TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/L Series

TMP93CW46A

TOSHIBA CORPORATION

Semiconductor Company

Preface

Thank you very much for making use of Toshiba microcomputer LSIs. Before use this LSI, refer the section, “Points of Note and Restrictions”.

Especially, take care below cautions.

CAUTION

How to release the HALT mode

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2/RUN are not applicable to this case). (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

Low Voltage/Low Power

CMOS 16-Bit Microcontroller

TMP93CW46AF

1. Outline and Device Characteristics

The TMP93CW46AF is high-speed advanced 16-bit microcontroller to enable low voltage and low power consumption operation. The TMP93CW46AF is housed in 100-pin mini flat package.

The device characteristics are as follows:

(1) Original 16-bit CPU (900/L 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
- Micro DMA: 4 channels (1.6 μs/2 bytes at 20 MHz)

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

(3) Internal RAM: 4 Kbytes

Internal ROM: 128 Kbytes

(4) External memory expansion
- Can be expanded up to 16 Mbytes (for both programs and data).
- Can mix 8- and 16-bit external data buses.

…Dynamic data bus sizing

(5) 8-bit timer: 2 channels

(6) 8-bit PWM timer: 2 channels

(7) 16-bit timer: 2 channels

(8) Serial interface: 5 channels
- UART/synchronous modes: 4 channels
- UART mode: 1 channel

(9) 10-bit AD converter: 8 channels

(10) Watchdog timer

(11) Chip select/wait controller: 3 blocks

(12) Interrupt functions: 35

- 9 CPU interrupts … SWI instruction, and Illegal instruction
- 20 internal interrupts
- 6 external interrupts

] 7-level priority can be set.

(13) I/O ports: 79

- Large current output: 6 pins, LED direct drive

(14) Standby function

4 HALT modes (RUN, IDLE2, IDLE1, STOP)

(15) Clock gear function

- High-frequency clock can be changed fc to fc/16.
- Dual clock operation

(16) Operating voltage

- V_{CC} = 2.7 to 5.5 V

(17) Package: P-LQFP100-1414-0.50F

Note: Note that TMP93CW46A is different from OTP type TMP93PW46A in the electrical characteristics as follows. See the respective electrical characteristics for details.

- Power supply current I_{CC}
- Large current port I_{OLA}

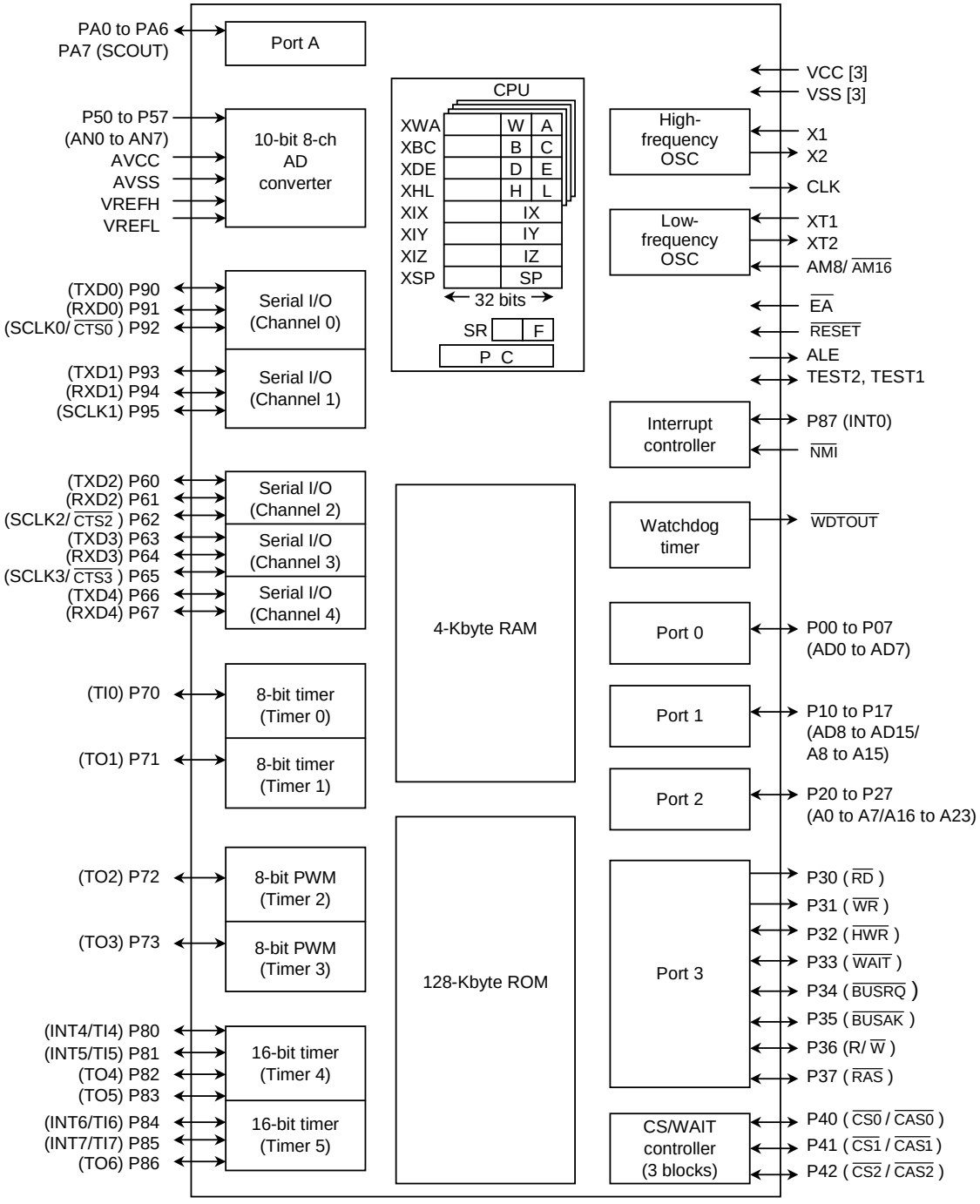
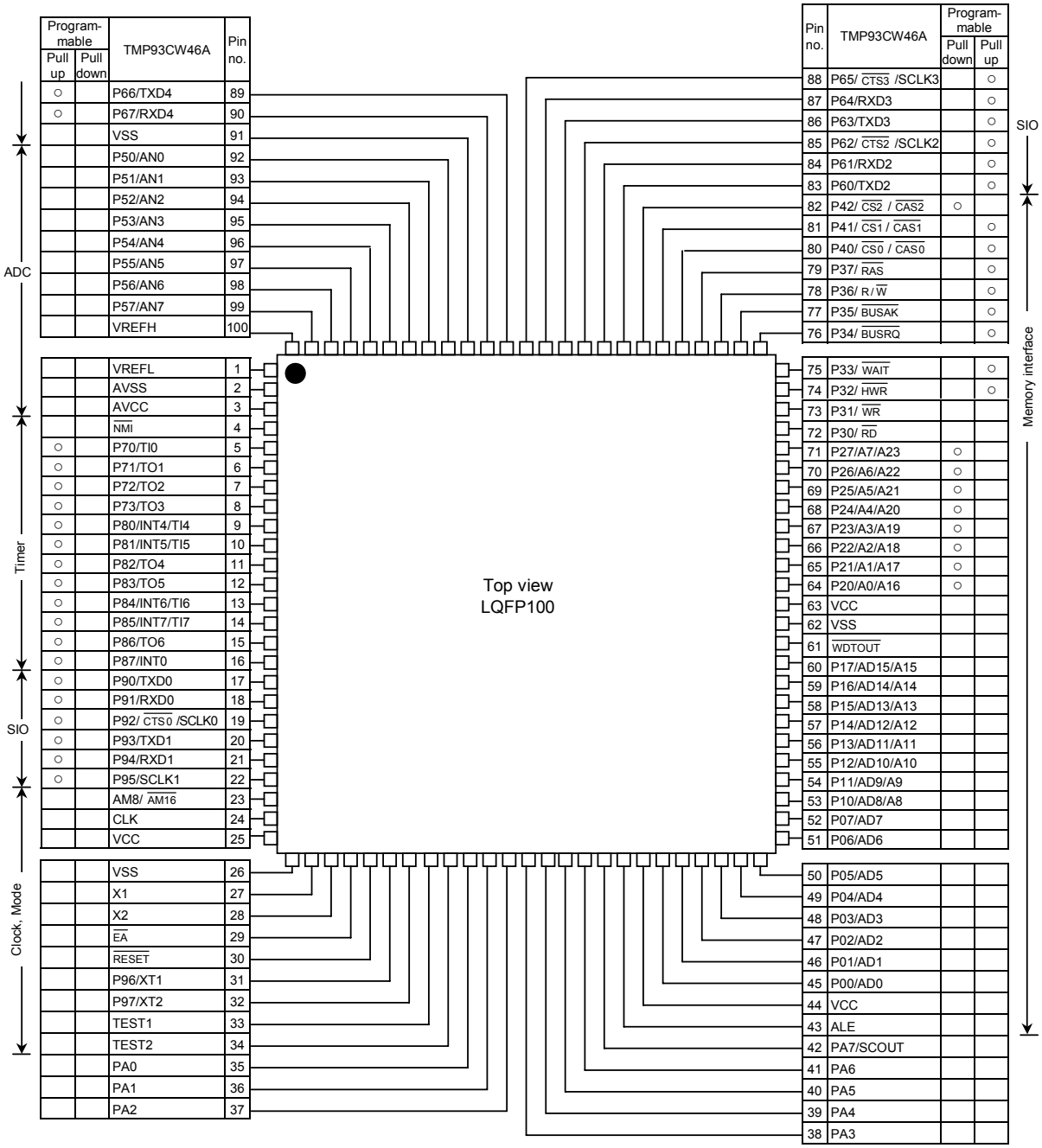


Figure 1.1 TMP93CW46A Block Diagram

2. Pin Assignment and Functions

The assignment of input/output pins for the TMP93CW46AF, their names and outline functions are described below.



2.1 Pin Assignment

Figure 2.1.1 shows pin assignment of the TMP93CW46AF.

Figure 2.1.1 Pin Assignment (100-Pin LQFP)

2.2 Pin Names and Functions

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

Table 2.2.1 to Table 2.2.4 show pin names and functions.

Table 2.2.1 Pin Names and Functions (1/4)

Pin Names	Number of Pins	I/O	Functions
P00 to P07 AD0 to AD7	8	I/O 3 states	Port 0: I/O port that allows selection of I/O on a bit basis Address/Data (Lower): Bits 0 to 7 of address/data bus
P10 to P17 AD8 to AD15 A8 to A15	8	I/O 3 states Output	Port 1: I/O port that allows selection of I/O on a bit basis Address data (Upper): Bits 8 to 15 of address/data bus Address: 8 to 15 of address bus
P20 to P27 A0 to A7 A16 to A23	8	I/O Output Output	Port 2: I/O port that allows selection of I/O on a bit basis (with pull-down resistor) Address: Bits 0 to 7 of address bus Address: Bits 16 to 23 of address bus
P30 $\overline{RD}$	1	Output Output	Port 30: Output port Read: Strobe signal for reading external memory
P31 $\overline{WR}$	1	Output Output	Port 31: Output port Write: Strobe signal for writing data on pins AD0 to AD7
P32 $\overline{HWR}$	1	I/O Output	Port 32: I/O port (with pull-up resistor) High write: Strobe signal for writing data on pins AD8 to AD15
P33 $\overline{WAIT}$	1	I/O Input	Port 33: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait
P34 $\overline{BUSRQ}$	1	I/O Input	Port34: I/O port (with pull-up resistor) Bus request: Signal used to request high impedance for AD0 to AD15, A0 to A23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$, $\overline{CS0}$, $\overline{CS1}$, and $\overline{CS2}$ pins. (For external DMAC)
P35 $\overline{BUSAK}$	1	I/O Output	Port 35: I/O port (with pull-up resistor) Bus acknowledge: Signal indicating that AD0 to AD15, A0 to A23, $\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$, $\overline{CS0}$, $\overline{CS1}$, and $\overline{CS2}$ pins are at high impedance after receiving BUSRQ. (For external DMAC)
P36 R/$\overline{W}$	1	I/O Output	Port 36: I/O port (with pull-up resistor) Read/Write: 1 represents read or dummy cycle. 0 represents write cycle.
P37 $\overline{RAS}$	1	I/O Output	Port 37: I/O port (with pull-up resistor) Row address strobe: Outputs $\overline{RAS}$ strobe for DRAM.
P40 $\overline{CS0}$ $\overline{CAS0}$	1	I/O Output Output	Port 40: I/O port (with pull-up resistor) Chip select 0: Outputs 0 when address is within specified address area. Column address strobe 0: Outputs $\overline{CAS}$ strobe for DRAM when address is within specified address area.

Note: This device's built-in memory or built-in I/O cannot be accessed with the external DMA controller, using the $\overline{BUSRQ}$ and $\overline{BUSAK}$ signals.

Table 2.2.2 Pin Names and Functions (2/4)

Pin Names	Number of Pins	I/O	Functions
P41 $\overline{CS1}$ $\overline{CAS1}$	1	I/O Output Output	Port 41: I/O port (with pull-up resistor) Chip select 1: Outputs 0 if address is within specified address area. Column address strobe 1: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area.
P42 $\overline{CS2}$ $\overline{CAS2}$	1	I/O Output Output	Port 42: I/O port (with pull-down resistor) Chip select 2: Outputs 0 if address is within specified address area. Column address strobe 2: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area.
P50 to P57 AN0 to AN7	8	Input Input	Port 5: Input port Analog input: Analog signal input for AD converter
VREFH	1	Input	Pin for high level reference voltage input to AD converter
VREFL	1	Input	Pin for low level reference voltage input to AD converter
P60 TXD2	1	I/O Output	Port 60: I/O port (with pull-up resistor) Serial send data 2
P61 RXD2	1	I/O Input	Port 61: I/O port (with pull-up resistor) Serial receive data 2
P62 $\overline{CTS2}$ SCLK2	1	I/O Input I/O	Port 62: I/O port (with pull-up resistor) Serial data send enable 2 (Clear to send) Serial Clock I/O 2
P63 TXD3	1	I/O Output	Port 63: I/O port (with pull-up resistor) Serial receive data 3
P64 RXD3	1	I/O Input	Port 64: I/O port (with pull-up resistor) Serial receive data 3
P65 $\overline{CTS3}$ SCLK3	1	I/O Input I/O	Port 65: I/O port (with pull-up resistor) Serial data send enable 3 (Clear to send) Serial Clock I/O 3
P66 TXD4	1	I/O Output	Port 66: I/O port (with pull-up resistor) Serial send data 4
P67 RXD4	1	I/O Input	Port 67: I/O port (with pull-up resistor) Serial receive data 4
P70 TI0	1	I/O Input	Port 70: I/O port (with pull-up resistor) Timer input 0: Timer 0 input
P71 TO1	1	I/O Output	Port 71: I/O port (with pull-up resistor) Timer output 1: Timer 0 or 1 output
P72 TO2	1	I/O Output	Port 72: I/O port (with pull-up resistor) PWM output 2: 8-bit PWM timer 2 output
P73 TO3	1	I/O Output	Port 73: I/O port (with pull-up resistor) PWM output 3: 8-bit PWM timer 3 output

Table 2.2.3 Pin Names and Functions (3/4)

Pin Names	Number of Pins	I/O	Functions
P80	1	I/O	Port 80: I/O port (with pull-up resistor)
TI4		Input	Timer input 4: Timer 4 count/capture trigger signal input Interrupt request pin 4:
INT4		Input	Interrupt request pin with programmable rising/falling edge
P81	1	I/O	Port 81: I/O port (with pull-up resistor)
TI5		Input	Timer input 5: Timer 4 count/capture trigger signal input
INT5		Input	Interrupt request pin 5: Interrupt request pin with rising edge
P82	1	I/O	Port 82: I/O port (with pull-up resistor)
TO4		Output	Timer output 4: Timer 4 output pin
P83	1	I/O	Port 83: I/O port (with pull-up resistor)
TO5		Output	Timer output 5: Timer 4 output pin
P84	1	I/O	Port 84: I/O port (with pull-up resistor)
TI6		Input	Timer input 6: Timer 5 count/capture trigger signal input
INT6		Input	Interrupt request pin 6: Interrupt request pin with programmable rising/falling edge
P85	1	I/O	Port 85: I/O port (with pull-up resistor)
TI7		Input	Timer input 7: Timer 5 count/capture trigger signal input
INT7		Input	Interrupt request pin 7: Interrupt request pin with rising edge
P86	1	I/O	Port 86: I/O port (with pull-up resistor)
TO6		Output	Timer output 6: Timer 5 output pin
P87	1	I/O	Port 87: I/O port (with pull-up resistor)
INT0		Input	Interrupt request pin 0: Interrupt request pin with programmable level/rising edge
P90	1	I/O	Port 90: I/O port (with pull-up resistor)
TXD0		Output	Serial send data 0
P91	1	I/O	Port 91: I/O port (with pull-up resistor)
RXD0		Input	Serial receive data 0
P92	1	I/O	Port 92: I/O port (with pull-up resistor)
$\overline{CTS0}$		Input	Serial data send enable 0 (Clear to send)
SCLK0		I/O	Serial Clock I/O 0
P93	1	I/O	Port 93: I/O port (with pull-up resistor)
TXD1		Output	Serial send data 1
P94	1	I/O	Port 94: I/O port (with pull-up resistor)
RXD1		Input	Serial receive data 1
P95	1	I/O	Port 95: I/O port (with pull-up resistor)
SCLK1		I/O	Serial clock I/O 1
PA0 to PA5	6	I/O	Port A0 to A5: I/O ports (large current output)
PA6	1	I/O	Port A6: I/O port

Table 2.2.4 Pin Names and Functions (4/4)

Pin Names	Number of Pins	I/O	Functions
PA7 SCOUT	1	I/O Output	Port A7: I/O port System clock output: Outputs system clock or 2 times oscillation clock for synchronizing to external circuit.
$\overline{\text{WDTOUT}}$	1	Output	Watchdog timer output pin
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at rising edge by program.
CLK	1	Output	Clock output: Outputs "System clock ÷ 2" clock. Pulled-up during reset. Can be disabled for reducing noise.
$\overline{\text{EA}}$	1	Input	Fixed to "1".
AM8/ $\overline{\text{AM16}}$	1	Input	Fixed to "1".
ALE	1	Output	Address latch enable (Can be disabled for reducing noise.)
$\overline{\text{RESET}}$	1	Input	Reset: Initializes LSI (with pull-up resistor).
X1/X2	2	I/O	High-frequency oscillator connecting pin
XT1 P96	1	Input I/O	Low-frequency oscillator connecting pin Port 96: I/O port (Open-drain output)
XT2 P97	1	Output I/O	Low-frequency oscillator connecting pin Port 97: I/O port (Open-drain output)
TEST1/TEST2	2	Output/Input	TEST1 should be connected with TEST2 pin. Do not connect to any other pins.
VCC	3		Power supply pin (All VCC pins are connected to the power supply source.)
VSS	3		GND pin (All VSS pins are connected to the GND (0 V).)
AVCC	1		Power supply pin for AD converter
AVSS	1		GND pin for AD converter (0 V)

Note: Built-in pull-up/pull-down resistors can be released from the pins other than the $\overline{\text{RESET}}$ pin by software.

3. Operation

This section describes the functions and basic operational blocks of the TMP93CW46A devices.

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

3.1 CPU

The TMP93CW46A device has a built-in high-performance 16-bit CPU (900/L CPU). (For CPU operation, see TLCS-900/L CPU in the previous chapter.)

This section describes CPU functions unique to the TMP93CW46A that are not described in the previous chapter.

3.1.1 Reset

When resetting the TMP93CW46A microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then set the $\overline{\text{RESET}}$ input to low level at least for 10 system clocks (16 μs at 20 MHz). Thus, when turn on the switch, be set to the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the $\overline{\text{RESET}}$ input to low level at least for 10 system clocks.

Clock gear is initialized 1/16 mode by reset operation. It means that the system clock mode f_{SYS} is set to fc/32 (= fc/16 × 1/2).

When reset is accepted, the CPU sets as follows:

- Program counter (PC) according to reset vector that is stored 8000H to 8002H.

  PC<7:0> ← Data located at 8000H

  PC<15:8> ← Data located at 8001H

  PC<23:16> ← Data located at 8002H

Note: The address in which the reset vector is stored depends on the respective derivative products.

- Stack pointer (XSP) for system mode to 100H.
- Status register <IFF2:0> to 111. (Sets mask register to interrupt level 7.)
- Status register <MAX> to 1. (Sets to maximum mode.)
- Status register <REP2:0> 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 are 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.
- Sets $\overline{\text{WDTOUT}}$ pin to "0". (Resetting enables the watchdog timer.)
- Pulls up the CLK pin to 1.
- Sets the ALE pin to high impedance (High-Z).

Note 1: 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.

Note 2: The CLK pin is pulled up during reset. When the voltage is put down externally, there is possible to cause malfunctions.

Figure 3.1.1 shows the reset timing chart of TMP93CW46A.

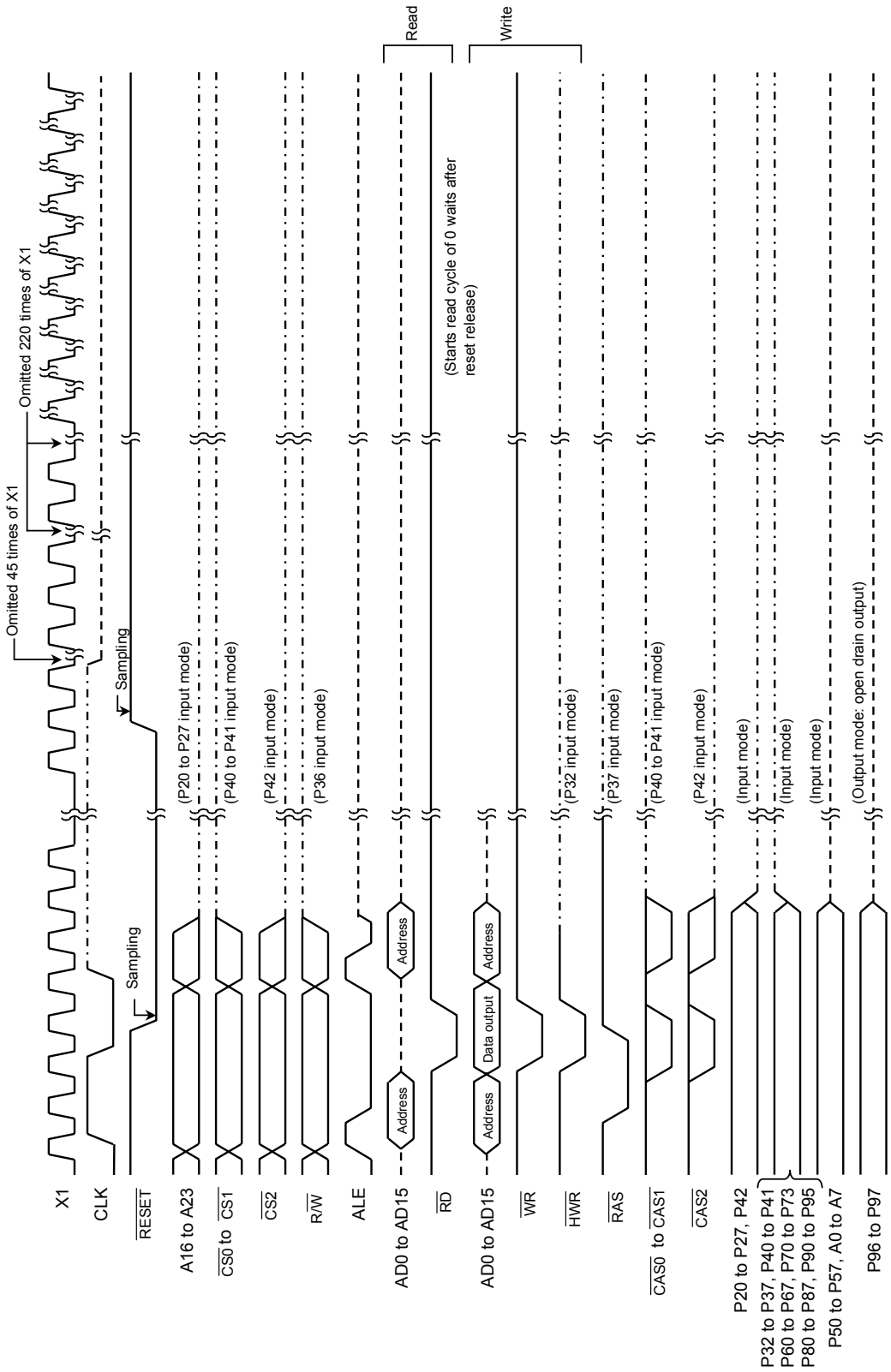
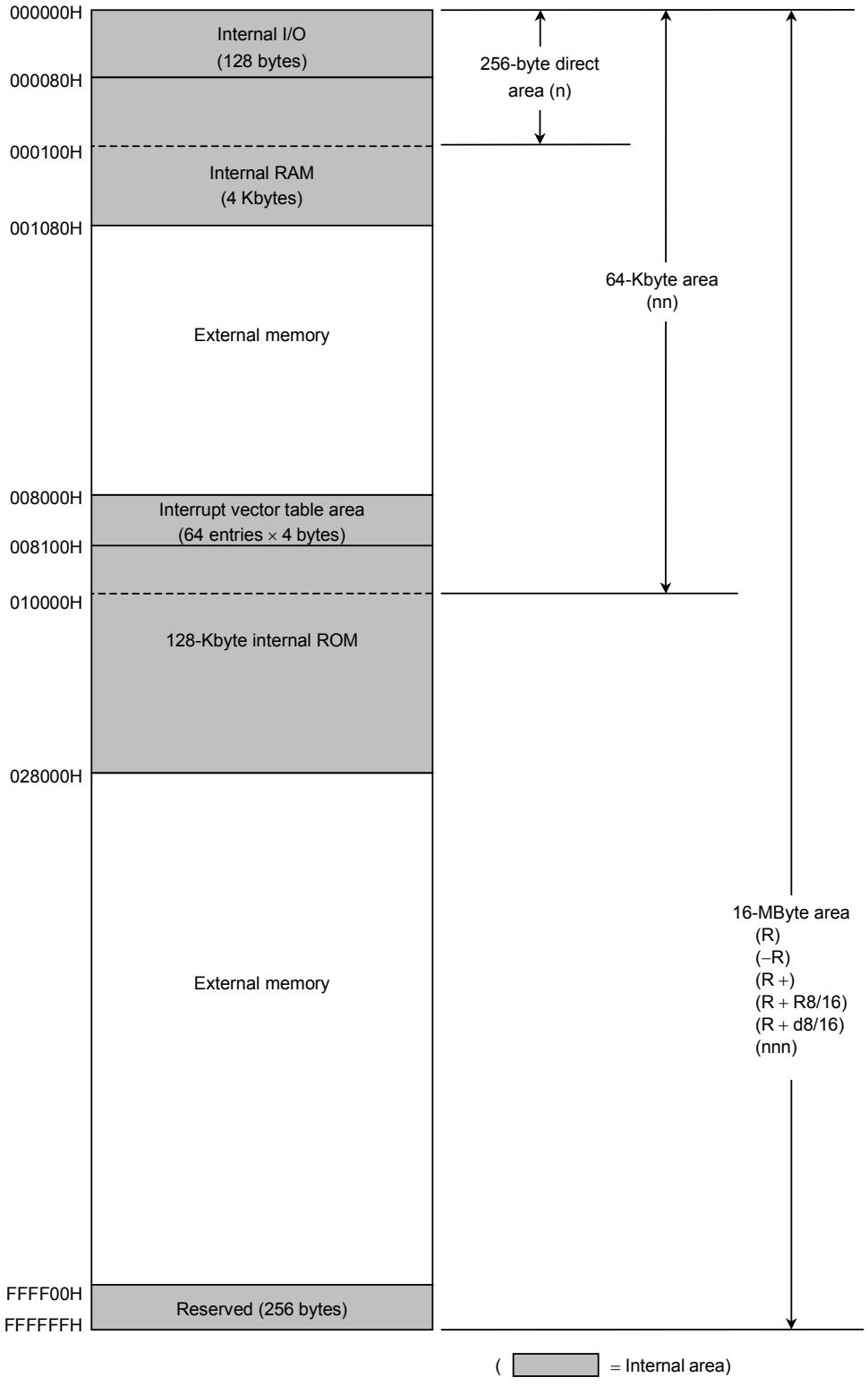


Figure 3.1.1 TMP93CW46A Reset Timing Chart

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP93CW46A.



Note: The 256-byte area from FFFF00H to FFFFFFH can not be used.

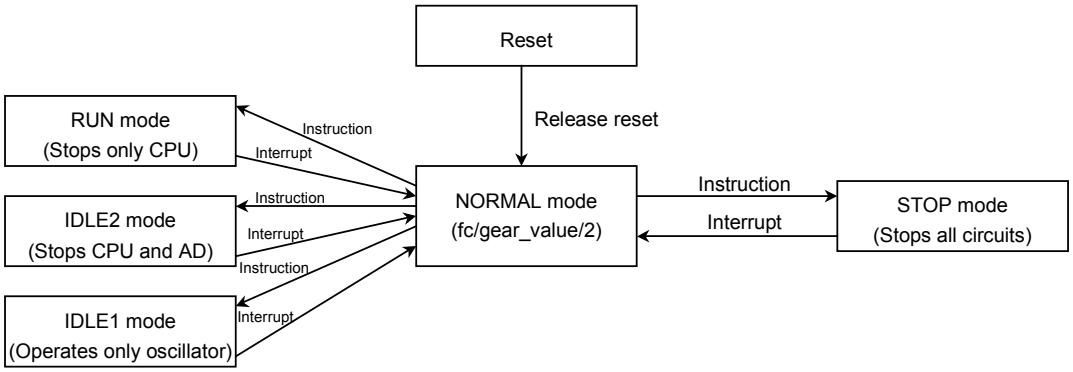
Figure 3.2.1 Memory Map

3.3 Standby Function

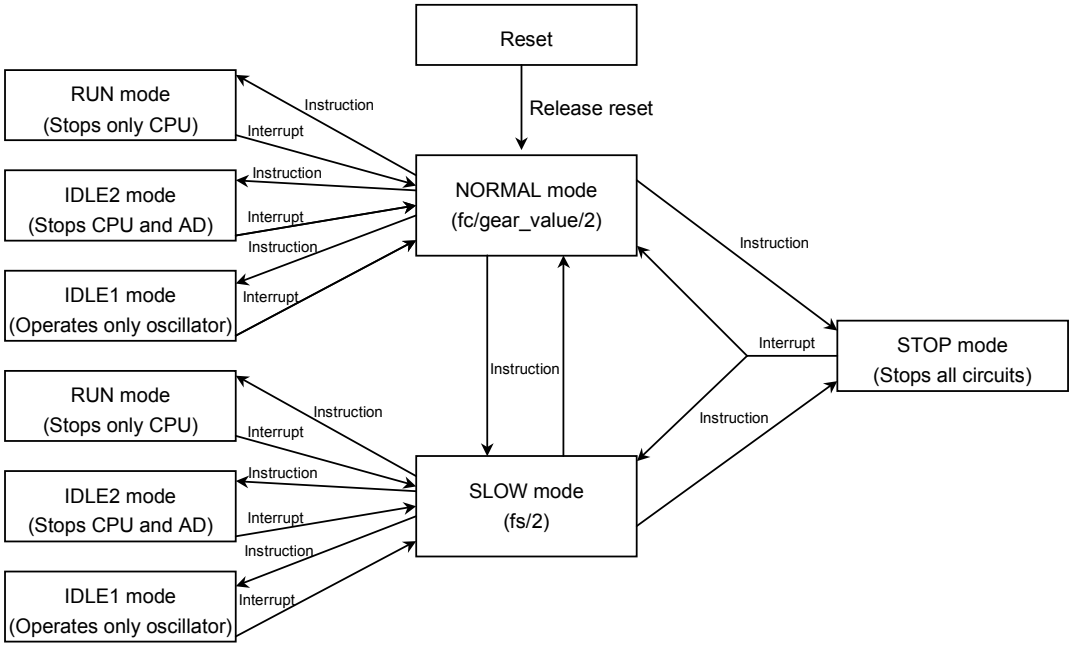
Standby control circuits consist of (1) System clock controller, (2) Prescaler clock controller, and (3) Standby controller.

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.



(a) Signal 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 selected by SYSCR1<SYSCK>, <GEAR2:0> 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

<table>
<tr><th colspan="2" rowspan="2">Operating Mode</th><th colspan="2">Oscillator</th><th rowspan="2">CPU</th><th rowspan="2">Internal I/O</th><th rowspan="2">System Clock f_{SYS}</th></tr>
<tr><th>High Frequency (fc)</th><th>Low Frequency (fs)</th></tr>
<tr><td rowspan="6">Single clock</td><td>RESET</td><td rowspan="5">Oscillation</td><td rowspan="6">Stop</td><td>Reset</td><td>Reset</td><td>fc/32</td></tr>
<tr><td>NORMAL</td><td>Operate</td><td rowspan="2">Operate</td><td rowspan="3">Programmable (fc/2, fc/4, fc/8, fc/16, fc/32)</td></tr>
<tr><td>RUN</td><td rowspan="3">Stop</td></tr>
<tr><td>IDLE2</td><td>Stop only AD</td></tr>
<tr><td>IDLE1</td><td rowspan="2">Stop</td></tr>
<tr><td>STOP</td><td>Stop</td><td>Stop</td></tr>
<tr><td rowspan="6">Dual clock</td><td>RESET</td><td rowspan="2">Oscillation</td><td>Stop</td><td>Reset</td><td>Reset</td><td>fc/32</td></tr>
<tr><td>NORMAL</td><td>Programmable</td><td rowspan="2">Operate</td><td rowspan="2">Operate</td><td>Programmable (fc/2, fc/4, fc/8, fc/16, fc/32)</td></tr>
<tr><td>SLOW</td><td>Programmable</td><td>Oscillation</td><td>fs/2</td></tr>
<tr><td>RUN</td><td colspan="2" rowspan="3">Oscillator using as system clock: Oscillation
Other oscillator: Programmable</td><td rowspan="3">Stop</td><td rowspan="2">Stop only AD</td><td rowspan="3">Programmable (fc/2, fc/4, fc/8, fc/16, fc/32, fs/2)</td></tr>
<tr><td>IDLE2</td></tr>
<tr><td>IDLE1</td><td rowspan="2">Stop</td></tr>
<tr><td>STOP</td><td colspan="2">Stop</td><td>Stop</td></tr>
</table>

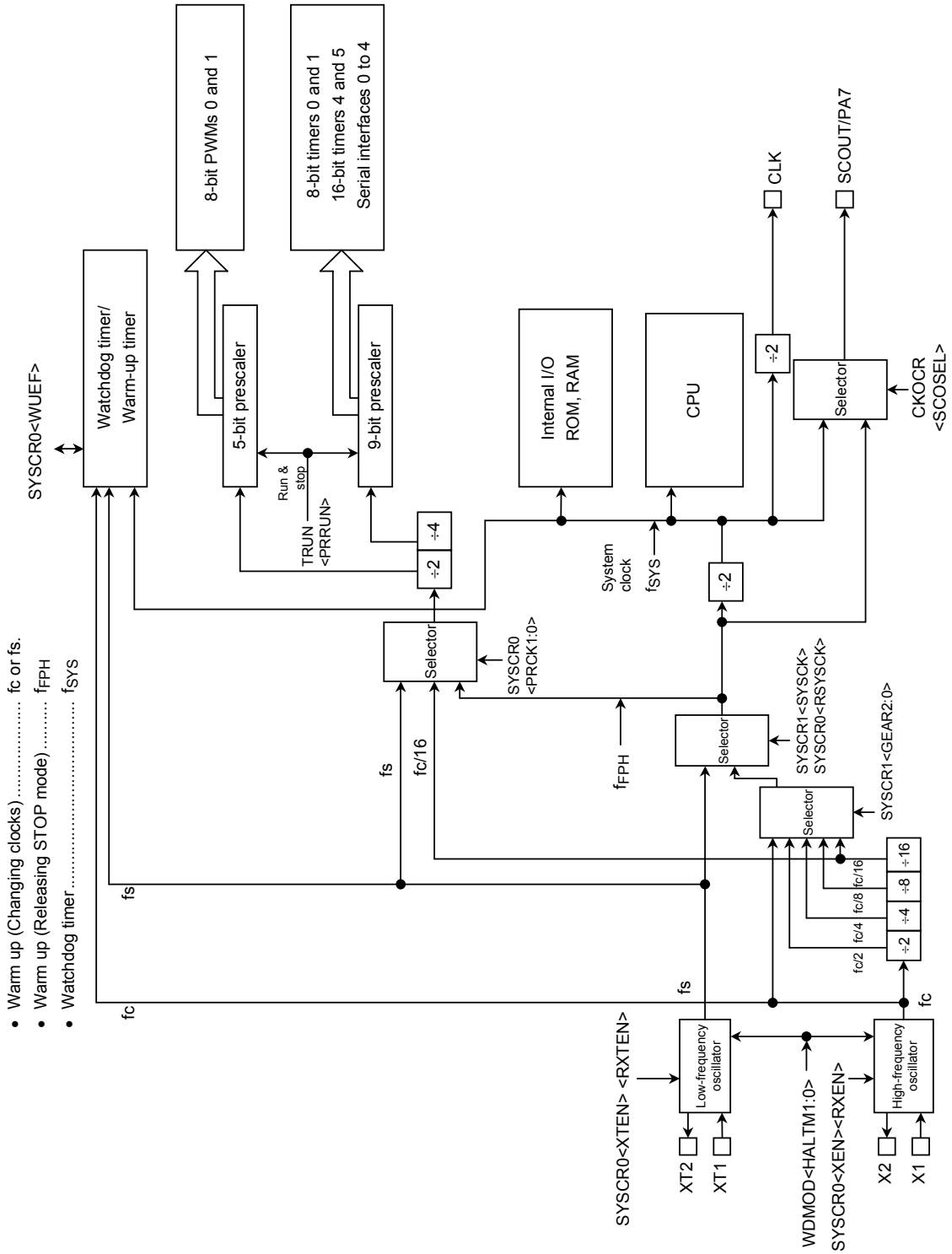


Figure 3.3.2 Block Diagram of Dual Clock, Standby Circuits

		7	6	5	4	3	2	1	0
SYSCR0 (006EH)	Bit symbol	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
	Read/Write	R/W							
	After reset	1	0	1	0	0	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 (fs) after released STOP mode 0: Stop 1: Oscillation	Slect clock after released STOP mode 0: fc 1: fs	Warm-up timer (Write) 0: Don't care 1: Start timer (Read) 0: End warm-up 1: Not end warm-up	Select prescaler clock 00: f_{FPH} 01: fs 10: fc/16 11: (Reserved)	

		7	6	5	4	3	2	1	0
SYSCR1 (006FH)	Bit symbol					SYSCK	GEAR2	GEAR1	GEAR0
	Read/Write					R/W			
	After reset					0	1	0	0
	Function					Select system clock 0: fc 1: fs	Select gear value of high frequency (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (Reserved) 110: (Reserved) 111: (Reserved)		

		7	6	5	4	3	2	1	0
WDMOD (005CH)	Bit symbol	WDTE	WDTP1	WDTP0	WARM	HALTM0	HALTM0	RESCR	DRVE
	Read/Write	R/W							
	After reset	1	0	0	0	0	0	0	0
	Function	WDT control 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}/ inputted frequency 1: 2^{16}/ inputted frequency	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		1: Connects WDT output to $\overline{RESET}$ pin internally.	1: Drives pin even in STOP mode

		7	6	5	4	3	2	1	0
CKOCR (006DH)	Bit symbol					SCOSEL	SCOEN	ALEEN	CLKEN
	Read/Write					R/W			
	After reset					0	1	0/1 (Note 2)	0/1 (Note 2)
	Function					SCOUT select 0: f_{FPH} 1: f_{SYS}	SCOUT output control 0: I/O ports 1: SCOUT output	ALE pin output control 0: High-Z output 1: ALE output	CLK pin output control 0: High-Z output 1: CLK output

Note 1: SYSCR1<bit7:4> and CKOCR<bit7:4> are read as "1".

Note 2: After reset, <ALEEN> and <CLKEN> bits are "0". (ALE and CLK is high impedance.)
During reset, CLK pin is internally pulled up regardless of the products.

Note 3: Writing "0" to SYSCR1<SYSCK> enables the low frequency oscillation circuit regardless of the value of SYSCR0<XTEN>.
Additionally, writing "1" to <SYSCK> register enables low frequency oscillation circuit regardless of the value of SYSCR0<XTEN>.

Figure 3.3.3 I/O Register about Dual Clock and Standby

3.3.1 System Clock Controller

The system clock controller generates system clock (f_{SYS}) for CPU core and internal I/O. It contains two oscillation circuits and clock gear circuit for high frequency (fc). The register SYSCR1<SYSCK> changes system clock to either fc or fs, SYSCR0<XEN>, <XTEN> controls enable/disable each oscillator, SYSCR1<GEAR2:0> changes high frequency clock gear to either 1, 2, 4, 8 or 16 (fc, fc/2, fc/4, fc/8 or fc/16), and these functions can reduce the power consumption.

The system clock (f_{SYS}) is set to fc/32 (fc/16 × 1/2) because of <XEN> = "1", <XTEN> = "0", <SYSCK> = "0", <GEAR2:0> = "100" by resetting. For example, f_{SYS} is set to 0.5 MHz by resetting the case of 16 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 96, 97 function. Therefore the case of single clock mode, the XT1, XT2 pins can be used as I/O port pins.

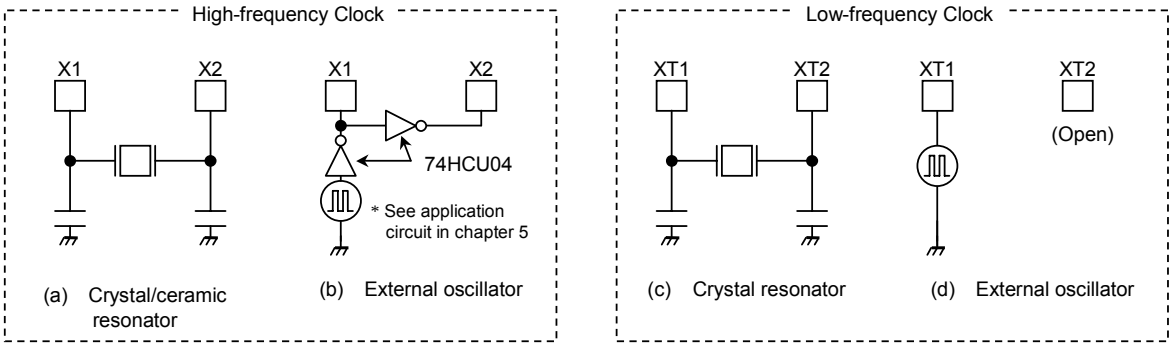


Figure 3.3.4 Examples of Resonator Connection

Note 1: Note on using low-frequency oscillation circuit

To connect the low-frequency resonator to port 96, 97, it is necessary to set the following to reduce the power consumption.
(Connecting with resonators)
P9CR<P96C, P97C> = "11", P9<P96:97> = "00"
(Connecting with oscillators)
P9CR<P96C, P97C> = "11", P9<P96:97> = "10"

Note 2: Accurate adjustment of the oscillation frequency

The CLK pin outputs 1/2 system clock frequency (f_{SYS}/2) to monitor the oscillation clock. With a system requiring adjustment of the oscillation frequency, the adjusting program must be created beforehand.

(1) Switching 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.

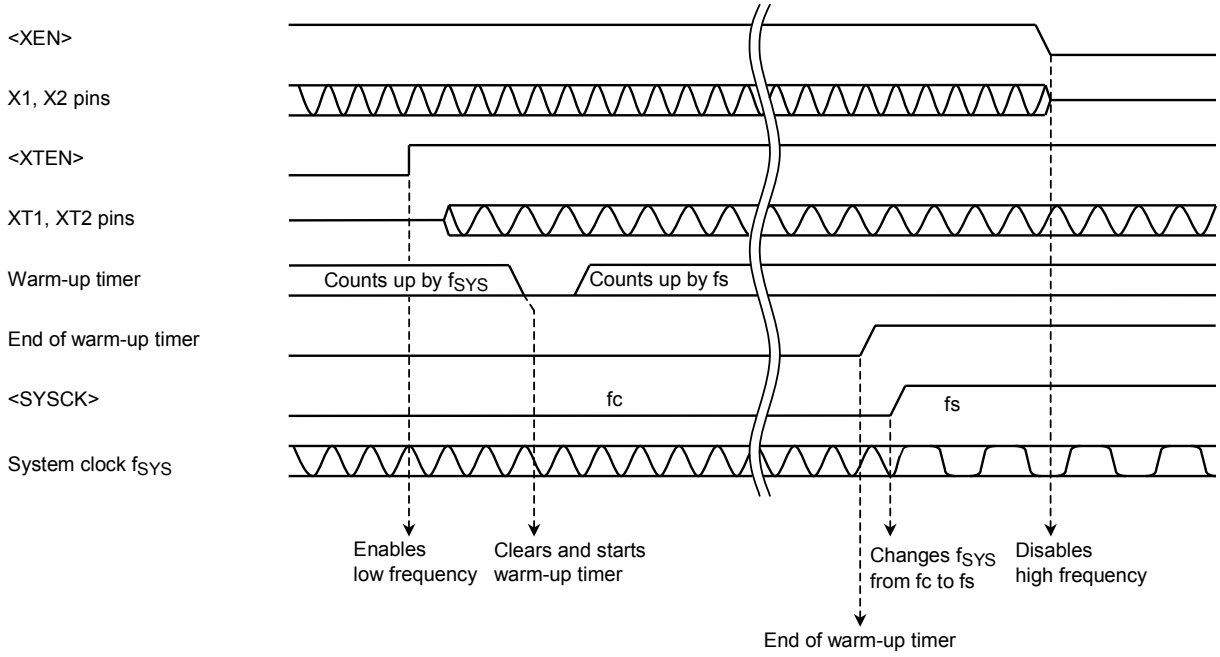
Table 3.3.2 Warm-up Time

Warm-up Time WDMOD<WARM>	Change to NORMAL	Change to SLOW	
0 (2^{14}/frequency)	0.8192 (ms)	500 (ms)	at fc = 20 MHz,
1 (2^{16}/frequency)	3.2768 (ms)	2000 (ms)	fs = 32.768 kHz

Setting example 1

The case of changing from high frequency (fc) to low frequency (fs).

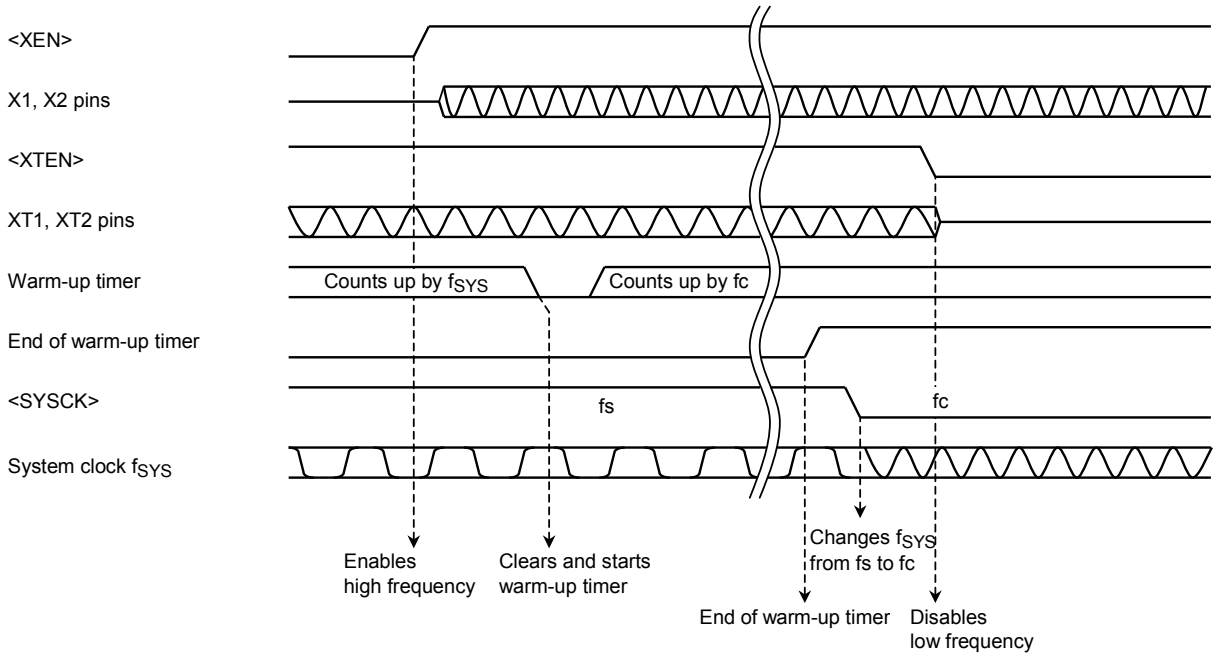
```
SYSCR0    EQU    006EH
SYSCR1    EQU    006FH
WDCR      EQU    005DH
WDMOD     EQU    005CH
          RES    7, (WDMOD)      ;
          LD     (WDCR), B1H     ; } Disables watchdog timer.
          SET    4, (WDMOD)      ; Sets warm-up time to 2^16/fs.
          SET    6, (SYSCR0)     ; Enables low-frequency oscillation
          SET    2, (SYSCR0)     ; Clears and starts warm-up timer.
WUP:      BIT    2, (SYSCR0)     ;
          JR     NZ, WUP         ; } Detects end of warm-up timer.
          SET    3, (SYSCR1)     ; Changes fSYS from fc to fs.
          RES    7, (SYSCR0)     ; Disables high-frequency oscillation.
          SET    7, (WDMOD)      ; Enables watchdog timer.
```



Setting example 2

The case of changing from low frequency (fs) to 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 starts 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)      ; Enable watchdog timer
```



(2) Clock gear controller

When the high-frequency clock fc is selected at SYSCR1<SYSCK> = "0", the clock gear select register SYSCR1<GEAR2:0> sets f_{FPH} to either fc, fc/2, fc/4, fc/8, fc/16. Switching f_{FPH} with the clock gear reduces the power consumption.

Setting Example 3

The case of changing gear value of high frequency

```
SYSCR1    EQU   006FH
          LD    (SYSCR1), XXXX0000B    ; Changes fSYS to fc/2
          LD    (SYSCR1), XXXX0100B    ; Changes fSYS to fc/32
X: Don't care
```

(High-speed clock gear changing)

To change the clock gear, write the register value to SYSCR1<GEAR2:0> register. It is necessary the warm-up time until changing after writing the register value.

There is the possibility that the instruction next to the clock gear changing instruction is executed by the clock gear before changing. To execute the instruction next to the clock gear switching instruction by the clock gear after changing, input the dummy instruction as follows (Instruction to execute the write cycle).

(Example)

```
SYSCR1    EQU   006FH
          LD    (SYSCR1), XXXX0001B    ; Changes fSYS to fc/4.
          LD    (DUMMY), 00H           ; Dummy instruction
      Instruction to be executed by the clock gear after changing
```

3.3.2 Prescaler Clock Controller

The 9-bit prescaler provides a clock to 8-bit timers 0 and 1, 16-bit timers 4 and 5, and serial interfaces 0 to 4. And the 5-bit prescaler provides a clock to 8-bit PWMs 0 and 1.

The clock input to the 5-bit prescaler is a clock divided by 2 which is selected by f_{FPH}, fc/16 or fs by SYSCR0<PRCK1:0> register.

The clock input to the 9-bit prescaler is a clock divided by 4 which is selected either f_{FPH} or fc/16 by SYSCR0<PRCK1:0> register.

<PRCK1:0> register is initialized to "00" by resetting. The clock selected by <SYSCK> is input.

When the IDLE 1 mode (Operates only oscillator) is used, set TRUN<PRRUN> to "0" to reduce the power consumption of 9, 5-bit prescaler before "HALT" instruction is executed.

3.3.3 Internal Clock Pin Output Function

(1) PA7/SCOUT pin

PA7/SCOUT pin outputs the internal clocks f_{FPH} or f_{SYS}.

The port A control register PACR<PA7C> and the clock output control register CKOCR<SCOEN, SCOSEL> specifies the clock and the pins. PA7/SCOUT pin is used as the input port by reset.

Table 3.3.3 shows pin states in the respective operation modes which is under condition that PA7/SCOUT pin is specified as SCOUT output.

Table 3.3.3 SCOUT Pin States in the Operation Modes

Operation Mode / Output Clock	NORMAL, SLOW	HALT Mode	
		RUN, IDLE2, IDLE1	STOP
f_{FPH}	Outputs f_{FPH} clock.		Fixed to "0" or "1".
f_{SYS}	Outputs f_{SYS} clock.		

(2) CLK pin

CLK pin outputs f_{SYS} divided by 2 internal clock.

Outputs are specified by the clock output control register CKOCR<CLKEN>. Writing "1" sets clock output, and writing "0" sets high impedance. After reset, CKOCR<CLKEN> is depended on each product types. It is necessary to set for each usage. Table 3.3.4 shows the value and operation after reset.

During reset, CLK pin is internally pulled up regardless of the value of <CLKEN> register.

Table 3.3.4 <CLKEN> and CLK Pin Operation after Reset

Type Number	CKOCR<CLKEN>	CLK Pin Operation
TMP93CW46A	0	High impedance

Note: To set <CLKEN> = "0" and set CLK pin to high impedance, pull up externally to prevent through current which follows to the input buffer of CLK pin.

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 3.3.5 shows the watchdog timer mode registers.

Watchdog Timer Mode Register

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	Watchdog timer control 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}/ select clock frequency 1: 2^{16}/ select clock frequency	HALT mode selection 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Runaway detect internal reset control 1: Executes internal reset by runaway detect	STOP mode pin control 1: Drive pins in STOP mode

Pin state control in STOP mode

0	I/O off
1	Retains the state before 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 (A part I/O operating)

Warm-up time selection at returning from the stop mode (see Table 3.3.7)

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

Figure 3.3.5 Watchdog Timer Mode Register

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

1. RUN: Only the CPU halts
2. IDLE2: The built-in oscillator and the specified I/O operates.
3. IDLE1: Only the built-in oscillator operates, while all other built-in circuits stop.
4. STOP: All internal circuits including the built-in oscillator stop. This greatly reduces power consumption.

The operations in the halt state is described in Table 3.3.5.

Table 3.3.5 I/O Operation during HALT Mode

HALT Mode		RUN	IDLE2	IDLE1	STOP
WDMOD<HALTM1:0>		00	11	10	01
Block	CPU	Stop			
	I/O port	Keep the state when the "HALT" instruction was executed.			See Table 3.3.8
	8-bit timer	Operate		Stop	
	8-bit PWM timer				
	16-bit timer				
	Serial channel				
	AD converter				
	Watchdog timer				
	Interrupt controller				

(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 <IFF2:0> 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 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".

Note: Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0) which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2/RUN are not applicable to this case). (In this case, an interrupt request is kept on hold internally.)
If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

- 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	IDLE2	IDLE1	STOP	RUN	IDLE2	IDLE1	STOP
Halt releasing source	Interrupt	NMI	♦	♦	♦	♦ *1	–	–	–	–
		INTWDT	♦	×	×	×	–	–	–	–
		INT0	♦	♦	♦	♦ *1	○	○	○	○ *1
		INT4 to 7	♦	♦	×	×	×	×	×	×
		INTT0 to 3	♦	♦	×	×	×	×	×	×
		INTTR4 to 7	♦	♦	×	×	×	×	×	×
		INTRX0 to 4, TX0 to 4	♦	♦	×	×	×	×	×	×
		INTAD	♦	×	×	×	×	×	×	×
	RESET		♦	♦	♦	♦	♦	♦	♦	♦

♦: After releasing the HALT mode, CPU starts interrupt processing. (RESET initializes LSI.)

○: 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.

–: The priority level (Interrupt request level) of non-maskable interrupts is fixed to highest priority level “7”. There is not this combination type.

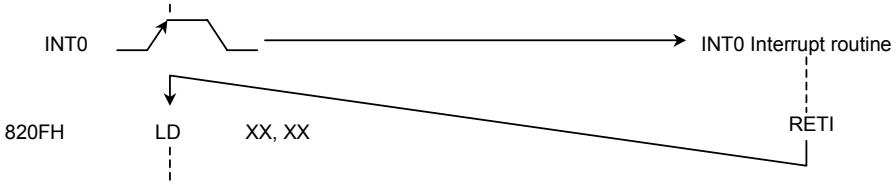
*1: Releasing the HALT mode is executed after passing the warm-up time.

Note: When releasing the HALT mode is executed by INT0 interrupt of the level mode in the interrupt enabled status, hold level “H” until starting interrupt processing. If level “L” is set before holding level “L”, interrupt processing is correctly started.

(Example releasing “RUN” mode)

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

```
Address
8203H   LD     (IIMC), 00H          ; Selects INT0 interrupt rising edge.
8206H   LD     (INTE0AD), 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.
```




```
820FH   LD     XX, XX
```

(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 "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 3.3.6 shows the interrupt timing for releasing the halt state by interrupts in the RUN/IDLE2 mode.

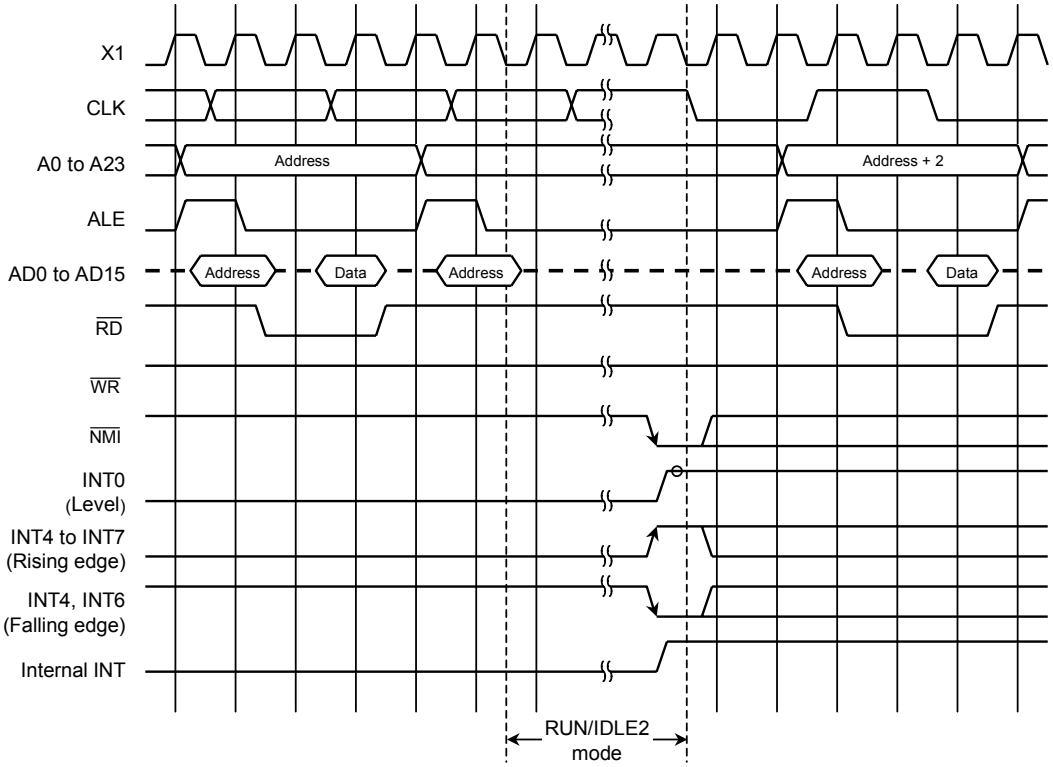


Figure 3.3.6 Timing Chart for Releasing the Halt State by Interrupt in RUN/IDLE2 Modes

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 INTWDT/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, the CLK pin is fixed at the level "H" in the output enable (CKOCR<CLKEN> = "1").

In the halt state, and interrupt request is sampled asynchronously with the system clock, however the halt release (Restart of operation) is performed synchronously with it.

IDLE1 mode is released by external interrupts (NMI, INT0). (See Table 3.3.6 "Halt Releasing Source and Halt Releasing Operation".)

When the IDLE1 mode is used, setting TRUN<PRRUN> to "0" to stop 9, 5-bit prescaler before "HALT" instruction reduces the power consumption.

Figure 3.3.7 illustrates the timing for releasing the halt state by interrupts in the IDLE1 mode.

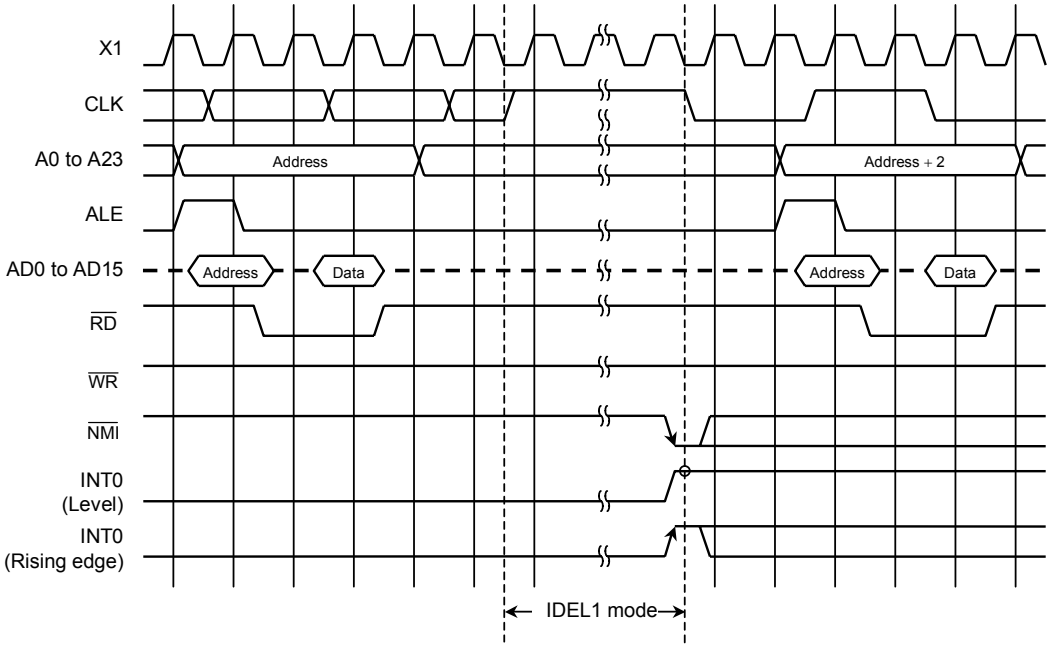


Figure 3.3.7 Timing Chart of Halt Released by Interrupts in IDLE1 Mode

4. STOP mode

The STOP mode is selected to stop all internal circuits including the internal oscillator. The pin status in the STOP mode is depended on setting the watchdog timer mode register WDMOD<DRVE>. (See Figure 3.3.5 for setting WDMOD<DRVE>.) Table 3.3.8 summarizes the state of these pins in the STOP mode.

The STOP mode is released by external interrupts (NMI, INT0). When the STOP mode is released, the system clock is started outputting after warm-up timer to get the stabilized oscillation. A warm-up time can be set using WDMOD<WARM>. See the example of warm-up time in Table 3.3.7.

In the system which supplies stable clock generated by an external oscillator, the warm-up time can be reduced using T45CR<QCU>.

Figure 3.3.8 illustrates the timing for releasing the halt state by interrupts in the STOP mode.

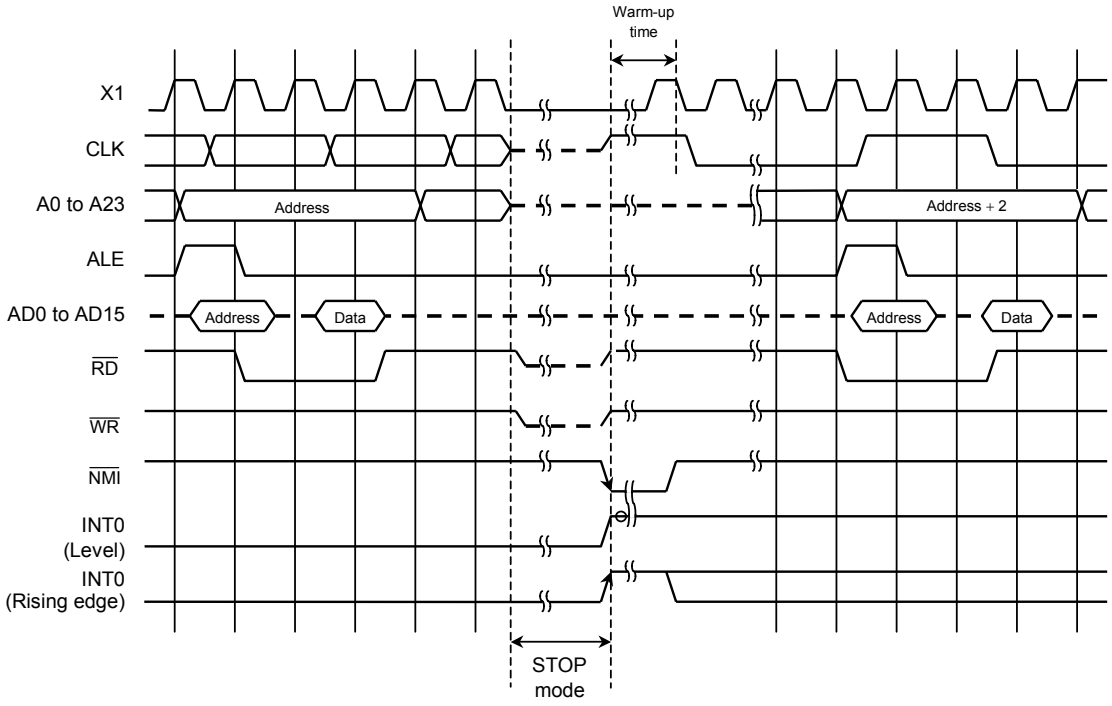


Figure 3.3.8 Timing Chart of Halt Released by Interrupt in STOP Mode

Table 3.3.7 The Example of Warm-up Time after Releasing the STOP Mode

Operation Clock after the STOP Mode	Warm-up Time [ms]		Clock
	WDMOD<WARM> = 0	WDMOD<WARM> = 1	
fc	0.8192	3.2768	fc = 20 MHz
fc/2	1.6384	6.5536	
fc/4	3.2768	13.1072	
fc/8	6.5536	26.2144	
fc/16	13.1072	52.4288	
fs	500	2000	fs = 32.768 kHz

How to calculate the warm-up time

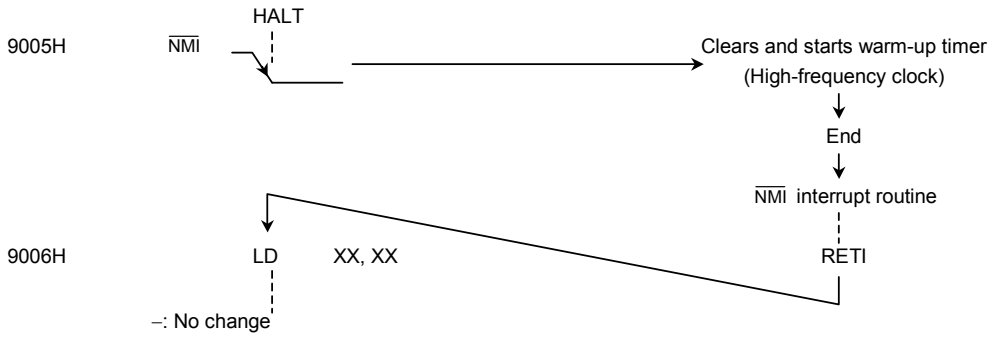
WDMOD<WARM> = "0": Operation clock after the 2^{14}/STOP mode

WDMOD<WARM> = "1": Operation clock after the 2^{16}/STOP mode

The NORMAL/SLOW mode selection is possible after releasing STOP mode. This is selected by SYSCR0<RSYSCK> register. Therefore, setting to <RSYSCK>, <RXEN>, <RXTEN> is necessary before "HALT" instruction is executed.

(Setting example) The STOP mode is entered when the low frequency operates, and high frequency operates after releasing due to NMI.

```
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 released.
```



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	TMP93CW46A	
		<DRVE> = 0	<DRVE> = 1
P00 to P07	Input mode Output mode AD8 to AD15	▲ – –	▲ Output –
P10 to P17	Input mode Output mode AD0 to AD7	▲ – –	▲ Output –
P20 to P27	Input mode Output mode A0 to A7/A16 to A23	▲ ▲	▲ Output
P30 ($\overline{RD}$), P31 ($\overline{WR}$)	Output pin	–	Output
P32 to P37	Input mode Output mode	PU* PU*	Input Output
P40, P41	Input mode Output mode	PU* PU*	Input Output
P42 ($\overline{CS2}$ / $\overline{CAS2}$)	Input mode Output mode	PD* PD*	Input Output
P5	Input mode	▲	▲
P6	Input mode Output mode	PU* PU*	Input Output
P7	Input mode Output mode	PU* PU*	Input Output
P80 to P86	Input mode Output mode	PU* PU*	Input Output
P87 (INT0)	Input mode Output mode Input mode (INT0)	PU PU Input	Input Output Input
P90 to P95	Input mode Output mode	PU* PU*	Input Output
PA0 to PA6	Input mode Output mode	– –	Input Output
PA7	Input mode Output mode, SCOUT	– –	Input Output
$\overline{NMI}$	Input pin	Input	Input
$\overline{WDTOUT}$	Output pin	Output	Output
ALE	Output (<ALEEN> = 1)	"L" level output	"L" level output
CLK	Output (<CLKEN> = 1)	–	"H" level output
$\overline{RESET}$	Input	Input	Input
$\overline{EA}$	Input	Input	Input
AM8/ $\overline{AM16}$	Input	Input	Input
X1	Input	–	–
X2	Output	"H" level output	"H" level output
P96	Input mode Output mode XT1	– – –	Input Output* –
P97	Input mode Output mode XT2	– – –	Input Output* –

–: Input for input mode/input pin is invalid; output mode/output pin is at high impedance.

Input: Input gate in operation. Fix input voltage to "L" or "H" so that input pin stays constant.

Output: Output state

Output*: Open-drain output state. Input gate in operation. Set output to "L" or attach pull-up pin so that the input gate stays constant.

PU: Programmable pull-up pin. Fix the pin to avoid through current since the input gate operates when a pull-up pin resistor is not set

PU*: Programmable pull-up pin. Input gate disable state. No through current even if the pin is set t high impedance.

PD*: Programmable pull-down pin. Input gate disable state. No through current even if the pin is set t high impedance.

▲: When HALT instruction is executed and CPU stops at the address of the port register, input gate in operation. Fix the pin to avoid through current and change the program. In the other cases, input for input mode is invalid, output mode is at high impedance.

×: Cannot set.

Note: Port registers are used for controlling programmable pull up/pull down. If a pin is also used for an output function (e.g., TO1) and the output function is specified, whether pull up or pull down is selected depends on the output function data. If a pin is also used for an input function, whether pull up or pull down is selected depends on the port register setting value only.

3.4 Interrupts

Interrupts are controlled by the CPU interrupt mask register SR<IFF2:0> and the built-in interrupt controller.

TMP93CW46A has altogether the following 35 interrupt sources:

- Interrupts from the CPU … 9
  (Software interrupts, and Illegal (Undefined) instruction execution)
- Interrupts from external pins ($\overline{\text{NMI}}$, INT0 and INT4 to INT7) … 6
- Interrupts from built-in I/Os … 20

A fixed individual interrupt vector number is assigned to each interrupt source; six levels of priority (Variable) 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 <IFF2:0>. If the value is greater than that the CPU interrupt mask register, the interrupt is accepted. However software interrupts and illegal instruction execution interrupts are not compared with <IFF2:0>, the interrupt is processed. The value in the CPU interrupt mask register <IFF2:0> can be changed using the EI instruction. Executing EI n changes the contents of <IFF2:0> 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 operates in the same way as the EI 7 instruction, setting <IFF2:0> = 7. Since the priority values for maskable interrupts are 0 to 6, the DI instruction is used to disable maskable interrupts to be accepted. The EI instruction becomes effective immediately after execution.

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 is a flowchart showing overall interrupt processing.

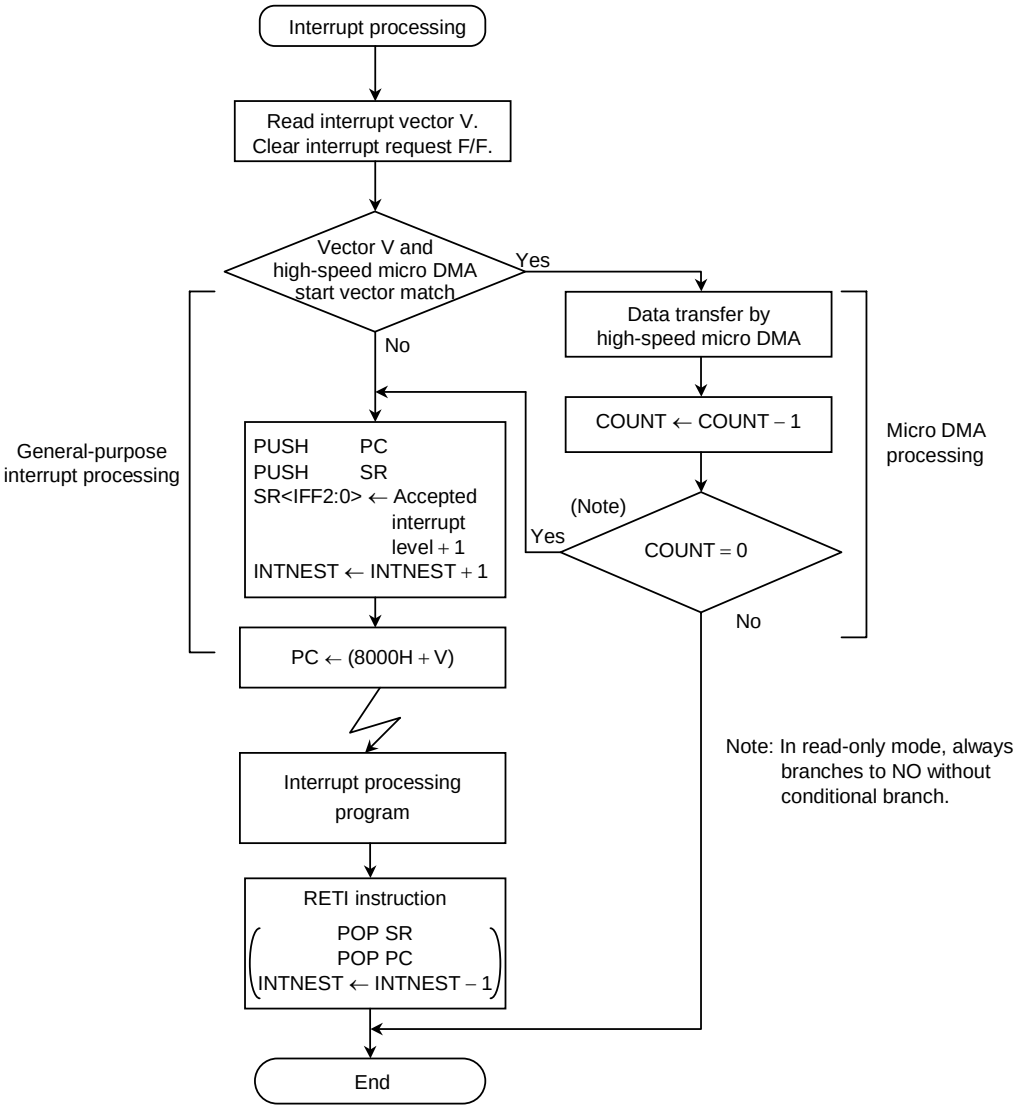


Figure 3.4.1 Interrupt Processing Flowchart

3.4.1 General-purpose Interrupt Processing

When accepting an interrupt, the CPU operates as follows. In the software interrupts or the illegal instruction execution interrupts from CPU, the following (1) and (3) are not executed.

(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, then clears the interrupt request.

The default priority is fixed as follows: The smaller the vector value, the higher the priority.

(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 <IFF2:0> 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 8000H + 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	
		MAX Mode	MIN Mode
8-bit	8 bits	35	31
	16 bits	31	27
16-bit	8 bits	29	27
	16 bits	25	23

To complete 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 <IFF2:0>. The CPU mask register <IFF2:0> 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.

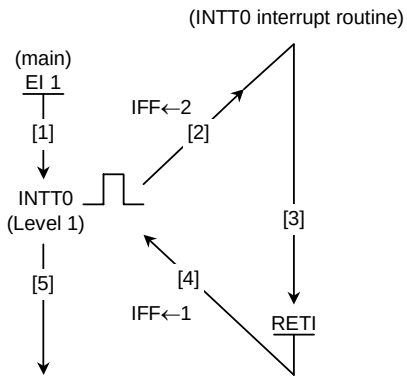
The interrupt request with a priority higher than the accepted now interrupt during the CPU is processing above (1) to (5) is accepted before the 1st instruction in the interrupt processing routine, causing interrupt processing to nest.

The CPU does not accept an interrupt request of the same level as that of the interrupt being processed. (Non-maskable interrupts can be accepted, causing interrupt processing to nest.)

Resetting initializes the CPU mask registers <IFF2:0> to 7; therefore, maskable interrupts are disabled.

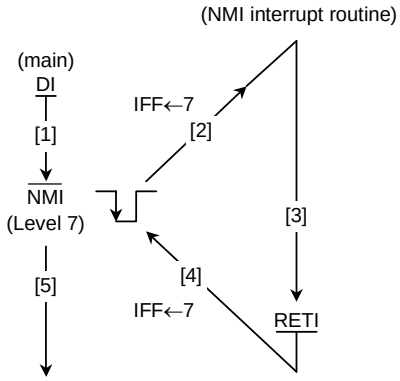
The following (1) to (5) show a flowchart of interrupt processing.

(1) Maskable interrupt



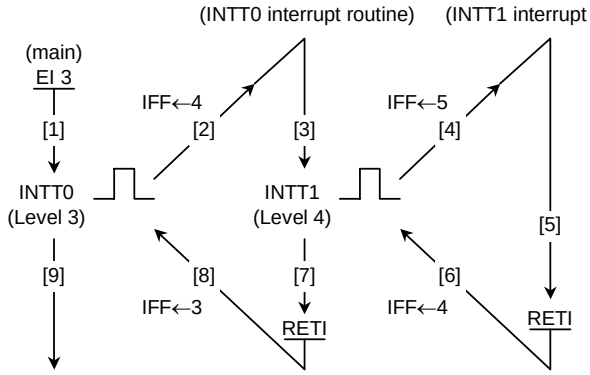
During execution of the main program, the CPU accepts an interrupt request. The CPU increments the IFF so that the interrupts of level 1 are not accepted during processing the interrupt routine.

(2) Non-maskable interrupt



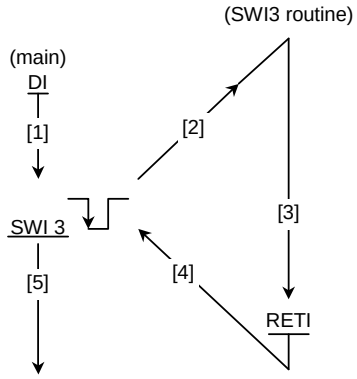
DI instruction is executed in the main program, so that the interrupts of only level 7 are accepted. The CPU does not increment the IFF even if the CPU accepts an interrupt request of level 7.

(3) Interrupt nesting



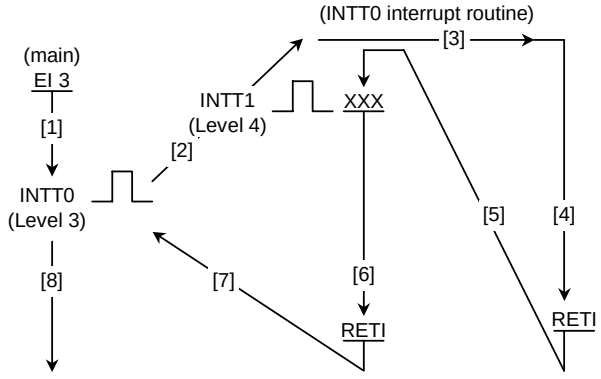
During processing the interrupts of level 3, the IFF is set to 4. When an interrupt with a level higher than level 4 is generated, the CPU accepts the interrupt with the higher level, causing interrupt processing to nest.

(4) Software interrupt



The CPU accepts the software interrupt request during DI status (IFF = 7) because of the level 7. The IFF is 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 addresses 008000H to 0080FFH (256 bytes) of the TMP93CW46A are assigned for interrupt vector area.

Table 3.4.1 TMP93CW46A Interrupt Table

Default Priority	Type	Interrupt Source	Vector Value "V"	Address Refer to Vector	Micro DMA Start Vector
1	Non-maskable	Reset, or SW10 instruction	0 0 0 0 H	8 0 0 0 H	–
2		SWI 1 instruction	0 0 0 4 H	8 0 0 4 H	–
3		Illegal instruction, or SWI2	0 0 0 8 H	8 0 0 8 H	–
4		SWI 3 instruction	0 0 0 C H	8 0 0 C H	–
5		SWI 4 instruction	0 0 1 0 H	8 0 1 0 H	–
6		SWI 5 instruction	0 0 1 4 H	8 0 1 4 H	–
7		SWI 6 instruction	0 0 1 8 H	8 0 1 8 H	–
8		SWI 7 instruction	0 0 1 C H	8 0 1 C H	–
9		NMI: $\overline{\text{NMI}}$ pin	0 0 2 0 H	8 0 2 0 H	08H
10		INTWD: Watchdog timer	0 0 2 4 H	8 0 2 4 H	09H
11	Maskable	INT0: INT0 pin	0 0 2 8 H	8 0 2 8 H	0AH
12		INT4: INT4 pin	0 0 2 C H	8 0 2 C H	0BH
13		INT5: INT5 pin	0 0 3 0 H	8 0 3 0 H	0CH
14		INT6: INT6 pin	0 0 3 4 H	8 0 3 4 H	0DH
15		INT7: INT7 pin	0 0 3 8 H	8 0 3 8 H	0EH
–		(Reserved)	0 0 3 C H	8 0 3 C H	–
16		INTT0: 8-bit timer0	0 0 4 0 H	8 0 4 0 H	10H
17		INTT1: 8-bit timer1	0 0 4 4 H	8 0 4 4 H	11H
18		INTT2: 8-bit timer2/PWM0	0 0 4 8 H	8 0 4 8 H	12H
19		INTT3: 8-bit timer3/PWM1	0 0 4 C H	8 0 4 C H	13H
20		INTTR4: 16-bit timer4 (TREG4)	0 0 5 0 H	8 0 5 0 H	14H
21		INTTR5: 16-bit timer4 (TREG5)	0 0 5 4 H	8 0 5 4 H	15H
22		INTTR6: 16-bit timer5 (TREG6)	0 0 5 8 H	8 0 5 8 H	16H
23		INTTR7: 16-bit timer5 (TREG7)	0 0 5 C H	8 0 5 C H	17H
24		INTRX0: Serial receive (Channel 0)	0 0 6 0 H	8 0 6 0 H	18H
25		INTTX0: Serial send (Channel 0)	0 0 6 4 H	8 0 6 4 H	19H
26		INTRX1: Serial receive (Channel 1)	0 0 6 8 H	8 0 6 8 H	1AH
27		INTTX1: Serial send (Channel 1)	0 0 6 C H	8 0 6 C H	1BH
28		INTAD: AD conversion completion	0 0 7 0 H	8 0 7 0 H	1CH
29		INTRX2: Serial receive (Channel 2)	0 0 7 4 H	8 0 7 4 H	1DH
30		INTTX2: Serial send (Channel 2)	0 0 7 8 H	8 0 7 8 H	1EH
31		INTRX3: Serial receive (Channel 3)	0 0 7 C H	8 0 7 C H	1FH
32		INTTX3: Serial send (Channel 3)	0 0 8 0 H	8 0 8 0 H	20H
33		INTRX4: Serial receive (Channel 4)	0 0 8 4 H	8 0 8 4 H	21H
34		INTTX4: Serial send (Channel 4)	0 0 8 8 H	8 0 8 8 H	22H
–		(Reserved)	0 0 8 C H	8 0 8 C H	–
to		to	to	to	to
–		(Reserved)	0 0 F C H	8 0 F C H	–

Setting to reset/interrupt vector

1. Reset vector

8000H	PC<7:0>
8001H	PC<15:8>
8002H	PC<23:16>
8003H	XX

2. Interrupt vector (except reset vector)

Address refer to vector		
	+0	PC<7:0>
	+1	PC<15:8>
	+2	PC<23:16>
	+3	XX

XX: Don't care

(Setting example)

Reset vector: 8100H, NMI Vector: 9ABCH, INTAD Vector: 123456H.

```
ORG     8000H
DL      008100H         ; Reset = 8100H

ORG     8020H
DL      009ABCH         ; NMI = 9ABCH

ORG     8070H
DL      123456H         ; INTAD = 123456H

ORG     8100H
LD      A, B
        ¦
ORG     9ABCH
LD      B, C
        ¦
ORG     123456H
LD      C, A
        ¦
        ¦
        ¦
```

(cf)

ORG, DL are the assembler directive.

- ORG: Control location counter
- DL: Define the long word (32 bits) data

3.4.2 Micro DMA

In addition to the conventional interrupt processing, the TMP93CW46A 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 micro DMA can process at very high speed compared with the TLCS-90 micro DMA 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 set in the interrupt controller. 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. In read-only mode, which is provided for DRAM refresh, the value in the counter is ignored and dummy read is repeated.

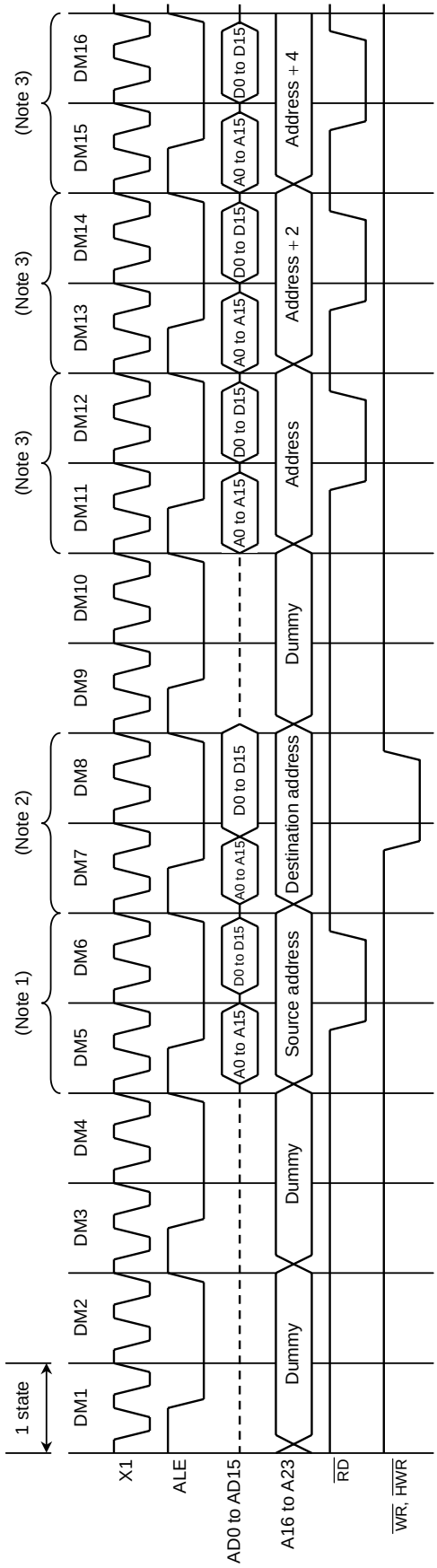
32-bit control registers are used for setting transfer source/destination addresses. However, the TMP93CW46A has only 24 address pins for output. A 16-Mbyte space is available for the micro DMA.

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.

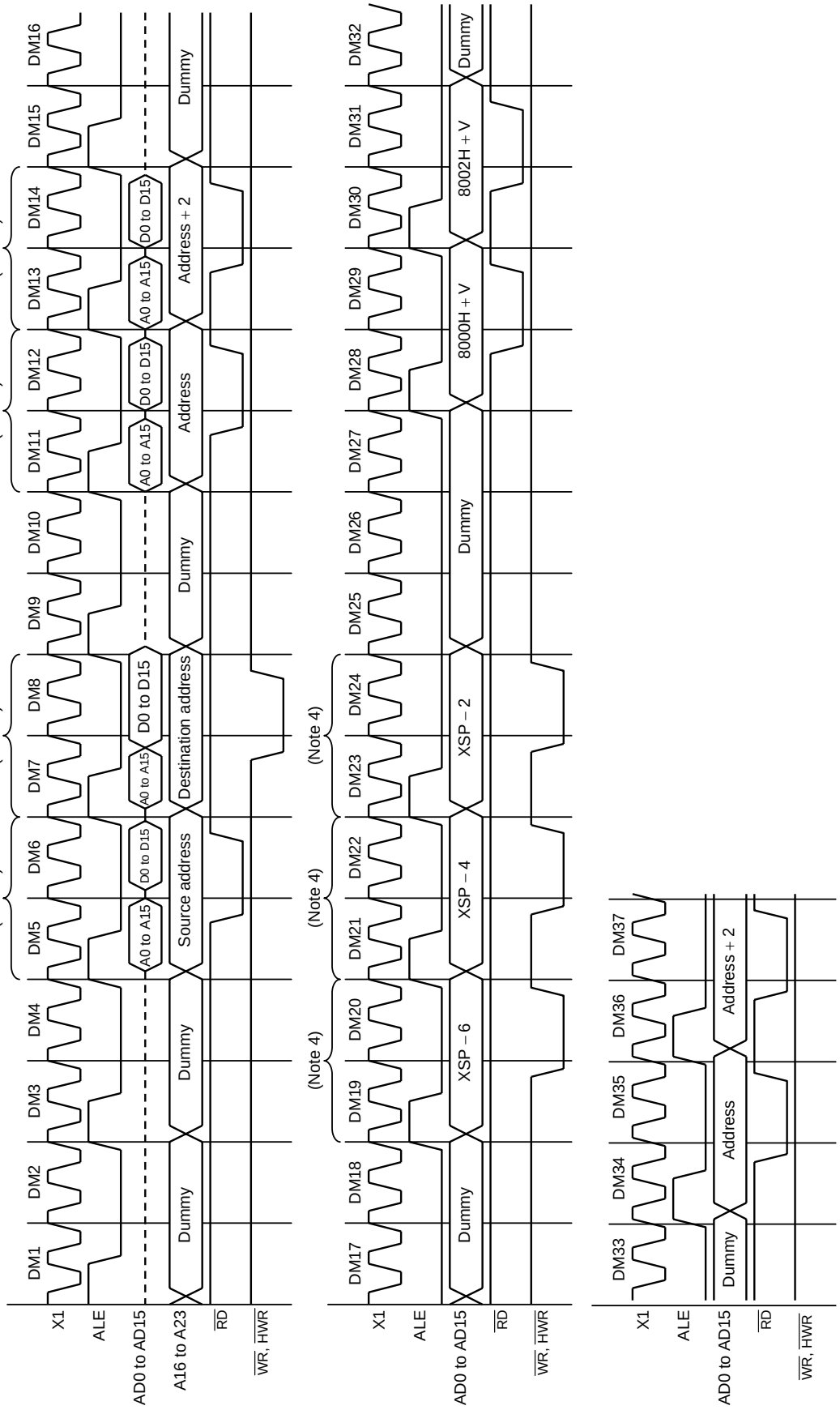
When the transfer counter is decremented to “0” after data is transferred with micro DMA, general-purpose interrupt processing is performed. After processing the general-purpose interrupt, starting the interrupts of the same channel restarts the transfer counter from 65536. If necessary, reset the transfer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

Interrupt sources processed by micro DMA processing are 26 sources with the micro DMA start vectors listed in Table 3.4.1.



Note 1: This is added 2 states the case of the bus width of source address area is 8 bits.
Note 2: This added 2 states the case of the bus width of destination address area is 8 bits.
Note 3: This may be a dummy cycle with instruction queue buffer.
Note 4: In the case of the word transfer mode.

Figure 3.4.2 Micro DMA Cycle (COUNT ≠ 0)

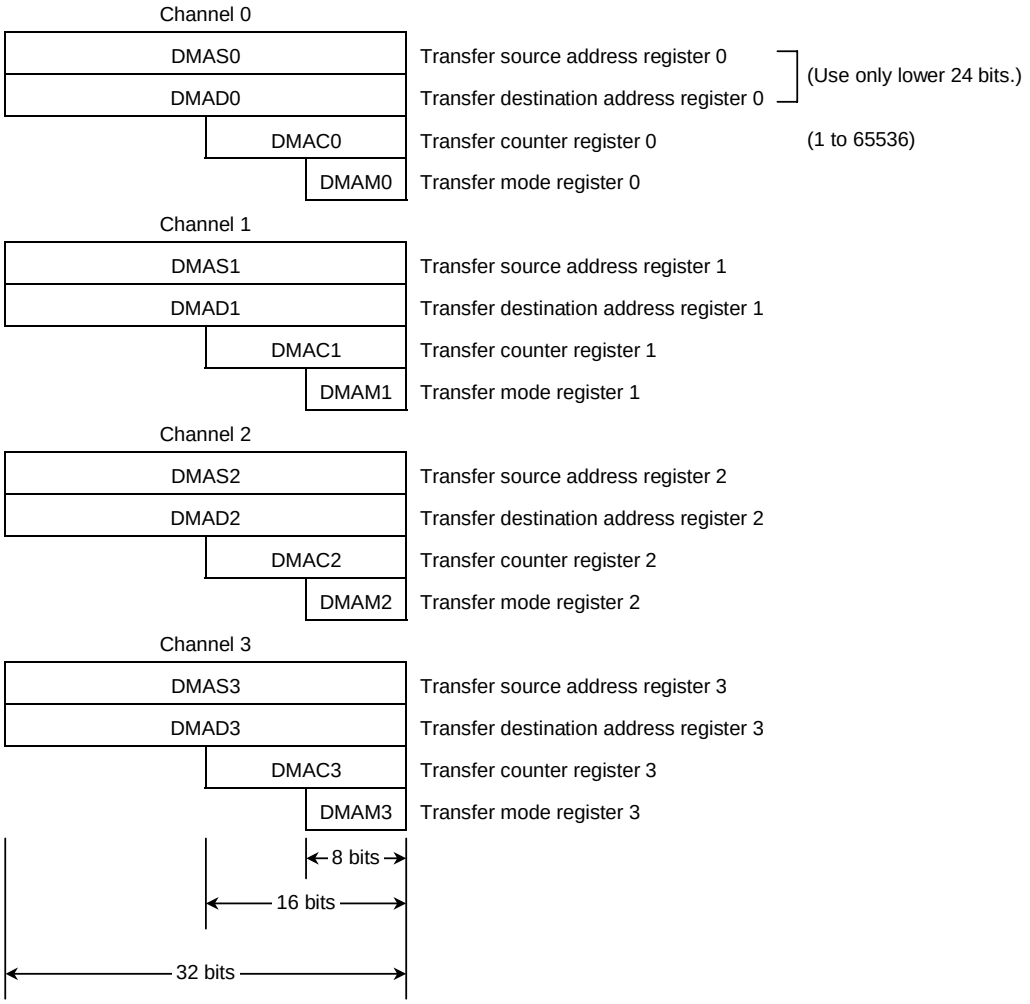


Note 1: This is added 2 states the case of the bus width of source address area is 8 bits.
Note 2: This added 2 states the case of the bus width of destination address area is 8 bits.
Note 3: This be a dummy cycle with instruction queue buffer.
Note 4: This is added 2 states the case of the bus width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

Figure 3.4.3 Micro DMA Cycle (COUNT=0)

Timing chart in the previous page is a micro DMA cycle of the transfer address increment mode. (The other mode except the read-only mode is the same as this.) (Condition: MAX mode, 16-bit bus width for 16 Mbytes, 0 waits)

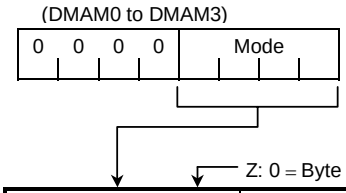
(2) Register configuration (CPU control register)



These control registers can be set only with “LDC cr, r” instruction.
(e.g.)

```
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: DMAM0 to DMAM3



					Execution time (Min) at 20 MHz
0	0	0	Z	Transfer destination address INC mode ..... I/O to memory (DMADn+) ← (DMASn) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	0	1	Z	Transfer destination address DEC mode ..... I/O to memory (DMADn−) ← (DMASn) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	0	Z	Transfer source address INC mode ..... memory to I/O (DMADn) ← (DMASn+) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	1	Z	Transfer source address DEC mode ..... memory to I/O (DMADn) ← (DMASn−) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	0	Z	Fixed address mode ..... I/O to I/O (DMADn) ← (DMASn) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	1	0	Read-only mode ..... for DRAM refresh Dummy ← (DMASn); Reads 4 bytes. DMASn ← DMASn + 4; Increments lower word only. DMACn ← DMACn − 1	14 states (1.4 μs)
1	0	1	1	Counter mode ..... for interrupt counter DMASn ← DMASn + 1 DMACn ← DMACn − 1 if DMACn = 0 then INT.	11 states (1.1 μs)

Note 1: 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.)

Note 2: Execution time: When setting source address/destination address area to 16-bit bus, 0 waits.

Clock condition: fc = 20 MHz, Clock gear: 1 (fc)

Note 3: Do not use the codes other than the above mentioned codes for transfer mode register.

<Example for usage of read only mode (DRAM refresh)>

* Clock condition
  System clock: High speed (fc)
  Clock gear: 1 (fc)

When the hardware configuration is as follows:

DRAM mapping size: 1 Mbyte

DRAM data bus size: 8 bits

DRAM mapping address range: 100000H to 1FFFFFH

Set the following registers first; refresh is performed automatically.

1. Register initial value setting

```
LD    XIX,   100000H
LDC   DMAS0, XIX      … Mapping start address
LD    A,     00001010B
LDC   DMAM0, A        … Read only mode (for DRAM refresh)
```

2. Timer setting

   Set the timers so that interrupts are generated at intervals of 62.5 μs or less.

3. Interrupt controller setting

   Set the timer interrupt mask higher than the other interrupts mask. Write the above timer interrupt vector value in the micro DMA start vector register, DMA0V.

   (Operation description)

   The DRAM data bus is an 8-bit bus and the micro DMA is in read-only mode (4 bytes), so refresh is performed for four times per interrupt.

   When a 512 refresh/8 ms DRAM is connected, DRAM refresh is performed sufficiently if the micro DMA is started every 15.625 μs × 4 = 62.5 μs or less, since the timing is 15.625 μs/refresh.

   (Overhead)

   Each processing time by the micro DMA is 1.8 μs (18 states) at 20 MHz with an 8-bit data bus.

   In the above example, the high-speed micro DMA is started every 62.5 μs, 1.8 μs/62.5 μs = 0.0288; thus, the overhead is 2.88%.

   (Note)

   When the bus is released which must wait to accept the interrupt, DRAM refresh is not performed because of the micro DMA is generated by an interrupt.

3.4.3 Interrupt Controller

Figure 3.4.4 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 flag, interrupt priority setting register, and a register for storing the micro DMA start vector. The interrupt request flag is used to latch interrupt requests from peripheral devices.

The flag is cleared to 0 at the following conditions.

- At reset
- When the CPU reads the interrupt vector after acceptance of interrupt
- When the CPU executes an instruction that clears the interrupt of that channel (Writes 0 in <IxxC> of the interrupt priority setting register).

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

INTE0AD ← – – – –0– – – Zero-clears the INT0 flip-flop.

The status of the interrupt request flag 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 (e.g., INTE0AD, INTE45, 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 ($\overline{\text{NMI}}$ pin, 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 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 <IFF2:0> 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<IFF2:0>. 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<IFF2:0>.

The interrupt controller also has four registers used to store the micro DMA start vector. These are I/O registers. 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 (e.g., DMAS and DMAD) prior to the micro DMA processing.

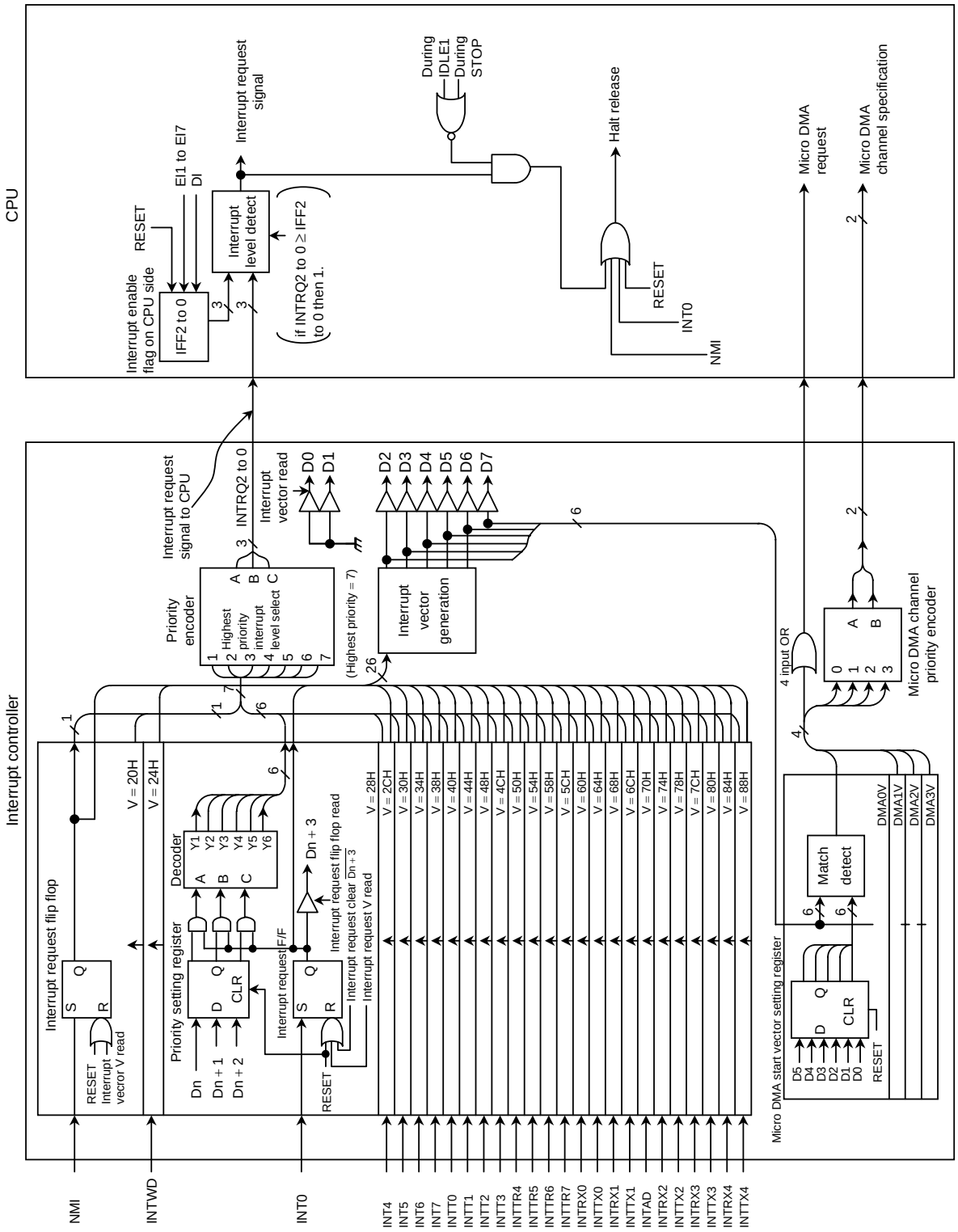


Figure 3.4.4 Block Diagram of Interrupt Controller

(1) Interrupt priority setting register

Symbol	Address	7	6	5	4	3	2	1	0	
INTE0AD	0070H	IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0	←Bit symbol
		R/W	W			R/W	W			←Read/Write
		0	0	0	0	0	0	0	0	←After reset
		INTAD				INT0				←Interrupt source
INTE45	0071H	I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INT5				INT4				
INTE67	0072H	I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INT7				INT6				
INTET10	0073H	IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INTT1 (Timer 1)				INTT0 (Timer 0)				
INTEPW10	0074H	IPW1C	IPW1M2	IPW1M1	IPW1M0	IPW0C	IPW0M2	IPW0M1	IPW0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INTT3 (Timer 3/PWM1)				INTT2 (Timer 2/PWM0)				
INTET54	0075H	IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INTTR5 (TREG5)				INTTR4 (TREG4)				
INTET76	0076H	IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INTTR7 (TREG7)				INTTR6 (TREG6)				
INTES0	0077H	ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INTTX0				INTRX0				
INTES1	0078H	ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INTTX1				INTRX1				
INTES2	0059H	ITX2C	ITX2M2	ITX2M1	ITX2M0	IRX2C	IRX2M2	IRX2M1	IRX2M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INTTX2				INTRX2				
INTES3	005AH	ITX3C	ITX3M2	ITX3M1	ITX3M0	IRX3C	IRX3M2	IRX3M1	IRX3M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INTTX3				INTRX3				
INTES4	005BH	ITX4C	ITX4M2	ITX4M1	ITX4M0	IRX4C	IRX4M2	IRX4M1	IRX4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
		INTTX4				INTRX4				

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Prohibits interrupt request.
0	0	1	Sets interrupt request level to “1”.
0	1	0	Sets interrupt request level to “2”.
0	1	1	Sets interrupt request level to “3”.
1	0	0	Sets interrupt request level to “4”.
1	0	1	Sets interrupt request level to “5”.
1	1	0	Sets interrupt request level to “6”.
1	1	1	Prohibits 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 INTAD, INTRX0 to INTRX4 are not cleared by writing “0” to IxxC because of they are level interrupts. They can be cleared only by resetting or reading the conversion value or the receive buffer.

(2) External interrupt control

Interrupt Input Mode Control Register

		7	6	5	4	3	2	1	0
IIMC (007BH)	Bit symbol						I0IE	I0LE	NMIREE
	Read/Write						W	W	W
	After reset						0	0	0
	Function						1: INT0 input enable	0: INT0 edge mode 1: INT0 level mode	1: Can be accepted in $\overline{\text{NMI}}$ rising edge.

INT0 input enable (Note 1)

0	INT0 disable (P87 function only)
1	Input enable

$\overline{\text{NMI}}$ rising edge enable

0	Interrupt request generation at falling edge
1	Interrupt request generation at rising/falling edge

INT0 level enable (Note 2)

0	Rising edge detect interrupt
1	High level interrupt

Note 1: The INT0 pin can also be used for standby release as described later. Even if the pin is not used for standby release, setting this register to "0" maintains the port function during standby mode.

Note 2: Case of changing from level to edge for INT0 pin mode execution example:

```
LD (INTE0AD), XXXX0000B  ; INT0 disable, clean the request flag
LD (IIMC), XXXXX10XB     ; Change from level to edge
LD (INTE0AD), XXXX0nnnB  ; Set interrupt level "n" for INT0, clear the request flag
```

Note 3: Read-modify-write is prohibited.

Note 4: IIMC<Bit7:3> is always read as "1".

Note 5: See electrical characteristics in section 4 for external interrupt input pulse.

Figure 3.4.5 Interrupt Input Mode Control Register

Table 3.4.2 Setting of External Interrupt Pin Functions

Interrupt	Pin Name	Mode		Setting Method
$\overline{\text{NMI}}$	–	↓	Falling edge	IIMC<NMIREE> = 0
		↓↑	Falling and rising edges	IIMC<NMIREE> = 1
INT0	P87	↑	Rising edge	IIMC<I0LE> = 0, <I0IE> = 1
		H	High level	IIMC<I0LE> = 1, <I0IE> = 1
INT4	P80	↑	Rising edge	T4MOD<CAP12M1:0> = 0, 0 or 0, 1 or 1, 1
		↓	Falling edge	T4MOD<CAP12M1:0> = 1, 0
INT5	P81	↑	Rising edge	–
INT6	P84	↑	Rising edge	T5MOD<CAP34M1:0> = 0, 0 or 0, 1 or 1, 1
		↓	Falling edge	T5MOD<CAP34M1:0> = 1, 0
INT7	P85	↑	Rising edge	–

(3) Micro DMA start vector

When the CPU reads the interrupt vector after accepting an interrupt, it simultaneously compares the bits 2 to 7 of the interrupt vector with each channel's micro DMA start vector. When both match, the interrupt is processed in micro DMA mode for the channel whose value matched.

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

Micro DMA 0 Start Vector

		7	6	5	4	3	2	1	0
DMA0V	Bit symbol			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
(007CH)	Read/Write			W					
	After reset			0	0	0	0	0	0
	Function	Micro DMA channel 0 processed by matching bits 2 to 7 of the interrupt vector.							

Micro DMA 1 Start Vector

		7	6	5	4	3	2	1	0
DMA1V	Bit symbol			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
(007DH)	Read/Write			W					
	After reset			0	0	0	0	0	0
	Function	Micro DMA channel 1 processed by matching bits 2 to 7 of the interrupt vector.							

Micro DMA 2 Start Vector

		7	6	5	4	3	2	1	0
DMA2V	Bit symbol			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
(007EH)	Read/Write			W					
	After reset			0	0	0	0	0	0
	Function	Micro DMA channel 2 processed by matching bits 2 to 7 of the interrupt vector.							

Micro DMA 3 Start Vector

		7	6	5	4	3	2	1	0
DMA3V	Bit symbol			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
(007FH)	Read/Write			W					
	After reset			0	0	0	0	0	0
	Function	Micro DMA channel 3 processed by matching bits 2 to 7 of the interrupt vector.							

Note: Read-modify-write is not possible for DMA0V to DMA3V.

Figure 3.4.6 Micro DMA Start Vector Register

(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.

To avoid the above occurring, clear the interrupt request flag by entering the instruction to clear the flag after the DI instruction. In the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing instruction and following more than one instruction are executed. When EI instruction is placed immediately after clearing instruction, an interrupt becomes enable before interrupt request flags are cleared.

In the case of changing the value of the interrupt mask register <IFF2:0> by execution of POR SR instruction, disable an interrupt by DI instruction before execution of POP SR instruction.

3.5 Functions of Ports

The TMP93CW46A has 79 bits for I/O 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 3.5.1 lists the function of each port pin. Table 3.5.2 lists I/O registers and specification.

Table 3.5.1 Functions of Ports

(R: ↑ = with programmable pull-up resistor
↓ = with programmable pull-down resistor)

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port 0	P00 to P07	8	I/O	–	Bit	AD0 to AD7
Port 1	P10 to P17	8	I/O	–	Bit	AD8 to AD15/A8 to A15
Port 2	P20 to P27	8	I/O	↓	Bit	A0 to A7/A16 to A23
Port 3	P30	1	Output	–	(Fixed)	$\overline{RD}$
	P31	1	Output	–	(Fixed)	$\overline{WR}$
	P32	1	I/O	↑	Bit	$\overline{HWR}$
	P33	1	I/O	↑	Bit	$\overline{WAIT}$
	P34	1	I/O	↑	Bit	$\overline{BUSRQ}$
	P35	1	I/O	↑	Bit	$\overline{BUSAK}$
	P36	1	I/O	↑	Bit	R/ $\overline{W}$
	P37	1	I/O	↑	Bit	$\overline{RAS}$
Port 4	P40	1	I/O	↑	Bit	$\overline{CS0}$ / $\overline{CAS0}$
	P41	1	I/O	↑	Bit	$\overline{CS1}$ / $\overline{CAS1}$
	P42	1	I/O	↓	Bit	$\overline{CS2}$ / $\overline{CAS2}$
Port 5	P50 to P57	8	Input	–	(Fixed)	AN0 to AN7
Port 6	P60	1	I/O	↑	Bit	TXD2
	P61	1	I/O	↑	Bit	RXD2
	P62	1	I/O	↑	Bit	$\overline{CTS2}$ /SCLK2
	P63	1	I/O	↑	Bit	TXD3
	P64	1	I/O	↑	Bit	RXD3
	P65	1	I/O	↑	Bit	$\overline{CTS3}$ /SCLK3
	P66	1	I/O	↑	Bit	TXD4
	P67	1	I/O	↑	Bit	RXD4
Port 7	P70	1	I/O	↑	Bit	TI0
	P71	1	I/O	↑	Bit	TO1
	P72	1	I/O	↑	Bit	TO2
	P73	1	I/O	↑	Bit	TO3
Port 8	P80	1	I/O	↑	Bit	TI4/INT4
	P81	1	I/O	↑	Bit	TI5/INT5
	P82	1	I/O	↑	Bit	TO4
	P83	1	I/O	↑	Bit	TO5
	P84	1	I/O	↑	Bit	TI6/INT6
	P85	1	I/O	↑	Bit	TI7/INT7
	P86	1	I/O	↑	Bit	TO6
	P87	1	I/O	↑	Bit	INT0
Port 9	P90	1	I/O	↑	Bit	TXD0
	P91	1	I/O	↑	Bit	RXD0
	P92	1	I/O	↑	Bit	$\overline{CTS0}$ /SCLK0
	P93	1	I/O	↑	Bit	TXD1
	P94	1	I/O	↑	Bit	RXD1
	P95	1	I/O	↑	Bit	SCLK1
	P96	1	I/O	–	Bit	XT1
	P97	1	I/O	–	Bit	XT2
Port A	PA0 to PA6	7	I/O	–	Bit	
	PA7	1	I/O	–	Bit	SCOUT

Table 3.5.2 I/O Registers and Specification (1/2)

Port	Name	Specification	I/O Register		
			Pn	PnCR	PnFC
Port 0	P00 to P07	Input port	×	0	None
		Output port	×	1	
		AD0 to AD7 bus	×	×	
Port 1	P10 to P17	Input port	×	0	0
		Output port	×	1	0
		AD8 to AD15 bus	×	0	1
		A8 to A15 output	×	1	0
Port 2	P20 to P27	Input port (without PD)	1	0	0
		Input port (with PD)	0	0	0
		Output port	×	1	0
		A0 to A7 output	1	0	1
		A16 to A23 output	1	1	0
Port 3	P30	Output port	×	None	0
		Outputs $\overline{RD}$ only when accessing external space	1		1
		Always outputs $\overline{RD}$	0		1
	P31	Output port	×	None	0
		Outputs $\overline{WR}$ only when accessing external space	×		1
	P32 to P37	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P32	$\overline{HWR}$ output	×	1	1
	P33	$\overline{WAIT}$ input (without PU)	0	0	None
		$\overline{WAIT}$ input (with PU)	1	0	
	P34	$\overline{BUSRQ}$ input (without PU)	0	0	1
		$\overline{BUSRQ}$ input (with PU)	1	0	1
	P35	$\overline{BUSAK}$ output	×	1	1
	P36	R/$\overline{W}$ output	×	1	1
	P37	$\overline{RAS}$ output	×	1	1
Port 4	P40 to P41	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P42	Input port (without PD)	1	0	0
		Input port (with PD)	0	0	0
		Output port	×	1	0
	P40	$\overline{CS0}$ output (Note 1)	×	1	1
	P41	$\overline{CS1}$ output (Note 1)	×	1	1
	P42	$\overline{CS2}$ output (Note 1)	×	1	1
Port 5	P50 to P57	Input port	×	None	
		AN0 to AN7 input (Note 2)	×		
Port 6	P60 to P67	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P60	TXD2 output	×	1	1
	P63	TXD3 output	×	1	1
	P66	TXD4 output	×	1	1
	P61	RXD2 input (without PU)	0	0	None
		RXD2 input (with PU)	1	0	
	P64	RXD3 input (without PU)	0	0	None
		RXD3 input (with PU)	1	0	

X: Don't care

Note 1: CS/WAIT control registers BnCH<BnCAS> selects the wave form output from P40 to P42 pin, $\overline{CS0}$ to $\overline{CS2}$ or $\overline{CAS0}$ to $\overline{CAS2}$.

Note 2: The channel for AD converter input is selected by ADMOD2<ADCHn>.

Table 3.5.3 I/O Registers and Specification (2/2)

Port	Name	Specification	I/O Register		
			Pn	PnCR	PnFC
Port 6	P67	RXD4 input (without PU)	0	0	None
		RXD4 input (with PU)	1	0	
	P62	SCLK2 output	×	1	1
		$\overline{CTS2}$ /SCLK2 input (without PU)	0	0	0
		$\overline{CTS2}$ /SCLK2 input (with PU)	1	0	0
	P65	SCLK3 output	×	1	1
		$\overline{CTS3}$ /SCLK3 input (without PU)	0	0	0
		$\overline{CTS3}$ /SCLK3 input (with PU)	1	0	0
Port 7	P70 to P73	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P70	TI0 input (without PU)	0	0	None
		TI0 input (with PU)	1	0	
	P71	TO1 output port	×	1	1
	P72	TO2 output port	×	1	1
	P73	TO3 output port	×	1	1
Port 8	P80 to P87	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P80	TI4/INT4 input (without PU)	0	0	None
		TI4/INT4 input (with PU)	1	0	
	P81	TI5/INT5 input (without PU)	0	0	None
		TI5/INT5 input (with PU)	1	0	
	P84	TI6/INT6 input (without PU)	0	0	None
		TI6/INT6 input (with PU)	1	0	
	P85	TI7/INT7 input (without PU)	0	0	None
		TI7/INT7 input (with PU)	1	0	
	P82	TO4 output	×	1	1
	P83	TO5 output	×	1	1
	P86	TO6 output	×	1	1
	P87 (Note 3)	INT0 input (without PU)	0	0	None
		INT0 input (with PU)	1	0	
Port 9	P90 to P95	Input port (without PU)	0	0	0
		Input port (with PU)	1	0	0
		Output port	×	1	0
	P90	TXD0 output	×	1	1
	P93	TXD1 output	×	1	1
	P91	RXD0 input (without PU)	0	0	None
		RXD0 input (with PU)	1	0	
	P94	RXD1 input (without PU)	0	0	None
		RXD1 input (with PU)	1	0	
	P92	SCLK0 output	×	1	1
		$\overline{CTS0}$ /SCLK0 input (without PU)	0	0	0
		$\overline{CTS0}$ /SCLK0 input (with PU)	1	0	0
	P95	SCLK1 output	×	1	1
		SCLK1 input (without PU)	0	0	0
		SCLK1 input (with PU)	1	0	0
	P96 to P97	Input port	×	0	None
		Output port (Note 4)	×	1	
		XT1/2 (Note 5)	×	0	
Port A	PA0 to PA7	Input port	×	0	None
		Output port	×	1	
	PA7	SCOUT output port (Note 6)	×	1	

Note 3: Using P87 pin as INT0, IIMC register has to be set enable interrupt.

Note 4: Using P96/P97 as output port, output is through the open-drain buffer.

Note 5: Using P96/P97 as XT1 to XT2, SYSCR0 register has to be set enable oscillation.

Note 6: Using PA7 as SCOUT, PAFC register has to be written suitable value.

Resetting makes the port pins listed below function as general-purpose I/O ports.

I/O pins programmable for input or output are set to input ports except P96/XT1, P97/XT2.

To set port pins for built-in functions, a program is required.

Note about the Bus Release and programmable pull-up/pull-down I/O ports.

When the bus is released ($\overline{BUSAK}$ = "0"), the output buffer of AD0 to AD15, A0 to A23, control signal ($\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$, $\overline{RAS}$, $\overline{CS0}$/$\overline{CAS0}$ to $\overline{CS2}$/$\overline{CAS2}$) is off and their state become high impedance.

However, the output of built-in programmable pull-up/pull-down resistors are kept before the bus is released. These programmable pull-up/pull-down resistors can be selected ON/OFF by programmable when they are used as the input ports.

The case of they are used as the output ports, they can not be selected ON/OFF by programmable.

Table 3.5.4 shows the pin state when the bus is released.

Table 3.5.4 The Pin State (when the Bus is Released)

Pin Name	The Pin State (when the Bus is Released)	
	Used as the Port	Used as the Function
P00 to P07 (AD0 to AD7) P10 to P17 (AD8 to AD15/A8 to A15)	The state is not changed. (Do not become to high impedance (High-Z).)	Become high impedance (High-Z).
P30 ($\overline{RD}$) P31 ($\overline{WR}$)	Becomes high impedance (High-Z).	Becomes high impedance (High-Z).
P32 ($\overline{HWR}$) P37 ($\overline{RAS}$)	The output buffer is OFF. The programmable pull-up resistor is ON the case of only the output latch is equal to "1".	The output buffer is OFF. The programmable pull-up resistor is ON irrespective of the output latch.
P36 (R/$\overline{W}$) P40 ($\overline{CS0}$/$\overline{CAS0}$) P41 ($\overline{CS1}$/$\overline{CAS1}$)	The output buffer is OFF. The programmable pull-up resistor is ON the case of only the output latch is equal to "1".	The output buffer is OFF. There is a possibility that the programmable pull-up resistor is ON or OFF due to the bus releasing timing irrespective of the output latch.
P42 ($\overline{CS2}$/$\overline{CAS2}$)	The output buffer is OFF. The programmable pull-down resistor is ON the case of only the output latch is equal to "0".	The output buffer is OFF. There is a possibility that the programmable pull-down resistor is ON or OFF due to the bus releasing timing irrespective of the output latch.
P20 to P27 (A16 to A23)	The state is not changed. (Do not become to high impedance (High-Z).)	The output buffer is OFF. The programmable pull-down resistor is ON the case of only the output latch is equal to "0".

Figure 3.5.1 shows the example of the external interface circuit the case of the bus releasing function is used.

When the bus is released, both internal memory and internal I/O can not be accessed. But the internal I/O continues to operate.

So, the watchdog timer also continues to run. Therefore, be careful about bus releasing time and setting the detection time of the WDT.

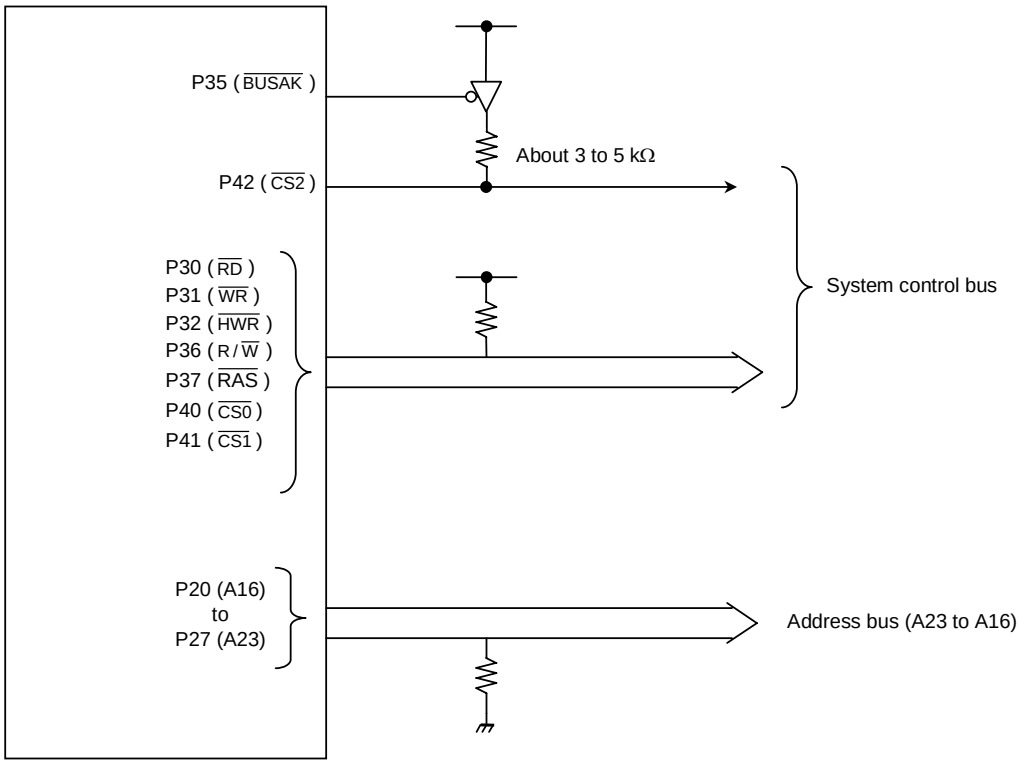


Figure 3.5.1 Example of the Interface Circuit (The case of using bus releasing function)

The above circuit is necessary to fix the signal level in the case of the bus is released.

Resetting sets P30 ($\overline{RD}$), P31 ($\overline{WR}$) to output, P40 ($\overline{CS0}$), P41 ($\overline{CS1}$), P32 ($\overline{HWR}$), P36 (R/$\overline{W}$), P37 ($\overline{RAS}$), and P35 ($\overline{BUSAK}$) to input with pull-up resistor, P42 ($\overline{CS2}$) and P20 to P27 (A16 to A23) to input with pull-down resistor.

The above circuit is necessary to fix the signal level after reset because of the external pull-up resistor collisions with the internal pull-down resistor.

The value of this external pull-up resistor must be 3 to 5 kΩ. (The value of the internal pull-down resistor is about 50 to 150 kΩ.) P20 to P27 (A16 to A23) also needs circuit like P42 ($\overline{CS2}$) to fix the signal level.

But for the P20 to P27 (A16 to A23) which does not have means "L" is active, add pull-down directly like above circuit.

3.5.1 Port 0 (P00 to P07)

Port 0 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using the control register P0CR. Resetting resets all bits of P0CR to 0 and sets port 0 to input mode.

In addition to functioning as a general-purpose I/O port, port 0 also functions as an address data bus (AD0 to AD7). To access external memory, port 0 functions as an address data bus (AD0 to AD7) and all bits of the control register P0CR are cleared to 0.

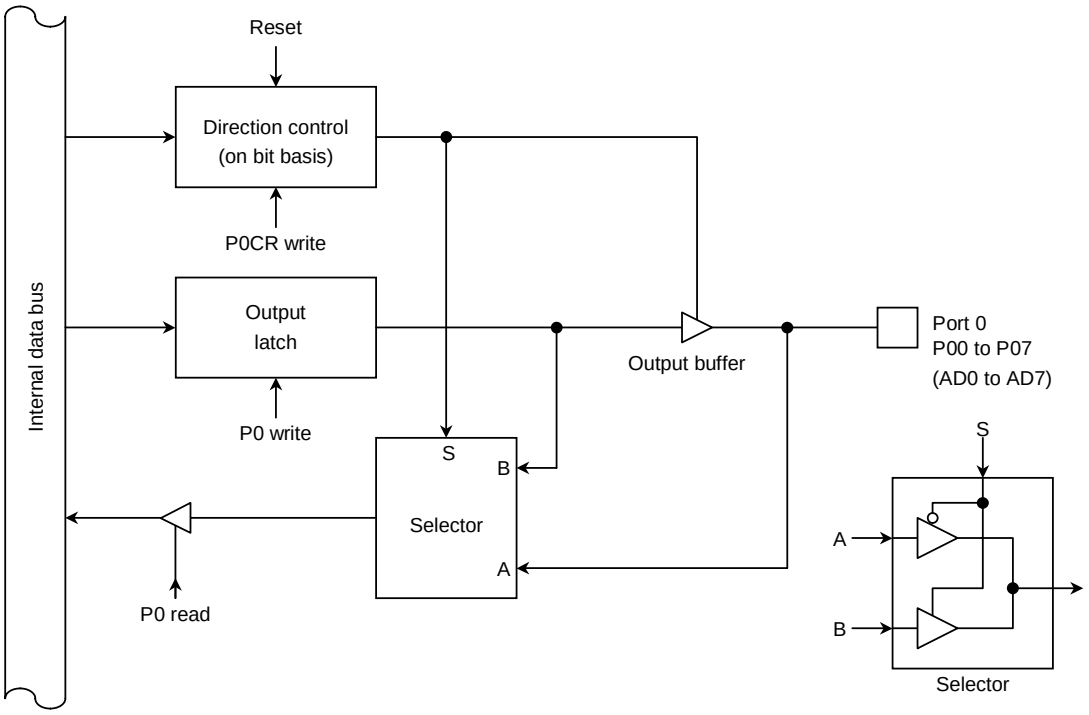


Figure 3.5.2 Port 0

3.5.2 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using control register P1CR and function register P1FC. Reset all bits of output latch P1, control register P1CR, and function register P1FC to 0 and sets port 1 to input mode.

In addition to functioning as a general-purpose I/O port, port 1 also functions as an address data bus (AD8 to AD15) or an address bus (A8 to A15).

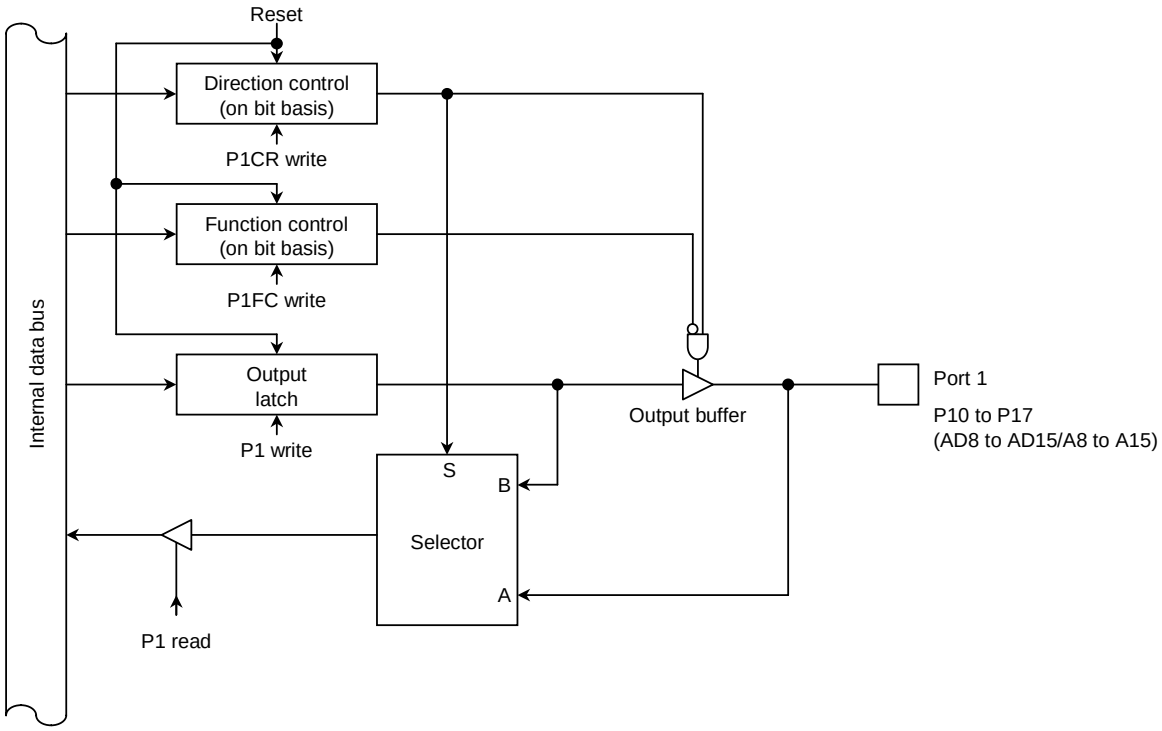


Figure 3.5.3 Port 1

Port 0 Register

		7	6	5	4	3	2	1	0
P0	Bit symbol	P07	P06	P05	P04	P03	P02	P01	P00
(0000H)	Read/Write	R/W							
	After reset	Input mode (Output latch register becomes undefined.)							

Port 0 Control Register

		7	6	5	4	3	2	1	0
P0CR	Bit symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
(0002H)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output (At external access, Port 0 becomes AD7 to AD0 and P0CR is cleared to 0.)							

→ Port 0 I/O setting

0	Input
1	Output

Port 1 Register

		7	6	5	4	3	2	1	0
P1	Bit symbol	P17	P16	P15	P14	P13	P12	P11	P10
(0001H)	Read/Write	R/W							
	After reset	Input mode (Output latch register is cleared to “0”.)							

Port 1 Control Register

		7	6	5	4	3	2	1	0
P1CR	Bit symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
(0004H)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	<<See P1FC below.>>							

Port 1 Function Register

		7	6	5	4	3	2	1	0
P1FC	Bit symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
(0005H)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	P1FC/P1CR = 00: Input 01: Output 10: AD15 to AD8 11: A15 to A8							

Read-modify-write is prohibited for registers P0CR, P1CR, and P1FC.

→ Port 1 function setting

P1CR <P1XC> \ P1FC<P1XF>	0	1
0	Input port	Address data bus (AD15 to AD8)
1	Output port	Address bus (A15 to A8)

Note: <P1XF> is bit X in register P1FC; <P1XC>, in register P1CR.

Figure 3.5.4 Registers for Ports 0 and 1

3.5.3 Port 2 (P20 to P27)

Port 2 is an 8-bit general-purpose I/O port. I/O can be set on bit basis using the control register P2CR and function register P2FC. Resetting resets all bits of output latch P2, control register P2CR and function register P2FC to 0. It also sets port 2 to input mode and connects a pull-down resistor.

In addition to functioning as a general-purpose I/O port, port 2 also functions as an address data bus (A0 to A7) and an address bus (A16 to A23). Using port 2 as address bus (A0 to A7 or A16 to A23), write "1" to output latches and be off the programmable pull-down resistors.

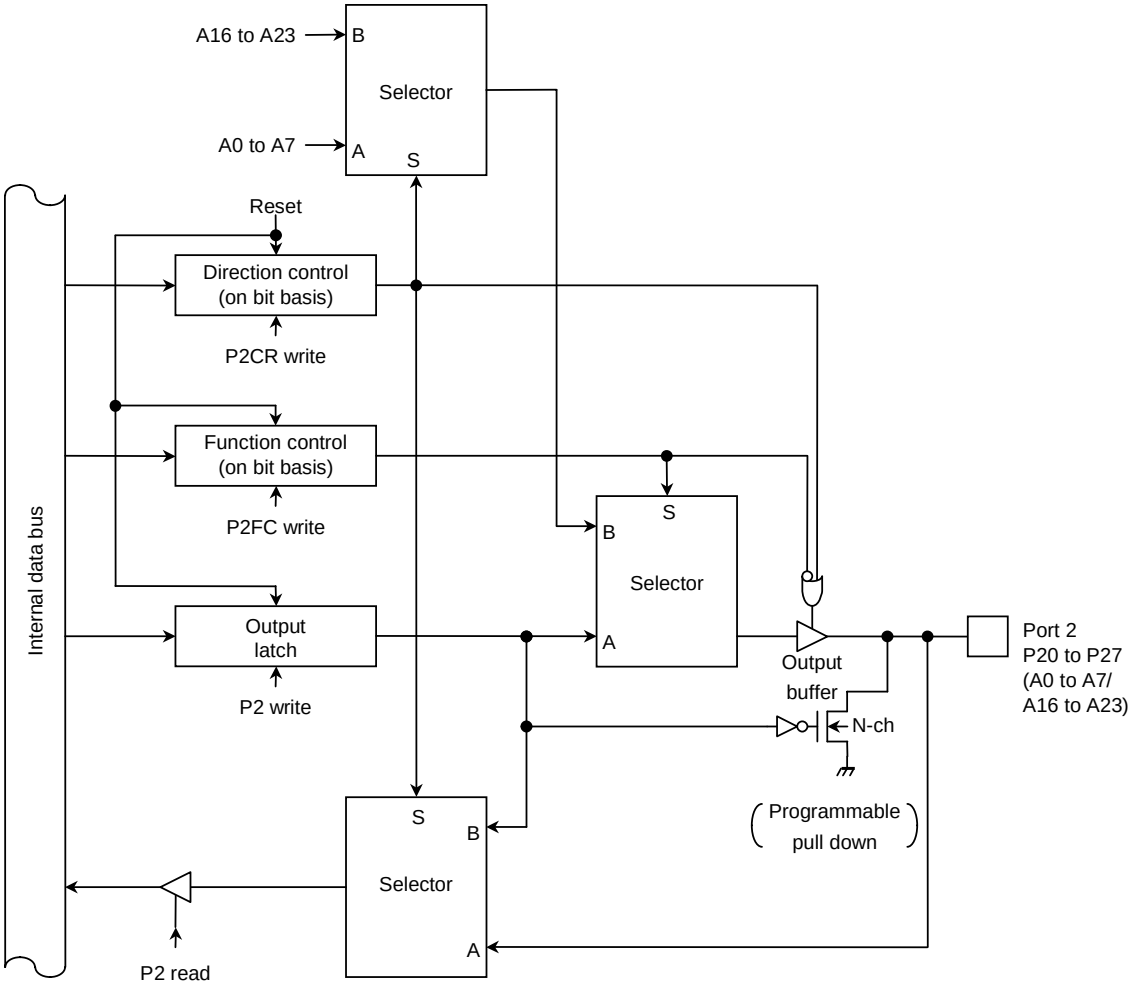


Figure 3.5.5 Port 2

Port 2 Register

		7	6	5	4	3	2	1	0
P2	Bit symbol	P27	P26	P25	P24	P23	P22	P21	P20
(0006H)	Read/Write	R/W							
	After reset	Input mode (Output latch register is cleared to “0”.)							

Port 2 Control Register

		7	6	5	4	3	2	1	0
P2CR	Bit symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
(0008H)	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	<<See P2FC below.>>							

Port 2 Function Register

		7	6	5	4	3	2	1	0
P2FC	Bit symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
(0009H)	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	P2FC/P2CR = 00: Input 01: Output 10: A7 to A0 11: A23 to A16							

→ Port 2 function setting

P2CR <P2XC> \ P2FC<P2XF>	0	1
0	Input port	Address data bus (A7 to A0)
1	Output port	Address bus (A23 to A16)

Note: <P2XF> is bit X in register P2FC; <P2XC>; in register P2CR. To set as an address bus A23 to A16, set P2FC after setting P2CR.

Note 1: Read-modify-write is prohibited for registers P2CR and P2FC.

Note 2: When port P2 is used in the input mode, P2 register controls the built-in pull-down resistor. Read-modify-write is prohibited in the input mode or the I/O mode.
Setting the built-in pull-down resistor may be depended on the states of the input pin.

Figure 3.5.6 Registers for Port 2

3.5.4 Port 3 (P30 to P37)

Port 3 is an 8-bit general-purpose I/O port.

I/O can be set on a bit basis, but note that P30 and P31 are used for output only. I/O is set using control register P3CR and function register P3FC. Resetting resets all bits of output latch P3, control register P3CR (Bits 0 and 1 are unused), and function register P3FC to 0. Resetting also outputs 1 from P30 and P31, sets P32 to P37 to input mode, and connects a pull-up resistor.

In addition to functioning as a general-purpose I/O port, port 3 also functions as an I/O for the CPU's control/status signal.

When P30 pin is defined as $\overline{RD}$ signal output mode (<P30F> = 1), clearing the output latch register <P30> to 0 outputs the $\overline{RD}$ strobe (Used for the pseudo static RAM) from the P30 pin even when the internal address area is accessed.

If the output latch register <P30> remains 1, the $\overline{RD}$ strobe signal is output only when the external address area is accessed.

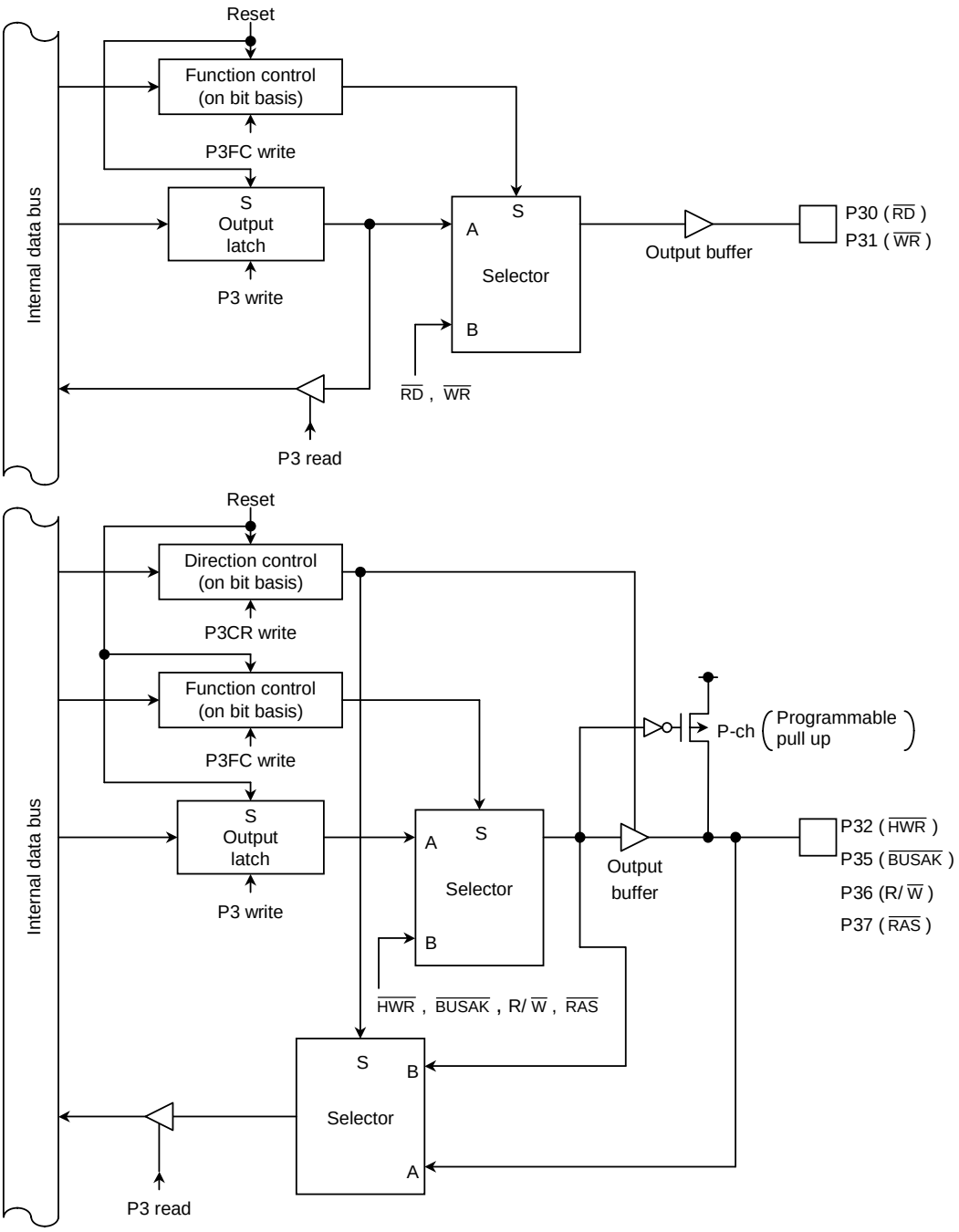


Figure 3.5.7 Port 3 (P30, P31, P32, P35, P36, P37)

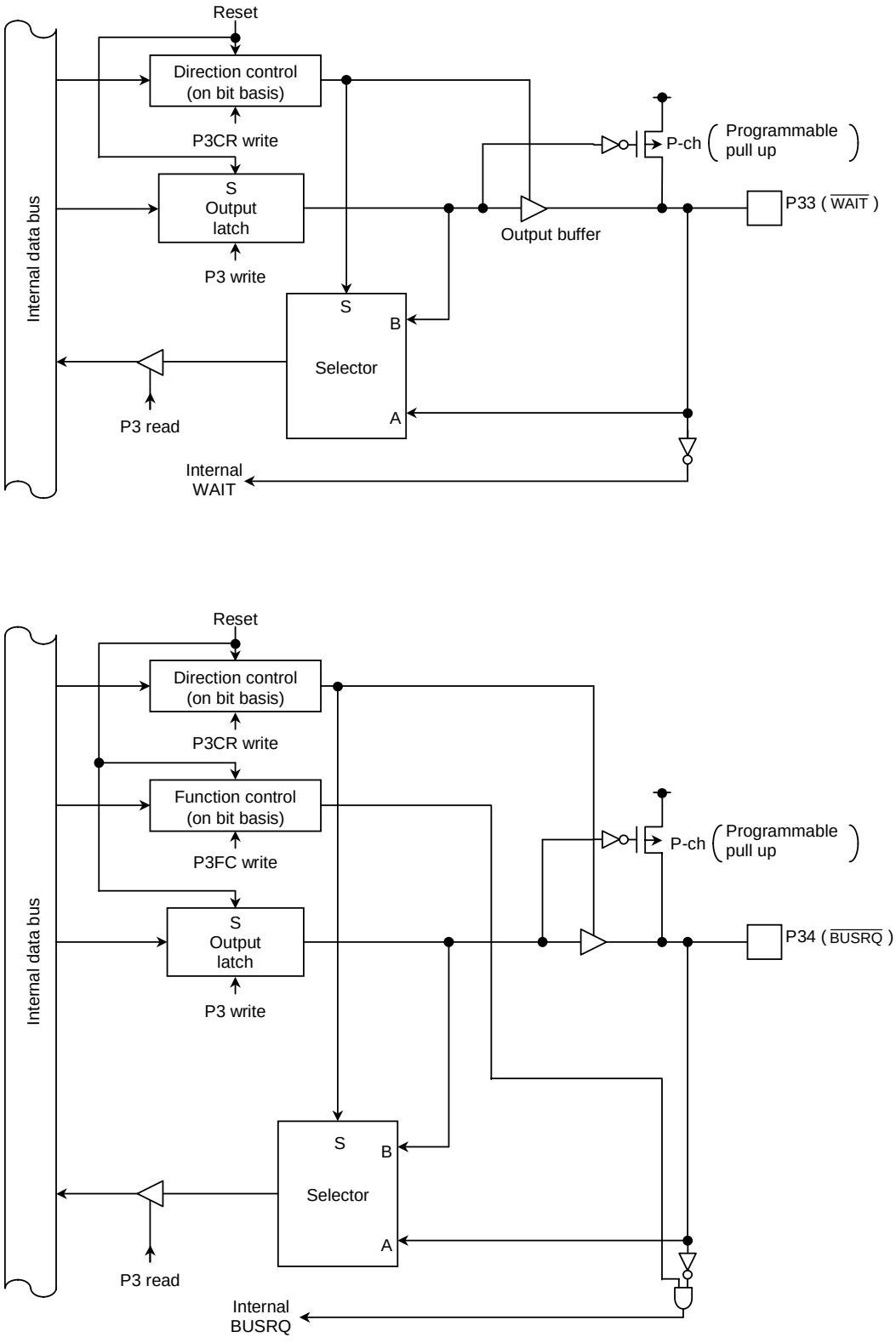


Figure 3.5.8 Port3 (P33, P34)

Port 3 Register

		7	6	5	4	3	2	1	0
P3	Bit symbol	P37	P36	P35	P34	P33	P32	P31	P30
(0007H)	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
		Input mode (pulled up)						Output mode	

Port 3 Control Register

		7	6	5	4	3	2	1	0
P3CR	Bit symbol	P37C	P36C	P35C	P34C	P33C	P32C		
(000AH)	Read/Write	W							
	After reset	0	0	0	0	0	0		
		0: Input 1: Output							

Output mode → I/O setting

0	Input
1	Output

Port 3 Function Register

		7	6	5	4	3	2	1	0
P3FC	Bit symbol	P37F	P36F	P35F	P34F		P32F	P31F	P30F
(000BH)	Read/Write	W							
	After reset	0	0	0	0		0	0	0
	Function	0: Port 1: $\overline{RAS}$	0: Port 1: R/$\overline{W}$	0: Port 1: $\overline{BUSAK}$	0: Port 1: $\overline{BUSRQ}$		0: Port 1: $\overline{HWR}$	0: Port 1: $\overline{WR}$	0: Port 1: $\overline{RD}$

$\overline{BUSRQ}$ setting

P3FC<P34F>	1
P3CR<P34C>	0

$\overline{BUSAK}$ setting

P3FC<P35F>	1
P3CR<P35C>	1

R/$\overline{W}$ setting

P3FC<P36F>	1
P3CR<P36C>	1

$\overline{RAS}$ setting

P3FC<P37F>	1
P3CR<P37C>	1

P30 ($\overline{RD}$) function setting

<P30F> \ <P30>	0	1
0	"0" output	"1" output
1	Always $\overline{RD}$ output (for pseudo SRAM)	$\overline{RD}$ output only for external access

P31 ($\overline{WR}$) function setting

<P31F> \ <P31>	0	1
0	"0" output	"1" output
1	$\overline{WR}$ output only for external access	

$\overline{HWR}$ setting

P3FC<P32F>	1
P3CR<P32C>	1

Note 1: Read-modify-write is prohibited for registers P3CR and P3FC.

Note 2: When port P3 is used in the input mode, P3 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Note 3: When P33/ $\overline{WAIT}$ pin is used as a $\overline{WAIT}$ pin, set P3CR<P33C> to "0" and chip select/wait control register<BnW1:0> to "10".

Figure 3.5.9 Registers for Port 3

3.5.5 Port 4 (P40 to P42)

Port 4 is a 3-bit general-purpose I/O port. I/O can be set on a bit basis using control register P4CR and function register P4FC. Resetting does the following:

- Sets the P40 and P42 output latch registers to 1.
- Resets all bits of the P42 output latch register, the control register P4CR, and the function register P4FC to 0.
- Sets P40 and P41 to input mode and connects a pull-up resistor.
- Sets P42 to input mode and connects a pull-down resistor.

In addition to functioning as a general-purpose I/O port, port 4 also functions as a chip select output signal ($\overline{CS0}$ to $\overline{CS2}$ or $\overline{CAS0}$ to $\overline{CAS2}$).

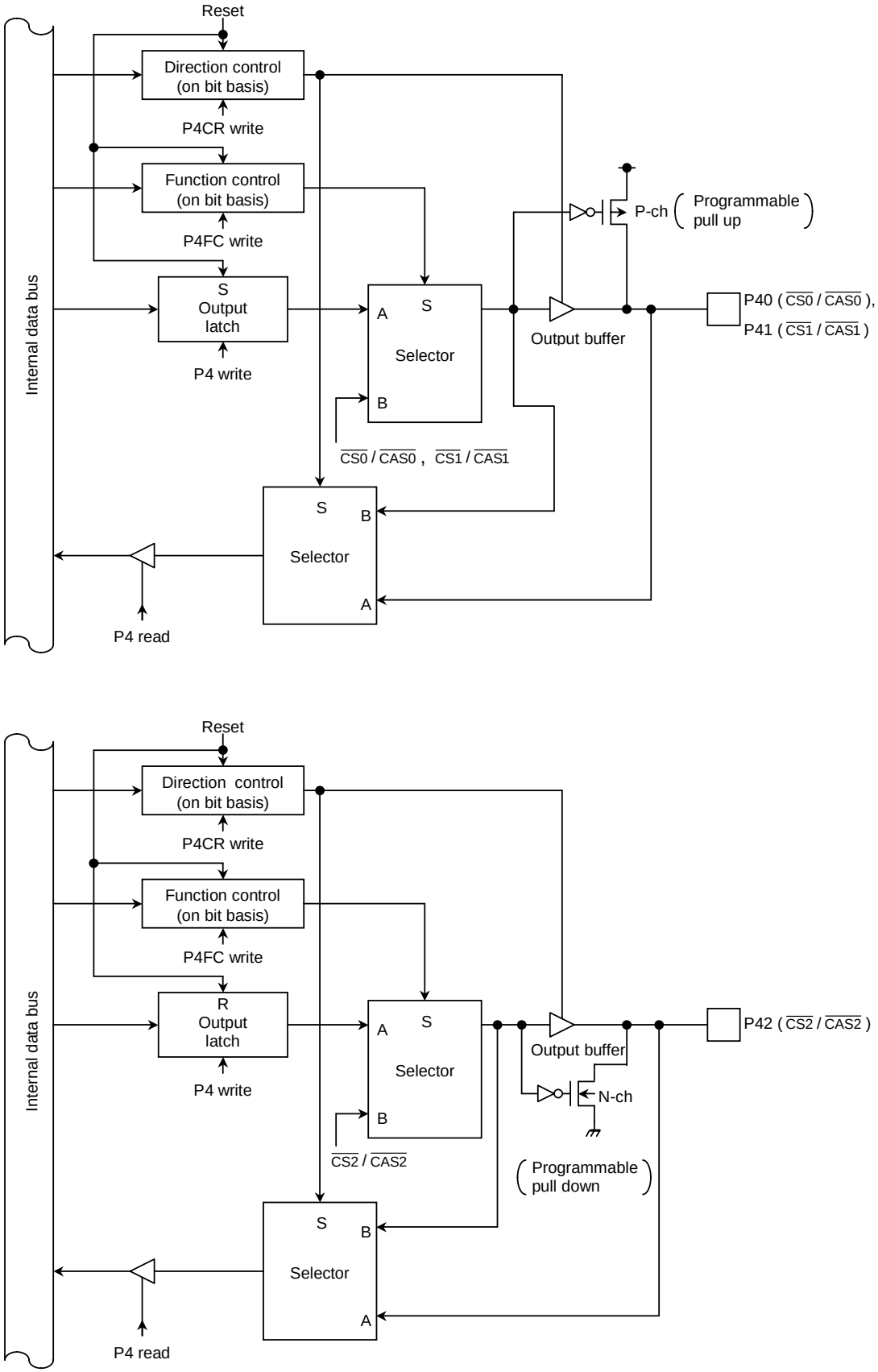


Figure 3.5.10 Port 4

Port 4 Register

		7	6	5	4	3	2	1	0
P4	Bit symbol						P42	P41	P40
(000CH)	Read/Write						R/W		
	After reset						0 (Pull down)	1 (Pull up)	1 (Pull up)
	Function						Input mode		

Port 4 Control Register

		7	6	5	4	3	2	1	0
P4CR	Bit symbol						P42C	P41C	P40C
(000EH)	Read/Write						W		
	After reset						0	0	0
	Function						0: Input 1: Output		

I/O setting

0	Input
1	Output

Port 4 Function Register

		7	6	5	4	3	2	1	0
P4FC	Bit symbol						P42F	P41F	P40F
(0010H)	Read/Write						W		
	After reset						0	0	0
	Function						0: Port	1: $\overline{CS}$ / $\overline{CAS}$	

0	Port (P40)
1	$\overline{CS0}$ / $\overline{CAS0}$

0	Port (P41)
1	$\overline{CS1}$ / $\overline{CAS1}$

0	Port (P42)
1	$\overline{CS2}$ / $\overline{CAS2}$

Note 1: Read-modify-write is prohibited for registers P4CR and P4FC.

Note 2: When port P4 is used in the input mode, P4 register controls the built-in pull-up/pull-down resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up/pull-down resistor may be depended on the states of the input pin.

Note 3: To output chip select signal ($\overline{CS0}$ / $\overline{CAS0}$ to $\overline{CS2}$ / $\overline{CAS2}$), set the corresponding bits of the control register P4CR and the function register P4FC to "1".

Chip select/wait controller (B0CS, B1CS, B2CS) registers select the function of $\overline{CS}$ / $\overline{CAS}$.

Note 4: P4<Bit7:3> is always read as "1".

Figure 3.5.11 Registers for Port 4

3.5.6 Port 5 (P50 to P57)

Port 5 is an 8-bit input port, also used as an analog input pin for the internal AD Converter.

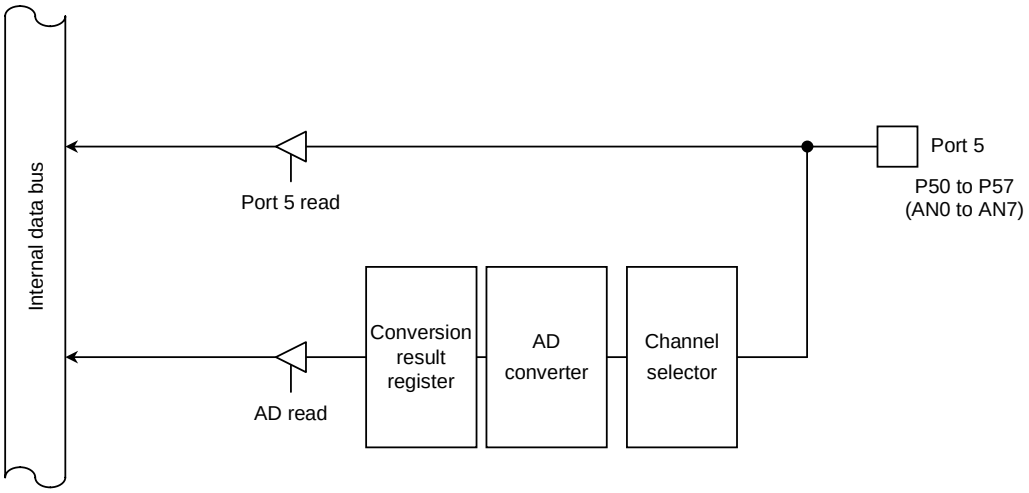


Figure 3.5.12 Port 5

Port 5 Register

		7	6	5	4	3	2	1	0
P5	Bit symbol	P57	P56	P55	P54	P53	P52	P51	P50
(000DH)	Read/Write	R							
	After reset	Input mode							

Note: The input channel selection of AD converter is set by AD converter mode register ADMOD2.

Figure 3.5.13 Register for Port 5

3.5.7 Port 6 (P60 to P67)

Port 60 to 67 are 8-bit general-purpose I/O ports. I/O can be set on bit basis. Resetting sets port 6 as an input port and connects a pull-up resistor. It also sets all bits of the output latch to 1. In addition to functioning as a general-purpose I/O port, port 60 to 67 also function as serial channels 2, 3, 4 I/O functions. Writing 1 in the corresponding bit of the port 6 function register (P6FC) enables the respective functions.

Resetting resets the function register P6FC to 0, and sets all bits to ports.

(1) Port 60, 63, 66 (TXD2, TXD3, TXD4)

In addition to functioning as a general-purpose I/O port, port 60, 63, 66 also function as serial channel TXD output pins.

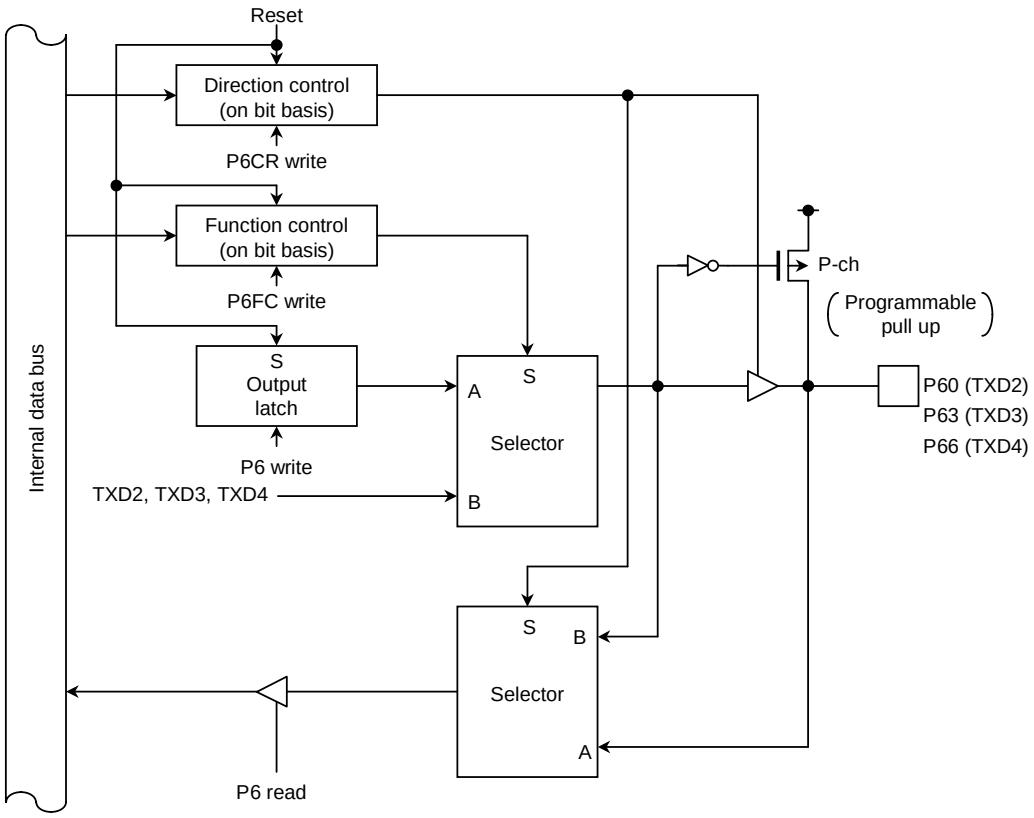


Figure 3.5.14 Port 60, 63, 66

(2) Port 61, 64, 67 (RXD2, RXD3, RXD4)

In addition to functioning as a general-purpose I/O port, port 61, 64, 67 also function as serial channel RXD input pins.

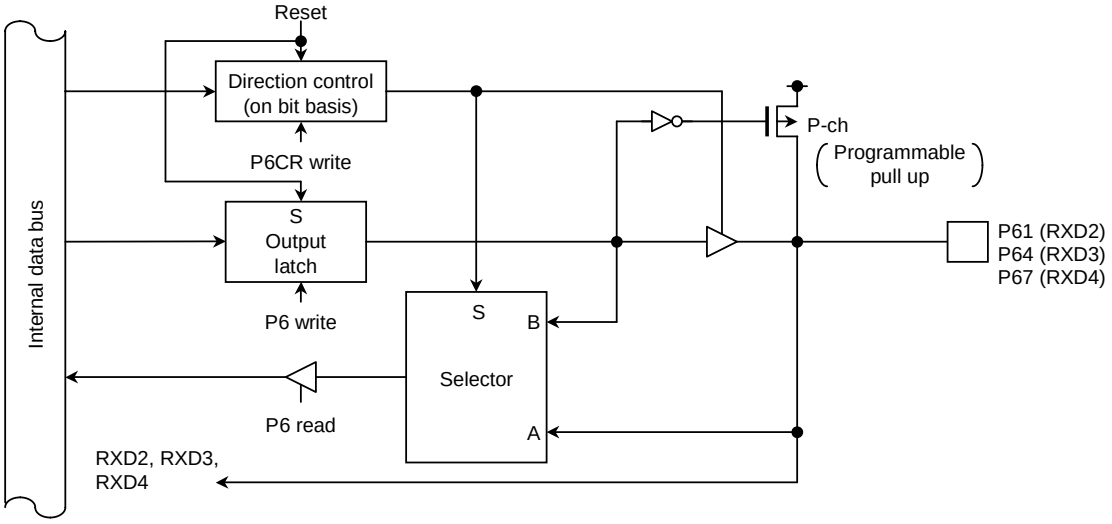


Figure 3.5.15 Port 61, 64, 67

(3) Port 62, 65 ($\overline{\text{CTS2}}$/SCLK2, $\overline{\text{CTS3}}$/SCLK3)

In addition to functioning as a general-purpose I/O port, port 62, 65 also function as serial channel 2, 3 $\overline{\text{CTS}}$ input pins or SCLK input/output pins.

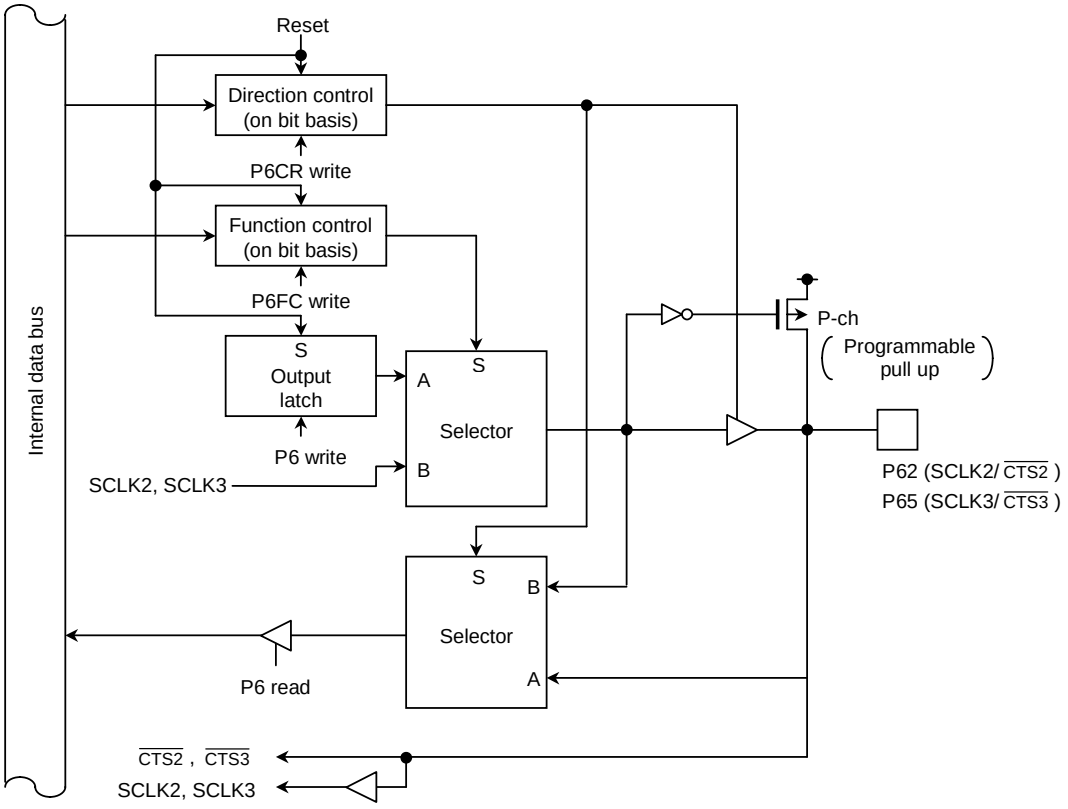


Figure 3.5.16 Port 62, 65

Port 6 Register

		7	6	5	4	3	2	1	0
P6	Bit symbol	P67	P66	P65	P64	P63	P62	P61	P60
(0012H)	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
		Input mode (Pull up)							

Port 6 Control Register

		7	6	5	4	3	2	1	0
P6CR	Bit symbol	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
(0014H)	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

Port 6 I/O setting

0	Input
1	Output

Port 6 Function Register

		7	6	5	4	3	2	1	0
P6FC	Bit symbol		P66F	P65F		P63F	P62F		P60F
(0016H)	Read/Write		W			W			W
	After reset		0	0		0	0		0
	Function		0: Port 1: TXD4	0: Port 1: SCLK3		0: Port 1: TXD3	0: Port 1: SCLK2		0: Port 1: TXD2

P60 TXD2 output setting

P6FC<P60F>	1
P6CR<P60C>	1

P62 SCLK2 output setting

P6FC<P62F>	1
P6CR<P62C>	1

P63 TXD3 output setting

P6FC<P63F>	1
P6CR<P63C>	1

P65 SCLK2 output setting

P6FC<P65F>	1
P6CR<P65C>	1

P66 TXD4 output setting

P6FC<P66F>	1
P6CR<P66C>	1

Note 1: Read-modify-write is prohibited for registers P6CR and P6FC.

Note 2: When port P6 is used in the input mode, P6 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Note 3: P61/RXD2, P64/RXD3, P67/RXD4 pins have no port/function switch registers. When using as input ports, the serial receive data is input to SIO.

Figure 3.5.17 Registers for Port 6

3.5.8 Port 7 (P70 to P73)

Port 7 is a 4-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets port 7 as an input port and connects a pull-up resistor. In addition to functioning as a general-purpose I/O port, port 70 also functions as an input clock pin TI0 of an 8-bit timer 0, port 71 as an 8-bit timer output (TO1), port 72 as a PWM0 output (TO2), and port 73 as a PWM1 output (TO3) pin. Writing 1 in the corresponding bit of the port 7 function register (P7FC) enables output of the timer. Resetting resets the function register P7CR, P7FC to 0, and sets all bits to ports.

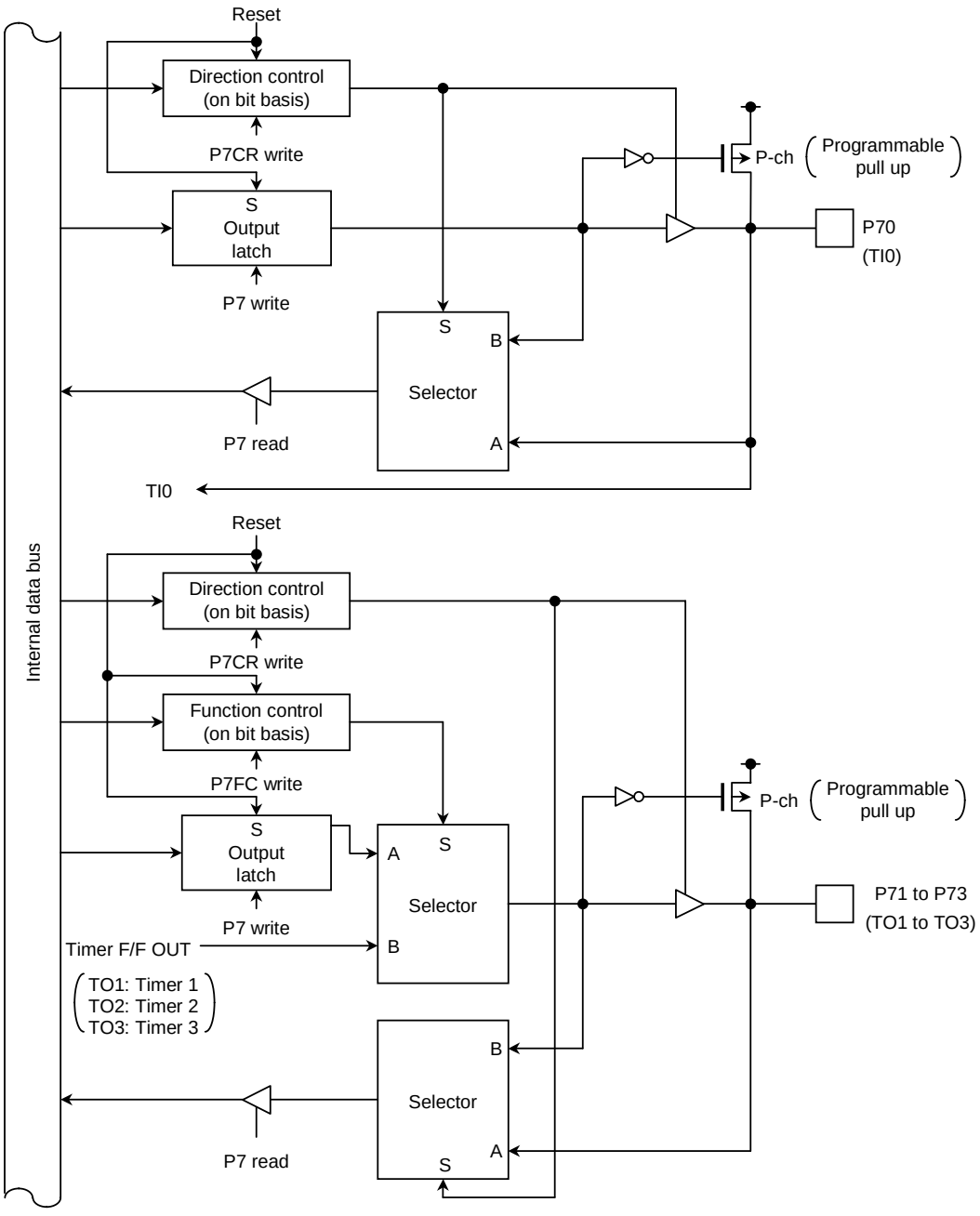


Figure 3.5.18 Port 7

Port 7 Register

		7	6	5	4	3	2	1	0
P7	Bit symbol					P73	P72	P71	P70
(0013H)	Read/Write					R/W			
	After reset					1	1	1	1
	Function					Input mode (Pull up)			

Port 7 Control Register

		7	6	5	4	3	2	1	0
P7CR	Bit symbol					P73C	P72C	P71C	P70C
(0015H)	Read/Write					W			
	After reset					0	0	0	0
	Function					0: Input 1: Output			

Port 7 I/O setting

0	Input
1	Output

Port 7 Function Register

		7	6	5	4	3	2	1	0
P7FC	Bit symbol					P73F	P72F	P71F	
(0017H)	Read/Write					W			
	After reset					0	0	0	
	Function					0: Port 1: TO3	0: Port 1: TO2	0: Port 1: TO1	

Setting P71 as TO1

P7FC<P71F>	1
P7CR<P71C>	1

Setting P72 as TO2

P7FC<P72F>	1
P7CR<P72C>	1

Setting P73 as TO3

P7FC<P73F>	1
P7CR<P73C>	1

Note 1: Read-modify-write is prohibited for registers P7CR and P7FC.

Note 2: When port P7 is used in the input mode, P7 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Note 3: P70/TI0 pin does not have a register changing port/function.
For example, when it is used as an input port, the input signal is inputted to 8-bit timer 0 as a timer input 0 (T10).

Note 4: P4<Bit7:4> is always read as “1”.

Figure 3.5.19 Registers for Port 7

3.5.9 Port 8 (P80 to P87)

Port 8 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis. Resetting sets port 8 as an input port and connects a pull-up resistor. It also sets all bits of the output latch register P8 to 1. In addition to functioning as a general-purpose I/O port, port 8 also functions as an input for 16-bit timer 4 and 5 clocks, an output for 16-bit timer F/F 4, 5, and 6 output, and an input for INT0. Writing "1" in the corresponding bit of the port 8 function register (P8FC) enables those functions. Resetting resets the function register P8CR, P8FC to "0" and sets all bits to ports.

(1) P80 to P86

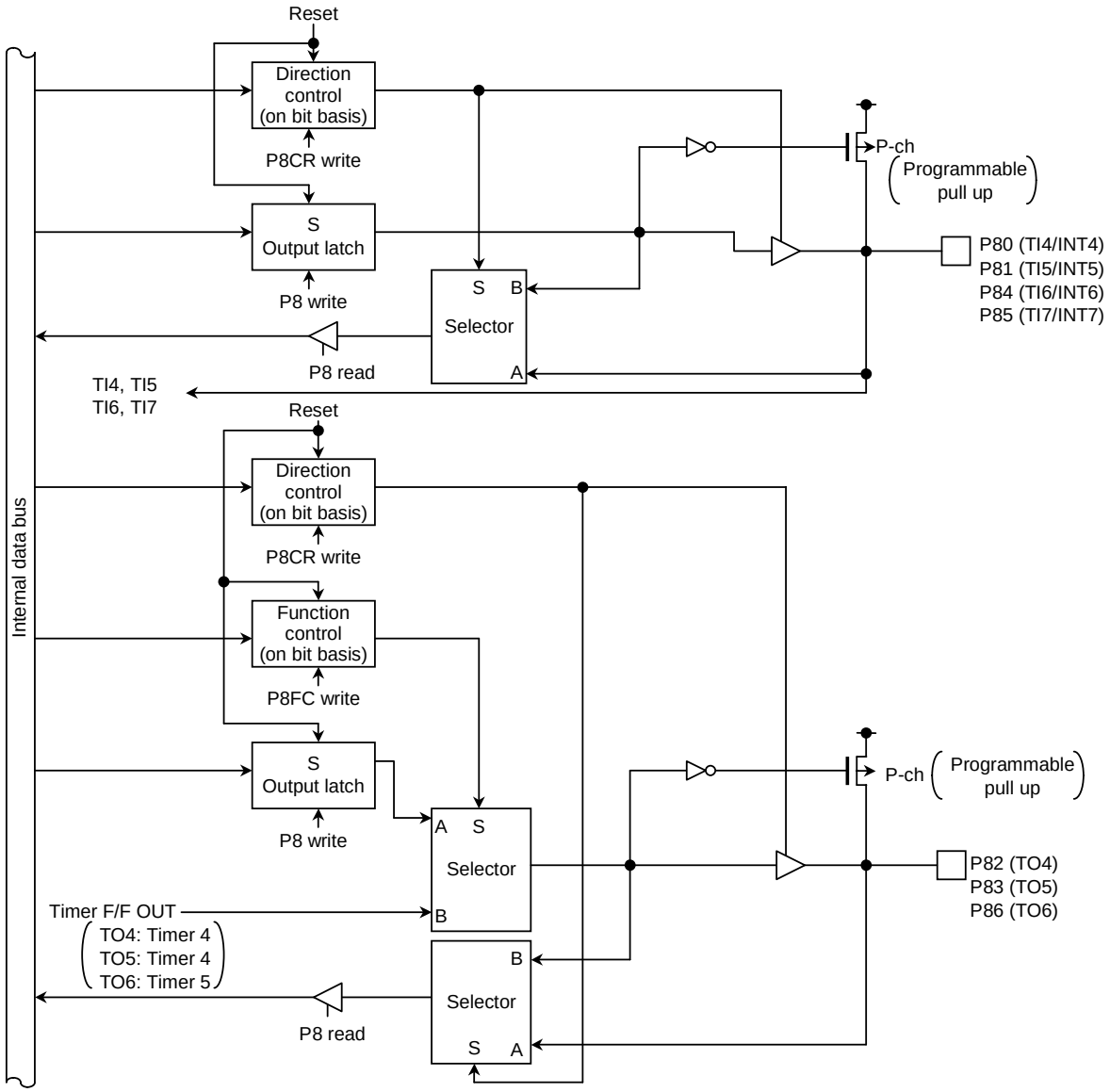


Figure 3.5.20 Port 8 (P80 to P86)

(2) P87 (INT0)

Port 87 is a general-purpose I/O port, and also used as an INT0 pin for external interrupt request input.

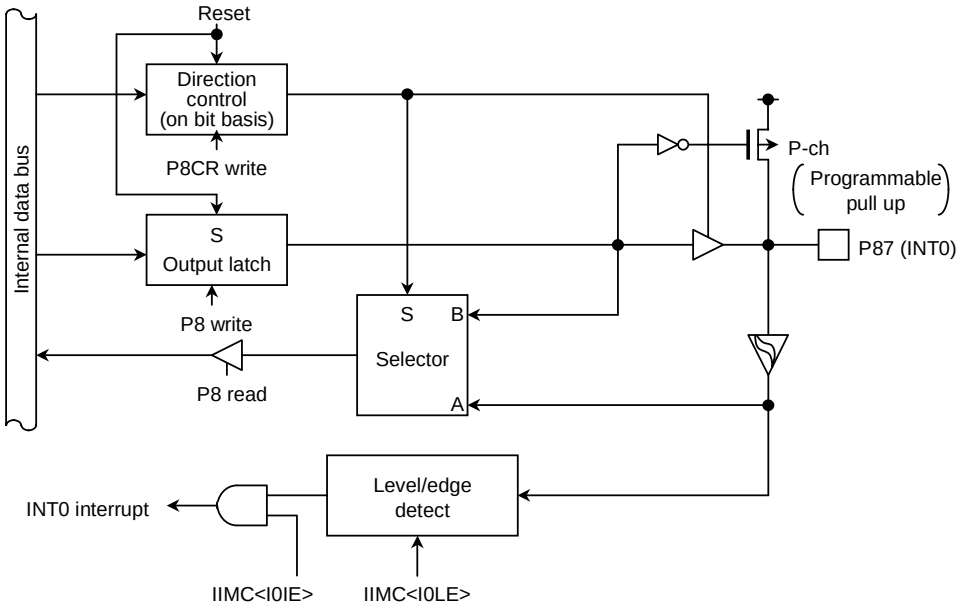


Figure 3.5.21 Port 87

Port 8 Register

		7	6	5	4	3	2	1	0
P8	Bit symbol	P87	P86	P85	P84	P83	P82	P81	P80
(0018H)	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
		Input mode (Pull up)							

Port 8 Control Register

		7	6	5	4	3	2	1	0
P8CR	Bit symbol	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
(001AH)	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

Port 8 I/O setting

0	Input
1	Output

Port 8 Function Register

		7	6	5	4	3	2	1	0
P8FC	Bit symbol		P86F			P83F	P82F		
(001CH)	Read/Write		W			W	W		
	After reset		0			0	0		
	Function		0: Port 1: TO6			0: Port 1: TO5	0: Port 1: TO4		

Setting P82 as TO4

P8FC<P82F>	1
P8CR<P82C>	1

Setting P83 as TO5

P8FC<P83F>	1
P8CR<P83C>	1

Setting P84 as TO6

P8FC<P86F>	1
P8CR<P86C>	1

Note 1: Read-modify-write is prohibited for registers P8CR and P8FC.

Note 2: When port P8 is used in the input mode, P8 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Note 3: P80/T14, P81/T15, P84/T16, P85/T17 pins do not have a register changing port/function.
For example, when it is used as an input port, the input signal is inputted to 16-bit timer as a time input.
When P87/INT0 pin is used as an INT0 pin, set P8CR<P87C> to “0” and IIMC<I0IE> to “1”.

Figure 3.5.22 Registers for Port 8

3.5.10 Port 9 (P90 to P97)

- Port 90 to 95

Port 90 to 95 is a 6-bit general-purpose I/O port. I/Os can be set on a bit basis. Resetting sets P90 to P95 to an input port and connects a pull-up resistor.

It also sets all bits of the output latch register to 1.

In addition to functioning as a general-purpose I/O port, P90 to P95 can also function as an I/O for serial channels 0 and 1. Writing "1" in the corresponding bit of the port 9 function register (P9FC) enables those functions.

Resetting resets the function register P9CR, P9FC to "0" and sets all bits to ports.

- Port 96 to 97

Port 96 to 97 is a 2-bit general-purpose I/O port. I/Os can be set on a bit basis. The output buffer for P96 to P97 is an open drain type buffer.

Resetting sets output latch and control registers to "1" and outputs high-impedance (High-Z).

In addition to functioning as a general-purpose I/O port, P96 to P97 can also function as a low-frequency oscillator connecting pin (XT1, XT2) for dual clock mode. The dual clock function can be set by programming system clock control register SYSCR0, 1.

(1) Port 90, 93 (TXD0/TXD1)

Ports 90 and 93 also function as serial channel TXD output pins in addition to I/O ports.

They have a programmable open-drain function.

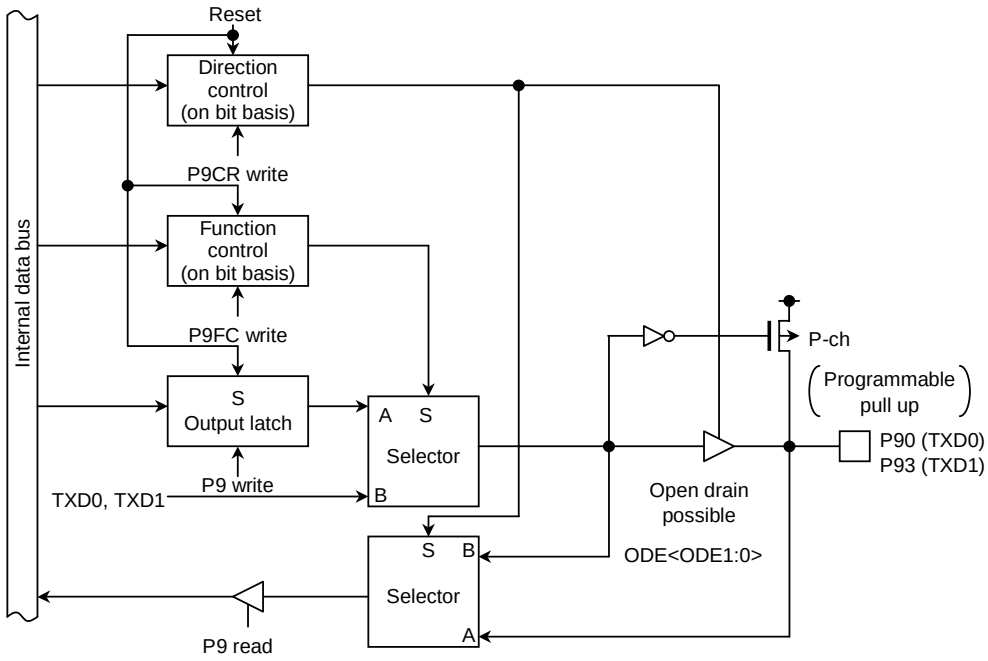


Figure 3.5.23 Ports 90 and 93

(2) Port 91, 94 (RXD0, RXD1)

Port 91and 94 are I/O ports, and also used as RXD input pins for serial channels.

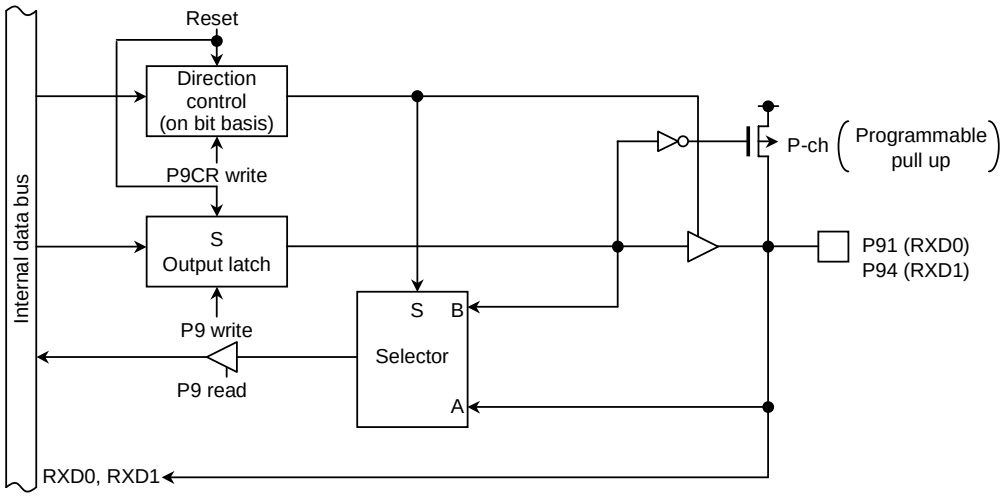


Figure 3.5.24 Ports 91 and 94

(3) Port 92 ($\overline{\text{CTS0}}$ /SCLK0)

Port 92 is an I/O port, and also used as a $\overline{\text{CTS0}}$ input pin and as a SCLK0 I/O pin for serial channel 0.

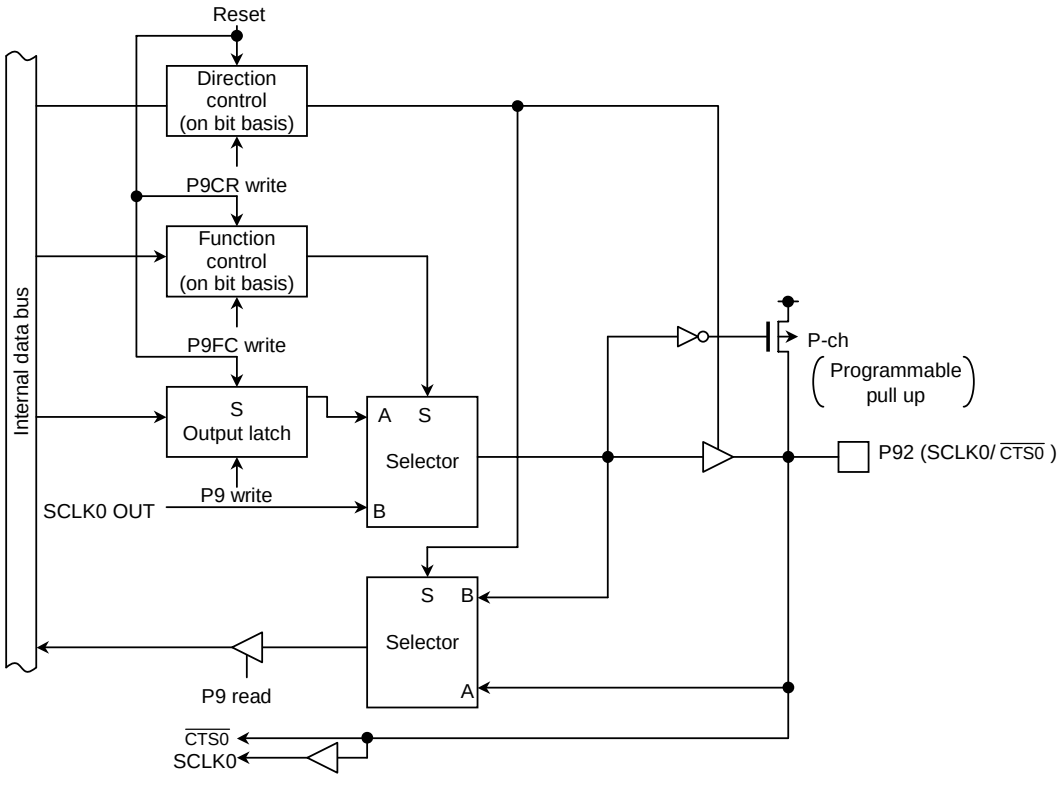


Figure 3.5.25 Ports 92

(4) Port 95 (SCLK1)

Port 95 is a general-purpose I/O port. It is also used as a SCLK1 I/O pin for serial channel 1.

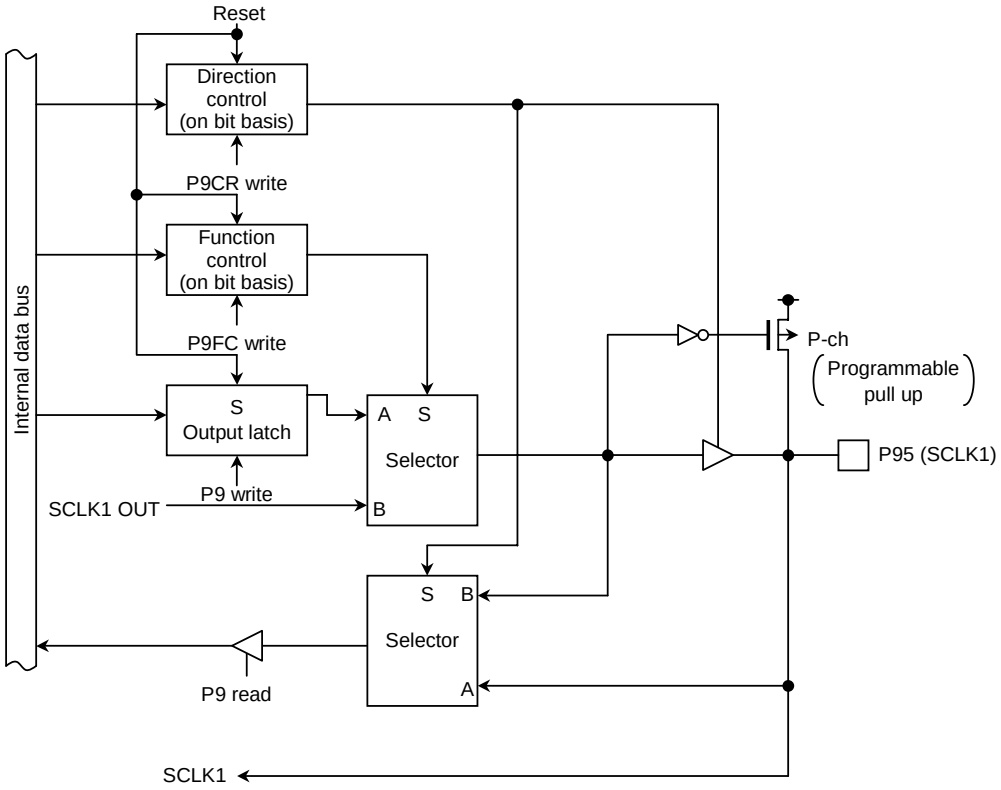


Figure 3.5.26 Port 95

(5) Port 96 (XT1), 97(XT2)

Port 96, 97 is general purpose I/O ports. It is also used as a low-frequency oscillator connecting pin.

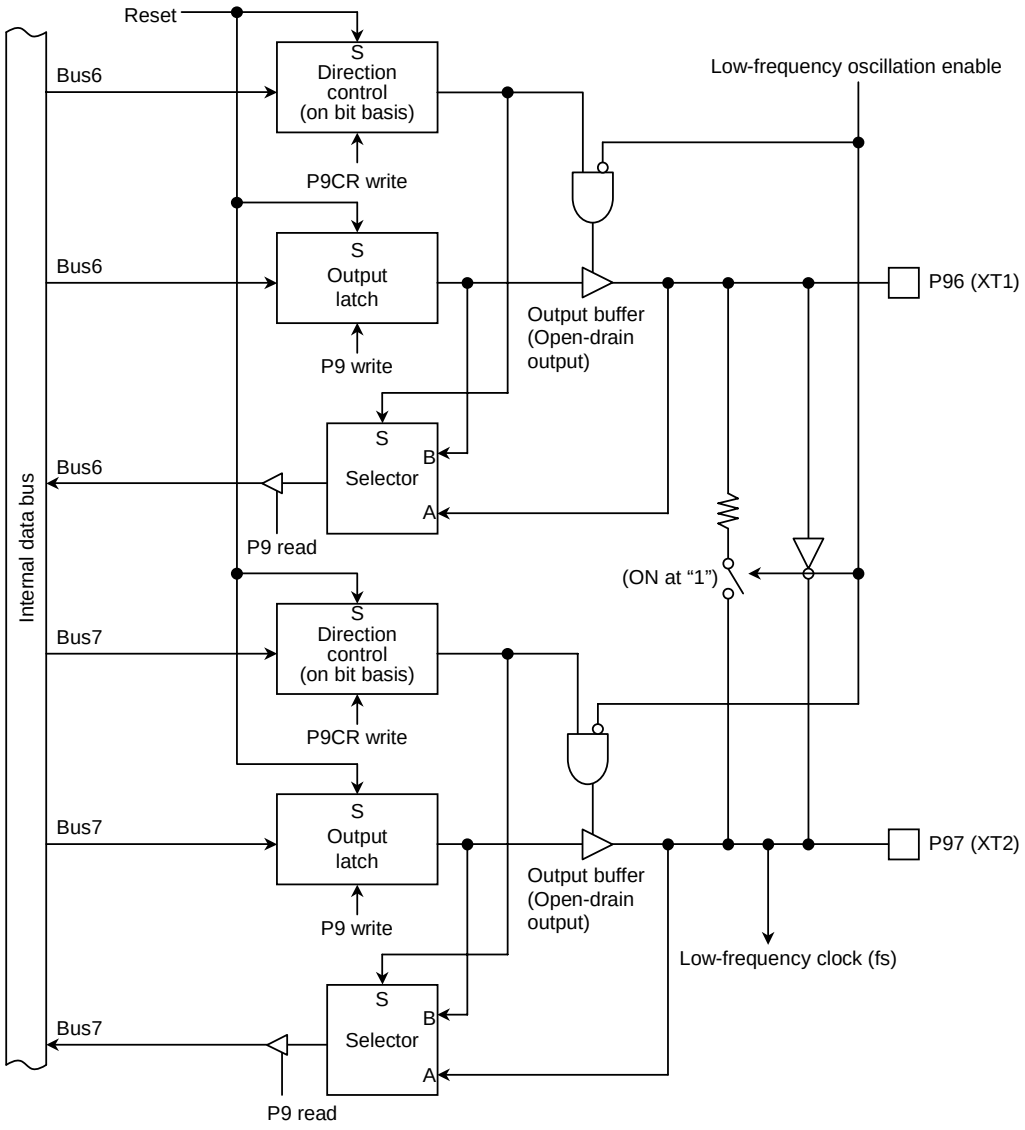


Figure 3.5.27 Port 96 to 97

Port 9 Register

		7	6	5	4	3	2	1	0
P9	Bit symbol	P97	P96	P95	P94	P93	P92	P91	P90
(0019H)	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
		Output mode		Input mode					

Port 9 Control Register

		7	6	5	4	3	2	1	0
P9CR	Bit symbol	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
(001BH)	Read/Write	W							
	After reset	1	1	0	0	0	0	0	0
	Function	0: Input 1: Output							

Port 9 I/O setting

0	Input
1	Output

Note: Port 96, 97's output buffer is an open drain output type.

Port 9 Function Register

		7	6	5	4	3	2	1	0
P9FC	Bit symbol			P95F		P93F	P92F		P90F
(001DH)	Read/Write			W		W	W		W
	After reset			0		0	0		0
	Function			0: Port 1: SCLK1		0: Port 1: TXD1	0: Port 1: SCLK0		0: Port 1: TXD0

P90 TXD0 output setting (Note)

P9FC<P90F>	1
P9CR<P90C>	1

P92 SCLK0 output setting

P9FC<P92F>	1
P9CR<P92C>	1

P93 TXD1 output setting (Note)

P9FC<P93F>	1
P9CR<P93C>	1

P95 SCLK1 output setting

P9FC<P95F>	1
P9CR<P95C>	1

Note 1: Read-modify-write is prohibited for registers P9CR and P9FC.

Note 2: When port P9 is used in the input mode, P9 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin. Port 96 and 97 have no built-in pull-up resistors.

Note 3: To set the TXD pin to open drain, write "1" in bit 0 (for TXD0 pin) or 1 (for TXD1) pin of the ODE register. P91/RXD0, P94/RXD1 pins have no port/function switch registers. When using as input port, serial receive data is input to SIO.

Note 4 Notes on using low-frequency oscillation circuit

To connect a low-frequency resonator to port 96, 97, it is necessary to set the following procedures to reduce the consumption power supply.

(connecting to a resonator)

Set P9CR<P96C, P97C> = "11", P9<P96, P97> = "00".

(connection to an oscillator)

Set P9CR<P96C, P97C> = "11", P9<P96, P97> = "10".

Note 5: When ports 96 and 97 is used in the output mode, input gate in operation. Set output to "L" or attach pull-up on pin to reduce the consumption of power, before the HALT instruction is executed.

Figure 3.5.28 Register for Port 9

3.5.11 Port A (PA0 to PA7)

Port A is an 8-bit general-purpose I/O port. Port A0 to A5 is possible to output large current and drive LED directly. I/Os can be set on a bit basis by control register PACR. After reset, PACR is reset to "0" and port A is set to an input port.

In addition to functioning as a general-purpose I/O port (only PA7), PA7 can also functions as a clock output pin. The output clock is f_{FPH} or f_{SYS} that is selected by the CKOCR<SCOSEL>. This function is enabled by setting PACR<PA7C> and CKOCR<SCOEN> to "1".

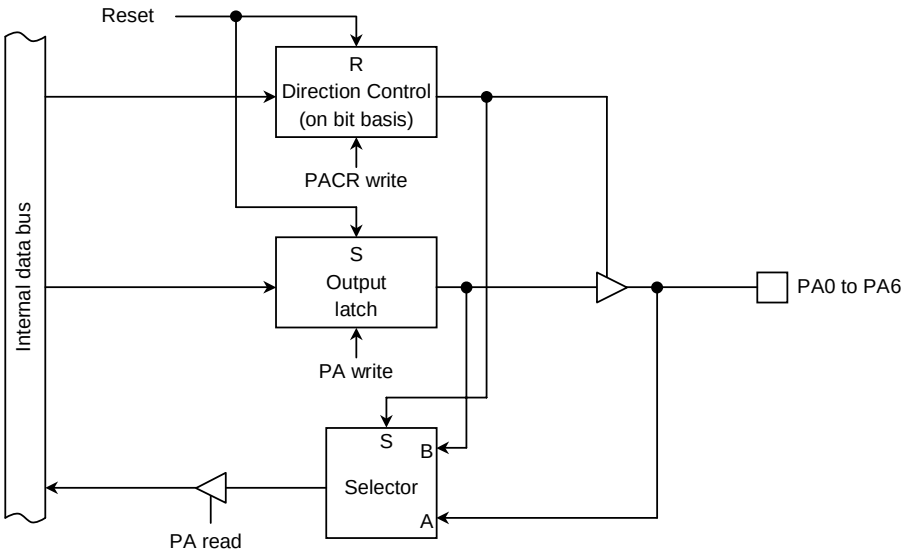


Figure 3.5.29 Port A0 to A6

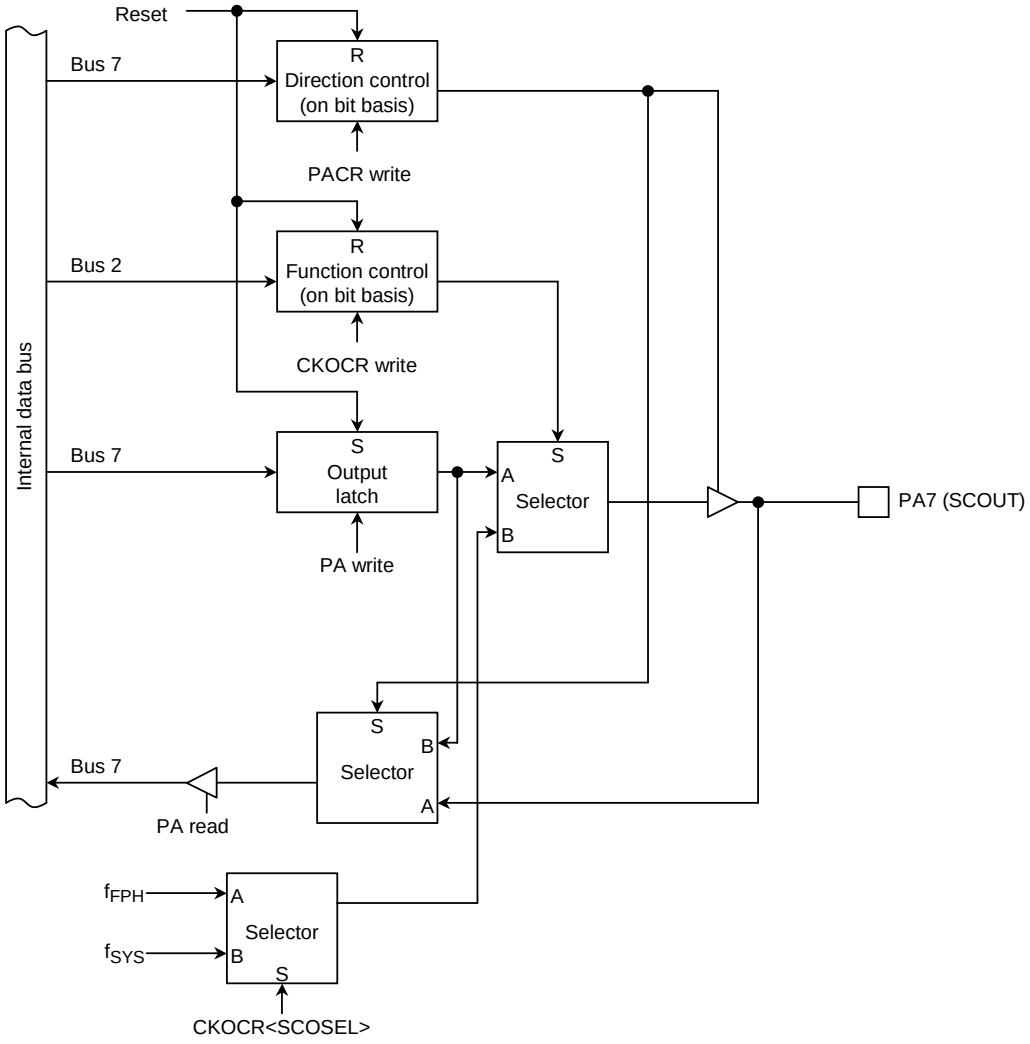


Figure 3.5.30 Port A7

Port A Register

		7	6	5	4	3	2	1	0
PA	Bit symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
(001EH)	Read/Write	R/W							
	After reset	Input mode							
		1	1	1	1	1	1	1	1

Port A Control Register

		7	6	5	4	3	2	1	0
PACR	Bit symbol	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
(001FH)	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

Clock Output Control Register

		7	6	5	4	3	2	1	0
CKOCR	Bit symbol					SCOSEL	SCOEN	ALEEN	CLKEN
(006DH)	Read/Write					R/W			
	After reset					0	0	0/1	0/1
	Function					Clock select	Clock enable	ALE enable	CLK enable

CLK pin output control (Note 2)

0	"High-Z" output
1	CLK output

ALE pin output control (Note 2)

0	"High-Z" output
1	ALE output

SCOUT/PA7 pin control

<SCOEN>	PACR<PA7C> <SCOSEL>	0	1
0	0	Input Port	Output mode
	1		
1	0		f_{FPH} clock output
	1		f_{SYS} clock output (Note 3)

Note 1: Read-modify-write is prohibited for registers PACR.

Note 2: The value after reset of <CLKEN>, <ALLEN> is "0" (High-impedance output). The CLK pin is pulled up internally during reset.

Note 3: The output clock from SCOUT pin is f_{FPH} or f_{SYS} clock.

e.g.) The case of connected 20 MHz oscillator to X1, X2 pin.
<SCOSEL> = "0" → 20 MHz clock
<SCOSEL> = "1" → 10 MHz clock

Figure 3.5.31 Registers for Port A

3.6 Chip Select/Wait Controller, AM8/$\overline{AM16}$ Pin

TMP93CW46A has a built-in chip select/wait controller used to control chip select ($\overline{CS0}$ to $\overline{CS2}$ pins), wait ($\overline{WAIT}$ pin), and data bus size (8 or 16 bits) for any of the three block address areas.

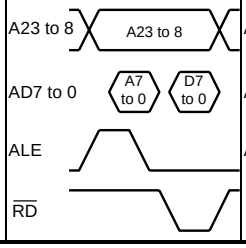
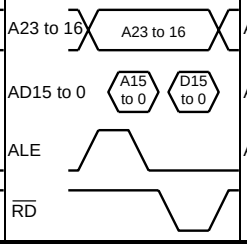
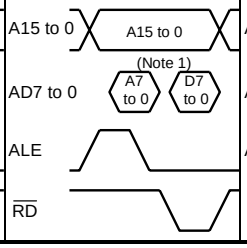
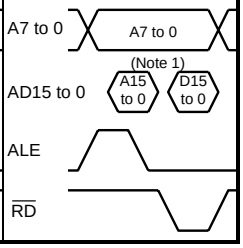
And AM8/$\overline{AM16}$ pin selects external data bus width for TMP93CW46A.

3.6.1 AM8/$\overline{AM16}$ Pin

Set this pin to "1". After reset, the CPU accesses the internal ROM with 16-bit bus width. The bus width when the CPU accesses an external area is set by chip select/wait control register (Described at 3.6.3.) and the registers of port 1.

3.6.2 Address/Data Bus Pins

Port 0/AD0 to AD7, Port 1/AD8 to AD15 and Port 2/AD16 to AD23/A0 to A7 function as address/data bus for connecting the external memories.

		a.	b.	c.	d.
Number of address bus pins		Max 24 (to 16 Mbytes)	Max 24 (to 16 Mbytes)	Max 16 (to 64 Kbytes)	Max 8 (to 256 bytes)
Number of data bus pins		8	16	8	16
Number of multiplexed pins		8	16	0	0
Port function	Port 0	AD0 to AD7	AD0 to AD7	AD0 to AD7	AD0 to AD7
	Port 1	A8 to A15	AD8 to AD15	A8 to A15	AD8 to AD15
	Port 2	A16 to A23	A16 to A23	A0 to A7	A0 to A7
Timing chart					

Note 1: In case of c. and d., the data bus signals output the addresses since the signals are also used as the address bus. Writing "0" to bit CKOCR<ALEEN>, ALE signal can be stopped outputting.

Note 2: After reset operation, Port 0, Port 1 and Port 2 function as input ports.

Note 3: All the options a. to d. can be made available using the P1CR, P1FC, P2CR and P2FC registers. ($\overline{EA}$ = VIH, AM8/$\overline{AM16}$ = VIH)

3.6.3 Chip Select/Wait Control Registers

Table 3.6.1 shows control registers.

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

(1) Enable

Control register bit7 (B0E, B1E, and B2E) is a master bit used to specify enabling ("1")/disabling ("0") of the setting.

Resetting sets B0E and B1E to disable ("0") and B2E to enable ("1").

(2) CS/CAS Waveform select

Control register bit5 (B0CAS, B1CAS, and B2CAS) is used to specify waveform mode output from the chip select pin ($\overline{CS0}$ / $\overline{CAS0}$ to $\overline{CS2}$ / $\overline{CAS2}$). Setting this bit to 0 specifies $\overline{CS0}$ to $\overline{CS2}$ waveforms; setting it to 1 specifies $\overline{CAS0}$ to $\overline{CAS2}$ waveforms.

Resetting clears bit5 to 0.

(3) Data bus size select

Bit4 (B0BUS, B1BUS, and 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 3.6.2 shows the details of the bus operation.

(4) Wait control

Control register bits 3 and 2 (B0W1 to B0W0, B1W1 to B1W0, B2W1 to B2W0) are used to specify the number of waits. Setting these bits to 00 inserts a 2 states wait regardless of the $\overline{WAIT}$ pin status. Setting them to 01 inserts a 1-state wait regardless of the $\overline{WAIT}$ status. Setting them to 10 inserts a 1-state wait and samples the $\overline{WAIT}$ pin status. If the pin is low, inserting the wait maintains the bus cycle until the pin goes high. Setting them to 11 completes the bus cycle without a wait regardless of the $\overline{WAIT}$ pin status.

Resetting sets these bits to 00 (2-state wait mode).

(5) Address area specification

Control register bits 1 and 0 (B0C1 to B0C0, B1C1 to B1C0, B2C1 to B2C0) are used to specify the target address area. Setting these bits to 00 enables settings (CS output, Wait state, Bus size, etc.) as follows:

* CS0 setting enabled when 7F00H to 7FFFH is accessed.
* CS1 setting enabled when 1080H to 7FFFH is accessed.
* CS2 setting enabled when 28000H to 3FFFFH is accessed.

Setting bits to 01 enables setting for all CS's blocks and outputs a low strobe signal ($\overline{CS0}$ / $\overline{CAS0}$ to $\overline{CS2}$ / $\overline{CAS2}$) from chip select pins 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.

Table 3.6.1 Chip Select/Wait Control Register

Code	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block0 CS/WAIT control register	0068H	B0E		B0CAS	B0BUS	B0W1	B0W0	B0C1	B0C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: Master bit of bit 0 to 6		0: $\overline{CS0}$ 1: $\overline{CAS0}$	0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + N) waits 11: 0 waits		00: 7F00H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B1CS	Block1 CS/WAIT control register	0069H	B1E		B1CAS	B1BUS	B1W1	B1W0	B1C1	B1C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: Master bit of bit 0 to 6		0: $\overline{CS1}$ 1: $\overline{CAS1}$	0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + N) waits 11: 0 waits		00: 1080H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B2CS	Block2 CS/WAIT control register	006AH	B2E		B2CAS	B2BUS	B2W1	B2W0	B2C1	B2C0
			W		W	W	W	W	W	W
			1		0	0	0	0	0	0
			1: Master bit of bit 0 to 6		0: $\overline{CS2}$ 1: $\overline{CAS2}$	0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + N) waits 11: 0 waits		00: 28000H to 01: 400000H to 10: 800000H to 11: C00000H to	

Table 3.6.2 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 wirte strobe signal remains non active.

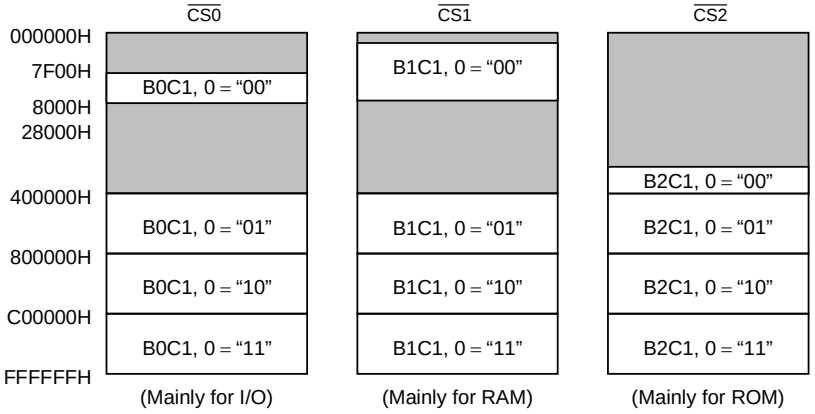
3.6.4 Chip Select Image

An image of the actual chip select is shown below. Out of the whole memory area, address areas that can be specified are divided into four parts. Addresses from 000000H to 3FFFFFH are divided differently: 7F00H to 7FFFH is specified for CS0; 1080H to 7FFFH, for CS1; and 28000H to 3FFFFFH, for CS2. The reason is that a device other than ROM (e.g., RAM or I/O) might be connected externally.

7F00H to 7FFFH (256 bytes) for CS0 are mapped mainly for possible expansions to external I/O.

1080H to 7FFFH (Approx. 31 Kbytes) for CS1 are mapped there mainly for possible extensions to external RAM.

28000H to 3FFFFFH (Approx. 4 Mbytes) for CS2 are mapped mainly for possible extensions to external ROM. With the TMP93CW46A which has a built-in ROM, addresses from 8000H to 27FFFH are used as the internal ROM area. After reset, the CPU reads the program from the built-in ROM in 16-bit bus, 0-wait mode.



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 $\overline{CS0}$ to $\overline{CS2}$ are accessed in 16-bit data bus (0 waits) mode. When using the chip select/wait controller, do not specify the same address area more than once. (However, when addresses 7F00H to 7FFFH for CS0 and 1080H to 7FFFH for CS1 are specified, in other words, specifications overlap, only the CS0 setting/pin is active.)

Note 3: When the bus is released ($\overline{BUSAK}$ = "0"), $\overline{CS0}$ to $\overline{CS2}$ pins are also released (the output buffer is OFF). Refer to note about the bus release in 3.5 "Functions of Ports" about the state of pins.

3.6.5 Example of Usage

Figure 3.6.1 is an example in which an external memory is connected to the TMP93CW46A. In this example, a ROM 128 Kbytes is connected using 16-bit bus, and RAM 256 Kbytes using 16-bit bus.

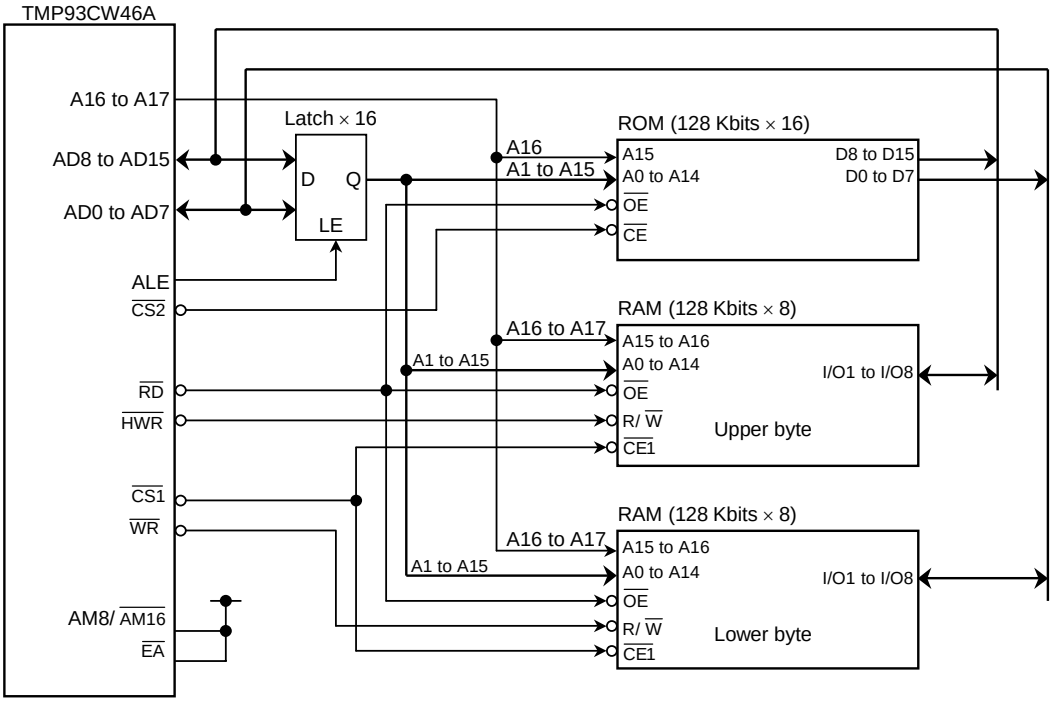


Figure 3.6.1 Example of External Memory Connection (ROM and RAM = 16 bits)

TMP93CW46A has built-in ROM and RAM. When ROM and RAM have insufficient capacity, it is possible to connect an external memory as the example of the external memory connection. In this example, the memory configuration is as follows.

Memory		Memory Size	Address	$\overline{CS}$ Pin	Data Bus
ROM	Internal	128 Kbytes	008000H to 027FFFH	–	16 bits
	External	128 Kbytes	400000H to 41FFFFH	$\overline{CS2}$	16 bits
SRAM	Internal	4 Kbytes	000080H to 00107FH	–	16 bits
	External	256 Kbytes	800000H to 83FFFFH	$\overline{CS1}$	16 bits

3.7 8-Bit Timers

The TMP93CW46A contains two 8-bit timers (Timers 0 and 1), each of which can be operated independently. The cascade connection allows these timers to be used as 16-bit timer. The following four operating modes are provided for the 8-bit timers.

- 8-bit interval timer mode (2 timers)
- 16-bit interval timer mode (1 timer)
- 8-bit programmable square wave pulse generation (PPG: Variable duty with variable cycle) output mode (1 timer)
- 8-bit pulse width modulation (PWM: variable duty with constant cycle) output mode (1 timer)

Figure 3.7.1 shows the block diagram of 8-bit timer (Timer 0 and timer 1).

Each timer consists of an 8-bit up counter, 8-bit comparator, and 8-bit timer register. Besides, one timer flip-flop (TFF1) is provided for pair of timer 0 and timer 1.

Among the input clock sources for the timers, the internal clocks of ϕT1, ϕT4, ϕT16, and ϕT256 are obtained from the 9-bit prescaler shown in Figure 3.7.2.

The operation modes and timer flip-flops of the 8-bit timer are controlled by three control registers TMOD, TFFCR, and TRUN.

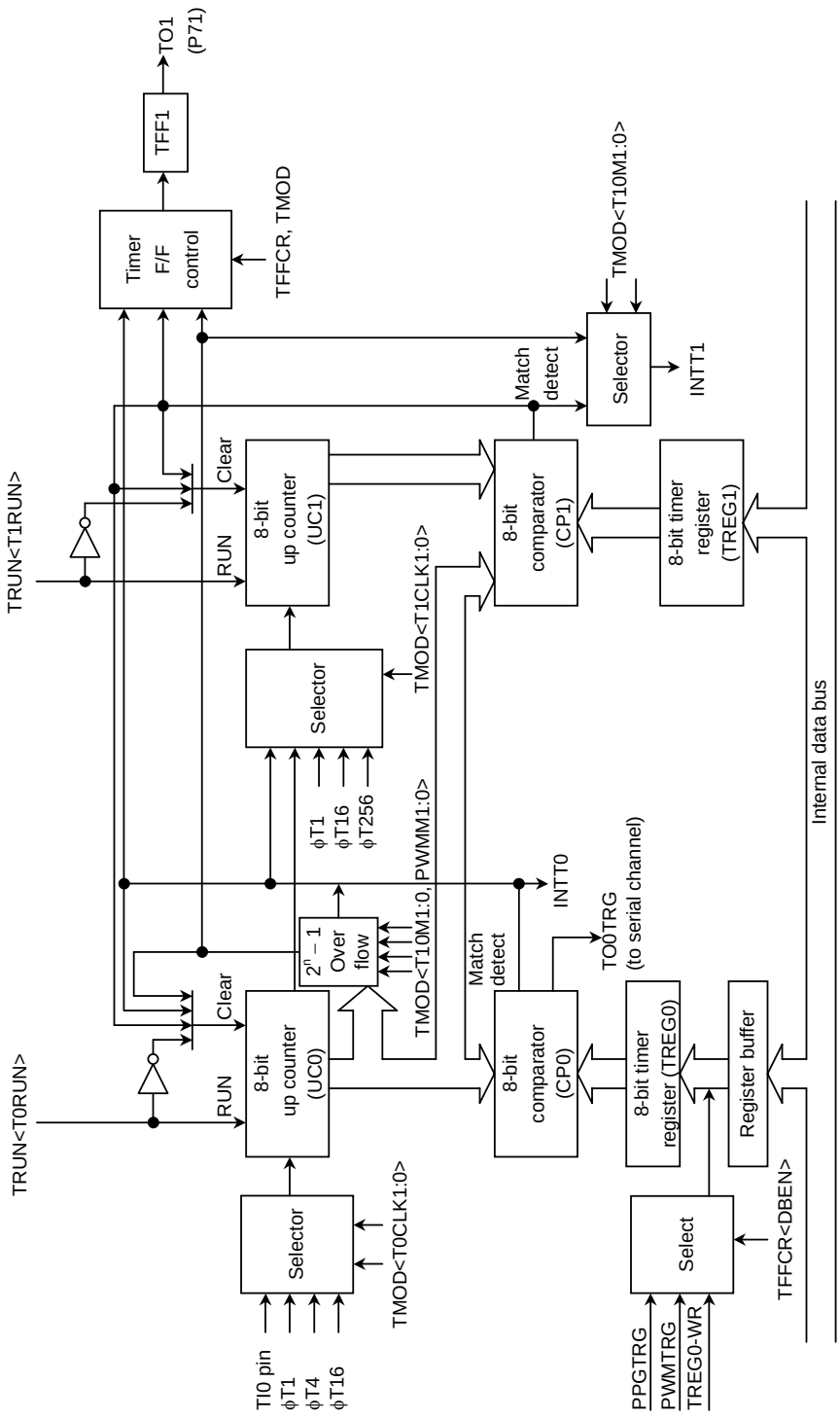


Figure 3.7.1 Block Diagram of 8-Bit Timers (Timers 0 and 1)

1. Prescaler, Prescaler clock select

There are 9 bit prescaler and prescaler clock selection register to generate input clock for 8-bit timers 0 and 1, 16-bit timers 4 and 5 and serial interfaces 0 to 4.

Figure 3.7.2 shows the block diagram. Table 3.7.1 shows prescaler clock resolution into 8, 16-bit timer.

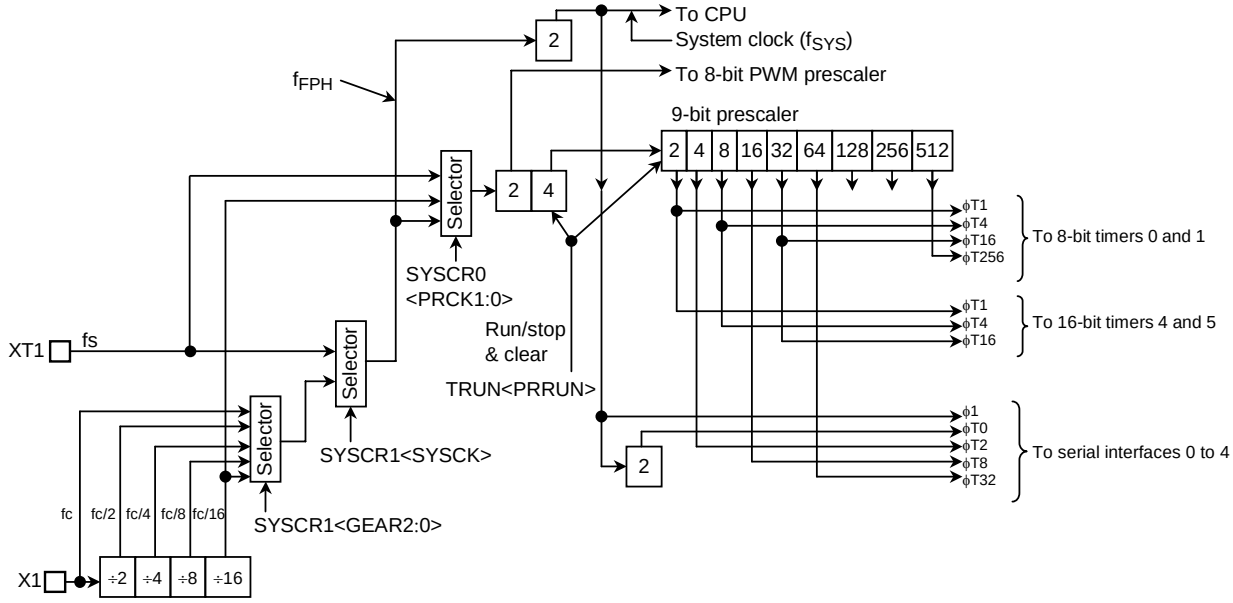


Figure 3.7.2 The Block Diagram of Prescaler

Table 3.7.1 Prescaler Clock Resolution to 8-/16-Bit Timer

at fc = 20 MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	Prescaler Clock Resolution			
			φT1	φT4	φT16	φT256
1 (fs)	00 (f_{FPH})	XXX	$fs/2^3$ (244 μs)	$fs/2^5$ (977 μs)	$fs/2^7$ (3.9 ms)	$fs/2^{11}$ (62.5 ms)
0 (fc)		000 (fc)	$fc/2^3$ (0.4 μs)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)	$fc/2^{11}$ (102.4 μs)
		001 (fc/2)	$fc/2^4$ (0.8 μs)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{12}$ (204.8 μs)
		010 (fc/4)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{13}$ (409.6 μs)
		011 (fc/8)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{10}$ (51.2 μs)	$fc/2^{14}$ (819.2 μs)
		100 (fc/16)	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{11}$ (102.4 μs)	$fc/2^{15}$ (1.638 ms)
XXX	01 (Low-frequency clock)	XXX	$fs/2^3$ (244 μs)	$fs/2^5$ (977 ms)	$fs/2^7$ (3.9 ms)	$fs/2^{11}$ (62.5 ms)
XXX	10 (Note) (fc/16 clock)	XXX	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{11}$ (102.4 μs)	$fc/2^{15}$ (1.638 ms)

XXX: Don't care

16-bit timer

8-bit timer

Note: The fc/16 clock as a prescaler clock can not be used when the fs is used as a system clock.

The clock selected among f_{FPH} clock, fc/16 clock and fs is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0<PRCK1:0>.

Resetting sets <PRCK1:0> to "00", selecting the f_{FPH} clock input divided by 4.

The 8-bit timer 0, 1 selects between 4 clock inputs: ϕT1, ϕT4, ϕT16, and ϕT256 among the prescaler output.

This prescaler can be run or stopped by the timer control register TRUN<PRRUN>. Counting starts when <PRRUN> is set to "1". The prescaler is cleared to zero and stops operation when <PRRUN> is set to "0". Resetting clear <PRRUN> to "0" and stops the prescaler.

When the IDLE1 mode (Only the oscillator operates) is used, set TRUN<PRRUN> to "0" to reduce the power consumption of the prescaler before the "HALT" instruction is executed.

2. Up counter

This is an 8-bit binary counter which counts up by the input clock pulse specified by TMOD.

The input clock of timer 0 is selected from the external clock from TI0 pin and the three internal clocks ϕT1, ϕT4, and ϕT16, according to the set value of TMOD register.

The input clock of timer 1 depends on the operation modes. When set to 16-bit timer mode, the overflow output of timer 0 is used as the input clock. When set to any other mode than 16-bit timer mode, the input clock is selected from the internal clocks ϕT1, ϕT16, and ϕT256 as well as the comparator output (Match detection signal) of timer 0 according to the set value of TMOD register.

Example: When TMOD<T10M1:0> = 01, the overflow output of timer 0 becomes the input clock of timer 1 (16-bit timer mode).

When TMOD<T10M1:0> = 00 and TMOD<T1CLK1:0> = 01, ϕT1 becomes the input of timer 1 (8-bit timer mode).

Operation mode is also set by TMOD register. When reset, it is initialized to TMOD<T10M1:0> = 00 whereby the up counter is placed in the 8-bit timer mode.

The counting and stop and clear of up counter can be controlled for each interval timer by the timer operation control register TRUN. When reset, all up counters will be cleared to stop the timers.

3. Timer register

This is an 8-bit register for setting an interval time. When the set value of timer registers TREG0, TREG1, matches the value of up counter, the comparator match detect signal becomes active. If the set value is 00H, this signal becomes active when the up counter overflows.

Timer register TREG0 is a double buffer structure, each of which makes a pair with register buffer.

The timer flip-flop control register TFFCR<DBEN> bit controls whether the double buffer structure should be enabled or disabled. It is disabled when <DBEN> = 0 and enabled when they are set to 1.

In the condition of double buffer enable state, the data is transferred from the register buffer to the timer register when the $2^n - 1$ overflow occurs in PWM mode, or at the PPG cycle in PPG mode. Therefore, during timer mode, the double buffer can not be used.

When reset, it will be initialized to <DBEN> = 0 to disable the double buffer. To use the double buffer, write data in the timer register, set <DBEN> to 1, and write the following data in the register buffer.

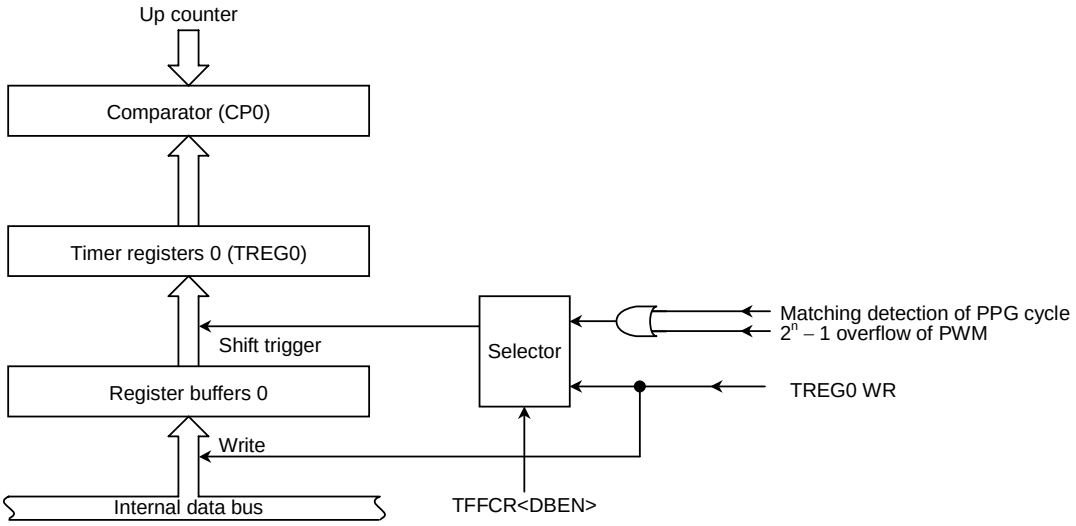


Figure 3.7.3 Configuration of Timer Register 0

Note: Timer register and the register buffer are allocated to the same memory address. When <DBEN> = 0, the same value is written in the register buffer as well as the timer register, while when <DBEN> = 1 only the register buffer is written.

The memory address of each timer register is as follows.

TREG0: 000022H

TREG1: 000023H

All the registers are write-only and cannot be read.

4. Comparator

   A comparator compares the value in the up counter with the values to which the timer register is set. When they match, the up counter is cleared to zero and an interrupt signal (INTT0, INTT1) is generated. If the timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

5. Timer flip-flop

   The timer flip-flop (TFF1) is a flip-flop inverted by the match detect signal (8-bit comparator output).

   Inverting is disabled or enabled by the timer flip-flop control register TFFCR<TFF1IE>.

   After reset operation, the value of TFF1 is undefined. Writing "01" or "10" to TFFCR<TFF1C1:0> sets "0" or "1" to TFF1. Additionally, writing "00" to this bit inverts the value of TFF1. (Software inversion)

   TFF1 is output to TO1 pin (Also used as P71). When using as the timer output, the timer flip-flop should be set by port 7 function register P7FC beforehand.

TRUN (0020H)

	7	6	5	4	3	2	1	0
Bit symbol	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Prescaler and timer run/stop control 0: Stop and clear 1: Run (Count up)							

Count operation

0	Stop and clear
1	Count

PRRUN: Operation of prescaler

T5RUN: Operation of 16-bit timer (Timer 5)

T4RUN: Operation of 16-bit timer (Timer 4)

P1RUN: Operation of PWM timer (PWM1/Timer 3)

P0RUN: Operation of PWM timer (PWM0/Timer 2)

T1RUN: Operation of 8-bit timer (Timer 1)

T0RUN: Operation of 8-bit timer (Timer 0)

SYSCR0 (006EH)

	7	6	5	4	3	2	1	0
Bit symbol	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
Read/Write	R/W							
After reset	1	0	1	0	0	0	0	0
Function	High-frequency oscillator (fc) 0: Stop 1: Oscillator	Low-frequency oscillator (fs) 0: Stop 1: Oscillator	High-frequency oscillator (fc) after released STOP mode 0: Stop 1: Oscillator	Low-frequency oscillator (fs) after released STOP mode 0: Stop 1: Oscillator	Select clock after released STOP mode 0: fc 1: fs	Warm-up timer (Write) 0: Don't care 1: Start timer (Read) 0: End warm up 1: Not end warm up	Select prescaler clock 00: f_{FPH} 01: fs 10: fc/16 11: (Reserved)	

Figure 3.7.4 Timer Operation Control Register/System Clock Control Register

TMOD (0024H)

	7	6	5	4	3	2	1	0
Bit symbol	T10M1	T10M0	PWMM1	PWMM0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		PWM cycle 00: – 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Source clock of timer 1 00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock of timer 0 00: TI0 01: ϕT1 10: ϕT4 11: ϕT16	

Input clock of timer 0

00	External input (TI0)
01	ϕT1 (Prescaler)
10	ϕT4 (Prescaler)
11	ϕT16 (Prescaler)

Input clock of timer 1

	TMOD<T10M1:0> ≠ 01	TMOD <T10M1:0> = 01
00	Comparator output of timer 0	Overflow output of timer 0 (16-bit timer mode)
01	Internal clock ϕT1	
10	Internal clock ϕT16	
11	Internal clock ϕT256	

Select PWM cycle

00	–
01	$2^6 - 1$
10	$2^7 - 1$
11	$2^8 - 1$

Set the operation mode of timers 0 and 1

00	Two 8-bit timers (Timer 0 and timer 1)
01	16-bit timer
10	8-bit PPG output
11	8-bit PWM output (Timer 0) 8-bit timer (Timer 1)

Note: Prohibit read-modify-write

Figure 3.7.5 Timer Mode Control Register (TMOD)

TFFCR (0025H)

	7	6	5	4	3	2	1	0
Bit symbol				DBEN	TFF1C1	TFF1C0	TFF1IE	TFF1IS
Read/Write				R/W	W		R/W	
After reset				0	1	1	0	0
Function				Double buffer 0: Disable 1: Enable	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care * Always read as "11"		TFF1 Inversion trigger 0: Disable 1: Enable	TFF1 Inversion source 0: Timer 0 1: Timer 1

→ Select inverse signal of timer F/F1
("Don't care" except in 8-bit timer mode)

	TMOD<T10M1:0 > = 00	01	10	11
0	Inversion by timer 0 match signal	16-bit timer mode Inversion by match signal	PPG mode Inversion by match signal of each timer 0 and timer 1	PWM mode Inversion by match and overflow signal of timer 0
1	Inversion by timer 1 match signal			

→ Inversion of timer F/F1 (TFF1)

0	Disable invert
1	Enable invert

→ Control of timer F/F1 (TFF1)

00	Invert the value of TFF1 (software inversion)
01	Set TFF1 to "1".
10	Clear TFF1 to "0".
11	Don't care

→ Double buffer control of TREG0

0	Disable double buffer
1	Enable double buffer

Note: TFFCR<bit7:5>, <bit3:2> are always read as "1".

Figure 3.7.6 Timer Flip-Flop Control Register (TFFCR)

The operation of 8-bit timers will be described below:

(1) 8-bit timer mode

Two interval timers 0, 1, can be used independently as 8-bit interval timer. All interval timers operate in the same manner, and thus only the operation of timer 1 will be explained below.

1. Generating interrupts in a fixed cycle

To generate timer 1 interrupt at constant intervals using timer 1 (INTT1), first stop timer 1 then set the operation mode, input clock, and a cycle to TMOD and TREG1 register, respectively. Then, enable interrupt INTT1 and start the counting of timer 1.

Example: To generate timer 1 interrupt every 1 s at fs = 32.768 MHz, set each register in the following manner.

* Clock condition
- System clock: Low frequency (fs)
- Clock gear: xxx
- Prescaler clock: Low frequency (fs)

	MSB 7	6	5	4	3	2	1	LSB 0	
TRUN	← –	X	–	–	–	–	0	–	Stop timer 1, and clear it to “0”.
TMOD	← 0	0	X	X	1	0	–	–	Set the 8-bit timer mode, and select ϕT16 (3.9 ms at fs = 32.768 kHz) as the input clock.
TREG1	← 0	0	0	0	0	0	0	0	Set the timer register 1 s ÷ ϕT16 = 256 (00H).
INTET10	← 1	1	0	1	–	–	–	–	Enable INTT1, and set it to level 5.
TRUN	← 1	X	–	–	–	–	1	–	Start timer 1 counting.

X: Don’t care, –: No change

Use the Table 3.7.1 for selecting the input clock.

Note: The input clock of timer 0 and timer 1 are different from as follows.
Timer 0: TI0 input, ϕT1, ϕT4, ϕT16
Timer 1: Match output of timer 0, ϕT1, ϕT16, ϕT256

2. Generating a 50% duty square wave pulse

The timer flip-flop (TFF1) is inverted at constant intervals, and its status is output to timer output pin (TO1).

Example: To output a 4.0 μs square wave pulse from TO1 pin at fc = 20 MHz, set each register in the following procedures. Either timer 0 or timer 1 may be used, but this example uses timer 1.

* Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: f_{FPH}

	7	6	5	4	3	2	1	0	
TRUN	← –	–	–	–	–	–	1	–	Stop timer 1, and clear it to "0".
TMOD	← 0	0	X	X	0	1	–	–	Set the 8-bit timer mode, and select ϕT1 (0.4 μs at fc = 20 MHz) as the input clock.
TREG1	← 0	0	0	0	0	1	0	1	Set the timer register 4.0 μs ÷ ϕT1 ÷ 2 = 5.
TFFCR	← –	–	–	–	1	0	1	1	Clear TFF1 to "0", and set to invert by the match detect signal from timer 1.
P7CR	← X	X	X	X	–	–	1	–	Select P71 as TO1 pin.
P7FC	← X	X	X	X	–	–	1	X	
TRUN	← 1	X	–	–	–	–	1	–	Start timer 1 counting.

X: Don't care, –: No change

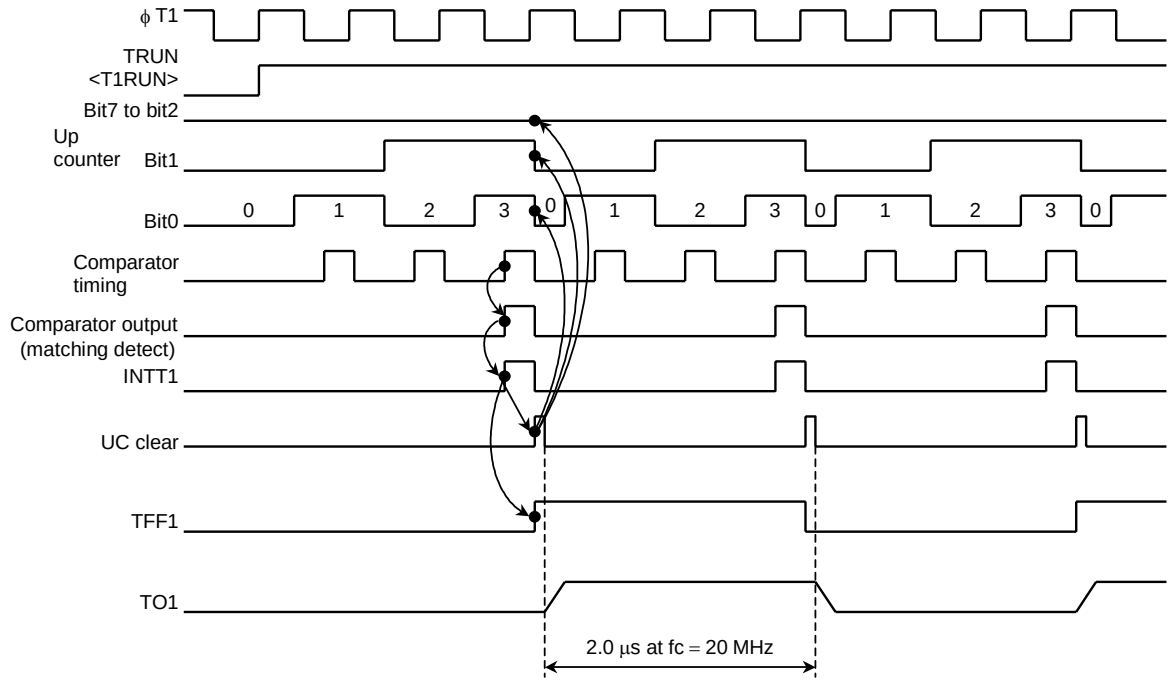


Figure 3.7.7 Square Wave (50% Duty) Output Timing Chart

3. Making timer 1 count up by match signal from timer 0 comparator

Set the 8-bit timer mode, and set the comparator output of timer 0 as the input clock to timer 1.

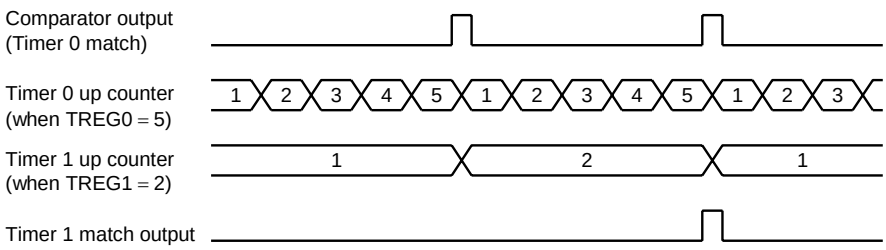


Figure 3.7.8 Timer 1 Count Up by Timer 0

(2) 16-bit timer mode

A 16-bit interval timer is configured by using the pair of timer 0 and timer 1.

To make a 16-bit timer mode, set timer 0/timer 1 mode register TMOD<T10M1:0> to "01".

When set in 16-bit timer mode, the overflow output of timer 0 will become the input clock of timer 1, regardless of the set value of TMOD<T1CLK1:0>. Table 3.7.1 shows the relation between the cycle of timer (Interrupt) and the selection of input clock.

The lower 8 bits of the timer (Interrupt) cycle are set by the timer register TREG0, and the upper 8 bits are set by TREG1. Note that TREG0 always must be set first. (Writing data into TREG0 disables the comparator temporarily, and the comparator is restarted by writing data into TREG1.)

Setting example: To generate an interrupt INTT1 every 0.4 seconds at fc = 20 MHz, set the following values for timer registers TREG0 and TREG1.

* Clock condition	System clock:	High frequency (fc)
	High frequency clock gear:	1 (fc)
	Prescaler clock:	f_{FPH}

When counting with input clock of ϕT16 (6.4 μs at 20 MHz)
0.4 s ÷ 6.4 μs = 62500 = F424H
Therefore, set TREG1 = F4H and TREG0 = 24H, respectively.

The comparator match signal is output from timer 0 each time the up counter UC0 matches TREG0, where the up counter UC0 is not be cleared.

With the timer 1 comparator, the match detect signal is output at each comparator timing when up counter UC1 and TREG1 values match. When the match detect signal is output simultaneously from both comparators of timer 0 and timer 1, the up counters UC0 and UC1 are cleared to "0", and the interrupt INTT1 is generated. If inversion is enabled, the value of the timer flip-flop TFF1 is inverted.

Example: When TREG1 = 04H and TREG0 = 80H

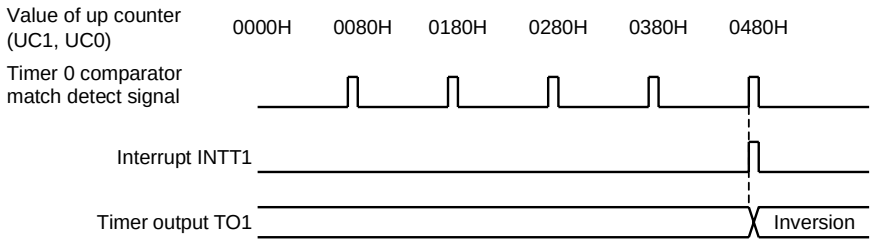


Figure 3.7.9 Timer Output by 16-Bit Timer Mode

(3) 8-bit PPG (Programmable pulse generation) output mode

Square wave pulse can be generated at any frequency and duty by timer 0. The output pulse may be either low-active or high-active.

In this mode, timer 1 cannot be used.

Timer 0 outputs pulse to TO1 pin (Also used as P71).

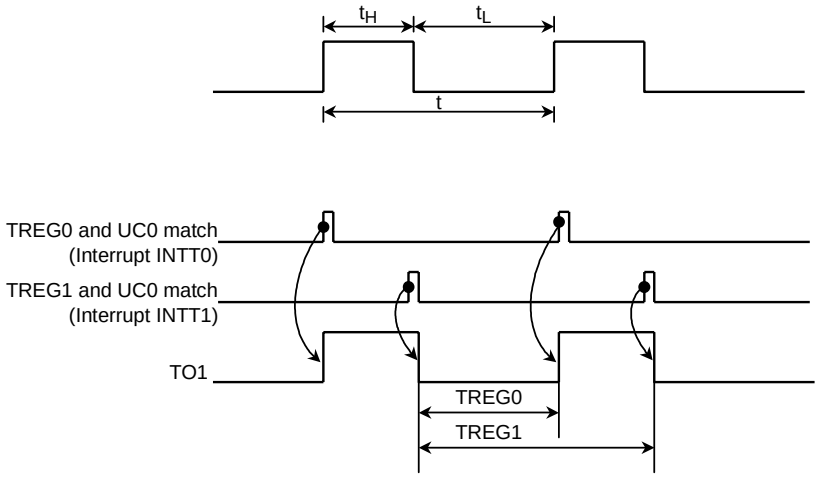


Figure 3.7.10 8-Bit PPG Output Waveforms

In this mode, a programmable square wave is generated by inverting timer output each time the 8-bit up counter (UC0) matches the timer registers TREG0 and TREG1.

However, it is required that the set value of TREG0 is smaller than that of TREG1.

Though the up counter (UC1) of timer 1 is not used in this mode, UC1 should be set for counting by setting TRUN<T1RUN> to 1.

Figure 3.7.11 shows the block diagram for this mode.

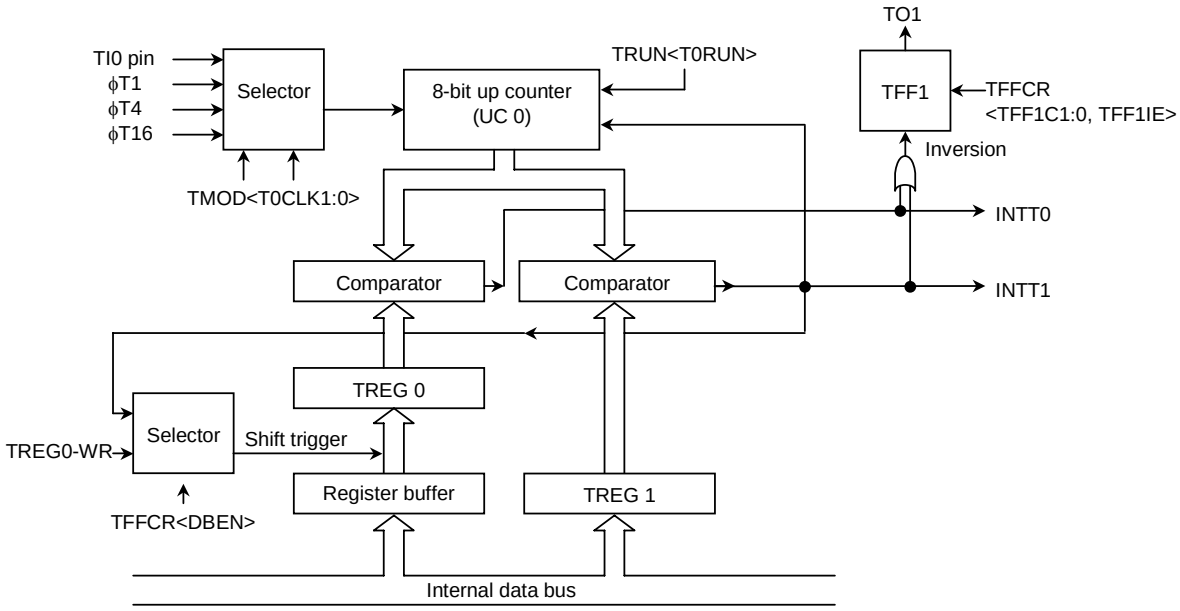


Figure 3.7.11 Block Diagram of 8-Bit PPG Output Mode

When the double buffer of TREG0 is enabled in this mode, the value of register buffer will be shifted in TREG0 each time TREG1 matches UC0.

Use of the double buffer makes the handling of low duty waves easily (when duty is varied).

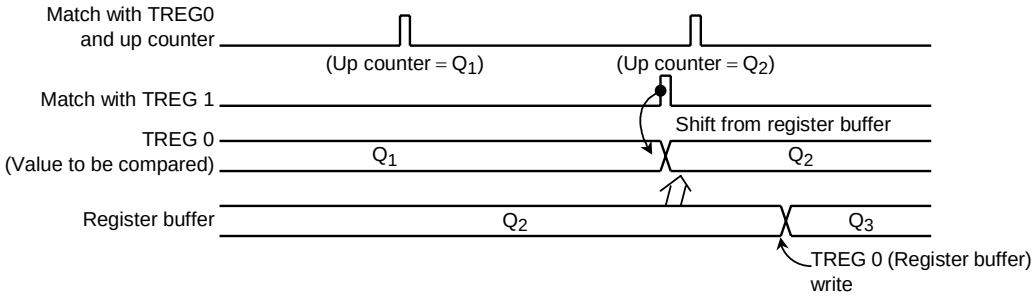
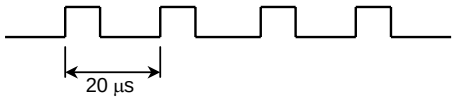


Figure 3.7.12 Operation of Register Buffer

Example: Generating 1/5 duty 50 kHz pulse (at fc = 20 MHz)



* Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: f_{FPH}

Calculate the value to be set for timer register.

To obtain the frequency 50 kHz, the pulse cycle t should be: t = 1/50 kHz = 20 μs.

Given ϕT1 = 0.4 μs (at 20 MHz),

20 μs ÷ 0.4 μs = 50

Consequently, to set the timer register 1 (TREG1) to TREG1 = 50 = 32H

and then duty to 1/5, t × 1/5 = 20 μs × 1/5 = 4 μs

4 μs ÷ 0.4 μs = 10

Therefore, set timer register 0 to TREG0 = 10 = 0AH.

	7	6	5	4	3	2	1	0	
TRUN	← –	X	–	–	–	–	0	0	Stop timer 0, and clear it to “0”.
TMOD	← 1	0	X	X	X	X	0	1	Set the 8-bit PPG mode, and select ϕT1 as input clock.
TREG0	← 0	0	0	0	1	0	1	0	Write “0AH”.
TREG1	← 0	0	1	1	0	0	1	0	Write “32H”.
TFFCR	← –	–	–	1	0	1	1	X	Sets TFF1 and enable the inversion and double buffer enable. Writing “10” provides negative logic pulse.
P7CR	← X	X	X	X	–	–	1	–	Set P71 as the TO1 pin.
P7FC	← X	X	X	X	–	–	1	X	
TRUN	← 1	X	–	–	–	–	1	1	Start timer 0 and timer 1 counting.

X: Don’t care, –: No change

(4) 8-bit PWM output mode

This mode is valid only for timer 0. In this mode, maximum 8-bit resolution of PWM pulse can be output.

PWM pulse is output to TO1 pin (also used as P71) when using timer 0. Timer 1 can also be used as 8-bit timer.

Timer output is inverted when up counter (UC0) matches the set value of timer register TREG0 or when $2^n - 1$ (n = 6, 7, or 8; specified by TMOD<PWMM1:0>) counter overflow occurs. Up counter UC0 is cleared when $2^n - 1$ counter overflow occurs.

To use this PWM mode, the following conditions must be satisfied.

(Set value of timer register) < (Set value of $2^n - 1$ counter overflow)

(Set value of timer register) ≠ 0

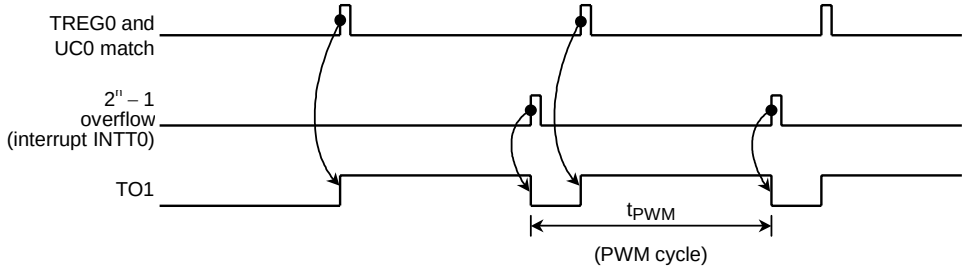


Figure 3.7.13 8-Bit PWM Waveforms

Figure 3.7.14 shows the block diagram of this mode.

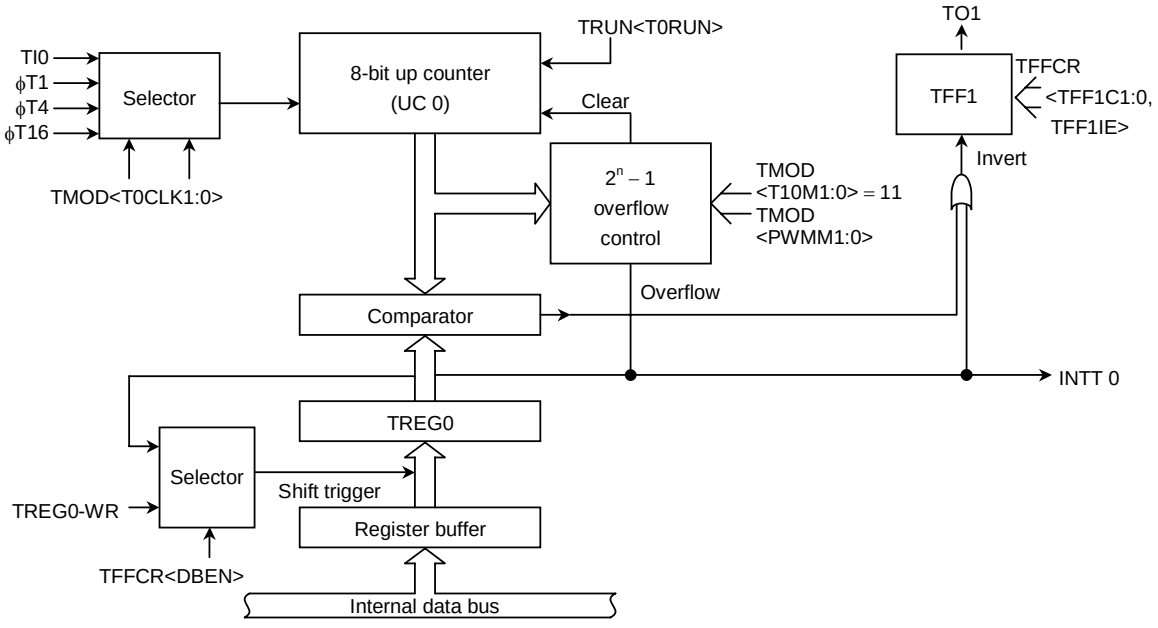


Figure 3.7.14 Block Diagram of 8-Bit PWM Mode

In this mode, the value of register buffer will be shifted in TREG0 if $2^n - 1$ overflow is detected when the double buffer of TREG0 is enabled.

Use of the double buffer makes easy the handling of small duty waves.

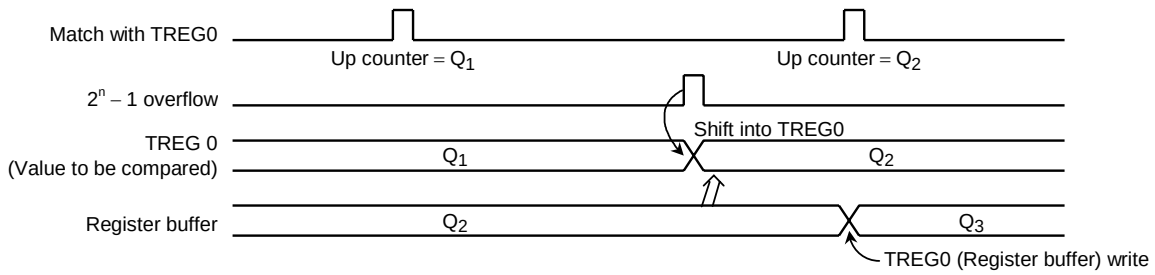
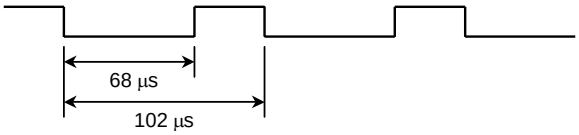


Figure 3.7.15 Operation of Register Buffer

Example: To output the following PWM waves to TO1 pin at fc = 20 MHz.



* Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: f_{FPH}

To realize 102 μs of PWM cycle by ϕT1 = 0.4 μs (at fc = 20 MHz),

102 μs ÷ 0.4 μs = 255 = $2^n - 1$

Consequently, n should be set to 8.

As the period of low level is 68 μs, for ϕT1 = 0.4 μs, set the following value for TREG0.

68 μs ÷ 0.4 μs = 170 = AAH

	MSB 7	6	5	4	3	2	1	LSB 0	
TRUN	← –	X	–	–	–	–	–	0	Stop timer 0, and clear it to "0".
TMOD	← 1	1	1	0	–	–	0	1	Set 8-bit PWM mode (cycle: $2^8 - 1$) and select ϕT1 as the input clock.
TREG0	← 1	0	1	0	1	0	1	0	Writes "AAH".
TFFCR	← X	X	X	X	1	0	1	X	Clears TFF1, enable the inversion and double buffer.
P7CR	← X	X	X	X	–	–	1	–	Set P71 as the TO1 pin.
P7FC	← X	X	X	X	–	–	1	X	
TRUN	← 1	X	–	–	–	–	–	1	Start timer 0 counting.

X: Don't care, –: No change

Table 3.7.2 PWM Cycle

at fc = 20 MHz, fs = 32.768 kHz

Select Prescaler Clock <SYSCK>	Select System Clock <PRCK1:0>	Gear Value <GEAR2:0>	PWM Cycle								
			$2^6 - 1$			$2^7 - 1$			$2^8 - 1$		
			φT1	φT4	φT16	φT1	φT4	φT16	φT1	φT4	φT16
1 (fs)	00 (f_{FPH})	XXX	15.4 ms	61.5 ms	246 ms	31.0 ms	124 ms	496 ms	62.3 ms	249 ms	996 ms
0 (fc)		000 (fc)	25.2 μs	100.8 μs	403.2 μs	50.8 μs	203.2 μs	812.8 μs	102.0 μs	408.0 μs	1.63 ms
		001 (fc/2)	50.4 μs	201.6 μs	806.4 μs	101.6 μs	406.4 μs	1.63 ms	204.0 μs	816.0 μs	3.26 ms
		010 (fc/4)	100.8 μs	403.2 μs	1.61 ms	203.2 μs	812.8 μs	3.26 ms	408.0 μs	1.63 ms	6.53 ms
		011 (fc/8)	201.6 μs	806.4 μs	3.23 ms	406.4 μs	1.63 ms	6.52 ms	816.0 μs	3.26 ms	13.06 ms
		100 (fc/16)	403.2 μs	1.61 ms	6.45 ms	812.8 μs	3.25 ms	13.04 ms	1.63 ms	6.53 ms	26.11 ms
XXX	01 (Low-frequency clock)	XXX	15.4 ms	61.5 ms	246 ms	31.0 ms	124 ms	496 ms	62.3 ms	249 ms	996 ms
XXX	10 (fc/16 clock)	XXX	403.2 μs	1.61 ms	6.45 ms	812.8 μs	3.25 ms	13.04 ms	1.63 ms	6.53 ms	26.11 ms

XXX: Don't care

(5) Table 3.7.3 shows the list of 8-bit timer modes.

Table 3.7.3 Timer Mode Setting Registers

Register Name	TMOD				TFFCR
Name of Function in Register	T10M	PWMM	T1CLK	T0CLK	TFF1IS
Function	Timer Mode	PWM0 Cycle	Upper Timer Input Clock	Lower Timer Input Clock	Timer F/F Invert Signal Select
16-bit timer mode	01	–	–	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	–
8-bit timer × 2 channels	00	–	Lower timer match: φT1, φT16, φT256 (00, 01, 10, 11)	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	0: Lower timer output 1: Upper timer output
8-bit PPG × 1 channel	10	–	–	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	–
8-bit PWM × 1 channel	11	$2^6 - 1$, $2^7 - 1$, $2^8 - 1$ (01, 10, 11)	–	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	–
8-bit timer × 1 channel	11	–	φT1, φT16, φT256 (01, 10, 11)	–	Output disabled

–: Don't care

3.8 8-Bit PWM Timers

The TMP93CW46A has two built-in 8-bit PWM timers (Timers 2 and 3). They have two operating modes.

- 8-bit PWM (Pulse width modulation: variable duty at fixed interval) output mode
- 8-bit interval timer mode

Figure 3.8.1, 3.8.2 are block diagram of 8-bit PWM timer (Timers 2 and 3).

PWM timers consist of an 8-bit up counter, 8-bit comparator, and 8-bit timer register. Two timer flip-flops (TFF2 for timer 2 and TFF3 for timer 3) are provided.

Input clocks ϕP1, ϕP4, and ϕP16 for the PWM timers can be obtained using the built-in prescaler.

PWM timer operating mode and timer flip-flops are controlled by four control registers (P0MOD, P1MOD, PFFCR, and TRUN).

PWM timer 0 and 1 can be used independently.

All PWM timers operate in the same manner, and thus only the operation of PWM timer 0 will be explained below.

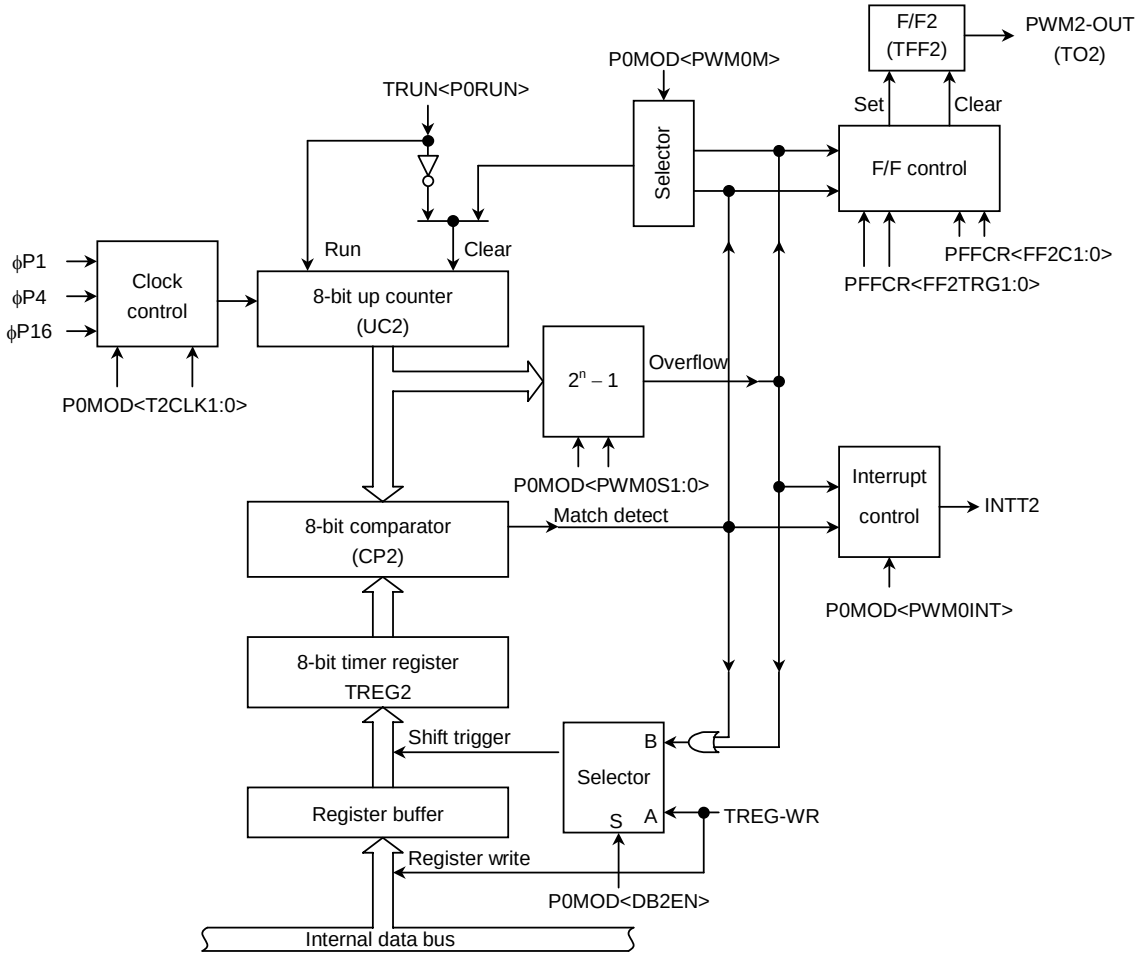


Figure 3.8.1 Block Diagram of 8-Bit PWM Timer 0 (Timer 2)

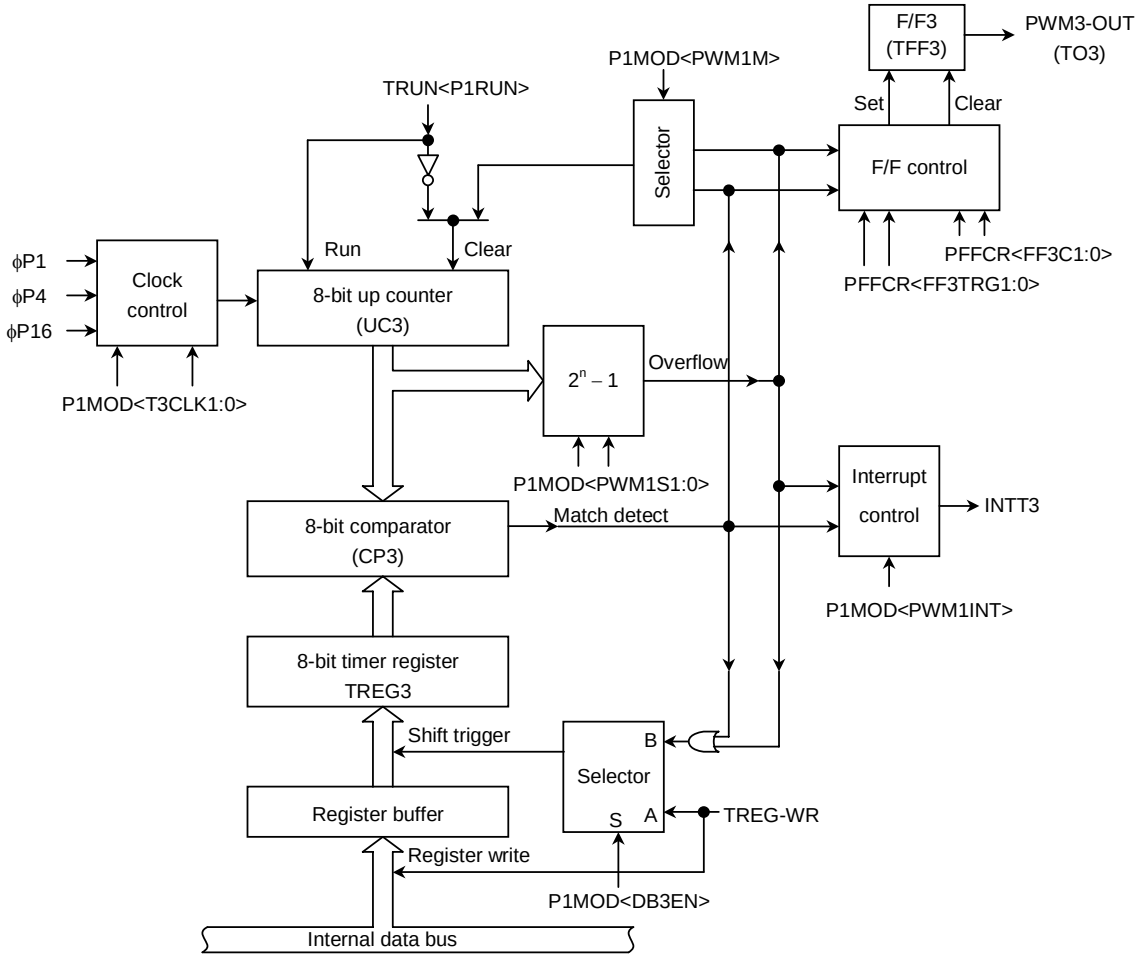


Figure 3.8.2 Block Diagram of 8-Bit PWM Timer 1 (Timer 3)

1. Prescaler, prescaler clock select

There are 5-bit prescaler and prescaler clock selection register to generate input clock for 8-bit PWM Timers 0 and 1.

Figure 3.8.3 shows the block diagram. Table 3.8.1 shows prescaler clock resolution into 8-bit PWM timers 0 and 1.

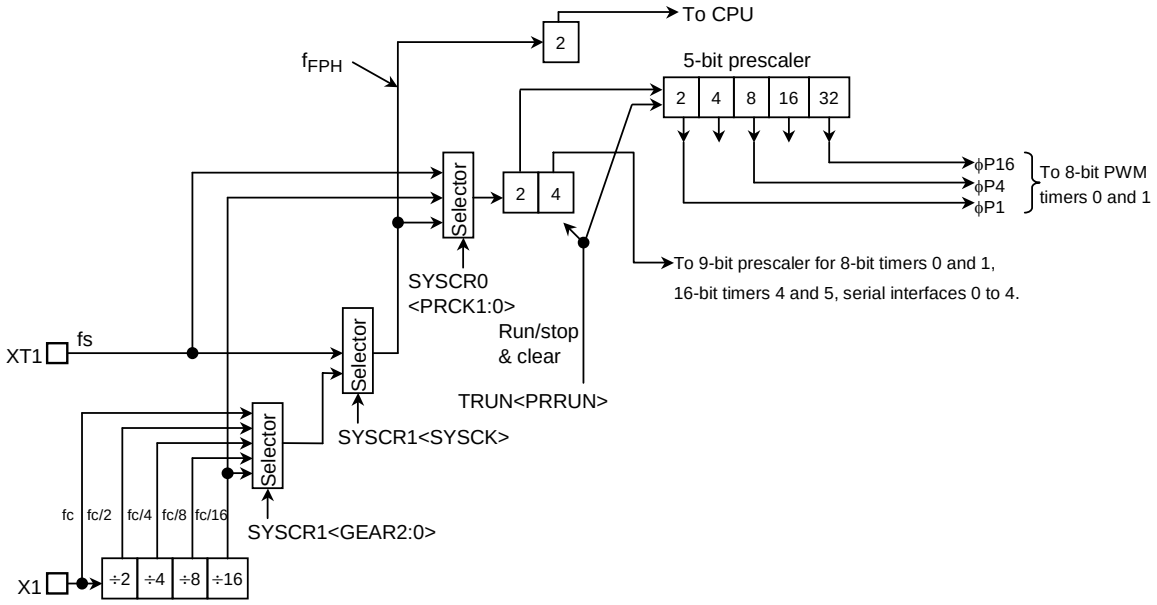


Figure 3.8.3 The Block Diagram of Prescaler

Table 3.8.1 Prescaler Clock Resolution to 8-Bit PWM Timers 0 and 1

at fc = 20 MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	Prescaler Clock Resolution		
			φP1	φP4	φP16
1 (fs)	00 (f_{FPH})	XXX	$fs/2^2$ (122 μs)	$fs/2^4$ (488 μs)	$fs/2^6$ (1.95 ms)
0 (fc)		000 (fc)	$fc/2^2$ (0.2 μs)	$fc/2^4$ (0.8 μs)	$fc/2^6$ (3.2 μs)
		001 (fc/2)	$fc/2^3$ (0.4 μs)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)
		010 (fc/4)	$fc/2^4$ (0.8 μs)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)
		011 (fc/8)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)
		100 (fc/16)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{10}$ (51.2 μs)
XXX	01 (Low-frequency clock)	XXX	$fs/2^2$ (122 μs)	$fs/2^4$ (488 μs)	$fs/2^6$ (1.95 ms)
XXX	10 (fc/16 clock)	XXX	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{10}$ (51.2 μs)

XXX: Don't care

Note: The fc/16 clock as a prescaler clock can not be used when the fs is used as a system clock.

The clock selected among f_{FPH} clock, fc/16 clock and fs is divided by 2 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0<PRCK1:0>.

Resetting sets <PRCK1:0> to "00" selects the f_{FPH} clock input divided by 2. The register TRUN<PRRUN> which controls this prescaler is also used as the other timers. So, this prescaler can not be operated independently.

The 8-bit PWM timer0, 1 selects between 3 clock inputs, ϕP1, ϕP4, and ϕP16 among the prescaler outputs.

This prescaler also can be run or stopped by TRUN<PRRUN> described of the 8-bit timer. Counting starts when <PRRUN> is set to "1". The prescaler is cleared zero and stops operation when <PRRUN> is set to "0". Resetting clear <PRRUN> to "0" and stops the prescaler.

When the IDLE1 mode (Only the oscillator operates) is used, set TRUN<PRRUN> to "0" to reduce the power consumption of the prescaler before the "HALT" instruction is executed.

2. Up counter

The up counter is an 8-bit binary counter which counts up according to the input clock specified by PWM mode register P0MOD<T2CLK1:0>.

The input clock for the up counter is selected from the internal clocks ϕP1, ϕP4, and ϕP16 (PWM dedicated prescaler output) depending on the <T2CLK1:0>.

Operating mode is set by P0MOD<PWM0M>. At reset, it is initialized to "0", thus, the up counter is placed in PWM mode. In PWM mode, the up counter is cleared when a $2^n - 1$ overflow occurs; in timer mode, the up counter is cleared at compare and match.

Count/stop and clear of the up counter can be controlled for each PWM timer using the timer operation control register TRUN. Resetting clears all up counters and stops timers.

3. Timer register

The 8-bit register is used for setting an interval time. When the value set in the timer register (TREG2) matches the value in the up counter, the match detect signal of the comparator becomes active.

Timer register TREG2 is each paired with register buffer to make a double buffer structure.

TREG2 controls double buffer enable/disable by P0MOD<DB2EN>: Disabled when <DB2EN> = 0, enabled when <DB2EN> = 1.

Data is transferred from register buffer to timer register when a $2^n - 1$ overflow occurs in PWM mode, or when compare and match occur in 8-bit timer mode. That is, with a PWM timer, the timer mode can be operated in double buffer enable state, unlike PWM mode and timer mode for timers 0 and 1.

At reset, <DB2EN> is initialized to 0 to disable double buffer. The same data value is written to both the register buffer and the timing register. To use double buffer, write the data in the timer register at first, then set <DB2EN> to 1, and write the following data in the register buffer.

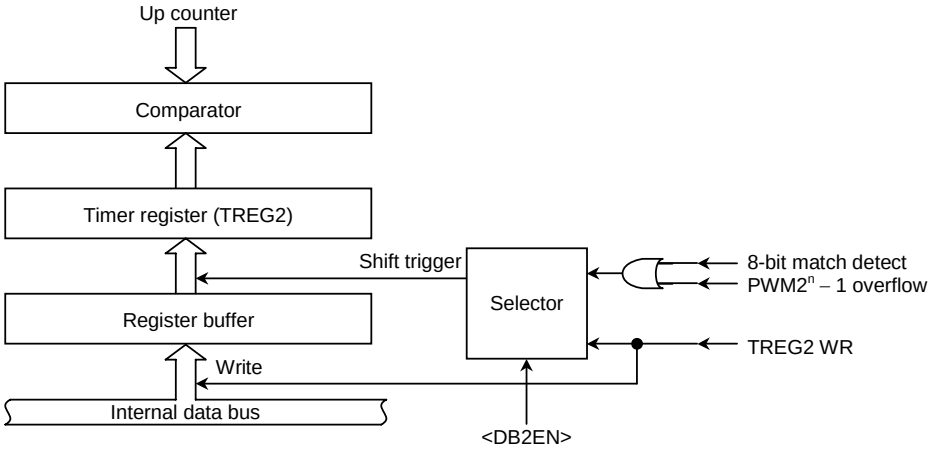


Figure 3.8.4 Structure of Timer Registers 2

Memory addresses of the timer registers are as follows:

TREG2: 000026H (for PWM0)

TREG3: 000027H (for PWM1)

The timer register and the register buffer are allocated to the same memory address. When <DB2EN> = "0", the same value is written to both register buffer and timer register. When <DB2EN> = "1", the value is written to the register buffer only.

Register buffer values can be read when reading the above addresses. The timer register is only writing, and it can not read.

4. Comparator

Compares the value in the up counter with the value in the timer register (TREG2). When they match, the comparator outputs the match detect signal. In timer mode, the comparator clears the up counter to 0 at compare and match. It also inverts the value of the timer flip-flop if timer flip-flop invert is enabled.

5. Timer flip-flop

The value of the timer flip-flop is inverted by the match detect signal (Comparator output) of each interval timer or $2^n - 1$ overflow. The value can be output to the timer output pin TO2 (Also used as P72).

P0MOD (0028H)

	7	6	5	4	3	2	1	0
Bit symbol	FF2RD	DB2EN	PWM0INT	PWM0M	T2CLK1	T2CLK0	PWM0S1	PWM0S0
Read/Write	R	W						
After reset	–	0	0	0	0	0	0	0
Function	Flip-flop (F/F2) output data	1: Double buffer 2 enable	0: $2^n - 1$ overflow interrupt 1: Compare and match interrupt	0: PWM mode 1: Timer mode	00: φP1 01: φP4 10: φP16 11: Don't care		00: $2^6 - 1$ 01: $2^7 - 1$ 10: $2^8 - 1$ 11: Don't care	

Select PWM0 cycle

00	$2^6 - 1$
01	$2^7 - 1$
10	$2^8 - 1$
11	Don't care

Select PWM0 input clock

00	φP1
01	φP4
10	φP16
11	Don't care

Select PWM0 mode

0	PWM mode
1	8-bit timer mode

Select PWM0 interrupt

0	Overflow interrupt
1	Compare and match interrupt

Control double buffer

0	Disable
1	Enable

PWM timer flip-flop2 (TFF2) output value (TO2)

Note: Read-modify-write is prohibited.

Figure 3.8.5 8-Bit PWM0 Mode Control Register

P1MOD (0029H)

	7	6	5	4	3	2	1	0
Bit symbol	FF3RD	DB3EN	PWM1INT	PWM1M	T3CLK1	T3CLK0	PWM1S1	PWM1S0
Read/Write	R	W						
After reset	–	0	0	0	0	0	0	0
Function	Flip-flop (F/F3) output data	1: Double buffer 3 enable	0: $2^n - 1$ overflow interrupt 1: Compare and match interrupt	0: PWM mode 1: Timer mode	00: φP1 01: φP4 10: φP16 11: Don't care		00: $2^6 - 1$ 01: $2^7 - 1$ 10: $2^8 - 1$ 11: Don't care	

Select PWM1 cycle

00	$2^6 - 1$
01	$2^7 - 1$
10	$2^8 - 1$
11	Don't care

Select PWM1 input clock

00	φP1
01	φP4
10	φP16
11	Don't care

Select PWM1 mode

0	PWM mode
1	8-bit timer mode

Select PWM1 interrupt

0	Overflow interrupt
1	Compare and match interrupt

Control double buffer

0	Disable
1	Enable

PWM timer flip-flop3 (TFF3) output value (TO3)

Note: Read-modify-write is prohibited.

Figure 3.8.6 8-Bit PWM1 Mode Control Register

PFFCR (002AH)

	7	6	5	4	3	2	1	0
Bit symbol	FF3C1	FF3C0	FF3TRG1	FF3TRG0	FF2C1	FF2C0	FF2TRG1	FF2TRG0
Read/Write	W		R/W		W		R/W	
After reset	1	1	0	0	1	1	0	0
Function	00: Don't care 01: Set TFF3 10: Clear TFF3 11: Don't care		00: Disable TFF3 inverted. 01: Invert by match. 10: Set by match; clear by overflow. 11: Clear by match; set by overflow.		00: Don't care 01: Set TFF2 10: Clear TFF2 11: Don't care		00: Disable TFF2 inverted. 01: Invert by match. 10: Set by match; clear by overflow. 11: Clear by match; set by overflow.	

Select PWM timer F/F2 (TFF2) trigger

00	Disable TFF2 trigger.
01	Invert by compare and match.
10	Set by compare and match. Clear by $2^n - 1$ overflow.
11	Clear by compare and match. Set by $2^n - 1$ overflow.

Control PWM timer F/F2 (TFF2)

00	Don't care
01	Set TFF2 to "1".
10	Clear TFF2 to "0".
11	Don't care

Select PWM timer F/F3 (TFF3) trigger

00	Disable TFF3 trigger.
01	Invert by compare and match.
10	Set by compare and match. Clear by $2^n - 1$ overflow.
11	Clear by compare and match. Set by $2^n - 1$ overflow.

Control PWM timer F/F3 (TFF3)

00	Don't care
01	Set TFF3 to "1".
10	Clear TFF3 to "0".
11	Don't care

Figure 3.8.7 8-Bit PWM F/F Control Register

TRUN (0020H)

	7	6	5	4	3	2	1	0
Bit symbol	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Prescaler and timer run/stop control 0: Stop and clear 1: Run (Count up)							

→ Count operation

0	Stop and clear
1	Count

PRRUN: Operation of prescaler

T5RUN: Operation of 16-bit timer (Timer 5)

T4RUN: Operation of 16-bit timer (Timer 4)

P1RUN: Operation of PWM timer (PWM1/timer 3)

P0RUN: Operation of PWM timer (PWM0/timer 2)

T1RUN: Operation of 8-bit timer (Timer 1)

T0RUN: Operation of 8-bit timer (Timer 0)

SYSCR0 (006EH)

	7	6	5	4	3	2	1	0
Bit symbol	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
Read/Write	R/W							
After reset	1	0	1	0	0	0	0	0
Function	High-frequency oscillator (fc) 0: Stop 1: Oscillator	Low-frequency oscillator (fs) 0: Stop 1: Oscillator	High-frequency oscillator (fc) after released STOP mode 0: Stop 1: Oscillator	Low-frequency oscillator (fs) after released STOP mode 0: Stop 1: Oscillator	Select clock after released STOP mode 0: fc 1: fs	Warm-up timer (Write) 0: Don't care 1: Start timer (Read) 0: End warm up 1: Not end warm up	Select prescaler clock 00: f_{FPH} 01: fs 10: fc/16 11: (Reserved)	

Figure 3.8.8 Timer Operation Control Register/System Clock Control Register

The following explains PWM timer operations.

(1) PWM timer mode

PWM output changes under the following two conditions.

Condition 1:

- TFF2 is cleared to 0 when the value in the up counter (UC2) and the value set in the TREG2 match.
- TFF2 is set to 1 when a $2^n - 1$ counter overflow (n = 6, 7, or 8) occurs.

Condition 2:

- TFF2 is set to 1 when the value in the up counter (UC2) and the value set in TREG2 match.
- TFF2 is cleared to 0 when a $2^n - 1$ counter overflow (n = 6, 7, or 8) occurs.

The up counter (UC2) is cleared by a $2^n - 1$ counter overflow.

The PWM timer can output 0% to 100% duty pulses because a $2^n - 1$ counter overflow has a higher priority. That is, to obtain 0% output (Always low), the mode used to set TFF2 to TFF0 due to overflow (PFFCR<FF2TRG1:0> = 1, 0) must be set and $2^n - 1$ (Value for overflow) must be set in TREG2. To obtain 100% output (Always high), the mode must be changed: PFFCR<FF2TRG1:0> = 1, 1 then the same operation is required.

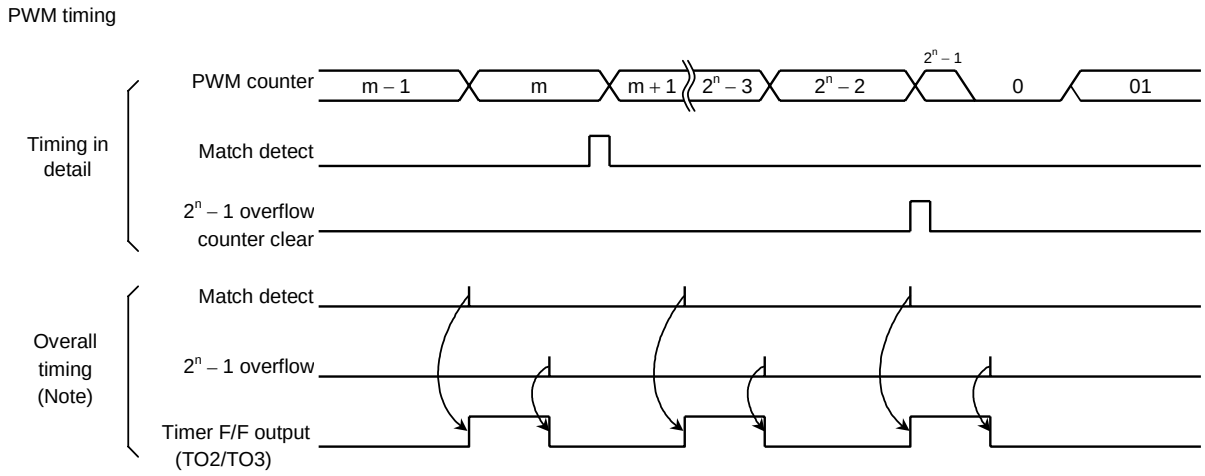


Figure 3.8.9 Output Waves in PWM Timer Mode

Note: The above waves are obtained in a mode where the F/F is set by a match with the timer register (TREG) and reset by an overflow.

Figure 3.8.10 is a block diagram of this mode.

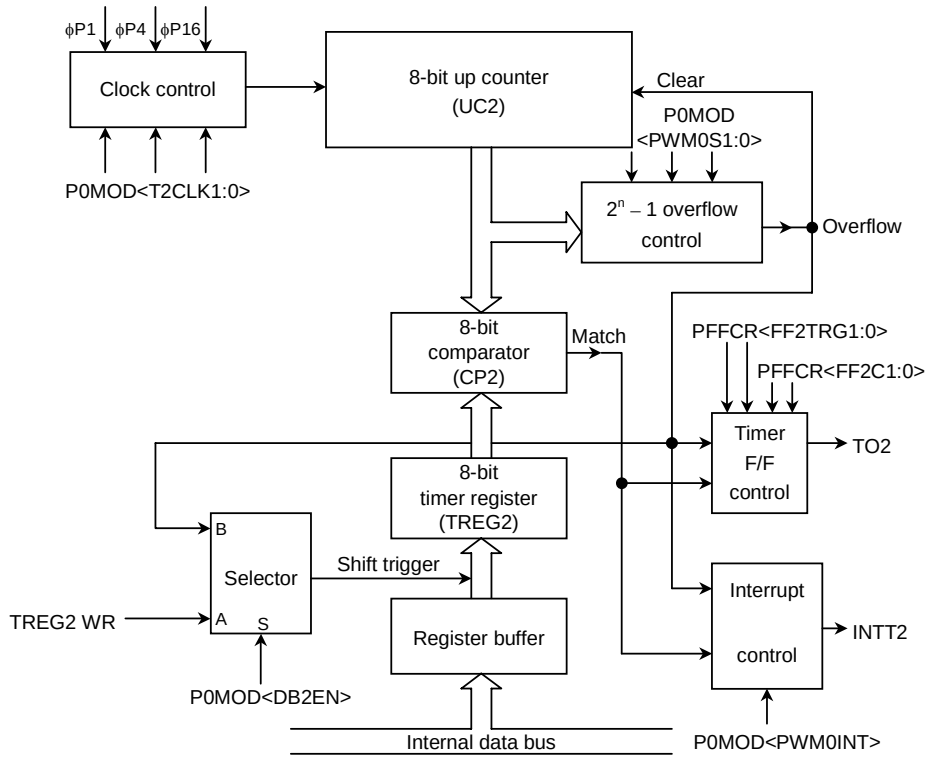


Figure 3.8.10 Block Diagram of PWM Timer Mode (PWM0)

In this mode, enabling double buffer is very useful. The register buffer value shifts into TREG2 when a $2^n - 1$ overflow is detected, when double buffer is enabled.

Using double buffer makes handling small duty waves easily.

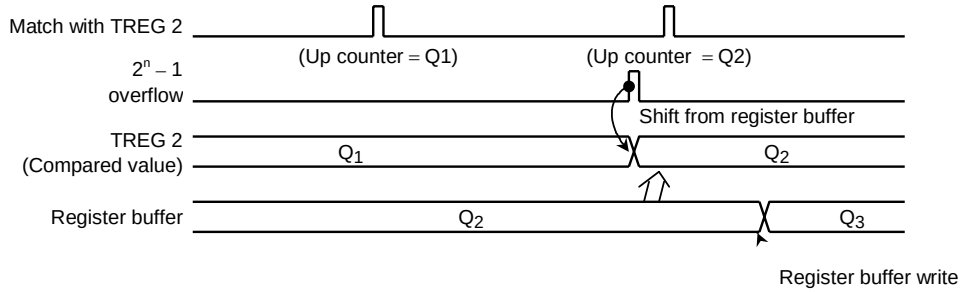
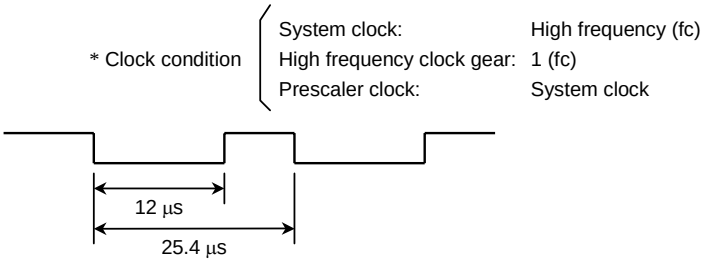


Figure 3.8.11 Register Buffer Operation

Example: To output the following PWM waves to TO2 pin using PWM0 at fc = 20 MHz



To implement 25.4 μs PWM cycle by φP1 = 0.2 μs (at fc = 20 MHz)

25.4 μs ÷ 0.2 μs = 127 = 2^7 – 1.

Consequently, set n to 7.

Since the low level cycle = 12 μs; for φP1 = 0.2 μs

12 μs ÷ 0.2 = 60 = 3CH

set the 3CH in TREG2.

	7	6	5	4	3	2	1	0	
TRUN ←	–	X	–	–	–	0	–	–	Stops PWM0 and clears it to 0.
P0MOD ←	–	0	0	0	0	0	0	1	Sets PWM (2^7 – 1) mode, input clock φP1, overflow interrupt, and disables double buffer.
TREG2 ←	0	0	1	1	1	1	0	0	Writes 3CH.
P0MOD ←	–	1	0	0	0	0	0	1	Enables double buffer.
PFFCR ←	–	–	–	–	0	1	1	0	Sets TFF2 and a mode where TFF2 is set by compare and match, and cleared by overflow.
P7CR ←	X	X	X	X	–	1	–	–	Sets P72 as TO2 pin
P7FC ←	X	X	X	X	–	1	–	X	
TRUN ←	1	X	–	–	–	1	–	–	Starts PWM0 counting.

X: Don't care, –: No change

Table 3.8.2 PWM Cycle

at fc = 20 MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	PWM Cycle								
			2^6 – 1			2^7 – 1			2^8 – 1		
			φP1	φP4	φP16	φP1	φP4	φP16	φP1	φP4	φP16
1 (fs)	00 (f_{FPH})	XXX	7.69 ms	30.8 ms	123 ms	15.5 ms	62.0 ms	248 ms	31.1 ms	125 ms	498 ms
0 (fc)		000 (fc)	12.6 μs	50.4 μs	201.6 μs	25.4 μs	101.6 μs	406.4 μs	51.0 μs	204.0 μs	816.0 μs
		001 (fc/2)	25.2 μs	100.8 μs	403.2 μs	50.8 μs	203.2 μs	812.8 μs	102.0 μs	408.0 μs	1.63 ms
		010 (fc/4)	50.4 μs	201.6 μs	806.4 μs	101.6 μs	406.4 μs	1.63 ms	204.0 μs	816.0 μs	3.26 ms
		011 (fc/8)	100.8 μs	403.2 μs	1.61 ms	203.2 μs	812.8 μs	3.25 ms	408.0 μs	1.63 ms	6.53 ms
		100 (fc/16)	201.6 μs	806.4 μs	3.23 ms	406.4 μs	1.63 ms	6.50 ms	816.0 μs	3.26 ms	13.06 ms
XXX	01 (Low frequency)	XXX	7.69 ms	30.8 ms	123 ms	15.5 ms	62.0 ms	248 ms	31.1 ms	125 ms	498 ms
XXX	10 (fc/16 clock)	XXX	201.6 μs	806.4 μs	3.23 ms	406.4 μs	1.63 ms	6.50 ms	816.0 μs	3.26 ms	13.06 ms

XXX: Don't care

(2) 8-bit timer mode

Both PWM timers can be used independently as 8-bit interval timers. Since both timers operate in exactly the same way, PWM0 (Timer 2) is used for the purposes of explanation.

1. Generating interrupts at a fixed interval

To generate timer 2 interrupts (INTT2) at a fixed interval using PWM0 timer, first stop PWM0, then set the operating mode, input clock, and interval in the P0MOD and TREG2 registers. Next, enable INTT2 and start counting PWM0.

Example: To generate a timer 2 interrupt every 40 μs at fc = 20 MHz, set registers as follows:

* Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: System clock

	7	6	5	4	3	2	1	0	
TRUN	← –	X	–	–	–	0	–	–	Stops PWM timer 0 and clears it to 0.
P0MOD	← X	0	1	1	0	0	X	X	Sets 8-bit timer mode and selects φP1 (0.2 μs) and compare interrupt.
TREG2	← 1	1	0	0	1	0	0	0	Sets 40 μs ÷ 0.2 μs = C8H in timer register.
INTEPW10	← –	–	–	–	1	1	0	0	Enables INTT2 and sets interrupt level 4.
TRUN	← 1	X	–	–	–	1	–	–	Starts counting PWM0.

X: Don't care, –: No change

Select an input clock using the Table 3.8.1.

Note: To generate interrupts in 8-bit timer mode, P0MOD<PWM0INT> must be set to 1.

2. Generating a 50% square wave

To generate a 50% square wave, invert the timer flip-flop at a fixed interval and output the timer flip-flop value to the timer output pin (TO2).

Example: To output a 2.0 μs square wave at fc = 20 MHz from TO2 pin, set registers as follows.

* Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: System clock

	7	6	5	4	3	2	1	0	
TRUN	← –	X	–	–	–	0	–	–	Stops PWM0 and clears it to 0.
P0MOD	← X	0	1	1	0	0	X	X	Sets 8-bit timer mode and selects φP1 (0.2 μs) as the input clock.
TREG2	← 0	0	0	0	0	1	0	1	Sets 2.0 μs ÷ 0.2 μs ÷ 2 = 5 in the timer register.
PFFCR	← –	–	–	–	1	0	0	1	Clears TFF2 to TFF0 and inverts using comparator output.
P7CR	← X	X	X	X	–	1	–	–	Sets P72 as TO2 pin.
P7FC	← X	X	X	X	–	1	–	X	
TRUN	← 1	X	–	–	–	1	–	–	Starts counting PWM0.

X: Don't care, –: No change

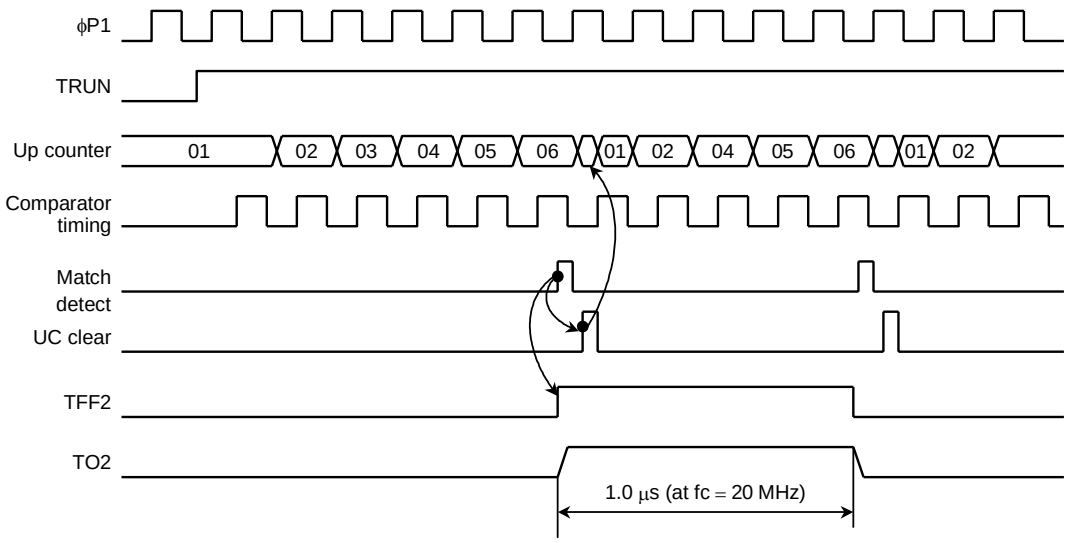


Figure 3.8.12 Square Wave (50% duty) Output Timing Chart

This mode is as shown in Figure 3.8.13 below.

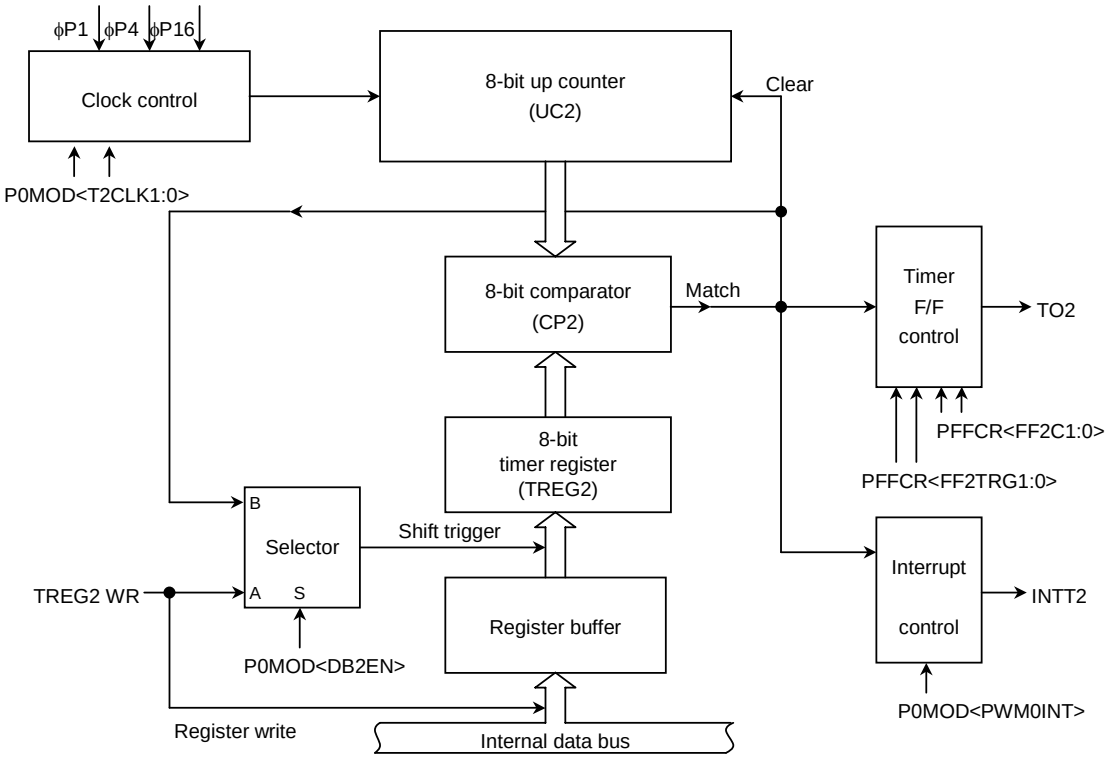


Figure 3.8.13 Block Diagram of 8-Bit Timer Mode

3.9 16-Bit Timers

The TMP93CW46A contains two (Timer 4 and timer 5) multifunctional 16-bit timer/event counters with the following operation modes.

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) mode
- Frequency measurement mode
- Pulse width measurement mode
- Time differential measurement mode

Timer/event counter consists of 16-bit up counter, two 16-bit timer registers (One of them applies double buffer), two 16-bit capture registers, two comparators, capture input controller, and timer flip-flop and the control circuit.

Timer/event counter is controlled by 4 control registers: T4MOD/T5MOD, T4FFCR/T5FFCR, TRUN and T45CR.

Figure 3.9.1, 3.9.2 show the block diagram of 16-bit timer/event counter (Timer 4 and timer 5).

Timer 4 and 5 can be used independently.

All timer operate in the same manner except the following points, and thus only the operation of timer 4 will be explained below.

(Different points between timer 4 and 5)

	Timer 4	Timer 5
Timer out pin	TO4 pin (TFF4) TO5 pin (TFF5)	TO6 pin (TFF6) no TO7 pin (TFF7)
Different phased pulse output mode	Yes	No (no TO7 pin)

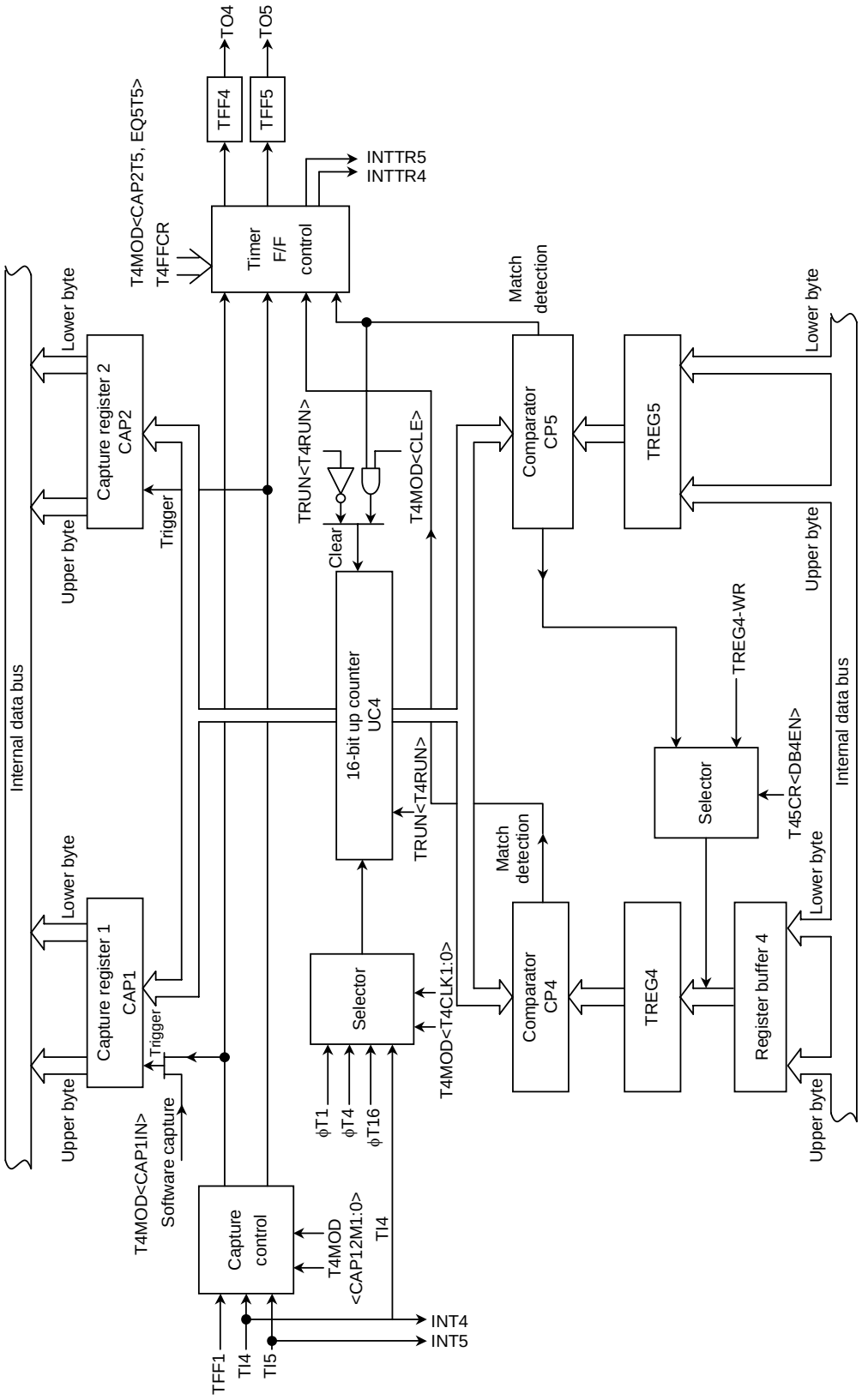


Figure 3.9.1 Block Diagram of 16-Bit Timer (Timer 4)

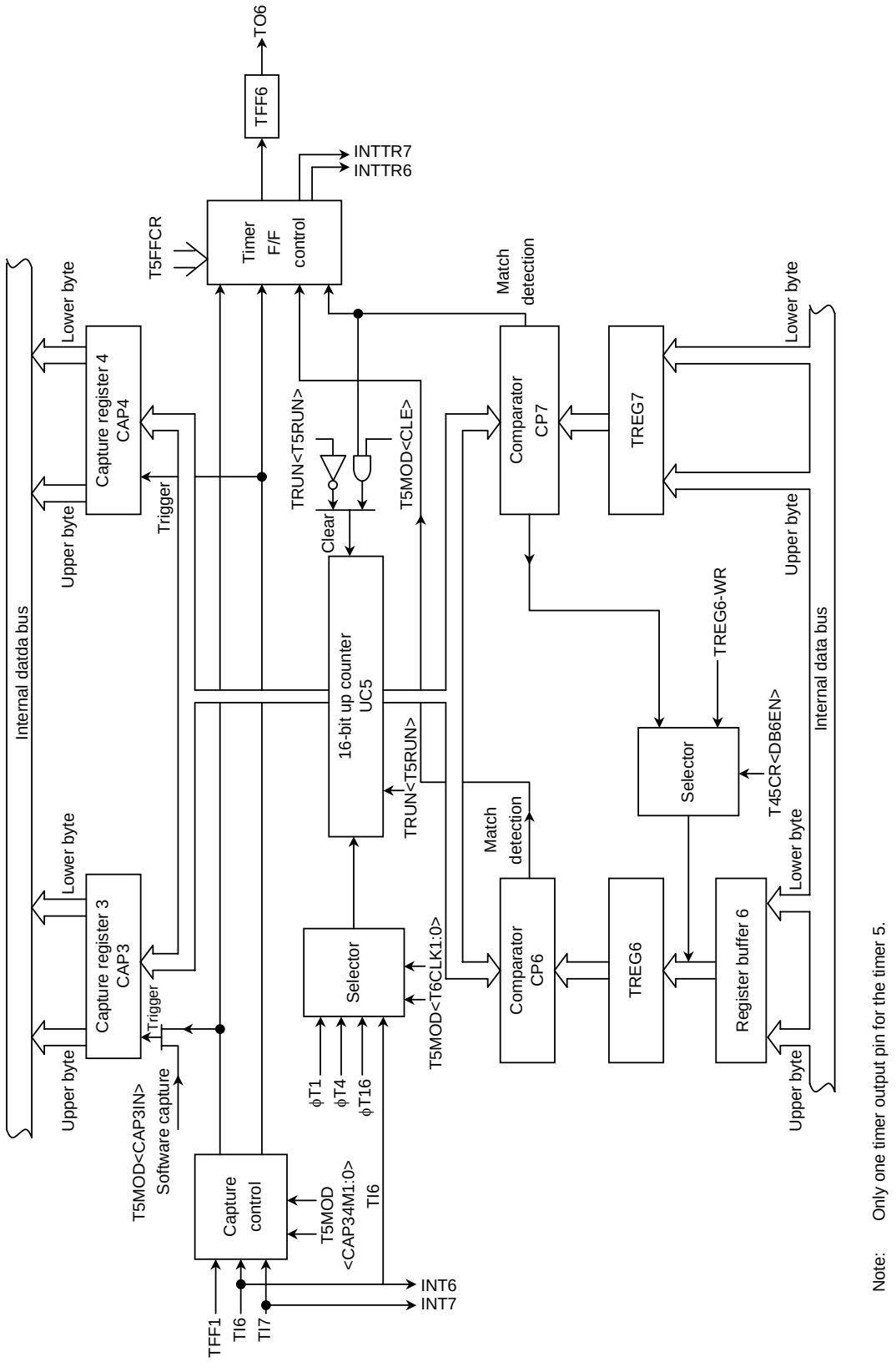


Figure 3.9.2 Block Diagram of 16-Bit Timer (Timer 5)

T4MOD (0038H)

	7	6	5	4	3	2	1	0
Bit symbol	CAP2T5	EQ5T5	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFF5 invert trigger 0: Disable trigger 1: Enable trigger		0: Software capture 1: Don't care	Capture timing 00: Disable INT4 occurs at rise edge. 01: TI4↑ TI5↑ INT4 occurs at rise edge. 10: TI4↑ TI4↓ INT4 occurs at fall edge. 11: TFF1↑ TFF1↓ INT4 occurs at rise edge.		1: UC4 clear enable	Timer 4 source clock 00: TI4 01: ϕT1 10: ϕT4 11: ϕT16	
	Invert when the UC value is loaded to CAP2	Invert when the up counter matches TREG5						

Timer 4 input clock

00	External clock (TI4)
01	ϕT1
10	ϕT4
11	ϕT16

Clearing the up counter UC4

0	Clear disable
1	Clear by match with TREG5.

Figure 3.9.3 16-Bit Timer Mode Controller Register (T4MOD) (1/2)

T4MOD (0038H)

	7	6	5	4	3	2	1	0
Bit symbol	CAP2T5	EQ5T5	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	TFF5 invert trigger 0: Disable trigger 1: Enable trigger Invert when the UC value is loaded to CAP2	Invert when the up counter matches TREG5	0: Software capture 1: Don't care	Capture timing 00: Disable INT4 occurs at rise edge. 01: TI4↑ TI5↑ INT4 occurs at rise edge. 10: TI4↑ TI4↓ INT4 occurs at fall edge. 11: TFF1↑ TFF1↓ INT4 occurs at rise edge.		1: UC4 clear enable	Timer 4 source clock 00: TI4 01: ϕT1 10: ϕT4 11: ϕT16	

Capture timing of timer 4

	Capture control	INT4 control
00	Capture disable	Interrupt occurs at the rise edge of TI4 (INT4) input.
01	CAP1 at TI4 rise CAP2 at TI5 rise	
10	CAP1 at TI4 rise CAP2 at TI4 fall	Interrupt occurs at the fall edge of TI4 (INT4) input.
11	CAP1 at TFF1 rise CAP2 at TFF1 fall	Interrupt occurs at the rise edge of TI4 (INT4) input.

Software capture

0	The up counter 4 value is loaded to CAP1 (Software capture).
1	Always read as "1".

Timer flip-flop 5 (TFF5) invert trigger

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

CAP2T5: Invert when the up counter value is loaded to CAP2
EQ5T5: Invert when the up counter matches TREG5

Figure 3.9.4 16-Bit Timer Controller Register (T4MOD) (2/2)

T4FFCR (0039H)

	7	6	5	4	3	2	1	0
Bit symbol	TFF5C1	TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
Read/Write	W		R/W	R/W	R/W	R/W	W	
After reset	1	1	0	0	0	0	1	1
Function	00: Invert TFF5 01: Set TFF5 10: Clear TFF5 11: Don't care Always read as "11".		TFF4 invert trigger 0: Disable trigger 1: Enable trigger				00: Invert TFF4 01: Set TFF4 10: Clear TFF4 11: Don't care Always read as "11"	
			Invert when the UC value is loaded to CAP2	Invert when the UC value is loaded to CAP1	Invert when the UC matches TREG5	Invert when the UC matches TREG4		

Timer flip-flop 4 (TFF4) control

00	Inverts the TFF4 value (Software inversion).
01	Sets TFF4 to "1".
10	Clear TFF4 to "0".
11	Don't care (Always read as "11").

Timer flip-flop 4 (TFF4) invert trigger

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

CAP2T4: Invert when the up counter value is loaded to CAP2
CAP1T4: Invert when the up counter value is loaded to CAP1
EQ5T4: Invert when up counter matches TREG5
EQ4T4: Invert when up counter matches TREG4

Timer flip-flop 5 (TFF5) control

00	Inverts the TFF5 value (Software inversion).
01	Sets TFF5 to "1".
10	Clear TFF5 to "0".
11	Don't care (Always read as "11".)

Figure 3.9.5 16-Bit Timer 4 F/F Control (T4FFCR)

T5MOD (0048H)

	7	6	5	4	3	2	1	0
Bit symbol			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
Read/Write			W	R/W		R/W	R/W	
After reset			1	0	0	0	0	0
Function			0: Software capture 1: Don't care	Capture timing 00: Disable INT6 occurs at rise edge. 01: TI6↑ TI7↑ INT6 occurs at rise edge. 10: TI6↑ TI6↓ INT6 occurs at fall edge. 11: TFF1↑ TFF1↓ INT6 occurs at rise edge.		1: UC5 clear enable	Timer 5 source clock 00: TI6 01: φT1 10: φT4 11: φT16	

Timer 5 input clock

00	External clock (TI6)
01	φT1
10	φT4
11	φT16

Clearing the up counter UC5

0	Clear disable
1	Clear by match with TREG7

Figure 3.9.6 16-Bit Timer Mode Control Register (T5MOD) (1/2)

T5MOD (0048H)

	7	6	5	4	3	2	1	0
Bit symbol			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
Read/Write			W	R/W		R/W	R/W	
After reset			1	0	0	0	0	0
Function			0: Software capture 1: Don't care	Capture timing 00: Disable INT6 occurs at rise edge. 01: TI6↑ TI7↑ INT6 occurs at rise edge. 10: TI6↑ TI6↓ INT6 occurs at fall edge. 11: TFF1↑ TFF1↓ INT6 occurs at rise edge.		1: UC5 clear enable	Timer 5 source clock 00: TI6 01: φT1 10: φT4 11: φT16	

Timer 5 capture timing

	Capture control	INT6 control
00	Capture disable	Interrupt occurs at the rise edge of TI6 (INT6) input.
01	CAP3 at TI6 rise CAP4 at TI7 rise	
10	CAP3 at TI6 rise CAP4 at TI6 fall	Interrupt occurs at the fall edge of TI6 (INT6) input.
11	CAP3 at TFF1 rise CAP4 at TFF1 fall	Interrupt occurs at the rise edge of TI6 (INT6) input.

Software capture

0	The up counter 5 value is loaded to CAP3.
1	Don't care. (Always read as "1".)

Figure 3.9.7 16-Bit Timer Control Register (T5MOD) (2/2)

T5FFCR (0049H)

	7	6	5	4	3	2	1	0
Bit symbol			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
Read/Write			R/W	R/W	R/W	R/W	W	
After reset			0	0	0	0	1	1
Function			TFF6 invert trigger 0: Disable trigger 1: Enable trigger				00: Invert TFF6 01: Set TFF6 10: Clear TFF6 11: Don't care Always read as "11"	
			Invert when the UC value is loaded to CAP4	Invert when the UC value is loaded to CAP3	Invert when the UC matches TREG7	Invert when the UC matches TREG6		

Timer flip-flop 6 (TFF6) control

00	Inverts the TFF6 value (Software inversion).
01	Sets TFF6 to "1".
10	Clear TFF6 to "0".
11	Don't care. (Always read as "11".)

Timer flip-flop 6 (TFF6) invert trigger

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

CAP4T6: Invert when the up counter value is loaded to CAP4
CAP3T6: Invert when the up counter value is loaded to CAP3
EQ7T6: Invert when up counter matches TREG7
EQ6T6: Invert when up counter matches TREG6

Figure 3.9.8 16-Bit Timer 5 F/F Control (T5FFCR)

T45CR (003AH)

	7	6	5	4	3	2	1	0
Bit symbol	QCU						DB6EN	DB4EN
Read/Write	R/W						R/W	
After reset	0						0	0
Function	Warm-up timer control						Double buffer 0: Disable 1: Enable	
							Double buffer of TREG6	Double buffer of TREG4

Double buffer control

0	Disable
1	Enable

DB6EN: Double buffer to TREG6
DB4EN: Double buffer to TREG4

Warm-up timer input control

0	Use 7-stage binary counter
1	Not Use 7-stage binary counter (Note)

Note 1: In case of unused the 7-stage binary counter as a warm-up timer, the stable clock must be input from external circuit.

Note 2: T45CR<bit6:2> is always read as “1”.

Figure 3.9.9 16-Bit Timer Trigger Control Register (T45CR)

TRUN (0020H)

	7	6	5	4	3	2	1	0
Bit symbol	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Prescaler and timer run/stop control 0: Stop and clear 1: Run (Count up)							

→ Count operation

0	Stop and clear
1	Count

PRRUN: Operation of prescaler
T5RUN: Operation of 16-bit timer (Timer 5)
T4RUN: Operation of 16-bit timer (Timer 4)
P1RUN: Operation of PWM timer (PWM1/timer 3)
P0RUN: Operation of PWM timer (PWM0/timer 2)
T1RUN: Operation of 8-bit timer (Timer 1)
T0RUN: Operation of 8-bit timer (Timer 0)

Note: TRUN<bit6> is always read as “1”.

SYSCR0 (006EH)

	7	6	5	4	3	2	1	0
Bit symbol	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
Read/Write	R/W							
After reset	1	0	1	0	0	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 (fs) after released STOP mode 0: Stop 1: Oscillation	Select clock after released STOP mode 0: fc 1: fs	Warm-up timer (Write) 0: Don’t care 1: Start timer (Read) 0: End warm up 1: Not end warm up	Select prescaler clock 00: f_{FPH} 01: fs 10: fc/16 11: (Reserved)	

Figure 3.9.10 Timer Operation Control Register/System Clock Control Register

1. Prescaler

There are 9-bit prescaler and prescaler clock selection registers to generate input clock for 8-bit timers 0 and 1, 16-bit timers 4 and 5 and serial interfaces 0 to 4. Figure 3.9.11 shows the block diagram. Table 3.9.1 shows prescaler clock resolution into 8-/16-bit timer.

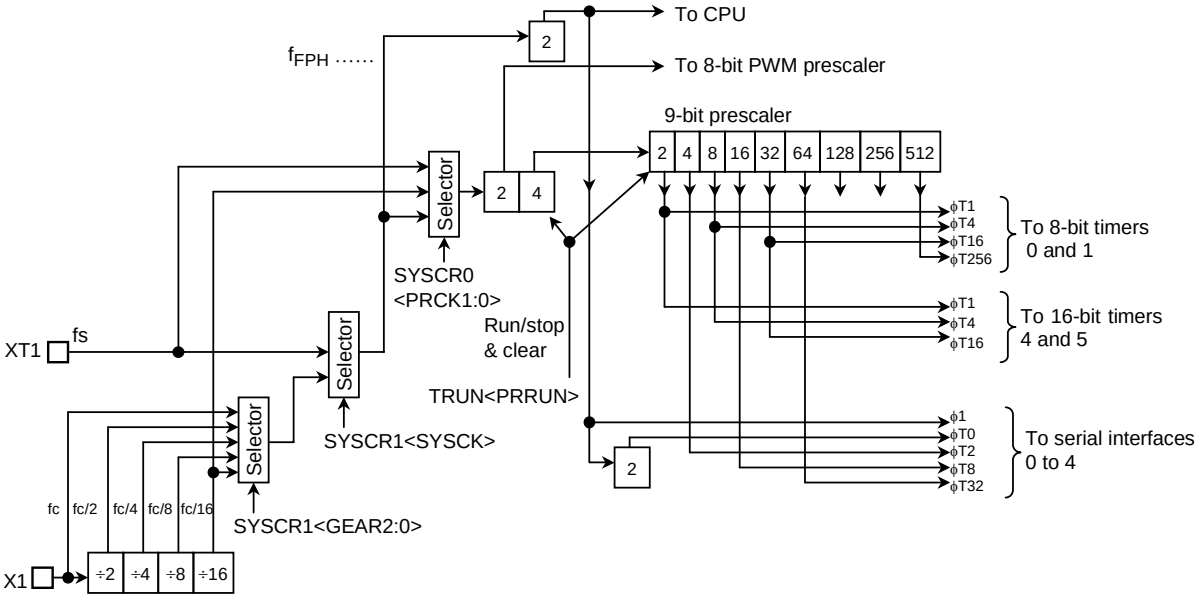


Figure 3.9.11 The Block Diagram of Prescaler

Table 3.9.1 Prescaler Clock Resolution to 8-/16-Bit Timer

at fc = 20 MHz, fs = 32.768 kHz

System Clock Selection <SYSCK>	Prescaler Clock Selection <PRCK1:0>	Clock Gear Value <GEAR2:0>	Prescaler Clock Resolution			
			φT1	φT4	φT16	φT256
1 (fs)	00 (f_{FPH})	XXX	$fs/2^3$ (244 μs)	$fs/2^5$ (977 μs)	$fs/2^7$ (3.9 ms)	$fs/2^{11}$ (62.5 ms)
0 (fc)		000 (fc)	$fc/2^3$ (0.4 μs)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)	$fc/2^{11}$ (102.4 μs)
		001 (fc/2)	$fc/2^4$ (0.8 μs)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{12}$ (204.8 μs)
		010 (fc/4)	$fc/2^5$ (1.6 μs)	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{13}$ (409.6 μs)
		011 (fc/8)	$fc/2^6$ (3.2 μs)	$fc/2^8$ (12.8 μs)	$fc/2^{10}$ (51.2 μs)	$fc/2^{14}$ (819.2 μs)
		100 (fc/16)	$fc/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{11}$ (102.4 μs)	$fc/2^{15}$ (1.64 ms)
XXX	01 (Low-frequency clock)	XXX	$fs/2^3$ (244 μs)	$fs/2^5$ (977 μs)	$fs/2^7$ (3.9 ms)	$fs/2^{11}$ (62.5 ms)
XXX	10 (fc/16 clock)	XXX	$fs/2^7$ (6.4 μs)	$fc/2^9$ (25.6 μs)	$fc/2^{11}$ (102.4 μs)	$fc/2^{15}$ (1.64 ms)

XXX: Don't care

16-bit timer

8-bit timer

Note: The fc/16 clock as a prescaler clock can not be usedwhen the fs is used as a system clock.

The clock selected among f_{FPH} clock, fc/16 clock and fs is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0<PRCK1:0>.

Resetting sets <PRCK1:0> to "00" selects the f_{FPH} clock input divided by 2.

The 16-bit timer selects between 3 clock inputs: ϕT1, ϕT4, and ϕT16 among the prescaler outputs.

This prescaler can be run or stopped by the timer operation control register TRUN<PRRUN>. Counting starts when <PRRUN> is set to "1". The prescaler is cleared zero and stops operation when <PRRUN> is set to "0". Resetting clear <PRRUN> to "0" and stops the prescaler.

When the IDLE1 mode (Only the oscillator operates) is used, set TRUN<PRRUN> to "0" to stop this prescaler before the "HALT" instruction is executed.

2. Up counter

The up counter is a 16-bit binary counter which counts up according to the input clock specified by T4MOD<T4CLK1:0> register.

As the input clock, one of the internal clocks ϕT1, ϕT4, and ϕT16 from 9-bit prescaler (also used for 8-bit timer), and external clock from TI4 pin (also used as P80/INT4 pin) can be selected. When reset, it will be initialized to <T4CLK1:0> = 00 to select TI4 input mode.

Counting or stop and clear of the counter is controlled by timer operation control register TRUN<T4RUN>.

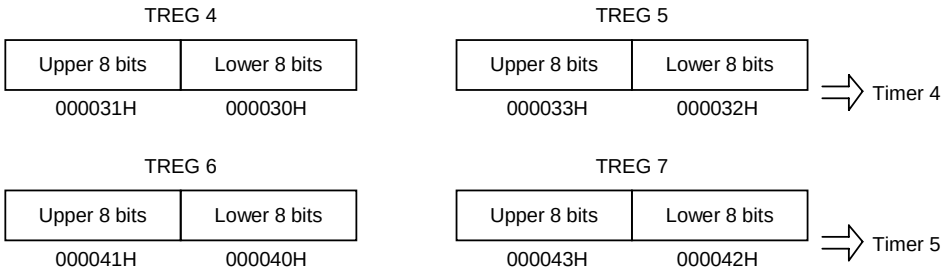
When clearing is enabled, up counter UC will be cleared to zero each timer it coincides matches the timer register TREG5. The "clear enable/disable" is set by T4MOD<CLE>.

If clearing is disabled, the counter operates as a free-running counter.

3. Timer register

These two 16-bit registers are used to set the counter value. When the value of up counter UC4 matches the set value of this timer register, the comparator match detect signal will be active.

Setting data for timer both registers (TREG4 and TREG5) is always needed. For example, either using 2-byte data transfer instruction or using 1-byte data transfer instruction twice for lower 8 bits and upper 8-bits in order.



TREG4 timer register is a double buffer structure, which is paired with register buffer. The timer control register T45CR<DB4EN> controls whether the double buffer structure should be enabled or disable d.: disabled when <DB4EN> = 0, while enabled when <DB4EN> = 1.

When the double buffer is enabled, the timing to transfer data from the register buffer to the timer register is at the match between the up counter (UC4) and timer register TREG5.

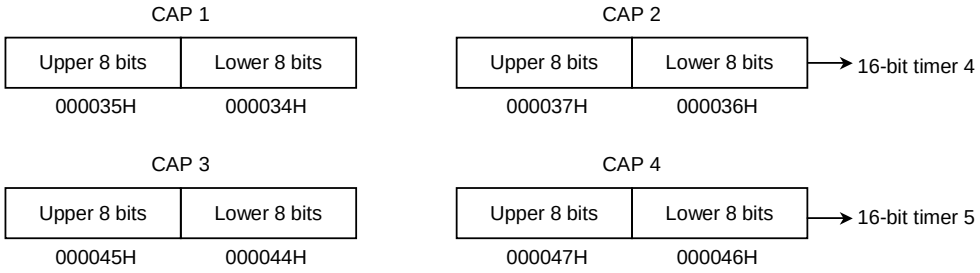
When reset, it will be initialized to <DB4EN> = 0, whereby the double buffer is disabled. To use the double buffer, write data in the timer register, set <DB4EN> = 1, and then write the following data in the register buffer.

TREG4 and register buffer are allocated to the same memory addresses 000030H/000031H. When <DB4EN> = 0, the same value will be written in both TREG4 and register buffer. When <DB4EN> = 1, the value is written into only the register buffer.

4. Capture register

These 16-bit registers are used to hold the values of the up counter.

Data in the capture registers should be read all 16 bits. For example, using a 2-byte data load instruction or two 1-byte data load instruction, from the lower 8 bits followed by the upper 8 bits.



5. Capture input control

This circuit controls the timing to latch the value of up counter UC4 into the capture register (CAP1, CAP2).

The latch timing of capture register is controlled by register T4MOD<CAP12M1:0>.

- When T4MOD<CAP12M1:0> = 00

  Capture function is disabled. Disable is the default on reset.

- When T4MOD<CAP12M1:0> = 01

  Data is loaded to CAP1 at the rising edge of TI4 pin (also used as P80/INT4) input, while data is loaded to CAP2 at the rising edge of TI5 pin (also used as P81/INT5) input. (Time difference measurement)

- When T4MOD<CAP12M1:0> = 10

  Data is loaded to CAP1 at the rising edge of TI4 pin input, while to CAP2 at the falling edge. Only in this setting, interrupt INT4 occurs at falling edge. (Pulse width measurement)

- When T4MOD<CAP12M1:0> = 11

  Data is loaded to CAP1 at the rising edge of timer flip-flop TFF1, while to CAP2 at the falling edge.

  Besides, the value of up counter can be loaded to capture registers by software. Whenever "0" is written in T4MOD<CAPIN> the current value of up counter will be loaded to capture register CAP1. It is necessary to keep the prescaler in RUN mode (TRUN<PRRUN> to be "1").

6. Comparator

These are 16-bit comparators which compare the up counter UC4 value with the set value of (TREG4, TREG5) to detect the match. When a match is detected, the comparators generate an interrupt (INTTR4, INTTR5) respectively.

The up counter UC4 is cleared only when UC4 matches TREG5. (The clearing of up counter UC4 can be disabled by setting T4MOD<CLE> = 0.)

7. Timer flip-flop (TFF4)

   This flip-flop is inverted by the match detect signal from the comparators and the latch signals to the capture registers. Disable/enable of inversion can be set for each element by T4FFCR<CAP2T4, CAP1T4, EQ5T4, EQ4T4>.

   TFF4 will be inverted when "00" is written in T4FFCR<TFF4C1:0>. Also it is set to "1" when "01" is written, and cleared to "0" when "10" is written. The value of TFF4 can be output to the timer output pin TO4 (also used as P82). TFF4 is undefined on reset.

8. Timer flip-flop (TFF5)

   This flip-flop is inverted by the match detect signal between the up counter (UC4) and the timer register TREG5 and the latch signal to the capture register CAP2. Disable/enable of inversion can be set for each element by T4MOD<CAP2T5, EQ5T5>. TFF5 will be inverted when "00" is written in T4FFCR<TFF5C1:0>. Also it is set to "1" when "01" is written, and cleared to "0" when "10" is written. The value of TFF5 can be output to the timer output pin TO5 (also used as P83). TFF5 is undefined on reset.

   Note: This flip-flop (TFF5) is contained only in the 16-bit timer 4.

(1) 16-bit timer mode

Generating interrupts at fixed intervals.

In this example, the interval time is set in the timer register TREG5 to generate the interrupt INTTR5.

		7	6	5	4	3	2	1	0	
TRUN	←	–	X	–	0	–	–	–	–	Stop timer 4.
INTET54	←	1	1	0	0	1	0	0	0	Enable INTTR5 and sets interrupt level 4. Disable INTTR4.
T4FFCR	←	1	1	0	0	0	0	1	1	Disable trigger.
T4MOD	←	0	0	1	0	0	1	*	*	Select internal clock for input and disable the capture function.
					(** = 01, 10, 11)					
TREG5	←	*	*	*	*	*	*	*	*	Set the interval time (16 bits).
		*	*	*	*	*	*	*	*	
TRUN	←	1	X	–	1	–	–	–	–	Start timer 4.

X: Don't care, –: No change

(2) 16-bit event counter mode

In 16-bit timer mode as described in above, the timer can be used as an event counter by selecting the external clock (TI4 pin input) as the input clock. To read the value of the counter, first perform "software capture" once and read the captured value.

The counter counts at the rising edge of TI4 pin input.

TI4 pin can also be used as P80/INT4.

		7	6	5	4	3	2	1	0	
TRUN	←	–	X	–	0	–	–	–	–	Stop timer 4.
P8CR	←	–	–	–	–	–	–	–	0	Set P80 to input mode.
INTET54	←	1	1	0	0	1	0	0	0	Enable INTTR5 and sets interrupt level 4, while disables INTTR4.
T4FFCR	←	1	1	0	0	0	0	1	1	Disable trigger.
T4MOD	←	0	0	1	0	0	1	0	0	Select TI4 as the input clock.
TREG5	←	*	*	*	*	*	*	*	*	Set the number of counts (16 bits).
		*	*	*	*	*	*	*	*	
TRUN	←	1	X	–	1	–	–	–	–	Start timer 4.

Note: When used as an event counter, set the prescaler in RUN mode.

(3) 16-bit programmable pulse generation (PPG) output mode

The PPG mode is obtained by inversion of the timer flip-flop TFF4 that is to be enabled by the match of the up counter UC4 with the timer register TREG4 or 5 and to be output to TO4 (also used as P82). In this mode, the following conditions must be satisfied.

(Set value of TREG4) < (Set value of TREG5)

		7	6	5	4	3	2	1	0	
T45CR	←	0	X	X	X	–	–	–	0	Double buffer of TREG4 disable.
TRUN	←	–	X	–	0	–	–	–	–	Stop timer 4.
TREG4	←	*	*	*	*	*	*	*	*	Set the duty. (16 bits)
		*	*	*	*	*	*	*	*	
TREG5	←	*	*	*	*	*	*	*	*	Set the cycle. (16 bits)
		*	*	*	*	*	*	*	*	
T45CR	←	0	X	X	X	–	–	–	1	Double Buffer of TREG4 enable. (Change the duty and cycle at the interrupt INTTR5)
T4FFCR	←	1	1	0	0	1	1	1	0	Set the mode to invert TFF4 at the match with TREG4/TREG5, and also set the TFF4 to "0".
T4MOD	←	0	0	1	0	0	1	*	*	Select the internal clock for the input, and disable the capture function.
					(** = 01, 10, 11)					
P8CR	←	–	–	–	–	–	1	–	–	Assign P82 as TO4.
P8FC	←	X	–	X	X	–	1	X	X	
TRUN	←	1	X	–	1	–	–	–	–	Start timer 4.

X: Don't care, –: No change

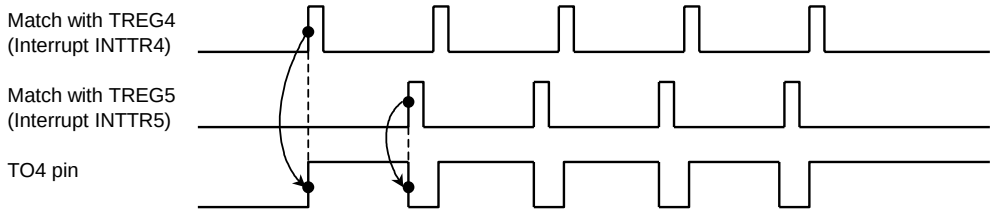


Figure 3.9.12 Programmable Pulse Generation (PPG) Output Waveforms

When the double buffer of TREG4 is enabled in this mode, the value of register buffer 4 will be shifted in TREG4 at match with TREG5. This feature makes easy the handling of low duty waves.

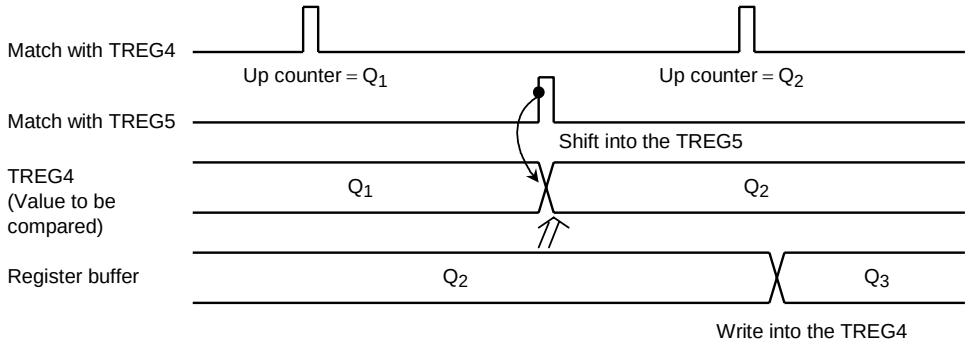


Figure 3.9.13 Operation of Register Buffer

Shows the block diagram of this mode.

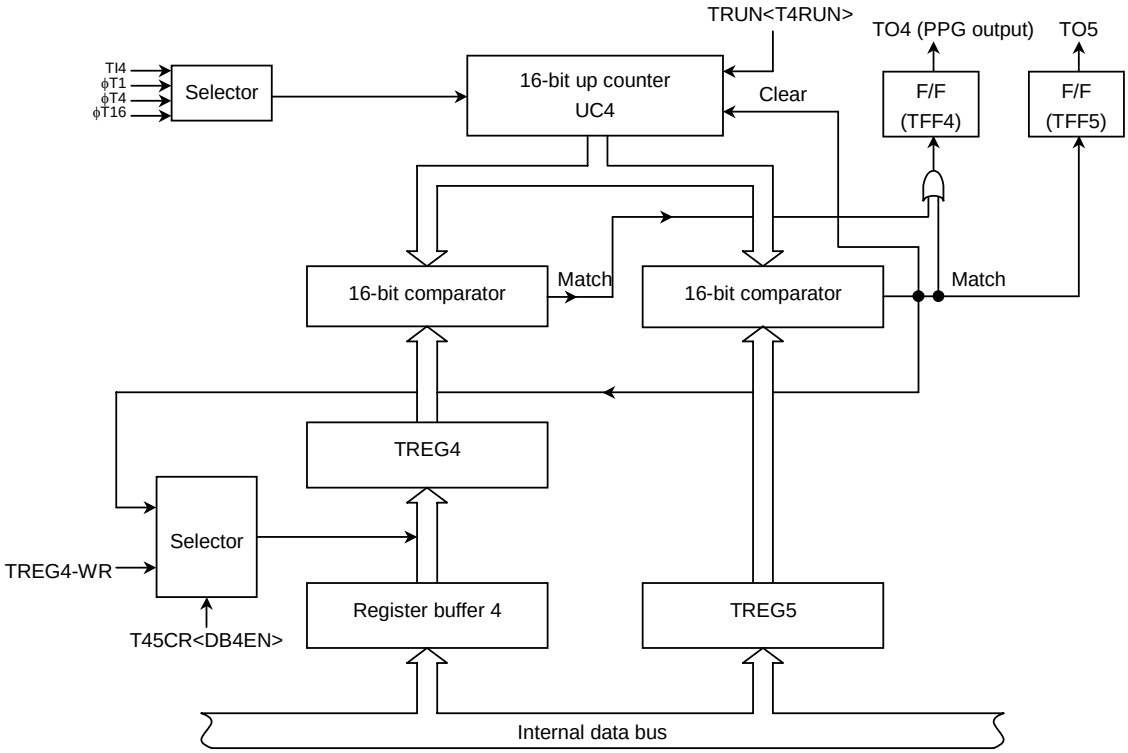


Figure 3.9.14 Block Diagram of 16-Bit PPG Mode

(4) Application examples of capture function

The loading of up counter (UC4) values into the capture registers CAP1 and CAP2, the timer flip-flop TFF4 inversion due to the match detection by comparators CP4 and CP5, and the output of the TFF4 status to TO4 pin can be enabled or disabled. Combined with interrupt function, they can be applied in many ways, for example:

1. One-shot pulse output from external trigger pulse
2. Frequency measurement
3. Pulse width measurement
4. Time difference measurement

1. One-shot pulse output from external trigger pulse

Set the up counter UC4 in free-running mode with the internal input clock, input the external trigger pulse from TI4 pin, and load the value of up counter into capture register CAP1 at the rising edge of the TI4 pin. Then set to T4MOD<CAP12M1:0> = 01.

When the interrupt INT4 is generated at the rising edge of TI4 input, set the CAP1 value (c) plus a delay time (d) to TREG4 (= c + d), and set the above set value (c + d) plus a one-shot pulse width (p) to TREG5 (= c + d + p). When the interrupt INT4 occurs the T4FFCR<EQ5T4, EQ4T4> register should be set that the TFF4 inversion is enabled only when the up counter value matches TREG4 or TREG5. When interrupt INTTR5 occurs, this inversion will be disabled.

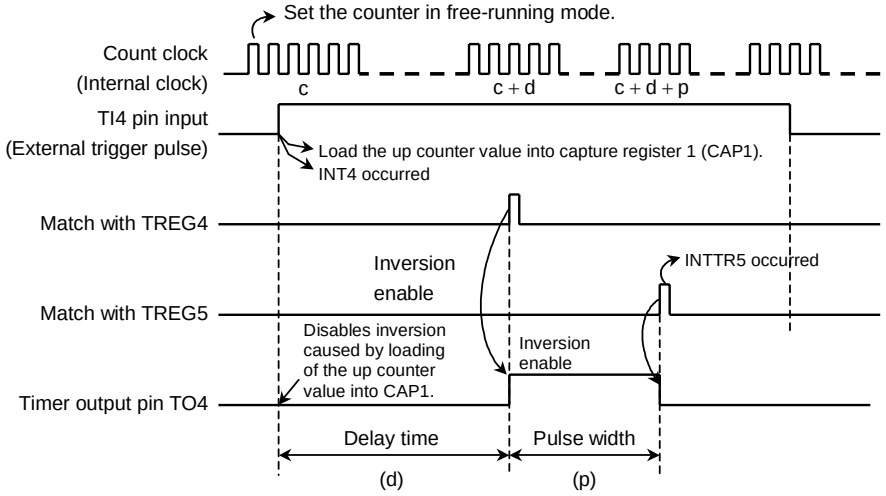
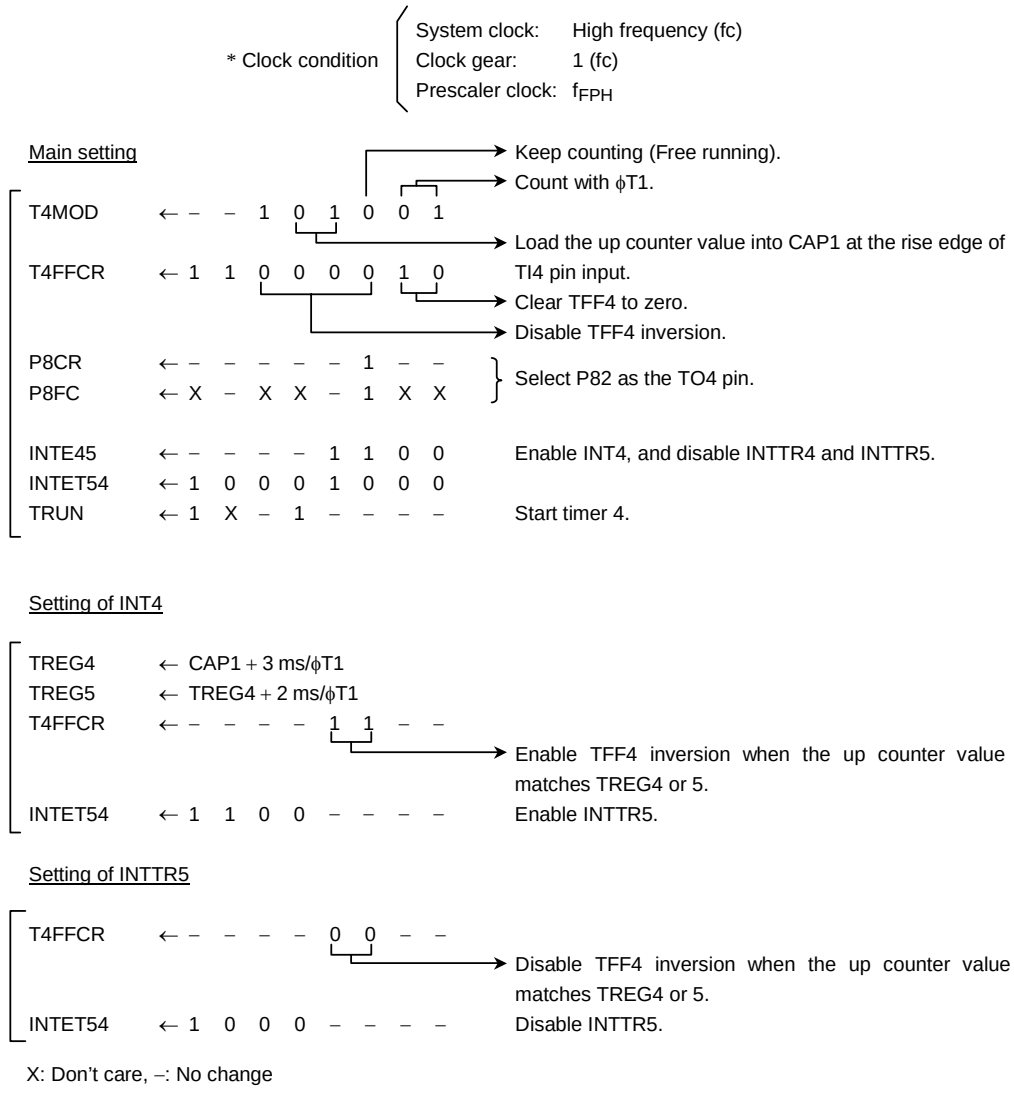


Figure 3.9.15 One-shot Pulse Output (with delay)

Setting example: To output 2 ms one-shot pulse with 3 ms delay to the external trigger pulse to TI4 pin



When delay time is unnecessary, invert timer flip-flop TFF4 when the up counter value is loaded into capture register 1 (CAP1), and set the CAP1 value (c) plus and the one-shot pulse width (p) to TREG5 when the interrupt INT4 occurs. The TFF4 inversion should be enabled when the up counter (UC4) value matches TREG5, and disabled when generating the interrupt INTTR5.

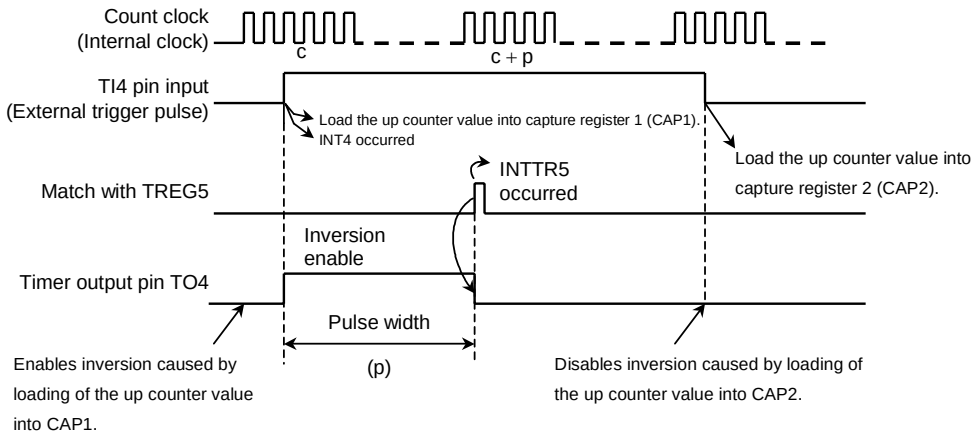


Figure 3.9.16 One-shot Pulse Output (without Delay)

2. Frequency measurement

The frequency of the external clock can be measured in this mode. The clock is input through the TI4 pin, and its frequency is measured by the 8-bit timers (Timer 0 and timer 1) and the 16-bit timer/event counter (Timer 4).

The TI4 pin input should be selected for the input clock of timer 4. The value of the up counter is loaded into the capture register CAP1 at the rising edge of the timer flip-flop TFF1 of 8-bit timers (Timer 0 and timer 1), and into CAP2 at its falling edge.

The frequency is calculated by the difference between the loaded values in CAP1 and CAP2 when the interrupt (INTT0 or INTT1) is generated by either 8-bit timer.

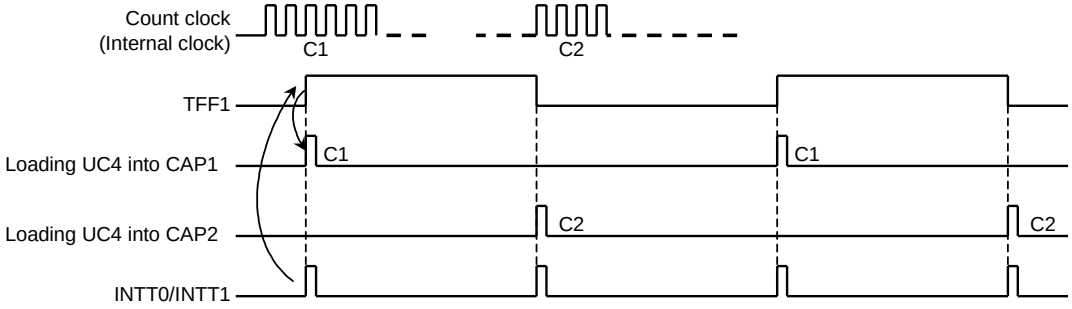


Figure 3.9.17 Frequency Measurement

For example, if the value for the level "1" width of TFF1 of the 8-bit timer is set to 0.5 [s], and the difference between CAP1 and CAP2 is 100, the frequency will be 100 ÷ 0.5 [s] = 200 [Hz].

3. Pulse width measurement

This mode allows to measure the "H" level width of an external pulse. While keeping the 16-bit timer/event counter counting (Free running) with the internal clock input, the external pulse is input through the TI4 pin. Then the capture function is used to load the UC4 values into CAP1 and CAP2 at the rising edge and falling edge of the external trigger pulse respectively. The interrupt INT4 occurs at the falling edge of TI4.

The pulse width is obtained from the difference between the values of CAP1 and CAP2 and the internal clock cycle.

For example, if the internal clock is 0.8 microseconds and the difference between CAP1 and CAP2 is 100, the pulse width will be $100 \times 0.8\ \mu s = 80\ \mu s$.

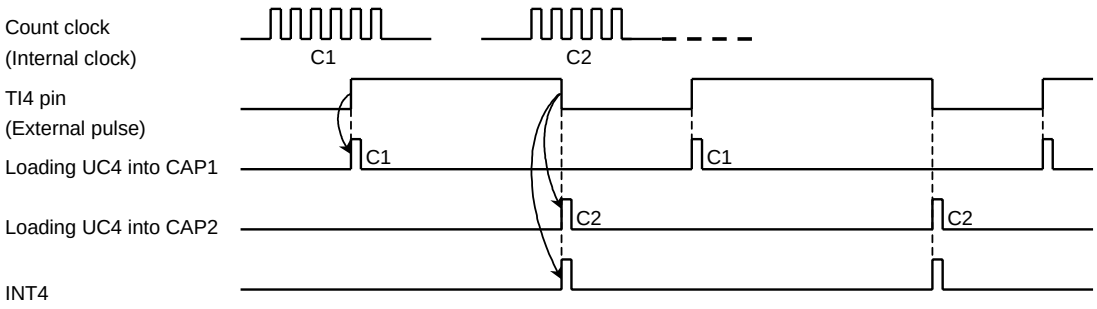


Figure 3.9.18 Pulse Width Measurement

Note: Only in this pulse width measuring mode (T4MOD<CAP12M1:0> = 10), external interrupt INT4 occurs at the falling edge of TI4 pin input. In other modes, it occurs at the rising edge.

The width of "L" level can be measured from the difference between the first C2 and the second C1 at the second INT4 interrupt.

4. Time difference measurement

This mode is used to measure the difference in time between the rising edges of external pulses input through TI4 and TI5.

Keep the 16-bit timer/event counter (Timer 4) counting (Free running) with the internal clock, and load the UC4 value into CAP1 at the rising edge of the input pulse to TI4. Then the interrupt INT4 is generated.

Similarly, the UC4 value is loaded into CAP2 at the rising edge of the input pulse to TI5, generating the interrupt INT5.

The time difference between these pulses can be obtained from the difference between the time counts at which loading the up counter value into CAP1 and CAP2 has been done.

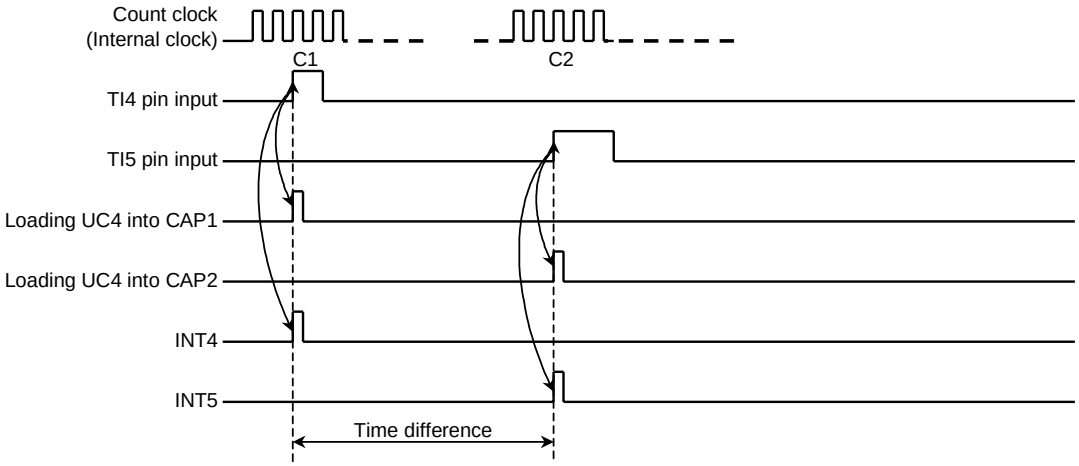


Figure 3.9.19 Time Difference Measurement

(5) Different phased pulses output mode (This mode can be used only timer 4.)

In this mode, signals with any different phase can be outputted by free-running up counter UC4.

When the value in up counter UC4 and the value in TREG4 (TREG5) match, the value in TFF4 (TFF5) is inverted and output to TO4 (TO5).

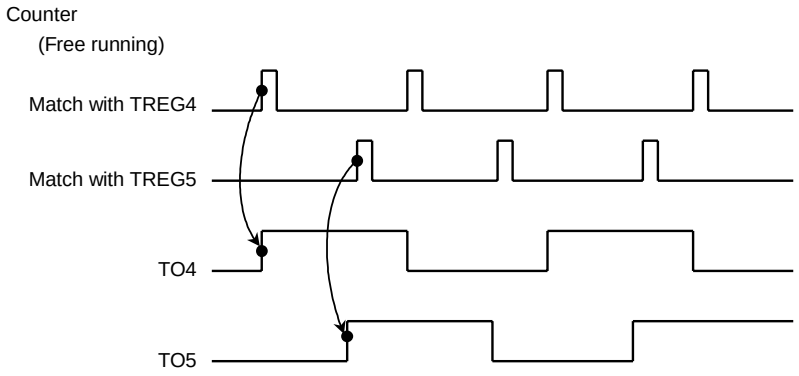


Figure 3.9.20 Phase Output

Cycles (Counter overflow time) of the above output waves are listed on Table 3.9.2.

Table 3.9.2 Timer Output Cycle on the Different Phased Pulse Output Mode

at fc = 20 MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	Counter Overflow Time		
			ϕT1	ϕT4	ϕT16
1 (fs)	00 (f_{FPH})	XXX	15.999 s	64.000 s	256.000 s
0 (fc)		000 (fc)	26.214 ms	104.858 ms	419.430 ms
		001 (fc/2)	52.429 ms	209.715 ms	838.861 ms
		010 (fc/4)	104.858 ms	419.430 ms	1.678 s
		011 (fc/8)	209.715 ms	838.861 ms	3.355 s
		100 (fc/16)	419.430 ms	1.678 s	6.711 s
XXX	01 (Low-frequency clock)	XXX	15.999 s	64.000 s	256.000 s
XXX	10 (fc/16 clock)	XXX	419.430 ms	1.678 s	6.711 s

XXX: Don't care

3.10 Serial Channel

TMP93CW46A contains 5 serial I/O channels. Channels 0 to 3 select UART mode (Asynchronous transmission) or I/O interface mode (Synchronous transmission). Channel 4 is used only in UART mode.

The serial channel has the following operation modes.

- I/O interface mode (Channel 0 to 3) — Mode 0: To transmit and receive I/O data using the synchronizing signal SCLK for extending I/O.
- UART mode (Channel 0 to 4) — 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 a wake-up function for making the master controller start slave controllers in a serial link system.

Figure 3.10.1 shows the data format in each mode.

Serial channels 0 to 4 can be used independently.

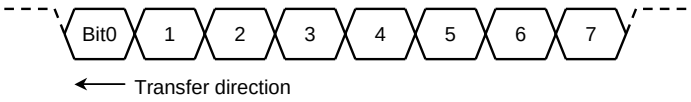
All channels have the same operations except the following points, thus only the operation of channel 0 will be explained below.

Different points among Channels 0 to 4

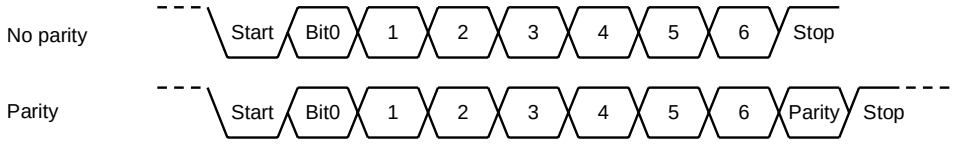
	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Pin name	TXD0 (P90) RXD0 (P91) $\overline{CTS0}$ /SCLK0 (P92)	TXD1 (P93) RXD1 (P94) SCLK1 (P95)	TXD2 (P60) RXD2 (P61) $\overline{CTS2}$ /SCLK2 (P62)	TXD3 (P63) RXD3 (P64) $\overline{CTS3}$ /SCLK3 (P65)	TXD4 (P66) RXD4 (P67)
UART mode	Yes	Yes	Yes	Yes	Yes
I/O interface mode	Yes	Yes	Yes	Yes	No
Handshake function	Yes	No (No $\overline{CTS1}$ pin)	Yes	Yes	No (No $\overline{CTS4}$ pin)

Note: Using the handshake function can transmit in units of one data format. Thus over run error is prevented.
See “Handshake function” for details.

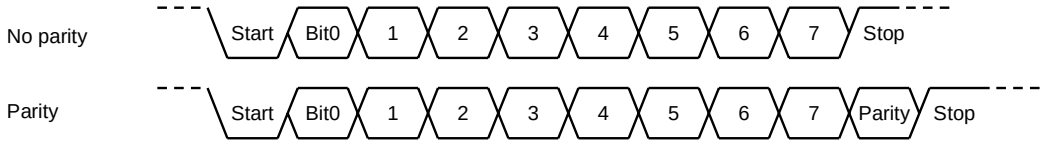
- Mode 0 (I/O interface mode)



- Mode 1 (7-bit UART mode)



- Mode 2 (8-bit UART mode)



- Mode 3 (9-bit UART mode)

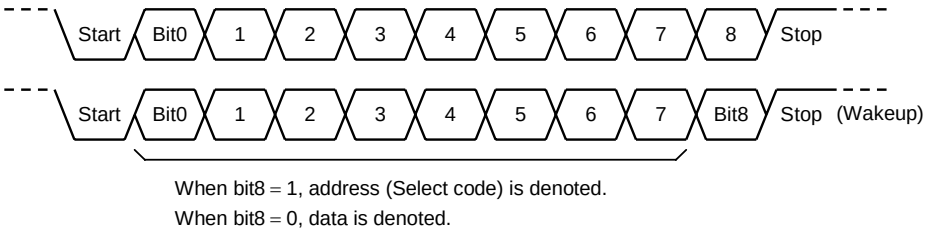


Figure 3.10.1 Data Formats

The serial channel has buffer registers for transmitting and receiving operations in order to temporarily store transmitted or received data. This is done so that transmitting and receiving operations can be done independently (Full duplex).

However, in I/O interface mode, the SCLK (Serial clock) pin is used for both transmitting and receiving, the channel becomes half duplex.

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

By using $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ (There is no $\overline{\text{RTS}}$ pin, so any single 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)

In the UART mode, a check function is added to not start the receiving operation by erroneous 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 of the start bit.

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 (Transmit interrupt) or INTRX (Receive interrupt) interrupt occurs. If an overrun error, parity error, or framing error occurs during receiving operation, flag SC0CR<OERR, PERR, FERR> will be set.

The serial channels 0 to 4 include a special baud rate generator, which can set to any baud rate by dividing the frequency of 4 clocks (ϕT0, ϕT2, ϕT8, and ϕT32 from the 9-bit prescaler shared by the 8-bit/16-bit timers by the value 1, 2 + n/16 to 15 + n/16, 16n = 0 to 15).

In I/O interface mode, it is possible to input synchronous signals as well as to transmit or receive data by using an external clock.

3.10.1 Control Registers

The serial channels are controlled by 4 control registers SC0CR, SC0MOD, BR0CR and BRADD0. Transmitted and received data are stored in register SC0BUF.

Note: The number of the control register name is equaled to the channel number.

SC0MOD (0052H)	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfer data bit8	Hand shake 0: CTS disable 1: CTS enable	Receiving function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO0 trigger 01: Baud rate generator 10: Internal clock ϕ1 11: External clock (SLCK0 input)	

Serial transmission clock source (UART)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	External clock (SLCK0 input)

Note: The clock selection for the I/O interface mode is controlled by the serial control register (SC0CR).

Serial transmission mode

00	I/O interface mode	
01	UART	7-bit length
10		8-bit length
11		9-bit length

Wakeup function

	9-bit UART	Other mode
0	Interrupt when data is received	Don't care
1	Interrupt only when RB8 = 1	

Receiving function

0	Receive disable
1	Receive enable

Hand shake function ($\overline{CTS}$ pin)

0	Disable (Always transferable)
1	Enable

Transmission data bit8

Figure 3.10.2 Serial Mode Control Register (Channel 0, SC0MOD)

SC0CR (0051H)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error			0: SCLK0 (↑) 1: SCLK0 (↓)	0: Baud rate generator 1: SCLK0 pin input
				Overrun	Parity	Framing		

Select I/O interface Input clock

0	Baud rate generator
1	SCLK0 pin input

Edge selection for SCLK pin (Input mode only)

0	Transmits and receives (↑) data at rise edge of SCLK0
1	Transmits and receives (↓) data at fall edge of SCLK0

Framing error flag
Parity error flag
Overrun error flag
} Cleared to 0 when read.

Enable parity addition

0	Disable
1	Enable

Addition/check of even parity

0	Odd parity
1	Even parity

Receiving data bit8

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

Figure 3.10.3 Serial Control Register (Channel 0, SC0CR)

BR0CR (0053H)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR0ADD	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	(Note) Always fixed to "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Setting of the divided frequency			

+ (16 − K)/16 division enable

0	Disable
1	Enable

Selecting the input clock of baud rate generator

00	Internal clock φT0
01	Internal clock φT2
10	Internal clock φT8
11	Internal clock φT32

BRADD0 (006BH)

	7	6	5	4	3	2	1	0
Bit symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets frequency divisor "K" (divided by N + (16 − K)/16)			

Sets baud rate generator frequency divisor

BR0CR <BR0S3:0> / BRADD0 <BR0K3:0>	BR0CR<BR0ADD> = 1		BR0CR<BR0ADD> = 0
0000	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	
0001 (K = 1) to 1111 (K = 15)	Disable	Divided by N + (16 − K)/16	Divided by N

Note 1: Set TRUN<PRRUN> to "1" when the baud rate generator is used.

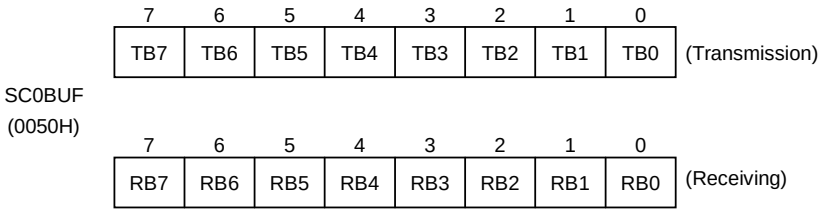
Note 2: Set BR0CE<BR0ADD> to "1" after setting K (K = 1 to 15) to BRADD0<BR0K3:0> when + (16 − K)/16 division function is used.

Note 3: + (16 − K)/16 division function is possible to use in only UART mode.
Set BR0CR<BR0ADD> to "0" and disable + (16 − K)/16 division function in I/O interface mode.

Note 4: BRADD0<bit7:4> is always read as "1".

Note 5: Don't read from or write to BR0CR register during sending or receiving.

Figure 3.10.4 Baud Rate Generator Control (Channel 0, BR0CR, BRADD0)



Note: Prohibit read-modify-write for SC0BUF.

Figure 3.10.5 Serial Transmission/Receiving Buffer Registers (Channel 0, BR0CR)

SC1MOD (0056H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transferred data bit8	Fix at "0".	Receiving function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO0 trigger 01: Baud rate generator 10: Internal clock ϕ1 11: External clock (SCLK1 input)	

Serial transmission clock source (for UART)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	External clock (SCLK1 input)

Note: The clock selection for the I/O interface mode is controlled by the serial control register (SC1CR).

Serial transmission mode

00	I/O interface mode	
01	UART mode	7-bit length
10		8-bit length
11		9-bit length

Wakeup function

	9-bit UART	Other mode
0	Interrupt when data is received	Don't care
1	Interrupt only when RB8 = 1	

Receiving control

0	Receive disable
1	Receive enable

Transmission data bit8

Figure 3.10.6 Serial Mode Control Register (Channel 1, SC1MOD)

SC1CR (0055H)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (clear to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error			0: SCLK1 (↑) 1: SCLK1 (↓)	0: Baud rate generator 1: SCLK1 pin input
				Overrun	Parity	Framing		

Select I/O interface input clock

0	Baud rate generate
1	SCLK1 pin input

Edge selection for SCLK pin (Input mode only)

0	Transmits and receives (↑) data at rise edge of SCLK1
1	Transmits and receives (↓) data at fall edge of SCLK 1

Framing error flag
Parity error flag
Overrun error flag
} Cleared to 0 when read

Enable parity addition

0	Disable
1	Enable

Addition/check of even parity

0	Odd parity
1	Even parity

Receiving data bit8

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

Figure 3.10.7 Serial Control Register (Channel 1, SC1CR)

BR1CR (0057H)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR1ADD	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Always fixed to "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Setting of the divided frequency			

+ (16 − K)/16 division enable

0	Disable
1	Enable

Selecting the input clock of baud rate generator

00	Internal clock φT0
01	Internal clock φT2
10	Internal clock φT8
11	Internal clock φT32

BRADD1 (006CH)

	7	6	5	4	3	2	1	0
Bit symbol					BR1K3	BR1K2	BR1K1	BR1K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets frequency divisor "K" (divided by N + (16 − K)/16)			

Sets baud rate generator frequency divisor

	BR1CR<BR1ADD> = 1		BR1CR<BR1ADD> = 0
BR1CR <BR1S3:0> / BRADD0 <BR1K3:0>	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	
0001 (K = 1) to 1111 (K = 15)	Disable	Divided by N + (16 − K)/16	Divided by N

Note 1: Set TRUN<PRRUN> to "1" when the baud rate generator is used.

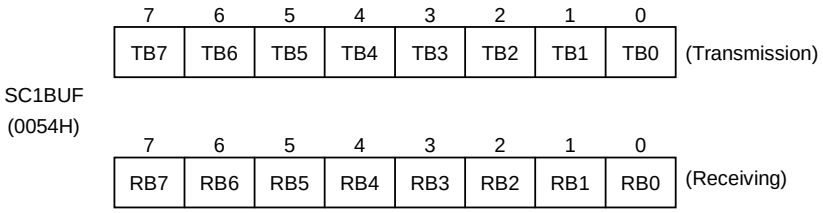
Note 2: Set BR1CR<BR1ADD> to "1" after setting K (K = 1 to 15) to BRADD1<BR1K3:0> when + (16 − K)/16 division function is used.

Note 3: + (16 − K)/16 division function is possible to use in only UART mode.
Set BR1CR<BR1ADD> to "0" and disable + (16 − K)/16 division function in I/O interface mode.

Note 4: BRADD1<bit7:4> is always read as "1".

Note 5: Don't read from or write to BR1CR register during sending or receiving.

Figure 3.10.8 Baud Rate Generator Control (Channel 1, BR1CR, BRADD1)



Note: Prohibit read-modify-write for SC1BUF.

Figure 3.10.9 Serial Transmission/Receiving Buffer Registers (Channel 1, SC1BUF)

SC2MOD (002EH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transferred data bit8	Hand shake 0: CTS disable 1: CTS enable	Receiving function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO0 trigger 01: Baud rate generator 10: Internal clock ϕ1 11: External clock (SCLK2 input)	

Serial transmission clock source (for UART)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	External clock (SCLK2 input)

Note: The clock selection for the I/O interface mode is controlled by the serial control register (SC2CR).

Serial transmission mode

00	I/O interface mode	
01	UART mode	7-bit length
10		8-bit length
11		9-bit length

Wakeup function

	9-bit UART	Other mode
0	Interrupt when data is received	Don't care
1	Interrupt only when RB8 = 1	

Receiving control

0	Receive disable
1	Receive enable

Hand shake function ($\overline{CTS}$ pin)

0	Disable (Always transferable)
1	Enable

Transmission data bit8

Figure 3.10.10 Serial Mode Control Register (Channel 2, SC2MOD)

SC2CR (002DH)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (clear to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error			0: SCLK2 (↑) 1: SCLK2 (↓)	0: Baud rate generator 1: SCLK2 pin input
				Overrun	Parity	Framing		

Select I/O interface input clock

0	Baud rate generate
1	SCLK2 pin input

Edge selection for SCLK pin (Input mode only)

0	Transmits and receives (↑) data at rise edge of SCLK2
1	Transmits and receives (↓) data at fall edge of SCLK2

Framing error flag
Parity error flag
Overrun error flag
} Cleared to 0 when read

Enable parity addition

0	Disable
1	Enable

Addition/check of even parity

0	Odd parity
1	Even parity

Receiving data bit8

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

Figure 3.10.11 Serial Control Register (Channel 2, SC2CR)

BR2CR (002FH)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR2ADD	BR2CK1	BR2CK0	BR2S3	BR2S2	BR2S1	BR2S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Always fixed to "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Setting of the divided frequency			

+ (16 − K)/16 division enable

0	Disable
1	Enable

Selecting the input clock of baud rate generator

00	Internal clock φT0
01	Internal clock φT2
10	Internal clock φT8
11	Internal clock φT32

BRADD2 (002BH)

	7	6	5	4	3	2	1	0
Bit symbol					BR2K3	BR2K2	BR2K1	BR2K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets frequency divisor "K" (divided by N + (16 − K)/16)			

Sets baud rate generator frequency divisor

BRADD2 <BR2K3:0> \ BR2CR <BR2S3:0>	BR2CR<BR2ADD> = 1		BR2CR<BR2ADD> = 0
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	
0001 (K = 1) to 1111 (K = 15)	Disable	Divided by N + (16 − K)/16	Divided by N

Note 1: Set TRUN<PRRUN> to "1" when the baud rate generator is used.

Note 2: Set BR2CR<BR2ADD> to "1" after setting K (K = 1 to 15) to BRADD2<BR2K3:0> when + (16 − K)/16 division function is used.

Note 3: + (16 − K)/16 division function is possible to use in only UART mode.
Set BR2CR<BR2ADD> to "0" and disable + (16 − K)/16 division function in I/O interface mode.

Note 4: BRADD2<bit7:4> is always read as "1".

Note 5: Don't read from or write to BR2CR register during sending or receiving.

Figure 3.10.12 Baud Rate Generator Control (Channel 2, BR2CR, BRADD2)

SC2BUF (002CH)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receiving)

Note: Prohibit read-modify-write for SC2BUF.

Figure 3.10.13 Serial Transmission/Receiving Buffer Registers (Channel 2, SC2BUF)

SC3MOD (003EH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transferred data bit8	Hand shake 0: CTS disable 1: CTS enable	Receiving function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO0 trigger 01: Baud rate generator 10: Internal clock ϕ1 11: External clock (SCLK3 input)	

Serial transmission clock source (for UART)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	External clock (SCLK3 input)

Note: The clock selection for the I/O interface mode is controlled by the serial control register (SC3CR).

Serial transmission mode

00	I/O interface mode	
01	UART mode	7-bit length
10		8-bit length
11		9-bit length

Wakeup function

	9-bit UART	Other mode
0	Interrupt when data is received	Don't care
1	Interrupt only when RB8 = 1	

Receiving control

0	Receive disable
1	Receive enable

Hand shake function ($\overline{CTS}$ pin)

0	Disable (always transferable)
1	Enable

Transmission data bit8

Figure 3.10.14 Serial Mode Control Register (Channel 3, SC3MOD)

SC3CR (003DH)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (clear to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error			0: SCLK3 (↑) 1: SCLK3 (↓)	0: Baud rate generator 1: SCLK3 pin input
				Overrun	Parity	Framing		

Select I/O interface input clock

0	Baud rate generate
1	SCLK3 pin input

Edge selection for SCLK pin (Input mode only)

0	Transmits and receives (↑) data at rise edge of SCLK3
1	Transmits and receives (↓) data at fall edge of SCLK3

Framing error flag
Parity error flag
Overrun error flag
} Cleared to 0 when read

Enable parity addition

0	Disable
1	Enable

Addition/check of even parity

0	Odd parity
1	Even parity

Receiving data bit8

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

Figure 3.10.15 Serial Control Register (Channel 3, SC3CR)

BR3CR (003FH)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR3ADD	BR3CK1	BR3CK0	BR3S3	BR3S2	BR3S1	BR3S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Always fixed to "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Setting of the divided frequency			

+ (16 − K)/16 division enable

0	Disable
1	Enable

Selecting the input clock of baud rate generator

00	Internal clock φT0
01	Internal clock φT2
10	Internal clock φT8
11	Internal clock φT32

BRADD3 (003BH)

	7	6	5	4	3	2	1	0
Bit symbol					BR3K3	BR3K2	BR3K1	BR3K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets frequency divisor "K" (divided by N + (16 − K)/16)			

Sets baud rate generator frequency divisor

BR3CR <BR3S3:0> / BRADD3 <BR3K3:0>	BR3CR<BR3ADD> = 1		BR3CR<BR3ADD> = 0
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	
0001 (K = 1) to 1111 (K = 15)	Disable	Divided by N + (16 − K)/16	Divided by N

Note 1: Set TRUN<PRRUN> to "1" when the baud rate generator is used.

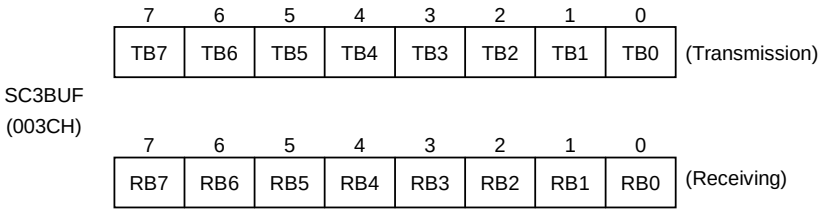
Note 2: Set BR3CR<BR3ADD> to "1" after setting K (K = 1 to 15) to BRADD3<BR2K3:0> when + (16 − K)/16 division function is used.

Note 3: + (16 − K)/16 division function is possible to use in only UART mode.
Set BR3CR<BR3ADD> to "0" and disable + (16 − K)/16 division function in I/O interface mode.

Note 4: BRADD3<bit7:4> is always read as "1".

Note 5: Don't read from or write to BR3CR register during sending or receiving.

Figure 3.10.16 Baud Rate Generator Control (Channel 3, BR3CR, BRADD3)



Note: Prohibit read-modify-write for SC3BUF.

Figure 3.10.17 Serial Transmission/Receiving Buffer Registers (Channel 3, SC3BUF)

SC4MOD (004EH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transferred data bit8	Fix at "0".	Receiving function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: Reserved 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO0 trigger 01: Baud rate generator 10: Internal clock ϕ1 11: Don't care	

Serial transmission clock source (for UART)

00	Timer 0 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	Don't care

Note: The clock selection for the I/O interface mode is controlled by the serial control register (SC4CR).

Serial transmission mode

00	Reserved	
01	UART mode	7-bit length
10		8-bit length
11		9-bit length

Wakeup function

	9-bit UART	Other mode
0	Interrupt when data is received	Don't care
1	Interrupt only when RB8 = 1	

Receiving control

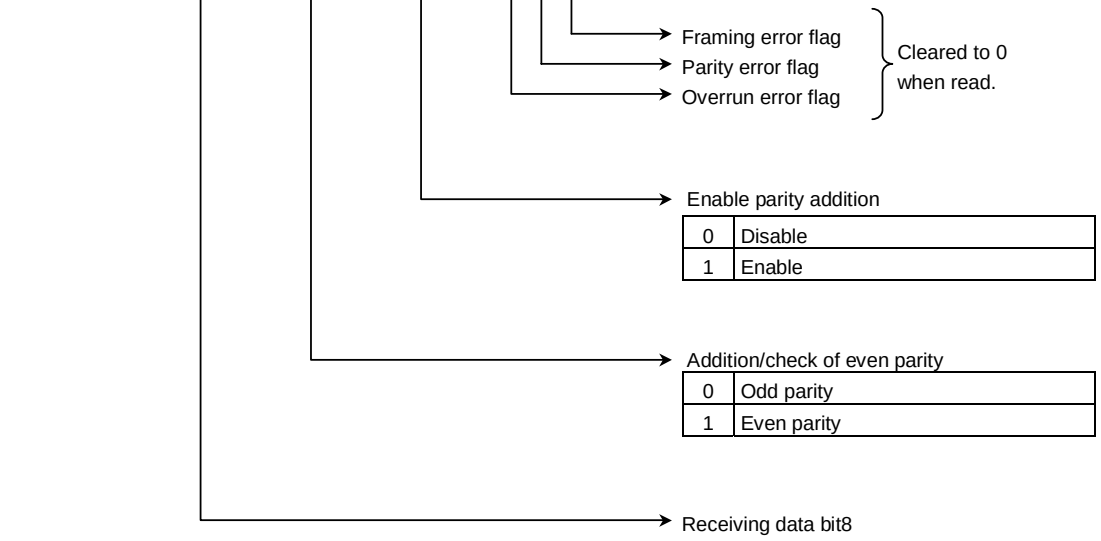
0	Receive disable
1	Receive enable

Transmission data bit8

Figure 3.10.18 Serial Mode Control Register (Channel 4, SC4MOD)

SC4CR (004DH)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	–	–
Read/Write	R	R/W		R (cleared to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error Overrun	1: Error Parity	1: Error Framing	Always fixed to "0".	



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

Figure 3.10.19 Serial Control Register (Channel 4, SC4CR)

BR4CR (004FH)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR4ADD	BR4CK1	BR4CK0	BR4S3	BR4S2	BR4S1	BR4S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Always fixed to "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Setting of the divided frequency			

+ (16 − K)/16 division enable

0	Disable
1	Enable

Selecting the input clock of baud rate generator

00	Internal clock φT0
01	Internal clock φT2
10	Internal clock φT8
11	Internal clock φT32

BRADD4 (004BH)

	7	6	5	4	3	2	1	0
Bit symbol					BR4K3	BR4K2	BR4K1	BR4K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets frequency divisor "K" (divided by N + (16 − K)/16)			

Sets baud rate generator frequency divisor

BRADD4 <BR4K3:0> \ BR4CR <BR4S3:0>	BR4CR<BR4ADD> = 1		BR4CR<BR4ADD> = 0
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	
0001 (K = 1) to 1111 (K = 15)	Disable	Divided by N + (16 − K)/16	Divided by N

Note 1: Set TRUN<PRRUN> to "1" when the baud rate generator is used.

Note 2: Set BR4CR<BR4ADD> to "1" after setting K (K = 1 to 15) to BRADD4<BR4K3:0> when + (16 − K)/16 division function is used.

Note 3: BRADD4<bit7:4> is always read as "1".

Note 4: Don't read from or write to BR4CR register during sending or receiving.

Figure 3.10.20 Baud Rate Generator Control (Channel 4, BR4CR, BRADD4)

SC4BUF (004CH)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receiving)

Note: Prohibit read-modify-write for SC4BUF.

Figure 3.10.21 Serial Transmission/Receiving Buffer Registers (Channel 4, SC4BUF)

P9FC (001DH)

	7	6	5	4	3	2	1	0
Bit symbol			P95F		P93F	P92F		P90F
Read/Write			W		W	W		W
After reset			0		0	0		0
Function			0: Port 1: SCLK1		0: Port 1: TXD1	0: Port 1: SCLK0		0: Port 1: TXD0

Setting TXD0 output

0	Port
1	TXD0 output

Setting SCLK0 output

0	Port
1	SCLK0 output

Setting TXD1 output

0	Port
1	TXD1 output

Setting SCLK1 output

0	Port
1	SCLK1 output

Note: Prohibit read-modify-write.

Figure 3.10.22 Port 9 Function Register (P9FC)

ODE (0058H)

	7	6	5	4	3	2	1	0
Bit symbol							ODE1	ODE0
Read/Write							R/W	
After reset							0	0
Function							P93 0: CMOS 1: Open drain	P90 0: CMOS 1: Open drain

Setting P90 as open-drain output

0	CMOS output
1	Open-drain output

Setting P93 as open-drain output

0	CMOS output
1	Open-drain output

Note: ODE<bit7:2> is read as “1”.

Figure 3.10.23 Port 9 Open-drain Enable Register (ODE)

P6FC (0016H)

	7	6	5	4	3	2	1	0
Bit symbol		P65F	P65F		P63F	P62F		P60F
Read/Write		W	W		W	W		W
After reset		0	0		0	0		0
Function		0: Port 1: TXD4	0: Port 1: SCLK3		0: Port 1: TXD3	0: Port 1: SCLK2		0: Port 1: TXD2

Setting TXD2 output

0	Port
1	TXD2 output

Setting SCLK2 output

0	Port
1	SCLK2 output

Setting TXD3 output

0	Port
1	TXD3 output

Setting TXD4 output

0	Port
1	TXD4 output

Setting SCLK3 output

0	Port
1	SCLK3 output

Note: Prohibit read-modify-write.

Figure 3.10.24 Port 6 Function Register (P6FC)

3.10.2 Configuration

Figure 3.10.25 shows the block diagram of the serial channel 0.

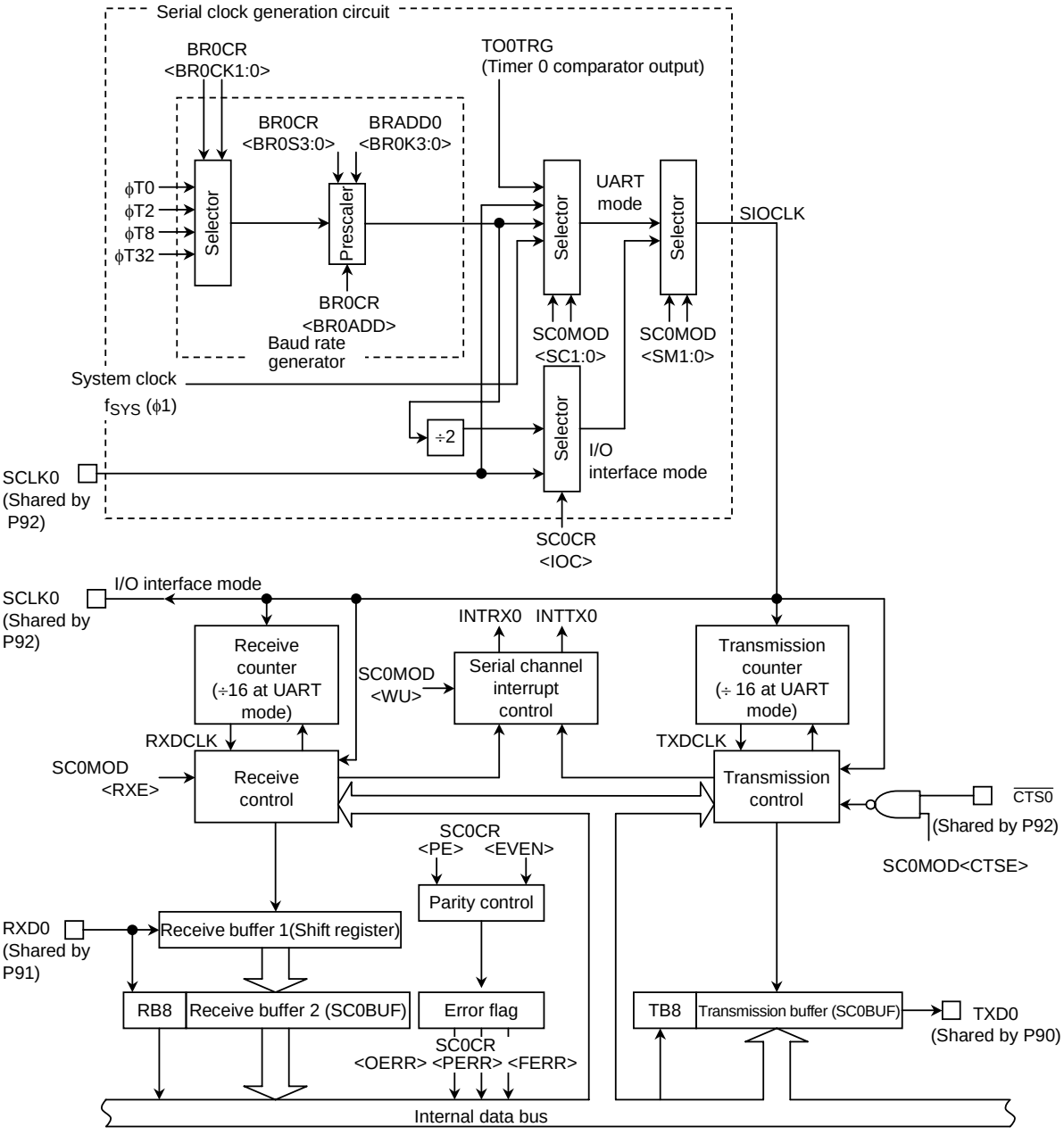
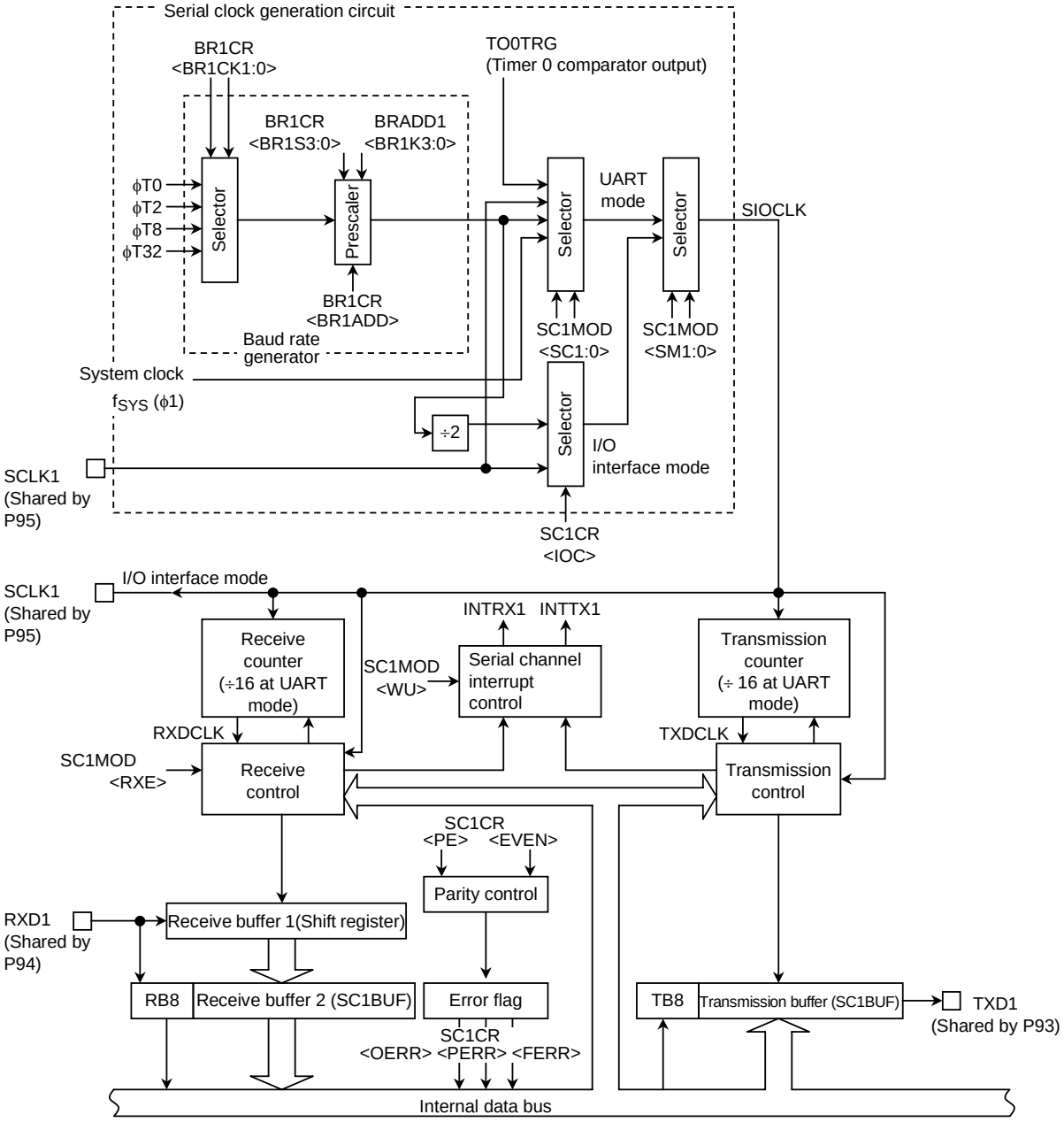


Figure 3.10.25 Block Diagram of the Serial Channel 0

Figure 3.10.26 shows the block diagram of the serial channel 1.



Note: No handshake function in channel 1.

Figure 3.10.26 Block Diagram of the Serial Channel 1

Figure 3.10.27 shows the block diagram of the serial channel 2.

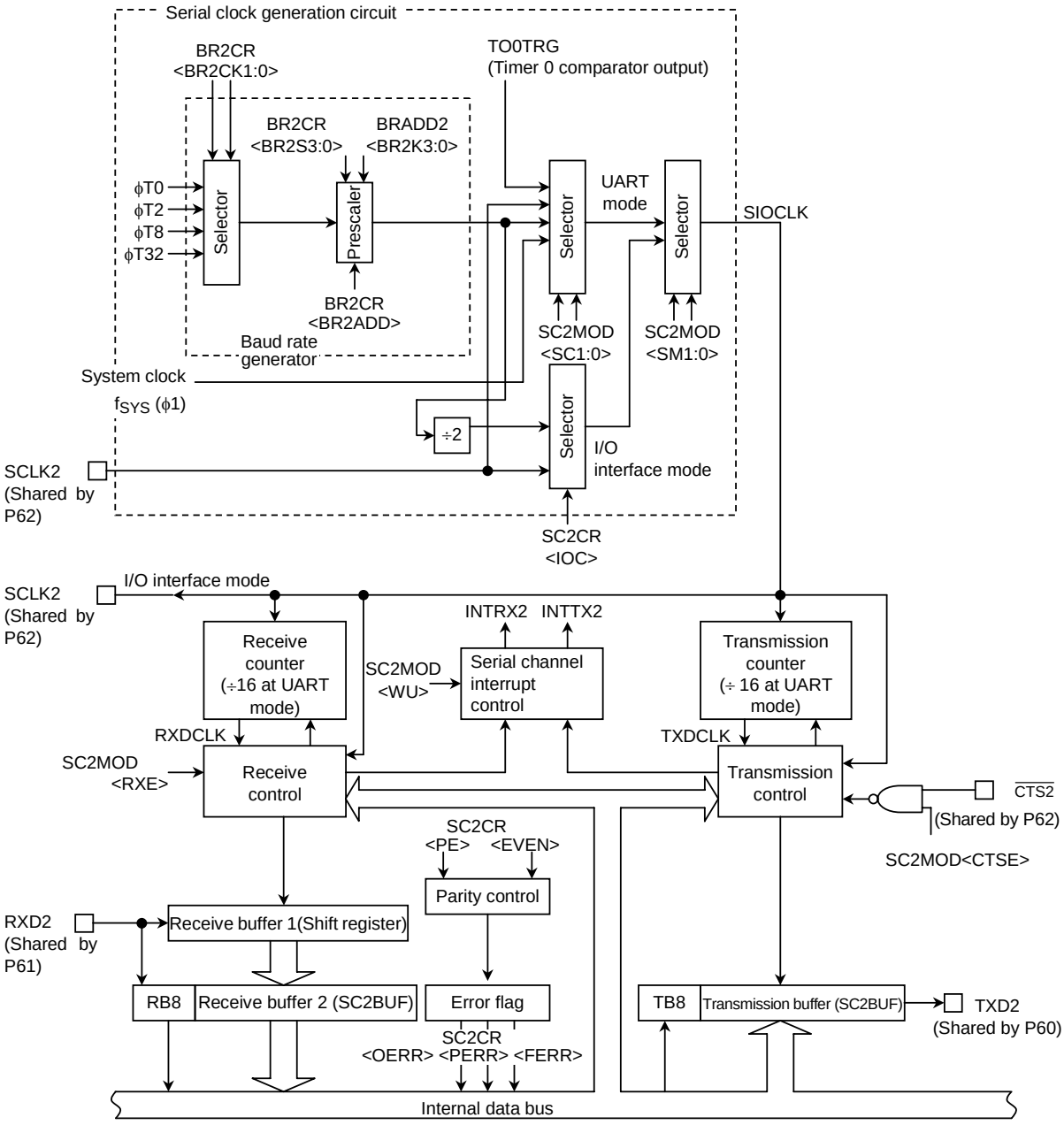


Figure 3.10.27 Block Diagram of the Serial Channel 2

Figure 3.10.28 shows the block diagram of the serial channel 3.

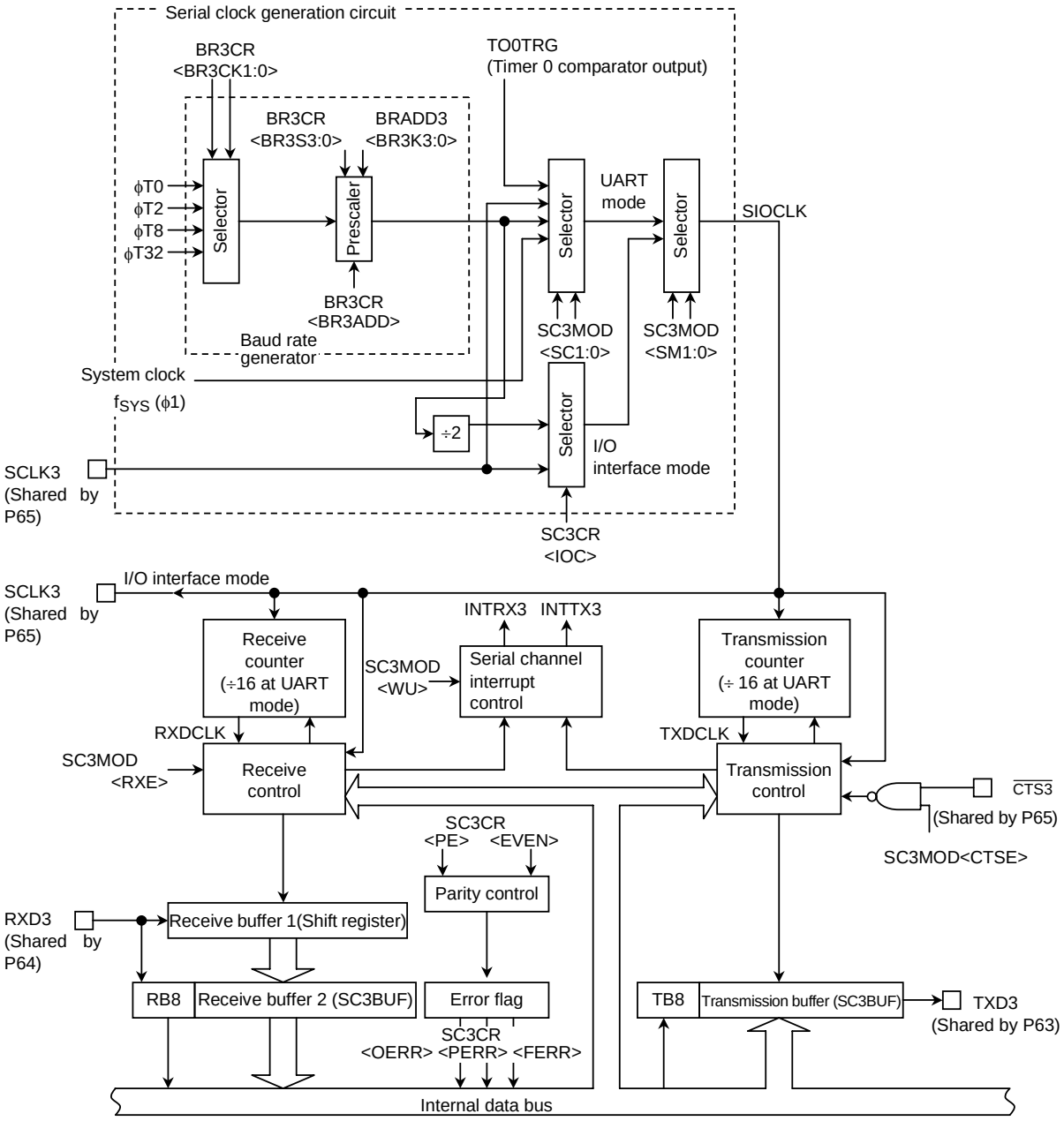
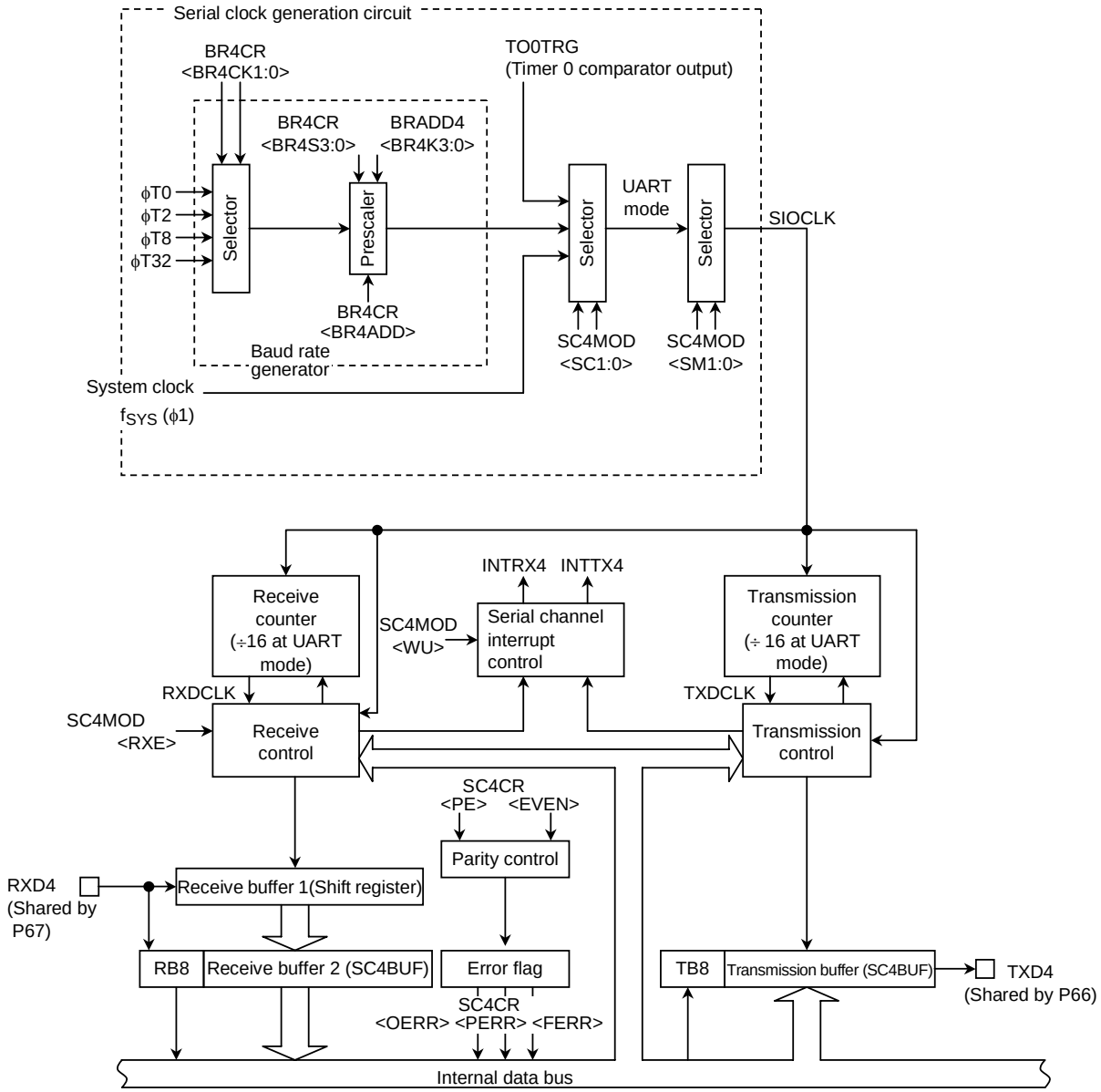


Figure 3.10.28 Block Diagram of the Serial Channel 3

Figure 3.10.29 shows the block diagram of the serial channel 4.



Note 1: No I/O interface mode in serial channel 4.

Note 2: No handshake function in serial channel 4.

Figure 3.10.29 Block Diagram of the Serial Channel 4

1. Prescaler, Prescaler clock select

There are 9-bit prescaler and prescaler clock selection registers to generate input clock for 8-bit timers 0 and 1, 16-bit timers 4 and 5, and Serial Interfaces 0 to 4.

Figure 3.10.30 shows the block diagram. Table 3.10.1 shows prescaler clock resolution into the baud rate generator.

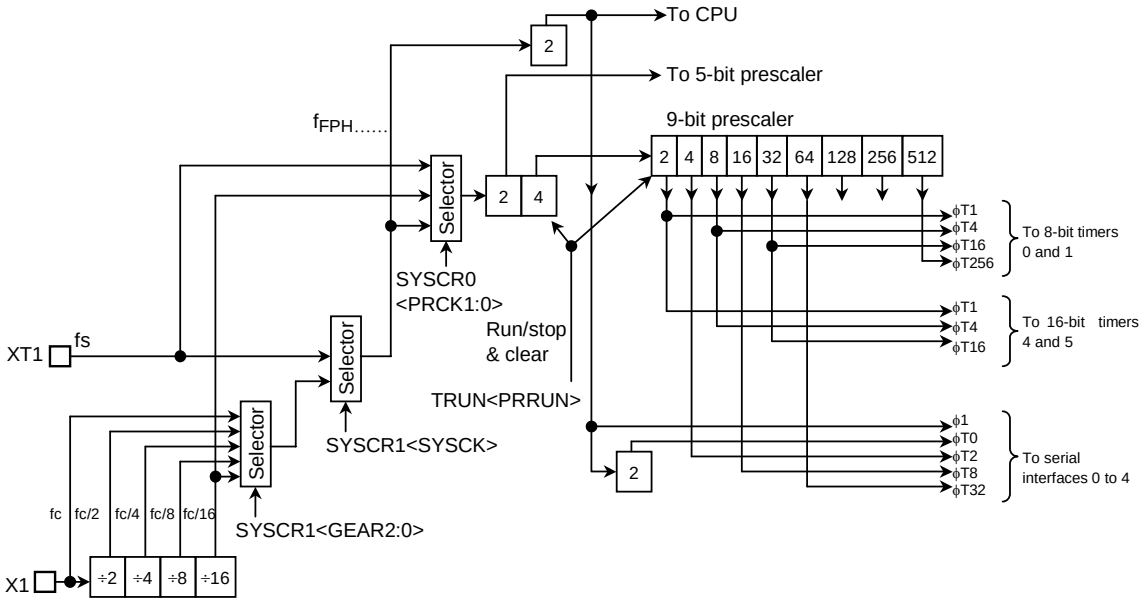


Figure 3.10.30 The Block Diagram of Prescaler

Table 3.10.1 Prescaler Clock Resolution to Baud Rate Generator

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	Prescaler Output Clock Resolution			
			φT0	φT2	φT8	φT32
1 (fs)	00 (f_{FPH})	XXX	$fs/2^2$	$fs/2^4$	$fs/2^6$	$fs/2^8$
0 (fc)		000 (fc)	$fc/2^2$	$fc/2^4$	$fc/2^6$	$fc/2^8$
		001 (fc/2)	$fc/2^3$	$fc/2^5$	$fc/2^7$	$fc/2^9$
		010 (fc/4)	$fc/2^4$	$fc/2^6$	$fc/2^8$	$fc/2^{10}$
		011 (fc/8)	$fc/2^5$	$fc/2^7$	$fc/2^9$	$fc/2^{11}$
		100 (fc/16)	$fc/2^6$	$fc/2^8$	$fc/2^{10}$	$fc/2^{12}$
XXX	01 (Low-frequency clock)	XXX	–	$fs/2^4$	$fs/2^6$	$fs/2^8$
XXX	10 (fc/16 clock)	XXX	–	$fc/2^8$	$fc/2^{10}$	$fc/2^{12}$

XXX: Don't care, –: Can not use

Note: The fc/16 clock as a prescaler clock can not be used when the fs is used as a system clock.

The clock selected among f_{FPH} clock, fc/16 clock, and fs clock is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0<PRCK1:0>.

Resetting sets <PRCK1:0> to "00" and selects the f_{FPH} clock input divided by 4.

The baud rate generator selects between 4 clock inputs: ϕT0, ϕT2, ϕT8, and ϕT32 among the prescaler outputs.

The prescaler can be run or stopped by the timer operation control register TRUN<PRRUN>. Counting starts when <PRRUN> is set to "1". The prescaler is cleared to zero and stops operation when <PRRUN> is set to "0".

When the IDLE1 mode (only the oscillator operates) is used, set TRUN<PRRUN> to "0" to stop this prescaler before the "HALT" instruction is executed.

2. Baud rate generator

The baud rate generator is a circuit that generates transmission and receiving clocks to determine the transfer rate of the serial channel.

The input clock to the baud rate generator, ϕT0, ϕT2, ϕT8, or ϕT32, is generated by the 9-bit prescaler which is shared by the timers. One of these input clocks is selected by the baud rate generator control register BR0CR<BR0CK1:0>.

The baud rate generator includes a frequency divider, which divides frequency by 1, n + m/16 (n = 2 to 15, m = 0 to 15) to 16 values to determine the transfer rate.

The transfer rate is determined by setting BR0CR<BR0ADD, BR0S3:0> and BRADD0<BR0K3:0>.

- UART mode

(1) BR0CR<BR0ADD> = 0

Setting BRADD0<BR0K3:0> is ignored. The baud rate generator divides the selected prescaler clock by N which is set to BR0CK<BR0S3:0>. (N = 1, 2, 3 ...16)

(2) BR0CR<BR0ADD> = 1

N + (16 – K)/16 division function is enabled. The baud rate generator divides the selected prescaler clock by N + (16 – K)/16 according to N set to BR0CR<BR0S3:0> (N = 2, 3...15) and K set to BRADD0<BR0K3:0> (K = 1, 2, 3...15).

Note: At N = 1 or 16, N + (16 – K)/16 division function is disabled. Set BR0CR<BR0ADD> to "0".

- I/O interface mode

N + (16 – K)/16 division function is not available in I/O interface mode. Set BK0CR<BR0ADD> to "0" before dividing by N.

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

- UART mode

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

- I/O interface mode

$$\text{Transfer rate} = \frac{\text{Input clock of baud rate generator}}{\text{Frequency divisor of baud rate generator}} \div 2$$

- Integer divisor (N divisor)

For example, when the source clock (fc) is 12.288 MHz, the input clock is ϕT2 (fc/16), and frequency divisor is N (BR0CR<BR0S3:0>) = 5 BR0CR<BRADD0> = 0, the transfer rate in UART mode becomes as follows:

* Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: System clock

$$\text{Baud rate} = \frac{fc/16}{5} \div 16$$
$$= 12.288 \times 10^6 \div 16 \div 5 \div 16 = 9600 \text{ (bps)}$$

Note: + (16 – K)/16 division function is disabled, and setting BRADD0<BR0K3:0> is invalid.

- N + (16 – K)/16 divisor (Only in UART mode)

Accordingly, when source clock (fc) is 4.8 MHz, the input clock is ϕT0 , and frequency
divisor is N (BR0CR<BR0S3:0>) = 7, "K" (BRADD0<BR0K3:0>) = 3,
BR0CR<BRADD0> = 1, the transfer rate in UART mode becomes as follows:

* Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: System clock

$$\text{Baud rate} = \frac{fc/4}{7 + (16 - 3)/16} \div 16$$
$$= 4.8 \times 10^6 \div 4 \div (7 + 13/16) \div 16 = 9600 \text{ (bps)}$$

Table 3.10.2, 3.10.3 show the examples of the transfer rate in UART mode.

Additionally, the external clock input is available in the serial clock. (Serial channel 0 to 3). How to calculate a baud rate is explained below.

- UART mode

Baud rate = External clock input ÷ 16

It is necessary to satisfy (External clock input cycle) ≥ 4/fc + 20 [ns]

- I/O interface mode

Baud rate = External clock input

It is necessary to satisfy (External clock input cycle) ≥ 16/fc [ns].

Table 3.10.2 Selection of Transfer Rate (1) (when baud rate generator is used)

Unit (kbps)

fc [MHz]	Input Clock / Frequency Divisor	φT0	φT2	φT8	φT32
9.830400	2	76.800	19.200	4.800	1.200
	4	38.400	9.600	2.400	0.600
	8	19.200	4.800	1.200	0.300
	0	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
	A	19.200	4.800	1.200	0.300
14.745600	3	76.800	19.200	4.800	1.200
	6	38.400	9.600	2.400	0.600
	C	19.200	4.800	1.200	0.300

Note 1: Transfer rates in I/O interface mode are 8 times faster than the values given in the above table.

Note 2: This table is calculated when fc is selected as a system clock, the clock gear is set for fc, and the system clock as the prescaler clock input.

Table 3.10.3 Selection of Transfer Rate (1) (when timer 0 (Input clock φt1) is used)

Unit (kbps)

fc / TREG0	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	96		76.8	62.5	48
2H	48		38.4	31.25	24
3H	32	31.25			16
4H	24		19.2		12
5H	19.2				9.6
8H	12		9.6		6
AH	9.6				4.8
10H	6		4.8		3
14H	4.8				2.4

How to calculate the transfer rate (when timer 0 is used):

$$\text{Transfer rate} = \frac{\text{The clock frequency selected by the register SYSCR0<PRCK1:0>}}{\text{TREG0} \times \underset{\substack{\uparrow \\ \text{(when timer 0 (Input clock } \phi\text{T1) is used)}}}{8} \times 16}$$

Note 1: Timer 0 match detect signal cannot be used as the transfer clock in I/O interface mode.

Note 2: This table is calculated when fc is selected as a system clock, the clock gear is set for fc, and f_{FPH} as the prescaler clock input.

3. Serial clock generation circuit

This circuit generates the basic clock for transmitting and receiving data.

- I/O interface mode (Channel 1 only)

When in SCLK output mode with the setting of SC0CR<IOC> = "0", the basic clock will be generated by dividing the output of the baud rate generator by 2 as described before. When in SCLK input mode with the setting of SC0CR<IOC> = "1", the rising edge or falling edge will be detected according to the setting of the SC0CR<SCLKS> register to generate the basic clock.

- UART mode

The setting of SC0MOD<SC1:0>, will select between the baud rate generator clock, internal clock ϕ1 (Max 625 Kbps at fc = 20 MHz), or the match detect signal from timer 0 or the external clock (Channel 0 to 3) to generate the basic clock SIOCLK.

4. Receiving counter

The receiving counter is a 4-bit binary counter used in asynchronous communication (UART) mode and counts up according to the SIOCLK clock. 16 pulses of SIOCLK are used for receiving 1 bit of data, and the data bit is sampled three times at the 7th, 8th and 9th clock.

With these three samples, the received data bit is evaluated by the majority rule.

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 such that the received data bit is determined to be "0".

5. Receiving control

- I/O interface mode

When in SCLK output mode with the setting of SC0CR<IOC> = "0", the RXD0 signal will be sampled at the rising edge of the shift clock which is output to the SCLK0 pin.

When in SCLK input mode with the setting SC1CR<IOC> = "1", the RXD0 signal will be sampled at the rising edge or falling edge of the SCLK0 input according to the setting of the SC0CR<SCLKS> register.

- UART mode

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

The data being received is also evaluated by the majority rule.

6. Receiving buffer

To prevent an overrun error, the receiving buffer has a double buffer structure.

Received data is stored bit by bit in receiving buffer 1 (Shift register type). When 7 bits or 8 bits of data are stored in receiving buffer 1, the stored data is transferred to receiving buffer 2 (SC0BUF) generating an interrupt INTRX0. The CPU reads only receiving buffer 2 (SC0BUF). Even before the CPU reads receiving buffer 2 (SC0BUF), the received data can be stored in receiving buffer 1. However, unless receiving buffer 2 (SC0BUF) is read before all bits of the next data are received by receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of receiving buffer 1 will be lost, although the contents of the receiving buffer 2 and SC0CR<RB8> are 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 SC0CR<RB8>.

When in 9-bit UART mode, the wakeup function of the slave controller is enabled by setting SC0MOD<WU> to "1", and interrupt INTRX0 occurs only when SC0CR<RB8> is set to "1".

7. Transmission counter

The transmission counter is a 4-bit binary counter which is used in asynchronous communication (UART) mode and, like a receiving counter, counts by the SIOCLK clock which generates TXDCLK every 16 clock pulses.

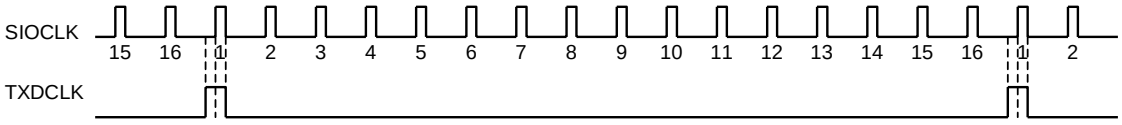


Figure 3.10.31 Generation of Transmission Clock

8. Transmission controller

- I/O interface mode

In SCLK output mode with the setting of SC0CR<IOC> = "0", the data in the transmission buffer is output bit by bit to TXD0 pin at the rising edge of the shift clock which is output from the SCLK0 pin.

In SCLK input mode with the setting of SC0CR<IOC> = "1", the data in the transmission buffer is output bit by bit to the TXD0 pin at the rising edge or falling edge of the SCLK0 input according to the setting of the SC0CR<SCLKS> register.

- UART mode

When transmission data is written to 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

Serial channel 0, 2, 3 has a $\overline{CTS}$ pin. Using this pin, data can be sent in units of one frame; thus, overrun errors can be avoided. The handshake function is enabled/ disabled by SC0MOD<CTSE>.

When the $\overline{CTS0}$ pin goes high, after completion of the current data send, data send is halted until the $\overline{CTS0}$ pin goes low again. When the INTTX0 Interrupt is generated, it requests the next data send to the CPU.

Though there is no $\overline{RTS}$ pin, a handshake function can be easily configured by setting any port assigned to be the $\overline{RTS}$ function. The $\overline{RTS}$ should be output "High" to request send data halt after data receive is completed by software in the RXD interrupt routine.

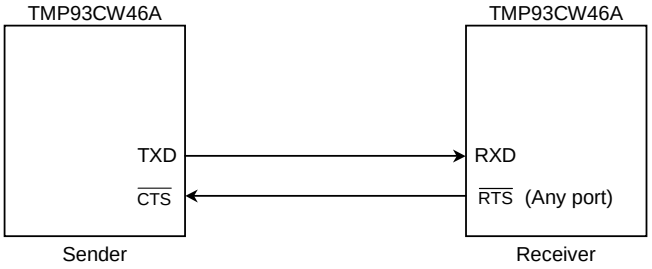
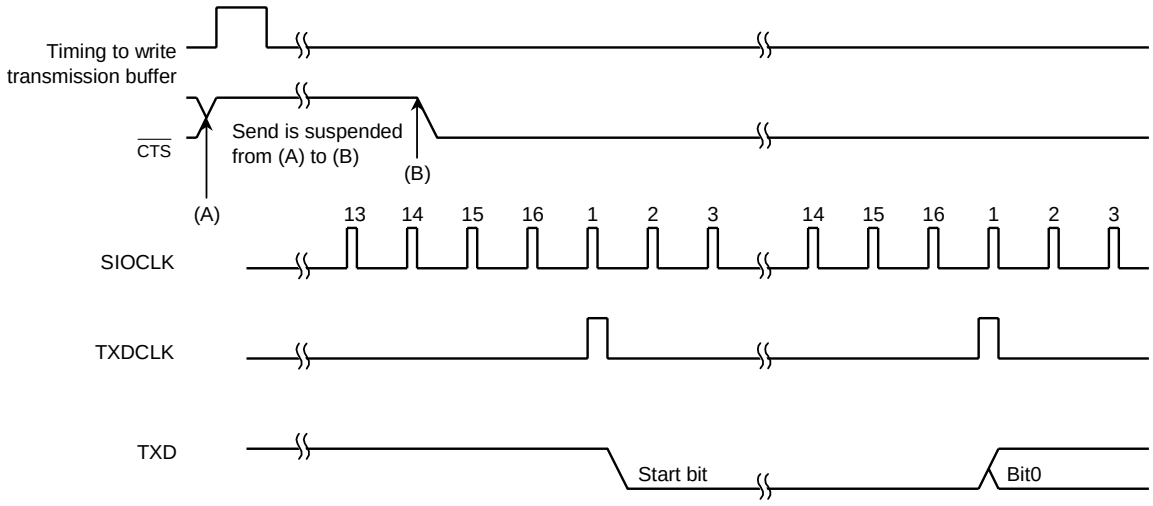


Figure 3.10.32 Handshake Function



Note 1: If the $\overline{CTS}$ signal rises during transmission, the next data is not sent after the completion of the current transmission.

Note 2: Transmission starts at the first TXDCLK clock falling edge after the $\overline{CTS}$ signal falls.

Figure 3.10.33 Timing of $\overline{CTS}$ (Clear to send)

9. Transmission buffer

   The transmission buffer (SC0BUF) shifts out and sends the transmission data written from the CPU from the least significant bit (LSB) in order. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX0 interrupt.

10. Parity control circuit

    When the serial channel control register SC0CR<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 modes. With SC0CR<EVEN> register, even or odd parity can be selected.

    For transmission, parity is automatically generated according to the data written in the transmission buffer SC0BUF. The data is transmitted after the parity bit is stored in SC0BUF<TB7> when in 7-bit UART mode or in SC0MOD<TB8> when in 8-bit UART mode. <PE> and <EVEN> must be set before the transmission data is written to the transmission buffer.

    For receiving, data are shifted in the receiving buffer 1, the parity is added after the data is transferred to receiving buffer 2 (SC0BUF), and then compared with SC0BUF<RB7> when in 7-bit UART mode and with SC0MOD<RB8> when in 8-bit UART mode. If they are not equal, a parity error occurs and SC0CR<PERR> flag is set.

11. 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 is stored in receiving buffer 2 (SC0BUF), an overrun error will occur.

    2) Parity error <PERR>

       The parity generated for the data shifted in receiving buffer 2 (SC0BUF) 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.

l. Signal generating timing

1) In UART mode

Receive

Mode	9-Bit	8-Bit + Parity	8-Bit, 7-Bit + Parity, 7-Bit
Timing for interrupt generation	Around center of bit8	Around center of parity bit	Around center of stop bit
Timing for framing generation	Around center of stop bit	Around center of stop bit	Around center of stop bit
Timing for parity error generation	–	Around center of parity bit	←
Timing for overrun error timing	Around center of bit8	Around center of parity bit	Around center of stop bit

Note: In 9-Bit and 8-Bit+Parity mode, 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.

Send

Mode	9-Bit	8-Bit + Parity	8-Bit, 7-Bit + Parity, 7-Bit
Timing for interrupt generation	Immediately before stop bit sent	←	←

2) In I/O interface mode

Timing for send interrupt generation	SCLK0 output mode	Immediately after rise of last SCLK0 signal (See Figure 3.10.36)
	SCLK0 input mode	Immediately after rise (Rising mode) or fall (Falling mode) of last SCLK0 signal (See Figure 3.10.36.)
Timing for receive interrupt generation	SCLK0 output mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF)) (See Figure 3.10.38.)
	SCLK0 input mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF)) (See Figure 3.10.39.)

3.10.3 Operational Description

(1) Mode 0 (I/O interface mode)

This mode is used to increase the number I/O pins for transmitting or receiving data to or from an external shifter register.

This mode includes the SCLK output mode to output synchronous clock SCLK and SCLK input mode to input external synchronous clock SCLK.

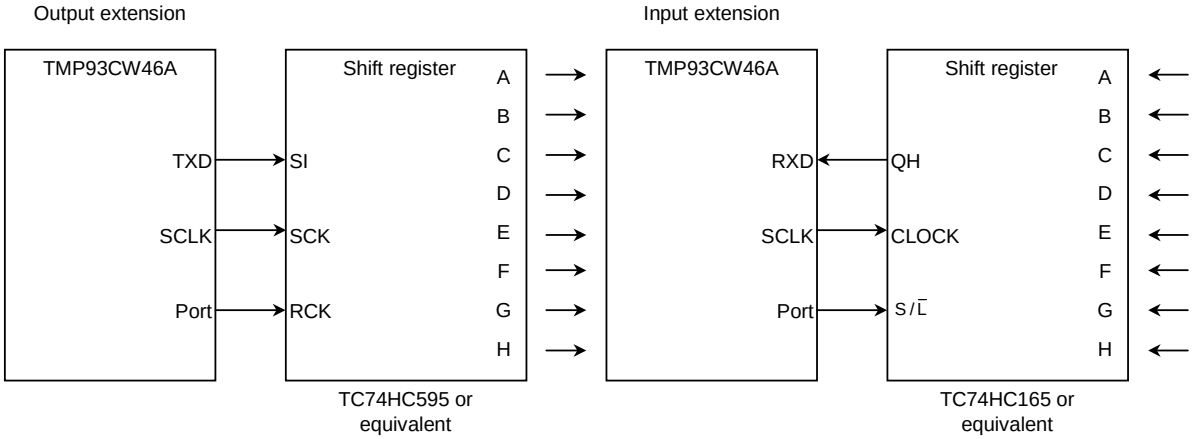


Figure 3.10.34 Example of SCLK Output Mode Connection

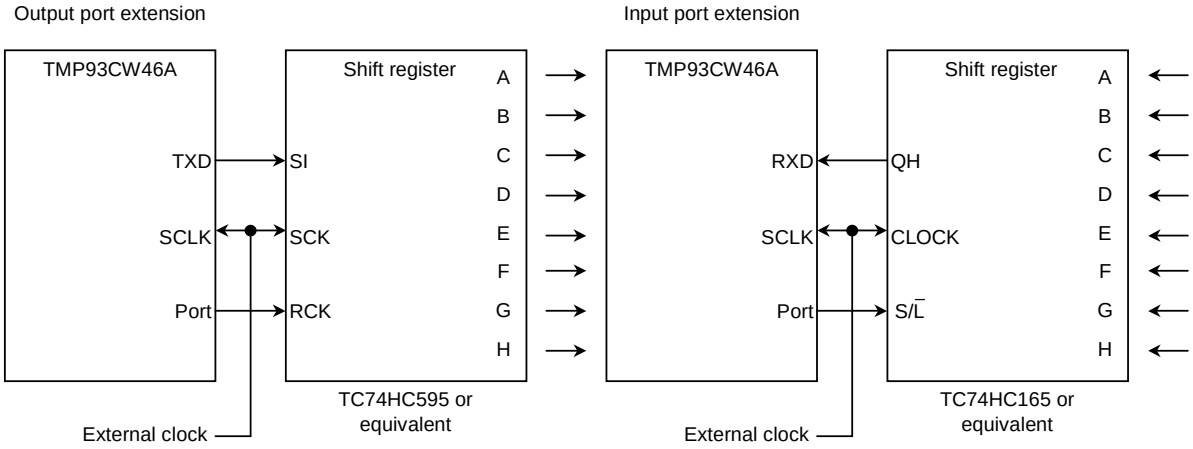


Figure 3.10.35 Example of SCLK Input Mode Connection

Note: Serial channel 4 has no I/O interface mode.

1. Transmission

In SCLK output mode, 8-bit data and synchronous clock are output from TXD0 pin and SCLK0 pin respectively, each time the CPU writes data to the transmission buffer. When all data is output, INTES0<ITX0C> will be set to generate the INTTX0 interrupt.

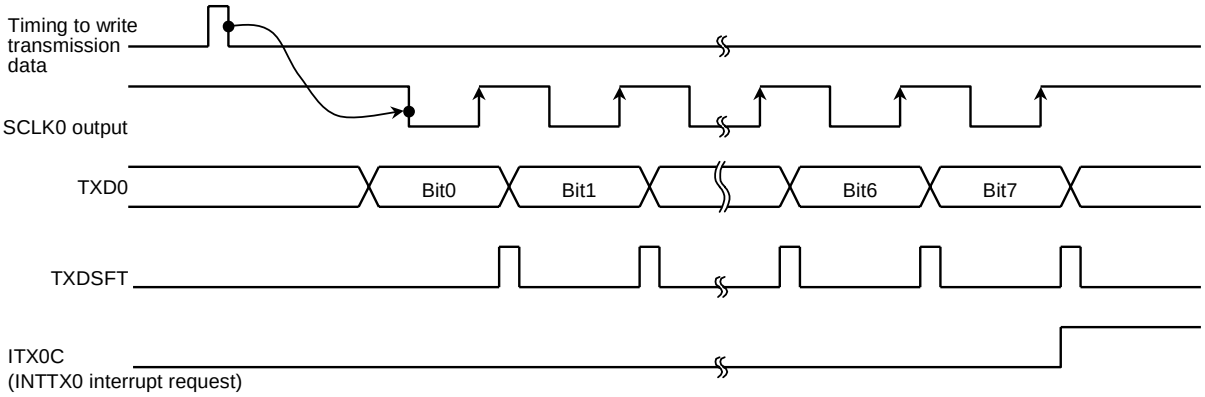


Figure 3.10.36 Transmitting Operation in I/O Interface Mode (SCLK output mode) (Channel 0)

In SCLK input mode, 8-bit data is output from TXD0 pin when SCLK0 input becomes active after data are written to the transmission buffer by CPU.

When all data is output, INTES0<ITX0C> will be set to generate INTTX0 interrupt.

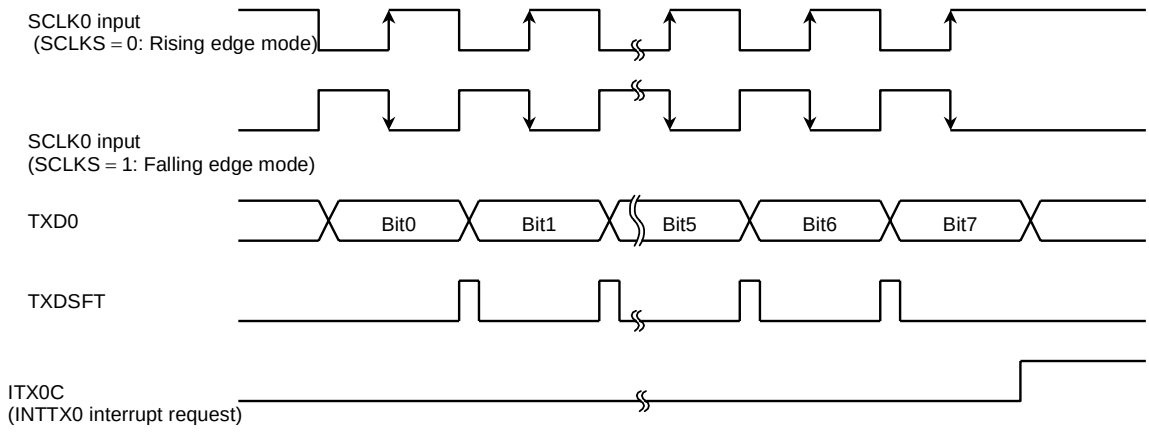


Figure 3.10.37 Transmitting Operation in I/O Interface Mode (SCLK input mode) (Channel 0)

2. Receiving

In SCLK output mode, the synchronous clock is outputted from SCLK0 pin and the data is shifted to receiving buffer 1. This starts when the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred to receiving buffer 2 (SC0BUF according to the timing shown below) and INTES0<IRX0C> will be set again to generate INTRX0 interrupt.

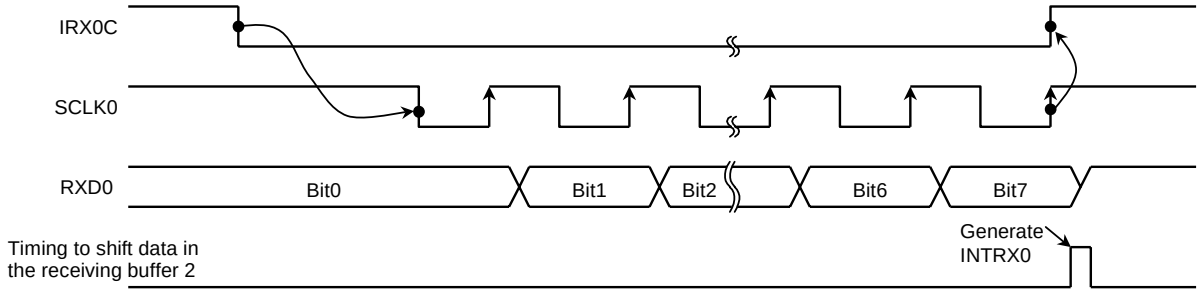


Figure 3.10.38 Receiving Operation in I/O Interface Mode (SCLK output mode) (Channel 0)

In SCLK input mode, the data is shifted to receiving buffer 1 when the SCLK input becomes active after the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data is received, the data will be shifted to receiving buffer 2 (SC0BUF according to the timing shown below) and INTES0<IRX0C> will be set again to generate INTRX0 interrupt.

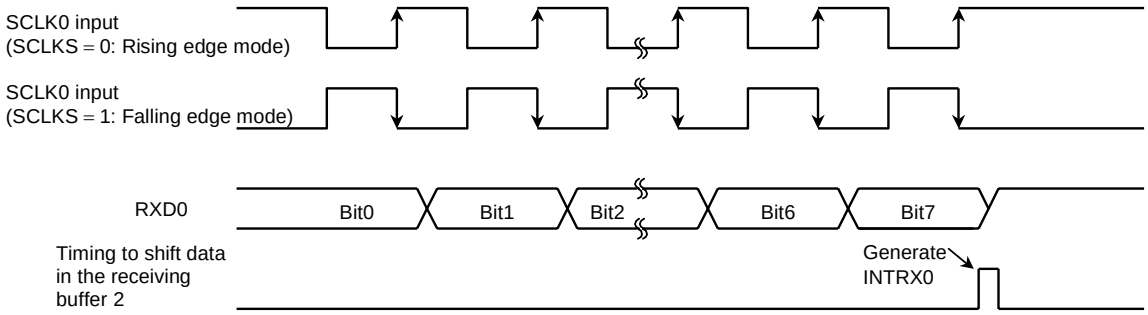


Figure 3.10.39 Receiving Operation in I/O Interface Mode (SCLK input mode) (Channel 0)

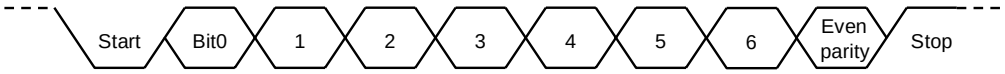
Note: For data receiving, the system must be placed in the receive enable state (SC0MOD<RXE> = “1”)

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

The 7-bit mode can be set by setting serial channel mode register SC0MOD<SM1:0> to "01".

In this mode, a parity bit can be added, and the addition of the parity bit can be enabled or disabled by serial channel control register SC0CR<PE>, and even parity or odd parity is selected by SC0CR<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. Channel 0 is explained here.



Direction of transmission (transmission rate: 2400 bps at fc = 12.288 MHz)

* Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: System clock

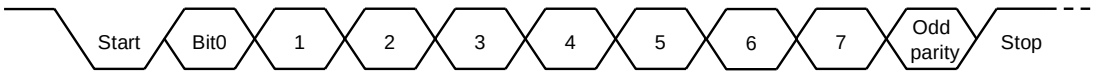
	7	6	5	4	3	2	1	0	
P9CR	X	X	–	–	–	–	–	1	Select P90 as the TXD0 pin.
P9FC	X	X	–	X	–	X	X	1	
SC0MOD	X	0	–	X	0	1	0	1	Set 7-bit UART mode.
SC0CR	X	1	1	X	X	X	0	0	Add even parity.
BR0CR	0	X	1	0	0	1	0	1	Set transfer rate at 2400 bps.
TRUN	1	X	–	–	–	–	–	–	Start the prescaler for the baud rate generator.
INTES0	1	1	0	0	–	–	–	–	Enable INTTX0 interrupt and set interrupt level 4.
SC0BUF	*	*	*	*	*	*	*	*	Set data for transmission.

X: Don't care, –: No change

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

The 8-bit UART mode can be specified by setting SC0MOD<SM1:0> to "10". In this mode, the parity bit can be added (The addition of a parity bit is enabled or disabled by SC0CR<PE>) and even parity or odd parity is selected by SC0CR<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.



Direction of transmission (transmission rate: 9600 bps at fc = 12.288 MHz)

* Clock condition
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: System clock

Main setting

		7	6	5	4	3	2	1	0	
P9CR	←	X	X	–	–	–	–	0	–	Select P91 (RXD0) as the input pin.
SC0MOD	←	–	0	1	X	1	0	0	1	Enable receiving in 8-bit UART mode.
SC0CR	←	X	0	1	X	X	X	0	0	Add odd parity.
BR0CR	←	0	X	0	1	0	1	0	1	Set transfer rate at 9600 bps.
TRUN	←	1	X	–	–	–	–	–	–	Start the prescaler for the baud rate generator.
INTES0	←	–	–	–	–	1	1	0	0	Enable INTRX0 interrupt and set interrupt level 4.

Interrupt processing

A_{CC} ← SC0CR AND 00011100
if $A_{CC} \neq 0$ then ERROR } Check for error.

A_{CC} ← SC0BUF Read the received data.

X: Don't care, –: No change

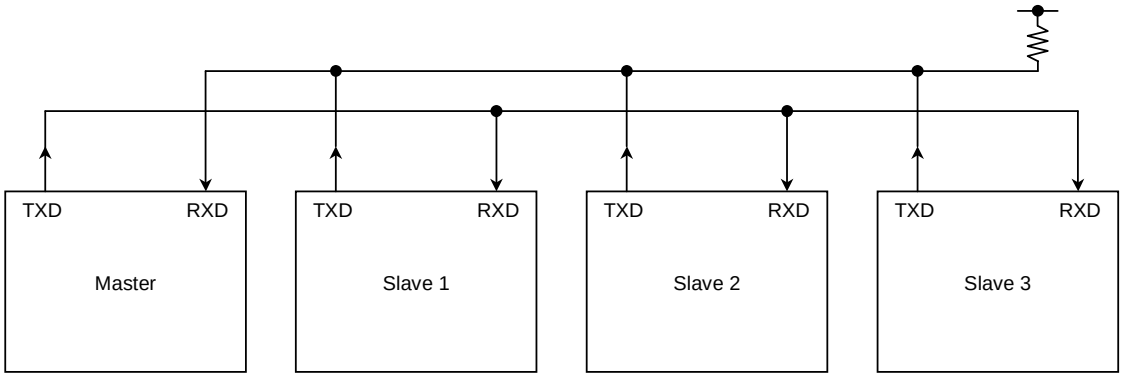
(4) Mode 3 (9-bit UART mode)

9-bit UART mode can be specified by setting SC0MOD<SM1:0> to "11". In this mode, parity bit cannot be added.

For transmission, the MSB (9th bit) is written in SC0MOD<TB8>. For receiving it is stored in SC0CR<RB8>. For writing and reading of the buffer, the MSB is read or written first then the rest of the data from SC0BUF.

Wakeup function

In 9-bit UART mode, the wakeup function of slave controllers is enabled by setting SC0MOD<WU> to "1". The interrupt INTRX0 occurs only when <RB8> = 1.

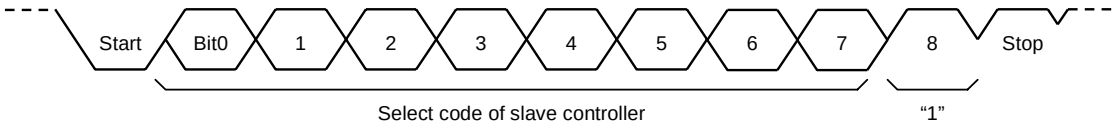


Note: TXD pin of the slave controllers must be in open drain output mode.

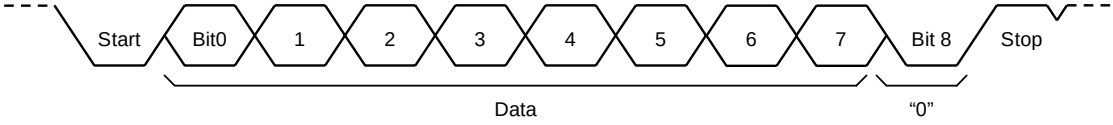
Figure 3.10.40 Serial Link using Wakeup Function

Protocol

1. Select the 9-bit UART mode for the master and slave controllers.
2. Set SC0MOD<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 (Bit8)<TB8> is set to "1".



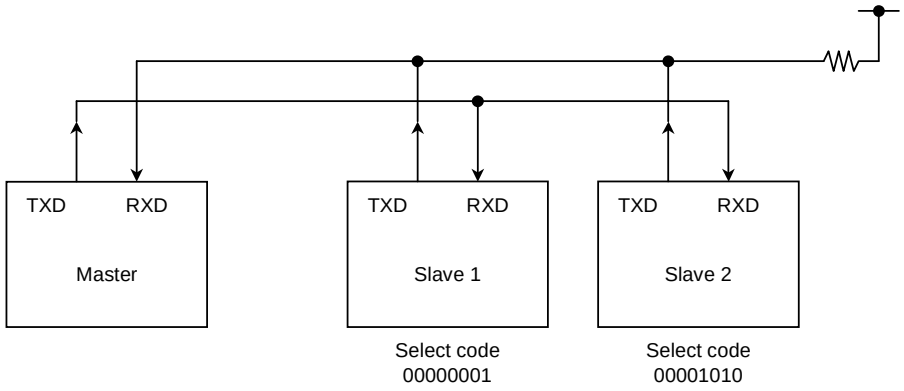
4. Each slave controller receives the above frame, and clears WU bit to "0" if the above select code matches its own select code.
5. The master controller transmits data to the specified slave controller whose SC0MOD<WU> bit is cleared to "0". The MSB (Bit8)<TB8> is cleared to "0".



6. The other slave controllers (with the <WU> bit remaining at "1") ignore the receiving data because their MSBs (Bit8 or <RB8>) are set to "0" to disable the interrupt INTRX0.

   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 ϕ1 as the transfer clock.



Since serial channels 0 and 1 operate in exactly the same way, channel 0 is used for the purposes of explanation.

- Setting the master controller

Main

P9CR	← X	X	–	–	–	–	0	1	Select P90 as TXD0 pin and P91 as RXD0 pin.
P9FC	← X	X	–	X	–	X	X	1	
INTES0	← 1	1	0	0	1	1	0	1	Enable INTTX0 and set the interrupt level 4. Enable INTRX0 and set the interrupt level 5.
SC0MOD	← 1	0	1	0	1	1	1	0	Set ϕ1 as the transmission clock in 9-bit UART mode.
SC0BUF	← 0	0	0	0	0	0	0	1	Set the select code for slave controller 1.

INTTX0 interrupt

SC0MOD	← 0	–	–	–	–	–	–	–	Sets TB8 to “0”.
SC0BUF	← *	*	*	*	*	*	*	*	Set data for transmission.

- Setting the slave controller

Main

P9CR	← X	X	–	–	–	–	0	1	Select P91 as RXD0 pin and P90 as TXD0 pin (Open-drain output).
P9FC	← X	X	–	X	–	X	X	1	
ODE	← X	X	X	X	X	X	–	1	
INTES0	← 1	1	0	1	1	1	1	0	Enable INTRX0 and INTTX0.
SC0MOD	← 0	0	1	1	1	1	1	0	Set <WU> to “1” in the 9-bit UART transmission mode with transfer clock ϕ1.

INTRX0 interrupt

A_{CC} ← SC0BUF

if A_{CC} = Select code

Then SC0MOD ← – – – 0 – – – – Clear <WU> to “0”.

3.11 Analog/Digital Converter

TMP93CW46A contains an analog/digital converter (AD converter) with 8-channel analog input that features 10-bit successive approximation.

Figure 3.11.1 shows the block diagram of the AD converter. 8-channel analog input pins (AN7 to AN0) are shared by input-only port P5 which also can be used as a general purpose input port.

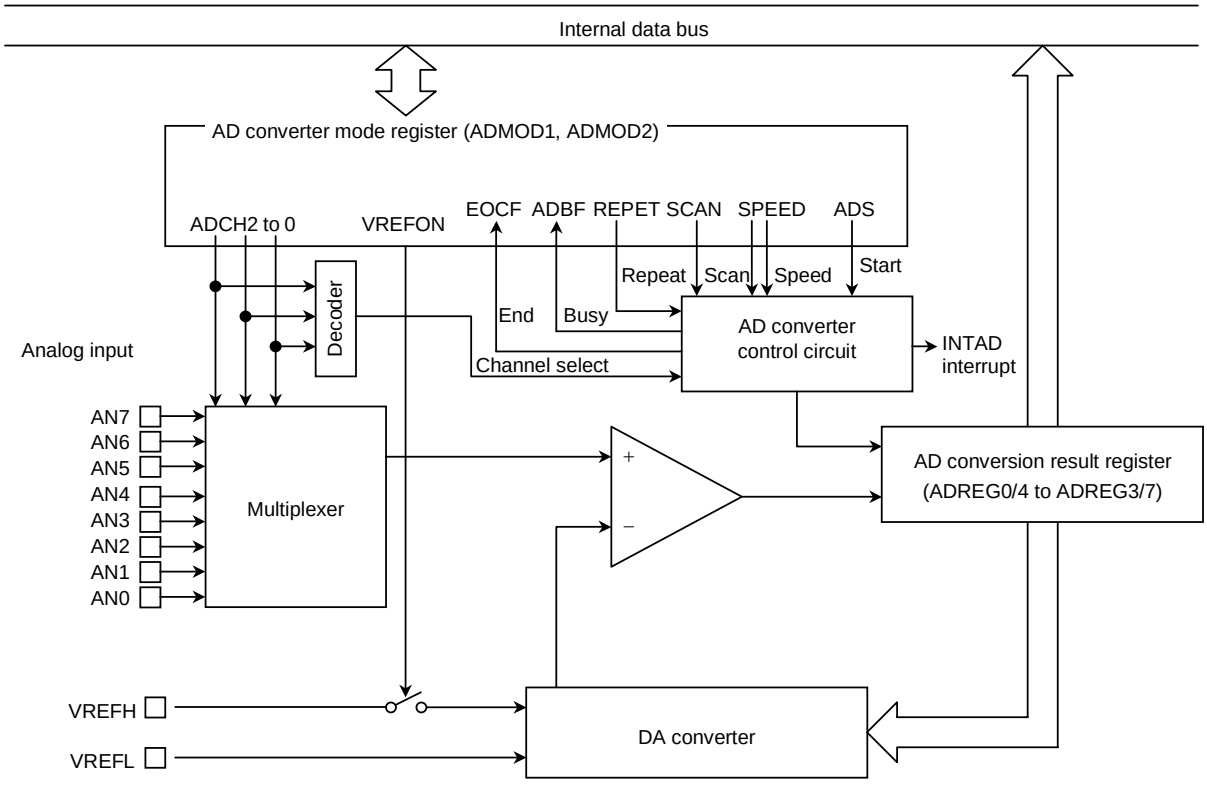


Figure 3.11.1 Block Diagram of AD Converter

Note 1: This AD converter does not have a built-in sample and hold circuit.

Note 2: When the power supply current is reduced in IDLE2, IDLE1, STOP mode, there is possible to set a standby enabling the internal comparator due to a timing. Stop operation of AD converter before execution of “HALT” instruction. And set ADMOD2<SPEED1:0> = “00”.

Note 3: The operation of AD converter is guaranteed only when fc (High-frequency oscillator) is used. (Not guaranteed when fs is used) It is guaranteed when with $f_{FPH} \geq 4$ MHz.

ADMOD1 (005EH)	7	6	5	4	3	2	1	0
Bit symbol	EOCF	ADBF	REPET	SCAN		ADS		
Read/Write	R		R/W			R/W		
After reset	0	0	0	0		0		
Function	AD conversion end flag 1: End	AD conversion busy flag 1: Busy	Repeat mode 0: Single mode 1: Repeat mode	Scan mode 0: Fixed channel mode 1: Channel scan mode		AD conversion start 1: Start Always read as "0".		

AD conversion start

0	–
1	Start conversion

Always "0" when data is read.

AD scan mode

0	Fixed channel mode
1	Channel scan mode

AD repeat mode

0	Single mode
1	Repeat mode

AD busy flag

0	Not busy
1	Busy

AD end flag

0	Not end
1	End

Note: ADMOD1<Bit3> and <Bit1:0> are read as "1".

Figure 3.11.2 AD Control Register (1/2)

	7	6	5	4	3	2	1	0
ADMOD2 (005FH) Bit symbol	VREFON		SPEED1	SPEED0		ADCH2	ADCH1	ADCH0
Read/Write	R/W		R/W			R/W		
After reset	1		0	0		0	0	0
Function	String resistance ON/OFF		Conversion speed 00: 160 states 01: 320 states 10: 640 states 11: 1280 states			Analog input channel selection		

Analog input channel selection

<ADCH2:0> \ <SCASN>	0	1
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	AN6	AN4 → AN5 → AN6
111	AN7	AN4 → AN5 → AN6 → AN7

Conversion speed

00	160 states (16 μs at 20 MHz)
01	320 states (32 μs at 20 MHz)
10	640 states (64 μs at 20 MHz)
11	1280 states (128 μs at 20 MHz)

Ladder resistance ON/OFF selection

0	Ladder resistance OFF
1	Ladder resistance ON

Note 1: Set <VREFON> bit to “1” before conversion start (Writing “1” to ADMOD1<ADS>).

Note 2: ADMOD2<bit6> and <bit3> are read as “1”.

Figure 3.11.3 AD Control Register (2/2)

ADREG04L (0060H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR01	ADR00						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of AD result for AN0 or AN4 are stored.							

ADREG04H (0061H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
After reset	Undefined							
Function	Upper 8 bits of AD result for AN0 or AN4 are stored.							

ADREG15L (0062H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR11	ADR10						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of AD result for AN1 or AN5 are stored.							

ADREG15H (0063H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
After reset	Undefined							
Function	Upper 8 bits of AD result for AN1 or AN5 are stored.							

Note: The result registers are used both as AN0 and AN4, AN1 and AN5, AN2 and AN6, AN3 and AN7. They are stored in to ADREG04, ADREG15, ADREG26 and ADREG37 respectively.

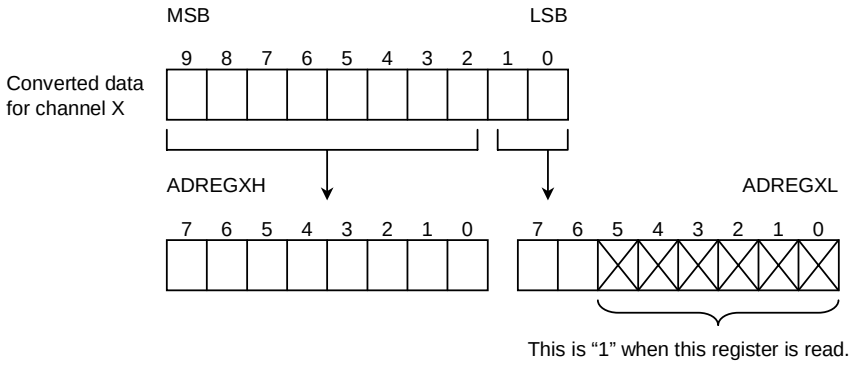


Figure 3.11.4 AD Conversion Result Register (ADREG04, ADREG15) (1/2)

ADREG26L (0064H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR21	ADR20						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of AD result for AN2 or AN6 are stored.							

ADREG26H (0065H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After reset	Undefined							
Function	Upper 8 bits of AD result for AN2 or AN6 are stored.							

ADREG37L (0066H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR31	ADR30						
Read/Write	R							
After reset	Undefined		1	1	1	1	1	1
Function	Lower 2 bits of AD result for AN3 or AN7 are stored.							

ADREG37H (0067H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After reset	Undefined							
Function	Upper 8 bits of AD result for AN3 or AN7 are stored.							

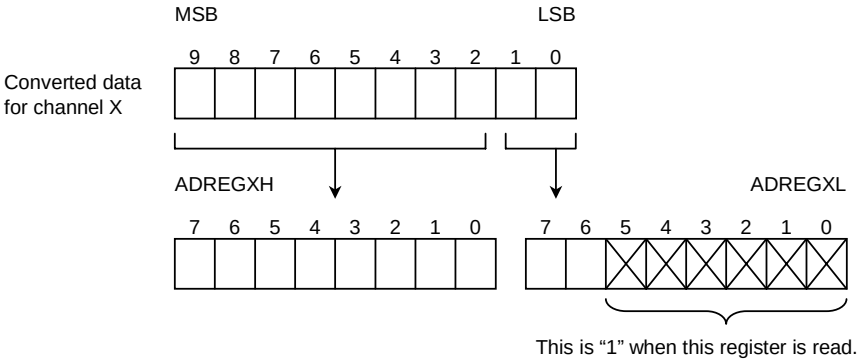


Figure 3.11.5 AD Conversion Result Register (ADREG26, ADREG37) (2/2)

3.11.1 Operation

(1) Analog reference voltage

The high analog reference voltage is applied to the VREFH pin, and the low analog reference voltage is applied to VREFL pin.

The reference voltage between VREFH and VREFL is divided by 1024 (Using string resistance) and compared with the analog input voltage for AD conversion.

The switch between VREFH and VREFL can be cut (OFF) by writing "0" to ADMOD2<VREFON>.

When the conversion can be started when <VREFON> = "0", a "1" must be written to <VREFON> and wait for 3 μs that the internal reference voltage is stable (Regardless to fc) before writing "1" to ADMOD1<ADS>.

(2) Analog input channels

The analog input channel is selected by ADMOD2<ADCH2:0>. However, which channel to select depends on the operation mode of the AD converter.

In fixed analog input mode, one channel is selected by <ADCH2:0> among eight pins: AN0 to AN7.

In analog input channel scan mode, the number of channels to be scanned is specified by <ADCH2:0>, such as only AN0, AN0 → AN1, AN0 → AN1 → AN2, AN0 → AN1 → AN2 → AN3, only AN4, AN4 → AN5, AN4 → AN5 → AN6 → , AN4 → AN5 → AN6 → AN7.

When reset, the AD conversion channel register will be initialized to ADMOD<ADCH2:0> = 000, so that the AN0 pin will be selected.

The pins which are not used as analog input channels can be used as ordinary input port pins on port P5.

(3) Starting AD conversion

AD conversion starts when "1" is written to AD conversion register ADMOD1<ADS>. When conversion starts, conversion busy flag ADMOD1<ADBF> which indicates "conversion is in progress" will be set to "1".

(4) AD conversion mode

Both fixed AD conversion channel mode and conversion channel scan mode have two conversion modes, single and repeat conversion modes.

In fixed channel repeat mode, conversion of the specified single channel is executed repeatedly.

In scan repeat mode, scanning from AN0, … → AN3 is executed repeatedly.

The AD conversion mode is selected by ADMOD1<REPET, SCAN>.

(5) AD conversion speed selection

There are four AD conversion speed modes. The selection is determined by ADMOD2<SPEED1:0> register.

When reset, ADMOD2<SPEED1:0> will be initialized to "00", so that the 160 state conversion mode will be selected. (16 μs at 20 MHz)

(6) AD conversion end and interrupt

- AD conversion single mode

  ADMOD1<EOCF> for AD conversion end will be set to "1", ADMOD1<ADBF> busy flag will be reset to "0", and INTAD interrupt will be enabled when AD conversion of specified channel ends in fixed conversion channel mode or when AD conversion of the last channel ends in channel scan mode.

- AD conversion repeat mode

  For both fixed conversion channel mode and conversion channel scan mode, INTAD should be disabled when in repeat mode. Always set the INTE0AD at "000" to disable the interrupt request.

  Write "0" to ADMOD1<REPET> to end the repeat mode. Then the repeat mode will be exited as soon as the conversion in progress completed.

(7) Storing the AD conversion result

The results of AD conversion are stored in ADREG04 to ADREG37 registers for each channel. The result registers are used both as AN0 and AN4, AN1 and AN5, AN2 and AN6, AN3 and AN7.

However, the current conversion data can not determine which channels.

In repeat mode, the registers are updated whenever conversion ends.

ADREG04 to ADREG37 are read-only registers.

(8) Reading the AD conversion result

The results of AD conversion are stored in ADREG04 to ADREG37 registers.

When the contents of one of the lower 2-bit registers ADREG04L, ADREG15L, ADREG26L, and ADREG37L are read, ADMOD1<EOCF> will be cleared to "0".

<EOCF> is not cleared to "0" when the contents of one of the upper 8 bits registers ADREG04H, ADREG15H, ADREG26H, and ADREG37H are read.

Setting example: a. When the analog input voltage of the AN3 pin is AD converted by 160 states speed and the result is transferred to the memory address 0100H by AD interrupt INTAD routine.

Main setting

INTE0AD	← 1	1	0	0	–	–	–	–	Enable INTAD and set interrupt level 4.
ADMOD2	← 1	X	0	0	X	0	1	1	Specify AN3 pin as an analog input channel and starts
ADMOD1	← X	X	0	0	X	1	X	X	AD conversion in 160 states speed mode.

INTAD routine

WA	← ADREG37	Read ADREG37L and ADREG37H values and write to WA (16 bits).
WA	>> 6	Right-shifts WA six times and writes 0 in upper bits.
(000100H)	← WA	Writes contents of WA in memory at 0100H.

b. When the analog input voltage of the AN4 to AN7 pins (4 pins) are AD converted by 320 states speed and set the channel scan and repeat mode.

Main setting

INTE0AD	← 1	0	0	0	–	–	–	–	INTAD disable.
ADMOD2	← 1	X	0	1	0	1	1	1	Specify AN4 to AN7 pins as input channel and scan and
ADMOD1	← X	X	1	1	X	1	0	0	repeat mode and starts AD conversion.

X: Don't care, –: No change

3.12 Watchdog Timer (Runaway detecting timer), Warm-up Timer

TMP93CW46A contains a watchdog timer for runaway detection.

The watchdog timer (WDT) is used to return the CPU to a 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 to notify the CPU of the malfunction, and outputs "0" externally from watchdog timer out pin $\overline{\text{WDTOUT}}$ to notify the peripheral devices of the malfunction.

Connecting the watchdog timer detect signal to the reset pin internally forces a reset.

This watchdog timer consists of 7-stage and 15-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 before changing the system clock.

3.12.1 Configuration

Figure 3.12.1 shows the block diagram of the watchdog timer (WDT).

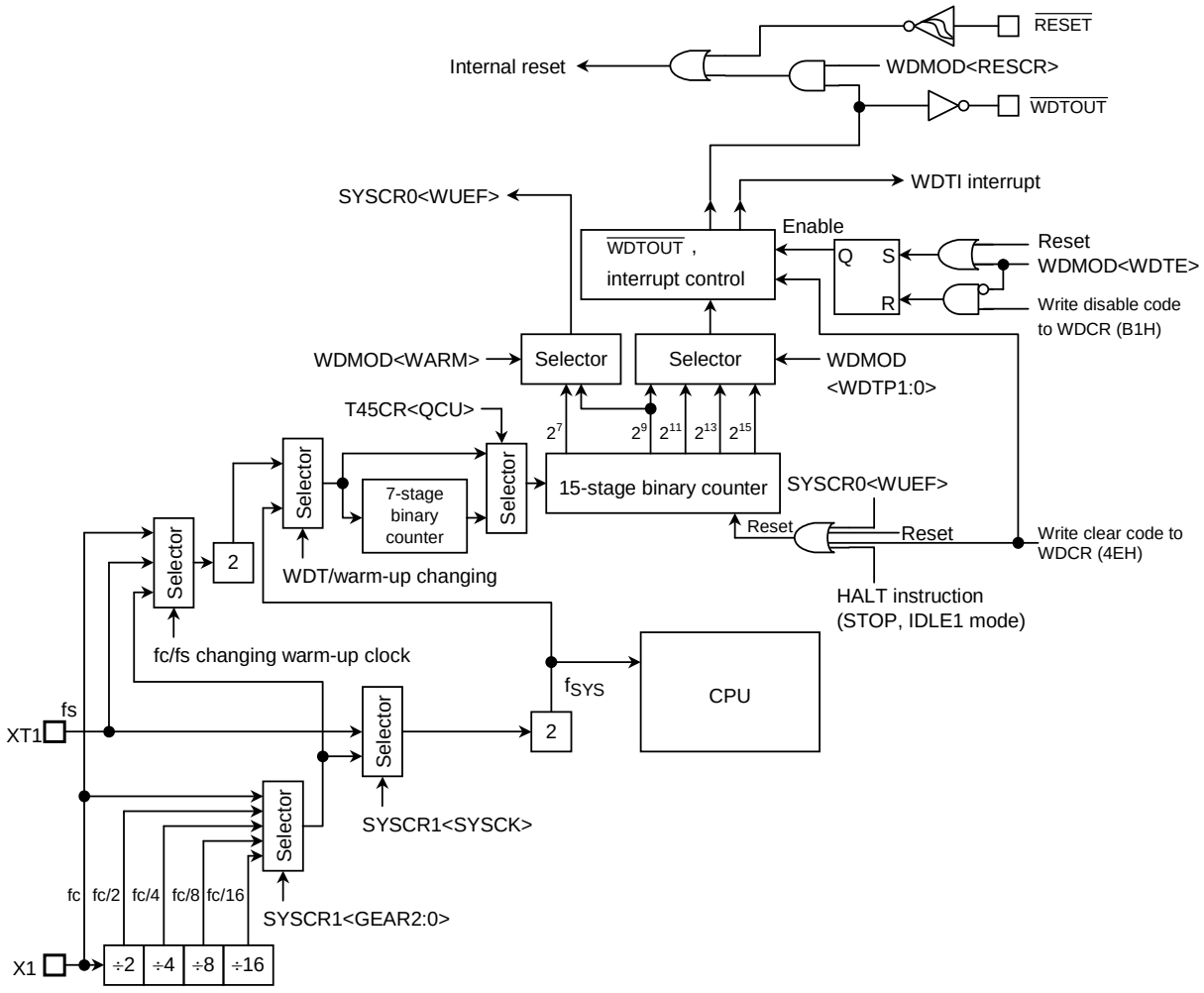


Figure 3.12.1 Block Diagram of Watchdog Timer/Warm-up Timer

The watchdog timer consists of 7-stage and 15-stage binary counters which use System clock (f_{SYS}) as the input clock. Using T45CR<QCU>, the 7-stage binary counter is set to be used or not. The 15-stage binary counter has $f_{SYS}/2^{15}$, $f_{SYS}/2^{17}$, $f_{SYS}/2^{19}$ and $f_{SYS}/2^{21}$ outputs. Selecting one of these outputs with the WDMOD register generates a watchdog interrupt and outputs watchdog timer out when an overflow occurs.

For the warm-up counter, 2^7 and 2^9 outputs of 15-stage binary counter can be selected using WDMOD<WARM> register. When a stable-external oscillator is used, shorter warm-up time is available using T45CR<QCU> register. When <QCU> = 1, counting value of 2^7 is selected.

When the watchdog timer is in operation, this shorter warm-up time function cannot be selected. This function can be available by setting <QCU> = 0.

Since the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs "0" for a watchdog timer overflow, the peripheral devices can be reset. The watchdog timer out pin is set to "1" after disabling WDT and clearing the watchdog timer (by writing a clear code 4EH in the WDCR register).

```
Example: LDW  (WDMOD), B100H  ; Disable
         LD   (WDCR), 4EH     ; Write clear code
         SET  7, (WDMOD)      ; Enable again
```

In other words, the WDTOUT keeps outputting "0" until the clear code is written.

The watchdog timer out pin can also be connected to the reset pin internally. In this case, the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs 0 for 8 to 20 states (12.8 to 32 μs at fc = 20 MHz) and then resets itself.

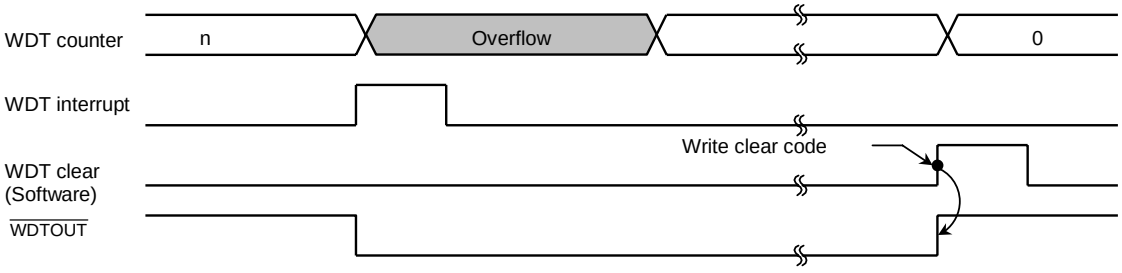


Figure 3.12.2 Normal Mode

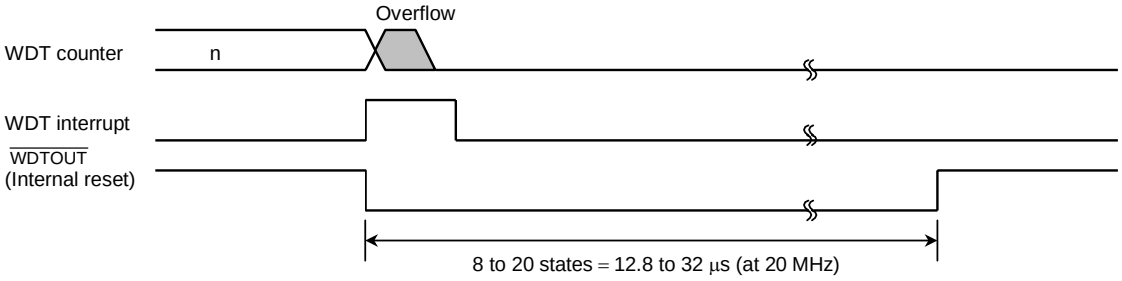


Figure 3.12.3 Reset Mode

3.12.2 Control Registers

Watchdog timer WDT is controlled by two control registers WDMOD and WDCR.

(1) Watchdog timer mode register (WDMOD)

a. 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 Figure 3.12.6.

b. Watchdog timer enable/disable control register <WDTE>

When reset, WDMOD<WDTE> is initialized to "1" to enable the watchdog timer.

To disable, it is necessary to clear 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".

c. Watchdog timer out reset connection<RESCR>

This bit is used to connect the output of the watchdog timer with RESET 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 of the watchdog timer function.

- 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 1: Enable	Select detecting time See Figure 3.12.6		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

DRVE (Explanation by stop mode)

Watchdog timer out control

0	–
1	Connects WDT out to reset

Select the standby mode HALT instruction

00	RUN mode (Only the CPU stops)
01	STOP mode (All circuits stop)
10	IDLE1 mode (Only the oscillator operates)
11	IDLE2 mode (CPU and AD stop)

Select the warm-up time at fc = 20 MHz, fs = 32.768 kHz

System Clock Selection <SYSCK>	Gear Value <GEAR2:0>	Warm-up Time		
		T45CR<QCU> = 0		T45CR<QCU> = 1
		WARM = 0	WARM = 1	
1 (fs)	XXX (Don't care)	0.50 s	2.00 s	3.9 ms
0 (fc)	000 (fc)	0.82 ms	3.28 ms	6.4 μs
	001 (fc/2)	1.64 ms	6.55 ms	12.8 μs
	010 (fc/4)	3.28 ms	13.11 ms	25.6 μs
	011 (fc/8)	6.55 ms	26.21 ms	51.2 μs
	100 (fc/16)	13.11 ms	52.43 ms	102.4 μs

Watchdog timer enable/disable control

0	Disable
1	Enable

Figure 3.12.4 Watchdog Timer Mode Register

		7	6	5	4	3	2	1	0
WDCR (005DH)	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 set

Figure 3.12.5 Watchdog Timer Control Register

at fc = 20 MHz, fs = 32.768 kHz

System Clock Selection <SYSCK>	Gear Value <GEAR2:0>	Watchdog Timer Detecting Time			
		WDMOD<WDTP1:0>			
		00	01	10	11
1 (fs)	XXX (Don't care)	2.00 s	8.00 s	32.00 s	128.0 s
0 (fc)	000 (fc)	3.28 ms	13.11 ms	52.43 ms	209.72 ms
	001 (fc/2)	6.55 ms	26.24 ms	104.86 ms	419.43 ms
	010 (fc/4)	13.11 ms	52.43 ms	209.72 ms	838.86 ms
	011 (fc/8)	26.21 ms	104.86 ms	419.43 ms	1.68 s
	100 (fc/16)	52.43 ms	209.72 ms	838.86 ms	3.36 s

Note: When using as the watchdog timer, write "0" to T45CR<QCU> bit.

Figure 3.12.6 Watchdog Timer Detecting Time

3.12.3 Operation

The watchdog timer generates interrupt INTWD after the detecting time set in the WDMOD<WDTP1:0> and T45CR<QCU> registers and outputs a low level signal to the $\overline{\text{WDTOUT}}$ pin. For normal operation, 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 use of recovery program. By connecting the watchdog timer out pin to peripheral devices' resets, a CPU malfunction can also be acknowledged by other devices.

The watchdog timer restarts operation immediately after reset is released.

The watchdog timer stops its operation in the IDLE1 and STOP modes. In the RUN mode, the watchdog timer is operational. When the bus is released (BUSAK = "L"), WDT continues counting.

The watchdog timer is enabled in IDLE2 mode, but the over flow interrupt is disabled. Disable the watchdog timer before entering IDLE2 mode.

Example: a. Clear the binary counter

WDCR	←	0	1	0	0	1	1	1	0	Write clear code (4EH).

b. Set the watchdog timer detecting time to $2^{17}/f_{SYS}$

WDMOD	←	1	0	1	–	–	–	X	X	

c. 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).

d. 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 standby mode.

e. Set the STOP mode (Warm-up time: 2^{16}/Inputted frequency)

WDMOD	←	–	–	–	1	0	1	X	X	Set the STOP mode.
Executes HALT command.										Execute HALT instruction. Set the standby mode.

4. Electrical Characteristics

"X" used in an expression shows a frequency of clock f_{FPH} selected by SYSCR1<SYSCK>. If a clock gear or a low speed oscillator is selected, a value of "X" is different. The value as an example is calculated at fc, gear = 1/fc (SYSCR1<SYSCK, GEAR2:0> = "0000").

4.1 Maximum Ratings

Parameter	Symbol	Rating	Unit
Power supply voltage	V_{CC}	−0.5 to 6.5	V
Input voltage	V_{IN}	−0.5 to V_{CC} + 0.5	
Output current (Per pin; PA0 to PA5)	I_{OL1}	20	mA
Output current (Per pin; except PA0 to PA5)	I_{OL2}	2	
Output current (PA0 to PA5 total)	ΣI_{OL1}	80	
Output current (Total)	ΣI_{OL}	120	
Output current (Total)	ΣI_{OH}	−80	
Power dissipation (Ta = 85°C)	P_D	600	mW
Soldering temperature (10 s)	T_{SOLDER}	260	°C
Storage temperature	T_{STG}	−65 to 150	
Operating temperature	T_{OPR}	−40 to 85	

Note: The 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 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 maximum rating value will ever be exceeded.

4.2 DC Characteristics (1/2) (V_{SS} = 0 V, Ta = −40 to 85°C)

Parameter		Symbol	Condition		Min	Typ. (Note)	Max	Unit
Power supply voltage ($AV_{CC} = V_{CC}$, $AV_{SS} = V_{SS}$)		V_{CC}	fc = 4 to 20 MHz	fs = 30 to 34 kHz	4.5		5.5	V
			fc = 4 to 12.5 MHz		2.7			
Input low voltage	AD0 to AD15	V_{IL}	$V_{CC} \geq 4.5$ V		−0.3		0.8	V
			$V_{CC} < 4.5$ V				0.6	
	Port 2 to port A (except P87)	V_{IL1}	V_{CC} = 2.7 to 5.5 V				0.3 V_{CC}	
	$\overline{RESET}$, $\overline{NMI}$, INT0	V_{IL2}					0.25 V_{CC}	
	$\overline{EA}$, AM8/$\overline{AM16}$	V_{IL3}					0.3	
	X1	V_{IL4}					0.2 V_{CC}	
Input high voltage	AD0 to AD15	V_{IH}	$V_{CC} \geq 4.5$ V		2.2		V_{CC} + 0.3	
			$V_{CC} < 4.5$ V		2.0			
	Port 2 to port A (except P87)	V_{IH1}	V_{CC} = 2.7 to 5.5 V		0.7 V_{CC}			
	$\overline{RESET}$, $\overline{NMI}$, INT0	V_{IH2}			0.75 V_{CC}			
	$\overline{EA}$, AM8/$\overline{AM16}$	V_{IH3}			V_{CC} − 0.3			
	X1	V_{IH4}			0.8 V_{CC}			

Note: Typical values are for Ta = 25°C and V_{CC} = 5 V unless otherwise noted.

4.2 DC Characteristics (2/2) (V_{SS} = 0 V, Ta = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
Output low voltage	V_{OL}	I_{OL} = 1.6 mA (V_{CC} = 2.7 to 5.5 V)			0.45	V
Output low current (PA0 to PA5)	I_{OLA}	V_{OL} = 1.0 V (V_{CC} = 3 V ± 10%)	7			mA
		V_{OL} = 1.0 V (V_{CC} = 5 V ± 10%)	16			
Output high voltage	V_{OH1}	I_{OH} = −400 μA (V_{CC} = 3 V ± 10%)	2.4			V
	V_{OH2}	I_{OH} = −400 μA (V_{CC} = 5 V ± 10%)	4.2			
Darlington drive current (8 output pins max)	I_{DAR} (Note 2)	V_{EXT} = 1.5 V, R_{EXT} = 1.1 kΩ (V_{CC} = 5 V ± 10% only)	−1.0		−3.5	mA
Input leakage current	I_{LI}	0.0 ≤ V_{IN} ≤ V_{CC}		0.02	±5	μA
Output leakage current	I_{LO}	0.2 ≤ V_{IN} ≤ V_{CC} − 0.2		0.05	±10	
Power down voltage (at STOP, RAM backup)	V_{STOP}	V_{IL2} = 0.2 V_{CC}, V_{IH2} = 0.8 V_{CC}	2.0		6.0	V
$\overline{RESET}$ pull-up resister	R_{RST}	V_{CC} = 5 V ± 10%	50		150	kΩ
		V_{CC} = 3 V ± 10%	80		200	
Pin capacitance	C_{IO}	fc = 1 MHz			10	pF
Schmitt width $\overline{RESET}$, $\overline{NMI}$, INT0	V_{TH}		0.4	1.0		V
Programmable Pull-down resistor	R_{KL}	V_{CC} = 5 V ± 10%	10		80	kΩ
		V_{CC} = 3 V ± 10%	30		150	
Programmable Pull-up resistor	R_{KH}	V_{CC} = 5 V ± 10%	50		150	
		V_{CC} = 3 V ± 10%	100		300	
NORMAL	I_{CC}	V_{CC} = 5 V ± 10%, fc = 20 MHz		21	28	mA
RUN				17	25	
IDLE2				12.5	17	
IDLE1				2.5	4	
NORMAL		V_{CC} = 3 V ± 10%, fc = 12.5 MHz (Typ.: V_{CC} = 3.0 V)		7	10	mA
RUN				5.5	9	
IDLE2				4.5	6	
IDLE1				0.7	1	
SLOW		V_{CC} = 3 V ± 10%, fs = 32.768 kHz (Typ.: V_{CC} = 3.0 V)		20	35	μA
RUN				16	30	
IDLE2				11	25	
IDLE1				4	15	
STOP		Ta ≤ 50°C, V_{CC} = 2.7 to 5.5 V		0.2	10	μA
		Ta ≤ 70°C			20	
		Ta ≤ 85°C			50	

Note 1: Typical values are for Ta = 25°C and V_{CC} = 5 V unless otherwise noted.

Note 2: I_{DAR} is guaranteed for total of up to 8 ports.

4.3 AC Characteristics

(1) $V_{CC} = 5\ V \pm 10\%$

No.	Parameter	Symbol	Variable		16 MHz		20 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Osc. period (= X)	t_{OSC}	50	33333	62.5		50		ns
2	CLK pulse width	t_{CLK}	2x – 40		85		60		ns
3	A0 to A23 valid → CLK hold	t_{AK}	0.5x – 20		11		5		ns
4	CLK valid→A0 to A23 hold	t_{KA}	1.5x – 70		24		5		ns
5	A0 to A15 valid → ALE fall	t_{AL}	0.5x – 15		16		10		ns
6	ALE fall → A0 to A15 hold	t_{LA}	0.5x – 20		11		5		ns
7	ALE high pulse width	t_{LL}	x – 40		23		10		ns
8	ALE fall → $\overline{RD}$ / $\overline{WR}$ fall	t_{LC}	0.5x – 25		6		0		ns
9	$\overline{RD}$ / $\overline{WR}$ rise → ALE rise	t_{CL}	0.5x – 20		11		5		ns
10	A0 to A15 valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{ACL}	x – 25		38		25		ns
11	A0 to A23 valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{ACH}	1.5x – 50		44		25		ns
12	$\overline{RD}$ / $\overline{WR}$ rise → A0 to A23 hold	t_{CA}	0.5x – 25		6		0		ns
13	A0 to A15 valid → D0 to D15 input	t_{ADL}		3.0x – 55		133		95	ns
14	A0 to A23 valid → D0 to D15 input	t_{ADH}		3.5x – 65		154		110	ns
15	$\overline{RD}$ fall → D0 to D15 input	t_{RD}		2.0x – 60		65		40	ns
16	$\overline{RD}$ low pulse width	t_{RR}	2.0x – 40		85		60		ns
17	$\overline{RD}$ rise → D0 to D15 hold	t_{HR}	0		0		0		ns
18	$\overline{RD}$ rise → A0 to A15 output	t_{RAE}	x – 15		48		35		ns
19	$\overline{WR}$ low pulse width	t_{WW}	2.0x – 40		85		60		ns
20	D0 to D15 valid → $\overline{WR}$ rise	t_{DW}	2.0x – 55		70		45		ns
21	$\overline{WR}$ rise → D0 to D15 hold	t_{WD}	0.5x – 15		16		10		ns
22	A0 to A23 valid → $\overline{WAIT}$ input $\left(\begin{smallmatrix}(1+N)\ WAIT\\ mode\end{smallmatrix}\right)$	t_{AWH}		3.5x – 90		129		85	ns
23	A0 to A15 valid → $\overline{WAIT}$ input $\left(\begin{smallmatrix}(1+N)\ WAIT\\ mode\end{smallmatrix}\right)$	t_{AWL}		3.0x – 80		108		70	ns
24	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold $\left(\begin{smallmatrix}(1+N)\ WAIT\\ mode\end{smallmatrix}\right)$	t_{CW}	2.0x + 0		125		100		ns
25	A0 to A23 valid → Port input	t_{APH}		2.5x – 120		36		5	ns
26	A0 to A23 valid → Port hold	t_{APH2}	2.5x + 50		206		175		ns
27	$\overline{WR}$ rise → Port valid	t_{CP}		200		200		200	ns
28	A0 to A23 valid → $\overline{RAS}$ fall	t_{ASRH}	1.0x – 40		23		10		ns
29	A0 to A15 valid → $\overline{RAS}$ fall	t_{ASRL}	0.5x – 15		16		10		ns
30	$\overline{RAS}$ fall → D0 to D15 input	t_{RAC}		2.5x – 70		86		55	ns
31	$\overline{RAS}$ fall → A0 to A15 hold	t_{RAH}	0.5x – 15		16		10		ns
32	$\overline{RAS}$ low pulse width	t_{RAS}	2.0x – 40		85		60		ns
33	$\overline{RAS}$ high pulse width	t_{RP}	2.0x – 40		85		60		ns
34	$\overline{CAS}$ fall → $\overline{RAS}$ rise	t_{RSH}	1.0x – 40		23		10		ns
35	$\overline{RAS}$ rise → $\overline{CAS}$ rise	t_{RSC}	0.5x – 25		6		0		ns
36	$\overline{RAS}$ fall → $\overline{CAS}$ fall	t_{RCD}	1.0x – 40		23		10		ns
37	$\overline{CAS}$ fall → D0 to D15 input	t_{CAC}		1.5x – 65		29		10	ns
38	$\overline{CAS}$ low pulse width	t_{CAS}	1.5x – 30		64		40		ns

AC measuring conditions

- Output level: High 2.2 V/Low 0.8 V, CL = 50 pF
  (However CL = 100 pF for AD0 to AD15, A0 to A23, ALE, RD, WR, HWR, R/W, CLK, RAS, CAS0 to CAS2)
- Input level: High 2.4 V/Low 0.45 V (AD0 to AD15)
  High 0.8 × V_{CC}/Low 0.2 × V_{CC} (Except for AD0 to AD15)

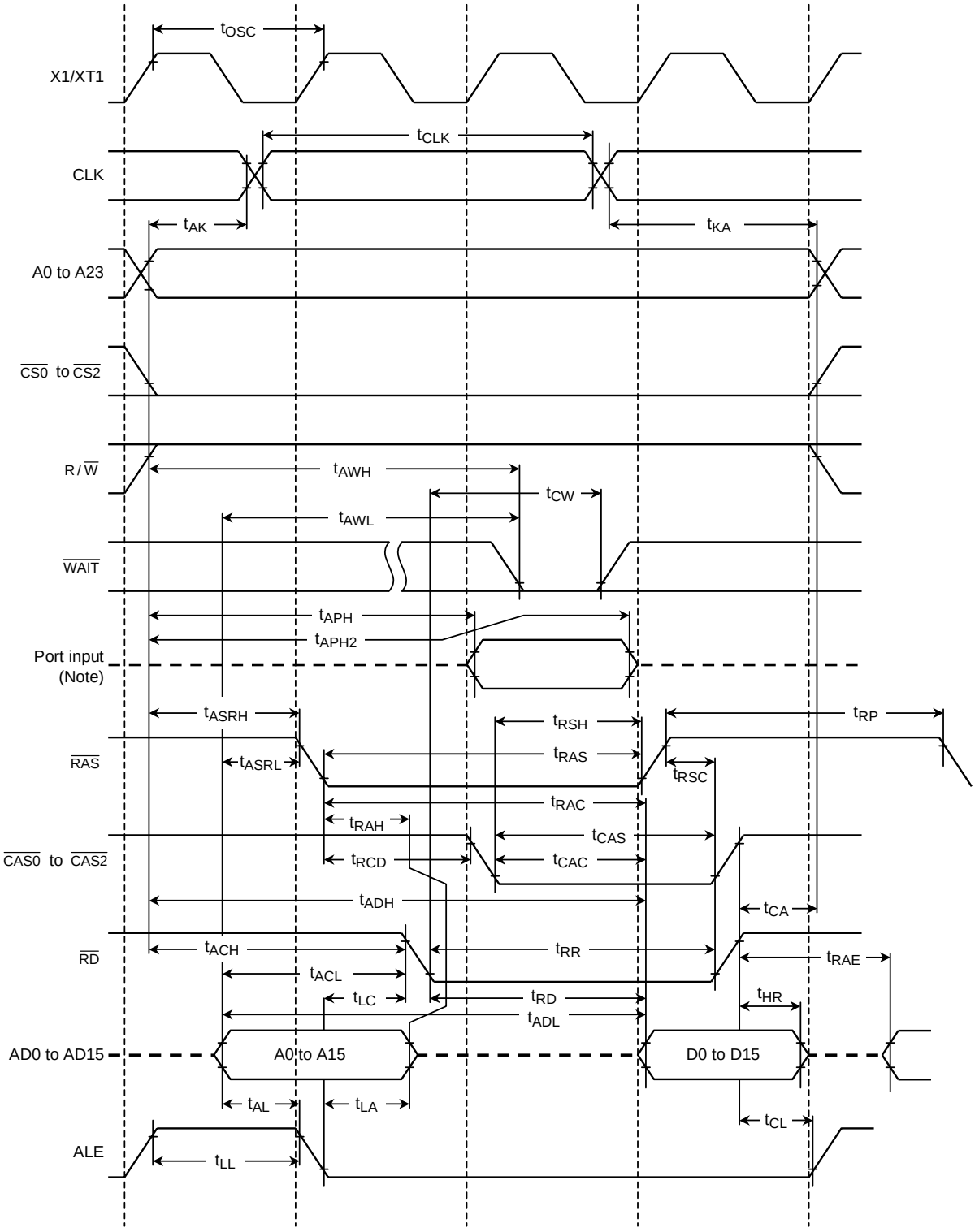
(2) V_{CC} = 3 V ± 10%

No.	Parameter	Symbol	Variable		12.5 MHz		Unit
			Min	Max	Min	Max	
1	Osc. period (= x)	t_{OSC}	80	33333	80		ns
2	CLK pulse width	t_{CLK}	2x – 40		120		ns
3	A0 to A23 valid → CLK hold	t_{AK}	0.5x – 30		10		ns
4	CLK valid → A0 to A23 hold	t_{KA}	1.5x – 80		40		ns
5	A0 to A15 valid → ALE fall	t_{AL}	0.5x – 35		5		ns
6	ALE fall → A0 to A15 hold	t_{LA}	0.5x – 35		5		ns
7	ALE high pulse width	t_{LL}	x – 60		20		ns
8	ALE fall → $\overline{RD}$ / $\overline{WR}$ fall	t_{LC}	0.5x – 35		5		ns
9	$\overline{RD}$ / $\overline{WR}$ rise → ALE rise	t_{CL}	0.5x – 40		0		ns
10	A0 to A15 valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{ACL}	x – 50		30		ns
11	A0 to A23 valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{ACH}	1.5x – 50		70		ns
12	$\overline{RD}$ / $\overline{WR}$ rise → A0 to A23 hold	t_{CA}	0.5x – 40		0		ns
13	A0 to A15 valid → D0 to D15 input	t_{ADL}		3.0x – 110		130	ns
14	A0 to A23 valid → D0 to D15 input	t_{ADH}		3.5x – 125		155	ns
15	$\overline{RD}$ fall → D0 to D15 input	t_{RD}		2.0x – 115		45	ns
16	$\overline{RD}$ low pulse width	t_{RR}	2.0x – 40		120		ns
17	$\overline{RD}$ rise → D0 to D15 hold	t_{HR}	0		0		ns
18	$\overline{RD}$ rise → A0 to A15 output	t_{RAE}	x – 25		55		ns
19	$\overline{WR}$ low pulse width	t_{WW}	2.0x – 40		120		ns
20	D0 to D15 valid → $\overline{WR}$ rise	t_{DW}	2.0x – 120		40		ns
21	$\overline{WR}$ rise → D0 to D15 hold	t_{WD}	0.5x – 40		0		ns
22	A0 to A23 valid → $\overline{WAIT}$ input $\left(\begin{smallmatrix}(1+N)\ WAIT\\ mode\end{smallmatrix}\right)$	t_{AWH}		3.5x – 130		150	ns
23	A0 to A15 valid → $\overline{WAIT}$ input $\left(\begin{smallmatrix}(1+N)\ WAIT\\ mode\end{smallmatrix}\right)$	t_{AWL}		3.0x – 100		140	ns
24	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold $\left(\begin{smallmatrix}(1+N)\ WAIT\\ mode\end{smallmatrix}\right)$	t_{CW}	2.0x + 0		160		ns
25	A0 to A23 valid → Port input	t_{APH}		2.5x – 195		5	ns
26	A0 to A23 valid → Port hold	t_{APH2}	2.5x + 50		250		ns
27	$\overline{WR}$ rise → Port valid	t_{CP}		200		200	ns
28	A0 to A23 valid → $\overline{RAS}$ fall	t_{ASRH}	1.0x – 60		20		ns
29	A0 to A15 valid → $\overline{RAS}$ fall	t_{ASRL}	0.5x – 40		0		ns
30	$\overline{RAS}$ fall → D0 to D15 input	t_{RAC}		2.5x – 90		110	ns
31	$\overline{RAS}$ fall → A0 to A15 hold	t_{RAH}	0.5x – 25		15		ns
32	$\overline{RAS}$ low pulse width	t_{RAS}	2.0x – 40		120		ns
33	$\overline{RAS}$ high pulse width	t_{RP}	2.0x – 40		120		ns
34	$\overline{CAS}$ fall → $\overline{RAS}$ rise	t_{RSH}	1.0x – 55		25		ns
35	$\overline{RAS}$ rise → $\overline{CAS}$ rise	t_{RSC}	0.5x – 25		15		ns
36	$\overline{RAS}$ fall → $\overline{CAS}$ fall	t_{RCD}	1.0x – 40		40		ns
37	$\overline{CAS}$ fall → D0 to D15 input	t_{CAC}		1.5x – 120		0	ns
38	$\overline{CAS}$ low pulse width	t_{CAS}	1.5x – 40		80		ns

AC measuring conditions

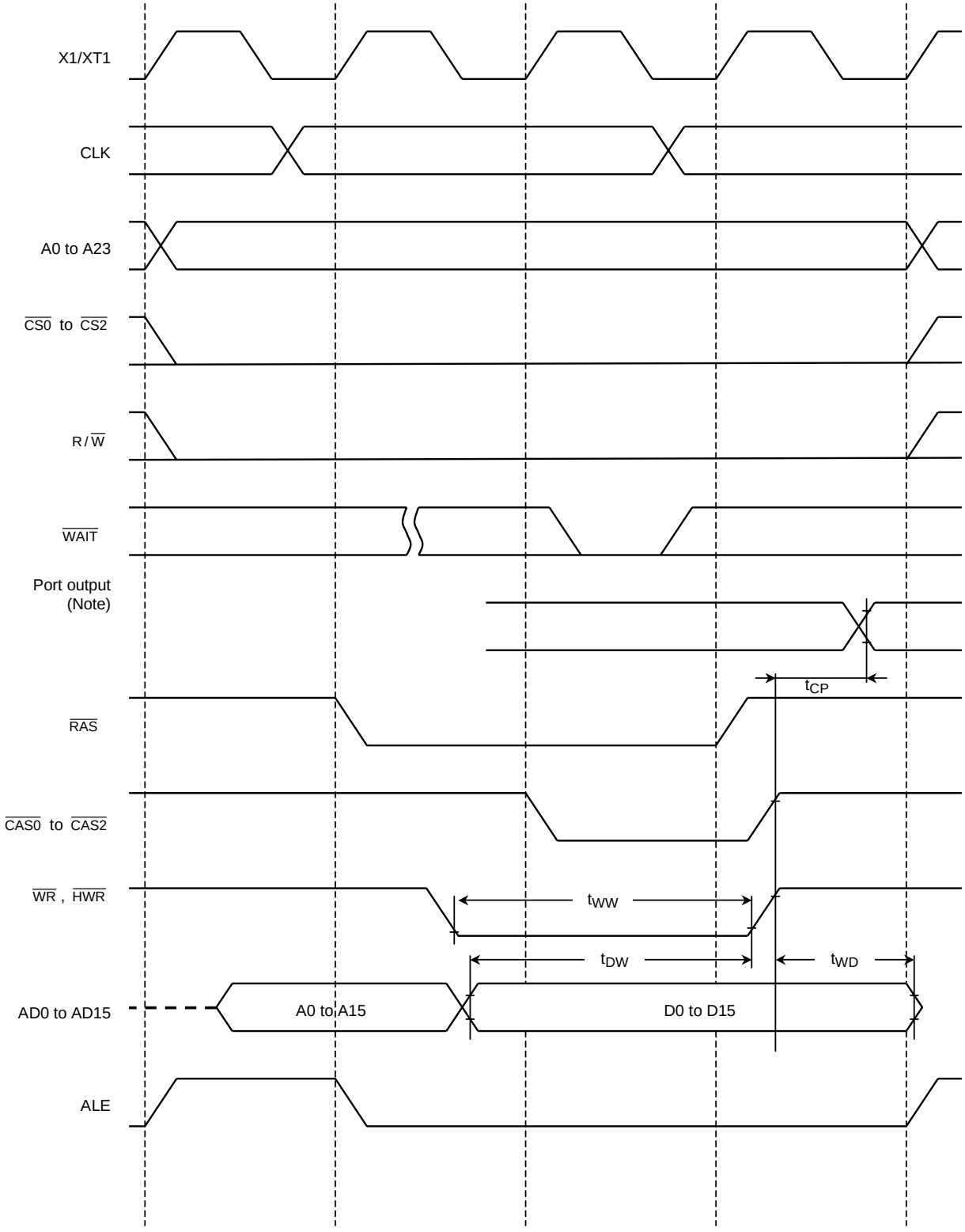
- Output level: High 0.7 × V_{CC}/Low 0.3 × V_{CC}, CL = 50 pF
- Input level: High 0.9 × V_{CC}/Low 0.1 × V_{CC}

(1) Read cycle



Note: Since the CPU accesses the internal area to read data from a port, the control signals of external pins such as $\overline{RD}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

(2) Write cycle



Note: Since the CPU accesses the internal area to write data to a port, the control signals of external pins such as $\overline{WR}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

4.4 AD Conversion Characteristics (V_{SS} = 0 V, AV_{CC} = V_{CC}, AV_{SS} = V_{SS}, Ta = –40 to 85°C)

Parameter	Symbol	Power Supply	Min	Typ.	Max	Unit
Analog reference voltage (+)	V_{REFH}	V_{CC} = 5 V ± 10%	V_{CC} – 1.5	V_{CC}	V_{CC}	V
		V_{CC} = 3 V ± 10%	V_{CC} – 0.2	V_{CC}	V_{CC}	
Analog reference voltage (–)	V_{REFL}	V_{CC} = 5 V ± 10%	V_{SS}	V_{SS}	V_{SS} + 0.2	
		V_{CC} = 3 V ± 10%	V_{SS}	V_{SS}	V_{SS} + 0.2	
Analog input voltage range	V_{AIN}		V_{REFL}		V_{REFH}	
Analog current for analog reference voltage <VREFON> = 1	I_{REF} (V_{REFL} = 0 V)	V_{CC} = 5 V ± 10%		0.5	1.5	mA
		V_{CC} = 3 V ± 10%		0.3	0.9	
<VREFON> = 0		V_{CC} = 2.7 to 5.5V		0.02	5.0	μA
Error	–	V_{CC} = 5 V ± 10%		±1.0	±3.0	LSB
		V_{CC} = 3 V ± 10%		±1.0	±3.0	

Note 1: 1LSB = (VREFH – VREFL)/2^{10} [V]

Note 2: Minimum operation frequency

The operation of this AD converter is guaranteed only when fc (High-frequency oscillator) is used. (It is not guaranteed when fs is used.) Additionally, it is guaranteed with $f_{FPH} \geq 4$ MHz.

Note 3: The value ICC includes the current with flows through AVCC pin.

Note 4: The operation of this AD converter is guaranteed at 5 V ± 10%.

4.5 Serial Channel Timing

(1) SCLK input mode

Parameter	Symbol	Variable		32.768 kHz (Note)		12.5 MHz		20 MHz	
		Min	Max	Min	Max	Min	Max	Min	Max
SCLK cycle	t_{SCY}	16X		488 μs		1.28 μs		0.8 μs	
Output data → Rising edge of SCLK	t_{OSS}	t_{SCY}/2 – 5X – 50		91.5 μs		190 ns		100 ns	
SCLK rising edge → Output data hold	t_{OHS}	5X – 100		152 μs		300 ns		150 ns	
SCLK rising edge → Input data hold	t_{HSR}	0		0		0		0	
SCLK rising edge → Effective data input	t_{SRD}		t_{SCY} – 5X – 100		336 μs		780 ns		450 ns

(2) SCLK output mode

Parameter	Symbol	Variable		32.768 kHz (Note)		12.5 MHz		20 MHz	
		Min	Max	Min	Max	Min	Max	Min	Max
SCLK cycle (Programmable)	t_{SCY}	16X	8192X	488 μs	250 ms	1.28 μs	655.36 μs	0.8 μs	409.6 μs
Output data → SCLK rising edge	t_{OSS}	t_{SCY} – 2X – 150		427 μs		970 ns		550 ns	
SCLK rising edge → Output data hold	t_{OHS}	2X – 80		60 μs		80 ns		20 ns	
SCLK rising edge → Input data hold	t_{HSR}	0		0		0		0	
SCLK rising edge → Effective data input	t_{SRD}		t_{SCY} – 2X – 150		428 μs		970 ns		550 ns

(3) SCLK input mode (UART mode)

Parameter	Symbol	Variable		32.768 kHz (Note)		12.5 MHz		20 MHz	
		Min	Max	Min	Max	Min	Max	Min	Max
SCLK cycle	t_{SCY}	4X + 20		122 μs		340 ns		220 ns	
Low level SCLK pulse width	t_{SCYL}	2X + 5		6 μs		165 ns		105 ns	
High level SCLK pulse width	t_{SCYH}	2X + 5		6 μs		165 ns		105 ns	

Note: fs is used as system clock (f_{SYS}) or fs is used as input clock to prescaler.

4.6 Timer/Counter Input Clock (TI0, TI4, TI5, TI6 and TI7)

Parameter	Symbol	Variable		12.5 MHz		20 MHz		
		Min	Max	Min	Max	Min	Max	
Clock cycle	t_{VCK}	8X + 100		740		500		ns
Low level clock pulse width	t_{VCKL}	4X + 40		360		240		ns
High level clock pulse width	t_{VCKH}	4X + 40		360		240		ns

4.7 Interrupt and Capture

(1) $\overline{NMI}$ and INT0 interrupts

Parameter	Symbol	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, INT0 low level pulse width	t_{INTAL}	4X		320		200		ns
$\overline{NMI}$, INT0 high level pulse width	t_{INTAH}	4X		320		200		ns

(2) INT4 to INT7

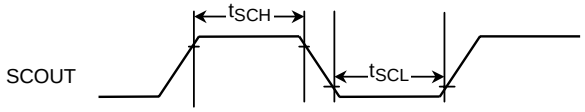
Parameter	Symbol	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
INT4 to INT7 low level pulse width	t_{INTBL}	4X + 100		420		300		ns
INT4 to INT7 high level pulse width	t_{INTBH}	4X + 100		420		300		ns

4.8 SCOUT Pin AC Characteristics

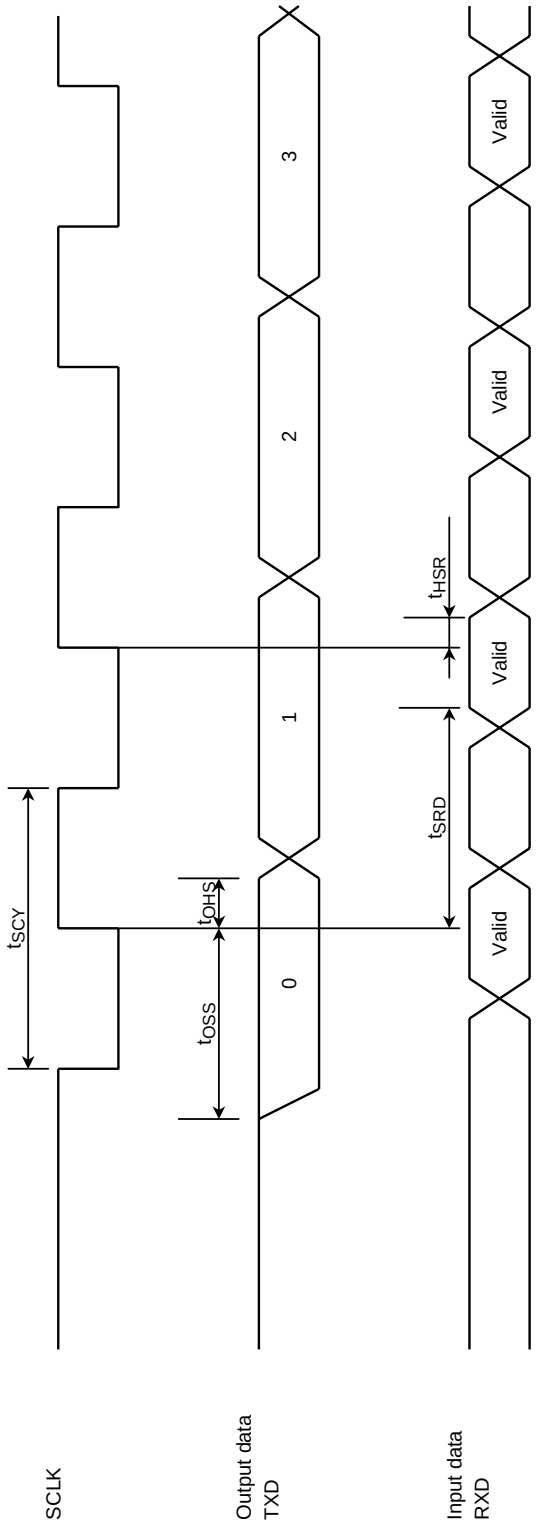
Parameter	Symbol	Variable		12.5 MHz		20 MHz	
		Min	Max	Min	Max	Min	Max
High-level pulse width	t_{SCH}						
V_{CC} = 5V ± 10%		0.5X – 10		30		15	
V_{CC} = 3V ± 10%		0.5X – 20		20		–	–
Low-level pulse width	t_{SCL}						
V_{CC} = 5V ± 10%		0.5X – 10		30		15	
V_{CC} = 3V ± 10%		0.5X – 20		20		–	–

Measurement condition

- Output level: High 2.2 V/Low 0.8 V, CL = 10 pF

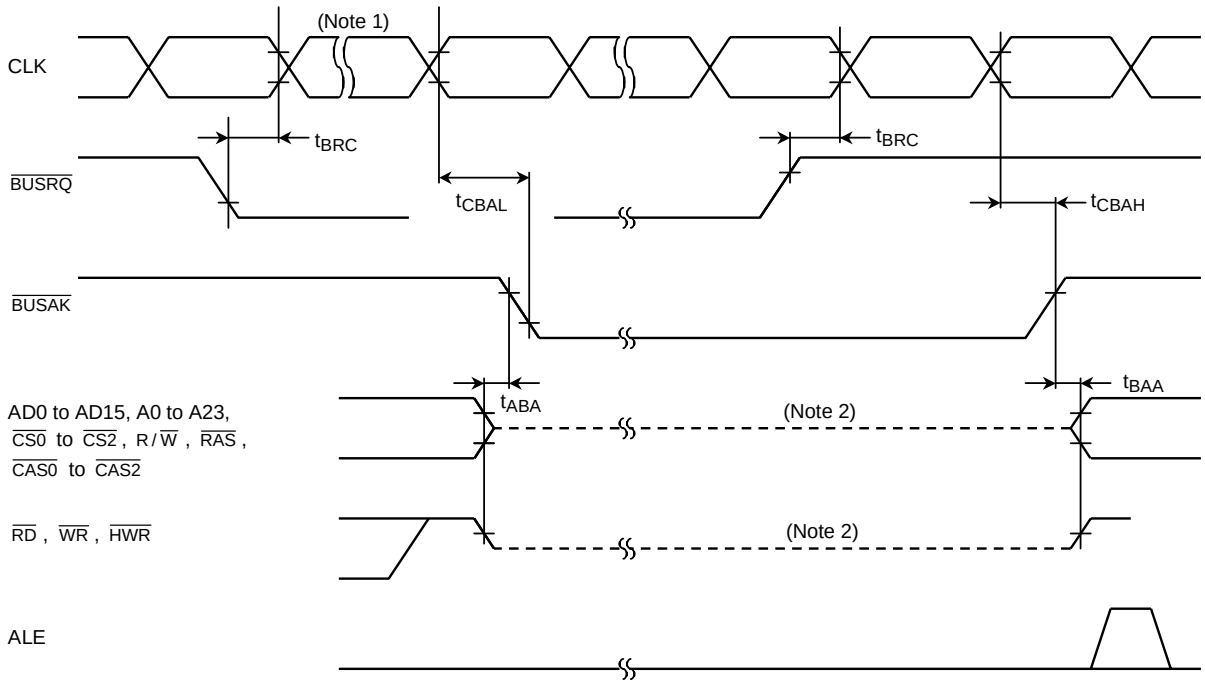


4.9 Timing Chart for I/O Interface Mode



Note: SCLK is reversed in SCLK input falling mode.

4.10 Timing Chart for Bus Request ($\overline{BUSRQ}$)/Bus Acknowledge ($\overline{BUSAK}$)



Parameter	Symbol	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{BUSRQ}$ setup time to CLK	t_{BRC}	120		120		120		ns
CLK→ $\overline{BUSAK}$ falling edge	t_{CBAL}		1.5X + 120		240		195	ns
CLK→ $\overline{BUSAK}$ rising edge	t_{CBAH}		0.5x + 40		80		65	ns
Output buffer is off to $\overline{BUSAK}$ ↘	t_{ABA}	0	80	0	80	0	80	ns
$\overline{BUSAK}$ ↗ to output buffer is on.	t_{BAA}	0	80	0	80	0	80	ns

Note 1: The bus will be released after the $\overline{WAIT}$ request is inactive, when the $\overline{BUSRQ}$ is set to “0” during “wait” cycle.

Note 2: This line only shows the output buffer is off state.
It doesn’t indicate the signal level is fixed.
Just after the bus is released, the signal level which is set before the bus is released is kept dynamically by the external capacitance. Therefore, to fix the signal level by an external resistor during bus releasing, designing is executed carefully because the level-fix will be delayed.
The internal programmable pull-up/pull-down resistor is switched active/non-active by an internal signal.

5. Table of Special Function Registers (SFRs)

The special function registers (SFRs) include the I/O ports and peripheral control registers allocated to the 128-byte addresses from 000000H to 00007FH.

(1) I/O port

(2) I/O port control

(3) Timer control

(4) Watchdog timer control

(5) Serial channel control

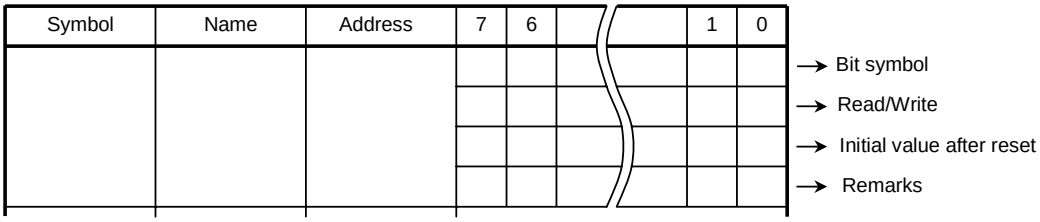
(6) AD converter control

(7) Interrupt control

(8) Chip select/wait control

(9) Clock control

Configuration of the table



Note: "Prohibit RMW" in table means that you cannot use RMW instructions to these registers.

Example: In case of setting only the bit 0 of register P0CR, you mustn't use "Set 0, (0002H)"

Table 5.1 I/O Register Address Map

Address	Name	Address	Name	Address	Name	Address	Name
000000H	P0	20H	TRUN	40H	TREG6L	60H	ADREG04L
1H	P1	21H		41H	TREG6H	61H	ADREG04H
2H	P0CR	22H	TREG0	42H	TREG7L	62H	ADREG15L
3H		23H	TREG1	43H	TREG7H	63H	ADREG15H
4H	P1CR	24H	TMOD	44H	CAP3L	64H	ADREG26L
5H	P1FC	25H	TFFCR	45H	CAP3H	65H	ADREG26H
6H	P2	26H	TREG2	46H	CAP4L	66H	ADREG37L
7H	P3	27H	TREG3	47H	CAP4H	67H	ADREG37H
8H	P2CR	28H	P0MOD	48H	T5MOD	68H	B0CS
9H	P2FC	29H	P1MOD	49H	T5FFCR	69H	B1CS
AH	P3CR	2AH	PFFCR	4AH		6AH	B2CS
BH	P3FC	2BH	BRADD2	4BH	BRADD4	6BH	BRADD0
CH	P4	2CH	SC2BUF	4CH	SC4BUF	6CH	BRADD1
DH	P5	2DH	SC2CR	4DH	SC4CR	6DH	CKOCR
EH	P4CR	2EH	SC2MOD	4EH	SC4MOD	6EH	SYSCR0
FH		2FH	BR2CR	4FH	BR4CR	6FH	SYSCR1
10H	P4FC	30H	TREG4L	50H	SC0BUF	70H	INTE0AD
11H		31H	TREG4H	51H	SC0CR	71H	INTE45
12H	P6	32H	TREG5L	52H	SC0MOD	72H	INTE67
13H	P7	33H	TREG5H	53H	BR0CR	73H	INTET10
14H	P6CR	34H	CAP1L	54H	SC1BUF	74H	INTEPW10
15H	P7CR	35H	CAP1H	55H	SC1CR	75H	INTET54
16H	P6FC	36H	CAP2L	56H	SC1MOD	76H	INTET76
17H	P7FC	37H	CAP2H	57H	BR1CR	77H	INTES0
18H	P8	38H	T4MOD	58H	ODE	78H	INTES1
19H	P9	39H	T4FFCR	59H	INTES2	79H	
1AH	P8CR	3AH	T45CR	5AH	INTES3	7AH	
1BH	P9CR	3BH	BRADD3	5BH	INTES4	7BH	IIMC
1CH	P8FC	3CH	SC3BUF	5CH	WDMOD	7CH	DMA0V
1DH	P9FC	3DH	SC3CR	5DH	WDCR	7DH	DMA1V
1EH	PA	3EH	SC3MOD	5EH	ADMOD1	7EH	DMA2V
1FH	PACR	3FH	BR3CR	5FH	ADMOD2	7FH	DMA3V

Note: Do not access the addresses without allocated register names.

(1) I/O port

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	Port 0	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Undefined							
			Input mode							
P1	Port 1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			0	0	0	0	0	0	0	0
			Input mode							
P2	Port 2	06H	P27	P26	P25	P24	P23	P22	P21	P20
			*R/W (Note 3)							
			0	0	0	0	0	0	0	0
			Input mode							
P3	Port 3	07H	P37	P36	P35	P34	P33	P32	P31	P30 (Note 1)
			*R/W (Note 3)							
			1	1	1	1	1	1	1	1
			Input mode						Output mode	
P4	Port 4	0CH						P42	P41	P40
								*R/W (Note 3)		
								0	1	1
								Input mode		
P5	Port 5	0DH	P57	P56	P55	P54	P53	P52	P51	P50
			R							
			Input mode							
P6	Port 6	12H	P67	P66	P65	P64	P63	P62	P61	P60
			*R/W (Note 3)							
			1	1	1	1	1	1	1	1
			Input mode							
P7	Port 7	13H					P73	P72	P71	P70
							*R/W (Note 3)			
							1	1	1	1
							Input mode			
P8	Port 8	18H	P87	P86	P85	P84	P83	P82	P81	P80
			*R/W (Note 3)							
			1	1	1	1	1	1	1	1
			Input mode							
P9	Port 9 (Note 2)	19H	P97	P96	P95	P94	P93	P92	P91	P90
			R/W	R/W	*R/W (Note 3)					
			1	1	1	1	1	1	1	1
			Output mode	Output mode	Input mode					
PA	Port A	1EH	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
			R/W							
			1	1	1	1	1	1	1	1
			Input mode							

Note 1: When P30 pin is defined as $\overline{RD}$ signal output mode (P30F = 1), clearing the output latch register P30 to "0" outputs the $\overline{RD}$ strobe from P30 pin for PSRAM, even when the internal address is accessed. If the output latch register P30 remains "1", the $\overline{RD}$ strobe is output only when the external address is accessed.

Note 2: Port 96, 97 is also used as XT1, XT2. Therefore these pins are open drain output type.

Read/Write

R/W: Either read or write is possible

R: Only read is possible

W: Only write is possible

Prohibit RMW: Prohibit read-modify-write. (Prohibit RES/SET/TSET/CHG/STCF /ANDCF/ORCF/XORCF instruction.)

Note 3: *R/W: Read-modify-write is prohibited when controlling the pull-up/pull-down resistors.

(2) I/O port control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0CR	Port 0 control	02H (Prohibit RMW)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0: In 1: Out (When external access, set as AD7 to AD0 and cleared to "0".)							
P1CR	Port 1 control	04H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			<<Refer to the "P1FC">>							
P1FC	Port 1 function	05H (Prohibit RMW)	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
			W							
			0	0	0	0	0	0	0	0
			P1FC/P1CR = 00: Input 01: Output 10: AD15 to AD8 11: A15 to A8							
P2CR	Port 2 control	08H (Prohibit RMW)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0	0	0	0	0	0	0	0
			<<Refer to the "P2FC">>							
P2FC	Port 2 function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			0	0	0	0	0	0	0	0
			P2FC/P2CR = 00: Input 01: Output 10: A7 to A0 11: A23 to A16							
P3CR	Port 3 control	0AH (Prohibit RMW)	P37C	P36C	P35C	P34C	P33C	P32C		
			W							
			0	0	0	0	0	0		
			0: Input 1: Output							
P3FC	Port 3 function	0BH (Prohibit RMW)	P37F	P36F	P35F	P34F		P32F	P31F	P30F
			W							
			0	0	0	0		0	0	0
			0: Port 1: $\overline{RAS}$	0: Port 1: R/$\overline{W}$	0: Port 1: $\overline{BUSAK}$	0: Port 1: $\overline{BUSRQ}$		0: Port 1: $\overline{HWR}$	0: Port 1: $\overline{WR}$	0: Port 1: $\overline{RD}$
P4CR	Port 4 control	0EH (Prohibit RMW)						P42C	P41C	P40C
								W		
								0	0	0
								0: Input 1: Output		
P4FC	Port 4 function	10H (Prohibit RMW)						P42F	P41F	P40F
								W		
								0	0	0
								0: Port 1: $\overline{CS}$ / $\overline{CAS}$		

(2) I/O port control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6CR	Port 6 control	14H (Prohibit RMW)	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P7CR	Port 7 control	15H (Prohibit RMW)					P73C	P72C	P71C	P70C
							W			
							0	0	0	0
							0: Input 1: Output			
P6FC	Port 6 function	16H (Prohibit RMW)		P66F	P65F		P63F	P62F		P60F
				W	W		W	W		W
				0	0		0	0		0
				0: Port 1: TXD4	0: Port 1: SCLK3		0: Port 1: TXD3	0: Port 1: SCLK2		0: Port 1: TXD2
P7FC	Port 7 function	17H (Prohibit RMW)					P73F	P72F	P71F	
							W			
							0	0	0	
							0: Port 1: TO3	0: Port 1: TO2	0: Port 1: TO1	
P8CR	Port 8 control	1AH (Prohibit RMW)	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P9CR	Port 9 control	1BH (Prohibit RMW)	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
			W	W	W					
			1	1	0	0	0	0	0	0
			0: Input 1: Output							
P8FC	Port 8 function	1CH (Prohibit RMW)		P86F			P83F	P82F		
				W			W	W		
				0			0	0		
				0: Port 1: TO6			0: Port 1: TO5	0: Port 1: TO4		
P9FC	Port 9 function	1DH (Prohibit RMW)			P95F		P93F	P92F		P90F
					W		W	W		W
					0		0	0		0
					0: Port 1: SCLK1		0: Port 1: TXD1	0: Port 1: SCLK0		0: Port 1: TXD0
PACR	Port A control	1FH (Prohibit RMW)	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							

(3) Timer control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN	Timer control	20H	PRRUN		T5RUN	T4RUN	P1RUN	P0RUN	T1RUN	T0RUN
			R/W		R/W					
			0		0	0	0	0	0	0
			Prescaler and timer run/stop control 0: Stop and clear 1: Run (Count up)							
TREG0	8-bit timer register 0	22H (Prohibit RMW)	–							
			W							
			Undefined							
TREG1	8-bit timer register 1	23H (Prohibit RMW)	–							
			W							
			Undefined							
TMOD	8-bit timer source CLK & mode	24H (Prohibit RMW)	T10M1	T10M0	PWMM1	PWMM0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			W							
			0	0	0	0	0	0	0	0
			00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		00: – 01: 2^6-1 10: 2^7-1 11: 2^8-1	PWM	00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TI0 Input 01: ϕT1 10: ϕT4 11: ϕT16	
TFFCR	8-bit timer flip-flop control	25H				DBEN	TFF1C1	TFF1C0	TFF1IE	TFF1IS
						R/W	W		R/W	
						0	1	1	0	0
						1: Double buffer enable	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		1: TFF1 invert enable	0: Inverted by timer 0
TREG2	PWM timer register 2	26H	–							
			(R)/W (Can read double buffer values.)							
			Undefined							
TREG3	PWM timer register 3	27H	–							
			(R)/W (Can read double buffer values.)							
			Undefined							
P0MOD	PWM0 mode	28H (Prohibit RMW)	FF2RD	DB2EN	PWM0INT	PWM0M	T2CLK1	T2CLK0	PWM0S1	PWM0S0
			R	W						
			–	0	0	0	0	0	0	0
			TFF2 output value	1: Double buffer enable	0:Overflow interrupt 1:Compare/ match interrupt	0: PWM mode 1: Timer mode	00: ϕP1 01: ϕP4 10: ϕP16 11: Don't care		00: 2^6-1 01: 2^7-1 10: 2^8-1 11: Don't care	
P1MOD	PWM1 mode	29H (Prohibit RMW)	FF3RD	DB3EN	PWM1INT	PWM1M	T3CLK1	T3CLK0	PWM1S1	PWM1S0
			R	W						
			–	0	0	0	0	0	0	0
			TFF3 output value	1: Double buffer enable	0: Overflow interrupt 1: Compare/ match interrupt	0: PWM mode 1: Timer mode	00: ϕP1 01: ϕP4 10: ϕP16 11: Don't care		00: 2^6-1 01: 2^7-1 10: 2^8-1 11: Don't care	

(3) Timer control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PFFCR	PWM flip-flop control	2AH	FF3C1	FF3C0	FF3TRG1	FF3TRG0	FF2C1	FF2C0	FF2TRG1	FF2TRG0
			W		R/W		W		R/W	
			1	1	0	0	1	1	0	0
			00: Don't care 01: Set TFF3 10: Clear TFF3 11: Don't care		00: Prohibit TFF3 invert 01: Invert if matched 10: Set if matched; clear if overflow 11: Clear if matched; set if overflow		00: Don't care 01: Set TFF2 10: Clear TFF2 11: Don't care		00: Prohibit TFF2 invert 01: Invert if matched 10: Set if matched; clear if overflow 11: Clear if matched; set if overflow	
TREG4L	16-bit timer register 4 low	30H (Prohibit RMW)	–							
			W							
			Undefined							
TREG4H	16-bit timer register 4 high	31H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5L	16-bit timer register 5 low	32H (Prohibit RMW)	–							
			W							
			Undefined							
TREG5H	16-bit timer register 5 high	33H (Prohibit RMW)	–							
			W							
			Undefined							
CAP1L	Capture register 1 low	34H	–							
			R							
			Undefined							
CAP1H	Capture register 1 high	35H	–							
			R							
			Undefined							
CAP2L	Capture register 2 low	36H	–							
			R							
			Undefined							
CAP2H	Capture register 2 high	37H	–							
			R							
			Undefined							
T4MOD	16-bit timer 4 source CLK & mode	38H	CAP2T5	EQ5T5	CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TFF5 INV TRG 0: TRG disable 1: TRG enable		0:Software capture 1: Don't care	Capture timing 00: Disable 01: TI4 ↑ TI5 ↑ 10: TI4 ↑ TI4 ↓ 11: TFF1 ↑ TFF1 ↓		1: UC4 clear enable	Source clock 00: TI4 01: ϕT1 10: ϕT4 11: ϕT16	
T4FFCR	16-bit timer 4 flip-flop control	39H	TFF5C1	TFF5C0	CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
			W		R/W				W	
			1	1	0	0	0	0	1	1
			00: Invert TFF5 01: Set TFF5 10: Clear TFF5 11: Don't care		TFF4 invert trigger 0: Trigger disable 1: Trigger enable				00: Invert TFF4 01: Set TFF4 10: Clear TFF4 11: Don't care	

(3) Timer control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
T45CR	T4, T5 control	3AH	QCU						DB6EN	DB4EN
			R/W						R/W	
			0						0	0
			Warm-up timer control						1: Double buffer enable	
TREG6L	16-bit timer register 6 low	40H (Prohibit RMW)	–							
			W							
			Undefined							
TREG6H	16-bit timer register 6 high	41H (Prohibit RMW)	–							
			W							
			Undefined							
TREG7L	16-bit timer register 7 low	42H (Prohibit RMW)	–							
			W							
			Undefined							
TREG7H	16-bit timer register 7 high	43H (Prohibit RMW)	–							
			W							
			Undefined							
CAP3L	Capture register 3 low	44H	–							
			R							
			Undefined							
CAP3H	Capture register 3 high	45H	–							
			R							
			Undefined							
CAP4L	Capture register 4 low	46H	–							
			R							
			Undefined							
CAP4H	Capture register 4 high	47H	–							
			R							
			Undefined							
T5MOD	16-bit timer 5 source CLK & mode	48H			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
					W	R/W				
					1	0	0	0	0	0
					0:Software capture 1: Don't care	Capture timing 00: Disable 01: TI6 ↑ TI7 ↑ 10: TI6 ↑ TI6 ↓ 11: TFF1 ↑ TFF1 ↓		1: UC5 clear enable	Source clock 00: TI6 01: ϕT1 10: ϕT4 11: ϕT16	
T5FFCR	16-bit timer 5 flip-flop control	49H			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
					R/W				W	
					0	0	0	0	1	1
					TFF6 invert trigger 0: Trigger disable 1: Trigger enable				00: Invert TFF6 01: Set TFF6 10: Clear TFF6 11: Don't care	

(4) Watchdog timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	Watchdog timer mode	5CH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
			R/W							
			1	0	0	0	0	0	0	0
			1: WDT enable	00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Warm-up time 0: 2^{14}/ inputted frequency 1: 2^{16}/ inputted frequency	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		1: Connect internally WDT out to reset pin	1: Drive the pin in stop mode
WDCR	Watchdog timer control register	5DH	–							
			W							
			–							
			B1H: WDT disable code				4EH: WDT clear code			

(5) Serial channel (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial channel 0 buffer	50H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC0CR	Serial channel 0 control	51H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity enable	1: Error Overrun	1: Error Parity	1: Error Framing	0: SCLK0 ↑ 1: SCLK0 ↓	1: Input SCLK0 pin
SC0MOD	Serial channel 0 mode	52H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmission data bit8	1: CTS enable	1: Receive enable	1: Wake up enable	00: I/O interface 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TO0 trigger 01: Baud rate generator 10: Internal clock ϕ1 11: SCLK input	
BR0CR	Baud rate control	53H	–	BR0ADD	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W	R/W						
			0	0	0	0	0	0	0	0
			Fix at "0".	1: + (16 − K) /16 divided enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set frequency divisor 0 to F			
BRADD0	Baud rate control	6BH					BR0K3	BR0K2	BR0K1	BR0K0
							R/W			
							0	0	0	0
							Set frequency divisor 1 to F ("0" prohibited)			
SC1BUF	Serial channel 1 buffer	54H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC1CR	Serial channel 1 control	55H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity enable	1: Error Overrun	1: Error Parity	1: Error Framing	0: SCLK1 ↑ 1: SCLK1 ↓	1: Input SCLK1 pin
SC1MOD	Serial channel 1 mode	56H	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmission data bit8	Fix at "0".	1: Receive enable	1: Wake up enable	00: I/O interface 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TO0 trigger 01: Baud rate generator 10: Internal clock ϕ1 11: SCLK input	
BR1CR	Baud rate control	57H	–	BR1ADD	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W	R/W	R/W					
			0	0	0	0	0	0	0	0
			Fix at "0".	1: + (16 − K) /16 divided enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set frequency divisor 0 to F			

(5) Serial channel (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
BRADD1	Baud rate control	6CH					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Set frequency divisor 1 to F ("0" prohibited)			
ODE	Serial open-drain enable	58H							ODE1	ODE0
									R/W	
									0	0
									1: P93 open drain	1: P90 open drain
SC2BUF	Serial channel buffer	2CH	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC2CR	Serial channel control	2DH	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity enable	1: Error			0: SCLK2 ↑ 1: SCLK2 ↓	1: Input SCLK2 pin
						Overrun	Parity	Framing		
SC2MOD	Serial channel mode	2EH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmission data bit8	1: CTS2 enable	1: Receive enable	1: Wake up enable	00: I/O interface 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TO0 trigger 01: Baud rate generator 10: Internal clock ϕ1 11: SCLK input	
BR2CR	Baud rate control	2FH	–	BR2ADD	BR2CK1	BR2CK0	BR2S3	BR2S2	BR2S1	BR2S0
			R/W	R/W	R/W					
			0	0	0	0	0	0	0	0
			Fix at "0".	1: + (16 − K) /16 divided enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set frequency divisor 0 to F			
BRADD2	Baud rate control	2BH					BR2K3	BR2K2	BR2K1	BR2K0
							R/W			
							0	0	0	0
							Set frequency divisor 1 to F ("0" prohibited)			
SC3BUF	Serial channel buffer	3CH	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC3CR	Serial channel 1 control	3DH	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity enable	1: Error			0: SCLK3 ↑ 1: SCLK3 ↓	1: Input SCLK3 pin
						Overrun	Parity	Framing		

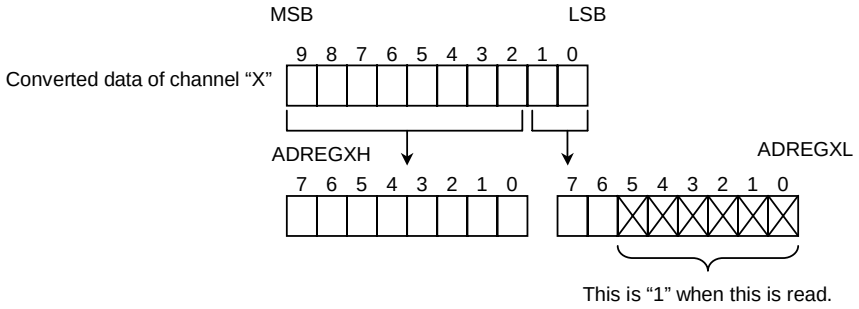
(5) Serial channel (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC3MOD	Serial channel mode	3EH	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmission data bit8	1: CTS3 enable	1: Receive enable	1: Wake up enable	00: I/O interface 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TO0 trigger 01: Baud rate generator 10: Internal clock φ1 11: SCLK input	
BR3CR	Baud rate control	3FH	–	BR3ADD	BR3CK1	BR3CK0	BR3S3	BR3S2	BR3S1	BR3S0
			R/W	R/W	R/W					
			0	0	0	0	0	0	0	0
			Fix at "0".	1: + (16 − K) /16 divided enable	00: φT0 01: φT2 10: φT8 11: φT32		Set frequency divisor 0 to F.			
BRADD3	Baud rate control	3BH					BR3K3	BR3K2	BR3K1	BR3K0
							R/W			
							0	0	0	0
							Set "K" (N + (16 − k)/16 divided) 1 to F ("0" prohibited)			
SC4BUF	Serial channel buffer	4CH	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving) /W (Transmission)							
			Undefined							
SC4CR	Serial channel control	4DH	RB8	EVEN	PE	OERR	PERR	FERR		
			R	R/W		R (Cleared to 0 by reading)				
			Undefined	0	0	0	0	0		
			Receiving data bit	Parity 0: Odd 1: Even	1: Parity enable	1: Error				
						Overrun	Parity	Framing		
SC4MOD	Serial channel mode	4EH	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmission data bit8	Fix at "0".	1: Receive enable	1: Wake up enable	00: Reserved 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TO0 trigger 01: Baud rate generator 10: Internal clock φ1 11: SCLK in	
BR4CR	Baud rate control	4FH	–	BR4ADD	BR4CK1	BR4CK0	BR4S3	BR4S2	BR4S1	BR4S0
			R/W	R/W	R/W					
			0	0	0	0	0	0	0	0
			Fix at "0".	1: + (16 − K) /16 divided enable	00: φT0 01: φT2 10: φT8 11: φT32		Set frequency divisor 0 to F			
BRADD4	Baud rate control	4BH					BR4K3	BR4K2	BR4K1	BR4K0
							R/W			
							0	0	0	0
							Set "K" (N + (16 − k)/16 divided) 1 to F ("0" prohibited)			

(6) AD converter control

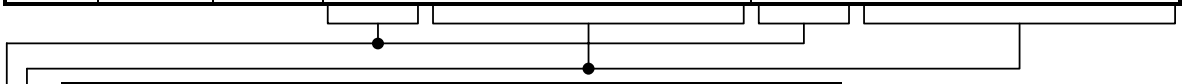
Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD1	AD mode register 1	5EH	EOCF	ADBF	REPET	SCAN		ADS		
			R		R/W			R/W		
			0	0	0	0		0		
			1: End	1: Busy	1: Repeat	1: Scan		1: Start		
ADMOD2	AD mode register 2	5FH	VREFON		SPEED1	SPEED0		ADCH2	ADCH1	ADCH0
			R/W		R/W			R/W		
			1		0	0		0	0	0
			String Resistance Switch ON/OFF		SPEED			Analog input channel select		
AD REG04L *1)	AD result register 0/4 low	60H	ADR01	ADR00						
			R							
			Undefined		1	1	1	1	1	1
AD REG04H	AD result register 0/4 high	61H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
AD REG15L *1)	AD result register 1/5 low	62H	ADR11	ADR10						
			R							
			Undefined		1	1	1	1	1	1
AD REG15H	AD result register 1/5 high	63H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
AD REG26L *1)	AD result register 2/6 low	64H	ADR21	ADR20						
			R							
			Undefined		1	1	1	1	1	1
AD REG26H	AD result register 2/6 high	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
AD REG37L *1)	AD result register 3/7 low	66H	ADR31	ADR30						
			R							
			Undefined		1	1	1	1	1	1
AD REG37H	AD result register 3/7 high	67H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

*1: Data to be stored in AD conversion result register low are the lower 2 bits of the conversion result. The contents of the lower 6 bits of this register are always read as “1”.



(7) Interrupt control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE 0AD	Interrupt enable 0 & AD	70H (Prohibit RMW)	IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INTAD				INT0			
INTE45	Interrupt enable 4/5	71H (Prohibit RMW)	I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INT5				INT4			
INTE67	Interrupt enable 6/7	72H (Prohibit RMW)	I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INT7				INT6			
INTET10	Interrupt enable timer 1/0	73H (Prohibit RMW)	IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INTT1 (Timer 1)				INTT0 (Timer 0)			
INTEPW 10	Interrupt enable PWM 1/0	74H (Prohibit RMW)	IPW1C	IPW1M2	IPW1M1	IPW1M0	IPW0C	IPW0M2	IPW0M1	IPW0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INTT3 (Timer 3/PWM1)				INTT2 (Timer 2/PWM0)			
INTET54	Interrupt enable T register 5/4	75H (Prohibit RMW)	IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INTTR5 (TREG5)				INTTR4 (TREG4)			
INTET76	Interrupt enable T register 7/6	76H (Prohibit RMW)	IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	
			INTTR7 (TREG7)				INTTR6 (TREG6)			
INTES0	Interrupt enable serial 0	77H (Prohibit RMW)	ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INTTX0				INTRX0			
INTES1	Interrupt enable serial 1	78H (Prohibit RMW)	ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INTTX1				INTRX1			
INTES2	Interrupt enable serial 2	59H (Prohibit RMW)	ITX2C	ITX2M2	ITX2M1	ITX2M0	IRX2C	IRX2M2	IRX2M1	IRX2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INTTX2				INTRX2			
INTES3	Interrupt enable serial 3	5AH (Prohibit RMW)	ITX3C	ITX3M2	ITX3M1	ITX3M0	IRX3C	IRX3M2	IRX3M1	IRX3M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INTTX3				INTRX3			
INTES4	Interrupt enable serial 4	5BH (Prohibit RMW)	ITX4C	ITX4M2	ITX4M1	ITX4M0	IRX4C	IRX4M2	IRX4M1	IRX4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
			INTTX4				INTRX4			



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	Indicate no interrupt request.	Clear interrupt request flag.
1	Indicate interrupt request.	– – – – – Don’t care – – – – –

(7) Interrupt control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA 0 request vector	7CH (Prohibit RMW)			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					W					
					0	0	0	0	0	0
					Micro DMA0 start vector					
DMA1V	DMA 1 request vector	7DH (Prohibit RMW)			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					W					
					0	0	0	0	0	0
					Micro DMA1 start vector					
DMA2V	DMA 2 request vector	7EH (Prohibit RMW)			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					W					
					0	0	0	0	0	0
					Micro DMA2 start vector					
DMA3V	DMA 3 request vector	7FH (Prohibit RMW)			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					W					
					0	0	0	0	0	0
					Micro DMA3 start vector					
IIMC	Interrupt input mode control	7BH (Prohibit RMW)						I0IE	I0LE	NMIREE
								W	W	W
								0	0	0
								1: INT0 input enable	0: INT0 edge mode 1: INT0 level mode	1: Operate even at $\overline{NMI}$ rise edge

(8) Chip select/wait controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT control register	68H (Prohibit RMW)	B0E		B0CAS	B0BUS	B0W1	B0W0	B0C1	B0C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: B0CS master bit		0: $\overline{CS0}$ 1: $\overline{CAS0}$	0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + N) waits 11: 0 waits		00: 7F00H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B1CS	Block 1 CS/WAIT control register	69H (Prohibit RMW)	B1E		B1CAS	B1BUS	B1W1	B1W0	B1C1	B1C0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			1: B1CS master bit		0: $\overline{CS1}$ 1: $\overline{CAS1}$	0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + N) waits 11: 0 waits		00: 1080H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
B2CS	Block 2 CS/WAIT control register	6AH (Prohibit RMW)	B2E		B2CAS	B2BUS	B2W1	B2W0	B2C1	B2C0
			W		W	W	W	W	W	W
			1		0	0	0	0	0	0
			1: B2CS master bit		0: $\overline{CS2}$ 1: $\overline{CAS2}$	0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + N) waits 11: 0 waits		00: 28000H to 01: 400000H to 10: 800000H to 11: C00000H to	

Note: After reset, only “Block 2” is set to enable.

(9) Clock control

Symbol	Name	Address	7	6	5	4	3	2	1	0
CKOCR	Clock output control register	006DH					SCOSEL	SCOEN	ALEEN	CLKEN
							R/W			
							0	0	0	0
							SCOUT select 0: f_{FPH} 1: f_{SYS}	SCOUT output control 0: I/O port 1: SCOUT output	ALE pin control 0:High-Z output 1: ALE output	CLK pin control 0:High-Z output 1: CLK output
SYSCR0	System clock control register 0	006EH	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
			R/W							
			1	0	1	0	0	0	0	0
			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 (fs) after released STOP mode 0: Stop 1: Oscillation	Select clock after released STOP mode 0: fc 1: fs	Warm-up timer (Write) 0: Don't care 1: Start timer (Read) 0: End warm up 1: Not end warm up	Select prescaler clock 00: f_{FPH} 01: fs 10: fc/16 11: (Reserved)	
SYSCR1	System clock control register 1	006FH					SYSCK	GEAR2	GEAR1	GEAR0
							R/W			
							0	1	0	0
							Select system clock 0: fc 1: fs (Note 1)	Select gear value of high frequency (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: Reserved) 110: (Reserved) 111: (Reserved)		

Note 1: The high-frequency oscillator will be enabled regardless the value of SYSCR0<XEN> when SYSCR1<SYSCK> is set to "0".
On the other hand, the low frequency oscillator will be enabled regardless the value of SYSCR0<XTEN> when SYSCR1<SYSCK> is set to "1".

Note 2: CKOCR<bit7:4> is read to "1".

6. Port Section Equivalent Circuit Diagram

- Reading the circuit diagram

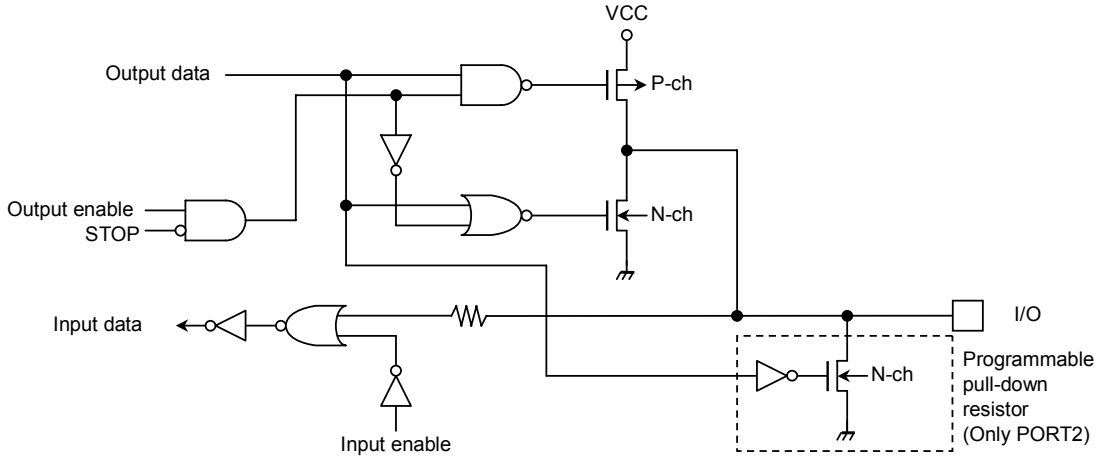
  Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

  The dedicated signal is described below.

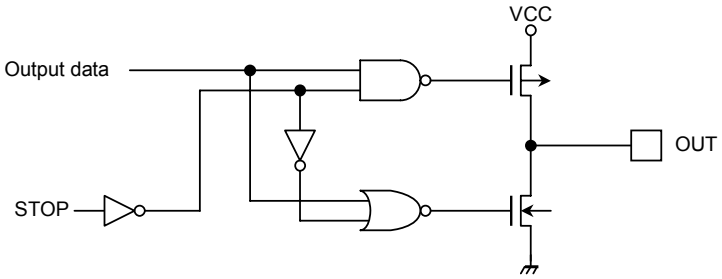
  STOP: This signal becomes active "1" when the halt mode setting register is set to the STOP mode and the CPU executes the halt instruction. When the drive enable bit WDMOD<DRVE> is set to "1", however, stop remains at "0".

- The input protection resistance ranges from several tens of ohms to several hundreds of ohms.

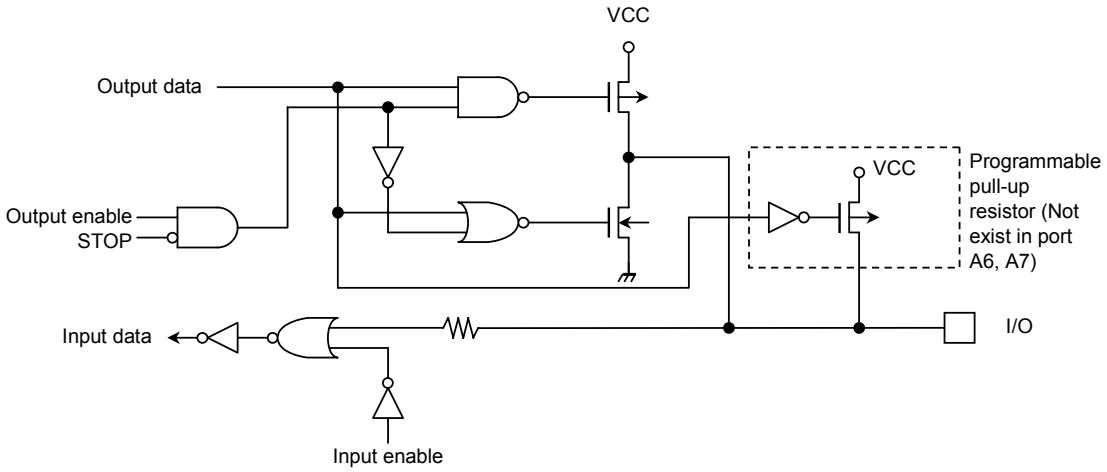
■ P0 (AD0 to AD7), P1 (AD8 to AD15, A8 to A15), P2 (A16 to A23, A0 to A7)



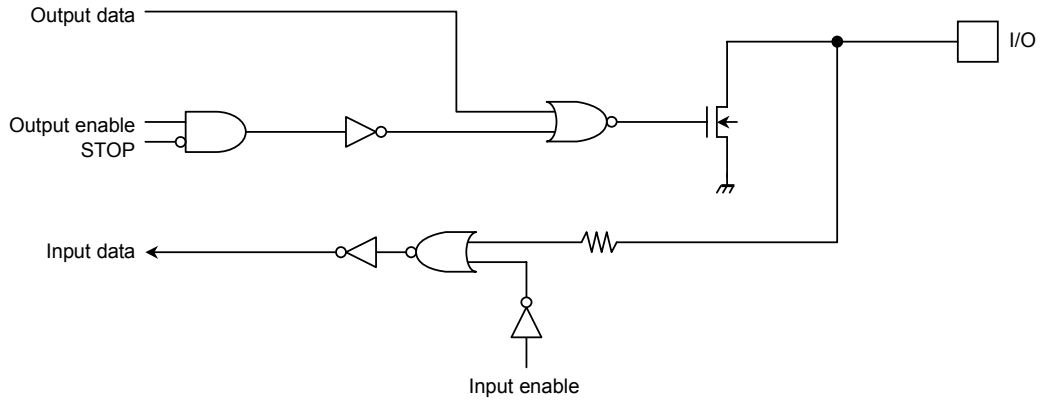
■ P30 ($\overline{RD}$), P31 ($\overline{WR}$)



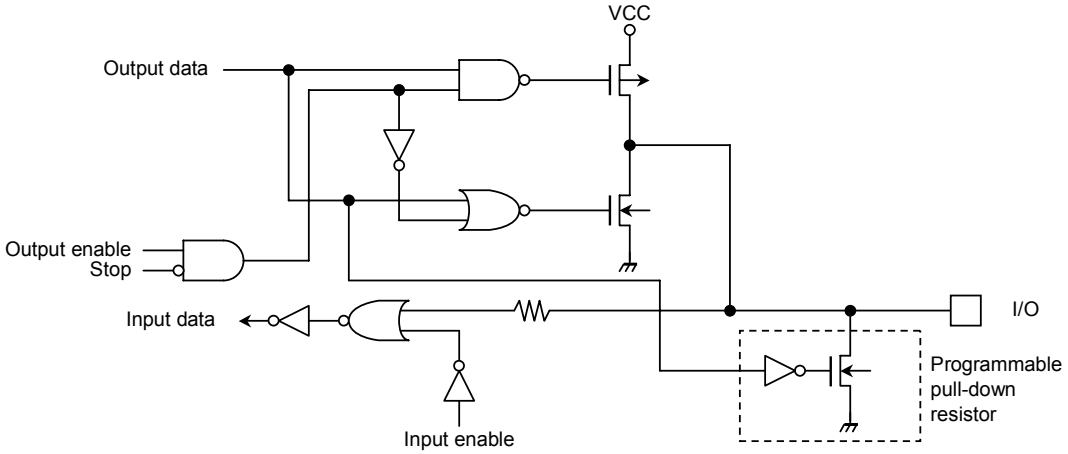
■ P32 to P37, P40 to P41, P61 to P62, P64 to P65, P67, P7, P80 to P86, P91 to P92, P94 to P95, PA6 to PA7



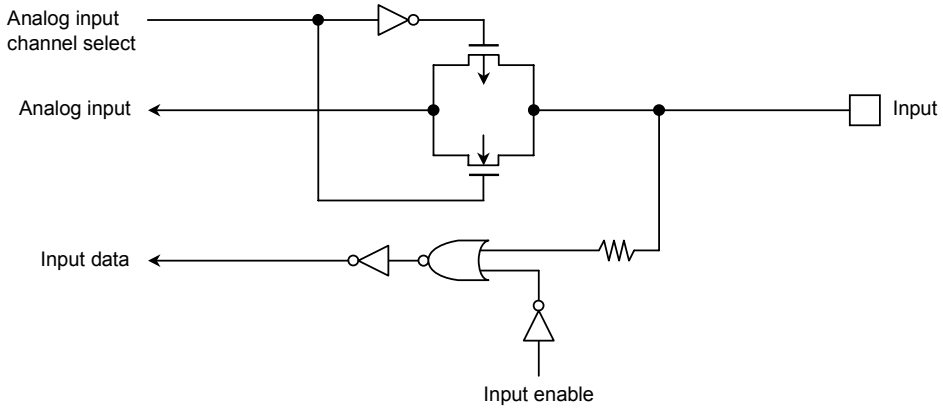
■ PA0 to PA5



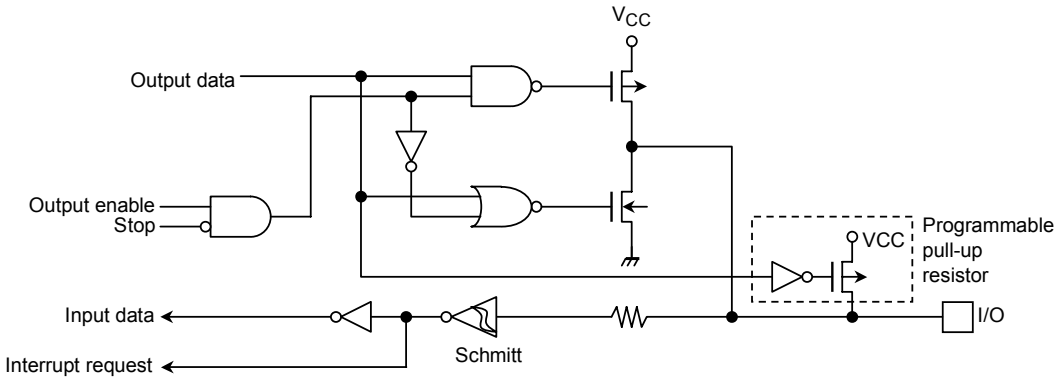
■ P42 ($\overline{CS2}$ and $\overline{CAS2}$)



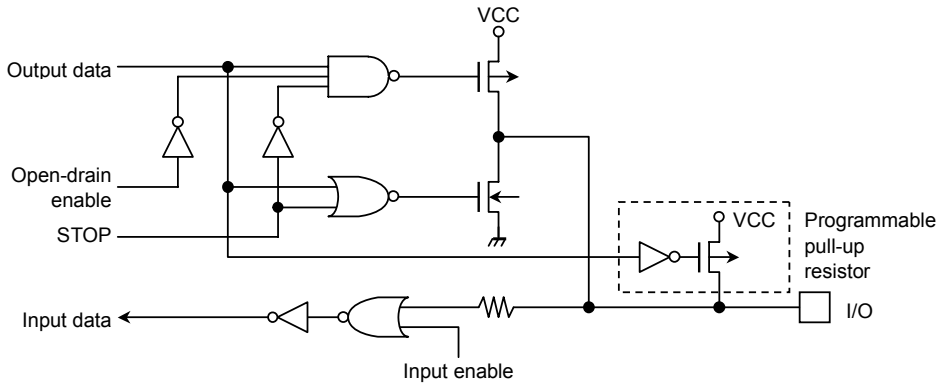
■ P5 (AN0 to AN7)



■ P87 (INT0)

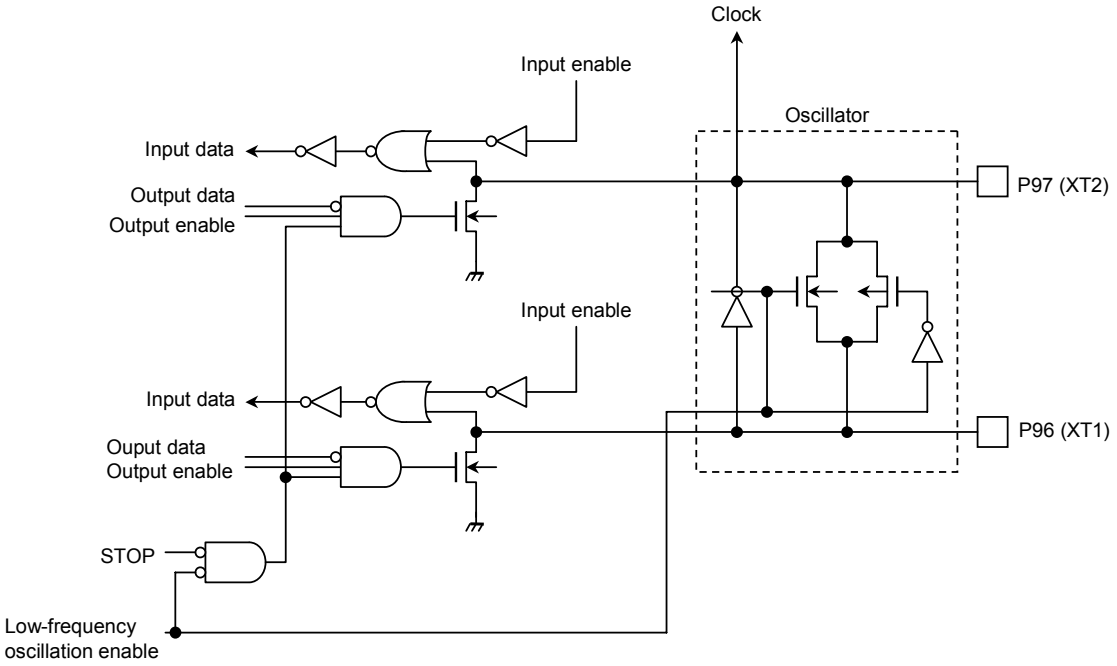


■ P90 (TXD0), P93 (TXD1), P60 (TXD2), P63 (TXD3), P66 (TXD4)

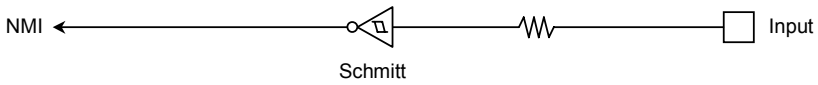


Note: P60, P63 and P66 have no open-drain function.

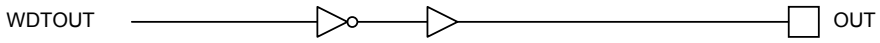
■ P96 (XT1) and P97 (XT2)



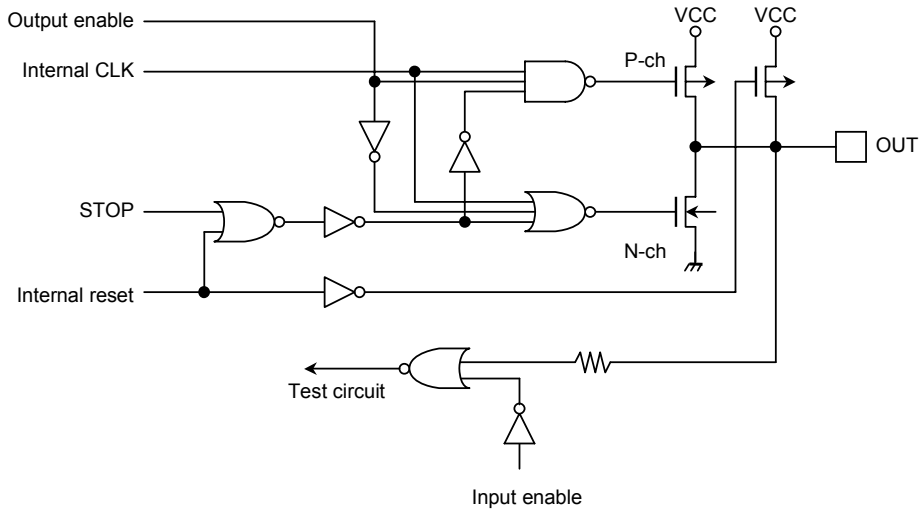
■ $\overline{\text{NMI}}$



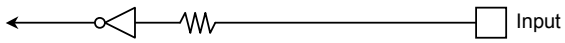
■ $\overline{\text{WDTOUT}}$



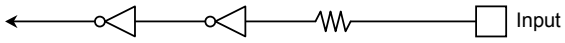
■ CLK



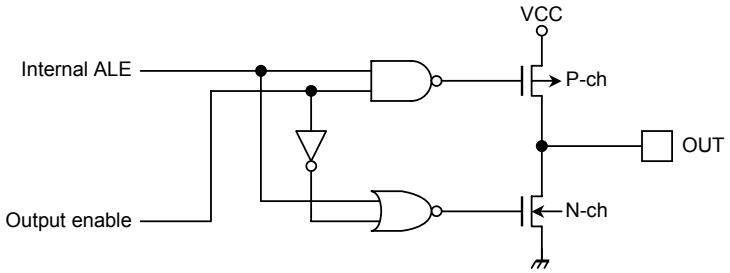
■ $\overline{EA}$



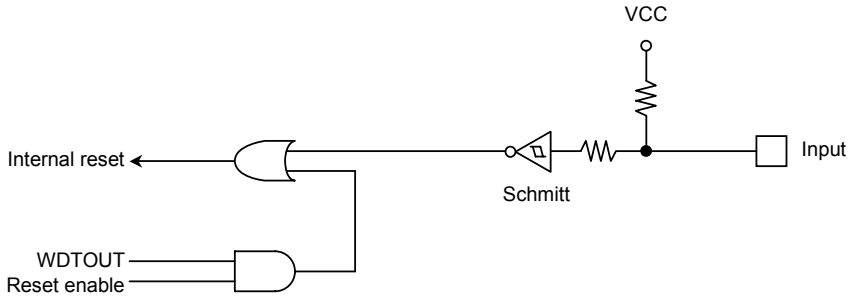
■ AM8/ $\overline{AM16}$



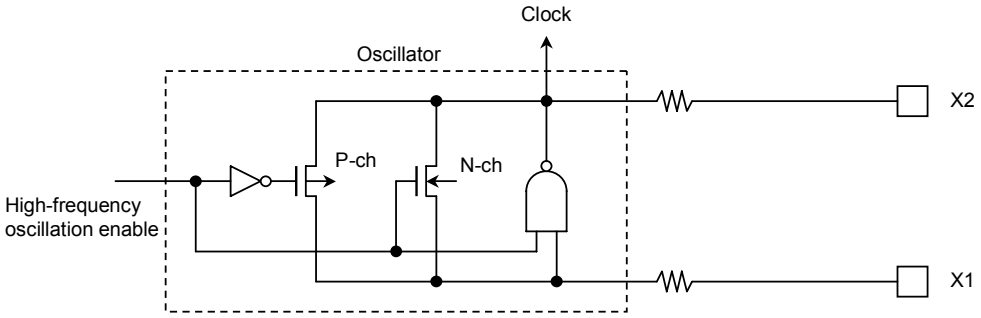
■ ALE



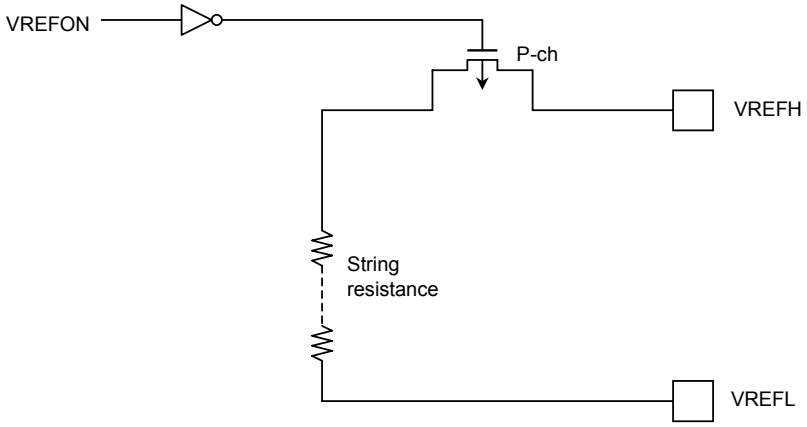
■ $\overline{RESET}$



■ X1, X2



■ VREFH, VREFL



7. Points of Note and Restrictions

(1) Special expression

1. Explanation of a built-in I/O register: Register symbol<Bit symbol>

   Example: TRUN<T0RUN> … Bit T0RUN of register TRUN

2. Read, modify and write instruction

   An instruction which CPU reads data from memory and writes the data to the same memory location by one instruction.

   Example 1: SET 3, (TRUN) … Set bit3 of TRUN

   Example 2: INC 1, (100H) … Increment the data of 100H

   - The representative read-modify-write instructions in the TLCS-900

   Exchange instruction

EX	(mem), R		

   Arithmetic operation

ADD	(mem), R/#	ADC	(mem), R/#
SUB	(mem), R/#	SBC	(mem), R/#
INC	#3, (mem)	DEC	#3, (mem)

   Logic operation

AND	(mem), R/#	OR	(mem), R/#
XOR	(mem), R/#		

   Bit manipulation

STCF	#3/A, (mem)	RES	#3, (mem)
SET	#3, (mem)	CHG	#3, (mem)
TSET	#3, (mem)		

   Rotate, Shift

RLC	(mem)	RRC	(mem)
RL	(mem)	RR	(mem)
SLA	(mem)	SRA	(mem)
SLL	(mem)	SRL	(mem)
RLD	(mem)	RRD	(mem)

3. fc, f_{FPH}, f_{SYS}, 1 state

   The clock frequency input from X1, X2 pins is called fc, and the clock selected by SYSCR1<SYSCK> is called f_{FPH}, and f_{FPH} divided by 2 clock frequency is called f_{SYS}. One cycle of f_{SYS} is called 1 state.

(2) Care points

1. $\overline{EA}$, AM8/$\overline{AM16}$ pin

   Fix these pins to V_{CC} or GND unless changing voltage.

2. TEST1, TEST2 pin

   Connect the TEST1 pin with the TEST2 pin.

3. Reserved area in memory space

   The 256 bytes of memory area between FFFF00H to FFFFFFH can not be used because they are reserved.

4. Standby mode (IDLE1)

   When the IDLE1 mode (Operates only oscillator) is used, set TRUN<PRRUN> to "0" to stop prescaler before "HALT" instruction is executed.

5. Warm-up counter

   The warm-up counter operates when the STOP mode is released even if the system uses an external oscillator. As a result, it takes the warm-up time from inputting the releasing request to outputting the system clock.

6. Micro DMA (DRAM refresh mode)

   When the bus is released ($\overline{BUSAK}$ = "0"). DRAM refresh cannot be performed because of the micro DMA cannot access the bus.

7. Programmable pull-up/pull-down resistance

   The programmable pull-up/pull-down resistors can be selected ON/OFF by the program when the ports are used as the input ports. In case where the ports are used as outputs, they can not be selected ON/OFF by program.

   The data registers (e.g., P2, P3 ...) are used for selecting ON/OFF of pull-up/pull-down resistors. As a result, read-modify-write instructions are prohibited.

8. Bus releasing function

   Refer to the note about the bus release in 3.5 "Functions of Ports" because the pin state, when the bus is released, is written.

9. Watchdog timer

   The watchdog timer starts operation immediately after the reset is released. When the watchdog timer is not used, disable watchdog timer.

10. Watchdog timer

    When the bus is released, both internal memory and internal I/O can not be accessed. But the internal I/O continues to operate. So, the watchdog timer continues to run. Therefore, be careful about the bus releasing time and set the detection timer of watchdog timer.

11. AD converter

    The ladder resistor between VREFH and VREFL pins can be cut by program to reduce the power consumption. When the standby mode is used, disable in the program before the "HALT" instruction is executed. And set ADMOD2<SPEED1:0> = "00".

12. CPU (Micro DMA)

    Only the "LDC cr, r", "LDC r, cr" instruction can be used to access the control registers like transfer source address register (DMASn) in the CPU.

13. POP SR instruction

    Please execute POP SR instruction during DI condition.

14. Pin states in STOP mode

    Open-drain output state. Input gate in operation. Set output to “L” or attach pull up on pin so that the input gate stays constant.

15. Releasing the HALT mode by requesting an interruption

    Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0) which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2/RUN are not applicable to this case). (In this case, an interrupt request is kept on hold internally)

    If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

8. Package Dimensions

P-LQFP100-1414-0.50F

Unit: mm