CMOS 32-bit Micro-controller

TMP92CD54IF

1. Outline and Device Characteristics

TMP92CD54I is high-speed advanced 32-bit micro-controller developed for controlling equipment which processes mass data.

TMP92CD54I is a micro-controller which has a high-performance CPU (900/H1 CPU) and various built-in I/Os. TMP92CD54I is housed in a 100-pin mini flat package.

Device characteristics are as follows:

(1) CPU : 32-bit CPU(900/H1 CPU)
- Compatible with TLCS-900,900/L,900/L1,900/H,900/H2's instruction code
- 16Mbytes of linear address space
- General-purpose register and register banks
- Micro DMA : 8channels (250ns / 4bytes at fc = 20MHz, best case)
- Minimum instruction execution time : 50ns(at 20MHz)
- Internal data bus : 32-bit

(2) Internal memory
- Internal RAM : 32K-byte
- Internal ROM : 512K-byte Mask ROM

(3) External memory expansion

16M-byte linear address space (memory mapped I/O)
External data bus : 8bit(for external I/O expansion)
* Can't use upper address bus when 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
16-bit event counter mode
16-bit programmable pulse generation (PPG) output mode
Frequency measurement mode
Pulse width measurement mode
Time differential 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 4/2/0.5Mbps at fc=20MHz.

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

Clocked-synchronous 8-bit serial interface mode
I^2C bus mode

(10) CAN controller : 1channel

Supports CAN version 2.0B.
16 mailboxes

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

A/D conversion time 8μsec @fc=20MHz.
Total tolerance +/- 3LSB (excluding quantization error)
Scan mode for all 12channels

(12) Watch dog timer (WDT)

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

Can operate with only low frequency oscillator.

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

9 interrupts from CPU
42 internal interrupt vectors
9 external interrupt vectors

(15) I/O Port : 68pins

(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

VCC5 = 4.5V to 5.25V

VCC3 = 3.3V (VCC3 Connect to REGOUT; built-in voltage regulator.)

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

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

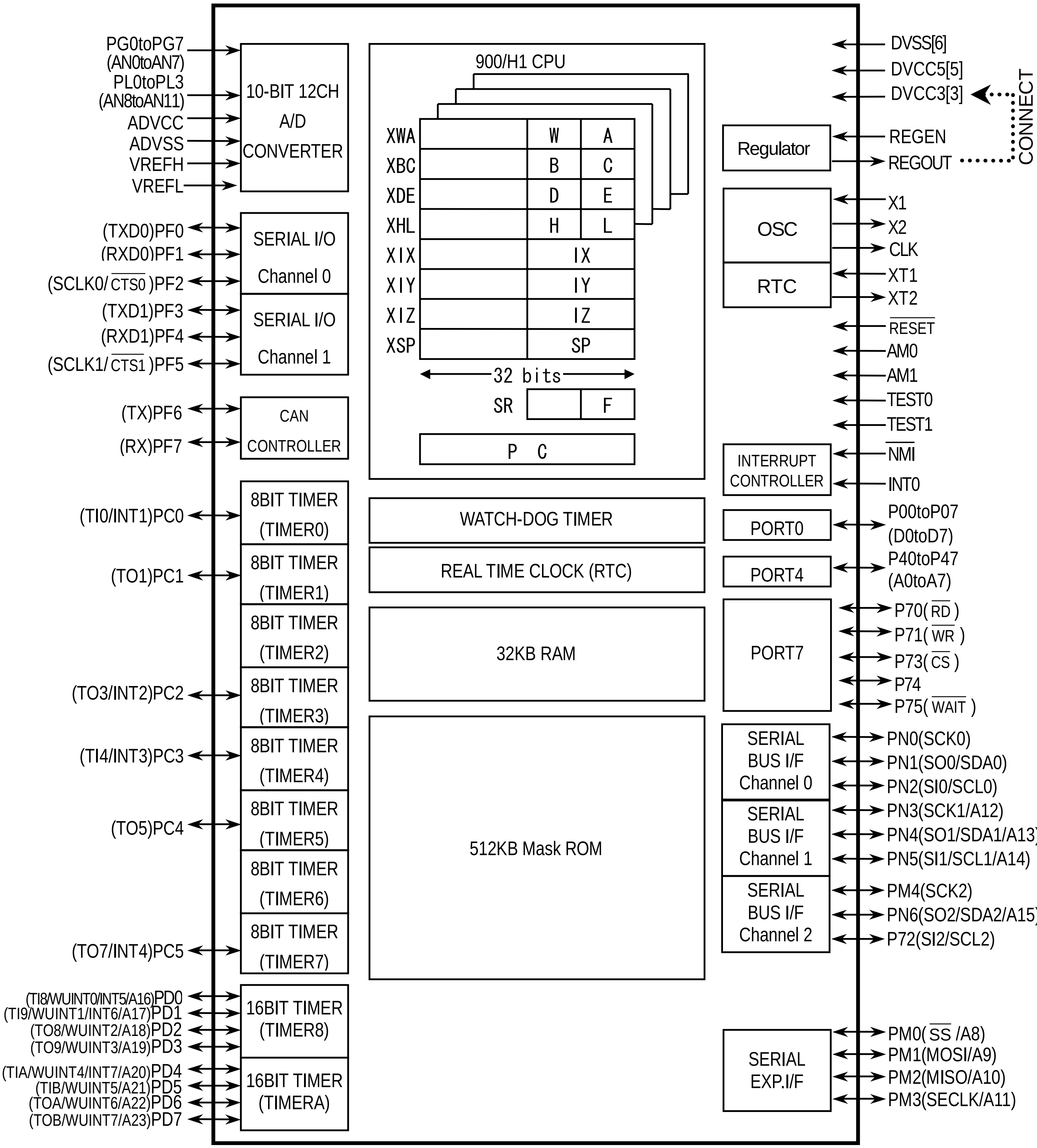


Figure 1 TMP92CD54I block diagram

2. Pin Assignment and Functions

2.1 Pin Assignment

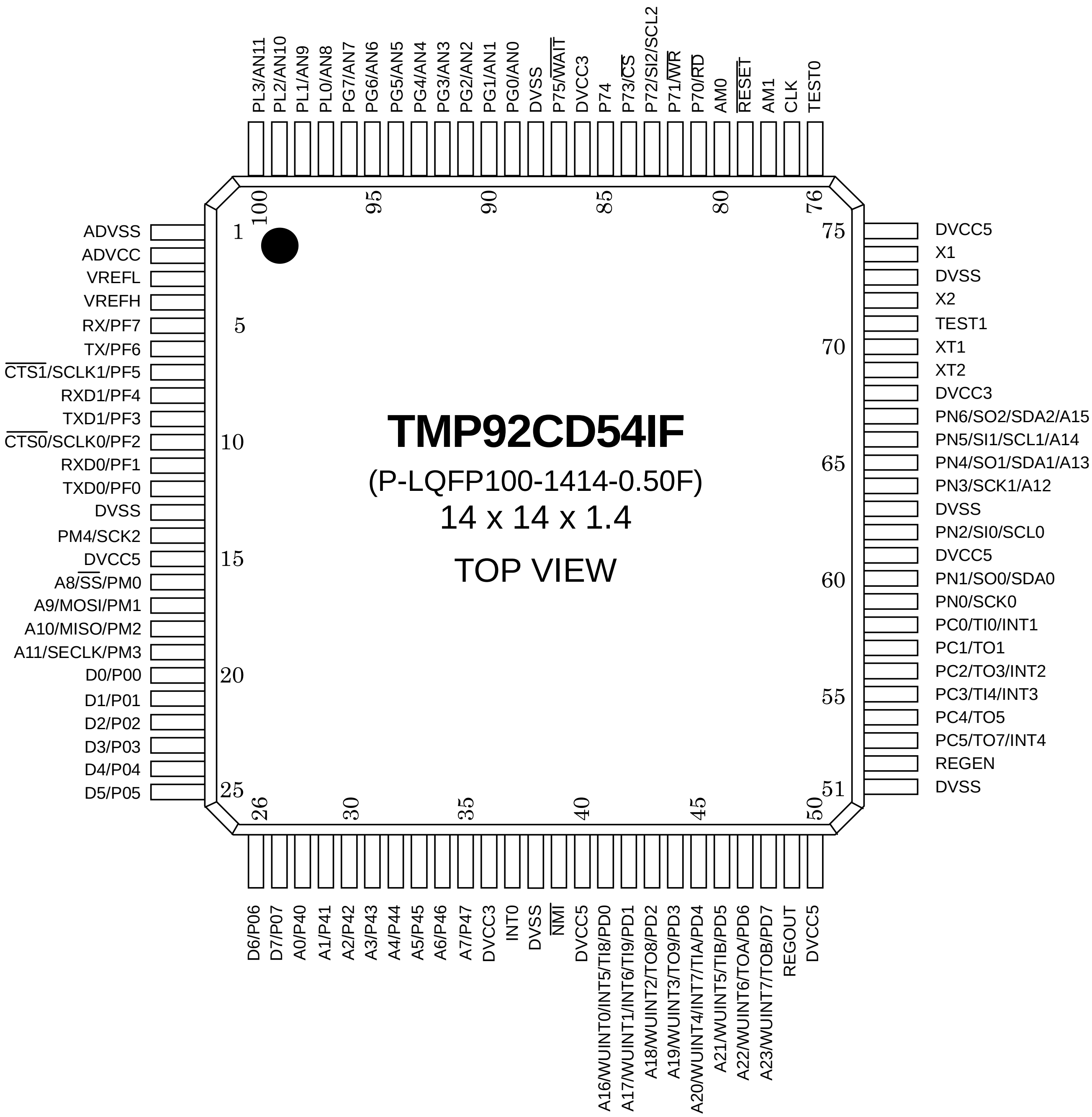
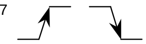
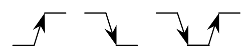


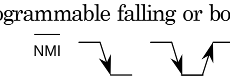
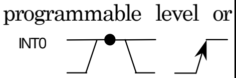
Figure 2.1 TMP92CD54I Pin Assignment

2.2 Pin names and functions

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

Pin name	Pin number	Number of pins	In/Out	Function
P00..P07 D0..D7	20^{th}…27^{th}	8 (CMOS) (TTL)	in/out in/out	Port 0: I/O port. Input or output specifiable in units of bits. Data: Data bus 0 to 7.
P40..P47 A0..A7	28^{th}…35^{th}	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^{st}	1	in/out out	Port70: I/O port. Read: Outputs strobe signal to read external memory.
P71 $\overline{WR}$	82^{nd}	1	in/out out	Port 71: I/O port. Write: Output strobe signal to write external memory.
P72 SI2 SCL2	83^{rd}	1	in/out	Port 72: I/O port. SBI channel 2: Input data at SIO mode SBI channel 2: Clock input/output at I^2C mode
P73 $\overline{CS}$	84^{th}	1	in/out out	Port 73: I/O port. Chip select: Outputs "low" if address is within specified address area.
P74	85^{th}	1	in/out	Port 74: I/O port.
P75 $\overline{WAIT}$	87^{th}	1	in/out in	Port 75: I/O port. Wait: Signal used to request CPU bus wait.
PC0 TI0 INT1	58^{th}	1	in/out in in	Port C0: I/O port. Timer input 0: Input pin for timer 0. INT1 Interrupt request pin 1: Rising-edge interrupt request pin.
PC1 TO1	57^{th}	1	in/out out	Port C1: I/O port. Timer output 1: Output pin for timer 1.
PC2 TO3 INT2	56^{th}	1	in/out out in	Port C2: I/O port. Timer output 3: Output pin for timer 3. INT2 Interrupt request pin 2: Rising-edge interrupt request pin.
PC3 TI4 INT3	55^{th}	1	in/out in in	Port C3: I/O port. Timer input 4: Input pin for timer 4. INT3 Interrupt request pin 3: Rising-edge interrupt request pin.
PC4 TO5	54^{th}	1	in/out out	Port C4: I/O port. Timer output 5: Output pin for timer 5.
PC5 TO7 INT4	53^{rd}	1	in/out out in	Port C5: I/O port. Timer output 7: Output pin for timer 7. INT4 Interrupt request pin 4: Rising-edge interrupt request pin.
PD0 TI8 INT5 A16 WUINT0	41^{st}	1	in/out in in out in	Port D0: I/O port. Timer input 8: Input pin for timer 8. INT5 Interrupt request pin 5: Interrupt request pin with programmable rising/falling edge. WUINT0 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^{nd}	1	in/out in in out in	Port D1: I/O port. Timer input 9: Input pin for timer 9. INT6 WUINT1 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^{rd}	1	in/out out out in	Port D2: I/O port. Timer output 8: Output pin for timer 8 WUINT2 Address: Address bus 18. Wake up input 2: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD3 TO9 A19 WUINT3	44^{th}	1	in/out out out in	Port D3: I/O port. Timer output 9: Output pin for timer 9 WUINT3 Address: Address bus 19. Wake up input 3: Wake up request pin with programmable rising, falling or both falling and rising edge.

Pin name	Pin number	Number of pins	In/Out	Function
PD4 TIA INT7 A20 WUINT4	45th	1	in/out in in out in	Port D4: I/O port. Timer input A: Input pin for timer A INT7  Interrupt request pin 7: Interrupt request pin with programmable rising/falling edge. Address: Address bus 20. WUINT4  Wake up input 4: Wake up request pin with programmable rising, falling or both falling and rising edge.
PD5 TIB A21 WUINT5	46th	1	in/out in out in	Port D5: I/O port. Timer input B: Input pin for timer B. WUINT5  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	47th	1	in/out out out in	Port D6: I/O port. Timer output A: Output pin for timer A. WUINT6  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	48th	1	in/out out out in	Port D7: I/O port. Timer output B: Output pin for timer B. WUINT7  Address: Address bus 23. Wake up input 7: Wake up request pin with programmable rising, falling or both falling and rising edge.
PF0 TXD0	12th	1	in/out out	Port F0: I/O port. Serial interface channel 0: Transmission data.
PF1 RXD0	11th	1	in/out in	Port F1: I/O port. Serial interface channel 0: Receive data.
PF2 SCLK0 $\overline{\text{CTS0}}$	10th	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	9th	1	in/out out	Port F3: I/O port. Serial interface channel 1: Transmission data.
PF4 RXD1	8th	1	in/out in	Port F4: I/O port. Serial interface channel 1: Receive data.
PF5 SCLK1 $\overline{\text{CTS1}}$	7th	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	6th	1	in/out out	Port F6: I/O port. CAN: Transmission data.
PF7 RX	5th	1	in/out in	Port F7: I/O port. CAN: Receive data.
PG0..PG7 AN0..AN7	89th…96th	8	in in	Port G: Input-only port. Analog input 0 to 7: AD converter input pins.
PL0..PL3 AN8..AN11	97th…100th	4	in in	Port L0 to L3: Input-only port. Analog input 8 to 11: AD converter input pins.
PM0 $\overline{\text{SS}}$ A8	16th	1	in/out in out	Port M0: I/O port. SEI: Slave select input. Address: Address bus 8.
PM1 MOSI A9	17th	1	in/out in/out out	Port M1: I/O port. SEI: Master output, slave input. Address: Address bus 9.
PM2 MISO A10	18th	1	in/out in/out out	Port M2: I/O port. SEI: Master input, slave output. Address: Address bus 10.
PM3 SECLK A11	19th	1	in/out in/out out	Port M3: I/O port. SEI: Clock input/output. Address: Address bus 11.
PM4 SCK2	14th	1	in/out in/out	Port M4: I/O port. SBI channel 2: Clock input/output at SIO mode.
PN0 SCK0	59th	1	in/out in/out	Port N0: I/O port. SBI channel 0: Clock input/output at SIO mode.

Pin name	Pin number	Number of pins	In/Out	Function
PN1 SO0 SDA0	60^{th}	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 I^2C mode
PN2 SI0 SCL0	62^{nd}	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 I^2C mode
PN3 SCK1 A12	64^{th}	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^{th}	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 I^2C mode Address: Address bus 13.
PN5 SI1 SCL1 A14	66^{th}	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 I^2C mode Address: Address bus 14
PN6 SO2 SDA2 A15	67^{th}	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.
$\overline{NMI}$	39^{th}	1	in	Non-maskable interrupt: Interrupt request pin with programmable falling or both falling and rising edge. 
INT0	37^{th}	1	in	Interrupt request pin 0: Interrupt request pin with programmable level or rising-edge. 
AM0,1	80^{th}, 78^{th}	2	in	Address Mode selection: Connect AM0 pin to L, AM1 pins to H.
TEST0,1	76^{th}, 71^{st}	2	in	Test mode pins: Should be set to L.
CLK	77^{th}	1	out	Programmable clock output (with pull-up register)
X1/X2	74^{th}, 72^{nd}	2	in/out	Oscillator connecting pins
XT1/XT2	70^{th}, 69^{th}	2	in/out	Low frequency oscillator connecting pins. Crystal or ceramic resonator is connected. RC oscillation is also possible
$\overline{RESET}$	79^{th}	1	in	Reset: Initializes LSI (with pull-up register).
VREFH	4^{th}	1	in	AD reference voltage high
VREFL	3^{rd}	1	in	AD reference voltage low
ADVCC	2^{nd}	1	-	Power supply pin for AD converter (+5V): Connect ADVCC pin to 5V power supply.
ADVSS	1^{st}	1	-	GND pin for AD converter: Connect ADVSS pin to GND (0V).
DVCC5	15^{th}, 40^{th}, 50^{th},61^{st},75^{th}	5	-	Power supply pins (+5V): Connect all DVCC5 pins to 5V power supply.
DVCC3	36^{th},68^{th},86^{th}	3	-	Power supply pins (+3.3V): Connect all DVCC3 pins to REGOUT pin.
DVSS	13^{th},38^{th},51^{st}, 63^{rd},73^{rd},88^{th}	6	-	GND: Connect all DVSS pins to GND (0V).
REGOUT	49^{th}	1	out	Regulator output 3.3V: Connect capacitor to stabilize the regulator output.
REGEN	52^{nd}	1	in	Regulator enable pin: Should be set to H or OPEN (with pull-up register).

3. OPERATION

This section describes the basic components, functions and operation of TMP92CD54I.

3.1 CPU

TMP92CD54I contains an advanced high-speed 32-bit CPU (900/H1 CPU)

3.1.1 CPU Outline

900/H1 CPU is high-speed and high-performance CPU based on 900/H CPU. 900/H1 CPU has expanded 32-bit internal data bus to process Instructions more quickly.

Outline of 900/H1 CPU are as follows:

	900/H1 CPU	
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 some waits)	
Minimum Instruction Execution Cycle	1-clock(50ns@f_{OSC}=10MHz)	
Conditional Jump	2-clock(100ns@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 instruction is deleted)	
Micro DMA	8-channels	

3.1.2 Reset Operation

When resetting TMP92CD54I microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the RESET input Low for at least 20 system clocks (4us). At reset the clock doubler is bypassed and system clock operates at 5MHz (f_{OSC}=10MHz).

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

- Sets the Program Counter (PC) as follows in accordance with the Reset Vector stored at address FFFF00H 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 the Reset is released, the CPU starts executing instructions according to the Program Counter settings. CPU internal registers not mentioned above do not change when the Reset is released.

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

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

When external reset is released, built-in clock doubler begins operation and after the stable time (1.6384ms @ f_{OSC}=10MHz) elapse of the circuit, internal reset is released.

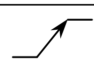
The operation of memory controller cannot be insured until power supply becomes stable after power-on reset. The external RAM data provided before turning on TMP92CD54I may be spoiled because the control signals are unstable until power supply becomes stable after power on reset.

TMP92CD54I can initialize all general-purpose ports and CLKOUT setting by reset, even if the device is not fed DVCC3 voltage to. When $\overline{\text{RESET}}$ = L level, CLKOUT will be initialized to High-z, but CLKOUT is pulled-up in internal logic. If the device is not fed DVCC3 voltage to, $\overline{\text{RESET}}$ = L level, CLKOUT will be High-z or pulled-up (H level output).

3.1.3 Setting of TEST0, TEST1, AM0 and AM1

Connect TEST0, TEST1 pin to "GND" to use at NORMAL mode.
Set AM0 pin to "0" and set AM1 pin to "1" to use.

Table 3.1.2 Operation Mode Setup Table

Operation Mode	Mode Setup input pin				
	RESET	AM1	AM0	TEST1	TEST0
Single-chip Mode		1	0	0	0

3.2 Memory Map

Figure 3.2 is a memory map of TMP92CD54I.

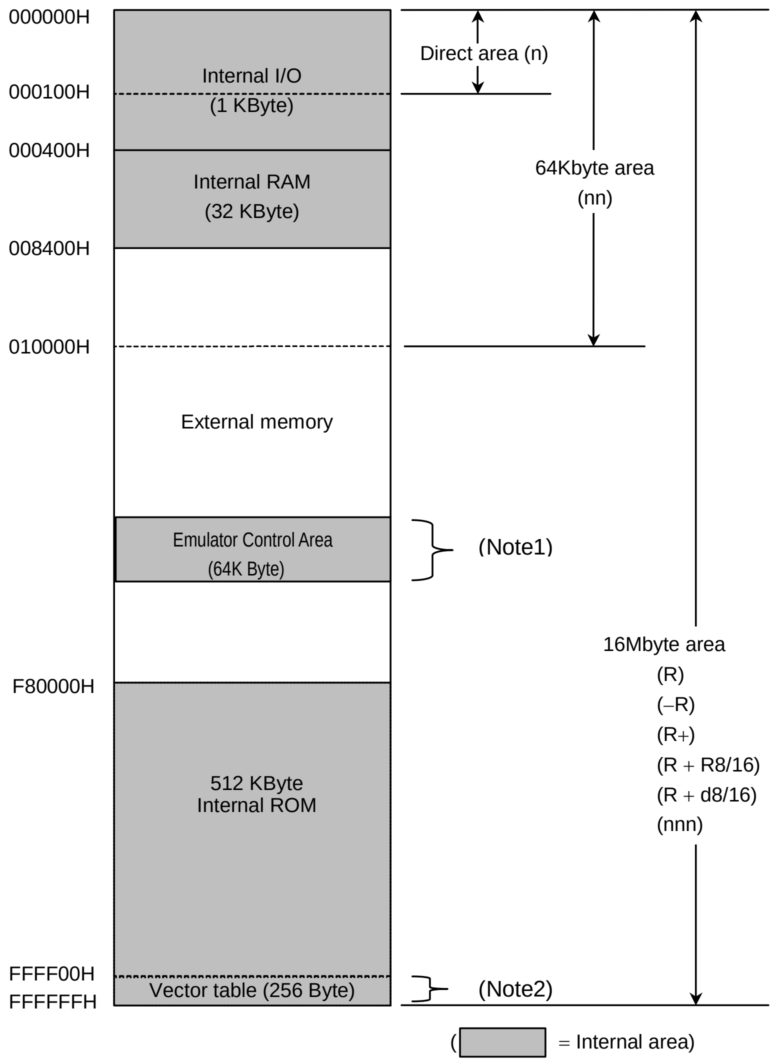
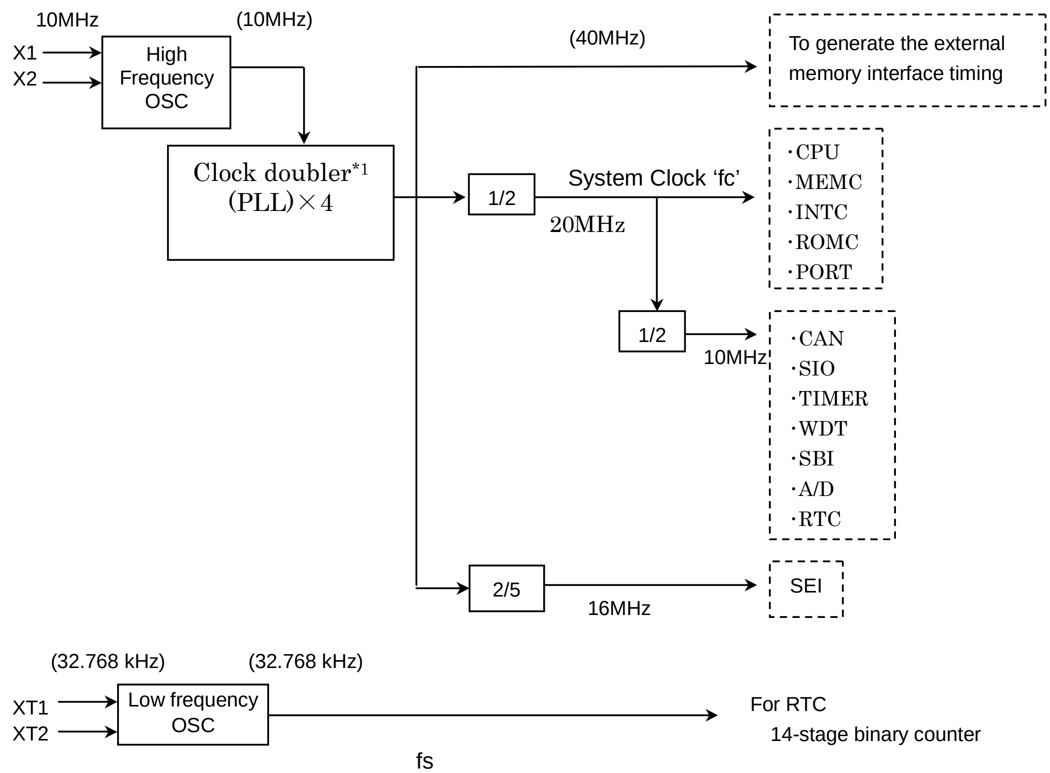


Figure 3.2 Memory Map

Note1: The emulator control area is for emulator, it is mapped F00000H to F10000H address after reset.
Note2: Don't use the last 16-byte area (FFFFF0H to FFFFFFH). This area is reserved.
Note3: On emulator WR signal and RD signal are asserted, when emulator control area is accessed. Be careful to use external memory.
Note4: Since there is a possibility of abnormal writing/reading of the data if Bus width put the different memories in consecutive address, do not execute an access which is placed on both memories with one command.

3.3 The Clock Function and Standby Function

3.3.1 Block diagram of system clock



*1) Clock-doubler outputs averaging 40MHz clock because it is corrected in clock unit of High Frequency OSC output (10MHz) though it has the possibility that the tolerance of 1.46ns at 40MHz (reference data) is included.

Figure 3.3.1 Block Diagram of System clock

3.3.2 Standby controller

(1) Halt Modes

When the HALT instruction is executed, the operating mode switches to Idle2, Idle1, Idle3 or Stop Mode, depending on the contents of the CLKMOD<HALTM1,HALTM0> register.

Clock Mode 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	Standby mode 00: IDLE3 01: STOP 10: IDLE1 11: IDLE2			Fix 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	Not output (Pull 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 subsequent actions performed in each mode are as follows:

① Idle2: The CPU only is halted.

In Idle2 Mode internal I/O operations can be performed by setting the following registers. Table 3.3.1 Shows the registers of setting operation during Idle2 Mode.

Table 3.3.1 Shows the registers of setting operation during Idle2 Mode

Internal I/O	SFR
TIMER0,TIMER1	TRUN01<I2T01>
TIMER2,TIMER3	TRUN23<I2T23>
TIMER4,TIMER5	TRUN45<I2T45>
TIMER6,TIMER7	TRUN67<I2T67>
TIMER8	TRUN8<I2T8>
TIMERA	TRUNA<I2TA>
SIO0	SC0MOD1<I2S0>
SIO1	SC1MOD1<I2S1>
SBI0	SBI0BR0<I2SBI0>
SBI1	SBI1BR0<I2SBI1>
SBI2	SBI2BR0<I2SBI2>
A/D converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>

② Idle1: Only the oscillator of low and high frequency continue to operate.

③ Idle3: Only the oscillator of low frequency and RTC are operated.

④ Stop: All internal circuits stop operating.

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

Table 3.3.2 I/O operation during Halt Modes

<table>
<tr><th colspan="2">Halt Mode</th><th>Idle2</th><th>Idle1</th><th>Idle3</th><th>Stop</th></tr>
<tr><td colspan="2">CLKMOD <HALT1:0></td><td>11</td><td>10</td><td>00</td><td>01</td></tr>
<tr><td rowspan="9">Block</td><td>CPU</td><td colspan="4">Halt</td></tr>
<tr><td>I/O ports</td><td colspan="2">Maintain same state as when HALT instruction was executed.</td><td colspan="2">See table 3.3.5</td></tr>
<tr><td>8-bit TMR, 16-bit TMR</td><td rowspan="4">Selectable
See table 3.3.1</td><td rowspan="4" colspan="3">Stopped</td></tr>
<tr><td>SIO, SBI</td></tr>
<tr><td>A/D converter</td></tr>
<tr><td>WDT</td></tr>
<tr><td>RTC, XT1</td><td rowspan="3">Operational</td><td colspan="2"></td><td rowspan="3"></td></tr>
<tr><td>CAN, SEI</td><td colspan="2" rowspan="2"></td></tr>
<tr><td>Interrupt controller</td></tr>
</table>

(2) How to clear a Halt mode

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

- Clearance by interrupt request

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

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

- Clearance by Reset

Any Halt state can be cleared by Reset.

When Stop Mode is cleared by RESET signal, sufficient time (at least10ms@f_{OSC}MHz) must be allowed after the Reset for the operation of the oscillator and clock doubler to stabilize.

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

Table 3.3.3 Source of Halt state clearance and Halt clearance operation

Status of Received Interrupt			Interrupt Enabled (interrupt level) ≥ (interrupt mask)				Interrupt Disabled (interrupt level) < (interrupt mask)			
Halt mode			Idle2	Idle1	Idle3	Stop	Idle2	Idle1	Idle3	Stop
Source of Halt state clearance	Interrupt	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	◎	◎	◎	◎	◎	◎	◎	◎

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

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

×: Cannot be used to clear the Halt mode.

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

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

*2: Any WUINT interrupt (WUINT0 to WUINT7) generate an INT0 interrupt.

Note 1: When the Halt mode is cleared by an INT0 interrupt of the level mode in the interrupt enabled status, hold level H until starting interrupt processing. If level L is set before holding level H, interrupt processing is not correctly started.

Note 2: When the external interrupts INT5 to INT7 are used during Idle2 Mode, set to 1 for TRUN8<I2T8> and TRUNA<I2TA>.

(Example - clearing Idle1 Mode)

An INT0 interrupt clears the Halt state when the device is 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 interrupt level to 5 for CPU.
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

① Idle2 Mode

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

Figure 3.3.3 illustrates an example of the timing for clearance of the Idle2 Mode Halt state by an interrupt.

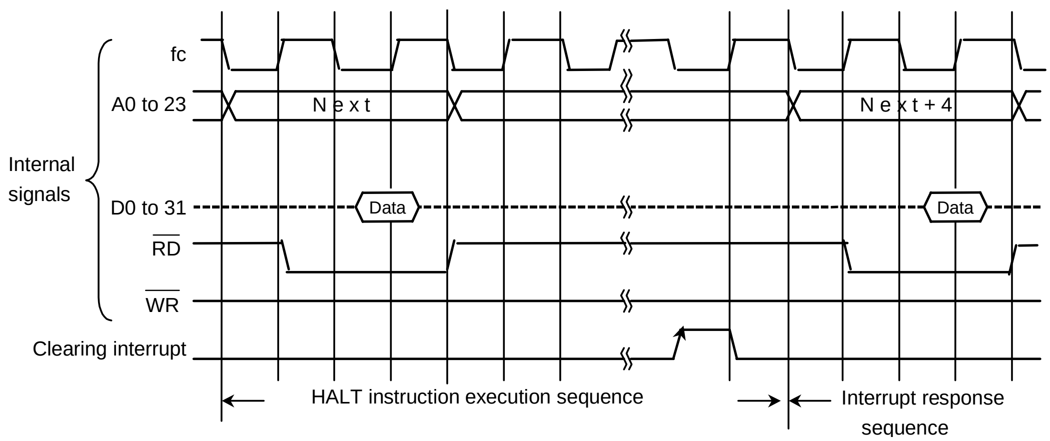


Figure 3.3.3 Timing chart for Idle2 Mode Halt state cleared by interrupt

② Idle1 Mode

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

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

Figure 3.3.4 illustrates the timing for clearance of the Idle1 Mode Halt state by an interrupt.

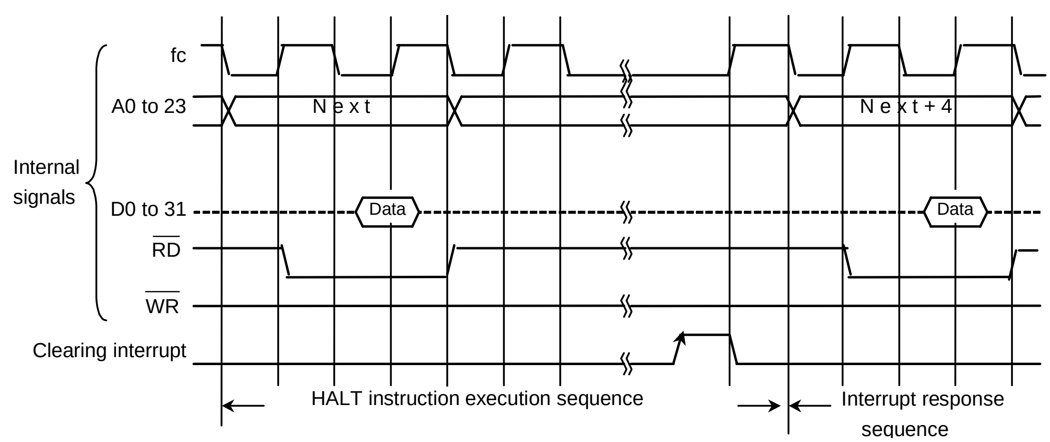


Figure 3.3.4 Timing chart for Idle1 Mode Halt state cleared by interrupt

③ Idle3 Mode

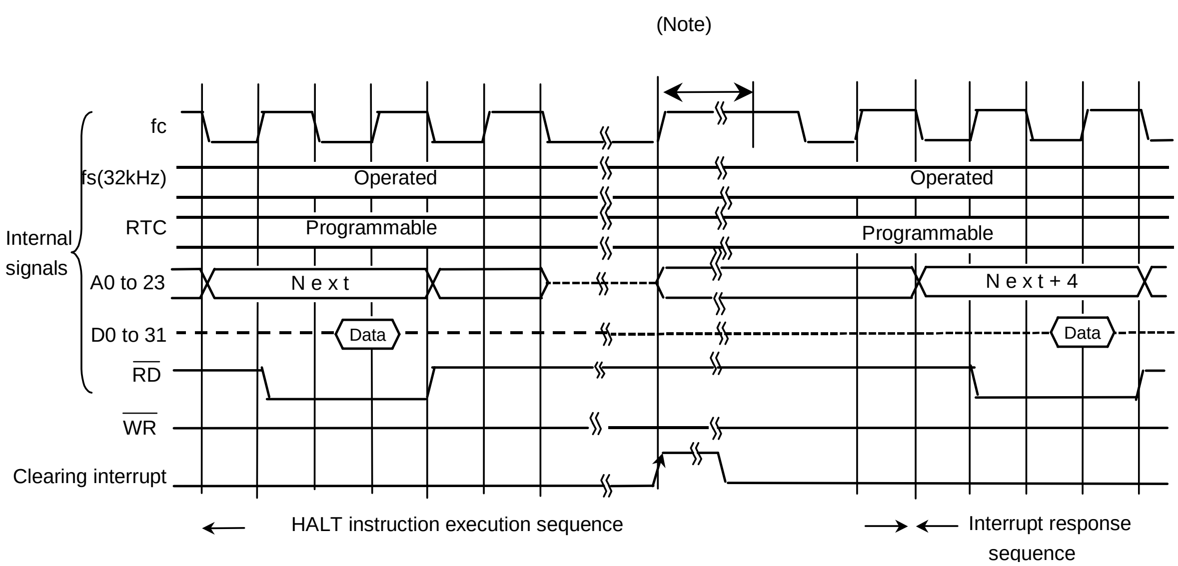
When Idle3 Mode is selected, internal circuits stop including the internal oscillator, except the oscillator of low frequency and RTC. Pin status in Stop Mode depends on the settings in the WDMOD<DRVE> register. Table 3.3.5 summarizes the state of these pins in Stop Mode and Idle3 mode.

After Idle3 Mode has been cleared system clock output starts when the warm-up time and clock doubler stable time have elapsed, in order to allow oscillation and clock doubler to stabilize. Figure 3.3.5 illustrates the timing for clearance of the Idle3 Mode Halt state by an interrupt.

Idle3 mode can only be released by an NMI pin, INT0 pin or WUINT0 to WUINT7 pins (generate a INT0 interrupt) interrupt, or by reset.

When Idle3 mode is released by other than reset, the system clock starts its output after the time set by the warm-up counter for the internal oscillation to stabilize. When using reset to release stop mode, input reset signals long enough for stable oscillation and clock doubler stable time.

In systems with an external oscillator, the warm-up counter also operates when Idle3 mode is released. Therefore, such systems also require a warm-up time between input of release signals and system clock output.



(Note); The interrupt processing starts after it completes for Startup time (T_{sta}) of Oscillator, Warm-up time and clock doubler stable time period, after releasing HALT (T_{sta} + 1.6 ms + 1.6 ms). Please inquire about Startup time (T_{sta}) to each oscillator manufacturer.

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

④ Stop Mode

When Stop Mode is selected, all internal circuits stop, including the internal oscillator. Pin status in Stop Mode depends on the settings in the WDMOD<DRVE> register. Table 3.3.5 summarizes the state of these pins in Stop Mode and Idle3 mode.

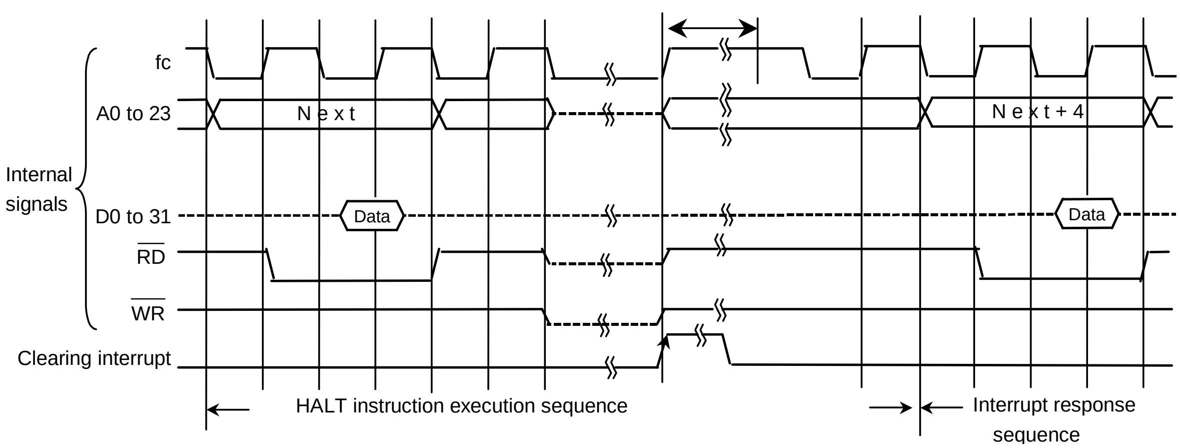
After Stop Mode has been cleared system clock output starts when the warm-up time and clock doubler stable time have elapsed, in order to allow oscillation and clock doubler to stabilize. Figure 3.3.6 illustrates the timing for clearance of the Stop Mode Halt state by an interrupt.

STOP mode can only be released by an NMI pin, INT0 pin or WUINT0 to WUINT7 pins(generate a INT0 interrupt) interrupt , or by reset.

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

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

And if it released from STOP mode, RTCFC register will be initialized without a RESET input. Therefore, it is necessary to set up RTCFC register again after releasing from STOP mode.



(Note); The interrupt processing starts after it completes for Startup time (T_{sta}) of Oscillator, Warm-up time and clock doubler stable time period, after releasing HALT (T_{sta} + 1.6 ms + 1.6 ms). Please inquire about Startup time (T_{sta}) to each 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/ $\overline{RD}$, $\overline{WR}$, $\overline{CS}$ to $\overline{WAIT}$	Input Mode	Invalid	
	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 / $\overline{SS}$ to SCK2	Input Mode	Invalid	
	Output Mode	High-z	Output
PN0 to PN6 /SCK0 to SO2&SDA2	Input Mode	Invalid	
	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: Input gate in operation. Input voltage should be fixed to “L” or “H” so that input pin stays constant.

Output: Output state

Invalid: Input pin invalid.

High-z: Output pin High-Impedance.

Note) At RTCFC<XTEN>=1.

3.4 Interrupts

Interrupts are controlled by the CPU Interrupt Mask Register <IFF2:0> (bits 12 to 14 of the Status Register) and by the built-in interrupt controller.

TMP92CD54I has a total of 60 interrupts divided into the following five types:

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)

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

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

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

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

However, software interrupts and Illegal Instruction interrupts generated by the CPU are processed irrespective of the value in <IFF2:0>.

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

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

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

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

In addition, TMP92CD54I also has a software start function in which micro DMA processing is requested in software rather than by an interrupt.

Figure 3.4.1 is a flowchart showing overall interrupt processing.

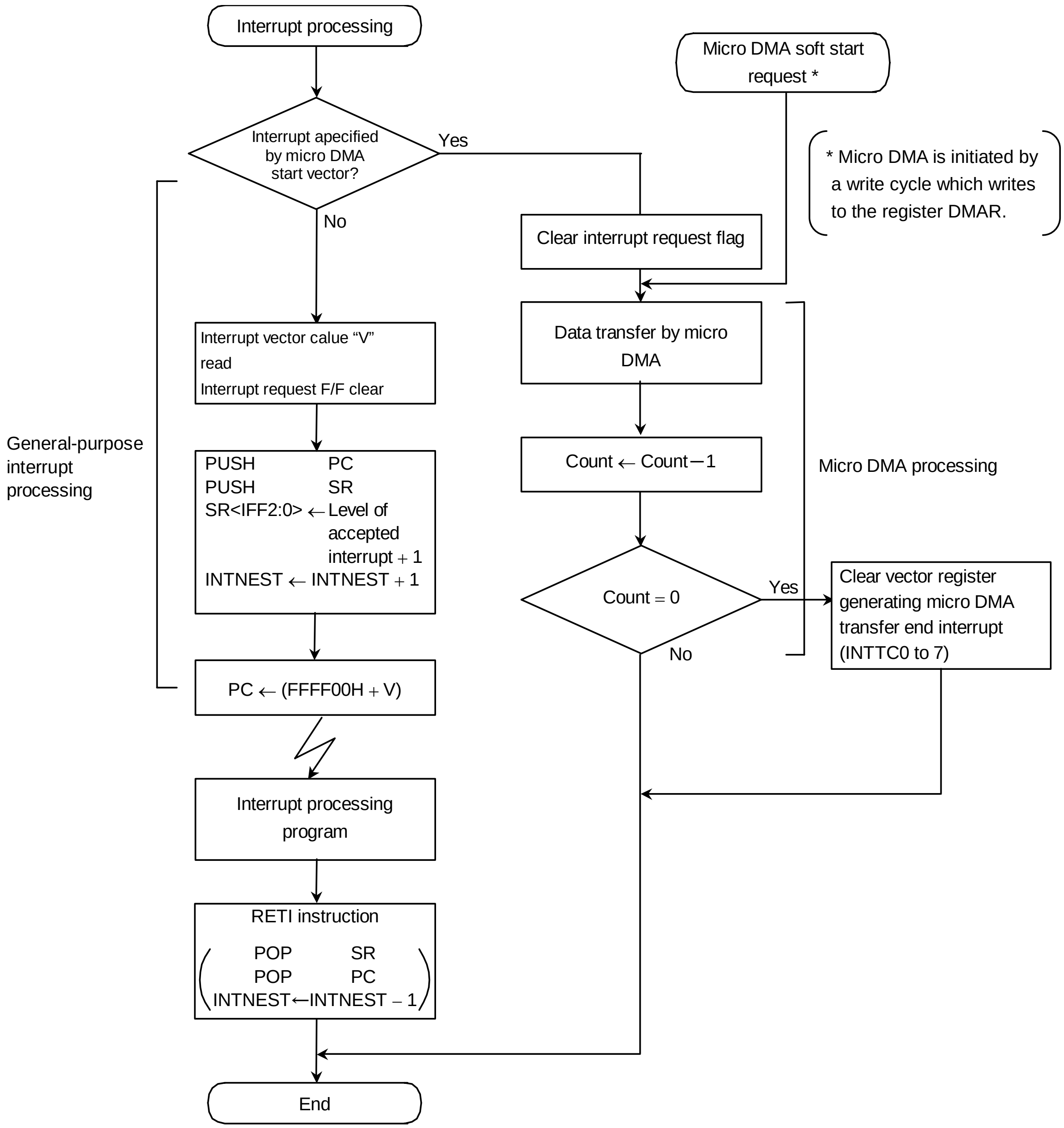


Figure 3.4.1 Interrupt and micro DMA processing sequence

3.4.1 General-purpose interrupt processing

When the CPU accepts an interrupt, it usually performs the following sequence of operations. However, in the case of software interrupts and Illegal Instruction interrupts generated by the CPU, the CPU skips steps (a) and (c) and executes only steps (b), (d) and (e).

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

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

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

(d) The CPU increments the interrupt nesting counter INTNEST by 1.

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

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

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

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

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

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

Table 3.4.1 shows TMP92CD54I interrupt vectors and micro DMA start vectors. FFFF00H to FFFFEFH (240 bytes) is designated as the interrupt vector area.

Table 3.4.1 TMP92CD54I interrupt vectors and micro DMA start vectors (1/2)

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

Table 3.4.2 TMP92CD54I interrupt vectors and micro DMA start vectors (2/2)

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 : 00FCH	FFFFF0H : FFFFFCH	- to -

Note1: Micro DMA default priority
If an interrupt request is generated by micro DMA, the interrupt has a higher priority than any other maskable interrupt (irrespective of default channel priority).

Note2: When standing-up micro DMA, set at edge detect mode.

Note3: Micro DMA processing cannot be applied.

Note4: This table mentions only the start address. Then each vector has 4 bytes.

3.4.2 Micro DMA processing

In addition to general-purpose interrupt processing, TMP92CD54I also includes a micro DMA function. Micro DMA processing for interrupt requests set by micro DMA is performed at the highest priority level for maskable interrupts (level 6), regardless of the priority level of the interrupt source.
Because the micro DMA function has been implemented with the cooperative operation of CPU, when CPU is a state of stand-by by HALT instruction, the requirement of micro DMA will be ignored (pending).

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

(1) Micro DMA operation

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

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

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

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

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

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

Three micro DMA transfer modes are supported: one-byte transfers, two-byte (one-word) transfer and four-byte transfer. After a transfer in any mode, the transfer source and transfer destination addresses will either be incremented or decremented, or will remain unchanged. This simplifies the transfer of data from I/O to memory, from memory to I/O, from I/O to I/O, and memory to memory. For details of the various transfer modes, see Section 3.4.2 (4), Detailed description of the Transfer Mode Register.

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

Micro DMA processing can be initiated by any one of 43 different interrupts – the 42 interrupts shown in the micro DMA start vectors in Table 3.4.1 and a micro DMA soft start. Figure 3.4.2 shows a 2-byte transfer carried out using a micro DMA cycle in Transfer Destination Address INC Mode (micro DMA transfers are the same in every mode except Counter Mode). (The conditions for this cycle are as follows: external 8-bit bus, 0 waits, and even-numbered transfer source and transfer destination addresses).

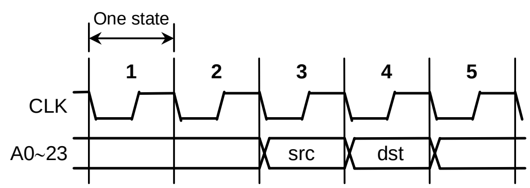


Figure 3.4.2 Timing for micro DMA cycle

States 1, 2:	Instruction fefetch cycle (pretches the next instruction code)
State 3:	Micro DMA read cycle
State 4:	Micro DMA write cycle
State 5:	(The same as in state 1, 2)

(2) Micro DMA operation

TMP92CD54I can initiate micro DMA either with an interrupt or by using the micro DMA soft start function, in which micro DMA is initiated by a Write cycle which writes to the register DMAR.

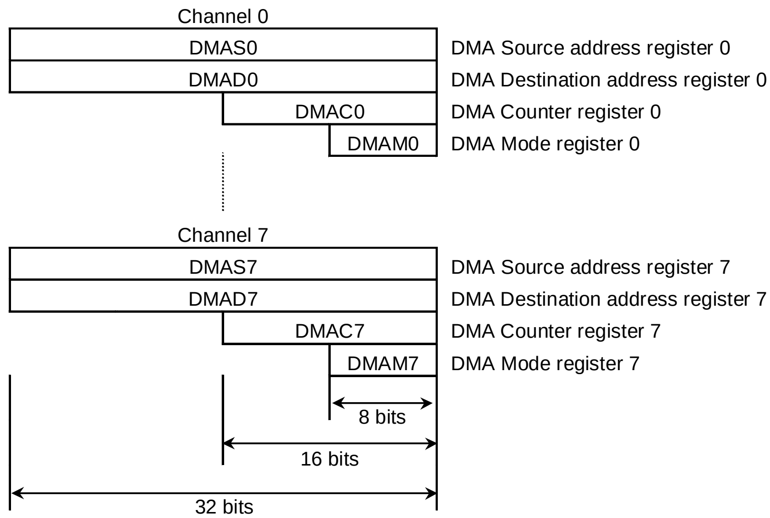
Writing 1 to any bit of the register DMAR causes micro DMA to be performed once. On completion of the transfer, the bits of DMAR which support the end channel are automatically cleared to 0.

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

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

(3) Transfer control registers

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



(4) Detailed description of the Transfer Mode Register

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

Note2: n stands for the micro DMA channel number (0 to 7)
DMADn+/DMASn+: Post-increment (register value is incremented after transfer)
DMADn–/DMASn–: Post-decrement (register value is decremented after transfer)

3.4.3 Interrupt controller operation

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

For each of the 51 interrupt channels there is an interrupt request flag (consisting of a flip-flop), an interrupt priority setting register and a micro DMA start vector register. The interrupt request flag latches interrupt requests from the peripherals. The flag is cleared to zero in the following cases: when a Reset occurs, when the CPU reads the channel vector of an interrupt it has received, when the CPU receives a micro DMA request (when micro DMA is set), when a micro DMA burst transfer is terminated, and when an instruction that clears the interrupt for that channel is executed (by writting a micro DMA start vector to the INTCLR register).

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

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

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

The interrupt controller also includes eight registers which are used to store the micro DMA start vector. Writing the start vector of the interrupt source for the micro DMA processing (see Table 3.4.1), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter registers (e.g. DMAS and DMAD) prior to micro DMA processing.

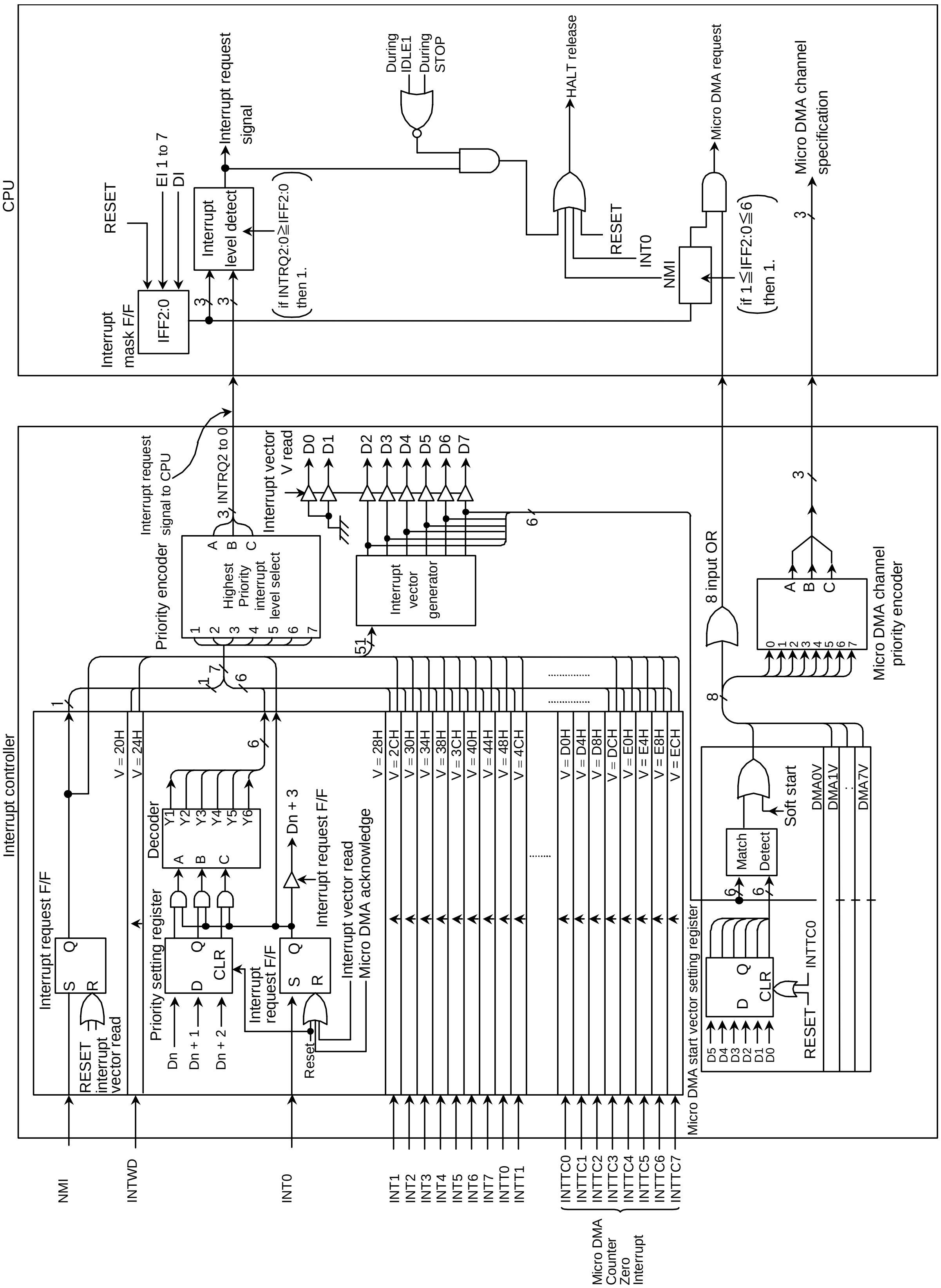


Figure 3.4.3 Block Diagram of Interrupt Controller

(1) Interrupt priority setting registers

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 & INTAD Enable	F0h	INTAD				INT0 (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: When any bit of WUPMASK<WMK7:0> is set to 1, INT0 will be disabled. Using INT0, set WUPMASK<WMK7:0> to “00H”.

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

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTNMWDT	NMI & INTWD Enable	F7h	NMI				INTWD			
			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: After executing DI command previously, the setting value of "Interrupt priority setting register" should change.

(2) External interrupt control

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

INT0 Level Enable

0	Rising edge detect INT
1	"H"level INT

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

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

Note 1 : Disable INT0 request before changing INT0 pin mode from level-sense to edge-sense.
Then, execute EI instruction after waiting 3-cycles (3 times NOP instruction).

Setting example:

```
DI                          ; Disable interrupts
LD      (IIMC), XXXXXX0-B   ; Switches from level to edge.
LD      (INTCLR), 0AH       ; Clears interrupt request flag.
NOP                         ; Wait 3-cycles
NOP
NOP
EI                          ; Enable interrupts
```

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

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

Table 3.4.2 Settings of External interrupt Pin Function

Interrupt	Pin name	Mode	Setting method
$\overline{NMI}$	$\overline{NMI}$	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 clear register

The interrupt request flag is cleared by writing the appropriate micro DMA start vector, as given in Table 3.4.1, to the register INTCLR.

For example, to clear the interrupt flag INT0, perform the following register operation after execution of the DI instruction.

INTCLR ← 0AH ; Clears interrupt request flag INT0.

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

(4) Micro DMA start vector registers

These registers assign an interrupt source which makes a micro DMA processing start. The interrupt source whose micro DMA start vector value matches the vector set in one of these registers is designated as the micro DMA start source.

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

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

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

Symbol	NAME	Address	7	6	5	4	3	2	1	0
DMA0V	DMA0 Start Vector	100h (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

(5) Specification of a micro DMA burst

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

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

(6) Notes

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

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

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

<table>
<tr><td>INT0 Level Mode</td><td>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</td></tr>
<tr><td>INTRX</td><td>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.</td></tr>
</table>

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

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

INTRX: Instructions which read the Receive Buffer

3.4.4 Interrupt Mask register

TMP92CD54I has Interrupt Mask registers. Unlike Interrupt priority register, Interrupt mask register only disables or enables interrupts. An interrupt will not be generated, if the interrupt is disabled by Interrupt mask register, even if the interrupt has been enabled by setting Interrupt priority register. One, two or more interrupt factors can be prohibited synchronous by setting of Interrupt Mask register.

After reset, all bits in Interrupt mask register are initialize 1 (enabled interrupts). It is necessary to write 0 in the corresponding bit in case of making interrupt Mask register to prohibit interrupt.

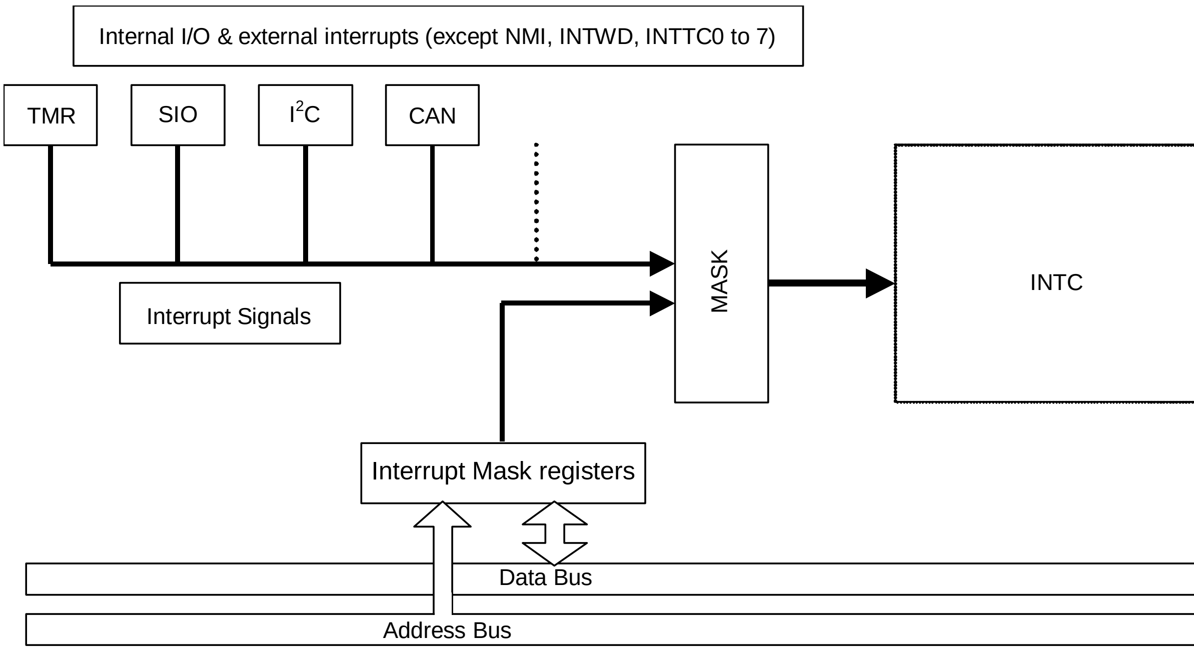


Figure 3.4.4 Block Diagram of Interrupt Mask Control

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

Symbol	NAME	Address	7	6	5	4	3	2	1	0
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: Port D0, D1 and D4 have 2 kinds of interrupt source (PD0:INT5/WUINT0, PD1:INT6/WUINT1, PD4:INT7/WUINT4). If both interrupt requests are generated in both interrupt enabled status, both interrupt processing will be executed. When any of these interrupts is used, set Interrupt Mask register or Wake UP Mask register to enable/disable.

Example of register setting:

In the case of setting INT0 interrupt priority level to 7 from 3.

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

3.4.5. ON/OFF LOGIC

TMP92CD54I has 8 pins (WUINT0 to WUINT7) for wake up from standby mode. These pins are multiplexed with Port D (PD0 to PD7).

All wake up events can release standby mode and triggering edge can be independently programmable as both rising and falling edge, rising edge or falling edge. It is possible to mask all wake up events independently.

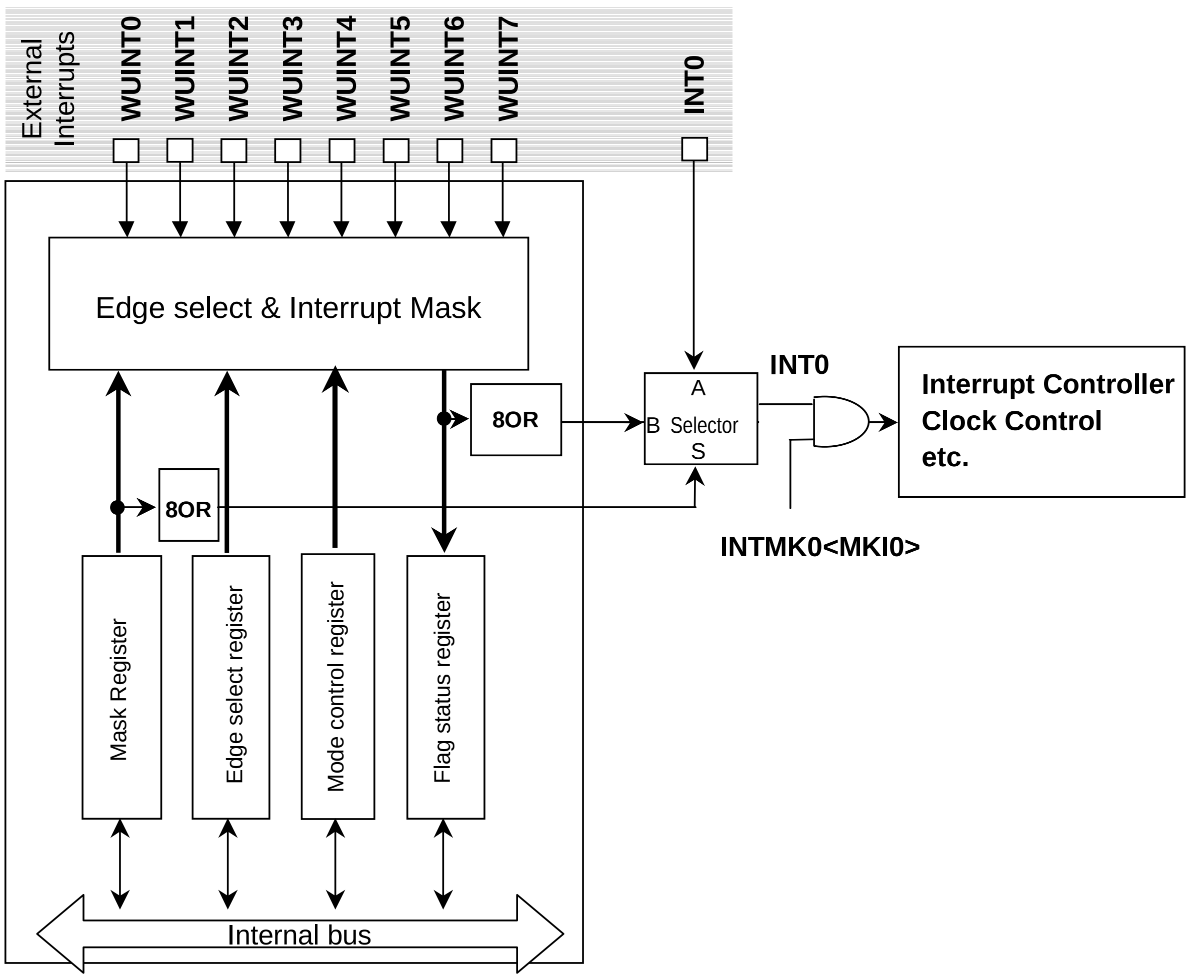


Figure 3.4.5 Block diagram of ON/OFF logic

Using ON/OFF logic, all interrupt signals of WUINT0 to 7 are sent to INT0 in internal logic. When any WUINTn requests are generated, INT0 interrupt request will be generated. Like external INT0, also INT0 from WUINTn is set disable/enable by Interrupt priority register or Interrupt mask register.

Writing 1 to any bit in WUPMASK register, INT0 switches ON/OFF logic mode. In this case, WUINTn written 1 in WUPMASK register are enabled, external INT0 cannot use. When external INT0 is used, write 00 to WUPMASK register.

Selection edge of WUINTn signal uses WUPMOD and WUPEDGE register, rising edge, falling edge or both falling and rising edge are selectable.

Reading WUPFLAG register, request/no-request of WUINTn will be confirmed.

Wake UP 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

Wake UP 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

Wake UP 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: WUPEDGE register is used with setting each WUPMOD<WMD7:0> to 1.
If each WUPMOD<WMD7:0> is clear to 0, WUPEDGE<WED7:0> is disabled.

Wake UP 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

→ Wake up interrupt mask control

0	WUINTn Disabled (MASK)
1	WUINTn Enabled

Note1: Port D0, D1 and D4 have 2 kinds of interrupt source (PD0: INT5/WUINT0, PD1: INT6/WUINT1, PD4:INT7/WUINT4). If both interrupt requests are generated in both interrupt enabled status, both interrupt processing will be executed. When each interrupts is used, set Interrupt Mask register or Wake UP Mask register to enable/disable. Even if port D is any of Input/Output port, INTn, and WUINTn, the level of port D is inputted into these interrupts. For details, refer to the block diagram of the port.

Note2: When any WUPMASK<WMK7:0> is set to 1, external INT0 will be disabled. Using INT0, set WUPMASK<WMK7:0> to “00H”.

Example of register setting:

To set WUINT0 with rising edge and set interrupt level 3, set the registers as follows:

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

3.5 Function of Ports

TMP92CD54I has I/O port pins that are shown in table 3.5.1. In addition to functioning as general-purpose I/O ports, these pins are also used by internal CPU and I/O functions.

Table 3.5.1 Port Functions (1/2)

Port Name	Pin Name	Number of Pins	I/O	I/O Setting	Pin Name for built-in function
Port 0	P00 to P07	8	I/O	Bit	D0 to D7
Port 4	P40 to P47	8	I/O	Bit	A0 to A7
Port 7	P70	1	I/O	Bit	$\overline{RD}$
	P71	1	I/O	Bit	$\overline{WR}$
	P72	1	I/O	Bit	SI2/SCL2
	P73	1	I/O	Bit	$\overline{CS}$
	P74	1	I/O	Bit	
	P75	1	I/O	Bit	$\overline{WAIT}$
Port C	PC0	1	I/O	Bit	TI0 / INT1
	PC1	1	I/O	Bit	TO1
	PC2	1	I/O	Bit	TO3 / INT2
	PC3	1	I/O	Bit	TI4 / INT3
	PC4	1	I/O	Bit	TO5
	PC5	1	I/O	Bit	TO7 / INT4
Port D	PD0	1	I/O	Bit	TI8 / INT5 / 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 / $\overline{CTS0}$
	PF3	1	I/O	Bit	TXD1
	PF4	1	I/O	Bit	RXD1
	PF5	1	I/O	Bit	SCLK1 / $\overline{CTS1}$
	PF6	1	I/O	Bit	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	$\overline{SS}$ / 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 to P07)

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

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

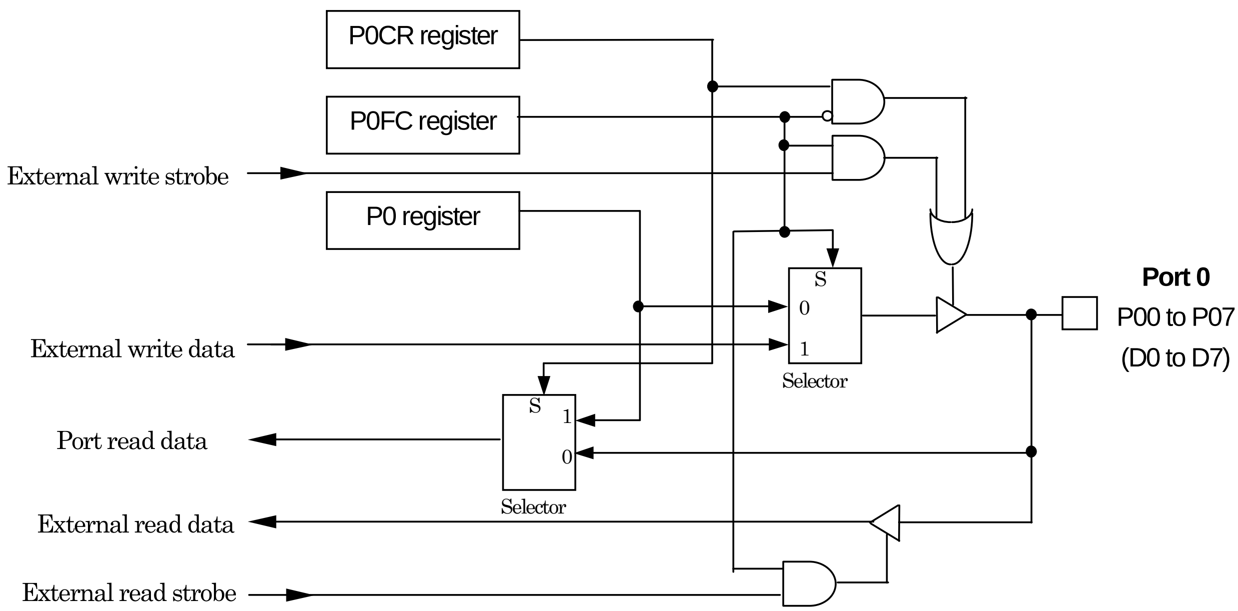


Figure 3.5.1 Port0

Table 3.5.2 Port0 Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
P0	PORT0	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P0CR	PORT0 Control Register	02H (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 to P47)

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

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

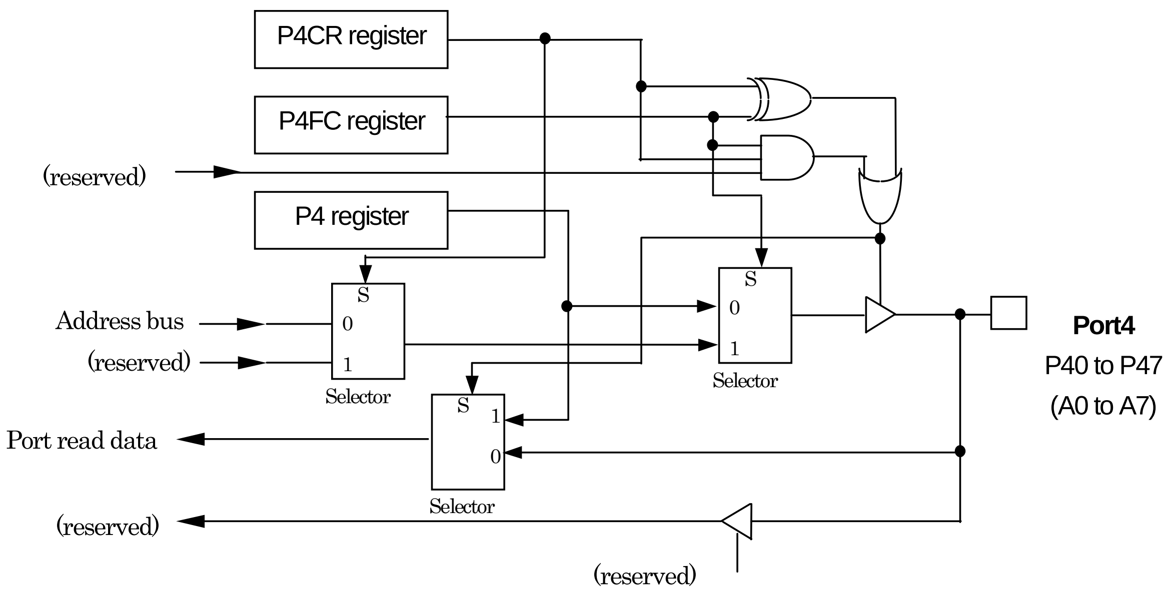


Figure 3.5.2 Port4

Table 3.5.3 Port4 Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
P4	PORT4	10H	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P4CR	PORT4 Control Register	12H (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)							

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

3.5.3 Port 7 (P70 to P75)

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

In addition to functioning as a general-purpose I/O port, P70, P71 and P73 pins can also function as read/write strobe signals and chip selection to connect with an external memory. P72 pin can also function as I/O functions of serial bus interface which employs clocked-synchronous 8-bit SIO and I²C. P75 pin can also function as wait input.

The pin is always enabled for the following input signals: SBI data input (SIO) SI2#1, SBI clock I/O (I²C) SCL2#1.

#1 : In IDLE3/STOP mode, input signal is valid (Input gate opened)

A reset initializes P70, P71, P73 and P74 pins to output port mode, and P72, P75 pin to input port mode.

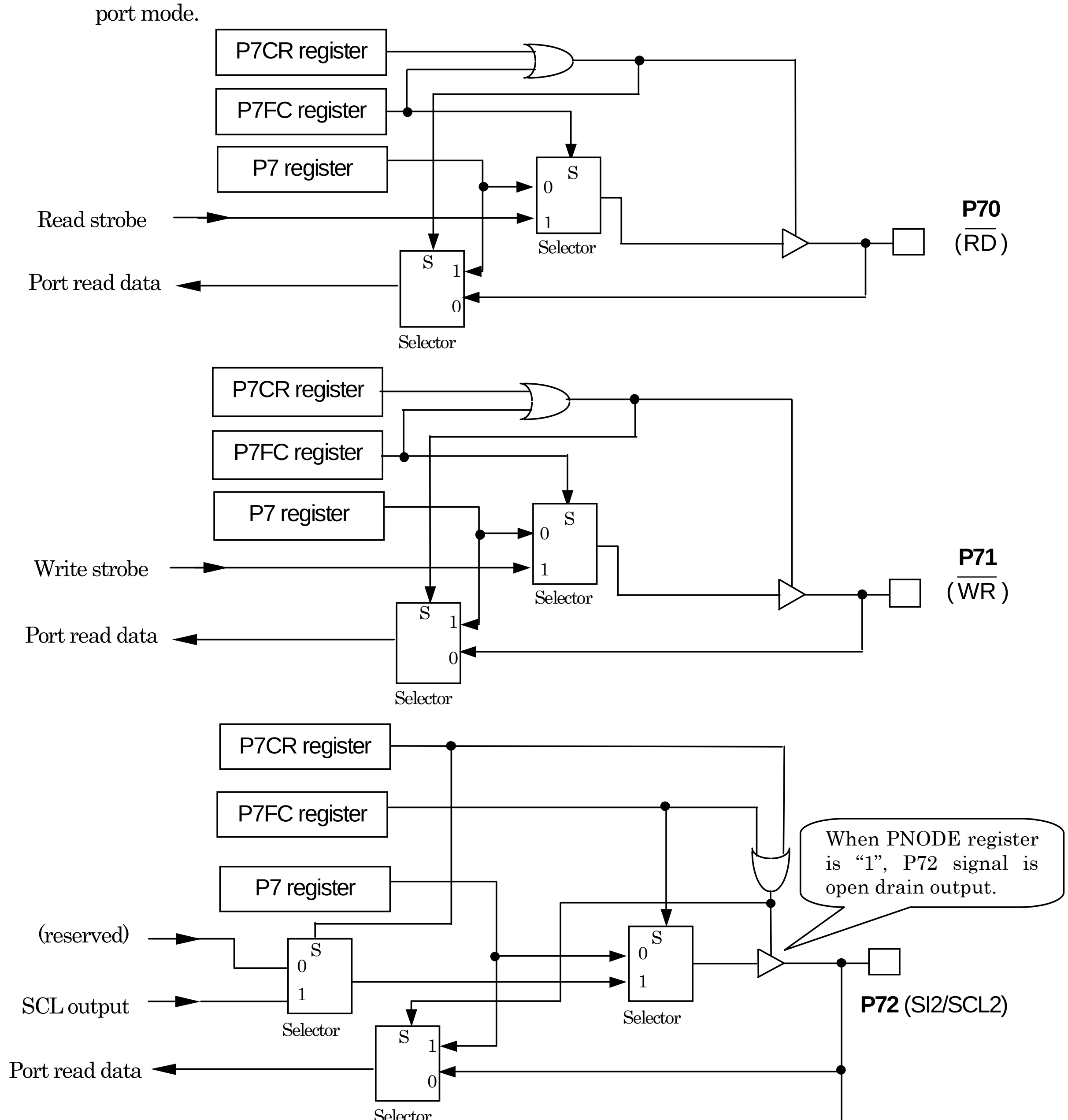


Figure 3.5.3 Port7 (P70 to P72)

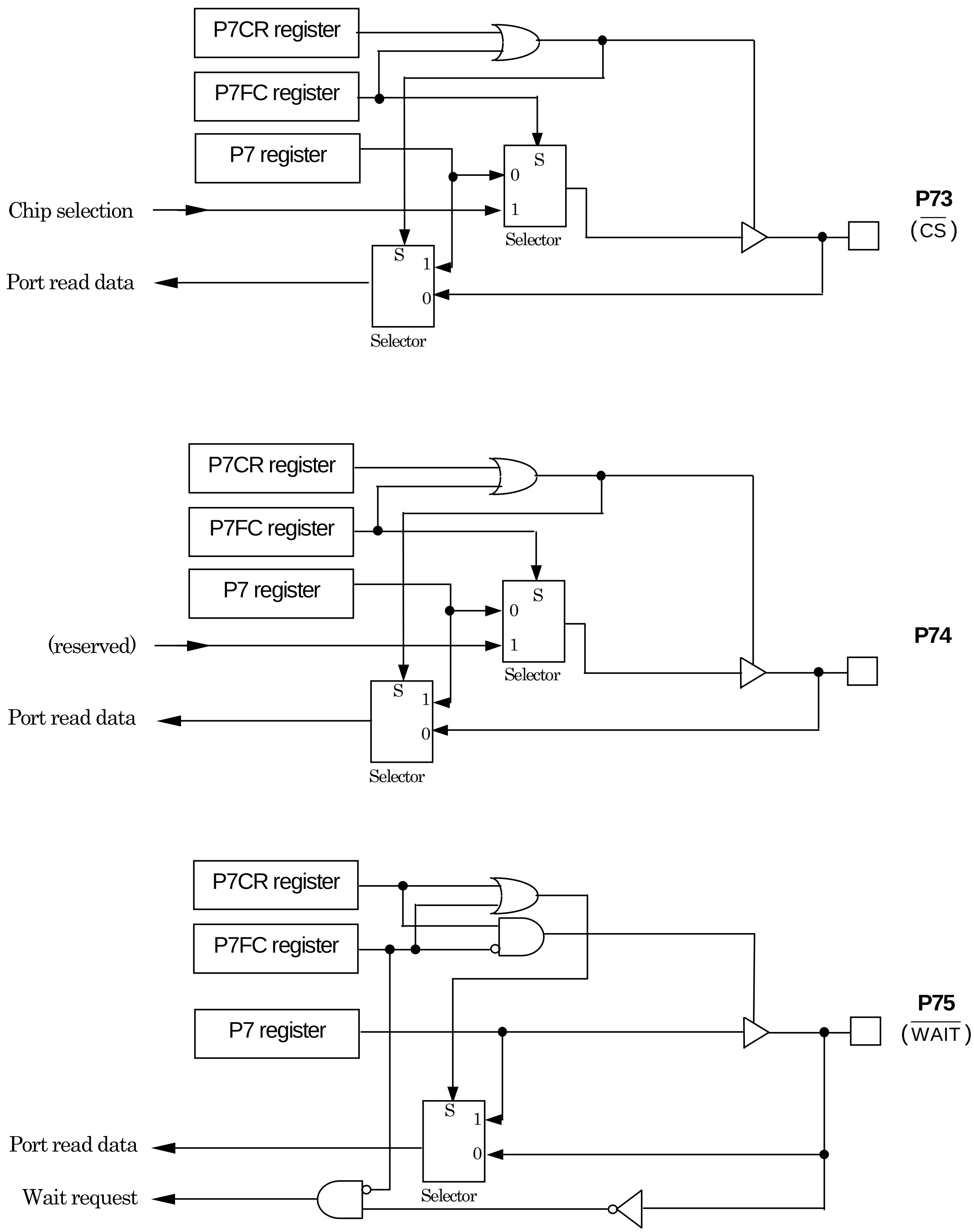


Figure 3.5.4 Port7 (P73 to P75)

Table 3.5.4 Port7 Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
P7	PORT7	1CH	-	-	P75	P74	P73	P72	P71	P70
					R/W					
			-	-	0	1	1	1	1	1
					Input/Output					
P7CR	PORT7 Control Register	1EH (no RMW)	-	-	P75C	P74C	P73C	P72C	P71C	P70C
					W					
			-	-	0	1	1	0	1	1
					0:Input 1:Output					
P7FC	PORT7 Function Register	1FH (no RMW)	-	-	P75F	P74F	P73F	P72F	P71F	P70F
					W					
			-	-	0	0	0	0	0	0
					0:PORT 1: $\overline{WAIT}$	0:PORT	0:PORT 1: $\overline{CS}$	0:PORT 1:SI2 SCL2 Note1	0:PORT 1: $\overline{WR}$	0:PORT 1: $\overline{RD}$

P7CR	P7FC	-	-	P75	P74	P73	P72	P71	P70
0	0			Input Port			Input Port, SI2	Input Port	
1	0			Output Port					
1	1			$\overline{WAIT}$	Don't use this setting.	$\overline{CS}$	Don't use this setting.	$\overline{WR}$	$\overline{RD}$
0	1			$\overline{WAIT}$	Don't use this setting.	$\overline{CS}$	SI2, SCL2	$\overline{WR}$	$\overline{RD}$

Note1: P72 SCL2, clock input/output at I2C mode, can be open-drain output by setting 1 to PNODE<ODE72>.

3.5.4 Port C (PC0 to PC5)

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

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

The pin is always enabled for the following input signals: timer inputs TI0#1, TI4#1.

#1 : In IDLE3/STOP mode, input signal is invalid (Input gate closed)

A reset initializes PortC to input port mode.

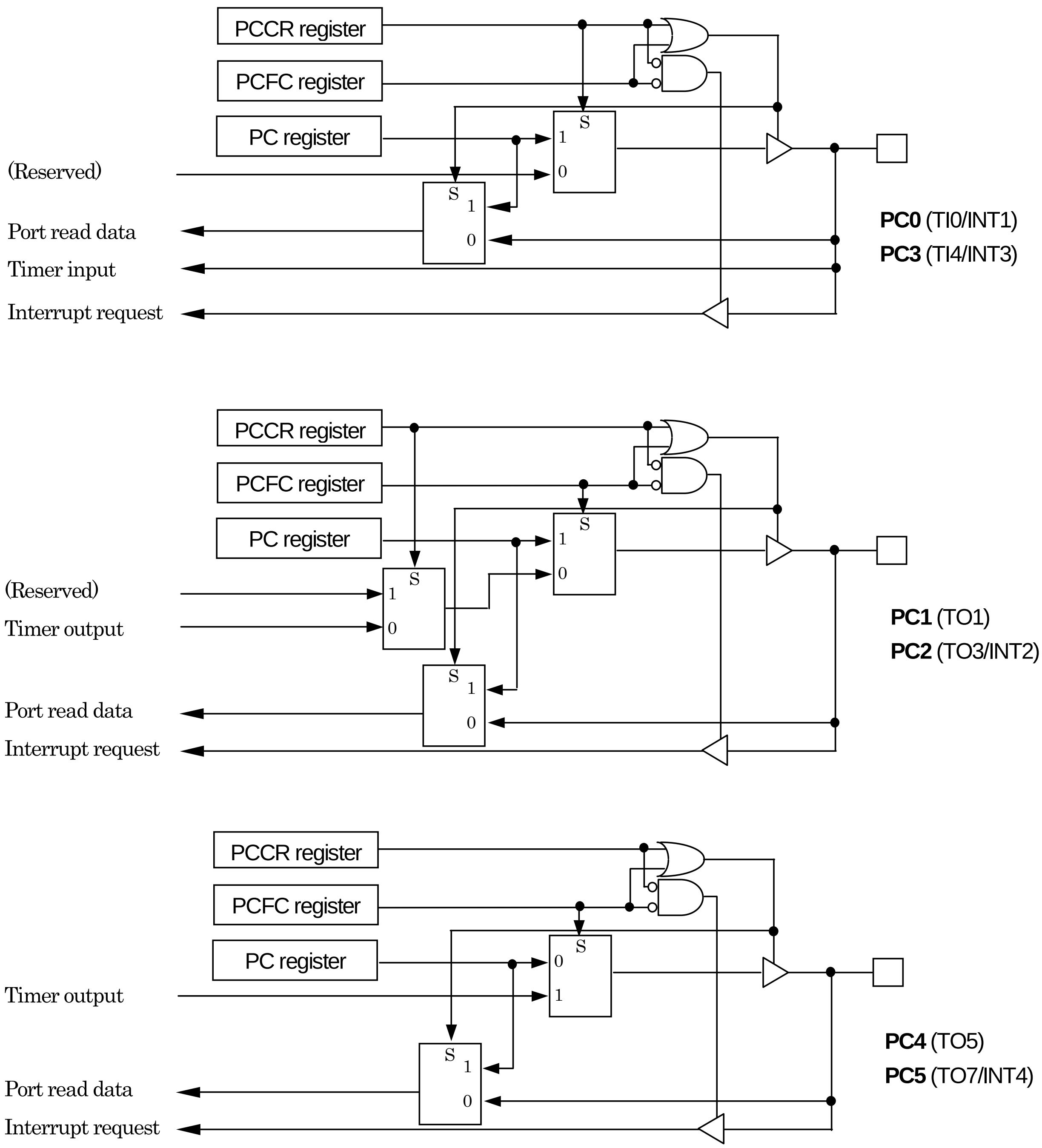


Figure 3.5.5 PortC (PC0 to PC5)

Table 3.5.5 PortC Registers

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
PC	PORTC	30H	-	-	PC5	PC4	PC3	PC2	PC1	PC0
					R/W					
			-	-	0	0	0	0	0	0
					Input/Output					
PCCR	PORTC Control Register	32H (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			

3.5.5 Port D (PD0 to PD7)

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

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

The pin is always enabled (excluding address bus setting) for the following input signals: 16-bit timer input TI8#1, TI9#1, TIA#1, TIB#1, external interrupt INT5#2 to INT7#2, wake up interrupt WUINT0#2 to WUINT7#2.

#1 : In IDLE3/STOP mode, input signal is invalid (Input gate closed)

#2 : In IDLE3/STOP mode, input signal is valid (Input gate opened)

A reset initializes Port D to input port mode.

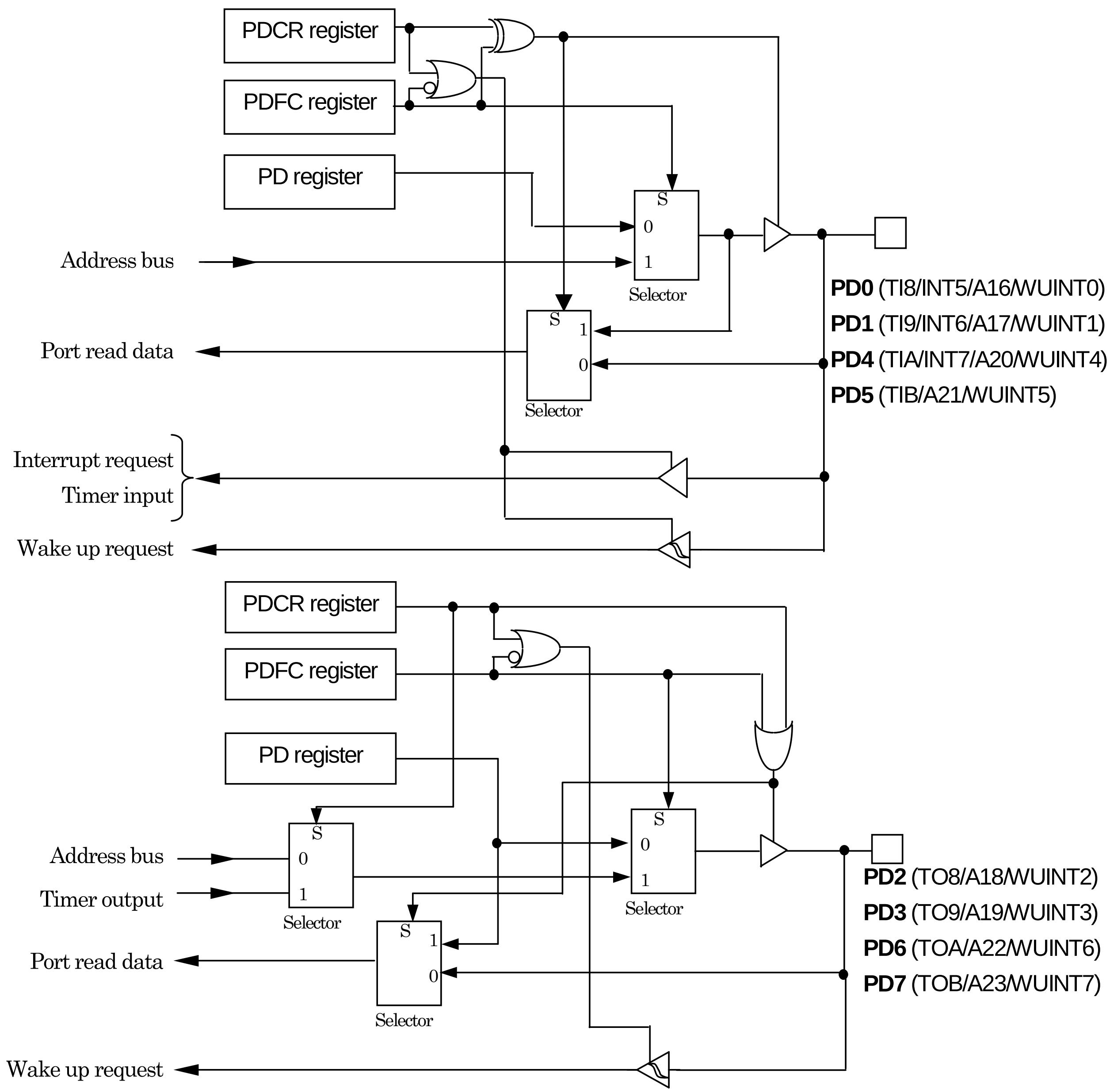


Figure 3.5.6 PortD

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: Port D0, D1 and D4 have 2 kinds of interrupt source (PD0: INT5/WUINT0, PD1: INT6/WUINT1, PD4: INT7/WUINT4).
If both interrupt requests are generated in both interrupt enabled status, both interrupt processing are executed.
When each interrupts is used, set Interrupt Mask register or Wake UP Mask register to enable/disable.
If these ports are used as output/input ports, first, disable interrupt request, then set PDFC and PDCR (cf Timers).

3.5.6 Port F (PF0 to PF7)

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

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

The pin is always enabled for the following input signals: serial receive data RXD0#1, RXD1#1, CAN receive data RX#1, Clear-to-send CTS0#1, CTS1#1, and serial clock SCLK0#1, SCLK1#1.

#1 : In IDLE3/STOP mode, input signal is invalid (Input gate closed)

A reset initializes PortF to input port mode.

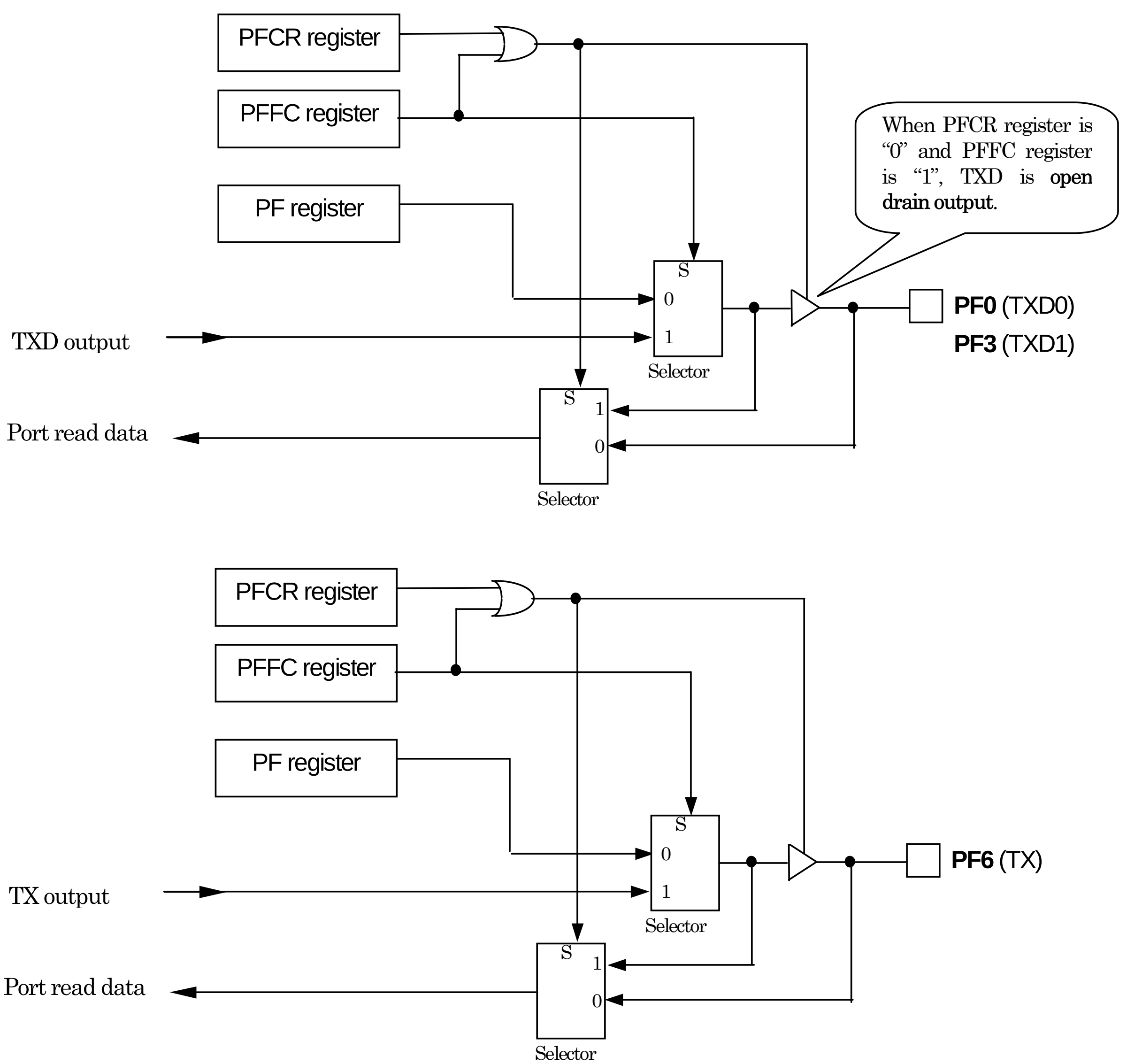


Figure 3.5.7 PortF (PF0, PF3, PF6)

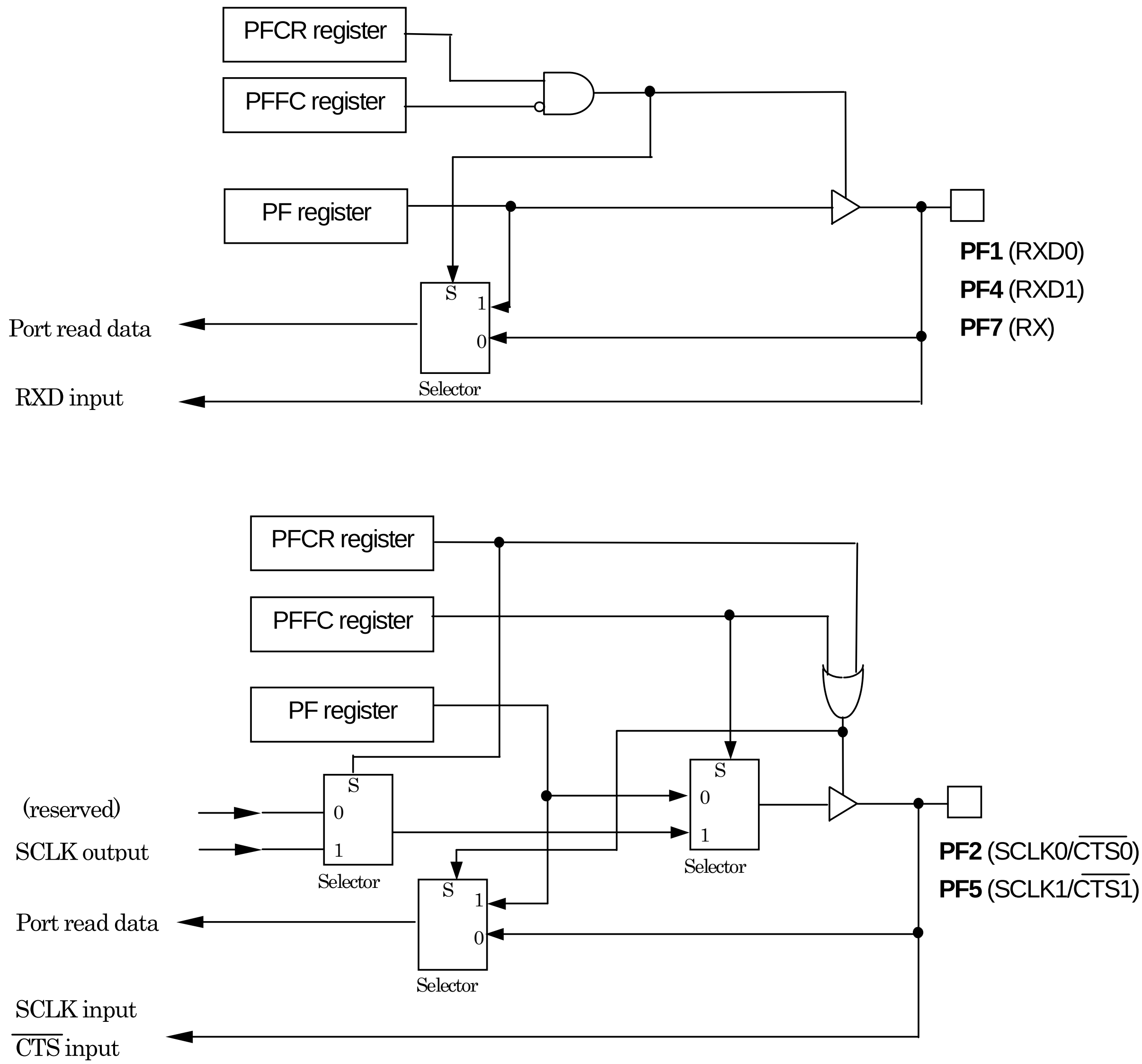


Figure 3.5.8 PortF (PF1, PF2, PF4, PF5, PF7)

Table 3.5.9 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 $\overline{\text{CTS1}}$ 1:SCLK1	0:PORT 1:RXD1	0:PORT 1:TXD1	0:PORT $\overline{\text{CTS0}}$ 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), CTS1	Input Port, RXD1	Input Port	Input Port, SCLK0 (Input), 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 to PG7)

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

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

The pin is always enabled for the following input signals: AD converter input AN0#1 to AN7#1.

#1 : In IDLE3/STOP mode, input signal is invalid (Input gate closed)

Figure 3.5.9 PortG

Table 3.5.8 PortG Register

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

3.5.8 Port L (PL0 to PL3)

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

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

The pin is always enabled for the following input signals: AD converter input AN8#1 to AN11#1.

#1 : In IDLE3/STOP mode, input signal is invalid (Input gate closed)

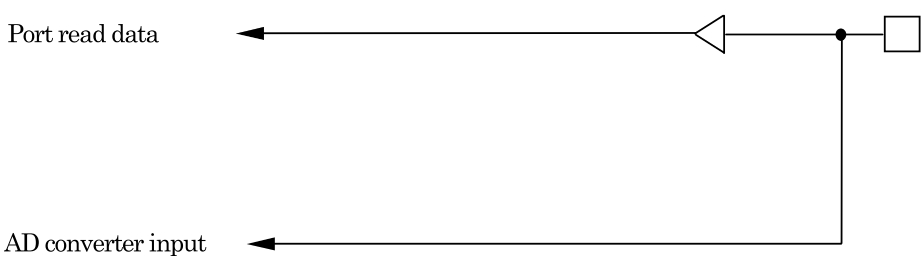


Figure 3.5.10 PortL

Table 3.5.9 PortL Register

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

3.5.9 Port M (PM0 to PM4)

PortM is a 5-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PMCR and function register PMFC.

In addition to functioning as a general-purpose I/O port, PM0 to PM3 can also function as I/O functions of serial expansion interface. PM4 can also function as I/O function of serial bus interface which employs clocked-synchronous 8-bit SIO.

The pin is always enabled for the following input signals: slave select $\overline{\text{SS}}$ #1, transmitting/receiving serial data MOSI#1, MISO#1, SEI clock SECLK#1.

#1 : In IDLE3/STOP mode, input signal is invalid (Input gate closed)

A reset initializes PortM to input port mode.

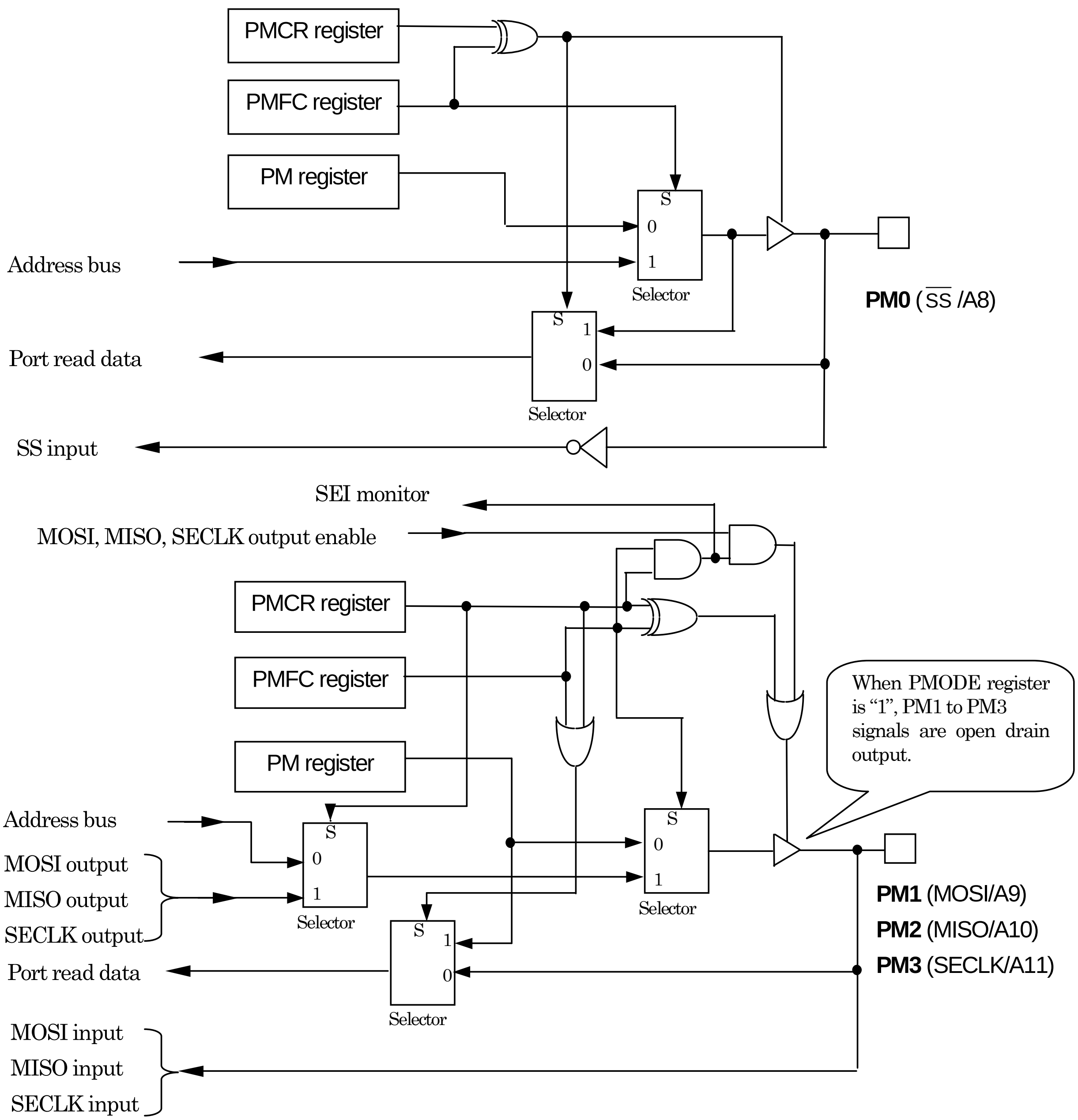


Figure 3.5.11 PortM (PM0 to PM3)

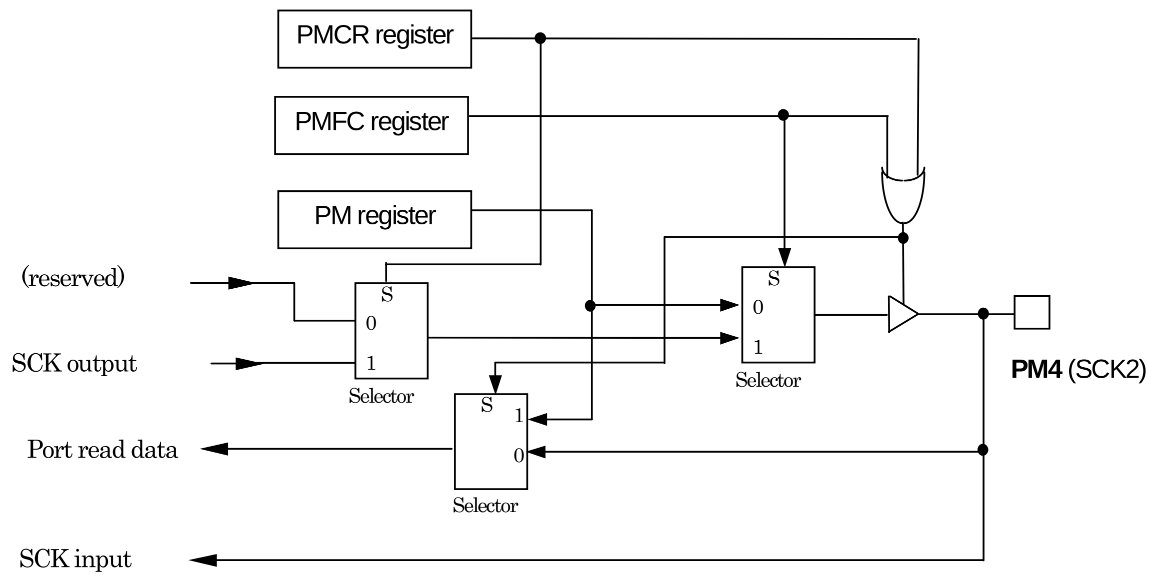


Figure 3.5.12 PortM (PM4)

Table 3.5.10 PortM Registers

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: $\overline{SS}$ 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

3.5.10 Port N (PN0 to PN6)

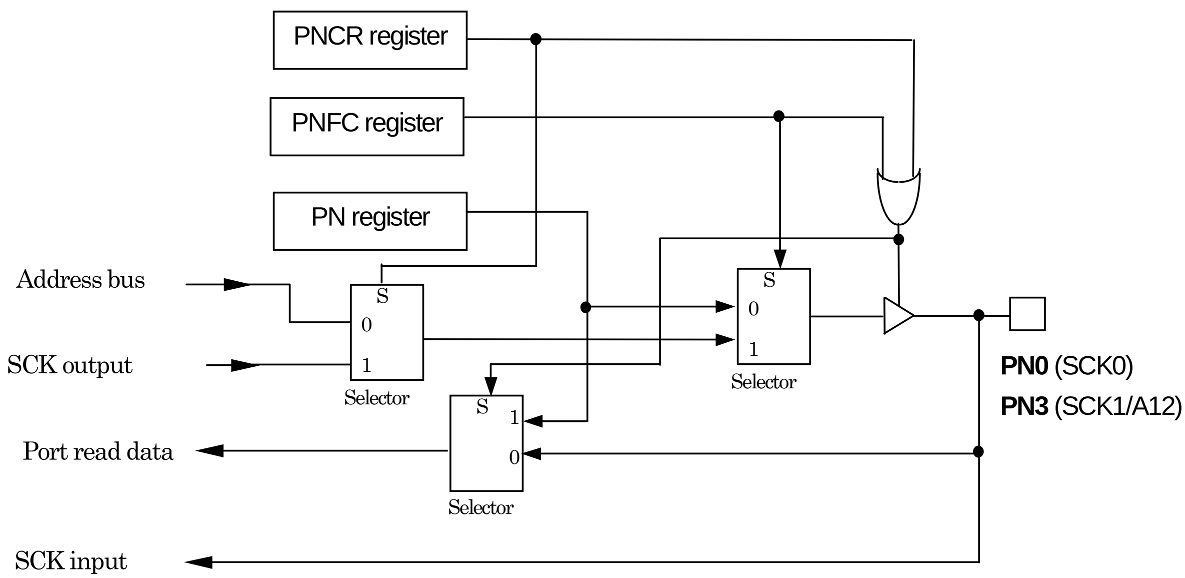
PortN is a 7-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PNCR and function register PNFC.

In addition to functioning as a general-purpose I/O port, PortN can also function as I/O functions of serial bus interface which employs clocked-synchronous 8-bit SIO and I²C.

The pin is always enabled for the following input signals: SBI clock I/O (SIO) SCK0#1, SCK1#1, SBI data input (SIO) SI0#1, SI1#1, SBI clock I/O (I²C) SCL0#1, SCL1#1, SBI data I/O (I²C) SDA0#1, SDA1#1, SDA2#1.

#1 : In IDLE3/STOP mode, input signal is invalid (Input gate closed)

A reset initializes PortN to input port mode.



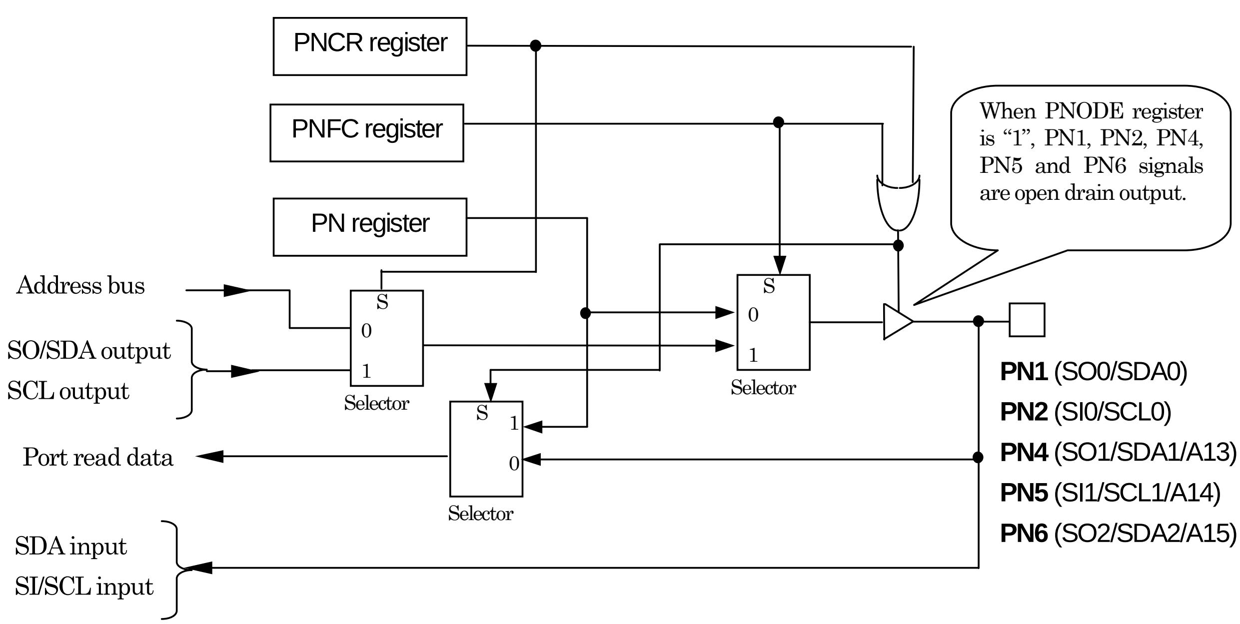


Figure 3.5.13 PortN

Table 3.5.11 PortN Registers

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, 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.		

3.6 Memory Controller

3.6.1 Memory controller functions

TMP92CD54I has a memory controller with a variable 1-block external address area. The function is as follows.

(1) 1-block external address area support. It specifies:

- A start address
- A block size for 1-block external address area

(2) Connecting memory specifications. It specifies:

- SRAM
- ROM

as memories to connect with the selected address area.

(3) Data bus size

8-bit

(4) Wait control

- Wait specification
- Wait input pin

Both control the number of waits in the external access bus cycle. Read and write cycles can specify the number of waits individually.

There are five modes all together:

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

3.6.2 Control register and Operation after reset release

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

(1) Control Registers

- Control registers (BCSH/BCSL: Block Chip Select High / Low)
  - Sets the connecting memory type. (SRAM, ROM)
  - Sets the number of waits to be read and written.
- Memory Start Address Register (MSAR)
  - Sets a start address in the selected address areas.
- Memory Address Mask Register (MAMR)
  - Sets a block size in the selected address areas.

(2) Operation after reset release

After reset release,

- The block address areas (specified by MSAR and MAMR) are set to address 000000H and FFFFEFH.
- Then BCSL / H is set.
- Set BCSH<BE> to 1 to enable the setting.

3.6.3 Basic functions and registers setting

In this section, Block address area specification, wait control and basic bus sizing are described.

(1) Block address area specification

If the bit BCSH<BM> is set to 0, then the block address area is set to addresses 000000H to FFFFEFH, which disables the use of both registers MSAR and MAMR.

If the bit BCSH<BM> is set to 1, then the block address area is programmable. Therefore, the start address is set using MSAR (Memory Start Address register). MAMR (Memory Address Mask Register) sets the size of the block in the selected address area. The principle is to mask or enable the comparison of each bit of the address. The combination of masked / enabled bits give the block size.

Then the memory controller compares (every bus cycle) the register value and the address in order to check whether it is an access to the external memory or not. Note that an address bit masked by MAMR (Memory Address Mask Register) is not compared. If the compared result is a match the memory controller sets the chip select signal $\overline{\text{CS}}$ to low.

Figure 3.6.1 shows an example of connecting external memory to TMP92CD54I. In the example, RAM is connected using an 8-bit bus.

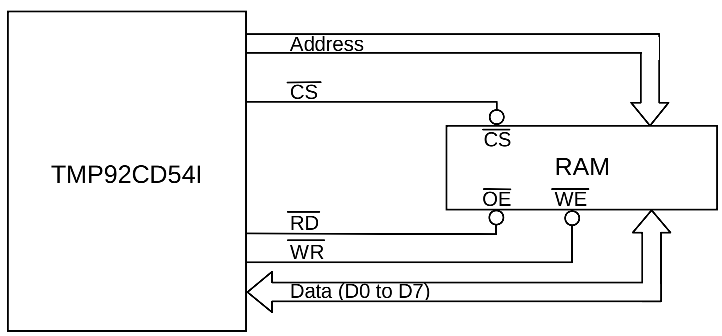


Figure 3.6.1 Example of connecting external memory (external RAM)

(i) Setting memory start address register

The MS23 to 16 bits of MSAR respectively correspond with addresses A23 to A16. The lower start address A15 to 0 is always set to address 0000H. Therefore, the start addresses of the block address area are set to addresses 000000H to FF0000H every 64KB (Because the settable LSB bit is 16^{th}; 2^{16} = 64 KB)

(ii) Setting memory address mask register

MAMR sets whether an address bit is compared or not. Set the register to 0 to compare, or to 1 not to compare. The combination of masked / enabled bits give the block size and therefore the address bit to be set depends on the block address area.

Note: **A23 is always compared**.

Thus, the block address area is between A22 and A15.

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

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

Note: After reset release, BCSH<BM> (block address area specification) is set to '0', and the block address area is set to addresses 000000H to FFFFEFH. Setting BCSH<BM> to "1" specifies the start address (using MSAR) and the address area size (using MAMR).

(iii) Example of register setting

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
```

Memory Start Address Register MSAR<MS23:16> correspond with address A23 to A16. A15 to A0 are set to '0'. Therefore setting MSAR to the above mentioned value specifies the start address of the block address area to address 110000H.

Memory Address Mask Register MAMR<MV22:15> set whether address A22 to 15 are compared or not. Set the register to '0' to compare, or to '1' not to compare. Remember that A23 is always compared. Setting the above-mentioned compares A23 to A16 with the values set as the start addresses. Therefore the block size is 64 KB (since the first bit set to 0 is A16 ➔ 2^{16} = 64 KB)

To summarize, 64 KB of addresses 110000H to 11FFFFH are set as the block address area, and compared with the addresses on the bus. If the compared result is a match, the chip select signal $\overline{\text{CS}}$ is set to Low.

(iv) Case of overlapping blocks

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

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

This means that the block address is not remapped but priorities are used to disable any conflict.

Also note that any accessed areas outside the address spaces are set to 1 wait bus cycle ($\overline{\text{CS}}$ signal is not outputted although $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signal are outputted.). This factor depends on the speed of the external memory. It is a fixed parameter.

(2) Wait control

The external bus cycle completes a wait of two states at least (i.e. 100ns @fc = 20MHz). Setting the control register BCSL<BWW2:0> and <BWR2:0> specifies the number of waits in the read cycle and the write cycle. <BWWn> is set using the same method as for <BWRn>.
Note that this setting is only for asynchronisation purpose.

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) Waits number fixed mode

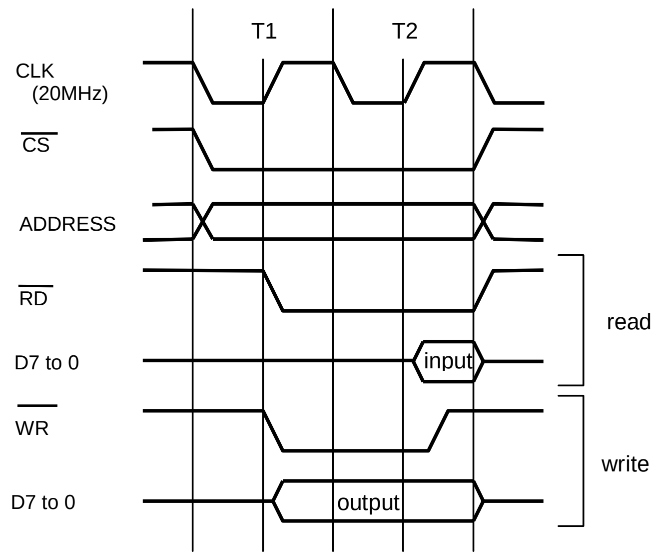
The bus cycle is completed with the set states. The number of states is selected from 2 states (0WAIT) to 5 states (3WAIT).

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

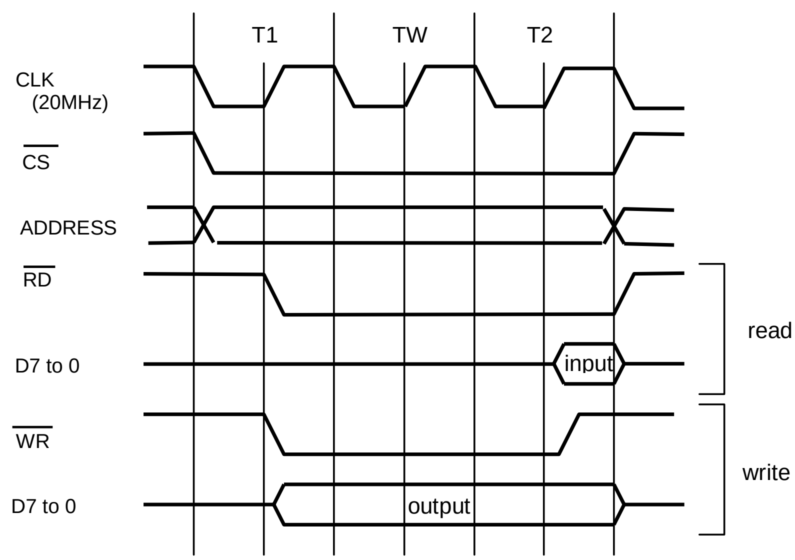
This mode continuously samples the $\overline{\text{WAIT}}$ input pins and inserts a wait if the pin is active. The bus cycle is minimum 2 states and is therefore completed at 2 states when the wait signal is non active (High level). The bus cycle extends if the wait signal is active at 2 states and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

(3) Basic bus timing

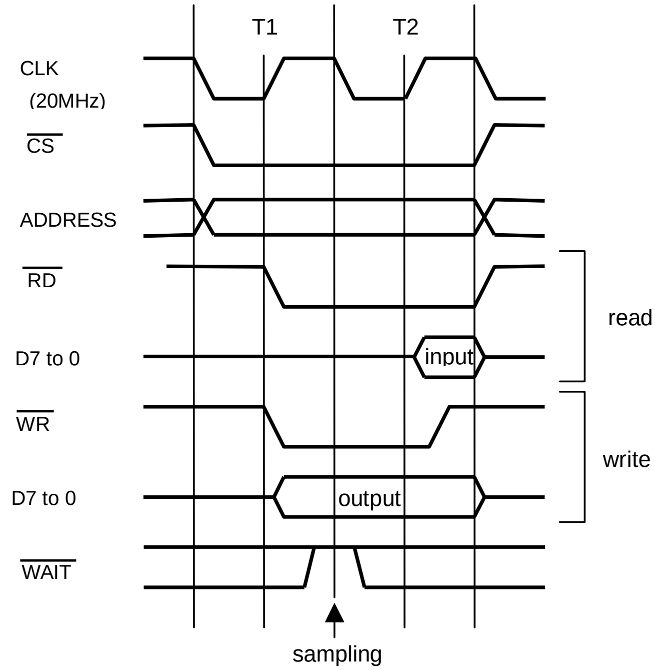
- External Read / Write Bus Cycle (0 WAIT)



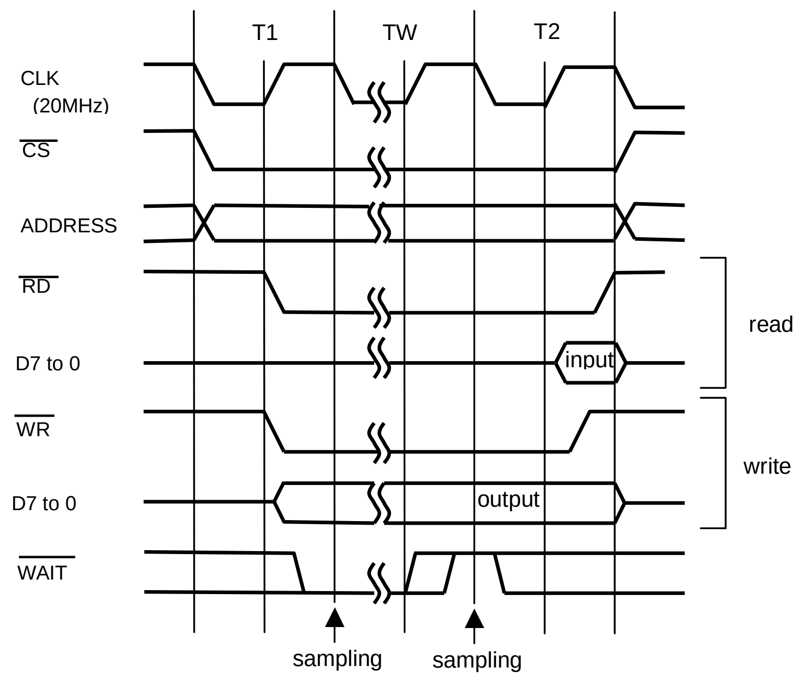
- External Read / Write Bus Cycle (1 WAIT)



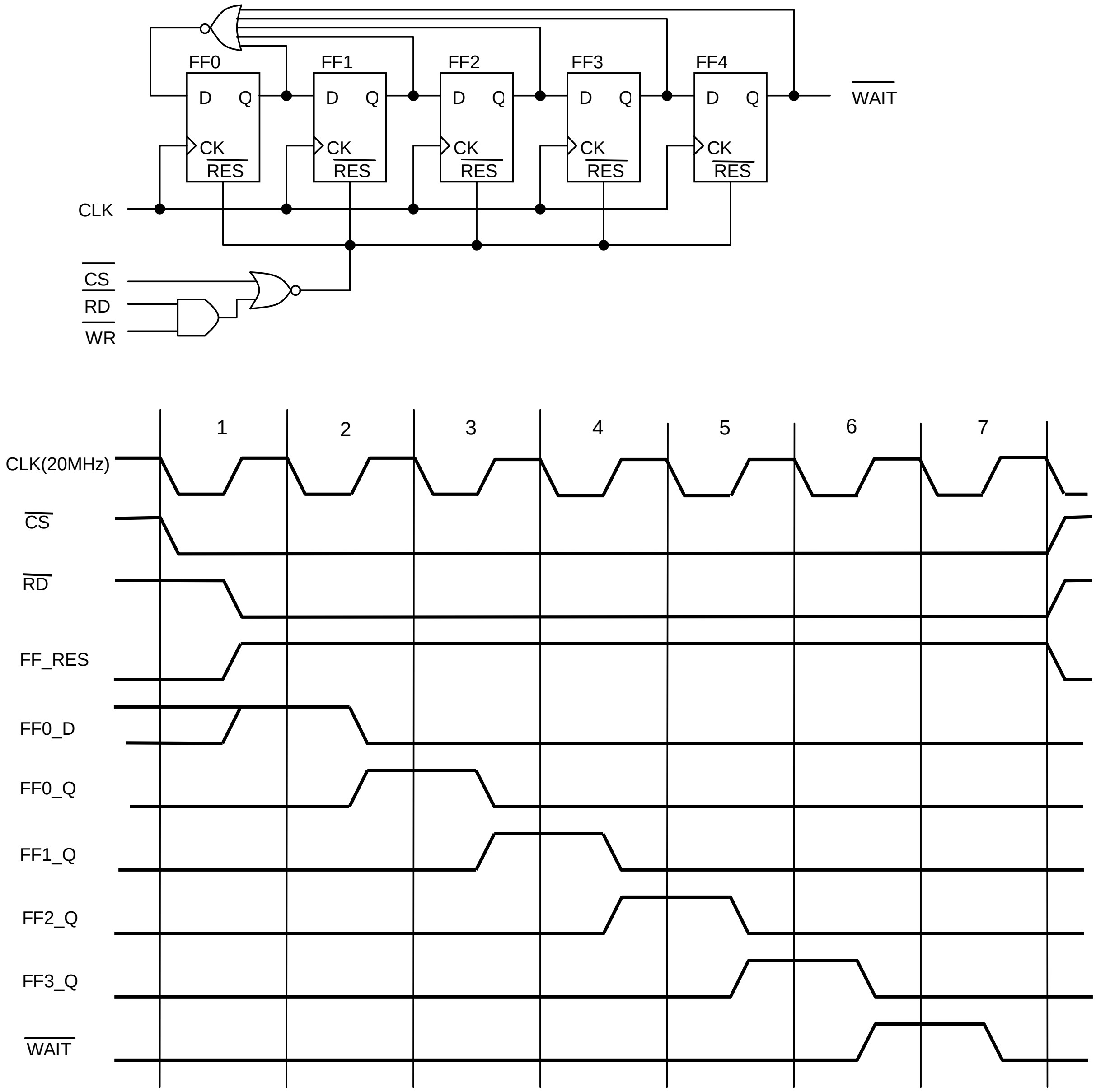
- External Read / Write Bus Cycle (0 WAIT @ WAIT pin input mode)



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



- Example of WAIT Input Cycle (5WAIT)



3.6.4 List of registers

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

(1) Control registers

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

Block CS/WAIT control register (Low)

BCSL (0148H)

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

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

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

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

011 = WAIT pin input mode Others = (Reserved)

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

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

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

011 = WAIT pin input mode Others = (Reserved)

Block CS/WAIT control register (High)

BCSH (0149H)

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

BE Enable bit

0 = No chip select signal output

1 = Chip select signal output (Default)

BM Block address area specification

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

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

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

BOM[1:0]

00 = SRAM or ROM(Default)

others = (Reserved)

BBUS[1:0] Sets the data bus width

00 = 8-bit (Default)

others = (Reserved)

(2) Block address register

A start address and an address area of the block address are specified by the memory start address register (MSAR) and the memory address mask register (MAMR). The bit to be set by the memory address mask register depends on the block address area.

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

MS[23:16] Sets a start address.

Sets the start address of the block address areas. <MS23:16> correspond to the address 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

MV[22:15]

Enables or masks comparison of the addresses. <MV22:15> correspond to addresses A22 to 15. If “0” is set, the comparison between the value of the address bus and the start address is enabled. If “1” is set, the comparison is masked.

3.7 8-bit Timers

TMP92CD54I features eight built-in 8-bit timers (timers 0 to 7).
These timers are paired into four modules: timers 01, timers 23, timers 45, and timers 67. Each module consists of two channels and can operate in any of the following four operating modes.

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

Figure 3.7.1 to Figure 3.7.4 show block diagrams for timers 01, timers 23, timers 45 and timers 67.
Each channel consists of an 8-bit up-counter, an 8-bit comparator and an 8-bit timer register. In addition, a timer flip-flop and a prescaler are provided for each pair of channels.
The operation mode and timer flip-flops are controlled by five control SFRs (special-function registers).
Each of the four modules (timers 01, timers 23, timers 45 and timers 67) can be operated independently. All modules operate in the same manner; hence only the operation of timers 01 is explained here.

Table 3.7.1 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 diagrams

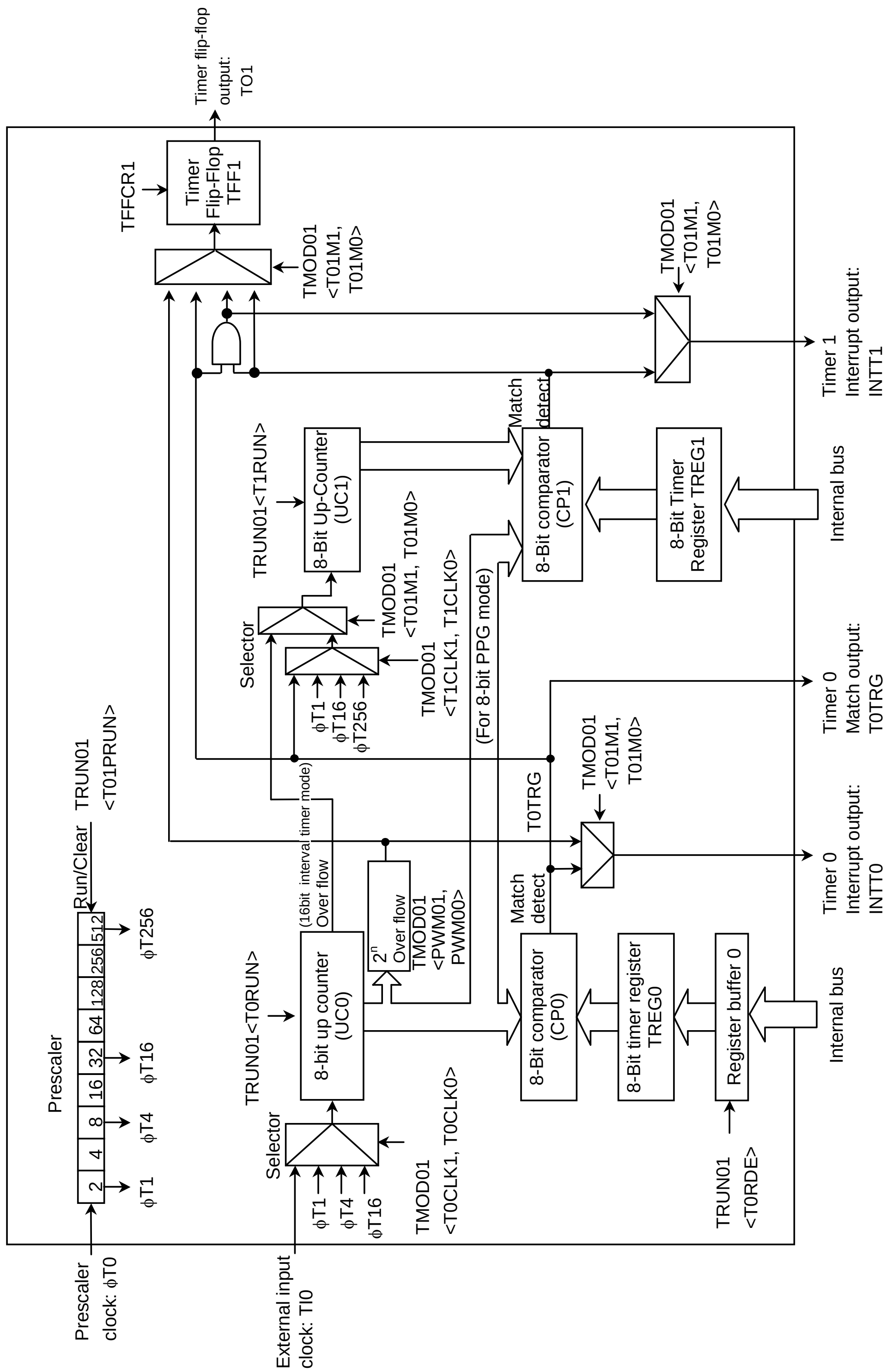


Figure 3.7.1 Timers 01 block diagram

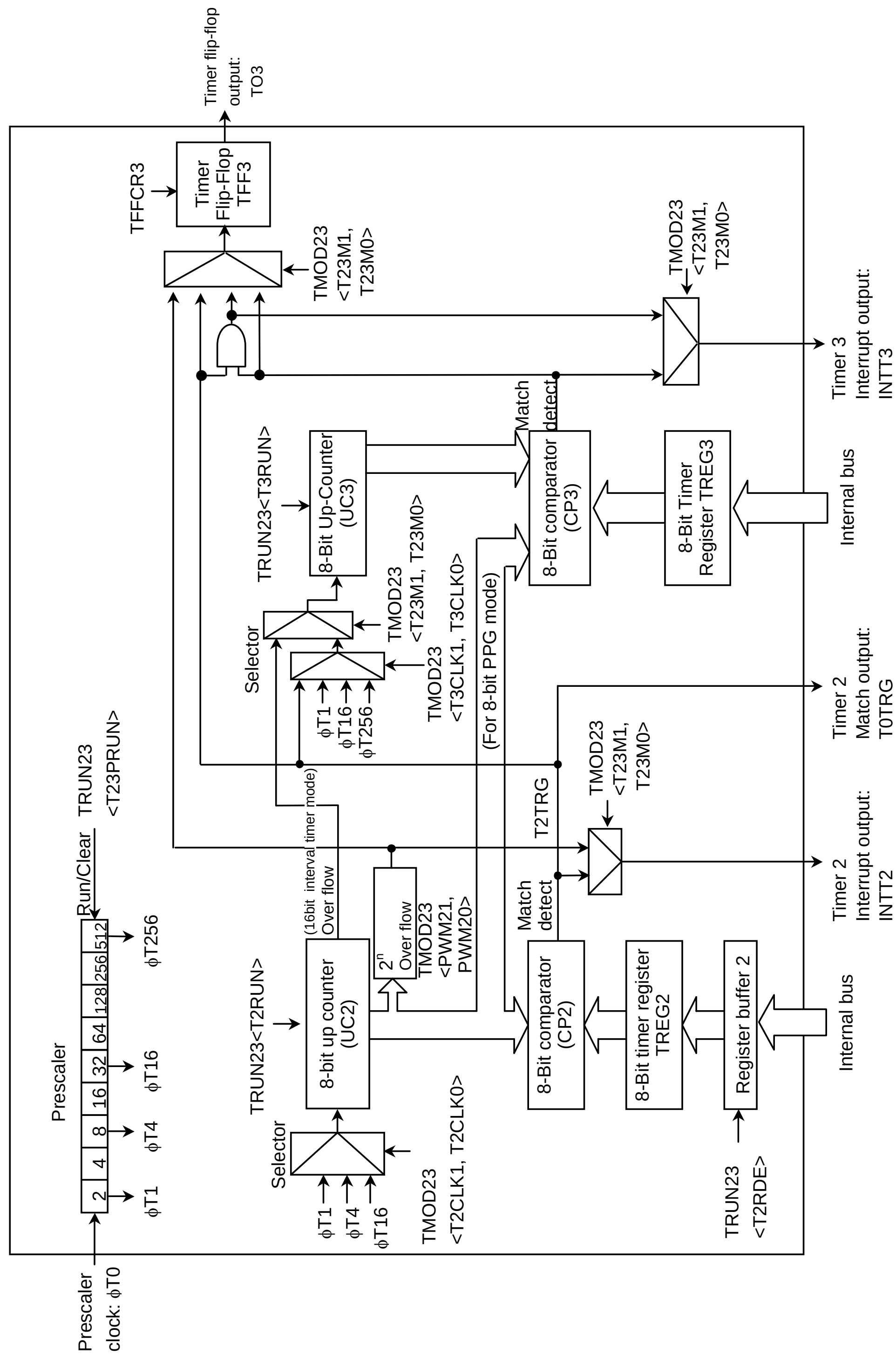


Figure 3.7.2 Timers 23 block diagram

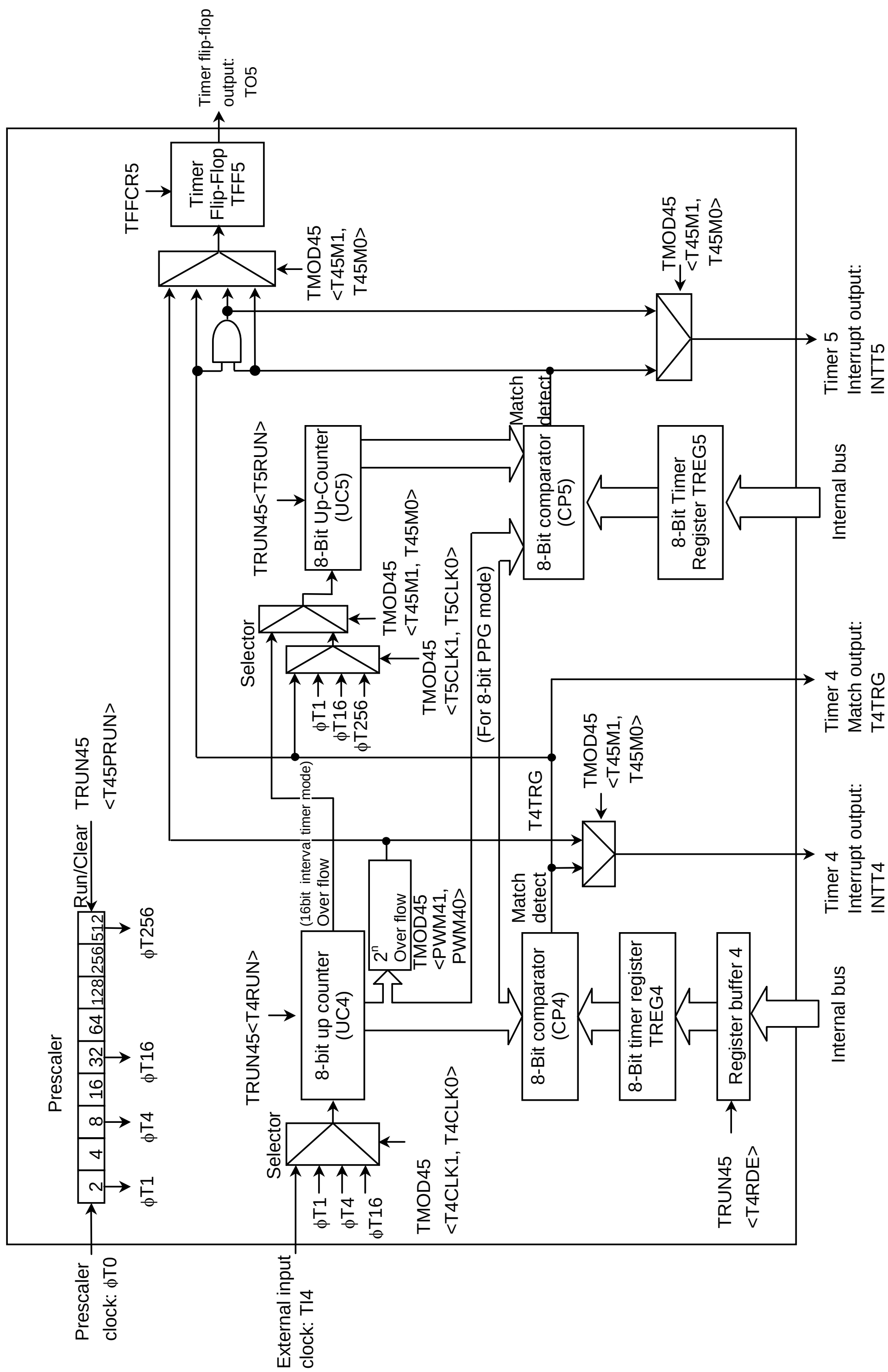


Figure 3.7.3 Timers 45 block diagram

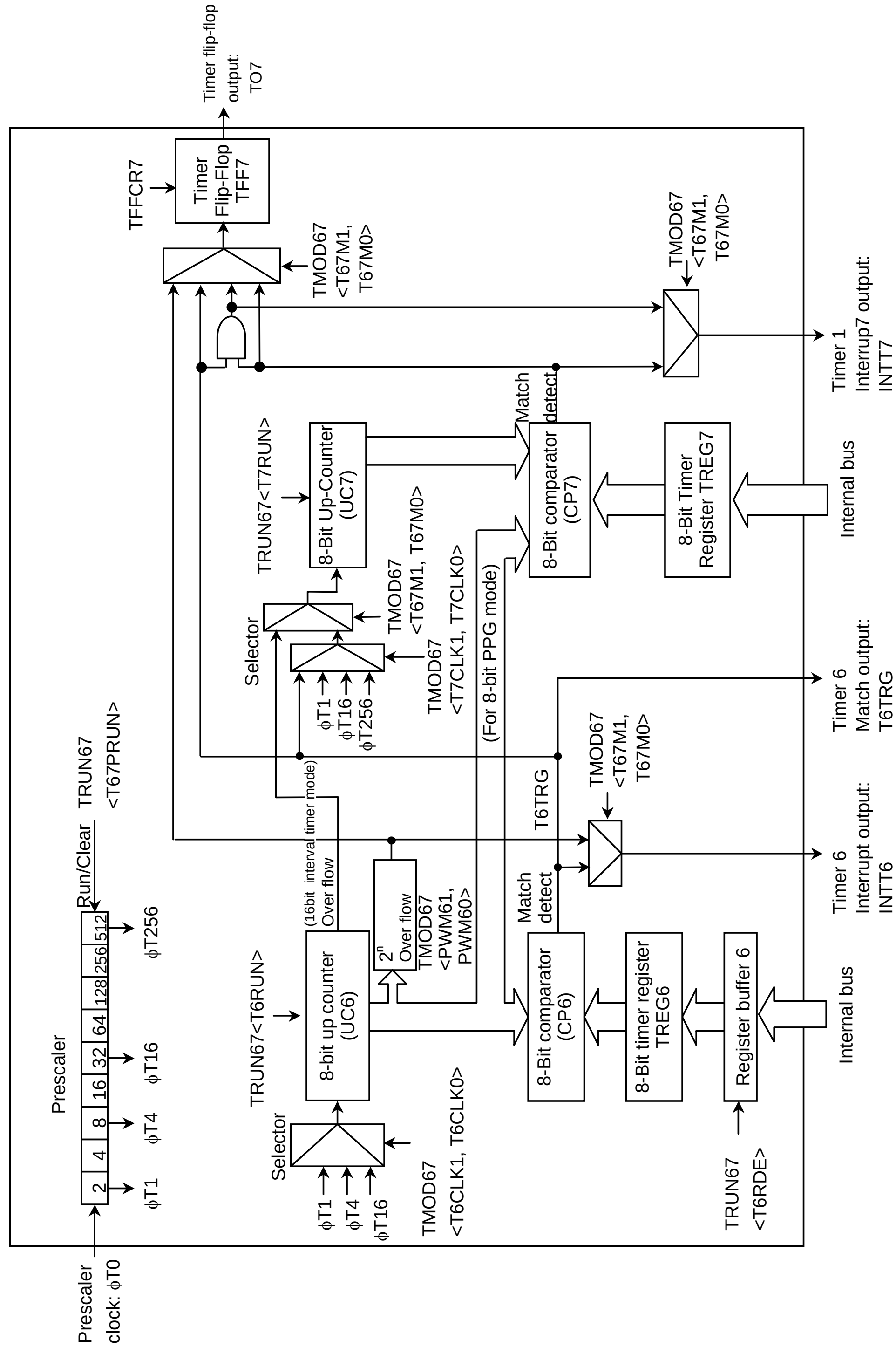


Figure 3.7.4 Timers 67 block diagram

3.7.2 Operation of each circuit

(1) Prescalers

A 9-bit prescaler generates the input clock to timers 01.
The clock T0 is the CPU clock fc divided by 4 and is the input to this prescaler.

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

At fc=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 following number in the parenthesis indicates the frequency when TMP92CD54I operates is the maximum frequency.

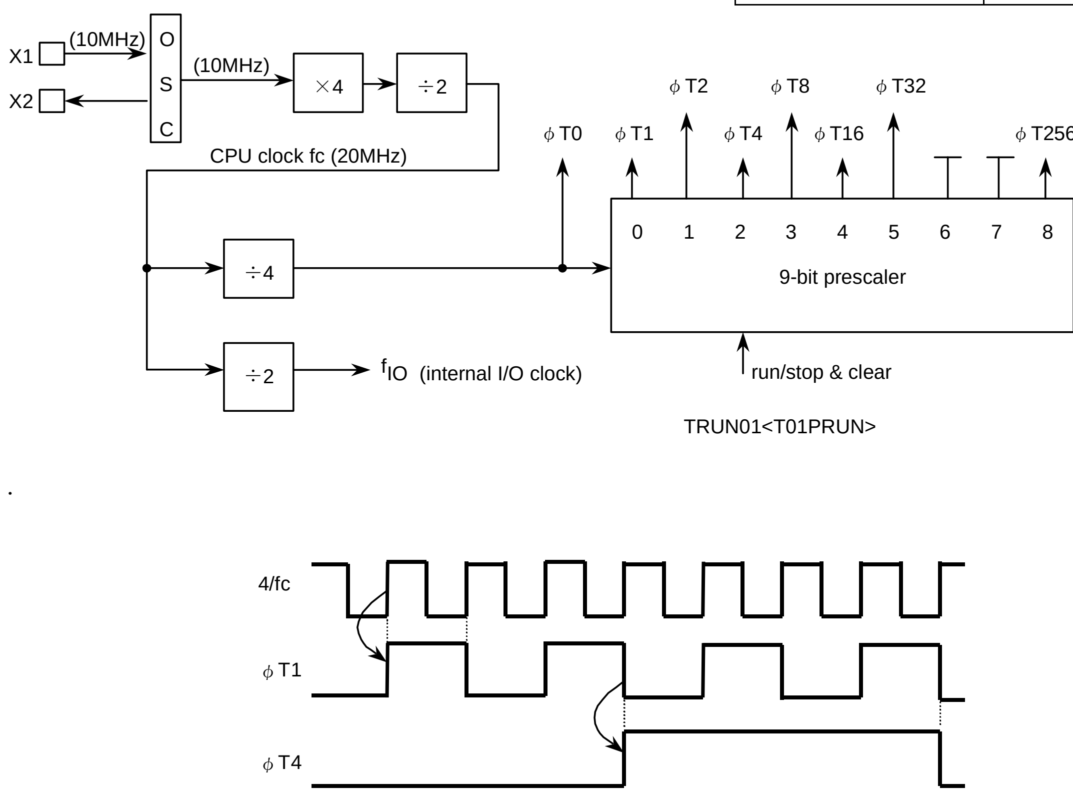


Figure 3.7.5 Prescaler

(2) Up-counters (UC0 and UC1)

These are 8-bit binary counters which count up the input clock pulses for the clock specified by TMOD01.
The input clock for UC0 is selectable and can be either the external clock input via the TI0 pin or one of the three internal clocks ϕT1, ϕT4 or ϕT16. The clock setting is specified by the value set in TMOD01<T01CLK1,T01CLK0>.
The input clock for UC1 depends on the operation mode. In 16-Bit Interval Timer Mode, the overflow output from UC0 is used as the input clock. In any mode other than 16-Bit Interval Timer Mode, the input clock is selectable and can either be one of the internal clocks ϕT1, ϕT16 or ϕT256, or the comparator output (the match detection signal) from timer 0.

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

(3) Timer registers (TREG0 and TREG1)

These are 8-bit registers which can be used to set a time interval. When the value set in the timer register TREG0 or TREG1 matches the value in the corresponding up-counter, the Comparator Match Detect signal goes Active. If the value set in the timer register is 00H, the signal goes Active when the up-counter overflows.
The TREG0 are double buffer structure, each of which makes a pair with register buffer.

The setting of the bit TRUN01<T0RDE> determines whether TREG0's double buffer structure is enabled or disabled. It is disabled if <T0RDE> = 0 and enabled if <T0RDE> = 1.

When the double buffer is enabled, data is transferred from the register buffer to the timer register when a 2^n overflow occurs in PWM Mode, or at the start of the PPG cycle in PPG Mode. Hence the double buffer cannot be used in Interval Timer Mode.

A Reset initializes <T0RDE> to 0, disabling the double buffer. To use the double buffer, write data to the timer register, set <T0RDE> to 1, and write the following data to the register buffer. Figure 3.7.6 shows the configuration of TREG0.

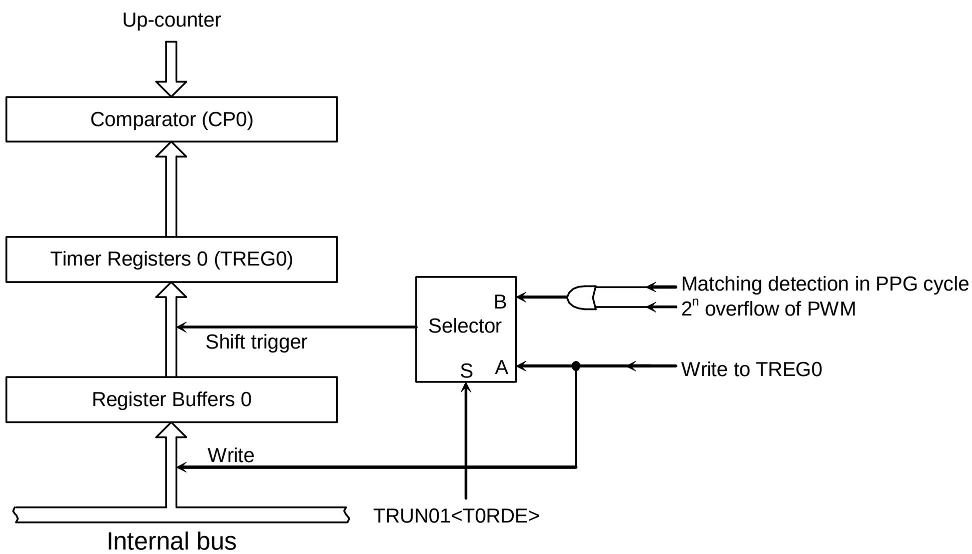


Figure 3.7.6 Configuration of TREG0

Note: The same memory address is allocated to the timer register and the register buffer. When <T0RDE> = 0, the data is written in both registers (i.e. the Register buffer 0 and the 8-bit timer register TREG0) at the same time; when <T0RDE> = 1, only the register buffer is written to.

The address of each timer register is as follows.

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

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

(4) Comparator (CP0)

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

(5) Timer flip-flop (TFF1)

The timer flip-flop (TFF1) is a flip-flop inverted by the match detect signal (8-bit comparator output) of each interval timer.
Whether inversion is enabled or disabled is determined by the setting of the bit TFFCR1<TFF1IE> in the Timer Flip-Flop Control Register.
A Reset clears the value of TFF1 to 0. Writing 01 or 10 to TFFCR1<TFF1C1,TFF1C0> sets TFF1 to 0 or 1. Writing 00 to these bits inverts the value of TFF1 (this is known as software inversion).
The TFF1 signal is output via the TO1 pin (which can also be used as PC1). When this pin is used as the timer output, the timer flip-flop should be set beforehand using the Port C Function Register PCFC.

TFF is inverted by ….

8-bit interval timer mode	: UC0 matches TREG0. Or when UC1 matches TREG1. (Either one of the two is chosen)
16-bit interval timer mode	: UC0 matches TREG0 and UC1 matches TREG1.
8-bit PWM mode	: UC0 matches TREG0 or 2^n overflow is occurred.
8-bit PPG mode	: UC0 matches TREG0 or UC0 matches TREG1.

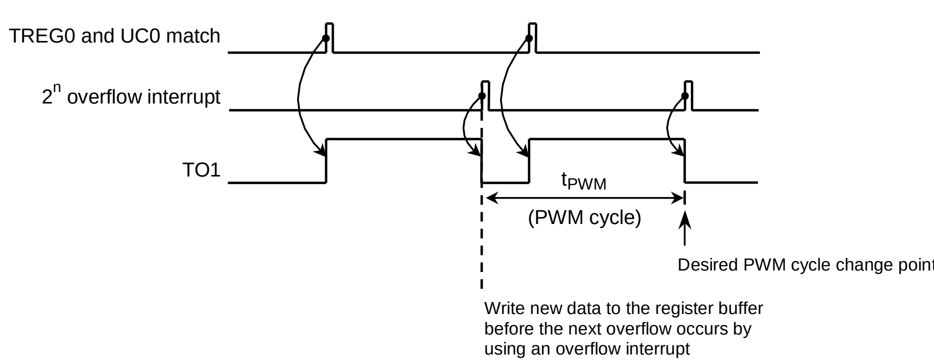
Note: When the double buffer is enabled for an 8-bit timer in PWM or PPG mode, caution is required as explained below.

If new data is written to the register buffer immediately before an overflow occurs by a match between the timer register value and the up-counter value, the timer flip-flop may output an unexpected value.

For this reason, make sure that in PWM mode new data is written to the register buffer by six cycles (fc × 6) before the next overflow occurs by using an overflow interrupt.

In the case of using PPG mode, make sure that new data is written to the register buffer by six cycles before the next cycle compare match occurs by using a cycle compare match interrupt.

Example when using PWM mode:



3.7.3 SFRs

Timers 01 Run Register

TRUN01 (0080H)

	7	6	5	4	3	2	1	0
Bit symbol	T0RDE	-	-	-	I2T01	T01PRUN	T1RUN	T0RUN
Read/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)		

TREG0 double buffer control

0	Disable
1	Enable

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

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

Note1: The values of bits 4 to 6 of TRUN01 are undefined when read.

Note2: Needs to set <T0RDE> bit and enable double buffer in PPG/PWM mode.

Timers 23 Run Register

TRUN23 (0088H)

	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)		

TREG2 double buffer control

0	Disable
1	Enable

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

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

Note1: The values of bits 4 to 6 of TRUN23 are undefined when read.

Note2: Needs to set <T2RDE> bit and enable double buffer in PPG/PWM mode.

Figure 3.7.7 Register for 8-bit Timers

Timers 45 Run 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 during IDLE2-Mode
T45PRUN: Run for prescaler
T5RUN: Run Timer 5
T4RUN: Run Timer 4

Note1: The values of bits 4 to 6 of TRUN45 are undefined when read.

Note2: Needs to set <T4RDE> bit and enable double buffer in PPG/PWM mode.

Timers 67 Run 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 during IDLE2 Mode
T67PRUN: Run prescaler
T7RUN: Run Timer 7
T6RUN: Run Timer 6

Note1: The values of bits 4 to 6 of TRUN67 are undefined when read.

Note2: Needs to set <T6RDE> bit and enable double buffer in PPG/PWM mode.

Figure 3.7.8 Register for 8-bit Timers

Timers 01 Mode Register

TMOD01 (0084H)		7	6	5	4	3	2	1	0
	Bit symbol	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
	Read/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 : When setting the TI0 pin, first set the Port C setting, then TMOD01.

Figure 3.7.9 Register for 8-bit Timers

Timers 23 Mode Register

TMOD23 (008CH)

	7	6	5	4	3	2	1	0
Bit Symbol	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
Read/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 (16-Bit Timer Mode)
01	ϕT1	
10	ϕT16	
11	ϕT256	

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

Timers 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 (16-Bit Timer Mode)
01	ϕT1	
10	ϕT16	
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 45

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

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

Figure 3.7.11 Register for 8-bit Timers

Timers 67 Mode register

TMOD67 (009CH)		7	6	5	4	3	2	1	0
	Bit symbol	T67M1	T67M0	PWM61	PWM60	T7CLK1	T7CLK0	T6CLK1	T6CLK0
	Read/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	

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 (16-Bit Timer Mode
01	ϕT1	
10	ϕT16	
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

Timer 1 Flip-Flop Control Register

TFFCR1 (0085H)		7	6	5	4	3	2	1	0
	Bit symbol	-	-	-	-	TFF1C1	TFF1C0	TFF1IE	TFF1IS
	Read/Write					R/W			
	After Reset	-	-	-	-	1	1	0	0
Read-Modify -Write instructions are prohibited.	Function					00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		TFF1 Control for inversion 0: Disable 1: 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: The values of bits 4 to 7 of TFFCR1 are undefined when read.

Figure 3.7.13 Register for 8-bit Timers

Timer 3 Flip-Flop Control Register

TFFCR3
(008DH)

Read-Modify-Write instructions are prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	TFF3C1	TFF3C0	TFF3IE	TFF3IS
Read/Write					R/W			
After Reset	-	-	-	-	1	1	0	0
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: The values of bits 4 to 7 of TFFCR3 are undefined when read.

Figure 3.7.14 Register for 8-bit Timers

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 instructions are prohibited.	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: The values of bits 4 to 7 of TFFCR5 are undefined when read.

Figure 3.7.15 Register for 8-bit Timers

Timer 7 Flip-Flop Control Register

TFFCR7 (009DH)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	TFF7C1	TFF7C0	TFF7IE	TFF7IS
Read/Write					R/W			
After Reset	-	-	-	-	1	1	0	0
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

Read-Modify-Write instructions are prohibited.

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

0	Inversion by Timer 6
1	Inversion by Timer 7

Inversion of TFF7

0	Disabled
1	Enabled

Control of TFF7

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

Note: The values of bits 4 to 7 of TFFCR7 are undefined when read.

Figure 3.7.16 Register for 8-bit Timers

Timer Register (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							

TREG is for the comparator (When UC matches TREG, occur match detect signal). Refer to setting example in Section 3.7.4, Operation in each mode.

Figure 3.7.17 Register for 8-bit Timers

3.7.4 Operation in each mode

(1) 8-Bit interval Timer Mode

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

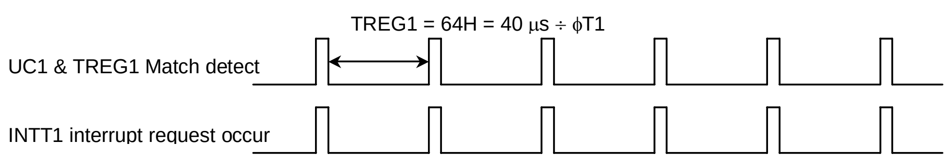
① Generating interrupts at a fixed interval (using timer 1)

To generate interrupts at constant intervals using timer 1 (INTT1), first stop timer 1 then set the operation mode, input clock and a cycle to TMOD01 and TREG1 register, respectively. Then, enable the interrupt INTT1 and start timer 1 counting.

Example: To generate an INTT1 interrupt every 40 μseconds at fc = 20 MHz, set each register as follows:

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

Note: X = Don't care; "–" = No change



Select the input clock using Table 3.7.2.

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

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

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

② Generating a 50% duty ratio square wave pulse

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

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

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

Note: X = Don't care; "–" = No change

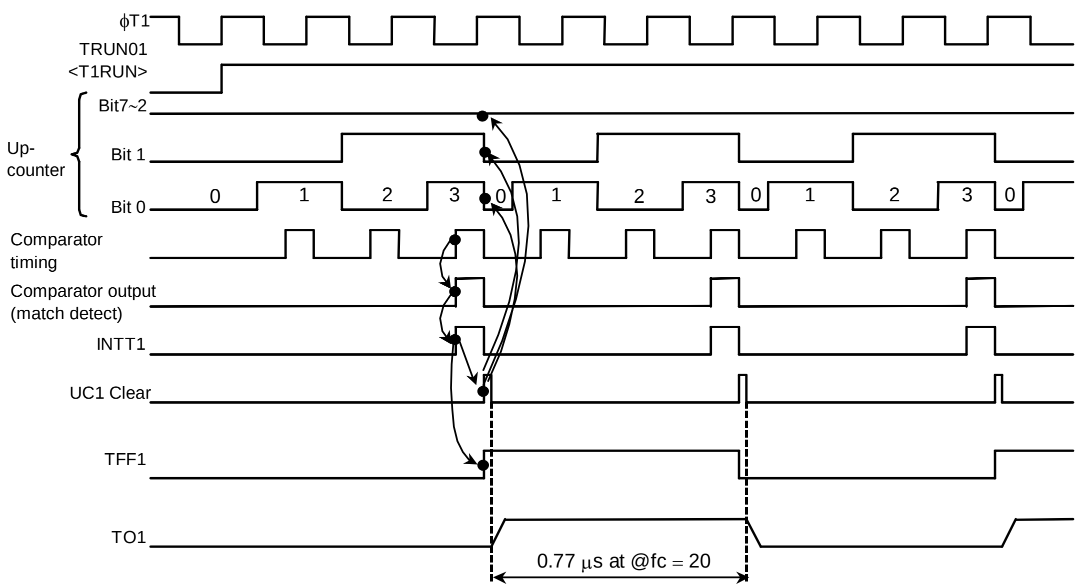


Figure 3.7.18 Square wave output timing chart (50% Duty)

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

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

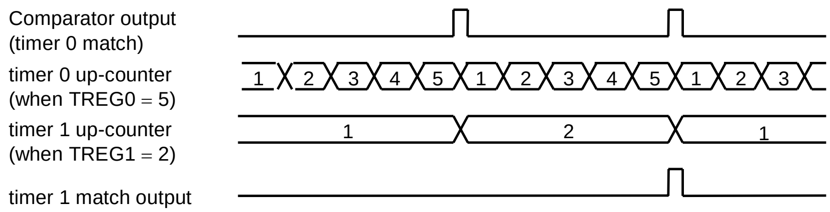


Figure 3.7.19 Timer 1 Count Up on Signal from Timer 0

(2) 16-Bit interval Timer Mode

A 16-bit interval timer is configured by pairing the two 8-bit timers timer 0 and timer 1.

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

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

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

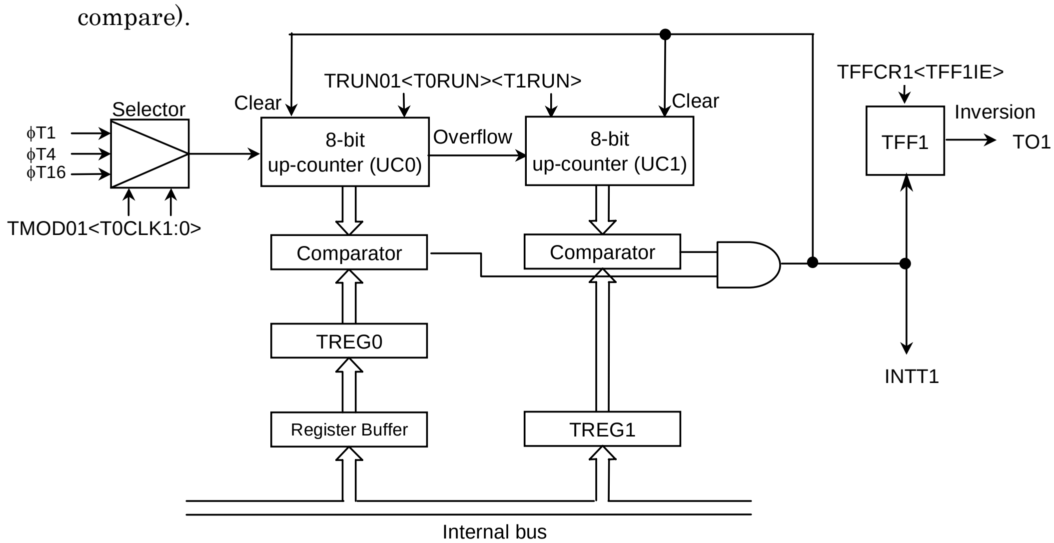


Figure 3.7.19 Block diagram of 16-Bit interval timer Mode

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

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

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

In the case of the timer 1 comparator, the match detect signal is output on each comparator pulse on which the values in the up-counter UC1 and TREG1 match.

When the match detect signal is output simultaneously from both the comparators timer 0 and timer 1, the up-counters UC0 and UC1 are cleared to 0 and the interrupt INTT1 is generated. Also, if inversion is enabled, the value of the timer flip-flop TFF1 is inverted.

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

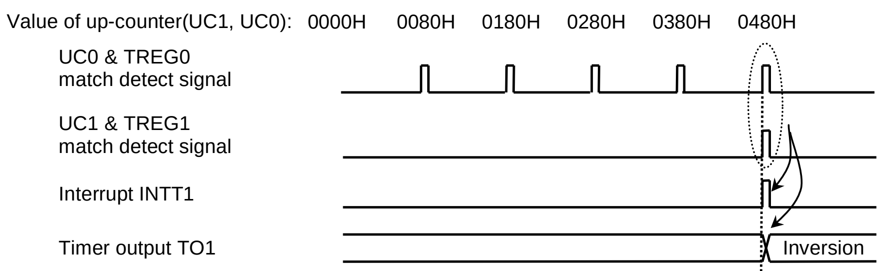


Figure 3.7.20 Timer output by 16-Bit Interval Timer Mode

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

Square wave pulses can be generated at any frequency and duty ratio by timer 0. The output pulses may be active-Low or active-High. In this mode timer 1 cannot be used. Timer 0 outputs pulses on the TO1 pin (which can also be used as PC1).

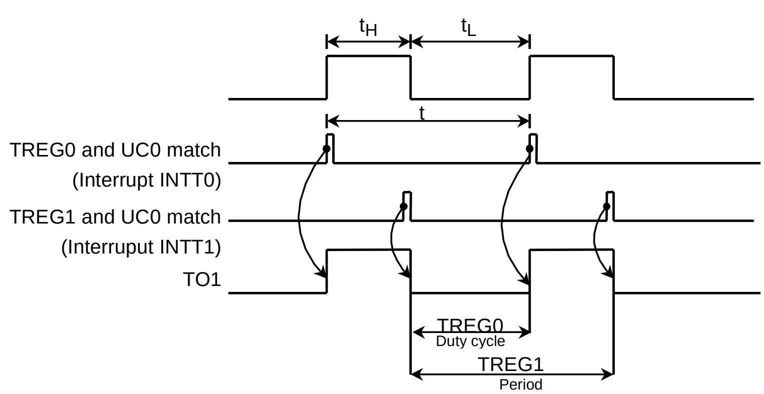


Figure 3.7.21 8 bit PPG output waveforms

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

The value set in TREG0 must be smaller than the value set in TREG1.

Although the up-counter for timer 1 (UC1) is not used in this mode, TRUN01<T1RUN> should be set to 1 (To enable the comparator to compare with TREG1) so that UC1 is set for counting.

Figure 3.7.22 shows a block diagram representing this mode.

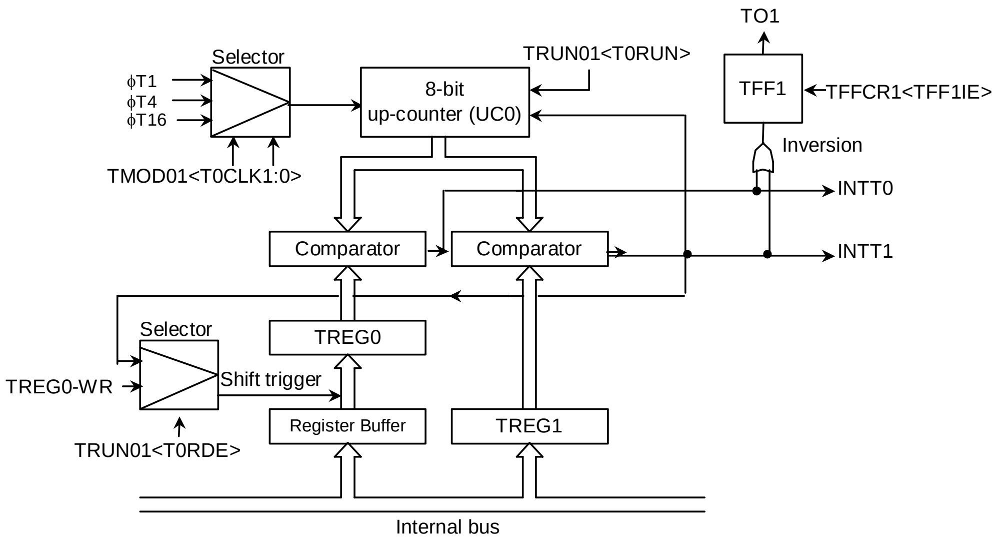


Figure 3.7.22 Block diagram of 8-Bit PPG Output Mode

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

Use of the double buffer facilitates the handling of low-duty waves (when duty is varied).

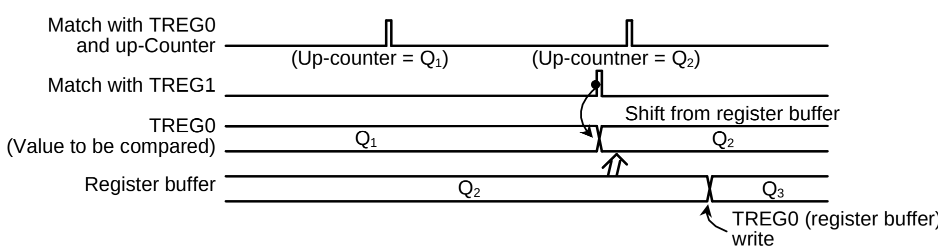
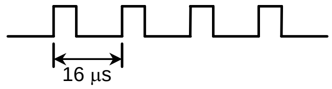


Figure 3.7.23 Operation of register buffer

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



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

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

ϕT1 = 0.4 µs (at 20 MHz);

16 µs ÷ 0.4 µs = 40

Therefore set TREG1 to 40 (28H)

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

4 µs ÷ 0.4 µs = 10

Therefore, set TREG0 = 10 = 0AH.

		7	6	5	4	3	2	1	0	
TRUN01	←	0	X	X	X	–	0	0	0	Stop timer 0 and timer 1 and clear it to “0”.
TMOD01	←	1	0	X	X	X	X	0	1	Set the 8-bit PPG mode, and select ϕT1 as input clock.
TREG0	←	0	0	0	0	1	0	1	0	Write 0AH
TREG1	←	0	0	1	0	1	0	0	0	Write 28H
TFFCR1	←	X	X	X	X	0	1	1	X	Set TFF1 and enable inversion.
										10 generates a negative logic pulse.
PCCR	←	X	X	–	–	–	–	1	–	Set PC1 as the TO1 pin.
PCFC	←	X	X	–	–	-	–	1	-	
TRUN01	←	1	X	X	X	–	1	1	1	Set double buffer enable, and start timer 0 and timer 1 counting.

Note: X = Don’t care; “–” = No change

(4) 8-Bit Pulse with Modulation (PWM) Output Mode

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

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

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

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

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

Value set in TREG0 ≠ 0

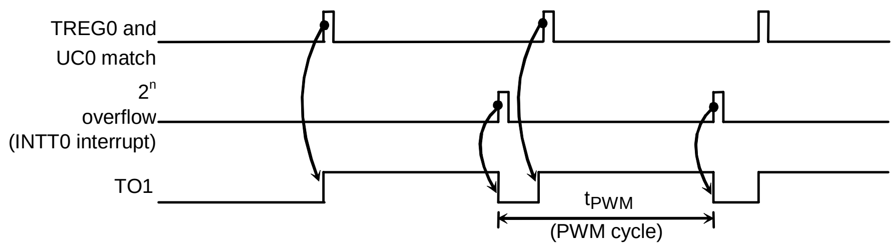


Figure 3.7.24 8-bit PWM waveforms

Figure 3.7.25 shows a block diagram representing this mode.

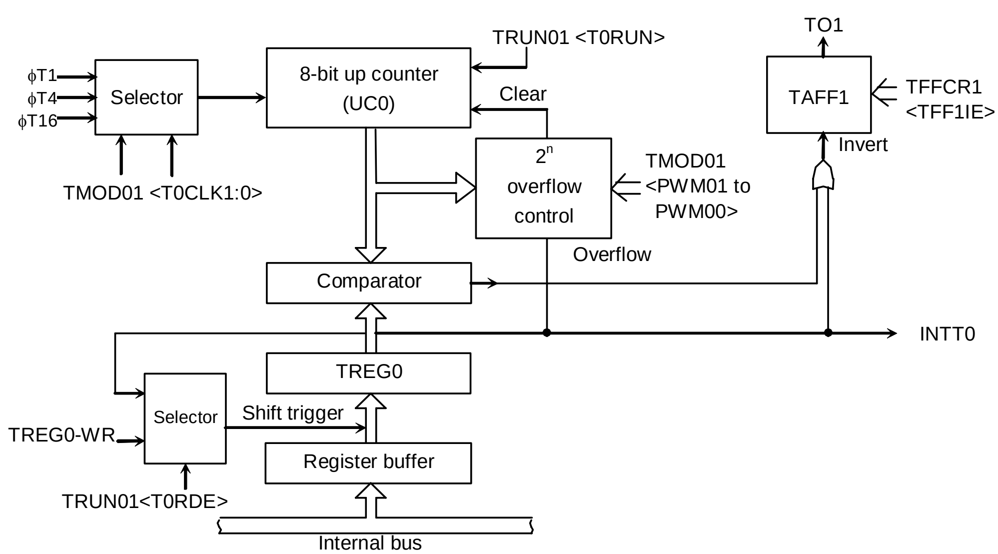


Figure 3.7.25 Block diagram of 8-Bit PWM Mode

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

Use of the double buffer facilitates the handling of low duty ratio waves.

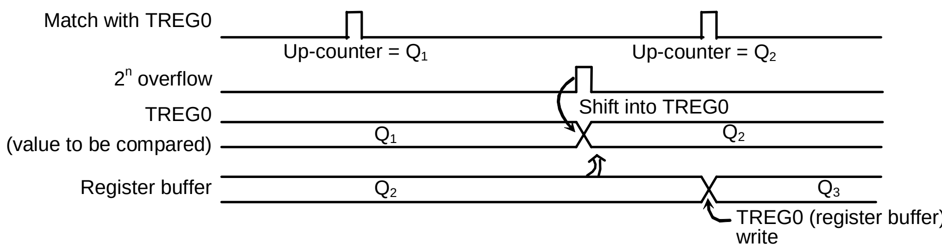
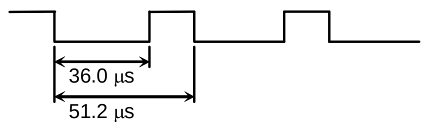


Figure 3.7.26 Register buffer operation

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



To achieve a 51.2-μs PWM cycle by setting φT1 to 0.4 μs (at fc = 20 MHz):

51.2 μs ÷ 0.4 μs = 128

$2^n = 128$

Therefore n should be set to 7.

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

set the following value for TREG0:

36.0 μs ÷ 0.4 μs = 90 = 5AH

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

Note: X = Don't care; "–" = No change

Table 3.7.3 PWM cycle

	PWM Interval (at fc = 20MHz)		
	ϕT1	ϕT4	ϕT16
2^6	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 mode

Table 3.7.4 shows the SFR settings for each mode.

Table 3.7.4 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

Note:“–” = Don’t care

3.8 16-Bit Timer/Event Counters

TMP92CD54I incorporates two multifunctional 16-bit timer/event counters (timer 8 and timer A) which have the following operation modes:

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

Can be used following operation modes by capture function:

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

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

Each timer/event counter channel consists of a 16-bit up-counter, two 16-bit timer registers (one of them with a double-buffer structure), two 16-bit capture registers, two comparators, a capture input controller, a timer flip-flop and a control circuit.
Each timer/event counter is controlled by an 11-byte control SFR.
The two channels (timer 8 and timer A) can be used independently.
Both channels feature the same operations except for those described in Table 3.8.1. Thus, only the operation of timer 8 is explained below.

Table 3.8.1 Differences between Timer 8 and Timer A

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

3.8.1 Block diagrams

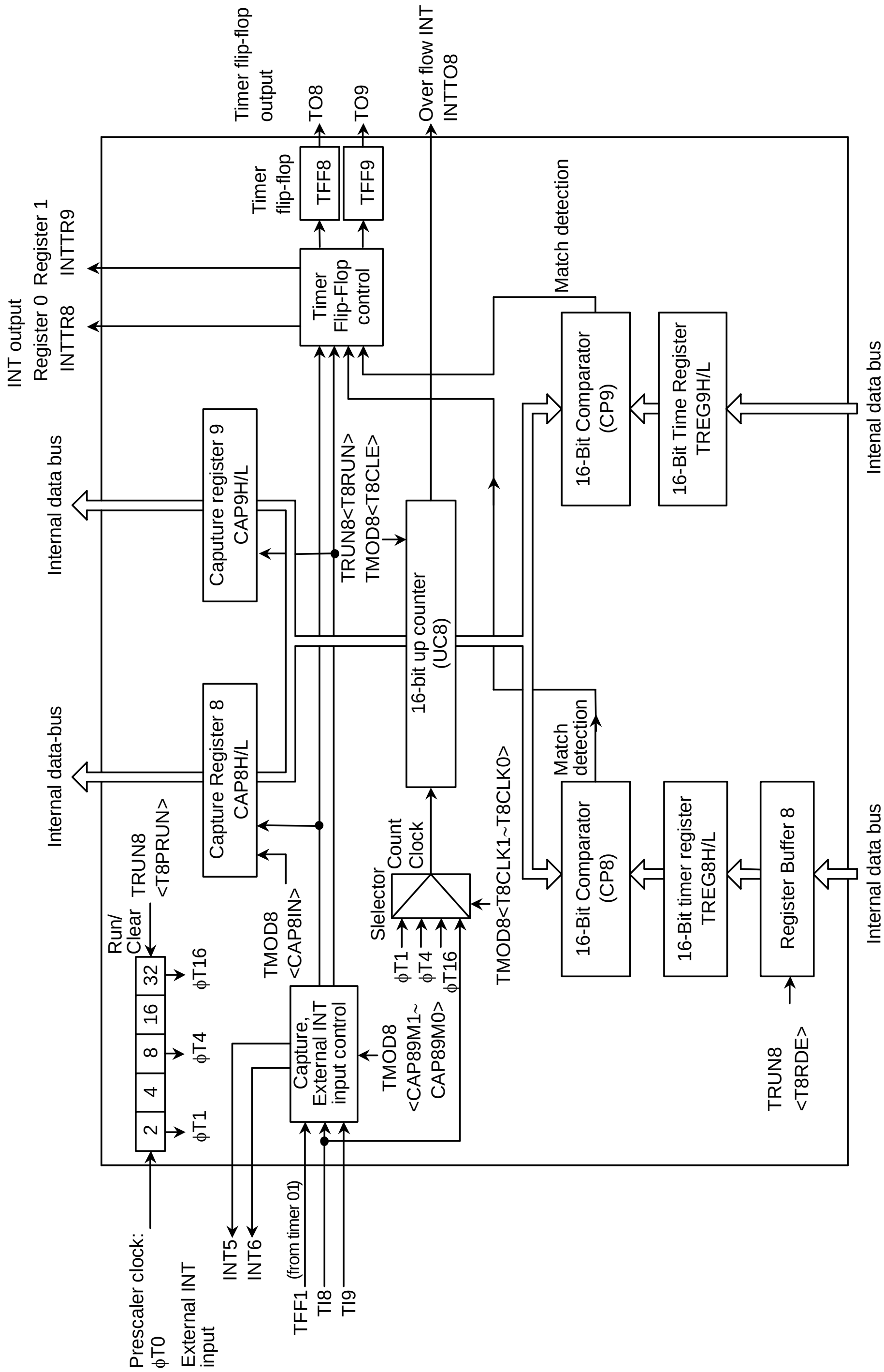


Figure 3.8.1 Block Diagram of Timer 8

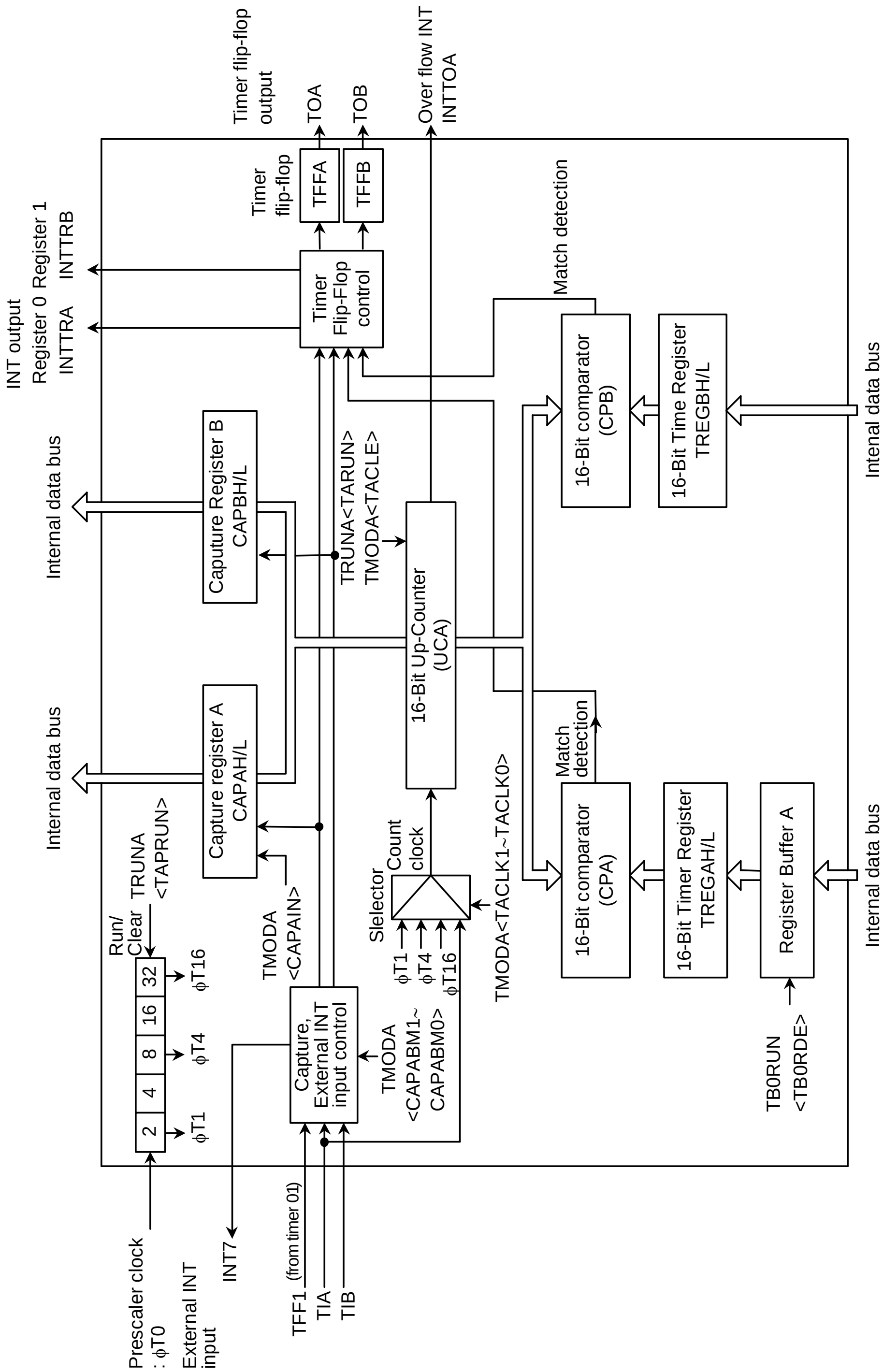


Figure 3.8.2 Block diagram of Timer A

3.8.2 Operation of each block

(1) Prescaler

The 5-bit prescaler generates the source clock for timer 8. The prescaler clock (ϕT0) is divided clock (divided by 4) from fc.
This prescaler can be started or stopped using TRUN8<T8PRUN>. Counting starts when <T8PRUN> is set to 1; the prescaler is cleared to zero and stops operation when <T8PRUN> is set to 0.

Table 3.8.2 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)

UC8 is a 16-bit binary counter which counts up pulses input from the clock specified by TMOD8 <T8CLK1,T8CLK0>.
Any one of the prescaler internal clocks ϕT1, ϕT4 and ϕT16 or an external clock input via the TI8 pin can be selected as the input clock. Counting or stopping & clearing of the counter is controlled by TRUN8<T8RUN>.
When clearing is enabled, the up-counter UC8 will be cleared to zero each time its value matches the value in the timer register TREG9H/L. Clearing can be enabled or disabled using TMOD8<T8CLE>.
If clearing is disabled, the counter operates as a free-running counter.
A Timer Overflow interrupt (INTTO8) is generated when UC8 overflow occurs.

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

These two 16-bit registers are used to set the interval time. When the value in the up-counter UC8 matches the value set in this timer register, the Comparator Match Detect signal will go Active.

Setting data for timer register TREG8H/L and TREG9H/L is executed using 2 byte data transfer instruction or using 1 byte date transfer instruction twice for lower 8 bits and upper 8 bits in order.

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

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

After a Reset, TREG8 and TREG9 are undefined. If the 16-bit timer is to be used after a Reset, data should be written to it beforehand.

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

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

The addresses of the Timer Registers are as follows:

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 registers and thus cannot be read.

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

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

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 registers and thus cannot be written to.

(5) Capture input control and external interrupt control

This circuit controls the timing to latch the value of up-counter UC8 into CAP8, CAP8 and the generation of external interrupts. The latch timing for the capture register and selection of edge for external interrupt is determined by TMOD8<CAP89M1,CAP89M0>.
The edge of external interrupt INT6 is fixed to rise edge.
In addition, the value in the up-counter UC8 can be loaded into a capture register by software. Whenever 0 is written to TMOD8<CAP8IN>, the current value in the up-counter UC8 is loaded into capture register CAP8. It is necessary to keep the prescaler in Run Mode (i.e. TRUN8<T8PRUN> must be held at a value of 1).

(6) Comparators (CP8 and CP9)

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

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

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

The values of TFF8 and TFF9 can be output via the Timer Output pins TO8 (which is shared with PD2) and TO9 (which is shared with PD3). Timer output should be specified using the Port D SFRs.

3.8.3 SFR

Timer 8 Run Register

TRUN8 (00A0H)

	7	6	5	4	3	2	1	0
Bit symbol	T8RDE	-	-	-	I2T8	T8PRUN	-	T8RUN
Read/Write	R/W	R/W			R/W	R/W		R/W
After Reset	0	0	-	-	0	0	-	0
Function	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 during IDLE2-mode
T8PRUN: Operation of prescaler
T8RUN: Operation of Timer 8

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

Timer A Run Register

TRUNA (00B0H)

	7	6	5	4	3	2	1	0
Bit symbol	TARDE	-	-	-	I2TA	TAPRUN	-	TARUN
Read/Write	R/W	R/W			R/W	R/W		R/W
After Reset	0	0	-	-	0	0	-	0
Function	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 during IDLE2-mode
TAPRUN: Operation of prescaler
TARUN: Operation of Timer A

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

Figure 3.8.3 Registers for 16-bit Timers

Timer 8 Mode Register

TMOD8 (00A2H)

	7	6	5	4	3	2	1	0
Bit symbol	CAP9T9	EQ9T9	CAP8IN	CAP89M1	CAP89M0	T8CLE	T8CLK1	T8CLK0
Read/Write	R/W		W	R/W				
After Reset	0	0	1	0	0	0	0	0
Function	TFF9 inversion 0: Disable trigger 1: Enable trigger 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 rising edge. 01: TI8 ↑ TI9 ↑ INT5 occurs on rising edge. 10: TI8 ↑ TI8 ↓ INT5 occurs on falling edge. 11: TFF1 ↑ TFF1 ↓ INT5 occurs on rising edge.		Control up-counter 0: Disable clearing 1: Enable clearing	Timer 8 source clock 00: TI8 pin 01: ϕT1 10: ϕT4 11: ϕT16	

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 3.8.4 Registers for 16-bit Timers

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 Invert when the UC value is captured to CAPB.	Invert when the UC value matches the value in TREGB.	Execute software capture 0: Execute 1: Don't care Note) Always read as 1.	Capture timing 00: Disable INT7 occurs on rising edge. 01: TIA ↑ TIB ↑ INT7 occurs on rising edge. 10: TIA ↑ TIA ↓ INT7 occurs on falling edge. 11: TFF1 ↑ TFF1 ↓ INT7 occurs on rising edge.		Control up-counter 0: Disable clearing 1: Enable clearing	Timer A source clock 00: TIA pin 01: ϕT1 10: ϕT4 11: ϕT16	

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	
10	CAPA at TIA rise CAPB at TIA fall	INT7 occurs on falling edge of TIA.
11	CAPA at TFF1 rise CAPB at TFF1 fall	INT7 occurs on rising edge of TIA.

Software capture

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

Figure 3.8.5 Registers for 16-bit Timers

Timer 8 Flip-Flop Control Register

<table>
<tr><td></td><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td rowspan="4">TFFCR8
(00A3H)</td><td>Bit symbol</td><td>TFF9C1</td><td>TFF9C0</td><td>CAP9T8</td><td>CAP8T8</td><td>EQ9T8</td><td>EQ8T8</td><td>TFF8C1</td><td>TFF8C0</td></tr>
<tr><td>Read/Write</td><td colspan="2">W</td><td colspan="4">R/W</td><td colspan="2">W</td></tr>
<tr><td>After Reset</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>Function</td><td colspan="2">Control TFF9
00: Invert
01: Set
10: Clear
11: Don’t care
Note)Always read as 11</td><td colspan="4">TFF8 inversion trigger
0: Disable trigger
1: Enable trigger

Invert when the UC value is loaded in to CAP9. | Invert when the UC value is loaded in to CAP8. | Invert when the UC value matches the value in TREG9. | Invert when the UC value matches the value in TREG8.</td><td colspan="2">Control TFF8
00: Invert
01: Set
10: Clear
11: Don’t care
Note)Always read as 11</td></tr>
</table>

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 3.8.6 Registers for 16-bit Timers

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 3.8.7 Registers for 16-bit Timers

Timer Register (Timer8, TimerA)

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

Capture Register (Timer8, TimerA)

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 3.8.8 Registers for 16-bit Timers.

3.8.4 Operation in each mode

(1) 16-Bit Interval Timer Mode

Generating interrupts at fixed intervals

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

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

Note: X = Don't care; "–" = No change

(2) 16-Bit Event Counter Mode

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

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

Note: X = Don't care; "–" = No change

When the timer is used as an event counter, set the prescaler in Run Mode (i.e. with TRUN8<T8PRUN> = 1).

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

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

The PPG mode is obtained by inversion of the timer flip-flop TFF8 that is to be enabled by the match of the up-counter UC8 with timer register TREG8 or TREG9 and to be output to TO8. In this mode the following conditions must be satisfied.

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

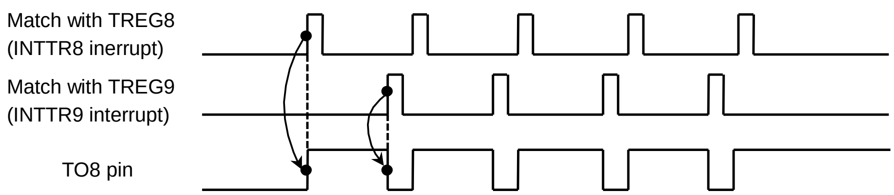


Figure 3.8.9 Programmable Pulse Generation (PPG) Output Waveforms

When the TREG8 double buffer is enabled in this mode, the value of Register Buffer 8 will be shifted into TREG8 at match with TREG9. This feature facilitates the handling of low-duty waves.

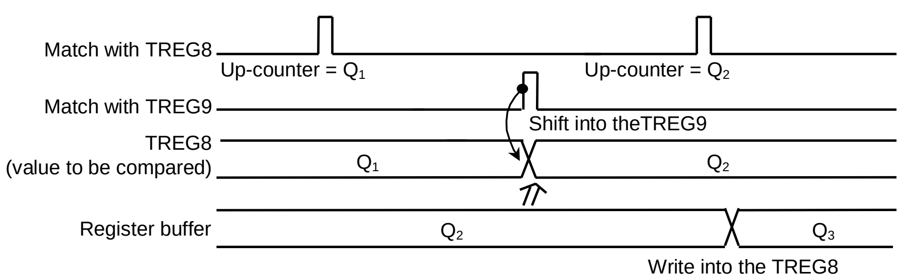


Figure 3.8.10 Operation of Register Buffer

The following block diagram illustrates this mode.

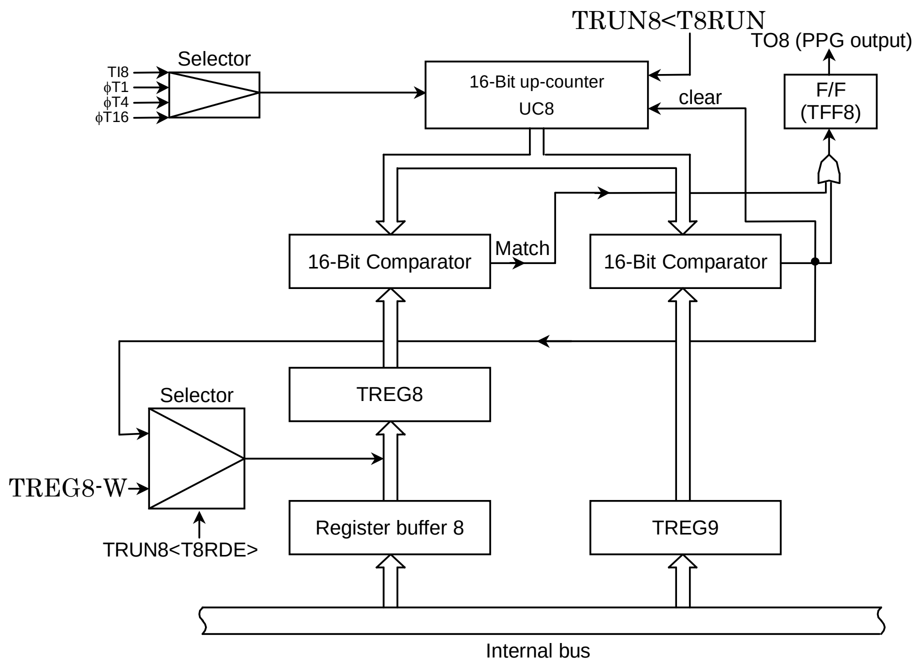


Figure 3.8.11 Block Diagram of 16-bit PPG Output Mode

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

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

Note: X = Don't care; "–" = No change

(4) Capture function

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

① As a one-shot pulse output from external trigger pulse

② For frequency measurement

③ For pulse width measurement

④ For time difference measurement

① One-shot pulse output from external trigger pulse

Set the up-counter UC8 to Free-Running Mode with the internal input clock, input an external trigger pulse via the TI8 pin, and load the value of the up-counter into the capture register CAP8 on the rising edge of the TI8 input signal.
When the interrupt INT5 is generated on the rising edge of the TI8 input, set the CAP8 value (c) plus a delay time (d) in TREG8 and set this value (c + d) plus the one-shot pulse width (p) in TREG9. (Thus TREG8 = c + d and TREG9 = c + d + p).
When the interrupt INT5 occurs, TFFCR8<EQ9T8,EQ8T8> should be set to 11 and that the TFF8 inversion is enabled only when the up-counter value matches TREG8 or TREG9. When an INTTR9 interrupt occurs, a one-shot pulse will be output and inversion will be disabled.
(c), (d) and (p) correspond to c, d and p in Figure 3.8.12.

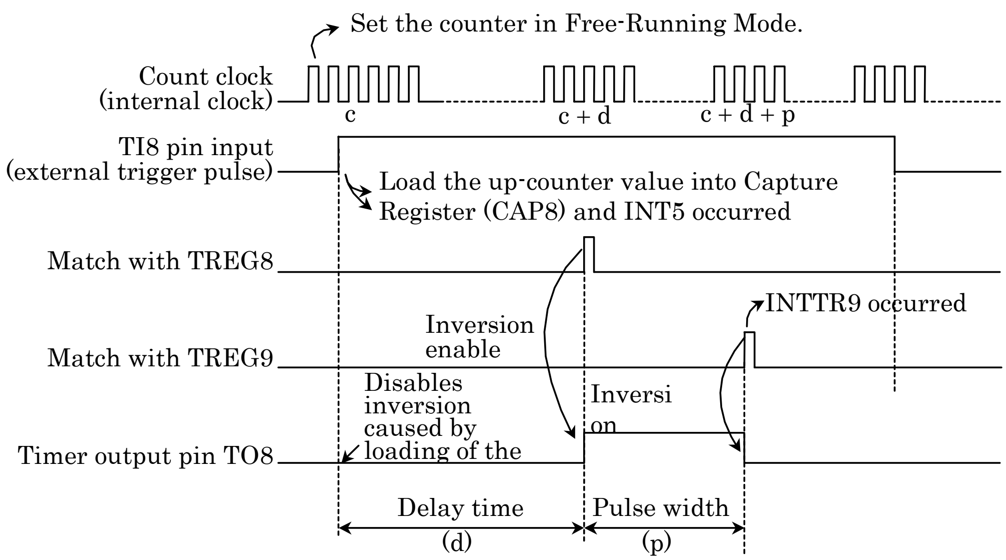


Figure 3.8.12 One-shot pulse output (with delay)

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

Main settings

```
                                            Keep counting (maintain free-running counter).
                                            Count using φT1.
TMOD8    ←  X  X  1  0  1  0  0  1
                                            Load the up-counter value into CAP8 on the rising edge
TFFCR8   ←  X  X  0  0  0  0  1  0          of the input to the TI8 pin.
                                            Clear TFF8 to zero.
                                            Disable TFF8 inversion.
PDCR     ←  -  -  -  -  -  1  -  -      }
PDFC     ←  -  -  -  -  -  1  -  -      }   Set PD2 to function as the TO8 pin.

INTE56   ←  X  -  -  -  X  1  0  0          Set INT5 Interrupt level to 4.
INTET89  ←  X  0  0  0  X  0  0  0          Disable INTTR8 and INTTR9.
TRUN8    ←  -  0  X  X  -  1  X  1          Start timer 8.
```

Setting of INT5

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

Setting INTTR9

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

Note: X = Don't care; "–" = No change

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

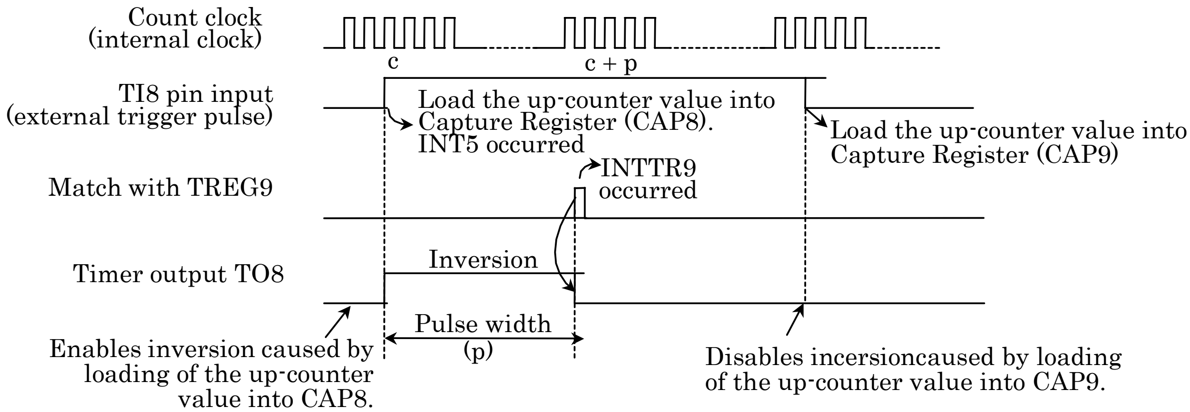


Figure 3.8.13 One-shot pulse output (without delay)

② Frequency measurement

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

The TI8 pin input should be selected as the clock input to timer 8. Set TMOD8<CAP89M1,CAP89M0> to 11. The value of the up-counter is loaded into the capture register CAP8 on the rising edge of the TFF1 signal from the timer flip-flop for the two 8-bit timers (timers 01), and loaded into CAP9 on the falling edge of the TFF1 signal.

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

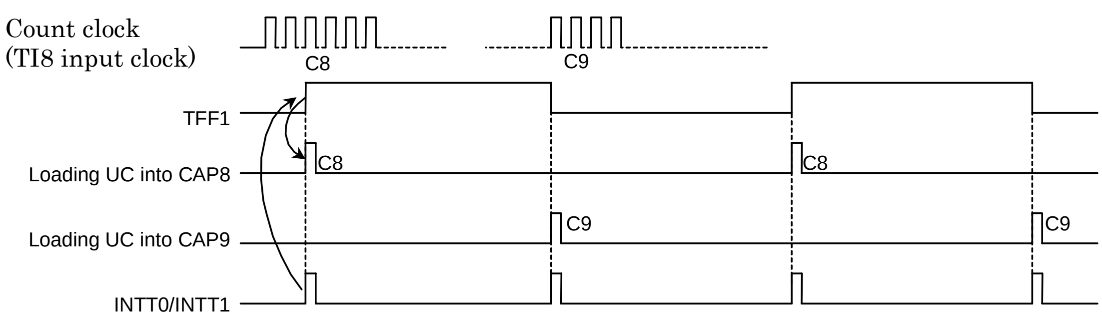


Figure 3.8.14 Frequency Measurement

For example, if the value for the level 1 width of TFF1 of the 8-bit timer is set to 0.5 s and the difference between the values in CAP8 and CAP9 is 100, the frequency is 100 ÷ 0.5 s = 200 Hz.

③ Pulse width measurement

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

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

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

In addition, the pulse width which is over the UC8 maximum count time specified by the clock source can be measured by changing software.

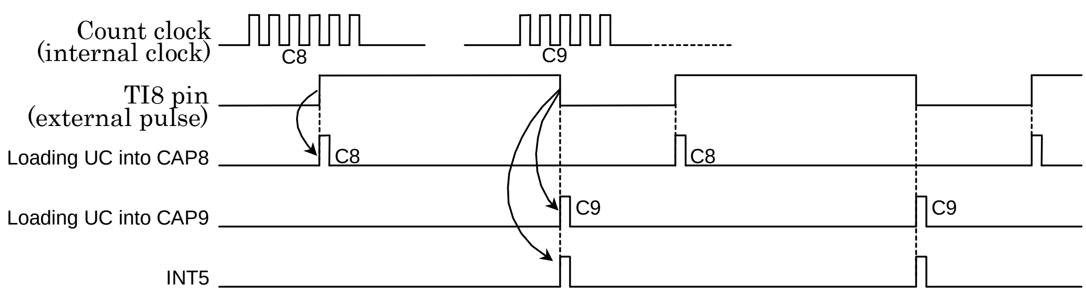


Figure 3.8.15 Pulse width measurement

Note: In Pulse Width Measuring Mode only (i.e. when TMOD8<CAP89M1,CAP89M0> = 10), the external interrupt INT5 occurs on the falling edge of the signal input to the TI8 pin. In other modes it occurs on the rising edge.

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

④ Time difference measurement

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

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

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

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

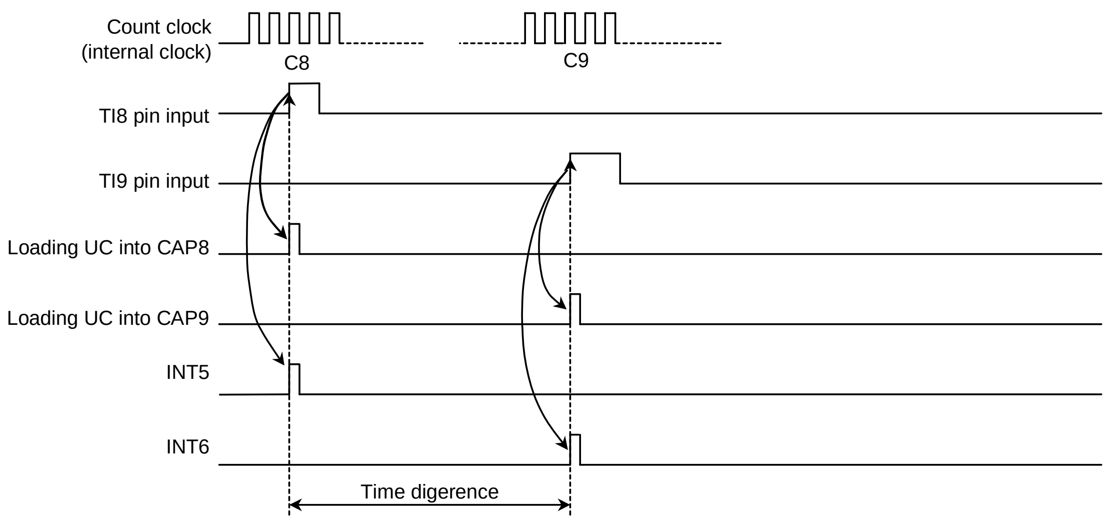


Figure 3.8.16 Time difference measurement

3.9 Serial Channels

TMP92CD54I includes two serial I/O channels. For both channels either UART Mode (asynchronous transmission) or I/O Interface Mode (synchronous transmission) can be selected.

- I/O Interface Mode — Mode 0: For transmitting and receiving I/O data using the synchronizing signal SCLK for extending I/O.
- UART Mode
  - Mode 1: 7-bit data
  - Mode 2: 8-bit data
  - Mode 3: 9-bit data

In Mode 1 and Mode 2, a parity bit can be added. Mode 3 has a wake-up function for making the master controller start slave controllers in a serial link (a multi-controller) system.
Figure 3.9.2 to Figure 3.9.3 are block diagrams for each channel.
Serial Channels 0 and 1 can be used independently.
Both channels operate in the same function except for the following points; thus only the operation of Channel 0 is explained below.

Table 3.9.1 Differences between Channels 0 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)

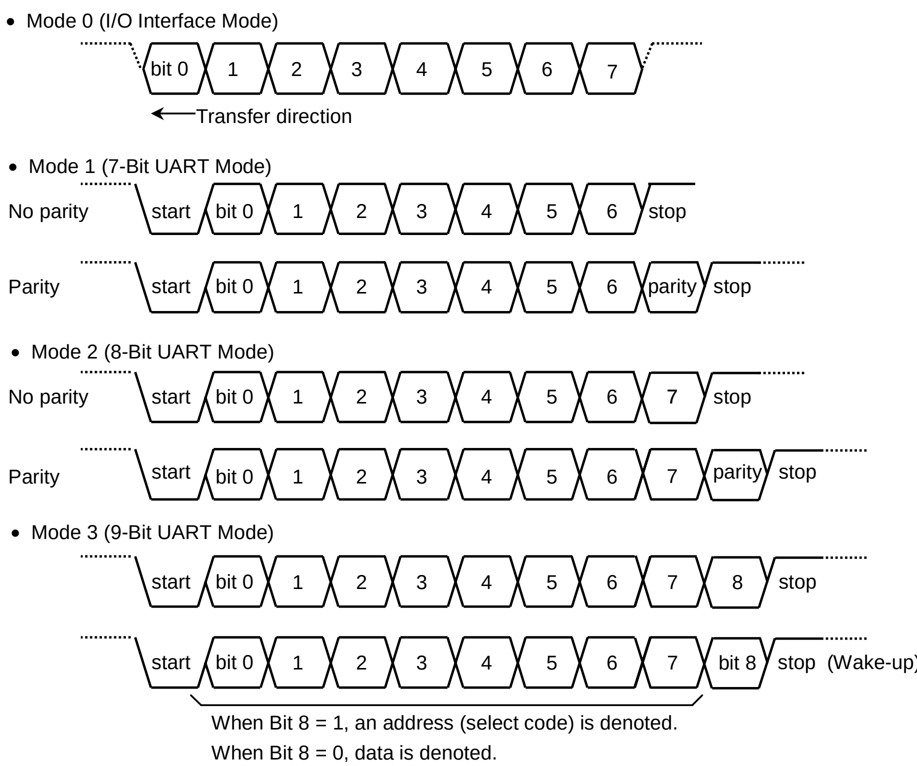


Figure 3.9.1 Data formats

3.9.1 Block diagrams

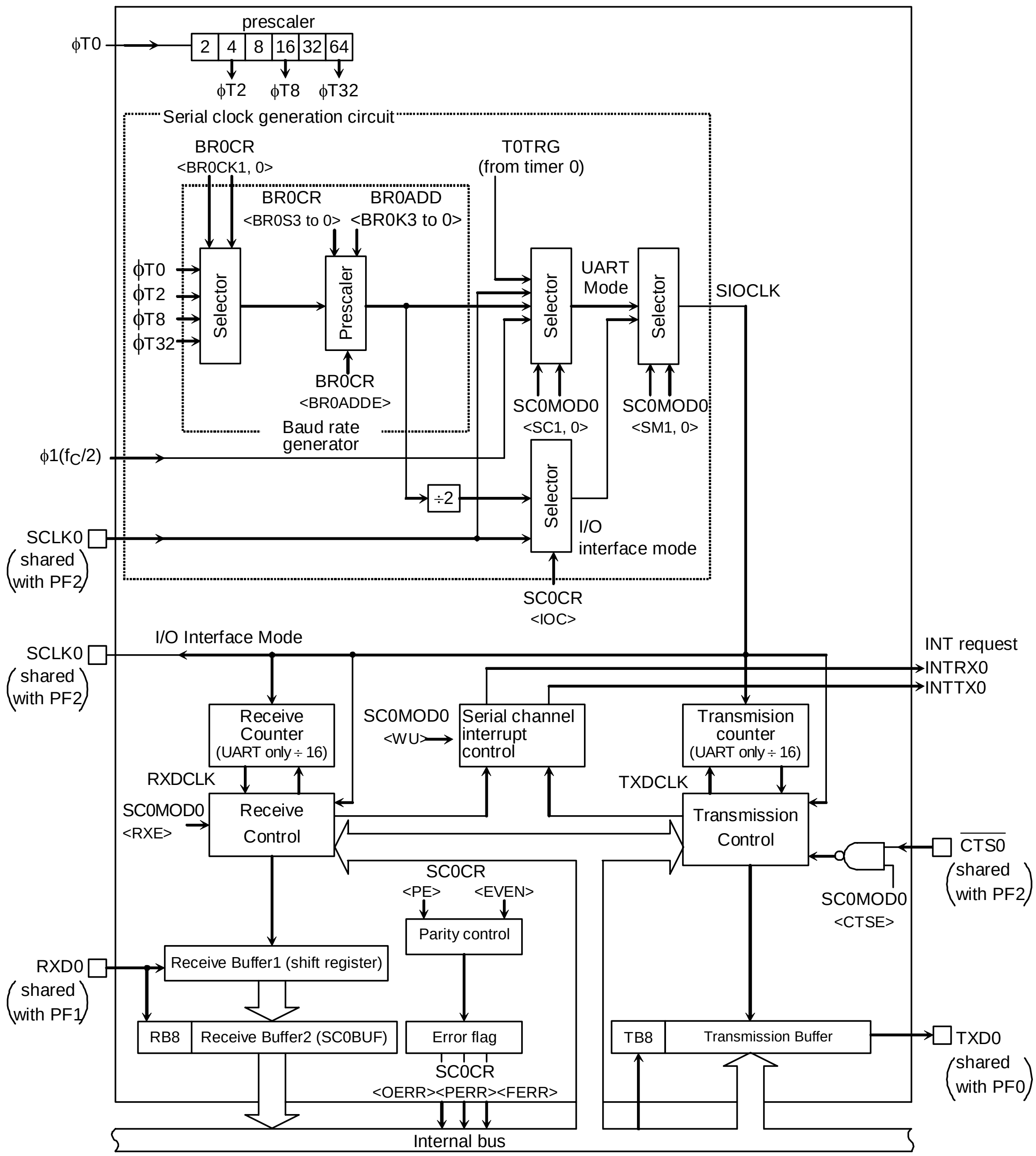


Figure 3.9.2 Block diagram of the Serial Channel 0

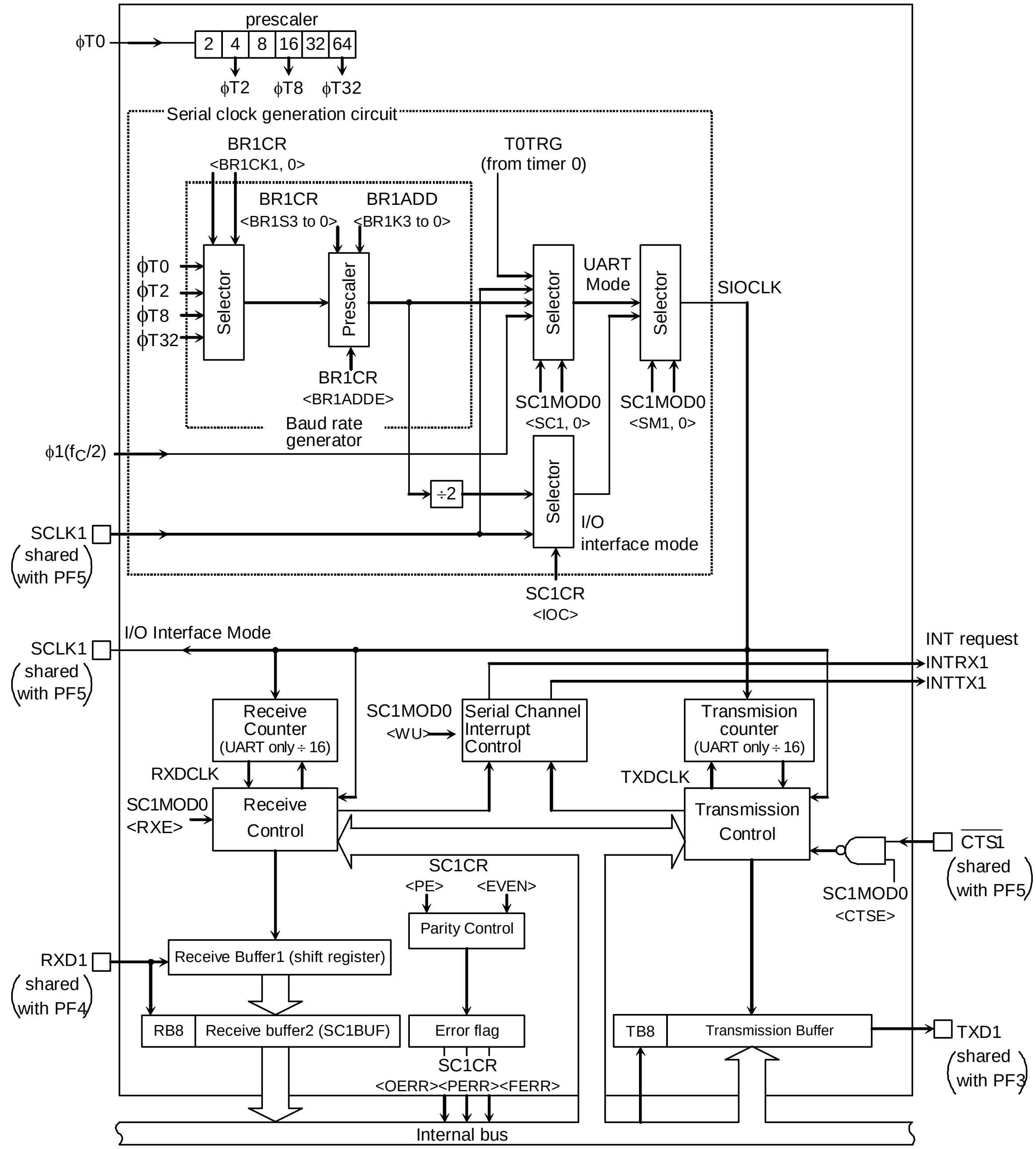


Figure 3.9.3 Block diagram of the Serial Channel 1

3.9.2 Operation for each circuit

(1) Prescaler, Prescaler clock select

There is a 6-bit prescaler for making serial clock.

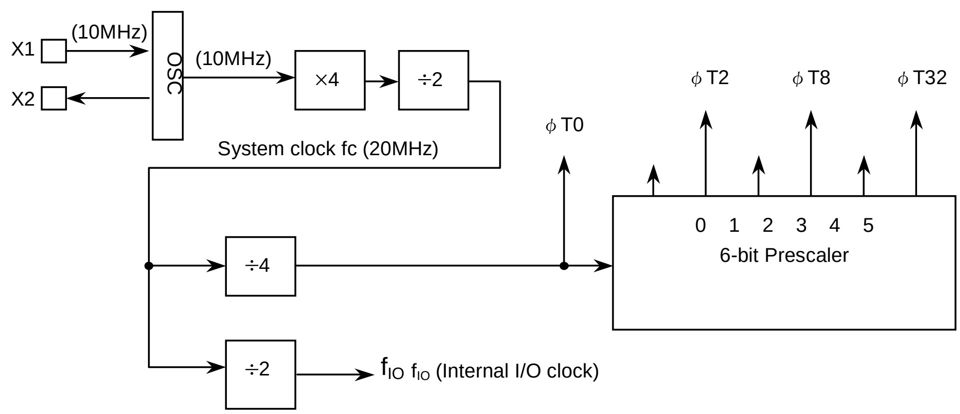
The prescaler can be run by selecting the baud rate generator as the making serial clock.
Table 3.9.2 shows prescaler clock resolution into the baud rate generator.

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



2006-01-27

(2) Baud rate generator

The baud rate generator is a circuit which generates transmission and receiving clocks which determine the transfer rate of the serial channels.
The input clock to the baud rate generator, ϕT0, ϕT2, ϕT8 or ϕT32, is generated by the 6-bit prescaler. One of these input clocks is selected using the BR0CR<BR0CK1 to BR0CK0> field in the Baud Rate Generator Control Register.
The baud rate generator includes a frequency divider, which divides the frequency by N (N=1 to 16) or by N + (16-K) / 16 (N=2 to 15 and K = 1 to 15). Note that the part (16-K)/16 can be disabled, resulting in a division of N.

A division of N+(16-K)/16 can be	[	2+1/16;	2+2/16;	...... ;	2+15/16;		
		3+1/16;	3+2/16;	...... ;	3+15/16;		
		: ;	: ;	: ;	: ;		
		: ;	: ;	: ;	: ;		
		15+1/16;	15+2/16;	...... ;	15+15/16;]		

A division of N can be	[	1;	2;	3;	...;	14;	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 by the settings of BR0CR<BR0ADDE, BR0S3 to BR0S0> and BR0ADD<BR0K3 to BR0K0>:

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

BR0CR<BR0S3 to BR0S0>: setting of the divided frequency
0000: N=16 (Unselectable when using the division N+(16-K)/16)
0001: N= 1
0010: N= 2
:
:
1111: N=15

BR0ADD<BR0K3 to BR0K0>: sets the frequency divisor "K" (when using the division N+(16-K)/16)
0000: Disabled
0001: K=1
:
:
1111: K=15

- In UART Mode

(1) When BR0CR<BR0ADDE> = 0

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

(2) When BR0CR<BR0ADDE> = 1

- The N + (16 – K) / 16 division function is enabled.
- N is set in BR0CR<BR0S3 to BR0S0> (N = 2, 3, 4, ⋯, 15)
- K is set in BR0ADD<BR0K3 to BR0K0> (K = 1, 2, 3,⋯, 15)

NOTE: At N=1 or N=16, the N+(16-K)/16 division function is disabled. Therefore set BR0CR<BR0ADDE> to "0".

- In I/O Interface Mode
  - The N + (16 - K) / 16 division function is not available in I/0 Interface Mode
  - Set BR0CR<BR0ADDE> to “0”
  - Therefore the settings BR0ADD<BR0K3 to BR0K0> are ignored
  - The baud rate generator divides the selected prescaler clock by N
  - N is set in BR0CR<BR0S3 to BR0S0> (N=1, 2, 3, …, 16)

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

- In UART Mode

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

- In I/O Interface Mode

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

- Integer divider (N divider)

For example, when the source clock frequency (fc) is 19.6608 MHz, the input clock is ϕ T2 (fc/16), the frequency divider N (BR0CR<BR0S3 to BR0S0>) = 8, and BR0CR<BR0ADDE> = 0, the baud rate in UART Mode is as follows:

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

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

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

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

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

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

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

Table 3.9.3 to 4 show examples of UART Mode transfer rates.

Additionally, the external clock input is available in the serial clock. (Serial Channels 0 & 1). The method for calculating the baud rate is explained below:

- In UART Mode

  Baud rate = external clock input frequency ÷ 16

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

- In I/O Interface Mode

  Baud rate = external clock input frequency

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

Table 3.9.3 Selection of Transfer Rate(1)

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

Unit (kbps)

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

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

Table 3.9.4 Selection of Transfer Rate(2)

(When timer 0 with input Clock ϕT1 is used)

Unit (kbps)

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

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

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

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

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

(3) Serial clock generation circuit

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

- In I/O Interface Mode

In SCLK Output Mode with the setting SC0CR<IOC> = 0, the basic clock is generated by dividing the output of the baud rate generator by 2, as described previously.
In SCLK Input Mode with the setting SC0CR<IOC> = 1, the rising edge or falling edge will be detected according to the setting of the SC0CR<SCLKS> register to generate the basic clock.

- In UART Mode

The SC0MOD0 <SC1, SC0> setting determines whether the baud rate generator clock, the internal clock $\phi 1$ (fc/2), the match detect signal from timer 0 or the external clock (SCLK0) is used to generate the basic clock SIOCLK.

(4) Receiving counter

The receiving counter is a 4-bit binary counter used in UART Mode which counts up the pulses of the SIOCLK clock. It takes 16 SIOCLK pulses to receive 1 bit of data; each data bit is sampled three times – on the 7th, 8th and 9th clock cycles.
The value of the data bit is determined from these three samples using the majority rule.
For example, if the data bit is sampled respectively as 1, 0 and 1 on 7th, 8th and 9th clock cycles, the received data bit is taken to be 1. A data bit sampled as 0, 0 and 1 is taken to be 0.

(5) Receiving control

- In I/O Interface Mode

In SCLK Output Mode with the setting SC0CR<IOC> = 0, the RXD0 signal is sampled on the rising edge of the shift clock which is output on the SCLK0 pin.
In SCLK Input Mode with the setting SC0CR<IOC> = 1, the RXD0 signal is sampled on the rising or falling edge of the SCLK0 input, according to the SC0CR<SCLKS> setting.

- In UART Mode

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

(6) The Receiving Buffers

To prevent Overrun errors, the Receiving Buffers are arranged in a double-buffer structure.
Received data is stored one bit at a time in Receiving Buffer 1 (which is a shift register). When 7 or 8 bits of data have been stored in Receiving Buffer 1, the stored data is transferred to Receiving Buffer 2 (SC0BUF); this causes an INTRX0 interrupt to be generated. The CPU only reads Receiving Buffer 2 (SC0BUF). Even before the CPU has finished reading the contents of Receiving Buffer 2 (SC0BUF), more data can be received and stored in Receiving Buffer 1. However, if Receiving Buffer 2 (SC0BUF) has not been read completely before all the bits of the next data item are received by Receiving Buffer 1, an Overrun error occurs. If an Overrun error occurs, the contents of Receiving Buffer 1 will be lost, although the contents of Receiving Buffer 2 and SC0CR<RB8> will be preserved.
SC0CR<RB8> is used to store either the parity bit – added in 8-Bit UART Mode – or the most significant bit (MSB) – in 9-Bit UART Mode.
In 9-Bit UART Mode the wake-up function for the slave controller is enabled by setting SC0MOD0<WU> to 1; in this mode INTRX0 interrupts occur only when the value of SC0CR<RB8> is 1.

(7) Transmission counter

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

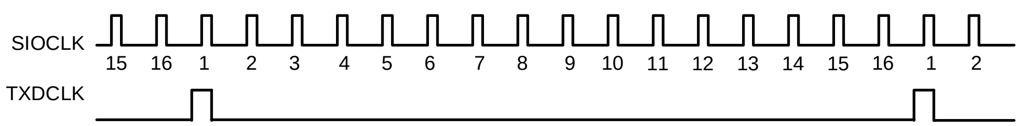


Figure 3.9.5 Generation of the transmission clock

(8) Transmission controller

- In I/O Interface Mode

In SCLK Output Mode with the setting SC0CR<IOC> = 0, the data in the Transmission Buffer is output one bit at a time to the TXD0 pin on the rising edge of the shift clock which is output on the SCLK0 pin.
In SCLK Input Mode with the setting SC0CR<IOC> = 1, the data in the Transmission Buffer is output one bit at a time to the TXD0 pin on the rising or falling edge of the SCLK0 input, according to the SC0CR<SCLKS> setting.

- In UART Mode

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

Handshake function

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

When the $\overline{\text{CTS0}}$ pin foes High on completion of the current data send, data transmission is halted until the $\overline{\text{CTS0}}$ pin foes Low again. However, the INTTX0 Interrupt is generated, it requests the next data send to the CPU. The next data is written in the Transmission Buffer and data sending is halted.

Although there is no RTS pin, a handshake function can easily be configured by assigning any port to perform the RTS function. The RTS should be output High to request send data halt after data receive is completed by software in the RXD interrupt routine.

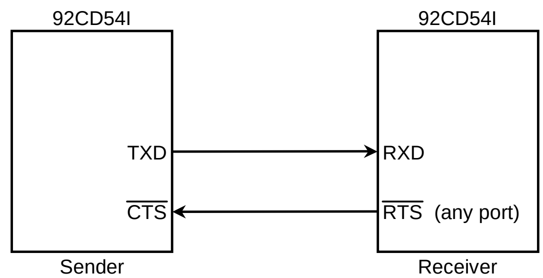
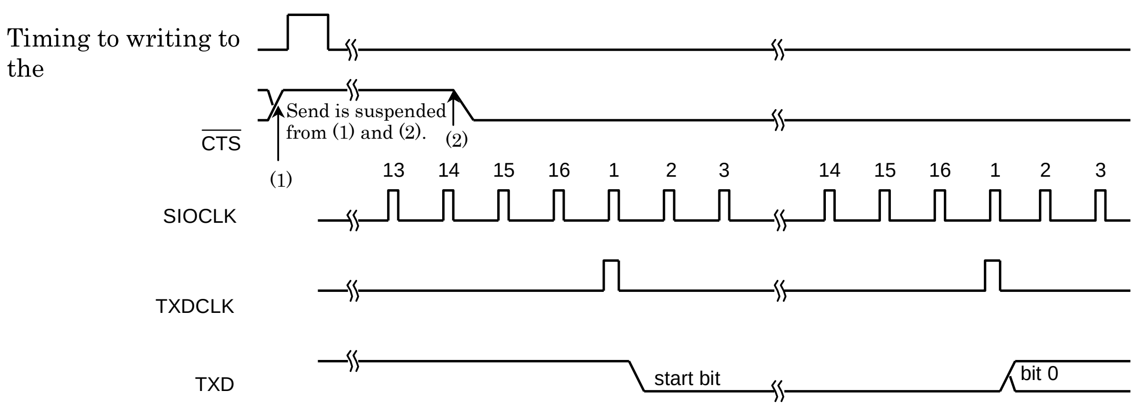


Figure 3.9.6 Handshake function



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

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

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

(9) Transmission Buffer

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

(10) Parity control circuit

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

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

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

(11) Error flags

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

1. Overrun error <OERR>

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

   The below is a processing example of when Overrun error is occurred.

   (INTRX routine)

   1)Read Received-Buffer
   2)Read error-flag
   3)if<OERR>=1
     then
       4)Disable receiving(write 0 to <RXE>)
       5)Wait for terminating current frame
       6)Read received-buffer
       7)Readerror-flag
       8)Enable receiving(write 1 to <RXE>)
       9)Request to resend
   10)Process other job

2. Parity error <PERR>

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

3. Framing error <FERR>

   The stop bit for the received data is sampled three times around the center. If the majority of the samples are 0, a Framing error is generated.

(12) Timing generation

① In UART Mode

Receiving

Mode	9-Bit (Note)	8-Bit + Parity (Note)	8-Bit, 7-Bit + Parity, 7-Bit
Interrupt timing	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 and 8-Bit + Parity Mode, interrupts coincide with the ninth bit pulse. Thus, when servicing the interrupt, it is necessary to allow a 1-bit period to elapse (so that the stop bit can be transferred) in order to allow proper framing error checking.

Transmitting

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

② I/O interface

Transmission Interrupt timing	SCLK Output Mode	Immediately after rise of last SCLK signal. (See figure 3.9 20.)
	SCLK Input Mode	Immediately after rise of last SCLK signal Rising Mode, or immediately after fall in Falling Mode. (See figure 3.9 21.)
Receiving Interrupt timing	SCLK Output Mode	Timing used to transfer received 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

SC0MOD0 (00C2H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After Reset	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)	

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

Serial Transmission Mode

00	I/O Interface Mode	
01	UART	7-bit 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 3.9.8 Serial Mode Control Register (channel 0, SC0MOD0)

SC1MOD0 (00CAH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After Reset	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)	

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

Serial Transmission Mode

00	I/O Interface Mode	
01	UART	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)

SC0CR (00C1H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R(cleared to 0 when read)			R/W	
After Reset	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 Overrun	Parity	Framing	0: SCLK0 1: SCLK0	0: baud rate generator 1: SCLK0 pin input

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
Overrun Error flag
cleared to 0 when read

Parity addition enable

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data 8

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

Figure 3.9.10 Serial Control Register (channel 0, SC0CR)

SC1CR (00C9H)

	7	6	5	4	3	2	1	0
bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (cleared to 0 when)			R/W	
After Reset	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 Overrun	Parity	Framing	0: SCLK1 1: SCLK1	0: baud rate generator 1: SCLK1 pin input

I/O interface input clock select

0	Baud rate generator
1	SCLK1 pin input

Edge selection for SCLK1 pin (input mode only)

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

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

Parity addition enable

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data bit 8

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

Figure 3.9.11 Serial Control Register (channel 1, SC1CR)

BR0CR (00C3H)

	7	6	5	4	3	2	1	0
bit Symbol	–	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	(Note) Always fixed to "0"	+(16–K)/16 division 0: Disable 1: Enable	00 : ϕT0 01 : ϕT2 10 : ϕT8 11 : ϕT32		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

	BR0CR<BR0ADDE> = 1 (UART only)		BR0CR<BR0ADDE> = 0 (UART and I/O interface modes)
BR0CR <BR0S3:0> / BR0ADD <BR0K3:0>	0000(N = 16) or 0001(N = 1)	0010(N = 2) to 1111(N = 15)	0001(N = 1) (UART Only) to 1111(N = 15) 0000(N = 16)
0000	Disable *	Disable *	
0001(K = 1) to 1111(K = 15)	Disable *	Divided by N + (16 – K) / 16	Divided by N

*: as the N+(16-K)/16 division function is disabled in UART mode, set BR0ADDE to "0"

Division by N with N=[1;2;3;…;16]
Division by N+(16-K)/16 = [2+1/16 ; 2+2/16 ; … ; 2+15/16 ; 3 ; 3+1/16 ; … ; 15+15/16]
Division by [1 ; 2 ; 2+1/16 ; 2+2/16 ; … ; 2+15/16 ; 3 ; 3+1/16 ; … ; 15+15/16; 16]

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

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

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<BR1ADDE> = 1 (UART only)		BR1CR<BR1ADDE> = 0 (UART and I/O interface modes)
BR1CR <BR1S3:0> / BR1ADD <BR1K3:0>	0000(N = 16) or 0001(N = 1)	0010(N = 2) to 1111(N = 15)	0001(N = 1) (UART Only) to 1111(N = 15) 0000(N = 16)
0000	Disable *	Disable *	
0001(K = 1) to 1111(K = 15)	Disable *	Divided by N + (16 − K) / 16	Divided by N

*: as the N+(16-K)/16 division function is disabled in UART mode, set BR1ADDE to “0”

Division by N with N=[1;2;3;…;16]
Division by N+(16-K)/16 = [2+1/16 ; 2+2/16 ; … ; 2+15/16 ; 3 ; 3+1/16 ; … ; 15+15/16]
Division by [1 ; 2 ; 2+1/16 ; 2+2/16 ; … ; 2+15/16 ; 3 ; 3+1/16 ; … ; 15+15/16]; 16

Note 1: Set BR1CR <BR1ADDE> to “1” after setting K (K = 1 to 15) to BR1ADD <BR1K3 to 0> when + (16 − K) / 16 division function is used.

Note 2: + (16 − K) / 16 division functions is possible to use in only UART mode.
Set BR1CR <BR1ADDE> to “0” to disable + (16 − K) / 16 division in I/O interface mode.

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

SC0BUF (00C0H)

	7	6	5	4	3	2	1	0	
SC0BUF (00C0H)	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Reveiving)

Note: Prohibit Read modify write for SC0BUF.

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	
SC1BUF (00C8H)	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receiving)

Note: Prohibit Read modify write for SC1BUF.

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 for each mode

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

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

This mode includes the SCLK output mode to output synchronous clock SCLK and SCLK input external synchronous clock SCLK.

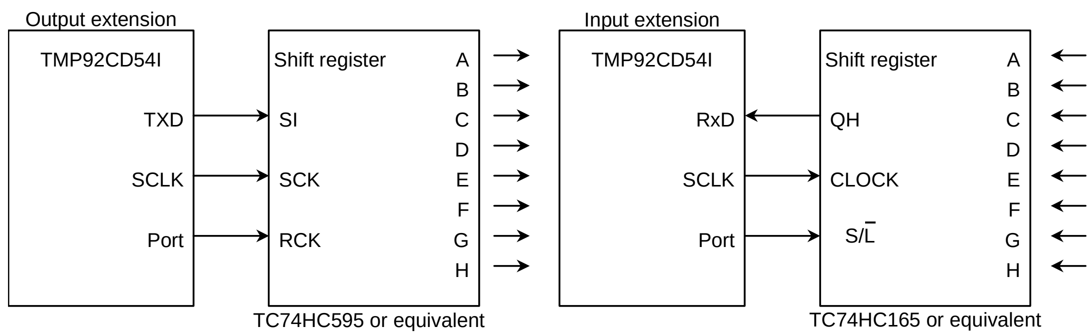


Figure 3.9.18 Example of SCLK Output Mode Connection

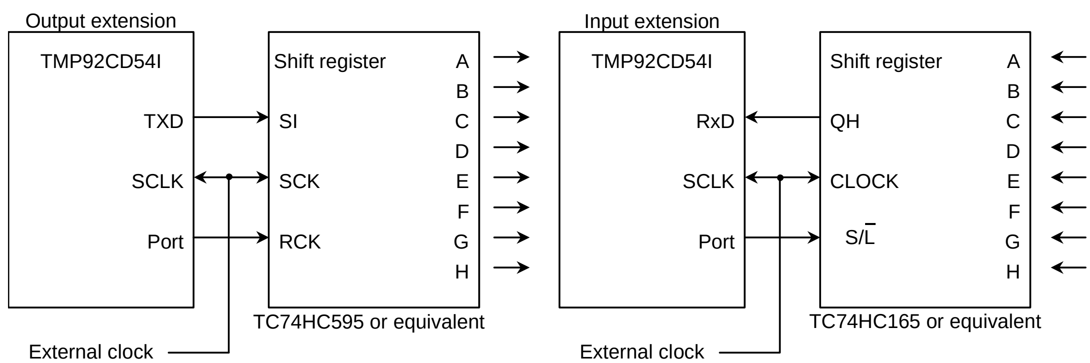


Figure 3.9.19 Example of SCLK Input Mode Connection

① Transmission

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

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

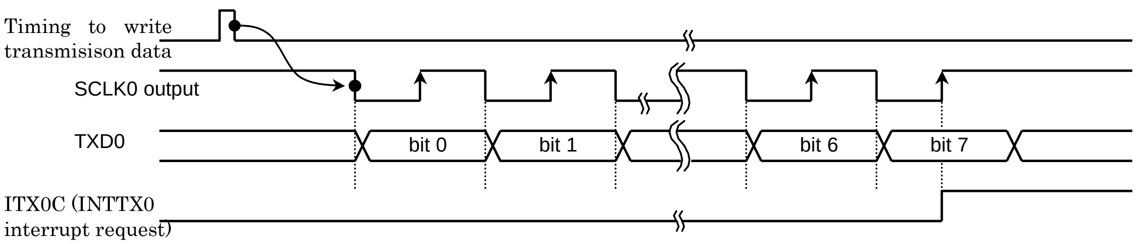


Figure 3.9.20 Transmitting Operation in I/O Interface Mode (SCLK0 Output Mode)

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

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

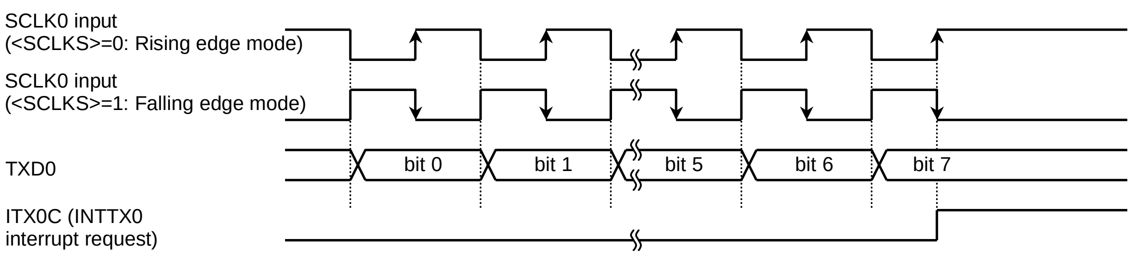


Figure 3.9.21 Transmitting Operation in I/O Interface Mode (SCLK0 Input Mode)

② Receiving

In SCLK Output Mode the synchronous clock is output on the SCLK0 pin and the data is shifted to Receiving Buffer 1. This is initiated when the Receive Interrupt flag INTES0<IRX0C> is cleared as the received data is read. When 8-bit data is received, the data is transferred to Receiving Buffer 2 (SC0BUF) following the timing shown below and INTES0<IRX0C> is set to 1 again, causing an INTRX0 interrupt to be generated.
Setting SC0MOD0<RXE> to 1 initiates SCLK0 output.

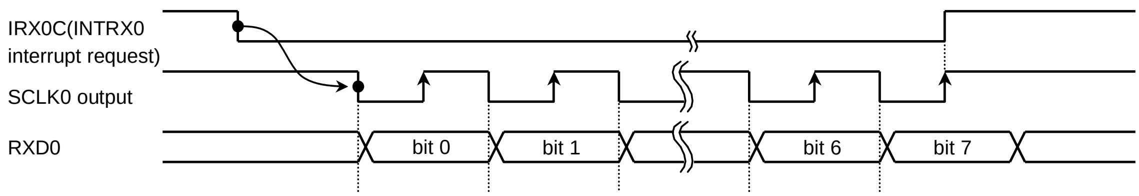


Figure 3.9.22 Receiving operation in I/O Interface Mode (SCLK0 Output Mode)

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

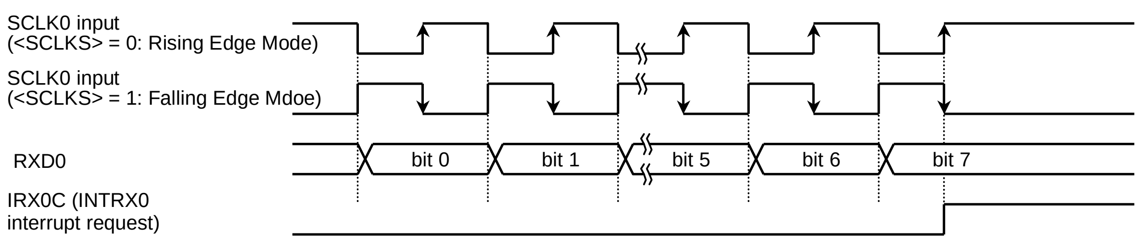


Figure 3.9.23 Receiving Operation in I/O interface Mode (SCLK0 Input Mode)

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

③ Transmission and Receiving (Full Duplex Mode)

When Full Duplex Mode is used, set the Receive Interrupt Level to 0 and set enable the level of transmit interrupt. Ensure that the program which transmits the interrupt reads the receiving buffer before setting the next transmit data.
The following is an example of this:

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

Main routine

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

INTTX0 interrupt routine

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

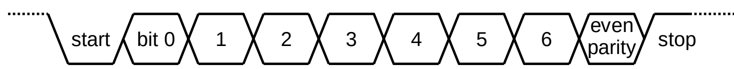
Note: X = Don't care; "–" = No change

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

7-Bit UART Mode is selected by setting the Serial Channel Mode Register SC0MOD0<SM1,SM0> field to 01.

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

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



Transmission direction (transmission rate: 2400 bps at fc = 19.6608 MHz)

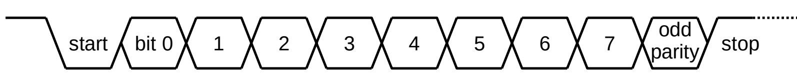
```
           7 6 5 4 3 2 1 0
PFCR    ← − − − − − − − 1   } Set PF0 to function as the TXD0 pin.
PFFC    ← − − − − − − − 1   }
SC0MOD0 ← X 0 − X 0 1 0 1     Select 7-Bit UART Mode.
SC0CR   ← X 1 1 X X X 0 0     Add even parity.
BR0CR   ← 0 0 1 0 1 0 0 0     Set the transfer rate to 2400 bps.
INTES0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

Note: X = Don't care; "–" = No change

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

8-Bit UART Mode is selected by setting SC0MOD0<SM1,SM0> to 10. In this mode a parity bit can be added (use of a parity bit is enabled or disabled by the setting of SC0CR<PE>); whether even parity or odd parity will be used is determined by the SC0CR<EVEN> setting when SC0CR<PE> is set to 1 (enabled).

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



Transmission direction (transmission rate: 9600 bps at fc = 19.6608 MHz)

Main settings

	7 6 5 4 3 2 1 0	
PFCR	← – – – – – – 0 –	Set PF1 to function as the RXD0 pin.
SC0MOD0	← – 0 1 X 1 0 0 1	Enable receiving in 8-Bit UART Mode.
SC0CR	← X 0 1 X X X 0 0	Add odd parity.
BR0CR	← 0 0 0 1 1 0 0 0	Set the transfer rate to 9600 bps.
INTES0	← – – – – X 1 0 0	Enable the INTTX0 interrupt and set it to Interrupt Level 4.

Interrupt processing

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

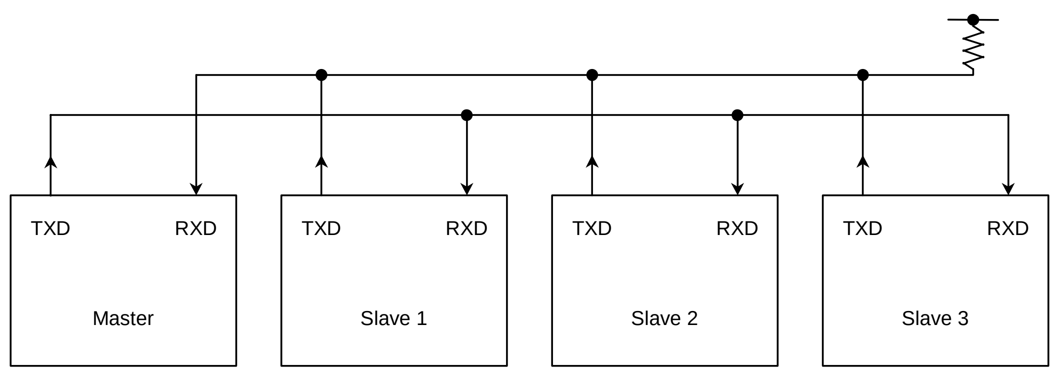
Note: X = Don't care; "–" = No change

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

9-Bit UART Mode is selected by setting SC0MOD0<SM1,SM0> to 11. In this mode parity bit cannot be added.
In the case of transmission the MSB (9th bit) is written to SC0MOD0<TB8>. In the case of receiving it is stored in SC0CR<RB8>. When the buffer is written and read, the MSB is read or written first, before the rest of the SC0BUF data.

Wake-up function

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



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

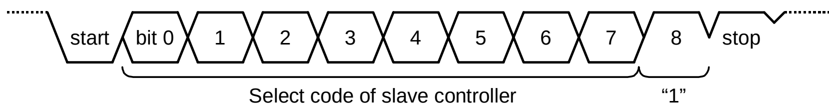
Figure 3.9.24 Serial Link using Wake-up function

Protocol

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

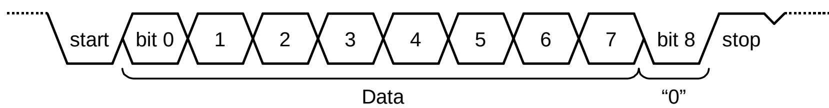
② Set the SC0MOD0<WU> bit on each slave controller to 1 to enable data receiving.

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



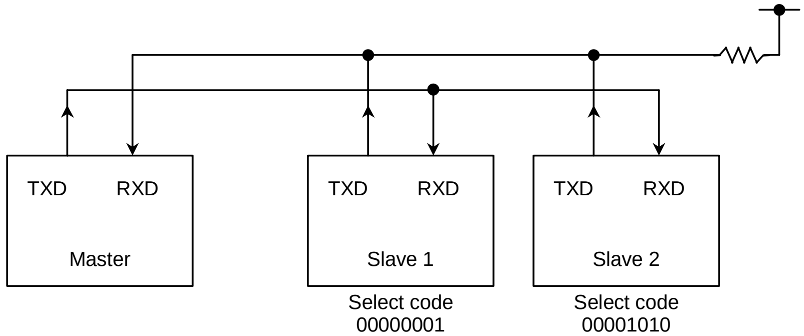
④ Each slave controller receives the above frame. Each controller checks the above select code against its own select code. The controller whose code matches clears its <WU> bit to 0.

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



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

Setting example: To link two slave controllers serially with the master controller using the internal clock ϕ 1 as the transfer clock.



- Setting the master controller

Main

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

INTTX0 interrupt

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

- Setting the slave controller

Main

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

INTRX0 interrupt

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

3.10 Serial Bus Interface (SBI)

TMP92CD54I has 3-channels serial bus interface which employs a clocked-synchronous 8-bit SIO mode and an I²C bus mode. It is called SBI0, SBI1 and SBI2.

	I²C 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)

Since each channel carries out the same operation, it explains only SBI0.

The serial bus interface is connected to an external device through PN1 (SDA0) and PN2 (SCL0) in the I²C bus mode; and through PN0 (SCK0), PN1 (SO0) and PN2 (SI0) in the clocked-synchronous 8-bit SIO mode.

Each pin is specified as follows.

	PNODE <ODEN2, ODEN1>	PNCR <PN2C, PN1C, PN0C>	PNFC <PN2F, PN1F, PN0F>
I²C Bus Mode	11	11X	11X
Clocked Synchronous 8-Bit SIO Mode	XX	011 010	011

X: Don't care

3.10.1 Configuration

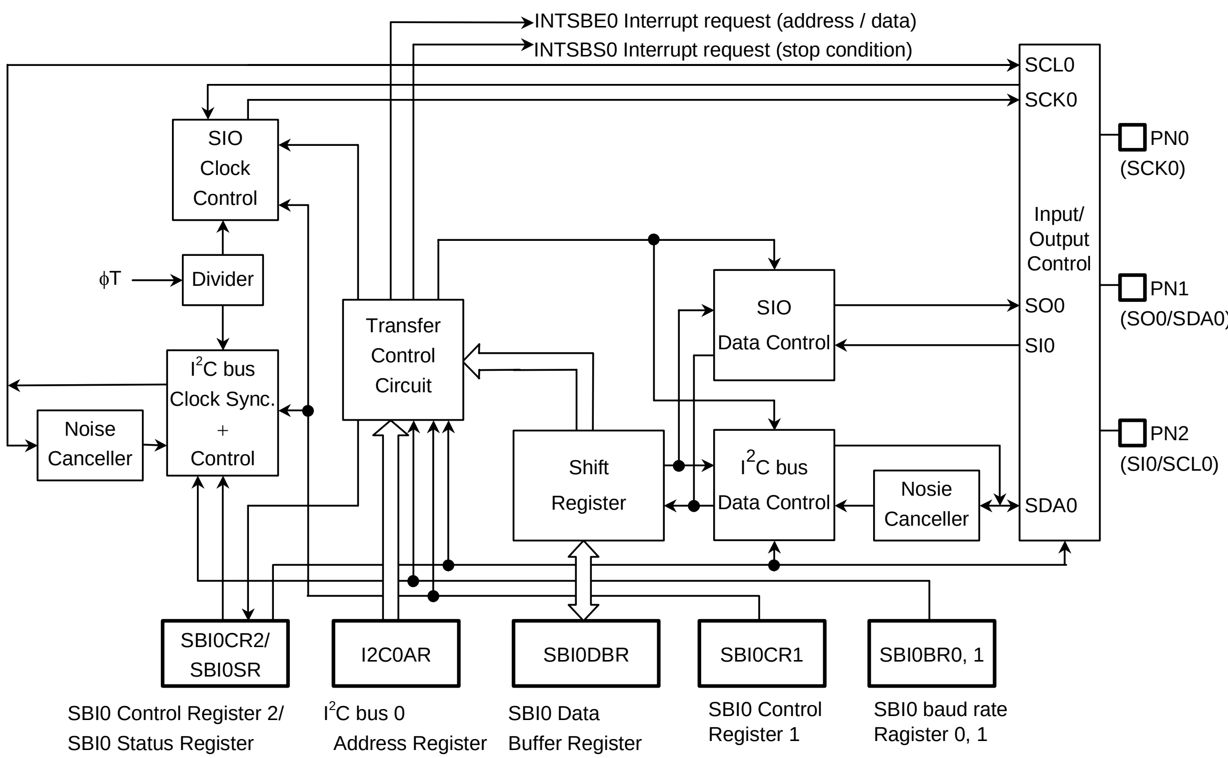


Figure 3.10.1 Serial Bus Interface 0 (SBI0)

3.10.2 Serial Bus Interface (SBI) Control

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

- Serial bus interface 0 control register 1 (SBI0CR1)
- Serial bus interface 0 control register 2 (SBI0CR2)
- Serial bus interface 0 data buffer register (SBI0DBR)
- I[2]C bus 0 address register (I2C0AR)
- Serial bus interface 0 status register (SBI0SR)
- Serial bus interface 0 baud rate register 0 (SBI0BR0)
- Serial bus interface 0 baud rate register 1 (SBI0BR1)

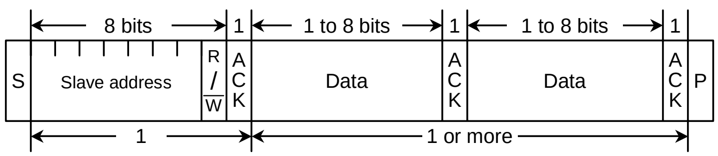
The above registers differ depending on a mode to be used.

Refer to Section “3.10.4 I2C bus Mode Control” and “3.10.7 Clocked-synchronous 8-bit SIO Mode Control”.

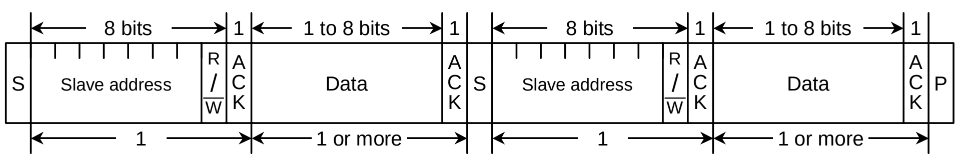
3.10.3 The Data Formats in the I^{2}C Bus Mode

The data formats in the I^{2}C bus mode are shown below.

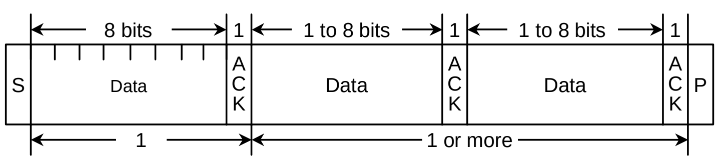
(a) Addressing format



(b) Addressing format (with restart)



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



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

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

3.10.4 I²C Bus Mode Control Register

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

Seirial Bus Interface 0 Conrol Register 1

SBI0CR1 (0170H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
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)
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 ($fscl = \frac{fc}{2^n + 8}$ [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: Set the <BC2 to 0> to “000” before switching to a clock-synchronous 8-bit SIO mode.
Note 2: For the frequency of the SCL line clock, see 3.10.5 (3) Serial clock.
Note 3: Initial data of SCK0 is “0”, SWRMON is “1”.

Figure 3.10.3 Registers for the I²C Bus Mode

Serial Bus Interface 0 Control Register 2

SBI0CR2 (0173H)		7	6	5	4	3	2	1	0
	Bit Symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
	Read/Write	W				W (Note 1)		W (Note 1)	
	After Reset	0	0	0	1	0	0	0	0
Prohibit Read-modify-write	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^2C 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^2C 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: Reading this register function as SBI0SR register.
Note2: Switch a mode to port mode after confirming that the bus is free.
Switch a mode between I^2C bus mode and clock-synchronous 8-bit SIO mode after confirming that input signals via port are high-level.

Figure 3.10.4 Registers for the I^2C 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
Prohibit Read-modify-write	Function	Master/ Slave status monitor	Transmitter/ Receiver status monitor	I^2C bus status monitor	INTSBE0 interrupt request monitor	Arbitration lost detection monitor 0: – 1: Detected	Slave address match detection monitor 0: 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

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: Writing in this register functions as SBI0CR2.
Note2: If SBI0SR<BB> drops down from 1 to 0 (falling edge), INTSBS0 will be generated in both case of Master mode and Slave mode.

Figure 3.10.5 Registers for the I^2C Bus Mode

Serial Bus Interface 0 Baud Rate Register 0

SBI0BR0 (0174H)

Prohibit Read-modify-write

	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

Sirial Bus Interface 0 Data Buffer Register

SBI0DBR (0171H)

Prohibit Read-modify-write

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

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

I^2C Bus 0 Address Register

I2C0AR (0172H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
After Reset	0	0	0	0	0	0	0	0
Function	Slave address selection for when device is operating as slave device							Addressing or free data format

Address recognition mode specification

0	Addressing format
1	Free data format

Addressing or free data format impact both slave and master configuration.

When addressing format is used (<ALS>=0), TRX bit is updated relying on R/W bit (=8th bit of first received byte after start condition). Moreover in slave mode, MCU spies the bus after start condition to recognize its address.

When free data format is used (<ALS>=1) all words on the bus are considered as data words, that means no address recognition is done and TRX is not updated.

Figure 3.10.6 Registers for the I^2C Bus Mode

Seirial Bus Interface 1 Conrol Register 1

SBI1CR1 (0178H)		7	6	5	4	3	2	1	0
	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)
Prohibit Read-modify-write	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 $(fscl = \frac{fc}{2^n + 8}$ [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: Set the <BC2 to 0> to “000” before switching to a clock-synchronous 8-bit SIO mode.
Note 2: For the frequency of the SCL line clock, see 3.10.5 (3) Serial clock.
Note 3: Initial data of SCK0 is “0”, SWRMON is “1”.

Figure 3.10.7 Registers for the I^2C Bus Mode

Serial Bus Interface 1 Control Register 2

SBI1CR2 (017BH)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
Read/Write	W				W (Note 1)		W (Note 1)	
After Reset	0	0	0	1	0	0	0	0
Function	Master/Slave selection	Transmitter/ Receiver selection	Start/Stop generation	Cancel INTSBE1 interrupt request	Serial bus interface operating mode selection (note 2) 00: Port Mode 01: SIO Mode 10: I^2C 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^2C 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: Reading this register function as SBI1SR register.

Note2: Switch a mode to port mode after confirming that the bus is free.
Switch a mode between I^2C bus mode and clock-synchronous 8-bit SIO mode after confirming that input signals via port are high-level.

Figure 3.10.8 Registers for the I^2C Bus Mode

Serial Bus Interface 1 Status Register

SBI1SR (017BH)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Function	Master/ Slave status monitor	Transmitter/ Receiver status monitor	I^2C bus status monitor	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: Writing in this register functions as SBI1CR2.
Note2: If SBI1SR<BB> drops down from 1 to 0 (falling edge), INTSBS1 will be generated in both case of Master mode and Slave mode.

Figure 3.10.9 Registers for the I^2C Bus Mode

Serial Bus Interface 1 Baud Rate Regster 0

SBI1BR0 (017CH) Prohibit Read-modify-write

	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 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

Sirial Bus Interface 1 Data Buffer Register

SBI1DBR (0179H) Prohibit Read-modify-write

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

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

I^2C Bus 1 Address Register

I2C1AR (017AH) Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
After Reset	0	0	0	0	0	0	0	0
Function	Slave address selection for when device is operating as slave device							Addressing or free data format

Address recognition mode specification

0	Addressing format
1	Free data format

Addressing or free data format impact both slave and master configuration.

When addressing format is used (<ALS>=0), TRX bit is updated relying on R/W bit (=8th bit of first received byte after start condition). Moreover in slave mode, MCU spies the bus after start condition to recognize its address.

When free data format is used (<ALS>=1) all words on the bus are considered as data words, that means no address recognition is done and TRX is not updated.

Figure 3.10.10 Registers for the I^2C Bus Mode

Seirial Bus Interface 2 Conrol 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)
Prohibit Read-modify-write	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 $(fscl = \frac{fc}{2^n + 8}$ [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: Set the <BC2 to 0> to “000” before switching to a clock-synchronous 8-bit SIO mode.
Note 2: For the frequency of the SCL line clock, see 3.10.5 (3) Serial clock.
Note 3: Initial data of SCK0 is “0”, SWRMON is “1”.

Figure 3.10.11 Registers for the I^2C Bus Mode

Serial Bus Interface 2 Control Register 2

SBI2CR2 (0183H)		7	6	5	4	3	2	1	0
	Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
	Read/Write	W				W (Note 1)		W (Note 1)	
	After Reset	0	0	0	1	0	0	0	0
Prohibit Read-modify-write	Function	Master/Slave selection	Transmitter/ Receiver selection	Start/Stop generation	Cancel INTSBE1 interrupt request	Serial bus interface operating mode selection (note 2) 00: Port Mode 01: SIO Mode 10: I^2C 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^2C 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: Reading this register function as SBI2SR register.
Note2: Switch a mode to port mode after confirming that the bus is free.
Switch a mode between I^2C bus mode and clock-synchronous 8-bit SIO mode after confirming that input signals via port are high-level.

Figure 3.10.12 Registers for the I^2C Bus Mode

Serial Bus Interface 2 Status Register

SBI2SR (017BH)		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
Prohibit Read-modify-write	Function	Master/ Slave status monitor	Transmitter/ Receiver status monitor	I^2C bus status monitor	INTSBE1 interrupt request monitor	Arbitration lost detection monitor 0: – 1: Detected	Slave address match detection monitor 0: 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: Writing in this register functions as SBI2CR2.
Note2: If SBI2SR<BB> drops down from 1 to 0 (falling edge), INTSBS2 will be generated in both case of Master mode and Slave mode.

Figure 3.10.13 Registers for the I^2C Bus Mode

Serial Bus Interface 2 Baud Rate Regster 0

SBI2BR0 (0184H) Prohibit Read-modify-write

	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)

	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) Prohibit Read-modify-write

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

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

I^2C Bus 2 Address Register

I2C2AR (0182H) Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
After Reset	0	0	0	0	0	0	0	0
Function	Slave address selection for when device is operating as slave device							Addressing or free data format

Address recognition mode specification

0	Addressing format
1	Free data format

Addressing or free data format impact both slave and master configuration.

When addressing format is used (<ALS>=0), TRX bit is updated relying on R/W bit (=8th bit of first received byte after start condition). Moreover in slave mode, MCU spies the bus after start condition to recognize its address.

When free data format is used (<ALS>=1) all words on the bus are considered as data words, that means no address recognition is done and TRX is not updated.

Figure 3.10.14 Registers for the I^2C Bus Mode

3.10.5 Control in I^{2}C Bus Mode

(1) Specifying acknowledge mode

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

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

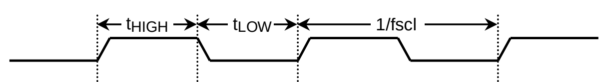
(2) Number of transfer bits

The SBI0CR1<BC2 to 0> setting determines the number of data bits to be transmitted or received.
Since the SBI0CR1<BC2 to 0> is cleared to “000” on start-up, a slave address and direction bit transmissions are executed in 8 bits. Other than these, the <BC2 to 0> retains a specified value.

(3) Serial clock

i) Clock source

The SBI0CR1 <SCK3 to 0> is used to specify the maximum transfer frequency for output on the SCL0 pin in the 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

ii) Clock synchronization

In the I²C bus mode, in order to wired-AND a bus, a master device which pulls down a clock line to the low-level, in the first place, invalidate a clock pulse of another master device which generates a high-level clock pulse. The master device with a high-level clock pulse needs to detect the situation and implement the following procedure.

This device has a clock synchronization function which allows normal data transfer even when more than one master exists on the bus.

The following example explains the clock synchronization procedures used when there are two masters present on the bus.

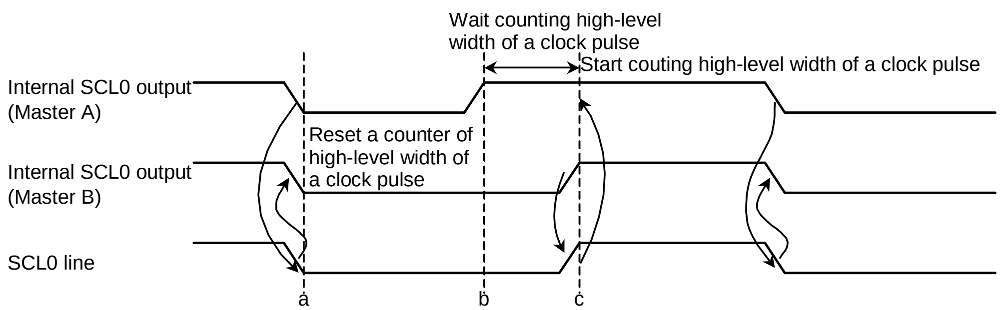


Figure 3.10.16 Clock Synchronization

When Master A pulls the internal SCL0 output to the low-level at point “a”, the SCL0 line of the bus goes to the low-level. After detecting this, Master B resets a counter of high-level width of an own clock pulse and sets the internal SCL0 output the low-level. Master A finishes counting low-level width of an own clock pulse at point “b” and sets the internal SCL0 output to the high-level. Since Master B is holding the SCL0 line of the bus at the low-level, Master A waits for counting high-level width of an own clock pulse. After Master B has finished counting low-level width of an own clock pulse at point “c” and Master A detects the SCL0 line of the bus at the high-level, and starts counting high-level of an own clock pulse.

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

(4) Slave address and address recognition mode specification

When this device is to be used as an I2C slave device, set the slave address <SA6 to 0> and <ALS> in I2C0AR. Clear the <ALS> to “0” for addressing format. When this devices is to be used as an I2C master device, clear <ALS> to “0” for addressing format.

When this device is tu be used in free data format system (as slave or master) set <ALS> to “1” for free data format.

(5) Master/slave selection

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

(6) Transmitter/receiver selection

To operate this device as a transmitter set the SBI0CR2<TRX> to “1”. To operate it as a receiver clear the SBI0CR2<TRX> to “0”. When data with an addressing format is transferred in the slave mode, when a slave address with the same value that an I2C0AR or a GENERAL CALL is received (all 8-bit data are “0” after a start condition), the <TRX> is set to “1” in hardware if the direction bit ($R/\overline{W}$) sent from the master device is “1”, and is cleared to “0” in hardware if the bit is “0”. In the master mode, when an acknowledge signal is returned from the slave device, the <TRX> is cleared to “0” in hardware if the value of the transmitted direction bit is “1”, and is set to “1” in hardware if the value of the bit is “0”. If an acknowledge signal is not returned, the current state is maintained.

The <TRX> is cleared to “0” in hardware when a stop condition is detected on the I^2C bus or when arbitration is lost.

(7) Start/Stop condition generation

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

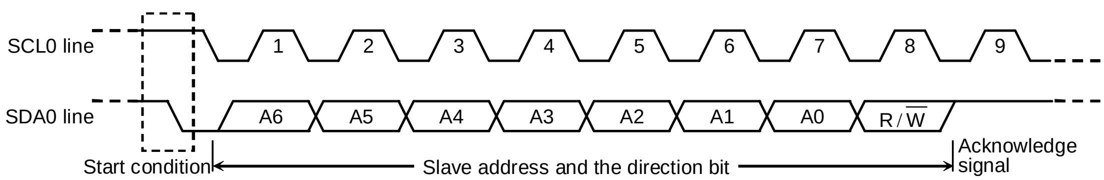


Figure 3.10.17 Start Condition Generation and Slave Address Generation

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

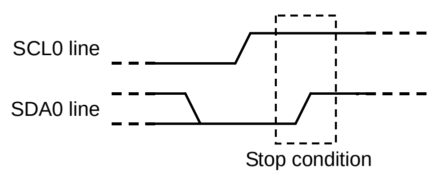


Figure 3.10.18 Stop Condition Generation

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

If SBI0SR<BB> drops down from 1 to 0 (falling edge), INTSBS0 will be generated in both case of Master mode and Slave mode.

(8) Interrupt service requests and interrupt cancellation

When a serial bus interface interrupt request 0 by transfer of the slave address or the data (INTSBE0) is generated, the SBI0SR<PIN> is cleared to “0”. The SCL0 line is pulled down to the low-level while the <PIN> = “0”.
The <PIN> is cleared to “0” when a single word of data is transmitted or received. Either writing data to or reading data from SBI0DBR sets the <PIN> to “1”.
The time from the <PIN> being set to “1” until the release of the SCL0 line is t_{LOW}.
In the address recognition mode (i.e. when <ALS> = “0”; Addressing format), the <PIN> is cleared to “0” when the slave address matches the value set in I2C0AR or when a GENERAL CALL is received (all 8-bit data are “0” after a start condition). Although the SBI0CR2<PIN> can be set to “1” by a program, writing “0” to the SBI0CR2<PIN> does not clear it to “0”.

(9) Serial bus interface operation mode selection

The SBI0CR2<SBIM1 to 0> is used to specify the serial bus interface operation mode. Set the SBI0CR2<SBIM1 to 0> to “10” when the device is to be used in I^{2}C Bus Mode.
Switch to port mode confirming that the bus is free.

(10) Arbitration lost detection monitor

Since more than one master device can exist simultaneously on the bus in I^{2}C Bus Mode, a bus arbitration procedure has been implemented in order to guarantee the integrity of transferred data.
Data on the SDA0 line is used for I^{2}C bus arbitration.
The following example illustrates the bus arbitration procedure when there are two master devices on the bus. Master A and Master B output the same data until point “a”. After Master A outputs “L” and Master B, “H”, the SDA0 line of the bus is wire-AND and the SDA0 line is pulled down to the low level by Master A. When the SCL0 line of the bus is pulled up at point “b”, the slave device reads the data on the SDA0 line, that is, data in Master A. Data transmitted from Master B becomes invalid. The Master B state is known as “ARBITRATION LOST”. Master B device which loses arbitration releases the internal SDA0 output in order not to affect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

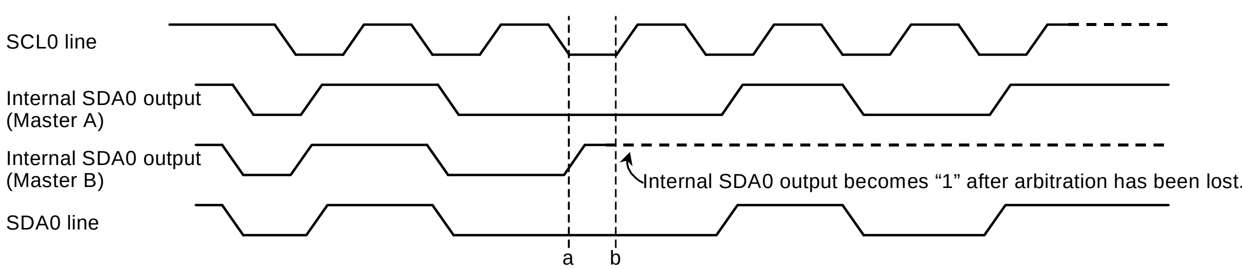


Figure 3.10.19 Arbitration Lost

This device compares the levels on the bus's SDA0 line with those of the internal SDA0 output on the rising edge of the SCL0 line. If the levels do not match, arbitration is lost and the SBI0SR<AL> is set to "1".

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

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

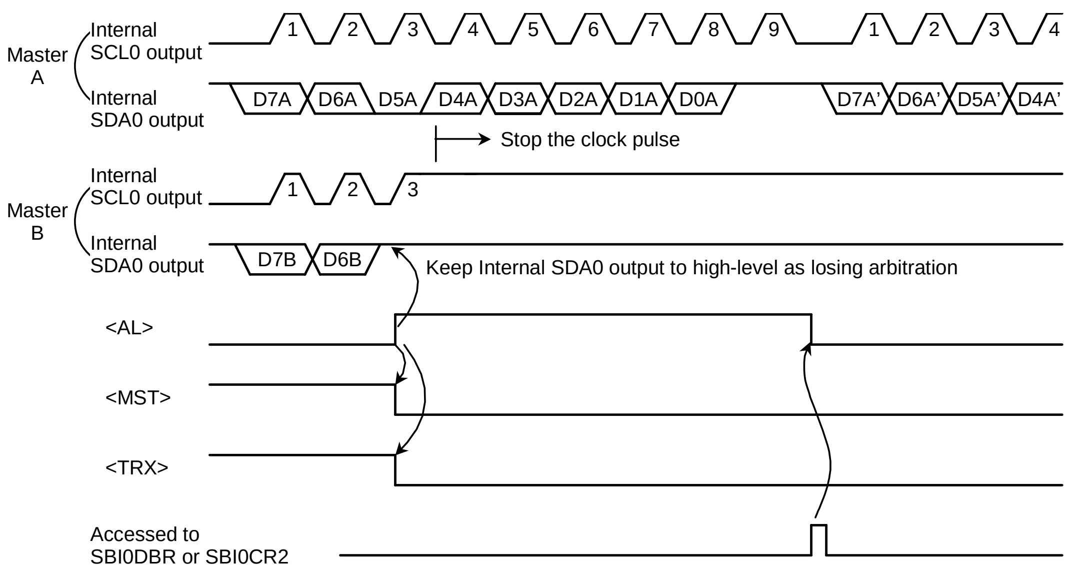


Figure 3.10.20 Example of a Master Device B
(D7A = D7B, D6A = D6B)

(11) Slave address match detection monitor

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

(12) GENERAL CALL detection monitor

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

(13) Last received bit monitor

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

(14) Software Reset function

The software Reset function is used to initialize the SBI circuit, when SBI is rocked by external noises, etc.
An internal Reset signal pulse can be generated by setting SBI0CR2<SWRST1 to 0> to "10" and "01". This initializes the SBI circuit internally. All control registers and status registers excluding SBI0CR2<SBIM1 to 0> are initialized as well.
The SBI0CR2<SWRST1 to 0> is automatically cleared to "00" after the SBI circuit has been initialized.

The initialization of SBI circuit can be confirmed by monitoring SBI0CR1<SWRMON>.

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

The received data can be read and the transferred data can be writtenby reading or writing the SBI0DBR.
When the start condition has been generated in the master mode, the slave address and the direction bit are set in this register.

(16) I^2C Bus Address Register (I2C0AR)

I2C0AR<SA6 to 0> is used to set the slave address when this device functions as a slave device.
ALS bit is used to select between addressing and free data format.

- For I2C bus, addressing format is used (<ALS>=0) ; then TRX bit is updated relying on R/W bit (=8th bit of first received byte after start condition). Moreover, in slave mode, MCU spies the bus after start condition to recognize its address
- For free data format (ALS=1) all words on the bus are considered as data words, that means no address recognition is done and TRX is not updated

(17) Baud Rate Register (SBI0BR1)

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

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

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

3.10.6 Data Transfer in I^{2}C Bus Mode

(1) Device Initialization

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

(2) Start Condition Generation and Slave Address Generation

i) Master mode

In the master mode the start condition and the slave address are generated as follows.
Check a bus free status (when <BB>= "0").
Set the SBI0CR1<ACK> to "1" (acknowledge mode) and specify a slave address and a direction bit to be transmitted to the SBI0DBR.
When the <BB> is "0", the start condition is generated by writing "1111" to the SBI0CR2<MST,TRX,BB,PIN>. Subsequently to the start condition, nine clocks are output from the SCL0 pin. The slave address and the direction bit set to the SBI0DBR will be outputting during the 8 clocks. At the 9th clock pulse the SDA0 line is released and the acknowledge signal is received from the slave device.
An INTSBE0 interrupt request occurs on the falling edge of the ninth clock pulse. The <PIN> is cleared to "0". In the master mode the SCL0 pin is pulled down to the low-level while the <PIN> is "0". When an INTSBE0 interrupt request occurs, the value of <TRX> is changed according to the direction bit setting only if the slave device returns an acknowledge signal.

ii) Slave mode

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

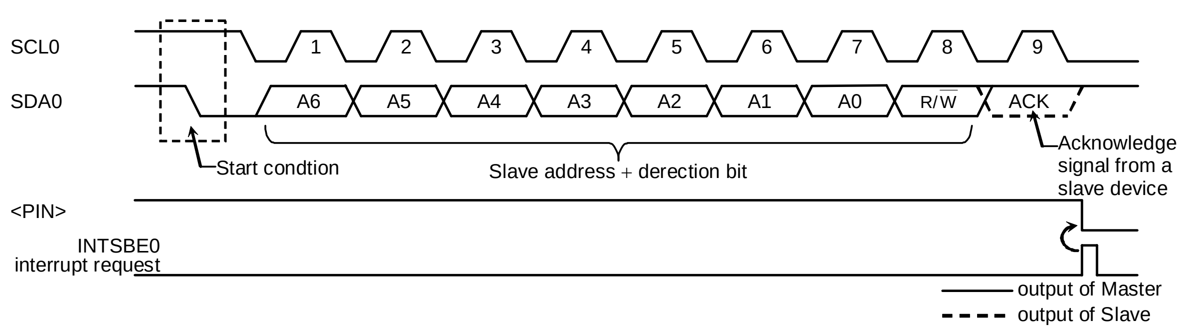


Figure 3.10.21 Start Condition Generation and Slave Address Transfer

(3) 1-word Data Transfer

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

i) When the <MST> is “1” (Master mode)

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

Note: TRX bit is only valid in addressing format (<ALS>=0).

When the <TRX> is “1” (Transmitter mode)

Check the <LRB> setting. When the <LRB> = “1”, there is no receiver requesting data. Implement the process for generating a stop condition (see Section 3.10.6 (4)) and terminate data transfer.

When the <LRB> = “0”, the receiver is requesting new data. When the next transmitted data is 8 bits, write the transmitted data to the SBI0DBR. When the next transmitted data is other than 8 bits, set the <BC2 to 0>, set the <ACK> to “1” and write the transmitted data to the SBI0DBR. After the data has been written, the <PIN> is set to “1”, a serial clock pulse is generated to trigger transfer of the next word of data via the SCL0 pin, and the word is transmitted. After the data has been transmitted, an INTSBE0 interrupt request is generated. The <PIN> is set to “0” and the SCL0 line is pulled down to the low-level. If the length of the data to be transferred is greater than one word, repeat the latter steps of the procedure, starting from the check of the <LRB> setting.

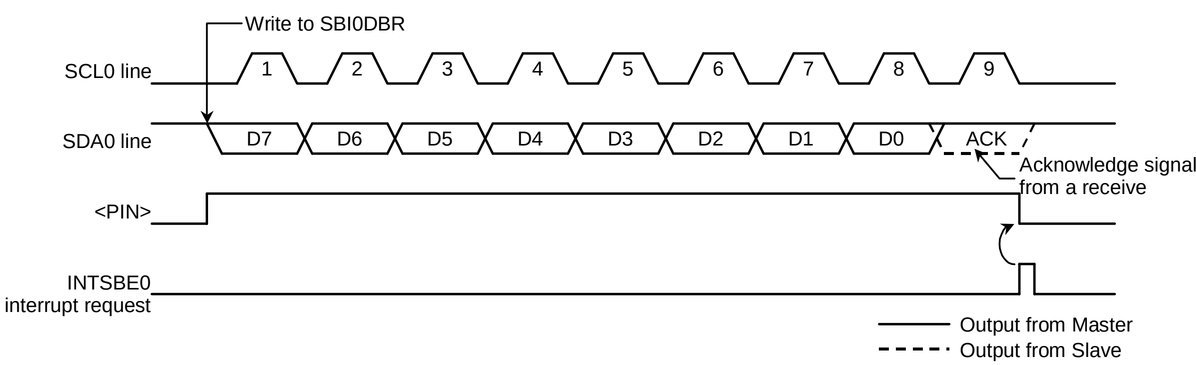


Figure 3.10.22 Example in which <BC2 to 0> = “000” and <ACK> = “1” in Transmitter Mode

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

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

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

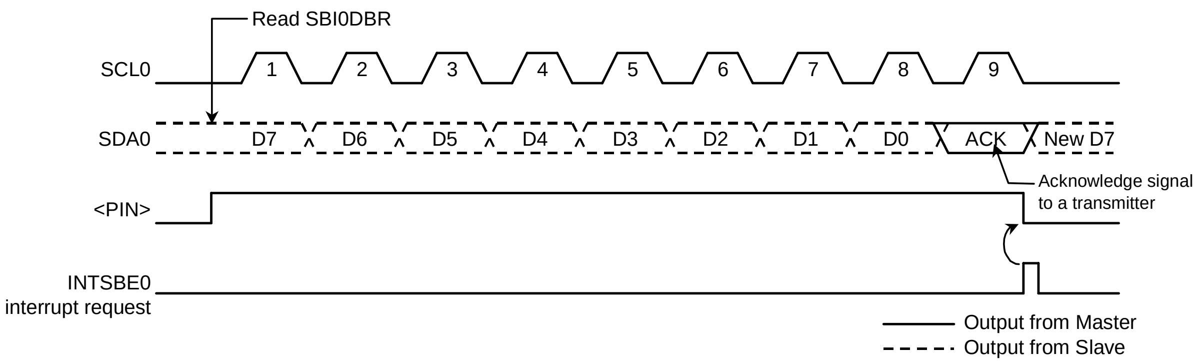


Figure 3.10.23 Example of when <BC2 to 0> = "000", <ACK> = "1" in Receiver Mode

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

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

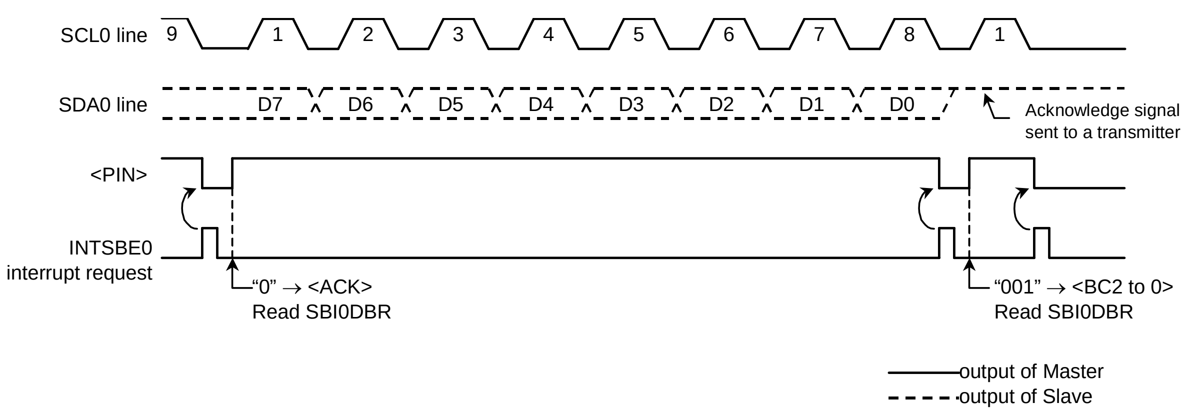


Figure 3.10.24 Termination of Data Transfer in Master Receiver Mode

ii) When the <MST> is “0” (Slave mode)

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

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

If the stop condetion is detected and SBI0SR<BB> drops down from 1 to 0, INTSBS0 will be generated.

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

Note: The <PIN> is set to "0" and the SCL0 pin is pulled down to the low-level, when this device as a master loses arbitration while sending slave address and is called as the slave. In the following 2 cases, the interrupt request is generated when data transfer is finished after losing arbitration, but <PIN> is not set to "0".

- The case that this device as a master loses arbitration while sending slave address and the slave address sent from another device does not correspond to this device.
- The case that this device as a master loses arbitration while sending the data.

Table 3.10.1 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	This device loses arbitration when transmitting a slave address and receives a slave address of which the value of the direction bit sent from another master is “1”.	Set the number of bits in 1-word to the <BC2 to 0> and write the transmitted data to the SBI0DBR.
	0	1	0	In the salve receiver Mode, this device receives a slave address of which the value of the direction bit sent from the master is “1”.	
		0	0	In the salve transmitter mode, 1-word data is transmitted.	Check the <LRB>. If the <LRB> is set to “1”, set the <PIN> to “1” since the receiver does not request the next data. Then, clear the <TRX> to “0” to release the bus. If the <LRB> is cleared to “0”, set the number of bits in a word to the <BC2 to 0> and write transmitted data to the SBI0DBR since the receiver requests next data.
0	1	1	1/0	This device loses arbitration when transmitting a slave address and receives a GENERAL CALL or slave address of which the value of the direction bit sent from another master is “0”.	Read the SBI0DBR for setting the <PIN> to “1” (reading dummy data) or set the <PIN> to “1”.
		0	0	This device loses arbitration when transmitting a slave address or data and terminates transferring word data.	Although INTSEBE0 interrupt occurs after finishing transmitting, this device is slave receiver mode. In this case the <PIN> is not cleared to '0'. Execute the program again in the case of transmitting again as a master.
	0	1	1/0	In the slave receiver mode, this device receives a GENERAL CALL or slave address of which the value of the direction bit sent from the master is “0”.	Read the SBI0DBR for setting the <PIN> to “1” (reading dummy data) or set the <PIN> to “1”.
		0	1/0	In the slave receiver mode, the device terminates receiving 1-word data.	Set the number of bits in a word to the <BC2 to 0> and read received data from the SBI0DBR.

(4) Stop condition generation

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

Whenever a stop condition is detected, an INTSBS0 interrupt request will be generated in both case of master mode and slave mode, regardless of whether it means to stop data transfer or not.

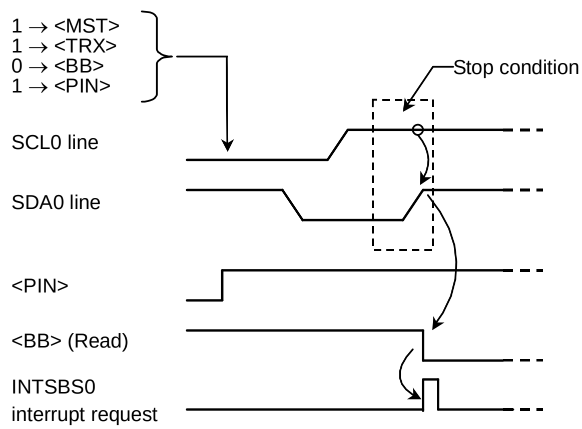


Figure 3.10.25 Stop Condition Generation

(5) Restart

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

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

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

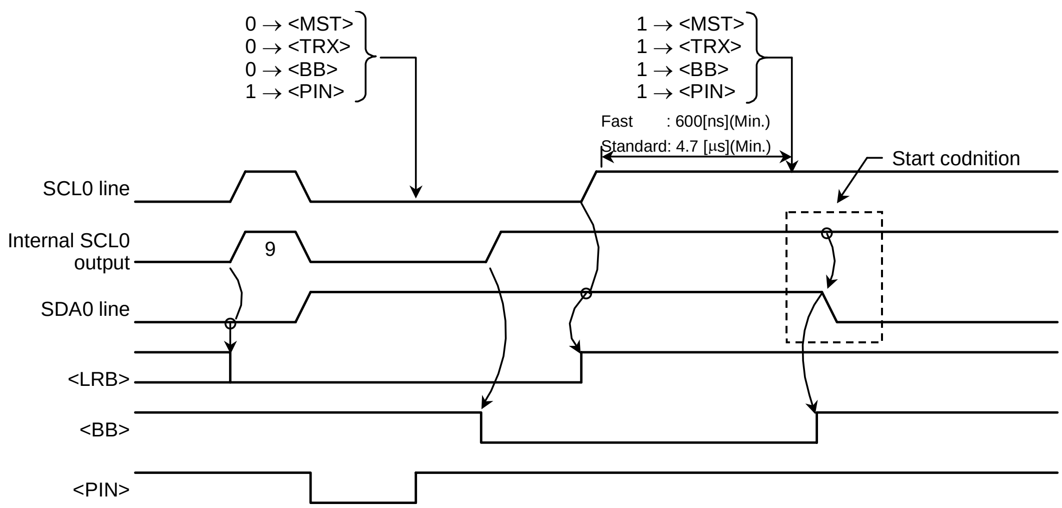


Figure 3.10.26 Timing Diagram when Restarting

3.10.7 Clocked Synchronous 8-Bit SIO Mode control

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

Serial Bus Interface 0 Control Register 1

SBI0CR1 (0170H)

Prohibit Read-modify-write

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

Software reset state monitor <SWRMON> @ read

0	During software reset
1	Not during software reset

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

Note2: Write 0 to this bit in SIO mode.

Serial Bus interface 0 Data Buffer Register

SBI0DBR (0171H)

Prohibit Read-modify-write

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

Figure 3.10.27 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
Prohibit Read-modify-write	Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I^2C 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	I^2C bus mode
11	(reserved)

Note1: Set the SBI0CR1<BC2 to 0> “000” before switching to a clocked-synchronous 8-bit SIO mode.
Note2: Please always write “00” to SBI0CR2<1:0>.

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
1	Transfer in progress

Figure 3.10.28 Registers for the SIO Mode

Serial B us Interface 0 Baud Rate Register 0

SBI0BR0 (0174H)		7	6	5	4	3	2	1	0
	Bit symbol	-	I2SBI0	-	-	-	-	-	-
	Read/Write	W	R/W						
	After Reset	0	0	-	-	-	-	-	-
Prohibit Read-modify-write	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

Figure 3.10.29 Registers for the SIO Mode

Serial Bus Interface 1 Control Register 1

SBI1CR1 (0178H)

	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
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		

Prohibit Read-modify-write

Serial clock selection <SCK2 to 0> @ write

000	n = 4	1.25 MHz	CPU clcok: fc fc = 20 MHz (output to SCK pin) $\text{fscl} = \frac{fc}{2^n}$ [Hz]
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	

Software reset state monitor <SWRMON> @ read

0	During software reset
1	Not during software reset

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

Note2: Write 0 to this bit in SIO mode.

Serial Bus interface 1 Data Buffer Register

SBI1DBR (0179H)

Prohibit Read-modify-write

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

Figure 3.10.30 Register for the SIO Mode

Serial Bus Interface 1 Control Register 2

SBI1CR2 (017BH)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	SBIM1	SBIM0	-	-
Read/Write					W		W	W
After Reset	-	-	-	-	0	0	0	0
Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I^2C 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	I^2C bus mode
11	(reserved)

Note1: Set the SBI1CR1<BC2 to 0> “000” before switching to a clocked-synchronous 8-bit SIO mode.
Note2: Please always write “00” to SBI1CR2<1:0>.

Serial Bus Interface 1 Status Register

SBI1SR (017BH)

	7	6	5	4	3	2	1	0
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
1	Transfer in progress

Figure 3.10.31 Registers for the SIO Mode

Serial Bus Interface 1 Baud Rate Register 0

SBI1BR0 (017CH)
Prohibit Read-modify-write

	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 3.10.32 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
Prohibit Read-modify-write	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	CPU clcok: fc fc = 20 MHz (output to SCK pin) $fscl = \frac{fc}{2^n}$ [Hz]
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	

Software reset state monitor <SWRMON> @ read

0	During software reset
1	Not during software reset

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

Note2: Write 0 to this bit in SIO mode.

Serial Bus interface 2 Data Buffer Register

SBI2DBR (0181H)		7	6	5	4	3	2	1	0
Prohibit Read-modify-write	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 3.10.33 Register for the SIO Mode

Serial Bus Interface 2 Control Register 2

SBI2CR2 (0183H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	SBIM1	SBIM0	-	-
Read/Write					W		W	W
After Reset	-	-	-	-	0	0	0	0
Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I^2C 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	I^2C bus mode
11	(reserved)

Note1: Set the SBI2CR1<BC2 to 0> “000” before switching to a clocked-synchronous 8-bit SIO mode.
Note2: Please always write “00” to SBI2CR2<1:0>.

Serial Bus Interface 2 Status Register

SBI2SR (0183H)

	7	6	5	4	3	2	1	0
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
1	Transfer in progress

Figure 3.10.34 Registers for the SIO Mode

Serial Bus Interface 2 Baud Rate Register 0

SBI2BR0 (0184H) Prohibit Read-modify-write

	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 2 Baud Rate Register 1

SBI2BR1 (0185H)

	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 3.10.35 Registers for the SIO Mode

(1) Serial Clock

i) Clock source

SBI0CR1<SCK2 to 0> is used to select the following functions:

Internal Clock

In an internal clock mode, any of seven frequencies can be selected. The serial clock is output to the outside on the SCK0 pin. The SCK0 pin becomes a high-level when data transfer starts. When the device is writing (in the transmit mode) or reading (in the receive mode) data cannot follow the serial clock rate, an automatic wait function is executed to stop the serial clock automatically and holds the next shift operation until reading or writing is complete.

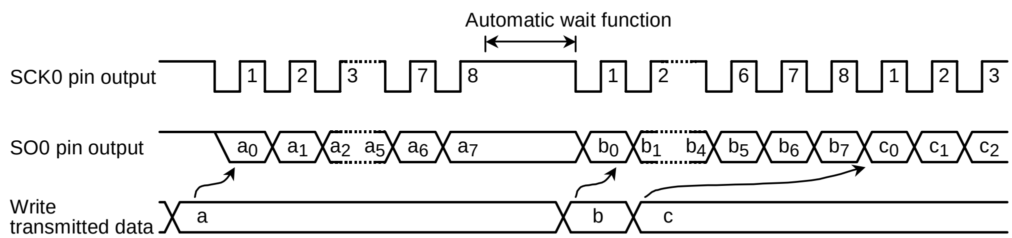


Figure 3.10.36 Automatic-wait Function

External clock (<SCK2 to 0> = "111")

An external clock input via the SCK0 pin is used as the serial clock. In order to ensure the integrity of shift operations, both the high and low-level serial clock pulse widths shown below must be maintained. The maximum data transfer frequency is 1.25 MHz (when fc = 20 MHz).

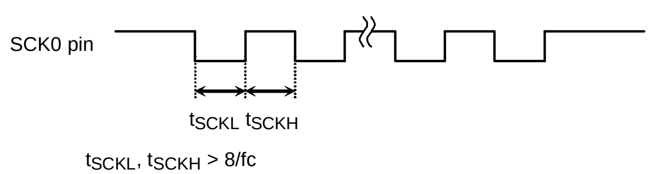


Figure 3.10.37 Maximum Data Transfer Frequency when External Clock Input

ii) Shift edge

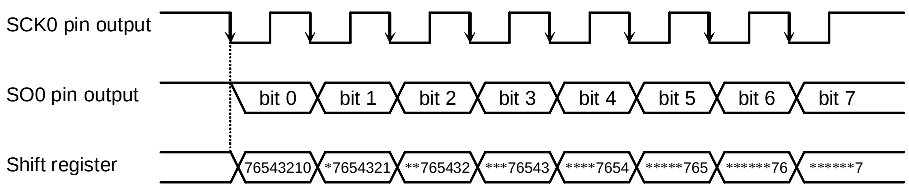
Data is transmitted on the falling edge of the clock and received on the rising edge.

Falling edge shift

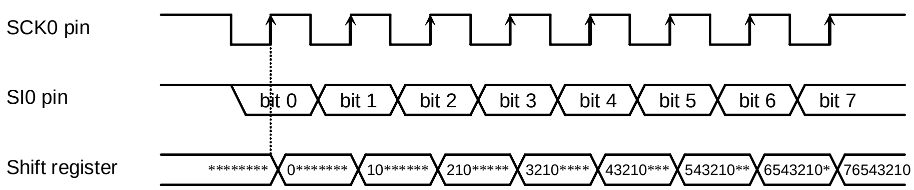
Data is shifted on the falling edge of the serial clock (on the falling edge of the SCK0 pin input/output).

Rising edge shift

Data is shifted on the rising edge of the serial clock (on the rising edge of the SCK0 pin input/output).



(a) Falling edge shift



(b) Rising edge shift   Note: * = Don't care

Figure 3.10.38 Shift Edge

(2) Transfer Modes

The SBI0CR1<SIOM1 to 0> is used to select a transmit, receive or transmit/receive mode.

i) 8-bit transmit mode

Set a control register to a transmit mode and write transmission data to the SBI0DBR. After the transmit data has been written, set the SBI0CR1<SIOS> to "1" to start data transfer. The transmitted data is transferred from the SBI0DBR to the shift register and output, starting with the least significant bit (LSB), via the SO0 pin and synchronized with the serial clock. When the transmission data has been transferred to the shift register, the SBI0DBR becomes empty. The INTSBE0 (buffer empty) interrupt request is generated to request new data.

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

When the external clock is used, data should be written to the SBI0DBR before new data is shifted. The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written to the SBI0DBR by the interrupt service program.

When the transmit is started, after the SBI0SR<SIOF> goes "1" output from the SO0 pin holds final bit of the last data until falling edge of the SCK0.

Data transmission ends when the <SIOS> is cleared to "0" by the INTSBE0 interrupt service program or when the <SIOINH> is set to "1". When the <SIOS> is cleared to "0", the transmitted mode ends when all data is output. In order to confirm whether data is being transmitted properly by the program, the <SIOF> (bit 3 of the SBI0SR) to be sensed. The SBI0SR<SIOF> is cleared to "0" when transmission has been completed. When the <SIOINH> is set to "1", transmitting datat stops. The <SIOF> turns "0".

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

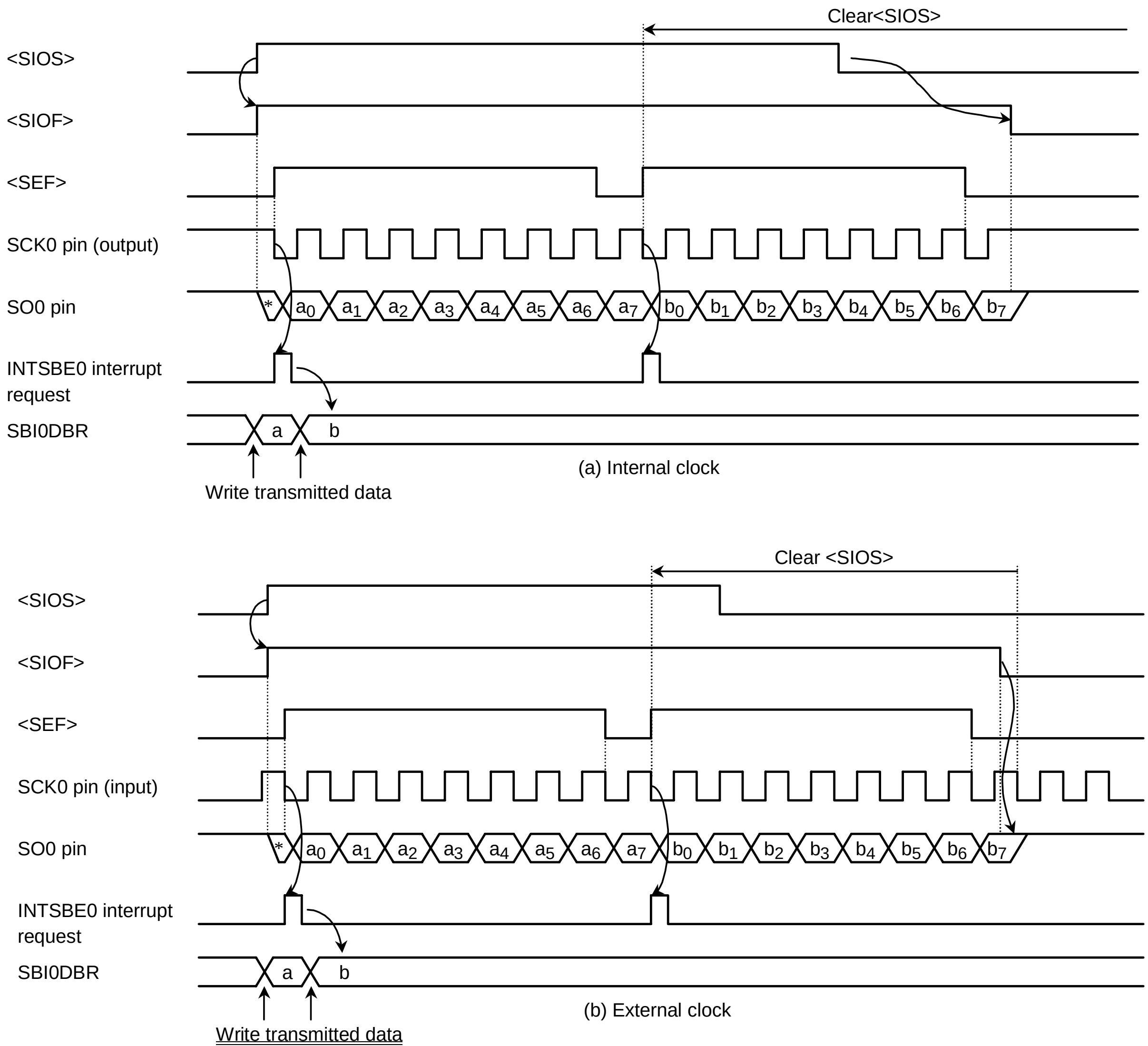


Figure 3.10.39 Transfer Mode

Example: Program to stop data transmission (when an external clock is used)

```
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
```

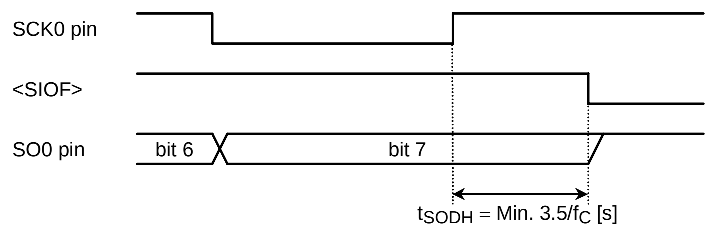


Figure 3.10.40 Transmitted Data Hold Time at End of Transmission

ii) 8-bit receive mode

Set the control register to receive mode and set the SBI0CR1<SIOS> to “1” for switching to receive mode. Data is received into the shift register via the SI0 pin and synchronized with the serial clock, starting from the least significant bit (LSB). When the 8-bit data is received, the data is transferred from the shift register to the SBI0DBR. The INTSBE0 (buffer full) interrupt request is generated to request that the received data be read. The data is then read from the SBI0DBR by the interrupt service program.

When the internal clock is used, the serial clock will stop and the automatic wait function will be in effect until the received data is read from the SBI0DBR.

When the external clock is used, since shift operation is synchronized with an external clock pulse, the received data should be read from the SBI0DBR before the next serial clock pulse is input. If the received data is not read, further data to be received is canceled. The maximum transfer speed when an external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when the received data is read.

Receiving of data ends when the <SIOS> is cleared to “0” by the INTSBE0 interrupt service program or when the <SIOINH> is set to “1”. If <SIOS> is cleared to “0”, received data is transferred to the SBI0DBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm whether data is being received properly by the program, the SBI0SR<SIOF> to be sensed. The <SIOF> is cleared to “0” when receiving is complete. When it is confirmed that receiving has been completed, the last data is read. When the <SIOINH> is set to “1”, data receiving stops. The <SIOF> is cleared to “0” (the received data becomes invalid, therefore no need to read it).

Note: The transfer mode needs to be changed, after reading the last received data with instruction to finish data receiving by clearing the <SIOS> to "0".

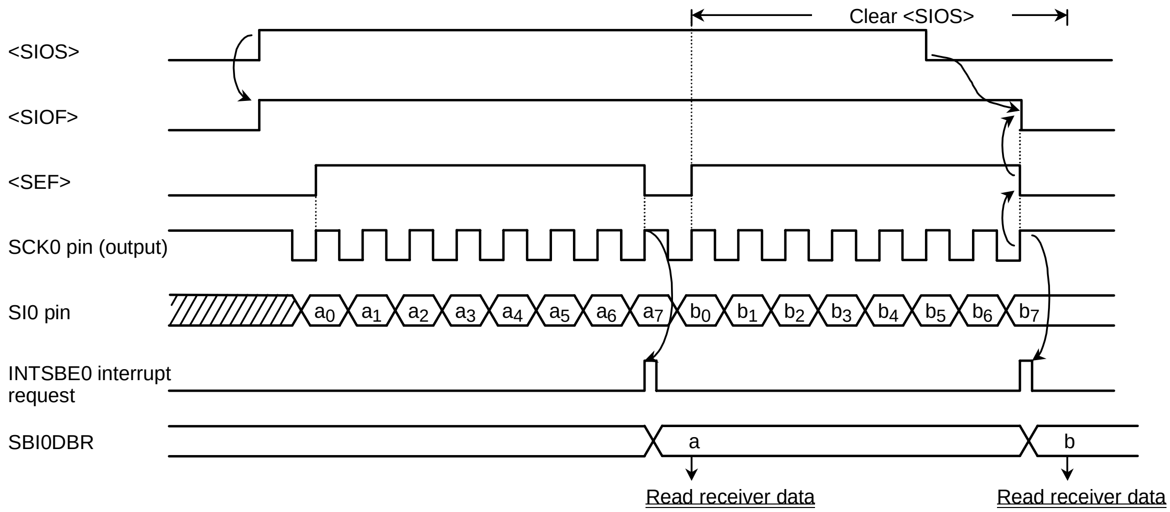


Figure 3.10.41 Receiver Mode (example: Internal clock)

iii) 8-bit transmit/receive mode

Set a control register to a transmit/receive mode and write data to the SBI0DBR. After the data is written, set the SBI0CR<SIOS> to “1” to start transmitting/receiving. When data is transmitted, the data is output from the SO0 pin, starting from the least significant bit (LSB) and synchronized with the falling edge of the serial clock signal. When data is received, the data is input via the SI0 pin on the rising edge of the serial clock signal. 8-bit data is transferred from the shift register to the SBI0DBR and the INTSBE0 interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the data which is to be transmitted. The SBI0DBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

When the internal clock is used, the automatic wait function will be in effect until the received data is read and the next data is written.

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

When the transmit is started, after the SBI0SR<SIOF> goes “1” output from the SO0 pin holds final bit of the last data until falling edge of the SCK0.

Transmitting/receiving data ends when the <SIOS> is cleared to “0” by the INTSBE0 interrupt service program or when the SBI0CR1<SIOINH> is set to “1”. When the <SIOS> is cleared to “0”, received data is transferred to the SBI0DBR in complete blocks. The transmit/receive mode ends when the transfer is complete. In order to confirm whether data is being transmitted/received properly by the program, set the SBI0SR to be sensed. The <SIOF> is set to “0” when transmitting/receiving is completed. When the <SIOINH> is set to “1”, data transmitting/receiving stops. The <SIOF> is then cleared to “0”.

Note: When the transfer mode is changed, the contents of the SBI0DBR will be lost. If the mode must be changed, conclude data transmitting/receiving by clearing the <SIOS> to “0”, read the last data, then change the transfer mode.

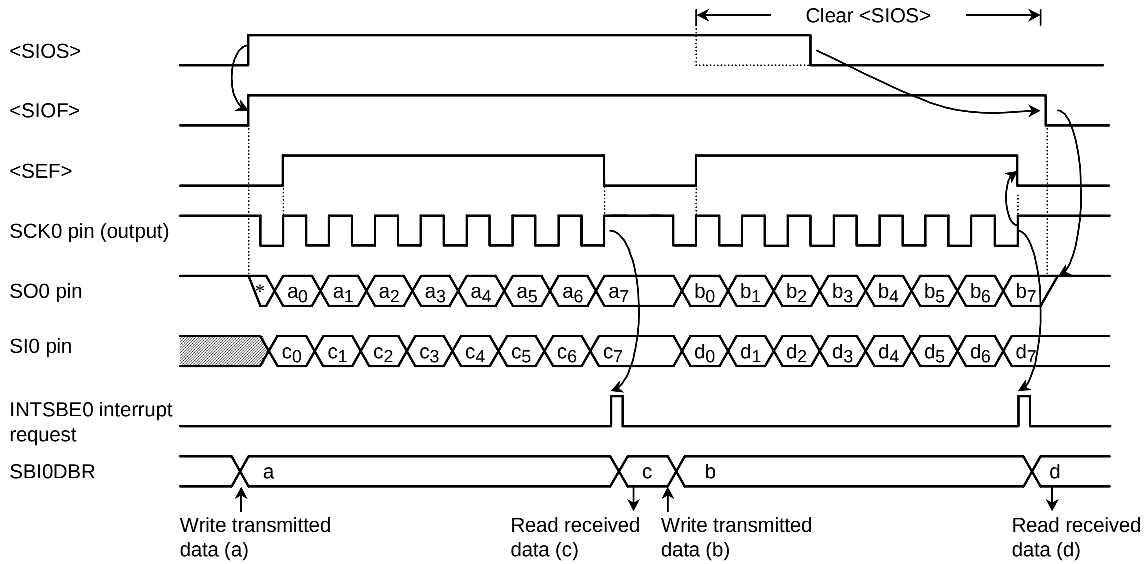


Figure 3.10.42 Transmit/Received Mode (Example : Internal clock)

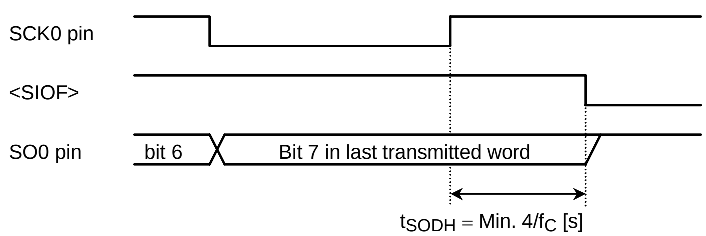


Figure 3.10.43 Transmitted Data Hold Time at End of Transmit/Receive

3.11 Serial Expansion Interface (SEI)

3.11.1 Overview

The SEI is one of the serial interfaces built in the TMP92CD54I, which can be connected to peripheral devices, by full duplex synchronous communication protocol. TMP92CD54I incorporates 1 channel of this SEI. Also the SEI can support the micro DMA mode corresponds to the micro DMA transfer.

(1) Features

- The master outputs the shift clock only during data transfer.
- The clock polarity and phase are programmable
- The data is 8 bits long
- The MSB first or LSB first can be selected
- Micro DMA mode support for micro DMA transfers
- Transfer rate: 4Mbps, 2Mbps or 500kbps (when operating at fc = 20MHz)
- Error detection function

① Write collision detection: when write to the shift register during the data transfer

② Overflow detection: when receive the new data with the transfer end flag is set (only slave)

③ Mode fault detection: when the input to the $\overline{SS}$ pin goes L in Master mode (driver output immediately turns off)

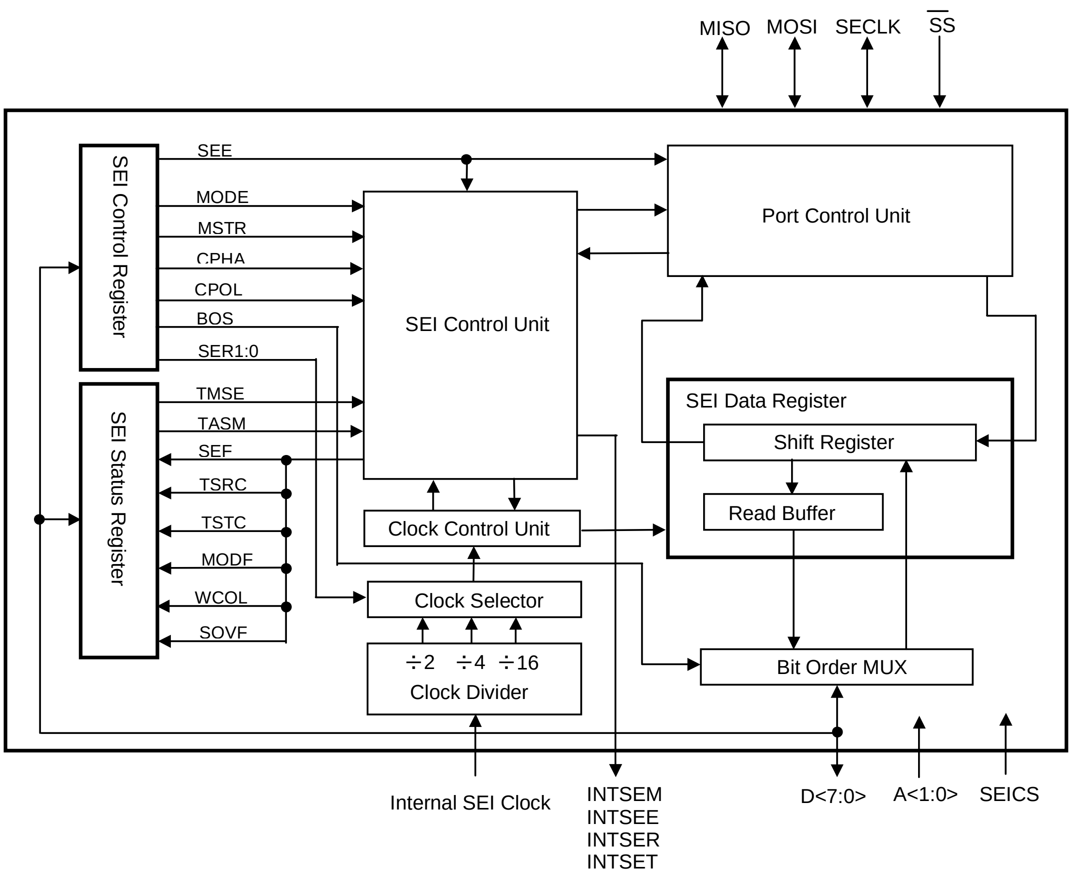


Figure 3.11.1 SEI Block Diagram

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 a SEI transfer, data is simultaneously transmitted (shifted out serially) and received serially (shifted in serially). In order to shift or sample the information on two serial data lines (MOSI/MISO), SEI clock (SECLK) takes the synchronization. Slave selection line ($\overline{SS}$) individually selects the slave device. The slave device not selected cannot use the SEI bus. Because the master function is turned off in the master device when the multi master bus is connected, slave selection line ($\overline{SS}$) can be used.

(1) SEI clock phase and polarity controls

Software can select any four combinations of serial clock phase and polarity using two bits in the SEI control register (SECR). The clock polarity is set by < CPOL > bit, and selects the clock of active "H" or active "L". The clock phase <CPHA> control bit selects one of two fundamentally different transfer formats. The clock phase and polarity should be identical for the master SEI device and the communicating slave device.

(2) SEI data and clock timing

The programmable clock timing and data of SEI can connect almost all devices around synchronous serial. Please see "3.11.4 SEI transfer format" for a detailed description of the transfer format.

3.11.3 SEI signal lines

There are four input/output pin signals associated with the SEI transfer. Every signal depends on the mode (master/slave) of the SEI device.

(1) SECLK

The SECLK pin functions as an output pin when the SEI is set for master and functions as an input pin when the SEI is set for slave.

When the SEI is set for master, the SECLK signal is supplied by the internal SEI clock generation circuit. When the master starts transferring data, eight cycles clock are automatically output at the SECLK pin.

When the SEI is set for slave, the SECLK pin functions as an input pin, in which case the SECLK signal from the master synchronizes data transfers between the master and slave. The slave device ignores the SECLK signal if the slave select $\overline{SS}$ pin is high.

In both master and slave SEI devices, data is shifted in or out at each rising or falling edge of the SECLK signal and is sampled at the opposite edge. Edge polarity is determined by the SEI transfer protocol.

(2) MISO/MOSI

The MISO and MOSI pins are used for transmitting and receiving serial data.

When the SEI is configured as a master, MISO is the data input line and MOSI is the data output line.

When the SEI is configured as a slave, these pins reverse roles.

In a multiple-master system, all SECLK pins are tied together, all MOSI pins are tied together and all MISO pins are tied together. Refer to Figure 3.11.5. A single SEI device is configured as a master, all other SEI devices on the SEI bus are configured as slaves. The single master drives the transfer clock and data out it's SECLK and MOSI pins to the SECLK and MOSI pins of the slaves. One selected slave device optionally drives data out it's MISO pin to the MISO master pin.

The SECLK, MISO and MOSI pins can be set to function as open-drain pins.

(3) $\overline{SS}$

The $\overline{SS}$ pin behaves differently on master and slave devices.

On a slave device, this pin is used to enable the SEI slave for transfer and receive. If the $\overline{SS}$ pin of a slave is inactive (high), the device ignores SECLK clocks and keeps the MISO output pin in the high-impedance state.

On a master device, the $\overline{SS}$ pin serves as an error-detection input for the SEI. If the $\overline{SS}$ pins go low while the SEI is a master, it indicates that some other device on the SEI bus is attempting to be master. This attempt causes the master device sensing the error to immediately exit the SEI bus to avoid potentially damaging driver contentions. This error is called mode fault.

Set whether to permit the mode fault detection by < MODE > bit of the SECR register or to prohibit it. When the <MODE> bit = 0, the $\overline{SS}$ pin is enabled for mode fault detection input. When the <MODE> bit = 1, the $\overline{SS}$ pin is disabled from mode fault detection input.

3.11.4 SEI transfer format

The transfer format is decided the setting of the <CPHA> bit and <CPOL> bit in the SECR register. <CPHA> bit switches between two different transfer protocols.

(1) Transfer Format of <CPHA>=0

Figure 3.11.2 shows the transfer format for a <CPHA>=0 transfer.

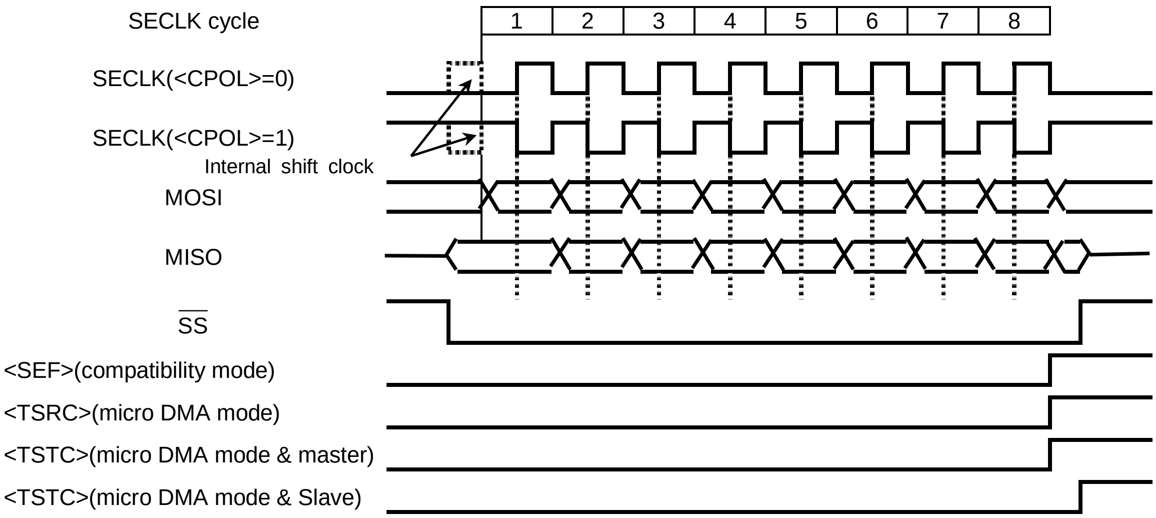


Figure 3.11.2 Transfer Format of <CPHA>=0

<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 starts the transfer. The new data are switched on the MOSI pin half a clock before the shift clock starts the operation. SECR<BOS> selects whether the data are shifted out from the MSB or from the LSB. After the final shift cycle, the <SEF> flag is set to 1 if Compatibility mode is selected, and the <TSRC> and <TSTC> flags are set to 1 if Micro DMA mode is selected.

In slave mode, writing to the SEDR register is prohibited while the $\overline{\text{SS}}$ pin is L. Attempting a write during this period triggers a write collision and sets the SESR register's <WCOL> flag to 1. This terminates the transfer. At this time the software must wait until the $\overline{\text{SS}}$ pin goes H again before writing the next data to the SEDR register, even if the <SEF> or <TSRC> flag is set to 1. When using micro DMA for transferring data to the SEDR register in slave mode, the setting of the <TSTC> flag is delayed until the $\overline{\text{SS}}$ pin goes H.

(2) Transfer format of <CPHA>=1

Figure 3.11.3 shows the transfer format for a <CPHA>=1 transfer.

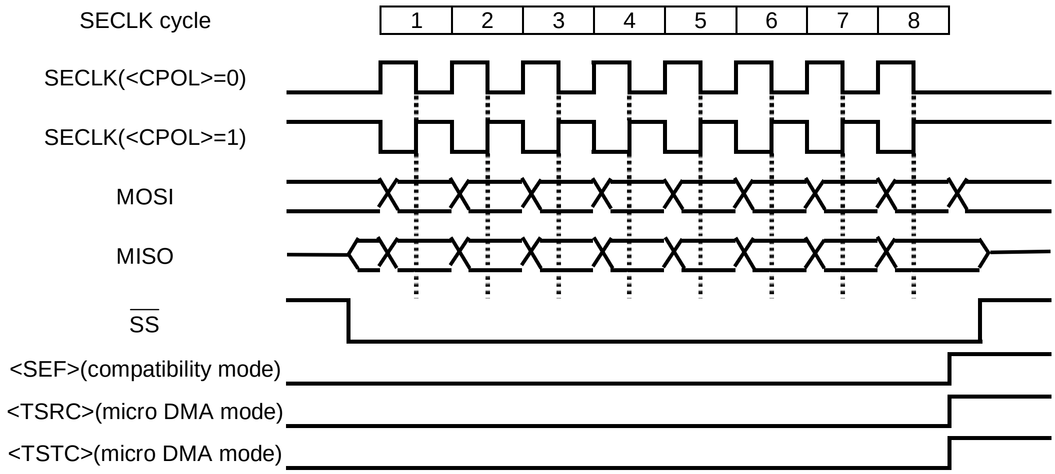


Figure 3.11.3 Transfer Format of <CPHA>=1

<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 starts the transfer. The new data are switched on the MOSI pin at the initial edge of the shift clock. SECR<BOS> selects whether the data are shifted out from the MSB or from the LSB.

In contrast to slave mode with <CPHA>=0, in slave mode when <CPHA>=1, the SEDR register can be written even while the $\overline{SS}$ pin is L. In both master and slave modes, after the final shift cycle the <SEF> flag is set to 1 if compatibility mode is selected, and the <TSRC> and <TSTC> flags are set to 1 if micro DMA mode is selected.

Attempting a write to the SEDR register during a data transfer triggers a write collision. Write the data to the data to SEDR after the <SEF> is set to 1, or the <TSRC> and <TSTC> flags are set.

3.11.5 Functional description

Figure 3.11.4 shows master-to-slave connection via the SEI.

The different nodes on a SEI bus function like a distributed shift register. When data is sent from the MOSI pin of the master device to the corresponding pin of the slave device, data from the slave is sent back from the MISO pin of the slave device to the corresponding pin of the master device.

This means that data is communicated in full-duplex mode and data output and data input are synchronized by the same clock signal. After a transfer, the transmission data of eight bit shift register is replaced with receive data.

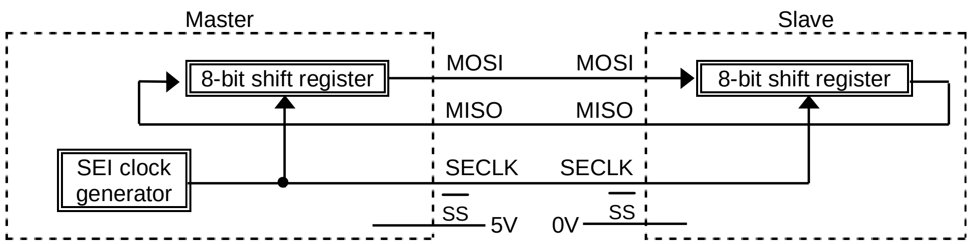
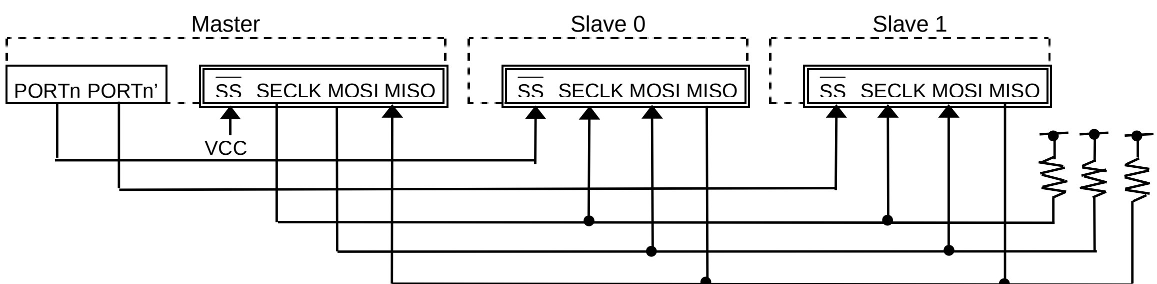


Figure 3.11.4 Connection between Master and Slave in SEI

Figure 3.11.5 shows a configuration of the SEI system.

The port used as the output of SEI, can be set for open-drain output programmable. Therefore, this port can be connected to multiple devices.



PORTn, n': any output ports

Figure 3.11.5 Configuration of SEI System (Comprised of One Master and Two Slaves)

3.11.6 Operation Modes

SEI allows the programmer the choice between 2 different operation modes, the compatibility mode and the micro DMA mode. Those operation modes differ in terms of flag clearing, interrupt generation and use propriety of micro DMA. The table below shows the differences between the two operation modes.

Table 3.11.2 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 SEDR register access	Writing a "1" to the status register
transfer status flag clearing	Reading a register with the Status flag set, followed by an access to the data register	Writing a "1" to the status register or by reading or writing the data register
interrupt generation	INTSEM: <MODF> INTSEE: <SEF>	INTSEM: <MODF> INTSEE: <WCOL> or <SOVF> INTSER: <TSRC> INTSET: <TSTC>
micro DMA usage	No	yes

SEI can be switched between these operation modes, if SEI is disabled (<SEE> = 0) by setting the <TMSE> bit in the SESR register.

3.11.7 SEI registers

Use SEI control register SECR, SEI status register SESR, and SEI data register SEDR to set SEI.

(1) SEI control register (SECR)

SEI Control Register

		7	6	5	4	3	2	1	0
SECR (0060H)	bit Symbol	MODE	SEE	BOS	MSTR	CPOL	CPHA	SER1	SER0
	Read/Write	W	R/W						
	After reset	0	0	0	0	0	1	1	1
Read-modify -write instructions prohibited.	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	

<MODE>: Mode fault detection enable

0: Mode fault detection enabled.

1: Mode fault detection disabled.

Only the master mode is effective and invalid at the slave mode.

<SEE>: SEI function enable

0: SEI function is off. It is necessary to disable the SEI function to switch between the micro DMA mode and the compatibility mode. Wait until the transfer in progress is completed before you clear the <SEE> bit to stop the SEI operation. Clear <SEE> to 0 before executing HALT instruction in IDLE1, IDLE3 or STOP mode.

1: SEI function is on. Before using the SEI, make sure that the port function needs to be set as SEI function.

<BOS>: Bit order select

The bit order selection bit <BOS> selects whether the data to be transferred is MSB first or LSB first.

0: The MSB bit of the SEDR register (bit 7) will be transmitted first.

1: The LSB bit of the SEDR register (bit 0) will be transmitted first.

<MSTR>: Master/Slave mode select

0: SEI is configured as slave.

1: SEI is configured as master.

<CPOL>: Clock polarity select

0: Select the clock of active "H". The SECLK clock is "L" level at non-communication state.

1: Select the clock of active "L". The SECLK clock is "H" level at non-communication state.

Refer to Figure 3.11.2 and Figure 3.11.3.

<CPHA>: Clock phase select

<CPHA> bit selects one of two, different transfer format.

Refer to Figure 3.11.2.and Figure 3.11.3.

<SER1:0>: SEI bit rate select

The following table shows the relationship between the <SER1> and <SER0> control bits and the bit rate for transfers when the TSEI is operating as a master. When the TSEI is operating as a slave, the serial clock is input from the master, therefore, the <SER1> and <SER0> control bits have no meaning.

Table 3.11.3 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	2	4 Mbps
1	0	4	2 Mbps
1	1	16	500 Kbps

Note: internal SEI clock = 2/5 × fc

(2) SEI status register (SESR)

SEI Status Register

		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
compati-bility 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:compati-bility mode 1:micro DMA mode

SEI Status Register

		7	6	5	4	3	2	1	0
SESR (0061H)	bit Symbol	-	WCOL	SOVF	MODF	TSRC	TSTC	TASM	TMSE
	Read/Write		R					R/W	
	After reset	-	0	0	0	0	0	0	0
micro DMA mode Read-modify-write instructions prohibited.	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:compati-bility mode 1:micro DMA mode

<SEF>: Transfer complete flag

Compatibility mode:

The <SEF> flag is automatically set to one at the end of a SEI transfer. The <SEF> flag is automatically cleared to 0 by reading the SESR register with <SEF> flag set to 1, followed by an access of the SEDR register.

Micro DMA mode:

Always reads as undefined, writes to this flag have no effect.

<WCOL>: Write collision error flag

Compatibility mode:

The <WCOL> flag is automatically set to 1, if the SEDR register is written while a transfer is in progress. The write itself has no effect on the running transmission. The <WCOL> flag is automatically cleared to 0 by reading the SESR register with <WCOL> bit set followed by an access to the SEDR register. No interrupt will be generated on the assertion of this flag.

Micro DMA mode:

The <WCOL> flag is automatically set to 1, if the SEDR register is written while a transfer is in progress. The write itself has no effect on the running transmission. The flag can only be reset by writing a "1" to it. Writing a "0" has no effect. An interrupt will be generated on INTSEE on a transition from "0" to "1", if the module is configured as a slave and the <TASM> bit is equal to "0".

<SOVF>: Slave mode overflow error flag

Master mode:

Always reads as undefined, writes to this flag have no effect.

Slave mode:

Compatibility mode:

The <SOVF> flag is automatically set to 1, if a new byte has been completely received and the <SEF> flag is still set to 1. The <SOVF> flag is automatically cleared to 0 by reading the SESR register with the <SOVF> flag is set to 1 followed by an access to the SEDR register. The <SOVF> flag will also be cleared to 0 by switch to the master mode. In compatibility mode, no interrupt will be generated on the setting of <SOVF> flag.

Micro DMA mode:

The <SOVF> flag is automatically set to 1, if a new byte has been completely received and the <TSRC> flag is still set to 1. The <SOVF> flag can only be cleared to 0 by writing a “1” to it. Writing a “0” to it has no effect. INTSEE is generated with <TASM> =1, if <SOVF> flag from 0 to 1.

<MODF>: Mode-fault error flag

Master mode:

Compatibility mode:

The <MODF> flag is set to 1, if the $\overline{\text{SS}}$ signal goes to active low while the SEI is configured as a master. In this case:

1. The SEI output pin drivers are disabled and the output pins are placed in high-impedance state.
2. The <MSTR> bit in the SECR register is cleared to 0.
3. The <SEE> bit is forcibly cleared to 0 to disable the SEI system.
4. An interrupt INTSEM is generated.

The <MODF> flag is automatically cleared to 0 by reading the SESR register with the <MODF> bit set to 1, followed by a write to SECR register.

Micro DMA mode:

It is the same as that of the compatibility mode, except the <MODF> flag's clearance.

This flag can only be cleared to 0 by writing a “1” to it. Writing a “0” to this flag has no effect.

Slave mode:

Always reads as undefined, writes to this flag have no effect.

<TSRC>: Receive completion flag

Compatibility mode:

Always reads as undefined, writes to this flag have no effect.

Micro DMA mode:

The <TSRC> flag is set to 1 when a receiving has been completed, that is when eight cycles where shifted on the SECLK signal. It is cleared to 0 by performing a read operation on the SEI data register, by switching to compatibility mode or by writing a “1” to this flag. Writing a “0” to this flag has no effect. An interrupt INTSER will be generated on the assertion of this flag.

<TSTC>: Transmit completion flag

Compatibility mode:

Always reads as undefined, writes to this flag have no effect.

Micro DMA mode:

Timing where the flag is set by transfer format and master/slave is different though < TSTC > flag is set when the transmission of the data of one byte is completed. Refer to Figure 3.11.2 and Figure 3.11.3. It is cleared to 0 by performing a write operation on the SEI data register, by switching to compatibility mode or by writing a "1" to this flag. Writing a "0" to this flag has no effect. An interrupt INTSET will be generated on the assertion on this flag.

<TASM>: Automated shift mode (master) ／INTSEE interrupt mask (slave)

Compatibility mode:

Always reads as undefined, writes to this flag have no effect.

Micro DMA mode:

The function of this bit is depending on <MSTR> bit setting.

Master mode:

0: Disables the automated shift mode.

1: Enables the automated shift mode.

In this mode a read access to the SEI data register SEDR will perform the following functions.

・ The SEI data register will be cleared to 00 hex.

・ A new transfer will be initiated, thus in master mode 8 low bits will be sent, 8 new bits will be received.

The automated shift mode also works when it is combined with a micro DMA. It has no effect, when SEI is in the slave mode.

Slave mode:

This bit functions as a mask for the interrupt INTSEE generation of the <SOVF> and <WCOL> flags.

0: An interrupt INTSEE will be generated when the <WCOL> flag is set to "1", but not effect on the <SOVF> flag.

1: An interrupt INTSEE will be generated when the <SOVF> flag is set to "1", but not effect on the <WCOL> flag.

<TMSE>: SEI mode select

0: Compatibility mode.

1: Micro DMA mode.

Selects the micro DMA mode, which also allows micro DMA transfers. It is necessary to disable the SEI system before switching to the micro DMA mode.

(3) SEI data register (SEDR)

SEI Data Register

SEDR (Transmission) (0062H)		7	6	5	4	3	2	1	0
	bit Symbol	SED7	SED6	SED5	SED4	SED3	SED2	SED1	SED0
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0

SEDR (Receiving) (0062H)		7	6	5	4	3	2	1	0
	bit Symbol	SED7	SED6	SED5	SED4	SED3	SED2	SED1	SED0
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0

Note: SEDR is not able to read, modify, write.

This register is used to transmit and receive data. When the SEI system configured as a master, transfers are started by a software write to the SEDR register.

After once starting transmission, please write after checking that the transmission end flag has surely set by interrupt or polling when master device writes to SEDR register.

Only when the <SEE> bit of the SECR register is “1”, a read/write to the SEDR register is possible. When the <SEE> bit is “0”, the write access is ignored and “00H” will be read whenever it read.

3.11.8 SEI system errors

Three system errors can be detected by the SEI device. The first type error arises in a multiple-master system when more than one SEI device simultaneously tries to be master. This error is called a mode fault. The second type error, a write collision, indicates that an attempt has been made to write data to the SEDR while a transfer was in progress. The third error occurs when the SEI system is configured as a slave and a new byte of data has been completely shifted in by the remote bus master before the old byte could be read.

(1) Mode-fault error

In the SEI system, if more than one device is simultaneously set as the master, competition arises among the drivers.

When an SEI device is set as the master, a mode fault error occurs when the $\overline{\text{SS}}$ pin input goes L and the driver output goes off. This phenomenon can be used to avoid competition among masters.

When a mode fault error occurs, the following action is immediately taken.

- The SECR register’s <MSTR> bit is forcibly cleared to 0 to set the SEI for slave.
- The SECR register’s <SEE> bit is forcibly cleared to 0 to disable the SEI system.
- The SESR register’s <MODF> flag is set to 1, and INTSEM interrupt pulse is generated.
- The SEI output pin drivers are disabled and the output pins are placed in the high-impedance state.

When the problem which has caused the mode fault is resolved in software, the <MODF> flag is cleared to 0 and the SEI system can be set up to return to normal operation. The writing is not able to the SECR register while the <MODF>flag is set. In compatibility mode the <MODF> flag is automatically cleared by reading the SESR register while the <MODF> flag is set to 1, and then writing to the SECR register. In micro DMA mode the <MODF> flag is cleared to 0 by writing a 1 to it.

Only when two or more devices are selected at the same time as the master, this product detects a mode fault error. The collision of the MISO output when two or more slave devices are selected on the SEI system cannot be detected.

The drivers can be protected from latch-up by means of an open-drain. This involves changing the SEI output driver to be of open-drain type. The SECLK pin, MOSI pin and MISO pin can be individually set as open-drain programmably. In the case, an additional external pull-up register is necessary.

(2) Write collision error

A write collision occurs is the SEDR register is written to while a transfer is in progress. Because the SEDR register is not a double buffer in the direction of the transmission, writing before transfer in the SEDR register is writing directly in the SEI shift register. Because this write corrupts any transfer in progress, a write collision error is generated. The transfer continues undisturbed and the write data which caused the error is not written to the shift register.

A write collision is normally a slave error because a slave has no control over when a master will initiate a transfer. A master knows when a transfer is in progress, thus, there is no excuse for a master generating a write collision error. Despite this, the SEI device can detect write collision in a master as well as in a slave.

In slave mode a write collision is likely to occur, since the master shifts data faster, than it can be handled by the slave. A write collision will occur, when the slave is transferring a new value to the data register after the master started the next shift cycle.

In micro DMA mode an interrupt on INTSEE will occur if the module is configured as a slave, the <TASM> bit is clear to 0 and the <WCOL> flag is set to 1.

(3) Slave mode overflow error

On an SEI bus the transmission bit rate is determined by the master. It becomes easy to cause the problem that the slave cannot follow to the master's transmission by a high-speed bit rate, i.e. that the data is shifted in faster than it can be processed by slave. The SEI device detects data overflowing with < SOVF > flag of the SESR register.

The <SOVF> flag will be set to 1 when:

- The SEI is configured as a slave.
- An old byte of data is still waiting to be read when a new byte of data has been completely received.

When <SOVF> is set to 1, it signifies that SEDR has been overwritten by new byte data.

Since this error only occurs in slave mode, the <TASM> bit can be used as an interrupt mask for this flag. If the <SOVF> flag in the status register is set to 1, an interrupt is only generated on INTSE0 if the current mode is micro DMA mode and the <TASM> bit is 1.

3.11.9 Interrupt generation

Interrupt processing differs for the two SEI operating modes, which can be selected using the <TMSE> bit in the SESR register. It generates four interrupts par one SEI that are INTSEM, INTSEE, INTSER and INTSET.

(1) Compatibility mode

In compatibility mode the INTSEM* and INTSEE are used. *The SEI generates the INTSEM interrupt, if the <MODF> flag in the SESR register shows a transition from “0” to “1”. And it generates the INTSEE interrupt, if the <SEF> flag shows a transition from “0” to “1”.

INTSEM	Interrupt on <MODF>
INTSEE	Interrupt on <SEF>
INTSER	Inactive
INTSET	Inactive

(2) Micro DMA mode

In micro DMA mode all four interrupts are used to allow the micro DMA transfers to and from the SEI data register. The INTSEM is generated on a transition of the <MODF> flag from “0” to “1”. The INTSEE is generated if the module is in slave mode on a transition of the <WCOL> flag from “0” to “1” with <TASM> bit is “0” or on a transition of the <SOVF> flag from “0” to “1” with <TASM> bit is “1”.

After a completed transfer both the <TSRC> flag and the <TSTC> flag in the SESR register are set to 1 simultaneously. However, there is an exception for <CPHA> equals “0” in slave mode. Please see “3.11.4(1) transfer format of <CPHA>=0”. Both flags trigger the INTSER and INTSET interrupts.

The <TSRC> flag generates an interrupt INTSER on a transition from “0” to “1”. The <TSRC> flag can be cleared by either reading the SEDR register or by writing a “1” value to this flag.

The <TSTC> flag generates an interrupt INTSET on a transition from “0” to “1”. The <TSTC> flag is cleared to 0 by either writing the SEDR register or by writing a “1” value to this flag.

In order to use the micro DMA, the INTSER interrupt and the INTSET interrupt are used as a trigger of micro DMA transmission.

The INTSER interruption: The data read from the SEDR register is used as a trigger of micro DMA transfer.
The INTSET interruption: A new data write to the SEDR register is used as a trigger of micro DMA transfer.

Thus initiating a new transfer.

INTSEM	Interrupt on <MODF>
INTSEE	Interrupt on <WCOL>[1)] or <SOVF>[2)]
INTSER	Interrupt on <TSRC>
INTSET	Interrupt on <TSTC>

Note 1) In slave mode, it is at the time of <TASM> =0

Note 2) In slave mode, it is at the time of <TASM> =1

The Interrupts can be disabled individually at the interrupt controller.

3.11.10 Usage of the micro DMA of SEI (micro DMA mode)

The usage of the micro DMA for larger SEI transfers allows speed up the communication on the SEI by

- reducing the CPU effort for interrupt processing,
- reducing the time gap between two successive transfers.

The micro DMA transfers can be used in both the master and the slave mode.

(1) Read/write micro DMA transfer

In this mode two micro DMA channels are used. One micro DMA channel is used to send the receive data from the SEDR register to the memory. The other micro DMA channel is used to send the new data from the memory to the SEDR register. The data transfer will be completely handled by the micro DMA controller.

① Initiation

In this mode, set < TMSE > bit of the SESR register to 1 and set it to micro DMA mode. Two micro DMA channels have to be set up for the transfer. One micro DMA is triggered on the INTSER to transfer the value that was received from the SEDR register to the memory. The other micro DMA is triggered on the INTSET to write new data from the memory to the SEDR register. Restart transfer by this setting in the master mode.

The micro DMA with the lower channel has to be assigned to the INTSER interrupt since it takes precedence over the micro DMA with the higher channel number.

The micro DMA transfer is initiated the first time by writing the first transfer value to the SEDR register. The following transfers will be handled automatically by the micro DMA controller.

Table 3.11.4 SEI setting when micro DMA transfer (read/write)

<SEE>	<MSTR>	<TASM>	<TMSE>
1	0:Slave	INTSEE interrupt mask	1
	1:Master	0	

② Micro DMA transfer

Once initiated the micro DMA wait to be triggered by a completed transfer. On a completed transfer both <TSRC> and <TSTC> flags are set to 1 and both SEI receive completed interrupt pulse INTSER and SEI transmit completed interrupt pulse INTSET are generated. Since the micro DMA channel with the lower channel number takes precedence, the read micro DMA transfer is performed before the write micro DMA transfer. The read micro DMA reads the value from the SEDR register and stores the value at the location specified within the micro DMA control registers. The read access also clears the <TSRC> flag to 0 in effect. After this the write micro DMA transfers a value from a specified memory address to the SEDR register. The write access to the SEDR register automatically clears the <TSTC> flag in the SESR register to 0 and starts a new transfer when the module is in master mode. After each micro DMA transfer the count registers for both micro DMA are decreased. This procedure continues until the counters reach the value of “0”. A micro DMA interrupt will be generated to indicate the end of the micro DMA transfer. An interrupt service routine triggered on the end of the micro DMA transfer can be used to re-initiate the micro DMA transfers.

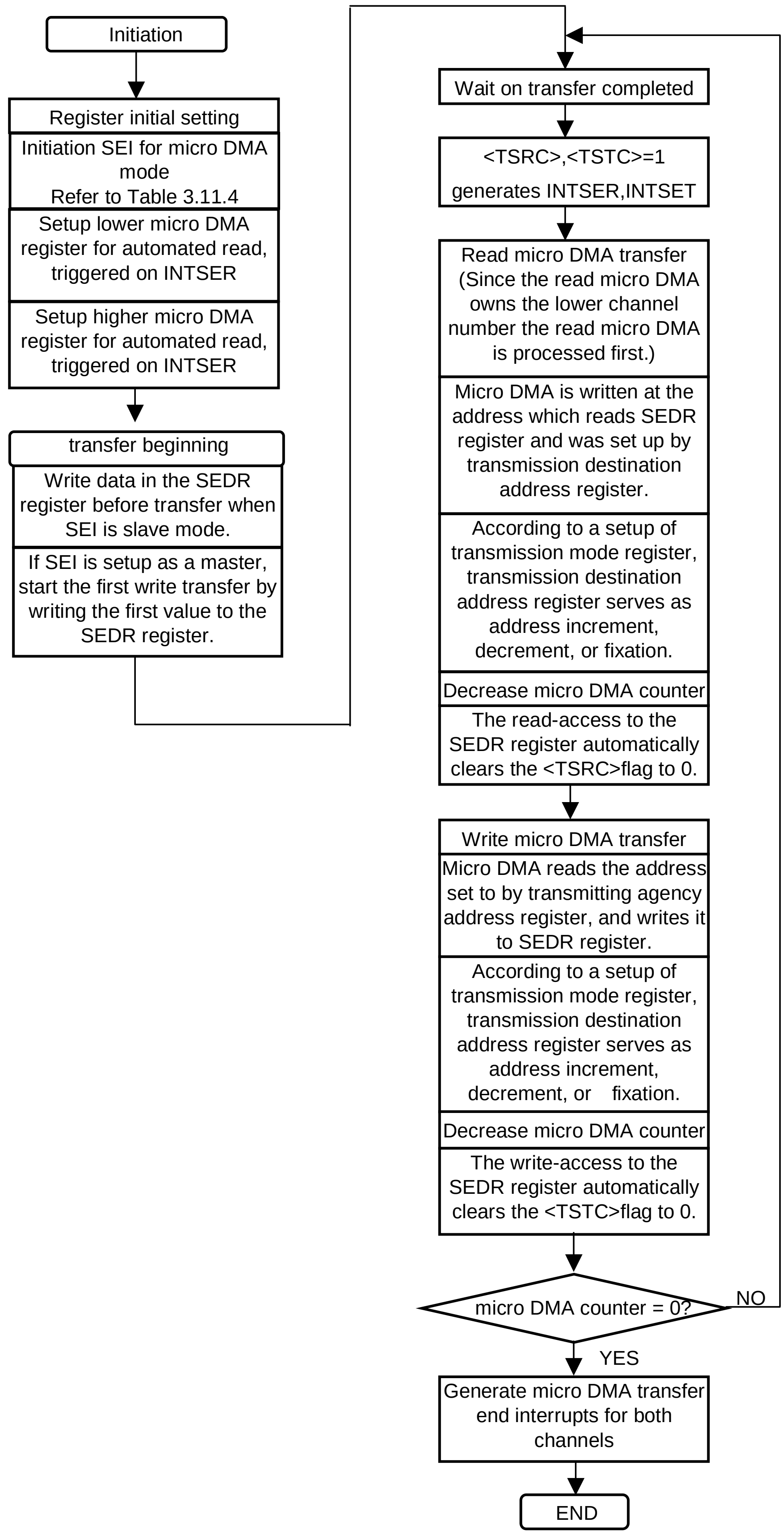


Figure 3.11.6 Flowchart for Micro DMA Read/Write Transfer

(2) Read only micro DMA transfer

This mode is used to shift in lager blocks of data, while “don’t care data” is shifted out (e.g.: reads from serial EEPROM). Only a single micro DMA is used to store the data read from the SEDR register to a specified RAM area.

① Initiation

For this mode the SEI has to be configured for micro DMA mode by setting the SESR<TMSE> to 1. When SEI is acting as master, the <TASM> bit has to be set additionally to allow the automated shifting. Just one micro DMA has to be set up to transfer the SEDR data to a memory location specified within the micro DMA destination address register. The SEI receive completion interrupt INTSER is used to trigger this micro DMA. The SEI transfer completion interrupt INTSET is disabled at the interrupt controller. If SEI is set up as a master, the first transfer has to be initiated by writing the SEDR register.

Table 3.11.5 SEI setting when micro DMA transfer (read)

<SEE>	<MSTR>	<TASM>	<TMSE>
1	0: Slave	INTSEE interrupt mask	1
	1: Master	1	

② Micro DMA transfer

After initiating the first transfer, the micro DMA waits for the transfer to be completed. With the completion of the transfer both the SESR<TSRC> and SESR<TSTC> are set to 1. On setting the <TSRC> to 1, the INTSER interrupt is generated to trigger the micro DMA. The <TSTC> flag will be set to 1 simultaneously and will remain set to 1 till the end of the block transfer.

The micro DMA moves the received value from the SEDR register to the memory location specified in its destination address register. After the micro DMA transfer, the count register of the micro DMA is decreased.

When the SEDR register is read, the SEDR register (shift register) is cleared to "00H" automatically because <TASM>bit is 1. Simultaneously a new transfer is started automatically. This procedure will repeat until the micro DMA counter reaches a value of “0”. A micro DMA interrupt will be generated to indicate the end of the micro DMA transfer.

Moreover, about the <TSTC> flag, it remains set to 1 after the first transmission end, unless it is reset.

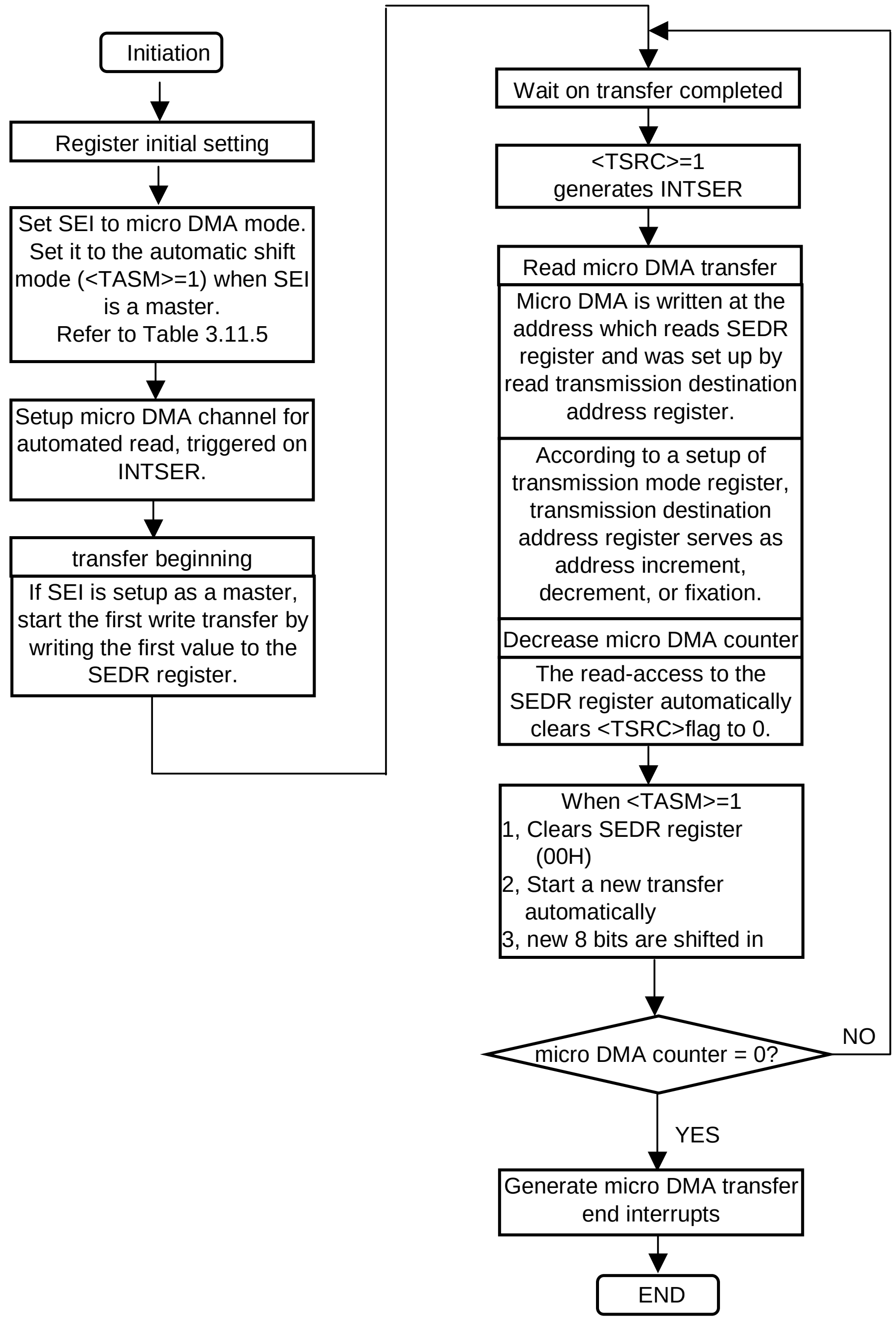


Figure 3.11.7 Flowchart for Micro DMA Read only Transfer

(3) Write only micro DMA transfer

The write only transfer mode is used to transmit larger blocks of data while the incoming data is ignored. Only a single micro DMA is used to transfer new transmit data from a memory location specified by the micro DMA source address register to the SEDR register.

① Initiation

For this mode the module has to be configured for micro DMA mode by setting the SESR<TMSE> to 1. One of the micro DMA channels has to be set up for the automated write to the SEDR register. This micro DMA is triggered by the SEI transmit completion interrupt INTSET. The SEI receive completion interrupt INTSER is disabled at the interrupt controller. If SEI is set up as a master, the first transfer is initiated by writing the first value to the SEDR register.

Table 3.11.6 SEI setting when micro DMA transfer (write)

<SEE>	<MSTR>	<TASM>	<TMSE>
1	0: Slave	INTSEE interrupt mask	1
	1: Master	0	

② Micro DMA transfer

After starting the first transfer the micro DMA waits for the transfer to be completed. On completion both the <TSRC> and <TSTC> flags in the SEI status register are set to 1. Disregard <TSRC>flag and <SOVF>flag because reception is not used.

After the first transmission end, the <TSRC> flag is set and it remains set to 1 unless it is reset. Once the <SOVF> flag is set to 1, the <SOVF> flag remains being 1, unless it is reset.

The <TSTC> flag generates the INTSET interrupt, which will trigger the micro DMA transfer.

The micro DMA reads a value from the memory address specified in its source register and transfers it to the SEDR register. After the micro DMA transfer, the count register of the micro DMA is decreased. The write access to the SEDR register clears the <TSTC> flag to 0 and starts a new transfer on the SEI bus when the module is in master mode. This procedure continues until the Micro DMA counter reaches a value of “0”. A micro DMA interrupt will be generated to indicate the end of the micro DMA transfer.

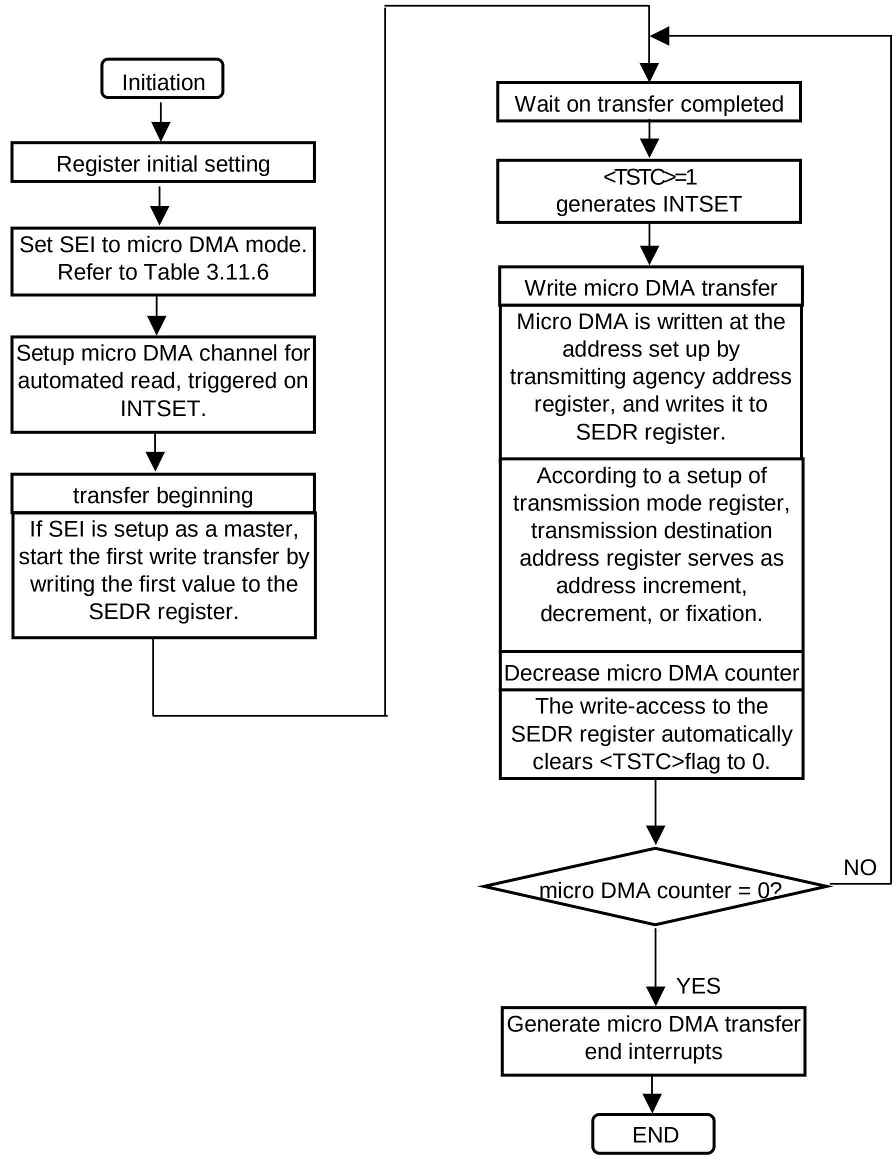


Figure 3.11.8 Flowchart for Micro DMA Write only Transfer

3.12 CAN Controller

(1) Overview

- Supports CAN version 2.0B
- Supports standard format and extended format
- Supports data frames and remote frames in both format
- 16 Mailboxes (15 Receive & Transmit + 1 Receive only)
- Baud rate up to 1Mbps on the CAN bus (at fc = 20MHz)
- Programmable baud rate with bit time parameter
- Built in baud rate prescaler
- 2 selectable mechanism for internal arbitration of transmit messages
  - ① mailbox number
  - ② identifier priority
- Time stamp for receive and transmit messages
- Operation modes
  - ① Normal operation mode
  - ② Configuration mode
  - ③ Sleep mode (Wake up on CAN bus activity or CPU access)
  - ④ Halt mode
  - ⑤ Test loop back mode (Enable the stand alone operation by self acknowledge)
  - ⑥ Test error mode (Write enable to error counter)
- Acceptance filter
  - ① Programmable global mask for mailboxes 0 to 14
  - ② Programmable local mask for mailbox 15
- Acceptance mask bit for identifier extended bit
- Flexible interrupt structure (3 interrupts)
  - ① INTCR: Receive interrupt
  - ② INTCT: Transmit interrupt
  - ③ INTCG: Global interrupt (include warning level, error passive, bus off, and so on)

(2) Nomenclature

- R/W Read and write access by the CPU
- R Read access by the CPU
- W Write access by the CPU
- R/S Read access and set (write with 1) by the CPU
- R/C Read access and clear (write with 1) by the CPU
- The bit Symbol " ╲ " in the mailbox denotes blank bits. The values of these bits are indeterminate when read.
- The column of after Reset " – " in the mailbox indicates that the initial value is indeterminate.
- The bit Symbol " ╲ " in the control register denotes reserved bits. They indicate that the value is indeterminate when read. Always write "0" when write.

(3) Architecture

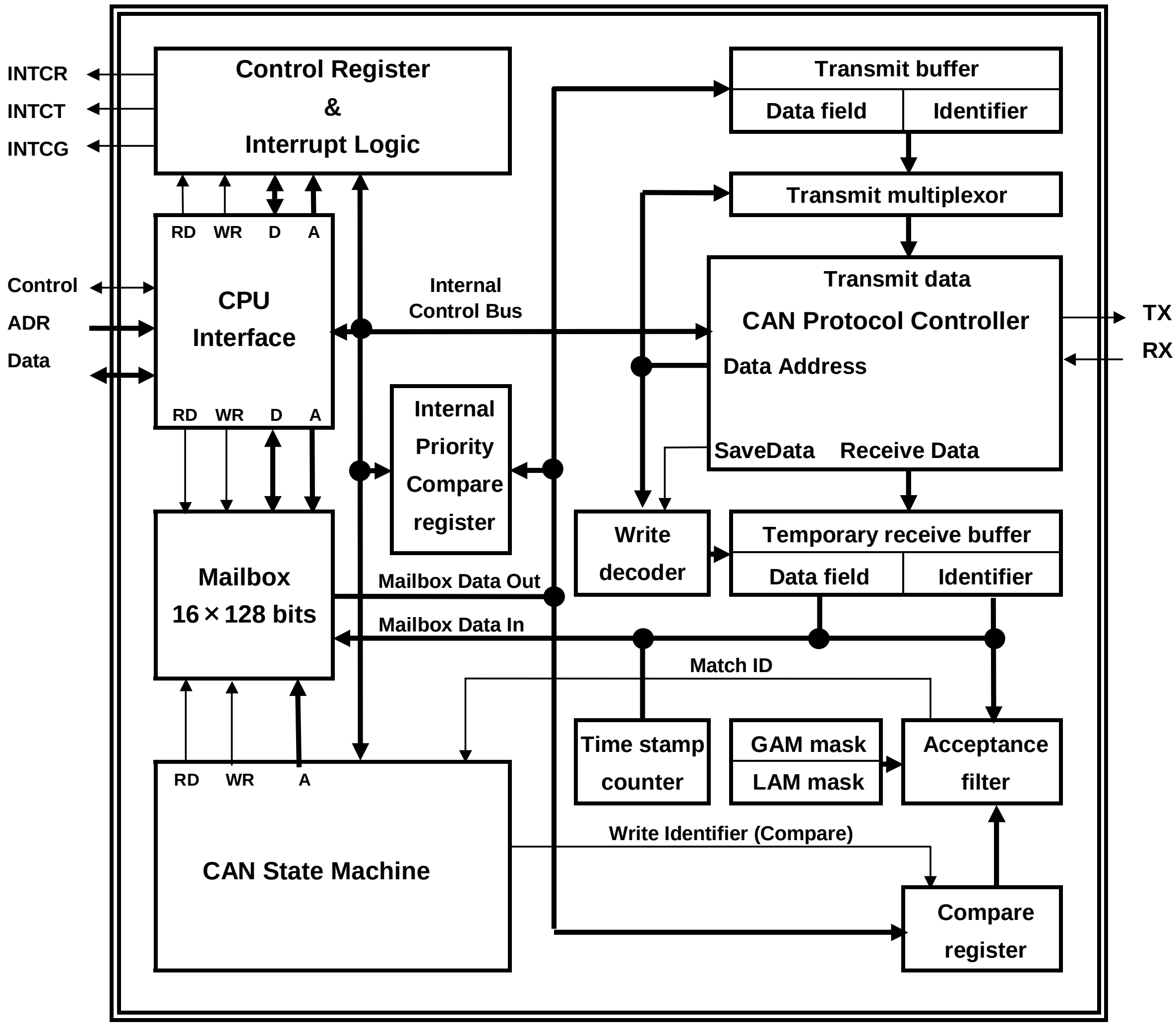


Figure 3.12.1 Block Diagram of CAN Controller

(4) CAN bus interface

The interface to the Can bus is a simple two-wire line, consisting of an input pin RX and an output pin TX. This CAN bus interface is suitable for the operation with CAN bus transceivers based on ISO/DIS 11898.

3.12.1 Memory map

The mailboxes and control registers used by the CAN are mapped to the memory locations shown below.

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: * Read-modify-write prohibited.

3.12.2 Mailboxes

The mailbox is configured with Register to store identifiers and transmit/receive data, which can be accessed by the CAN controller and the CPU. The CPU controls the CAN controller by modifying the contents of the mailboxes and control registers. The contents of the mailboxes and control registers are used to perform the functions of the acceptance filtering, transmit message and interrupt handling.

In order to initiate a transfer, the transmission request bit has to be written to the corresponding register. The entire transmission procedure is done then without any CPU involvement. If a mailbox has been configured as receive messages the CPU easily reads its data registers using CPU read instructions. The mailbox may be configured to interrupt the CPU after every successful message transmission or reception.

The mailbox module provides 16 mailboxes, each of which has 8 bytes long data, 29-bit identifier and several control bits. Each mailbox, except the last one, can be set for either transmit or receive operation. Mailbox 15 is a receive-only mailbox with a special acceptance mask designed to allow groups of different message identifiers to be received. Each mailbox is 16 bytes in size.

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)

Each mailbox is configures as shown below.

(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

The components of each mailbox are explained in the next pages.

Message Identifier Field 0 (MI0)

Message Identifier Field 0 Low

MBnMI0L (MBn+00H)		7	6	5	4	3	2	1	0
	bit Symbol	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
Read modify-write instructions prohibited.	Read/Write	R/W							
	After reset	-	-	-	-	-	-	-	-

Bits 7 to 0: Identifiers <ID23> to <ID16> are stored.

Message Identifier Field 0 High

MBnMI0H (MBn+01H)		15	14	13	12	11	10	9	8
	bit Symbol	IDE	GAME	RFH	ID28	ID27	ID26	ID25	ID24
Read modify-write instructions prohibited.	Read/Write	R/W							
	After reset	-	-	-	-	-	-	-	-

Bits 12 to 8: Identifiers <ID28> to <ID24> are stored.

RFH: 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.)

GAME: 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>.

IDE: 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.

The priority of a message ID becomes so high that 0 continues from the MSB (<ID28> bit) of ID.

Note: When ID of the received remote frame is corresponding to ID of the transmission mailbox <RFH>=1 and <GAME>=1, ID of remote frame is overwritten to this mailbox. Afterward, it responds applying overwritten ID automatically.

Message Identifier Field 1 (MI1)

Message Identifier Field 1 Low

MBnMI1L (MBn+02H)

Read modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
Read/Write	R/W							
After reset	-	-	-	-	-	-	-	-

Identifiers <ID7> to <ID0> are stored.

Message Identifier Field 1 High

MBnMI1H (MBn+03H)

Read modify-write instructions prohibited.

	15	14	13	12	11	10	9	8
bit Symbol	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
Read/Write	R/W							
After reset	-	-	-	-	-	-	-	-

Identifiers <ID15> to <ID8> are stored.

Note1: For standard format, identifiers <ID17> to <ID0> are indeterminate.

Note2 : Set the mailbox ID at initial configuration. When rewriting to MI0 or MI1 field of the mailbox which is permitted, after forbidding a mailbox by resetting the <MC> bit, and then carry out. However, reception is stopped, when it resets to <MC> =0, while a mailbox is receiving. When a mailbox is transmitting (<TRS>=1), after transmission is completed (<TRS>=0), please rewrite the MI0 or MI1 field.

Message Control Field (MCF)

Message Control Field Low

MBnMCFL
(MBn+04H)

Read modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol				RTR	DLC3	DLC2	DLC1	DLC0
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: Do not use data length codes other than those listed above.

Message Control Field High

MBnMCFH
(MBn+05H)

Read modify-write instructions prohibited.

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write								
After reset								

In the case of a receiving mailbox, there is no necessity for an initial configuration. RTR and DLC of the received message are stored in the MCF register. In the case of a transmitting mailbox, please set at the initial configuration.

Data field (D0 to D7)

This is a read/write register that stores up to 8 bytes of transmit/receive data. However, in the case of receive mailboxes, the write access to the data field is disabled.

For transmit, data in a length of bytes set by the mailbox's data length code is transmitted.

For receive, the data length code in the receive message is copied to the mailbox's data length code, so that the byte in a length equal to this data length code is receives as valid data.

Data Field 0

MBnD0 (MBn+06H)
Read modify-write instructions prohibited.

	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 instructions prohibited.

	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 instructions prohibited.

	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 instructions prohibited.

	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 instructions prohibited.

	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 instructions prohibited.

	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)
Read modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	D67	D66	D65	D64	D63	D62	D61	D60
Read/Write	R/W							
After reset	-	-	-	-	-	-	-	-

Data Field 7

MBnD7 (MBn+0DH)
Read modify-write instructions prohibited.

	15	14	13	12	11	10	9	8
bit Symbol	D77	D76	D75	D74	D73	D72	D71	D70
Read/Write	R/W							
After reset	-	-	-	-	-	-	-	-

Time Stamp Value (TSV)

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	-	-	-	-	-	-	-	-

This is a 16-bit read only register into which the value of the time stamp counter is loaded when data is successfully transmitted or received.

The counter value is not loaded this register when transmit or receive operation failed.

3.12.3 Control registers

Mailbox configuration register (MC)

Mailbox Configuration Register Low

		7	6	5	4	3	2	1	0
MCL (0300H)	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

		15	14	13	12	11	10	9	8
MCH (0301H)	bit Symbol	MC15	MC14	MC13	MC12	MC11	MC10	MC9	MC8
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0

Each bit corresponds to mailbox 0 through 15.

Each mailbox can be enabled or disabled.

When <MCn> = 0, access to mailbox “n” is disabled.

When <MCn> = 1, access to mailbox “n” is enabled.

Set the mailbox ID at the initial configuration. Before rewriting the mailbox’s MI0 or MI1 field, be sure to clear the <MC> bit to disable the corresponding mailbox. However, when <MC> bit is cleared to 0 during reception, the reception is stopped immediately. When a mailbox is transmitting (<TRS>=1), please rewrite the MI0 or MI1 field after transmission is completed (<TRS>=0).

The transmit mailbox data and control fields can be accessed for write at any time. However, in the case of transmit mailboxes with the <RFH> bit is set to 1, the write access to the message control field is enabled during the <MC> bit is cleared to 0.

Mailbox direction register (MD)

Mailbox Direction Register Low

		7	6	5	4	3	2	1	0
MDL (0302H)	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

		15	14	13	12	11	10	9	8
MDH (0303H)	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

Each bit corresponds to mailbox 0 through 15.

Each mailbox except mailbox 15 can be directed for transmit or receive.

When <MDn> = 0, the mailbox MBn is directed for transmit.

When <MDn> = 1, the mailbox MBn is directed for receive.

Mailbox 15 is a receive-only mailbox, so that <MD15> bit is fixed to “1”. This bit can only be read; you cannot write to it.

Set the MD register at initial configuration. When changing MD register, please carry out after clearing <MCn> bit of a corresponding mailbox.

(1) Transmit control registers

Transmission request set register (TRS)

Transmission Request Set Register Low

TRSL (0304H)

Read modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	TRS7	TRS6	TRS5	TRS4	TRS3	TRS2	TRS1	TRS0
Read/Write	R/S							
After reset	0	0	0	0	0	0	0	0

Transmission Request Set Register High

TRSH (0305H)

Read modify-write instructions prohibited.

	15	14	13	12	11	10	9	8
bit Symbol		TRS14	TRS13	TRS12	TRS11	TRS10	TRS9	TRS8
Read/Write		R/S						
After reset		0	0	0	0	0	0	0

Each bit corresponds to mailboxes 0 through 15. Since mailbox 15 is a receive-only mailbox, bit 15 is nonexistent.

If after writing data and identifier to mailbox "n" that has directed for transmit (<MDn> = 0) the <TRSn> bit is set to 1 when the said mailbox is enabled (<MCn> = 1), a message is transmitted from mailbox "n". If there are multiple transmit requests, messages are transmitted sequentially. The order in which messages are transmitted depends on the master control register MCR bit 3 <MTOS>.

If <MTOS> bit is clear to 0, the mailbox with the lower number has the higher priority. For example: if the mailboxes MB0, MB2 and MB5 are configured for transmission and the corresponding TRS bits are set to 1, then the messages will be transmitted in the following order: MB0, MB2 and MB5. If a new transmission request is set for MB0 during the processing of MB2 then in the next internal arbitration-run MB0 will be selected for the next transmission. This will also happen, when the CAN controller loses arbitration while transmitting MB2. In this case, MB0 will be sent at the next opportunity instead of MB2.

If <MTOS> bit is set to "1", the priority of the identifier stored in the mailbox will determine the sending order. The mailbox with the higher priority identifier will be sent first. In case of a lost arbitration on the CAN bus line a new internal arbitration run will be started and the message with the highest priority will be sent at next possible time.

The <TRSn> bit is reset when transmit has succeeded or when the transmission request concerned is cleared by setting the <TRRn> bit to 1.

If transmit has failed, transmit is retried repeatedly until it succeeds or the transmission request concerned is cleared by setting the <TRRn> bit to 1.

When the <TRSn> bit is "1", the write access to the corresponding mailbox is denied.

The <TRSn> bit cannot be set from the CPU if mailbox "n" is directed for receive.

When mailbox "n" is directed for transmit, the <TRSn> bit is set by writing a "1" from the CPU and is cleared to 0 by the internal logic. Writing a "0" from the CPU has no effect.

Transmission request reset register (TRR)

Transmission Request Reset Register Low

TRRL (0306H)

Read modify-write instructions prohibited.

	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

Transmission Request Reset Register High

TRRH (0307H)

Read modify-write instructions prohibited.

	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

Each bit corresponds to mailboxes 0 through 15. Since mailbox 15 is a receive-only mailbox, bit 15 is nonexistent.

If the <TRRn> bit is set to "1", the transmit request that has been asserted by setting the corresponding <TRSn> bit is canceled. This cancellation takes place in one of the following three ways:

① If a message has not been transmitted yet, the message transmit request is canceled.
(<TRSn> = 0, <TRRn> = 0, <AAn> = 1)

② If a message is currently being transmitted but a lost arbitration or an error occurs, the message transmit request is cleared and transmit operation is aborted.
(<TRSn> = 0, <TRRn> = 0, <AAn> = 1)

③ If a message is currently being transmitted and no lost arbitration or error occurs, transmit operation is completed without ever clearing the message transmit request.
(<TRSn> = 0, <TRRn> = 0, <TAn> = 1)

When the <TRRn> bit is "1", the write access to the corresponding mailbox is denied.
The <TRRn> bit cannot be set from the CPU if mailbox "n" is directed for receive.
When mailbox "n" is directed for transmit the <TRRn> bit is set by writing a "1" from the CPU and is cleared to 0 by the internal logic in case of a successful transmission or an aborted transmission. Writing a "0" from the CPU has no effect.

Transmission acknowledge register (TA)

Transmission Acknowledge Register Low

TAL (0308H)		7	6	5	4	3	2	1	0
	bit Symbol	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
Read modify-write instructions prohibited.	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Transmission Acknowledge Register High

TAH (0309H)		15	14	13	12	11	10	9	8
	bit Symbol		TA14	TA13	TA12	TA11	TA10	TA9	TA8
Read modify-write instructions prohibited.	Read/Write		R/C						
	After reset		0	0	0	0	0	0	0

Each bit corresponds to mailboxes 0 through 15. Since mailbox 15 is a receive-only mailbox, bit 15 is nonexistent.

The <TAn> bit is set when the message of mailbox “n” has been transmitted successfully. In this case, a transmission successful interrupt is generated if it has been enabled.

The <TAn> bit is cleared to 0 by writing a “1” to the <TAn> bit or the <TRSn> bit from the CPU. Writing a “0” from the CPU has no effect.

Abort acknowledge register (AA)

Abort Acknowledge Register Low

AAL (030AH)		7	6	5	4	3	2	1	0
	bit Symbol	AA7	AA6	AA5	AA4	AA3	AA2	AA1	AA0
Read modify-write instructions prohibited.	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Abort Acknowledge Register High

AAH (030BH)		15	14	13	12	11	10	9	8
	bit Symbol		AA14	AA13	AA12	AA11	AA10	AA9	AA8
Read modify-write instructions prohibited.	Read/Write		R/C						
	After reset		0	0	0	0	0	0	0

Each bit corresponds to mailboxes 0 through 15. Since mailbox 15 is a receive-only mailbox, bit 15 is nonexistent.

The <AAn> bit is set when the transmission of the message of mailbox “n” has been aborted. In this case, a global interrupt (transmit abort) is generated if it has been enabled.

The <AAn> bit is cleared to 0 by writing a “1” to the <AAn> bit or the <TRSn> bit from the CPU. Writing a “0” from the CPU has no effect.

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

Each bit corresponds to mailboxes 0 through 15. Since mailbox 15 is a receive-only mailbox, bit 15 is nonexistent.

If the <CDRn> bit is 1, a transmission request for mailbox "n" will be ignored. That means, that a mailbox "n" with the <TRSn> and <CDRn> bit is set to 1, it will not be considered in the internal arbitration-run: the mailbox "n" is locked for transmission. The processing of mailbox "n" in the arbitration-run will be considered again after clearing the <CDRn> bit.

The <CDR> bit is useful for dealing with remote frames. It is intended for updating the data field of a transmit mailbox, which is configured for automatic reply to remote frames (the <RFH> bit is set). By using the <CDR> bit, the user can update the data field without a need of taking additional care of the data consistency.

(2) Receive control registers

The identifier of each incoming message is compared with the identifiers held in the mailboxes that have been directed for receive operation. The comparison of the identifiers depends on the value of the global/local acceptance mask enable bits <GAME>/<LAME> in the mailbox and the data held in the global/local acceptance mask registers GAM/LAM.

When a matching identifier is detected, the received identifier, control bits, and data bytes are written to the mailbox that has matched. At this time, the corresponding receive message pending bit <RMPn> is set and receive successful interrupt is generated if it has been enabled. Once a matching identifier is found, no other identifiers are compared.

If not match is detected, the message is rejected.

The CPU must reset the <RMPn> bit after reading the data. If a second message is received for this mailbox when the <RMPn> bit has already been set to 1, the corresponding receive message lost bit <RMLn> is set to 1. In this case, the data stored in mailbox "n" is overwritten with the new data. In this case, a global interrupt (receive message lost) is generated if it has been enabled.

Receive-only mailbox

Only if the identifier of a received message does not match any identifiers of the mailboxes 0 through 14, the identifier is compared with the identifier of the receive-only mailbox 15. When a matching identifier is detected, the contents of the received message are written to the mailbox 15.

Receive message pending register (RMP)

Receive Message Pending Register Low

RMPL (030CH)

Read modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	RMP7	RMP6	RMP5	RMP4	RMP3	RMP2	RMP1	RMP0
Read/Write	R/C							
After reset	0	0	0	0	0	0	0	0

Receive Message Pending Register High

RMPH (030DH)

Read modify-write instructions prohibited.

	15	14	13	12	11	10	9	8
bit Symbol	RMP15	RMP14	RMP13	RMP12	RMP11	RMP10	RMP9	RMP8
Read/Write	R/C							
After reset	0	0	0	0	0	0	0	0

Each bit corresponds to mailbox 0 through 15.

When a message is received and its content is stored in mailbox "n", the <RMPn> bit is set to 1.

If a second message is received by mailbox "n" for which the <RMPn> bit has been set to 1, mailbox "n" is overwritten with the new data. In this case, the corresponding <RMLn> bit is set.

The <RMPn> bit is set to 1 by the internal logic and is cleared by writing a "1" to the <RMPn> bit from the CPU. The CPU cannot write a 0 to <RMPn> bit.

Receive message lost register (RML)

Receive 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 modify-write instructions prohibited.	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Receive 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 modify-write instructions prohibited.	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Each bit corresponds to mailbox 0 through 15.

If a second message is received by mailbox "n" for which the <RMPn> bit has been set to 1, mailbox "n" is overwritten with the new data and the <RMLn> bit is set to 1.

The <RMLn> bit is set by the internal logic and is cleared to 0 by writing a "1" to the <RMPn> bit from the CPU. Writing a "0" to <RMPn> bit and writing a "1" or "0" to <RMLn> bit from the CPU have no effect.

Table 3.12.2 Operation when message is received

ID	Before <RMP>	After <RMP>	After <RML>	Operation
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.)
Matched	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).

(3) Handling of remote frames

If a remote frame is received, it is compared with the identifiers of all mailboxes. The comparison of identifiers depends on the value of the global/local acceptance mask enable bits <GAME>/<LAME> in the mailbox and the data held in the global/local acceptance mask registers GAM/LAM.
If a received remote frame matches the identifier of a mailbox that is directed for transmit and the <RFH> bit for that mailbox is set to 1, the <TRSn> bit is set to 1 in order to send a message in response to the remote frame. Even when there is a matching identifier, if the <RFH> bit for that mailbox is reset (even though it may be a transmit mailbox), the remote frame is not responded to.
If there is a matching identifier and this mailbox is directed for receive, the remote frame is processed as data frame, in which case the <RMP> and <RFP> bits are set.
Once a matching identifier is found, no other identifiers are 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 : When <GAME> = 1 of this mail box, ID of remote frame is overwritten to this mailbox. and carries out an automatic response by new ID.

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 modify-write instructions prohibited.	Read/Write	R/C							
	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 modify-write instructions prohibited.	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

When a remote frame is received by mailbox “n” directed for receive the corresponding <RFPn> and <RMPn> bits are set. The <RFPn> bit is cleared to 0 by writing a “1” to the <RMPn> bit. Writing a “0” has no effect. Also, the <RFPn> bit is reset automatically when the remote frame received in mailbox “n” is overwritten by a newly received data frame.

(4) Acceptance filter

The global acceptance mask registers GAM0 and GAM1 are used for filtering messages when the <GAME> bit for mailboxes 0 through 14 is set to "1". An incoming message is stored in the first mailbox with a matching identifier. Only if there is no matching identifier in the mailboxes 0 to 14, the incoming message is compared with the mailbox 15, a receive-only mailbox. The local acceptance mask registers LAM0, LAM1 are used for filtering messages when the <LAME> bit for mailbox 15 is set.

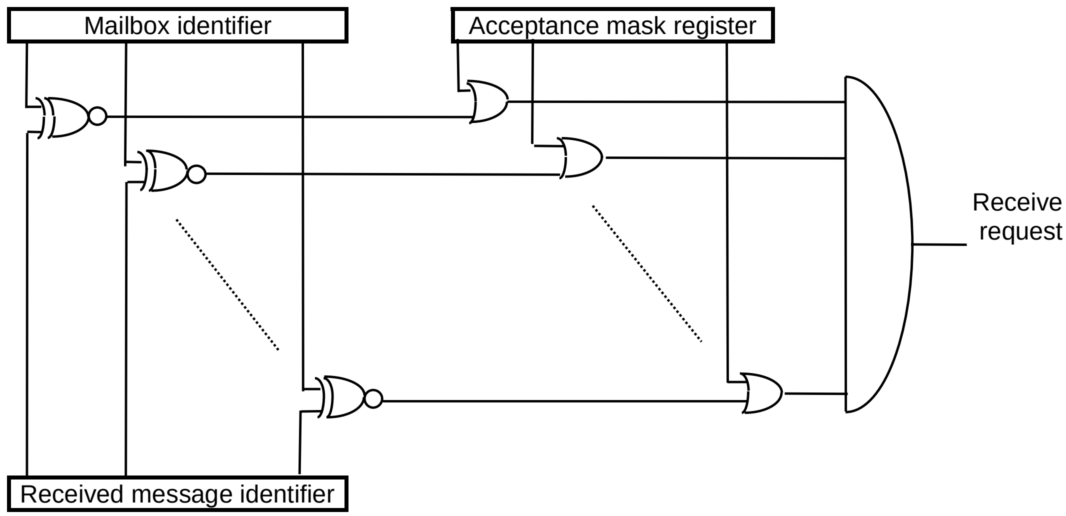


Figure 3.12.2 Acceptance Filter

Local acceptance mask registers (LAM0, LAM1)

Local Acceptance 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 Acceptance 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 Acceptance 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 Acceptance 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

The LAM0 and LAM1 registers are used for only filtering messages for mailbox 15. This feature allows the user to choose whether or not to locally mask any identifier bit of the incoming message for mailbox 15. Incoming messages are first checked to see if they match mailboxes 0 to 14 before being forwarded to mailbox 15.

If the <LAMn> bit is “0”, messages are received only when the corresponding bit of the incoming message identifier matches that of the mailbox identifier. If the <LAMn> bit is “1”, messages are received regardless of whether the corresponding bit of the incoming message identifier is “0” or “1”. The GAM0 and GAM1 registers do not affect mailbox 15.

For messages in extended format, the identifier extension <IDE> bit and the whole 29bits of the identifier are compared. For messages in standard format, only the <IDE> bit and the first 11bits of the identifier (<ID28> to <ID18>) are compared.

The <LAMI> bit (local acceptance mask identifier extension bit) is used to mask the <IDE> bit of mailbox 15.

If the <LAMI> bit is “0”, messages in extended or standard format are received according to the <IDE> bit of mailbox 15.

If the <LAMI> bit is “1”, messages in both extended and standard formats are received regardless of whether the <IDE> bit of mailbox 15 is “0” or “1”. For messages in extended format, the whole 29bits of the mailbox identifier and the whole 29 mask bits of the LAM register are used for filtering. For messages in standard format, only the first 11bits of the mailbox identifier (<ID28> to <ID18>) and the first 11 bits of the LAM register (<LAM28> to <LAM18>) are used for filtering.

Please perform the setup of LAM0 and LAM1 at initial configuration. Please do not change a setup during operation. When setting change is performed during reception, receiving message ID is compared for the receiving mask information in the middle of setting change.

Global acceptance mask registers (GAM0, GAM1)

Global Acceptance 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 Acceptance 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 Acceptance 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 Acceptance 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

The GAM0 and GAM1 registers are used for filtering messages for mailbox 0 to 14.

If the <GAME> bit for mailboxes 0 to 14 is set to 1 the GAM0 and GAM1 registers are used for incoming messages. A received message is stored in only the first mailbox with a matching identifier.

If the <GAMn> bit is "0", messages are received only when the corresponding bit of the incoming message identifier matches that of the mailbox identifier. If the <GAMn> bit is "1", messages are received regardless of whether the corresponding bit of the incoming message identifier is "0" or "1".

For messages in extended format, the identifier extension <IDE> bit and the whole 29bits of the identifier are compared. For messages in standard format, only the <IDE> bit and the first 11bits of the identifier (<ID28> to <ID18>) are compared.

The <GAMI> bit (global acceptance mask identifier extension bit) is used to mask the <IDE> bits of mailbox 0 to 14.

If the <GAMI> bit is "0", messages in extended or standard format are received according to the <IDE> bits of mailbox 0 to 14.

If the <GAMI> bit is "1", messages in both extended and standard formats are received regardless of whether the <IDE> bits of mailbox 0 to 14 are "0" or "1". For messages in extended format, the whole 29bits of the mailbox identifier and the whole 29 mask bits of the GAM register are used for filtering. For messages in standard format, only the first 11bits of the mailbox identifier (<ID28> to <ID18>) and the first 11 bits of the GAM register (<GAM28> to <GAM18>) are used for filtering.

Please perform the setup of GAM0 and GAM1 at initial configuration. Please do not change a setup during operation. When setting change is performed during reception, receiving message ID is compared for the receiving mask information in the middle of setting change.

(5) 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

TSTLB: Test Loop Back

0: Cancels the test loop back mode. (Normal operation)

1: Requests the test loop back mode.
This mode supports stand-alone operation.

TSTERR: Test Error

0: Cancels the test error mode. (Normal operation)

1: Requests the test error mode.
In this mode it is possible to write the error counter register CEC.

CCR: Change Configuration Request

0: Cancels the configuration mode. (Normal operation)

1: Request the configuration mode.
This mode allows for writing to the bit configuration registers BCR1, BCR2.

SMR: Sleep Mode Request

0: The sleep mode is not requested. (Normal operation)

1: Requests the sleep mode.
When this mode is entered, the CAN controller clock stops oscillating and the error counter and transmit requests are cleared.

HMR: Halt Mode Request

0: Cancels the halt mode. (Normal operation)

1: Requests the halt mode.
When this mode entered, the CAN controller does no longer transmit and receive messages. It only sends error and acknowledge flags.

WUBA: Wake Up on Bus Activity

0: Wakes up the module only by detecting a write access to the MCR register.

1: Wakes up the module when active bus state is detected or by detecting a write access to the MCR register.

MTOS: Mailbox Transmission Order Select

0: Mailbox transmission order by mailbox number. The mailbox with the lower number will be sent first.

1: Mailbox transmission order by identifier priority. The mailbox with the higher priority identifier will be sent first.

TSCC: Time Stamp Counter Clear

0: No effect

1: The time stamp counter will be cleared to 0.

Note 1: This is a write-only bit; it is always “0” when read.

Note 2: The time stamp counter is also cleared to 0 by a write to the TSP register, or writing a “0” to the TSC register.

SRES: Software Reset

0: No effect

1: Resets the CAN controller in software. All internal registers are initialized.

Note: This is a write-only bit; it is always “0” when read.

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> is the baud rate prescaler value. It can be set in the range of 0 to 255.

Bit Configuration Register 1 High

BCR1H (031DH)		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write								
	After reset								

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

The bit length is determined by parameters TSEG1, TSEG2, and BRP. All CAN controllers on the CAN bus must operate at the same baud rate. If individual CAN controllers operate with different frequencies the baud rate has to be adjusted by the mentioned parameters. In the bit timing logic, the conversion of the parameters to the required bit timing is materialized. The configuration registers BCR1 and BCR2 contains the data about the bit timing.

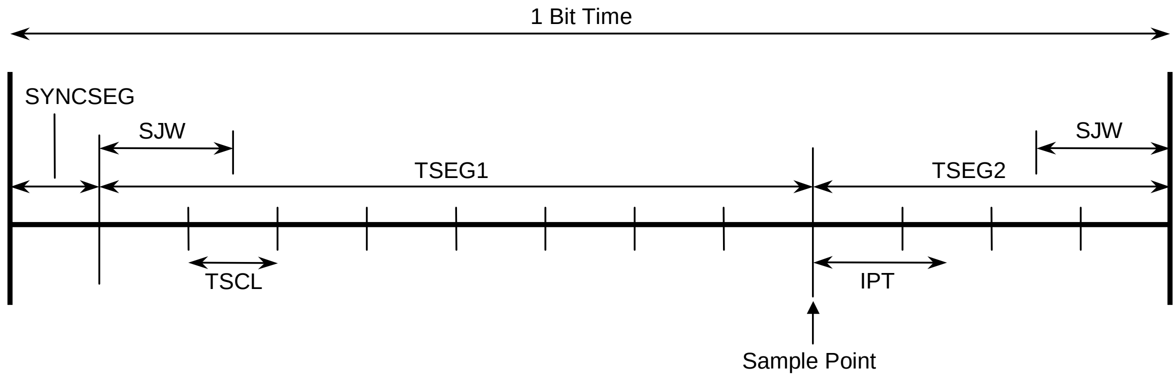


Figure 3.12.3 Bit Timing

The length of TSCL is defined by:

TSCL = (<BRP7:0>+1) / f_{IO} (f_{IO} = fc divided by 2)

f_{IO} is used to the CAN controller system clock frequency (input clock of the CAN controller).

The length of one bit is determined by the equation below:

1 Bit Time = SYNCSEG+TSEG1+TSEG2

1 bit time is equal or greater than 10 ÷ f_{IO}.

The synchronization segment SYNCSEG has always the length of 1 × TSCL.

The length of TSEG1 should be equal or greater than the length of TSEG2.

TSEG1 ≧ TSEG2.

The baud rate is defined by:

Baud rate = f_{IO} ÷ [(<BRP7:0>+1) × ((<TSEG13:10>+1)+(<TSEG22:20>+1)+1)]

IPT (information processing time) is the time segment starting with the sample point reserved for processing of the sampled bit level. IPT is equal to 3 f_{IO} clock cycles.

The parameter SJW (2bit) indicates by how many units of TSCL is allowed to be lengthened or to be shortened when re-synchronizing. Values between 1 (SJW = 00b) and 4 (SJW = 11b) are adjustable. The bus line is re-synchronized at each falling edge. The maximum length of SJW is equal to the length of TSEG2.

SJW ≦ TSEG2

With the corresponding bit timing, it is possible to reach a multiple sampling of the bus line at the sample point by setting <SAM> bit. The level determined by the CAN bus then corresponds to the result from the majority decision of the last three values. The three-time sampling is not allowed for <BRP7:0> < 4. For <BRP7:0> < 4 always a one-time sampling will be performed regardless of the value of <SAM> bit.

There is a restriction as follows:

<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:

A transmission rate of 1Mbps will be adjusted, i.e. a bit has a length of 1 μs. The CAN input clock frequency f_{IO} is 10MHz. The baud rate prescaler is set to "0". That means a bit for this data transmission rate has to be programmed with a length of 10 × TSCL. Since SYNCSEG is 1 × TSCL, it is set as 9 × TSCL by TSEG1+TSEG2.

E.g. <BRP7:0> = 00H
<TSEG13:10> = 0100B (5 × TSCL)
<TSEG22:20> = 011B (4 × TSCL)

In this case, sampling point is 60%. With this setting a threefold sampling of the bus is not possible (<BRP7:0> < 4), thus SAM = 0 should be set. SJW is not allowed to be greater than TSEG2, so the maximum value could be set to <SJW1:0> = 11B (4 × TSCL)

Example2:

Baud rate : 500kbps (1 bit time = 2 μs)
Sampling point : 80%
f_{IO} : 10MHz

(a) In case of <BRP7:0> = 00H
TSCL = (<BRP7:0>+1) / f_{IO} = 1 / 10MHz = 100ns
1 bit time = 2 μs / 100ns = 20 × TSCL
<TSEG13:10> = 1110B (15 × TSCL)
<TSEG22:20> = 011B (4 × TSCL)

(b) In case of <BRP7:0> = 01H
TSCL = (<BRP7:0>+1) / f_{IO} = 2 / 10MHz = 200ns
1 bit time = 2 μs / 200ns = 10 × TSCL
<TSEG13:10> = 0110B (7 × TSCL)
<TSEG22:20> = 001B (2 × TSCL)

Example3:

Baud rate : 500kbps (1 bit time = 2 μs)
Sampling point : 85%
f_{IO} : 10MHz

(a) In case of <BRP7:0> = 00H
TSCL = (<BRP7:0>+1) / f_{IO} = 1 / 10MHz = 100ns
1 bit time = 2 μs / 100ns = 20 × TSCL
<TSEG13:10> = 1111B (16 × TSCL)
<TSEG22:20> = 010B (3 × TSCL)

(b) In case of <BRP BRP7:0> = 01H
TSCL = (<BRP7:0>+1) / f_{IO} = 2 / 10MHz = 200ns
1 bit time = 2 μs / 200ns = 10 × TSCL
In this case, 85% sampling point cannot be set up.

Time stamp feature

There is a free-running 16-bit time stamp counter TSC implemented in the CAN controller to get an indication of the time of reception or transmission of messages. The content of the TSC is written into the time stamp value TSV of the corresponding mailbox when a received message has been stored or a message has been transmitted.

The TSC is driven from the bit clock of the CAN bus line. When the CAN controller is in configuration mode or sleep mode, the TSC will be stopped. After a reset, the TSC can be cleared to 0 by writing a value to the time stamp counter prescaler TSP. The TSC can be written and read by CPU in configuration mode and in normal operation mode.

Time stamp counter register (TSC)

Time Stamp Counter Register Low

TSCL (0332H)

Read modify-write instructions prohibited.

	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 instructions prohibited.

	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

Overflow of the TSC can be detected by the time stamp counter overflow flag <TSO> of the global status register GSR and the time stamp counter overflow interrupt flag <TSOIF> of the global interrupt flag register GIF. Both flags are cleared to 0 by writing a “1” to the corresponding bit location in GIF.

There is a 4-bit prescaler for the TSC. It is the time stamp counter prescaler register TSP that stores the value to be reloaded into this prescaler. After reset, the TSP register is cleared to “0”, so a value “0” is loaded into the prescaler. The TSC counter’s count-up period, TTSC, is shown below:

TTSC = TBIT × (<TSP3:0>+1) (TBIT : bit cycle)

Time stamp 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								

To be sure, that the value of the TSC will not change during the write cycle to the mailbox, there is a hold register implemented. The value of the TSC will be copied to this register if a message has been received or transmitted successfully. The reception is successful for the receiver, if there is no error until the last but one bit of End-of-frame. The transmission is successful for the transmitter, if there is no error until the last bit of End-of-frame. (Refer to the CAN version 2.0B)

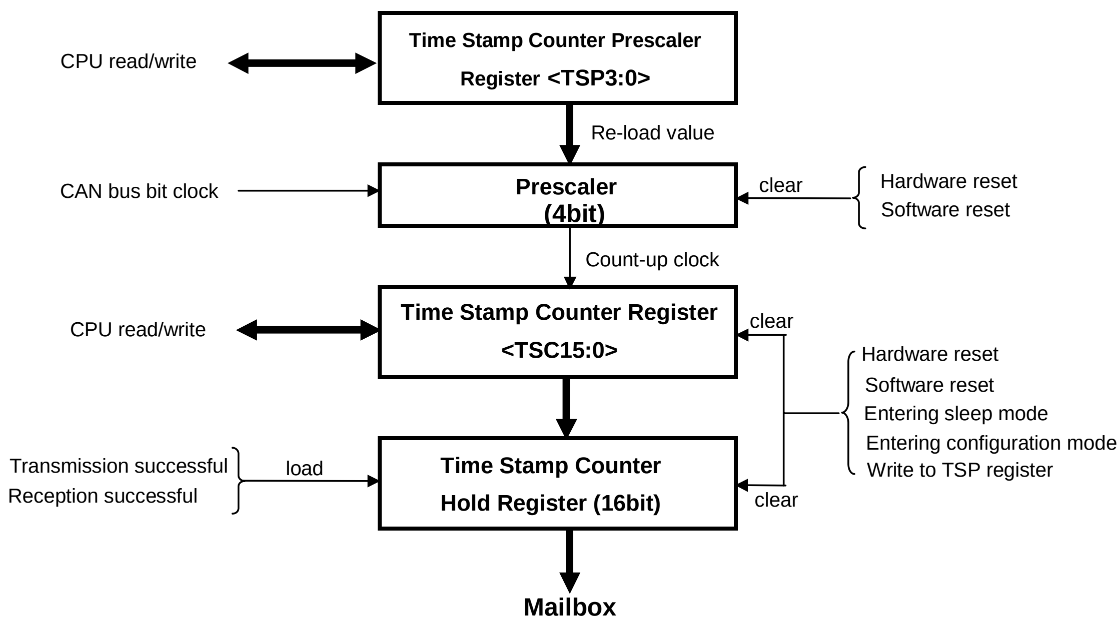


Figure 3.10.4 Time Stamp Counter

The free running time stamp counter and the time stamp hold register will be cleared in the following cases:

- After reset (hardware reset or software reset)
- When the module enters configuration mode
- When the module enters sleep mode
- When a write access to the time stamp prescaler register is performed

(6) 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		

MsgInSlot: Message In Slot

Indicates a message in the transmission slot.

0000: Message of mailbox 0

0001: Message of mailbox 1

:

1110: Message of mailbox 14

1111: No transmission message

RM: Receive Mode

0: The CAN controller is not receiving a message.

1: The CAN controller is receiving a message. That means the CAN controller is not the transmission of the message and the bus is not idle.

TM: Transmit Mode

0: The CAN controller is not transmitting a message.

1: The CAN controller is transmitting a message. That means the CAN controller stays transmitter until the bus is idle or it loses arbitration.

CCE: Change Configuration Enable

0: The CAN controller is not in the configuration mode. (Normal operation)

1: The CAN controller has entered the configuration mode.

SMA: Sleep Mode Acknowledge

0: The CAN controller is not in the sleep mode. (Normal operation)

1: The CAN controller has entered the sleep mode.

HMA: Halt Mode Acknowledge

0: The CAN controller is not in the halt mode. (Normal operation)

1: The CAN controller has entered the halt mode.

TSO: Time Stamp Overflow Flag

0: There was no overflow of the time stamp counter.

1: There was at least one overflow of the time stamp counter since this bit has been cleared to 0. To clear this bit, clear the <TSOIF> bit to 0 in the GIF register.

BO: Bus Off Status

0: The CAN controller is in the bus on status. (Normal operation)

1: The CAN controller is in the bus off status.

There is an abnormal rate of occurrences of errors on the CAN bus. This condition occurs when the transmit error counter TEC has reached the limit of 256. During bus off no messages can be received or transmitted. The CAN controller will go to bus on automatically after the bus off recovery sequence. After entering bus off, the error counters are undefined.

EP: Error Passive Status

0: The CAN controller is in the error active mode. Both values of the error counters TEC and REC are less than 128.

1: The CAN controller is in the error passive mode. At least one of the error counters has reached the error passive status of 128.

EW: Warning Status

0: Both values of the error counters TEC and REC are less than or equal to 96.

1: At least one of the error counters is greater than 96 and reached the warning level.

CAN error counter register (CEC)

CAN Error Counter Register Low

CECL (032EH)

Read modify-write instructions prohibited.

	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 instructions prohibited.

	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

The CAN controller contains two error counters: receive error counter REC and transmit error counter TEC. The values of both counters can be read by the CPU. A write access to the error counters is only in the test error mode possible at the same time with the same value of lower 8bit. (<TSTERR> bit in MCR register is set). These error counters are incremented or decremented according to the CAN version 2.0B.

A controller takes the following three states according to the value of REC and TEC.

(1) Error active state (TEC < 128 and REC < 128)

The state where the error has hardly occurred

CAN controller is in an error active state after reset release.

When an error is detected, an active error flag is transmitted.

(2) Error passive state (TEC >= 128 or REC >=128)

The state where many errors have occurred

When an error is detected, a passive error flag is transmitted.

(3) Bus off state (TEC >=256)

CAN controller cannot perform the message transmission and reception to CAN bus.

Receive error counter REC is not incremented after exceeding the error passive limit (128). After the correct reception of a message when REC = 128, the counter is set to a value between 119 and 127. After reaching the bus off status, the counts are undefined.

CAN controller which changed to the bus off state will be in an error active state automatically, if the 11 continuous recessive bits on the CAN bus is detected 128 times. All internal flags are reset, and the error counters are cleared. The configuration registers keep the programmed values. The values of the error counters are undefined during bus off status.
When CAN controller enters configuration mode (see 3.12.4(1) Configuration mode) the error counters will be cleared to 0.

(7) Interrupt control registers

The CAN controller has the following interrupt sources:

- Transmit interrupt
  When a message has been transmitted successfully
- Receive interrupt
  When a message has been received successfully
- Remote frame pending interrupt
  When a remote frame is received
- Wake-up interrupt
  When the CAN controller is awakened from sleep mode
- Receive message lost interrupt
  When a receive message is lost
- Transmission abort interrupt
  When at least one of the bits in the AA register is set to 1
- Time stamp counter overflow interrupt
  When the time stamp counter has overflowed
- Bus off interrupt
  When the CAN controller enters the bus off mode
- Error passive interrupt
  When the CAN controller enters the error passive mode
- Warning level interrupt
  When at least one of the two error counters is greater than 96 and reached the warning level

These interrupt sources are divided in three groups:

- Transmit interrupt (INTCT)
- Receive interrupt (INTCR)
- Global interrupt (INTCG)

There is one interrupt output line for each group. INTCR is dedicated for receive interrupts, INTCT is dedicated for transmit interrupts and INTCG for the global interrupts.

Global interrupt flag register (GIF)

Global Interrupt Flag Register Low

GIFL (0320H)

Read modify-write instructions prohibited.

		7	6	5	4	3	2	1	0
GIFL (0320H)	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 instructions prohibited.

		15	14	13	12	11	10	9	8
GIFH (0321H)	bit Symbol								
	Read/Write								
	After reset								

The interrupt flag bits will be set to 1 if the corresponding interrupt condition has occurred. If the corresponding interrupt mask bit is set to 1 in the GIM register, an interrupt pulse on the global interrupt line INTCG will be generated. As long as an interrupt flag in the GIF register is set to 1, if the corresponding interrupt source generates a new interrupt event, a new interrupt pulse on INTCG will not be generated. If an interrupt flag in the GIF register is set and another interrupt source generates an interrupt event, then a new interrupt pulse on INTCG will be generated.

If one or more interrupt flags have been cleared to 0 and one or more interrupt flags are still set to 1, a new global interrupt pulse INTCG will be generated.

The interrupt flags will be cleared to 0 by writing a "1" to the corresponding bit location.

RFPF: Remote Frame Pending Flag
- 0: No remote frame has been received.
- 1: A remote frame has been received (in a receive-mailbox).
  This bit will not be set if the identifier of the remote frame matches to a transmit-mailbox with <RFH> set to 1.

WUIF: Wake-Up Interrupt Flag
- 0: The CAN controller is in the sleep mode or the normal operation mode.
- 1: The CAN controller has left the sleep mode.

RMLIF: Receive Message Lost Interrupt Flag
- 0: No receive message has been lost.
- 1: At least one of the receive-mailboxes, receive message lost has been occurred. At least one of the bits in the RML register is set to 1.

TRMABF: Transmission Abort Flag
- 0: No transmission has been aborted.
- 1: Transmission has been aborted.
  At least one of the bits in the AA register is set to 1.

TSOIF: Time Stamp Counter Overflow Interrupt Flag
- 0: There was no overflows of the time stamp counter since this bit has been cleared.
- 1: There was at least one overflow of the time stamp counter since this bit has been cleared.

BOIF: Bus Off Interrupt Flag

0: The CAN controller is still in the bus on mode.

1: The CAN controller has entered the bus off mode.

EPIF: Error Passive Interrupt Flag

0: The CAN controller is still in error active mode.

1: The CAN controller has entered the error passive mode.

WLIF: Warning Level Interrupt Flag

0: none of the error counters has reached the warning level.

1: At least one of the error counters has reached the 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								

Each interrupt flag bits in GIF register is masked by the corresponding mask bit in GIM register.

If a bit in GIM register is 0, the interrupt generation for the corresponding global interrupt event is disabled and if it is 1, the interrupt generation is enabled. After reset, all bits in GIM register are cleared to 0, there by disabling global interrupt.

Mailbox interrupts

Separare interrupt outputs are provided for mailbox interrupts independently of global interrupts. These include mailbox transmit interrupt INTCT and mailbox receive interrupt INTCR that depend on mailbox settings. A mailbox transmit interrupt flag register MBTIF is provided for mailbox transmit interrupts, and a mailbox receive interrupt flag register MBRIF is provided for mailbox receive interrupts. In addition, there is a mailbox interrupt mask register MBIM that enables or disables 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

Each bit corresponds to mailboxes 0 through 15.
The MBIM register settings determine to enable or disable each mailbox interrupt.
If a bit in MBIM register is “0”, the interrupt generation for the corresponding mailbox is disabled.
If a bit in MBIM register is “1”, the interrupt generation for the corresponding mailbox is enabled.

Mailbox transmit interrupt flag register (MBTIF)

Mailbox Transmit Interrupt Flag Register Low

MBTIFL (0324H)

Read modify-write instructions prohibited.

		7	6	5	4	3	2	1	0
MBTIFL (0324H)	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 instructions prohibited.

		15	14	13	12	11	10	9	8
MBTIFH (0325H)	bit Symbol		MBTIF14	MBTIF13	MBTIF12	MBTIF11	MBTIF10	MBTIF9	MBTIF8
	Read/Write		R/C						
	After reset		0	0	0	0	0	0	0

This register is provided for mailbox transmit interrupts. Each bit in this register corresponds to mailboxes 0 through 15. The interrupt flag for mailbox 15, the <MBTIF15> flag, is nonexistent because mailbox 15 is the receive-only mailbox. If mailbox “n” is directed for receive, the corresponding interrupt flag in this register, the <MBTIFn> flag, will always be read as “0”.

If a message in mailbox “n” has been transmitted successfully and the mask bit <MBIMn> is set to “1”, the corresponding transmit interrupt flag <MBTIFn> will be set. If no other bit was set before in MBTIF register, transmit interrupt pulse INTCT will be generated.

If for a mailbox the mask bit in MBIM register is “0”, the transmit interrupt flag in MBTIF register will not be set and no transmit interrupt pulse INTCT will be generated. The information about a successful transmission could be read from the TA register respectively.

If one or more transmit interrupt flags have been set in MBTIF register and another interrupt condition has been occurred no interrupt will be generated, but the corresponding flag in MBTIF register will be set.

If there is one or more transmit interrupt flags set after clearing one or more transmit interrupt flags, another mailbox transmit interrupt pulse INTCT will be generated.

The interrupt flags in MBTIF register will be cleared by writing a “1” from the CPU to MBTIF register. Writing a “0” has no effect.

Note that the interrupt flags in MBTIF register is checked to be 1 (active), before doing a clear-access.

Mailbox receive interrupt flag register (MBRIF)

Mailbox Receive Interrupt Flag Register Low

MBRIFL (0326H)		7	6	5	4	3	2	1	0
	bit Symbol	MBRIF7	MBRIF6	MBRIF5	MBRIF4	MBRIF3	MBRIF2	MBRIF1	MBRIF0
Read modify-write instructions prohibited.	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

Mailbox Receive Interrupt Flag Register High

MBRIFH (0327H)		15	14	13	12	11	10	9	8
	bit Symbol	MBRIF15	MBRIF14	MBRIF13	MBRIF12	MBRIF11	MBRIF10	MBRIF9	MBRIF8
Read modify-write instructions prohibited.	Read/Write	R/C							
	After reset	0	0	0	0	0	0	0	0

This register is provided for mailbox receive interrupts. Each bit in this register corresponds to mailboxes 0 through 15. If mailbox “n” is directed for transmit, the corresponding interrupt flag in this register, the <MBRIFn> flag, will always be read as “0”.

If a message in mailbox “n” has been received successfully and the mask bit <MBIMn> is set to “1”, the corresponding receive interrupt flag <MBRIFn> will be set. If no other bit was set before in MBRIF register, receive interrupt pulse INTCR will be generated.

If for a mailbox the mask bit in MBIM register is 0, the receive interrupt flag in MBRIF register will not be set and no receive interrupt pulse INTCR will be generated. The information about a successful reception could be read from the RMP register respectively.

If one or more receive interrupt flags have been set in MBRIF register and another interrupt condition has been occurred no interrupt will be generated, but the corresponding flag in MBRIF register will be set.

If there is one or more receive interrupt flags set after clearing one or more receive interrupt flags, another mailbox receive interrupt pulse INTCR will be generated.

The interrupt flags in MBRIF register will be cleared by writing a “1” from the CPU to MBRIF register. Writing a “0” has no effect.

Note that the interrupt flags in MBRIF register is checked to be 1 (active), before doing a clear-access.

3.12.4 Description of Mode

(1) Configuration mode

The CAN controller has to be initialized (set the bit configuration registers BCR1 and BDR2) before the activation. The BCR1 and BCR2 registers can only be modified when the module is in the configuration mode. After reset, the configuration mode is active and the <CCR> bit of MCR register and the <CCE> bit of GSR register are set to "1". The CAN controller can be set to the normal operation mode by writing a "0" to <CCR> bit. After leaving the configuration mode, the <CCE> bit will be set to "0" and the power-up sequence will start. The power-up sequence consists of detecting eleven consecutive recessive bits on the CAN bus line. After the power-up sequence, the CAN controller is bus on and ready for operation.

When the <CCR> bit is set to "1", the CAN controller will be entered to the configuration mode from the normal operation mode. After the CAN controller has entered the configuration mode, the <CCE> bit will be set to "1". See also the flowchart in Figure 3.12.5 Flowchart of CAN Initialization. When at the configuration mode, the error counter CEC, the time stamp counter TSC and the time stamp hold register will be cleared.

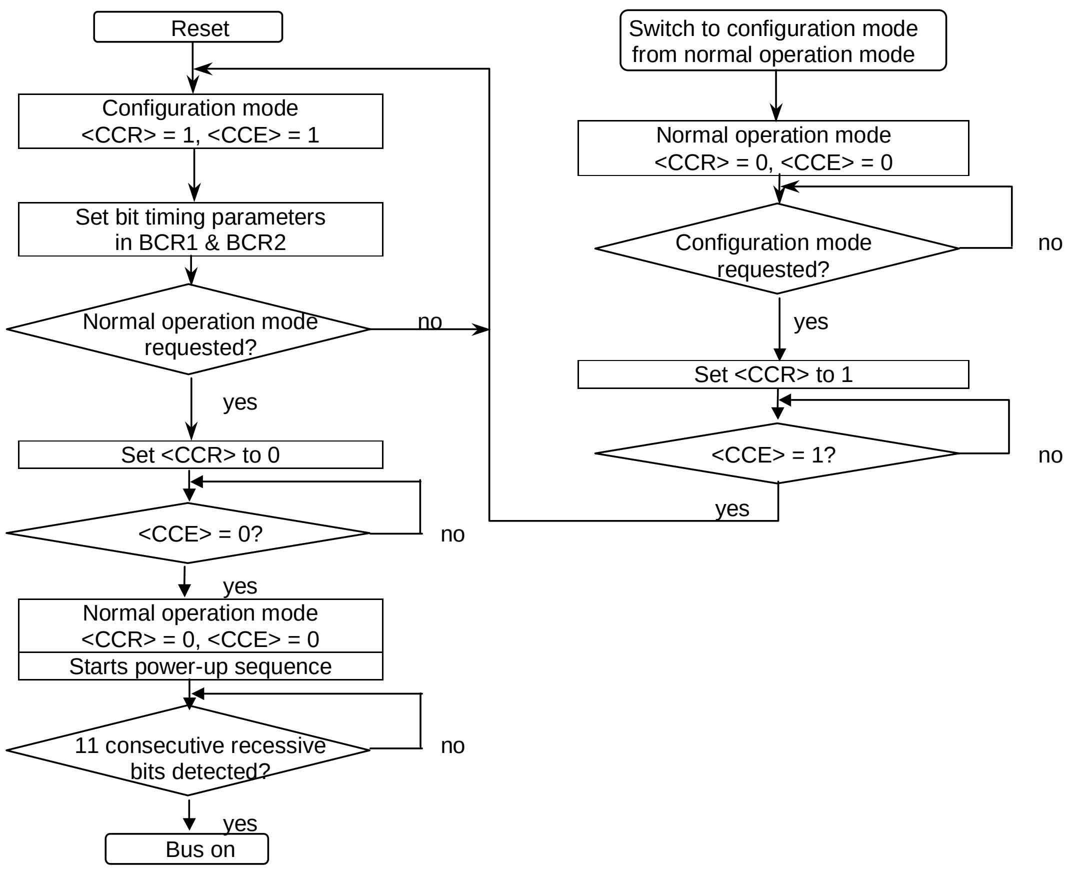


Figure 3.12.5 Flowchart of CAN Initialization

(2) Sleep mode

The sleep mode will be requested by writing 1 to the <SMR> bit of the MCR register. When the CAN controller enters the sleep mode, the status bit <SMA> of the GSR register will be set to 1.

During the sleep mode the clock of the CAN controller is switched off. Only the wake up logic will be active. The read value of the GSR register will be F040H, this means, there is no message in transmit buffer and the sleep mode is active (<SMA> bit is set to 1). Read accesses to all other registers will deliver the value 0000H. Write accesses to all registers but the MCR register will be denied.

The CAN controller leaves the sleep mode if a write access to the MCR register has been detected or there is any bus activity detected on the CAN bus line (with <WUBA> = 1), the CAN controller begins its power up sequence. The CAN controller waits until detecting 11 consecutive recessive bits on the RX input line and goes to bus active after them. The first message that initiates the bus activity can not be received.

In sleep mode the CAN error counters and all 'transmission request set bits <TRSn>' and 'transmission request reset bits <TRRn> will be cleared to 0. After leaving the sleep mode, <SMR> bit in the MCR register and <SMA> bit in the GSR register will be cleared to 0.

If the CAN controller is transmitting a message when the <SMR> bit is set to 1, the CAN controller will not switch to the sleep mode immediately. It will continue until a successful transmission or after losing the arbitration, until a successful reception or until an error condition occurs on the CAN bus line. By this means the CAN controller will initiates no error condition on the CAN bus line.

(3) Halt mode

The halt mode will be requested by writing 1 to the <HMR> bit of the MCR register. When the CAN controller enters the halt mode, the <HMA> bit of the GSR register will be set. During the halt mode the CAN controller does not send or receive any messages. The CAN controller is still active on the CAN bus line. Error Flags and Acknowledge Flags will be sent. The CAN controller leaves the halt mode if the <HMR> bit is reset to 0.

If the CAN controller is transmitting a message when the <HMR> bit is set, the transmission will be continued until a successful transmission or detect a lost arbitration. So the CAN controller initiates no error condition on the CAN bus line.

(4) Test loop back mode

In this mode, the CAN controller can receive its own transmitted message and will generate its own acknowledge bit. No other CAN controller is necessary for the operation. The only supposition is that the RX and TX lines must be connected to a CAN bus transceiver or directly together.

In the test loop back mode, the CAN controller can transmit a message from one mailbox and receive it in another mailbox. The set-up for the mailboxes is the same as in the normal operation mode.

The test loop back mode shall only be enabled or disabled in the configuration mode. Figure 3.12(6) shows the flowchart of the test loop back mode / the test error mode set-up.

(5) Test error mode

The error counters can only be written when the CAN controller is in the test error mode.

When the CAN controller is in the test error mode, both error counters will be written at the same time with the same value (lower 8 bit). The maximum value that can be written into the error counters is 255. Thus, the error counter value of 256 which forces the CAN controller into the bus off mode can not be written into the error counters.

The test error mode shall only be enabled or disabled in the configuration mode. Figure 3.12(6) shows the flowchart of the test loop back mode / the test error mode set-up.

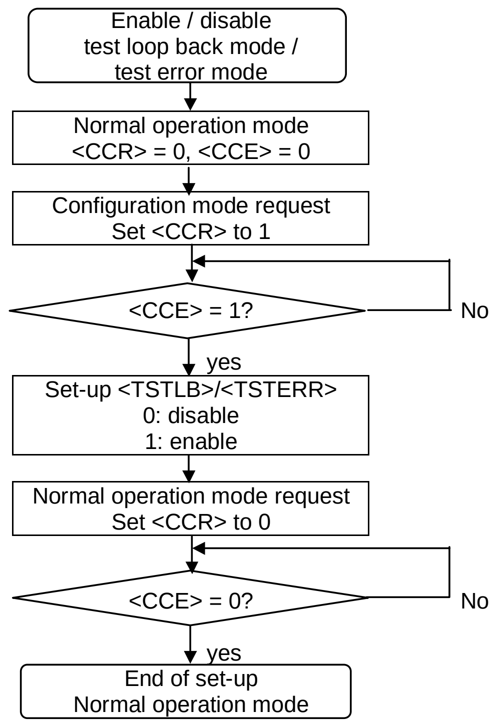


Figure 3.12.6 Flowchart of the test loop back mode / the test error mode set-up

3.12.5 Functional description

(1) Transmit mode

Figure 3.12.7 shows one example of the flowchart of message transmit by using the transmit interrupt INTCT.

It is also possible to use polling instead of interrupt. In this case, “Transmit interrupt generated?” is replaced by “<TAn> = 1?”. “Set <MBIMn> to 1” and “Clear <MBTIFn>” must be removed from the flow.

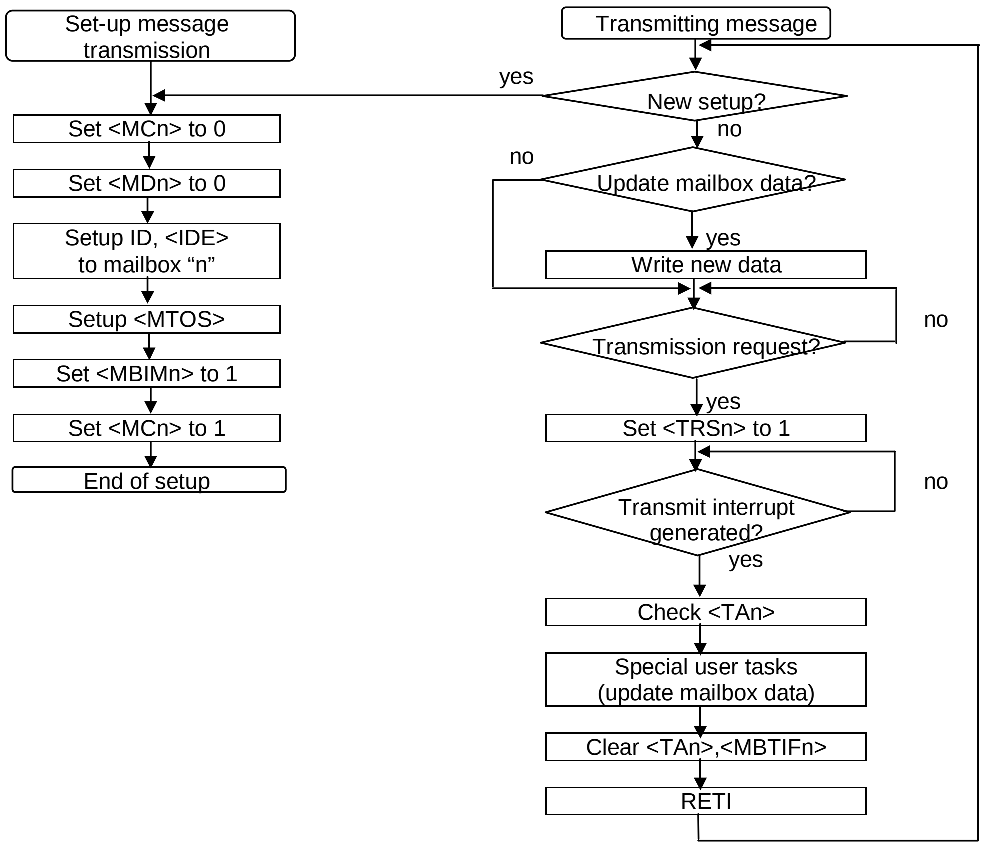


Figure 3.12.7 Flowchart of message transmission (Example)

(2) Receive mode

If the CAN controller has received a message from the CAN bus line, this message will be located in the receive buffer. The message stored in the received buffer will be compared to the identifier of mailbox. If <GAME>/<LAME> bit is set, the global/local acceptance mask register GAM/LAM will be used. If there is one of the following conditions found, no further compare will be done.

- Data fame and a matching identifier in a mailbox configured as receive
- Remote frame and a matching identifier in a mailbox configured as receive
- Remote frame and a matching identifier in a mailbox configured as transmit and <RFH> bit is set

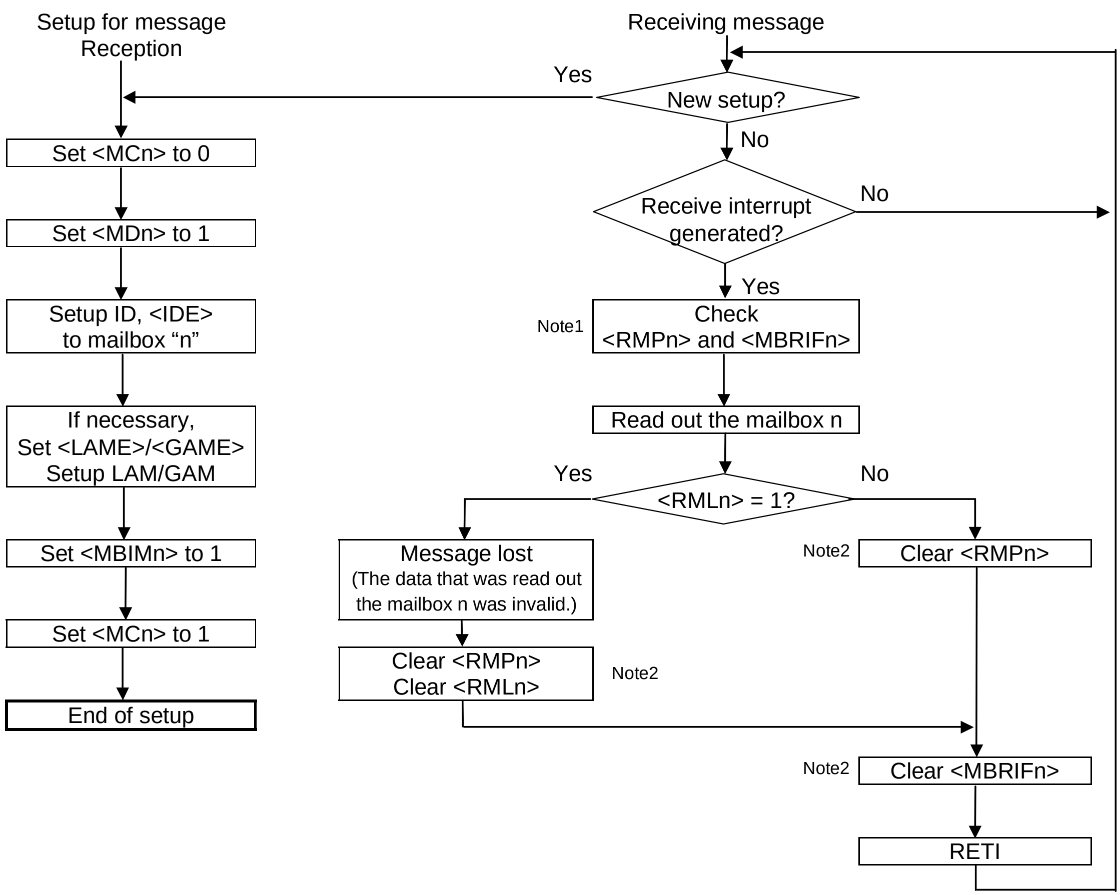
The minimum time to save a next received message after the <RMP> bit set depends on the configured bit timing. In case of the data length code = 0, the minimal time is as follows.

- Standard format: 47 bit times – 16 f_{IO}
- Extended format: 67 bit times – 16 f_{IO}

① Data frames

Figure 3.12.8 shows one example of the flowchart of message reception by using the reception interrupt INTCR.

It is also possible to use polling instead of the interrupt. In this case, “Receive interrupt generated?” is replaced by “<RMPn> = 1?”. “Set <MBIMn> to 1” and “Clear <MBRIFn>” must be removed from the flow.



Note1: Be sure to check <RMPn> and <MBRIFn>.

Note2: If “Clear <RMPn>” is executed and then the mailbox n receives the message before “Clear <MBRIFn>”, <RMPn> will be set 1 (<MBRIFn>=0) depending on the situation.

Figure 3.12.8 Flowchart of message reception (Example)

② Remote frame

Figure 3.12.9 shows one example of the flowchart the handling of remote frame by using the automatic reply feature. This feature is available when the <RFH> bit of a mailbox, which is configured for transmission is set. To avoid data inconsistency problems when updating the mailbox data the CDR register is used.

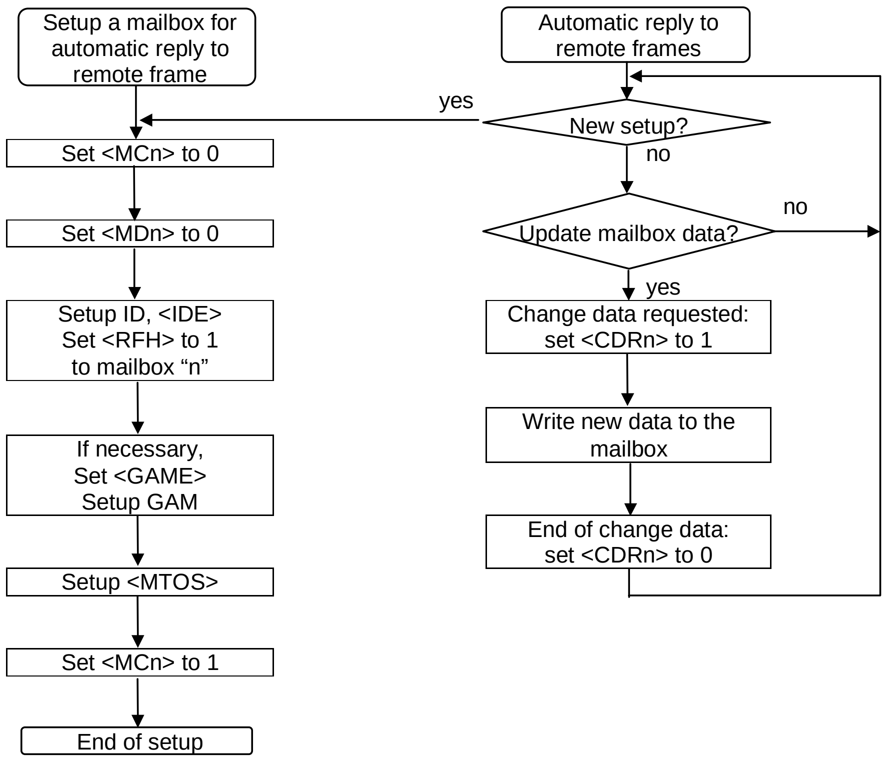


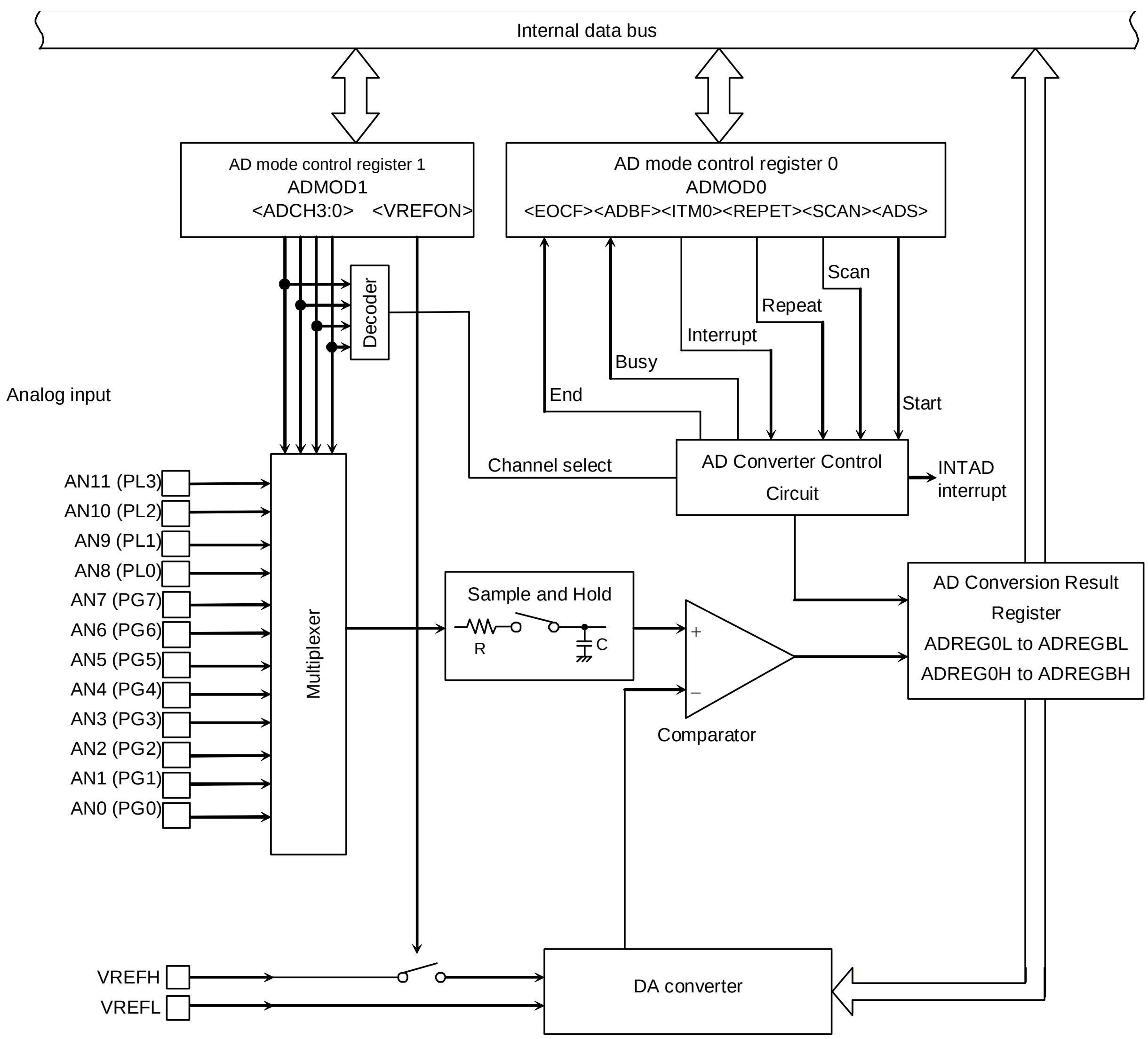
Figure 3.12.9 Flowchart of remote frame handling with the automatic reply feature (Example)

3.13 Analog/Digital Converter

TMP92CD54I incorporates a 10-bit successive approximation-type analog/digital converter (AD converter) with 12-channel analog input.

Figure 3.13.1 is a block diagram of the AD converter. The 12-channel analog input pins (AN0 to AN11) are shared with the input-only port Port G and Port L, so they can be used as an input port.

Note: When IDLE1, IDLE2, IDLE3 or STOP Mode is selected, as to reduce the power, with some timings the system may enter a standby mode even though the internal comparator is still enabled. Therefore be sure to check that AD converter operations are halted before a HALT instruction is executed.



Note: R: internal resistance = 7k ohm (reference data)

C: internal capacitance = 10pF+4pF (reference data)

Figure 3.13.1 Block diagram of AD converter

3.13.1 Analog/Digital converter registers

The AD converter is controlled by the two AD Mode Control Registers: ADMOD0 and ADMOD1. The twelve AD Conversion Data Result Registers (ADREG0H/L, ADREGBH/L) store the results of AD conversion.

Figure 3.13.2 shows the registers related to the AD converter.

AD Mode Control Register 0

ADMOD0 (0138H)

	7	6	5	4	3	2	1	0
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	A D Conversion Busy flag 0: conversion stopped 1: Conversion in progress	Note: Always fixed to 0	Note: Always fixed to 0	Interrupt specification in conversion channel fixed repeat mode 0: every conversion 1: every fourth conversion	Repeat mode specification 0: Single Conversion 1: Repeat Conversion Mode	Scan mode specification 0: Conversion Channel Fixed Mode 1: Conversion Channel Scan Mode	AD conversion start 0: Don't care 1: start conversion Always 0 when read

AD conversion start

0	Don't care
1	Start AD conversion

Note: Always read as 0.

AD scan mode setting

0	AD Conversion Channel Fixed 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

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

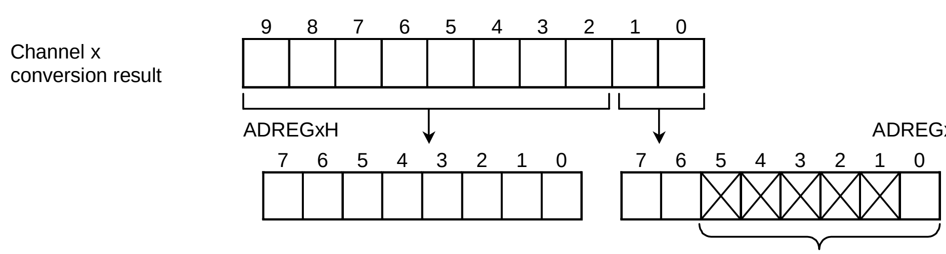
ADREG0H (0121H)		7	6	5	4	3	2	1	0
	Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
	Read/Write	R							
	After Reset	Undefined							
	Function	Stores upper eight bits AD conversion result.							

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 data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.4 AD Converter Related Registers

AD Conversion Result Register 2 Low

ADREG2L (0124H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR21	ADR20						ADR2RF
Read/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

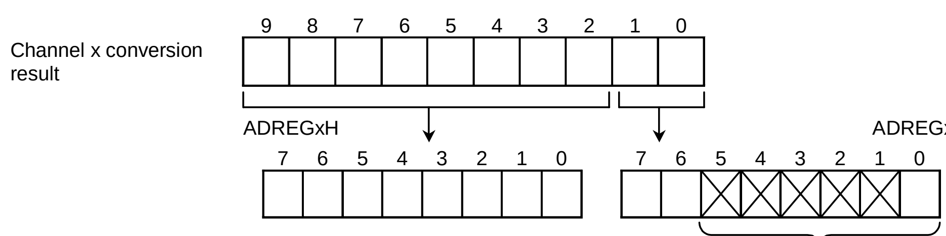
ADREG3L (0126H)

	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.							



- Bits 5 to 1 are always read as 1.
- Bit 0 is the AD conversion data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.5 AD Converter Related Registers

AD Conversion Result Register 4 Low

ADREG4L (0128H)		7	6	5	4	3	2	1	0
	Bit symbol	ADR41	ADR40						ADR4RF
	Read/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

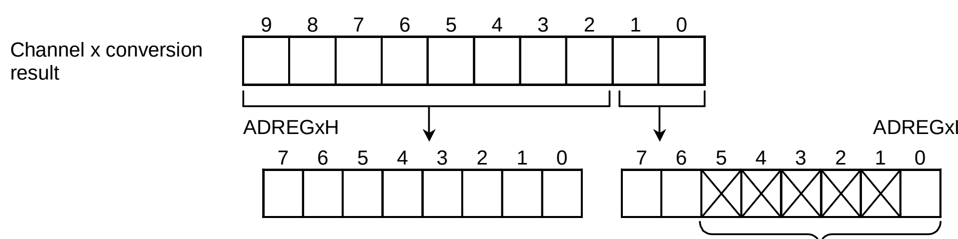
ADREG4H (0129H)		7	6	5	4	3	2	1	0
	Bit symbol	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
	Read/Write	R							
	After Reset	Undefined							
	Function	Stores upper eight bits of AD conversion result.							

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 data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.6 AD Converter Related Registers

AD Conversion Result Register 6 Low

ADREG6L (012CH)		7	6	5	4	3	2	1	0
	Bit symbol	ADR61	ADR60						ADR6RF
	Read/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

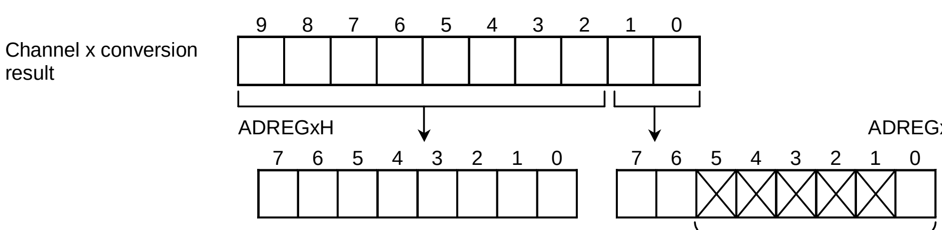
ADREG6H (012DH)		7	6	5	4	3	2	1	0
	Bit symbol	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
	Read/Write	R							
	After Reset	Undefined							
	Function	Stores upper eight bits of AD conversion result.							

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 data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.7 AD Converter Related Registers

AD Conversion Result Register 8 Low

ADREG8L (0130H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR81	ADR80						ADR8RF
Read/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

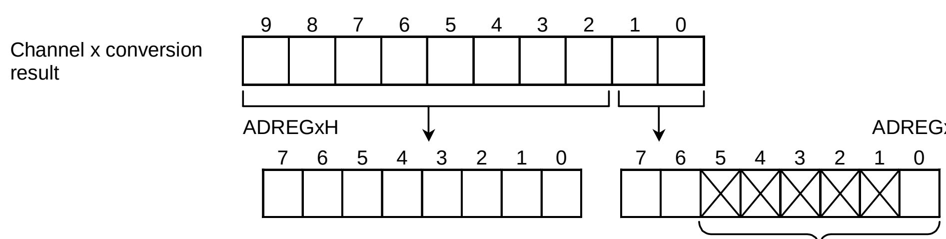
ADREG9L (0132H)

	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.							



- Bits 5 to 1 are always read as 1.
- Bit 0 is the AD conversion data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.8 AD Converter Related Registers

AD Conversion Result 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

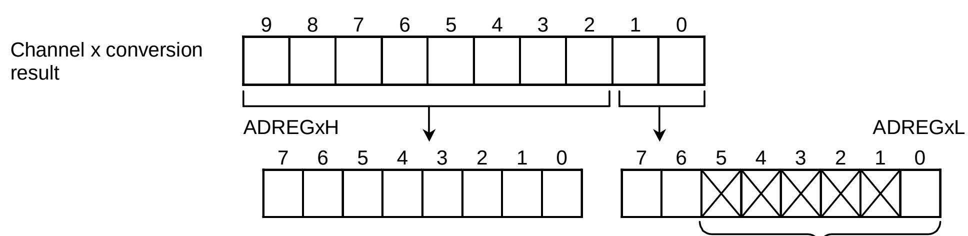
ADREGBL (0136H)

	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 data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.13.9 AD Converter Related Registers

3.13.2 Description of operation

(1) Analog reference voltage

A high-level analog reference voltage is applied to the VREFH pin; a low-level analog reference voltage is applied to the VREFL pin. To perform AD conversion, the reference voltage, the difference between VREFH and VREFL, is divided by 1024 using string resistance. The result of the division is then compared with the analog input voltage.

To turn off the switch between VREFH and VREFL, write a 0 to ADMOD1<VREFON> in AD Mode Control Register 1. To start AD conversion in the OFF state, first write a 1 to ADMOD1<VREFON>, wait 3 μs until the internal reference voltage stabilizes (this is not related to fc), then set ADMOD0<ADS> to 1.

(2) Analog input channel selection

The analog input channel selection varies depends on the operation mode of the AD converter.

- In Analog Input Channel Fixed Mode (ADMOD0<SCAN> = 0)
  Setting ADMOD1<ADCH3:0> selects one of the input pins AN0~AN11 as the input channel.
- In Analog Input Channel Scan Mode (ADMOD0<SCAN> = 1)
  Setting ADMOD1<ADCH3:0> selects one of the twelve scan modes.

Table 3.13.1 illustrates analog input channel selection in each operation mode.

On a Reset, ADMOD0<SCAN> is set to 0 and ADMOD1<ADCH3~ADCH0> is initialized to 0000. Thus pin AN0 is selected as the fixed input channel. Pins not used as analog input channels can be used as standard input port pins.

Table 3.13.1 Analog input channel selection

<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

(3) Starting AD Conversion

To start AD conversion, write a 1 to ADMOD0<ADS> in AD Mode Control Register 0 .When AD conversion starts, the AD Conversion Busy flag ADMOD0<ADBF> will be set to 1, indicating that AD conversion is in progress.

(4) AD conversion modes and the AD Conversion End interrupt

The four AD conversion modes are:

- Channel Fixed Single Conversion Mode
- Channel Scan Single Conversion Mode
- Channel Fixed Repeat Conversion Mode
- Channel Scan Repeat Conversion Mode

The ADMOD0<REPET> and ADMOD0<SCAN> settings in AD Mode Control Register 0 determine the AD mode setting.

Completion of AD conversion triggers an INTAD AD Conversion End interrupt request. Also, ADMOD0<EOCF> will be set to 1 to indicate that AD conversion has been completed.

① Channel Fixed Single Conversion Mode

Setting ADMOD0<REPET> and ADMOD0<SCAN> to 00 selects Conversion Channel Fixed Single Conversion Mode.
In this mode data on one specified channel is converted once only. When the conversion has been completed, the ADMOD0<EOCF> flag is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

② Channel Scan Single Conversion Mode

Setting ADMOD0<REPET> and ADMOD0<SCAN> to 01 selects Conversion Channel Scan Single Conversion Mode.
In this mode data on the specified scan channels is converted once only. When scan conversion has been completed, ADMOD0<EOCF> is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

③ Channel Fixed Repeat Conversion Mode

Setting ADMOD0<REPET> and ADMOD0<SCAN> to 10 selects Conversion Channel Fixed Repeat Conversion Mode.
In this mode data on one specified channel is converted repeatedly. When conversion has been completed, ADMOD0<EOCF> is set to 1 and ADMOD0<ADBF> is not cleared to 0 but held at 1. INTAD interrupt request generation timing is determined by the setting of ADMOD0<ITM0>.
Setting <ITM0> to 0 generates an interrupt request every time an AD conversion is completed.
Setting <ITM0> to 1 generates an interrupt request on completion of every fourth conversion.

④ Channel Scan Repeat Conversion Mode

Setting ADMOD0<REPET> and ADMOD0<SCAN> to 11 selects Conversion Channel Scan Repeat Conversion Mode.
In this mode data on the specified scan channels is converted repeatedly. When each scan conversion has been completed, ADMOD0<EOCF> is set to 1 and an INTAD interrupt request is generated. ADMOD0<ADBF> is not cleared to 0 but held at 1.
To stop conversion in a repeat conversion mode (i.e. in cases ③ and ④), write a 0 to ADMOD0<REPET>. After the current conversion has been completed, the repeat conversion mode terminates and ADMOD0<ADBF> is cleared to 0.
Switching to a halt state (IDLE2 Mode with ADMOD1<I2AD> cleared to 0, IDLE1 Mode or STOP Mode) immediately stops operation of the AD converter even when AD conversion is still in progress. In repeat conversion modes (i.e. in cases ③ and ④), when the halt is released, conversion restarts from the beginning. In single conversion modes (i.e. in cases ① and ②), conversion does not restart when the halt is released (the converter remains stopped).

Table 3.13.2 shows the relationship between the AD conversion modes and interrupt requests.

Table 3.13.2 Relationship Between AD Conversion Modes and Interrupt Requests

Mode	Interrupt Request Generation	ADMOD0		
		<ITM0>	<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

160/fc (8 μs @ f_C = 20 MHz) are required for the AD conversion of one channel.

(6) Storing and reading the results of AD conversion

The AD Conversion Data Upper and Lower Registers (ADREG0H/L to ADREGBH/L) store the results of AD conversion. (ADREG0H/L to ADREGBH/L are read-only registers.)
In Channel Fixed Repeat Conversion Mode, the conversion results are stored successively in registers ADREG0H/L to ADREG3H/L. In other modes the AN0, AN1, AN2, AN3, AN4, AN5, AN6, AN7 conversion results are stored in ADREG0H/L, ADREG1H/L, ADREG2H/L, ADREG3H/L, ADREG4H/L, ADREG5H/L, ADREG6H/L, ADREG7H/L, ADREG8H/L, ADREG9H/L, ADREGAH/L, ADREGBH/L respectively.
Table 3.13.3 shows the correspondence between the analog input channels and the registers which are used to hold the results of AD conversion.

Table 3.13.3 Correspondence Between Analog Input Channels and AD Conversion Result Registers

Analog input channel (Port G/Port L)	AD Conversion Result Register	
	Conversion modes other than at right	Channel fixed repeat conversion mode (every 4 th conversion)
AN0	ADREG0H/L	ADREG0H/L ↓ ADREG1H/L ↓ ADREG2H/L ↓ ADREG3H/L
AN1	ADREG1H/L	
AN2	ADREG2H/L	
AN3	ADREG3H/L	
AN4	ADREG4H/L	
AN5	ADREG5H/L	
AN6	ADREG6H/L	
AN7	ADREG7H/L	
AN8	ADREG8H/L	
AN9	ADREG9H/L	
AN10	ADREGAH/L	
AN11	ADREGBH/L	

<ADRxRF>, bit 0 of the AD conversion data lower register, is used as the AD conversion data storage flag. The storage flag indicates whether the AD conversion result register has been read or not. When a conversion result is stored in the AD conversion result register, the flag is set to 1. When either of the AD conversion result registers (ADREGxH or ADREGxL) is read, the flag is cleared to 0.
Reading the AD conversion result also clears the AD Conversion End flag ADMOD0<EOCF> to 0.

Setting example:

① Convert the analog input voltage on the AN3 pin and write the result, to memory address 0800H using the AD interrupt (INTAD) processing routine.

Main routine:

		7 6 5 4 3 2 1 0	
INTE0AD	←	1 1 0 0 - - - -	Enable INTAD and set it to Interrupt Level 4.
ADMOD1	←	1 1 0 0 0 0 1 1	Set pin AN3 to be the analog input channel.
ADMOD0	←	X X 0 0 0 0 0 1	Start conversion in Channel Fixed Single Conversion Mode.

Interrupt routine processing example:

WA	←	ADREG3	Read value of ADREG3L and ADREG3H into 16-bit general-purpose register WA.
WA	>>6		Shift contents read into WA six times to right and zero-fill upper bits.
(0800H)	←	WA	Write contents of WA to memory address 0800H.

② This example repeatedly converts the analog input voltages on the three pins AN0, AN1 and AN2, using Channel Scan Repeat Conversion Mode.

INTE0AD	←	1 0 0 0 - - - -	Disable INTAD.
ADMOD1	←	1 1 0 0 0 0 1 0	Set pins AN0~AN2 to be the analog input channels.
ADMOD0	←	X X 0 0 0 1 1 1	Start conversion in Channel Scan Repeat Conversion Mode.

Note: X = Don't care; "–" = No change

3.14 Watchdog Timer (Runaway Detection Timer)

TMP92CD54I contains a watchdog timer of runaway detecting.

The watchdog timer (WDT) is used to return the CPU to the normal state when it detects that the CPU has started to malfunction (runaway) due to causes such as noise. When the watchdog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU of the malfunction.

Connecting the watchdog timer output to the reset pin internally forces a reset.

3.14.1 Configuration

Figure 3.14.1 is a block diagram of the watchdog timer (WDT).

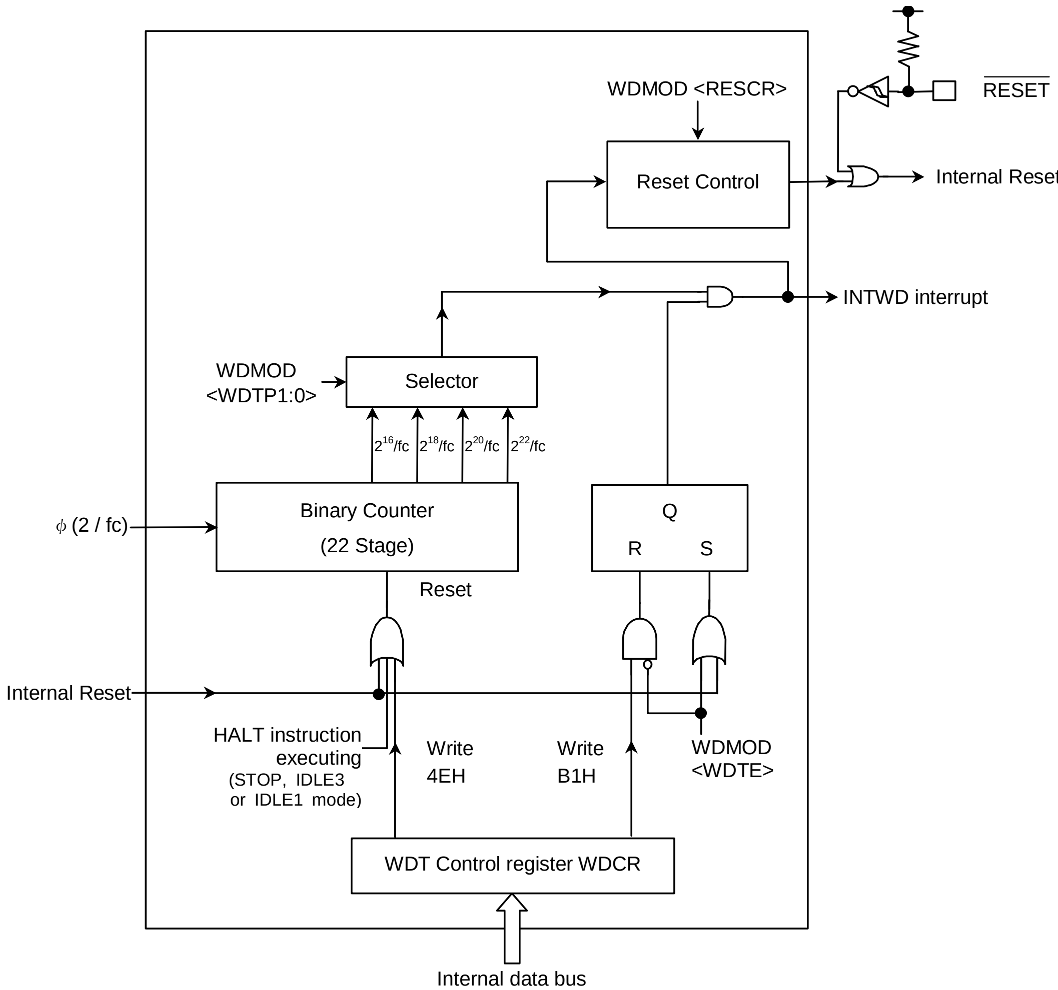


Figure 3.14.1 Block Diagram of Watchdog Timer

The watchdog timer consists of a 22-stage binary counter which uses the clock ϕ (2/fc) as the input clock. The binary counter can output 2^{16}/fc, 2^{18}/fc, 2^{20}/fc and 2^{22}/fc. Selecting one of the outputs using WDMOD<WDTP1,WDTP0> generates a watchdog timer interrupt when an overflow occurs. In the case of using watchdog timer after INTWD request generated, the clear code (4EH) should be written to the WDCR register in other to clear the binary counter.

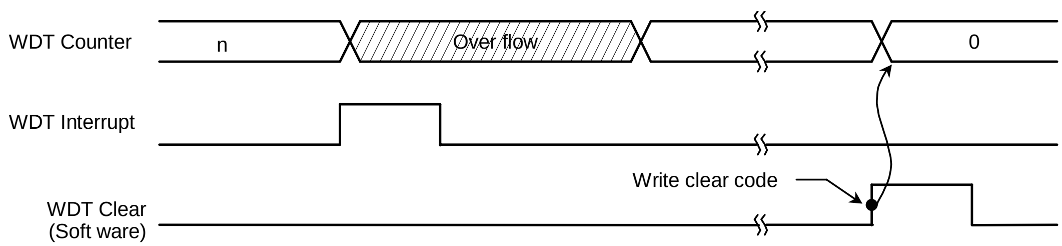


Figure 3.14.2 Normal Mode

The runaway detection result can also be connected to the Reset pin internally.
In this case, the reset time will be between 44×4/fc and 58×4/fc system clocks (8.8~11.6 μs @ fc= 20 MHz) as shown in figure 3.14.3

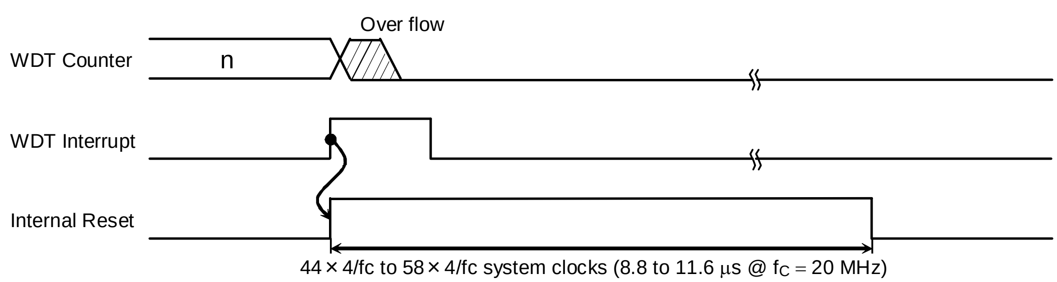


Figure 3.14.3 Reset Mode

3.14.2 Control registers

The watchdog timer WDT is controlled by three control registers WDMOD, WDCR and CLKMOD.

(1) Watchdog Timer Mode Register (WDMOD)

i) Setting the detection time for the watchdog timer in <WDTP1,WDTP0>

This 2-bit register is used for setting the watchdog timer interrupt time used when detecting runaway. On a Reset this register is initialized to WDMOD<WDTP1,WDTP0> = 00.
The detection times for WDT is 2^{16}/fc [s]. (The number of system clocks is approximately 65,536.)

ii) Watchdog timer enable/disable control register <WDTE>

At reset, the WDMOD<WDTE> is initialized to 1, enabling the watchdog timer.
To disable the watchdog timer, it is necessary to set this bit to 0 and to write the disable code (B1H) to the Watchdog Timer Control Register WDCR. This makes it difficult for the watchdog timer to be disabled by runaway.
However, it is possible to return the watchdog timer from the disabled state to the enabled state merely by setting <WDTE> to 1.

iii) Watchdog timer out reset connection <RESCR>

This register is used to connect the output of the watchdog timer with the RESET terminal internally. Since WDMOD<RESCR>is initialized to 0 at reset, a reset by the watchdog timer will not be performed.

(2) Watchdog Timer Control Register (WDCR)

This register is used to disable and clear the binary counter for the watchdog timer.

- Disable control

  The watchdog timer can be disabled by clearing WDMOD<WDTE> to 0 and then writing the disable code (B1H) to the WDCR register.

```
WDMOD   ← 0 - - - - - - -     Clear WDMOD<WDTE> to 0.
WDCR    ← 1 0 1 1 0 0 0 1     Write the disable code (B1H).
```

- Enable control

  Set WDMOD<WDTE>to 1.

- Watchdog timer clear control

  To clear the binary counter and cause counting to resume, write the clear code (4EH) to the WDCR register. In the case of using watchdog timer after INTWD request generated, the clear code (4EH) should be written to the WDCR register in other to clear the binary counter.

```
WDCR    ← 0 1 0 0 1 1 1 0     Write the clear code (4EH).
```

(3) Clock Mode Register (CLKMOD)

This register is used to set the warming up time after the stop mode ends.
The output of CLK pin is chosen from fc and 2/5fc by the setup of CLKMOD <CLKM1,CLKM0>.
Moreover, CLK pin output can be stopped by writing "0" in CLKMOD <CLKOE>.
By the setup of CLKMOD <HALTM1,HALTM0>, it becomes the HALT mode of IDLE1, IDLE2, IDLE3 or STOP.

WDMOD (0110H)

	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 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

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

WDCR (0111H)

	7	6	5	4	3	2	1	0
Bit symbol	–							
Read/Write	W							
After reset	–							
Function	B1H: WDT disable code 4EH: WDT clear code							

WDT disable/clear control

B1H	Disable code
4EH	Clear code
Others	Don't care

Figure 3.14.5 Watchdog Timer Control Register

		7	6	5	4	3	2	1	0
CLKMOD (010AH)	bit Symbol	HALTM1	HALTM0	-	-	-	CLKOE	CLKM1	CLKM0
	Read/Write	R/W			R/W		R/W		
	After reset	1	1	-	0	-	0	0	0
	Function	Standby 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	Not output (Pull 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 Operation

The watchdog timer generates an INTWD interrupt when the detection time set in the WDMOD<WDTP1,WDTP0> has elapsed. The watchdog timer must be zero-cleared in software before an INTWD interrupt will be generated. If the CPU malfunctions (i.e. if runaway occurs) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter will overflow and an INTWD interrupt will be generated. The CPU will detect malfunction (runaway) due to the INTWD interrupt and in this case it is possible to return to the CPU to normal operation by means of an anti-mulfunction program.

The watchdog timer begins operating immediately on release of the watchdog timer reset.

The watchdog timer is reset and halted in IDLE1, IDLE3 or STOP Modes.

When the device is in IDLE2 mode, the operation of WDT depends on the WDMOD<I2WDT> setting. Ensure that WDMOD<I2WDT> is set before the device enters IDLE2 Mode.

Example:

i) Clear the binary counter.

```
WDCR    ← 0 1 0 0 1 1 1 0     Write the clear code (4EH).
```

ii) Set the watchdog timer detection time to $2^{18}/f_C$.

```
WDMOD   ← 1 0 1 - - - - -
```

iii) Disable the watchdog timer.

```
WDMOD   ← 0 - - X - - - -     Clear <WDTE> bit to 0.
WDCR    ← 1 0 1 1 0 0 0 1     Write the disable code (B1H).
```

3.15 RAM control

RAM control register (RAMCR) has <RAMWI>bit for inhibition to write data to internal RAM and <RAMSTB> bit to detect lower voltage under VSTB level. VSTB level is the voltage level impossible to keep the data of Internal RAM.

Only data “1” can be written to RAMCR<RAMSTB>, and data “0” can’t be written.

When RAMCR<RAMSTB> is set to “1” by software, in the case of voltage drop under VSTB level RAMCR<RAMSTB> is reset to “0”. After power on RAMCR<RAMSTB> is reset to “0”.

RAMCR<RAMSTB> is not changed by standby operation and reset operation. The detection of reset operation (Warm reset / Power-on reset) and the condition of RAM data (kept / lost) is enable, to read RAMCR<RAMSTB>.

RAM Write Inhibit<RAMWI> bit is used for inhibition to write data to internal RAM. After reset RAMCR<RAMWI> is set to “1”, writing data to internal RAM is accepted. When RAMCR<RAMWI> is set to “0”, writing data to internal RAM is inhibited.

RAMCR (016DH)

	7	6	5	4	3	2	1	0
Bit 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 VCC3 $\leqq$VSTB. After power on reset.
1	After “1” is set by software, this data isn’t changed at VCC3$>$VSTB.

Note1: After power-on reset, initialized to 0. No change by warm reset. Use after setting to 1 by software. 0 cannot be written by software.

Note2: When changed to STOP or Idle3 in HALT mode with RAMCR<RAMSTB> set to 1, current flows.

Note3: Emulator doesn’t support RAM control functions.

Note4: To set to RAMCR<RAMSTB> bit to “1”, need 8 state (@ fc = 20MHz; For that time, do not set Idle2, 3 or STOP mode.). After that, the power-supply detection circuit runs.

Note5: VCC3 means internal voltage.

(Note) There are restrictions at un-use of Voltage regulator (see section “4.2 DC Electrical Characteristics”).

3.16 Timer for Real Time Clock (RTC)

TMP92CD54I features a timer which is used for real time count. An interrupt (INTRTC) can be generated every 0.0625 s, 0.125 s, 0.25 s, 0.50 s, 1 s or 2 s by using the low-frequency 32.768kHz clock. A clock function can be easily used.

Timer for Real Time Clock can operate in all mode in which a low frequency oscillation is operated. (except STOP mode)

In addition, INTRTC can return the device from every standby mode except STOP mode to the NORMAL mode.

3.16.1 Block diagram

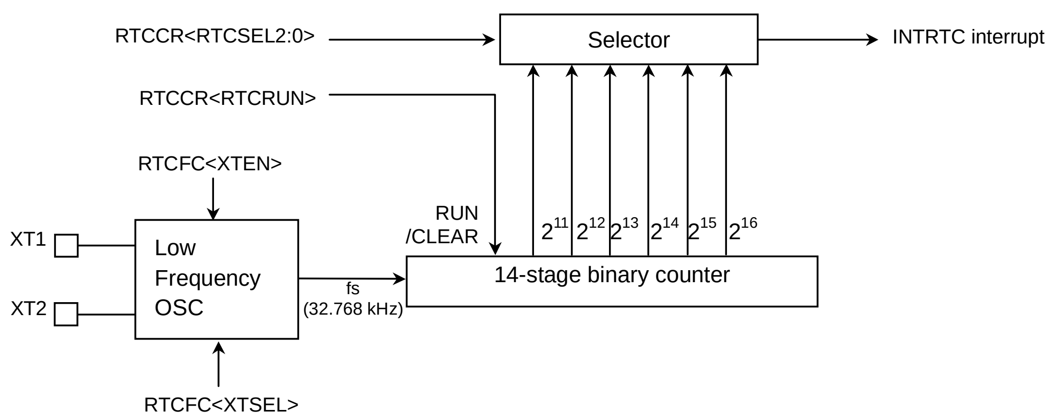


Figure 3.16.1 Block diagram for timer for real-time clock

3.16.2 SFRs

RTC has 2 registers.

Timer for Real Time Clock is controlled by Timer for Real-Time Clock Control Register (RTCCR). The period of interrupt request INTRTC is selected from 6 types by setting <RTCSEL2 to 0>. To start/stop the counter is controlled by <RTCRUN>.

The low frequency oscillator is controlled by Timer for Real Time Function Register(RTCFC). If it is released from STOP mode, without RESET input, RTCFC will be initialized.

Figure 3.16.2 and 3.16.3 show these registers.

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 3.16.1 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

Note1: Please consider the stability-time for the oscillator.
Note2: If it released from STOP mode, RTCFC register will be initiallized without a RESET input. Therefore, it is necessary to set up RTCFC register again after releaseing from STOP mode.

Figure 3.16.2 Timer for Real Time Clock Function Register

Example of register setting:

```
LD    (RTCFC),01h    ; L-OSC start
      :              ; (Wait for the stability-time)
LD    (RTCCR),03h    ; Run at 2^13/fs
```

3.16.3 CR oscillation

RTC can also work by using CR oscillator within. And oscillation type is controlled by the RTCFC. If XTSEL bit is set, CR oscillation is available. And when CR oscillation is used in the application, it is necessary to supplement external resistor and capacitor to XT1, XT2 pins. A recommended external circuit is shown the following figure “Figure 3.16.3”. And “Figure3.16.4” shows CR oscillation frequency related to the combination of resistor and capacitor, provided that measurement environment is the typical condition described below.

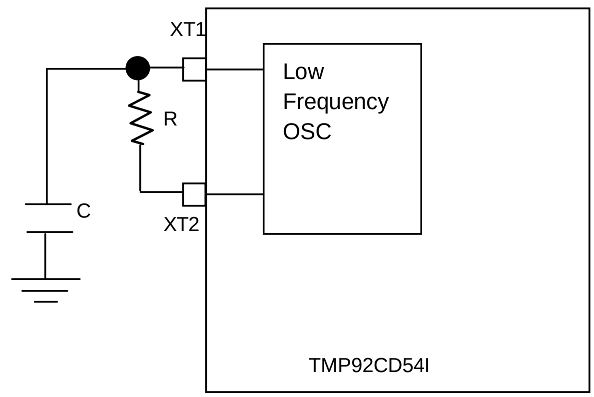


Figure 3.16.3 A external circuit for CR oscillation

(Note) Please adjust the value R and C for the application set. For example, we confirmed as follows, 1) R = 40kOhm and C = 470pF 2) R = 82kOhm and C = 220pF at condition of 32.768kHz and room temperature.

3.17 Voltage Regulator

3V output regulator for the internal logic power supply is installed in TMP92CD54I. The power supply is supplied to internal logic by connecting each DVCC3 terminal with regulator output terminal REGOUT.

This regulator can control use/nonuse with the terminal REGEN.

Table3.17.1 REGOUT output by REGEN setting

REGEN input	REGOUT output
”H”	3V output for internal logic
”OPEN” Note)	3V output for internal logic
”L”	0V output (Do not connect GND)

Note) As for REGEN, use with OPEN is also possible because of an internal pull-up.

When the regulator is not used, it is necessary to supply the power supply to internal logic separately.

3.17.1 Block diagram

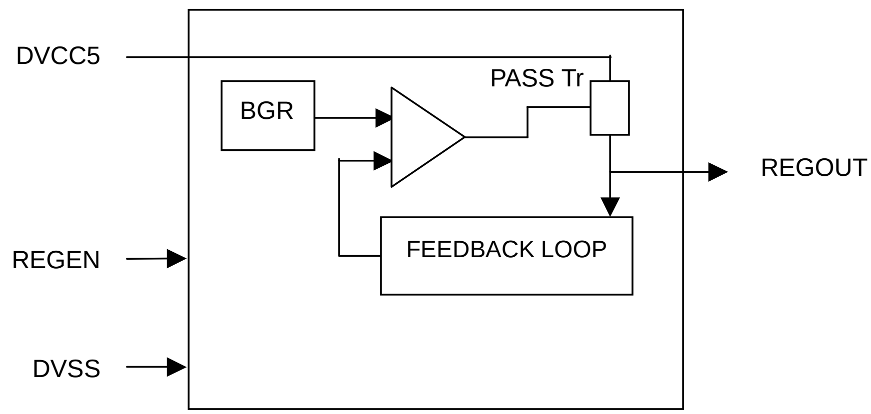


Diagram 3.17.1 Regulator block

3.17.2 External connecting

For the oscillation prevention of the output voltage, connect stabilization capacitor (Cs) with the place between REGOUT and DVSS as near the terminal as possible. It is necessary to add resistance (ESR) to Cs serially according to the substrate capacity as shown in Figure 3.17.2.

Because the change in internal resistance by the temperature might become the destabilizing factor of the regulator output about the selection of the capacitor, we will recommend the use of the capacitor with a good temperature characteristic.

Moreover, recommend bypass capacitor (Cb) to be connected as a noise tolerance improvement of the REGOUT output between DVCC3-DVSS.

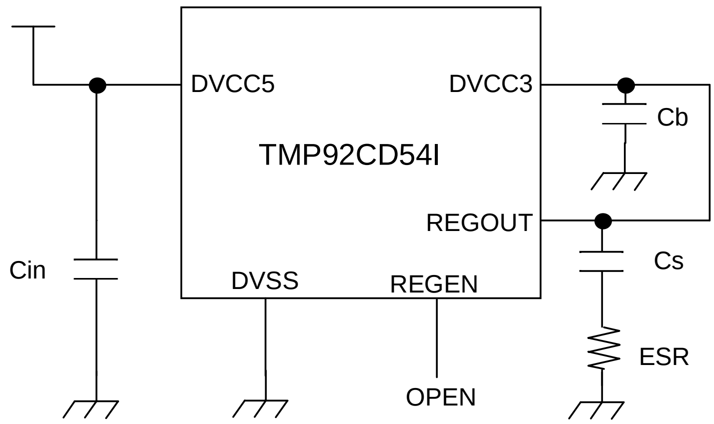


diagram 3.17.2 Regulator connection

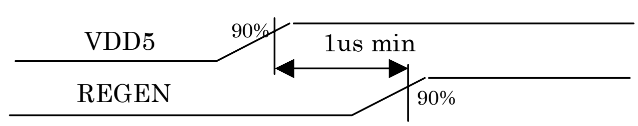
3.17.3 Directions

- Application

This regulator is designed for TMP92CD54I. Do not connect the output from REGOUT except the terminal DVCC3 of TMP92CD54I.

- Timing of when power supply is turned on and REGEN input signal

When the power supply is turning on, keep the REGEN terminal OPEN or input the enable signal (H level) to the terminal REGEN after at least 1us passes from the power supply turning on.



- The number of wires of Cs, Cb and ESR

Depending on modular composition, its stray capacitance and parasitic capacitance might influence the regulator characteristic.

Investigating the characteristic about the static characteristic and the transient characteristic along actual use conditions, the number of wires should be decided according to the margin of Cin, Cs, Cb and ESR.

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

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)	Σ_{IOL}	100	mA
Output Current(total)	Σ_{IOH}	-100	mA
Power Dissipation(Ta=85degree C)	P_D	600	mW
Soldering Temperature(10s)	T_{SOLDER}	260	degree C
Storage Temperature	T_{STG}	-65 to 150	degree C
Operation Temperature	T_{OPR}	-40 to 85	degree C

Note: The absolute maximum ratings are rated values that must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products that include this device, ensure that no absolute maximum rating value will ever be exceeded.

4.2 DC Electrical Characteristics

Vcc5 =4.5V to 5.25V / fc = 16 to 20MHz / Ta = -40 to 85 degree 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*VCC5	V
Input Low Voltage $\overline{INT0}$ $\overline{NMI}$ $\overline{RESET}$ P70, P71, P73 to P75 PC0 to PC5 PD0 to PD7 PF0 to PF7 PM0 to PM4	V_{IL2}		-0.3	0.25*VCC5	V
P72, PN0 to PN6	V_{IL6}		-0.3	0.3*VCC5	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*VCC3	V
Input Low Voltage XT1 (CR)	V_{IL5}	* Vcc3 = 3.3V	-0.3	0.2*VCC3	V
Input High Voltage P00 to P07(D0 to 7) PG0 to PG7 PL0 to PL3	V_{IH0}		2.2	VCC5+0.3	V
Input High Voltage P00 to P07 P40 to P47	V_{IH1}		0.7*VCC5	VCC5+0.3	V
Input High Voltage $\overline{INT0}$ $\overline{NMI}$ $\overline{RESET}$ P70, P71, P73 to P75 PC0 to PC5 PD0 to PD7 PF0 to PF7 PM0 to PM4	V_{IH2}		0.75*VCC5	VCC5+0.3	V
P72, PN0 to PN6	V_{IH6}		0.7*VCC5	VCC5+0.3	V
Input High Voltage AM0 to AM1 TEST0 to TEST1	V_{IH3}		VCC5-0.3	VCC5+0.3	V
Input High Voltage X1, XT1 (Crystal)	V_{IH4}	* Vcc3 = 3.3V	0.8*VCC3	VCC3+0.3	V
Input High Voltage XT1 (CR)	V_{IH5}	* Vcc3 = 3.3V	0.7*VCC3	VCC3+0.3	V

Parameter	Symbol	Condition		Min	Max	Unit
Output Low Voltage	V_{OL}	I_{OL} = 3.0mA			0.4	V
Output High Voltage	V_{OH0}	I_{OH} = -400uA		2.4		V
	V_{OH1}	I_{OH} = -100uA		0.75*VCC5		
	V_{OH2}	I_{OH} = -20uA		0.9*VCC5		
	V_{OHn}	I_{OH} = -200uA, PF6(TX) pin		0.82*VCC5		
Input Leakage Current	I_{LI}	0.0 ≦ Vin ≦ VCC5		0.02(typ.)	+/- 5	uA
Output Leakage Current	I_{LO}	0.2 ≦ Vin ≦ VCC5-0.2		0.05(typ.)	+/- 10	uA
Operating Current (Single Chip)*	I_{CC5}	V_{CC5}=5.25V , X1=10MHz(Internal 20MHz)		70(typ)	100	mA
Operating Current (Stand-by)	$I_{CC5IDLE2}$	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 degree C V_{CC5}=5.25V, Ta = -10 to 55 degree C		220 140	uA
	$I_{CC5STOP}$	STOP Mode	V_{CC5}=5.25V, Ta = -40 to 85 degree C V_{CC5}=5.25V, Ta = -10 to 55 degree C		200 120	uA
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}	$\overline{\text{RESET}}$		60	220	K ohm
	R_{CLK}	CLK				
	R_{REGEN}	REGEN				
Schmitt Width	V_{TH}	INT0, $\overline{\text{NMI}}$, 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

*: On condition that external bus don't operate

4.3 AC Characteristics

Read cycle

VCC5=4.5 to 5.25V±5%, TA=-40 to 85 degree C

No.	Parameter	Symbol	Min	Max	@20MHz	@16MHz	Unit
1	OSC period (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System Clock period (=T)	t_{CYC}	50	62.5	50	62.5	ns
3	CLK Low Width	t_{CL}	0.5×T-15		10	16	ns
4	CLK High Width	t_{CH}	0.5×T-15		10	16	ns
5-1	A0 to A23 Valid → D0 to D7 Input @0WAIT	t_{AD}		2.0×T-50	50	75	ns
5-2	A0 to A23 Valid → D0 to D7 Input @1WAIT	t_{AD3}		3.0×T-50	100	138	ns
6-1	$\overline{RD}$ Fall → D0 to D7 Input @0WAIT	t_{RD}		1.5×T-45	30	49	ns
6-2	$\overline{RD}$ Fall → D0 to D7 Input @1WAIT	t_{RD3}		2.5×T-45	80	111	ns
7-1	$\overline{RD}$ Low Width @0WAIT	t_{RR}	1.5×T-20		55	74	ns
7-2	$\overline{RD}$ Low Width @1WAIT	t_{RR3}	2.5×T-20		105	136	ns
8	A0 to A23 Valid → $\overline{RD}$ Fall	t_{AR}	0.5×T-20		5	11	ns
9	$\overline{RD}$ Fall → CLK Fall	t_{RK}	0.5×T-20		5	11	ns
10	A0 to A23 Valid → D0 to D7 Hold	t_{HA}	0		0	0	ns
11	$\overline{RD}$ Rise → D0 to D7 Hold	t_{HR}	0		0	0	ns
12	A0 to A23 Valid → PORT Input	t_{APR}		2.0×T-120	-20	5	ns
13	A0 to A23 Valid → PORT Hold	t_{APH}	2.0×T		100	125	ns
14	$\overline{WAIT}$ Set-up Time	t_{TK}	15		15	15	ns
15	$\overline{WAIT}$ Hold Time	t_{KT}	5		5	5	ns

Write cycle

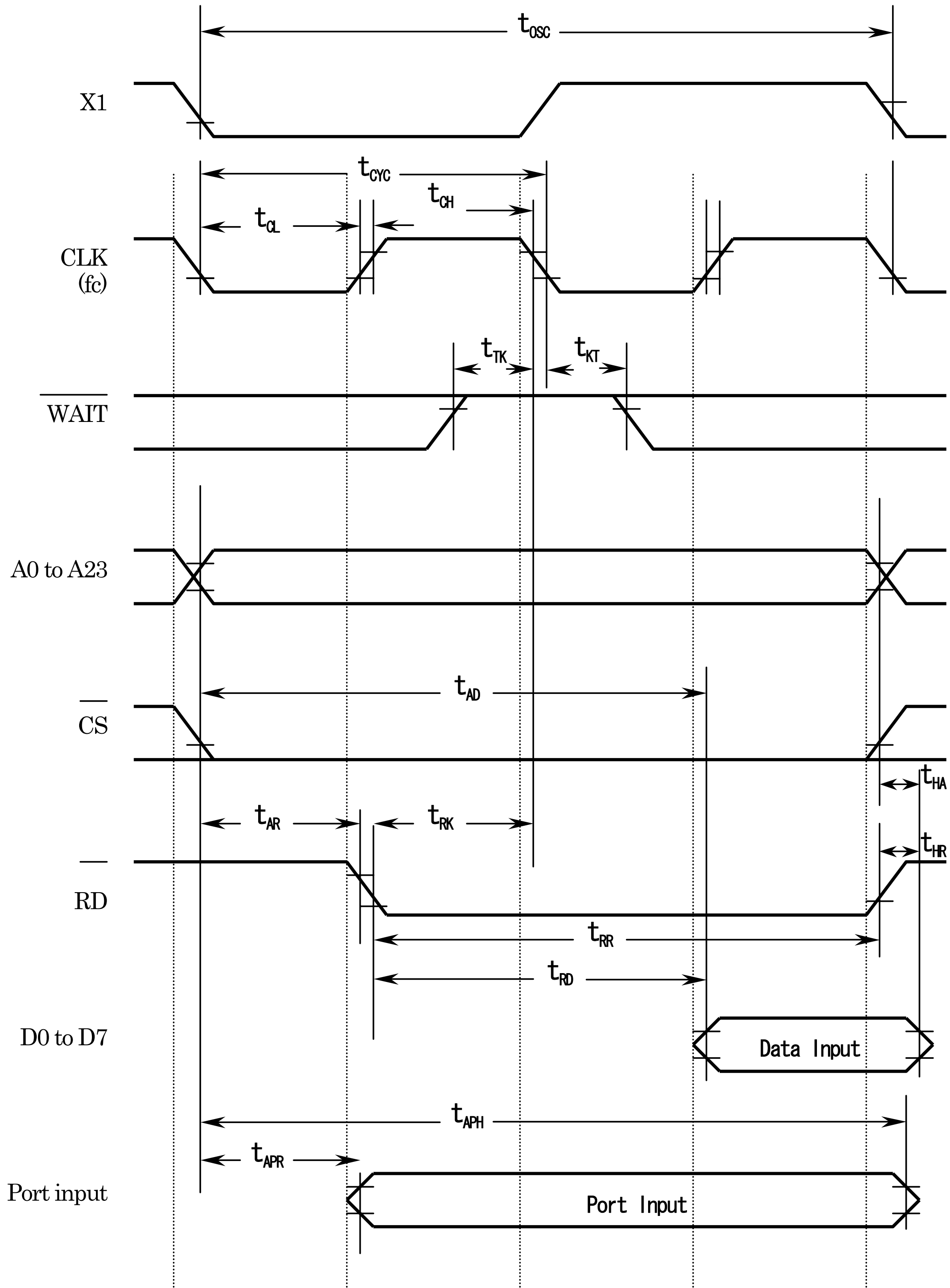
VCC5=5.0V±5%, TA=-40 to 85 degree C

No.	Parameter	Symbol	Min	Max	@20MHz	@16MHz	Unit
1	OSC period (X1/X2)	t_{OSC}	100	125	100	125	ns
2	System Clock period (=T)	t_{CYC}	50	62.5	50	62.5	ns
3	CLK Low Width	t_{CL}	0.5×T-15		10	16	ns
4	CLK High Width	t_{CH}	0.5×T-15		10	16	ns
5-1	D0 to D7 Valid → $\overline{WR}$ Rise @0WAIT	t_{DW}	1.25×T-35		28	43	ns
5-2	D0 to D7 Valid → $\overline{WR}$ Rise @1WAIT	t_{DW3}	2.25×T-35		78	106	ns
6-1	$\overline{WR}$ Low Width @0WAIT	t_{WW}	1.25×T-30		33	48	ns
6-2	$\overline{WR}$ Low Width @1WAIT	t_{WW3}	2.25×T-30		83	111	ns
7	A0 to A23 Valid → $\overline{WR}$ Fall	t_{AW}	0.5×T-20		5	11	ns
8	$\overline{WR}$ Fall → CLK Fall	t_{WK}	0.5×T-20		5	11	ns
9	$\overline{WR}$ Fall → A0 to A23 Hold	t_{WA}	0.25×T-5		8	11	ns
10	$\overline{WR}$ Fall → D0 to D7 Hold	t_{WD}	0.25×T-5		8	11	ns
11	A0 to A23 Valid → PORT Output	t_{APW}		2.0×T+70	170	195	ns
12	$\overline{WAIT}$ Set-up Time	t_{TK}	15		15	15	ns
13	$\overline{WAIT}$ Hold Time	t_{KT}	5		5	5	ns
14	$\overline{RD}$ Rise → D0 to D7 Output	t_{RDO}	1.25×T-35		20	26	ns

AC Condition

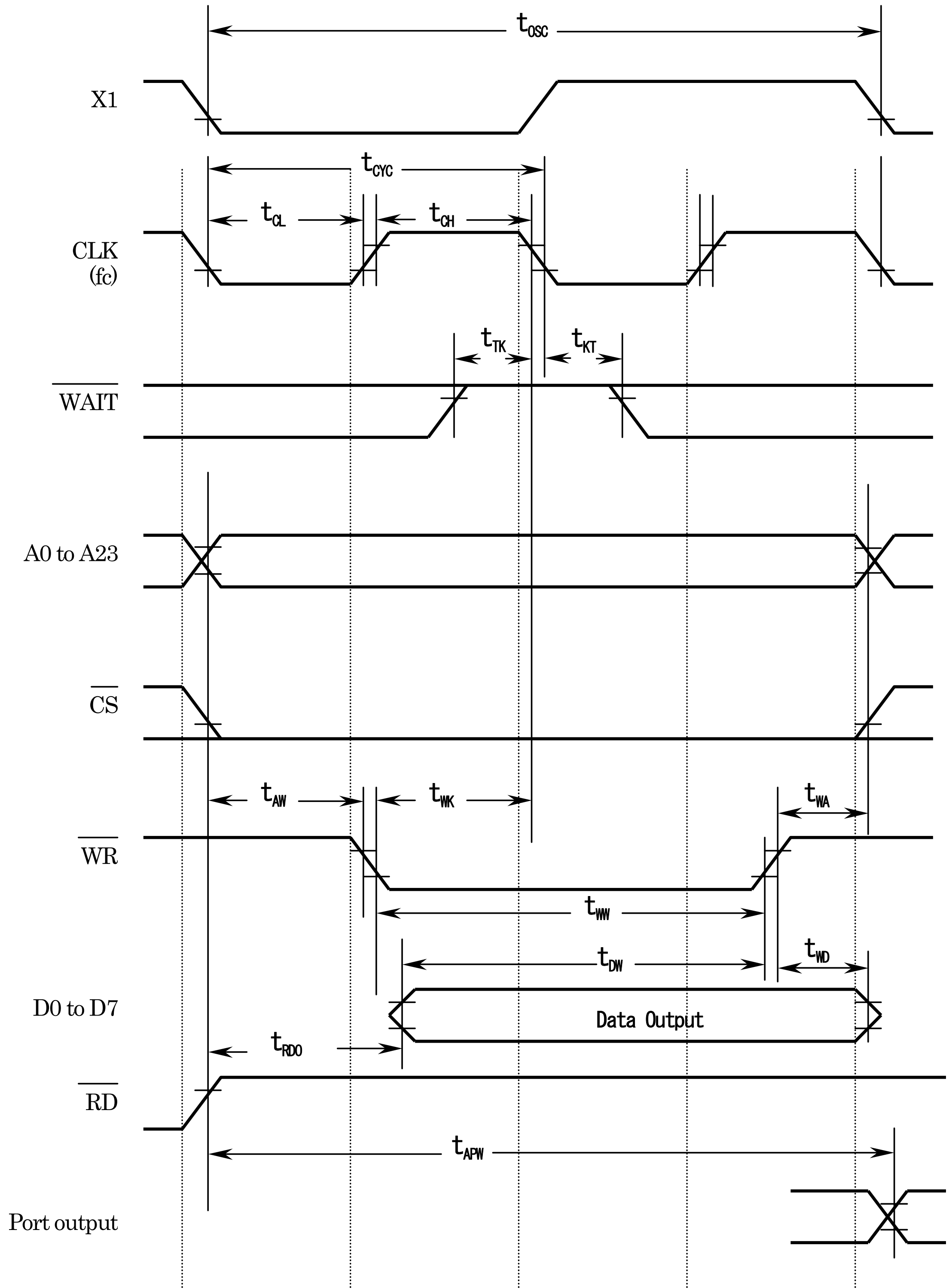
- Output : D0 to D7, A0 to A7, A8 to A15, A16 to A23, $\overline{RD}$, $\overline{WR}$
  High 2.0V, Low 0.8V, CL=50pF
  Others
  High 2.0V, Low 0.8V, CL=50pF
- Input : D0 to D7
  High 2.4V, Low 0.45V, CL=50pF
  Others
  High 0.8×VCC5, Low 0.2×VCC5

(1) Read cycle (0 wait)



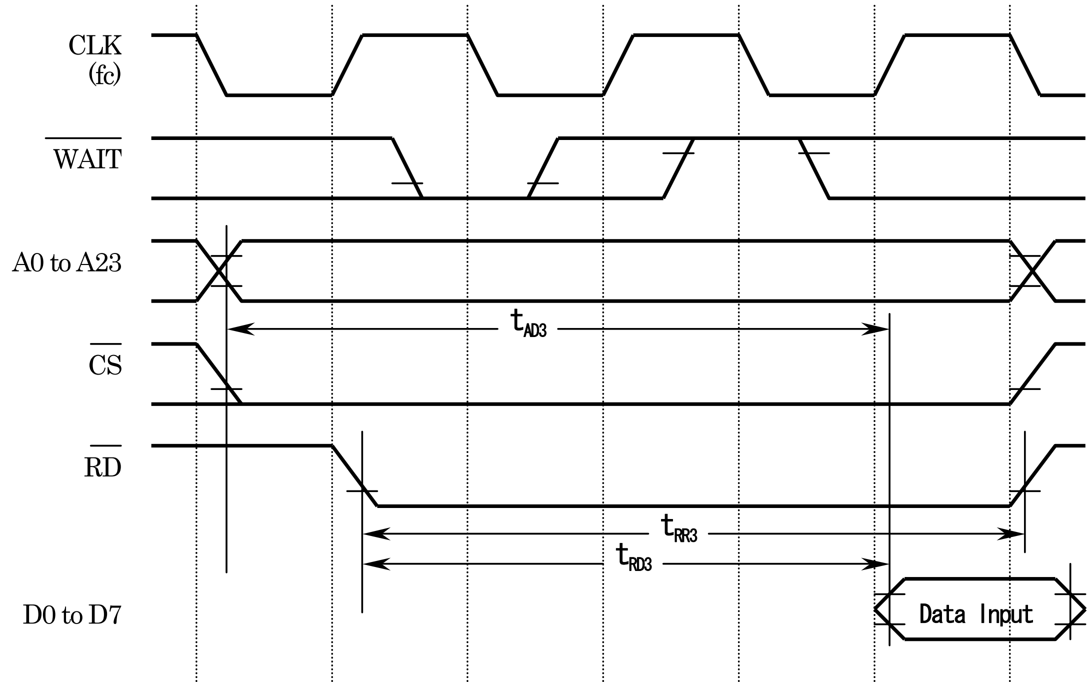
Note : The phase relation between X1 input signal and the other signals is unsettled. The timing chart above is an example.

(2) Write cycle (0 wait)

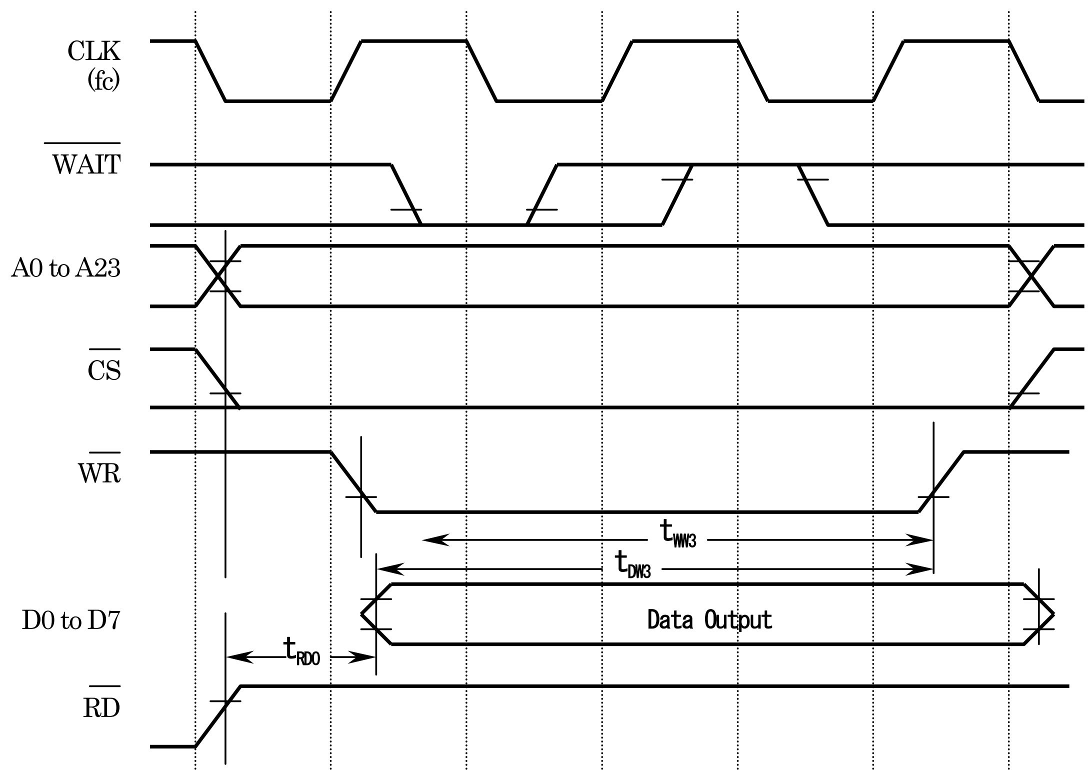


Note : The phase relation between X1 input signal and the other signals is unsettled.
The timing chart above is an example.

(3) Read cycle (1 wait)



(4) Write cycle (1 wait)



4.4 AD Conversion Characteristics

Symbol	Parameter	Min	Typ	MAX	Unit
VREFH	Analog reference voltage(+)	VCC5-0.2	VCC5	VCC5	V
VREFL	Analog reference voltage(−)	VSS5	VSS5	VSS5	
AVCC	AD Converter Power Supply Voltage	VCC5-0.2	VCC5	VCC5	
AVSS	AD Converter Ground	VSS5	VSS5	VSS5	
AVIN	Analog Input Voltage	VREFL		VREFH	
I_{REF}	Analog Current for analog reference voltage <VREFON>=1		0.8	1.2	mA
	<VREFON>=0		0.02	5	uA
E_T	Total error (excluding quantize error)			±3.0	LSB

Note) "LSB" is the UNIT which means the resolution of AD CONVERTER. (+/- 3 LSB = 3 * VCC/1024 = +/-15mV)

4.5 Event Counter (TI0, TI4, TI8, TI9, TIA, TIB)

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock Cycle	t_{VCK}	8T+100		500		600		ns
Clock Low Width	t_{VCKL}	4T+40		240		290		ns
Clock High Width	t_{VCKH}	4T+40		240		290		ns

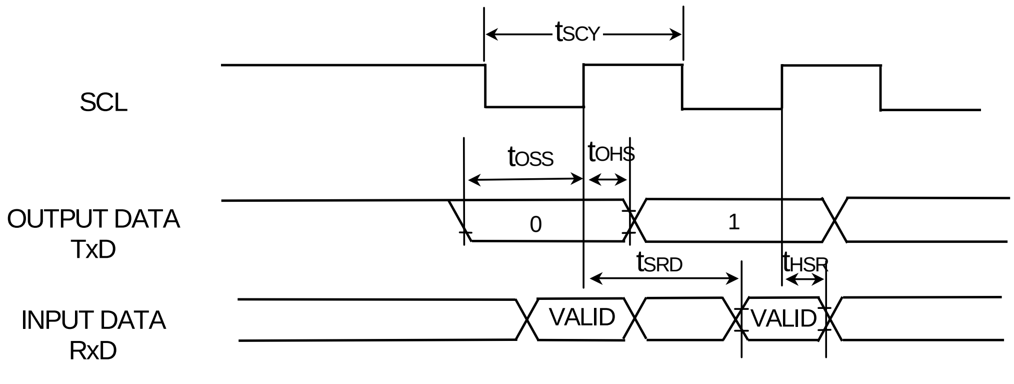
4.6 Serial Channel Timing

(1) SCLK Input mode (I/O Interface mode)

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Cycle	t_{SCY}	16T		0.8		1.0		us
Output Data → SCLK Rise	t_{OSS}	t_{SCY}/2-4T -110		90		140		ns
SCLK Rise → Output Data Hold	t_{OHS}	t_{SCY}/2+2T		500		625		
SCLK Rise → Input Data Hold	t_{HSR}	3T+10		160		197.5		
SCLK Rise → Input Data Valid	t_{SRD}		t_{SCY}		800		1000	

(2) SCLK Output mode (I/O Interface mode)

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Cycle (programmable)	t_{SCY}	16T	8192T	0.8	409.6	1.0	512	us
Output Data → SCLK Rise	t_{OSS}	t_{SCY}/2-40		360		460		ns
SCLK Rise → Output Data Hold	t_{OHS}	t_{SCY}/2-40		360		460		
SCLK Rise → Input Data Hold	t_{HSR}	0		0		0		
SCLK Rise → Input Data Valid	t_{SRD}		t_{SCY}/2-T -180		570		757.5	



(3) SCLK Input mode (UART mode) (Preliminary)

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

Parameter	Symbol	Variable		20MHz		16MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{\text{NMI}}$,INT0 Low Width	T_{INTAL}	4T		200		250		ns
$\overline{\text{NMI}}$,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		

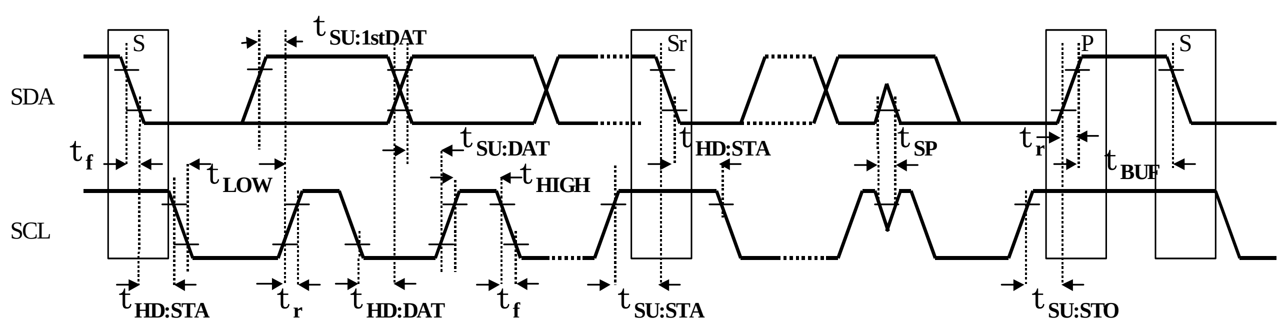
4.8 Serial bus interface

I2CBUS-AC-SPEC TABLE

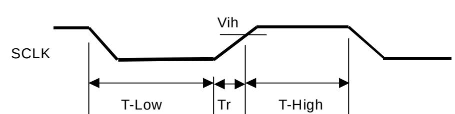
No	PARAMETER	SYMBOL	UNIT	(fc=20MHz) 400KHz MIN	MAX	100KHz MIN	MAX	(fc=System clock) Existing rate MIN	MAX
1	SCL clock frequency	f_{scl}	KHz	0	400	0	100	0	$fc/(2^n+8)$
2	Hold time (repeated) START condition. After this period, the first clock pulse is generated.	$t_{HD:STA}$	ns	650	-	4500	-	$2^{n-1}/fc$	-
3	LOW period of the SCL clock	t_{LOW}	ns	1300	-	4700	-	$2^{n-1}/fc$	-
4	HIGH period of the SCL clock	t_{HIGH}	ns	600	-	4000	-	$(2^{n-1}+8)/fc$	-
5	Set-up time for a repeated START condition	$t_{SU:STA}$	ns	by software		by software		by software	
6	Data hold time: for CBUS compatible masters for I2C-bus devices	$t_{HD:DAT}$	ns	0	900	0	3450	0	6/fc
7	Data set-up time	$t_{SU:DAT}$	ns	100	-	250	-	$(2^{n-1}-6)/fc$	-
7'	Data set-up time (The case in the first bit after transfer)	$t_{SU:1stDAT}$	↑	↑	↑	↑	↑	$(2^{n-1}-12)/fc$	-
8	Rise time of both SDA and ACL signals (*1)	t_r	ns	-	300 (receive)	-	1000 (receive)	-	
9	Fall time of both SDA and ACL signals	t_f	ns	-	300	-	300	-	
10	Set-up time for STOP condition	$t_{SU:STO}$	ns	950	-	4200	-	$(2^{n-1}+12)/fc$	-
11	Bus free time between a STOP and START condition	t_{BUF}	ns	by software		by software		by software	
12	Capacitive load for each cus line	C_b	pF		400		400		400
13	Noise margin at the LOW level for each connected device (including hysteresis)	V_{nL}	v	$0.2V_{DD5}$	-	$0.2V_{DD5}$	-	$0.2V_{DD5}$	-
14	Noise margin at the HIGH level for each connected device (including hysteresis)	V_{nH}	v	$0.2V_{DD5}$	-	$0.2V_{DD5}$	-	$0.2V_{DD5}$	-
15	Pulse width of spikes which must be suppressed by the input filter	t_{sp}	ns	0	50	n/a	n/a	n/a	n/a

Note

1 All values referred to V_{IHmin} and V_{ILmax} levels.



*1) **I2BUS CLK AC SPEC : Tr (Transmitter selection)**



	T-Low	Tr	T-High
SCK(0001 - 0110)	2*(n-1)/fc		(2*(n-1)+8)/fc
SCK(1111) :100KHz	100/fc		100/fc
SCK(1000) :400KHz	32/fc		18/fc

	Tr	0 to 2/fc	2/fc to 4/fc	4/fc to 6/fc	6/fc to 8/fc	8/fc to 10/fc	....
SCK(0001 - 0110)	T-R	0	4/fc		8/fc		....
SCK(1111) :100KHz	T-R	0	4/fc		8/fc		....
SCK(1000) :400KHz	T-R	0	2/fc	4/fc	6/fc	8/fc	....

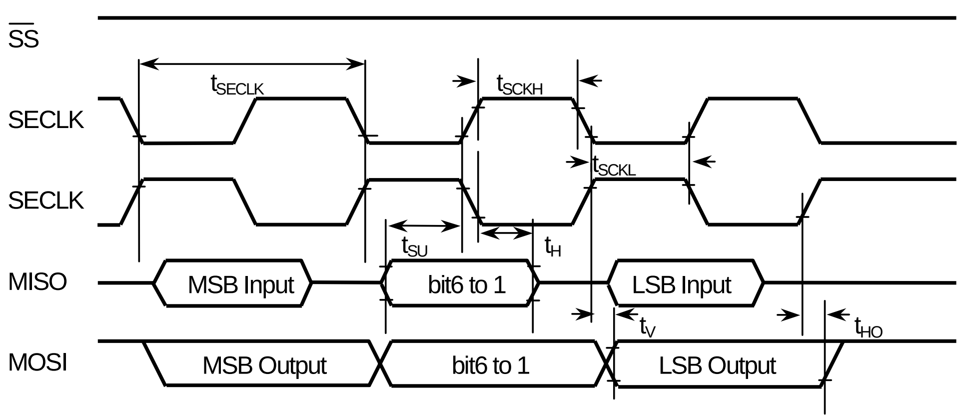
T-period = T-Low + T-R + T-High

Example:in the case of fc=20MHz, SCK3,2,1,0=(0001), Tr=200ns
1) Tr=200ns so T-R=4/fc
2) T-period = 2*(n-1)/fc + 4/fc + (2*(n-1)+8)/fc
=76/fc =3.8us

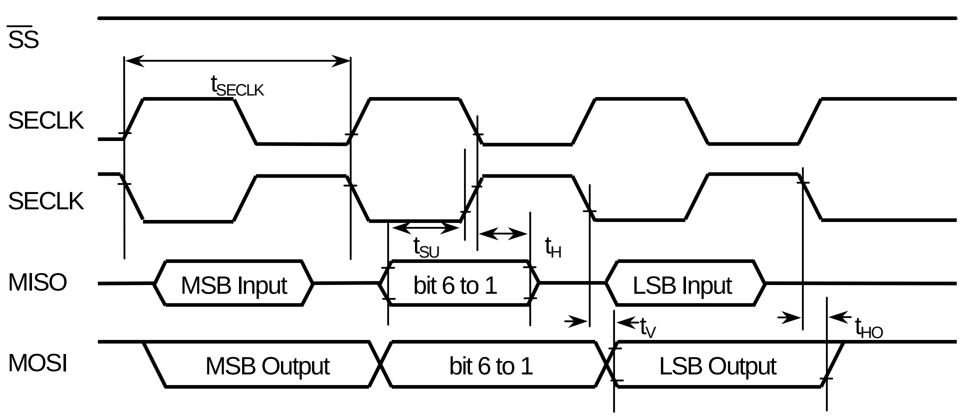
4.9 Serial Expansion Interface

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

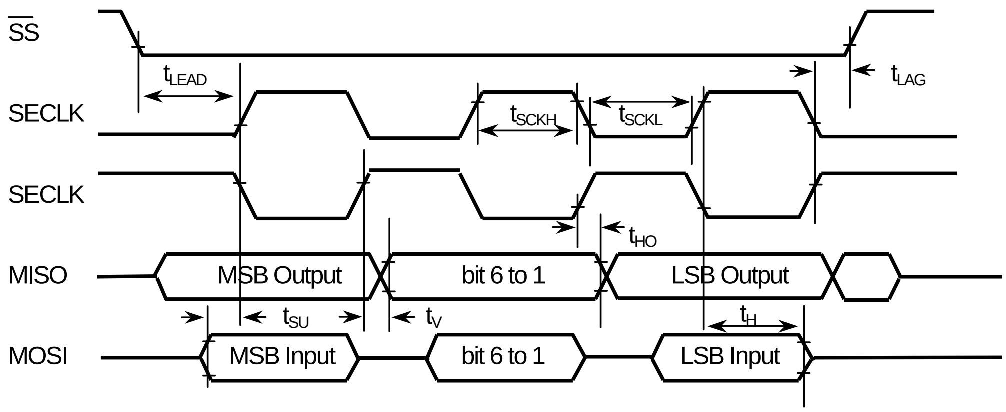
a) SEI Master (CPHA=0)



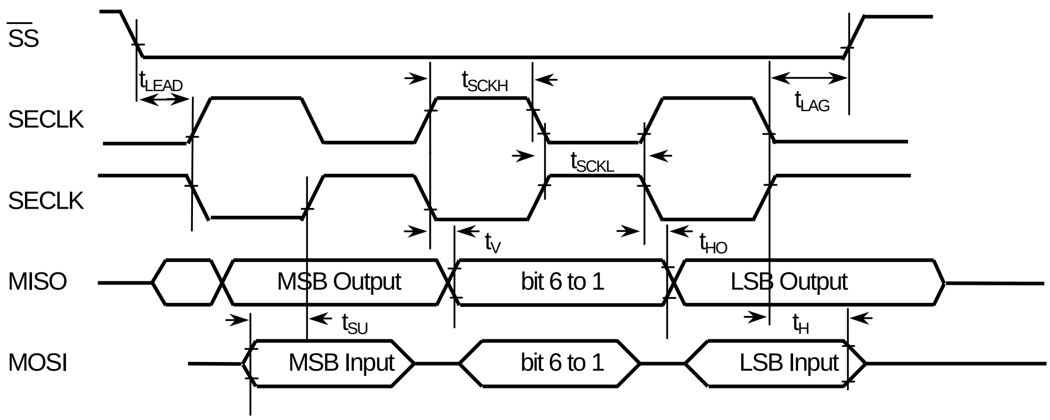
b) SEI Master (CPHA=1)



c) SEI Slave (CPHA=0)

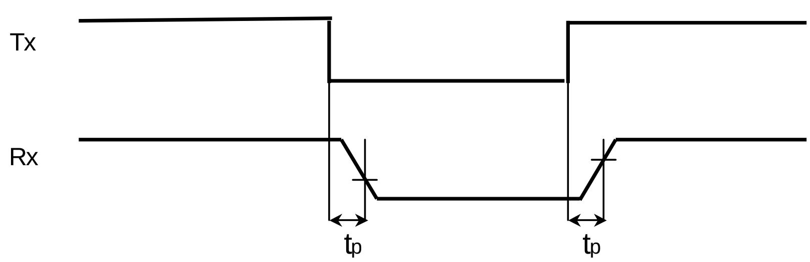


d) SEI Slave (CPHA=1)



4.10 Controller Area Network (CAN)

Symbol	Parameter	Variable		20MHz		Unit
		Min	Max	Min	Max	
t_{cclk}	CAN Clock period	2T		100		ns
t_p	Tx edge →Rx Input		2tcclk-20		180	ns



4.11 Voltage regulator

Voltage Regurator Vcc5 =4.5V to 5.25V / fc = 16 to 20MHz / Ta = -40 to 85 degree C / Iload =10uA

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit.
Output Voltage	REGOUT		3.0	—	3.6	V
Output Current	Iro	Vin-REGOUT=1.0V	0	—	150	mA
Quiescent Current	Iq	Iro≦10 uA	30	50	100	μA
	Iq1	10 uA＜Iro＜100mA (Ta=25℃)	15	250	800	μA
	Iop	Iro=150mA	6	8	10	mA
Standby Current	Is	REGEN=0 (Regulator Only)	—	0.1	0.2	μA

0.5[Ohm] ≦ESR≦5.0[Ohm]

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit.
Stabilization capactor	Cs	Cb=10uF, ESR=4.7Ω	0.1	—	10	μF
Bypass capactor	Cb	Cs=10uF, ESR=4.7Ω (Cs>=Cb)	0.1	—	10	μF
Input capactor	Cin (Note)	Cs=10uF, ESR=4.7Ω	4.7	—	22	μF
Equivalent Series Resistor	ESR	Cs=10uF Cb=0.1uF	0.5	—	5	Ω

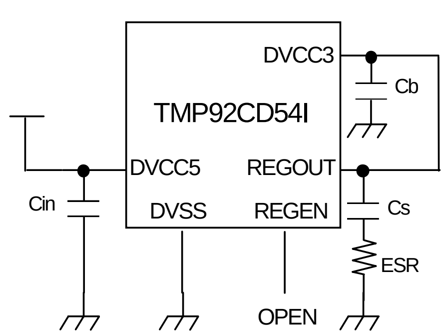
0.5[Ohm] ≦ESR≦50[Ohm]

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit.
Stabilization capactor	Cs	Cb=0.6uF, ESR=47Ω	0.1	—	10	μF
Bypass capactor	Cb	Cs=10uF, ESR=47Ω (Cs>=Cb)	0.6	—	10	μF
Input capactor	Cin (Note)	Cs=10uF, ESR=47Ω	4.7	—	22	μF
Equivalent Series Resistor	ESR	Cs=10uF Cb=0.6uF	0.5	—	50	Ω

0.5[Ohm] ≦ESR≦100[Ohm]

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit.
Stabilization capactor	Cs	Cb=1.0uF, ESR=100Ω	0.1	—	10	μF
Bypass capactor	Cb	Cs=10uF, ESR=100Ω (Cs>=Cb)	1.0	—	10	μF
Input capactor	Cin (Note)	Cs=10uF, ESR=100Ω	4.7	—	22	μF
Equivalent Series Resistor	ESR	Cs=10uF Cb=1.0uF	0.5	—	100	Ω

Note: Recommend Tantalum Capacitor.



5. Table of special function registers (SFRs)
(SFR ; Special Function Register)

The special function registers (SFRs) include the I/O ports and peripheral control registers allocated to the 1024 byte addresses from 000000H to 0003FFH.

(1) I/O port
(2) 8-bit Timer control
(3) 16-bit Timer control
(4) Serial Channel control
(5) Serial Expansion Interface control
(6) Interrupt control
(7) DMA controller
(8) Control register
(9) A/D converter control
(10)Memory controller
(11)Serial Bus Interface control
(12)CAN control
(13)RTC control

Configuration of the table

Symbol	Name	Address	7	6	5	4	3	2	1	0	
											→ bit Symbol
											→ Read/Write
											→ Initial value after reset
											→ Remarks

Explanations of symbols

R/W : Either read or write is possible
R : Only read is possible
W : Only write is possible

no RMW : Prohibit Read Modify Write
(Prohibit RES / SET / TSET / CHG / STCF / ANDCF / ORCF / XORCF etc.)

Table 6 I/O register address map

[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	(Reserved)
16H	(Reserved)
17H	(Reserved)
18H	(Reserved)
19H	(Reserved)
1AH	(Reserved)
1BH	(Reserved)
1CH	P7
1DH	(Reserved)
1EH	P7CR
1FH	P7FC

ADDRESS	NAME
0020H	(Reserved)
21H	(Reserved)
22H	(Reserved)
23H	(Reserved)
24H	(Reserved)
25H	(Reserved)
26H	(Reserved)
27H	(Reserved)
28H	(Reserved)
29H	(Reserved)
2AH	(Reserved)
2BH	(Reserved)
2CH	(Reserved)
2DH	(Reserved)
2EH	(Reserved)
2FH	(Reserved)

ADDRESS	NAME
0030H	PC
31H	(Reserved)
32H	PCCR
33H	PCFC
34H	PD
35H	(Reserved)
36H	PDCR
37H	PDFC
38H	(Reserved)
39H	(Reserved)
3AH	(Reserved)
3BH	(Reserved)
3CH	PF
3DH	(Reserved)
3EH	PFCR
3FH	PFFC

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

[2] SEI :

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 the without allocated names.

[3] 8-bit Timer :

ADDRESS	NAME
0080H	TRUN01
81H	(Reserved)
82H	TREG0
83H	TREG1
84H	TMOD01
85H	TFFCR1
86H	(Reserved)
87H	(Reserved)
88H	TRUN23
89H	(Reserved)
8AH	TREG2
8BH	TREG3
8CH	TMOD23
8DH	TFFCR3
8EH	(Reserved)
8FH	(Reserved)

ADDRESS	NAME
0090H	TRUN45
91H	(Reserved)
92H	TREG4
93H	TREG5
94H	TMOD45
95H	TFFCR5
96H	(Reserved)
97H	(Reserved)
98H	TRUN67
99H	(Reserved)
9AH	TREG6
9BH	TREG7
9CH	TMOD67
9DH	TFFCR7
9EH	(Reserved)
9FH	(Reserved)

[4] 16-bit Timer :

ADDRESS	NAME
00A0H	TRUN8
A1H	(Reserved)
A2H	TMOD8
A3H	TFFCR8
A4H	(Reserved)
A5H	(Reserved)
A6H	(Reserved)
A7H	(Reserved)
A8H	TREG8L
A9H	TREG8H
AAH	TREG9L
ABH	TREG9H
ACH	CAP8L
ADH	CAP8H
AEH	CAP9L
AFH	CAP9H

ADDRESS	NAME
00B0H	TRUNA
B1H	(Reserved)
B2H	TMODA
B3H	TFFCRA
B4H	(Reserved)
B5H	(Reserved)
B6H	(Reserved)
B7H	(Reserved)
B8H	TREGAL
B9H	TREGAH
BAH	TREGBL
BBH	TREGBH
BCH	CAPAL
BDH	CAPAH
BEH	CAPBL
BFH	CAPBH

[5] SIO :

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
00D0H	INTE12
D1H	INTE34
D2H	INTE56
D3H	INTE7
D4H	INTET01
D5H	INTET23
D6H	INTET45
D7H	INTET67
D8H	INTET89
D9H	INTETAB
DAH	INTETO8A
DBH	INTES0
DCH	INTES1
DDH	INTECRT
DEH	INTECG
DFH	INTESEE0

ADDRESS	NAME
00E0H	INTESED0
E1H	INTERTC
E2H	INTESB2
E3H	INTESB0
E4H	INTESB1
E5H	INTMK0
E6H	INTMK1
E7H	INTMK2
E8H	INTMK3
E9H	INTMK4
EAH	INTMK5
EBH	(Reserved)
ECH	WUPFLAG
EDH	WUPMOD
EEH	WUPEDGE
EFH	WUPMASK

ADDRESS	NAME
00F0H	INTE0AD
F1H	INTETC01
F2H	INTETC23
F3H	INTETC45
F4H	INTETC67
F5H	(Reserved)
F6H	IIMC
F7H	INTNMWDT
F8H	INTCLR
F9H	(Reserved)
FAH	(Reserved)
FBH	(Reserved)
FCH	(Reserved)
FDH	(Reserved)
FEH	(Reserved)
FFH	(Reserved)

ADDRESS	NAME
0100H	DMA0V
101H	DMA1V
102H	DMA2V
103H	DMA3V
104H	DMA4V
105H	DMA5V
106H	DMA6V
107H	DMA7V
108H	DMAB
109H	DMAR
10AH	CLKMOD
10BH	(Reserved)
10CH	(Reserved)
10DH	(Reserved)
10EH	(Reserved)
10FH	(Reserved)

[7] WDT, RTC :

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 (Note)
16CH	(Reserved)
16DH	RAMCR
16EH	FLSR (Note)
16FH	(Reserved)

(Note) Only TMP92FD54AI.

[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)

ADDRESS	NAME
0180H	SBI2CR1
181H	SBI2DBR
182H	I2C2AR
183H	SBI2CR2/SBI2SR
184H	SBI2BR0
185H	SBI2BR1
186H	(Reserved)
187H	(Reserved)
188H	(Reserved)
189H	(Reserved)
18AH	(Reserved)
18BH	(Reserved)
18CH	(Reserved)
18DH	(Reserved)
18EH	(Reserved)
18FH	(Reserved)

ADDRESS	NAME
0190H	(Reserved)
191H	(Reserved)
192H	(Reserved)
193H	(Reserved)
194H	(Reserved)
195H	(Reserved)
196H	(Reserved)
197H	(Reserved)
198H	(Reserved)
199H	(Reserved)
19AH	(Reserved)
19BH	(Reserved)
19CH	(Reserved)
19DH	(Reserved)
19EH	(Reserved)
19FH	(Reserved)

ADDRESS	NAME
01A0H	(Reserved)
1A1H	(Reserved)
1A2H	(Reserved)
1A3H	(Reserved)
1A4H	(Reserved)
1A5H	(Reserved)
1A6H	(Reserved)
1A7H	(Reserved)
1A8H	(Reserved)
1A9H	(Reserved)
1AAH	(Reserved)
1ABH	(Reserved)
1ACH	(Reserved)
1ADH	(Reserved)
1AEH	(Reserved)
1AFH	(Reserved)

ADDRESS	NAME
01B0H	(Reserved)
1B1H	(Reserved)
1B2H	(Reserved)
1B3H	(Reserved)
1B4H	(Reserved)
1B5H	(Reserved)
1B6H	(Reserved)
1B7H	(Reserved)
1B8H	(Reserved)
1B9H	(Reserved)
1BAH	(Reserved)
1BBH	(Reserved)
1BCH	(Reserved)
1BDH	(Reserved)
1BEH	(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)

[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

[11] CAN:

ADDRESS	NAME
0240H	MB4MI0L
241H	MB4MI0H
242H	MB4MI1L
243H	MB4MI1H
244H	MB4MCFL
245H	MB4MCFH
246H	MB4D0
247H	MB4D1
248H	MB4D2
249H	MB4D3
24AH	MB4D4
24BH	MB4D5
24CH	MB4D6
24DH	MB4D7
24EH	MB4TSVL
24FH	MB4TSVH

ADDRESS	NAME
0250H	MB5MI0L
251H	MB5MI0H
252H	MB5MI1L
253H	MB5MI1H
254H	MB5MCFL
255H	MB5MCFH
256H	MB5D0
257H	MB5D1
258H	MB5D2
259H	MB5D3
25AH	MB5D4
25BH	MB5D5
25CH	MB5D6
25DH	MB5D7
25EH	MB5TSVL
25FH	MB5TSVH

ADDRESS	NAME
0260H	MB6MI0L
261H	MB6MI0H
262H	MB6MI1L
263H	MB6MI1H
264H	MB6MCFL
265H	MB6MCFH
266H	MB6D0
267H	MB6D1
268H	MB6D2
269H	MB6D3
26AH	MB6D4
26BH	MB6D5
26CH	MB6D6
26DH	MB6D7
26EH	MB6TSVL
26FH	MB6TSVH

ADDRESS	NAME
0270H	MB7MI0L
271H	MB7MI0H
272H	MB7MI1L
273H	MB7MI1H
274H	MB7MCFL
275H	MB7MCFH
276H	MB7D0
277H	MB7D1
278H	MB7D2
279H	MB7D3
27AH	MB7D4
27BH	MB7D5
27CH	MB7D6
27DH	MB7D7
27EH	MB7TSVL
27FH	MB7TSVH

ADDRESS	NAME
0280H	MB8MI0L
281H	MB8MI0H
282H	MB8MI1L
283H	MB8MI1H
284H	MB8MCFL
285H	MB8MCFH
286H	MB8D0
287H	MB8D1
288H	MB8D2
289H	MB8D3
28AH	MB8D4
28BH	MB8D5
28CH	MB8D6
28DH	MB8D7
28EH	MB8TSVL
28FH	MB8TSVH

ADDRESS	NAME
0290H	MB9MI0L
291H	MB9MI0H
292H	MB9MI1L
293H	MB9MI1H
294H	MB9MCFL
295H	MB9MCFH
296H	MB9D0
297H	MB9D1
298H	MB9D2
299H	MB9D3
29AH	MB9D4
29BH	MB9D5
29CH	MB9D6
29DH	MB9D7
29EH	MB9TSVL
29FH	MB9TSVH

ADDRESS	NAME
02A0H	MB10MI0L
2A1H	MB10MI0H
2A2H	MB10MI1L
2A3H	MB10MI1H
2A4H	MB10MCFL
2A5H	MB10MCFH
2A6H	MB10D0
2A7H	MB10D1
2A8H	MB10D2
2A9H	MB10D3
2AAH	MB10D4
2ABH	MB10D5
2ACH	MB10D6
2ADH	MB10D7
2AEH	MB10TSVL
2AFH	MB10TSVH

ADDRESS	NAME
02B0H	MB11MI0L
2B1H	MB11MI0H
2B2H	MB11MI1L
2B3H	MB11MI1H
2B4H	MB11MCFL
2B5H	MB11MCFH
2B6H	MB11D0
2B7H	MB11D1
2B8H	MB11D2
2B9H	MB11D3
2BAH	MB11D4
2BBH	MB11D5
2BCH	MB11D6
2BDH	MB11D7
2BEH	MB11TSVL
2BFH	MB11TSVH

[11] CAN:

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

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
340H : : : : : 3FFH	(Reserved)

(1) I/O Port

Port0

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
P0	PORT0 Register	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P0CR	PORT0 Control Register	02H (no RMW)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
P0FC	PORT0 Function Register	03H (no RMW)	–	–	–	–	–	–	–	P0F
										W
			–	–	–	–	–	–	–	0
			0:PORT 1:Data Bus(D7 to D0)							

Port4

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
P4	PORT4 Register	10H	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P4CR	PORT4 Control Register	12H (no RMW)	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
P4FC	PORT4 Function Register	13H (no RMW)	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
			W							
			0	0	0	0	0	0	0	0
			0:PORT 1:A7	0:PORT 1:A6	0:PORT 1:A5	0:PORT 1:A4	0:PORT 1:A3	0:PORT 1:A2	0:PORT 1:A1	0:PORT 1:A0

P4CR	P4FC	P47	P46	P45	P44	P43	P42	P41	P40
0	0	Input Port							
1	0	Output Port							
1	1	(Reserved)							
0	1	A7 to A0							

Port7

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
P7	PORT7 Register	1CH	–	–	P75	P74	P73	P72	P71	P70
					R/W					
			–	–	0	1	1	1	1	1
					Input/Output					
P7CR	PORT7 Control Register	1EH (no RMW)	–	–	P75C	P74C	P73C	P72C	P71C	P70C
					W					
			–	–	0	1	1	0	1	1
					0:Input 1:Output					
P7FC	PORT7 Function Register	1FH (no RMW)	–	–	P75F	P74F	P73F	P72F	P71F	P70F
					W					
			–	–	0	0	0	0	0	0
					0:PORT 1:$\overline{WAIT}$	0:PORT	0:PORT 1:$\overline{CS}$	0:PORT 1:SI2 SCL2 Note1	0:PORT 1:$\overline{WR}$	0:PORT 1:$\overline{RD}$

Note1: P72 SCL2, clock input/output at I2C mode, can be open-drain output by setting 1 to PNODE<ODE72>.

PortC

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PC	PORTC Register	30H	–	–	PC5	PC4	PC3	PC2	PC1	PC0
					R/W					
			–	–	0	0	0	0	0	0
					Input/Output					
PCCR	PORTC Control Register	32H (no RMW)	–	–	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
					W					
			–	–	0	0	0	0	0	0
					0:Input 1:Output					
PCFC	PORTC Function Register	33H (no RMW)	–	–	PC5F	PC4F	PC3F	PC2F	PC1F	PC0F
					W					
			–	–	0	0	0	0	0	0
					0:PORT INT4 1:TO7	0:PORT 1:TO5	0:PORT INT3 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	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:T0B A23	0:PORT WUINT6 1:T0A A22	0:PORT TIB WUINT5 1:A21	0:PORT INT7 TIA WUINT4 1:A20	0:PORT WUINT3 1:T09 A19	0:PORT WUINT2 1:T08 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	T0B	T0A	TIB, WUINT5	TIA, INT7, WUINT4	T09	T08	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	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)

PortG

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PG	PORTG Register	40H	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
			R							
			Input							

PortL

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
PL	PORTL Register	54H	–	–	–	–	PL3	PL2	PL1	PL0
							R			
			–	–	–	–	Input			

PortM

SYMBOL	NAME	Address	7	6	5	4	3	2	1	0
PM	PORTM	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:$\overline{SS}$ 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	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	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.		

*To switch P72-output from push-pull type to Open-drain type, set 1 to PNODE<ODE72>.

(2) 8-bit Timer

8-Bit Timer 01,23,45,67

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
TRUN01	8bit Timer01 Run Register	80H	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	8bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG0	8Bit Timer Register 0	82H (no RMW)	–							
			W							
			Undefined							
TREG1	8Bit Timer Register 1	83H (no RMW)	–							
			W							
			Undefined							
TMOD01	8Bit Timer0,1 Source CLK & MODE Register	84H	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8bit Timer 01:16bit Timer 10:8bit PPG 11:8bit PWM		PWM cycle 00:reserved 01:2^6 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	8bit Timer23 Run Register	88H	T2RDE	–	–	–	I2T23	T23PRUN	T3RUN	T2RUN
			R/W				R/W	R/W		
			0	–	–	–	0	0	0	0
			Double Buffer 0:Disable 1:Enable				IDLE2 0:Stop 1:Operate	8bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG2	8Bit Timer Register 2	8AH (no RMW)	–							
			W							
			Undefined							
TREG3	8Bit Timer Register 3	8BH (no RMW)	–							
			W							
			Undefined							
TMOD23	8Bit Timer2,3 Source CLK & MODE Register	8CH	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8bit Timer 01:16bit Timer 10:8bit PPG 11:8bit PWM		PWM cycle 00:reserved 01:2^6 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	8bit Timer45 Run Register	90H	T4RDE	–	–	–	I2T45	T45PRUN	T5RUN	T4RUN
			R/W				R/W	R/W		
			0	–	–	–	0	0	0	0
			Double Buffer 0:Disable 1:Enable				IDLE2 0:Stop 1:Operate	8bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG4	8Bit Timer Register 4	92H (no RMW)	–							
			W							
			Undefined							
TREG5	8Bit Timer Register 5	93H (no RMW)	–							
			W							
			Undefined							
TMOD45	8Bit Timer4,5 Source CLK & MODE Register	94H	T45M1	T45M0	PWM41	PWM40	T5CLK1	T5CLK0	T4CLK1	T4CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Operate mode 00:8bit Timer 01:16bit Timer 10:8bit PPG 11:8bit PWM		PWM cycle 00:reserved 01:2^6 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	8bit Timer67 Run Register	98H	T6RDE	–	–	–	I2T67	T67PRUN	T7RUN	T6RUN
			R/W				R/W	R/W		
			0	–	–	–	0	0	0	0
			Double Buffer 0:Disable 1:Enable				IDLE2 0:Stop 1:Operate	8bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TREG6	8Bit Timer Register 6	9AH (no RMW)	–							
			W							
			Undefined							
TREG7	8Bit Timer Register 7	9BH (no RMW)	–							
			W							
			Undefined							

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
TMOD67	8Bit 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 00:8bit Timer 01:16bit Timer 10:8bit PPG 11:8bit 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 Timer

16-Bit Timer 8,A

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
TRUN8	16bit Timer8 Run Register	A0H	T8RDE	–	–	–	I2T8	T8PRUN	–	T8RUN
			R/W	R/W			R/W	R/W		R/W
			0	0	–	–	0	0	–	0
			Double Buffer 0:Disable 1:Enable	Fix to "0"			IDLE2 0:Stop 1:Operate	16bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TMOD8	16bit Timer8 Source CLK & Mode Register	A2H	CAP9T9	EQ9T9	CAP8IN	CAP89M1	CAP89M0	T8CLE	T8CLK1	T8CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TFF9 invert trigger 0: Disable 1: Enable		0:Soft Capture 1:Don't care	Capture Timing 00:disable 01:TI8↑ TI9↑ 10:TI8↑ TI8↓ 11:TFF1↑ TFF1↓		1:UC8 Clear Enable	Source Clock 00:TI8 01:ϕT1 10:ϕT4 11:ϕT16	
TFFCR8	16Bit Timer8 Flip-Flop Control Register	A3H	TFF9C1	TFF9C0	CAP9T8	CAP8T8	EQ9T8	EQ8T8	TFF8C1	TFF8C0
			W		R/W				W	
			1	1	0	0	0	0	1	1
			00:Invert TFF9 01:Set TFF9 10:Clear TFF9 11:Don't Care		TFF8 invert trigger 0: Disable 1: Enable				00:Invert TFF8 01:Set TFF8 10:Clear TFF8 11:Don't Care	
TREG8L	16Bit Timer Register 8 Low	A8H (no RMW)	–							
			W							
			Undefined							
TREG8H	16Bit Timer Register 8 High	A9H (no RMW)	–							
			W							
			Undefined							
TREG9L	16Bit Timer Register 9 Low	AAH (no RMW)	–							
			W							
			Undefined							
TREG9H	16Bit Timer Register 9 High	ABH (no RMW)	–							
			W							
			Undefined							
CAP8L	Capture Register 8 Low	ACH	–							
			R							
			Undefined							
CAP8H	Capture Register 8 High	ADH	–							
			R							
			Undefined							
CAP9L	Capture Register 9 Low	AEH	–							
			R							
			Undefined							
CAP9H	Capture Register 9 High	AFH	–							
			R							
			Undefined							

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
TRUNA	16bit TimerA Run Register	B0H	TARDE	–	–	–	I2TA	TAPRUN	–	TARUN
			R/W	R/W			R/W	R/W		R/W
			0	0	–	–	0	0	–	0
			Double Buffer 0:Disable 1:Enable	Fix to "0"			IDLE2 0:Stop 1:Operate	16bit Timer Run/Stop Control 0:Stop & Clear 1:Run (Count up)		
TMODA	16bit TimerA Source CLK & Mode Register	B2H	CAPBTB	EQBTB	CAPAIN	CAPABM1	CAPABM0	TACLE	TACLK1	TACLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TFFB invert trigger 0: Disable 1: Enable		0:Soft Capture 1:Don't care	Capture Timing 00:disable 01:TIA↑ TIB↑ 10:TIA↑ TIA↓ 11:TFF1↑ TFF1↓		1:UCA Clear Enable	Source Clock 00:TIA 01:φT1 10:φT4 11:φT16	
TFFCRA	16Bit TimerA Flip-Flop Control Register	B3H	TFFBC1	TFFBC0	CAPBTA	CAPATA	EQBTA	EQATA	TFFAC1	TFFAC0
			W		R/W				W	
			1	1	0	0	0	0	1	1
			00:Invert TFFB 01:Set TFFB 10:Clear TFFB 11:Don't Care		TFFA invert trigger 0: Disable 1: Enable				00:Invert TFFA 01:Set TFFA 10:Clear TFFA 11:Don't Care	
TREGAL	16Bit Timer Register A Low	B8H (no RMW)	–							
			W							
			Undefined							
TREGAH	16Bit Timer Register A High	B9H (no RMW)	–							
			W							
			Undefined							
TREGBL	16Bit Timer Register B Low	BAH (no RMW)	–							
			W							
			Undefined							
TREGBH	16Bit Timer Register B High	BBH (no RMW)	–							
			W							
			Undefined							
CAPAL	Capture Register A Low	BCH	–							
			R							
			Undefined							
CAPAH	Capture Register A High	BDH	–							
			R							
			Undefined							
CAPBL	Capture Register B Low	BEH	–							
			R							
			Undefined							
CAPBH	Capture Register B High	BFH	–							
			R							
			Undefined							

(4) Serial Channels

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer Register	C0H (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 Overrun	Parity	Framing	0:SCLK0↑ 1:SCLK0↓	0:Baud Rate Generator 1:SCLK0 Pin Input
SC0MOD0	Serial Channel 0 Mode 0 Register	C2H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmission Databit 8	0:CTS Disable 1:CTS Enable	0:Receive Disable 1:Receive Enable	Wake up 0:Disable 1:Enable	00:I/O Interface Mode 01:7bit UART Mode 10:8bit UART Mode 11:9bit UART Mode		00:TimerT0TRG 01:Baud Rate Generator 10:Internal clock 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 Overrun	Parity	Framing	0:SCLK1↑ 1:SCLK1↓	0:Baud Rate Generator 1:SCLK1 Pin Input
SC1MOD0	Serial Channel 1 Mode 0 Register	CAH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			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:TimerT0TRG 01:Baud Rate Generator 10:Internal clock $\phi 1$ 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: $\phi T0$ 01: $\phi T2$ 10: $\phi T8$ 11: $\phi T32$		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)

<table>
<tr><th>Symbol</th><th>Name</th><th>ADDRESS</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="4">SECR</td><td rowspan="4">SEI
Control
Register</td><td rowspan="4">60H</td><td>MODE</td><td>SEE</td><td>BOS</td><td>MSTR</td><td>CPOL</td><td>CPHA</td><td>SER1</td><td>SER0</td></tr>
<tr><td>W</td><td colspan="7">R/W</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>SEI0 MODF
Detection
0:Enable
1:Disable</td><td>SEISystem
Enable
0:Stop
1:Run</td><td>Bit Order
Select
bit
0:MSB
1:LSB</td><td>Master
Select
bit
0:Slave
1:Master</td><td>Clock
polarity
selection
See figure
3.11.2,
3.11.3</td><td>Clock
Phase
Selection
See figure
3.11.2,
3.11.3</td><td colspan="2">SEI Transfer Rate
Select
00:reserved
01:Divided by 2
10:Divided by 4
11:Divided by 16</td></tr>
<tr><td rowspan="8">SESR</td><td rowspan="8">SEI Status
Register</td><td rowspan="8">61H</td><td>SEF</td><td>WCOL</td><td>SOVF</td><td>MODF</td><td>–</td><td>–</td><td>–</td><td>TMSE</td></tr>
<tr><td colspan="4">R</td><td></td><td></td><td></td><td>R/W</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>–</td><td>–</td><td>–</td><td>0</td></tr>
<tr><td>SEI
Transfer
0:busy or
Stop
1:End</td><td>WCOL Flag
1:Error</td><td>SOVF Flag
(Slave)
1:Error</td><td>MODF Flag
(Master)
1:Error</td><td></td><td></td><td></td><td>SEI Mode
Select
0:Compat
ibility
Mode
1:Micro
DMA Mode</td></tr>
<tr><td>–</td><td>WCOL</td><td>SOVF</td><td>MODF</td><td>TSRC</td><td>TSTC</td><td>TASM</td><td>TMSE</td></tr>
<tr><td></td><td colspan="5">R</td><td colspan="2">R/W</td></tr>
<tr><td>–</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td></td><td>WCOL Flag
1:Error</td><td>SOVF Flag
(Slave)
1:Error</td><td>MODF Flag
(Master)
1:Error</td><td>SEI
Receive
1:End</td><td>SEI
Transfer
1:End</td><td>Auto
Shift
Enable
(Master)
INTSEE0
Mask
(Slave)</td><td>SEI Mode
Select
0:Compat
ibility
Mode
1:Micro
DMA Mode</td></tr>
<tr><td rowspan="4">SEDR</td><td rowspan="4">SEI Data
Register</td><td rowspan="4">62H</td><td>SED7</td><td>SED6</td><td>SED5</td><td>SED4</td><td>SED3</td><td>SED2</td><td>SED1</td><td>SED0</td></tr>
<tr><td colspan="8">R/W</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="8">Transfer/Receive Data</td></tr>
</table>

(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	I0C	I0M2	I0M1	I0M0
			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
INTET08A	INTT08 & INTT0A (Overflow) Enable Register	DAh	INTT0A				INTT08			
			IT0AC	IT0AM2	IT0AM1	IT0AM0	IT08C	IT08M2	IT08M1	IT08M0
			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 register

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
			Stand by mode 00:IDLE3 mode 01:STOP mode 10:IDLE1 mode 11:IDLE2 mode			Fixed to "0"		CLK Output Enable 0 : Not output 1 : Output	00:fc output 01:(reserved) 10:2/5·fc output 11:(reserved)	
WDMOD	Watchdog Timer Mode Register	110H	WDTE	WDTP1	WDTP0	–	DRVE	I2WDT	RESCR	–
			R/W				R/W			
			1	0	0	–	0	0	0	0
			1:WDT Enable	00 : 2^{16}/fc 01 : 2^{18}/fc 10 : 2^{20}/fc 11 : 2^{22}/fc			1:Drive pin in STOP mode	IDLE2 0:Stop 1:Operate	1:Reset connect internally WDT out to RESET pin	Fix to "0"
WDCR	Watchdog Timer Control Register	111H	–							
			W							
			–							
			B1H : WDT Disable 4EH : WDT Clear							

(9) AD converter

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
ADMOD0	AD Mode Control Register 0	138H	EOCF	ADBF	–	–	ITM0	REPET	SCAN	ADS
			R				R/W			
			0	0	0	0	0	0	0	0
			AD Conversion End Flag 1:END	AD Conversion BUSY Flag 1:Busy	Fix to "0"	Fix to "0"	0: Every 1 time 1: Every 4 times	Repeat mode 0:Single mode 1:Repeat mode	Scan mode 0:Fixed channel mode 1:Channel scan mode	AD Conversion start 1:Start Always read as "0"
ADMOD1	AD Mode Control Register 1	139H	VREFON	I2AD	–	–	ADCH3	ADCH2	ADCH1	ADCH0
			R/W	R/W			R/W			
			0	0	0	0	0	0	0	0
			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
			Undefined		–	–	–	–	–	0
ADREG0H	AD Result Register 0 High	121H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
ADREG1L	AD Result Register 1 Low	122H	ADR11	ADR10	–	–	–	–	–	ADR1RF
			R							R
			Undefined		–	–	–	–	–	0
ADREG1H	AD Result Register 1 High	123H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
ADREG2L	AD Result Register 2 Low	124H	ADR21	ADR20	–	–	–	–	–	ADR2RF
			R							R
			Undefined		–	–	–	–	–	0
ADREG2H	AD Result Register 2 High	125H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
ADREG3L	AD Result Register 3 Low	126H	ADR31	ADR30	–	–	–	–	–	ADR3RF
			R							R
			Undefined		–	–	–	–	–	0
ADREG3H	AD Result Register 3 High	127H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
ADREG4L	AD Result Register 4 Low	128H	ADR41	ADR40	–	–	–	–	–	ADR4RF
			R							R
			Undefined		–	–	–	–	–	0
ADREG4H	AD Result Register 4 High	129H	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
			R							
			Undefined							
ADREG5L	AD Result Register 5 Low	12AH	ADR51	ADR50	–	–	–	–	–	ADR5RF
			R							R
			Undefined		–	–	–	–	–	0
ADREG5H	AD Result Register 5 High	12BH	ADR59	ADR58	ADR57	ADR56	ADR55	ADR54	ADR53	ADR52
			R							
			Undefined							
ADREG6L	AD Result Register 6 Low	12CH	ADR61	ADR60	–	–	–	–	–	ADR6RF
			R							R
			Undefined		–	–	–	–	–	0
ADREG6H	AD Result Register 6 High	12DH	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
			R							
			Undefined							
ADREG7L	AD Result Register 7 Low	12EH	ADR71	ADR70	–	–	–	–	–	ADR7RF
			R							R
			Undefined		–	–	–	–	–	0
ADREG7H	AD Result Register 7 High	12FH	ADR79	ADR78	ADR77	ADR76	ADR75	ADR74	ADR73	ADR72
			R							
			Undefined							
ADREG8L	AD Result Register 8 Low	130H	ADR81	ADR80	–	–	–	–	–	ADR8RF
			R							R
			Undefined		–	–	–	–	–	0
ADREG8H	AD Result Register 8 High	131H	ADR89	ADR88	ADR87	ADR86	ADR85	ADR84	ADR83	ADR82
			R							
			Undefined							
ADREG9L	AD Result Register 9 Low	132H	ADR91	ADR90	–	–	–	–	–	ADR9RF
			R							R
			Undefined		–	–	–	–	–	0
ADREG9H	AD Result Register 9 High	133H	ADR99	ADR98	ADR97	ADR96	ADR95	ADR94	ADR93	ADR92
			R							
			Undefined							
ADREGAL	AD Result Register A Low	134H	ADRA1	ADRA0	–	–	–	–	–	ADRARF
			R							R
			Undefined		–	–	–	–	–	0
ADREGAH	AD Result Register A High	135H	ADRA9	ADRA8	ADRA7	ADRA6	ADRA5	ADRA4	ADRA3	ADRA2
			R							
			Undefined							
ADREGBL	AD Result Register B Low	136H	ADRB1	ADRB0	–	–	–	–	–	ADRBRF
			R							R
			Undefined		–	–	–	–	–	0
ADREGBH	AD Result Register B High	137H	ADRB9	ADRB8	ADRB7	ADRB6	ADRB5	ADRB4	ADRB3	ADRB2
			R							
			Undefined							

(10) Memory controller

Symbol	Name	ADDRESS	7	6	5	4	3	2	1	0
BCSL	BLOCK CS/WAIT Control Register Low	148H	–	BWW2	BWW1	BWW0	–	BWR2	BWR1	BWR0
				W				W		
			–	0	1	0	–	0	1	0
				Number of write waits 001:0wait 010:1wait 011:Nwait 101:2wait 110:3wait others : reserved				Number of read waits 001:0wait 010:1wait 011:Nwait 101:2wait 110:3wait others : reserved		
BCSH	BLOCK CS/WAIT Control Register High	149H	BE	BM	–	–	BOM1	BOM0	BBUS1	BBUS0
			W	W			W		W	
			1	0	0	0	0	0	0	0
			CS select 0:Disable 1:Enable	0:16MB 1:Sets area	Fix to "0"	Fix to "0"	00:SRAM/ROM 01,10,11:Resetved		00:8bit 01,10,11:reserved	
MAMR	Memory Address Mask Register	14AH	MV22	MV21	MV20	MV19	MV18	MV17	MV16	MV15
			R/W							
			1	1	1	1	1	1	1	1
			0:Compare enable 1:Compare disable							
MSAR	Memory Start Address Register	14BH	MS23	MS22	MS21	MS20	MS19	MS18	MS17	MS16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
FSWE *note2	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						
FLSR *note2	Flash Status Register	16FH	–	–	–	–	–	R/BSY	–	–
			R/W	R/W	R/W	R/W		R		
			0	0	0	0	–	1	–	–
			Note) Set to 0.	Note) Set to 0.	Note) Set to 0.	Note) Set to 0.		Ready /Busy flag 0:Busy (auto operation in progress) 1:Ready (auto operation finished)		

Note1: After power-on reset. Warm reset does not change this bit.

Note2: Only TMP92FD54AI.

(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	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.	
		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	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"
		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	–	–	–	–	–	–	–
			Clock control 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	–	–	–	–	–	–	–
			Clock control 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	–	–	–	–	–	–	–
			Clock control 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							
			–	–	–	–	–	–	–	–

* MBn = 200H + n x 10H, n = 0, 1, 2, 3, … , 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 (no RMW)	RML7	RML6	RML5	RML4	RML3	RML2	RML1	RML0
			R/C							
			0	0	0	0	0	0	0	0
RMLH	Receive Message Lost Register H	30FH (no RMW)	RML15	RML14	RML13	RML12	RML11	RML10	RML9	RML8
			R/C							
			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 Acceptance Mask Register 0L	310H	LAM23	LAM22	LAM21	LAM20	LAM19	LAM18	LAM17	LAM16
			R/W							
			0	0	0	0	0	0	0	0
LAM0H	Local Acceptance Mask Register 0H	311H	LAMI	–	–	LAM28	LAM27	LAM26	LAM25	LAM24
			R/W			R/W				
			0	–	–	0	0	0	0	0
LAM1L	Local Acceptance Mask Register 1L	312H	LAM7	LAM6	LAM5	LAM4	LAM3	LAM2	LAM1	LAM0
			R/W							
			0	0	0	0	0	0	0	0
LAM1H	Local Acceptance 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 (no RMW)	RFP7	RFP6	RFP5	RFP4	RFP3	RFP2	RFP1	RFP0
			R/C							
			0	0	0	0	0	0	0	0
RFPH	Remote Frame Pending Register H	32DH (no RMW)	RFP15	RFP14	RFP13	RFP12	RFP11	RFP10	RFP9	RFP8
			R/C							
			–	–	–	–	–	–	–	–
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 control

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 Section Equivalent Circuit Diagram.

- Reading The Circuit Diagram

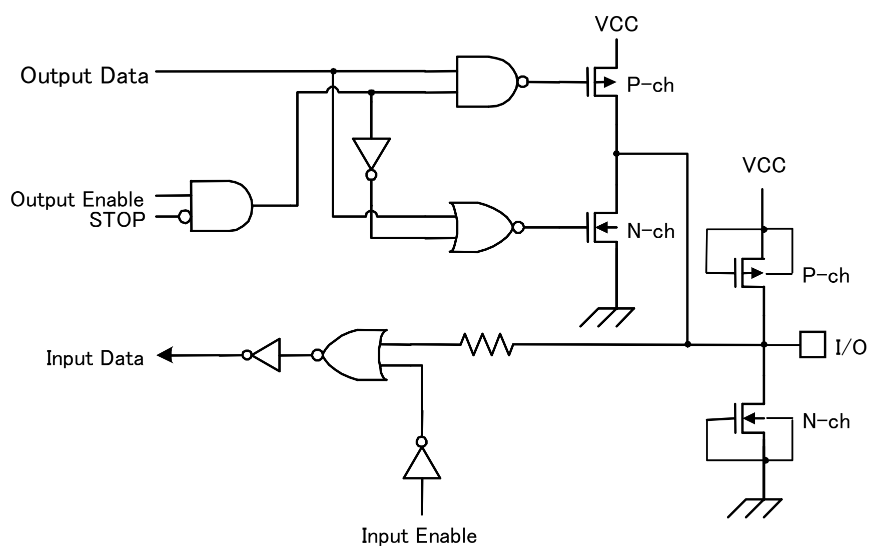
  Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

  The dedicated signal is described below.

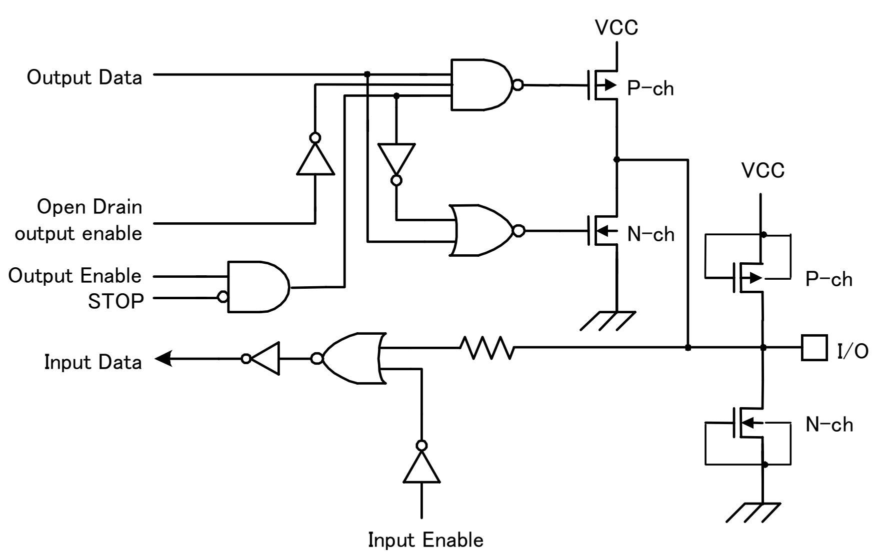
  STOP: This signal becomes active "1" when the halt mode setting register is set to the Stop mode and the CPU executes the HALT instruction. When the drive enable bit <DRVE> is set to "1", however, Stop remains at "0".

- The input protection resistance ranges from several tens of ohms to several hundreds of ohms.

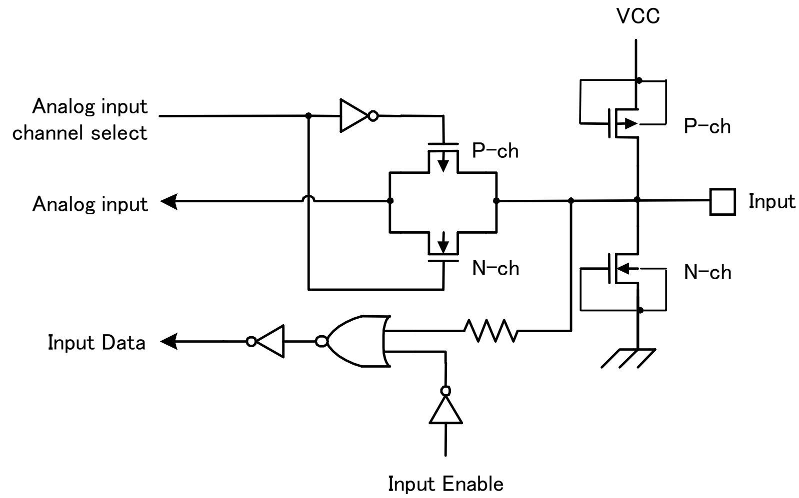
■ P0 (D0 to D7), P4 (A0 to A7), P70, P71, P73 to P75, PC0 to PC5, PD0 to PD7, PF1(RXD0), PF2 (CTS0, SCLK0), PF4 (RXD1), PF5 ($\overline{\text{CTS1}}$, SCLK1), PF6 (TX), PF7 (RX), PM0 ($\overline{\text{SS}}$), PN0 (SCK0), PN3 (SCK1), PM4 (SCK2)



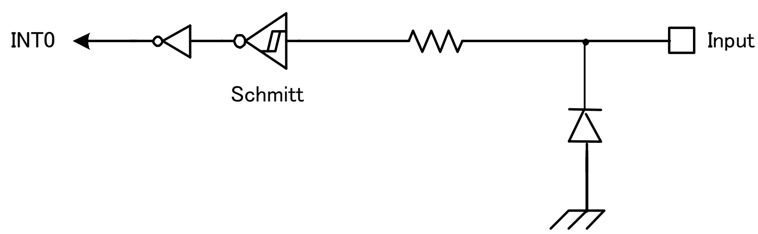
■ P72 (SI2/SCL2), PF0 (TXD0), PF3 (TXD1), PM1 (MOSI), PM2 (MISO), PM3 (SECLK), PN1 (SO0/SDA0), PN2 (SI0/SCL0), PN4 (SO1/SDA1), PN5 (SI1/SCL1), PN6 (SO2/SDA2)



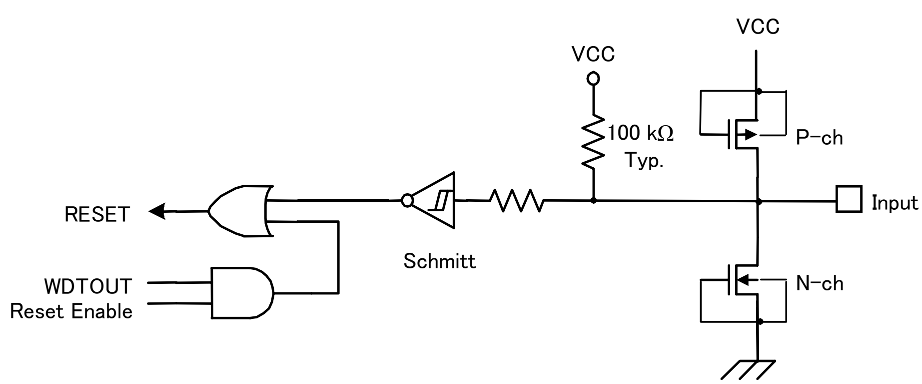
■ PG(AN0 to 7), PL0 to 3(AN8 to 11)



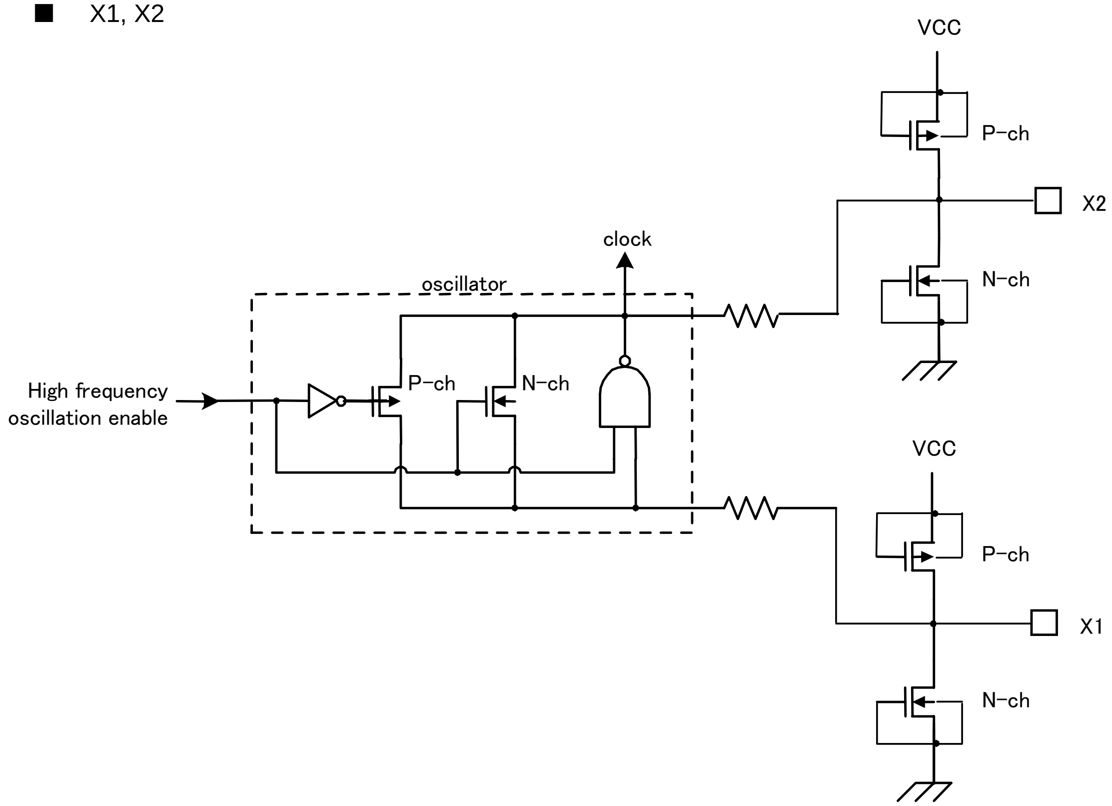
■ INT0



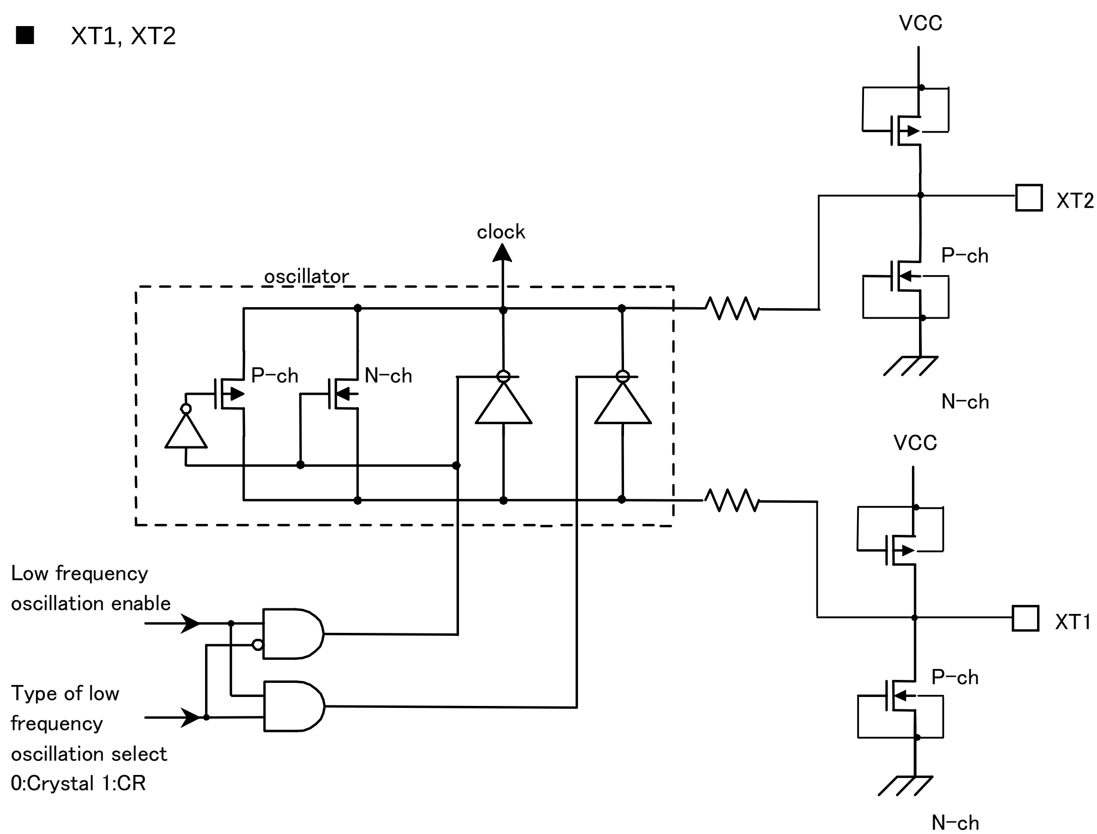
■ $\overline{\text{RESET}}$



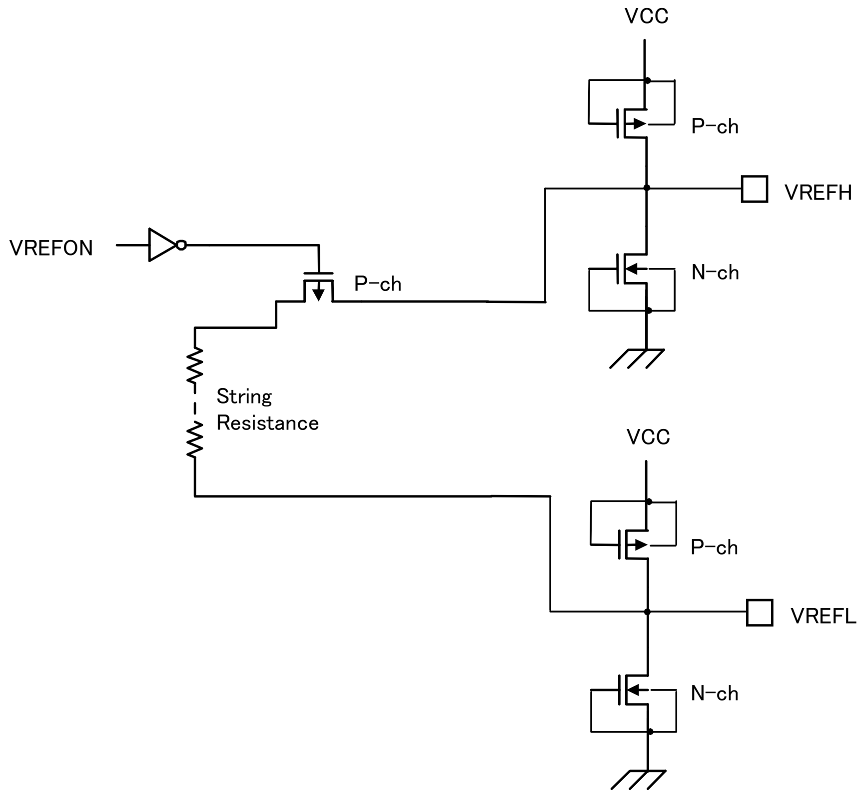
■ X1, X2


VCC
P−ch
X2
clock
oscillator
N−ch
High frequency oscillation enable
P−ch
N−ch
VCC
P−ch
X1
N−ch

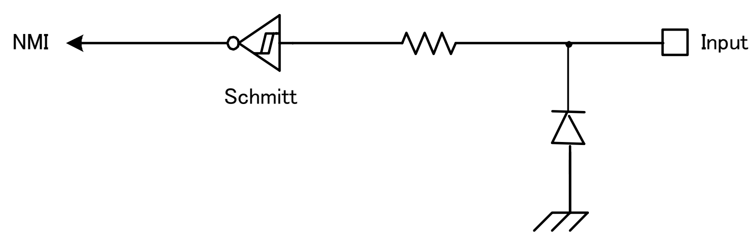
■ XT1, XT2


VCC
XT2
clock
P−ch
oscillator
P−ch
N−ch
N−ch
VCC
Low frequency oscillation enable
XT1
P−ch
Type of low frequency oscillation select 0:Crystal 1:CR
N−ch

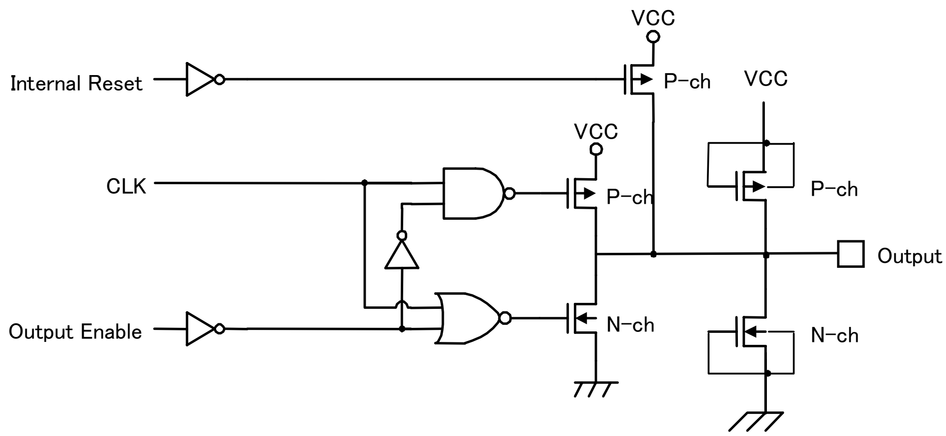
■ VREFH, VREFL



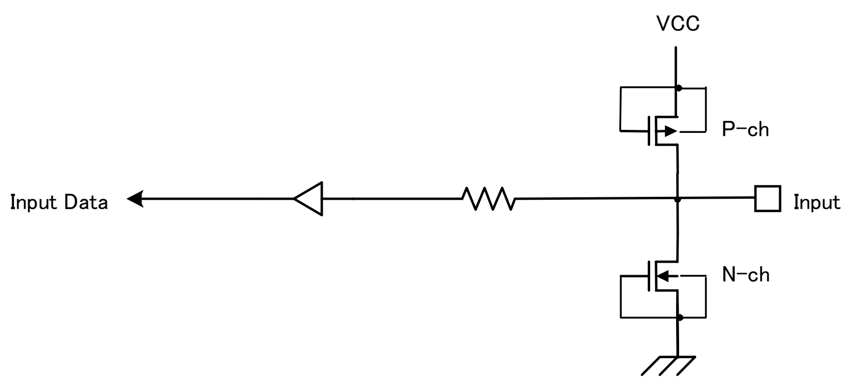
■ $\overline{\text{NMI}}$



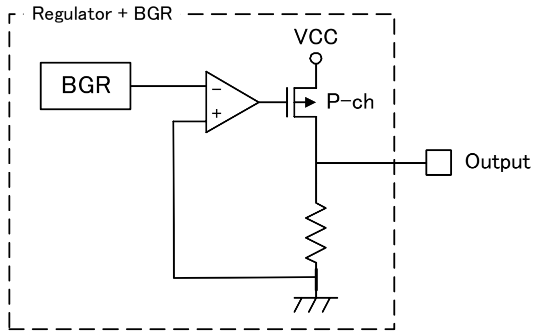
■ CLK



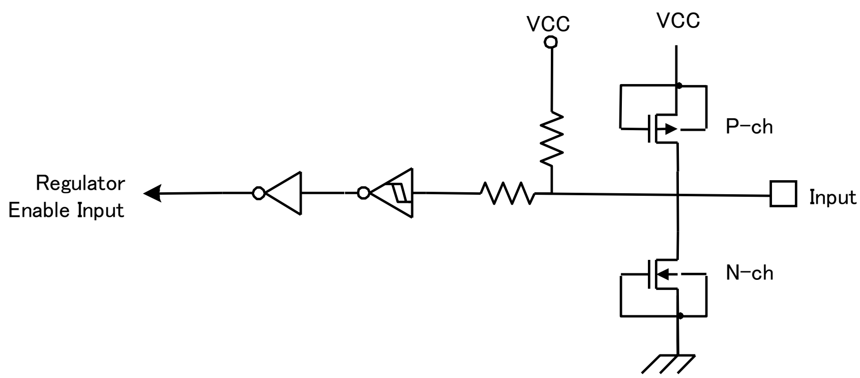
■ AM0 to 1, TEST0 to 1



■ REGOUT



■ REGEN



7. Points to Note and Restrictions

7.1 Notation

(1) The notation for built-in I/O registers is as follows register symbol <Bit symbol>

Example: TRUN01<T0RUN> denotes bit T0RUN of register TRUN01.

(2) Read-modify-write instructions (RMW)

An instruction in which the CPU reads data from memory and writes the data to the same memory location in one instruction.

Example 1: SET 3, (TRUN01); Set bit3 of TRUN01.

Example 2: INC 1, (400H); Increment the data at 400H.

- Examples of read-modify-write instructions on the TLCS-900/H1

Exchange instruction

EX (mem), R

Arithmetic operations

ADD (mem), R/#	ADC (mem), R/#
SUB (mem), R/#	SBC (mem), R/#
INC #3, (mem)	DEC #3, (mem)

Logic operations

AND (mem), R/#	OR (mem), R/#
XOR (mem), R/#	

Bit manipulation operations

STCF#3/A, (mem)	RES #3, (mem)
SET #3, (mem)	CHG #3, (mem)
TSET#3, (mem)	

Rotate and shift operations

RLC (mem)	RRC (mem)
RL (mem)	RR (mem)
SLA (mem)	SRA (mem)
SLL (mem)	SRL (mem)
RLD (mem)	RRD (mem)

.

7.2 Points to Note

(1) Watchdog timer

The watchdog timer starts operation immediately after a reset is released. When the watchdog timer is not to be used, disable it.

(2) The stable time of the internal clock

When releasing the external reset using “built-in clock doubler” until the internal reset is released, the requiring time to stabilize the circuit is automatically set. See section 3.1.2 “Reset Operation” for details. Also when releasing standby mode in STOP mode using an interrupt until the internal circuit starts the operation, the stable time of the oscillator is automatically input. See section 3.4 “Standby Function (3) STOP mode” for details.

(3) Undefined bit in the built-in I/O register

When reading the undefined bit in the built-in I/O register, the undefined value is output.

Thus, when creating program, it should not be depending on this bit condition.

(4) Reserved address areas

The 16 bytes area (FFFFF0H to FFFFFFH) cannot be used for it is reserved as internal area. If using emulator, optional 64 Kbytes of 16M bytes area are used for control emulator. Therefore, if using emulator, its area cannot be used.

(5) POP SR instruction

Execute the POP SR instruction during DI condition.

8. Package

Package Dimensions：P-LQFP100-1414-0.50F