TOSHIBA

Preliminary

TMP92CW10F

Specification

Revision 1.1 17 January 2001

Histor

Revision	Date	Note
1.0	28 Dec. 2000	Initial draft
1.1	17 Jan. 2001	A mistake correction

CMOS 32-bit Micro-controller

TMP92CW10F

Preliminary

1. Outline and Device Characteristics

TMP92CW10 is high-function 32-bit microcontroller incorporating Toshiba's proprietary TLCS-900/H1 CPU. It is suitable for various embedded control equipment applications which require high-spped-processing data capability.

The product comes in a compact 100-pin flat package and tus help to meet user system requirements for high-density mounting.

(1) CPU : 32-bit CPU (900/H1 CPU)
- Compatible with TLCS-900, 900/L, 900/H, 900/L1 and 900/H2's instruction code
- 16Mbytes of linear address space
- General-purpose register and register banks
- Micro DMA : 8channels. 250ns / 4bytes (when operating internally at fc = 20 MHz)

(2) Minimum instruction execution time : 50ns (when operating internally at fc = 20 MHz)
- Internal data bus : 32-bit

(3) Internal memory
- Internal RAM : 6Kbyte (can use for code section)
- Internal ROM : 128Kbyte Mask ROM

(4) External memory expansion
- 16M-byte linear address space (memory mapped I/O)
- External data bus : 8-bit
- * Can't use upper address bus when built-in I/Os are selected

(5) Memory controller
- Chip select output : 1 channel

(6) 8-bit timer : 8 channels
- 8-bit interval timer mode (8 channels)
- 16-bit interval timer mode (4 channels)
- 8-bit programmable pulse generation (PPG) output mode (4 channels)
- 8-bit pulse width modulation (PWM) output mode (4 channels)

(7) 16-bit timer : 2 channels
- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) output mode
- Frequency measurement mode
- Pulse width measurement (PWM) mode
- Time differential measurement mode

(8) Serial interface : 2 channels
- I/O interface mode
- Universal asynchronous receiver transmitter (UART) mode

(9) Serial expansion interface : 2 channels
- Baud rate 8/4/2/0.5Mbps at fc=20MHz.

(10) Serial bus interface : 2 channels
- Clocked-synchronous mode
- I^2C bus mode

(11) 10-bit A/D converter : 12 channels
- A/D conversion time 8μsec at fc=20MHz.
- Total tolerance ±3LSB (excluding quantization error)
- Scan mode for all 12channels

(12) Watch dog timer

(13) Interrupt controller
- 32 internal interrupts
- 9 external interrupts

Any one of 7 intrrupt priority lebels can be set (but not for NMI)

(14) I/O Port : 70pins

(15) Power supply voltage
- VCC5 = 5V±10% (4.5V to 5.5V)
- VCC3 = 3.3V±0.3V (3V to 3.6V)

(17) Operating temperature : -40 to 85 ℃

(18) Package : 100-pin LQFP (14×14×1.4 mm. 0.5 mm pitch)

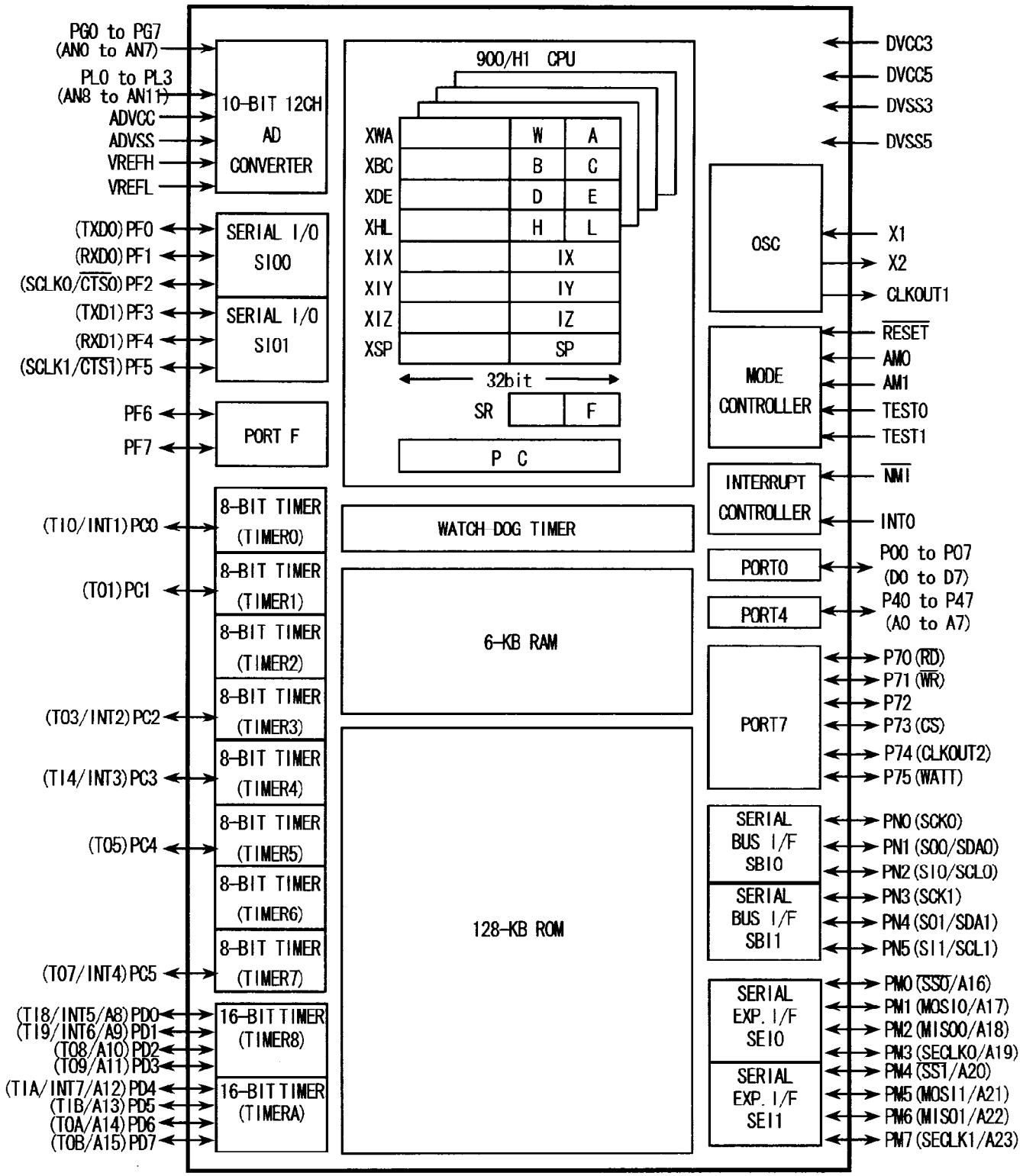


Figure 1 TMP92CW10 block diagram

2. Pin Assignment and Functions

2.1 Pin Assignment

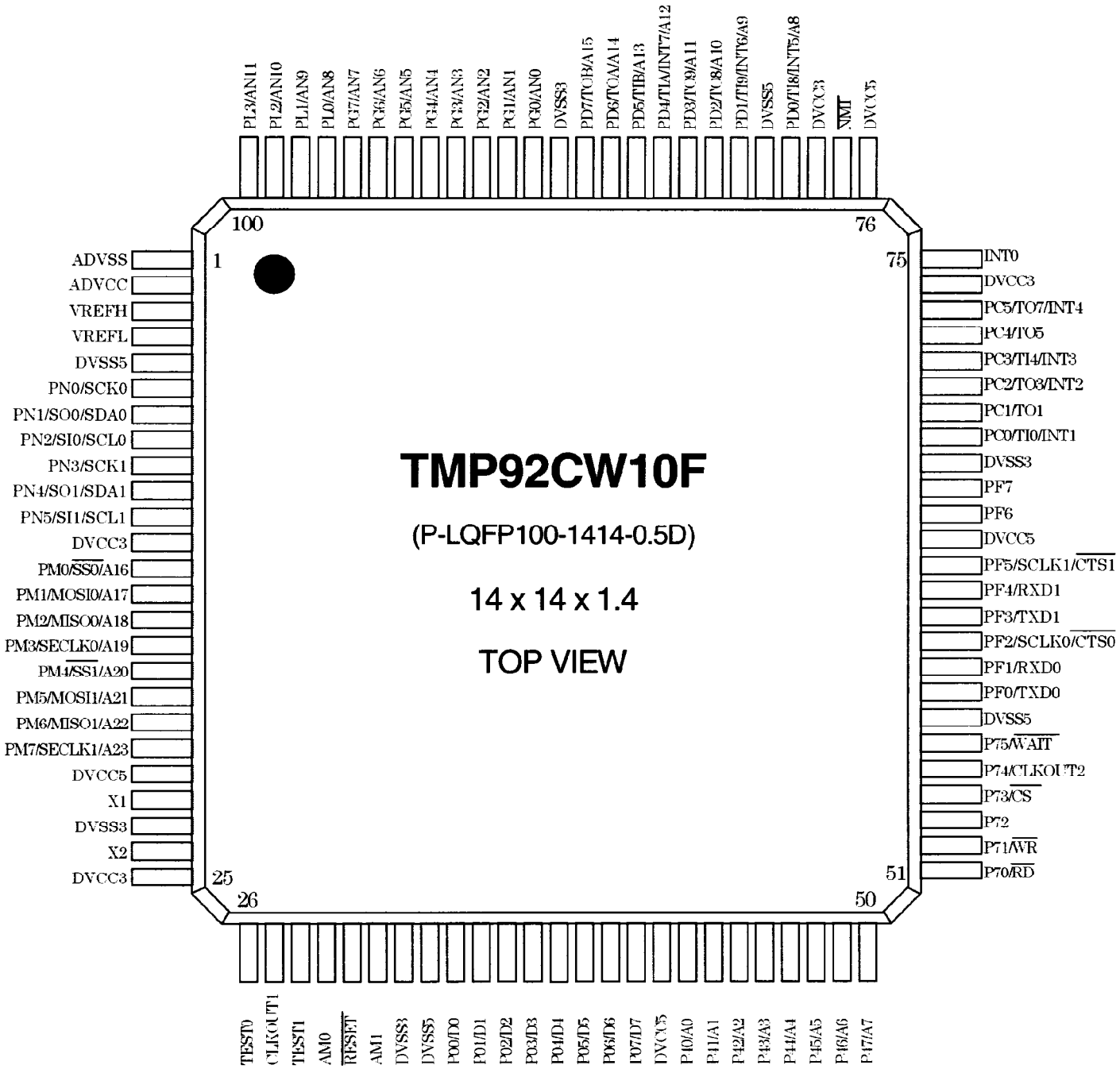


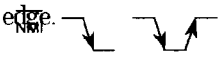
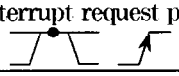
Figure 2.1 Pin Assignment

2.2 Pin names and functions

The following table shows the names and functions of the input/output pi

Pin Name	Number of pins	In/Out	Function
P00..P07 D0..D7	8 (CMOS) (TTL)	in/out in/out	Port 0: I/O port. Input or output specifiable in units of bits. Data: Data bus 0 to 7.
P40..P47 A0..A7	8	in/out out	Port4: I/O port. Input or output specifiable in units of bits. Address: Address bus 0 to 7.
P70 $\overline{RD}$	1	in/out out	Port70: I/O port. Read: Outputs strobe signal to read external memory.
P71 $\overline{WR}$	1	in/out out	Port 71: I/O port. Write: Output strobe signal to write data on pins.
P72	1	in/out	Port 72: I/O port.
P73 $\overline{CS}$	1	in/out out	Port 73: I/O port. Chip select: Outputs "low" if address is within specified address area.
P74 CLKOUT2	1	in/out out	Port 74: I/O port. Clock output 2: CLKOUT2 output 4 MHz clock at fc = 20 MHz.
P75 $\overline{WAIT}$	1	in/out in	Port 75: I/O port. Wait: Signal used to request CPU bus wait.
PC0 TI0 INT1	1	in/out in in	Port C0: I/O port. Timer input 0: Input pin for timer 0. Interrupt request pin 1: Rising-edge interrupt request pin.
PC1 TO1	1	in/out out	Port C1: I/O port. Timer output 1: Output pin for timer 1.
PC2 TO3 INT2	1	in/out out in	Port C2: I/O port. Timer output 3: Output pin for timer 3. Interrupt request pin 2: Rising-edge interrupt request pin.
PC3 TI4 INT3	1	in/out in in	Port C3: I/O port. Timer input 4: Input pin for timer 4. Interrupt request pin 3: Rising-edge interrupt request pin.
PC4 TO5	1	In/out Out	Port C4: I/O port. Timer output 5: Output pin for timer 5.
PC5 TO7 INT4	1	In/out Out In	Port C5: I/O port. Timer output 7: Output pin for timer 7. Interrupt request pin 4: Rising-edge interrupt request pin.
PD0 TI8 INT5 A8	1	In/out In In out	Port D0: I/O port. Timer input 8: Input pin for timer 8. Interrupt request pin 5: Interrupt request pin with programmable rising/falling edge. Address: Address bus 8.
PD1 TI9 INT6 A9	1	in/out in in out	Port D1: I/O port. Timer input 9: Input pin for timer 9. Interrupt request pin 6: Rising-edge interrupt request pin. Address: Address bus 9.

Pin Name	Number of pins	In/Out	Function
PD2 TO8 A10	1	in/out out out	Port D2: I/O port. Timer output 8: Output pin for timer 8 Address: Address bus 10.
PD3 TO9 A11	1	in/out out out	Port D3: I/O port. Timer output 9: Output pin for timer 9 Address: Address bus 11.
PD4 TIA INT7 A12	1	in/out in in out	Port D4: I/O port. Timer input A: Input pin for timer A Interrupt request pin 7: Interrupt request pin with programmable rising/falling edge. Address: Address bus 12.
PD5 TIB A13	1	in/out in out	Port D5: I/O port. Timer input B: Input pin for timer B. Address: Address bus 13.
PD6 TOA A14	1	in/out out out	Port D6: I/O port. Timer output A: Output pin for timer A. Address: Address bus 14.
PD7 TOB A15	1	in/out out out	Port D7: I/O port. Timer output B: Output pin for timer B. Address: Address bus 15.
PF0 TXD0	1	in/out out	Port F0: I/O port. Serial transmission data 0.
PF1 RXD0	1	in/out in	Port F1: I/O port. Serial receive data 0.
PF2 SCLK0 /CTS0	1	in/out in/out in	Port F2: I/O port. Serial clock input/output 0. Serial data ready to send 0. (Clear-to-send)
PF3 TXD1	1	in/out out	Port F3: I/O port. Serial transmission data 1.
PF4 RXD1	1	in/out in	Port F4: I/O port. Serial receive data 1.
PF5 SCLK1 /CTS1	1	in/out in/out in	Port F5: I/O port. Serial clock input/output 1. Serial data ready to send 0. (Clear-to-send)
PF6	1	in/out	Port F6: I/O port.
PF7	1	in/out	Port F7: I/O port.
PG0..PG7 AN0..AN7	8	in in	Port G: Input-only port. Analog input 0 to 7: AD converter input pins.
PL0..PL3 AN8..AN11	4	in in	Port L0 to L3: Input-only port. Analog input 8 to 11: AD converter input pins.
PM0 /SS0 A16	1	in/out in out	Port M0: I/O port. SEI slave select input 0. Address: Address bus 16
PM1 MOSI0 A17	1	in/out in/out out	Port M1: I/O port. SEI master output, slave input 0. Address: Address bus 17.
PM2 MISO0 A18	1	in/out in/out out	Port M2: I/O port. SEI master input, slave output 0. Address: Address bus 18.

Pin Name	Number of pins	In/Out	Function
PM3 SECLK0 A19	1	in/out in/out out	Port M3: I/O port. SEI clock input/output 0. Address: Address bus 19.
PM4 $\overline{SST}$ A20	1	in/out in out	Port M4: I/O port. SEI slave select input. Address: Address bus 20.
PM5 MOSI1 A21	1	in/out in/out out	Port M5: I/O port. SEI master output, slave input 1. Address: Address bus 21.
PM6 MISO1 A22	1	in/out in/out out	Port M6: I/O port. SEI master input, slave output 1. Address: Address bus 22.
PM7 SECLK1 A23	1	in/out in/out out	Port M7: I/O port. SEI clock input/output 1. Address: Address bus 23
PN0 SCK0	1	in/out in/out	Port N0: I/O port. SBI interface 0: clock during SIO mode
PN1 SO0 SDA0	1	in/out out in/out	Port N1: I/O port. SBI interface 0: output data at SIO mode SBI interface 0: data at I^2C mode
PN2 SI0 SCL0	1	in/out in in/out	Port N2: I/O port. SBI interface 0: input data at SIO mode SBI interface 0: clock at I^2C mode
PN3 SCK1	1	in/out in/out	Port N3: I/O port. SBI interface 1: clock during SIO mode
PN4 SO1 SDA1	1	in/out out in/out	Port N4: I/O port. SBI interface 1: output data at SIO mode SBI interface 1: data at I^2C mode
PN5 SI1 SCL1	1	in/out in in/out	Port N5: I/O port. SBI interface 1: input data at SIO mode SBI interface 1: clock at I^2C mode
$\overline{NMI}$	1	in	Non-maskable interrupt: Interrupt request pin with programmable falling or both falling and rising edge. 
INT0	1	in	Interrupt request pin 0: Interrupt request pin with programmable level/rising-edge. 
AM0,1	2	in	Address Mode selection: Connect AM0 and AM1 pins to VCC.
TEST0,1	2	in	TEST mode pins :Input "low" when using
CLKOUT1	1	out	Programmable clock output 1
X1/X2	2	in/out	Oscillator connecting pins. Input 3.3V clock signal using external clock.
$\overline{RESET}$	1	in	Reset : Initializes the device.
VREFH	1	in	AD reference voltage high
VREFL	1	in	AD reference voltage low
ADVCC	1	-	Power supply pin for AD converter
ADVSS	1	-	GND pin for AD converter

Pin Name	Number of pins	In/Out	Function
DVCC5	4	-	Power supply pins (+5V): Conect all DVCC5 pins to 5V power supply.
DVSS5	4	-	GND 5V: Connect all DVSS pins to GND (0V).
DVCC3	4	-	Power supply pins (+3.3V): Connect all DVCC pins to 3.3V power supply.
DVSS3	4	-	GND 3.3V: Connect all DVSS3 pins to GND.(0V)

3. OPERATION

This section describes the functions and the basic operation of the TMP92CW10.

3.1 CPU

The CPU functions exclusive to the TMP92CW10 are explained here.

3.1.1 CPU Outline

TMP92CW10 devices have a built-in high-performance 32-bit CPU (the TLCS-900/H1 CPU). (For basics of the CPU for more information about the TLCS-900/H2 CPU, please refer to the previous chapter).

Details of the TMP92CW10 CPU are as follows:

Table 3.1.1 Outline of CPU

Width of CPU Address Bus	24-bit
Width of CPU Data Bus	32-bit
Internal Operating Frequency	20MHz
Minimum Bus Cycle	1-clock access (50ns @20MHz)
Internal RAM	32-bit 1-clock access
Internal ROM	32-bit 1-clock access
Internal I/O	8/16-bit 2-clock access TLCS-900/H1 I/O 8/16-bit 5 to 6-clock access TLCS-900/L1 I/O
External Device	8-bit 2-clock access (can insert some waits)
Minimum Instruction Execution Cycle	1-clock(50ns @20MHz)
Conditional Jump	2-clock(100ns @20MHz)
Instruction Queue Buffer	12-byte
Instruction Set	Compatible with TLCS-900/L1, TLCS-900/H2 (LDX instructions have been omitted.)
CPU mode	Only maximum mode
Micro DMA	8-channel

3.1.2 Reset Operation

When resetting the TMP92CW10 microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the RESET input Low for at least20 system clocks(4 μ s).At reset the clock doubler is bypassed and system clock operates at 5MHz(fc=20MHz).

When the Reset has been accepted, the CPU performs the following:

- Sets the Program Counter (PC) as follows in accordance with the Reset Vector stored at address FFFF00H~FFFF02H:

  PC<0~7> ← data in location FFFF00H
  PC<8~15> ← data in location FFFF01H
  PC<16~23> ← data in location FFFF02H

- Sets the Stack Pointer (XSP) to 00000000H.
- Sets bits <IFF0~IFF2> of the Status Register (SR) to 111 (thereby setting the Interrupt Level Mask Register to level 7).
- Clears bits <RFP0~RFP1> of the Status Register to 00 (thereby selecting Register Bank 0).

When the Reset is released, the CPU starts executing instructions according to the

Program Counter settings. CPU internal registers not mentioned above do not change when the Reset is released.

When the Reset is accepted, the CPU sets internal I/O, ports and other pins as follows.

- Initializes the internal I/O registers as table of "Special Function Register" in Section 5.
- Sets the port pins, including the pins that also act as internal I/O, to General-Purpose Input or Output Port Mode.

When the RESET input pin is released High, the internal reset is released.

After a power-on reset, the operation of the memory controller cannot be guaranteed until the power supply becomes stable. External memory data receives befor e the TNP92CW10 was turned on may be corrupted since the control signals will be unstable until the power supply has stabilized after the power-on reset.

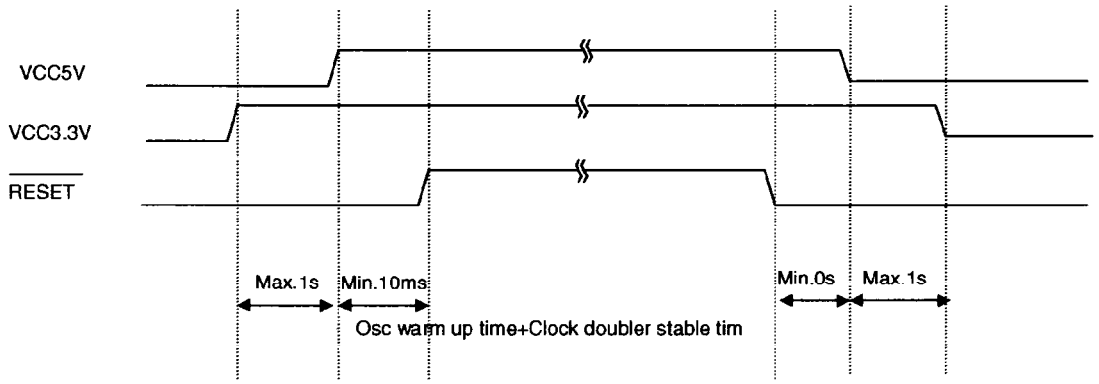


Figure 3.1.1 Power on Reset Timing Example

3.1.3 Setting of TEST0,TEST1,AM0,AM1

Connect TEST0.TEST1 pin to "GND", set AM0.AM1 pin to "1" to use.

Table 3.1.2 Operation Mode Setup Table

Operation Mode	Input pin for mode setting				
	RESET	AM1	AM0	TEST1	TEST0
Single-chip Mod	↗	1	1	0	0

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP92CW10.

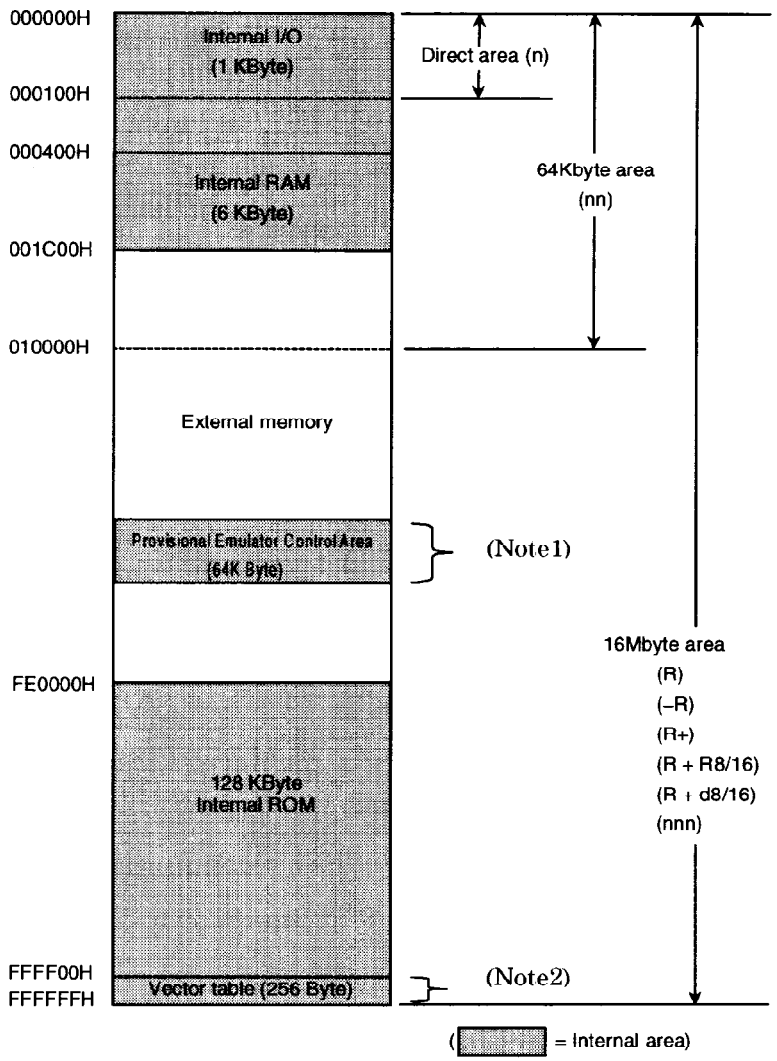


Figure 3.2.1 Memory Map

Note1: Provisional emulator control area is for emulator, it is mapped F00000H to F10000H address after reset.

Note2: Don't use the last 16-byte area (FFFFF0H to FFFFFFH). This area is reserved.

Note3: When the provisional emulator control area is accessed, the WR and RD signals are output. For this reason, care should be exercised when the expanded memory area is used.

3.3 The Clock Function and Standby Function

3.3.1 Block diagram of system clock

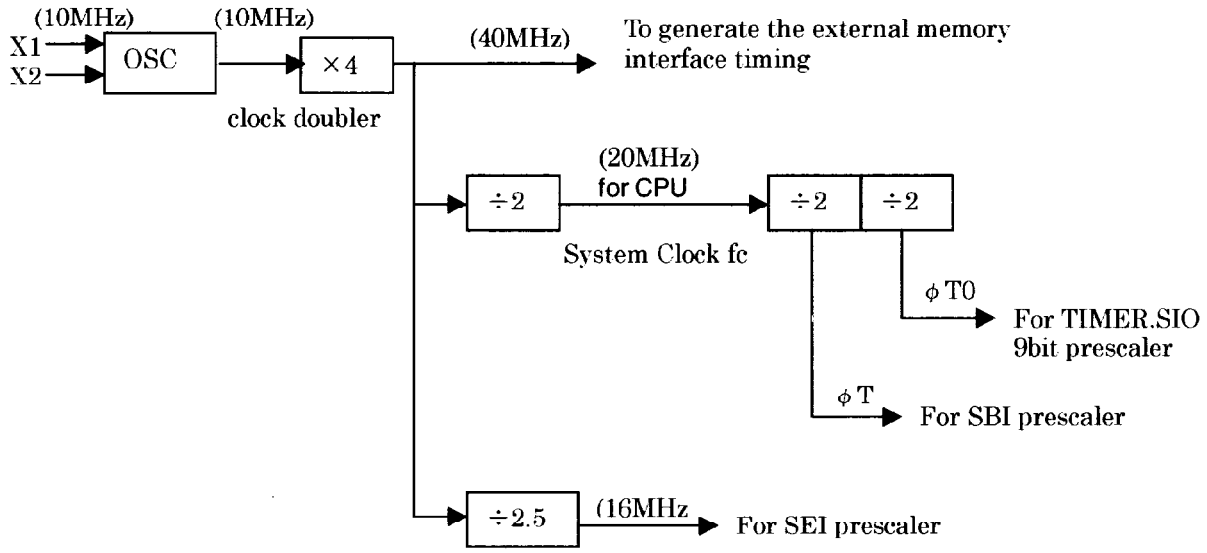


Figure 3.3.1 Block Diagram of System clock

3.3.2 Standby controller

(1) Halt Modes

When the HALT instruction is executed, the operating mode switches to Idle2, Idle1 or Stop Mode, depending on the contents of the CLKMOD<HALTM1,HALTM0> register.

The subsequent actions performed in each mode are as follows:

① IDLE2: The CPU only is halted.

In Idle2 Mode internal I/O operations can be performed by setting the following registers.

Table 3.3.1 Shows the registers of setting operation during Idle2 Mode

Table 3.3.1 Shows the registers of setting operation during Idle2 Mode

Internal I/O	SFR
TIMER0,TIMER1	TRUN01<I2T01>
TIMER2,TIMER3	TRUN23<I2T23>
TIMER4,TIMER5	TRUN45<I2T45>
TIMER6,TIMER7	TRUN67<I2T67>
TIMER8	TRUN8<I2T8>
TIMERA	TRUNA<I2TA>
SIO0	SC0MOD1<I2S0>
SIO1	SC1MOD1<I2S1>
SBI0	SBI0BR0<I2SBI0>
SBI1	SBI1BR0<I2SBI1>
A/D converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>

② Idle1: Only the oscillator continue to operate.

③ Stop: All internal circuits stop operating.

The operation of each of the different Halt Modes is described inTable 3.3.2.

Table 3.3.2 I/O operation during Halt Modes

Halt Mode		Idle2	Idle1	Stop
CLKMOD<HALTM1:0>		11	10	01
Block	CPU	Halt		
	I/O ports	Maintain same state as when HALT instruction was executed.		See Table 3.3.5
	8-bit TMR, 16-bit TMR	Can be selected	Stopped	
	SIO, SBI			
	A/D converter			
	WDT			
	CAN, SEI	Operational		
	Interrupt controller			

(2) How to clear a Halt mode

The Halt state can be cleared by a Reset or by an interrupt request.The combination of the value in <IFF0~IFF2> of the Interrupt Mask Register and the current Halt mode determine in which ways the Halt mode may be cleared. The details associated with each type of Halt state clearance are shown in Table 3.3.5.

- Clearance by interrupt request

Whether or not the Halt mode is cleared and subsequent operation depends on the status of the generated interrupt. If the interrupt request level set before execution of the HALT instruction is greater than or equal to the value in the Interrupt Mask Register, the following sequence takes place: the Halt mode is cleared, the interrupt is then processed, and the CPU then resumes execution starting from the instruction following the HALT instruction. If the interrupt request level set before execution of the HALT instruction is less than the value in the Interrupt Mask Register, the Halt mode is not cleared. (If a non-maskable interrupt is generated, the Halt mode is cleared and the interrupt processed, regardless of the value in the Interrupt Mask Register.)

However, for INT0 only, even if the interrupt request level set before execution of the HALT instruction is less than the value in the Interrupt Mask Register, the Halt mode is cleared. In this case, the interrupt is not processed and the CPU resumes execution starting from the instruction following the HALT instruction. The interrupt request flag remains set to 1.

- Clearance by Reset

Any Halt state can be cleared by Reset.

When Stop Mode is cleared by RESET signal, sufficient time

(at least 10ms@20MHz) must be allowed after the Reset for the operation of the oscillator and clock doubler to stabilize.

When a Halt mode is cleared by resetting, the contents of the internal RAM remain the same as they were before execution of the HALT instruction. However, all other settings are re-initialized. (Clearance by an interrupt affects neither the RAM contents nor any other settings – the state which existed before the HALT instruction was executed is retained.)

Table 3.3.3 Source of Halt state clearance and Halt clearance operation

Status of Received Interrupt			Interrupt Enabled (interrupt level) ≥ (interrupt mask)			Interrupt Disabled (interrupt level) < (interrupt mask)		
Halt mode			Idle2	Idle1	Stop	Idle2	Idle1	Stop
Source of Halt state clearance	Interrupt	NMI	◉	◉	◉*1	–	–	–
		INTWDT	◉	×	×	–	–	–
		INT0	◉	◉	◉*1	○	○	○*1
		INT1 to 7	◉	×	×	×	×	×
		INTT0 to 7	◉	×	×	×	×	×
		INTTR8 to B	◉	×	×	×	×	×
		INTTO8, INTTOA	◉	×	×	×	×	×
		INTRX0 to 1, TX0 to	◉	×	×	×	×	×
		INTCR, INTCT, INTCG	◉	×	×	×	×	×
		INTSEM0, E0, R0, T0	◉	×	×	×	×	×
		INTSEM1, E1, R1, T1	◉	×	×	×	×	×
		INTSBE0, S0, E1, S1	◉	×	×	×	×	×
		INTAD	◉	×	×	×	×	×
	RESET		◉	◉	◉	◉	◉	◉

◉: After clearing the Halt mode, CPU starts interrupt processing. (RESET initializes the microcont.)

○: After clearing the Halt mode, CPU resumes executing starting from instruction following the HALT instruction.

×: Cannot be used to clear the Halt mode.

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

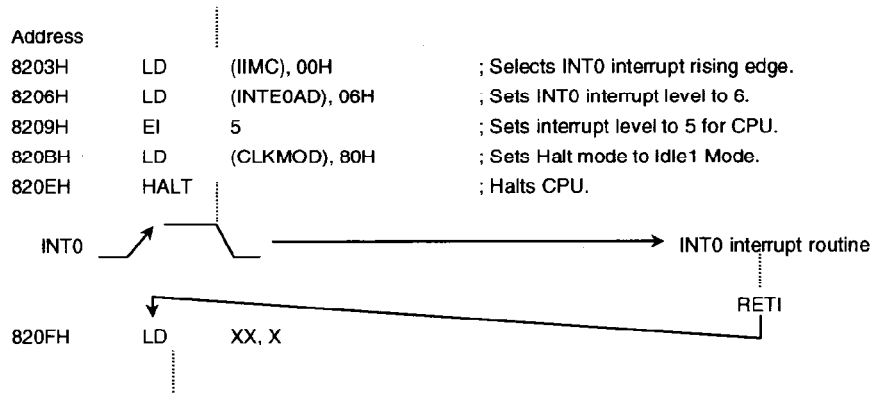
*1: The Halt mode is cleared when the warm-up time has elapsed.

Note 1: When the Halt mode is cleared by an 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 H, interrupt processing is not correctly started.

Note 2: If one of the external interrupts INT5~INT7 is generated in Idle2 Mode, TRUN8<I2T8> and TRUNA<I2TA> are set to 1

(Example - clearing Idle1 Mode)

An INT0 interrupt clears the Halt state when the device is in Idle1 Mode.



(3) Operation

① Idle2 Mode

In Idle2 Mode only specific internal I/O operations, as designated by the Idle2 Setting Register, can take place. Instruction execution by the CPU stops.

Figure 3.3.2 illustrates an example of the timing for clearance of the Idle2 Mode Halt state by an interrupt.

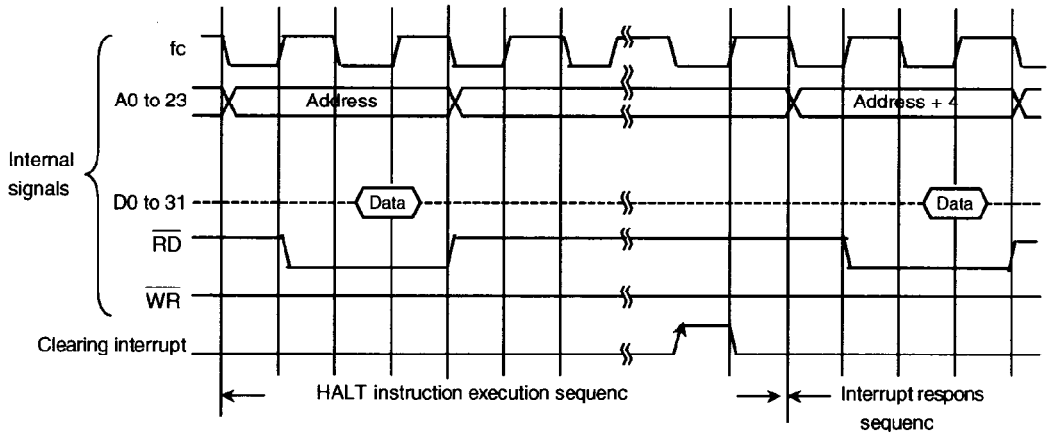


Figure 3.3.2 Timing chart for Idle2 Mode Halt state cleared by interrupt

② Idle1 Mode

In Idle1 Mode, only the internal oscillator continue to operate. The system clock in the MCU stops.

In the Halt state, the interrupt request is sampled asynchronously with the system clock; however, clearance of the Halt state (i.e. restart of operation) is synchronous with it.

Figure 3.3.3 illustrates the timing for clearance of the Idle1 Mode Halt state by an interrupt.

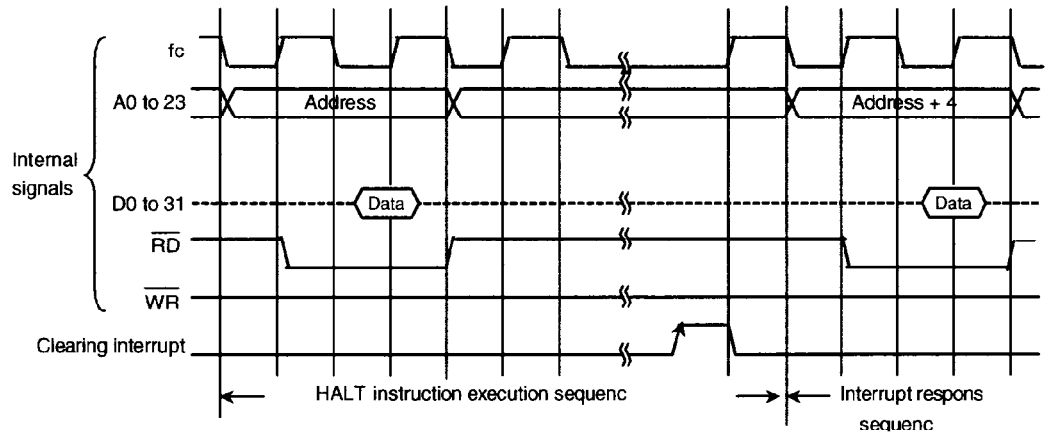


Figure 3.3.3 Timing chart for Idle1 Mode Halt state cleared by interrupt

③ Stop Mode

When Stop Mode is selected, all internal circuits stop, including the internal oscillator Pin status in Stop Mode depends on the settings in the WDMOD<DRVE> register. Table 3.3.5 summarizes the state of these pins in Stop Mode.

After Stop Mode has been cleared system clock output starts when the warm-up time and clock doubler stable time have elapsed, in order to allow oscillation and clock doubler to stabilize. Figure 3.3.4 illustrates the timing for clearance of the Stop Mode Halt state by an interrupt.

STOP mode can only be released by an NMI pin or INT0 pin interrupt, or by reset.

When STOP mode is released by other than reset, the system clock starts its output after the time set by the warm-up counter for the internal oscillation to stabilize. When using reset to release stop mode, input reset signals long enough for stable oscillation and clock doubler stable time.

In systems with an external oscillator, the warm-up counter also operates when STOP mode is released. Therefore, such systems also require a warm-up time between input of release signals and system clock output.

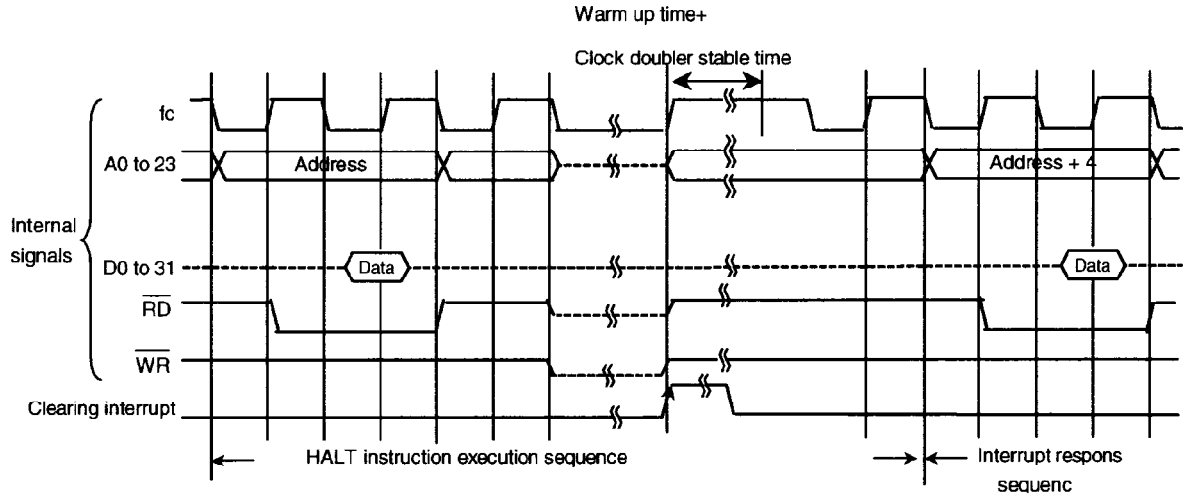


Figure 3.3.4 Timing chart for Stop Mode Halt state cleared by interrupt

Table 3.3.4 Sample warm-up time and clock doubler stable time after clearance of Stop Mode

fc=20MHz

	CLKMOD<WARM>	
	0	1
Warm-up tim	1.6 ms (2^{15}/fc)	6.6 ms (2^{17}/fc)
Clock doubler stable tim	1.6 ms (2^{15}/fc)	1.6 ms (2^{15}/fc)

Table 3.3.5 Pin states in Stop Mode

Pin Names	I/O	<DRVE> = 0	<DRVE> = 1
P00 to 07	Input Mode Output Mode D0~D7	Invalid Output High-z	Invalid Output High-z
P40 to 47/A0 to 7	Input Mode Output Mode	Invalid High-z	Invalid Output
P70 to 75/$\overline{RD}$ to /WAIT *Except P74/CLKOUT2	Input Mode Output Mode	Invalid High-z	Invalid Output
P74/CLKOUT2	Input Mode Output Mode CLKOUT2	Invalid High-z Output	Invalid Output Output
PC0 to PC5/TI0 to TO7	Input Mode Output Mode	Invalid High-z	Invalid Output
PD0 to PD7/TI8 to TOB	Input Mode Output Mode	Invalid High-z	Invalid Output
PF0 to PF7/TXD0 to RX	Input Mode Output Mode	Invalid High-z	Invalid Output
PG0 to PG7/AN0 to AN7	Input Mode	Invalid	Invalid
PL0 to PL3/AN8 to AN11	Input Mode	Invalid	Invalid
PM0 to PM7 /SS0 to SECLK1	Input Mode Output Mode	Invalid High-z	Invalid Output
PN0 to PN5 /SCK0 to SI1	Input Mode Output Mode	Invalid High-z	Invalid Output
$\overline{NMI}$	Input pin	Input	Input
$\overline{INT0}$	Input pin	Input	Input
$\overline{RESET}$	Input	Input	Input
AM0, AM1	Input	Input	Input
TEST0, TEST1	Input	Input	Input
X1	Input	Invalid	Invalid
X2	Output	H Level Output	H Level Output
CLKOUT1	Output	Output (PU) Output	Output (PU) @TMP94FD53F (Note) Output @TMP92CW10F

Input: Input gate in operation. Input voltage should be fixed to L or H so that input pin stays constant.

Output: Output state

Invalid: Input pin invalid

High-z: Output pin High-Impedance

PU: A pull-up is always carried out.

Note: CLKOUT1 pin may be in the state of "Lo " level output at Stop mode.
Please set CLKMOD <CLKOE> bit to "0" and be sure to forbid CLKOUT1 output, before the "HALT" instruction is executed.

3.4 Interrupts

Interrupts are controlled by the CPU Interrupt Mask Register <IFF20> (reflected by bits 12 to 14 of the Status Register) and by the built-in interrupt controller.

The TMP92CW10 has a total of 58 interrupts divided into the following five types:

Interrupts generated by CPU: 9 sources
- Software interrupts: 8 sources
- Illegal Instruction interrupt: 1 source

Internal interrupts: 43 sources
- Internal I/O interrupts: 32 sources
- Micro DMA Transfer End interrupts: 8 sources

External interrupts: 9 sources
- Interrupts on external pins ($\overline{NMI}$, INT0 to INT7)

A fixed individual interrupt vector number is assigned to each interrupt source.

Any one of seven levels of priority can also be assigned to each maskable interrupt. Non-maskable interrupts have a fixed priority level of 7, the highest level.

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

The CPU compares the interrupt priority level which it receives with the value held in the CPU Interrupt Mask Register <IFF2:0>. If the priority level of the interrupt is greater than or equal to the value in the Interrupt Mask Register, the CPU accepts the interrupt.

However, software interrupts and Illegal Instruction interrupts generated by the CPU are processed irrespective of the value in <IFF20>.

The value in the Interrupt Mask Register <IFF2 :0> can be changed using the EI instruction (EI num sets <IFF2:0> to num). For example, the command EI 3 enables the acceptance of non-maskable interrupts and of maskable interrupts whose priority level, as set in the interrupt controller, is 3 or higher. The commands EI and EI 0 enable the acceptance of all non-maskable interrupts and of maskable interrupts with a priority level of 1 or above (hence both are equivalent to the command EI 1).

The DI instruction (sets <IFF2:0> to 7) is exactly equivalent to the EI 7 instruction. The DI instruction is used to disable all maskable interrupts (since the priority level for maskable interrupts ranges from 0 to 6). The EI instruction takes effect as soon as it is executed.

In addition to the general-purpose Interrupt Processing Mode described above, there is also a Micro DMA Processing Mode.

In Micro DMA Mode the CPU automatically transfers data in one-byte, two-byte or four-byte blocks: this mode allows high-speed data transfer to and from internal and external memory and internal I/O ports.

In addition, the TMP94FD53 also has a software start function in which micro DMA processing is requested in software rather than by an interrupt.

Figure 3.4.1 is a flowchart showing overall interrupt processing.

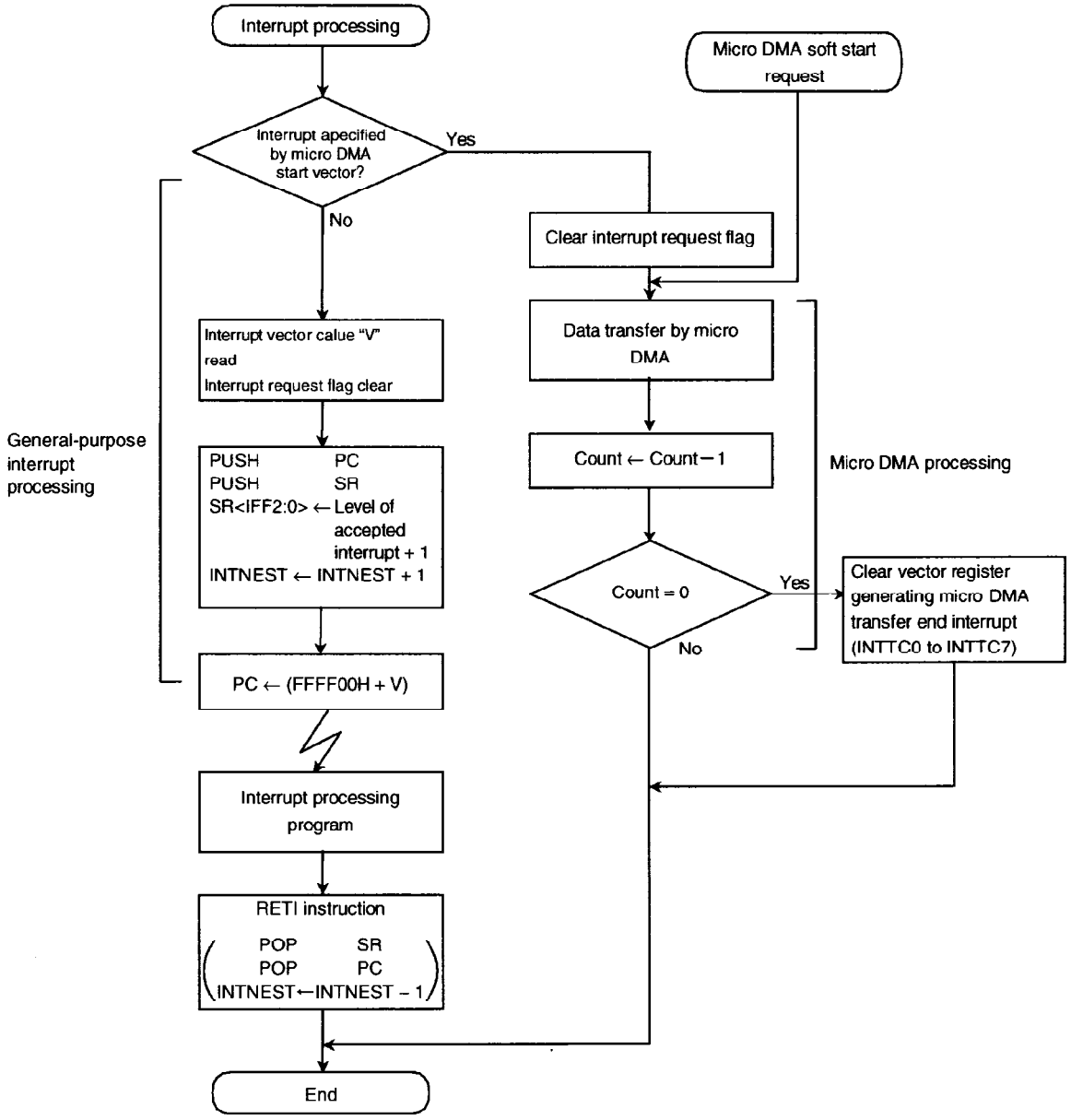


Figure 3.4.1 Interrupt and micro DMA processing sequence

3.4.1 General-purpose interrupt processing

When the CPU accepts an interrupt, it usually performs the following sequence of operations. However, in the case of software interrupts and Illegal Instruction interrupts generated by the CPU, the CPU skips steps ① and ③ and executes only steps ②, ④ and ⑤.

① The CPU reads the interrupt vector from the interrupt controller.
When more than one interrupt with the same priority level have been generated simultaneously, the interrupt controller generates an interrupt vector in accordance with the default priority and clears the interrupt requests.
(The default priority is determined as follows: the smaller the vector value, the higher the priority.)

② The CPU pushes the Program Counter (PC) and Status Register (SR) onto the top of the stack (pointed to by XSP).

③ The CPU sets the value of the CPUs Interrupt Mask Register <IFF2:0> to the priority level for the accepted interrupt plus 1. However, if the priority level for the accepted interrupt is 7, the register's value is set to 7.

④ The CPU increments the interrupt nesting counter INTNEST by 1.

⑤ The CPU jumps to the address given by adding the contents of address FFFF00H + the interrupt vector, then starts the interrupt processing routine.

On completion of interrupt processing, the RETI instruction is used to return control to the main routine. RETI restores the contents of the Program Counter and the Status Register from the stack and decrements the Interrupt Nesting counter INTNEST by 1.
Non-maskable interrupts cannot be disabled by a user program. Maskable interrupts, however, can be enabled or disabled by a user program. A program can set the priority level for each interrupt source. (A priority level setting of 0 or 7 will disable an interrupt request.)

If an interrupt request is received for an interrupt with a priority level equal to or greater than the value set in the CPU Interrupt Mask Register <IFF2:0>, the CPU will accept the interrupt. The CPU Interrupt Mask Register <IFF2:0> is then set to the value of the priority level for the accepted interrupt plus 1.

If during interrupt processing, an interrupt is generated with a higher priority than the interrupt currently being processed, or if during the processing of a non-maskable interrupt processing, a non-maskable interrupt request is generated from another source, the CPU will suspend the routine which it is currently executing and accept the new interrupt. When processing of the new interrupt has been completed, the CPU will resume processing of the suspended interrupt.

If the CPU receives another interrupt request while performing processing steps ① to ⑤, the new interrupt will be sampled immediately after execution of the first instruction of its interrupt processing routine. Specifying DI as the start instruction disables nesting of maskable interrupts.

After a reset, initializes the Interrupt Mask Register <IFF2:0> to 111, disabling all maskable interrupts.

Table 3.4.1 shows the TMP92CW10 interrupt vectors and micro DMA start vectors. FFFF00H~FFFFFFH (256 bytes) is designated as the interrupt vector area.

Table 3.4.1 TMP92CW10 interrupt vectors and micro DMA start vectors

Default Priority	Type	Interrupt Source and Source of Micro DMA Request	Vector Value	Address refer to Vector	Micro DMA Start Vector
1	Non maskable	Reset or [SWI0] instruction	0000H	FFFF00H	
2		[SWI1] instruction	0004H	FFFF04H	
3		Illegal instruction or [SWI2] instruction	0008H	FFFF08H	
4		[SWI3] instruction	000CH	FFFF0CH	
5		[SWI4] instruction	0010H	FFFF10H	
6		[SWI5] instruction	0014H	FFFF14H	
7		[SWI6] instruction	0018H	FFFF18H	
8		[SWI7] instruction	001CH	FFFF1CH	
9		NMI: pin input	0020H	FFFF20H	
10		INTWD: Watchdog Timer	0024H	FFFF24H	
11	Maskable	INT0: INT0 pin input	0028H	FFFF28H	0AH (Note1)
12		INT1: INT1 pin input	002CH	FFFF2CH	0BH
13		INT2: INT2 pin input	0030H	FFFF30H	0CH
14		INT3: INT3 pin input	0034H	FFFF34H	0DH
15		INT4: INT4 pin input	0038H	FFFF38H	0EH
16		INT5: INT5 pin input	003CH	FFFF3CH	0FH
17		INT6: INT6 pin input	0040H	FFFF40H	10H
18		INT7: INT7 pin input	0044H	FFFF44H	11H
19		INTT0: 8-bit timer 0	0048H	FFFF48H	12H
20		INTT1: 8-bit timer 1	004CH	FFFF4CH	13H
21		INTT2: 8-bit timer 2	0050H	FFFF50H	14H
22		INTT3: 8-bit timer 3	0054H	FFFF54H	15H
23		INTT4: 8-bit timer 4	0058H	FFFF58H	16H
24		INTT5: 8-bit timer 5	005CH	FFFF5CH	17H
25		INTT6: 8-bit timer 6	0060H	FFFF60H	18H
26		INTT7: 8-bit timer 7	0064H	FFFF64H	19H
27		INTTR8: 16-bit timer 8	0068H	FFFF68H	1AH
28		INTTR9: 16-bit timer 8	006CH	FFFF6CH	1BH
29		INTTRA: 16-bit timer A	0070H	FFFF70H	1CH
30		INTTRB: 16-bit timer A	0074H	FFFF74H	1DH
31		INTTO8: 16-bit timer 8 (overflow)	0078H	FFFF78H	1EH
32		INTTOA: 16-bit timer A (overflow)	007CH	FFFF7CH	1FH
33		INTRX0: Serial receive (Channel 0)	0080H	FFFF80H	-
34		INTTX0: Serial transmission (Channel 0)	0084H	FFFF84H	21H
35		INTRX1: Serial receive (Channel 1)	0088H	FFFF88H	-
36		INTTX1: Serial transmission (Channel 1)	008CH	FFFF8CH	23H
37		(Reserved)	0090H	FFFF90H	-
38		(Reserved)	0094H	FFFF94H	-
39		(Reserved)	0098H	FFFF98H	-
40		INTSEM0: SEI mode (Channel 0)	009CH	FFFF9CH	-
41		INTSEE0: SE transfer end / slave error (Channel 0)	00A0H	FFFFA0H	-
42		INTSER0: SEI receive (Channel 0)	00A4H	FFFFA4H	29H
43		INTSET0: SEI transmission (Channel 0)	00A8H	FFFFA8H	2AH
44		INTSEM1: SEI mode (Channel 1)	00ACH	FFFFACH	-
45		INTSEE1: SE transfer end / slave error (Channel 1)	00B0H	FFFFB0H	-
46		INTSER1: SE receive (Channel 1)	00B4H	FFFFB4H	2DH
47		INTSET1: SEI transmission (Channel 1)	00B8H	FFFFB8H	2EH
48		INTSBE0: SB I2CBUS transfer end (Channel 0)	00BCH	FFFFBCH	2FH
49		INTSBS0: SBI I2CBUS stop condition (Channel 0)	00C0H	FFFFC0H	30H
50		INTSBE1: SBI I2CBUS transfer end (Channel 1)	00C4H	FFFFC4H	31H

51	Maskable	INTSBS1: SBI I2CBUS stop condition (Channel 1)	00C8H	FFFFC8H	32H
52		INTAD: AD conversion end	00CCH	FFFFCCH	33H
53		INTTC0: Micro DMA end (Channel 0)	00D0H	FFFFD0H	34H
54		INTTC1: Micro DMA end (Channel 1)	00D4H	FFFFD4H	35H
55		INTTC2: Micro DMA end (Channel 2)	00D8H	FFFFD8H	36H
56		INTTC3: Micro DMA end (Channel 3)	00DCH	FFFFDCH	37H
57		INTTC4: Micro DMA end (Channel 4)	00E0H	FFFFE0H	38H
58		INTTC5: Micro DMA end (Channel 5)	00E4H	FFFFE4H	39H
59		INTTC6: Micro DMA end (Channel 6)	00E8H	FFFFE8H	3AH
60		INTTC7: Micro DMA end (Channel 7)	00ECH	FFFFECH	3BH
- to -		(reserved)	00F0H : 00FCH	FFFFF0H : FFFFFCH	- to -

Note1 : When starting-up micro DMA, set at edge detect mode.

Note2: Micro DMA default priority

If an interrupt request is generated by micro DMA, the interrupt has a higher priority than any other maskable interrupt (irrespective of default channel priority).

3.4.2 Micro DMA processing

In addition to general-purpose interrupt processing, the TMP92CW10 also includes a micro DMA function. Micro DMA processing for interrupt requests set by micro DMA is performed at the highest priority level for maskable interrupts (level 6), regardless of the priority level of the interrupt source.

Micro DMA is suppoted 8 channels and can be transferred continuously by specifying the micro DMA burst function in the following.

(1) Micro DMA operation

When an interrupt request is generated by an interrupt source specified by the Micro DMA Start Vector Register, the micro DMA triggers a micro DMA request to the CPU at interrupt priority level 6 and starts processing the request. Theeight micro DMA channels allow micro DMA processing to be set for up toeight types of interrupt at once.

When micro DMA is accepted, the interrupt request flip-flop assigned to that channel is cleared. Data in one-byte or two-byte or four-byte blocks, is automatically transferred at once from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decremented by 1. If the value of the counter after it has been decremented is not 0, DMA processing ends with no change in the value of the micro DMA start vector register. If the value of the decremented counter is 0, a Micro DMA Transfer End interrupt (INTTC0 to INTTC7) is sent from the CPU to the interrupt controller. In addition, the micro DMA start vector register is cleared to 0, the next micro DMA operation is disabled and micro DMA processing terminates.

If micro DMA requests are set simultaneously for more than one channel, priority is not based on the interrupt priority level but on the channel number: the lower the channel number, the higher the priority (Channel 0 thus has the highest priority and Channel 7 the lowest).

If an interrupt request is triggered for the interrupt source in use during the interval between the time at which the micro DMA start vector is cleared and the next setting, general-purpose interrupt processing is performed at the interrupt level set. Therefore, if the interrupt is only being used to initiate micro DMA (and not as a general-purpose interrupt), the interrupt level should first be set to 0 (i.e. interrupt requests should be disabled).

If micro DMA and general-purpose interrupts are being used together as described above, the level of the interrupt which is being used to initiate micro DMA processing should first be set to a lower value than all the other interrupt levels. In this case, edge-triggered interrupts are the only kinds of general interrupts which can be accepted.

Although the control registers used for setting the transfer source and transfer destination addresses are 32 bits wide, this type of register can only output 24-bit addresses. Accordingly, micro DMA can only access 16 Mbytes (the upper eight bits of a 32-bit address are not valid).

Three micro DMA transfer modes are supported: one-byte transfers, two-byte (one-word) transfer and four-byte transfer. After a transfer in any mode, the transfer source and transfer destination addresses will either be incremented or decremented, or will remain unchanged. This simplifies the transfer of data from I/O to memory, from memory to I/O, and from I/O to I/O. For details of the various transfer modes, see Section 34.2 (1), Detailed description of the Transfer Mode Register.

Since a transfer counter is a 16-bit counter, up to 65536 micro DMA processing operations can be performed per interrupt source (provided that the transfer counter for the source is initially set to 0000H).

Micro DMA processing can be initiated by any one of 42 different interrupts – the 41 interrupts shown in the micro DMA start vectors in Table 34.1 and a micro DMA soft start. Figure 3.4.2 shows a 2-byte transfer carried out using a micro DMA cycle in Transfer Destination Address INC Mode (micro DMA transfers are the same in every mode except Counter Mode). (The conditions for this cycle are as follows: external 8-bit bus, 0 waits, and even-numbered transfer source and transfer destination addresses).

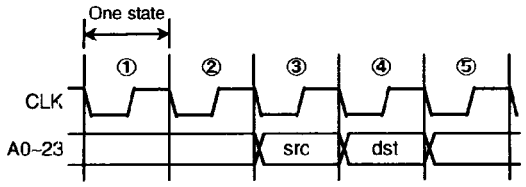


Figure 3.4.2 Timing for micro DMA cycle

States 1, 2: **Instruction fetch cycle** (prefetches the next instruction code)

State 3: Micro DMA read cycle

State 4: Micro DMA write cycle

State 5: (The same as in state 1, 2)

(2) Soft start function

The TMP92CW10 can initiate micro DMA either with an interrupt or by using the micro DMA soft start function, in which micro DMA is initiated by a Write cycle which writes to the register DMAR.

Writing 1 to any bit of the register DMAR causes micro DMA to be performed once. On completion of the transfer, the bits of DMAR which support the end channel are automatically cleared to 0.

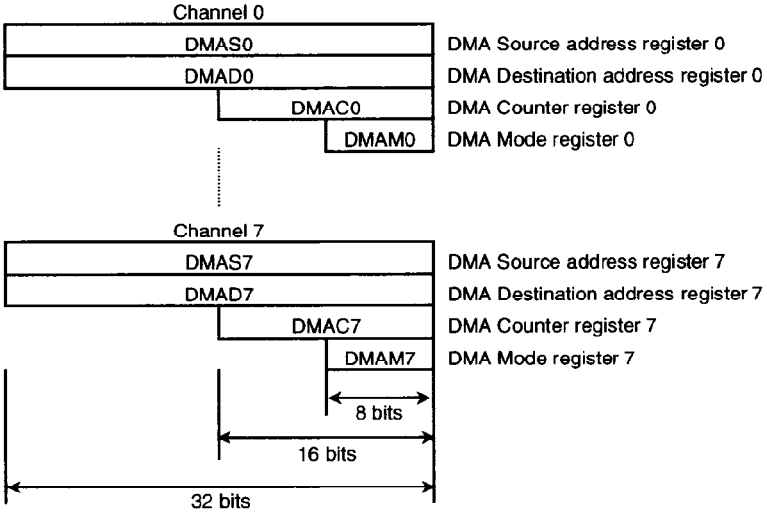
When a burst is specified by the register DMAB, data is transferred continuously from the initiation of micro DMA until the value in the micro DMA transfer counter is 0.

Symbol	NAME	Address	7	6	5	4	3	2	1	0
DMAR	DMA Request	109h	DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
			R/W							
			0	0	0	0	0	0	0	0

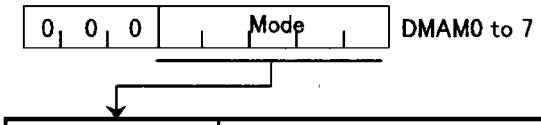
DMAR must not be used any of the read-modify-write instructions.

(3) Transfer control registers

The transfer source address and the transfer destination address are set in the following registers. An instruction of the form LDC cr,r can be used to set these registers.



(4) Detailed description of the Transfer Mode Register



DMAM[4:0]	Mode Description	Execution time
0 0 0 z z	Destination INC mode (DMADn +) ← (DMASn) DMACn ← DMACn − 1 if DMACn = 0 then INTTCn	5states
0 0 1 z z	Destination DEC mode (DMADn −) ← (DMASn) DMACn ← DMACn − 1 if DMACn = 0 then INTTCn	5states
0 1 0 z z	Source INC mode (DMADn) ← (DMASn +) DMACn ← DMACn − 1 if DMACn = 0 then INTTCn	5states
0 1 1 z z	Source DEC mode (DMADn) ← (DMASn −) DMACn ← DMACn − 1 if DMACn = 0 then INTTCn	5states
1 0 0 z z	Source and Destination INC mode (DMADn +) ← (DMASn +) DMACn ← DMACn − 1 If DMACn = 0 then INTTCn	6states
1 0 1 z z	Source and Destination DEC mode (DMADn −) ← (DMASn −) DMACn ← DMACn − 1 If DMACn = 0 then INTTCn	6states
1 1 0 z z	Destination and Fixed mode (DMADn) ← (DMASn) DMACn ← DMACn − 1 If DMACn = 0 then INTTCn	5states
1 1 1 z z	Counter mode DMASn ← DMASn + 1 DMACn ← DMACn − 1 if DMACn = 0 then INTTCn	5states

ZZ: 00 = 1-byte transfer
01 = 2-byte transfer
10 = 4-byte transfer
11 = (reserved)

Note : The execution time is measured at 1states = 50ns (operation @internal 20 MHz)

Note: n stands for the micro DMA channel number (0 to 7)
DMADn+/DMASn+: Post-increment (register value is incremented after transfer)
DMADn−/DMASn−: Post-decrement (register value is decremented after transfer)

3.4.3 Interrupt controller operation

The block diagram in Figure 3.4.3 shows the interrupt circuits. The left-hand side of the diagram shows the interrupt controller circuit. The right-hand side shows the CPU interrupt request signal circuit and the halt release circuit.

For each of the 52 interrupt channels there is an interrupt request flag (consisting of a flip-flop), an interrupt priority setting register and a micro DMA start vector register. The interrupt request flag latches interrupt requests from the peripherals. The flag is cleared to zero in the following cases: when a Reset occurs, when the CPU reads the channel vector of an interrupt it has received, when the CPU receives a micro DMA request (when micro DMA is set), when a micro DMA burst transfer is terminated, and when an instruction that clears the interrupt for that channel is executed (by writting a micro DMA start vector to the INTCLR register).

An interrupt priority can be set independently for each interrupt source by writing the priority to the interrupt priority setting register (e.g. INTE0AD or INTE12). Six interrupt priorities levels (1 to 6) are provided. Setting an interrupt source's priority level to 0 (or 7) disables interrupt requests from that source. The priority of non-maskable interrupts (NMI pin interrupts and Watchdog Timer interrupts) is fixed at 7. If more than one interrupt request with a given priority level are generated simultaneously, the default priority (the interrupt with the lowest priority or, in other words, the interrupt with the lowest vector value) is used to determine which interrupt request is accepted first.

The 3rd and 7th bits of the interrupt priority setting register indicate the state of the interrupt request flag and thus whether an interrupt request for a given channel has occurred.

If several interrupts are generated simultaneously, the interrupt controller sends the interrupt request for the interrupt with the highest priority and the interrupt's vector address to the CPU. The CPU compares the mask value set in <IFF2:0> of the Status Register (SR) with the priority level of the requested interrupt; if the latter is higher, the interrupt is accepted. Then the CPU sets SR <IFF20> to the priority level of the accepted interrupt + 1. Hence, during processing of the accepted interrupt, new interrupt requests with a priority value equal to or higher than the value set in SR <IFF2 :0> (i.e. interrupts with a priority higher than the interrupt being processed) will be accepted.
When interrupt processing has been completed (i.e. after execution of a RETI instruction), the CPU restores to SR<IFF20> the priority value which was saved on the stack before the interrupt was generated.
The interrupt controller also includes eight registers which are used to store the micro DMA start vector. 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 micro DMA processing.

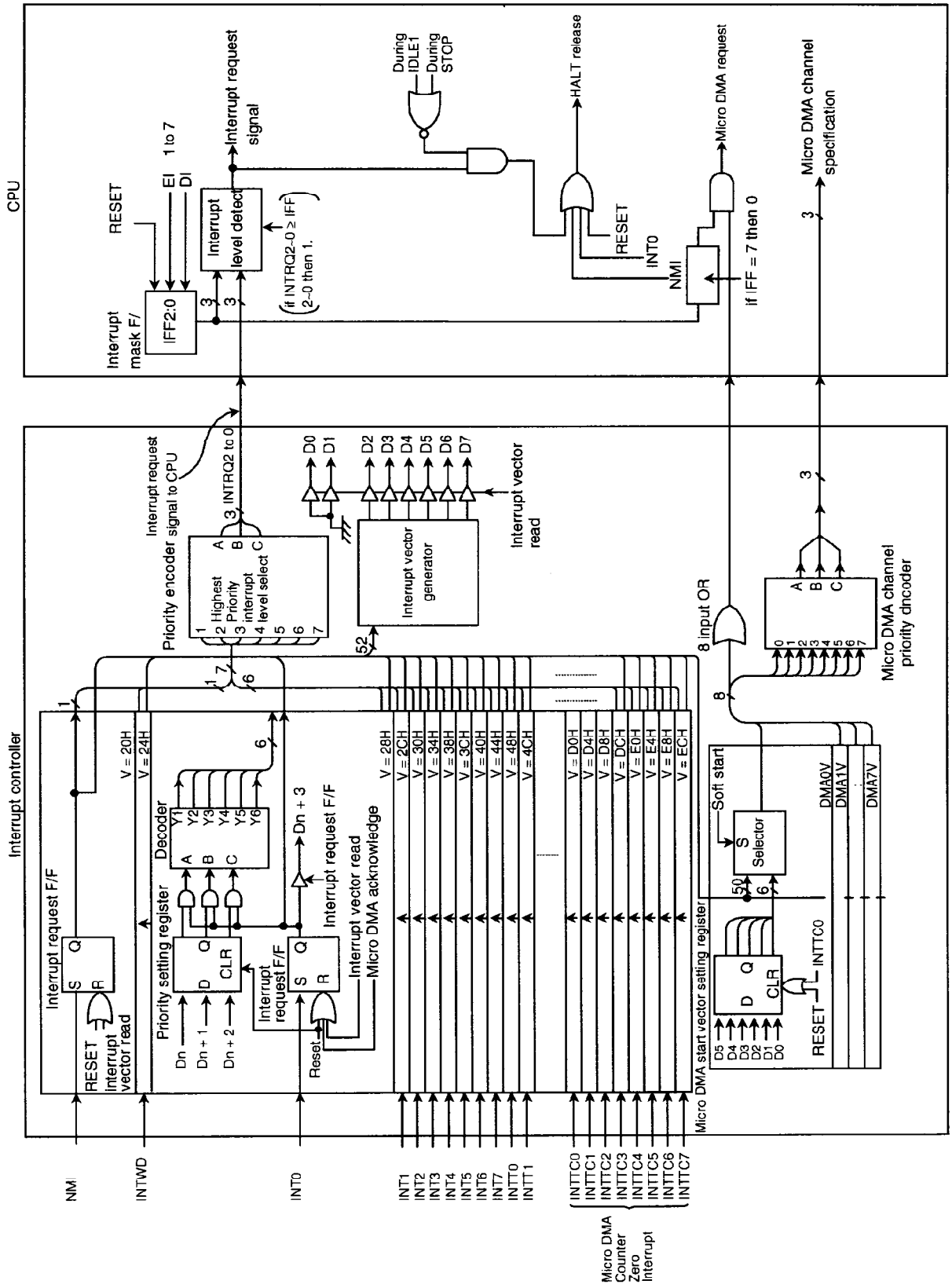


Figure 3.4.3 Block Diagram of Interrupt Controller

(1) Interrupt priority setting registers

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 & INTAD Enable	F0h	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE12	INT1 & INT2 Enable	D0h	INT2				INT1			
			I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE34	INT3 & INT4 Enable	D1h	INT4				INT3			
			I4C	I4M2	I4M1	I4M0	I3C	I3M2	I3M1	I3M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE56	INT5 & INT6 Enable	D2h	INT6				INT5			
			I6C	I6M2	I6M1	I6M0	I5C	I5M2	I5M1	I5M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE7	INT7 Enable	D3h					INT7			
			-	-	-	-	I7C	I7M2	I7M1	I7M0
							R	R/W		
			-	-	-	-	0	0	0	0
INTET01	INTT0 & INTT1 Enable	D4h	INTT1(Timer1)				INTT0(Timer0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET23	INTT2 & INTT3 Enable	D5h	INTT3(Timer3)				INTT2(Timer2)			
			IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET45	INTT4 & INTT5 Enable	D6h	INTT5(Timer5)				INTT4(Timer4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET67	INTT6 & INTT7 Enable	D7h	INTT7(Timer7)				INTT6(Timer6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET89	INTTR8 & INTTR9 Enable	D8h	INTTR9(Timer8)				INTTR8(Timer8)			
			IT9C	IT9M2	IT9M1	IT9M0	IT8C	IT8M2	IT8M1	IT8M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETAB	INTTRA & INTTRB Enable	D9h	INTTRB(TimerA)				INTTRA(TimerA)			
			ITBC	ITBM2	ITBM1	ITBM0	ITAC	ITAM2	ITAM1	ITAM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETO8A	INTTO8 & INTTOA (Overflow) Enable	DAh	INTTOA				INTTO8			
			ITOAC	ITOAM2	ITOAM1	ITOAM0	ITO8C	ITO8M2	ITO8M1	ITO8M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTES0	INTRX0 & INTTX0 Enable	DBh	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES1	INTRX1 & INTTX1 Enable	DCh	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
(Reserved)		DDh								
			–	–	–	–	–	–	–	–
			–	–	–	–	–	–	–	–
(Reserved)		DEh								
			–				–	–	–	–
				–	–	–				
			–	–	–	–	–	–	–	–
INTESEE0	INTSEM0 & INTSEE0 Enable	DFh	INTSEE0				INTSEM0			
			ISEE0C	ISEE0M2	ISEE0M1	ISEE0M0	ISEM0C	ISEM0M2	ISEM0M1	ISEM0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESED0	INTSER0 & INTSET0 Enable	E0h	INTSET0				INTSER0			
			ISET0C	ISET0M2	ISET0M1	ISET0M0	ISER0C	ISER0M2	ISER0M1	ISER0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESEE1	INTSEM1 & INTSEE1 Enable	E1h	INTSEE1				INTSEM1			
			ISEE1C	ISEE1M2	ISEE1M1	ISEE1M0	ISEM1C	ISEM1M2	ISEM1M1	ISEM1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESED1	INTSER1 & INTSET1 Enable	E2h	INTSET1				INTSER1			
			ISET1C	ISET1M2	ISET1M1	ISET1M0	ISER1C	ISER1M2	ISER1M1	ISER1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESB0	INTSBE0 & INTSBS0 Enable	E3h	INTSBS0				INTSBE0			
			ISBS0C	ISBS0M2	ISBS0M1	ISBS0M0	ISBE0C	ISBE0M2	ISBE0M1	ISBE0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESB1	INTSBE1 & INTSBS1 Enable	E4h	INTSBS1				INTSBE1			
			ISBS1C	ISBS1M2	ISBS1M1	ISBS1M0	ISBE1C	ISBE1M2	ISBE1M1	ISBE1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC01	INTTC0 & INTTC1 Enable	F1h	INTTC1(DMA1)				INTTC0(DMA0)			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC23	INTTC2 & INTTC3 Enable	F2h	INTTC3(DMA3)				INTTC2(DMA2)			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTETC45	INTTC4 & INTTC5 Enable	F3h	INTTC5(DMA5)				INTTC4(DMA4)			
			ITC5C	ITC5M2	ITC5M1	ITC5M0	ITC4C	ITC4M2	ITC4M1	ITC4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC67	INTTC6 & INTTC7 Enable	F4h	INTTC7(DMA7)				INTTC6(DMA6)			
			ITC7C	ITC7M2	ITC7M1	ITC7M0	ITC6C	ITC6M2	ITC6M1	ITC6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTNMWDT	NMI & INTWD Enable	F7h	NMI				INTWD			
			ITCNM	-	-	-	ITCWD	-	-	-
			R				R			
			0	-	-	-	0	-	-	-

Interrupt request flag

IxxM2	IxxM1	IxxM0	Function (write)
0	0	0	Disables interrupt requests
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt requests

(2) External interrupt control

Symbol	NAME	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt Input Mode Control	F6H	–	–	–	–	–	–	I0LE	NMIREE
									R/W	
			–	–	–	–	–	–	0	0
									0:INT0 edge mode 1:INT0 level mode	$\overline{\text{NMI}}$ 0:Falling edge 1:Falling and rising edges

IIMC must not be used any of the read-modify-write instructions.

Note:

*INT0 Level Enable

0	Rising edge detect INT
1	"H" level INT

*$\overline{\text{NMI}}$ rising edge Enable

0	INTrequest generation at falling edge
1	INT request generation at rising/falling edge

Note 1 : Disable INT0 request before changing INT0 pin mode from level-sense to edge-sense.

Setting example:

```
DI
LD      (IIMC), XXXXXX0-B   ; Switches from level to edge.
LD      (INTCLR), 0AH       ; Clears interrupt request flag.
EI
```

Note: X = Don't care; "-" = No change.

Note 2 : See electrical characteristics in section 4 for external interrupt input pulse width.

Settings of External interrupt Pin Function

Interrupt	Pin name	Mode		Setting method
$\overline{\text{NMI}}$	–		Falling Edge	IIMC<NMIREE> = 0
			Rising and Falling Edges	IIMC<NMIREE> = 1
INT0	INT0		Rising Edge	IIMC<I0LE> = 0
			High Level	IIMC<I0LE> = 1
INT1	PC0		Rising Edge	–
INT2	PC2		Rising Edge	–
INT3	PC3		Rising Edge	–
INT4	PC5		Rising Edge	–
INT5	PD0		Rising Edge	TMOD8<CAP89M1:0> = 0,0 or 0,1 or 1,1
			Falling Edge	TMOD8<CAP89M1:0> = 1,0
INT6	PD1		Rising Edge	–
INT7	PD4		Rising Edge	TMODA<CAPABM1:0> = 0,0 or 0,1 or 1,1
			Falling Edge	TMODA<CAPABM1:0> = 1,0

(3) Interrupt request flag clear register

The interrupt request flag is cleared by writing the appropriate micro DMA start vector, as given in Table 3.4 (1), to the register INTCLR.
For example, to clear the interrupt flag INT0, perform the following register operation after execution of the DI instruction.

INTCLR ← 0AH Clears interrupt request flag INT0.

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt Clear control	F8H	–	–	–	–	–	–	–	–
			W							
			0	0	0	0	0	0	0	0
			Interrupt Vector							

INTCLR must not be used any of the read-modify-write instructions.

(4) Micro DMA start vector registers

These registers assign micro DMA processing to an sets which source corresponds to DMA. The interrupt source whosemicro DMA start vector value matches the vector set in one of these registers is designated as the micro DMA start source.

When the micro DMA transfer counter value reaches zero, the micro DMA transfer end interrupt corresponding to the channel is sent to the interrupt controller, the micro DMA start vector register is cleared, and the micro DMA start source for the channel is cleared. Therefore, in order for micro DMA processing to continue, the micro DMA start vector register must be set again during processing of the micro DMA transfer end interrupt.

If the same vector is set in the micro DMA start vector registers of more than one channel, the lowest numbered channel takes priority.

Accordingly, if the same vector is set in the micro DMA start vector registers for two different channels, the interrupt generated on the lower-numbered channel is executed until micro DMA transfer is complete. If the micro DMA start vector for this channel has not been set in the channels micro DMA start vector register again, micro DMA transfer for the higher-numbered channel will be commenced. (This process is known as micro DMA chaining.)

Symbol	NAME	Address	7	6	5	4	3	2	1	0
DMA0V	DMA0 Start Vector	100h			DMA0 Start Vector					
			–	–	DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
			–	–	0	0	0	0	0	0
DMA1V	DMA1 Start Vector	101h			DMA1 Start Vector					
			–	–	DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
			–	–	0	0	0	0	0	0
DMA2V	DMA2 Start Vector	102h			DMA2 Start Vector					
			–	–	DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
			–	–	0	0	0	0	0	0
DMA3V	DMA3 Start Vector	103h			DMA3 Start Vector					
			–	–	DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
			–	–	0	0	0	0	0	0
DMA4V	DMA4 Start Vector	104h			DMA4 Start Vector					
			–	–	DMA4V5	DMA4V4	DMA4V3	DMA4V2	DMA4V1	DMA4V0
					R/W					
			–	–	0	0	0	0	0	0
DMA5V	DMA5 Start Vector	105h			DMA5 Start Vector					
			–	–	DMA5V5	DMA5V4	DMA5V3	DMA5V2	DMA5V1	DMA5V0
					R/W					
			–	–	0	0	0	0	0	0
DMA6V	DMA6 Start Vector	106h			DMA6 Start Vector					
			–	–	DMA6V5	DMA6V4	DMA6V3	DMA6V2	DMA6V1	DMA6V0
					R/W					
			–	–	0	0	0	0	0	0
DMA7V	DMA7 Start Vector	107h			DMA7 Start Vector					
			–	–	DMA7V5	DMA7V4	DMA7V3	DMA7V2	DMA7V1	DMA7V0
					R/W					
			–	–	0	0	0	0	0	0

(5) Specification of a micro DMA burst

Specifying the micro DMA burst function causes micro DMA transfer, once started, to continue until the value in the Transfer Counter Register reaches zero. Setting any of the bits in the register DMAB which correspond to a micro DMA channel (as shown below) to 1 specifies that any micro DMA transfer on that channel will be a burst transfer.

Symbol	NAME	Address	7	6	5	4	3	2	1	0
DMAB	DMA Burst	108h	DBST7	DBST6	DBST5	DBST4	DBST3	DBST2	DBST1	DBST0
			R/W							
			0	0	0	0	0	0	0	0

(6) Notes

The instruction execution unit and the bus interface unit in this CPU operate independently. Therefore if, immediately before an interrupt is generated, the CPU fetches an instruction which clears the corresponding interrupt request flag, the CPU may execute this instruction in between accepting the interrupt and reading the interrupt vector. In this case, the CPU will read the default vector 0004H and jump to interrupt vector address FFFF04H.

To avoid this, an instruction which clears an interrupt request flag should always be preceded by a DI instruction.

In addition, please note that the following two circuits are exceptional and demand special attention.

INT0 Level Mode	In Level Mode INT0 is not an edge-triggered interrupt. Hence, in Level Mode the interrupt request flag for INT0 does not function. The peripheral interrupt request passes through the S input of the flip-flop and becomes the Q output. If the interrupt input mode is changed from Edge Mode to Level Mode, the interrupt request flag is cleared automatically.
	If the CPU enters the interrupt response sequence as a result of INT0 going from 0 to 1, INT0 must then be held at 1 until the interrupt response sequence has been completed. If INT0 is set to Level Mod so as to release a Halt state, INT0 must be held at 1 from the time INT0 changes from 0 to 1 until the Halt state is released. (Hence, it is necessary to ensure that input noise is not interpreted as a 0, causing INT0 to revert to 0 before the Halt state has been released.)
	When the mode changes from Level Mode to Edge Mode, interrupt request flags which were set in Level Mode will not be cleared. Interrupt request flags must be cleared using the following sequence.
	DI
	LD (IIMC), 00H; Switches from level to edge.
	LD (INTCLR), 0AH; Clears interrupt request flag.
	EI

INTRX	The interrupt request flip-flop can only be cleared by a Reset or by reading the Serial Channel Receive Buffer. It cannot be cleared by an instruction.

Note: The following instructions or pin input state changes are equivalent to instructions which clear the interrupt request flag.

INT0: Instructions which switch to Level Mode after an interrupt request has been generated in Edge Mode.
The pin input changes from High to Low after an interrupt request has been generated in Level Mode. ("H" → "L")

INTRX Instructions which read the Receive Buffer

3.5 Function of Ports

TMP92CW10 has I/O port pins that are shown in table 3.5.1 In addition to functioning as general-purpose I/O ports, these pins are also used by internal CPU and I/O functions.

Table 3.5.1 Port Functions (1/2)

Port Name	Pin Name	Number of Pins	I/O	I/O Setting	Pin Name for built-in function
Port 0	P00 to P07	8	I/O	Bit	D0 to D7
Port 4	P40 to P47	8	I/O	Bit	A0 to A7
Port 7	P70	1	I/O	Bit	$\overline{RD}$
	P71	1	I/O	Bit	$\overline{WR}$
	P72	1	I/O	Bit	
	P73	1	I/O	Bit	$\overline{CS}$
	P74	1	I/O	Bit	CLKOUT2
	P75	1	I/O	Bit	$\overline{WAIT}$
Port C	PC0	1	I/O	Bit	TI0/INT1
	PC1	1	I/O	Bit	TO1
	PC2	1	I/O	Bit	TO3/INT2
	PC3	1	I/O	Bit	TI4/INT3
	PC4	1	I/O	Bit	TO5
	PC5	1	I/O	Bit	TO7/INT4
Port D	PD0	1	I/O	Bit	TI8/INT5/A8
	PD1	1	I/O	Bit	TI9/INT6/A9
	PD2	1	I/O	Bit	TO8/A10
	PD3	1	I/O	Bit	TO9/A11
	PD4	1	I/O	Bit	TIA/INT7/A12
	PD5	1	I/O	Bit	TIB/A13
	PD6	1	I/O	Bit	TOA/A14
	PD7	1	I/O	Bit	TOB/A15
Port F	PF0	1	I/O	Bit	TXD0
	PF1	1	I/O	Bit	RXD0
	PF2	1	I/O	Bit	SCLK0, $\overline{CTS0}$
	PF3	1	I/O	Bit	TXD1
	PF4	1	I/O	Bit	RXD1
	PF5	1	I/O	Bit	SCLK1, $\overline{CTS1}$
	PF6	1	I/O	Bit	
	PF7	1	I/O	Bit	
Port G	PG0 to PG7	8	Input	(Fixed)	AN0 to AN7
Port L	PL0 to PL3	4	Input	(Fixed)	AN8 to AN11
Port M	PM0	1	I/O	Bit	$\overline{SS0}$/A16
	PM1	1	I/O	Bit	MOSI0/A17
	PM2	1	I/O	Bit	MISO0/A18
	PM3	1	I/O	Bit	SECLK0/A19
	PM4	1	I/O	Bit	$\overline{SS1}$/A20
	PM5	1	I/O	Bit	MOSI1/A21
	PM6	1	I/O	Bit	MISO1/A22
	PM7	1	I/O	Bit	SECLK1/A23

Table 3.5.1 Port Functions (2/2)

Port Name	Pin Name	Number of Pins	I/O	I/O Setting	Pin Name for built-in function
Port N	PN0	1	I/O	Bit	SCK0
	PN1	1	I/O	Bit	SO0/SDA0
	PN2	1	I/O	Bit	SCL0
	PN3	1	I/O	Bit	SCK1
	PN4	1	I/O	Bit	SO1/SDA1
	PN5	1	I/O	Bit	SCL1

3.5.1 Port 0 (P00 to P07)

Port0 is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P0CR and function register P0FC.

In addition to functioning as a general-purpose I/O port. port0 can also function as a data bus (D0 to D7).

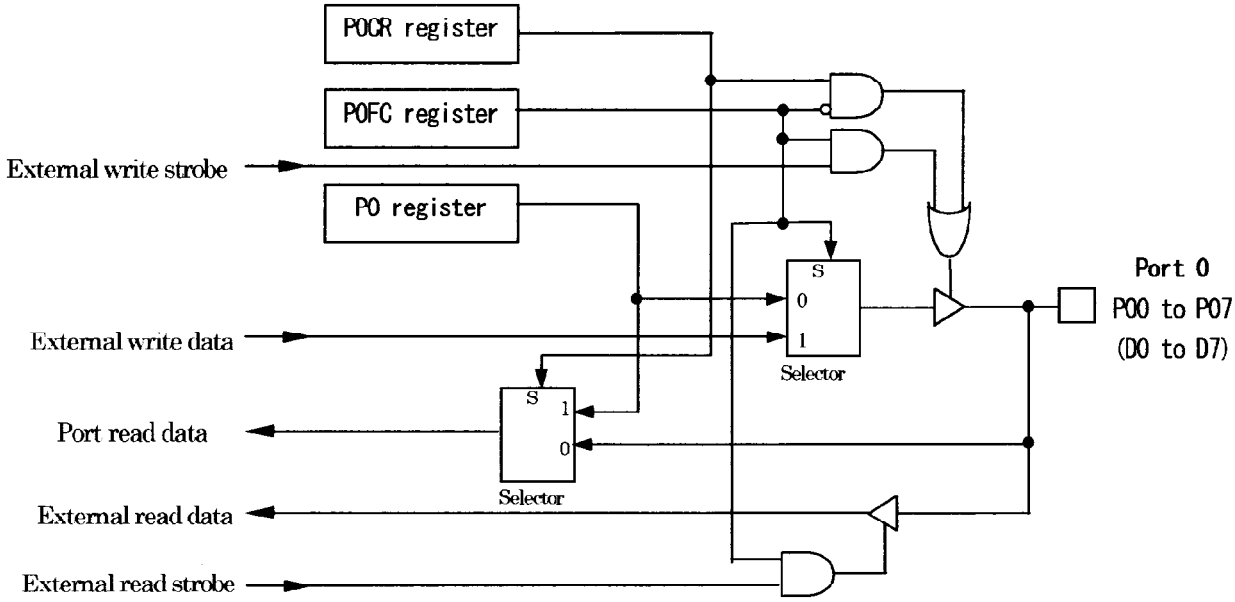


Figure 3.5.1 Port0

Table 3.5.2 Port0 Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
P0	PORT0	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P0CR	PORT0 Control Register	02H	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P0FC	PORT0 Function Register	03H	-	-	-	-	-	-	-	P0F
										W
										0
			0: PORT 1: Data Bus (D7 to D0)							

P0CR<P0xC> \ P0FC<P0F>	0	1
0	Input port	Data bus (D0 to D7)
1	Output port	Data bus (D0 to D7)

3.5.2 Port 4 (P40 to P47)

Port4 is an 8-bit general-purpose I/O ports. Bits can be individually set as either inputs or outputs by control register P4CR and function register P4FC.

In addition to functioning as a general-purpose I/O port, port4 can also function as an address bus (A0 to A7).

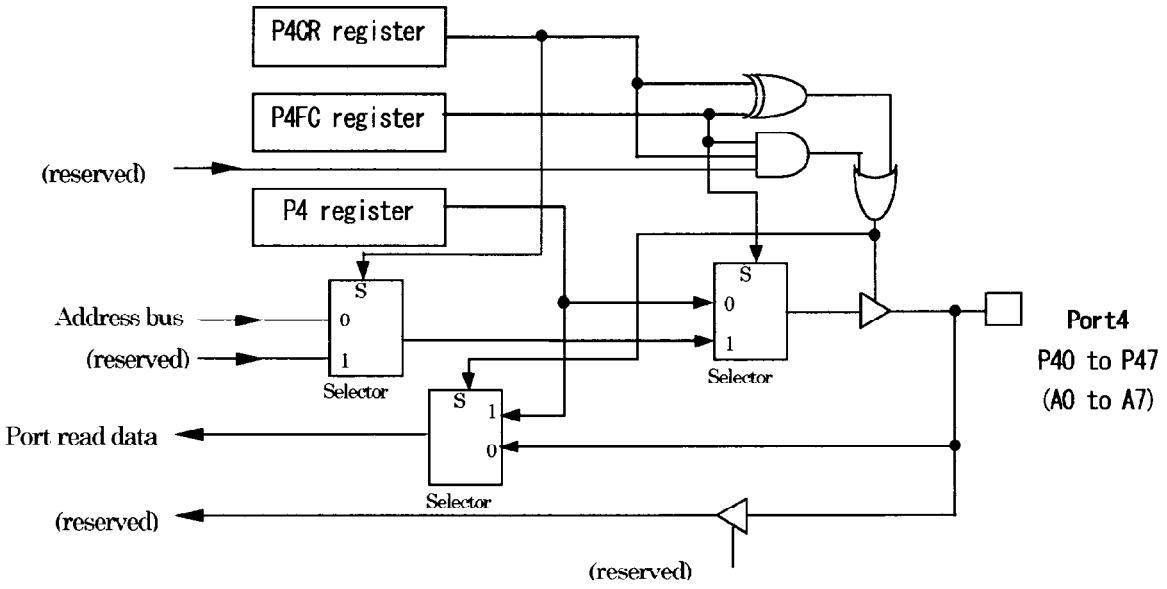


Figure 3.5.2 Port4

Table 3.5.3 Port4 Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
P4	PORT4	10H	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P4CR	PORT4 Control Register	12H	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P4FC	PORT4 Function Register	13H	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
			W							
			0	0	0	0	0	0	0	0
			0: PORT 1: Address Bus (A0 to A7)							

P4CR<P4xC> \ P4FC<P4xF>	0	1
0	Input port	Address bus (A0 to A7)
1	Output port	Don't use this setting.

3.5.3 Port 7 (P70 to P75)

Port7 is a 6-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P7CR and function register P7FC.

In addition to functioning as a general-purpose I/O port. P70. P71 and P73 pins can also function as read/write strobe signals and chip selection to connect with an external memory. P74 pin can also function as clock out signal. P75 pin can also function as wait input.

A reset initializes P70 to P74 pins to output port mode. and P75 pin to input port mode.

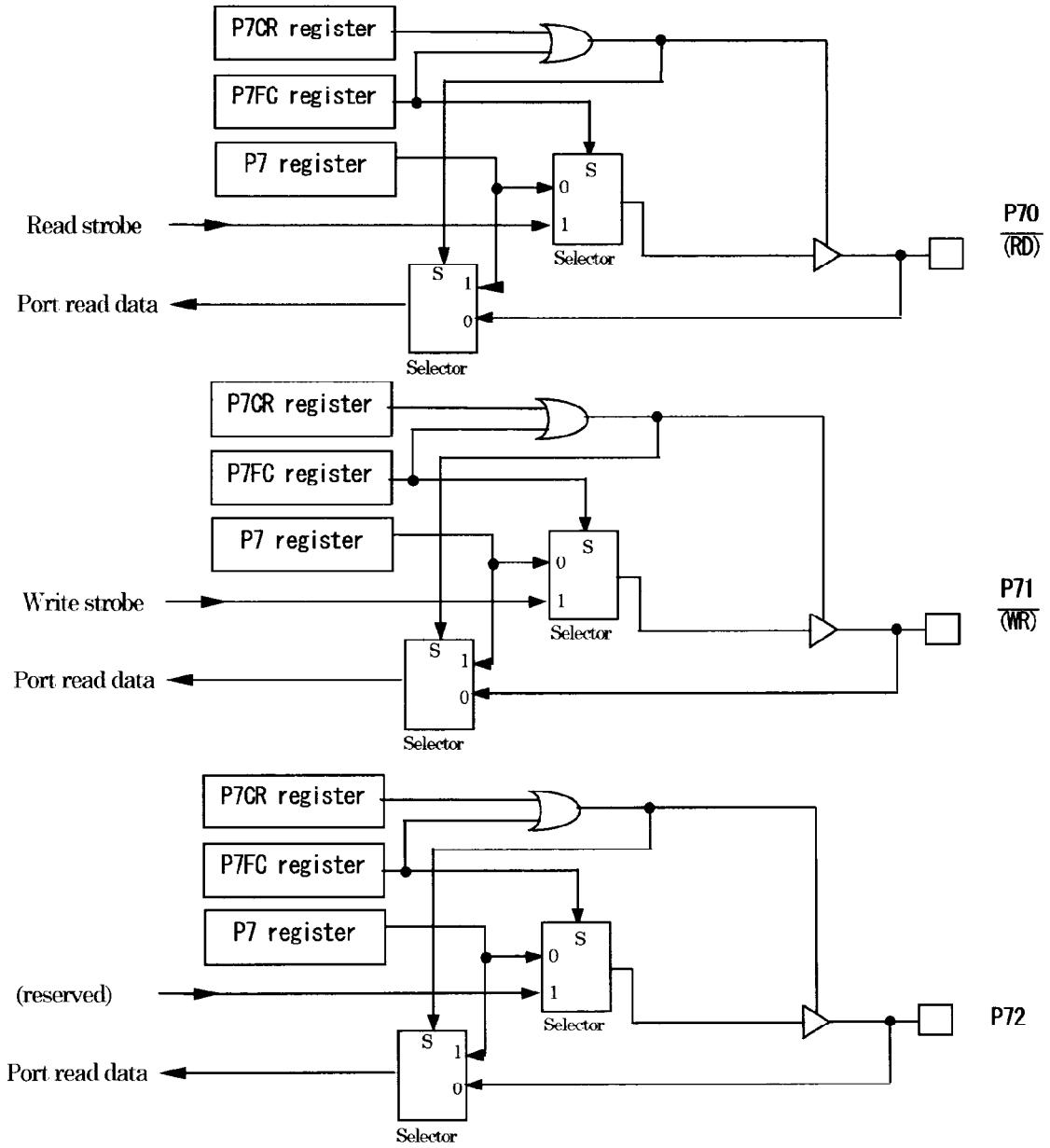


Figure 3.5.3 Port7 (P70 to P72)

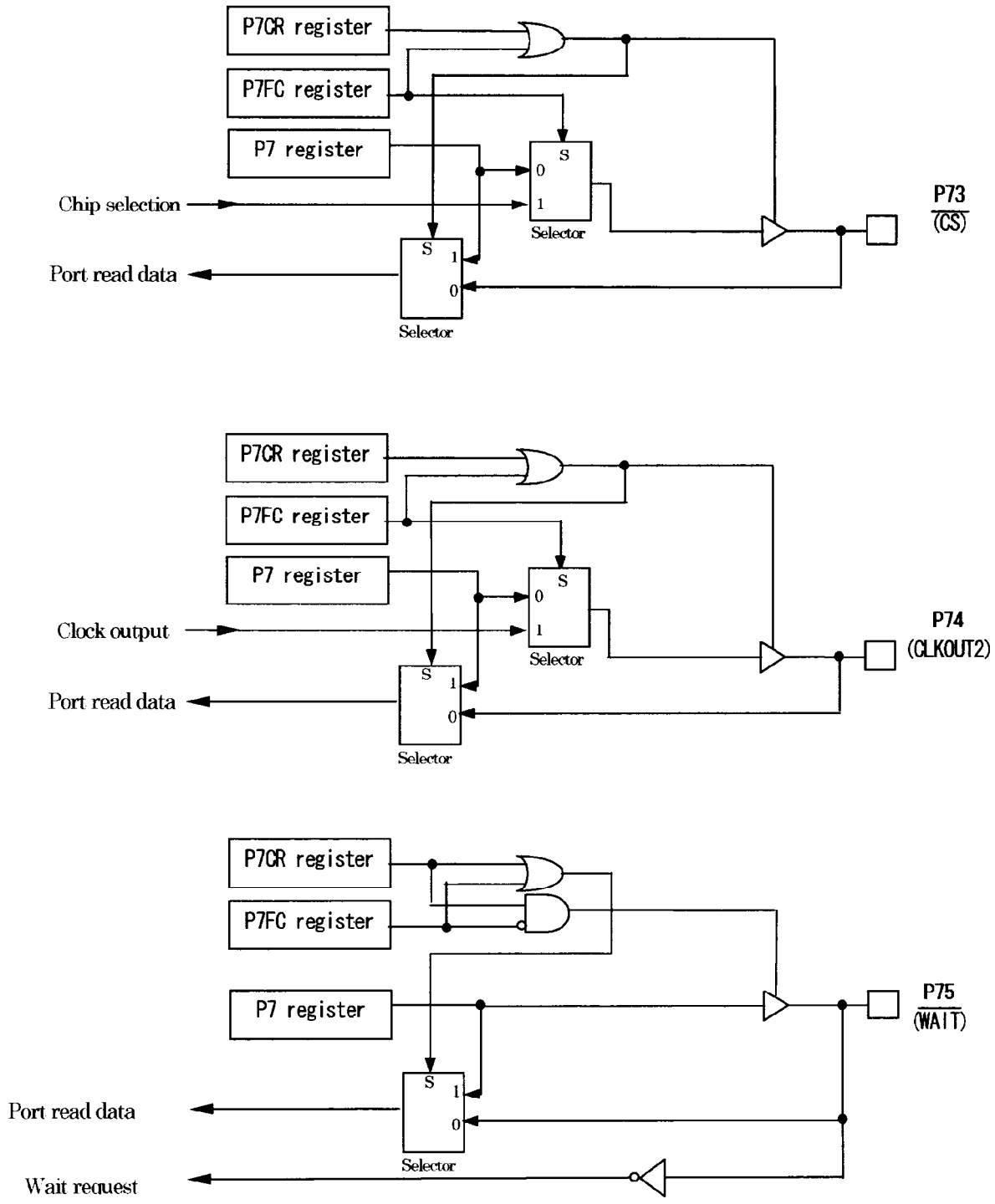


Figure 3.5.4 Port7 (P73 to P75)

Table 3.5.4 Port7 Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
P7	PORT7	1CH	-	-	P75	P74	P73	P72	P71	P70
			R/W							
			-	-	0	1	1	1	1	1
			Input/Output							
P7CR	PORT7 Control Register	1EH	-	-	P75C	P74C	P73C	P72C	P71C	P70C
			W							
			-	-	0	1	1	1	1	1
			0:Input 1:Output							
P7FC	PORT7 Function Register	1FH	-	-	P75F	P74F	P73F	P72F	P71F	P70F
			W							
			-	-	0	0	0	0	0	0
					0:PORT 1:$\overline{WAIT}$	0:PORT 1:CLKOUT2	0:PORT 1:$\overline{CS}$	0:PORT	0:PORT 1:$\overline{WR}$	0:PORT 1:$\overline{RD}$

P7CR	P7FC	-	-	P75	P74	P73	P72	P71	P70
0	0			Input Port. $\overline{WAIT}$	Input Port				
1	0			Output Port					
1	1			$\overline{WAIT}$	CLKOUT2 (4MHz)	$\overline{CS}$	Don't use this setting.	$\overline{WR}$	$\overline{RD}$
0	1			$\overline{WAIT}$	CLKOUT2 (4MHz)	$\overline{CS}$	Don't use this setting.	$\overline{WR}$	$\overline{RD}$

3.5.4 Port C (PC0 to PC5)

PortC is a 6-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PCCR and function register PCFC.

In addition to functioning as a general-purpose I/O port. PortC can also function as 8-bit timer I/O and interrupt input.

A reset initializes PortC to input port mode.

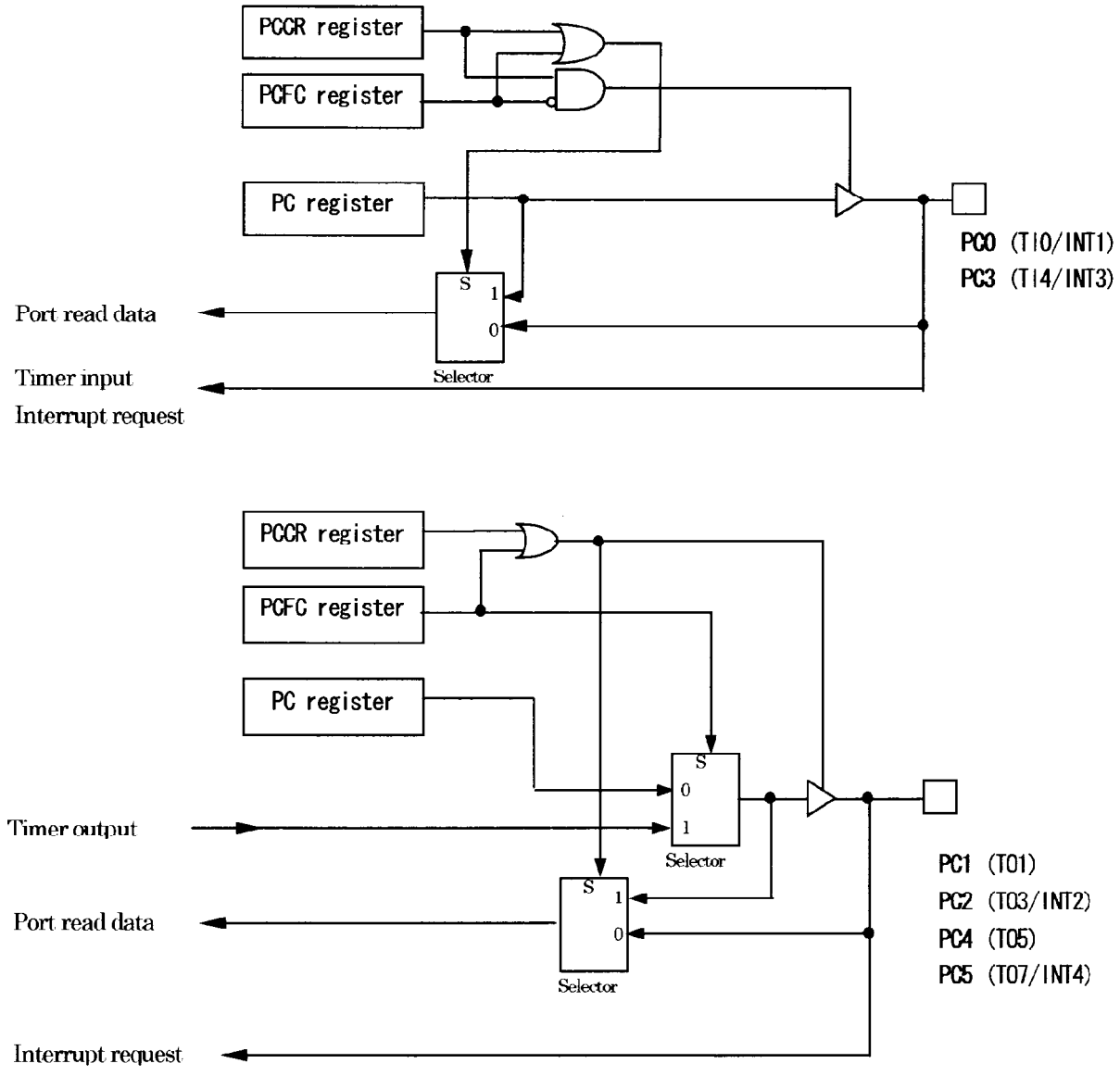


Figure 3.5.5 PortC (PC0 to PC5)

Table 3.5.5 PortC Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
PC	PORTC	30H	-	-	PC5	PC4	PC3	PC2	PC1	PC0
					R/W					
			-	-	0	0	0	0	0	0
					Input/Output					
PCCR	PORTC Control Register	32H	-	-	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
					W					
			-	-	0	0	0	0	0	0
					0: Input 1:Output					
PCFC	PORTC Function Register	33H	-	-	PC5F	PC4F	PC3F	PC2F	PC1F	PC0F
					W					
			-	-	0	0	0	0	0	0
					0:PORT INT4 1:TO7	0:PORT 1:TO5	0:PORT INT3 1:TI4	0:PORT INT2 1:TO3	0:PORT 1:TO1	0:PORT INT1 1:TI0

PCCR	PCFC	-	-	PC5	PC4	PC3	PC2	PC1	PC0
0	0			Input Port, INT4	Input Port	Input Port, INT3, TI4	Input Port, INT2	Input Port	Input Port, INT1, TI0
1	0			Output Port					
1	1			TO7	TO5	TI4	TO3	TO1	TI0
0	1			TO7	TO5	TI4	TO3	TO1	TI0

3.5.5 Port D (PD0 to PD7)

PortD is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PDCR and function register PDFC.

In addition to functioning as a general-purpose I/O port, PortD can also function as 16-bit timer I/O and interrupt input.

A reset initializes PortD to input port mode.

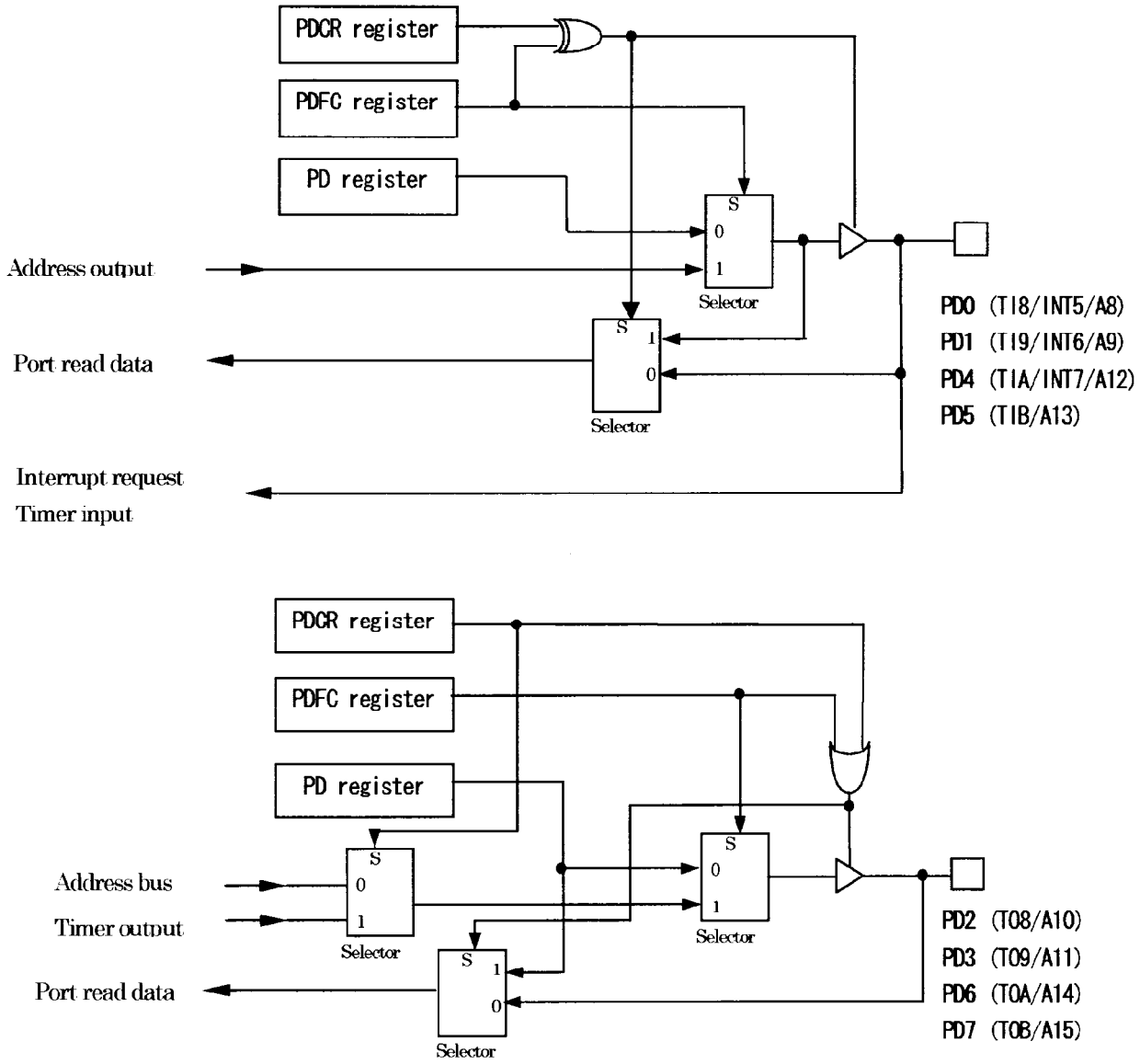


Figure 3.5.6 PortD

Table 3.5.6 PortD Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
PD	PORTD	34H	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
PDCR	PORTD Control Register	36H	PD7C	PD6C	PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
PDFC	PORTD Function Register	37H	PD7F	PD6F	PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
			W							
			0	0	0	0	0	0	0	0
			0:PORT 1:TOB A15	0:PORT 1:TOA A14	0:PORT 1:TIB A13	0:PORT INT7 1:TIA A12	0:PORT 1:TO9 A11	0:PORT 1:TO8 A10	0:PORT INT6 1:TI9 A9	0:PORT INT5 1:TI8 A8

PDCR	PDFC	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
0	0	Input Port	Input Port	Input Port, TIB	Input Port, INT7, TIA	Input Port	Input Port	Input Port, INT6, TI9	Input Port, INT5, TI8
1	0	Output Port							
1	1	TOB	TOA	TIB	TIA INT7	TO9	TO8	TI9 INT6	TI8 INT5
0	1	A15	A14	A13	A12	A11	A10	A9	A8

3.5.6 Port F (PF0 to PF7)

PortF is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PFCR and function register PFFC.

In addition to functioning as a general-purpose I/O port, PortF can also function as I/O functions of serial interface and controller area network (CAN).

A reset initializes PortF to input port mode.

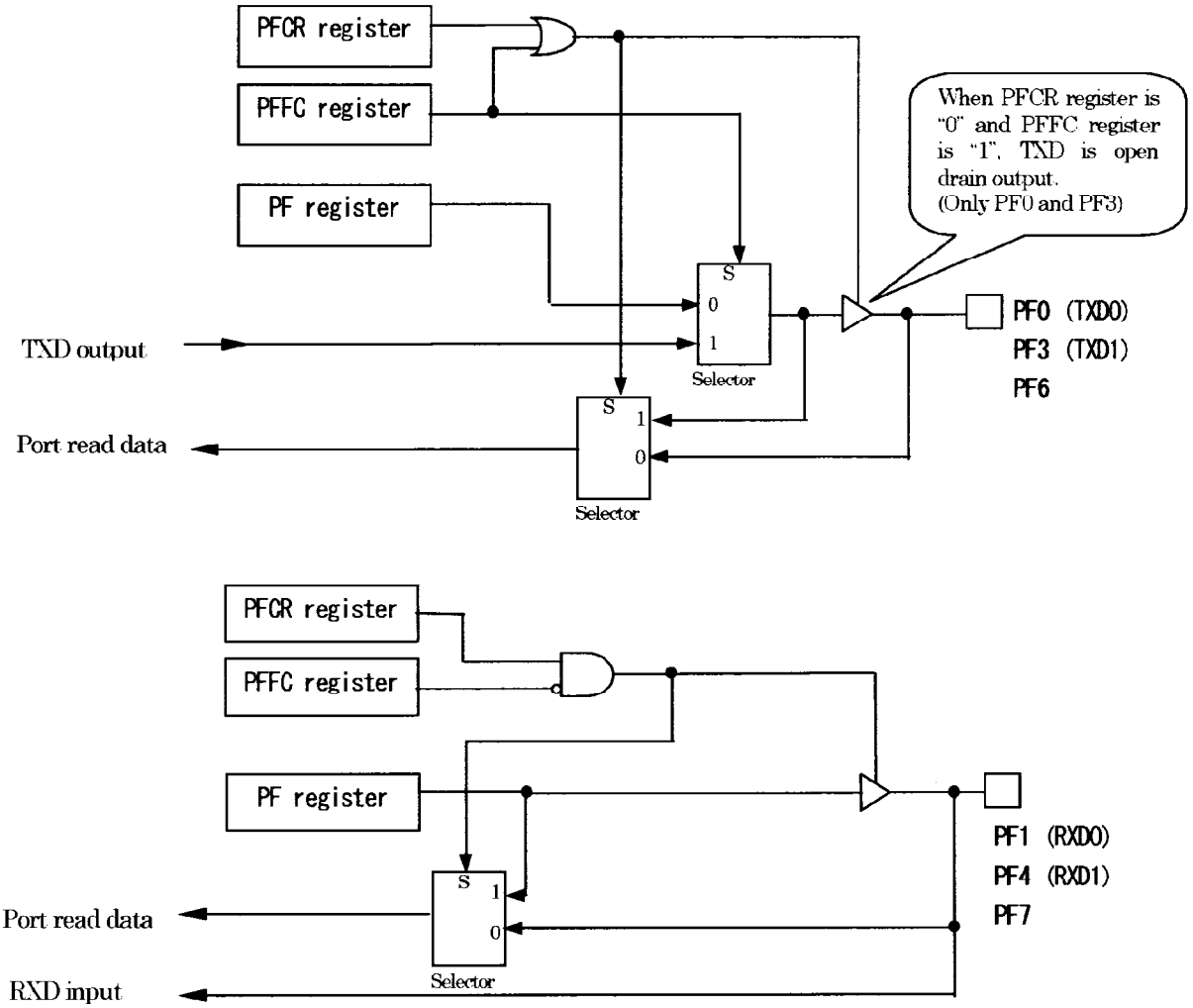


Figure 3.5.7 PortF (PF0, PF1, PF3, PF4, PF6, PF7)

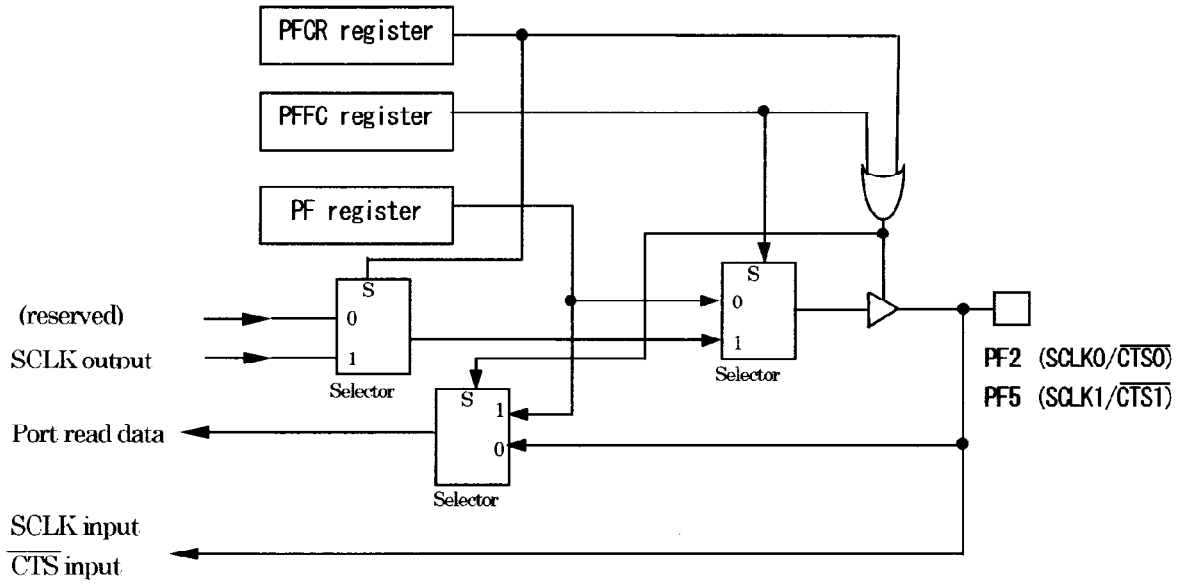


Figure 3.5.8 PortF (PF2, PF5)

Table 3.5.7 PortF Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
PF	PORTF	3CH	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
PFCR	PORTF Control Register	3EH	PF7C	PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
PFFC	PORTF Function Register	3FH	PF7F	PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
			W							
			0	0	0	0	0	0	0	0
			0: PORT Note) Fix to 0.	0: PORT Note) Fix to 0.	0: PORT 1: $\overline{CTS1}$ SCLK1	0: PORT 1: RXD1	0: PORT 1: TXD1	0: PORT 1: $\overline{CTS0}$ SCLK0	0: PORT 1: RXD0	0: PORT 1: TXD0

PFCR	PFFC	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
0	0	Input Port	Input Port	Input Port, SCLK1, $\overline{CTS1}$	Input Port, RXD1	Input Port	Input Port, SCLK0, $\overline{CTS0}$	Input Port, RXD0	Input Port
1	0	Output Port							
1	1	Don't use this setting.	Don't use this setting.	SCLK1	RXD1	TXD1	SCLK0	RXD0	TXD0
0	1	Don't use this setting.	Don't use this setting.	Don't use this setting.	RXD1	TXD1 (Open Drain)	Don't use this setting.	RXD0	TXD0 (Open Drain)

3.5.7 Port G (PG0 to PG7)

PortG is an 8-bit general-purpose input-only port.

In addition to functioning as a general-purpose input-only port, portG can also function as input functions of AD converter.

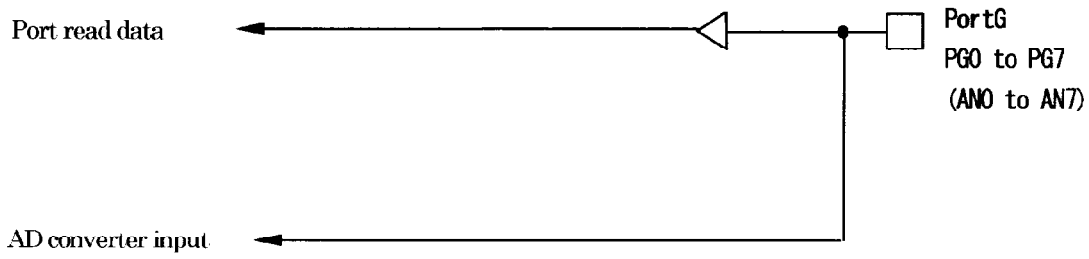


Figure 3.5.9 PortG

Table 3.5.8 PortG Register

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
PG	PORTG	40H	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
			R							
			Input							

3.5.8 Port L (PL0 to PL3)

PortL is a 4-bit general-purpose input-only port.

In addition to functioning as a general-purpose input-only port, portL can also function as input functions of AD converter.

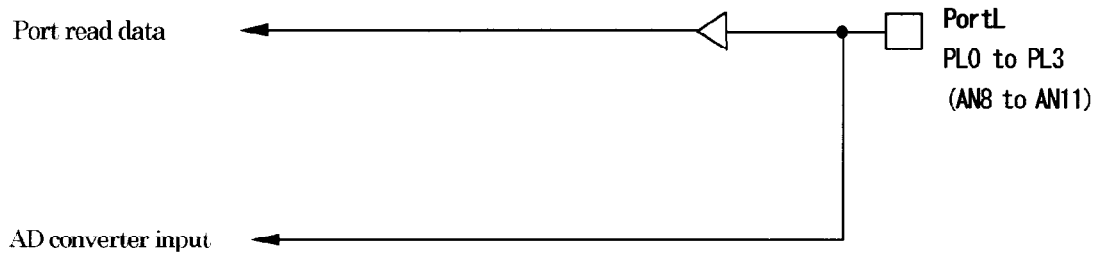


Figure 3.5.10 PortL

Table 3.5.9 PortL Register

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
PL	PORTL	54H	-	-	-	-	PL3	PL2	PL1	PL0
			R							
			Input							

3.5.9 Port M (PM0 to PM7)

PortM is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PMCR and function register PMFC.

In addition to functioning as a general-purpose I/O port. PortM can also function as I/O functions of serial expansion interface.

A reset initializes PortM to input port mode.

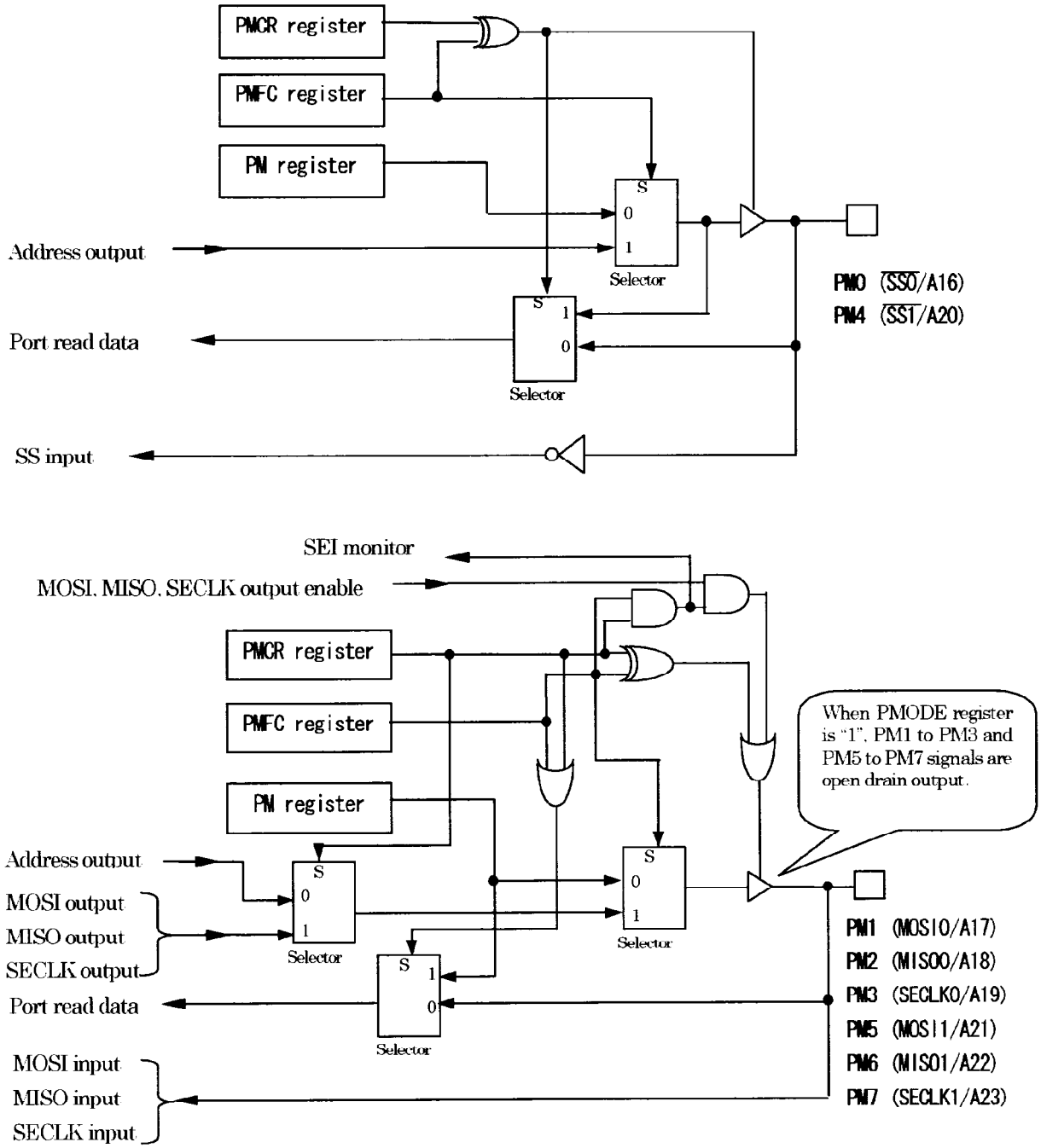


Figure 3.5.11 PortM (PM0 to PM7)

Table 3.5.10 PortM Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
PM	PORTM	58H	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
PMODE	PORTM Open Drain Enable Register	59H	ODEM7	ODEM6	ODEM5	-	ODEM3	ODEM2	ODEM1	-
			R/W					R/W		
			0	0	0	-	0	0	0	-
			PM7 output 0:CMOS 1:Open Drain	PM6 output 0:CMOS 1:Open Drain	PM5 output 0:CMOS 1:Open Drain		PM3 output 0:CMOS 1:Open Drain	PM2 output 0:CMOS 1:Open Drain	PM1 output 0:CMOS 1:Open Drain	
PMCR	PORTM Control Register	5AH	PM7C	PM6C	PM5C	PM4C	PM3C	PM2C	PM1C	PM0C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
PMFC	PORTM Function Register	5BH	PM7F	PM6F	PM5F	PM4F	PM3F	PM2F	PM1F	PM0F
			W							
			0	0	0	0	0	0	0	0
			0:PORT 1:SECLK1 A23	0:PORT 1:MISO1 A22	0:PORT 1:MOSI1 A21	0:PORT 1:$\overline{SS1}$ A20	0:PORT 1:SECLK0 A19	0:PORT 1:MISO0 A18	0:PORT 1:MOSI0 A17	0:PORT 1:$\overline{SS0}$ A16

PMCR	PMFC	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
0	0	Input Port, SECLK1	Input Port, MISO1	Input Port, MOSI1	Input Port, $\overline{SS1}$	Input Port, SECLK0	Input Port, MISO0	Input Port, MOSI0	Input Port, $\overline{SS0}$
1	0	Output Port							
1	1	SECLK1	MISO1	MOSI1	$\overline{SS1}$	SECLK0	MISO0	MOSI0	$\overline{SS0}$
0	1	A23	A22	A21	A20	A19	A18	A17	A16

3.5.10 Port N (PN0 to PN5)

PortN is a 6-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PNCR and function register PNFC.

In addition to functioning as a general-purpose I/O port, PortN can also function as I/O functions of serial channels I/O functions.

A reset initializes PortN to input port mode.

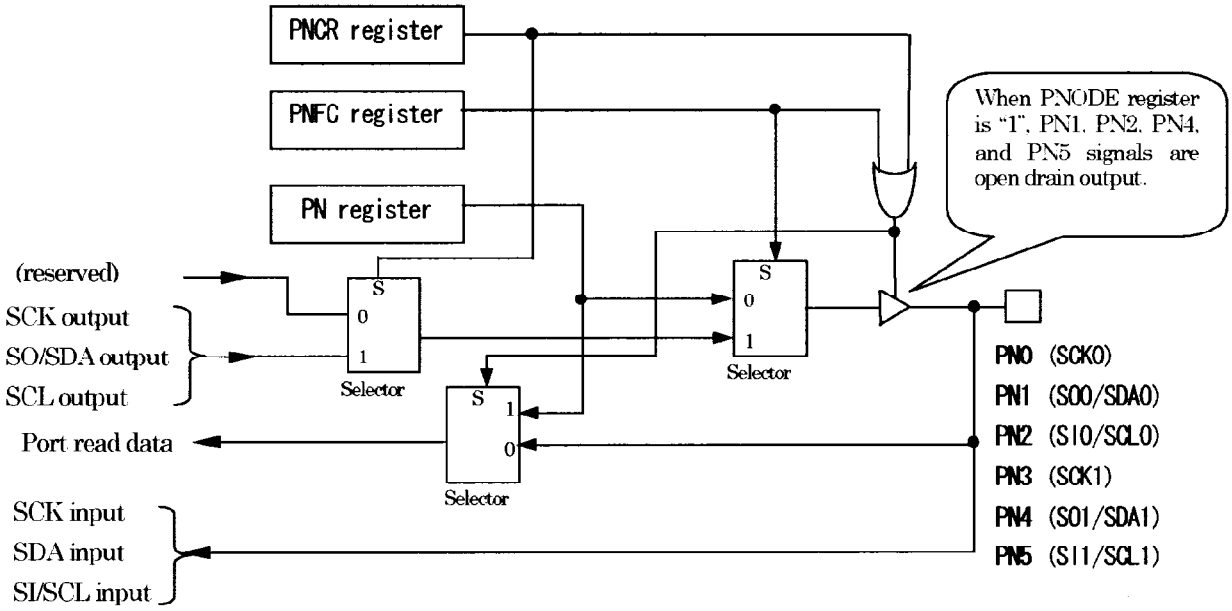


Figure 3.5.12 PortN

Table 3.5.11 PortN Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
PN	PORTN	5CH	-	-	PN5	PN4	PN3	PN2	PN1	PN0
			R/W							
			-	-	0	0	0	0	0	0
			Input/Output							
PNODE	PORTN Open Drain Enable Register	5DH	-	-	ODEN5	ODEN4	-	ODEN2	ODEN1	-
			R/W					R/W		
			-	-	0	0	-	0	0	-
					PN5 Output 0:CMOS 1:Open Drain	PN4 Output 0:CMOS 1:Open Drain		PN2 Output 0:CMOS 1:Open Drain	PN1 Output 0:CMOS 1:Open Drain	
PNCR	PORTN Control Register	5EH	-	-	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
			W							
			-	-	0	0	0	0	0	0
			0:Input 1:Output							
PNFC	PORTN Function Register	5FH	-	-	PN5F	PN4F	PN3F	PN2F	PN1F	PN0F
			W							
			-	-	0	0	0	0	0	0
					0:PORT 1:SI1 SCL1	0:PORT 1:SO1 SDA1	0:PORT 1:SCK1	0:PORT 1:SI0 SCL0	0:PORT 1:SO0 SDA0	0:PORT 1:SCK0

PNCR	PNFC	-	-	PN5	PN4	PN3	PN2	PN1	PN0
0	0			Input Port, SI1/SCL1	Input Port, SDA1	Input Port, SCK1	Input Port, SI0/SCL0	Input Port, SDA0	Input Port, SCK0
1	0			Output Port					
1	1			SCL1	SO1/SDA1	SCK1	SCL0	SO0/SDA0	SCK0
0	1			Don't use this setting.					

3.6 Memory Controller

3.6.1 Function

TMP92CW10 has a memory controller with a variable 1-block address area whitch is controls as follows.

(1) 1-block address area support

Specifies a start address and a block size for 1-block address area.

(2) Connecting memory specifications

Specifies SRAM and ROM memory devices to be associated with selected address areas.

(3) Data bus size selection

Only 8-bit is selected as the data bus size of the selected address area.

(4) Wait control

The wait specification bit in the control register and WAIT input pin can be used to control the number of waits in the external cycle. Tue number of waits for Read cycles and Read cycle can be specified.

The number of waits can be determined by selecting one of the five mode mentioned below.

0 wait, 1wait.
2 wait, 3 wait
N wait(controls with $\overline{\text{WAIT}}$ pin)

3.6.2 Control register and Operation after reset release

This section describes the registers to control the memory controller, the state after reset release and necessary settings.

(1) Control Register

The control registers of the memory controller are as follows.

- Control registers: BCSH/BCSL
  Sets the basic functions of the memory controller, that is the connecting memory type, the number of waits to be read and written.
- Memory start address register: MSAR
  Sets a start address in the selected address areas.
- Memory address mask register: MAMR
  Sets a block size in the selected address areas.

(2) Operation after reset release

After reset, the selected address area is set to address 000000H to FFFFFFH.

After reset release, the block address areas are specified by the memory start address register(MSAR) and the memory address mask register(MAMR). Then the control register (BCS) is set.

Set the enable bit(BE) of the control register to"1" to enable the setting.

3.6.3 Basic functions and register setting

In this section, setting of the block address area, the connecting memory and the number of waits out of the memory controller's functions are described.

(1) Block address area specification

The block address area is specified by two registers.

The memory start addressregister(MSAR) sets the start address of the block address areas. The memory controller compares between the register value and the address every bus cycles. The address bit which is masked by the memory address maskregister(MAMR) is not compared by the memory controller. The block address area size is determined by setting the memory address mask register. The set value in the register is compared with the block address area on the bus. If the compared result is a match, the memory controller sets the chip select signal($\overline{CS}$) to "Low".

(i) Setting memory start address register

The MS23 to 16 bits of the memory start address register respectively correspond with addresses A23 to A16. The lower start address A15 to 0 are always set to address 0000H. Therefore the startaddress of the block address area are set to addresses 000000H to FF0000H every 64K bytes.

(ii) Setting memory address mask registers

The memory address mask register sets whether an address bit is compared or not. Set the register to "0" to compare, or to "1" not to compare.

The address bit to be set is depended on the block address area.

Block addressarea : A22 to A15

The above-mentioned bits are always compared. The block address area size is determined by the compared result.

The size to be set depending on the block address area is as follows.

Size (bytes) / CS area	256	512	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M
CS			O	O	O	O	O	O	O	O	O

Note: After reset release, <BM> bit of the control register is set to'0', and the block address area is set to addresses 000000H to FFFFFFH. Setting <BM> bit to"1" specifies the start address and the address area size.

(iii) Example of register setting

To set the block address area 64kbytes from address 110000H, set the register as follows.

MSAR Register

bit	7	6	5	4	3	2	1	0
bit Symbol	MS23	MS22	MS21	MS20	MS19	MS18	MS17	MS16
value	0	0	0	1	0	0	0	1

MS23 to 16 bits of the memory start address register MSAR correspond with address A23 to A16.

A15 to 0 are set to"0". Therefore setting MSAR to the above-mentioned value specifies the start address of the block address area to address 110000H.

MAMR Register

bit	7	6	5	4	3	2	1	0
bit Symbol	MV22	MV21	MV20	MV19	MV18	MV17	MV16	MV15
value	0	0	0	0	0	0	0	1

MV22 to 15 bits of the memory address mask register MAMR set whether address A22 to 15 are compared or not. Set the register to "0" to compare, or to "1" not to compare. A23 are always compared

Setting the above-mentioned compares A23 to A16 with the values set as the start address. Therefore 64kbytes of addresses 110000H to 11FFFFH are set as the block address area, and compared with the addresses on the bus. If the compared result is a match, the chip select signal $\overline{CS}$ is set to"LOW".

Note: When the set block address area overlaps with the built-in memory area, the block address area is processed according to priority as follows.

Built-in I/O > Built-in memory > Block address area

Note also that any accessed areas outside the address spaces set are set to 2wait bus cycle.

(2) Wait control

The external bus cycle completes a wait of two states at least (100ns @20MHz). Setting the BWW2 to 0 and BWR2 to 0 of the control register (BCSL) specifies the number of waits in the read cycle and the write cycle. BWW is set with the same method as BWR.

BWW/BWR bit (BCSL Regsiter)

BWW2 BWR2	BWW1 BWR1	BWW0 BWR0	Function
0	0	1	2states(0 wait) access fixed mode
0	1	0	3states(1 wait) access fixed mode(Default)
1	0	1	4states(2 wait) access fixed mode
1	1	0	5states(3 wait) access fixed mode
0	1	1	$\overline{\text{WAIT}}$ pin input mode
other			(Reserved)

(i) Waits number fixed mode

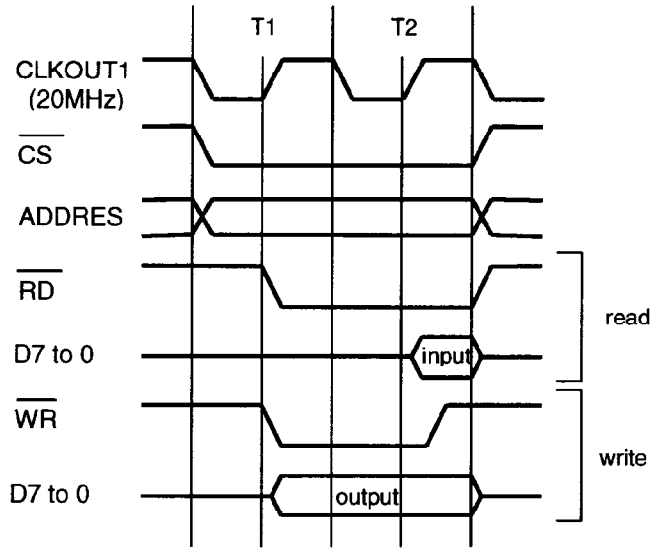
The bus cycle is completed with the set states. The number of states is selected from 2 states(0WAIT) to 5states(3WAIT).

(ii) $\overline{\text{WAIT}}$ pin input mode

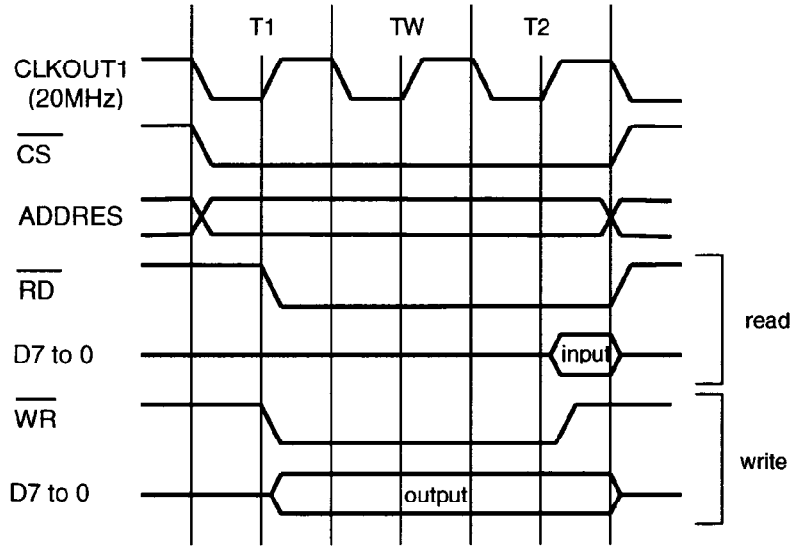
This mode samples the $\overline{\text{WAIT}}$ input pins. It continuously samples the $\overline{\text{WAIT}}$ pin state and inserts a wait if the pin is active. The bus cycle is minimum 2 states. The bus cycle is completed when the wait signal is non-active(goes High) at 2 states. The bus cycle extends if the wait signal is active at 2 states and more.

(3) Basic bus timing

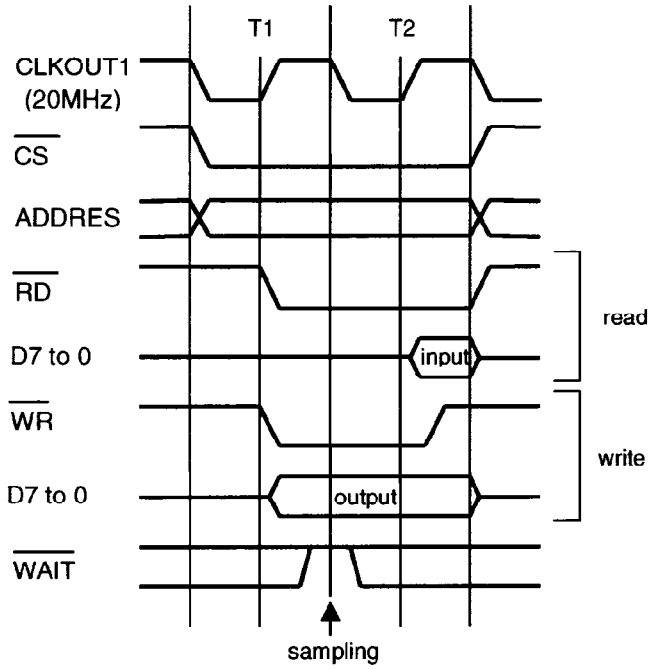
• External Read / Write Bus Cycle (0 WAIT)



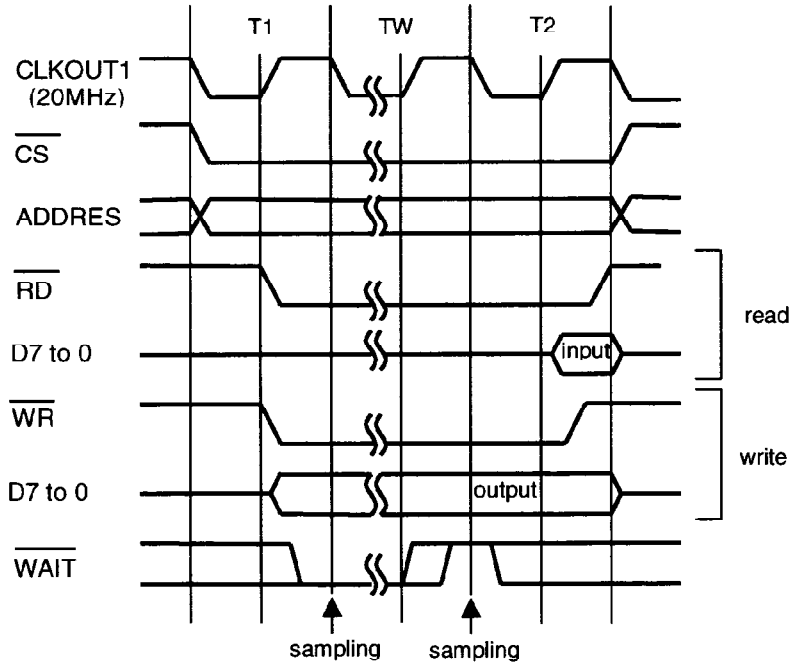
• External Read / Write Bus Cycle (1 WAIT)



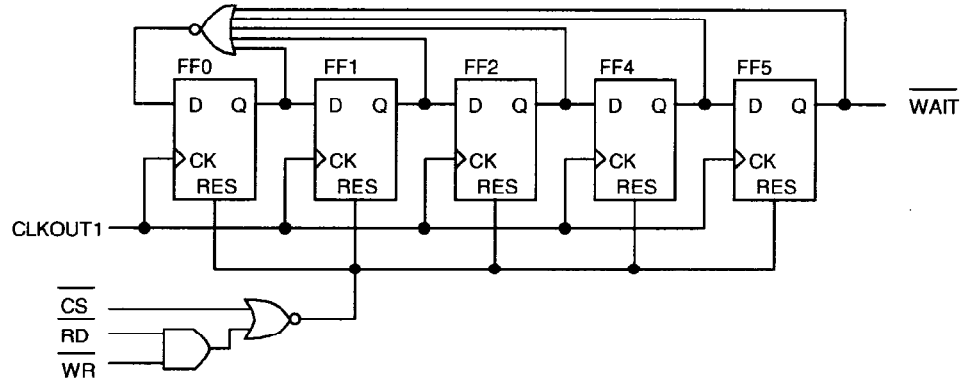
- External Read / Write Bus Cycle (0 WAITS in WAIT pin input mode)

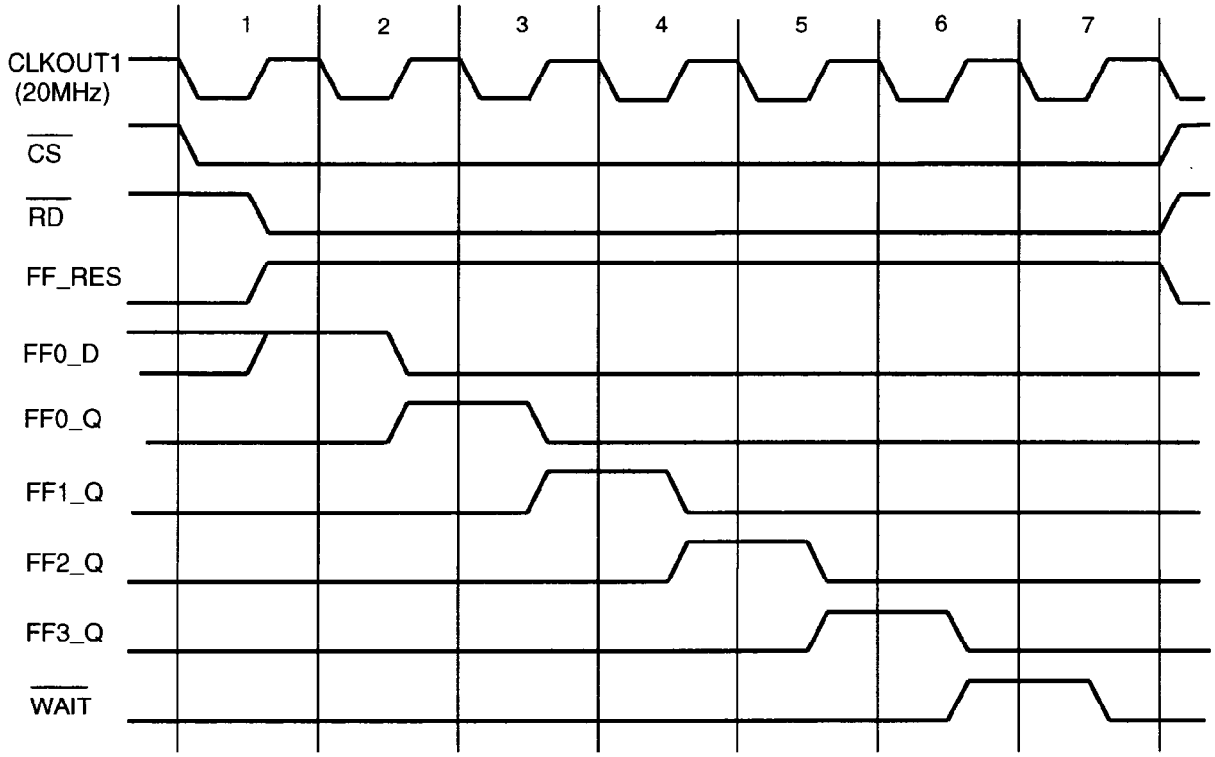


- External Read / Write Bus Cycle (n WAITS in WAIT pin input mode)



• Example of WAIT Input Cycle (6WAIT)





3.6.4 List of registers

The memory control registers and the settings are described as follows. For the addresses of the registers, please refer to the List of Special Function Registers in section 5.

(1) Control registers

The control register is a pair of BCSL and BCSH. BCSL has the same configuration regardless of the block address areas.

Block CS/Wait Control Register (L)

BCSL (0148H)

	7	6	5	4	3	2	1	0
bit Symbol	-	BWW2	BWW1	BWW0	-	BWR2	BWR1	BWR0
Read/Write	W							
After Reset	-	0	1	0	-	0	1	0

BWW[2:0] Specifies the number of write waits.

001 = 2 states (0 WAIT) access 010 = 3 state (1 WAIT) access

101 = 4 states (2 WAIT) access 110 = 5 state (3 WAIT) access

011 = WAIT pin input mode Others = (Reserved)

BWR[2:0] Specifies the number of read waits.

001 = 2 states (0 WAIT) access 010 = 3 state (1 WAIT) access

101 = 4 states (2 WAIT) access 110 = 5 state (3 WAIT) access

011 = WAIT pin input mode Others = (Reserved)

Block CS/Wait Control Register (H)

BCSH (0149H)

	7	6	5	4	3	2	1	0
bit Symbol	BE	BM	-	-	BOM1	BOM0	BBUS1	BBUS0
Read/Write	W							
After reset	1	0	0	0	0	0	0	0

BE Enable bit

0 = No chip select signal output

1 = Chip select signal output(Default)

Note: After reset release, only the enable bit BE of BCS register is valid("1").

BM Block address area specification

0 = Sets the block address area of CS to addresses 000000H to FFFFFFH. (Default)

1 = Sets the block address area of CS to programmable.

Note: After reset release, the block address area of CS is set to addresses 000000H to FFFFFFH.

BOM[1:0]

00 = SRAM or ROM(Default)

others = (Reserved)

BBUS[1:0] Sets the data bus width

00 = 8bit(Default)

others = (Reserved)

(2) Block address register

A start address and an address area of the block address are specified by the memory start address register(MSAR) and the memory address mask register(MAMR). The memory start address register sets all start address similarly regardless of the block address areas. The bit to be set by the memory address mask register is depended on the block address area.

Memory Start Address Register

MSA (014BH)

	7	6	5	4	3	2	1	0
bit Symbol	MS23	MS22	MS21	MS20	MS19	MS18	MS17	MS16
Read/Write	R/W							
After Reset	1	1	1	1	1	1	1	1

MS[23:16] Sets a start address.

Sets the start address of the block address areas. The bit are corresponding to the address A23 to 16.

Memory Address Mask Register

MAMR (014AH)

	7	6	5	4	3	2	1	0
bit Symbol	MV22	MV21	MV20	MV19	MV18	MV17	MV16	MV15
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

MV[22:15]

Enables or masks comparison of the addresses. MV22 to MV15 are corresponding to addresses A22 to 15. If "0" is set, the comparison between the value of the address bus and the start address is enabled. If "1" is set, the comparison is inhibited.

3.7 8-bit Timers

The TMP92CW10 features eight built-in 8-bit timers (timers 0 to 7).

- 8-Bit Interval Timer Mode
- 16-Bit Interval Timer Mode
- 8-Bit Programmable Square Wave Pulse Generation Output Mode (PPG – variable duty with variable cycle)
- 8-Bit Pulse Width Modulation Output Mode (PWM – variable duty with constant cycle)

Figure 3.7.1 to Figure 3.7.4show block diagrams for Timer01, Timer 23, Timer 45 and Timer 67. Each channel consists of an 8-bit up-counter, an 8-bit comparator and an 8-bit timer register. In addition, a timer flip-flop and a prescaler are provided for each pair of channels. The operation mode and timer flip-flops are controlled by five control SFRs (special-function registers).

Each of the four modules (timers 01, timers 23, timers 45 and timers 67) can be operated independently. All modules operate in the same manner; hence only the operation of timers 01 is explained here.

Table 3.7.1 Module Specification Differences

Specification \ Module		Timer 01	Timer 23	Timer 45	Timer 67
External pin	Input pin for external clock	TI0 (shared with PC0)	No	TI4 (shared with PC3)	No
	Output pin for timer flip-flop	TO1 (shared with PC1)	TO3 (shared with PC2)	TO5 (shared with PC4)	TO7 (shared with PC5)
SFR (address)	Timer run register	TRUN01 (0080H)	TRUN23 (0088H)	TRUN45 (0090H)	TRUN67 (0098H)
	Timer register	TREG0 (0082H) TREG1 (0083H)	TREG2 (008AH TREG3 (008BH)	TREG4 (0092H) TREG5 (0093H)	TREG6 (009AH TREG7 (009BH)
	Timer mode register	TMOD01 (0084H)	TMOD23 (008CH)	TMOD45 (0094H)	TMOD67 (009CH)
	Timer flip-flop control register	TFFCR1 (0085H)	TFFCR3 (008DH)	TFFCR5 (0095H)	TFFCR7 (009DH)

3.7.1 Block diagrams

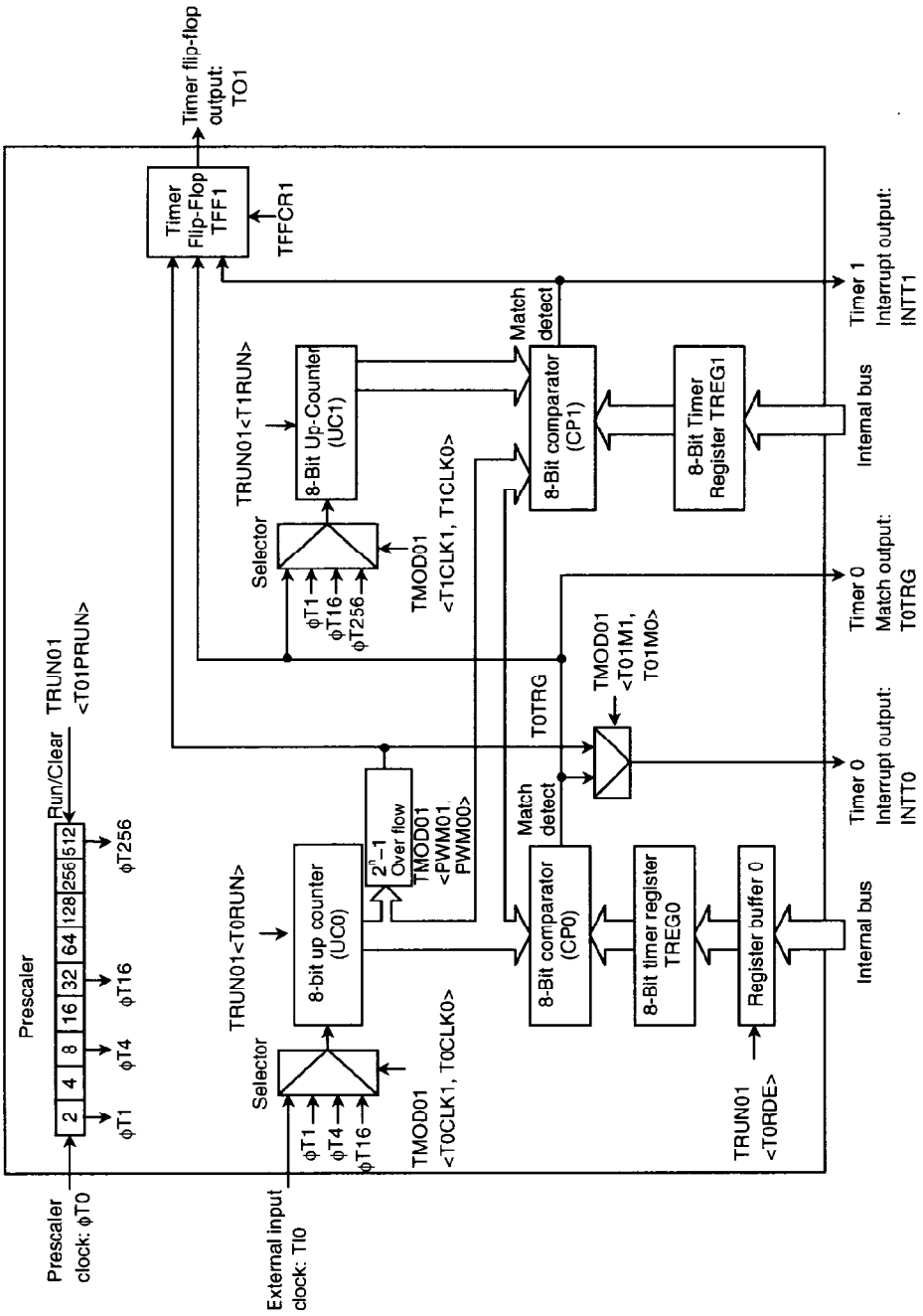


Figure 3.7.1 Timers 01 block diagram

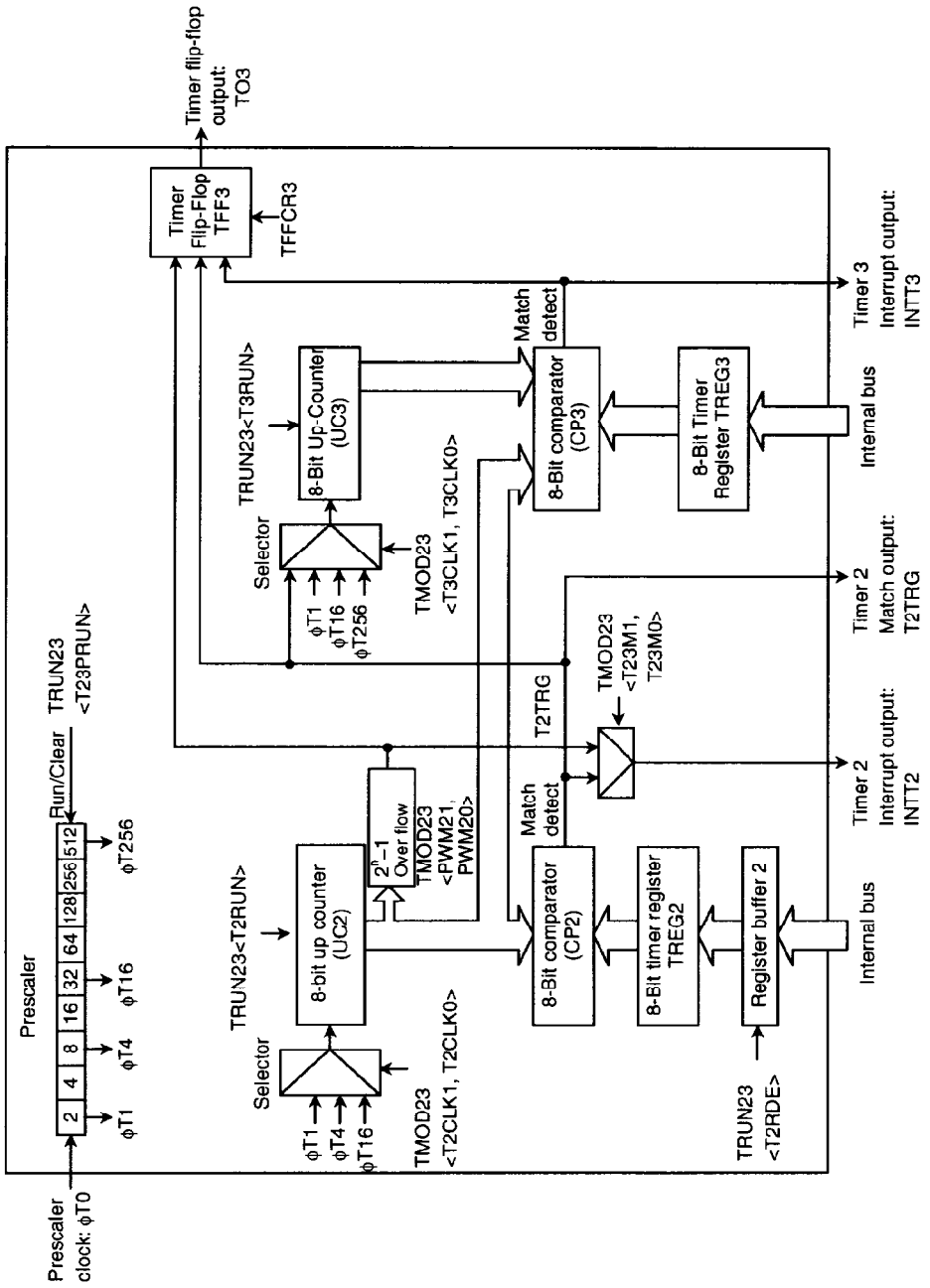


Figure 3.7.2 Timers 23 block diagram

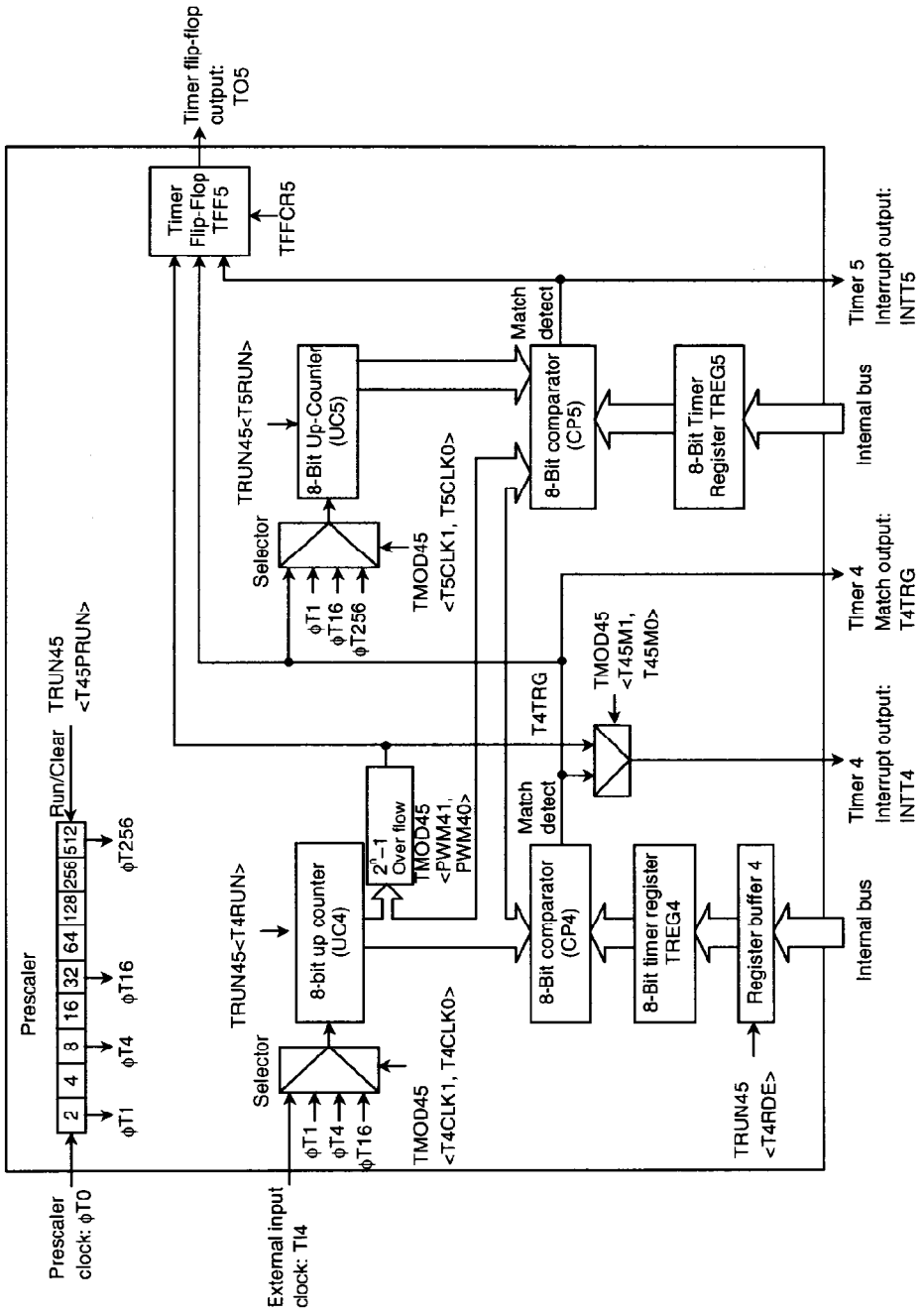


Figure 3.7.3 Timers 45 block diagram

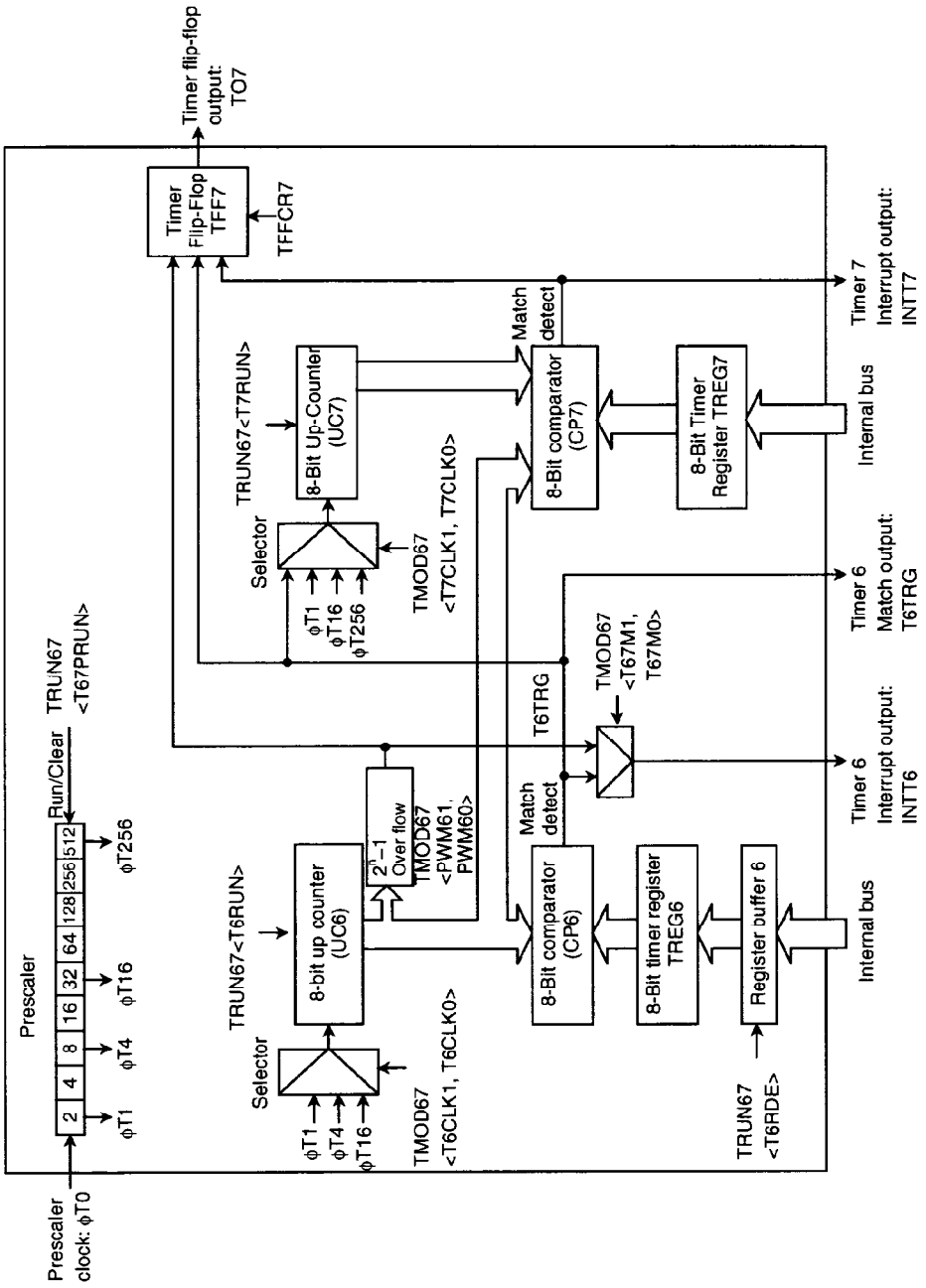


Figure 3.7.4 Timers 67 block diagram

3.7.2 Operation of each circuit

(1) Prescalers

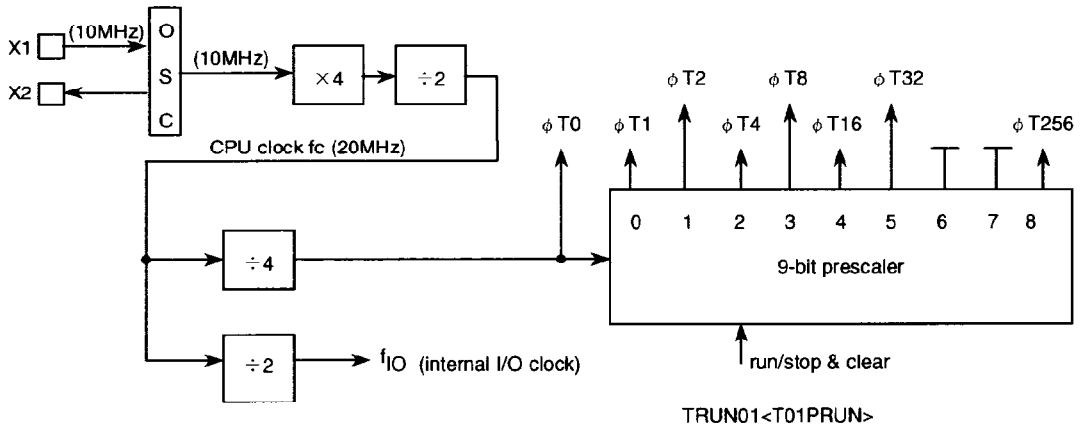
A 9-bit prescaler generates the input clock to timers 01.
The clock ϕT0 is divided by 4 the CPU clock fc and input to this prescaler.

The prescaler's operation can be controlled using TRUN01<T01PRUN> in the timer control register. Setting <T01PRUN> to 1 starts the count: setting <T10PRUN> to 0 clears the prescaler to zero and stops operation.

At fc = 20 MHz

Input clock	Interval
ϕT1 (8/fc)	400 n
ϕT4 (32/fc)	1.6 µs
ϕT16 (128/fc)	6.4 µs
ϕT256 (2048/fc)	102.4 µs

Note: The following number in the parenthesis indicates the frequency when TMP92CW10 operates is the maximum frequency.



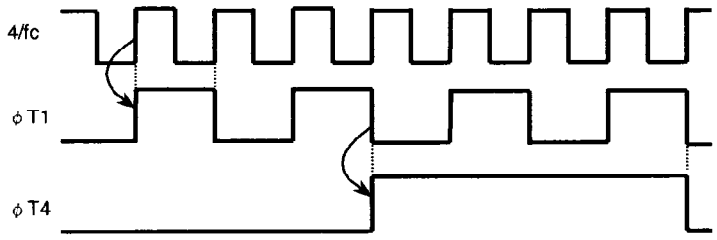


Figure 3.7.5 Prescaler

(2) Up-counters (UC0 and UC1)

These are 8-bit binary counters which count up the input clock pulses for the clock specified by TMOD01.

The input clock for UC0 is selectable and can be either the external clock input via the TI0 pin or one of the three internal clocks ϕT1, ϕT4 or ϕT16. The clock setting is specified by the value set in TMOD01<T01CLK1,T01CLK0>.

The input clock for UC1 depends on the operation mode. In 16-Bit Timer Mode, the overflow output from UC0 is used as the input clock. In any mode other than 16-Bit Timer Mode, the input clock is selectable and can either be one of the internal clocks ϕT1, ϕT16 or ϕT256, or the comparator output (the match detection signal) from timer 0.

For each interval timer the timer operation control register bits TRUN01<T0RUN> and TRUN01<T1RUN> can be used to stop and clear the up-counters and to control their count. A Reset clears both up-counters, stopping the timers.

(3) Timer registers (TREG0 and TREG1)

These are 8-bit registers which can be used to set a time interval. When the value set in the timer register TREG0 or TREG1 matches the value in the corresponding up-counter, the Comparator Match Detect signal goes Active. If the value set in the timer register is 00H, the signal goes Active when the up-counter overflows.
The TREG0 and Register Buffer conprise a double-buffer structure.
The setting of the bit TRUN01<T0RDE> determines whether TREG0's double-buffer structure is enabled or disabled. It is disabled if <T0RDE> = 0 and enabled if <T0RDE> = 1.
When the double buffer is enabled, data is transferred from the register buffer to the timer register when a 2^n-1 overflow occurs in PWM Mode, or at the start of the PPG cycle in PPG Mode. Hence the double buffer cannot be used in Timer Mode.
A Reset initializes <T0RDE> to 0, disabling the double buffer. To use the double buffer, write data to the timer register, set <T0RDE> to 1, and write the following data to the register buffer. Figure 3.7.6 shows the configuration of TREG0.

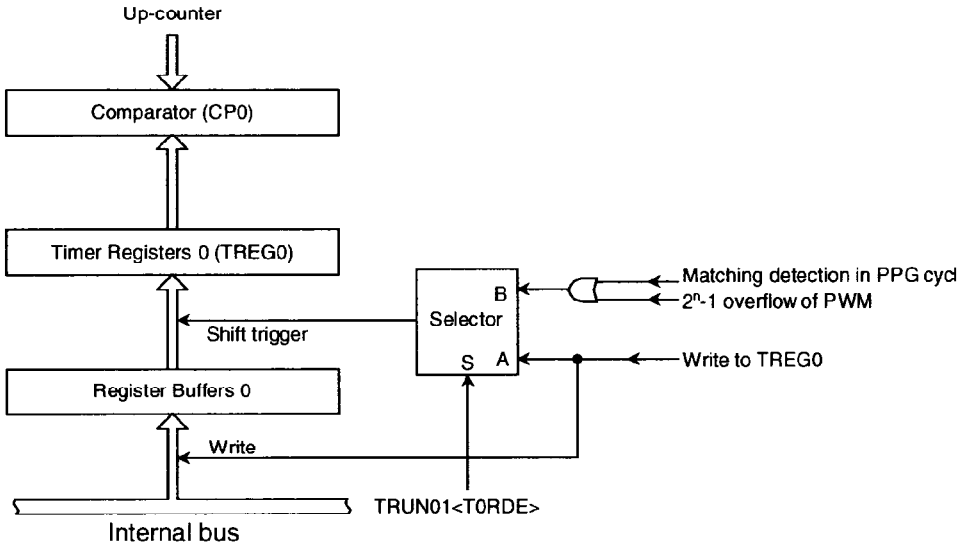


Figure 3.7.6 Configuration of TREG0

Note: The same memory address is allocated to the timer register and the register buffer. When <T0RDE> = 0, the same value is written to the register buffer and the timer register; when <T0RDE> = 1, only the register buffer is written to.

The address of each timer register is as follows.

TREG0: 000082H	TREG1: 000083H
TREG2: 00008AH	TREG3: 00008BH
TREG4: 000092H	TREG5: 000093H
TREG6: 00009AH	TREG7: 00009BH

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

(4) Comparator (CP0)

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

(5) Timer flip-flop (TFF1)

The timer flip-flop (TFF 1) is a flip-flop inverted by the match detect signal (8-bit comparator output) generated by each internal timer..

Whether inversion is enabled or disabled is determined by the setting of the bit TFFCR1<TFF1IE> in the Timer Flip-Flop Control Register.

A Reset clears the value of TFF1 to 0. Writing 01 or 10 to TFFCR1<TFF1C1.TFF1C0> sets TFF1 to 0 or 1. Writing 00 to these bits inverts the value of TFF1 (this is known as software inversion).

The TFF1 signal is output via the TO1 pin (which can also be used as PC1). When this pin is used as the timer output, the timer flip-flop should be set beforehand using the Port C Function Register PCFC.

3.7.3 SFRs

Timers 01 Run Register

TRUN01 (0080H)

	7	6	5	4	3	2	1	0
Bit symbol	T0RDE				I2T01	T01PRUN	T1RUN	T0RUN
Read/Writ	R/W				R/W			
After Reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enabl				IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		

TREG0 double buffer control

0	Disable
1	Enabl

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

I2T01: Operation in IDLE2 Mod
T01PRUN: Run prescaler
T1RUN: Run Timer 1
T0RUN: Run Timer 0

Note: The values of bits 4 to 6 of TRUN01 are undefined when read.

Timers 23 Run Register

TRUN23 (0088H)

	7	6	5	4	3	2	1	0
Bit symbol	T2RDE				I2T23	T23PRUN	T3RUN	T2RUN
Read/Writ	R/W				R/W			
After Reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enabl				IDLE2 0: Stop 1: Rung	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		

TREG2 double buffer control

0	Disable
1	Enabl

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

I2T23: Operation in IDLE2 Mod
T23PRUN: Run prescaler
T3RUN: Run Timer 3
T2RUN: Run Timer 2

Note: The values of bits 4 to 6 of TRUN23 are undefined when read.

Figure 3.7.7 Register for 8-bit Timers (TRUN01 and TRUN23)

Timers 45 Run Register

TRUN45 (0090H)

	7	6	5	4	3	2	1	0
Bit symbol	T4RDE				I2T45	T45PRUN	T5RUN	T4RUN
Read/Writ	R/W					R/W		
After Reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enabl				IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		

TREG4 double buffer control

0	Disable
1	Enabl

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

I2T45: Operation during IDLE2-Mod
T45PRUN: Run for prescaler
T5RUN: Run Timer 5
T4RUN: Run Timer 4

Note: The values of bits 4 to 6 of TRUN45 are undefined when read.

Timers 67 Run Register

TRUN67 (0098H)

	7	6	5	4	3	2	1	0
Bit symbol	T6RDE				I2T67	T67PRUN	T7RUN	T6RUN
Read/Writ	R/W					R/W		
After Reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enabl				IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		

TREG6 double buffer control

0	Disable
1	Enalbe

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

I2T67: Operation during IDLE2 Mod
T67PRUN: Run prescaler
T7RUN: Run Timer 7
T6RUN: Run Timer 6

Note: The values of bits 4 to 6 TRUN67 are undefined when read.

Figure 3.7.8 Register for 8-bit Timers (TRUN45 and TRUN67)

Timers 01 Mode Register

TMOD01 (0084H)

	7	6	5	4	3	2	1	0
Bit symbol	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read/Writ	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-Bit Timer Mod 01: 16-Bit Timer Mode 10: 8-Bit PPG Mod 11: 8-Bit PWM Mod		PWM cycle 00: reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		Source clock for Timer 1 00: T0TRG 01: φT1 10: φT16 11: φT256		Source clock for Timer 0 00: TI0 pin (Note) 01: φT1 10: φT4 11: φT16	

Timer 0 source clock selection

00	TI0 (external input)
01	φT1 (prescaler)
10	φT4 (prescaler)
11	φT16 (prescaler)

Timer 1 source clock selection

	TMOD01 <T01M1-T01M0>≠01	TMOD01 <T01M1-T01M 0>=01
00	Comparator output from Timer 0	Overflow output from Timer 0
01	φT1	
10	φT16	
11	φT256	(16-Bit Timer Mode)

PWM cycle selection

00	reserved
01	(2^6-1) × clock source
10	(2^7-1) × clock source
11	(2^8-1) × clock source

Timers 01 operation mode selection

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

Note : When setting the TI0 pin, first set the Port C setting, then TMOD01.

Figure 3.7.9 Register for 8-bit Timers (TMOD01)

Timers 23 Mode Register

TMOD23 (008CH)

	7	6	5	4	3	2	1	0
Bit Symbol	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
Read/Writ	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-Bit Timer Mod 01: 16-Bit Timer Mode 10: 8-Bit PPG Mod 11: 8-Bit PWM Mod		PWM cycle 00: reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		Source clock for Timer 3 00: T2TRG 01: φT1 10: φT16 11: φT256		Source clock for Timer 2 00: reserved 01: φT1 10: φT4 11: φT16	

Timer 2 source clock selection

00	Do not set
01	φT1 (prescaler)
10	φT4 (prescaler)
11	φT16 (prescaler)

Timer 3 source clock selection

	TMOD23 <T23M1~T23M0>≠01	TMOD23 <T23M1~T23M0>=01
00	Comparator output from Timer 2	Overflow output from Timer 2
01	φT1	
10	φT16	
11	φT256	(16-Bit Timer Mode)

PWM cycle selection

00	reserved
01	$(2^6\text{-}1) \times$ clock source
10	$(2^7\text{-}1) \times$ clock source
11	$(2^8\text{-}1) \times$ clock source

Timers 23 operation mode selection

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (Timer 2) + 8-bit timer (Timer 3)

Figure 3.7.10 Register for 8-bit Timers (TMOD23)

Timers 45 Mode Register

TMOD45 (0094H)

	7	6	5	4	3	2	1	0
Bit symbol	T45M1	T45M0	PWM41	PWM40	T5CLK1	T5CLK0	T4CLK1	T4CLK0
Read/Writ	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-Bit Timer Mod 01: 16-Bit Timer Mode 10: 8-Bit PPG Mod 11: 8-Bit PWM Mod		PWM cycle 00: reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		Source clock for Timer 5 00: T4TRG 01: φT1 10: φT16 11: φT256		Source clock for Timer 4 00: TI4 pin (Note) 01: φT1 10: φT4 11: φT16	

Source clock for Timer 4

00	TI4 (external input)
01	φT1 (prescaler)
10	φT4 (prescaler)
11	φT16 (prescaler)

Source clock for Timer 5

	TMOD45 <T45M1-T45M0>≠01	TMOD45 <T45M1-T45M0>=01
00	Comparator output from Timer 4	Overflow output from Timer 4
01	φT1	
10	φT16	
11	φT256	(16-Bit Timer Mode)

PWM cycle

00	reserved
01	(2^6-1) × clock source
10	(2^7-1) × clock source
11	(2^8-1) × clock source

Operation mode for Timers 45

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (Timer 4) + 8-bit timer (Timer 5)

Note : When setting the TI4 pin, first set the Port C setting, then TMOD45.

Figure 3.7.11 Register for 8-bit Timers (TMOD45)

Timers 67 Mode register

TMOD67 (009CH)	7	6	5	4	3	2	1	0
Bit symbol	T67M1	T67M0	PWM61	PWM60	T7CLK1	T7CLK0	T6CLK1	T6CLK0
Read/Writ	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-Bit Timer Mod 01: 16-Bit Timer Mode 10: 8-Bit PPG Mod 11: 8-Bit PWM Mod		PWM cycle 00: reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		Source clock for Timer 7 00: T6TRG 01: φT1 10: φT16 11: φT256		Source clock for Timer6 00: reserved 01: φT1 10: φT4 11: φT16	

Source clock for Timer 6

00	Do not set
01	φT1 (prescaler)
10	φT4 (prescaler)
11	φT16 (prescaler)

Source clock for Timer 7

	TMOD67 <T67M1-T67M0>≠01	TMOD67 <T67M1-T67M0>=01
00	Comparator output from Timer 6	Overflow output from Timer 6
01	φT1	
10	φT16	
11	φT256	(16-Bit Timer Mod

PWM cycle

00	reserved
01	(2^6-1) × clock source
10	(2^7-1) × clock source
11	(2^8-1) × clock source

Operation mode for Timers 67

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (Timer 6) + 8-bit timer (Timer 7)

Figure 3.7.12 Register for 8-bit Timers (TMOD67)

Timer 1 Flip-Flop Control Register

		7	6	5	4	3	2	1	0
TFFCR1 (0085H)	Bit symbol					TFF1C1	TFF1C0	TFF1IE	TFF1IS
	Read/Writ					R/W			
	After Reset					1	1	0	0
Read Modify-Write instructions are prohibited.	Function					00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		TFF1 Control for inversion 0: Disable 1: Enabl	TFF1 Inversion select 0: Timer 0 1: Timer 1

Inverse signal for Timer Flop-Flop 1 (TFF1)
(Don't care except in 8-Bit Timer Mode)

0	Inversion by Timer 0
1	Inversion by Timer 1

Inversion of TFF1

0	Disabled
1	Enabled

Control of TFF1

00	Inverts the value of TFF1
01	Sets TFF1 to 1
10	Clears TFF1 to 0
11	Don't care

Note: The values of bits 4 to 6 of TFFCR1 are undefined when read.

Figure 3.7.13 Register for 8-bit Timers (TFFCR1)

Timer 3 Flip-Flop Control Register

		7	6	5	4	3	2	1	0
TFFCR3 (008DH)	Bit symbol					TFF3C1	TFF3C0	TFF3IE	TFF3IS
	Read/Writ					R/W			
	After Reset					1	1	0	0
Read Modify-Write instructions are prohibited.	Function					00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don't care		TFF3 Control for inversion 0: Disable 1: Enabl	TFF3 Inversion select 0: Timer 2 1: Timer 3

Inverse signal for Timer Flip-Flop 3 (TFF3)
(Don't care except in 8-Bit Timer Mode)

0	Inversion by Timer 2
1	Inversion by Timer 3

Inversion of TFF3

0	Disabled
1	Enabled

Control of TFF3

00	Inverts the value of TFF3
01	Sets TFF3 to 1
10	Clears TFF3 to 0
11	Don't care

Note: The values of bits 4 to 6 of TFFCR3 are undefined when read.

Figure 3.7.14 Register for 8-bit Timers (TFFCR3)

Timer 5 Flip-Flop Control Register

TFFCR5 (0095H)	7	6	5	4	3	2	1	0
Bit symbol					TFF5C1	TFF5C0	TFF5IE	TFF5IS
Read/Writ					R/W			
After Reset					1	1	0	0
Function					00: Invert TFF5 01: Set TFF5 10: Clear TFF5 11: Don't care		TFF5 Control for inversion 0: Disable 1: Enabl	TFF5 Inversion select 0: Timer 4 1: Timer 5

Read Modify-Write instructions are prohibited.

Inverse signal for Timer Flip-Flop 5 (TFF5)
(Don't care except in 8-Bit Timer Mode)

0	Inversion by Timer 4
1	Inversion by Timer 5

Inversion of TFF5

0	Disabled
1	Enabled

Control of TFF5

00	Inverts the value TFF5
01	Sets TFF5 to 1
10	Clears TFF5 to 0
11	Don't care

Note: The values of bits 4 to 6 of TFFCR5 are undefined when read.

Figure 3.7.15 Register for 8-bit Timers (TFFCR5)

Timer 7 Flip-Flop Control Register

		7	6	5	4	3	2	1	0
TFFCR7 (009DH)	Bit symbol					TFF7C1	TFF7C0	TFF7IE	TFF7IS
	Read/Writ					R/W			
	After Reset					1	1	0	0
Read Modify-Write instructions are prohibited.	Function					00: Invert TFF7 01: Set TFF7 10: Clear TFF7 11: Don't care		TFF7 Control for invertsion 0: Disable 1: Enabl	TFF7 Inversion select 0: Timer 6 1: Timer 7

Inverse signal for Timer Flip-Flop 7 (TFF7)
(Don't care except in 8-Bit Timer Mode)

0	Inversion by Timer 6
1	Inversion by Timer 7

Inversion of TFF7

0	Disabled
1	Enabled

Control of TFF7

00	Inverts the value of TFF7
01	Sets TFF7 to 1
10	Clears TFF7 to 0
11	Don't care

Note: The values of bits 4 to 6 of TFFCR7 are undefined when read.

Figure 3.7.16 Register for 8-bit Timers (TFFCR7)

Timer Register (TREG 0 to 7)

Symbol	Address	7	6	5	4	3	2	1	0
TREG0	82H	-							
		W							
		undefined							
TREG1	83H	-							
		W							
		undefined							
TREG2	8AH	-							
		W							
		undefined							
TREG3	8BH	-							
		W							
		undefined							
TREG4	92H	-							
		W							
		undefined							
TREG5	93H	-							
		W							
		undefined							
TREG6	9AH	-							
		W							
		undefined							
TREG7	9BH	-							
		W							
		Undefined							

Figure 3.7.17 Register for 8-bit Timers (TREG0 to TREG7

3.7.4 Operation in each mode

(1) 8-Bit Timer Mode

Both timer 0 and timer 1 can be used independently as 8-bit interval timers.

① Generating interrupts at a fixed interval (using timer 1)

To generate interrupts at constant intervals using timer 1 (INTT1), first stop timer 1, then set the operation mode and input clockusing the TMOD01 register and the counter value using the TREG1 registers. Then enable the interrupt INTT1 and start timer 1 counting.

Example: To generate an INTT1 interrupt every 4 μseconds at fc = 20 MHz, set each register as follows:

	MSB 7	6	5	4	3	2	1	LSB 0	
TRUN01	← –	X	X	X	–	–	0	–	Stop timer 1 and clear it to 0.
TMOD01	← 0	0	X	X	0	1	–	–	Select 8-Bit Timer Mode and select φT1 (0.4 μs at fc = 20 MHz) as the input clock.
TREG1	← 0	1	1	0	0	1	0	0	Set TREG1 to 40 μs ÷ φT1 = 100 = 64H
INTET01	← X	1	0	1	–	–	–	–	Enable INTT1 and set it to Level 5.
TRUN01	← –	X	X	X	–	1	1	–	Start timer 1 counting.

Note: X = Don't care; "–" = No change

Select the input clock using Table 3.7 2.

Table 3.7.2 Selecting Interrupt Interval and the Input Clock Using 8-Bit Timer

Input Clock	Interrupt Interval (at fc = 20 MHz)	Resolution
φT1 (8/fc)	0.4 μs to 102.4 μs	0.4 μs
φT4 (32/fc)	1.6 μs to 409.6 μs	1.6 μs
φT16 (128/fc)	6.4 μs to 1.639 ms	6.4 μs
φT256 (2048/fc)	102.4 μs to 26.22 ms	102.4 μs

Note: The input clocks for timer 0 and timer 1 differ as follows:
timer 0: Uses timer 0 input (TI0) and can be selected from φT1, φT4 or φT16
timer 1: Match output of timer 0 (T0TRG) and can be selected from φT1, φT16 or φT256

② Generating a 50% duty ratio square wave pulse

The state of the timer flip-flop (TFF1) is inverted at constant intervals and its status output via the timer output pin (TO1).

Example: To output a 2.4-μs square wave pulse from the TO1 pin at fc = 20 MHz, use the following procedure to make the appropriate register settings. This example uses timer 1; however, either timer 0 or timer 1 may be used.

		7	6	5	4	3	2	1	0	
TRUN01	←	-	X	X	X	-	-	0	-	Stop timer 1 and clear it to 0.
TMOD01	←	0	0	X	X	0	1	-	-	Select 8-Bit Timer Mode and select φT1 (0.4 μs at fc = 20 MHz) as the input clock.
TREG1	←	0	0	0	0	0	0	1	1	Set the timer register to 2.4 μs ÷ φT1 ÷ 2 = 3
TFFCR1	←	X	X	X	X	1	0	1	1	Clear TFF1 to 0 and set it to invert on the match detect signal from timer 1.
PCCR	←	X	X	-	-	-	-	1	-	Set PC1 to function as the TO1 pin.
PCFC	←	X	X	-	-	-	-	1	-	
TRUN01	←	-	X	X	X	-	1	1	-	Start timer 1 counting.

Note: X = Don't care; "–" = No change

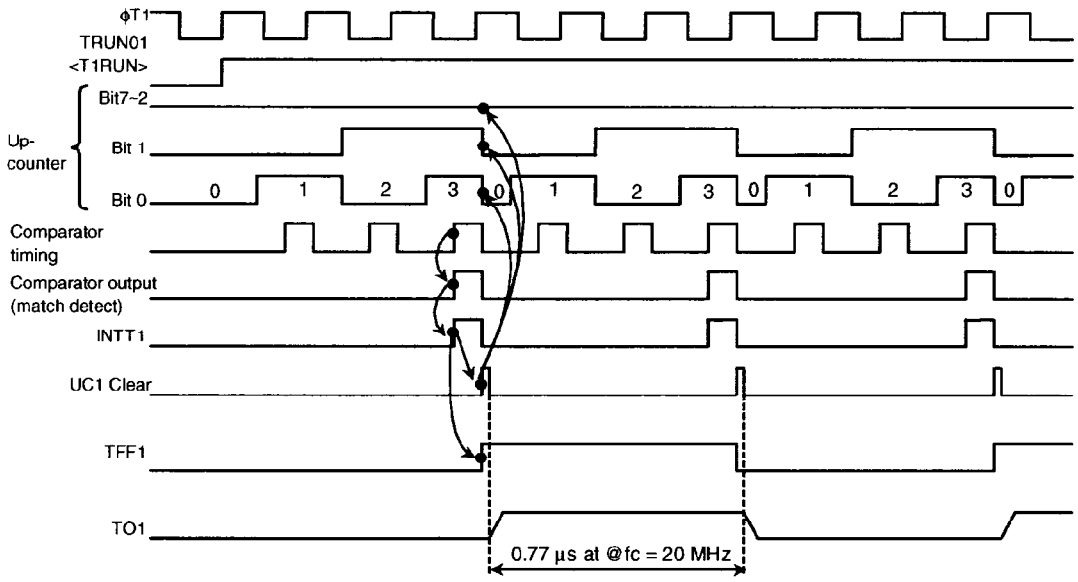


Figure 3.7.18 Square wave output timing chart (50% Duty)

③ Making timer 1 count up on the match signal from the timer 0 comparator

Select 8-Bit Timer Mode and set the comparator output from timer 0 to be the input clock to timer 1.

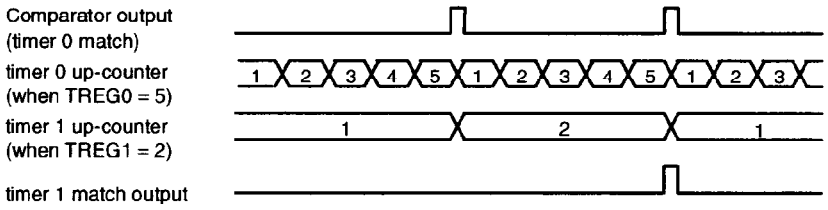


Figure 3.7.19 Timer 1 Count Up on Signal from Timer 0

(2) 16-Bit Timer Mode

The two 8-bit timers timer 0 and timer 1 can be cascaded together to comprise a 16-bit interval timer.

To make a 16-bit interval timer in which timer 0 and timer 1 are cascaded together, set TMOD01 <T01M1,T01M0> to 01.

In 16-Bit Timer Mode, the overflow output from timer 0 is used as the input clock for timer 1, regardless of the value set in TMOD01<T1CLK1,T1CLK0>. Table 3.7 4(1) shows the relationship between the timer (interr upt) cycle and the input clock selection.

To set the timer interrupt interval, set the lower eight bits in timer register TREG0 and the upper eight bits in TREG1. Be sure to set TREG0 first (as entering data in TREG0 temporarily disables the compare, while entering data in TREG1 starts the compare).

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

If ϕT16 (6.4 μs at 20 MHz) is used as the input clock for counting, setthe following value in the registers: 0.4 s ÷ 6.4 μs = 62500 = F424H: i.e. set TREG1 to F4H and TREG0 to 24H.

The comparator's match-detect signal is output from timer 0 each time the up-counter UC0 matches TREG0; however, the much does not clear the up-counter UC0. In the case of the timer 1 comparator, the match detect signal is output on each comparator pulse on which the values in the up-counter UC1 and TREG1 match. When the match detect signal is output simultaneously from both the comparators timer 0 and timer 1, the up-counters UC0 and UC1 are cleared to 0 and the interrupt INTT1 is generated. Also, if inversion is enabled, the value of the timer flip-flop TFF1 is inverted.

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

Figure 3.7.20 Timer output by 16-Bit Timer Mode

(3) 8-Bit PPG (Programmable Pulse Generation) Output Mode

Square wave pulses can be generated at any frequency and duty ratio by timer 0. The output pulses may be active-Low or active-High. In this mode timer 1 cannot be used.

Timer 0 outputs pulses on the TO1 pin (which can also be used as PC1).

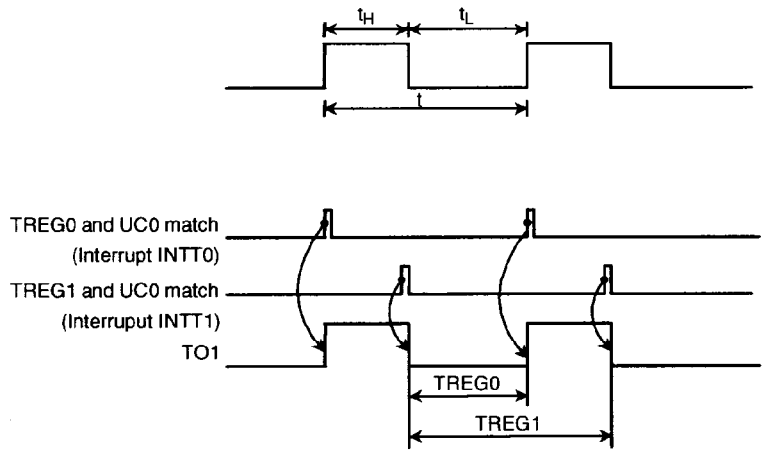


Figure 3.7.21 8 bit PPG output waveforms

In this mode a programmable square wave is generated by inverting the timer output each time the 8-bit up-counter (UC0) matches the value in one of the timer registers TREG0 or TREG1.
The value set in TREG0 must be smaller than the value set in TREG1.
Although the up-counter for timer 1 (UC1) is not used in this mode. TRUN01<T1RUN> should be set to 1 so that UC1 is set for counting.
Figure 3.7.22 shows a block diagram representing this mode.

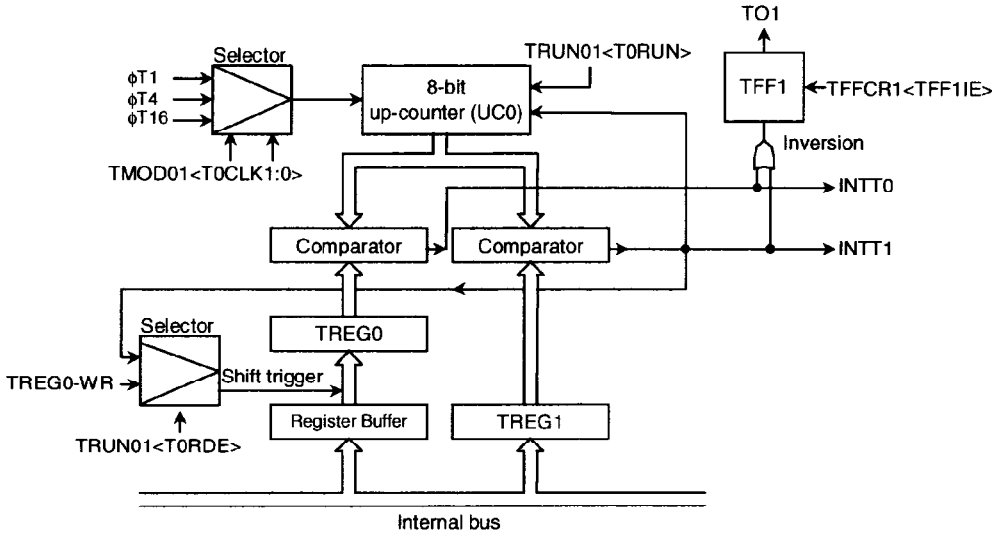


Figure 3.7.22 Block diagram of 8-Bit PPG Output Mode

If the TREG0 double buffer is enabled in this mode, the value of the register buffer will be shifted into TREG0 each time TREG1 matches UC0.
Use of the double buffer facilitates the handling of low-duty waves (when duty is varied).

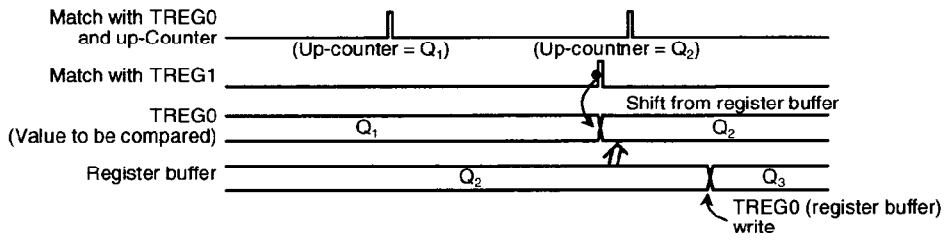
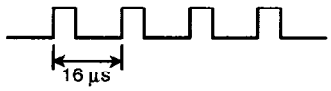


Figure 3.7.23 Operation of register buffer

Example: To generate 1/4-duty 62.5 kHz pulses (at fc = 20 MHz):



Calculate the value which should be set in the timer register.

To obtain a frequency of 62.5 kHz, the pulse cycle t should be: t = 1/62.5 kHz = 16 μ s

ϕT1 = 0.4 µs (at 20 MHz):

16 µs ÷ 0.4 µs = 40

Therefore set TREG1 to 40 (28H)

The duty is to be set to 1/4: t × 1/4 = 16 µs × 1/4 = 4 µs

4 µs ÷ 0.4 µs = 10

Therefore, set TREG0 = 10 = 0AH.

		7	6	5	4	3	2	1	0	
TRUN01	←	0	X	X	X	-	0	0	0	Stop timer 0 and timer 1 and clear it to "0".
TMOD01	←	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	0	1	0	0	0	Write 28H
TFFCR1	←	X	X	X	X	0	1	1	X	Set TFF1, enabling both inversion and the double buffer.
										→ 10 generates a negative logic pulse.
PCCR	←	X	X	-	-	-	-	1	-	Set PC1 as the TO1 pin.
PCFC	←	X	X	-	-	-	-	1	-	
TRUN01	←	1	X	X	X	-	1	1	1	Start timer 0 and timer 1 counting.

Note: X = Don't care; "–" = No change

(4) 8-Bit PWM Output Mode

This mode is only valid for timer 0. In this mode, a PWM pulse with the maximum resolution of 8 bits can be output.

When timer 0 is used the PWM pulse is output on the TO1 pin (which is also used as PC1). Timer 1 can also be used as an 8-bit timer.

The timer output is inverted when the up-counter (UC0) matches the value set in the timer register TREG0 or when 2^n-1 counter overflow occurs (n = 6, 7 or 8 as specified by TMOD01<PWM01~PWM00>). The up-counter UC0 is cleared when 2^n-1 counter overflow occurs.

The following conditions must be satisfied before this PWM mode can be used.

Value set in TREG0 < value set for 2^n-1 counter overflow

Value set in TREG0 ≠ 0

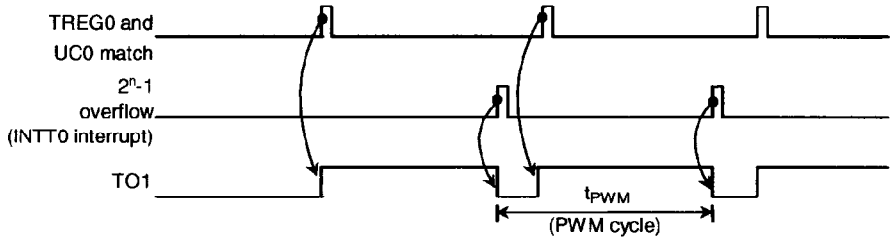


Figure 3.7.24 8-bit PWM waveforms

Figure 3.7.25 shows a block diagram representing this mode.

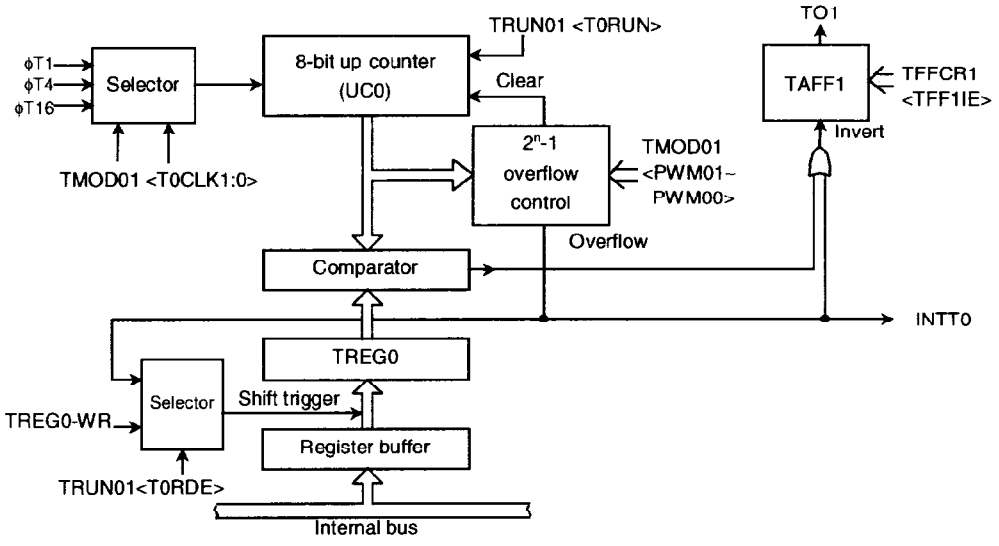


Figure 3.7.25 Block diagram of 8-Bit PWM Mode

In this mode the value of the register buffer will be shifted into TREG0 if $2^n - 1$ overflow is detected when the TREG0 double buffer is enabled.
Use of the double buffer facilitates the handling of low duty ratio waves.

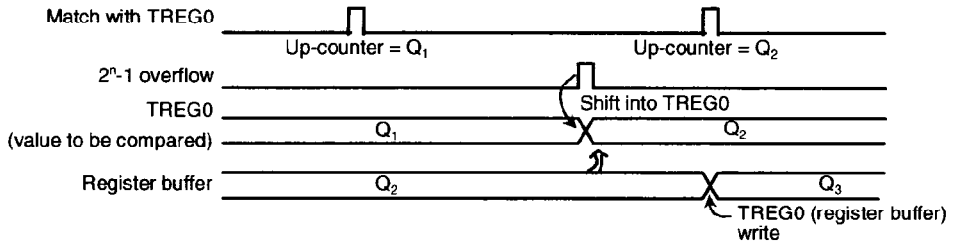
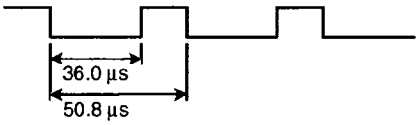


Figure 3.7.26 Register buffer operation

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



To achieve a 50.8-µs PWM cycle by setting φT1 to 0.4 µs (at fc = 20 MHz):

$50.8\ \mu s \div 0.4\ \mu s = 127$

$2^n - 1 = 127$

Therefore n should be set to 7.

Since the low-level period is 36.0 µs when φT1 = 0.4 µs,

set the following value for TREG0:

$36.0\ \mu s \div 0.4\ \mu s = 90 = 5AH$

	MSB 7	6	5	4	3	2	1	LSB 0	
TRUN01	← –	X	X	X	–	–	–	0	Stop timer 0 and clear it to 0.
TMOD01	← 1	1	1	0	–	–	0	1	Select 8-Bit PWM Mode (cycle: 2^7-1) and select φT1 as the input clock.
TREG0	← 0	1	0	1	1	0	1	0	Write 5AH.
TFFCR1	← X	X	X	X	1	0	1	X	Clear TFF1 to 0, enable the inversion and double buffer.
PCCR	← X	X	–	–	–	–	1	–	Set PC1 and the TO1 pin.
PCFC	← X	X	–	–	–	–	1	–	
TRUN01	← 1	X	X	X	–	1	–	1	Start timer 0 counting.

Note: X = Don't care; "–" = No change

Table 3.7.3 PWM cycle

	PWM Interval (at fc = 20MHz)		
	φT1	φT4	φT16
$2^6 - 1$	25.2 μs (39.7 kHz)	100.8 μs (9.92 kHz)	403.2 μs (2.48 kHz)
$2^7 - 1$	50.8 μs (19.6 kHz)	203.2 μs (4.92 kHz)	810 μs (1.23 kHz)
$2^8 - 1$	102 μs (9.80 kHz)	408 μs (2.45 kHz)	1.63 ms (0.61 kHz)

(5) Settings for each mode

Table 3.7.4 shows the SFR settings for each mode.

Table 3.7.4 Timer mode setting registers

Register name	TMOD01				TFFCR1
<Bit Symbol>	<T01M1:0>	<PWM01:00>	<T1CLK1:0>	<T0CLK1:0>	<TFF1IS>
Function	Timer mode	PWM cycle	Upper timer input clock	Lower timer input clock	Timer F/F invert signal select
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: timer0 output 1: timer1 output
16-bit timer mod	01	–	–	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	–
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

Note:"–" = Don't care

3.8 16-Bit Timer/Event Counters

The TMP92CW10 incorporates two multifunctional 16-bit timer/event counters (timer 8 and timer A) which have the following operation modes:

- 16-Bit Interval Timer Mode
- 16-Bit Event Counter Mode
- 16-Bit Programmable Pulse Generation (PPG) Mode

The capture function enables selection of the following modes.

- Frequency Measurement Mode
- Pulse Width Measurement Mode
- Time Differential Measurement Mode

Figure 3.8.1 and 3.8.2 show block diagrams for timer 8 and timer A.

Each timer/event counter channel consists of a 16-bit up-counter, two 16-bit timer registers (one of them with a double-buffer structure), two 16-bit capture registers, two comparators, a capture input controller, a timer flip-flop and a control circuit.

Each timer/event counter is controlled by an 11-byte control SFR.

The two channels (timer 8 and timer A) can be used independently.

The two channels feature the same operations except for those descrivewd in Table 3.8.1. Thus, only the operation of timer 8 is explained below.

Table 3.8.1 Differences between Timer 8 and Timer A

Specification \ Channel		Timer 8	Timer A
External Pins	External clock / Capture trigger input pins	TI8 (also used as PD0) TI9 (also used as PD1)	TIA (also used as PD4) TIB (also used as PD5)
	Timer flip-flop output pins	TO8 (also used as PD2) TO9 (also used as PD3)	TOA (also used as PD6) TOB (also used as PD7)
SFR (address)	Timer Run Register	TRUN8 (00A0H)	TRUNA (00B0H)
	Timer Mode Register	TMOD8 (00A2H)	TMODA (00B2H)
	Timer Flip-Flop Control Register	TFFCR8 (00A3H)	TFFCRA (00B3H)
	Timer Register	TREG8L (00A8H) TREG8H (00A9H) TREG9L (00AAH) TREG9H (00ABH)	TREGAL (00B8H) TREGAH (00B9H) TREGBL (00BAH) TREGBH (00BBH)
	Capture Register	CAP8L (00ACH) CAP8H (00ADH) CAP9L (00AEH) CAP9H (00AFH)	CAPAL (00BCH) CAPAH (00BDH) CAPBL (00BEH) CAPBH (00BFH)

3.8.1 Block diagrams

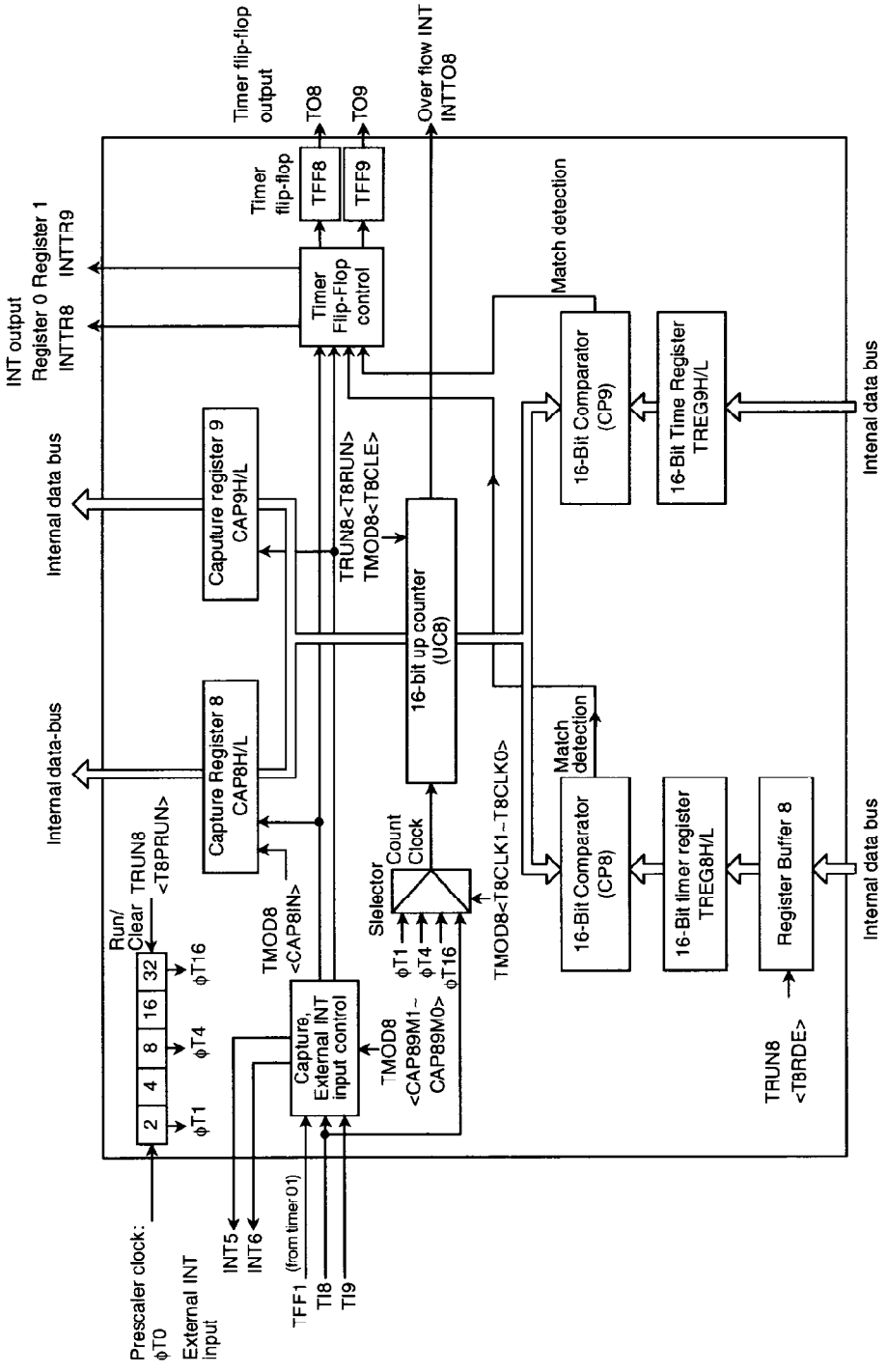


Figure 3.8.1 Block Diagram of Timer 8

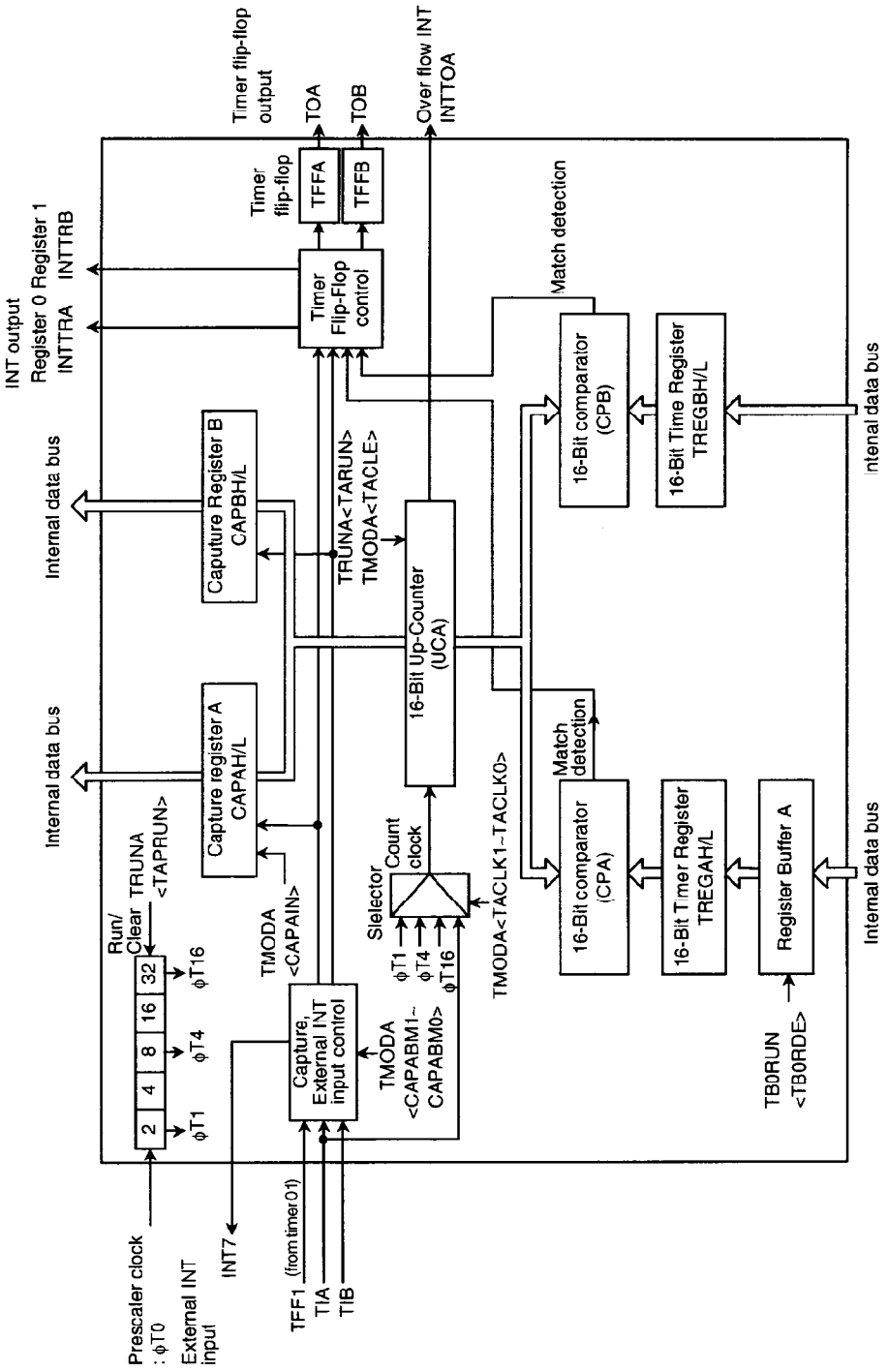


Figure 3.8.2 Block diagram of Timer A

3.8.2 Operation of each block

(1) Prescaler

The 5-bit prescaler generates the source clock for Timer 8. The prescaler clock (ϕ T0) is generated by dividing fc by 4.

The prescaler can be started or stopped using TRUN8<T8PRUN>. Counting starts when <T8PRUN> is set to 1: the prescaler is cleared to zero and stops operation when <T8PRUN> is set to 0.

Table 3.8.2 shows the prescaler clock resolution.

Table 3.8.2 Prescaler clock resolution

Input clock	Interval (@ fc = 20MHz)
ϕT1	0.4 µs
ϕT4	1.6 µs
ϕT16	6.4 µs

(2) Up-counter (UC8)

UC8 is a 16-bit binary counter which counts up pulses input from the clock specified by TMOD8 <T8CLK1,T8CLK0>.

Any one of the prescaler internal clocks ϕT1, ϕT4 and ϕT16 or an external clock input via the TI8 pin can be selected as the input clock. Counting or stopping & clearing of the counter is controlled by TRUN8<T8RUN>.

When clearing is enabled, the up-counter UC8 will be cleared to zero each time its value matches the value in the timer register TREG9H/L. Clearing can be enabled or disabled using TMOD8<T8CLE>.

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

A Timer Overflow interrupt (INTTO8) is generated when UC8 overflow occurs.

(3) Timer registers (TREG8H/L and TREG9H/L)

These two 16-bit registers are used to set the interval time. When the value in the up-counter UC8 matches the value set in this timer register, the comparator match - detect signal will output.

The timer registers TREG8H/L and TREG9H/L can be set using either a 2-byte data transfer instruction or two 1-byte data transfer instructions, the first to set the lower 8 bits and second to set the upper 8 bits.

The TREG8 timer register has a double-buffer structure, and is paired with Register Buffer 8. The value set in TRUN8<T8RDE> determines whether the double-buffer structure is enabled or disabled: it is disabled when <T8RDE> = 0, and enabled when <T8RDE> = 1.

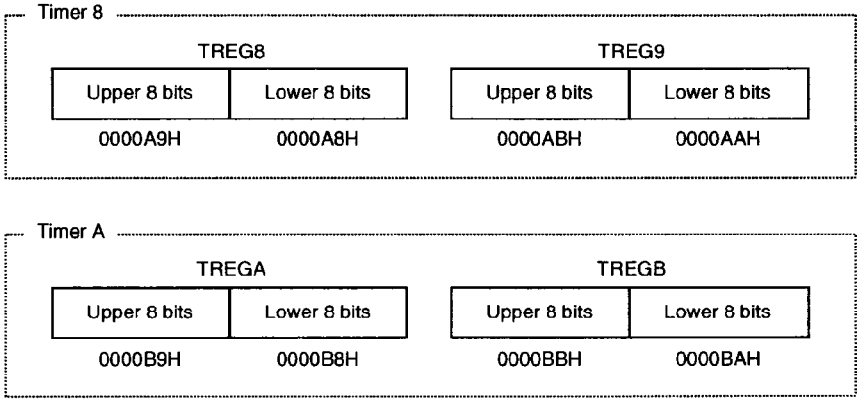
When the double buffer is enabled, data is transferred from the register buffer to the timer register when the values in the up-counter (UC8) and the timer register TREG9 match.

After a reset, TR EG8 and TREG9 are undefined. If the 16-bit timer is to be used after a reset, data should be written to it beforehand.

On a Reset <T8RDE> is initialized to 0, disabling the double buffer. To use the double buffer, write data to the timer register, set <T8RDE> to 1, then write data to the register buffer as shown below.

TREG8 and the register buffer both have the same memory addresses (0000A8H and 0000A9H) allocated to them. If <T8RDE> = 0, values are written to both the timer register and the register buffer. If <T8RDE> = 1, the values are written to the register buffer only.

The addresses of the Timer Registers are as follows:

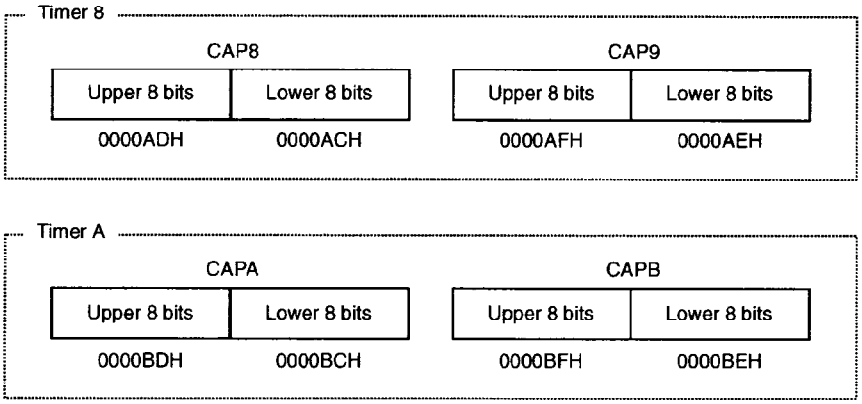


The Timer Registers are write-only registers and thus cannot be read.

(4) Capture Registers (CAP8H/L and CAP9H/L)

These 16-bit registers are used to latch the values in the up-counter UC8. Data in the Capture Registers should be read using a 2-byte data load instruction or two 1-byte data load instructions. The least significant byte is read first, followed by the most significant byte.

The addresses of the Capture Registers are as follows:



The Capture Registers are read-only registers and thus cannot be written to.

(5) Capture input control and external interrupt control

This circuit controls the timing for latching the value of up-counter UC8 into CAP8, for controlling CAP8 and for generating external interrupts. The latch timing for capture register and the edge selection for external interrupts are determined by TMOD8<CAP89M1,CAP89M0>.

The external interrupt INT6 is always generated on the rising edge.

In addition, the value in the up-counterUC8 can be loaded into a capture register by software. Whenever 0 is written to TMOD8<CAP8IN>, the current value in the up-counter UC8 is loaded into the capture register CAP8. It is necessary to keep the prescaler in Run Mode (i.e. TRUN8<T8PRUN> must be held at a value of 1).

(6) Comparators (CP8 and CP9)

CP8 and CP9 are 16-bit comparators which compare the value in the up-counter UC8 with the value set in TREG8 or TREG9 respectively, in order to detect a match. If a match is detected, the comparator generates an interrupt INTTR8 or INTTR9 respectively.

(7) Timer flip-flops (TFF8 and TFF9)

These flip-flops are inverted by the match detect signals from the comparators and the latch signals to the Capture Registers. Inversion can be enabled and disabled for each element using TFFCR8<CAP9T8,CAP8T8,EQ9T8,EQ8T8>. After a reset the value of TFF8 and TFF9 are undefined. If 00 is written to TFFCR8<TFF8C1,TFF8C0> or <TFF9C1,TFF9C0>, TFF8 or TFF9 will be inverted. If 01 is written to the capture registers, the value of TFF8 or TFF9 will be set to 1. If 10 is written to the capture registers, the value of TFF8 or TFF9 will be set to 0.

The values of TFF8 and TFF9 can be output via the Timer Output pins TO8 and TO9 (which also function as PD2 and PD3 respectively). The timer output should be specified using the Port 8 SFR.

3.8.3 SFR

Timer 8 Run Register

TRUN8 (00A0H)

	7	6	5	4	3	2	1	0
Bit symbol	T8RDE	–	–	–	I2T8	T8PRUN	–	T8RUN
Read/Write	R/W	R/W			R/W	R/W		R/W
After Reset	0	0	–	–	0	0	–	0
Function	Doubl Buffer 0: Disable 1: Enable	Write 0			IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		

Count operation

0	Stop and Clear
1	Count

I2T8: Operation during IDLE2-mod
T8PRUN: Operation of prescaler
T8RUN: Operation of Timer 8

Note: The 1, 4 and 5 of TRUN8 are read as underfined value.

Timer A Run Register

TRUNA (00B0H)

	7	6	5	4	3	2	1	0
Bit symbol	TARDE	–	–	–	I2TA	TAPRUN	–	TARUN
Read/Write	R/W	R/W			R/W	R/W		R/W
After Reset	0	0	–	–	0	0	–	0
Function	Doubl Buffer 0: Disable 1: Enable	Write 0			IDLE2 0: Stop 1: Operate	16 Bit Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		

Count Operation

0	Stop and Clear
1	Count

I2TA: Operation during IDLE2-mod
TAPRUN: Operation of prescaler
TARUN: Operation of Timer A

Note: The 1, 4 and 5 of TRUNA are read as underfined value.

Figure 3.8.3 Registers for 16-bit Timers (TRUN8, TRUNA)

Timer 8 Mode Register

TMOD8 (00A2H)

	7	6	5	4	3	2	1	0
Bit symbol	CAP9T9	EQ9T9	CAP8IN	CAP89M1	CAP89M0	T8CLE	T8CLK1	T8CLK0
Read/Write	R/W		W*	R/W				
After Reset	0	0	1	0	0	0	0	0
Function	TFF9 inversion 0: Disable trigger 1: Enable trigger		Execute software capture 0: Execute 1: Don't care	Capture timing 00: Disable INT5 occurs on rising edge. 01: TI8 ↑ TI9 ↑ INT5 occurs on rising edge. 10: TI8 ↑ TI8 ↓ INT5 occurs on falling edge. 11: TFF1 ↑ TFF1 ↓ INT5 occurs on rising edge.		Control up-counter 0. Disable clearing 1: Enable clearing	Timer 8 source clock 00: TI8 pin 01: φT1 10: φT4 11: φT16	
	Invert when the UC value is captured to CAP9.	Invert when the UC value matches the value in TREG9.						

Timer 8 source clock

00	External input clock (TI8 input)
01	φT1
10	φT4
11	φT16

Up-counter (UC8) clear control

0	Disable
1	Enable clearing by match with TREG9.

Capture/Interrupt timing

	Capture control	INT5 contro
00	Disable	INT5 occurs on rising edge of TI8.
01	CAP8 at TI8 rise CAP9 at TI9 rise	
10	CAP8 at TI8 rise CAP9 at TI8 fall	INT5 occurs on falling edge of TI8.
11	CAP8 at TFF1 rise CAP9 at TFF1 fall	INT5 occurs on rising edge of TI8.

Software capture

0	The value in the up-counter is captured to CAP8.
1	Don't care

Figure 3.8.4 Registers for 16-bit Timers (TMOD8)

Timer A Mode Register

TMODA (00B2H)

	7	6	5	4	3	2	1	0
Bit symbol	CAPBTB	EQBTB	CAPAIN	CAPABM1	CAPABM0	TACLE	TACLK1	TACLK0
Read/Write	R/W		W*	R/W				
After Reset	0	0	1	0	0	0	0	0
Function	TFFB inversion 0: Disable trigger 1: Enable trigger		Execute software capture 0: Execute 1: Don't care	Capture timing 00: Disable INT7 occurs on rising edge. 01: TIA ↑ TIB ↑ INT7 occurs on rising edge. 10: TIA ↑ TIA ↓ INT7 occurs on falling edge. 11: TFF1 ↑TFF1 ↓ INT7 occurs on rising edge.		Control up-counter 0: Disable clearing 1: Enable clearing	Timer A source clock 00: TIA pin 01: φT1 10: φT4 11: φT16	
	Invert when the UC value is captured to CAPB.	Invert when the UC value matches the value in TREGB.						

Timer A source clock

00	External input clock (TIA input)
01	φT1
10	φT4
11	φT16

Up-counter (UCA) clear control

0	Disable
1	Enable clearing on match with TREGB.

Capture/Interrupt timing

	Capture control	INT7 contro
00	Disable	INT7 occurs on rising edge of TIA.
01	CAPA at TIA rise CAPB at TIB rise	
10	CAPA at TIA rise CAPB at TIA fall	INT7 occurs on falling edge of TIA.
11	CAPA at TFF1 rise CAPB at TFF1 fall	INT7 occurs on rising edge of TIA.

Software capture

0	The value in the up-counter is captured to CAPA.
1	Don't care

Figure 3.8.5 Registers for 16-bit Timers (TMODA

Timer 8 Flip-Flop Control Register

TFFCR8 (00A3H)

	7	6	5	4	3	2	1	0
Bit symbol	TFF9C1	TFF9C0	CAP9T8	CAP8T8	EQ9T8	EQ8T8	TFF8C1	TFF8C0
Read/Write	W*		R/W				W*	
After Reset	1	1	0	0	0	0	1	1
Function	Control TFF9 00: Invert 01: Set 10: Clear 11: Don't care * Always read as 11		TFF8 inversion trigger 0: Disable trigger 1: Enable trigger				Control TFF8 00: Invert 01: Set 10: Clear 11: Don't care * Always read as 11	
			Invert when the UC value is loaded in to CAP9.	Invert when the UC value is loaded in to CAP8.	Invert when the UC value matches the value in TREG9.	Invert when the UC value matches the value in TREG8.		

TFF8 control

00	Invert
01	Set to 1
10	Clear to 0
11	Don't care

TFF8 Inverted when the UC value is matched to TREG8.

0	Disable trigger (disable inversion)
1	Enable trigger (enable inversion)

TFF8 Inverted when the UC value is match d to TREG9.

0	Disable trigger (disable inversion)
1	Enable trigger (enable inversion)

TFF8 Inverted when the UC value is loaded to CAP8.

0	Disable trigger (disable inversion)
1	Enable trigger (enable inversion)

TFF8 Inverted when the UC value is loaded to CAP9.

0	Disable trigger (disable inversion)
1	Enable trigger (enable inversion)

TFF9 control

00	Invert
01	Set to 1
10	Clear to 0
11	Don't care

Figure 3.8.6 Registers for 16-bit Timers (TFFCR8)

Timer A Flip-Flop Control Register

TFFCRA (00B3H)

	7	6	5	4	3	2	1	0
Bit symbol	TFFBC1	TFFBC0	CAPBTA	CAPATA	EQBTA	EQATA	TFFAC1	TFFAC0
Read/Write	W*		R/W				W*	
After Reset	1	1	0	0	0	0	1	1
Function	Control TFFB 00: Invert 01: Set 10: Clear 11: Don't care * Always read as 11		TFFA inversion trigger 0: Disable trigger 1: Enable trigger				Control TFFA 00: Invert 01: Set 10: Clear 11: Don't care * Always read as 11	
			Invert when the UC value is loaded in to CAPB.	Invert when the UC value is loaded in to CAPA.	Invert when the UC value matches the value in TREGB.	Invert when the UC value matches the value in TREGA.		

TFFA control

00	Invert
01	Set to 1
10	Clear to 0
11	Don't care

TFFA Inverted when the UC value matches to TREGA.

0	Disable trigger (disable inversion)
1	Enable trigger (enable inversion)

TFFA Inverted when the UC value matches to TREGB.

0	Disable trigger (disable inversion)
1	Enable trigger (enable inversion)

TFFA Inverted when the UC value is loaded to CAPA.

0	Disable trigger (disable inversion)
1	Enable trigger (enable inversion)

TFFA Inverted when the UC value is loaded to CAPB.

0	Disable trigger (disable inversion)
1	Enable trigger (enable inversion)

TFFB control

00	Invert
01	Set to 1
10	Clear to 0
11	Don't care

Figure 3.8.7 Registers for 16-bit Timers

Timer Register (Timer8, TimerA)

		7	6	5	4	3	2	1	0
TREG8L (00A8H)	Bit symbol	-							
	Read/Writ	W							
	After reset	Undefined							
TREG8H (00A9H)	Bit symbol	-							
	Read/Writ	W							
	After reset	Undefined							
TREG9L (00AAH	Bit symbol	-							
	Read/Writ	W							
	After reset	Undefined							
TREG9H (00ABH)	Bit symbol	-							
	Read/Writ	W							
	After reset	Undefined							
TREGAL (00B8H)	Bit symbol	-							
	Read/Writ	W							
	After reset	Undefined							
TREGAH (00B9H)	Bit symbol	-							
	Read/Writ	W							
	After reset	Undefined							
TREGBL (00BAH)	Bit symbol	-							
	Read/Writ	W							
	After reset	Undefined							
TREGBH (00BBH)	Bit symbol	-							
	Read/Writ	W							
	After reset	Undefined							

Capture Register (Timer8, TimerA)

		7	6	5	4	3	2	1	0
CAP8L (00ACH)	Bit symbol	-							
	Read/Writ	R							
	After reset	Undefined							
CAP8H (00ADH)	Bit symbol	-							
	Read/Writ	R							
	After reset	Undefined							
CAP9L (00AEH)	Bit symbol	-							
	Read/Writ	R							
	After reset	Undefined							
CAP9H (00AFH)	Bit symbol	-							
	Read/Writ	R							
	After reset	Undefined							
CAPAL (00BCH)	Bit symbol	-							
	Read/Writ	R							
	After reset	Undefined							
CAPAH (00BDH)	Bit symbol	-							
	Read/Writ	R							
	After reset	Undefined							
CAPBL (00BEH)	Bit symbol	-							
	Read/Writ	R							
	After reset	Undefined							
CAPBH (00BFH)	Bit symbol	-							
	Read/Writ	R							
	After reset	Undefined							

Figure 3.8.8 Registers for 16-bit Timers (TREG8L/H to TREGBL/H and CAP8L/H to CAPBL/H)

3.8.4 Operation in each mode

(1) 16-Bit Interval Timer Mode

Generating interrupts at fixed intervals

In this example, the interrupt INTTR9 is set to be generated at fixed intervals. The interval time is set in the timer register TREG9.

```
                    7  6  5  4  3  2  1  0
TRUN8       ←       0  0  X  X  -  0  X  0     Stop Timer 8.
INTET89     ←       X  1  0  0  X  0  0  0     Enable INTTR9 and set Interrupt Level 4. Disable INTTR8.
TFFCR8      ←       1  1  0  0  0  0  1  1     Disable the trigger.
TMOD8       ←       0  0  1  0  0  1  *  *     Select Internal clock for input and
                                  (** = 01, 10, 11)  disable the capture function.
TREG9       ←       *  *  *  *  *  *  *  *     Set the interval time (16 bits).
                    *  *  *  *  *  *  *  *
TRUN8       ←       0  0  X  X  -  1  X  1     Start Timer 8.
```

Note: X = Don't care; "–" = No change

(2) 16-Bit Event Counter Mode

If the external clock (TI8 pin input) is selected as the input clock in 16-Bit Timer Mode, the timer can be used as an event counter. Up counter on the rising edge of TI8 pin input. To read the value of the counter, first perform "software capture" once, then read the captured value.

```
                    7  6  5  4  3  2  1  0
TRUN8       ←       0  0  X  X  -  0  X  0     Stop Timer 8.
PDCR        ←       -  -  -  -  -  -  -  0     Set PD0 to Input Mode.
INTET89     ←       X  1  0  0  X  0  0  0     Enable INTTR9 and set Interrupt Level 4. Disable INTTR8.
TFFCR8      ←       1  1  0  0  0  0  1  1     Disable the trigger.
TMOD8       ←       0  0  1  0  0  1  0  0     Select TI8 as the input clock.
TREG9       ←       *  *  *  *  *  *  *  *     Set the number of counts (16 bits).
                    *  *  *  *  *  *  *  *
TRUN8       ←       0  0  X  X  -  1  X  1     Start Timer 8.
```

Note: X = Don't care; "–" = No change

When the timer is used as an event counter, set the prescaler in Run Mode (by setting TRUN8<T8PRUN> to 1).

(3) 16-Bit Programmable Pulse Generation (PPG) Output Mode

Square wave pulses can be generated at any frequency and duty ratio. The output pulse may be either active-Low or active-High.

IN PPG Mode a match between the valu of the up-counter UC8 and either timer register TREG8 or TREG9 inberts the output value for timer flip-flop TFF8. The TFF8 output valu is output on TO8. In this mode the following conditions must be satisfied.

(Value set in TREG8) < (Value set in TREG9)

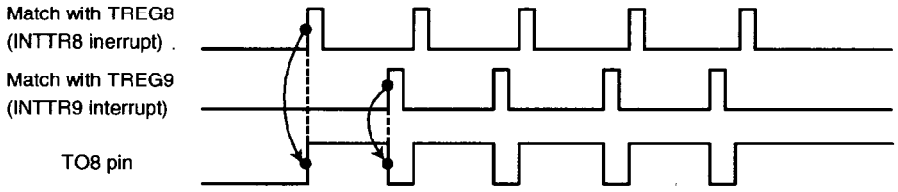


Figure 3.8.9 Programmable Pulse Generation (PPG) Output Waveforms

When the TREG8 double buffer is enabled in this mode, the value of Register Buffer 8 will be shifted into TREG8 when the up-counter value matchesTREG9. This feature facilitates the handling of low-duty waves.

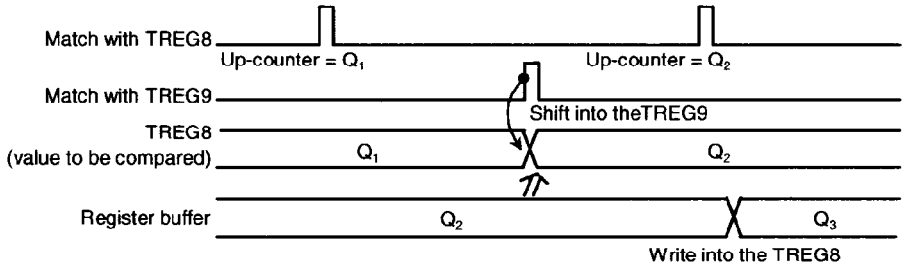


Figure 3.8.10 Operation of Register Buffer

The following block diagram illustrates this mode.

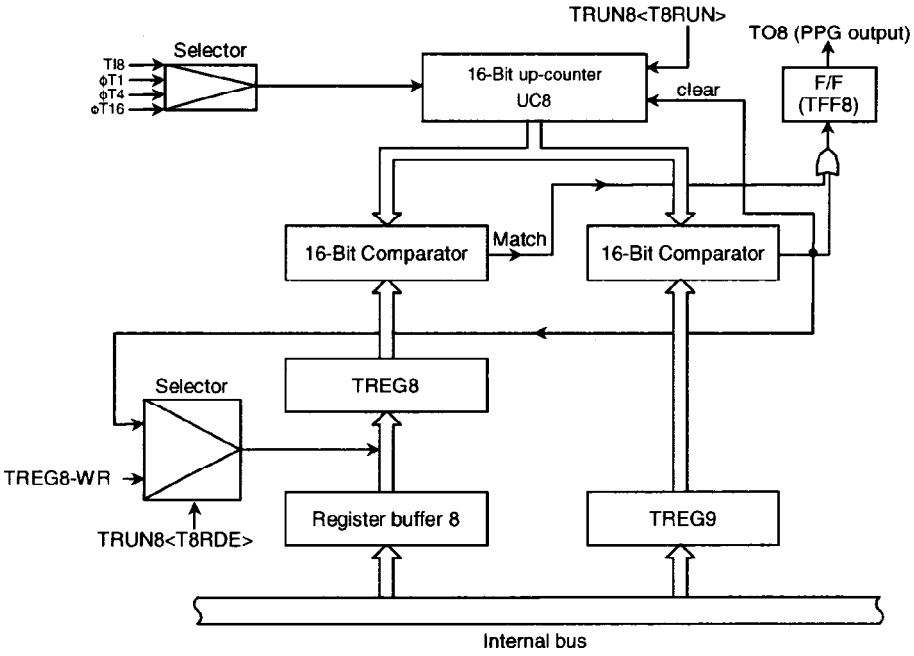


Figure 3.8.11 Block diagram of 16-bit PPG Output Mode

The following example shows how to set 16-Bit PPG Output Mode:

```
                  7  6  5  4  3  2  1  0
TRUN8       ←     0  0  X  X  -  0  X  0      Disable the TREG8 double buffer and stop timer 8.
TREG8       ←     *  *  *  *  *  *  *  *      Set the duty ratio (16 bits).
                  *  *  *  *  *  *  *  *
TREG9       ←     *  *  *  *  *  *  *  *      Set the frequency (16 bits).
                  *  *  *  *  *  *  *  *
TRUN8       ←     1  0  X  X  -  0  X  0      Enable the TREG8 double buffer.
                                               (The duty and frequency are changed on an INTTR9
                                               interrupt.)
TFFCR8      ←     X  X  0  0  1  1  1  0      Set the mode to invert TFF8 at the match with
                                               TREG8/TREG9. Set TFF8 to 0.
TMOD8       ←     0  0  1  0  0  1  *  *      Select the internal clock as the input clock and disable
                        (** = 01, 10, 11)      the capture function.
PDCR        ←     -  -  -  -  -  1  -  -   }
PDFC        ←     -  -  -  -  -  1  -  -   }  Set PD2 to function as TO8.
TRUN8       ←     1  0  X  X  -  1  X  1      Start Timer 8.
```

Note: X = Don't care; "–" = No change

(4) Capture function examples

The capture function can be used in many ways. The following are examples:

① One-shot pulse triggered by an external tregger pulse input

② For frequency measurement

③ For pulse width measurement

④ For time difference measurement

① One-shot pulse output triggered by an external trigger pulse input

Set the up-counter UC8 to Free-Running Mode and to count the pulses from the internal clock input, input an external trigger pulse via the TI8 pin, and load the value of the up-counter into the capture register CAP8 on the rising edge of the TI8 input signal.

When the interrupt INT5 is generated on the rising edge of the TI8 input set TREG8 to the CAP8 value (c) plus a delay time (d) and set TREG9 to the value (c + d) plus the one-shot pulse width (p). (Thus TREG8 = c + d and TREG9 = c + d + p.)

When the interrupt INT5 occurs, TFFCR8<EQ9T8,EQ8T8> should be set to 11. The output value for TFF8 is inverted when the up-counter value matches TREG8 or TREG9. INTTR9 interrupt occurs, a one-shot pulse will be output and inversion will be disabled.

(c), (d) and (p) correspond to c, d and p in Figure 3.8.12.

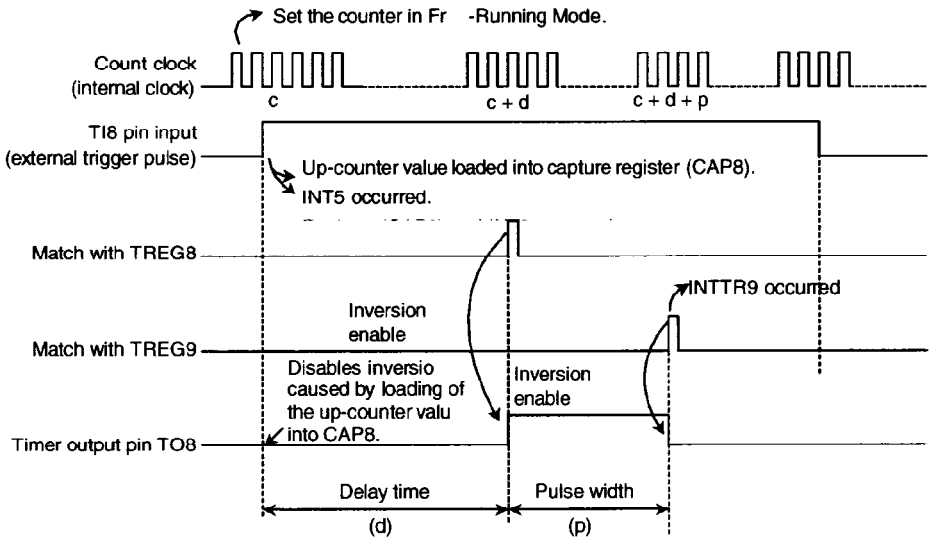


Figure 3.8.12 One-shot pulse output triggered by an external trigger pulse input (with delay)

Setting example: To output a 2-ms one-shot pulse with a 3-ms delay to the external trigger pulse via the TI8 pin.

Main settings

```
TMOD8    ← X X 1 0 1 0 0 1    Keep counting (maintain free-running counter).
                              Count using φT1.
TFFCR8   ← X X 0 0 0 0 1 0    Load the up-counter value into CAP8 on the rising edge
                              of the input to the TI8 pin.
                              Clear TFF8 to zero.
                              Disable TFF8 inversion.
PDCR     ← - - - - - 1 - -    Set PD2 to function as the TO8 pin.
PDFC     ← - - - - - 1 - -
INTE56   ← X - - - X 1 0 0    Enable INT5.
INTET89  ← X 0 0 0 X 0 0 0    Disable INTTR8 and INTTR9.
TRUN8    ← - 0 X X - 1 X 1    Start timer 8.
```

Setting of INT5

```
TREG8    ← CAP8 + 3 ms φT1
TREG9    ← TREG8 + 2 ms φT1
TFFCR8   ← X X - - 1 1 - -    Enable TFF8 inversion when the up-counter valu
                              matches the value of TREG8 or TREG9.
INTET89  ← X 1 0 0 X - - -    Enable INTTR9.
```

Setting INTTR9

```
TFFCR8   ← X X - - 0 0 - -    Disable TFF8 inversion when the up-counter valu
                              matches the value of TREG8 or TREG9.
INTET89  ← X 0 0 0 X - - -    Disable INTTR9.
```

Note: X = Don't care; "–" = No change

If no delay time is necessary, invert the timer flip-flop TFF8 when the up-counter value is loaded into the capture register (CAP8) and set the value of TREG9 to the value of CAP8 (c) plus the one-shot pulse width (p) when the interrupt INT5 occurs. TFF8 inversion should be enabled when the up-counter (UC8) value matches TREG9, and disabled when the interrupt INTTR9 is generated.

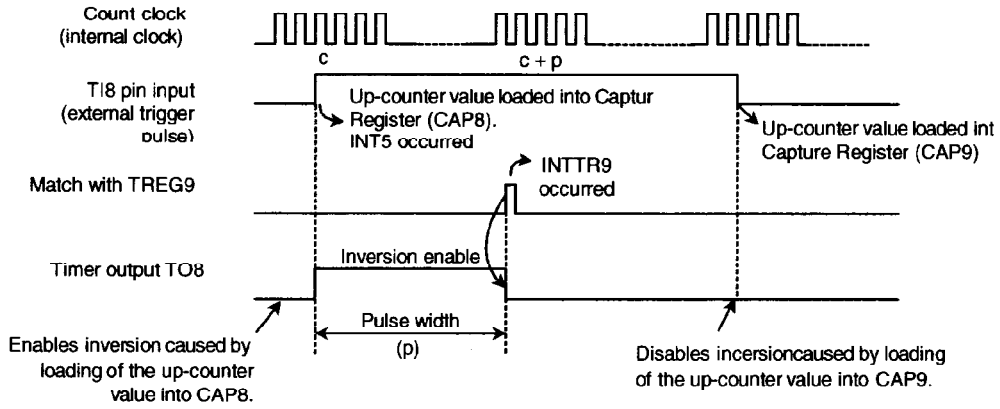


Figure 3.8.13 One-shot pulse output triggered by an external trigger pulse input (without delay)

② Frequency measurement

The frequency of the external clock can be measured in this mode. The clock is input via the TI8 pin and its frequency is measured using the two 8-bit timers of Timer 01 and the 16-bit timer / event counter Timer 8.

The TI8 pin input should be selected as the clock input to Timer 8. Set TMOD8<CAP89M1.CAP89M0> to 11. The value of the up-counter is loaded into the capture register CAP8 on the rising edge of the TFF1 signal from the timer flip-flop for the two 8-bit timers (Timer 01). and loaded into AP9 on the falling edge of the TFF1 signal.

The frequency is calculated using the difference between the values loaded into CAP8 and CAP9 when the interrupt (INTT0 or INTT1) is generated by either one of the 8-bit timers.

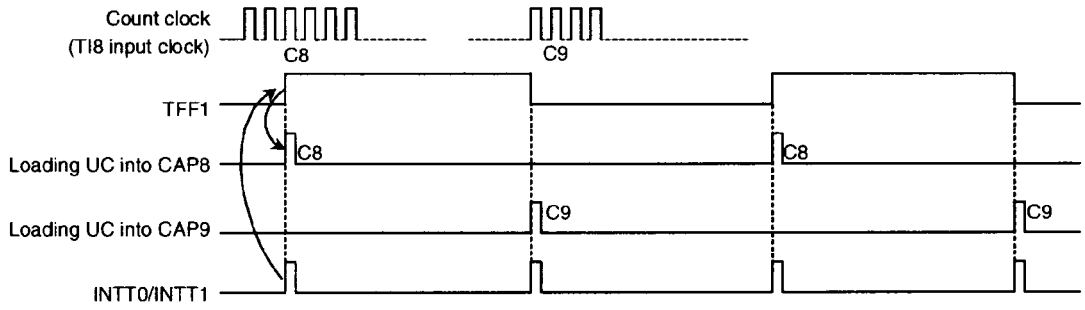


Figure 3.8.14 Frequency measurement

For example. if the value for the level 1 width of TFH of the 8-bit timer is set to 0.5 s and the difference between the values in C AP8 and CAP9 is 100. the frequency is 100 ÷ 0.5 s = 200 Hz.

③ Pulse width measurement

This mode allows the H-level width of an external pulse to be measured. With the 16-bit timer / event counter operating as a free-running counter counting the pulses from the internal clock input, the external pulse is input via the TI8 pin. Then, the capture function is used to load values from UC8 into CAP8 and CAP9 on the rising and falling edges of the external trigger pulse respectively. The interrupt INT5 is generated on the falling edge of TI8.

The pulse width is obtained from the difference between the values in CAP8 and CAP9 and the period of the internal clock.

For example, if the period of the internal clock is 0.8μs and the difference between the values in CAP8 and CAP9 is 100, the pulse width is 100 × 0.8 μs = 80 μs.

In addition, the pulse width in excess of the UC8 maximum count time specified by the clock source can be measured in software.

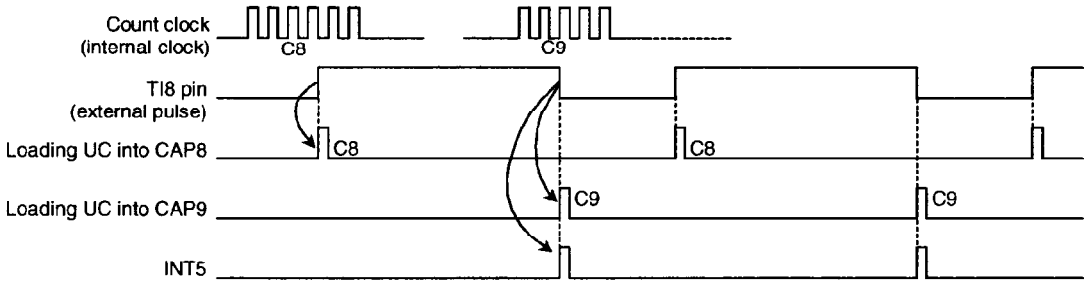


Figure 3.8.15 Pulse width measurement

Note: In Pulse Width Measuring Mode only (i.e. when TMOD 8<CAP89M1,CAP89M0> = 10), th external interrupt INT5 is generated on the falling edge of the signal input to the TI8 pin. In other modes it is generated on the rising edge.

The width of the L level is obtained at the second INT5 interrupt by multiplying the difference between the first C9 pulse and the second C8 pulse by the period of the internal clock.

④ Time difference measurement

This mode is used to measure the time difference between the rising edges of the external pulses input via TI8 and TI9.

With the 16-bit timer / event counter (timer 8) operating a s a free-running counter counting the pulses from the internal clock input, load the UC8 value into CAP8 on the rising edge of the signal input via TI8. This generates the interrupt INT5.

Similarly, the UC8 value is loaded into CAP9 on the rising edge of the signal input via TI9, generating the interrupt INT6.

The time difference between these pulses can be obtained by multiplying the value subtracted CAP8 from CAP9 and the internal clock cycle together at which loading the up-counter value into CAP8 and CAP9 has been done.

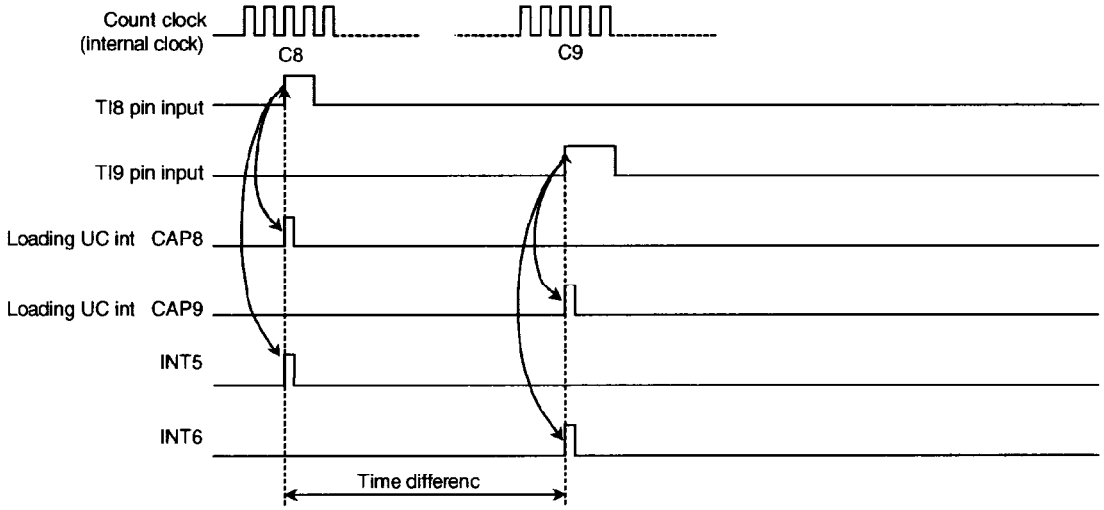


Figure 3.8.16 Time difference measurement

3.9 Serial Channels

TMP92CW10 incorporates two serial I/O channels. UART Mode (asynchronous transmission) or I/O Interface Mode (synchronous transmission) can be selected for both channels independently..

- I/O Interface Mode —— Mode 0: For transmitting and receiving I/O data using the synchronizing signal SCLK for extending I/O.
- UART Mode
  - 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.

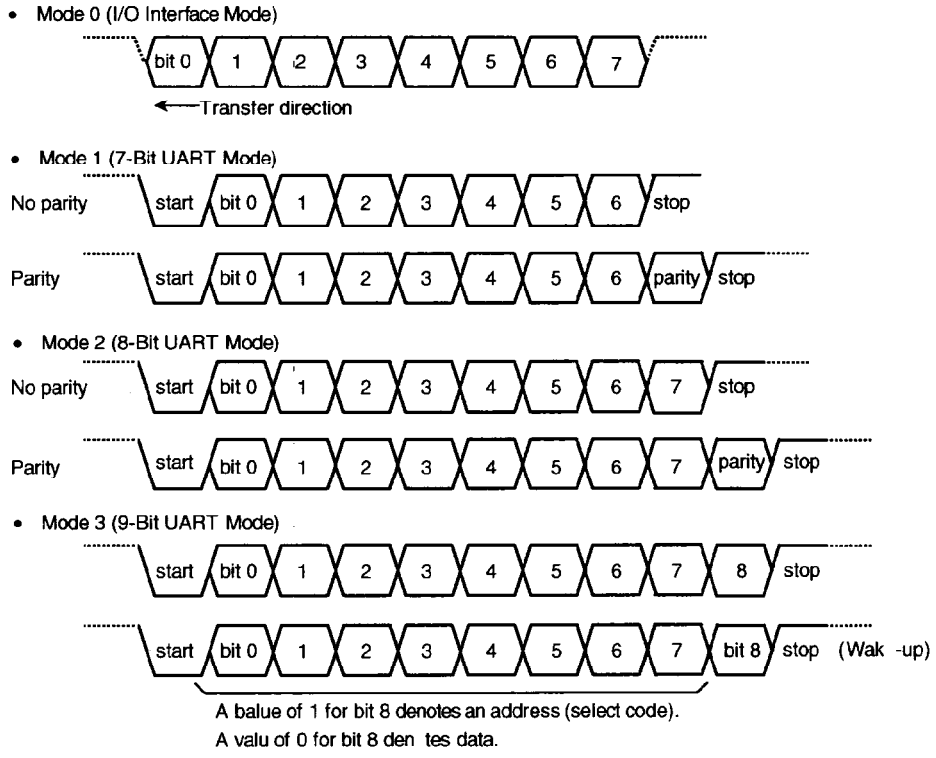
Figure 3.9.2 and Figure 3.9.3 are block diagrams for the two channels.

Serial Channels 0 and 1 can be used independently.

Both channels operate in the same function except for the following points: thus only the operation of Channel 0 is explained below.

Table 3.9.1 Differences between Channels 0 and 1

	Channel 0	Channel 1
Pin Name	TXD0 (PF0) RXD0 (PF1) $\overline{CTS0}$ /SCLK0 (PF2)	TXD1 (PF3) RXD1 (PF4) $\overline{CTS1}$ /SCLK1 (PF5)



w.

Figure 3.9.1 Data formats

3.9.1 Block diagrams for each circuit

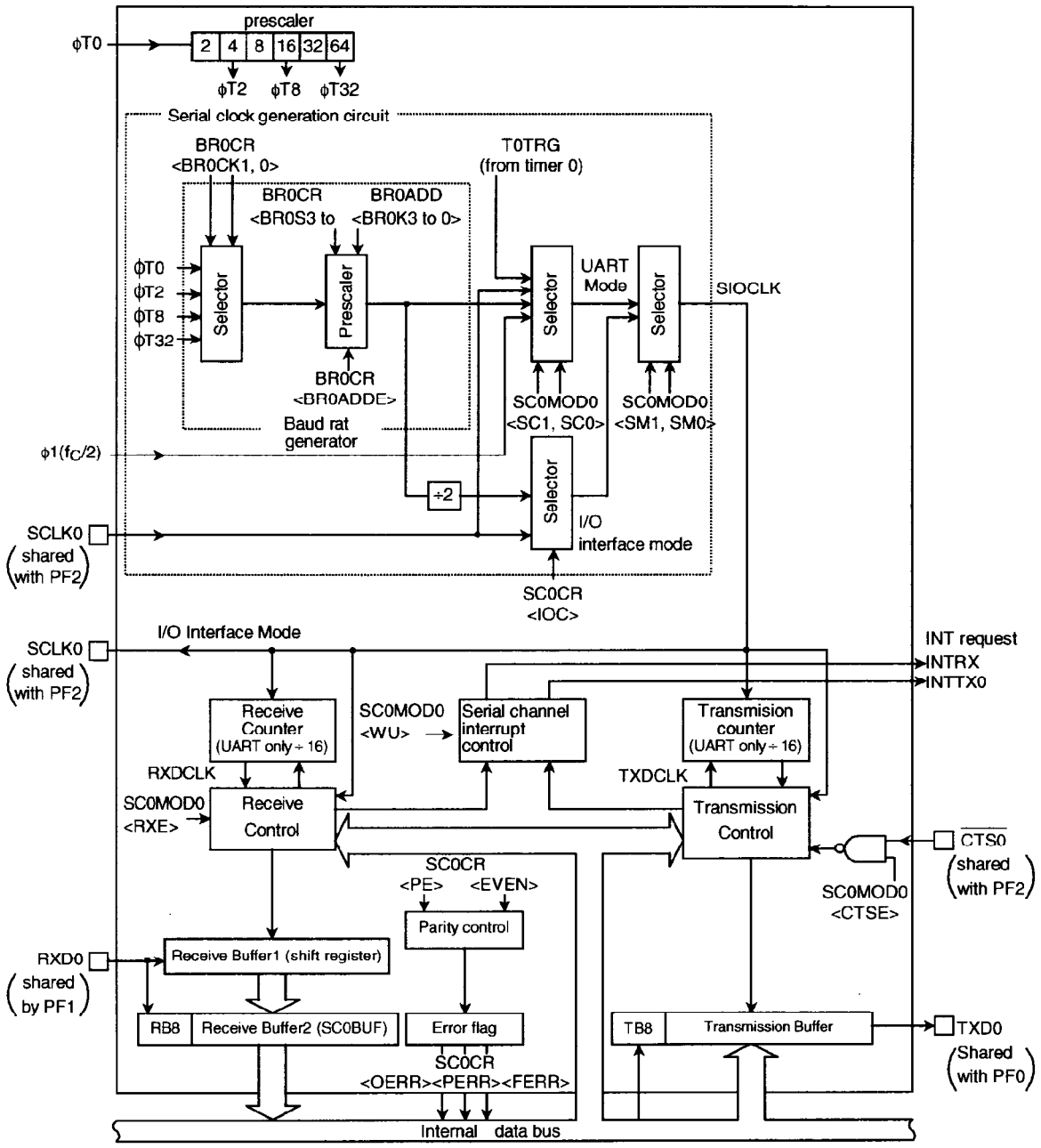


Figure 3.9.2 Block diagram for the Serial Channel 0

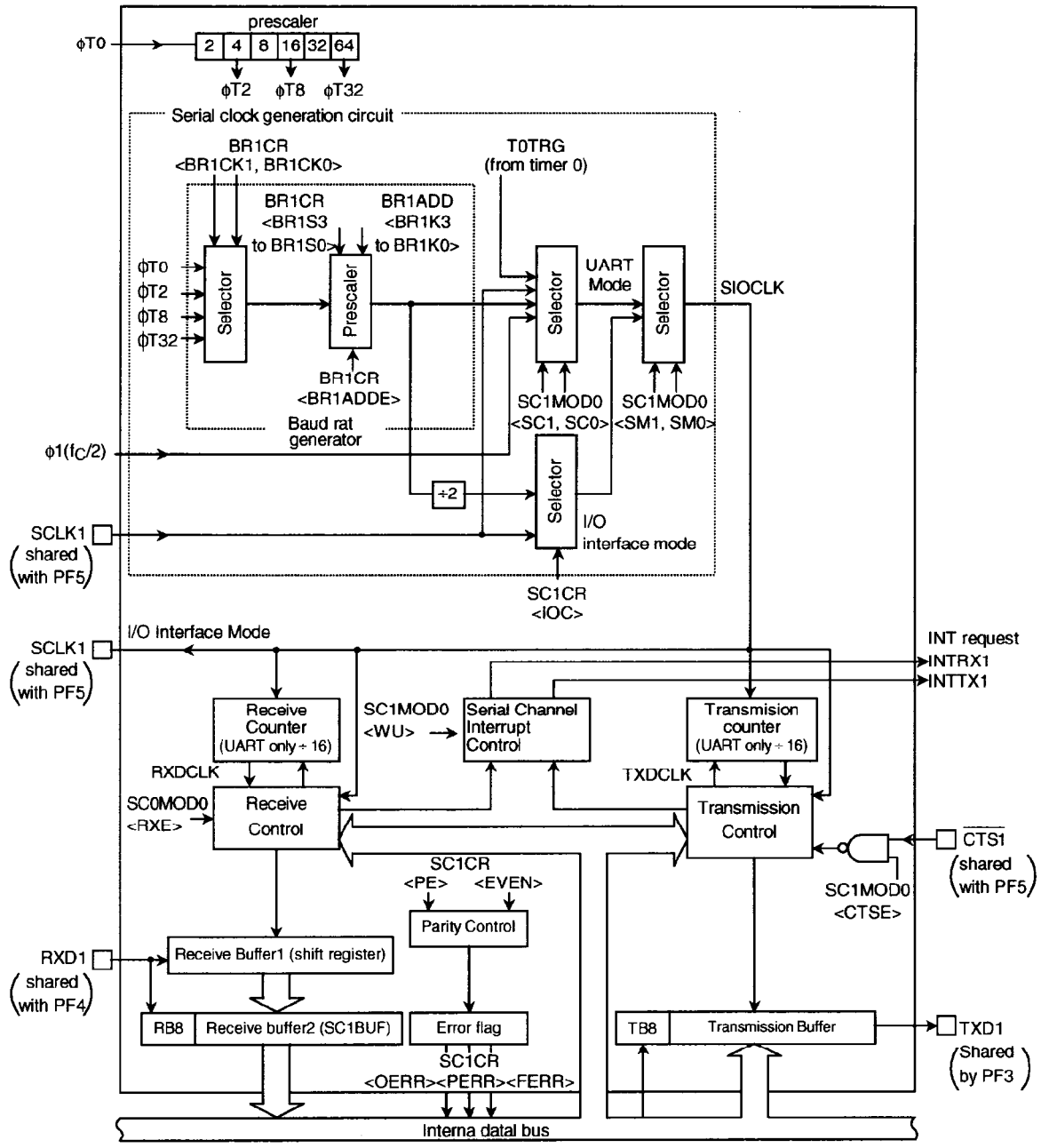


Figure 3.9.3 Block diagram for the Serial Channel 1

3.9.2 Operation for each circuit

(1) Prescaler

The two serial clocks are generated using a 6-bit prescaler. The clock φT0 is divided by 4 the CPU clock fc and input to this prescaler. The prescaler can be run by selecting the baud rate generator.

Table 3.9.2 shows prescaler clock resolution into the Baud rate generator.

Table 3.9.2 Prescaler Clock Resolution to Baud Rate Generator

Input clock	Clock resolution
φT0	$fc/2^2$
φT2	$fc/2^4$
φT8	$fc/2^6$
φT32	$fc/2^8$

The Baud Rate Generator selects between 4 clock inputs :φT0, φT2, φT8, and φT32 among the prescaler outputs.

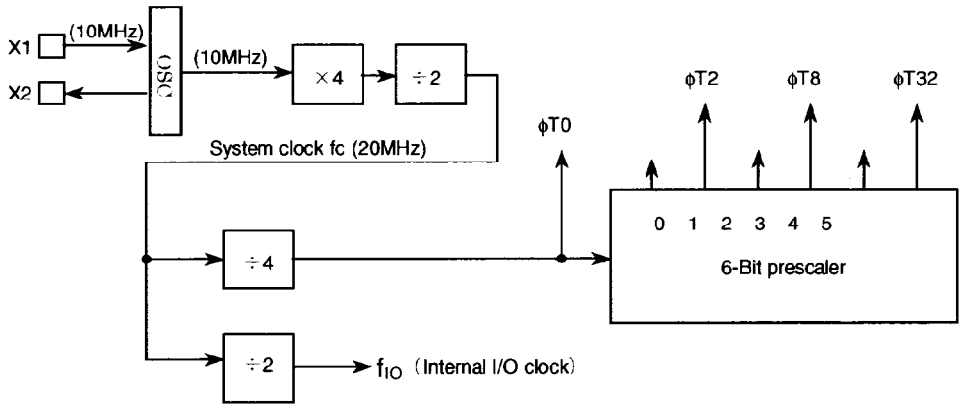


Figure 3.9.4 6-bit prescaler

(2) Baud rate generator

The baud rate generator is a circuit which generates transmission and receiving clocks which determine the transfer rate of the serial channels.
The input clock to the baud rate generator,φT0, φT2, φT8 or φT32, is generated by the 6-bit prescaler which is shared by the timers. One of these input clocks is selected using the BR0CR<BR0CK1 to BR0CK0> field in the Baud Rate Generator Control Register.

The baud rate generator includes a frequency divider, which divides the frequency by 1 or by n + m / 16 (n = 2 to 15, m = 0 to 15) for a total of 16 different values, determining the transfer rate.

The transfer rate is determined by the settings BR0CR<BR0ADDE, BR0S3 to BR0S0> and BR0ADD<BR0K3 to BR0K0>.

- In UART Mode

(1) When BR0CR<BR0ADDE> = 0

The settings BR0ADD<BR0K3 to BR0K0> are ignored. The baud rate generator divides the selected prescaler clock by N, which is set in BR0CK<BR0S3 to BR0S0>. (N = 1, 2, 3, ··· 16)

(2) When BR0CR<BR0ADDE> = 1

The N + (16 – K) / 16 division function is enabled. The baud rate generator divides the selected prescaler clock by N + (16– K) / 16 using the value of N set in BR0CR<BR0S3 to BR0S0> (N = 2, 3 ··· 15) and the value of K set in BR0ADD<BR0K3 to BR0K0> (K = 1, 2, 3 ··· 15)

Note: At N = 1 or N = 16, the N + (16– K) / 16 division function is disabled. Set BR0CR<BR0ADDE> to "0".

- In I/O Interface Mode

The N + (16 – K) / 16 division function is not available in I/O Interface Mode. Set BR0CR<BR0ADDE> to "0" before dividing by N.

The method for calculating the transfer rate when the baud rate generator is used is explained below.

- In UART Mode

$$\text{Baud Rate} = \frac{\text{Baud rate generator input clock frequency}}{\text{Frequency divider for baud rate generator}} \div 16$$

- In I/O Interface Mode

$$\text{Baud Rate} = \frac{\text{Baud rate generator input clock frequency}}{\text{Frequency divider for baud rate generator}} \div 2$$

- Integer divider (N divider)

If the source clock frequency (fc) is 19.6608 MHzfor example, the input clockis $\phi T2$ (fc/16), the frequency divider N (BR0CR<BR0S3 to BR0S0>) = 8, and BR0CR<BR0ADDE> = 0, the baud rate in UART Mode is as follows:

$$\text{Baud Rate} = \frac{fc/16}{8} \div 16$$

$$= 19.6608 \times 10^6 \div 16 \div 8 \div 16 = 9600 \text{ (bps)}$$

Note: The N + (16 – K) / 16 division function is disabled and setting BR0ADD<BR0K3 to BR0K0> is invalid.

- N+(16-K)/16 divider (UART Mode only)

Accordingly, when the source clock frequency (fc) = 15.9744 MHz, the input clockis $\phi T2$ (fc/16), the frequency divider N (BR0CR<BR0S3 to BR0S0>) = 6, K (BR0ADD<BR0K3 to BR0K0>) = 8, and BR0CR <BR0ADDE> = 1, the baud rate in UART Mode is as follows:

$$\text{Baud Rate} = \frac{fc/16}{6 + (16 - 8)/16} \div 16$$

$$= 15.9744 \times 10^6 \div 16 \div (6+8/16) \div 16 = 9600 \text{ (bps)}$$

Table 3.9.3 and Table 3.9.4 show examples of UART Mode transfer rates.

Alternatively, the external clock can be used as the serial clock. The method for calculating the baud rate is explained below:

- In UART Mode

  Baud rate = external clock input frequency÷ 16

  (External clock input frequency) must be less than or equal to fc / 4

- In I/O Interface Mode

  Baud rate = external clock input frequency

  (External clock input frequency) must be less than or equal to fc / 16

Table 3.9.3 Selection of UART Baud Rate(1)

(when baud rate generator is used and BR0CR <BR0ADDE> = 0)

Unit (kbps)

fc [MHz]	Input Clock / Frequency Divider	φT0 (4/fc)	φT2 (16/fc)	φT8 (64/fc)	φT32 (256/fc)
18.432000	15	19.200	4.800	1.200	0.300
19.660800	8	38.400	9.600	2.400	0.600
↑	16	19.200	4.800	1.200	0.300

Note: Transfer rates in I/O Interface Mode are eight times faster than the values given above.

Table 3.9.4 Selection of UART Baud Rate(2)

(when timer 0 is used with input Clock φT1)

Unit (kbps)

fc / TREG0	20 MHz	19.6608 MHz	16 MHz
02H		76.8	62.5
04H		38.4	31.25
05H	31.25		
08H		19.2	
10H		9.6	

Method for calculating the transfer rate (when timer 0 is used):

$$\text{Transfer rat} = \frac{fc}{TREG0 \times 8 \times 16}$$

(when timer 0 (input clock φT1) is used)

Note: The Timer 0 match-detect signal cannot be used as the transfer clock in I/O Interface Mode.

(3) Serial clock generation circuit

This circuit generates the basic clock for transmitting and receiving data.

- In I/O Interface Mode

In SCLK Output Mode (i.e. when SC0CR<IOC> = 0) the basic clock is generated by dividing the output of the baud rate generator by 2, as described previously.

In SCLK Input Mode (i.e. when SC0CR<IOC> = 1) either the rising edge or falling edge, as determined by the setting of the SC0CR<SCLKS>, will be used to generate the basic clock.

- In UART Mode

The SC0MOD0 <SC1, SC0> setting determines whether the baud rate generator clock, the internal clock ϕ1 (fc/2), the match detect signal from Timer 0 or the external clock (SCLK0) is used to generate the basic clock SIOCLK.

(4) Receiving counter

The receiving counter is a 4-bit binary counter used in UART Mode which counts up the pulses of the SIOCLK clock. It takes 16 SIOCLK pulses to receive 1 bit of data: each data bit is sampled three times – on the 7th, 8th and 9th clock cycles.

The value of the data bit is determined from these three samples using the majority rule.

For example, if the data bit is sampled respectively as 1, 0 and 1 on 7th, 8th and 9th clock cycles, the received data bit is taken to be 1. A data bit sampled as 0, 0 and 1 is taken to be 0.

(5) Receiving control

- In I/O Interface Mode

In SCLK Output Mode (i.e. when SC0CR<IOC> = 0) the RXD0 signal is sampled on the rising edge of the shift clock output on the SCLK0 pin.

In SCLK Input Mode (i.e. when SC0CR<IOC> = 1) the RXD0 signal is sampled on either the rising or falling edge of the SCLK0 input, as determined by the setting of SC0CR<SCLKS> setting.

- In UART Mode

The receiving control block has a circuit which detects a start bit using the majority rule. Received bits are sampled three times: when two or more out of three samples are 0, the bit is recognized as the start bit and the receiving operation commences.

The values of the data bits which are received are also determined using the majority rule.

(6) The receiving buffers

To prevent Overrun errors, the receiving buffers are arranged in a double-buffer structure.

Received data is stored one bit at a time in receiving buffer 1 (which is a shift register). When 7 or 8 bits of data have been stored in receiving buffer 1, the stored data is transferred to receiving buffer 2 (SC0BUF); this causes an INTRX0 interrupt to be generated. The CPU only reads receiving buffer 2 (SC0BUF). Even before the CPU has finished reading the contents of receiving buffer 2 (SC0BUF), more data can be received and stored in receiving buffer 1.

However, if receiving buffer 2 (SC0BUF) has not been read completely before all the bits of the next data item are received by receiving buffer 1, an Overrun errorwill occur. If an Overrun error occurs, the contents of receiving buffer 1 will be lost; although the contents of receiving buffer 2 and SC0CR<RB8> will be preserved, however.

SC0CR<RB8> is used to store either the parity bit – added in 8-Bit UART Mode – or the most significant bit (MSB) – in 9-Bit UART Mode.

In 9-Bit UART Mode the wake-up function for the slave controller is enabled by setting SC0MOD0<WU> to 1; in this mode INTRX0 interrupts occur only when the value of SC0CR<RB8> is 1.

(7) Transmission counter

The transmission counter is a 4-bit binary counter which is used in UART Mode and which, like the receiving counter, counts the SIOCLK clock pulses; a TXDCLK pulse is generated every 16 SIOCLK clock pulses.

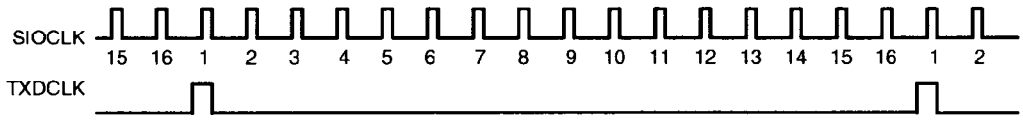


Figure 3.9.5 Generation of the transmission clock

(8) Transmission controller

- In I/O Interface Mode

In SCLK Output Mode (i.e. when SC0CR<IOC> = 0) the data in the transmission buffer is output one bit at a time to the TXD0 pin on the rising edge of the shift clock which is output on the SCLK0 pin.

In SCLK Input Mode with the setting SC0CR<IOC> = 1, the data in the transmission buffer is output one bit at a time to the TXD0 pin on the rising or falling edge of the SCLK0 input, according to the SC0CR<SCLKS> setting.

- In UART Mode

When transmission data sent from the CPU is written to the transmission buffer, transmission starts on the rising edge of the next TXDCLK, generating a transmission shift clock TXDSFT.

Handshake function

Serial Channels 0 and 1 each have a $\overline{CTS}$ pin. Use of this pin allows data to be sent in units of one frame; thus. Overrun errors can be avoided. The handshake function can be enabled or disabled using the SC0MOD0 <CTSE> setting.

If the $\overline{CTS0}$ pin goes High on completion of transmission of the current data , transmission of the next data byte will not start until $\overline{CTS0}$ goes Low again. Irrespective of the level of $\overline{CTS0}$. however. the transmittiing microcontroller generates an INTTX0 interrupt, requesting its CPU to write the next data byte to the transmittion buffer.

Although there is no $\overline{RTS}$ pin. a handshake function can easily be implemented by assigning any port to perform the $\overline{RTS}$ function. The $\overline{RTS}$ should b e output High signal to a half to data transmision after data receiving has been completed in software in the RXD interrupt routine.

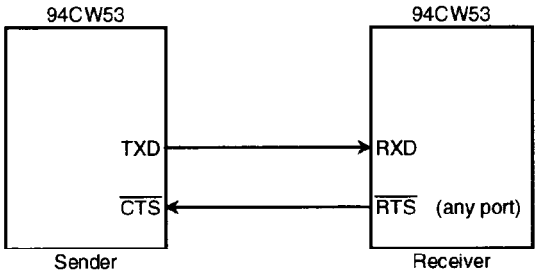
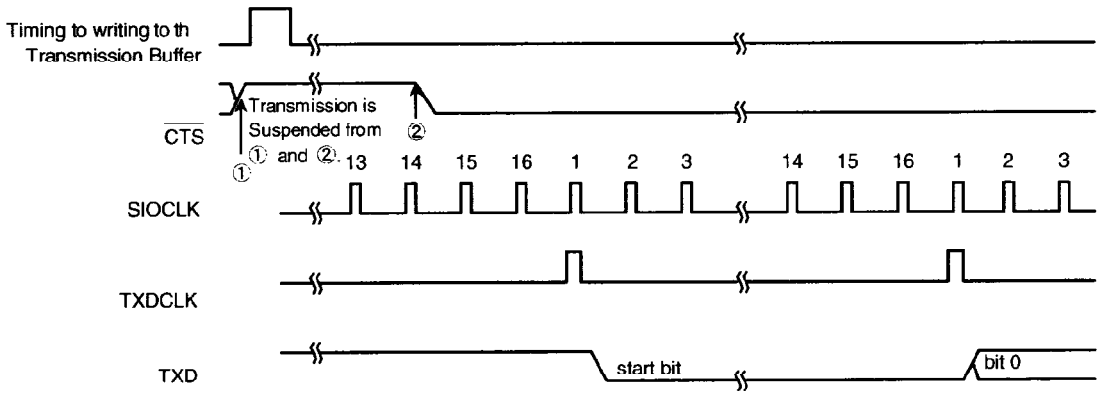


Figure 3.9.6 Handshake function



Note 1: If the $\overline{CTS}$ signal goes High during transmission, no more data will be sent after completion of the current transmission.

Note 2: Transmission starts on the first falling edge of the TXDCLK clock after th $\overline{CTS}$ signal has fallen.

Figure 3.9.7 $\overline{CTS}$ (Clear to send) Timing

(9) Transmission Buffer

The transmission buffer (SC0BUF) shifts out and sends the transmission data written from the CPU, in order one bit at a time starting with the least significant bit (LSB) and finishing with the most significant bit (MSB). When all the bits have been shifted out, the empty transmission buffer generates an INTTX0 interrupt.

(10) Parity control circuit

When SC0CR<PE> in the Serial Channel Control Register is set to 1, it is possible to transmit and receive data with parity. However, parity can be added only in 7-Bit UART Mode or 8-Bit UART Mode. The SC0CR<EVEN> field in the Serial Channel Control Register allows either even or odd parity to be selected.

In the case of transmission, parity is automatically generated when data is written to the transmission buffer SC0BUF. The data is transmitted after the parity bit has been stored in SC0BUF<TB7> in 7-Bit UART Mode or in SC0MOD0<TB8> in 8-Bit UART Mode. SC0CR<PE> and SC0CR<EVEN> must be set before the transmission data is written to the transmission buffer.

In the case of receiving, data is shifted into receiving buffer 1, and the parity is added after the data has been transferred to receiving buffer 2 (SC0BUF), and then compared with SC0BUF<RB7> in 7-Bit UART Mode or with SC0CR<RB8> in 8-Bit UART Mode. If the parity bit does not match, a Parity error is generated and the SC0CR<PERR> flag is set.

(11) Error flags

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

1. Overrun error <OERR>

   If all the bits of the next data item have been received in receiving buffer 1 while valid data still remains stored in receiving buffer 2 (SC0BUF), an Overrun error is generated.

2. Parity error <PERR>

   The parity generated for the data shifted into receiving buffer 2 (SC0BUF) is compared with the parity bit received via the RXD pin. If they are not equal, a Parity error is generated.

3. Framing error <FERR>

   The stop bit for the received data is sampled three times close to the middle of the pulse. If two or three of the samples are 0, a Framing error is generated.

(12) Timing generation

① In UART Mode

Receiving

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

Note: In 9-Bit Mode and 8-Bit + Parity Mode, interrupts coincide with the ninth bit pulse. Thus, when servicing the interrupt, it is necessary to allow a 1-bit period to elapse (so that the stop bit can be transferred) in order to allow proper checking for Framing error.

Transmitting

Mode	9-Bit	8-Bit + Parity	8-Bit, 7-Bit + Parity, 7-Bit
Interrupt timing	Just before stop bit is transmitted	Just before stop bit is transmitted	Just before stop bit is transmitted

② I/O interface

Transmission Interrupt timing	SCLK Output Mode	Immediately after rise of last SCLK signal. (See figure 3.9.20.)
	SCLK Input Mode	Immediately after rise of last SCLK signal Rising Mode, or immediately after fall in Falling Mode. (See figure 3.9 21.)
Receiving Interrupt timing	SCLK Output Mode	Timing used to transfer received dat to Receive Buffer 2 (SC0BUF) (i.e. immediately after last SCLK). (See figure 3.9 22.)
	SCLK Input Mode	Timing used to transfer received data to Receive Buffer 2 (SC0BUF) (i.e. immediately after last SCLK). (See figure 3.9 23.)

3.9.3 SFR

SC0MOD0 (00C2H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	Transfer data bit 8	Hand shak 0: CTS disable 1: CTS enable	Receive function 0: Receiv disable 1: Receiv enable	Wake up function 0: disable 1: enable	Serial Transmission Mode 00: I/O interface Mode 01: 7-bit UART Mode 10: 8-bit UART Mode 11: 9-bit UART Mode		Serial transmission clock (UART) 00: T0TRG 01: Baud rat generator 10: Internal clo ϕ1 11: External clco (SCLK0 input)	

Serial transmission clock source (UART)

00 Timer 0 match detect signal (T0TRG)
01 Baud rate generator
10 Internal clo ϕ1
11 External clo (SCLK0 input)

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

Serial Transmission Mod

00	I/O Interface Mode	
01	UART	7-bit mode
10		8-bit mode
11		9-bit mode

Wake-up function

	9-Bit UART	Other Modes
0	Interrupt generated when data is received	Don't care
1	Interrupt generated onl when RB8 = 1	

Receiving Function

0	Receive disabled
1	Receive enabled

Handshake function ($\overline{CTS}$ pin) Enable

0	Disabled (transmission permanently enabled)
1	Enabled

Transmission data bit 8

Figure 3.9.8 Serial Mode Control Register (channel 0, SC0MOD0)

SC1MOD0 (00CAH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	Transfer data bit 8	Hand shak 0: CTS disable 1: CTS enable	Receive function 0: Receiv disable 1: Receiv enable	Wake up function 0: disable 1: enable	Serial Transmission Mode 00: I/O interface Mode 01: 7-bit UART Mode 10: 8-bit UART Mode 11: 9-bit UART Mode		Serial transmission clock (UART) 00: T0TRG 01: Baud rat generator 10: Internal clo ϕ1 11: External clco (SCLK1 input)	

Serial transmission clock source (UART)

00	Timer 0 match detect signal (T0TRG)
01	Baud rate generator
10	Internal clo ϕ1
11	External clo (SCLK1 input)

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

Serial Transmission Mod

00	I/O Interface Mode	
01	UART	7-bit mode
10		8-bit mode
11		9-bit mode

Wake-up function

	9-Bit UART	Other Modes
0	Interrupt generated when data is received	Don't care
1	Interrupt generated onl when RB8 = 1	

Receiving Function

0	Receive disabled
1	Receive enabled

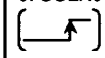
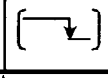
Handshake function ($\overline{CTS}$ pin) Enable

0	Disabled (transmission permanently enebled)
1	Enabled

Transmission data bit 8

Figure 3.9.9 Serial Mode Control Register (channel 1, SC1MOD0)

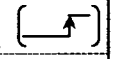
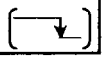
SC0CR (00C1H)

	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	0	0	0	0	0	0	0	0
Function	Received data bit 8	Parity 0: odd 1: even	Parity addition 0: disable 1: enable	1: error			0: SCLK0  1: SCLK0 	0: baud rate generator 1: SCLK0 pin input
				Overrun	Parity	Framing		

I/O interface input clock selection

0	Baud rat generator
1	SCLK0 pin input

Edge selection for SCLK0 pin (Input Mode Only)

0	Transmits and receivers data on rising edge of SCLK0. 
1	Transmits and receivers data on falling edge SCLK0. 

Framing Error flag
Parity Error flag
Overrun Error flag
} cleared to 0 when read

Parity addition enabl

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parit
1	Even parit

Received data 8

Note: Since any instruction which reads the register will automatically clear all three errors flags to 0, do not test the flag individually using a bit-testing instruction.

Figure 3.9.10 Serial Control Register (channel 0, SC0CR)

SC1CR (00C9H)

	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)			R/W	
After Reset	0	0	0	0	0	0	0	0
Function	Received data bit 8	Parity 0: odd 1: even	Parity addition 0: disable 1: enable	1: error			0: SCLK1 ↑ 1: SCLK1 ↓	0: baud rate generator 1: SCLK1 pin inpu
				Overru	Parity	Framing		

I/O interface input clock select

0	Baud rate generator
1	SCLK1 pin input

Edge selection for SCLK1 pin (input mode only)

0	Transmits and receives data on rising edge of SCLK1. ↑
1	Transmits and receives data on falling edge of SCLK1. ↓

Framing Error flag
Parity Error flag
Overrun Error fl
} cleared to 0 when read

Parity addition enable

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parit
1	Even parit

Received data bit 8

Note: Since any instruction which reads the register will automatically clear all three errors flags to 0, do not test the flag individually using a bit-testing instruction.

Figure 3.9.11 Serial Control Register (channel 1, SC1CR)

BR0CR (00C3H)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR0ADDE	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		Frequency divisor N setting			

+(16–K)/16 division enabl

0	Disable
1	Enable

Setting the input clock of baud rate generator

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

BR0ADD (00C4H)

	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)			

Baud rate generator frequency divisor setting

BR0CR <BR0S3:0> / BR0ADD <BR0K3:0>	BR0CR<BR0ADDE> = 1		BR0CR<BR0ADDE> = 0
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) (UART only) to 1111(N = 15) 0000(N = 16)
0	Disable	Disable	
0001(K = 1) to 1111(K = 15)	Disable	Divided b N + (16-K) /16	Divided by N

Note 1: When the +(16 – K)/16 division function is to be used set BR0CR <BR0ADDE> to "1" after first setting BR0ADD<BR0K3 to 0> to K (K = 1 to 15).

Note 2: The + (16 – K)/16 division function is possible to use in only UART mode.
In I/O interface mode set BR0CR <BR0ADDE> to "0" and disable + (16 – K)/16 division function.

Figure 3.9.12 Baud rate generator control (channel 0, BR0CR, BR0ADD)

BR1CR (00CBH)

	7	6	5	4	3	2	1	0
bit Symbol	—	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
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		Frequency devisor N settin			

+(16 - K) / 16 division enabl

0	Disabled
1	Enabled

Input clock selection for baud rate generator

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

BR1ADD (00CCH)

	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					Set frequency divisor K (divided by N + (16 – K)/16)			

Baud rate generator frequency divisor setting

BR1CR <BR1S3–BR1S0> / BR1ADD <BR1K3–BR1K0>	BR1CR<BR1ADDE> = 1		BR1CR<BR1ADDE> = 0
	0000(N = 16) or 0001(N = 1)	0010(N = 2) to 1111(N = 15)	0001(N = 1) (UART Only) to 1111(N = 15) 0000(N = 16)
0000	Disable	Disable	
0001(K = 1) to 1111(K = 15)	Disable	Disable by N + (16 – K) / 16	Divided by N

Note 1: When the + (16 – K) / 16 division function is to be used, set BR1CR <BR1ADDE> to "1" after first setting BR1ADD <BR1K3 to 0> to K (K = 1 to 15).

Note 2: The + (16 – K) / 16 division functions can only be use in only UART mode.
In I/O interface mode set BR1CR <BR1ADDE> to "0" and disable + (16 – K) / 16 division.

Figure 3.9.13 Baud rate generator control (channel 1, BR1CR, BR1ADD)

SC0BUF (00 0H)

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	(Reveiving)

Note: SC0BUF must not be used any of the read-modify-write instructions.

Figure 3.9.14 Serial Transmission/Receiving Buffer Registers (channel 0, SC0BUF)

SC0MOD1 (00C5H)

	7	6	5	4	3	2	1	0
Bit symbol	I2S0	FDPX0	-	-	-	-	-	-
Read/Writ	R/W	R/W						
After Reset	0	0	-	-	-	-	-	-
Function	IDLE2 0: Stop 1: Run	duplex 0: half 1: full						

Figure 3.9.15 Serial Mode Control Register 1 (channel 0, SC0MOD1)

SC1BUF (00C8H)

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: SC1BUF must not be used any of the read-modify-write instructions.

Figure 3.9.16 Serial Transmission/Receiving Buffer Registers (channel 1, SC1BUF)

SC1MOD1 (00CDH)

	7	6	5	4	3	2	1	0
bit Symbol	I2S1	FDPX1	-	-	-	-	-	-
Read/Writ	R/W	R/W						
After Reset	0	0	-	-	-	-	-	-
Function	IDLE2 0: Stop 1: Run	Duplex 0: half 1: full						

Figure 3.9.17 Serial Mode Control Register 1 (channel 1, SC1MOD1)

3.9.4 Operation for each mode

(1) Mode 0 (I/O Interface Mode)

This mode allows an increase in the number of I/O pins available for transmitting data to or receiving data from an external shift register.

This mode is further subdivided into the SCLK Output Mode for outputting the synchronous clock SCLK and SCLK Input Mode in which an external synchronous clock is input.

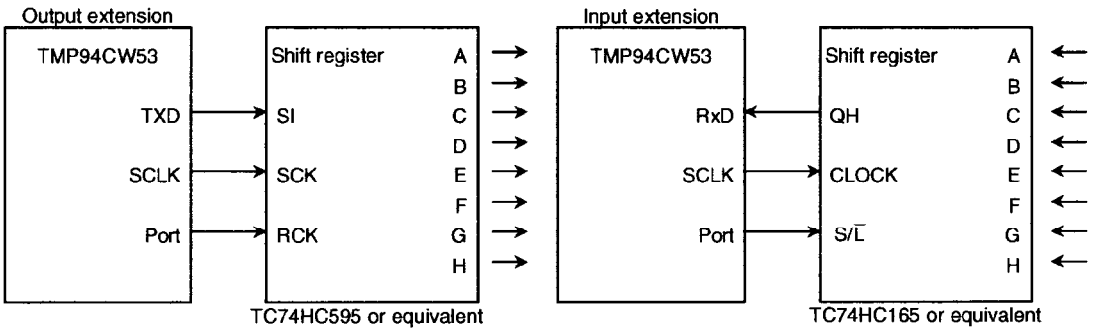


Figure 3.9.18 Example of Connection for use with SCLK Output Mode

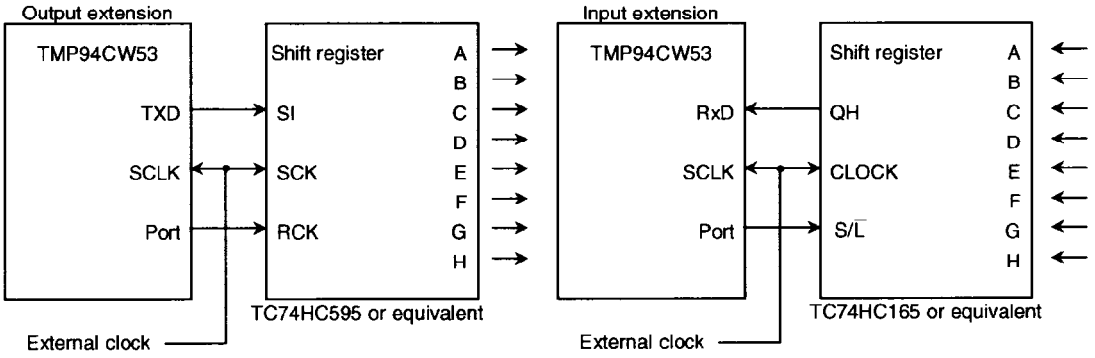


Figure 3.9.19 Example of Connection for use with SCLK Input Mode

① Transmission

In SCLK Output Mode 8-bit data and a synchronous clock are output on the TXD0 and SCLK0 pins respectively each time the CPU writes the data to the transmission buffer.

When all the data has been output. INTES0 <ITX0C> is set to 1. causing an INTTX0 interrupt to be generated.

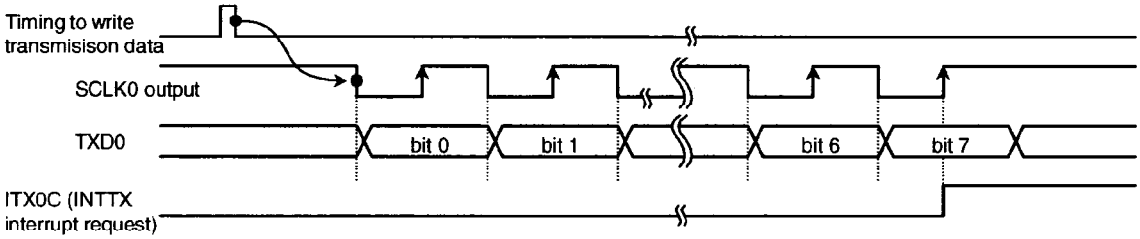


Figure 3.9.20 Transmitting Operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK Input Mode. 8-bit data is output on the TXD0 pin when the SCLK0 input becomes active after the data has been written to the Transmission Buffer by the CPU.

When all the data has been output. INTES0 <ITX0C> is set to 1. causing an INTTX0 interrupt to be generated.

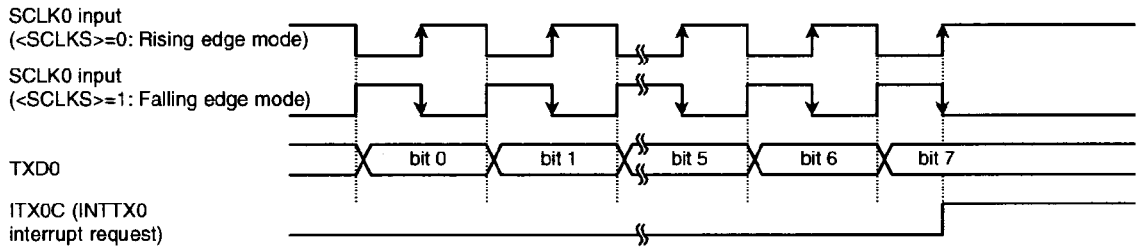


Figure 3.9.21 Transmitting Operation in I/O Interface Mode (SCLK0 Input Mode)

② Receiving

In SCLK Output Mode the synchronous clock is output on the SCLK0 pin and the data is shifted to receiving buffer 1. This is initiated when the Receive Interrupt flag INTES0<IRX0C> is cleared due to the received data being read. When 8-bit data is received, the data is transferred to receiving buffer 2 (SC0BUF) following the timing shown below and INTES0<IRX0C> is set to 1 again, causing an INTRX0 interrupt to be generated.

Setting SC0MOD0<RXE> to 1 initiates SCLK0 output.

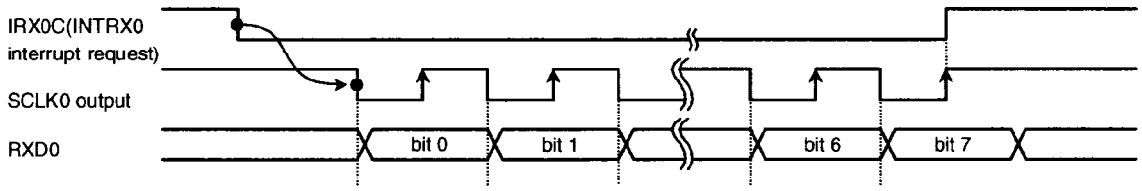


Figure 3.9.22 Receiving operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK Input Mode the data is shifted to Receiving Buffer 1 when the SCLK input goes active. The SCLK input goes active when the Receive Interrupt flag INTES0 <IRX0C> is cleared due to the received data being read. When 8-bit data is received, the data is shifted to receiving buffer 2 (SC0BUF) following the timing shown below and INTES0 <IRX0C> is set to 1 again, causing an INTRX0 interrupt to be generated.

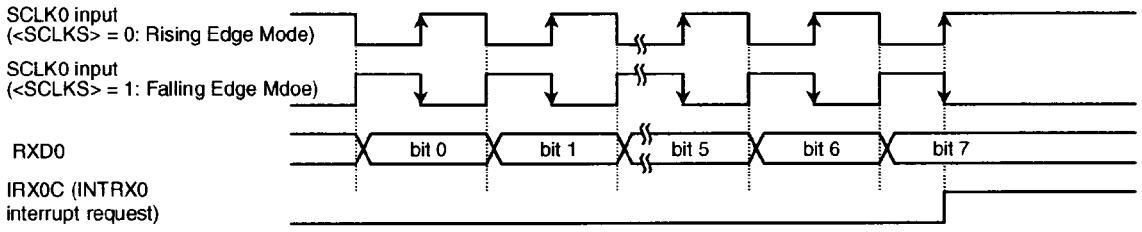


Figure 3.9.23 Receiving Operation in I/O interface Mode (SCLK0 Input Mode)

Note: The system must be put in the Receive Enable state (i.e SC0MOD0<RXE> must be set to 1) before data can be received.

③ Transmission and receiving (Full Duplex Mode)

When Full Duplex Mode is used, set the Receive interrupt level to 0 and set enable the level of Transmission interrupt to any value from 1 to 6. Ensure that the program which transmits the interrupt reads the receiving buffer before settingthe value of the next item of transmission data.
The following is an example of this:

Example: Channel 0, SCLK output
Baud rate = 9600 bps
fc = 19.6608 MHz

Main routine

	7	6	5	4	3	2	1	0	
INTES0	X	0	0	1	X	0	0	0	Set the INTTX0 level to 1. Set the INTRX0 level to 0.
PFCR	-	-	-	-	-	1	0	1	Set PF0, PF1 and PF2 to function as the TXD0, RXD0
PFFC	-	-	-	-	-	1	-	1	and SCLK0 pins respectivel
SC0MOD0	0	0	1	0	0	0	0	0	Enable receiving and select I/O Interface Mode.
SC0MOD1	1	1	0	0	0	0	0	0	Select Full Duplex Mode.
SC0CR	0	0	0	0	0	0	0	0	Sclk_out, transmit on negative edge, receive on positive edg
BR0CR	0	0	0	1	1	0	0	0	Baud rate = 9600 bps
SC0MOD0	0	0	1	0	0	0	0	0	Enable receiving
SC0BUF	*	*	*	*	*	*	*	*	Set the transmit data and start.

INTTX0 interrupt routine

Acc	←	SC0BUF							Read the receiving buffer.
SC0BUF	*	*	*	*	*	*	*	*	Set the next transmit data.

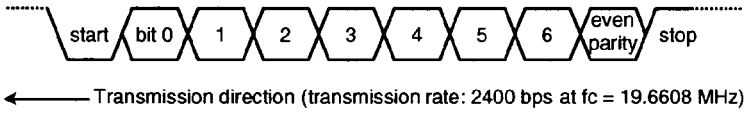
Note: X = Don't care; "–" = No change

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

7-Bit UART Mode can be selected by setting SC0MOD0<SM1.SM0> to 01.

In this mode a parity bit can be added. Use of a parity bit is enabled or disabled by the setting of the Serial Channel Control Register SC0CR<PE> bit: whether even parity or odd parity will be used is determined by the SC0CR<EVEN> setting when SC0CR<PE> is set to 1 (enabled).

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



	7 6 5 4 3 2 1 0	
PFCR	← – – – – – – – 1	} Set PF0 to function as the TXD0 pin.
PFFC	← – – – – – – – 1	
SC0MOD0	← X 0 – X 0 1 0 1	Select 7-Bit UART Mode.
SC0CR	← X 1 1 X X X 0 0	Add even parity.
BR0CR	← 0 0 1 0 1 0 0 0	Set the transfer rate to 2400 bps.
INTES0	← X 1 0 0 – – – –	Enable the INTTX0 interrupt and set it to Interrupt Level 4.
SC0BUF	← * * * * * * * *	Set data for transmission.

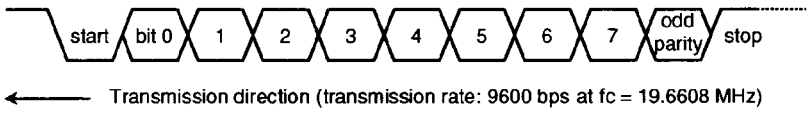
Note: X = Don't care; "–" = No change

(3) Mode 2 (8-Bit UART Mode)

8-Bit UART Mode is selected by setting SC0MOD0<SM1.SM0> to 10.

In this mode a parity bit can be added. Use of a parity bit is enabled or disabled by the setting of SC0CR<PE>: whether even parity or odd parity will be used is determined by the SC0CR<EVEN> setting when SC0CR<PE> is set to 1 (enabled).

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



Main settings

		7	6	5	4	3	2	1	0	
PFCR	←	–	–	–	–	–	–	0	–	Set PF1 to function as the RXD0 pin.
SC0MOD0	←	–	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	0	0	1	1	0	0	0	Set the transfer rate to 9600 bps.
INTES0	←	–	–	–	–	X	1	0	0	Enable the INTTX0 interrupt and set it to Interrupt Level 4.

Interrupt processing

Acc	← SC0CR AND 00011100	Check for errors.
if Acc	≠ 0 then ERROR	
Acc	← SC0BUF	Read the received data.

Note: X = Don't care; "–" = No change

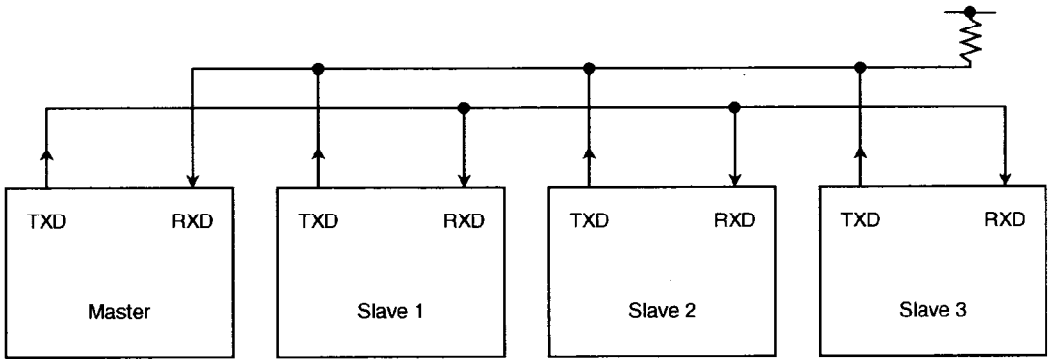
(4) Mode 3 (9-Bit UART Mode)

9-Bit UART Mode is selected by setting SC0MOD0<SM1,SM0> to 11.

In this mode parity bit cannot be added. In the case of transmission the MSB (9th bit) is written to SC0MOD0<TB8>. In the case of receiving it is stored in SC0CR<RB8>. When the buffer is written and read, the MSB is read or written first, before the rest of the SC0BUF data.

Wake-up function

In 9-Bit UART Mode, the wake-up function for slave controllers is enabled by setting SC0MOD0<WU> to 1. The interrupt INTRX0 can only be generated when<RB8> = 1.



Note: The TXD pin of each slave controller must be in Open-Drain.

Figure 3.9.24 Serial Link using Wake-up function

Protocol

① Select 9-Bit UART Mode on the master and slave controllers.

② Set the SC0MOD0<WU> bit on each slave controller to 1 to enable data receiving.

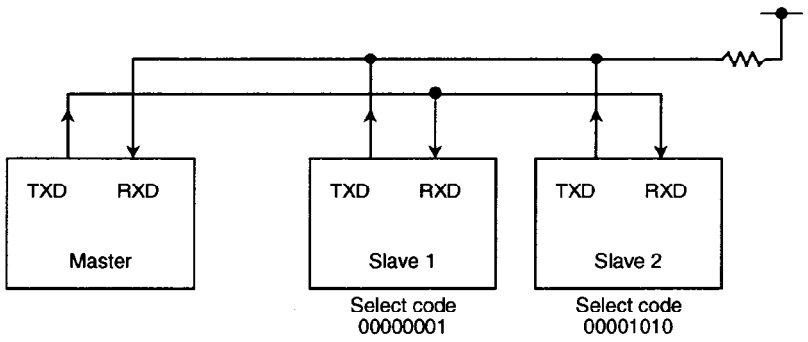
③ The master controller transmits data one frame at a time. Each frame includes an 8-bit select code which identifies a slave controller. The MSB (bit 8) of the data (<TB8>) is set to 1.

④ Each slave controller receives the above frame. Each controller checks the above select code against its own select code. The controller whose code matches clears its <WU> bit to 0.

⑤ The master controller transmits data to the specified slave controller (the controller whose SC0MOD0<WU> bit has been cleared to 0). The MSB (bit 8) of the data (<TB8>) is cleared to 0.

⑥ The other slave controllers (whose <WU> bits remain at 1) ignore the received data because their MSBs (bit 8 or <RB8>) are set to 0. disabling INTRX0 interrupts.
The slave controller whose <WU> bit = 0 can also transmit to the master controller. In this way it can signal the master controller that the data transmission from the master controller has been completed.

Setting example: To link two slave controllers serially with the master controller using the internal clock φ1 as the transfer clock.



- Setting the master controller

Main routin

```
PFCR    ← - - - - - - 0 1   } Set PF0 and PF1 to function as the TXD0 and RXD0 pins
PFFC    ← - - - - - - - 1   } respectively.
INTES0  ← X 1 0 0 X 1 0 1   Enable the INTTX0 interrupt and set it to Interrupt Level 4.
                            Enable the INTRX0 interrupt and set it to Interrupt Level 5.
SC0MOD0 ← 1 0 1 0 1 1 1 0   Set φ1 as the transmission clock for 9-Bit UART Mode.
SC0BUF  ← 0 0 0 0 0 0 0 1   Set the select code for slave controller 1.
```

INTTX0 interrupt

```
SC0MOD0 ← 0 - - - - - - -   Set TB8 to 0.
SC0BUF  ← * * * * * * * *   Set data for transmission.
```

- Setting the slave controller

Main routin

```
PFCR    ← - - - - - - 0 0   } Select PF1 and PF0 to function as the RXD0 and TXD0 pins
PFFC    ← - - - - - - - 1   } respectively (open-drain output).
INTES0  ← X 1 0 1 X 1 1 0   Enable INTRX0 and INTTX0.
SC0MOD0 ← 0 0 1 1 1 1 1 0   Set <WU> to 1 in 9-Bit UART Transmission Mode using φ1 as
                            the transfer clock.
```

INTRX0 interrupt

```
Acc ← SC0BUF
if Acc = select code
then SC0MOD0 ← - - - 0 - - - -   Clear <WU> to 0.
```

3.10 Serial Bus Interface (SBI)

The TMP92CW10 has a 2-channel serial bus interface which can operate in either Clocked-Synchronous 8-bit SIO Mode or I²C Bus Mode. The channels are referred to as SBI0 and SBI1. Since each channel operates in the same way, only SBI0 is explained here.

The serial bus interface is connected to an external device via PN1 (SDA0) and PN2 (SCL0) in the I²C Bus Mode; and via PN0 (SCK0), PN1 (SO0) and PN2 (SI0) in the Clocked-Synchronous 8-bit SIO Mode.

To set the mode, the pins should be set as follows:

	ODE <ODEN1, ODEN1>	PNCR <PN2C, PN1C, PN0C>	PNFC <PN2F, PN1F, PN0F>
I²C Bus Mode	11	11X	11X
Clocked Synchronous 8-Bit SIO Mode	XX	011 010	X11

X: Don't care

3.10.1 Configuration

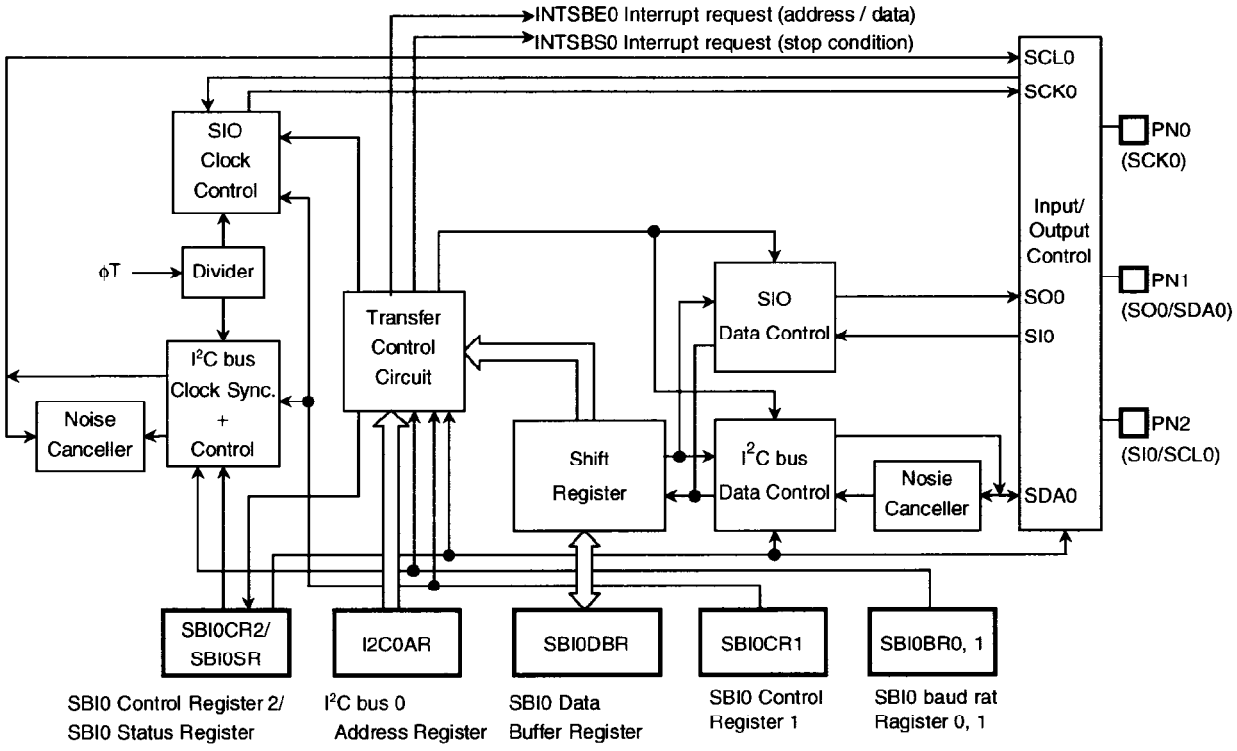


Figure 3.10.1 Serial Bus Interface 0 (SBI0)

3.10.2 Serial Bus Interface (SBI) Control

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

- Serial Bus Interface 0 Control Register 1 (SBI0CR1)
- Serial Bus Interface 0 Control Register 2 (SBI0CR2)
- Serial Bus Interface 0 Data Buffer Register (SBI0DBR)
- I^2C bus 0 Address Register (I2C0AR)
- Serial Bus Interface 0 Status Register (SBI0SR)
- Serial Bus Interface 0 Baud rate Register 0 (SBI0BR0)
- Serial Bus Interface 0 Baud rate Register 1 (SBI0BR1)

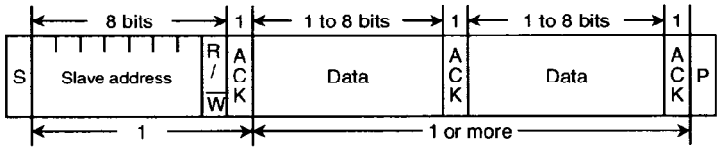
The above registers differ depending on a mode to be used.

Please refer to Section 3.10.4, I2C bus Mode Control and Section 3.10.7, Clocked-Synchronous 8-bit SIO Mode Control.

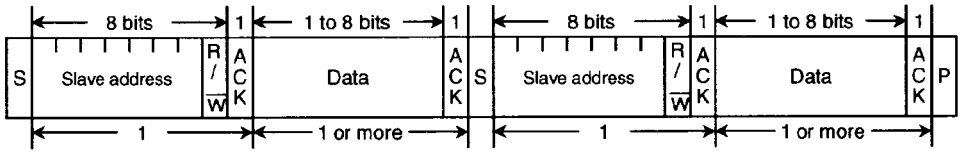
3.10.3 The Data Formats in the I^2C Bus Mode

The following data formats are used in the I^2C Bus Mode:

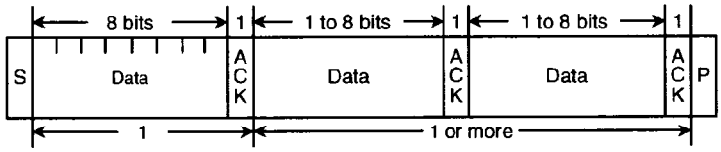
(a) Addressing format



(b) Addressing format (with restart)



(c) Free data format (data transferred from master device to slave device)



Note: S: Start condition
R/$\overline{W}$: Direction bit
ACK: Acknowledge bit
P: Stop condition

Figure 3.10.2 Data Format in the I^2C Bus Mode

3.10.4 I^2C Bus Mode Control Register

The following registers are used to control and monitor the operation status when using the serial bus interface (SBI) in the I^2C bus mode.

Seirial Bus Interface 0 Conrol Register 1

SBI0CR1 (0170H)

	7	6	5	4	3	2	1	0
Bit symbol	BC2	BC1	BC0	ACK	-	SCK2	SCK1	SCK0/ SWRMON
Read/Write	W			R/W		W		R/W
After Reset	0	0	0	0	-	0	0	0
Function	Number of transferred bits (Note 1)			Acknowledge mode specification 0: Not generate 1: Generate		Internal serial output clock frequency selection and software reset monitor (Note 2)		

SBI0CR1 must not be used with any of the read-modify-write instructions.

Internal serial clock selection <SCK2 to 0> for write operation.

000	n = 5	500	kHz
001	n = 6	277.8	kHz
010	n = 7	147.1	kHz
011	n = 8	75.8	kHz
100	n = 9	38.5	kHz
101	n = 10	19.4	kHz
110	n = 11	9.73	kHz
111	(res rved)	(reserved)	

CPU clock: fc
fc = 20 MHz
(output to SCL)
$fscl = \frac{fc}{2^n + 8}$ [Hz]

Software reset state monitor <SWRMON> for read operation.

0	Software reset in progress
1	Software reset not in progress

Clock pulse selection for acknowledge mode specification

0	No clock pulse generated as acknowledge signal
1	Clock pulse generated as acknowledge signal

Number of bits transferred

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

Note 1: Before switching to a clock-synchronous 8-bit SIO mode, set the <BC2 to 0> to "000".
Note 2: For the frequency of the SCL line clock, see 3.10.5 (3) Serial clock.

Figure 3.10.3 Registers for the I^2C Bus Mode (SBI0CR1)

Serial Bus Interface 0 Control Register 2

SBI0CR2 (0173H)

	7	6	5	4	3	2	1	0
Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
Read/Write	W				W (Note 1)		W (Note 1)	
After Reset	0	0	0	1	0	0	0	0
Function	Master/Slave selection	Transmitter/ Receiver selection	Start/Stop generation	Cancel INTSBE0 interrupt request	Serial bus interface operating mode selection (note 2) 00: Port Mode 01: SIO Mode 10: I2C Bus Mode 11: (reserved)		Write 10 then 01 to initiate a software reset.	

SBI0CR2 must not be used with any of the read-modify-write instructions.

Serial bus interface operating mode selection (Note 2)

00	Port Mode (Serial Bus Interface output disabled)
01	Clocke -Synchronous 8-Bit SIO Mode
10	I2C Bus Mode
11	(reserved)

INTSBE0 interrupt request

0	–
1	Cancel interrupt request

Start / stop generation

0	Generates the stop conditi n
1	Generates the start condition

Transmitter / receiver selection

0	Receiver
1	Transmitter

Master / Slave selection

0	Slave
1	Master

Note1: The value in this register can be read out from register SBI0SR.

Note2: Having confir ed that the bus is free, switch the mode to Port Mode.
Before switching between I^2C Bus Mode and Clocked-Synchronous 8-bit SIO Mode, first confirm that the input signals to the port are High.

Figure 3.10.4 Registers for the I^2C Bus Mode (SBI0CR2)

Serial Bus Interface 0 Status Register

SBI0SR (0173H)

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Function	Master/ Slave status monitor	Transmitter/ Receiver status monitor	I^2C bus status monitor	INTSBE0 interrupt request monitor	Arbitration lost detection monitor 0: – 1: Detected	Slave address match detection monitor 0: – 1: Detected	GENERAL CALL detection monitor 0: – 1: Detected	Last received bit monitor 0: "0" 1: "1"

SBI0SR must not be used with any of the read-modify-write instructions.

Last received bit monitor

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

GENERAL CALL detection monitor

0	–
1	GENERAL CALL detected

Slave address match detection monitor

0	–
1	Slave address match or GENERAL CALL detected

Arbitration lost detection monito

0	–
1	Arbitration lost

INTSBE0 interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I^2C bus status monitor

0	Free
1	Busy

Transmitter / receiver status monitor

0	Receiver
1	Transmitter

Master / Slave status monitor

0	Slave
1	Master

Note1:Writing in this register functions as SBI0CR2.

Figure 3.10.5 Registers for the I^2C Bus Mode (SBI0SR)

Serial Bus Interface 0 Baud Rate Regster 0

SBI0BR0 (0174H)

	7	6	5	4	3	2	1	0
bit Symbol	-	I2SBI0	-	-	-	-	-	-
Read/Write		R/W						
After Reset	0	0	-	-	-	-	-	-
Function	(Note) Fixed to "0"	IDLE2 0: Stop 1: Run						

Operation during IDLE 2 Mod

0	Stopped
1	Operating

Serial Bus Interface 0 Baud Rate Register 1

SBI0BR1 (0175H)

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	-	-	-	-	-	-	-
Read/Write	R/W							
After Reset	0	-	-	-	-	-	-	-
Function	Internal clock 0: Stop 1: Operate							

Baud rate clock control

0	Stopped
1	Operating

Sirial Bus Interface 0 Data Buffer Register

SBI0DBR (0171H)

Prohibit Read -modif -write

	7	6	5	4	3	2	1	0
Bit symbol	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
Read/Write	R (received)/W (transfer)							
After Reset	Undefined							

Note: When writing transmitted data, start from the MSB (bit 7).

I^2C Bus 0 Address Register

I2C0AR (0172H)

Prohibit Read -modif -write

	7	6	5	4	3	2	1	0
bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
After Reset	0	0	0	0	0	0	0	0
Function	Slave address selection for when device is operating as slave devic							Address recognition mode specification

Address recognition mod specification

0	Slave address recognition
1	Non slave address recognition

Figure 3.10.6 Registers for the I^2C Bus Mode (SBI0BR0, SBI0BR1, SBI0DBR, I2C0AR)

Seirial Bus Interface 1 Conrol Register 1

		7	6	5	4	3	2	1	0
SBI1CR1 (0178H)	Bit symbol	BC2	BC1	BC0	ACK	-	SCK2	SCK1	SCK0/ SWRMON
	Read/Write	W			R/W		W		R/W
	After Reset	0	0	0	0	-	0	0	0
Prohibit Read-modify-write	Function	Number of transferred bits (Note 1)			Acknowledge signal 0: Not generate 1: Generate		Internal serial output clock selection frequency and software reset monitor (Note 2)		

Internal serial clock selection <SCK2 to 0> for write operation

000	n = 5	500 kHz	CPU clock: fc fc = 20 MHz (internal SCL output) $fscl = \frac{fc}{2^n + 8}$ [Hz]
001	n = 6	277.8 kHz	
010	n = 7	147.1 kHz	
011	n = 8	75.8 kHz	
100	n = 9	38.5 kHz	
101	n = 10	19.4 kHz	
110	n = 11	9.73 kHz	
111	(reserved)	(reserved)	

Software reset state monitor <SWRMON> for read operation

0	Software reset in progress
1	Software reset not in progress

Clock pulse selection for acknowledge signal

0	No clock pulse generated for acknowledge signal
1	Clock pulse generated for acknowledge signal

Number of bits transferred

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

Note 1: Before switching to a clock-synchronous 8-bit SIO mode, set the <BC2 to 0> to "000"
Note 2: For the frequency of the SCL line clock, see 3.10.5 (3) Serial clock.

Figure3.10.7 Registers for the I^2C Bus Mode (SBICR1)

Serial Bus Interface 1 Control Register 2

		7	6	5	4	3	2	1	0
SBI1CR2 (017BH)	Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
	Read/Write	W				W (Note 1)		W (Note 1)	
	After Reset	0	0	0	1	0	0	0	0
Prohibit Read-modify-write	Function	Master/Slave selection	Transmitter/ Receiver selection	Start/Stop generation	Cancel INTSBE1 interrupt request	Serial bus interface operating mode selection (note 2) 00: Port Mode 01: SIO Mode 10: I2C Bus Mode 11: (reserved)		Write 10 then 01 to initiate a software reset.	

Serial bus interface operating mode selection (Note 2)

00	Port Mode (Serial Bus Interface output disabled)
01	Clocke -Synchronous 8-Bit SIO Mode
10	I^2C Bus Mode
11	(reserved)

INTSBE1 interrupt request

0	–
1	Cancel interrupt request

Start / stop generation

0	Generates the stop conditi n
1	Generates the start condition

Transmitter / receiver selection

0	Receiver
1	Transmitter

Master / Slave selection

0	Slave
1	Master

Note1: Reading this register function as SBI1SR register.
Note2: Having confirmed that the bus is free, switch the mode to Port Mode.
Before switching between I2C Bus Mode and Clock-Synchronous 8-bit SIO Mode, first confirm that the input signals to the port are High.

Figure 3.10.8 Registers for the I^2C Bus Mode (SBI1CR2)

Serial Bus Interface 1 Status Register

	7	6	5	4	3	2	1	0
SBI1SR (017BH) bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Prohibit Read-modif -write Function	Master/ Slave status monitor	Transmitter/ Receiver status monitor	I^2C bus status monitor	INTSBE1 interrupt request monitor	Arbitration lost detection monitor 0: — 1: Detected	Slave address match detection monitor 0: — 1: Detected	GENERAL CALL detection monitor 0: — 1: Detected	Last received bit monitor 0: "0" 1: "1"

Last received bit monitor

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

GENERAL CALL detection monitor

0	—
1	GENERAL CALL detected

Slave address match detection monitor

0	—
1	Slave address match or GENERAL CALL detected

Arbitration lost detection monito

0	—
1	Arbitration lost

INTSBE1 interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I^2C bus status monitor

0	Free
1	Busy

Transmitter / receiver status monitor

0	Receiver
1	Transmitter

Master / Slave status monitor

0	Slave
1	Master

Note1: Writing in this register functions as SBI1CR2.

Figure 3.10.9 Registers for the I^2C Bus Mode (SBI1SR)

Serial Bus Interface 1 Baud Rate Regster 0

SBI1BR0 (017CH)

	7	6	5	4	3	2	1	0
bit Symbol	-	I2SBI1	-	-	-	-	-	-
Read/Write		R/W						
After Reset	0	0	-	-	-	-	-	-
Function	Fixed to "0"	IDLE2 0: Stop 1: Run						

Operation during IDLE 2 Mod

0	Stopped
1	Operating

Serial Bus Interface 1 Baud Rate Register 1

SB1BR1 (017DH)

	7	6	5	4	3	2	1	0
Bit symbol	P4EN							
Read/Write	R/W							
After Reset	0							
Function	Internal clock 0: Stop 1: Operate							

Baud rate clock control

0	Stopped
1	Operating

Sirial Bus Interface 1 Data Buffer Register

SBI1DBR (0179H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
Read/Write	R (received)/W (transfer)							
After Reset	Undefined							

Note: When writing transmitted data, start from the MSB (bit 7).

I^2C Bus 1 Address Register

I2C1AR (017AH)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
After Reset	0	0	0	0	0	0	0	0
Function	Slave address selection for when device is operating as slave devic							Address recognition mode specification

Address recognition mode specification

0	Slave address recognition
1	No slave address recognition

Figure 3.10.10 Registers for the I^2C Bus Mode (SBI1BR0, SBI1BR1, SBI1DBR, I2C1AR)

3.10.5 Control in I²C Bus Mode

(1) Specifying acknowledge mode

To operate the device in the acknowledge mode set the SBI0CR1<ACK> to "1". When operating in the master mode this device generates an additional clock pulse as an acknowledge signal; when operating in the slave mode it counts a clock pulse as an acknowledge signal. In the transmitter mode the SDA0 pin is released during the clock pulse cycle so that it can receive the acknowledge signal from the receiver. In the receiver mode the SDA0 pin is set to the low-level during the clock pulse cycle in order to generate the acknowledge signal.

To operate the device in non-acknowledge mode, clear the SBI0CR1<ACK> to "0". When operating in the master mode this device does not generate a clock pulse as an acknowledge signal; when operating in the slave mode it does not count a clock pulse as an acknowledge signal.

(2) Number of transfer bits

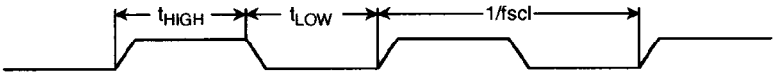
The SBI0CR1<BC2 to 0> setting determines the number of data bits to be transmitted or received.

Since the SBI0CR1<BC2 to 0> is cleared to "000" on start-up, a slave address and direction bit transmissions are executed in 8 bits. Other than these, the <BC2 to 0> retains a specified value.

(3) Serial clock

① Clock source

The SBI0CR1 <SCK2 to 0> is used to specify the maximum transfer frequency for output on the SCL0 pin in the master mode.



$t_{LOW} = 2^n/f_{SBI}$

$t_{HIGH} = 2^n/f_{SBI} + 8/f_{SBI}$

$fscl = 1/(t_{Low} + t_{HIGH})$

$= \frac{f_{SBI}}{2^n + 8}$

SBI0CR1 <SCK2 to 0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Note: f_{SBI} is the clock f_C

Figure 3.10.11 Clock Source

② Clock synchronization

In the I²C bus mode, in order to wired-AND a bus, a master device which pulls down a clock line to the low-level, in the first place, invalidate a clock pulse of another master device which generates a high-level clock pulse. The master device with a high-level clock pulse needs to detect the situation and implement the following procedure.
This device has a clock synchronization function which allows normal data transfer even when more than one master exists on the bus.
The following example explains the clock synchronization procedures used when there are two masters present on the bus.

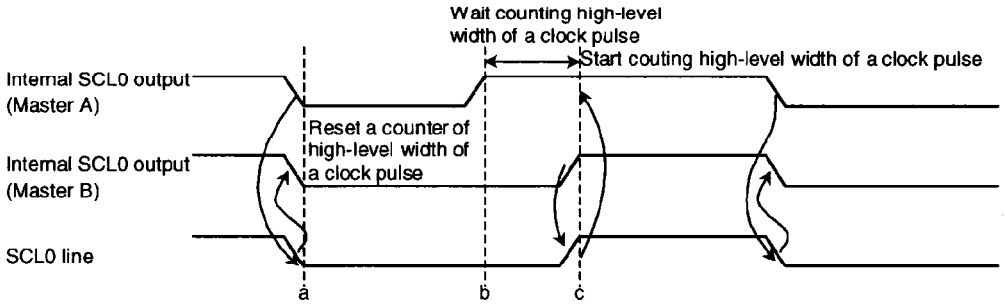


Figure 3.10.12 Clock Synchronization

When Master A pulls the internal SCL0 output to the low-level at point "a", the bus's SCL0 line goes to the low-level. After detecting this, Master B resets a counter of high-level width of an own clock pulse and sets the internal SCL0 output the low-level.
Master A finishes counting low-level width of an own clock pulse at point "b" and sets the internal SCL0 output to the high-level. Since Master B is holding the bus 's SCL0 line the low-level, Master A waits for counting high-level width of an own clock pulse. After Master B has finished counting low-level width of an own clock pulse at point"c" and Master A detects the SCL0 line of the bus at the high-level, and starts counting high-level of an own clock pulse.
The clock pulse on the bus is determined by the master device with the shortest high-level width and the master device with the longest low-level width from among those master devices connected to the bus.

(4) Slave address and address recognition mode specification

When this device is to be used as a slave device, set the slave address <SA6 to 0> and <ALS> in I2C0AR. Clear the <ALS> to "0" for the address recognition mode.

(5) Master/slave selection

To operate this device as a master device set the SBI0CR2<MST> to "1". To operate it as a slave device clear the SBI0CR2<MST> to "0". The <MST> is cleared to "0" in hardware when a stop condition is detected on the bus or when arbitration is lost.

(6) Transmitter/receiver selection

To operate this device as a transmitter set the SBI0CR2<TRX> to "1". To operate it as a receiver clea r the SBI0CR2<TRX> to "0". When data with an addressing format is transferred in the slave mode, when a slave address with the same value that an I20AR or a GENERAL CALL is received (all 8-bit data are "0" after a start condition), the <TRX> is set to "1" in hardware if the direction bit ($R/\overline{W}$) sent from the master device is "1", and is cleared to "0" in hardware if the bit is "0". In the master mode, when an acknowledge signal is returned from the slave device, the <TRX> is cleared to "0" in hardware if the value of the transmitted direction bit is "1", and is set to "1" in hardware if the value of the bit is "0". If an acknowledge signal is not returned, the current state is maintained.
The <TRX> is cleared to "0" in hardware when a stop condition is detected on the I²C bus or when arbitration is lost.

(7) Start/Stop condition generation

When the SBI0SR<BB> = "0", 8-bit data set in SBI0DBR is output on the bus after generating a start condition by writing "1111" to the SBI0CR2 <MST, TRX, BB, PIN>. It is necessary to set transmitted data to the data buffer register (SBI 0DBR) and set "1" to the <ACK> beforehand.

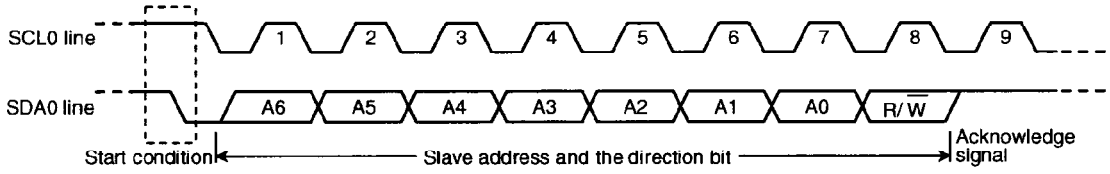


Figure 3.10.13 Start Condition Generation and Slave Address Generation

When the SBI0SR<BB> = "1", the sequence for generating a stop condition can be initiated by writing "111" to the SBI0CR2<MST,TRX,PIN> and writing "0" to the SBI0CR2<BB>. Do not modify the contents of the SBI0CR2<MST, TRX, BB, PIN> until a stop condition has been generated on the bus.

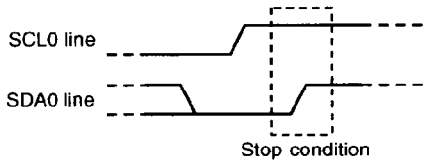


Figure 3.10.14 Stop Condition Generation

The state of the bus can be ascertained by reading the contents of the SBI0SR<BB>. The SBI0SR<BB> will be set to "1" if a start condition has been detected on the bus ,and will be cleared to "0" if a stop condition has been detected.

(8) Interrupt service requests and interrupt cancellation

When a serial bus interface interrupt request0 by transfer of the slave address or the data (INTSBE0) is generated, the SBI0SR<PIN> is cleared to "0". The SCL0 line is pulled down to the low-level while the <PIN> = "0".

The <PIN> is cleared to "0" when a single word of data is transmitted or received. Either writing data to or reading data from SBI0DBR sets the <PIN> to "1".

The time from the <PIN> being set to "1" until the release of the SCL0 line is t_{LOW}.

In the address recognition mode (i.e. when <ALS> = "0"), the <PIN> is cleared to "0" when the slave address matches the value set in I2C0AR or when a GENERAL CALL is received (all 8-bit data are "0" after a start condition). Although the SBI0CR2<PIN> can be set to "1" by a program, writing "0" to the SBI0CR2<PIN> does not clear it to "0".

(9) Serial bus interface operation mode selection

The SBI0CR2<SBIM1 to 0> is used to specify the serial bus interface operation mode. Set the SBI0CR2<SBIM1 to 0> to "10" when the device is to be used in I²C Bus Mode.

Switch a mode to port after confirming a bus is free.

(10) Arbitration lost detection monitor

Since more than one master device can exist simultaneously on the bus in I²C Bus Mode, a bus arbitration procedure has been implemented in order to guarantee the integrity of transferred data.

Data on the SDA0 line is used for I²C bus arbitration.

The following example illustrates the bus arbitration procedure when there are two master devices on the bus. Master A and Master B output the same data until point "a". After Master A outputs "L" and Master B, "H", the SDA0 line of the bus is wire-AND and the SDA0 line is pulled down to the low level by Master A. When the SCL0 line of the bus is pulled up at point "b", the slave device reads the data on the SDA0 line, that is, data in Master A. Data transmitted from Master B becomes invalid. The Master B state is known as "ARBITRATION LOST". Master B device which loses arbitration releases the internal SDA0 output in order not to affect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

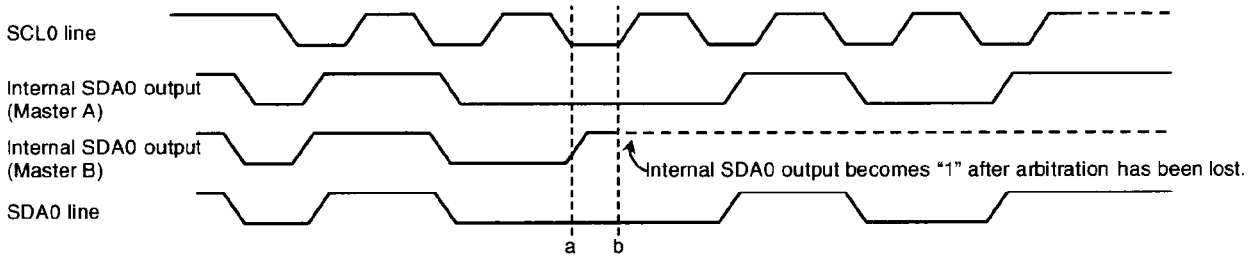


Figure 3.10.15 Arbitration Lost

This device compares the levels on the bus's SDA0 line with those of the internal SDA0 output on the rising edge of the SCI0 line. If the levels do not match, arbitration is lost and the SBI0SR<AL> is set to "1".

When the <AL> is set to "1", the SBI0SR<MST,TRX> are cleared to "00" and the mode is switched to a slave receiver mode. This device generates the clock pulse until data is transmitted when the <AL> is "1".

The <AL> is cleared to "0" when data is written to or read from SBI0DBR or when data is written to SBI0CR2.

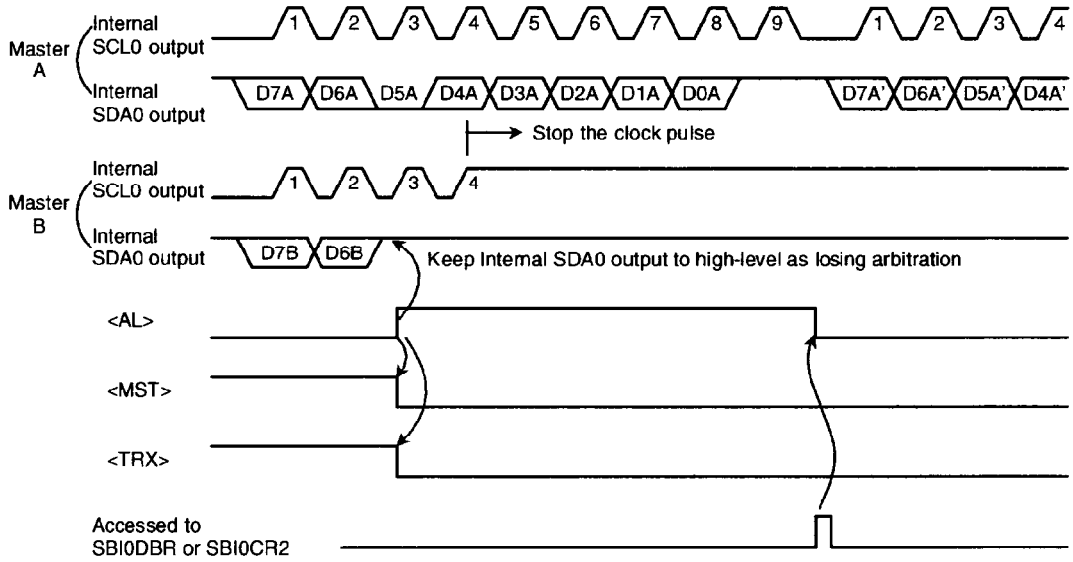


Figure 3.10.16 Example of a Master Device B
(D7A = D7B, D6A = D6B)

(11) Slave address match detection monitor

The SBI0SR<AAS> is set to "1" in the slave mode, in the address recognition mode (i.e. when the I2C0AR<ALS> = "0"), when a GENERAL CALL is received, or when a slave address matches the value set in I2C0AR. When the I2C0AR<ALS> = "1", the SBI0SR<AAS> is set to "1" after the first word of data has been received. The SBI0SR<AAS> is cleared to "0" when data is written to or read from the data buffer register SBI0DBR.

(12) GENERAL CALL detection monitor

The SBI0SR<AD0> is set to "1" in the slave mode, when a GENERAL CALL is received (all 8-bit received data is "0", after a start condition). The SBI0SR<AD0> is cleared to "0" when a start condition or stop condition is detected on the bus.

(13) Last received bit monitor

The value on the SDA0 line detected on the rising edge of the SCL0 line is stored in the SBI0SR<LRB>. In the acknowledge mode, immediately after an INTSBE0 interrupt request has been generated, an acknowledge signal is read by reading the contents of the SBI0SR<LRB>.

(14) Software Reset function

The software Reset function is used to initialize the SBI circuit, when SBI is rocked by external noises, etc.

An internal Reset signal pulse can be generated by setting SBI0CR2<SWRST1 to 0> to "10" and "01". This initializes the SBI circuit internally. All control registers and status registers are initialized as well.

The SBI0CR2<SWRST1 to 0> is automatically cleared to "00" after the SBI circuit has been initialized.

(15) Serial Bus Interface Data Buffer Register (SBI0DBR)

The received data can be read andthe transferred data can be writtenby reading or writing the SBI0DBR.

When the start condition has been generated in the master mode. the slave address and the direction bit are set in this register.

(16) I²C Bus Address Register (I2C0AR)

I2C0AR<SA6 to 0> is used to set the slave address when this device functions as a slave device.

The slave address output from the master device is recognizedby setting I2C0AR<ALS> is set to "0". The data format is the addressing format. When the slave address in not recognized at the <ALS> is set to "1". the data format is the free data format.

(17) Baud Rate Register (SBI0BR1)

Write "1" to the SBI0BR1<P4EN> before operation commences.

(18) Setting register for IDLE2 mode operation (SBI0BR0)

The setting of SBI0BR0<I2SBI0> determines whether the device is operating or is stopped in IDLE2 Mode. Therefore. setting <I2SBI0> is necessary before the HALT instruction is executed.

3.10.6 Data Transfer in I²C Bus Mode

(1) Device Initialization

Set the SBI0BR1<P4EN> and the SBI0CR1<ACK,SCK2 to 0>. Set the SBI0BR1<P4EN> to "1" and clear bits 7 to 5 and 3 of the SBI0CR1 to "0".
Set a slave address in I2C0AR<SA6 to 0> and the I2C0AR<ALS> (<ALS> = "0" when an addressing format.)
For specifying the default setting to a slave receiver mode, clear "000" to the <MST, TRX, BB>, set "1" to the <PIN>, set "10" to the <SBIM1 to 0>and set "00" to the <SWRST1 to 0>.

(2) Start Condition Generation and Slave Address Generation

① Master mode

In the master mode the start condition and the slave address are generated as follows.
Check a bus free status (when <BB>="0").
Set the SBI0CR1<ACK> to "1" (acknowledge mode) and specify a slave address and a direction bit to be transmitted to the SBI0DBR.
When the <BB> is "0", the start condition is generated by writing "1111" to the SBI0CR2<MST,TRX,BB,PIN>. Subsequently to the start condition, nine clocks are output from the SCL0 pin. While eight clocks are output, the slave address and the direction bit which are set to the SBI0DBR. At the 9th clock pulse the SDA0 line is released and the acknowledge signal is received from the slave device.
An INTSBE0 interrupt request occurs on the falling edge of the ninth clock pulse. The <PIN> is cleared to "0". In the master mode the SCL0 pin is pulled down to the low-level while the <PIN> is "0". When an INTSBE0 interrupt request occurs, the value of <TRX> is changed according to the direction bit setting only if the slave device returns an acknowledge signal.

② Slave mode

In the slave mode the start condition and the slave address are received.
After the start condition has been received from the master device, while eight clocks are output from the SCL0 pin, the slave address and the direction bit which are output from the master device are received.
When a GENERAL CALL or an address matching the slave address set in I2C0AR is received, the SDA0 line is pulled down to the low level at the 9th clock pulse and an acknowledge signal is output.
An INTSBE0 interrupt request occurs on the falling edge of the ninth clock pulse. The <PIN> is cleared to "0". In the slave mode the SCL0 line is pulled down to the low-level while the <PIN> = "0". When an interrupt request occurs, the value of <TRX> is changed according to the direction bit setting only if the slave device returns an acknowledge signal.

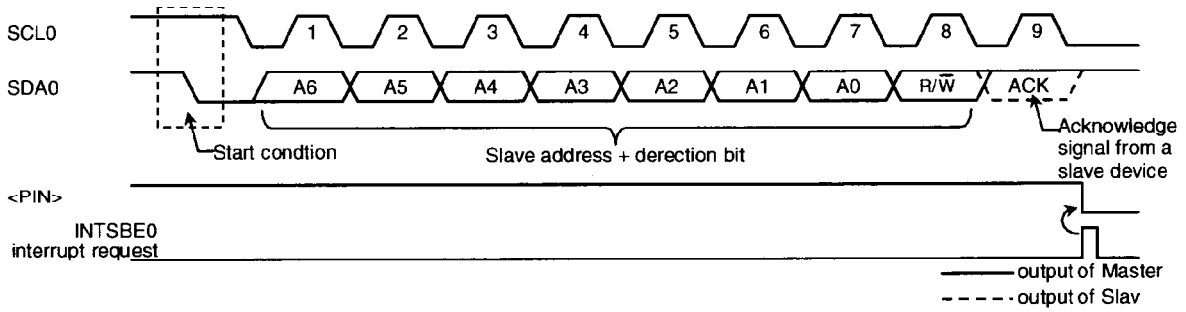


Figure 3.10.17 Start Condition Generation and Slave Address Transfer

(3) 1-word Data Transfer

Check the <MST> setting using an INTSBE0 interrupt process after the transfer of each word of data is completed and determine whether the device is in the master mode or the slave mode.

① When the <MST> is "1" (Master mode)

Check the <TRX> setting and determine whether the device is in the transmitter mode or the receiver mode.

When the <TRX> is "1" (Transmitter mode)

Check the <LRB> setting. When the <LRB> ≠"1", there is no receiver requesting data. Implement the process for generating a stop condition (see Section 3.10.6 (4)) and terminate data transfer.
When the <LRB> = "0", the receiver is requesting new data. When the next transmitted data is 8 bits, write the transmitted data tothe SBI0DBR. When the next transmitted data is other than 8 bits, set the <BC2 to 0>, set the <ACK> to "1" and write the transmitted data to the SBI0DBR. After the data has been written,the <PIN> is set to "1", a serial clock pulse is generated to trigger transfer of the next word of data via the SCL 0 pin, and the word is transmitted. After the data has been transmitted, an INTSBE0 interrupt request is generated. The <PIN> is set to "0" and the SCL0 line is pulled down to the low-level. If the length of the data to be transferred is greater than one word, repeat the latter steps of the procedure, starting from the check of the <LRB> setting.

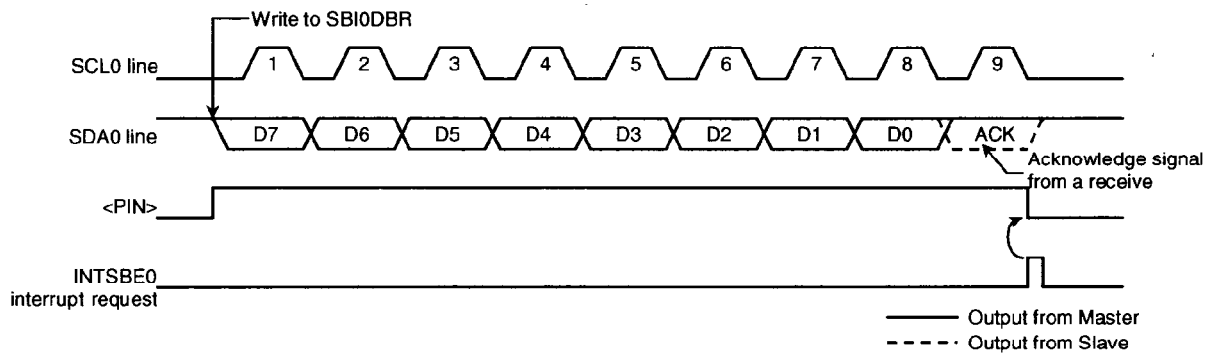


Figure 3.10.18 Example in which <BC2 to 0> = "000" and <ACK> = "1" in Transmitter Mode

When the <TRX> is "0" (Receiver mode)

When the next transmitted data is 8 bits, write the transmitted data the SBI0DBR. When the next transmitted data is other than 8 bits, set the <BC2 to 0> again. Set the <ACK> to "1" and read the received data from the SBI0DBR so as to release the SCL0 line (the value of data which is read immediately after a slave address is sent is undefined). After the data has been read, the <PIN> is set to "1". This device outputs a serial clock pulse on SCL0 line to transfer new 1-word of data and set the SDA0 pin to "0", when the acknowledge signal is set to the low-level at the final bit.

An INTSBE0 interrupt request is generated and the <PIN> is set to "0". Then this device pulls down the SCL0 pin to the low-level. This device outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from SBI0DBR.

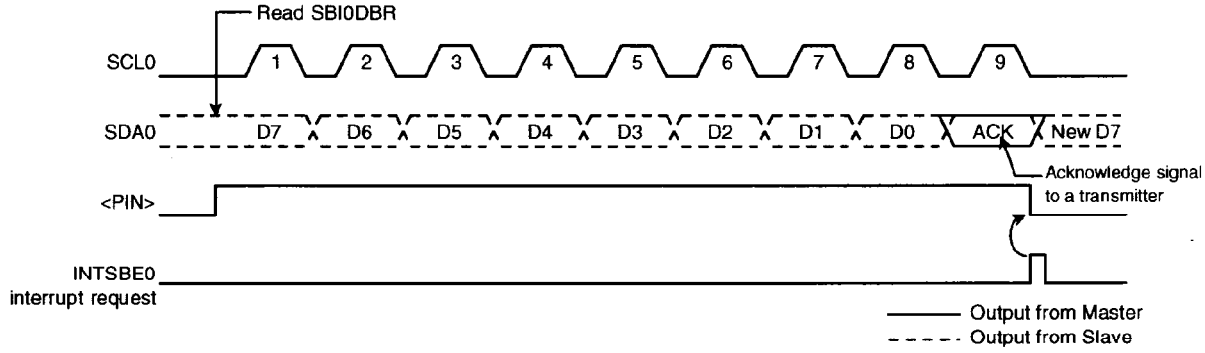


Figure 3.10.19 Example of when <BC2 to 0> = "000", <ACK> = "1" in Receiver Mode

In order to terminate the transmission of data to a transmitter, clear the <ACK> to "0" before reading data which is 1-word before the last data to be received. The last data does not generate a clock pulse for the acknowledge signal. After the data has been transmitted and an interrupt request has been generated, set the <BC2 to 0> to "001" and read the data. This device generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA0 line on a bus keeps the high-level. The transmitter receives the high-level signal as an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, this device generates a stop condition (see Section 3.10.6 (4)) and terminates data transfer. Because of a stop condition generation, an INTSBS0 interrupt request occurs.

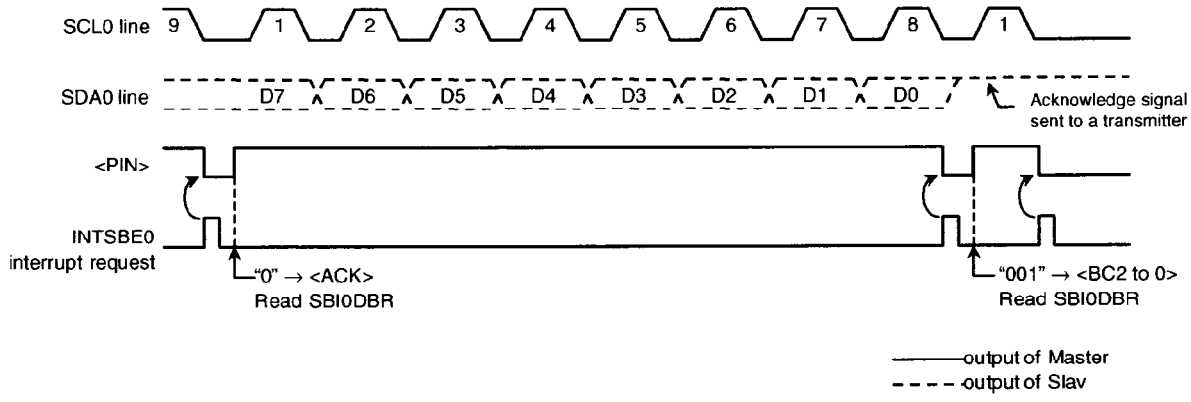


Figure 3.10.20 Termination of Data Transfer in Master Receiver Mode

② When the <MST> is "0" (Slave mode)

In the slave mode, an INTSBE0 interrupt request occurs when this device receives a slave address or a GENERAL CALL from the master device, or when a GENERAL CALL is received and data transfer is complete, or after matching a received slave address. In the master mode, this device operates in a slave mode if it is losing arbitration. An INTSBE0 interrupt request occurs when word data transfer terminates after losing arbitration. When an INTSBE0 interrupt request occurs,the <PIN> is cleared to "0", and the SCL0 pin is pulled down to the low-level. Either reading data to or writing data from the SBI0DBR, or setting the <PIN> to "1" releases the SCL0 pin after taking t_{LOW} time.

In the slave mode, this device operates either in normal slave mode or in slave mode after losing arbitration.

Check the SBI0SR<AL>, <TRX>, <AAS> and <AD0> and implements processes according to conditions listed in the next table.

Table 3.10.1 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	This device loses arbitration when transmitting a slave address and receives a slave address of which the value of the direction bit sent from another master is "1".	Set the number of bits in 1-word to the <BC2 to 0> and write the transmitted data to the SBI0DBR.
	0	1	0	In the salve receiver Mode, this device receives a slave address of which the value of the direction bit sent from the master is "1".	
		0	0	In the salve transmitter mode, 1-word data is transmitted.	Check the <LRB>. If the <LRB> is set to "1", set the <PIN> to "1" since the receiver does not request the next data. Then, clear the <TRX> t "0" to release the bus. If the <LRB> is cleared to "0", set the number of bits in a word to the <BC2 to 0> and write transmitted data to the SBI0DBR since the receiver requests next data.
0	1	1	1/0	This device loses arbitration when transmitting a slave address and receives a GENERAL CALL or slave address of which the value of the direction bit sent from another master is "0".	Read the SBI0DBR for setting the <PIN> to "1" (reading dummy data) or set the <PIN> to "1".
		0	0	This device loses arbitration when transmitting a slave address or data and terminates transferring word data.	
	0	1	1/0	In the slave receiver mode, this device receives a GENERAL CALL or slave address of which the value of the direction bit sent from the master is "0".	
		0	1/0	In the slave receiver mode, the device terminates receiving 1-word data.	Set the number of bits in a word to th <BC2 to 0> and read received data from the SBI0DBR.

(4) Stop condition generation

When the SBI0SR<BB> is "1", the sequence of generating a stop condition is started by setting "111" to the SBI0CR2<MST,TRX,PIN> and "0" to the SBI0CR2<BB>. Do not modify the contents of the SBI0CR2<MST,TRX,PIN,BB> until a stop condition is generated on a bus. When a SCL0 line of bus is pulled down by other devices, this device generates a stop condition after they release a SCL0 line and the SDA0 becomes "1". An INTSBS0 interrupt request occurs at the timing of the SBI0SR<BB> becomes "0".

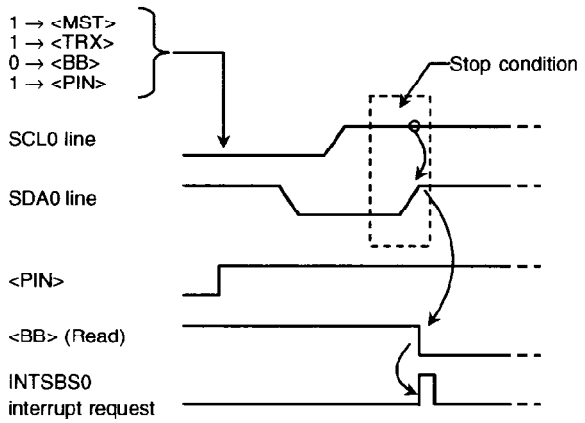


Figure 3.10.21 Stop Condition Generation

(5) Restart

Restart is used during data transfer between a master device and a slave device to change the data transfer direction. The following description explains how to restart when this device is in the master mode.

Clear the SBI0CR2<MST,TRX,BB> to "000" and set the SBI0CR2<PIN> to "1" to release the bus. The SDA0 line remains the high-level and the SCL0 pin is released. Since a stop condition is not generated on the bus, other devices assume the bus to be in a busy state. Check the SBI0SR<BB> until it becomes "0" to check that the SCL0 pin of this device is released. Check the <LRB> until it becomes 1 to check that the SCL0 line on a bus is not pulled down to the low-level by other devices. After confirming that the bus stays in a free state, generate a start condition with procedure described in 3.10.6 (2).

In order to meet set-up time when restarting, take at least 4.7μs of waiting time by software from the time of restarting to confirm that the bus is free until the time to generate the start condition.

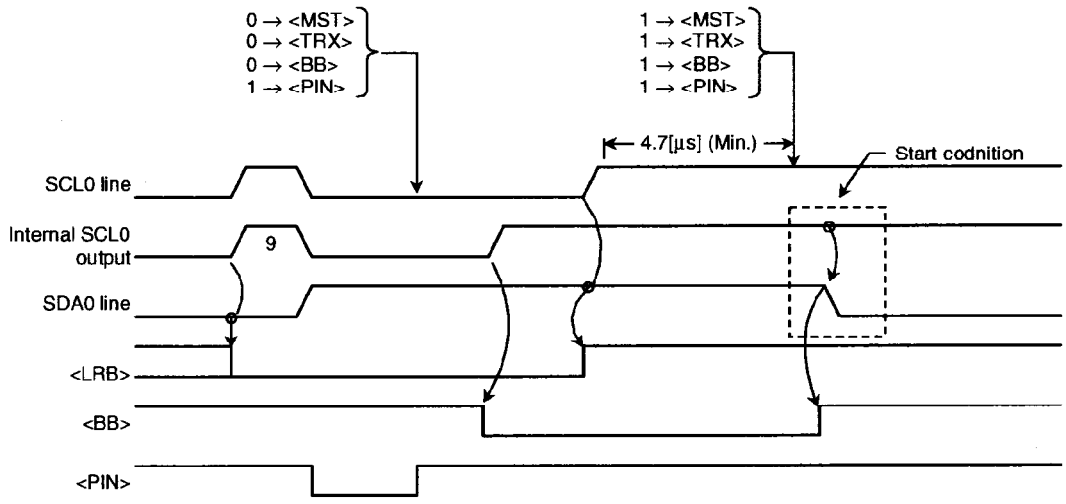


Figure 3.10.22 Timing Diagram when Restarting

3.10.7 Clocked Synchronous 8-Bit SIO Mode control

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

Serial Bus Interface 0 Control Register 1

SBI0CR1 (0170H)	7	6	5	4	3	2	1	0
Bit symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0/ SWRMON
Read/Writ	W					W		R/W
After Reset	0	0	0	0		0	0	0
Function	Transfer start 0: stop 1: start	Continue/ abort transfer 0: Continue transfer 1: Abort transfer	Transfer mode select 00: Transmit Mode 01: (reserved) 10: Transmit/Receive Mode 11: Receive Mode			Serial clock selection and Reset monitor		

Prohibit Read-modify-write

Serial clock selection <SCK2 to 0> @ writ

000	n = 4	1.25 MHz
001	n = 5	625 kHz
010	n = 6	313 kHz
011	n = 7	156 kHz
100	n = 8	78.1 kHz
101	n = 9	39.1 kHz
110	n = 10	19.5 kHz
111	–	external clock : SCK0

CPU clcok: fc
fc = 20 MHz
(output to SCK pin)
$fscl = \frac{fc}{2^n}$ [Hz]

Software reset state monitor <SWRMON> @ read

0	During software reset
1	Not during software reset

Transfer mode selection

00	8-bit transmit mod
01	(reserved)
10	8-bit transmit / receive mod
11	8-bit receive mode

Continue / abort transfer

0	Continue transfer
1	Abort transfer (automatically cleared after transfer aborted)

Indicate transfer start / stop

0	Stop
1	Start

Note: Set the tranfer mode and the serial clock after setting <SIOS> to "0" and <SIOINH> to "1".

Serial Bus interface 0 Data Buffer Register

SBI0DBR (0171H) Prohibit Read-modify-write	7	6	5	4	3	2	1	0
Bit symbol	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
Read/Writ	R (receiver) / W (transfer)							
After Reset	Undefined							

Figure 3.10.23 Register for the SIO Mode

Serial Bus Interface 0 Control Register 2

		7	6	5	4	3	2	1	0
SBI0CR2 (0173H)	Bit symbol					SBIM1	SBIM0		
	Read/Writ					W			
	After Reset					0	0		
Prohibit Read-modify-write	Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (reserved)			

Serial bus interface operation mode selection

00	Port mode (serial bus interface output disabled)
01	Clocked-Synchronous 8-bit SIO mod
10	I^2C bus mode
11	(reserved)

Note: Set the SBI0CR1<BC2 to 0> "00 " before switching to a clocked-synchronous 8-bit SIO mode.

Serial Bus Interface 0 Status Register

		7	6	5	4	3	2	1	0
SBI0SR (0173H)	bit Symbol					SIOF	SEF		
	Read/Writ					R			
	After reset					0	0		
	Function					Serial transfer operation status monitor	Shift operation status monitor		

Shift operation status monitor

0	Shift operation terminated
1	Shift operation in progress

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in progress

Figure 3.10.24 Registers for the SIO Mode

Serial Bus Interface 0 Baud Rate Register 0

SBI0BR0 (0174H)

	7	6	5	4	3	2	1	0
Bit symbol	-	I2SBI0						
Read/Writ		R/W						
After Reset	0	0						
Function	(Note) Fixed to "0"	IDLE2 0: STOP 1: RUN						

Operation during IDLE 2 mode

0	Stop
1	Operate

Serial Bus Interface 0 Baud Rate Register 1

SBI0BR1 (0175H)

	7	6	5	4	3	2	1	0
Bit symbol	P4MON/ P4EN							
Read/Writ	R/W							
After Reset	0							
Function	Internal clock 0: Stop 1: Operate							

Baud rate clock control

0	Stop
1	Operate

Figure 3.10.25 Registers for the SIO Mode

Serial Bus Interface 1 Control Register 1

SBI1CR1 (0178H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0/ SWRMON
Read/Writ	W					W		R/W
After Reset	0	0	0	0		0	0	0
Function	Transfer start 0: stop 1: start	Continue/ abort transfer 0: Continue transfer 1: Abort transfer	Transfer mode select 00: Transmit Mode 01: (reserved) 10: Transmit/Receive Mode 11: Receive Mode			Serial clock selection and Reset monitor		

Serial clock selection <SCK2 to 0> @ writ

000	n = 4	1.25 MHz
001	n = 5	625 kHz
010	n = 6	313 kHz
011	n = 7	156 kHz
100	n = 8	78.1 kHz
101	n = 9	39.1 kHz
110	n = 10	19.5 kHz
111	–	external clock : SCK1

CPU clcok: fc
fc = 20 MHz
(output to SCK pin)
$fscl = \frac{fc}{2^n}$ [Hz]

Software reset state monitor <SWRMON> @ read

0	During software reset
1	Not during software reset

Transfer mode selection

00	8-bit transmit mod
01	(reserved)
10	8-bit transmit / receive mod
11	8-bit receive mode

Continue / abort transfer

0	Continue transfer
1	Abort transfer (automatically cleared after transfer aborted)

Indicate transfer start / stop

0	Stop
1	Start

Note: Set the tranfer mode and the serial clock after setting <SIOS> to "0" and <SIOINH> to "1".

Serial Bus interface 1 Data Buffer Register

SBI1DBR (0179H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
Read/Writ	R (receiver) / W (transfer)							
After Reset	Undefined							

Figure 3.10.26 Register for the SIO Mode

Serial Bus Interface 1 Control Register 2

		7	6	5	4	3	2	1	0
SBI1CR2 (017BH)	Bit symbol					SBIM1	SBIM0		
	Read/Writ					W			
	After Reset					0	0		
Prohibit Read-modify-write	Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (reserved)			

Serial bus interface operation mode selection

00	Port mode (serial bus interface output disabled)
01	Clocked-Synchronous 8-bit SIO mod
10	I^2C bus mode
11	(reserved)

Note: Set the SBI1CR1<BC2 to 0> "00 " before switching to a clocked-synchronous 8-bit SIO mode.

Serial Bus Interface 1 Status Register

		7	6	5	4	3	2	1	0
SBI1SR (017BH)	bit Symbol					SIOF	SEF		
	Read/Writ					R			
	After reset					0	0		
	Function					Serial transfer operation status monitor	Shift operation status monitor		

Shift operation status monitor

0	Shift operation terminated
1	Shift operation in progress

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in progress

Figure 3.10.27 Registers for the SIO Mode

Serial Bus Interface 1 Baud Rate Register 0

SBI1BR0 (017CH)

	7	6	5	4	3	2	1	0
Bit symbol	-	I2SBI1						
Read/Writ		R/W						
After Reset	0	0						
Function	(Note) Fixed to "0"	IDLE2 0: STOP 1: RUN						

Operation during IDLE 2 mode

0	Stop
1	Operate

Serial Bus Interface 1 Baud Rate Register 1

SBI1BR1 (017DH)

	7	6	5	4	3	2	1	0
Bit symbol	P4MON/ P4EN							
Read/Writ	R/W							
After Reset	0							
Function	Internal clock 0: Stop 1: Operate							

Baud rate clock control

0	Stop
1	Operate

Figure 3.10.28 Registers for the SIO Mode

(1) Serial Clock

① Clock source

SBI0CR1<SCK2 to 0> is used to select the following functions:

Internal Clock

In an internal clock mode, any of seven frequencies can be selected. The serial clock is output to the outside on the SCK0 pin. The SCK0 pin becomes a high-level when data transfer starts. When the device is writing (in the transmit mode) or reading (inthe receive mode) data cannot follow the serial clock rate, an automatic wait function is executed to stop the serial clock automatically and holds the next shift operation until reading or writing is complete.

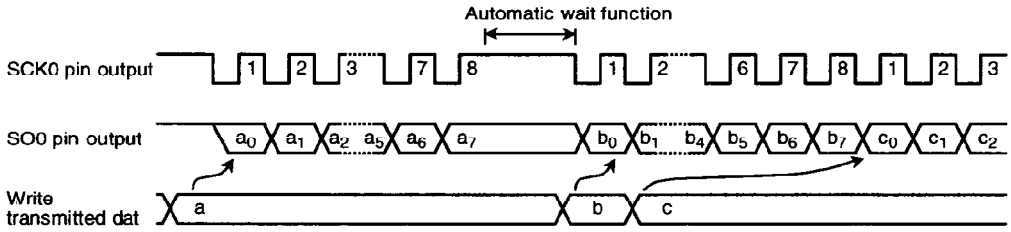


Figure 3.10.29 Automatic-wait Function

External clock (<SCK2 to 0> = "111")

An external clock input via the SCK0 pin is used as the serial clock. In order to ensure the integrity of shift operations, both the high and low-level serial clock pulse widths shown below must be maintained. The maximum data transfer frequency is 1.25 MHz (when fc = 20 MHz).

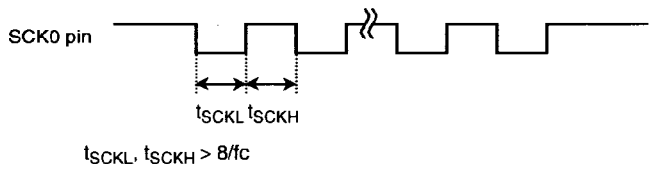


Figure 3.10.30 Maximum Data Transfer Frequency when External Clock Input

② Shift edge

Data is transmitted on the leading edge of the clock and received on the trailing edge.

Leading edge shift

Data is shifted on the leading edge of the serial clock (on the falling edge of the SCK0 pin input/output).

Trailing edge shift

Data is shifted on the trailing edge of the serial clock (on the rising edge of the SCK0 pin input/output).

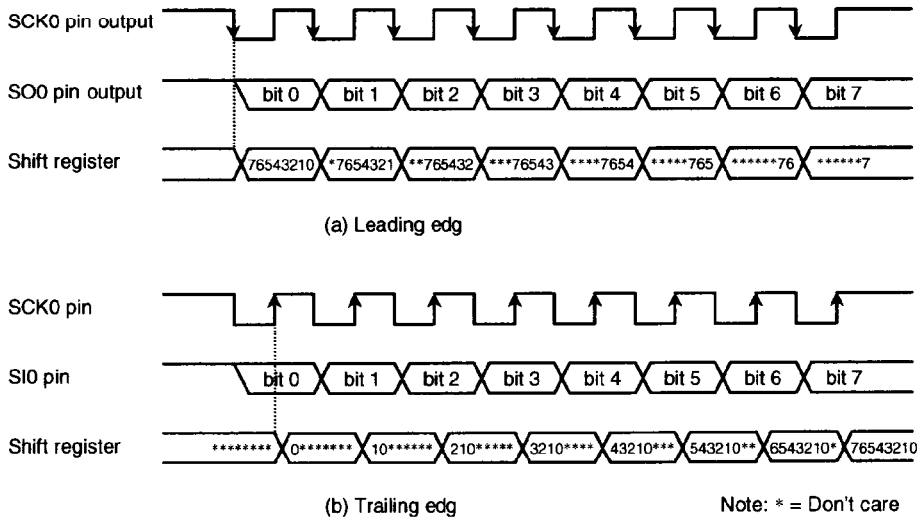


Figure 3.10.31 Shift Edge

(2) Transfer Modes

The SBI0CR1<SIOM1 to 0> is used to select a transmit, receive or transmit/receive mode.

① 8-bit transmit mode

Set a control register to a transmit mode and write transmission data tothe SBI0DBR. After the transmit data has been written, setthe SBI0CR1<SIOS> to "1" to start data transfer. The transmitted data is transferred from the SBI0DBR to the shift register and output, starting with the least significant bit (LSB), via the SO0 pin and synchronized with the serial clock. When the transmission data has been transferred to the shift register, the SBI0DBR becomes empty. The INTSBE0 (buffer empty) interrupt request is generated to request new data.

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

When the external clock is used, data should be written tothe SBI0DBR before new data is shifted. The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written tothe SBI0DBR by the interrupt service program.

When the transmit is started, after the SBI 0SR<SIOF> goes "1" output from the SO0 pin holds final bit of the last data until falling edge of the SCK0.

Data transmission ends when the <SIOS> is cleared to "0" by the INTSBE0 interrupt service program or whenthe <SIOINH> is set to "1". When the <SIOS> is cleared to "0", the transmitted mode ends when all data is output. In order to confirm whether data is being transmitted properly by the program, the <SIOF> (bit 3 of the SBI0SR) to be sensed. The SBI0SR<SIOF> is cleared to "0" when transmission has been completed. When the <SIOINH> is set to "1", transmitting datat stops. The <SIOF> turns "0".

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

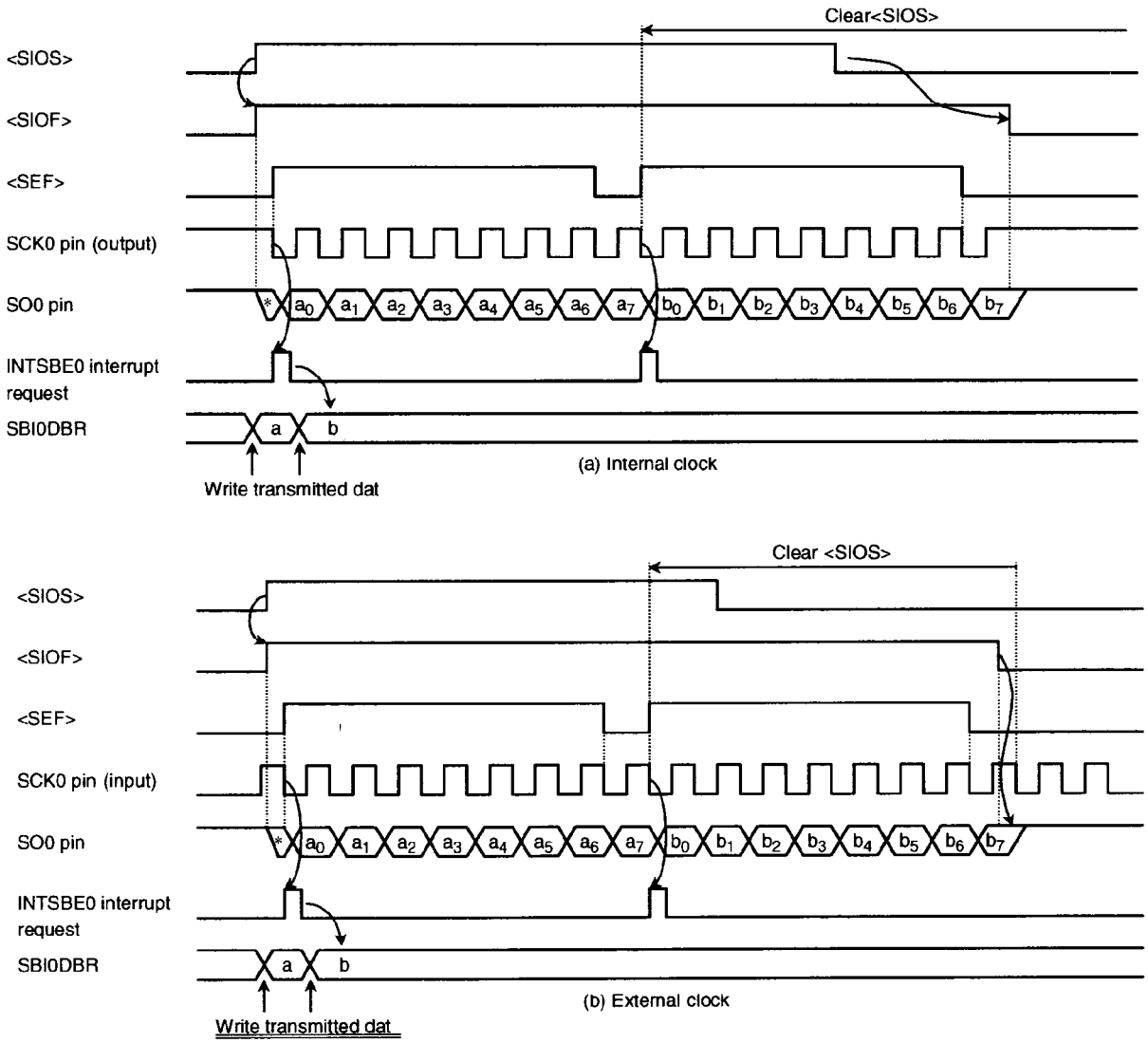


Figure 3.10.32 Transfer Mode

Example: Program to stop data transmission (when an external clock is used)

```
STEST1: BIT    SEF. (SBI0SR)          : If <SEF> = 1 then loop
        JR     NZ. STEST1
STEST2: BIT    0. (PN)                : If SCK0 = 0 then loop
        JR     Z. STEST2
        LD     (SBI0CR1). 00000111B   : <SIOS> ← 0
```

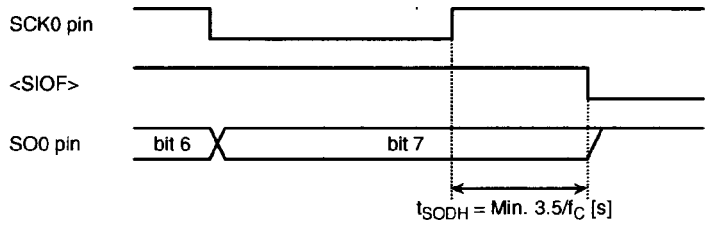


Figure 3.10.33 Transmitted Data Hold Time at End of Transmission

② 8-bit receive mode

Set the control register to receive mode and setthe SBI0CR1<SIOS> to "1" for switching to receive mode. Data is received into the shift register via the SI0 pin and synchronized with the serial clock, starting from the least significant bit (LSB). When the 8-bit data is received, the data is transferred from the shift register tothe SBI0DBR. The INTSBE0 (buffer full) interrupt request is generated to request that the received data be read. The data is then read from the SBI0DBR by the interrupt service program.

When the internal clock is used, the serial clock will stop and the automatic wait function will be in effect until the received data is read fromthe SBI0DBR.

When the external clock is used, since shift operation is synchronized with an external clock pulse, the received data should be read fromthe SBI0DBR before the next serial clock pulse is input. If the received data is not read, further data to be received is canceled. The maximum transfer speed when an external clock is used is determined by the delay between the time when an interrupt request is generated and the time when the received data is read.

Receiving of data ends when the <SIOS> is cleared to"0" by the INTSBE0 interrupt service program or when the <SIOINH> is set to "1". If <SIOS> is cleared to "0", received data is transferred to the SBI0DBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm whether data is being received properly by the program,the SBI0SR<SIOF> to be sensed. The <SIOF> is cleared to "0" when receiving is complete. When it is confirmed that receiving has been completed, the last data is read. Whenthe <SIOINH> is set to "1", data receiving stops. The <SIOF> is cleared to "0" (the received data becomes invalid, therefore no need to read it).

Note: When the transfer mode is changed, the contents ofthe SBI0DBR will be lost. If the mode must be changed, conclude data receiving by clearing the <SIOS> to "0", read the last data, then change the mode.

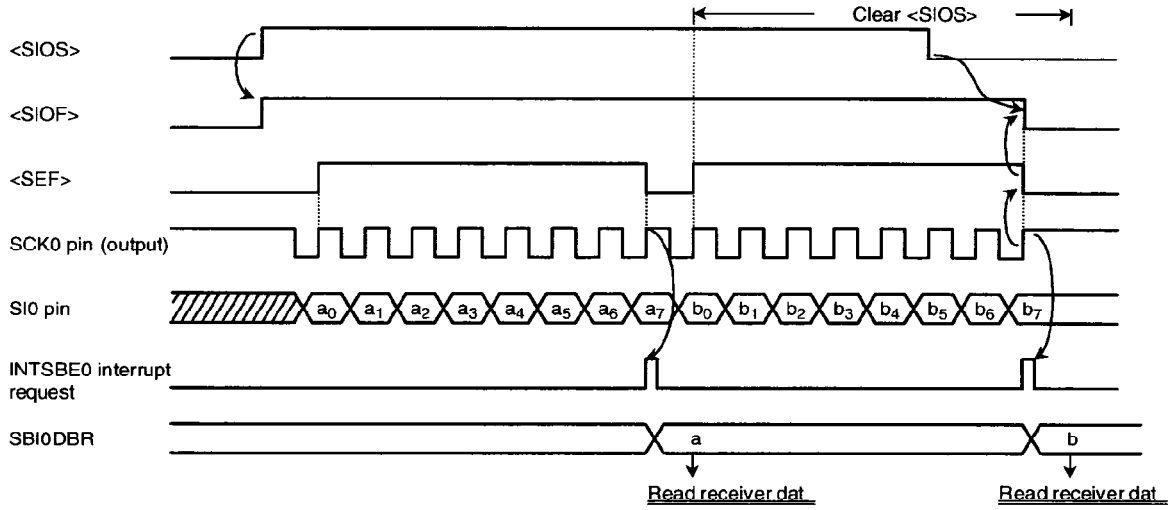


Figure 3.10.34 Receiver Mode (example: Internal clock)

③ 8-bit transmit/receive mode

Set a control register to a transmit/receive mode and write data tothe SBI0DBR. After the data is written, set the SBI0CR<SIOS> to "1" to start transmitting/receiving. When data is transmitted, the data is output from the SO0 pin, starting from the least significant bit (LSB) and synchronized with the leading edge of the serial clock signal. When data is received, the data is input via the SI0 pin on the trailing edge of the serial clock signal. 8-bit data is transferred from the shift register tothe SBI0DBR and the INTSBE0 interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the data which is to be transmitted. The SBI0DBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

When the internal clock is used, the automatic wait function will be in effect until the received data is read and the next data is written.

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

When the transmit is started, after the SBI 0SR<SIOF> goes "1" output from the SO0 pin holds final bit of the last data until falling edge of the SCK0.

Transmitting/receiving data ends when the <SIOS> is cleared to "0" by the INTSBE0 interrupt service program or whenthe SBI0CR1<SIOINH> is set to "1". When the <SIOS> is cleared to "0", received data is transferred to the SBI0DBR in complete blocks. The transmit/receive mode ends when the transfer is complete. In order to confirm whether data is being transmitted/received properly by the program, set the SBI0SR to be sensed. The <SIOF> is set to "0" when transmitting/receiving is completed. Whenthe <SIOINH> is set to "1", data transmitting/receiving stops. The <SIOF> is then cleared to "0".

Note: When the transfer mode is changed, the contents of the SBI0DBR will be lost. If the mode must be changed, conclude data transmitting/receiving by clearingthe <SIOS> to "0", read the last data, then change the transfer mode.

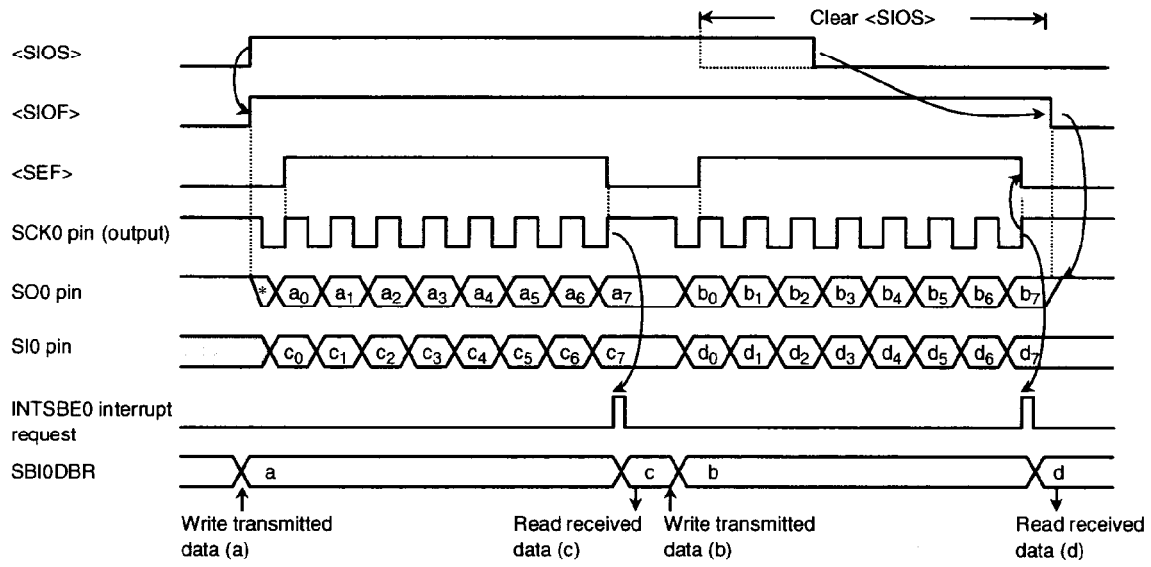


Figure 3.10.35 Transmit/Received Mode (Example : Internal clock)

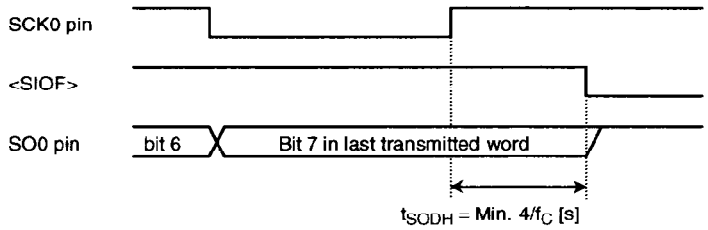


Figure 3.10.36 Transmitted Data Hold Time at End of Transmit/Receive

3.11 Serial Expansion Interface (SEI)

3.11.1 Overview

The SEI is one of the serial interfaces built in the TMP92CW10, which can be connected to peripheral devices, by full duplex synchronous communication protocol. The TMP92CW10 incorporates 2 channels of this SEI (SEI0 and SEI1). Also the SEI can support the micro DMA mode correspond to the micro DMA transfer.

The SEI0 and SEI1 have identical function.

(1) Features

- The shift clock is output by the master for only a data transfer period
- The clock polarity and phase are programmabl
- The data is 8 bits long
- The MSB first or LSB first can be selected
- Micro DMA mode support for micro DMA transfers
- Transfer rate : 8Mbps, 4Mbps, 2Mbps or 500kbps (when operating at fc = 20MHz)
- Error detection function

① Write collision detection : when write to the shift register during the data transfer

② Overflow detection : when receive the new data with the transfer end flag is set (only slave)

③ Mode fault detection : when two or more SEI devices are set as the master simultaneously (only master)

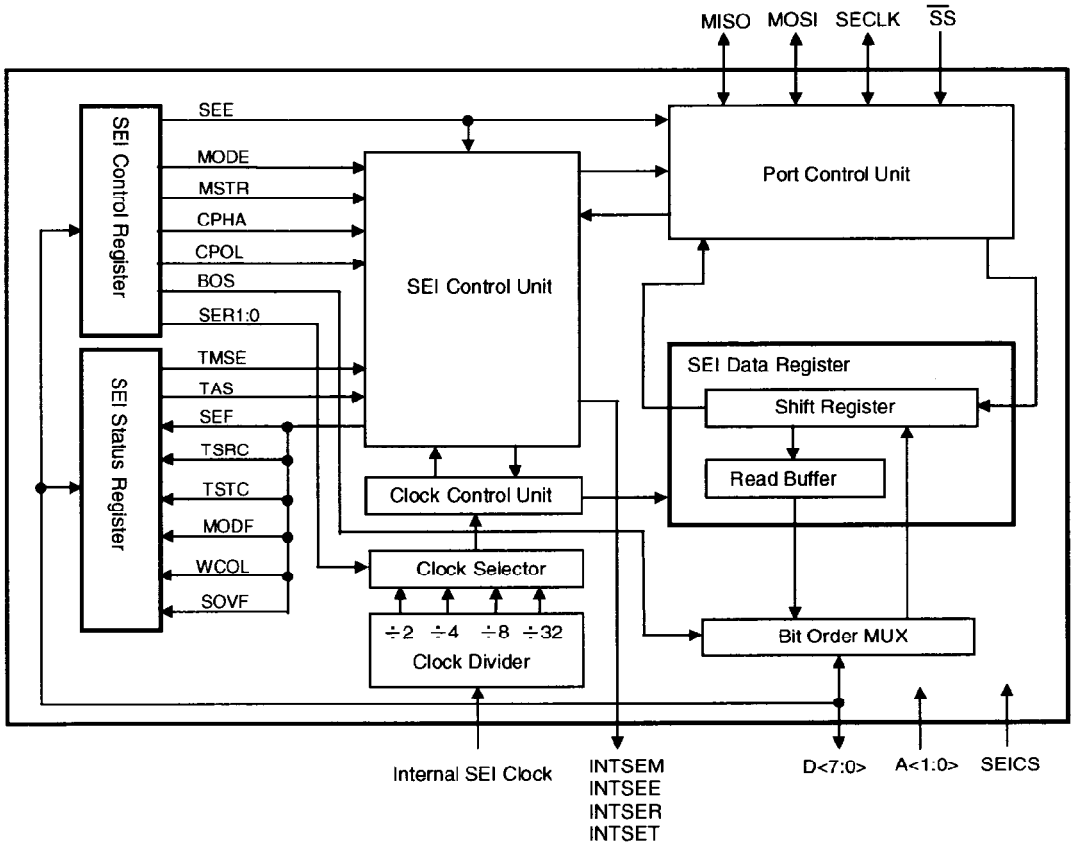


Figure 3.11.1(1) SEI Block Diagram

Table 3.11.1( ) Pin Function of SEI Channels

SEI0	SEI1
$\overline{SS0}$ (PM0)	$\overline{SS1}$ (PM4)
MOSI0 (PM1	MOSI1 (PM5
MISO0 (PM2	MISO1 (PM6
SECLK0 (PM3)	SECLK1 (PM7)

3.11.2 SEI operation

During a SEI transfer, data is simultaneously transmitted (shifted out serially) and received serially (shifted in serially). A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows individual selection of a slave SEI device; slave devices that are not selected do not interfere with SEI bus activities. On master SEI device, the slave select line can optionally be used to indicate a multiple-master bus connection.

(1) SEI clock phase and polarity controls

Software can select any four combinations of serial clock phase and polarity using two bits in the SEI contro register (SECR). The clock polarity is specified by the <CPOL> control bit, which selects an active high o active low clock and has no significant effects on the transfer format. The clock phase <CPHA> control bit selects one of two fundamentally different transfer formats. The clock phase and polarity should be identical for the master SEI device and the communicating slave device. In some cases, the phase and polarity ar changed between transfers to allow a master device to communicate with peripheral slaves having different requirements.

(2) SEI data and clock timing

The adjustable data clock timings of the SEI module can connect almost any synchronous serial peripheral device. Please see "3.11.4 SEI transfer format" for a detailed description of the transfer format.

3.11.3 SEI signal lines

There are four input/output pin signals associated with the SEI transfer. Every signal depends on the mod (master/slave) of the SEI device.

(1) SECLK

The SECLK pin functions as an output pin when the SEI is set for master and functions as an input pin when the SEI is set for slave.

When the SEI is set for master, th SECLK signal is supplied by the internal SEI clock generation circuit. When the master starts transferring data, eight clock cycles are automatically generated at the SECLK pin.

When the SEI is set for slave, the SECLK pin functions as an input pin, in which case the SECLK signal from the master synchronizes data transfers between the master and slave. The slave device ignores the SECL signal unless the slave select $\overline{SS}$ pin is low.

In both master and slave SEI devices, data is shifted in or out at each rising or falling edge of the SECLK signal and is sampled at the opposite edge where the data is stable. Edge polarity is determined by the SEI transfer protocol.

(2) MISO/MOSI

The MISO and MOSI pins are used for transmitting and receiving serial data.

When the SEI is configured as a master, MISO is the data input line and MOSI is the data output line.

When the SEI is configured as a slave, these pins reverse roles.

In a multiple-master system, all SECLK pins are tied together, all MOSI pins are tied together and all MISO pins are tied together. A single SEI device is configured as a master, all other SEI devices on the SEI bus are configured as slaves. The single master drives data out it's SECLK and MOSI pins to the SECLK and MOSI pins of the slaves. One selected slave device optionally drives data out it's MISO pin to the MISO master pin.

The SECLK,MISO and MOSI pins can be set to function as open-drain pins. Th port M open drain enabl register PMODE are used for this setting

(3) $\overline{SS}$

The $\overline{SS}$ pin behaves differently on master and slave devices.

On a slave device, this pin is used to enable the SEI slave for a transfer. If the $\overline{SS}$ pin of a slave is inactiv (high), the device ignores SECLK clocks and keeps the MISO output pin in the high-impedance state.

On a master device, the $\overline{SS}$ pin serves as an error-detection input for the SEI. If the $\overline{S}$ pins goes low while th SEI is a master, it indicates that some other device on the SEI bus is attempting to be master. This attempt causes the master device sensing the error to immediately exit the SEI bus to avoid potentially damaging driver contentions. This detection is called mode fault.

For the TMP94FD53, the SECR0/1 register's bit <MODE> is used to enable or disable mode fault detection. When the <MODE> bit = 0, the $\overline{SS}$ pin is enabled for mode fault detection input. When the <MODE> bit = 1, the $\overline{SS}$ pin is disabled from mode fault detection input.

3.11.4 SEI transfer format

The transfer format depends on the setting of the <CPHA> bit and <CPOL> bit in the SECR register. <CPHA> bit switches between two fundamentally different transfer protocols.

(1) Transfer Format of <CPHA>=0

Figure 3.11.4(1) shows the transfer format for a <CPHA>=0 transfer.

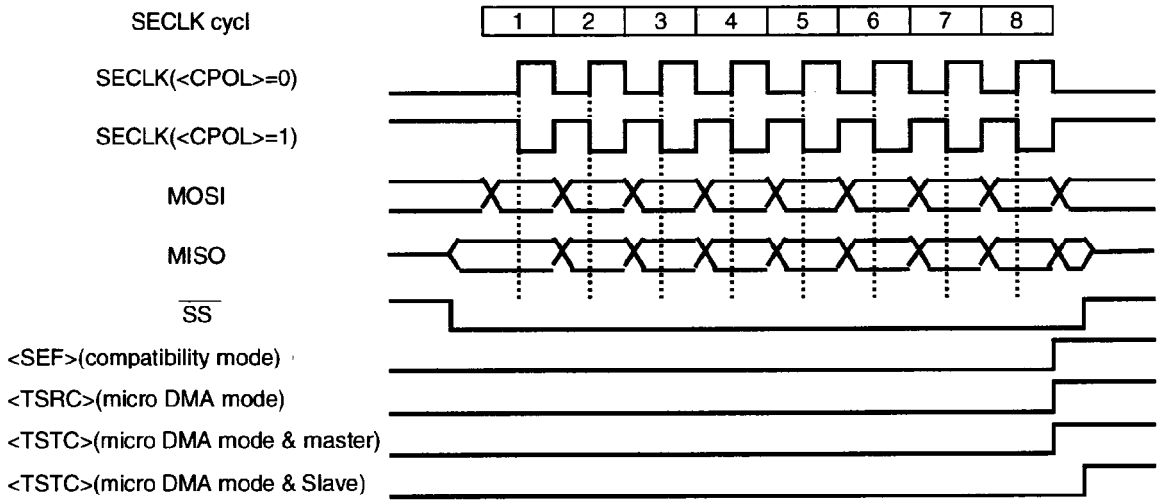


Figure 3.11.4(1) Transfer Format of <CPHA>=0

In this transfer format, the first bit is sampled in on the first clock edge. This will be on a rising edge when <CPOL>=0 and on a falling edge when <CPOL>=1. With <CPOL>=0 the shift clock will idle low, wit <CPOL>=1 it will idle high.

In master mode, when a transfer is initiated by writing new data to the SEDR register the new data is placed on the MOSI pin for half a clock cycle before the shift clock starts to operate. The <BOS> bit in the SECR registe determines whether the data will be shifted out in a MSB or LSB order. After the last shift cycle, the <SEF> flag (in the compatibility mode) or the <TSRC> flag and <TSTC> flags (in the micro DMA mode) will be asserted.

In slave mode the SEDR register is not allowed to be written, if the $\overline{SS}$ signal is low. A write attempt in this stat will result in a write collision and the <WCOL> bit will be asserted in the SESR register. Therefore even if th transfer has been completed and the <SEF> flag or <TSRC> flag bit has been asserted, software has to wait until the $\overline{SS}$ signal goes high again before writing a new data to SEDR register. To allow the usage of a micr DMA to transfer data to the SEDR register in the slave mode the <TSTC> flag is delayed until $\overline{SS}$ goes high.

(2) Transfer format of <CPHA>=1

Figure 3.11.4(2) shows the transfer format for a <CPHA>=1 transfer.

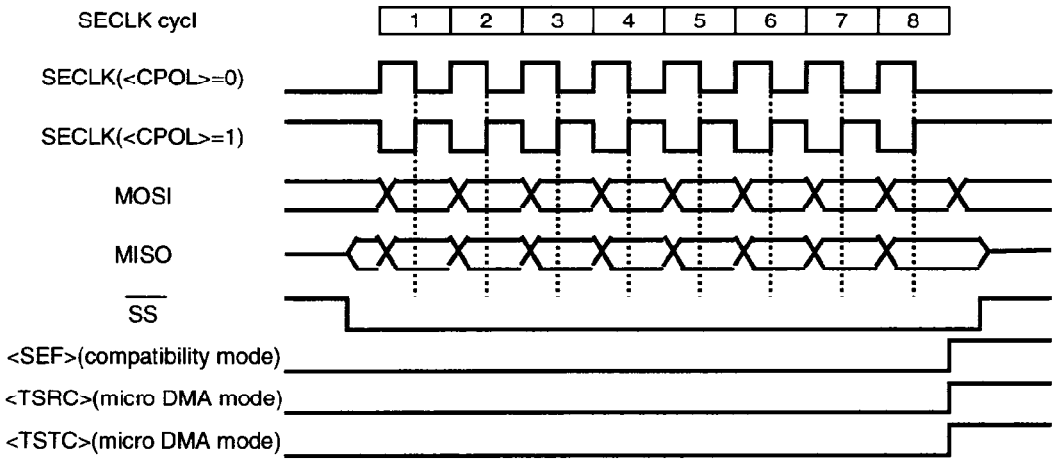


Figure 3.11.4(2) Transfer Format of <CPHA>=1

In this transfer format, the first bit is sampled in on the second clock edge. This will be on a falling edge when <CPOL>=0 and on a rising edge when <CPOL>=1. If <CPOL>=0, the shift clock is a rising edge, wit <CPOL>=1 it is a falling edge.

In master mode, when a transfer is initiated by writing new data to the SEDR register, the data is placed on th MOSI pin with the first edge of the shift clock. Again, the first bit to be transferred will be determined by th <BOS> bit in the SECR register.

Unlike in the <CPHA>=0 format, SEDR register is allowed to be written in slave mode even if the $\overline{SS}$ signal is low.

In both, master and slave mode, the <SEF> flag (in Compatibility mode) or the <TSRC> flag and <TSTC> flag (in micro DMA mode) will be asserted simultaneously after the completion of the last shift cycle. An attempt t write the SEDR register while the data shifting is still in progress will result in a write collision.

3.11.5 Functional description

Figure 3.11.5(1) shows master-to-slave connection via the SEI.

The different nodes on a SEI bus function like a distributed shift register. When data is sent from the MOSI pin of the master device to the corresponding pin of the slave device, data from the slave is sent back from th MISO pin of the slave device to the corresponding pin of the master device.

This means that data is communicated in full-duplex mode and data output and data input are synchronized by the same clock signal. After a transfer, the transmitted bytes have been replaced with the received bytes.

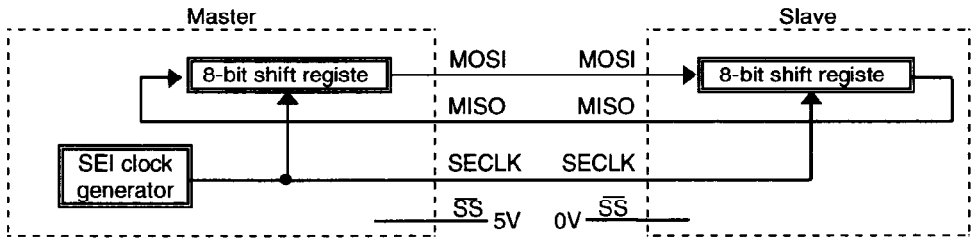


Figure 3.11.5(1) Connection between Master and Slave in SEI

Figure 3.11.5(2) shows a configuration of the SEI system.

Port M, the SEI output, can be set for open-drain output. Therefore, this port can be connected to multiple devices.

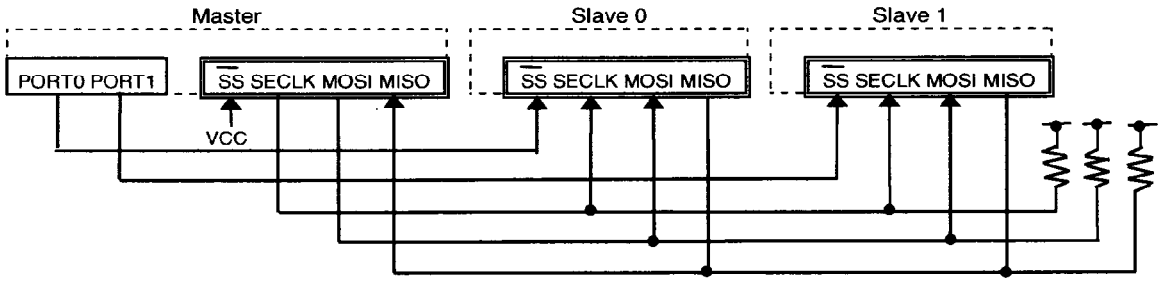


Figure 3.11.5(2) Configuration of SEI System (Comprised of One Master and Two Slaves)

3.11.6 Operation Modes

SEI allows the programmer the choice between 2 different operation modes, the compatibility mode and th micro DMA mode. Those operation modes differ in terms of flag clearing , interrupt generation and transfer initiation. The table below shows the differences between the two operation modes.

Table 3.11.6( ) Differences between the Two Operation Mode

	compatibility mode	micro DMA mod
error flag clearing	Reading a register with the Status flag set, followed by an another register access	Writing a "1" to the status register
transfer status flag clearing	Reading a register with the Status flag set, followed by an access to the data register	Writing a "1" to the status register or by reading or writing the data register
interrupt generation	INTSEM: <MODF> INTSEE: <SEF>	INTSEM: <MODF> INTSEE: <WCOL or <SOVF> INTSER: <TSRC> INTSET: <TSTC>
micro DMA usag	no	yes

SEI can be switched between these operation modes, if SEI is disabled (<SEE> = 0) by setting th <TMSE> bit in the SESR register.

3.11.7 SEI registers

The SEI control register SECR, SEI status register SESR and SEI data register SEDR are used to configur and operate the SEI system

Note: When accessing the SEI registers, at least 4 states must be inserted between SEI register write and SEI register read. Please remember this when programming.

Example :

```
LD   (SEDR), data1      : write SEDR
NOP                     :} or another instructions which does not access the SEI registers
NOP          :          }
LD   A ,(SESR)          : read SESR
LD   (SESR), data2      : write SESR
NOP                     :} or another instructions which does not access the SEI registers
NOP          :          }
LD   A ,(SESR)          : read SESR
```

(1) SEI control register (SECR0, SECR1)

SEI0 Control Register

		7	6	5	4	3	2	1	0
SECR0 (0060H)	bit Symbol	MODE	SEE	BOS	MSTR	CPOL	CPHA	SER1	SER0
	Read/Writ	W	R/W						
	After reset	0	0	0	0	0	1	0	0
Read modify-writ instructions prohibited.	Function	Mod fault detection 0:enabled 1:disabled	SEI operation 0:stopped 1: operating	Bit orde selection 0:MSB firs 1:LSB firs	Mod selection 0:slave 1:master	Clock polarity selection see figur 3.11.4 (1),(2)	Clock Phase selection see figur 3.11.4 (1),(2)	SEI transfer rate selection 00: divide-by- 2 01: divide-by- 4 10: divide-by- 8 11: divide-by-32	

SEI1 Control Register

		7	6	5	4	3	2	1	0
SECR1 (0064H)	bit Symbol	MODE	SEE	BOS	MSTR	CPOL	CPHA	SER1	SER0
	Read/Writ	W	R/W						
	After reset	0	0	0	0	0	1	0	0
Read modify-writ instructions prohibited.	Function	Mod fault Detection 0:enabled 1:disabled	SEI operation 0:stopped 1: operating	Bit orde selection 0:MSB firs 1:LSB firs	Mod selection 0:slave 1:master	Clock polarity selection see figur 3.11.4 (1),(2)	Clock Phase selection see figur 3.11.4 (1),(2)	SEI transfer rate selection 00: divide-by- 2 01: divide-by- 4 10: divide-by- 8 11: divide-by-32	

<MODE>: Mode fault detection enabl

0: Mode fault detection enabled.

1: Mode fault detection disabled.

<SEE>: SEI system enabl

0:SEI system is off. It is necessary to disable the SEI system to switch between the micro DMA mode and the compatibility mode. Wait until the transfer in progress is completed before you clear the <SEE> bit to stop the SEI operation. Please carry out after disabling the SEI system, when going into the STO or IDLE1 mode by HALT instruction.

1:SEI system is on. Before using the SEI system, make sure that the port M control register PMCR an function register PMFC are set for the SEI function.

<BOS>: Bit order select

The bit order selection bit <BOS> selects whether the data to be transferred is MSB first or LSB first.

0:The MSB bit of the SEDR register (bit 7) will be transmitted first.

1:The LSB bit of the SEDR register (bit 0) will be transmitted first.

<MSTR>: Master/Slave mode select

0:SEI is configured as slave.

1:SEI is configured as master.

The function of the <TASM> bit in the SESR register depends on the setting of this bit.

<CPOL>: Clock polarity select

0:Active high clock selected. SECLK idles low.

1:Active low clock selected. SECLK idles high.

<CPHA>: Clock phase select

<CPHA> bit selects one of two, fundamentally different transfer format.

<SER1:0>: SEI bit rate select

The following table shows the relationship between the <SER1> and <SER0> control bits and the bit rat for transfers when the TSEI is operating as a master. When the TSEI is operating as a slave, the serial clock is input from the master, therefore, the <SER1> and <SER0> control bits have no meaning.

<SER1>	<SER0>	Divide-by-rate of internal SEI clock	Transfer rat (@ fc = 20 MHz)
0	0	2	8 Mbps
0	1	4	4 Mbps
1	0	8	2 Mbps
1	1	32	500 Kbps

(2) SEI status register (SESR0, SESR1)

SEI0 Status Register

		7	6	5	4	3	2	1	0
SESR0 (0061H)	bit Symbol	SEF	WCOL	SOVF	MODF				TMSE
	Read/Writ	R							R/W
	After reset	0	0	0	0				0
compati-bility mode	Function	SEI transfer complete flag 1:transfer completed	Write collisio flag 1:write collided	Overflow flag (slave) 1:overflow occurred	Mod fault fla (master) 1:fault occurred				SEI mod select 0:compati-bility mode 1:micro DMA mod

SEI0 Status Register

		7	6	5	4	3	2	1	0
SESR0 (0061H)	bit Symbol		WCOL	SOVF	MODF	TSRC	TSTC	TASM	TMSE
	Read/Writ		R					R/W	
	After reset		0	0	0	0	0	0	0
micro DMA mod Read modify-writ instructions prohibited.	Function		Write collisio flag 1:write collided	Overflow flag (slave) 1:overflow occurred	Mod fault fla (master) 1:fault occurred	SEI receive complete flag 1:receive completed	SEI transmit complete flag 1:transmit completed	SEI automated shift mode (master) interrupt mask (slave)	SEI mod select 0:compati-bility mode 1:micro DMA mod

SEI1 Status Register

		7	6	5	4	3	2	1	0
SESR1 (0065H)	bit Symbol	SEF	WCOL	SOVF	MODF				TMSE
	Read/Writ	R							R/W
	After reset	0	0	0	0				0
compati-bility mode	Function	SEI transfer complete flag 1:transfer completed	Write collisio flag 1:write collided	Overflow flag (slave) 1:overflow occurred	Mod fault fla (master) 1:fault occurred				SEI mod select 0:compati-bility mode 1:micro DMA mod

SEI1 Status Register

		7	6	5	4	3	2	1	0
SESR1 (0065H)	bit Symbol		WCOL	SOVF	MODF	TSRC	TSTC	TASM	TMSE
	Read/Writ		R					R/W	
	After reset		0	0	0	0	0	0	0
micro DMA mod Read modify-writ instructions prohibited.	Function		Write collisio flag 1:write collided	Overflow flag (slave) 1:overflow occurred	Mod fault fla (master) 1:fault occurred	SEI receive complete flag 1:receive completed	SEI transmit complete flag 1:transmit completed	SEI automated shift mode (master) interrupt mask (slave)	SEI mod select 0:compati-bility mode 1:micro DMA mod

<SEF>: Transfer complete flag

Compatibility mode:

The <SEF> flag is automatically set to one at the end of a SEI transfer. The <SEF> flag is automatically cleared by reading the SESR register with <SEF> flag set, followed by an access of the SEDR register.

Micro DMA mode:

Always reads as undefined, writes to this flag have no effect.

<WCOL>: Write collision error flag

Compatibility mode:

The <WCOL> flag is automatically asserted, if the SEDR register is written while a transfer is in progress. The write itself has no effect on the running transmission. The <WCOL> flag is automatically cleared by reading the SESR register with <WCOL> bit set followed by an access to the SEDR register. No interrupt will be generated on the assertion of this flag.

Micro DMA mode:

The <WCOL> flag is automatically asserted, if the SEDR register is written while a transfer is in progress. The write itself has no effect on the running transmission. The flag can only be reset by writing a "1" to it. Writing a "0" has no effect. An interrupt will be generated on INTSEE on a transition from "0" to "1", if th module is configured as a slave and the <TASM> bit is equal to "0".

<SOVF>: Slave mode overflow error flag

Master mode:

Always reads as undefined, writes to this flag have no effect.

Slave mode:

Compatibility mode:

The <SOVF> flag is automatically asserted, if a new byte has been completely received and the <SEF> flag is still asserted. The <SOVF> flag is automatically cleared by reading the SESR register with the <SOVF> flag is set followed by an access to the SEDR register. The <SOVF> flag will also be cleared by switch to the master mode. In compatibility mode, no interrupt will be generated on the assertion of <SOVF> flag

Micro DMA mode:

The <SOVF> flag is automatically asserted, if a new byte has been completely received and the <TSRC> flag is still asserted. The <SOVF> flag can only be cleared by writing a "1" to it. Writing a "0" to it has no effect. INTSEE is generated with <TASM> =1, if <SOVF> flag from 0 to 1.

<MODF>: Mode-fault error flag

Master mode:

Compatibility mode:

The <MODF> flag is set, if the $\overline{SS}$ signal goes to active low while the SEI is configured as a master. In this case:

1. The SEI output pin drivers are disabled and the output pins are placed in high-impedance state.
2. The <MSTR> bit in the SECR register is cleared.
3. The <SEE> bit is forcibly cleared to disable the SEI system.
4. An interrupt INTSEM is generated.

The <MODF> flag is automatically cleared by reading the SESR register with the <MODF> bit set, followed by a write to SECR register.

Micro DMA mode:

It is the same as that of the compatibility mode, except the <MODF> flag's clearance.

This flag can only be cleared by writing "1" to it. Writing a "0" to this flag has no effect.

Slave mode:

Always reads as undefined, writes to this flag have no effect.

<TSRC>: Receive completion flag

Compatibility mode:

Always reads as undefined, writes to this flag have no effect.

Micro DMA mode:

The <TSRC> flag is set when a transfer has been completed, that is when eight cycles where shifted on the SECLK signal. It is cleared by performing a read operation on the SEI data register, by switching t compatibility mode or by writing a "1" to this flag. Writing a "0" to this flag has no effect. An interrupt INTSER will be generated on the assertion of this flag.

<TSTC>: Transmit completion flag

Compatibility mode:

Always reads as undefined, writes to this flag have no effect.

Micro DMA mode:

This <TSTC> flag is set when one byte of data has been completely shifted out and when new data is allowed to be written to the SEDR register. It is cleared by performing a write operation on the SEI dat register, by switching to compatibility mode or by writing a "1" to this flag. Writing a "0" to this flag has no effect. An interrupt INTSET will be generated on the assertion on this flag.

<TASM>: Automated shift mode(master)／INTSEE interrupt mask(slave)

Compatibility mode:

Always reads as undefined, writes to this flag have no effect.

Micro DMA mode:

Master mode:

0: Disables the automated shift mode.

1: Enables the automated shift mode.

In this mode a read access to the SEI data register SEDR will perform the following functions.

- The SEI data register will be cleared to 00 hex.
- A new transfer will be initiated, thus in master mode 8 low bits will be shifted out, 8 new bits will b shifted in.

The automated shift mode also works when it is combined with a micro DMA. It has no effect, when SEI is in the slave mode.

Slave mode:

This bit functions as a mask for the interrupt INTSEE generation of the <SOVF> and <WCOL> flags.

0: An interrupt INTSEE will be generated when the <WCOL> flag is set t "1", but not effect on th <SOVF> flag

1: An interrupt INTSEE will be generated when the <SOVF> flag is set t "1", but not effect on the <WCOL> flag.

<TMSE>: SEI mode select

0: Compatibility mode.

1: Micro DMA mode.

Selects the micro DMA mode, which also allows micro DMA transfers. It is necessary to disable the SEI system before switching to the micro DMA mode.

(3) SEI data register (SEDR0, SEDR1)

SEI0 Data Register

		7	6	5	4	3	2	1	0
SEDR0 (0062H)	bit Symbol	SED7	SED6	SED5	SED4	SED3	SED2	SED1	SED0
	Read/Writ	R/W							
	After reset	0	0	0	0	0	0	0	0

SEI1 Data Register

		7	6	5	4	3	2	1	0
SEDR1 (0066H)	bit Symbol	SED7	SED6	SED5	SED4	SED3	SED2	SED1	SED0
	Read/Writ	R/W							
	After reset	0	0	0	0	0	0	0	0

This register is used to transmit and receive data. When the SEI system configured as a master, transfers ar started by a software write to the SEDR register.

After once starting transmission, please write after checking that the transmission end flag has surely set by interrupt or polling when master device writes to SEDR register.

Only when the <SEE> bit of the SECR register is "1", a read/write to the SEDR register is possible. When th <SEE> bit is "0", the write access is ignored and "00 " will be read whenever it read.

3.11.8 SEI system errors

Three system errors can be detected by the SEI system. The first type error arises in a multiple-master system when more than one SEI device simultaneously tries to be master. This error is called a mode fault. Th second type error, a write collision, indicates that an attempt has been made to write data to the SEDR while a transfer was in progress. The third error occurs when the SEI system is configured as a slave and a new byte of data has been completely shifted in by the remote bus master before the old byte could be read.

(1) Mode-fault error

When SEI system is configured as master and the $\overline{SS}$ input line goes to active low, a mode-fault error has occurred. Only a SEI master can experience a mode-fault error, caused when a second SEI device becomes a master and selects this device as if it were a slave. In cases where more than one device is concurrently configured as master, there is a change of contention between two pin drivers. For push-pull CMOS drivers, this contention can cause catastrophic latch-up. When this type of error is detected, the following actions ar taken immediately.

- The SECR register's <MSTR> bit is forcibly cleared to set the SEI for slave.
- The SECR register's <SEE> bit is forcibly cleared to disable the SEI system.
- The SESR register's <MODF> flag is set, and INTSEM interrupt pulse is generated.
- The SEI output pin drivers are disabled and the output pins are placed in the high-impedance state.

When the problem that has caused the mode fault is solved in software, the <MODF> flag is cleared and the SEI system can be setup to return to normal operation. In the compatibility mode the <MODF> flag is automatically cleared by reading SESR register while <MODF> flag is set, followed by write to the SECR register. In micro DMA mode the <MODF> flag is cleared by writin "1" value to it.

There may be a case where the mode fault mechanism cannot surely protect a multi-master system against driver collisions. For example, such a case may occur when a second SEI device becomes the master, but th $\overline{SS}$ pin of this device is not driven high immediately after that. Two slave devices are selected and thes devices attempt to drive the MISO pin simultaneously. In this case, both devices result in a collision of output drivers, but mode fault is detected in neither device

The method to protect the drivers from latch-up is to use an open drain option. This is to change the SEI output driver to the driver to the of open drain type. The SECLK pin, MOSI pin and MISO pin can be set open drai respectively by the port M open drain enable register PMODE. In this case, outside additional pull-up resistor is necessary.

(2) Write collision error

A write collision occurs if the SEDR register is written while a transfer is in progress. Since the SEDR register is not double buffered in the transmit direction, writes to SEDR register cause data to be written directly into th SEI shift register. Because this write corrupts any transfer in progress, a write-collision error is generated. Th transfer continues undisturbed, and the write data that caused the error is not written to the shifter.

A write collision is normally a slave error because a slave has no control over when a master will initiate a transfer. A master knows when a transfer is in progress, thus, there is no excuse for a master to generate a write collision error although the SEI logic can detect write collision in a master as well as in a slave.

In slave mode a write collision is likely to occur, when the master shifts faster, than it can be handled by th slave. This will be, when the slave is transferring a new value to the data register when the master has already begun with the next shift cycle. In this case a write collision occurs.
In micro DMA mode an interrupt on INTSEE will occur, if the module is configured as slave, <TASM> bit is chosen to be equal to zero and <WCOL> flag shows a positive edge.

(3) Slave mode overflow error

On a SEI bus the bit rate of transmission is determined by the master. At higher bit rates the problem arises that a slave might not be able to follow the transmission of a master. This means, that the data is shifted i faster than it can be processed on the slave side. Therefore the SEI module offers the <SOVF> flag in its status register which allows the detection of a possible loss of data.
<SOVF> flag will be asserted, when,

- The SEI module is configured as a slave.
- An old byte of data is still waiting to be read when a new byte of data has been completely received.

When <SOVF> is set, SEDR has been overwritten by new byte data.
Since this error is only a subject in the slave mode, the <TASM> bit can be used as an interrupt mask for this flag. An interrupt is generated on INTSE0 only in the micro DMA mode and the <TASM> bit is "1", if the <SOVF> flag in the status register is asserted.

3.11.9 Interrupt generation

Interrupt processing differs for the two SEI operating modes, which can be selected using the <TMSE> bit in the SESR register. It generates four interrupts par one SEI module that are INTSEM, INTSEE, INTSER an INTSET.

(1) Compatibility mode

In compatibility mode the INTSEM and INTSEE are used. The SEI module generates the INTSEM interrupt, if the <MODF> flag in the SESR register shows a transition from "0" to "1". And it generates the INTSEE interrupt, if the <SEF> flag shows a transition from "0" to "1".

INTSEM	Interrupt on <MODF>
INTSEE	Interrupt on <SEF>
INTSER	Inactiv
INTSET	Inactiv

(2) Micro DMA mod

In micro DMA mode all four interrupts are used to allow the micro DMA transfers to and from the SEI data register. The INTSEM is generated on a transition of the <MODF> flag from "0" to "1". The INTSEE is generated if the module is in slave mode on a transition of the <WCOL> flag from "0" to "1" with <TASM> bit is "0" or on a transition of the <SOVF> flag from "0" to "1" with <TASM> bit is "1".

After a completed transfer both the <TSRC> flag and the <TSTC> flag in the SESR register are asserted simultaneously. However, there is an exception for <CPHA> equals "0" in slave mode. Please s "3.11.4(1) transfer format of <CPHA>= ". Both flags trigger the INTSER and INTSET interrupts.

The <TSRC> flag generates an interrupt INTSER on a transition from "0" to "1". The <TSRC> flag can be cleared by either reading the SEDR register or by writin "1" value to this flag

The <TSTC> flag generates an interrupt INTSET on a transition from "0" to "1". The <TSTC> flag is cleared by either writing the SEDR register or by writing a "1" value to this flag.

In order to use the micro DMA, the INTSER can be used to trigger a micro DMA that reads the data from th SEDR register and the INTSET can be used to trigger a micro DMA that writes a new value to the SEDR register. Thus initiating a new transfer.

INTSEM	Interrupt on <MODF>
INTSEE	Interrupt on <WCOL>[1] or <SOVF>[2]
INTSER	Interrupt on <TSRC>
INTSET	Interrupt on <TSTC>

Note 1) In slave mode, it is at the time of <TASM> =0

Note 2) In slave mode, it is at the time of <TASM> =1

The Interrupts can be disabled individually at the interrupt controller.

3.11.10 Usage of the micro DM (micro DMA mode)

The usage of the micro DMA for larger SEI transfers allows speed up the communication on the SEI by

- reducing the CPU effort for interrupt processing,
- reducing the time gap between two successive transfers.

The micro DMA transfers can be used in both the master and the slave mode.

(1) Read/write micro DMA transfer

In this mode two micro DMA channels are used. One micro DAM channel is used to send the receive data from the SEDR register to the memory. The other micro DMA channel is used to send the new data from th memory to the SEDR register. The data transfer will be completely handled by the micro DMA controller.

① Initiation

Two micro DMA channels have to be set up for the transfer. One micro DMA is triggered on the INTSER t transfer the value that was received from the SEDR register to the memory. The other micro DMA is triggered on the INTSET to write new data from the memory to the SEDR register. It is used to restart the transfer i master mode.

The micro DMA with the lower channel has to be assigned to the INTSE interrupt since it takes precedenc over the micro DMA with the higher channel number.

The micro DMA transfer is initiated the first time by writing the first transfer value to the SEDR register. The following transfers will be handled automatically by the micro DMA controller.

<SEE>	<MSTR>	<TASM>	<TMSE>
1	0:Slave	INTSEE interrupt mask	1
	1:Master	0	

② Micro DMA transfer

Once initiated the micro DMA wait to be triggered by a completed transfer. On a completed transfer both <TSRC> and <TSTC> flags set and both SEI receive completed interrupt pulse INTSER and SEI transmit completed interrupt pulse INTSET are generated. Since the micro DMA channel with the lower channel number takes precedence, the read micro DMA transfer is performed before the write micro DMA transfer. The read micro DMA reads the value from the SEDR register and stores the value at the location specified within th micro DMA control registers. The read access also clears the <TSRC> flag in effect. After this the write micro DMA transfers a value from a specified memory address to the SEDR register. The write access to the SEDR register automatically clears the <TSTC> flag in the SESR register and starts a new transfer when the modul is in master mode. After each micro DMA transfer the count registers for both micro DMA are decreased. This procedure continues until the counters reach the value of "0". A micro DMA interrupt will be generated to indicate the end of the micro DMA transfer. An interrupt service routine triggered on the end of the micro DMA transfer can be used to re-initiate the micro DMA transfers.

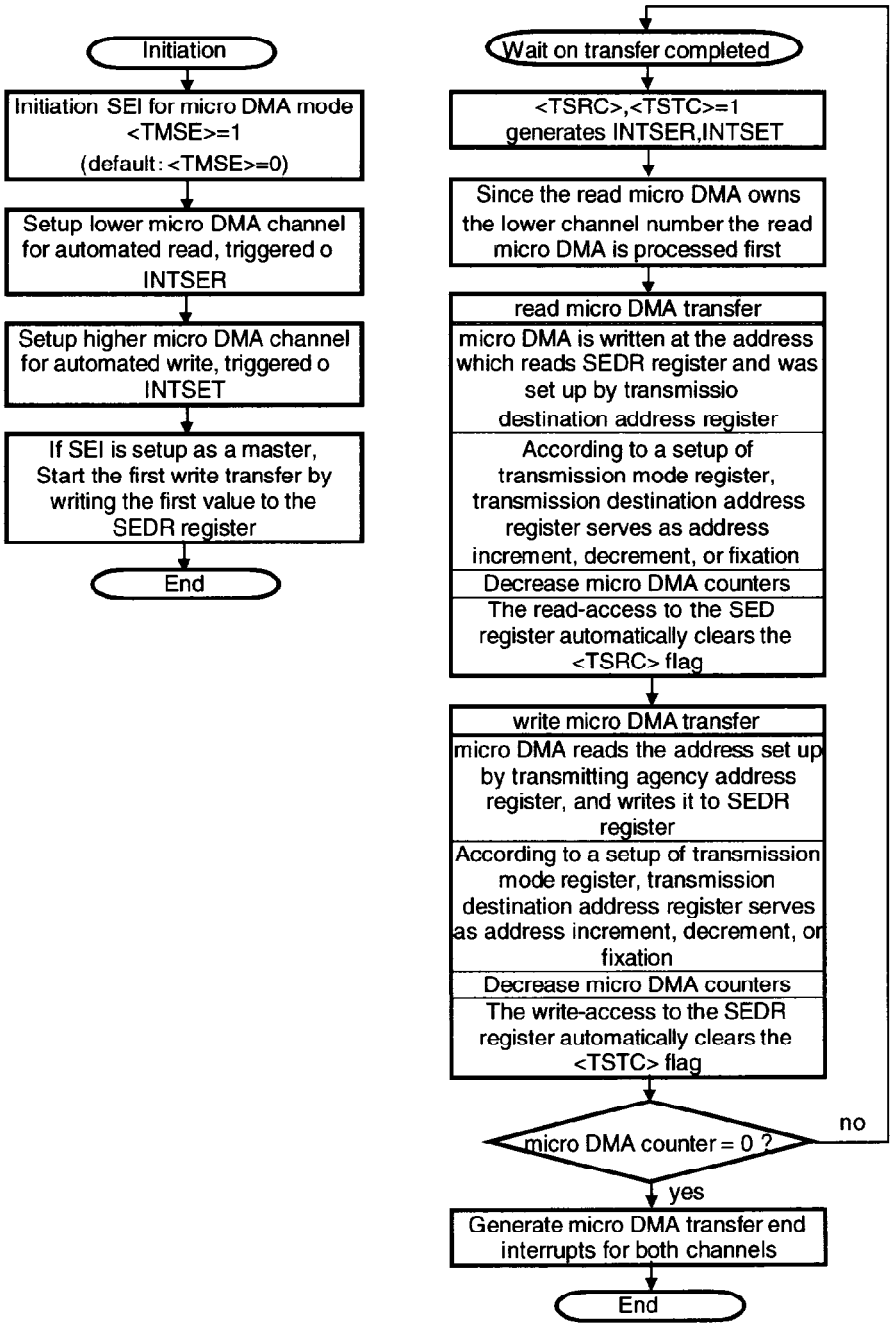


Figure 3.11.10(1) Flowchart for Micro DMA Read/Write Transfer

(2) Read only micro DMA transfer

This mode is used to shift in lager blocks of data, whil "don't care data " is shifted out (e.g.: reads from serial EEPROM). Only a single micro DMA is used to store the data read from the SEDR register to a specified RAM area.

① Initiation

For this mode the module has to be configured for micro DMA mode by asserting the <TMSE> bit in the SESR register. When SEI is acting as master, the <TASM> bit has to be set additionally to allow the automated shifting. Just one micro DMA has to be set up to transfer the SEDR data to a memory location specified within the micro DMA destination address register. The SEI receive completion interrupt INTSER is used to trigger this micro DMA. The SEI transfer completion interrupt INTSET is disabled at the interrupt controller. If SEI is set up as a master, the first transfer has to be initiated by writing the SEDR register.

<SEE>	<MSTR>	<TASM>	<TMSE>
1	0:Slave	INTSEE interrupt mask	1
	1:Master	1	

② Micro DMA transfer

After initiating the first transfer, the micro DMA waits for the transfer to be completed. With the completion of th transfer both the <TSRC> and <TSTC> flags in the SESR register are asserted. On the assertion of th <TSRC> flag the INTSER interrupt is generated to trigger the micro DMA. The <TSTC> flag will be asserted simultaneously and will remain set till the end of the block transfer.

The micro DMA moves the received value from the SEDR register to the memory location specified in destination address register. After the micro DMA transfer, the count register of the micro DMA is decreased. When the data register is read and the <TASM> bit in the SESR register is "1", the SEDR register automatically clears to "00" Hex. Simultaneously a new transfer is started automatically. This procedure will repeat until th micro DMA counter reaches a value of "0". A micro DMA interrupt will be generated to indicate the end of th micro DMA transfer.

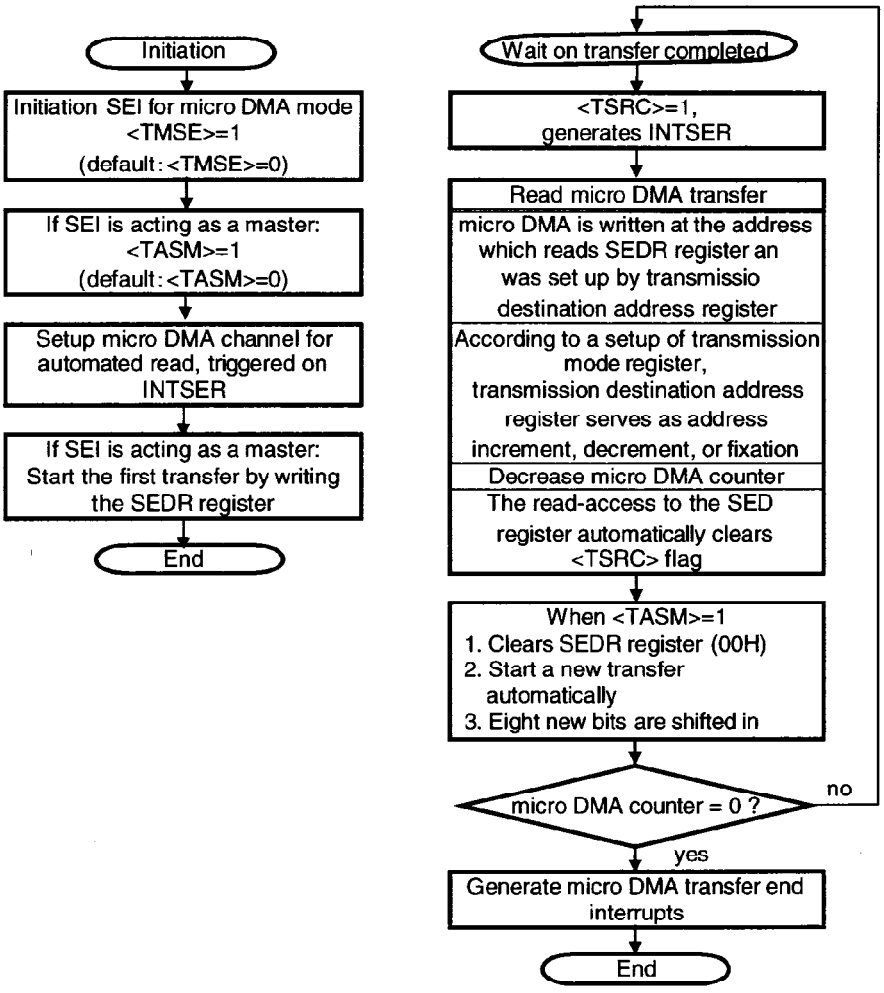


Figure 3.11.10(2) Flowchart for Micro DMA Read only Transfer

(3) Write only micro DMA transfe

The write only transfer mode is used to transmit larger blocks of data while the incoming data is ignored. Only a single micro DMA is used to transfer new transmit data from a memory location specified by the micro DMA source address register to the SEDR register.

① Initiation

For this mode the module has to be configured for micro DMA mode by asserting the <TMSE> bit in the SESR register. One of the micro DMA channels has to be set up for the automated write to the SEDR register. This micro DMA is triggered by the SEI transmit completion interrupt INTSET. The SEI receive completion interrupt INTSER is disabled at the interrupt controller. If SEI is set up as a master, the first transfer is initiated by writing the first value to the SEDR register.

<SEE>	<MSTR>	<TASM>	<TMSE>
1	0:Slave	INTSEE interrupt mask	1
	1:Master	0	

② Micro DMA transfer

After starting the first transfer the micro DMA waits for the transfer to be completed. On completion both th <TSRC> and <TSTC> flags in the SEI status register are asserted. The <TSRC> flag and in slave mod additionally the <SOVF> flag has to be ignored for the rest of the transfer. The <TSTC> flag generates th INTSET interrupt, which will trigger the micro DMA transfer.

The micro DMA reads a value from the memory address specified in its source register and transfers it to th SEDR register. After the micro DMA transfer, the count register of the micro DMA is decreased. The writ access to the SEDR register clears the <TSTC> flag and starts a new transfer on the SEI bus when the modul is in master mode. This procedure continues until the Micro DMA counter reaches a value of "0". A micro DMA interrupt will be generated to indicate the end of the micro DMA transfer.

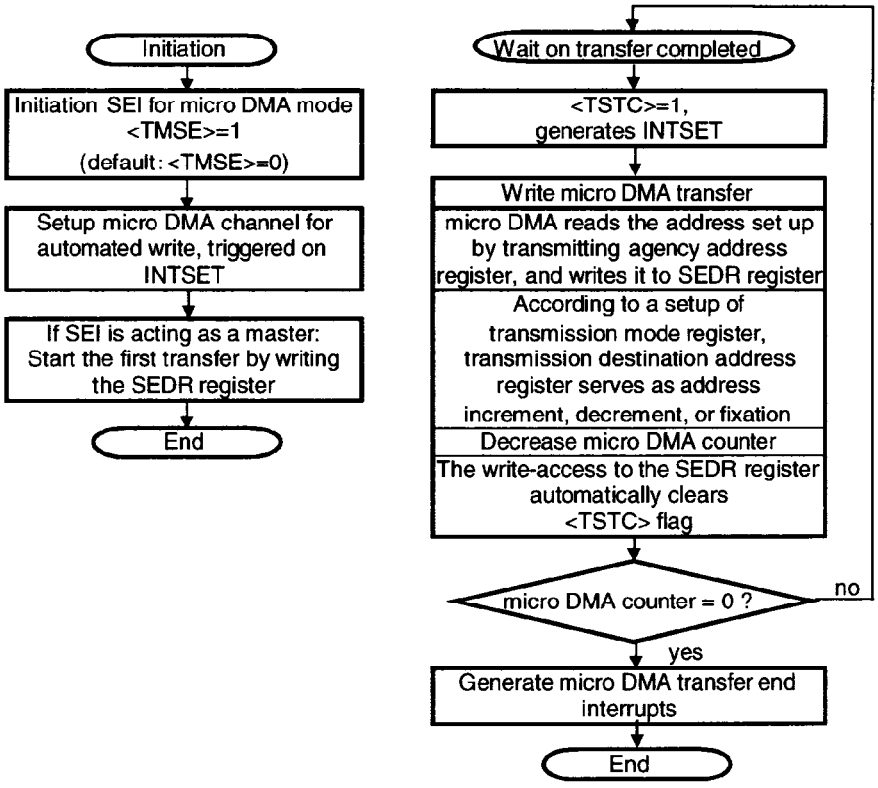


Figure 3.11.10(3) Flowchart for Micro DMA Write only Transfer

3.13 Analog/Digital Converter

The TMP92CW10 incorporates a 10-bit successive approximation-type analog/digital converter (AD converter) with 12-channel analog input.

Figure 3.13.1 is a block diagram of the AD converter. The 12-channel analog input pins (AN0 to AN11) are shared with the input-only port Port G and Port L, so they can be used as an input port.

Note: When IDLE2, IDLE1 or STOP Mode is selected, as to reduce the power, with some timings the system may enter a standby mode even though the internal comparator is still enabled. Therefore be sure to check that AD converter operations are halted before a HALT instruction is executed.

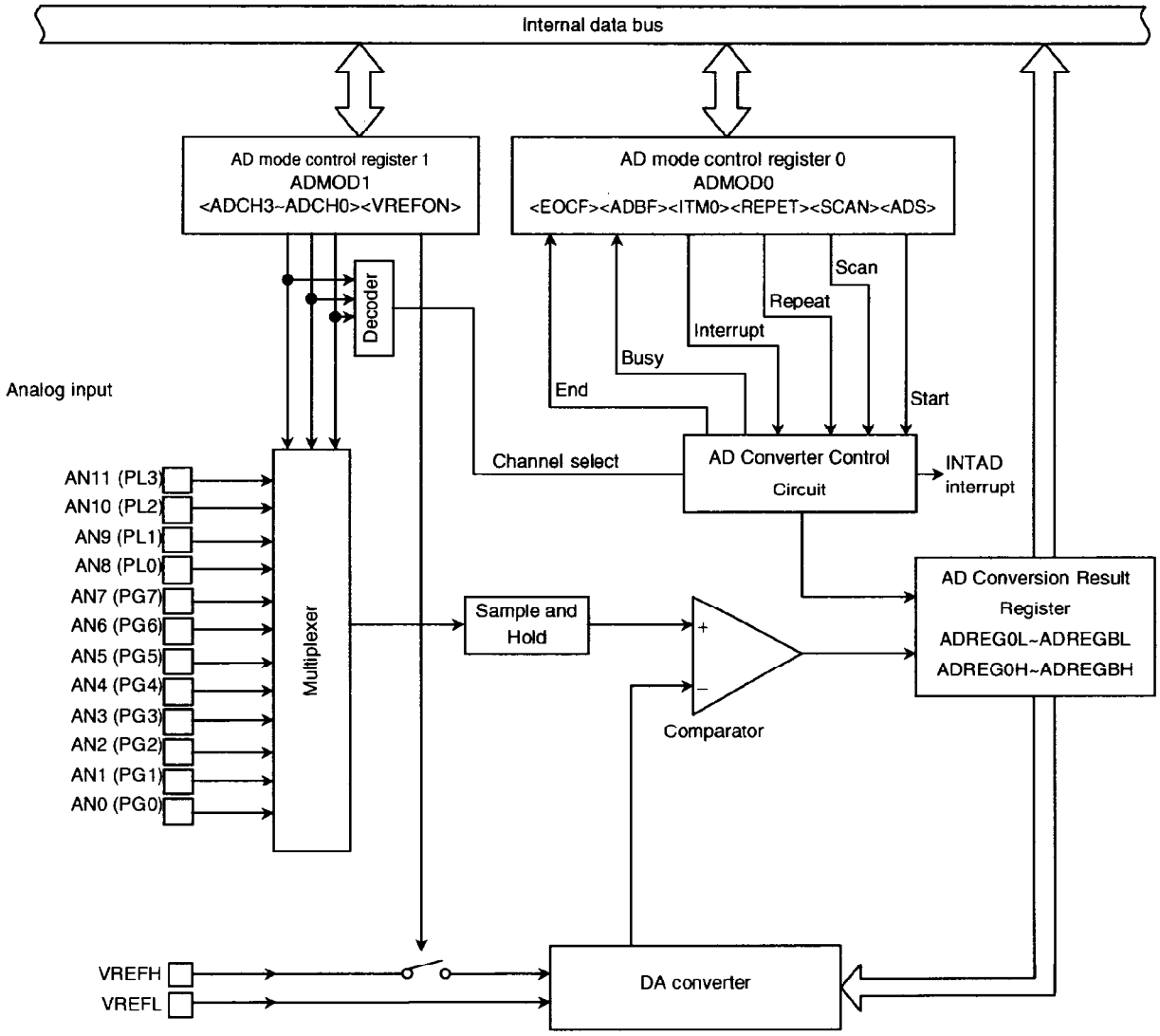


Figure 3.13.1 Block diagram of AD converter

3.13.1 Analog/Digital converter registers

The AD converter is controlled by the two AD Mode Control Registers: ADMOD0 and ADMOD1. The twelve AD Conversion Data Result Registers (ADREG0H/L, ADREGBH/L) store the results of AD conversion.

Figure 3.13.2 shows the registers related to the AD converter.

AD Mode Control Register 0

		7	6	5	4	3	2	1	0
ADMOD0 (0138H)	Bit symbol	EOCF	ADBF	–	–	ITM0	REPEAT	SCAN	ADS
	Read/Writ	R				R/W			
	After Reset	0	0	0	0	0	0	0	0
	Function	AD Conversion End flag 0: conversion in progress 1: conversion complete	A D Conversion Busy flag 0: conversion stopped 1: Conversion in progress	Note: Always fixed to 0	Note: Always fixed to 0	Interrupt specification in conversion channel fixed repeat mode 0: every conversion 1: every fourth conversion	Repeat mode specification 0: Singie Conversion 1: Repeat Conversion Mode	Scan mode specification 0: Conversion Channel Fixed Mode 1: Conversion Channel Scan Mode	AD conversion start 0: Don't care 1: start conversion Always 0 when read

AD conversion start

0	Don't care
1	Start AD conversion

Note: Always read as 0.

AD scan mode setting

0	AD Conversion Channel Fixed Mod
1	AD Conversion Channel Scan Mod

AD repeat mode setting

0	AD Single Conversion Mod
1	AD Repeat Conversion Mod

Specify AD conversion interrupt for Channel Fixed Repeat Conversion Mod

	Channel Fixed Repeat Conversion Mod <SCAN> = "0", <REPEAT> = "1"
0	Generates interrupt every conversion.
1	Generates interrupt every fourth conversion.

AD Conversion Busy flag

0	AD conversion stopped
1	AD conversion in progress

AD Conversion End flag

0	Before or during AD conversion
1	AD conversion complet

Figure 3.13.2 AD Converter Related Register

AD Mode Control Register 1

ADMOD1 (0139H)	7	6	5	4	3	2	1	0
Bit symbol	VREFON	I2AD	–	–	ADCH3	ADCH2	ADCH1	ADCH0
Read/Writ	R/W	R/W	R/W					
After Reset	0	0	0	0	0	0	0	0
Function	VREF application control 0: OFF 1: ON	IDLE2 0: Stop 1: Operate	Note: Always fixed to 0	Note: Always fixed to 0	Analog input channel selection			

Analog input channel selection

<ADCH2, ADCH1, ADCH0> \ <SCAN>	0 (channel fixed)	1 (channel scanned)
0000	AN0	AN0
0001	AN1	AN0 → AN1
0010	AN2	AN0 → AN1 → AN2
0011	AN3	AN0 → AN1 → AN2 → AN3
0100	AN4	AN0 → AN1 → AN2 → AN3 → AN4
0101	AN5	AN0 → AN1 → AN2 → AN3 → AN4 → AN5
0110	AN6	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6
0111	AN7	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7
1000	AN8	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8
1001	AN9	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8 → AN9
1010	AN10	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8 → AN9 → AN10
1011	AN11	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8 → AN9 → AN10 → AN11

IDLE2 control

0	Stopped
1	In operation

Control of application of reference voltage to AD converter

0	OFF
1	ON

Before starting conversion (before writing 1 to ADMOD0 <ADS>), set the <VREFON> bit to 1.

Figure 3.13.3 AD Converter Related Register

AD Conversion Result Register 0 Low

ADREG0L (0120H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR01	ADR00						ADR0RF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result							AD Conversion Data Storage flag 1: Conversion result stored

AD Conversion Result Register 0 High

ADREG0H (0121H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits AD conversion result.							

AD Conversion Result Register 1 Low

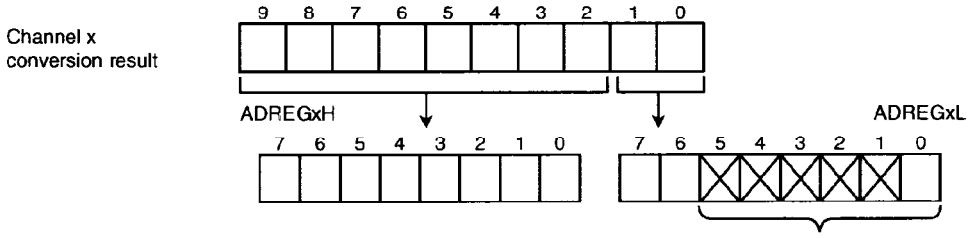
ADREG1L (0122H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR11	ADR10						ADR1RF
Read/Writ	R							R
After Reset	Undefined							0
Function	stores lower 2 bits of AD conversion result							AD Conversion Result flag 1: Conversion result stored

AD Conversion Result Register 1 High

ADREG1H (0123H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							



- Bits 5-1 are always read as 1.
- Bit 0 is the AD conversion data storage flag <ADRxRF>. When the A conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.4 AD Converter Related Registers

AD Conversion Result Register 2 Low

ADREG2L (0124H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR21	ADR20						ADR2RF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							A/D conversion data storage flag 1: Conversion result stored

AD Conversion Result Register 2 High

ADREG2H (0125H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							

AD Conversion Result Register 3 Low

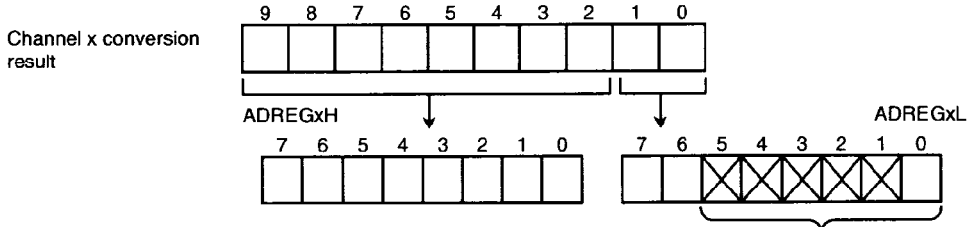
ADREG3L (0126H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR31	ADR30						ADR3RF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							AD Conversion Data Storage flag 1: conversion result stored

AD Conversion Result Register 3 High

ADREG3H (0127H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							



- Bits 5-1 are always read as 1.
- Bit 0 is the AD conversion data storage flag <ADRxRF>. When the A conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.5 AD Converter Related Registers

AD Conversion Result Register 4 Low

ADREG4L (0128H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR41	ADR40						ADR4RF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							A/D conversion data storage flag 1: Conversion result stored

AD Conversion Result Register 4 High

ADREG4H (0129H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							

AD Conversion Result Register 5 Low

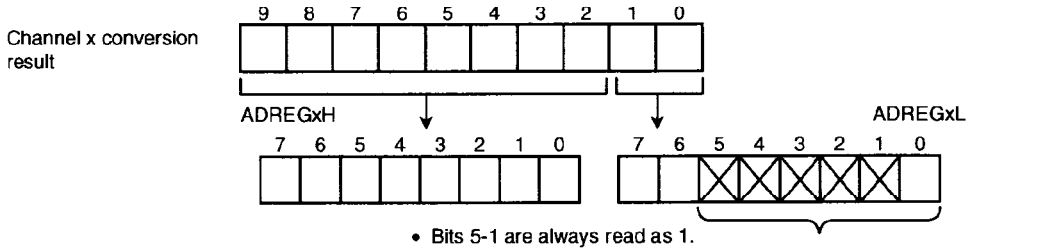
ADREG5L (012AH)

	7	6	5	4	3	2	1	0
Bit symbol	ADR51	ADR50						ADR5RF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							AD Conversion Data Storage flag 1: conversion result stored

AD Conversion Result Register 5 High

ADREG5H (012BH)

	7	6	5	4	3	2	1	0
Bit symbol	ADR59	ADR58	ADR57	ADR56	ADR55	ADR54	ADR53	ADR52
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							



- Bits 5-1 are always read as 1.
- Bit 0 is the AD conversion data storage flag <ADRxRF>. When the A conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.6 AD Converter Related Registers

AD Conversion Result Register 6 Low

ADREG6L (012CH)

	7	6	5	4	3	2	1	0
Bit symbol	ADR61	ADR60						ADR6RF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							A/D conversion data storage flag 1: Conversion result stored

AD Conversion Result Register 6 High

ADREG6H (012DH)

	7	6	5	4	3	2	1	0
Bit symbol	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							

AD Conversion Result Register 7 Low

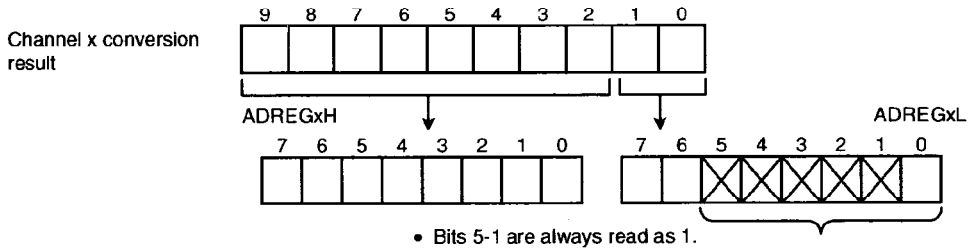
ADREG7L (012EH)

	7	6	5	4	3	2	1	0
Bit symbol	ADR71	ADR70						ADR7RF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							AD Conversion Data Storage flag 1: conversion result stored

AD Conversion Result Register 7 High

ADREG7H (012FH)

	7	6	5	4	3	2	1	0
Bit symbol	ADR79	ADR78	ADR77	ADR76	ADR75	ADR74	ADR73	ADR72
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							



- Bits 5-1 are always read as 1.
- Bit 0 is the AD conversion data storage flag <ADRxRF>. When the A conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.7 AD Converter Related Registers

AD Conversion Result Register 8 Low

ADREG8L (0130H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR81	ADR80						ADR8RF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							A/D conversion data storage flag 1: Conversion result stored

AD Conversion Result Register 8 High

ADREG8H (0131H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR89	ADR88	ADR87	ADR86	ADR85	ADR84	ADR83	ADR82
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							

AD Conversion Data Register 9 Low

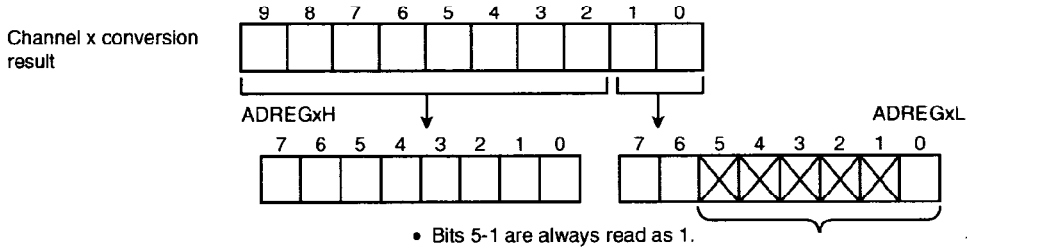
ADREG9L (0132H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR91	ADR90						ADR9RF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							AD Conversion Data Storage flag 1: conversion result stored

AD Conversion Result Register 9 High

ADREG9H (0133H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR99	ADR98	ADR97	ADR96	ADR95	ADR94	ADR93	ADR92
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							



- Bits 5-1 are always read as 1.
- Bit 0 is the AD conversion data storage flag <ADRxRF>. When the A conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.8 AD Converter Related Registers

AD Conversion Result RegisterA Low

ADREGAL (0134H)

	7	6	5	4	3	2	1	0
Bit symbol	ADRA1	ADRA0						ADRARF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							A/D conversion data storage flag 1: Conversion result stored

AD Conversion Result RegisterA High

ADREGAH (0135H)

	7	6	5	4	3	2	1	0
Bit symbol	ADRA9	ADRA8	ADRA7	ADRA6	ADRA5	ADRA4	ADRA3	ADRA2
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							

AD Conversion Result Register B Low

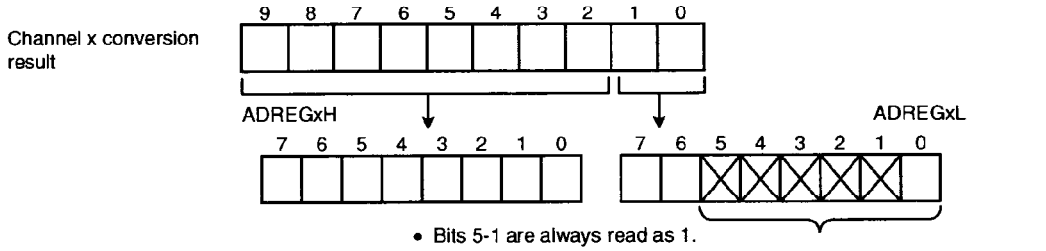
ADREGBL (0136H)

	7	6	5	4	3	2	1	0
Bit symbol	ADRB1	ADRB0						ADRBRF
Read/Writ	R							R
After Reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							AD Conversion Data Storage flag 1: conversion result stored

AD Conversion Result Register B High

ADREGBH (0137H)

	7	6	5	4	3	2	1	0
Bit symbol	ADRB9	ADRB8	ADRB7	ADRB6	ADRB5	ADRB4	ADRB3	ADRB2
Read/Writ	R							
After Reset	Undefined							
Function	Stores upper eight bits of AD conversion result.							



- Bits 5-1 are always read as 1.
- Bit 0 is the AD conversion data storage flag <ADRxRF>. When the A conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.9 AD Converter Related Registers

3.13.2 Description of operation

(1) Analog reference voltage

A high-level analog reference voltage is applied to the VREFH pin; a low-level analog reference voltage is applied to the VREFL pin. To perform AD conversion, the reference voltage, the difference between VREFH and VREFL, is divided by 1024 using string resistance. The result of the division is then compared with the analog input voltage.

To turn off the switch between VREFH and VREFL, write a 0 to ADMOD1<VREFON> in AD Mode Control Register 1. To start AD conversion in the OFF state, first write a 1 to ADMOD1<VREFON>, wait 3μs until the internal reference voltage stabilizes (this is not related to fc), then set ADMOD0<ADS> to 1.

(2) Analog input channel selection

The analog input channel selection varies depends on the operation mode of the AD converter.

- In Analog Input Channel Fixed Mode (ADMOD0<SCAN> = 0)
  Setting ADMOD1<ADCH3~ADCH0> selects one of the input pins AN0~AN11 as the input channel.
- In Analog Input Channel Scan Mode (ADMOD0<SCAN> = 1)
  Setting ADMOD1<ADCH3~ADCH0> selects one of the twelve scan modes.

Table 3.13.1 illustrates analog input channel selection in each operation mode.

On a Reset, ADMOD0<SCAN> is set to 0 and ADMOD1<ADCH 3~ADCH0> is initialized to 0000. Thus pin AN0 is selected as the fixed input channel. Pins not used as analog input channels can be used as standard input port pins.

Table 3.13.1 Analog input channel selection

<ADCH2~0>	Channel fixed <SCAN> = "0"	Channel scan <SCAN> = "1"
0000	AN0	AN0
0001	AN1	AN0 → AN1
0010	AN2	AN0 → AN1 → AN2
0011	AN3	AN0 → AN1 → AN2 → AN3
0100	AN4	AN0 → AN1 → AN2 → AN3 → AN4
0101	AN5	AN0 → AN1 → AN2 → AN3 → AN4 → AN5
0110	AN6	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6
0111	AN7	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7
1000	AN8	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8
1001	AN9	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8 → AN9
1010	AN10	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8 → AN9 → AN10
1011	AN11	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8 → AN9 → AN10 → AN11

(3) Starting AD Conversion

To start AD conversion, write a 1 to ADMOD0<ADS> in AD Mode Control Register 0 .When AD conversion starts, the AD Conversion Busy flag ADMOD0<ADBF> will be set to 1, indicating that AD conversion is in progress.

(4) AD conversion modes and the AD Conversion End interrupt

The four AD conversion modes are:

- Channel Fixed Single Conversion Mode
- Channel Scan Single Conversion Mode
- Channel Fixed Repeat Conversion Mode
- Channel Scan Repeat Conversion Mode

The ADMOD0<REPET> and ADMOD0<SCAN> settings in AD Mode Control Register 0 determine the AD mode setting.

Completion of AD conversion triggers an INTAD AD Conversion End interrupt request. Also, ADMOD0<EOCF> will be set to 1 to indicate that AD conversion has been completed.

① Channel Fixed Single Conversion Mode

Setting ADMOD0<REPET> and ADMOD0<SCAN> to 00 selects Conversion Channel Fixed Single Conversion Mode.
In this mode data on one specified channel is converted once only. When the conversion has been completed, the ADMOD0<EOCF> flag is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

② Channel Scan Single Conversion Mode

Setting ADMOD0<REPET> and ADMOD0<SCAN> to 01 selects Conversion Channel Scan Single Conversion Mode.
In this mode data on the specified scan channels is converted once only. When scan conversion has been completed, ADMOD0<EOCF> is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

③ Channel Fixed Repeat Conversion Mode

Setting ADMOD0<REPET> and ADMOD0<SCAN> to 10 selects Conversion Channel Fixed Repeat Conversion Mode.
In this mode data on one specified channel is converted repeatedly. When conversion has been completed, ADMOD0<EOCF> is set to 1 and ADMOD0<ADBF> is not cleared to 0 but held at 1. INTAD interrupt request generation timing is determined by the setting of ADMOD0<ITM0>.
Setting <ITM0> to 0 generates an interrupt request every time an AD conversion is completed.
Setting <ITM0> to 1 generates an interrupt request on completion of every fourth conversion.

④ Channel Scan Repeat Conversion Mode

Setting ADMOD0<REPET> and ADMOD0<SCAN> to 11 selects Conversion Channel Scan Repeat Conversion Mode.
In this mode data on the specified scan channels is converted repeatedly. When each scan conversion has been completed, ADMOD0<EOCF> is set to 1 and an INTAD interrupt request is generated. ADMOD0<ADBF> is not cleared to 0 but held at 1.
To stop conversion in a repeat conversion mode (i.e. in cases③ and ④), write a 0 to ADMOD0<REPET>. After the current conversion has been completed, the repeat conversion mode terminates and ADMOD0<ADBF> is cleared to 0.
Switching to a halt state (IDLE2 Mode with ADMOD1<I2AD> cleared to 0, IDLE1 Mode or STOP Mode) immediately stops operation of the AD converter even when AD conversion is still in progress. In repeat conversion modes (i.e. in case③ and ④), when the halt is released, conversion restarts from the beginning. In single conversion modes (i.e. in cases ① and ②), conversion does not restart when the halt is released (the converter remains stopped).

Table 3.13.2 shows the relationship between the AD conversion modes and interrupt requests.

Table 3.13.2 Relationship Between AD Conversion Modes and Interrupt Requests

Mode	Interrupt Request Generation	ADMOD0		
		<ITM0>	<REPEAT>	<SCAN>
Channel Fixed Single Conversion Mod	After completion of conversion	X	0	0
Channel Scan Single Conversion Mod	After completion of scan conversion	X	0	1
Channel Fixed Repeat Conversion Mod	Every conversion	0	1	0
	Every forth conversion	1		
Channel Scan Repeat Conversion Mod	After completion of every scan conversion	X	1	1

X: Don't care

(5) AD conversion time

160/fc (8 μs @ f_c = 20 MHz) are required for the AD conversion of one channel.

(6) Storing and reading the results of AD conversion

The AD Conversion Data Upper and Lower Registers (ADREG0H/L to ADREGBH/L) store the results of AD conversion. (ADREG0H/L to ADREGBH/L are read-only registers.)

In Channel Fixed Repeat Conversion Mode, the conversion results are stored successively in registers ADREG0H/L to ADREG3H/L. In other modes the AN0, AN1, AN2, AN3, AN4, AN5, AN6, AN7 conversion results are stored in ADREG0H/L, ADREG1H/L, ADREG2H/L, ADREG3H/L, ADREG4H/L, ADREG5H/L, ADREG6H/L, ADREG7H/L, ADREG8H/L, ADREG9H/L, ADREGAH/L, ADREGBH/L respectively. Table 3.13.3 shows the correspondence between the analog input channels and the registers which are used to hold the results of AD conversion.

Table 3.13.3 Correspondence Between Analog Input Channels and AD Conversion Result Registers

Analog input channel (Port G/Port L)	AD Conversion Result Register	
	Conversion modes other than at right	Channel fixed repeat conversion mode (every 4th conversion)
AN0	ADREG0H/L	ADREG0H/L → ADREG1H/L → ADREG2H/L → ADREG3H/L → ADREG0H/L
AN1	ADREG1H/L	
AN2	ADREG2H/L	
AN3	ADREG3H/L	
AN4	ADREG4H/L	
AN5	ADREG5H/L	
AN6	ADREG6H/L	
AN7	ADREG7H/L	
AN8	ADREG8H/L	
AN9	ADREG9H/L	
AN10	ADREGAH/L	
AN11	ADREGBH/L	

<ADRxRF>, bit 0 of the AD conversion data lower register, is used as the AD conversion data storage flag. The storage flag indicates whether the AD conversion result register has been read or not. When a conversion result is stored in the AD conversion result register, the flag is set to 1. When either of the AD conversion result registers (ADREGxH or ADREGxL) is read, the flag is cleared to 0.

Reading the AD conversion result also clears the AD Conversion End flag ADMOD0<EOCF> to 0.

Setting example:

① Convert the analog input voltage on the AN3 pin and write the result, to memory address 0800H using the AD interrupt (INTAD) processing routine.

Main routine:

```
                7 6 5 4 3 2 1 0
INTE0AD  ← 1 1 0 0 - - - -   Enable INTAD and set it to Interrupt Level 4.
ADMOD1   ← 1 1 0 0 0 0 1 1   Set pin AN3 to be the analog input channel.
ADMOD0   ← X X 0 0 0 0 0 1   Start conversion in Channel Fixed Single Conversion Mode.
```

Interrupt routine processing example:

```
WA       ← ADREG3    Read value of ADREG3L and ADREG3H into 16-bit general-
                     purpose register WA.
WA       >>6         Shift contents read into WA six times to right and zero-fill upper
                     bits.
(0800H)  ← WA        Write contents of WA to memory address 0800H.
```

② This example repeatedly converts the analog input voltages on the three pins AN0, AN1 and AN2, using Channel Scan Repeat Conversion Mode.

```
INTE0AD  ← 1 0 0 0 - - - -   Disable INTAD.
ADMOD1   ← 1 1 0 0 0 0 1 0   Set pins AN0~AN2 to be the analog input channels.
ADMOD0   ← X X 0 0 0 1 1 1   Start conversion in Channel Scan Repeat Conversion Mode.
```

Note: X = Don't care; "–" = No change

3.14 Watchdog Timer (Runaway Detection Timer)

The TMP92CW10 contains a watchdog timer ofrunaway detecting.
The watchdog timer (WDT) is used to return the CPU to the normal state when it detects the CPU has started to malfunction (runaway) due to causes such as noise. When the watchdog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU of the malfunction.
Connecting the watchdog timer output to the reset pin internally forces a reset.

3.14.1 Configuration

Figure 3.14.1(1) is a block diagram of the watchdog timer (WDT).

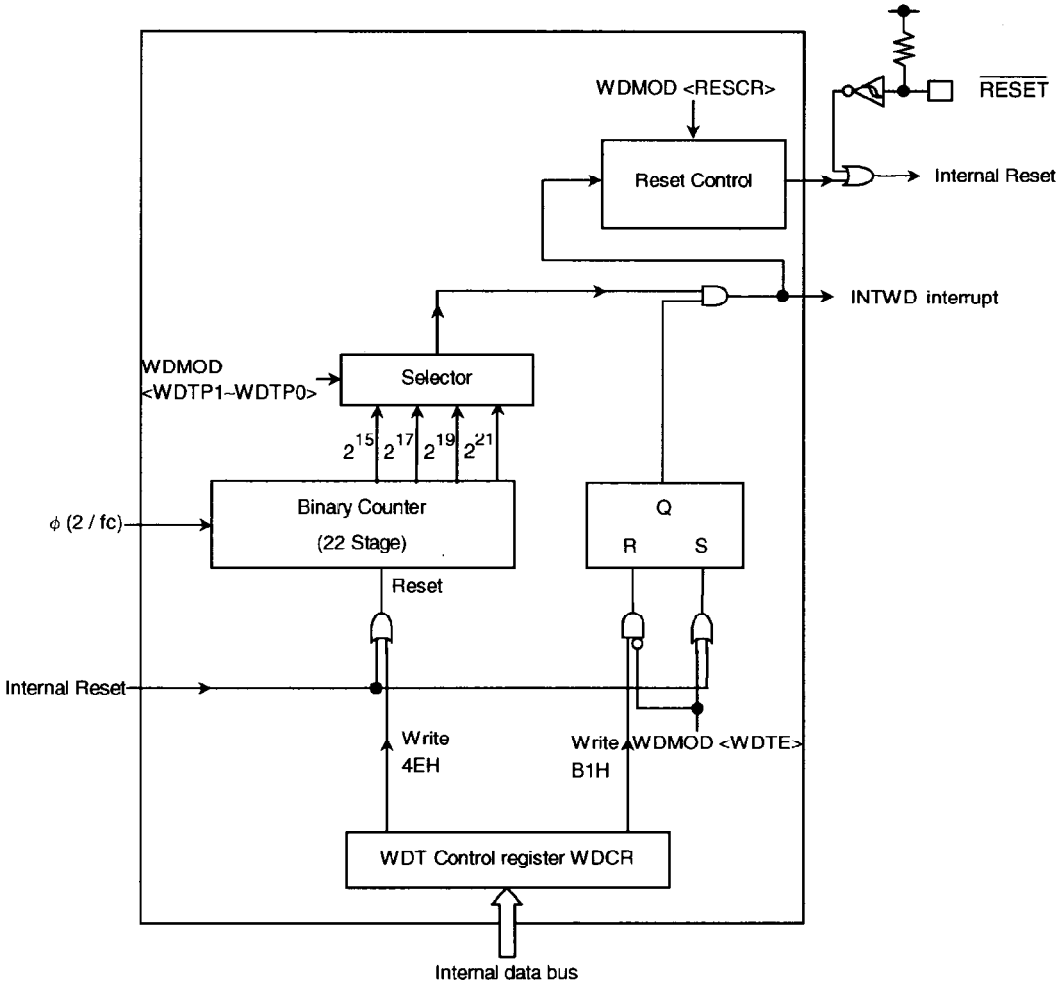


Figure 3.14.1(1) Block Diagram of Watchdog Timer

The watchdog timer consists of a 22-stage binary counter which uses the clock ϕ (2/fc) as the input clock. The binary counter can output2^{16}/fc, 2^{18}/fc, 2^{20}/fc and 2^{22}/fc. Selecting one of the outputs using WDMOD<WDTP1,WDTP0> generates a watchdogtimer interrupt when an overflow occurs.

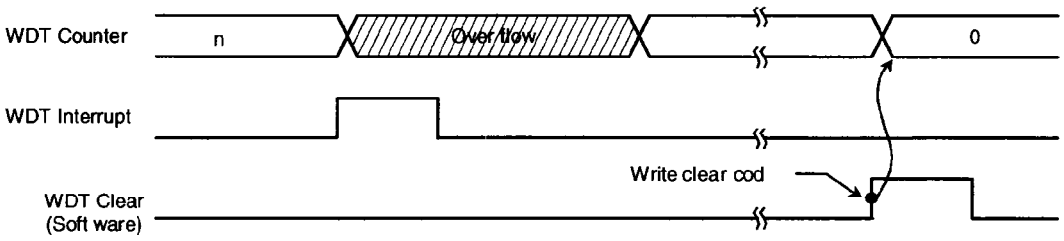


Figure 3.14.1(2) Normal Mode

The runaway detection result can also be connected to the Reset pin internally.
In this case, the reset time will be between 44 and 58 system clocks (2.2~2.9 μs @ f_C = 20 MHz) as shown in figure 3.14.1(3).

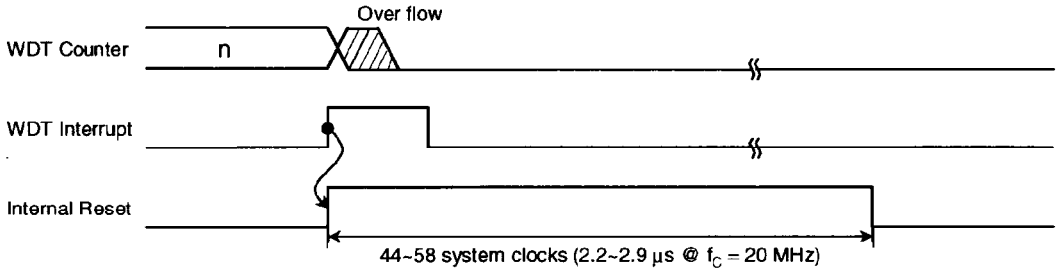


Figure 3.14.1(3) Reset Mode

3.14.2 Control registers

The watchdog timer WDT is controlled by three control registers WDMOD, WDCR and CLKMOD.

(1) Watchdog Timer Mode Register (WDMOD)

① Setting the detection time for the watchdog timer in <WDTP1,WDTP0>

This 2-bit register is used for setting the watchdog timer interrupt time used when detecting runaway. On a Reset this register is initialized to WDMOD<WDTP1,WDTP0> = 00.
The detection times for WDT is 2^{16}/fc [S]. (The number of system clocks is approximately 65,536.)

② Watchdog timer enable/disable control register <WDTE>

At reset, the WDMOD<WDTE> is initialized to 1, enabling the watchdog timer.
To disable the watchdog timer, it is necessary to set this bit to 0 and to write the disable code (B1H) to 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 the watchdog timer from the disabled state to the enabled state merely by setting <WDTE> to 1.

③ Watchdog timer out reset connection <RESCR>

This register is used to connect the output of the watchdog timer with the RESET terminal internally. Since WDMOD<RESCR>is initialized to 0 at reset, a reset by the watchdog timer will not be performed.

(2) Watchdog Timer Control Register (WDCR)

This register is used to disable and clear the binary counter for the watchdog timer.

- Disable control

  The watchdog timer can be disabled by clearing WDMOD<WDTE> to 0 and then writing the disable code (B1H) to the WDCR register.

```
WDMOD   ← 0 - - - - - - -   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

  To clear the binary counter and cause counting to resume, write the clear code (4EH) to the WDCR register.

```
WDCR    ← 0 1 0 0 1 1 1 0   Write the clear code (4EH).
```

(3) Clock Mode Register (CLKMOD)

This register is used to set the warming up time after the stop mode ends.
Writing "0" to the CLKMOD <WARM> bit, 2^5/fc (approximately 1.6ms @ 20MHz) is selected and writing "1", 2^{17}/fc (approximately 6.6ms @ 20MHz) is selected.
The output of CLK pin is chosen from fc and 2/5fc by the setup of CLKMOD <CLKM1,CLKM0>.
Moreover, CLK pin output can be stopped by writing"0" in CLKMOD <CLKOE>.
By the setup of CLKMOD <HALTM1,HALTM0>, it becomes the HALT mode of IDLE2, IDLE1 or STOP.

WDMOD (0110H)

	7	6	5	4	3	2	1	0
bit symbol	WDTE	WDTP1	WDTP0		DRVE	I2WDT	RESCR	–
Read/Writ	R/W				R/W			
After reset	1	0	0		0	0	0	0
Function	WDT control 1: enable	Select detecting tim 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$			1: Drives pins in STOP mode	IDLE2 0: Stop 1: Operate	1: Internall connects WDT out to the reset pin	Always write 0

Watchdog timer out control

0	—
1	Connects WDT out to a reset

IDLE2 control

0	Stop
1	Operation

Watchdog timer detection tim

00	$2^{16}/f_C$ (approximately 3.28ms @ fc = 20MHz)
01	$2^{18}/f_C$ (approximately 13.1ms @ fc = 20MHz)
10	$2^{20}/f_C$ (approximately 52.4ms @ fc = 20MHz)
11	$2^{22}/f_C$ (approximately 210ms @ fc = 20MHz)

Watchdog timer enable/disable control

0	Disabled
1	Enabled

Figure 3.14.2(1) Watchdog Timer Mode Register

WDCR (0111H)

	7	6	5	4	3	2	1	0
Bit symbol	–							
Read/Writ	W							
After reset	–							
Function	B1H: WDT disable code 4EH: WDT clear cod							

WDT disable/clear control

B1H	Disable code
4EH	Clear cod
Others	Don't care

Figure 3.14.2(2) Watchdog Timer Control Register

CLKMOD (010AH)

	7	6	5	4	3	2	1	0
bit Symbol	HALTM1	HALTM0		WARM		CLKOE	CLKM1	CLKM0
Read/Writ	R/W			R/W		R/W		
After reset	1	1		0		0	0	0
Function	Standby mod 00: Reserved 01: STOP 10: IDLE1 11: IDLE2			Warming up tim 0: $2^{15}/f_C$ 1: $2^{17}/f_C$		CLKOUT1 output enable 0: not output 1: output	CLKOUT1 output select 00: fc 01: Reserved 10: 2/5 fc 11: Reserved	

CLKOUT1 output clock select

00	Fc
01	Reserved
10	2/5 fc
11	Reserved

CLKOUT1 output enabl

0	Not output (Pull up)	TMP94FD53F
	High impedance	TMP92CW10F
1	Output (Pull up)	TMP94FD53F
	Output	TMP92CW10F

Select warming up time after stop mod ends

0	$2^{15}/f_C$
1	$2^{17}/f_C$

Selects standby mode byHALT instruction

00	Reserved
01	STOP
10	IDLE1
11	IDLE2

Figure 3.14.2(3) Clock Mode Register

3.14.3 Operation

The watchdog timer generates an INTWD interrupt when the detection time set in the WDMOD<WDTP1,WDTP0> has elapsed. The watchdog timer must be zero-cleared in software before an INTWD interrupt will be generated. If the CPU malfunctions (i.e. if runaway occurs) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter will overflow and an INTWD interrupt will be generated. The CPU will detect malfunction (runaway) due to the INTWD interrupt and in this case it is possible to return to the CPU to normal operation by means of an anti-mulfunction program.

The watchdog timer begins operating immediately on release of the watchdog timer reset.

The watchdog timer is reset and halted in IDLE1 or STOP Modes. The watchdog counter continues counting during bus release (When $\overline{\text{BUSAK}}$ goes Low).

When the device is in IDLE2 mode. the operation of WDT depends on the WDMOD<I2WDT> setting. Ensure that WDMOD<I2WDT> is set before the device enters IDLE2 Mode.

Example: ① Clear the binary counter.

```
WDCR    ← 0 1 0 0 1 1 1 0      Write the clear code (4EH).
```

② Set the watchdog timer detection time to $2^{18}/f_c$.

```
WDMOD   ← 1 0 1 - - - - -
```

③ Disable the watchdog timer.

```
WDMOD   ← 0 - - X - - - -      Clear <WDTE> bit to 0.
WDCR    ← 1 0 1 1 0 0 0 1      Write the disable code (B1H).
```

3.15 RAM control

RAM control register(RAMCR) has <RAMWI>bit for inhibition to write data to internal RAM and <RAMSTB> bit to detect lower voltage under VSTB level. VSTB level is the voltage level imposible to keep the data of Internal RAM.

Only data "1" can be written to RAMCR<RAMSTB>, and data "0" can't be written.

When RAMCR<RAMSTB> is set to "1" by software, in the case of voltage drop under VSTB level RAMCR<RAMSTB> is reset to "0". After power on RAMCR<RAMSTB> is reset to "0".

RAMCR<RAMSTB> is not changed by standby operation and reset operation. The detection of reset operation (Warm reset / Power-on reset) and the conditon of RAM data (kept / lost) is enable, to read RAMCR<RAMSTB>.

RAM Write Inhibit<RAMWI> bit is used for inhibition to write data to internal RAM. After reset RAMCR<RAMWI> is set to "1", writing data to internal RAM is accepted. When RAMCR<RAMWI> is set to "0", writing data to internal RAM is inhibited.

RAMCR (016DH)

	7	6	5	4	3	2	1	0
Bit Symbo	RAMSTB	RAMWI	-	-	-	-	-	-
Read/Writ	R/W							
After reset	Not Changed	1						
Function	RAM data / Reset 0:Lost / Power-on reset 1:Kept	Internal RAM write 0:Inhibit 1:accept						

Write contorol to Internal RAM

0	Inhibit to write to Internal RAM
1	Accept to write to Internal RAM

RAM standby flag

0	After "1" is set by software, this bit is reset to "0" at VCC3 ≦VSTB. After power on reset.
1	After "1" is set by software, this data isn't changed at VCC3>VSTB.

Note1: After power on RAMCR<RAMSTB> is reset to "0", but warm reset don't change RAMCR<RAMSTB>. When this bit is used it need to set "1" by software, and so data "0" can't be written.

Note2: Standby current occurs to set standby mode when RAMCR<RAMSTB> bit is set to "1".

Note3: RAM control functions aren't supported by emulator.

Note4: When RAMCR<RAMSTB> set to "1", a voltage detection circuit operate after a wait of 8-States(@ fc = 20MHz; The while, do not set Idle2, 3 or STOP mode.). After that, the power-supply detection circuit runs.

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Power Supply Voltage	V_{CC3}	-0.5~4.5	V
	V_{CC5}	-0.5~6.5	
Input Voltage	V_{IN}	-0.5~VCC3+0.5	V
		-0.5~VCC5+0.5	
Output Current(total)	ΣI_{OL}	100	mA
Output Current(total)	ΣI_{OH}	-100	mA
Power Dissipation(Ta=85°C)	P_D	600	mW
Soldering Temperature(10s)	T_{SOLDER}	260	°C
Storage Temperature	T_{STG}	-65~150	°C
Operation Temperature	T_{OPR}	-40~85	°C

Note: The absolute maximum ratings are rated values that must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products that include this device, ensure that no absolute maximum rating value will ever be exceeded.

4.2 DC Electrical Characteristics

Vcc3 = 3.0 to 3.6V / Vcc5 =4.5V to 5.5V / fc = 16 to 20MHz / Ta = -40 to 85°C

Parameter	Symbol	Condition	Min	Max	Unit
Supply Voltage	V_{CC3}		3.0	3.6	V
	V_{CC5}		4.5	5.5	
Input Low Voltage P00 to P07 (D0 to 7) PG0 to PG7 PL0 to PL3	V_{IL0}		-0.3	0.8	V
Input Low Voltage P00 to P07 (PORT) P40 to P47 P70 to P75 PC0 to PC5 PD0 to PD7 PF0 to PF7 PM0 to PM7 PN0 to PN5	V_{IL1}		-0.3	0.3*VCC5	V
Input Low Voltage INT0 $\overline{NMI}$ $\overline{RESET}$	V_{IL2}		-0.3	0.25*VCC5	V
Input Low Voltage AM0 to AM1 TEST0 to TEST1	V_{IL3}		-0.3	0.3	V
Input Low Voltage X1	V_{IL4}		-0.3	0.2*VCC3	V
Input High Voltage P00 to P07 (D0 to 7) PG0 to PG7 PL0 to PL3	V_{IH0}		2.2	VCC5+0.3	V
Input High Voltage P00 to P07 P40 to P47 P70 to P75 PC0 to PC5 PD0 to PD7 PF0 to PF7 PM0 to PM7 PN0 to PN5	V_{IH1}		0.7*VCC5	VCC5+0.3	V
Input High Voltage INT0 $\overline{NMI}$ $\overline{RESET}$	V_{IH2}		0.75*VCC5	VCC5+0.3	V
Input High Voltage AM0 to AM1 TEST0 to TEST1	V_{IH3}		VCC5-0.3	VCC5+0.3	V
Input High Voltage X1	V_{IH4}		0.8*VCC3	VCC3+0.3	V

Parameter	Symbol	Condition	Min	Max	Unit
Output Low Voltage	V_{OL}	I_{OL} = 1.6mA		0.45	V
Output High Voltage	V_{OH0}	I_{OH} = -400μA	2.4		V
	V_{OH1}	I_{OH} = -100μA	0.75*VCC5		
	V_{OH2}	I_{OH} = -20μA	0.9*VCC5		
Input Leakage Current	I_{LI}	0.0 ≦ Vin ≦ VCC5	0.02(typ.)	±5	μA
	I_{L2}	0.0 ≦ Vin ≦ VCC5 (PortG, PortL)	0.02(typ.)	±0.5	
Output Leakage Current	I_{LO}	0.2 ≦ Vin ≦ VCC5-0.2	0.05(typ.)	±10	μA
Operating Current (Single Chip)*	I_{CC3}	V_{CC3}=3.3V, X1=10MHz (Internal 20MHz) (Includes I_{CCPLL})	85(typ)	100	mA
	I_{CC5}	V_{CC5}=5.0V, X1=10MHz (Internal 20MHz)	0.5(typ)	30	
Operating Current (Stand-by)	$I_{CC3IDLE2}$	IDLE2 Mode V_{CC3}=3.6V, X1=10MHz (Internal 20MHz)		90	mA
	$I_{CC5IDLE2}$	IDLE2 Mode V_{CC5}=5.5V, X1=10MHz (Internal 20MHz)		10	
	$I_{CC3IDLE1}$	IDLE1 Mode V_{CC3}=3.6V, X1=10MHz (Internal 20MHz)		20	
	$I_{CC5IDLE1}$	IDLE1 Mode V_{CC5}=5.5V, X1=10MHz (Internal 20MHz)		10	
	$I_{CC3STOP}$	STOP Mode V_{CC3}=3.6V		300	μA
	$I_{CC5STOP}$	STOP Mode V_{CC5}=5.5V		300	
Stand-by Voltage	V_{STB3}	$V_{DD3} < V_{DD5}$, $V_{IH1} < V_{DD5}$, $V_{IH2} < V_{DD5}$, $V_{IH3} < V_{DD5}$	2.5	3.6	V
	V_{STB5}		2.5	5.5	
Pull-up Resistor	R_{RST}	$\overline{RESET}$	60	220	KΩ
	R_{CLK}	CLK			
Pin Capacitance	C_{IO}	fc=1MHz		10	pF
Schmitt Width	V_{TH}	INT0, $\overline{NMI}$, $\overline{RESET}$	0.4	1.0(typ.)	V

*: On condition that external bus don't operate

4.3 AC Characteristics

Read cycle

VCC3=3.3V±0.3V, VCC5=5.0V±10%, TA=−40 to 85°C

No.	Parameter	Symbol	Min	Max	@20MHz	@16MHz	Unit
1	OSC period (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System Clock period (=T)	t_{CYC}	50	62.5	50	62.5	ns
3	CLKOUT1 Low Width	t_{CL}	0.5×T-15		10	16	ns
4	CLKOUT1 High Width	t_{CH}	0.5×T-15		10	16	ns
5-1	A0 to A23 Valid → D0 to D7 Input @0WAIT	t_{AD}		2.0×T-50	50	75	ns
5-2	A0 to A23 Valid → D0 to D7 Input @1WAIT	t_{ADW}		3.0×T-50	100	138	ns
6-1	$\overline{RD}$ Fall → D0 to D7 Input @0WAIT	t_{RD}		1.5×T-45	30	49	ns
6-2	$\overline{RD}$ Fall → D0 to D7 Input @1WAIT	t_{RDW}		2.5×T-45	80	111	ns
7-1	$\overline{RD}$ Low Width @0WAIT	t_{RR}	1.5×T-20		55	74	ns
7-2	$\overline{RD}$ Low Width @1WAIT	t_{RRW}	2.5×T-20		105	136	ns
8	A0 to A23 Valid → $\overline{RD}$ Fall	t_{AR}	0.5×T-20		5	11	ns
9	$\overline{RD}$ Fall → CLK Fall	t_{RK}	0.5×T-20		5	11	ns
10	A0 to A23 Valid → D0 to D7 Hold	t_{ADH}	0		0	0	ns
11	$\overline{RD}$ Rise → D0 to D7 Hold	t_{RD}	0		0	0	ns
12	A0 to A23 Valid → PORT Input	t_{APO}		2.0×T-120	-20	5	ns
13	A0 to A23 Valid → PORT Hold	t_{APH}	2.0×T		100	125	ns
14	$\overline{WAIT}$ Set-up Time	t_{WS}	15		15	15	ns
15	$\overline{WAIT}$ Hold Time	t_{WH}	5		5	5	ns

Write cycle

VCC3=3.3V±0.3V, VCC5=5.0V±10%, TA=−40 to 85°C

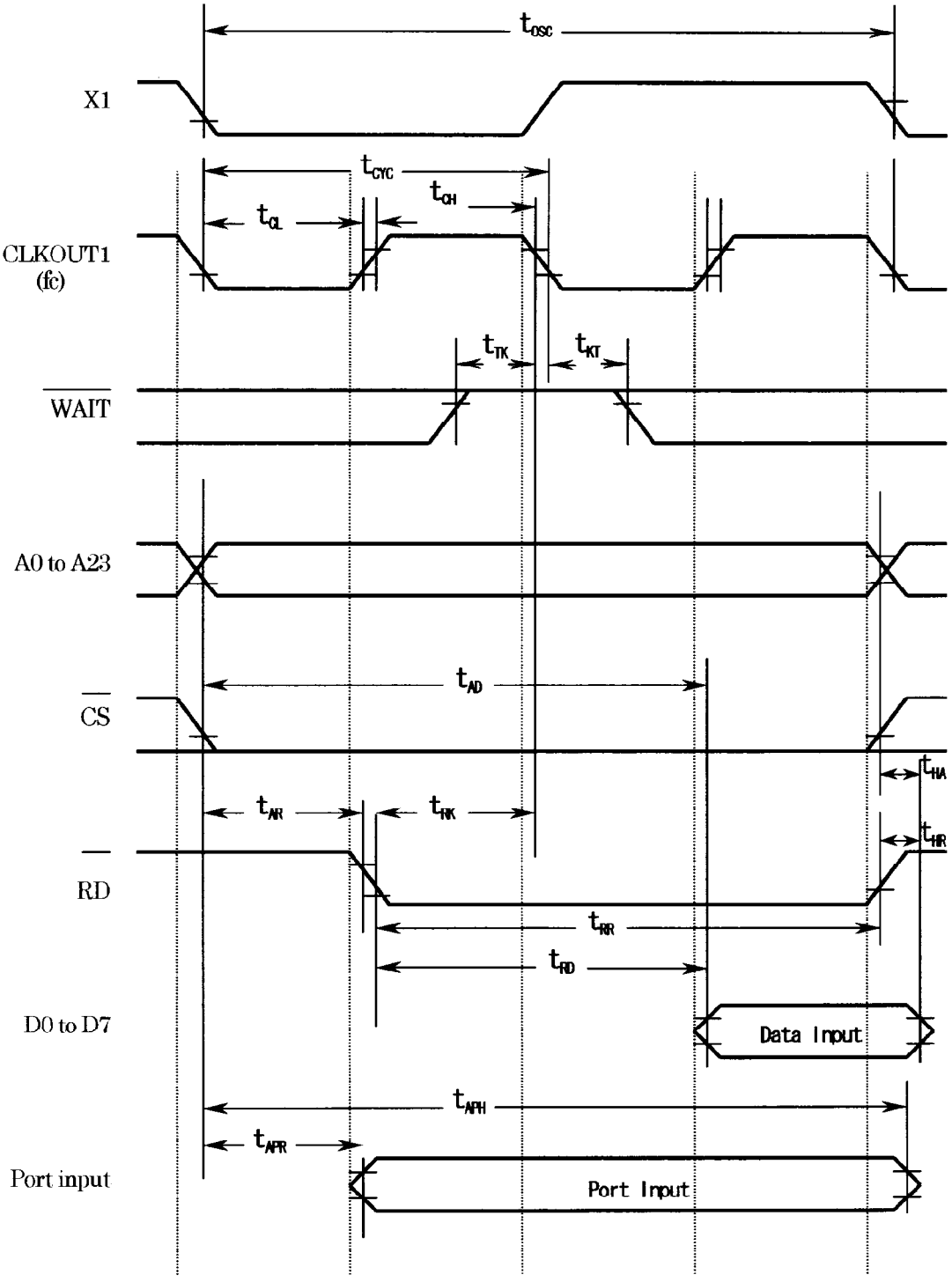
No.	Parameter	Symbol	Min	Max	@20MHz	@16MHz	Unit
1	OSC period (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System Clock period (=T)	t_{CYC}	50	62.5	50	62.5	ns
3	CLKOUT1 Low Width	t_{CL}	0.5×T-15		10	16	ns
4	CLKOUT1 High Width	t_{CH}	0.5×T-15		10	16	ns
5-1	D0 to D7 Valid → $\overline{WR}$ Rise @0WAIT	t_{DW}	1.25×T-35		28	43	ns
5-2	D0 to D7 Valid → $\overline{WR}$ Rise @1WAIT	t_{DWW}	2.25×T-35		78	106	ns
6-1	$\overline{WR}$ Low Width @0WAIT	t_{WW}	1.25×T-30		33	48	ns
6-2	$\overline{WR}$ Low Width @1WAIT	t_{WWW}	2.25×T-30		83	111	ns
7	A0 to A23 Valid → $\overline{WR}$ Fall	t_{AW}	0.5×T-20		5	11	ns
8	$\overline{WR}$ Fall → CLK Fall	t_{WK}	0.5×T-20		5	11	ns
9	$\overline{WR}$ Fall → A0 to A23 Hold	t_{WA}	0.25×T-5		8	11	ns
10	$\overline{WR}$ Fall → D0 to D7 Hold	t_{WD}	0.25×T-5		8	11	ns
11	A0 to A23 Valid → PORT Output	t_{APW}		2.0×T+70	170	195	ns
12	$\overline{WAIT}$ Set-up Time	t_{YS}	15		15	15	ns
13	$\overline{WAIT}$ Hold Time	t_{WT}	5		5	5	ns
14	$\overline{RD}$ Rise → D0 to D7 Output	t_{RDO}	1.25×T-35		20	26	ns

AC Condition

●Output : P0(D0 to D7), P4(A0 to A7), PD(A8 to A15), PM(A16 to A23), P70(RD), P71(WR)
High 2.0V, Low 0.8V, CL=50pF
Others
High 2.0V, Low 0.8V, CL=50pF

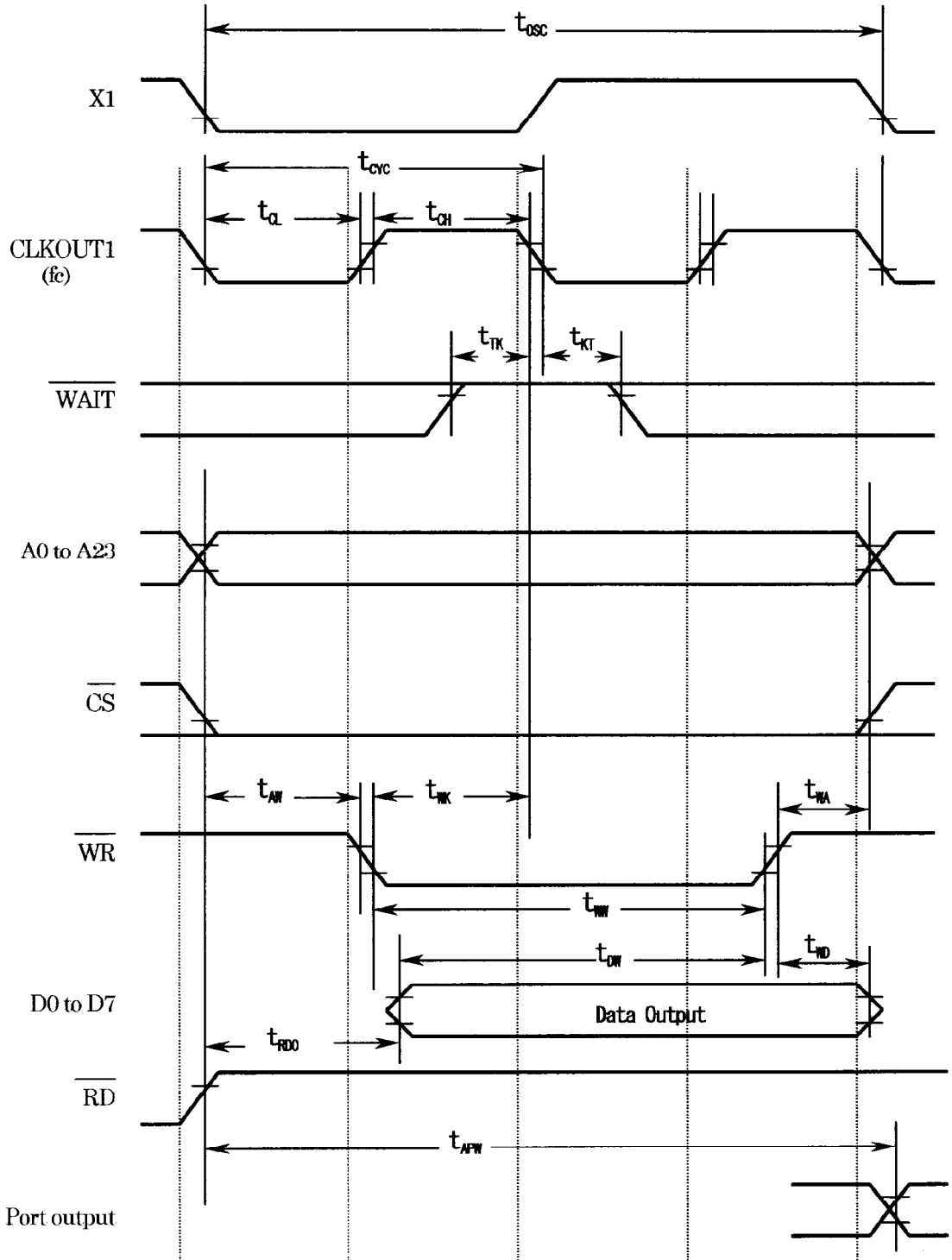
●Input : P0(D0 to D7)
High 2.4V, Low 0.45V, CL=50pF
Others
High 0.8×VCC5, Low 0.2×VCC5

(1) Read cycle (0 wait)



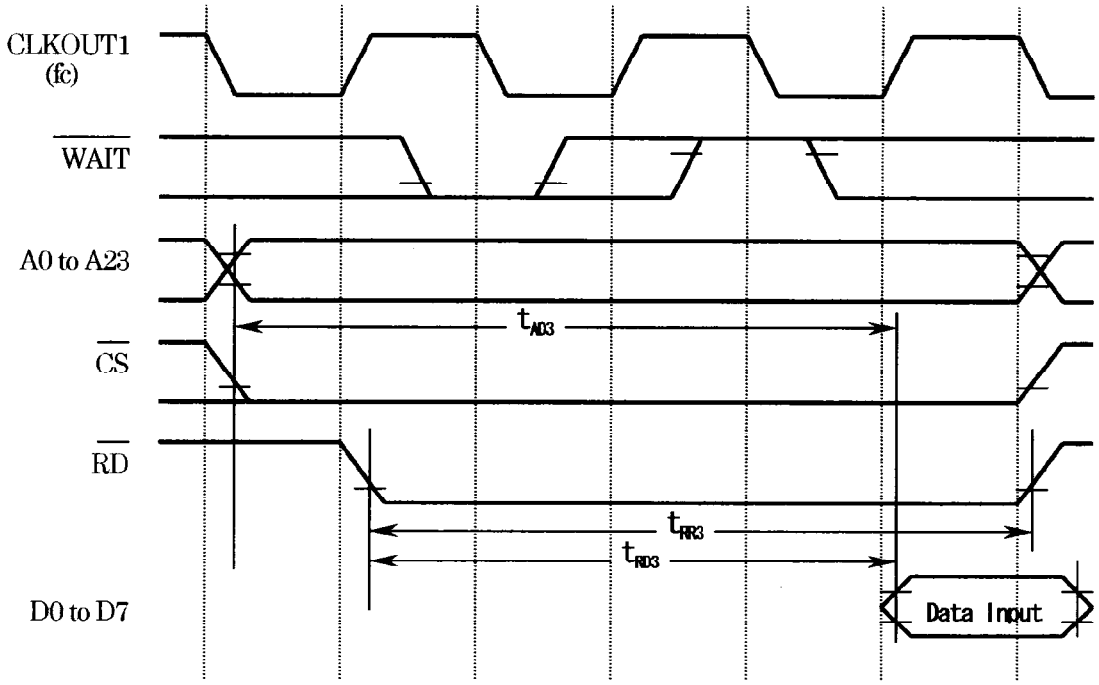
Note : The phase relation between X1 input signal and the other signals is unsettled.
The timing chart above is an example.

(2) Write cycle (0 wait)

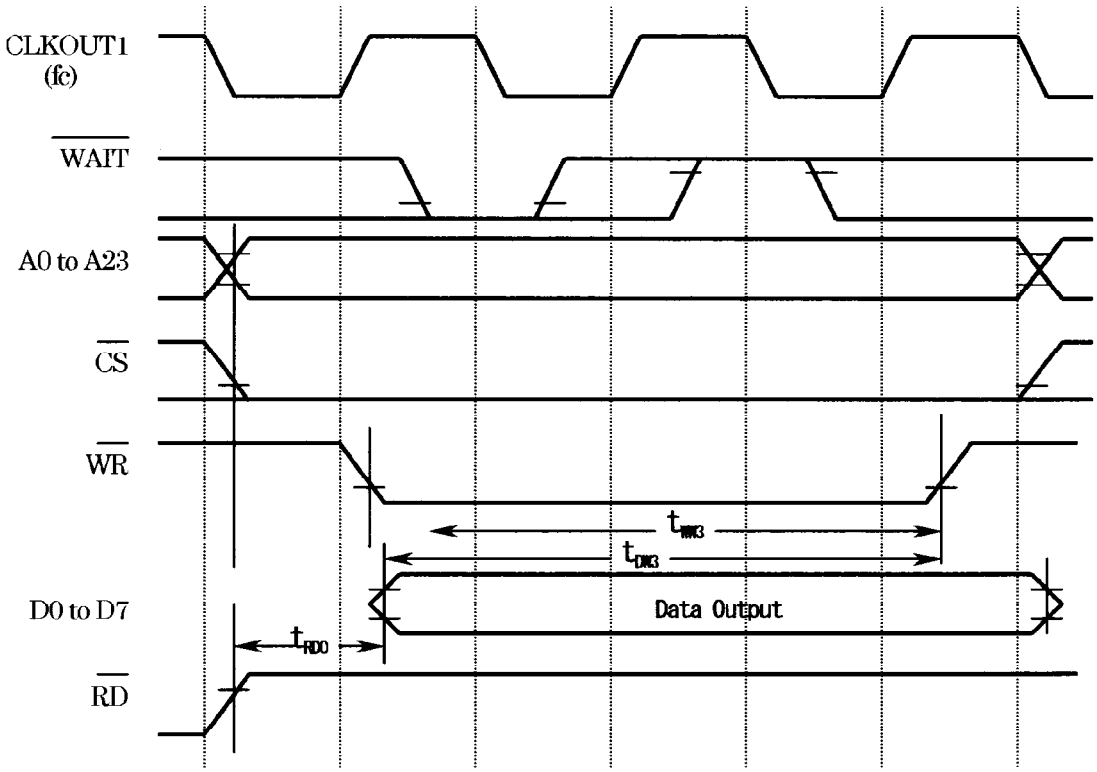


Note : The phase relation between X1 input signal and the other signals is unsettled. The timing chart above is an example.

(3) Read cycle (1 wait)



(4) Write cycle (1 wait)



4.4 AD Conversion Characteristics

Symbol	Parameter	Min	Typ	MAX	Unit
VREFH	Analog reference voltage (+)	VCC5-0.2	VCC5	VCC5	V
VREFL	Analog reference voltage (−)	VSS5	VSS5	VSS5	
AVCC	AD Converter Power Supply Voltage	VCC5-0.2	VCC5	VCC5	
AVSS	AD Converter Ground	VSS5	VSS5	VSS5	
AVIN	Analog Input Voltage	VREFL		VREFH	
IREF	Analog Current for analog reference voltage		0.8	1.2	mA
E_T	Total error (Quantize error of ±0.5LSB is included)			±3.0	LSB

4.5 Event Counter (TI0, TI4, TI8, TI9, TIA, TIB)

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock Cycle	T_{VCK}	8T+100		500		600		ns
Clock Low Width	T_{VCKL}	4T+40		240		290		ns
Clock High Width	T_{VCKH}	4T+40		240		290		ns

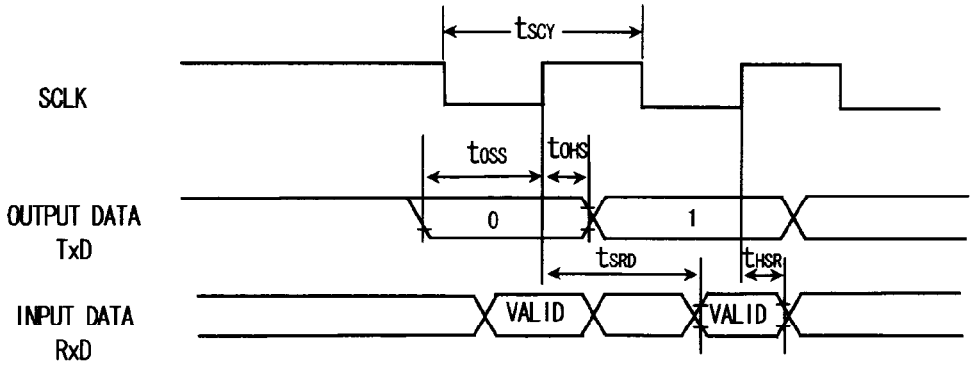
4.6 Serial Channel Timing

(1) SCLK Input mode (I/O Interface mode)

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Cycle	T_{SCY}	16T		0.8		1.0		μs
Output Data → SCLK Rise	T_{OSS}	T_{SCY}/2-50		350		450		ns
SCLK Rise → Output Data Hold	T_{OHS}	T_{SCY}/2-100		300		400		
SCLK Rise → Input Data Hold	T_{HSR}	0		0		0		
SCLK Rise → Input Data Valid	T_{SRD}		T_{SCY}/2-100		300		400	

(2) SCLK Output mode (I/O Interface mode)

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Cycle (programmable)	T_{SCY}	16T	8192T	0.8	409.6	1.0	512	μs
Output Data → SCLK Rise	T_{OSS}	T_{SCY}/2-150		250		350		ns
SCLK Rise → Output Data Hold	T_{OHS}	T_{SCY}/2-80		320		420		
SCLK Rise → Input Data Hold	T_{HSR}	0		0		0		
SCLK Rise → Input Data Valid	T_{SRD}		T_{SCY}/2-150		250		350	



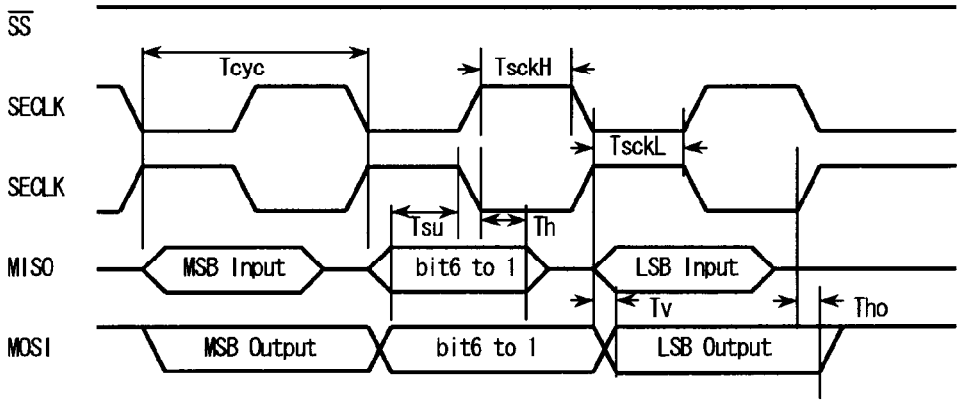
4.7 Interrupt Operation

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, INT0 Low Width	T_{INTAL}	4T		200		250		ns
$\overline{NMI}$, INT0 High Width	T_{INTAH}	4T		200		250		
INT1~INT7 Low Width	T_{INTBL}	8T+100		500		600		
INT1~INT7 High Width	T_{INTBH}	8T+100		500		600		

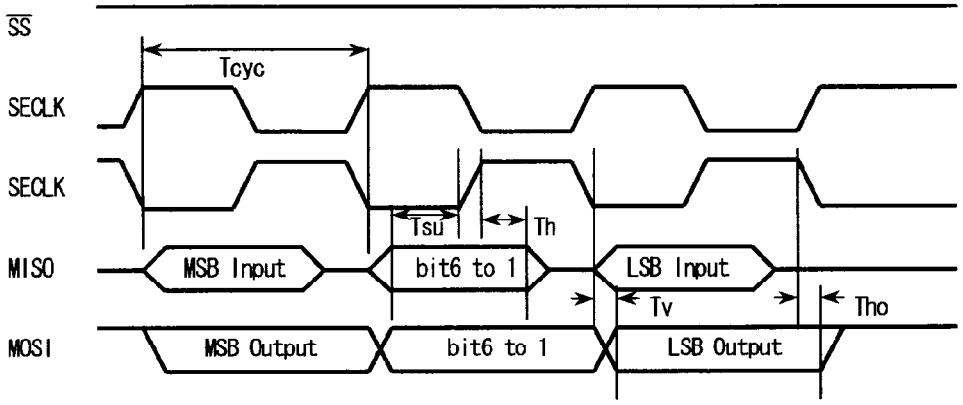
4.8 Serial Expansion Interface

Symbol	Parameter	Variable		20MHz		Unit
		Min	Max	Min	Max	
Tcyc	SECLK Cycle	2.5T	4T	125		ns
Tlead	SS fall → SECLK	2T		100		ns
Tlag	SECLK → SS rise	2T		100		ns
TsckH	SECLK High Pulse Width	Tcyc/2-15		48		ns
TsckL	SECLK Low Pulse Width	Tcyc/2-15		48		ns
Tsu	Input Data Set-up	Tcyc/4		31		ns
Th	Input Data Hold	Tcyc/4		31		ns
Tv	Output Data Va		Tcyc/4		31	ns
Tho	Output Data H	0		0		ns

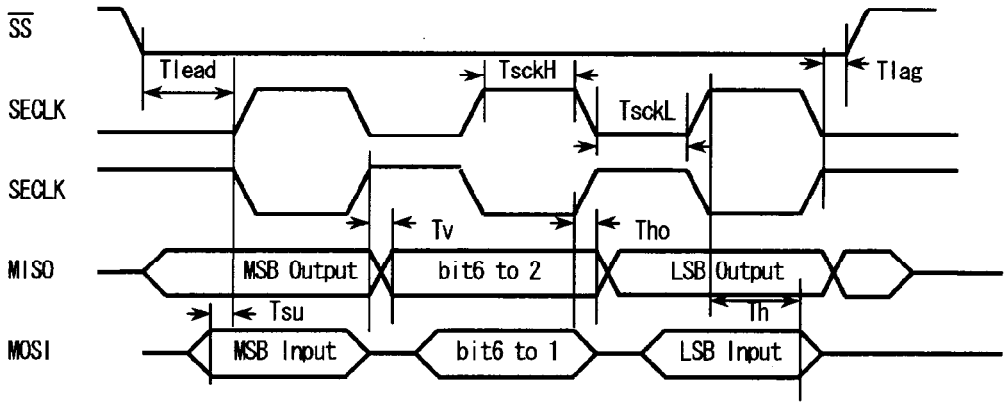
a) SEI Master (CPHA=0)



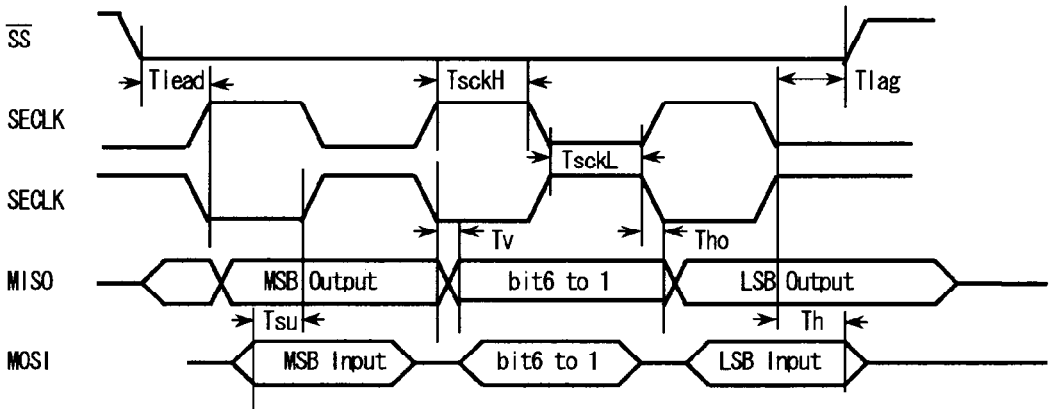
b) SEI Master (CPHA=1)



c) SEI Slave (CPHA=0)



d) SEI Slave (CPHA=1)



5. Table of special function registers (SFRs)
(SFR : Special Function Register)

The special function registers (SFRs) include the I/O ports and peripheradontrol registers allocated to the 1024 byte addresses from 000000H to 0003FFH.

(1) I/O port
(2) 8-bit Timer control
(3) 16-bit Timer control
(4) Serial Channel control
(5) Serial Expansion Interface control
(6) Interrupt control
(7) DMA controller
(8) Control register
(9) A/D converter control
(10) Memory controller
(11) Serial Bus Interface control

Configuration of the table

Symbol	Name	Address	7	6	5	4	3	2	1	0	
											→ bit Symbol
											→ Read/Write
											→ Initial value after reset
											→ Remarks

Explanations of symbols

R/W : Either read or write is possib
R : Only read is possible
W : Only write is possible

no RMW : Prohibit Read Modify Write
(Prohibit RES / SET / TSET / CHG / STCF / ANDCF / ORCF / XORCF etc.)

Table 5 I/O register address map

[1] Port:: 900/H1 type I/O

ADDRESS	NAME
0000H	P0
1H	(Reserved)
2H	P0CR
3H	P0FC
4H	(Reserved)
5H	(Reserved)
6H	(Reserved)
7H	(Reserved)
8H	(Reserved)
9H	(Reserved)
AH	(Reserved)
BH	(Reserved)
CH	(Reserved)
DH	(Reserved)
EH	(Reserved)
FH	(Reserved)

ADDRESS	NAME
0010H	P4
11H	(Reserved)
12H	P4CR
13H	P4FC
14H	(Reserved)
15H	(Reserved)
16H	(Reserved)
17H	(Reserved)
18H	(Reserved)
19H	(Reserved)
1AH	(Reserved)
1BH	(Reserved)
1CH	P7
1DH	(Reserved)
1EH	P7CR
1FH	P7FC

ADDRESS	NAME
0020H	(Reserved)
21H	(Reserved)
22H	(Reserved)
23H	(Reserved)
24H	(Reserved)
25H	(Reserved)
26H	(Reserved)
27H	(Reserved)
28H	(Reserved)
29H	(Reserved)
2AH	(Reserved)
2BH	(Reserved)
2CH	(Reserved)
2DH	(Reserved)
2EH	(Reserved)
2FH	(Reserved)

ADDRESS	NAME
0030H	PC
31H	(Reserved)
32H	PCCR
33H	PCFC
34H	PD
35H	(Reserved)
36H	PDCR
37H	PDFC
38H	(Reserved)
39H	(Reserved)
3AH	(Reserved)
3BH	(Reserved)
3CH	PF
3DH	(Reserved)
3EH	PFCR
3FH	PFFC

[2] SEI : 900/L1 type I/O

ADDRESS	NAME
0040H	PG
41H	(Reserved)
42H	(Reserved)
43H	(Reserved)
44H	(Reserved)
45H	(Reserved)
46H	(Reserved)
47H	(Reserved)
48H	(Reserved)
49H	(Reserved)
4AH	(Reserved)
4BH	(Reserved)
4CH	(Reserved)
4DH	(Reserved)
4EH	(Reserved)
4FH	(Reserved)

ADDRESS	NAME
0050H	(Reserved)
51H	(Reserved)
52H	(Reserved)
53H	(Reserved)
54H	PL
55H	(Reserved)
56H	(Reserved)
57H	(Reserved)
58H	PM
59H	PMODE
5AH	PMCR
5BH	PMFC
5CH	PN
5DH	PNODE
5EH	PNCR
5FH	PNFC

ADDRESS	NAME
0060H	SECR0
61H	SESR0
62H	SEDR0
63H	(Reserved)
64H	SECR1
65H	SESR1
66H	SEDR1
67H	(Reserved)
68H	(Reserved)
69H	(Reserved)
6AH	(Reserved)
6BH	(Reserved)
6CH	(Reserved)
6DH	(Reserved)
6EH	(Reserved)
6FH	(Reserved)

ADDRESS	NAME
0070H	(Reserved)
71H	(Reserved)
72H	(Reserved)
73H	(Reserved)
74H	(Reserved)
75H	(Reserved)
76H	(Reserved)
77H	(Reserved)
78H	(Reserved)
79H	(Reserved)
7AH	(Reserved)
7BH	(Reserved)
7CH	(Reserved)
7DH	(Reserved)
7EH	(Reserved)
7FH	(Reserved)

Note: Do not access the without allocated names.

[3] 8-bit Timer: 900/L1 type I/O

ADDRESS	NAME
0080H	TRUN01
81H	(Reserved)
82H	TREG0
83H	TREG1
84H	TMOD01
85H	TFFCR1
86H	(Reserved)
87H	(Reserved)
88H	TRUN23
89H	(Reserved)
8AH	TREG2
8BH	TREG3
8CH	TMOD23
8DH	TFFCR3
8EH	(Reserved)
8FH	(Reserved)

ADDRESS	NAME
0090H	TRUN45
91H	(Reserved)
92H	TREG4
93H	TREG5
94H	TMOD45
95H	TFFCR5
96H	(Reserved)
97H	(Reserved)
98H	TRUN67
99H	(Reserved)
9AH	TREG6
9BH	TREG7
9CH	TMOD67
9DH	TFFCR7
9EH	(Reserved)
9FH	(Reserved)

[4] 16-bit Timer: 900/L1 type I/O

ADDRESS	NAME
00A0H	TRUN8
A1H	(Reserved)
A2H	TMOD8
A3H	TFFCR8
A4H	(Reserved)
A5H	(Reserved)
A6H	(Reserved)
A7H	(Reserved)
A8H	TREG8L
A9H	TREG8H
AAH	TREG9L
ABH	TREG9H
ACH	CAP8L
ADH	CAP8H
AEH	CAP9L
AFH	CAP9H

ADDRESS	NAME
00B0H	TRUNA
B1H	(Reserved)
B2H	TMODA
B3H	TFFCRA
B4H	(Reserved)
B5H	(Reserved)
B6H	(Reserved)
B7H	(Reserved)
B8H	TREGAL
B9H	TREGAH
BAH	TREGBL
BBH	TREGBH
BCH	CAPAL
BDH	CAPAH
BEH	CAPBL
BFH	CAPBH

[5] SIO: 900/L1 type I/O

ADDRESS	NAME
00C0H	SC0BUF
C1H	SC0CR
C2H	SC0MOD0
C3H	BR0CR
C4H	BR0ADD
C5H	SC0MOD1
C6H	(Reserved)
C7H	(Reserved)
C8H	SC1BUF
C9H	SC1CR
CAH	SC1MOD0
CBH	BR1CR
CCH	BR1ADD
CDH	SC1MOD1
CEH	(Reserved)
CFH	(Reserved)

[6] INTC: 900/H1 type I/O

ADDRESS	NAME
00D0H	INTE12
D1H	INTE34
D2H	INTE56
D3H	INTE7
D4H	INTET01
D5H	INTET23
D6H	INTET45
D7H	INTET67
D8H	INTET89
D9H	INTETAB
DAH	INTETO8A
DBH	INTES0
DCH	INTES1
DDH	(Reserved)
DEH	(Reserved)
DFH	INTESEE0

ADDRESS	NAME
00E0H	INTESED0
E1H	INTESEE1
E2H	INTESED1
E3H	INTESB0
E4H	INTESB1
E5H	(Reserved)
E6H	(Reserved)
E7H	(Reserved)
E8H	(Reserved)
E9H	(Reserved)
EAH	(Reserved)
EBH	(Reserved)
ECH	(Reserved)
EDH	(Reserved)
EEH	(Reserved)
EFH	(Reserved)

ADDRESS	NAME
00F0H	INTE0AD
F1H	INTETC01
F2H	INTETC23
F3H	INTETC45
F4H	INTETC67
F5H	(Reserved)
F6H	IIMC
F7H	INTNMWDT
F8H	INTCLR
F9H	(Reserved)
FAH	(Reserved)
FBH	(Reserved)
FCH	(Reserved)
FDH	(Reserved)
FEH	(Reserved)
FFH	(Reserved)

ADDRESS	NAME
0100H	DMA0V
101H	DMA1V
102H	DMA2V
103H	DMA3V
104H	DMA4V
105H	DMA5V
106H	DMA6V
107H	DMA7V
108H	DMAB
109H	DMAR
10AH	CLKMOD
10BH	(Reserved)
10CH	(Reserved)
10DH	(Reserved)
10EH	(Reserved)
10FH	(Reserved)

[7] WDT: 900/L1 type I/O

ADDRESS	NAME
0110H	WDMOD
111H	WDCR
112H	(Reserved)
113H	(Reserved)
114H	(Reserved)
115H	(Reserved)
116H	(Reserved)
117H	(Reserved)
118H	(Reserved)
119H	(Reserved)
11AH	(Reserved)
11BH	(Reserved)
11CH	(Reserved)
11DH	(Reserved)
11EH	(Reserved)
11FH	(Reserved)

[8] 10-bit ADC: 900/L1 type I/O

ADDRESS	NAME
0120H	ADREG0L
121H	ADREG0H
122H	ADREG1L
123H	ADREG1H
124H	ADREG2L
125H	ADREG2H
126H	ADREG3L
127H	ADREG3H
128H	ADREG4L
129H	ADREG4H
12AH	ADREG5L
12BH	ADREG5H
12CH	ADREG6L
12DH	ADREG6H
12EH	ADREG7L
12FH	ADREG7H

ADDRESS	NAME
0130H	ADREG8L
131H	ADREG8H
132H	ADREG9L
133H	ADREG9H
134H	ADREGAL
135H	ADREGAH
136H	ADREGBL
137H	ADREGBH
138H	ADMOD0
139H	ADMOD1
13AH	(Reserved)
13BH	(Reserved)
13CH	(Reserved)
13DH	(Reserved)
13EH	(Reserved)
13FH	(Reserved)

[9]MEMC: 900/H1 type I/O

ADDRESS	NAME
0140H	(Reserved)
141H	(Reserved)
142H	(Reserved)
143H	(Reserved)
144H	(Reserved)
145H	(Reserved)
146H	(Reserved)
147H	(Reserved)
148H	BCSL
149H	BCSH
14AH	MAMR
14BH	MSAR
14CH	(Reserved)
14DH	(Reserved)
14EH	(Reserved)
14FH	(Reserved)

ADDRESS	NAME
0150H	(Reserved)
151H	(Reserved)
152H	(Reserved)
153H	(Reserved)
154H	(Reserved)
155H	(Reserved)
156H	(Reserved)
157H	(Reserved)
158H	(Reserved)
159H	(Reserved)
15AH	(Reserved)
15BH	(Reserved)
15CH	(Reserved)
15DH	(Reserved)
15EH	(Reserved)
15FH	(Reserved)

[10] SBI:900/L1 type I/O

ADDRESS	NAME
0160H	(Reserved)
161H	(Reserved)
162H	(Reserved)
163H	(Reserved)
164H	(Reserved)
165H	(Reserved)
166H	(Reserved)
167H	(Reserved)
168H	(Reserved)
169H	(Reserved)
16AH	(Reserved)
16BH	(Reserved)
16CH	(Reserved)
16DH	RAMCR
16EH	(Reserved)
16FH	(Reserved)

ADDRESS	NAME
0170H	SBI0CR1
171H	SBI0DBR
172H	I2C0AR
173H	SBI0CR2/SBI0SR
174H	SBI0BR0
175H	SBI0BR1
176H	(Reserved)
177H	(Reserved)
178H	SBI1CR1
179H	SBI1DBR
17AH	I2C1AR
17BH	SBI1CR2/SBI1SR
17CH	SBI1BR0
17DH	SBI1BR1
17EH	(Reserved)
17FH	(Reserved)

[

ADDRESS	NAME
0200H : : : : : 3FFH	(Reserved)

(1) I/O Port

Port0

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
P0	PORT0 Register	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P0CR	PORT0 Control Register	02H (no RMW)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
P0FC	PORT0 Function Register	03H (no RMW)	-	-	-	-	-	-	-	P0F
										W
			-	-	-	-	-	-	-	0
			0:PORT 1:Data Bus (D7~D0)							

Port4

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
P4	PORT4 Register	10H	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P4CR	PORT4 Control Register	12H (no RMW)	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
P4FC	PORT4 Function Register	13H (no RMW)	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
			W							
			0	0	0	0	0	0	0	0
			0:PORT 1:A7	0:PORT 1:A6	0:PORT 1:A5	0:PORT 1:A4	0:PORT 1:A3	0:PORT 1:A2	0:PORT 1:A1	0:PORT 1:A0

P4CR	P4FC	P47	P46	P45	P44	P43	P42	P41	P40
0	0	Input Port							
1	0	Output Port							
1	1	(Reserved)							
0	1	A7-0							

Port7

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
P7	PORT7 Register	1CH	-	-	P75	P74	P73	P72	P71	P70
					R/W					
			-	-	0	1	1	1	1	1
					Input/Output					
P7CR	PORT7 Control Register	1EH (no RMW)	-	-	P75C	P74C	P73C	P72C	P71C	P70C
					W					
			-	-	0	1	1	1	1	1
					0:Input 1:Output					
P7FC	PORT7 Function Register	1FH (no RMW)	-	-	P75F	P74F	P73F	P72F	P71F	P70F
					W					
			-	-	0	0	0	0	0/1	0/1
					0:PORT 1:WAIT	0:PORT 1:CLKOUT2	0:PORT 1:CS	0:PORT	0:PORT 1:WR	0:PORT 1:RD

PortC

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PC	PORTC Register	30H	-	-	PC5	PC4	PC3	PC2	PC1	PC0
					R/W					
			-	-	0	0	0	0	0	0
					Input/Output					
PCCR	PORTC Control Register	32H (no RMW)	-	-	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
					W					
			-	-	0	0	0	0	0	0
					0:Input 1:Output					
PCFC	PORTC Function Register	33H (no RMW)	-	-	PC5F	PC4F	PC3F	PC2F	PC1F	PC0F
					W					
			-	-	0	0	0	0	0	0
					0:PORT INT4 1:TO7	0:PORT 1:TO5	0:PORT INT3 1:TI4	0:PORT INT2 1:TO3	0:PORT 1:TO1	0:PORT INT1 1:TI0

PortD

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PD	PORTD Register	34H	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
PDCR	PORTD Control Register	36H (no RMW)	PD7C	PD6C	PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
PDFC	PORTD Function Register	37H (no RMW)	PD7F	PD6F	PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
			W							
			0	0	0	0	0	0	0	0
			0:PORT 1:TOB A15	0:PORT 1:TOA A14	0:PORT 1:TIB A13	0:PORT INT7 1:TIA A12	0:PORT 1:TO9 A11	0:PORT 1:TO8 A10	0:PORT INT6 1:TI9 A9	0:PORT INT5 1:TI8 A8

PDCR	PDFC	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
0	0	Input Port	Input Port	Input Port TIB	Input Port TIA INT7	Input Port	Input Port	Input Port TI9 INT6	Input Port TI8 INT5
1	0	Output Port							
1	1	TOB	TOA	TIB	TIA INT7	TO9	TO8	TI9 INT6	TI8 INT5
0	1	A15	A14	A13	A12	A11	A10	A9	A8

PortF

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PF	PORTF Register	3CH	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
PFCR	PORTF Control Register	3EH (no RMW)	PF7C	PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
PFFC	PORTF Function Register	3FH (no RMW)	PF7F	PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
			W							
			0	0	0	0	0	0	0	0
			0:PORT Note) Fix to 0.	0:PORT Note) Fix to 0.	0:PORT 1:CTS1 SCLK1	0:PORT 1:RXD1	0:PORT 1:TXD1	0:PORT 1:CTS0 SCLK0	0:PORT 1:RXD0	0:PORT 1:TXD0

PFCR	PFFC	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
0	0	Input Port	Input Port	Input Port SCLK1 CTS1	Input Port RXD1	Input Port	Input Port SCLK0 CTS0	Input Port RXD0	Input Port
1	0	Output Port							
1	1	Don't use this setting	Don't use this setting	SCLK1	RXD1	TXD1	SCLK0	RXD0	TXD0
0	1			Don't use this setting		Open Drain output	Don't use this setting		Open Drain output

PortG

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PG	PORTG Register	40H	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
			R							
			Input							

PortL

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PL	PORTL Register	54H	-	-	-	-	PL3	PL2	PL1	PL0
							R			
			-	-	-	-	Input			

PortM

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PM	PORTM Register	58H	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
PMODE	PORTM Open Drain Enable Register	59H	ODEM7	ODEM6	ODEM5	-	ODEM3	ODEM2	ODEM1	-
			R/W				R/W			
			0	0	0	-	0	0	0	-
			PM7 output 0:CMOS 1:Open Drain	PM6 output 0:CMOS 1:Open Drain	PM5 output 0:CMOS 1:Open Drain		PM3 output 0:CMOS 1:Open Drain	PM2 output 0:CMOS 1:Open Drain	PM1 output 0:CMOS 1:Open Drain	
PMCR	PORTM Control Register	5AH (no RMW)	PM7C	PM6C	PM5C	PM4C	PM3C	PM2C	PM1C	PM0C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
PMFC	PORTM Function Register	5BH (no RMW)	PM7F	PM6F	PM5F	PM4F	PM3F	PM2F	PM1F	PM0F
			W							
			0	0	0	0	0	0	0	0
			0:PORT 1:SECLK1 A23	0:PORT 1:MISO1 A22	0:PORT 1:MOSI1 A21	0:PORT 1:$\overline{SS1}$ A20	0:PORT 1:SECLK0 A19	0:PORT 1:MISO0 A18	0:PORT 1:MOSI0 A17	0:PORT 1:$\overline{SS0}$ A16

PMCR	PMFC	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
0	0	Input Port SECLK1	Input Port MISO1	Input Port MOSI1	Input Port $\overline{SS1}$	Input Port SECLK0	Input Port MISO0	Input Port MOSI0	Input Port $\overline{SS0}$
1	0	Output Port							
1	1	SECLK1	MISO1	MOSI1	$\overline{SS1}$	SECLK0	MISO0	MOSI0	$\overline{SS0}$
0	1	A23	A22	A21	A20	A19	A18	A17	A16

PortN

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PN	PORTN Register	5CH	-	-	PN5	PN4	PN3	PN2	PN1	PN0
					R/W					
			-	-	0	0	0	0	0	0
					Input/Output					
PNODE	PORTN Open Drain Enable Register	5DH	-	-	ODEN5	ODEN4	-	ODEN2	ODEN1	-
					R/W			R/W		
			-	-	0	0	-	0	0	-
					PN5 output 0:CMOS 1:Open Drain	PN4 output 0:CMOS 1:Open Drain		PN2 output 0:CMOS 1:Open Drain	PN1 output 0:CMOS 1:Open Drain	
PNCR	PORTN Control Register	5EH (no RMW)	-	-	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
					W					
			-	-	0	0	0	0	0	0
					0:Input 1:Output					
PNFC	PORTN Function Register	5FH (no RMW)	-	-	PN5F	PN4F	PN3F	PN2F	PN1F	PN0F
					W					
			-	-	0	0	0	0	0	0
					0:PORT 1:SI1 SCL1	0:PORT 1:SO1 SDA1	0:PORT 1:SCLK1	0:PORT 1:SI0 SCL0	0:PORT 1:SO0 SDA0	0:PORT 1:SCK0

PNCR	PNFC			PN5	PN4	PN3	PN2	PN1	PN0
0	0			Input Port SI1/SCL1	Input Port SDA1	Input Port SCK1	Input Port SI0/SCL0	Input Port SDA0	Input Port SCK0
1	0			Output Port	Output Port	Output Port	Output Port	Output Port	Output Port
1	1			SCL1	SO1/SDA1	SCK1	SCL0	SO0/SDA0	SCK0
0	1	(reserved)							

(2) 8-bit Timer

8-Bit Timer 01, 23, 45, 67

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
TRUN01	8bit Timer01 Run Register	80H	TORDE	-	-	-	I2T01	T01PRUN	T1RUN	TORUN
			R/W				R/W	R/W		
			0	-	-	-	0	0	0	0
			Double Buffer 0:Disable 1:Enable				IDLE2 0:Stop 1:Operate	8bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG0	8Bit Timer Register 0	82H (no RMW)	-							
			W							
			Undefined							
TREG1	8Bit Timer Register 1	83H (no RMW)	-							
			W							
			Undefined							
TMOD01	8Bit Timer0, 1 Source CLK & MODE Register	84H (no RMW)	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8bit Timer 01:16bit Timer 10:8bit PPG 11:8bit PWM		PWM cycle 00:reserved 01:2^6-1 10:2^7-1 11:2^8-1		Timer1 source clock 00:T0TRG 01:ϕT1 10:ϕT16 11:ϕT256		Timer0 source clock 00:T10 01:ϕT1 10:ϕT4 11:ϕT16	
TFFCR1	Timer1 Flip-Flop Control Register	85H (no RMW)	-	-	-	-	TFF1C1	TFF1C0	TFF1IE	TFF1IS
							R/W		R/W	
			-	-	-	-	1	1	0	0
							00:Invert TFF1 01:Set TFF1 10:Clear TFF1 11:Don't care		TFF1 Invert 0:Disable 1:Enable	TFF1 Invert 0:Timer0 1:Timer1
TRUN23	8bit Timer23 Run Register	88H	T2RDE	-	-	-	I2T23	T23PRUN	T3RUN	T2RUN
			R/W				R/W	R/W		
			0	-	-	-	0	0	0	0
			Double Buffer 0:Disable 1:Enable				IDLE2 0:Stop 1:Operate	8bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG2	8Bit Timer Register 2	8AH (no RMW)	-							
			W							
			Undefined							
TREG3	8Bit Timer Register 3	8BH (no RMW)	-							
			W							
			Undefined							
TMOD23	8Bit Timer2, 3 Source CLK & MODE Register	8CH (no RMW)	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8bit Timer 01:16bit Timer 10:8bit PPG 11:8bit PWM		PWM cycle 00:reserved 01:2^6-1 10:2^7-1 11:2^8-1		Timer3 source clock 00:T2TRG 01:ϕT1 10:ϕT16 11:ϕT256		Timer2 source clock 00:reserved 01:ϕT1 10:ϕT4 11:ϕT16	
TFFCR3	Timer3 Flip-Flop Control Register	8DH (no RMW)	-	-	-	-	TFF3C1	TFF3C0	TFF3IE	TFF3IS
							R/W		R/W	
			-	-	-	-	1	1	0	0
							00:Invert TFF3 01:Set TFF3 10:Clear TFF3 11:Don't Care		TFF3 Invert 0:Disable 1:Enable	TFF3 Invert 0:Timer2 1:Timer3

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
TRUN45	8bit Timer45 Run Register	90H	T4RDE	-	-	-	I2T45	T45PRUN	T5RUN	T4RUN
			R/W				R/W	R/W		
			0	-	-	-	0	0	0	0
			Double Buffer 0:Disable 1:Enable				IDLE2 0:Stop 1:Operate	8bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG4	8Bit Timer Register 4	92H (no RMW)	-							
			W							
			Undefined							
TREG5	8Bit Timer Register 5	93H (no RMW)	-							
			W							
			Undefined							
TMOD45	8Bit Timer4.5 Source CLK & MODE Register	94H (no RMW)	T45M1	T45M0	PWM41	PWM40	T5CLK1	T5CLK0	T4CLK1	T4CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8bit Timer 01:16bit Timer 10:8bit PPG 11:8bit PWM		PWM cycle 00:reserved 01:2^6-1 10:2^7-1 11:2^8-1		Timer5 source clock 00:T4TRG 01:ϕT1 10:ϕT16 11:ϕT256		Timer4 source clock 00:TI4 01:ϕT1 10:ϕT4 11:ϕT16	
TFFCR5	Timer5 Flip-Flop Control Register	95H (no RMW)	-	-	-	-	TFF5C1	TFF5C0	TFF5IE	TFF5IS
							R/W		R/W	
			-	-	-	-	1	1	0	0
							00:Invert TFF5 01:Set TFF5 10:Clear TFF5 11:Don't care		TFF5 Invert 0:Disable 1:Enable	TFF5 Invert 0:Timer4 1:Timer5
TRUN67	8bit Timer67 Run Register	98H	T6RDE	-	-	-	I2T67	T67PRUN	T7RUN	T6RUN
			R/W				R/W	R/W		
			0	-	-	-	0	0	0	0
			Double Buffer 0:Disable 1:Enable				IDLE2 0:Stop 1:Operate	8bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG6	8Bit Timer Register 6	9AH (no RMW)	-							
			W							
			Undefined							
TREG7	8Bit Timer Register 7	9BH (no RMW)	-							
			W							
			Undefined							
TMOD67	8Bit Timer6.7 Source CLK & MODE Register	9CH (no RMW)	T67M1	T67M0	PWM61	PWM60	T7CLK1	T7CLK0	T6CLK1	T6CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8bit Timer 01:16bit Timer 10:8bit PPG 11:8bit PWM		PWM cycle 00:reserved 01:2^6-1 10:2^7-1 11:2^8-1		Timer7 source clock 00:T6TRG 01:ϕT1 10:ϕT16 11:ϕT256		Timer6 source clock 00:reserved 01:ϕT1 10:ϕT4 11:ϕT16	
TFFCR7	Timer7 Flip-Flop Control Register	9DH (no RMW)	-	-	-	-	TFF7C1	TFF7C0	TFF7IE	TFF7IS
							R/W		R/W	
			-	-	-	-	1	1	0	0
							00:Invert TFF7 01:Set TFF7 10:Clear TFF7 11:Don't Care		TFF7 Invert 0:Disable 1:Enable	TFF7 Invert 0:Timer6 1:Timer7

(3) 16-bit Timer

16-Bit Timer 8, A

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
TRUN8	16bit Timer8 Run Register	A0H	T8RDE	-	-	-	I2T8	T8PRUN	-	T8RUN
			R/W				R/W	R/W		R/W
			0	0	-	-	0	0	-	0
			Double Buffer 0:Disable 1:Enable	Fix to "0"			IDLE2 0:Stop 1:Operate	16bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TMOD8	16bit Timer8 Source CLK & Mode Register	A2H	CAP9T9	EQ9T9	CAP8IN	CAP89M1	CAP89M0	T8CLE	T8CLK1	T8CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TFF9 invert trigger 0: Disable 1: Enable		0:Soft Capture 1:Don't care	Capture Timing 00:disable 01:TI81↑ TI91↑	10:TI81↑ TI8↓ 11:TFF1↑ TFF1↓ 1:UC8 Clear Enable		Source Clock 00:TI8 01:φT1	10:φT4 11:φT16
TFFCR8	16Bit Timer8 Flip-Flop Control Register	A3H	TFF9C1	TFF9C0	CAP9T8	CAP8T8	EQ9T8	EQ8T8	TFF8C1	TFF8C0
			W		R/W				W	
			1	1	0	0	0	0	1	1
			00:Invert TFF9 01:Set TFF9 10:Clear TFF9 11:Don't Care		TFF8 invert trigger 0: Disable 1: Enable				00:Invert TFF8 01:Set TFF8 10:Clear TFF8 11:Don't Care	
TREG8L	16Bit Timer Register 8 Low	A8H (no RMW)	-							
			W							
			Undefined							
TREG8H	16Bit Timer Register 8 High	A9H (no RMW)	-							
			W							
			Undefined							
TREG9L	16Bit Timer Register 9 Low	AAH (no RMW)	-							
			W							
			Undefined							
TREG9H	16Bit Timer Register 9 High	ABH (no RMW)	-							
			W							
			Undefined							
CAP8L	Capture Register 8 Low	ACH	-							
			R							
			Undefined							
CAP8H	Capture Register 8] High	ADH	-							
			R							
			Undefined							
CAP9L	Capture Register 9 Low	AEH	-							
			R							
			Undefined							
CAP9H	Capture Register 9 High	AFH	-							
			R							
			Undefined							

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
TRUNA	16bit TimerA Run Register	B0H	TARDE	-	-	-	I2TA	TAPRUN	-	TARUN
			R/W				R/W	R/W		R/W
			0	0	-	-	0	0	-	0
			Double Buffer 0:Disable 1:Enable	Fix to "0"			IDLE2 0:Stop 1:Operate	16bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TMODA	16bit TimerA Source CLK & Mode Register	B2H	CAPBTB	EQBTB	CAPAIN	CAPABM1	CAPABM0	TACLE	TACLK1	TACLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TFFB invert trigger 0: Disable 1: Enable		0:Soft Capture 1:Don't care	Capture Timing 00:disable 01:TIA↑ TIB↑ 10:TIA↑ TIA↓ 11:TFF1↑ TFF1↓		1:UCA Clear Enable	Source Clock 00:TIA 01:φT1 10:φT4 11:φT16	
TFFCRA	16Bit TimerA Flip-Flop Control Register	B3H	TFFBC1	TFFBC0	CAPBTA	CAPATA	EQBTA	EQATA	TFFAC1	TFFAC0
			W		R/W				W	
			1	1	0	0	0	0	1	1
			00:Invert TFFB 01:Set TFFB 10:Clear TFFB 11:Don't Care		TFFA invert trigger 0: Disable 1: Enable				00:Invert TFFA 01:Set TFFA 10:Clear TFFA 11:Don't Care	
TREGAL	16Bit Timer Register A Low	B8H (no RMW)	-							
			W							
			Undefined							
TREGAH	16Bit Timer Register A High	B9H (no RMW)	-							
			W							
			Undefined							
TREGBL	16Bit Timer Register B Low	BAH (no RMW)	-							
			W							
			Undefined							
TREGBH	16Bit Timer Register B High	BBH (no RMW)	-							
			W							
			Undefined							
CAPAL	Capture Register A Low	BCH	-							
			R							
			Undefined							
CAPAH	Capture Register A High	BDH	-							
			R							
			Undefined							
CAPBL	Capture Register B Low	BEH								
			R							
			Undefined							
CAPBH	Capture Register B High	BFH	-							
			R							
			Undefined							

(4) Serial Channels

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer Register	C0H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RB0 TB0
			R(Receiving) / W(Transmission)							
			Undefined							
SC0CR	Serial Channel 0 Control Register	C1H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Clear 0 after reading)			R/W	
			0	0	0	0	0	0	0	0
			Receive data bit 8	Parity 0:Odd 1:Even	Parity 0:Disable 1:Enable	1:Error Overrun	Parity	Framing	0:SCLK0 ↑ 1:SCLK0 ↓	0: Baud Rate Generator 1:SCLK0 Pin Input
SC0MOD0	Serial Channel 0 Mode 0 Register	C2H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Trans-mission data bit 8	0:CTS Disable 1:CTS Enable	0:Receive Disable 1:Receive Enable	Wake up 0:Disable 1:Enable	00:I/O Interface Mode 01:7bit UART Mode 10:8bit UART Mode 11:9bit UART Mode		00:TimerT0TRG 01:Baud Rate Generator 10:Internal clock ϕ 1 11:External clock (SCLK0 Input)	
BR0CR	Serial Channel 0 Baud Rate Control Register	C3H	-	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
				R/W						
			0	0	0	0	0	0	0	0
			Fix to "0"	(16-K)/16 divided 0:Disable 1:Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set the frequency divisor "N" 0 to F			
BR0ADD	Serial Channel 0 K setting Register	C4H	-	-	-	-	BR0K3	BR0K2	BR0K1	BR0K0
							R/W			
			-	-	-	-	0	0	0	0
							Set the frequency divisor "K" (1 to F)			
SC0MOD1	Serial Channel 0 Mode 1 Register	C5H	I2S0	FDPX0	-	-	-	-	-	-
			R/W	R/W						
			0	0	-	-	-	-	-	-
			IDLE2 0:Stop 1:Operate	I/O interface mode 1:Full duplex 0:Half duplex						

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
SC1BUF	Serial Channel 1 Buffer Register	C8H	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RB0 TB0
			R(Receiving) / W(Transmission)							
			Undefined							
SC1CR	Serial Channel 1 Control Register	C9H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Clear 0 after reading)			R/W	
			0	0	0	0	0	0	0	0
			Receive data bit 8	Parity 0:Odd 1:Even	Parity 0:Disable 1:Enable	1:Error Overrun	Parity	Framing	0:SCLK1 ↑ 1:SCLK1 ↓	0: Baud Rate Generator 1:SCLK1 Pin Input
SC1MOD0	Serial Channel 1 Mode 0 Register	CAH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Trans-mission data bit 8	0:CTS Disable 1:CTS Enable	0:Receive Disable 1:Receive Enable	Wake up 0:Disable 1:Enable	00:I/O Interface Mode 01:7bit UART Mode 10:8bit UART Mode 11:9bit UART Mode		00:TimerTOTRG 01:Baud Rate Generator 10:Internal clock ϕ1 11:External clock (SCLK1 Input)	
BR1CR	Serial Channel 1 Baud Rate Control Register	CBH	-	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
				R/W						
			0	0	0	0	0	0	0	0
			Fix to "0"	(16-K)/16 divided 0:Disable 1:Enable	00:ϕT0 01:ϕT2 10:ϕT8 11:ϕT32		Set the frequency divisor "N" 0 to F			
BR1ADD	Serial Channel 1 K setting Register	CCH	-	-	-	-	BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
			-	-	-	-	0	0	0	0
							Set the frequency divisor "K" (1 to F)			
SC1MOD1	Serial Channel 1 Mode 1 Register	CDH	I2S1	FDPX1	-	-	-	-	-	-
			R/W	R/W						
			0	0	-	-	-	-	-	-
			IDLE2 0:Stop 1:Operate	I/O interface mode 1:Full duplex 0:Half duplex						

(5) Serial Expansion Interface (SEI)

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
SECR0	SEI Control Register 0	60H	MODE	SEE	BOS	MSTR	CPOL	CPHA	SER1	SER0
			W	R/W						
			0	0	0	0	0	1	0	0
			SEI0 MODF Detection 0:Enable 1:Disable	SEI System Enable 0:Stop 1:Run	Bit Order Select bit 0:MSB 1:LSB	Master Select bit 0:Slave 1:Master	Clock Polarity Select bit 0:"H" level 1:"L" level	Clock Phase Select bit	SEI Transfer Rate Select 00:divided by 2 01:divided by 4 10:divided by 8 11:divided by 32	
SESR0	SEI Status Register 0	61H	SEF	WCOL	SOVF	MODF	-	-	-	TMSE
			R							R/W
			0	0	0	0	-	-	-	0
			SEI Transfer 0:busy or Stop 1:End	WCOL Flag 1:Error	SOVF Flag (Slave) 1:Error	MODF Flag (Master) 1:Error				SEI Mode Select 0:Compatibility Mode 1:Micro DMA Mode
			-	WCOL	SOVF	MODF	TSRC	TSTC	TASM	TMSE
				R					R/W	
			-	0	0	0	0	0	0	0
				WCOL Flag 1:Error	SOVF Flag (Slave) 1:Error	MODF Flag (Master) 1:Error	SEI Receive 1:End	SEI Transfer 1:End	Auto Shift Enable (Master) INTSEE0 Mask (Slave)	SEI Mode Select 0:Compatibility Mode 1:Micro DMA Mode
SEDR0	SEI Data Register 0	62H	SED7	SED6	SED5	SED4	SED3	SED2	SED1	SED0
			R/W							
			0	0	0	0	0	0	0	0
			Transfer/Receive Data							
SECR1	SEI Control Register 1	64H	MODE	SEE	BOS	MSTR	CPOL	CPHA	SER1	SER0
			W	R/W						
			0	0	0	0	0	1	0	0
			SEI1 MODF Detection 0:Enable 1:Disable	SEI System Enable 0:Stop 1:Run	Bit Order Select bit 0:MSB 1:LSB	Master Select bit 0:Slave 1:Master	Clock Polarity Select bit 0:"H" level 1:"L" level	Clock Phase Select bit	SEI Transfer Rate Select 00:divided by 2 01:divided by 4 10:divided by 8 11:divided by 32	
SESR1	SEI Status Register 1	65H	SEF	WCOL	SOVF	MODF	-	-	-	TMSE
			R							R/W
			0	0	0	0	-	-	-	0
			SEI Transfer 0:busy or Stop 1:End	WCOL Flag 1:Error	SOVF Flag (Slave) 1:Error	MODF Flag (Master) 1:Error				SEI Mode Select 0:Compatibility Mode 1:Micro DMA Mode
			-	WCOL	SOVF	MODF	TSRC	TSTC	TASM	TMSE
				R					R/W	
			-	0	0	0	0	0	0	0
				WCOL Flag 1:Error	SOVF Flag (Slave) 1:Error	MODF Flag (Master) 1:Error	SEI Receive 1:End	SEI Transfer 1:End	Auto Shift Enable (Master) INTSEE1 Mask (Slave)	SEI Mode Select 0:Compatibility Mode 1:Micro DMA Mode
SEDR1	SEI Data Register 1	66H	SED7	SED6	SED5	SED4	SED3	SED2	SED1	SED0
			R/W							
			0	0	0	0	0	0	0	0
			Transfer/Receive Data							

(6) Interrupt controller

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
INTEOAD	INTO & INTAD Enable Register	FOh	INTAD				INTO			
			IADC	IADM2	IADM1	IADMO	IOC	IOM2	IOM1	IOMO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE12	INT1 & INT2 Enable Register	D0h	INT2				INT1			
			I2C	I2M2	I2M1	I2MO	I1C	I1M2	I1M1	I1MO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE34	INT3 & INT4 Enable Register	D1h	INT4				INT3			
			I4C	I4M2	I4M1	I4MO	I3C	I3M2	I3M1	I3MO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE56	INT5 & INT6 Enable Register	D2h	INT6(CAP9)				INT5(CAP8)			
			I6C	I6M2	I6M1	I6MO	I5C	I5M2	I5M1	I5MO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE7	INT7 Enable Register	D3h					INT7(CAPA)			
			-	-	-	-	I7C	I7M2	I7M1	I7MO
							R	R/W		
			-	-	-	-	0	0	0	0
INTET01	INTTO & INTT1 Enable Register	D4h	INTT1(Timer1)				INTTO(TimerO)			
			IT1C	IT1M2	IT1M1	IT1MO	ITOC	ITOM2	ITOM1	ITOMO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET23	INTT2 & INTT3 Enable Register	D5h	INTT3(Timer3)				INTT2(Timer2)			
			IT3C	IT3M2	IT3M1	IT3MO	IT2C	IT2M2	IT2M1	IT2MO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET45	INTT4 & INTT5 Enable Register	D6h	INTT5(Timer5)				INTT4(Timer4)			
			IT5C	IT5M2	IT5M1	IT5MO	IT4C	IT4M2	IT4M1	IT4MO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET67	INTT6 & INTT7 Enable Register	D7h	INTT7(Timer7)				INTT6(Timer6)			
			IT7C	IT7M2	IT7M1	IT7MO	IT6C	IT6M2	IT6M1	IT6MO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET89	INTTR8 & INTTR9 Enable Register	D8h	INTTR9(Timer8)				INTTR8(Timer8)			
			IT9C	IT9M2	IT9M1	IT9MO	IT8C	IT8M2	IT8M1	IT8MO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETAB	INTTRA & INTTRB Enable Register	D9h	INTTRB(TimerA)				INTTRA(TimerA)			
			ITBC	ITBM2	ITBM1	ITBMO	ITAC	ITAM2	ITAM1	ITAMO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET08A	INTT08 & INTTOA (Overflow) Enable Register	DAh	INTTOA				INTT08			
			ITOAC	ITOAM2	ITOAM1	ITOAMO	IT08C	IT08M2	IT08M1	IT08MO
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
INTES0	INTRX0 & INTTX0 Enable Register	DBh	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES1	INTRX1 & INTTX1 Enable Register	DCh	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESEE0	INTSEM0 & INTSEE0 Enable Register	DFh	INTSEE0				INTSEM0			
			ISEE0C	ISEE0M2	ISEE0M1	ISEE0M0	ISEM0C	ISEM0M2	ISEM0M1	ISEM0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESED0	INTSER0 & INTSET0 Enable Register	E0h	INTSET0				INTSER0			
			ISET0C	ISET0M2	ISET0M1	ISET0M0	ISER0C	ISER0M2	ISER0M1	ISER0M0
			R	R/W	R	R/W				
			0	0	0	0	0	0	0	0
INTESEE1	INTSEM1 & INTSEE1 Enable Register	E1h	INTSEE1				INTSEM1			
			ISEE1C	ISEE1M2	ISEE1M1	ISEE1M0	ISEM1C	ISEM1M2	ISEM1M1	ISEM1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESED1	INTSER1 & INTSET1 Enable Register	E2h	INTSET1				INTSER1			
			ISET1C	ISET1M2	ISET1M1	ISET1M0	ISER1C	ISER1M2	ISER1M1	ISER1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESB0	INTSBE0 & INTSBS0 Enable Register	E3h	INTSBS0				INTSBE0			
			ISBS0C	ISBS0M2	ISBS0M1	ISBS0M0	ISBE0C	ISBE0M2	ISBE0M1	ISBE0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESB1	INTSBE1 & INTSBS1 Enable Register	E4h	INTSBS1				INTSBE1			
			ISBS1C	ISBS1M2	ISBS1M1	ISBS1M0	ISBE1C	ISBE1M2	ISBE1M1	ISBE1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC01	INTTC0 & INTTC1 Enable Register	F1h	INTTC1 (DMA1)				INTTC0 (DMA0)			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC23	INTTC2 & INTTC3 Enable Register	F2h	INTTC3 (DMA3)				INTTC2 (DMA2)			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC45	INTTC4 & INTTC5 Enable Register	F3h	INTTC5 (DMA5)				INTTC4 (DMA4)			
			ITC5C	ITC5M2	ITC5M1	ITC5M0	ITC4C	ITC4M2	ITC4M1	ITC4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC67	INTTC6 & INTTC7 Enable Register	F4h	INTTC7 (DMA7)				INTTC6 (DMA6)			
			ITC7C	ITC7M2	ITC7M1	ITC7M0	ITC6C	ITC6M2	ITC6M1	ITC6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTNMWDT	NMI & INTWD Enable Register	F7h	NMI				INTWD			
			ITCNM	-	-	-	ITCWD	-	-	-
			R				R			
			0	-	-	-	0	-	-	-

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
IIMC	Interrupt Input Mode Control Register	F6H (no RMW)	-	-	-	-	-	-	IOLE	NMIREE
			R/W							
			-	-	-	-	-	-	0	0
									0: INT0 edge mode 1: INT0 level mode	1: Operate even at NMI rise edge
INTCLR	Interrupt Clear Control Register	F8H (no RMW)	-	-	-	-	-	-	-	-
			W							
			0	0	0	0	0	0	0	0
			Interrupt Vector							

(7) DMA controller

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
DMA0V	DMA0 Start Vector Register	100h			DMA0 Start Vector					
			-	-	DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
			-	-	0	0	0	0	0	0
DMA1V	DMA1 Start Vector Register	101h			DMA1 Start Vector					
			-	-	DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
			-	-	0	0	0	0	0	0
DMA2V	DMA2 Start Vector Register	102h			DMA2 Start Vector					
			-	-	DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
			-	-	0	0	0	0	0	0
DMA3V	DMA3 Start Vector Register	103h			DMA3 Start Vector					
			-	-	DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
			-	-	0	0	0	0	0	0
DMA4V	DMA4 Start Vector Register	104h			DMA4 Start Vector					
			-	-	DMA4V5	DMA4V4	DMA4V3	DMA4V2	DMA4V1	DMA4V0
					R/W					
			-	-	0	0	0	0	0	0
DMA5V	DMA5 Start Vector Register	105h			DMA5 Start Vector					
			-	-	DMA5V5	DMA5V4	DMA5V3	DMA5V2	DMA5V1	DMA5V0
					R/W					
			-	-	0	0	0	0	0	0
DMA6V	DMA6 Start Vector Register	106h			DMA6 Start Vector					
			-	-	DMA6V5	DMA6V4	DMA6V3	DMA6V2	DMA6V1	DMA6V0
					R/W					
			-	-	0	0	0	0	0	0
DMA7V	DMA7 Start Vector Register	107h			DMA7 Start Vector					
			-	-	DMA7V5	DMA7V4	DMA7V3	DMA7V2	DMA7V1	DMA7V0
					R/W					
			-	-	0	0	0	0	0	0
DMAB	DMA Burst Register	108h	DMA Burst							
			DBST7	DBST6	DBST5	DBST4	DBST3	DBST2	DBST1	DBST0
			R/W							
			0	0	0	0	0	0	0	0
DMAR	DMA Request Register	109h (no RMW)	DMA Request							
			DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
			R/W							
			0	0	0	0	0	0	0	0

(8) Control register

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
CLKMOD	Clock Mode Register	10AH	HALTM1	HALTM0	-	WARM	-	CLKOE	CLKM1	CLKM0
			R/W			R/W		R/W		
			1	1	-	0	-	0	0	0
			Stand by mode 00: (reserved) 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode			Warming up time 0 : 2^{15}/fc 1 : 2^{17}/fc		CLKOUT1 Output Enable 0 : Not output 1 : Output	00: fc output 01: (reserved) 10: 2/5·fc output 11: (reserved)	
WDMOD	Watchdog Timer Mode Register	110H	WDTE	WDTP1	WDTP0	-	DRVE	I2WDT	RESCR	-
			R/W				R/W			
			1	0	0	-	0	0	0	0
			1: WDT Enable	00 : 2^{16}/fc 01 : 2^{18}/fc 10 : 2^{20}/fc 11 : 2^{22}/fc			1: Drive pin in STOP mode	IDLE2 0: Stop 1: Operate	1: Reset connect internally WDT out to RESET pin	Fix to "0"
WDCR	Watchdog Timer Control Register	111H	-							
			W							
			-							
			B1H : WDT Disable 4EH : WDT Clear							

(9) AD converter

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
ADMOD0	AD Mode Control Register 0	138H	EOCF	ADBF	-	-	ITM0	REPET	SCAN	ADS
			R				R/W			
			0	0	0	0	0	0	0	0
			AD Conversion End Flag 1:END	AD Conversion BUSY Flag 1:Busy	Fix to "0"	Fix to "0"	0: Every 1 time 1: Every 4 times	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"
ADMOD1	AD Mode Control Register 1	139H	VREFON	I2AD	-	-	ADCH3	ADCH2	ADCH1	ADCH0
			R/W	R/W			R/W			
			0	0	0	0	0	0	0	0
			Ladder resistance 0:OFF 1:ON	IDLE2 0:Stop 1:Operate	Fix to "0"	Fix to "0"	Input channel 0000: AN0 AN0 : : 1011: AN11 AN0→AN1→AN2→ ··· →AN11 1100, 1101, 1110, 1111 : reserved			
ADREG0L	AD Result Register 0 Low	120H	ADR01	ADR00	-	-	-	-	-	ADR0RF
			R						R	R
			Undefined		-	-	-	-	-	0
ADREG0H	AD Result Register 0 High	121H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
ADREG1L	AD Result Register 1 Low	122H	ADR11	ADR10	-	-	-	-	-	ADR1RF
			R							R
			Undefined		-	-	-	-	-	0
ADREG1H	AD Result Register 1 High	123H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
ADREG2L	AD Result Register 2 Low	124H	ADR21	ADR20	-	-	-	-	-	ADR2RF
			R							R
			Undefined		-	-	-	-	-	0
ADREG2H	AD Result Register 2 High	125H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
ADREG3L	AD Result Register 3 Low	126H	ADR31	ADR30	-	-	-	-	-	ADR3RF
			R							R
			Undefined		-	-	-	-	-	0
ADREG3H	AD Result Register 3 High	127H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
ADREG4L	AD Result Register 4 Low	128H	ADR41	ADR40	-	-	-	-	-	ADR4RF
			R							R
			Undefined		-	-	-	-	-	0
ADREG4H	AD Result Register 4 High	129H	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
			R							
			Undefined							
ADREG5L	AD Result Register 5 Low	12AH	ADR51	ADR50	-	-	-	-	-	ADR5RF
			R							R
			Undefined		-	-	-	-	-	0
ADREG5H	AD Result Register 5 High	12BH	ADR59	ADR58	ADR57	ADR56	ADR55	ADR54	ADR53	ADR52
			R							
			Undefined							
ADREG6L	AD Result Register 6 Low	12CH	ADR61	ADR60	-	-	-	-	-	ADR6RF
			R							R
			Undefined		-	-	-	-	-	0
ADREG6H	AD Result Register 6 High	12DH	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
			R							
			Undefined							
ADREG7L	AD Result Register 7 Low	12EH	ADR71	ADR70	-	-	-	-	-	ADR7RF
			R							R
			Undefined		-	-	-	-	-	0
ADREG7H	AD Result Register 7 High	12FH	ADR79	ADR78	ADR77	ADR76	ADR75	ADR74	ADR73	ADR72
			R							
			Undefined							
ADREG8L	AD Result Register 8 Low	130H	ADR81	ADR80	-	-	-	-	-	ADR8RF
			R							R
			Undefined		-	-	-	-	-	0
ADREG8H	AD Result Register 8 High	131H	ADR89	ADR88	ADR87	ADR86	ADR85	ADR84	ADR83	ADR82
			R							
			Undefined							
ADREG9L	AD Result Register 9 Low	132H	ADR91	ADR90	-	-	-	-	-	ADR9RF
			R							R
			Undefined		-	-	-	-	-	0
ADREG9H	AD Result Register 9 High	133H	ADR99	ADR98	ADR97	ADR96	ADR95	ADR94	ADR93	ADR92
			R							
			Undefined							
ADREGAL	AD Result Register A Low	134H	ADRA1	ADRA0	-	-	-	-	-	ADRARF
			R							R
			Undefined		-	-	-	-	-	0
ADREGAH	AD Result Register A High	135H	ADRA9	ADRA8	ADRA7	ADRA6	ADRA5	ADRA4	ADRA3	ADRA2
			R							
			Undefined							
ADREGBL	AD Result Register B Low	136H	ADRB1	ADRB0	-	-	-	-	-	ADRBRF
			R							R
			Undefined		-	-	-	-	-	0
ADREGBH	AD Result Register B High	137H	ADRB9	ADRB8	ADRB7	ADRB6	ADRB5	ADRB4	ADRB3	ADRB2
			R							
			Undefined							

(10) Memory controller

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
BCSL	BLOCK CS/WAIT Control Register Low	148H	-	BWW2	BWW1	BWW0	-	BWR2	BWR1	BWR0
				W				W		
			-	0	1	0	-	0	1	0
				Number of write waits 001:0wait 010:1wait 011:Nwait 101:2wait 110:3wait others : reserved				Number of read waits 001:0wait 010:1wait 011:Nwait 101:2wait 110:3wait others : reserved		
BCSH	BLOCK CS/WAIT Control Register High	149H	BE	BM	-	-	BOM1	BOM0	BBUS1	BBUS0
			W	W			W		W	
			1	0	0	0	0	0	0	0
			CS select 0:Disable 1:Enable	0:16MB 1:Sets area	Fix to "0"	Fix to "0"	00:SRAM/ROM 01, 10, 11:Resetved		00:8bit 01, 10, 11:reserved	
MAMR	Memory Address Mask Register	14AH	MV22	MV21	MV20	MV19	MV18	MV17	MV16	MV15
			R/W							
			1	1	1	1	1	1	1	1
			0:Compare enable 1:Compare disable							
MSAR	Memory Start Address Register	14BH	MS23	MS22	MS21	MS20	MS19	MS18	MS17	MS16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
RAMCR	RAM Write Control Register	16DH	RAMSTB	RAMWI	-	-	-	-	-	-
			R/W							
			Keep	1	-	-	-	-	-	-
			0: VCC3≦VSTB 1: VCC3>VSTB	RAM write 0:Disable 1:Enable						

(11) Serial Bus Interface (SBI)

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
SBI0CR1	SBI0 Control Register 1	170H (no RMW) I2C mode	BC2	BC1	BC0	ACK	-	SCK2	SCK1	SCK0/SWRMON
			W			R/W		W		R/W
			0	0	0	0	-	0	0	0
			Number of transfer bits 000:8 001:1 010:2 011:3 100:4 101:5 110:6 11			Acknowledge mode 0:Disable 1:Enable		Setting of the divide value "n" 000:5 001:6 010:7 011:8 100:9 101:10 110:11 111:reserved		
		170H (no RMW) SIO mode	SIOS	SIOINH	SIOM1	SIOM0	-	SCK2	SCK1	SCK0/SWRMON
			W					W		R/W
			0	0	0	0	-	0	0	0
			Transfer 0:Stop 1:Start	Transfer 0:Continue 1:Abort	Transfer mode 00:8bit transmit 10:8bit transmit/receive 11:8bit receive			Setting of the divide value "n" 000:4 001:5 010:6 01 100:8 101:9 110:10 111:external clock SCK0		
SBI0DBR	SBI0 Buffer Register	171H (no RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R(Receiving)/W(Transmission)							
			Undefine							
I2C0AR	I2CBUS0 Address Register	172H (no RMW)	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			W							
			0	0	0	0	0	0	0	0
			Setting Slave Address							Address recognition 0:Enable 1:Disable
SBI0CR2	SBI0 Control Register 2	173H (no RMW) I2C mode	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
			W							
			0	0	0	1	0	0	0	0
			0:Slave 1:Master	0:Receive 1:Transmit	Start/stop generation 0:Stop 1:Start	INTSBE0 interrupt 0:Request 1:Cancel	Operation mode selection 00:Poer mode 10:I2C mode 01:SIO mode 11:reserved		Software reset generate write "10" and "01", then an internal reset signal is generated.	
		173H (no RMW) SIO mode	-	-	-	-	SBIM1	SBIM0	-	-
							W			
			-	-	-	-	0	0	-	-
							Operation mode selection 00:Poer mode 10:I2C mode 01:SIO mode 11:reserved			
SBI0SR	SBI0 Status Register	173H (no RMW) I2C mode	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
			R							
			0	0	0	1	0	0	0	0
			0:Slave 1:Master	0:Receive 1:transmit	Bus status monitor 0:Free 1:Busy	INTSBE0 interrupt 0:request 1:Cancel	Arbitration lost detection monitor 1:Detect	Slave address match detection monitor 1:Detect	General call detection 1:Detect	Last receive bit monitor 0: "0" 1: "1"
		173H (no RMW) SIO mode	-	-	-	-	SIOF	SEF	-	-
							R			
			-	-	-	-	0	0	-	-
							Transfer status 0:Stopped 1:In progress	Shift status 0:Stopped 1:In progress		

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
SBI1CR1	SBI1 Control Register 1	178H (no RMW) I2C mode	BC2	BC1	BC0	ACK	-	SCK2	SCK1	SCKO/SWRMON
			W			R/W		W		R/W
			0	0	0	0	-	0	0	0
			Number of transfer bits 000:8 001:1 010:2 011:3 100:4 101:5 110:6 111:7			Acknowledge mode 0:Disable 1:Enable		Setting of the divide value "n" 000:5 001:6 010:7 011:8 100:9 101:10 110:11 111:reserved		
		178H (no RMW) SIO mode	SIOS	SIOINH	SIOM1	SIOMO	-	SCK2	SCK1	SCKO/SWRMON
			W					W		R/W
			0	0	0	0	-	0	0	0
			Transfer 0:Stop 1:Start	Transfer 0:Continue 1:Abort	Transfer mode 00:8bit transmit 10:8bit transmit/receive 11:8bit receive			Setting of the divide value "n" 000:4 001:5 010:6 01 100:8 101:9 110:10 111:external clock SCK1		
SBI1DBR	SBI1 Buffer Register	179H (no RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R(Receiving)/W(Transmission)							
			Undefine							
I2C1AR	I2CBUS1 Address Register	17AH (no RMW)	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			W							
			0	0	0	0	0	0	0	0
			Setting Slave Address							Address recognition 0:Enable 1:Disable
SBI1CR2	SBI1 Control Register 2	17BH (no RMW) I2C mode	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
			W							
			0	0	0	1	0	0	0	0
			0:Slave 1:Master	0:Receive 1:Transmit	Start/stop generation 0:Stop 1:Start	INTSBE1 interrupt 0:Request 1:Cancel	Operation mode selection 00:Poer mode 10:I2C mode 01:SIO mode 11:reserved		Software reset generate write "10" and "01", then an internal reset signal is generated.	
		17BH (no RMW) SIO mode	-	-	-	-	SBIM1	SBIM0	-	-
							W			
			-	-	-	-	0	0	-	-
							Operation mode selection 00:Poer mode 10:I2C mode 01:SIO mode 11:reserved			
SBI1SR	SBI1 Status Register	17BH (no RMW) I2C mode	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
			R							
			0	0	0	1	0	0	0	0
			0:Slave 1:Master	0:Receive 1:transmit	Bus status monitor 0:Free 1:Busy	INTSBE1 interrupt 0:request 1:Cancel	Arbitration lost detection monitor 1:Detect	Slave address match detection monitor 1:Detect	General call detection 1:Detect	Last receive bit monitor 0: "0" 1: "1"
		17BH (no RMW) SIO mode	-	-	-	-	SIOF	SEF	-	-
							R			
			-	-	-	-	0	0	-	-
							Transfer status 0:Stopped 1:In progress	Shift status 0:Stopped 1:In progress		

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
SBI0BR0	SBI0 Baud rate Register 0	174H	-	I2SBI0	-	-	-	-	-	-
				R/W						
			-	0	-	-	-	-	-	-
				IDLE2 0:Abort 1:Operate						
SBI0BR1	SBI0 Baud rate Register 1	175H	P4EN	-	-	-	-	-	-	-
			R/W							
			0	-	-	-	-	-	-	-
			Clock control 0:Abort 1:Operate							
SBI1BR0	SBI1 Baud rate Register 0	17CH	-	I2SBI0	-	-	-	-	-	-
				R/W						
			-	0	-	-	-	-	-	-
				IDLE2 0:Abort 1:Operate						
SBI1BR1	SBI1 Baud rate Register 1	17DH	P4EN	-	-	-	-	-	-	-
			R/W							
			0	-	-	-	-	-	-	-
			Clock control 0:Abort 1:Operate							

6. Port Section Equivalent Circuit Diagram.

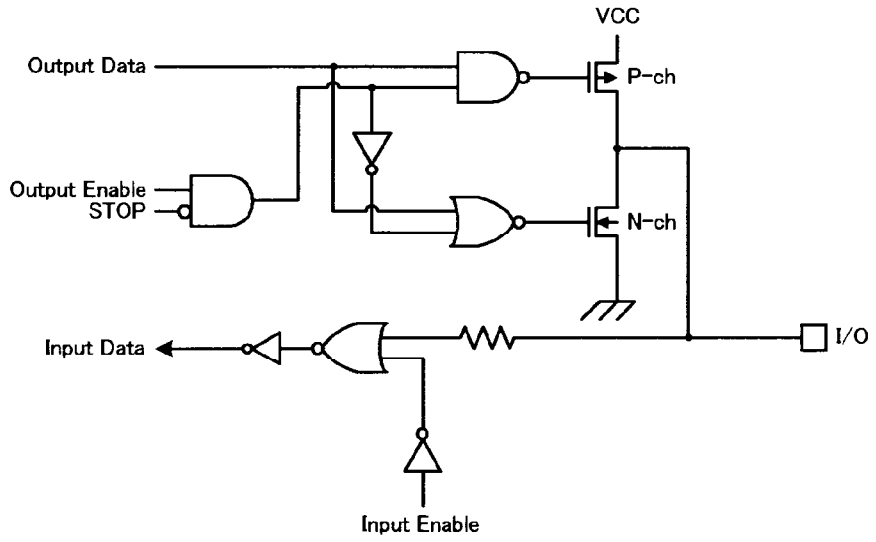
- Reading The Circuit Diagram

  Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

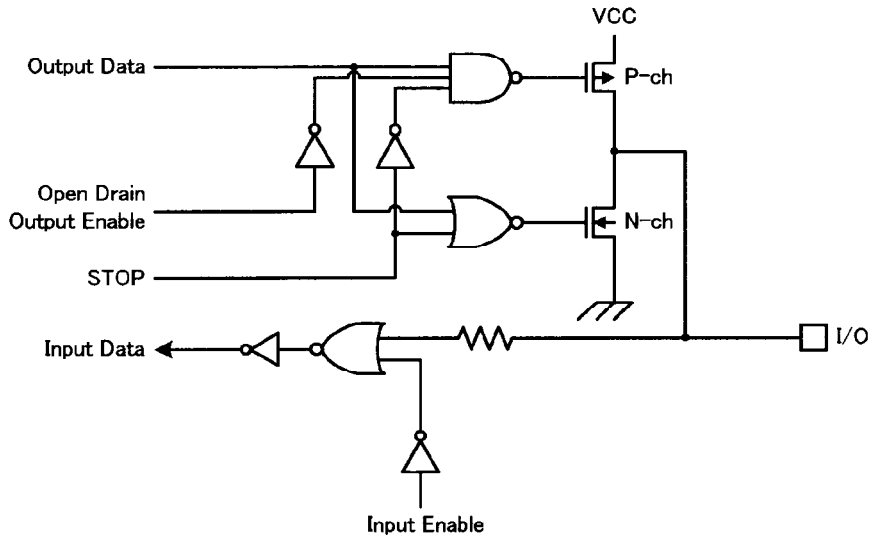
  The dedicated signal is described below.

  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 <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.

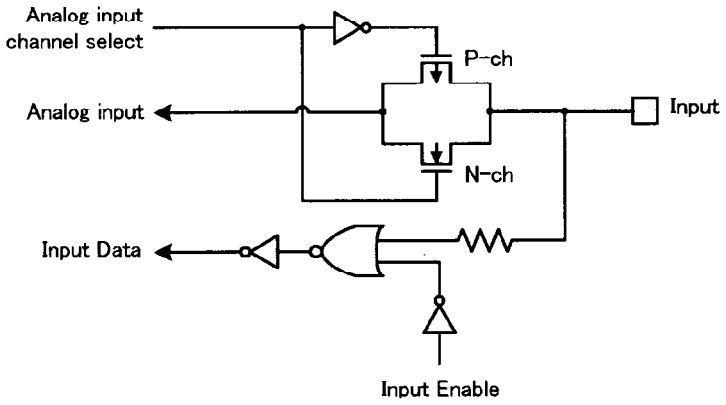
■ P0(D0 to D7), P4(A0 to A7), P70 to P75, PC0 to PC5, PD0 to PD7, PF1(RXD0), PF2(CTS0, SCLK0), PF4 (RXD1), PF5 ($\overline{\text{CTS1}}$, SCLK1), PF6, PF7, PM0 ($\overline{\text{SS0}}$), PM4 ($\overline{\text{SS1}}$), PN0 (SCK0), PN3 (SCK1)



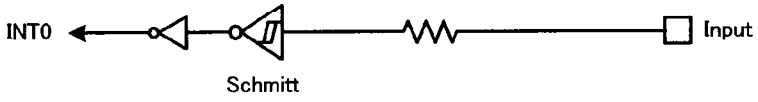
■ PF0(TXD0), PF3(TXD1), PM1(MOSI0), PM2(MISO0), PM3(SECLK0), PM5(MOSI1), PM6 (MISO1), PM7 (SECLK1), PN1 (SO0/SDA0), PN2 (SI0/SCL0), PN4 (SO1/SDA1), PN5(SI1/SCL1)



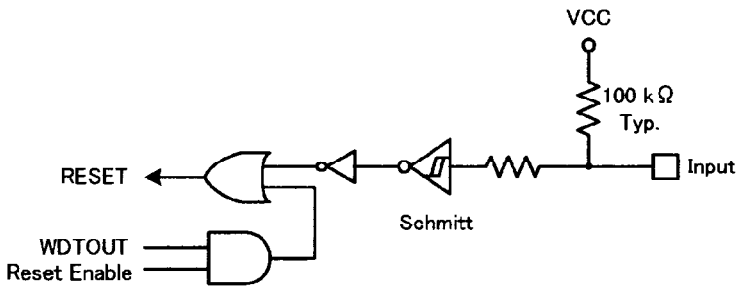
■ PG(AN0 to 7), PL0 to 3(AN8 to 11)



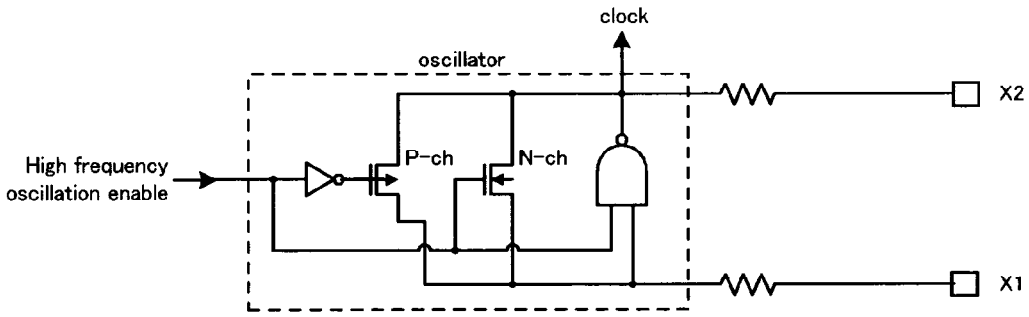
■ INT0



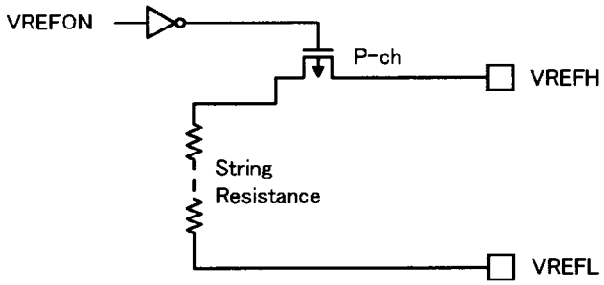
■ $\overline{\text{RESET}}$



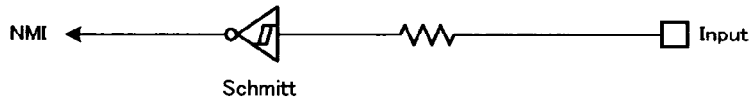
■ X1, X2



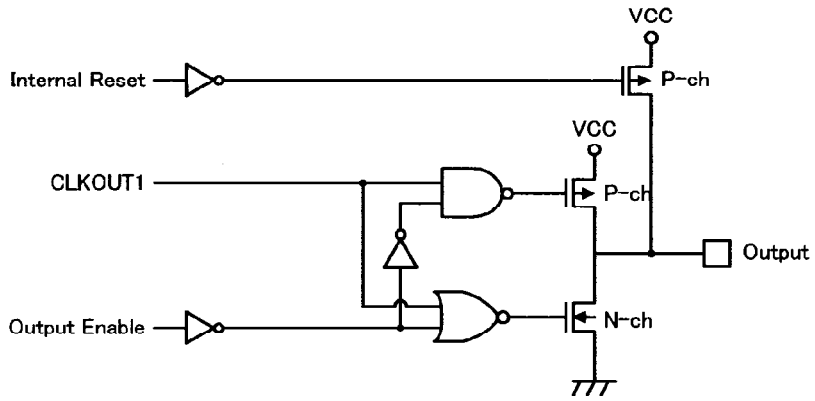
■ VREFH, BREFL



■ $\overline{\text{NMI}}$



■ CLKOUT1



■ (AM0 to 1), (TEST0 to 1)