TOSHIBA

TOSHIBA Original CMOS 32-Bit Microcontroller

TLCS-900/H1 Series

TMP92CD54IFG

Tentative

TOSHIBA CORPORATION

Semiconductor Company

Preface

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

CMOS 32-bit Micro-controller

TMP92CD54IFG

1. Outline and Device Characteristics

The TMP92CD54I is a high-performance 32-bit microcontroller incorporating a Toshiba-proprietary CPU, the TLCS-900/H1 core. The TMP92CD54I is developed for various automotive equipments which require high-speed data processing.

Housed in a 100-pin mini-flat package, the TMP92CD54I is best suited for high-density implementation of user systems.

The characteristics of the TMP92CD54I are listed below:

(1) Toshiba-proprietary high-speed 32-bit CPU (TLCS-900/H1 CPU)

Fully-compatible with the instruction codes of the TLCS-900, TLCS-900/L, ELCS-900/L1, TLCS-900/H and TLCS-900/H2

16 Mbytes of linear address space

General-purpose registers and register banks

Micro DMA: 8 channels (250 ns/4 bytes at fc = 20 MHz)

Minimum instruction execution time: 50 ns (at fc = 20 MHz)

Internal data bus: 32-bit wide

(2) Internal memory

Internal RAM : 32K-byte (32 bit/one clock access time, can be used for instructions.

Internal ROM : 512K-byte Mask ROM

(3) External memory expansion

Expandable up to 16-Mbyte (for code and data)

External data bus: 8-bit wide (The upper address bus is not available when the built-in I/Os are selected.)

(4) Memory controller (MEMC)

Chip select output: 1 channel

(5) 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)

(6) 16-bit timer : 2 channels

16-bit interval timer mode (2 channels)

16-bit event counter mode (2 channels)

16-bit programmable pulse generation (PPG) output mode (2 channels)

Frequency measurement mode

Pulse width measurement mode

Time difference measurement mode

(7) Serial interface (SIO) : 2 channels

I/O interface mode

Universal asynchronous receiver transmitter (UART) mode

(8) Serial expansion interface (SEI) : 1 channel

Baud rate 4M / 2M / 500Kbps at fc = 20MHz.

(9) Serial bus interface (SBI) : 3 channels

Clock-synchronous 8-bit serial interface mode

I²C bus mode

(10) CAN controller : 1 channel

Supports CAN version 2.0B.

16 mailboxes

(11) 10-bit A/D converter (ADC) : 12 channels

A/D conversion time: 8 μsec (at fc = 20 MHz)

Total tolerance: ± 3 LSB (excluding quantization error)

Scan mode for all 12 channels

(12) Watch dog timer (WDT)

(13) Timer for real-time clock (RTC)

Can operate with low-frequency oscillator only.

(14) Interrupt controller (INTC) : 60 interrupt sources

9 interrupts from CPU (Software interrupts and undefined instruction interrupt)

42 internal interrupt vectors

9 external interrupt vectors (INT0 to INT7, $\overline{\text{NMI}}$)

(15) I/O Port : 68 pins

(16) Standby mode

Four modes: IDLE3, IDLE2, IDLE1 and STOP

STOP mode can be released by 9 external inputs.

(17) Internal voltage detection flag (RAMSTB)

(18) Power supply voltage

V_{CC5} = 4.5 V to 5.25 V

V_{CC3} = 3.3 V (Connect REGOUT (built-in voltage regulator output) to DVCC3.)

(19) Operating temperature : -40 to 85 degree C

(20) Package : LQFP100-P-1414-0.50F
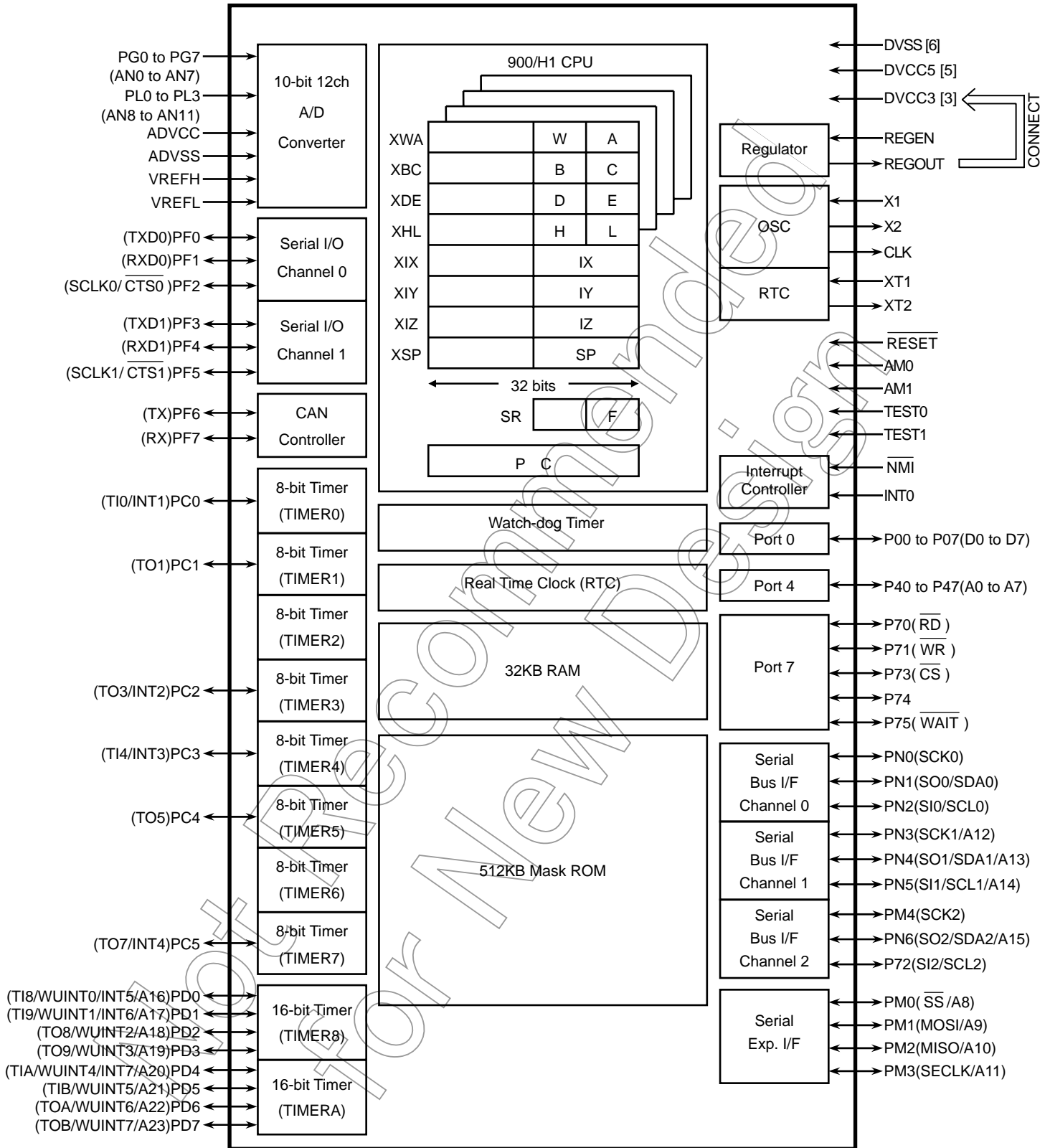
Figure 1.1   TMP92CD54I block diagram

2.   Pin Assignment and Functions

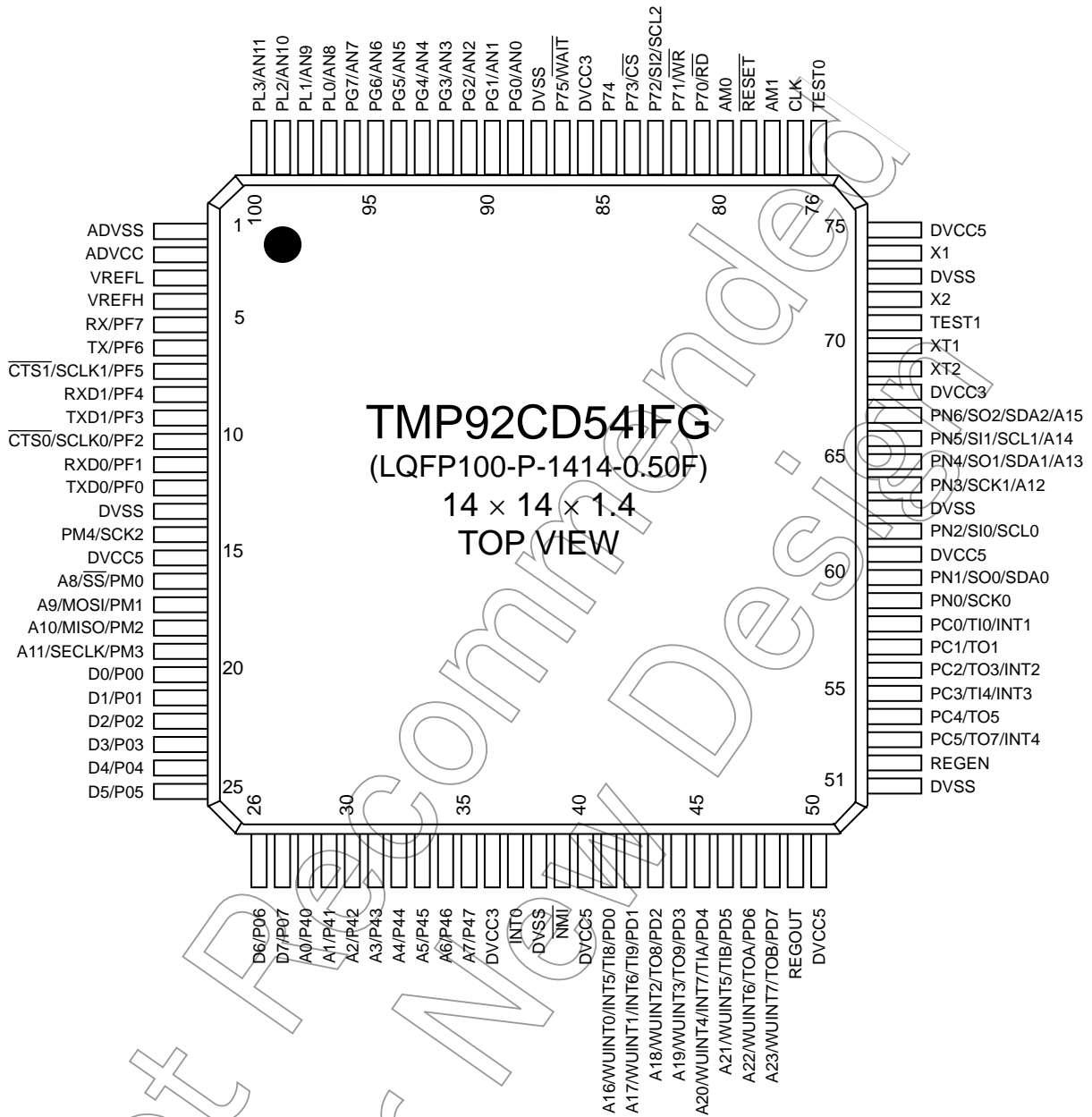### 2.1   Pin Assignment

Figure 2.1   TMP92CD54I   Pin Assignment

2.2   Pin names and functions

The names and functions of the input/output pins are described in are described in the Tables 2.2.1 to 2.2.4.
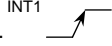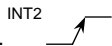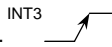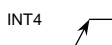
Table 2.2.1 Input/output pins (1/4)
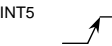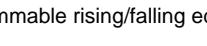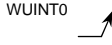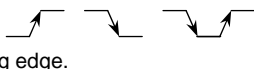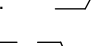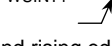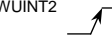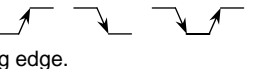
Pin name	Pin number	Number of pins	In/Out	Function
P00 to P07 D0 to D7	20 to 27	8 (CMOS) (TTL)	in/out in/out	Port 0: I/O port. Input or output specifiable in units of bits. Data: Data bus 0 to 7.
P40 to P47 A0 to A7	28 to 35	8	in/out out	Port4: I/O port. Input or output specifiable in units of bits. Address: Address bus 0 to 7.
P70 $\overline{RD}$	81	1	in/out out	Port70: I/O port. Read: Outputs strobe signal to read external memory.
P71 $\overline{WR}$	82	1	in/out out	Port 71: I/O port. Write: Output strobe signal to write external memory.
P72 SI2 SCL2	83	1	in/out	Port 72: I/O port. SBI channel 2: Input data at SIO mode SBI channel 2: Clock input/output at I²C mode
P73 $\overline{CS}$	84	1	in/out out	Port 73: I/O port. Chip select: Outputs "low" if address is within specified address area.
P74	85	1	in/out	Port 74: I/O port.
P75 $\overline{WAIT}$	87	1	in/out in	Port 75: I/O port. Wait: Signal used to request CPU bus wait.
PC0 TI0 INT1	58	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	57	1	in/out out	Port C1: I/O port. Timer output 1: Output pin for timer 1.
PC2 TO3 INT2	56	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	55	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	54	1	in/out out	Port C4: I/O port. Timer output 5: Output pin for timer 5.
PC5 TO7 INT4	53	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 A16 WUINT0	41	1	in/out in in out in	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 16. Wake up input 0: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD1 TI9 INT6 A17 WUINT1	42	1	in/out in in out in	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 17. Wake up input 1: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD2 TO8 A18 WUINT2	43	1	in/out out out in	Port D2: I/O port. Timer output 8: Output pin for timer 8 Address: Address bus 18. Wake up input 2: Wake up request pin with programmable rising, falling or both falling and rising edge.

Table 2.2.2 Input/output pins (2/4)
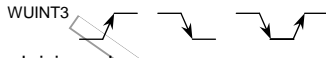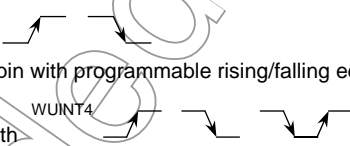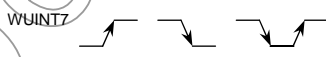
Pin name	Pin number	Number of pins	In/Out	Function
PD3 TO9 A19 WUINT3	44	1	in/out out out in	Port D3: I/O port. Timer output 9: Output pin for timer 9 Address: Address bus 19. Wake up input 3: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD4 TIA INT7 A20 WUINT4	45	1	in/out in in out in	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 20. Wake up input 4: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD5 TIB A21 WUINT5	46	1	in/out in out in	Port D5: I/O port. Timer input B: Input pin for timer B. Address: Address bus 21. Wake up input 5: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD6 TOA A22 WUINT6	47	1	in/out out out in	Port D6: I/O port. Timer output A: Output pin for timer A. Address: Address bus 22. Wake up input 6: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD7 TOB A23 WUINT7	48	1	in/out out out in	Port D7: I/O port. Timer output B: Output pin for timer B. Address: Address bus 23. Wake up input 7: Wake up request pin with programmable rising, falling or both falling and rising edge.
PF0 TXD0	12	1	in/out out	Port F0: I/O port. Serial interface channel 0: Transmission data.
PF1 RXD0	11	1	in/out in	Port F1: I/O port. Serial interface channel 0: Receive data.
PF2 $\overline{SCLK0}$ CTS0	10	1	in/out in/out in	Port F2: I/O port. Serial interface channel 0: Clock input/output. Serial interface channel 0: Data ready to send. (Clear-to-send)
PF3 TXD1	9	1	in/out out	Port F3: I/O port. Serial interface channel 1: Transmission data.
PF4 RXD1	8	1	in/out in	Port F4: I/O port. Serial interface channel 1: Receive data.
PF5 SCLK1 $\overline{CTS1}$	7	1	in/out in/out in	Port F5: I/O port. Serial interface channel 1: Clock input/output. Serial interface channel 1: Data ready to send. (Clear-to-send)
PF6 TX	6	1	in/out out	Port F6: I/O port. CAN: Transmission data.
PF7 RX	5	1	in/out in	Port F7: I/O port. CAN: Receive data.
PG0 to PG7 AN0 to AN7	89 to 96	8	in in	Port G: Input-only port. Analog input 0 to 7: AD converter input pins.
PL0 to PL3 AN8 to AN11	97 to 100	4	in in	Port L0 to L3: Input-only port. Analog input 8 to 11: AD converter input pins.
PM0 $\overline{SS}$ A8	16	1	in/out in out	Port M0: I/O port. SEI: Slave select input. Address: Address bus 8.
PM1 MOSI A9	17	1	in/out in/out out	Port M1: I/O port. SEI: Master output, slave input. Address: Address bus 9.

Table 2.2.3 Input/output pins (3/4)

Pin name	Pin number	Number of pins	In/Out	Function
PM2 MISO A10	18	1	in/out in/out out	Port M2: I/O port. SEI: Master input, slave output. Address: Address bus 10.
PM3 SECLK A11	19	1	in/out in/out out	Port M3: I/O port. SEI: Clock input/output. Address: Address bus 11.
PM4 SCK2	14	1	in/out in/out	Port M4: I/O port. SBI channel 2: Clock input/output at SIO mode.
PN0 SCK0	59	1	in/out in/out	Port N0: I/O port. SBI channel 0: Clock input/output at SIO mode.
PN1 SO0 SDA0	60	1	in/out out in/out	Port N1: I/O port. SBI channel 0: Output data input/output at SIO mode SBI channel 0: Data input/output at I²C mode
PN2 SI0 SCL0	62	1	in/out in in/out	Port N2: I/O port. SBI channel 0: Input data at SIO mode SBI channel 0: Clock input/output at I²C mode
PN3 SCK1 A12	64	1	in/out in/out out	Port N3: I/O port. SBI channel 1: Clock input/output at SIO mode Address: Address bus 12.
PN4 SO1 SDA1 A13	65	1	in/out out in/out out	Port N4: I/O port. SBI channel 1: Output data at SIO mode SBI channel 1: Data input/output at I²C mode Address: Address bus 13.
PN5 SI1 SCL1 A14	66	1	in/out in in/out out	Port N5: I/O port. SBI channel 1: Input data at SIO mode SBI channel 1: Clock input/output at I²C mode Address: Address bus 14
PN6 SO2 SDA2 A15	67	1	in/out out	Port N6: I/O port. SBI channel 2: Output data at SIO mode SBI channel 2: data input output at I2C mode Address: Address bus 15.
$\overline{\text{NMI}}$	39	1	in	Non-maskable interrupt: Interrupt request pin with programmable falling or both falling and rising edge.
INT0	37	1	in	Interrupt request pin 0: Interrupt request pin with programmable level or rising-edge.
AM0,1	80, 78	2	in	Address Mode selection: Connect AM0 pin to L and AM1 pin to H for Single Chip mode.
TEST0,1	76, 71	2	in	Test mode pins: Should be tied to GND.
CLK	77	1	out	Programmable clock output (with pull-up resistor)
X1/X2	74, 72	2	in/out	High-frequency oscillator connecting pins: To drive these pins with an external clock, apply clock signals of 3.3 V.
XT1/XT2	70, 69	2	in/out	Low-frequency oscillator connecting pins: To drive these pins with an external clock, apply clock signals of 3.3 V.
$\overline{\text{RESET}}$	79	1	in	Reset: Initializes LSI (with pull-up resistor).
VREFH	4	1	in	AD reference voltage high
VREFL	3	1	in	AD reference voltage low
ADVCC	2	1	-	Power supply pin for AD converter (+5V): Connect the ADVCC pin to 5-V power supply.
ADVSS	1	1	-	GND pin for AD converter: Connect the ADVSS pin to GND (0V).

Table 2.2.4 Input/output pins (4/4)

Pin name	Pin number	Number of pins	In/Out	Function
DVCC5	15, 40, 50, 61, 75	5	-	Power supply pins (+5V): Connect all the DVCC5 pins to 5-V power supply.
DVCC3	36, 68, 86	3	-	Power supply pins (+3.3V): Connect all the DVCC3 pins to REGOUT pin.
DVSS	13, 38, 51, 63, 73, 88	6	-	GND: Connect all DVSS pins to GND (0V).
REGOUT	49	1	out	Regulator output 3.3V: Connect capacitor to stabilize the regulator output.
REGEN	52	1	in	Regulator enable pin: Should be set to H or OPEN (with pull-up resistor).

3. Operation

This section describes the basic functions and operations of the TMP92CD54I for each functional block.

3.1 CPU

The TMP92CD54I incorporates a high-performance, high-speed 32-bit CPU, the TLCS-900/H1.

3.1.1 CPU Overview

The TLCS-900/H1 is a high-performance, high-speed CPU based on the TLCS-900/L1 and has a built-in data bus extended to 32 bits to enable faster processing.

Table 3.1.1 shows an overview of the CPU built into the TMP92CD54I.:

Table 3.1.1 CPU Overview

Properties	TLCS-900/H1	
Width of CPU Address Bus	24 bit	
Width of CPU Data Bus	32 bit	
Internal Operating Frequency	16 to 20MHz (f_{OSC} = 8 to 10MHz)	
Minimum Bus Cycle (Internal RAM)	1 clock access (50ns @ f_{OSC} = 10MHz)	
Internal RAM	32 bit 1 clock access	
Internal ROM	32 bit interleave 2-1-1-1 clock access	
Internal I/O	8/16 bit 2 clock access	PORT, INTC, MEMC
	8/16 bit 5 to 6 clock access	SEI, SIO, WDT, 8 bit Timer, 16 bit Timer, RTC, 10-bit ADC, SBI, CAN
External Device	8 bit 2 clock access (can insert wait cycles)	
Minimum Instruction Execution Cycle	1 clock (50ns at f_{OSC} = 10MHz)	
Conditional Jump	2 clock (100ns at f_{OSC} = 10MHz)	
Instruction Queue Buffer	12 byte	
Instruction Set	Compatible with TLCS-900, 900/H, 900/L, 900/L1 and 900/H2 (NORMAL, MIN, MAX and LDX instructions are not supported)	
Micro DMA	8 channels	

3.1.2   Reset

To apply a reset to the TMP92CD54I, drive the $\overline{\text{RESET}}$ input pin Low for at least 4 μs (when f_{OSC} = 10 MHz) when the internal oscillator and clock multiplier are operating stably with the supply voltage in the normal operating range.

The clock multiplier is bypassed during the reset period so that the system clock frequency, f_C, becomes 5 MHz (when f_{OSC} = 10 MHz).

When a reset is accepted, the CPU operates as follows:

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

    PC<0 to 7>   ← data in location FFFF00H
    PC<8 to 15>   ← data in location FFFF01H
    PC<16 to 23> ← data in location FFFF02H
- Sets the Stack Pointer (XSP) to 00000000H.
- Sets bits <IFF0 to IFF2> of the Status Register (SR) to 111 (thereby setting the Interrupt Level Mask Register to level 7).
- Clears bits <RFP0 to RFP1> of the Status Register to 00 (thereby selecting Register Bank 0).

When a reset is released, the CPU starts fetching and executing instructions according to the program counter (PC). The registers within the CPU other than those shown above remain unchanged.

A reset being accepted also causes the built-in I/O, input/output port and other pins to be initialized 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 $\overline{\text{RESET}}$ input pin is driven High, the built-in clock multiplier starts operating and the internal reset is released after the setting time for the circuit (1.6384 ms when f_{OSC} = 10MHz) elapses.

Upon a power-on reset, the control signals for the memory controller are unstable, possibly resulting in backup data being rewritten in external RAM connected to the TMP92CD54I.

When the $\overline{\text{RESET}}$ input pin goes Low, the input/output ports are initialized to input mode and the CLKOUT pin output setting is initialized to High-Z output. The CLKOUT pin outputs High because it is pulled up within the device. Since the pull-up circuit operates on the DVCC3 supply, however, the internal transistor on/off operation is not stable while the DVCC3 supply is rising, resulting in either a High-Z or High (pulled up) output.

3.1.3 Selecting a Startup Mode

Set TEST0 and TEST1 to GND, AM0 to Low and AM1 to High to select single-chip mode.
.

Table 3.1.2 Operation Mode Setup

Operation Mode	Mode Setup input pin				
	RESET	AM1	AM0	TEST1	TEST0
Single-chip Mode	╱	H	L	L	L

3.2 Memory Map

Figure 3.2.1 shows a memory map of the TMP92CD541I.
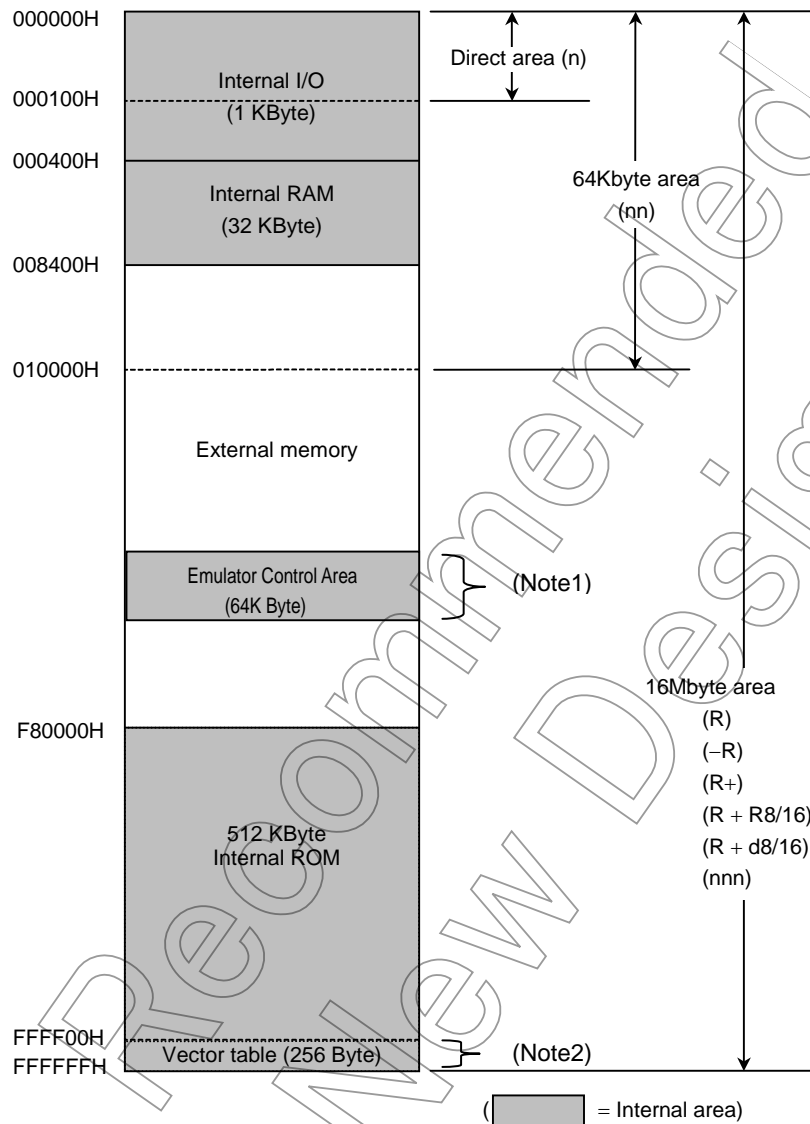
Figure 3.2.1 Memory Map

Note 1: When an emulator is used, 64 Kbytes of the 16-Mbyte space are used to control the emulator and not available to the user.

Note 2: Accessing the emulator control space causes the $\overline{WR}$ and $\overline{RD}$ signals to be output. This should be taken into account when using expanded memory.

Note 3: The last 16 bytes (addresses FFFFF0H to FFFFFFH) in the vector table are reserved as internal space and cannot be used.

Note 4: If memory devices having different bus widths are located at contiguous addresses, any access spanning those devices should not be executed with a single instruction. Such an attempt may prevent data from being read or written normally.

3.3 Clock Function and Standby Functions

3.3.1 System Clock Block Diagram
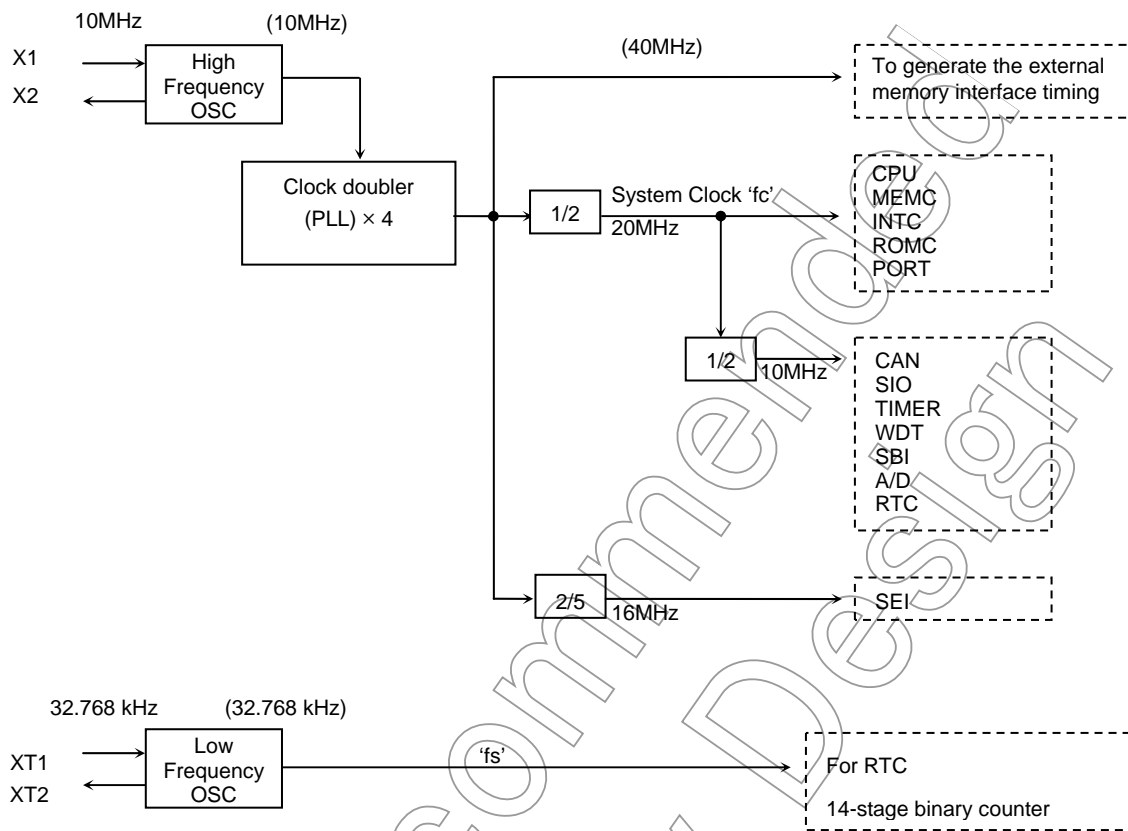
Figure 3.3.1 Block Diagram of System clock

3.3.2 Standby Controller

（1） Halt Mode

Executing the HALT instruction (stop instruction) sets the operating mode to any of IDLE2, IDLE1, IDLE3 and STOP depending on the setting of CLKMOD<HALTM1:0>.

Clock Mode Register

	7	6	5	4	3	2	1	0
bit Symbol	HALTM1	HALTM0	-	-	-	CLKOE	CLKM1	CLKM0
Read/Write	R/W			R/W		R/W		
After reset	1	1	-	0	-	0	0	0
Function	Halt mode 00: IDLE3 01: STOP 10: IDLE1 11: IDLE2			Fix to "0"		CLK output enable 0: not output 1: output	CLK output select 00: fc 01: Reserved 10: 2/5 fc 11: Reserved	

CLKMOD (010AH)

CLK output clock select

00	fc
01	Reserved
10	2/5 fc
11	Reserved

CLK output enable

0	No output (High-Z, Pulled up)
1	Output

Selects standby mode by HALT instruction

00	IDLE3
01	STOP
10	IDLE1
11	IDLE2

Figure 3.3.2 Clock Mode Register

The following shows whether individual blocks operate or stop in each mode:

1. IDLE2 mode: Only the CPU is stopped.

Each built-in I/O block has a bit that controls whether it operates or stops in IDLE2 mode.
The bits shown in Table 3.3.1 are used to control the operation of built-in I/O blocks.

Table 3.3.1 the registers to control operation during Idle2 Mode

Internal I/O	SFR Registers
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>
SBI2	SBI2BR0<I2SBI2>
A/D converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>

2. IDLE1 mode: Only the low-speed oscillator and high-speed oscillator operate.
3. IDLE3 mode: Only the low-speed oscillator and RTC operate.
4. STOP mode: All internal circuits are stopped.

Table 3.3.2 shows which blocks operate and stop during halt mode.

Table 3.3.2 I/O operation during Halt Modes

Halt Mode		IDLE2	IDLE1	IDLE3	STOP
CLKMOD <HALT1:0>		11	10	00	01
	CPU	Halt			
	I/O ports	Hold the same state since the HALT instruction was executed.		See table 3.3.5	
Block	8-bit TMR, 16-bit TMR	Selectable See table 3.3.1	Stop		
	SIO, SBI				
	A/D converter				
	WDT				
	RTC, XT1	Operational			
	CAN, SEI				
	Interrupt controller				

(2) Releasing a halt mode

A halt mode can be released with a reset or an interrupt request. Available halt release sources depend on the state of the interrupt mask register <IFF2:0> and the halt mode.

Table 3.3.3 shows details.

- Release using an interrupt request

Whether a halt state is released with an interrupt request depends on the interrupt enable status. If the interrupt request level set before the execution of the HALT instruction is greater than or equal to the value stored in the interrupt mask register, the halt mode is released, followed by interrupt handling for that interrupt source, after which processing is started from the instruction next to the HALT instruction. If the interrupt request level is lower than the value in the interrupt mask register, the halt mode is not released (a nonmaskable interrupt, however, always causes the halt mode to be released and the interrupt to be handled, regardless of the mask register value).

Only an INT0 interrupt, however, releases the halt mode even if the interrupt request level is lower than the value in the interrupt mask register. In that case, interrupt handling is not performed and processing is started from the instruction next to the HALT instruction (the INT0 interrupt request flag maintains the value of "1").

- Release with a reset

A reset causes all halt modes to be released.

To release STOP or IDLE3 mode, however, it requires a sufficient reset time (10 ms or longer when f_{OSC} = 10 MHz) for the internal oscillator to operate stably.

When a halt mode is released with a reset, the data in built-in RAM can hold the values it had immediately before entering the halt state but other settings are initialized (a release with an interrupt allows both RAM data and other settings to maintain their pre-halt values).

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	Idle3	Stop	Idle2	Idle1	Idle3	Stop
Source of Halt state clearance	Interrupt Source	NMI	◎	◎	◎*1	◎*1	–	–	–	–
		INTWDT	◎	×	×	×	–	–	–	–
		INT0	◎	◎	◎*1*2	◎*1*2	○	○	○*1*2	○*1*2
		INT0 [MASK]	○	○	○*1*2	○*1*2	○	○	○*1*2	○*1*2
		INT1 to 7	◎	×	×	×	×	×	×	×
		INTT0 to 7	◎	×	×	×	×	×	×	×
		INTTR8 to B	◎	×	×	×	×	×	×	×
		INTTO8, INTTOA	◎	×	×	×	×	×	×	×
		INTRX0 to 1, TX0 to 1	◎	×	×	×	×	×	×	×
		INTCR0, INTCT0, INTCG0	◎	×	×	×	×	×	×	×
		INTSEM0, E0, R0, T0	◎	×	×	×	×	×	×	×
		INTSBE0, S0, E1, S1, E2, S2	◎	×	×	×	×	×	×	×
		INTAD	◎	×	×	×	×	×	×	×
		All the above-mentioned interrupts [MASK]	×	×	×	×	×	×	×	×
		INTRTC	◎	◎	◎*1	×	○	○	○*1	×
		INTRTC [MASK]	○	○	○*1	×	○	○	○*1	×
	RESET		◎	◎	◎	◎	◎	◎	◎	◎

◎: Upon release from a halt, the CPU starts handling the interrupt (RESET causes the device to be initialized).

○: Upon release from a halt, the CPU starts processing from the instruction next to the HALT instruction.

×: Cannot be used to release a halt.

–: These combinations are not available because the interrupt priority level (interrupt request level) for nonmaskable interrupts is fixed to 7 (top priority).

*1: A halt is released after a warm-up time elapses.

*2: Any WUINT interrupt (WUINT0 to 7) causes an INT0 interrupt to occur.

Note 1: To release a halt using a level-mode INT0 interrupt in the interrupt-enabled state, hold it High until interrupt handling starts. If it is driven Low before interrupt handling starts, the interrupt cannot be handled normally.

Note 2: When using an INT5 to INT7 external interrupt in IDLE2 mode, set TRUN8<I2T8> and TRUNA<I2TA> to 1.

(Example - clearing IDLE1 Mode)

An INT0 interrupt in edge mode is used to release a halt in IDLE1 mode.

```
Address
8203H    LD    (IIMC), 00H        ; Selects INT0 interrupt rising edge.
8206H    LD    (INTE0AD), 06H     ; Sets INT0 interrupt level to 6.
8209H    EI    5                  ; Sets the CPU interrupt mask register to 5.
820BH    LD    (CLKMOD), 80H      ; Sets Halt mode to Idle1 Mode.
820EH    HALT                     ; Halts CPU.

INT0 ___/‾‾‾_____→  INT0 interrupt routine
                                                        ⋮
                                                       RETI
820FH    LD    XX, XX
```
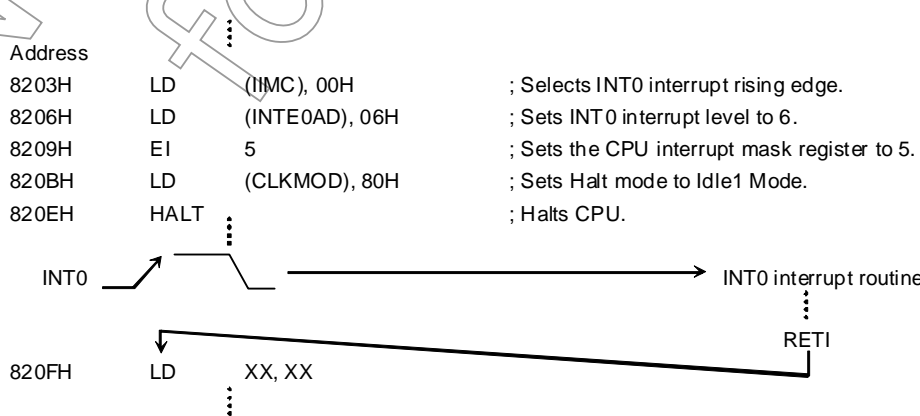
(3) Operation in each mode

1. IDLE2 Mode

In IDLE2 mode, the system clock is supplied only to the built-in I/O blocks specified with the built-in I/O operation control bits and the CPU stops executing instructions.

Figure 3.3.3 shows an example timing for releasing a halt state using an interrup.
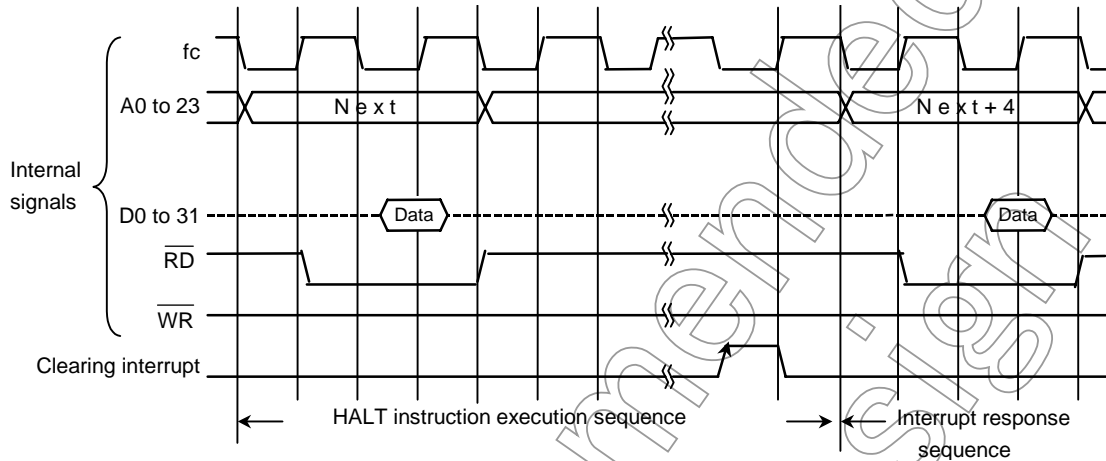
Figure 3.3.3 Timing chart for Idle2 Mode Halt state cleared by interrupt

2. IDLE1 mode

In IDLE1 mode, only the internal oscillator operates with the system clock for the CPU stopped.

In the halt state, interrupt request sampling is performed asynchronously to the system clock. The halt state is, however, released in sync with the system clock.

Figure 3.3.4 shows an example timing for releasing a halt state using an interrupt.
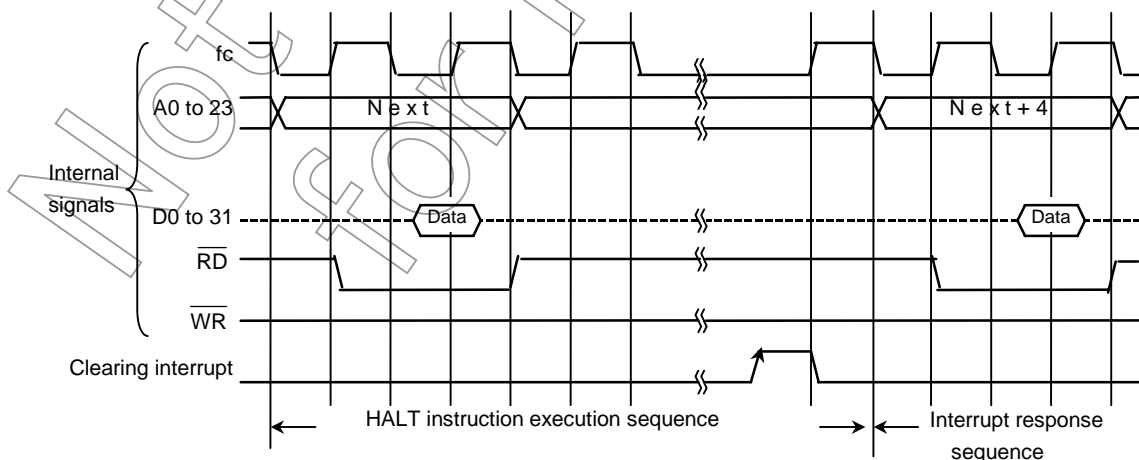
Figure 3.3.4 Timing chart for Idle1 Mode Halt state cleared by interrupt

3. IDLE3 mode

In IDLE3 mode, all internal circuits other than the low-speed oscillator and RTC, including the high-speed oscillator, are stopped. The pin states in IDLE3 mode depends on the setting of WDMOD<DRVE>. Table 3.3.5 shows the states of pins in IDLE3 mode.

The halt state in IDLE3 mode can be released with an external interrupt using the $\overline{\text{NMI}}$, INT0, or WUINT0 to 7 pin (INT0 interrupt), an internal interrupt using INTRTC, or a reset.

Upon released from the halt state, the system clock output starts after the high-speed oscillator warm-up time and clock multiplier settling time elapse.

The warm-up time for the high-speed oscillator is counted using the warm-up counter within the TMP92CD54I. Once that counting ends, the device starts counting the settling time for the clock multiplier. This mechanism results in the high-speed oscillator warm-up time (1.6 ms) being included in the time required between the halt release signal being input and the system clock being output even in a system using an external oscillator that does not need oscillation settling time.

When using a reset to release a halt, ensure that the reset signal is held Low until the high-speed oscillator operates stably.

Figure 3.3.5 shows an example timing for releasing a halt state using an interrupt.
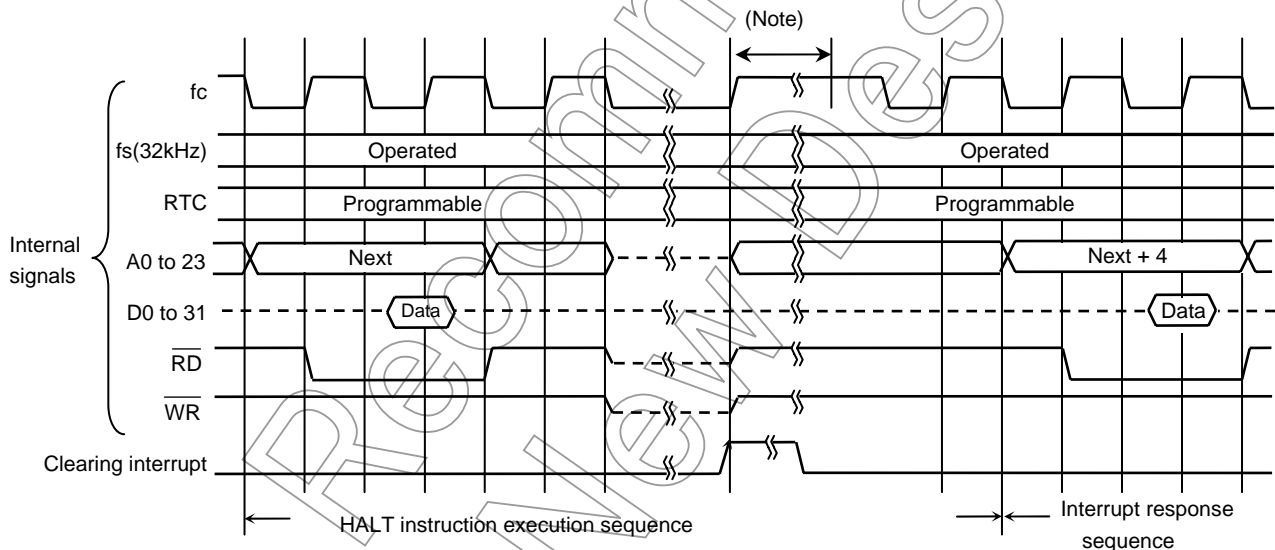
(Note); Once the halt state is released, interrupt handling starts after the oscillator startup time (t_{STA}), warm-up time (approx. 1.6 ms) and clock multiplier settling time (approx. 1.6 ms) elapse. For details of the startup time (t_{STA}), contact the oscillator manufacturer..

Figure 3.3.5 Timing chart for Idle3 Mode Halt state cleared by interrupt

4. STOP mode

In STOP mode, all internal circuits are stopped. The pin states in STOP mode depends on the setting of WDMOD<DRVE>. Table 3.3.5 shows the states of pins in STOP mode.

The halt state in STOP mode can be released with an external interrupt using the $\overline{\text{NMI}}$, INT0, or WUINT0 to 7 pin (INT0 interrupt) or a reset.

Upon released from the halt state, the system clock output starts after the high-speed oscillator warm-up time and clock multiplier settling time elapse.

The warm-up time for the high-speed oscillator is counted using the warm-up counter within the TMP92CD54I. Once that counting ends, the device starts counting the settling time for the clock multiplier. This mechanism results in the high-speed oscillator warm-up time (1.6 ms) being included in the time required between the halt release signal being input and the system clock being output even in a system using an external oscillator that does not need oscillation settling time.

When using a reset to release a halt, ensure that the reset signal is held Low until the high-speed oscillator operates stably.

In STOP mode, the value of the RTCFC register is initialized even if the halt is released with an interrupt. It is, therefore, necessary to re-set RTCFC after releasing the halt.
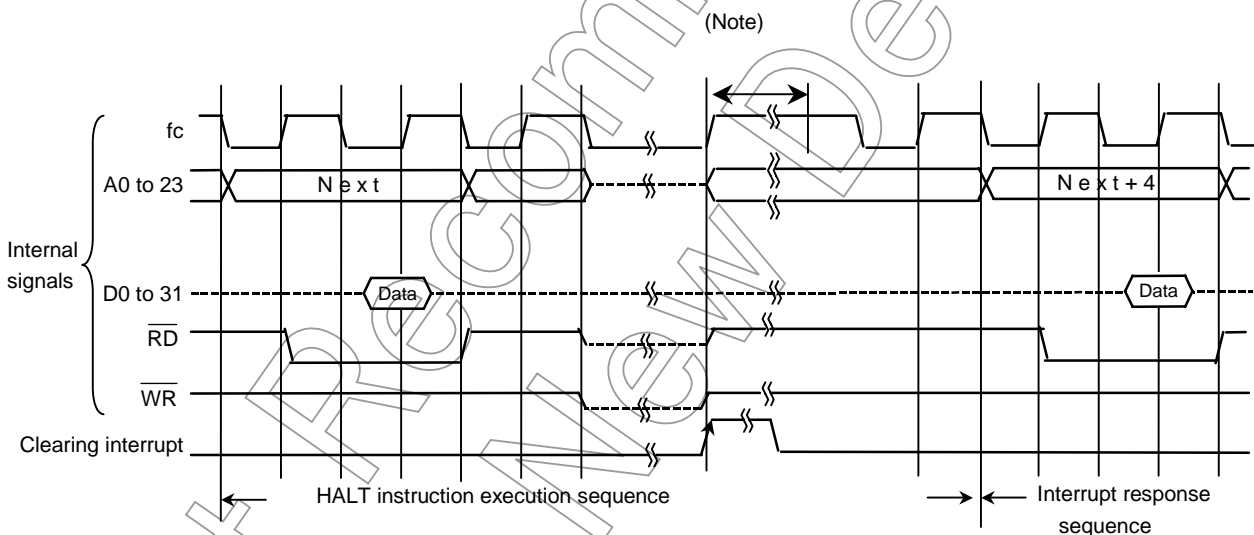
.

(Note)

(Note); Once the halt state is released, interrupt handling starts after the oscillator startup time (t_{STA}), warm-up time (approx. 1.6 ms) and clock multiplier settling time (approx. 1.6 ms) elapse. For details of the startup time (t_{STA}), contact the oscillator manufacturer..

Figure 3.3.6 Timing chart for Stop Mode Halt state cleared by interrupt

Table 3.3.4   Warming-up time and clock doubler stable time after clearance of Stop Mode and Idle3 Mode
(@ fc=20MHz)

| Warm-up time | 1.6 ms ($2^{14}/f_{OSC}$) |
| Clock doubler stable time | 1.6 ms ($2^{14}/f_{OSC}$) |

fc = 2 × f$_{OSC}$

Table 3.3.5 Pin states in IDLE3 and STOP Mode

Pin Names	I/O	\<DRVE\> = 0	\<DRVE\> = 1
P00 to 07	Input Mode	Invalid	
	Output Mode	Output	
	D0 to D7	High-Z	
P40 to 47/A0 to 7	Input Mode	Invalid	
	Output Mode	High-Z	Output
P70,P71,P73 to 75/	Input Mode	Invalid	
$\overline{RD}$, $\overline{WR}$, $\overline{CS}$ to $\overline{WAIT}$	Output Mode	High-Z	Output
P72/SI2/SCL2	Input Mode	Input	
	Output Mode	Input	Output
PC0 to PC5/TI0 to TO7	Input Mode	Invalid	
	Output Mode	High-Z	Output
PD0 to PD7/TI8 to TOB	Input Mode	Input	
	Output Mode	High-Z	Output
	WUINT0 to 7	Input	
PF0 to PF7/TXD0 to RX	Input Mode	Invalid	
	Output Mode	High-Z	Output
PG0 to PG7/AN0 to AN7	Input Mode	Invalid	
PL0 to PL3/AN8 to AN11	Input Mode	Invalid	
PM0 to PM4	Input Mode	Invalid	
/ $\overline{SS}$ to SCK2	Output Mode	High-Z	Output
PN0 to PN6	Input Mode	Invalid	
/SCK0 to SO2&SDA2	Output Mode	High-Z	Output
$\overline{NMI}$	Input	Input	
INT0	Input	Input	
$\overline{RESET}$	Input	Input	
AM0, AM1	Input	Input	
TEST0, TEST1	Input	Input	
X1	Input	Invalid	
X2	Output	H Level Output	
XT1	Input	Invalid (STOP) Operate (IDLE3, RTCFC\<XTEN\>=1)	
XT2	Output	H Level Output (STOP) Operate (IDLE3, RTCFC\<XTEN\>=1)	
CLK	Output	H level output (CLKMOD\<CLKOE\>=0) L level output (CLKMOD\<CLKM1:0\>=00) H or L level Output (CLKMOD\<CLKM1:0\>=10)	

Input : The input gate is functioning. Apply a Low or High level to prevent the input pin from floating

Output : Placed in the output state

Invalid : The input is invalid.

High-Z : High impedance.

Note) when RTCFC\<XTEN\>=1.

3.4 Interrupts

Interrupts for the TLCS-900/H1 are controlled using the CPU interrupt mask flip-flop (SR<IFF2:0>) and the interrupt controller.

The TMP92CD54I supports the following 60 interrupt sources:

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

Internal interrupts: 42 sources
- Internal I/O interrupts: 34 sources
- Micro DMA Transfer End interrupts: 8 sources

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

Each interrupt source is assigned a unique interrupt vector number (fixed) and each maskable interrupt can be assigned one of six priority levels (variable). Nonmaskable interrupts have a fixed priority level of 7 (top priority).

When an interrupt occurs, the interrupt controller sends the priority level of that interrupt source to the CPU. When more than one interrupt occurs simultaneously, it sends the highest priority level to the CPU (the highest possible level is 7 for nonmaskable interrupts).

The CPU compares the sent priority level with the value in the CPU interrupt mask register (IFF2:0) and, if the priority level is higher than the interrupt mask register setting, accepts the interrupt.

Software interrupts and undefined instruction execution interrupts, however, occur independently of the IFF2:0 setting.

The value of the interrupt mask register SR<IFF2:0> can be modified using the EI instruction (EI num, where num specifies the contents of SR<IFF2:0>). For example, programming "EI 3" enables maskable interrupts having a priority level of 3 or higher, as specified with the interrupt controller, and nonmaskable interrupts to be accepted. Executing the EI or "EI 0" instruction enables all nonmaskable interrupts and maskable interrupts having a priority level of 1 or higher to be accepted. (Therefore, they are equivalent to "EI 1".)

The DI instruction (specifying 7 for SR<IFF2:0>) is functionally equivalent to "EI 7" and used to disable the acceptance of maskable interrupts because they have priority levels of 0 to 6. The EI instruction is effective immediately after it is executed.

Interrupts for the TLCS-900/H1 also supports micro DMA handling mode in addition to the general-purpose interrupt handling mode described above. In micro DMA mode, the CPU automatically transfers data (1, 2, or 4 bytes). It enables fast data transfer to internal/external memory and built-in I/O.

Moreover, the TMP92CD54I supports a soft start function, which enables software to issue a micro DMA request, rather than given from an interrupt source.

Figure 3.4.1 shows the entire interrupt handling flow.
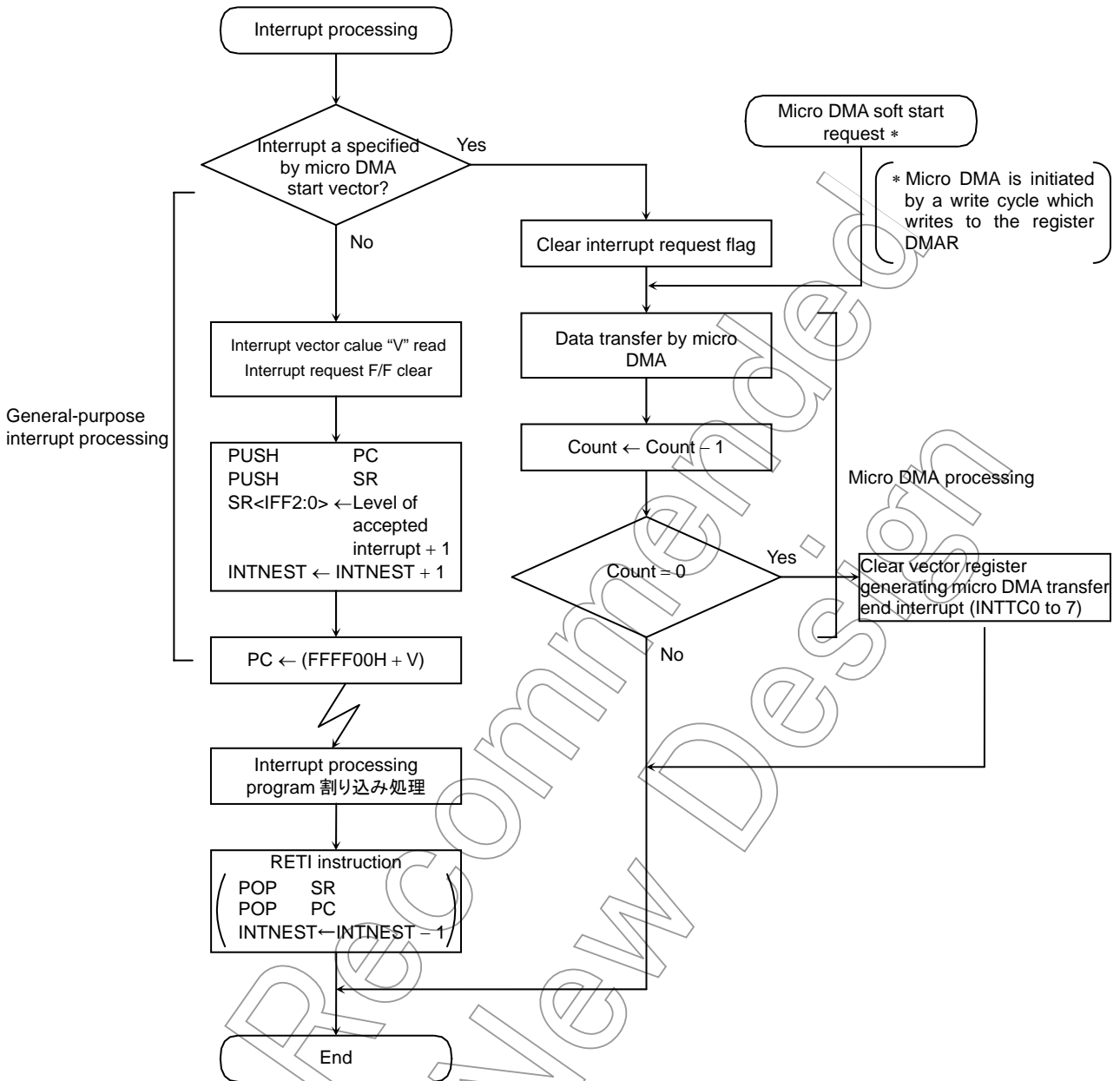
Figure 3.4.1 Interrupt and Micro DMA Processing Sequence

3.4.1　General-purpose interrupt handling

When the CPU accepts an interrupt, it performs the following operation. For software interrupts issued by the CPU and undefined instruction execution interrupts, the CPU only performs steps (2), (4), and (5) without executing steps (1) and (3). The following steps are similar to those for the TLCS-900/L, TLCS-900/H, TLCS-900/L1, and TLCS-900/H2.

(1) The CPU reads an interrupt vector from the interrupt controller.
If two or more interrupts having the same priority level occur simultaneously, the CPU issues an interrupt vector according to the default priorities (fixed: smaller vector values have higher priority) and clears the interrupt request.

(2) The CPU pushes the program counter (PC) and status register (SR) into the stack area (pointed to by XSP).

(3) Set the CPU interrupt mask register SR<IFF2:0> to the value of the accepted interrupt level plus one. If the value is 7, however, the value of 7 is set without being incremented.

(4) Increment the interrupt nesting counter INTNEST by one.

(5) The CPU jumps to the address indicated with the data at address (FFFF00H + interrupt vector) and then starts the interrupt handling routine.

Upon the completion of interrupt handling, usually use the RETI instruction to return to the main routine. This instruction restores the values of the program counter (PC) and status register (SR) from the stack and then decrement the interrupt nesting counter (INTNEST) by one.

The acceptance of nonmaskable interrupts cannot be disabled programmatically. On the other hand, maskable interrupts can be enabled or disabled programmatically and can be assigned priorities for each interrupt source (an interrupt level of 0 or 7 disables the interrupt request). The CPU accepts an interrupt if the priority level of the interrupt request is higher than the value of its interrupt mask register SR<IFF2:0>. It then sets the mask register <IFF2:0> to the value of the accepted priority plus one. Therefore, if any interrupt having a priority higher than the interrupt currently being handled occurs, the CPU accepts that interrupt request, resulting in interrupt handling being nested.

If another interrupt request is issued while the CPU is performing steps (1) to (5) above for an interrupt it has accepted, the new interrupt request is sampled immediately after the first instruction of the interrupt handling routine is executed. The DI instruction can be used as the first instruction to prohibit the nesting of maskable interrupts.

Upon a reset, the CPU mask register SR<IFF2:0> is initialized to 7, which disables maskable interrupts.

In the TMP92CD54I, memory addresses FFFF00H to FFFFEFH (240 bytes) are assigned to the interrupt vector area. Table 3.4.1 shows the interrupt table.

Table 3.4.1 TMP92CD54I 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	
-	Maskable	Micro DMA	-	-	-
11		INT0: INT0 pin input (Note2)	0028H	FFFF28H	0AH
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	20H (Note3)
34		INTTX0: Serial transmission (Channel 0)	0084H	FFFF84H	21H
35		INTRX1: Serial receive (Channel 1)	0088H	FFFF88H	22H (Note3)
36		INTTX1: Serial transmission (Channel 1)	008CH	FFFF8CH	23H
37		INTCR: CAN receive	0090H	FFFF90H	24H (Note3)
38		INTCT: CAN transmission	0094H	FFFF94H	25H (Note3)
39		INTCG: CAN global	0098H	FFFF98H	26H (Note3)
40		INTSEM: SEI mode fault error	009CH	FFFF9CH	27H (Note3)
41		INTSEE: SEI transfer end / slave error	00A0H	FFFFA0H	28H (Note3)
42		INTSER: SEI receive	00A4H	FFFFA4H	29H
43		INTSET: SEI transmission	00A8H	FFFFA8H	2AH
44		INTRTC: Read Time Counter	00ACH	FFFFACH	2BH
45		(reserved)	00B0H	FFFFB0H	-
46		INTSBE2: SBI I2CBUS transfer end (Channel 2)	00B4H	FFFFB4H	2DH
47		INTSBS2: SBI I2CBUS stop condition (Channel 2)	00B8H	FFFFB8H	2EH
48		INTSBE0: SBI 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

Default Priority	Type	Interrupt Source and Source of Micro DMA Request	Vector Value	Address refer to Vector	Micro DMA Start Vector
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 to 00FCH	FFFFF0H to FFFFFCH	− to −

Note1: To start micro DMA, select edge detection mode.

Note2: Micro DMA processing cannot be assigned.

Note3: If an interrupt occurs with an interrupt source specified for micro DMA, it is given the highest priority among maskable interrupts (independently of the default priority assigned to each channel).

Note4: The above table lists only start addresses. Each vector consists of four bytes.

3.4.2 Micro DMA

The TMP92CD54I supports the micro DMA function. Interrupt requests specified for the micro DMA function are handled with the highest priority among maskable interrupts regardless of the set interrupt level.

Eight channels are provided for micro DMA and support continuous transfer using a burst specification, described later.

（1） Micro DMA operation

When an interrupt request specified with the micro DMA start vector register is issued, the micro DMA function transfers data to the CPU with the highest priority among maskable interrupts regardless of the interrupt level assigned to the interrupt source. If SR<IFF2:0> = 7, micro DMA requests are not accepted.

The micro DMA function supports eight channels; micro DMA can be specified for up to eight types of interrupt source simultaneously.

When micro DMA is accepted, the function clears the interrupt request flag assigned to that channel, performs a single data transfer (1, 2, or 4 bytes) from the source address to the destination address, as set in the control register, and then decrements the transferred data counter. If the counter becomes 0 after being decremented, the following operation is performed:

- The CPU notifies the interrupt controller of the completion of micro DMA transfer.
- The interrupt controller issues a micro DMA transfer completion interrupt (INTTCn).
- The micro DMA start vector register, DMAnV, is cleared to 0, thus disabling the start of subsequent micro DMA.
- Micro DMA processing is completed.

If the counter is not 0 after being decremented, micro DMA processing is terminated unless a burst is specified as described later. In that case, a transfer completion interrupt (INTTCn) does not occur.

If an interrupt source is used only to start micro DMA, the interrupt level for that source should be set to 0. This is because, if that interrupt request is issued before the micro DMA start vector is set, the CPU performs general-purpose interrupt handling if the interrupt level is 1 to 6.

If micro DMA interrupts are shared with general-purpose interrupts, interrupt sources used to start micro DMA should have an interrupt level lower than those for all other interrupt sources.

The priority of a micro DMA transfer completion interrupt is determined according to the interrupt level and default priority, in the same way as with other maskable interrupts.

If micro DMA requests for two or more channels are issued simultaneously, requests for lower channel numbers are given higher priorities (CH0 is the highest and CH7 is the lowest), regardless of their interrupt levels.

The registers that specify the transfer source and destination addresses are 32-bit control registers. The micro DMA function, however, handles 16 Mbytes of space because there are only 24 address output lines.

Three transfer modes, 1, 2 and 4 bytes, are supported. For each transfer mode, the transfer source and destination addresses can be incremented, decremented or fixed after transfer. This facilitates data transfer from memory to memory, from I/O to memory, from memory to I/O, and from I/O to I/O. For details of transfer modes, see "(4) Details of transfer mode registers."

The transferred data counter consists of 16 bits so that up to 65536 micro DMA transfers can be performed for a single interrupt source (the maximum number is allowed when the initial value of the counter is 0000H).

Micro DMA supports 44 types of interrupt sources: 43 sources listed in Table 3.4.1 with micro DMA start vectors, plus soft start.

Figure 3.4.2 shows micro DMA cycles in transfer address INC mode (similar in other modes, except counter mode) (with an 8-bit external bus, 0 waits, and even-numbered source and destination addresses).
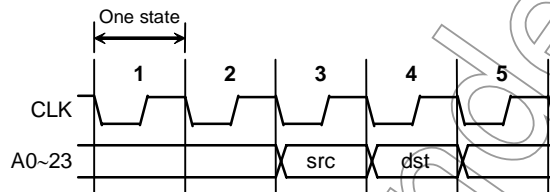
Figure 3.4.2 Timing for Micro DMA Cycle

1st and 2nd states:	Instruction fetch cycle (prefetch the next instruction code)
	If the instruction queue buffer is full, this cycle becomes a dummy cycle.
3rd state:	Micro DMA read cycle
4th state:	Micro DMA write cycle
5th state:	(Same as 1st and 2nd states)

(2) Soft start function

The TMP92CD54I supports a micro DMA soft start function, which starts micro DMA when a DMAR register write cycle occurs rather than when an interrupt request is issued.

Specifically, a write of 1 to a bit in the DMAR register can start a single micro DMA transfer. Upon the completion of transfer, the DMAR register bit corresponding to the transferred channel is automatically cleared to 0.

Rewriting a 1 to the DMAR register can perform a soft start again unless the micro DMA transfer counter is 0.

If a burst is specified with the DMAR register, once micro DMA is started, data is transferred continuously until the micro DMA transfer counter becomes 0.

Symbol	NAME	Address	7	6	5	4	3	2	1	0
DMAR	DMA Request	109h (no RMW)	DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
			R/W							
			0	0	0	0	0	0	0	0

RMW prohibited: A read-modify-write operation cannot be performed.

Figure 3.4.3 Micro DMA Request Register

(3) Transfer control registers

The following registers specify the transfer source and destination addresses. The LDC cr,r instruction is used to set data in these registers.
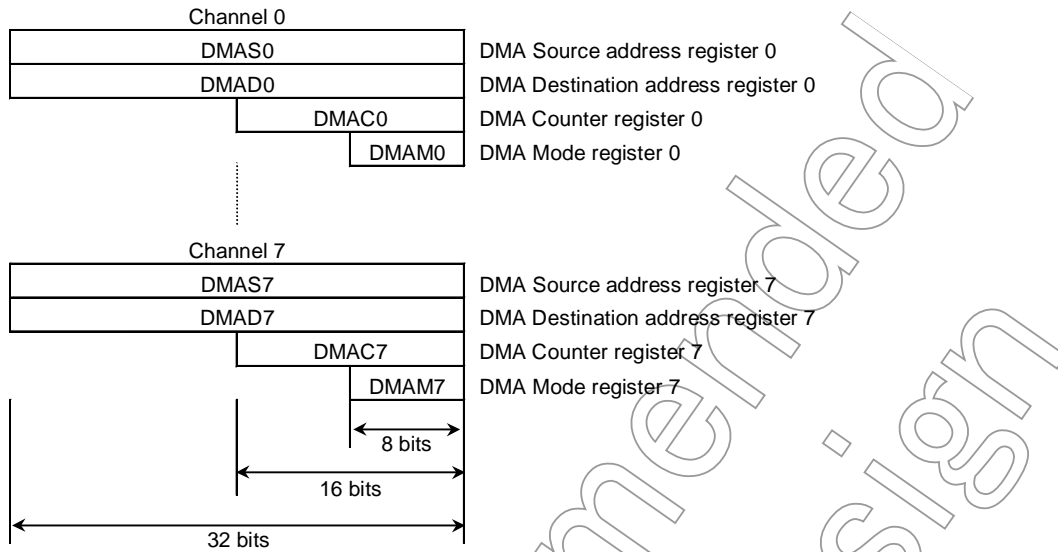
Figure 3.4.4 Micro DMA Transfer Register

(4) Details of transfer mode registers

| 0 | 0 | 0 | Mode | | DMAM0 to 7 |
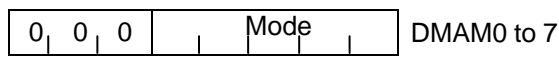
DMAM[4:0]	Mode Description	Execution time

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)

Note1: The execution times shown above are best-case values (assuming that memory access is completed in a single clock cycle).
1 state = 50 ns (when fc = 20 MHz)

Note2: n indicates 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)

Figure 3.4.5 Details of Transfer Mode Registers

3.4.3   Control by the interrupt controller

Figure 3.4.16 shows a block diagram of the interrupt circuit. The left half of the figure represents the interrupt controller while the right half represents the CPU interrupt request signal circuit and halt release circuit.

The interrupt controller has an interrupt request flag, interrupt priority setup register and micro DMA start vector setup register for each interrupt channel (51 channels in total). The interrupt request flag is used to latch an interrupt request from a peripheral device.

This flag is cleared under the following conditions:

- Upon a reset

- When the CPU accepts the interrupt and reads the vector for that interrupt.

- When the instruction that clears the interrupt is executed (the micro DMA start vector for the interrupt source to be cleared is written to the INTCLR register).

- The CPU accepts a micro DMA request for that interrupt.

- The micro DMA burst transfer for that interrupt is completed.

Interrupt priorities can be set by writing them to the interrupt priority setup registers (INTE0AD, INTE12, and so on) that are provided for each interrupt source. One of six interrupt levels, 1 to 6, can be set. Writing a priority level of 0 (or 7) causes the corresponding interrupt request to be disabled. Nonmaskable interrupts (NMI pin) have a fixed priority level of 7.

If more than one interrupt request having the same priority level occurs simultaneously, the CPU accepts an interrupt according to the default priorities (lower priority value = smaller vector). Reading bits 3 and 7 in the interrupt priority setup register returns the state of the interrupt request flag, which indicates whether an interrupt request has been issued for each channel.

Among the interrupts that have occurred simultaneously, the interrupt controller sends the highest interrupt priority level and its vector address to the CPU. The CPU compares the sent interrupt level with the interrupt mask register value <IFF2:0> in the status register and, if the sent level is higher, accepts the interrupt. The CPU then sets the accepted interrupt level + 1 in its SR<IFF2:0> so that only interrupt requests having a priority higher than that can be accepted in a nested manner. Upon the completion of interrupt handling (execution of the RETI instruction), the CPU restores the value of the interrupt mask register existing before the interrupt occurred in SR<IFF2:0>.

The interrupt controller has registers (eight channels) that store micro DMA start vectors. Writing a start vector (see Table 3.4.1) in this register enables micro DMA to start when the corresponding interrupt request is issued. It is necessary to set values in the micro DMA parameter registers (such as DMAS and DMAD) before enabling micro DMA processing.
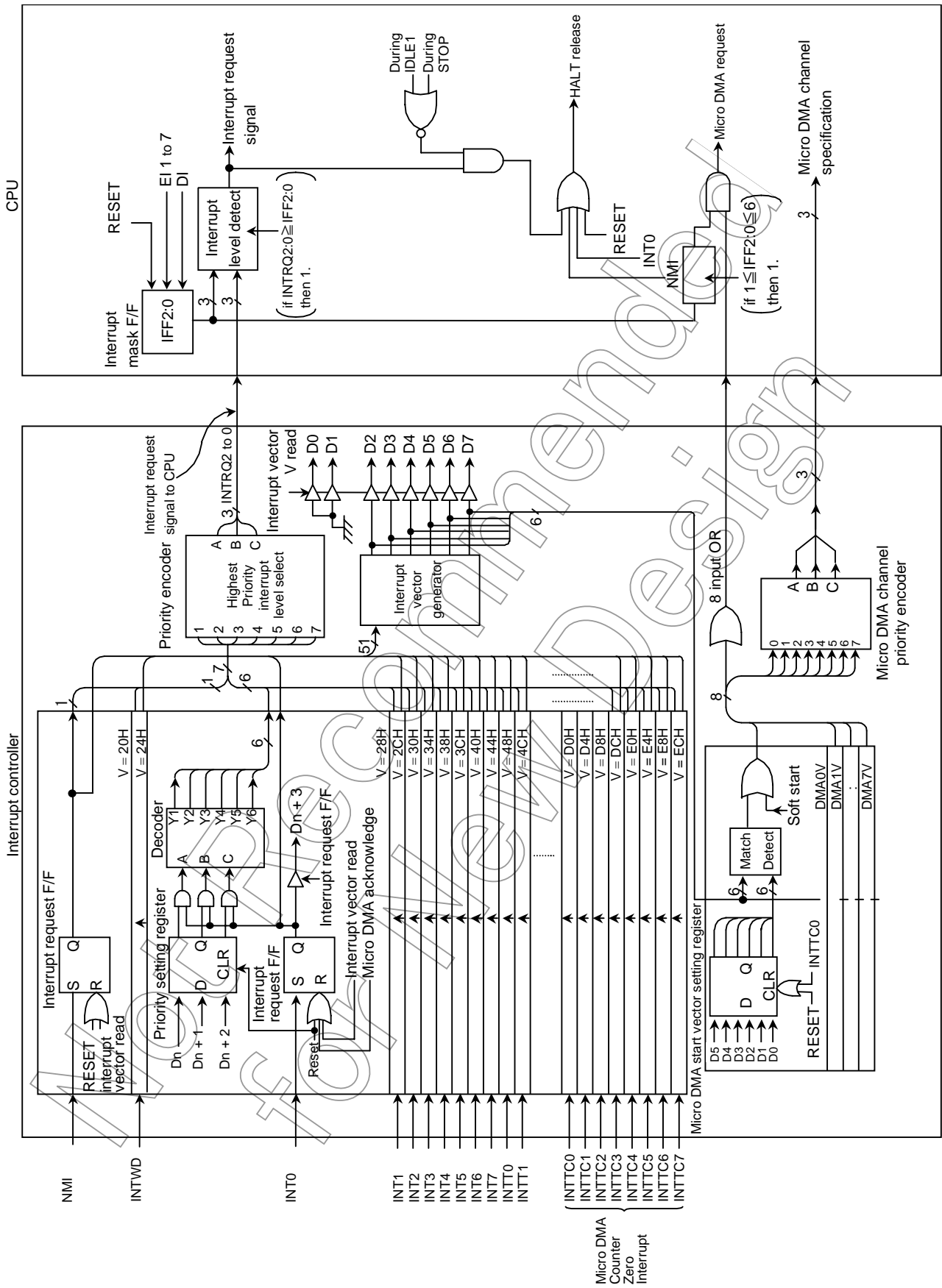
Figure 3.4.6 Block Diagram of Interrupt Controller

(1) Interrupt priority setup registers

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 & INTAD Enable	F0h	INTAD				INT0			(Note)
			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	D7h	-	-	-	-	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

Note 1: If any bit of WUPMASK<WMK7:0> is set to 1, the input signal from the external INT0 pin is disabled. To use the external INT0 pin, write 00H to WUPMASK<WMK7:0> to disable the wakeup interrupt function.

Figure 3.4.7 Interrupt Priority Setup Registers (1/3)

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
INTECRT	INTCR & INTCT Enable	DDh	INTCT				INTCR			
			ICTC	ICTM2	ICTM1	ICTM0	ICRC	ICRM2	ICRM1	ICRM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTECG	INTCG Enable	DEh	-	-	-	-	INTCG			
							ICGC	ICGM2	ICGM1	ICGM0
							R	R/W		
			-	-	-	-	0	0	0	0*
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
INTERTC	INTRTC Enable	E1h	-	-	-	-	INTRTC			
							IRTCC	IRTCM2	IRTCM1	IRTCM0
							R	R/W		
			-	-	-	-	0	0	0	0
INTESB2	INTSBE2 & INTSBS2 Enable	E2h	INTSBS2				INTSBE2			
			ISBS2C	ISBS2M2	ISBS2M1	ISBS2M0	ISBE2C	ISBE2M2	ISBE2M1	ISBE2M0
			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
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

Figure 3.4.8 Interrupt Priority Setup Registers (2/3)

Symbol	NAME	Address	7	6	5	4	3	2	1	0
			NMI				INTWD			
INTNMWDT	NMI & INTWD Enable	F7h	INMIC	-	-	-	IWDC	-	-	-
			R				R			
			0	-	-	-	0	-	-	-

Interrupt request flag

IxxM2	LxxM1	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

Note: To modify an interrupt priority setup register, first execute the DI instruction to disable the acceptance of interrupts.

Figure 3.4.9 Interrupt Priority Setup Registers (3/3)

(2) Controlling external interrupts

Symbol	NAME	Address	7	6	5	4	3	2	1	0
			-	-	-	-	-	-	I0LE	NMIREE
									R/W	
IIMC	Interrupt Input Mode Control	F6H (no RMW)	-	-	-	-	-	-	0	0
									INT0 mode 0:edge mode 1:level mode	NMI̅ mode 0:Falling edge 1:Falling & rising edges

INT0 Level Enable

0	Rising edge detect INT
1	"H" level INT

NMI̅ rising edge Enable

0	INT request generation at falling edge
1	INT request generation at rising and falling edge

Note 1: To switch the INT0 pin mode from level to edge (IIMC<I0LE> from 1 to 0), first disable INT0. In that case, execute EI instruction after three cycles (after three NOP instructions are executed).

Example settings: The following shows an example of settings for switching the INT0 interrupt from level to edge mode.

```
DI                              ; Disable interrupt
LD      (IIMC), XXXXXX0-B       ; Switch from level to edge
LD      (INTCLR), 0AH           ; Clear interrupt request flag
NOP                             ; Wait for 3 cycles
NOP
NOP
EI                              ; Enable interrupt    X = Don't care  "-" = No change.
```

Note 2: The input pulse width for an external interrupt must satisfy the specification. For details, see "4. Electrical Characteristics."

Figure 3.4.10 Controlling External Interrupts

Table 3.4.2 Settings of External Interrupt Pin Function

Interrupt	Pin name	Mode		Setting method
$\overline{\text{NMI}}$	$\overline{\text{NMI}}$		Falling Edge	IIMC<NMIREE> = 0
			Falling and Rising 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
WUINT0	PD0		Falling and Rising Edges	WUPMOD<WMD0> = 0
			Falling Edge	WUPMOD<WMD0> = 1 and WUPEDGE<WED0> = 0
			Rising Edge	WUPMOD<WMD0> = 1 and WUPEDGE<WED0> = 1
WUINT1	PD1		Falling and Rising Edges	WUPMOD<WMD1> = 0
			Falling Edge	WUPMOD<WMD1> = 1 and WUPEDGE<WED1> = 0
			Rising Edge	WUPMOD<WMD1> = 1 and WUPEDGE<WED1> = 1
WUINT2	PD2		Falling and Rising Edges	WUPMOD<WMD2> = 0
			Falling Edge	WUPMOD<WMD2> = 1 and WUPEDGE<WED2> = 0
			Rising Edge	WUPMOD<WMD2> = 1 and WUPEDGE<WED2> = 1
WUINT3	PD3		Falling and Rising Edges	WUPMOD<WMD3> = 0
			Falling Edge	WUPMOD<WMD3> = 1 and WUPEDGE<WED3> = 0
			Rising Edge	WUPMOD<WMD3> = 1 and WUPEDGE<WED3> = 1
WUINT4	PD4		Falling and Rising Edges	WUPMOD<WMD4> = 0
			Falling Edge	WUPMOD<WMD4> = 1 and WUPEDGE<WED4> = 0
			Rising Edge	WUPMOD<WMD4> = 1 and WUPEDGE<WED4> = 1
WUINT5	PD5		Falling and Rising Edges	WUPMOD<WMD5> = 0
			Falling Edge	WUPMOD<WMD5> = 1 and WUPEDGE<WED5> = 0
			Rising Edge	WUPMOD<WMD5> = 1 and WUPEDGE<WED5> = 1
WUINT6	PD6		Falling and Rising Edges	WUPMOD<WMD6> = 0
			Falling Edge	WUPMOD<WMD6> = 1 and WUPEDGE<WED6> = 0
			Rising Edge	WUPMOD<WMD6> = 1 and WUPEDGE<WED6> = 1
WUINT7	PD7		Falling and Rising Edges	WUPMOD<WMD7> = 0
			Falling Edge	WUPMOD<WMD7> = 1 and WUPEDGE<WED7> = 0
			Rising Edge	WUPMOD<WMD7> = 1 and WUPEDGE<WED7> = 1

(3) Interrupt request flag register

The interrupt request flag can be cleared by writing a micro DMA start vector, listed in Table 3.4.1, to the INTCLR register.

To clear the INT0 interrupt flag, for example, perform the following register operation after the DI instruction:

INTCLR ← 0AH ; Clear INT0 interrupt request flag

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt Clear control	F8H (no RMW)	–	–	–	–	–	–	–	–
			W							
			0	0	0	0	0	0	0	0
			Interrupt Vector							

Figure 3.4.11 Interrupt Request Flag Register

(4) Micro DMA start vector registers

The micro DMA start vector registers are used to select the interrupt sources to which micro DMA processing is assigned. Interrupt sources having micro DMA start vectors that match the vector values in the registers are assigned as micro DMA start sources.

When the micro DMA transfer counter becomes 0, the interrupt controller is notified of a micro DMA transfer completion interrupt for that channel and the corresponding micro DMA start vector register is cleared, causing the micro DMA start source for the channel to be cleared. To continue micro DMA processing, therefore, it is necessary to set the micro DMA start vector register again while the micro DMA transfer completion interrupt is handled.

If the same vector is set in micro DMA start vector registers for more than one channel, smaller channels take precedence.

If the same vector is set in micro DMA start vector register for two channels, therefore, micro DMA for the channel having the smaller number is performed until the micro DMA transfer completion interrupt is issued, after which micro DMA is started for the larger-number channel unless the micro DMA start vector for the smaller-number channel is set again.

Symbol	NAME	Address	7	6	5	4	3	2	1	0
DMA0V	DMA0 Start Vector	100h (no RMW)	-	-	DMA0 Start Vector					
			-	-	DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
			-	-	0	0	0	0	0	0
DMA1V	DMA1 Start Vector	101h (no RMW)	-	-	DMA1 Start Vector					
			-	-	DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
			-	-	0	0	0	0	0	0
DMA2V	DMA2 Start Vector	102h (no RMW)	-	-	DMA2 Start Vector					
			-	-	DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
			-	-	0	0	0	0	0	0
DMA3V	DMA3 Start Vector	103h (no RMW)	-	-	DMA3 Start Vector					
			-	-	DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
			-	-	0	0	0	0	0	0
DMA4V	DMA4 Start Vector	104h (no RMW)	-	-	DMA4 Start Vector					
			-	-	DMA4V5	DMA4V4	DMA4V3	DMA4V2	DMA4V1	DMA4V0
					R/W					
			-	-	0	0	0	0	0	0
DMA5V	DMA5 Start Vector	105h (no RMW)	-	-	DMA5 Start Vector					
			-	-	DMA5V5	DMA5V4	DMA5V3	DMA5V2	DMA5V1	DMA5V0
					R/W					
			-	-	0	0	0	0	0	0
DMA6V	DMA6 Start Vector	106h (no RMW)	-	-	DMA6 Start Vector					
			-	-	DMA6V5	DMA6V4	DMA6V3	DMA6V2	DMA6V1	DMA6V0
					R/W					
			-	-	0	0	0	0	0	0
DMA7V	DMA7 Start Vector	107h (no RMW)	-	-	DMA7 Start Vector					
			-	-	DMA7V5	DMA7V4	DMA7V3	DMA7V2	DMA7V1	DMA7V0
					R/W					
			-	-	0	0	0	0	0	0

Figure 3.4.12 Micro DMA Start Vector Registers

(5) Micro DMA burst specification

A burst specification allows data to be transferred continuously with a single micro DMA start until the transfer count register becomes zero. A burst can be specified by writing a 1 to the bit corresponding to the micro DMA channel in the DMAB register.

Symbol	NAME	Address	7	6	5	4	3	2	1	0
DMAB	DMA Burst	108h (no RMW)	DBST7	DBST6	DBST5	DBST4	DBST3	DBST2	DBST1	DBST0
			R/W							
			0	0	0	0	0	0	0	0

Figure 3.4.13 Micro DMA Burst Specification

(6) Precautions

This CPU consists of an instruction execution unit and a bus interface unit. If an instruction that clears the interrupt request flag for the interrupt controller is executed immediately before a corresponding interrupt occurs, the CPU may execute the instruction clearing the interrupt request flag(Note) before it reads the interrupt vector after accepting the interrupt. In such a case, the CPU reads the source lost vector "0004H" (shared with SWI1) and then reads the interrupt vector at address FFFF04H.

To avoid the above situation, any instruction that clears an interrupt request flag should be placed after the DI instruction. To change the interrupt request level to 0, first clear the corresponding interrupt request using the INTCLR instruction before setting the interrupt request level to 0.

In addition, note that the following two interrupts are different from other interrupt circuits:

INT0 Level Mode	In Level Mode INT0 is not an edge-triggered interrupt. Hence, in Level Mode the interrupt request flip-flop 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 Mode 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. Also EI instruction should be execuse after waiting 3-cycle.
	
	DI
	LD (IIMC), 00H ; Switches from level to edge.
	LD (INTCLR), 0AH ; Clears interrupt request flag.
	NOP ; Wait 3-cycle
	NOP
	NOP
	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 and pin state change are also equivalent to an instruction that clears an interrupt request flag:

INT0:   Instruction that switches to level mode after an interrupt request is issued in edge mode
        Change in pin input state (High to Low) after an interrupt request is issued in level mode

INTRX: Instruction that reads the receive buffer

3.4.4 Interrupt mask registers

The TMP92CD54I contains interrupt mask registers. Unlike the interrupt priority setup registers, the interrupt mask registers only enable or disable interrupt handling. If an interrupt source is disabled in the interrupt mask register, interrupts for that source will not occur even if it is enabled in the interrupt priority setup register. Interrupt mask registers can disable more than one interrupt source simultaneously.

Upon a reset, all bits in the interrupt mask registers are initialized to 1 (enable interrupts). To disable interrupts using the interrupt mask registers, it is necessary to write a 0 to the bit corresponding to the interrupt source.
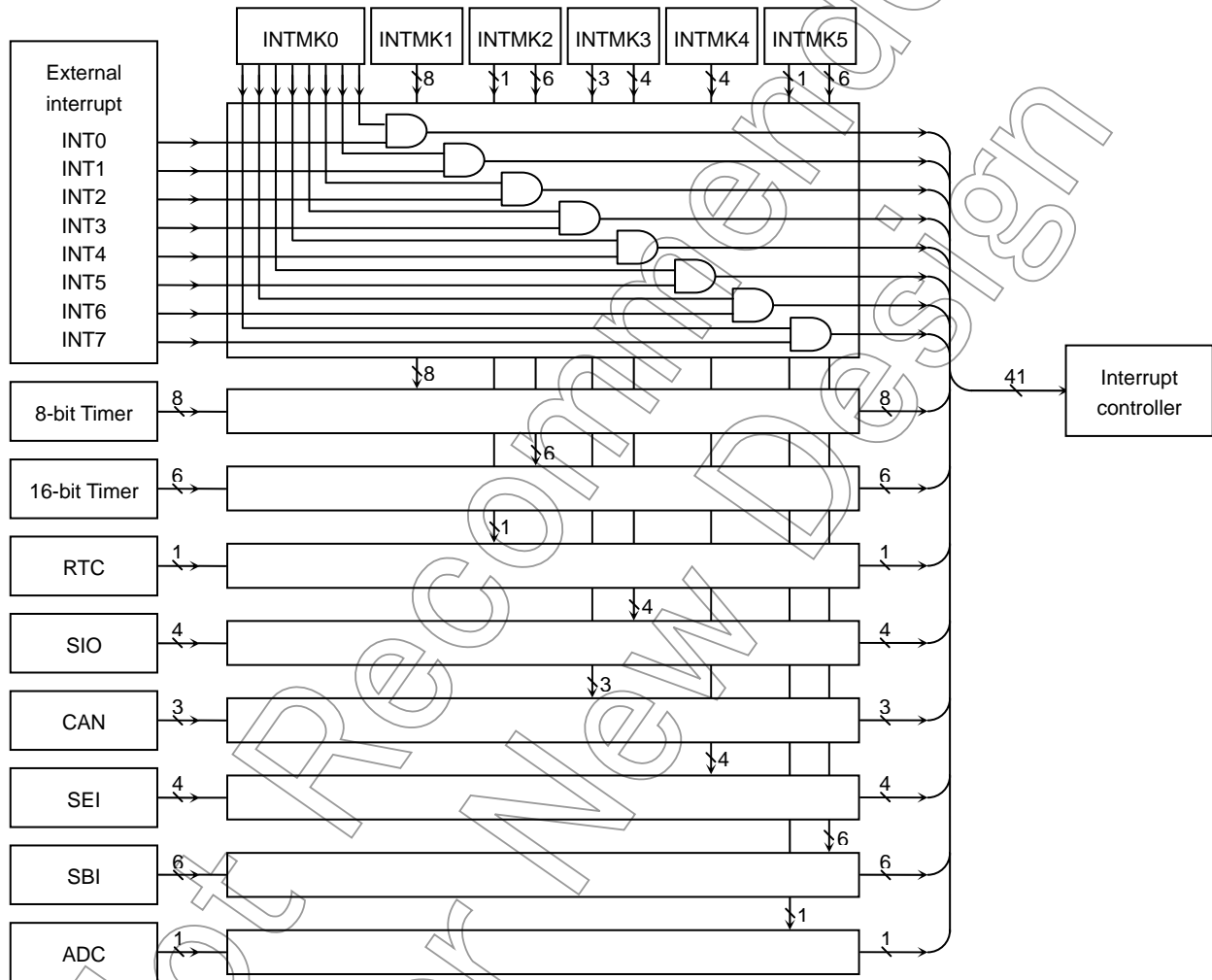
Figure 3.4.14 Block Diagram of Interrupt Mask Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTMK0	Interrupt Mask Control 0	E5H	MKI7	MKI6	MKI5	MKI4	MKI3	MKI2	MKI1	MKI0
			R/W							
			1	1	1	1	1	1	1	1
			INT7 0: Mask 1: Enable	INT6 0: Mask 1: Enable	INT5 0: Mask 1: Enable	INT4 0: Mask 1: Enable	INT3 0: Mask 1: Enable	INT2 0: Mask 1: Enable	INT1 0: Mask 1: Enable	INT0 0: Mask 1: Enable
INTMK1	Interrupt Mask Control 1	E6H	MKIT7	MKIT6	MKIT5	MKIT4	MKIT3	MKIT2	MKIT1	MKIT0
			R/W							
			1	1	1	1	1	1	1	1
			INTT7 0: Mask 1: Enable	INTT6 0: Mask 1: Enable	INTT5 0: Mask 1: Enable	INTT4 0: Mask 1: Enable	INTT3 0: Mask 1: Enable	INTT2 0: Mask 1: Enable	INTT1 0: Mask 1: Enable	INTT0 0: Mask 1: Enable
INTMK2	Interrupt Mask Control 2	E7H	−	MKIRTC	MKITOA	MKITO8	MKITRB	MKITRA	MKITR9	MKITR8
				R/W						
			−	1	1	1	1	1	1	1
			−	INTRTC 0: Mask 1: Enable	INTTOA 0: Mask 1: Enable	INTTO8 0: Mask 1: Enable	INTTRB 0: Mask 1: Enable	INTTRA 0: Mask 1: Enable	INTTR9 0: Mask 1: Enable	INTTR8 0: Mask 1: Enable
INTMK3	Interrupt Mask Control 3	E8H	−	MKICG	MKICT	MKICR	MKITX1	MKIRX1	MKITX0	MKIRX0
				R/W						
			−	1	1	1	1	1	1	1
			−	INTCG 0: Mask 1: Enable	INTCT 0: Mask 1: Enable	INTCR 0: Mask 1: Enable	INTTX1 0: Mask 1: Enable	INTRX1 0: Mask 1: Enable	INTTX0 0: Mask 1: Enable	INTRX0 0: Mask 1: Enable
INTMK4	Interrupt Mask Control 4	E9H	−	−	−	−	MKISET0	MKISER0	MKISEE0	MKISEM0
							R/W			
			−	−	−	−	1	1	1	1
							INTSET 0: Mask 1: Enable	INTSER 0: Mask 1: Enable	INTSEE 0: Mask 1: Enable	INTSEM 0: Mask 1: Enable
INTMK5	Interrupt Mask Control 5	EAH	−	MKISBS2	MKISBE2	MKIAD	MKISBS1	MKISBE1	MKISBS0	MKISBE0
				R/W						
			−	1	1	1	1	1	1	1
				INTSBS2 0: Mask 1: Enable	INTSBE2 0: Mask 1: Enable	INTAD 0: Mask 1: Enable	INTSBS1 0: Mask 1: Enable	INTSBE1 0: Mask 1: Enable	INTSBS0 0: Mask 1: Enable	INTSBE0 0: Mask 1: Enable

Maskable bit for INTAD request

0	INTAD is disabled
1	INTAD is enabled

Note: Ports D0, D1, and D4 are assigned two interrupt sources each (PD0: INT5/WUINT0, PD1: INT6/WUINT1, PD4: INT7/WUINT4). If both interrupt requests are issued when interrupts are enabled, both are handled. To use only either of the two interrupt sources, disable (mask) the other interrupt source using the interrupt mask register or wakeup mask register.

Figure 3.4.15 Interrupt Mask Registers

Example register settings:

To change the INT0 interrupt priority level from 3 to 7, set as follows:

```
DI                          ; Disable interrupt
LD   (INTMK0), 00H          ; Disable INT0
LD   (INTE0AD), 03H         ; Set INT0 interrupt level to 3
LD   (INTCLR), 0AH          ; Clear INT0 interrupt request flag
NOP                         ; Wait for 3 cycles
NOP
NOP
LD   (INTMK0), 01H          ; Enable INT0
EI                          ; Enable interrupt
     :      :               ; Programmed operation
DI                          ; Disable interrupt
LD   (INTMK0), 00H          ; Disable INT0
LD   (INTE0AD), 07H         ; Set INT0 interrupt level to 7
LD   (INTCLR), 0AH          ; Clear INT0 interrupt request flag
NOP                         ; Wait for 3 cycles
NOP
NOP
LD   (INTMK0), 01H          ; Enable INT0
EI                          ; Enable interrupt
```

3.4.5   Wakeup interrupt controller

The TMP92CD54I has eight wakeup pins (WUINT0-7). Input signals to those pins can be used to recover from the halt state. These pins are shared with port D (PD0-7).

The input signal attribute can be set to rising edge, falling edge or rising/falling edges, separately for each pin. The signals can also be masked on a pin-by-pin basis.
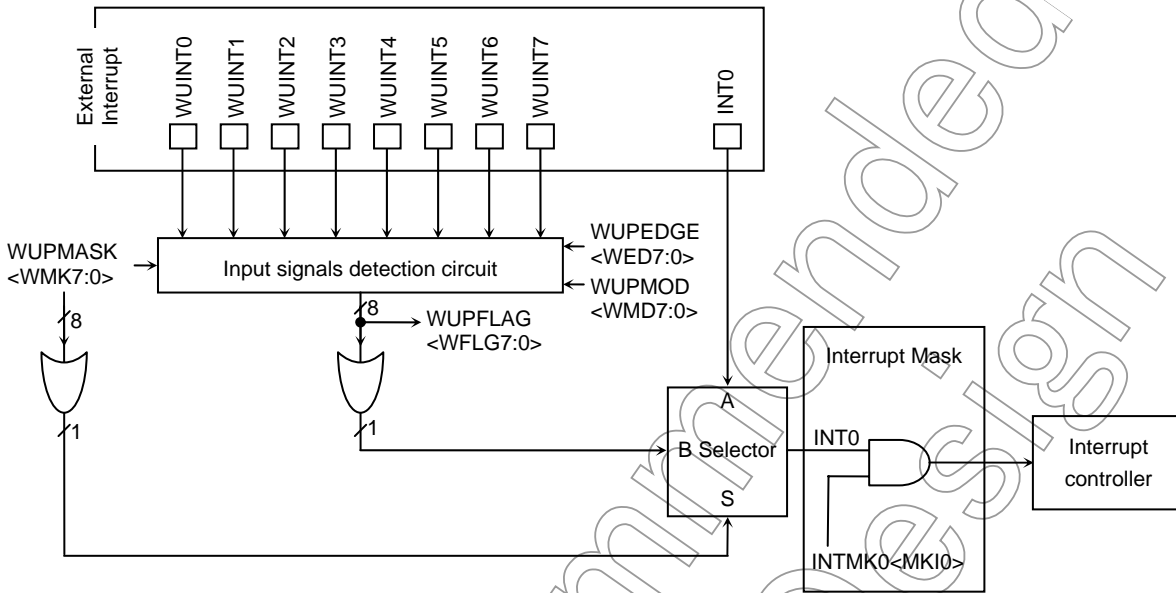
Figure 3.4.16 Block Diagram of ON/OFF Logic

The wakeup interrupt controller internally sends all interrupt signals on WUINT0-7 to INT0. Any WUINTn request that has been issued causes an INT0 interrupt to be issued. Like INT0 interrupt request from external pins, INT0 interrupt requests from the WUINTn pin are also enabled or disabled using the interrupt priority setup and interrupt mask registers.

A write of 1 to any bit in the WUPMASK register causes INT0 to be placed in wakeup interrupt mode. In this mode, the WUINTn signal for which a 1 is written in the WUPMASK register becomes valid and the input signal from the external INT0 pin is invalidated. To use the external INT0 pin, set the WUPMASK register to 00H.

The edge selection for the WUINTn signal can be set to rising edge, falling edge or rising/falling edges using the WUPMOD and WUPEDGE registers.

Reading the WUPFLAG register can determine whether a WUINTn interrupt request has been issued.

Wakeup FLAG status Register

WUPFLAG (00ECH)		7	6	5	4	3	2	1	0
	Symbol	WFLG7	WFLG6	WFLG5	WFLG4	WFLG3	WFLG2	WFLG1	WFLG0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	WUINT7 0:NO request 1: request	WUINT6 0:NO request 1: request	WUINT5 0:NO request 1: request	WUINT4 0:NO request 1: request	WUINT3 0:NO request 1: request	WUINT2 0:NO request 1: request	WUINT1 0:NO request 1: request	WUINT0 0:NO request 1: request

Wakeup Mode Control Register

WUPMOD (00EDH)		7	6	5	4	3	2	1	0
	Symbol	WMD7	WMD6	WMD 5	WMD4	WMD3	WMD2	WMD1	WMD0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	WUINT7 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT6 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT5 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT4 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT3 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT2 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT1 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT0 0:Falling & Rising Edge 1:Falling or Rising Edge

Wakeup Edge Select Register

WUPEDGE (00EEH)		7	6	5	4	3	2	1	0
	Symbol	WED7	WED6	WED 5	WED4	WED3	WED2	WED1	WED0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	WUINT7 0:Falling Edge 1:Rising Edge	WUINT6 0:Falling Edge 1:Rising Edge	WUINT5 0:Falling Edge 1:Rising Edge	WUINT4 0:Falling Edge 1:Rising Edge	WUINT3 0:Falling Edge 1:Rising Edge	WUINT2 0:Falling Edge 1:Rising Edge	WUINT1 0:Falling Edge 1:Rising Edge	WUINT0 0:Falling Edge 1:Rising Edge

Note: The WUPEDGE<WED7:0> setting becomes valid when a 1 is written to the corresponding bit in WUPMOD<WMD7:0>.

Wakeup Mask Register

WUPMASK (00EFH)		7	6	5	4	3	2	1	0
	Symbol	WMK7	WMK6	WMK5	WMK4	WMK3	WMK2	WMK1	WMK0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	function	WUINT7 0: Disable 1: Enable	WUINT6 0: Disable 1: Enable	WUINT5 0: Disable 1: Enable	WUINT4 0: Disable 1: Enable	WUINT3 0: Disable 1: Enable	WUINT2 0: Disable 1: Enable	WUINT1 0: Disable 1: Enable	WUINT0 0: Disable 1: Enable

Wakeup interrupt mask control

0	WUINTn Disabled (MASK)
1	WUINTn Enabled

Note1: Ports D0, D1, and D4 are assigned two interrupt sources each (PD0: INT5/WUINT0, PD1: INT6/WUINT1, PD4: INT7/WUINT4). If both interrupt requests are issued when interrupts are enabled, both are handled. To use only either of the two interrupt sources, disable (mask) the other interrupt source using the interrupt mask register or wakeup mask register. The input signal to port D is detected as an interrupt regardless of whether port D is set to input/output port, INTn, or WUINTn. For details, see the port block diagram.

Note2: If any bit of WUPMASK<WMK7:0> is set to 1, the input signal from the external INT0 pin is disabled. To use the external INT0 pin, write 00H to WUPMASK<WMK7:0> to disable the wakeup interrupt function.

Figure 3.4.17 Wakeup Registers

Example register settings:
The following example sets WUINT0 to rising edge and interrupt level 3:

```
DI                          ; Disable interrupt handling
LD  (INTMK0), 00H           ; Disable INT0
LD  (PDFC), 00H             ; Set PD0 to port
LD  (PDCR), 00H             ; Set PD0 to input mode
LD  (WUPMOD), 01H           ; Set WUINT0 to "falling or rising edge"
LD  (WUPEDGE), 01H          ; Set WUINT0 to "rising edge"
LD  (WUPFLAG), 00H          ; Clear WUINT0 flag
LD  (INTE0AD), 03H          ; Set INT0 interrupt level (as WUINT0) to 3
LD  (INTCLR), 0AH           ; Clear INT0 interrupt request flag
NOP                         ; Wait for 3 cycles
NOP
NOP
LD  (INTMK0), 01H           ; Enable WUINT0
EI                          ; Enable interrupt handling
```

3.5　Port Functions

The TMP92CD54I has input/output ports listed in Table 3.5.1. These port pins are shared pins; they are not only used for general-purpose input/output port functions but also used as internal CPU or I/O function pins.

Table 3.5.1　Port functions

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	$\overline{\text{RD}}$
	P71	1	I/O	Bit	$\overline{\text{WR}}$
	P72	1	I/O	Bit	SI2/SCL2
	P73	1	I/O	Bit	$\overline{\text{CS}}$
	P74	1	I/O	Bit	
	P75	1	I/O	Bit	$\overline{\text{WAIT}}$
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 / A16 / WUINT0
	PD1	1	I/O	Bit	TI9 / INT6 / A17 / WUINT1
	PD2	1	I/O	Bit	TO8 / A18 / WUINT2
	PD3	1	I/O	Bit	TO9 / A19 / WUINT3
	PD4	1	I/O	Bit	TIA / INT7 / A20 / WUINT4
	PD5	1	I/O	Bit	TIB / A21 / WUINT5
	PD6	1	I/O	Bit	TOA / A22 / WUINT6
	PD7	1	I/O	Bit	TOB / A23 / WUINT7
Port F	PF0	1	I/O	Bit	TXD0
	PF1	1	I/O	Bit	RXD0
	PF2	1	I/O	Bit	SCLK0 / $\overline{\text{CTS0}}$
	PF3	1	I/O	Bit	TXD1
	PF4	1	I/O	Bit	RXD1
	PF5	1	I/O	Bit	SCLK1 / $\overline{\text{CTS1}}$
	PF6	1	I/O	Bit	TX
	PF7	1	I/O	Bit	RX
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	$\overline{\text{SS}}$ / A8
	PM1	1	I/O	Bit	MOSI / A9
	PM2	1	I/O	Bit	MISO / A10
	PM3	1	I/O	Bit	SECLK / A11
	PM4	1	I/O	Bit	SCK2
Port N	PN0	1	I/O	Bit	SCK0
	PN1	1	I/O	Bit	SO0 / SDA0
	PN2	1	I/O	Bit	SI0 / SCL0
	PN3	1	I/O	Bit	SCK1 / A12
	PN4	1	I/O	Bit	SO1 / SDA1 / A13
	PN5	1	I/O	Bit	SI1 / SCL1 / A14
	PN6	1	I/O	Bit	SO2 / SDA2 / A15

3.5.1 Port 0 (P00-P07/D0-D7)

Port 0 is an 8-bit general-purpose input/output port for which each bit can be individually specified as input or output. The control register, P0CR, and function register, P0FC, are used to specify input or output.

In addition to the general-purpose input/output port function, the pins can also function as a data bus (D0-D7).
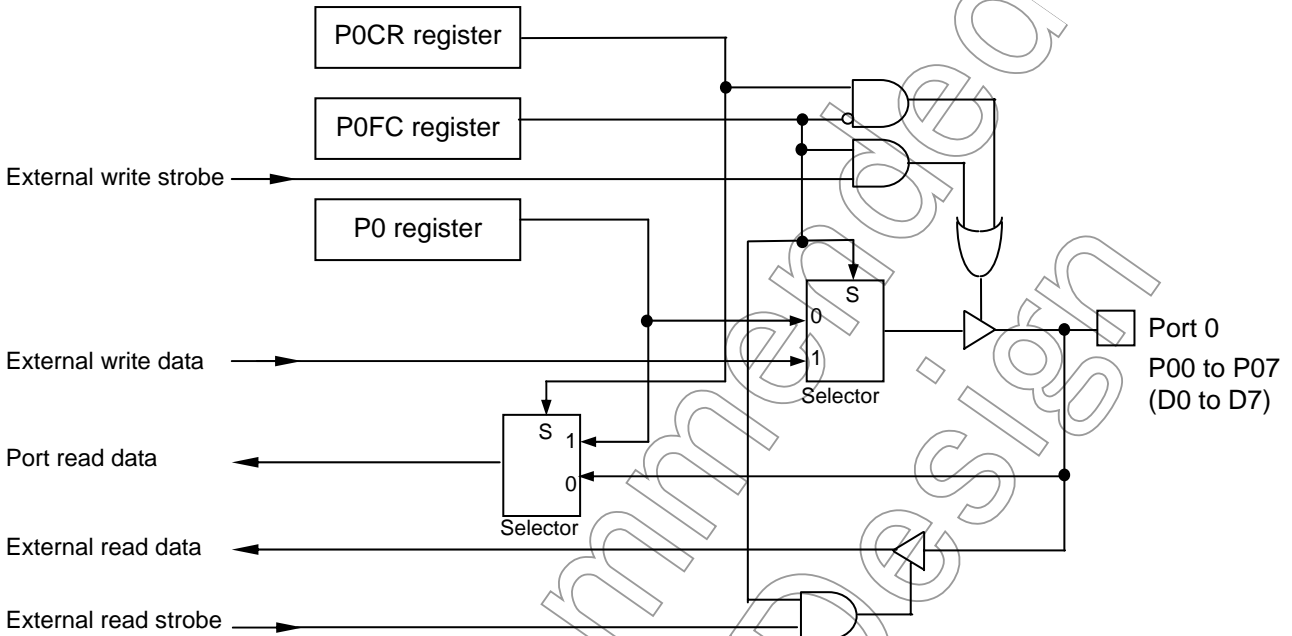
Figure 3.5.1   Port0

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 (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 to D0)							

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

3.5.2   Port 4 (P40-P47)

Port 4 is an 8-bit general-purpose input/output port for which each bit can be individually specified as input or output. The control register, P4CR, and function register, P4FC, are used to specify input or output.

In addition to the general-purpose input/output port function, the pins can also function as an address bus (A0-A7).
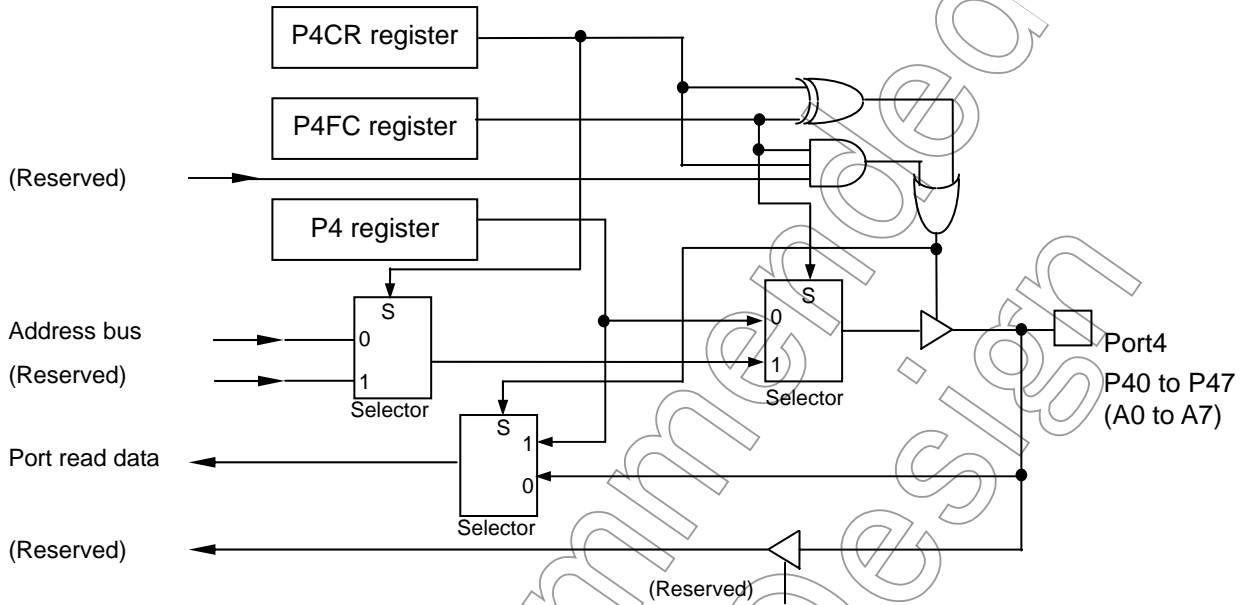
Figure 3.5.2   Port4

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 (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: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-P75)

Port 7 is an 6-bit general-purpose input/output port for which each bit can be individually specified as input or output. The control register, P7CR, and function register, P7FC, are used to specify input or output.

In addition to the general-purpose input/output port function, pins 70, 71 and 73 can also function as read, write strobe and chip select signals respectively, pin 72 as an I/O pin for the clock synchronous 8-bit SIO or the serial bus interface that operates as an I²C bus, and pin 75 as a wait input.

The SBI data input (SIO), SI2, and SBI clock input/output (I²C), SCL2, are always input-enabled.

A reset initializes port pins 70, 71, 73 and 74 to output port mode and pins 72 and 75 to input port mode.
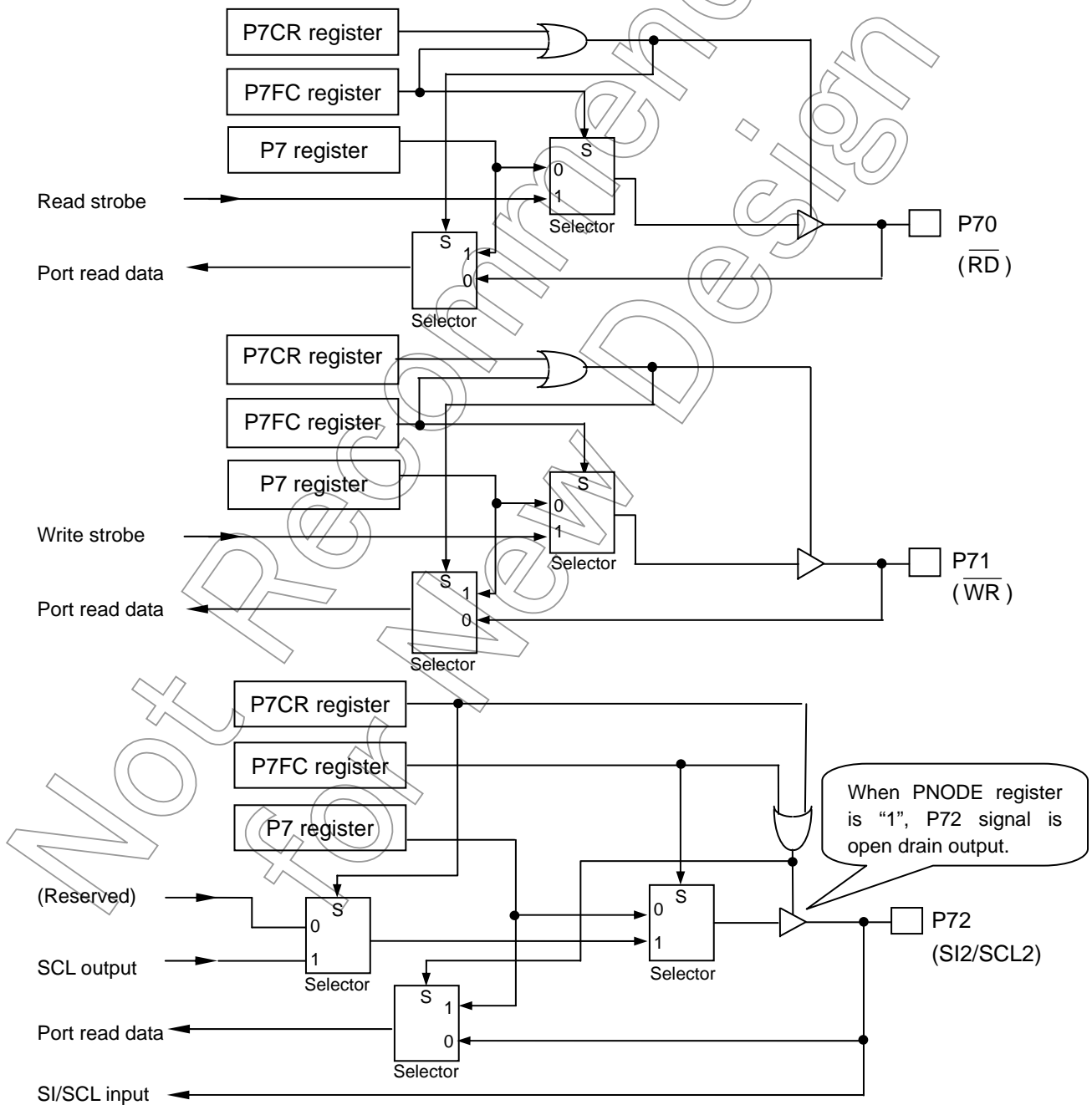
Figure 3.5.3　Port7 (P70 to P72)
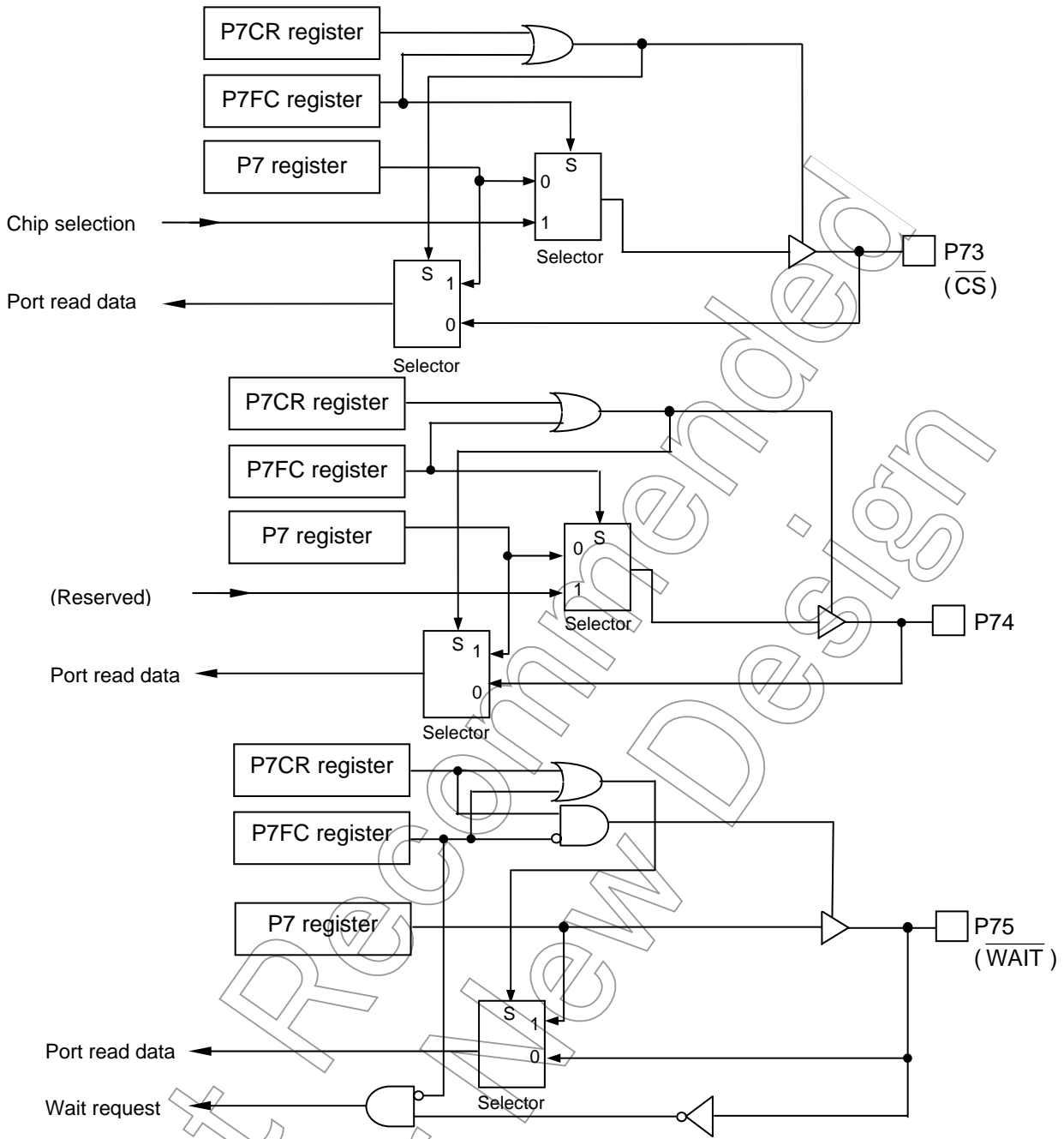
Figure 3.5.4 Port7 (P73 to P75)

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 (no RMW)	–	–	P75C	P74C	P73C	P72C	P71C	P70C
					W					
			–	–	0	1	1	0	1	1
					Input 1:Output					
P7FC	PORT7 Function Register	1FH (no RMW)	–	–	P75F	P74F	P73F	P72F	P71F	P70F
					W					
			–	–	0	0	0	0	0	0
					0:PORT 1: WAIT	0:PORT	0:PORT 1: CS	0:PORT 1:SI2 SCL2[Note1]	0:PORT 1: WR	0:PORT 1: RD

P7CR	P7FC	–	–	P75	P74	P73	P72	P71	P70
0	0			Input Port			Input Port, SI2	Input Port	
1	0			Output Port					
1	1			WAIT	Don't use this setting.	CS	Don't use this setting.	WR	RD
0	1			WAIT	Don't use this setting.	CS	SI2, SCL2	WR	RD

Note: The SCL2 (P72) pin (clock input/output pin for I²C mode) can be set to open-drain by setting
        PNODE<ODE72> to 1.

3.5.4   Port C (PC0-PC5)

Port C is an 6-bit general-purpose input/output port for which each bit can be individually specified as input or output. The control register, PCCR, and function register, PCFC, are used to specify input or output.

In addition to the general-purpose input/output port function, the pins can also function as 8-bit timer input/output or interrupt input.

Timer inputs TI0 and TI4 are always input-enabled except when in IDLE3 or STOP mode.

A reset initializes port C to input port mode.
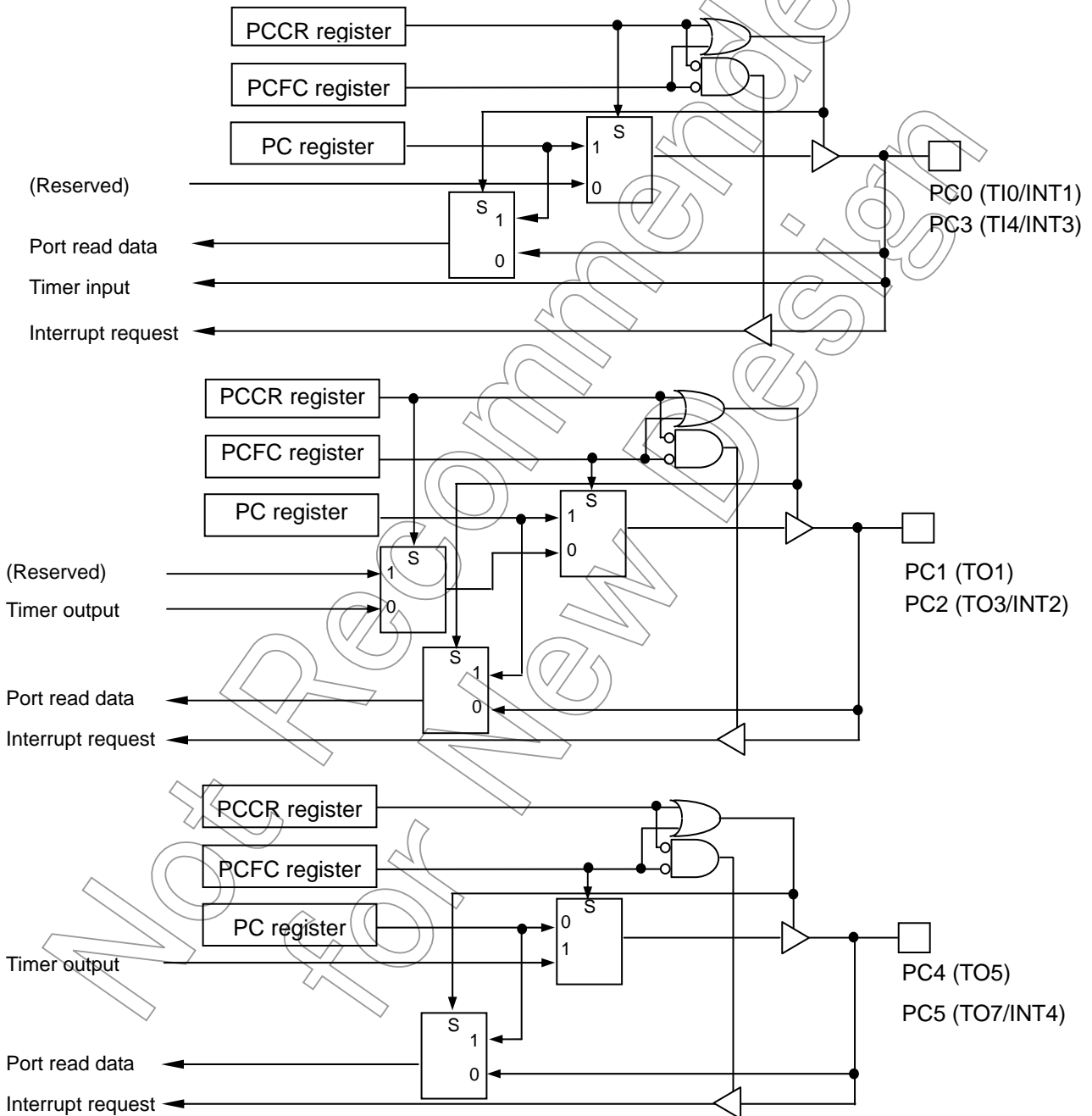
Figure 3.5.5   PortC (PC0 to PC5)

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 (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 TI4	0:PORT INT2 1:TO3	0:PORT 1:TO1	0:PORT INT1 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	Output Port	TO3	TO1	Output Port
0	1			TO7	TO5	Do not use this setting			

Note: Do not set <PC3C>:<PC3F>, <PC2C>:<PC2F>, <PC1C>:<PC1F>, and <PC0C>:<PC0F> to "0:1".

3.5.5   Port D (PD0-PD7)

Port D is an 8-bit general-purpose input/output port for which each bit can be individually specified as input or output. The control register, PDCR, and function register, PDFC, are used to specify input or output.

In addition to the general-purpose input/output port function, the pins can also function as 16-bit timer input/output or interrupt input.

Timer inputs TI8, TI9, TIA, TIB and external interrupts INT5, INT6, and INT7 are always input-enabled except when in IDLE3 or STOP mode. Wakeup interrupts WUINT0 to WUINT7 are always input-enabled.

A reset initializes port D to input port mode.
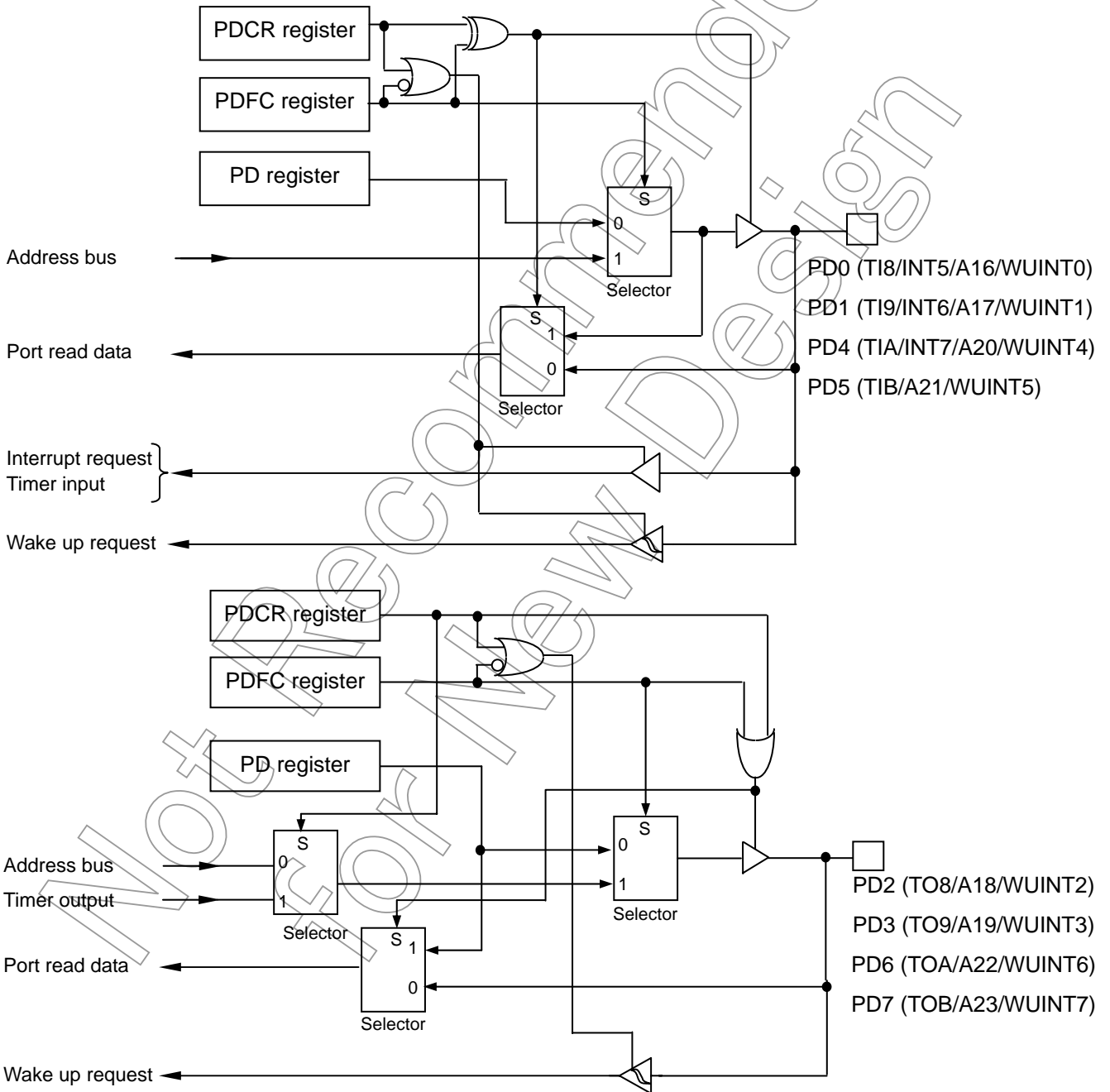
Figure 3.5.6   PortD

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 (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 WUINT7 1:TOB A23	0:PORT WUINT6 1:TOA A22	0:PORT TIB WUINT5 1:A21	0:PORT TIA INT7 WUINT4 1:A20	0:PORT WUINT3 1:TO9 A19	0:PORT WUINT2 1:TO8 A18	0:PORT TI9 INT6 WUINT1 1:A17	0:PORT TI8 INT5 WUINT0 1: A16

PDCR	PDFC	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
0	0	Input Port, WUINT7	Input Port, WUINT6	Input Port, TIB, WUINT5	Input Port, INT7, TIA, WUINT4	Input Port, WUINT3	Input Port, WUINT2	Input Port, INT6, TI9, WUINT1	Input Port, INT5, TI8, WUINT0
1	0	Output Port							
1	1	TOB	TOA,	TIB, WUINT5	TIA, INT7, WUINT4	TO9	TO8	TI9, INT6, WUINT1	TI8, INT5, WUINT0
0	1	A23	A22	A21	A20	A19	A18	A17	A16

Note 1: Ports D0, D1, and D4 are assigned two interrupt sources each (PD0: INT5/WUINT0, PD1: INT6/WUINT1, PD4: INT7/WUINT4).   If both interrupt requests are issued when these interrupts are enabled, both are handled. To use only either of the two interrupt sources, disable (mask) the other interrupt source using the interrupt mask register or wakeup mask register.

Note 2: To use any pin shared with an interrupt input as an input/output port pin, ensure that interrupt requests are disabled before setting the PDFC and PDCR registers.

3.5.6   Port F (PF0-PF7)

Port F is an 8-bit general-purpose input/output port for which each bit can be individually specified as input or output. The control register, PFCR, and function register, PFFC, are used to specify input or output.

In addition to the general-purpose input/output port function, the pins can also function as serial interface and controller area network (CAN) pins.

Serial receive data pins RXD0 and RXD1, CAN receive data pin RX, clear-to-send pins CTS0 and CTS1, and serial clock pins SCLK0 and SCLK1 are always input-enabled except when in IDLE3 or STOP mode.

A reset initializes port F to input port mode.
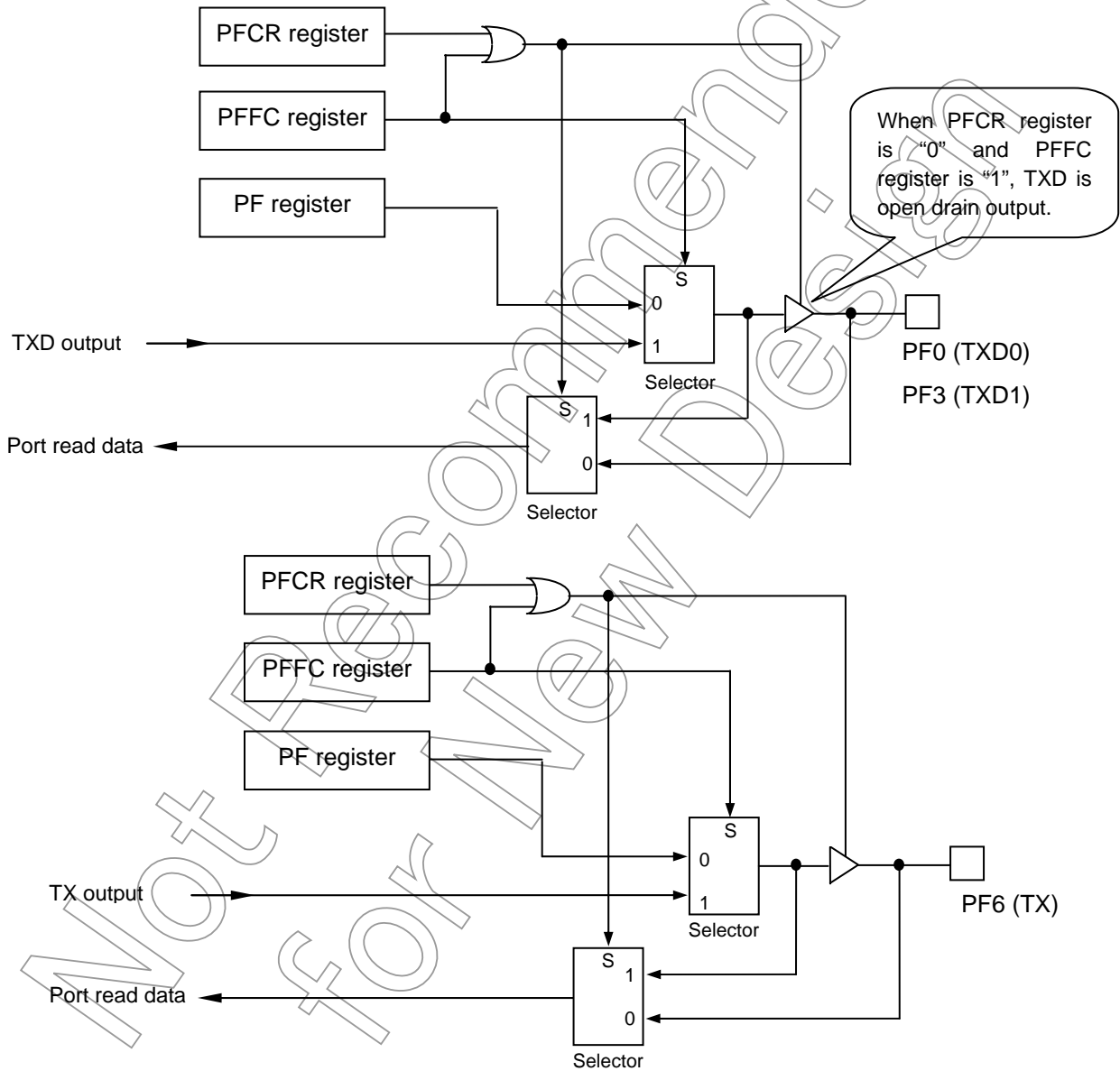
Figure 3.5.7   PortF (PF0, PF3 and PF6)
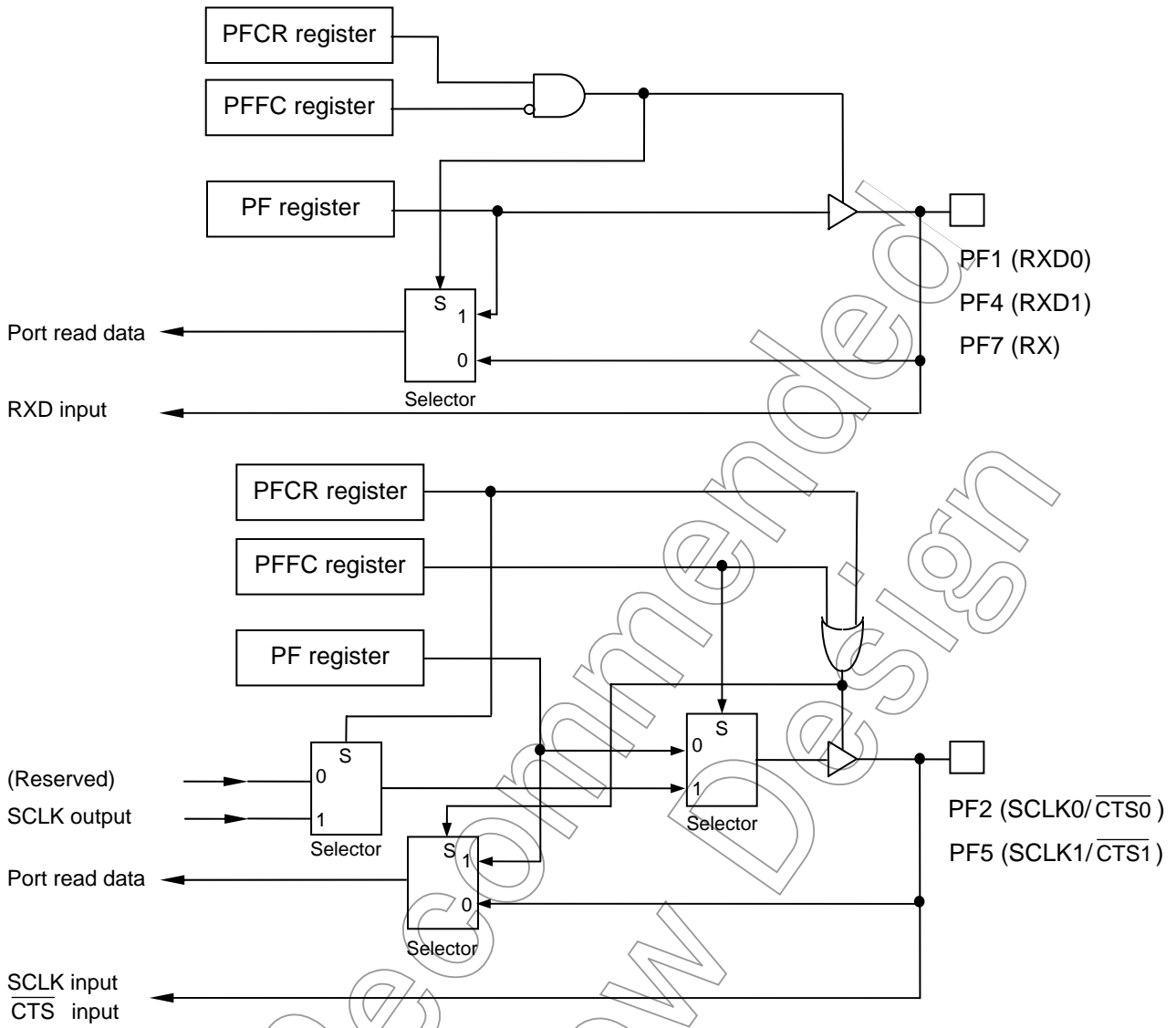
Figure 3.5.8   PortF (PF1, PF4, PF7, PF2 and PF5)

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 (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 1:RX	0:PORT 1:TX	0:PORT $\overline{CTS1}$ 1:SCLK1	0:PORT 1:RXD1	0:PORT 1:TXD1	0:PORT $\overline{CTS0}$ 1:SCLK0	0:PORT 1:RXD0	0:PORT 1:TXD0

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

3.5.7 Port G (PG0-PG7)

Port G is an 8-bit general-purpose input-only port.

In addition to the general-purpose input port function, the pins can also function as A/D converter input pins.

A/D conversion inputs AN0 to AN7 are always input-enabled except when in IDLE3 or STOP mode.

Port read data ◄────────────────◄── PortG

PG0 to PG7
(AN0 to AN7)

AD converter input ◄────────────────
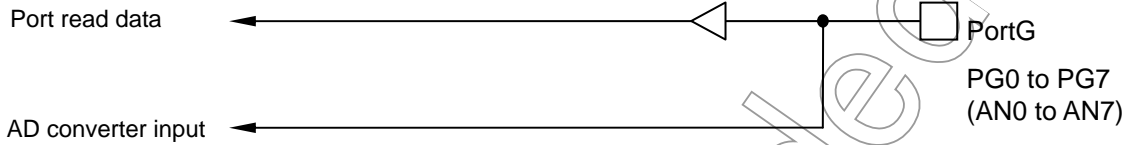
Figure 3.5.9 PortG

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-PL3)

Port L is an 4-bit general-purpose input-only port.

In addition to the general-purpose input port function, the pins can also function as A/D converter input pins.

A/D conversion inputs AN8 to AN11 are always input-enabled except when in IDLE3 or STOP mode.

Port read data ◄────────────────◄── PortL

PL0 to PL3
(AN8 to AN11)

AD converter input ◄────────────────

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-PM4)

Port M is an 5-bit general-purpose input/output port for which each bit can be individually specified as input or output. The control register, PMCR, and function register, PMFC, are used to specify input or output.

In addition to the general-purpose input/output port function, the pins can also function as serial general-purpose interface input/output pins.

The slave select pin SS, serial data transmit/receive pins MOSI and MISO, SEI clock pin SECLK, and SBI clock input/output (SIO) pin SCK2 are always input-enabled except when in IDLE3 or STOP mode.

A reset initializes port M to input port mode.
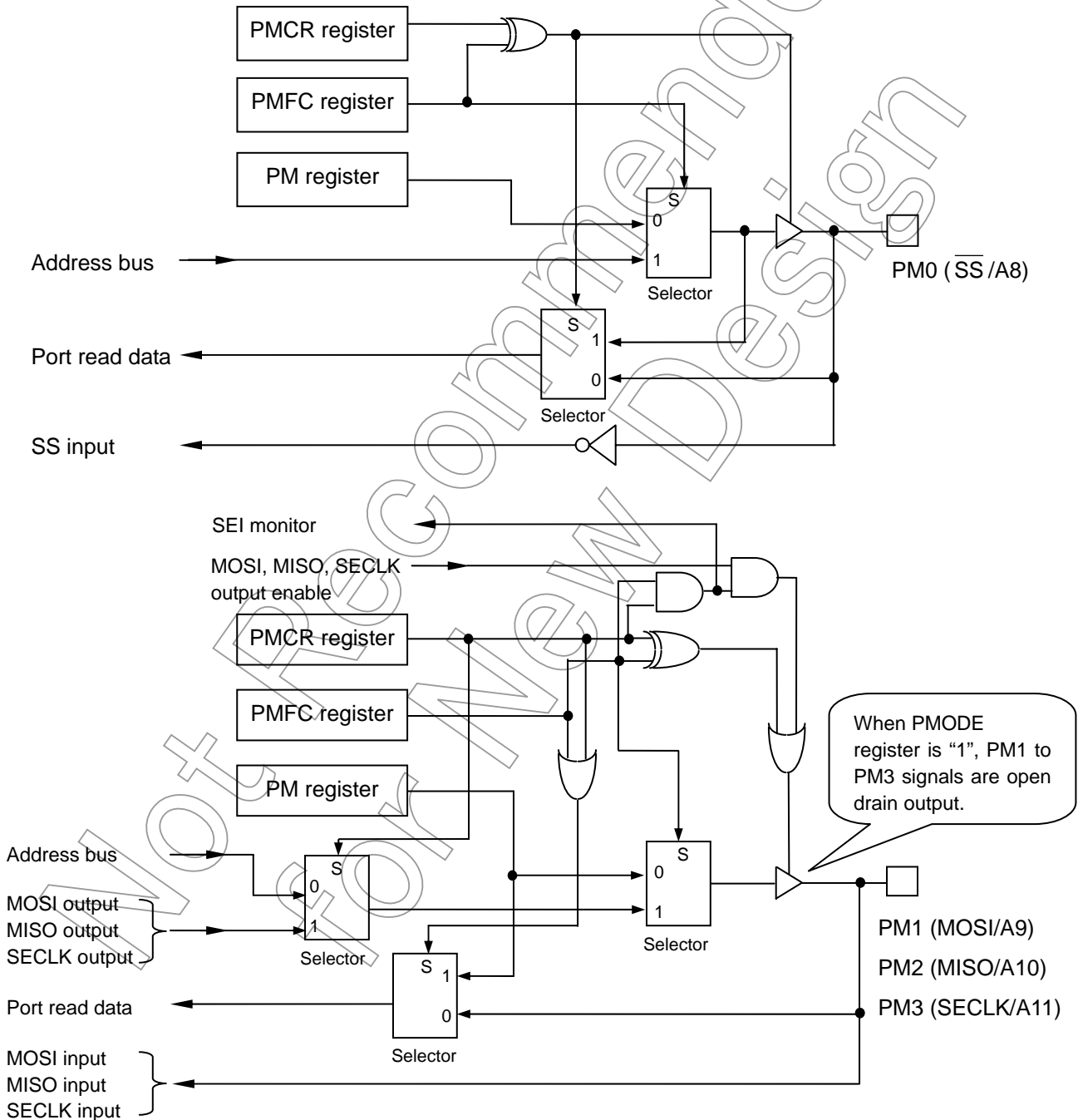
Figure 3.5.11   PortM (PM0 to PM3)
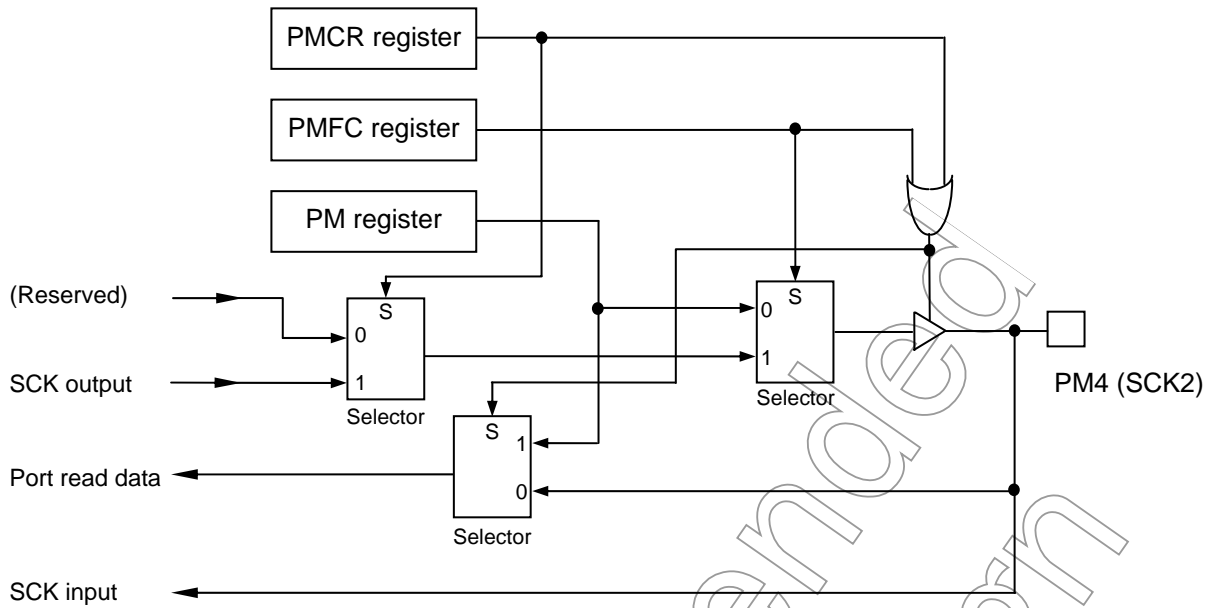
Figure 3.5.12   PortM (PM4)

Table 3.5.10   PortM Register

Symbol	Name	Address	7	6	5	4	3	2	1	0
PM	PORTM	58H	–	–	–	PM4	PM3	PM2	PM1	PM0
								R/W		
			–	–	–	0	0	0	0	0
								Input/Output		
PMODE	PORTM Open Drain Enable Register	59H	–	–	–	–	ODEM3	ODEM2	ODEM1	-
								R/W		
			–	–	–	–	0	0	0	-
							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)	–	–	–	PM4C	PM3C	PM2C	PM1C	PM0C
								W		
			–	–	–	0	0	0	0	0
							0:Input 1:Output			
PMFC	PORTM Function Register	5BH (no RMW)	–	–	–	PM4F	PM3F	PM2F	PM1F	PM0F
								W		
			–	–	–	0	0	0	0	0
						0:PORT 1:SCK2	0:PORT 1: SECLK A11	0:PORT 1:MISO A10	0:PORT 1:MOSI A9	0:PORT 1: SS A8

PMCR	PMFC	–	–	–	PM4	PM3	PM2	PM1	PM0
0	0	–	–	–	Input Port, SCK2 (Input)	Input Port	Input Port	Input Port	Input Port, SS
1	0	–	–	–	Output Port				
1	1	–	–	–	SCK2 (Output)	SECLK	MISO	MOSI	SS
0	1	–	–	–	Don't use this setting	A11	A10	A9	A8

3.5.10 Port N (PN0-PN6)

Port N is an 7-bit general-purpose input/output port for which each bit can be individually specified as input or output. The control register, PNCR, and function register, PNFC, are used to specify input or output.

In addition to the general-purpose input/output port function, the pins can also function as serial channel input/output pins.

The SBI clock input/output (SIO) pins SCK0 and SCK1, SBI data input (SIO) pins SI0 and SI1, SBI clock input/output (I2C) pins SCL0 and SCL1, and SBI data input/output (I2C) pins SDA0 and SDA1 are always input-enabled except when in IDLE3 or STOP mode.

A reset initializes port N to input port mode.
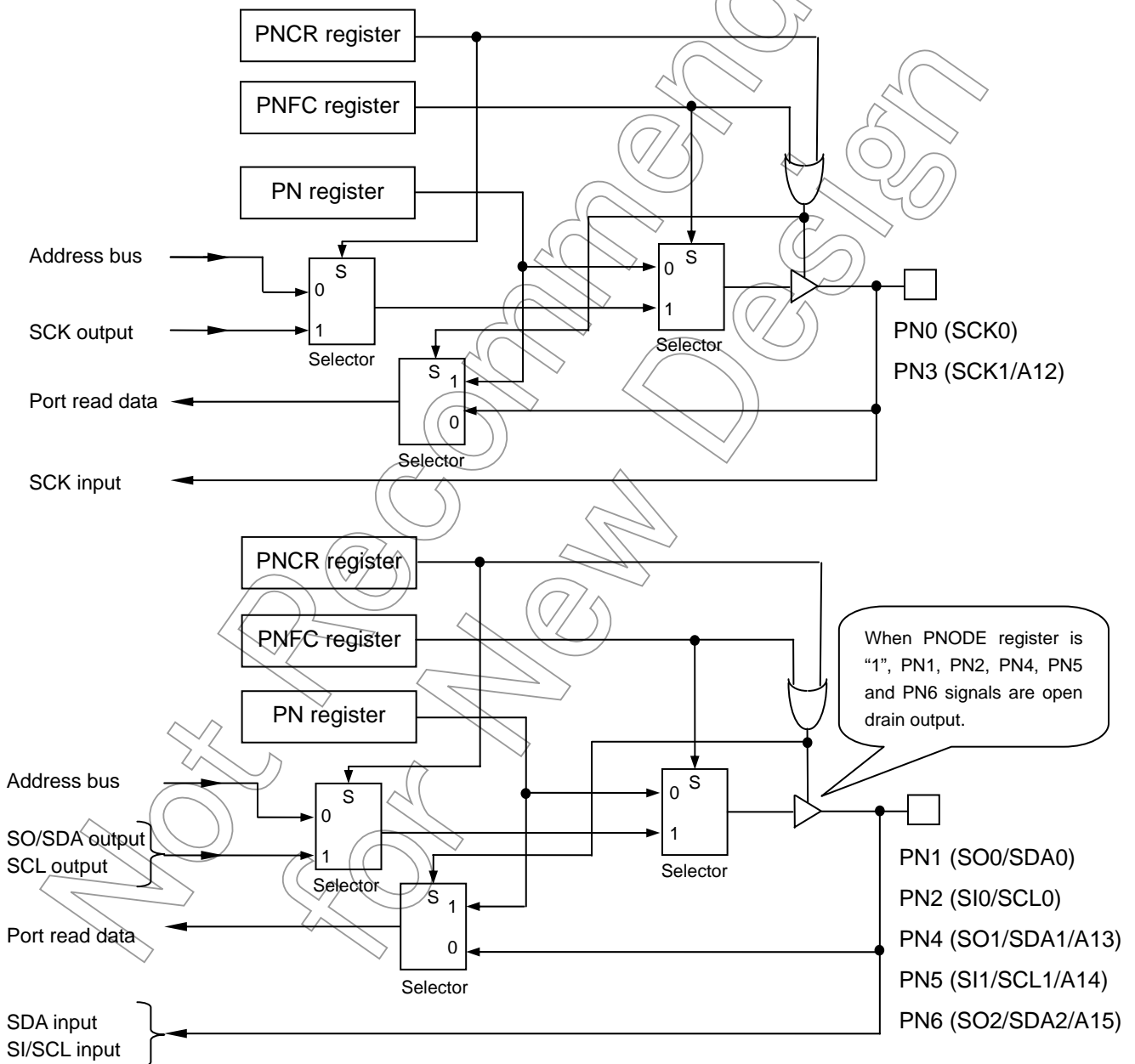
Figure 3.5.13   PortN

Table 3.5.11 PortN Register

Symbol	Name	Address	7	6	5	4	3	2	1	0
PN	PORTN	5CH	−	PN6	PN5	PN4	PN3	PN2	PN1	PN0
				R/W						
			−	0	0	0	0	0	0	0
				Input/Output						
PNODE	PORTN Open Drain Enable Register	5DH	ODE72	ODEN6	ODEN5	ODEN4	−	ODEN2	ODEN1	−
				R/W				R/W		
			0	0	0	0	−	0	0	−
			P72 output 0:CMOS 1:Open Drain	PN6 output 0:CMOS 1:Open Drain	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)	−	PN6C	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
				W						
			−	0	0	0	0	0	0	0
				0:Input　1:Output						
PNFC	PORTN Function Register	5FH (no RMW)	−	PN6F	PN5F	PN4F	PN3F	PN2F	PN1F	PN0F
				W						
			−	0	0	0	0	0	0	0
				0:PORT 1: SO2 SDA2 A15	0:PORT SI1 1:SCL1 A14	0:PORT 1:SO1 SDA1 A13	0:PORT 1:SCK1 A12	0:PORT SI0 1:SCL0	0:PORT 1:SO0 SDA0	0:PORT 1:SCK0

PNCR	PNFC	−	PN6	PN5	PN4	PN3	PN2	PN1	PN0
0	0	−	Input Port	Input Port	Input Port	Input Port, SCK1 (Input)	Input Port, SI0	Input Port	Input Port, SCK0 (Input)
1	0	−	Output Port						
1	1	−	SO2/SDA2	SCL1	SO1/SDA1	SCK1 (Output)	SCL0	SO0/SDA0	SCK0 (Output)
0	1	−	A15	A14	A13	A12	Don't use this setting.		

3.6 Memory Controller

3.6.1 Overview of functions

The TMP92CD54I memory controller can control a block address space as follows:

(1) Accessing a block address space in an external area

The memory controller can specify a block size and start address for a single block address space allocated in an external area.

(2) Specifying a memory type

The memory controller can specify either SRAM or ROM as the type of memory to be connected to a block address space.

(3) Specifying a data bus width

The data bus width of a block address space is fixed to eight bits.

(4) Controlling wait states

The memory controller can control the number of wait states for external bus cycles using the wait specification bit in a control register and the $\overline{\text{WAIT}}$ input pin. It can specify the number of wait states separately for a read cycle and write cycle. The memory controller supports the following five modes to control the number of wait states:

0 wait states, 1 wait state,

2 wait states, 3 wait states,

N wait states (controlled using the $\overline{\text{WAIT}}$ pin)

> 0 wait, 1 wait, 2 wait, 3 wait,
> N wait (N is controlled with $\overline{\text{WAIT}}$ pin)

3.6.2 Control registers and operation upon a reset

This section describes the registers used to control the memory controller as well as the status upon a reset and necessary settings.

(1) Control registers

The following control registers are used for the memory controller:

> ● Control register (BCSH/BCSL: Block chip select High/Low)
>   · Configures the basic functions of the memory controller, such as the type of memory to be connected and the number of wait states for read and write.
> ● Memory start address register (MSAR)
>   · Specifies the start address of the selected block address space.
> ● Memory address mask register (MSMR)
>   · Specifies the block size of the selected block address space.

(2) Operation upon a reset

Upon a reset, the block address space is set to addresses 000000H to FFFFEFH.

After a reset has been released, use the memory start address register (MSAR) and memory address mask register (MAMR) to specify the block address space and configure the control register (BCSL/H).

To make the settings effective, set BCSL<BE> to 1.

3.6.3 Basic functions and register settings

This section describes the memory controller functions for setting the block address area, memory type, and number of wait states.

(1) Specifying the block address space

Clearing BCSH<BM> to 0 causes the block address space to be fixed to the range of 000000H to FFFFEFH with the settings of MSAR (memory start address register) and MAMR (memory address mask register) disabled.

Setting BCSH<BM> to 1 enables the MSAR and MAMR settings, thus allowing the user to specify any block address space. The MSAR and MAMR specify the start address and block address space size, respectively. To specify the block address space size, either mask or enable comparison for each bit of the address. The memory controller compares the address with the value in the register in each bus cycle to determine whether it is accessing an external memory location. Note that any address bits masked with MAMR are not compared. If the compared addresses match, the memory controller pulls the chip select signal ($\overline{\text{CS}}$) low.

Figure 3.6.1 shows an example of connection between the TMP92CD54I and external memory. In this example, RAM is connected via an 8-bit bus.
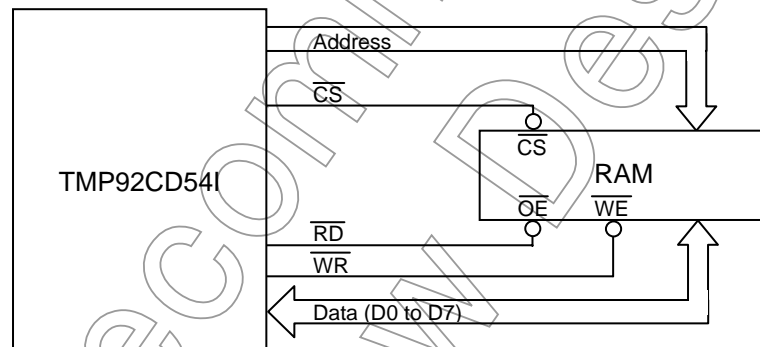
Figure 3.6.1 Example of connecting external memory (external RAM)

(i) Setting the memory start address register

Bits MS23 to MS16 in the memory start address register correspond to address bits A23 to A16, respectively. The start lower address, A15 to A0, are always 0000H. The start address of the block address space can, therefore, be specified within the range from 000000H to FF0000H, in 64-Kbyte units.

(ii) Setting the memory address mask register

The memory address mask register specifies whether each bit in the address will be compared or not. The bits cleared to 0 will be compared while those set to 1 will not be compared. Bit A23 is always compared.

The address bits for the block address space that can be masked are A22 to A15.

The following sizes can be specified for the block address space:

Table 3.6.1 Block Address Space

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			○	○	○	○	○	○	○	○	○

Note: Upon a reset, BCSH<BM> is set to 0. The block address space is, therefore, set to addresses 000000H to FFFFEFH. Setting BCSH<BM> to 1 enables the start address (MSAR register) and address space size (MAMR register) to be specified.

(iii) Example register settings

To set the block address area 64 KB from address 110000H, set the registers as follows:

```
        MSB                 LSB
        7   6   5   4   3   2   1   0
MSAR    0   0   0   1   0   0   0   1   ; set start address to 110000H
MAMR    0   0   0   0   0   0   0   1   ; set  block address area size to 64k-bytes
```

Bits MS23 to MS16 in the memory start address register (MSAR) correspond to address bits A23 to A16, respectively. A15 to A0 are 0. If the value of MASR is set as shown above, therefore, the start address of the block address space becomes 110000H.

Bits MV22 to MV15 in the memory address mask register specify whether bits A22 to A15 will be compared in address comparison. The bits cleared to 0 will be compared while those set to 1 will not be compared. Bit A23 is always compared.

The above settings specify that bits A23 to A16 will be compared with the value specified as the start address. This results in the 64-Kbyte range of 110000H to 11FFFFH being set as the block address space. If the address on the bus is matched, the memory controller drives the chip select signal ($\overline{\text{CS}}$) low.

(iv) If the block overlaps built-in memory space

If the specified block address space overlaps the built-in memory space, the block address space will be handled according to the following order of priority:

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

This means that priorities are assigned to prevent collision rather than remapping the block addresses.

If any address outside the specified block address space is accessed, the number of wait bus cycles is set to 1 (with the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals output but the $\overline{\text{CS}}$ signal not output). It is a fixed parameter.

(2) Controlling wait states

An external bus cycle is completed in two states (100 ns at 20 MHz) at a minimum. The number of wait states for read and write cycles can be specified by setting <BWR2:0> and <BWW2:0> in control register BCSL. BWW and BWR can be set in the same way, as shown below.

Table 3.6.2 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
Others			(Reserved)

(i) Fixed wait mode

In this mode, a bus cycle is always completed in a specified number of states. The number of states can be specified in the range from two states (zero waits) to five states (three waits).

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

In this mode, the $\overline{\text{WAIT}}$ input pin is sampled and wait states are inserted as long as the signal is low. In this mode, a bus cycle requires two states at a minimum. If the wait signal is high in the second state, the bus cycle is completed. A bus cycle may be extended to more than two states as long as the wait signal is low.

(3) Bus access timing

- External read/write bus cycle (0 wait states)
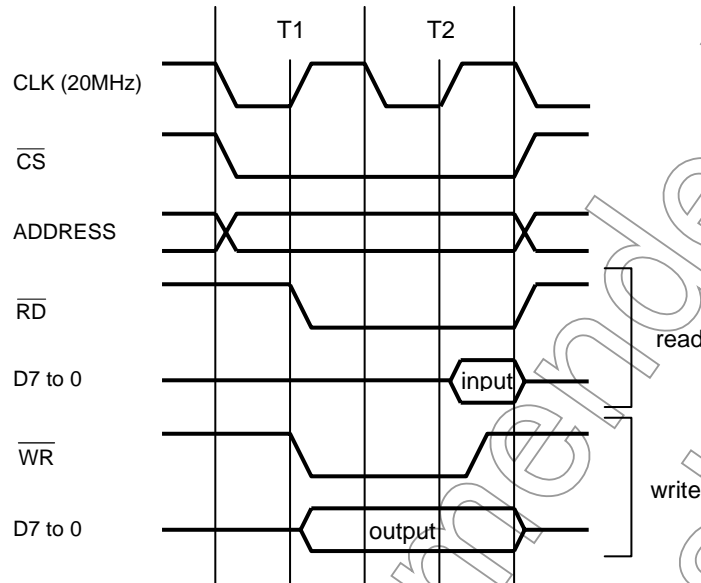
- External read/write bus cycle (1 wait state)
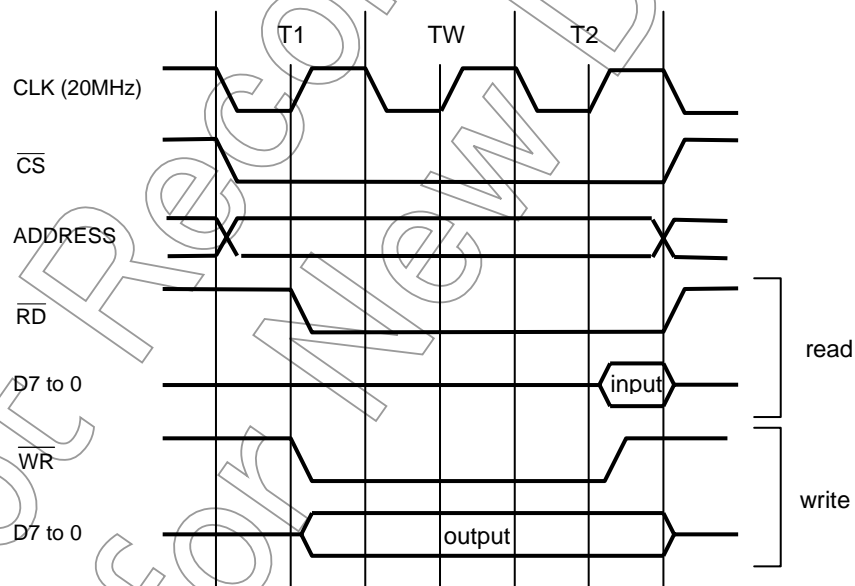
Figure 3.6.2 External Read/Write Bus Cycle (0 and 1 wait status)

- External read/write bus cycle (0 wait states in $\overline{\text{WAIT}}$ pin input mode)
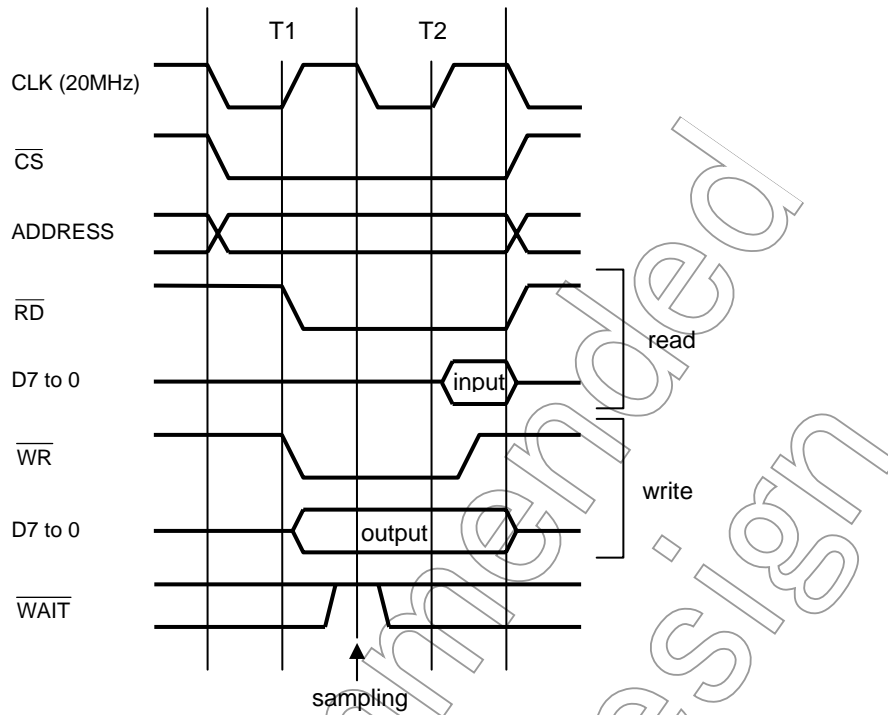
- External read/write bus cycle (N wait states in $\overline{\text{WAIT}}$ pin input mode)
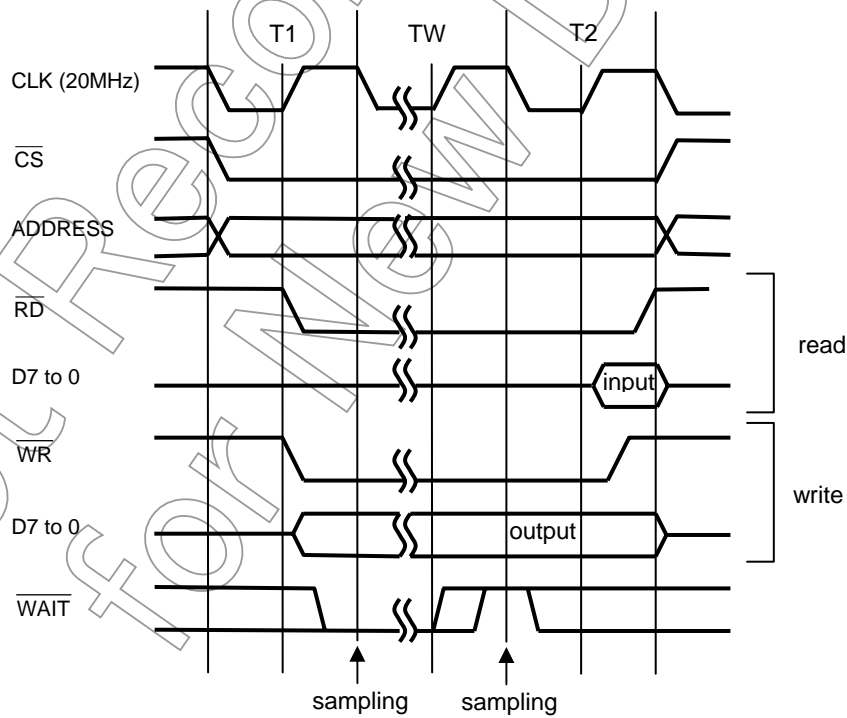
Figure 3.6.3 External Read/Write Bus Cycle ($\overline{\text{WAIT}}$ pin input mode)

- Example $\overline{\text{WAIT}}$ input circuit (for 5 wait states)
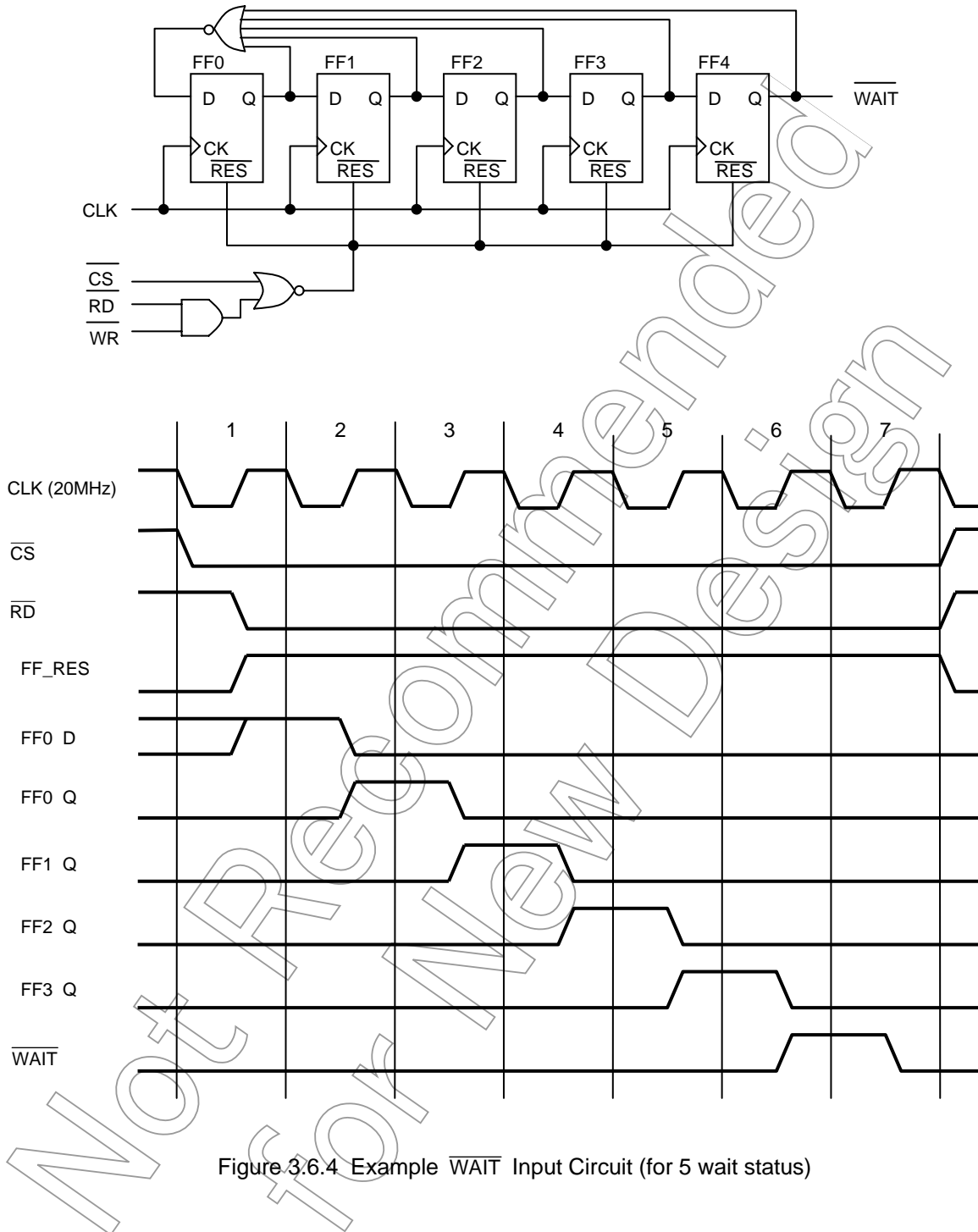
Figure 3.6.4 Example $\overline{\text{WAIT}}$ Input Circuit (for 5 wait status)

3.6.4 Registers

This section summarizes the memory control registers and their settings. For the address of each register, refer to Chapter 5, "List of Special Function Registers."

(1) Control registers

The memory is controlled with the BCSL and BCSH registers.

Block CS/WAIT control register (L)

	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

BCSL
(0148H)

<BWW2:0>: Specify the number of write wait states.

    001 = 2-state (0-wait) access   010 = 3-state (1-wait) access

    101 = 4-state (2-wait) access   110 = 5-state (3-wait) access

    011 = $\overline{\text{WAIT}}$ pin input mode   Others = (reserved)

<BWR2:0>: Specify the number of read wait states.

    001 = 2-state (0-wait) access   010 = 3-state (1-wait) access

    101 = 4-state (2-wait) access   110 = 5-state (3-wait) access

    011 = $\overline{\text{WAIT}}$ pin input mode   Others = (reserved)

Block CS/WAIT control register (H)

	7	6	5	4	3	2	1	0
bit Symbol	BE	BM	–	–	BOM1	BOM0	BBUS1	BBUS0
Read/Write				W				
After reset	1	0	0(Fix to 0)	0(Fix to 0)	0	0	0	0

BCSH
(0149H)

<BE>:   Enable bit

  0 = Does not output the chip select signal.

  1 = Outputs the chip select signal (default).

<BM>:   Specify block address space.

  0 = Sets the CS block address space to 000000H-FFFFEFH (default).

  1 = Enables the CS block address space to be programmed.

  Note: Upon a reset, the CS block address space is set to 000000H-FFFFEFH.

<BOM1:0>

  00 = SRAM or ROM (default)

  Others = (reserved)

<BBUS1:0:>   Specify data bus width.

  00 = 8 bits (default)

  Others = (reserved)

Figure 3.6.5 Block CS/WAIT Control Register

(2) Block address space specification registers

The start address and range of the block address space are specified using two registers, memory start address register (MSAR) and memory address mask register (MAMR).

Memory start address register

MSAR
(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

<MS23:16>:   Specify start address.

These bits specify the start address of each block address space. The bits in this register correspond to address bits A23 to A16.

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

<MV22:15>:

These bits specify whether the corresponding address bits will be compared in address comparison. Bits MV22 to MV15 correspond to address bits A22 to A15. Setting a bit to 0 causes the corresponding bit of the value on the address bus to be compared with the start address bit. Setting a bit to 1 causes the bit not to be compared. Bit A23 is always compared.

Figure 3.6.6 Memory Start Address/Memory Address Mask Registers

3.7 8-bit Timers

The TMP92CD54I contains eight channels of 8-bit timers (timers 0 to 7).

The timers are grouped into four modules, each consisting of two channels (timer 01, timer 23, timer 45 and timer 67) and can operate in one of the following four modes:

- 8-bit interval timer mode
- 16-bit interval timer mode
- 8-bit programmable square wave (PPG, with variable cycle and duty ratio) output mode
- 8-bit pulse width modulation (PWM, with fixed cycle and variable duty ratio) output mode

Figure 3.7.1 to Figure 3.7.4 show block diagrams of timers 01, 23, 45 and 67.

Each channel consists of an 8-bit up-counter, 8-bit comparator and 8-bit timer register. A timer flip-flop and prescaler are provided for each pair of two channels.

The timer operating mode and flip-flop are controlled using five special function registers (SFR).

Four modules (timers 01, 23, 45 and 67) operate independently of each other. All modules operate in the same way except the differences in specification listed in Table 3.7.1. This section only describes the operation of timer 01.

Table 3.7.1 Registers and Pins for each Module

Specification \ Module		timers 01	timers 23	timers 45	timers 67
External pin	Input pin for external clock	TI0 (shared with PC0)	-	TI4 (shared with PC3)	-
	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 diagram for each module
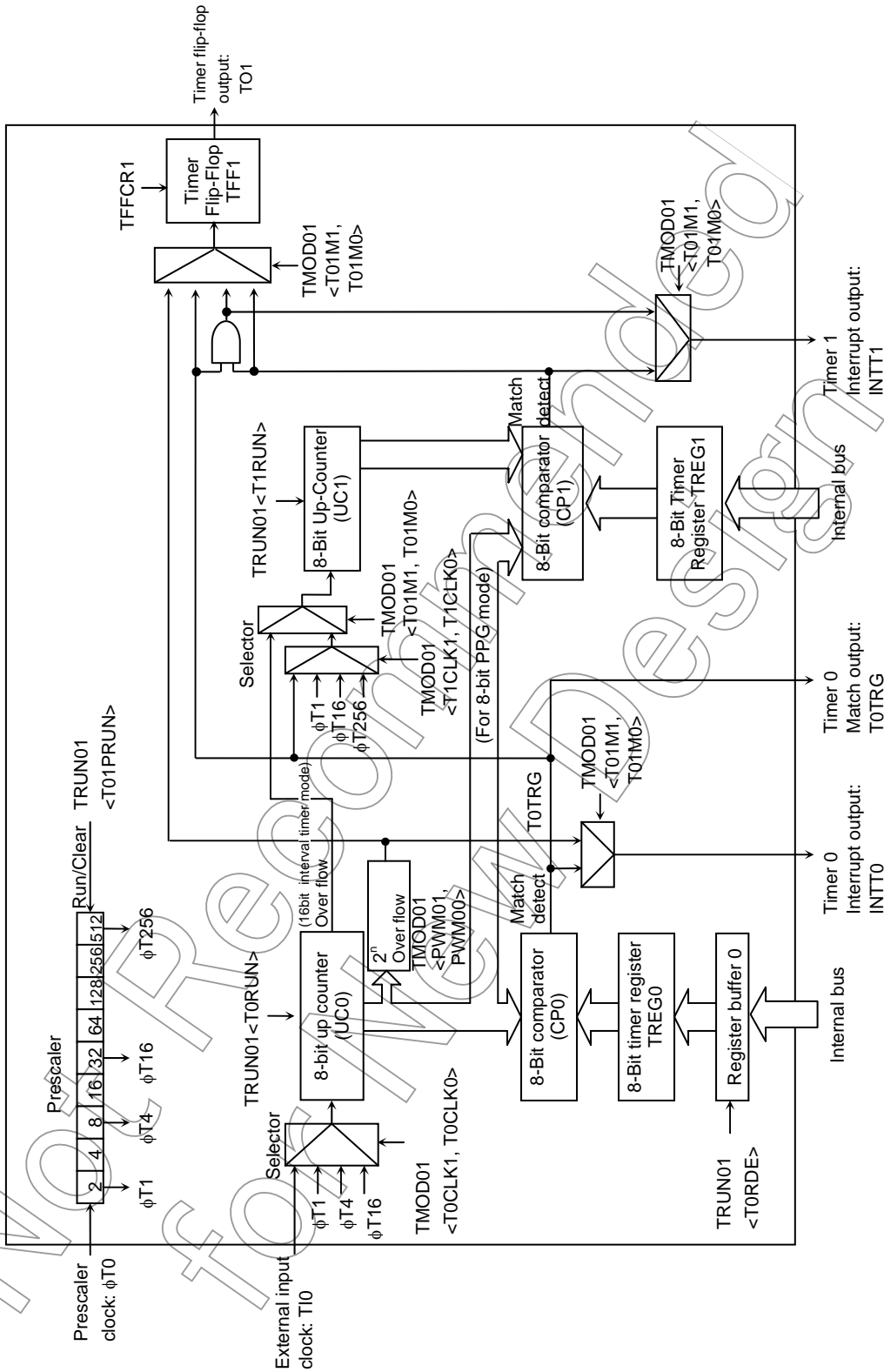
Figure 3.7.1 Timers 01 Block Diagram
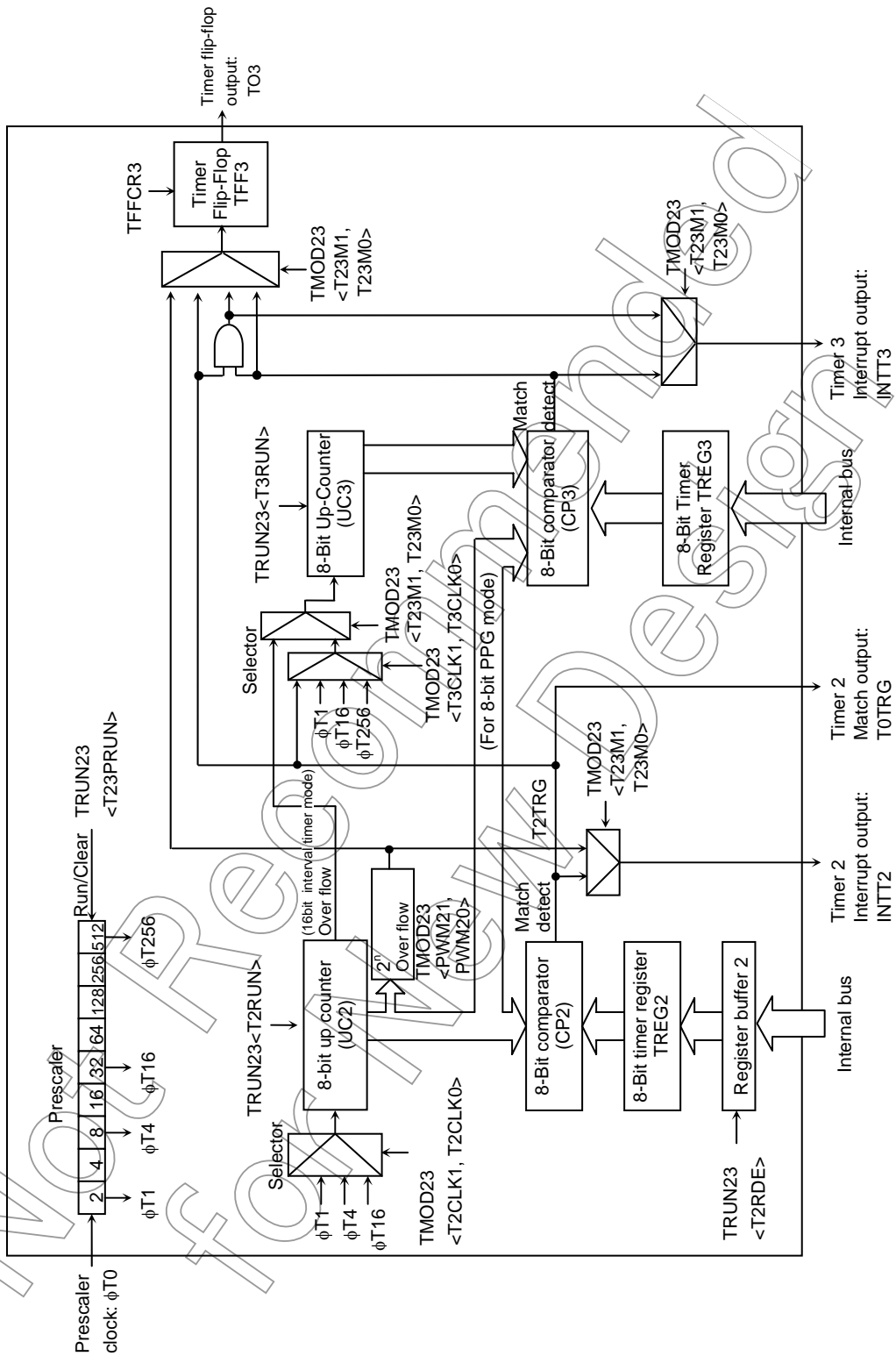
Figure 3.7.2 Timers 23 Block Diagram
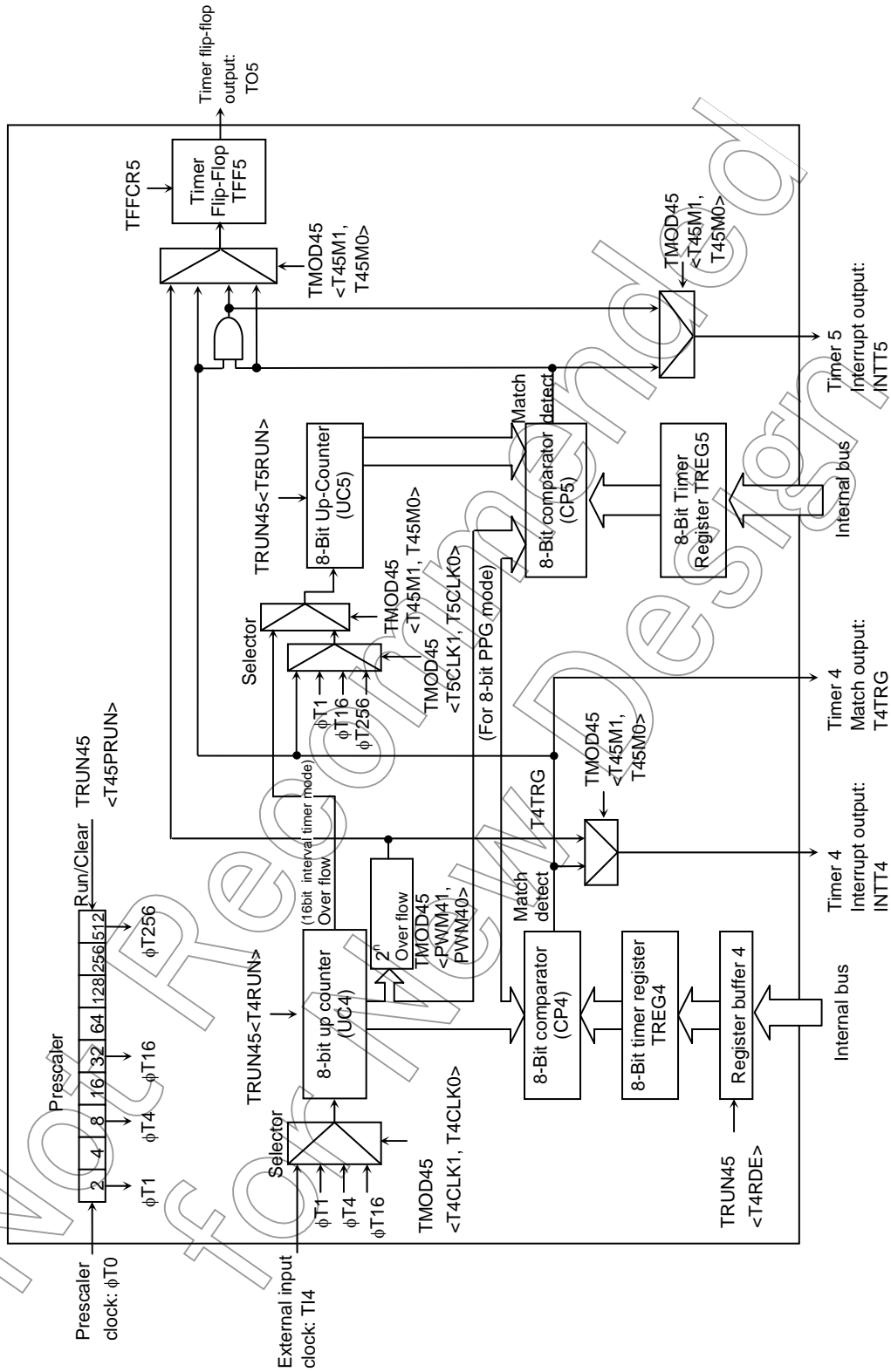
Figure 3.7.3 Timers 45 Block Diagram
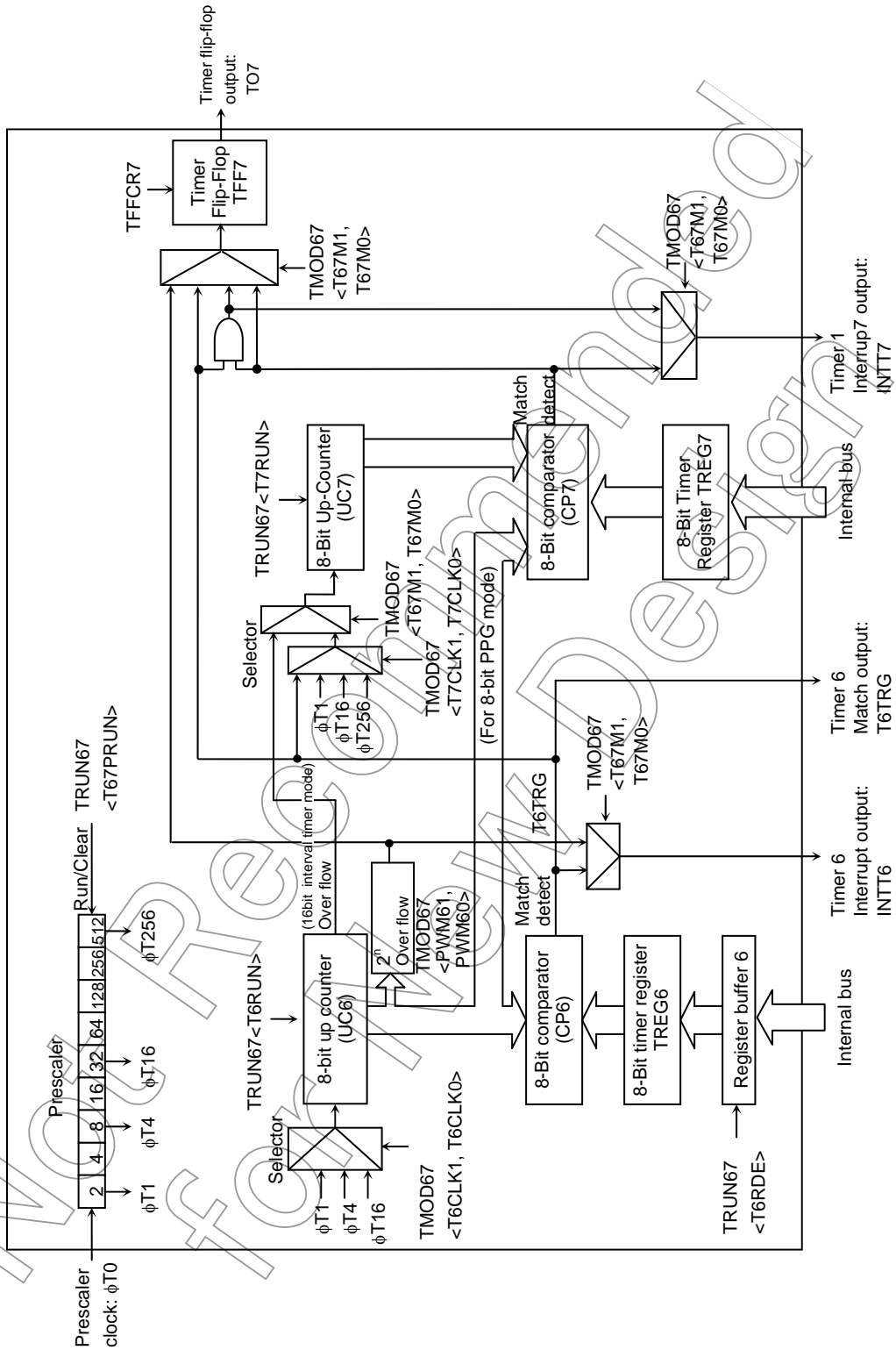
Figure 3.7.4 Timers 67 Block Diagram

3.7.2 Description of each circuit

（1）Prescaler

The 9-bit prescaler divides the 1/4 CPU clock (fc/4) to generate the input clock for timer 01.

The operation of the prescaler can be controlled using TRUN01<T01PRUN> in the timer operation control register. Setting TRUN01<T01PRUN> to 1 causes the prescaler to start counting. Setting the bit to 0 causes the prescaler to be zero-cleared and stopped.

Table 3.7.2 Prescaler Output Clock

At fc=20MHz

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

Note: The numbers in parentheses indicate the values at the maximum operating frequency.
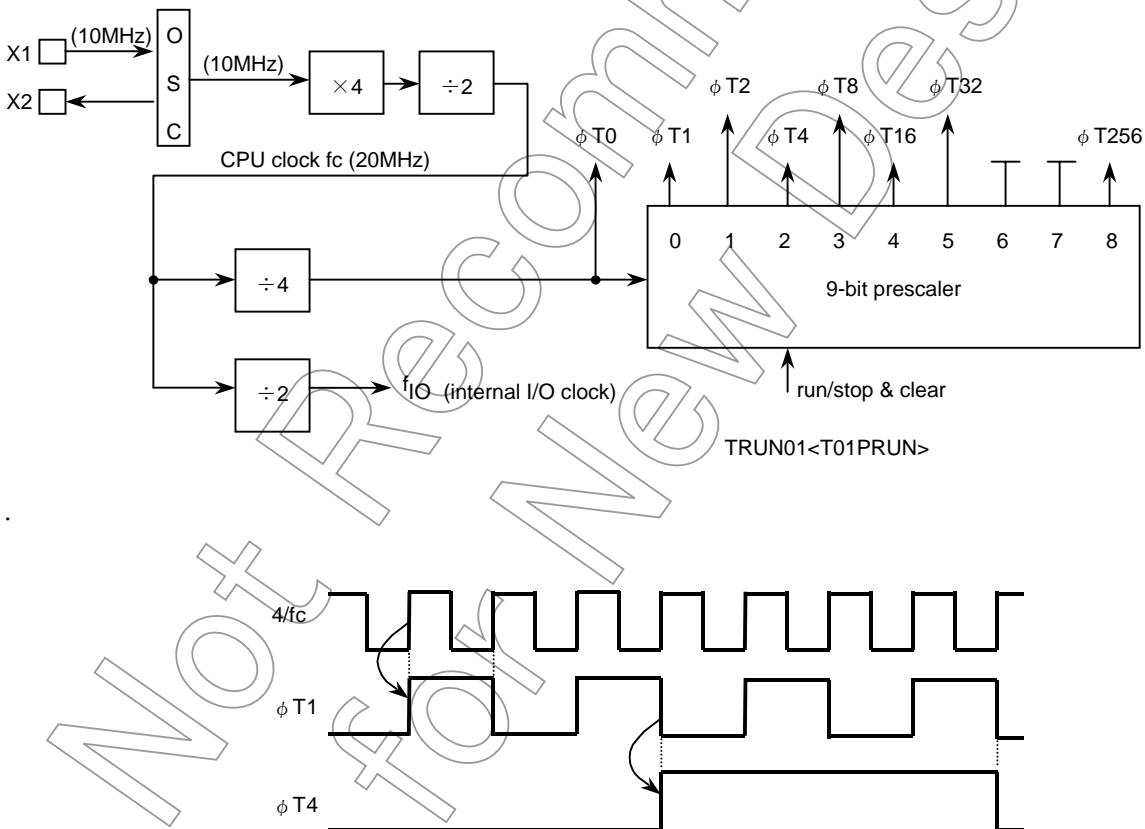
Figure 3.7.5 Prescaler

(2) Up-counters (UC0 and UC1)

The up-counters are 8-bit binary counters that increment the count according to the input clock specified with the timer mode register, TMOD01.

The input clock for UC0 is selected from among an external clock supplied through the TI0 pin and three types of prescaler output clock, $\phi T1$, $\phi T4$ and $\phi T16$, according to the setting of TMOD01<T01CLK1:0>.

The input clock for UC1 depends on the operating mode. In 16-bit timer mode, the overflow output from UC0 is used as the input clock for UC1. In other modes, the input clock is selected from among input clock $\phi T1$, $\phi T16$ and $\phi T256$ or the timer 0 comparator output (match detection).

The up-counters are set to either "stop and clear" or "count up" using TRUN01<T0RUN> and TRUN01<T1RUN>. Upon a reset, the up-counters are cleared and the timer is stopped.

(3) Timer registers (TREG0 and TREG1)

A timer register is an 8-bit register that specifies an interval time. If the up-counter value matches the value set in the timer register, the comparator match detection signal is activated. If the timer register is set to 00H, the match signal is activated when the up-counter overflows.

TREG0 is paired with a register buffer to form a double-buffer configuration.

The double buffer is controlled using TRUN01<T0RDE>. The double buffer is disabled if TRUN01<T0RDE> = 0 and enabled if TRUN01<T0RDE> = 1.

If the double buffer is enabled, data transfer from the register buffer to the timer register takes place when a 2^n overflow occurs in PWM mode or when period comparison results in a match in PPG mode. In timer mode, therefore, the double buffer cannot be used.

Upon a reset, TRUN01<T0RDE> is initialized to 0, thus disabling the double buffer. To use the double buffer, first write a value to the timer register and set <T0RDE> to 1 before writing a next setting value to the register buffer.

Figure 3.7.6 shows the configuration of TREG0.
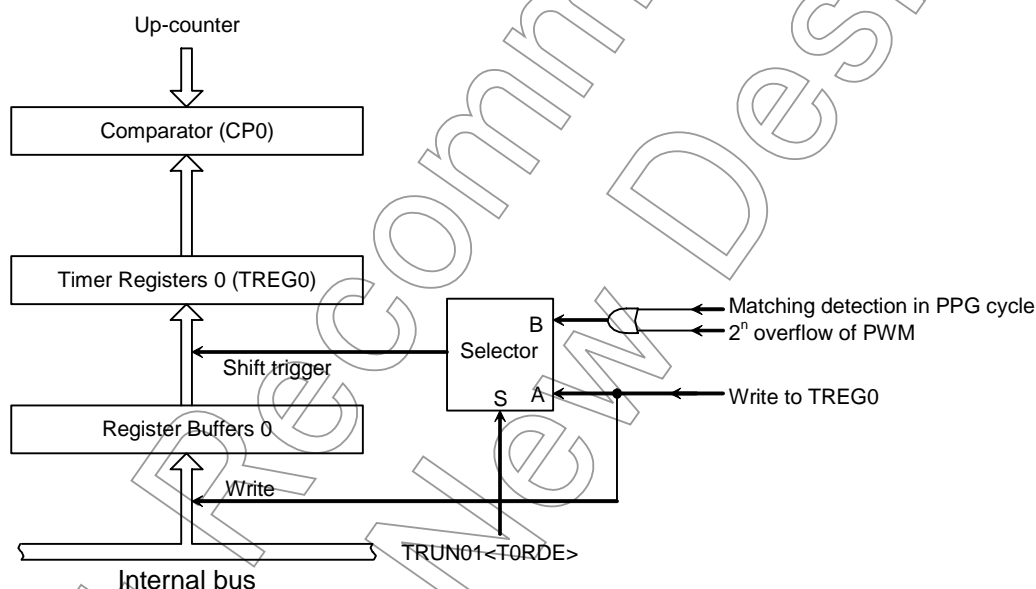
Figure 3.7.6 Configuration of TREG0

Note: The timer register and register buffer are assigned to the same address. If TRUN01<T0RDE> = 0, the same number is written to both the register buffer and timer register. If TRUN01<T0RDE> = 1, the number is only written to the register buffer.

The timer registers are located at the following addresses:

TREG0: 000082H   TREG1: 000083H

TREG2: 00008AH   TREG3: 00008BH

TREG4: 000092H   TREG5: 000093H

TREG6: 00009AH   TREG7: 00009BH

These registers are write-only and cannot be read.

(4) Comparator (CP0)

The comparator compares the up-counter value with the value set in the timer register and, if they match, clears the up-counter to 0 and issues an interrupt (INTT0-1). It also inverts the value of the timer flip-flop if inversion is enabled.

(5) Timer flip-flop (TFF1)

The timer flip-flop (TFF1) is inverted with a match detection signal from the comparator. The timer flip-flop control register, TFFCR1<TFF1IE>, enables or disables the inversion of the flip-flop.

Upon a reset, the values of TFF1 and TFF0 are initialized to 0. To set TFF1 to 1 or 0, write 01 or 10 to TFFCR1<TFF1C1:0>, respectively. Writing 00 to these bits inverts the value of TFF1 (soft inversion).

The value of TFF1 can be output through timer output pin TO1 (shared with PC1). To output the timer value, it is necessary to first set the port to enable output, using the port C function register (PCFC).

TFF is inverted when the following conditions are satisfied, depending on the mode:

8-bit interval timer mode : A match between UC0 and TREG0 or a match between UC1 and TREG1 (as selected).

16-bit interval timer mode : A match between UC0 and TREG0 and a match between UC1 and TREG1.

8-bit PWM mode : A match between UC0 and TREG0 or a 2^n overflow.

8-bit PPG mode : A match between UC0 and TREG0 or a match between UC0 and TREG1.

Note: Care should be taken when the 8-bit timer is used with a double buffer in PWM or PPG mode.

If data in the register buffer is updated immediately before an overflow occurs with a match between the up-counter value and the timer register setting, a signal having a waveform different from the set value may be output. To prevent that problem, in PWM mode, use an overflow interrupt to ensure that the register buffer update is completed more than six cycles (fc x 6) before a next overflow occurs.

Similarly, when using PPG mode, use a period comparison match interrupt to ensure that the register buffer update is completed more than six cycles before a next match in period comparison.

Example in PWM mode:

| Match between TREG0 and UC0 |
| 2^n overflow interrupt |
| TO1 |
| t_{PWM} (PWM period) |
| Point at which to modify PWM period |
| Use an overflow interrupt to rewrite the value of the timer register before a next overflow occurs. |

3.7.3 8-bit timer registers

Timers 01 Operating Control Register

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

TRUN01
(0080H)

TREG0 double buffer control

0	Disable
1	Enable

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

I2T01: Operation in IDLE2 mode (for details, see "3.3.2 Standby Controller")
T01PRUN: Prescaler operation
T1RUN: Timer 1 operation
T0RUN: Timer 0 operation

Note1: TRUN01 bits 4 to 6 return undefined values if read.

Note2: In PPG/PWM mode, <T0RDE> should be set to 1 to enable the double buffer.

Timer 23 Operation Control Register

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

TRUN23
(0088H)

TREG2 double buffer control

0	Disable
1	Enable

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

I2T23: Operation in IDLE2 mode (for details, see "3.3.2 Standby Controller")
T23PRUN: Prescaler operation
T3RUN: Timer 3 operation
T2RUN: Timer 2 operation

Note1: TRUN23 bits 4 to 6 return undefined values if read.

Note2: In PPG/PWM mode, <T2RDE> should be set to 1 to enable the double buffer.

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

Timer 45 Operation Control Register

TRUN45 (0090H)		7	6	5	4	3	2	1	0
	Bit symbol	T4RDE	-	-	-	I2T45	T45PRUN	T5RUN	T4RUN
	Read/Write	R/W				R/W			
	After Reset	0	-	-	-	0	0	0	0
	Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		

TREG4 double buffer control

0	Disable
1	Enable

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

I2T45: Operation in IDLE2 mode (for details, see "3.3.2 Standby Controller")
T45PRUN: Prescaler operation
T5RUN: Timer 5 operation
T4RUN: Timer 4 operation

Note1: TRUN45 bits 4 to 6 return undefined values if read.

Note2: In PPG/PWM mode, <T4RDE> should be set to 1 to enable the double buffer.

Timer 67 Operation Control Register

TRUN67 (0098H)		7	6	5	4	3	2	1	0
	Bit symbol	T6RDE	-	-	-	I2T67	T67PRUN	T7RUN	T6RUN
	Read/Write	R/W				R/W			
	After Reset	0	-	-	-	0	0	0	0
	Function	Double buffer 0: Disable 1: Enable				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 in IDLE2 mode (for details, see "3.3.2 Standby Controller")
T67PRUN: Prescaler operation
T7RUN: Timer 7 operation
T6RUN: Timer 6 operation

Note1: TRUN67 bits 4 to 6 return undefined values if read.

Note2: In PPG/PWM mode, <T6RDE> should be set to 1 to enable the double buffer.

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

Timer 01 Mode Register

		7	6	5	4	3	2	1	0
TMOD01 (0084H)	Bit symbol	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	0	0
	Function	Operation mode 00: 8-Bit Timer Mode 01: 16-Bit Timer Mode 10: 8-Bit PPG Mode 11: 8-Bit PWM Mode		PWM cycle 00: reserved 01: 2^6 10: 2^7 11: 2^8		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	$2^6 \times$ clock source
10	$2^7 \times$ clock source
11	$2^8 \times$ 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 : To set the TI0 pin, first set port C and then set TMOD01.

Figure 3.7.9 Register for 8-bit Timers (TMOD01)

Timer 23 Mode Register

		7	6	5	4	3	2	1	0
TMOD23 (008CH)	Bit Symbol	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	0	0
	Function	Operation mode 00: 8-Bit Timer Mode 01: 16-Bit Timer Mode 10: 8-Bit PPG Mode 11: 8-Bit PWM Mode		PWM cycle 00: reserved 01: 2^6 10: 2^7 11: 2^8		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	$2^6 \times$ clock source
10	$2^7 \times$ clock source
11	$2^8 \times$ 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)

Timer 45 Mode Register

		7	6	5	4	3	2	1	0
TMOD45 (0094H)	Bit symbol	T45M1	T45M0	PWM41	PWM40	T5CLK1	T5CLK0	T4CLK1	T4CLK0
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	0	0
	Function	Operation mode 00: 8-Bit Timer Mode 01: 16-Bit Timer Mode 10: 8-Bit PPG Mode 11: 8-Bit PWM Mode		PWM cycle 00: reserved 01: 2^6 10: 2^7 11: 2^8		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	$2^6 \times$ clock source
10	$2^7 \times$ clock source
11	$2^8 \times$ 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)
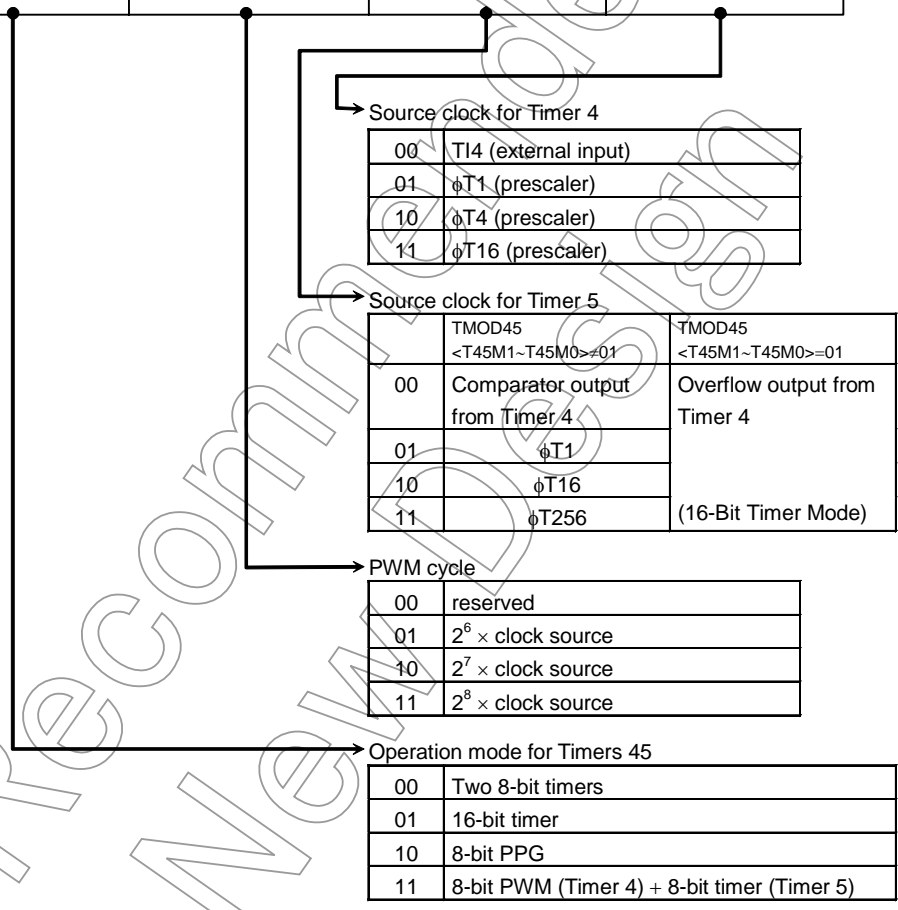
Note : To set the TI4 pin, first set port C and then set TMOD45.

Figure 3.7.11 Register for 8-bit Timers (TMOD45)

Timer 67 Mode Register

	7	6	5	4	3	2	1	0
Bit symbol	T67M1	T67M0	PWM61	PWM60	T7CLK1	T7CLK0	T6CLK1	T6CLK0
Read/Write	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-Bit Timer Mode 01: 16-Bit Timer Mode 10: 8-Bit PPG Mode 11: 8-Bit PWM Mode		PWM cycle 00: reserved 01: 2^6 10: 2^7 11: 2^8		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	

TMOD67 (009CH)

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	(16-Bit Timer Mode
11	ϕT256	

PWM cycle

00	reserved
01	$2^6 \times$ clock source
10	$2^7 \times$ clock source
11	$2^8 \times$ 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/Write					R/W			
	After Reset	-	-	-	-	1	1	0	0
Read-modify-write not allowed	Function					00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		TFF1 Control for inversion 0: Disable 1: Enable	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
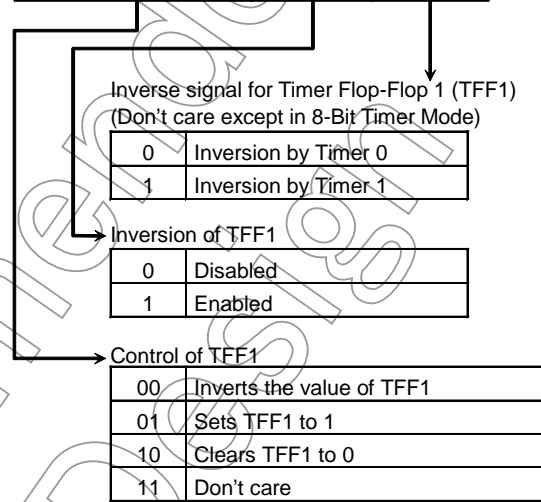
Note: TFFCR1 bits 4 to 7 return undefined values if read.

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/Write					R/W			
	After Reset	-	-	-	-	1	1	0	0
Read-modify-write not allowed	Function					00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don't care		TFF3 Control for inversion 0: Disable 1: Enable	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
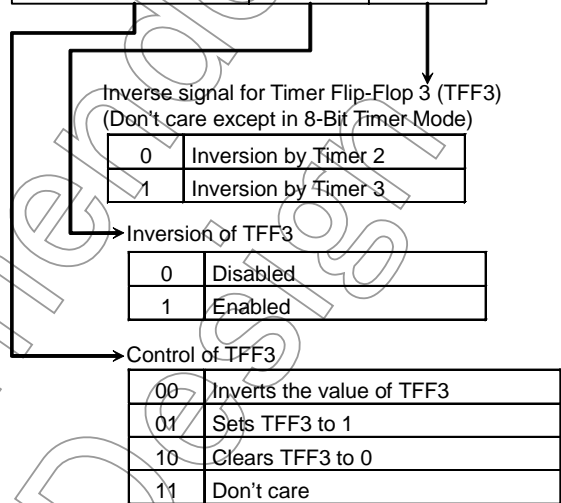
Note: TFFCR3 bits 4 to 7 return undefined values if read.

Figure 3.7.14 Register for 8-bit Timers (TFFCR3)

Timer 5 Flip-Flop Control Register

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

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
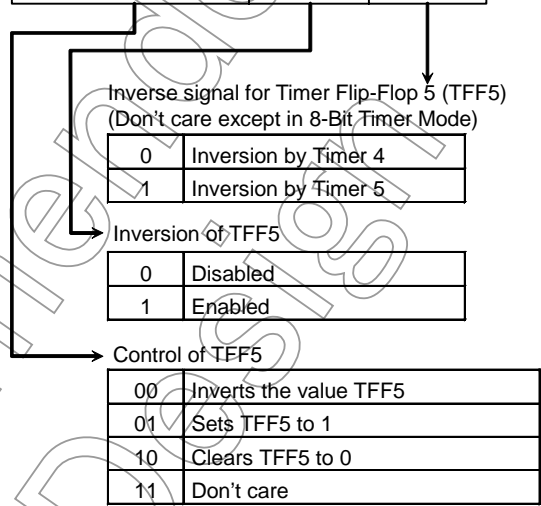
Note: TFFCR5 bits 4 to 7 return undefined values if read.

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/Write					R/W			
	After Reset	-	-	-	-	1	1	0	0
Read-modify-write not allowed	Function					00: Invert TFF7 01: Set TFF7 10: Clear TFF7 11: Don't care		TFF7 Control for invertsion 0: Disable 1: Enable	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
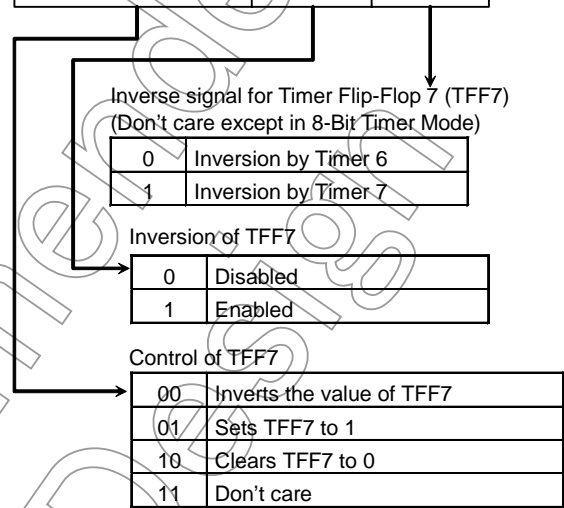
Note: TFFCR7 bits 4 to 7 return undefined values if read.

Figure 3.7.16 Register for 8-bit Timers (TFFCR7)

Timer Registers (TREG 0 to 7)

Symbol	Address	7	6	5	4	3	2	1	0
TREG0	82H (no RMW)	-							
		W							
		Undefined							
TREG1	83H (no RMW)	-							
		W							
		Undefined							
TREG2	8AH (no RMW)	-							
		W							
		Undefined							
TREG3	8BH (no RMW)	-							
		W							
		Undefined							
TREG4	92H (no RMW)	-							
		W							
		Undefined							
TREG5	93H (no RMW)	-							
		W							
		Undefined							
TREG6	9AH (no RMW)	-							
		W							
		Undefined							
TREG7	9BH (no RMW)	-							
		W							
		Undefined							

Note: The TREG registers are used for the comparator. A match between UC and TREG causes a match detection signal to be generated.
See examples in "3.7.4 Operation in Each Mode."

Figure 3.7.17 Register for 8-bit Timers (TREG0~TREG7)

3.7.4 Operation in each mode

（1） 8-bit timer mode

Each of timers 0 and 1 can be used as an independent 8-bit interval timer.

a. Generating interrupts at regular intervals (using timer 1)

To use timer 1 to generate timer 1 interrupts (INTT1) at regular intervals, first stop timer 1 and then set the operating mode, input clock and interval in TMOD01 and TREG1. Next, enable INTT1 and then start counting with timer 1.

Example: To generate INTT1 interrupts every 40 μs when fc = 20 MHz, set the registers in the following order:

```
              MSB                LSB
              7  6  5  4  3  2  1  0
  TRUN01   ←  -  X  X  X  -  -  0  -     Stop timer 1 and clear it to zero.
  TMOD01   ←  0  0  X  X  0  1  -  -     Select 8-bit timer mode and set the input clock to φT1
                                         (0.4-μs resolution, at fc = 20 MHz).
  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 the interrupt level to 5.
  TRUN01   ←  -  X  X  X  -  1  1  -     Start counting with timer 1.
```

X = Don't care   "−" = No change

TREG1 = 64H = 40 μs ÷ φT1

UC1 & TREG1 Match detect

INTT1 interrupt request occur
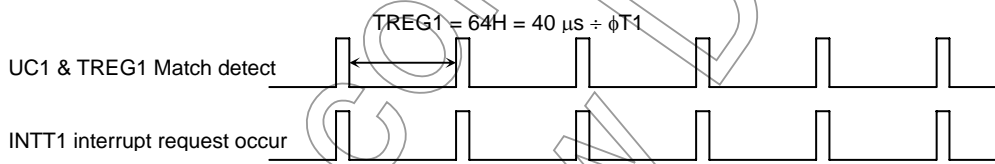
See Table 3.7.3 for how to select the input clock.

Table 3.7.3 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 available input clocks for timer 0 and timer 1 differ as follows:

Timer 0: Timer 0 input (TI0), φT1, φT4, or φT16

Timer 1: Timer 0 match detection signal (T0TRG), φT1, φT16, or φT256

b.   Outputting a square wave of 50% duty ratio

Invert the value of the timer flip-flop (TFF1) at regular intervals and output the inverted value to the timer flip-flop output pin (TO1).

Example: To output a square wave having a period of 2.4 μs when fc = 20 MHz, set the registers in the following order. Either timer 0 or timer 1 can be used for that purpose. The example uses timer 1.

```
                7  6  5  4  3  2  1  0
 ┌ TRUN01   ←  −  X  X  X  −  −  0  −        Stop timer 1 and clear it to zero.
 │ TMOD01   ←  0  0  X  X  0  1  −  −        Select 8-bit timer mode and set the input clock to φT1 (0.4
 │                                           μs, at fc = 20 MHz).
 │ TREG1    ←  0  0  0  0  0  0  1  1        Set TREG1 to 2.4 μs ÷ φT1 ÷ 2 = 3.
 │ TFFCR1   ←  X  X  X  X  1  0  1  1        Clear TFF1 to 0 and set it to be inverted with a match
 │                                           detection signal from timer 1.
 │ PCCR     ←  X  X  −  −  −  −  1  −  ┐
 │ PCFC     ←  X  X  −  −  −  −  1  −  ├     Set PC1 to the TO1 output pin.
 └ TRUN01   ←  −  X  X  X  −  1  1  −        Start counting with timer 1.
```

X = Don't care   "−" = No change
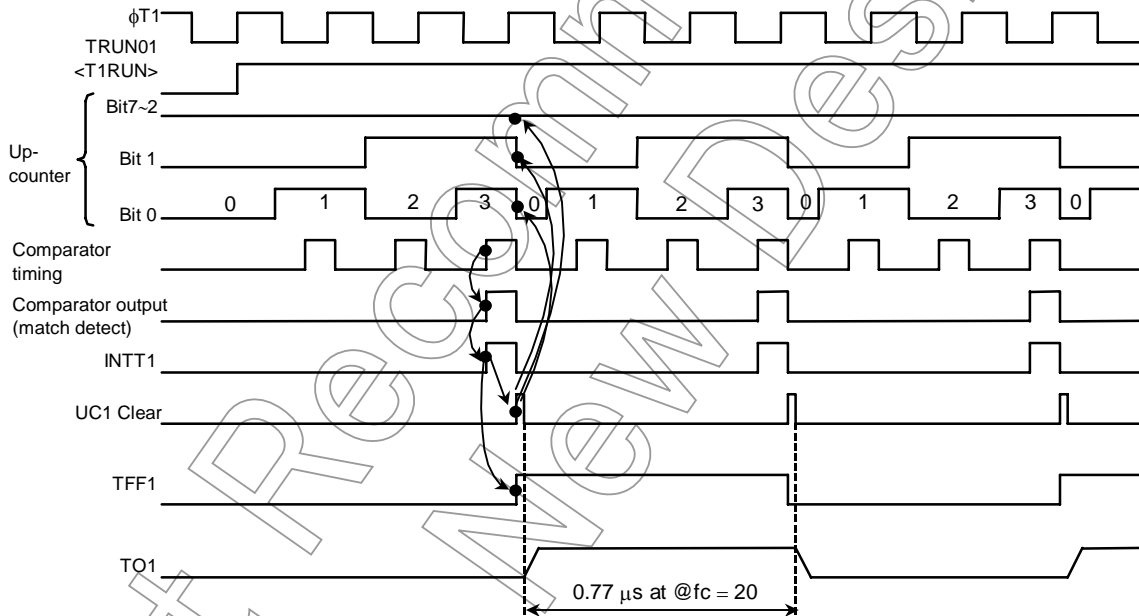
Figure 3.7.18 Square Wave Output Timing Chart (50% Duty)

c. Incrementing the timer 1 count with a match output from timer 0

Select 8-bit timer mode and set the input clock for timer 1 to the timer 0 comparator output.

Comparator output
(timer 0 match)

timer 0 up-counter
(when TREG0 = 5)

timer 1 up-counter
(when TREG1 = 2)

timer 1 match output
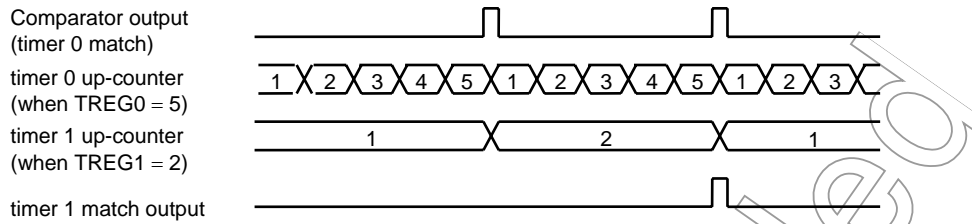
Figure 3.7.19 Timer 1 Count Up on Signal from Timer 0

(2) 16-bit timer mode

A pair of timers 0 and 1 can be used as a 16-bit interval timer. Setting TMOD01 <T01M1:0> to 01 selects 16-bit timer mode.

In 16-bit timer mode, an overflow output from timer 0 is used as the input clock for timer 1 regardless of the value of TMOD01<T1CLK1:0>. For details of relationship between the timer (interrupt) interval and input clock, see Table 3.7.3.

The lower eight bits of the timer interrupt interval are specified with timer register TREG0 and the upper eight bits with TREG1. Ensure that TREG0 is always set first because a write to TREG0 causes comparison to be temporarily disabled, after which a write to TREG1 starts comparison.
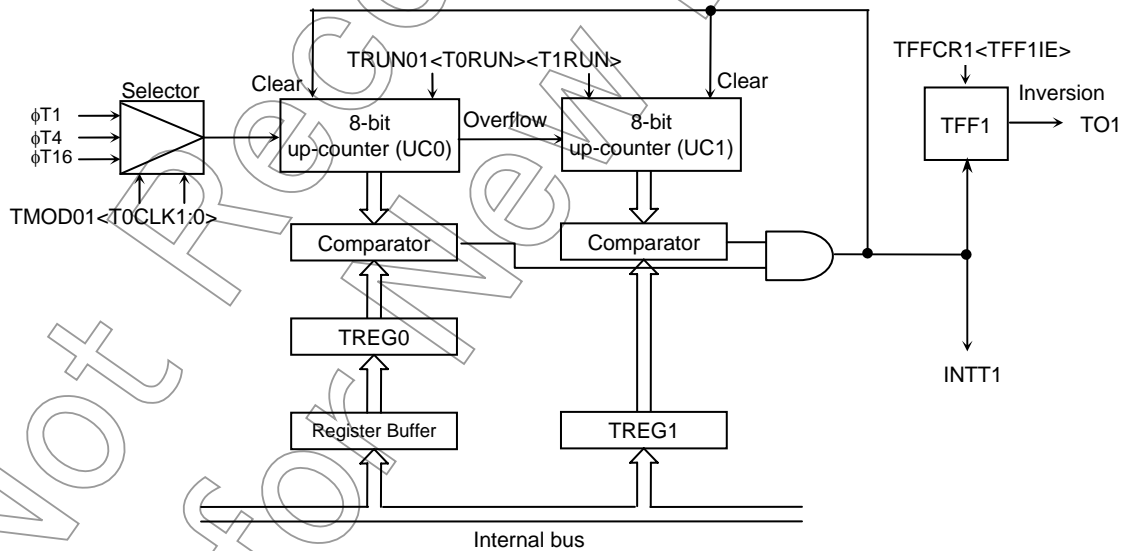
Figure 3.7.20 Block Diagram of 16-Bit Interval Timer Mode

Example: To generate INTT1 interrupts every 0.4 second when fc = 20 MHz, set the following values in timer registers TREG0 and TREG1:

If ϕT16 (6.4 μs at 20 MHz) is counted as the input clock:

0.4 s ÷ 6.4 μs = 62500 = F424H

Therefore, set TREG0 = 24H and TREG1 = F4H.

The timer 0 comparator outputs a match detection signal every time up-counter UC0 matches TREG0 but UC0 is not cleared at that time.

The timer 1 comparator outputs a match detection signal at every comparison timing where up-counter UC1 matches TREG1. If both timers 0 and 1 output match detection signals simultaneously, up-counters UC0 and UC1 are cleared to zero and an INTT1 interrupt occurs. The value of timer flip-flop TFF1 is also inverted if inversion is enabled.

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

Value of up-counter(UC1, UC0):   0000H   0080H   0180H   0280H   0380H   0480H

UC0 & TREG0
match detect signal

UC1 & TREG1
match detect signal

Interrupt INTT1

Timer output TO1   Inversion
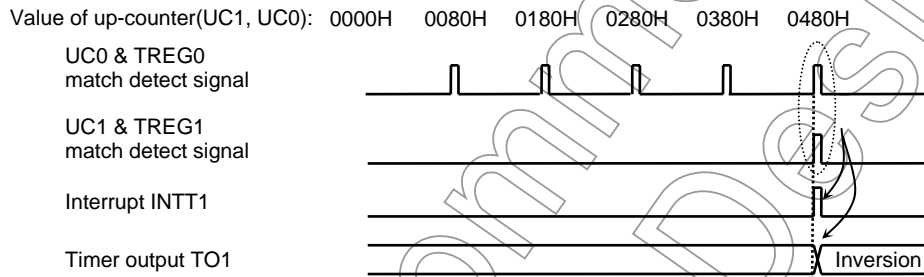
Figure 3.7.21 Timer Output by 16-Bit Interval Timer Mode

(3) 8-bit PPG (programmable square wave) output mode

Timer 0 can be used to output a square wave having any specified frequency and duty ratio. Either low-active or high-active output pulses can be selected. In this mode, timer 1 is disabled. The square wave is output through TO1 (shared with PC1).

t_H t_L

t

TREG0 and UC0 match
(Interrupt INTT0)

TREG1 and UC0 match
(Interruput INTT1)

TO1

TREG0
Duty cycle
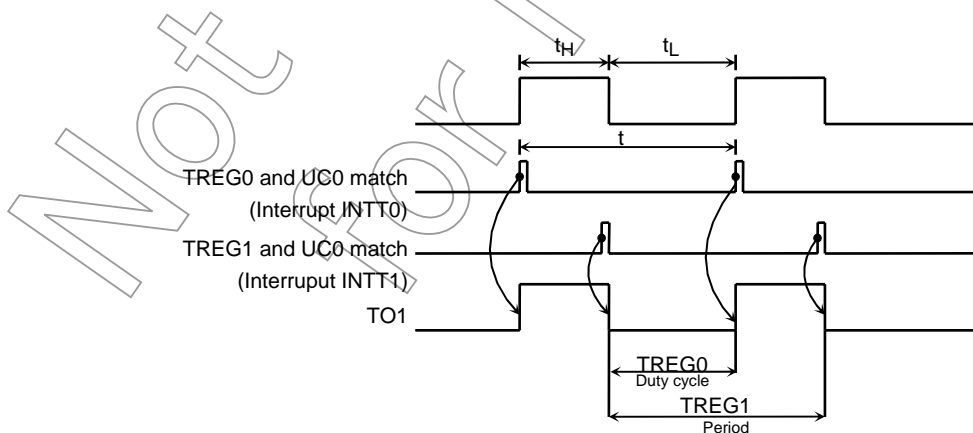
TREG1
Period

Figure 3.7.22 8 bit PPG Output Waveforms

In this mode, 8-bit up-counter UC0 inverts the timer output every time its value matches the value in timer register TREG0 or TREG1 to output a programmable square wave.

The value of TREG0 must be smaller than that of TREG1.

In this mode, the up-counter for timer 1, UC1, cannot be used. Timer 1 must, however, be set to the counting state by setting TRUN01 <T1RUN> to 1.

Figure 3.7.23 shows a block diagram of this mode:
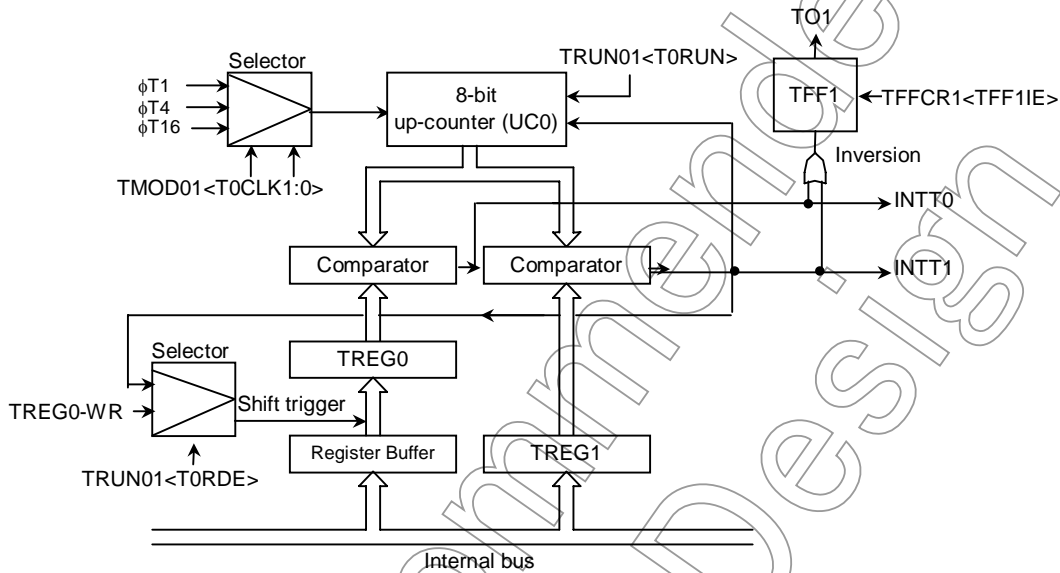
Figure 3.7.23 Block Diagram of 8-Bit PPG Output Mode

In this mode, if the double buffer for TREG0 is enabled, the value of the register buffer is shifted into TREG0 when TREG1 and UC0 match.

Using the double buffer facilitates processing for a small duty ratio (if the duty ratio is varied).
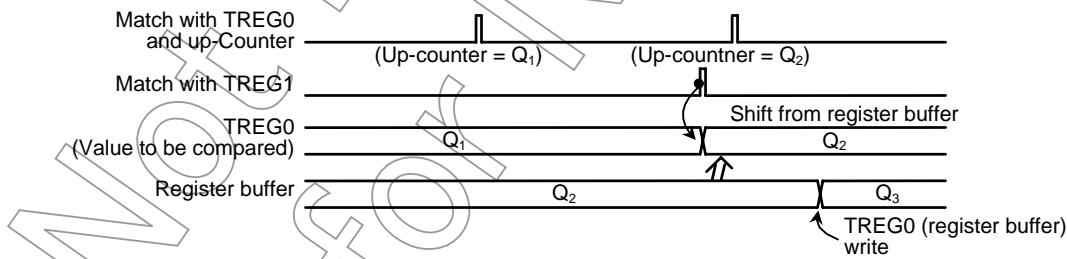
Figure 3.7.24 Operation of Register Buffer

Example: Outputting pulses having a duty ratio of 1/4 at 62.5 kHz (when fc = 20 MHz)

16 μs
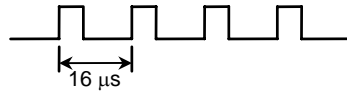
Calculate the value to set in the timer register, as follows:

To obtain a frequency of 62.5 kHz, create a waveform having a period of t = 1/62.5 kHz = 16 μs.

ϕT1 = 0.4 μs (at 20 MHz):

    16 μs ÷ 0.4 μs = 40

Therefore, TREG1 = 40 = 28H.

Next, to obtain a duty ratio of 1/4, using t x 1/4 = 16 μs x 1/4 = 4 μs:

    4 μs ÷ 0.4 μs = 10

Therefore, TREG0 = 10 = 0AH.

		7	6	5	4	3	2	1	0	
TRUN01	←	0	X	X	X	–	0	0	0	Stop timers 0 and 1 and clear them to zero.
TMOD01	←	1	0	X	X	X	X	0	1	Select 8-bit PPG mode and set the input clock to ϕT1.
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 and enable inversion.
										Setting 10 results in a negative-logic output waveform.
PCCR	←	X	X	–	–	–	–	1	–	Set PC1 to the TO1 pin.
PCFC	←	X	X	–	–	–	–	1	–	
TRUN01	←	1	X	X	X	–	1	1	1	Enable double buffer and start counting with timers 0 and 1.

X = Don't care   "–" = No change

(4) 8-bit PWM output mode

This mode is supported only for timer 0. In this mode, a PWM signal having a resolution of up to eight bits can be output. The PWM signal is output through TO1 (shared with PC1).

In this mode, timer 1 can be output as an 8-bit timer.

The timer output is inverted when the up-counter (UC0) value matches the value set in the timer register (TREG0) or when the 2^n counter overflows ($n = 6$, 7 or 8, as specified with TMOD01<PWM01:00>). UC0 is cleared when the 2^n counter overflows.

The following conditions must be satisfied to use PWM mode:

(TREG0 setting) < (2^n counter overflow setting)
(TREG0 setting) ≠ 0
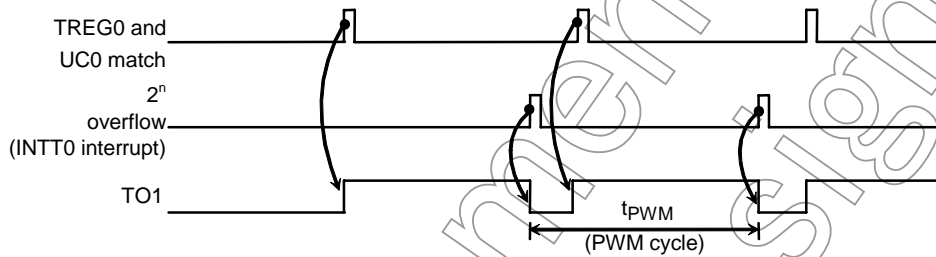
Figure 3.7.25 8-bit PWM Waveforms

Figure 3.7.26 shows a block diagram of this mode:
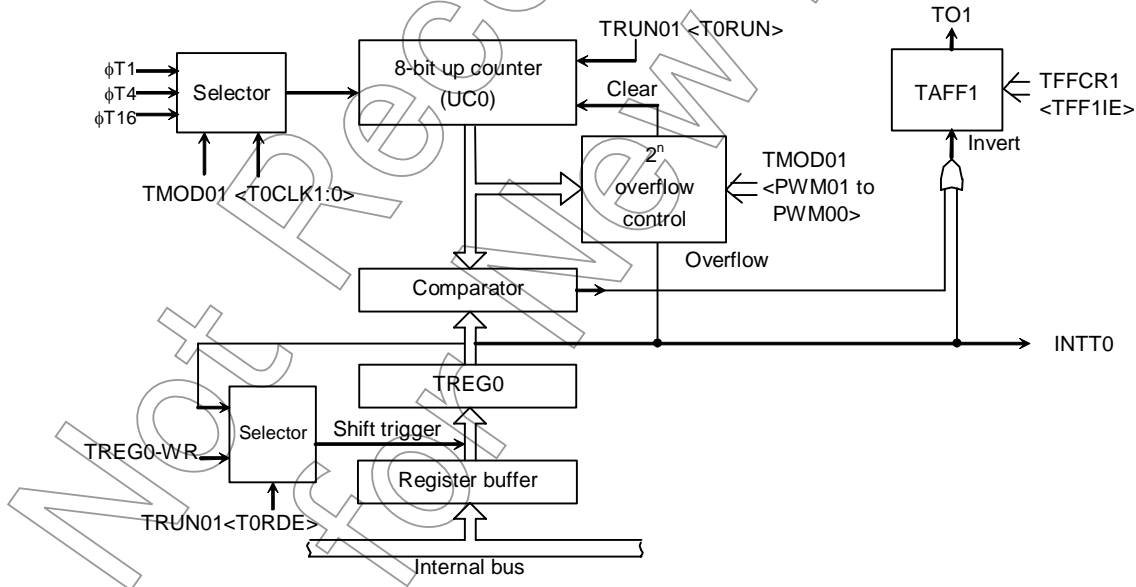
Figure 3.7.26 Block Diagram of 8-Bit PWM Mode

In this mode, if the double buffer for TREG0 is enabled, the value of the register buffer is shifted into TREG0 when an 2^n overflow is detected.

Using the double buffer facilitates processing for a small duty ratio.
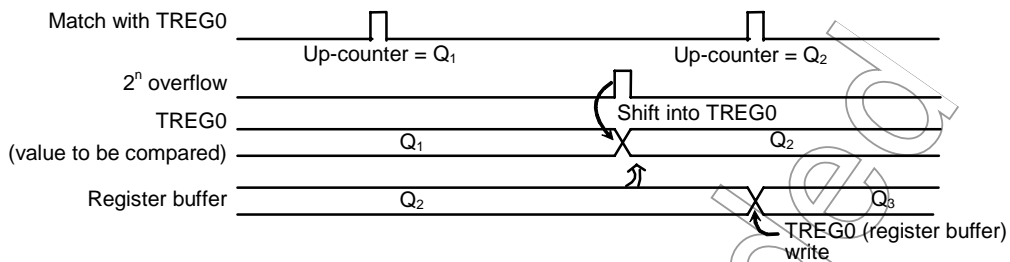
Figure 3.7.27 Register Buffer Operation

Example: Using timer 0 to output the following PWM waveform through the TO1 pin (fc = 20 MHz)
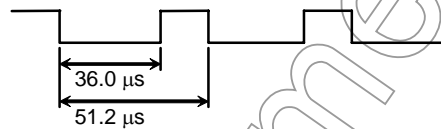
To obtain a PWM period of 51.2 μs with ϕT1 = 0.4 μs (at fc = 20 MHz):

51.2 μs ÷ 0.4 μs = 2^n = 128

Therefore, set n to 7.

Since the Low-level period is 36.0 μs, set TREG0 to the following value when ϕT1 = 0.4 μs:

36.0 μs ÷ 0.4 μs = 90 = 5AH

	MSB							LSB	
	7	6	5	4	3	2	1	0	
TRUN01	←	−	X	X	X	−	−	−	0	Stop timer 0 and clear it to zero.
TMOD01	←	1	1	1	0	−	−	0	1	Select 8-bit PWM mode (period = 2^7) and set the input clock to ϕT1.
TREG0	←	0	1	0	1	1	0	1	0	Write 5AH.
TFFCR1	←	X	X	X	X	1	0	1	X	Clear TFF1 and enable inversion.
PCCR	←	X	X	−	−	−	−	1	−	Set PC1 to the TO1 pin.
PCFC	←	X	X	−	−	−	−	1	−	
TRUN01	←	1	X	X	X	−	1	−	1	Enable double buffer and start counting with timer 0.

X = Don't care   "−" = No change

Table 3.7.4 PWM Cycle

	PWM Interval (at fc = 20MHz)		
	ϕT1	ϕT4	ϕT16
2^6	25.6 μs (39.06 kHz)	102.4 μs (9.77 kHz)	409.6 μs (2.44 kHz)
2^7	51.3 μs (19.53 kHz)	204.8 μs (4.88 kHz)	819.2 μs (1.22 kHz)
2^8	102.4 μs (9.77 kHz)	409.6 μs (2.44 kHz)	1.6384 ms (0.61 kHz)

(5) Settings for each timer mode

Table 3.7.5 shows the SFR settings for each mode.

Table 3.7.5 Interval Timer Mode Setting Registers

Register name	TMOD01				TFFCR1
<Bit Symbol>	<T01M1:0>	<PWM01:00>	<T1CLK1:0>	<T0CLK1:0>	<TFF1IS>
Function	Interval 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: Lower timer output 1: Upper timer output
16-bit interval timer mode	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, 2^7, 2^8 (01, 10, 11)	–	External clock, ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit timer × 1 channel	11	–	ϕT1, ϕT16 , ϕT256 (01, 10, 11)	–	Output disabled

"–" = Don't care

3.8 16-bit Timers/Event Counters

The TMP92CD54I contains two channels of 16-bit timers/event counters (timer 8 and timer A), which can operate in the following modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable square wave (PPG) output mode

The following operating modes are also supported by using the capture function:

- Frequency measurement mode
- Pulse width measurement mode
- Time difference measurement mode

Figure 3.8.1 and Figure 3.8.2 show block diagrams of timers 8 and A.

Each channel mainly consists of a 16-bit up-counter, two 16-bit timer registers (one with double-buffer configuration), two 16-bit capture registers, two comparators, a capture input controller, timer flip-flops and their controller. Each timer is controlled with 11 register (SFR) bytes.

The two channels, timer 8 and timer A, operate independently of each other. Both channels operate in the same way except the differences in specification listed in Table 3.8.1. This section only describes the operation of timer 8.

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 diagram
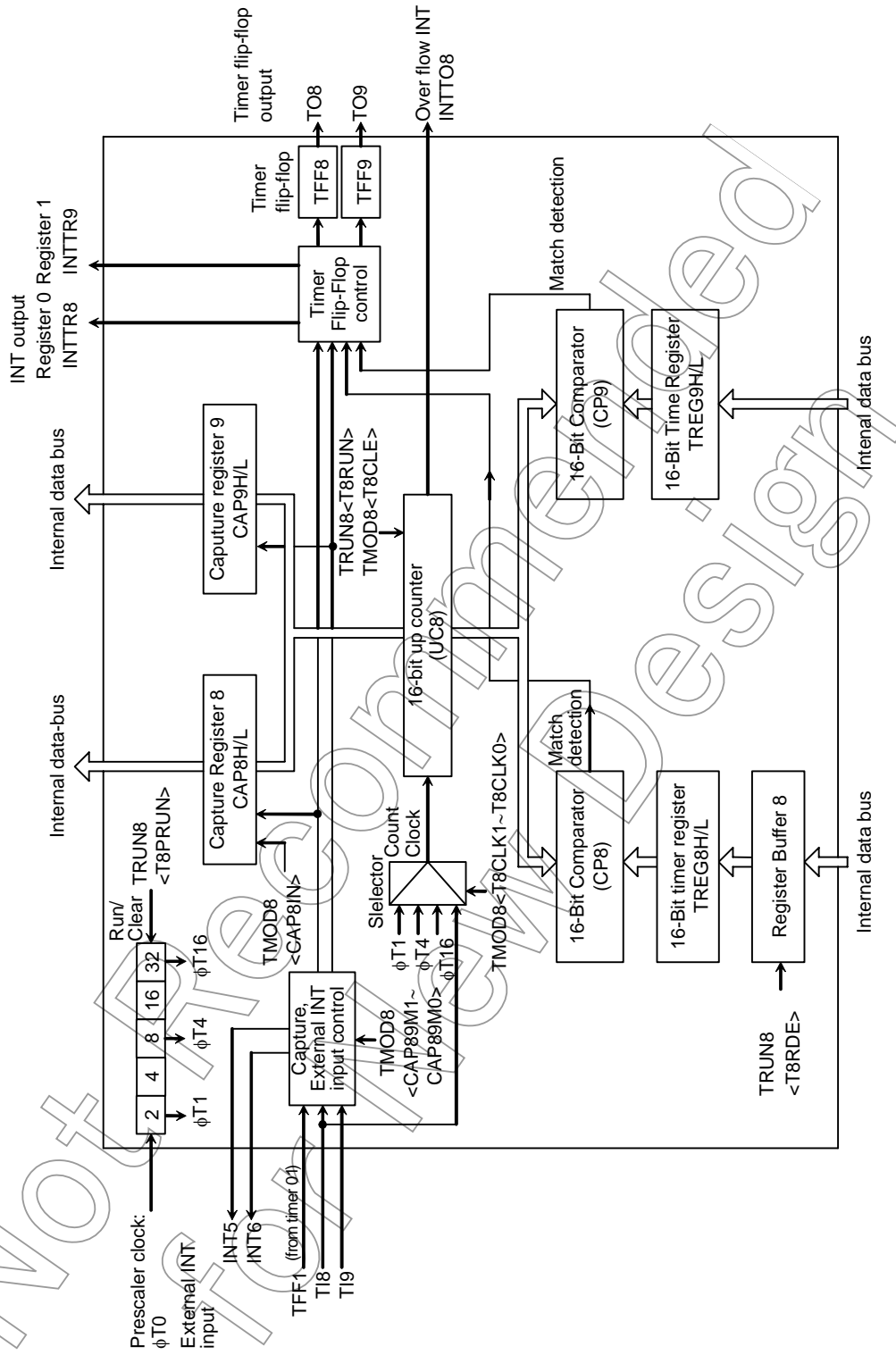
Figure 3.8.1 Block Diagram of Timer 8
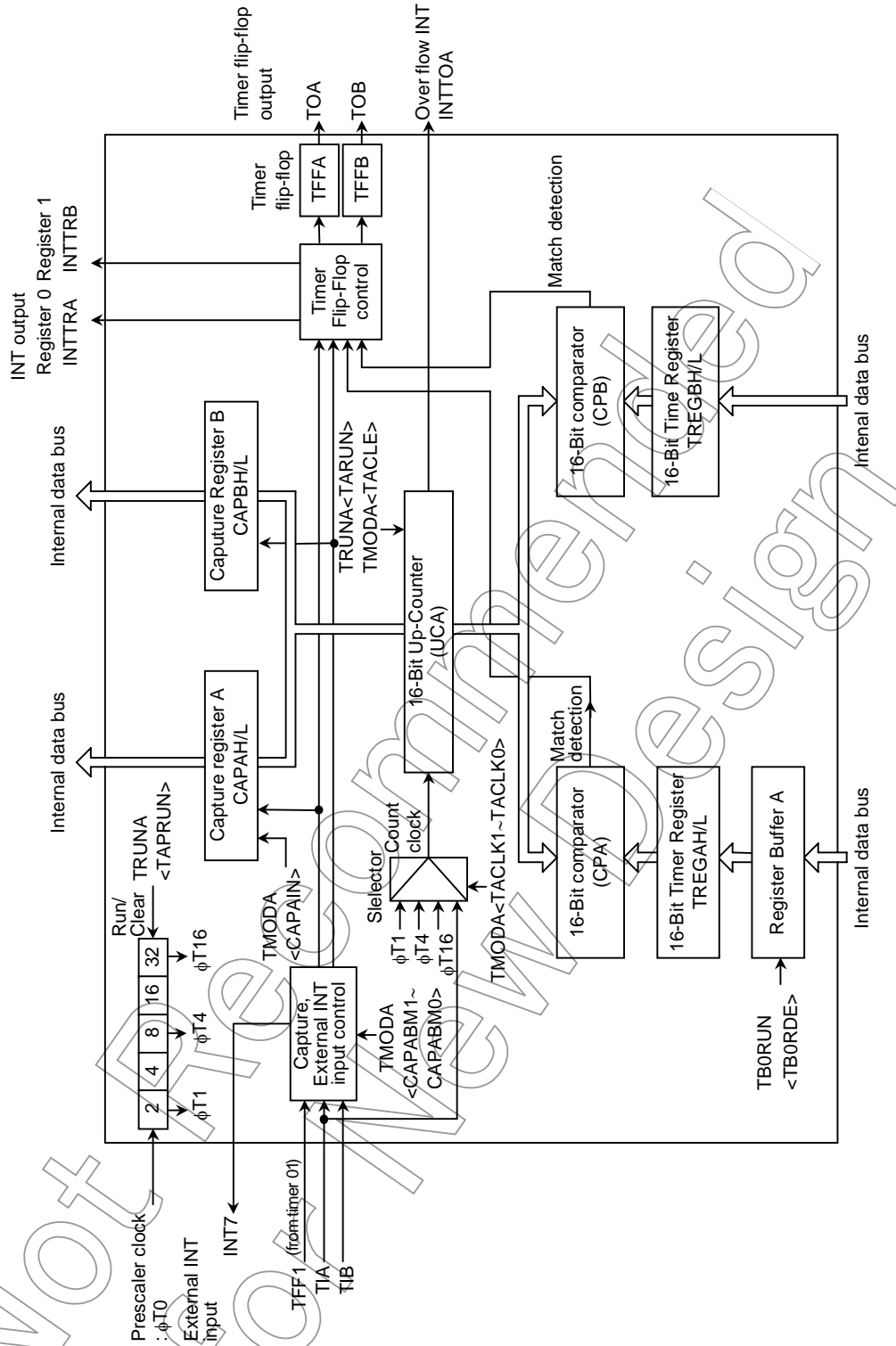
Figure 3.8.2 Block diagram of Timer A

3.8.2 Operation of each circuit

（1）Prescaler

A 5-bit prescaler provides a clock source for timer 8. The input clock for the prescaler, φT0, is obtained by dividing fc by four. The TRUN8 <T8PRUN> bit enables or stops the prescaler operation. A write of 1 to the bit causes the prescaler to start counting and a write of 0 causes it to be cleared and stopped. Table 3.8.2 shows the resolutions of prescaler output clocks.

Table 3.8.2 Prescaler Clock Resolution

At fc=20MHz

Output clock	Interval
φT1 (8/fc)	0.4 μs
φT4 (32/fc)	1.6 μs
φT16 (128/fc)	102.4 μs

（2）Up-counter (UC8)

The up-counter is a 16-bit binary counter according to the input clock specified with TMOD8 <T8CLK1, T8CLK0>.

The input clock can be selected from among φT1, φT4, φT16 and an external clock on the TI8 pin. The TRUN8 <T8RUN> bit controls counting, stopping and clearing the counter.

The up-counter, UC8, is cleared to zero when its value matches the timer register, TREG9H/L, if clearing is enabled. The TMOD8 <T8CLE> bit is used to enable or disable clearing.

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

If an overflow occurs in UC8, an overflow interrupt (INTTO8) is generated.

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

These two 16-bit registers are used to frequencies specify A comparator match detection signal is output if the value in up-counter UC8 matches that in the timer register. To set data in 16-bit timer registers TREG8H/L and TREG9H/L, use a 2-byte data transfer instruction or use two 1-byte data transfer instructions to set the lower eight bits and then the upper eight bits.

The TREG8 timer register has a double-buffer configuration and is paired with register buffer 8. The double buffer can be enabled or disabled using the timer 8 control register. The double buffer is disabled if the register bit is set to 0 and enabled if it is set to 1.

When the double buffer is enabled, a data transfer from the register buffer to the timer register takes place if the value in the up-counter (UC8) matches the value in the timer register (TREG9).

Upon a reset, the values in TREG8 and TREG9 are undefined. To use the 16-bit timer, therefore, it is necessary to first write data to the registers.

Upon a reset, TRUN8<T8RDE> is initialized to 0, thus disabling the double buffer. To use the double buffer, first write data to the timer register and set TRUN8<T8RDE> to 1 before writing next data to the register buffer.

The TREG8 and register buffer are assigned to the same address, 0000A8H / 0000A9H. If TRUN8<T8RDE> = 0, the same value is written to both TREG8 and register buffer. If TRUN8<T8RDE> = 1, the value is only written to the register buffer.

The timer registers are located at the following addresses:

Timer 8

TREG8		TREG9	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
0000A9H	0000A8H	0000ABH	0000AAH

Timer A

TREGA		TREGB	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
0000B9H	0000B8H	0000BBH	0000BAH
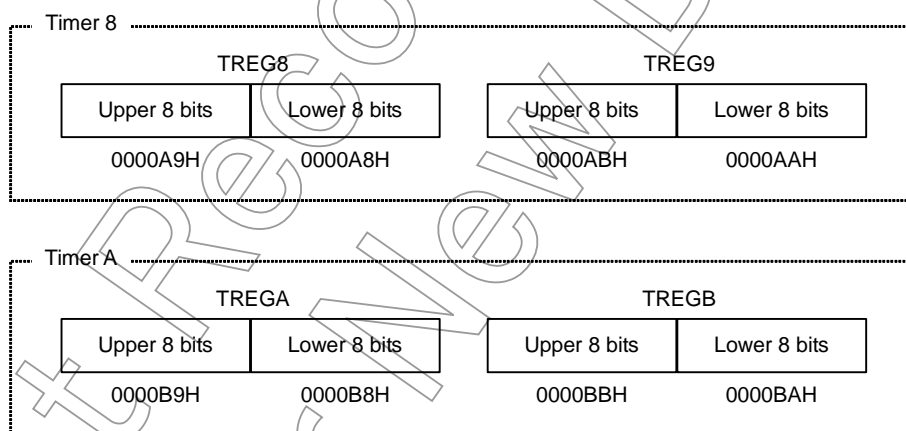
The timer registers are write-only. They cannot be read using a program.

Figure 3.8.3 Address of Timer Registers

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

The capture registers are 16-bit registers that latch the value of UC8.

To read data from a capture register, use a 2-byte data transfer instruction or use two 1-byte data transfer instructions to read the lower eight bits and then the upper eight bits.

The capture registers are located at the following addresses:

Timer 8

CAP8		CAP9	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
0000ADH	0000ACH	0000AFH	0000AEH

Timer A

CAPA		CAPB	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
0000BDH	0000BCH	0000BFH	0000BEH
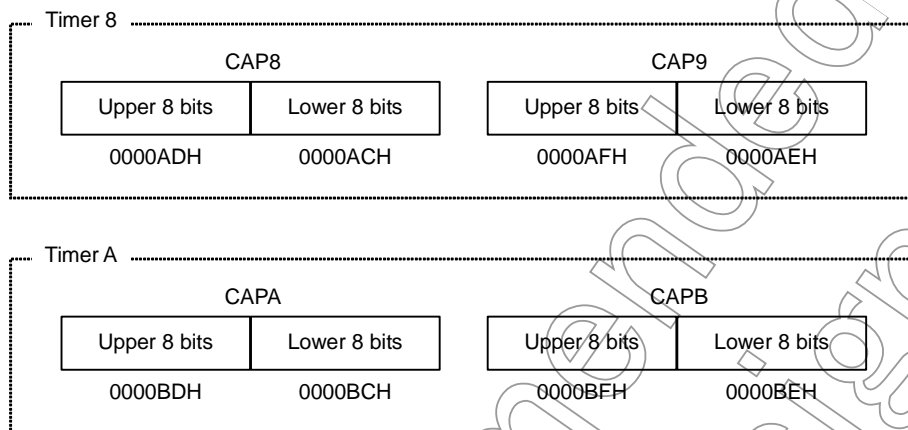
The capture registers are read-only. They cannot be written using a program.

Figure 3.8.4 Address of Cature Registers

(5) Capture input and external interrupt control

This circuit controls the timing for latching the value of up-counter UC8 into capture register CAP8 and the generation external interrupt INT5. The TMOD8 <CAP89M1, CAP89M0> bits specify the capture register interrupt timing and external interrupt edge selection. (External interrupt INT6 is fixed to the rising edge.)

The prescaler must be set to the RUN state (TRUN8 <T8PRUN> = 1).

The value of up-counter UC8 can also be latched into the capture register using software (Software capture). By software capture, writing a 0 to TMOD8 <CAP8IN> causes the current value of UC8 to be captured into capture register CAP8.

(6) Comparators (CP8 and CP9)

The 16-bit comparators compare the value in UC8 with the values set in TREG8 and TREG9 to detect a match.

Upon the detection of a match, they generate INTTR8 and INTTR9, respectively.

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

The timer flip-flops (TFF8 and TFF9) are inverted with a match detection signal from the comparator or a capture register latch signal. Inversion triggers for TFF8 and TFF9 can be controlled using TFFCR8 <CAP9T8, CAP8T8, EQ9T8, EQ8T8> and TMOD8<CAP9T9, EQ9T9>, respectively. Upon a reset, the values in TFF8 and TFF9 are undefined. Writing 00 to <TFF8C1:0> and <TFF9C1:0> triggers the inversion of the flip-flop. Writing 01 causes the flip-flop to be set to 1 while writing 10 causes it to be cleared to 0.

The values of TFF8 and TFF9 can be output through timer output pins TO8 (shared with PD2) and TO9 (shared with PD3). To output the timer value, it is necessary to first set the port to enable output, using the port D SFR.

3.8.3   16-bit timer registers

Timer 8 Operation Control Register

		7	6	5	4	3	2	1	0
TRUN8 (00A0H)	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	Double 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
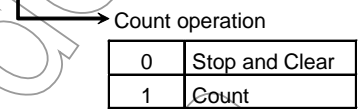
I2T8:   Operation in IDLE2 mode (for details, see "3.3.2 Standby Controller")
T8PRUN: Prescaler operation
T8RUN: Timer 8 operation

Note: TRUN8 bits 1, 4, and 5 return undefined values if read.

Timer A Operation Control Register

		7	6	5	4	3	2	1	0
TRUNA (00B0H)	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	Double 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 in IDLE2 mode (for details, see "3.3.2 Standby Controller")
TAPRUN: Prescaler operation
TARUN: Timer 8 operation

Note: TRUNA bits 1, 4, and 5 return undefined values if read.

Figure 3.8.5 Registers for 16-bit Timers (TRUN8, TRUNA)

Timer 8 Mode Register

	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

TMOD8 (00A2H)

Function:

TFF9 inversion
0: Disable trigger
1: Enable trigger

| Invert when the UC value is captured to CAP9. | Invert when the UC value matches the value in TREG9. |

Execute software capture
0: Execute
1: Don't care
Note) Always read as 1.

Capture timing
00: Disable INT5 occurs on TI8 rising edge.
01: CAP8:TI8 ↑ CAP9:TI9 ↑ INT5 occurs on TI8 rising edge.
10: CAP8:TI8 ↑ CAP9:TI8 ↓ INT5 occurs on TI8 falling edge.
11: CAP8: TFF1 ↑ CAP9: TFF1 ↓ INT5 occurs on TI8 rising edge.

Control up-counter
0: Disable clearing
1: Enable clearing

Timer 8 source clock
00: TI8 pin
01: φT1
10: φT4
11: φT16

Timer 8 source clock

00	TI8 pin
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 control
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 3.8.6 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 Note) Always read as 1.	Capture timing 00: Disable INT7 occurs on TIA rising edge. 01: CAPA:TIA↑ CAPB:TIB↑ INT7 occurs on TIA rising edge. 10: CAPA:TIA↑ CAPB:TIA↓ INT7 occurs on TIA falling edge. 11: CAPA: TFFA↑ CAPB: TFFA↓ INT7 occurs on TIA 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	TIA pin
01	φT1
10	φT4
11	φT16

Up-counter clear control

0	Disable
1	Enable clearing on match with TREGB.

Capture/Interrupt timing

	Capture control	INT7 control
00	Disable	INT7 occurs on rising edge of TIA.
01	CAPA at TIA rise CAPB at TIB rise	INT7 occurs on rising edge of TIA.
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 3.8.7   Registers for 16-bit Timers (TMODA)

Timer 8 Flip-Flop Control Register

	7	6	5	4	3	2	1	0
TFFCR8 (00A3H) 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 Note)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 Note)Always read as 11	
			Invert when the UC value is loaded into CAP9.	Invert when the UC value is loaded into 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
1	Enable trigger

TFF8 Inverted when the UC value is matched to TREG9.

0	Disable trigger
1	Enable trigger

TFF8 Inverted when the UC value is loaded to CAP8.

0	Disable trigger
1	Enable trigger

TFF8 Inverted when the UC value is loaded to CAP9.

0	Disable trigger
1	Enable trigger

TFF9 control

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

Figure 3.8.8 Registers for 16-bit Timers (TFFCR8)

Timer A Flip-Flop Control Register

		7	6	5	4	3	2	1	0
TFFCRA (00B3H)	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 Note)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 Note)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
1	Enable trigger

TFFA Inverted when the UC value matches to TREGB.

0	Disable trigger
1	Enable trigger

TFFA Inverted when the UC value is loaded to CAPA.

0	Disable trigger
1	Enable trigger

TFFA Inverted when the UC value is loaded to CAPB.

0	Disable trigger
1	Enable trigger

TFFB control

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

Figure 3.8.9　Registers for 16-bit Timers (TFFCRA)

Timer Registers (Timer 8 and Timer A)

Symbol	Address	7	6	5	4	3	2	1	0
TREG8L (no RMW)	A8H	-							
		W							
		Undefined							
TREG8H (no RMW)	A9H	-							
		W							
		Undefined							
TREG9L (no RMW)	AAH	-							
		W							
		Undefined							
TREG9H (no RMW)	ABH	-							
		W							
		Undefined							
TREGAL (no RMW)	B8H	-							
		W							
		Undefined							
TREGAH (no RMW)	B9H	-							
		W							
		Undefined							
TREGBL (no RMW)	BAH	-							
		W							
		Undefined							
TREGBH (no RMW)	BBH	-							
		W							
		Undefined							

Capture Registers (Timer 8 and Timer A)

Symbol	Address	7	6	5	4	3	2	1	0
CAP8L	ACH	-							
		R							
		Undefined							
CAP8H	ADH	-							
		R							
		Undefined							
CAP9L	AEH	-							
		R							
		Undefined							
CAP9H	AFH	-							
		R							
		Undefined							
CAPAL	BCH	-							
		R							
		Undefined							
CAPAH	BDH	-							
		R							
		Undefined							
CAPBL	BEH	-							
		R							
		Undefined							
CAPBH	BFH	-							
		R							
		Undefined							

Figure 3.8.10 Registers for 16-bit Timers. (TREG8 to B (L/H), CAP8 to B (L/H))

3.8.4 Operation in each mode

（1） 16-bit interval timer mode

Generating interrupts at regular intervals

Set an interval time in timer register TREG9 to generate an INTTR9 interrupt.

```
            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, set the level to 4, and disable INTTR8.
TFFCR8   ←  1  1  0  0  0  0  1  1     Disable trigger.
TMOD8    ←  0  0  1  0  0  1  *  *     Set the input clock to the prescaler output clock and
            (** = 01, 10, 11)         disable the capture function.
TREG9    ←  *  *  *  *  *  *  *  *     Set an interval time (16 bits).
            *  *  *  *  *  *  *  *
TRUN8    ←  0  0  X  X  -  1  X  1     Start timer 8.
```

X = Don't care   "−" = No change

（2） 16-bit event counter mode

The 16-bit timer can function as an event counter by using an external clock (supplied on the TI8 pin) as the input clock.

The up-counter is incremented on the rising edge of the TI8 pin input. The count value can be obtained by performing a software capture and then reading the captured value.

```
            7  6  5  4  3  2  1  0
TRUN8    ←  0  0  X  X  -  0  X  0     Stop timer 8.
PDCR     ←  -  -  -  -  -  -  -  1     Set PD0 to TI8.
PDFC     ←  -  -  -  -  -  -  -  1
INTET89  ←  X  1  0  0  X  0  0  0     Enable INTTR9, set the level to 4, and disable INTTR8.
TFFCR8   ←  1  1  0  0  0  0  1  1     Disable trigger.
TMOD8    ←  0  0  1  0  0  1  0  0     Set the input clock to the TI8 pin input.
TREG9    ←  *  *  *  *  *  *  *  *     Set the count value (16 bits).
            *  *  *  *  *  *  *  *
TRUN8    ←  0  0  X  X  -  1  X  1     Start timer 8.
```

X = Don't care   "-" = No change

Note: The prescaler must also be set to "Run" mode (TRUN8 <T8PRUN> = "1") when the timer is used as an event counter.
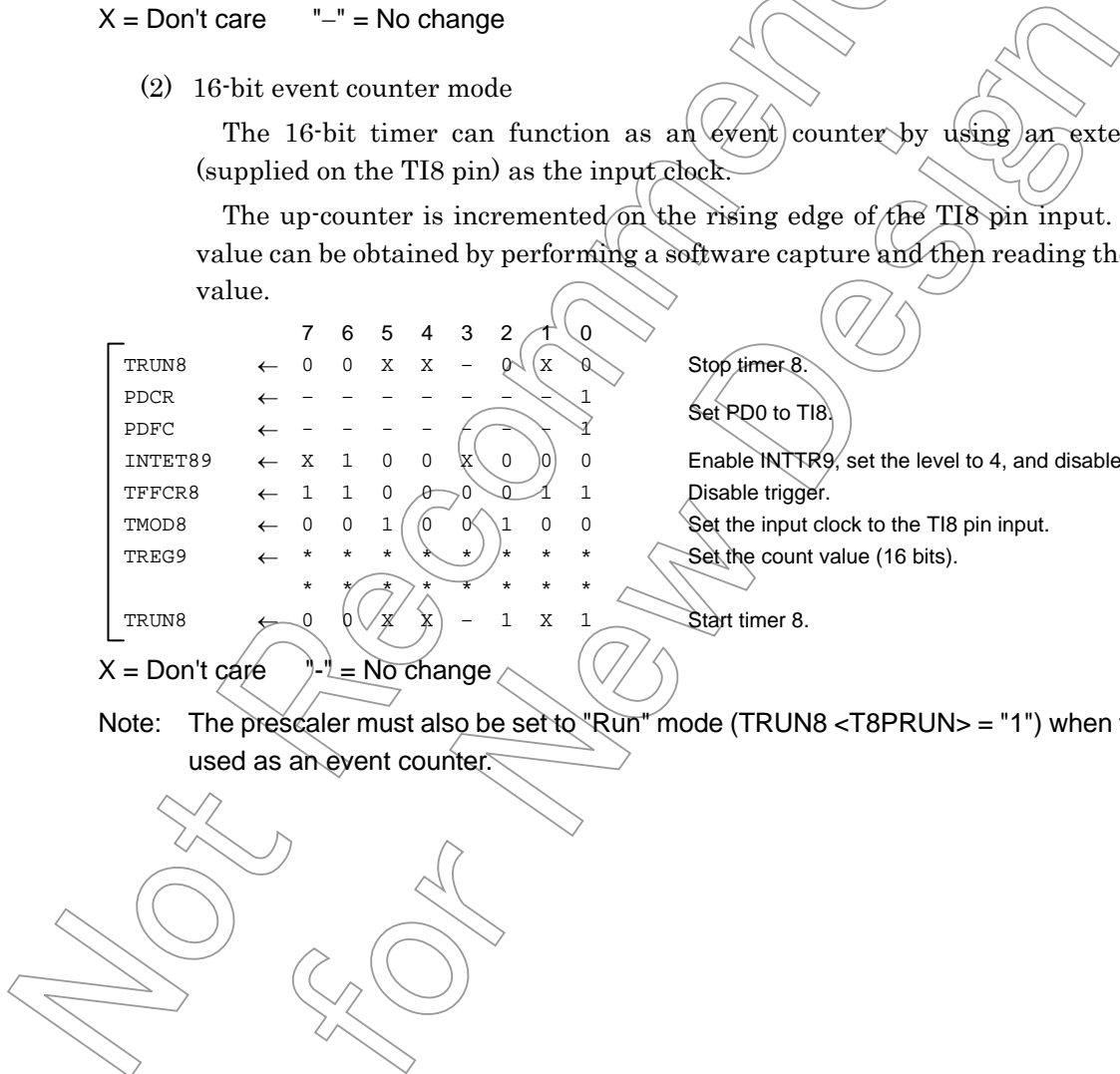
(3) 16-bit PPG (programmable square wave) output mode

In this mode, the timer can be used to output a square wave having any specified frequency and duty ratio (programmable square wave). Either low-active or high-active output pulses can be selected.

The inversion of timer flip-flop TFF8 is triggered with a match between UC8 and the timer register (TREG8 and TREG9) setting, resulting in a programmable square wave being output through the TO8 pin. The values of TREG8 and TREG9 must satisfy the following condition:

(TREG8 value) < (TREG9 value)

Match with TREG8
(INTTR8 inerrupt)

Match with TREG9
(INTTR9 interrupt)

TO8 pin
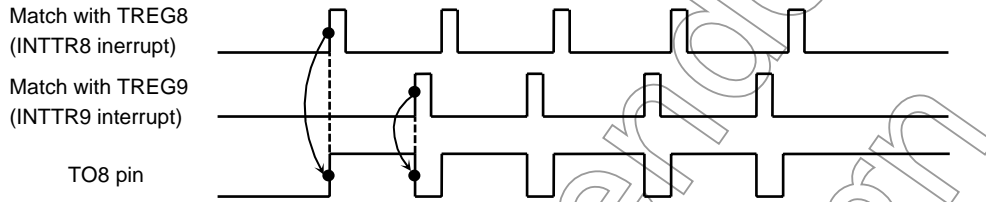
Figure 3.8.11 Programmable Pulse Generation (PPG) Output Waveforms

In this mode, if the double buffer for TREG8 is enabled, the value of register buffer 8 is shifted into TREG8 upon a match with TREG9. Using the double buffer facilitates processing for a small duty ratio.

Match with TREG8

Up-counter = Q_1 Up-counter = Q_2

Match with TREG9

Shift into theTREG9

TREG8
(value to be compared) Q_1 Q_2

Register buffer Q_2 Q_3

Write into the TREG8
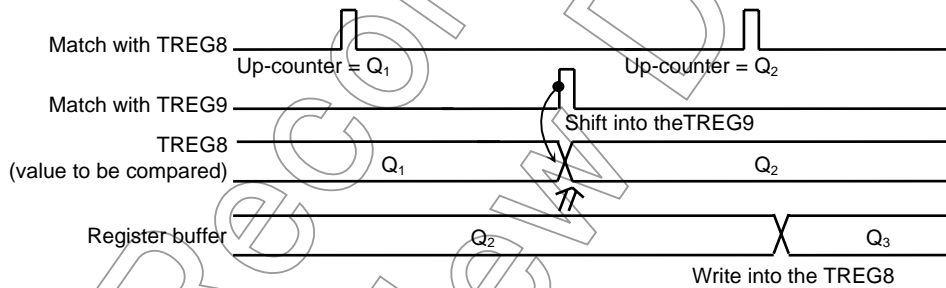
Figure 3.8.12 Operation of Register Buffer

The following shows a block diagram of this mode:
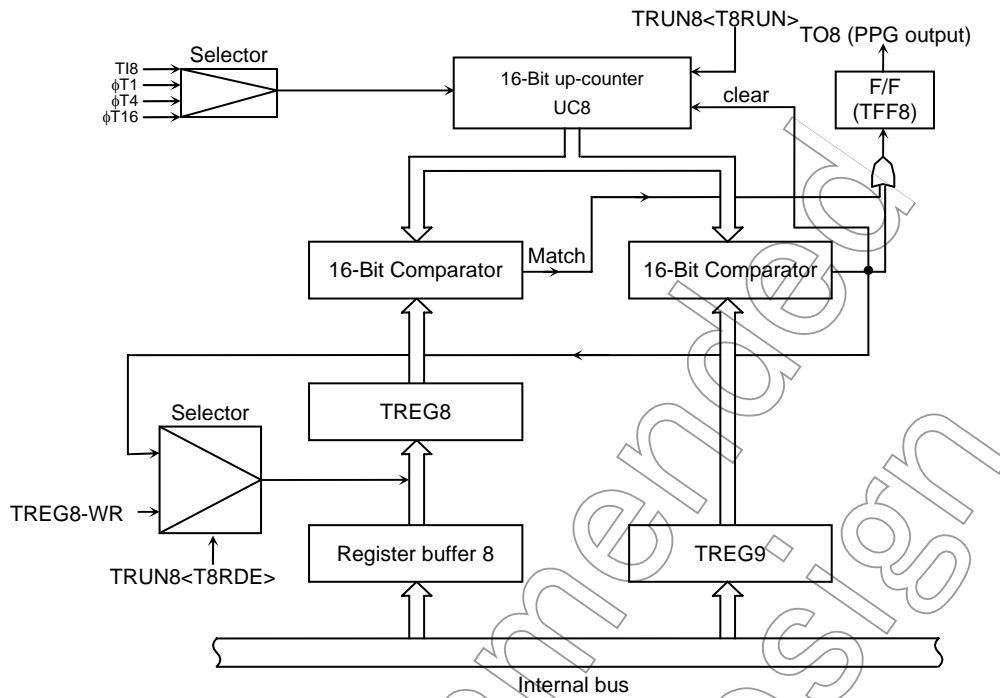
Figure 3.8.13 Block Diagram of 16-bit PPG Output Mode

In 16-bit PPG output mode, set the registers as follows:

```
              7   6   5   4   3   2   1   0
TRUN8    ←    0   0   X   X   –   0   X   0      Disable TREG8 double buffer and stop timer 8.
TREG8    ←    *   *   *   *   *   *   *   *      Set a duty ratio (16 bits).
              *   *   *   *   *   *   *   *
TREG9    ←    *   *   *   *   *   *   *   *      Set a frequency (16 bits).
              *   *   *   *   *   *   *   *
TRUN8    ←    1   0   X   X   –   0   X   0      Enable TREG8 double buffer.
                                                (Duty ratio/frequency modified with an INTTR9 interrupt)
TFFCR8   ←    X   X   0   0   1   1   1   0      Set TFF8 to invert upon detection of a match with TREG8
                                                and TREG9. Initialize TFF8 to 0.
TMOD8    ←    0   0   1   0   0   1   *   *      Set the input clock to the prescaler output clock and
                     (** = 01, 10, 11)          disable the capture function.
PDCR     ←    –   –   –   –   –   1   –   –  ⎫   Assign TO8 to the PD2 pin.
PDFC     ←    –   –   –   –   –   1   –   –  ⎬
TRUN8    ←    1   0   X   X   –   1   X   1  ⎭   Start timer 8.
```

X = Don't care; "–" = No change

(4) Examples using the capture function

The capture function can be used for many applications, including the following examples:

a. One-shot pulse output from external trigger pulse

b. Frequency measurement

c. Pulse width measurement

d. Time difference measurement

a. One-shot pulse output from external trigger pulse

Operate up-counter UC8 in free-running mode using the prescaler output clock. Supply external trigger pulses through the TI8 pin and use the capture function to latch the up-counter value into capture register CAP8 on the rising edge of a trigger pulse.

An INT5 interrupt occurs on the rising edge of an external trigger pulse. In INT5 interrupt handling, set timer register TREG8 with the sum (c + d) of the CAP8 value (c) and delay time (d). For timer register TREG9, set the sum (c + d + p) of the TREG8 value (c + d) and the width of the one-shot pulse (p). (That is, TREG8 = c + d and TREG9 = c + d + p.)

In addition, set <EQ9T8, EQ8T8> to 11 to enable a trigger so that timer flip-flop TFF8 will be inverted upon a match between UC8 and TREG8 as well as a match between UC8 and TREG9.

Once a one-shot pulse has been output, use the INTTR9 interrupt handling to redisable a trigger.

The symbols (c), (d), and (p) in the above description correspond to symbols c, d, and p in Figure 3.8.14, "One-shot pulse output from an external trigger pulse (with delay)."
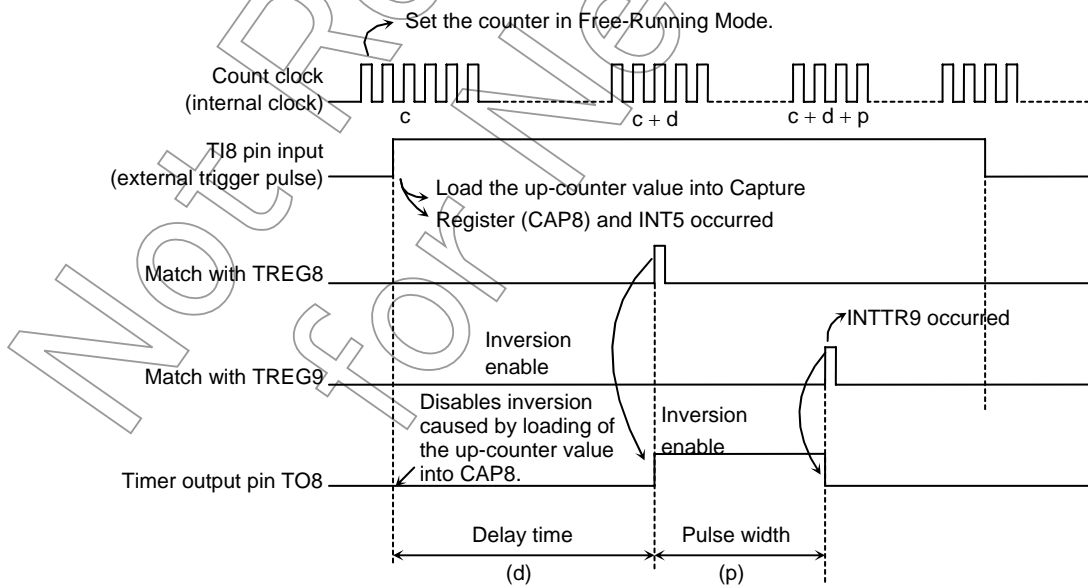
Figure 3.8.14 One-shot Pulse Output (with delay)

Example settings：Outputting a one-shot pulse of 2 ms with a 3-ms delay using an external trigger pulse on the TI8 pin

Main settings

```
TMOD8    ←  X  X  1  0  1  0  0  1
TFFCR8   ←  X  X  0  0  0  0  1  0
PDCR     ←  -  -  -  -  -  1  -  -
PDFC     ←  -  -  -  -  -  1  -  -
INTE56   ←  X  -  -  -  X  1  0  0
INTET89  ←  X  0  0  0  X  0  0  0
TRUN8    ←  -  0  X  X  -  1  X  1
```

- Set the counter to free-running mode.
- Start counting with ϕT1.
- Capture into CAP8 on the rising edge of TI8 input.
- Clear TFF8 to zero.
- Disable TFF8 inversion.
- Assign TO8 to PD2.
- Enable INT5, set the level to 4, and disable INTTR8 and INTTR9.
- Start timer 8.

Setting of INT5

```
TREG8    ←  CAP8 + 3 ms/ϕT1
TREG9    ←  TREG8 + 2 ms/ϕT1
TFFCR8   ←  X  X  -  -  1  1  -  -
INTET89  ←  X  1  0  0  X  -  -  -
```

- Enable TFF8 inversion upon a match with TREG8 and TREG9.
- Enable INTTR9.

Setting INTTR9

```
TFFCR8   ←  X  X  -  -  0  0  -  -
INTET89  ←  X  0  0  0  X  -  -  -
```

- Disable TFF8 inversion upon a match with TREG8 and TREG9.
- Disable INTTR9.

X = Don't care; "–" = No change

    If a delay time is not necessary, invert timer flip-flop TFF8 when the counter value is captured into CAP8 and use an INT5 interrupt to set timer register TREG9 with the sum (c + p) of the CAP8 value (c) and the one-shot pulse width (p). Enable TFF8 so that it will be inverted upon a match between TREG9 and UC8. And upon an INTTR9 interrupt, redisable TFF8.
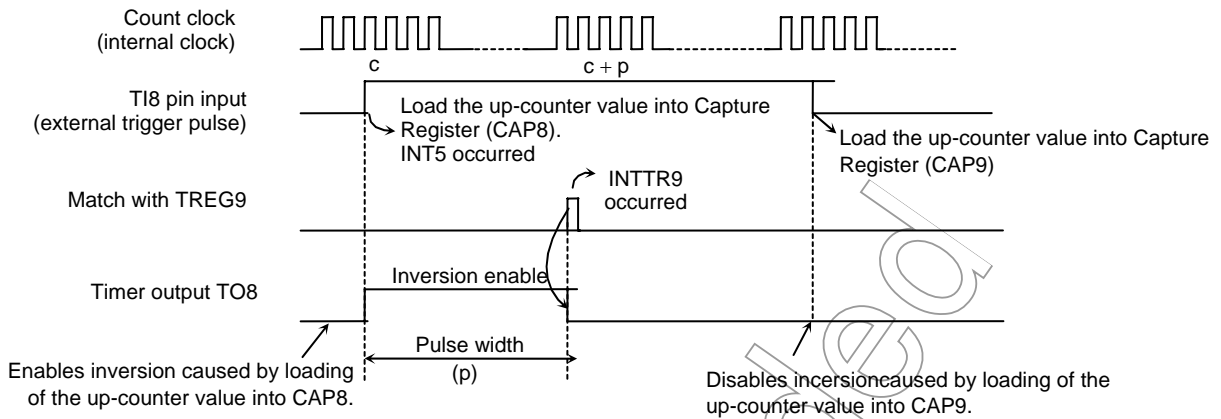
Figure 3.8.15 One-shot Pulse Output (without delay)

b. Frequency measurement

In this mode, the timer is used to measure the frequency of an external clock. Supply an external clock through the TI8 pin and measure it using 8-bit timers (timers 0 and 1) and a 16-bit timer/event counter (timer 8). Set the timer 8 input clock to the TI8 input and set TMOD8 <CAP89M1, CAP89M0> to 11. Capture the value of up-counter UC8 into CAP8 on the rising edge of timer flip-flop TFF1 for the 8-bit timers (timers 0 and 1) and into CAP9 on its falling edge.

Use 8-bit timer interrupts (INTT0 and INTT1) to obtain the frequency from the difference between the values in capture registers CAP8 and CAP9.
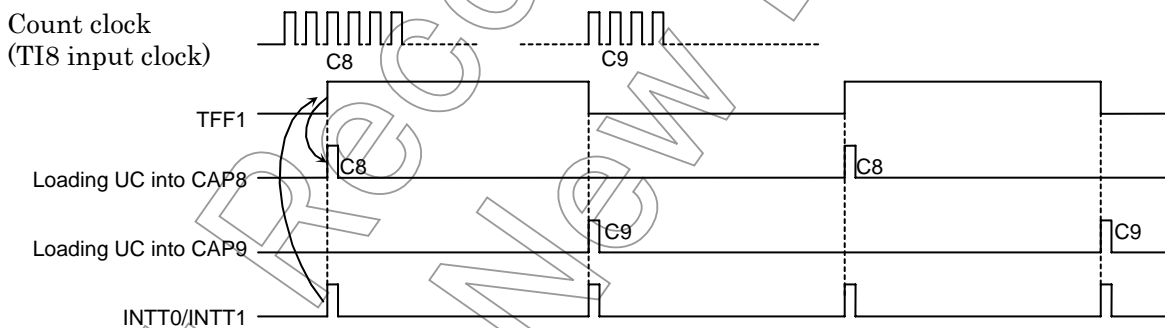
Figure 3.8.16 Frequency Measurement

For example, if the TFF1 "1" level width is set to 0.5 s with the 8-bit timers and the difference between CAP8 and CAP9 is 100, then the frequency is $100 \div 0.5$ s = 200 Hz.

c. Pulse width measurement

In this mode, the timer is used to measure the High-level width of an external pulse. Supply an external pulse through the TI8 pin and operate the 16-bit timer/event counter in free-running mode using the internal clock. Use the capture function to trigger capturing on both the rising and falling edges of an external pulse to capture the value of the up-counter (UC8) into capture registers CAP8 and CAP9. An INT5 interrupt occurs on the falling edge of the TI8 pin input.

The pulse width can be determined from the difference between CAP8 and CAP9 and the period of the internal clock.

For example, if the prescaler output clock period is 0.8 µs and the difference between CAP8 and CAP9 is 100, then the pulse width is 100 x 0.8 [µs] = 80 [µs].

Software-based processing is necessary if the pulse width to be measured exceeds the maximum count time for UC8.
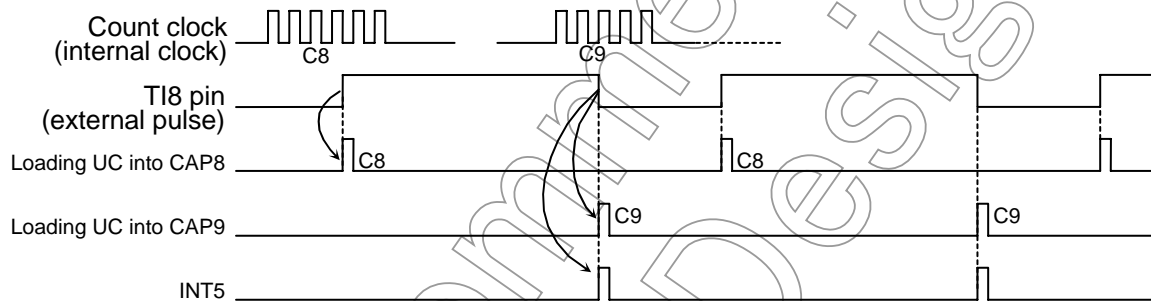
Figure 3.8.17 Pulse Width Measurement

Note: Only in pulse width measurement mode (TMOD8 <CAP89M1:0> = 10), an INT5 external interrupt occurs on the falling edge of the TI8 input. It occurs on the rising edge in all other modes.

To measure the Low-level width, multiply the period of the prescaler output clock by the difference between the first C9 value and the second C8 value in the INT5 interrupt handling.

d. Time difference measurement

In this mode, the timer is used to measure the time difference between the rising edges of an external pulse input on the TI8 and TI9 pins. Operate the 16-bit timer/event counter (timer 8) in free-running mode using the internal clock and capture the value of up-counter UC8 into capture register CAP8 on the rising edge of the TI8 input, at which time an INT5 interrupt occurs.

Similarly, capture the value of up-counter UC8 into capture register CAP9 on the rising edge of the TI9 input, at which time an INT6 interrupt occurs.

Once the values have been captured into the capture registers, the time difference can be determined by multiplying the period of the internal clock by the difference between CAP9 and CAP8.
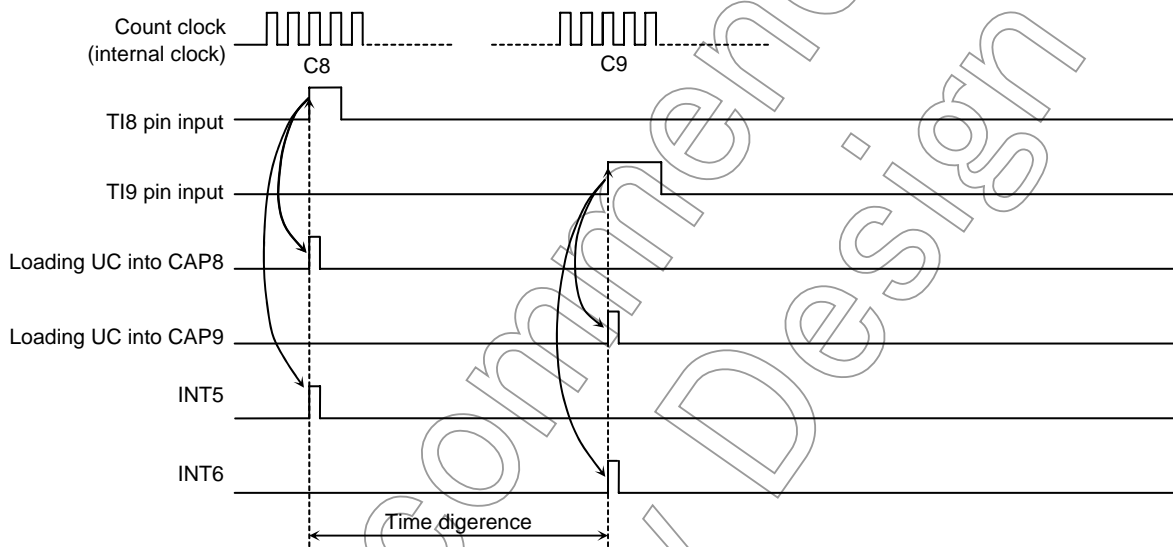
Figure 3.8.18 Time Difference Measurement

3.9 Serial Channels

The TMP92CD54I contains two serial input/output channels. Both channels support UART mode (asynchronous communication) and I/O interface mode (synchronous communication).

- I/O interface mode ——— Mode 0: Transmits and receives I/O data and its synchronization signal (SCLK) for expanding I/O.

- UART mode ——— Mode 1: Transmits/receives 7-bit data.
  Mode 2: Transmits/receives 8-bit data.
  Mode 3: Transmits/receives 9-bit data.

In mode 1 and mode 2, a parity bit can be added. In mode 3, a wakeup function is supported for the master controller to activate the slave controller in a serial link system.

Figure 3.9.2 and Figure 3.9.3 show block diagrams for each channel.

Serial channels 0 and 1 operate independently of each other. Both channels operate in the same way except the differences in specification listed in Table 3.9.1. This section only describes the operation of channel 0.

Table 3.9.1 Differences between Channels 0 to 1

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

- Mode 0 (I/O Interface Mode)

| bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

←——Transfer direction

- Mode 1 (7-Bit UART Mode)

No parity | start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | stop |

Parity | start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | parity | stop |

- Mode 2 (8-Bit UART Mode)

No parity | start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | stop |

Parity | start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | parity | stop |

- Mode 3 (9-Bit UART Mode)

| start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | stop |

| start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | bit 8 | stop (Wake-up) |
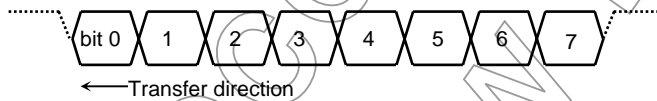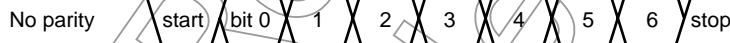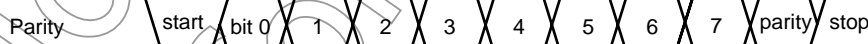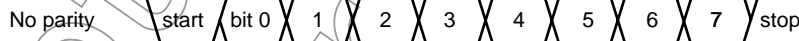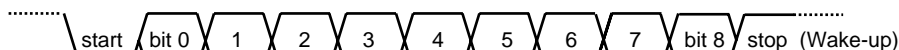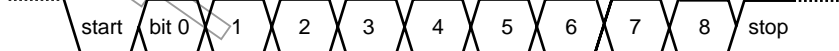
When Bit 8 = 1, an address (select code) is denoted.
When Bit 8 = 0, data is denoted.

Figure 3.9.1 Data Formats

3.9.1 Block diagram for each channel
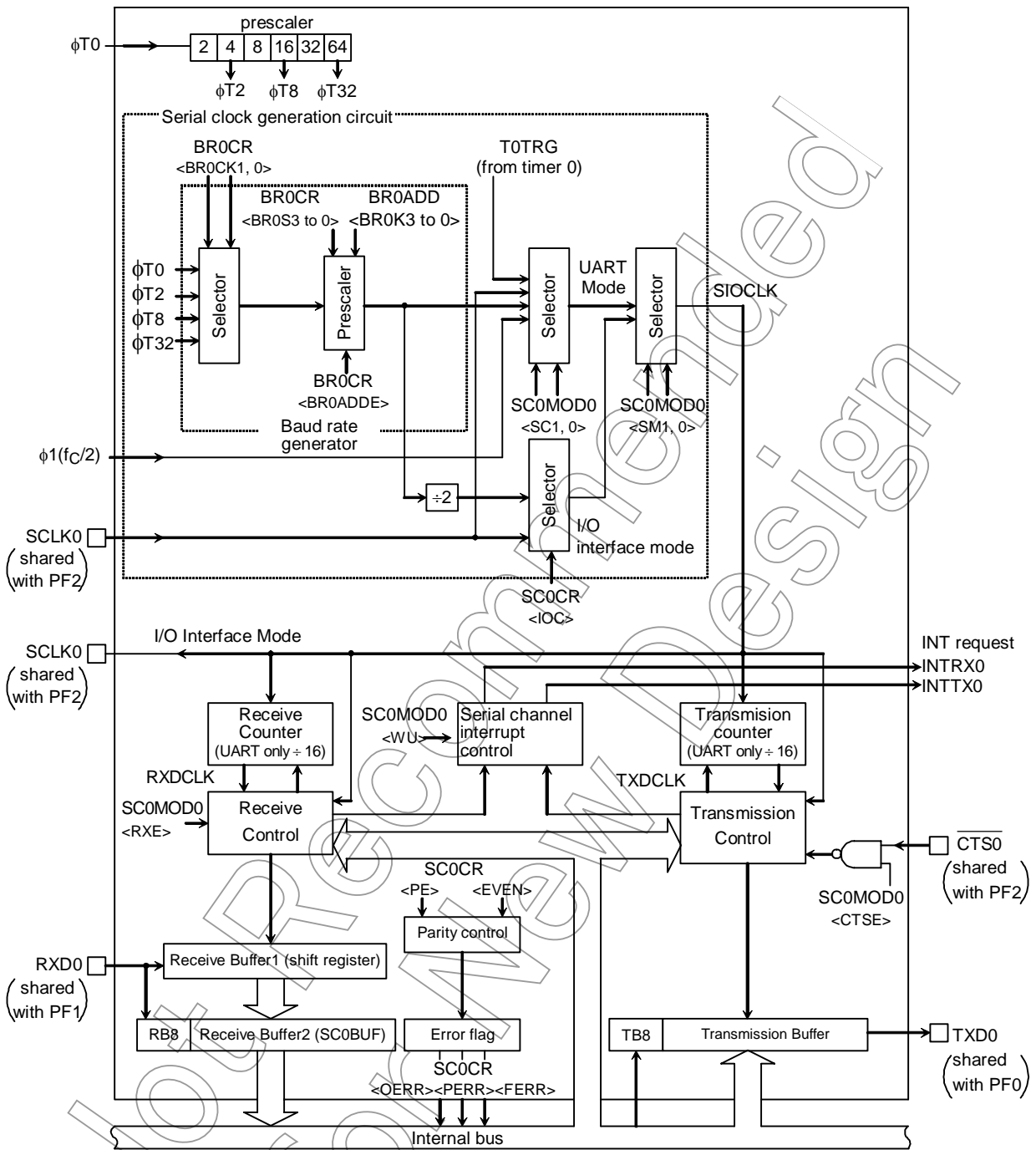
Figure 3.9.2 Block Diagram of the Serial Channel 0
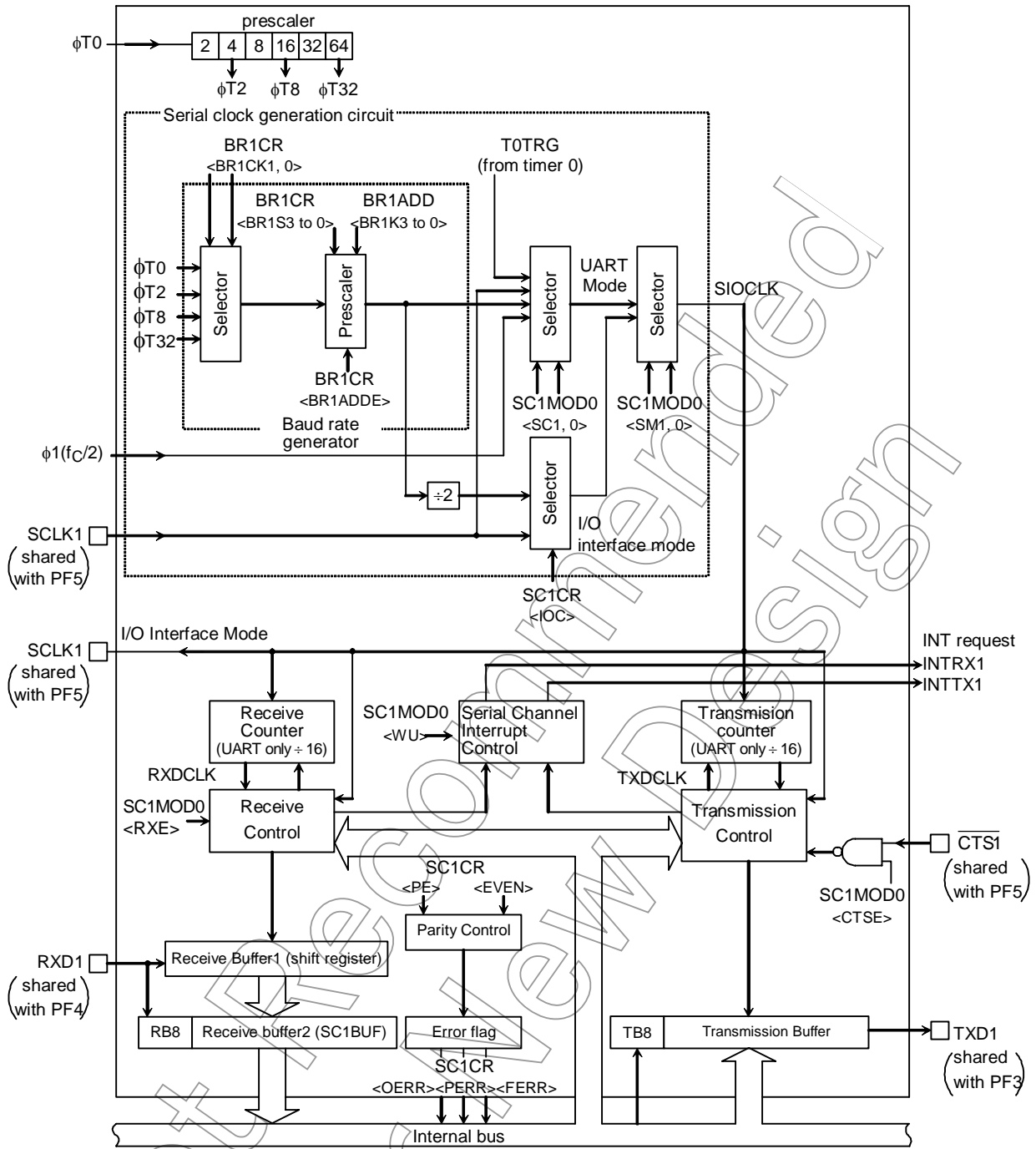
Figure 3.9.3 Block Diagram of the Serial Channel 1

3.9.2 Operation of each circuit

（1）Prescaler

The 6-bit prescaler divides the 1/4 system clock (fc/4) to generate the input clock for the baud rate generator. The BR0CR<BR0CK1:0> bits in the baud rate generator control register specify the input clock from the prescaler.

Table 3.9.2 shows the resolutions of prescaler output clocks.

Table 3.9.2 Prescaler Clock Resolution to Baud Rate Generator

At fc=20MHz

Output clock	Clock resolution
φT0 (4/fc)	0.2μs
φT2 (16/fc)	0.8μs
φT8 (64/fc)	3.2μs
φT32 (256/fc)	12.8μs

The baud rate generator uses one of four prescaler output clocks, φT0, φT2, φT8 and φT32.
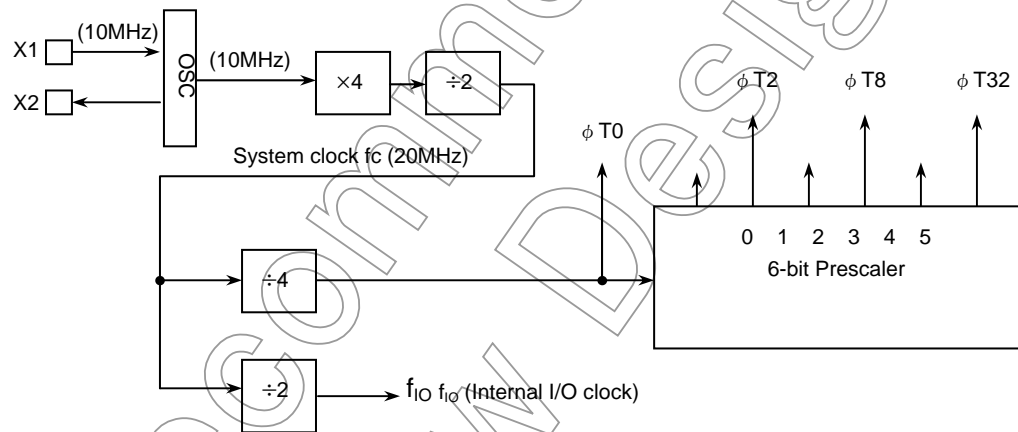
Figure 3.9.4 6-bit Prescaler

(2) Baud rate generator

The baud rate generator generates a transmit/receive clock that defines the transfer speed on a serial channel.

The clock input to the baud rate generator is generated with the 6-bit prescaler from ϕT0, ϕT2, ϕT8, or ϕT32. The BR0CR<BR0CK1:0> bits in the baud rate generator control register specify the input clock.

The baud rate generator contains a frequency divider that can divide the clock by N (N = 1 to 16) or N + (16 - K)/16 (N = 2 to 15 and K = 1 to 15). Note that specifying division by N causes the (16 - K)/16 portion to be disabled.

N \ K	Division by N setting −	Division by N + (16 - K)/16 setting 1	2	…	15
1	1	Do not use this setting			
2	2	2+1/16	2+2/16	…	2+15/16
3	3	3+1/16	3+2/16	…	3+15/16
4	4	4+1/16	4+2/16	…	4+15/16
5	:	:	:	…	:
14	14	14+1/16	14+2/16	…	14+15/16
15	15	15+1/16	15+2/16	…	15+15/16

so the overall division can take any value in the range [1; N+(16-K)/16; 16] with N = 2, 3, …, 15 and K = 1, 2, …, 15.

The transfer rate is determined from the settings of BR0CR<BR0ADDE, BR0S3:0> and BR0ADD<BR0K3:0>.

      BR0CR<BR0ADDE>: Division by N + (16 - K)/16
          0: Disabled
          1: Enabled

      BR0CR<BR0S3:0>: Set division ratio
          0000: N = 16 (cannot be used when division by N + (16 - K)/
                  16 is enabled)
          0001: N= 1
          0010: N= 2
            :
            :
          1111: N=15

      BR0ADD<BR0K3:0>: Set division ratio K (when division by N + (16 - K)/16 is enabled)
          0000: Disabled
          0001: K=1
            :
            :
          1111: K=15

- In UART mode

(1) When BR0CR<BR0ADDE> = 0

The setting of BR0ADD<BR0K3:0> is ignored and the frequency is divided by N as specified with BR0CK<BR0S3:0> (N = 1, 2, 3 ... 16).

(2) When BR0CR<BR0ADDE> = 1

Division by N + (16 - K)/16 is enabled. The frequency is divided by N + (16 - K)/16 according to N specified with BR0CR<BR0S3:0> (N = 2, 3 ... 15) and K

specified with BR0ADD<BR0K3:0> (K = 1, 2, 3 ... 15).

Note: If N = 1 or 16, division by N + (16 - K)/16 is disabled and BR0CR <BR0ADDE> must be set to 0.

- In I/O interface mode

    In I/O interface mode, division by N + (16 - K)/16 cannot be used. Always set BR0CR<BR0ADDE> to 0 to perform division by N.

    The following shows how to calculate the baud rate when using the baud rate generator:

    - UART mode

        $$\text{Baud Rate} = \frac{\text{Baud rate generator input clock frequency}}{\text{Baud rate generator frequency division value}} \div 16$$

    - I/O interface mode

        $$\text{Baud Rate} = \frac{\text{Baud rate generator input clock frequency}}{\text{Baud rate generator frequency division value}} \div 16$$

- Division by an integer (N)

    The baud rate in UART mode is calculated as follows when fc = 19.6608 MHz, the input clock is ϕT2 (frequency: fc/16), division value N (BR0CR<BR0S3:0>) = 8, and BR0CR<BR0ADDE> = 0:

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

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

Note: The setting of BR0ADD<BR0K3:0> is ignored for division by an integer because division by N + (16 - K)/16 is disabled.

- Division by N + (16 - K)/16 (in UART mode only)

    The baud rate is calculated as follows when fc = 15.9744 MHz, the input clock is ϕT2 (frequency: fc/16), division value N (BR0CR<BR0S3:0>) = 6, K (BR0ADD<BR0K3:0>) = 8, and BR0CR<BR0ADDE> = 1:

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

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

Table 3.9.3 and Table 3.9.4 show example baud rates in UART mode.

An external clock input can also be used as the serial clock. The following shows how to calculate the baud rate in that case:

- UART mode

    Baud rate = external clock input frequency ÷ 16

    The external clock input frequency must be less than or equal to fc/4.

- I/O interface mode

    Baud rate = external clock input frequency

    The external clock input frequency must be less than or equal to fc/16.

Table 3.9.3 Selection of Transfer Rate (1)

(When using the baud rate generator with 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: In I/O interface mode, the transfer rate is eight times the value shown in the table.

Table 3.9.4 Selection of Transfer Rate (2)

(When using timer 0 input clock ϕT1)

Unit: kbps

TREG0 \ fc	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	

How to calculate the baud rate (when using timer 0)

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

(When the timer 0 input clock is ϕT1)

Note: In I/O interface mode, a match signal from timer 0 cannot be used as a transfer clock.
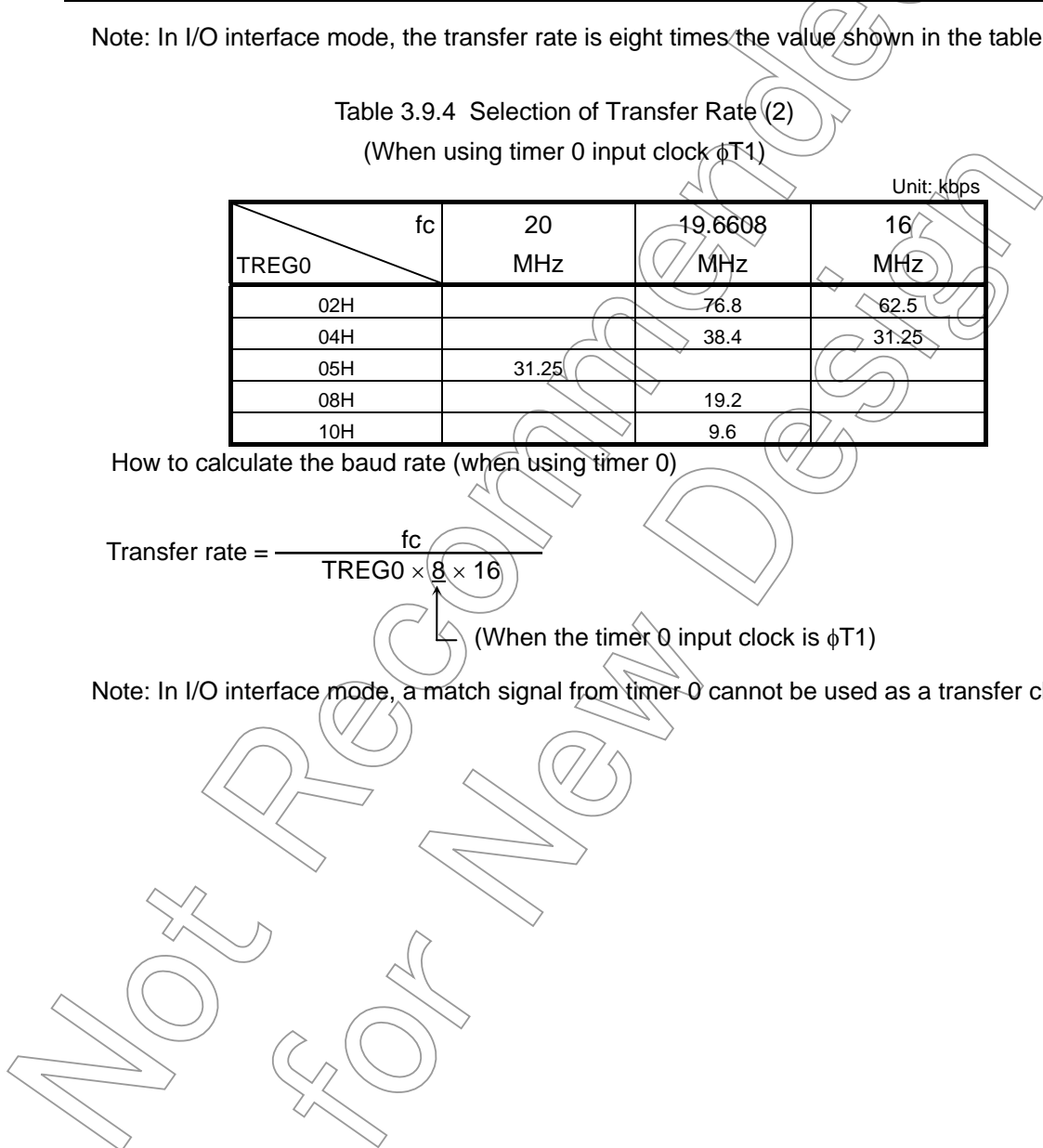
(3) Serial clock generator

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

- In UART mode

The SC0MOD0<SC1:0> register selects the clock to be used to generate the basic clock, SIOCLK, from the baud rate generator, internal clock $\phi 1$ (fc/2), a match detection signal from timer 0, or an external clock (SCLK0).

- In I/O interface mode

In SCLK output mode (SC0CR<IOC> = 0), the frequency of the baud rate generator output is divided by two to generate the basic clock.

In SCLK input mode (SC0CR<IOC> = 1), the basic clock is generated by detecting the rising or falling edge, as specified with the SC0CR<SCLKS> register.

(4) Receive counter

The receive counter is a 4-bit binary counter used in UART mode that increments with SIOCLK. A bit of data is received using 16 SIOCLK cycles and data is sampled in the seventh, eighth, and ninth clock cycles.

The received data is determined based on majority rule using three samples.

For example, if data is sampled as 1, 0, and 1 in the seventh, eighth, and ninth clock cycles, respectively, the received data is determined to be 1. If the sampled data is 0, 0, and 1, the received data is determined to be 0.

(5) Receive controller

- In I/O interface mode

In SCLK output mode (SC0CR<IOC> = 0), the RXD0 pin is sampled on the rising edge of the shift clock output to the SCLK0 pin.

In SCLK input mode (SC0CR<IOC> = 1), the RXD0 pin is sampled on the rising or falling edge of the SCLK0 pin input, depending on the SC0CR<SCLKS> setting.

- In UART mode

The receive controller has a start bit detector based on majority rule. If at least two of the three samples are 0, the controller determines that the start bit is valid and starts receiving data.

It also determines received data based on majority rule during data reception.

(6) Receive buffer

The receive buffer has double-buffer structure to prevent an overrun error. Received data is stored in receive buffer 1 (shift register) one bit at a time. Once seven or eight bits have been stored, the data is moved to the other buffer, receive buffer 2 (SC0BUF), at which time an INTRX0 interrupt occurs.

The CPU only reads data from receive buffer 2 (SC0BUF). Data received next can be stored in receive buffer 1 before the CPU reads the received data from receive buffer 2 (SC0BUF). An overrun error occurs, however, if the CPU does not read data from receive buffer 2 (SC0BUF) before all bits of next data are stored in receive buffer 1. If an overrun error occurs, the contents of receive buffer 2 and SC0CR<RB8> are maintained but those of receive buffer 1 are lost.

The parity bit when a parity is added to 8-bit UART data or the most significant bit in 9-bit UART mode is stored in SC0CR<RB8>.

In 9-bit UART mode, setting SC0MOD<WU> to 1 enables slave controller wakeup operation and an INTRX0 interrupt occurs only if SC0CR<RB8> = 1.

(7) Transmit counter

The transmit counter is a 4-bit binary counter used in UART mode. It is also counted with SIOCLK and generates a transmit clock, TXDCLK, once every 16 clock cycles.
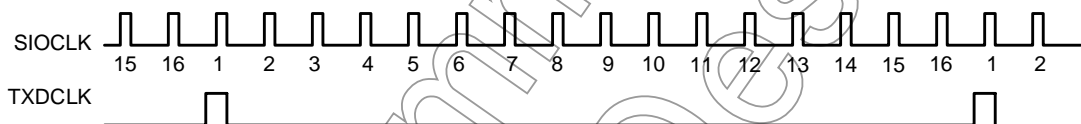
Figure 3.9.5 Generation of the transmission clock

(8) Transmit controller

- In I/O interface mode

In SCLK output mode (SC0CR<IOC> = 0), the data in the transmit buffer is output to the TXD0 pin, one bit at a time, on the rising edge of the shift clock output through the SCLK0 pin.

In SCLK input mode (SC0CR<IOC> = 1), the data in the transmit buffer is output to the TXD0 pin, one bit at a time, on the rising or falling edge of the SCLK input, depending on the SC0CR<SCLKS> setting.

- In UART mode

Once the CPU writes transmit data to the transmit buffer, the transmit controller starts transmission on the next rising edge of TXDCLK.

Handshaking

Serial channels 0 and 1 have the $\overline{\text{CTS}}$ pins, which enable transmission in frame units, thus preventing an overrun error. This function can be disabled or enabled using SC0MOD0<CTSE>.

If the $\overline{\text{CTS0}}$ pin is driven High during transmission, the transmitter completes the transmission of the data currently being transmitted and then stop transmission until the $\overline{\text{CTS0}}$ pin is driven back Low. An INTTX0 interrupt, however, occurs, with which the transmit controller requests next transmit data from the CPU, writes the data to the transmit buffer and then waits until transmission is ready.

The TMP92CD54I does not have a dedicated RTS pin. Any single port can be assigned to the RTS function. Once the receiver completes receiving data (in the RXD interrupt routine), it can drive the assigned RTS port High to request the transmitter to suspend transmission, thus easily implementing handshaking.
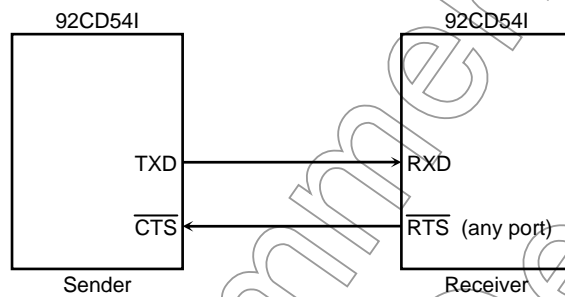
Figure 3.9.6 Handshake Function
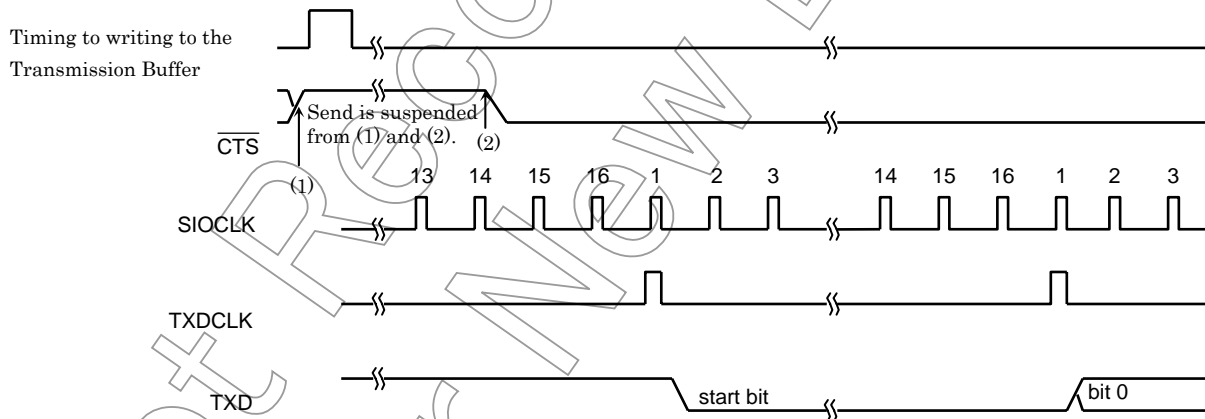
Note 1: If the $\overline{\text{CTS}}$ signal is driven High during transmission, next data transmission stops upon the completion of the current transmission.

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

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

(9) Transmit buffer

The CPU writes transmit data to the transmit buffer (SC0BUF). The transmit buffer shifts out the data, one bit at a time, in an LSB-first manner, with the transmit shift clock, TXDSFT, generated from the transmit controller. Once all bits have been shifted out, an INTTX0 interrupt occurs indicating that the transmit buffer is empty.

(10) Parity controller

Setting the SC0CR<PE> bit in the serial channel control register to 1 enables transmission with a parity. A parity can, however, be added only in 7-bit UART or 8-bit UART mode. The SC0CR<EVEN> register bit specifies whether an even or odd parity is used.

When transmitting data, the parity controller automatically generates a parity from the data written to the transmit buffer, SC0BUF. The parity is transmitted using SC0BUF<TB7> in 7-bit UART mode or SC0MOD0<TB8> in 8-bit UART mode. Ensure that the SC0CR<PE> and SC0CR<EVEN> bits are set before writing transmit data to the receive buffer.

When receiving data, the parity controller automatically generates a parity from the data that has been shifted into receive buffer 1 and then moved to receive buffer 2 (SC0BUF). The generated parity is compared with the parity contained in SC0BUF<RB7> in 7-bit UART mode or SC0CR<RB8> in 8-bit UART mode. If they differ, a parity error occurs and the SC0CR<PERR> flag is set.

(11) Error flags

Three error flags are provided to improve reliability in received data.

1. Overrun error <OERR>

An overrun error occurs if all bits of next data are received into receive buffer 1 with valid data still contained in receive buffer 2 (SC0BUF).

Recommended processing flow when an overrun error occurs:

(Receive interrupt routine)

1) Read the receive buffer.

2) Read the error flag.

3) if <OERR>=1

then

4) Write 0 to <RXE> to disable reception.

5) Wait until the current frame is completed.

6) Read the receive buffer.

7) Read the error flag.

8) Write 1 to <RXE> to enable reception.

9) Request retransmission.

10) Miscellaneous processing

2. Parity error <PERR>

A parity error occurs if the parity generated from the data moved to receive buffer 2 (SC0BUF) differs from the parity bit received through the RXD pin.

3. Framing error <FERR>

The stop bit in the received data is sampled three times near the middle of the reception period. A framing error occurs if it proves to be 0 based on majority rule.

(12) Start and stop timings for each signal

a. In UART mode

Reception

Table 3.9.5 Start and Stop Timings

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

Note: In 9-bit mode or 8-bit + parity mode, the ninth bit pulse and an interrupt occur simultaneously. To normally check for a framing error, therefore, it is necessary to wait for a single bit cycle to transmit a stop bit.

Transmission

Table 3.9.6 Stop Timings

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

b. I/O interface

Table 3.9.7 Interrupt Timings

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 data 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 description

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After Reset	undefined	0	0	0	0	0	0	0
Function	Transfer data bit 8	Hand shake 0: CTS disable 1: CTS enable	Receive function 0: Receive disable 1: Receive 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 rate generator 10: Internal clock φ1 11: External clcok (SCLK0 input)	

SC0MOD0 (00C2H)

Serial transmission clock source (UART)

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

Note: In I/O interface mode, select the clock using the serial control register (SC0CR).

Serial Transmission Mode

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 only when RB8 = 1	

Receiving Function

0	Receive disabled
1	Receive enabled

Handshake function (CTS pin) Enable

0	Disabled (always transferable)
1	Enabled

Transmission data bit 8

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

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After Reset	undefined	0	0	0	0	0	0	0
Function	Transfer data bit 8	Hand shake 0: CTS disable 1: CTS enable	Receive function 0: Receive disable 1: Receive 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 rate generator 10: Internal clock φ1 11: External clcok (SCLK1 input)	

SC1MOD0 (00CAH)

Serial transmission clock source (UART)

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

Note: In I/O interface mode, select the clock using the serial control register (SC1CR).

Serial Transmission Mode

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 only when RB8 = 1	

Receiving Function

0	Receive disabled
1	Receive enabled

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

0	Disabled (always transferable)
1	Enabled

Transmission data bit 8

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

		7	6	5	4	3	2	1	0
SC0CR (00C1H)	bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
	Read/Write	R	R/W		R(cleared to 0 when read)			R/W	
	After Reset	undefined	0	0	0	0	0	0	0
	Function	Received data 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 rate 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   } cleared to 0 when read
Overrun Error flag

Parity addition enable

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data 8

Note: Reading any of the error flags causes all of them to be cleared. Do not use a bit test instruction to test a single bit only.

Figure 3.9.10 Serial Control Register (channel 0, SC0CR)

	7	6	5	4	3	2	1	0
bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (cleared to 0 when read)			R/W	
After Reset	undefined	0	0	0	0	0	0	0
Function	Received data bit 8	Parity 0: odd 1: even	Parity addition 0: disable 1: enable	1: error			0: SCLK1 (rising) 1: SCLK1 (falling)	0: baud rate generator 1: SCLK1 pin input
				Overrun	Parity	Framing		

SC1CR (00C9H)

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   } cleared to 0 when read
Overrun Error flag

Parity addition enable

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data bit 8

Note: Reading any of the error flags causes all of them to be cleared. Do not use a bit test instruction to test a single bit only.

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		Setting of the divided frequency			

+(16 - K) / 16 division enable

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

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

Baud rate generator frequency divisor setting

BR0ADD <BR0K3:0> \ BR0CR <BR0S3:0>	BR0CR<BR0ADDE> = 1 (UART only)		BR0CR<BR0ADDE> = 0 (UART and I/O interface modes)
	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 *	Divided by N + (16 − K) / 16	Divided by N

*: When N = 1 or 16, division by N + (16 - K)/16 in UART mode cannot be used. Division by N + (16 - K)/16 with <BR0K3:0>=0000 is also not supported. If any of those settings are used, set BR0CR<BR0ADDE> to 0 to disable division by N + (16 - K)/16.

Note 1: When using division by N + (16 - K)/16, first set the value of K (K = 1 to 15) in BR0ADD<BR0K3:0> before setting BR0CR<BR0ADDE> to 1.

Note 2: Division by N + (16 - K)/16 can only be used in UART mode. In I/O interface mode, set BR0CR<BR0ADDE> to 0 to disable division by N + (16 - K)/16.

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		Setting of the divided frequency			

+(16 - K) / 16 division enable

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 ⟨BR1S3:0⟩ / BR1ADD ⟨BR1K3:0⟩	BR1CR⟨BR1ADDE⟩ = 1 (UART only)		BR1CR⟨BR1ADDE⟩ = 0 (UART and I/O interface modes)
	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 *	Divided by N + (16 − K) / 16	Divided by N

*: When N = 1 or 16, division by N + (16 - K)/16 in UART mode cannot be used. Division by N + (16 - K)/16 with ⟨BR0K3:0⟩=0000 is also not supported. If any of those settings are used, set BR0CR⟨BR0ADDE⟩ to 0 to disable division by N + (16 - K)/16.

Note 1: When using division by N + (16 - K)/16, first set the value of K (K = 1 to 15) in BR1ADD⟨BR1K3:0⟩ before setting BR1CR⟨BR1ADDE⟩ to 1.

Note 2: Division by N + (16 - K)/16 can only be used in UART mode. In I/O interface mode, set BR1CR⟨BR1ADDE⟩ to 0 to disable division by N + (16 - K)/16.

Figure 3.9.13 Baud Rate Generator Control (channel 1, BR1CR, BR1ADD)

		7	6	5	4	3	2	1	0	
		TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)

SC0BUF
(00C0H)

	7	6	5	4	3	2	1	0	
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Reveiving)
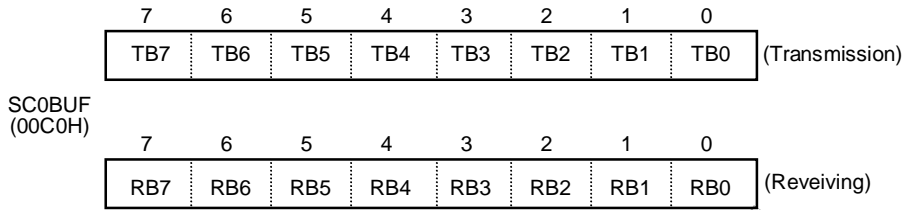
Note: SC0BUF does not support a read-modify-write operation.

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/Write	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)

	7	6	5	4	3	2	1	0	
	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)

SC1BUF
(00C8H)

	7	6	5	4	3	2	1	0	
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receiving)
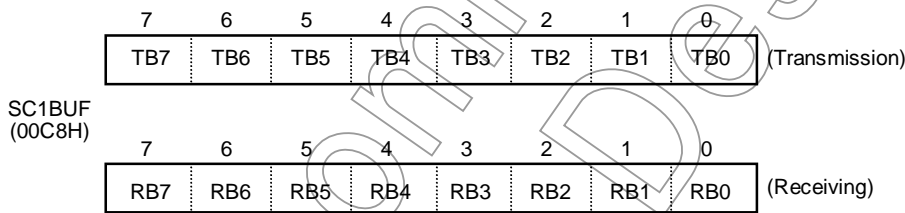
Note: SC1BUF does not support a read-modify-write operation.

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/Write	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 in each mode

（1）Mode 0 (I/O interface mode)

This mode is used to increase the number of input/output pins (I/O). In this mode, a serial channel transmits and receives data to and from a shift register or other devices connected externally.

The I/O interface mode can be selected between SCLK output mode, in which the TMP92CD54I outputs a synchronization clock (SCLK) and SCLK input mode, in which SCLK is supplied from an external device.
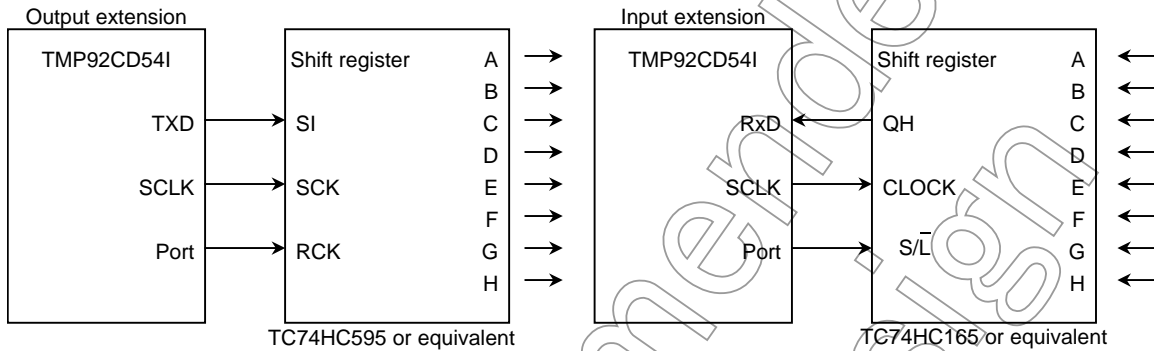
Figure 3.9.18 Example of SCLK Output Mode Connection
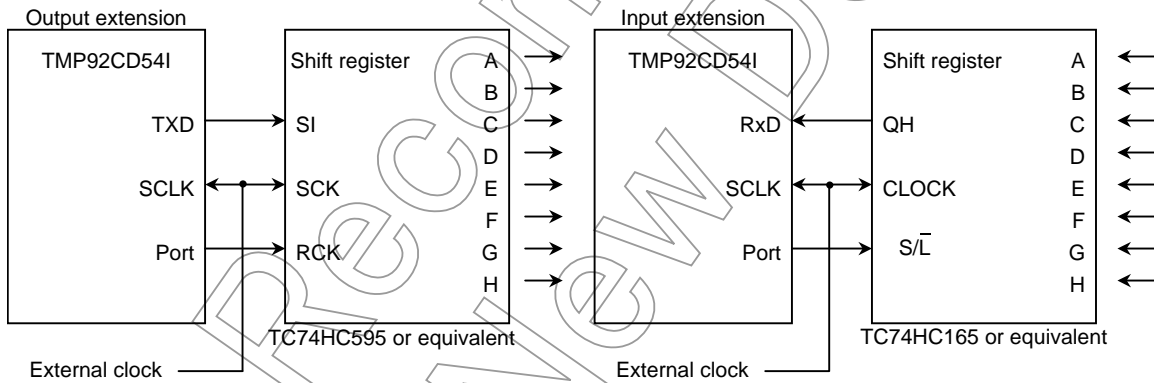
Figure 3.9.19 Example of SCLK Input Mode Connection

a.   Transmission

In SCLK output mode, every time the CPU writes data to the transmit buffer, 8-bit data is output through the TXD0 pin and the synchronization clock through the SCLK0 pin.

Once all data has been output, INTES0<ITX0C> is set to 1 and an INTTX0 interrupt occurs.

Timing to write
transmisison data

SCLK0 output

TXD0   bit 0   bit 1   bit 6   bit 7

ITX0C (INTTX0
interrupt request)
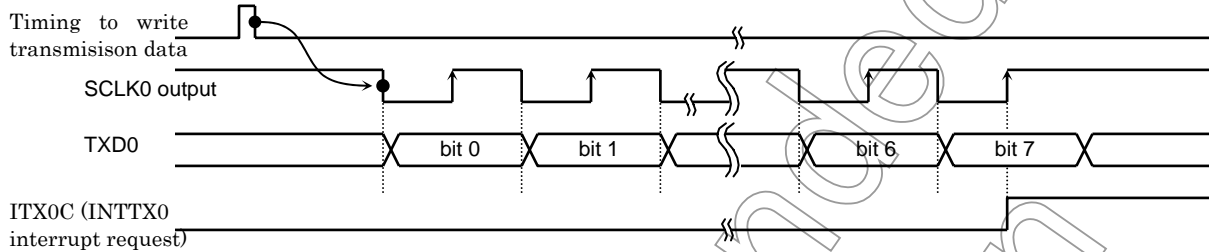
Figure 3.9.20 Transmitting Operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK input mode, when the transmit buffer contains data written by the CPU, activating the SCLK0 input causes 8-bit data to be output through the TXD0 pin.

Once all data has been output, INTES0<ITX0C> is set to 1 and an INTTX0 interrupt occurs.

SCLK0 input
(<SCLKS>=0: Rising edge mode)

SCLK0 input
(<SCLKS>=1: Falling edge mode)

TXD0   bit 0   bit 1   bit 5   bit 6   bit 7

ITX0C (INTTX0
interrupt request)
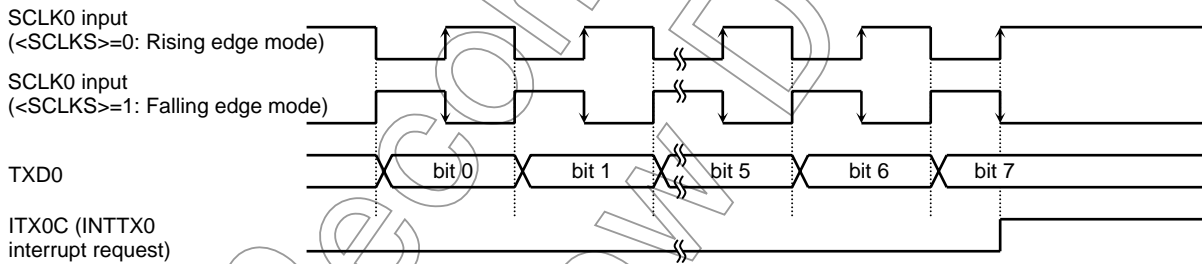
Figure 3.9.21 Transmitting Operation in I/O Interface Mode (SCLK0 Input Mode)

b.   Reception

In SCLK output mode, every time the CPU reads received data and the receive interrupt flag, INTES0<IRX0C>, is cleared, the synchronization clock is output through the SCLK0 pin and next data is shifted into receive buffer 1. Once 8-bit data has been received, the data is moved to receive buffer 2 (SC0BUF), causing INTES0<IRX0C> to be re-set to 1 and an INTRX0 interrupt to occur.

The initial start of SCLK output is triggered by setting SC0MOD0<RXE> to 1.
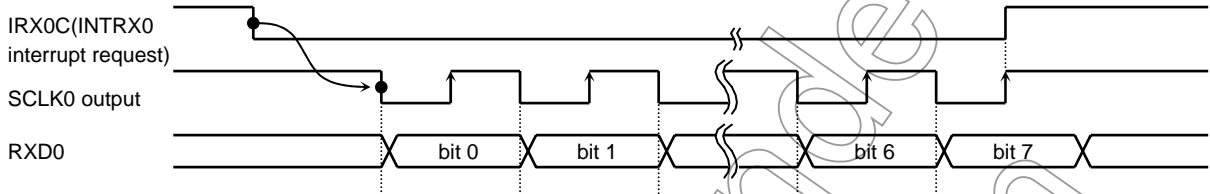
Figure 3.9.22 Receiving Operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK input mode, when the CPU has read received data and the receive interrupt flag, INTES0<IRX0C>, has been cleared, activating the SCLK0 input causes next data to be shifted into receive buffer 1. Once 8-bit data has been received, the data is moved to receive buffer 2 (SC0BUF), causing INTES0<IRX0C> to be re-set to 1 and an INTRX0 interrupt to occur.
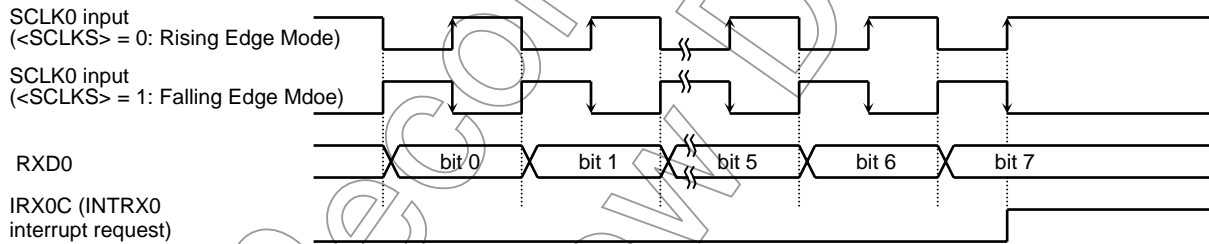
Figure 3.9.23 Receiving Operation in I/O interface Mode (SCLK0 Input Mode)

Note: To receive data, in either SCLK input or output mode, ensure that SC0MOD0 <RXE> is set to 1 to enable reception.

c. Transmission/reception (full duplex)

To transmit and receive data in full duplex mode, set the receive interrupt level to 0 and the transmit interrupt level to any of 1 to 6.

Perform receive processing in the transmit interrupt handling routine, as shown below, before setting next data to be transmitted:

Example: Channel 0, SCLK output
         Transmit and receive data at 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 transmit interrupt level to 1 and
                                      set receive interrupt level to 0 (disable).
PFCR      -  -  -  -  -  -  1  0  1   Set PF0, PF1 and PF2 to the TXD0, RXD0 and
PFFC      -  -  -  -  -  -  1  -  1   SCLK0 pins, respectively.
SC0MOD0   0  0  0  0  0  0  0  0      Enable reception and set I/O interface mode.
SC0MOD1   1  1  0  0  0  0  0  0      Specify full duplex mode.
SC0CR     0  0  0  0  0  0  0  0      Sclk_out, transmit on falling edge and receive on
                                      rising edge.
BR0CR     0  0  1  0  0  0  0  0      Select 9600 bps.
SC0MOD0   0  0  1  0  0  0  0  0      Enable reception.
SC0BUF    *  *  *  *  *  *  *  *      Set transmit data and activate.
```

INTTX0 interrupt routine

```
Acc      ←   SC0BUF                   Read the received data.
SC0BUF   *  *  *  *  *  *  *  *        Set transmit data.
```

X = Don't care   "-" = No change

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

Setting SC0MOD0<SM1:0> to 01 in the serial channel mode register selects 7-bit UART mode.

In this mode, a parity bit can be added and SC0CR<PE> in the serial channel control register enables or disables the addition of a parity bit. When <PE>= 1 (enabled), either an even or odd parity can be selected using SC0CR<EVEN>.

Example: The table below shows control register settings for transmitting data in the following format:

Direction of transfer (transfer rate: 2400 bps when fc = 19.6608 MHz)
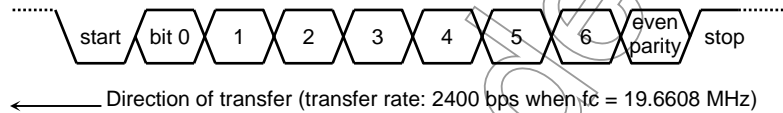
```
              7 6 5 4 3 2 1 0
PFCR     ← – – – – – – – 1          ⎫ Set PF0 to function as the TXD0 pin.
PFFC     ← – – – – – – – 1          ⎬
SC0MOD0  ← X 0 – X 0 1 0 1          Select 7-bit UART mode.
SC0CR    ← X 1 1 X X X 0 0          Select even parity mode.
BR0CR    ← 0 0 1 0 1 0 0 0          Set transfer rate to 2400 bps.
INTES0   ← X 1 0 0 – – – –          Enable INTTX0 interrupt and set the level to 4.
SC0BUF   ← * * * * * * * *          Set transmit data.
```

X = Don't care; "–" = No change

Figure 3.9.24 Transmit Data Example (mode 1)

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

Setting SC0MOD0<SM1:0> to 10 selects 8-bit UART mode.

In this mode, a parity bit can be added and SC0CR<PE> enables or disables the addition of a parity bit. When <PE>= 1 (enabled), either an even or odd parity can be selected using SC0CR<EVEN>.

Example: The table below shows control register settings for receiving data in the following format:

Direction of transfer (transfer rate: 9600 bps when fc = 19.6608 MHz)
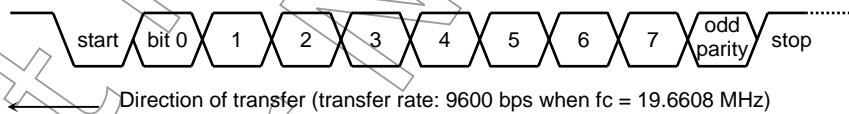
Settings in main routine

```
                7 6 5 4 3 2 1 0
PFCR      ← - - - - - - 0 -       Set PF1 to the RXD0 pin.
SC0MOD0   ← - 0 1 X 1 0 0 1       Enable reception and select 8-bit UART mode.
SC0CR     ← X 0 1 X X X 0 0       Select odd parity mode.
BR0CR     ← 0 0 0 1 1 0 0 0       Set transfer rate to 9600 bps.
INTES0    ← - - - - X 1 0 0       Enable INTRX0 interrupt and set the level to 4.
```

Settings in interrupt routine

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

X = Don't care   "-" = No change

Figure 3.9.25 Transmit Data Example (mode 2)

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

Setting SC0MOD0<SM1,SM0> to 11 selects 9-bit UART mode. This mode does not support a parity bit.

The most significant bit (bit 9) is written to SC0MOD0<TB8> in the serial channel mode register for transmission or stored into SC0CR<RB8> in the serial channel control register for reception. When data is written to or read from the buffer, the most significant bit must always be transferred first, followed by the bits in SC0BUF.

Wakeup function

In 9-bit UART mode, setting SC0MOD<WU> to 1 enables slave controller wakeup operation and an INTRX0 interrupt occurs only if <RB8> = 1.
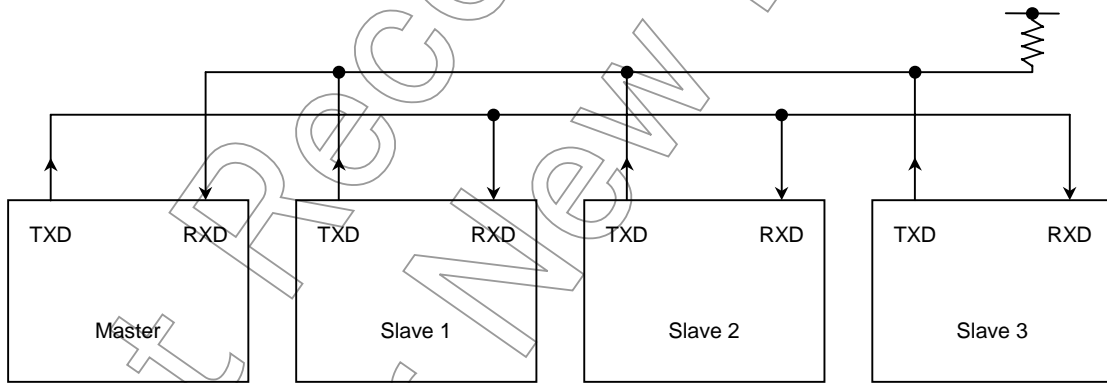
Note: The TXD pin on the slave controller must be set to open-drain.

Figure 3.9.26 Serial Link Using Wakeup Function

Protocol

a. Set the master and slave controllers to 9-bit UART mode.

b. In each slave controller, set SC0MOD0<WU> to 1 to enable reception.

c. The master controller transmits a single frame including the slave controller selection code (8 bits). The most significant bit (bit 8) of the frame, <TB8>, must be set to 1.
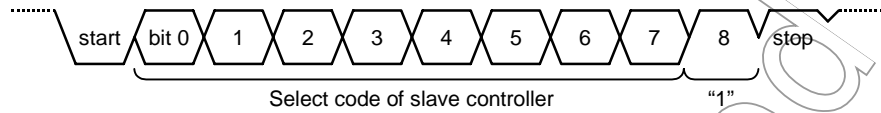
Figure 3.9.27 Frame (1)

d. Each slave controller receives the above frame. The slave controller whose code matches the received selection code clears the WU bit to 0.

e. The master controller transmits data to the selected slave controller (with SC0MOD0<WU> cleared to 0). The most significant bit (bit 8) of the data, <TB8>, must be set to 0.

Figure 3.9.28 Frame (2)

f. The other slave controllers, with <WU> set to 1, ignore the received data because the most significant bit (bit 8) of <RB8> is 0 so that an INTRX0 interrupt does not occur. The slave controller with <WU> cleared to 0 can also transmit data to the master controller to notify that it has completed receiving the data.

Example: Serially linking with two slave controllers using internal clock $\phi1$ as the transfer clock
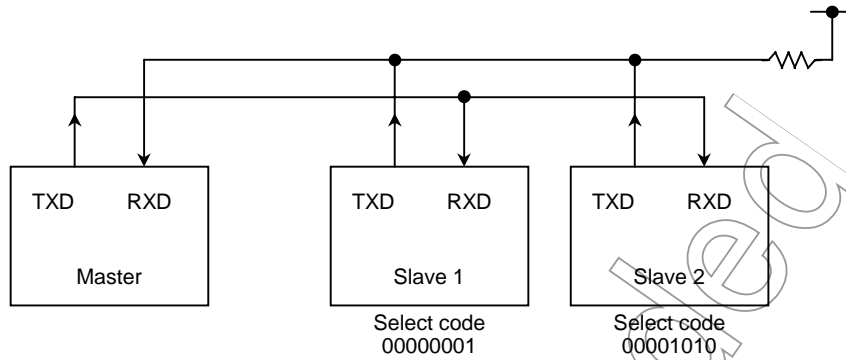
Figure 3.9.29 Transfer Clock Example

- Settings in master controller

    Main routine

    ```
    PFCR      ← – – – – – – 0 1 ⎫   Set PF0 and PF1 to the TXD0 and RXD0 pins, respectively.
    PFFC      ← – – – – – – – 1 ⎭
    INTES0    ← X 1 0 0 X 1 0 1     Enable INTTX0 interrupt and set the level to 4.
                                    Enable INTRX0 interrupt and set the level to 5.
    SC0MOD0   ← 1 0 1 0 1 1 1 0     Select 9-bit UART mode and set transfer rate to φ1.
    SC0BUF    ← 0 0 0 0 0 0 0 1     Set the selection code for slave 1.
    ```

    INTTX0 interrupt routine

    ```
    SC0MOD0   ← 0 – – – – – – –     Set TB8 to 0.
    SC0BUF    ← * * * * * * * *     Set transmit data.
    ```

- Settings in slave controller

    Main routine

    ```
    PFCR      ← – – – – – – 0 0 ⎫   Set PF1 and PF0 to RXD0 and TXD0 (open-drain output),
    PFFC      ← – – – – – – – 1 ⎭   respectively.
    INTES0    ← X 1 0 1 X 1 1 0     Enable INTRX0 and INTTX0.
    SC0MOD0   ← 0 0 1 1 1 1 1 0     Select 9-bit UART mode, set transfer rate to φ1 and WU to 1.
    ```

    INTRX0 interrupt routine

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

3.10 Serial Bus Interface (SBI)

The TMP92CD54I contains three serial bus interface (SBI) channels, SBI0, SBI1, and SBI2.

The serial bus interface supports the following two operating modes:

- I^2C bus mode (multi-master)
- Clock synchronous 8-bit SIO mode

Table 3.10.1 Used Pins

	I^2C bus		Clocked-synchronous 8-bit SIO
SBI0	SCL0 (PN2), SDA0 (PN1)	PNODE<ODEN2, ODEN1>	SCK0 (PN0), SO0 (PN1), SI0 (PN2)
SBI1	SCL1 (PN5), SDA1 (PN4)	PNODE<ODEN5, ODEN4>	SCK1 (PN3), SO1 (PN4), SI1 (PN5)
SBI2	SCL2 (P72), SDA2 (PN6)	PNODE<ODE72, ODEN6>	SCK2 (PM4), SO2 (PN6), SI2 (P72)

Each channel operates in the same way. This section describes only SBI0.

In I^2C bus mode, the TMP92CD54I is connected to an external device through PN1 (SDA0) and PN2 (SCL0). In clock synchronous 8-bit SIO mode, the TMP92CD54I is connected to an external device through PN0 (SCK0), PN1 (SO0), and PN2 (SI0).

The following table shows the pin settings for each mode:

Table 3.10.2 Pin Settings

	PNODE <ODEN2, ODEN1>	PNCR <PN2C, PN1C, PN0C>	PNFC <PN2F, PN1F, PN0F>
I^2C Bus Mode	11	11X	11X
Clocked Synchronous 8-Bit SIO Mode	XX	011 010	011

X: Don't care

3.10.1 Configuration
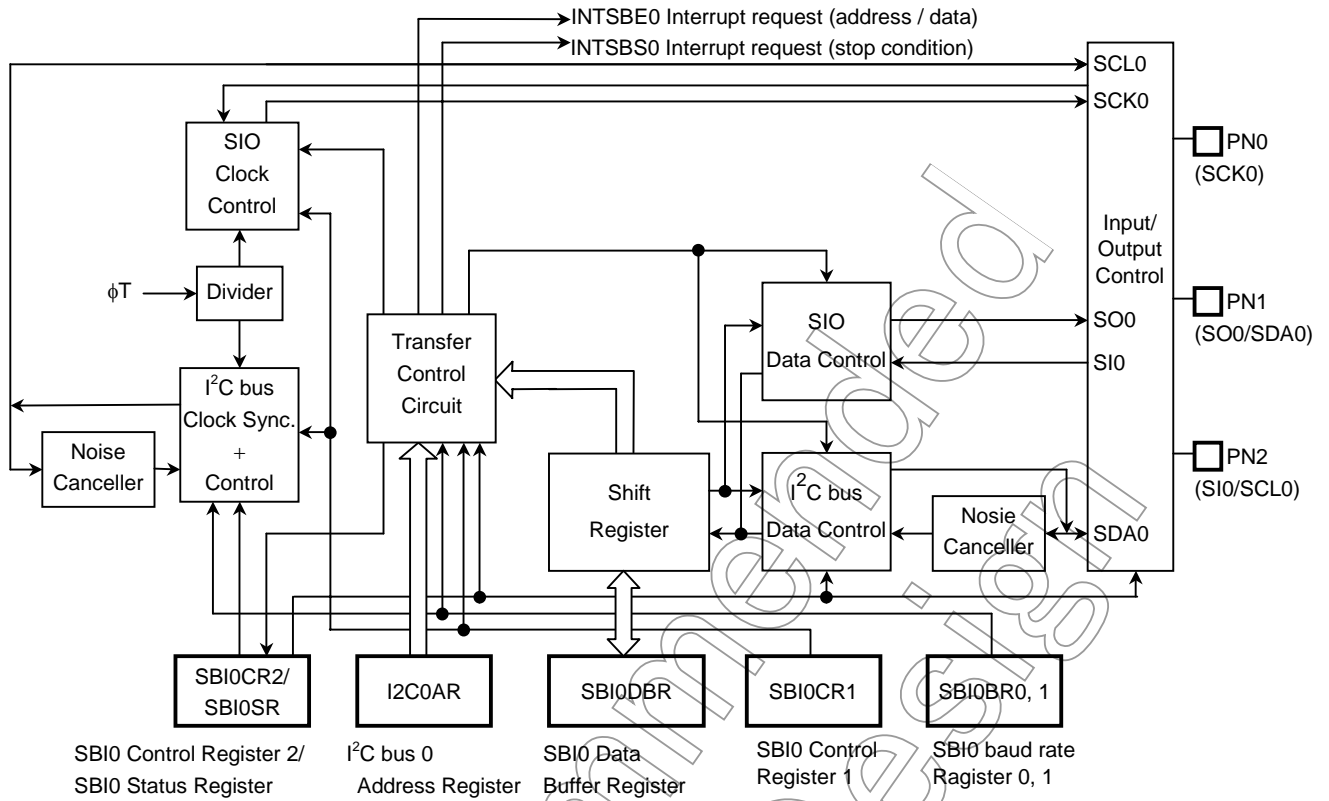
Figure 3.10.1 Serial Bus Interface 0 (SBI0)

3.10.2 Control

The following registers are used to control the serial bus interface and monitor its operating state:

- Serial bus interface 0 control register 1 (SBI0CR1)
- Serial bus interface 0 control register 2 (SBI0CR2)
- Serial bus interface 0 data buffer register (SBI0DBR)
- I²C 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 have different functions depending on the mode in which they are used. For details, see "3.10.4 Control Registers in I²C Bus Mode" and "3.10.7 Control in Clock Synchronous 8-bit SIO Mode."

3.10.3 Data formats in I²C bus mode

Figure 3.10.2 shows the data formats used in I²C bus mode.

(a) Addressing format
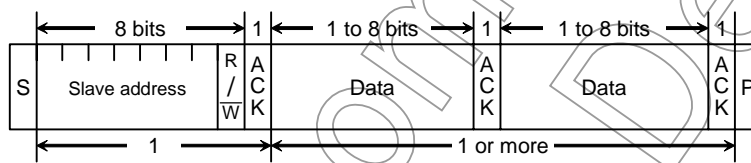
(b) Addressing format (with restart)
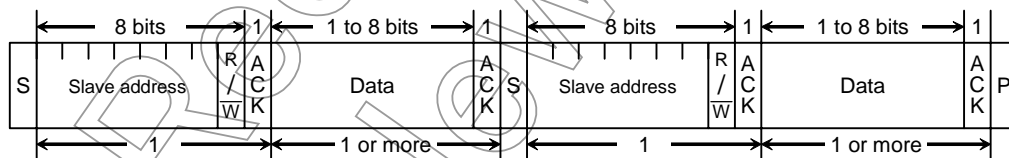
(c) Free data format (Transfer format for transferring data from the master device to a slave device)
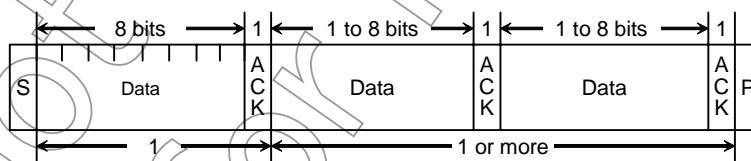
Note:
S: Start condition
R/W̄: Direction bit
ACK: Acknowledge bit
P: Stop condition

Figure 3.10.2 Data Format in the I²C Bus Mode

3.10.4 Control Registers in I²C Bus Mode

The following registers are used to control the serial bus interface (SBI) and monitor its operating state in I²C bus mode:

Serial Bus Interface 0 Control Register 1

		7	6	5	4	3	2	1	0
SBI0CR1 (0170H)	Bit Symbol	BC2	BC1	BC0	ACK	SCK3	SCK2	SCK1	SWRMON/ SCK0
	Read/Write	W			R/W	W			R/W
	After Reset	0	0	0	0	1	0	0	1/0(Note3)
Read-modify-write not allowed	Function	Number of transferred bits (Note 1)			Acknowledge mode specification 0: Not generate 1: Generate	Internal serial clock selection and software reset monitor (Note 2)			

Internal serial clock selection <SCK3 to 0> @ write

0001	n = 6	–	
0010	n = 7	–	CPU clock: fc = 20 MHz internal SCL output
0011	n = 8	75.8 kHz	
0100	n = 9	38.5 kHz	
0101	n = 10	19.4 kHz	
0110	n = 11	9.73 kHz	
1000	Fast	400 kHz	(fscl = fc/50 [Hz])
1111	Standard	100 kHz	(fscl = fc/200 [Hz])
other	-	(reserved)	

$$fscl = \frac{fc}{2^n + 8} \ [\,Hz\,]$$

Software reset state monitor <SWRMON> @ read

0	During software reset
1	Initial data

Acknowledge mode specification

0	Not generate clock pulse for acknowledge signal
1	Generate clock pulse 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: It is necessary to clear <BC2:0> to 000 before attempting to change the operating mode to clock synchronous 8-bit SIO mode.
Note 2: For details of the SCL line clock frequency, see "3.10.5 (3) Serial clock."
Note 3: The initial values of SCK0 and SWRMON are 0 and 1, respectively.

Figure 3.10.3 Registers for the I²C Bus Mode

Serial Bus Interface 0 Control Register 2

		7	6	5	4	3	2	1	0
SBI0CR2 (0173H)	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
Read-modify-write not allowed	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: I^{2}C Bus Mode 11: (reserved)		Software reset generate write "10" and "01", then an internal reset signal is generated.	

Serial bus interface operating mode selection (Note 2)

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

INTSBE0 interrupt request

0	–
1	Cancel interrupt request

Start / stop generation

0	Generates the stop condition
1	Generates the start condition

Transmitter / receiver selection

0	Receiver
1	Transmitter

Master / Slave selection

0	Slave
1	Master
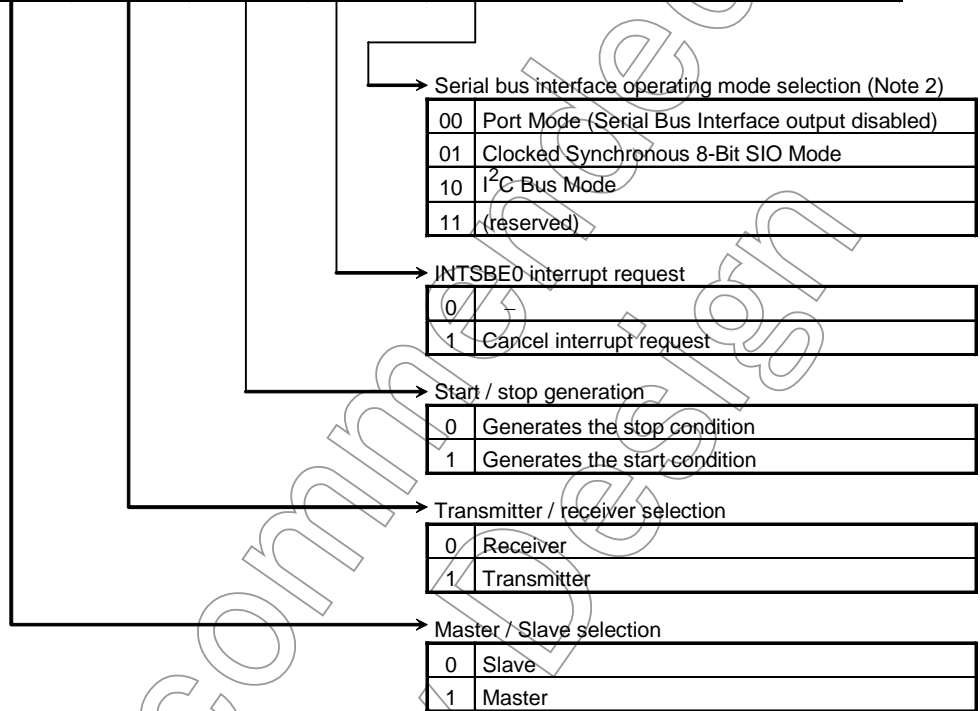
Note1: When read, this register functions as SBI0SR.
Note2: Ensure that the bus is free before attempting to select port mode.
    Also ensure that the port state is High before attempting to change the mode from port
    mode to I^{2}C bus or clock synchronous 8-bit SIO mode.

Figure 3.10.4 Registers for the I^{2}C Bus Mode

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
Read-modify-write not allowed	Function	Master/ Slave status monitor	Transmitter/ Receiver status monitor	I²C bus status monitor	INTSBE0 interrupt request monitor	Arbitration lost detection monitor 0: − 1: Detected	Slave address match detection monitor 0: Undetected 1: Detected	GENERAL CALL detection monitor 0: Undetected 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	Undetected
1	GENERAL CALL detected

Slave address match detection monitor

0	Undetected
1	Slave address match or GENERAL CALL detected

Arbitration lost detection monitor

0	−
1	Arbitration lost

INTSBE0 interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I²C bus status monitor

0	Free (Note2)
1	Busy

Transmitter / receiver status monitor

0	Receiver
1	Transmitter

Master / Slave status monitor

0	Slave
1	Master

Note1: When written, this register functions as SBI0CR2.
Note2: When the BB flag changes its state from 1 to 0 (falling edge), INTSBS0 occurs.

Figure 3.10.5 Registers for the I²C Bus Mode

Serial Bus Interface 0 Baud Rate Register 0

SBI0BR0 (0174H) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	Bit Symbol	-	I2SBI0	-	-	-	-	-	-
	Read/Write	W	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/Write	R/W							
	After Reset	0	-	-	-	-	-	-	-
	Function	Internal clock 0: Stop 1: Operate							

→ Baud rate clock control

0	Stop
1	Operate

Serial Bus Interface 0 Data Buffer Register

SBI0DBR (0171H) Read-modify-write not allowed		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 transmit data, justify the data toward the MSB (bit 7) side.

I^{2}C Bus 0 Address Register

I2C0AR (0172H) Read-modify-write not allowed		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 device							Addressing or free data format

→ Address recognition mode specification

0	Addressing format
1	Free data format

The addressing or free data format affects both the slave and master.

When using the addressing format (<ALS>=0), the TRX bit is updated with the direction bit, R/W (bit 8 of the first byte received after the start condition). In addition, in slave mode, the MCU finds the bus when it recognizes the address which follows the start condition.

When using the free data format (<ALS>=1), TRX remains unchanged because all words on the bus are not recognized as an address but as data words.

Figure 3.10.6 Registers for the I^{2}C Bus Mode

Seirial Bus Interface 1 Conrol Register 1

		7	6	5	4	3	2	1	0
SBI1CR1 (0178H)	Bit symbol	BC2	BC1	BC0	ACK	SCK3	SCK2	SCK1	SWRMON/ SCK0
	Read/Write	W			R/W	W			R/W
	After Reset	0	0	0	0	1	0	0	1/0(Note3)
Read-modify-write not allowed	Function	Number of transferred bits (Note 1)			Acknowledge mode specification 0: Not generate 1: Generate	Internal serial clock selection and software reset monitor (Note 2)			

Internal serial clock selection <SCK3 to 0> @ write

0001	n = 6	–	
0010	n = 7	–	CPU clock: fc = 20 MHz
0011	n = 8	75.8 kHz	internal SCL output
0100	n = 9	38.5 kHz	$(f_{scl} = \dfrac{fc}{2^n + 8}$ [Hz] $)$
0101	n = 10	19.4 kHz	
0110	n = 11	9.73 kHz	
1000	Fast	400 kHz	$(f_{scl} = fc/50$ [Hz] $)$
1111	Standard	100 kHz	$(f_{scl} = fc/200$ [Hz] $)$
other	-	(reserved)	

Software reset state monitor <SWRMON> @ read

0	During software reset
1	Initial data

Acknowledge mode specification

0	Not generate clock pulse for acknowledge signal
1	Generate clock pulse 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: It is necessary to clear <BC2:0> to 000 before attempting to change the operating mode to clock synchronous 8-bit SIO mode.

Note 2: For details of the SCL line clock frequency, see "3.10.5 (3) Serial clock."

Note 3: The initial values of SCK1 and SWRMON are 0 and 1, respectively.

Figure 3.10.7 Registers for the I^2C Bus Mode

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
Read-modify-write not allowed	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: I^{2}C Bus Mode 11: (reserved)		Software reset generate write "10" and "01", then an internal reset signal is generated.	

Serial bus interface operating mode selection (Note 2)

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

INTSBE1 interrupt request

0	−
1	Cancel interrupt request

Start / stop generation

0	Generates the stop condition
1	Generates the start condition

Transmitter / receiver selection

0	Receiver
1	Transmitter

Master / Slave selection

0	Slave
1	Master

Note1: When read, this register functions as SBI1SR.
Note2: Ensure that the bus is free before attempting to select port mode.
　　　　Also ensure that the port state is High before attempting to change the mode from port mode to I^{2}C bus or clock synchronous 8-bit SIO mode.

Figure 3.10.8 Registers for the I^{2}C Bus Mode

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
Read-modify-write not allowed	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: Undetected 1: Detected	GENERAL CALL detection monitor 0: Undetected 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	Undetected
1	GENERAL CALL detected

Slave address match detection monitor

0	Undetected
1	Slave address match or GENERAL CALL detected

Arbitration lost detection monitor

0	−
1	Arbitration lost

INTSBE1 interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I^2C bus status monitor

0	Free (Note2)
1	Busy

Transmitter / receiver status monitor

0	Receiver
1	Transmitter

Master / Slave status monitor

0	Slave
1	Master

Note1: When written, this register functions as SBI1CR2.
Note2: When the BB flag changes its state from 1 to 0 (falling edge), INTSBS1 occurs.

Figure 3.10.9 Registers for the I^2C Bus Mode

Serial Bus Interface 1 Baud Rate Register 0

		7	6	5	4	3	2	1	0
SBI1BR0 (017CH)	Bit Symbol	-	I2SBI0	-	-	-	-	-	-
	Read/Write	W	R/W						
	After Reset	0	0	-	-	-	-	-	-
Read-modify-write not allowed	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

		7	6	5	4	3	2	1	0
SBI1BR1 (017DH)	Bit symbol	P4MON/ P4EN	-	-	-	-	-	-	-
	Read/Write	R/W							
	After Reset	0	-	-	-	-	-	-	-
Read-modify-write not allowed	Function	Internal clock 0: Stop 1: Operate							

Baud rate clock control

0	Stop
1	Operate

Serial Bus Interface 1 Data Buffer Register

		7	6	5	4	3	2	1	0
SBI1DBR (0179H)	Bit symbol	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
	Read/Write	R (received)/W (transfer)							
Read-modify-write not allowed	After Reset	Undefined							

Note: When writing transmit data, justify the data toward the MSB (bit 7) side.

I^2C Bus 1 Address Register

		7	6	5	4	3	2	1	0
I2C1AR (017AH)	Bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
	Read/Write	W							
	After Reset	0	0	0	0	0	0	0	0
Read-modify-write not allowed	Function	Slave address selection for when device is operating as slave device							Addressing or free data format

Address recognition mode specification

0	Addressing format
1	Free data format

The addressing or free data format affects both the slave and master.

When using the addressing format (<ALS>=0), the TRX bit is updated with the direction bit, R/W (bit 8 of the first byte received after the start condition). In addition, in slave mode, the MCU finds the bus after the start condition with which it recognizes the address.

When using the free data format (<ALS>=1), TRX remains unchanged because all words on the bus are not recognized as an address but as data words.

Figure 3.10.10 Registers for the I^2C Bus Mode

Serial Bus Interface 2 Control Register 1

		7	6	5	4	3	2	1	0
SBI2CR1 (0180H)	Bit symbol	BC2	BC1	BC0	ACK	SCK3	SCK2	SCK1	SWRMON/ SCK0
	Read/Write		W		R/W		W		R/W
	After Reset	0	0	0	0	1	0	0	1/0(Note3)
Read-modify-write not allowed	Function	Number of transferred bits (Note 1)			Acknowledge mode specification 0: Not generate 1: Generate	Internal serial clock selection and software reset monitor (Note 2)			

Internal serial clock selection <SCK3 to 0> @ write

0001	n = 6	–	CPU clock: fc = 20 MHz internal SCL output $(fscl = \dfrac{fc}{2^n + 8}\ [\,Hz\,])$
0010	n = 7	–	
0011	n = 8	75.8 kHz	
0100	n = 9	38.5 kHz	
0101	n = 10	19.4 kHz	
0110	n = 11	9.73 kHz	
1000	Fast	400 kHz	$(fscl = fc/50\ [\,Hz\,])$
1111	Standard	100 kHz	$(fscl = fc/200\ [\,Hz\,])$
other	-	(reserved)	

Software reset state monitor <SWRMON> @ read

0	During software reset
1	Initial data

Acknowledge mode specification

0	Not generate clock pulse for acknowledge signal
1	Generate clock pulse 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: It is necessary to clear <BC2:0> to 000 before attempting to change the operating mode to clock synchronous 8-bit SIO mode.
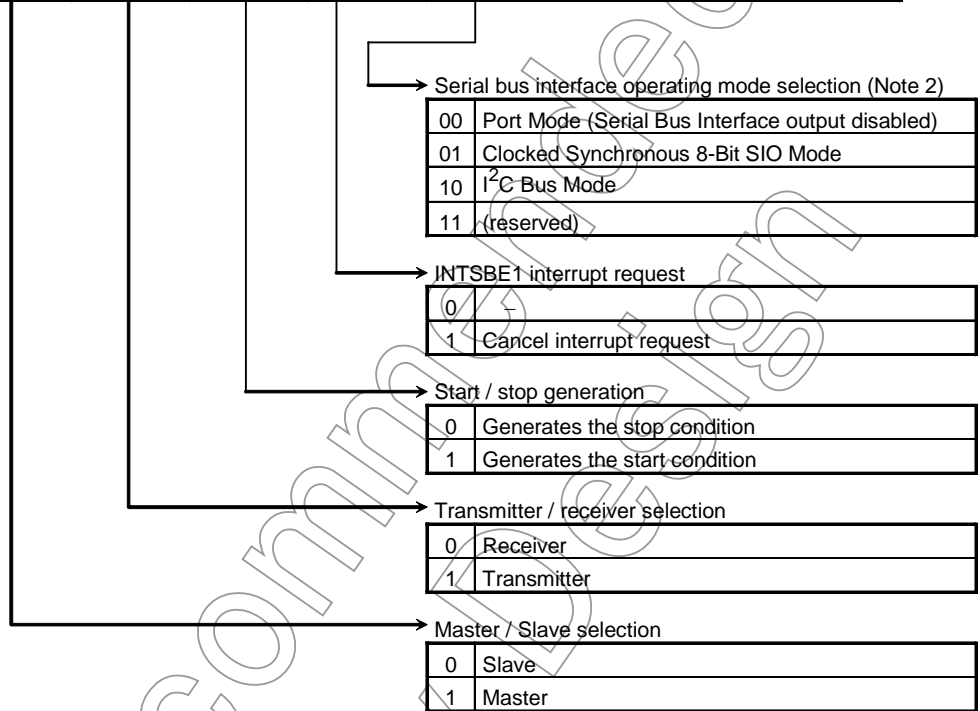
Note 2: For details of the SCL line clock frequency, see "3.10.5 (3) Serial clock."

Note 3: The initial values of SCK2 and SWRMON are 0 and 1, respectively.

Figure 3.10.11 Registers for the I²C Bus Mode

Serial Bus Interface 2 Control Register 2

		7	6	5	4	3	2	1	0
SBI2CR2	Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
(0183H)	Read/Write	W				W (Note 1)		W (Note 1)	
	After Reset	0	0	0	1	0	0	0	0
Read-modify-write not allowed	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: I^{2}C Bus Mode 11: (reserved)		Software reset generate write "10" and "01", then an internal reset signal is generated.	

Serial bus interface operating mode selection (Note 2)

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

INTSBE1 interrupt request

0	―
1	Cancel interrupt request

Start / stop generation

0	Generates the stop condition
1	Generates the start condition

Transmitter / receiver selection

0	Receiver
1	Transmitter

Master / Slave selection

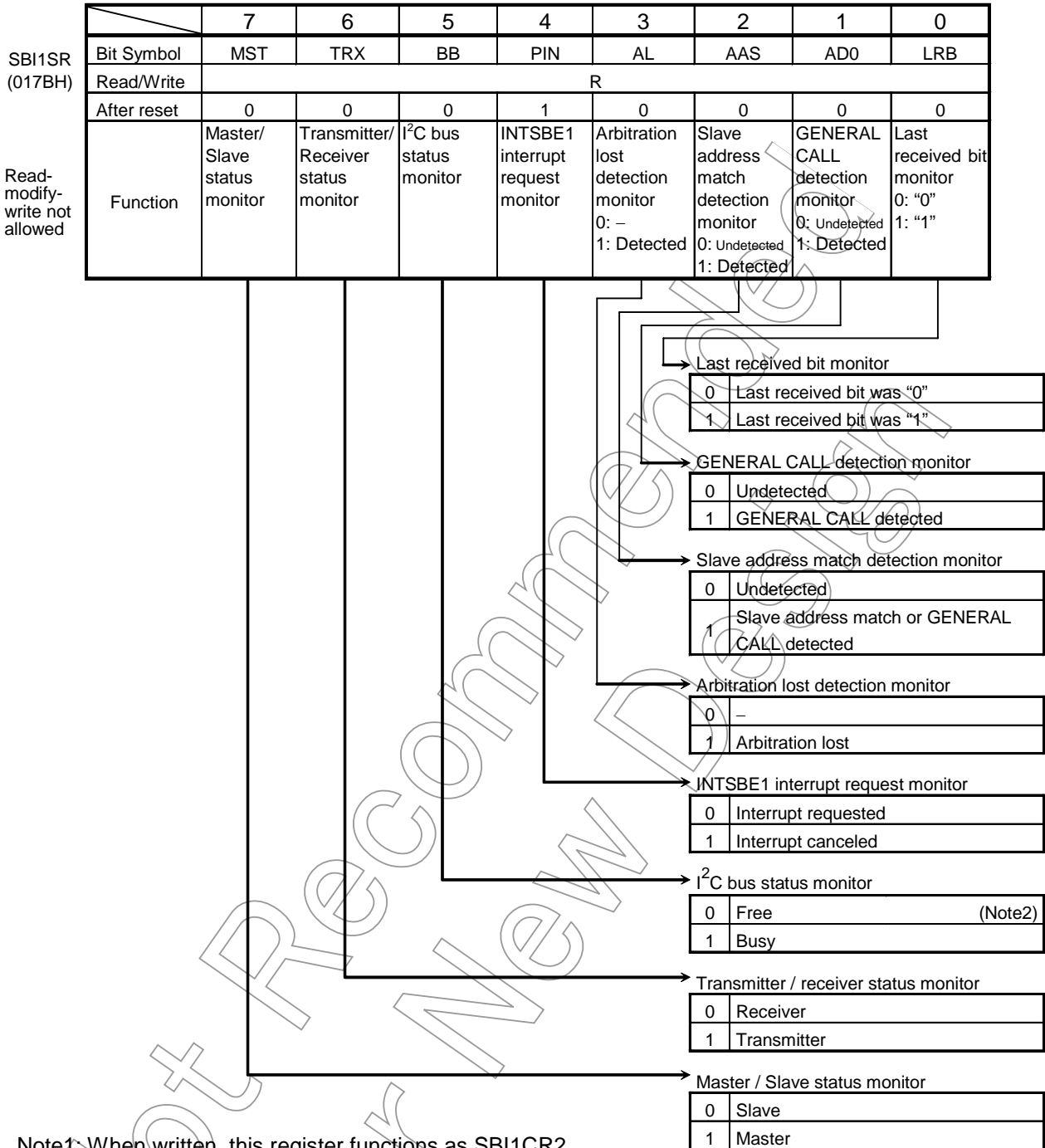| 0	Slave
1	Master

Note1: When read, this register functions as SBI2SR.
Note2: Ensure that the bus is free before attempting to select port mode.
Also ensure that the port state is High before attempting to change the mode from port mode to I^{2}C bus or clock synchronous 8-bit SIO mode.

Figure 3.10.12 Registers for the I^{2}C Bus Mode

Serial Bus Interface 2 Status Register

		7	6	5	4	3	2	1	0
SBI2SR (017BH)	Bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
	Read/Write	R							
	After reset	0	0	0	1	0	0	0	0
Read-modify-write not allowed	Function	Master/ Slave status monitor	Transmitter/ Receiver status monitor	I^{2}C bus status monitor	INTSBE1 interrupt request monitor	Arbitration lost detection monitor 0: − 1: Detected	Slave address match detection monitor 0: Undetected 1: Detected	GENERAL CALL detection monitor 0: Undetected 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	Undetected
1	GENERAL CALL detected

Slave address match detection monitor

0	Undetected
1	Slave address match or GENERAL CALL detected

Arbitration lost detection monitor

0	−
1	Arbitration lost

INTSBE1 interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I^{2}C bus status monitor

0	Free	(Note2)
1	Busy	

Transmitter / receiver status monitor

0	Receiver
1	Transmitter

Master / Slave status monitor
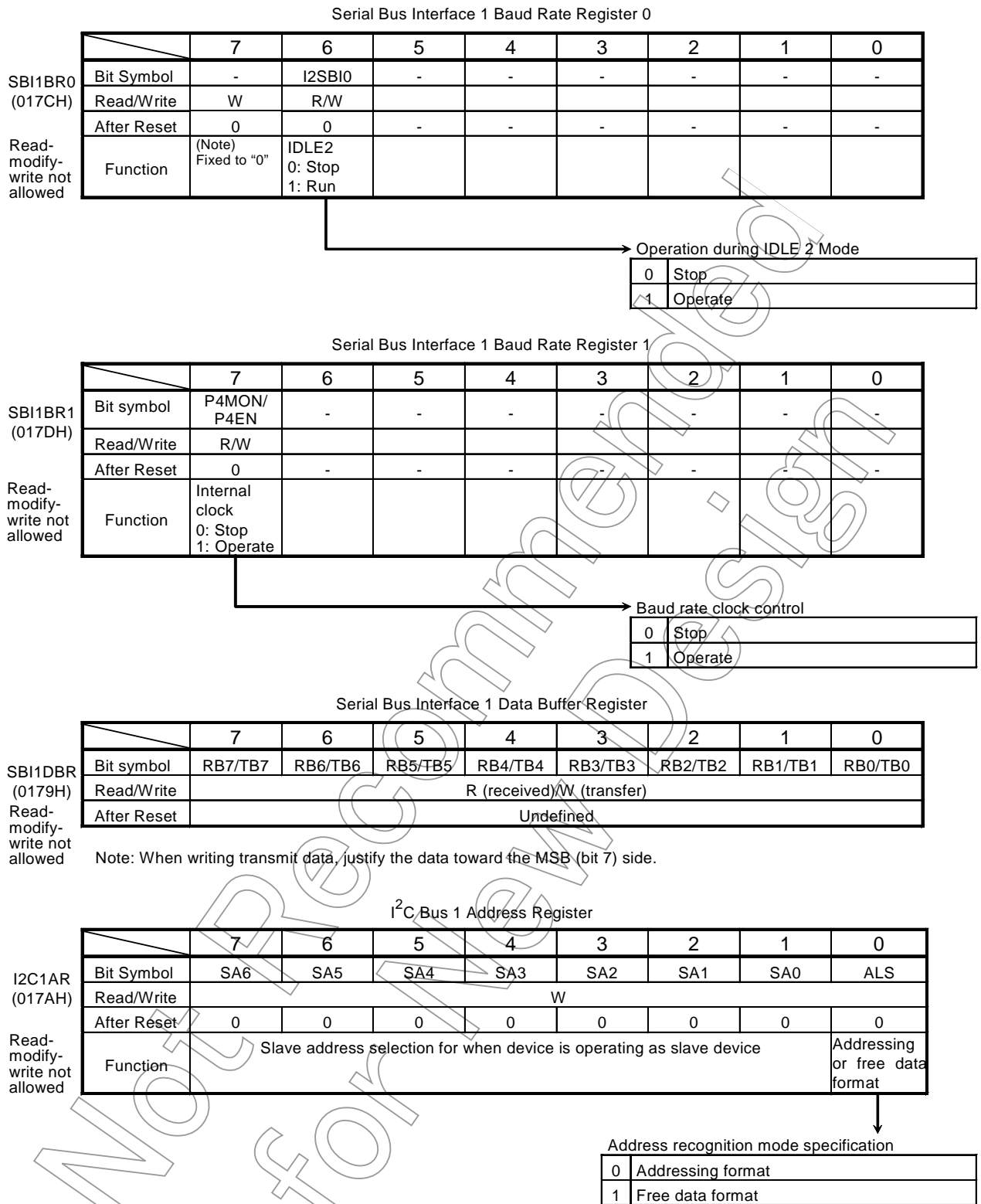
0	Slave
1	Master

Note1: When written, this register functions as SBI2CR2.
Note2: When the BB flag changes its state from 1 to 0 (falling edge), INTSBS2 occurs.

Figure 3.10.13 Registers for the I^{2}C Bus Mode

Serial Bus Interface 2 Baud Rate Regster 0

SBI2BR0 (0184H) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	Bit Symbol	-	I2SBI0	-	-	-	-	-	-
	Read/Write	W	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 2 Baud Rate Register 1

SBI2BR1 (0185H) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	Bit symbol	P4MON/ P4EN	-	-	-	-	-	-	-
	Read/Write	R/W							
	After Reset	0	-	-	-	-	-	-	-
	Function	Internal clock 0: Stop 1: Operate							

→ Baud rate clock control

0	Stop
1	Operate

Sirial Bus Interface 2 Data Buffer Register

SBI2DBR (0181H) Read-modify-write not allowed		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 transmit data, justify the data toward the MSB (bit 7) side.

I^2C Bus 2 Address Register

I2C2AR (0182H) Read-modify-write not allowed		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 device							Addressing or free data format

↓ Address recognition mode specification

0	Addressing format
1	Free data format

The addressing or free data format affects both the slave and master.

When using the addressing format (<ALS>=0), the TRX bit is updated with the direction bit, R/W (bit 8 of the first byte received after the start condition). In addition, in slave mode, the MCU finds the bus after the start condition with which it recognizes the address.

When using the free data format (<ALS>=1), TRX remains unchanged because all words on the bus are not recognized as an address but as data words.

Figure 3.10.14 Registers for the I^2C Bus Mode

3.10.5 Control in I^{2}C Bus Mode

(1) Specifying acknowledgment mode

When SBI0CR1 <ACK> is set to 1, the serial bus interface operates in acknowledgment mode. When operating as the master, it adds a single clock cycle for an acknowledge signal. When operating as a slave, it counts a clock cycle for an acknowledge signal. In transmitter mode, the serial bus interface relinquishes the SDA0 pin during that clock cycle so that it can receive an acknowledge signal from the receiver. In receive mode, it pulls the SDA0 pin Low during that clock cycle to generate an acknowledge signal.

When SBI0CR1<ACK> is set to 0, the serial bus interface operates in non-acknowledgment mode: When operating as the master, it does not add a clock cycle for an acknowledge signal. When operating as a slave, it does not count a clock cycle for an acknowledge signal.

(2) Selecting the number of bits to be transferred

The number of bits to be transmitted or received is selected using SBI0CR1 <BC2:0>.

The slave address and the direction bit are always transferred in eight bits because the start condition clears SBI0CR1 <BC2:0> to 000. In other cases, SBI0CR1 <BC2:0> maintains the value once it has been set.

(3) Serial clock

a. Clock source

The SBI0CR1 <SCK3:0> bits select the maximum transfer frequency for the serial clock that is output through the SCL0 pin in master mode.
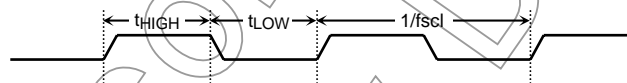
Formula	SBI0CR1 <SCK3 to 0>	n
$t_{LOW} = 2^{n-1} / fc$ $t_{HIGH} = 2^{n-1} / fc + 8 / fc$ $fscl = 1 / (t_{LOW} + t_{HIGH})$ $= fc / (2^n + 8)$	0011 0100 0101 0110	8 9 10 11
$t_{LOW} = 32 / fc$, $t_{HIGH} = 18 / fc$ $fscl = fc / 50$	1000	–
$t_{LOW} = 100 / fc$, $t_{HIGH} = 100 / fc$ $fscl = fc / 200$	1111	–

Figure 3.10.15 Clock Source

b.   Clock synchronization

The I²C bus is driven in a wired-AND manner due to the pin structure. Therefore, the first master that has pulled the clock line Low disables the clock for any other master outputting a High level. The master outputting a High level should detect that condition and take appropriate action.

The TMP92CD54I supports clock synchronization to ensure normal transfer even if multiple masters with different transfer rates exist on the bus.

The following describes an example clock synchronization procedure when two masters are simultaneously operating on the bus:
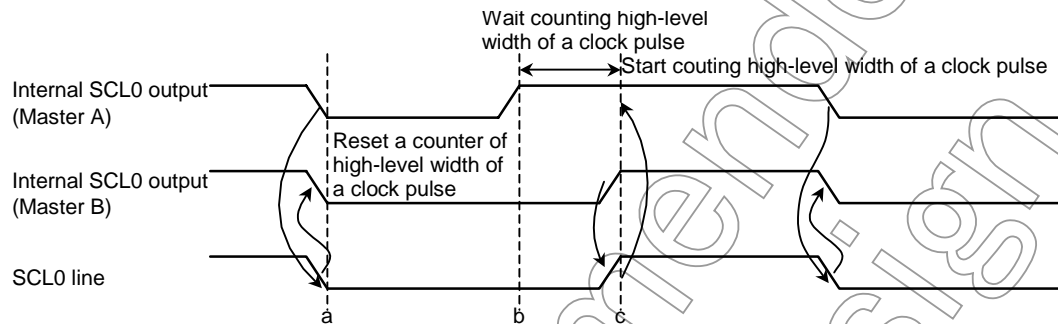
Figure 3.10.16 Clock Synchronization

At point "a", master A pulls its internal SCL0 output Low, causing the SCL0 line on the bus to be driven Low. Detecting that transition, master B resets the High period count and pulls its internal SCL0 output Low.

At point "b", master A completes counting the Low period and drives its internal SCL0 output High. However, since master B is maintaining the SCL0 bus line Low, master A stops counting the High period. At point "c", master B drives its internal SCL0 output High, causing the SCL0 bus line to be driven High. Upon detecting that transition, master A starts counting the High period.

As shown above, when more than one master is connected on the bus, the clock on the bus is determined by the master with the shortest High period and that with the longest Low period.

c.   Effects of the SCL rise time on the transfer rate

Clock synchronization inserts a wait time for the rise time on the SCL line. In that case, the actual transfer rate is slower than the value described in the data sheet.

The following shows details and examples:

SCL-period (T-period) = T-LOW + T-HIGH
T-HIGH = T-R + tHIGH, T-R = tr + [Wait time for counting at high level]

SCL0 line

	T-LOW	T-R	T-HIGH
	tLOW / t_r	T-R	tHIGH
SCK3:0 (0001 ~ 0110)	$2^{n-1}/fc$	T-R	$(2^{n-1}+8)/fc$
SCK3:0 (1111) : 100KHz at fc=20MHz	100/fc	T-R	100/fc
SCK3:0 (1000) : 400KHz at fc=20MHz	32/fc	T-R	18/fc

	t_r	0~2/fc	2/fc~4/fc	4/fc~6/fc	6/fc~8/fc	8/fc~10/fc
SCK3:0 (0001 ~ 0110)	T-R	0	4/fc		8/fc	
SCK3:0 (1111) : 100KHz	T-R	0	4/fc		8/fc	
SCK3:0 (1000) : 400KHz	T-R	0	2/fc	4/fc	6/fc	8/fc
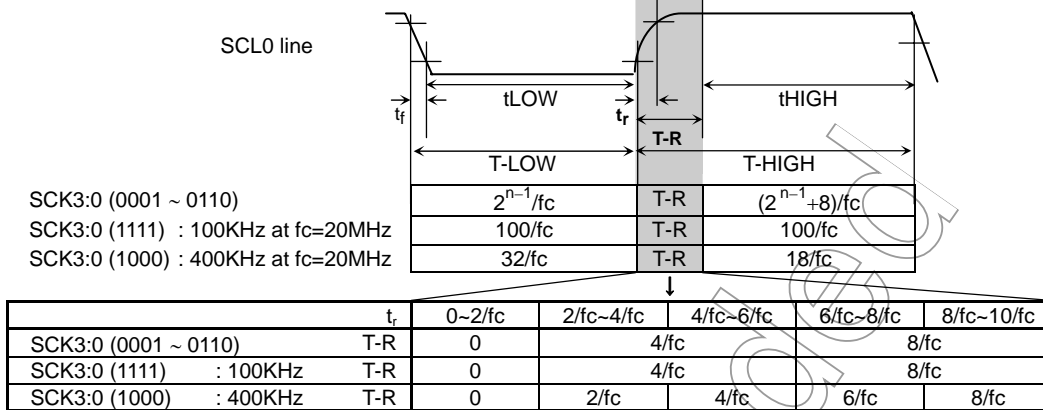
Figure 3.10.17 Insert Wait Time (T-R) by Rising Time of SCL0 Line (t_r)

If the SCL0 rise time, tr, is less than 2/fc, a synchronization wait time is not inserted.

If it is 2/fc or greater, T-HIGH is extended by the period of T-R, resulting in a slower transfer rate.

Example1: (1) In the case, fc = 20MHz, <SCK3:0> = 0011 and t_r = 50ns:
            T-R = 0ns. since t_r = 50ns.
            T-Period = $2^{n-1}/fc + 0 + (2^{n-1} + 8)/fc$ = 266/fc = 13.2 µs (75.8KHz)

          (2) In the case, fc = 20MHz, <SCK3:0> = 0011 and t_r = 250ns:
            T-R = 4/fc, since t_r = 250ns.
            T-Period = $2^{n-1}/fc + 4/fc + (2^{n-1} + 8)/fc$ = 268/fc = 13.4µs (74.6KHz)

Example2: (1) In the case, fc = 20MHz, <SCK3:0> = 1111 and t_r = 50ns:
            T-R = 0ns, since t_r = 50ns.
            T-Period = 100/fc + 0 + 100/fc = 10µs (100KHz)

          (2) In the case, fc = 20MHz, <SCK3:0> = 1111 and t_r = 150ns:
            T-R = 4/fc, since t_r = 150ns.
            T-Period = 100/fc + 4/fc + 100/fc = 10.2µs (98.0KHz)

Example3: (1) In the case, fc = 20MHz, <SCK3:0> = 1000 and t_r = 50ns:
            T-R = 0ns, since t_r = 50ns.
            T-Period = 32/fc + 0 + 18/fc = 2.5µs (400KHz)

          (2) In the case, fc = 20MHz, <SCK3:0> = 1000 and t_r = 150ns:
            T-R = 2/fc, since t_r = 150ns.
            T-Period = 32/fc + 2/fc + 18/fc = 2.6µs (384.6KHz)

(4) Setting the slave address and address recognition mode

To operate the TMP92CD54I as a slave device, set the slave address <SA6:0> and <ALS> in I2C0AR.

Clearing I2C0AR <ALS> to 0 selects address recognition mode (addressing format).

(5) Selecting the master or slave

Setting SBI0CR2 <MST> to 1 causes the TMP92CD54I to operate as a master device.

Clearing SBI0CR2 <MST> to 0 causes the TMP92CD54I to operate as a slave device. SBI0CR2<MST> is cleared by hardware upon the detection of a stop condition or arbitration lost on the bus.

(6) Selecting the transmitter or receiver

SBI0CR2 <TRX> selects either transmitter or receiver operation. Setting <TRX> to 1 causes the TMP92CD54I to operate as a transmitter. Setting <TRX> to 0 causes the TMP92CD54I to operate as a receiver.

When transferring data using the addressing format in slave mode, the device receives the slave address and direction bit in the first byte. If the received slave address matches the value of I2C0AR (the device's slave address), the value of <TRX> varies according to the direction bit. If $R/\overline{W}$ = 0 (slave reception), <TRX> is cleared to 0 and an acknowledge signal is returned to receive subsequent data. If R/W = 1 (slave transmission), <TRX> is set to 1 and an acknowledge signal is returned to transmit subsequent data. For a general call, where all bits of the first byte are 0, R/W = 0 so that <TRX> is cleared to 0 and an acknowledge signal is returned to receive subsequent data.

In master mode, when an acknowledge signal is returned from the slave device, the value of <TRX> varies depending on the value of R/W that has been sent. If R/W = 0 (master transmission), <TRX> is set to 1. If R/W = 1 (master reception), <TRX> is cleared to 0. If no acknowledge signal is returned, <TRX> maintains the previous value.

<TRX> is cleared by hardware upon the detection of a stop condition or arbitration lost on the I²C bus.

(7) Generating start and stop conditions

When SBI0SR <BB> is 0, writing 1111 to SBI0CR2 <MST, TRX, BB, PIN> causes a start condition and the 8-bit data in SBI0DBR to be output on the bus. SBI0CR1<ACK> must be set to 1 before the start condition.
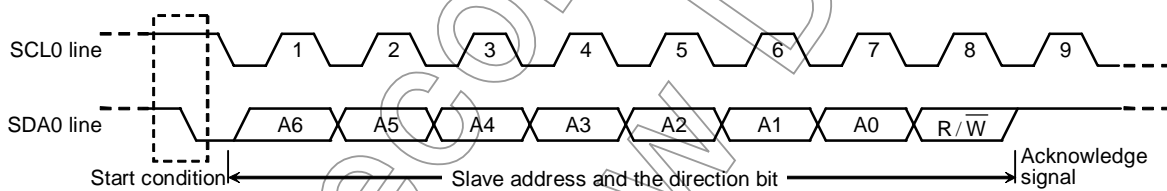
Figure 3.10.18 Start Condition Generation and Slave Address Generation

When SBI0SR <BB> is 1, writing 111 to SBI0CR2 <MST, TRX, PIN> and 0 to SBI0CR2<BB> causes a stop condition output sequence to start on the bus. Do not rewrite the contents of SBI0CR2 <MST, TRX, BB, PIN> until a stop condition occurs on the bus.
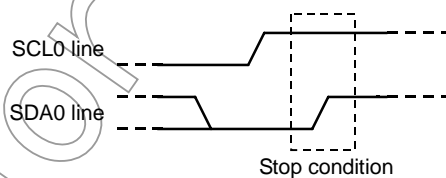
Figure 3.10.19 Stop Condition Generation

The bus status can be determined by reading SBI0SR <BB>. SBI0SR <BB> is set to 1 if a start condition is detected on the bus (bus busy state) and cleared to 0 if a stop condition is detected on the bus (bus free state). When the SBI0SR<BB> flag changes its state from 1 to 0 (falling edge), INTSBS0 occurs.

(8) Issuing and releasing an interrupt service request

When serial bus interface interrupt request 0 (INTSBE0) is issued due to a slave address or data transfer, SBI0SR <PIN> is cleared to 0. The SCL0 line is pulled Low while SBI0SR <PIN> is 0.

SBI0SR <PIN> is cleared to 0 once a single word has been transmitted or received. It is set to 1 once data has been written to SBI0DBR or read from SBI0DBR.

It requires a time of tLOW between SBI0SR <PIN> being set to 1 and the SCL0 line being relinquished.

In address recognition mode (I2C0AR<ALS> = 0), SBI0CR2<PIN> is cleared to 0 if the received slave address matches the value set in I2C0AR or when a general call (where all bits of the 8-bit data after the start condition are 0) is received. A program can write a 1 to SBI0CR2 <PIN> to set it to 1. When it writes a 0, however, the bit is not cleared to 0.

(9) Operating mode of the serial bus interface

The SBI0CR2 <SBIM1:0> bits specify the operating mode of the serial bus interface.

To use it in I²C bus mode, set SBI0CR2<SBIM1:0> to 10.

Ensure that the bus is free before attempting to change the operating mode to port mode.

(10) Arbitration lost detection monitor

The I²C bus allows multi-master operation (two or more masters can simultaneously exist on a single bus), thus requiring a bus arbitration procedure to guarantee the contents of transferred data.

The I²C bus uses data on the SDA0 line for bus arbitration.

The following describes an example arbitration procedure when two masters are simultaneously operating on the bus: Both masters A and B output the same data up until the bit at point "a". At point "a", master A outputs a Low level while master B outputs a High level. Since the SDA0 line on the bus is driven in a wired-AND manner, it is pulled Low by master A. When the SCL0 bus line rises at point "b", the slave device fetches data on the SDA0 line, that is, data from master A. At this time, data output from master B is invalid. That state of master B is called "arbitration lost." Master B relinquishes the SDA pin to prevent it from affecting data output from other masters. If more than one master transmits exactly the same data in the first word, the arbitration procedure continues for the second and subsequent words.
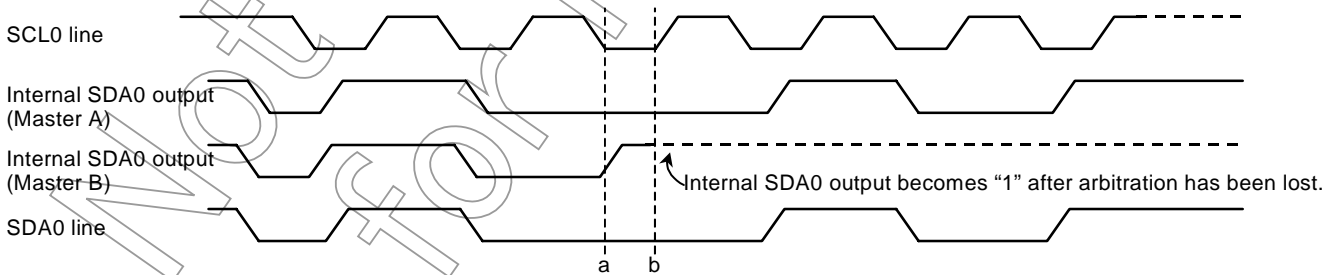
Figure 3.10.20 Arbitration Lost

The level of the SDA0 bus line is compared with the internal SDA0 output level on the rising edge of the SCL0 line. If they do not match, SBI0SR <AL> is set to 1 to indicate the arbitration lost state.

When SBI0SR <AL> is set to 1, the SBI0SR <MST, TRX> bits are reset to 00, causing a transition to slave receive mode. The serial clock is, however, output until the end of data transfer that was being transmitted when SBI0SR <AL> changed to 1.

SBI0SR <AL> is reset to 0 by writing data to SBI0DBR, reading data from SBI0DBR, or writing data to SBI0CR2.
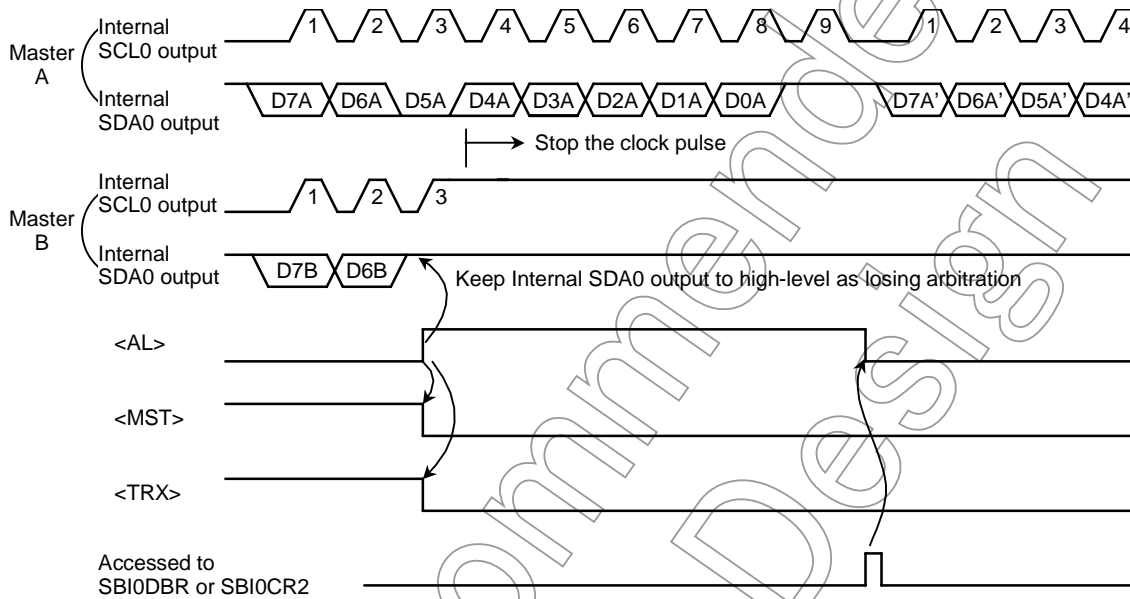
Figure 3.10.21 Example of a Master Device B
(D7A = D7B, D6A = D6B)

(11) Slave address match detection monitor

In slave mode, SBI0SR <AAS> is set to 1 if the device receives a general call or the same slave address as that set in I2C0AR in address recognition mode (I2C0AR <ALS> = 0). If I2C0AR <ALS> = 1, SBI0SR <AAS> is set to 1 when the first word is received. SBI0SR <AAS> is cleared to 0 by writing data to SBI0DBR or reading data from SBI0DBR.

(12) General call detection monitor

In slave mode, SBI0SR <AD0> is set to 1 when a general call (where all bits of the 8-bit data after the start condition are 0) is received, and cleared to 0 when a start or stop condition is detected on the bus.

(13) Last received bit monitor

The value on the SDA0 line is captured on the rising edge of the SCL0 line and set in SBI0SR <LRB>.

In acknowledgment mode, reading SBI0SR <LRB> immediately after an INTSBE0 interrupt request is issued results in the ACK signal being read.

(14) Software reset

If the serial bus interface circuit is locked due to external noise, the software reset function can be used to initialize the serial bus interface circuit.

To initialize the serial bus interface circuit, first write 10 and then 01 to SBI0CR2 <SWRST1:0>, causing a reset signal to be applied to the circuit. This initializes the values in all control and status registers.

Initializing the serial bus interface causes <SWRST1:0> to be automatically cleared to 00.

Note: Initialization requires approximately 1.4 μs (when fc = 20 MHz). SBI0CR1 <SWRMON> can be monitored to determine whether initialization has been completed.

(15) Serial bus interface data buffer register (SBI0DBR)

SBI0DBR is read or written to read received data or write transmit data.

In master mode, the device generates a start condition after the slave address and direction bit are sets in this register.

(16) I²C bus address register (I2C0AR)

The I2C0AR <SA6:0> bits set a slave address when the TMP92CD54I operates as a slave device.

If I2C0AR <ALS> is set to 0, the TMP92CD54I recognizes the slave address output from the master device and uses the addressing data format.

If I2C0AR <ALS> is set to 1, the TMP92CD54I does not recognizes the slave address and uses the free data format.

(17) Baud rate register (SBI0BR1)

It is necessary to write a 1 to SBI0BR1 <P4EN> before attempting to use the I²C bus.

(18) IDLE2 setup register (SBI0BR0)

The SBI0BR0 <I2SBI0> bit enables or disable the operation when the TMP92CD54I enters IDLE2 mode.

It is necessary to set this bit before attempting to execute the HALT instruction.

3.10.6 Data Transfer Procedure in I²C Bus Mode

(1) Initializing the device

First, it is necessary to set SBI0BR1 <P4EN> and SBI0CR1 <ACK, SCK2:0>. Write a 1 to SBI0BR1 <P4EN> and 0s to SBI0CR1 bits 7, 6, 5 and 3.

Next, set <SA6:0> (slave address) and <ALS> (0 for the addressing format) in I2C0AR.

Then, write 000 to SBI0CR2 <MST, TRX, BB>, 1 to <PIN>, 10 to <SBIM1:0>, 00 to <SWRST1:0>, and set the initial state to slave receiver mode.

(2) Generating a start condition and slave address

a. In master mode

In master mode, a start condition and slave address are generated using the following procedure:

First, ensure that the bus is free (SBI0CR2<BB> = 0).

Next, write a 1 to SBI0CR1 <ACK> to select acknowledgment mode. In SBI0DBR, write the slave address and the direction bit to which data will be transmitted.

When SBI0CR2<BB> is 0, writing 1111 to SBI0CR2 <MST, TRX, BB, PIN> causes a start condition to be generated on the bus. Following the start condition, output nine clock cycles on the SCL0 pin. In the first eight clock cycles, output the slave address and direction bit stored in SBI0DBR. In the ninth clock cycle, relinquish the SDA0 line and receive an acknowledge signal from the slave device.

On the falling edge of the ninth clock cycle, an INTSBE0 interrupt request occurs and SBI0CR2<PIN> is cleared to 0. In master mode, pull the SCL0 line Low while SBI0CR2<PIN> is 0. Only when an acknowledge signal is returned from the slave device, an INTSBE0 interrupt request occurs and the value of SBI0CR2<TRX> changes depending on the direction bit transmitted.

b. In slave mode

In slave mode, a start condition and slave address are received.

After receiving a start condition from the master device, receive the slave address and direction bit from the master device in the first eight clock cycles on the SCL0 line. If the received address indicates a general call or matches the slave address set in I2C0AR, pull the SDA0 line Low in the ninth clock cycle to output an acknowledge signal.

On the falling edge of the ninth clock cycle, an INTSBE0 interrupt request occurs and SBI0CR2<PIN> is cleared to 0. In slave mode, pull the SCL0 line Low while SBI0CR2<PIN> is 0. Only when an acknowledge signal is returned from the slave device, an INTSBE0 interrupt request occurs and the value of SBI0CR2<TRX> changes depending on the direction bit received.
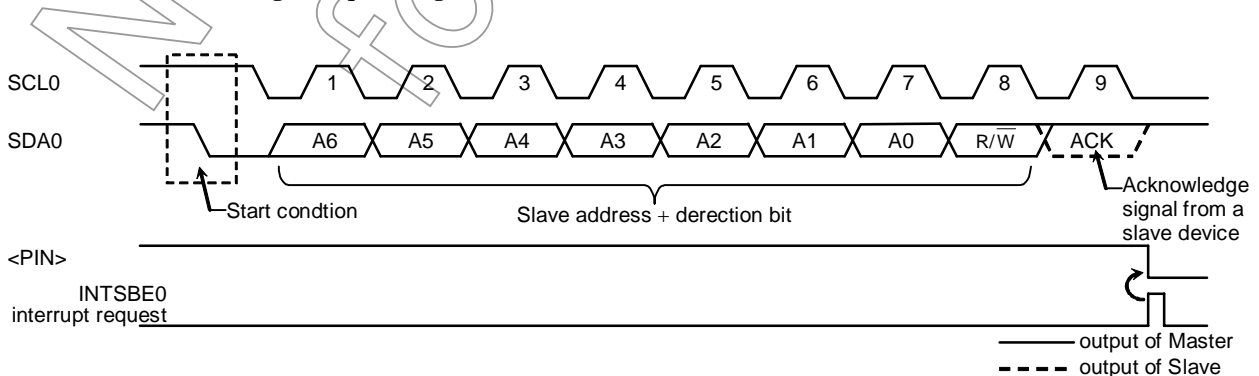
Figure 3.10.22 Start Condition Generation and Slave Address Transfer

(3) Transferring a single word of data

When handling an INTSBE0 interrupt upon the end of transferring a single word, test SBI0CR2<MST> to determine whether the mode is master or slave mode.

a.   In master mode (SBI0CR2<MST> = 1)

Test SBI0CR2<TRX> to determine whether the TMP92CD54I is the transmitter or receiver.

In transmitter mode (SBI0CR2<TRX> = 1)

Test SBI0SR<LRB>. If SBI0SR <LRB> = 1, the receiver is not requesting data. In that case, generate a stop condition (see 3.10.6 (4)) and complete data transfer.

If SBI0SR <LRB> is 0, the receiver is requesting next data. If the data to be transferred next consists of eight bits, write the transfer data to SBI0DBR. It the data consists of other than eight bits, set SBI0CR1<BC2:0> and <ACK> before writing the transfer data to SBI0DBR.

Once the data has been written, SBI0SR <PIN> is set to 1, after which the SCL0 pin generates a serial clock for transferring the next word of data and the SDA0 pin transfers the word. Upon the completion of transfer, an INTSBE0 interrupt request occurs, SBI0SR <PIN> is set to 0, and the SCL0 line is pulled Low. To transfer multiple words, repeat the above SBI0SR <LRB> test and subsequent steps.
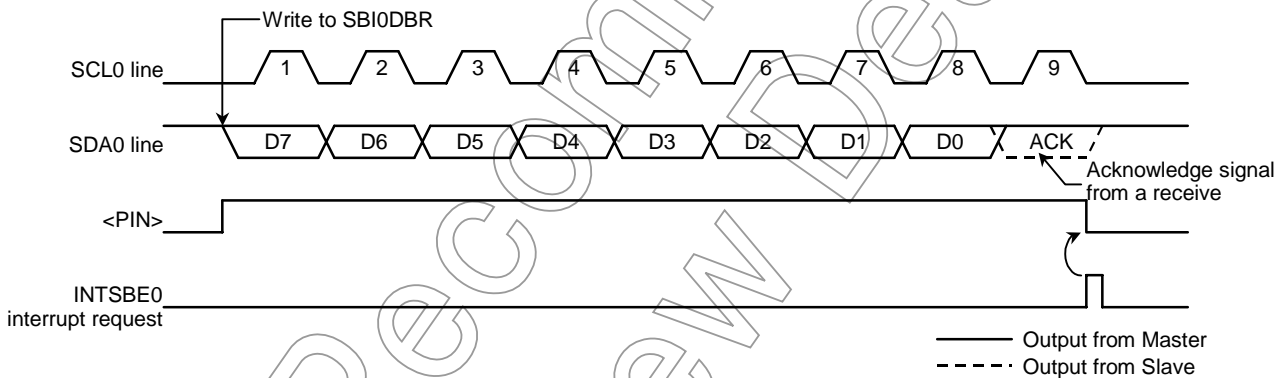
Figure 3.10.23 Example in which <BC2 to 0> = "000" and <ACK> = "1" in Transmitter Mode

In receiver mode (SBI0SR<TRX> = 0)

If the data to be transferred consists of other than eight bits, set SBI0CR1 <BC2:0> and <ACK> and read the received data from SBI0DBR to relinquish the SCL0 line (the data is undefined if it is read immediately after the slave address is transmitted). Reading data causes SBI0CR2<PIN> to be set to 1 and a serial clock for transferring the next data word to be output on the SCL0 pin. For the last bit, a 0 is output on the SDA0 pin when the acknowledge signal goes Low.

Then, an INTSBE0 interrupt request occurs, SBI0CR2<PIN> is set to 0, and the SCL0 line is pulled Low. A transfer clock for a single word and an acknowledge signal are output every time the received data is read from SBI0DBR.
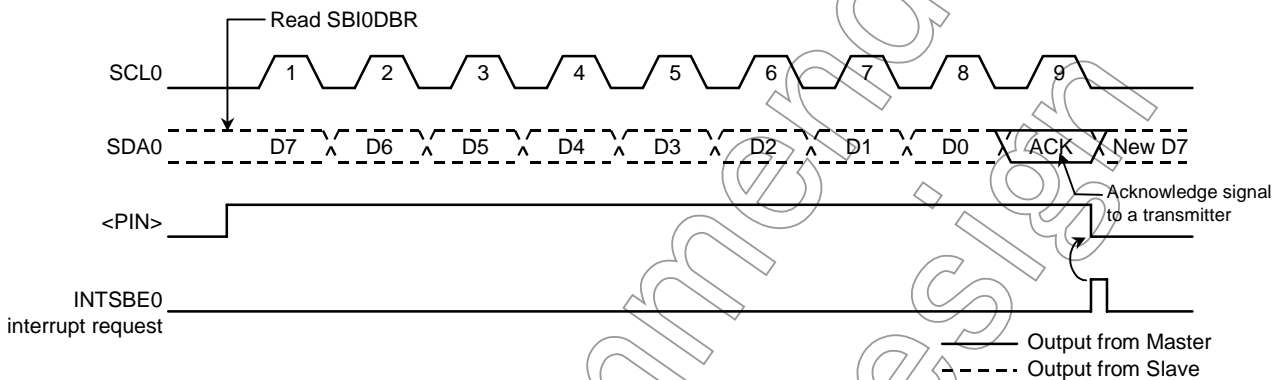
Figure 3.10.24 Example of when <BC2 to 0> = "000", <ACK> = "1" in Receiver Mode

To request the transmitter to terminate data transmission, clear SBI0CR1 <ACK> to 0 before reading the word of data preceding the word to be received last. This prevents a clock for acknowledging the last data from being generated. During processing after the transfer end interrupt request is issued, set SBI0CR1<BC2:0> to 001 and read data, which causes a clock for single-bit transfer to be generated. At this time, the master is the receiver so that the SDA0 line on the bus remains High level. The transmitter receives this High level as an ACK signal, with which the receiver can notify the transmitter of the completion of transfer.

During processing after the reception end interrupt request for that single-bit transfer, generate a stop condition to complete data transfer. The generation of the stop condition (see 3.10.6 (4)) causes an INTSBS0 interrupt request.
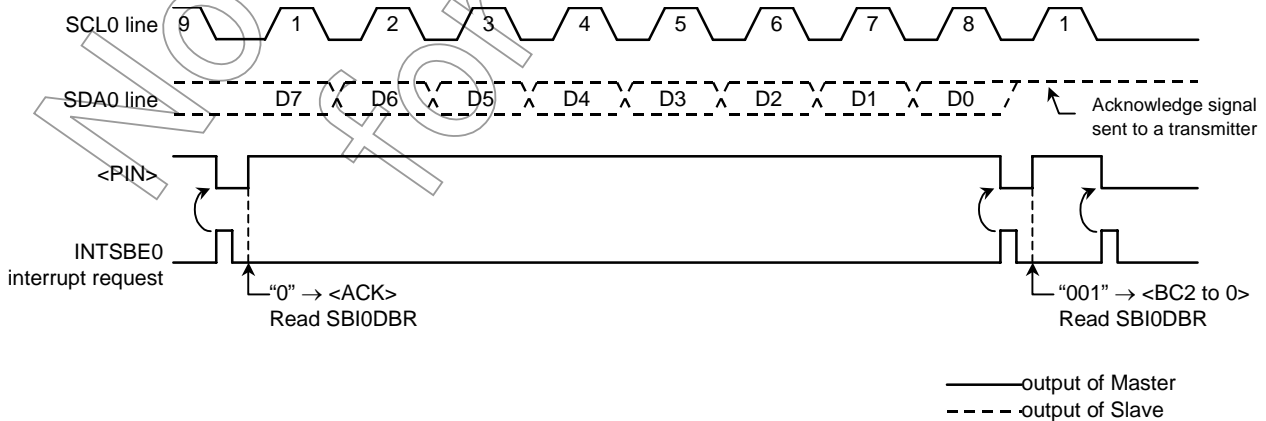
Figure 3.10.25 Termination of Data Transfer in Master Receiver Mode

b.   In slave mode (SBI0CR2<MST> = 0)

Processing in slave mode is classified into processing normally performed in slave mode and processing performed when the device detects arbitration lost and enters slave mode.

The following describes when an INTSBE0 interrupt request is issued in each case:

- In normal slave mode:
    (1) When the addressing format is used and the received slave address matches the address set in I2CAR. Alternatively, when a general call is received.
    (2) When data transfer has been completed
- If the device changes from master mode to slave mode due to arbitration lost:
    (1) When the transfer of the word with which arbitration lost was detected is completed
        When an INTSBE0 interrupt request occurs, SBI0CR2<PIN> is cleared to 0, and the SCL0 line is pulled Low. Once data is written to or read from SBI0DBR or SBI0CR2<PIN> is set to 1, the SCL0 pin is relinquished in a period of t_{LOW}.

Note: SBI0CR2<PIN> is set to 0 and the SCL0 pin is pulled Low only if the TMP92CD54I detects arbitration lost while transmitting a slave address as a master and the TMP92CD54I itself is called as a slave device (slave address match).
If it detects arbitration lost while transmitting a slave address as a master but the slave address does not match, or if it detects arbitration lost while transmitting data as a master, an INTSBE0 interrupt request occurs upon the completion of transferring the word with which arbitration lost was detected, but SBI0CR2<PIN> is not cleared to 0.

When the SBI0SR<BB> flag changes its state from 1 to 0 (falling edge), INTSBS0 occurs.

In slave mode, test the SBI0SR <AL>, <TRX>, <AAS>, and <AD0> bits and then determine the status from the combination of those values to take appropriate action.

Table 3.10.3 shows the states in slave mode and required operations.

Table 3.10.3 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	In the master transmitter mode, this device detects the arbitration lost during transferring the slave address. It turns into the slave receiver mode. After that this device finishes receiving from other masters the slave address which is the same as that of this device and the direction bit "1".	This device operates in the slave transmitter mode. Set the number of bits in 1-word to the <BC2 to 0> and write the transmitted data to the SBI0DBR. (<PIN> is set to "1" and SCL0 is released.)
	0	1	0	In the slave receiver mode, this device finishes receiving from master the slave address which is the same as that of this device and the direction bit "1".	
		0	0	In the slave transmitter mode, this device finishes transmitting 1-word data.	This device operates in the slave transmitter mode. 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> to "0" to release the bus. If the <LRB> is cleared to "0", set the number of bits in 1-word to the <BC2 to 0> and write transmitted data to the SBI0DBR since the receiver requests next data.
0	1	1	1/0	In the master transmitter mode, this device detects the arbitration lost during transferring the slave address. It turns into the slave receiver mode. After that this device finishes receiving from other masters the slave address which is that of this device and the direction bit "0" or receiving the GENERAL CALL.	This device operates in the slave receiver mode. Read the SBI0DBR to set the <PIN> to "1" (reading dummy data) or set the <PIN> to "1".
		0	0	In the master transmitter mode, this device detects the arbitration lost during transferring the slave address. It turns into the slave receiver mode. After that this device finishes receiving from other masters the slave address which is not the same as that of this device and the direction bit. (The <PIN> is not cleared to "0".)	This device operates in the slave receiver mode. The <PIN> is not cleared to "0". In case that this device transmits again as a master, set up with software.
	0	1	1/0	In the slave receiver mode, this device finishes receiving from master the slave address which is the same as that of this device and the direction bit "0" or receiving the GENERAL CALL.	This device operates in the slave receiver mode. Read the SBI0DBR to set the <PIN> to "1" (reading dummy data) or set the <PIN> to "1".
		0	1/0	In the slave receiver mode, this device finishes receiving 1-word data.	This device operates in the slave receiver mode. Set the number of bits in 1-word to the <BC2 to 0> and read the received data in the SBI0DBR. (<PIN> is set to "1" and SCL0 is released.)

(4) Generating a stop condition

When SBI0SR <BB> is 1, writing 111 to SBI0CR2 <MST, TRX, PIN> and 0 to <BB> causes a stop condition output sequence to start on the bus. Do not rewrite the contents of SBI0CR2 <MST, TRX, BB, PIN> until a stop condition occurs on the bus.

If the SCL0 line on the bus has already been pulled by another device, the SDA0 line rises after the SCL0 line is relinquished, thus generating a stop condition. When SBI0SR <BB> is cleared to 0, an INTSBS0 interrupt request is issued as a result of a stop condition.
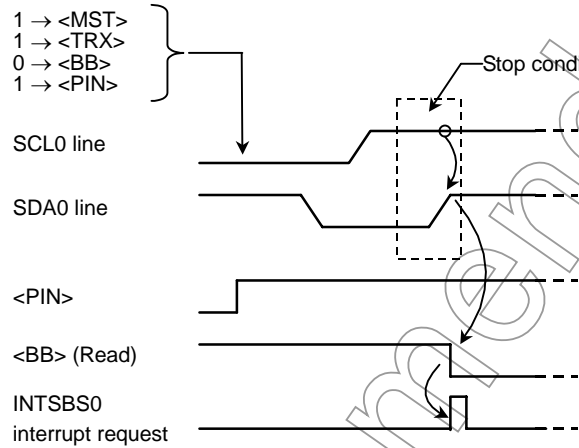
Figure 3.10.26 Stop Condition Generation

(5) Restart procedure

The restart procedure is used when the master device changes the direction of transfer for the slave device without terminating data transfer. The following describes the procedure for performing a restart in master mode.

First, write 000 to SBI0CR2 <MST, TRX, BB> and 1 to <PIN> to relinquish the bus. At this time, the SDA0 pin maintains a High level and the SCL0 pin is relinquished so that the bus is still busy as viewed from other devices because no stop condition occurs. Then, test SBI0SR <BB> and wait until it becomes 0 to determine that the SCL0 pin has been relinquished. Next, test SBI0SR<LRB> and wait until it becomes 1 to determine that no other device is pulling the SCL0 bus line Low. After determining that the bus is free using the above steps, generate a start condition as described in 3.10.6 (2).

To satisfy the setup time condition for a restart, a software wait time is required between the bus free state being determined and a start condition being generated. The time is at least 600 ns in fast mode or at least 4.7 µs in standard mode.
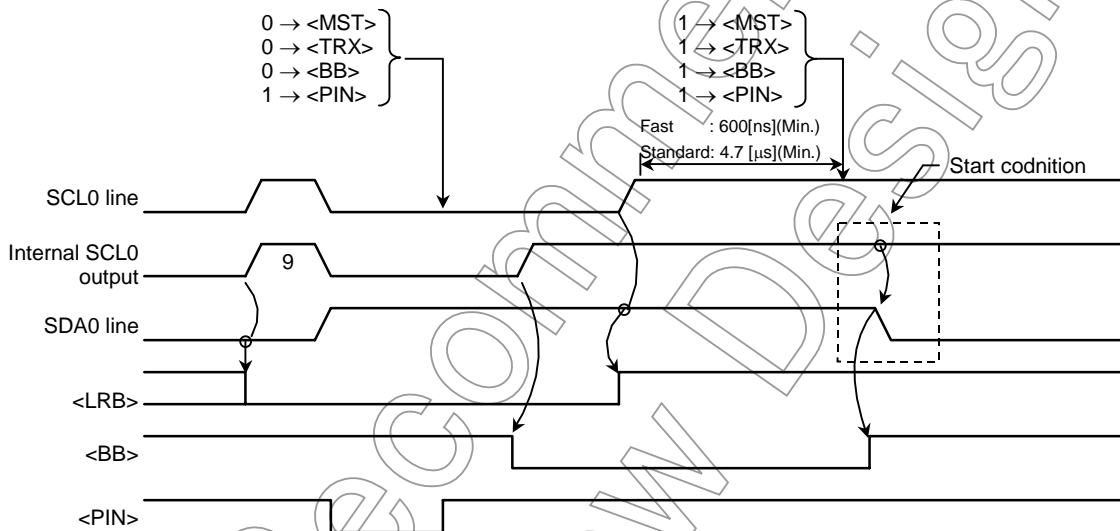
Figure 3.10.27 Timing Diagram when Restarting

3.10.7 Control in clock synchronous 8-bit SIO mode

The following registers are used to control the serial bus interface and monitor its operating state in clock synchronous 8-bit SIO mode:

Serial Bus Interface 0 Control Register 1

		7	6	5	4	3	2	1	0
SBI0CR1	Bit symbol	SIOS	SIOINH	SIOM1	SIOM0	-	SCK2	SCK1	SCK0
(0170H)	Read/Write		W			W		W	
	After Reset	0	0	0	0	1	0	0	0
Read-modify-write not allowed	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		Note2) Write 0 to this bit.	Serial clock selection		

Serial clock selection <SCK2 to 0> @ write

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)
$fscl = \dfrac{fc}{2^{n}}$ [Hz]

Transfer mode selection

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

Note1: When using SIO mode, write a 0 to this bit.

Note2: After setting the transfer mode and serial clock, write a 1 to <SIOS> to start transfer. To set the transfer mode and serial clock, first set <SIOS> to 0 and <SIOINH> to 1.

Serial Bus interface 0 Data Buffer Register

		7	6	5	4	3	2	1	0
SBI0DBR (0171H)	Bit symbol	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
Read-modify-write not allowed	Read/Write				R (receiver) / W (transfer)				
	After Reset				Undefined				

Figure 3.10.28 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/Write					W		W	W
	After Reset	-	-	-	-	0	0	0	0
Read-modify-write not allowed	Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (reserved)		(Note2)	(Note2)

Serial bus interface operation mode selection

00	Port mode (serial bus interface output disabled)
01	Clocked-Synchronous 8-bit SIO mode
10	I²C bus mode
11	(reserved)

Note1: SBI0CR2<1:0> must always be set to 00.
Note2: It is necessary to clear SBI0CR1 <BC2:0> to 000 before attempting to change the operating mode to clock 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/Write					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

Figure 3.10.29 Registers for the SIO Mode

Serial Bus Interface 0 Baud Rate Register 0

		7	6	5	4	3	2	1	0
SBI0BR0 (0174H)	Bit symbol	-	I2SBI0	-	-	-	-	-	-
	Read/Write	W	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

		7	6	5	4	3	2	1	0
SBI0BR1 (0175H)	Bit symbol	P4MON/ P4EN	-	-	-	-	-	-	-
	Read/Write	R/W							
	After Reset	0	-	-	-	-	-	-	-
	Function	Internal clock 0: Stop 1: Operate							

→ Baud rate clock control

0	Stop
1	Operate
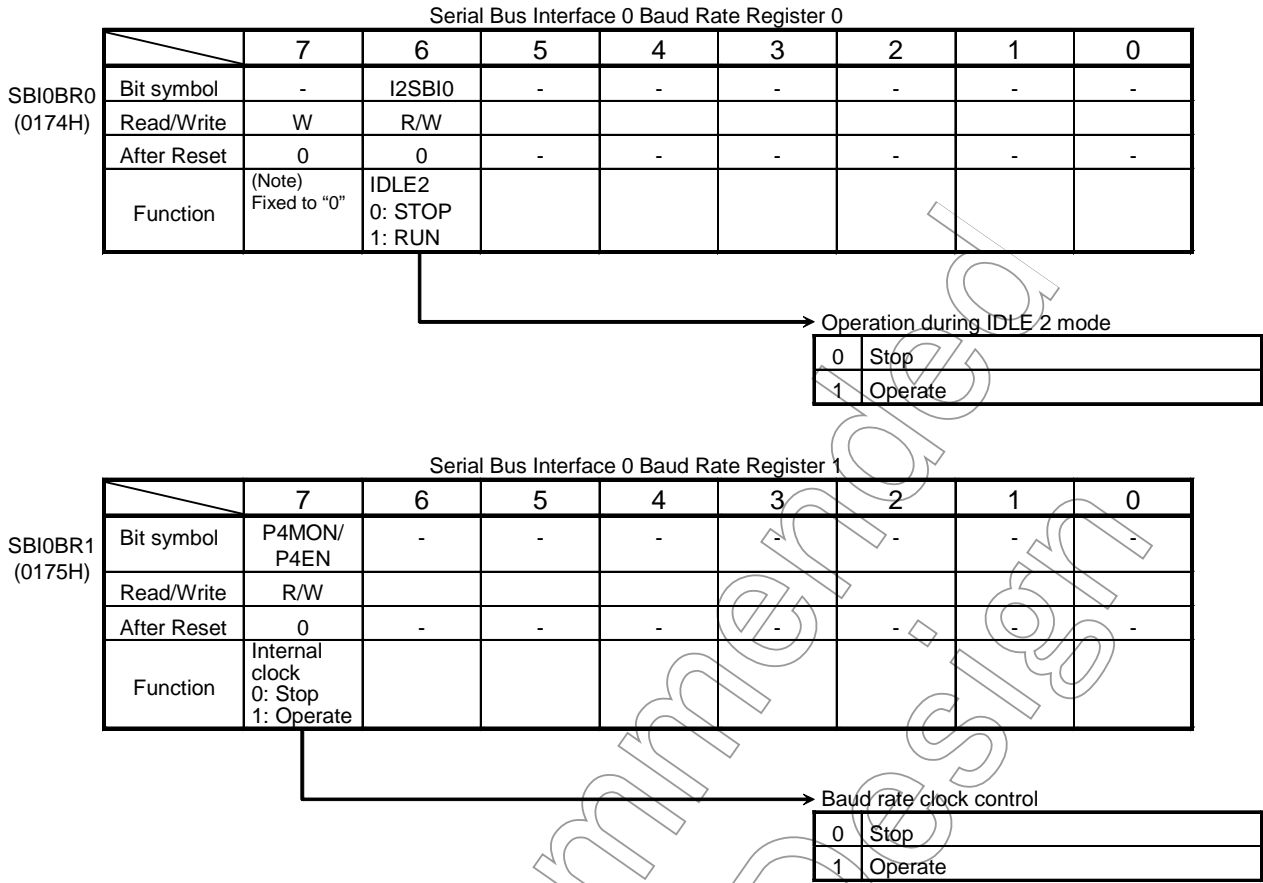
Figure 3.10.30 Registers for the SIO Mode

Serial Bus Interface 1 Control Register 1

		7	6	5	4	3	2	1	0
SBI1CR1	Bit symbol	SIOS	SIOINH	SIOM1	SIOM0	-	SCK2	SCK1	SCK0
(0178H)	Read/Write			W		W		W	
	After Reset	0	0	0	0	1	0	0	0
Read-modify-write not allowed	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		Note2) Write 0 to this bit.	Serial clock selection		

Serial clock selection <SCK2 to 0> @ write

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)
$fscl = \dfrac{fc}{2^n}$ [Hz]

Transfer mode selection

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

Note1: When using SIO mode, write a 0 to this bit.
Note2: After setting the transfer mode and serial clock, write a 1 to <SIOS> to start transfer. To set the transfer mode and serial clock, first set <SIOS> to 0 and <SIOINH> to 1.

Serial Bus interface 1 Data Buffer Register

		7	6	5	4	3	2	1	0
SBI1DBR	Bit symbol	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
(0179H)	Read/Write				R (receiver) / W (transfer)				
Read-modify-write not allowed	After Reset				Undefined				

Figure 3.10.31 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/Write					W		W	W
	After Reset	-	-	-	-	0	0	0	0
Read-modify-write not allowed	Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (reserved)		(Note2)	(Note2)

Serial bus interface operation mode selection

00	Port mode (serial bus interface output disabled)
01	Clocked-Synchronous 8-bit SIO mode
10	I²C bus mode
11	(reserved)

Note1: SBI1CR2<1:0> must always be set to 00.
Note2: It is necessary to clear SBI1CR1 <BC2:0> to 000 before attempting to change the operating mode to clock 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/Write					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

Figure 3.10.32 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/Write	W	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/Write	R/W							
	After Reset	0	-	-	-	-	-	-	-
	Function	Internal clock 0: Stop 1: Operate							

Baud rate clock control

0	Stop
1	Operate
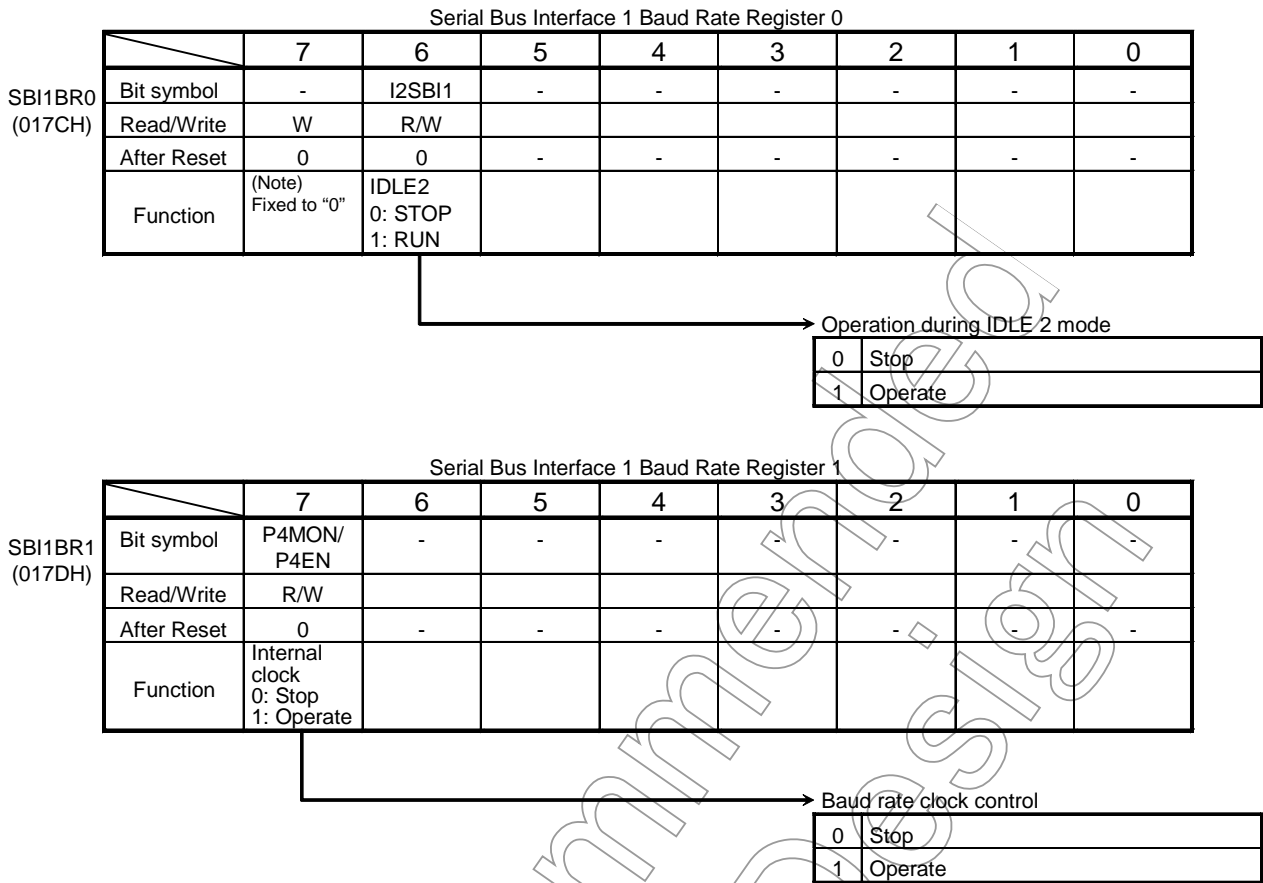
Figure 3.10.33 Registers for the SIO Mode

Serial Bus Interface 2 Control Register 1

SBI2CR1 (0180H)		7	6	5	4	3	2	1	0
	Bit symbol	SIOS	SIOINH	SIOM1	SIOM0	-	SCK2	SCK1	SCK0
	Read/Write	W				W	W		
	After Reset	0	0	0	0	1	0	0	0
Read-modify-write not allowed	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		Note2) Write 0 to this bit.	Serial clock selection		

Serial clock selection <SCK2 to 0> @ write

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 : SCK2

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

Transfer mode selection

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

Note1: When using SIO mode, write a 0 to this bit.
Note2: After setting the transfer mode and serial clock, write a 1 to <SIOS> to start transfer. To set the transfer mode and serial clock, first set <SIOS> to 0 and <SIOINH> to 1.

Serial Bus interface 2 Data Buffer Register

SBI2DBR (0181H)		7	6	5	4	3	2	1	0
Read-modify-write not allowed	Bit symbol	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
	Read/Write	R (receiver) / W (transfer)							
	After Reset	Undefined							

Figure 3.10.34 Register for the SIO Mode

Serial Bus Interface 2 Control Register 2

		7	6	5	4	3	2	1	0
SBI2CR2 (0183H)	Bit symbol	-	-	-	-	SBIM1	SBIM0	-	-
	Read/Write					W		W	W
	After Reset	-	-	-	-	0	0	0	0
Read- modify- write not allowed	Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I²C bus mode 11: (reserved)		(Note2)	(Note2)

Serial bus interface operation mode selection

00	Port mode (serial bus interface output disabled)
01	Clocked-Synchronous 8-bit SIO mode
10	I²C bus mode
11	(reserved)

Note1: SBI2CR2<1:0> must always be set to 00.
Note2: It is necessary to clear SBI2CR1 <BC2:0> to 000 before attempting to change the operating mode to clock synchronous 8-bit SIO mode.

Serial Bus Interface 2 Status Register

		7	6	5	4	3	2	1	0
SBI2SR (0183H)	bit Symbol	-	-	-	-	SIOF	SEF	-	-
	Read/Write					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

Figure 3.10.35 Registers for the SIO Mode

Serial Bus Interface 2 Baud Rate Register 0

		7	6	5	4	3	2	1	0
SBI2BR0	Bit symbol	-	I2SBI1	-	-	-	-	-	-
(0184H)	Read/Write	W	R/W						
	After Reset	0	0	-	-	-	-	-	-
Read-modify-write not allowed	Function	(Note) Fixed to "0"	IDLE2 0: STOP 1: RUN						

Operation during IDLE 2 mode

0	Stop
1	Operate

Serial Bus Interface 2 Baud Rate Register 1

		7	6	5	4	3	2	1	0
SBI2BR1	Bit symbol	P4MON/ P4EN	-	-	-	-	-	-	-
(0185H)	Read/Write	R/W							
	After Reset	0	-	-	-	-	-	-	-
	Function	Internal clock 0: Stop 1: Operate							

Baud rate clock control

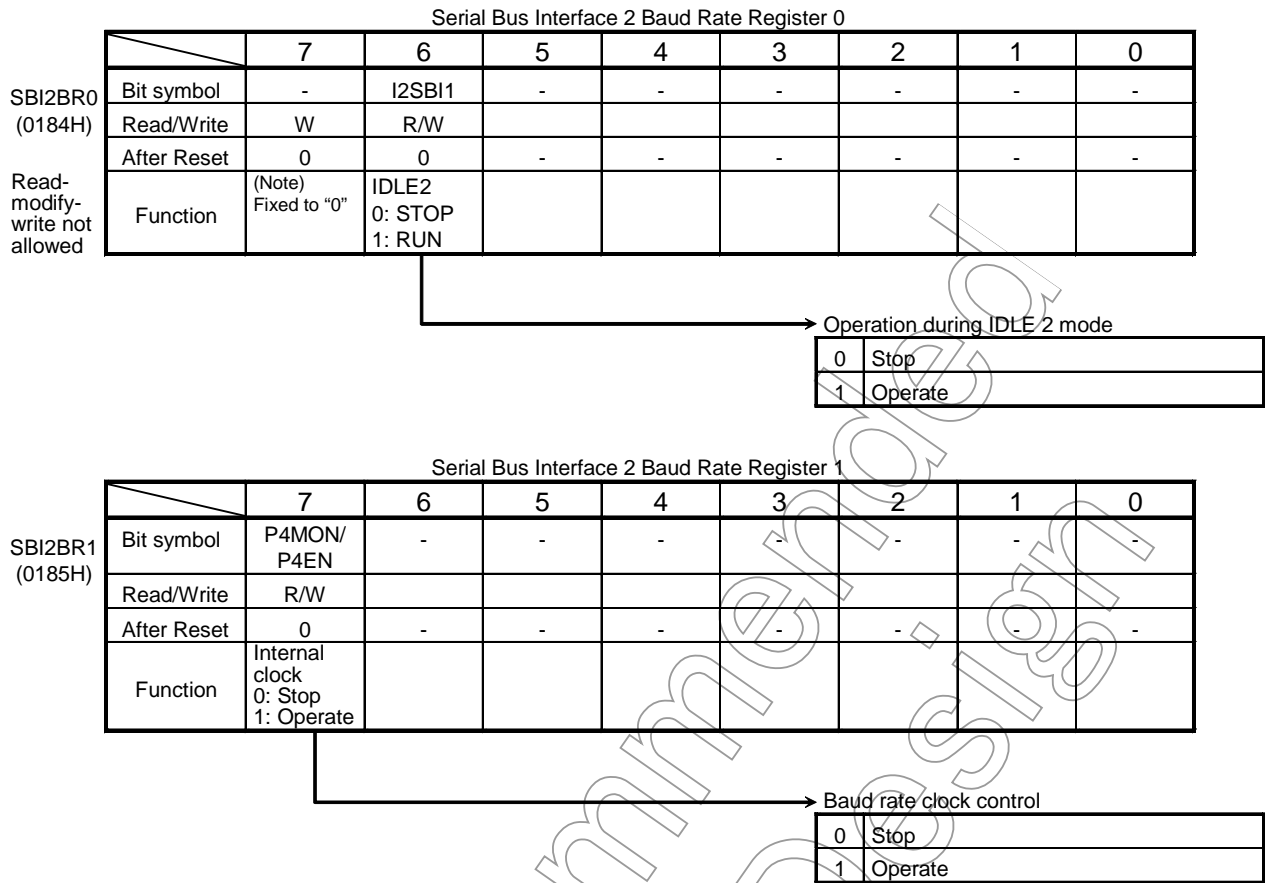| 0	Stop
1	Operate

Figure 3.10.36 Registers for the SIO Mode

(1) Serial clock

    a.   Clock source

The following clock sources can be selected using SBI0CR1 <SCK2:0>.

<u>Internal clock</u>

In internal clock mode, one of seven frequencies can be selected. The serial clock is supplied to an external device through the SCK0 pin. At the start of transfer, the SCK0 pin output is High.

If a data write (for transmission) or a data read (for reception) in the program cannot keep up with the serial clock rate, the automatic wait function stops the serial clock automatically and delays the next shift operation until the processing is completed.
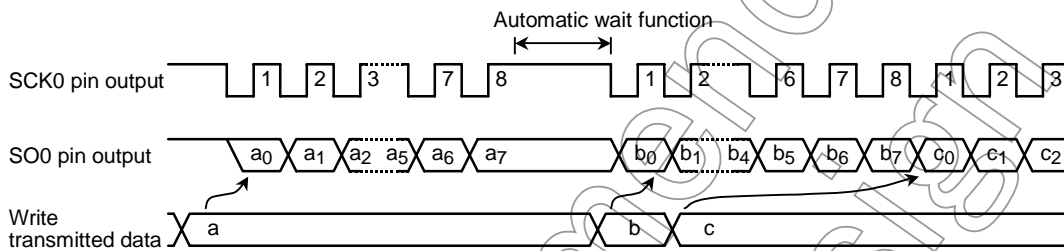
Figure 3.10.37 Automatic-wait Function

<u>External clock (SBI0CR1 <SCK2:0> = 111)</u>

In this mode, an external clock supplied through the SCK0 pin is used as the serial clock. To ensure that shift operation is performed normally, the High and Low widths of the serial clock must satisfy the following condition. The maximum transfer frequency is, therefore, 1.25 MHz (when fc = 20 MHz).

$t_{SCKL}, t_{SCKH} > 8/fc$
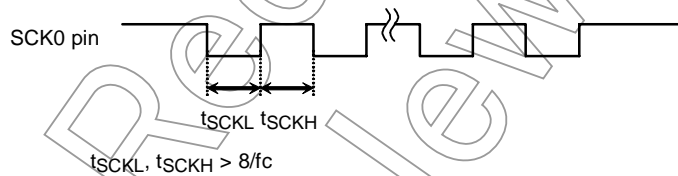
Figure 3.10.38 Maximum Data Transfer Frequency when External Clock Input

b. Shift edge

Data is transmitted using a leading-edge shift and received using a trailing-edge shift.

<u>Leading-edge shift</u>

Data is shifted on the leading edge of the serial clock (falling edge of the SCK0 pin input/output).

<u>Trailing-edge shift</u>

Data is shifted on the trailing edge of the serial clock (rising edge of the SCK0 pin input/output).

(a) Falling edge shift

(b) Rising edge shift   Note: * = Don't care
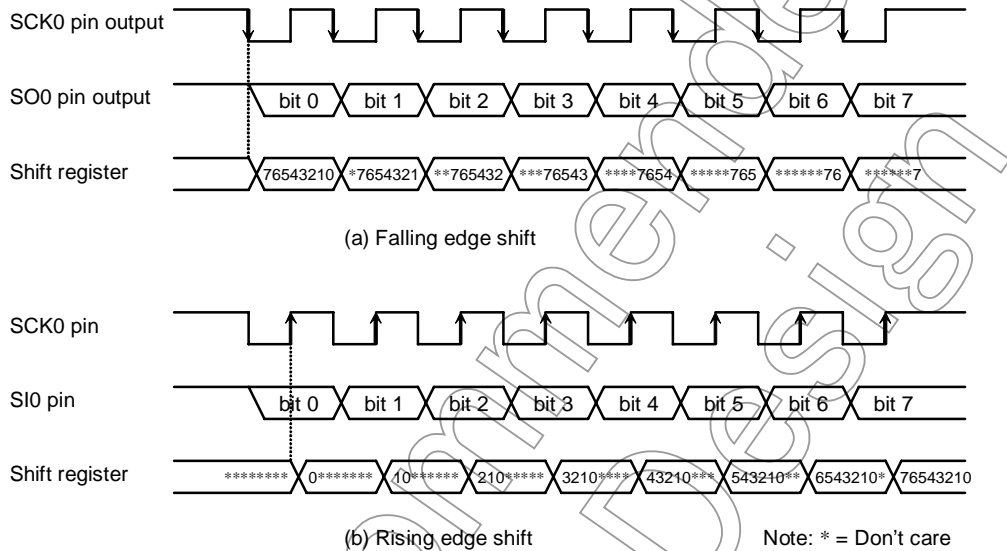
Figure 3.10.39 Shift Edge

(2) Transfer modes

The SBI0CR1 <SIOM1:0> bits select the transfer mode: transmit, receive, or transmit/receive.

a.   8-bit transmit mode

After specifying transmit mode in the control register, write transmit data to SBI0DBR.

Then, setting SBI0CR1 <SIOS> to 1 causes transmission to start. The transmit data is moved from SBI0DBR to the shift register and then, in synchronization with the serial clock, output through the SO0 pin in an LSB-first manner. Once the transmit data has been moved to the shift register, SBI0DBR becomes empty, thus causing an INTSBE0 interrupt (buffer empty) to occur that requests next transmit data.

In internal clock operation, if next data is not set after all of 8-bit data has been transmitted, the serial clock is stopped for automatic wait. Writing next transmit data terminates automatic wait.

In external clock operation, data must be written to SBI0DBR before shift operation for next data starts. The transfer rate is, therefore, determined from the maximum delay between an interrupt request being issued and data being written to SBI0DBR in the interrupt handling routine.

At the beginning of transmission, the same value as the last bit of the data transmitted last is output between SBI0SR <SIOF> being set to 1 and the falling edge of SCK0.

To terminate transmission, either set SBI0CR1<SIOS> to 0 or SBI0CR1<SIOINH> to 1 in the INTSBE0 interrupt handling routine. If SBI0CR1<SIOS> is cleared, transmission is terminated once all data has been output. The program can determine the termination of transmission using SBI0SR <SIOF>. SBI0SR <SIOF> is set to 0 upon the termination of transmission. Setting SBI0CR1<SIOINH> to 1 causes transmission to be aborted immediately and SBI0CR1<SIOF> to be cleared to 0.

In external clock operation, SBI0CR1 <SIOS> must be cleared to 0 before shift operation for next transmit data starts. If SBI0CR1<SIOS> is not cleared before shift-out operation starts, the serial bus interface transmits dummy data and then stops.
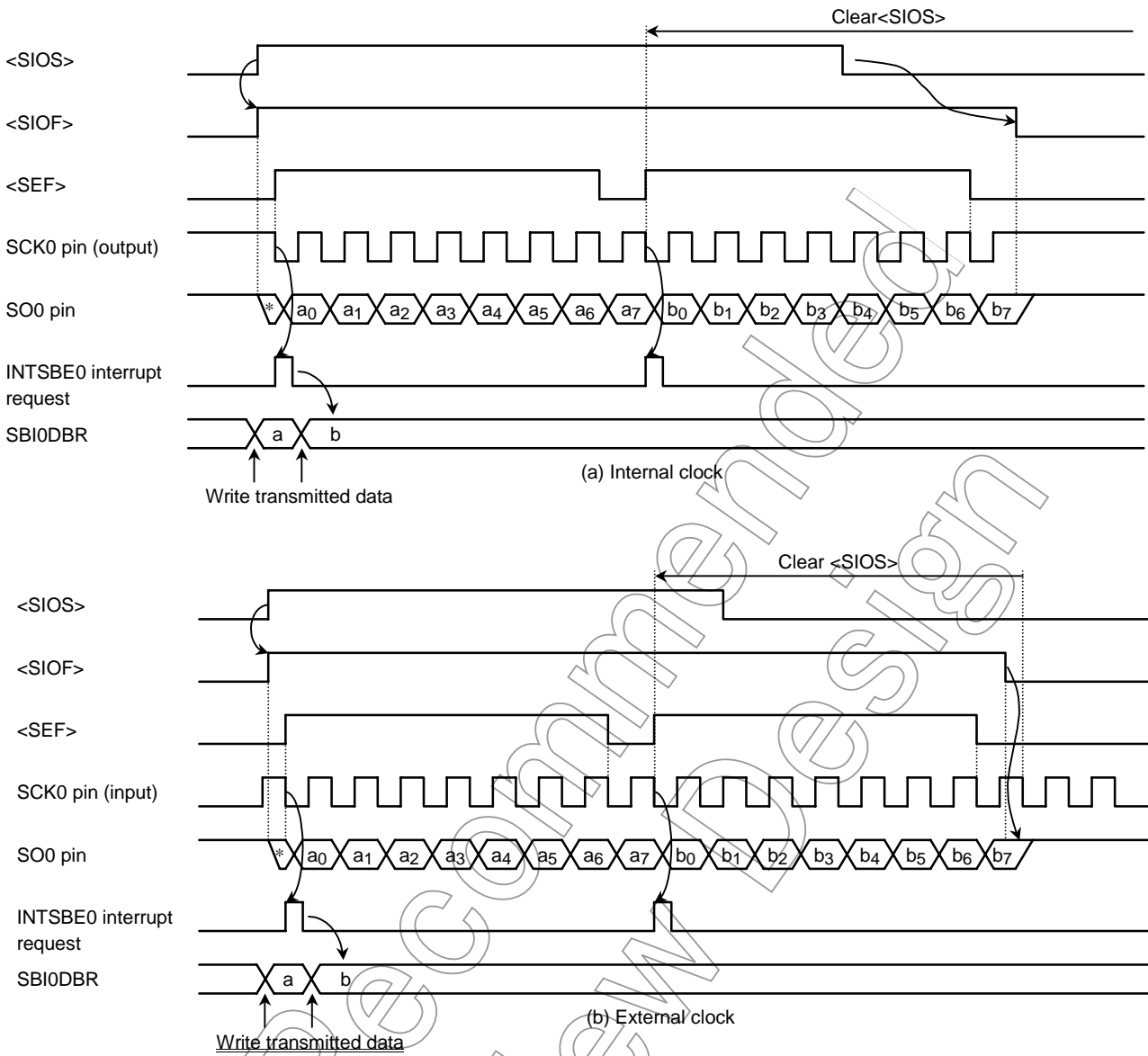
Figure 3.10.40 Transfer Mode

Example: Specifying the termination of transmission for <SIO> (when using an external clock)

```
STEST1: BIT     2, (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
```
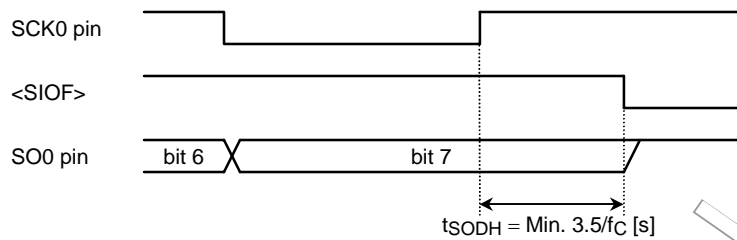
Figure 3.10.41 Transmitted Data Hold Time at End of Transmission

b.　8-bit receive mode

After specifying receive mode in the control register, write a 1 to SBI0CR1 <SIOS> to enable reception. In synchronization with the serial clock, data from the SI0 pin is captured into the shift register in an LSB-first manner. Once 8-bit data has been captured, the received data is moved from the shift register to SBI0DBR, thus causing an INTSBE0 interrupt (buffer full) to occur that requests reading the received data. The interrupt handling routine should read the received data from SBI0DBR.

In internal clock operation, the automatic wait function stops the serial clock until the received data is read from SBI0DBR.

In external clock operation, read the received data before a next serial clock is input because shift operation is synchronized with an externally supplied clock. If the received data is not read, subsequently input received data will be cancelled. The maximum transfer rate with an external clock is determined from the maximum delay between an interrupt request being issued and the received data being read.

To terminate reception, either set SBI0CR1<SIOS> to 0 or SBI0CR1<SIOINH> to 1 in the INTSBE0 interrupt handling routine. If SBI0CR1<SIOS> is cleared, reception is terminated once all bits of the data have been received and written to SBI0DBR. The program can determine the termination of reception using SBI0SR <SIOF>. <SIOF> is cleared to 0 upon the termination of reception. After determining that reception has been terminated, read the last received data. Setting SBI0CR1<SIOINH> to 1 causes reception to be aborted immediately and SBI0SR<SIOF> to be cleared to 0 (the received data becomes invalid and need not be read).

Note: When the transfer mode is switched, the contents of SBI0DBR are not maintained. If it is necessary to switch the transfer mode, first write a 0 to <SIOS> to terminate transfer and read the last received data.

Figure 3.10.42 Receiver Mode (example: Internal clock)

c.　8-bit transmit/receive mode

After specifying transmit/receive mode in the control register, write transmit data to SBI0DBR. Then, setting SBI0CR1 <SIOS> to 1 enables transmission and reception. The transmit data is output through the SO0 pin on the rising edge of the serial clock, in an LSB-first manner, while the received data is captured from the SI0 pin on the falling edge of the clock. Once 8-bit data has been captured, the received data is moved from the shift register to SBI0DBR, thus causing an INTSBE0 interrupt request to be issued. The interrupt handling routine reads the received data from the data buffer register and then writes transmit data. Ensure that the received data is read before transmit data is written to SBI0DBR because SBI0DBR is shared for transmission and reception.

In internal clock operation, automatic wait is performed between the received data being read and next transmit data being written.

In external clock operation, it is necessary to read the received data and then write next transmit data before next shift operation starts because shift operation is synchronized with an externally supplied serial clock. The maximum transfer rate with an external clock is determined from the maximum delay between an interrupt request being issued and the received data being read, followed by transmit data being written.

At the beginning of transmission, the same value as the last bit of the data transmitted last is output between SBI0SR<SIOF> being set to 1 and the falling edge of SCK0.

To terminate transmission/reception, either set SBI0CR1<SIOS> to 0 or SBI0CR1<SIOINH> to 1 in the INTSBE0 interrupt handling routine. If SBI0CR1<SIOS> is cleared, transmission/reception is terminated once all bits of the data have been received and written to SBI0DBR. The program can determine the termination of transmission/reception using SBI0SR <SIOF>. SBI0SR <SIOF> is cleared to 0 upon the termination of transmission/reception. Setting SBI0CR1<SIOINH> to 1 causes transmission/reception to be aborted immediately and SBI0SR<SIOF> to be cleared to 0.

Note:　When the transfer mode is switched, the contents of SBI0DBR are not maintained. If it is necessary to switch the transfer mode, first write a 0 to <SIOS> to terminate transfer and read the last received data.
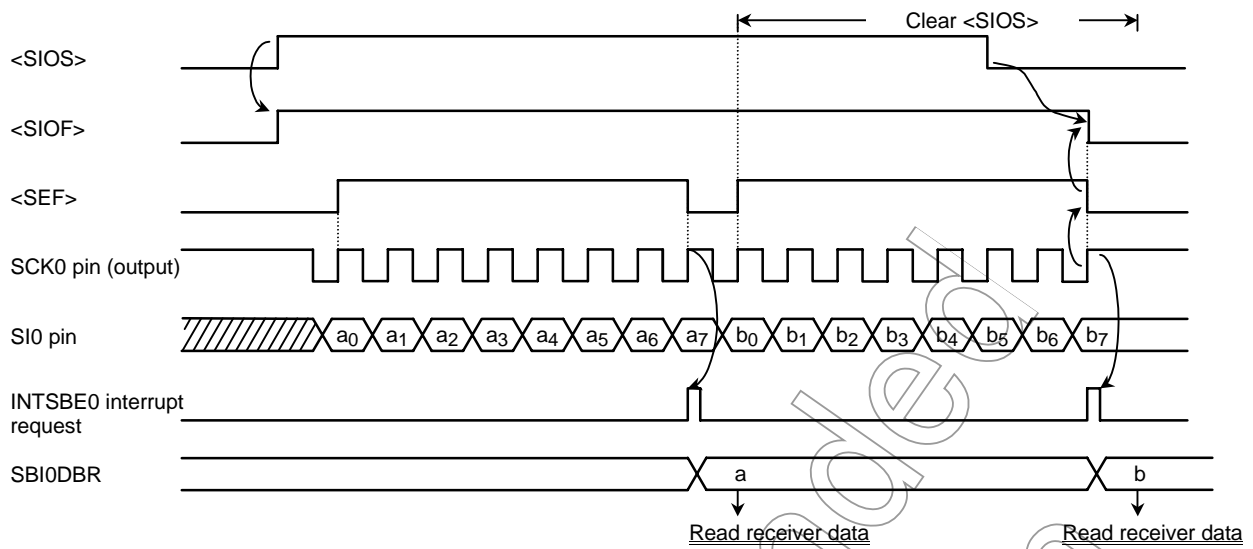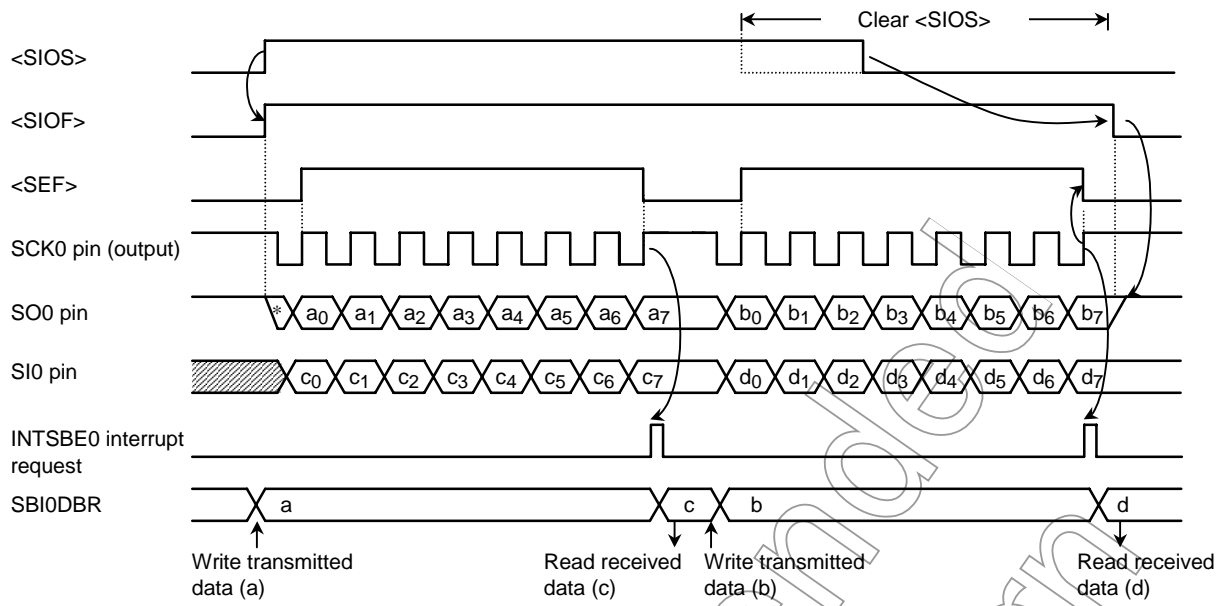
Figure 3.10.43 Transmit/Received Mode (Example : Internal clock)

$$t_{SODH} = \text{Min. } 4/f_C \text{ [s]}$$
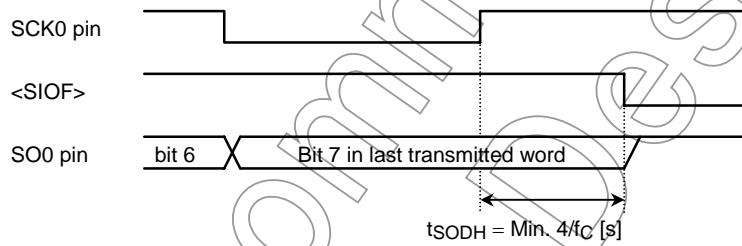
Figure 3.10.44 Transmitted Data Hold Time at End of Transmit/Receive

3.11 Serial Expansion Interface (SEI)

3.11.1 Overview

The serial expansion interface (SEI) is one of the interfaces built into the TMP92CD54I and can connect to peripheral devices using a full-duplex synchronous communication protocol. It also supports micro DMA mode, in which it transfers data using micro DMA.

The TMP92CD54I contains a single SEI channel (SEI0).

(1) Features

- The master outputs a shift clock only when data is being transferred.

- The clock polarity and phase are programmable.

- The data length is eight bits.

- MSB- or LSB-first transfer can be selected.

- Supports transfer using micro DMA (micro DMA mode).

- The master can select one of the following three transfer rates:
  4 Mbps, 2 Mbps, and 500 kbps (when fc = 20 MHz)

- The error detection circuit supports the following functions:

  a. Write collision detection: If the shift register is written during a transfer.

  b. Overflow detection: If new data is received when the transfer completion flag is set to 1 (slave mode only).

  c. Mode fault detection: If the input to the $\overline{SS}$ pin is driven Low in master mode (driver output turned off immediately).
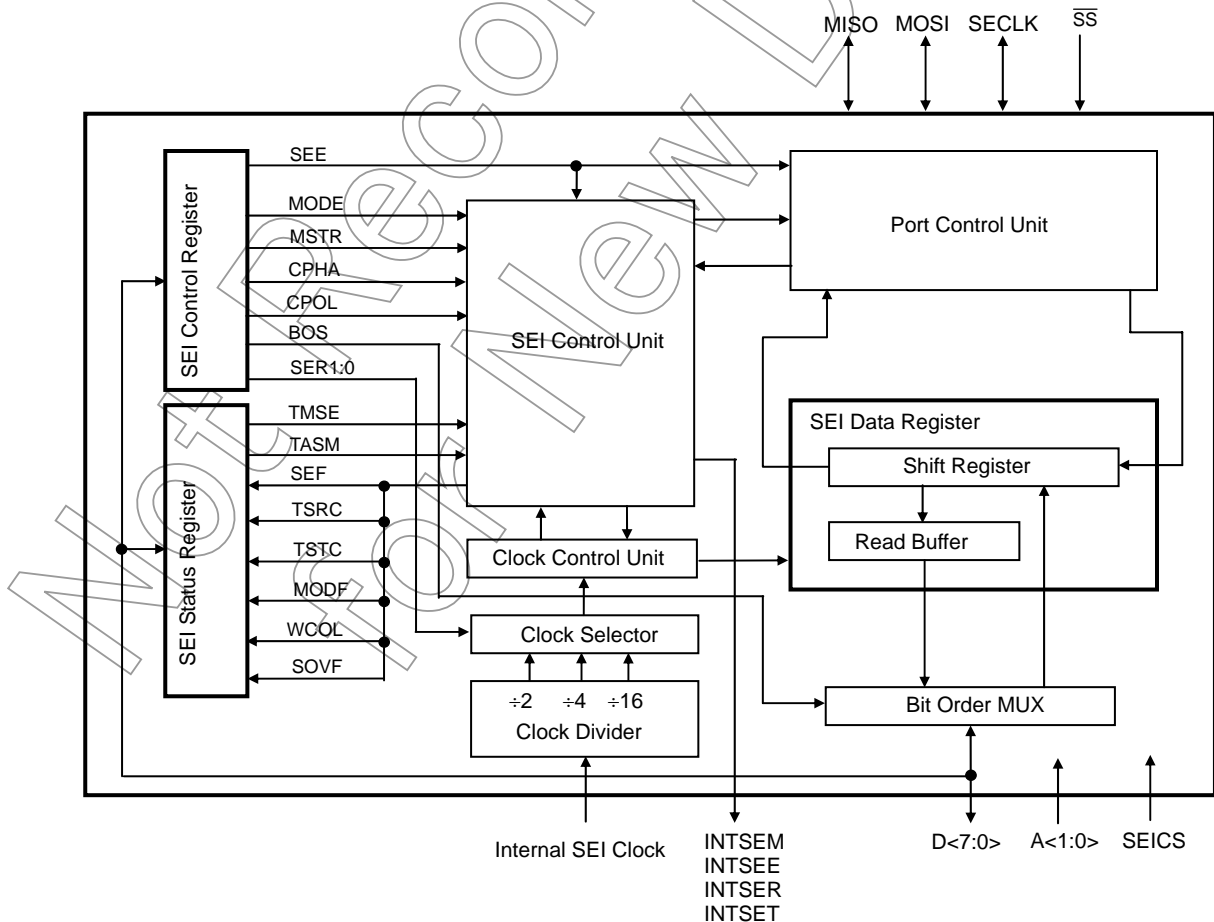
Figure 3.11.1 SEI Block Diagram

Table 3.11.1 Pin Function of SEI Channels

SEI
$\overline{SS}$ (PM0)
MOSI (PM1)
MISO (PM2)
SECLK (PM3)

3.11.2 SEI operation

During an SEI transfer, data transmission (serial shift-out) and reception (serial shift-in) are performed simultaneously. The SEI clock (SECLK) provides synchronization for shifting and sampling information on the two serial data lines (MOSI and MISO). The slave select line ($\overline{SS}$) selects an individual slave device. Only the selected slave device can do the SEI transfer using the SEI bus.

（1） Controlling the SEI clock phase and polarity

Four types of SEI clock can be selected by two bits in the SEI control register (SECR) that control the phase and polarity of the clock. The clock polarity is controlled with the <CPOL> bit, which selects either an active-high or active-low clock. The clock phase is controlled with the <CPHA> bit, which selects one of two different transfer formats. The clock phase and polarity must be the same between the master device and the slave device it communicates with.

（2） SEI data and clock timing

The SEI has programmable clock timing and data, which support most synchronous serial peripheral devices. See "3.11.4 SEI Transfer Format."

3.11.3　SEI pin functions

The SEI has four input and output pins for data transfer. The function of each pin depends on the SEI device mode (master or slave).

(1)　SECLK pin

The SECLK pin functions as an output when the SEI is set to master mode or as an input when the SEI is set to slave mode.

When the SEI is a master, the SECLK signal is supplied from its internal SEI clock generator. Once the master has started a transfer, eight clock cycles are automatically supplied on the SECLK pin.

When the SEI is a slave, the SECLK pin functions as an input and the SECLK signal supplied from the master synchronizes data transfer between the master and slave. If the slave select pin, $\overline{SS}$, is driven High, the slave device ignores the SECLK signal.

Both the master and slave devices shift data on the rising or falling edge of the SECLK signal and sample data on the opposite edge. The edge polarity depends on the SEI transfer protocol.

(2)　MISO and MOSI pins

The MISO and MOSI pins are used to transmit and receive serial data.

When the SEI is set to a master, the MISO turns into the input signal and the MOSI turns into the output signal.

When the SEI is set to a slave, the functions of the pins are reversed.

In SEI system, all SECLK pins are interconnected, all MOSI pins are interconnected, and all MISO pins are interconnected. See Figure 3.11.5. A single SEI device is set as a master while all other SEI devices on the SEI bus are set to slaves. The master device transmits the transfer clock and data from its SECLK and MOSI pins to the SECLK and MOSI pins of slave devices, respectively. The single selected slave device transmits data from its MISO pin to the master device's MISO pin.

The SECLK, MISO, and MOSI pins can also be programmatically set to open-drain, using the corresponding bits in the port M open-drain enable register, PMODE.

(3)　$\overline{SS}$ pin

The $\overline{SS}$ pin functions differently depending on whether the SEI is set to a master or slave.

A slave device uses the pin to enable SEI slave transmission/reception. If its $\overline{SS}$ pin is High (not active), the slave device ignores the SECLK clock and places its MISO output pin in high-impedance state.

A master device uses the $\overline{SS}$ pin to detect an SEI error. If its $\overline{SS}$ pin is driven Low when the SEI is a master, it indicates that another device on the SEI bus is attempting to become a master. The master device thus detects an error and immediately releases the SEI bus to prevent damage from a driver collision. Such an error is called a mode fault. The <MODE> bit in the SECR register enables or disables detection for a mode fault. When the <MODE> bit is set to 0, the $\overline{SS}$ pin is enabled as a mode fault detection input. When the <MODE> bit is set to 1, mode fault detection using the $\overline{SS}$ pin is disabled.

3.11.4 SEI transfer format

The transfer format is determined by the <CPHA> and <CPOL> bit settings in the SECR register. The <CPHA> bit selects one of two different transfer protocols.

（1）Transfer format when <CPHA> = 0

Figure 3.11.2 shows the transfer format when <CPHA> = 0.
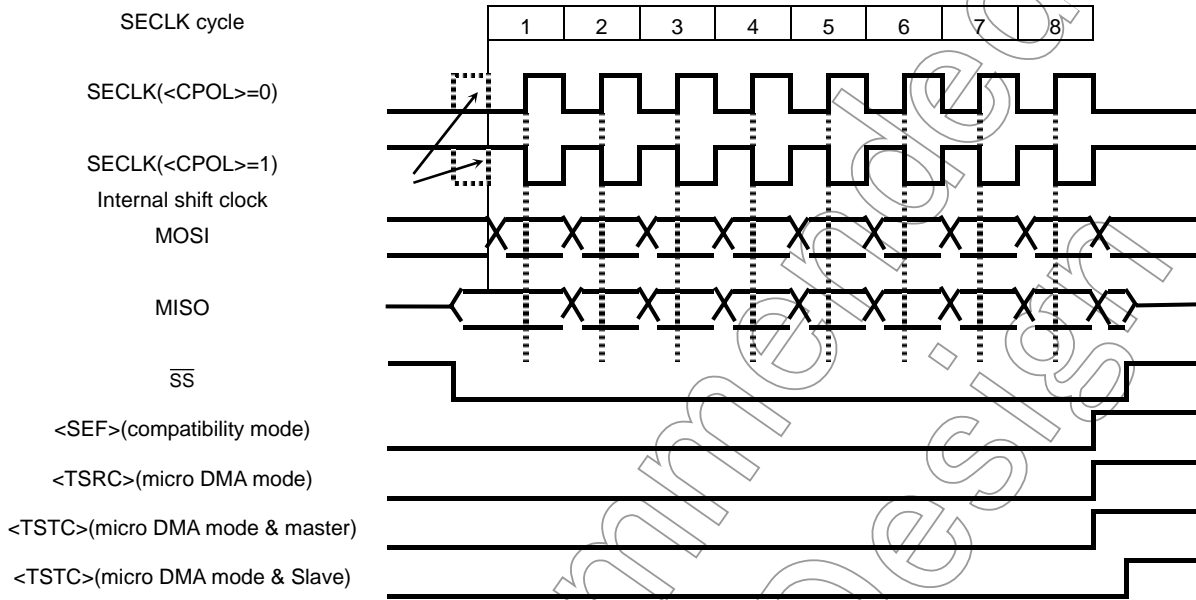
Figure 3.11.2 Transfer Format when <CPHA> = 0

Table 3.11.2 Data Timing when <CPHA> = 0

<CPHA>=0

	No communication (idle) SECLK level	Data shift	Data sampling
<CPOL>=0	L	Shift clock falling edge	Shift clock rising edge
<CPOL>=1	H	Shift clock rising edge	Shift clock falling edge

In master mode, writing new data to the SEDR register causes a data transfer to start. Data on the MOSI pin is switched a half clock cycle before the shift clock starts operating. The SECR<BOS> bit specifies whether data will be shifted out in MSB- or LSB-first manner. After the last shift cycle, the SESR <SEF> flag is set to 1 in compatibility mode or the <TSRC> and SESR<TSTC> flags are set to 1 in micro DMA mode.

In slave mode, a write to the SEDR register is prohibited while the $\overline{SS}$ pin is Low. Writing data during that period results in a write collision, causing the <WCOL> flag in the SESR register to be set to 1. Therefore, if the SESR<SEF> or SESR <TSRC> flag is set to 1 upon the completion of data transfer, the program must wait until the $\overline{SS}$ pin is driven back High before attempting to write next data to the SEDR register. In slave mode, if micro DMA is used to transfer data to the SEDR register, the SESR<TSTC> flag is not set until the $\overline{SS}$ pin is driven High.

(2) Transfer format when <CPHA> = 1

Figure 3.11.3 shows the transfer format when <CPHA> = 1.
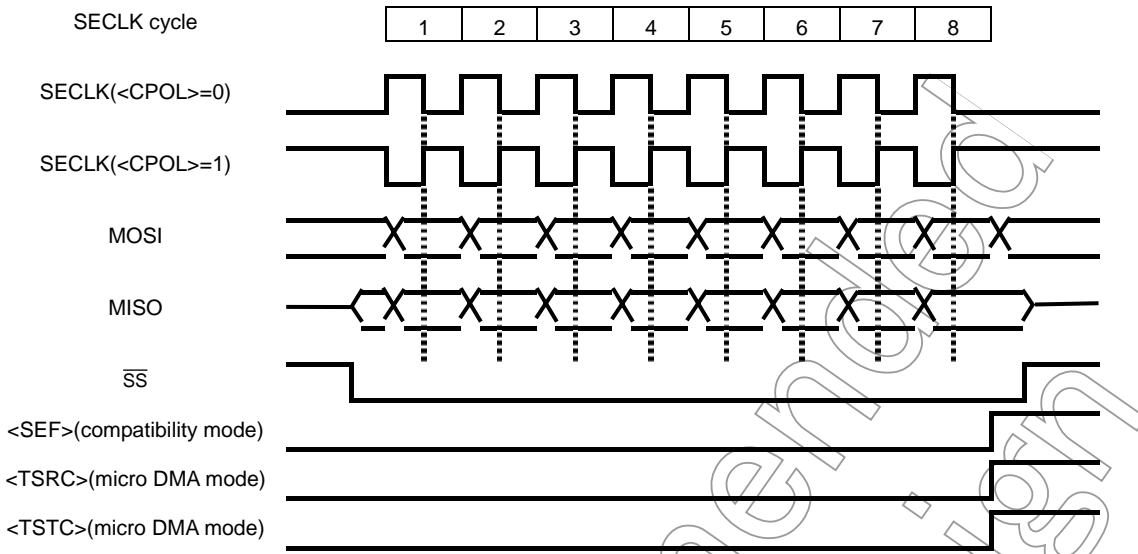
Figure 3.11.3 Transfer Format when <CPHA> = 1

Table 3.11.3 Data Timing when <CPHA> = 1

<CPHA>=1

	No communication (idle) SECLK level	Data shift	Data sampling
<CPOL>=0	L	Shift clock rising edge	Shift clock falling edge
<CPOL>=1	H	Shift clock falling edge	Shift clock rising edge

In master mode, writing new data to the SEDR register causes a data transfer to start. The data on the MOSI pin is switched on the first edge of the shift clock. The SECR<BOS> bit specifies whether data will be shifted out in MSB- or LSB-first manner.

In slave mode, unlike the format used when SECR<CPHA> = 0, a write to the SEDR register is allowed even when the $\overline{SS}$ pin is Low. In both master and slave modes, after the last shift cycle, the SESR <SEF> flag is set to 1 in compatibility mode or the SESR<TSRC> and SESR<TSTC> flags are simultaneously set to 1 in micro DMA mode.

Writing to the SEDR register during a data transfer results in a write collision. Do not write data to SEDR before the SESR<SEF> flag or the SESR<TSRC> and <TSTC> flags are set to 1.

3.11.5 Functional description

Figure 3.11.4 shows connection between master and slave on SEI system.

When the master device transmits data from its MOSI pin to the slave device's MOSI pin, the slave device transmits data from its MISO pin to the master device's MISO pin.

It indicates that data output and input are synchronized using the same clock signal in full-duplex communication. Upon the completion of transfer, the transmit data in the 8-bit shift register is replaced with the received data.
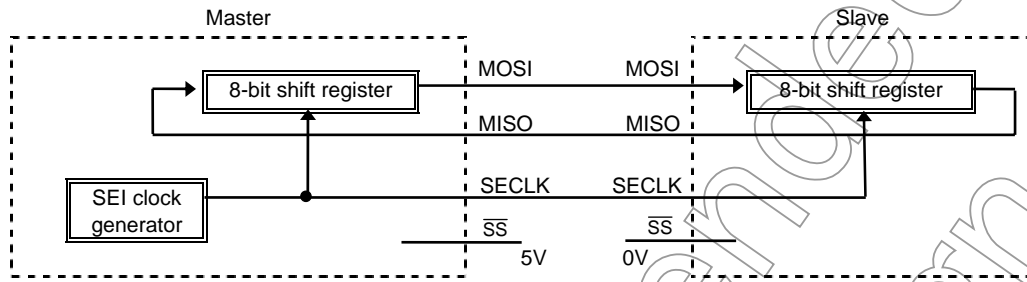
Figure 3.11.4 Connection between Master and Slave in SEI

Figure 3.11.5 shows an example SEI system configuration.

SEI output ports can be programmatically set to open-drain output. Multiple devices can thus be connected.

PORTn, n': any output ports
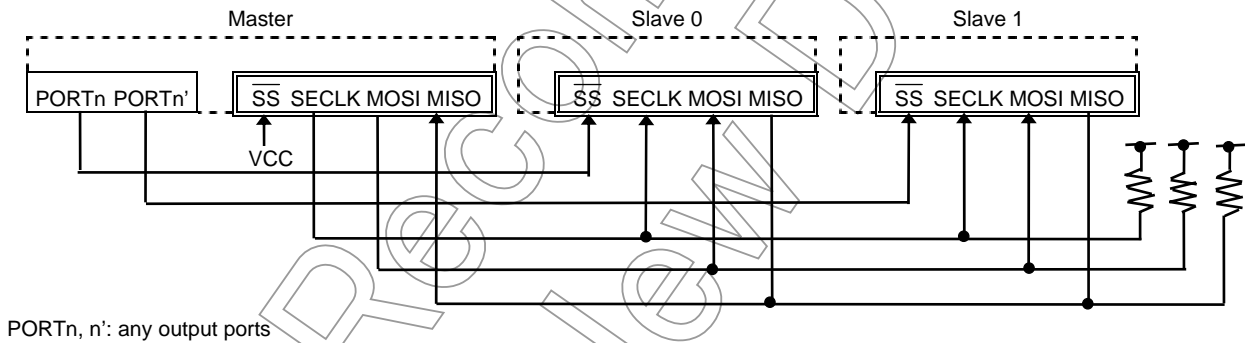
Figure 3.11.5 Configuration of SEI System (Comprised of One Master and Two Slaves)

3.11.6 Operating modes

The SEI supports two different operating modes, compatibility mode and micro DMA mode, and operates in the selected mode. These modes differ in how a flag is cleared and an interrupt is generated as well as whether micro DMA can be used.

Table 3.11.4 Differences between the Two Operation Modes

	compatibility mode	micro DMA mode
error flag clearing	Reading a register with the Status flag set, followed by SECR register or reading or writing SEDR register	Writing a "1" to the status register
transfer status flag clearing	Reading a register with the Status flag set, followed by an reading or writing 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 usage	No	yes

The SEI operating mode can be switched using SESR<TMSE> when the SEI is disabled (SECR<SEE> = 0).

3.11.7   SEI registers

The SEI can be configured using the SEI control register (SECR), SEI status register (SESR), and SEI data register (SEDR).

Note: When reading SEI registers (SECR, SESR, and SEDR) after writing to them, there must be an interval of at least four states between the write and read. The program should take that interval into account.

Programming example:

```
LD    (SEDR), data1      ;   Write to SEDR
NOP                      ;
NOP                      ;   NOP or other instruction not reading SEI register
LD    A,(SESR)           ;   Read from SESR
LD    (SESR), data2      ;   Write to SEDR
NOP                      ;
NOP                      ;   NOP or other instruction not reading SEI register
LD    A,(SESR)           ;   Read from SESR
```

(1)   SEI control register (SECR)

SEI Control Register

		7	6	5	4	3	2	1	0
SECR	bit Symbol	MODE	SEE	BOS	MSTR	CPOL	CPHA	SER1	SER0
(0060H)	Read/Write	W	R/W						
	After reset	0	0	0	0	0	1	1	1
Read-modify-write not allowed	Function	Mode fault detection 0:enabled 1:disabled	SEI operation 0:stopped 1:operating	Bit order selection 0:MSB first 1:LSB first	Mode selection 0:slave 1:master	Clock polarity selection see figure 3.11.2, 3.11.3	Clock Phase selection see figure 3.11.2, 3.11.3	SEI transfer rate selection 00: Reserved 01: divide-by- 2 10: divide-by- 4 11: divide-by-16	

Figure 3.11.6 SEI Registers (SECR)

<MODE>: Mode fault detection enable
0: Enables mode fault detection.
1: Disables mode fault detection.
This bit is valid only in master mode and invalid in slave mode.

<SEE>: SEI function enable
0: Disables the SEI function. To switch between micro DMA mode and compatibility mode, first disable the SEI function. Ensure that data transfer has been completed before attempting to disable the SEI function.
Also, when using the HALT instruction to enter IDLE1, IDLE3, or STOP mode, first disable the SEI function.
1: Enables the SEI function. To use the SEI, first set the relevant ports to SEI pins.

<BOS>: Bit order selection
The <BOS> bit selects whether data will be transmitted in MSB-first or LSB-first manner.
0: Transmits the MSB (bit 7) of the SEDR register first.
1: Transmits the LSB (bit 0) of the SEDR register first.

<MSTR>: Master/slave mode selection
0: Sets the SEI to slave.
1: Sets the SEI to master.

<CPOL>: Clock polarity selection

        0: Selects an active-high clock. The SECLK clock is Low when communication is not performed.

        1: Selects an active-low clock. The SECLK clock is High when communication is not performed.

        See Figure 3.11.2 and Figure 3.11.3.

<CPHA>: Clock phase selection

        The <CPHA> bit selects one of two different transfer formats.

        See Figure 3.11.2 and Figure 3.11.3.

<SER1:0>: SEI bit rate selection

        The following table shows the relationship between the transfer bit rate and the settings of the <SER1> and <SER0> bits when the SEI operates as the master. When the SEI operates as a slave, the serial clock is supplied from the master and the settings of the <SER1> and <SER0> bits are ignored.

Table 3.11.5 SEI Transfer Bit Rate

<SER1>	<SER0>	Divide-by-rate of internal SEI clock	Transfer rate (@ fc = 20 MHz)
0	0	Don't use this setting.	
0	1	4	4 Mbps
1	0	8	2 Mbps
1	1	32	500 Kbps

Note: internal SEI clock = 2/5 × fc

(2) SEI status register (SESR)

SEI Status Register (Compatibility Mode)

		7	6	5	4	3	2	1	0
SESR (0061H)	bit Symbol	SEF	WCOL	SOVF	MODF	-	-	-	TMSE
	Read/Write	R							R/W
	After reset	0	0	0	0	-	-	-	0
Compatibility mode	Function	SEI transfer complete flag 1:transfer completed	Write collision flag 1:write collided	Overflow flag (slave) 1:overflow occurred	Mode fault flag (master) 1:fault occurred				SEI mode select 0:compatibility mode 1:micro DMA mode

SEI Status Register (Micro DMA Mode)

		7	6	5	4	3	2	1	0
SESR (0061H)	bit Symbol	-	WCOL	SOVF	MODF	TSRC	TSTC	TASM	TMSE
	Read/Write		R/C (Note)					R/W	
	After reset	-	0	0	0	0	0	0	0
Micro DMA mode Read-modify-write not allowed	Function		Write collision flag 1:write collided	Overflow flag (slave) 1:overflow occurred	Mode fault flag (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 mode select 0:compatibility mode 1:micro DMA mode

Note: R/C indicates that read access and clear (by writing a 1) from the CPU are allowed.

Figure 3.11.7 SEI Registers (SESR)

<SEF>: Transfer completion flag
Compatibility mode:
The <SEF> flag is automatically set to 1 upon the completion of data transfer. When the <SEF> flag is set to 1, reading the SESR register and reading or writing to the SEDR register causes the <SEF> flag to be automatically cleared to 0.
Micro DMA mode:
The flag value is undefined when read. A write to the flag is invalid.

<WCOL>: Write collision error flag
Compatibility mode:
The <WCOL> flag is automatically set to 1 when the SEDR register is written during data transfer. A write to the SEDR register is invalid during data transfer. When the <WCOL> flag is set to 1, reading the SESR register and reading or writing to the SEDR register causes the <WCOL> flag to be automatically cleared to 0. No interrupt occurs when the <WCOL> flag is set.
Micro DMA mode:
The <WCOL> flag is automatically set to 1 when the SEDR register is written during data transfer. A write to the SEDR register is invalid during data transfer. The <WCOL> flag is cleared to 0 only by writing a 1 to the <WCOL> bit. A write of 0 is invalid. If the <WCOL> flag changes its state from 0 to 1 when the <TASM> bit is 0 in slave mode, an INTSEE interrupt pulse is generated.

<SOVF>: Overflow error flag
Master mode:
The flag value is undefined when read. A write to the flag is invalid.
Slave mode:
Compatibility mode:
The <SOVF> flag is automatically set to 1 upon the completion of receiving next data when the <SEF> flag is set to 1. When the <SOVF> flag is set to 1, reading the SESR register and reading or writing to the SEDR register causes the <SOVF> flag to be automatically cleared to 0. The <SOVF> flag is also cleared when the operating mode is switched to master mode. In compatibility mode, no interrupt occurs when the <SOVF> flag is set.
Micro DMA mode:
The <SOVF> flag is automatically set to 1 upon the completion of receiving next data when the <TSRC> flag is set to 1. The <SOVF> flag is cleared to 0 only by writing a 1 to the <SOVF> bit. A write of 0 is invalid. If the <SOVF> flag changes its state from 0 to 1 when the <TASM> bit is 0, an INTSEE interrupt pulse is generated.

<MODF>: Mode fault error flag
Master mode:
Compatibility mode:
The <MODF> flag is set to 1 when the $\overline{SS}$ pin is driven Low. At that time, the SEI operates as follows:
1. Disable the SEI output pin driver, thus placing the output pin in high-impedance state.
2. Clear the <MSTR> bit in the SECR register to 0.
3. Forcibly clear the <SEE> bit in the SECR register to 0, thus disabling the SEI system.
4. Generate an INTSEM interrupt pulse.
When the <MODF> flag is set to 1, reading the SESR register and writing to the SEDR register causes the <MODF> flag to be automatically cleared to 0.
Micro DMA mode:
Operation is the same as that in compatibility mode, except how the <MODF> flag is cleared.
The <MODF> flag is cleared to 0 only by writing a 1 to the <MODF> bit. A write of 0 is invalid.
Slave mode:
The flag value is undefined when read. A write to the flag is invalid.

<TSRC>: Receive completion flag
Compatibility mode:
The flag value is undefined when read. A write to the flag is invalid.
Micro DMA mode:
Once eight clock cycles have been shifted onto the SECLK pin, reception is completed and the <TSRC> flag is set to 1. The <TSRC> flag is cleared to 0 by reading the SEDR register, switching

to compatibility mode, or writing a 1 to the <TSRC> bit. A write of 0 to this flag is invalid. An INTSER interrupt pulse is generated when the <TSRC> flag is set.

<TSTC>: Transmit completion flag
    Compatibility mode:
    The flag value is undefined when read. A write to the flag is invalid.
    Micro DMA mode:
    The <TSTC> flag is set upon the completion of transmitting a single byte of data, but the timing is different depending on the transfer format and whether the device is a master or slave. See Figure 3.11.2 and Figure 3.11.3. The <TSTC> flag is cleared to 0 by writing to the SEDR register, switching to compatibility mode, or writing a 1 to the <TSTC> bit. A write of 0 to this flag is invalid. An INTSET interrupt pulse is generated when the <TSTC> flag is set.

<TASM>: Automatic shift mode (master) / INTSEE interrupt mask (slave)
    Automatic shift mode makes micro DMA transfer cooperate with the SEI transfer. The function of this bit depends on the <MSTR> bit setting.
    Compatibility mode:
    The flag value is undefined when read. A write to the flag is invalid.
    Micro DMA mode:
    Master mode:
    0: Disables automatic shift mode.
    1: Enables automatic shift mode.
    In this mode, reading from the SEDR register causes the following operation:
    - Clear the SEDR register to 00H.
    - Start next data transfer; transmit 00H and receive new 8-bit data.
    By assigning INTSER interrupt as the startup of micro DMA, master device can receive the data block. When the SEI operates in slave mode, automatic shift mode is invalid.
    Slave mode:
    The bit functions as a mask for generating an INTSEE interrupt with the <SOVF> and <WCOL> flags.
    0: Generates an INTSEE interrupt pulse when the <WCOL> flag is set.
    1: Generates an INTSEE interrupt pulse when the <SOVF> flag is set.

<TMSE>: Mode selection
    0: Selects compatibility mode.
    1: Selects micro DMA mode.
    In DMA mode, micro DMA transfer is allowed. Ensure that the SEI function is disabled before attempting to change the mode.

(3) SEI data register (SEDR)

SEI Data Register (for Reception)

SEDR		7	6	5	4	3	2	1	0
(0062H)	bit Symbol	SED7	SED6	SED5	SED4	SED3	SED2	SED1	SED0
Read-modify-write not allowed	Read/Write					R			
	After reset	0	0	0	0	0	0	0	0

SEI Data Register (for Transmission)

SEDR		7	6	5	4	3	2	1	0
(0062H)	bit Symbol	SED7	SED6	SED5	SED4	SED3	SED2	SED1	SED0
Read-modify-write not allowed	Read/Write					W			
	After reset	0	0	0	0	0	0	0	0

Figure 3.11.8 SEI Registers (SEDR)

The SEI data register (SEDR) is used for data transmission and reception. When the SEI is set to a master, writing data to the SEDR register starts data transfer.

Once a transfer has been started, the master device must use an interrupt or polling to ensure that the transfer completion flag is set to 1 before attempting to write new data to the SEDR register.

The SEDR register can be read or written only if the <SEE> bit in the SECR register is set to 1. If the SECR<SEE> bit is set to 0, a write to the SEDR register is ignored and reading the register always returns a value of 00H.

3.11.8 SEI system errors

The SEI device detects three types of system errors. The first type of error occurs if the input to the $\overline{SS}$ pin on the master device is driven Low. This error is called a mode fault. The second type of error, a write collision, occurs if data is written to the SEDR register during data transfer. The third type of error, an overflow error, occurs if a new data byte has been shifted in before the previous data byte has been read when the SEI device is operating as a slave.

（1） Mode fault error

If more than one SEI device is set to a master, contention among drivers may occur.

When an SEI device is set to a master, if its $\overline{SS}$ pin input is driven Low, a mode fault error occurs and the device turns off its driver output. This function prevents contention among masters.

If this error occurs, the device immediately takes the following actions:

- Forcibly clear the <MSTR> bit in the SECR register to 0, thus re-setting the SEI to a slave.

- Forcibly clear the <SEE> bit in the SECR register to 0, thus disabling the SEI function.

- Set the <MODF> flag in the SESR register to 1, which generates an INTSEM interrupt pulse.

- Disable the SEI output pin driver, thus placing the output pin in high-impedance state.

Once the SESR<MODF> flag is cleared to 0 after the software has resolved the problem causing a mode fault, the SEI device can accept setup for recovering normal operation. The SECR register cannot be written if the SESR<MODF> flag is set to 1. In compatibility mode, when the SESR<MODF> flag is set to 1, reading the SESR register and writing to the SEDR register causes the SESR<MODF> flag to be cleared to 0. In micro DMA mode, write a 1 to the SESR<MODF> flag to clear it.

A mode fault error is detected only if more than one device is simultaneously selected as a master. The SEI device cannot detect a collision between MISO pins when more than one slave device is selected on the SEI system.

An open-drain option is provided to protect the device from latch-up. This option changes the SEI output driver to an open-drain driver. Each of the SECLK, MOSI, and MISO pins can be individually set to open-drain programmatically. In that case, an external pull-up resistor needs to be added.

(2) Write collision error

A write collision occurs if the SEDR register is written while data is being transferred. The SEDR register does not have a double-buffer configuration in the direction of transmission so that data written to the SEDR register before transfer is written directly to the SEI shift register. A write during a data transfer thus fails and results in a write collision error. In this case, the on-going data transfer is completed but the written data causing a write collision error is not written to the shift register.

- In slave mode
  The SEDR register is written while the $\overline{SS}$ pin is driven Low. (<CPHA> = 0)
  The SEDR register is written while data is being transferred. (<CPHA> = 1)

- In master mode
  The SEDR register is written while data is being transferred.

A write collision is usually an error on the slave side because a slave cannot control when the master starts a data transfer. The master, which knows when it transfers data, does not cause a write collision error although both master and slave SEI devices can detect a write collision error.

In slave mode, a write collision occurs if the master has already started a shift cycle for a next byte before the slave transfers a new data to the SEDR register.

In slave and micro DMA mode, if the SESR<WCOL> flag is set when the SESR<TASM> bit is 0, an INTSEE interrupt pulse is generated.

(3) Overflow error

The transfer bit rate on the SEI bus is determined by the master. At a high bit rate, the slave may fail to keep up with transfer from the master. Overflow occurs when the following two conditions are satisfied:

- The SEI device is set to a slave.

- A new data has been received but the previous data has not yet been read.

The SEI device can detect a data overflow using the SESR<SOVF> flag. When the SESR <SOVF> flag is set to 1, the SEDR register is overwritten with the new data byte.

In slave mode, if the SESR<SOVF> flag is set when the SESR<TASM> bit is 1, an INTSEE interrupt pulse is generated in micro DMA mode only. The SESR <TASM> bit is used as an interrupt mask bit because this error occurs in slave mode only.

3.11.9 Generating an interrupt

Interrupt handling differs between two SEI operating modes and can be selected using the SESR<TMSE> bit. The SEI generates four types of interrupts: INTSEM, INTSEE, INTSER, and INTSET.

（1）Compatibility mode

In compatibility mode, the SEI generates two types of interrupts, INTSEM and, INTSEE. An INTSEM interrupt pulse is generated if the SESR<MODF> flag changes its state from 0 to 1. An INTSEE interrupt pulse is generated if the SESR<SEF> flag changes its state from 0 to 1.

Table 3.11.6 Compatibility Mode

INTSEM	Interrupt on <MODF>
INTSEE	Interrupt on <SEF>
INTSER	Inactive
INTSET	Inactive

（2）Micro DMA mode

In micro DMA mode, all of four interrupt types are used to enable micro DMA transfer with the SEDR register. An INTSEM interrupt pulse is generated if the SESR<MODF> flag changes its state from 0 to 1. An INTSEE interrupt is generated if, in slave mode, the SESR<WCOL> flag changes its state from 0 to 1 when the SESR <TASM> bit is set to 0. It is also generated if, in slave mode, the SESR<SOVF> flag changes its state from 0 to 1 when the SESR <TASM> bit is set to 1.

Upon the completion of transfer, both the SESR<TSRC> and <TSTC> flags are set to 1 simultaneously, except when SECR<CPHA> is set to 0 in slave mode. See "3.11.4(1) Format when <CPHA> = 0." Those flags trigger the generation of INTSER and INTSET interrupt pulses, respectively.

An INTSER interrupt pulse is generated if the SESR<TSRC> flag changes its state from 0 to 1. The SESR <TSRC> flag is cleared to 0 by reading the SEDR register or writing a 1 to the SESR <TSRC> bit.

An INTSET interrupt pulse is generated if the SESR<TSTC> flag changes its state from 0 to 1. The SESR <TSTC> flag is cleared to 0 by writing to the SEDR register or writing a 1 to the SESR <TSTC> bit.

When using micro DMA transfer to and from the SEDR register, use INTSER and INTSET interrupts as triggers for micro DMA transfer.

INTSER interrupt: Use as a trigger to read the SEDR register.

INTSET interrupt: Use as a trigger to write to the SEDR register.

Next data transfer is started that way.

Table 3.11.7 Compatibility Mode

INTSEM	Interrupt on <MODF>
INTSEE	Interrupt on <WCOL>[1] or <SOVF>[2]
INTSER	Interrupt on <TSRC>
INTSET	Interrupt on <TSTC>

Note 1: When SESR <TASM> = 0 in slave mode
Note 2: When SESR <TASM> = 1 in slave mode

Each interrupt handling should be enabled or disabled using the interrupt controller. (See 3.4.3.)

3.11.10 Using micro DMA with the SEI (micro DMA mode)

Micro DMA improves the SEI transfer speed by:

• taking the load off the CPU for interrupt handling, and

• reducing the time interval between transfers.

Micro DMA transfer is used in both master and slave modes.

(1) Micro DMA transfer (read/write)

In this mode, set the <TMSE> bit in the SESR register to 1 to select micro DMA mode. Two micro DMA channels are used. One channel is used to transfer received data from the SEDR register to a specified RAM area while the other channel is used to transfer transmit data from a specified RAM area to the SEDR register. Data transfer is completely under the control of the micro DMA controller.

a. Initialization

Two micro DMA channels are used for SEI transfer. One micro DMA channel is set to be activated with an INTSER interrupt pulse and transfer received data from the SEDR register to memory. The other channel is set to be activated with an INTSET interrupt pulse and transfer new data from memory to the SEDR register. In master mode, data transfer is restarted with those initial settings.

Set the micro DMA channel having the smaller channel number to an INTSER interrupt. This ensures that received data is read before a write to the SEDR register to start new data transfer.

In master mode, writing to the SEDR register starts the first data transfer. In slave mode, write data to the SEDR register as a preparation for transfer started from the connected master. Subsequent transfers are automatically performed by the micro DMA controller.

Table 3.11.8 SEI Setting when Micro DMA Transfer (Read/Write)

<SEE>	<MSTR>	<TASM>	<TMSE>
1	0:Slave	INTSEE interrupt mask	1
	1:Master	0	

b. Micro DMA transfer

Once initialized, micro DMA waits for a data transfer completion trigger. Upon the completion of transfer, the <TSRC> and <TSTC> flags are set to 1, thus causing an SEI reception completion interrupt (INTSER) pulse and SEI transmission completion interrupt (INTSET) pulse to be generated. The micro DMA channel having the smaller channel number is processed first. Therefore, read processing with micro DMA transfer caused by reception completion is performed before write processing with micro DMA transfer caused by transmission completion. Read processing with micro DMA transfer consists of reading the SEDR register and writing to the address specified with the micro DMA transfer destination address register. In addition, read the SEDR register and clear the <TSRC> flag to 0. Then, for write processing with micro DMA transfer, read the address specified with the micro DMA transfer source address register and write to the SEDR register. In addition, read the SEDR register and clear the <TSTC> flag to 0. If the SEI is the master, start a new data transfer.

After each micro DMA transfer, decrement the transfer count register for both micro DMA transfers. The above procedure continues until the transfer count register becomes 0. Once the transfer count register becomes 0, a micro DMA transfer completion interrupt occurs. The service routine for a micro DMA transfer completion interrupt is used to reinitialize micro DMA transfer
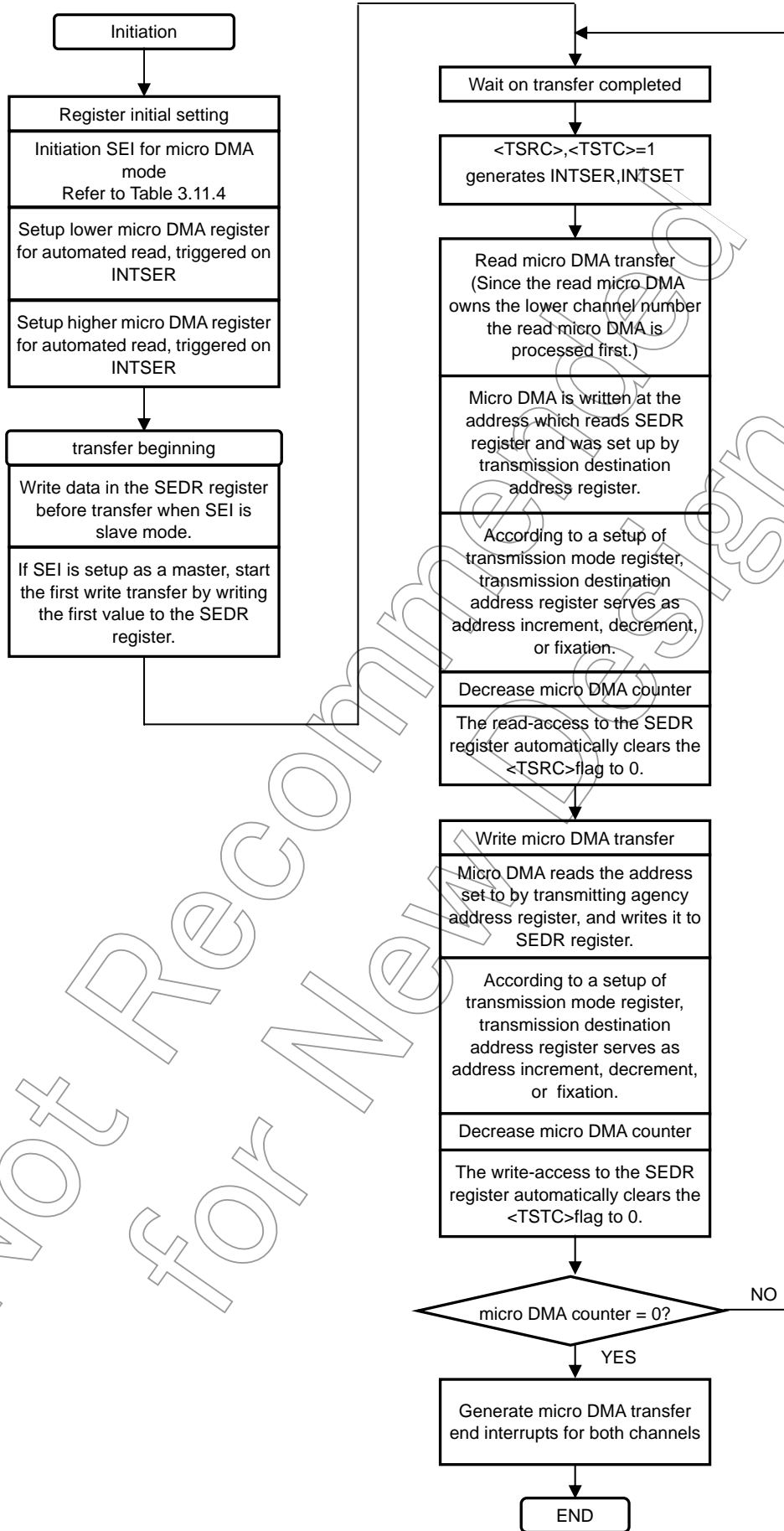
Figure 3.11.9 Flowchart for Micro DMA Read/Write Transfer

(2) Micro DMA transfer (read only)

This mode is used to receive a data block (for example, to read data from serial E²PROM). Meaningless data is transmitted simultaneously. A single micro DMA channel is used to read received data from the SEDR register and store it in a specified RAM area.

a. Initialization

In this mode, set the <TMSE> bit in the SESR register to 1 to select micro DMA mode. The SESR<TASM> bit is used as an automatic shift enable bit while the SEI is operating as a master. One micro DMA channel is set to be activated with an INTSER interrupt pulse and transfer received data from the SEDR register to memory. An INTSER interrupt activates micro DMA transfer. An INTSET interrupt must be disabled using the interrupt controller. When the SEI is a master, writing data to the SEDR register starts the first data transfer. (When the SEI is a slave, it waits until it receives data transmitted from the master.)

Table 3.11.9 SEI Setting when Micro DMA Transfer (Read)

<SEE>	<MSTR>	<TASM>	<TMSE>
1	0: Slave	INTSEE interrupt mask	1
	1: Master	1	

b. Micro DMA transfer

After starting the first data transfer, micro DMA waits until the data transfer is completed. Upon the completion of data transfer, both <TSRC> and <TSTC> flags in the SESR register are set to 1. An INTSER interrupt pulse caused by the SESR<TSRC> flag being set activates micro DMA transfer. The SESR <TSTC> flag is also set to 1 simultaneously and remains set until the block transfer is completed.

Micro DMA reads the received data from the SEDR register and writes it to the memory address specified with the micro DMA transfer destination address register. After each data transfer, the micro DMA transfer count register is decremented. Reading the SEDR register causes it to be automatically cleared to 00H because the SESR <TASM> bit is set to 1. At that time, a new data transfer starts automatically. The above processing continues until the micro DMA transfer count register becomes 0. Once the transfer count register becomes 0, a micro DMA transfer completion interrupt occurs.

After the first data transfer is completed, the SESR <TSTC> flag remains set to 1 unless it is explicitly cleared.

```
┌──────────┐                                    ┌──────────────────────────┐
│ Initiation│                              ┌──→│ Wait on transfer completed │
└────┬─────┘                               │    └────────────┬─────────────┘
     │                                     │                 │
┌──────────────────────┐                   │    ┌──────────────────────────┐
│ Register initial setting │                │    │      <TSRC>=1             │
└────────┬─────────────┘                   │    │   generates INTSER        │
         │                                 │    └────────────┬─────────────┘
┌──────────────────────────┐              │                 │
│ Set SEI to micro DMA mode.│              │    ┌──────────────────────────┐
│ Set it to the automatic   │              │    │ Read micro DMA transfer   │
│ shift mode (<TASM>=1) when │             │    │                           │
│ SEI is a master.          │              │    │ Micro DMA is written at   │
│ Refer to Table 3.11.5     │              │    │ the address which reads   │
└────────┬─────────────────┘              │    │ SEDR register and was set │
         │                                 │    │ up by read transmission   │
┌──────────────────────────┐              │    │ destination address reg.  │
│ Setup micro DMA channel   │              │    ├──────────────────────────┤
│ for automated read,       │              │    │ According to a setup of   │
│ triggered on INTSER.      │              │    │ transmission mode reg...  │
└────────┬─────────────────┘              │    ├──────────────────────────┤
         │                                 │    │ Decrease micro DMA counter│
┌──────────────────────────┐              │    │ The read-access to SEDR...│
│ transfer beginning        │              │    └────────────┬─────────────┘
│ If SEI is setup as master │              │    ┌──────────────────────────┐
│ ...                       │              │    │ When <TASM>=1 ...         │
└───────────┬──────────────┘              │    └────────────┬─────────────┘
            └──────────────────────────────┘                │
                                                  ┌──────────────────┐  NO
                                                  │ micro DMA counter=0? ├──┘
                                                  └─────────┬────────┘
                                                       YES  │
                                              ┌──────────────────────────┐
                                              │ Generate micro DMA        │
                                              │ transfer end interrupts   │
                                              └────────────┬─────────────┘
                                                      ┌────────┐
                                                      │  END   │
                                                      └────────┘
```

Figure 3.11.10 Flowchart for Micro DMA Read only Transfer

(3) Micro DMA transfer (write only)

This mode is used to transmit a data block. Received data is ignored. Only a single micro DMA channel is used to read data from the memory address specified with the micro DMA transfer source address register and write new data to the SEDR register.

a. Initialization

In this mode, set the <TMSE> bit in the SESR register to 1 to select micro DMA mode. One micro DMA channel is set to transfer transmit data from the memory address specified with the micro DMA transfer source address register to the SEDR register. An INTSET interrupt activates this micro DMA transfer. An INTSER interrupt must be disabled using the interrupt controller. When the SEI is a master, writing data to the SEDR register starts the first transfer. (When the SEI is a slave, it waits until it receives data transmitted from the master.)

Table 3.11.10 SEI setting when micro DMA transfer (write)

<SEE>	<MSTR>	<TASM>	<TMSE>
1	0: Slave	INTSEE interrupt mask	1
	1: Master	0	

b. Micro DMA transfer

After starting the first data transfer, micro DMA waits until the data transfer is completed. Upon the completion of data transfer, both <TSRC> and <TSTC> flags in the SESR register are set to 1. Ignore the SESR <TSRC> and SESR <SOVF> flags because reception is not performed. After the first transfer is completed, the SESR <TSRC> flag remains set to 1 unless it is explicitly cleared. The SESR <SOVF> flag, once set, also remains set to 1 unless it is explicitly cleared. An INTSET interrupt pulse caused by the SESR<TSTC> flag being set activates micro DMA transfer.

Micro DMA reads transmit data from the memory address specified with the micro DMA transfer source address register and writes it to the SEDR register. A write to the SEDR register causes the SESR <TSTC> flag to be cleared to 0 and, if the SEI is the master, a new data transfer to start. After each data transfer, the micro DMA transfer count register is decremented. The above processing continues until the micro DMA transfer count register becomes 0. Once the transfer count register becomes 0, a micro DMA transfer completion interrupt occurs.
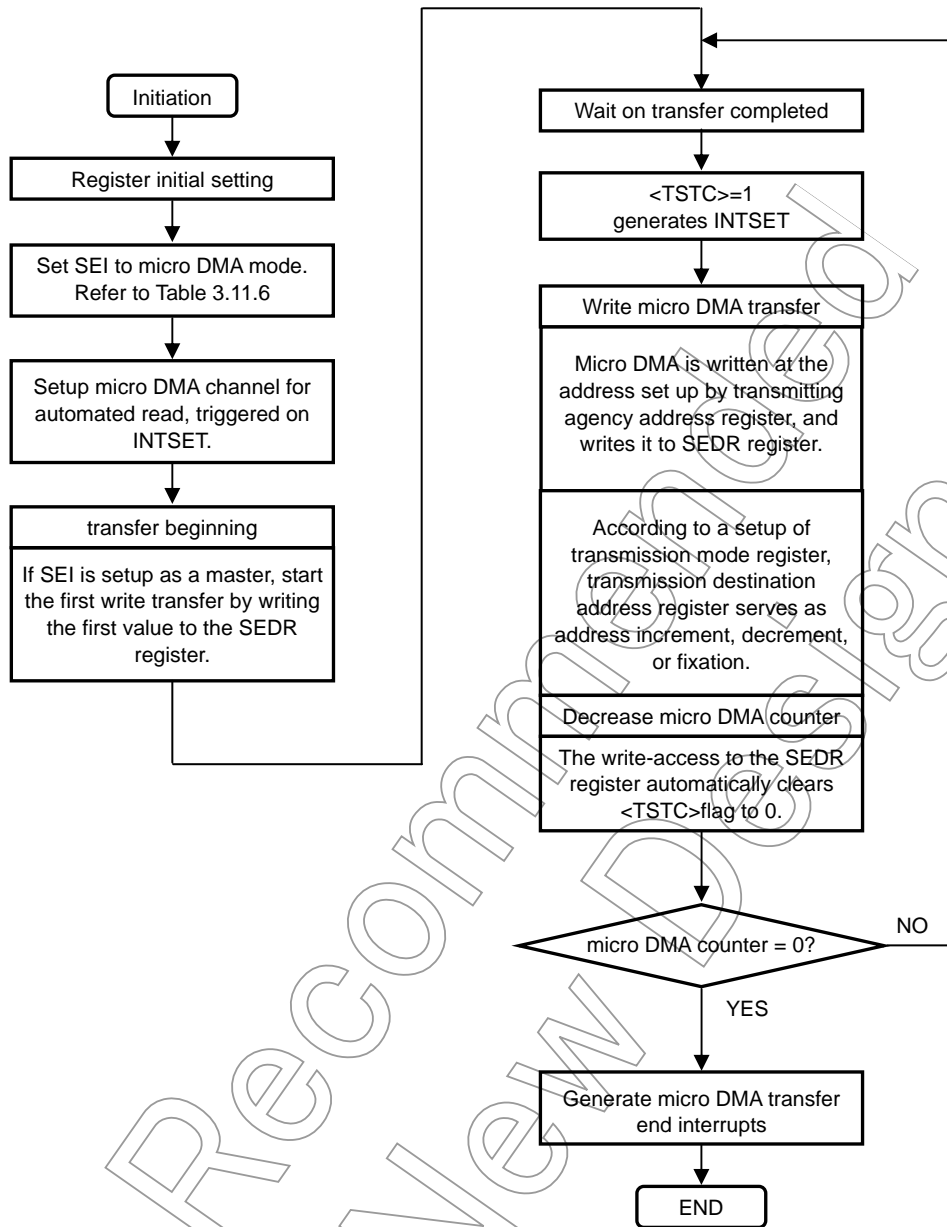
Figure 3.11.11 Flowchart for Micro DMA Write only Transfer

3.12 CAN Controller

(1) Overview

- Complies with CAN version 2.0B.

- Supports the standard and extended formats.

- Supports data and remote frames in each format.

- 16 mailboxes (15 shared for transmission and reception, and 1 for reception only)

- CAN bus baud rate: Up to 1 Mbps (when operating frequency $fc = 20$ MHz)

- Programmable baud rate using bit time parameters

- Built-in baud rate prescaler

- Two types of internal arbitration to select the order of message transmission:
    - a. Ascending order of mailbox number
    - b. Descending order of ID priority

- Timestamp for message transmission/reception

- Operating modes
    - a. Normal operation mode
    - b. Configuration mode
    - c. Sleep mode (can wake up upon detection of CAN bus active state or upon CPU access)
    - d. Halt mode
    - e. Test loopback mode (stand-alone operation possible with self-acknowledge)
    - f. Test error mode (can write to error counter)

- Two types of message reception masking
    - a. Programmable global reception mask (common to mailboxes 0 to 14)
    - b. Programmable local reception mask (dedicated to mailbox 15)

- Reception mask bit for ID extension bit

- Flexible interrupt structure (three interrupt signals)
    - a. INTCR: Reception completion interrupt
    - b. INTCT: Transmission completion interrupt
    - c. INTCG: Global interrupt
      (with eight interrupt sources, including warning level, error passive, and bus off)

(2) Legend

- R/W   CPU read and write access allowed

- R   Only CPU read access allowed

- W   Only CPU write access allowed

- R/S   CPU read access and setting (by writing a 1) allowed

- R/C   CPU read access and clearing (by writing a 1) allowed

- For a mailbox, a dash "⟍" in the bit symbol field indicates an empty bit. Its state is undefined when read.

- For a mailbox, a dash "–" in the "Upon reset" field indicates that the initial value is undefined.

- For a control register, a dash "⟍" in the bit symbol field indicates a reserved bit. Its state is undefined when read. When writing to the register, write a 0 to that bit.

(3) Architecture
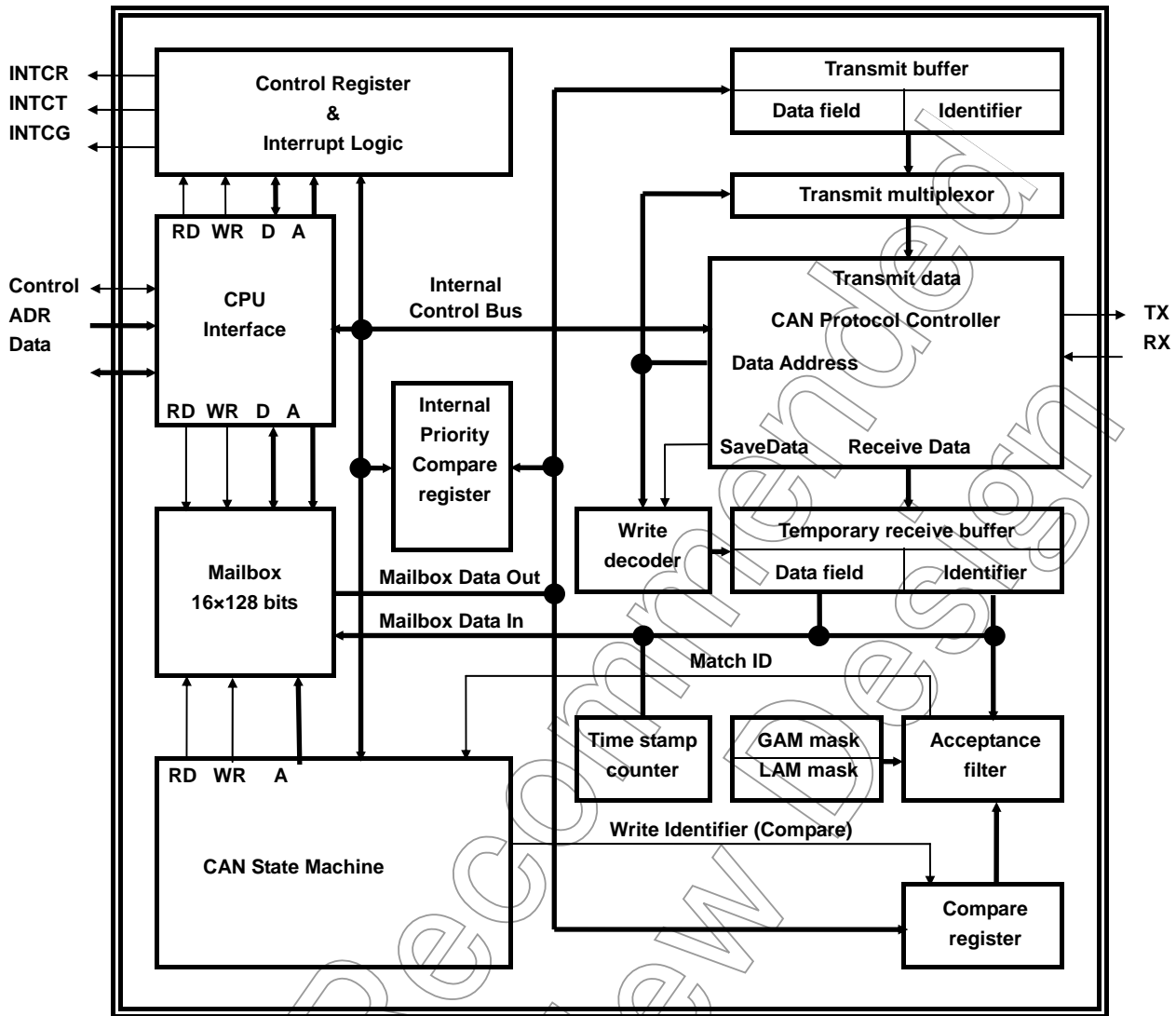
Figure 3.12.1 Block Diagram of CAN Controller

(4) CAN input/output pins

The CAN controller uses RX and TX as its input and output pins, respectively. It should connect to the CAN bus through a CAN transceiver (complying with ISO/DIS 11898).

3.12.1 Memory map

The mailboxes and control registers used for CAN are mapped to the following areas:

Table 3.12.1 CAN Mailboxes and Control Registers

Address	Register	Description
000200H * 000202H * : 0002FEH *	MB0MI0 MB0MI1 : MB15TSV	Mailbox
000300H	MC	Mailbox Configuration Register
000302H	MD	Mailbox Direction Register
000304H *	TRS	Transmit Request Set Register
000306H *	TRR	Transmit Request Reset Register
000308H *	TA	Transmission Acknowledge Register
00030AH *	AA	Abort Acknowledge Register
00030CH *	RMP	Receive Message Pending Register
00030EH *	RML	Receive Message Lost Register
000310H	LAM0 (high)	Local Acceptance Mask Register 0 (bit 28 to 16)
000312H	LAM1 (low)	Local Acceptance Mask Register 1 (bit 15 to 0)
000314H	GAM0 (high)	Global Acceptance Mask Register 0 (bit 28 to 16)
000316H	GAM1 (low)	Global Acceptance Mask Register 1 (bit 15 to 0)
000318H	MCR	Master Control Register
00031AH	GSR	Global Status Register
00031CH	BCR1	Bit Configuration Register 1
00031EH	BCR2	Bit Configuration Register 2
000320H *	GIF	Global Interrupt Flag Register
000322H	GIM	Global Interrupt Mask Register
000324H *	MBTIF	Mailbox Transmit Interrupt Flag Register
000326H *	MBRIF	Mailbox Receive Interrupt Flag Register
000328H	MBIM	Mailbox Interrupt Mask Register
00032AH	CDR	Change Data Request Register
00032CH *	RFP	Remote Frame Pending Register
00032EH *	CEC	CAN Error Counter Register
000330H	TSP	Time Stamp Counter Prescaler Register
000332H *	TSC	Time Stamp Counter Register

Note: RMW prohibited: A read-modify-write operation should not be used.

3.12.2 Mailboxes

A mailbox consists of registers for storing an ID and transmit/receive data and is accessed from the CAN controller or CPU. The CPU controls the CAN controller by modifying the contents of mailboxes and control registers. The contents of mailboxes and control registers are used for reception filtering, message transmission and interrupt handling.

To start transmission, set the corresponding transmission request bit. Subsequently, the CAN controller performs all transmission procedures and error handling, as required, without the intervention of the CPU. If a mailbox is set for reception, the CPU can use a read instruction to read data from the mailbox. A mailbox can also be set to issue an interrupt to the CPU every time a message has been transmitted or received successfully.

There are 16 mailbox, each of which contains 8-byte data, a 29-bit ID and several control bits. Each mailbox can be set for either transmission or reception, except the last one. Mailbox 15 is a receive-only mailbox that has been designed to receive a different group of message IDs using a reception mask different from the one used for mailboxes 0 to 14. A single mailbox consists of 16 bytes.

Address	Mailboxes
0200H to 020FH	MB0 (Used for transmit/receive)
0210H to 021FH	MB1 (Used for transmit/receive)
:	:
:	:
02E0H to 02EFH	MB14 (Used for transmit/receive)
02F0H to 02FFH	MB15 (Used for receive-only)
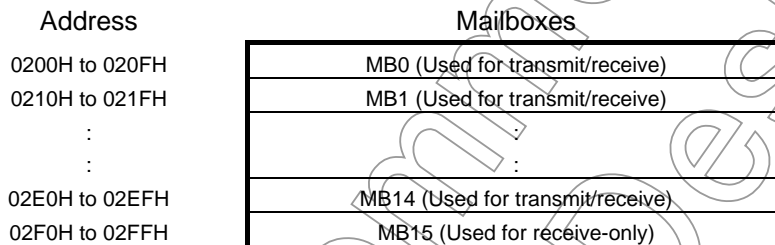
Figure 3.12.2 Mailbox Address

Each mailbox has the following structure:

(Mailbox "n")	b15	b0	
MBn + 00H	MI0		(Message identifier field 0)
02H	MI1		(Message identifier field 1)
04H	MCF		(Message control field)
06H	D1	D0	(Data field 0,1)
08H	D3	D2	(Data field 2,3)
0AH	D5	D4	(Data field 4,5)
0CH	D7	D6	(Data field 6,7)
0EH	TSV		(Time stamp value)
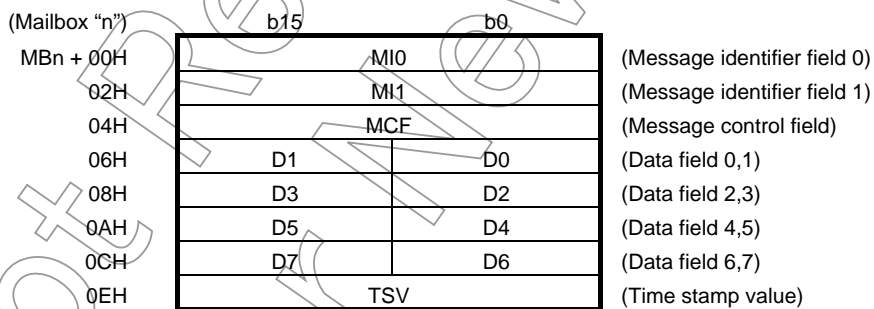
Note: MBn = 0200H + n × 10H, n = 0, 1, 2, …, 15

Figure 3.12.3 Mailbox Structure

The following describes the components of each mailbox.

Message ID field 0 (MI0)

Message Identifier Field 0 Low

MBnMI0L (MBn+00H)		7	6	5	4	3	2	1	0
Read-modify-write not allowed	bit Symbol	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
	Read/Write	R/W							
	After reset	-	-	-	-	-	-	-	-

Identifiers <ID23> to <ID16> are stored.

Message Identifier Field 0 High

MBnMI0H (MBn+01H)		15	14	13	12	11	10	9	8
Read-modify-write not allowed	bit Symbol	IDE	GAME	RFH	ID28	ID27	ID26	ID25	ID24
	Read/Write	R/W							
	After reset	-	-	-	-	-	-	-	-

Identifiers <ID28> to <ID24> are stored.

Remote frame processing bit

0	For transmit mailbox, remote frame are not responded to.
1	For transmit mailbox, remote frame are responded to. (The <TRS> bit is set.)
0/1	For receive mailbox, they are processed as data frames. (The <RMP> and <RFP> bits are set.)

Global (local) acceptance mask enable

0	Acceptance mask is not used for acceptance filtering.
1	Acceptance mask is used for acceptance filtering.

For mailbox 15, it functions as local acceptance mask enable bit <LAME>.

Identifier extension bit

0	Standard format (11-bit identifier) Identifiers <ID28> to <ID18> are used.
1	Extended format (29-bit identifier) Identifiers <ID28> to <ID0> are used.

Note: If the received remote frame has the same ID as that of a transmit mailbox for which <RFH>=1 and <GAME>=1, that mailbox is overwritten with the remote frame ID and automatically responds with the overwritten ID.

Figure 3.12.4 Message ID Field 0

Message ID field 1 (MI1)

Message Identifier Field 1 Low

MBnMI1L (MBn+02H)		7	6	5	4	3	2	1	0
	bit Symbol	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
Read- modify- write not allowed	Read/Write	R/W							
	After reset	-	-	-	-	-	-	-	-

Identifiers <ID7> to <ID0> are stored.

Message Identifier Field 1 High

MBnMI1H (MBn+03H)		15	14	13	12	11	10	9	8
	bit Symbol	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
Read- modify- write not allowed	Read/Write	R/W							
	After reset	-	-	-	-	-	-	-	-

Identifiers <ID15> to <ID8> are stored.

Note: For standard format (11-bit ID), bits <ID17> to <ID0> are undefined.

Figure 3.12.5 Message ID Field 1

A message ID has a higher priority when it contains a longer sequence of zeros from the most significant bit (<ID28>).

Mailbox IDs should be registered as part of initialization. If the message ID field for the mailbox needs to be modified after the mailbox has been enabled, first clear the MC<MCn> bit to 0 to disable the mailbox for the CAN controller before writing a new ID.

Message control field (MCF)

The MCF register consists of the remote transmission request bit (RTR) and data length code (DLC).

A receive mailbox does not need initialization. When a received message is stored into the mailbox, RTR and DLC are also stored into the control field. A transmit mailbox needs initialization.

If the control field for a transmit mailbox for which <RFH> = 1 needs to be modified after the mailbox has been enabled, first clear the MC<MCn> bit to 0 to disable the mailbox for the CAN controller before writing new RTR and DLC. The control field for a transmit mailbox for which <RFH> = 0 can be modified regardless of the <MCn> setting. It is, however, necessary to ensure that the TRS<TRSn> bit is 0 before writing to new RTR and DLC.

Message Control Field Low

MBnMCFL		7	6	5	4	3	2	1	0
(MBn+04H)	bit Symbol				RTR	DLC3	DLC2	DLC1	DLC0
Read-modify-write not allowed	Read/Write						R/W		
	After reset				-	-	-	-	-

Remote transmit request bit ◄

0	Data frame
1	Remote fame

Data length code ◄

<DLC3:0>	Data Bytes	Corresponding Mailbox Data
0000	0 byte	None
0001	1 byte	D0
0010	2 bytes	D1, D0
0011	3 bytes	D2, D0, D0
0100	4 bytes	D3, D2, D1, D0
0101	5 bytes	D4, D3, D2, D1, D0
0110	6 bytes	D5, D4, D3, D2, D1, D0
0111	7 bytes	D6, D5, D4, D3, D2, D1, D0
1000	8 bytes	D7, D6, D5, D4, D3, D2, D1, D0

Note: Any data length code other than the above should not be used.

Message Control Field High

MBnMCFH		15	14	13	12	11	10	9	8
(MBn+05H)	bit Symbol								
Read-modify-write not allowed	Read/Write								
	After reset								
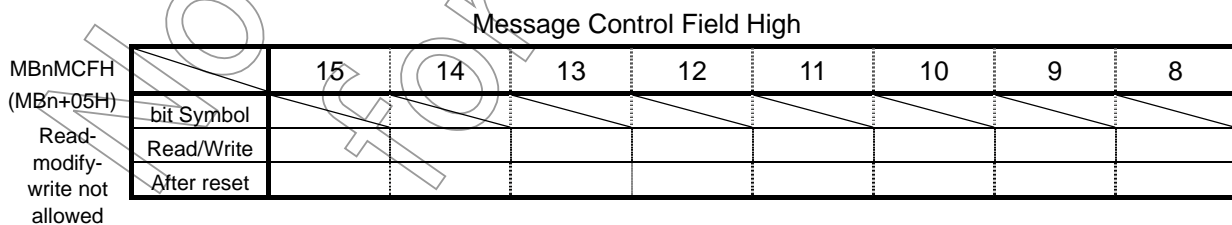
Figure 3.12.6 Message Control Field Register

Data field (D0-D7)

This read/write register contains up to eight bytes of data to be transmitted or received. For a receive mailbox, however, the data field should not be written. A write may result in received data being inconsistent.

For transmission, a number of bytes specified with the DLC in the mailbox will be transmitted.

For reception, the data length code in the received message is copied to the DLC in the mailbox and only that number of data bytes are valid.

To update the data field in a transmission mailbox for which <RFH> = 1, first set CDR<CDRn> to 1 to suspend transmission requests before writing new data. To update the data field in a transmission mailbox for which <RFH> = 0, first ensure that TRS<TRSn> bit is 0 before writing new data.

Data Field 0

MBnD0 (MBn+06H) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	D07	D06	D05	D04	D03	D02	D01	D00
	Read/Write				R/W				
	After reset	-	-	-	-	-	-	-	-

Data Field 1

MBnD1 (MBn+07H) Read-modify-write not allowed		15	14	13	12	11	10	9	8
	bit Symbol	D17	D16	D15	D14	D13	D12	D11	D10
	Read/Write				R/W				
	After reset	-	-	-	-	-	-	-	-

Data Field 2

MBnD2 (MBn+08H) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	D27	D26	D25	D24	D23	D22	D21	D20
	Read/Write				R/W				
	After reset	-	-	-	-	-	-	-	-

Data Field 3

MBnD3 (MBn+09H) Read-modify-write not allowed		15	14	13	12	11	10	9	8
	bit Symbol	D37	D36	D35	D34	D33	D32	D31	D30
	Read/Write				R/W				
	After reset	-	-	-	-	-	-	-	-

Data Field 4

MBnD4 (MBn+0AH) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	D47	D46	D45	D44	D43	D42	D41	D40
	Read/Write				R/W				
	After reset	-	-	-	-	-	-	-	-

Data Field 5

MBnD5 (MBn+0BH) Read-modify-write not allowed		15	14	13	12	11	10	9	8
	bit Symbol	D57	D56	D55	D54	D53	D52	D51	D50
	Read/Write				R/W				
	After reset	-	-	-	-	-	-	-	-

Data Field 6

MBnD6 (MBn+0CH)		7	6	5	4	3	2	1	0
	bit Symbol	D67	D66	D65	D64	D63	D62	D61	D60
Read-modify-write not allowed	Read/Write	R/W							
	After reset	-	-	-	-	-	-	-	-

Data Field 7

MBnD7 (MBn+0DH)		15	14	13	12	11	10	9	8
	bit Symbol	D77	D76	D75	D74	D73	D72	D71	D70
Read-modify-write not allowed	Read/Write	R/W							
	After reset	-	-	-	-	-	-	-	-

Figure 3.12.7 Data Field Register

Timestamp value (TSV)

This 16-bit register is loaded with the value of the timestamp counter when data has been transmitted or received successfully.

It is not loaded if transmission or reception fails.

Time Stamp Value Low

MBnTSVL (MBn+0EH)		7	6	5	4	3	2	1	0
	bit Symbol	TSV7	TSV6	TSV5	TSV4	TSV3	TSV2	TSV1	TSV0
	Read/Write	R							
	After reset	-	-	-	-	-	-	-	-

Time Stamp Value High

MBnTSVH (MBn+0FH)		15	14	13	12	11	10	9	8
	bit Symbol	TSV15	TSV14	TSV13	TSV12	TSV11	TSV10	TSV9	TSV8
	Read/Write	R							
	After reset	-	-	-	-	-	-	-	-

Figure 3.12.8 Timestamp Value Register

3.12.3　Control registers

（1）Mailbox control registers

Mailbox configuration register (MC)

Mailbox Configuration Register Low

MCL (0300H)		7	6	5	4	3	2	1	0
	bit Symbol	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0

Mailbox Configuration Register High

MCH (0301H)		15	14	13	12	11	10	9	8
	bit Symbol	MC15	MC14	MC13	MC12	MC11	MC10	MC9	MC8
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.9　Mailbox Configuration Register

Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively.

Each mailbox can be either enabled or disabled.

　1) When <MCn> = 1, access to mailbox n is enabled for the CAN controller.

　2) When <MCn> = 0, access to mailbox n is disabled for the CAN controller.

A write of 0 or 1 is immediately reflected in the state of the <MCn> bit.

To change the state of <MCn> from 1 (access enabled) to 0 (access disabled), first ensure that the corresponding TRS<TRSn> bit is 0 before writing a 0 to <MCn>.

The control field in a transmit mailbox for which MBnMI0H <RFH> = 1 and the message ID field can only be written after the <MCn> bit is cleared to 0.

The data and control fields in a transmit mailbox for which <RFH> = 0 can be written regardless of whether access to the mailbox is enabled or disabled (if <MCn> = 1, however, it is necessary to ensure that the TRS<TRSn> bit is 0 before writing to the register).

If <MCn> is cleared to 0 while the CAN controller is receiving data, it stops receiving that frame immediately. When the CAN controller is transmitting data (TRS<TRSn> = 1), do not clear <MCn> to 0 before the transmission is completed (TRS <TRSn> = 0).

Mailbox direction register (MD)

Mailbox Direction Register Low

MDL (0302H)		7	6	5	4	3	2	1	0
	bit Symbol	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Mailbox Direction Register High

MDH (0303H)		15	14	13	12	11	10	9	8
	bit Symbol	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8
	Read/Write	R				R/W			
	After reset	1	0	0	0	0	0	0	0

Figure 3.12.10 Mailbox Direction Register

Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively. Each mailbox can be specified as either a transmit or receive mailbox.

Setting <MDn> to 0 causes mailbox n to be used as a transmit mailbox.

Setting <MDn> to 1 causes mailbox n to be used as a receive mailbox.

The <MD15> bit is read-only and fixed to 0 because mailbox 15 is used only for reception. The MD register should be set as part of initialization. To modify settings in the MD register, first set the corresponding <MCn> bit to 0.

(2) Transmission control registers

When data and an ID have been written to mailbox n which has been set as a transmit mailbox (MD<MDn> = 0) and the access to mailbox n is enabled (MC <MCn> = 1), setting the TRS<TRSn> bit to 1 causes a message in the mailbox to be transmitted. If there is more than one transmit request, messages are transmitted sequentially. The order in which messages are transmitted depends on bit 3 (<MTOS>) in the master control register, MCR.

If the MCR<MTOS> bit is set to 1, a message is transmitted from the mailbox having the ID assigned the highest priority among the mailboxes with transmission requests. After the occurrence of arbitration lost, a message is also transmitted from the mailbox having the ID assigned the highest priority among the mailboxes for which transmission requests are pending at that time.

If the MCR<MTOS> bit is set to 0, mailboxes having smaller mailbox numbers have higher priority. For example, if MB0, MB2 and MB5 are specified as transmit mailboxes with their TRS<TRSn> bits set to 1, messages are transmitted in the following order: MB0, MB2 and MB5. If a new transmit request is set for MB0 while a message from MB2 is being processed, the next internal arbitration process selects MB0 for a next transmit message, and starts transmitting an MB0 message after completing transmission for MB2. This procedure also applies if arbitration lost occurs during message transmission for MB2. A message for MB0 is transmitted in place of that for MB2.

Transmit request set register (TRS)

Transmit Request Set Register Low

TRSL		7	6	5	4	3	2	1	0
(0304H)	bit Symbol	TRS7	TRS6	TRS5	TRS4	TRS3	TRS2	TRS1	TRS0
Read-modify-write not allowed	Read/Write					R/S			
	After reset	0	0	0	0	0	0	0	0

Transmit Request Set Register High

TRSH		15	14	13	12	11	10	9	8
(0305H)	bit Symbol		TRS14	TRS13	TRS12	TRS11	TRS10	TRS9	TRS8
Read-modify-write not allowed	Read/Write					R/S			
	After reset		0	0	0	0	0	0	0

Figure 3.12.11 Transmit Request Set Register

Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively. The register does not have bit 15 because mailbox 15 is receive-only.

The TRS<TRSn> bit is cleared to 0 if transmission is successful or if the transmission request is reset by setting the TRR<TRRn> bit to 1. If transmission fails, transmission is retried until if it is successful or if the transmission request is reset by setting the TRR<TRRn> bit to 1. Mailbox n should not be written when the TRS<TRSn> bit is set to 1. If mailbox n is set as a receive mailbox, the CPU cannot set the TRS<TRSn> bit.

If mailbox n is set as a transmit mailbox, the TRS<TRSn> bit is set to 1 when the CPU writes a 1 to it and cleared to 0 by the internal logic. A write of 0 by the CPU is invalid.

Transmit request reset register (TRR)

Transmit Request Reset Register Low

TRRL (0306H) Read- modify- write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	TRR7	TRR6	TRR5	TRR4	TRR3	TRR2	TRR1	TRR0
	Read/Write	R/S							
	After reset	0	0	0	0	0	0	0	0

Transmit Request Reset Register High

TRRH (0307H) Read- modify- write not allowed		15	14	13	12	11	10	9	8
	bit Symbol		TRR14	TRR13	TRR12	TRR11	TRR10	TRR9	TRR8
	Read/Write	R/S							
	After reset		0	0	0	0	0	0	0

Figure 3.12.12 Transmit Request Reset Register

Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively. The register does not have bit 15 because mailbox 15 is receive-only.

Setting the <TRRn> bit to 1 cancels the transmit request set with the corresponding TRS<TRSn> bit. The operation, however, depends on which of the following three conditions applies:

a. If the transmission of a message has not yet started, the message transmission request is canceled.
    (TRS<TRSn> = 0, TRR<TRRn> = 0, and AA<AAn> = 1)

b. If a message is currently being transmitted and if arbitration lost or an error is detected, the message transmission request is cleared and the transmission is stopped.
    (TRS<TRSn> = 0, TRR<TRRn> = 0, and AA<AAn> = 1)

c. If a message is currently being transmitted without arbitration lost or an error being detected, the message transmission request is not cleared and the transmission is completed.
    (TRS<TRSn> = 0, TRR<TRRn> = 0, and TA<TAn> = 1)

Mailbox n should not be written when the TRR<TRRn> bit is set to 1.

If mailbox n is set as a receive mailbox, the CPU cannot set the TRR<TRRn> bit.

If mailbox n is set as a transmit mailbox, the TRR<TRRn> bit is set to 1 when the CPU writes a 1 to it and cleared to 0 by the internal logic when transmission is either successful or aborted. A write of 0 by the CPU is invalid.

Transmit acknowledge register (TA)

Transmit Acknowledge Register Low

TAL (0308H) Read- modify- write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Transmit Acknowledge Register High

TAH (0309H) Read- modify- write not allowed		15	14	13	12	11	10	9	8
	bit Symbol		TA14	TA13	TA12	TA11	TA10	TA9	TA8
	Read/Write	R/C							
	After reset		0	0	0	0	0	0	0

Figure 3.12.13 Transmit Acknowledge Register

Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively. The register does not have bit 15 because mailbox 15 is receive-only.

If a message in mailbox n has been transmitted successfully, the <TAn> bit is set to 1 and a transmission completion interrupt (INTCT) occurs if it is enabled.

The <TAn> bit is cleared to 0 when the CPU writes a 1 to the <TAn> or TRS<TRSn> bit. A write of 0 by the CPU is invalid.

Abort acknowledge register (AA)

Abort Acknowledge Register Low

AAL (030AH) Read- modify- write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	AA7	AA6	AA5	AA4	AA3	AA2	AA1	AA0
	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Abort Acknowledge Register High

AAH (030BH) Read- modify- write not allowed		15	14	13	12	11	10	9	8
	bit Symbol		AA14	AA13	AA12	AA11	AA10	AA9	AA8
	Read/Write	R/C							
	After reset		0	0	0	0	0	0	0

Figure 3.12.14 Transmit Acknowledge Register

Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively. The register does not have bit 15 because mailbox 15 is receive-only.

If message transmission for mailbox n has been canceled, the <AAn> bit and the <TRMABF> bit in the global interrupt flag register (GIF) are set to 1. At that time, a global interrupt (transmission abort), INTCG, occurs if a transmission abort interrupt has been enabled by setting the GIM <TRMABM> bit to 1.

The <AAn> bit is cleared to 0 when the CPU writes a 1 to the <AAn> or TRS <TRSn> bit. A write of 0 by the CPU is invalid.

Change data request register (CDR)

Change Data Request Register Low

CDRL (032AH)		7	6	5	4	3	2	1	0
	bit Symbol	CDR7	CDR6	CDR5	CDR4	CDR3	CDR2	CDR1	CDR0
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Change Data Request Register High

CDRH (032BH)		15	14	13	12	11	10	9	8
	bit Symbol		CDR14	CDR13	CDR12	CDR11	CDR10	CDR9	CDR8
	Read/Write					R/W			
	After reset		0	0	0	0	0	0	0

Figure 3.12.15 Change Data Request Register

Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively. The register does not have bit 15 because mailbox 15 is receive-only.

A transmission request for mailbox n is ignored if the <CDRn> bit is set to 1. In other words, if the TRS<TRSn> and <CDRn> bits are set to 1 for mailbox n, a transmission request for the mailbox is temporarily held and a message is not transmitted unless transmission has already been started. Once the <CDRn> bit is cleared, mailbox n is again subject to internal arbitration.

The function of the <CDRn> bit is valid when updating data fields in a transmission mailbox for which automatic response to a remote frame is enabled (MBnMI0H<RFH> = 1). Using the automatic response function may result in a data field being updated during transmission because data transmission is started in response to a remote frame (in that case, updated data is output from an intermediate point during transmission). To prevent such an update to a data field, the <CDRn> bit can be set to 1 to temporarily hold data transmission.

（3）Reception control registers

The ID of a received message is compared with the ID of a mailbox specified as a receive mailbox. The comparison of IDs depends on the values of the global/local receive mask enable bits (MBnMI0H<GAME>/<LAME>) in the mailbox and the data stored in the global/local receive mask registers (GAM/LAM).

If a match is detected, the received ID, control bits and data bytes are written to the matched mailbox. At this time, the corresponding received message pending bit (RMP<RMPn>) is set to 1 and a reception completion interrupt (INTCR) occurs if it is enabled. Once a match is detected, IDs are not compared subsequently.

If a match is not detected, the message is rejected with the mailbox left intact.

Receive-only mailbox

If the ID of the received message does not match any of the IDs of mailboxes 0 to 14, it is then compared with the ID of receive-only mailbox 15. If a match is detected, the contents of the received message are stored into mailbox 15.

Received message pending register (RMP)

Received Message Pending Register Low

RMPL (030CH)		7	6	5	4	3	2	1	0
	bit Symbol	RMP7	RMP6	RMP5	RMP4	RMP3	RMP2	RMP1	RMP0
Read-modify-write not allowed	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Received Message Pending Register High

RMPH (030DH)		15	14	13	12	11	10	9	8
	bit Symbol	RMP15	RMP14	RMP13	RMP12	RMP11	RMP10	RMP9	RMP8
Read-modify-write not allowed	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.16 Received Message Pending Register

Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively.

The <RMPn> bit is set to 1 once a message has been received and its contents stored in mailbox n.

After reading the received data, write a 1 to the RMP<RMPn> bit to clear the bit. If the mailbox receives a next message with the RMP<RMPn> bit still set to 1, the corresponding <RMLn> bit in the received message lost register (RML) is set to 1. In such a case, the data stored in mailbox n is overwritten with new data. A global interrupt (received message lost), INTCG, also occurs if a received message lost interrupt has been enabled by setting the GIM <RMLIF> bit to 1.

The <RMPn> bit is set to 1 by the internal logic and cleared to 0 when the CPU writes a 1 to the <RMPn> bit. A write of 0 to the <RMPn> bit by the CPU is invalid.

Received message lost register (RML)

Received Message Lost Register Low

RMLL (030EH)		7	6	5	4	3	2	1	0
	bit Symbol	RML7	RML6	RML5	RML4	RML3	RML2	RML1	RML0
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0

Received Message Lost Register High

RMLH (030FH)		15	14	13	12	11	10	9	8
	bit Symbol	RML15	RML14	RML13	RML12	RML11	RML10	RML9	RML8
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.17 Received Message Lost Register

Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively.

If a mailbox for which the RMP<RMPn> bit is set to 1 receives a next message, the mailbox is overwritten with the new data and the <RMLn> bit set to 1.

The <RMLn> bit is set to 1 by the internal logic. It is cleared to 0 by the internal logic when the CPU writes a 1 to the RMP<RMPn> bit. A write of 1 or 0 to the <RMLn> bit by the CPU is invalid.

Table 3.12.2 Operation when Message is Received

ID	Before	After		Operation
	<RMP>	<RMP>	<RML>	
Unmatched	Don't care	No change	No change	The data in receive buffer hasn't been transferred to any mailbox.
Matched	0	1	No change	The data in receive buffer is transferred to a mailbox which matched the identifier of incoming message. (Old data in the mailbox was read out, and cleared <RMP> to 0. Then, the mailbox is written with new data; RECEIVE MESSAGE PENDING BIT is set.)
	1	1	1	The data in receive buffer is transferred to a mailbox which matched the identifier of incoming message (Old data is in the mailbox. Then, the mailbox is overwritten with new data; RECEIVE MESSAGE LOST BIT and RECEIVE MESSAGE PENDING BIT are set).

(4) Remote frame control registers

When a remote frame is received, it is compared with the IDs of all mailboxes. The comparison of IDs depends on the values of the global/local receive mask enable bits (MBnMI0L <GAME>/<LAME>) in the mailbox and the data stored in the global/local receive mask registers (GAM/LAM).

If it matches the ID of transmit mailbox n for which the MBnMI0H<RFH> bit is set to 1, the TRS<TRSn> bit is set to 1 to transmit a message in response to the remote message. A transmit mailbox with the MBnMI0H<RFH> bit set to 0 does not response to the remote frame even if it has the matched ID.

If the ID matches that of a receive mailbox, the remote frame is handled as a data frame and the RMP<RMPn> and RFP<RFPn> bits are set to 1.

Once a match is detected, subsequent IDs are not compared.

Table 3.12.3 Operation when Remote Frame is Received

ID	Mailbox	<RFH> bit	Handling of Remote Frame
Matched	Transmit	0	Not responded to.
		1	Responded to. (<TRS> bit is set) *Note
	Receive	1/0	Not responded to. Processed as data frame. (<RMP> and <RFP> bits are set.)
Unmatched	Transmit/Receive	1/0	Not responded to.

Note: If the ID matches that of a mailbox with MBnMI0L<GAME>=1, the ID of the mailbox is overwritten with the remote frame ID and automatic response is performed with that ID. Therefore, a response may be made to more than one ID for a single mailbox.

Remote frame pending register (RFP)

Remote Frame Pending Register Low

RFPL (032CH)		7	6	5	4	3	2	1	0
	bit Symbol	RFP7	RFP6	RFP5	RFP4	RFP3	RFP2	RFP1	RFP0
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0

Remote Frame Pending Register High

RFPH (032DH)		15	14	13	12	11	10	9	8
	bit Symbol	RFP15	RFP14	RFP13	RFP12	RFP11	RFP10	RFP9	RFP8
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.18 Remote Frame Control Register

If mailbox n that is set as a receive mailbox receives a remote frame, the <RFPn> and RMP<RMPn> bits are set to 1. The <RFPn> bit is cleared to 0 when the CPU writes a 1 to the <RMPn> bit. A write of 0 is invalid. The <RFPn> bit is also cleared to 0 if mailbox n that has received a remote frame receives a data frame and is overwritten.

(5) Receive filter registers

The global receive mask registers, GAM0 and GAM1, are used to filter messages when the MBnMI0H <GAME> bit is set to 1 for mailboxes 0 to 14. The received message is stored into the first mailbox with its ID matched. Only if the ID does not match any of mailboxes 0 to 14, the received message is compared with receive-only mailbox 15. The local receive mask registers, LAM0 and LAM1, are used to filter messages when the MBnMI0H <LAME> bit is set to 1 for mailbox 15.
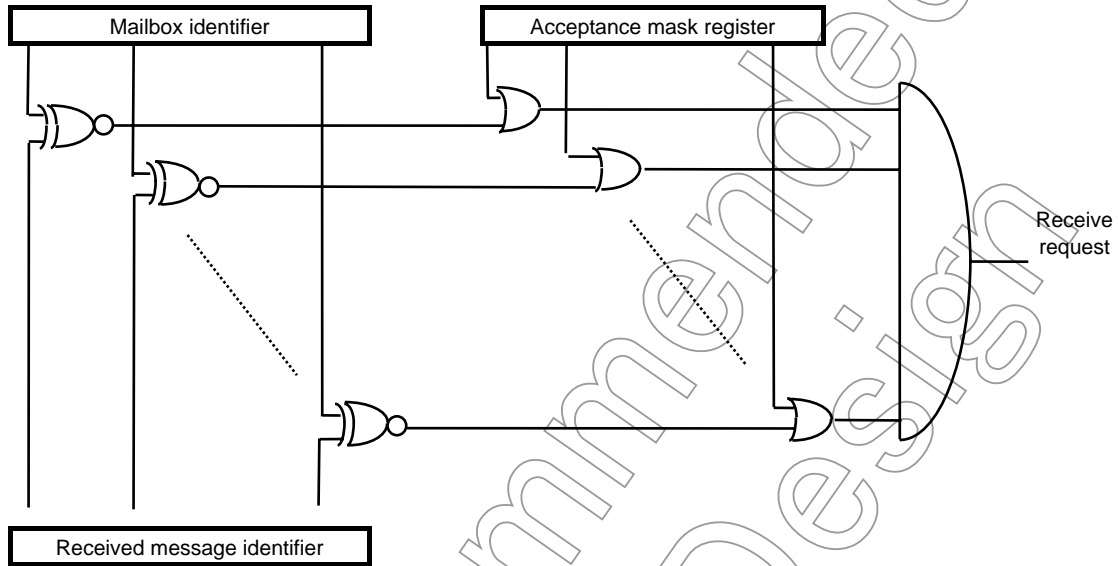
Figure 3.12.19 Acceptance Filter

Local receive mask registers (LAM0 and LAM1)

Local Receive Mask Register 0 Low

LAM0L (0310H)		7	6	5	4	3	2	1	0
	bit Symbol	LAM23	LAM22	LAM21	LAM20	LAM19	LAM18	LAM17	LAM16
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Local Receive Mask Register 0 High

LAM0H (0311H)		15	14	13	12	11	10	9	8
	bit Symbol	LAMI			LAM28	LAM27	LAM26	LAM25	LAM24
	Read/Write	R/W					R/W		
	After reset	0			0	0	0	0	0

Local Receive Mask Register 1 Low

LAM1L (0312H)		7	6	5	4	3	2	1	0
	bit Symbol	LAM7	LAM6	LAM5	LAM4	LAM3	LAM2	LAM1	LAM0
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Local Receive Mask Register 1 High

LAM1H (0313H)		15	14	13	12	11	10	9	8
	bit Symbol	LAM15	LAM14	LAM13	LAM12	LAM11	LAM10	LAM9	LAM8
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.20 Local Receive Mask Register

The LAM0 and LAM1 registers are only used to filter messages for mailbox 15. These registers allow the user to locally mask any ID bits of a received message for mailbox 15. A received message is first checked with mailboxes 0 to 14 for a match before compared with mailbox 15.

If the <LAMn> bit is set to 0, a message is received only when the corresponding bit of the received message ID matches the corresponding bit of the mailbox ID. If the <LAMn> bit is set to 1, a message is received regardless of whether the corresponding bit of the received message ID is 0 or 1. The GAM0 and GAM1 register do not affect mailbox 15.

For the extended format, the MBnMI0H <IDE> bit and all 29 bits of the ID are compared. For the standard format, the MBnMI0H <IDE> bit and the first 11 bits of the ID (<ID28> to <ID18>) are compared.

The <LAMI> bit (local receive mask ID extension bit) is a mask bit for the MB15MI0H <IDE> bit for mailbox 15.

If the <LAMI> bit is set to 0, messages in either the standard or extended format is received depending on the MB15MI0H <IDE> bit for mailbox 15.

If the <LAMI> bit is set to 1, messages in the standard and extended formats are received regardless of the value of the MB15MI0H <IDE> bit for mailbox 15. For messages in the extended format, all 29 bits of the mailbox ID and all 29 mask bits in the LAM register are used for filtering. For messages in the standard format, only the first 11 bits of the mailbox ID (<ID28> to <ID18>) and the first 11 bits in the LAM register (<LAM28> to <LAM18>) are used.

The LAM0 and LAM1 registers can only be set during initialization and should not be modified during operation. If their settings are modified during reception, modified receive mask information becomes valid for message ID comparison halfway through the reception.

Global receive mask registers (GAM0 and GAM1)

Global Receive Mask Register 0 Low

GAM0L (0314H)		7	6	5	4	3	2	1	0
	bit Symbol	GAM23	GAM22	GAM21	GAM20	GAM19	GAM18	GAM17	GAM16
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Global Receive Mask Register 0 High

GAM0H (0315H)		15	14	13	12	11	10	9	8
	bit Symbol	GAMI			GAM28	GAM27	GAM26	GAM25	GAM24
	Read/Write	R/W					R/W		
	After reset	0			0	0	0	0	0

Global Receive Mask Register 1 Low

GAM1L (0316H)		7	6	5	4	3	2	1	0
	bit Symbol	GAM7	GAM6	GAM5	GAM4	GAM3	GAM2	GAM1	GAM0
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Global Receive Mask Register 1 High

GAM1H (0317H)		15	14	13	12	11	10	9	8
	bit Symbol	GAM15	GAM14	GAM13	GAM12	GAM11	GAM10	GAM9	GAM8
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.21 Global Receive Mask Register

The GAM0 and GAM1 registers are used to filter messages for mailboxes 0 to 14.

The GAM0 and GAM1 registers are used for received messages when the MBnMI0H <GAME> bit is set to 1 for mailboxes 0 to 14. The received message is stored only into the first mailbox with its ID matched.

If the MBnMI0H <GAMn> bit is set to 0, a message is received only when the corresponding bit of the received message ID matches the corresponding bit of the mailbox ID. If the MBnMI0H <GAMn> bit is set to 1, a message is received regardless of whether the corresponding bit of the received message ID is 0 or 1.

For the extended format, the MBnMI0H <IDE> bit and all 29 bits of the ID are compared. For the standard format, the MBnMI0H <IDE> bit and the first 11 bits of the ID (<ID28> to <ID18>) are compared.

The <GAMI> bit (global receive mask ID extension bit) is a mask bit for the MBnMI0H <IDE> bit for mailboxes 0 to 14.

If the <GAMI> bit is set to 0, messages in either the standard or extended format is received depending on the MBnMI0H <IDE> bit for mailboxes 0 to 14.

If the <GAMI> bit is set to 1, messages in the standard and extended formats are received regardless of the value of the MBnMI0H <IDE> bit for mailboxes 0 to 14. For

messages in the extended format, all 29 bits of the mailbox ID and all 29 mask bits in the GAM register are used for filtering. For messages in the standard format, only the first 11 bits of the mailbox ID (<ID28> to <ID18>) and the first 11 bits in the GAM register (<GAM28> to <GAM18>) are used.

The GAM0 and GAM1 registers can only be set during initialization and should not be modified during operation. If their settings are modified during reception, modified receive mask information becomes valid for message ID comparison halfway through the reception.

(6) Control registers

Master control register (MCR)

Master Control Register Low

MCRL (0318H)		7	6	5	4	3	2	1	0
	bit Symbol	CCR	SMR	HMR	WUBA	MTOS		TSCC	SRES
	Read/Write	R/W						W	
	After reset	1	0	0	0	0		0	0

Master Control Register High

MCRH (0319H)		15	14	13	12	11	10	9	8
	bit Symbol							TSTLB	TSTERR
	Read/Write							R/W	
	After reset							0	0

Figure 3.12.22 Master Control Register

TSTLB: Test loopback
  0: Cancels test loopback mode. (Normal operation)
  1: Requests test loopback mode.
     This mode supports stand-alone operation.

TSTERR: Test error
  0: Cancels test error mode. (Normal operation)
  1: Requests test error mode.
     This mode enables a write to the error counter register (CEC).

CCR: Change configuration request
  0: Cancels configuration mode. (Normal operation)
  1: Requests configuration mode.
     This mode enables a write to the bit configuration registers (BCR1 and BCR2).

SMR: Sleep mode request
  0: Releases sleep mode. (Normal operation)
  1: Requests sleep mode.
     In this mode, the CAN controller clock is stopped and the error counter and transmit requests are reset.

HMR: Halt mode request
  0: Releases halt mode. (Normal operation)
  1: Requests halt mode.
     In this mode, the CAN controller does not transmit or receive messages. It only transmits error flags and acknowledge flags.

WUBA: Wakeup on bus activity
- 0: Wakes up only with write access to the MCR register.
- 1: Wakes up either upon the detection of a bus active state or with write access to the MCR register.

MTOS: Mailbox transmission order select
- 0: Transmits messages in ascending order of the mailbox number.
- 1: Transmits messages in the order of the message ID priority.

TSCC: Timestamp counter clear
- 0: Invalid
- 1: Clears the timestamp counter to 0.

    Note 1: This bit is write-only. When read, it always returns 0.

    Note 2: The timestamp counter is also cleared with a write to the TSP register or a write of 0 to the timestamp counter (TSC).

SRES: Software reset
- 0: Invalid
- 1: Applies a software reset to the CAN controller. It initializes all the registers.

    Note 1: This bit is write-only. When read, it always returns 0.

Bit configuration register 1 (BCR1)

Bit Configuration Register 1 Low

BCR1L (031CH)		7	6	5	4	3	2	1	0
	bit Symbol	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

<BRP7:0> specify the value of the baud rate prescaler. A value of 0 to 255 can be set.

Bit Configuration Register 1 High

BCR1H (031DH)		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write								
	After reset								

Figure 3.12.23 Bit Configuration Register 1

Bit configuration register 2 (BCR2)

Bit Configuration Register 2 Low

BCR2L (031EH)		7	6	5	4	3	2	1	0
	bit Symbol	SAM	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Setting of SAM

<SAM>	Sampling Time
0	1
1	3

Setting of TSEG2

<TSEG22:20>	Unit Time of TSCL
000	Not available
001	2 × TSCL
010	3 × TSCL
011	4 × TSCL
100	5 × TSCL
101	6 × TSCL
110	7 × TSCL
111	8 × TSCL

Setting of TSEG1

<TSEG13:10>	Unit Time of TSCL
0000	Not available
0001	2 × TSCL
0010	3 × TSCL
0011	4 × TSCL
0100	5 × TSCL
0101	6 × TSCL
0110	7 × TSCL
0111	8 × TSCL
1000	9 × TSCL
1001	10 × TSCL
1010	11 × TSCL
1011	12 × TSCL
1100	13 × TSCL
1101	14 × TSCL
1110	15 × TSCL
1111	16 × TSCL

Bit Configuration Register 2 High

BCR2H (031FH)		15	14	13	12	11	10	9	8
	bit Symbol							SJW1	SJW0
	Read/Write							R/W	
	After reset							0	0

Setting of SJW

<SJW1:0>	Adjust Time
00	1 × TSCL
01	2 × TSCL
10	3 × TSCL
11	4 × TSCL

Figure 3.12.24 Bit Configuration Register 2

The bit length is determined from the parameters in BCR2L<TSEG1n>, <TSEG2n>, and BCR1L<BRPn>. All CAN controllers on the CAN bus must operate at the same baud rate. If the operating frequency differs between CAN controllers, adjust the baud rate using the above parameters. The bit timing circuit provides requested bit timings by converting parameters appropriately. The BCR1 and BCR2 registers contain data related to bit timings.
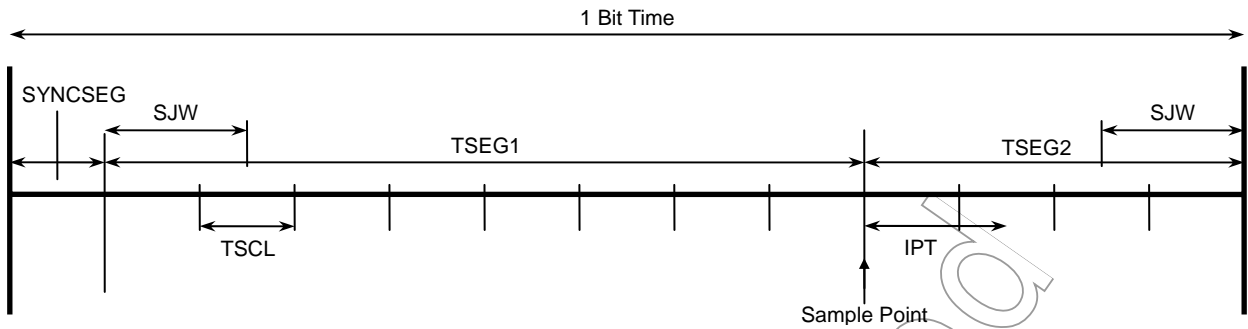
Figure 3.12.25 Bit Timing

The value of TSCL is obtained from the following expression:

$$TSCL = (<BRP7:0> + 1)/f_{IO}$$

        (where f_{IO} is a clock obtained by halving an external clock.)

f_{IO} is the input clock for the CAN controller.

The single-bit length is determined from the following expression:

$$Single\text{-}bit\ length = SYNCSEG + TSEG1 + TSEG2$$

The single-bit length should be set so that it is greater than or equal to $10/f_{IO}$.

The length of the synchronization segment (SYNCSEG) is always $1 \times TSCL$.

For TSEG1, specify a value of TSEG2 or greater.

$$TSEG1 \geq TSEG2$$

The baud rate is obtained from the following expression:

$$Baud\ rate = f_{IO} \div [(<BRP7:0> + 1) \times ((<TSEG13:10> + 1) + (<TSEG22:20> + 1) + 1)]$$

IPT (information processing time) is a time segment starting from the sample point and represents the time required to process a bit read.

$$IPT = 3 / f_{IO}$$

SJW indicates the TSCL time by which the bit length can be increased or reduced during resynchronization. Timing is always synchronized on the rising edge of a signal on the bus. For SJW, specify a value of TSEG2 or less.

$$SJW \leq TSEG2$$

Setting the <SAM> bit enables multisampling on the bus according to the bit timing. The level is determined based on majority rule from three sampled values. If <BRP7:0> < 4, three samples cannot be used.

If <BRP7:0> < 4, only a single sampling is performed regardless of the <SAM> bit setting.

The following restrictions are imposed on the baud rate prescaler:

Table 3.12.4 Baud Rate Prescaler

<BRP7:0>	TSCL length (CAN clock cycles : f_{IO})	IPT length (CAN clock cycles : f_{IO})	TSEG2 minimum length (TSCL)
0	1	3	3
1	2	3	2
> 1	<BRP7:0>+1	3	2

Example1: Setting a baud rate of 1 Mbps (1-bit length = 1 μs)

    CAN clock frequency: f_{IO} = 10 MHz

    Baud rate prescaler: BCR1L<BRP7:0> = 00H

    Since TSCL = 0.1 μs, the single-bit length for data transmission should be programmed with 10 × TSCL. The following shows example parameter settings for that purpose. Program TSEG1 + TSEG2 = 9 × TSCL because SYNCSEG = 1 × TSCL.

        BCR2L<TSEG13:10> = 0100B (5 × TSCL)

        BCR2L<TSEG22:20> = 011B (4 × TSCL)

    Multisampling on the bus cannot be used because <BRP7:0> = 00H, which is less than 4. Therefore, set the <SAM> bit to 0.

    SJW cannot be set to a value greater than TSEG2. In this case, however, SJW can be set to the maximum value.

        BCR2H<SJW1:0> = 11B (4 × TSCL)

Example2: Setting a baud rate of 500 kbps (1-bit length = 2 μs)

    Sample point: 80%

    CAN clock frequency: f_{IO} = 10 MHz

(a)   When BCR1L<BRP7:0> = 00H

        TSCL = (<BRP7:0>+1) / f_{IO} = 1 / 10 MHz = 0.1 μs

    Therefore, the single-bit length for data transmission should be programmed with 20 × TSCL. To set the sample point to 80%:

        TSEG2 = (20 × TSCL) × 0.2 = 4 × TSCL

    Therefore, the BCR2L register bits are set as follows:

        BCR2L<TSEG13:10> = 1110B (15 × TSCL)

        BCR2L<TSEG22:20> = 011B (4 × TSCL)

(b)   When BCR1L<BRP7:0> = 01H

        TSCL = (<BRP7:0>+1) / f_{IO} = 2 / 10 MHz = 0.2 μs

    Therefore, the single-bit length for data transmission should be programmed with 10 × TSCL. To set the sample point to 80%:

        TSEG2 = (10 × TSCL) × 0.2 = 2 × TSCL

    Therefore, the BCR2L register bits are set as follows:

        BCR2L<TSEG13:10> = 0110B (7 × TSCL)

        BCR2L<TSEG22:20> = 001B (2 × TSCL)

Example3: Setting a baud rate of 500 kbps (1-bit length = 2 μs)

    Sample point: 85%

    CAN clock frequency: f_{IO} = 10 MHz

(a)   When BCR1L<BRP7:0> = 00H

        TSCL = (<BRP7:0>+1) / f_{IO} = 1 / 10 MHz = 0.1 μs

    Therefore, the single-bit length for data transmission should be programmed with 20 × TSCL. To set the sample point to 85%:

        TSEG2 = (20 × TSCL) × 0.15 = 3 × TSCL

Therefore, the BCR2L register bits are set as follows:

BCR2L<TSEG13:10> = 1111B (16 × TSCL)

BCR2L<TSEG22:20> = 010B (3 × TSCL)

Timestamp function

The CAN controller has a 16-bit free-running timestamp counter, TSC, to determine the time at which a message was transmitted or received. Upon the completion of storing a received message or transmitting a message, the value of the TSC is written to the timestamp value, TSV, for the corresponding mailbox.

The bit clock on the CAN bus line is supplied through the prescaler to the TSC. The TSC is stopped in configuration mode or sleep mode. Upon reset, the TSC is cleared to 0 by a write to the timestamp counter prescaler register, TSP. The CPU can read and write to the TSC in configuration mode or normal operation mode.

Timestamp counter register (TSC)

Time Stamp Counter Register Low

TSCL (0332H) Read- modify- write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	TSC7	TSC6	TSC5	TSC4	TSC3	TSC2	TSC1	TSC0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0

Time Stamp Counter Register High

TSCH (0333H) Read- modify- write not allowed		15	14	13	12	11	10	9	8
	bit Symbol	TSC15	TSC14	TSC13	TSC12	TSC11	TSC10	TSC9	TSC8
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.26 Timestamp Counter Register

A TSC overflow can be detected using the <TSO> flag in the GSR register and the <TSOIF> flag in the GIF register. Both flags are cleared to 0 by a write of a 1 to the <TSOIF> flag in the GIF register.

A 4-bit prescaler is provided for the TSC. The value to be reloaded to the prescaler is specified with the timestamp counter prescaler register, TSP. Upon a reset, the TSP is cleared to 0 and the prescaler is also loaded with 0. The following shows the count-up period, TTSC, for the TSC:

$T_{TSC} = T_{BIT} \times (<TSP3:0> + 1)$ (where T_{BIT} is a bit period)

Timestamp counter prescaler register (TSP)

Time Stamp Counter Prescaler Register Low

TSPL (0330H)		7	6	5	4	3	2	1	0
	bit Symbol					TSP3	TSP2	TSP1	TSP0
	Read/Write						R/W		
	After reset					0	0	0	0

Time Stamp Counter Prescaler Register High

TSPH (0331H)		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write								
	After reset								

Figure 3.12.27 Timestamp Counter Register

A hold register is implemented to prevent the value of the TSC from varying during a mailbox write cycle. When a message has been transmitted or received successfully, the value of the TSC is copied to the hold register, from which it is written to the mailbox. Reception is successful for the receiver if the message does not contain an error except for the last end-of-frame bit. Transmission is successful for the transmitter if the message does not contain an error including the last end-of-frame bit.
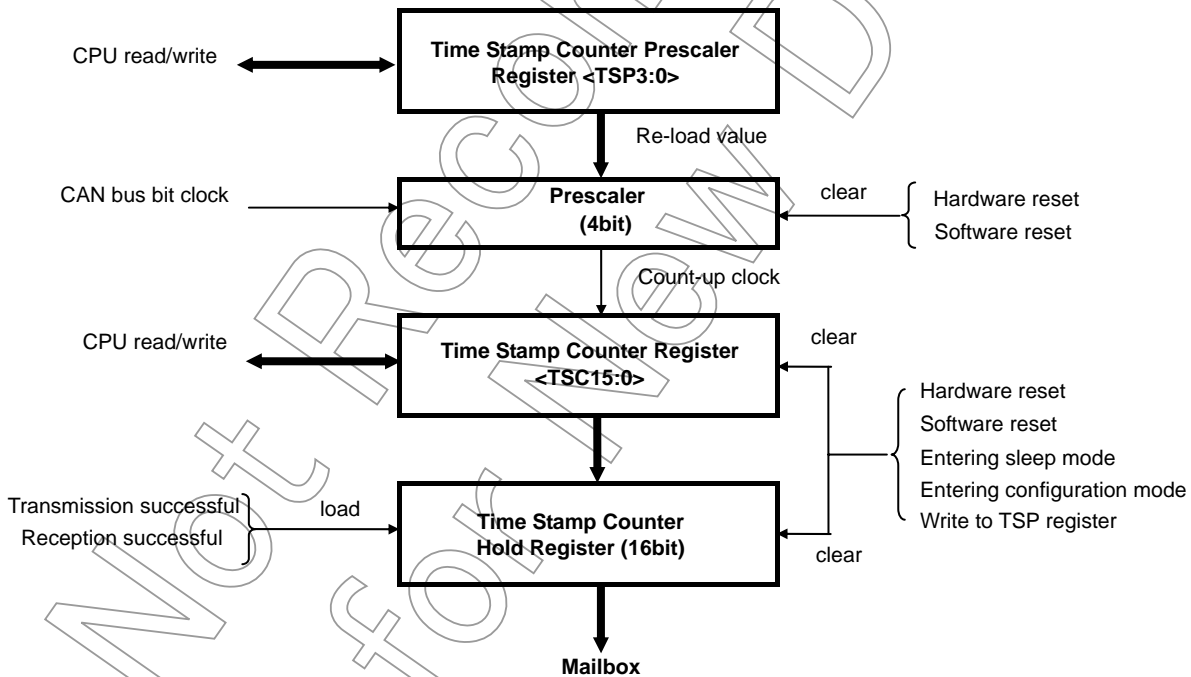
Figure 3.12.28 Time Stamp Counter

The timestamp counter register (TSC) and timestamp hold register are cleared under the following conditions:

- Upon a reset (hardware/software)
- When the device enters configuration mode
- When the device enters sleep mode
- When write access is performed to the TSP register

（7）Status registers

Global status register (GSR)

Global Status Register Low

GSRL (031AH)		7	6	5	4	3	2	1	0
	bit Symbol	CCE	SMA	HMA		TSO	BO	EP	EW
	Read/Write	R					R		
	After reset	1	0	0		0	0	0	0

Global Status Register High

GSRH (031BH)		15	14	13	12	11	10	9	8
	bit Symbol	MsgInSlot<3:0>				RM	TM		
	Read/Write	R							
	After reset	1	1	1	1	0	0		

Figure 3.12.29 Global Status Register

If mailbox n that is set as a receive mailbox receives a remote frame, the <RFPn> and RMP<RMPn> bits are set to 1. The <RFPn> bit is cleared to 0 when the CPU writes a 1 to the <RMPn> bit. A write of 0 is invalid. The <RFPn> bit is also cleared to 0 if mailbox n that has received a remote frame receives a data frame and is overwritten.

MsgInSlot: Message in slot
    Indicates the message contained in the transmit buffer.
    0000: Message from mailbox 0
    0001: Message from mailbox 1
       :
    1110: Message from mailbox 14
    1111: No message contained in the transmit buffer

RM: Receive mode
    0: The CAN controller is not receiving a message.
    1: The CAN controller is receiving a message.

TM: Transmit mode
    0: The CAN controller is not transmitting a message.
    1: The CAN controller is transmitting a message.

CCE: Change configuration enable
    0: The CAN controller is not placed in configuration mode. (Normal operation)
    1: The CAN controller is placed in configuration mode.

SMA: Sleep mode acknowledge
    0: The CAN controller is not placed in sleep mode. (Normal operation)
    1: The CAN controller is placed in sleep mode.

HMA: Halt mode acknowledge
    0: The CAN controller is not placed in halt mode. (Normal operation)
    1: The CAN controller is placed in halt mode.

TSO: Timestamp overflow flag
>   0: The timestamp counter has not overflowed.
>   1: The timestamp counter has overflowed at least once after this bit was cleared to 0.
>   To clear the bit to 0, clear the <TSOIF> bit in the GIF register to 0.

BO: Bus-off status
>   0: Bus-on state (Normal operation)
>   1: Bus-off state
>   The CAN controller enters the bus-off state if the transmit error counter (TEC) reaches a limit of 256 due to abnormally frequent occurrence of errors on the CAN bus. In the bus-off state, messages cannot be transmitted or received. The error counter is undefined in that state. A bus-off recovery sequence causes the CAN controller to enter the bus-on state automatically.

EP: Error passive status
>   0: The CAN controller is placed in error active mode.
>   The values of the transmit error counter (TEC) and receive error counter (REC) are both less than 128.
>   1: The CAN controller is placed in error passive mode.
>   It indicates that either or both of the transmit error counter (TEC) and receive error counter (REC) have reached 128, which indicates the error passive limit.

EW: Warning status
>   0: The values of the transmit error counter (TEC) and receive error counter (REC) are both less than or equal to 96.
>   1: It indicates that either or both of the transmit error counter (TEC) and receive error counter (REC) have exceeded 96, which indicates a warning level.

CAN error counter register (CEC)

CAN Error Counter Register Low

CECL (032EH) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

CAN Error Counter Register High

CECH (032FH) Read-modify-write not allowed		15	14	13	12	11	10	9	8
	bit Symbol	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.30 CAN Error Counter Register

The CAN controller has two error counters, the receive error counter (REC) and transmit error counter (TEC). The CPU can read the values of both error counters. Error counters can only be written in test error mode (when the <TSTERR> bit in the MCR register is set to 1). When writing to the CEC register, writing a value to the lower eight bits (CECL) causes the same value to be also written to the upper eight bits (CECH). A write to the upper eight bits (CECH) is invalid. Error counters are incremented or decremented according to CAN version 2.0B.

The CAN controller is placed in one of the following three states depending on the values of REC and TEC:

(1) Error active status (if TEC < 128 and REC < 128)

The CAN controller enters this state upon a reset release. In this state, it transmits an active error flag if it detects an error.

(2) Error passive status (if TEC ≥ 128 or REC = 128)

In this state, the CAN controller transmits a passive error flag if it detects an error.

(3) Bus-off state (if TEC ≥ 256)

In this state, the CAN controller cannot transmit or receive messages.

The value of REC does not exceed the error passive limit (128). When REC = 128, a successful reception of another message causes the REC to be set back to a value of between 119 and 127. When the CAN controller enters the bus-off state, both of the count values become undefined.

Once placed in the bus-off state, the CAN controller automatically returns to the error active state if it detects a sequence of eleven recessive bits 128 times.

Both error counters are cleared to 0 when the CAN controller enters configuration mode. For details, see "3.12.4(1) Configuration mode."

(8) Interrupt control registers

The CAN controller supports the following interrupt sources:

- Transmit interrupt

  Occurs upon the completion of message transmission.

- Receive interrupt

  Occurs upon the completion of message reception.

- Remote frame receive interrupt

  Occurs when a remote frame is received.

- Wakeup interrupt

  Occurs upon a wakeup from sleep mode.

- Received message lost interrupt

  Occurs upon the detection of received message lost.

- Transmission abort interrupt

  Occurs when message transmission is aborted (when a bit in the AA register is set to 1).

- Timestamp counter overflow interrupt

  Occurs when the timestamp counter overflows.

- Bus-off interrupt

  Occurs when the CAN controller enters the bus-off state.

- Error passive interrupt

  Occurs when the CAN controller enters the error passive state.

- Warning interrupt

  Occurs when either of the two error counters has exceeded 96, reaching a warning level.

These interrupt sources are classified into the following three groups:

- Reception completion interrupt (INTCR) ⎫
- Transmission completion interrupt (INTCT) ⎬ Mailbox interrupts
- Global interrupt (INTCG)

Each interrupt group has a single interrupt output signal assigned. An INTCR interrupt occurs upon the completion of reception. An INTCT interrupt occurs upon the completion of transmission. An INTCG interrupt occurs for any other reasons.

Global interrupt

Global interrupt, INTCG is provided by any interrupt reasons except a mailbox transmission completion and a mailbox reception completion. The global interrupt flag register, GIF, is provided for global interrupt. The global interrupt mask register, GIM, is also provided to enable or disable global interrupt.

Global interrupt flag register (GIF)

Global Interrupt Flag Register Low

GIFL (0320H) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	RFPF	WUIF	RMLIF	TRMABF	TSOIF	BOIF	EPIF	WLIF
	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Global Interrupt Flag Register High

GIFH (0321H) Read-modify-write not allowed		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write								
	After reset								

Figure 3.12.31 Global Interrupt Flag Register

Each interrupt flag in the global interrupt flag register (GIF) is set if the corresponding global interrupt condition is satisfied. A global interrupt flag being set to 1 causes a global interrupt pulse, INTCG, to be generated if the corresponding bit in the interrupt mask register (GIM) is set to 1 (interrupt enabled). If the interrupt condition for the same source is satisfied subsequently, a global interrupt pulse (INTCG) is not generated as long as the interrupt flag in the GIF register is set to 1.

When global interrupt flag is cleared to 0, if another flag has been set to 1, new global interrupt pulse (INTCG) is generated.

Each interrupt flag in the GIF register which is set to 1 is cleared to 0 when the CPU writes a 1 to the flag. A write of 0 is invalid.

RFPF: Remote frame pending flag
    0: A remote frame has not been received.
    1: A remote frame has been received (to a receive mailbox).
    The <RFPF> bit is not, however, set to 1 if the ID matches that of a transmit mailbox for which the <RFH> bit is set to 1.

WUIF: Wakeup interrupt flag
    0: The CAN controller is placed in either sleep mode or normal operation mode.
    1: The CAN controller has woken up from sleep mode.

RMLIF: Received message lost interrupt flag
        0: Received message lost has not occurred.
        1: Received message lost has occurred in at least one receive mailbox. At least one bit in the RML
           register is set to 1.

TRMABF: Transmission abort flag
        0: Transmission abort has not occurred.
        1: Transmission abort has occurred. At least one bit in the AA register is set to 1.

TSOIF: Timestamp counter overflow interrupt flag
        0: No timestamp counter overflow has occurred since this bit was cleared.
        1: A timestamp counter overflow has occurred at least once since this bit was cleared.

BOIF: Bus-off interrupt flag
        0: The CAN controller is placed in bus-on mode.
        1: The CAN controller is placed in bus-off mode.

EPIF: Error passive interrupt flag
        0: The CAN controller is placed in error active mode.
        1: The CAN controller is placed in error passive mode.

WLIF: Warning level interrupt flag
        0: No error counter has reached a warning level.
        1: At least one of the error counters has reached a warning level.

Global interrupt mask register (GIM)

Global Interrupt Mask Register Low

GIML (0322H)		7	6	5	4	3	2	1	0
	bit Symbol	RFPM	WUIM	RMLIM	TRMABM	TSOIM	BOIM	EPIM	WLIM
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0

Global Interrupt Mask Register High

GIMH (0323H)		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write								
	After reset								

Figure 3.12.32 Global Interrupt Mask Register

    The global interrupt mask register (GIM) enables or disables global interrupts for each interrupt condition in the GIF register. Global interrupts for an interrupt condition are disabled when the corresponding bit in the GIM register is set to 0 and enabled when it is set to 1. Upon a reset, all bits in the register are cleared to 0, thus disabling global interrupts.

Mailbox interrupts

Besides global interrupts, interrupts for mailboxes are provided. They include a mailbox transmission completion interrupt, INTCT, and a mailbox reception completion interrupt, INTCR, which depend on mailbox settings. The mailbox transmit interrupt flag register, MBTIF, is provided for mailbox transmission completion interrupts. The mailbox receive interrupt flag register, MBRIF, is provided for mailbox reception completion interrupts. The mailbox interrupt mask register, MBIM, is also provided to enable or disable each mailbox interrupt.

Mailbox interrupt mask register (MBIM)

Mailbox Interrupt Mask Register Low

MBIML (0328H)		7	6	5	4	3	2	1	0
	bit Symbol	MBIM7	MBIM6	MBIM5	MBIM4	MBIM3	MBIM2	MBIM1	MBIM0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0

Mailbox Interrupt Mask Register High

MBIMH (0329H)		15	14	13	12	11	10	9	8
	bit Symbol	MBIM15	MBIM14	MBIM13	MBIM12	MBIM11	MBIM10	MBIM9	MBIM8
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.33 Mailbox Interrupt Mask Register

Bits 0 to 15 in the MBIM register corresponds to mailboxes 0 to 15, respectively.

The MBIM register enables or disables an interrupt for each mailbox.

If the <MBIMn> bit is 0, an interrupt for the corresponding mailbox is disabled.

If the <MBIMn> bit is 1, an interrupt for the corresponding mailbox is enabled.

Mailbox transmit interrupt flag register (MBTIF)

Mailbox Transmit Interrupt Flag Register Low

MBTIFL (0324H) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	MBTIF7	MBTIF6	MBTIF5	MBTIF4	MBTIF3	MBTIF2	MBTIF1	MBTIF0
	Read/Write				R/C				
	After reset	0	0	0	0	0	0	0	0

Mailbox Transmit Interrupt Flag Register High

MBTIFH (0325H) Read-modify-write not allowed		15	14	13	12	11	10	9	8
	bit Symbol		MBTIF14	MBTIF13	MBTIF12	MBTIF11	MBTIF10	MBTIF9	MBTIF8
	Read/Write					R/C			
	After reset		0	0	0	0	0	0	0

Figure 3.12.34 Mailbox Transmit Interrupt Flag Register

The mailbox transmit interrupt flag register, MBTIF, is provided for mailbox transmission completion interrupts. Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively. The MBTIF register does not have bit 15 <MBTIF15> because mailbox 15 is receive-only. If mailbox n is set as a receive mailbox, the corresponding interrupt flag <MBTIFn> in the MBTIF register is always read as 0.

When a message in mailbox n has been transmitted, the <MBTIFn> flag is set to 1 and a mailbox transmission completion interrupt pulse (INTCT) is generated if the corresponding mask bit <MBIMn> in the MBIM register is set to 1 (interrupt enabled).

If the corresponding mask bit in the MBIM register is set to 0, the completion of message transmission does not result in the <MBTIF> flag being set or an INTCT interrupt being generated. To determine whether transmission has been completed, it is necessary to read the TA register.

An INTCT interrupt pulse is generated when an interrupt flag in the MBTIF register is set to 1. If another mailbox transmission completion interrupts condition occurs before that flag is cleared to 0, the corresponding interrupt flag in the MBTIF register is set to 1 but an INTCT interrupt pulse is not generated.

If an interrupt flag in the MBTIF register is cleared to 0 with another interrupt flag still set to 1, an INTCT interrupt pulse is generated.

An interrupt flag in the MBTIF register is cleared to 0 when the CPU writes a 1 to the flag. A write of 0 is invalid.

Mailbox receive interrupt flag register (MBRIF)

Mailbox Receive Interrupt Flag Register Low

MBRIFL (0326H) Read-modify-write not allowed		7	6	5	4	3	2	1	0
	bit Symbol	MBRIF7	MBRIF6	MBRIF5	MBRIF4	MBRIF3	MBRIF2	MBRIF1	MBRIF0
	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Mailbox Receive Interrupt Flag Register High

MBRIFH (0327H) Read-modify-write not allowed		15	14	13	12	11	10	9	8
	bit Symbol	MBRIF15	MBRIF14	MBRIF13	MBRIF12	MBRIF11	MBRIF10	MBRIF9	MBRIF8
	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Figure 3.12.35 Mailbox Receive Interrupt Flag Register

The mailbox receive interrupt flag register, MBRIF, is provided for mailbox reception completion interrupts. Bits 0 to 15 in this register corresponds to mailboxes 0 to 15, respectively. If mailbox n is set as a transmit mailbox, the corresponding interrupt flag <MBRIFn> in the MBRIF register is always read as 0.

When a message for mailbox n has been received, the <MBRIFn> flag is set to 1 and a mailbox reception completion interrupt pulse (INTCR) is generated if the corresponding mask bit <MBIMn> in the MBIM register is set to 1 (interrupt enabled).

If the corresponding mask bit in the MBIM register is set to 0, the completion of message reception does not result in the <MBRIF> flag being set or an INTCR interrupt being generated. To determine whether reception has been completed, it is necessary to read the RMP register.

An INTCR interrupt pulse is generated when an interrupt flag in the MBRIF register is set to 1. If another mailbox reception completion interrupts condition occurs before that flag is cleared to 0, the corresponding interrupt flag in the MBRIF register is set to 1 but an INTCR interrupt pulse is not generated.

If an interrupt flag in the MBRIF register is cleared to 0 with another interrupt flag still set to 1, an INTCR interrupt pulse is generated.

An interrupt flag in the MBRIF register is cleared to 0 when the CPU writes a 1 to the flag. A write of 0 is invalid.

3.12.4   Description of operating modes

（1）Configuration mode

The CAN controller requires initialization (by setting the bit configuration registers, BCR1 and BCR2) before it can start operation. The BCR1 and BCR2 registers can be written in configuration mode only. Upon a reset, the CAN controller enters configuration mode if the <CCR> bit in the MCR register and the <CCE> bit in the GSR register are set to 1. Writing a 0 to the MCRL<CCR> bit places the controller in normal operation mode. When the CAN controller exits from configuration mode, the GSRL <CCE> bit is set to 0 and a power-up sequence starts. In the power-up sequence, the CAN controller detects a sequence of eleven recessive bits on the CAN bus. It then enters the bus-on state and is ready to start operation.

Writing a 1 to the MCRL<CCR> bit causes the CAN controller to exit from normal operation mode and enter configuration mode. When it enters configuration mode, the GSRL <CCE> bit is set to 1.

Figure 3.12.36 shows a CAN initialization flowchart.

In configuration mode, the error counter (CEC), timestamp counter (TSC), and timestamp hold register are cleared.
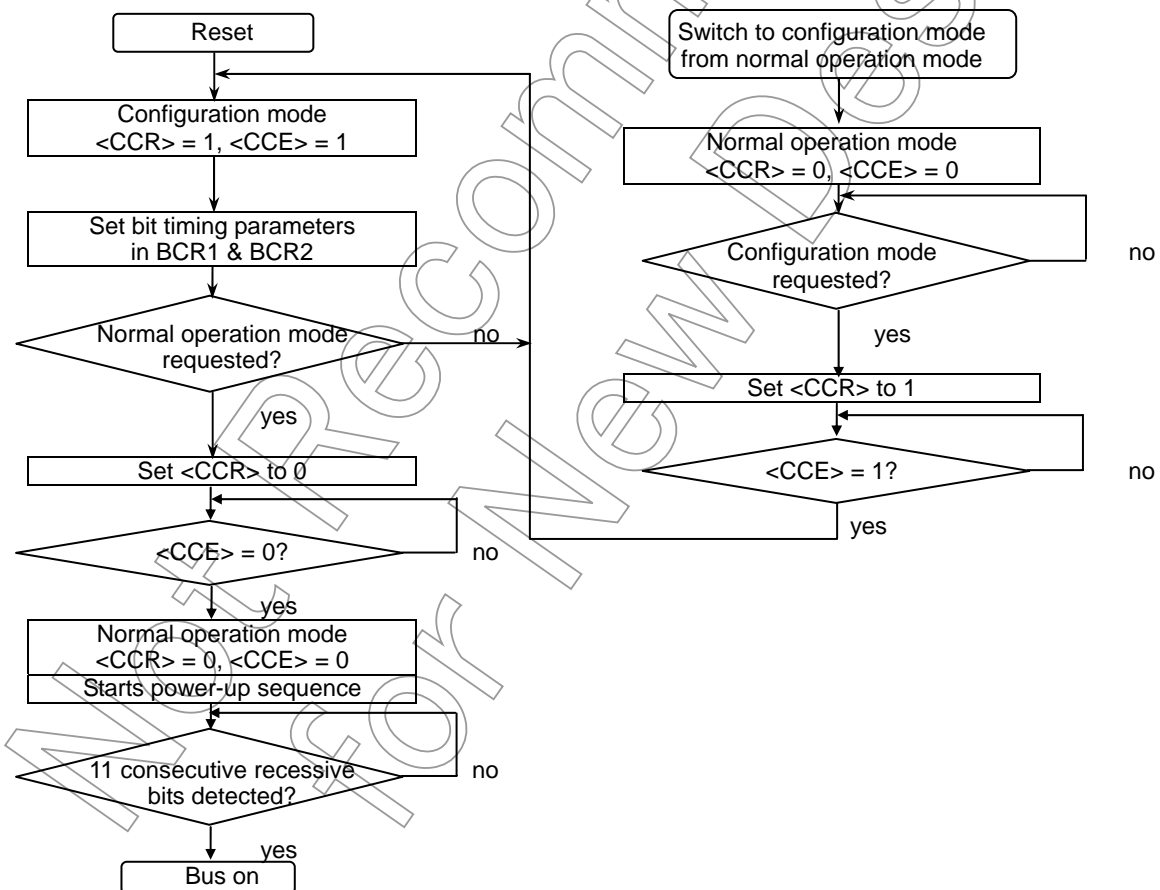
Figure 3.12.36 Flowchart of CAN Initialization

(2) Sleep mode

Writing a 1 to the <SMR> bit in the MCR register request a transition to sleep mode. When the CAN controller enters sleep mode, the <SMA> bit in the GSR register is set to 1.

In sleep mode, the clock for the CAN controller is stopped and only the wakeup circuit is active. The GSR register returns a value of F040H when it is read. It indicates that the transmit buffer contains no message and that sleep mode is active with the GSRL<SMA> bit set to 1. All other registers are read as 0000H. Write access is disabled for all registers other than MCR.

If the CAN controller detects write access to the MCR register or detects a bus active state on the CAN bus when the <WUBA> bit in the MCR register is set to 1, it releases sleep mode (wakes up) and starts a power-up sequence. It waits until a sequence of eleven recessive bits are detected on the RX input pin and then enters a bus active state. The first message that has triggered a transition to a bus active state cannot be received.

In sleep mode, the CAN error counter and all transmit request set (TRS<TRSn>) bits and transmission request reset (TRR<TRRn>) bits are cleared to 0. When the CAN controller exits from sleep mode, the <SMR> bit in the MCR register and the <SMA> bit in the GSR register are cleared to 0.

If sleep mode is requested (MCR<SMR>=1) when the CAN controller is transmitting a message, it enters the sleep mode after either condition as follows:

- The CAN controller completes the transmission successfully.

- After arbitration lost, the CAN controller completes the reception of a message successfully.

- After arbitration lost, the CAN controller detects an error on the CAN bus during the reception of a message.

(3) Halt mode

Writing a 1 to the <HMR> bit in the MCR register request a transition to halt mode. When the CAN controller enters halt mode, the <HMA> bit in the GSR register is set to 1. In halt mode, the CAN controller transmits or receives no messages but it is still active on the CAN bus and can transmit error flags and acknowledge signals. Resetting the MCR<HMR> bit to 0 causes the CAN controller to exit from halt mode.

If halt mode is requested (MCR<HMR>=1) when the CAN controller is transmitting a message, it enters the halt mode after either condition as follows:

- The CAN controller completes the transmission successfully.

- The CAN controller detects arbitration lost.

(4) Test loopback mode

In this mode, the CAN controller receives a message it has transmitted and also generates acknowledge signals. This mode requires only connection to the RX and TX pins and no other CAN devices. The CAN controller transmits a message from one mailbox and receives it to another mailbox. Mailbox settings are the same as those used in normal operation mode.

Test loopback mode can only be enabled or disabled in configuration mode. See the flowchart for setting test loopback and test error modes in Figure 3.12.37.

(5) Test error mode

The error counter can be written in this mode only. The values of the lower eight bits are written to both TEC and REC simultaneously. The maximum value that can be written is 255. A count value of 256, which places the CAN controller in the bus-off mode, cannot be written.

Test error mode can only be enabled or disabled in configuration mode. See the flowchart for setting test loopback and test error modes in Figure 3.12.37.
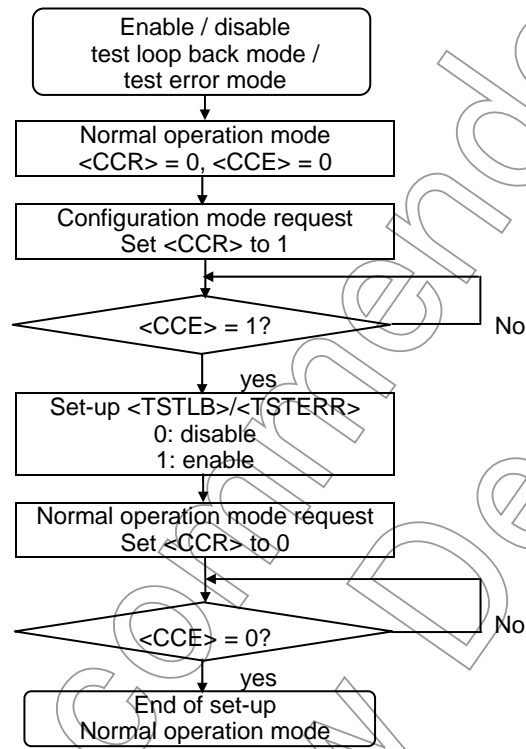
Figure 3.12.37 Flowchart of the Test Loop Back Mode / the Test Error Mode Set-up

3.12.5 Description of operation

（1）Transmission mode

Figure 3.12.38 shows an example message transmission flowchart using a transmission completion interrupt, INTCT.

Polling can also be used in place of an interrupt. In that case, the step "Transmit interrupt occurred?" is replaced with "<TAn> = 1?" and the steps "Write 1 to <MBIMn>" and "Clear <MBTIFn>" are not required.
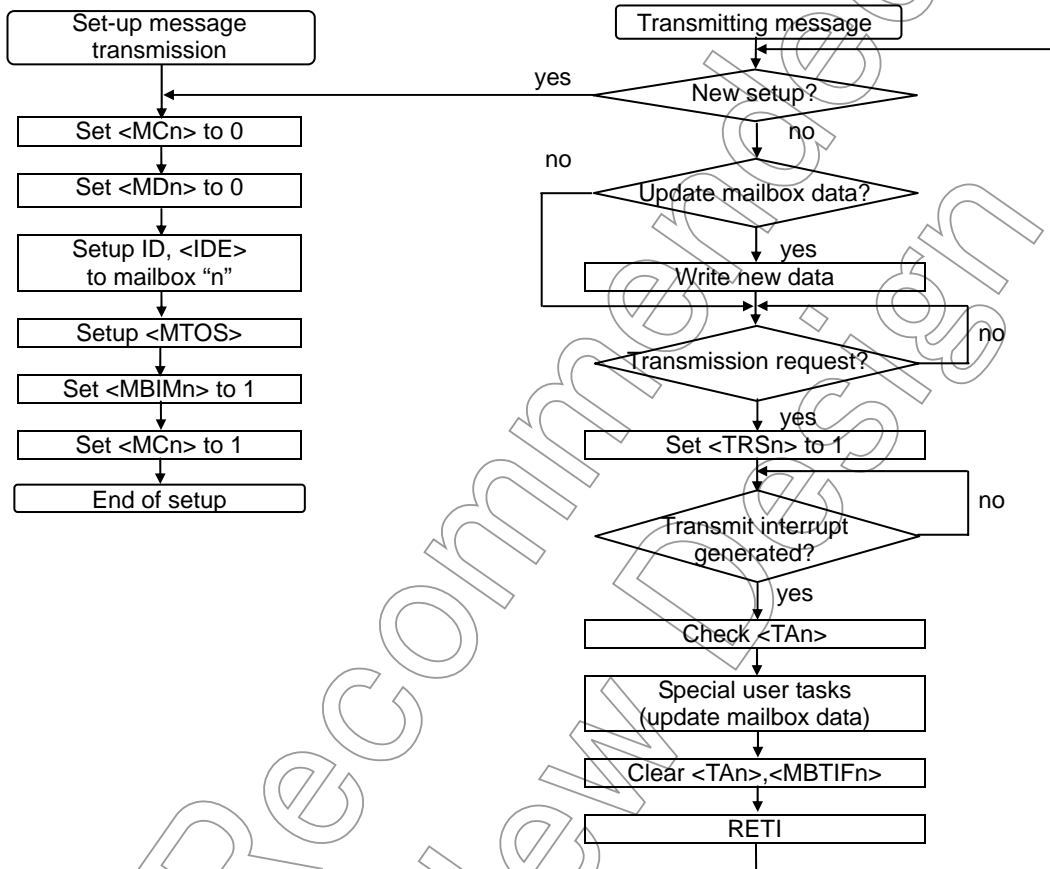
Figure 3.12.38 Flowchart of Message Transmission (Example)

(2) Reception mode

When the CAN controller receives a message on the CAN bus, it stores the message into the receive buffer. The ID of the message stored in the receive buffer is compared with the IDs of mailboxes. If the MBnMI0H<GAME>/<LAME> bit is set to 1, they are compared using the global/local receive mask register, GAM/LAM. If one of the following conditions is satisfied, subsequent IDs are not compared:

- A data frame matches the ID of the receive mailbox.
- A remote frame matches the ID of the receive mailbox.
- A remote frame matches the ID of the transmit mailbox for which the <RFH> bit is set to 1.

The minimum time between the RMP<RMPn> bit being set to 1 and a next receive message being stored into a mailbox depends on the bit time setting. If the data length code is 0, the minimum time is as follows:

- Standard format: 47-bit time – 16 f_{IO}
- Extended format: 67-bit time – 16 f_{IO}

a   Data frame

Figure 3.12.39 shows an example message reception flowchart using a reception completion interrupt, INTCR. Polling can also be used in place of an interrupt. In that case, the step "Receive interrupt occurred?" is replaced with "<RMPn> = 1?" and the steps "Write 1 to <MBIMn>" and "Clear <MBRIFn>" are not required.
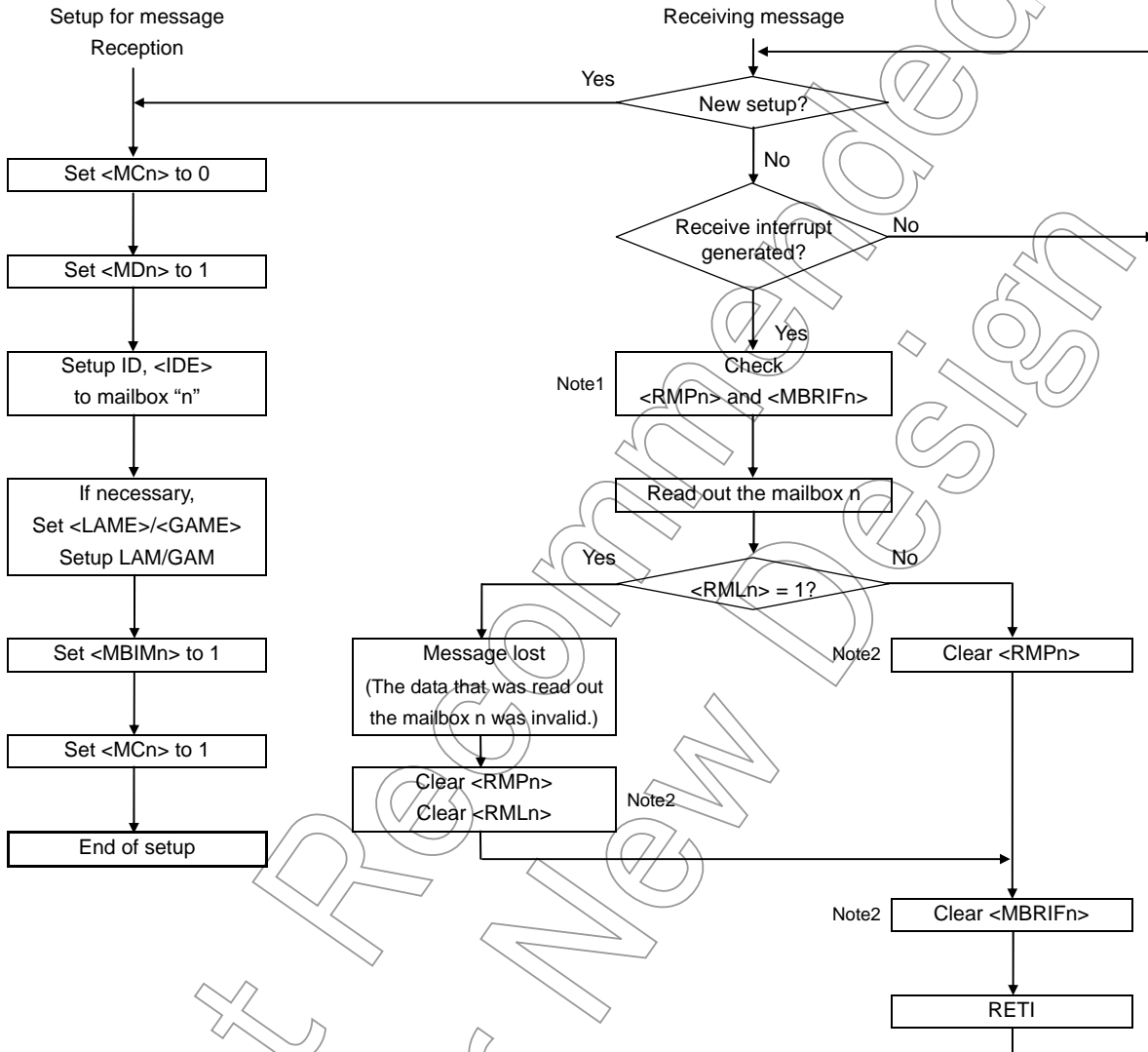
Note 1: Always check <RMPn> and <MBRIFn>.

Note 2: After the step "Clear <RMPn>", if a message is received in mailbox n before <MBRIFn> is cleared, <RMPn> may be set back to 1 (<MBRIFn> = 0).

Figure 3.12.39 Flowchart of Message Reception (Example)

b   Remote frame

Figure 3.12.40 shows an example flowchart for processing a remote frame using the automatic response function. This function is enabled when the <RFH> bit for a transmit mailbox is set to 1. To prevent a data mismatch, update data in the mailbox by using the CDR register to control transmission.
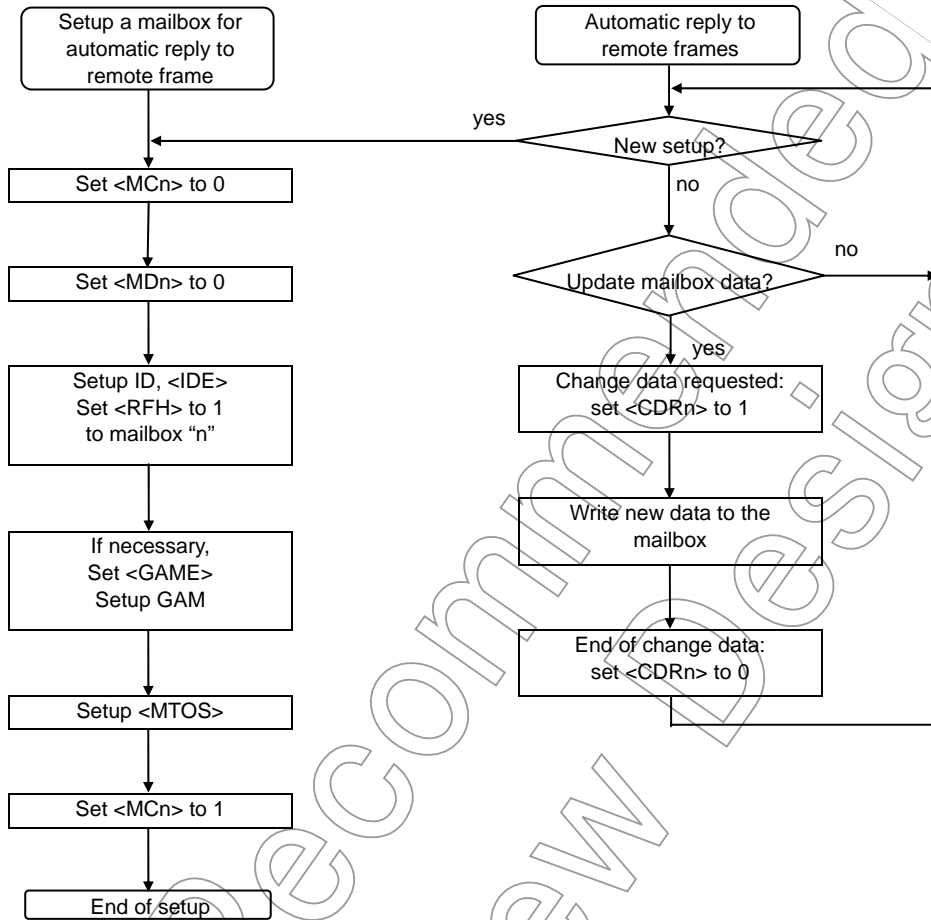
Figure 3.12.40 Flowchart of Remote Frame Handling with the Automatic Reply Feature (Example)

3.13 Analog-to-Digital Converter

The TMP92CD54I incorporates a 10-bit successive approximation analog-to-digital converter (AD converter) with 12 analog input channels.

The following shows a block diagram of the AD converter.

The 12 analog input pins (AN0-AN11) are shared with input-only ports G and L so that they can also be used as input ports.

Note: To reduce supply current in IDLE2, IDLE1, IDLE3, or STOP mode, ensure that the AD converter is not operating before attempting to execute the HALT instruction because the device may enter a standby mode with the internal comparator still enabled depending on the timing.
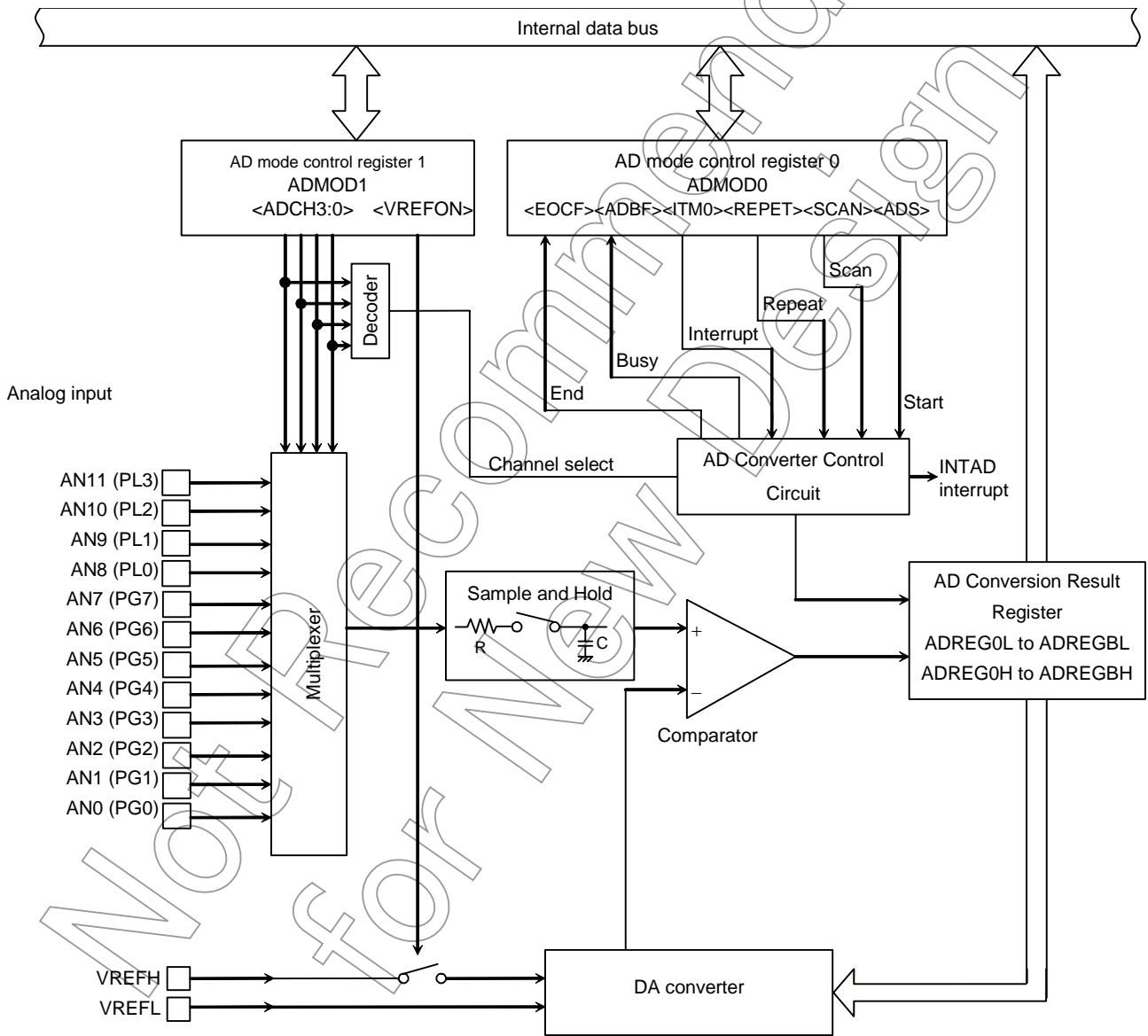
Figure 3.13.1 Block Diagram of AD Converter

3.13.1 Analog-to-digital converter registers

The AD converter is controlled using two AD mode control registers (ADMOD0 and ADMOD1). The results of AD conversion are stored in 12 pairs of AD conversion result upper/lower registers (ADREG0H/L to ADREGBH/L). The following describes 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/Write	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	AD 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: Single Conversion Mode 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 Mode
1	AD Conversion Channel Scan Mode

AD repeat mode setting

0	AD Single Conversion Mode
1	AD Repeat Conversion Mode

Specify AD conversion interrupt for Channel Fixed Repeat Conversion Mode

	Channel Fixed Repeat Conversion Mode <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 complete

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/Write	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

<ADCH3:0> \ <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
1100, 1101, 1110, 1111	(Reserved)	(Reserved)

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/Write	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/Write	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/Write	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/Write	R							
	After Reset	Undefined							
	Function	Stores upper eight bits of AD conversion result.							
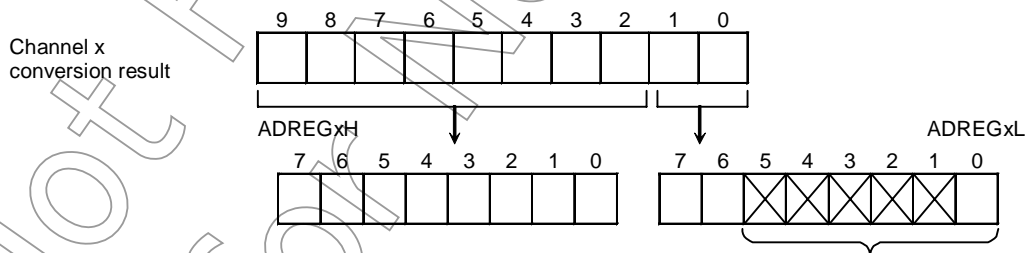
- Bits 5 to 1 are always read as 1.
- Bit 0 is the AD conversion storage flag <ADRxRF>. It is set to 1 when a AD conversion value has been stored. Reading either of the registers (ADREGxH or ADREGxL) causes the corresponding flag to be creared to 0.

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/Write	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/Write	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/Write	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/Write	R							
	After Reset	Undefined							
	Function	Stores upper eight bits of AD conversion result.							

Channel x conversion result
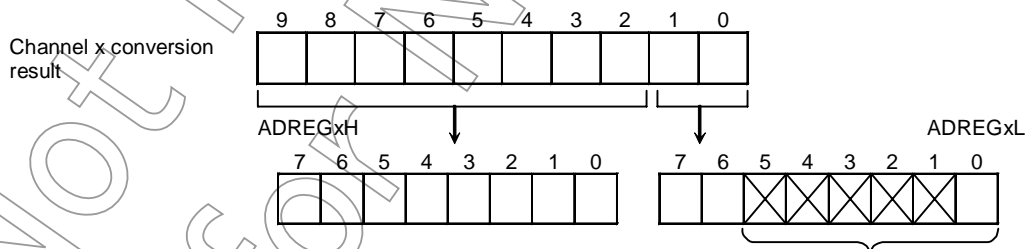
- Bits 5 to 1 are always read as 1.
- Bit 0 is the AD conversion result storage flag <ADRxRF>. It is set to 1 when a AD conversion value has been stored. Reading either of the registers (ADREGxH or ADREGxL) causes the corresponding flag to be cleared to 0.

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/Write	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/Write	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/Write	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/Write	R							
	After Reset	Undefined							
	Function	Stores upper eight bits of AD conversion result.							
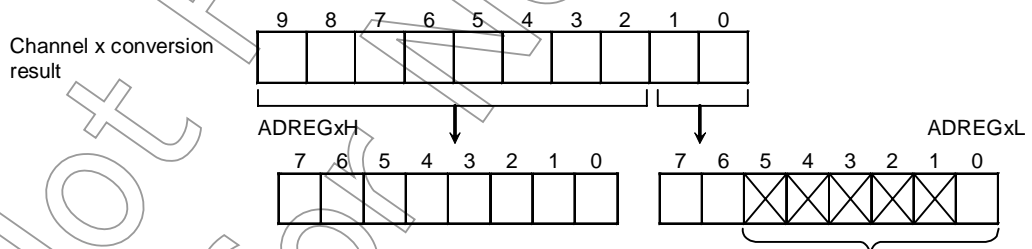
- Bits 5 to1 are always read as 1.
- Bit 0 is the AD conversion result storage flag <ADRxRF>. It is set to 1 when a AD conversion value has been stored. Reading either of the registers (ADREGxH or ADREGxL) causes the corresponding flag to be cleared to 0.

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/Write	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/Write	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/Write	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/Write	R							
	After Reset	Undefined							
	Function	Stores upper eight bits of AD conversion result.							
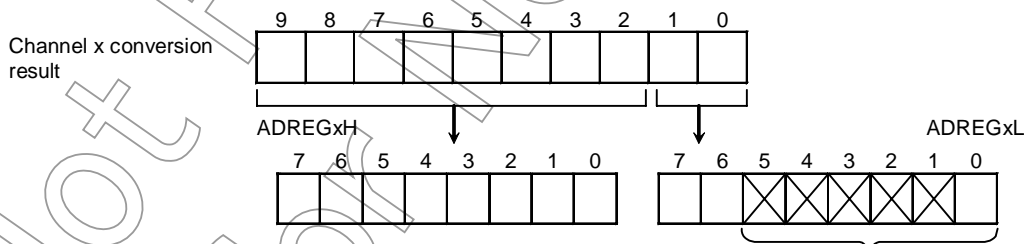
- Bits 5 to 1 are always read as 1.
- Bit 0 is the AD conversion result storage flag <ADRxRF>. It is set to 1 when a AD conversion value has been stored. Reading either of the registers (ADREGxH or ADREGxL) causes the corresponding flag to be cleared to 0.

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/Write	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/Write	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/Write	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/Write	R							
	After Reset	Undefined							
	Function	Stores upper eight bits of AD conversion result.							

Channel x conversion result
9 8 7 6 5 4 3 2 1 0

ADREGxH
7 6 5 4 3 2 1 0
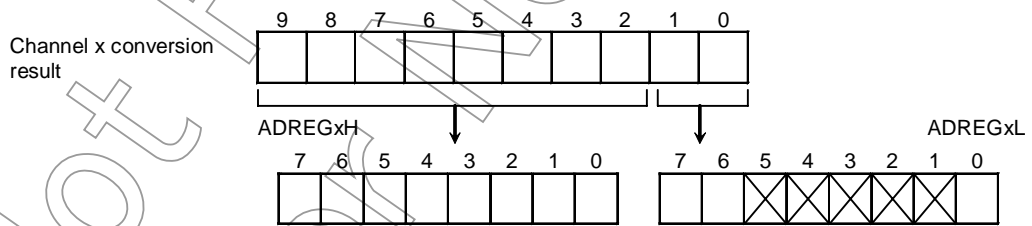
ADREGxL
7 6 5 4 3 2 1 0

- Bits 5 to 1 are always read as 1.
- Bit 0 is the AD conversion result storage flag <ADRxRF>. It is set to 1 when a AD conversion value has been stored. Reading either of the registers (ADREGxH or ADREGxL) causes the corresponding flag to be cleared to 0.

Figure 3.13.8 AD Converter Related Registers

AD Conversion Result Register A Low

ADREGAL (0134H)		7	6	5	4	3	2	1	0
	Bit symbol	ADRA1	ADRA0						ADRARF
	Read/Write	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 A High

ADREGAH (0135H)		7	6	5	4	3	2	1	0
	Bit symbol	ADRA9	ADRA8	ADRA7	ADRA6	ADRA5	ADRA4	ADRA3	ADRA2
	Read/Write	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/Write	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/Write	R							
	After Reset	Undefined							
	Function	Stores upper eight bits of AD conversion result.							
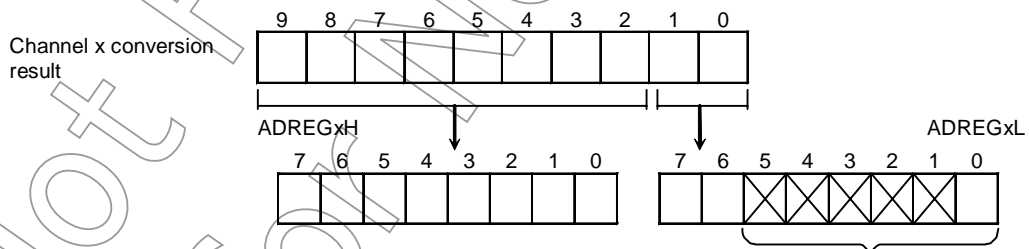
- Bits 5 to 1 are always read as 1.
- Bit 0 is the AD conversion result storage flag <ADRxRF>. It is set to 1 when a AD conversion value has been stored. Reading either of the registers (ADREGxH or ADREGxL) causes the corresponding flag to be cleared to 0.

Figure 3.13.9 AD Converter Related Registers

3.13.2　Description of operation

（1）Analog reference voltage

The high level of the analog reference voltage is applied to the VREFH pin and the low level applied to the VREFL pin. The reference voltage across VREFH and VREFL is divided by 1024 using string resistors. The divided voltages are compared with the analog input voltage to perform AD conversion.

Writing a 0 to the ADMOD1<VREFON> bit causes the switch between VREFH and VREFL to be turned off. To start AD conversion when the switch is turned off, first write a 1 to <VREFON>, then wait for 3 μs (independent of the system clock frequency fc) until the internal reference voltage settles before writing a 1 to ADMOD0<ADS>.

（2）Selecting an analog input channel

How to select an analog input channel depends on the AD converter operating mode.

- When using a fixed analog input channel (ADMOD0<SCAN> = 0)
  Use settings in ADMOD1<ADCH3:0 > to select one of the AN0 to AN11 analog input pins.

- When scanning through analog input channels (ADMOD0<SCAN> = 1)
  Use settings in ADMOD1<ADCH3:0> to select one of the 12 scan modes.

Table 3.13.1 shows the selection of analog input channels in each operating mode.

Upon a reset, ADMOD0<SCAN> and ADMOD1<ADCH3:0> are initialized to 0 and 0000, respectively, so that channel fixed input using the AN0 pin is selected. Pins that are not used as an analog input channel can be used as ordinary input ports. (See "3.5.7 Port G" and "3.5.8 Port L.")

Table 3.13.1 Analog Input Channel Selection

<ADCH3:0>	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
1100~1111	Invalid	Invalid

(3) Starting AD conversion

Setting ADMOD0<ADS> to 1 starts AD conversion. Once AD conversion has started, the AD conversion BUSY flag (ADMOD0<ADBF>) is set to 1, indicating that AD conversion is currently in progress.

(4) AD conversion mode and AD conversion end interrupt

The following four AD conversion modes are supported:

- Channel-fixed single conversion mode
- Channel-scanned single conversion mode
- Channel-fixed repetitive conversion mode
- Channel-scanned repetitive conversion mode

The AD conversion mode is selected using AD mode control register 0, ADMOD0<REPEAT, SCAN>.

Upon the completion of AD conversion, an AD conversion end interrupt, INTAD is issued. The ADMOD0<EOCF> bit, which indicates the end of AD conversion, is also set to 1.

a. Channel-fixed single conversion mode

Setting ADMOD0<REPEAT, SCAN> to 00 selects channel-fixed single conversion mode.

In this mode, the AD converter performs conversion only once for the selected single channel. Upon the completion of conversion, ADMOD0<EOCF> is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is issued.

b. Channel-scanned single conversion mode

Setting ADMOD0<REPEAT, SCAN> to 01 selects channel-scanned single conversion mode.

In this mode, the AD converter performs conversion once for each of the selected scan channels. Upon the completion of conversion for all selected channels, ADMOD0<EOCF> is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is issued.

c. Channel-fixed repetitive conversion mode

Setting ADMOD0<REPEAT, SCAN> to 10 selects channel-fixed repetitive conversion mode.

In this mode, the AD converter repeatedly performs conversion for the selected single channel. Upon the completion of conversion, ADMOD0<EOCF> is set to 1. ADMOD0<ADBF> is not, however, cleared to 0 and maintains the state of 1. The INTAD interrupt request timing can be selected using the setting of ADMOD0<ITM0>.

Setting <ITM0> to 0 causes an interrupt request to be issued upon the completion of every single AD conversion. Setting <ITM0> to 1 causes an interrupt request to be issued upon the completion of every four AD conversions.

d. Channel-scanned repetitive conversion mode

Setting ADMOD0<REPEAT, SCAN> to 11 selects channel-scanned repetitive conversion mode.

In this mode, the AD converter repeatedly performs conversion for the selected scan channels. Upon the completion of a single conversion, ADMOD0<EOCF> is set to 1 and an INTAD interrupt request is issued. ADMOD0<ADBF> is not cleared to 0 and maintains the state of 1.

To stop operation in a repetitive conversion mode (c or d), write a 0 to ADMOD0<REPEAT>. Once the conversion currently being executed is completed, the repetitive conversion mode terminates and ADMOD0<ADBF> is cleared to 0.

When ADMOD1<I2AD> is cleared to zero, causing a transition to halt mode, the AD converter immediately stops operation even if AD conversion is still in progress. When the AD converter exits from a halt, it starts AD conversion from the beginning if it operates in a repetitive conversion mode (c or d). In a single conversion mode (a or b), it does not restart conversion (remains stopped).

Table 3.13.2 shows the relationship between the AD conversion mode and the occurrence of an interrupt request.

Table 3.13.2 Relationship Between AD Conversion Modes and Interrupt Requests

Mode	Interrupt Request Generation	ADMOD0		
		<ITM0>	<REPET>	<SCAN>
Channel Fixed Single Conversion Mode	After completion of conversion	X	0	0
Channel Scan Single Conversion Mode	After completion of scan conversion	X	0	1
Channel Fixed Repeat Conversion Mode	Every conversion	0	1	0
	Every forth conversion	1		
Channel Scan Repeat Conversion Mode	After completion of every scan conversion	X	1	1

X: Don't care

(5) AD conversion time

An AD conversion for a single channel requires 160 states (8 μs when f_c = 20 MHz).

(6) Storing and reading the results of AD conversion

The results of AD conversion are stored in the AD conversion result upper/lower registers (ADREG0H/L to ADREGBH/L), which are read-only.

In channel-fixed repetitive conversion mode, the results of AD conversion are stored sequentially in ADREG0H/L through ADREG3H/L. In other modes, the results of conversion for channels AN0 to AN11 are stored in ADREG0H/L to ADREGBH/L, respectively.

Table 3.13.3 shows the correspondence between the analog input channels and the AD conversion result registers.

Table 3.13.3 Correspondence Between Analog Input Channels and
AD Conversion Result Registers

Analog input channel (PortG/PortL)	AD Conversion Result Register	
	Conversion modes other than at right	Channel fixed repeat conversion mode (every 4 th conversion)
AN0	ADREG0H/L	
AN1	ADREG1H/L	
AN2	ADREG2H/L	
AN3	ADREG3H/L	ADREG0H/L
AN4	ADREG4H/L	↓
AN5	ADREG5H/L	ADREG1H/L
AN6	ADREG6H/L	↓
AN7	ADREG7H/L	ADREG2H/L
AN8	ADREG8H/L	↓
AN9	ADREG9H/L	ADREG3H/L
AN10	ADREGAH/L	
AN11	ADREGBH/L	

The AD conversion result storage flag, ADREGxL<ADRxRF>, is bit 0 in the AD conversion result lower register and indicates whether the corresponding AD conversion result registers have been read. This flag is set to 1 when a converted value is stored into the AD conversion result registers and cleared to 0 when either of the AD conversion result registers (ADREGxH or ADREGxL) is read.

Reading the results of AD conversion causes the AD conversion end flag, ADMOD0<EOCF>, to be cleared to 0.

Example settings:

a. When performing AD conversion for analog input voltage on the AN3 pin and using
    the AD interrupt (INTAD) handling routine to write the converted value to memory
    address 0800H

```
Settings in main routine
                7 6 5 4 3 2 1 0
INTE0AD  ← 1 1 0 0 - - - -        Enable INTAD and set the interrupt level to 4.
ADMOD1   ← 1 1 0 0 0 0 1 1        Set the analog input channel to AN3.
ADMOD0   ← X X 0 0 0 0 0 1        Start conversion in channel-fixed single conversion mode.

 Example processing in interrupt routine
WA       ← ADREG3                 Read the values of ADREG3L and ADREG3H into general
                                  register WA (16 bits).
WA       > > 6                    Shift the contents of WA six times to the right and pad the upper
                                  bits with 0s.
(0800H)  ← WA                     Write the contents of WA to address 0800H.
```

b. When continuously performing AD conversion for analog input voltages on three pins,
    AN0 to AN2, in channel-scanned repetitive conversion mode

```
INTE0AD  ← 1 0 0 0 - - - -        Disable INTAD.
ADMOD1   ← 1 1 0 0 0 0 1 0        Set the analog input channels to AN0-AN2.
ADMOD0   ← X X 0 0 0 1 1 1        Start conversion in channel-scanned repetitive conversion
                                  mode.
```

X = Don't care   "-" = No change

3.14 Watchdog Timer (Runaway Detection Timer)

The TMP92CD54I contains a watchdog timer for runaway detection.

The watchdog timer (WDT) is designed to detect any malfunction (runaway) of the CPU due to noise or for other reasons and help the CPU recover its normal operating status. If the watchdog timer detects a runaway, it issues a nonmaskable INTWD interrupt to notify the CPU.

This watchdog timer output can also be connected to the reset input (within the chip) to forcibly apply a reset.

3.14.1 Configuration

Figure 3.14.1 shows a block diagram of the watchdog timer.
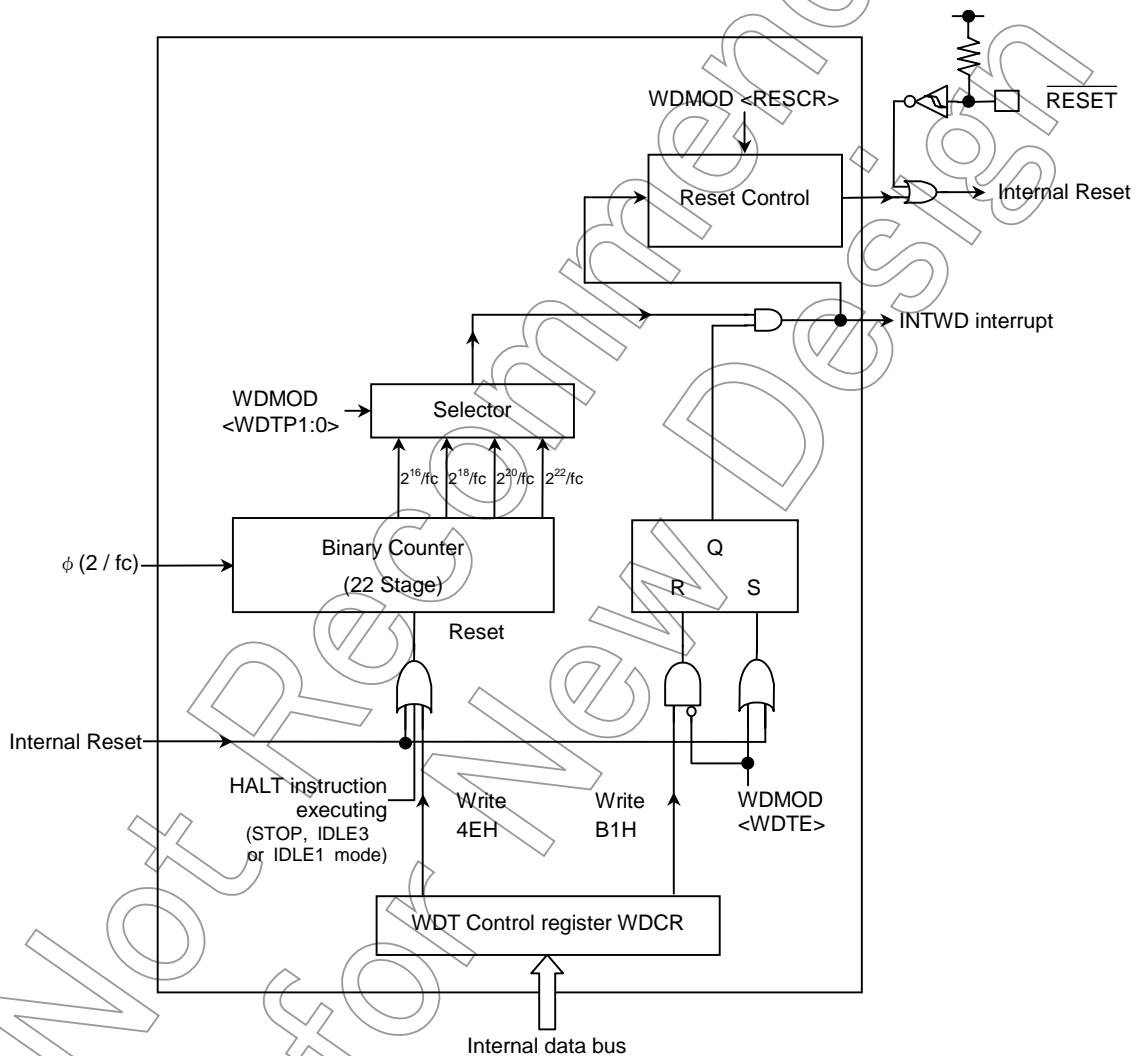
Figure 3.14.1 Block Diagram of Watchdog Timer

The watchdog timer consists of a 22-stage binary counter that uses ϕ (2/fc) as an input clock. The binary counter outputs 2^{16}/fc, 2^{18}/fc, 2^{20}/fc, and 2^{22}/fc. With one of those outputs selected using WDMOD<WDTP1:0>, a watchdog timer interrupt occurs if an overflow occurs for that output, as shown in Figure 3.14.2. To continue using the watchdog timer after an INTWD request is issued, write a clear code (4EH) to the WDCR register to clear the binary counter.
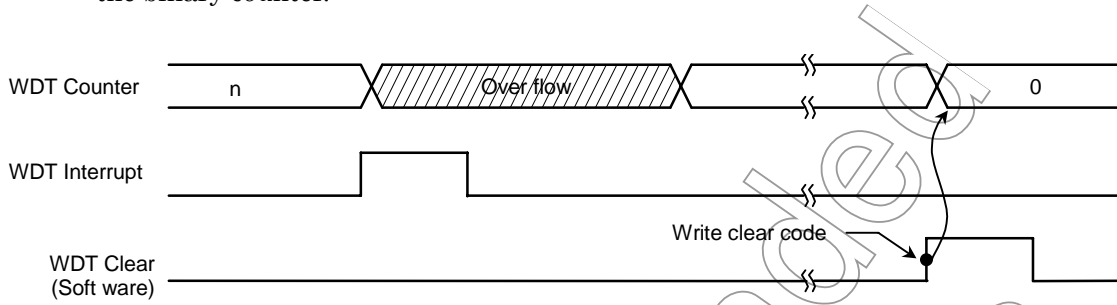
Figure 3.14.2 Normal Mode

The result of runaway detection can also be internally connected to the reset pin.

In that case, a reset is applied for a period of between 44×4/fc and 58×4/fc system clock cycles (8.8 to 11.6 μs when f_C = 20 MHz), as shown in Figure 3.14.3.
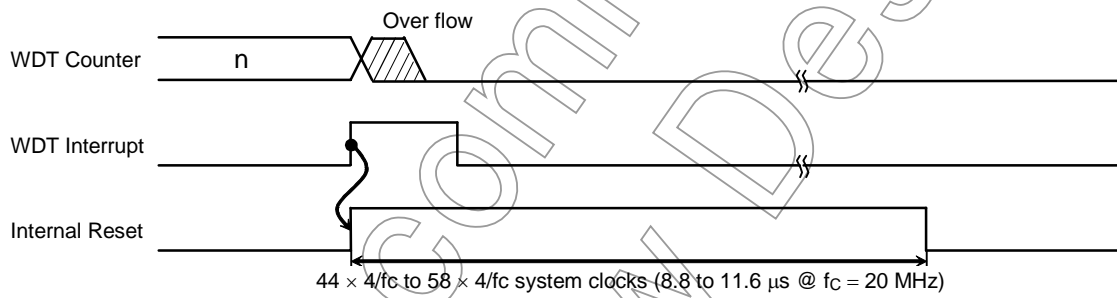
Figure 3.14.3 Reset Mode

3.14.2 Control registers

The watchdog timer (WDT) is controlled using three control registers: WDMOD, WDCR, and CLKMOD.

(1) Watchdog timer mode register (WDMOD)

    a. Setting the watchdog timer detection time <WDTP1:0>

        This 2-bit register specifies a watchdog timer interrupt time for runaway detection. Upon a reset, the WDMOD<WDTP1:0> bits are initialized to 00 so that the detection time is $2^{16}/fc$ [s] (approximately 65,536 system clock cycles).

    b. Enabling/disabling the watchdog timer <WDTE>

        Upon a reset, WDMOD<WDTE> is initialized to 1 so that the watchdog timer is enabled.

        Disabling the watchdog timer requires writing a disable code (B1H) to the WDCR register in addition to clearing this bit to 0. This dual configuration makes it difficult for the watchdog timer to be disabled due to a runaway.

        Enabling the disabled watchdog timer requires only setting the <WDTE> bit to 1.

    c. Connecting the watchdog timer output to a reset <RESCR>

        This register specifies whether the watchdog timer resets itself when it detects a runaway. Upon a reset, WDMOD<RESCR> is initialized to 0 so that the watchdog timer output is not used to reset itself.

(2) Watchdog timer control register (WDCR)

This register controls disabling the watchdog timer and clearing the binary counter.

- Controlling disable
  After clearing WDMOD<WDTE> to 0, writing a disable code (B1H) to the WDCR register disables the watchdog timer.

```
WDMOD    ← 0 - - - - - - -    Clear WDTE to 0.
WDCR     ← 1 0 1 1 0 0 0 1    Write disable code (B1H).
```

- Controlling enable
  Set WDMOD<WDTE> to 1.

- Controlling watchdog timer clear
  Writing a clear code (4EH) to the WDCR register causes the binary counter to be cleared and restart counting. To continue using the watchdog timer after an INTWD interrupt is issued, write a clear code to the WDCR register to clear the binary counter.

```
WDCR     ← 0 1 0 0 1 1 1 0    Write clear code (4EH).
```

(3) Clock mode register (CLKMOD)

This register controls the output signal on the CLK pin.

Writing a 0 to the CLKMOD<CLKOE> bit causes the CLK pin output to be stopped. The output on the CLK pin can be selected from one of fc and 2/5 fc by setting CLKMOD<CLKM1:0>.

The CLKMOD<HALTM1:0> bits specify the halt mode as IDLE2, IDLE1, IDLE3, or STOP.

	7	6	5	4	3	2	1	0
bit symbol	WDTE	WDTP1	WDTP0	-	DRVE	I2WDT	RESCR	-
Read/Write	R/W					R/W		
After reset	1	0	0	-	0	0	0	0
Function	WDT control 0: disable 1: enable	Select detecting time 00: $2^{16}/f_C$ 01: $2^{18}/f_C$ 10: $2^{20}/f_C$ 11: $2^{22}/f_C$			1: Drives pins in STOP mode	IDLE2 0: Stop 1: Operate	1: Internally connects WDT out to the reset pin	Always write 0

WDMOD (0110H)

Watchdog timer out control

0	-
1	Connects WDT out to a reset

IDLE2 control

0	Stop
1	Operation

Watchdog timer detection time

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 3.14.4 Watchdog Timer Mode Register

	7	6	5	4	3	2	1	0
Bit symbol	–							
Read/Write	W							
After reset	–							
Function	B1H: WDT disable code 4EH: WDT clear code							

WDCR (0111H)

WDT disable/clear control

B1H	Disable code
4EH	Clear code
Others	Don't care
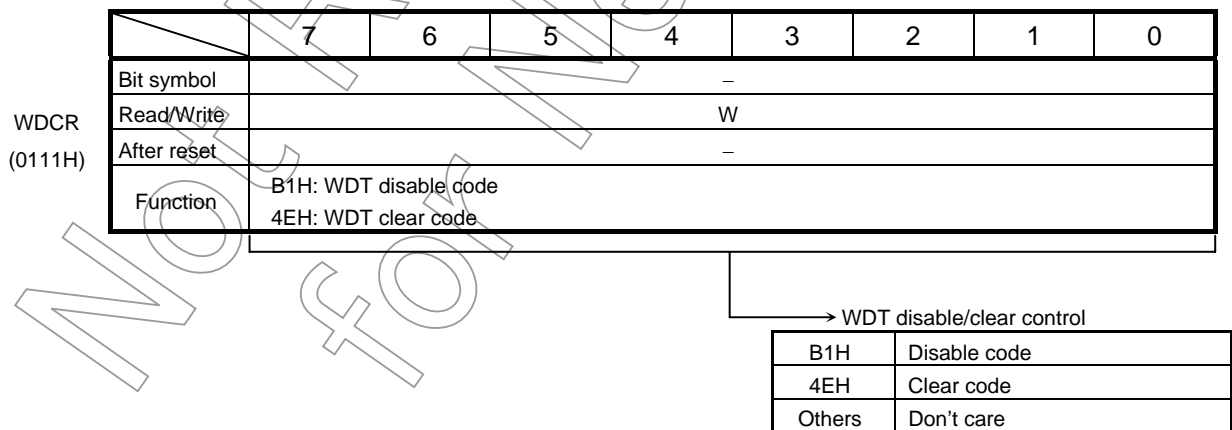
Figure 3.14.5 Watchdog Timer Control Register

CLKMOD (010AH)		7	6	5	4	3	2	1	0
	bit Symbol	HALTM1	HALTM0	-	-	-	CLKOE	CLKM1	CLKM0
	Read/Write	R/W			R/W		R/W		
	After reset	1	1	-	0	-	0	0	0
	Function	Halt mode 00: IDLE3 01: STOP 10: IDLE1 11: IDLE2			Fixed to "0"		CLKoutput enable 0: not output 1: output	CLK output select 00: fc 01: Reserved 10: 2/5 fc 11: Reserved	

CLK output clock select

00	fc
01	Reserved
10	2/5 fc
11	Reserved

CLK output enable

0	No output (High-Z, Pulled up)
1	Output

Selects standby mode by HALT instruction

00	IDLE3
01	STOP
10	IDLE1
11	IDLE2

Figure 3.14.6 Clock Mode Register

3.14.3　Description of operation

The watchdog timer issues an INTWD interrupt when the detection time specified with the WDMOD<WDTP1:0> bits has elapsed. The software (instruction) should clear the binary counter for the watchdog timer to 0 before an INTWD interrupt occurs. If the CPU is malfunctioning due to noise or for other reasons (runaway), it fails to execute an instruction for clearing the binary counter, which will overflow and cause an INTWD interrupt to occur. Once an INTWD interrupt occurs, indicating that the CPU is malfunctioning (runaway), the runaway handling program can restore it to normal condition.

The watchdog timer starts operation immediately after a reset is released.

In IDLE1, IDLE3, or STOP mode, the watchdog timer is reset and stopped.

In IDLE2 mode, its state depends on the setting of WDMOD<I2WDT>. Set WDMOD<I2WDT>, as required, before entering IDLE2 mode.

Example:　a. Clear the binary counter.

a.　Clear the binary counter.

```
WDCR    ← 0 1 0 0 1 1 1 0      Write clear code (4EH).
```

b.　Set the watchdog timer detection time to $2^{18}/fC$.

```
WDMOD   ← 1 0 1 - - - - -
```

c.　Disable the watchdog timer.

```
WDMOD   ← 0 - - X - - - -      Clear <WDTE> to 0.
WDCR    ← 1 0 1 1 0 0 0 1      Write disable code (B1H).
```

3.15 RAM Controller

The RAM controller enables/disables writes to the built-in RAM and detects a low supply voltage to DVCC3. DVCC3 is a voltage supplied to the built-in RAM and internal logic. If DVCC3 falls below the VSTB level, the built-in RAM may not be able to maintain data.

The RAMCR<RAMSTB> flag, which detects a low voltage, is always written with a 1. A write of 0 to this flag is invalid. The flag is cleared to 0 if DVCC3 falls below the VSTB level (including a power-on reset). It is not cleared by a transition to halt mode or a warm reset.

This flag can be read to determine a reset status (warm reset or power-on reset) and RAM data status (maintained or lost). The flag returns a 1 for a warm reset and a 0 for a power-on reset. It returns a 1 when RAM data is maintained and a 0 if it may be lost.

The <RAMWI> bit controls data writes to the built-in RAM. Upon a reset, <RAMWI> is set to 1 so that writes to the built-in RAM are enabled. Clearing <RAMWI> to 0 disables writes to the built-in RAM.

RAMCR
(016DH)

		7	6	5	4	3	2	1	0
	Bit Symbol	RAMSTB	RAMWI	-	-	-	-	-	-
	Read/Write	R/W							
	After reset	0 *Note1	1	-	-	-	-	-	-
	Function	0:lost data or Power on reset 1:kept data	Internal RAM write 0:Inhibit 1:accept						

Write control 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 DVCC3 ≤ VSTB. After power on reset.
1	After "1" is set by software, this data isn't changed at DVCC3 > VSTB.

Note 1: It is initialized to 0 upon a power-on reset but not affected by a warm reset. The software should first write a 1 to the flag before using it. A write of 0 to this flag is invalid.

Note 2: If the device enters a halt mode (STOP/IDLE3) with <RAMSTB> set to 1, current consumption is not sufficiently reduced due to a current that flows through resistance within the voltage detection circuit. In a system for which low power dissipation is required, the voltage detection circuit can be disabled to suppress current consumption.

Note 3: A period of eight states is required between a 1 being written to <RAMSTB> and the voltage detection circuit starting operation (when fc = 20 MHz). Do not execute the HALT instruction during a warm-up period of the voltage detection circuit.

Note 4: The emulator does not support the RAM controller function.

Figure 3.15.1   RAM Control Register

3.16 Real-Time Clock (RTC)

The TMP92CD54I contains a real-time clock, which is dedicated to measuring a specified time. The real-time clock issues INTRTC interrupts at regular intervals. The interrupt interval can be selected from among 0.0625 s, 0.125 s, 0.25 s, 0.50 s, 1 s, and 2 s (when fs = 32.768 kHz).

The TMP92CD54I supports a low current consumption mode in which only the real-time clock operates, called IDLE3 mode. It also operates in IDLE1 and IDLE2 modes and can release each hold mode upon the occurrence of an INTRTC interrupt request.

3.16.1 Block diagram
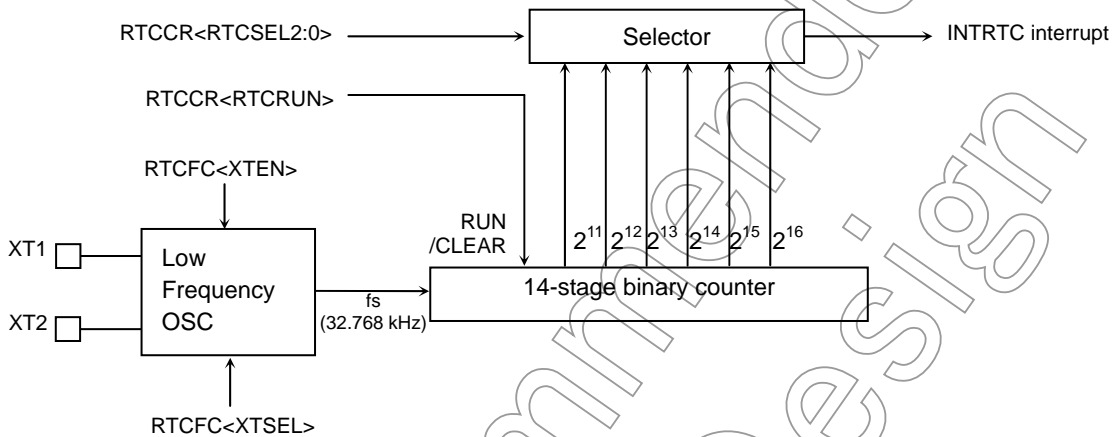
Figure 3.16.1 Block Diagram for Timer for Real-time Clock

3.16.2 Registers

Two registers are provided to control the real-time clock and low-speed oscillator.

The real-time clock control register, RTCCR, controls the real-time clock. The RTCCR <RTCSEL2:0> bits specify one of six intervals for INTRTC interrupt requests.

The real-time clock function register, RTCFC, controls the low-speed oscillator. Either a crystal or CR oscillator can be used for the low-speed oscillator. Set RTCFC <XTSEL> according to the oscillator to be used.

The RTCFC register is initialized when the device recovers from STOP mode with an interrupt. It is, therefore, necessary to re-set RTCFC after a halt release. (The RTCFC register is not initialized upon a recovery from IDLE3, IDLE1, or IDLE2 mode.)

Figure 3.16.2 and Figure 3.16.3 show register tables.

Timer for Real Time Clock Control Register

RTCCR (118H)		7	6	5	4	3	2	1	0
	Bit symbol	-	-	-	-	RTCSEL2	RTCSEL1	RTCSEL0	RTCRUN
	Read/Write	R/W				R/W			R/W
	After Reset	0	-	-	-	0	0	0	0
	Function	Write 0				$1x0:2^{16}/fs$ (2s) $1x1:2^{15}/fs$ (1s) x:Don't care	$000:2^{14}/fs(0.50s)$ $001:2^{13}/fs(0.25s)$ $010:2^{12}/fs(0.125s)$ $011:2^{11}/fs(0.0625s)$		0: Stop& Clear 1: Run

Counting operation

0	Stop & Clear
1	Count

Interrupt generation cycle (fs = 32.768 kHz)

000	0.50s
001	0.25s
010	0.125s
011	0.0625s
1x0	2s
1x1	1s

Figure 3.16.2 Timer for Real Time Clock Control Register

Timer for Real Time Clock Function Register

RTCFC (119H)		7	6	5	4	3	2	1	0
	Bit symbol	XTSEL	-	-	-	-	-	-	XTEN
	Read/Write	R/W							R/W
	After Reset	0	-	-	-	-	-	-	0
	Function	Type of low frequency oscillator(fs) 0: Crystal 1: CR							Low Frequency oscillator (fs) 0:Stop 1:Oscillation

Low frequency oscillator (fs=32.768 kHz)

0	Stop
1	Oscillation
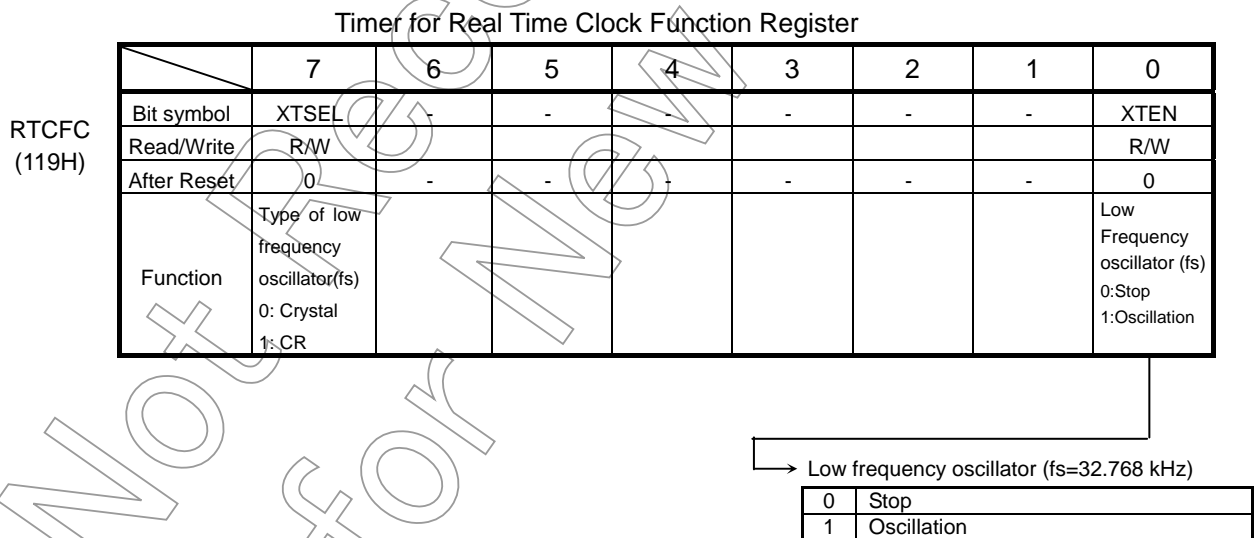
Note 1: Setting RTCFC<XTEN> to 1 causes the low-speed oscillator to start oscillation but it requires a wait time until oscillation is stabilized. For the value of tSTA for a crystal resonator, contact the manufacturer of the resonator.

Note 2: This register is initialized when the device recovers from STOP mode with an interrupt. It is, therefore, necessary to re-set the register after a halt release. (It is not initialized upon a recovery from IDLE3, IDLE1, or IDLE2 mode.)

Figure 3.16.3 Timer for Real Time Clock Function Register

Example of register setting:

```
LD   (RTCFC), 01h       ; Start low-speed oscillation.
         :              ; Oscillator stabilization time
LD   (RTCCR), 03h       ; INTRTC interrupt occurs every 2^13/fs.
```

3.16.3 CR oscillation

Either a crystal or CR oscillator can be used for the low-speed oscillator. Set RTCFC <XTSEL> according to the oscillator to be used.

When using CR oscillation, connect a resistor and capacitor to the XT1 and XT2 pins.

Figure 3.16.4 shows a recommended CR oscillation circuit.
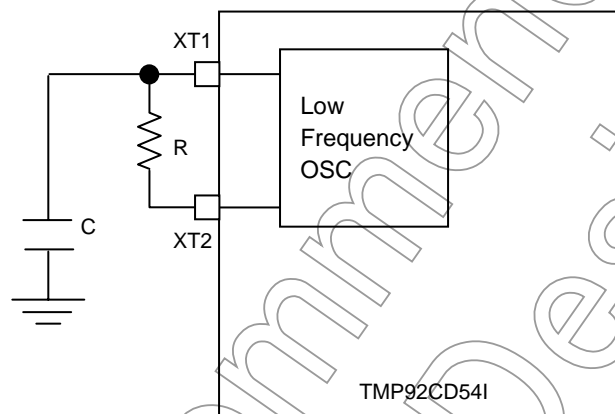
Example constants for 32.768 kHz:
$$R = 40\ k\Omega,\ C = 470\ pF$$
$$R = 82\ k\Omega,\ C = 220\ pF$$

Note: The above combination of constants has been tested under room temperature conditions. The values should be adjusted according to the end product considering the characteristics of the capacitor and resistor.

Figure 3.16.4 A External Circuit for CR Oscillation

3.17 Power Regulator

The TMP92CD54I contains a 3-V output regulator for internal logic power supplies. Connecting each DVCC3 pin to the regulator output pin, REGOUT, enables the regulator to supply power to internal logic circuits.

Table 3.17.1 REGOUT output by REGEN setting

REGEN input	REGOUT output
"H"	3V output for internal logic
"OPEN" Note)	3V output for internal logic

Note 1: The REGEN pin has a pull-up resistor connected internally and thus can be left open. It is recommended to leave it open to ensure a rise time for the REGEN signal.

3.17.1 Block diagram
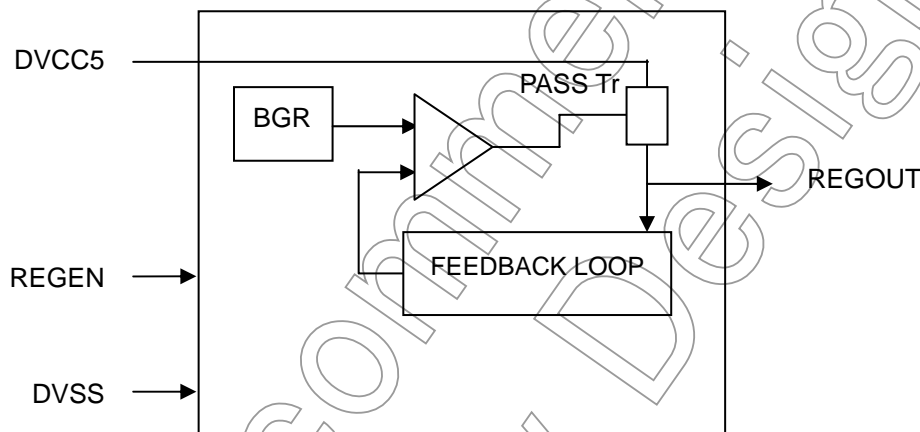
Figure 3.17.1 Regulator Block

3.17.2 External connection

To prevent the output voltage from oscillating, connect a stabilizing capacitor (Cs) to a point between REGOUT and DVSS as close to them possible. Depending on the board capacitance, a resistor in series with Cs (ESR) may also be necessary, as shown in Figure 3.17.2.

It is recommended to use a capacitor having good temperature characteristics because variations in internal resistance with temperature may cause the regulator output to be unstable.

A bypass capacitor (Cb) between DVCC3 and DVSS is also recommended to improve noise immunity of the REGOUT output.
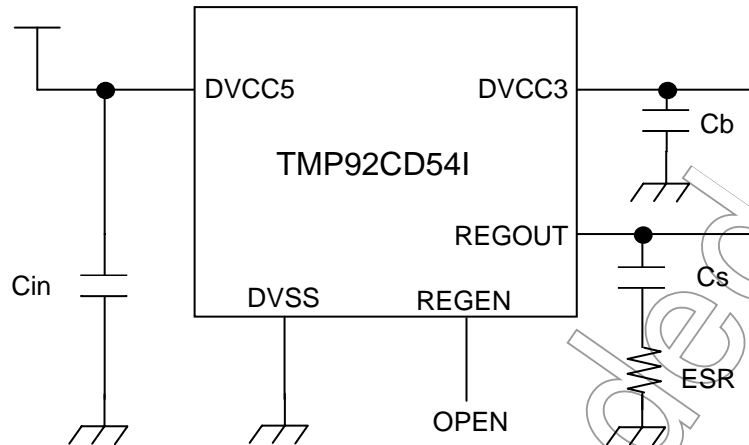
Figure 3.17.2 Regulator Connection

3.17.3 Handling precautions

1. Application

This regulator is designed for the TMP92CD54I. The output from REGOUT must not be connected to anywhere other than the DVCC3 pin on the TMP92CD54I.

2. Power-on and REGEN input signal timing

When the device is powered on, the REGEN pin should be left open or an enable signal (High level) should be input to the pin at least 1 μs after the power-on.

3. Constant settings (Cin, Cs, Cb, ESR)

The characteristics of stray capacitance or parasitic capacitance according to the module configuration may affect the regulator characteristics. When using the device, investigate static and transient characteristics based on the actual operating conditions to set constants with sufficient margins.

4.   Electrical Characteristics

4.1   Absolute Maximum Ratings

The absolute maximum ratings of a semiconductor device are a set of ratings that must not be exceeded, even for a moment. Do not exceed any of these ratings. Exceeding the rating(s) may cause the device breakdown, damage or deterioration, and may result injury by explosion or combustion. The equipment manufacturer should design so that no maximum rating value is exceeded.

Parameter	Symbol	Rating	Unit
Power Supply Voltage	V_{CC5}	−0.5 to 6.0	V
Input Voltage	V_{IN}	−0.5 to VCC5 + 0.5	V
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 to 150	°C
Operation Temperature	T_{OPR}	−40 to 85	°C

Solderability

Test parameter	Test condition	Note
Solderability	(1) Use of Sn-37Pb solder bath Solder bath temperature = 230°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: Solderability rate until forming ≥ 95%
	(2) Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	

4.2　　DC Electrical Characteristics

V_{CC5} = 4.5V to 5.25V / fc = 16 to 20MHz / Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min	Max	Unit
Supply Voltage	V_{CC5}		4.5	5.25	V
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	V_{IL1}		−0.3	$0.3 \times V_{CC5}$	V
Input Low Voltage INT0 $\overline{\text{NMI}}$ $\overline{\text{RESET}}$ P70, P71, P73 to P75 PC0 to PC5 PD0 to PD7 PF0 to PF7 PM0 to PM4	V_{IL2}		−0.3	$0.25 \times V_{CC5}$	V
P72, PN0 to PN6	V_{IL6}		−0.3	$0.3 \times V_{CC5}$	V
Input Low Voltage AM0 to AM1 TEST0 to TEST1	V_{IL3}		−0.3	0.3	V
Input Low Voltage X1, XT1 (Crystal)	V_{IL4}	Vcc3 = 3.3V	−0.3	$0.2 \times VCC3$	V
Input Low Voltage XT1 (CR)	V_{IL5}	Vcc3 = 3.3V	−0.3	$0.2 \times VCC3$	V
Input High Voltage P00 to P07(D0 to 7) PG0 to PG7 PL0 to PL3	V_{IH0}		2.2	$V_{CC5} + 0.3$	V
Input High Voltage P00 to P07 P40 to P47	V_{IH1}		$0.7 \times V_{CC5}$	$V_{CC5} + 0.3$	V
Input High Voltage INT0 $\overline{\text{NMI}}$ $\overline{\text{RESET}}$ P70, P71, P73 to P75 PC0 to PC5 PD0 to PD7 PF0 to PF7 PM0 to PM4	V_{IH2}		$0.75 \times V_{CC5}$	$V_{CC5} + 0.3$	V
P72, PN0 to PN6	V_{IH6}		$0.7 \times V_{CC5}$	$V_{CC5} + 0.3$	V
Input High Voltage AM0 to AM1 TEST0 to TEST1	V_{IH3}		$V_{CC5} - 0.3$	$V_{CC5} + 0.3$	V
Input High Voltage X1, XT1 (Crystal)	V_{IH4}	Vcc3 = 3.3V	$0.8 \times VCC3$	VCC3+0.3	V
Input High Voltage XT1 (CR)	V_{IH5}	Vcc3 = 3.3V	$0.7 \times VCC3$	VCC3+0.3	V

V_{CC5} = 4.5V to 5.25V / fc = 16 to 20MHz / Ta = −40 to 85 °C

Parameter	Symbol	Condition		Min	Max	Unit
Output Low Voltage	V_{OL}	I_{OL} = 3.0 mA			0.4	V
Output High Voltage	V_{OH0}	I_{OH} = −400 μA		2.4		V
	V_{OH1}	I_{OH} = −100 μA		$0.75 \times V_{CC5}$		
	V_{OH2}	I_{OH} = −20 μA		$0.9 \times V_{CC5}$		
	V_{OHn}	I_{OH} = −200 μA, PF6(TX) pin only		$0.82 \times V_{CC5}$		
Input Leakage Current	I_{LI}	$0.0 \le Vin \le V_{CC5}$, Vin: Input voltage		0.02 (typ.)	± 5	μA
Output Leakage Current	I_{LO}	$0.2 \le Vin \le V_{CC5}{-}0.2$, Vin: Input voltage		0.05 (typ.)	± 10	μA
Operating Current (Single Chip) (Note1)	I_{CC5}	V_{CC5}=5.25V , X1=10MHz(Internal 20MHz)		70 (typ)	100	mA
Operating Current (Stand-by) (Note2)	$I_{CC5IDLE2}$	IDLE2 Mode	V_{CC5}=5.25V, X1=10MHz (Internal 20MHz)		90	mA
	$I_{CC5IDLE1}$	IDLE1 Mode	V_{CC5}=5.25V, X1=10MHz (Internal 20MHz)		30	
	$I_{CC5IDLE3}$	IDLE3 Mode	V_{CC5}=5.25V, Ta = −40 to 85 °C		220	μA
			V_{CC5}=5.25V, Ta = −10 to 55 °C		140	
	$I_{CC5STOP}$	STOP Mode	V_{CC5}=5.25V, Ta = −40 to 85 °C		200	μA
			V_{CC5}=5.25V, Ta = −10 to 55 °C		120	
Stand-by Voltage	V_{STB5}	$V_{CC3} < V_{CC5}$, $V_{IH1} < V_{CC5}$, $V_{IH2} < V_{CC5}$, $V_{IH3} < V_{CC5}$		3.0	5.25	V
Pull-up Resistor	R_{RST}	$\overline{RESET}$		60	220	KΩ
	R_{CLK}	CLK				
	R_{REGEN}	REGEN				
Schmitt Width	V_{TH}	INT0, NMI , $\overline{RESET}$, P70 to P75, PC0 to PC5, PD0 to PD7, PF0 to PF7, PM0 to PM4, PN0 to PN6		0.4	1.0 (typ.)	V

Note 1: Value when the external bus is not operating.

Note 2: $I_{CC5IDLE3}$ and $I_{CC5STOP}$ are values when the voltage detection circuit for the RAM controller is not operating. (RAMCR <RAMSTB> = 0)

4.3 AC Electrical Characteristics

Read Cycle

V_{CC5} = 4.5 to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

No.	Parameter	Symbol	Min	Max	20MHz	16MHz	Unit
1	Oscillator frequency (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System clock cycle period (= T)	t_{CYC}	50	62.5	50	62.5	ns
3	CLK pulse width low	t_{CL}	$0.5 \times T - 15$		10	16	ns
4	CLK pulse width high	t_{CH}	$0.5 \times T - 15$		10	16	ns
5-1	A0-A23 transition to D0-D7 data in at 0 wait state	t_{AD}		$2.0 \times T - 50$	50	75	ns
5-2	A0-A23 transition to D0-D7 data in at 1 wait state	t_{AD3}		$3.0 \times T - 50$	100	138	ns
6-1	$\overline{RD}$ asserted to D0-D7 data in at 0 wait state	t_{RD}		$1.5 \times T - 45$	30	49	ns
6-2	$\overline{RD}$ asserted to D0-D7 data in at 1 wait state	t_{RD3}		$2.5 \times T - 45$	80	111	ns
7-1	$\overline{RD}$ pulse width low at 0 wait state	t_{RR}	$1.5 \times T - 20$		55	74	ns
7-2	$\overline{RD}$ pulse width low at 1 wait state	t_{RR3}	$2.5 \times T - 20$		105	136	ns
8	A0-A23 valid to $\overline{RD}$ asserted	t_{AR}	$0.5 \times T - 20$		5	11	ns
9	$\overline{RD}$ asserted to CLK low	t_{RK}	$0.5 \times T - 20$		5	11	ns
10	A0-A23 transition to D0-D7 hold	t_{HA}	0		0	0	ns
11	$\overline{RD}$ negated to D0-D7 hold	t_{HR}	0		0	0	ns
12	$\overline{WAIT}$ setup time	t_{TK}	15		15	15	ns
13	$\overline{WAIT}$ hold time	t_{KT}	5		5	5	ns

Write Cycle

V_{CC5} = 4.5 to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

No.	Parameter	Symbol	Min	Max	20MHz	16MHz	Unit
1	Oscillator frequency (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System clock cycle period	t_{CYC}	50	62.5	50	62.5	ns
3	CLK pulse width low	t_{CL}	$0.5 \times T - 15$		10	16	ns
4	CLK pulse width high	t_{CH}	$0.5 \times T - 15$		10	16	ns
5-1	D0-D7 valid to $\overline{WR}$ negated at 0 wait state	t_{DW}	$1.25 \times T - 35$		28	43	ns
5-2	D0-D7 valid to $\overline{WR}$ negated at 1 wait state	t_{DW3}	$2.25 \times T - 35$		78	106	ns
6-1	$\overline{WR}$ pulse width low at 0 wait state	t_{WW}	$1.25 \times T - 30$		33	48	ns
6-2	$\overline{WR}$ pulse width low at 1 wait state	t_{WW3}	$2.25 \times T - 30$		83	111	ns
7	A0-A23 transition to $\overline{WR}$ asserted	t_{AW}	$0.5 \times T - 20$		5	11	ns
8	$\overline{WR}$ asserted to CLK low	t_{WK}	$0.5 \times T - 20$		5	11	ns
9	$\overline{WR}$ negated to A0-A23 hold	t_{WA}	$0.25 \times T - 5$		8	11	ns
10	$\overline{WR}$ negated to D0-D7 hold	t_{WD}	$0.25 \times T - 5$		8	11	ns
11	$\overline{WAIT}$ setup time	t_{TK}	15		15	15	ns
12	$\overline{WAIT}$ hold time	t_{KT}	5		5	5	ns
13	$\overline{RD}$ negated to D0-D7 out	t_{RDO}	$1.25 \times T - 35$		20	26	ns

AC test conditions:

Output conditions of the D0 to D7, A0 to A7, A8 to A15, A16 to A23, $\overline{RD}$ and $\overline{WR}$ pins:
High = 2.0 V, Low = 0.8 V, CL = 50 pF

Output conditions of pins other than the above-mentioned ones:
High = 2.0 V, Low = 0.8 V, CL = 50 pF

Input conditions of the P00 to P07 (D0 – D7) pins:
High = 2.4 V, Low = 0.45 V, CL = 50 pF

Input conditions of pins other than the above-mentioned ones:
High = $0.8 \times V_{CC5}$, Low = $0.2 \times V_{CC5}$, CL = 50 pF

(1)   Read Cycle Timing (0 Wait State)

Note: The signals other than the X1 signal are derived from the X1 signal. Thus, certain timing delays occur in the generation of these signals. Since these delay times vary depending on each sample device, the phase differences between the X1 signal and the other signals cannot be specified. The phase relationship shown in the above timing diagram is only an example.

(2) Write Cycle Timing (0 Wait State)
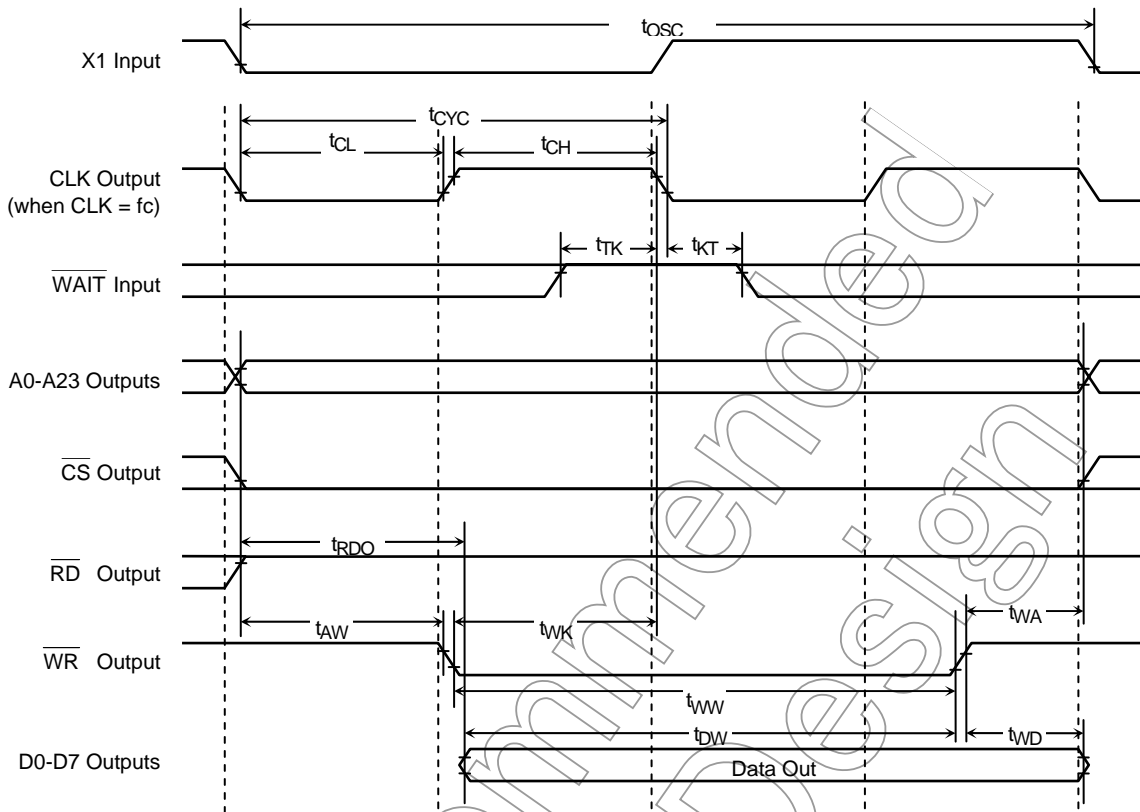
Note: The signals other than the X1 signal are derived from the X1 signal. Thus, certain timing delays occur in the generation of these signals. Since these delay times vary depending on each sample device, the phase differences between the X1 signal and the other signals cannot be specified. The phase relationship shown in the above timing diagram is only an example.

(3) Read Cycle Timing (1 Wait State)

(4) Write Cycle Timing (1 Wait State)
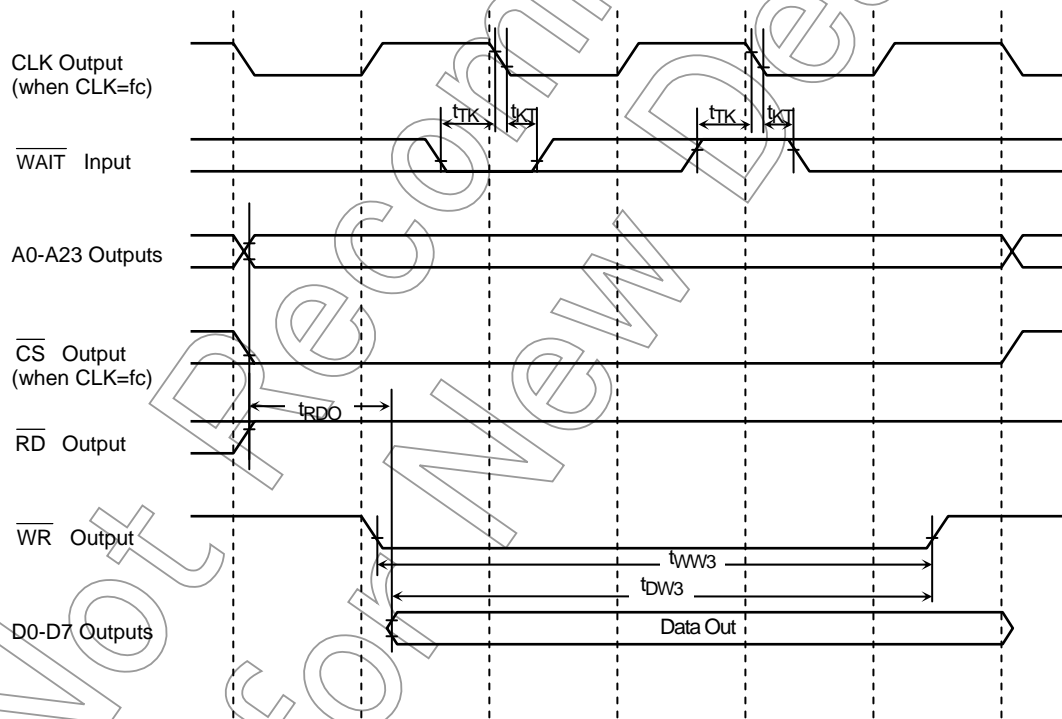
4.4 AD Converter Characteristics

V_{CC5} = 4.5V to 5.25V / fc = 16 to 20MHz / Ta = −40 to 85 °C

Parameter	Symbol	Min	Typ.	Max	Unit
Analog reference voltage (+)	V_{REFH}	$V_{CC5} - 0.2$	V_{CC5}	V_{CC5}	V
Analog reference voltage (−)	V_{REFL}	GND	GND	GND	
Supply voltage for AD converter	AV_{CC}	$V_{CC5} - 0.2$	V_{CC5}	V_{CC5}	
Ground for AD converter	AV_{SS}	GND	GND	GND	
Analog input voltage	AV_{IN}	V_{REFL}		V_{REFH}	
Supply current for analog reference voltage <VREFON> = 1	I_{REF}		0.8	1.2	mA
Supply current for analog reference voltage <VREFON> = 0			0.02	5	µA
Total error (excluding quantization error)	E_T			±3.0	LSB

Note: "LSB" is a unit that represents the resolution of the AD converter.

±3 LSB = 3 × (V_{REFH} - V_{REFL})/1024 ≈ ±15 mV (V_{REFH} = 5.0 V, V_{REFL} = 0.0 V)

4.5 Event Counters (TI0, TI4, TI8, TI9, TIA, TIB)

V_{CC5} = 4.5V to 5.25V / fc = 16 to 20MHz / Ta = −40 to 85 °C

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock cycle period	t_{VCK}	8T + 100		500		600		ns
Clock pulse width low	t_{VCKL}	4T + 40		240		290		ns
Clock pulse width high	t_{VCKH}	4T + 40		240		290		ns

4.6 Serial Channel Timing

(1) SCLK Input mode (I/O Interface mode)

V_{CC5} = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

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}	$t_{SCY}/2 - 4T - 110$		90		140		
SCLK Rise → Output Data Hold	t_{OHS}	$t_{SCY}/2 + 2T$		500		625		ns
SCLK Rise → Input Data Hold	t_{HSR}	3T+10		160		197		
SCLK Rise → Input Data Valid	t_{SRD}		t_{SCY}		800		1000	

(2) SCLK output mode (I/O interface mode)

V_{CC5} = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

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}	$t_{SCY}/2 - 40$		360		460		
SCLK Rise → Output Data Hold	t_{OHS}	$t_{SCY}/2 - 40$		360		460		ns
SCLK Rise → Input Data Hold	t_{HSR}	0		0		0		
SCLK Rise → Input Data Valid	t_{SRD}		$t_{SCY} - T - 180$		570		758	
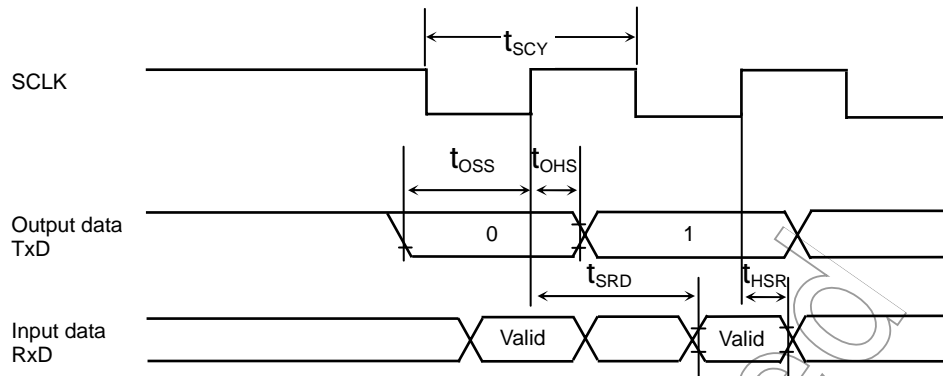
(3) SCLK input mode (UART mode)

V_{CC5} = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Cycle	T_{SCY}	4T + 20		220		270		ns
SCLK Low level Pulse width	T_{SCYL}	2T + 5		105		130		
SCLK High level Pulse width	T_{SCYH}	2T + 5		105		130		

4.7   Interrupt Operation

V_{CC5} = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

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		
WUINT0 to WUINT7, INT1 to INT7 Low Width	T_{INTBL}	8T + 100		500		600		
WUINT0 to WUINT7, INT1 to INT7 High Width	T_{INTBH}	8T + 100		500		600		

4.8 Serial Bus Interface

V_{CC5} = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter	Symbol	fc = 20 MHz						Unit
		400 KHz `<SCK3:0>=1000`		100 KHz `<SCK3:0>=1111`		`<SCK3:0>=0011 to 0110` (Note 2)		
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	f_{SCL}	0	400	0	100	0	$fc/(2^n+8)$	KHz
Hold time (repeated) START condition. After this period, the first clock pulse is generated.	$t_{HD:STA}$	650		4500		$2^{n-1}/fc$		
Low period of the SCL clock	t_{LOW}	1300		4700		$2^{n-1}/fc$		
High period of the SCL clock	t_{HIGH}	600		4000		$(2^{n-1}+8)/fc$		
Set-up time for a repeated START condition	$t_{SU:STA}$	By software		By software		By software		
Data hold time	$t_{HD:DAT}$	0	900	0	3450	0	$6/fc$	
Data set-up time	$t_{SU:DAT}$	100		250		$(2^{n-1}-6)/fc$		ns
Data set-up time (The case in the first bit after transfer)	$t_{SU:1stDAT}$	100		250		$(2^{n-1}-12)/fc$		
Rise time of both SDA and SCL signals (Note 1)	t_r		300 (Receive)		1000 (Receive)		−	
Fall time of both SDA and SCL signals	t_f		300		300		−	
Set-up time for STOP condition	$t_{SU:STO}$	950		4200		$(2^{n-1}+12)/fc$		
Bus free time between a STOP and START condition	t_{BUF}	By software		By software		By software		
Capacitive load for each bus line	C_b		400		400		400	pF
Noise margin at the Low level for each connected device (including hysteresis)	V_{nL}	$0.2 \times V_{CC5}$		$0.2 \times V_{CC5}$		$0.2 \times V_{CC5}$		V
Noise margin at the High level for each connected device (including hysteresis)	V_{nH}	$0.2 \times V_{CC5}$		$0.2 \times V_{CC5}$		$0.2 \times V_{CC5}$		V
Pulse width of spikes which must be suppressed by the input filter	t_{SP}	0	50	n/a	n/a	n/a	n/a	ns

Note 1: The above values are referred to V_{IHmin} and V_{ILmax}.

Note 2: The values for <SCK 3:0> = 0011 to 0110 (n = 8 to 11) include the t_f f and t_r periods.

Note 3: n/a: Not defined

S: Start   P: Stop   Sr: Restart
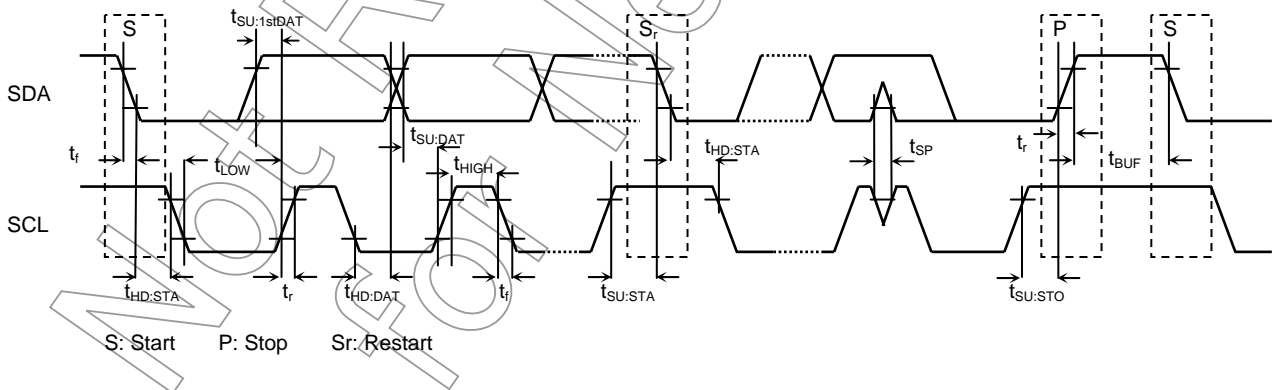
4.9 Serial Expansion Interface (SEI)

V_{CC5} = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Symbol	Parameter	Variable		20MHz		Unit
		Min	Max	Min	Max	
t_{SECLK}	SECLK Cycle	5T	40T	250	2000	ns
t_{LEAD}	SS fall → SECLK	4T		200		ns
t_{LAG}	SECLK → SS rise	4T		200		ns
t_{SCKH}	SECLK High Pulse Width	t_{SECLK}/2-9		116		ns
t_{SCKL}	SECLK Low Pulse Width	t_{SECLK}/2-9		116		ns
t_{SU}	Input Data Set-up	t_{SECLK}/4-10		52		ns
t_H	Input Data Hold	t_{SECLK}/4		62		ns
t_V	Output Data Valid		t_{SECLK}/4		62	ns
t_{HO}	Output Data Hold	0		0		ns

a) SEI master (CPHA = 0)
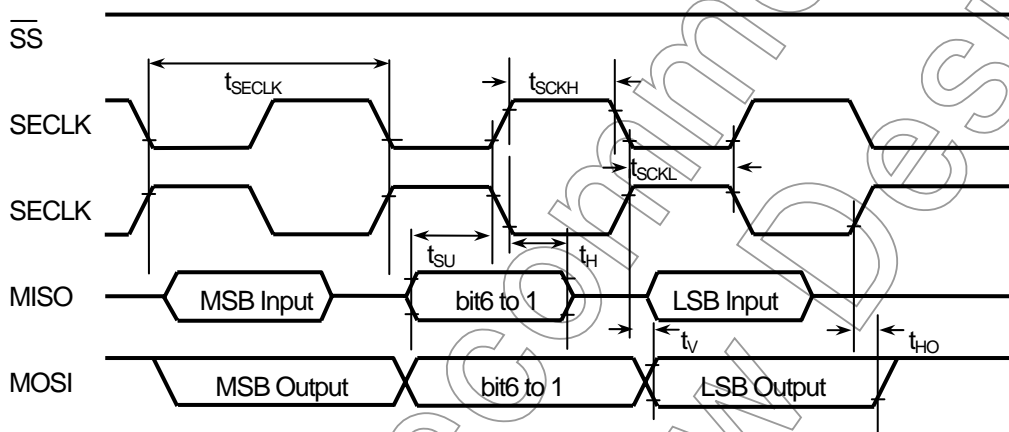
b) SEI master (CPHA = 1)
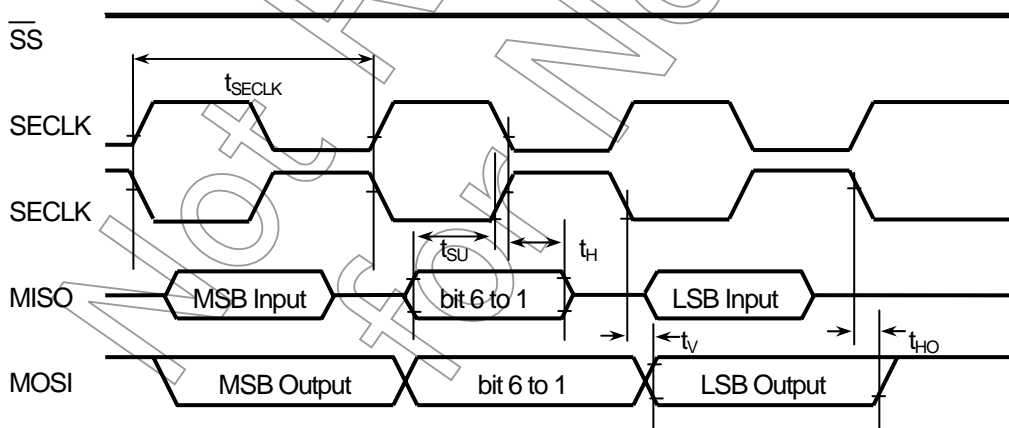
c) SEI slave (CPHA = 0)
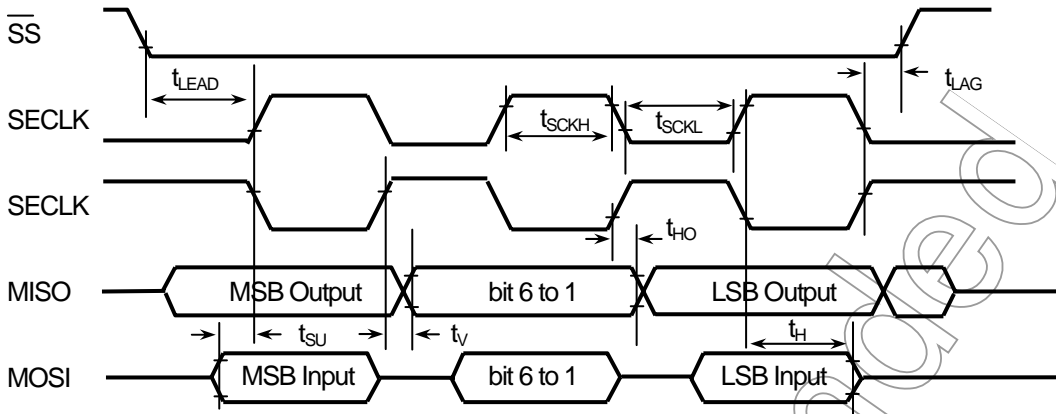
d) SEI slave (CPHA = 1)
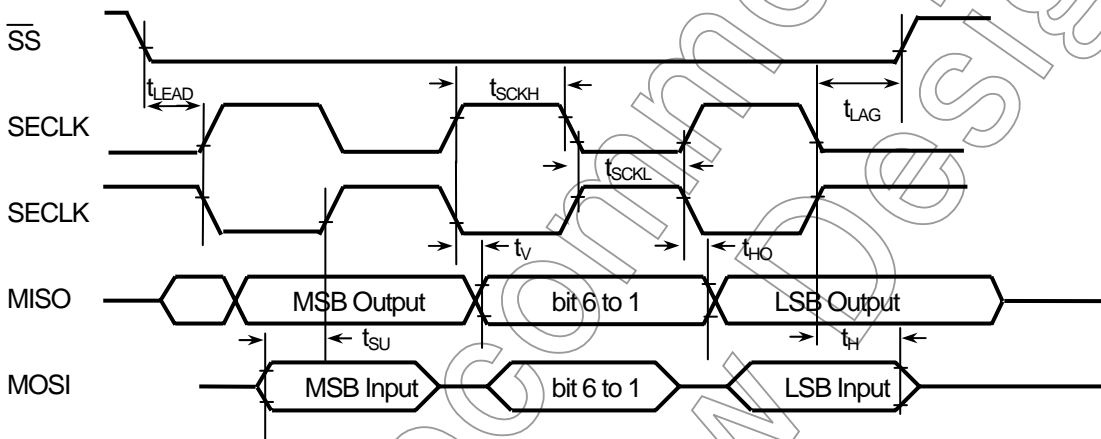
4.10 CAN Controller

V_{CC5} = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Symbol	Parameter	Variable		20MHz		Unit
		Min	Max	Min	Max	
t_{CCLK}	CAN Clock period	2T		100		ns
t_P	Tx edge → Rx Input		$2t_{CCLK}$-20		180	ns
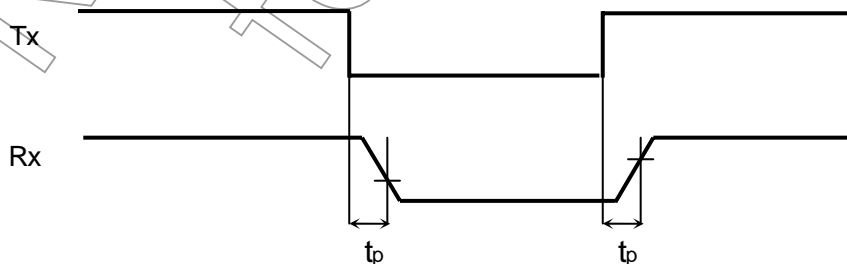
4.11 Recommended Oscillator Circuits

The following shows recommended oscillator circuits for the TMP92CD54I.

(1) Example resonator connections
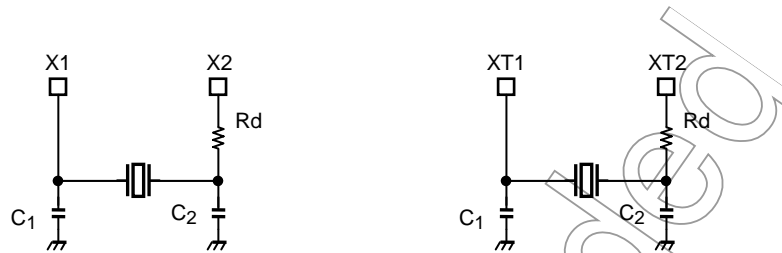
(a) Connection with High-frequency oscillator   (b) Connection with Low-frequency oscillator

Figure 4.11.1 Oscillation Circuits

Note: The load capacitance on the oscillator connection pins is the sum of C1 and C2 in the oscillator circuit (or incorporated in a resonator) and stray board capacitance. Since the total load capacitance varies with the board layout, the resonator might fail to work properly. To prevent this problem, the board traces near the oscillator circuit should be as short as possible. It is recommended to evaluate the oscillator using the actual application board.

(2) Recommended ceramic resonators

The TMP92FD54AI high-frequency oscillator circuit has been evaluated by Murata Manufacturing Co., Ltd. For details, please contact your Murata representative.
Figure 4.11.1 shows the recommended circuit constants for the ceramic resonator manufactured by Murata.

Table 4.11.1　Recommended ceramic resonator for the TMP92CD54AI (manufactured by Murata))

Oscillation Frequency [MHz]	Resonator Part Number		Parameter				Operating Condition	
			C1 [pF] (Note 1)	C2 [pF] (Note 1)	Rf [Ω]	Rd [Ω]	Voltage [V]	Temperature [°C]
8.0	SMD	CSTCE8M00G15C()-R0	(33)	(33)	Open	330	4.5 to 5.25	−40 to 85
10.0	SMD	CSTCE10M0G15C()-R0	(33)	(33)	Open	330		

Note 1: Enclosed in parentheses are the built-in load capacitor values.

Note 2: Part numbers and specifications of resonators manufactured by Murata are subject to change without notice. For details, please visit Murata's website at http://www.murata.co.jp.

4.12   Voltage Regulator

V_{CC5} = 4.5 V to 5.25 V / fc = 16 to 20 MHz / Ta = −40 to 85°C

Parameter		Symbol	Condition	Min.	Typ.	Max.	Unit
Input Voltage		V_{CC5}	Include ripple fluctuation voltage defined as Vp-p	4.5	5.0	5.25	V
	Peak-to-Peak voltage (ripple fluctuation voltage) Note)	Vp-p	Ripple frequency ≤ 100Hz (Sine-wave)	−	0	0.75	V
			Ripple frequency > 100Hz (Sine-wave)	−	0	0.3	V
			All Ripple frequency (except for Sine-wave)	−	0	0.2	V
Output Voltage		REGOUT	4.5 ≤ Vin ≤ 5.25, ILoad = 100mA (Vin = V_{CC5}) Ta = −40 to 85°C	3.0	3.3	3.6	V
Output Current		Iro	Vin − REGOUT = 1.0 V Ta = −40 to 85°C	0	−	150	mA
Quiescent Current		Iq	ILoad ≤ 10μA, Ta = −40 to 85°C	−	−	100	μA
		Iq1	10μA < ILoad < 100mA, Ta = 25°C	−	−	800	μA
		Iop	ILoad = 150mA, Ta = −40 to 85°C	−	−	10	mA
Standby Current		Is	REGEN = 0 (Regulator Only)	−	0.1	0.2	μA

0.5[Ω] ≤ ESR ≤ 5.0[Ω]

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Stabilization capacitor	Cs	Cb = 10 μF, ESR = 4.7 Ω	0.1	−	10	μF
Bypass capacitor	Cb	Cs = 10 μF, ESR = 4.7 Ω (Cs ≥ Cb)	0.1	−	10	μF
Input capacitor	Cin (Note)	Cs = 10 μF, ESR = 4.7 Ω	4.7	−	22	μF
Equivalent Series Resistor	ESR	Cs = 10 μF, Cb = 0.1 μF	0.5	−	5	Ω

0.5[Ω] ≤ ESR ≤ 50[Ω]

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Stabilization capacitor	Cs	Cb = 0.6 μF, ESR = 47 Ω	0.1	−	10	μF
Bypass capacitor	Cb	Cs = 10 μF, ESR = 47 Ω (Cs ≥ Cb)	0.6	−	10	μF
Input capacitor	Cin (Note)	Cs = 10 μF, ESR = 47 Ω	4.7	−	22	μF
Equivalent Series Resistor	ESR	Cs = 10 μF, Cb = 0.6μF	0.5	−	50	Ω

0.5[Ω] ≤ ESR ≤ 100[Ω]

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Stabilization capacitor	Cs	Cb = 1.0 μF, ESR = 100 Ω	0.1	−	10	μF
Bypass capacitor	Cb	Cs = 10 μF, ESR = 100 Ω (Cs ≥ Cb)	1.0	−	10	μF
Input capacitor	Cin (Note)	Cs = 10 μF, ESR = 100 Ω	4.7	−	22	μF
Equivalent Series Resistor	ESR	Cs = 10 μF, Cb = 1.0 μF	0.5	−	100	Ω

Note: Tantalum capacitors are recommended.
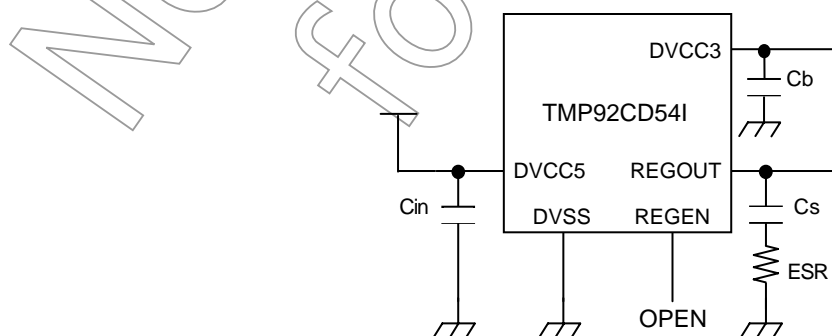
5.   Summary of Special Function Registers

Special function registers (SFRs) are control registers for input/output ports and peripheral units. They are allocated to 1024-byte address space from 000000H to 0003FFH.

(1) Input/output ports

(2) 8-bit timers

(3) 16-bit timers

(4) Serial channels

(5) Serial expansion interface

(6) Interrupt controller

(7) DMA controller

(8) Control registers

(9) AD converter

(10) Memory controller

(11) Serial bus interface controller

(12) CAN controller

(13) RTC controller

Table format

Symbol	Name	Address	7	6	5	4	3	2	1	0	
											→ Bit Symbol
											→ Read/Write
											→ Initial value after reset
											→ Remarks

Symbol definitions

R/W: CPU read and write access allowed
R: Only CPU read access allowed
W: Only CPU write access allowed
R/S: CPU read access and setting(Note) allowed
R/C: CPU read access and clearing(Note) allowed
RMW-prohibited: Read-modify-write not allowed
            (Prohibited instructions: RES/SET/TSET/CHG/STCF/ANDCF/ORCF/XORCF/etc.)
(Reserved):   Cannot be set

Note: R/S and R/C bits are set or cleared when the CPU writes a 1 to them.

Table 5.1   I/O Register Address Maps (1)

[1] Port

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	
16H	
17H	
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

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	(Reserved)
65H	(Reserved)
66H	(Reserved)
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 reserved registers.

Table 5.2   I/O Register Address Map (2)

[3] 8-bit timers:

ADDRESS	NAME	ADDRESS	NAME
0080H	TRUN01	0090H	TRUN45
81H	(Reserved)	91H	(Reserved)
82H	TREG0	92H	TREG4
83H	TREG1	93H	TREG5
84H	TMOD01	94H	TMOD45
85H	TFFCR1	95H	TFFCR5
86H	(Reserved)	96H	(Reserved)
87H	(Reserved)	97H	(Reserved)
88H	TRUN23	98H	TRUN67
89H	(Reserved)	99H	(Reserved)
8AH	TREG2	9AH	TREG6
8BH	TREG3	9BH	TREG7
8CH	TMOD23	9CH	TMOD67
8DH	TFFCR3	9DH	TFFCR7
8EH	(Reserved)	9EH	(Reserved)
8FH	(Reserved)	9FH	(Reserved)

[4] 16-bit timers:

ADDRESS	NAME	ADDRESS	NAME
00A0H	TRUN8	00B0H	TRUNA
A1H	(Reserved)	B1H	(Reserved)
A2H	TMOD8	B2H	TMODA
A3H	TFFCR8	B3H	TFFCRA
A4H	(Reserved)	B4H	(Reserved)
A5H	(Reserved)	B5H	(Reserved)
A6H	(Reserved)	B6H	(Reserved)
A7H	(Reserved)	B7H	(Reserved)
A8H	TREG8L	B8H	TREGAL
A9H	TREG8H	B9H	TREGAH
AAH	TREG9L	BAH	TREGBL
ABH	TREG9H	BBH	TREGBH
ACH	CAP8L	BCH	CAPAL
ADH	CAP8H	BDH	CAPAH
AEH	CAP9L	BEH	CAPBL
AFH	CAP9H	BFH	CAPBH

[5] SIO:

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:

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
00D0H	INTE12	00E0H	INTESED0	00F0H	INTE0AD
D1H	INTE34	E1H	INTERTC	F1H	INTETC01
D2H	INTE56	E2H	INTESB2	F2H	INTETC23
D3H	INTE7	E3H	INTESB0	F3H	INTETC45
D4H	INTET01	E4H	INTESB1	F4H	INTETC67
D5H	INTET23	E5H	INTMK0	F5H	(Reserved)
D6H	INTET45	E6H	INTMK1	F6H	IIMC
D7H	INTET67	E7H	INTMK2	F7H	INTNMWDT
D8H	INTET89	E8H	INTMK3	F8H	INTCLR
D9H	INTETAB	E9H	INTMK4	F9H	(Reserved)
DAH	INTETO8A	EAH	INTMK5	FAH	(Reserved)
DBH	INTES0	EBH	(Reserved)	FBH	(Reserved)
DCH	INTES1	ECH	WUPFLAG	FCH	(Reserved)
DDH	INTECRT	EDH	WUPMOD	FDH	(Reserved)
DEH	INTECG	EEH	WUPEDGE	FEH	(Reserved)
DFH	INTESEE0	EFH	WUPMASK	FFH	(Reserved)

Note: Do not access the without allocated names.

Table 5.3 I/O Register Address Map (3)

[6] INTC:

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:

ADDRESS	NAME
0110H	WDMOD
111H	WDCR
112H	(Reserved)
113H	(Reserved)
114H	(Reserved)
115H	(Reserved)
116H	(Reserved)
117H	(Reserved)
118H	RTCCR
119H	RTCFC
11AH	(Reserved)
11BH	(Reserved)
11CH	(Reserved)
11DH	(Reserved)
11EH	(Reserved)
11FH	(Reserved)

[8] 10-bit ADC:

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:

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)

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	FSWE [Note2]
16CH	(Reserved)
16DH	RAMCR
16EH	FLSR [Note2]
16FH	(Reserved)

[10] SBI:

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)

Note: This register is contained only in the TMP92FD54AI. It does not exist in the TMP92CD54I.

Table 5.4   I/O Register Address Map (4)

[10] SBI:

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
0180H	SBI2CR1	0190H	(Reserved)	01A0H	(Reserved)	01B0H	(Reserved)
181H	SBI2DBR	191H	(Reserved)	1A1H	(Reserved)	1B1H	(Reserved)
182H	I2C2AR	192H	(Reserved)	1A2H	(Reserved)	1B2H	(Reserved)
183H	SBI2CR2 /SBI2SR	193H	(Reserved)	1A3H	(Reserved)	1B3H	(Reserved)
184H	SBI2BR0	194H	(Reserved)	1A4H	(Reserved)	1B4H	(Reserved)
185H	SBI2BR1	195H	(Reserved)	1A5H	(Reserved)	1B5H	(Reserved)
186H	(Reserved)	196H	(Reserved)	1A6H	(Reserved)	1B6H	(Reserved)
187H	(Reserved)	197H	(Reserved)	1A7H	(Reserved)	1B7H	(Reserved)
188H	(Reserved)	198H	(Reserved)	1A8H	(Reserved)	1B8H	(Reserved)
189H	(Reserved)	199H	(Reserved)	1A9H	(Reserved)	1B9H	(Reserved)
18AH	(Reserved)	19AH	(Reserved)	1AAH	(Reserved)	1BAH	(Reserved)
18BH	(Reserved)	19BH	(Reserved)	1ABH	(Reserved)	1BBH	(Reserved)
18CH	(Reserved)	19CH	(Reserved)	1ACH	(Reserved)	1BCH	(Reserved)
18DH	(Reserved)	19DH	(Reserved)	1ADH	(Reserved)	1BDH	(Reserved)
18EH	(Reserved)	19EH	(Reserved)	1AEH	(Reserved)	1BEH	(Reserved)
18FH	(Reserved)	19FH	(Reserved)	1AFH	(Reserved)	1BFH	(Reserved)

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
01C0H	(Reserved)	01D0H	(Reserved)	01E0H	(Reserved)	01F0H	(Reserved)
1C1H	(Reserved)	1D1H	(Reserved)	1E1H	(Reserved)	1F1H	(Reserved)
1C2H	(Reserved)	1D2H	(Reserved)	1E2H	(Reserved)	1F2H	(Reserved)
1C3H	(Reserved)	1D3H	(Reserved)	1E3H	(Reserved)	1F3H	(Reserved)
1C4H	(Reserved)	1D4H	(Reserved)	1E4H	(Reserved)	1F4H	(Reserved)
1C5H	(Reserved)	1D5H	(Reserved)	1E5H	(Reserved)	1F5H	(Reserved)
1C6H	(Reserved)	1D6H	(Reserved)	1E6H	(Reserved)	1F6H	(Reserved)
1C7H	(Reserved)	1D7H	(Reserved)	1E7H	(Reserved)	1F7H	(Reserved)
1C8H	(Reserved)	1D8H	(Reserved)	1E8H	(Reserved)	1F8H	(Reserved)
1C9H	(Reserved)	1D9H	(Reserved)	1E9H	(Reserved)	1F9H	(Reserved)
1CAH	(Reserved)	1DAH	(Reserved)	1EAH	(Reserved)	1FAH	(Reserved)
1CBH	(Reserved)	1DBH	(Reserved)	1EBH	(Reserved)	1FBH	(Reserved)
1CCH	(Reserved)	1DCH	(Reserved)	1ECH	(Reserved)	1FCH	(Reserved)
1CDH	(Reserved)	1DDH	(Reserved)	1EDH	(Reserved)	1FDH	(Reserved)
1CEH	(Reserved)	1DEH	(Reserved)	1EEH	(Reserved)	1FEH	(Reserved)
1CFH	(Reserved)	1DFH	(Reserved)	1EFH	(Reserved)	1FFH	(Reserved)

Note: Do not access the without allocated names.

Table 5.5 I/O Register Address Map (5)

[11] CAN:

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
0200H	MB0MI0L	0210H	MB1MI0L	0220H	MB2MI0L	0230H	MB3MI0L
201H	MB0MI0H	211H	MB1MI0H	221H	MB2MI0H	231H	MB3MI0H
202H	MB0MI1L	212H	MB1MI1L	222H	MB2MI1L	232H	MB3MI1L
203H	MB0MI1H	213H	MB1MI1H	223H	MB2MI1H	233H	MB3MI1H
204H	MB0MCFL	214H	MB1MCFL	224H	MB2MCFL	234H	MB3MCFL
205H	MB0MCFH	215H	MB1MCFH	225H	MB2MCFH	235H	MB3MCFH
206H	MB0D0	216H	MB1D0	226H	MB2D0	236H	MB3D0
207H	MB0D1	217H	MB1D1	227H	MB2D1	237H	MB3D1
208H	MB0D2	218H	MB1D2	228H	MB2D2	238H	MB3D2
209H	MB0D3	219H	MB1D3	229H	MB2D3	239H	MB3D3
20AH	MB0D4	21AH	MB1D4	22AH	MB2D4	23AH	MB3D4
20BH	MB0D5	21BH	MB1D5	22BH	MB2D5	23BH	MB3D5
20CH	MB0D6	21CH	MB1D6	22CH	MB2D6	23CH	MB3D6
20DH	MB0D7	21DH	MB1D7	22DH	MB2D7	23DH	MB3D7
20EH	MB0TSVL	21EH	MB1TSVL	22EH	MB2TSVL	23EH	MB3TSVL
20FH	MB0TSVH	21FH	MB1TSVH	22FH	MB2TSVH	23FH	MB3TSVH

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
0240H	MB4MI0L	0250H	MB5MI0L	0260H	MB6MI0L	0270H	MB7MI0L
241H	MB4MI0H	251H	MB5MI0H	261H	MB6MI0H	271H	MB7MI0H
242H	MB4MI1L	252H	MB5MI1L	262H	MB6MI1L	272H	MB7MI1L
243H	MB4MI1H	253H	MB5MI1H	263H	MB6MI1H	273H	MB7MI1H
244H	MB4MCFL	254H	MB5MCFL	264H	MB6MCFL	274H	MB7MCFL
245H	MB4MCFH	255H	MB5MCFH	265H	MB6MCFH	275H	MB7MCFH
246H	MB4D0	256H	MB5D0	266H	MB6D0	276H	MB7D0
247H	MB4D1	257H	MB5D1	267H	MB6D1	277H	MB7D1
248H	MB4D2	258H	MB5D2	268H	MB6D2	278H	MB7D2
249H	MB4D3	259H	MB5D3	269H	MB6D3	279H	MB7D3
24AH	MB4D4	25AH	MB5D4	26AH	MB6D4	27AH	MB7D4
24BH	MB4D5	25BH	MB5D5	26BH	MB6D5	27BH	MB7D5
24CH	MB4D6	25CH	MB5D6	26CH	MB6D6	27CH	MB7D6
24DH	MB4D7	25DH	MB5D7	26DH	MB6D7	27DH	MB7D7
24EH	MB4TSVL	25EH	MB5TSVL	26EH	MB6TSVL	27EH	MB7TSVL
24FH	MB4TSVH	25FH	MB5TSVH	26FH	MB6TSVH	27FH	MB7TSVH

Note: Do not access the without allocated names.

Table 5.6   I/O Register Address Map (6)

[11] CAN:

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
0280H	MB8MI0L	0290H	MB9MI0L	02A0H	MB10MI0L	02B0H	MB11MI0L
281H	MB8MI0H	291H	MB9MI0H	2A1H	MB10MI0H	2B1H	MB11MI0H
282H	MB8MI1L	292H	MB9MI1L	2A2H	MB10MI1L	2B2H	MB11MI1L
283H	MB8MI1H	293H	MB9MI1H	2A3H	MB10MI1H	2B3H	MB11MI1H
284H	MB8MCFL	294H	MB9MCFL	2A4H	MB10MCFL	2B4H	MB11MCFL
285H	MB8MCFH	295H	MB9MCFH	2A5H	MB10MCFH	2B5H	MB11MCFH
286H	MB8D0	296H	MB9D0	2A6H	MB10D0	2B6H	MB11D0
287H	MB8D1	297H	MB9D1	2A7H	MB10D1	2B7H	MB11D1
288H	MB8D2	298H	MB9D2	2A8H	MB10D2	2B8H	MB11D2
289H	MB8D3	299H	MB9D3	2A9H	MB10D3	2B9H	MB11D3
28AH	MB8D4	29AH	MB9D4	2AAH	MB10D4	2BAH	MB11D4
28BH	MB8D5	29BH	MB9D5	2ABH	MB10D5	2BBH	MB11D5
28CH	MB8D6	29CH	MB9D6	2ACH	MB10D6	2BCH	MB11D6
28DH	MB8D7	29DH	MB9D7	2ADH	MB10D7	2BDH	MB11D7
28EH	MB8TSVL	29EH	MB9TSVL	2AEH	MB10TSVL	2BEH	MB11TSVL
28FH	MB8TSVH	29FH	MB9TSVH	2AFH	MB10TSVH	2BFH	MB11TSVH

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
02C0H	MB12MI0L	02D0H	MB13MI0L	02E0H	MB14MI0L	02F0H	MB15MI0L
2C1H	MB12MI0H	2D1H	MB13MI0H	2E1H	MB14MI0H	2F1H	MB15MI0H
2C2H	MB12MI1L	2D2H	MB13MI1L	2E2H	MB14MI1L	2F2H	MB15MI1L
2C3H	MB12MI1H	2D3H	MB13MI1H	2E3H	MB14MI1H	2F3H	MB15MI1H
2C4H	MB12MCFL	2D4H	MB13MCFL	2E4H	MB14MCFL	2F4H	MB15MCFL
2C5H	MB12MCFH	2D5H	MB13MCFH	2E5H	MB14MCFH	2F5H	MB15MCFH
2C6H	MB12D0	2D6H	MB13D0	2E6H	MB14D0	2F6H	MB15D0
2C7H	MB12D1	2D7H	MB13D1	2E7H	MB14D1	2F7H	MB15D1
2C8H	MB12D2	2D8H	MB13D2	2E8H	MB14D2	2F8H	MB15D2
2C9H	MB12D3	2D9H	MB13D3	2E9H	MB14D3	2F9H	MB15D3
2CAH	MB12D4	2DAH	MB13D4	2EAH	MB14D4	2FAH	MB15D4
2CBH	MB12D5	2DBH	MB13D5	2EBH	MB14D5	2FBH	MB15D5
2CCH	MB12D6	2DCH	MB13D6	2ECH	MB14D6	2FCH	MB15D6
2CDH	MB12D7	2DDH	MB13D7	2EDH	MB14D7	2FDH	MB15D7
2CEH	MB12TSVL	2DEH	MB13TSVL	2EEH	MB14TSVL	2FEH	MB15TSVL
2CFH	MB12TSVH	2DFH	MB13TSVH	2EFH	MB14TSVH	2FFH	MB15TSVH

Note: Do not access the without allocated names.

Table 5.7   I/O Register Address Map (7)

[11] CAN:

ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME
0300H	MCL	0310H	LAM0L	0320H	GIFL	0330H	TSPL
301H	MCH	311H	LAM0H	321H	GIFH	331H	TSPH
302H	MDL	312H	LAM1L	322H	GIML	332H	TSCL
303H	MDH	313H	LAM1H	323H	GIMH	333H	TSCH
304H	TRSL	314H	GAM0L	324H	MBTIFL	334H	(Reserved)
305H	TRSH	315H	GAM0H	325H	MBTIFH	335H	(Reserved)
306H	TRRL	316H	GAM1L	326H	MBRIFL	336H	(Reserved)
307H	TRRH	317H	GAM1H	327H	MBRIFH	337H	(Reserved)
308H	TAL	318H	MCRL	328H	MBIML	338H	(Reserved)
309H	TAH	319H	MCRH	329H	MBIMH	339H	(Reserved)
30AH	AAL	31AH	GSRL	32AH	CDRL	33AH	(Reserved)
30BH	AAH	31BH	GSRH	32BH	CDRH	33BH	(Reserved)
30CH	RMPL	31CH	BCR1L	32CH	RFPL	33CH	(Reserved)
30DH	RMPH	31DH	BCR1H	32DH	RFPH	33DH	(Reserved)
30EH	RMLL	31EH	BCR2L	32EH	CECL	33EH	(Reserved)
30FH	RMLH	31FH	BCR2H	32FH	CECH	33FH	(Reserved)

ADDRESS	NAME
0340H to 3FFH	(Reserved)

Note: Do not access the without allocated names.

(1) Input/output ports

Port0

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	Port 0 Register	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P0CR	Port 0 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	Port 0 Function Register	03H (no RMW)	-	-	-	-	-	-	-	P0F
										W
			-	-	-	-	-	-	-	0
			0:PORT 1:Data Bus(D7 to D0)							

Port4

Symbol	Name	Address	7	6	5	4	3	2	1	0
P4	Port 4 Register	10H	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P4CR	Port 4 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	Port 4 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 to A0							

Port7

Symbol	Name	Address	7	6	5	4	3	2	1	0
P7	Port 7 Register	1CH	-	-	P75	P74	P73	P72	P71	P70
							R/W			
			-	-	0	1	1	1	1	1
							Input/Output			
P7CR	Port 7 Control Register	1EH (no RMW)	-	-	P75C	P74C	P73C	P72C	P71C	P70C
							W			
			-	-	0	1	1	0	1	1
							0:Input 1:Output			
P7FC	Port 7 Function Register	1FH (no RMW)	-	-	P75F	P74F	P73F	P72F	P71F	P70F
							W			
			-	-	0	0	0	0	0	0
					0:Port 1: $\overline{\text{WAIT}}$	0:Port	0:Port 1: $\overline{\text{CS}}$	0:Port 1:SI2 SCL2 Note	0:Port 1: $\overline{\text{WR}}$	0:Port 1: $\overline{\text{RD}}$

Note: To switch the P72 output from C-MOS to open-drain output, set PNODE<ODE72> to 1.

PortC

Symbol	Name	Address	7	6	5	4	3	2	1	0
PC	Port C Register	30H	-	-	PC5	PC4	PC3	PC2	PC1	PC0
							R/W			
			-	-	0	0	0	0	0	0
							Input/Output			
PCCR	Port C Control Register	32H (no RMW)	-	-	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
							W			
			-	-	0	0	0	0	0	0
							0:Input 1:Output			
PCFC	Port C 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 TI4	0:Port INT2 1:TO3	0:Port 1:TO1	0:Port INT1 TI0

PortD

Symbol	Name	Address	7	6	5	4	3	2	1	0
PD	Port D	34H	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
PDCR	Port D 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	Port D Function Register	37H (no RMW)	PD7F	PD6F	PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
			W							
			0	0	0	0	0	0	0	0
			0:Port WUINT7 1:TOB A23	0:Port WUINT6 1:TOA A22	0:Port TIB WUINT5 1:A21	0:Port INT7 TIA WUINT4 1:A20	0:Port WUINT3 1:TO9 A19	0:Port WUINT2 1:TO8 A18	0:Port INT6 TI9 WUINT1 1:A17	0:Port INT5 TI8 WUINT0 1:A16

PDCR	PDFC	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
0	0	Input Port, WUINT7	Input Port, WUINT6	Input Port, TIB, WUINT5	Input Port, INT7, TIA, WUINT4	Input Port, WUINT3	Input Port, WUINT2	Input Port, INT6, TI9, WUINT1	Input Port, INT5, TI8, WUINT0
1	0	Output Port							
1	1	TOB	TOA	TIB, WUINT5	TIA, INT7, WUINT4	TO9	TO8	TI9, INT6, WUINT1	TI8, INT5, WUINT0
0	1	A23	A22	A21	A20	A19	A18	A17	A16

PortF

Symbol	Name	Address	7	6	5	4	3	2	1	0
PF	Port F	3CH	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
PFCR	Port F 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	Port F Function Register	3FH (no RMW)	PF7F	PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
			W							
			0	0	0	0	0	0	0	0
			0:Port 1:RX	0:Port 1:TX	0:Port $\overline{CTS1}$ 1:SCLK1	0:Port 1:RXD1	0:Port 1:TXD1	0:Port $\overline{CTS0}$ 1:SCLK0	0:Port 1:RXD0	0:Port 1:TXD0

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

PortG

Symbol	Name	Address	7	6	5	4	3	2	1	0
PG	Port G 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	Port L Register	54H	-	-	-	-	PL3	PL2	PL1	PL0
							R			
			-	-	-	-	Input			

PortM

Symbol	Name	Address	7	6	5	4	3	2	1	0
PM	Port M	58H	-	-	-	PM4	PM3	PM2	PM1	PM0
						R/W				
			-	-	-	0	0	0	0	0
						Input/Output				
PMODE	Port M Open Drain Enable Register	59H	-	-	-	-	ODEM3	ODEM2	ODEM1	-
							R/W			
			-	-	-	-	0	0	0	-
							PM3 Output 0:CMOS 1:Open Drain	PM2 Output 0:CMOS 1:Open Drain	PM1 Output 0:CMOS 1:Open Drain	
PMCR	Port M Control Register	5AH (no RMW)	-	-	-	PM4C	PM3C	PM2C	PM1C	PM0C
						W				
			-	-	-	0	0	0	0	0
						0:Input 1:Output				
PMFC	Port M Function Register	5BH (no RMW)	-	-	-	PM4F	PM3F	PM2F	PM1F	PM0F
						W				
			-	-	-	0	0	0	0	0
						0:Port 1:SCK2	0:Port 1:SECLK A11	0:Port 1:MISO A10	0:Port 1:MOSI A9	0:Port 1:$\overline{SS}$ A8

PMCR	PMFC	-	-	-	PM4	PM3	PM2	PM1	PM0
0	0	-	-	-	Input Port, SCK2 (Input)	Input Port	Input Port	Input Port	Input Port, $\overline{SS}$
1	0	-				Output Port			
1	1	-	-	-	SCK2 (Output)	SECLK	MISO	MOSI	$\overline{SS}$
0	1	-	-	-	Don't use this setting	A11	A10	A9	A8

PortN

Symbol	Name	Address	7	6	5	4	3	2	1	0
PN	PORTN	5CH	-	PN6	PN5	PN4	PN3	PN2	PN1	PN0
				R/W						
			-	0	0	0	0	0	0	0
				Input/Output						
PNODE	Port N Open Drain Enable Register	5DH	ODE72	ODEN6	ODEN5	ODEN4	-	ODEN2	ODEN1	-
			R/W					R/W		
			0	0	0	0	-	0	0	-
			P72 Output 0:CMOS 1:Open Drain	PN6 Output 0:CMOS 1:Open Drain	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	Port N Control Register (no RMW)	5EH	-	PN6C	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
				W						
			-	0	0	0	0	0	0	0
				0:Input 1:Output						
PNFC	Port N Function Register (no RMW)	5FH	-	PN6F	PN5F	PN4F	PN3F	PN2F	PN1F	PN0F
				W			W			
			-	0	0	0	0	0	0	0
				0:Port 1:SO2 SDA2 A15	0:Port SI1 1:SCL1 A14	0:Port 1:SO1 SDA1 A13	0:Port 1:SCK1 A12	0:Port SI0 1:SCL0	0:Port 1:SO0 SDA0	0:Port 1:SCK0

PNCR	PNFC	-	PN6	PN5	PN4	PN3	PN2	PN1	PN0
0	0	-	Input Port	Input Port, SI1	Input Port	Input Port, SCK1 (Input)	Input Port, SI0	Input Port	Input Port, SCK0 (Input)
1	0	-	Output Port						
1	1	-	SO2/SDA2	SCL1	SO1/SDA1	SCK1 (Output)	SCL0	SO0/SDA0	SCK0 (Output)
0	1	-	A15	A14	A13	A12	Don't use this setting.		

Note: To switch the P72 output from C-MOS to open-drain output, set PNODE<ODE72> to 1.

(2) 8-bit timers

8-bit timers 01, 23, 45, and 67

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN01	8-bit Timer01 Run Register	80H	T0RDE	-	-	-	I2T01	T01PRUN	T1RUN	T0RUN
			R/W				R/W		R/W	
			0	-	-	-	0	0	0	0
			Double Buffer 0:Disable 1:Enable				IDLE2 0:Stop 1:Operate	8-bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG0	8-bit Timer Register 0	82H (no RMW)	W Undefined							
TREG1	8-bit Timer Register 1	83H (no RMW)	- W Undefined							
TMOD01	8-bit Timer0,1 Source CLK & MODE Register	84H	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8-bit Timer 01:16-bit Timer 10:8-bit PPG 11:8-bit PWM		PWM cycle 00:reserved 01:2^6 10:2^7 11:2^8		Timer1 source clock 00:T0TRG 01:ϕT1 10:ϕT16 11:ϕT256		Timer0 source clock 00:TI0 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	8-bit 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	8-bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG2	8-bit Timer Register 2	8AH (no RMW)	- W Undefined							
TREG3	8-bit Timer Register 3	8BH (no RMW)	- W Undefined							
TMOD23	8-bit Timer2,3 Source CLK & MODE Register	8CH	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8-bit Timer 01:16-bit Timer 10:8-bit PPG 11:8-bit PWM		PWM cycle 00:reserved 01:2^6 10:2^7 11:2^8		Timer3 source clock 00:T2TRG 01:ϕT1 10:ϕT16 11:ϕT256		Timer2 source clock 00:reserved 01:ϕT1 10:ϕT4 11:ϕT16	

Symbol	Name	Address	7	6	5	4	3	2	1	0
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
TRUN45	8-bit 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	8-bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG4	8-bit Timer Register 4	92H (no RMW)	-							
			W							
			Undefined							
TREG5	8-bit Timer Register 5	93H (no RMW)	-							
			W							
			Undefined							
TMOD45	8-bit Timer4,5 Source CLK & MODE Register	94H	T45M1	T45M0	PWM41	PWM40	T5CLK1	T5CLK0	T4CLK1	T4CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8-bit Timer 01:16-bit Timer 10:8-bit PPG 11:8-bit PWM		PWM cycle 00:reserved 01:2^6 10:2^7 11:2^8		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	8-bit 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	8-bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG6	8-bit Timer Register 6	9AH (no RMW)	-							
			W							
			Undefined							
TREG7	8-bit Timer Register 7	9BH (no RMW)	-							
			W							
			Undefined							

Symbol	Name	Address	7	6	5	4	3	2	1	0
TMOD67	8-bit Timer6,7 Source CLK & MODE Register	9CH	T67M1	T67M0	PWM61	PWM60	T7CLK1	T7CLK0	T6CLK1	T6CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8-bit Timer 01:16-bit Timer 10:8-bit PPG 11:8-bit PWM		PWM cycle 00:reserved 01:2^6 10:2^7 11:2^8		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 timers

16-bit timers 8 and A

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN8	16-bit Timer8 Run Register	A0H	T8RDE	-	-	-	I2T8	T8PRUN	-	T8RUN
			R/W	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	16-bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TMOD8	16-bit 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:TI8↑ TI9↑ 10:TI8↑ TI8↓ 11:TFF1↑ TFF1↓		1:UC8 Clear Enable	Source Clock 00:TI8 01: φT1 10: φT4 11: φT16	
TFFCR8	16-bit 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	16-bit Timer Register 8 Low	A8H (no RMW)	-							
			W							
			Undefined							
TREG8H	16-bit Timer Register 8 High	A9H (no RMW)	-							
			W							
			Undefined							
TREG9L	16-bit Timer Register 9 Low	AAH (no RMW)	-							
			W							
			Undefined							
TREG9H	16-bit 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	16-bit TimerA Run Register	B0H	TARDE	-	-	-	I2TA	TAPRUN	-	TARUN
			R/W	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	16-bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TMODA	16-bit 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	16-bit 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	16-bit Timer Register A Low	B8H (no RMW)	-							
			W							
			Undefined							
TREGAH	16-bit Timer Register A High	B9H (no RMW)	-							
			W							
			Undefined							
TREGBL	16-bit Timer Register B Low	BAH (no RMW)	-							
			W							
			Undefined							
TREGBH	16-bit 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 (no RMW)	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	
			Undefined	0	0	0	0	0	0	0
			Receive data bit 8	Parity 0:Odd 1:Even	Parity 0:Disable 1:Enable	1:Error			0:SCLK0 ↑ 1:SCLK0 ↓	0:Baud Rate Generator 1:SCLK0 Pin Input
						Overrun	Parity	Framing		
SC0MOD0	Serial Channel 0 Mode 0 Register	C2H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
				R/W						
			Undefined	0	0	0	0	0	0	0
			Transmission 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 (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 (no RMW)	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	
			Undefined	0	0	0	0	0	0	0
			Receive Data bit 8	Parity 0:Odd 1:Even	Parity 0:Disable 1:Enable	1:Error			0:SCLK1 ↑ 1:SCLK1 ↓	0:Baud Rate Generator 1:SCLK1 Pin Input
						Overrun	Parity	Framing		
SC1MOD0	Serial Channel 1 Mode 0 Register	CAH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmission 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
SECR	SEI Control Register	60H	MODE	SEE	BOS	MSTR	CPOL	CPHA	SER1	SER0
			W	R/W						
			0	0	0	0	0	1	1	1
			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 selection See figure 3.11.2, 3.11.3	Clock Phase Selection See figure 3.11.2, 3.11.3	SEI Transfer Rate Select 00:reserved 01:Divided by 2 10:Divided by 4 11:Divided by 16	
SESR	SEI Status Register	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/C					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
SEDR	SEI Data Register	62H	SED7	SED6	SED5	SED4	SED3	SED2	SED1	SED0
			R(Reception)/W(Transmission)							
			0	0	0	0	0	0	0	0
			Receive Data / Transfer Data							

(6) Interrupt controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 & INTAD Enable Register	F0h	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	IOC	IOM2	IOM1	IOM0
			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	I2M0	I1C	I1M2	I1M1	I1M0
			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	I4M0	I3C	I3M2	I3M1	I3M0
			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	I6M0	I5C	I5M2	I5M1	I5M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE7	INT7 Enable Register	D3h	-	-	-	-	INT7(CAPA)			
							I7C	I7M2	I7M1	I7M0
							R	R/W		
			-	-	-	-	0	0	0	0
INTET01	INTT0 & INTT1 Enable Register	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 Register	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 Register	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 Register	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 Register	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 Register	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 Register	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 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
INTECRT	INTCR & INTCT Enable Register	DDh	INTCT				INTCR			
			ICTC	ICTM2	ICTM1	ICTM0	ICRC	ICRM2	ICRM1	ICRM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTECG	INTCG Enable Register	Deh	-	-	-	-	INTCG			
							ICGC	ICGM2	ICGM1	ICGM0
							R	R/W		
			-	-	-	-	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
INTERTC	INTRTC Enable	E1h	-	-	-	-	INTRTC			
							IRTCC	IRTCM2	IRTCM1	IRTCM0
							R	R/W		
			-	-	-	-	0	0	0	0
INTESB2	INTSBE2 & INTSBS2 Enable Register	E2h	INTSBS2				INTSBE2			
			ISBS0C	ISBS0M2	ISBS0M1	ISBS0M0	ISBE0C	ISBE0M2	ISBE0M1	ISBE0M0
			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
INTMK0	Interrupt Mask Control 0	E5h	MKI7	MKI6	MKI5	MKI4	MKI3	MKI2	MKI1	MKI0
			R/W							
			1	1	1	1	1	1	1	1
			0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable
INTMK1	Interrupt Mask Control 1	E6h	MKIT7	MKIT6	MKIT5	MKIT4	MKIT3	MKIT2	MKIT1	MKIT0
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			1	1	1	1	1	1	1	1
			0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTMK2	Interrupt Mask Control 2	E7h	–	MKIRTC	MKITDA	MKITD	MKITRB	MKITRA	MKITR9	MKITR8
				R/W	R/W	R/W	R/W	R/W	R/W	R/W
			-	1	1	1	1	1	1	1
				0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable
INTMK3	Interrupt Mask Control 3	E8h	–	MKICG	MKICT	MKICR	MKITX1	MKIRX1	MKITX0	MKIRX0
				R/W	R/W	R/W	R/W	R/W	R/W	R/W
			-	1	1	1	1	1	1	1
				0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable
INTMK4	Interrupt Mask Control 4	E9h	–	–	–	–	MKISET0	MKISER0	MKISEE0	MKISEM0
							R/W	R/W	R/W	R/W
			-	-	-	-	1	1	1	1
							0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable
INTMK5	Interrupt Mask Control 5	EAh	-	MKISBS2	MKISBE2	MKIAD	MKISBE1	MKISBE1	MKISBS0	MKISBE0
				R/W	R/W	R/W	R/W	R/W	R/W	R/W
			-	1	1	1	1	1	1	1
				0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable	0: Mask 1: Enable
WUPFLAG	Wake-up flag Control Register	ECh	WFLG7	WFLG6	WFLG5	WFLG4	WFLG3	WFLG2	WFLG1	WFLG0
			R							
			0	0	0	0	0	0	0	0
			WUINT7 0:No-request 1:request	WUINT6 0:No-request 1:request	WUINT5 0:No-request 1:request	WUINT4 0:No-request 1:request	WUINT3 0:No-request 1:request	WUINT2 0:No-request 1:request	WUINT1 0:No-request 1:request	WUINT0 0:No-request 1:request
WUPMOD	Wake-up Mode Control Register	EDh	WMD7	WMD6	WMD5	WMD4	WMD3	WMD2	WMD1	WMD0
			R/W							
			0	0	0	0	0	0	0	0
			WUINT7 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT6 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT5 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT4 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT3 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT2 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT1 0:Falling & Rising Edge 1:Falling or Rising Edge	WUINT0 0:Falling & Rising Edge 1:Falling or Rising Edge
WUPEDGE	Wake-up Edge select Register	EEh	WED7	WED6	WED5	WED4	WED3	WED2	WED1	WED0
			R/W							
			0	0	0	0	0	0	0	0
			WUINT7 0:Falling Edge 1:Rising Edge	WUINT6 0:Falling Edge 1:Rising Edge	WUINT5 0:Falling Edge 1:Rising Edge	WUINT4 0:Falling Edge 1:Rising Edge	WUINT3 0:Falling Edge 1:Rising Edge	WUINT2 0:Falling Edge 1:Rising Edge	WUINT1 0:Falling Edge 1:Rising Edge	WUINT0 0:Falling Edge 1:Rising Edge
WUPMASK	Wake-up Mask Register	EFh	WMK7	WMK6	WMK5	WMK4	WMK3	WMK2	WMK1	WMK0
			R/W							
			0	0	0	0	0	0	0	0
			WUINT7 0:Disable 1:Enable	WUINT6 0:Disable 1:Enable	WUINT5 0:Disable 1:Enable	WUINT4 0:Disable 1:Enable	WUINT3 0:Disable 1:Enable	WUINT2 0:Disable 1:Enable	WUINT1 0:Disable 1:Enable	WUINT0 0:Disable 1:Enable

Symbol	Name	Address	7	6	5	4	3	2	1	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			
			INMIC	-	-	-	IWDC	-	-	-
			R				R			
			0	-	-	-	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 (no RMW)					DMA0 Start Vector			
			-	-	DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
							R/W			
			-	-	0	0	0	0	0	0
DMA1V	DMA1 Start Vector Register	101h (no RMW)					DMA1 Start Vector			
			-	-	DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
							R/W			
			-	-	0	0	0	0	0	0
DMA2V	DMA2 Start Vector Register	102h (no RMW)					DMA2 Start Vector			
			-	-	DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
							R/W			
			-	-	0	0	0	0	0	0
DMA3V	DMA3 Start Vector Register	103h (no RMW)					DMA3 Start Vector			
			-	-	DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
							R/W			
			-	-	0	0	0	0	0	0
DMA4V	DMA4 Start Vector Register	104h (no RMW)					DMA4 Start Vector			
			-	-	DMA4V5	DMA4V4	DMA4V3	DMA4V2	DMA4V1	DMA4V0
							R/W			
			-	-	0	0	0	0	0	0
DMA5V	DMA5 Start Vector Register	105h (no RMW)					DMA5 Start Vector			
			-	-	DMA5V5	DMA5V4	DMA5V3	DMA5V2	DMA5V1	DMA5V0
							R/W			
			-	-	0	0	0	0	0	0
DMA6V	DMA6 Start Vector Register	106h (no RMW)					DMA6 Start Vector			
			-	-	DMA6V5	DMA6V4	DMA6V3	DMA6V2	DMA6V1	DMA6V0
							R/W			
			-	-	0	0	0	0	0	0
DMA7V	DMA7 Start Vector Register	107h (no RMW)					DMA7 Start Vector			
			-	-	DMA7V5	DMA7V4	DMA7V3	DMA7V2	DMA7V1	DMA7V0
							R/W			
			-	-	0	0	0	0	0	0
DMAB	DMA Burst Register	108h (no RMW)					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 registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
CLKMOD	Clock Mode Register	10AH	HALTM1	HALTM0	-	-	-	CLKOE	CLKM1	CLKM0
			R/W			R/W		R/W		
			1	1	-	0	-	0	0	0
			HALT mode 00:IDLE3 mode 01:STOP mode 10:IDLE1 mode 11:IDLE2 mode			Fixed to "0"		CLK Output Enable 0 : High-z (Pull up) 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
			String 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 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					
(Note2) FSWE	Flash Security Write Enable Register	16BH	-	-	-	-	-	-	-	-
						R/W				
			0	0	0	0	0	0	0	0
			C9H: Auto Chip Erase & Unprotect command Enable Code Others: Auto Chip Erase & Unprotect command Disable Code							
RAMCR	RAM Write Control Register	16DH	RAMSTB	RAMWI	-	-	-	-	-	-
				R/W						
			0 (Note1)	1	-	-	-	-	-	-
			0:lost data or Power on reset 1:kept data	RAM write 0:Disable 1:Enable						
(Note2) FLSR	Flash Status Register	16EH	-	-	-	-	-	R/BSY	-	-
			R/W	R/W	R/W	R/W		R		
			0	0	0	0	-	1	-	-
			Set to 0.	Set to 0.	Set to 0.	Set to 0.		Ready /Busy flag 0:Busy (auto ope -ration in progress) 1:Ready (auto operation finished)		

Note1: This register is contained only in the TMP92FD54AI. It does not exist in the TMP92CD54I.

Note2: This bit is initialized to 0 upon a power-on reset but not affected by a warm reset (a reset applied when the power is on).

(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	SCK3	SCK2	SCK1	SWRMON/ SCK0
			W			R/W	W			R/W
			0	0	0	0	1	0	0	1/0
			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" / fast / standard 0001:– 0010:– 0011: 8 0100: 9 0101: 10 0110:11 1000:fast 1111:standard other: Reserved			
		170H (no RMW) SIO mode	SIOS	SIOINH	SIOM1	SIOM0	-	SCK2	SCK1	SCK0
			W				W		W	
			0	0	0	0	1	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		Note) Write 0 to this bit in SIO mode.	Setting of the divide value "n" 000:4 001:5 010:6 011:7 100:8 101:9 110:10 111: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:Port 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		W	W
			-	-	-	-	0	0	0	0
							Operation mode selection 00:Port mode 10:I2C mode 01:SIO mode 11:reserved		Fix to "00"	
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	SCK3	SCK2	SCK1	SWRMON/ SCK0
			W			R/W	W			R/W
			0	0	0	0	1	0	0	1/0
			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" / fast / standard 0001:– 0010:– 0011: 8 0100: 9 0101: 10 0110:11 1000:fast 1111:standard other: Reserved			
		178H (no RMW) SIO mode	SIOS	SIOINH	SIOM1	SIOM0	-	SCK2	SCK1	SCK0
			W				W		W	
			0	0	0	0	1	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		Note) Write 0 to this bit in SIO mode.	Setting of the divide value "n" 000:4 001:5 010:6 011:7 100:8 101:9 110:10 111: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:Port 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		W	W
			-	-	-	-	0	0	0	0
							Operation mode selection 00:Port mode 10:I2C mode 01:SIO mode 11:reserved		Fix to "00"	
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
SBI2CR1	SBI2 Control Register 1	180H (no RMW) I2C mode	BC2	BC1	BC0	ACK	SCK3	SCK2	SCK1	SWRMON/SCK0
			W			R/W	W			R/W
			0	0	0	0	1	0	0	1/0
			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" / fast / standard 0001:- 0010:- 0011: 8 0100: 9 0101: 10 0110:11 1000:fast 1111:standard other: Reserved			
		180H (no RMW) SIO mode	SIOS	SIOINH	SIOM1	SIOM0	-	SCK2	SCK1	SCK0
			W				W		W	
			0	0	0	0	1	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		Note) Write 0 to this bit in SIO mode.	Setting of the divide value "n" 000:4 001:5 010:6 011:7 100:8 101:9 110:10 111:external clock SCK2		
SBI2DBR	SBI2 Buffer Register	181H (no RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R(Receiving)/W(Transmission)							
			Undefine							
I2C2AR	I2CBUS2 Address Register	182H (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
SBI2CR2	SBI2 Control Register 2	183H (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	INTSBE2 interrupt 0:Request 1:Cancel	Operation mode selection 00:Port mode 10:I2C mode 01:SIO mode 11:reserved		Software reset generate write "10" and "01", then an internal reset signal is generated.	
		183H (no RMW) SIO mode	-	-	-	-	SBIM1	SBIM0	-	-
							W		W	W
			-	-	-	-	0	0	0	0
							Operation mode selection 00:Port mode 10:I2C mode 01:SIO mode 11:reserved		Fix to "00"	
SBI2SR	SBI2 Status Register	183H (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	INTSBE2 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"
		183H (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	-	-	-	-	-	-
			W	R/W						
			0	0	-	-	-	-	-	-
			Fix to "0"	IDLE2 0:Abort 1:Operate						
SBI0BR1	SBI0 Baud rate Register 1	175H	P4EN	-	-	-	-	-	-	-
			R/W							
			0	-	-	-	-	-	-	-
			Baud rate control circuit 0:Abort 1:Operate							
SBI1BR0	SBI1 Baud rate Register 0	17CH	-	I2SBI0	-	-	-	-	-	-
			W	R/W						
			0	0	-	-	-	-	-	-
			Fix to "0"	IDLE2 0:Abort 1:Operate						
SBI1BR1	SBI1 Baud rate Register 1	17DH	P4EN	-	-	-	-	-	-	-
			R/W							
			0	-	-	-	-	-	-	-
			Baud rate control circuit 0:Abort 1:Operate							
SBI2BR0	SBI2 Baud rate Register 0	184H	-	I2SBI0	-	-	-	-	-	-
			W	R/W						
			0	0	-	-	-	-	-	-
			Fix to "0"	IDLE2 0:Abort 1:Operate						
SBI2BR1	SBI2 Baud rate Register 1	185H	P4EN	-	-	-	-	-	-	-
			R/W							
			0	-	-	-	-	-	-	-
			Baud rate control circuit 0:Abort 1:Operate							

(12) CAN controller (1/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
MBnMI0L	Message Identifier 0L	MBn* + 00H (no RMW)	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
			R/W							
			-	-	-	-	-	-	-	-
MBnMI0H	Message Identifier 0H	MBn* + 01H (no RMW)	IDE	GAME	RFH	ID28	ID27	ID26	ID25	ID24
			R/W							
			-	-	-	-	-	-	-	-
MBnMI1L	Message Identifier 1L	MBn* + 02H (no RMW)	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
			R/W							
			-	-	-	-	-	-	-	-
MBnMI1H	Message Identifier 1H	MBn* + 03H (no RMW)	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
			R/W							
			-	-	-	-	-	-	-	-
MBnMCFL	Message Control Field L	MBn* + 04H (no RMW)	-	-	-	RTR	DLC3	DLC2	DLC1	DLC0
						R/W				
			-	-	-	-	-	-	-	-
MBnMCFH	Message Control Field H	MBn* + 05H (no RMW)	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-
MBnD0	Data 0	MBn* + 06H (no RMW)	D07	D06	D05	D04	D03	D02	D01	D00
			R/W							
			-	-	-	-	-	-	-	-
MBnD1	Data 1	MBn* + 07H (no RMW)	D17	D16	D15	D14	D13	D12	D11	D10
			R/W							
			-	-	-	-	-	-	-	-
MBnD2	Data 2	MBn* + 08H (no RMW)	D27	D26	D25	D24	D23	D22	D21	D20
			R/W							
			-	-	-	-	-	-	-	-
MBnD3	Data 3	MBn* + 09H (no RMW)	D37	D36	D35	D34	D33	D32	D31	D30
			R/W							
			-	-	-	-	-	-	-	-
MBnD4	Data 4	MBn* + 0AH (no RMW)	D47	D46	D45	D44	D43	D42	D41	D40
			R/W							
			-	-	-	-	-	-	-	-
MBnD5	Data 5	MBn* + 0BH (no RMW)	D57	D56	D55	D54	D53	D52	D51	D50
			R/W							
			-	-	-	-	-	-	-	-
MBnD6	Data 6	MBn* + 0CH (no RMW)	D67	D66	D65	D64	D63	D62	D61	D60
			R/W							
			-	-	-	-	-	-	-	-
MBnD7	Data 7	MBn* + 0DH (no RMW)	D77	D76	D75	D74	D73	D72	D71	D70
			R/W							
			-	-	-	-	-	-	-	-
MBnTSVL	Time Stamp Value L	MBn* + 0EH	TSV7	TSV6	TSV5	TSV4	TSV3	TSV2	TSV1	TSV0
			R							
			-	-	-	-	-	-	-	-
MBnTSVH	Time Stamp Value H	MBn* + 0FH	TSV15	TSV14	TSV13	TSV12	TSV11	TSV10	TSV9	TSV8
			R							
			-	-	-	-	-	-	-	-

Note: MBn = 200H + n × 10H, n = 0 to 15

CAN controller (2/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
MCL	Mailbox Configuration Register L	300H	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
			R/W							
			0	0	0	0	0	0	0	0
MCH	Mailbox Configuration Register H	301H	MC15	MC14	MC13	MC12	MC11	MC10	MC9	MC8
			R/W							
			0	0	0	0	0	0	0	0
MDL	Mailbox Direction Register L	302H	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
			R/W							
			0	0	0	0	0	0	0	0
MDH	Mailbox Direction Register H	303H	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8
			R	R/W						
			1	0	0	0	0	0	0	0
TRSL	Transmission Request Set Register L	304H (no RMW)	TRS7	TRS6	TRS5	TRS4	TRS3	TRS2	TRS1	TRS0
			R/S							
			0	0	0	0	0	0	0	0
TRSH	Transmission Request Set Register H	305H (no RMW)	-	TRS14	TRS13	TRS12	TRS11	TRS10	TRS9	TRS8
			R/S							
			-	0	0	0	0	0	0	0
TRRL	Transmission Request Reset Register L	306H (no RMW)	TRR7	TRR6	TRR5	TRR4	TRR3	TRR2	TRR1	TRR0
			R/S							
			0	0	0	0	0	0	0	0
TRRH	Transmission Request Reset Register H	307H (no RMW)	-	TRR14	TRR13	TRR12	TRR11	TRR10	TRR9	TRR8
			R/S							
			-	0	0	0	0	0	0	0
TAL	Transmission Acknowledge Register L	308H (no RMW)	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
			R/C							
			0	0	0	0	0	0	0	0
TAH	Transmission Acknowledge Register H	309H (no RMW)	-	TA14	TA13	TA12	TA11	TA10	TA9	TA8
			R/C							
			-	0	0	0	0	0	0	0
AAL	Abort Acknowledge Register L	30AH (no RMW)	AA7	AA6	AA5	AA4	AA3	AA2	AA1	AA0
			R/C							
			0	0	0	0	0	0	0	0
AAH	Abort Acknowledge Register H	30BH (no RMW)	-	AA14	AA13	AA12	AA11	AA10	AA9	AA8
			R/C							
			-	0	0	0	0	0	0	0
RMPL	Receive Message Pending Register L	30CH (no RMW)	RMP7	RMP6	RMP5	RMP4	RMP3	RMP2	RMP1	RMP0
			R/C							
			0	0	0	0	0	0	0	0
RMPH	Receive Message Pending Register H	30DH (no RMW)	RMP15	RMP14	RMP13	RMP12	RMP11	RMP10	RMP9	RMP8
			R/C							
			0	0	0	0	0	0	0	0
RMLL	Receive Message Lost Register L	30EH	RML7	RML6	RML5	RML4	RML3	RML2	RML1	RML0
			R							
			0	0	0	0	0	0	0	0
RMLH	Receive Message Lost Register H	30FH	RML15	RML14	RML13	RML12	RML11	RML10	RML9	RML8
			R							
			0	0	0	0	0	0	0	0

CAN controller (3/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LAM0L	Local Accept-ance Mask Register 0L	310H	LAM23	LAM22	LAM21	LAM20	LAM19	LAM18	LAM17	LAM16
			R/W							
			0	0	0	0	0	0	0	0
LAM0H	Local Accept-ance Mask Register 0H	311H	LAMI	-	-	LAM28	LAM27	LAM26	LAM25	LAM24
			R/W			R/W				
			0	-	-	0	0	0	0	0
LAM1L	Local Accept-ance Mask Register 1L	312H	LAM7	LAM6	LAM5	LAM4	LAM3	LAM2	LAM1	LAM0
			R/W							
			0	0	0	0	0	0	0	0
LAM1H	Local Accept-ance Mask Register 1H	313H	LAM15	LAM14	LAM13	LAM12	LAM11	LAM10	LAM9	LAM8
			R/W							
			0	0	0	0	0	0	0	0
GAM0L	Global Acceptance Mask Register 0L	314H	GAM23	GAM22	GAM21	GAM20	GAM19	GAM18	GAM17	GAM16
			R/W							
			0	0	0	0	0	0	0	0
GAM0H	Global Acceptance Mask Register 0H	315H	GAMI	-	-	GAM28	GAM27	GAM26	GAM25	GAM24
			R/W			R/W				
			0	-	-	0	0	0	0	0
GAM1L	Global Acceptance Mask Register 1L	316H	GAM7	GAM6	GAM5	GAM4	GAM3	GAM2	GAM1	GAM0
			R/W							
			0	0	0	0	0	0	0	0
GAM1H	Global Acceptance Mask Register 1H	317H	GAM15	GAM14	GAM13	GAM12	GAM11	GAM10	GAM9	GAM8
			R/W							
			0	0	0	0	0	0	0	0
MCRL	Master Control Register L	318H	CCR	SMR	HMR	WUBA	MTOS	-	TSCC	SRES
			R/W						W	
			1	0	0	0	0	-	0	0
MCRH	Master Control Register H	319H	-	-	-	-	-	-	TSTLB	TSTERR
									R/W	
			-	-	-	-	-	-	0	0
GSRL	Global Status Register L	31AH	CCE	SMA	HMA	-	TSO	BO	EP	EW
			R				R			
			1	0	0	-	0	0	0	0
GSRH	Global Status Register H	31BH	MsgInSlot<3:0>				RM	TM	-	-
			R							
			1	1	1	1	0	0	-	-
BCR1L	Bit Configuration Register 1L	31CH	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
			R/W							
			0	0	0	0	0	0	0	0
BCR1H	Bit Configuration Register 1H	31DH	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-
BCR2L	Bit Configuration Register 2L	31EH	SAM	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
			R/W							
			0	0	0	0	0	0	0	0
BCR2H	Bit Configuration Register 2H	31FH	-	-	-	-	-	-	SJW1	SJW0
									R/W	
			-	-	-	-	-	-	0	0

CAN controller (4/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
GIFL	Global Interrupt Flag L	320H (no RMW)	RFPF	WUIF	RMLIF	TRMABF	TSOIF	BOIF	EPIF	WLIF
			R/C							
			0	0	0	0	0	0	0	0
GIFH	Global Interrupt Flag H	321H (no RMW)	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-
GIML	Global Interrupt Mask L	322H	RFPM	WUIM	RMLIM	TRMABM	TSOIM	BOIM	EPIM	WLIM
			R/W							
			0	0	0	0	0	0	0	0
GIMH	Global Interrupt Mask H	323H	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-
MBTIFL	Mailbox Transmit Int. Flag L	324H (no RMW)	MBTIF7	MBTIF6	MBTIF5	MBTIF4	MBTIF3	MBTIF2	MBTIF1	MBTIF0
			R/C							
			0	0	0	0	0	0	0	0
MBTIFH	Mailbox Transmit Int. Flag H	325H (no RMW)	-	MBTIF14	MBTIF13	MBTIF12	MBTIF11	MBTIF10	MBTIF9	MBTIF8
			R/C							
			-	0	0	0	0	0	0	0
MBRIFL	Mailbox Receive Int. Flag L	326H (no RMW)	MBRIF7	MBRIF6	MBRIF5	MBRIF4	MBRIF3	MBRIF2	MBRIF1	MBRIF0
			R/C							
			0	0	0	0	0	0	0	0
MBRIFH	Mailbox Receive Int. Flag H	327H (no RMW)	MBRIF15	MBRIF14	MBRIF13	MBRIF12	MBRIF11	MBRIF10	MBRIF9	MBRIF8
			R/C							
			0	0	0	0	0	0	0	0
MBIML	Mailbox Interrupt Flag L	328H	MBIM7	MBIM6	MBIM5	MBIM4	MBIM3	MBIM2	MBIM1	MBIM0
			R/W							
			0	0	0	0	0	0	0	0
MBIMH	Mailbox Interrupt Flag H	329H	MBIM15	MBIM14	MBIM13	MBIM12	MBIM11	MBIM10	MBIM9	MBIM8
			R/W							
			0	0	0	0	0	0	0	0
CDRL	Change Data Request Register L	32AH	CDR7	CDR6	CDR5	CDR4	CDR3	CDR2	CDR1	CDR0
			R/W							
			0	0	0	0	0	0	0	0
CDRH	Change Data Request Register H	32BH	-	CDR14	CDR13	CDR12	CDR11	CDR10	CDR9	CDR8
			R/W							
			-	0	0	0	0	0	0	0
RFPL	Remote Frame Pending Register L	32CH	RFP7	RFP6	RFP5	RFP4	RFP3	RFP2	RFP1	RFP0
			R							
			0	0	0	0	0	0	0	0
RFPH	Remote Frame Pending Register H	32DH	RFP15	RFP14	RFP13	RFP12	RFP11	RFP10	RFP9	RFP8
			R							
			-	-	-	-	-	-	-	-
CECL	CAN Error Counter L	32EH (no RMW)	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
			R/W							
			0	0	0	0	0	0	0	0
CECH	CAN Error Counter H	32FH (no RMW)	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
			R/W							
			0	0	0	0	0	0	0	0

CAN controller (5/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TSPL	Time Stamp Prescaler L	330H	-	-	-	-	TSP3	TSP2	TSP1	TSP0
							R/W			
			-	-	-	-	0	0	0	0
TSPH	Time Stamp Prescaler H	331H	-	-	-	-	-	-	-	-
			-	-	-	-	-	-	-	-
TSCL	Time Stamp Counter L	332H (no RMW)	TSC7	TSC6	TSC5	TSC4	TSC3	TSC2	TSC1	TSC0
			R/W							
			0	0	0	0	0	0	0	0
TSCH	Time Stamp Counter H	333H (no RMW)	TSC15	TSC14	TSC13	TSC12	TSC11	TSC10	TSC9	TSC8
			R/W							
			0	0	0	0	0	0	0	0

(13) RTC

Symbol	Name	Address	7	6	5	4	3	2	1	0
RTCCR	RTC Control Register	118h	-	-	-	-	RTCSEL2	RTCSEL1	RTCSEL0	RTCRUN
			R/W				R/W			R/W
			0	-	-	-	0	0	0	0
			Write to "0"				1x0: 2^{16}/fs 1x1: 2^{15}/fs	00: 2^{14}/fs 01: 2^{13}/fs 10: 2^{12}/fs 11: 2^{11}/fs		0: Stop & Clear 1: Run
RTCFC	RTC Function Control Register	119h	XTSEL	-	-	-	-	-	-	XTEN
			R/W							R/W
			0	-	-	-	-	-	-	0
			0:Crystal 1:CR							Low frequency Oscillator (fs) 1:oscillation

6. Port Equivalent Circuits

- Circuit diagram convention

    Basically, the circuit diagrams use the same gate symbols as those used for the 74HCXX standard CMOS logic IC series.

    A special signal name is as follows:

    STOP: This signal is activated (set to 1) when the CPU executes the HALT instruction with STOP mode specified in the halt mode setup register (CLKMOD<HALTM1: 0> = 0, 1). The STOP signal, however, remains set to 0 if the driver enable bit, WDMOD<DRVE>, is set to 1.

- Input protection resistance is approximately several tens of ohms to several hundreds of ohms.

■ P0 (D0 to D7), P4 (A0 to A7), P70, P71, P73 to P75, PC0 to PC5, PD0 to PD7, PF1(RXD0), PF2 (CTS0, SCLK0), PF4 (RXD1), PF5 ($\overline{\text{CTS1}}$, SCLK1), PF6 (TX), PF7 (RX), PM0 ($\overline{\text{SS}}$), PN0 (SCK0), PN3 (SCK1), PM4 (SCK2)
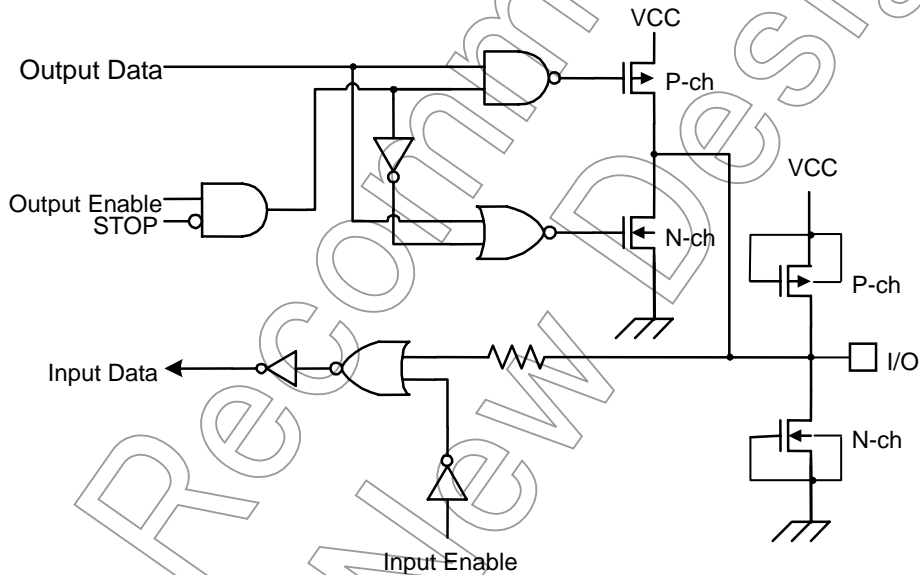
■ P72 (SI2/SCL2), PF0 (TXD0), PF3 (TXD1), PM1 (MOSI), PM2 (MISO), PM3 (SECLK), PN1 (SO0/SDA0), PN2 (SI0/SCL0), PN4 (SO1/SDA1), PN5 (SI1/SCL1), PN6 (SO2/SDA2)
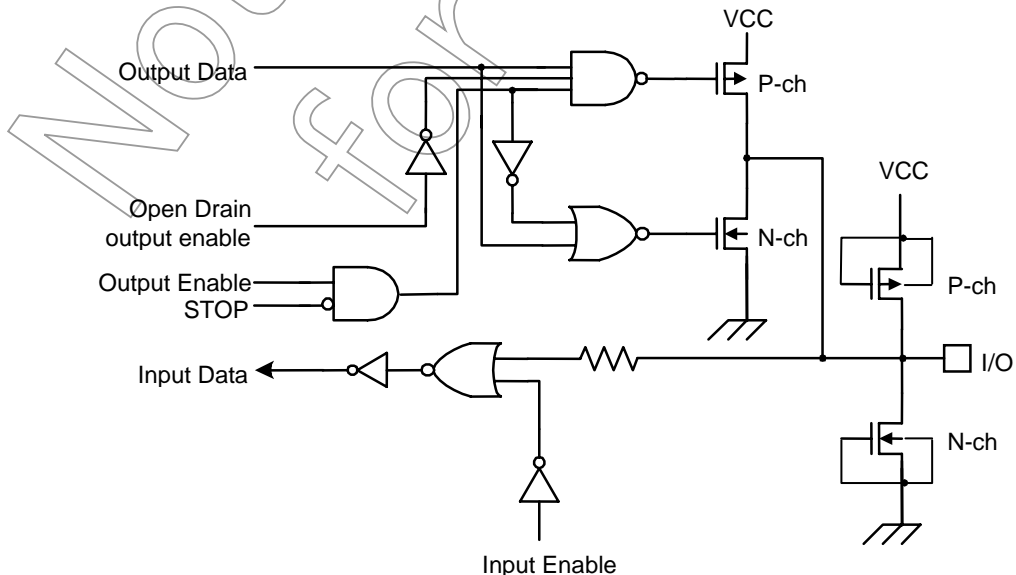
■ PG(AN0 to 7), PL0 to 3(AN8 to 11)

■ INT0
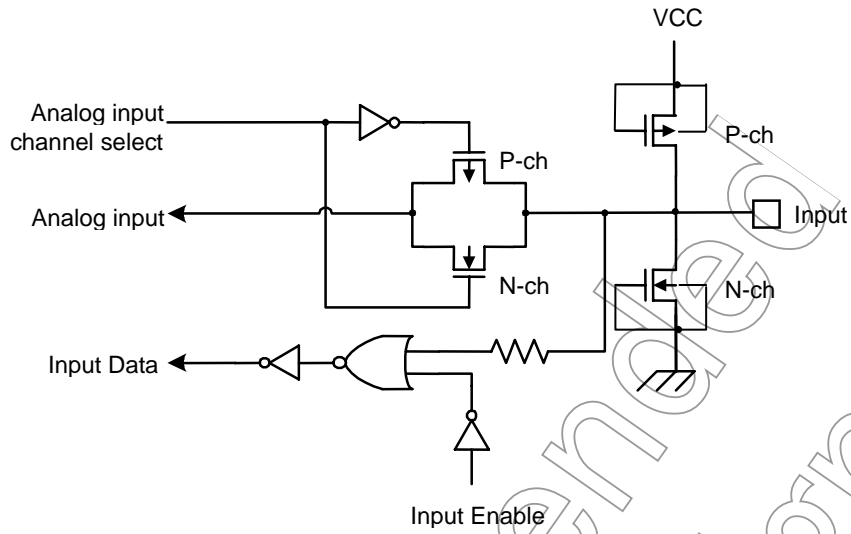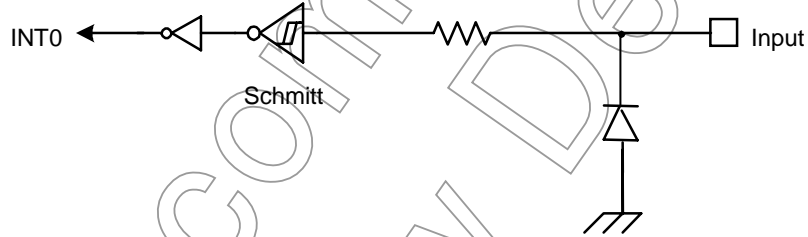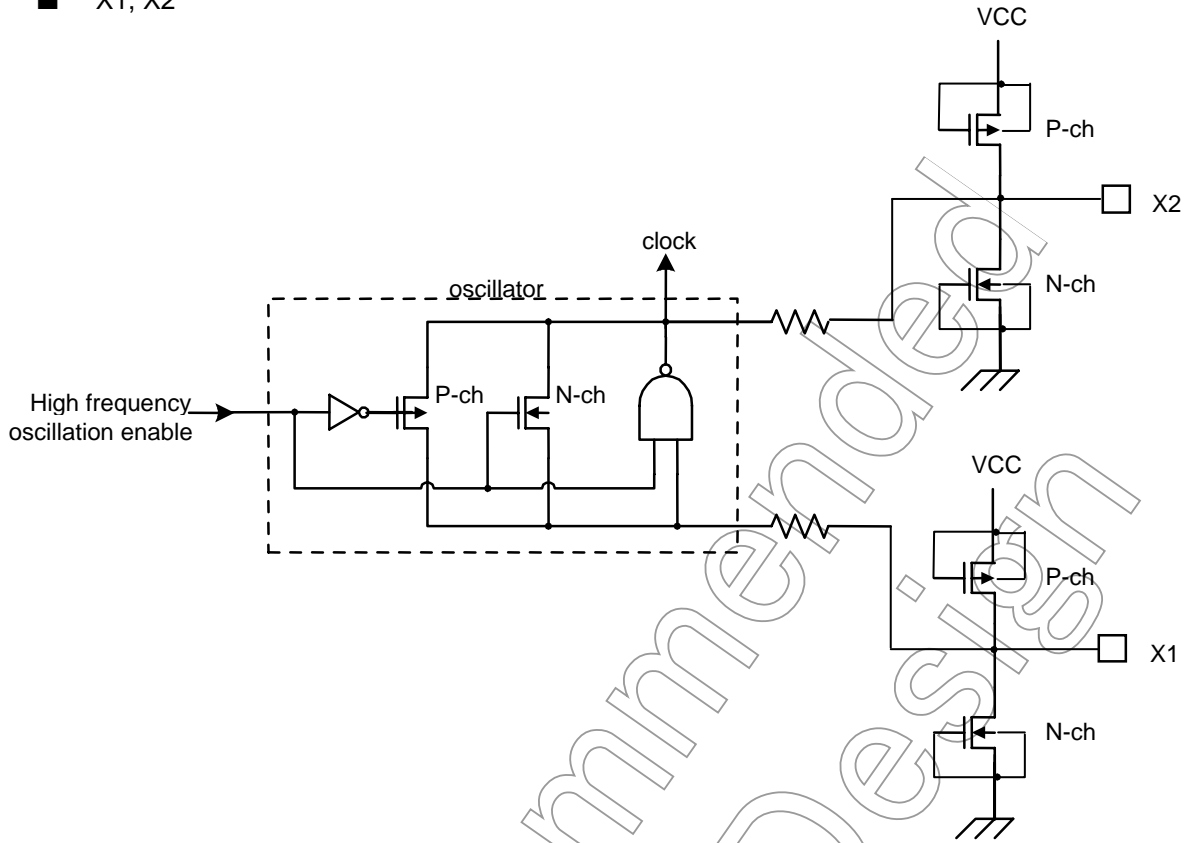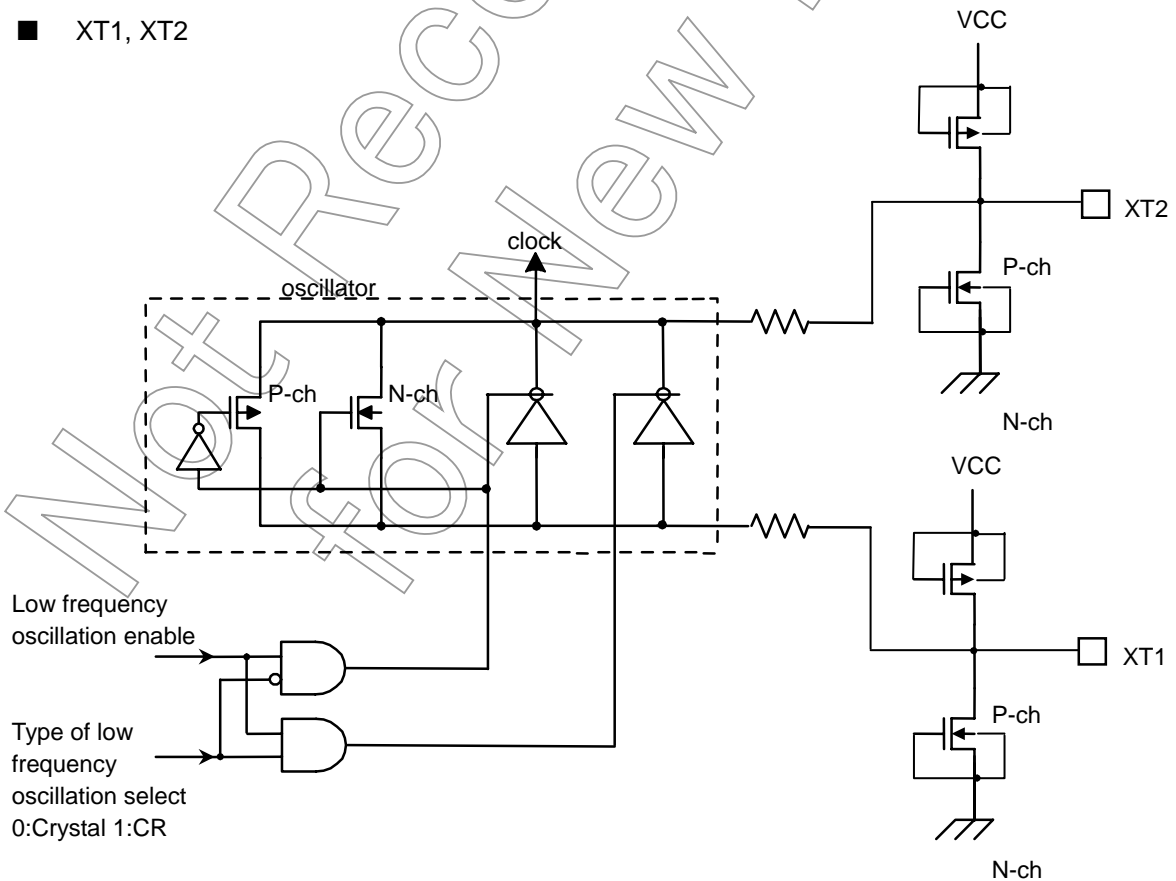
■ $\overline{\text{RESET}}$

■ X1, X2

■ XT1, XT2

■ VREFH, VREFL
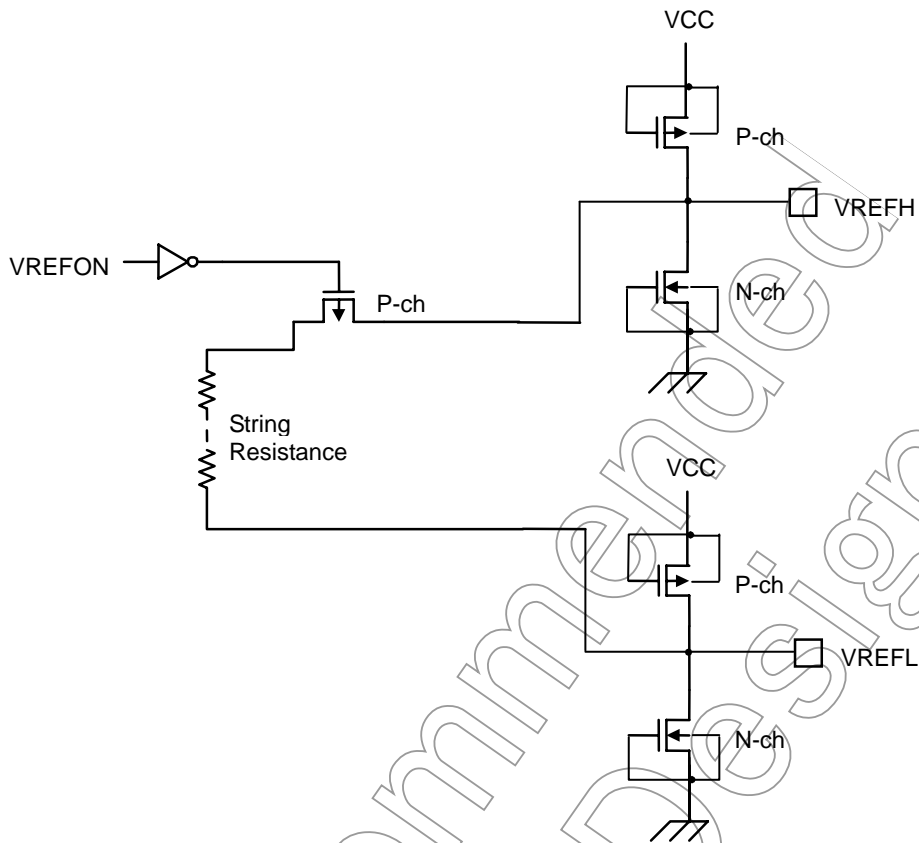
■ $\overline{\text{NMI}}$
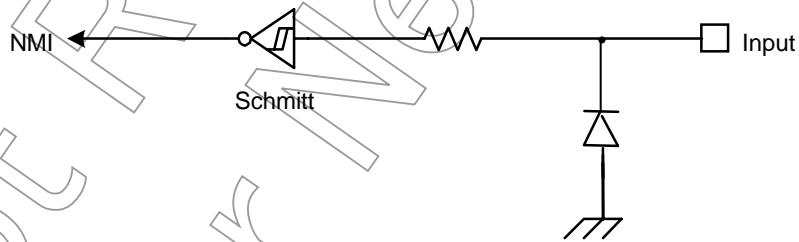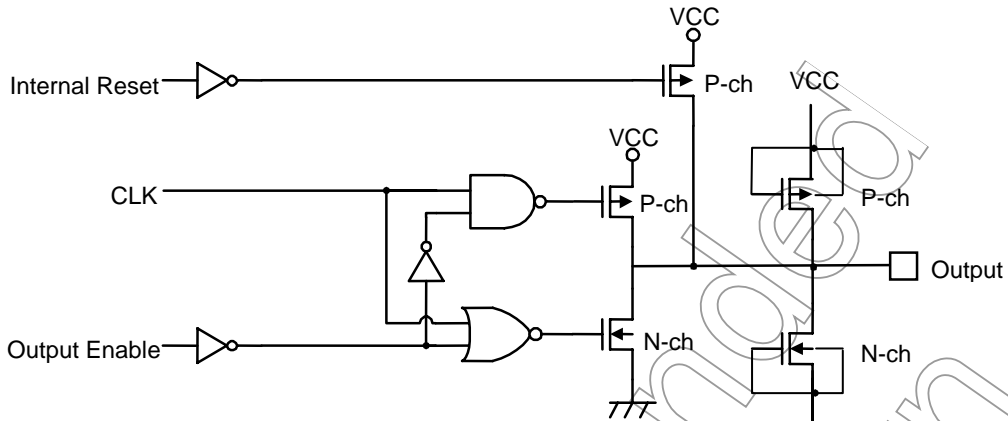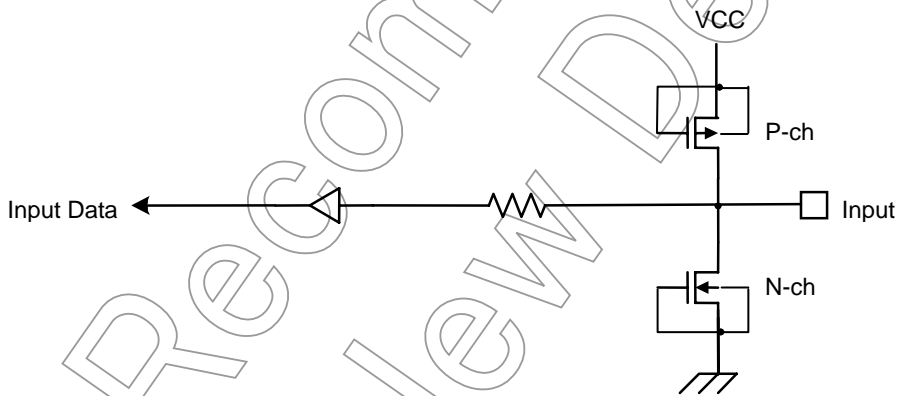
■   CLK

■   AM0 to 1, TEST0 to 1

■   REGOUT
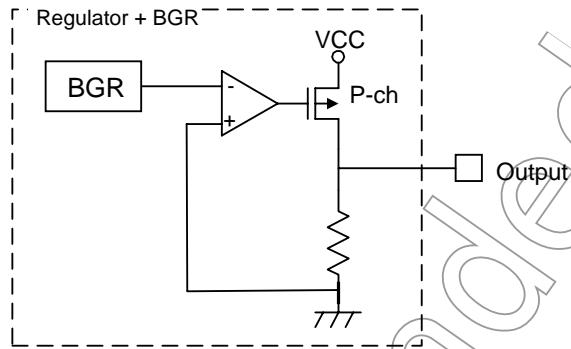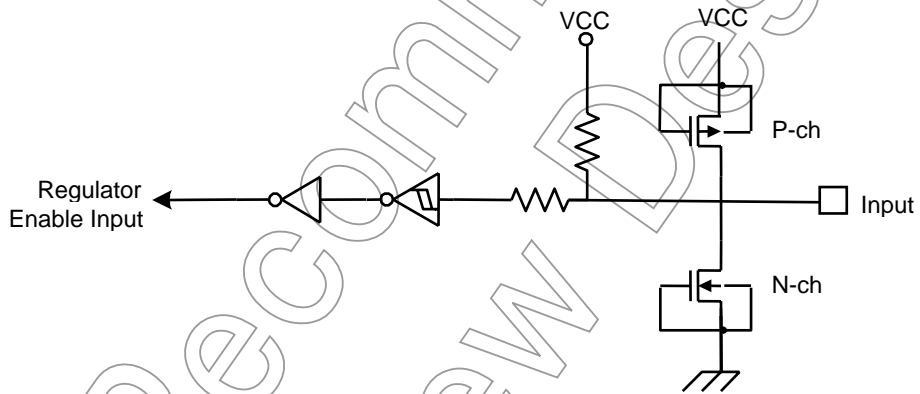
■   REGEN

7. Handling Precautions and Restrictions

（1） Special representations and terms

　　1. Description of built-in I/O register: register-symbol<bit-symbol>

　　　Example: TRUN01<T0RUN> indicates bit T0RUN in register TRUN.

　　2. Read-modify-write instruction (RMW)

　　　A read-modify-write instruction is a single instruction executed by the CPU that reads data from a memory address, manipulates the data and then writes the data back to the same memory address.

　　　Example 1：SET 3, (TRUN01) ... Sets bit 3 in the TRUN01 register.

　　　Example 2：INC 1, (100H)　　... Increments data at address 100H by one.

- Read-modify-write instructions in TLCS-900

  Exchange instruction

  　EX　　(mem), R

  Arithmetic instructions

  　ADD　(mem), R/#　　　ADC　　(mem), R/#

  　SUB　(mem), R/#　　　SBC　　(mem), R/#

  　INC　#3, (mem)　　　　DEC　　#3, (mem)

  Logical operation

  　AND　(mem), R/#　　　OR　　 (mem), R/#

  　XOR　(mem), R/#

  Bit manipulation

  　STCF　#3/A, (mem)　　RES　　#3, (mem)

  　SET　#3, (mem)　　　　CHG　　#3, (mem)

  　TSET　#3, (mem)

  　XOR　(mem), R/#

  Rotate and shift

  　RLC　(mem)　　　　　　RRC　　(mem)

  　RL　 (mem)　　　　　　RR　　 (mem)

  　SLA　(mem)　　　　　　SRA　　(mem)

  　SLL　(mem)　　　　　　SRL　　(mem)

  　RLD　(mem)　　　　　　RRD　　(mem)

(2) Handling precautions and restrictions

  a) Watchdog timer

    Upon a reset, the watchdog timer is enabled. It should be disabled if it is not used for operation.

  b) Clock settling time

    After an external reset is released, the device waits during the settling time for the clock multiplier within the device before starting operation. For details, see "3.1.2 Reset." When the device is restored from STOP standby mode with an interrupt, an oscillator settling time and other intervals are automatically inserted before the internal circuitry starts operation. For details, see "④ STOP mode" in "(3) Operation in each mode" in Section 3.4, "Standby Controller."

  c) Undefined SFR bits

    Undefined bits in special function registers (SFRs) return undefined values when read. The program should not depend on the states of those bits.

  d) Reserved areas in address space

    The 16-byte space from FFFFF0H to FFFFFFH is reserved as an internal area and cannot be used. When an emulator is used, 64 Kbytes of the 16-Mbyte space are used to control the emulator and not available to the user.

  e) POP SR instruction

    The POP SR instruction should be executed in DI (interrupt disabled) state.

8. Package

Package Dimensions : LQFP100-P-1414-0.50F
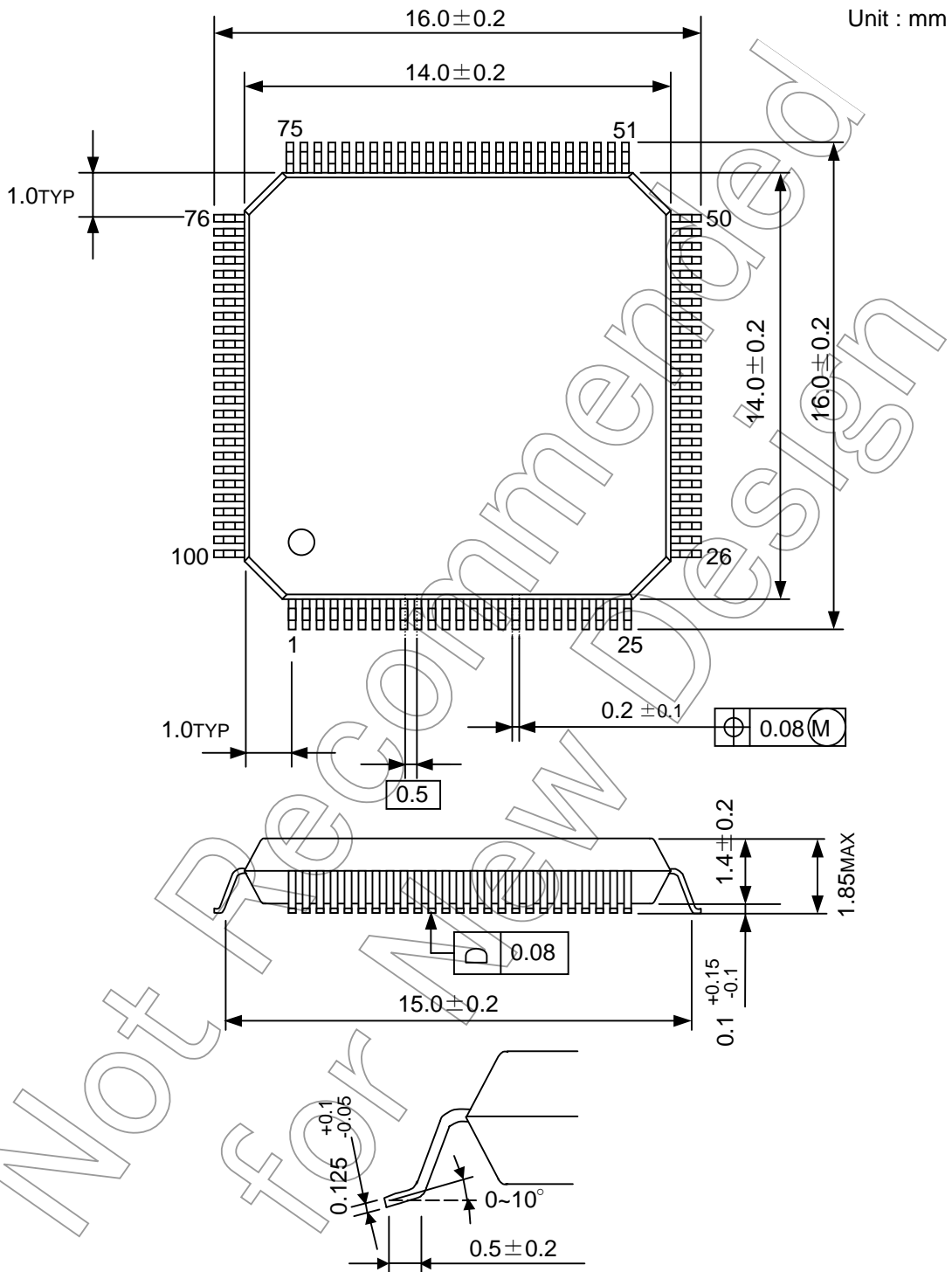
Note1: The drawings shown may not accurately represent the actual shape or dimensions.

RESTRICTIONS ON PRODUCT USE