H.
  - PC (7 to 0) ← stored data in location FFFF00H
  - PC (15 to 8) ← stored data in location FFFF01H
  - PC (23 to 16) ← stored data in location FFFF02H
- Stack pointer (XSP) for system mode to 100H.
- IFF2 to 0 bits of status register to 111. (Sets mask register to interrupt level 7.)
- MAX bit of status register to 1. (Sets to maximum mode)
- Bits RFP2 to 0 of status register to 000. (Sets register banks to 0.)

When reset is released, instruction execution starts from PC (reset vector). CPU internal registers other than the above are not changed.

When reset is accepted, processing for built-in I/Os, ports, and other pins is as follows

- Initializes built-in I/O registers as per specifications.
- Sets port pins (including pins also used as built-in I/Os) to general-purpose input / output port mode.

Note: By resetting, register in the CPU except program counter (PC), status register (SR) and stack pointer (XSP) and the data in internal RAM are not changed.

Figure 3.1.1 show the reset timing chart of TMP93C071.

3.1.2 AM8/$\overline{16}$ pin

① With fixed 16-bit data bus, external 16-bit data bus or 8-bit data bus is selectable

Set this pin to "L".

The external data bus width is set by the chip select/wait control register which is described in section 3.6.3.

It is necessary to set the program memory to be accessed to 16-bit data bus after reset.

② With fixed external 8-bit data bus

Set this pin to "H".

The values of bit 4 <B0BUS>, <B1BUS> and <B2BUS> in the chip select/wait control register described in section 3.6.3 are invalid. The external 8-bit data bus is fixed.

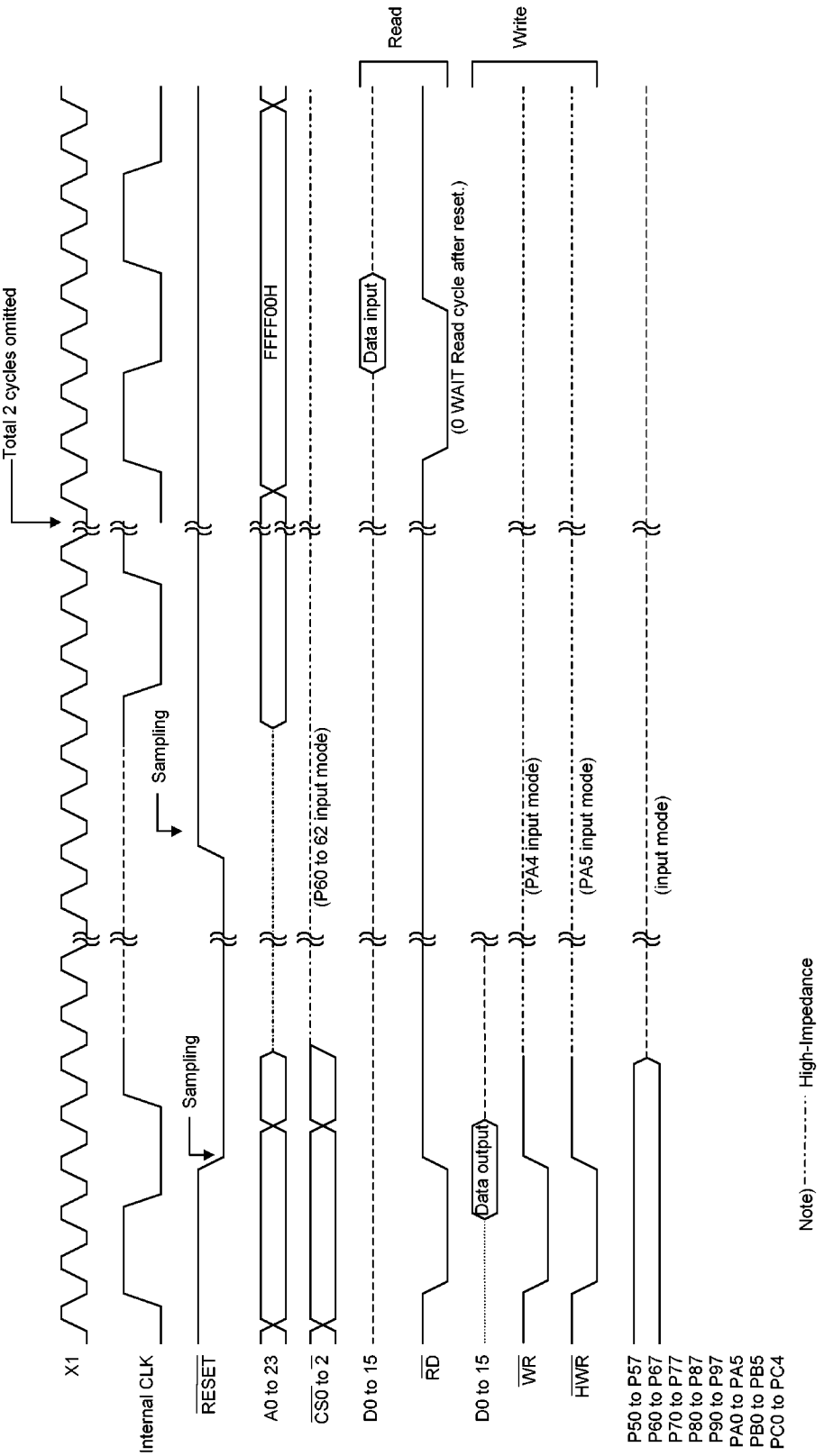


Figure 3.1.1 TMP93C071 Reset Timing Chat

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP93C071.

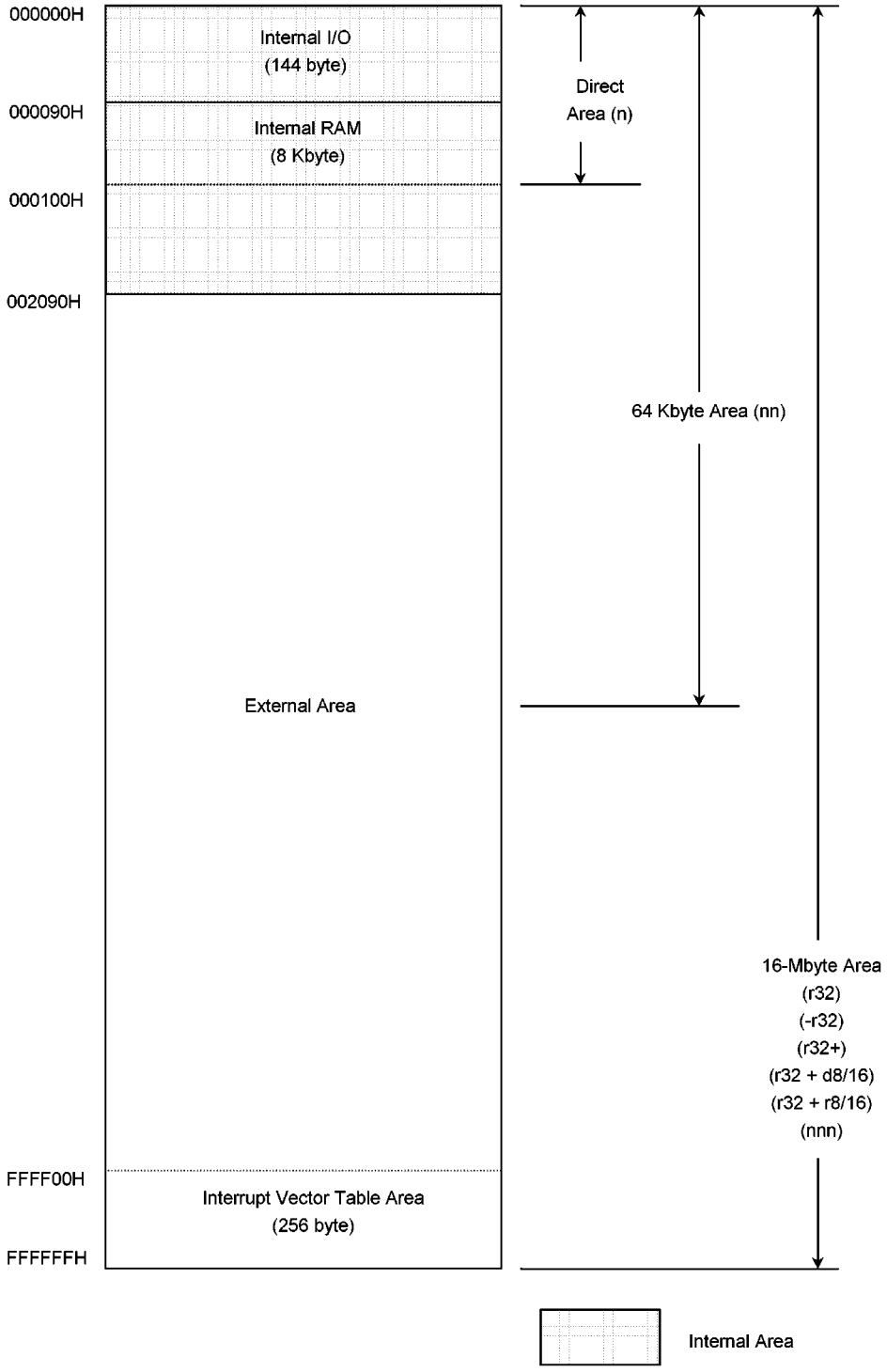


Figure 3.2.1 Memory map

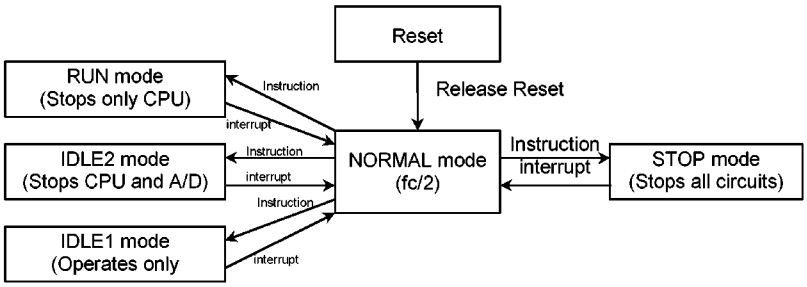
3.3 Dual Clock, Standby Function

Dual Clock, Stand by Control Circuits consist of (1) System clock Controller, 2 Timing clock Generator for I/O Block, 3 Real Time Clock Generator and 4 Standby Controller.

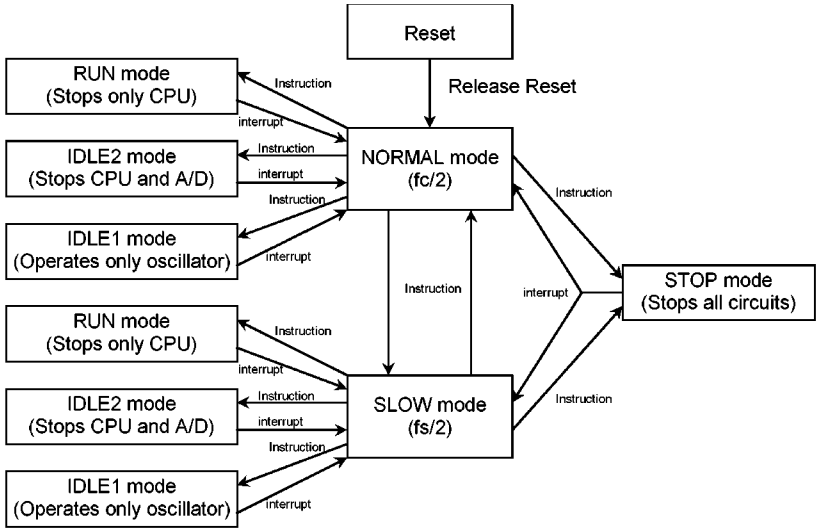
The Oscillator operating mode is classified to (a) Single Clock mode (only X1, X2 pin), and (b) Dual Clock mode (X1, X2, XT1, XT2 pin).

Figure 3.3.1 shows a transition figure. Figure 3.3.2 shows the block diagram.

Figure 3.3.3 shows I/O registers. Table 3.3.1 shows the internal operation and system clock.



(a) Single Clock mode transition figure



(b) Dual Clock mode transition figure

Figure 3.3 1 Transition Figure

The Clock Frequency input from X1, X2 pin is called fc, and the Clock Frequency input from XT1, XT2 pin is called fs. The clock frequency selected by SYSCR1<SYSCK> is called system clock f_{FPH}. The devided clock of f_{FPH} is called system clock f_{SYS}, and the 1 cycle of f_{SYS} is called 1 state.

Table 3.3.1 Internal operation and system clock

Operating Mode		Oscillator		CPU	internal I/O	System clock f_{SYS}
		High Frequency (fc)	Low Frequency (fs)			
Single Clock	RESET	oscillation	stop	reset	reset	$fc/_2$
	NORMAL			operate	operate	
	RUN			stop		
	IDLE2				stop only A/D	
	IDLE1				stop	
	STOP	stop				stop
Dual Clock	RESET	oscillation	stop	reset	reset	$fc/_2$
	NORMAL		progamable	operate	operate	
	SLOW	programable	oscillation			$fs/_2$
	RUN	Oscillator being used as system clock: oscillation Other oscillator: programmable		stop		programable ($fc/_2$, $fs/_2$)
	IDLE2				stop only A/D	
	IDLE1				stop	
	STOP	stop				stop

The TMP93C071 has not a clock gear circuit.

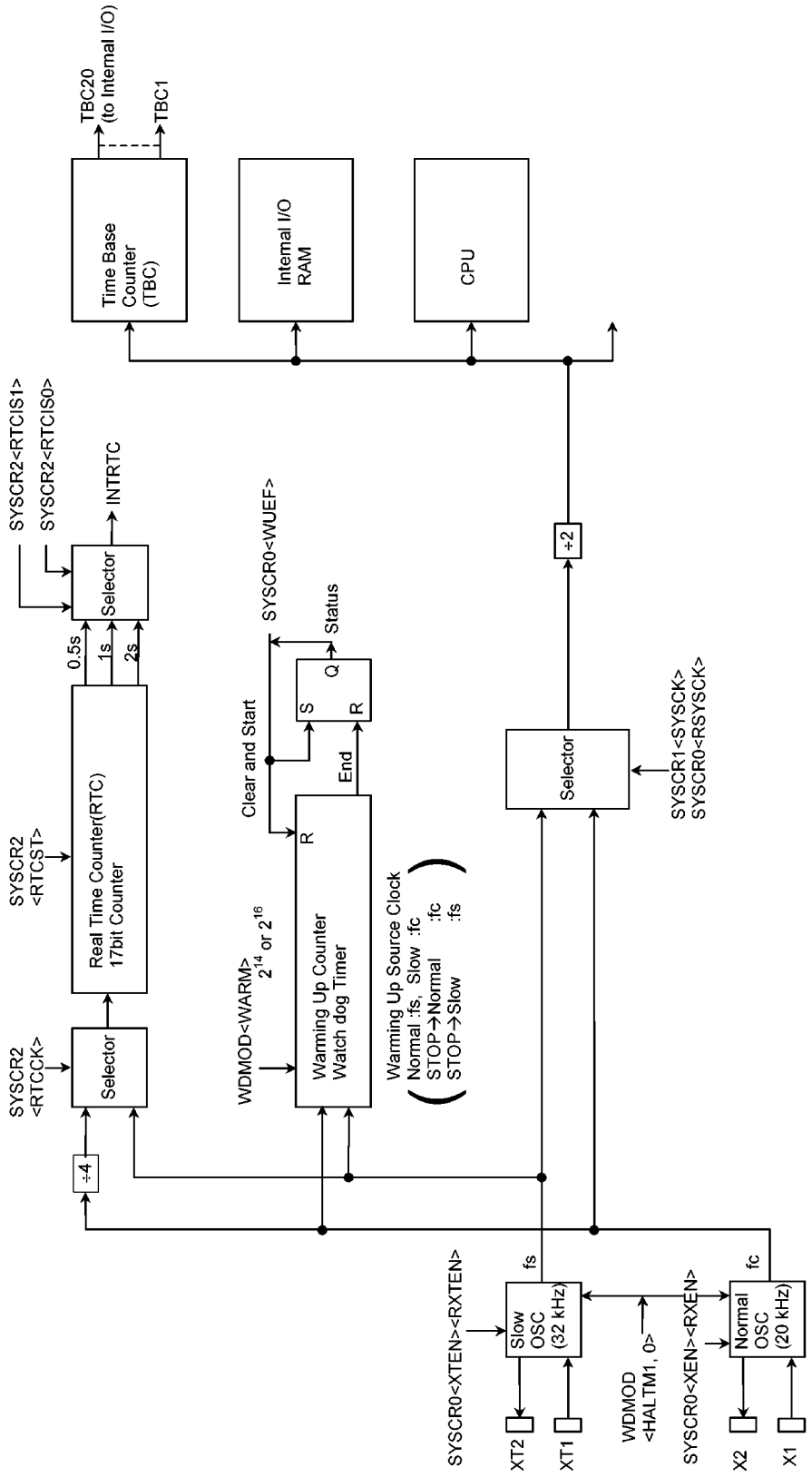


Figure 3.3.2 Block Diagram of Dual Clock, Standby circuits

SYSCR0 (006EH)

	7	6	5	4	3	2	1	0
bit Symbol	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF		
Read/Write	R/W							
After reset	1	0	1	0	0	0		
Function	High Frequency oscillator (fc) 0: stop 1: oscillation	Low Frequency oscillator (fs) 0: stop 1: oscillation	High Frequency oscillator (fc) after released STOP mode 0: stop 1: oscillation	Low Frequency oscillator (fc) after released STOP mode 0: stop 1: oscillation	select clock after released STOP mode 0: fc 1: fs	Warm-Up Timer 0 write: don't care 1 write: start timer 0 read: end warm-up 1 read: not end warm-up		

SYSCR1 (006FH)

	7	6	5	4	3	2	1	0
bit Symbol					SYSCK			
Read/Write	(R/W)				R/W	(R/W)		
After reset	(0)				0	(0)	(0)	(0)
Function	Always set to "0"				select system clock 0: fc 1: fs	Always set to "0"		

SYSCR2 (006CH)

	7	6	5	4	3	2	1	0
bit Symbol				RTCCK	RTCST	RTCIS1	RTCIS0	
Read/Write	(R/W)	(R/W)		R/W	R/W	R/W		
After reset	(0)	(0)		0	0	0		
Function				RTC clock source select 0: fs (32 kHz) 1: fc (20 MHz)	RTC count 0: Stop & Counter Clear 1: Start	Interval time control of RTC interrupt 00: $f_{SYS}/2^{17}$ or fs/2^{15} (6.55 ms) (1 s) 01: $f_{SYS}/2^{18}$ or fs/2^{16} (131 ms) (2 s) 10: $f_{SYS}/2^{16}$ or fs/2^{14} (328 ms) (0.8 s) 11: Reserved [Hz]		

SYSCR3 (006DH)

	7	6	5	4	3	2	1	0
bit Symbol								CLKEN
Read/Write	(set to "1")	(set to "1")	(set to "1")	(set to "1")	(set to "0")	(set to "0")		R/W
After reset	1	1	1	1	0	0		0
Function								0: Disable 1: Enable

WDMOD (005CH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	WDT control 0: disable 1: enable	WDT Detection Time 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Warm-Up Timer 0: 2^{14}/ frequency inputted 1: 2^{16}/ frequency inputted	HALTmode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		0: Don't care 1: Connects WDT output to RESET pin internally	Pin status controlling STOP mode 0: I/O off 1: Remains the state before HALT

Note1: SYSCR1 <bit 6-4> and SYSCR0 <bit 1-0> area read as "1".

Note2: Writing "0" to SYSCR1 <SYSCK> enables the high-frequency oscillator regardless of the value of SYSCR0 <XEN>. Additionally, writing "1" to <SYSCK> register enables the low-frequency oscillator regardless of the value of SYSCR0 <XTEN>.

Note3: Set "1" to SYSCR3<CLKEN> for stabilizing the operation current in slow mode and stop mode.

Figure 3.3.3 I/O registers about Dual Clock, Standby

3.3.1 System Clock Controller

The system clock controller generates system clock (fSYS) for CPU core and internal I/O. It contains two oscillation circuits. The register SYSCR1<SYSCK> changes system clock to either fc or fs, SYSCR0<XEN>, <XTEN> controls enable / disable each oscillator.

The system clock (fSYS) is set to fc/2 because of <XEN>="1", <XTEN>="0", <SYSCK>="0" by resetting.

For example, fSYS is set to 10 MHz by resetting the case of 20 MHz oscillator is connected to X1, X2 pins.

The high frequency (fc) and low frequency (fs) clocks can be easily obtained by connecting a resonator to the X1 / X2, XT1 / XT2 pins, respectively. Clock input from an external oscillator is also possible.

The XT1, XT2 pins have also Port PB0, PB1 function. Therefore the case of single clock mode, the XT1, XT2 pins can be used as I/O port pins.

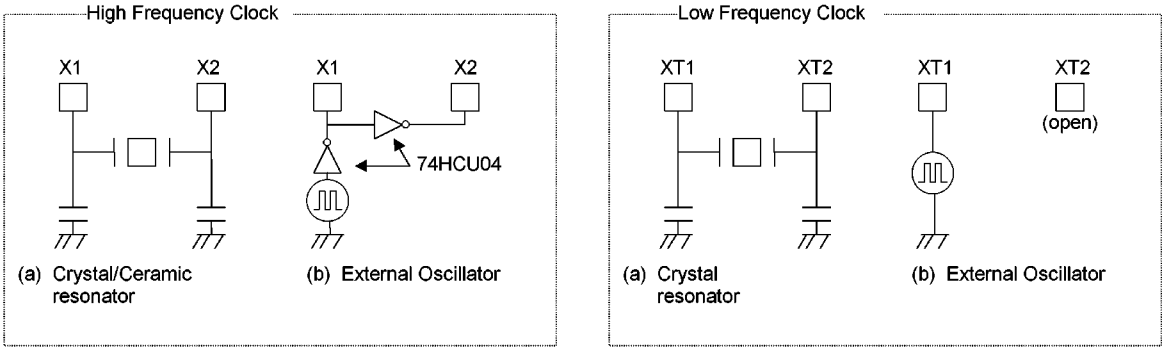


Figure 3.3.4 Examples of Resonator Connection

Note: Note on using the low frequency oscillation circuit.

In connecting the low frequency resonator to ports PB0 and PB1, it is necessary to make the following settings to reduce the power consumption.

(connecting with resonators) PBCR <PB0C, PB1C> = "11", PB <PB0, PB1> = "00"

(connecting with oscillators) PBCR <PB0C, PB1C> = "11", PB <PB0, PB1> = "10"

(1) Switching from NORMAL to SLOW mode

When the resonator is connected to X1, X2, or XT1, XT2 pin, the warm-up timer is used to change the operation frequency after getting stabilized oscillation.

The warm-up time can be selected by WDMOD<WARM>.

This starting and ending of warm-up timer are performed like the following example 1, 2 by program.

Note 1: The warm-up timer is also used as a watchdog timer. So, when it is used as a warm-up timer, the watchdog timer must be disabled.

Note 2: The case of using oscillator (not resonator) with stabilized oscillation, a warm-up timer is not need.

Note 3: The warm-up timer is operated by a oscillation clock. Therefore, warm-up time has an error.

Table 3.3.2 Warm-up Time

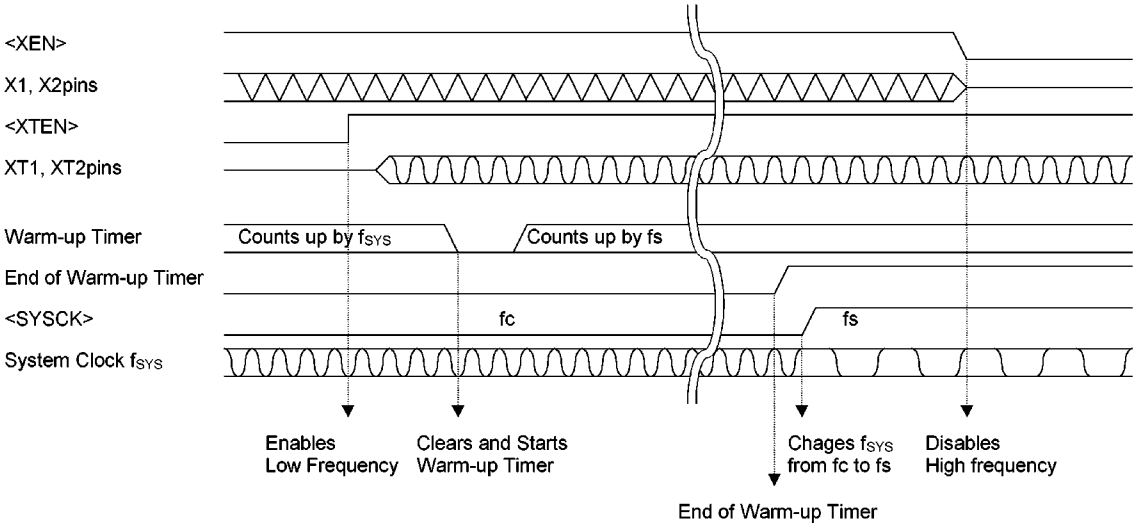
Warm-up Time WDMOD<WARM>	Change to NORMAL	Change to SLOW
0 (2^{14}/frequency)	0.8192 (ms)	500 (ms)
1 (2^{16}/frequency)	3.2768 (ms)	2000 (ms)

at fc = 20 MHz,
fs= 32.768 kHz

Clock Setting Example 1:

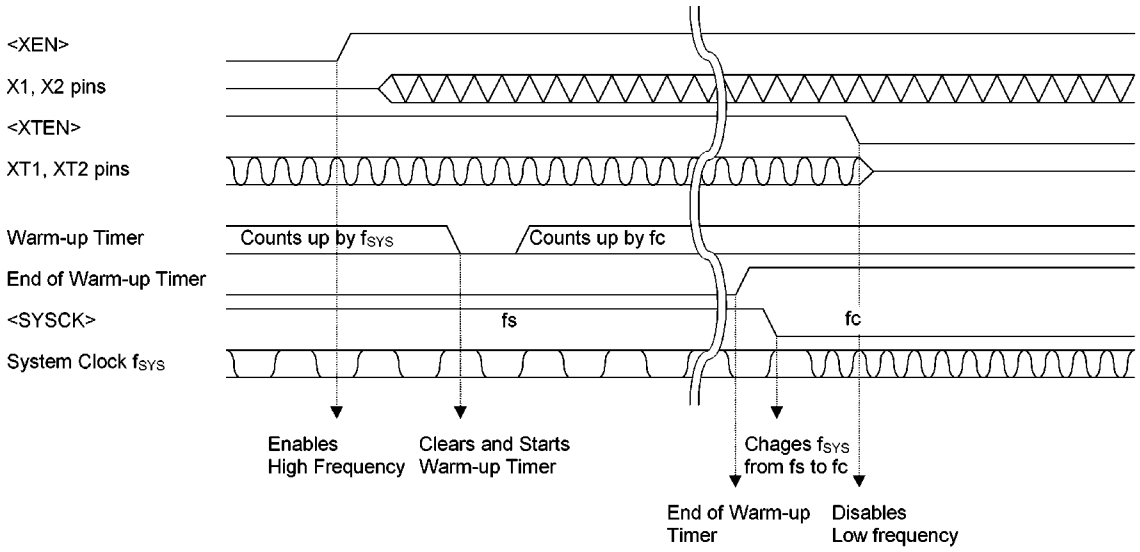
Changing from the high frequency (fc) to the low frequency (fs).

```
SYSCR0      EQU     006EH
SYSCR1      EQU     006FH
WDCR        EQU     005DH
WDMOD       EQU     005CH
            RES     7, (WDMOD);   } Disables Watchdog Timer.
            LD      (WDCR),B1H;   }
            SET     4, (WDMOD); Sets Warm-up Time to 2^16/fs.
            SET     6, (SYSCR0); Enables Low Frequency (fs).
            SET     2, (SYSCR0); Clears and stars Warm-up Timer.
WUP:        BIT     2, (SYSCR0);  } Detects End of Warm-up Timer.
            JR      NZ,WUP;       }
            SET     3, (SYSCR1); Changes fSYS from Frequency Oscillation.
            RES     7, (SYSCR0); Disables High Frequency Oscillation.
            SET     7, (WDMOD); Enables Watchdog Timer.
```



Clock Setting Example 2:
changing from the low frequency (fs) to the high frequency (fc).

```
SYSCR0    EQU   006EH
SYSCR1    EQU   006FH
WDCR      EQU   005DH
WDMOD     EQU   005CH
          RES   7, (WDMOD);   }
          LD    (WDCR),B1H;   } Disables Watchdog Timer.
          RES   4, (WDMOD); Sets Warm-up Time to 2^14/fc.
          SET   7, (SYSCR0); Enables High Frequency (fc).
          SET   2, (SYSCR0); Clears and stars Warm-up Timer.
WUP:      BIT   2, (SYSCR0);  }
          JR    NZ, WUP;      } Detects End of Warm-up Timer.
          RES   3, (SYSCR1); Changes fSYS from fs to fc.
          RES   6, (SYSCR0); Disables Low Frequency Oscillation.
          SET   7, (WDMOD); Enables Watchdog Timer.
```



3.3.2 Timing Clock Generator

The timing clock generator generates sorts of system clock from the basic clock (fc or fs), providing for CPU core and peripheral hardwares.

(1) Architecture

The timing clock generator consists of the system clock generator and the time base counter (TBC) which generates system clock for peripheral hardwares. After resetting, the system clock is generated from high frequency clock (fc) (NORMAL mode). Both Executing the instruction and operating the internal hardwares are synchronized by this system clock.

(2) Time Base Counter

The time base counter consists of a 20-bit up-counter counted by a basic clock divided-by 2 (fc/2 or fs/2), 16-bit data register and control register.

Figure 3.3.5 Shows the structure of the time-base counter (TBC).

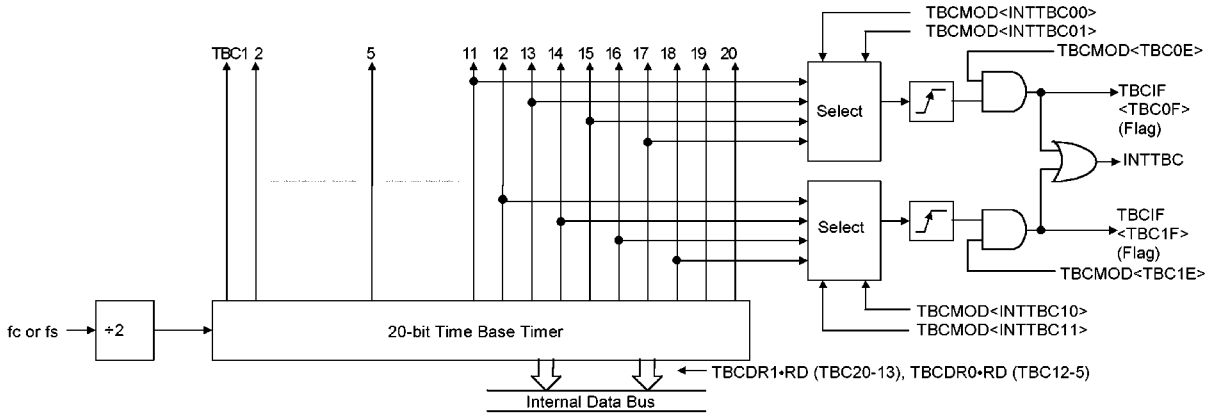


Figure 3.3.5 Shows the structure of the time-base counter (TBC).

Time Base Counter.Control Register

TBCMOD (0023H)

7	6	5	4	3	2	1	0
		TBC1E	TBC0E	INT TBC11	INT TBC10	INT TBC01	INT TBC00

(Reset Value **00 0000)

TBC1E TBC0E	INTTBC Interrupt Enable / Disable	00: INTTBC Interrupt Disable 01: INTTBC0 Interrupt Enable 10: INTTBC1 Interrupt Enable 11: INTTBC0 / INTTBC1 Interrupt Enable	R/W
INTTBC11 INTTBC10	INTTBC1 Interrupt Source Clock Selection	00: TBC12 01: TBC14 10: TBC16 11: TBC18	
INTTBC01 INTTBC00	INTTBC0 Interrupt Source Clock Selection	00: TBC11 01: TBC13 10: TBC15 11: TBC17	

Time Base Counter Data register 0

TBCDR0 (0024H)

7	6	5	4	3	2	1	0
TBC12	TBC11	TBC10	TBC9	TBC8	TBC7	TBC6	TBC5

(Reset Value 0000 0000) Read-only

Time Base Counter Data Register 1

TBCDR1 (0025H)

7	6	5	4	3	2	1	0
TBC20	TBC19	TBC18	TBC17	TBC16	TBC15	TBC14	TBC13

(Reset Value 0000 0000) Read-only

Time Base Counter Interrupt Request Flag Register

TBCIF (0026H)

7	6	5	4	3	2	1	0
		TBC1F	TBC0F				

TBC1E	INTTBC1 Interrupt Request Flag	0 (W): Clear 1 (W): (Inhibit) 0 (R): No interrupt request 1 (R): Interrupt request	R/W
TBC0F	INTTBC0 Interrupt Request Flag	0 (W): Clear 1 (W): (Inhibit) 0 (R): No interrupt request 1 (R): Interrupt request	

① Operation

The time-base counter outputs (TBC1 to TBC20) are used as clock source or timing data for Timer/Counter, Capture (CAP0/CAP1/CAP2), timing pulse generator (TPG), and other peripheral I/O blocks. The contents of time-base counter outputs TBC5 to TBC20 can be read by reading the time-base counter data registers, TBCDR0 and TBCDR1. Note that the data registers must be read in order of TBCDR0 and then TBCDR1.

Time-base counter interrupt requests (INTTBC) can be generated on the rising edges of counter outputs TBC11 to TBC18. The interrupt source is selected by the time-base counter control register TBCMOD <INTTBC11, INTTBC10, INTTBC01, and INTTBC00>. The INTTBC interrupt requests are comprised of two interrupt request signals INTTBC0 and INTTBC1 that are logical OR' ed to generate an interrupt request. Which interrupt is requested can be identified by reading the time base counter interrupt request flag register TBCIF <TBC0F> and <TBC1F>.

The INTTBC0 flag <TBC0F> and INTTBC1 flag <TBC1F> can be cleared by writing "0" in the register.

Table 3.3.3 lists the interval time of time-base counter outputs.

Table 3.3.3 interval time of Time-base Counter (f = fc)

	TBC1	TBC2	TBC3	TBC4	TBC5	TBC6	TBC7	TBC8	TBC9	TBC10
Interval Time [s]	$2^2/f$	$2^3/f$	$2^4/f$	$2^5/f$	$2^6/f$	$2^7/f$	$2^8/f$	$2^9/f$	$2^{10}/f$	$2^{11}/f$
at 20 MHz [μs]	0.2	0.4	0.8	1.6	3.2	6.4	12.8	25.6	51.2	102.4

TBC11	TBC12	TBC13	TBC14	TBC15	TBC16	TBC7	TBC18	TBC19	TBC20
$2^{12}/f$	$2^{13}/f$	$2^{14}/f$	$2^{15}/f$	$2^{16}/f$	$2^{17}/f$	$2^{18}/f$	$2^{19}/f$	$2^{20}/f$	$2^{21}/f$
204.8	409.6	819.2	1638.4	3276.8	6553.6	13107.2	26214.4	52428.8	104857.6

3.3.3 Real Time Counter (RTC)

The TMP93C071 has the real time counter (RTC) which generates a periodic interrupt request. The RTC is controlled by System Control Register2 (SYSCR2).

The RTC is a 17bit binary counter and its clock source is selected either low frequency clock (fs) or system clock (f_{SYS}/2). To start/stop the counter is controlled by <RTCST>.

The period of interrupt request INTRTC is selected from 3 types by setting <RTCIS1, RTCIS0>.

Table 3.3.4 shows the period of interrupt request INTRTC.

SYSCR2 (006CH)

	7	6	5	4	3	2	1	0
bit Symbol				RTCCK	RTCST	RTCIS1	RTCIS0	
Read/Write	(R/W)	(R/W)		R/W	R/W	R/W		
After reset	(0)	(0)		0	0	0		
Function				RTC clock source select 0: fs(32KHz) 1 : f_{SYS}/2	RTC count 0: Stop & Counter Clear 1: Start	Interval time control of RTC interrupt 00: $f_{SYS}/2^{16}$ or $fs/2^{15}$ (1sec) 01: $f_{SYS}/2^{17}$ or $fs/2^{16}$ (0.5sec) 11: Inhibit [Hz]		

Table 3.3.4 INTRTC Interrupt Interval

<RTCCK>	<RTCIS>	INTRTC interrupt interval
0 (fs = 32.768 kHz)	00	1 s
	01	2 s
	10	0.5 s
1 (f_{SYS} = fc/2, fc = 20 MHz)	00	6.55 ms
	01	13.1 ms
	10	3.28 ms

3.3.4 Standby Controller

(1) Halt mode

When the HALT instruction is executed, the operating mode changes RUN, IDLE2, IDLE1 or STOP mode depending on the contents of the HALT mode setting register WDMOD<HALTM1,0>. Figure 3.3.6 shows the alternative states of watchdog timer mode registers.

WDMOD (005CH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	Watchdog timer 0: disable 1: enable	Watchdog timer detect time selection 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Warm-Up Timer 0: 2^{14}/ frequency inputted 1: 2^{16}/ frequency selection	Halt mode selection 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Runaway detection internal reset control 1: Executes internal reset by runaway detection	STOP mode pin control 1: drive pins in STOP mode

Pin state control in STOP mode

0	I/O off
1	Retains the state befor halt

Halt mode setting

00	RUN mode (only CPU stop)
01	STOP mode (all circuits stop)
10	IDLE1 mode (only oscillator operating)
11	IDLE2 mode (partial I/O operating)

Warm-up time selection at returning from the stop mode (see table 3.3 5)

0	2^{14} select clock frequency
1	2^{16}/ select clock frequency

Figure 3.3.6 Watchdog timer mode register

The features of RUN, IDLE2, IDLE1 and STOP modes are as follows.

① RUN: Only the CPU halts; power consumption remains unchanged.

② IDLE2: The built-in oscillator and the specified I/O operates.
The Power Consumption is redced to 1/2 than that during NORMAL operation.

③ IDLE1: Only the built-in oscillator operates, while all other built-in circuits stop. The Power Consumption is reduced to 1/5 or less than that during NORMAL operation.

④ STOP: All internal circuits including the built-in oscillator stop. This greatly reduces power consumption.

Table 3.3.5 I/O Operation During Halt mode

Halt mode	RUN	IDLE2	IDLE1	STOP
WDMOD<HALTM1,0>	00	11	10	01
CPU	Stop			
I/O port	Keep the state when the "HALT" instruction was executed.			See table 3.3.8
8bit Timer(TC0)	Operate		Stop	
16-bit Timer (TC 1,2,3,4,5)				
Dual Clock				
Watchdog Timer				
Interrupt controller				
Chip select / Wait counter				
SIO 0,1				
12CBUS				
UART				
8-bit PWM				
14-bit PWM				
Timing Pulse Generator (TPG 0,1)				
Color Rotary				
VISS / VASS				
CSYNC				
PV / PH				
Capture input (RMTIN)				
Capture 0				
Remote Control Input (RMTIN)				
Time Base Counter (TBC)				
A/D Converter				
Read Time Counter (RTC)				

(2) How to Release the Halt mode

These HALT states can be released by resetting or requesting an interrupt. The halt release sources are determined by the combinations between the states of interrupt mask register <IFF2 to 0> and the halt modes. The details for releasing the HALT status are shown in Table 3.3.6.

• Released by requesting an interrupt

The operating released from the halt mode depends on the interrupt enabled status. When the interrupt request level set before executing the HALT instruction exceeds the value of the interrupt mask register, the interrupt due to the source is processed after releasing the halt mode, and CPU starts executing an instruction that follows the HALT instruction. When the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the halt mode is not executed. (in non-maskable interrupts, interrupt processing is processed after releasing the halt mode regardless of the value of the mask register.)

However only for INT0 and INT1 interrupts, even if the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the halt mode is executed. In this case, interrupt processing is not processed, and CPU starts executing the instruction next to the HALT instruction, but the interrupt request flag is held at "1".

• Release by resetting

Releasing all halt status is executed by resetting.

When the STOP mode is released by RESET, it is necessary enough resetting time (3 ms or more) to set the operation of the oscillator to be stable.

When releasing the halt mode by resetting, the internal RAM data keeps the state before the "HALT" instruction is executed. However the other setting contents are initialized. (Releasing due to interrupts keep the state before the "HALT" instruction is executed.)

Table 3.3.6 Halt releasing source and Halt releasing operation

Interrupt receiving status			Interrupt enable (Interrupt level)≥(Interrupt mask)				Interrupt disable (Interrupt level)<(Interrupt mask)			
Halt mode			RUN	IDEL2	IDEL1	STOP	RUN	IDEL2	IDEL1	STOP
Halt releasing source	Interrupt	INTWD	⊙	×	×	×	—	—	—	—
		INT0, 1	⊙	⊙	⊙	⊙*1	O	O	O	O*1
		INTCAP0, 1	⊙	⊙	×	×	×	×	×	×
		INTTTG0, 1	⊙	⊙	×	×	×	×	×	×
		INT12CB, INT12DMA	⊙	⊙	×	×	×	×	×	×
		INTTBC	⊙	⊙	×	×	×	×	×	×
		INTSIO0, 1	⊙	⊙	×	×	×	×	×	×
		INTRX, INTTX	⊙	⊙	×	×	×	×	×	×
		INTVA	⊙	⊙	×	×	×	×	×	×
		INT2, 3, 4	⊙	⊙	×	×	×	×	×	×
		INTT0-5	⊙	⊙	×	×	×	×	×	×
		INTAD	⊙	⊙	×	×	×	×	×	×
		INTRTC	⊙	⊙	×	×	×	×	×	×
	Reset Input			⊙	⊙	⊙	⊙	⊙	⊙	⊙

⊙: After releasing the halt mode, CPU starts interrupt processing (RESET initializes LSI)

O: After releasing the halt mode, CPU starts executing an instruction that follows the HALT instruction.

×: It can not be used to release the halt mode.

—: This combination type does not exist because the priority level (interrupt request level) of non-maskable interrupts is fixed to highest priority level "7".

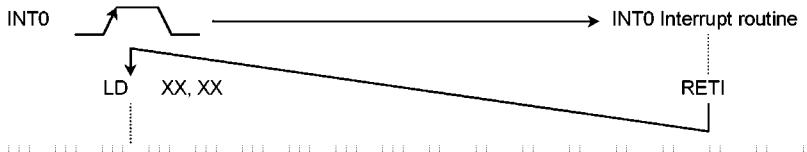
*1: Releasing the halt mode is executed after passing the warming-up time.

Note: When releasing the halt mode is executed by INT0 or INT1 interrupt of the level mode in the interrupt enabled status, hold level "H" until starting interrupt processing. If level "L" is set before starting interrupt processing, interrupt processing is incorrectly started.

(Example releasing "RUN" mode)

INT0 interrupt releases HALT state when the RUN mode is on.

```
Address     :
8200H       LD      (IIMC0), 01H      ; INTO interrupt input enable
8203H       LD      (IIMC1), 00H      ; Selects interrupt rising edge for INT0
8206H       LD      (INT0CP1), 06H    ; Sets interrupt level to '6' for INT0
8209H       EI      5                 ; Sets interrupt level to '5' for CPU
820BH       LD      (WDMOD),00H       ; Sets HALT mode to 'RUN'
820EH       HALT                      ; halts CPU
            :
```



When halt is released by reset, the states (including those of the internal RAM) before halt state was entered can be maintained. However, if the HALT instruction is executed within the internal RAM, the contents of the RAM way not be maintained. In this case, we recommend releasing the halt state using INT0.

(3) Operation

① RUN mode

In the RUN mode, the system clock continues to operate even after a HALT instruction is executed. Only the CPU stops executing the instruction. In the HALT state, an interrupt request is sampled with the falling edge of the internal "CLK" signal.

Releasing the RUN mode is executed by the external/internal interrupts. (See Table 3.3.6 Halt releasing source and Halt releasing operation.)

Figure 3.3.7 shows the interrupt timing for releasing the HALT state by interrupts in the RUN/IDLE2 mode.

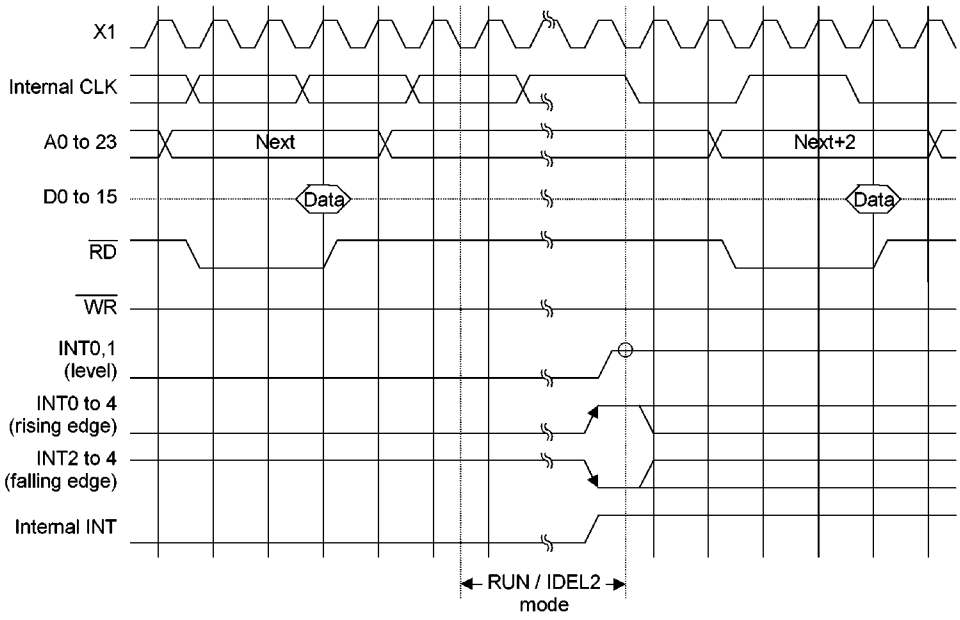


Figure 3.3.7 Timing Chart for Releasing the HALT State by Interrupt in RUN/IDLE2 modes

② IDLE2 mode

In the IDLE2 mode, the system clock is supplied to only specific internal I/O devices, and the CPU stops executing the current instruction.

In the IDLE2 mode, the HALT state is released by an interrupt with the same timing as in the RUN mode. The IDLE2 mode is released by external/internal interrupt, except INTWD/INTAD interrupts. (See table 3.3.6 Halt releasing source and Halt releasing operation.)

In the IDLE2 mode, the watchdog timer should be disabled before entering the halt status to prevent the watchdog timer interrupt occurring just after releasing the halt mode.

③ IDLE1 mode

In the IDLE1 mode, only the internal oscillator operates. The system clock in the MCU stops.

In the HALT state, and interrupt request is sampled aynchronunsly with the system clock, however the HALT release (restart of operation) is performed synchronously with it.

IDLE1 mode is released by external interrupts (INT0, INT1). (See table 3.3.6 Halt releasing source and Halt releasing operation.)

Figure 3.3.8 illustrates the timing for releasing the HALT state by interrupts in the IDLE1 mode.

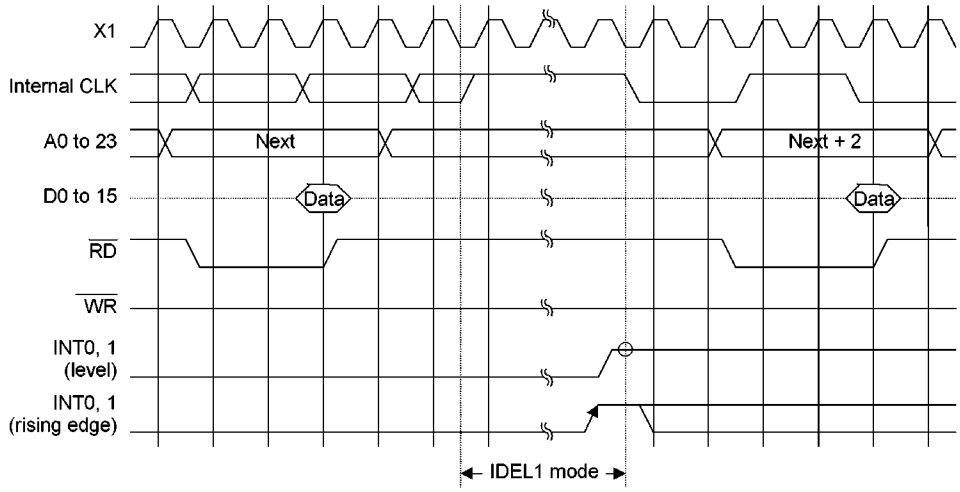


Figure 3.3.8 Timing Chart of HALT Released by Interrupts in IDLE1 Mode

④ STOP mode

The STOP mode is selected to stop all internal circuits including the internal oscillator. The pin status in the STOP mode depends on setting of a bit in the watchdog timer mode register WDMOD<DRVE>. (See Figure 3.3.6 for setting of WDMOD <DRVE>.) Table 3.3.8 summarizes the state of these pins in the STOP mode.

The STOP mode is released by external interrupts (INT0, INT1). When the STOP mode is released, the system clock output starts after warm-up time required to attain stable oscillation. The warm-up time can be set using WDMOD<WARM>. See the example of warm-up time (Table 3.3.7).

Figure 3.3.9 illustrates the timing for releasing the HALT state by interrupts during the STOP mode.

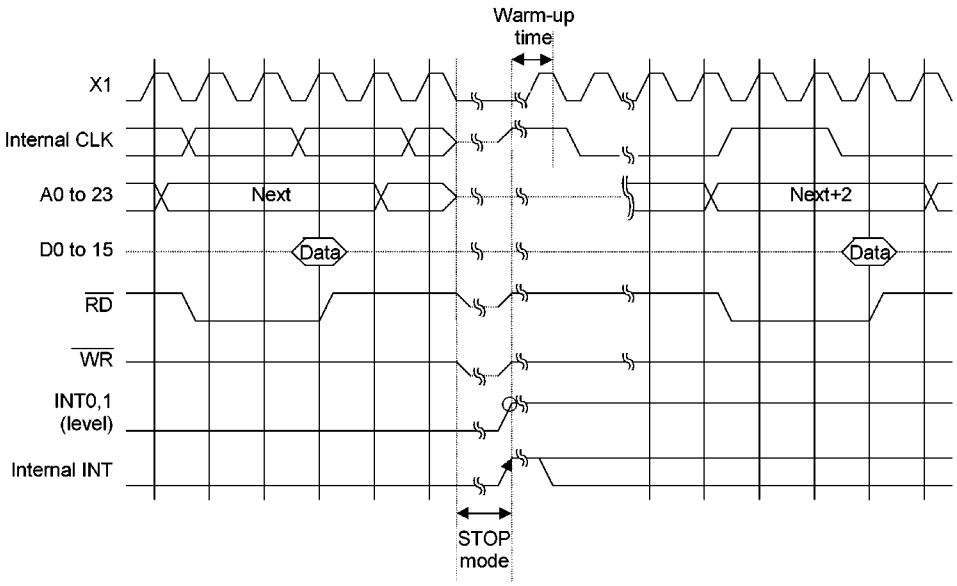


Figure 3.3.9 Timing Chart of HALT State Release by Interrupts in STOP Mode

Table 3.3.7 The example of Warm-up time after releasing the stop mode

Clock operation frequency after the stop mode	Warm-up time [ms]		Clock frequency
	WDMOD<WARM> = 0	WDMOD<WARM> = 1	
fc	0.8192	3.2768	fc = 20 MHz
fs	500	2000	fs = 32.768 kHz

How to calculate the warm-up time

WDMOD<WARM> = "0": Clock operation frequency after the 2^{14}/STOP mode

WDMOD<WARM> = "1": Clock operation frequency after the 2^{16}/STOP mode

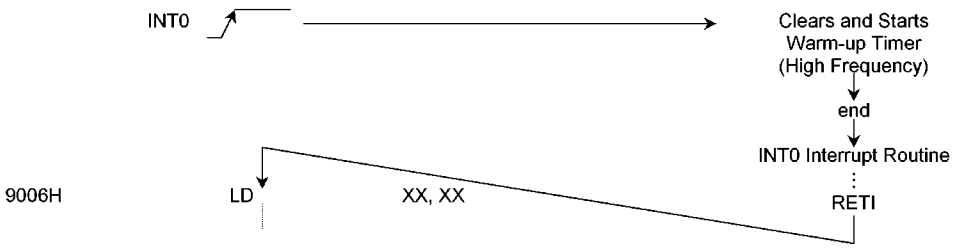
The NORMAL / SLOW mode selection is possible after released STOP mode.

This is selected by SYSCR0 <RSYSCK>. Therefore, Setting to <RSYSCK>, <RXEN>, <RXTEN> is necessary before "HALT" instruction is executed.

(Setting Example)

The STOP mode is entered when the low frequency (fs) operates, and after that high frequency operates after releasing by INT0.

```
Address
            SYSCR0 EQU      006EH
            SYSCR1 EQU      006FH
            WDMOD  EQU      005CH
8FFDH       LD              (SYSCR1), 08H       ; fsys = fs/2
9000H       RES             4, (WDMOD)          ; Sets Warm-up Time to 2^14/fc
9002H       LD              (SYSCR0),-11000- -B; Operates High Frequency after relesed.
                                (Note)          -: no change
9005H       HALT
```



Note: When different modes are used before and after STOP mode as the above mentioned, there is possible to release the HALT mode without changing the operation mode by acceptance of the halt release interrupt request during execution of "HALT" instruction (during 8 states). In the system which accepts the interrupts during execution "HALT" instruction, set the same operation mode before and after the STOP mode.

Table 3.3.8 Pin states in STOP mode

Pin Name	I/O	TMP93C071F	
		WDMOD <DRVE> = "0"	WDMOD <DRVE> = "1"
D0 to D15	I/O	Hz	Hz
A16 to A19 A20 to A23/P24 to P27	Output	Hz	Output
A0 to A15	Output	Hz	Output
RD	Output	Hz	"H" Level Output
X1	Input	—	—
X2	Output	"H" Level Output	"H" Level Output
PB0/PB1 XT1 XT2	I/O Input Output	— / Hz — "H" Level Output	— / Output — "H" Level Output
P40 to P47/AIN3 to AIN10	Input	—	—
P50 to P52, P55 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, PA0 to PA5, PB2 to PB5, PC0 to PC4	Input Output	— Hz	— Output
PWM0, 1	Output	Hz	Output
P53/INT1, P54/INT0	Input Output	Input Hz	Input Output
AM8/$\overline{16}$	Input	Input	Input
$\overline{EA}$	Input	"L" Level fixed	"L" Level fixed
$\overline{RESET}$	Input	Input	Input
ADREF	Input	(Open status is available)	(Open status is available)

—: Inputs is not accepted.

Input: Input gate in operation. Fix input voltage level to 0 or 1 so that the input pin stays constant.

Output: Output state

Hz: High Impedance

3.4 Interrupts

TLCS-900 interrupts are controlled by the CPU interrupt mask flip-flop <IFF2 to 0> and the built-in interrupt controller.

Altogether the TMP93C071 has the following 34 interrupt sources:

Internal interrupts 29
- Software interrupts: 8
  (Illegal instruction execution is included: 1)
- Interrupts from built-in I/Os: 21

External instructions 5
- External pins (INT0, INT1, INT2 to INT4)

A fixed individual interrupt vector number is assigned to each interrupt source; six levels of priority can also be assigned to each maskable interrupt. Non-maskable interrupts have a fixed priority of 7.

When an interrupt is generated, the interrupt controller sends the value of the priority of the interrupt source to the CPU. When more than one interrupt is generated simultaneously, the interrupt controller sends the value of the highest priority (7 for non-maskable interrupts is the highest) to the CPU.

The CPU compares the value of the priority sent with the value in the CPU interrupt mask register (IFF2 to 0). If the value is greater than that the CPU interrupt mask register, the interrupt is accepted. The value in the CPU interrupt mask register (IFF2 to 0) can be changed using the EI instruction (Executing EI n changes the contents of <IFF2 to 0> to n). For example, programming EI 3 enables acceptance of maskable interrupts with a priority of 3 or greater, and non-maskable interrupts which are set in the interrupt controller. The DI instruction (<IFF2 to 0> = 7) operates in the same way as the EI 7 instruction. Since the priority values for maskable interrupts are 0 to 6, the DI instruction is used to disable acceptance of maskable interrupts. The EI instruction becomes effective immediately after execution (With the TLCS-90, the EI instruction becomes effective after execution of the subsequent instruction).

In addition to the general-purpose interrupt processing mode described above, there is also a Micro DMA processing mode. Micro DMA is a mode used by the CPU to automatically transfer byte or word data. It enables the CPU to process interrupts such as data saves to built-in I/Os at high speed.

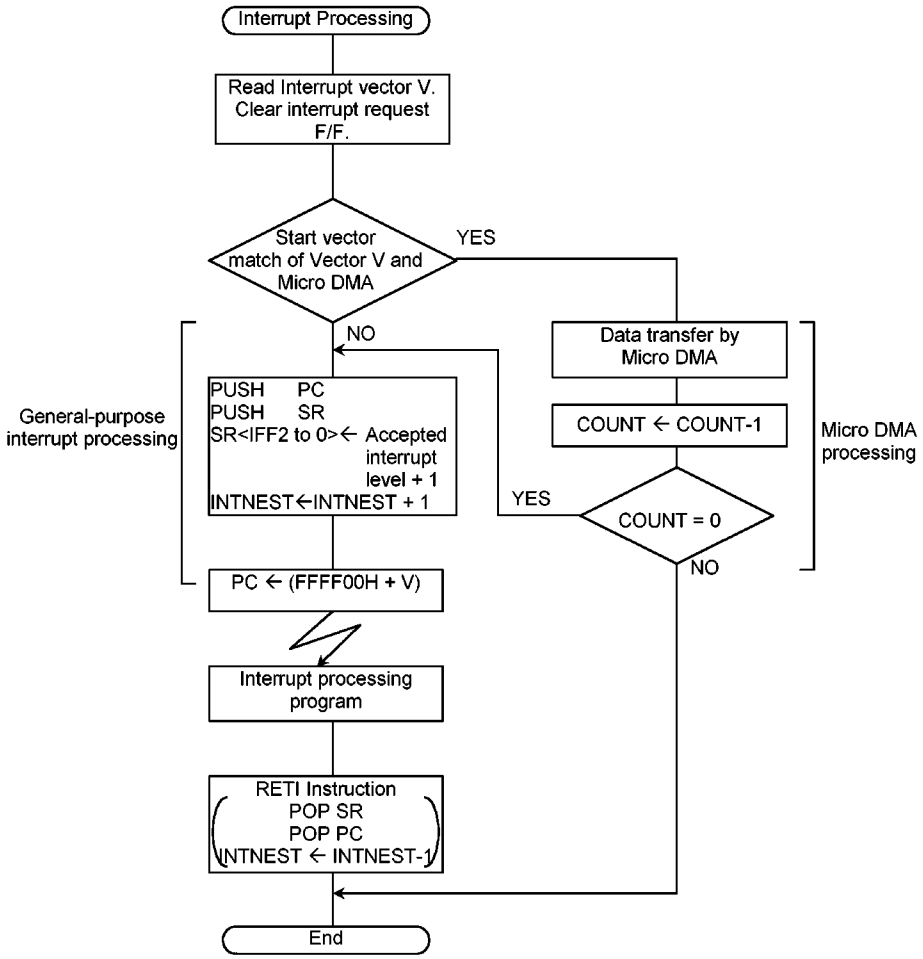


Figure 3.4.1 Interrupt Processing Flowchart

3.4.1 General-Purpose Interrupt Processing

When accepting an interrupt, the CPU operates as follows:

(1) The CPU reads the interrupt vector from the interrupt controller. When more than one interrupt with the same level is generated simultaneously, the interrupt controller generates interrupt vectors in accordance with the default priority (which is fixed as follows: the smaller the vector value, the higher the priority), then clears the interrupt request.
(2) The CPU pushes the program counter and the status register to the system stack area (area indicated by the system mode stack pointer (XSP)).
(3) The CPU sets a value in the CPU interrupt mask register <IFF2 to 0> that is higher by 1 than the value of the accepted interrupt level. However, if the value is 7, 7 is set without an increment.
(4) The CPU increments the INTNEST (Interrupt Nesting Counter).
(5) The CPU jumps to address stored at FFFF00H + interrupt vector, then starts the interrupt processing routine.

The following diagram shows all the above processing state number.

Bus Width of Stack Area	Bus Width of Interrupt Vector Area	Interrupt processing state number
8 bit	8 bit	35
	16 bit	31
16 bit	8 bit	29
	16 bit	25

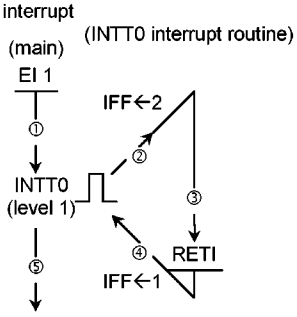
To return to the main routine after completion of the interrupt processing, the RETI instruction is usually used. Executing this instruction restores the contents of the program counter and the status registers and decrements INTNEST (Interrupt Nesting Counter).

Though acceptance of non-maskable interrupts cannot be disabled by program, acceptance of maskable interrupts can. A priority can be set for each source of maskable interrupts. The CPU accepts an interrupt request with a priority higher than the value in the CPU mask register <IFF2 to 0>. The CPU mask register <IFF2 to 0> is set to a value higher by 1 than the priority of the accepted interrupt. Thus, if an interrupt with a level higher than the interrupt being processed is generated, the CPU accepts the interrupt with the higher level, causing interrupt processing to nest.

The interrupt request with a priority higher than the accepted now interrupt during the CPU is processing above (1) to (5) is accepted before the 1'st instruction in the interrupt processing routine, causing interrupt processing to nest. (This is the same case of over lapped each Non-Maskable interrupt (level "7").) The CPU does not accept an interrupt request of the same level as that of the interrupt being processed.

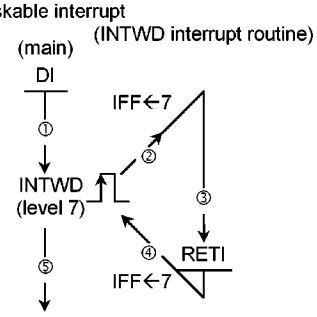
Resetting initializes the CPU mask registers <IFF2 to 0> to 7; therefore, maskable interrupts are disabled. The following (1) to (5) show a flowchart of interrupt processing.

(1) Maskable interrupt



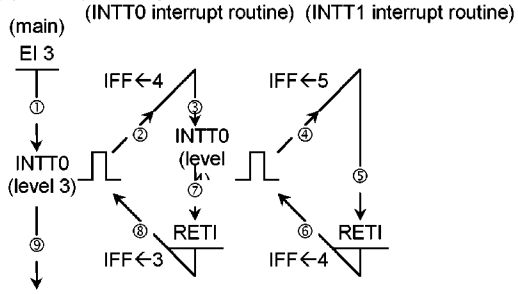
During execution of the main program, the CPU accepts an interrupt request. The CPU increments the IFF so that the interrupts of level 1 are not accepted during processing the interrupt routine.

(2) Non-maskable interrupt



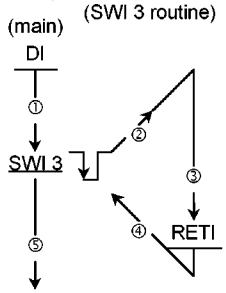
DI instruction is executed In the main program, so that the interrupts of only level 7 are accepted. The CPU does not increment the IFF even if the CPU accepts an interrupt request of level 7.

(3) Interrupt nesting



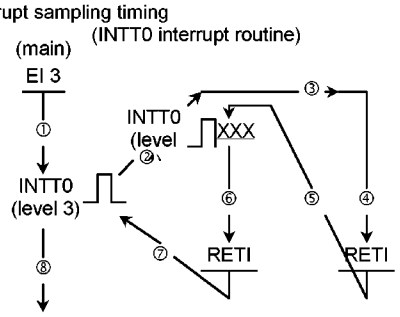
During processing the interrupts of level 3, the IFF is set to 4. When an interrupt with a level higher then level 4 is generated, the CPU accepts the interrupt processing to nest.

(4) Software interrupt



The CPU accepts the software interrupt request during DI status(IFF=7) because of the level 7. The IFF not changed by the software interrupts.

(5) Interrupt sampling timing



If an interrupt with a level higher than the interrupt being processed is generated, the CPU accepts the interrupt with the higher level. The program counter which returns at ⑤ is the start address of INTT0 interrupt routine.

(example)___(underline) : Instruction
①,②,… : Execution flow

The address FFFF00H to FFFFFFH (256 byte) of the TMP93C071 are assigned for interrupt vector area.

Table 3.4.1 TMP93C071 Interrupt Table

Default priority	Type	Interrupt source	Vector value "V"	Address refer to vector	High-speed micro DMA start vector	Input Level/ Edge
1	non-	Reset, or SWI0 instruction	0000H	FFFF00H	—	
2	maskable	SWI1 instruction	0004H	FFFF04H	—	
3		INTUNDEF: illegal instruction, or SWI2	0008H	FFFF08H	—	
4		SWI3 instruction	000CH	FFFF0CH	—	
5		SWI4 instruction	0010H	FFFF10H	—	
6		SWI5 instruction	0014H	FFFF14H	—	
7		SWI6 instruction	0018H	FFFF18H	—	
8		SWI7 instruction	001CH	FFFF1CH	—	
9		(Reserved: NMI)	0020H	FFFF20H	—	Edge
10		INTWD: watchdog timer	0024H	FFFF24H	09H	Edge
11	maskable	INT0 pin: External interrupt input 0	0028H	FFFF28H	0AH	Edge/Level
12		INTCAP1: Capture 1 interrupt	002CH	FFFF2CH	0BH	Level
13		INTCAP0: Capture 0 interrupt	0030H	FFFF30H	0CH	Level
14		INTTPG0: Timing Pulse Generator0 interrupt	0034H	FFFF34H	0DH	Edge (note)
15		INTTPG1: Timing Pulse Generator1 interrupt	0038H	FFFF38H	0EH	Edge
16		INTI2DMA: I2CBUS micro DMA Activation Interrupt	003CH	FFFF3CH	0FH	Edge
17		INTI2CB: I2CBUS interrupt	0040H	FFFF40H	10H	Edge
18		INTTBC: Time Base Timer interrupt	0044H	FFFF44H	11H	Edge
19		INTSIO0: Serial I/O 0 interrupt	0048H	FFFF48H	12H	Level
20		INTSIO1: Serial I/O 1 interrupt	004CH	FFFF4CH	13H	Level
21		INTRX: UART receive interrupt	0050H	FFFF50H	14H	Level
22		INTTX: UART transmit interrupt	0054H	FFFF54H	15H	Edge
23		INTVA: VISS/VASS detection	0058H	FFFF58H	16H	Edge
24		INT1 pin: External interrupt input 1	005CH	FFFF5CH	17H	Edge/Level
25		INT2 pin: External interrupt input 2	0060H	FFFF60H	18H	Edge
26		INT3 pin: External interrupt input 3	0064H	FFFF64H	19H	Edge
27		INT4 pin: External interrupt input 4	0068H	FFFF68H	1AH	Edge
28		INTT0: 8-bit Timer TC0	006CH	FFFF6CH	1BH	Edge
29		INTT1: 16-bit Timer TC1	0070H	FFFF70H	1CH	Edge
30		INTT2: 16-bit Timer TC2	0074H	FFFF74H	1DH	Edge
31		INTT3: 16-bit Timer TC3	0078H	FFFF78H	1EH	Edge
32		INTT4: 16-bit Timer TC4	007CH	FFFF7CH	1FH	Edge
33		INTT5: 16-bit Timer TC5	0080H	FFFF80H	20H	Edge
34		INTAD: A/D conversion completion	0084H	FFFF84H	21H	Level
35		INTRTC: Real Time Counter	0088H	FFFF88H	22H	Edge
—		(Reserved)	008CH	FFFF8CH	23H	
to		to	to	to	to	
—		(Reserved)	00FCH	FFFFFCH	3FH	

Note: The INTTPG0 signal is generated as a level signal (FIFO empty) or as an edge signal (TPG03 output), but the interrupt controller receives the INTTPG0 only as an edge signal. When the INTTPG0 is used for a FIFO empty interrupt (a level signal), the interrupt controller also leaves a request flag (Flip/Flop) after clearing FIFO empty by setting next TPG0 data in an interrupt routine. Therefore, in this case, the INTTPG0 request flag has to be cleared before executing RETI instruction.

Setting to Reset / Interrupt Vector

① Reset Vector

FFFF00H	PC (7 to 0)
FFFF01H	PC (15 to 8)
FFFF02H	PC (23 to 16)
FFFF03H	XX

The vector base addresses are depended on the products.

Type No.	Vector base address	PC setting sequence after reset	Notes
TMP93C071	FFFF00H	PC (7 to 0) ← address FFFF00H PC (15 to 8) ← address FFFF01H PC (23 to 16) ← address FFFF02H	P27 to P24 / A23 to A20 are used as address bus: A23 to A20 after reset.

② Interrupt Vector (except Reset Vector)

Address refer to vector	PC (7 to 0)
+0	
+1	PC (15 to 8)
+2	PC (23 to 16)
+3	XX

XX: don't care

(Setting Example)

Sets the Reset Vector: FF0000H, INTWD Vector: FF9ABCH, INTAD Vector: FE3456H.

```
ORG     FFFF00H
DL      FF0000H; Reset = FF0000H

ORG     FFFF24H
DL      FF9ABCH; INTWD = FF9ABCH

ORG     FFFF84H
DL      FE3456H; INTAD = FE3456H

ORG     FF0000H
LD      A, B
        :
ORG     FF9ABCH
LD      B, C

ORG     FE3456H
LD      C, A
        :
```

Note:

ORG, DL are Assembler Directives.

- ORG: control location counter
- DL: defines long word (32-bits) data

3.4.2 Micro DMA

In addition to the conventional interrupt processing, the TLCS-900 also has a Micro DMA function. When an interrupt is accepted, in addition to an interrupt vector, the CPU receives data indicating whether processing is Micro DMA mode or general-purpose interrupt. If Micro DMA mode is requested, the CPU performs Micro DMA processing.

The TLCS-900 can process at very high speed because it has transfer parameters in dedicated registers in the CPU. Since those dedicated registers are assigned as CPU control registers, they can only be accessed by the LDC instruction.

(1) Micro DMA operation

Micro DMA operation starts when the accepted interrupt vector value matches the Micro DMA start vector value. The Micro DMA has four channels so that it can be set for up to four types of interrupt source.

When a Micro DMA interrupt is accepted, data is automatically transferred from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decremented. If the value in the counter after decrementing is other than 0, Micro DMA processing is completed; if the value in the counter after decrementing is 0, general-purpose interrupt processing is performed.

32-bit control registers are used for setting transfer source / destination addresses. However, the TLCS-900 has only 24 address pins for output. A 16 Mbyte space is available for the Micro DMA.

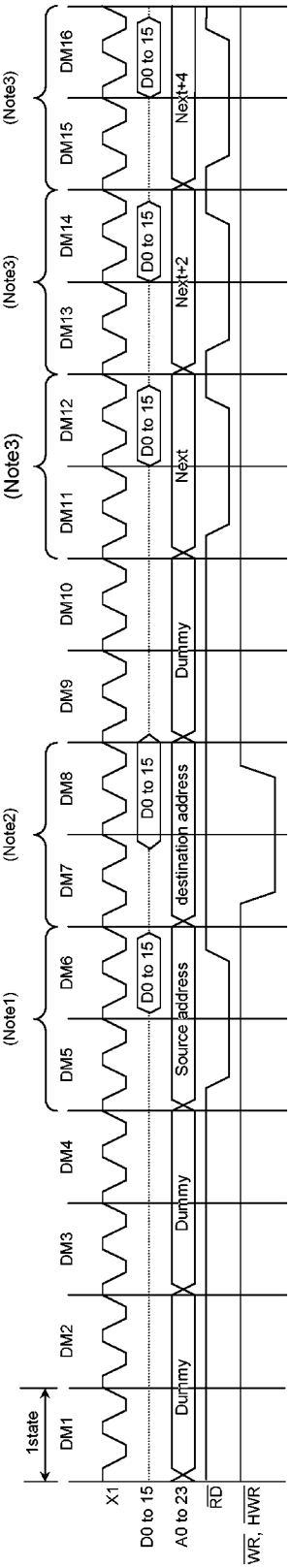
There are two data transfer modes: one-byte mode and one-word mode. Incrementing, decrementing, and fixing the transfer source / destination address after transfer can be done in both modes. Therefore data can easily be transferred between I/O and memory and between I/Os. For details of transfer modes, see the description of transfer mode registers.

The transfer counter has 16 bits, so up to 65536 transfers (the maximum when the initial value of the transfer counter is 0000H) can be performed for one interrupt source by Micro DMA processing.

When the transfer counter is decremented to "0" after data is transferred with micro DMA, general-purpose interrupt processing is performed. After processing the general-purpose interrupt, starting the interrupts of the same channel restarts the transfer counter from 65536. If necessary, reset the transfer counter.

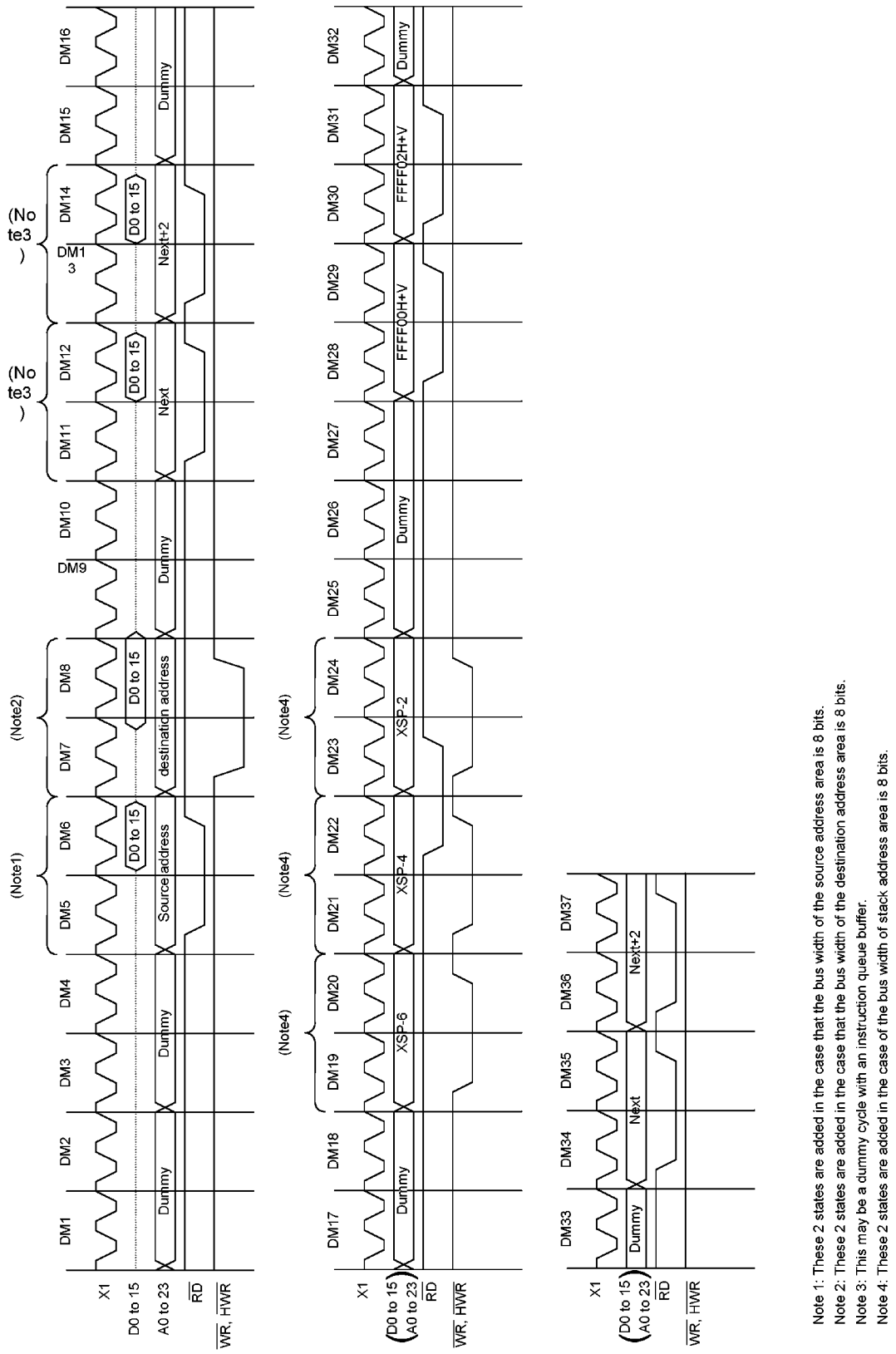
Interrupt sources processed by Micro DMA processing are those with the Micro DMA start vectors listed in Table 3.4.1.

The following timing chart is a Micro DMA cycle of the Transfer Address INC rement mode (Condition: 16 bit Bus width for 16 MByte, 0 waits).



Note 1: These 2 states are added in the case that the bus width of the source address area is 8 bits.
Note 2: These 2 states are added in the case that the bus width of the destination address area is 8 bits.
Note 3: This may be a dummy cycle with an instruction queue buffer

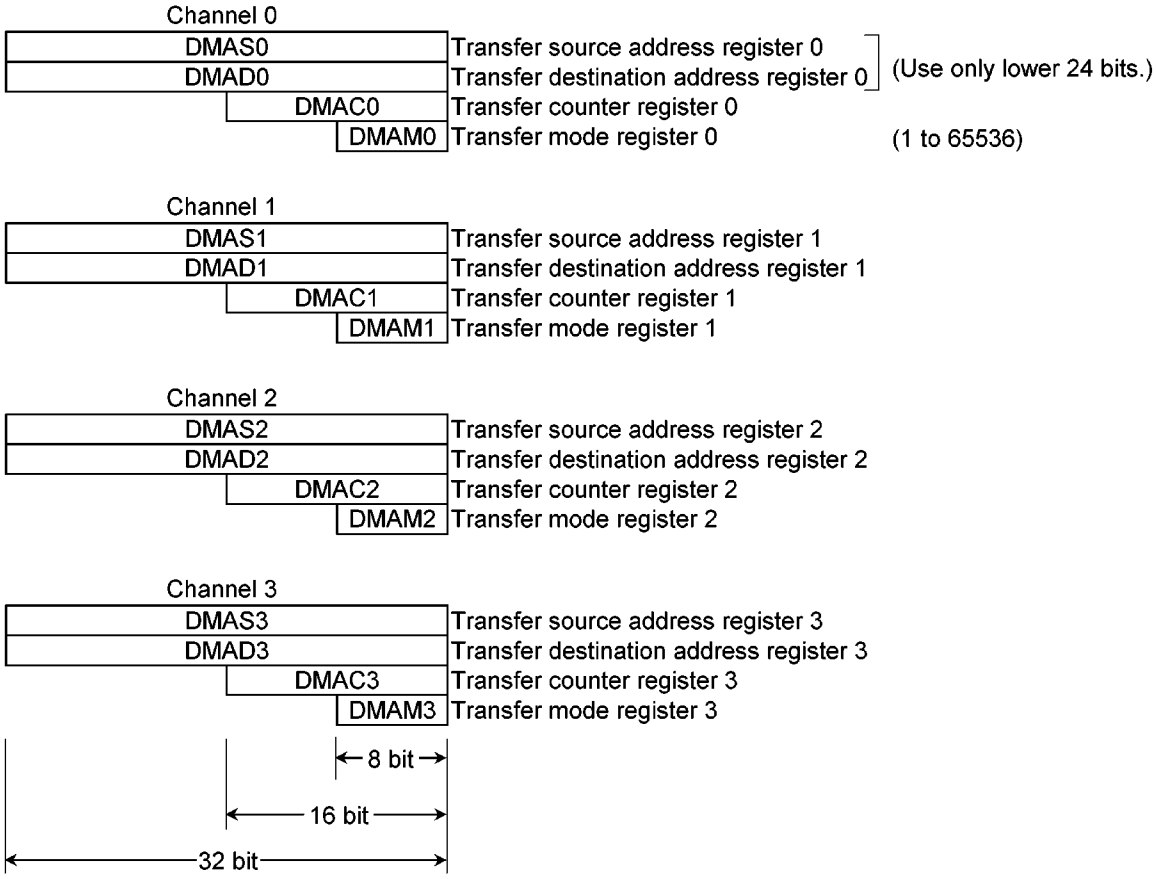
Figure 3.4.1 Micro DMA cycle (COUNT ≠ 0)



Note 1: These 2 states are added in the case that the bus width of the source address area is 8 bits.
Note 2: These 2 states are added in the case that the bus width of the destination address area is 8 bits.
Note 3: This may be a dummy cycle with an instruction queue buffer.
Note 4: These 2 states are added in the case of the bus width of stack address area is 8 bits.

Figure 3.4.2 Micro DMA cycle (COUNT = 0)

(2) Register configuration (CPU control register)



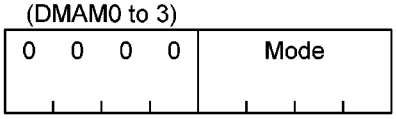
These Control Register can not be set only “LDC cr, r" instruction.

Example:

```
LD      XWA, 100H
LDC     DMAS0, XWA
LD      XWA, 50H
LDC     DMAD0, XWA
LD      WA, 40H
LDC     DMAC0, WA
LD      A, 05H
LDC     DMAM0, A
```

(3) Transfer mode register details

(DMAM0 to 3)



Note: When setting values for this register, set the upper 4 bits to 0

Z: 0 = byte transfer, 1 = word transfer

						execution time (Min) at 20 MHz
0	0	0	Z	Transfer destination address INC mode (DMADn +) ← (DMASn) DMACn←DMACn -1 if DMACn = 0 then INT.	for I/O to memory	16 states (1.6 μs)
0	0	1	Z	Transfer destination address DEC mode (DMADn -) ← (DMASn) DMACn←DMACn -1 if DMACn = 0 then INT.	for I/O to memory	16 states (1.6 μs)
0	1	0	Z	Transfer source address INC mode (DMADn) ← (DMASn +) DMACn←DMACn -1 if DMACn = 0 then INT.	for memory to I/O	16 states (1.6 μs)
0	1	1	Z	Transfer source address DEC mode (DMADn +) ← (DMASn) DMACn←DMACn -1 if DMACn = 0 then INT.	for I/O to memory	16 states (1.6 μs)
1	0	0	Z	Fixed address mode (DMADn) ← (DMASn) DMACn←DMACn -1 if DMACn = 0 then INT.	I/O to I/O	16 states (1.6 μs)
1	0	1	1	Counter mode (DMADn +) ← (DMASn) DMACn←DMACn -1 if DMACn = 0 then INT.	for interrupt counter	11 states (1.1 μs)

(1 states = 100 ns at 20 MHz, High frequency mode)

Note1: n: corresponds to micro DMA channels 0 to 3.

DMADn + / DMASn +: Post-increment (Increments register value after transfer.)

DMADn - / DMASn -: Post-decrement (Decrement register value after transfer.)

Note2: Execution time: When setting source address/destination address area to 16-bit bus, 0 WAIT.

Clock condition: fc = 20 MHz

Note3: Do not use the codes other than the above mantioned codes for transfer mode register.

3.4.3 Interrupt Controller

Figure 3.4 (2) is a block diagram of the interrupt circuits. The left half of the diagram shows the interrupt controller; the right half includes the CPU interrupt request signal circuit and the HALT release signal circuit.

Each interrupt channel (total of 26 channels) in the interrupt controller has an interrupt request flip-flop, and interrupt priority setting register. The interrupt controller also has registers for storing the Micro DMA start vector. The interrupt request fip-flop is used to latch interrupt requests from peripheral devices. The flip-flop is cleared to 0 at reset, when the CPU reads the interrupt channel vector after the acceptance of interrupt, or when the CPU executes an instruction that clears the interrupt of that channel (writes 0 in the clear bit of the interrupt priority setting register).

For example, to clear the INT0 interrupt request, set the register after the DI instruction as follows.

INT0CP1 ← - - - - 0 - - - B

The status of the interrupt request flip-flop is detected by reading the clear bit. Detects whether there is an interrupt request for an interrupt channel.

The interrupt priority can be set by writing the priority in the interrupt priority setting register (eg, INT0CP1, INTCP0TG0, etc.) provided for each interrupt source. Interrupt levels to be set are from 1 to 6. Writing 0 or 7 as the interrupt priority disables the corresponding interrupt request. The priority of the non-maskable interrupt (watchdog timer, etc.) is fixed to 7. If interrupt requests with the same interrupt level are generated simultaneously, interrupts are accepted in accordance with the default priority (the smaller the vector value, the higher the priority).

The interrupt controller sends the interrupt request with the highest priority among the simultaneous interrupts and its vector address to the CPU. The CPU compares the priority value <IFF2 to 0> set in the Status Register by the interrupt request signal with the priority value sent; if the latter is higher, the interrupt is accepted. Then the CPU sets a value higher than the priority value by 1 in the CPU SR<IFF2 to 0>. Interrupt requests where the priority value equals or is higher than the set value are accepted simultaneously during the previous interrupt routine. When interrupt processing is completed (after execution of the RETI instruction) , the CPU restores the priority value saved in the stack before the interrupt was generated to the CPU SR<IFF2 to 0>.

The interrupt controller also has four registers used to store the Micro DMA start vector. These are I/O registers; unlike other Micro DMA registers (DMAS, DMAD, DMAM, and DMAC). Writing the start vector of the interrupt source for the Micro DMA processing (see Table 3.4.1), enables the corresponding interrupt to be processed by Micro DMA processing. The values must be set in the Micro DMA parameter registers (eg, DMAS and DMAD) prior to the Micro DMA processing.

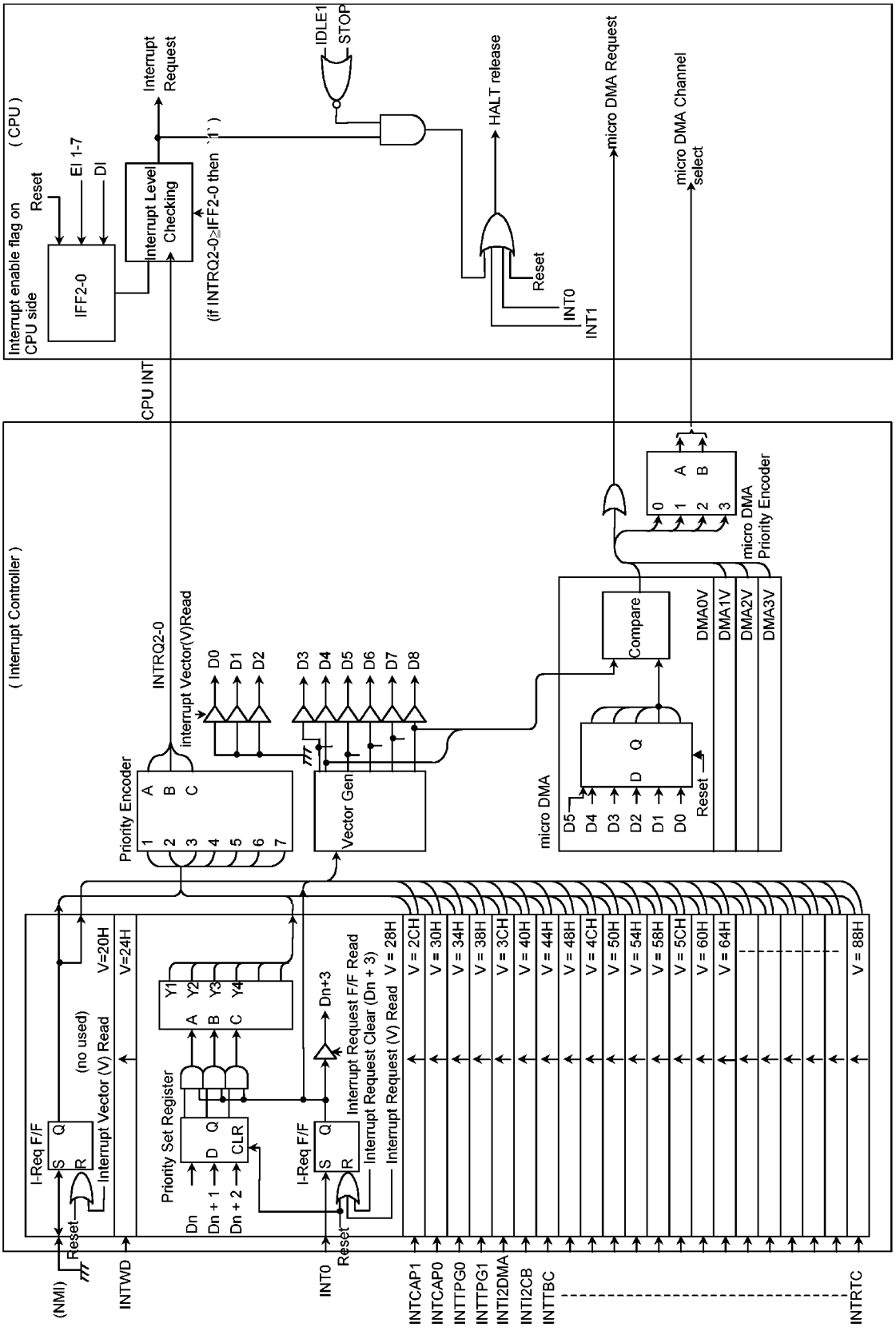


Figure 3.4.2 Block Diagram of Interrupt Controller

(1) Interrupt priority setting register (1/2)

(Prohibit read-modify-write)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0	
INT0CP1	INT0/	0070H	INTCAP1				INT0				← Interrupt source
	CAP1		ICAP1C	ICAP1M2	ICAP1M1	ICAP1M0	I0C	I0M2	I0M1	I0M0	← bit Symbol
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	← Read/Write
	Setting	RMW	0	0	0	0	0	0	0	0	← After reset
INTCP0TG0	CAP0/	0071H	INTTPG0				INTCAP0				
	TPG0		ITPG0C	ITPG0M2	ITPG0M1	ITPG0M0	ICAP0C	ICAP0M2	ICAP0M1	ICAP0M0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	
INTTG1DMA	TPG1/	0072H	INTI2DMA				INTTPG1				
	I2DMA		II2DMAC	II2DMAM 2	II2DMAM 1	II2DMAM 0	ITPG1C	ITPG1M2	ITPG1M1	ITPG1M0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	
INTI2CTBC	I2CBUS/	0073H	INTTBC				INTI2CB				
	TBC		ITBCC	ITBCM2	ITBCM1	ITBCM0	II2CC	II2CM2	II2CM1	II2CM0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	
INTSI0SI1	SIO0/	0074H	INTSIO1				INTSIO0				
	SIO1		ISIO1C	ISIO1M2	ISIO1M1	ISIO1M0	ISIO0C	ISIO0M2	ISIO0M1	ISIO0M0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	
INTRXTX	RX/	0075H	INTTX				INTRX				
	TX		ITXC	ITXM2	ITXM1	ITXM0	IRXC	IRXM2	IRXM1	IRXM0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	
INTVA1	VA/	0076H	INT1				INTVA				
	INT1		I1C	I1M2	I1M1	I1M0	IVAC	IVAM2	IVAM1	IVAM0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	
INT23	INT2/	0077H	INT3				INT2				
	INT3		I3C	I3M2	I3M1	I3M0	I2C	I2M2	I2M1	I2M0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	
INT4T0	INT4/	0078H	INTT0				INT4				
	Timer0		IT0C	IT0M2	IT0M1	IT0M0	I4C	I4M2	I4M1	I4M0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	

Interrupt priority setting register (2/2)

(Prohibit read-modify-write)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0	
INTT1T2	Timer1/	0079H	INTT2				INTT1				← Interrupt source
	Timer2		IT2C	IT2M2	IT2M1	IT2M0	IT1C	IT1M2	IT1M1	IT1M0	← bit Symbol
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	← Read/Write
	Setting	RMW	0	0	0	0	0	0	0	0	← After reset
INTT3T4	Timer3/	007AH	INTT4				INTT3				
	Timer4		IT4C	IT4M2	IT4M1	IT4M0	IT3C	IT3M2	IT3M1	IT3M0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	
INTT5AD	Timer5/	007BH	INTAD				INTT5				
	AD		IADCC	IADM2	IADM1	IADM0	IADC	IADM2	IADM1	IADM0	
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W	
	Setting	RMW	0	0	0	0	0	0	0	0	
INTRTCR		0080H	(Always set to 0)				INTRTC				
	RTC		(set to 0)	(set to 0)	(set to 0)	(set to 0)	IRTCMA	IRTCM2	IRTCM1	IRTCM0	
	Interrupt	Prohibit	(R/W)	(R/W)	(W)	(W)	R/W	W	W	W	
	Setting	RMW	(0)	(0)	(0)	(0)	0	0	0	0	

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Prohibit interrupt request.
0	0	1	Set interrupt request level to 1.
0	1	0	Set interrupt request level to 2.
0	1	1	Set interrupt request level to 3.
1	0	0	Set interrupt request level to 4.
1	0	1	Set interrupt request level to 5.
1	1	0	Set interrupt request level to 6.
1	1	1	Prohibit interrupt request.

IxxC	Function (Read)	Function (Write)
0	Indicates no interrupt request.	Clears interrupt request flag.
1	Indicates interrupt request.	don·t care

Note 1: Read-modify-write is prohibited.

Note 2: Note about clearing interrupt request flag.

The interrupt request flag of INTCAP1, INTCAP0, INTSIO0, INTSIO1, INTRX and INTAD are not cleared by writing 0 to IxxC because of they are level interrupts. They can be cleared only by resetting, reading captured data / ADREG /SC2BUF or reading/writing SC0BUF / SC1BUF.

Note 3: Note about clearing interrupt request flag.

When the INTTPG0 is used for a FIFO empty interrupt (a level signal), the interrupt controller also leaves a request flag (Flip/Flop) after clearing FIFO empty by setting next TPG0 data in an interrupt routin. Therefore, in this case, the INTTPG0 request flag has to be cleared before executing RETI instruction.

(2) External interrupt control

Interrupt Input Mode Control Register 1

IIMC0 (005EH)

	7	6	5	4	3	2	1	0
bit Symbol	—	—	—	I4IE (INT4)	I3IE (INT3)	I2IE (INT2)	I1IE (INT1)	I0IE (INT0)
Read/Write	W							
After reset				0	0	0	0	0
Function				External interrupt input enable/disable 0: Disable 1: Enable				

Prohibit read-modify-write

Interrupt Input Mode Control Register 1

IIMC1 (005FH)

	7	6	5	4	3	2	1	0
bit Symbol	I4EG	I3EG	I2EG	I1EG	I0EG	—	INTTPG0E	INTTPG0S
Read/Write	W						R/W	R/W
After reset	0	0	0	0	0		0	0
Function	INT4 edge selection 0: Rising 1: Falling	INT3 edge selection 0: Rising 1: Falling	INT2 edge selection 0: Rising 1: Falling	INT1 edge/level selection 0: Rising edge 1: Level	INT0 edge/level selection 0: Rising edge 1: Level		INTTPG0 interrupt TPG03 edge selection 0: Rising 1: Falling	INTTPG0 interrupt selection 0:FIFO empty inter-rupt 1: FIFO empty/ TPG03 inter-rupt

Prohibit read-modify-write

Note1: The INT0 and INT1 pins can also be used for standby release as described later. When these pins are not used for standby release, setting IIMC0<I1IE, I0IE> to 00 maintain the port function during standby mode.

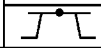
Note2: When the active edge is changed by the IIMC1<I4EG, I3EG, I2EG, I1EG, I0EG>, they must be changed after disabling interrupt.

Execution example:

```
LD    (INT0CP1)    , XXXX0000B; Disable the INT0 interrupt, Clear the INT0 interrupt request flag.
LD    (IIMC0)      , XXXXXXX0B; Disable the INT0 input
LD    (IIMC1)      , XXXX1XXXB; Change the active edge to Level from Rising edge.
LD    (INT0CP1)    , XXXX0nnnB; Set interrupt level "nnn" for INT0, Clear the interrupt request flag.
```

Note3: The minimum pulse width for the active edge is 4/fc [s] (200 ns at fc = 20 MHz).

Setting of External Interrupt Pin Functions

Interrupt	Shared pin	Mode	Setting method
INT0	P54	 Rising edge	IIMC1<I0EG> = 0, IIMC0<I0IE> = 1
		 Level	IIMC1<I0EG> = 1, IIMC0<I0IE> = 1
INT1	P53	 Rising edge	IIMC1<I1EG> = 0, IIMC0<I1IE> = 1
		 Level	IIMC1<I1EG> = 1, IIMC0<I1IE> = 1
INT2	P52	 Rising edge	IIMC1<I2EG> = 0, IIMC0<I2IE> = 1
		 Falling edge	IIMC1<I2EG> = 1, IIMC0<I2IE> = 1
INT3	P51	 Rising edge	IIMC1<I3EG> = 0, IIMC0<I3IE> = 1
		 Falling edge	IIMC1<I3EG> = 1, IIMC0<I3IE> = 1
INT4	P50	 Rising edge	IIMC1<I4EG> = 0, IIMC0<I4IE> = 1
		 Falling edge	IIMC1<I4EG> = 1, IIMC0<I4IE> = 1

(3) Micro DMA start vector

When the CPU reads the interrupt vector after accepting an interrupt, it simultaneously compares the interrupt vector with each channel's Micro DMA start vector (bits 3 to 8 of the interrupt vector). When the two match, the interrupt from the channel whose value matched is processed in Micro DMA mode.

If the interrupt vector matches more than one channel, the channel with the lower channel number has a higher priority.

Micro DMA0 Start Vector

DMA0V (007CH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol			DMA0V5	DMA0V 4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
Read/Write			W					
After reset			0	0	0	0	0	0
Function	Micro DMA channel 0 processed by matching bits 3 to 8 of the interrupt vector.							

Micro DMA1 Start Vector

DMA1V (007DH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol			DMA1V5	DMA1V 4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
Read/Write			W					
After reset			0	0	0	0	0	0
Function	Micro DMA channel 1 processed by matching bits 3 to 8 of the interrupt vector.							

Micro DMA2 Start Vector

DMA2V (007EH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol			DMA2V5	DMA2V 4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
Read/Write			W					
After reset			0	0	0	0	0	0
Function	Micro DMA channel 2 processed by matching bits 3 to 8 of the interrupt vector.							

Micro DMA3 Start Vector

DMA3V (007FH) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol			DMA3V5	DMA3V 4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
Read/Write			W					
After reset			0	0	0	0	0	0
Function	Micro DMA channel 3 processed by matching bits 3 to 8 of the interrupt vector.							

(4) Notes

The instruction execution unit and the bus interface unit of this CPU operate independently of each other. Therefore, if the instruction used to clear an interrupt request flag of an interrupt is fetched before the interrupt is generated, it is possible that the CPU might execute the fetched instruction to clear the interrupt request flag while reading the interrupt vector after accepting the interrupt. If so, the CPU would read the default vector FFFF28H and start the interrupt processing from the address FFFF28H.

To avoid this, make sure that the instruction used to clear the interrupt request flag comes after the DI instruction.

Execute the POP SR instruction, which modifies the IFF value, during DI.

3.5 Functions of ports

The TMP93C071 has 57 bits for I/O ports, 8 bits for Input ports and 4 bits for Output ports. These port pins have I/O functions for the built-in CPU and internal I/Os as well as general-purpose I/O port functions. Table 3.5 lists the function of each port pin. Resetting makes the pins listed below function as general-purpose I/O ports.

I/O pins programmable for input or output are set to input ports including PB0/XT1, PB1/XT2. To set port pins for built-in functions, a program is required.

Table 3.5.1 Functions of ports

Port Name	Pin Name	Pin No.	Direction	Output Mode	Direction / Output Mode setting unit	Pin name for built-in function
Port 2	P24 to P27	4	Output	Push-Pull	(fixed)	A20 to A23
Port 4	P40 to P47	8	Input	—	(fixed)	AIN3 to AIN10
Port 5	P50	1	I/O	Push-Pull	Bit	INT4/TI3
	P51	1	I/O	Push-Pull	Bit	INT3/TI2
	P52	1	I/O	Push-Pull	Bit	INT2/TI1
	P53	1	I/O	Push-Pull	Bit	INT1
	P54	1	I/O	Push-Pull	Bit	INT0
	P55	1	I/O	Push-Pull	Bit	TI5/AIN0
	P56	1	I/O	Push-Pull	Bit	TI4/AIN1
	P57	1	I/O	Push-Pull	Bit	TI0/AIN2
Port 6	P60	1	I/O	Push-Pull/Open-Drain	Bit	PWM4/$\overline{CS0}$
	P61	1	I/O	Push-Pull/Open-Drain	Bit	PWM5/$\overline{CS1}$
	P62	1	I/O	Push-Pull/Open-Drain	Bit	PWM6/$\overline{CS2}$
	P63 to P67	5	I/O	Push-Pull/Open-Drain	Bit	PWM7 to PWM11
Port 7	P70	1	I/O	Push-Pull/Open-Drain	Bit	TXD
	P71	1	I/O	Push-Pull	Bit	RXD
	P72	1	I/O	Push-Pull	Bit	$\overline{CTS}$
	P73	1	I/O	Push-Pull/Open-Drain	Bit	SDA0
	P74	1	I/O	Push-Pull/Open-Drain	Bit	SCL0
	P75	1	I/O	Push-Pull/Open-Drain	Bit	SO0
	P76	1	I/O	Push-Pull	Bit	SI0
	P77	1	I/O	Push-Pull/Open-Drain	Bit	SCK0
Port 8	P80	1	I/O	Push-Pull	Bit	CTLIN
	P81	1	I/O	Push-Pull	Bit	DFGIN
	P82	1	I/O	Push-Pull	Bit	RMTIN
	P83	1	I/O	Push-Pull	Bit	EXT
	P84	1	I/O	Push-Pull	Bit	DPGIN
	P85	1	I/O	Push-Pull	Bit	CFGIN
	P86	1	I/O	Push-Pull	Bit	CSYNCIN
	P87	1	I/O	Push-Pull	Bit	COMPIN
Port 9	P90	1	I/O	Push-Pull/Open-Drain	Bit	TPG12
	P91 to P94	4	I/O	Push-Pull/Open-Drain	Bit	TPG01 to TPG04
	P95	1	I/O	Push-Pull/Open-Drain	Bit	TPG13
	P96	1	I/O	Push-Pull/Open-Drain	Bit	TPG10
	P97	1	I/O	Push-Pull/Open-Drain	Bit	TPG11
Port A	PA0	1	I/O	Push-Pull	Bit	PV-PH
	PA1	1	I/O	Push-Pull	Bit	HA(TPG05)
	PA2	1	I/O	Push-Pull	Bit	CR(TPG00)
	PA3	1	I/O	Push-Pull/Open-Drain	Bit	PWM2
	PA4	1	I/O	Push-Pull/Open-Drain	Bit	$\overline{WR}$
	PA5	1	I/O	Push-Pull/Open-Drain	Bit	PWM3/$\overline{HWR}$
Port B	PB0	1	I/O	Open-Drain	Bit	XT1
	PB1	1	I/O	Open-Drain	Bit	XT2
	PB2	1	I/O	Push-Pull/Open-Drain	Bit	SO1/SI1
	PB3	1	I/O	Push-Pull/Open-Drain	Bit	SCK1
	PB4	1	I/O	Push-Pull/Open-Drain	Bit	SDA1
	PB5	1	I/O	Push-Pull/Open-Drain	Bit	SCL1
Port C	PC0 to PC4	5	I/O	Push-Pull	Bit	AIN11 to AIN15

3.5.1 Port 2 (P24 to P27)

Port 2 is an 4-bit general-purpose Output port. In addition to functioning as a general-purpose Output port, Port 2 also shares functions as an address bus (A20 to A23) output. All bits of the output latch P2 and function register P2FC are set to "1" by reset, and then Port 2 works as the address bus (A20 to A23) output. Switching of the general-purpose output port function and the address bus output function can be done on bit basis by function register P2FC.

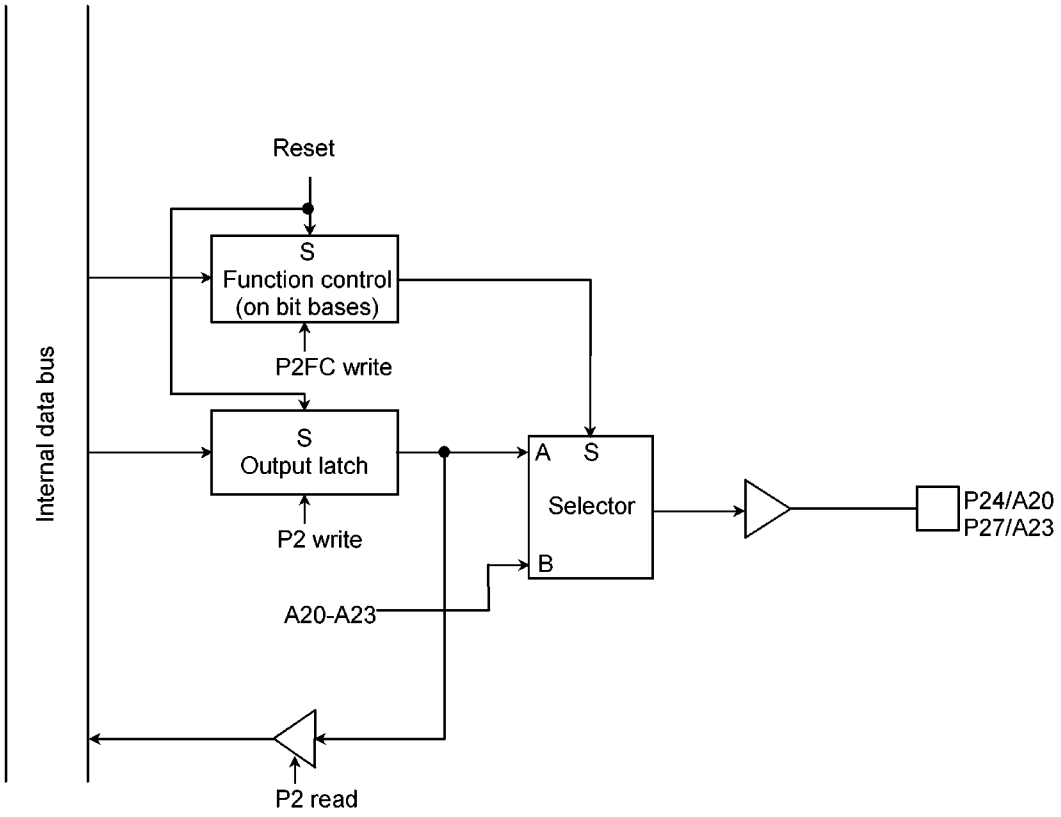


Figure 3.5.1 Port 2 (P24 to P27)

Port 2 Register

P2 (0006H)	7	6	5	4	3	2	1	0
bit Symbol	P27	P26	P25	P24				
Read/Write	R/W							
After reset	(Output latch register is set to 1)				1 (Always set to 1)			

Port 2 Function register

P2FC (0009H)	7	6	5	4	3	2	1	0
bit Symbol	P27F	P26F	P25F	P24F				
Read/Write	W							
After reset	1				1 (Alway set to 1)			
Function	0: PORT 1: ADDRESS BUS (A23 to A20)				—			

Prohibit read-modify-write

Figure 3.5.2 Registers for Port 2

3.5.2 Port 4 (P40 to P47)

Port 4 is an 8-bit input port, also used as an analog input pin (AIN3 to AIN10) for the internal A/D converter.

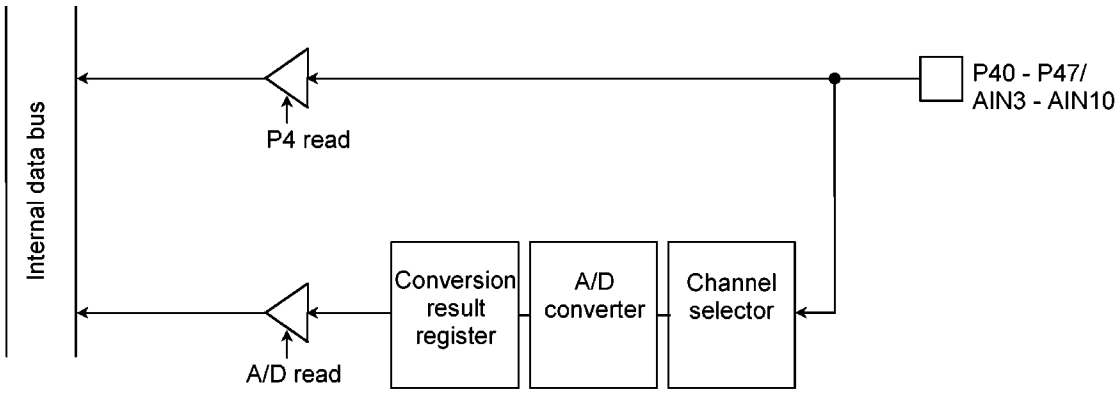


Figure 3.5.3 Port 4

Port, SRegister

		7	6	5	4	3	2	1	0
P4	bit Symbol	P47	P46	P45	P44	P43	P42	P41	P40
(000AH)	Read/Write	R							
	After reset	Input mode							

Figure 3.5.4 Register for Port 4

Note: The input channel selection of A/D converter is set by A/D converter mode register ADMOD.

3.5.3 Port 5(P50 to P57)

Port 5 is an 8-bit general-purpose I/O port. I/O can be set on bit basis using the control register P5CR. All bits of the output latch P5 are set to 1 by reset, and all bits of P5CR are cleared to 0. Therefore, Port 5 becomes the input mode port. Port 5 also shares functions as an external interrupt input pin (INT0 to INT4), an external input pin for timer counter (TI0 to TI5) and an analog input pin (AIN0 to AIN2) for A/D converter. In the output mode (P5CR<bit> = 1), the output latch data are read from the P5 register.

In the input mode (P5CR<bit> = 0), the pin data are read from P5 register.

Note: TI0, TI4 and TI5 are permitted to input when these inputs are selected by the timer counter control registers: TCR10 TCCR54. When P55/TI5/AIN0, P56/TI4/AIN1 and P57/TI0/AIN2 are used as an analog input (AIN0, AIN1 and AIN2), TI5, TI4 and TI0 for timer counter must be disabled by TCCR54 and TCCR10. (Select another clock source for timer counter.)

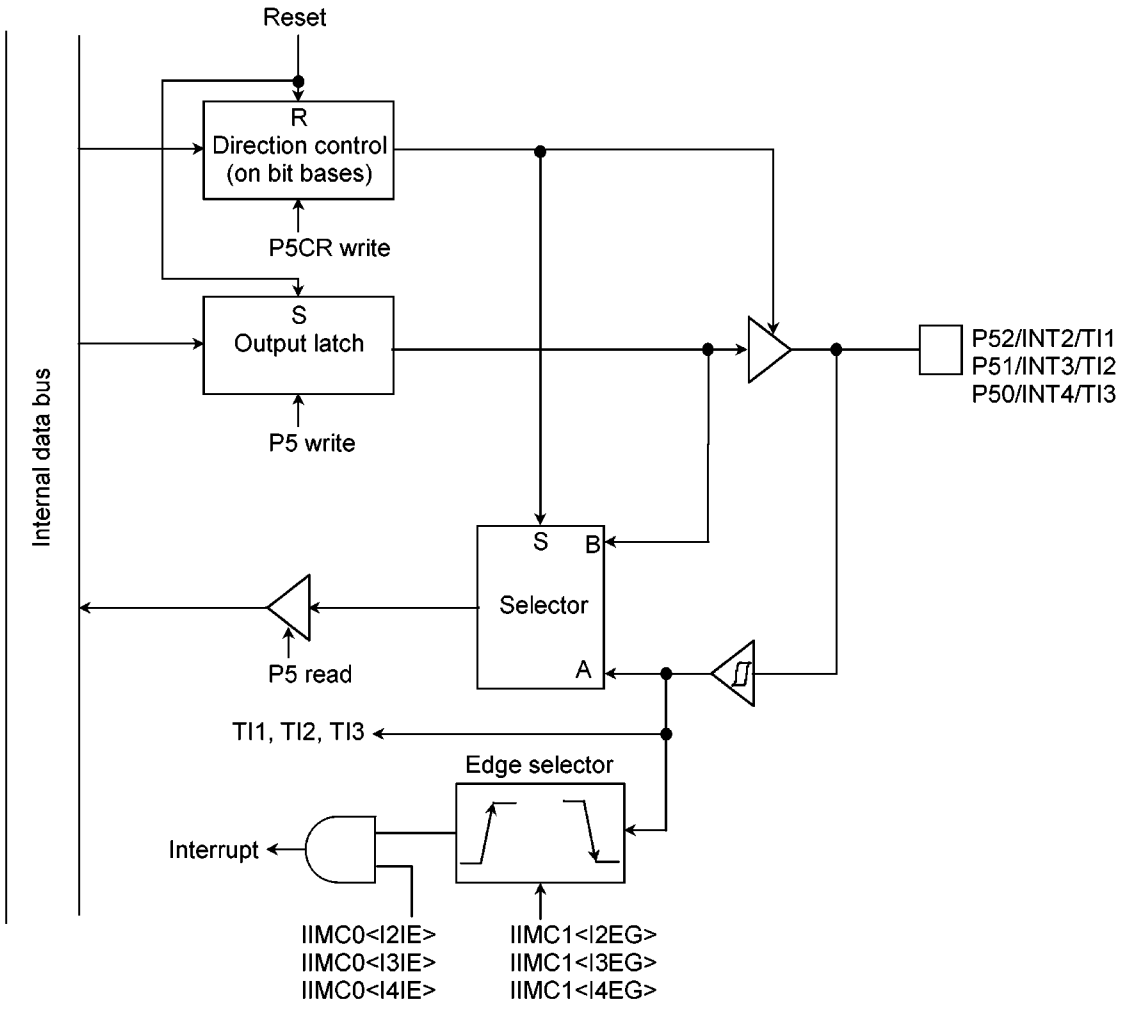


Figure 3.5.5 Port ,T (P50, P51, P52)

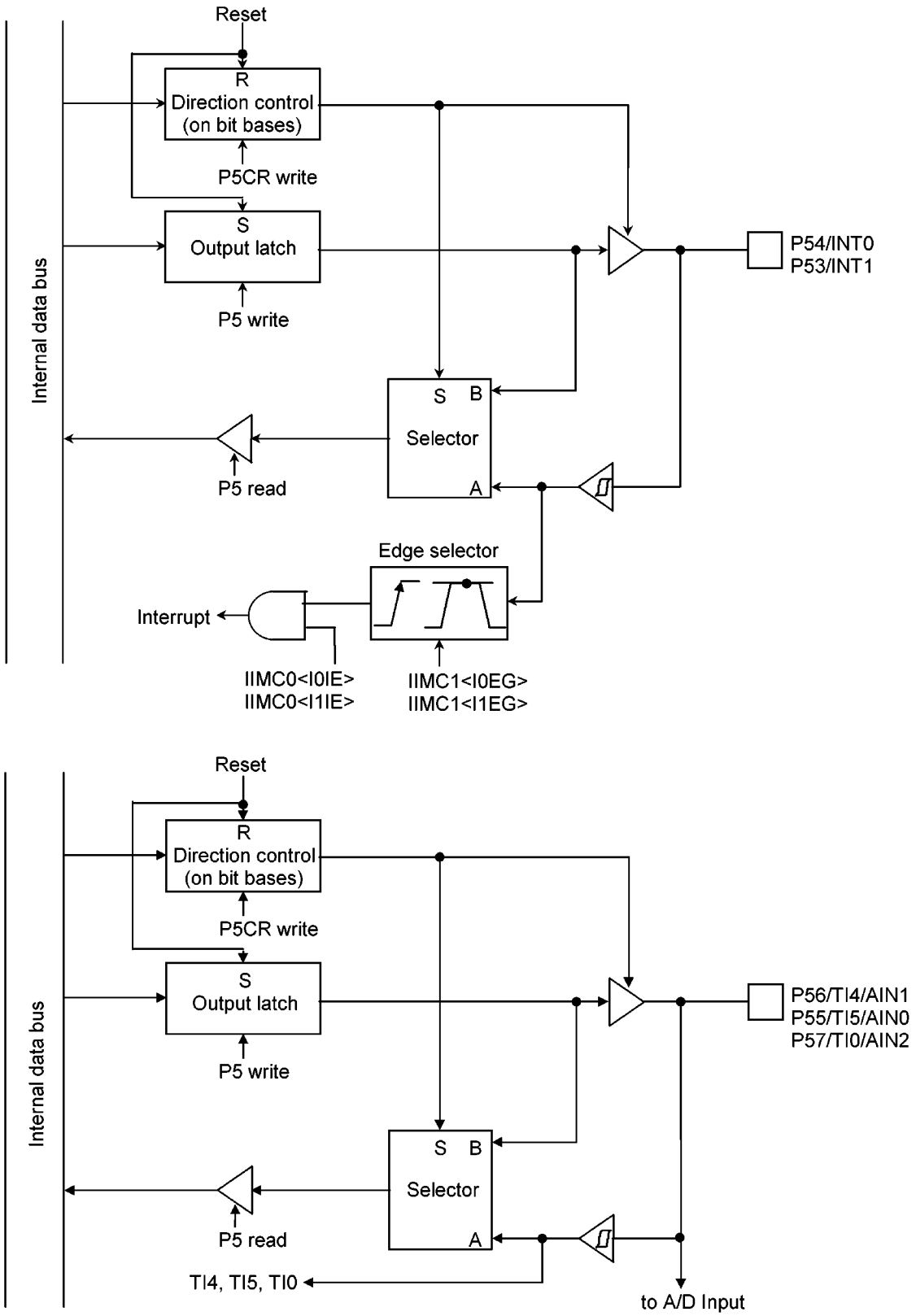


Figure 3.5.6 Port 5 (P53, P54, P56, P57)

Port 5 Register

P5 (000BH)	7	6	5	4	3	2	1	0
bit Symbol	P57	P56	P55	P54	P53	P52	P51	P50
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Port 5 Control Register

P5CR (000CH)	7	6	5	4	3	2	1	0
bit Symbol	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input mode 1: Output mode							

Prohibit read-modify-write

Figure 3.5.7 Registers for Port 5

3.5.4 Port 6 (P60 to P67)

Port 6 is an 8-bit general-purpoe I/O port. I/O can be set on bit basis using the control register P6CR.

In the output mode, switching of the push-pull-output mode and the open-drain-output mode can be done on bit basis by open drain output control register ODCR0.

Port 6 becames the input mode by reset.

(1) P60/PWM4/$\overline{CS0}$, P61/PWM5/$\overline{CS1}$, P62/PWM6/$\overline{CS2}$

P60, P61 and P62 also share functions as an 8-bit PWM output (PWM4 to PWM6) and a chip select output.

Switching of the output port function and the PWM output function can be done by the function register P6FC.

In addition, switching of the chip select output function and the function selected by P6FC can be done by chip select/wait control register bits B0CS<CS0EN>, B1CS<CS1EN> and B2CS<CS2EN>.

In the output mode (P6CR<bit> = 1), the output data selected by B0CS<CS0EN> to B2CS<CS2EN> are read from the P6 register.

In the input mode P6CR<bit> = 0), the pin data are read from P6 register.

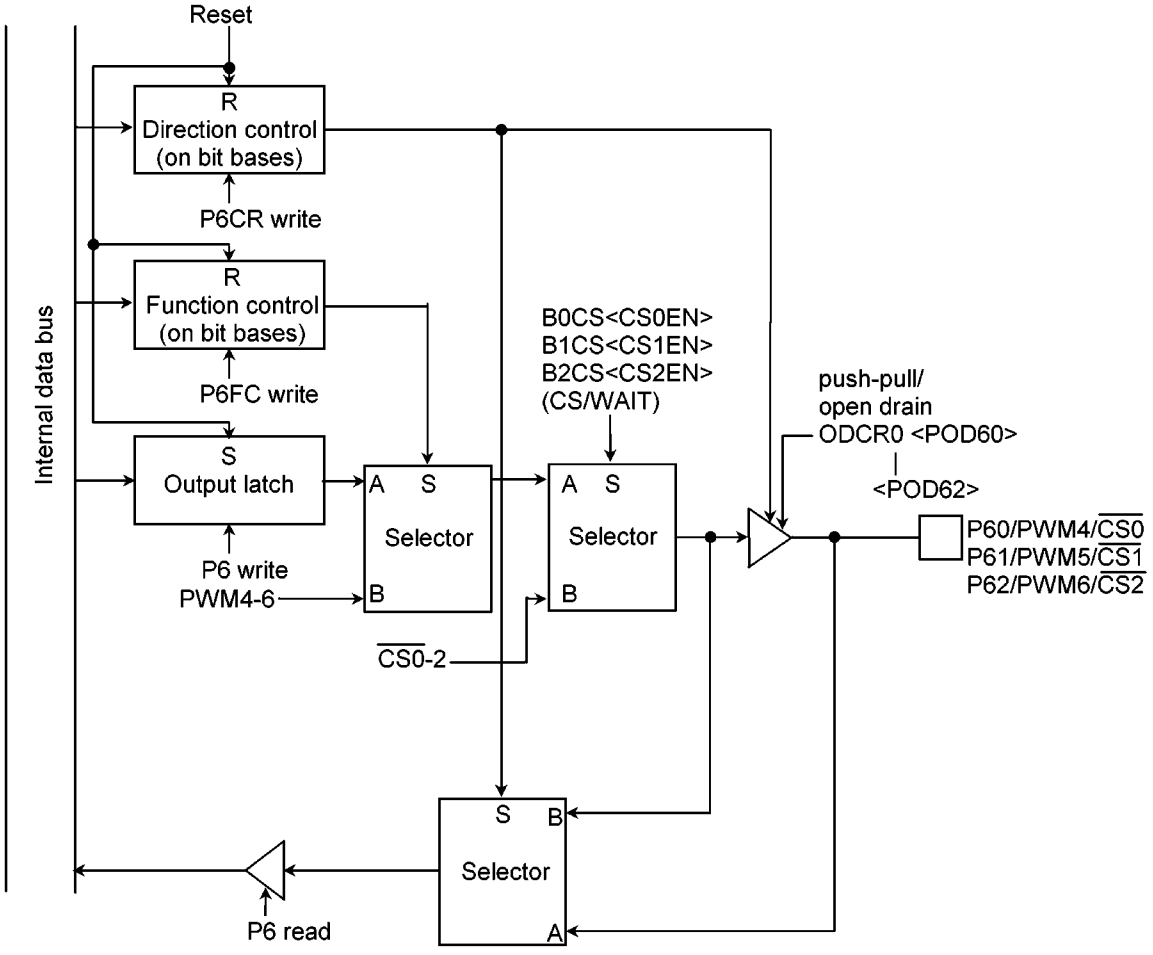


Figure 3.5.8 Port 6 (P60, P61, P62)

(2) P63/PWM7,P64/PWM8,P65/PWM9,P66/PWM10,P67/PWM11

P63 to P67 also share functions as an 8-bit PWM output (PWM7 to PWM11).

Switching of the output port function and the PWM output function can be done by the function register P6FC.

In the output mode (P6CR<bit> = 1), the output latch data are read from the P6 register.

In the input mode (P6CR<bit> = 0), the pin data are read from P6 register.

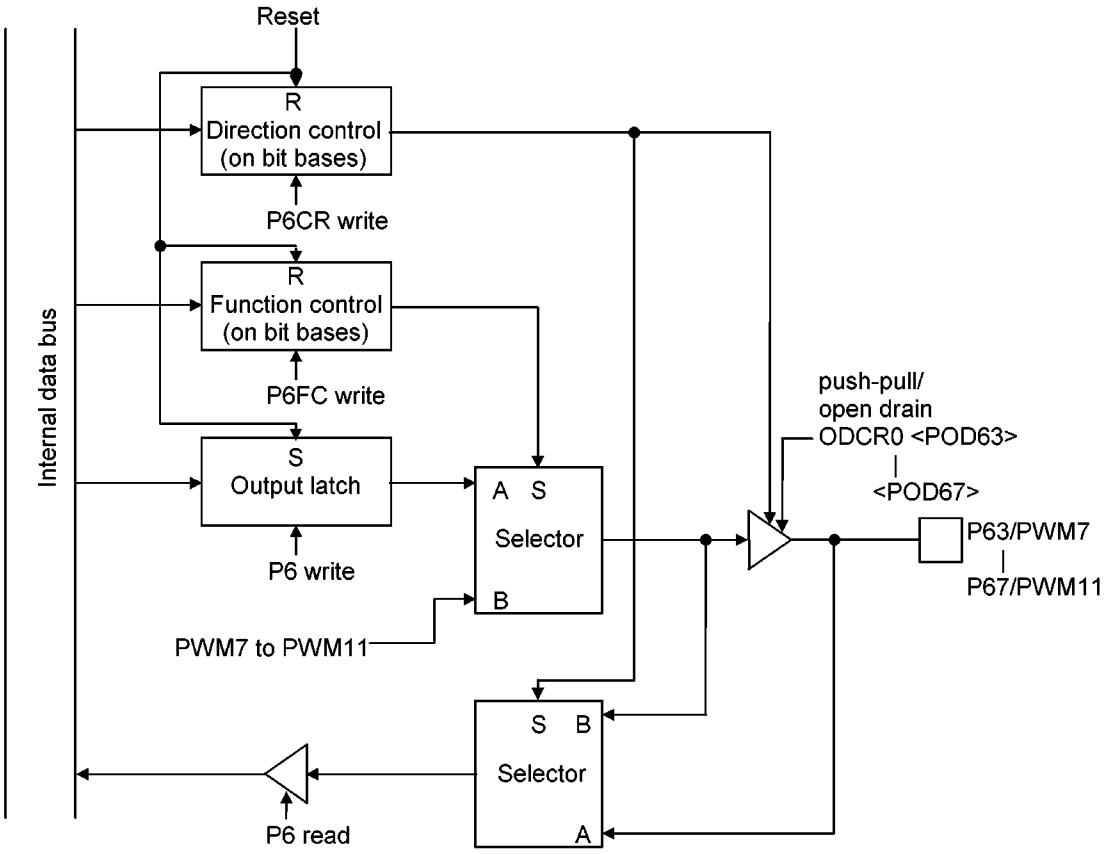


Figure 3.5.9 Port 6 (P63, P64, P65, P66, P67)

Port 6 Register

		7	6	5	4	3	2	1	0
P6	bit Symbol	P67	P66	P65	P64	P63	P62	P61	P60
(000EH)	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Port 6 Control Register

		7	6	5	4	3	2	1	0
P6CR	bit Symbol	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
(0010H)	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input mode 1: Output mode							

Prohibit read-modify-write

Port 6 Function Register

P6FC (0012H)

	7	6	5	4	3	2	1	0
bit Symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT67 1: PWM11	0: PORT66 1 :PWM10	0: PORT65 1: PWM9	0: PORT64 1: PWM8	0: PORT63 1: PWM7	B2CS <CS2EN> = 0 0: PORT62 1: PWM6	B1CS <CS1EN> = 0 0: PORT61 1: PWM5	B0CS <CS0EN> = 0 0: PORT60 1: PWM4

Prohibit read-modify-write

Open Drain Output Control Register 0

ODCR0 (0014H)

	7	6	5	4	3	2	1	0
bit Symbol	POD67	POD66	POD65	POD64	POD63	POD62	POD61	POD60
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Push-Pull 1: Open Drain							

Prohibit read-modify-write

Chip select/Wait Control Register 0

B0CS (0068H)

	7	6	5	4	3	2	1	0
bit Symbol		CS0EN		B0BUS	B0W1	B0W0	B0C1	B0C0
Read/Write		W		W	W		W	
After reset		0		0	0		0	
Function		0: Port60 /PWM4 (P6FC <P60F>) 1: $\overline{CS0}$		BUS Width Control 0: 16-bit Bus 1: 8-bit Bus	Wait Control 00: 0 WAIT 01: 1 WAIT 10: 1 WAIT 11: 2 WAIT		00: 2090H to 3FFFFFH 01: 400000H to 7FFFFF 10: 800000H to BFFFFF 11: C00000H to FFFFFF	

Prohibit read-modify-write

Chip select/Wait Control Register 1

B1CS (0069H)

	7	6	5	4	3	2	1	0
bit Symbol		CS1EN		B1BUS	B1W1	B1W0	B1C1	B1C0
Read/Write		W		W	W		W	
After reset		0		0	0		0	
Function		0: Port61 /PWM5 (P6FC <P61F>) 1: $\overline{CS1}$		BUS Width Control 0: 16-bit Bus 1: 8-bit Bus	Wait Control 00: 0 WAIT 01: 1 WAIT 10: 1 WAIT 11: 2 WAIT		00: 2090H to 3FFFFFH 01: 400000H to 7FFFFF 10: 800000H to BFFFFF 11: C00000H to FFFFFF	

Prohibit read-modify-write

Chip select/Wait Control Register 2

B2CS (0070H)	7	6	5	4	3	2	1	0
bit Symbol		CS2EN		B2BUS	B2W1	B2W0	B2C1	B2C0
Read/Write		W		W	W		W	
After reset		0		0	0		0	
Function		0: Port62 /PWM6 (P6FC <P62F>) 1: $\overline{CS2}$		BUS Width Control 0: 16-bit Bus 1: 8-bit Bus	Wait Control 00: 0 WAIT 01: 1 WAIT 10: 1 WAIT 11: 2 WAIT		00: 2090H to 3FFFFFH 01: 400000H to 7FFFFF 10: 800000H to BFFFFF 11: C00000H to FFFFFF	

Prohibit read-modify-write

Figure 3.5.10 Registers for Port 6

3.5.5 Port 7 (P70 to P77)

Port 7 is an 8-bit general-purpoe I/O port. I/O can be set on bit basis using the control register P7CR.

Port 7 also shares functions as a I^2C-bus I/F (I^2CBUS), a synchronous-SIO (SIO0) and an asynchronous serial I/F (UART) . Port 7 becames the input mode by reset.

(1) P73/SDA0, P74/SCL0

P73 and P74 are also used for a SDA bus line 0 (SDA0) and a SCL bus line 0 (SCL0) of the I^2CBUS block.

Switching of the general-purpose I/O port function and the I^2CBUS function can be done by the function register P7FC. In the output mode, switching of the push-pull-output mode and the open-drain-output mode can be done on bit basis by open drain output control register ODCR1. When these pin are used for the I2CBUS function, firstly, set P7FC<P73F> and <P74F> to 1, secondly, set ODCR1<POD73> and <POD74> to 1, finally, set P7CR<P73> and <P74C> to 1.

By Read data selection register bits PRDSEL<PR73> and <PR74>, the data read from P7 register can be chosen one of which the output data selected by P7FC or the pin data.

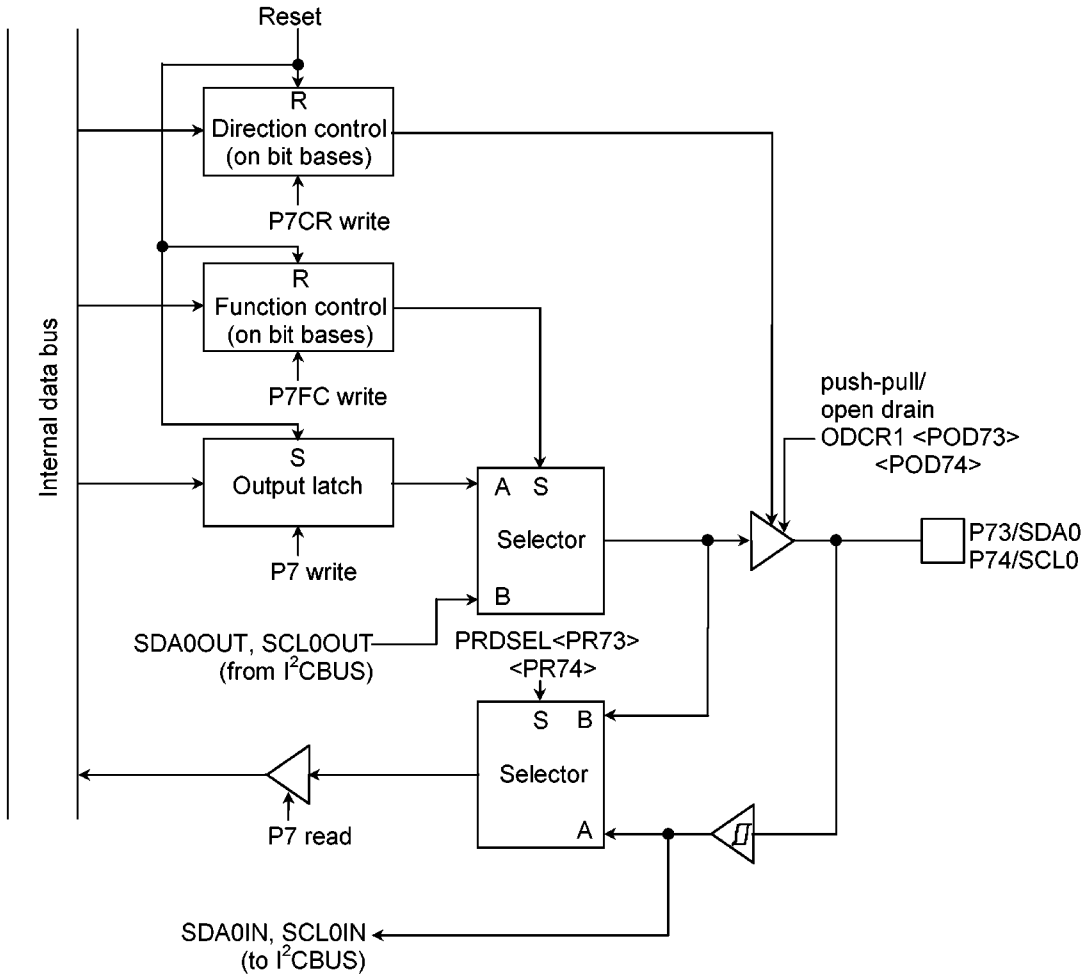


Figure 3.5.11 Port 7 (P73, P74)

(2) P70/TXD, P75/SO0, P77/SCK0

P70, P75 and P77 are also used as a transmission data output (TXD) for the UART block, a transmission data output (SO0) for the SIO0 block and a transfer clock input/output (SCK0) for the SIO0 block.

Switching of the general-purpose I/O port function and the UART/SIO0 function can be done by the function register P7FC. In the output mode, switching of the push-pull-output mode and the open-drain-output mode can be done on bit basis by open drain output control register ODCR1.

By Read data selection register PRDSEL, the data read from P7 register can be chosen one of which the output data selected by P7FC or the pin data.

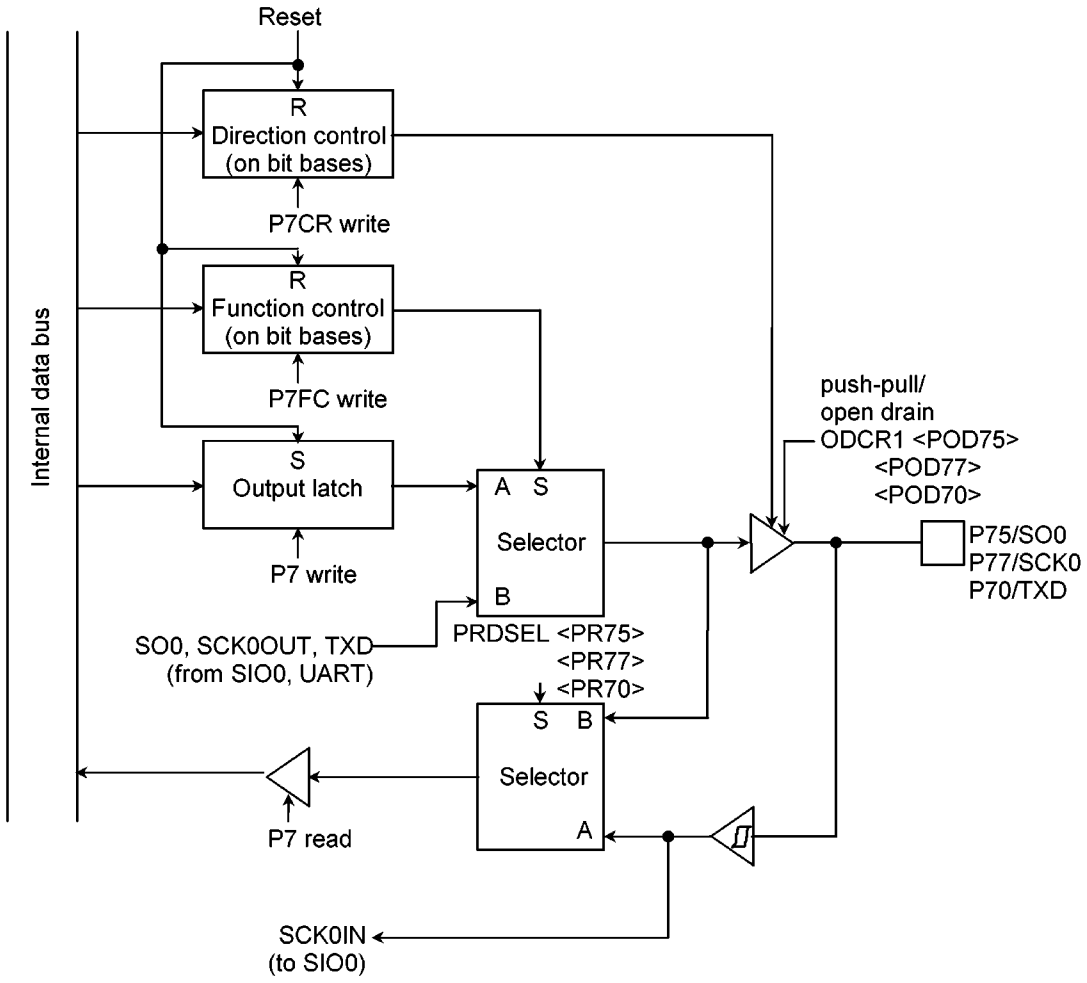


Figure 3.5.12 Port 7 (P70, P75, P77)

(3) P71/RXD, P72/CTS, $\overline{P76}$/SI0

P71, P72 and P76 are also used as a reception data input (RXD) and a $\overline{CTS}$ input for the UART block and a reception data input(SI0) for the SIO0 block.
In the output mode (P7CR<bit> = 1), the output latch data are read from the P7 register.
In the input mode (P7CR<bit> = 0), the pin data are read from P7 register.

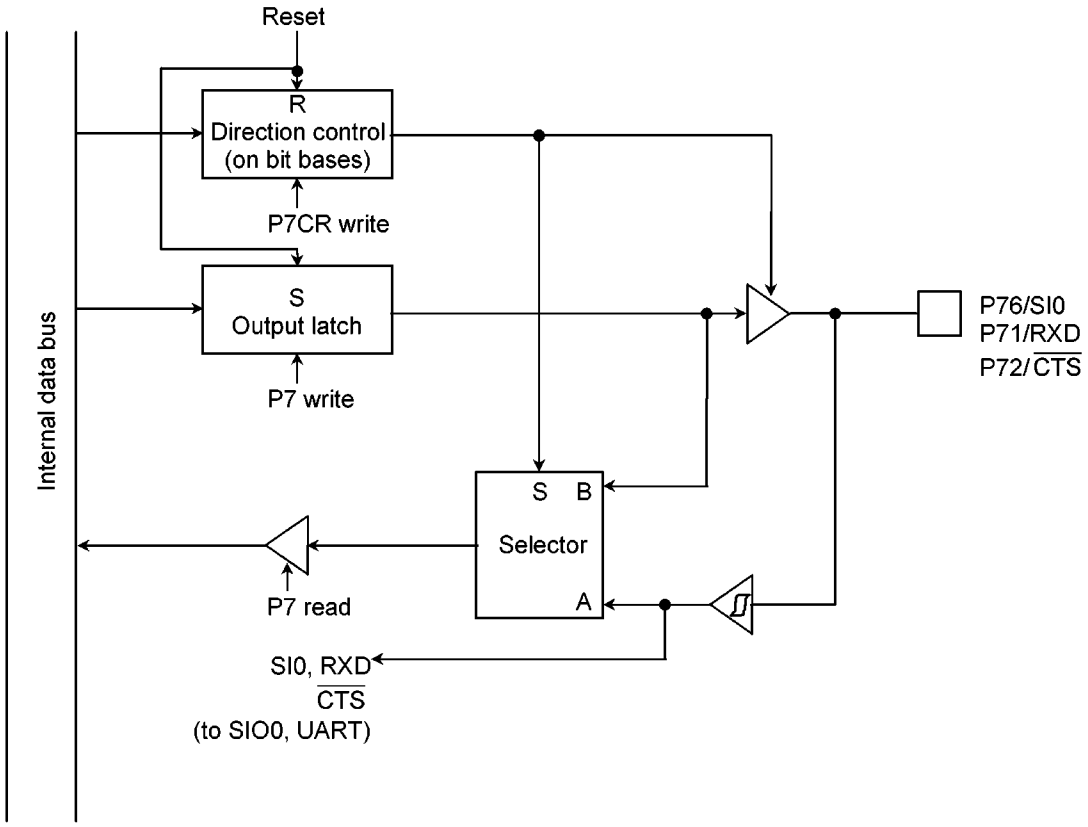


Figure 3.5.13 Port 7 (P71, P72, P76)

Port 7 Register

P7 (000FH)		7	6	5	4	3	2	1	0
	bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Port 7 Control Register

P7CR (0011H)		7	6	5	4	3	2	1	0
	bit Symbol	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input mode 1: Output mode							

Prohibit read-modify-write

Port 7 Function Register

P7FC (0013H)

	7	6	5	4	3	2	1	0
bit Symbol	P77F		P75F	P74F	P73F			P70F
Read/Write	W		W					W
After reset	0		0	0	0			0
Function	0: PORT77 1: SCK0		0: PORT75 1: SO0	0: PORT74 1: SCL0	0: PORT73 1: SDA0			0: PORT70 1: TXD

Prohibit read-modify-write

Open Drain Output Control Register 1

ODCR1 (0015H)

	7	6	5	4	3	2	1	0
bit Symbol	POD77		POD75	POD74	POD73			POD70
Read/Write	W		W					W
After reset	0		0	0	0			0
Function	0: Push-Pull 1: Open-Drain		0: Push-Pull 1: Open-Drain					0: Push-Pull 1: Open-Drain

Prohibit read-modify-write

Port Read Data Selection Register

PRDSEI (000DH)

	7	6	5	4	3	2	1	0
bit Symbol	PR77		PR75	PR74	PR73			PR70
Read/Write	W		W					W
After reset	0		0	0	0			0
Function	0: Pin Value 1: Output Value		0: Pin value 1: Output Value					0: Pin value 1: Output Value

Prohibit read-modify-write

Figure 3.5.14 Registers for Port 7

3.5.6 Port 8 (P80 to P87)

Port 8 is an 8-bit general-purpose I/O port. I/O can be set on bit basis using the control register P8CR. All bits of the output latch P8 are set to "1" by reset, and all bits of P8CR are cleared to "0". Therefore, Port 8 becames the input mode port. Port 8 also shares functions as a capture input pin (CTLIN, DFGIN, RMTIN, EXT, DPGIN, CFGIN, CSYNCIN) and a comparator signal input pin (COMPIN) for controlling an output of Head Amp Switch/Color Rotary.

In the output mode (P8CR<bit> = 1), the output latch data are read from the P8 register.

In the input mode (P8CR<bit> = 0), the pin data are read from P8 register.

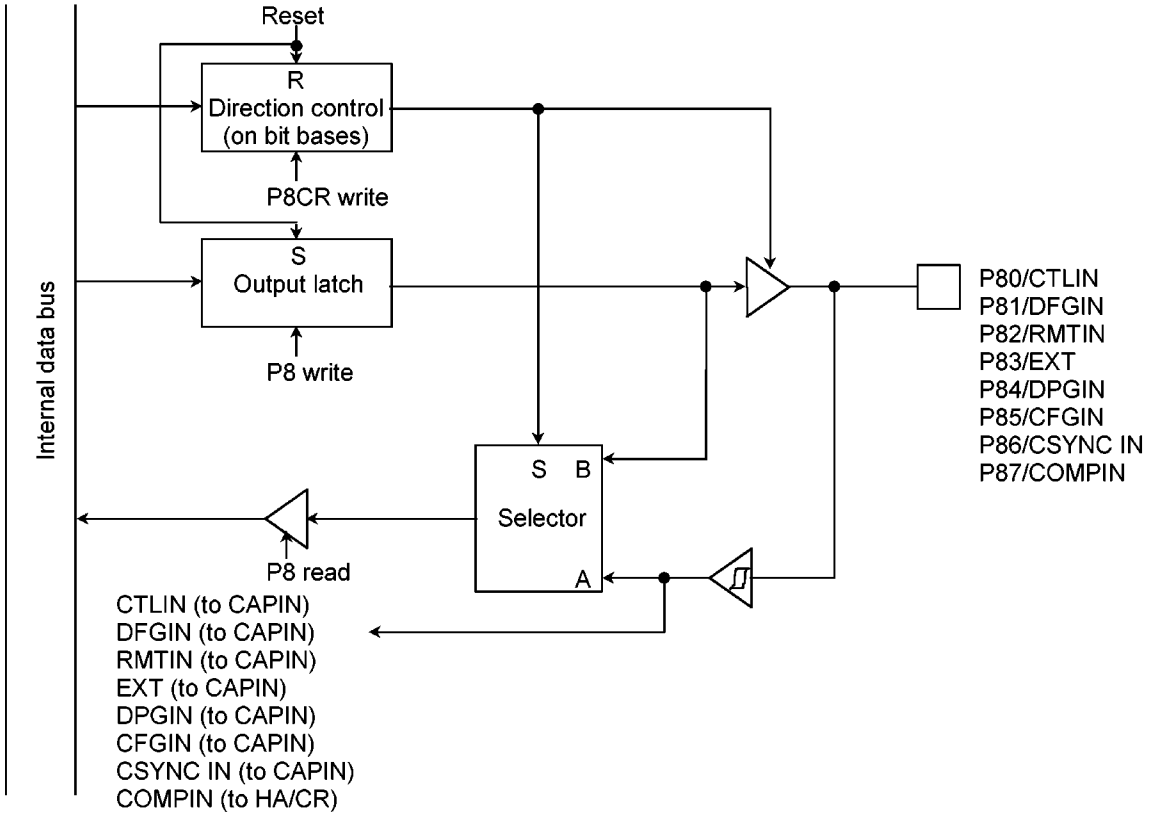


Figure 3.5.15 Port 8 (P80 to P87)

Port 8 Register

P8 (0016H)	7	6	5	4	3	2	1	0
bit Symbol	P87	P86	P85	P84	P83	P82	P81	P80
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Port 8 Control Register

P8CR (0018H)	7	6	5	4	3	2	1	0
bit Symbol	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0:Input mode 1:Output mode							

Prohibit read-modify-write

Figure 3.5.16 Registers for Port 8

3.5.7 Port 9 (P90 to P97)

Port 9 is an 8-bit general-purpoe I/O port. I/O can be set on bit basis using the control register P9CR.

In the output mode, switching of the push-pull-output mode and the open-drain-output mode can be done on bit

basis by open drain output control register ODCR2.

Port 9 becames the input mode by reset.

(1) P90/TPG12, P91/TPG01 to P94/TPG04, P95/TPG13

P90 and P95 are also used as output pins of TPG12 and TPG13 for the timing pulse generator 1 (TPG1) block. P91 to P94 are also used as output pins of TPG01 to TPG04 for the timing pulse generator 0 (TPG0) block.Switching of the output port function and the TPG output function can be done by the function register P9FC.

In the output mode (P9CR<bit> = 1), the output latch data are read from the P9 register.

In the input mode (P9CR<bit> = 0), the pin data are read from P9 register.

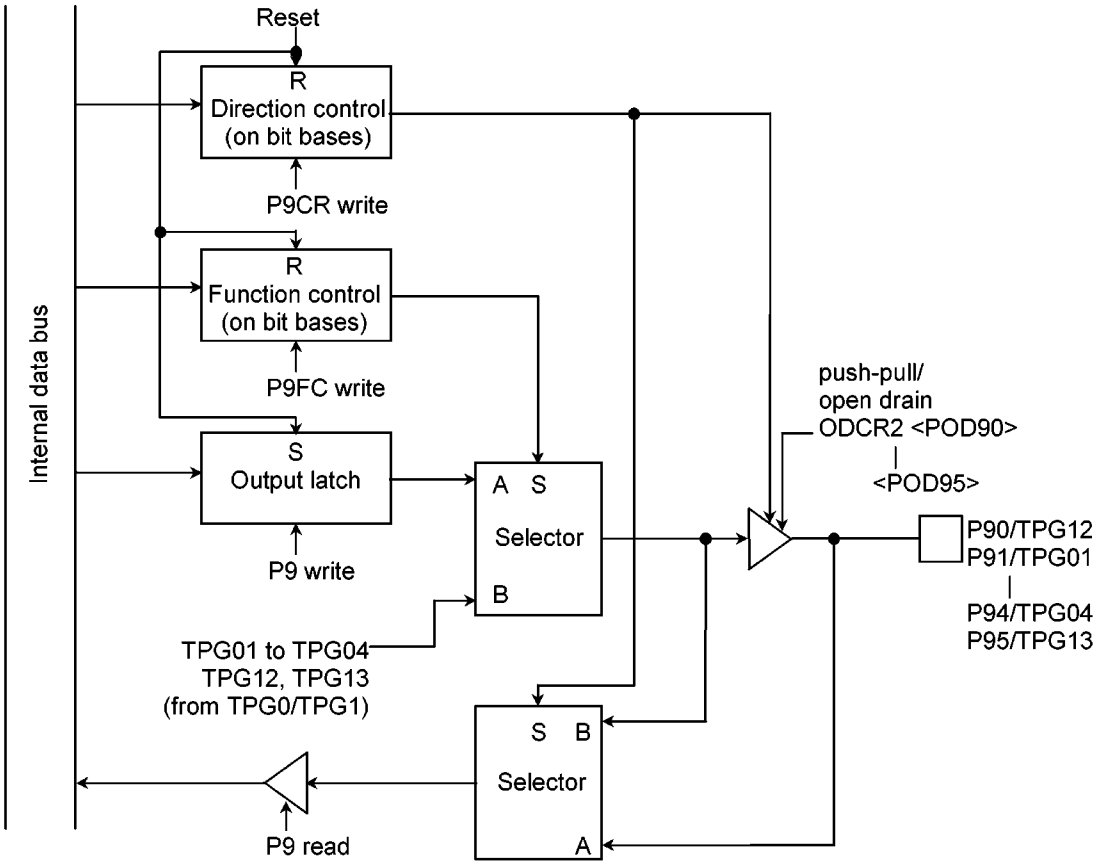


Figure 3.5.17 Port 9 (P90 to P95)

(2) P96/TPG10, P97/TPG11

P96 and P97 are also used as output pins of TPG10 and TPG11 for the timing pulse generator 1(TPG1) block. Switching of the output port function and the TPG output function can be done by the function register P9FC.

In the output mode (P9CR<bit> = 1), the output latch data are read from the P9 register.

In the input mode (P9CR<bit> = 0), the pin data are read from P9 register.

In addition to output from P96, the TPG10 also controls 3-state-output of P97. When P9CR<P97C> is set to 0, P97 is an output mode at TPG10 = 1, and is a high-impedance (Hz) output mode at TPG10 = 0.

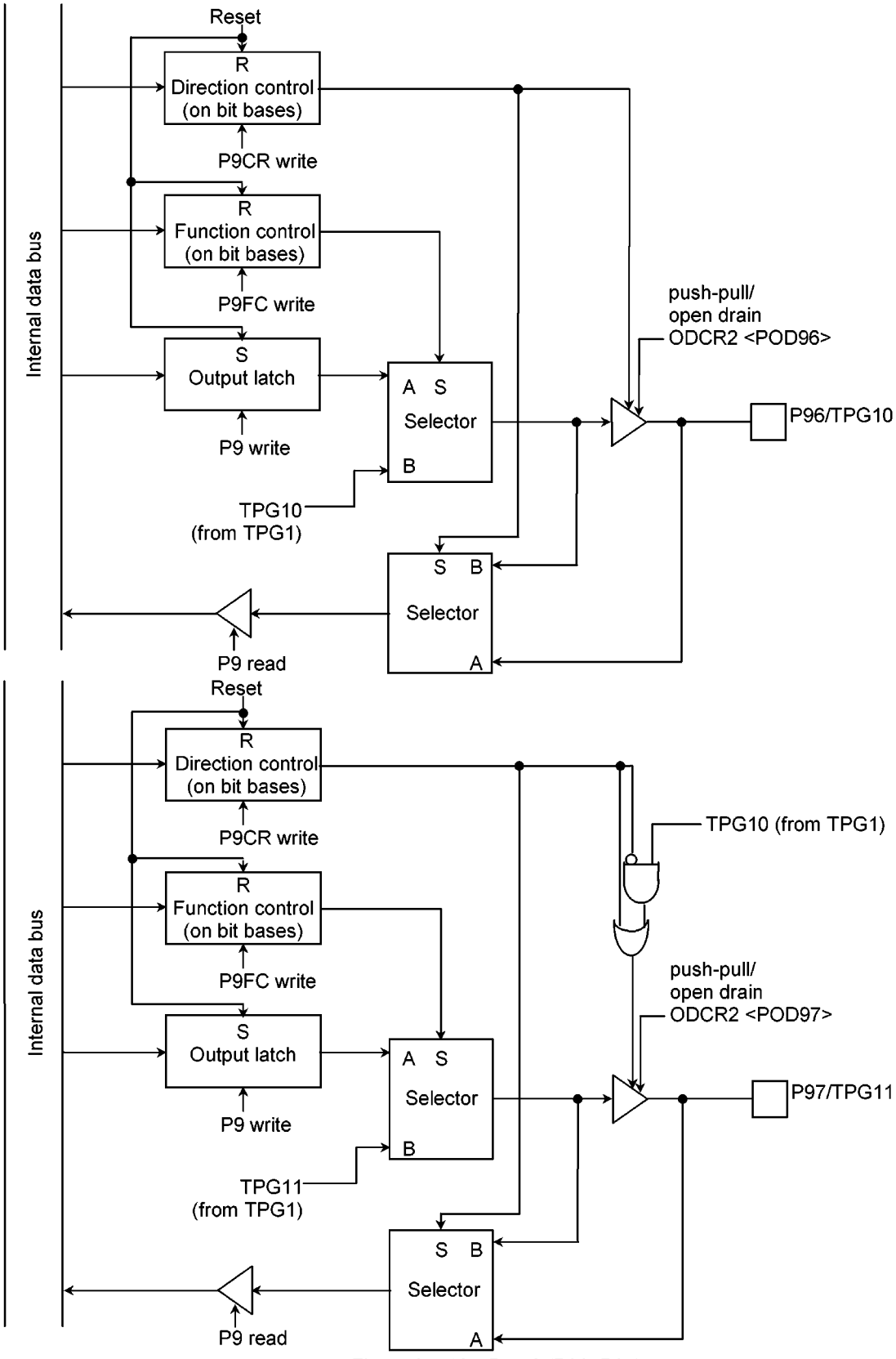


Figure 3.5.18 Port 9 (P96, P97)

Port 9 Register

P9 (0017H)

	7	6	5	4	3	2	1	0
bit Symbol	P97	P96	P95	P94	P93	P92	P91	P90
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Port 9 Control Register

P9CR (0019H)

	7	6	5	4	3	2	1	0
bit Symbol	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input mode 1: Output mode							

Prohibit read-modify-write

Port 9 Function Register

P9FC (001AH)

	7	6	5	4	3	2	1	0
bit Symbol	P97F	P96F	P95F	P94F	P93F	P92F	P91F	P90F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT97 1: TPG11	0: PORT96 1: TPG10	0: PORT95 1: TPG13	0: PORT94 1: TPG04	0: PORT93 1: TPG03	0: PORT92 1: TPG02	0: PORT91 1: TPG01	0 :PORT90 1: TPG12

Prohibit read-modify-write

Open Drain Output control Register 2

ODCR2 (001BH)

	7	6	5	4	3	2	1	0
bit Symbol	POD97	POD96	POD95	POD94	POD93	POD92	POD91	POD90
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Push-Pull 1: Open Drain							

Prohibit read-modify-write

Figure 3.5.19 Registers for Port 9

3.5.8 Port A (PA0 to PA5)

Port A is an 6-bit general-purpoe I/O port. I/O can be set on bit basis using the control register PACR.

Port A also shares functions as a Pseudo-Vsync/Pseudo-Hsync output (PV-PH), Head Amp Switch/Color Rotary output (HA (TPG05), CR (TPG00)) by TPG0 block, a 14-bit PWM output (PWM2), an 8-bit PWM output (PWM3) and write strobe output for external memory ($\overline{WR}$, $\overline{HWR}$). Port A becames the input mode by reset.

(1) PA0/PV-PH

PA0 is also used as a Pseudo-Vsync/Pseudo-Hsync output (PV-PH). Switching of the output port function and the PV-PH output function can be done by the function register PAFC.

In the output mode (PACR<bit> = 1), the output latch data are read from the PA register.

In the input mode (PACR<bit> = 0), the pin data are read from PA register.

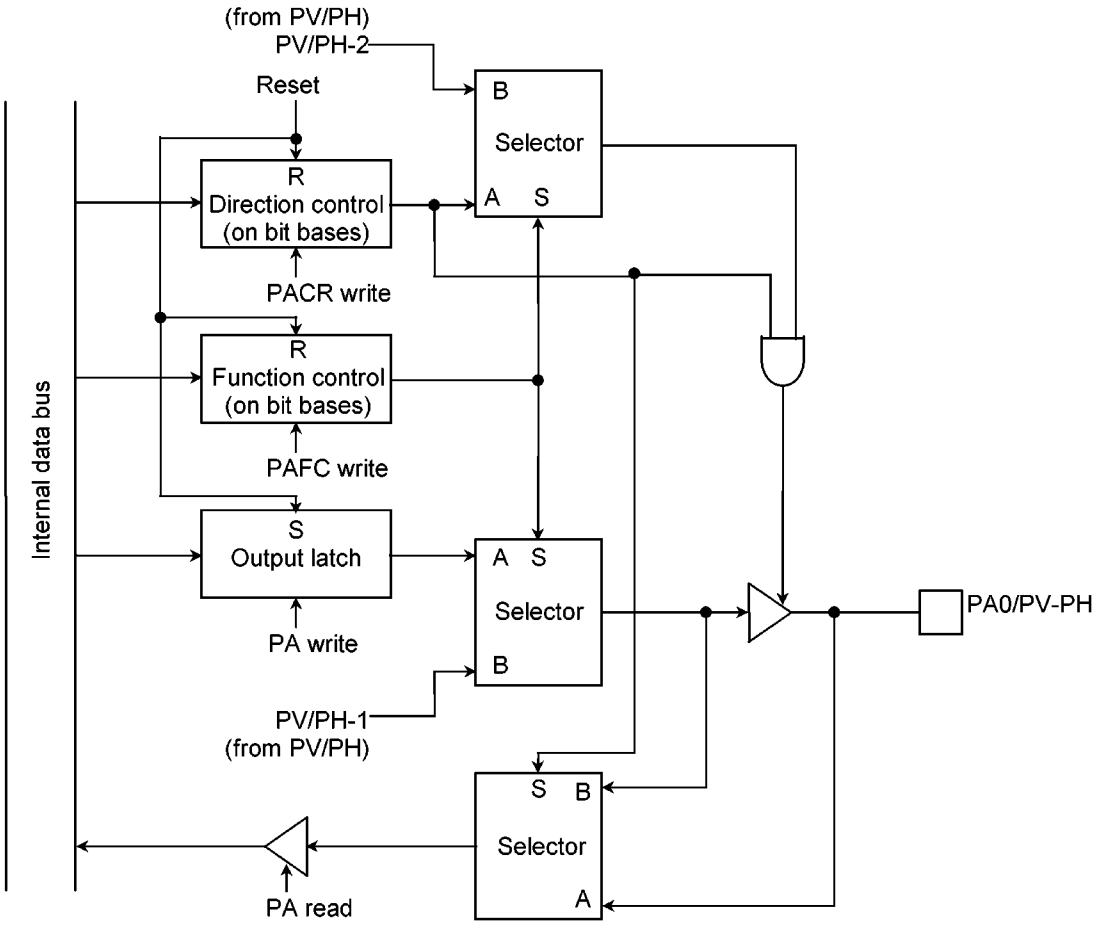


Figure 3.5.20 Port A (PA0)

(2) PA1/HA (TPG05), PA2/CR (TPG00)

PA1 and PA2 are also used as Head Amp Switch output (HA (TPG05)) and Color Rotary output (CR (TPG00)).

Switching of the output port function and the output of the HA (TPG05) and CR (TPG00) can be done by the function register PAFC.

In the output mode (PACR<bit> = 1), the output latch data are read from the PA register.

In the input mode (PACR<bit> = 0), the pin data are read from PA register.

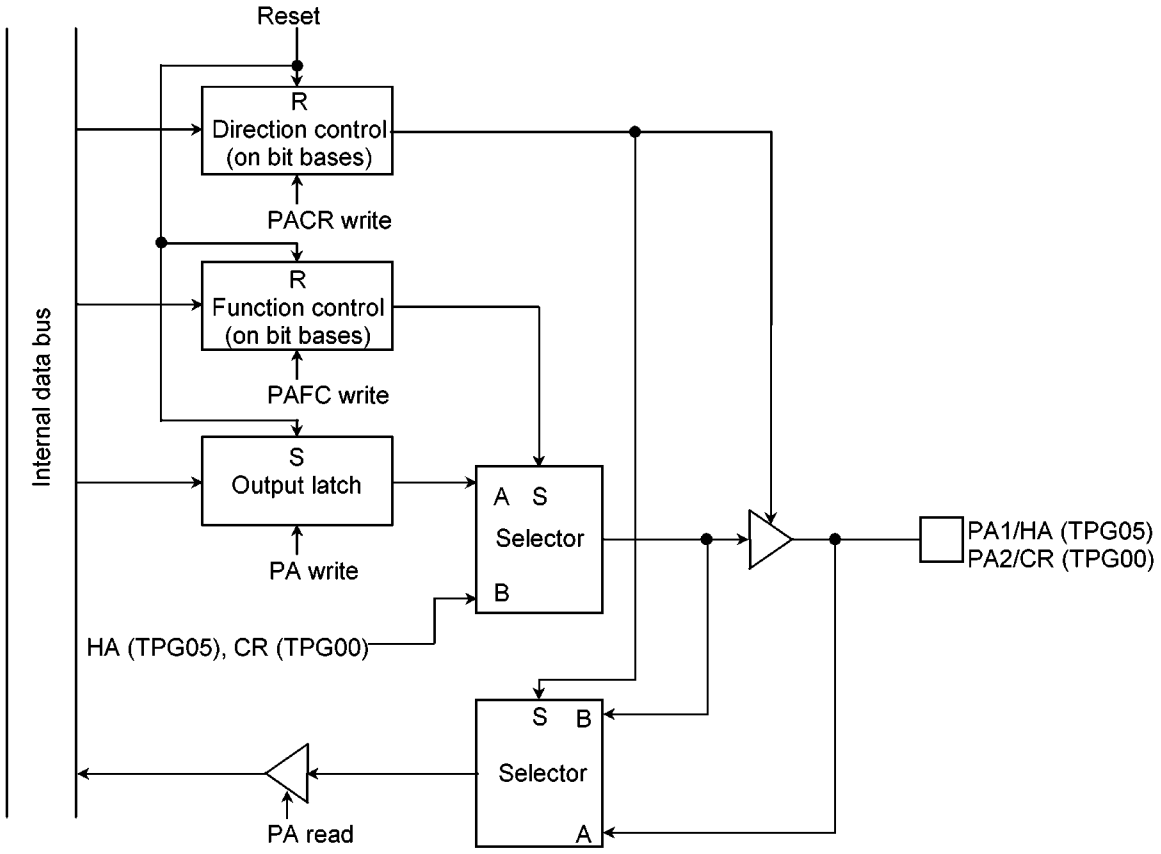


Figure 3.5.21 Port A (PA1, PA2)

(3) PA3/PWM2, PA4/$\overline{WR}$

PA3 and PA4 are also used as 14-bit PWM output (PWM2) and write strobe for lower byte of external memory ($\overline{WR}$). Switching of the output port function and the output of the PWM and $\overline{WR}$ can be done by the function register PAFC. In the output mode, switching of the push-pull-output mode and the open-drain-output mode can be done on bit basis by open drain output control register ODCR3.

In the output mode (PACR<bit> = 1), the output latch data are read from the PA register.

In the input mode (PACR<bit> = 0), the pin data are read from PA register.

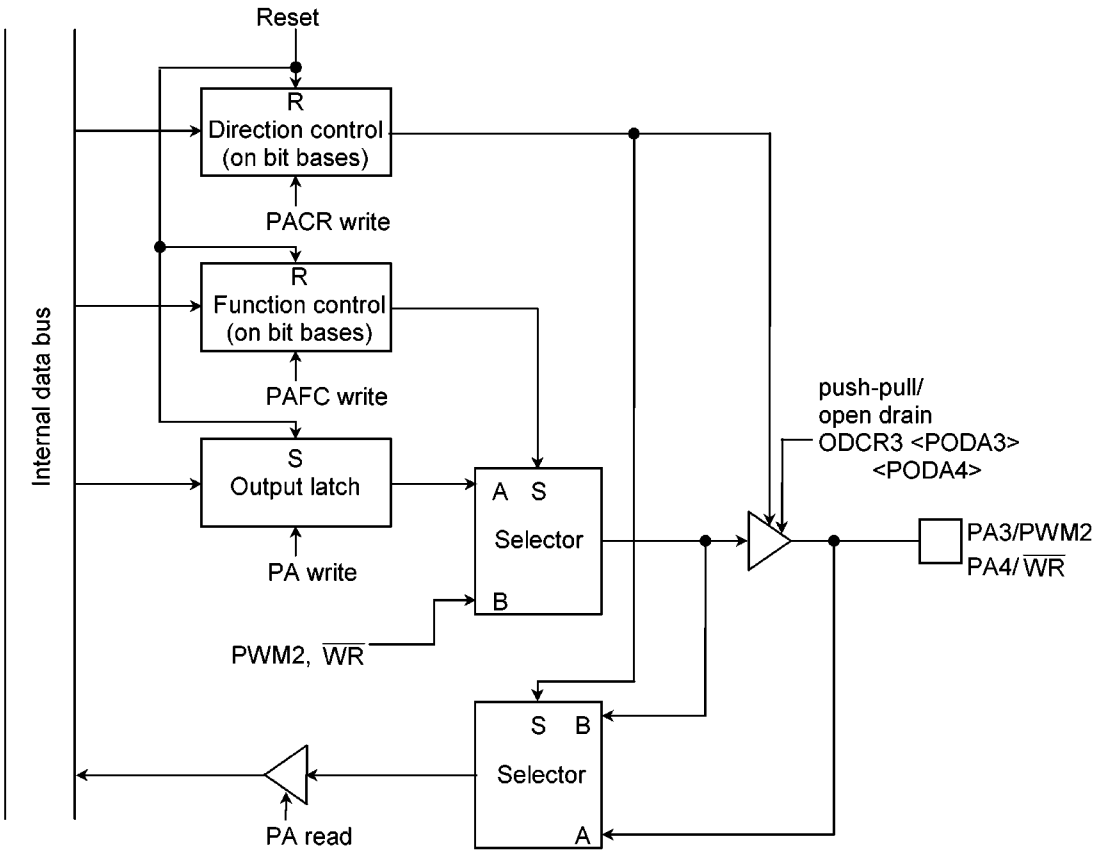


Figure 3.5.22 Port A (PA3, PA4)

(4) PA5/PWM3/$\overline{HWR}$

PA5 is also used as 8-bit PWM output (PWM3) or write strobe for higher byte of external memory ($\overline{HWR}$).

Switching of the output port function and the PWM output can be done by the function register bit PAFC<PA5F0>.

In addition, Switching of the $\overline{HWR}$ output and the output selected by PAFC<PA5F0> can be done by the function register bit PAFC<PA5F1>. In the output mode, switching of the push-pull-output mode and the open-drain-output mode can be done on bit basis by open drain output control register bit ODCR3<PODA5>.

In the output mode (PACR<bit> = 1), the output data selected by PAFC is read from the PA register.

In the input mode (PACR<bit> = 0), the pin data is read from PA register.

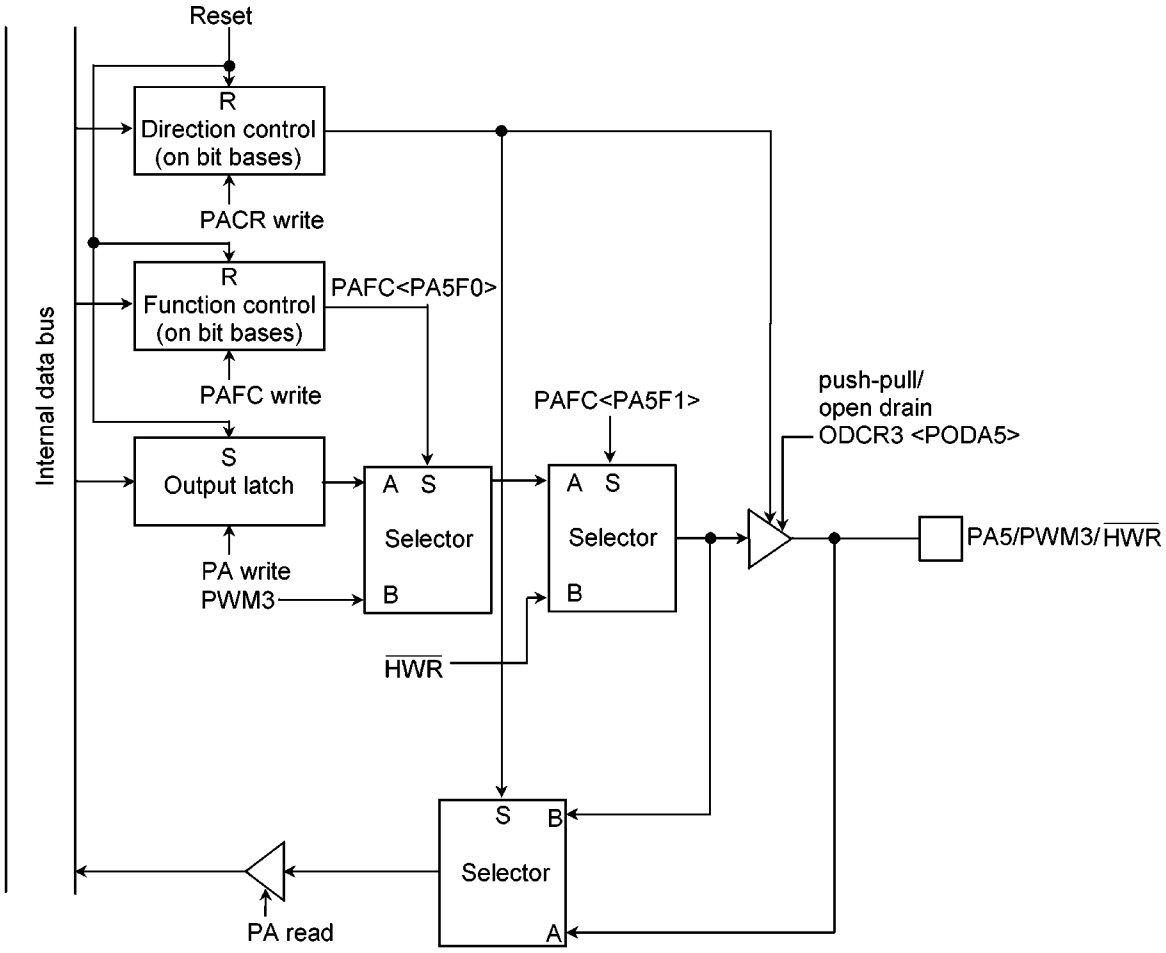


Figure 3.5.23 Port A (PA5)

Port A Register

PA (001CH)

	7	6	5	4	3	2	1	0
bit Symbol			PA5	PA4	PA3	PA2	PA1	PA0
Read/Write			R/W					
After reset			1	1	1	1	1	1

Port A Control Register

PACR (001EH)

	7	6	5	4	3	2	1	0
bit Symbol			PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
Read/Write			W					
After reset			0	0	0	0	0	0
Function			0: Input mode 1: Output mode					

Prohibit read-modify-write

Open Drain Output Control Register 3

ODCR3 (0021H)

	7	6	5	4	3	2	1	0
bit Symbol				PWMOD1 (PWM1)	PWMOD0 (PWM0)	PODA5 (PortA5)	PODA4 (PortA4)	PODA3 (PortA3)
Read/Write				W		W		
After reset				0	0	0	0	0
Function				PWM0,1 Open drain output control 0: Push-Pull 1: Open Drain		Port A Open drain output control 0: Push-Pull 1: Open drain		

Prohibit read-modify-write

Port A Function Register

PAFC (006BH)

	7	6	5	4	3	2	1	0
bit Symbol		PA5F1	PA5F0	PA4F	PA3F	PA2F	PA1F	PA0F
Read/Write		W						
After reset		0	0	0	0	0	0	0
Function		0: PORTA5 /PWM3 (<PA5F0>) 1: $\overline{HWR}$	0: PORTA5 1: PWM3	0: PORTA4 1: $\overline{WR}$	0: PORTA3 1: PWM2	0: PORTA2 1: CR (TPG00)	0: PORTA1 1: HA (TPG05)	0: PORTA0 1: PV/PH

Prohibit read-modify-write

Figure 3.5.24 Registers for Port A

3.5.9 Port B(PB0 to PB5)

Port B is an 6-bit general-purpoe I/O port. I/O can be set on bit basis using the control register PBCR.

Port B also shares functions as a low-frequency oscillator connecting pin (XT1, XT2), a transmission/reception data input/output pin (S01/SI1) and a transfer clock input/output pin (SCK1) for the synchronous SIO1 block, SDA bus line 1 (SDA1) and SCL bus line 1 (SCL1) for I^2CBUS block. Port B becames the input mode by reset.

(1) PB0/XT1, PB1/XT2

PB0 and PB1are also used as a low-frequency oscillator connecting pin (XT1, XT2). In the output mode, these pin are an open drain output. In the output mode (PBCR<bit> = 1), the output latch data are read from the PB register. In the input mode (PBCR<bit> = 0), the pin data is read from PB register.

Connecting with resonators for a low-frequency oscillator pin is necessary to set PBCR<PB0C, PB1C> to "11" and PB<PB0, PB1> to 00.

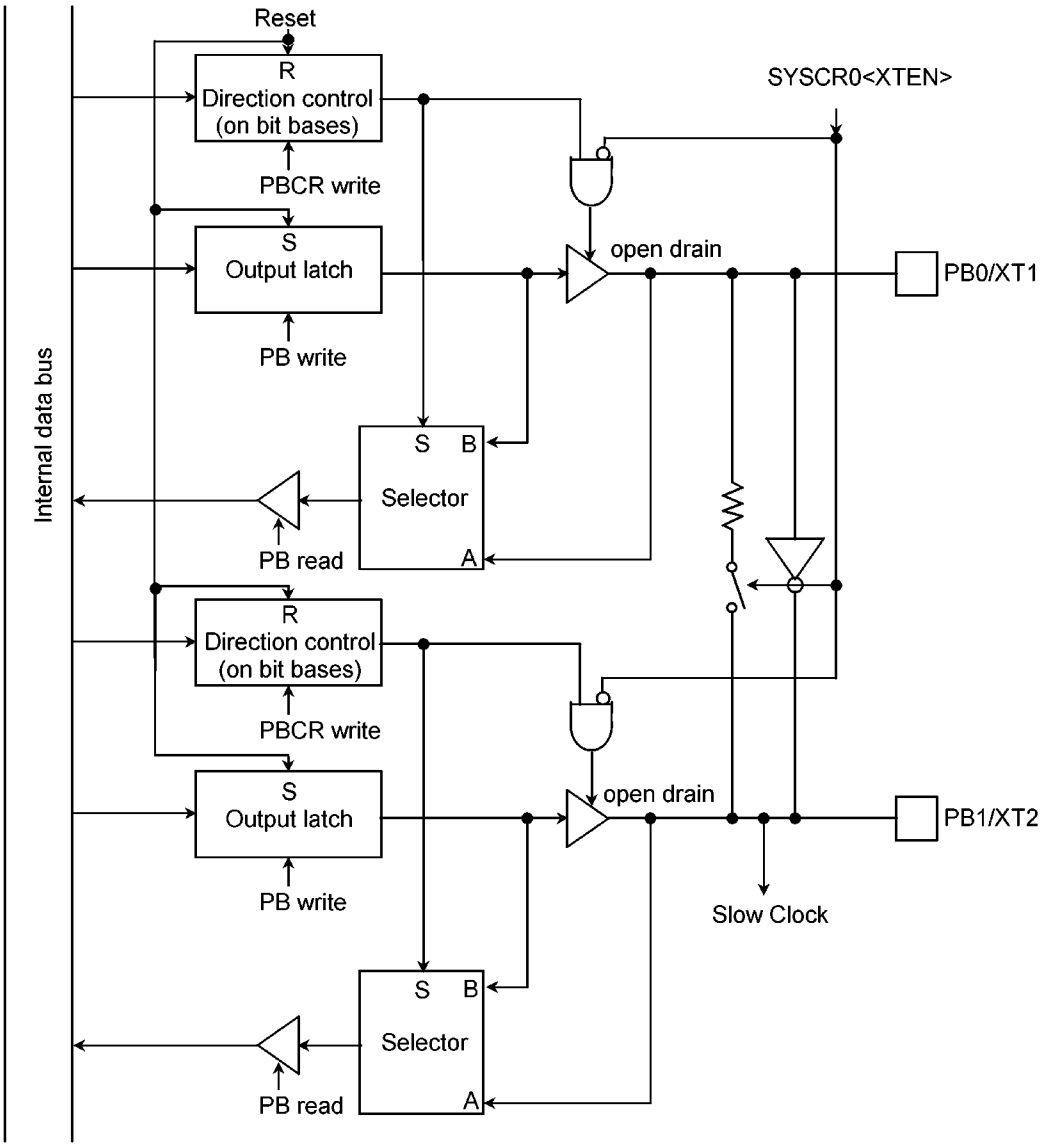


Figure 3.5.25 Port B (PB0, PB1)

(2) PB2/SO1/SI1, PB3/SCK1

PB2 and PB3 are also used as an transmission/reception data input/output pin (SO1/SI1) and a transfer clock input/output pin (SCK1) for the synchronous SIO1 block. Switching of the output port function and the SIO1 function can be done by the function register PBFC. In the output mode, switching of the push-pull-output mode and the open-drain-output mode can be done on bit basis by open drain output control register bit ODCR4<PODB2, PODB3>.

By ODCR4<PRB2, PRB3>, the data read from PB register can be chosen one of which the output data selected by PBFC or the pin data.

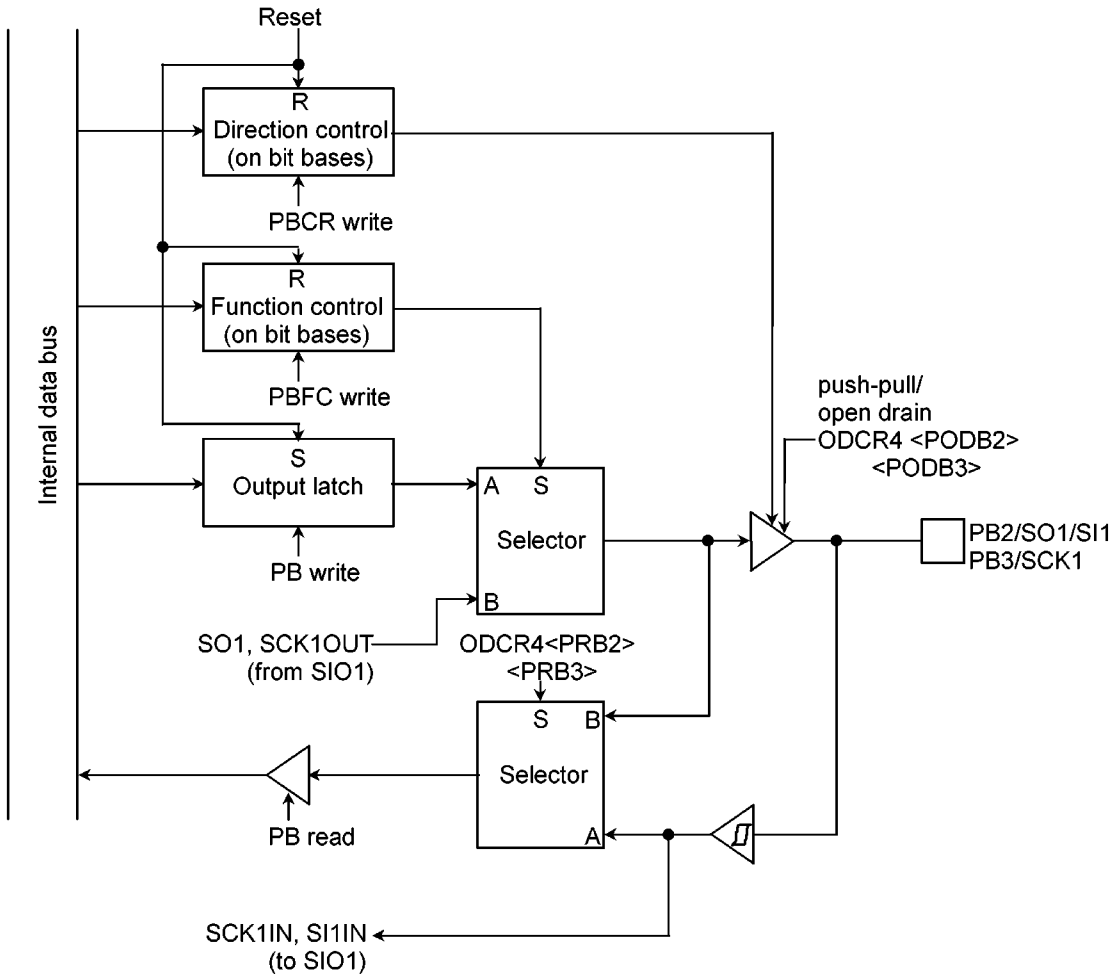


Figure 3.5.26 Port B (PB2, PB3)

(3) PB4/SDA1, PB5/SCL1

PB4 and PB5 are also used for a SDA bus line 1 (SDA1) and a SCL bus line 1 (SCL1) of the I2CBUS block.

Switching of the general-purpose I/O port function and the I^2CBUS function can be done by the function register PBFC. In the output mode, switching of the push-pull-output mode and the open-drain-output mode can be done on bit basis by open drain output control register bit ODCR4<PODB4,PODB5>. When these pin are used for the I^2CBUS function, firstly, set PBFC<PB4F> and <PB5F> to 1, secondly, set ODCR4<PODB4> and <PODB5> to 1, finally, set PBCR<PB4C> and <PB5C> to 1.

By ODCR4<PRB4, PRB5>, the data read from PB register can be chosen one of which the output data selected by PBFC or the pin data.

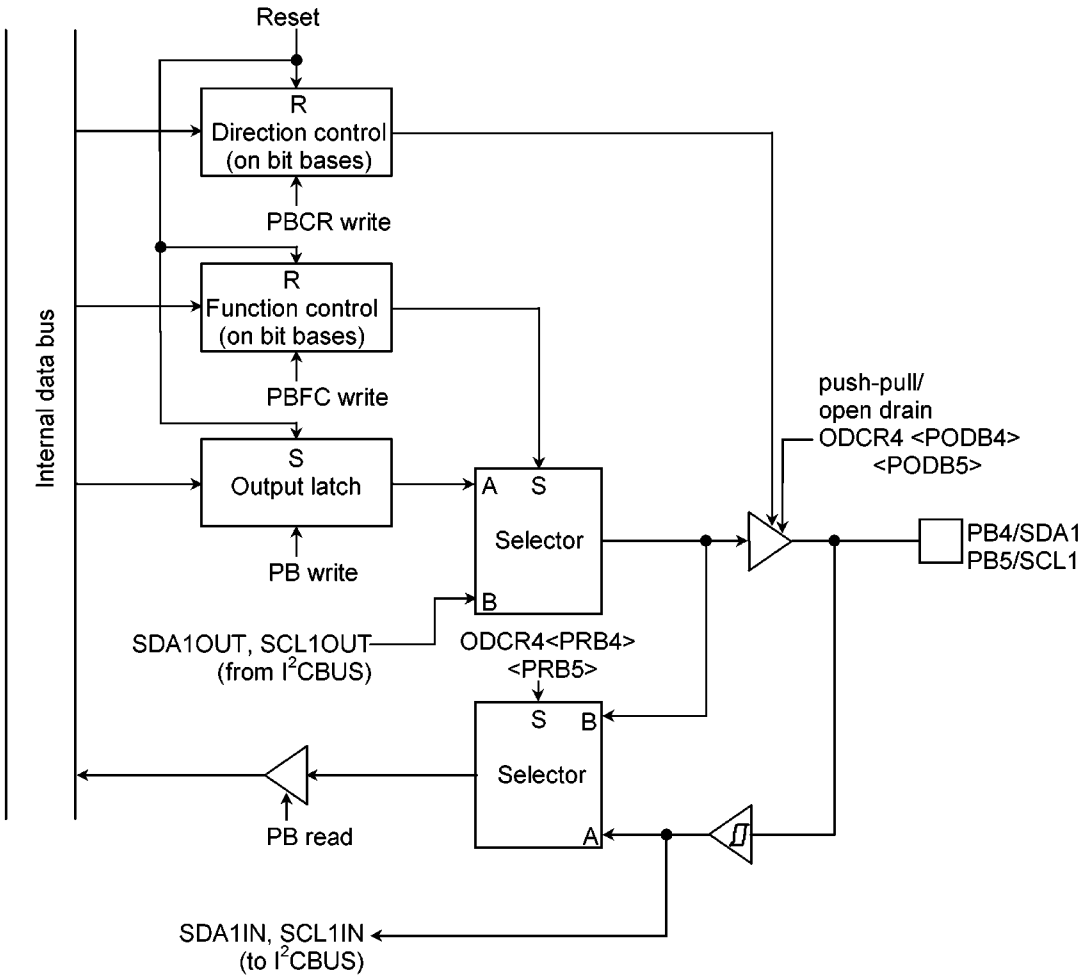


Figure 3.5.27 Port B (PB4, PB5)

Port B Register

PB (001DH)

	7	6	5	4	3	2	1	0
bit Symbol			PB5	PB4	PB3	PB2	PB1	PB0
Read/Write			R/W				R/W	
After reset			1	1	1	1	1	1
							(SYSCR0<XTEN> = 0)	

Port 8 Control Register

PBCR (001FH)

	7	6	5	4	3	2	1	0
bit Symbol			PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
Read/Write			W				W	
After reset			0	0	0	0	0	0
Function			0: Input mode 1: Output mode					

Prohibit read-modify-write

Port B Function Register

PBFC (0020H)

	7	6	5	4	3	2	1	0
bit Symbol			PB5F	PB4F	PB3F	PB2F		
Read/Write			W					
After reset			0	0	0	0		
Function			0: PORTB5 1: SCL1	0: PORTB4 1: SDA1	0: PORTB3 1: SCK1	0: PORTB2 1: SO1/SI1		

Prohibit read-modify-write

Open Drain Output Control Register 4

ODCR4 (0022H)

	7	6	5	4	3	2	1	0
bit Symbol	PRB5 (PortB5)	PRB4 (PortB4)	PRB3 (PortB3)	PRB2 (PortB2)	PODB5 (PortB5)	PODB4 (PortB4)	PODB3 (PortB3)	PODB2 (PortB2)
Read/Write	W				W			
After reset	0	0	0	0	0	0	0	0
Function	Port B Input data selection 0: Pin Value 1: Output Value				Port B Open drain output control 0: Push-Pull 1: Open Drain			

Prohibit read-modify-write

Figure 3.5.28 Registers for Port B

3.5.10 Port C (PC0 to PC4)

Port C is an 5-bit general-purpoe I/O port. I/O can be set on bit basis using the control register PCCR. All bits of the output latch PC are set to 1 by reset, and all bits of PCCR are cleared to 0. Therefore, Port c becames the input mode port. Port C also shares functions as an analog input pin (AIN11 to AIN15) for A/D converter.

In the output mode (PCCR<bit> = 1), the output latch data are read from the PC register.

In the input mode (PCCR<bit> = 0), the pin data are read from PC register.

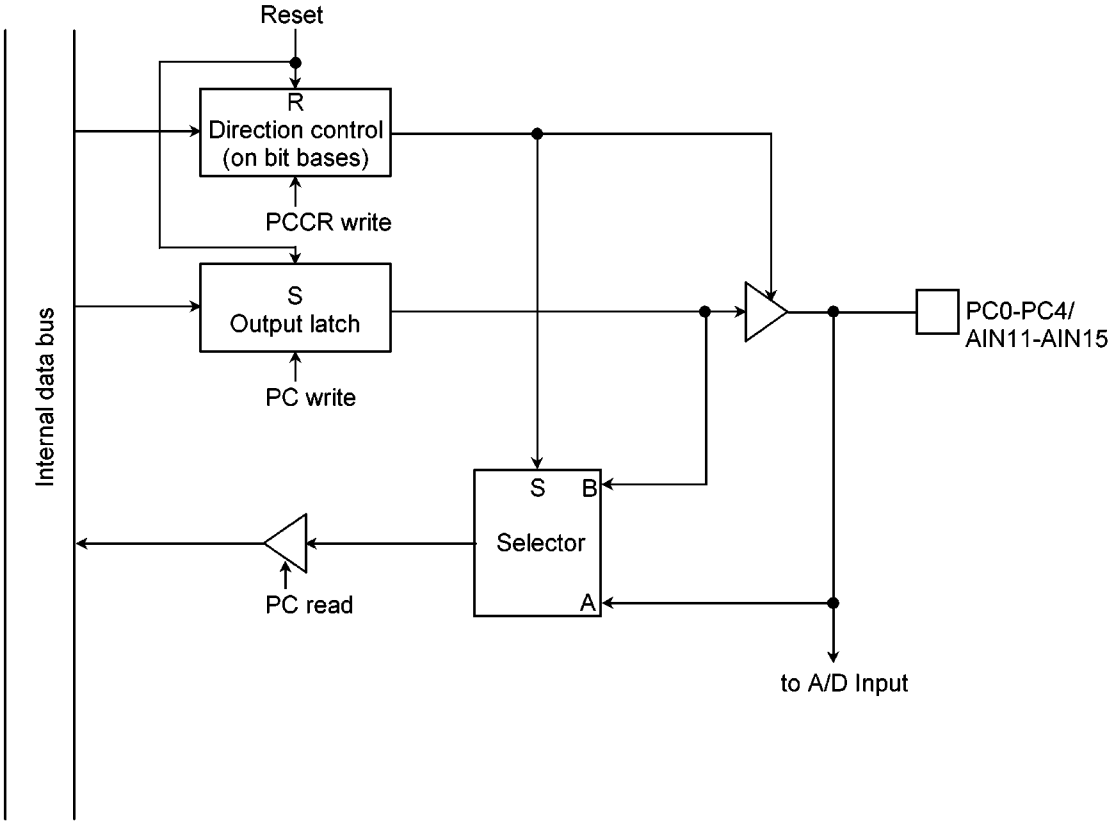


Figure 3.5.29 Port C (PC0 to PC4)

Port C Register

		7	6	5	4	3	2	1	0
PC (0087H)	bit Symbol				PC4	PC3	PC2	PC1	PC0
	Read/Write				R/W				
	After reset				1	1	1	1	1

Port C Control Register

		7	6	5	4	3	2	1	0
PCCR (0088H)	bit Symbol				PC4C	PC3C	PC2C	PC1C	PC0C
	Read/Write				W				
	After reset				0	0	0	0	0
	Function				0: Input mode		1: Output mode		

Prohibit read-modify-write

Figure 3.5.30 Registers for Port C

3.6 Chip select / Wait Controller, AM8 / $\overline{16}$ pin

TMP93C071 has a built-in controller used to control chip select ($\overline{CS0}$ to $\overline{CS2}$), wait and data bus size (8 or 16 bits) for any of the three block address areas.

And AM8 / $\overline{16}$ pin selects external data bus width for TMP93C071.

3.6.1 AM8 / $\overline{16}$ pin

① With fixed external 16-bit data bus and external 16-bit data bus or 8-bit data bus is selectable

Set this pin to "L".

The external data bus width is set by the chip select / wait control register which is described in section 3.6.3.

It is necessary to set the program memory to be accessed to 16-bit data bus after reset.

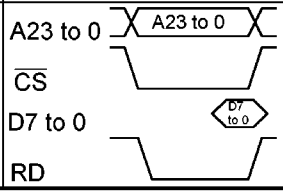
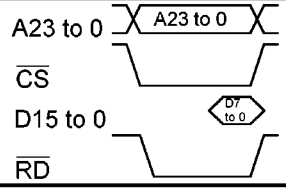
② With fixed external 8-bit data bus

Set this pin to "H".

The values of bit 4: <B0BUS>, <B1BUS>, and <B2BUS>, in the chip select / wait control register described in section 3.6.2 are invalid. The external 8-bit data bus is fixed.

3.6.2 Address / Data bus pins

D0 to 7, D8 to 15, Port 2/A20 to 23, A16 to 19, A8 to 15 and A0 to 7 function as address / data bus for connecting the external memories and so on.

		①	②
Products		TMP93C071F	
Number of address bus pins		max24 (to 16 MB)	max24 (to 16 MB)
Number of data bus pins		8	16
Number of multiplexed pins		8	16
Mode pins	$\overline{EA}$	VIL	
	AM8/ $\overline{16}$	VIH	VIL
Timing chart			

3.6.3 Chip select / Wait Control Registers

Figure 3.6.1 shows control registers.

One block address areas are controlled by 1-byte Chip select / WAIT control registers (B0CS, B1CS, B2CS).

(1) Selection bit for CS output

The control register bit 6: <CS0EN>, <CS1EN> and <CS2EN> are used for the chip select output ($\overline{CS0}$ to $\overline{CS2}$) from port terminals. When these bits are set to 1, the $\overline{CS0}$ to $\overline{CS2}$ are output. When these bits are set to 0, functions of the P60 / PWM4, P61 / PWM5 and P62 / PWM6 are selected. When these terminals are used as $\overline{CS0}$ to $\overline{CS2}$, the port function must be set to the output mode. These port terminals become the input port mode after reset. Therefore, the external pull-up or pull-down resister is necessary for initialization corresponding to usage.

(2) Data bus size select

Bit 4 (<B0BUS>, <B1BUS>, <B2BUS>) of the control register is used to specify data bus size. Setting this bit to 0 accesses the memory in 16-bit data bus mode; setting it to 1 accesses the memory in 8-bit data bus mode.

Changing data bus size depending on the access address is called dynamic bus sizing. Table 3.6.1) shows the details of the bus operation.

This bit is changed by the state of AM8 / $\overline{16}$ pin.

(3) Wait control

Control register bits 3 and 2 (<B0W1, 0>, <B1W1, 0>, <B2W1, 0>) are used to specify the number of waits.

These bits execute the following operation by setting.

"11" A 2-state wait is inserted.

"01" A 1-state wait is inserted.

"10" A 1-state wait is inserted.

"00" The bus cycle is completed without a wait (0 WAIT).

These bits are cleared to "00" (0 WAIT mode) by reset.

(4) Address area specification

Control register bits 1 and 0 (<B0C1, 0>, <B1C1, 0>, <B2C1, 0>) are used to specify the target address area. Setting these bits to 00 enables settings as follows:

- B0CS setting enabled when 2090H to 3FFFFFH is accessed.
- B1CS setting enabled when 2090H to 3FFFFFH is accessed.
- B2CS setting enabled when 2090H to 3FFFFFH is accessed.

Setting bits to 01 enables setting for each block when 400000H to 7FFFFFH is accessed. Setting bits to 10 enables them 800000H to BFFFFFH is accessed. Setting bits to 11 enables them when C00000H to FFFFFFH is accessed.

When the CSWAIT controller (CS0 to CS2) is used, do not set the same area with other CSs.
The channel of a smaller number must be set another area than the selected CS regardless of enable/disable (whichever B*CS<CS*EN> is 1 or 0).

Chip select/Wait Control Register 0

B0CS (0068H)

	7	6	5	4	3	2	1	0
bit Symbol		CS0EN		B0BUS	B0W1	B0W0	B0C1	B0C0
Read/Write		W		W	W		W	
After reset		0		0	0		0	
Function		0: Port60 /PWM4 (P6FC <P60F>) 1: CS0		BUS width control 0: 16-bit Bus 1: 8-bit Bus	Wait control 00: 0 WAIT 01: 1 WAIT 10: 1 WAIT 11: 2 WAIT		00: 2090H to 3FFFFFH 01: 400000H to 7FFFFF 10: 800000H to BFFFFF 11: C00000H to FFFFFF	

Prohibit read-modify-write

Chip select/Wait Control Register 1

B1CS (0069H)

	7	6	5	4	3	2	1	0
bit Symbol		CS1EN		B1BUS	B1W1	B1W0	B1C1	B1C0
Read/Write		W		W	W		W	
After reset		0		0	0		0	
Function		0: Port61 /PWM5 (P6FC <P61F>) 1: CS1		BUS width control 0: 16-bit Bus 1: 8-bit Bus	Wait control 00: 0 WAIT 01: 1 WAIT 10: 1 WAIT 11: 2 WAIT		00: 2090H to 3FFFFFH 01: 400000H to 7FFFFF 10: 800000H to BFFFFF 11: C00000H to FFFFFF	

Prohibit read-modify-write

Chip select/Wait Control Register 2

B2CS (0070H)

	7	6	5	4	3	2	1	0
bit Symbol		CS2EN		B2BUS	B2W1	B2W0	B2C1	B2C0
Read/Write		W		W	W		W	
After reset		0		0	0		0	
Function		0: Port62 /PWM6 (P6FC <P62F>) 1: CS2		BUS width control 0: 16-bit Bus 1: 8-bit Bus	Wait control 00: 0 WAIT 01: 1 WAIT 10: 1 WAIT 11: 2 WAIT		00: 2090H to 3FFFFFH 01: 400000H to 7FFFFF 10: 800000H to BFFFFF 11: C00000H to FFFFFF	

Prohibit read-modify-write

Figure 3.6.1 Chip select/Wait Control Registers

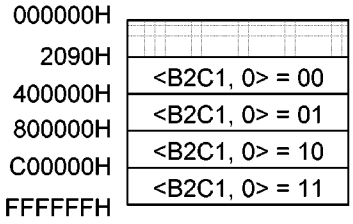
Table 3.6.1 Dynamic bus sizing

Operand data size	Operand start address	Memory data size	CPU address	CPU data	
				D15 to D8	D7 to D0
8 bits	2n + 0 (even number)	8 bits	2n + 0	xxxxx	b7 to b0
		16 bits	2n + 0	xxxxx	b7 to b0
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 to b0
		16 bits	2n + 1	b7 to b0	xxxxx
16 bits	2n + 0 (even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
		16 bits	2n + 0	b15 to b8	b7 to b0
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	xxxxx	b15 to b8
32 bits	2n + 0 (even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
			2n + 2	xxxxx	b23 to b16
			2n + 3	xxxxx	b31 to b24
		16 bits	2n + 0	b15 to b8	b7 to b0
			2n + 2	b31 to b24	b23 to b16
	2n + 1 (odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
			2n + 3	xxxxx	b23 to b16
			2n + 4	xxxxx	b31 to b24
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	b23 to b16	b15 to b8
			2n + 4	xxxxx	b31 to b24

xxxxx: During a read, data input to the bus is ignored. At write, the bus is at high impedance and the write strobe signal remains non-active.

3.6.4 Chip select / Wait control

An image of the actual Chip select / Wait control is shown below. The address area can be specified only one of the whole external memory area divided into four parts.



Note 1: Access priority is highest for built-in I/O, then built-in memory, and lowest for the chip select / wait controller.

Note 2: External areas other than B0CS to B2CS are accessed in 0 wait mode. When the AM8/$\overline{16}$ pin is set to "L", the data bus width is fixed to 16-bit. When the AM8/16 pin is set to "H", it is fixed to 8-bit. When using the chip select/wait controller, do not specify the same address area more than once. (The bit 4 to 2 of the B0CS to B2CS must be the same setting respectively when the same address area is specified.)

3.7 Timer Counter

3.7.1 8-bit Timer Counter 0 (TC0)

TC0 is an 8-bit timer counter. The TC0 is counted by a source clock selected by the Timer Counter Control Register10 (TCCR10). The TC0 is composed of an 8-bit Up-counter, Timer register (TREG0), and 8-bit Comparator.

(1) Block diagram

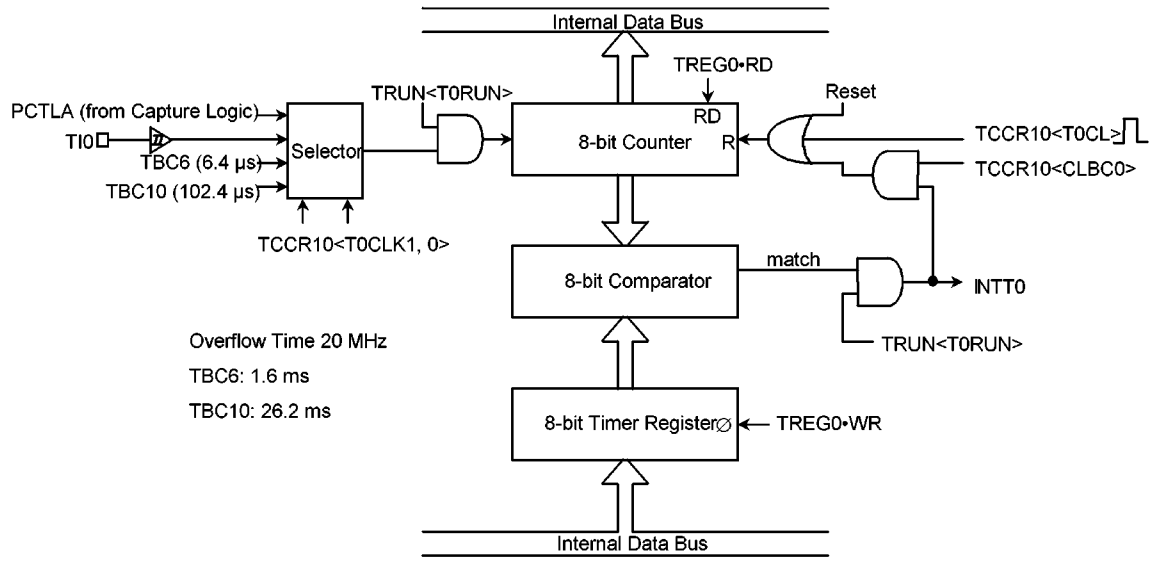


Figure 3.7.1 composition of the timer counter 0

① Up-counter

The Up-counter is an 8-bit binary counter. The Up-counter counts the input clock specified with <T0CLK1, 0> of the Timer Counter Control Register 10 (TCCR10). The input clock of the TC0 can be selected from TBC6 (2^7/fc), TBC10 (2^{11}/fc) of the Time Base Counter (TBC), TI0 (P57) of an external input and PCTLA from the Capture Input controller (CAPIN). The Up-counter can be controlled the start and stop of counting by the Timer start control register (TRUN).

The value of the Up-counter is cleared by reset and the timer counter stops at the same time.TCCR10<CLBC0> can set the clear enable/disable of the Up-counter by the match output of the comparator. When TCCR10<T0CL> is set to "1", the Up-counter is compulsorily cleared.

② Timer register (TREG0)

The Timer Register (TREG0) is an 8-bit register for setting the interval time.

When a value of the timer register matches to the value of Up-counter, the match signal is output from the comparator. If the value of the timer register is set to "00H", the match signal is output at the time of the up-counter overflow. When a new value is set to the timer register, the new value is forwarded to the comparator at once. Reading the timer register can read the value of the 8-bit up-counter in real time. (The value of the interval time is not read.)

Because the timer register is not initialized by the reset operation (The value of the timer register becomes an unknown value), attention is necessary.

③ Comparator

The 8-bit comparator compares a value of the up-counter and a value of the timer register.

When the value of the timer register matches to the value of the Up-counter, the interrupt request (INTT0) is occurred. TCCR10<CLBC0> can set the clear enable/disable of the up-counter by the match output of the comparator.

(2) Control Register

Timer start control register

TRUN (0027H)

	7	6	5	4	3	2	1	0
bit Symbol			T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write	R/W							
After reset			0	0	0	0	0	0
Function			0: Stop 1: Start					

Timer Counter Control register10

TCCR10 (0028H)

	7	6	5	4	3	2	1	0
bit Symbol	CLBC1	T1CL	T1CLK1	T1CLK0	CLBC0	T0CL	T0CLK1	T0CLK0
Read/Write	R/W	R/W	R/W		R/W	R/W	R/W	
After reset	0	0	0		0	0	0	
Function	TC1 timer/ counter match clear enable 0: Disable 1: Enable	TC1 timer/ counter clear 0: — 1: Clear (One-shot)	TC1 source clock selection 00: TI1 01: TBC2 10: TBC4 11: TBC6		TC0 timer/ counter match clear enable 0: Disable 1: Enable	TC0 timer/ counter clear 0: — 1: Clear (One-shot)	TC0 source clock selection 00: PCTLA (from CAPIN) 01: TI0 10: TBC6 11: TBC10	

Timer Register 0

TREG0 (0029H)

	7	6	5	4	3	2	1	0
bit Symbol	TC/TR07	TC/TR06	TC/TR05	TC/TR04	TC/TR03	TC/TR02	TC/TR01	TC/TR00
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG0 register R: Reading TC0 counter							

Figure 3.7.2 Registers for TC0

(3) Operation explanation

① Generating interrupts at a constant cycle.

First of all, TC0 is stopped. The input clock and the cycle are set to TCCR10 and TREG0 respectively. Next, the TC0 interrupt(INTT0) is enabled. The TC0 is counted at the end.

Example: When the TC0 interrupt of each 32 us is generated with fc = 20 MHz, Set each register in the following order.

	MSB LSB	
TRUN	– – – – – – – 0	The count of the TC0 is stopped.
TCCR10	– – – – 1 * 1 0	The source clock is set to TBC6 (6.4 us).
TREG0	0 0 0 0 0 1 0 1	32 us ÷ 6.4 us = 05 H is set to the timer register.
INT4T0	0 n n n – – – –	INTT0 interrupt is enabled with Level "nnn".
TRUN	– – – – – – – 1	The count of the TC0 is started.

(Note) *:don't care —: no change

Table 3.7.1 Interrupt Interval for the TC0

Interrupt Interval (fc = 20 MHz)	Resolution	Input Clock
6.4 us to 1.638 ms	6.4 us	TBC6 (2^7/fc)
102.4 us to 26.214 ms	102.4 us	TBC10 (2^{11}/fc)

② Using the TC0 as an event counter

It is a mode to which the count up is done with rising edge of TI0 (P57)input. The INTT0 interrupt request is occurred when the value of the counter matches to the value of the TREG0.

The maximum input frequency is fc/2^4 [Hz] (1.25 MHz at fc = 20 MHz). The minimum input pulse width is 2^3/fc [s] (400 ns at fc = 20 MHz) at both H levels and L levels.

③ Using the TC0 for counting of the Play-Back-Control signal (CTL)

It is a mode to which the count up is done with rising edge of the PCTLA. The PCTLA is the CTL signal switched polarity with the capture input controller, and the CTL signal is input ted from CTLIN (P80) pin. The INTT0 interrupt request is occurred when the value of the counter matches to the value of the TREG0. The maximum input frequency is fc/2^4 [Hz] (1.25 MHz at fc = 20 MHz).

The minimum input pulse width is 2^3/fc [s] (400 ns at fc = 20 MHz) at both H levels and L levels.

3.7.2 16-bit Timer counter 1, 2, 3, 4, 5 (TC1, TC2, TC3, TC4, TC5)

The TC1 to TC5 are the 16-bit timer counter which count with the source clock selected by the Timer counter control register (TCCR10, TCCR32, TCCR54). The TC1 to TC5 are composed of the 16-bit up-counter, the timer register (TREG1L/TREG1H, TREG2L/TREG2H, TREG3L/TREG3H, TREG4L/TREG4H, TREG5L/TREG5H), and the 16-bit comparator. The basic composition of these timer counter are all the same except for the input clock source, and the basic operation are similar to the TC0 operation with 16-bit.

(1) Block diagram

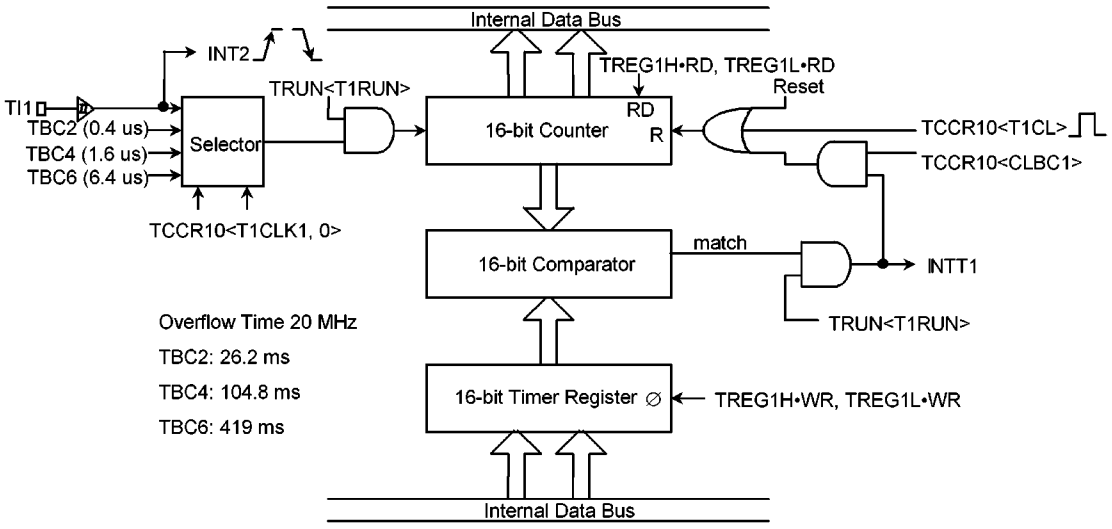


Figure 3.7.3 Composition of the Timer Counter 1

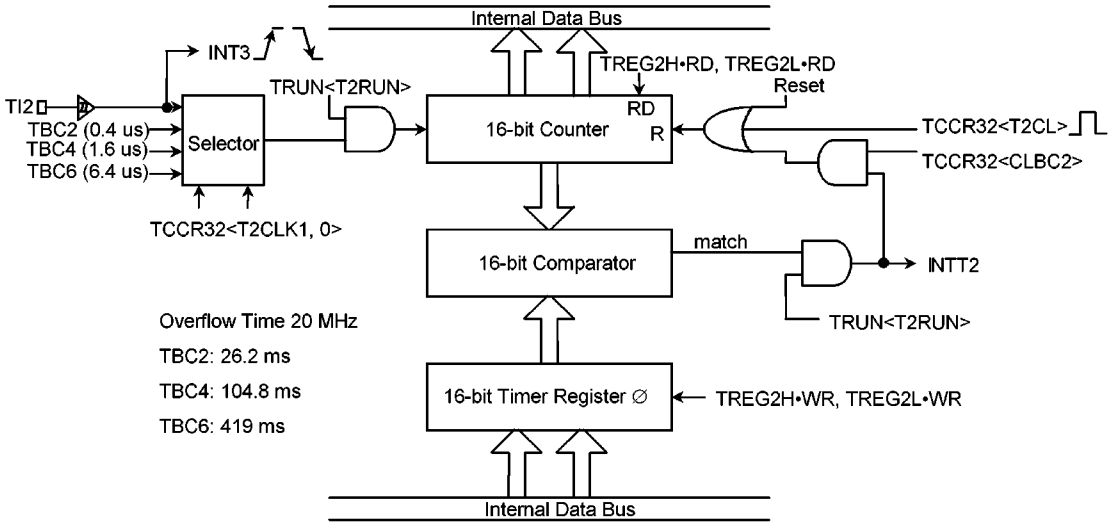


Figure 3.7.4 Composition of the Timer Counter 2

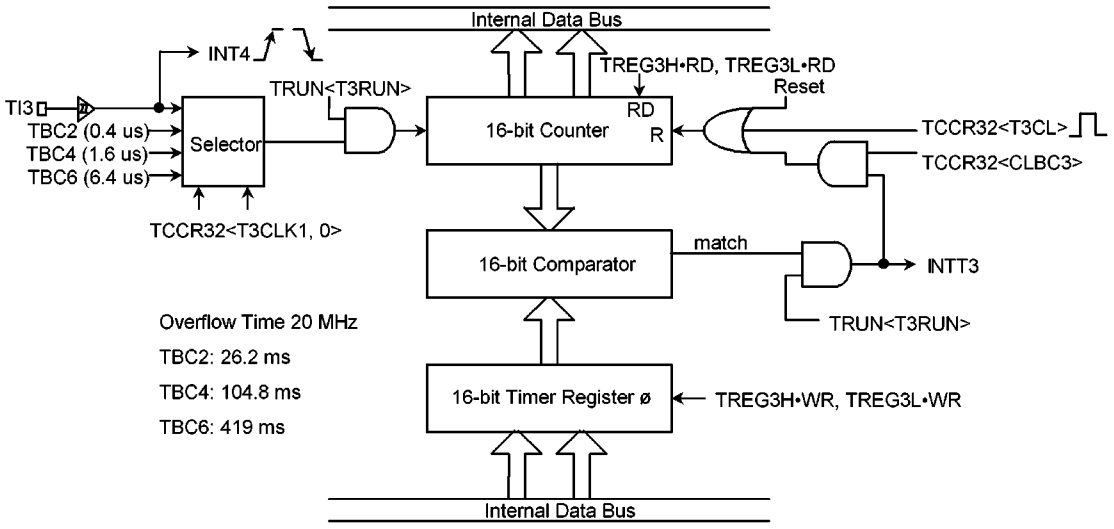


Figure 3.7.5 Composition of the Timer Counter 3

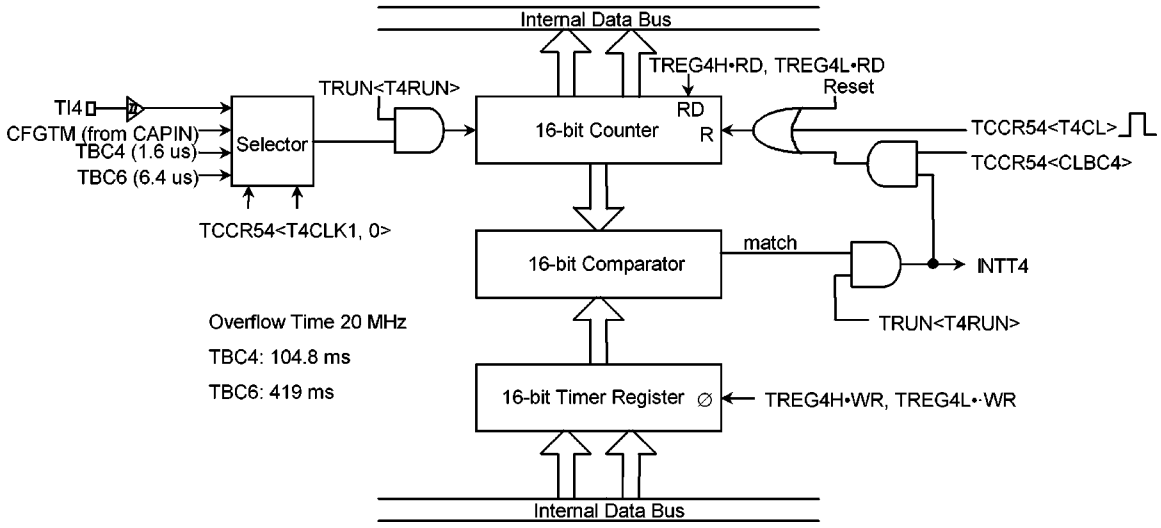


Figure 3.7.6 Composition of the Timer Counter 4

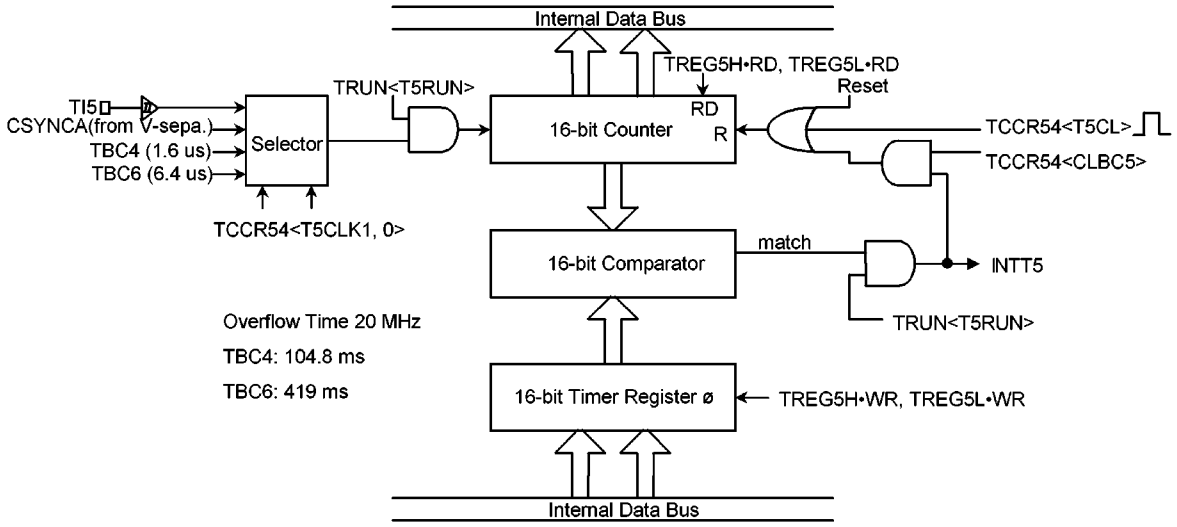


Figure 3.7.7 Composition of the Timer Counter 5

(2) Timer Counter 1(TC1)

TC1 is the 16-bit timer counter. The TC1 is counted by the source clock selected by <T1CLK1,0> of the Timer counter control register10(TCCR10). The input clock of the TC1 can be selected from TBC2 (2^3/fc), TBC4 (2^5/fc), TBC6 (2^7/fc) of the Time Base Counter (TBC), TI1 (P52) of an external input. The up-counter can be controlled the start and stop of counting by the Timer start control register (TRUN). The value of the up-counter is cleared by reset and the timer counter stops at the same time. TCCR10<CLBC1> can set the clear enable/disable of the Up-counter by the match output of the comparator. When TCCR10<T1CL> is set to "1", the Up-counter is compulsorily cleared. The timer register are the 8-bit*2byte registers (TREG1L, TREG1H) for setting the value of the interval time. Set the value in order of TREG1L → TREG1H.When the value of the timer register matches to the value of the Up-counter, the match signal is output from the comparator, and the interrupt request (INTT1) is occurred. If the value of the timer register is set to 0000H, the match signal is output at the time of the up-counter overflow. When a new value is set to the timer register, the new value is forwarded to the comparator at once. Reading the timer register in order of TREG1L → TREG1H can read the value of the 16-bit up-counter in real time. (The value of the interval time is not read.)

Because the timer register is not initialized by the reset operation (The value of the timer register becomes an unknown value), attention is necessary.

Timer Start Control Register

TRUN (0027H)

	7	6	5	4	3	2	1	0
bit Symbol			T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write	R/W							
After reset	0		0	0	0	0	0	0
Function	0: Stop 1: Start							

Timer Counter Control Register 10

TCCR10 (0028H)

	7	6	5	4	3	2	1	0
bit Symbol	CLBC1	T1CL	T1CLK1	T1CLK0	CLBC0	T0CL	T0CLK1	T0CLK0
Read/Write	R/W	R/W	R/W		R/W	R/W	R/W	
After reset	0	0	0		0	0	0	
Function	TC1 timer/ counter match clear enable 0: Disable 1: Enable	TC1 timer/ counter clear 0: — 1: Clear (One-shot)	TC1 source clock selection 00: TI1 01: TBC2 10: TBC4 11: TBC6		TC0 timer/ counter match clear enable 0: Disable 1: Enable	TC0 timer/ counter clear 0: — 1: Clear (One-shot)	TC0 source clock selection 00: PCTLA (from CAPIN) 01: TI0 10: TBC6 11: TBC10	

Timer Register 1L

TREG1L (002AH)

	7	6	5	4	3	2	1	0
bit Symbol	TC/TR17	TC/TR16	TC/TR15	TC/TR14	TC/TR13	TC/TR12	TC/TR11	TC/TR10
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG1L register R: Reading TC1L counter							

Timer Register 1H

TREG1H (002BH)

	7	6	5	4	3	2	1	0
bit Symbol	TC/TR1F	TC/TR1E	TC/TR1D	TC/TR1C	TC/TR1B	TC/TR1A	TC/TR19	TC/TR18
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG1H register R: Reading TC1H counter							

Note: Setting and reading in order of TREG1L→TREG1H are necessary.

Figure 3.7.8 Registers for TC1

(3) Timer Counter 2 (TC2)

TC2 is the 16-bit timer counter. The TC2 is counted by the source clock selected by <T2CLK1,0> of the Timer counter control register32(TCCR32).The input clock of the TC2 can be selected from TBC2 (2^3/fc), TBC4 (2^5/fc), TBC6 (2^7/fc) of the Time Base Counter (TBC), TI2 (P51) of an external input. The up-counter can be controlled the start and stop of counting by the Timer start control register (TRUN). The value of the up-counter is cleared by reset and the timer counter stops at the same time. TCCR10<CLBC2> can set the clear enable/disable of the Up-counter by the match output of the comparator. When TCCR32<T2CL> is set to 1, the Up-counter is compulsorily cleared. The timer register are the 8-bit*2 byte registers (TREG2L, TREG2H) for setting the value of the interval time. Set the value in order of TREG2L → TREG2H. When the value of the timer register matches to the value of the Up-counter, the match signal is output from the comparator, and the interrupt request (INTT2) is occurred. If the value of the timer register is set to 0000H, the match signal is output at the time of the up-counter overflow. When a new value is set to the timer register, the new value is forwarded to the comparator at once. Reading the timer register in order of TREG2L → TREG2H can read the value of the 16-bit up-counter in real time. (The value of the interval time is not read.)

Because the timer register is not initialized by the reset operation (The value of the timer register becomes an unknown value), attention is necessary.

Timer start control Register

TRUN (0027H)

	7	6	5	4	3	2	1	0
bit Symbol			T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write	R/W							
After reset			0	0	0	0	0	0
Function	0: Stop 1: Start							

Timer Counter Control Register 32

TCCR32 (002CH)

	7	6	5	4	3	2	1	0
bit Symbol	CLBC3	T3CL	T3CLK1	T3CLK0	CLBC2	T2CL	T2CLK1	T2CLK0
Read/Write	R/W	R/W	R/W		R/W	R/W	R/W	
After reset	0	0	0		0	0	0	
Function	TC3 timer/ counter match clear enable 0: Disable 1: Enable	TC3 timer/ counter clear 0: — 1: Clear (One-shot)	TC3 source clock selection 00: TI3 01: TBC2 10: TBC4 11: TBC6		TC2 timer/ counter match clear enable 0: Disable 1: Enable	TC2 timer/ counter clear 0: — 1: Clear (One-shot)	TC2 source clock selection 00: TI2 01: TBC2 10: TBC4 11: TBC6	

Timer Register 2L

TREG2L (002EH)

	7	6	5	4	3	2	1	0
bit Symbol	TC/TR27	TC/TR26	TC/TR25	TC/TR24	TC/TR23	TC/TR22	TC/TR21	TC/TR20
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG2L register R: Reading TC2L counter							

Timer Register 2H

TREG2H (002FH)

	7	6	5	4	3	2	1	0
bit Symbol	TC/TR2F	TC/TR2E	TC/TR2D	TC/TR2C	TC/TR2B	TC/TR2A	TC/TR29	TC/TR28
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG2H register R: Reading TC2H counter							

Note: Setting and reading in order of TREG2L→TREG2H are necessary.

Figure 3.7.9 Registers for TC2

(4) Timer Counter 3 (TC3)

TC3 is the 16-bit timer counter. The TC3 is counted by the source clock selected by <T3CLK1, 0> of the Timer counter be register32 (TCCR32). The input clock of the TC3 can be selected from TBC2 (2^3/fc), TBC4 (2^5/fc), TBC6 (2^7/fc) of the Time Base Counter (TBC), TI3 (P50) of an external input. The up-counter can be controlled the start and stop of counting by the Timer start control register (TRUN).The value of the up-counter is cleared by reset and the timer counter stops at the same time. TCCR32<CLBC3> can set the clear enable/disable of the Up-counter by the match output of the comparator. When TCCR32<T3CL> is set to 1, the Up-counter is compulsorily cleared. The timer register are the 8-bit*2byte registers (TREG3L, TREG3H) for setting the value of the interval time. Set the value in order of TREG3L → TREG3H. When the value of the timer register matches to the value of the Up-counter, the match signal is output from the comparator, and the interrupt request (INTT3) is occurred. If the value of the timer register is set to 0000H, the match signal is output at the time of the up-counter overflow. When a new value is set to the timer register, the new value is forwarded to the comparator at once. Reading the timer register in order of TREG3L → TREG3H can read the value of the 16-bit up-counter in real time. (The value of the interval time is not read.)

Because the timer register is not initialized by the reset operation (The value of the timer register becomes an unknown value), attention is necessary.

Timer Start Control Register

TRUN (0027H)

	7	6	5	4	3	2	1	0
bit Symbol			T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write	R/W							
After reset			0	0	0	0	0	0
function			0: Stop 1: Start					

Timer Counter Control Register 32

TCCR32 (002CH)

	7	6	5	4	3	2	1	0
bit Symbol	CLBC3	T3CL	T3CLK1	T3CLK0	CLBC2	T2CL	T2CLK1	T2CLK0
Read/Write	R/W	R/W	R/W		R/W	R/W	R/W	
After reset	0	0	0		0	0	0	
Function	TC3 timer/ counter match clear enable 0: Disable 1: Enable	TC3 timer/ counter clear 0: — 1: Clear (One-shot)	TC3 source clock selection 00: TI3 01: TBC2 10: TBC4 11: TBC6		TC2 timer/ counter match clear enable 0: Disable 1: Enable	TC2 timer/ counter clear 0: — 1: Clear (One-shot)	TC2 source clock selection 00: TI2 01: TBC2 10: TBC4 11: TBC6	

Timer Register 3L

TREG3L (0030H)	7	6	5	4	3	2	1	0
bit Symbol	TC/TR37	TC/TR36	TC/TR35	TC/TR34	TC/TR33	TC/TR32	TC/TR31	TC/TR30
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG3L register R: Reading TC3L counter							

Timer Register 3H

TREG3H (0031H)	7	6	5	4	3	2	1	0
bit Symbol	TC/TR3F	TC/TR3E	TC/TR3D	TC/TR3C	TC/TR3B	TC/TR3A	TC/TR39	TC/TR38
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG3H register R: Reading TC3H counter							

Note: Setting and reading in order of TREG3L → TREG3H are necessary.

Figure 3.7.10 Registers for TC3

(5) Timer Counter 4 (TC4)

TC4 is the 16-bit timer counter. The TC4 is counted by the source clock selected by <T4CLK1, 0> of the Timer counter control register54(TCCR54).The input clock of the TC4 can be selected from TBC4 (2^5/fc), TBC6 (2^7/fc) of the Time Base Counter (TBC), TI4 (P56) of an external input, CFGTM from the capture input controller (CAPIN). The up-counter can be controlled the start and stop of counting by the Timer start control register (TRUN). The value of the up-counter is cleared by reset and the timer counter stops at the same time. TCCR54<CLBC4> can set the clear enable/disable of the Up-counter by the match output of the comparator. When TCCR54<T4CL> is set to 1, the Up-counter is compulsorily cleared. The timer register are the 8-bit*2 byte registers (TREG4L, TREG4H) for setting the value of the interval time. Set the value in order of TREG4L → TREG4H. When the value of the timer register matches to the value of the Up-counter, the match signal is output from the comparator, and the interrupt request (INTT4) is occurred. If the value of the timer register is set to 0000H, the match signal is output at the time of the up-counter overflow. When a new value is set to the timer register, the new value is forwarded to the comparator at once. Reading the timer register in order of TREG4L → TREG4H can read the value of the 16-bit up-counter in real time. (The value of the interval time is not read.) Because the timer register is not initialized by the reset operation (The value of the timer register becomes an unknown value), attention is necessary.

Timer Start Control Register

TRUN (0027H)	7	6	5	4	3	2	1	0
bit Symbol			T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write	R/W							
After reset			0	0	0	0	0	0
Function			0: Stop 1: Start					

Timer Counter Control Register 54

TCCR54 (002DH)	7	6	5	4	3	2	1	0
bit Symbol	CLBC5	T5CL	T5CLK1	T5CLK0	CLBC4	T4CL	T4CLK1	T4CLK0
Read/Write	R/W	R/W	R/W		R/W	R/W	R/W	
After reset	0	0	0		0	0	0	
Function	TC5 timer/ counter match clear enable 0: Disable 1: Enable	TC5 timer/ counter clear 0: — 1: Clear (One-shot)	TC5 source clock selection 00: TI5 01: CSYNCA (from V-sepa) 10: TBC4 11: TBC6		TC4 timer/ counter match clear enable 0: Disable 1: Enable	TC4 timer/ counter clear 0: — 1: Clear (One-shot)	TC4 source clock selection 00: TI4 01: CFGTM (from CAPIN) 10: TBC4 11: TBC6	

Timer Register 4L

TREG4L (0033H)	7	6	5	4	3	2	1	0
bit Symbol	TC/TR47	TC/TR46	TC/TR45	TC/TR44	TC/TR43	TC/TR42	TC/TR41	TC/TR40
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG4L register R: Reading TC4L counter							

Timer Register 4H

TREG4H (0033H)	7	6	5	4	3	2	1	0
bit Symbol	TC/TR4F	TC/TR4E	TC/TR4D	TC/TR4C	TC/TR4B	TC/TR4A	TC/TR49	TC/TR48
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG4H register R: Reading TC4H counter							

Note: Setting and reading in order of TREG4L → TREG4H are necessary.

Figure 3.7.11 Registers for TC4

(6) Timer Counter 5 (TC5)

TC5 is the 16-bit timer counter. The TC5 is counted by the source clock selected by <T5CLK1, 0> of the Timer counter control register54 (TCCR54). The input clock of the TC5 can be selected from TBC4 (2^5/fc), TBC6 (2^7/fc) of the Time Base Counter (TBC), TI5 (P55) of an external input, CSYNCA from the composit-sync-separation (CSYNC). The up-counter can be controlled the start and stop of counting by the Timer start control register (TRUN). The value of the up-counter is cleared by reset and the timer counter stops at the same time. TCCR54<CLBC5> can set the clear enable/disable of the Up-counter by the match output of the comparator. When TCCR54<T5CL> is set to 1, the Up-counter is compulsorily cleared. The timer register are the 8-bit*2 byte registers (TREG5L, TREG5H) for setting the value of the interval time. Set the value in order of TREG5L → TREG5H. When the value of the timer register matches to the value of the Up-counter, the match signal is output from the comparator, and the interrupt request (INTT5) is occurred. If the value of the timer register is set to 0000H, the match signal is output at the time of the up-counter overflow. When a new value is set to the timer register, the new value is forwarded to the comparator at once. Reading the timer register in order of TREG5L → TREG5H can read the value of the 16-bit up-counter in real time. (The value of the interval time is not read.) Because the timer register is not initialized by the reset operation (The value of the timer register becomes an unknown value), attention is necessary.

Timer Control Register

TRUN (0027H)

	7	6	5	4	3	2	1	0
bit Symbol			T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write			R/W					
After reset			0	0	0	0	0	0
Function			0: Stop 1: Start					

Timer Counter Control Register 54

TCCR54 (002DH)

	7	6	5	4	3	2	1	0
bit Symbol	CLBC5	T5CL	T5CLK1	T5CLK0	CLBC4	T4CL	T4CLK1	T4CLK0
Read/Write	R/W	R/W	R/W		R/W	R/W	R/W	
After reset	0	0	0		0	0	0	
Function	TC5 timer/ counter match clear enable 0: Disable 1: Enable	TC5 timer/ counter clear 0: — 1: Clear (One-shot)	TC5 source clock selection 00: TI5 01: CSYNCA (from V-sepa) 10: TBC4 11: TBC6		TC4 timer/ counter match clear enable 0: Disable 1: Enable	TC4 timer/ counter clear 0: — 1: Clear (One-shot)	TC4 source clock selection 00: TI4 01: CFGTM (from CAPIN) 10: TBC4 11: TBC6	

Timer Register 5L

TREG5L (0034H)

	7	6	5	4	3	2	1	0
bit Symbol	TC/TR57	TC/TR56	TC/TR55	TC/TR54	TC/TR53	TC/TR52	TC/TR51	TC/TR50
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG5L register R: Reading TC5L counter							

Timer Register 5H

TREG5H (0035H)

	7	6	5	4	3	2	1	0
bit Symbol	TC/TR5F	TC/TR5E	TC/TR5D	TC/TR5C	TC/TR5B	TC/TR5A	TC/TR59	TC/TR58
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	W: Writing TREG5H register R: Reading TC5L counter							

Note: Setting and reading in order of TREG5L→TREG5H are necessary.

Figure 3.7.12 Registers for TC5

(7) Operation explanation

The follows explains a setting method with using an example of the TC1.

<Generating interrupts at a constant cycle.>

First of all, TC1 is stopped. The input clock and the cycle are set to TCCR10 and TREG1L/ TREG1H respectively. Next, the TC1 interrupt(INTT1) is enabled. The TC1 is counted at the end.

Example: When the TC1 interrupt of each 64 ms is generated with fc = 20 MHz, Set each register in the following order.

	MSB LSB	
TRUN	– – – – – – 1 –	The count of the TC1 is stopped.
TCCR10	1 * 1 1 – – – –	The source clock is set to TBC6 (6.4 us).
TREG1L	0 0 0 1 0 0 0 0	64 ms ÷ 6.4 us = 2710 H is set to the timer register.
TREG1H	0 0 1 0 0 1 1 1	
INTT1T2	– – – – 0 n n n	INTT1 interrupt is enabled with Level "nnn"
TRUN	– – – – – – 1 –	The count of the TC1 is started.

(Note)*: don't care –: no change

Table3.7.2 Interrupt Interval for the TC1 to TC5

Interrupt Interval (at fc = 20 MHz)	Resolution	Input Clock
0.4 us to 26.214 ms	0.4 us	TBC2 (2^3/fc)
1.6 us to 104.856 ms	1.6 us	TBC4 (2^5/fc)
6.4 us to 419.424 ms	6.4 us	TBC6 (2^7/fc)

· The TI1 to TI5, CSYNCA, CFGTM are counted up with the rising edge of each input.

· The maximum input frequency is fc/2^4 [Hz] (1.25 MHz at fc = 20 MHz). The minimum input pulse width is 2^3/fc [s] (400 ns at fc = 20 MHz) at both H levels and L levels.

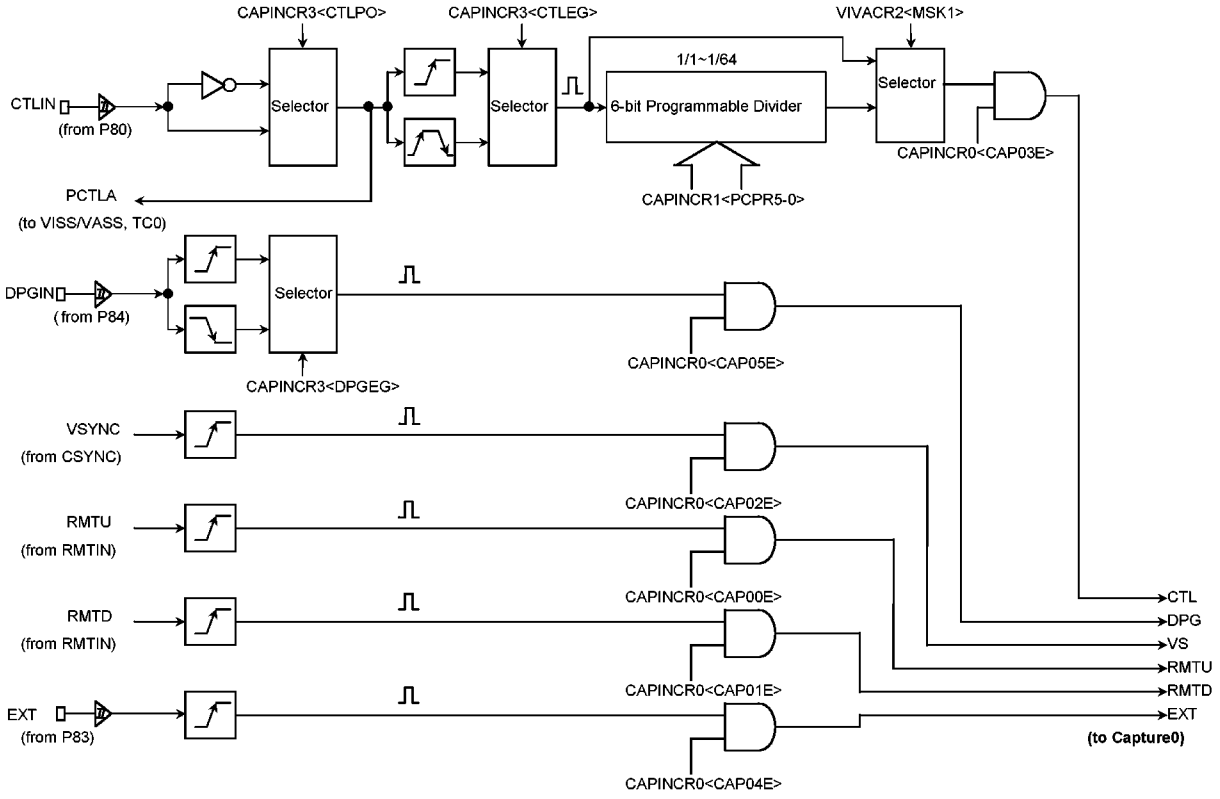
3.8 CAPTURE CIRCUIT

The capture circuit latches the time data of the time base counter (TBC) and the captures input status (Trigger input signal). The data of TBC can be latched by the trigger input signal and the interrupt request is generated to CPU. By using the capture circuit, the time can be measured in high resolution for servo control.

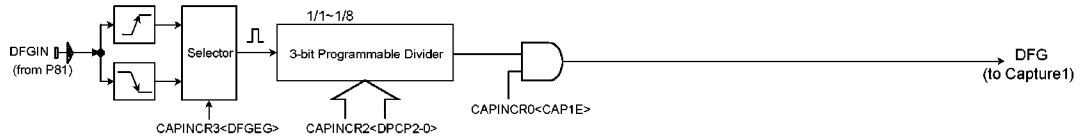
The capture circuit consists of the capture 0 (CAP0) with eight (8) level of 24-bit FIFO (first-in first-out) as buffer, the capture 1 (CAP1) and the capture 2 (CAP2) with single level of 17-bit FIFO as buffers.

3.8.1 Capture Input Control Circuit (CAPIN)

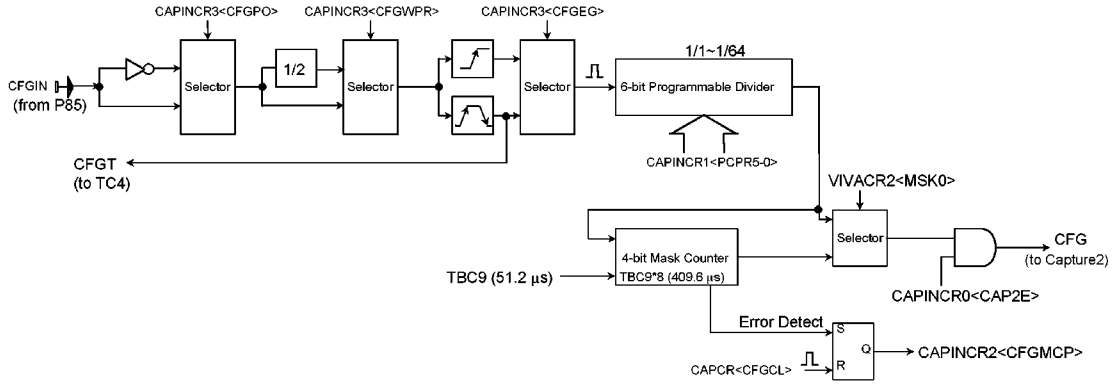
Capture input control circuit (CAPIN) controls the trigger input signals of the three channels of capture circuits (CAP0, CAP1 and CAP2), and comprises the capture 0 input control circuit, capture 1 input control circuit and capture 2 input control circuit.



(a) Capture 0 (CAP0) input control circuit



(b) Capture 1 (CAP1) input control circuit



(c) Capture 2 (CAP2) input control circuit

Figure 3.8.1 Capture input control circuit

(1) Capture 0 (CAP0) Input Control

① Remote control signal rising edge (RMTU)

This is the rising edge signal of the remote control signal detected by the remote control signal input circuit (RMTIN).

Enabling this trigger input to capture 0 (CAP0) is controlled by Capture input control register 0 CAPINCR0<CAP00E>.

② Remote control signal falling edge (RMTD)

This is the falling edge signal of the remote control signal detected by the remote control signal input circuit (RMTIN).

Enabling this trigger input to capture 0 (CAP0) is controlled by Capture input control register 0 CAPINCR0<CAP01E>.

③ Sync. separation input (VSYNC)

This is the vertical Synchronizing signal (VSYNC) separated from Composite Sync signal by Sync signal sparator (CSYNC).

Enabling this trigger input to capture 0 (CAP0) controlled by CAPINCR0 <CAP02E>.

④ Control signal input (CTLIN)

This is the input pin for the control pulse signal (CTL) which formed into a square wave with the external CTL amplifier circuit. The polarity of the external input (CTLIN) can be selected by the capture input control register 3 (CAPINCR3) <CTLPO>. The capturing timing can be selected from either rising edge of control signal or both rising and falling edges by the capture input control register 3 (CAPINCR3) <CTLEG>. The detected edge signal can be divided by a 6-bit programmable frequency divider. The frequency division ratio can be set by the capture input control register 1 (CAPINCR1) <PCPR5-0>. The frequency division ratios can be set from <PCPR5-0> = 00H (1/1 frequency division) to <PCPR5-0> = 3FH (1/64 frequency division). Selection of whether edge signal is divided or bypassed is carried out by the VISS/VASS control register 2 (VIVACR2) <MSK1>.

Enabling the trigger input to capture 0 (CAP0) is controlled by CAPINCR0 <CAP03E>.

⑤ EXT Input (EXT)

This is the input pin for the rising edge signal from EXT terminal.

Enabling the trigger input to capture 0 (CAP0) is controlled by CAPINCR0 <CAP04E>.

⑥ Drum PG input (DPGIN)

The external Drum PG signal is inputted to this pin. Detection of the rising edge or falling edge of the Drum PG signal is carried out by CAPINCR3 <DPGEG>.

Enabling the trigger input to capture 0 (CAP0) is controlled by CAPINCR0 <CAP05E>.

(2) Capture 1 (CAP 1) Input Control

The trigger of capture 1 (CAP1) is the external Drum FG signal (DFGIN). Detection of the rising edge or falling edge of the DFG signal from terminal P81 (DFGIN) is carried out by CAPINCR3 <DFGEG>. The detected edge signal can be divided by a 3-bit programmable frequency divider. The frequency division ratio can be set by <DPCP2-0> of capture input control register 2 (CAPINCR2). The frequency division ratios can be set from <DPCP2-0> = 0H (1/1 frequency division) to <DPCP2-0> = 7H (1/8 frequency division). Enabling the trigger input to capture 1 (CAP1) is controlled by <CAP1E> of capture input control register 0 (CAPINCR0).

(3) Capture 2 (CAP2) Input Control

The trigger of capture 2 (CAP2) is the external capstan FG signal (CFGIN). The polarity of the external input (CFGIN) can be selected by CAPINCR3 <CFGPO>. The input signal can be divided 1/1 or 1/2 by using <CFGWPR> of capture input control register 3 (CAPINCR3). Either the rising edge or both the rising and falling edges of the CFGIN signal can be selected by CAPINCR3<CFGEG>. The detected signal can be divided by a 6-bit programmable frequency divider. The frequency division ratio can be set by <PCPR5-0> of CAPINCR1 in common with the capturing for CTL. The divided CFGIN signal can be masked during an interval of TBC4 (2^{10}/fc) x 8 [s] by a 4-bit mask counter. CFGIN signal during the mask interval are not inputted to CAP2, but CFG flag (CAPINCR2 <CFGMCP>) is set to 1. The CFG flag can be reset to "0" by <CFGCL> of capture control register (CAPCR). The selection between subjecting the CFGIN signal to mask processing or by passing it can be selected by <MSK0> of VISS/VASS control register 2 (VIVACR2).

Enabling the trigger input to capture 2 (CAP2) is controlled by <CAP2E> of capture input control register 0 (CAPINCR0). The CFGTM signal is the pulse signal which output at the rising edge and falling edge of the CAPIN signal. The CFGTM is counted up by the 16-bit timer counter 4 (TC4).

Figure 3.5.2 shows the timing chart for capture 2 input control.

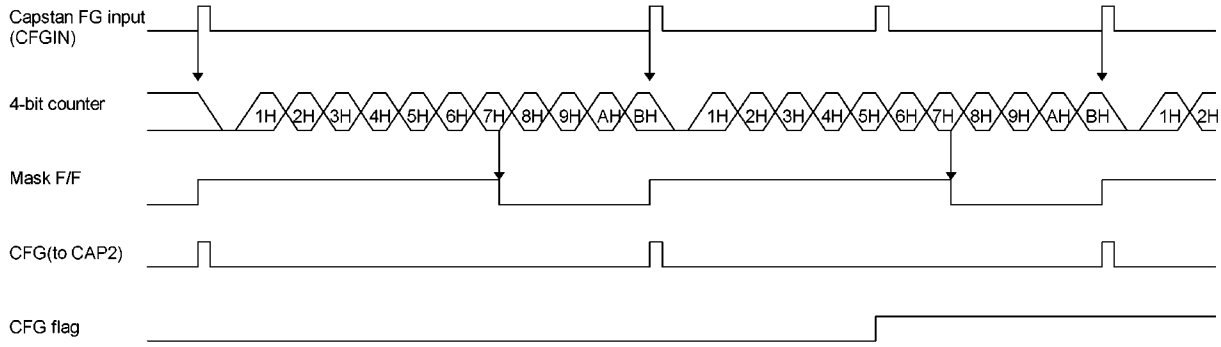


Figure3.8.2 Capture 2 Input Control Timing Chart (VIVACR2<MSK0> = 0)

(4) Capture Input Control Register

Capture Input Control Register 0

CAPINCR0 (0046H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2E (CFG)	CAP1E (DFG)	CAP05E (DPG)	CAP04E (EXT)	CAP03E (CTL)	CAP02E (VS)	CAP01E (RMTD)	CAP00E (RMTU)
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Capture 0, 1/2 input source enable/disable control 0: Disable 1: Enable							

Capture Input Control Register 1

CAPINCR1 (0047H)

	7	6	5	4	3	2	1	0
bit Symbol	PCTLCK1	PCTLCK0	PCPR5	PCPR4	PCPR3	PCPR2	PCPR1	PCPR0
Read/Write	R/W		R/W					
After reset	0	0	0	0	0	0	0	0
Function	CTL duty measure clock select (VIVACR2<PCTLCKS> = 0) 00: TBC3 01: TBC5 10: TBC7 11: TBC10		6-bit programmable divider control Dividing rate selection for CTLIN (from P80) and CFGIN (from P85). 000000: 1/1 000001: 1/2 --- 111111:1/64					

Capture Input Control Register 2

CAPINCR2 (0048H)

	7	6	5	4	3	2	1	0
bit Symbol		RMTST	RMTP0	RMTBP	CFGMCP	DPCP2	DPCP1	DPCP0
Read/Write		R/W	R/W	R/W	R	R/W		
After reset		0	0	0	0	0	0	0
Function		RMTIN start/stop control 0: Stop & counter clear 1: Start	RMTIN input polarity selection 0: Positive (+) 1: Negative (-)	Noise canceller/ correction circuit control 0: Active 1: Bypass	CFG status flag 0: Normal 1: Error	3-bit programmable divider control Dividing rate selection for DFGIN (from P81) 000: 1/1 001: 1/2 -- 111: 1/8		

Capture Input Control Register 3

CAPINCR3 (0049H)

	7	6	5	4	3	2	1	0
bit Symbol		CTLPO	CFGPO	DPGEG	DFGEG	CFGWPR	CFGEG	CTLEG
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0
Function		CTL input polarity selection 0: Positive (+) 1: Negative (-)	CFG input polarity selection 0: Positive (+) 1: Negative (-)	Edge selection for DPG input 0: Rising 1: Falling	Edge selection for DFG input 0: Rising 1: Falling	Dividing rate selection for CFG input 0: 1/1 1: 1/2	Edge selection for CFG input 0: Both rising and falling 1: Rising	Edge selection for CTLIN input 0: Rising 1: Both rising and falling

VISS/VASS Control Register 2

VIVACR2 (0055H)

	7	6	5	4	3	2	1	0
bit Symbol	PCTLPO	PCTLCKS	MSK1	MSK0	VISS3	VISS2	VISS1	VISS0
Read/Write	R/W	R/W	R/W	R/W	R/W			
After reset	0	0	0	0	0	0	0	0
Function	CTL signal polarity selection 0: Positive (+)(when duty is less than 50%, VIVACR1 <CTLDTY> is set to 1) 1: Negative (-)(when duty is less than 50%, VIVACR1 <CTLDTY> is set to 0)	CTL duty measuring clock selection 0: Selected by CAPINCR1 <PCTLCK1, 2> 1: Automatic selection	Dividing control for CTLIN input 0: Divide 1: Bypass	masking control for CFG input 0: Mask 1: Bypass	VISS compare data These 4-bit data are compared with higher 4 bit of the 6-bit CTL counter			

Capture Control Register

CAPCR (0052H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T (CFG)	CAP1T (DFG)	CAPCL	VISFRS	VASFRS	TPRS0	CFGCL	CAFRS
Read/Write	R	R	R/W (Always 0 at reading)					
After reset	0	0	0	0	0	0	0	0
Function	CAP2 trigger input status 0: No trigger input 1: Trigger input	CAP1 trigger input status 0: No trigger input 1: Trigger input	CAP1/ CAP2 status clear 0: — 1: Clear (One-shot)	VISS detect flag clear 0: — 1: Clear (One-shot)	VASS detect flag clear 0: — 1: Clear (One-shot)	TPG0 FIFO counter clear & INTTPG0 disable 0: — 1: Clear & Inhibit (One-shot)	CFG flag clear 0: — 1: Clear (One-shot)	Capture0 FIFO counter/ status clear 0: — 1: Clear (One-shot)

Figure 3.8.3 Registers for CAPIN

3.8.2 Capture 0 (CAP0)

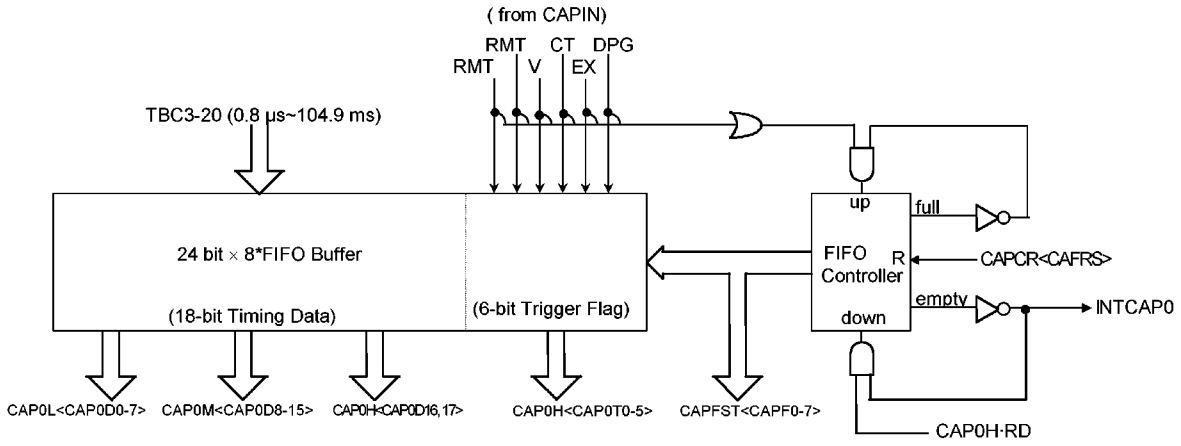


Figure 3.8.4 Structure of Capture 0

(1) Description of Operation

① Capture

The trigger signals for capture 0 are the six signals that are outputted from the capture 0 input control circuit: RMTU, RMTD, VS, CTL, EXT and DPG signals. The CAP0 latches 18 bits of timing data (Output from Time Base Counter: TBC3 to TBC20) and 6 bits of trigger data in 3 bytes of Capture Data Register (Memory address 004BH to 004DH). It latches a total of 24 bits data.

The 24 bits of latched data can be obtained by reading the data from memory addresses 004BH, 004CH and 004DH in this order.

Note1 : By reading data from memory address 004DH, the address of the FIFO buffer is shifted, so please ensure that this data is read last.

Note2 : CAP0M and CAP0H need to read in the continuous read cycle. Therefore, CAP0M and CAP0H need to be read by word operation, or CAPFST, CAP0L, CAP0M and CAP0H need to be read by long-word operation.

```
ex1) LDB  reg. (CAP0L)
     LDW  reg. (CAP0M)  ;  CAP0M/CAP0H word read operation
ex2) LDL  reg. (CAPFST) ;  CAPFST/CAP0L/CAP0M/CAP0H long-word read operation
```

② FIFO Status Register

The 24-bit 8-stage buffer features a FIFO (First In First Out) function, allowing the data latched first to be read first. The FIFO status register (CAPFST) indicates the shift status of FIFO, and the status bits that correspond to the stage being latched are set to 1. When the 8-stage FIFO buffer become full, capture operations are disabled and the FIFO status register shows FFH.

When the FIFO status register shows 00H, in other words the 8-stage FIFO buffer is empty, the reading data of the capture data register is FFH. The capture data can be read out when INTCAP0 interruption has been generated after data has been latched by a trigger signal, or when the FIFO status register is not 00H.

③ Capture Reset

In addition to a system reset function (initialization by resetting), capture 0 also has a software reset function. Software reset is performed by writing 1 at <CAFRS> of the capture control register (CAPCR). In performing a software reset, the following circuits are initialized.

a. The FIFO address counter is addressed to the first stage of the 8-stage FIFO buffer.
b. The FIFO status register (CAPFST) is initialized to 00H.

④ INTCAP0 Interruption

When latch operations are carried out by a trigger input signal, an INTCAP0 interruption request is generated. The INTCAP0 interruption request signal is maintained in an active state until the FIFO status register reaches 00H (empty).

Once INTCAP0 interruption is received, capture data in CAP0L, CAP0M and CAP0H <CAP0D16 to 17> and trigger input data in CAP0H<CAP0T0 to 5> can be read-out and verified the time data and interrupt sources.

INTCAP0 interruption requests are canceled by reading out the FIFO buffer until the content of the FIFO status register reaches 00H or writing 1 to <CAFRS> of CAPCR.

⑤ Data Processing

The upper 6 bits of the 24 bits of data latched by the FIFO buffer are the status data for the trigger signal. The data of the bit that corresponds to the trigger input signal is set to 1, so by reading out this data the input signal can be identified.

Store the latched data in RAM, and by subtracting the data previously stored in the RAM from the data latched by the next trigger signal, highly precise time measurement can be carried out. Detection precision is 400ns when operating at 20 MHz, and quantum error is extremely small.

(2) Control Register

Capture 0 FIFO Status Register

CAPFST (004AH)

	7	6	5	4	3	2	1	0
bit Symbol	CAPF7 (FIFO8)	CAPF6 (FIFO7)	CAPF5 (FIFO6)	CAPF4 (FIFO5)	CAPF3 (FIFO4)	CAPF2 (FIFO3)	CAPF1 (FIFO2)	CAPF0 (FIFO1)
Read/Write	Read-only							
After reset	0	0	0	0	0	0	0	0
Function	Capture0 FIFO status 0: Empty, 1: Capture data stored							

Capture 0 data register-low order

CAP0L (004BH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP0D7 (TBC10)	CAP0D6 (TBC9)	CAP0D5 (TBC8)	CAP0D4 (TBC7)	CAP0D3 (TBC6)	CAP0D2 (TBC5)	CAP0D1 (TBC4)	CAP0D0 (TBC3)
Read/Write	Read-only							
After reset	*	*	*	*	*	*	*	*

* : unknown after reset

Capture 0 data register-middle order

CAP0M (004CH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP0D15 (TBC18)	CAP0D14 (TBC17)	CAP0D13 (TBC16)	CAP0D12 (TBC15)	CAP0D11 (TBC14)	CAP0D10 (TBC13)	CAP0D9 (TBC12)	CAP0D8 (TBC11)
Read/Write	Read-only							
After reset	*	*	*	*	*	*	*	*

* : unknown after reset

Capture 0 data register-high order

CAP0H (004DH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP0T5 (DPG)	CAP0T4 (EXT)	CAP0T3 (CTL)	CAP0T2 (VS)	CAP0T1 (RMTD)	CAP0T0 (RMTU)	CAP0D17 (TBC20)	CAP0D16 (TBC19)
Read/Write	Read-only						Read-only	
After reset	*	*	*	*	*	*	*	*
Function	Trigger input status 0: No trigger input 1: Trigger input						Capture0 higher data	

*: unknown after reset

Capture Control Register

CAPCR (0052H)	7	6	5	4	3	2	1	0
bit Symbol	CAP2T (CFG)	CAP1T (DFG)	CAPCL	VISFRS	VASFRS	TPRS0	CFGCL	CAFRS
Read/Write	R	R	R/W (Always 0 at reading)					
After reset	0	0	0	0	0	0	0	0
Function	CAP2 trigger input status 0: No trigger input 1: Trigger input	CAP1 trigger input status 0: No trigger input 1: Trigger input	CAP1/ CAP2 status clear 0: — 1: Clear (One-shot)	VISS detect flag clear 0: — 1: Clear (One-shot)	VASS detect flag clear 0: — 1: Clear (One-shot)	TPG0 FIFO counter clear & INTTPG0 disable 0: — 1:Clear & Inhibit (One-shot)	CFG flag clear 0: — 1: Clear (One-shot)	Capture0 FIFO counter/ status clear 0: — 1: Clear (One-shot)

Figure 3.8.5 Registers for Capture 0

3.8.3 Capture 1/Capture 2 (CAP1/CAP2)

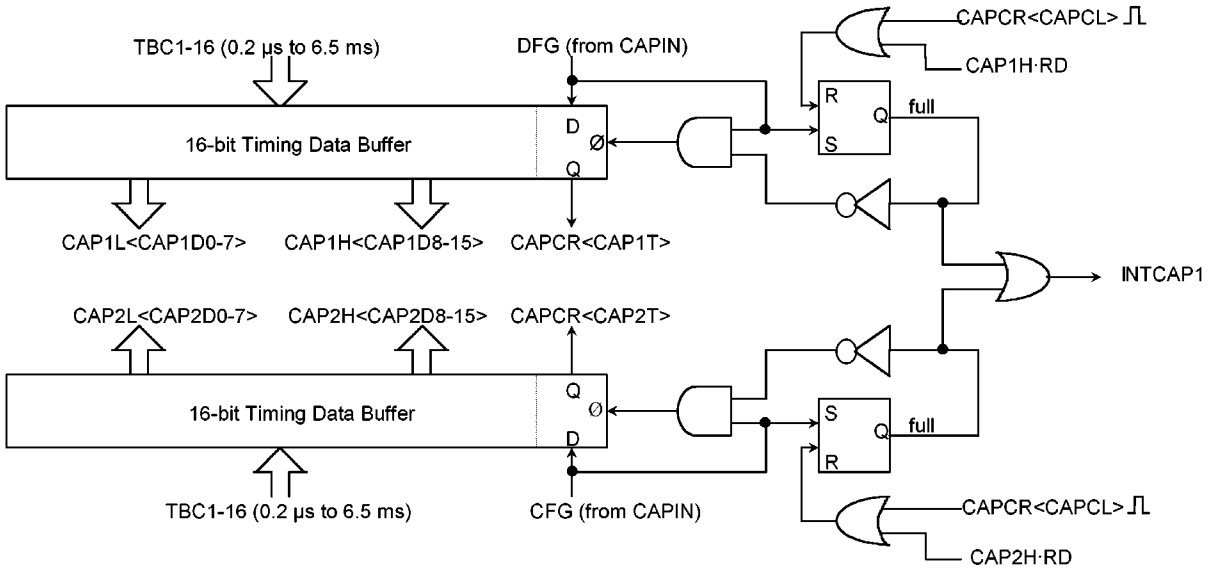


Figure 3.8.6 Structure of Capture 1/Capture 2

(1) Operation

① Capture

The trigger input signals are DFG and CFG signals for capture 1 and capture 2 respectively.

The output values of time base counters (TBC1-TBC16) are latched onto the 2 bytes of capture data register (capture 1 has memory addresses 004EH and 004FH, while capture 2 has memory addresses 0050H and 0051H) using the edge of the trigger input signal.

The trigger input signal is latched onto <CAP1T> and <CAP2T> of the capture control register (CAPCR).

When the CAPCR<CAP1T> is 1, the Capture 1 data can be obtained by reading out the data from the low order of the capture 1 data register (CAP1L) and from the high order capture 1 data register (CAP1H) in this order. By reading out CAP1H, the trigger input status <CAP1T> is cleared. When <CAP1T> is 0 the read-out of CAP1L and CAP1H is FFH.

When the CAPCR<CAP2T> is 1, the Capture 2 data can be obtained by reading out the data from the low order of the capture 2 data register (CAP2L) and from the high order capture 2 data register (CAP2H) in this order. By reading out CAP2H, the trigger input status <CAP2T> is cleared. When <CAP2T> is 0 the read-out of CAP2L and CAP2H is FFH.

② Capture Reset

In capture 1 and 2, trigger input signal <CAP1T> and <CAP2T> can be cleared by writing 1 to <CAPCL> of CAPCR.

③ INTCAP1 Interruption

When latch operations are carried out by the edge of a DFG signal or CFG signal, an INTCAP1 interruption request is generated.

INTCAP1 interruption is common to both capture 1 and capture 2, so please read out <CAP1T> and <CAP2T>, and discriminate between them before carrying out processing.

INTCAP1 interruption requests are released by reading out either CAP1H or CAP2H and clearing <CAP1T> <CAP2T> to 0, or by writing <CAPCL> of CAPCR to 1.

(2) Control Register

Capture 1 data register-low order

CAP1L (004EH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP1D7 (TBC8)	CAP1D6 (TBC7)	CAP1D5 (TBC6)	CAP1D4 (TBC5)	CAP1D3 (TBC4)	CAP1D2 (TBC3)	CAP1D1 (TBC2)	CAP1D0 (TBC1)
Read/Write	Read-only							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

Capture 1 data register-high order

CAP1H (004FH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP1D15 (TBC16)	CAP1D14 (TBC15)	CAP1D13 (TBC14)	CAP1D12 (TBC13)	CAP1D11 (TBC12)	CAP1D10 (TBC11)	CAP1D9 (TBC10)	CAP1D8 (TBC9)
Read/Write	Read-only							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

Capture Control Register

CAPCR (0052H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T (CFG)	CAP1T (DFG)	CAPCL	VISFRS	VASFRS	TPRS0	CFGCL	CAFRS
Read/Write	R	R	R/W (Always 0 at reading)					
After reset	0	0	0	0	0	0	0	0
Function	CAP2 trigger input status 0: No trigger input 1: Trigger input	CAP1 trigger input status 0: No trigger input 1: Trigger input	CAP1/CAP2 status clear 0: — 1: Clear (One-shot)	VISS detect flag clear 0: — 1: Clear (One-shot)	VASS detect flag clear 0: — 1: Clear (One-shot)	TPG0 FIFO counter clear & INTTPG0 disable 0: — 1: Clear & Inhibit (One-shot)	CFG flag clear 0: — 1: Clear (One-shot)	Capture0 FIFO counter/ status clear 0: — 1: Clear (One-shot)

Figure 3.8.7 Registers for Capture 1

Capture 2 data register-low order

CAP2L (0050H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2D7 (TBC8)	CAP2D6 (TBC7)	CAP2D5 (TBC6)	CAP2D4 (TBC5)	CAP2D3 (TBC4)	CAP2D2 (TBC3)	CAP2D1 (TBC2)	CAP2D0 (TBC1)
Read/Write	Read-only							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

Capture 2 data register-high order

CAP2H (0051H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2D15 (TBC16)	CAP2D14 (TBC15)	CAP2D13 (TBC14)	CAP2D12 (TBC13)	CAP2D11 (TBC12)	CAP2D10 (TBC11)	CAP2D9 (TBC10)	CAP2D8 (TBC9)
Read/Write	Read-only							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

Capture Control Register

CAPCR (0052H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T (CFG)	CAP1T (DFG)	CAPCL	VISFRS	VASFRS	TPRS0	CFGCL	CAFRS
Read/Write	R	R	R/W (Always 0 at reading)					
After reset	0	0	0	0	0	0	0	0
Function	CAP2 trigger input status 0: No trigger input 1: Trigger input	CAP1 trigger input status 0: No trigger input 1: Trigger input	CAP1/CAP2 status clear 0: — 1: Clear (One-shot)	VISS detect flag clear 0: — 1: Clear (One-shot)	VASS detect flag clear 0: — 1: Clear (One-shot)	TPG0 FIFO counter clear & INTTPG0 disable 0: — 1: Clear & Inhibit (One-shot)	CFG flag clear 0: — 1: Clear (One-shot)	Capture0 FIFO counter/ status clear 0: — 1: Clear (One-shot)

Figure 3.8.8 Registers for Capture 2

3.9 Remote control signal input circuit (RMTIN)

The remote control signal input circuit (RMTIN) is a circuit which detects the rising and falling edge of the remote control signal element by removing the noise element which the received remote control signal includes. The detected rising signal (RMTU) and falling signal (RMTD) are input to the capture 0(CAP0) through the capture input circuit (CAPIN). The information on the remote control signal can be obtained by measuring those pulse widths with capture 0.

(1) Block diagram

The remote control signal input circuit consists of the H level noise canceller and the L level noise canceller, and each noise canceller is composed by 4-bit up-counter, the comparison/the match detection circuits.

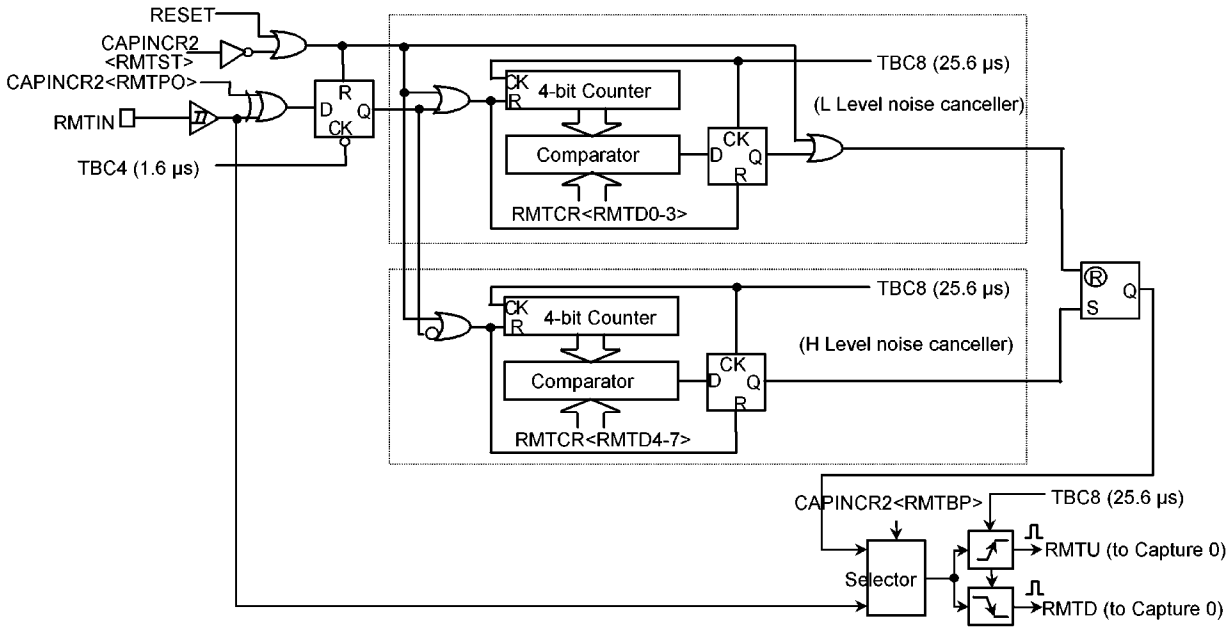


Figure 3.9.1 Composition of the remote control signal input circuit (RMTIN)

(2) Control register

Remote Control Signal Input Control Register

RMTCR (0053H)

	7	6	5	4	3	2	1	0
bit Symbol	RMTD7	RMTD6	RMTD5	RMTD4	RMTD3	RMTD2	RMTD1	RMTD0
Read/Write	R/W				R/W			
	0	0	0	0	0	0	0	0
	H level noise canceller width (TBC8: 25.6 μs) Setting of comparative value of 4-bit counter				L level noise canceller width(TBC8: 25.6 μs) Setting of comparative value of 4-bit counter			

Capture Input Control Register 2

CAPINCR2 (0048H)

	7	6	5	4	3	2	1	0
bit Symbol		RMTST	RMTP0	RMTBP	CFGMCP	DPCP2	DPCP1	DPCP0
Read/Write		R/W	R/W	R/W	R	R/W		
After reset		0	0	0	0	0		
Function		RMTIN start/stop control 0: Stop & counter clear 1: Start	RMTIN input polarity selection 0: Positive (+) 1: Negative (-)	Noise canceller/ correction circuit control 0: Active 1: Bypass	CFG status flag 0: Normal 1: Error	3-bit programmable divider control Dividing rate selection for DFGIN (from P81) 000: 1/1 001: 1/2 -- 111: 1/8		

Figure 3.9.2 Register for RMTIN

(3) Operation of the remote control signal input circuit

The remote control signal input circuit begins operating by setting < RMTST > of the capture input control register 2 (CAPINCR2) to 1.

Setting the noise removal width (RMTCR<RMTD7-RMTD0>) and the input polarity (CAPINCR2<RMTPO>) before beginning of the operation is necessary.

① Control of remote control signal input

The polarity of the remote control signal input from the RMTIN (P82) pin can be switched by < RMTPO > of the capture input control register 2 (CAPINCR2).

The remote control signal is input to the noise canceller circuit after being sampled by the falling edge of TBC4 of the time base counter (TBC) output (1.6 us at fc = 20 MHz).

The operation when assuming <RMTPO>= 0 (positive polarity) is explained at the following.

② Operation of the noise canceller

The noise canceller removes the pulse of less than width set with RMTCR<RMTD7-RMTD4> and <RMTD3-RMTD0> as the noise.

As for the L level noise canceller, the 4-bit counter is counted up by TBC8 of the TBC output (It is 25.6 us at 20 MHz) while the remote control signal is on the L level. Moreover, the 4-bit counter is cleared for the period at the H level. When the count value matches to the value of RMTCR<RMTD3-RMTD0>, the L level noise canceller regards this state as the falling edge of the remote control signal and resets the flip-flop.

As for the H level noise canceller, the 4-bit counter is counted up by TBC8 of the TBC output (It is 25.6 us at 20 MHz) while the remote control signal is on the H level. Moreover, the 4-bit counter is cleared for the period at the L level. When the count value matches to the value of RMTCR<RMTD7-RMTD4>, the H level noise canceller regards this state as the rising edge of the remote control signal and sets the flip-flop.

③ Measurement of the pulse width for the remote control signal

The flip-flop output obtained from the L level noise canceller and the H level noise canceller is a signal which the noise is removed from the input remote control signal.

As for this flip-flop output, the rising edge is assumed to be RMTU signal, the falling edge is assumed to be

RMTD signal, and the RMTU and RMTD signals are input to the capture 0 (CAP0) through the capture input control circuit (CAPIN).

The pulse width of the remote control signal can be calculated by the operation of the RMTU time data and the RMTD time data.

When the RMTCR<RMTD7-RMTD4> or <RMTD3-RMTD0> is set to 0H, the noise removal operation is not done. Moreover, the noise canceller can be bypassed by setting CAPINCR2<RMTBP > to 1.

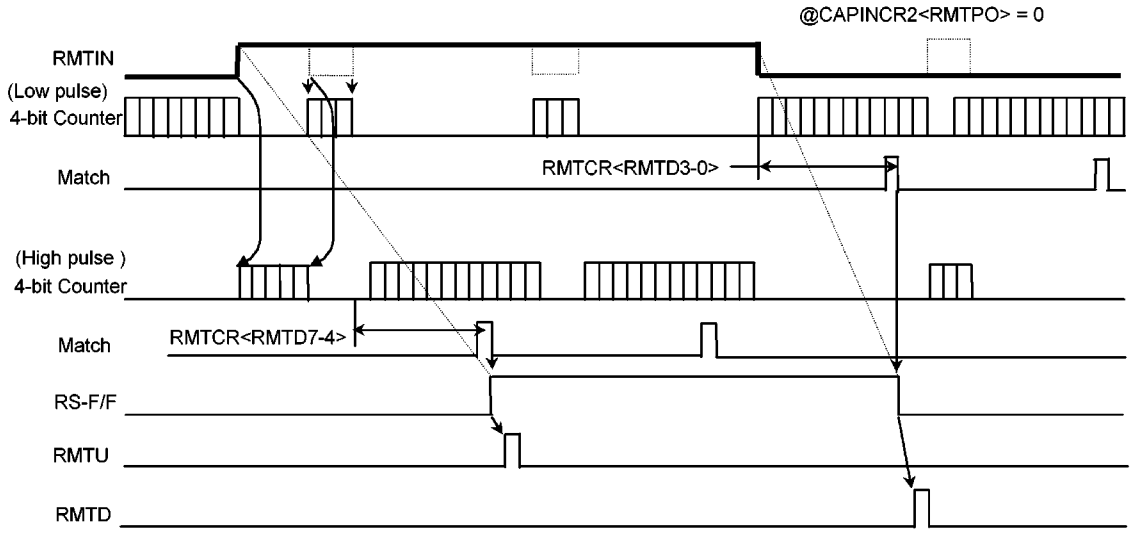


Figure 3.9.3 Timing chart of the remote control signal input circuit

3.10 TIMING PULSE GENERATOR (TPG)

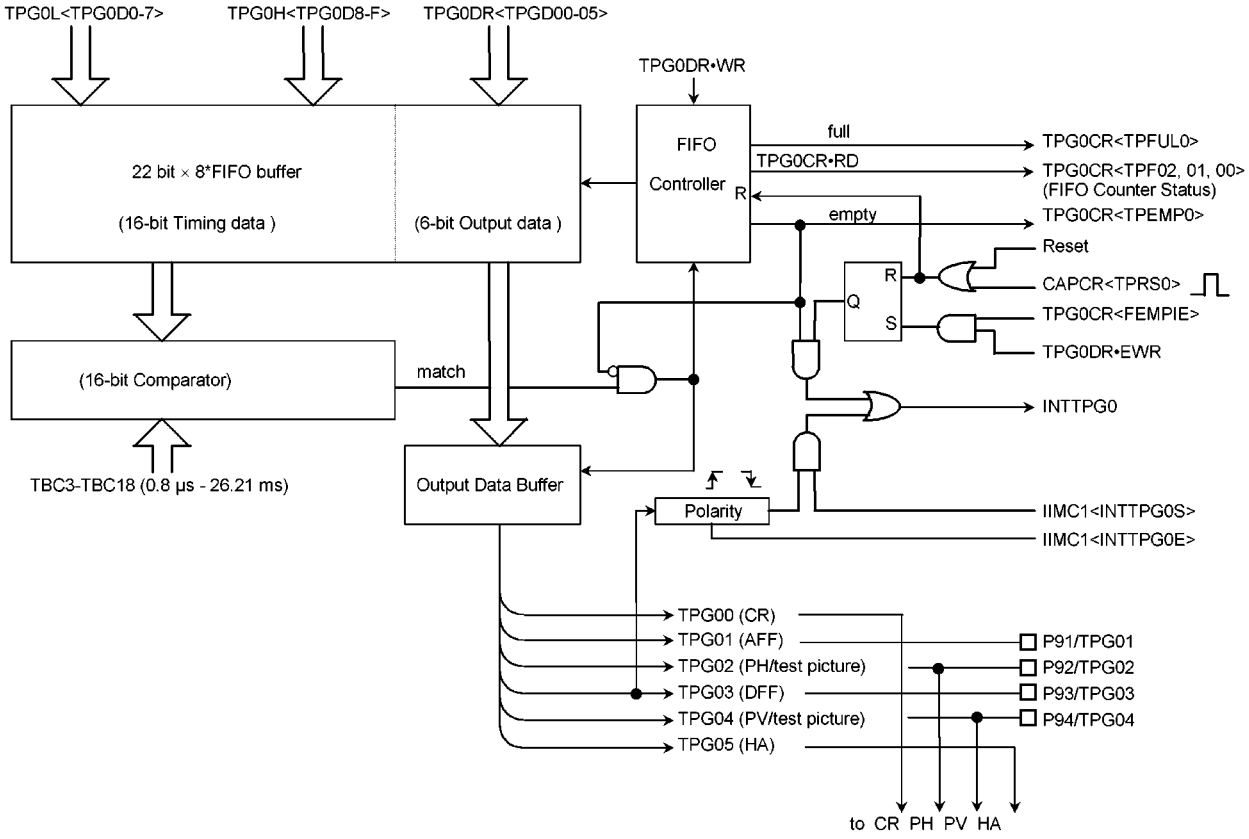
In order to generate the various timing pulses necessary for VTR system control, the TMP93C071 has a timing pulse generator (TPG0) with 22-bit 8-stage FIFO buffer and a 20-bit timing pulse generator (TPG1).

The TPG0 and TPG1 can output the timing pulse synchronized with the time base counter (TBC). Their resolution for both TPG0 and TPG1 is 400 ns (when operating at 20 MHz).

3.10.1 Timing Pulse Generator 0 (TPG0)

(1) Configuration

The TPG0, as shown in Figure 3.10.1, is composed of 22-bit 8-stage FIFO data register (16-bit timing data + 6-bit output data), 16-bit comparator, 6-bit output data buffer and FIFO control circuit.



(2) Operation

① 22-bit 8-stage FIFO buffer

This is a 22-bit data register which is composed of 16-bit timing data and 6-bit output data. As this register has a 8-stage FIFO structure, the first written timing data and output data are transferred first to the comparator and output data register.

Set to the lower timing data register (TPG0L), the higher timing data register (TPG0H) and the output data register (TPG0DR) in this order. The FIFO address is incremented by writing of TPG0DR.

6-bit output data is transferred to the output data buffer when 16-bit timing data matches the value of TBC3 to TBC18. And they become the TPG00 to TPG05.

② 16-bit comparator

When the set data of the TPG0L and TPG0H matches the value of TBC3 to TBC18, the comparator outputs the match signal. The value of TPG0DR is transferred to output data buffer and the FIFO address is incremented by the match signal.

③ Output data buffer

The data set in the output data register (TPG0DR) is latched by the match signal from the 16-bit comparator, and TPG00 to TPG05 outputs are changed. When resetting, this buffer is cleared to 0 and TPG00 to TPG05 outputs become 0.

④ FIFO control circuit

The FIFO control circuit controls the 22-bit 4-stage FIFO buffer and has a status flag to monitor the FIFO address.

The current number of retained data can be verified by reading out the FIFO status flags <TPF02, TPF01, TPF00> of the TPG0 control register (TPG0CR). In case that the value of the FIFO status flag is 000, the FIFO empty flag <TPEMP0> is set to 1 when retained data is nothing and the FIFO full flag <TPFUL0> is set to 1 when retained data are 8 words. And, writing data to the FIFO buffer is disabled while the <TPFUL0> is set to 1. The contents of the FIFO status flags is varied each time the match signal is outputted from the comparator. The contents of the FIFO status flags is cleared to 000 by resetting. In addition, the FIFO address can be cleared by writing 1 to <TPRS0> of the capture control register (CAPCR).

⑤ Timing pulse generator 0 interrupt (INTTPG0)

When the contents of the FIFO buffer becomes empty, an INTTPG0 (empty) interrupt to request the writing of the next data is generated. The INTTPG0 (empty) interrupt request can be controlled by TPG0CR <FEMPIE> and CAPCR <TPRS0>. By setting <FEMPIE> to "1", INTTPG0 (empty) interrupt request is enabled by writing of TPG0 output data register (TPG0DR). And by writing "1" to CAPCR <TPRS0>, it can be disabled (in this case FIFO address will also be cleared).

In addition, INTTPG0 (TPG03) interrupt request can be generated synchronized with the rising edge or falling edge of TPG03. Either rising or falling edge of TPG03 can be selected by <INTTPG0E> of interrupt input mode control register (IIMC1). To enable or disable INTTPG0 (TPG03) interrupt request can be selected using IIMC1 <INTTPG0S>.

INTTPG0 interrupt request is generated as logical-OR with INTTPG0 (empty) interrupt request and INTTPG0 (TPG03) interrupt request.

(3) Control Register

TPG0 Control Register

TPG0CR (0036H)

	7	6	5	4	3	2	1	0
bit symbol			FEMPIE	TPFUL0	TPEMP0	TPF02	TPF01	TPF00
Read/Write			W	R	R	R		
After reset	0		0	0	1	0	0	0
Function	Always set to 0		INTTPG0 FIFO empty interrupt enable 0: — 1: Enable	TPG0 FIFO full flag 0: — 1:FIFO full	TPG0 FIFO empty 0: — 1: FIFO empty	TPG0 FIFO status flag (Stored data) 000:Empty or Full 001: 1 word 010: 2 word 011: 3 word 100: 4 word 101: 5 word 110: 6 word 111: 7 word		

TPG0 lower timing data register

TPG0L (0037H)

	7	6	5	4	3	2	1	0
bit Symbol	TPG0D7 (TBC10)	TPG0D6 (TBC9)	TPG0D5 (TBC8)	TPG0D4 (TBC7)	TPG0D3 (TBC6)	TPG0D2 (TBC5)	TPG0D1 (TBC4)	TPG0D0 (TBC3)
Read/Write	Write-only							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

TPG0 higher timing data register

TPG0H (0038H)

	7	6	5	4	3	2	1	0
bit Symbol	TPG0DF (TBC18)	TPG0DE (TBC17)	TPG0DD (TBC16)	TPG0DC (TBC15)	TPG0DB (TBC14)	TPG0DA (TBC13)	TPG0D9 (TBC12)	TPG0D8 (TBC11)
Read/Write	Write-only							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

TPG0 output data register

TPG0DR (0039H)

	7	6	5	4	3	2	1	0
bit Symbol			TPGD05 (TPG05)	TPGD04 (TPG04)	TPGD03 (TPG03)	TPGD02 (TPG02)	TPGD01 (TPG01)	TPGD00 (TPG00)
Read/Write	Write-only							
After reset			*	*	*	*	*	*

*: unknown after reset

Capture control register

CAPCR (0052H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T (CFG)	CAP1T (DFG)	CAPCL	VISFRS	VASFRS	TPRS0	CFGCL	CAFRS
Read/Write	R	R	R/W (Always "0" at reading)					
After reset	0	0	0	0	0	0	0	0
Function	CAP2 trigger input status 0: No trigger input 1: Trigger input	CAP1 trigger input status 0: No trigger input 1: Trigger input	CAP1/ CAP2 status clear 0: — 1: Clear (One-shot)	VISS detect flag clear 0: — 1: Clear (One-shot)	VASS detect flag clear 0: — 1:Clear (One-shot)	TPG0 FIFO counter clear & INTTPG0 disable 0: — 1: Clear & Inhibit (One-shot)	CFG flag clear 0: — 1: Clear (One-shot)	Capture0 FIFO counter/ status clear 0: — 1: Clear (One-shot)

Interrupt input mode control register 1

IIMC1 (005FH)

	7	6	5	4	3	2	1	0
bit Symbol	I4EG	I3EG	I2EG	I1EG	I0EG		INTTPG0E	INTTPG0S
Read/Write	W						R/W	R/W
After reset	0	0	0	0	0		0	0
Function	INT4 edge selection 0: Rising 1: Falling	INT3 edge selection 0: Rising 1: Falling	INT2 edge selection 0: Rising 1: Falling	INT1 edge/level selection 0: Rising edge 1: Level	INT0 edge/level selection 0: Rising edge 1: Level		INTTPG0 interrupt TPG03 edge selection 0: Rising 1: Falling	INTTPG0 interrupt selection 0: FIFO empty interrupt 1: FIFO empty/ TPG03 interrupt

Prohibit read-modify-write

Figure 3.10.2 Registers for TPG0

(4) Output of timing pulse generator 0

① TPG00

TPG00 controls the color rotary output (CR).
(Refer to 3.13 Head Amplifier (HA) /Color Rotary (CR) Control Circuit.)

② TPG01

TPG01 can be used as AFF (audio head switching) signal.

③ TPG02

TPG02 controls the pseudo synchronizing signal output (PV).
(Refer to 3.15 Pseudo Synchronizing Signal Output Circuit.)

④ TPG03

TPG03 can be used as DFF (cylinder head switching) signal.

⑤ TPG04

TPG04 controls the pseudo synchronizing signal output (PV).
(Refer to 3.15 Pseudo Synchronizing Signal Output Circuit.

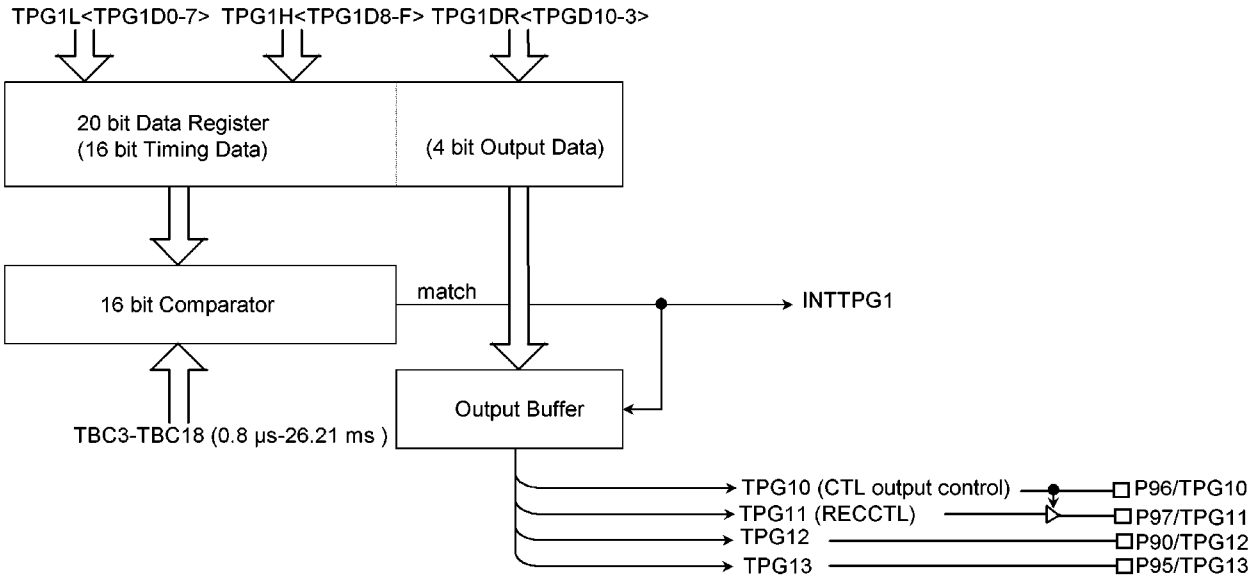
⑥ TPG05

TPG05 controls the head amp (HA) output.
(Refer to 3.13 Head Amplifier (HA) /Color Rotary (CR) Control Circuit.)

3.10.2 Timing Pulse Generator (TPG1)

(1) Configuration

TPG1 consist of a 20-bit data register (16-bit timing data + 4-bit output data), 16-bit comparator and 4-bit output data buffer.



(2) Operation

① 20-bit data register

This is a 20-bit data register which is composed of 16-bit timing data and 4-bit output data. Set to the lower timing data register (TPG1L), higher timing data register (TPG1H), and the output data register (TPG0DR). 4-bit output data is transferred to the output data buffer when 16-bit timing data matches the value of TBC3 to TBC18. And they become the TPG10 to TPG13.

② 16-bit comparator

When the set data of the TPG1L and TPG1H matches the value of TBC3 to TBC18, the comparator outputs the match signal. The value of the TPG1DR is transferred to output data buffer and INTTPG1 interrupt request is generated by the match signal.

③ Output data buffer

The data set in the output data register (TPG1DR) is latched by the match signal from the 16-bit comparator, and TPG00 to TPG05 are outputted. When resetting, this buffer is cleared to 0 and TPG10 to TPG13 output become 0.

(3) Control register

TPG1 lower timing data register

TPG1L (003AH)

	7	6	5	4	3	2	1	0
bit Symbol	TPG1D7 (TBC10)	TPG1D6 (TBC9)	TPG1D5 (TBC8)	TPG1D4 (TBC7)	TPG1D3 (TBC6)	TPG1D2 (TBC5)	TPG1D1 (TBC4)	TPG1D0 (TBC3)
Read/Write	Write-only							
After reset	0	0	0	0	0	0	0	0

TPG1 higher timing data register

TPG1H (003BH)

	7	6	5	4	3	2	1	0
bit Symbol	TPG1DF (TBC18)	TPG1DE (TBC17)	TPG1DD (TBC16)	TPG1DC (TBC15)	TPG1DB (TBC14)	TPG1DA (TBC13)	TPG1D9 (TBC12)	TPG1D8 (TBC11)
Read/Write	Write-only							
After reset	0	0	0	0	0	0	0	0

TPG1 output data register

TPG1DR (003CH)

	7	6	5	4	3	2	1	0
bit Symbol					TPGD13 (TPG13)	TPGD12 (TPG12)	TPGD11 (TPG11)	TPGD10 (TPG10)
Read/Write					Write-only			
After reset					0	0	0	0

Figure 3.10.3 Registers for TPG1

(4) Output of timing pulse generator 1

① TPG10

TPG10 controls the recording amplifier (external IC) for CTL signal.

② TPG11

TPG11 controls the recording amplifier (external IC) for CTL signal.

③ TPG12

TPG12 can be output from TPG12 (P90) pin.

④ TPG13

TPG13 can be output from TPG13 (P95) pin.

3.11 Pulse Width Modulation output(PWM)

TMP93C071F has the pulse width modulation output with 14-bit resolution of 3 channels (PWM0, PWM1, PWM2) and with 8-bit resolution of 9 channels (PWM3-PWM11).

It is possible to use it to apply the DC motor control for the drum and capstan, the tuner control (voltage synthesize method) , and the volume control, etc. by turning on the low-pass filter outside.

3.11.1 8-bit PWM (PWM3-PWM11)

PWM3 to PWM11 are the 8-bit PWM output of 9 channels. These are controlled by the PWM control register (PWMCR) and the PWM data buffer register (PWMDBR).

(1) Block diagram

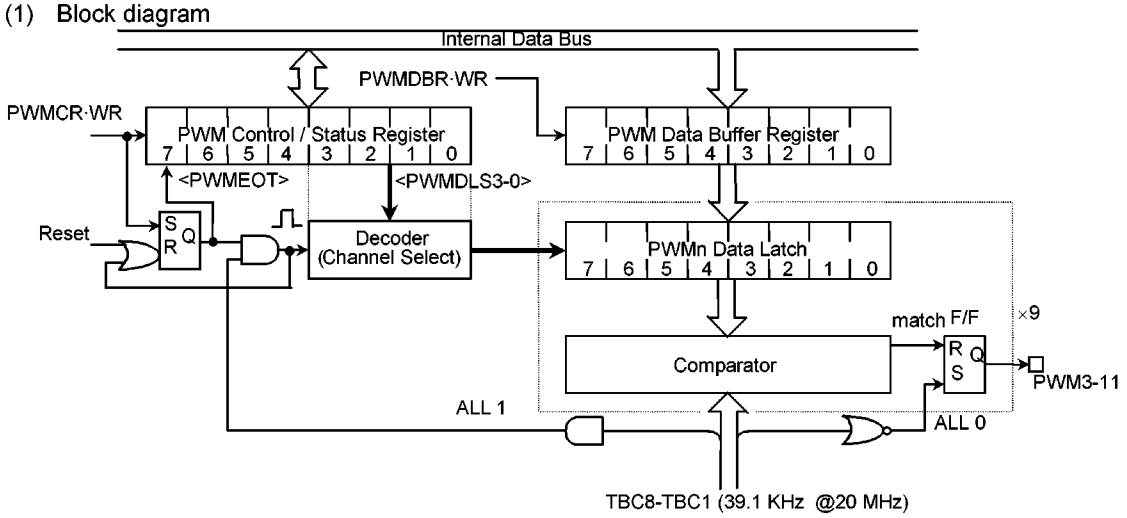


Figure 3.11.1 8-bit PWM output (PWM3 to PWM11)

(2) Control of PWM output

PWM3 to PWM11 are the pulse output with 8-bit resolution and the one cycle time is $T_M = 2^8/(fc/2)$ s (25.6 us/39.06 kHz at 20 MHz operation). When the value of the 8-bit PWM data latch is n, the H level pulse width becomes n × t0 (t0 = 1/(fc/2)) [s].

The PWM control register (PWMCR) and the PWM data buffer register (PWMDBR) are shared with all channels.

The data set in PWMDBR is transferred to the data latch of the channel selected by PWMCR.

Moreover, It can be known the state of the PWM data transfer from the data buffer register to the data latch by reading PWMCR<PWMEOT >.

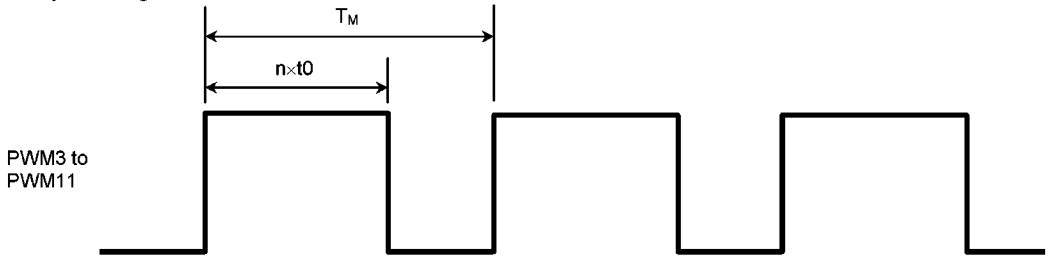


Figure 3.11.2 Wave form of the 8-bit PWM output

PWM data buffer register

PWMDBR (0044H)

	7	6	5	4	3	2	1	0
bit Symbol	PWMDBR7	PWMDBR6	PWMDBR5	PWMDBR4	PWMDBR3	PWMDBR2	PWMDBR1	PWMDBR0
Read/Write	Write-only							
After reset	*	*	*	*	*	*	*	*

* : unknown after reset

PWM control register

PWMCR (0045H)

	7	6	5	4	3	2	1	0
bit Symbol	PWMEOT				PWMDLS3	PWMDLS2	PWMDLS1	PWMDLS0
Read/Write	R				Write-only			
After reset	0				0			
Function	End flag of PWM data transfer 0: End of transfer 1: Under transfer				PWM channel select and data transfer The PWMDBR data is stored to the selected PWM channel after setting this channel code. This code must be set when <PWMEOT> = 0 0000: PWM3 0100: PWM7 1000: PWM11 0001: PWM4 0101: PWM8 0010: PWM5 0110: PWM9 0011: PWM6 0111: PWM10			

Figure 3.11.3 Registers for the 8-bit PWM

(3) Writing of the PWM output data

The PWM output is controlled by writing the output data in the data latch.

The output data is written by using PWMCR<PWMDLS3-0>, <PWMEOT> and PWMDBR according to the following procedure.

1. It is confirmed that it is PWMCR<PWMEOT> = 0.
2. The output data is written in PWMDBR.
3. By writing the channel code to PWMCR<PWMDLS3-0>, the data transfer demand is generated at the same time as selecting the channel, and PWMCR<PWMEOT> becomes 1.
4. The output data is transferred to the selected channel according to the timing that (TBC8 to TBC0) of the time base counter (TBC) becomes FFH, and PWMCR<PWMEOT> is cleared to 0 at the same time. When PWMCR<PWMEOT > becomes 0, the next data can be written.

Note: Please rewrite PWMDBR at PWMCR<PWMEOT> = 0.

3.11.2 14-bit PWM (PWM0 to PWM2)

The 14-bit PWM are controlled by the data register (PWM0DRL/PWM0DRH, PWM1DRL/PWM1DRH, PWM2DRL/PWM2DRH) and the PWM start register (PWMRUN).

(1)Block diagram

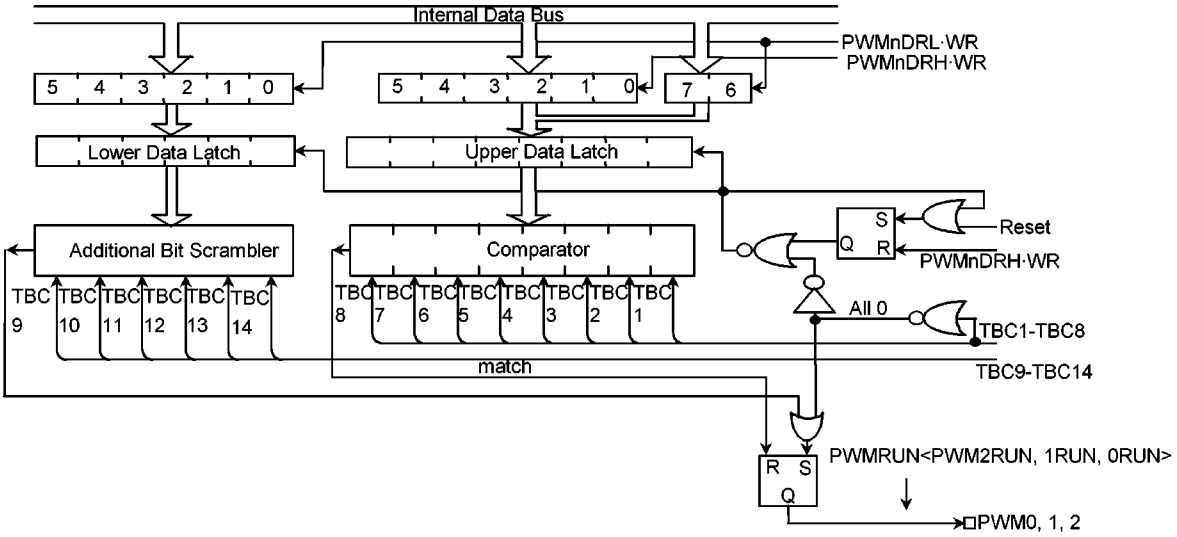


Figure 3.11.4 14-bit PWM output (PWM0 to PWM2)

(2) Control of PWM output

The PWM0 to PWM2 output are the pulse output with 14-bit resolution and the one cycle time is $T_M = 2^{14}/(fc/2)$ s (1.638 ms at 20 MHz operation).

The carrier frequency is $T_S = T_M/64$ [s] (25.6 us/39.06 kHz at 20 MHz operation).

The data register to set the pulse width is composed of 14-bit (PWM0DRL/PWM0DRH (PWM0), PWM1DRL/PWM1DRH (PWM1), and PWM2DRL/PWM2DRH (PWM2)) respectively.

It is necessary to write the data in the data register in order of the lower-data register (PWM0DRL, PWM1DRL, PWM2DRL) → the higher-data register (PWM0DRH, PWM1DRH, PWM2DRH).

The data changes when the following T_S cycle is begun after writing in the higher-data register.

When the value of the upper data latch is "n" and the value of the lower-data latch is "m", the carrier pulse width is "n × t0 (t0=1/(fc/2))[s]", the additional pulse with "t0 (t0=1/(fc/2))[s]" is added at the "m" points of the carrier pulses which output for the one T_M cycle. Therefore, the width of the carrier pulse with the additional pulse becomes "(n+1) × t0[s]". (When the value of 14-bit data register is increased by "k", the total pulse width is increased by "(k × t0)" for the one T_M cycle.)

The PWM0 to PWM2 begins outputting by setting PWMRUN<PWM0RUN, PWM1RUN, PWM2RUN > to 1.

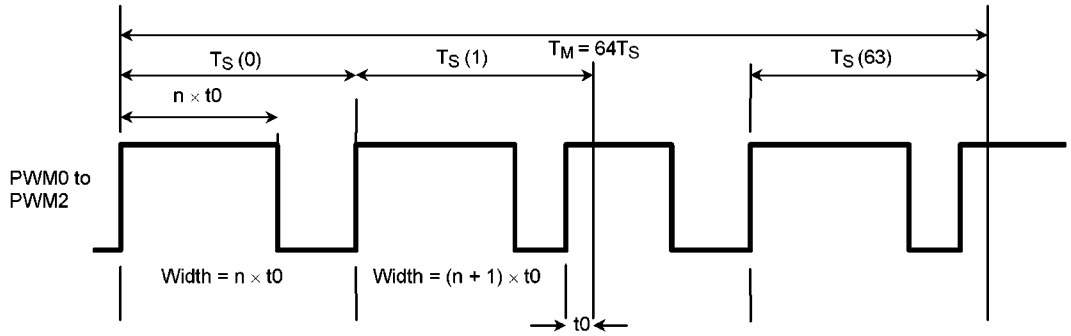


Figure 3.11.5 Wave form of the 14-bit PWM output

PWM start register

PWMRUN (003DH)

	7	6	5	4	3	2	1	0
bit Symbol						PWM2RUN	PWM1RUN	PWM0RUN
Read/Write						R/W		
After reset						0	0	0
Function						0: Stop 1: Start		

PWM0 lower data register

PWM0DRL (003EH)

	7	6	5	4	3	2	1	0
bit Symbol	PWM0D7	PWM0D6	PWM0D5	PWM0D4	PWM0D3	PWM0D2	PWM0D1	PWM0D0
Read/Write	Write-only							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

PWM0 higher data register

PWM0DRH (003FH)

	7	6	5	4	3	2	1	0
bit Symbol			PWM0DD	PWM0DC	PWM0DB	PWM0DA	PWM0D9	PWM0D8
Read/Write			Write-only					
After reset			*	*	*	*	*	*

*: unknown after reset

PWM1 lower data register

PWM1DRL (0040H)

	7	6	5	4	3	2	1	0
bit Symbol	PWM1D7	PWM1D6	PWM1D5	PWM1D4	PWM1D3	PWM1D2	PWM1D1	PWM1D0
Read/Write	Write-only							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

PWM1 higher data register

PWM1DRH (0041H)

	7	6	5	4	3	2	1	0
bit Symbol			PWM1DD	PWM1DC	PWM1DB	PWM1DA	PWM1D9	PWM1D8
Read/Write			Write-only					
After reset			*	*	*	*	*	*

*: unknown after reset

PWM2 lower data register

PWM2DRL (0042H)

	7	6	5	4	3	2	1	0
bit Symbol	PWM2D7	PWM2D6	PWM2D5	PWM2D4	PWM2D3	PWM2D2	PWM2D1	PWM2D0
Read/Write	Write-only							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

PWM2 higher data register

PWM2DRH (0043H)

	7	6	5	4	3	2	1	0
bit Symbol			PWM2DD	PWM2DC	PWM2DB	PWM2DA	PWM2D9	PWM2D8
Read/Write			Write-only					
After reset			*	*	*	*	*	*

*: unknown after reset

Open drain control register 3

ODCR3 (0021H)

	7	6	5	4	3	2	1	0
bit Symbol				PWMOD1 (PWM1)	PWMOD0 (PWM0)	PODA5 (PortA5)	PODA4 (PortA4)	PODA3 (PortA3)
Read/Write				Write-only		Write-only		
After reset				0	0	0	0	0
Function				PWM0, 1 Open drain control 0: Push-pull 1: Open drain		Port A Open drain control 0: Push-pull 1: Open drain		

Figure 3.11.6 Registers for 14-bit PWM

3.12 VISS / VASS DETECTOR (VIVA)

This circuit is to support the Video Index Search System (VISS) and Video Address Search System (VASS) for VHS VCR. By using this circuit, the duty of control signal (CTL) recorded on video tapes is can be measured and the VISS code can be detected. Further, the address code of VASS can be read out.

3.12.1 Configuration

VISS / VASS detector consists of the CTL duty discrimination circuit, VISS detection circuit, VASS header detection circuit and 16-bit address code register.

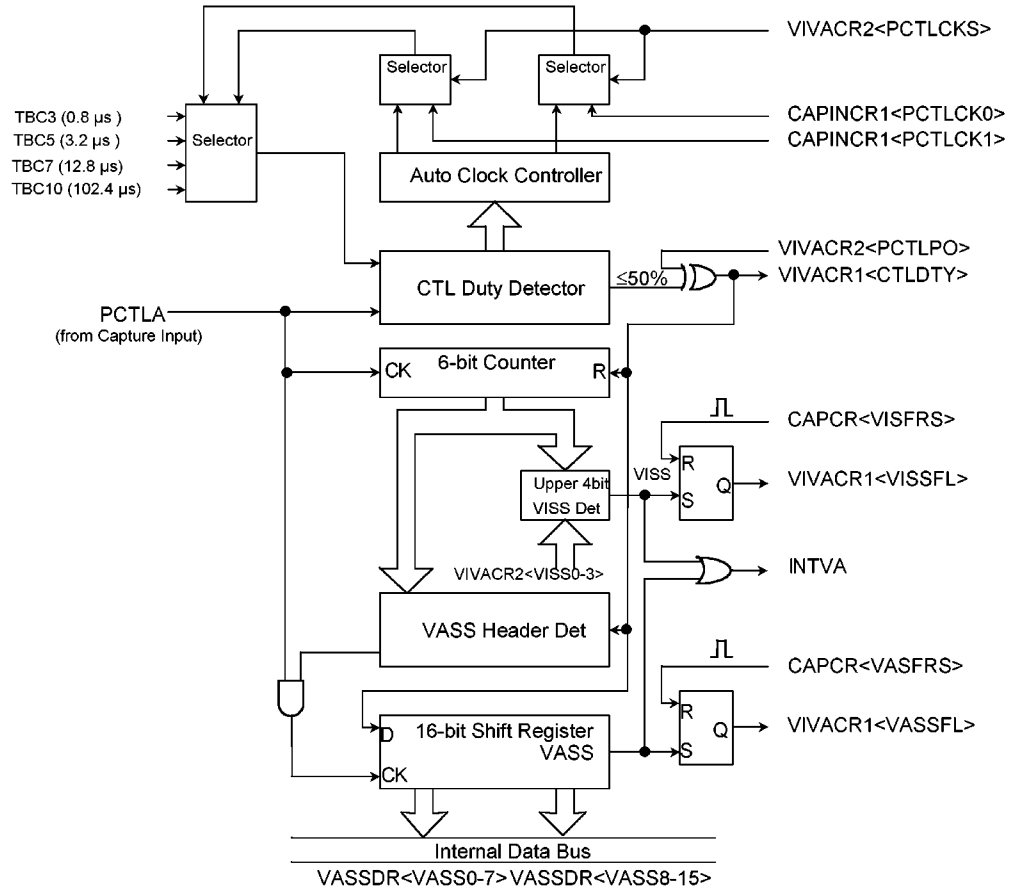


Figure 3.12.1 VISS/VASS detector

3.12.2 Control Registers

VISS/VASS control register 1

VIVACR1 (0054H)

	7	6	5	4	3	2	1	0
bit Symbol						CTLDTY	VISSFL	VASSFL
Read/Write						R	R	R
After reset						0	0	0
Function	Always set to 0					CTL signal duty detector monitor flag 0: CTL duty greater than 50% (VIVACR2 <PCTLPO> = 0) 1: CTL duty less than 50% (VIVACR2 <PCTLPO> = 0)	VISS detect flag 0: — 1: VISS detected	VASS detect flag 0: — 1: VASS detected

VISS/VASS control register 2

VIVACR2 (0055H)

	7	6	5	4	3	2	1	0
bit Symbol	PCTLPO	PCTLCKS	MSK1	MSK0	VISS3	VISS2	VISS1	VISS0
Read/Write	R/W	R/W	R/W	R/W	R/W			
After reset	0	0	0	0	0	0	0	0
Function	CTL signal polarity selection 0: Positive (+) (when duty is less than 50%, VIVACR1 <CTLDTY> is set to 1) 1: Negative (-) (when duty is less than 50%, VIVACR1 <CTLDTY> is set to 0)	CTL duty measuring clock selection 0: Selected by CAPINCR1 <PCTLCK 1, 2> 1: Automatic selection	Dividing control for CTLIN input 0: Divide 1: Bypass	masking control for CFG input 0: Mask 1: Bypass	VISS compare data These 4-bit data are compared with higher 4 bit of the 6-bit CTL counter			

Capture input control register 1

CAPINCR1 (0047H)

	7	6	5	4	3	2	1	0
bit symbol	PCTLCK1	PCTLCK0	PCPR5	PCPR4	PCPR3	PCPR2	PCPR1	PCPR0
Read/Write	R/W		R/W					
After reset	0	0	0	0	0	0	0	0
Function	CTL duty measure clock select (VIVACR2<PCTLCKS> = 0) 00: TBC3 01: TBC5 10: TBC7 11: TBC10		6-bit programmable divider control Dividing rate selection for CTLIN (from P80) and CFGIN (from P85). 000000: 1/1 000001: 1/2 --- 111111: 1/64					

Capture control register

CAPCR (0052H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP2T (CFG)	CAP1T (DFG)	CAPCL	VISFRS	VASFRS	TPRS0	CFGCL	CAFRS
Read/Write	R	R	R/W (Always 0 at reading)					
After reset	0	0	0	0	0	0	0	0
Function	CAP2 trigger input status 0: No trigger input 1: Trigger input	CAP1 trigger input status 0: No trigger input 1:Trigger input	CAP1/ CAP2 status clear 0: — 1: Clear (One-shot)	VISS detect flag clear 0: — 1: Clear (One-shot)	VASS detect flag clear 0: — 1: Clear (One-shot)	TPG0 FIFO counter clear & INTTPG0 disable 0: — 1:Clear & Inhibit (One-shot)	CFG flag clear 0: — 1: Clear (One-shot)	Capture0 FIFO counter/ status clear 0: — 1: Clear (One-shot)

VASS data register

VASSDR (0056H)

	7	6	5	4	3	2	1	0
bit Symbol	VASS7	VASS6	VASS5	VASS4	VASS3	VASS2	VASS1	VASS0
	VASS15	VASS14	VASS13	VASS12	VASS11	VASS10	VASS9	VASS8
Read/Write	Read-only							
After reset	*	*	*	*	*	*	*	*
Function	Twice reading from VASSDR The First is lower 8-bit, the second is higher 8-bit							

*: unknown after reset

Figure 3.12.2 Registers for VISS/VASS

3.12.3 Control of the CTL duty discrimination circuit

(1) Control of CTL duty discrimination

The CTL signal which is inputted from CTLIN pin is inputted, as a PCTLA signal, to VISS/VASS detector (VIVA) via the Capture input control circuit (CAPIN). The CTL duty discrimination circuit judges by threshold; the threshold has half the time of PCTLA signal term. If duty of the PCTLA signal ≧ 50%, the CTL duty discrimination circuit outputs 0. And if its duty ≦ 50%, it outputs 1. This output can be read as <CTLDTY> of the VISS/VASS control register 1 (VIVACR1). And, the polarity of output can be inverted by using <PCTLPO> of the VISS/VASS control register 2 (VIVACR2).

(2) Control of Clock switching

The Clock source, which is to measure CTL signal term, is selected by software or by hardware. If its clock source is selected by software, it is required to set <PCTLCK0> and <PCTLCK1> in Capture Input Control Register 1 (CAPINCR1) to connect adequate TBC while <PCTLCKS> in VIVACR2 is 0. On the other hand, TBC is selected automatically while <PCTLCKS> is 1.

The following table indicates the relation between the term of CTL and its clock source.

Table 3.12.1 Relation between the CTL signal term and its measuring clock source

<PCTCK1> <PCTLCK0>	Clock source for CTL duty discrimination (at fc = 20 MHz)	Term of CTL signal
00	TBC3 (0.8 μs)	1221 to 25000 Hz
01	TBC5 (3.2 μs)	305 to 6250 Hz
10	TBC7 (12.8 μs)	76 to 1562 Hz
11	TBC10 (102.4 μs)	9 to 195 Hz

3.12.4 Control of VISS detection

VISS detection circuit consists of 6-bit up counter counts PCTLA signal, comparator for detecting VISS index code, and R/S flip-flop (VISS flag).

Since CTL duty discriminating output is "L" active reset input for 6-bit counter, 6-bit counter is held on count operation while CTL duty discriminating output is "H" (in case that <PCTLPO> = 0). The upper 4bits in 6-bit counter are compared with value of VISS detection circuit which is a 4-bit comparator. The data on the comparator is set by <VISS3> to <VISS0> of VISS/VASS Control Register 2 (VIVACR2). VISS signal is outputted when the upper 4bits in 6-bit counter and the data on <VISS3> to <VISS0> are matched; the matching sets the VISS flag and requests INTVA interrupt. The state of VISS flag can be read out by <VISSFL> of VISS/VASS Control Register 1 (VIVACR1), and the VISS flag can be cleared by using <VISFRS> of the Capture control register (CAPCR). Table 3.7 2 shows relation between the value of <VISS3> to <VISS0> and count value of the PCTLA signal.

In order to detect VISS, set VIVACR2 <PCTLPO> to 0 when tape operates forward and set VIVACR2 <PCTLPO> to 1 when it operates reversely.

Table 3.12.2 Relation between the value of <VISS3> to <VISS0> and count value of the PCTLA signal

<VISS3>to <VISS0>				The count value for detecting VISS index code
3	2	1	0	
0	0	0	0	don't use
0	0	0	1	4
0	0	1	0	8
0	0	1	1	12
0	1	0	0	16
0	1	0	1	20
0	1	1	0	24
0	1	1	1	28
1	0	0	0	32
1	0	0	1	36
1	0	1	0	40
1	0	1	1	44
1	1	0	0	48
1	1	0	1	52
1	1	1	0	56
1	1	1	1	60

3.12.5 Control of VASS detection circuit

VASS detector consists of the header detection circuit and address code register (16-bit shift register). The header detection circuit detects the CTL duty discriminating output status that begins with "0" (6-bit counter is reset) before 9-bits of continuous 1 (the PCTLA signal is counted by 6-bit counter).

16-bit shift register latches 16 bits of CTL duty discriminating output as the address code, which follows 0 after 9-bits of 1 in header. When the register finishes latching whole data of 16 bits, R/S flip-flop (VASS flag) is set and INTVA interrupt is requested.

Which interrupt request is generated; it can be checked by reading <VISSFL> and <VASSFL> of the VISS/VASS control register 1 (VIVACR1) in INTVA interrupt processing routine. VASS flag can be reset to 0 by using <VASFRS> of the Capture control register (CAPCR).

VASS address code generates INTVA interrupt 4 times by one set because address code has 4 headers. But the 4th code data becomes dummy. And, when the INTVA interrupt is generated, it is required to read address code before the next address code is started to latch. The 16-bit address code can be got by reading VASS data register (VASSDR) twice; the first read data is lower 8-bit (<VASS7> to <VASS0>) of the address code and the second read data is higher 8-bit (<VASS15> to <VASS8>). In addition, if data is written to VASSDR (dummy write), the VASSDR gets ready for reading lower 8-bit.

In order to detect VASS, set VIVACR2 <PCTLPO> to 0 when the tape operates forward and set VIVACR2<PCTLPO> to 1 when the tape operates reversely. Notice that the LSB and MSB of address code are reversed each other in case the tape operates reversely.

3.13 Head Amp Switch/Color Rotary control circuit (HA/CR)

TMP93C071 has the head amp switch (HA)/color rotary (CR) control output circuit. The HA output/the CR output are controlled by TPG00 and TPG05 of the timing pulse generator 0 (TPG0), and the COMPIN (P87) pin input value. The input enable and disable of the COMPIN can be controlled by CSYNCR<COMPS >.

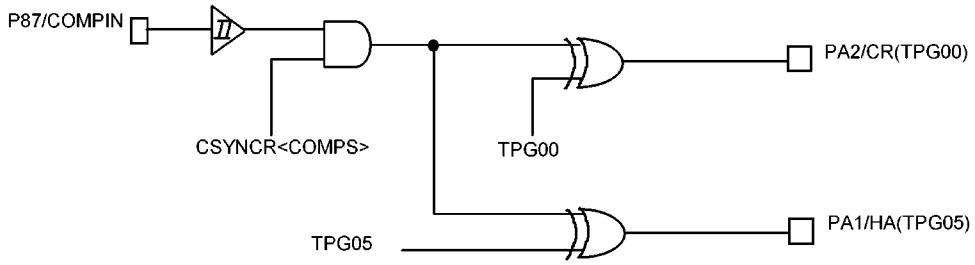


Figure 3.13.1 HA/CR control circuit

CSYNC control register

CSYNCR (0057H)

	7	6	5	4	3	2	1	0
bit Symbol	COMPS			CSYNCPO	SEPMOD	CSYNBP	HSEN	MASK
Read/Write	R/W			R/W	R/W	R/W	R/W	R/W
After reset	0			0	0	0	0	0
Function	COMPIN input control 0: Disable (TPG00/ TPG05 Output) 1: Enable			CSYNC input signal polarity selection 0: Positive 1: Negative	7-bit counter up/down count selection (duty) 0: up/down = 1/1 1: up/down = 1/2	CSYNC bypass control 0: V-sep. 1: Bypass	Pseudo-Hsync generation to PV/PH 0: Non-synchronized with Csync input 1: synchronize with Csync input	Vsync mask control 0: — 1: Release masking (One-shot)

Figure 3.13.2 Registers for HA/CR

3.14 SYNC SIGNAL SEPARATOR (CSYNC)

The Sync Signal Separator separates the Vertical Synchronizing Signal (V.SYNC) and Horizontal Synchronizing Signal (H.SYNC) from composite synchronizing signal (C.SYNC signal).

3.14.1 Configuration

The Sync Signal Separator consists of H/V Separator and H.Pulse Generator.
A configuration of the Sync Signal Separator is shown in Figure 3.14.1.

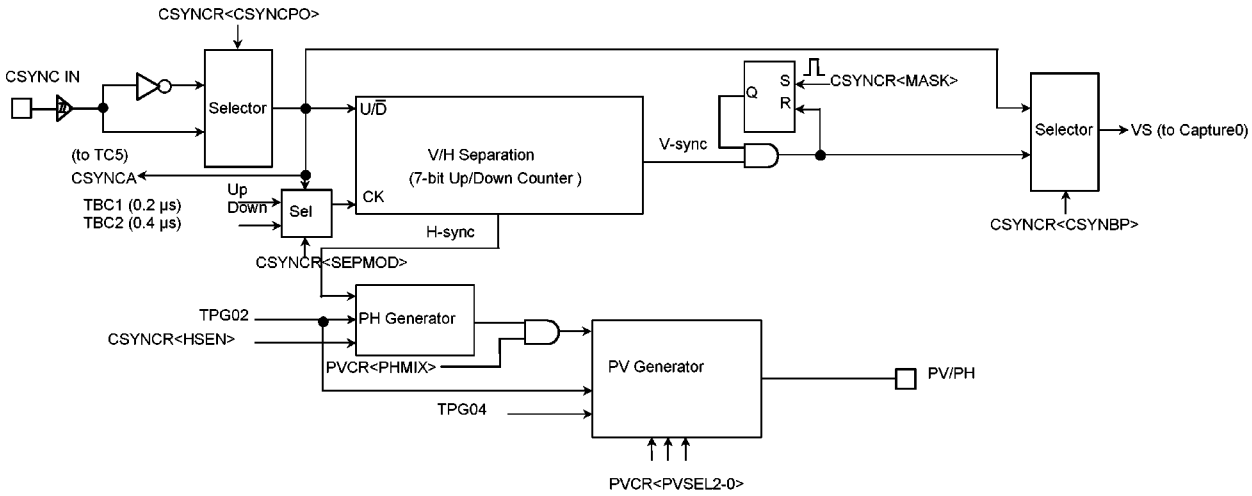


Figure 3. 14.1 Configuration of Sync Signal Separator

(1) H/V Separator

The H/V separator separates H.SYNC and V.SYNC signals from the C.SYNC inputted from CSYNCIN pin. A separated V.SYNC signal is transferred to Capture 0 (CAP0: VS) via the Capture input control circuit (CAPIN). And it is used for reference signal in the servo processing routine.
A separated H.SYNC signal is inputted to the H.Pulse generator.

(2) H.Pulse Generator

The H.Pulse generator generates serrated-pulse (HP signal), synchronizing with H.SYNC signal from C.SYNC, in pseudo-V.SYNC signal.
The HP signal is transferred to the Pseudo-sync signal output circuit (PV/PH), and it is superimposed to the pseudo-V.SYNC signal as serrated-pulse.
(Refer to 3.15 Pseudo-sync signal output circuit.)

3.14.2 Control Registers

CSYNC control register

CSYNCR (0057H)

	7	6	5	4	3	2	1	0
bit Symbol	COMPS			CSYNCPO	SEPMOD	CSYNBP	HSEN	MASK
Read/Write	R/W			R/W	R/W	R/W	R/W	R/W
After reset	0			0	0	0	0	0
Function	COMPIN input control 0: Disable (TPG00/ TPG05 Output) 1: Enable			CSYNC input signal polarity selection 0: Positive 1: Negative	7-bit counter up/down count selection (duty) 0: up/down = 1/1 1: up/down = 1/2	CSYNC bypass control 0: V-sep. 1: Bypass	Pseudo-Hsync generation to PV/PH 0: Non-synchronized with Csync input 1: Synchronize with Csync input	Vsync mask control 0: – 1: Release masking (One-shot)

Figure 3.14.2 Registers for CSYNC

3.14.3 H/V Separator

The H/V Separator Separates the Vertical Synchronizing Signal (V.SYNC) and Horizontal Synchronizing Signal (H.SYNC) from Composite Synchronizing Signal inputted from CSYNC pin.

The H/V Separator consists of 7-bit up/down counter and pattern detector (compare / match circuit).

7-bit up/down counter counts TBC1 (2^2/fc) or TBC2 (2^3/fc) output from the Time Base Counter. And its direction for counting is controlled by input polarity of CSYNCIN pin; CSYNCA = 1 is for up count and CSYNCA = 0 is for down count. The source clock for up count is only TBC1 (0.2 µs: fc = 20 MHz). The source clock for down count is selected either TB1 or TBC2 (0.4 µs: fc = 20 MHz) by CSYNCR <SEPMOD>. The input polarity of CSYNCIN pin (CSYNCA signal) is selected by setting <SYNCPO> in CSYNCR and it controls the direction for counting 7-bit counter. In case that CSYNCA is 1, counter stops when counter output becomes all 1. And in case that CSYNCA is 0, counter stops when counter output becomes all 0. CSYNCA is counted up by the 16-bit timer counter 5 (TC5). When CSYNCR <CSYNBP> is set to 1, CSYNCA instead of V-SYNC is inputted to capture 0 (CAP0).

(1) V.SYNC separation

If pattern detector (compare/match circuit) detects "114 (72H)" (TBC1 term: 200 [ns] at fc = 20 [MHz]. Therefore, threshold rate is 200 [ns] × 114 = 22.8 [µs]), it outputs the V.SYNC signal. VS signal is inputted to capture 0 (CAP0) at the same time. V.SYNC resets flip-flop for masking. Flip-flop for masking are reset once, the following V.SYNC signals are not accepted until masking is released. Setting flip-flop (to release masking) is executed on <MASK> of CSYNC control register (CSYNCR). Timing chart of V.SYNC separation is show in Figure 3.14.3.

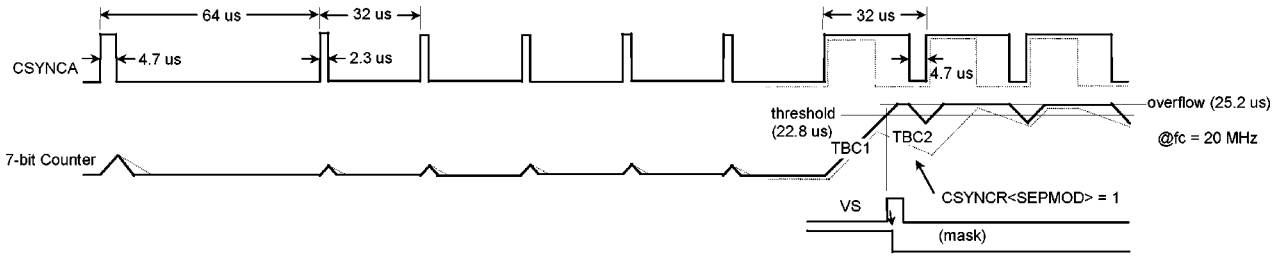


Figure 3.14.3 Timing Chart for V.SYNC separation

(2) H.SYNC separation

If pattern detector detects "15 (FH)" (TBC1 term: 200 [ns] at fc = 20 [MHz]. Therefore, threshold rate is 200 [ns] × 15 = 3.0 [µs]), it outputs the H.SYNC signal. In case that CSYNCA signal is 0, H.SYNC signal is reset when 7-bit counter becomes all 0. H.SYNC (3.0 µs) signal, which are separated by H/V Separator, is transferred to H.Pulse generator.

3.14.4 H.PULSE Generator

H.SYNC Generator generates serrated pulse in V.SYNC signal. This generated pulse (HP signal) is transferred to Pseudo-sync signal output circuit (PV/PH), and it can be mixed to pseudo-V.SYNC signal. Configuration of the H.Pulse generator is shown in Figure 3.14.(4).

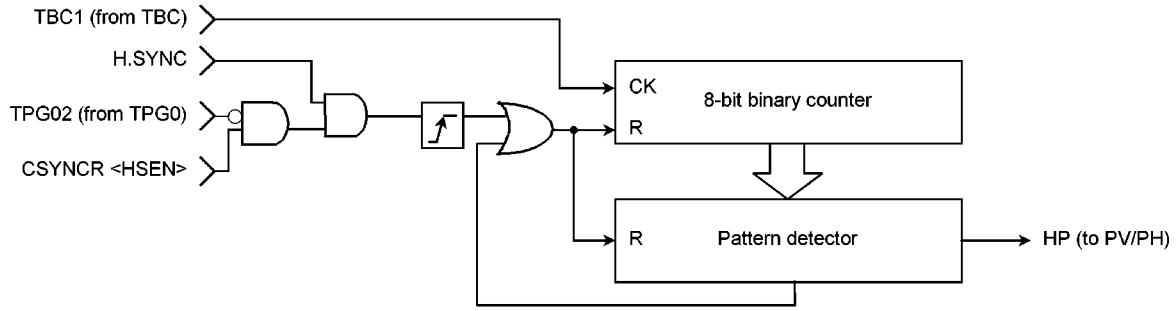


Figure 3.14.4 H.Pulse Generator

H.Pulse Generator generates serrated pulse (HP signal) to mix with pseudo-V.SYNC. The timing to mix pseudo-V.SYNC with HP signal is controlled by TPG02 from the Timing pulse generator 0 (TPG0) (Refer to section 3.15 Pseudo-sync signal output circuit). By setting <HSEN> in CSYNC control register (CSYNCR) to 1, HP signal can be synchronized with the H.SYNC signal, outputted from H/V Separator, during TPG02 is "L". In case that <HSEN> is 0, HP signal is not synchronized with C.SYNC signal input. Wave form of the HP signal output is shown in Figure 3.14.5.

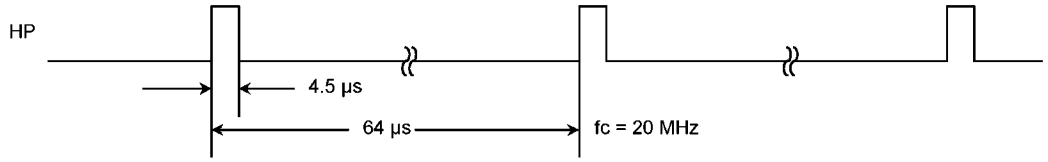


Figure 3.14.5 Wave form of HP signal output

3.15 Pseudo-sync Signal Output Circuit (PV/PH)

The TMP93C071 has a function to output a pseudo-sync signal (PV) in place of the playback sync signal during special effect reproduction. The PV output is controlled by the timing pulse generator 0 (TPG0)'s TPG02 and TPG04 outputs and the PV control register (PVCR).

3.15.1 Circuit Configuration

The pseudo-sync signal output circuit consists of a sync signal mixing circuit and a 3-level output circuit. The sync signal mixing circuit is used to superimpose the serrated pulse (HP) that is generated by the H pulse generator of the sync signal separation circuit (CSYNC) on a pseudo-V.SYNC signal.

3.15.2 Control Register

PV/PH control register

PVCR (0058H)	7	6	5	4	3	2	1	0
bit Symbol					PHMIX	PVSEL2	PVSEL1	PVSEL0
Read/Write					R/W	R/W		
After reset					0	0	0	0
Function					PH mixing 0: OFF 1: ON	PV/PH output format control		

Figure 3.15.1 Registers for PV/PH

3.15.3 Control of Pseudo-sync Signal Output

Output of the pseudo-sync signal (PV) is controlled by TPG02 and TPG04 outputs of the timing pulse generator 0 (TPG0) and the PV control register (PVCR).

The vertical sync signal (V.SYNC) is patterned by the TPG0 which is output from TPG04. The TPG02 output is used to set a period at which time the serration (serrated pulse) of V.SYNC is inserted. The serration (HP signal) is generated by the H pulse generator of the sync signal separation circuit (CSYNC). The HP signal is inserted into V.SYNC by setting the PVCR register PHMIX bit to 1.

The pseudo-sync signal has six output formats which can be selected by the PVCR register's PVSEL2-0 bits. Figure 3.15.2 shows the output formats of the pseudo-sync signal.

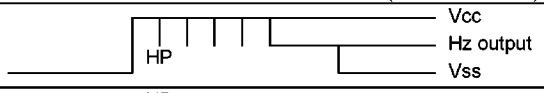
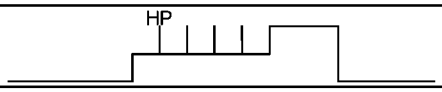
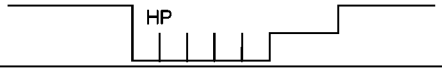
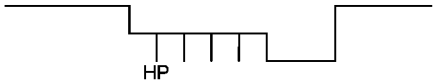
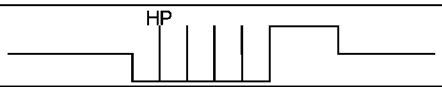
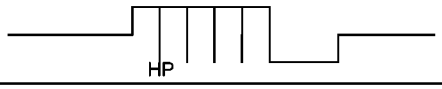
<PVSEL 2 to 0>			TPG02 / TPG04 (<PHMIX> =1)
2	1	0	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	*	0	
1	*	1	

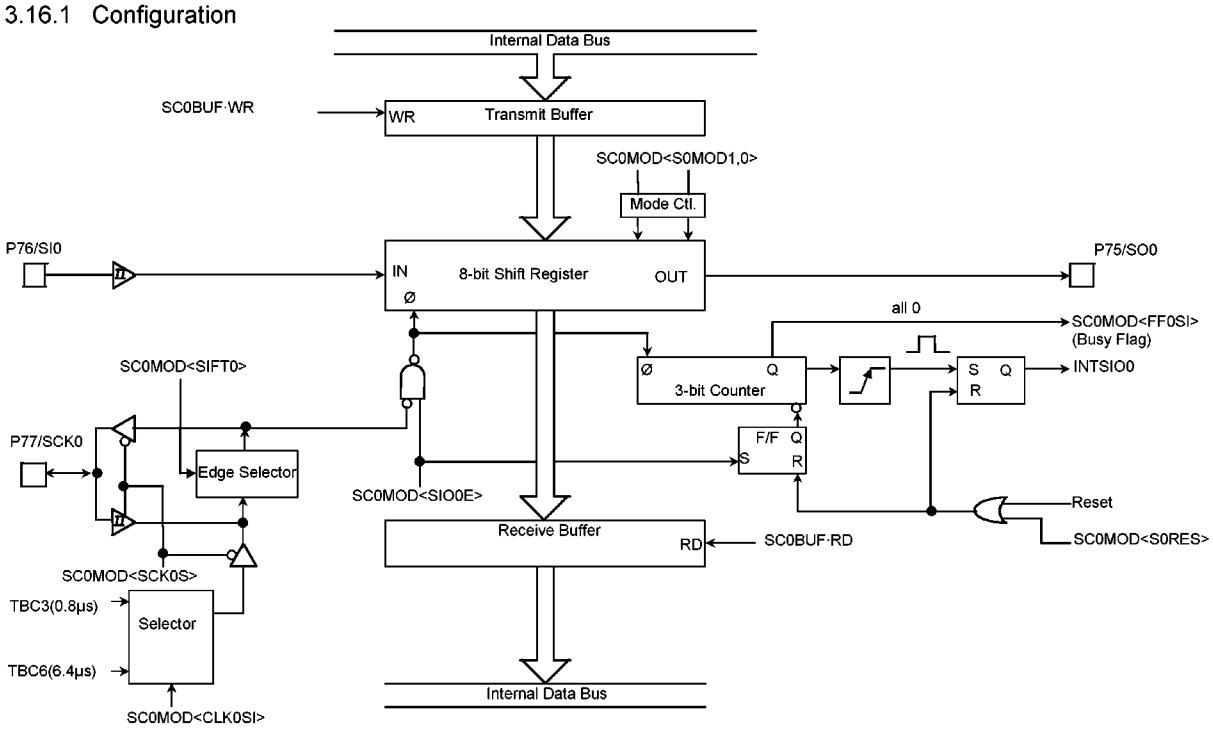
Figure 3.15.2 Pseudo-Vsync output format

3.16 Serial Channels (SIO0 / SIO1)

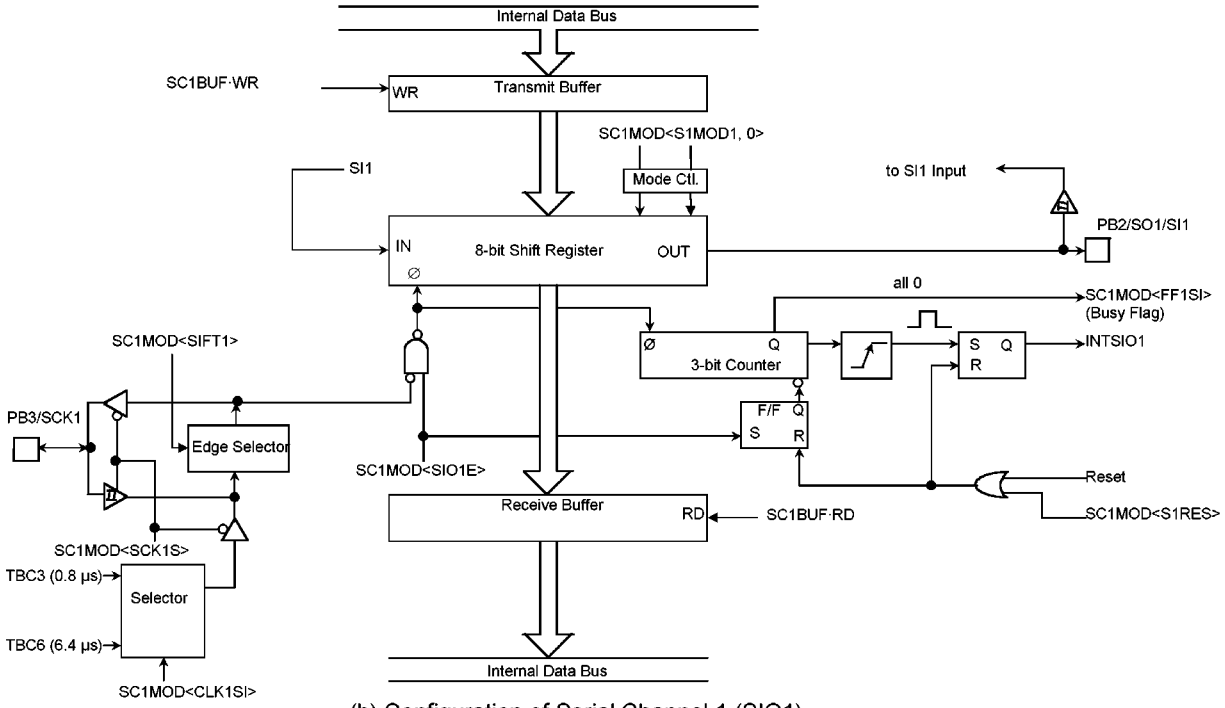
The TMP93C071 has two built-in 8-bit synchronous serial channels. Serial channel 0 (SIO0) is connected to an external circuit via P77 (SCK0), P75 (SO0), P76 (SI0), and serial channel 1 (SIO1) is connected to an external circuit via PB3 (SCK1), PB2 (SO1/SI1). Serial channel 0 and 1 are identical circuits, configured independently.

SIO1 is limited for some function, because both transmit data output (SO1) and receive data input (SI1) are shared with PB2 pin.

3.16.1 Configuration



(a) Configuration of Serial Channel 1 (SIO0)



(b) Configuration of Serial Channel 1 (SIO1)

Figure 3.16.1 Configuration of Serial Channel 0 (SIO1)

3.16.2 Control Registers

The serial channels are controlled by two (2) control registers SC0MOD, SC1MOD and two (2) buffer registers SC0BUF, SC1BUF.

SIO0 control register

SC0MOD (0060H)	7	6	5	4	3	2	1	0
bit Symbol	FF0SI	S0RES	S0MOD1	S0MOD0	SIFT0	CLK0SI	SCK0S	SIO0E
Read/Write	R	R/W	R/W		R/W	R/W	R/W	R/W
After reset	1	0	0	0	0	0	0	0
Function	Transfer monitor flag 0: Under transferring 1: Stop	Serial transfer terminate 0: — 1: Terminate	Transfer mode select 00: Transmit mode 01: Receive mode 10: — 11: Transmit/receive mode		Shift edge select 0: Leading (Falling) 1: Trailing (Rising)	Internal clock rate select 0: TBC3 (0.8 µs) 1: TBC7 (6.4 µs)	Serial clock source select 0: Internal clock 1: External clock	Serial transfer enable/ disable 0: Disable 1: Enable

SIO0 buffer register

SC0BUF (0061H)	7	6	5	4	3	2	1	0
bit Symbol	TRB07	TRB06	TRB05	TRB04	TRB03	TRB02	TRB01	TRB00
Read/Write	R/W							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

SIO1 control register SIO1

SC1MOD (0062H)	7	6	5	4	3	2	1	0
bit Symbol	FF1SI	S1RES	S1MOD1	S1MOD0	SIFT1	CLK1SI	SCK1S	SIO1E
Read/Write	R	R/W	R/W		R/W	R/W	R/W	R/W
After reset	1	0	0	0	0	0	0	0
Function	Transfer monitor flag 0: Under transferring 1: Stop	Serial transfer terminate 0: — 1: Terminate	Transfer mode select 00: Transmit mode 01: Receive mode 10: — 11: Transmit/receive mode		Shift edge select 0: Leading (Falling) 1: Trailing (Rising)	Internal clock rate select 0: TBC3 (0.8 µs) 1: TBC7 (6.4 µs)	Serial clock source select 0: Internal clock 1: External clock	Serial transfer enable/ disable 0: Disable 1: Enable

(Note): The SIO1 can not be used as the transmit/receive mode because the transmit line and receive line of the SIO1 are connected to the common pin (PB2).

SIO1 buffer register

SC1BUF (0063H)	7	6	5	4	3	2	1	0
bit Symbol	TRB17	TRB16	TRB15	TRB14	TRB13	TRB12	TRB11	TRB10
Read/Write	R/W							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

Figure 3.16.2 Registers for SIO0/SIO1

3.16.3 Operation

(1) Serial Clock

① Clock Source Selection

The clock of SIO can be selected from either external clock or internal clock by setting <SCK0S> bit in SC0MOD register (<SCK1S> bit in SC1MOD register for channel 1).

a. Internal clock

The clock rate can be selected from either TBC3 (2^4/fc) or TBC6 (2^7/fc) by setting SC0MOD <CLK0SI> (SC1MOD <CLK1SI> for channel 1). Table 3.16.1 shows the maximum transfer rate using the internal clock. The serial clock outputs from SCK0 (SCK1 for channel 1).

The serial clock automatically stops after the end of one-frame serial operation, and waits for the next serial operation. The serial clock holds high-level are not transferred.

Table 3.16.1 The Maximum Transfer Rate by Internal Clock

Internal Clock	Maximum Transfer Rate (at fc = 16 MHz)
TBC3 (2^4/fc)	1.25 Mbps
TBC6 (2^7/fc)	156.25 kbps

b. External clock

The clock input to the SCK0 (SCK1 for channel 1) pin is used as the serial clock. To make certain of the shift operation, set SC0MOD <CLK0SI> (SC1MOD <CLK1SI> for channel 1) to 0. Using certain shift operation, it is necessary to set more than 8/fc at both high-level and low-level of the serial clock width. When SC0MOD <SIO0E> (SC1MOD <SIO1E> for channel 1) are set to 0 under transferring and ending the transfer, it is necessary to set more than 16/fc at both high-level and low-level of the serial clock width.

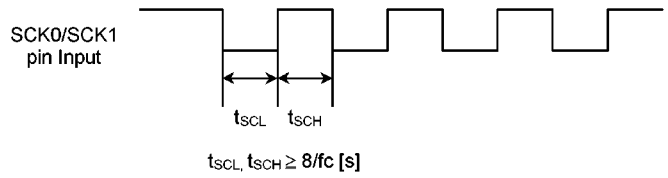


Figure 3.16.3 External Clock Input

② Shift Edge Selection

The leading or trailing edge shift can be selected by setting <SIFT0> bit in SC0MOD register (<SIFT1> bit in SC1MOD register for channel 1).

a. Leading edge

The serial data are shifted on the leading edge of the serial clock (falling edge of SCK0 or SCK1 pin input/output).

b. Trailing edge

The serial data are shifted on the trailing edge of the serial clock (rising edge of SCK0 or SCK1 pin input/output).

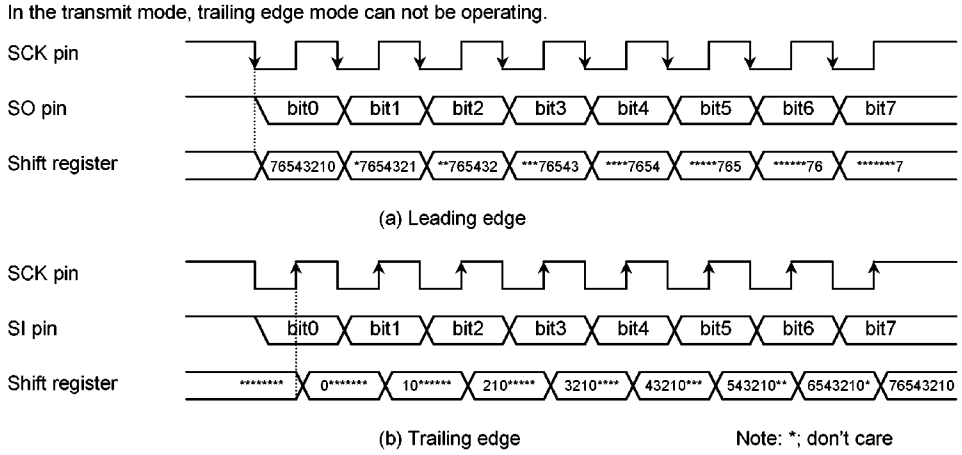


Figure 3.16.4 Shift Edge

(2) Serial Operation

Three transfer mode such as transmit, receive and simultaneous transmit-receive modes for serial channel 0 and 1 are selected by SC0MOD <S0MOD1, 0> (SC1MOD <S1MOD1, 0 for channel 1). After reset, SC0MOD <S0MOD1, 0> and SC1MOD <S0MOD1, 0> are cleared to 0, and transmit mode is selected. SIO1 can not be used as simultaneous transmit-receive mode, because both transmit data output (SO1) and receive data input (SI1) are shared with PB2 pin. The following explains the operation in each transfer mode.

① Transmit mode

After setting transmit mode to the control register, the first transmit data is written into buffer registers SC0BUF or SC1BUF (address 0061H or 0063H in memory). (When transmit mode is not set, transmit data can not be written into the buffer registers.) Setting SC0MOD<SIO0E> or SC1MOD<SIO1E> to 1 starts transmit operation. As the transmit starts, the transmit data area synchronized with the leading edge of the serial clock (falling edge shift), and sequentially output from the SO pin of the LSB side. At the same time, the transmit data area transferred from the buffer registers to the shift registers. Since the buffer registers are empty, the buffer empty interrupt (INTSIO0 or INTSIO1) is generated to request new data. When the next transmit data is written into the buffer register in the interrupt service program, the interrupt request signal is cleared.

Note: After the serial transfer enable/disable register SC0CR<SCO0E> or SC1CR<SIO1E> are set to 1, undefined data are output from SO pin till the first falling edge of the serial clock.

(Internal clock mode)

In the internal clock mode, When all data are transmitted and no subsequent data is set in the register, the serial clock output stops and a wait begins.

Figure 3.16.5 (a) shows the timing chart of internal clock operation in transmit mode (with wait).

(External clock mode)

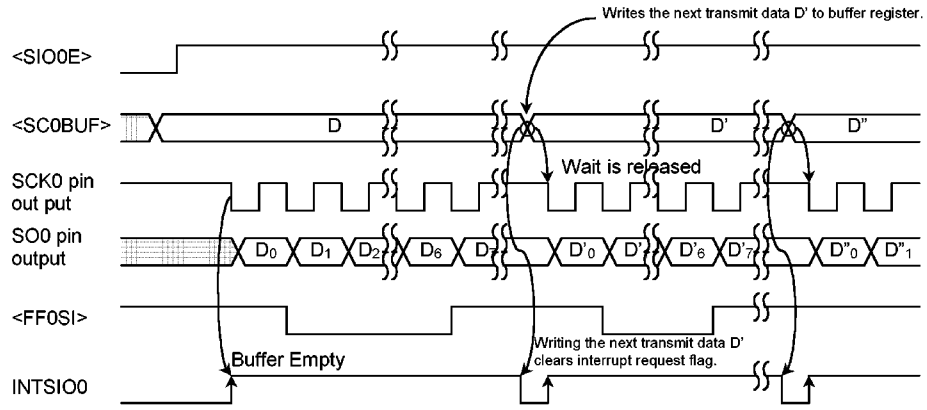
In the external clock mode, data must be set in the buffer registers before the next data shift operation begins. Therefore the transfer rate is determined by the maximum delay time from interrupt request generation to writing the transmit data into the buffer register in the interrupt service program.

Figure 3.16.5 (b) shows the timing chart of external clock operation in transmit mode.

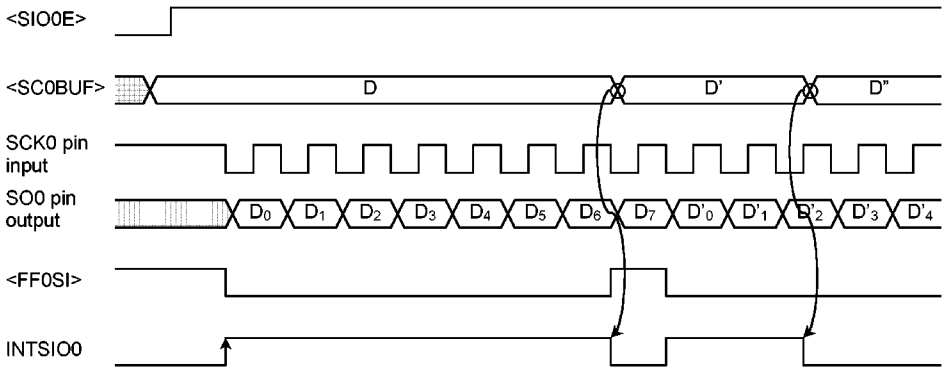
To end the transmit operation, set SC0MOD <SIO0E> or SC1MOD <SIO1E> to 0 instead of writing the next transmit data into the buffer register in the interrupt service program. When <SIO0E> or <SIO1E> are cleared to 0, the transmit operation stops at the end of transmitting a data which is shifted out currently. The transmit operation also stops at once by setting SC0MOD <S0RES> or SC1MOD <S1RES> to 1, and <S0RES> or <S1RES> are automatically cleared to 0.

The end of transmit operation can be confirmed by reading out the serial transfer monitor flag SC0MOD<FF0SI> or SC1MOD<FF1SI>. (When the transmit operation is finished, the serial transfer monitor flag is set to 1.)

In the external clock mode, the serial transfer enable/disable register<SIO0E> or <SIO1E> must be cleared to 0 before starting the next transmit data shift operation. If <SIO0E> or <SIO1E> is not cleared to 0 before the shift operation begins, operations stop after transferring the next transmit data (dummy).



(a) Internal clock operation in transmit mode (with wait)



(b) External clock operation in transmit mode

Figure 3.16.5 Serial channel Timing Chart in Transmit mode

② Receive Mode

After setting the control register to receive mode, setting SC0MOD <SIO0E> or SC1MOD <SIO1E> to 1 makes receive possible. The shift data is synchronized with the serial clock and fetched from the SI pin. When 8-bit data is fetched, it is transferred from the shift register to the buffer register, and buffer-full interrupt INTSIO0 or INTSIO1 is generated to request a read of receive data. The receive data are read from the buffer register in the interrupt service program. The interrupt request signal is cleared when they are read.

(Internal clock mode)

In the internal clock mode, if the previous receive data has not been read from the buffer register after the next data is fetched, the serial clock stops and waits until the previous data is read.

Figure 3.16.6 (a) shows the timing chart of internal clock operation in receive mode (leading edge shift with wait).

Figure 3.16.7 (a) shows the timing chart of internal clock operation in receive mode (trailing edge shift with wait).

(External clock mode)

In the external clock mode, as the shift operation synchronizes with supplied external clock, it is necessary to read from the buffer register before transferring the next receive data. If the previous data has not read, the receive data will not be transferred to the buffer register, and subsequent receive data will be canceled.

The maximum transfer rate of the external clock operation is determined by the maximum delay time from the generation of interrupt requests to receive data read.

Figure 3.16.6 (b) shows the timing chart of external clock operation in receive mode (leading edge shift).

Figure 3.16.7 (b) shows the timing chart of external clock operation in receive mode (trailing edge shift).

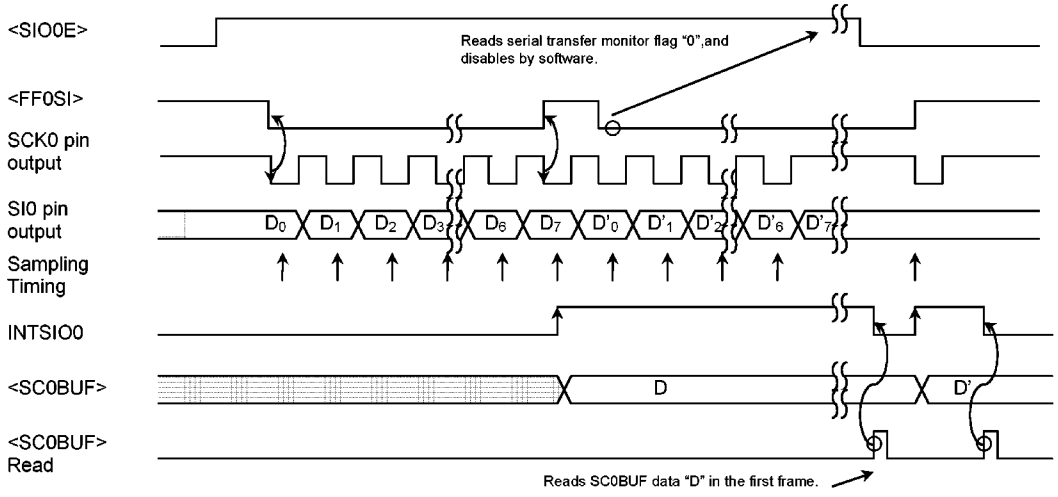
In the receive mode, either leading edge shift or trailing edge shift can be selected. When the leading edge is used for shifting, because data is fetched at the leading edge of the serial clock, the first shift data must already be input to the SI pin when the initial serial clock pulses are applied at transfer start.

To end the receive operation, set the serial transfer enable/disable register (SC0MOD) <SIO0E> or (SC1MOD) <SIO1E> to 0. When the serial transfer enable/disable register <SIO0E> or <SIO1E> are cleared to 0, the receive operation ends after 8 bits of receive data are fetched. Don't read the buffer register after clearing the serial transfer enable/disable register.

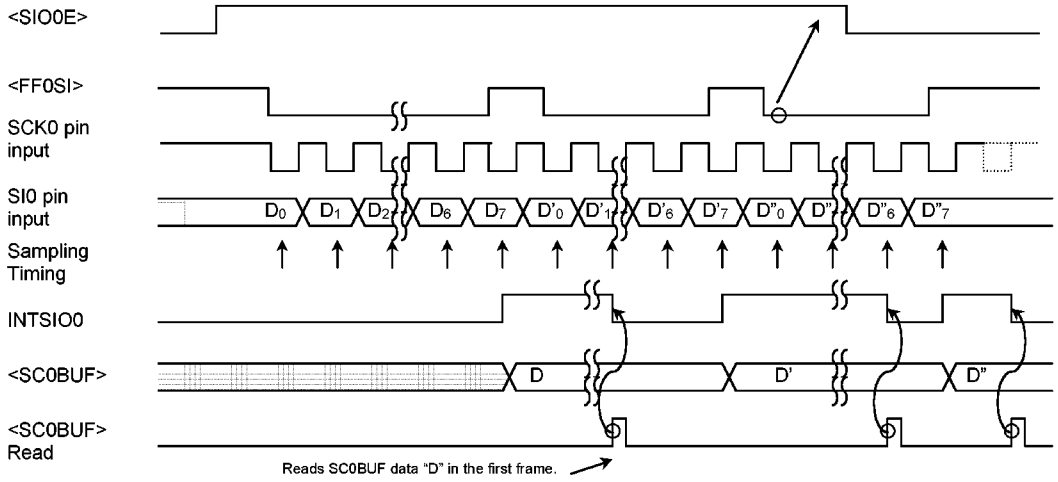
Setting the serial transfer enable/disable register (SC0MOD) <S0RES> or (SC1MOD) <S1RES> are set to 1 stops the serial transfer at once and <SIO0E> or <SIO1E> are cleared to 0.

Note : If the transfer mode is switched, the contents of the buffer registers can not be kept. If necessary to switch the transfer mode, the transfer mode should be switched after clearing the transfer enable/disable register <SIO0E> or <SIO1E> to 0 and reading out the last bit of the receive data.

The end of receive can be confirmed by reading serial transfer monitor flags SC0MOD <FF0SI> or SC1MOD <FF1SI>.

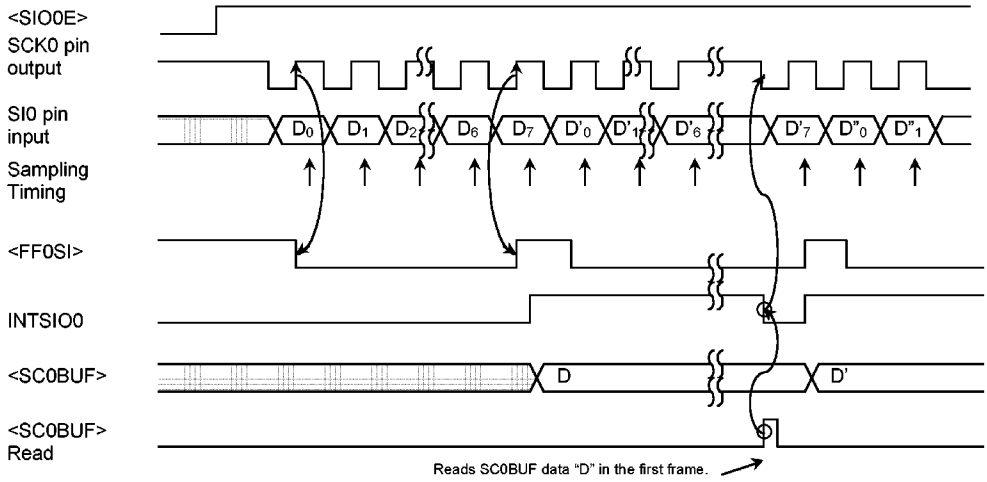


(a) Internal clock operation in receive mode (with leading edge and wait)

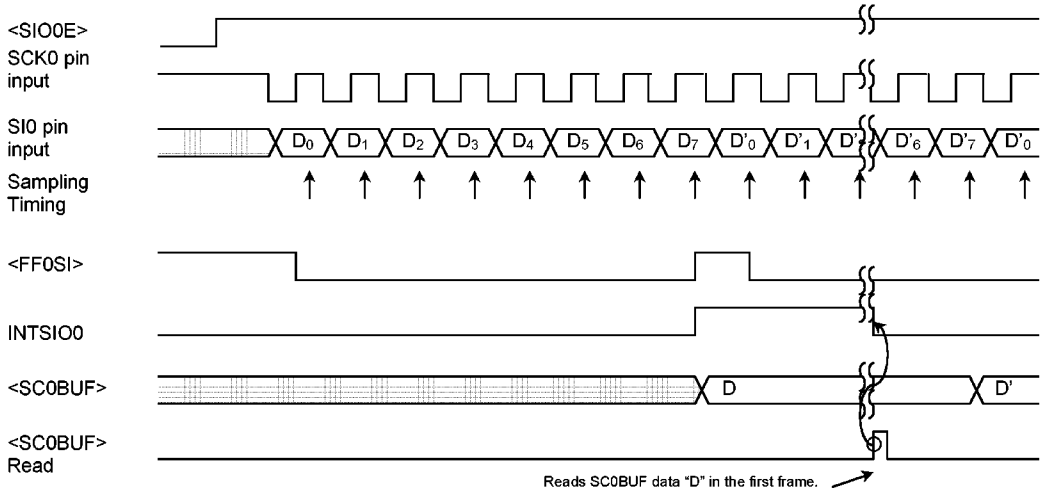


(b) External clock operation in receive mode (with leading edge)

Figure 3.16.6 Serial channel Timing chart in Receive mode (Leading edge)



(a) Internal clock operation in receive mode (with trailing edge and wait)



(b) External clock operation in receive mode (with trailing edge)

Figure 3.16.7 Serial channel Timing chart in Receive mode (Trailing edge)

③ Simultaneous Transmit / Receive Mode

The first transmit data are written into the buffer register SC0BUF (SC1BUF for channel 1) after the simultaneous transmit-receive mode is set to the control register. Then, setting the serial channel control register SC0MOD <SIO0E> or SC1MOD <SIO1E> to 1 enables transmitting or receiving data. The transmit data area output from the SO pin at the leading edge of the serial clock. The receive data area fetched from the SI pin at the trailing edge of the serial clock.

When the 8-bit receive data are fetched, the data are transferred from the shift register to the buffer register, and the interrupt request (INTSIO0 or INTSIO1) is generated to request receive data read. In the interrupt service program, the received data are read out from the buffer register and the next transmit data are written into the buffer register.

Note: After the serial transfer enable/disable register SC0MOD <SIO0E> or SC1MOD <SIO1E> is set to 1, undefined data are output from SO pin till the first falling edge of serial clock. SIO1 can not be used as simultaneous transmit / receive mode.

(Internal clock mode)

In the internal clock mode, a wait begins until the receive data are read and the next transmit data are written into the buffer register. Figure 3.16.8 (a) shows the timing chart of internal clock operation is simultaneous transmit / receive mode.

(External clock mode)

In the external clock mode, the receive data must be read and the next transmit data written before the next shift operation, because the shift operation is synchronized with external supplied clock pulses. The maximum transfer rate of the external clock operation is determined by the maximum delay time from interrupt request generation to receive data read and transmit data write.

Figure 3.16.8 (b) shows the timing chart of external clock operation in simultaneous transmit/receive mode.

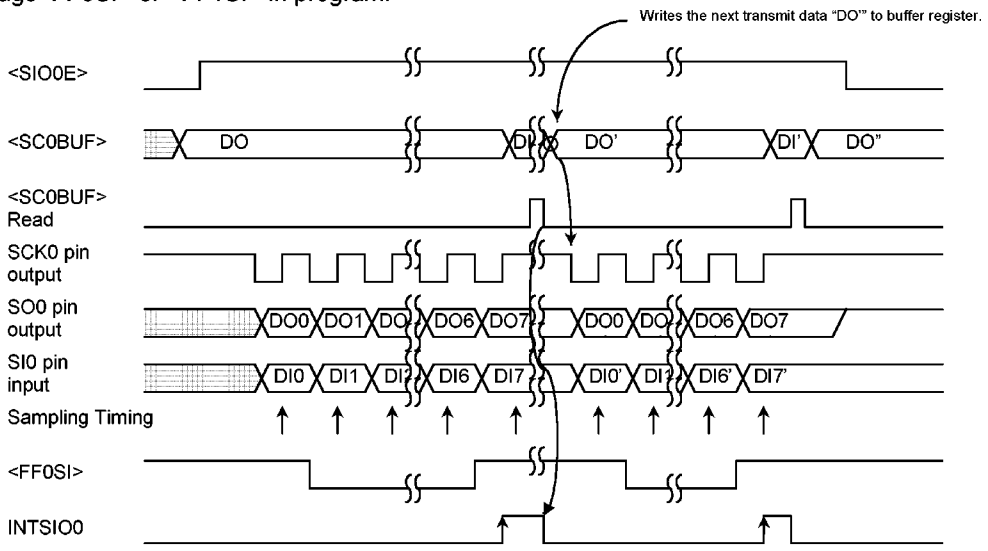
Since the buffer registers are used for both transmit and receive data, always ensure that transmit data is written after 8 bits of receive data are read.

To end the simultaneous transmit-receive operation, clear the serial transfer register<SIO0E> or <SIO1E> to 0.

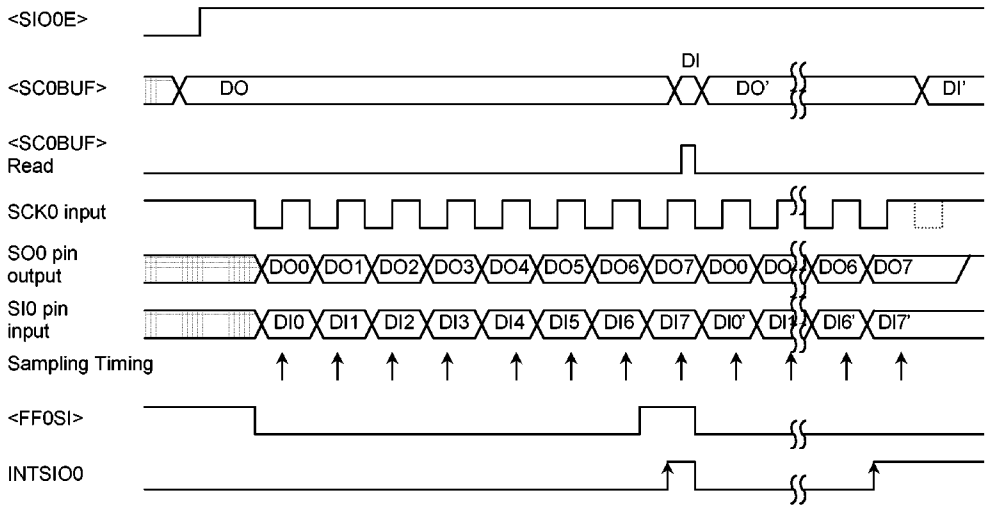
When the serial transfer enable/disable register<SIO0E> or <SIO1E> is cleared to 0, the simultaneous transmit-receive operation ends after the 8 bits of receive data are fetched. Don't read the buffer register after clearing the serial transfer enable/disable register.

In the simultaneous transmit-receive mode, the serial transfer operation ends just after setting <S0RES> or <S1RES> to 1, and the serial transfer enable/disable register <SIO0E> or <SIO1E> is cleared to 0.

The end of simultaneous transmit-receive can be confirmed by reading the serial transfer monitor flags<FF0SI> or <FF1SI> in program.



(a) Internal clock operation in simultaneous transmit/receive mode (with wait)



(b) External clock operation in simultaneous transmit/receive mode

Figure 3.16.8 Serial channel Timing chart in Simultaneous Transmit / Receive mode

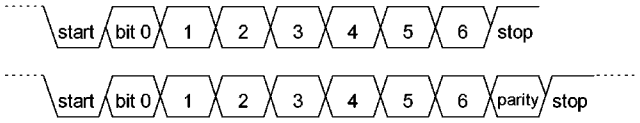
3.17 Serial Channel (UART)

TMP93C071 has an asynchronous serial I/F (UART).
The UART has the following operation modes.

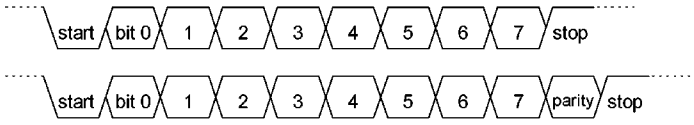
- UART mode (channel 0)
  - Mode 1: 7-bit data
  - Mode 2: 8-bit data
  - Mode 3: 9-bit data

In mode 1 and mode 2, a parity bit can be added. Mode 3 has wake-up function for making the master controller start slave controllers in serial link (multi-controller system).
Figure 3.17.1 shows the data format (for one frame) in each mode.

• Mode 1 (7-bit UART mode)



• Mode 2 (8-bit UART mode)



• Mode 3 (9-bit UART mode)

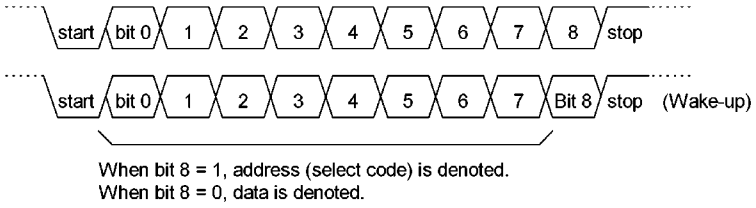


Figure 3.17.1 Data Formats

The UART has a buffer register for transmitting and receiving operations, in order to temporarily store transmitted or received data, so that transmitting and receiving operations can be done independently (full duplex).

The receiving data register is of a double buffer structure to prevent the occurrence of overrun error and provides one frame of margin before CPU reads the received data. The receiving data register stores the already received data while the buffer register receives the next frame data.

By using $\overline{CTS}$ and $\overline{RTS}$ (there is no $\overline{RTS}$ pin, so any 1 port must be controlled by software), it is possible to halt data send until the CPU finishes reading receive data every time a frame is received. (Handshake function)

A check function is added not to start the receiving operation by error start bits due to noise. The channel starts receiving data only when the start bit is detected to be normal at least twice in three samplings.

When the transmission buffer becomes empty and requests the CPU to send the next transmission data, or when data is stored in the receiving data register and the CPU is requested to read the data, INTTX or INTRX interrupt occurs. Besides, if an overrun error, parity error, or framing error occurs during receiving operation, flag SC2CR <OERR, PERR, FERR> will be set.

The UART includes a special baud rate generator, which can set any baud rate by dividing the frequency of 4 clocks (TBC1, TBC2, TBC3, and TBC4) from the time base counter (TBC). In addition, the UART can be operated by using the timeout output of the timer counter 0 (TC0).

3.17.1 Control Registers

The UART is controlled by 3 control registers SC2CR, SC2MOD and BR2CR. Transmitted and received data are stored in register SC2BUF.

SC2MOD (0064H)

bit Symbol	7 TB8	6 CTSE	5 RXE	4 WU	3 SM1	2 SM0	1 SC1	0 SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfer data Bit 8	Hand shake Function 0: $\overline{CTS}$ disable 1: $\overline{CTS}$ enable	Receiving Function 0: Receive disable 1: Receive enable	Wake up Function 0: disable 1: enable	Serial transmission mode 00: inhibit 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TC0 Trigger 01: baud rate generator 10: Internal clock fc/2 11: inhibit	

Serial transmission clock (For UART mode)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock (fc/2)
11	inhibit

Serial transmission mode

00	inhibit	
01	UART	7-bit length
10		8-bit length
11		9-bit length

Wake up Function
(Don't care in the modes other than 9-bit UART)

0	Disable
1	Enable

Receiving Function

0	Receive Disable
1	Receive Enable

Hand shake function ($\overline{CTS}$ Pin) enable

0	Disable (always Transferable)
1	Enable

Transmission data bit 8

8-bit UART mode (parity)	Stores received parity bit
9-bit UART mode	Stores received data bit 8

Figure 3.17.2 Serial Mode Control Register (SC2MOD)

SC2CR (0065H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	–	–
Read/Write	R	R/W		R (cleared to Zero when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data Bit 8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: error			Always set to 00.	

Overrun | Parity | Framing

Framing error flag
Parity error flag
Overrun error flag
} Cleared to Zero when read. (Note 1)

Enable parity addition

0	Prohibition (disable)
1	Permission (enable)

Addition / check of even parity

0	Odd parity
1	Even parity

Receiving data bit 8

8-bit UART mode (parity)	Stores transmission parity bit
9-bit UART mode	Stores transmission data bit 8

Note1: As all error flags are cleared after reading, do not test only a single bit with a bit-testing instruction.

Figure 3.17.3 Serial Control Register (SC2CR)

BR2CR (0066H)

	7	6	5	4	3	2	1	0
bit Symbol	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	(Note) Always Write 0		00: TBC1 (0.2 µs) 00: TBC2 (0.4 µs) 10: TBC3 (0.8 µs) 00: TBC4 (1.6 µs)		Setting of the Divided frequency			

Setting of the divided frequency of baud rate generator

0000	16 divisions
0001 to 1111	1 division (not divided) to 15 divisions

Selecting the input clock of baud rate generator

00	Internal clock (0.2µs)
01	Internal clock (0.4µs)
10	Internal clock (0.8µs)
11	Internal clock (1.6µs)

Note: Bit 6 of BR2CR is read as 1.

Figure 3.17.4 Serial Channel Control (BR2CR)

SC2BUF (0067H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
Read/Write	R(Receiving) / W(Transmission)							
After reset	Undefined							

Figure 3.17.5 Serial Transmission / Receiving Buffer Registers (SC2BUF)

3.17.2 Configuration

Figure 3.17 (6) shows the block diagram of the serial channel 0.

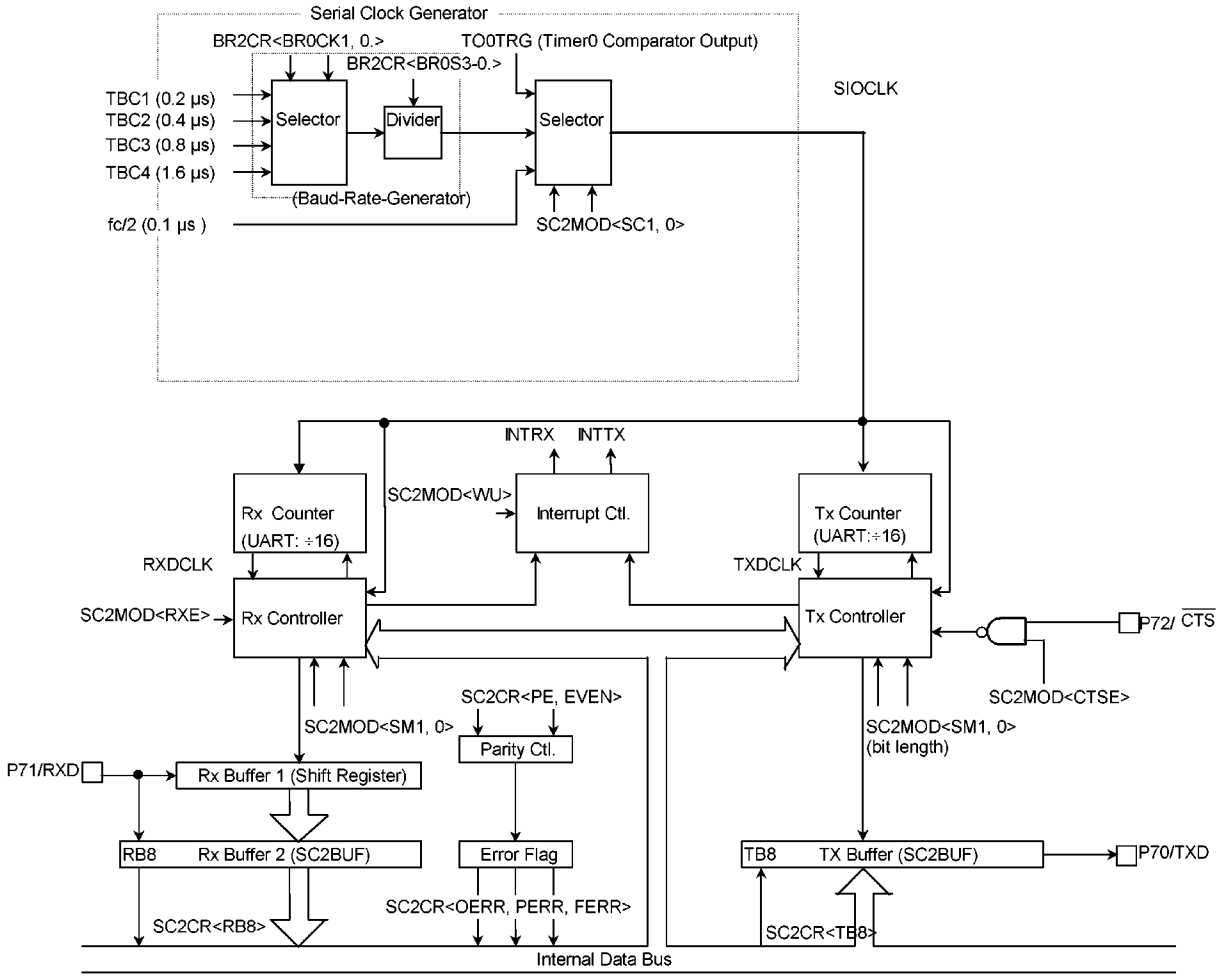


Figure 3.17.6 Block Diagram of the UART

① Baud Rate Generator

Baud rate generator comprises a circuit that generates transmission and receiving clocks to determine the transfer rate of the UART.

The input clock to the baud rate generator, TBC1, TBC2, TBC3, or TBC4 is generated by the time base counter (TBC). One of these input clocks is selected by the baud rate generator control register BR2CR<BR0CK1, 0>.

The baud rate generator includes a 4-bit frequency divider, which divides frequency by 1 to 16 values to determine the transfer rate.

How to calculate a transfer rate when the baud rate generator is used is explained below.

$$\text{Baud rate} = \frac{\text{Input clock of baud rate generator}}{\text{Frequency divisor of baud rate generator}} \div 16$$

Accordingly, when source clock fc is 12.288 MHz, input clock is TBC3 (fc/16), and frequency divisor is 5, the transfer rate in UART becomes as follows:

$$\text{Baud rate} = \frac{fc/16}{5} \div 16$$

$$= 12.288 \times 10^6 \div 16 \div 5 \div 16 = 9600 \text{ (bps)}$$

The maximum baud rate of this baud rate generator is 312.5 kbps (fc = 20 MHz).

Table 3.17.1 shows an example of the transfer rate.

Also with 8-bit timer 0 (TC0), the UART can get a transfer rate.

Table 3.17.1 Selection of UART Transfer Rate (1) (When Baud Rate Generator is used.)

Unit (Kbps)

fc[MHz]	Input clock Divider	TBC1 (4/fc)	TBC2 (8/fc)	TBC3 (16/fc)	TBC4 (32/fc)
12.288000	5	38.400	19.200	9.600	4.800
	10	19.200	9.600	4.800	2.400
19.660800	2	153.600	76.800	38.400	19.200
	4	76.800	38.400	19.200	9.600
	8	38.400	19.200	9.600	4.800
	16	19.200	9.600	4.800	2.400
20.000000	2	156.250	78.125	39.063	19.531
	4	78.125	39.063	19.531	9.766
	8	39.063	19.531	9.766	4.883
	16	19.531	9.766	4.883	2.441

② Serial Clock Generation Circuit

This circuit generates the basic clock for transmitting and receiving data.
According to the setting of SC2MOD <SC1, 0>, the above baud rate generator clock, internal clock fc/2 (max 625 kbps at fc = 20 MHz), or the match detect signal from timer 0, will be selected to generate the basic clock SIOCLK.

③ Receiving Counter

The receiving counter is a 4-bit binary counter and counts up by SIOCLK clock. 16 pulses of SIOCLK are used for receiving 1 bit of data, and the data bit is sampled three times at 7th, 8th and 9th clock.
With the three samples, the received data is evaluated by the rule of majority.
For example, if the sampled data bit is 1, 0 and 1 at 7th, 8th and 9th clock respectively, the received data is evaluated as 1. The sampled data 0, 0 and 1 is evaluated that the received data is 0.

④ Receiving Control

The receiving control has a circuit for detecting the start bit by the rule of majority. When two or more "0" are detected during 3 samples, it is recognized as start bit and the receiving operation is started.
Data being received are also evaluated by the rule of majority.

⑤ Receiving Buffer

To prevent overrun error, the receiving buffer has a double buffer structure.
Received data are stored one bit by one bit in the receiving buffer 1 (shift register type). When 7 bits or 8 bits of data is stored in the receiving buffer 1, the stored data are transferred to the receiving buffer 2 (SC2BUF), generating an interrupt INTRX. The CPU reads only receiving buffer 2 (SC2BUF). Even before the CPU reads the receiving buffer 2 (SC2BUF), the received data can be stored in the receiving buffer 1. However, unless the receiving buffer 2 (SC2BUF) is read before all bits of the next data are received by the receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of the receiving buffer 1 will be lost, although the contents of the receiving buffer 2 and SC2CR<RB8> is still preserved.
The parity bit added in 8-bit UART mode and the most significant bit (MSB) in 9-bit UART mode are stored in SC2CR<RB8>.
When in 9-bit UART mode, the wake-up function of the slave controllers is enabled by setting SC2MOD<WU> to 1, and interrupt INTRX occurs only when SC2CR<RB8> is set to 1.

⑥ Transmission Counter

Transmission counter is a 4-bit binary counter and counts by SIOCLK clock, generating TxDCLK every 16 clock pulses.

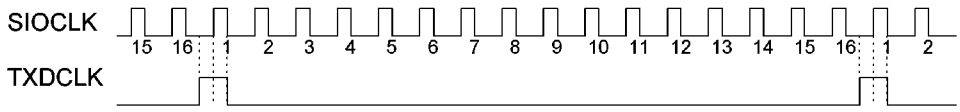


Figure 3.17.7 Generation of transmission Clock

⑦ Transmission Controller

When transmission data are written in the transmission buffer sent from the CPU, transmission starts at the rising edge of the next TxDCLK, generating a transmission shift clock TxDSFT.

Handshake Function

The serial channels use the $\overline{CTS}$ pin to transmit data in units of frames, thus preventing an overrun error. Use SC2MOD<CTSE> to enable or disable the handshake function.

When $\overline{CTS}$ goes high, data transmission is halted after the completion of the current transmission and is not restarted until $\overline{CTS}$ returns to low. An INTTX interrupt is generated to request the CPU for the next data to transmit. When the CPU write the data to the transmit buffer, processing enters standby mode.

An $\overline{RTS}$ pin is not provided, but a handshake function can easily be configured if the receiver sets any port assigned to the $\overline{RTS}$ function to high (in the receive interrupt routine) after data receive, and requests the transmitter to temporarily halt transmission.

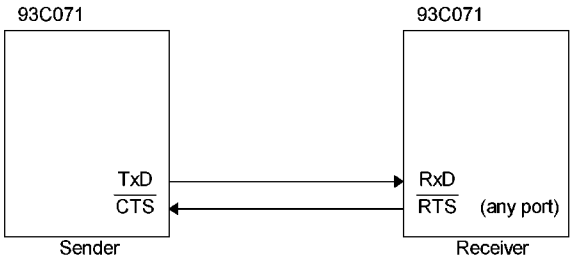
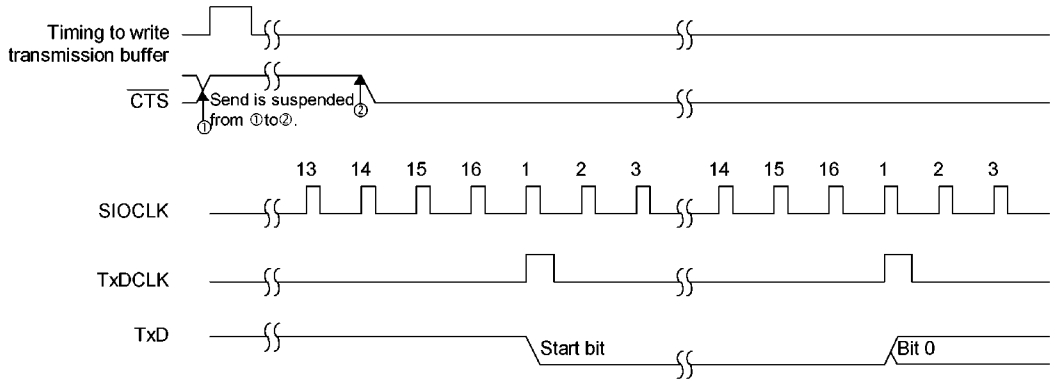


Figure 3.17.8 Handshake Function



Note1: If the $\overline{CTS}$ signal rises during transmission, the next data is not sent after the completion of the current transmission.

Note2: Transmission starts at the first TxDCLK clock fall after the $\overline{CTS}$ signal falls.

Figure 3.17.9 Timing of $\overline{CTS}$ (Clear to send)

⑧ Transmission Buffer

Transmission buffer (SC2BUF) shifts out and sends the transmission data written from the CPU. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX interrupt.

⑨ Parity Control Circuit

When serial channel control register SC2CR<PE>is set to 1, it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART or 8-bit UART mode. With SC2CR <EVEN> register, even (odd) parity can be selected.

For transmission, parity is automatically generated according to the data written in the transmission buffer SC2BUF, and data are transmitted after being stored in SC2BUF<TB7> when in 7-bit UART mode, while in SC2MOD <TB8> when in 8-bit UART mode. <PE> and <EVEN> must be set before transmission data are written in the transmission buffer.

For receiving, data are shifted in the receiving buffer 1, and parity is added after the data are transferred in the receiving buffer 2 (SC2BUF), and then compared with SC2BUF<RB7> when in 7-bit UART mode and with SC2MOD<RB8> when in 8-bit UART mode. If they are not equal, a parity error occurs, and SC2CR<PERR> flag is set.

⑩ Error Flag

Three error flags are provided to increase the reliability of receiving data.

1. Overrun error <OERR>

If all bits of the next data are received in receiving buffer 1 while valid data are stored in receiving buffer 2 (SC2BUF), an overrun error will occur.

2. Parity error <PERR>

The parity generated for the data shifted in receiving buffer 2 (SC2BUF) is compared with the parity bit received from RxD pin. If they are not equal, a parity error occurs.

3. Framing error <FERR>

The stop bit of received data is sampled three times around the center. If the majority is "0", a framing error occurs.

⑪ Generating Timing

Receiving

Mode	9 bit (Note)	8 bit + parity (Note)	8 bit, 7 bit + parity, 7 bit
Interrupt timing	Center of last bit (Bit 8)	Center of last bit (parity bit)	Center of stop bit
Framing error timing	Center of stop bit	Center of stop bit	Center of stop bit
Parity error timing	—	Center of last bit (parity bit)	Center of stop bit
Overrun error timing	Center of last bit (Bit 8)	Center of last bit (parity bit)	Center of stop bit

Note: In 9-bit and 8-bit + parity modes, interrupts coincide with the ninth bit pulse. Thus, When servicing the interrupt, it is necessary to wait for a 1-bit period (to allow the stop bit to be transferred) to allow checking for a framing error.

Transmitting

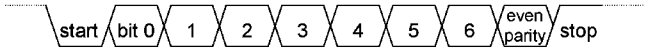
Mode	9 bit	8 bit + parity	8 bit, 7 bit + parity, 7 bit
Interrupt timing	Just before last bit is transmitted.	Just before last bit is transmitted.	Just before last bit is transmitted.

3.17.3 Operational Description

(1) Mode 1 (7-bit UART Mode)

7-bit mode can be set by setting serial channel mode register SC2MOD <SM1,0> to "01".
In this mode, a parity bit can be added, and the addition of a parity bit can be enabled or disabled by serial channel control register SC2CR<PE>, and even parity or odd parity is selected by SC2CR <EVEN> when <PE> is set to "1" (enable).

Setting example: When transmitting data with the following format, the control registers should be set as described below.



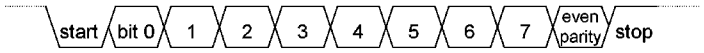
		7	6	5	4	3	2	1	0	
P7FC	←	–	X	–	–	–	X	X	1	} Select P70 as the TxD pin.
P7CR	←	–	–	–	–	–	–	–	1	
SC2MOD	←	X	0	–	X	0	1	0	1	Set 7 -bit UART mode.
SC2CR	←	X	1	1	X	X	X	0	0	Add an even parity.
BR2CR	←	0	X	1	1	1	0	1	0	Set transfer rate at 2400 bps.
INTRXTX	←	0	1	0	0	–	–	–	–	Enable INTTX interrupt and set interrupt level 4.
SC2BUF	←	*	*	*	*	*	*	*	*	Set data for transmission.

Note : X; don't care –; no change

(2) Mode 2 (8-bit UART Mode)

8-bit UART mode can be specified by setting SC2MOD<SM1,0> to "10". In this mode, parity bit can be added, the addition of a parity bit is enabled or disabled by SC2CR<PE>, and even parity or odd parity is selected by SC2CR<EVEN> when <PE> is set to "1" (enable).

Setting example: When receiving data with the following format, the control register should be set as described below.



Main setting

		7	6	5	4	3	2	1	0	
P7CR	←	–	–	–	–	–	–	0	–	Select P71 (RxD) as the input pin.
SC2MOD	←	–	0	1	X	1	0	0	1	Enable receving in 8-bit UART mode.
SC2CR	←	X	0	1	X	X	X	0	0	Add an odd parity.
BR2CR	←	0	X	1	0	0	1	0	1	Set transfer rate at 9600 bps.
INTRXTX	←	–	–	–	–	0	1	0	0	Enable INTRX interrupt and set interrupt level 4.

Interrupt processing

```
Acc ←        SC2CR AND 00011100 }
if Acc ≠ 0 then ERROR           } Check for error.
Acc ←        SC2BUF               Read the received data.
```

Note: X; don't care –; no change

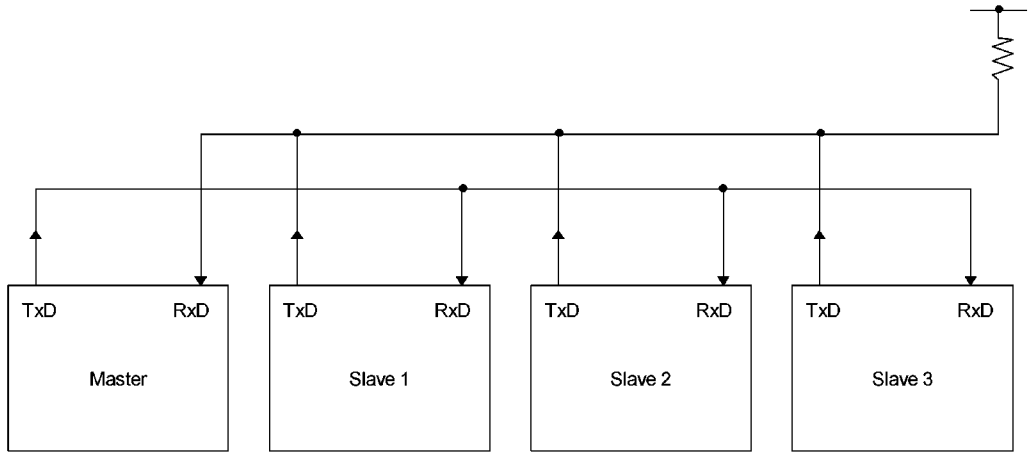
(3) Mode 3 (9-bit UART Mode)

9-bit UART mode can be specified by setting SC2MOD<SM1,0> to 11. In this mode, parity bit cannot be added.

For transmission, the MSB (9th bit) is written in SC2MOD <TB8>, while in receiving it is stored in SC2CR<RB8>. For writing and reading the buffer, the MSB is read or written first then SC2BUF.

Wake-up function

In 9-bit UART mode, the wake-up function of slave controllers is enabled by setting SC2MOD<WU> to 1. The interrupt INTRX occurs only when<RB8> = 1.



Note: TxD pin of the slave controllers must be in open drain output mode with ODCR1.

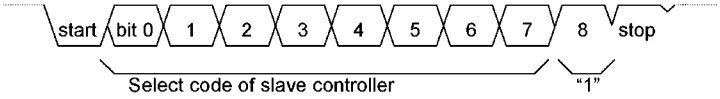
Figure 3.17.10 Serial Link Using Wake-Up Function

Protocol

① Select the 9-bit UART mode for the master and slave controllers.

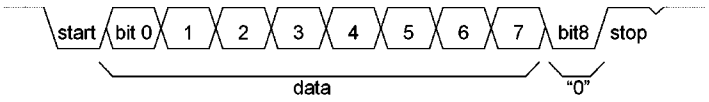
② Set SC2MOD<WU> bit of each slave controller to "1" to enable data receiving.

③ The master controller transmits one-frame data including the 8-bit select code for the slave controllers. The MSB (bit 8)<TB8> is set to 1.



④ Each slave controller receives the above frame, and clears WU bit to "0" if the above select code matches its own select code.

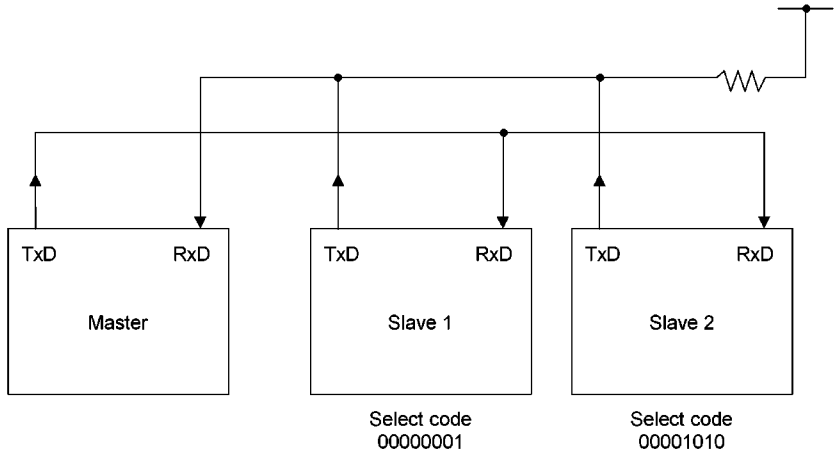
⑤ The master controller transmits data to the specified slave controller whose SC2MOD<WU> bit is cleared to "0". The MSB (bit 8)<TB8> is cleared to 0.



⑥ The other slave controllers (with the <WU> bit remaining at "1") ignore the receiving data because their MSBs (bit 8 or <RB8>) are set to 0 to disable the interrupt INTRX.

The slave controllers (<WU> = 0) can transmit data to the master controller, and it is possible to indicate the end of data receiving to the master controller by this transmission.

Setting example: To link two slave controllers serially with the master controller, and use the internal clock fc/2 as the transfer clock.



- Setting the master controller

Main

```
P7FC      ← – – – – – X X 1 }
P7CR      ← – – – – – – 0 1 }  Select P70 as TxD pin and P71 as RxD pin.
INTRXTX   ← 0 1 0 0 0 1 0 1    Enable INTTX and set the interrupt level 4.
                               Enable INTTX and set the interrupt level 5.
SC2MOD    ← 1 0 1 0 1 1 1 0    Set fc/2 as the transmission clock in 9–bit UART mode.
SC2BUF    ← 0 0 0 0 0 0 0 1    Set the select code for slave controller 1.
```

INTTX interrupt

```
SC2MOD    ← 0 – – – – – – –    Sets TB8 to 0.
SC2BUF    ← * * * * * * * *    Set data for transmission.
```

- setting the slave controller 2

Main

```
ODCR1     ← – X – – – X X 1 }
P7FC      ← – X – – – X X 1 }  Select P71 as RxD pin and P70 as TxD pin (open drain output).
P7CR      ← – – – – – – 0 1 }
INTRXTX   ← 0 1 0 1 0 1 1 0    Enable INTRX and INTTX.
SC2MOD    ← 0 0 1 1 1 1 1 0    Set <WU> to 1 in the 9–bit UART transmission mode with transfer clock fc/2.
```

INTRX interrupt

```
Acc       ← SC2BUF
if Acc = Select code
Then SC2MOD ← – – – 0 – – – –  Clear <WU> to 0.
```

3.18 Serial Bus Interface (I2CBUS)

The TMP93C071 has a 1-channel I²C bus I/F (I2CBUS).

I2CBUS operates as a multi-master, and controls two bus I/F (P73/SDA0, P74/SCL0) and (PB4/SDA1, PB5/SCL1)). At the master mode, the I²CBUS can transfer many byte data continuously with the micro DMA.

3.18.1 Configuration

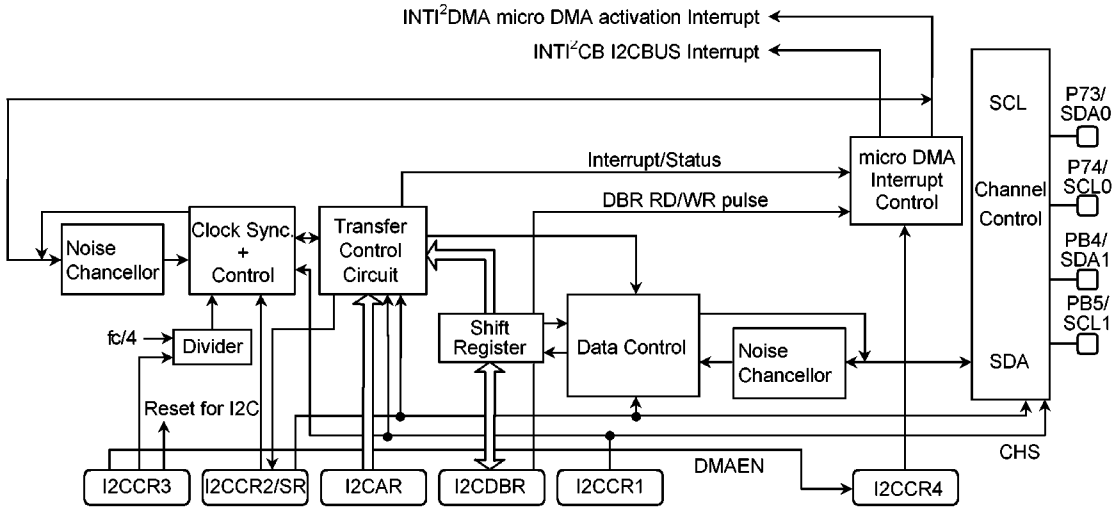


Figure 3.18.1 Serial Bus Interface (I²CBUS)

3.18.2 Serial Bus Interface (I^2CBUS) Control

The following registers are used for control and operation status monitoring when using the serial bus interface (I^2CBUS).

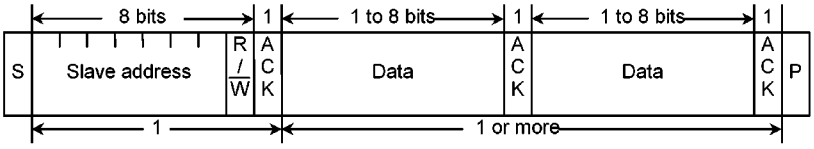
- I^2CBUS control register 1 (I2CCR1)
- I^2CBUS control register 2 (I2CCR2)
- I^2CBUS control register 3 (I2CCR3)
- I^2CBUS control register 4 (I2CCR4)
- I^2CBUS data buffer register (I2CDBR)
- I^2C bus address register (I2CAR)
- I^2CBUS status register (I2CSR)

Refer to Section 3.18.4 I^2Cbus Mode Control.

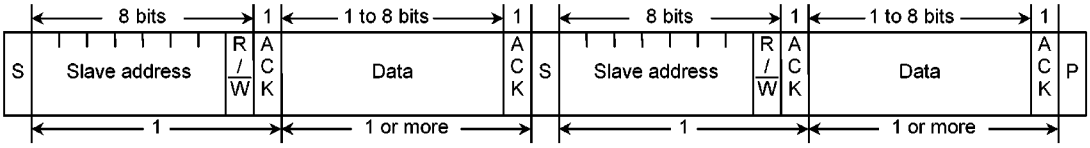
3.18.3 The Data Formats in the I^2CBus

The data formats when using the TMP93C071 in the I^2Cbus are shown below.

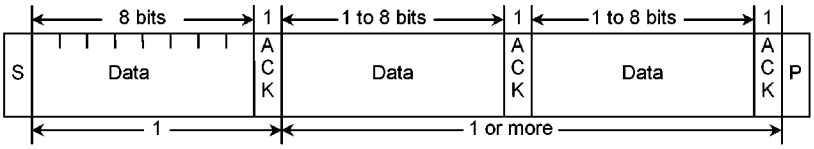
(a) Addressing format



(b) Addressing format (With restart)



(c) Free data format



(Notes) S : Start condition
R/$\overline{W}$: Direction bit
ACK : Acknowlege bit
P : Stop condition

Figure 3.18.2 Data format in the I^2CBus Mode

3.18.4 I²CBus Control

I2CBUS Control Register 1

I2CCR1 (0081H)	7	6	5	4	3	2	1	0
bit Symbol	BC2	BC1	BC0	ACK	CHS	—	—	—
Read/Write	W			R/W	R/W		—	
After reset	0	0	0	0		—	—	—
Function	Number of transferred bits (Note)			Acknowledge mode specification	Channel selection 0: Channel 0 (SDA0, SCL0) 1: Channel 1 (SDA1, SCL1)			
Prohibit read-modify-write								

Acknowledge mode specification

0	Not generate clock pulse for acknowledge signal
1	Generate clock pulse for acknowledge signal

Number of transferred bits

<BC2 to 0>	<ACK> = 0		<ACK> = 1	
	Number of Clock	Bits	Number of Clock	Bits
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

Figure 3.18.3 Register for I²CBus

I²CBUS Control Register 2

I2CCR2 (0084H)

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	—	—
Read/Write	W(R/W Note1)				W			
After reset	0	0	0	1	0	0	0	0
Function	Master / slave selection	Transmitter / receiver selection	Start / stop generation	Cancel INTI2CB request	Serial bus interface operating mode selection (Note 2)		Always set to "00"	

Prohibit read-modify-write

SBI operating mode selection

00	Port mode (I2CBUS output disable)
01	(Reserved)
10	I²C bus mode
11	(Reserved)

Cancel INTI2CB request

0	—
1	Cancel interrupt service request

Start / stop generation

0	Generate the stop condition
1	Generate the start condition

Transmitter / receiver selection

0	Receiver
1	Transmitter

Master / slave selection

0	Slave
1	Master

Note 1: This register functions as the I2CSR by reading.

Note 2: Switch a mode to the port mode after confirming that the bus is free.
Switch a mode to the I²C bus mode after confirming that input signals via port are high level.

Figure 3.18.4 Register for I²CBus

I²CBUS Status Register

I2CSR (0084H)

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Function	Master / slave selection status monitor	Transmitter / receiver selection status monitor	I²C bus status monitor	INTI2CB request status monitor	Arbitration lost detection monitor	Slave address match detection monitor	"GENERAL CALL" detection monitor	Last received bit monitor
Prohibit read-modify-write								

Last received bit monitor

0	Last received bit "0"
1	Last received bit "1"

"GENERAL CALL" detection monitor

0	—
1	"GENERAL CALL" detected

Slave address match detection monitor

0	—
1	Slave address match or "GENERAL CALL" detected

Arbitration lost detection monitor

0	—
1	Arbitration lost detected

INTI2CB request status

0	Interrupt service requested
1	Interrupt service canceled

I²C bus status monitor

0	Bus free
1	Bus busy

Transmitter / receiver selection status monitor

0	Receiver
1	Transmitter

Master / slave selection status monitor

0	Slave
1	Master

Note: Bits 7 to 2 of this register function as the I2CCR2 by writing.

Figure 3.18.5 Register for I²CBus

I²CBUS control register 3

I2CCR3 (0085H)	7	6	5	4	3	2	1	0
bit Symbol	DMAEN	DV4	DV3	DV2	DV1	DV0	—	SWRST
Read/Write	R/W	W						R/W
After reset	0	0	0	0	0	0	*	0
Function	micro DMA transfer mode is enabled 0: Disable 1: Enable	Setting N-divider for generating transfer clock source. 00000: (Reserved) 10000: 74.6 (n = 16) 00001: (Reserved) 10001: 70.4 (n = 17) 00010: 454.5 (n = 2) 10010: 66.7 (n = 18) 00011: 333.3 (n = 3) 10011: 63.3 (n = 19) 00100: 263.2 (n = 4) 10100: 60.2 (n = 20) 00101: 217.4 (n = 5) 10101: 57.5 (n = 21) 00110: 185.2 (n = 6) 10110: 54.9 (n = 22) 00111: 161.3 (n = 7) 10111: 52.6 (n = 23) 01000: 142.9 (n = 8) 11000: 50.5 (n = 24) 01001: 128.2 (n = 9) 11001: 48.5 (n = 25) 01010: 116.3 (n = 10) 11010: 46.7 (n = 26) 01011: 106.4 (n = 11) 11011: 45.0 (n = 27) 01100: 98.0 (n = 12) 11100: 43.5 (n = 28) 01101: 90.9 (n = 13) 11101: 42.0 (n = 29) 01110: 84.7 (n = 14) 11110: 40.7 (n = 30) 01111: 79.4 (n = 15) 11111: 39.4 (n = 31) (kHz) @20 MHz							I²CBUS Software Reset

I²CBUS Software Reset

0	(Initial state)
1	Initialize I²CBUS block (After initialize I²CBUS, SWRST is automatically cleared to 0.)

Figure 3.18.6

I²CBUS control register 4

I2CCR4 (0086H)

	7	6	5	4	3	2	1	0
bit Symbol	RSTR	T/$\overline{R}$	DMAEND	RCONT	C3	C2	C1	C0
Read/Write	W	R/W	R	R/W	R/W			
After reset		0	0	0	0	0		0
Function	Continuous transfer command at micro-DMA transfer mode.	micro DMA transfer Transmit/ Receive mode selection.	micro DMA transfer end flag	Acknowl-edge output control for the last byte on micro DMA receive mode	Setting the number of transfer byte 0000: 16 byte 1000: 8 byte 0001: 1 byte 1001: 9 byte 0010: 2 byte 1010: 10 byte 0011: 3 byte 1011: 11 byte 0100: 4 byte 1100: 12 byte 0101: 5 byte 1101: 13 byte 0110: 6 byte 1110: 14 byte 0111: 7 byte 1111: 15 byte (slave address + R/ $\overline{W}$) is not contained. The value of (C3-0) is decreased by each micro DMA transfer (I2CDBR access). This value must be set after (slave address + R/ $\overline{W}$) has been set to I2CDBR.			

Acknowledge output control for the last byte on micro DMA receive mode.

0	Acknowledge output for the last byte: "High" (the others: "Low")
1	Acknowledge output for the last byte: "Low" (receiving is continued.)

Set to "1" if the reception is continued after the reception for the number of (C3 to C0) is finished.

micro DMA transfer end flag

0	—
1	Transfer end

(set condition)
Transmit: I²CBUS has transmitted the last data.
Receive: micro DMA has read a data after I²CBUS has received the last data.
(reset condition)
Reset, Start/Stop command have been executed.
RSTR command has been executed.

micro DMA transfer Transmit/Receive mode selection.

0	Receive
1	Transmit

Continuous transfer command at micro DMA transfer mode.
(This command is available while DMAEND = 1.)

0	—
1	Restart micro DMA transfer for (C3-0) set at a time. (note)T/$\overline{R}$ bit must be set a previous value.

Figure 3.18.7

Serial Bus Interface Data Buffer Register

I2CDBR (0082H) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (receive) / W (send)							
After reset	Undefined							

Note: When writing the send data, start from the MSB (bit7)

I^2C Bus Address Register

I2CAR (0083H) Prohibit read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	SA7	SA6	SA5	SA4	SA3	SA2	SA1	ALS
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Slave address Selection.							Address recognition mode specification

Address recognition mode specification

0	Save address recognition
1	Non slave address recognition

Figure 3.18.8 Registers for I^2C Bus Mode

(1) Acknowledge mode specification

Set I2CCR1<ACK> to 1 for operation in the acknowledge mode. The TMP93C071 generates an additional clock pulse for an acknowledge signal when operating in the master mode. In the transmitter mode during the clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low level in order to generate the acknowledge signal.

Set <ACK> to 0 for operation in the non-acknowledge mode. The TMP93C071 does not generate a clock pulse for the acknowledge signal when operating in the master mode.

(2) Number of transfer bits

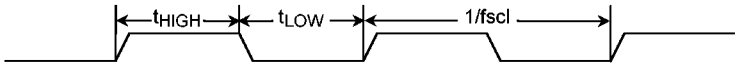
I2CCR1<BC2 to 0> are used to select a number of bits for transmitting and receiving data.

Since <BC2 to 0> are cleared to 000 as a start condition, a slave address and direction bit transmissions are executed in 8 bits. Other than these, <BC2 to 0> retain a specified value.

(3) Serial clock

① Clock source

I2CCR3 <DV4 to 0> are used to select a maximum transfer frequency output on the SCL pin in the master mode.



t_{LOW} = 8N/fc

t_{HIGH} = 8N/fc + 12/fc

fscl = 1/(t_{LOW} + t_{HIGH})

$= \frac{fc}{16N+12}$

I2CCR3 <DV4:0>	n
00000	(reserved)
00001	(reserved)
00010	2
00011	3
⋮	
11100	28
11101	29
11110	30
11111	31

Figure 3.18.9 Clock Souce

② Clock synchronization

In the I^2C bus mode, in order to wired-AND a bus, a master device which drives a clock line to low-level, in the first place, invalidates a clock pulse of another master device which generates a high-level clock pulse. The master device with a high-level clock pulse needs to detect the situation and implement the following procedure.

The TMP93C071 has a clock synchronization function for normal data transfer even when more than one master exists on a bus.

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

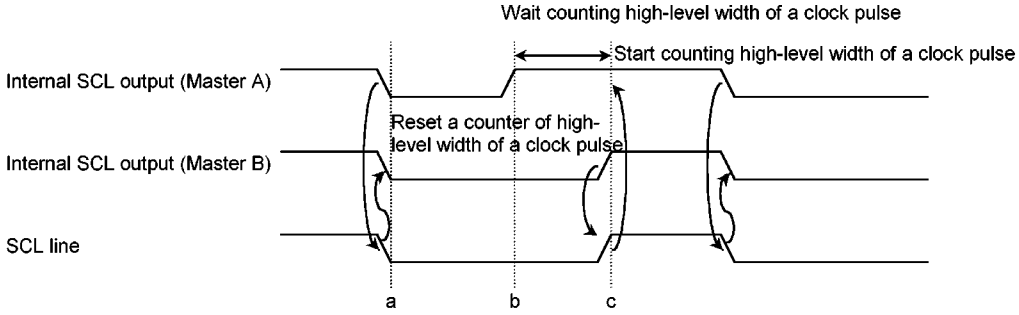


Figure 3.18.10 Clock Synchronization

As Master A drives the internal SCL output to the low level at point "a", the SCL line of the bus becomes the low level. After detecting this situation, Master B resets a counter of an own clock pulse and sets the internal SCL output to the low level.

Master A finishes counting low-level width of an own clock pulse at point "b" and sets the SCL pin to the high level. Since Master B holds the SCL line of the bus at the low level, Master A waits for counting high-level width of an own clock pulse. After Master B sets the internal SCL output to the high level at point "c" and Master A detects the SCL line of the bus at the high level and starts counting high-level of an own clock pulse.

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

(4) Slave address and Address recognition mode specification

When the TMP93C071 is used as a slave device, set the slave address (I2CAR <SA6 to 0>) and I2CAR <ALS>. Set <ALS> to 0 for the address recognition mode.

(5) Master/slave selection

Set I2CCR2 <MST> to 1 for operating the TMP93C071 as a master device. <MST> is cleared to 0 by the hardware after a stop condition on a bus is detected or arbitration is lost.

(6) Transmitter/receiver selection

Set I2CCR2 <TRX> to 1 for operating the TMP93C071 as a transmitter. Set <TRX> to 0 for operation as a receiver. When data with an addressing format is transferred in the slave mode, when a slave address with the same value that an I2CAR or the GENERAL CALL is received (all 8-bit data are 0 after the start condition), <TRX> is set to 1 by the hardware if the direction bit (R/$\overline{W}$) sent from the master device is 1, and is set to 0 by the hardware if the bit is 0. In the master mode, after the acknowledge signal is returned from the slave device, <TRX> is set to 0 by the hardware if a transmitted direction bit is 1, and set to 1 by the hardware if it is 0. When the acknowledge signal is not returned, the current condition is maintained.

<TRX> is cleared to 0 by the hardware after the stop condition on the I^2C bus is detected or arbitration is lost.

(7) Start/Stop Condition generation

When I2CCR2 <BB> is 0, the start condition and 8-bit data are output by writing 1 to I2CCR2<MST, TRX, BB, PIN>. It is necessary to set 1 to I2CCR1<ACK> beforehand.

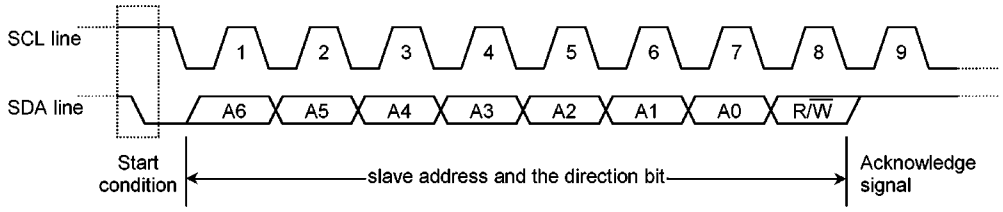


Figure 3.18.11 Start Condition Generation and Slave Address Generation

When I2CCR2 <BB> is 1, a sequence of generating the stop condition is started by writing 1 to <MST, TRX, PIN> and 0 to <BB>. Do not modify the contents of <MST, TRX, BB, PIN> until the stop condition is generated on a bus.

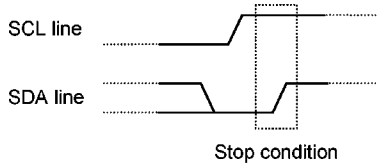


Figure 3.18.12 Stop Condition Generation

The bus condition can be indicated by reading the contents of <BB>. <BB> is set to 1 when the start condition on a bus is detected, and is set to 0 when the stop condition is detected.

(8) Cancel interrupt service request

When the serial bus interrupt request (INTI2CB) occurs, I2CCR2 <PIN> is set to 0. During the time that <PIN> is 0, the SCL pin is driven to the low level.

<PIN> is set to 0 when 1-word of data is transmitted or received. Either writing/reading data to/from the I2CDBR sets <PIN> to 1.

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

In the address recognition mode (I2CAR <ALS> = 0), <PIN> in the slaver mode is set to 0 when the received slave address is the same as the value set at the I2CAR or when the GENERAL CALL is received (all 8-bit data are 0 after the start condition). Although I2CCR2 <PIN> can be set to 1 by the program, <PIN> is not set to 0 when 0 is written.

(9) Serial bus interface operation mode selection

I2CCR2 <SBIM1, 0> is used to specify the serial bus interface operation mode. Set <SBIM1, 0> to "10" when used in the I^2C bus mode after confirming that input signal via port is high level.

Switch a mode to port after making sure 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 the transferred data.

A data on the SDA line is used for bus arbitration of the I^2C bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master A and Master B output the same data until point "a". After Master A outputs "L" and Master B "H", the SDA line of the bus is wire-AND and the SDA line is driven to the low level by Master A. When the SCL line of the bus is pulled up at point "b", the slave device reads data on the SDA line, that is, data in Master A. A data transmitted from Master B becomes invalid. The state in Master B is called "arbitration lost". B master device which loses arbitration releases the SDA 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.

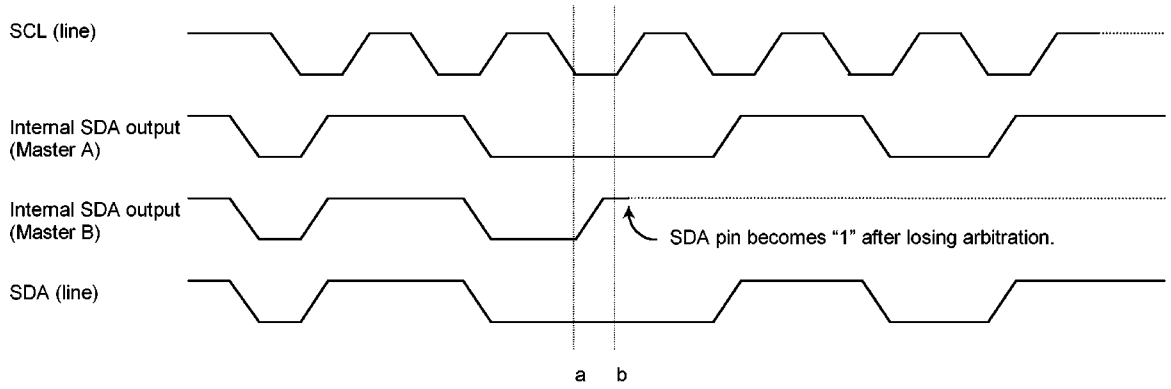


Figure 3.18.13 Arbitration Lost

The TMP93C071 compares levels of the SDA line of the bus with those of the TMP93C071 internal SDA output at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and I2CSR <AL> is set to 1.

When <AL> is set to 1, <MST, TRX> are set to 0, the mode is switched to the slave receiver mode, and the TMP93C071 stops the clock pulse.

<AL> is set to 0 by writing/reading data to/from the I2CDBR or writing data to the I2CCR2.

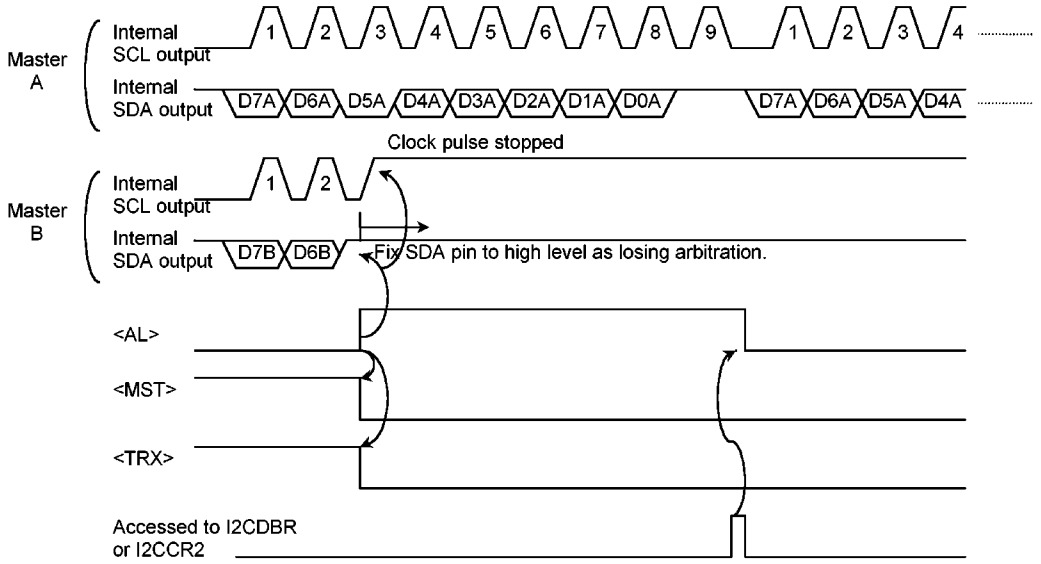


Figure 3.18.14 Example of when TMP93C071 is a Master device

(11) Slave address match detection monitor

I2CSR <AAS> is set to 1 in the slave mode, in the address recognition mode (I2CAR <ALS> = 0) when receiving the GENERAL CALL or the slave address with the same value that is set to the I2CAR. When <ALS> is 1, <AAS> is set to 1 after receiving the first 1-word of data. <AAS> is set to 0 by writing/reading data to/from a data buffer register.

(12) GENERAL CALL detection monitor

I2CSR<AD0> is set to 1 in the slave mode, when the GENERAL CALL is received (all 8-bit data are 0 after the start condition). <AD0> is set to 0 when the start or stop condition is detected on a bus.

(13) Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is sent to I2CSR <LRB>. In the acknowledge mode, immediately after the INTI2CB interrupt request is generated, the acknowledge signal is read by reading the contents of <LRB>.

(14) Software Reset Function

Software reset function is used to initialize the I^2CBUS which is rocked by external noise, etc. When I2CCR3 <SWRST> is set to 1, the internal reset signal pulse is generated and inputted into the I^2CBUS circuit.

All command registers and state registers are initialized to initial values. <SWRST> is automatically set to 0 after the I^2CBUS circuit is initialized.

(15) I²CBUS Data Buffer Register (I2CDBR)

The I2CDBR register can read out the receiving data and write the sending data.
After the start condition generated in the master mode, set the slave address and the direction bit in this register.

(16) I²C BUS Address Register (I2CAR)

I2CAR<SA6 to 0> sets the slave address when the TMP93C071 are operated as the slave devices. Setting I2CAR<ALS> = 0, the slave address output from master device is recognized, and the data format is changed to the addressing format. Setting I2CAR<ALS> = 1, the slave address is not recognized, and the data format is changed to the free data format.

3.18.5 Data Transfer in I²C Bus

(1) Device Initialization

First, set I2CCR3 <DV4:0>, I2CCR1 <ACK, CHS>.
Set the slave address <SA6 to 0> and <ALS> to I2CAR (<ALS> = 0 when the addressing format).
Subsequently, set 0 to<MST, TRX, BB>; 1 to <PIN>; 10 to <SBIM1, 0>; and 0 to bits 0 and 1 in the I2CCR2. The slave receiver mode is set.

(2) Start Condition and Slave Address Generation

① In cace of the master mode

In case of the master mode, the start condition and the slave address are generated according to the following procedures.
First, confirm a bus free status (when <BB>=0).
Set<ACK> to 1 and specify the slave address and the direction bit to be transmitted to the I2CDBR.
When <BB> is 0 and I2CCR2<MST, TRX, BB, PIN> is 1111, the start condition is generated. Subsequently, nine clocks are output from the SCL pin. Eight clocks output the slave address which is set in the I2CDBR and the direction bit. The SDA line is released at the ninth clock and the acknowledge signal is received. The INTI2CB interrupt request occurs at the ninth falling edge of the SCL clock cycle, and <PIN> is 0. The SCL pin is driven to the low level while <PIN> is 0. When an interrupt request occurs, <TRX> changes according to the direction bit only when an acknowledge signal is returned from the slave device.

② In case of the slave mode

In case of the slave mode, the start condition and the slave address are received.
After the start condition output from the master device is received, the slave address and the direction bit are received until the eighth clock. When the GENERAL CALL or the same address as the slave address set in I2CAR is received, the SDA line of the bus is set to the low level in the ninth clock, and the acknowledge signal is output.
At the falling edge of the ninth clock, the INTI2CB interrupt request is generated and <PIN> is set to 0. Operating the slave mode, the SCL line is set to the low level during <PIN>="0". Only when the acknowledge signal is returned from the slave device, <TRX> is changed by the INTI2CB interrupt request according to the receiving direction bit.

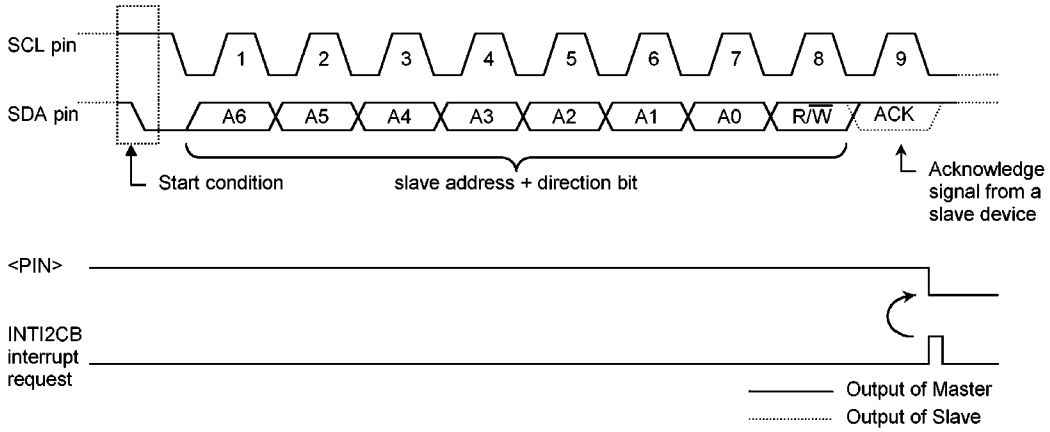


Figure 3.18.15 Start Condition Generation and Slave Address Transfer

(3) 1-word Data Transfer

Test <MST> by the INTI2CB interrupt process after a 1-word data transfer is completed, and determine whether the mode is a master or slave.

① When <MST> is 1 (Master mode).

Test <TRX> and determine whether the mode is a transmitter or receiver.
When <TRX> is 1 (Transmitter mode)
Check I2CSR <LRB>. When <LRB> is 1, a receiver does not request data. Implement the process to generate the stop condition (described later) and terminate data transfer.

When <LRB> is 0, the receiver requests new data. When the next transmitted data is 8-bits, write it to the I2CDBR. When the next transmitted data is other than 8 bits, set I2CCR <BC2 to 0>, set <ACK> to 1, and write the transmitted data to the I2CDBR. After writing the data, <PIN> becomes 1, the serial clock pulse is generated for transferring a new 1-word of data from the SCL pin, and then the 1-word data is transmitted. After the data is transmitted, the INTI2CB interrupt request occurs. <PIN> becomes 0 and the SCL pin is set to the low level. If the data to be transferred is more than one word in length, repeat the procedures from <LRB> test mentioned above.

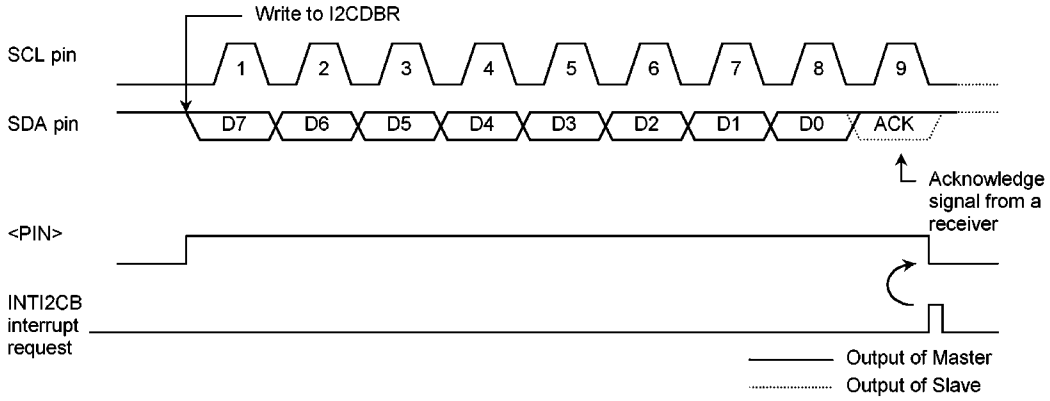


Figure 3.18.16 Example of when <BC2 to 0> = 000 <ACK> = 1 (Transmitter mode)

When <TRX> is 0 (Receiver mode)

When the next transmitted data is 8 bits, write the transmitted data to the I2CDBR. When the next transmitted data is other than 8 bits, set I2CCR1<BC2 to 0> again. Set <ACK> to 1 and read the received data from the I2CDBR to release the SCL line. (The read data is undefined immediately after the slave address is set.) After the data is read, <PIN> becomes 1. The TMP93C071 outputs the serial clock pulse to the SCL pin to transfer new 1-word of data and sets the SDA pin to 0, when the acknowledge signal is set to low level at the final bit.

The INTI2CB interrupt request then occurs and <PIN> becomes 0. The SCL pin is set to the low level. The TMP93C071 outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the I2CDBR.

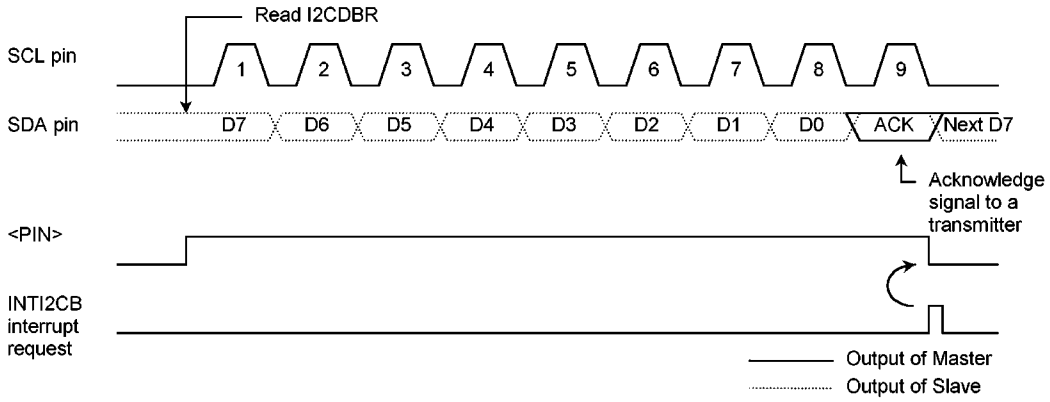


Figure 3.18.17 Example of when <BC2 to 0> = 000, <ACK> = 1 (Receiver mode)

In order to terminate the transmitting data to the transmitter, set <ACK> to 0 before reading data which is 1 word before the last data to be received. The last data does not generate a clock pulse for the acknowledge signal. After the data is transmitted and an interrupt request has occurred, set <BC2 to 0> to "001" and read the data. The TMP93C071 generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA line of the bus keeps the high level. The transmitter receives the high-level signal as the ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and the interrupt request has occurred, the TMP93C071 generates the stop condition and terminates data transfer.

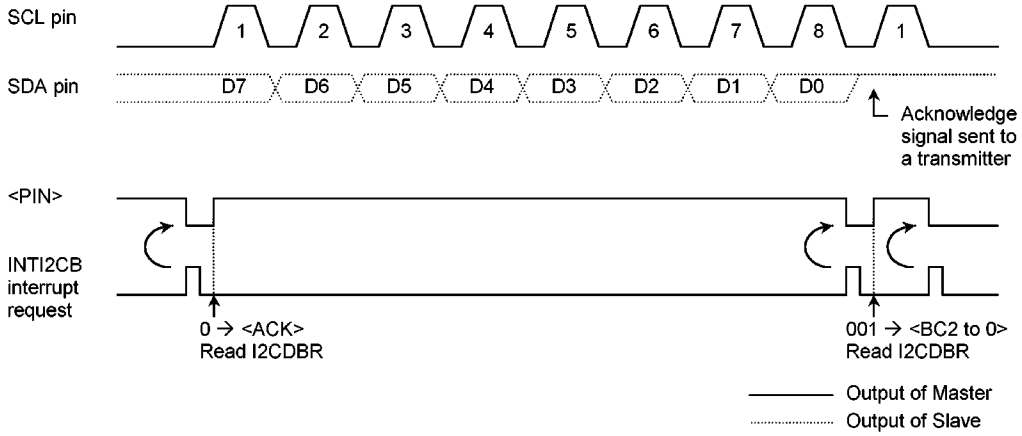


Figure 3.18.18 Termination of data transfer in master receiver mode

② When < MST> is 0 (Slave mode)

In the slave mode, an INTI2CB interrupt request occurs at the following timings.

- When receiving the slave address set in the I2CAR of the TMP93C071 or the GENERAL CALL.
- When data transfer is complete after the slave address or the GENERAL CALL is received.

In the master mode, the TMP93C071 operates in the slave mode if the arbitration lost is detected. The INTI2CB interrupt request occurs when word data transfer terminates after losing arbitration. When the INTI2CB interrupt request occurs, I2CCR2<PIN> is set to 0, and the SCL pin is driven to the low level.

Either reading/writing from/to the I2CDBR or setting<PIN> to 1 releases the SCL pin after taking t_{LOW} time.

In the slave mode, the TMP93C071 operates either in the normal slave mode or in the slave mode after losing arbitration.

The TMP93C071 tests I2CSR<AL>, <TRX>, <AAS> and <AD0> and implements processes according to the conditions listed in the next table.

Table 3.18.1 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	The TMP93C071 loss arbitration when transmitting a slave address and receives a slave address of which the direction bit sent from the master is 1.	Set the number of bits in 1 word to <BC2 to 0> and write transmitted data to the I2CDBR.
	0	1	0	In the slave receiver mode, the TMP93C071 receives a slave address of which the direction bit sent from the master is 1.	
		0	0	In the slave transmitter mode, 1-word data is transmitted.	Check <LRB>. If <LRB> is set to 1, set <PIN> to 1 since the receiver does not request next data. Then, clear <TRX> to 0 release the bus. If <LRB> is set to 0, set the number of bits in a word to <BC2 to 0> and write transmitted data to the I2CDBR since the receiver requests next data.
0	1	1	1/0	The TMP93C071 loses arbitration when transmitting a slave address and receives a slave address or GENERAL CALL of which the direction bit sent from another master is 0.	Read the I2CDBR for setting <PIN> to 1 (reading dummy data) or write 1 to <PIN>.
		0	0	The TMP93C071 loses arbitration when transmitting a slave address or data and terminates transferring word data.	
	0	1	1/0	In the slave receiver mode, the TMP93C071 receives a slave address or GENERAL CALL of which the direction bit sent from the master is 0.	
		0	1/0	In the slave receiver mode, the TMP93C071 terminates receiving of 1-word data.	Set the number of bits in a word to <BC to 0> and read received data from the I2CDBR.

(4) Stop Condition Generation

When I2CSR <BB> is 1, a sequence of generating a stop condition is started by writing 1 to I2CCR2 <MST, TRX, PIN>, and 0 to <BB>. Do not modify the contents of <MST, TRX, BB, PIN> until the stop condition is generated on the bus. When the SCL line of the bus is driven by other device, the TMP93C071 generates the stop condition at rising of SDA pin after the SCL line is released.

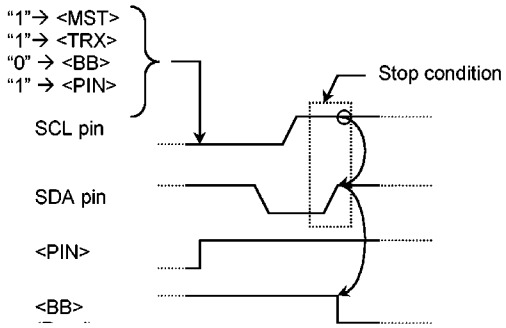


Figure 3.18.19 Stop Condition Generation

(5) Restart

Restart is used to change the direction of data transfer between the master device and the slave device during transferring data. The following explains how to restart when the TMP93C071 is in the master mode.

Specify 0 to I2CCR2 <MST, TRX, BB> and 1 to I2CCR2 <PIN> and release the bus. The SDA pin retains the high level and the SCL pin is released. Since the stop condition is not generated on the bus, the other devices acknowledges the bus to be in a busy state. Check I2CSR <BB> until it becomes "0" to confirm that the SCL pin of the TMP93C071 is released. Check I2CSR <LRB> until it becomes "1" to confirm that the SCL line of the bus is not driven to the low level by other devices. After confirming that the bus stays in a free state, generate the start condition with procedure 3.18.4 (7).

In order to meet setup time to restart, it is necessary to take the waiting time at least 4.7 μs from the time of restarting to confirm that the bus is free until the time to generate the start condition, which is set by software.

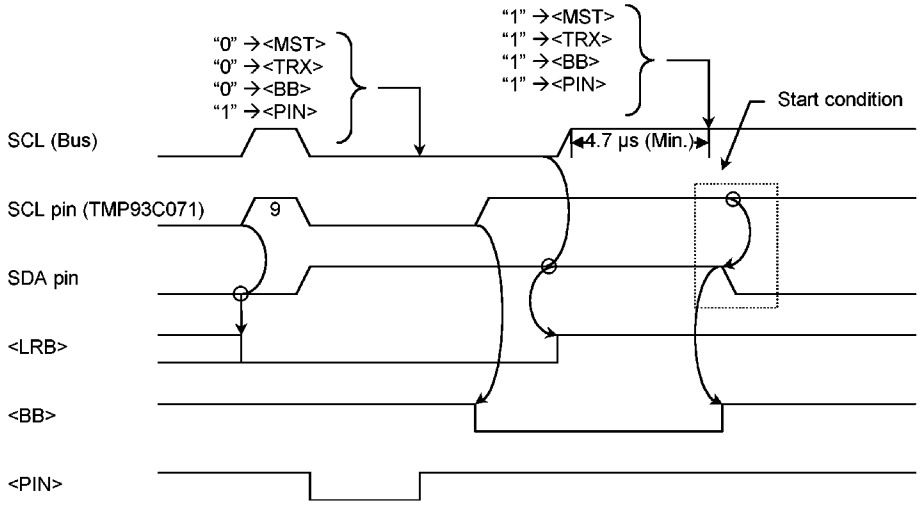


Figure 3.18.20 Timing diagram when restarting the TMP93C071

3.18.6 Micro DMA transfer mode (Only for the master mode)

In addition to the transfer function to describe in 3.18.5,
the I2CBUS has the data transfer function with the micro DMA for the master mode.
(Hereafter, it is called the micro DMA transfer mode)
The I2CBUS becomes the micro DMA transfer mode by setting I2CCR3<DMAEN> to "1".
The micro DMA transfer mode is a mode which the I2CDBR access while data transfers from the start condition to the stop condition is operated by using the micro DMA function that the CPU core builds it into. The micro DMA transfer function can do a continuous transferring in two or more bytes without increasing the CPU load.

(1) Function of the micro DMA transfer mode
The micro DMA transfer mode has the following functions.

· A continuous transferring once is 16 bytes or less. It is possible to continuously transfer them two or more times every 16 bytes or less.

· Two Interrupt request sources (INTI2DMA, INTI2CB)
INTI2DMA: Interrupt request signal only for the micro DMA activation.
The INTI2DMA occurs when the I2CDBR access is necessary.
The reading and writing of the I2CDBR with the micro DMA is activated by the INTI2DMA.

INTI2CB: Interrupt request signal of the transfer end/transfer error.
The INTI2CB is occurred every end of transferring the number of bytes set by I2CCR4<C3:0>.The INTI2CB is also occurred when the I^2CBUS has detected the arbitration lost or missing receiving of the acknowledge from the slaver-receiver(acknowledge error).

· Acknowledge output control for the master-receiver
The acknowledge output which instructs the transmit continuance and end to the slaver-transmitter is controlled. The acknowledge output is controlled by the state of the I2CCR4<C3:0> and <RCONT>.

(2) Setting for the micro DMA

Initializing the micro DMA beforehand when the micro DMA transferring mode is used is necessary. I/O to memory/byte transfe": (for the master- receiver) or Memory to I/O/byte transfer:(for the master-transmitter) are used as transfer mode. "I/O" is set to the I2CDBR address:(0082H). "Memory" is set to the memory address which correspond to the transmission buffer or the reception buffer. The INTI2DMA interrupt is enabled and the vector value "0FH" which corresponds to the INTI2DMA is set in micro DMA start vector register (DMA0V-DMA3V) so that the micro DMA operates for the INTI2DMA interrupt request.

Example 1): The master-transmitter with the micro DMA channel 0

```
LD    XWA, 100H;
LDC   DMAS0, XWA; transmission buffer = 100H
LD    XWA, 82H
LDC   DMAD0, XWA; I2CDBR address
LD    WA, 0000H
LDC   DMAC0, WA; transfer counter=65536 times *1)
LD    A, 04H
LDC   DMAM0, A; "Memory to I/O" "byte transfer"
LD    DMA0V, 0FH; INTI2DMA vector value *2)
LD    INTTG1DMA, 6xH; INTI2DMA interrupt enable (Level 6)
```

Example 2): The master-receiver with the micro DMA channel 1

```
LD    XWA, 82H
LDC   DMAS1, XWA; I2CDBR address
LD    XWA, 200H;
LDC   DMAD1, XWA; reception buffer=200H
LD    WA, 0000H
LDC   DMAC1, WA; transfer counter=65536 times *1)
LD    A, 00H
LDC   DMAM1, A; "I/O to Memory" "byte transfer"
LD    DMA1V, 0FH; INTI2DMA vector value *2)
LD    INTTG1DMA, 6xH; INTI2DMA interrupt enable (Level 6)
```

*1) The transfer counter value of the micro DMA is decreased every micro DMA transfer.
When this value becomes 0, the interrupt process branches out into the general-purpose interrupt process. The micro DMA transfer mode of the I2CBUS does not use this transfer counter. Therefore, the micro DMA transfer counter should be set regularly so that the general-purpose interrupt process is not occurred, or it should be set that the general-purpose interrupt process can be normally ended.

*2) The INTI2DMA is used for both transmitter and receiver.
When the transmission and the reception are processed with the different micro DMA channel such as example 1) and 2), Please do not set the micro DMA start vector register corresponding to respectively at the same time .
Only the micro DMA start vector register on the actual processing side has to be set.
On the other hand, the micro DMA start vector register on the other side has to be set to "00H" and prohibit from activation of the micro DMA.

(3) Procedure of the master transmission

When the micro DMA transfer mode is used, the I2CCR1<ACK> must be set to 1, and the clock pulse for the acknowledge is generated.

After initializing the I^2CBUS and the micro DMA, (slave address + R/$\overline{W}$ = 0) is set in the I2CDBR.

When the I2CCR3<DMAEN> is set to "1", The I2CBUS operates as the micro DMA transfer mode, and the I2CCR4 can be set to any value. The I2CCR4<T/$\overline{R}$ > is set to 1, the micro DMA transfer mode is set to the transmit mode, and the number of bytes of transferring data is set in the I2CCR4<C3 to 0> at the same time.

After checking bus free, Sending of the start condition and slave address begins by setting the start command in the I2CCR2.

If the acknowledge from the slaver can be received, the transmit data of the first byte is transferred to the I2CDBR with the micro DMA by generating the INTI2DMA interrupt.

As for each byte following transferring, the micro DMA transfer is repeated by generating the INTI2DMA interrupt whenever the acknowledge from the slaver is received.

The value of the I2CCR4<C3 to 0> is decreased at the time of each transmit data writing in the I2CDBR. When transferring the number of bytes set in I2CCR4<C3 to0> ends, the I2CCR4<DMAEND> is set to 1, and the INTI2CB interrupt is occurred. (The INTI2DMA interrupt is not occurred.)

When transferring it continuously, the I2CCR4<RSTR> is set to 1 at the same time as setting the next transferred number of bytes in I2CCR4<C3 to 0>.

The micro DMA transfer is restarted with the INTI2DMA interrupt by setting the I2CCR4<RSTR> to 1. At this time, the I2CCR4<DMAEND> is cleared to 0.

Setting the I2CCR4<RSTR> is effective when the I2CCR4<DMAEND> is on 1.

After transferring a setting byte ends, the I^2CBUS transfer is ended with the stop condition by setting the

stop command in the I2CCR2.

When the arbitration lost is detected or the acknowledge from the slaver cannot be received while transferring the number of setting bytes from the slave address sending, the INTI2CB interrupt is occurred instead of the INTI2DMA interrupt. The transfer end and transfer error are distinguished by reading I2CSR<AL, LRB> and I2CCR4<DMAEND> in the INTI2CB interrupt routine.

When the transfer error is occurred, the micro DMA transfer mode is released by setting the I2CCR3<DMAEN> to 0. When the I2CCR3<DMAEN> is set to 0, the I2CCR4 register is initialized. In case of the acknowledge error, transferring is ended by generating the stop condition with the I2CCR2. In case of the arbitration lost, the I^2CBUS is operated as a slaver.

(4) Procedure of the master reception

When the micro DMA transfer mode is used, the I2CCR1<ACK> must be set to 1, and the clock pulse for the acknowledge is generated.

After initializing the I^2CBUS and the micro DMA, (slave address + R/$\overline{W}$ = 1) is set in the I2CDBR.

When the I2CCR3<DMAEN> is set to 1, The I^2CBUS operates as the micro DMA transfer mode, and the I2CCR4 can be set to any value. The I2CCR4<T/$\overline{R}$ > is set to 0, the micro DMA transfer mode is set to the receive mode, and the number of bytes of transferring data is set in the I2CCR4<C3 to 0> at the same time. At this time, when transferring is ended only by receiving the number of setting bytes, I2CCR4<RCONT> is set to 0. When the reception is continued up to the next number of transferring bytes, I2CCR4<RCONT> is set to 1.

After checking bus free, Sending of the start condition and slave address begins by setting the start command in the I2CCR2.

If the acknowledge from the slaver can be received, the clock output restarts automatically and the first byte data reception begins. At each byte reception end, the INTI2DMA interrupt is occurred, and the reception data is taken out from the I2CDBR by the micro DMA.

The acknowledge is sent at each acknowledge timing from the 1st byte ahead of the final byte .

The output of the acknowledge is controlled by the condition of <RCONT> at the acknowledge timing of the final byte. When the <RCONT> is set to 0, the acknowledge of the final byte is not sent and the slaver-transmitter ends the transmission. When the <RCONT> is set to 1, the acknowledge of the final byte is sent and the slaver-transmitter continues transmitting.

The value of the I2CCR4<C3 to 0> is decreased at the time of each reading the reception data from the I2CDBR. When transferring the number of bytes set in I2CCR4<C3 to 0> ends, the I2CCR4<DMAEND> is set to 1 and the INTI2CB interrupt is occurred. (The INTI2DMA interrupt is not occurred)

When transferring it continuously, the I2CCR4<RSTR> is set to "1" at the same time as setting the next transferred number of bytes to the I2CCR4<C3 to 0>. The clock for the next byte is restarted by setting the I2CCR4<RSTR> to 1. At each time the byte receive ends, the INTI2DMA interrupt is occurred and the micro DMA transfer is restarted.

At this time, the I2CCR4<DMAEND> is cleared to 0. Setting the I2CCR4<RSTR> is effective when the I2CCR4<DMAEND> is on 1.

After transferring a setting byte ends, transferring is ended with the stop condition by setting the stop command in the I2CCR2.

When the arbitration lost is detected or the acknowledge from the slaver cannot be received while transferring the slave address byte, the INTI2CB interrupt is occurred instead of the INTI2DMA interrupt. The transfer end and transfer error are distinguished by reading I2CSR<AL, LRB> and I2CCR4<DMAEND> in the INTI2CB interrupt routine.

When the transfer error is occurred, the micro DMA transfer mode is released by setting the I2CCR3<DMAEN> to 0. When the I2CCR3<DMAEN> is set to 0, the I2CCR4 register is initialized. In case of the acknowledge error, transferring is ended by generating the stop condition with the I2CCR2. In case of the arbitration lost, the I^2CBUS is operated as a slaver.

(5) Procedure of restart

The restart operation can be done for the micro DMA transfer mode as well as a usual master operation. The procedure from the bus releasing to the restart is the same as 3.18.5 (5).

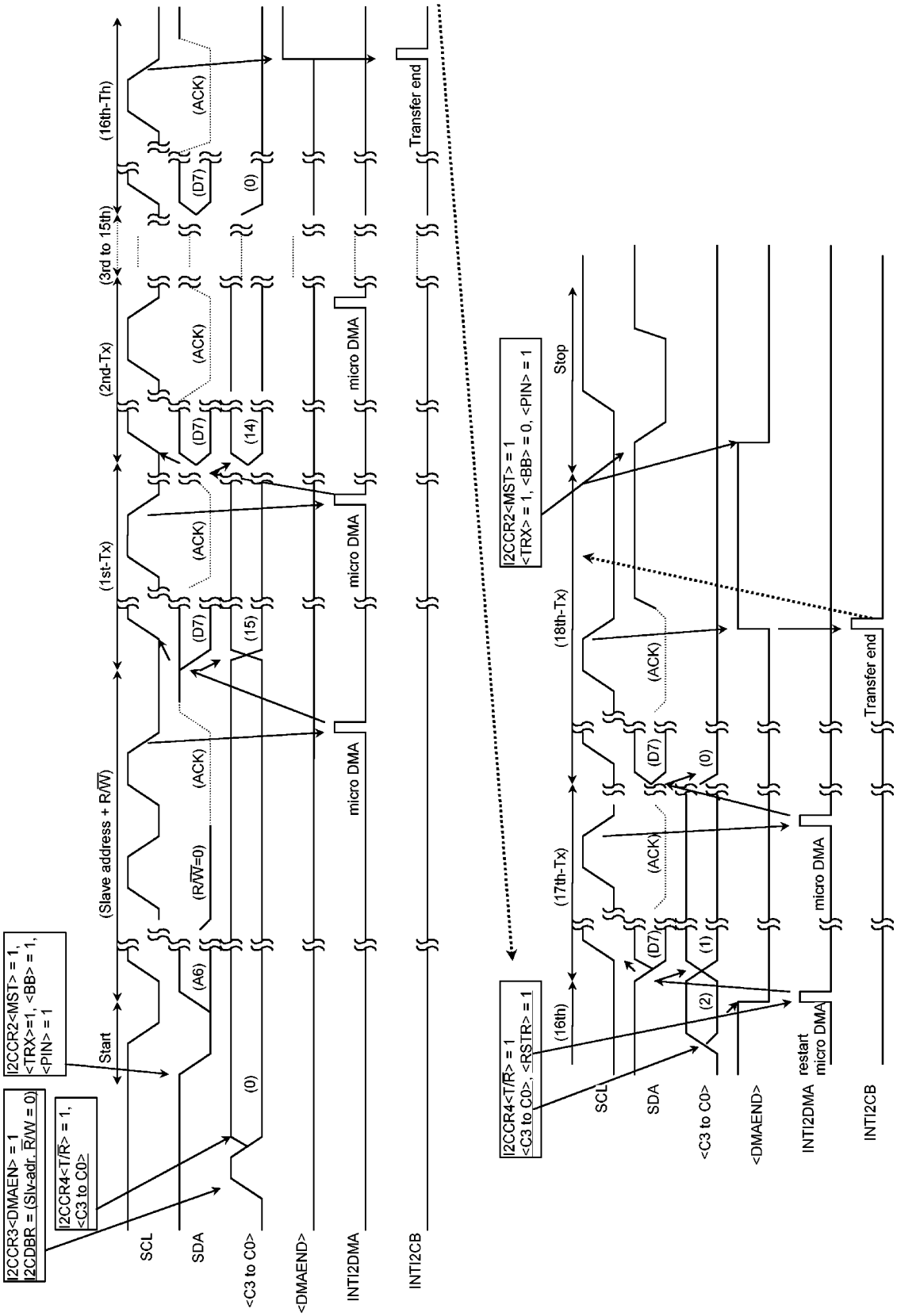


Figure 3.18.6-1 Example for the 18-bytes transmit with micro DMA

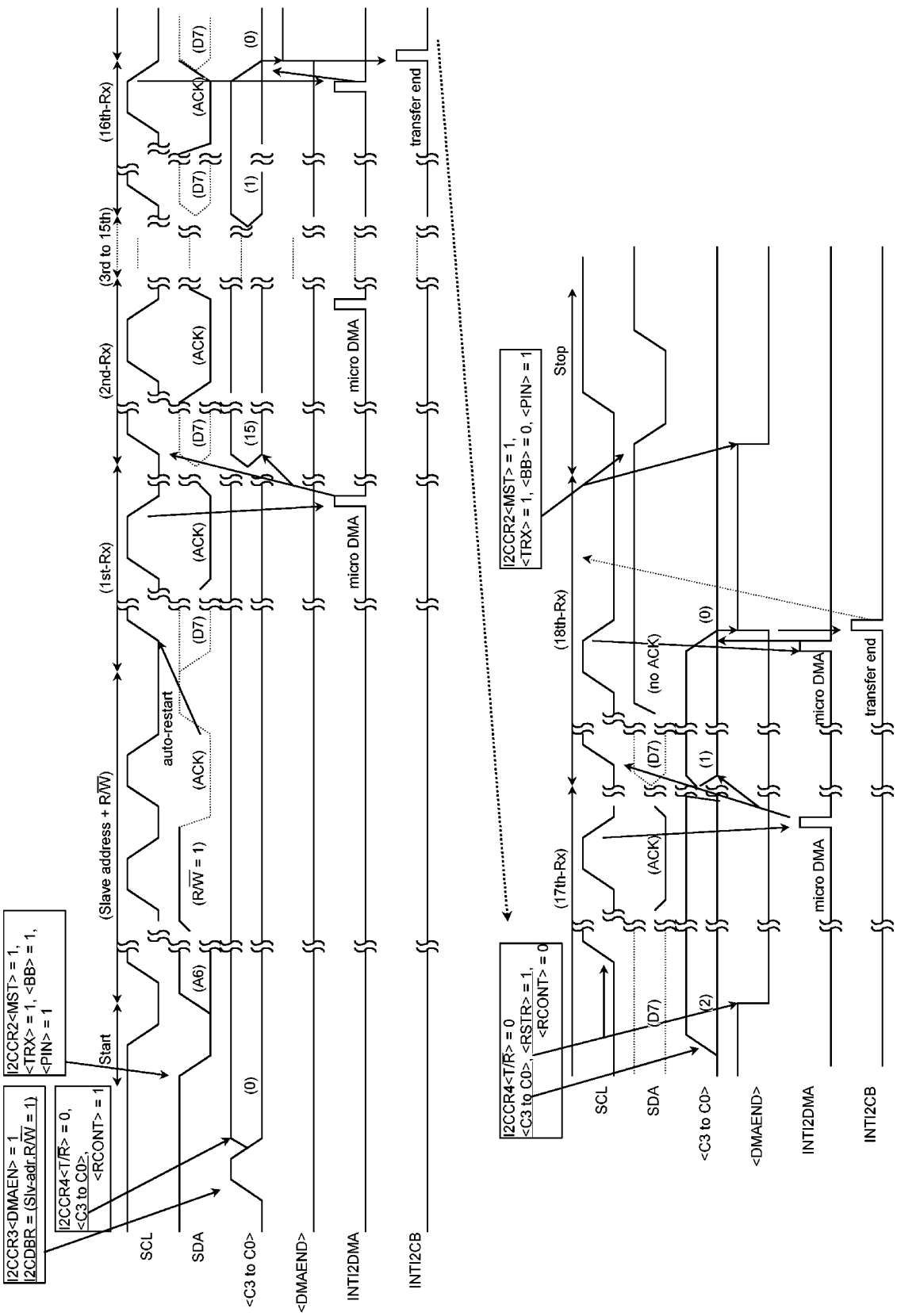


Figure 3.18.6-2 Example for the 18-bytes receive with micro DMA

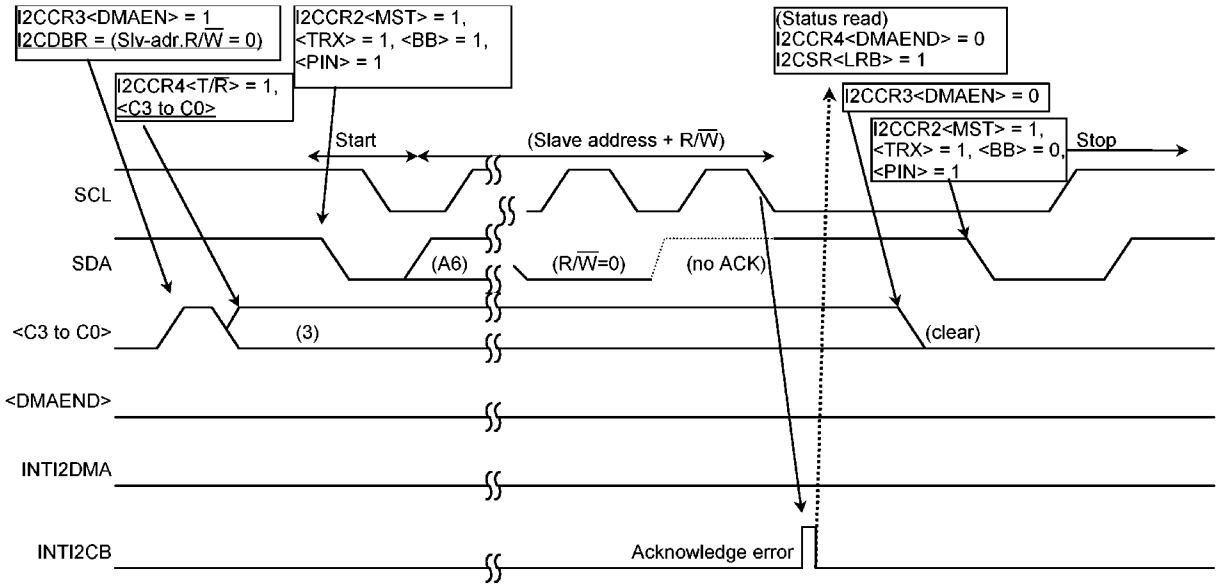


Figure 3.18.6-3 Example for the Acknowledge error at micro DMA transfer mode

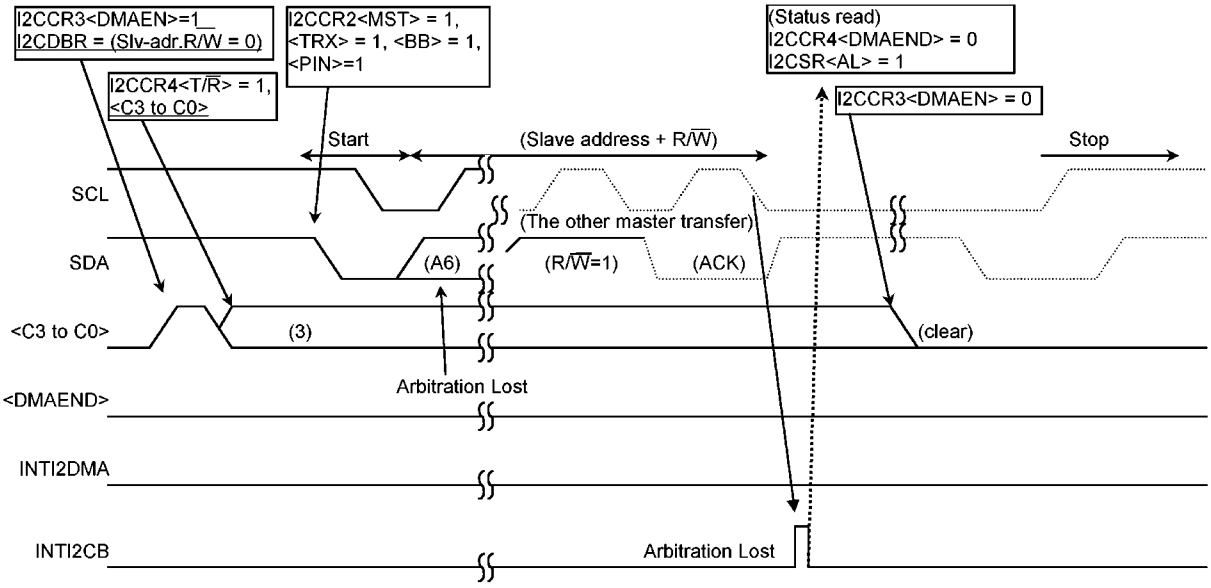


Figure 3.18.6-4 Example for the Arbitration lost at micro DMA transfer mode

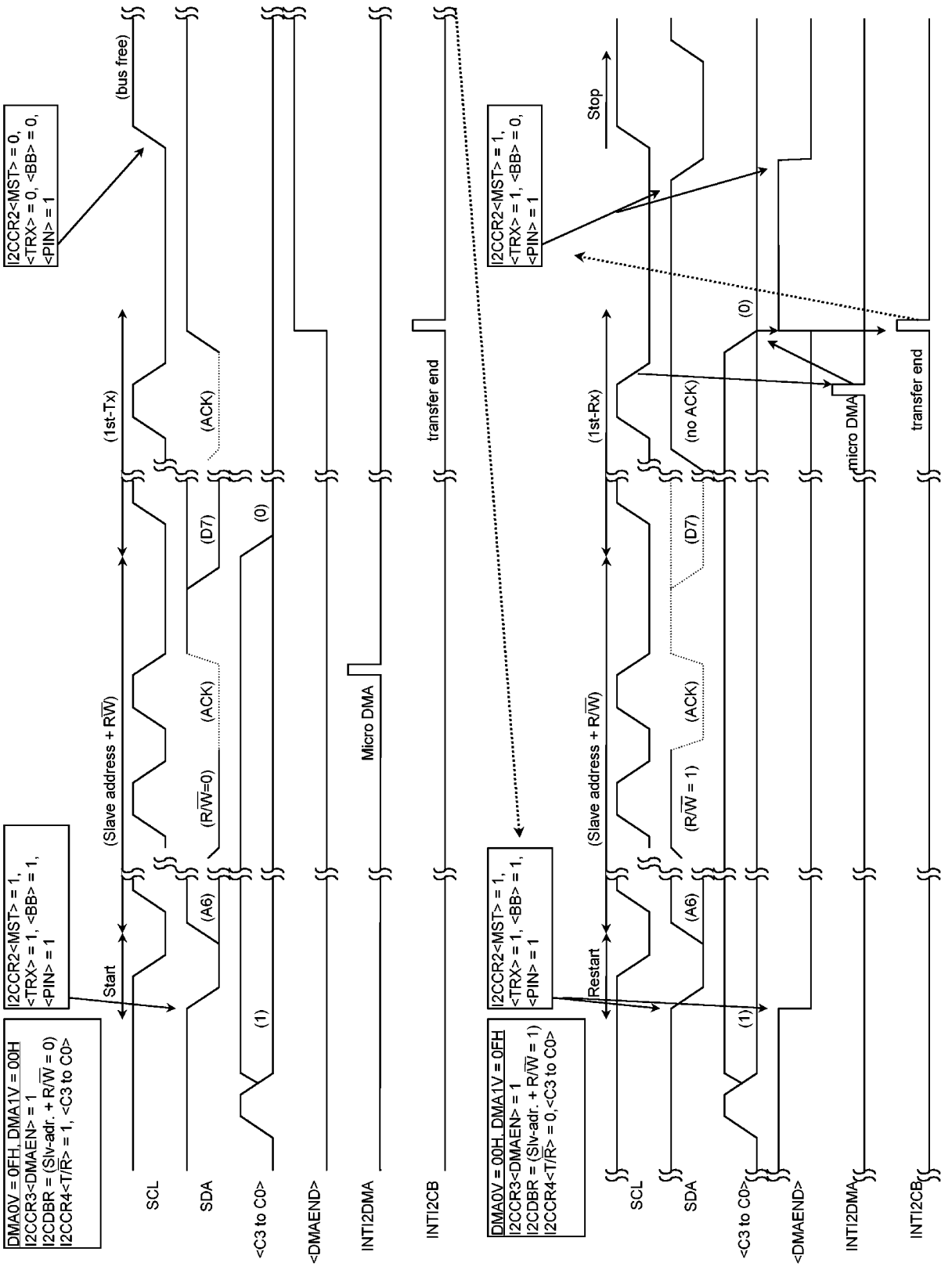


Figure 3.18.6-5 Example for the Restart sequence with micro DMA

3.18.7 Notes when I^2CBUS is used

① When the I^2CBUS is used for the multi master, being end the initialization of the I^2CBUS including setting the port function is necessary before another master begins transferring.

② When the I^2CBUS is used for the multi master, Please write the slave address byte after confirming bus free. Moreover, after writing the slave address byte, please end reconfirming bus free and setting the start command within one cycle at the minimum transferring cycle (start → slave address transmission → stop) assumed doing on the bus.

3.19 8-BIT A/D CONVERSION CIRCUIT (A/D)

The TMP93C071 has an 8-bit A/D conversion circuit of high precision, the successive comparison type with 16-channels analog input. The 8-cannels (AIN3 to AIN10) of 16-channels analog input pin are also used as general purpose input ports (P40 to P47). The others 8-channels (AIN0 to AIN2, AIN11 to AIN15) are also used as I/O port (P55 to P57, PC0 to PC4).

The A/D conversion ends in 9.5 μs (at 20 MHz) from the start of conversion.

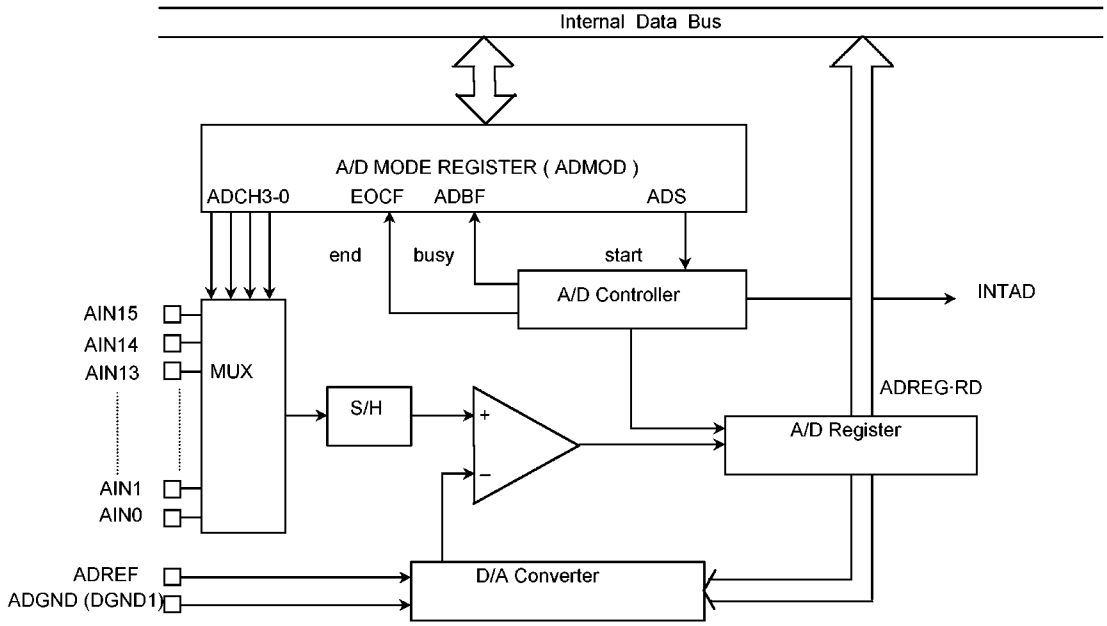


Figure 3.19.1-1 Configuration of 8-bit A/D Conversion Circuit

3.19.1 Operation of A/D Conversion Circuit

(1) A/D conversion reference voltage

The positive electrode of A/D conversion reference voltage connects to ADREF pin, and the negative electrode of A/D conversion reference voltage connects to ADGND (DGND2) pin.

Apply positive of analog reference voltage to the ADREF pin and negative to the ADGND pin. The A/D conversion is carried out by splitting reference voltage between ADREF pin and ADGND pin to bit divided by 256 by ladder resister and making a judgment by comparing it with analog input voltage.

(2) Analog input channels

One of the 16-channels analog input (AIN0 to AIN15) is selected by the A/D conversion control register ADMOD<ADCH3-0>.

The analog input channel selection register ADMOD<ADCH3 to 0> are initialized to “0, 0, 0” by reset operation, then the AIN0 (P55) is selected. When these ports are not used as the analog input ports, these ports can be used as general purpose input ports (Port 4, Port 5, Port C).

When these ports are used as the output port, output level must not be changed during A/D conversion. (the A/D conversion value may be influenced.)

Note: When P55/TI5/AIN0, P56/TI4/AIN1 and P57/TI0/AIN2 are used as an analog input (AIN0, AIN1 and AIN2), TI5, TI4 and TI0 for timer counter must be disabled by TCCR54 and TCCR10. (Select another source clock for timer counter.)

(3) A/D conversion time

The result of A/D conversion is stored into the A/D conversion value register (ADREG) after the passage of 95 states from setting the A/D conversion start register ADMOD<ADS> to 1.

(4) Start A/D conversion start

A/D conversion is started by setting the A/D conversion control register ADMOD<ADS> to "1". After A/D conversion starts, the A/D conversion busy flag ADMOD<ADBF> is set to 1.

Note: If A/D conversion is restarted when <ADBF> is 1, A/D conversion is afraid stopped, after confirming <ADBF> to 0.

(5) A/D conversion end

After A/D conversion ends, the A/D conversion end flag ADMOD<EOCF> which indicates the end of A/D conversion is set to 1, and the interrupt request signal (INTAD) is generated, and the <ADBF> is cleared to 0.

(6) A/D conversion interruption (INTAD)

After A/D conversion ends, the interrupt request signal (INTAD) is generated, and the A/D circuit requests CPU to interrupt. The interrupt request signal (INTAD) is cleared to "0" by reading out the ADREG in program.

(7) Reading of A/D conversion values

The results of A/D conversion is put into the A/D conversion value register (ADREG).
The A/D conversion end flag ADCR<EOCF> is cleared to "0" by reading the ADREG.
The value of the ADREG is an undefined data if the ADREG is read during A/D conversion.
Figure 3.19 (2) shows the timing chart of A/D conversion operation.

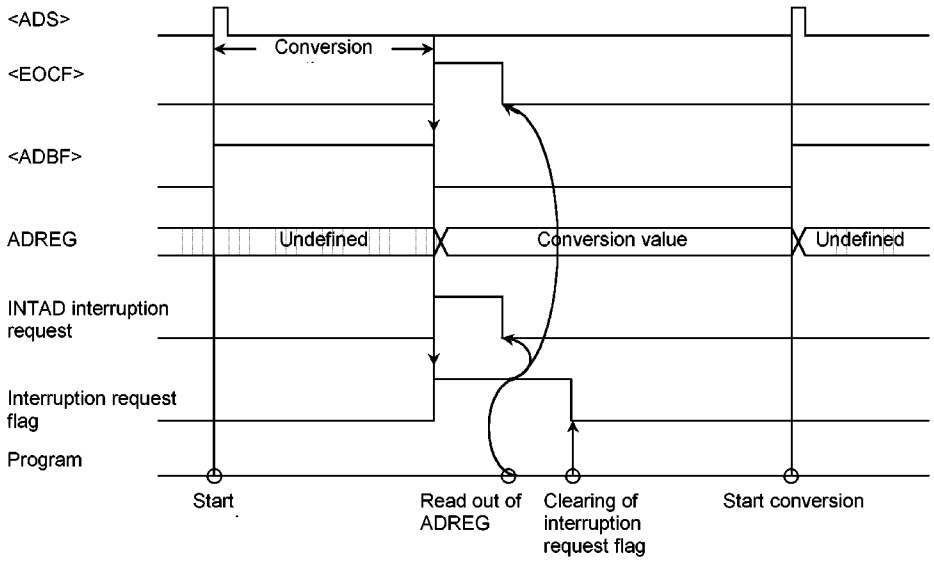


Figure 3.19.2 Shown in Timingchart of A/D Conversion Operation.

Note: Executing the HALT instruction during A/D conversion that conversion operation is forced stop, and result in undefined ADREG value.

At the same time, A/D control register (ADMOD) is initialized to initial value.

3.19.2 Control Register

A/D converter control register

ADMOD (005AH)

	7	6	5	4	3	2	1	0
bit Symbol		EOCF	ADBF	ADS	ADCH3	ADCH2	ADCH1	ADCH0
Read/Write		R	R	R/W	R/W			
After reset	1	0	0	0	0	0	0	0
Function	Always set to 1	A/D conversion flag 0: During or before converting 1: Finish conversion	A/D conversion busy flag 0: free (stop) 1: busy	A/D conversion start /status 0: — 1: A/D conversion start	Analog channel selection 0000: AIN0 0001: AIN1 0010: AIN2 0011: AIN3 0100: AIN4 0101: AIN5 0110: AIN6 0111: AIN7 1000: AIN8 1001: AIN9	1010: AIN10 1011: AIN11 1100: AIN12 1101: AIN13 1110: AIN14 1111: AIN15		

A/D conversion value register

ADREG (005BH)

	7	6	5	4	3	2	1	0
bit Symbol	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
Read/Write	Read							
After reset	*	*	*	*	*	*	*	*

*: unknown after reset

Figure 3.19.3 Registers for A/D converter

3.20 Watchdog Timer (Runaway Detecting Timer)

TMP93C071 contains a watchdog timer of Runaway detecting.

The watchdog timer (WDT) is used to return the CPU to the normal state when it detects that the CPU has started to malfunction (runaway) due to causes such as noise. When the watchdog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU of the malfunction.

Connecting the watchdog timer output to the reset pin internally forces a reset.

This watchdog timer consists of 22-stage binary counters.

These binary counters are also used as a warm-up timer for the internal oscillator stabilization. This is used for releasing the STOP and before changing system clock.

3.20.1 Configuration

Figure 3.20.1 shows the block diagram of the watchdog timer (WDT).

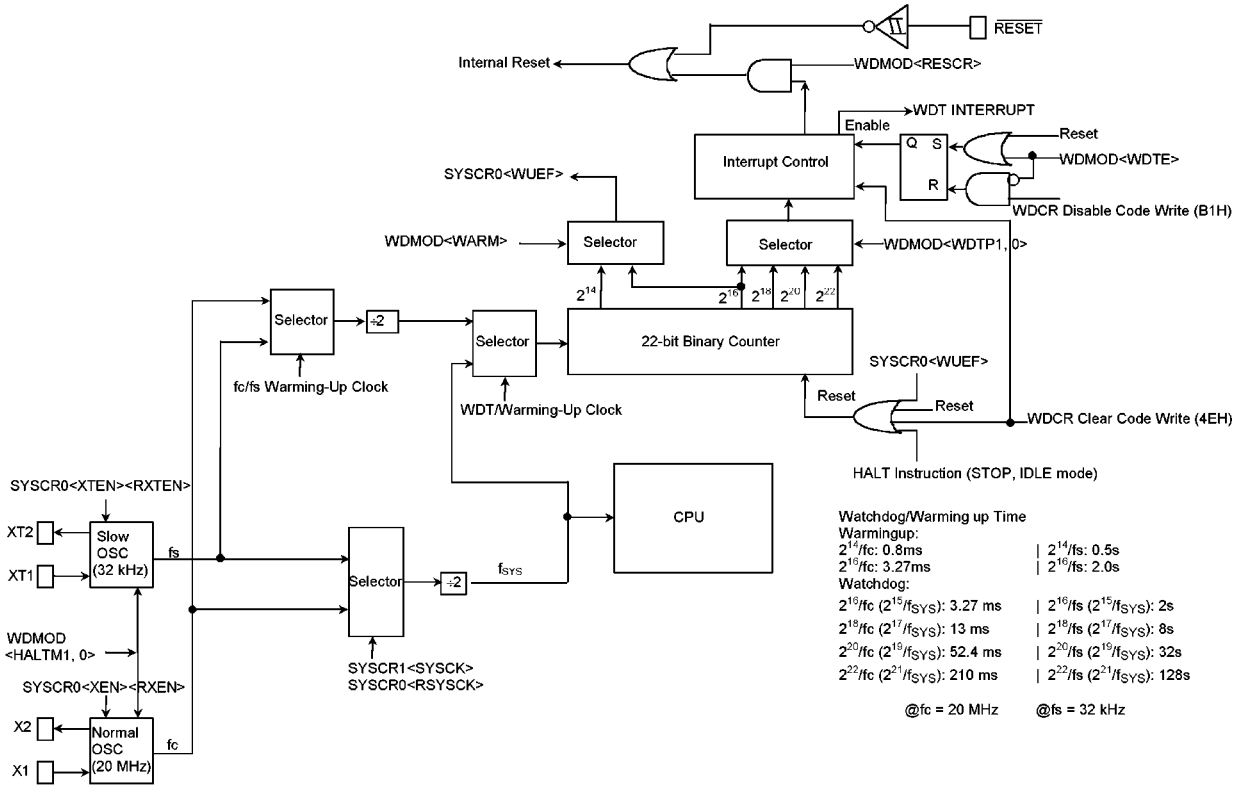


Figure 3.20.1 Block Diagram of Watchdog Timer / Warm-up Timer

The watchdog timer consists of 22-stage binary counters which use System clock (f_{SYS}) as the input clock. The 22-stage binary counter has $f_{SYS}/2^{15}$, $f_{SYS}/2^{17}$, $f_{SYS}/2^{19}$ and $f_{SYS}/2^{21}$ output. Selecting one of the outputs with the WDMOD<WDTP1, 0> register generates a watchdog interrupt and outputs watchdog timer out when an overflow occurs. The binary counter for the watchdog timer should be cleared to "0" with runaway detecting result software (instruction) before an interrupt occurs.

```
(Example)
    LDW    (WDMOD), B100H    ;  disable
    LD     (WDCR), 4EH       ;  write clear code
    SET    7, (WDMOD)        ;  enable again
```

The runaway detecting result can also be connected to the reset pin internally. In this case, the watchdog timer resets itself.

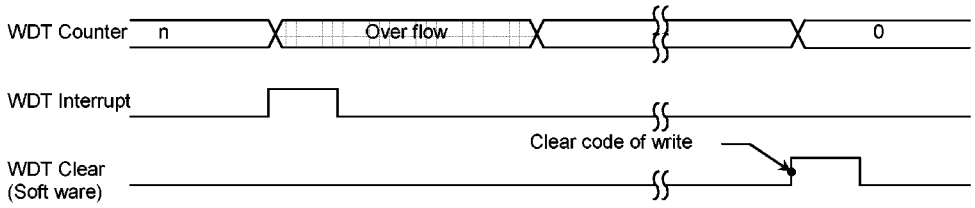


Figure 3.20.2 Normal Mode

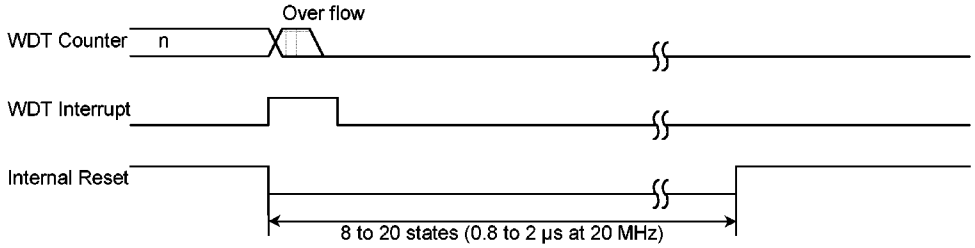


Figure 3.20.3 Reset Mode

For warm-up counter, 2^{14} and 2^{16} output of 22-stage binary counter can be selected using WDMOD<WARM> register.

3.20.2 Control Registers

Watchdog timer WDT is controlled by two control registers WDMOD and WDCR.

(1) Watchdog Timer Mode Register (WDMOD)

① Setting the detecting time of watchdog timer <WDTP>

This 2-bit register is used to set the watchdog timer interrupt time for detecting the runaway. This register is initialized to WDMOD<WDTP1, 0> = 00 when reset.
The detecting time of WDT is shown Table 3.20.1.

② Watchdog timer enable/disable control register <WDTE>

When reset, WDMOD<WDTE> is initialized to 1 enable the watchdog timer.
To disable, it is necessary to set this bit to 0 and write the disable code (B1H) in the watchdog timer control register WDCR. This makes it difficult for the watchdog timer to be disabled by runaway.
However, it is possible to return from the disable state to enable state by merely setting <WDTE> to 1.

③ Watchdog timer out reset connection<RESCR>

This register is used to connect the output of the watchdog timer with $\overline{\text{RESET}}$ terminal, internally. Since WDMOD<RESCR>is initialized to 0 at reset, a reset by the watchdog timer will not be performed.

(2) Watchdog Timer Control Register (WDCR)

This register is used to disable and clear of binary counter the watchdog timer function.

• Disable control

By writing the disable code (B1H) in this WDCR register after clearing WDMOD<WDTE> to 0, the watchdog timer can be disabled.

```
WDMOD   ← 0 - - - - - X X    Clear WDMOD <WDTE> to 0.
WDCR    ← 1 0 1 1 0 0 0 1    Write the disable code (B1H).
```

• Enable control

Set WDMOD<WDTE>to 1.

• Watchdog timer clear control

The binary counter can be cleared and resume counting by writing clear code (4EH) into the WDCR register.

```
WDCR    ← 0 1 0 0 1 1 1 0    Write the clear code (4EH).
```

WDMOD (005CH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	WDT control 0: Disable 1: Enable	Select detecting time 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$ See Table 3.14.1		Warm-up time	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		1: Internally connects WDT out to the reset pin	1: Drive the pin even in STOP mode

[3.3 Standby function]

Watchdog timer Enable / Disable control

0	Disable
1	Enable

Watchdog timer out control

0	Don't care
1	Connects WDT out to a reset

Select the Warm-up time at fc = 20 MHz, fs = 32.768 kHz

WDMOD<WARM>	Changing to NORMAL (fc)	Changing to SLOW (fs)
0(2^{14}/fc or fs)	0.8192 ms	500 ms
1(2^{16}/fc or fs)	3.2768 ms	2000 ms

Figure 3.20.4 Watchdog Timer Mode Register

WDCR (005DH)

	7	6	5	4	3	2	1	0
bit Symbol	–							
Read/Write	W							
After reset	–							
Function	B1H: WDT disable code 4EH: WDT clear code							

Disable/clear WDT

B1H	Disable code
4EH	Clear code
Others	Don't care

Figure 3.20.5 Watchdog Timer Control Register

Table 3.20.1 Watchdog Timer Detecting Time

at fc = 20 MHz, fs = 32.768 kHz

System clock selection <SYSCK>	Watchdog Timer Detecting Time WDMOD<WDTP1, 0>			
	00	01	10	11
1 (fs)	2.000 s	8.000 s	32.000 s	128.000 s
0 (fc)	3.277 ms	13.107 ms	52.429 ms	209.715 ms

3.20.3 Operation

The watchdog timer generates interrupt INTWD after the detecting time set in the WDMOD<WDTP1, 0>. The watchdog timer must be zero-cleared by software before an INTWD interrupt is generated. If the CPU malfunctions (runaway) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter overflows and an INTWD interrupt is generated. The CPU detects malfunction (runaway) due to the INTWD Interrupt and it is possible to return to normal operation by an anti-mulfunction program.

The watchdog timer restarts operation immediately after resetting is released.

The watchdog timer stops its operation in the IDLE1 and STOP modes. In the RUN mode, the watchdog timer is enabled.

However, the function can be disabled when entering the RUN, IDLE2 mode.

Example : ① Clear the binary counter

```
WDCR    ←0 1 0 0 1 1 1 0    Write clear code (4EH).
```

② Set the watchdog timer detecting time to $2^{17} / f_{SYS}$

```
WDMOD   ←1 0 1 - - - X X
```

③ Disable the watchdog timer.

```
WDMOD   ←0 - - - - - X X    Clear WDTE to "0".
WDCR    ←1 0 1 1 0 0 0 1    Write disable code (B1H).
```

④ Set IDLE1 mode.

```
WDMOD   ←0 - - - 1 0 X X    Disables WDT and sets IDLE1 mode.
WDCR    ←1 0 1 1 0 0 0 1
Executes HALT command       Set the HALT mode
```

⑤ Set the STOP mode (warming up time: $2^{16} / f_{SYS}$)

```
WDMOD   ←- - - 1 0 1 X X    Set the STOP mode.
Executes HALT command.      Execte HALT instruction. Set the HALT mode.
```

Note: X; Don't care -; no change

4. Electrical Characteristics(Preliminary)

4.1 Absolute Maximum Rating

Parameter	Symbol	Rating	Unit
Power Supply Voltage	Vcc	-0.5 to 6.5	V
Input Voltage	VIN	-0.5 to Vcc + 0.5	V
(Including Open drain ports)	VOUT	-0.5 to Vcc + 0.5	V
Output Current (Per 1 pin)	IOL	3.2	mA
Output Current (Per 1 pin)	IOH	-3.2	mA
Output Current (total)	ΣIOL	120	mA
Output Current (total)	ΣIOH	-120	mA
Power Dissipation (Ta = 70°C)	PD	600	mW
Soldering Temperature (10sec)	TSOLDER	260	°C
Storage Temperature	TSTG	-65 to 150	°C
Operating Temperature	TOPR	-10 to 70	°C

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.

4.2 DC Characteristics 1/2

Ta = -10 to 70°C

Parameter		Symbol	Condition	Min	Typ.	Max	Unit
Power Supply Voltage		Vcc	fc = 4 to 20 MHz	4.5		5.5	V
			fs = 30 to 34 kHz	2.7			
Input Low Voltage	D0-D15	VIL (TTL)	Vcc ≧ 4 .5 V	-0.3		0.8	V
			Vcc < 4.5 V			0.6	
	P4, P6, P9, PA, PB0, 1, PC	VIL1 (CMOS)				0.3 Vcc	
	$\overline{RESET}$, INT0-4, P5, P7, P8, PB2-5	VIL2 (Schmit)	Vcc = 2.7 to 5.5 V			0.25 Vcc	
	$\overline{EA}$, AM8/16	VIL3 (Fixed)				0.3	
	X1	VIL4 (Xtal)				0.2 Vcc	
Input High Voltage	D0-D15	VIH (TTL)	Vcc ≧ 4.5 V	2.2		Vcc + 0.3	V
			Vcc < 4.5 V	2.0			
	P4, P6, P9, PA, PB0, 1, PC	VIH1 (CMOS)		0.7 Vcc			
	$\overline{RESET}$, INT0-4, P5, P7, P8, PB2-5	VIH2 (Schmit)	Vcc = 2.7 to 5.5 V	0.75 Vcc			
	$\overline{EA}$, AM8/$\overline{16}$	VIH3 (Fixed)		Vcc – 0.3			
	X1	VIH4 (Xtal)		0.8 Vcc			

DC Characteristics 2/2

Ta= −10 to 70°C

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Output Low Voltage	VOL	IOL = 1.6 mA (Vcc = 2.7 to 5.5 V)			0.45	V
Output High Voltage	VOH	IOH = -400 μA (Vcc = 2.7 to 5.5 V)	2.4			V
	VOH1	IOH = -700 μA (Vcc = 4.5 to 5.5V)	4.1			
Input Leakage Current	ILI	0.0 ≦ Vin ≧ Vcc		0.02	±5	μA
Output Leakage Current	ILO	0.2 ≦ Vin ≧ Vcc-0.2		0.05	±10	
Power Down Voltage	VSTOP	VIL2 = 0.2 Vcc, VIH2 = 0.8 Vcc	2.0		6.0	V
$\overline{\text{RESET}}$ Pull Up Resister	R RST	Vcc = 5 V ± 10%	50		150	Ω
		Vcc = 3 V ± 10%	80		200	
Pin Capacitance	CIO	osc = 1 MHz/100 mVpp			10	pF
Schmitt Width $\overline{\text{RESET}}$, INT0-4, P5, P7, P8, PB2-5	VTH			1.0		V
Operating Current (NORMAL)	Icc	Vcc = 5 V ± 10% fc = 20 MHz		28	45	mA
RUN				22	35	
IDLE2				16	25	
IDLE1				3.5	8	
Operating Current (SLOW)		Vcc = 3 V ± 10% fs = 32.768 kHz (Typ: Vcc = 3.0 V)		40	70	μA
RUN				32	45	
IDLE2				27	40	
IDLE1				17	30	
STOP		Vcc = 2.7 to 5.5 V		0.2	10	μA

Note 1: Typical value are for Ta = 25°C and Vcc = 5 V unless otherwise noted.

Note 2: Icc measurement conditions (NORMAL, SLOW).
Only CPU is operational; output pins are open and input pins are fixed.

4.3 A/D Conversion Characteristics

Ta = −10 to 70°C, Vcc = 4.5 to 5.5 V

Parameter	Symbol	Min	Typ.	Max	Unit
Analog Reference Voltage Supply	ADREF	Vcc-1.5	Vcc	Vcc	V
	ADGND	Vss	Vss	Vss	V
Analog Input Voltage Range	VAIN	ADGND	—	ADREF	V
Analog Current for ADREF	IREF	—	1.0	1.5	mA
Error				±3	LSB

4.4 AC Electrical Characteristics(Preliminary)
(Separated Bus)

(1) Vcc = 5 V ± 10% Ta = −10 to 70°C

Parameter	Symbol	Variable		20 MHz		Unit
		Min	Max	Min	Max	
OSC.Period(= X)	tOSC	50	250	50		ns
A0-23 Valid --> $\overline{RD}$ / $\overline{WR}$ Fall	tAC	1.0X − 25		25		ns
$\overline{RD}$ / $\overline{WR}$ Rise --> A0-23 Hold	tCA	0.5X − 25		0		ns
A0-23 Valid --> D0-15 input	tAD		3.5X-65		110	ns
$\overline{RD}$ fall --> D0 − 15 input	tRD		2.5X-60		65	ns
$\overline{RD}$ Low width	tRR	2.5X − 40		85		ns
$\overline{RD}$ rise --> D0 − 15 Hold	tHR	0		0		ns
WR Low Pulse Width	tWW	2.5X − 40		85		ns
D0-15 Valid --> WR Rise	tDW	2.0X − 55		45		ns
$\overline{WR}$ Rise --> D0 − 15 Hold	tWD	0.5X − 15		10		ns
A0-23 Valid --> Port Input	tAPH		2.5X − 120		5	ns
A0-23 Valid --> Port Hold	tAPH2	2.5X + 50		175		ns
$\overline{WR}$ Rise --> Port Valid	tCP		200		200	ns

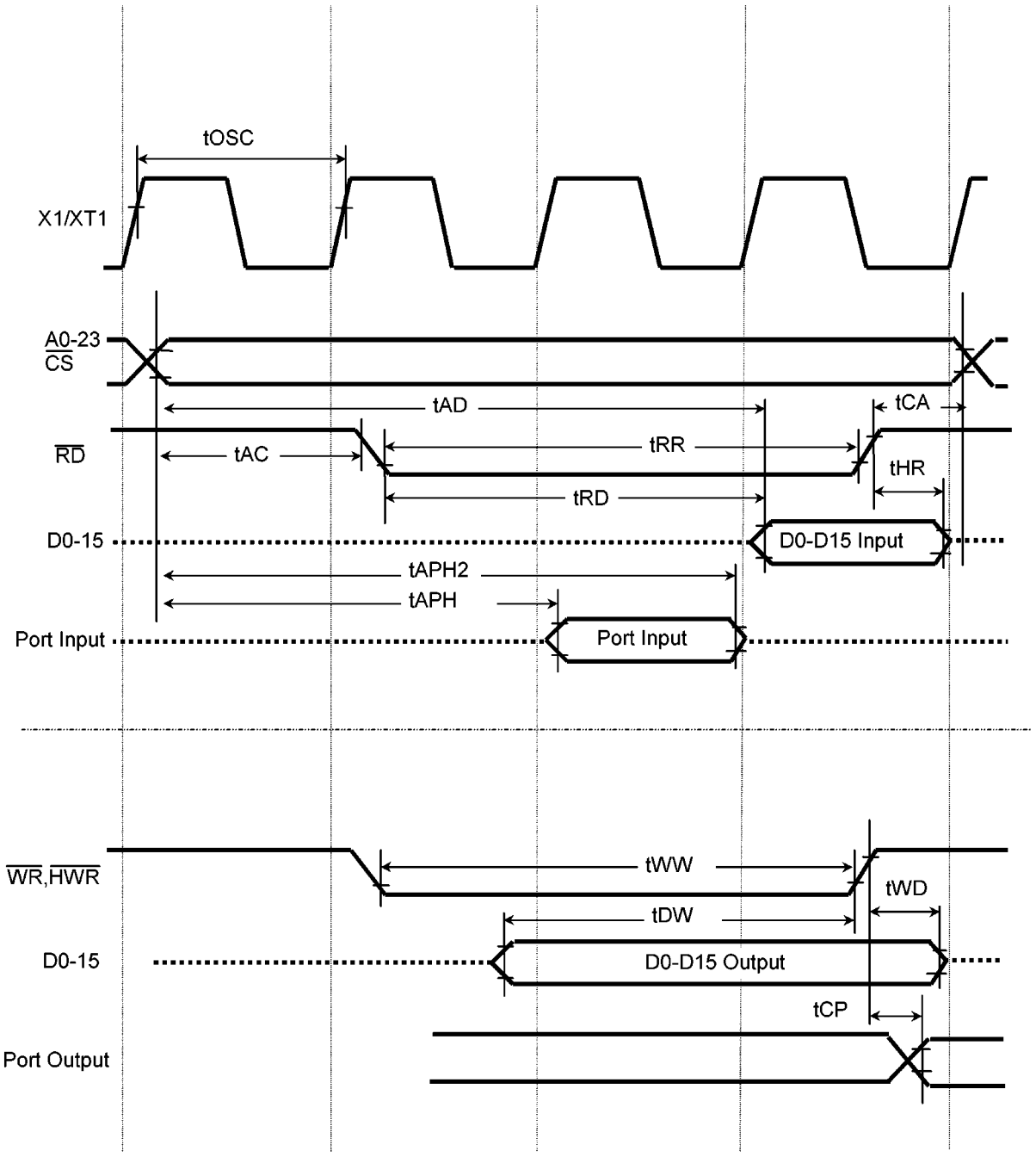
AC Test Condition:
Output Level: High = 2.2 V, Low = 0.8 V CL = 50pF
(D0-D15, A0-23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, CS0-CS2: CL = 100pF)
Input Level: High = 2.4 V / Low = 0.45 V: D0-D15
High = 0.8 Vcc / Low = 0.2 Vcc: (except D0-D15)

(2) Vcc = 3 V ± 10% Ta = −10 to 70°C

Parameter	Symbol	Variable		Unit
		Min	Max	
OSC.Period(= X)	tOSC	29400	33300	ns
A0-23 Valid --> $\overline{RD}$ / $\overline{WR}$ Fall	tAC	1.0X − 50		ns
$\overline{RD}$ / $\overline{WR}$ Rise --> A0 − 23 Hold	tCA	0.5X − 40		ns
A0-23 Valid --> D0 − 15 input	tAD		3.5X − 125	ns
$\overline{RD}$ fall --> D0 − 15 input	tRD		2.5X − 115	ns
$\overline{RD}$ Low width	tRR	2.5X − 40		ns
$\overline{RD}$ rise --> D0 − 15 Hold	tHR	0		ns
WR Low Pulse Width	tWW	2.5X − 40		ns
D0 − 15 Valid --> WR Rise	tDW	2.0X − 120		ns
$\overline{WR}$ Rise --> D0 − 15 Hold	tWD	0.5X − 40		ns
A0 − 23 Valid --> Port Input	tAPH		2.5X − 120	ns
A0 − 23 Valid --> Port Hold	tAPH2	2.5X + 50		ns
$\overline{WR}$ Rise --> Port Valid	tCP		200	ns

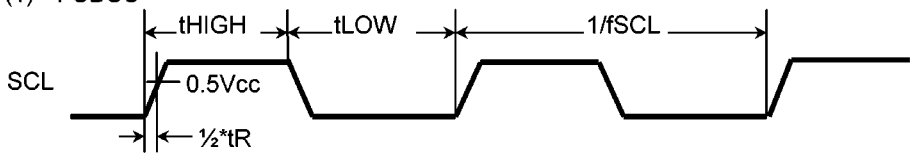
AC Test Condition:
Output Level: High = 0.7 × Vcc, Low = 0.3 × Vcc CL = 50pF
Input Level: High = 0.9 × Vcc, Low = 0.1 × Vcc

(1) Read / Write cycle timing chart [Separated Bus]



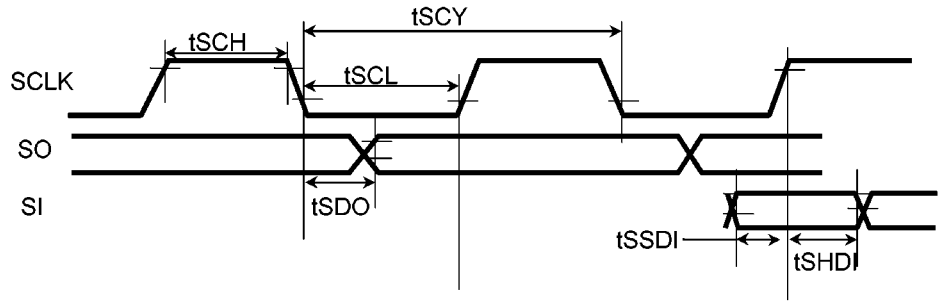
4.5 Serial Channel Timing Chart

(1) I^2CBUS



tLOW = 8N/fc
tHIGH =8N/fc + 12/fc + (1/2*tR)
fSCL = 1/(tLOW + tHIGH)
N = Value of I2CCR3<DV4 – 0>

(2) SIO0,1



SIO AC Electrical Characteristics

(SCLK external input) Ta= –10 to 70°C Vcc = 5 V ± 10%

Parameter	Symbol	Variable		20 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle	tSCY	2^5X (2^4X)		1.6 (0.8)		us
SCLK High pulse width	tSCH	tSCY/2-50 (tSYC/2)		750 (400)		ns
SCLK Low pulse width	tSCL	tSCY/2-50 (tSYC/2)		750 (400)		ns
SCLK shift edge -> SO delay	tSDO		6X + 50		350	ns
SCLK shift edge -> SI setup	tSSDI	-X + 50		0		ns
SCLK shift edge -> SI hold	tSHDI	6X + 50		350		ns

note: These are value when SCxMOD<SIOxE> don't be disabled under transferring.

(SCLK internal output)

Parameter	Symbol	Variable		20 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle	tSCY	2^4X	2^7X	0.8	6.4	us
SCLK High pulse width	tSCH	tSCY/2-50	tSCY/2	350	3200	ns
SCLK Low pulse width	tSCL	tSCY/2-50	tSCY/2	350	3200	ns
SCLK shift edge -> SO delay	tSDO		X + 50		100	ns
SCLK shift edge -> SI setup	tSSDI	X + 50		100		ns
SCLK shift edge -> SI hold	tSHDI	0		0		ns

5. Table of Special Function Registers (SFR; Special Function Register)

The special function registers (SFRs) include the I/O ports and peripheral control registers allocated to the 144-byte addresses from 000000H to 00008FH.

Configuration of the table

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
			bit Symbol							
			Read/Write							
			Value after Reset							
			Function							

Note: Prohibit RMW in the table means that you cannot use RMW instructions on these registers. The LD (transfer) instruction must be used to write for Prohibit RMW registers.

R/W: Read/Write
R: Read-only
W: Write-only
Prohibit RMW: Prohibit read-modify-write
(cannot be used the instruction such as RES, SET, TEST, CHG, STCF, ANDCF, ORCF, XORCF)

Table 5. I/O register address map

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
000000H	(Reserved)	20H	PBFC	40H	PWM1DRL	60H	SC0MOD	80H	INTRTCR
1H	(Reserved)	21H	ODCR3	41H	PWM1DRH	61H	SC0BUF	81H	I2CCR1
2H	(Reserved)	22H	ODCR4	42H	PWM2DRL	62H	SC1MOD	82H	I2CDBR
3H	(Reserved)	23H	TBCMOD	43H	PWM2DRH	63H	SC1BUF	83H	I2CAR
4H	(Reserved)	24H	TBCDR0	44H	PWMDBR	64H	SC2MOD	84H	I2CCR2
5H	(Reserved)	25H	TBCDR1	45H	PWMCR	65H	SC2CR	85H	I2CCR3
6H	P2	26H	TBCIF	46H	CAPINCR0	66H	BR2CR	86H	I2CCR4
7H	(Reserved)	27H	TRUN	47H	CAPINCR1	67H	SC2BUF	87H	PC
8H	(Reserved)	28H	TCCR10	48H	CAPINCR2	68H	B0CS	88H	PCCR
9H	P2FC	29H	TREG0	49H	CAPINCR3	69H	B1CS	89H	(Reserved)
AH	P4	2AH	TREG1L	4AH	CAPFST	6AH	B2CS	8AH	(Reserved)
BH	P5	2BH	TREG1H	4BH	CAP0L	6BH	PAFC	8BH	(Reserved)
CH	P5CR	2CH	TCCR32	4CH	CAP0M	6CH	SYSCR2	8CH	(Reserved)
DH	PRDSEL	2DH	TCCR54	4DH	CAP0H	6DH	SYSCR3	8DH	(Reserved)
EH	P6	2EH	TREG2L	4EH	CAP1L	6EH	SYSCR0	8EH	(Reserved)
FH	P7	2FH	TREG2H	4FH	CAP1H	6FH	SYSCR1	8FH	(Reserved)
10H	P6CR	30H	TREG3L	50H	CAP2L	70H	INT0CP1		
11H	P7CR	31H	TREG3H	51H	CAP2H	71H	INTCP0TG0		
12H	P6FC	32H	TREG4L	52H	CAPCR	72H	INTTG1DMA		
13H	P7FC	33H	TREG4H	53H	RMTCR	73H	INTI2CTBC		
14H	ODCR0	34H	TREG5L	54H	VIVACR1	74H	INTSI0SI1		
15H	ODCR1	35H	TREG5H	55H	VIVACR2	75H	INTRXTX		
16H	P8	36H	TPG0CR	56H	VASSDR	76H	INTVA1		
17H	P9	37H	TPG0L	57H	CSYNCR	77H	INT23		
18H	P8CR	38H	TPG0H	58H	PVCR	78H	INT4T0		
19H	P9CR	39H	TPG0DR	59H	(Reserved)	79H	INTT1T2		
1AH	P9FC	3AH	TPG1L	5AH	ADMOD	7AH	INTT3T4		
1BH	ODCR2	3BH	TPG1H	5BH	ADREG	7BH	INTT5AD		
1CH	PA	3CH	TPG1DR	5CH	WDMOD	7CH	DMA0V		
1DH	PB	3DH	PWMRUN	5DH	WDCR	7DH	DMA1V		
1EH	PACR	3EH	PWM0DRL	5EH	IIMC0	7EH	DMA2V		
1FH	PBCR	3FH	PWM0DRH	5FH	IIMC1	7FH	DMA3V		

System Control/RTC

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
SYSCR0	System Control Register0	00006EH	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	—	—
			R/W	R/W	R/W	R/W	R/W	R/W		
			1	0	1	0	0	0		
			High frequency oscillator (fc: 20 MHz) 0: Stop 1: Oscillation	Low frequency oscillator (fs: 32 kHz) 0: Stop 1: Oscillation	High frequency oscillator (fc) after released STOP mode 0: Stop 1: Oscillation	Low frequency oscillator (fs) after released STOP mode 0: Stop 1: Oscillation	System clock after releasing STOP mode 0: fc (Normal) 1: fs (Slow)	Warming up timer start/ status flag (Write) 0: Don't care 1: Start timer (Read) 0: End of warming up 1: Under warming up		
SYSCR1	System control Register1	00006FH		—	—	—	SYSCK			
			(set to 0)				R/W	(set to 0)	(set to 0)	(set to 0)
			0				0	0	0	0
							system clock 0:fc (Normal) 1:fs (Slow)			
SYSCR2	System Control Register2	00006CH			—	RTCCK	RTCST	RTCIS1	RTCIS0	—
			(set to 0)	(set to 0)		R/W	R/W	R/W		
			0	0		0	0	0		
						RTC clock source select 0: fs (32 kHz) 1:fc (20 MHz)	RTC start control 0: Stop & Counter Clear 1: Start	Interval time control of RTC interrupt 00: fc/2^{17} or fs/2^{15} (6.55 ms) (1 s) 01: fc/2^{18} or fs/2^{16} (13.1 ms) (2 s) 10: fc/2^{16} or fs/2^{14} (3.28 ms) (0.5 s) 11: Reserved [Hz]		
SYSCR3	System Control Register2	00006DH								CLKEN
			(set to 1)	(set to 1)	(set to 1)	(set to 1)	(set to 0)	(set to 0)		R/W
			1	1	1	1	0	0		0
										0: Disable 1: Enable

Watch DogTimer

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
WDMOD	Watch Dog Timer Mode Register	00005CH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
			R/W	R/W		R/W	R/W		R/W	R/W
			1	0		0	0		0	0
			WDT control 0: Disable 1: Enable	WDT detection time 00: 2^{15}/fsys 01: 2^{17}/fsys 10: 2^{19}/fsys 11: 2^{21}/fsys fsys = 1/fc or 1/fs		Warming up timer source clock selection 0: 2^{14}/input frequency 1: 2^{16}/input frequency Input frequency = fc or fs	Standby mode 00: RUN 01: STOP 10: IDLE1 11: IDLE2		0: — 1: Connect internally WDTout to Reset pin	1:Drive the pin in STOP mode
WDCR	Watch Dog Timer Control Register	00005DH	—							
			W							
			—							
			B1H: WDT disable code 4EH: WDT clear code							

Time Base Counter (TBC)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TBCMOD	TBC Mode Register	000023H			TBC1E	TBC0E	INTTBC11	INTTBC10	INTTBC01	INTTBC00
			(set to "0")	(set to "0")	R/W		R/W		R/W	
			"0"	"0"	0		0		0	
		"Prohibit RMW"			INTTBC interrupt enable/disable 00: Disable 01: INTTBC0 10: INTTBC1 11: INTTBC0/1		TBC1 interrupt source clock selection 00: TBC12 01: TBC14 10: TBC16 11: TBC18		TBC0 interrupt source clock selection 00: TBC11 01: TBC13 10: TBC15 11: TBC17	
TBCDR0	TBC Data Register0	000024H	TBCD7 (TBC12)	TBCD6 (TBC11)	TBCD5 (TBC10)	TBCD4 (TBC9)	TBCD3 (TBC8)	TBCD2 (TBC7)	TBCD1 (TBC6)	TBCD0 (TBC5)
			R							
			0							
TBCDR1	TBC Data Register1	000025H	TBCD15 (TBC20)	TBCD14 (TBC19)	TBCD13 (TBC18)	TBCD12 (TBC17)	TBCD11 (TBC16)	TBCD10 (TBC15)	TBCD9 (TBC14)	TBCD8 (TBC13)
			R							
			0							
TBCIF	TBC Interrupt Flag Register	000026H			TBC1F	TBC0F	—		—	—
			(set to 0)	(set to 0)	R/W			(set to 0)		
			0	0	0			0		
		"Prohibit RMW"			TBC1 interrupt request flag 0: (W)Clear 1: (R) Interrupt request	TBC0 interrupt request flag 0: (W) Clear 1: (R) Interrupt request				

Timer/Counter0, 1

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TRUN	Timer Start Control Register	000027H	—	—	T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
			R/W							
			0							
			0:STOP 1:RUN							
TCCR10	Timer/ Counter Control Register10	000028H	CLBC1	T1CL	T1CLK1	T1CLK0	CLBC0	T0CL	T0CLK1	T0CLK0
			R/W	R/W	R/W		R/W	R/W	R/W	
			0	0	0		0	0	0	
			TC1 timer/ counter match clear enable 0: Disable 1: Enable	TC1 timer/ counter clear 0: - 1: Clear (One-shot)	TC1 source clock selection 00: TI1 01: TBC2 10: TBC4 11: TBC6		TC0 timer/ counter match clear enable 0: Disable 1: Enable	TC0 timer/ counter clear 0: - 1: Clear (One-shot)	TC0 source clock selection 00: PCTLA (from CAPIN) 01: TI0 10: TBC6 11: TBC10	
TREG0	Timer/ Counter Register0	000029H	TC/TR07	TC/TR06	TC/TR05	TC/TR04	TC/TR03	TC/TR02	TC/TR01	TC/TR00
			R/W							
			0							
			W: Writing TREG0 register R: Reading TC0 counter							
TREG1L	Timer/ Counter Register1L	00002AH	TC/TR17	TC/TR16	TC/TR15	TC/TR14	TC/TR13	TC/TR12	TC/TR11	TC/TR10
			R/W							
			0							
			W: Writing TREG1L register R: Reading TC1L counter							
TREG1H	Timer/ Counter Register1H	00002BH	TC/TR1F	TC/TR1E	TC/TR1D	TC/TR1C	TC/TR1B	TC/TR1A	TC/TR19	TC/TR18
			R/W							
			0							
			W: Writing TREG1H register R: Reading TC1H counter							

Timer/Counter2,3

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TCCR32	Timer/ Counter Control Register32	00002CH	CLBC3	T3CL	T3CLK1	T3CLK0	CLBC2	T2CL	T2CLK1	T2CLK0
			R/W	R/W	R/W		R/W	R/W	R/W	
			0	0	0		0	0	0	
			TC3 timer/ counter match clear enable 0: Disable 1: Enable	TC3 timer/ counter clear 0: — 1: Clear (One-shot)	TC3 source clock selection 00: TI3 01: TBC2 10: TBC4 11: TBC6		TC2 timer/ counter match clear enable 0: Disable 1: Enable	TC2 timer/ counter clear 0: — 1: Clear (One-shot)	TC2 source clock selection 00: TI2 01: TBC2 10: TBC4 11: TBC6	
TREG2L	Timer/ Counter Register2L	00002EH	TC/TR27	TC/TR26	TC/TR25	TC/TR24	TC/TR23	TC/TR22	TC/TR21	TC/TR20
			R/W							
			0							
			W: Writing TREG2L register R: Reading TC2L counter							
TREG2H	Timer/ Counter Register2H	00002FH	TC/TR2F	TC/TR2E	TC/TR2D	TC/TR2C	TC/TR2B	TC/TR2A	TC/TR29	TC/TR28
			R/W							
			0							
			W: Writing TREG2H register R: Reading TC2H counter							
TREG3L	Timer/ Counter Register3L	000030H	TC/TR37	TC/TR36	TC/TR35	TC/TR34	TC/TR33	TC/TR32	TC/TR31	TC/TR30
			R/W							
			0							
			W: Writing TREG3L register R: Reading TC3L counter							
TREG3H	Timer/ Counter Register3H	000031H	TC/TR3F	TC/TR3E	TC/TR3D	TC/TR3C	TC/TR3B	TC/TR3A	TC/TR39	TC/TR38
			R/W							
			0							
			W: Writing TREG3H register R: Reading TC3H counter							

Timer/Counter4, 5

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TCCR54	Timer/ Counter Control Register54	00002DH	CLBC5	T5CL	T5CLK1	T5CLK0	CLBC4	T4CL	T4CLK1	T4CLK0
			R/W	R/W	R/W		R/W	R/W	R/W	
			0	0	0		0	0	0	
			TC5 timer/ counter match clear enable 0: Disable 1: Enable	TC5 timer/ counter clear 0: — 1: Clear (One-shot)	TC5 source clock selection 00: TI5 01: CSYNCA (from V-sepa) 10: TBC4 11: TBC6		TC4 timer/ counter match clear enable 0: Disable 1: Enable	TC4 timer/ counter clear 0: — 1: Clear (One-shot)	TC4 source clock selection 00: TI4 01: CFGTM (from CAPIN) 10: TBC4 11: TBC6	
TREG4L	Timer/ Counter Register4L	000032H	TC/TR47	TC/TR46	TC/TR45	TC/TR44	TC/TR43	TC/TR42	TC/TR41	TC/TR40
			R/W							
			0							
			W: Writing TREG4L register R: Reading TC4L counter							
TREG4H	Timer/ Counter Register4H	000033H	TC/TR4F	TC/TR4E	TC/TR4D	TC/TR4C	TC/TR4B	TC/TR4A	TC/TR49	TC/TR48
			R/W							
			0							
			W: Writing TREG4H register R: Reading TC4H counter							
TREG5L	Timer/ Counter Register5L	000034H	TC/TR57	TC/TR56	TC/TR55	TC/TR54	TC/TR53	TC/TR52	TC/TR51	TC/TR50
			R/W							
			0							
			W: Writing TREG5L register R: Reading TC5L counter							
TREG5H	Timer/ Counter Register4H	000035H	TC/TR5F	TC/TR5E	TC/TR5D	TC/TR5C	TC/TR5B	TC/TR5A	TC/TR59	TC/TR58
			R/W							
			0							
			W: Writing TREG5H register R: Reading TC5H counter							

Timing Pulse Generator 0 (TPG 0)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TPG0CR	TPG0 Control Register	000036H		—	FEMPIE	TPFUL0	TPEMP0	TPF02	TPF01	TPF00
			(set to 0)		W	R	R	R		
			0		0	0	1	0		
					INTTPG0 FIFO empty interrupt enable 0: — 1: Enable	TPG0 FIFO full flag 0: — 1: FIFO full	TPG0 FIFO empty 0: — 1: FIFO empty	TPG0 FIFO status flag (Stored data) 000: Empty or Full 001: 1 word 010: 2 word 011: 3 word 100: 4 word 101: 5 word 110: 6 word 111: 7 word		
TPG0L	TPG0 Lower Timing Data Register	000037H	TPG0D7	TPG0D6	TPG0D5	TPG0D4	TPG0D3	TPG0D2	TPG0D1	TPG0D0
			Write-only							
			—							
TPG0H	TPG0 Higher Timing Data Register	000038H	TPG0DF	TPG0DE	TPG0DD	TPG0DC	TPG0DB	TPG0DA	TPG0D9	TPG0D8
			Write-only							
			—							
TPG0DR	TPG0 Output Data Register	000039H	-	-	TPGD05	TPGD04	TPGD03	TPGD02	TPGD01	TPGD00
			Write-only							
			—							
CAPCR	Capture Control Register	000052H	CAP2T (CFG)	CAP1T (DFG)	CAPCL	VISFRS	VASFRS	TPRS0	CFGCL	CAFRS
			R	R	R/W					
			0	0	0					
			CAP2 trigger input status 0: No trigger input 1: Trigger input	CAP1 trigger input status 0: No trigger input 1: Trigger input	CAP1/ CAP2 status clear 0: — 1: Clear (One-shot)	VISS detect flag clear 0: — 1: Clear (One-shot)	VASS detect flag clear 0: — 1: Clear (One-shot)	TPG0 FIFO counter clear & INTTPG0 disable 0: — 1: Clear & Inhibit (One-shot)	CFG flag clear 0: — 1: Clear (One-shot)	Capture0 FIFO counter/ status clear 0: — 1: Clear (One-shot)

Timing Pulse Generator 1 (TPG1)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
TPG1L	TPG1 Lower Timing Data Register1L	00003AH	TPG1D7	TPG1D6	TPG1D5	TPG1D4	TPG1D3	TPG1D2	TPG1D1	TPG1D0
			W							
			0							
TPG1H	TPG1 Higher Timing Data Register1H	00003BH	TPG1DF	TPG1DE	TPG1DD	TPG1DC	TPG1DB	TPG1DA	TPG1D9	TPG1D8
			W							
			0							
TPG1DR	TPG1 Output Data Register	00003CH	—	—	—	—	TPGD13	TPGD12	TPGD11	TPGD10
							W			
							0			

14BIT PWM 0, 1, 2

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
PWMRUN	PWM Start Control Register	00003DH	—	—	—	—	—	PWM2--RUN	PWM1--RUN	PWM0--RUN
								R/W		
								0		
								0: Stop 1: Run		
PWM0DRL	PWM0 lower Data Register	00003EH	PWM0D7	PWM0D6	PWM0D5	PWM0D4	PWM0D3	PWM0D2	PWM0D1	PWM0D0
			Write-only							
			—							
PWM0DRH	PWM0 Higher Data Register	00003FH	—	—	PWM0DD	PWM0DC	PWM0DB	PWM0DA	PWM0D9	PWM0D8
					Write-only					
					—					
PWM1DRL	PWM1 Lower Data Register	000040H	PWM1D7	PWM1D6	PWM1D5	PWM1D4	PWM1D3	PWM1D2	PWM1D1	PWM1D0
			Write-only							
			—							
PWM1DRH	PWM1 Higher Data Register	000041H	—	—	PWM1DD	PWM1DC	PWM1DB	PWM1DA	PWM1D9	PWM1D8
					Write-only					
					—					
PWM2DRL	PWM2 Lower Data Register	000042H	PWM2D7	PWM2D6	PWM2D5	PWM2D4	PWM2D3	PWM2D2	PWM2D1	PWM2D0
			Write-only							
			—							
PWM2DRH	PWM2 Higher Data Register	000043H	—	—	PWM2DD	PWM2DC	PWM2DB	PWM2DA	PWM2D9	PWM2D8
					Write-only					
					—					

8BIT PWM 3 - 11

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
PWMDBR	8bit PWM Data Buffer Register	000044H	PWMDBR7	PWMDBR6	PWMDBR5	PWMDBR4	PWMDBR3	PWMDBR2	PWMDBR1	PWMDBR0
			Write-only							
			—							
			8-bit PWM data buffer register							
PWMCR	8bit PWM Control Register	000045H	PWMEOT	—	—	—	PWMDLS3	PWMDLS2	PWMDLS1	PWMDLS0
			R				WRITE ONLY			
			0				0			
			End flag of PWM data transfer 0: End of transfer 1: Under transfer				PWM channel select and data transfer The PWMDBR data is stored to the selected PWM channel after setting this channel code. This code must be set when <PWMEOT> = 0 0000: PWM3 0100: PWM7 1000: PWM11 0001: PWM4 0101: PWM8 0010: PWM5 0110: PWM9 0011: PWM6 0111: PWM10			

Capture Control

SYMBOL	Name	Address	7	6	5	4	3	2	1	,O
CAPINCR0	Capture Input Control Register0	000046H	CAP2E (CFG)	CAP1E (DFG)	CAP05E (DPG)	CAP04E (EXT)	CAP03E (CTL)	CAP02E (VS)	CAP01E (RMTD)	CAP00E (RMTU)
			R/W							
			0							
			Capture 0, 1/2 input source enable/disable control 0: Disable 1: Enable							
CAPINCR1	Capture Input Control Register1	000047H	PCTLCK1	PCTLCK0	PCPR5	PCPR4	PCPR3	PCPR2	PCPR1	PCPR0
			R/W		R/W					
			0		0					
			CTL duty measure clock select (VIVACR2<PCTLCKS> = 0) 00: TBC3 01: TBC5 10: TBC7 11: TBC10		6-bit programmable divider control Dividing rate selection for CTLIN (from P80) and CFGIN (from P85). 000000: 1/1 000001:1 /2 --- 111111: 1/64					
CAPINCR2	Capture Input Control Register2	000048H	—	RMTST	RMTP0	RMTBP	CFGMCP	DPCP2	DPCP1	DPCP0
				R/W	R/W	R/W	R	R/W		
				0	0	0	0	0		
				RMTIN start/stop control 0: Stop & counter clear 1: Start	RMTIN input polarity selection 0: Positive (+) 1: Negative (-)	Noise canceller/ correction circuit control 0: Active 1: Bypass	CFG status flag 0: Normal 1: Error	3-bit programmable divider control Dividing rate selection for DFGIN (from P81) 000: 1/1 001: 1/2 -- 111: 1/8		
CAPINCR3	Capture Input Control Register3	000049H		CTLPO	CFGPO	DPGEG	DFGEG	CFGWPR	CFGEG	CTLEG
				R/W	R/W	R/W	R/W	R/W	R/W	R/W
				0	0	0	0	0	0	0
				CTL input polarity selection 0: Positive (+) 1: Negative (-)	CFG input polarity selection 0: Positive (+) 1: Negative (-)	Edge selection for DPG input 0: Rising 1: Falling	Edge selection for DFG input 0: Rising 1: Falling	Dividing rate selection for CFG input 0: 1/1 1 :1/2	Edge selection for CFG input 0: Both rising and falling 1: Rising	Edge selection for CTLIN input 0:Rising 1: Both rising and falling

Capture 0

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
VIVACR2	VISS/ VASS Control Register2	000055H	PCTLPO	PCTLCKS	MSK1	MSK0	VISS3	VISS2	VISS1	VISS0
			R/W	R/W	R/W	R/W	R/W			
			0	0	0	0	0			
			CTL signal polarity selection 0: Positive (+)(when duty is less than 50%, VIVACR1 <CTLDTY> is set to 1) 1: Negative (-)(when duty is less than 50%, VIVACR1 <CTLDTY> is set to 0)	CTL duty measuring clock selection 0: Selected by CAPINCR1 <PCTLCK1, 2> 1: Automatic selection	Dividing control for CTLIN input 0: Divide 1: Bypass	masking control for CFG input 0: Mask 1: Bypass	VISS compare data These 4-bit data are compared with higher 4bit of the 6-bit CTL counter			
CAPFST	Capture0 FIFO Status Register	00004AH	CAPF7 (FIFO8)	CAPF6 (FIFO7)	CAPF5 (FIFO6)	CAPF4 (FIFO5)	CAPF3 (FIFO4)	CAPF2 (FIFO3)	CAPF1 (FIFO2)	CAPF0 (FIFO1)
			Read-only							
			0							
			Capture0 FIFO status 0: Empty, 1: Capture data stored							
CAP0L	Capture0 Lower Data Register	00004BH	CAP0D7	CAP0D6	CAP0D5	CAP0D4	CAP0D3	CAP0D2	CAP0D1	CAP0D0
			Read-only							
			—							
CAP0M	Capture0 Middle Data Register	00004CH	CAP0D15	CAP0D14	CAP0D13	CAP0D12	CAP0D11	CAP0D10	CAP0D9	CAP0D8
			Read-only							
			—							
CAP0H	Capture0 Higher Data Register	00004DH	CAP0T5	CAP0T4	CAP0T3	CAP0T2	CAP0T1	CAP0T0	CAP0D17	CAP0D16
			Read-only						Read-only	
			—						—	
			Trigger input status 0: No trigger input 1: Trigger input						Capture0 higher data	

Capture 1, 2

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
CAP1L	Capture1 Lower Data Register	00004EH	CAP1D7	CAP1D6	CAP1D5	CAP1D4	CAP1D3	CAP1D2	CAP1D1	CAP1D0
			Read-only							
			—							
CAP1H	Capture1 Higher Data Register	00004FH	CAP1D15	CAP1D14	CAP1D13	CAP1D12	CAP1D11	CAP1D10	CAP1D9	CAP1D8
			Read-only							
			—							
CAP2L	Capture2 Lower Data Register	000050H	CAP2D7	CAP2D6	CAP2D5	CAP2D4	CAP2D3	CAP2D2	CAP2D1	CAP2D0
			Read-only							
			—							
CAP2H	Capture2 Higher Data Register	000051H	CAP2D15	CAP2D14	CAP2D13	CAP2D12	CAP2D11	CAP2D10	CAP2D9	CAP2D8
			Read-only							
			—							

Remote Control Input (RMTIN)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
RMTCR	RMTIN Control Register	000053H	RMTD7	RMTD6	RMTD5	RMTD4	RMTD3	RMTD2	RMTD1	RMTD0
			R/W				R/W			
			0				0			
			H level Noise cancel width 4-bit data (TBC8: 25.6 µs)				L level Noise cancel width 4-bit data (TBC8: 25.6 µs)			

VISS/VASS

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
VIVACR1	VISS/ VASS Control Register1	000054H		—	—	—	—	CTLDTY	VISSFL	VASSFL
			(set to 0)					R	R	R
			0					0	0	0
								CTL signal duty detector monitor flag 0: CTL duty greater than 50% (VIVACR2 <PCTLPO> = 0) 1: CTL duty less than 50% (VIVACR2 <PCTLPO> = 0)	VISS detect flag 0: — 1: VISS detected	VASS detect flag 0: — 1: VASS detected
VASSDR	VASS Data Register	000056H	VASS7	VASS6	VASS5	VASS4	VASS3	VASS2	VASS1	VASS0
			VASS lower 8-bit data							
			VASS15	VASS14	VASS13	VASS12	VASS11	VASS10	VASS9	VASS8
			VASS higher 8-bit data							
			Read-only							
			—							
			Twice reading from VASSDR The First is lower 8-bit, the second is higher 8-bit							

CSYNC, PV, CR/HA

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
CSYNCR	CSYNC Control Register	000057H	COMPS	-	-	CSYNCPO	SEPMOD	CSYNBP	HSEN	MASK
			R/W			R/W	R/W	R/W	R/W	R/W
			0			0	0	0	0	0
			COMPIN input control 0: Disable (TPG00/ TPG05 Output) 1: Enable			CSYNC input signal polarity selection 0: Positive 1: Negative	7-bit counter up/down count selection (duty) 0: up/down = 1/1 1: up/down = 1/2	CSYNC bypass control 0: V-sep. 1: Bypass	Pseudo-Hsync generation to PV/PH 0: Non-synchronized with Csync input 1: synchronize with Csync input	Vsync mask control 0: — 1: Release masking (One-shot)
PVCR	PV Control Register	000058H	—	—	—	—	PHMIX	PVSEL2	PVSEL1	PVSEL0
							R/W	R/W		
							0	0		
							PH mixing 0: OFF 1: ON	PV/PH output format control		

8-bit A/D Converter

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
ADMOD	A/D Converter Control Register	00005AH		EOCF	ADBF	ADS	ADCH3	ADCH2	ADCH1	ADCH0
			(set to 1)	R	R	R/W	R/W			
			1	0	0	0	0			
				A/D conversion flag 0: During or before converting 1: Finish conversion	A/D conversion busy flag 0: free (stop) 1: busy	A/D conversion start /status 0: — 1: A/D conversion start	Analog channel selection 0000: AIN0 1001: AIN9 0001: AIN1 1010: AIN10 0010: AIN2 1011: AIN11 0011: AIN3 1100: AIN12 0100: AIN4 1101: AIN13 0101: AIN5 1110: AIN14 0110: AIN6 1111: AIN15 0111: AIN7 1000: AIN8			
ADREG	A/D Converted Data Register	00005BH	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
			Read-only							
			—							

SIO 0

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
SC0MOD	Serial I/O Channel 0 Mode Register	000060H	FF0SI	S0RES	S0MOD1	S0MOD0	SIFT0	CLK0SI	SCK0S	SIO0E
			R	R/W	R/W		R/W	R/W	R/W	R/W
			1	0	0		0	0	0	0
			Transfer monitor flag 0: Under transferring 1: Stop	Serial transfer terminate 0: — 1: Terminate	Transfer mode select 00: Transmit mode 01: Receive mode 10: — 11: Transmit/receive mode		Shift edge select 0: Leading (Falling) 1: Trailing (Rising)	Internal clock rate select 0: TBC3 (0.8 µs) 1: TBC7 (6.4 µs)	Serial clock source select 0: Internal clock 1: External clock	Serial transfer enable/ disable 0: Disable 1: Enable
SC0BUF	Serial I/O Channel 0 Buffer Register	000061H	TRB07	TRB06	TRB05	TRB04	TRB03	TRB02	TRB01	TRB00
			R/W							
			—							

SIO 1

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
SC1MOD	Serial I/O Channel 1 Mode Register	000062H	FF1SI	S1RES	S1MOD1	S1MOD0	SIFT1	CLK1SI	SCK1S	SIO1E
			R	R/W	R/W		R/W	R/W	R/W	R/W
			1	0	0		0	0	0	0
			Transfer monitor flag 0: Under transferring 1: Stop	Transfer terminate 0: — 1: Terminate	Transfer mode select 00: Transmit mode 01: Receive mode 10: — 11: Transmit/receive mode		Shift edge select 0: Leading (Falling) 1: Trailing (Rising)	Internal clock rate select 0: TBC3 (0.8 µs) 1: TBC7 (6.4 µs)	Serial clock source select 0: Internal clock 1: External clock	Serial transfer enable/ disable 0: Disable 1: Enable
SC1BUF	Serial I/O Channel 1 Buffer Register	000063H	TRB17	TRB16	TRB15	TRB14	TRB13	TRB12	TRB11	TRB10
			R/W							
			—							

SIO 2 (UART)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
SC2MOD	Serial I/O Mode Control Register2	000064H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W	R/W	R/W	R/W	R/W		R/W	
			0	0	0	0	0		0	
			Transmit data bit 8	Hand shake control 0: CTS disable 1: CTS enable	Receiver function 0: Receive disable 1: Receive enable	Wake up function 0: Disable 1: Enable	Serial transfer mode 00: (Reserved) 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transfer clock (UART) 00: Timer 0 trigger 01: Baud rate generator 10: Internal clock: fc/2 11: (Reserved)	
SC2CR	Serial Control Register2	000065H	RB8	EVEN	PE	OERR	PERR	FERR	—	—
			R	R/W	R/W	R(cleared to Zero when read)				
				0	0	0	0	0		
			Received data bit 8	Parity check mode 0: Odd 1: Even	Parity check 0: Disable 1: Enable	Overrun error flag 0: — 1: Error	Parity error flag 0: — 1:Error	Framing error flag 0: — 1: Error		
BR2CR	Baud Rate Generator Control Register	000066H		—	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			(set to 0)		R/W		R/W			
			0		0		0			
					Selecting the input clock of baud rate generator 00: TBC1 (0.2 μs) 01: TBC2 (0.4 μs) 10: TBC3 (0.8 μs) 11; TBC4 (1.6 μs) @fc = 20 MHz		Divider for baud rate generator 0000: 1/16 0001: 1/1 0010 to 1111: 1/2 to 1/15			
SC2BUF	Serial Transmit/ Receive Buffer2	000067H	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
			W: (Transmit)							
			RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
		Prohibit RMW	R: (Receive)							

I²CBUS (1)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
I2CCR1	I²CBUS Control Register1	000081H Prohibit RMW	BC2	BC1	BC0	ACK	CHS	-	-	-
			WRITE			R/W	R/W			
			0	0	0	0	0			
			Number of transferred bits ACK = 0 | ACK = 1 clocks bits | clock bits 000: 8 8 | 9 8 001: 1 1 | 2 1 010: 2 2 | 3 2 011: 3 3 | 4 3 100: 4 4 | 5 4 101: 5 5 | 6 5 110: 6 6 | 7 6 111: 7 7 | 8 7			Clock pulse generation for acknow-ledge bit 0: Don't 1: Generate	Channel selection 0: Channel 0 (SCL0, SDA0) 1: Channel1 (SCL1, SDA1)			
I2CDBR	I²CBUS Data Buffer Register	000082H	I2DBR7	I2DBR6	I2DBR5	I2DBR4	I2DBR3	I2DBR2	I2DBR1	I2DBR0
			R/W							
			Note: Start from the MSB for transmit data							
I2CAR	I²CBUS Address Register	000083H	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			Write-only							W
			0	0	0	0	0	0	0	0
			TMP93C071 slave address selection							Addressing mode specifi-cation 0: Slave address 1: (free data format)

I²CBUS (2)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
I2CCR2	I²CBUS Control Register2	000084H (Write) Prohibit RMW	MST	TRX	BB	PIN	SBIM1	SBIM0	(set to "0")	(set to "0")
			0	0	0	1	0	0	*	*
			Master/ slave selection 0: Slave 1: Master	Transmitter/ receiver selection 0: Receiver 1: Transmit-ter	Start/stop condition generation 0: Stop condition 1: Start condition	Cancel interrupt service request 0: — 1: Cancel interrupt service request	Serial bus interface operation mode selection 00: Port mode (I²CBUS output disable) 01: (Reserved) 10: I²CBUS mode 11: (Reserved)			
I2CSR	I²CBUS Status Register	000084H (Read)	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
			0	0	0	1	0	0	*	*
			Status monitor 0: Slave 1: Master	Status monitor 0: Receiver 1: Transmit-ter	Status monitor 0: Bus free 1: Bus busy	Interrupt service request status monitor 0: Requested 1: Released	Arbitration lost detection monitor 0: — 1: Arbitra-tion lost detected	Slave address match detection monitor 0: — 1: Slave address match or GENERAL CALL detected	GENERAL CALL detection monitor 0: — 1: GENERAL CALL detected	Last received bit monitor 0: Last received bit 0 1: Last received bit 1

I²CBUS (3)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
I2CCR3	I²CBUS Control Register3	000085H	DMAEN	DV4	DV3	DV2	DV1	DV0	—	SWRST
			R/W	W						R/W
			0	0	0	0	0	*	*	0
			micro DMA transfer mode is enabled 0: Disable 1: Enable	Setting N-divider for generating transfer clock source. 00000: (Reserved) 10000: (N=16); 74.6 00001: (Reserved) 10001: (N=17); 70.4 00010: (N = 2); 454.5 10010: (N = 18); 66.7 00011: (N = 3); 333.3 10011: (N = 19); 63.3 00100: (N = 4); 263.2 10100: (N = 20); 60.2 00101: (N = 5); 217.4 10101: (N = 21); 57.5 00110: (N = 6); 185.2 10110: (N = 22); 54.9 00111: (N = 7); 161.3 10111: (N = 23); 52.6 01000: (N = 8); 142.9 11000: (N = 24); 50.5 01001: (N = 9); 128.2 11001: (N = 25); 48.5 01010: (N = 10); 116.3 11010: (N = 26); 46.7 01011: (N = 11); 106.4 11011: (N = 27); 45.0 01100: (N = 12); 98.0 11100: (N = 28); 43.5 01101: (N = 13); 90.9 11101: (N = 29); 42.0 01110 :(N = 14); 84.7 11110: (N = 30); 40.7 01111: (N = 15); 79.4 11111: (N = 31); 39.4 (kHz) @20 MHz			I2CBUS Software Reset 0: — (Initial state) 1: Initialize I²CBUS block (After initialize I²CBUS, SWRST is automatically cleared to 0.)
		Prohibit RMW								

I²CBUS (4)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
I2CCR4	I²CBUS Control Register4	000086H	RSTR	T/$\overline{R}$	DMAEND	RCONT	C3	C2	C1	C0
			W	R/W	R	R/W	R/W			
				0	0	0	0	0	0	0
			Continuous transfer command at micro DMA transfer mode. This command is available while DMAEND = 1. 0: — 1: Restart micro DMA transfer for (C3-0) set at a time. (note) T/$\overline{R}$ bit must be set a previous value.	micro DMA transfer Transmit/ Receive mode selection. 0: Receive 1: Transmit	micro DMA transfer end flag 0:— 1: Transfer end (set condition) Transmit: I²CBUS has transmitted the last data. Receive: micro DMA has read a data after I²CBUS has received the last data. (reset condition) Reset, Start/Stop command have been executed. RSTR command has been executed.	Continuous receive mode set at micro DMA transfer mode (receive). If the receive mode is set to "Receiving over(C3-0) byte", set this bit at a time. 0: Receive end mode 1: Receive continuous mode	Setting the number of transfer byte 0000: 16 byte 1000: 8 byte 0001: 1 byte 1001: 9 byte 0010: 2 byte 1010: 10 byte 0011: 3 byte 1011: 11 byte 0100: 4 byte 1100: 12 byte 0101: 5 byte 1101: 13 byte 0110: 6 byte 1110: 14 byte 0111: 7 byte 1111: 15 byte (slave address + R/$\overline{W}$) is not contained. The value of (C3-0) is decreased by each micro DMA transfer (I2CDBR access). This value must be set after (slave address + R/$\overline{W}$) has been set to I2CDBR.			
		Prohibit RMW								

Interrupt priority setting register (1/2)

(Prohibit read-modify-write)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
INT0CP1	INT0/	0070H	INTCAP1				INT0			
	CAP1		ICAP1C	ICAP1M2	ICAP1M1	ICAP1M0	I0C	I0M2	I0M1	I0M0
	Interrupt	"Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW"	0	0	0	0	0	0	0	0
INTCP0TG0	CAP0/	0071H	INTTPG0				INTCAP0			
	TPG0		ITPG0C	ITPG0M2	ITPG0M1	ITPG0M0	ICAP0C	ICAP0M2	ICAP0M1	ICAP0M0
	Interrupt	"Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW"	0	0	0	0	0	0	0	0
INTTG1DMA	TPG1/	0072H	INTI2DMA				INTTPG1			
	I2DMA		II2DMAC	II2DMAM 2	II2DMAM 1	II2DMAM 0	ITPG1C	ITPG1M2	ITPG1M1	ITPG1M0
	Interrupt	"Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW"	0	0	0	0	0	0	0	0
INTI2CTBC	I2CBUS/	0073H	INTTBC				INTI2CB			
	TBC		ITBCC	ITBCM2	ITBCM1	ITBCM0	II2CC	II2CM2	II2CM1	II2CM0
	Interrupt	"Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW"	0	0	0	0	0	0	0	0
INTSI0SI1	SIO0/	0074H	INTSIO1				INTSIO0			
	SIO1		ISIO1C	ISIO1M2	ISIO1M1	ISIO1M0	ISIO0C	ISIO0M2	ISIO0M1	ISIO0M0
	Interrupt	"Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW"	0	0	0	0	0	0	0	0
INTRXTX	RX/	0075H	INTTX				INTRX			
	TX		ITXC	ITXM2	ITXM1	ITXM0	IRXC	IRXM2	IRXM1	IRXM0
	Interrupt	"Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW"	0	0	0	0	0	0	0	0
INTVA1	VA/	0076H	INT1				INTVA			
	INT1		I1C	I1M2	I1M1	I1M0	IVAC	IVAM2	IVAM1	IVAM0
	Interrupt	"Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW"	0	0	0	0	0	0	0	0
INT23	INT2/	0077H	INT3				INT2			
	INT3		I3C	I3M2	I3M1	I3M0	I2C	I2M2	I2M1	I2M0
	Interrupt	"Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW"	0	0	0	0	0	0	0	0
INT4T0	INT4/	0078H	INTT0				INT4			
	Timer0		IT0C	IT0M2	IT0M1	IT0M0	I4C	I4M2	I4M1	I4M0
	Interrupt	"Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW"	0	0	0	0	0	0	0	0

Interrupt priority setting register (2/2)

(Prohibit read-modify-write)

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
INTT1T2	Timer1/	0079H	INTT2				INTT1			
	Timer2		IT2C	IT2M2	IT2M1	IT2M0	IT1C	IT1M2	IT1M1	IT1M0
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW	0	0	0	0	0	0	0	0
INTT3T4	Timer3/	007AH	INTT4				INTT3			
	Timer4		IT4C	IT4M2	IT4M1	IT4M0	IT3C	IT3M2	IT3M1	IT3M0
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW	0	0	0	0	0	0	0	0
INTT5AD	Timer5/	007BH	INTAD				INTT5			
	AD		IADCC	IADM2	IADM1	IADM0	IADC	IADM2	IADM1	IADM0
	Interrupt	Prohibit	R/W	W	W	W	R/W	W	W	W
	Setting	RMW	0	0	0	0	0	0	0	0
INTRTCR		0080H	(Always set to "0")				INTRTC			
	RTC		(set to 0)	(set to 0)	(set to 0)	(set to 0)	IRTCMA	IRTCM2	IRTCM1	IRTCM0
	Interrupt	Prohibit	(R/W)	(R/W)	(W)	(W)	R/W	W	W	W
	Setting	RMW	(0)	(0)	(0)	(0)	0	0	0	0

IxxM2	IxxM1	IxxM0	Function(Write)
0	0	0	Prohibit interrupt request.
0	0	1	Set interrupt request level to 1.
0	1	0	Set interrupt request level to 2.
0	1	1	Set interrupt request level to 3.
1	0	0	Set interrupt request level to 4.
1	0	1	Set interrupt request level to 5.
1	1	0	Set interrupt request level to 6.
1	1	1	Prohibit interrupt request.

IxxC	Function (Read)	Function (Write)
0	Indicates no interrupt request.	Clears interrupt request flag.
1	Indicates interrupt request.	don't care

Note 1: Read-modify-write is prohibited.

Note 2: Note about clearing interrupt request flag.
The interrupt request flag of INTCAP1, INTCAP0, INTSIO0, INTSIO1, INTRX and INTAD are not cleared by writing 0 to IxxC of they are level interrupts.
They can be cleared only by resetting, reading captured data / ADREG /SC2BUF or reading/writing SC0BUF / SC1BUF.

Note 3: Note about clearing interrupt request flag
When the INTTPG0 is used for a FIFO empty interrupt (a level signal), the interrupt controller also leaves a request flag (Flip/Flop) after clearing FIFO empty by setting next TPG0 data in an interrupt routin.
Therefore, in this case, the INTTPG0 request flag has to be cleared before executing RETI instruction.

Micro DMA Start Vector

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
DMA0V	micro DMA0 Start Vector	00007CH			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					W (Read-modify-write is not possible)					
					0					
					Set micro DMA start vector					
DMA1V	micro DMA1 Start Vector	00007DH			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					W (Read-modify-write is not possible)					
					0					
					Set micro DMA start vector					
DMA2V	micro DMA2 Start Vector	00007EH			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					W (Read-modify-write is not possible)					
					0					
					Set micro DMA start vector					
DMA3V	micro DMA3 Start Vector	00007FH			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					W (Read-modify-write is not possible)					
					0					
					Set micro DMA start vector					

Port2

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
P2	PORT2 Register	000006H	P27	P26	P25	P24	-	-	-	-
			R/W							
			1				1	1	1	1
P2FC	PORT2 Function Register	000009H	P27F	P26F	P25F	P24F	(set to "1")	(set to "1")	(set to "1")	(set to "1")
			W							
			1							
			0: PORT 1: ADDRESS BUS (A23 to A20)							

Port 4

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
P4	Port4 Register	00000AH	P47	P46	P45	P44	P43	P42	P41	P40
			R							
			Input mode							

Port 5

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
P5	Port5 Register	00000BH	P57	P56	P55	P54	P53	P52	P51	P50
			R/W							
			1							
P5CR	Port5 Control Register	00000CH	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
			W							
			0							
			0:IN 1:OUT							
IIMC0	Interrupt Input Mode Control Register0	00005EH	—	—	—	I4IE (INT4)	I3IE (INT3)	I2IE (INT2)	I1IE (INT1)	I0IE (INT0)
						W				
						0				
						External interrupt enable 0: Disable 1: Enable				
IIMC1	Interrupt Input Mode Control Register1	00005FH	I4EG	I3EG	I2EG	I1EG	I0EG	—	INTTPG0E	INTTPG0S
			W						R/W	R/W
			0						0	0
			INT4 edge selection 0: Rising 1: Falling	INT3 edge selection 0: Rising 1: Falling	INT2 edge selection 0: Rising 1: Falling	INT1 edge/level selection 0: Rising edge 1: Level	INT0 edge/level selection 0: Rising edge 1: Level		INTTPG0 interrupt TPG03 edge selection 0: Rising 1: Falling	INTTPG0 interrupt selection 0: FIFO empty interrupt 1: FIFO empty/ TPG03 interrupt

Port 6

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
P6	Port6 Register	00000EH	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			1							
P6CR	Port6 Mode Register	000010H	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			0							
			0:IN 1:OUT							
P6FC	Port6 Function Register	000012H	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
			W							
			0							
			0: Port67 1: PWM11	0: Port66 1: PWM10	0: Port65 1: PWM9	0: Port64 1: PWM8	0: Port63 1: PWM7	B2CS <CS2EN> = 0 0: Port62 1: PWM6	B1CS <CS1EN> = 0 0: Port61 1: PWM5	B0CS <CS0EN> = 0 0: Port60 1: PWM4
ODCR0 (P6ODCR)	Open Drain Control Register0	000014H	POD67	POD66	POD65	POD64	POD63	POD62	POD61	POD60
			W							
			0							
			0: Push-pull 1: Open drain							

Port 7

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
P7	Port7 Register	00000FH	P77	P76	P75	P74	P73	P72	P71	P70
			R/W							
			1							
P7CR	Port7 Control Register	000011H	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
			W							
			0							
			0:IN 1:OUT							
P7FC	Port6 Function Register	000013H	P77F	—	P75F	P74F	P73F	—	—	P70F
			W		W					W
			0		0					0
			0: Port77 1: SCK0		0: Port75 1: SO0	0: Port74 1: SCL0	0: Port73 1: SDA0			0: Port70 1: TXD
ODCR1	Open Drain Control Register1	000015H	POD77	—	POD75	POD74	POD73	—	—	POD70
			W		W					W
			0		0					0
			0: Push-pull 1: Open-drain		0: Push-pull 1: Open drain					0: Push-pull 1: Open-drain
PRDSEL	P7 Register Read Value Selection	00000DH	PR77	—	PR75	PR74	PR73	—	—	PR70
			W		W					W
			0		0					0
			0: Pin value 1: Output value		0: Pin value 1: Output value					0: Pin value 1: Output value

Port 8

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
P8	P8 Register	000016H	P87	P86	P85	P84	P83	P82	P81	P80
			R/W							
			1							
P8CR	P8 Control Register	000018H	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
			W							
			0							
			0: IN 1: OUT							

Port 9

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
P9	P9 Register	000017H	P97	P96	P95	P94	P93	P92	P91	P90
			R/W							
			1							
P9CR	P9 Control Register	000019H	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
			W							
			0							
			0: IN 1: OUT							
P9FC	P9 Function Register	00001AH	P97F	P96F	P95F	P94F	P93F	P92F	P91F	P90F
			W							
			0							
			0: Port97 1: TPG11	0: Port96 1: TPG10	0: Port95 1: TPG13	0: Port94 1: TPG04	0: Port93 1: TPG03	0: Port92 1: TPG02	0: Port91 1: TPG01	0 :Port90 1: TPG12
ODCR2	Open Drain Control Register2	00001BH	POD97	POD96	POD95	POD94	POD93	POD92	POD91	POD90
			W							
			0							
			0: Push-pull 1: Open-drain							

Port A

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
PA	PortA Register	00001CH	—	—	PA5	PA4	PA3	PA2	PA1	PA0
					R/W					
					1					
PACR	PortA Control Register	00001EH	—	—	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
					W					
					0					
					0: IN 1: OUT					
ODCR3	Open Drain Control Register3	000021H	—	—	—	PWMOD1 (PWM1)	PWMOD0 (PWM0)	PODA5 (PortA5)	PODA4 (PortA4)	PODA3 (PortA3)
						W		W		
						0		0		
						PWM0, 1 Open drain control 0: Push-pull 1: Open drain		PortA Open drain control 0: Push-pull 1: Open drain		
PAFC	PortA Function Register	00006BH	—	PA5F1	PA5F0	PA4F	PA3F	PA2F	PA1F	PA0F
				W	W	W	W	W	W	W
				0	0	0	0	0	0	0
				0: PORTA5 /PWM3 <PA5F0> 1: $\overline{HWR}$	0: PORTA5 1: PWM3	0: PORTA4 1: $\overline{WR}$	0: PORTA3 1: PWM2	0: PORTA2 1: CR (TPG00)	0: PORTA1 1: HA (TPG05)	0: PORTA0 1: PV/PH

Port B

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
PB	PortB Register	00001DH	—	—	PB5	PB4	PB3	PB2	PB1	PB0
					R/W				R/W	
					1				1	
									(SYSCR0<XTEN> = 0)	
PBCR	PortB Control Register	00001FH	—	—	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
					W				W	
					0				0	
					0: IN 1: OUT				0: IN 1: OUT	
PBFC	PortB Function Register	000020H	—	—	PB5F	PB4F	PB3F	PB2F	—	—
					W	W	W	W		
					0	0	0	0		
					0: PORTB5 1: SCL1	0: PortB4 1: SDA1	0: PORTB3 1: SCK1	0: PORTB2 1: SO1/SI1		
ODCR4	Open Drain Control Register4	000022H	PRB5 (PortB5)	PRB4 (PortB4)	PRB3 (PortB3)	PRB2 (PortB2)	PODB5 (Portb5)	PODB4 (PortB4)	PODB3 (PortB3)	PODB2 (PortB2)
			W				W			
			0				0			
			PortB Read Value selection 0: Pin value 1: Output value				PortB Open drain control 0: Push-pull 1: Open drain			

Port C

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
PC	PC Register	000087H	—	—	—	PC4	PC3	PC2	PC1	PC0
						R/W				
						1				
PCCR	PC Control Register	000088H	—	—	—	PC4C	PC3C	PC2C	PC1C	PC0C
						W				
						0				
						0: IN 1: OUT				

Chip select/wait control register

SYMBOL	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block0 CS/WAIT Control Register	000068H	—	CS0EN	—	B0BUS	B0W1	B0W0	B0C1	B0C0
				W		W	W		W	
				0		0	0		0	
				0: PORT60 /PWM4 (P6FC <P60F>) 1: CS0		BUS Width Control 0: 16-bit bus 1: 8-bit bus	Wait Control 00: 0 Wait 01: 1 Wait 10: 1 Wait 11: 2 Wait		00: 2090H to 3FFFFFH 01: 400000H to 7FFFFFH 10: 800000H to BFFFFFH 11: C00000H to FFFFFFH	
B1CS	Block1 CS/WAIT Control Register	000069H	—	CS1EN	—	B1BUS	B1W1	B1W0	B1C1	B1C0
				W		W	W		W	
				0		0	0		0	
				0: PORT61 /PWM5 (P6FC <P61F>) 1: $\overline{CS1}$		BUS Width Control 0:16-bit bus 1: 8-bit bus	Wait Control 00: 0 Wait 01: 1 Wait 10: 1 Wait 11: 2 Wait		00: 2090H to 3FFFFFH 01: 400000H to 7FFFFFH 10: 800000H to BFFFFFH 11: C00000H to FFFFFFH	
B2CS	Block2 CS/WAIT Control Register	00006AH	—	CS2EN	—	B2BUS	B2W1	B2W0	B2C1	B2C0
				W		W	W		W	
				0		0	0		0	
				0 :PORT62 /PWM6 (P6FC <P62F>) 1: $\overline{CS2}$		BUS Width Control 0: 16bit bus 1: 8bit bus	Wait Control 00: 0 Wait 01: 1 Wait 10: 1 Wait 11: 2 Wait		00: 2090H to 3FFFFFH 01: 400000H to 7FFFFFH 10: 800000H to BFFFFFH 11: C00000H to FFFFFFH	

6. Port Section Equivalent Circuit Diagram

· Reading the circuit diagram

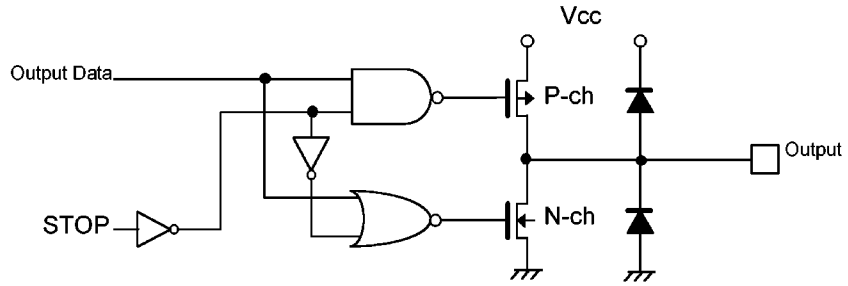
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCxx] series.

The dedicated signal is described below.

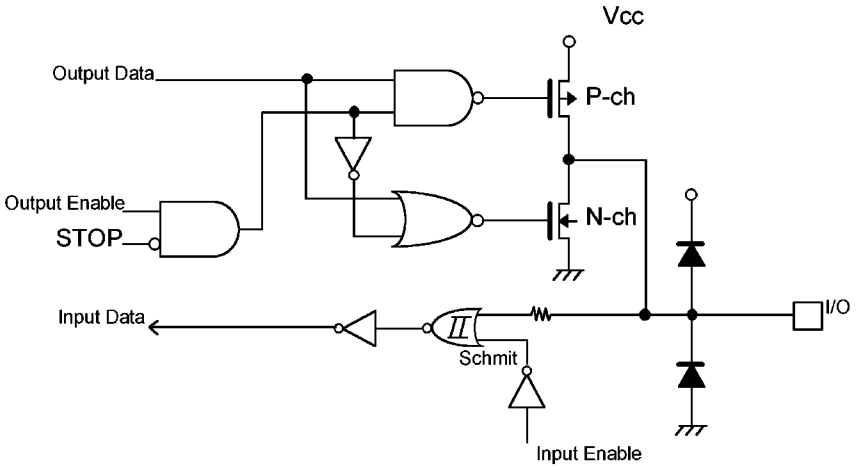
STOP: This signal becomes active 1 when the halt mode setting register is set to the STOP mode (WDMOD<HALT1, 0> = 0, 1) and the CPU executes the HALT instruction. When the drive enable bit WDMOD<DRVE> is set to 1, however, STOP remains at 0.

· The input protection resistans ranges from several tens ohms to several hundreds of ohms.

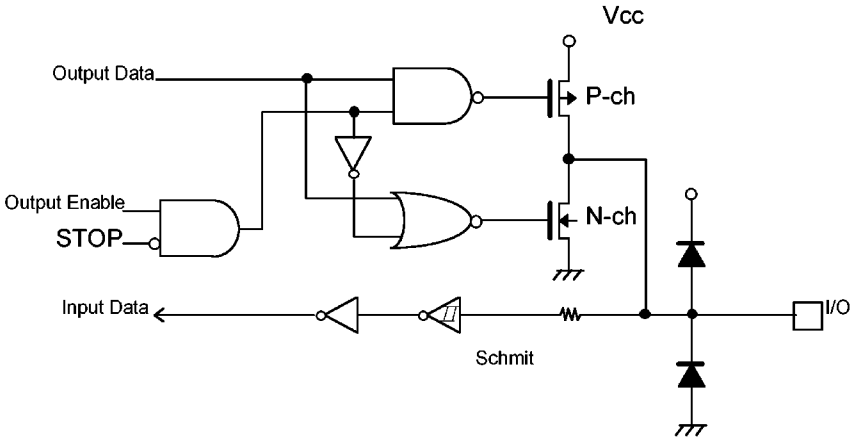
P24/A20 to P27/A23, A0 to A19, $\overline{RD}$



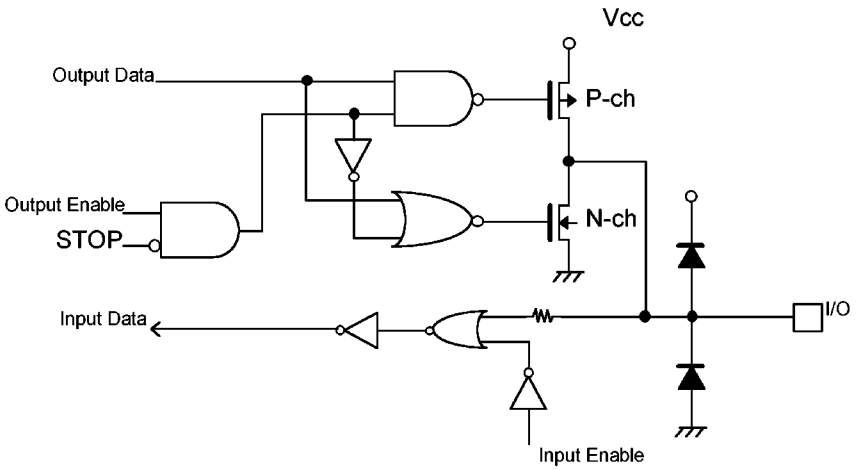
P50/INT4/TI3 to P52/INT2/TI1, P71/RXD, P72/ $\overline{CTS}$, P76/SI0, P80/CTLIN to P87/COMPIN



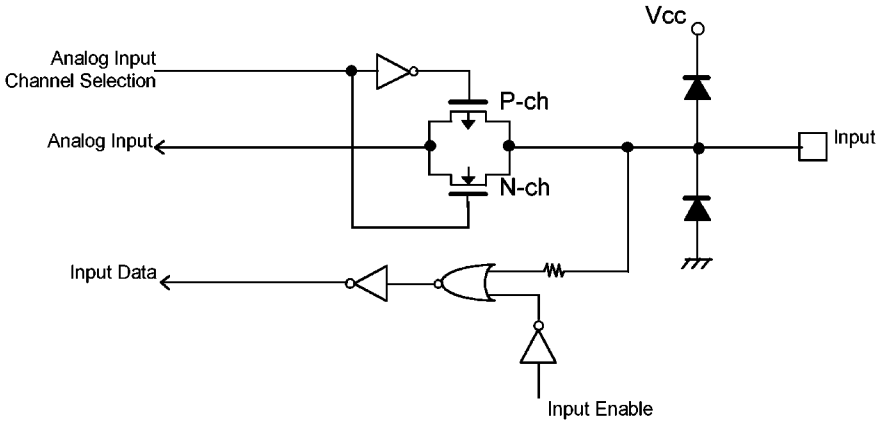
P53/INT1 to P54/INT0



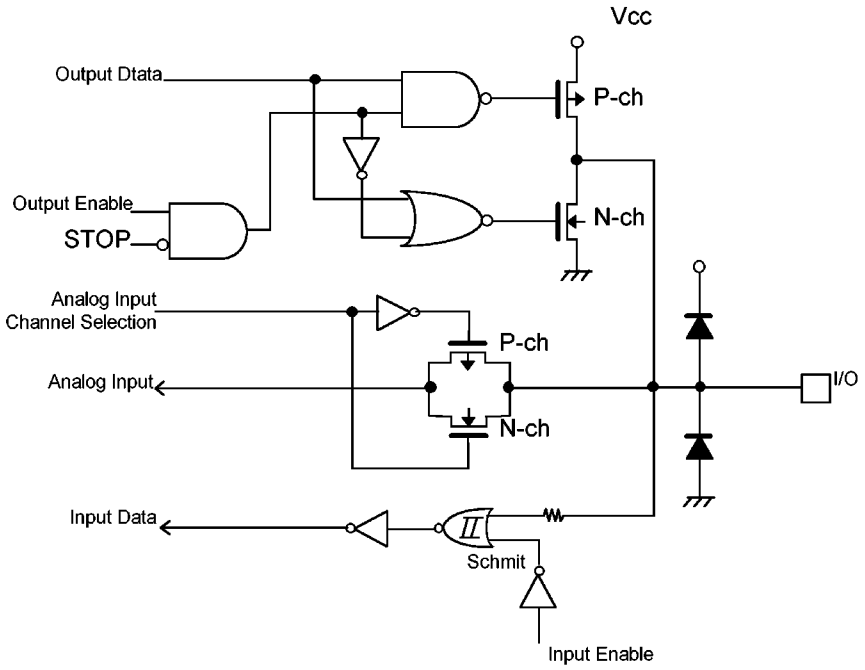
D0 to D15, PA0/PV-PH to PA2/CR (TPG00)



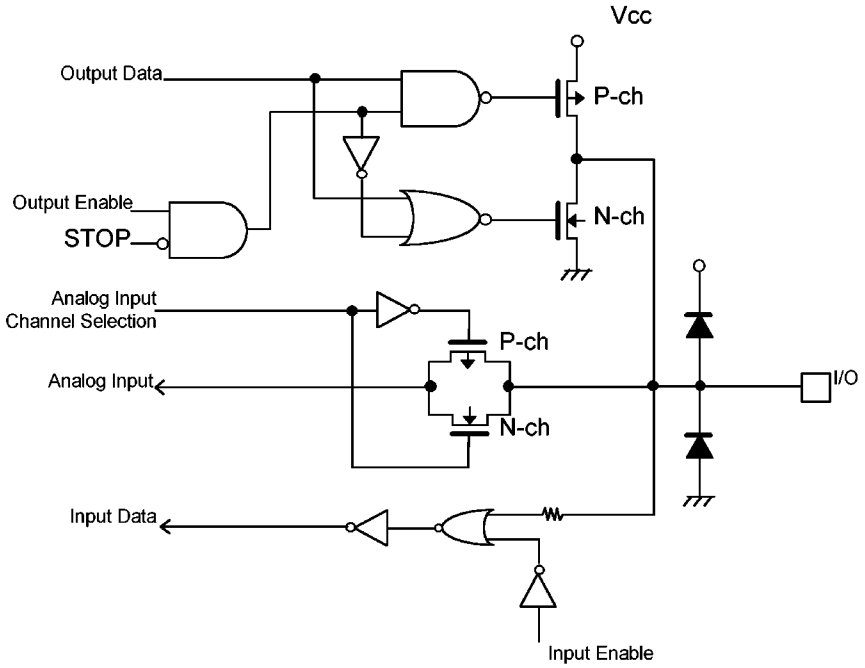
P40/AIN3 to P47/AIN10



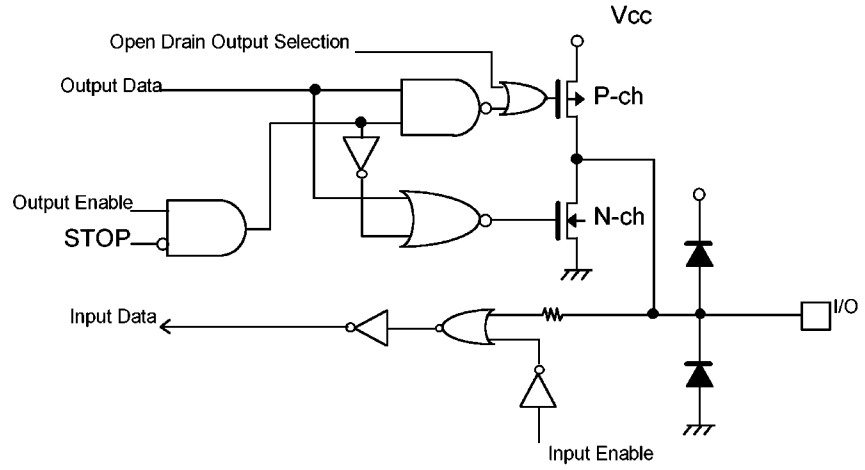
P55/TI5/AIN0 to P57/TI0/AIN2



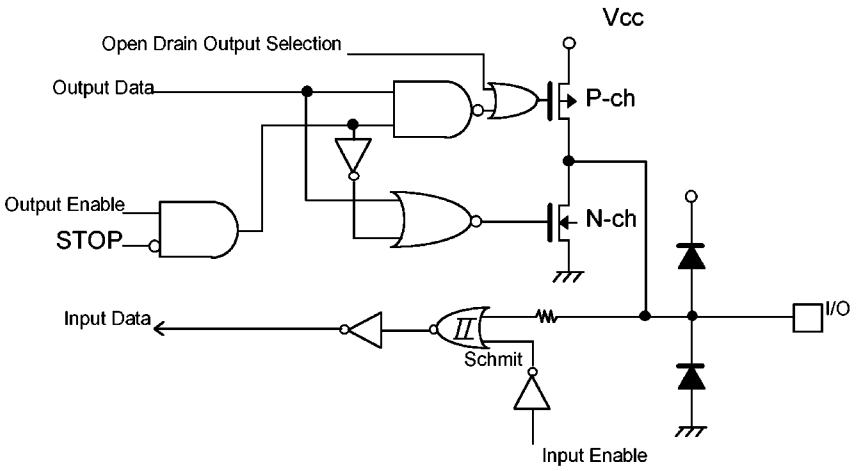
PC0/AIN11 to PC4/AIN15



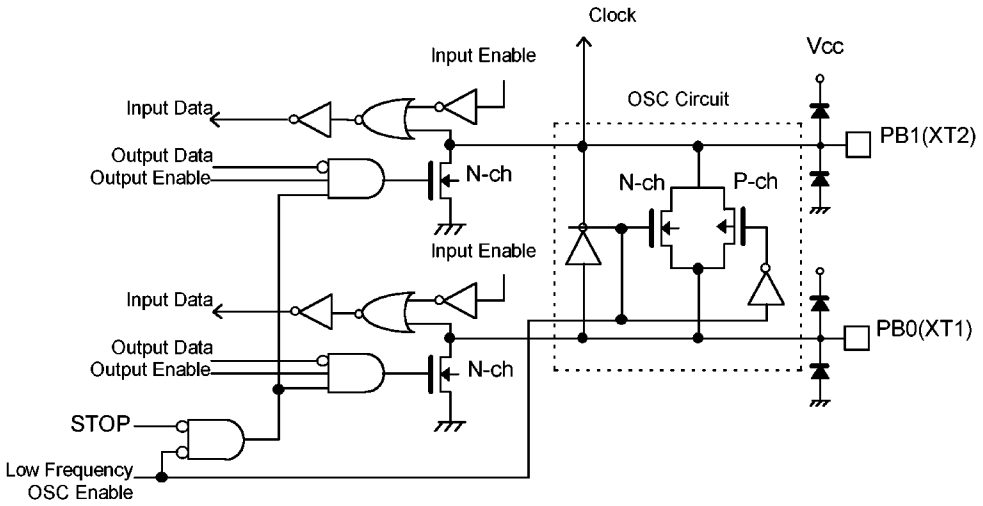
P60/PWM4/$\overline{CS0}$ to P67/PWM11, P90/TPG12 to P97/TPG11, PA3/PWM2 to PA5/PWM3/$\overline{HWR}$



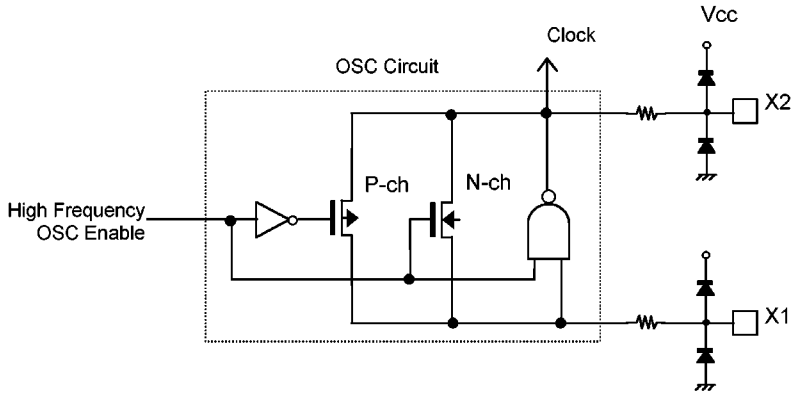
P70/TXD, P73/SDA0 to P75/SO0, P77/SCK0, PB2/SO1/SI1 to PB5/SCL1



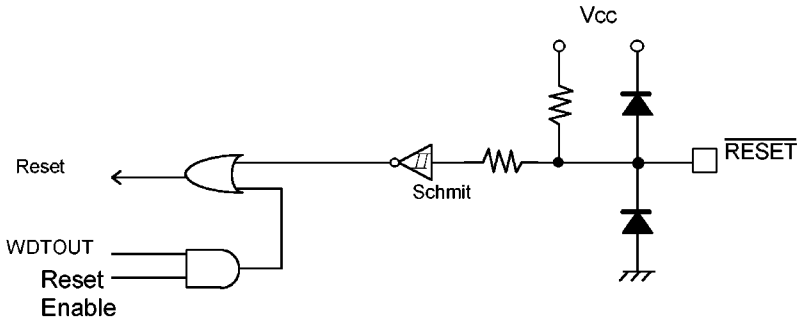
PB0/XT1, PB1/XT2



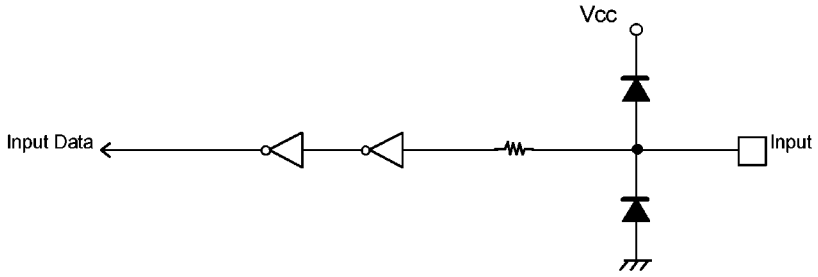
X1, X2



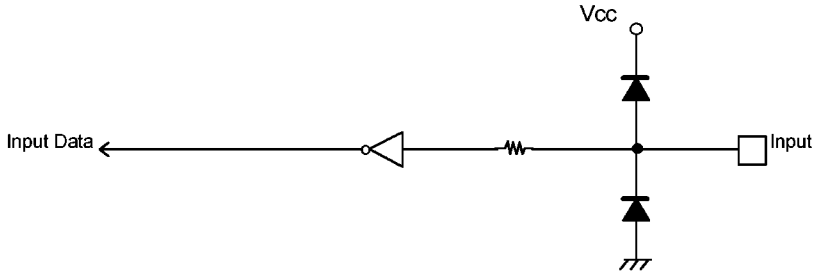
$\overline{\text{RESET}}$



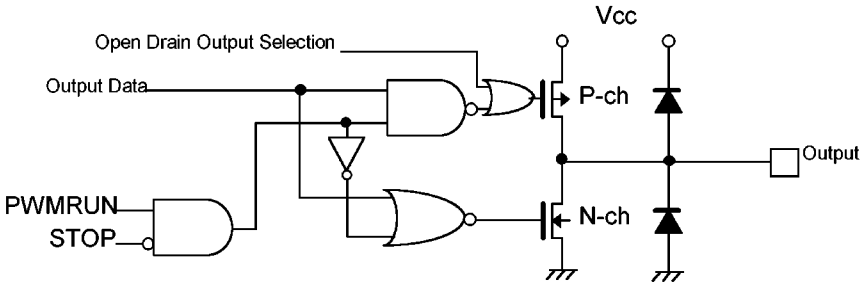
AM8/$\overline{16}$



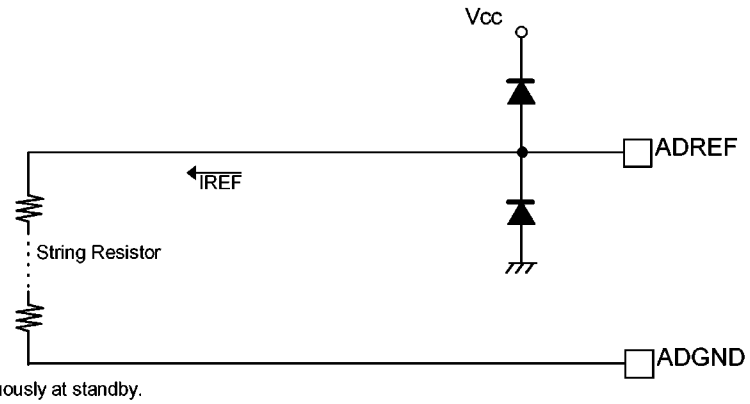
$\overline{\text{EA}}$



PWM0, PWM1



ADREF, ADGND



Note: IREF flows continuously at standby.

7. Points of Concern and Restriction

(1) Notation

① Explanation of a built-in I/O register : Register Symbol <Bit Symbol>

e.g.) TRUN<T0RUN> ··· Bit T0RUN of Register TRUN

② Read, Modify and Write Instruction

An instruction in which the CPU executes following by one instruction.
1. CPU reads data of the memory.
2. CPU modifies the data.
3. CPU writes the data to the same memory.

ex1) SET 3, (TRUN) set bit 3 of TRUN
ex2) INC 1, (100H) increment the data of 100H

- A sample Read, Modify and Write instructions using the TLCS-900

Exchange			
EX	(mem), R		
Arithmetic Operation			
ADD	(mem), R/#	ADC	(mem), R/#
SUB	(mem), R/#	SBC	(mem), R/#
INC	#3, (mem)	DEC	#3, (mem)
Logical Operation			
AND	(mem), R/#	OR	(mem), R/#
XOR	(mem), R/#		
Bit Manipulation			
STCF	#3/A, (mem)	SET	#3, (mem)
RES	#3, (mem)	TEST	#3, (mem)
CHG	#3, (mem)		
Rotate and Shift			
RLC	(mem)	RRC	(mem)
RL	(mem)	RR	(mem)
SLA	(mem)	SRA	(mem)
SLL	(mem)	SRL	(mem)
RLD	(mem)	RRD	(mem)

③ fc, fs, f_{FPH}, f_{SYS}, 1 state

The clock frequency input from pins X1 and X2 pin is called fc, and the clock frequency input from XT1, XT2 pin is called fs. The clock frequency selected by SYSCR1<SYSCK, GEAR2 to 0> is called system clock f_{FPH}, and the clock frequency given by f_{FPH} divided by 2 is called f_{SYS}. One cycle of f_{SYS} is called 1 state.

(2) Care Points

① $\overline{EA}$ EA, AM8 / $\overline{16}$ pin

Fix these pins Vcc or Vss unless changing voltage.

② Warmingup Counter

The warm-up counter operates when STOP mode is released even if the system is using an external oscillator. As a result, it takes warm-up time from inputting the releasing request to outputting the system clock.

③ WatchDog Timer

The watchdog timer starts operation immediately after the reset is released. When the watchdog timer is not used, disable it.

④ CPU (Micro DMA)

Only the "LDC cr, r", "LDC r, cr" instructions can be used to access the control registers in the CPU (like the transfer source address register (DMASn)).

⑤ POP SR instruction

Execute the POP SR instruction during DI.

⑥ Set "1" to SYSCR3<CLKEN> for stabilizing the operation current in slow mode and stop mode.