TOSHIBA

TOSHIBA Original CMOS 32-Bit Microcontroller

TLCS-900/H1 Series

TMP92CM27

TOSHIBA CORPORATION

Semiconductor Company

Preface

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

CMOS 32-bit Micro-controller

TMP92CM27FG

1. Outline and Device Characteristics

TMP92CM27 is high-speed advanced 32-bit micro-controller developed for controlling equipment which processes mass data.

TMP92CM27 is a micro-controller which has a high-performance CPU (TLCS-900/H1 CPU) and various built-in I/Os.

TMP92CM27FG is housed in a 144-pin flat package.

Device characteristics are as follows:

(1) CPU : 32-bit CPU(TLCS-900/H1 CPU)

- Compatible with TLCS-900/L1 instruction code
- 16Mbytes of linear address space
- General-purpose register and register banks
- Micro DMA : 8channels (250ns/4bytes at fc = 40MHz, best case)

(2) Minimum instruction execution time : 50ns(at fc=40MHz)

(3) Internal memory

- Internal RAM : 32K-byte (32-bit 1 clock access and program execution are possible)
- Internal ROM : None

(4) External memory expansion

- Expandable up to 16M bytes (shared program/data area)
- Can simultaneously support 8/16-bit width external data bus
  ⋯ Dynamic data bus sizing
- Separate bus system

(5) Memory controller

- Chip select output : 6 channels

(6) 8-bit timers : 8 channels

(7) 16-bit timers : 6 channels

(8) Pattern generator : 2 channels

(9) General-purpose serial interface : 4 channels

- UART/Synchronous mode : 4 channels (ch.0 to ch.3)
- IrDA Ver.1.0(115kbps) mode selectable : 1 channels (ch.0)

(10) Serial bus interface : 2 channels

- I^2C bus mode/clock synchronous mode selectable

(11) High Speed serial interface : 2 channels

(12) SDRAM controller : 1 channels

- Supported 16M, 64M-bit SDR (Single Data Rate)-SDRAM
- Supported not only operate as RAM and Data for LCD display but also programming directly from SDRAM

(13) 10-bit AD converter : 12 channels

(14) 8-bit DA converter : 2 channels

(15) Watchdog timer

(16) Key-on wake up (only for HALT release) : 8 channels

(17) Interrupts : 71 interrupts

- 9 CPU interrupts : Software interrupt instruction and illegal instruction
- 49 internal interrupts : Seven selectable priority levels
- 13 external interrupts(INT0 to INTB, $\overline{NMI}$) : Seven selectable priority levels (INT0 to INTB)

  (INT0 to INTB are selectable edge or level interrupt)

(18) External bus release function

(19) Input/output ports : 83 pins

(20) Stand-by function

- Three Halt modes : Idle2 (programmable), Idle1, Stop

(21) Clock controller

- Clock doubler (PLL) : fc = f_{OSCH}×4 (fc=40MHz @ f_{OSCH}=10MHz)
- Clock gear function : Select a High-frequency clock fc to fc/16

(22) Operating voltage

- VCC = 3.0 V to 3.6 V (fc max = 40MHz)

(23) Package

- 144 pin QFP : P-LQFP144-1616-0.40C

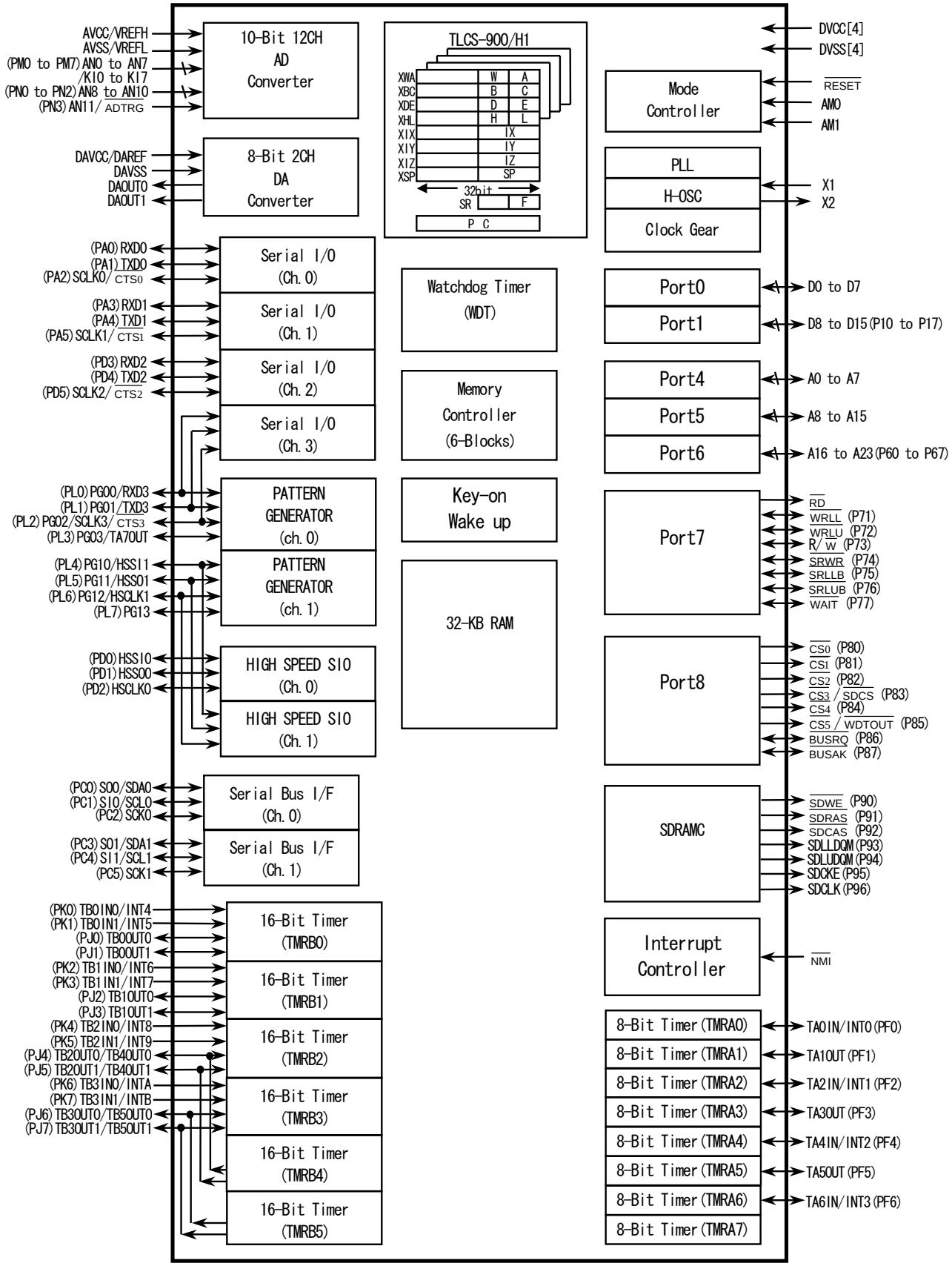


Figure 1.1 TMP92CM27 block diagram

2. Pin assignment and pin functions

The assignment of input/output pins for the TMP92CM27, their names and functions are as follows:

2.1 Pin assignment Diagram

Figure 2.1.1 shows the pin assignment of the TMP92CM27FG.

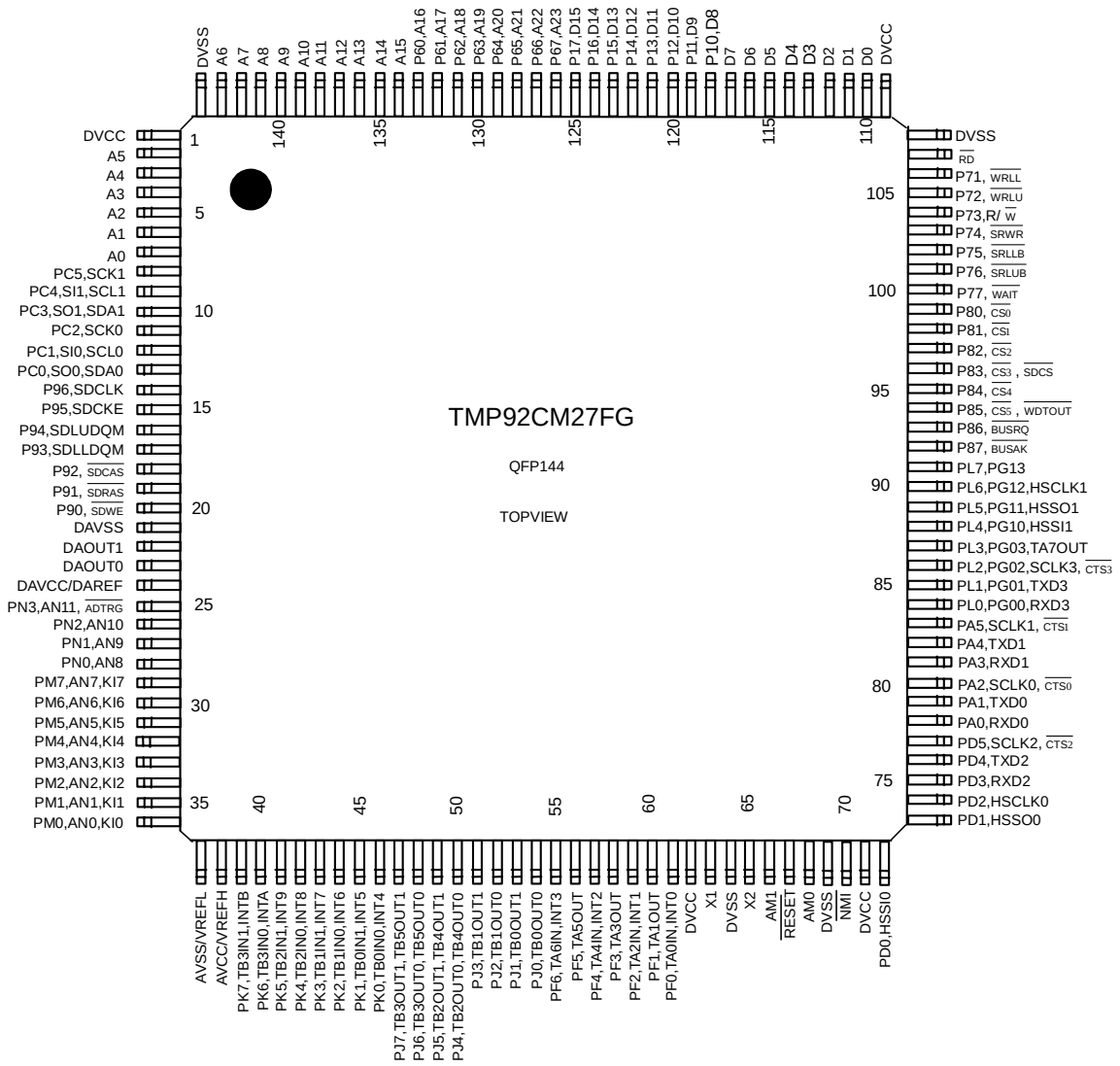


Figure 2.1.1 Pin assignment diagram (144 pin LQFP)

2.2 Pin names and functions

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

Table 2.2.1 Pin names and functions (1/5)

Pin name	Number of Pin	I/O	Function
D0 to D7	8	I/O	Data: Data bus D0 to D7
P10 to P17 D8 to D15	8	I/O I/O	Port 1: I/O port Input or output specifiable in units of bits Data: Data bus D8 to D15
A0 to A7	8	Output	Address: Address bus A0 to A7
A8 to A15	8	Output	Address: Address bus A8 to A15
P60 to P67 A16 to A23	8	I/O Output	Port 6: I/O port Address: Address bus A16 to A23
$\overline{RD}$	1	Output	Read: Outputs strobe signal for read external memory (with pull-up register)
P71 $\overline{WRLL}$	1	I/O Output	Port 71: I/O port (Schmitt input, with pull-up register) Write: Output strobe signal for writing data on pins D0 to D7
P72 $\overline{WRLU}$	1	I/O Output	Port 72: I/O port (schmitt input, with pull-up register) Write: Output strobe signal for writing data on pins D8 to D15
P73 R/$\overline{W}$	1	I/O Output	Port 73: I/O port (schmitt input) Read/Write: 1 represents read or dummy cycle; 0 represents write cycle
P74 $\overline{SRWR}$	1	I/O Output	Port 74: I/O port (Schmitt input, with pull-up register) Write enable for SRAM: Strobe signal for writing data
P75 $\overline{SRLLB}$	1	I/O Output	Port 75: I/O port (Schmitt input, with pull-up register) Data enable for SRAM on pins D0 to D7
P76 $\overline{SRLUB}$	1	I/O Output	Port 76: I/O port (Schmitt input, with pull-up register) Data enable for SRAM on pins D8 to D15
P77 $\overline{WAIT}$	1	I/O Input	Port 77: I/O port (Schmitt input) Wait: Signal used to request CPU bus wait
P80 $\overline{CS0}$	1	Output Output	Port 80: Output port Chip select 0: Outputs “Low” when address is within specified address area
P81 $\overline{CS1}$	1	Output Output	Port 81: Output port Chip select 1: Outputs “Low” when address is within specified address area
P82 $\overline{CS2}$	1	Output Output	Port 82: Output port Chip select 2: Outputs “Low” when address is within specified address area
P83 $\overline{CS3}$ $\overline{SDCS}$	1	Output Output Output	Port 83: Output port Chip select 3: Outputs “Low” when address is within specified address area Chip select for SDRAM: Outputs “Low” when address is within SDRAM address area
P84 $\overline{CS4}$	1	Output Output	Port 84: Output port Chip select 4: Outputs “Low” when address is within specified address area
P85 $\overline{CS5}$ $\overline{WDTOUT}$	1	Output Output Output	Port 85: Output port Chip select 5: Outputs “Low” when address is within specified address area Watchdog timer output pin
P86 $\overline{BUSRQ}$	1	I/O Input	Port 86: I/O port (Schmitt input) Bus request: request pin that set external memory bus to high-impedance (for External DMAC)
P87 $\overline{BUSAK}$	1	I/O Output	Port 87: I/O port (Schmitt input) Bus acknowledge: this pin show that external memory bus pin is set to high-impedance by receiving $\overline{BUSRQ}$ (for External DMAC)

Table 2.2.2 Pin names and functions (2/5)

Pin name	Number of Pin	I/O	Function
P90 $\overline{SDWE}$	1	Output Output	Port 90: Output port Write enable for SDRAM
P91 $\overline{SDRAS}$	1	Output Output	Port 91: Output port Row address strobe for SDRAM
P92 $\overline{SDCAS}$	1	Output Output	Port 92: Output port Column address strobe for SDRAM
P93 SDLLDQM	1	Output Output	Port 93: Output port Data enable for SDRAM on pins D0 to D7
P94 SDLUDQM	1	Output Output	Port 94: Output port Data enable for SDRAM on pins D8 to D15
P95 SDCKE	1	Output Output	Port 95: Output port Clock enable for SDRAM
P96 SDCLK	1	Output Output	Port 96: Output port Clock for SDRAM
PA0 RXD0	1	I/O Input	Port A0: I/O port (Schmitt input) Serial 0 receive data
PA1 TXD0	1	I/O Output	Port A1: I/O port (Schmitt input) Serial 0 send data: Open-drain output programmable
PA2 SCLK0 $\overline{CTS0}$	1	I/O I/O Input	Port A2: I/O port (Schmitt input) Serial 0 clock I/O Serial 0 data send enable (Clear To Send)
PA3 RXD1	1	I/O Input	Port A3: I/O port (Schmitt input) Serial 1 receive data
PA4 TXD1	1	I/O Output	Port A4: I/O port (Schmitt input) Serial 1 send data: Open-drain output programmable
PA5 SCLK1 $\overline{CTS1}$	1	I/O I/O Input	Port A5: I/O port (Schmitt input) Serial 1 clock I/O Serial 1 data send enable (Clear To Send)
PC0 SO0 SDA0	1	I/O Output I/O	Port C0: I/O port (Schmitt input) Serial bus interface 0 send data at SIO mode Serial bus interface 0 send/receive data at I^2C mode Open-drain output programmable
PC1 SI0 SCL0	1	I/O Input I/O	Port C1: I/O port (Schmitt input) Serial bus interface 0 receive data at SIO mode Serial bus interface 0 clock I/O data at I^2C mode Open-drain output programmable
PC2 SCK0	1	I/O I/O	Port C2: I/O port (Schmitt input) Serial bus interface 0 clock I/O data at SIO mode
PC3 SO1 SDA1	1	I/O Output I/O	Port C3: I/O port (Schmitt input) Serial bus interface 1 send data at SIO mode Serial bus interface 1 send/receive data at I^2C mode Open-drain output programmable
PC4 SI1 SCL1	1	I/O Input I/O	Port C4: I/O port (Schmitt input) Serial bus interface 1 receive data at SIO mode Serial bus interface 1 clock I/O data at I^2C mode Open-drain output programmable
PC5 SCK1	1	I/O I/O	Port C5: I/O port (Schmitt input) Serial bus interface 1 clock I/O data at SIO mode

Table 2.2.3 Pin names and functions (3/5)

Pin name	Number of Pin	I/O	Function
PD0 HSSI0	1	I/O Input	Port D0: I/O port High speed Serial 0 receive data
PD1 HSSO0	1	I/O Output	Port D1: I/O port (Schmitt input) High speed Serial 0 send data
PD2 HSCLK0	1	I/O Output	Port D2: I/O port (Schmitt input) High speed Serial 0 clock I/O
PD3 RXD2	1	I/O Input	Port D3: I/O port (Schmitt input) Serial 2 receive data
PD4 TXD2	1	I/O Output	Port D4: I/O port (Schmitt input) Serial 2 send data: Open-drain output programmable
PD5 SCLK2 $\overline{\text{CTS2}}$	1	I/O I/O Input	Port D5: I/O port (Schmitt input) Serial 2 clock I/O Serial 2 data send enable (Clear To Send)
PF0 TA0IN INT0	1	I/O Input Input	Port F0: I/O port (Schmitt input) 8-bit timer 0 input: Input pin of 8-bit timer TMRA0 Interrupt request pin 0: Interrupt request pin with programmable level/rising/falling edge
PF1 TA1OUT	1	I/O Output	Port F1: I/O port (Schmitt input) 8-bit timer 1 output: Output pin of 8-bit timer TMRA0 or TMRA1
PF2 TA2IN INT1	1	I/O Input Input	Port F2: I/O port (Schmitt input) 8-bit timer 2 input: Input pin of 8-bit timer TMRA2 Interrupt request pin 1: Interrupt request pin with programmable level/rising/falling edge
PF3 TA3OUT	1	I/O Output	Port F3: I/O port (Schmitt input) 8-bit timer 3 output: Output pin of 8-bit timer TMRA2 or TMRA3
PF4 TA4IN INT2	1	I/O Input Input	Port F4: I/O port (Schmitt input) 8-bit timer 4 input: Input pin of 8-bit timer TMRA4 Interrupt request pin 2: Interrupt request pin with programmable level/rising/falling edge
PF5 TA5OUT	1	I/O Output	Port F5: I/O port (Schmitt input) 8-bit timer 5 output: Output pin of 8-bit timer TMRA4 or TMRA5
PF6 TA6IN INT3	1	I/O Input Input	Port F6: I/O port (Schmitt input) 8-bit timer 6 input: Input pin of 8-bit timer TMRA6 Interrupt request pin 3: Interrupt request pin with programmable level/rising/falling edge
PJ0 TB0OUT0	1	I/O Output	Port J0: I/O port (Schmitt input) 16-bit timer 0 output 0: Output pin of 16-bit timer TMRB0
PJ1 TB0OUT1	1	I/O Output	Port J1: I/O port (Schmitt input) 16-bit timer 0 output 1: Output pin of 16-bit timer TMRB0
PJ2 TB1OUT0	1	I/O Output	Port J2: I/O port (Schmitt input) 16-bit timer 1 output 0: Output pin of 16-bit timer TMRB1
PJ3 TB1OUT1	1	I/O Output	Port J3: I/O port (Schmitt input) 16-bit timer 1 output 1: Output pin of 16-bit timer TMRB1
PJ4 TB2OUT0 TB4OUT0	1	I/O Output Output	Port J4: I/O port (Schmitt input) 16-bit timer 2 output 0: Output pin of 16-bit timer TMRB2 16-bit timer 4 output 0: Output pin of 16-bit timer TMRB4
PJ5 TB2OUT1 TB4OUT1	1	I/O Output Output	Port J5: I/O port (Schmitt input) 16-bit timer 2 output 1: Output pin of 16-bit timer TMRB2 16-bit timer 4 output 1: Output pin of 16-bit timer TMRB4
PJ6 TB3OUT0 TB5OUT0	1	I/O Output Output	Port J6: I/O port (Schmitt input) 16-bit timer 3 output 0: Output pin of 16-bit timer TMRB3 16-bit timer 5 output 0: Output pin of 16-bit timer TMRB5
PJ7 TB3OUT1 TB5OUT1	1	I/O Output Output	Port J7: I/O port (Schmitt input) 16-bit timer 3 output 1: Output pin of 16-bit timer TMRB3 16-bit timer 5 output 1: Output pin of 16-bit timer TMRB5

Table 2.2.4 Pin names and functions (4/5)

Pin name	Number of Pin	I/O	Function
PK0	1	Input	Port K0: Input port (Schmitt input)
TB0IN0		Input	16-bit timer 0 input 0: Input of count/capture trigger in 16-bit TMRB0
INT4		Input	Interrupt request pin 4 : Interrupt request pin with programmable level/rising/falling edge
PK1	1	Input	Port K1: Input port (Schmitt input)
TB0IN1		Input	16-bit timer 0 input 1: Input of count/capture trigger in 16-bit TMRB0
INT5		Input	Interrupt request pin 5 : Interrupt request pin with programmable level/rising/falling edge
PK2	1	Input	Port K2: Input port (Schmitt input)
TB1IN0		Input	16-bit timer 1 input 0: Input of count/capture trigger in 16-bit TMRB1
INT6		Input	Interrupt request pin 6 : Interrupt request pin with programmable level/rising/falling edge
PK3	1	Input	Port K3: Input port (Schmitt input)
TB1IN1		Input	16-bit timer 1 input 1: Input of count/capture trigger in 16-bit TMRB1
INT7		Input	Interrupt request pin 7 : Interrupt request pin with programmable level/rising/falling edge
PK4	1	Input	Port K4: Input port (Schmitt input)
TB2IN0		Input	16-bit timer 2 input 0: Input of count/capture trigger in 16-bit TMRB2
INT8		Input	Interrupt request pin 8 : Interrupt request pin with programmable level/rising/falling edge
PK5	1	Input	Port K5: Input port (Schmitt input)
TB2IN1		Input	16-bit timer 2 input 1: Input of count/capture trigger in 16-bit TMRB2
INT9		Input	Interrupt request pin 9 : Interrupt request pin with programmable level/rising/falling edge
PK6	1	Input	Port K6: Input port (Schmitt input)
TB3IN0		Input	16-bit timer 3 input 0: Input of count/capture trigger in 16-bit TMRB3
INTA		Input	Interrupt request pin A : Interrupt request pin with programmable level/rising/falling edge
PK7	1	Input	Port K7: Input port (Schmitt input)
TB3IN1		Input	16-bit timer 3 input 1: Input of count/capture trigger in 16-bit TMRB3
INTB		Input	Interrupt request pin B : Interrupt request pin with programmable level/rising/falling edge
PL0	1	I/O	Port L0: I/O port (Schmitt input)
PG00		Output	Pattern generator output 00
RXD3		Input	Serial 3 receive data
PL1	1	I/O	Port L1: I/O port (Schmitt input)
PG01		Output	Pattern generator output 01
TXD3		Output	Serial 3 send data: Open-drain output programmable
PL2	1	I/O	Port L2: I/O port (Schmitt input)
PG02		Output	Pattern generator output 02
SCLK3		I/O	Serial 3 clock I/O
$\overline{\text{CTS3}}$		Input	Serial 3 data send enable (Clear To Send)
PL3	1	I/O	Port L3: I/O port (Schmitt input)
PG03		Output	Pattern generator output 03
TA7OUT		Output	8-bit timer 7 output: Output pin of 8-bit timer TMRA6 or TMRA7
PL4	1	I/O	Port L4: I/O port
PG10		Output	Pattern generator output 10
HSSI1		Input	High speed Serial 1 receive data
PL5	1	I/O	Port L5: I/O port (Schmitt input)
PG11		Output	Pattern generator output 11
HSSO1		Output	High speed Serial 1 send data
PL6	1	I/O	Port L6: I/O port (Schmitt input)
PG12		Output	Pattern generator output 12
HSCLK1		Output	High speed Serial 1 clock I/O
PL7	1	I/O	Port L7: I/O port (Schmitt input)
PG13		Output	Pattern generator output 13

Table 2.2.5 Pin names and functions (5/5)

Pin name	Number of Pin	I/O	Function
PM0 to PM7 AN0 to AN7 KI0 to KI7	8	Input	Port M: Input port (Schmitt input) Analog input 0 to 7: Pin used to input to AD converter Key input 0 to 7: Pin used of Key-on wakeup 0 to 7
PN0 to PN3 AN8 to AN11 $\overline{\text{ADTRG}}$	4	Input	Port N: Input port (Schmitt input) Analog input 8 to 11: Pin used to input to AD converter AD trigger: Signal used for request AD start (Shared with PN3)
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin: Interrupt request pin with programmable falling edge level or with both edge levels programmable (Schmitt input)
DAOUT0	1	Output	Digital output 0: Pin used to output to DA converter 0
DAOUT1	1	Output	Digital output 1: Pin used to output to DA converter 1
AM0, AM1	2	Input	Operation mode: Fixed to AM1="0",AM0="1" External 16-bit bus start Fixed to AM1="1",AM0="0" External 8-bit bus start Fixed to AM1="1",AM0="1" Reserved Fixed to AM1="0",AM0="0" Reserved
X1 / X2	2	I/O	High-frequency oscillator connection I/O pins
$\overline{\text{RESET}}$	1	Input	Reset: Initializes TMP92CM27 (Schmitt input, with pull-up register)
AVCC / VREFH	1	Input	Pin used to both power supply pin for AD converter and standard power supply for AD converter (H)
AVSS / VREFL	1	Input	Pin used to both GND pin for AD converter (0V) and standard power supply pin for AD converter (L)
DAVCC / DAREF	1	Input	Pin used to both power supply pin for DA converter and standard power supply for DA converter
DAVSS	1	Input	Pin used to both GND pin for DA converter (0V)
DVCC	4	-	Power supply pin (All DVCC pins should be connected with the power supply pin)
DVSS	4	-	GND pins (0V) (All DVSS pins should be connected with GND (0V))

3. Operation

This section describes the basic components, functions and operation of the TMP92CM27.

3.1 CPU

The TMP92CM27 contains an advanced high-speed 32-bit CPU (TLCS-900/H1 CPU)

3.1.1 CPU Outline

TLCS-900/H1 CPU is high-speed and high-perforrmance CPU based on TLCS-900/L1 CPU.TLCS-900/H1 CPU has expanded 32-bit internal data bus to process instructions more quickly.

Outline is as follows:

Table 3.1.1 TMP92CM27 Outline

Parameter	TMP92CM27	
Width of CPU address bus	24 bits	
Width of CPU data bus	32 bits	
Internal operating frequency	Max 20MHz	
Minimum bus cycle	1-clock access (50ns at f_{SYS} = 20MHz)	
Internal RAM	32-bit 1-clock access	
Internal I/O	8-bit, 2-clock access	CGEAR, INTC, PORT, MEMC, TMRA, TMRB, PG, SIO, SBI, SDRAMC, ADC, DAC, WDT
	16-bit, 2-clock access	HSIO
External memory (SRAM etc)	8 ro 16-bit 2-clock access (can insert some waits)	
External memory (SDRAM)	16-bit 1-clock access	
Minimum instruction Execution cycle	1-clock(50ns at f_{SYS} = 20MHz)	
Conditional jump	2-clock(100ns at f_{SYS} = 20MHz)	
Instruction queue buffer	12 bytes	
Instruction set	Compatible with TLCS-900/L1 (LDX instruction is deleted)	
CPU mode	Only maximum mode	
Micro DMA	8 channel	

3.1.2 Reset Operation

When resetting the TMP92CM27, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the $\overline{\text{RESET}}$ input low for at least 20 system clocks (16 µs at fc = 40 MHz).

At reset, since the clock doubler (PLL) is bypassed and clock-gear is set to 1/16, system clock operates at 1.25 MHz (fc = 40 MHz).

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<7:0> ← data in location FFFF00H

  PC<15:8> ← data in location FFFF01H

  PC<23:16> ← data in location FFFF02H
- Sets the stack pointer (XSP) to 00000000H.
- Sets bits <IFF2:0> of the status register (SR) to “111"
  (thereby setting the interrupt level mask register to level 7).
- Clears bits <RFP1:0> of the status register to “00”
  (there by 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.

Internal reset is released as soon as external 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 the TMP92CM27 may be spoiled because the control signals are unstable until power supply becomes stable after power-on reset.

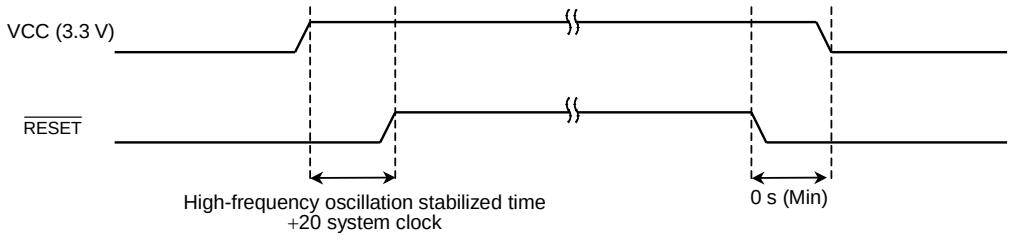


Figure 3.1.1 Power on Reset Timing Example

3.1.3 Setting of AM0 and AM1

Set AM1 and AM0 pins like Table 3.1.2 according to system usage.

Table 3.1.2 Operation Mode Setup Table

Operation mode	Mode Setup Input Pin		
	$\overline{\text{RESET}}$	AM1	AM0
16-bit external bus start (Multi 16 Mode)		0	1
8-bit external bus start (Multi 8 Mode)		1	0
Reserved		1	1
Reserved		0	0

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMPP92CM27.

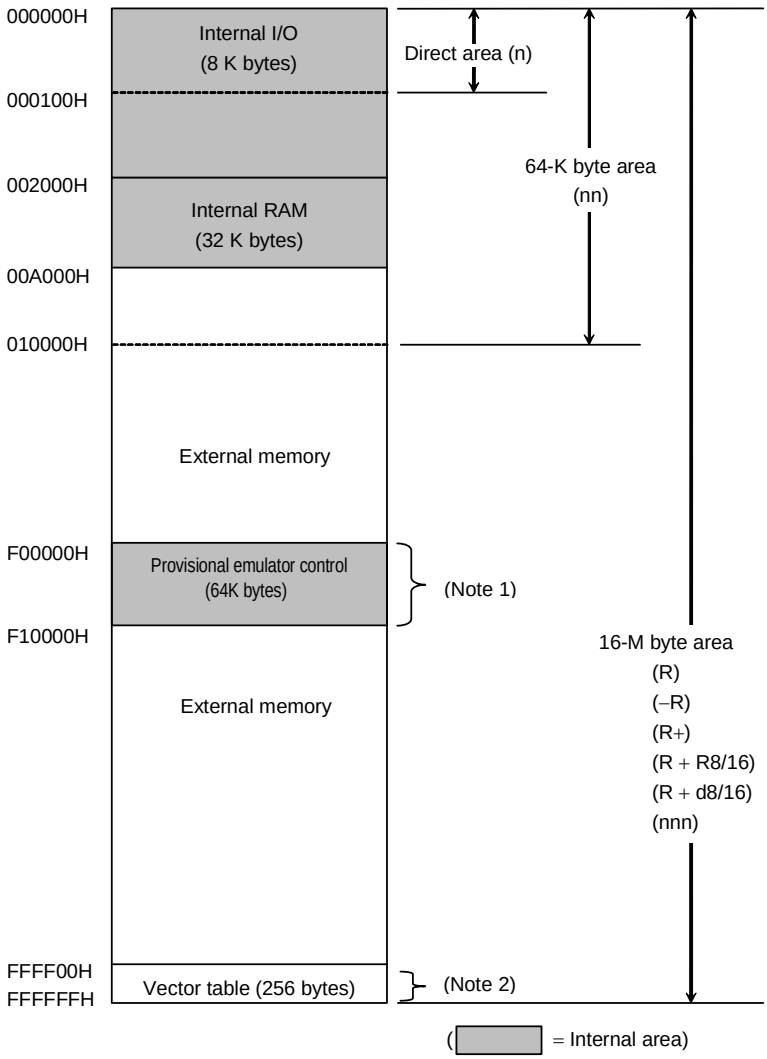


Figure 3.2.1 Memory Map

Note 1: Provisional emulator control area is for an emulator, it is mapped F00000H to F0FFFFH after reset. On emulator $\overline{WR}$ signal and $\overline{RD}$ signal are asserted, when this area is accessed. Be carefull to use external memory.

Note 2: Don't use the last 16-bytes area (FFFFF0H to FFFFFFH). This area is reserved for an emulator.

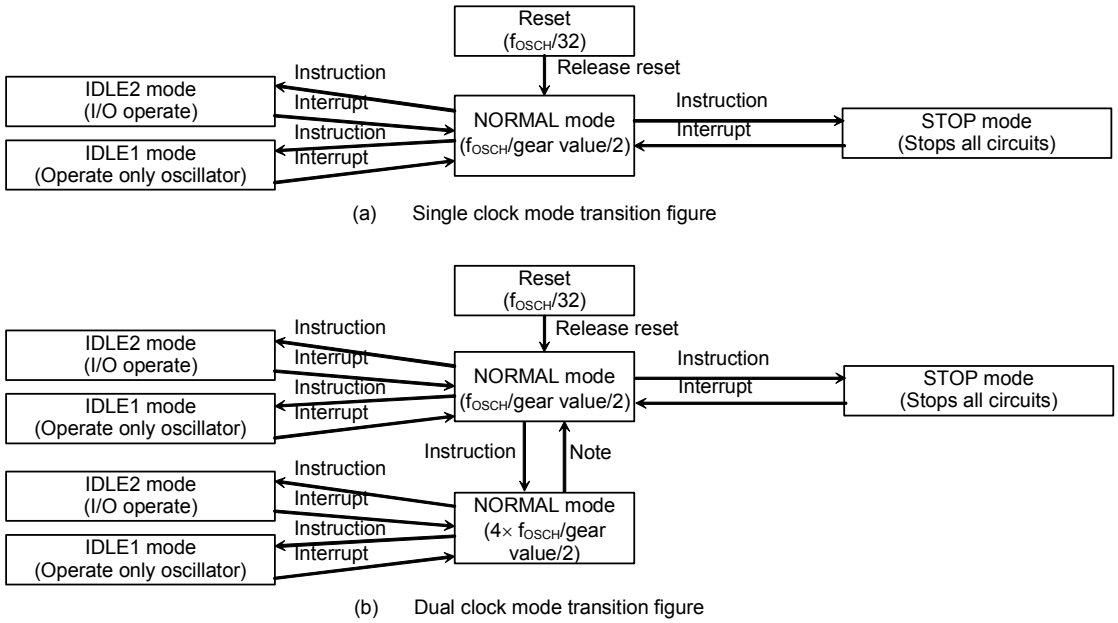
3.3 Clock Function and Stand-by Function

TMP92CM27 contains (1) clock gear, (2) clock doubler (PLL), (3) stand-by controller and (4) noise reducing circuit. They are used for low power, low noise systems.

This chapter is organized as follows:

The clock operating modes are as follows: (a) single clock mode (X1, X2 pins only), (b) dual clock mode (X1, X2 pins and PLL).

Figure 3.3.1 shows a transition figure.



Note 1: If you shift from NORMAL mode with use of PLL to NORMAL mode, execute following setting in the same order.

1) Change CPU clock (PLLCR0 <FCSEL> ← "0")

2) Stop PLL circuit (PLLCR1 <PLLON> ← "0")

Note 2: It's prohibited to shift from NORMAL mode with use of PLL to STOP mode directly.

You should set NORMAL mode once, and then shift to STOP mode. (You should stop high-frequency oscillator after you stop PLL.)

Figure 3.3.1 System Clock Block Diagram

The clock frequency input from the X1 and X2 pins is called f_{OSCH} and clock frequency selected by SYSCR1<SYSCK> is called the system clock f_{FPH}. The system clock f_{SYS} is defined as the divided clock of f_{FPH}, and one cycle of f_{SYS} is defined to as one state.

3.3.1 Block Diagram of System Clock

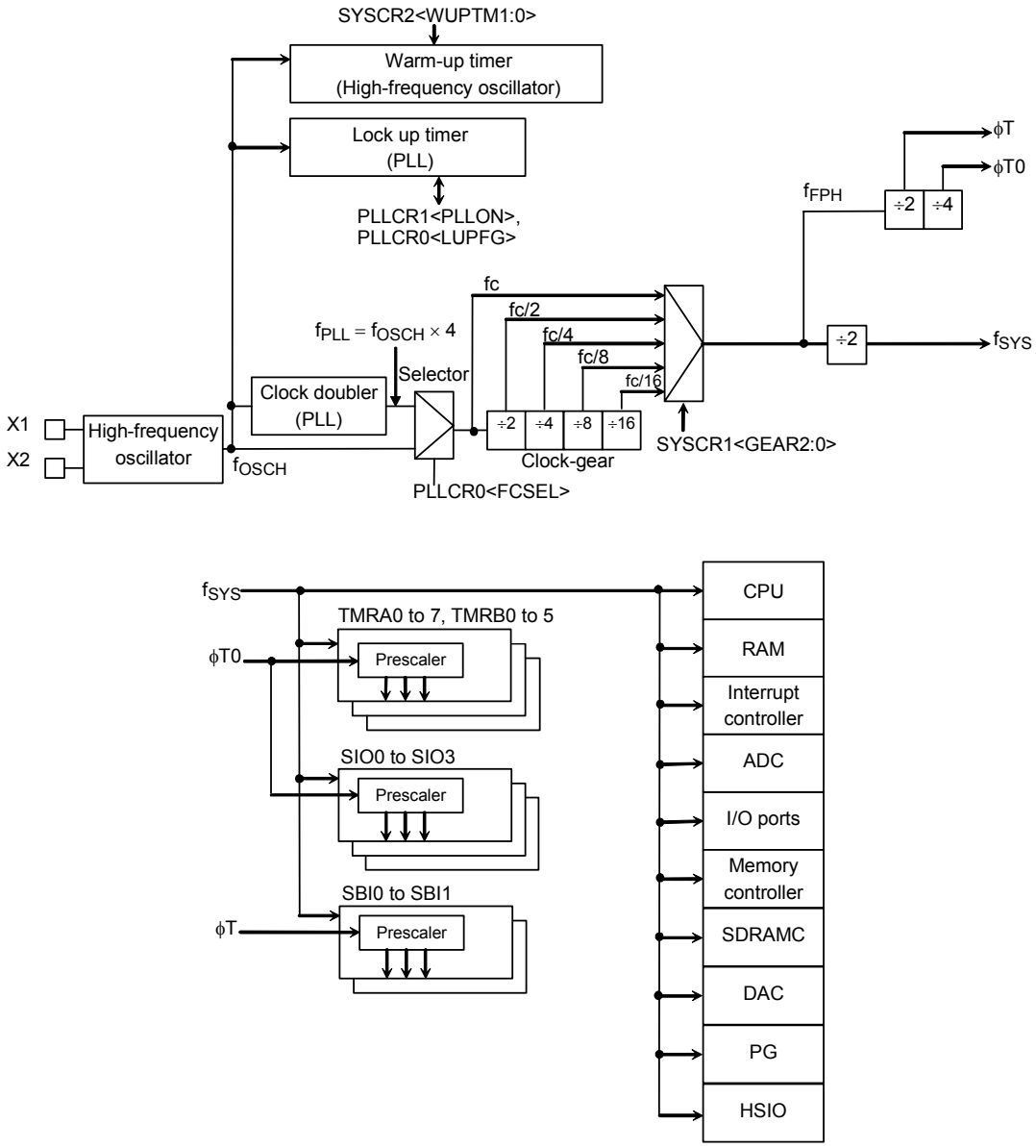


Figure 3.3.2 Block Diagram of System Clock

3.3.2 SFR

SYSCR0 (10E0H)

	7	6	5	4	3	2	1	0
Bit symbol						–		
Read/Write						R/W		
After reset						0		
Function						Always write “0”		

SYSCR1 (10E1H)

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

SYSCR2 (10E2H)

	7	6	5	4	3	2	1	0
Bit symbol	–		WUPTM1	WUPTM0	HALTM1	HALTM0		DRVE
Read/Write	R/W		R/W	R/W	R/W	R/W		R/W
After reset	0		1	0	1	1		0
Function	Always write “0”		Warm-up timer 00: Reserved 01: 2^8/input frequency 10: 2^{14}/input frequency 11: 2^{16}/input frequency		HALT mode 00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode			1: The inside of STOP mode also drives a pin

Note 1: SYSCR0<bit7> can read “1”.

Note 2: SYSCR0<bit6:3>, SYSCR0<bit1:0>, SYSCR1<bit7:3>, and SYSCR2<bit6,1> can read "0".

Figure 3.3.3 SFR for System Clock

		7	6	5	4	3	2	1	0
EMCCR0 (10E3H)	Bit symbol	PROTECT					EXTIN	DRVOSCH	
	Read/Write	R					R/W	R/W	
	After reset	0					0	1	
	Function	Protect flag 0: OFF 1: ON					1: External clock	fc oscillator driver ability 1: Normal 0: Weak	
EMCCR1 (10E4H)	Bit symbol	Switching the protect ON/OFF by write to following 1st-KEY, 2nd-KEY 1st-KEY: EMCCR1 = 5AH, EMCCR2 = A5H in succession write 2nd-KEY: EMCCR1 = A5H, EMCCR2 = 5AH in succession write							
	Read/Write								
	After reset								
	Function								
EMCCR2 (10E5H)	Bit symbol								
	Read/Write								
	After reset								
	Function								

Note 1: EMCCR0<bit0> can read “1”.

Note 2: EMCCR0<bit6:3> can read “0”.

Note 3: In case restarting the oscillator in the stop oscillation state (e.g. Restart the oscillator in STOP mode), set (EMCCR0)<DRVOSCH> = ”1”.

Figure 3.3.4 SFR for System Clock

PLLCR0 (10E8H)

	7	6	5	4	3	2	1	0
Bit symbol		FCSEL	LUPFG					
Read/Write		R/W	R					
After reset		0	0					
Function		Select fc clock 0: f_{OSCH} 1: f_{PLL}	Lock up timer status flag 0: Not end 1: End					

Note 1: Be carefull that logic of PLLCR0<LUPFG> is different from 900/L's DFM.

Note 2: PLLCR0<bit7>,<bit4:0> can read "0".

PLLCR1 (10E9H)

	7	6	5	4	3	2	1	0
Bit symbol	PLLON							
Read/Write	R/W							
After reset	0							
Function	Control on/off 0: OFF 1: ON							

Note 1: PLLCR1<bit6:0> can read "0".

Figure 3.3.5 SFR for PLL

3.3.3 System Clock Controller

The system clock controller generates the system clock signal (f_{SYS}) for the CPU core and internal I/O. It contains a oscillation circuits, PLL and a clock gear circuit for high-frequency (fc) operation. The register SYSCR1<GEAR2:0> sets the high-frequency clock gear to either 1, 2, 4, 8 or 16 (fc, fc/2, fc/4, fc/8 or fc/16). These functions can reduce the power consumption of the equipment in which the device is installed.

The initialization <GEAR2:0> = "100" will cause the system clock (f_{SYS}) to be set to fc/32 (fc/16 × 1/2) after reset.

For example, f_{SYS} is set to 1.25 MHz when the 40 MHz oscillator is connected to the X1 and X2 pins.

(1) Clock gear controller

The f_{FPH} is set according to the contents of the clock gear select register SYSCR1<GEAR2:0> to either fc, fc/2, fc/4, fc/8 or fc/16. Using the clock gear to select a lower value of f_{FPH} reduces power consumption.

Example: Changing to a high-frequency gear

```
SYSCR1    EQU    10E1H

          LD     (SYSCR1), XXXXX001B   ;   Changes fSYS to fc/2.
          LD     (DUMMY), 00H              Dummy instruction
```

X: Don't care

(High-speed clock gear changing)

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

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

Example:

```
SYSCR1    EQU    10E1H
          LD     (SYSCR1), XXXXX010B   ;   Changes fSYS to fc/4.
          LD     (DUMMY), 00H          ;   Dummy instruction
          Instruction to be executed after clock gear has changed
```

3.3.4 Clock Doubler (PLL)

PLL outputs the f_{PLL} clock signal, which is four times as fast as f_{OSCH}. It can use the low-speed-frequency oscillator, even though the internal clock is high-frequency.

A reset initializes PLL to stop status, setting to PLLCR0, PLLCR1 register is needed before use.

Like an oscillator, this circuit requires time to stabilize. This is called the lock up time and it is measured by 16-stage binary counter. Lock up time is about 1.6 ms at f_{OSCH} = 10 MHz.

Note 1: Input frequency limitation for PLL
The limitation of input frequency (High-frequency oscillation) for PLL is following.
f_{OSCH} = 6 to 10 MHz (V_{CC} = 3.0 to 3.6 V)

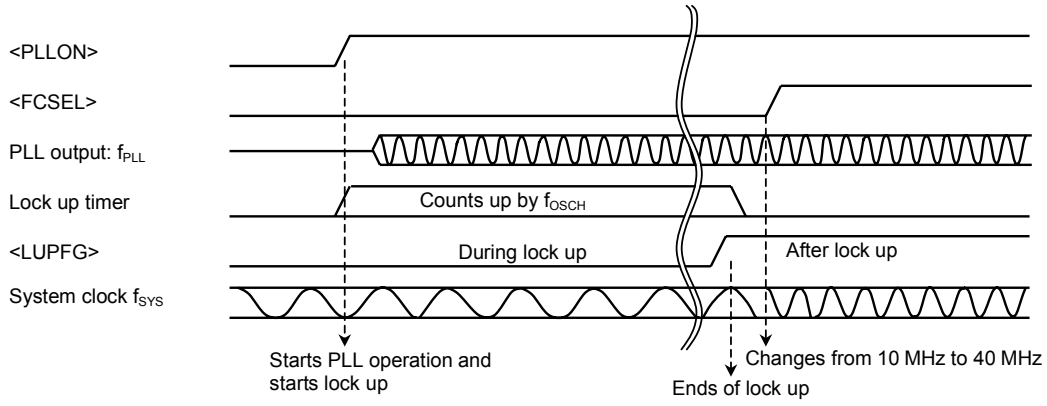
Note 2: PLLCR0 <LUPFG>
The logic of PLLCR0 <LUPFG> is different from 900/L1's DFM.
Be careful to judge an end of lock up time

.

The following is an setting example for PLL starting and PLL stopping.

Example 1: PLL starting

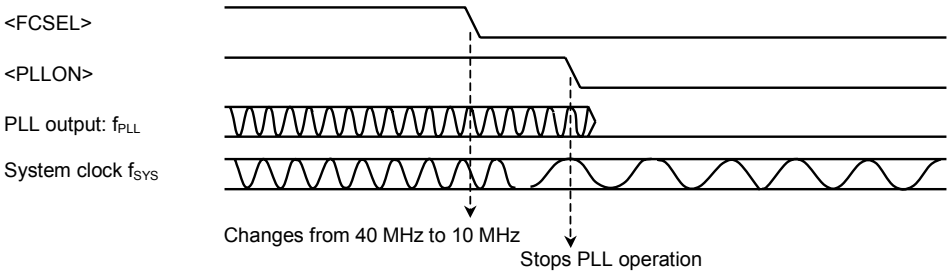
```
PLLCR0    EQU   10E8H
PLLCR1    EQU   10E9H
          LD    (PLLCR1),   1 X X X X X X X B ;  Enables PLL operation and starts lock up.
LUP:      BIT   5, (PLLCR0)                   ;
          JR    Z, LUP                        ;  Detects end of lock up.
          LD    (PLLCR0),   X 1 X X X X X X B ;  Changes fc from 10 MHz to 40 MHz.
X: Don't care
```



Example 2: PLL stopping

```
PLLCR0    EQU    10E8H
PLLCR1    EQU    10E9H
          LD     (PLLCR0), X0XXXXXXB    ;  Changes fc from 40 MHz to10 MHz.
          LD     (PLLCR1), 0XXXXXXXB    ;  Stop PLL.
```

X: Don't care



Limitation point on the use of PLL

1. If you stop PLL operation during using PLL, you should execute following setting in the same order.

```
LD      (PLLCR0), 00H       ;   Change the clock fPLL to fOSCH
LD      (PLLCR1), 00H       ;   PLL stop
```

Examples of settings are below.

(2) Change/stop control

(OK) PLL use mode (f_{PLL}) → Set the STOP mode → High-frequency oscillator operation mode (f_{OSCH}) → PLL stop → Halt (High-frequency oscillator stop)

```
LD      (SYSCR2),     0 X – – 0 1 X – B ;   Set the STOP mode
                                              (This command can execute before use of PLL)
LD      (PLLCR0),     X 0 – X X X X X B ;   Change the system clock fPLL to fOSCH
LD      (PLLCR1),     0 X X X X X X X B ;   PLL stop
HALT                                    ;   Shift to STOP mode
```

(Error) PLL use mode (f_{PLL}) → Set the STOP mode → Halt (High-frequency oscillator stop)

```
LD      (SYSCR2),     0 X – – 0 1 X – B ;   Set the STOP mode
                                              (This command can execute before use of PLL)
HALT                                    ;   Shift to STOP mode
```

3.3.5 Noise Reduction Circuits

Noise reduction circuits are built-in, allowing implementation of the following features.

(1) Reduced drivability for high-frequency oscillator

(2) Single drive for high-frequency oscillator

(3) SFR protection of register contents

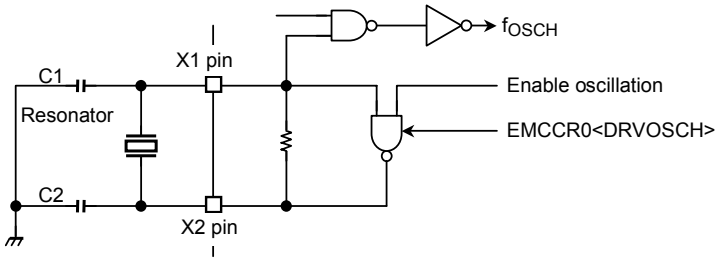
These functions need a setup by EMCCR0, EMCCR1, and EMCCR2 register. (1) to (3) is explained below.

(1) Reduced drivability for high-frequency oscillator

(Purpose)

Reduces noise and power for oscillator when a resonator is used.

(Block diagram)



(Setting method)

The drive ability of the oscillator is reduced by writing "0" to EMCCR0 <DRVOSCH> register. By reset, <DRVOSCH> is initialized to "1" and the oscillator starts oscillation by normal drivability when the power-supply is on.

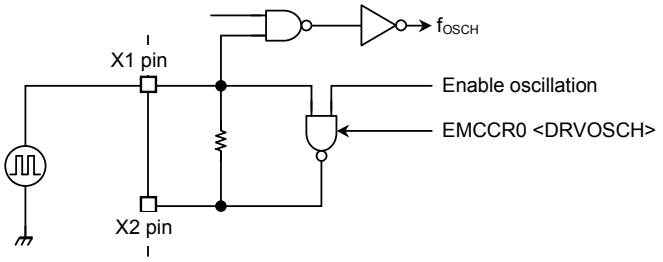
Note: This function (EMCCR0 <DRVOSCH> = "0") is available to use in case of f_{OSCH} = 6 to 10 MHz condition.

(2) Single drive for high-frequency oscillator

(Purpose)

Not need twin drive and protect mistake operation by inputted noise to X2 pin when the external oscillator is used.

(Block diagram)



(Setting method)

The oscillator is disabled and starts operation as buffer by writing "1" to EMCCR0 <EXTIN> register. X2 pin is always outputted "1".

By reset, <EXTIN> is initialized to "0".

(3) Runaway provision with SFR protection register

(Purpose)

Provision in runaway of program by noise mixing.

Write operation to specified SFR is prohibited so that provision program in runaway prevents that it is in the state which is fetch impossibility by stopping of clock, memory control register (memory controller) is changed.

And error handling in runaway becomes easy by INTP0 interruption.

Specified SFR list

1. Memory controller
   B0CSL/H, B1CSL/H, B2CSL/H, B3CSL/H, B4CSL/H, B5CSL/H, BEXCSL/H
   MSAR0, MSAR1, MSAR2, MSAR3, MSAR4, MSAR5
   MAMR0, MAMR1, MAMR2, MAMR3, MAMR4, MAMR5, PMEMCR
2. Clock gear
   SYSCR0, SYSCR1, SYSCR2, EMCCR0
3. PLL
   PLLCR0, PLLCR1

(Operation explanation)

Execute and release of protection (write operation to specified SFR) becomes possible by setting up a double key to EMCCR1 and EMCCR2 register.

(Double key)

1st KEY: Succession writes in 5AH at EMCCR1 and A5H at EMCCR2

2nd KEY: Succession writes in A5H at EMCCR1 and 5AH at EMCCR2

A state of protection can be confirmed by reading EMCCR0 <PROTECT>.

By reset, protection becomes OFF.

And INTP0 interruption occurs when write operation to specified SFR was executed with protection on state.

3.3.6 Stand-by Controller

(1) HALT modes and port drive register

When the HALT instruction is executed, the operating mode switches to IDLE2, IDLE1 or STOP mode, depending on the contents of the SYSCR2 <HALTM1:0> register.

The subsequent actions performed in each mode are as follows:

1. IDLE2: Only the CPU halts.

   The internal I/O is available to select operation during IDLE2 mode by setting the following register.

   Table 3.3.1 shows the registers of setting operation during IDLE2 mode.

Table 3.3.1 SFR Setting Operation during IDLE2 Mode

Internal I/O	SFR
TMRA01	TA01RUN<I2TA01>
TMRA23	TA23RUN<I2TA23>
TMRA45	TA45RUN<I2TA45>
TMRA67	TA67RUN<I2TA67>
TMRB0	TB0RUN<I2TB0>
TMRB1	TB1RUN<I2TB1>
TMRB2	TB2RUN<I2TB2>
TMRB3	TB3RUN<I2TB3>
TMRB4	TB4RUN<I2TB4>
TMRB5	TB5RUN<I2TB5>
SIO0	SC0MOD1<I2S0>
SIO1	SC1MOD1<I2S1>
SIO2	SC2MOD1<I2S2>
SIO3	SC3MOD1<I2S3>
SBI0	SBI0BR0<I2SBI0>
SBI1	SBI1BR0<I2SBI1>
AD Converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>

2. IDLE1: Only the oscillator and the Special timer for clock operate.
3. 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

HALT Mode		IDLE2	IDLE1	STOP
SYSCR2<HALTM1:0>		11	10	01
Block	CPU	Stop		
	I/O ports	The state at the time of "HALT" instruction execution is held.	Table 3.3.8 references	
	TMRA, TMRB	Available to select operation block	Stop	
	SIO, SBI			
	AD converter			
	WDT			
	SDRAMC, Interrupt controller, HSIO, PG (Note)	Operate		

Note 1: When operating PG in the IDLE2 mode, it is necessary to set operation at the time of the IDLE2 mode of the block (TMRA or TMRB) chosen as a trigger as permission.

Note 2: It is necessary to set up the state in each HALT mode of a D/A converter in DAC0CNT0/DAC1CNT0 register before HALT instruction execution.

(2) How to release the HALT mode

These halt states can be released by resetting or requesting an interrupt. The halt release sources are determined by the combination between the states of interrupt mask register <IFF2:0> and the HALT modes. The details for releasing the halt status are shown in Table 3.3.3.

- Released by requesting an interrupt

The operating released from the HALT mode depends on the interrupt enabled status. When the interrupt request level set before executing the "HALT" instruction exceeds the value of interrupt mask register, the interrupt due to the source is processed after releasing the HALT mode, and CPU status executing an instruction that follows the "HALT" instruction. When the interrupt request level set before executing the "HALT" instruction is less than the value of the interrupt mask register, releasing the HALT mode is not executed. (in non-maskable interrupts, interrupt processing is processed after releasing the HALT mode regardless of the value of the mask register.) However only for INT0 to INT7 interrupts, even if the interrupt request level set before executing the halt instruction is less than the value of the interrupt mask register, releasing the HALT mode is executed. In this case, interrupt processing is not performed, and CPU starts executing the instruction next to the halt instruction, but the interrupt request flag is held at "1".

- Releasing by resetting

Releasing all halt status is executed by resetting.

When the STOP mode is released by RESET, it is necessry enough resetting time (see Table 3.3.4 Example of a setting of Warm-up time of oscillator) to set the operation of the oscillator to be stable.

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

Table 3.3.3 Source of Halt State Clearance and Halt Clearance Operation

Status of Received Interrupt			Interrupt Enabled (Interrupt level) ≥ (Interrupt mask)			Interrupt Disabled (Interrupt level) < (Interrupt mask)		
HALT Mode			IDLE2	IDLE1	STOP	IDLE2	IDLE1	STOP
Source of Halt State Clearance	Interrupt	NMI	♦	♦	♦ *1	–	–	–
		INTWDT	♦	×	×	–	–	–
		INT0 to 3 (Note 1)	♦	♦	♦ *1	○	○	○ *1
		INT4 to 7 (PORT) (Note 1) (Note 3)	♦	♦	♦ *1	○	○	○ *1
		INT4 to 7 (TMRB0 to 1) (Note 3)	♦	×	×	×	×	×
		INT8 to B (PORT) (Note 1) (Note 3)	♦	×	×	×	×	×
		INT8 to B (TMRB2 to 3) (Note 3)	♦	×	×	×	×	×
		INTTA0 to 7	♦	×	×	×	×	×
		INTTB00 to 51, INTTBOX	♦	×	×	×	×	×
		INTRX0 to 3, INTTX0 to 3	♦	×	×	×	×	×
		INTAD	♦	×	×	×	×	×
		INTSBI0 to 1	♦	×	×	×	×	×
		INTHSC0 to 1	♦	×	×	×	×	×
		KI (Key On WakeUp) (Note 2)	○	○	○ *1	○	○	○ *1
	RESET		Initialize LSI					

♦: After clearing the HALT mode, CPU starts interrupt processing.

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

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

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

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

Note 1: When the HALT mode is cleared by an INT0 to B interrupt of the level mode in the interrupt enabled status, hold level “H” until starting interrupt processing. If level “L” is set before holding level “L”, interrupt processing is correctly started.

Note 2: Although a KI can cancel all HALT mode states, the function as interruption does not have it.

Note 3: The operation of the HALT release of INT4 to INTB becomes operation of (PORT) when setting it to the INTn input by the port setting. It becomes operation of (TMRB) when setting it to 16 bit timer input.

Note 4: Set the INTSEL register when you use interrupt to which the interrupt factor is used combinedly. Details wish the reference to "Interrupt control of 3.3.4 interrupt controllers (3)".

Example: Releasing IDLE1 mode

An INT0 interrupt clears the halt state when the device is in IDLE1 mode.

```
Address
10003H   LD     (IIMC1), 00H      ;  Selects INT0 interrupt rising edge.
10006H   LD     (IIMC2), 00H      ;  Selects INT0 interrupt edge
10009H   LD     (INTE01), 06H        Sets INT0 interrupt level to 6.
1000CH   EI     5                 ;  Sets interrupt level to 5 for CPU.
1000EH   LD     (SYSCR2), 28H     ;  Sets HALT mode to IDLE1 mode.
10011H   HALT                     ;  Halts CPU.
```


```
10012H   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 3.3.6 illustrates an example of the timing for clearance of the IDLE2 mode halt state by an interrupt.

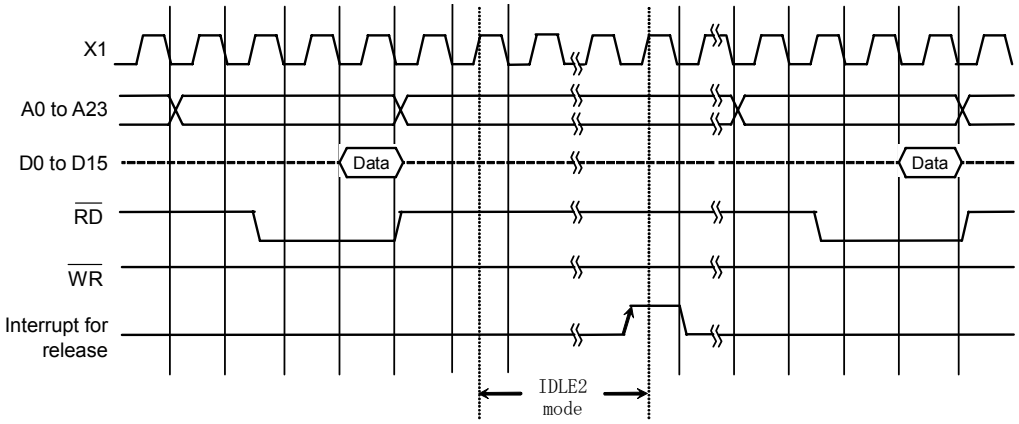


Figure 3.3.6 Timing Chart for IDLE2 Mode Halt State Cleared by Interrupt

2. IDLE1 mode

In IDLE1 mode, only the internal oscillator and Special timer for Clock continue to operate. The system clock stops.

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

Figure 3.3.7 illustrates the timing for clearance of the IDLE1 mode halt state by an interrupt.

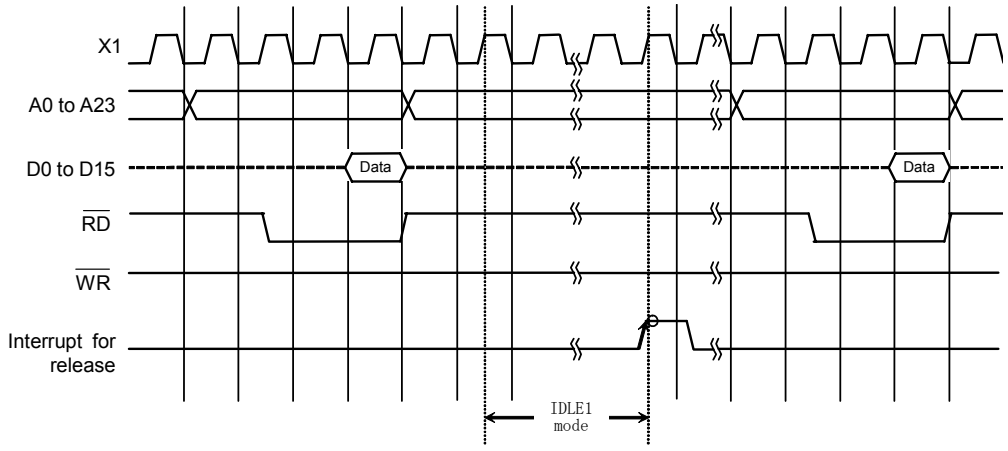


Figure 3.3.7 Timing Chart for IDLE1 Mode Halt State Cleared by Interrupt

3. STOP mode

When STOP mode is selected, all internal circuits stop, including the internal oscillator.

After STOP mode has been cleared system clock output starts when the warm-up time has elapsed, in order to allow oscillation to stabilize.

The example of a setting of the Warm-up time at the time of STOP mode release is shown in Table 3.3.4.

Figure 3.3.8 illustrates the timing for clearance of the STOP mode halt state by an interrupt.

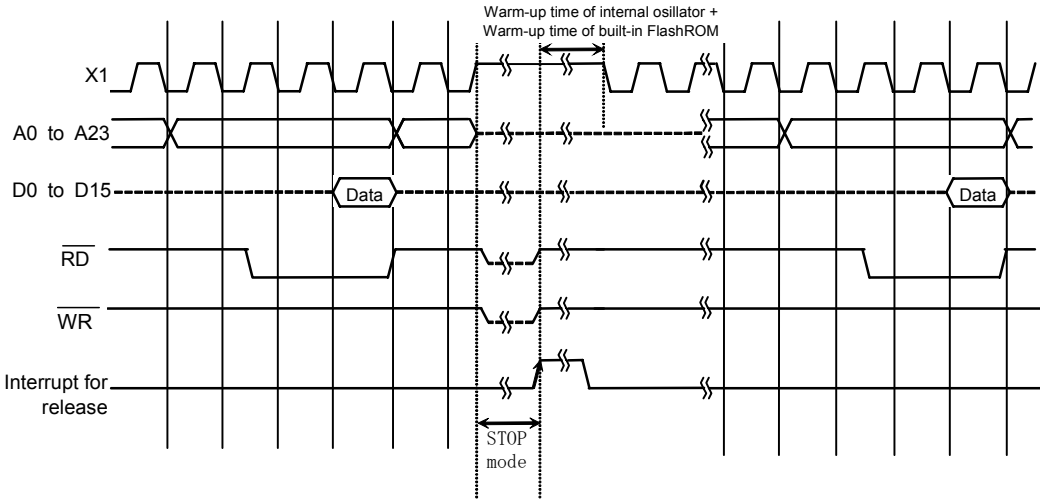


Figure 3.3.8 Timing Chart for STOP Mode Halt State Cleared by Interrupt

Table 3.3.4 Example of a setting of Warm-up time of oscillator (at the time of STOP mode release)

at f_{OSCH} = 16 MHz

SYSCR2<WUPTM1:0>		
01 (2^8)	10 (2^{14})	11 (2^{16})
16 μs	1.024 ms	4.096 ms

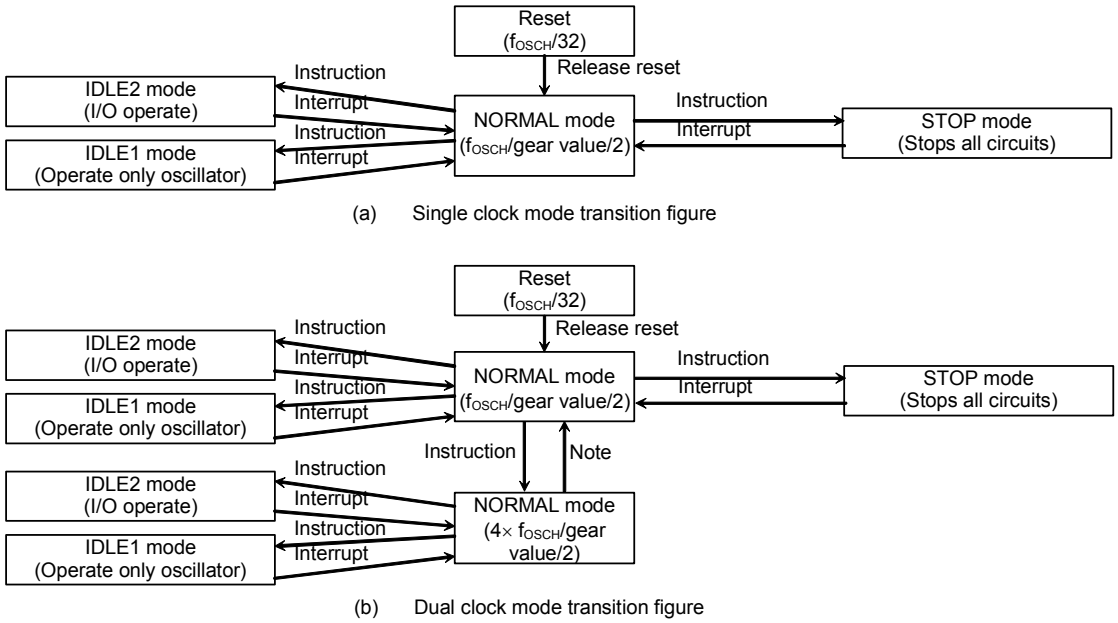
3.3 Clock Function and Stand-by Function

TMP92CM27 contains (1) clock gear, (2) clock doubler (PLL), (3) stand-by controller and (4) noise reducing circuit. They are used for low power, low noise systems.

This chapter is organized as follows:

The clock operating modes are as follows: (a) single clock mode (X1, X2 pins only), (b) dual clock mode (X1, X2 pins and PLL).

Figure 3.3.1 shows a transition figure.



Note 1: If you shift from NORMAL mode with use of PLL to NORMAL mode, execute following setting in the same order.

1) Change CPU clock (PLLCR0 <FCSEL> ← "0")

2) Stop PLL circuit (PLLCR1 <PLLON> ← "0")

Note 2: It's prohibited to shift from NORMAL mode with use of PLL to STOP mode directly.

You should set NORMAL mode once, and then shift to STOP mode. (You should stop high-frequency oscillator after you stop PLL.)

Figure 3.3.1 System Clock Block Diagram

The clock frequency input from the X1 and X2 pins is called f_{OSCH} and clock frequency selected by SYSCR1<SYSCK> is called the system clock f_{FPH}. The system clock f_{SYS} is defined as the divided clock of f_{FPH}, and one cycle of f_{SYS} is defined to as one state.

3.3.1 Block Diagram of System Clock

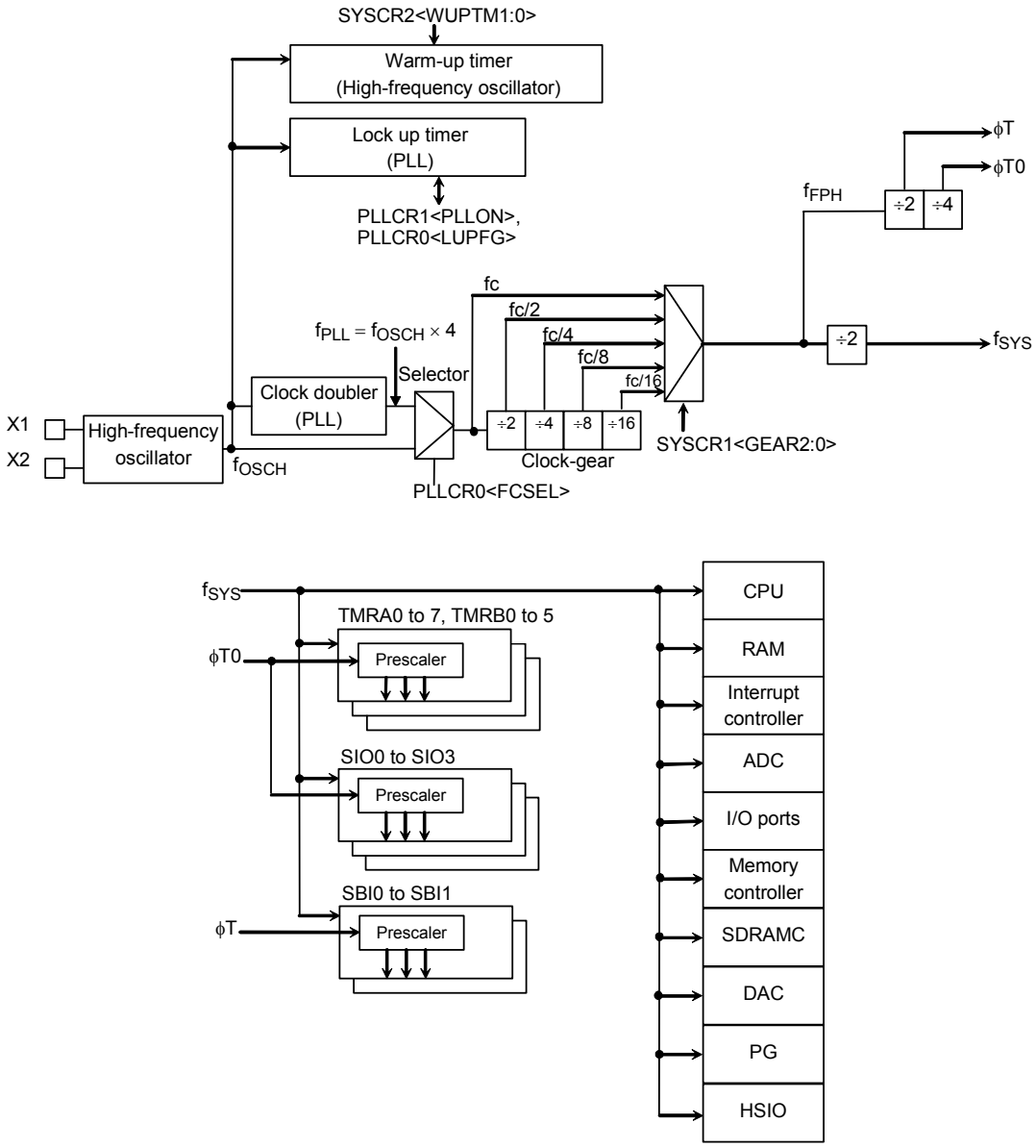


Figure 3.3.2 Block Diagram of System Clock

3.3.2 SFR

SYSCR0 (10E0H)

	7	6	5	4	3	2	1	0
Bit symbol						–		
Read/Write						R/W		
After reset						0		
Function						Always write “0”		

SYSCR1 (10E1H)

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

SYSCR2 (10E2H)

	7	6	5	4	3	2	1	0
Bit symbol	–		WUPTM1	WUPTM0	HALTM1	HALTM0		DRVE
Read/Write	R/W		R/W	R/W	R/W	R/W		R/W
After reset	0		1	0	1	1		0
Function	Always write “0”		Warm-up timer 00: Reserved 01: 2^8/input frequency 10: 2^{14}/input frequency 11: 2^{16}/input frequency		HALT mode 00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode			1: The inside of STOP mode also drives a pin

Note 1: SYSCR0<bit7> can read “1”.

Note 2: SYSCR0<bit6:3>, SYSCR0<bit1:0>, SYSCR1<bit7:3>, and SYSCR2<bit6,1> can read "0".

Figure 3.3.3 SFR for System Clock

		7	6	5	4	3	2	1	0
EMCCR0 (10E3H)	Bit symbol	PROTECT					EXTIN	DRVOSCH	
	Read/Write	R					R/W	R/W	
	After reset	0					0	1	
	Function	Protect flag 0: OFF 1: ON					1: External clock	fc oscillator driver ability 1: Normal 0: Weak	
EMCCR1 (10E4H)	Bit symbol	Switching the protect ON/OFF by write to following 1st-KEY, 2nd-KEY 1st-KEY: EMCCR1 = 5AH, EMCCR2 = A5H in succession write 2nd-KEY: EMCCR1 = A5H, EMCCR2 = 5AH in succession write							
	Read/Write								
	After reset								
	Function								
EMCCR2 (10E5H)	Bit symbol								
	Read/Write								
	After reset								
	Function								

Note 1: EMCCR0<bit0> can read “1”.

Note 2: EMCCR0<bit6:3> can read “0”.

Note 3: In case restarting the oscillator in the stop oscillation state (e.g. Restart the oscillator in STOP mode), set (EMCCR0)<DRVOSCH> = ”1”.

Figure 3.3.4 SFR for System Clock

PLLCR0 (10E8H)

	7	6	5	4	3	2	1	0
Bit symbol		FCSEL	LUPFG					
Read/Write		R/W	R					
After reset		0	0					
Function		Select fc clock 0: f_{OSCH} 1: f_{PLL}	Lock up timer status flag 0: Not end 1: End					

Note 1: Be carefull that logic of PLLCR0<LUPFG> is different from 900/L's DFM.

Note 2: PLLCR0<bit7>,<bit4:0> can read "0".

PLLCR1 (10E9H)

	7	6	5	4	3	2	1	0
Bit symbol	PLLON							
Read/Write	R/W							
After reset	0							
Function	Control on/off 0: OFF 1: ON							

Note 1: PLLCR1<bit6:0> can read "0".

Figure 3.3.5 SFR for PLL

3.3.3 System Clock Controller

The system clock controller generates the system clock signal (f_{SYS}) for the CPU core and internal I/O. It contains a oscillation circuits, PLL and a clock gear circuit for high-frequency (fc) operation. The register SYSCR1<GEAR2:0> sets the high-frequency clock gear to either 1, 2, 4, 8 or 16 (fc, fc/2, fc/4, fc/8 or fc/16). These functions can reduce the power consumption of the equipment in which the device is installed.

The initialization <GEAR2:0> = "100" will cause the system clock (f_{SYS}) to be set to fc/32 (fc/16 × 1/2) after reset.

For example, f_{SYS} is set to 1.25 MHz when the 40 MHz oscillator is connected to the X1 and X2 pins.

(1) Clock gear controller

The f_{FPH} is set according to the contents of the clock gear select register SYSCR1<GEAR2:0> to either fc, fc/2, fc/4, fc/8 or fc/16. Using the clock gear to select a lower value of f_{FPH} reduces power consumption.

Example: Changing to a high-frequency gear

```
SYSCR1    EQU    10E1H

          LD     (SYSCR1), XXXXX001B   ;   Changes fSYS to fc/2.
          LD     (DUMMY), 00H              Dummy instruction
```

X: Don't care

(High-speed clock gear changing)

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

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

Example:

```
SYSCR1    EQU    10E1H
          LD     (SYSCR1), XXXXX010B   ;   Changes fSYS to fc/4.
          LD     (DUMMY), 00H          ;   Dummy instruction
          Instruction to be executed after clock gear has changed
```

3.3.4 Clock Doubler (PLL)

PLL outputs the f_{PLL} clock signal, which is four times as fast as f_{OSCH}. It can use the low-speed-frequency oscillator, even though the internal clock is high-frequency.

A reset initializes PLL to stop status, setting to PLLCR0, PLLCR1 register is needed before use.

Like an oscillator, this circuit requires time to stabilize. This is called the lock up time and it is measured by 16-stage binary counter. Lock up time is about 1.6 ms at f_{OSCH} = 10 MHz.

Note 1: Input frequency limitation for PLL
The limitation of input frequency (High-frequency oscillation) for PLL is following.
f_{OSCH} = 6 to 10 MHz (V_{CC} = 3.0 to 3.6 V)

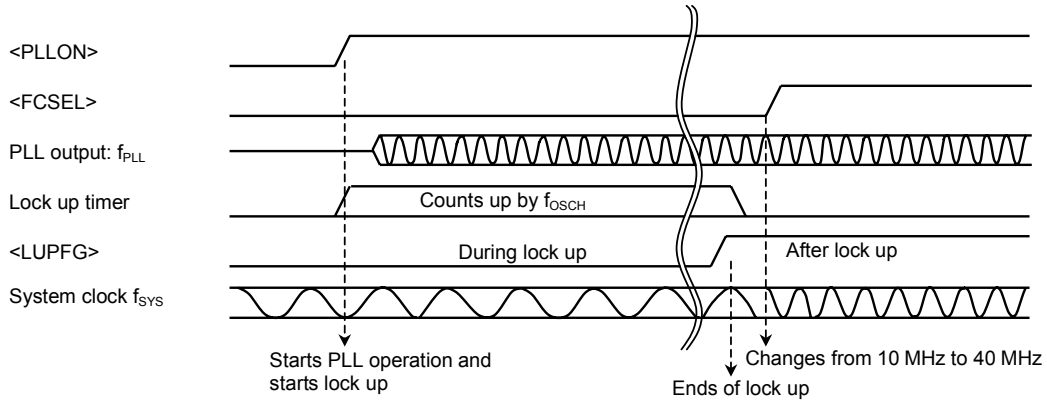
Note 2: PLLCR0 <LUPFG>
The logic of PLLCR0 <LUPFG> is different from 900/L1's DFM.
Be careful to judge an end of lock up time

.

The following is an setting example for PLL starting and PLL stopping.

Example 1: PLL starting

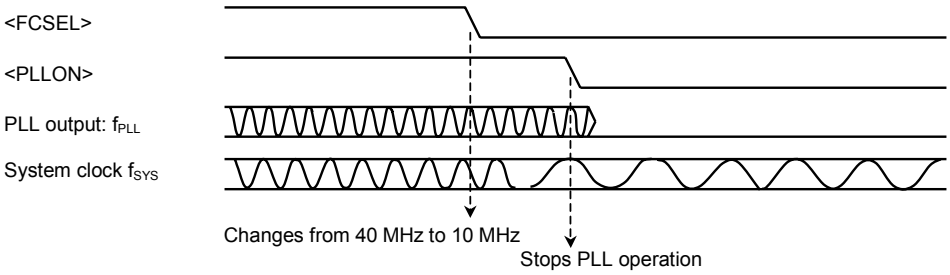
```
PLLCR0    EQU   10E8H
PLLCR1    EQU   10E9H
          LD    (PLLCR1),   1 X X X X X X X B ;  Enables PLL operation and starts lock up.
LUP:      BIT   5, (PLLCR0)                   ;
          JR    Z, LUP                        ;  Detects end of lock up.
          LD    (PLLCR0),   X 1 X X X X X X B ;  Changes fc from 10 MHz to 40 MHz.
X: Don't care
```



Example 2: PLL stopping

```
PLLCR0    EQU    10E8H
PLLCR1    EQU    10E9H
          LD     (PLLCR0), X0XXXXXXB    ;  Changes fc from 40 MHz to10 MHz.
          LD     (PLLCR1), 0XXXXXXXB    ;  Stop PLL.
```

X: Don't care



Limitation point on the use of PLL

1. If you stop PLL operation during using PLL, you should execute following setting in the same order.

```
LD      (PLLCR0), 00H       ;   Change the clock fPLL to fOSCH
LD      (PLLCR1), 00H       ;   PLL stop
```

Examples of settings are below.

(2) Change/stop control

(OK) PLL use mode (f_{PLL}) → Set the STOP mode → High-frequency oscillator operation mode (f_{OSCH}) → PLL stop → Halt (High-frequency oscillator stop)

```
LD      (SYSCR2),     0 X – – 0 1 X – B ;   Set the STOP mode
                                              (This command can execute before use of PLL)
LD      (PLLCR0),     X 0 – X X X X X B ;   Change the system clock fPLL to fOSCH
LD      (PLLCR1),     0 X X X X X X X B ;   PLL stop
HALT                                    ;   Shift to STOP mode
```

(Error) PLL use mode (f_{PLL}) → Set the STOP mode → Halt (High-frequency oscillator stop)

```
LD      (SYSCR2),     0 X – – 0 1 X – B ;   Set the STOP mode
                                              (This command can execute before use of PLL)
HALT                                    ;   Shift to STOP mode
```

3.3.5 Noise Reduction Circuits

Noise reduction circuits are built-in, allowing implementation of the following features.

(1) Reduced drivability for high-frequency oscillator

(2) Single drive for high-frequency oscillator

(3) SFR protection of register contents

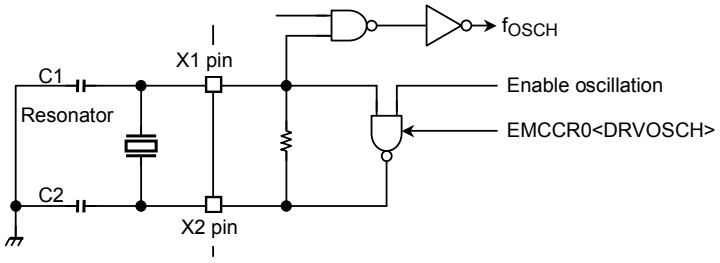
These functions need a setup by EMCCR0, EMCCR1, and EMCCR2 register. (1) to (3) is explained below.

(1) Reduced drivability for high-frequency oscillator

(Purpose)

Reduces noise and power for oscillator when a resonator is used.

(Block diagram)



(Setting method)

The drive ability of the oscillator is reduced by writing "0" to EMCCR0 <DRVOSCH> register. By reset, <DRVOSCH> is initialized to "1" and the oscillator starts oscillation by normal drivability when the power-supply is on.

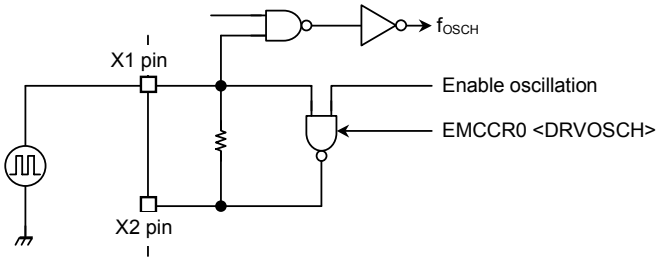
Note: This function (EMCCR0 <DRVOSCH> = "0") is available to use in case of f_{OSCH} = 6 to 10 MHz condition.

(2) Single drive for high-frequency oscillator

(Purpose)

Not need twin drive and protect mistake operation by inputted noise to X2 pin when the external oscillator is used.

(Block diagram)



(Setting method)

The oscillator is disabled and starts operation as buffer by writing "1" to EMCCR0 <EXTIN> register. X2 pin is always outputted "1".

By reset, <EXTIN> is initialized to "0".

(3) Runaway provision with SFR protection register

(Purpose)

Provision in runaway of program by noise mixing.

Write operation to specified SFR is prohibited so that provision program in runaway prevents that it is in the state which is fetch impossibility by stopping of clock, memory control register (memory controller) is changed.

And error handling in runaway becomes easy by INTP0 interruption.

Specified SFR list

1. Memory controller
   B0CSL/H, B1CSL/H, B2CSL/H, B3CSL/H, B4CSL/H, B5CSL/H, BEXCSL/H
   MSAR0, MSAR1, MSAR2, MSAR3, MSAR4, MSAR5
   MAMR0, MAMR1, MAMR2, MAMR3, MAMR4, MAMR5, PMEMCR
2. Clock gear
   SYSCR0, SYSCR1, SYSCR2, EMCCR0
3. PLL
   PLLCR0, PLLCR1

(Operation explanation)

Execute and release of protection (write operation to specified SFR) becomes possible by setting up a double key to EMCCR1 and EMCCR2 register.

(Double key)

1st KEY: Succession writes in 5AH at EMCCR1 and A5H at EMCCR2

2nd KEY: Succession writes in A5H at EMCCR1 and 5AH at EMCCR2

A state of protection can be confirmed by reading EMCCR0 <PROTECT>.

By reset, protection becomes OFF.

And INTP0 interruption occurs when write operation to specified SFR was executed with protection on state.

3.3.6 Stand-by Controller

(1) HALT modes and port drive register

When the HALT instruction is executed, the operating mode switches to IDLE2, IDLE1 or STOP mode, depending on the contents of the SYSCR2 <HALTM1:0> register.

The subsequent actions performed in each mode are as follows:

1. IDLE2: Only the CPU halts.

   The internal I/O is available to select operation during IDLE2 mode by setting the following register.

   Table 3.3.1 shows the registers of setting operation during IDLE2 mode.

Table 3.3.1 SFR Setting Operation during IDLE2 Mode

Internal I/O	SFR
TMRA01	TA01RUN<I2TA01>
TMRA23	TA23RUN<I2TA23>
TMRA45	TA45RUN<I2TA45>
TMRA67	TA67RUN<I2TA67>
TMRB0	TB0RUN<I2TB0>
TMRB1	TB1RUN<I2TB1>
TMRB2	TB2RUN<I2TB2>
TMRB3	TB3RUN<I2TB3>
TMRB4	TB4RUN<I2TB4>
TMRB5	TB5RUN<I2TB5>
SIO0	SC0MOD1<I2S0>
SIO1	SC1MOD1<I2S1>
SIO2	SC2MOD1<I2S2>
SIO3	SC3MOD1<I2S3>
SBI0	SBI0BR0<I2SBI0>
SBI1	SBI1BR0<I2SBI1>
AD Converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>

2. IDLE1: Only the oscillator and the Special timer for clock operate.
3. 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

HALT Mode		IDLE2	IDLE1	STOP
SYSCR2<HALTM1:0>		11	10	01
Block	CPU	Stop		
	I/O ports	The state at the time of "HALT" instruction execution is held.	Table 3.3.8 references	
	TMRA, TMRB	Available to select operation block	Stop	
	SIO, SBI			
	AD converter			
	WDT			
	SDRAMC, Interrupt controller, HSIO, PG (Note)	Operate		

Note 1: When operating PG in the IDLE2 mode, it is necessary to set operation at the time of the IDLE2 mode of the block (TMRA or TMRB) chosen as a trigger as permission.

Note 2: It is necessary to set up the state in each HALT mode of a D/A converter in DAC0CNT0/DAC1CNT0 register before HALT instruction execution.

(2) How to release the HALT mode

These halt states can be released by resetting or requesting an interrupt. The halt release sources are determined by the combination between the states of interrupt mask register <IFF2:0> and the HALT modes. The details for releasing the halt status are shown in Table 3.3.3.

- Released by requesting an interrupt

The operating released from the HALT mode depends on the interrupt enabled status. When the interrupt request level set before executing the "HALT" instruction exceeds the value of interrupt mask register, the interrupt due to the source is processed after releasing the HALT mode, and CPU status executing an instruction that follows the "HALT" instruction. When the interrupt request level set before executing the "HALT" instruction is less than the value of the interrupt mask register, releasing the HALT mode is not executed. (in non-maskable interrupts, interrupt processing is processed after releasing the HALT mode regardless of the value of the mask register.) However only for INT0 to INT7 interrupts, even if the interrupt request level set before executing the halt instruction is less than the value of the interrupt mask register, releasing the HALT mode is executed. In this case, interrupt processing is not performed, and CPU starts executing the instruction next to the halt instruction, but the interrupt request flag is held at "1".

- Releasing by resetting

Releasing all halt status is executed by resetting.

When the STOP mode is released by RESET, it is necessry enough resetting time (see Table 3.3.4 Example of a setting of Warm-up time of oscillator) to set the operation of the oscillator to be stable.

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

Table 3.3.3 Source of Halt State Clearance and Halt Clearance Operation

Status of Received Interrupt			Interrupt Enabled (Interrupt level) ≥ (Interrupt mask)			Interrupt Disabled (Interrupt level) < (Interrupt mask)		
HALT Mode			IDLE2	IDLE1	STOP	IDLE2	IDLE1	STOP
Source of Halt State Clearance	Interrupt	NMI	♦	♦	♦ *1	–	–	–
		INTWDT	♦	×	×	–	–	–
		INT0 to 3 (Note 1)	♦	♦	♦ *1	○	○	○ *1
		INT4 to 7 (PORT) (Note 1) (Note 3)	♦	♦	♦ *1	○	○	○ *1
		INT4 to 7 (TMRB0 to 1) (Note 3)	♦	×	×	×	×	×
		INT8 to B (PORT) (Note 1) (Note 3)	♦	×	×	×	×	×
		INT8 to B (TMRB2 to 3) (Note 3)	♦	×	×	×	×	×
		INTTA0 to 7	♦	×	×	×	×	×
		INTTB00 to 51, INTTBOX	♦	×	×	×	×	×
		INTRX0 to 3, INTTX0 to 3	♦	×	×	×	×	×
		INTAD	♦	×	×	×	×	×
		INTSBI0 to 1	♦	×	×	×	×	×
		INTHSC0 to 1	♦	×	×	×	×	×
		KI (Key On WakeUp) (Note 2)	○	○	○ *1	○	○	○ *1
	RESET		Initialize LSI					

♦: After clearing the HALT mode, CPU starts interrupt processing.

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

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

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

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

Note 1: When the HALT mode is cleared by an INT0 to B interrupt of the level mode in the interrupt enabled status, hold level “H” until starting interrupt processing. If level “L” is set before holding level “L”, interrupt processing is correctly started.

Note 2: Although a KI can cancel all HALT mode states, the function as interruption does not have it.

Note 3: The operation of the HALT release of INT4 to INTB becomes operation of (PORT) when setting it to the INTn input by the port setting. It becomes operation of (TMRB) when setting it to 16 bit timer input.

Note 4: Set the INTSEL register when you use interrupt to which the interrupt factor is used combinedly. Details wish the reference to "Interrupt control of 3.3.4 interrupt controllers (3)".

Example: Releasing IDLE1 mode

An INT0 interrupt clears the halt state when the device is in IDLE1 mode.

```
Address
10003H   LD     (IIMC1), 00H      ;  Selects INT0 interrupt rising edge.
10006H   LD     (IIMC2), 00H      ;  Selects INT0 interrupt edge
10009H   LD     (INTE01), 06H        Sets INT0 interrupt level to 6.
1000CH   EI     5                 ;  Sets interrupt level to 5 for CPU.
1000EH   LD     (SYSCR2), 28H     ;  Sets HALT mode to IDLE1 mode.
10011H   HALT                     ;  Halts CPU.
```


```
10012H   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 3.3.6 illustrates an example of the timing for clearance of the IDLE2 mode halt state by an interrupt.

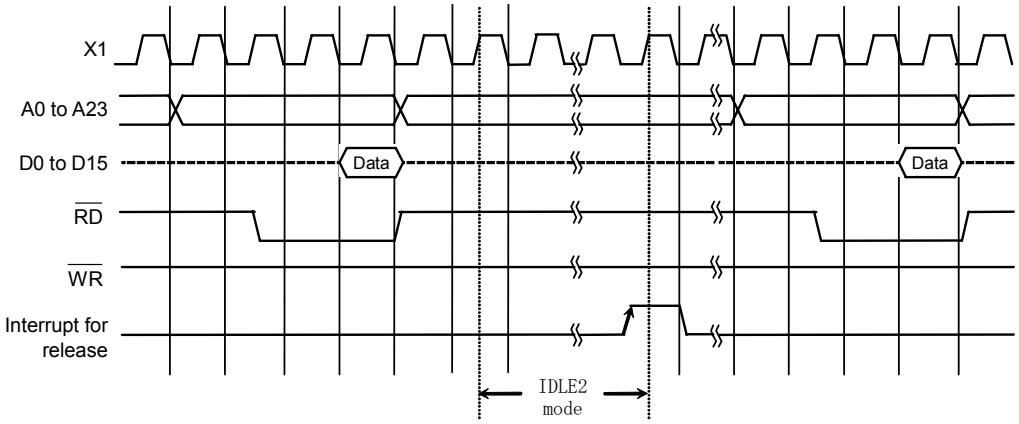


Figure 3.3.6 Timing Chart for IDLE2 Mode Halt State Cleared by Interrupt

2. IDLE1 mode

In IDLE1 mode, only the internal oscillator and Special timer for Clock continue to operate. The system clock stops.

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

Figure 3.3.7 illustrates the timing for clearance of the IDLE1 mode halt state by an interrupt.

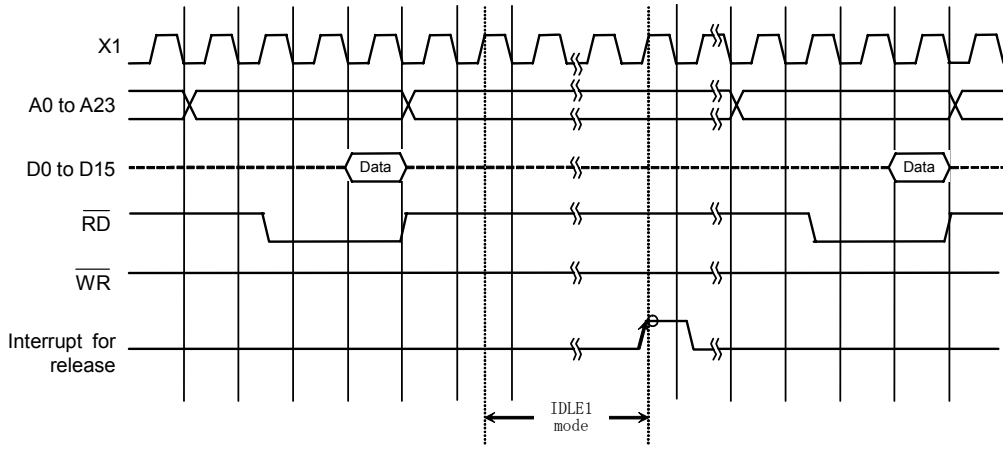


Figure 3.3.7 Timing Chart for IDLE1 Mode Halt State Cleared by Interrupt

3. STOP mode

When STOP mode is selected, all internal circuits stop, including the internal oscillator.

After STOP mode has been cleared system clock output starts when the warm-up time has elapsed, in order to allow oscillation to stabilize.

The example of a setting of the Warm-up time at the time of STOP mode release is shown in Table 3.3.4.

Figure 3.3.8 illustrates the timing for clearance of the STOP mode halt state by an interrupt.

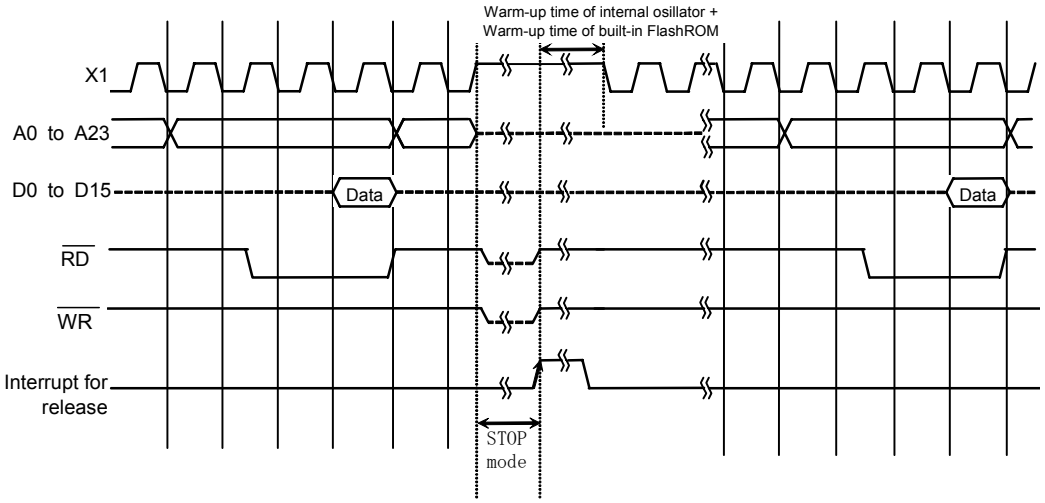


Figure 3.3.8 Timing Chart for STOP Mode Halt State Cleared by Interrupt

Table 3.3.4 Example of a setting of Warm-up time of oscillator (at the time of STOP mode release)

at f_{OSCH} = 16 MHz

SYSCR2<WUPTM1:0>		
01 (2^8)	10 (2^{14})	11 (2^{16})
16 μs	1.024 ms	4.096 ms

3.4 Interrupt

Interrupts of TLCS-900/H1 are controlled by the CPU interrupt mask flip-flop (IFF2:0) and by the built-in interrupt controller.

The TMP92CM27 has a total of 71 interrupts divided into the following types:

- Interrupts generated by CPU: 9 sources
  (Software interrupts: 8 sources, illegal instruction interrupt: 1 source)
- External interrupts ($\overline{\text{NMI}}$ and INT0 to INTB): 13 sources
- Internal I/O interrupts: 41 sources
- Micro DMA transfer end interrupts: 8 sources

A individual interrupt vector number (Fixed) is assigned to each interrupt.

One of six priority level (Variable) can be assigned to each maskable interrupt.

The priority level of non-maskable interrupts are fixed at "7" as the highest level.

When an interrupt is generated, the interrupt controller sends the priority of that interrupt to the CPU. If multiple interrupts are generated simultaneously, the interrupt controller sends the interrupt with the highest priority to the CPU. (The highest priority is level 7 using for non-maskable interrupts.)

The CPU compares the priority level of the interrupt with the value of the CPU interrupts mask register <IFF2:0>. If the priority level of the interrupt is higher than the value of the interrupt mask register, the CPU accepts the interrupt.

The interrupt mask register <IFF2:0> value can be updated using the value of the EI instruction (EI num sets <IFF2:0> data to num).

For example, specifying "EI3" enables the maskable interrupts which priority level set in the interrupt controller is 3 or higher, and also non-maskable interrupts.

Operationally, the DI instruction (<IFF2:0> = 7) is identical to the EI7 instruction. DI instruction is used to disable maskable interrupts because of the priority level of maskable interrupts is 0 to 6. The EI instruction is valid immediately after execution.

In addition to the above general-purpose interrupt processing mode, TLCS-900/H1 has a micro DMA interrupt processing mode as well. The CPU can transfer the data (1/2/4 bytes) automatically in micro DMA mode, therefore this mode is used for speed-up interrupt processing, such as transferring data to the internal or external peripheral I/O. Moreover, TMP92CM27 has software start function for micro DMA processing request by the software not by the hardware interrupt.

Figure 3.4.1 shows the overall interrupt processing flow.

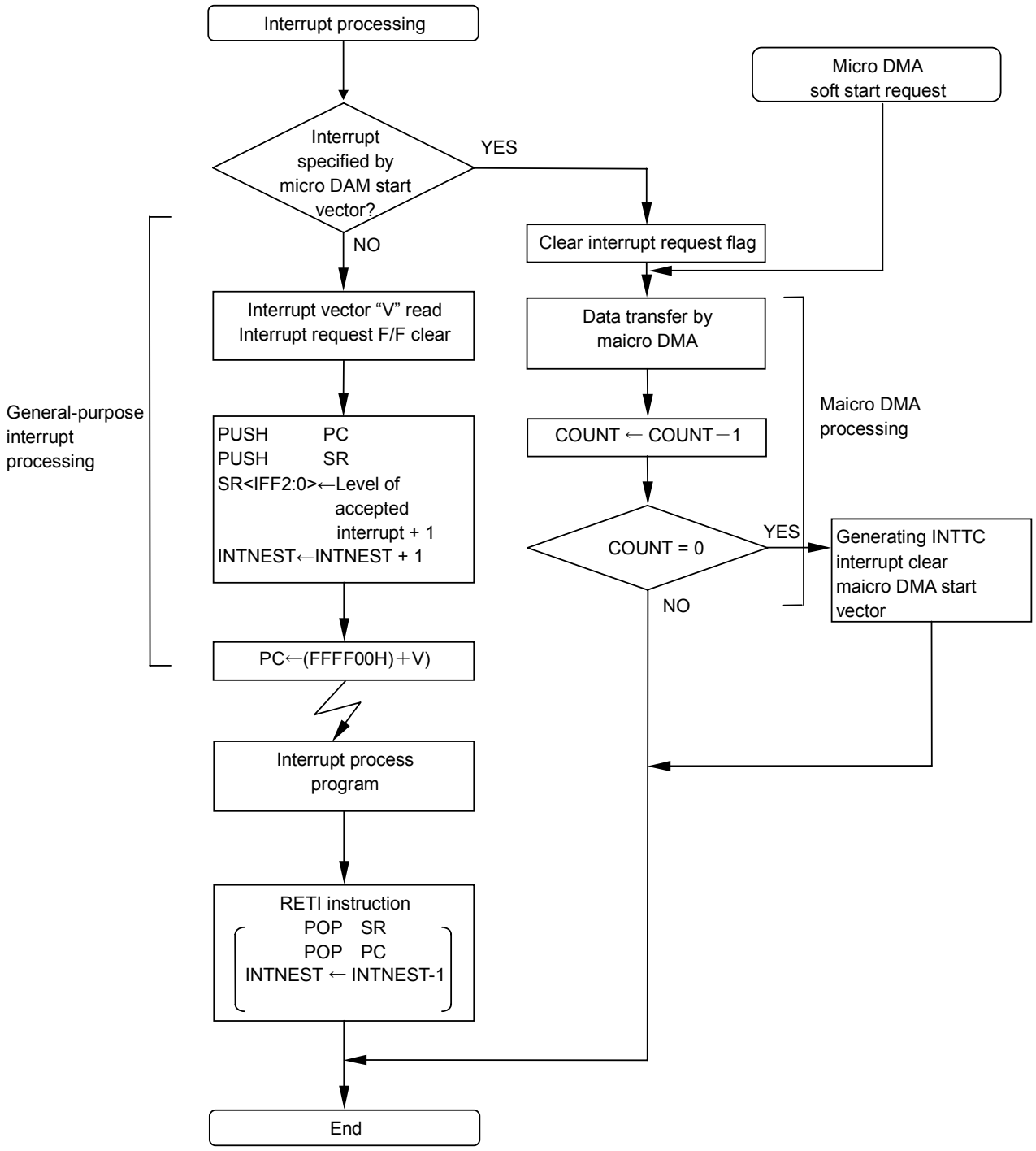


Figure 3.4.1 Interrupt and Micro DMA Processing Sequence

3.4.1 General-purpose Interrupt Processing

When the CPU accepts an interrupt, it usually performs the following sequence of operations. However, when a software interrupt and illegal instruction interrupt are generated by CPU, CPU flies (1) and (3) and performs only the process of (2), (4), and (5).

(1) The CPU reads the interrupt vector from the interrupt controller.
If the same level interrupts occur simultaneously, the interrupt controller generates an interrupt vector in accordance with the default priority and clears the interrupt request.
(The default priority is already fixed for each interrupt: The smaller vector value has the higher priority level.)

(2) The CPU pushes the value of program counter (PC) and status register (SR) onto the stack area (indicated by XSP).

(3) The CPU sets the value which is the priority level of the accepted interrupt plus 1 (+1) to the interrupt mask register <IFF2:0>. However, if the priority level of the accepted interrupt is "7", the register's value is set to "7".

(4) The CPU increases the interrupt nesting counter INTNEST by 1 (+1).

(5) The CPU jumps to the address indicated by the data at address "FFFF00H + Interrupt vector" and starts the interrupt processing routine.

When the CPU completed the interrupt processing, use the "RETI" instruction to return to the main routine. "RETI" restores the contents of program counter (PC) and status register (SR) from the stack and decreases the interrupt nesting counter INTNEST by 1(–1).

Non-maskable interrupts cannot be disabled by a user program. However maskable interrupts 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 which has a priority level equal to or greater than the value of the CPU interrupt mask register <IFF2:0> comes out, the CPU accepts its interrupt. Then, the CPU interrupt mask register <IFF2:0> is set to the value of the priority level for the accepted interrupt plus 1(+1). Therefore, if an interrupt is generated with a higher level than the current interrupt during it's processing, the CPU accepts the later interrupt and goes to the nesting status of interrupt processing.

Moreover, if the CPU receives another interrupt request while performing the said (1) to (5) processing steps of the current interrupt, the latest interrupt request is sampled immediately after execution of the first instruction of the current interrupt processing routine. Specifying "DI" as the start instruction disables maskable interrupt nesting.

A reset initializes the interrupt mask register <IFF2:0> to "7", disabling all maskable interrupts.

Table 3.4.1 shows the TMP92CM27 interrupt vectors and micro DMA start vectors. The address FFFF00H to FFFFFFH (256 bytes) is assigned for the interrupt vector area.

Table 3.4.1 TMP92CM27 Inerrupt Vectors and Micro DMA Start Vectors

Default Priority	Type	Interrupt Source	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		$\overline{\text{NMI}}$: External interrupt input pin	0020H	FFFF20H	
10		INTWD: Watchdog Timer	0024H	FFFF24H	
-		Micro DMA (Note 1)	-	-	-
11		INT0: External interrupt input pin	0028H	FFFF28H	0AH (Note 1)
12		INT1: External interrupt input pin	002CH	FFFF2CH	0BH (Note 1)
13		INT2: External interrupt input pin	0030H	FFFF30H	0CH (Note 1)
14		INT3: External interrupt input pin	0034H	FFFF34H	0DH (Note 1)
15		INT4: External interrupt input pin	0038H	FFFF38H	0EH (Note 1)
16		INT5: External interrupt input pin	003CH	FFFF3CH	0FH (Note 1)
17		INT6: External interrupt input pin	0040H	FFFF40H	10H (Note 1)
18		INT7: External interrupt input pin	0044H	FFFF44H	11H (Note 1)
19		INTTA0: 8-bit timer 0	0048H	FFFF48H	12H
20		INTTA1: 8-bit timer 1	004CH	FFFF4CH	13H
21		INTTA2: 8-bit timer 2	0050H	FFFF50H	14H
22		INTTA3: 8-bit timer 3	0054H	FFFF54H	15H
23		INTTA4: 8-bit timer 4	0058H	FFFF58H	16H
24		INTTA5: 8-bit timer 5 INT8: External interrupt input pin	005CH	FFFF5CH	17H (Note 1) (Note 2)
25		INTTA6: 8-bit timer 6	0060H	FFFF60H	18H
26		INTTA7: 8-bit timer 7 INT9: External interrupt input pin	0064H	FFFF64H	19H (Note 1) (Note 2)
27		INTRX0: Serial 0 (SIO0) receive	0068H	FFFF68H	1AH (Note 1)
28		INTTX0: Serial 0 (SIO0) transmission	006CH	FFFF6CH	1BH
29		INTRX1: Serial 1 (SIO1) receive	0070H	FFFF70H	1CH (Note 1)
30		INTTX1: Serial 1 (SIO1) transmission	0074H	FFFF74H	1DH
31		INTRX2: Serial 2 (SIO2) receive	0078H	FFFF78H	1EH (Note 1)
32		INTTX2: Serial 2 (SIO2) transmission	007CH	FFFF7CH	1FH
33		INTRX3: Serial 3 (SIO3) receive	0080H	FFFF80H	20H (Note 1)
34		INTTX3: Serial 3 (SIO3) transmission	0084H	FFFF84H	21H
35		INTSBI0: SBI0 I2CBUS transfer end	0088H	FFFF88H	22H
36		INTSBI1: SBI1 I2CBUS transfer end	008CH	FFFF8CH	23H
37		INTA: External interrupt input pin	0090H	FFFF90H	24H
38		INTHSC0: High speed serial (HSC0)	0094H	FFFF94H	25H
39		INTB: External interrupt input pin	0098H	FFFF98H	26H
40		INTHSC1: High speed serial (HSC1)	009CH	FFFF9CH	27H
41		INTTB00: 16-bit timer 0	00A0H	FFFFA0H	28H
42		INTTB01: 16-bit timer 0	00A4H	FFFFA4H	29H
43		INTTB10: 16-bit timer 1	00A8H	FFFFA8H	2AH
44		INTTB11: 16-bit timer 1	00ACH	FFFFACH	2BH
45		INTTB20: 16-bit timer 2	00B0H	FFFFB0H	2CH
46		INTTB21: 16-bit timer 2	00B4H	FFFFB4H	2DH

47	Maskable	INTTB30: 16-bit timer 3 INTTB31: 16-bit timer 3	00B8H	FFFFB8H	2EH (Note 2)
48		INTTB40: 16-bit timer 4 INTTB41: 16-bit timer 4	00BCH	FFFFBCH	2FH (Note 2)
49		INTTB50: 16-bit timer 5 INTTB51: 16-bit timer 5	00C0H	FFFFC0H	30H (Note 2)
50		INTTBOX: 16-bit timer (Overflow) Interruption occurs in one overflow interruption of the followings. INTTBOF0: 16-bit timer 0 (Overflow) INTTBOF1: 16-bit timer 1 (Overflow) INTTBOF2: 16-bit timer 2 (Overflow) INTTBOF3: 16-bit timer 3 (Overflow) INTTBOF4: 16-bit timer 4 (Overflow) INTTBOF5: 16-bit timer 5 (Overflow)	00C4H	FFFFC4H	31H (Note 3)
51		INTAD: AD conversion end	00C8H	FFFFC8H	32H
52		INTP0: Protect 0 (Write to special SFR)	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 -

Note 1: When standing-up micro DMA, set at edge detect mode.

Note 2: The default priorities 24, 26, 47 to 49 are making the interruption factor serve a double purpose. It is necessary to choose the interruption factor used in an interruption factor selection register. Therefore, interruption cannot be used simultaneously.

Note 3: The default priority 50 is making the interruption factor serve a double purpose. The interruption factor assigned to this default priority 50 can be used simultaneously. Of which interruption factor interruption occurred should interrupt, and please check it in a generating flag register.

Note 4: Micro DMA stands up prior to other maskable interrupt.

3.4.2 Micro DMA

In addition to general purpose interrupt processing, the TMP92CM27 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 stand-by state by HALT instruction, the requirement of micro DMA will be ignored (pending).

Micro DMA is supported 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 specified by the micro DMA start vector register is generated, the micro DMA triggers a micro DMA request to the CPU at interrupt priority highest level and starts processing the request in spite of any interrupt source's level. The micro DMA is ignored on <IFF2:0> = "7". The 8 micro DMA channels allow micro DMA processing to be set for up to 8 types of interrupts at any one time.

When micro DMA is accepted, the interrupt request flip-flop assigned to that channel is cleared. The data are automatically transferred once (1/2/4 bytes) from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decreased by 1 (–1).

If the decreased result is "0",

- CPU send micro DMA transfer end interrupt (INTTCn) to interrupt controller
- Interrupt controller is generated micro DMA transfer end interrupt
- Micro DMA start vector register is cleared to "0", the next micro DMA operation is disabled
- Micro DMA processing terminates

If the decreased result is not "0", the micro DMA processing completes if it isn't specified the say later burst mode. In this case, the micro DMA transfer end interrupt (INTTCn) aren't generated.

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

The priority of the micro DMA transfer end interrupt is defined by the interrupt level and the default priority as the same as the other maskable interrupt. If a micro DMA request is set for more than one channel at the same time, the priority is not based on the interrupt priority level but on the channel number. The smaller channel number has the higher priority (Channel 0 (High) > Channel 7 (Low)).

While the register for setting the transfer source/transfer destination addresses is a 32-bit control register, this register can only effectively output 24-bit addresses. Accordingly, micro DMA can access 16 Mbytes.

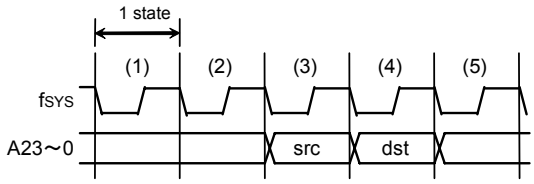
Three micro DMA transfer modes are supported: one-byte transfers, 2-byte transfer and 4-byte transfer. After a transfer in any mode, the transfer source and transfer destination

addresses will either be incremented or decremented, or will remain unchanged. This simplifies the transfer of data from memory to memory, from I/O to memory, from memory to I/O, and from I/O to I/O. For details of the various transfer modes, refer 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 51 different interrupts – the 50 interrupts shown in the micro DMA start vectors in Table 3.4.1 and a micro DMA soft start.

Figure 3.4.2 shows 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: Both source and destination memory are internal RAM and multiples by 4 numbered source and destination addresses.)



(Note)Actually, src and dst address are not output to A23 to A0 pins because they are address of internal RAM.

Figure 3.4.2 Timing for Micro DMA Cycle

State (1),(2): Instruction fetch cycle (Prefetches the next instruction code)

State (3) : Micro DMA read cycle

State (4) : Micro DMA write cycle

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

(2) Soft start function

In addition to starting the micro DMA function by interrupts, TMP92CM27 includes a micro DMA software start function that starts micro DMA on the generation of the write cycle to the DMAR register.

Writing "1" to each bit of DMAR register causes micro DMA once. At the end of transfer, the corresponding bit of the DMAR register is automatically cleared to "0".

Only one channel can be set once for micro DMA.

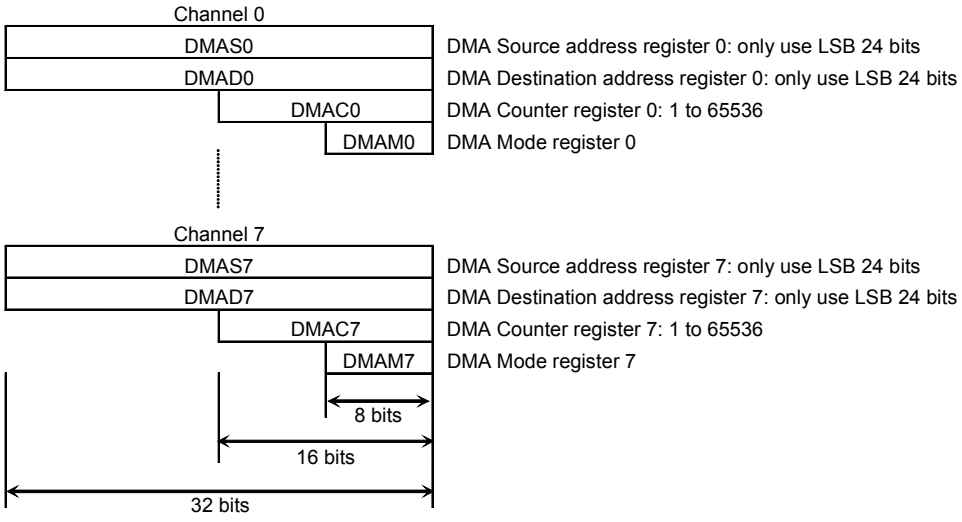
When programming again "1" to the DMAR register, check whether the bit is "0" before programming "1".

When a burst is specified by DMAB register, data is continuously transferred until the value in the micro DMA transfer counter is "0" after start up of the micro DMA.

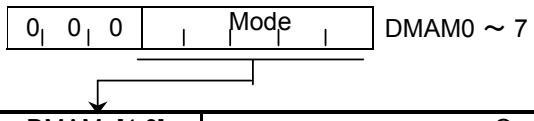
Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA request	109H (Prohibit RMW)	DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
			R/W							
			0	0	0	0	0	0	0	0

(3) Transfer control registers

The transfer source address and the transfer destination address are set in the following registers. Data setting for these registers is done by an "LDC cr, r" instruction.



(4) Detailed description of the transfer mode register



DMAMn[4:0]	Operation	Execution Time
0 0 0 z z	Destination address INC mode (DMADn +) ← (DMASn) DMACn ← DMACn - 1 If DMACn = 0 then INTTCn	5 states
0 0 1 z z	Destination address DEC mode (DMADn -) ← (DMASn) DMACn ← DMACn - 1 If DMACn = 0 then INTTCn	5 states
0 1 0 z z	Source address INC mode (DMADn) ← (DMASn +) DMACn ← DMACn - 1 If DMACn = 0 then INTTCn	5 states
0 1 1 z z	Source address DEC mode (DMADn) ← (DMASn -) DMACn ← DMACn – 1 If DMACn = 0 then INTTCn	5 states
1 0 0 z z	Source address and Destination address INC mode (DMADn +) ← (DMASn +) DMACn ← DMACn – 1 If DMACn = 0 then INTTCn	6 states
1 0 1 z z	Source address and Destination address DEC mode (DMADn -) ← (DMASn -) DMACn ← DMACn – 1 If DMACn = 0 then INTTCn	6 states
1 1 0 z z	Source address and Destination address Fixed mode (DMADn) ← (DMASn) DMACn ← DMACn – 1 If DMACn = 0 then INTTCn	5 states
1 1 1 0 0	Counter mode DMASn ← DMASn + 1 DMACn ← DMACn – 1 If DMACn = 0 then INTTCn	5 states

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

Note1: 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)
“I/O” signifies fixed memory addresses; “memory” signifies incremented or decremented memory addresses.

Note2: The transfer mode register should not be set to any value other than those listed above.

Note3: The execution state number shows number of best case (1-state memory access).

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 62 interrupts 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 reset occurs
- When the CPU reads the channel vector after accepted its interrupt
- When executing an instruction that clears the interrupt (Write DMA start vector to INTCLR register)
- When the CPU receives a micro DMA request
- When the micro DMA burst transfer is terminated

An interrupt priority can be set independently for each interrupt source by writing the priority to the interrupt priority setting register (e.g., INTEPAD or INTB01). 6 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. If interrupt request with the same level are generated at the same time, 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.

The interrupt controller sends the interrupt request with the highest priority among the simultaneous interrupts and its vector address to the CPU. The CPU compares the priority value <IFF2:0> in the status register by the interrupt request signal with the priority value set; if the latter is higher, the interrupt is accepted. Then the CPU sets a value higher than the priority value by 1 (+1) in the CPU SR<IFF2:0>. Interrupt request where the priority value equals or is higher than the set value are accepted simultaneously during the previous interrupt routine.

When interrupt processing is completed (after execution of the RETI instruction), the CPU restores the priority value saved in the stack before the interrupt was generated to the CPU SR<IFF2:0>.

The interrupt controller also has registers (8 channels) 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 register (e.g., DMAS and DMAD) prior to the micro DMA processing.

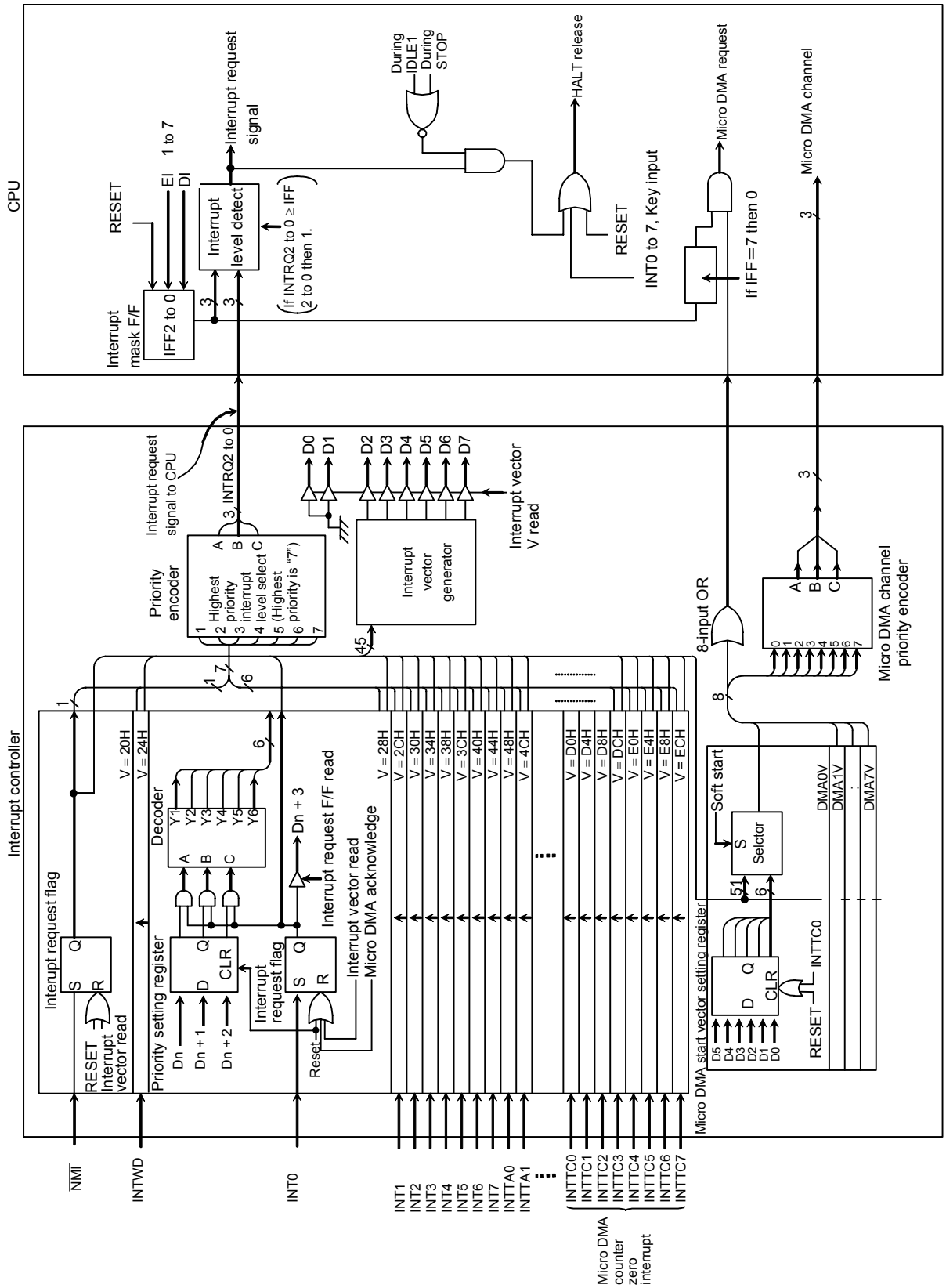


Figure 3.4.3 Block Diagram of Interrupt Controller

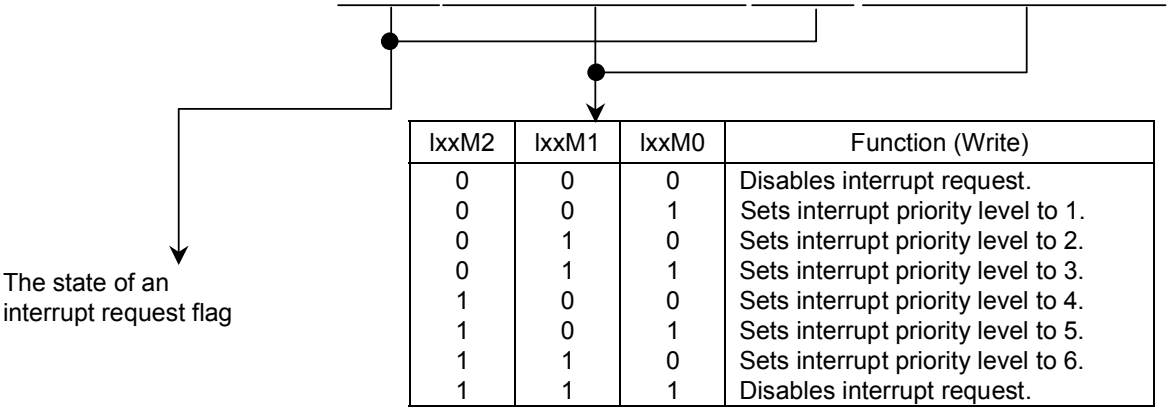
(1) Interrupt priority setting registers

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTE01	INT0 & INT1 Enable	D0H	INT1				INT0			
			I1C	I1M2	I1M1	I1M0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0				0			
INTE23	INT2 & INT3 Enable	D1H	INT3				INT2			
			I3C	I3M2	I3M1	I3M0	I2C	I2M2	I2M1	I2M0
			R	R/W			R	R/W		
			0				0			
INTE45	INT4 & INT5 Enable	D2H	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R	R/W			R	R/W		
			0				0			
INTE67	INT6 & INT7 Enable	D3H	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R	R/W			R	R/W		
			0				0			
INTETA01	INTTA0 & INTTA1 Enable	D4H	INTTA1(Timer1)				INTTA0(Timer0)			
			ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0
			R	R/W			R	R/W		
			0				0			
INTETA23	INTTA2 & INTTA3 Enable	D5H	INTTA3(Timer3)				INTTA2(Timer2)			
			ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0
			R	R/W			R	R/W		
			0				0			
INTE8TA45	INTTA4 & INT8/INTTA5 Enable	D6H	INT8/INTTA5(Timer5)				INTTA4(Timer4)			
			ITA5C	ITA5M2	ITA5M1	ITA5M0	ITA4C	ITA4M2	ITA4M1	ITA4M0
			R	R/W			R	R/W		
			0				0			
INTE9TA67	INTTA6 & INT9/INTTA7 Enable	D7H	INT9/INTTA7(Timer7)				INTTA6(Timer6)			
			ITA7C	ITA7M2	ITA7M1	ITA7M0	ITA6C	ITA6M2	ITA6M1	ITA6M0
			R	R/W			R	R/W		
			0				0			

The state of an interrupt request flag

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request.
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 request.

Symbol	NAME	address	7	6	5	4	3	2	1	0
INTES0	INTRX0 & INTTX0 Enable	D8H	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0				0			
INTES1	INTRX1 & INTTX1 Enable	D9H	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0				0			
INTES2	INTRX2 & INTTX2 Enable	DAH	INTTX2				INTRX2			
			ITX2C	ITX2M2	ITX2M1	ITX2M0	IRX2C	IRX2M2	IRX2M1	IRX2M0
			R	R/W			R	R/W		
			0				0			
INTES3	INTRX3 & INTTX3 Enable	DBH	INTTX3				INTRX3			
			ITX3C	ITX3M2	ITX3M1	ITX3M0	IRX3C	IRX3M2	IRX3M1	IRX3M0
			R	R/W			R	R/W		
			0				0			
INTESB0	INTSBI0 Enable	DCH	-				INTSBI0			
			-	-	-	-	ISBI0C	ISBI0M2	ISBI0M1	ISBI0M0
							R	R/W		
			Note: Write “0”				0			
INTESB1	INTSBI1 Enable	DDH	-				INTSBI1			
			-	-	-	-	ISBI1C	ISBI1M2	ISBI1M1	ISBI1M0
							R	R/W		
			Note: Write “0”				0			
INTEAHSC0	INTA & INTHSC0 Enable	DEH	INTHSC0				INTA			
			IHSC0C	IHSC0M2	IHSC0M1	IHSC0M0	IAC	IAM2	IAM1	IAM0
			R	R/W			R	R/W		
			0				0			
INTEBHSC1	INTB & INTHSC1 Enable	DFH	INTHSC1				INTB			
			IHSC1C	IHSC1M2	IHSC1M1	IHSC1M0	IBC	IBM2	IBM1	IBM0
			R	R/W			R	R/W		
			0				0			
INTETB0	INTTB00 & INTTB01 Enable	E0H	INTTB01				INTTB00			
			ITB01C	ITB01M2	ITB01M1	ITB01M0	ITB00C	ITB00M2	ITB00M1	ITB00M0
			R	R/W			R	R/W		
			0				0			



IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request.
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 request.

Symbol	NAME	address	7	6	5	4	3	2	1	0
INTETB1	INTTB10 & INTTB11 Enable	E2H	INTTB11				INTTB10			
			ITB11C	ITB11M2	ITB11M1	ITB11M0	ITB10C	ITB10M2	ITB10M1	ITB10M0
			R	R/W			R	R/W		
			0				0			
INTETB2	INTTB20 & INTTB21 Enable	E5H	INTTB21				INTTB20			
			ITB21C	ITB21M2	ITB21M1	ITB21M0	ITB20C	ITB20M2	ITB20M1	ITB20M0
			R	R/W			R	R/W		
			0				0			
INTETB3	INTTB30 & INTTB31 Enable	E6H	-				INTTB31/INTTB30			
			-	-	-	-	ITB3XC	ITB3XM2	ITB3XM1	ITB3XM0
							R	R/W		
			Note: Write “0”				0			
INTETB4	INTTB40 & INTTB41 Enable	E7H	-				INTTB41/INTTB40			
			-	-	-	-	ITB4XC	ITB4XM2	ITB4XM1	ITB4XM0
							R	R/W		
			Note: Write “0”				0			
INTETB5	INTTB50 & INTTB51 Enable	E8H	-				INTTB51/INTTB50			
			-	-	-	-	ITB5XC	ITB5XM2	ITB5XM1	ITB5XM0
							R	R/W		
			Note: Write “0”				0			
INTETBOX	INTTBOX (Overflow) Enable	E9H	-				INTTBOX			
			-	-	-	-	ITBOXC	ITBOXM2	ITBOXM1	ITBOXM0
							R	R/W		
			Note: Write “0”				0			

The state of an interrupt request flag

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request.
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 request.

Note 1: The interruption level setting register of combination interruption should clear an interruption demand flag in an INTCLR register, before changing an INTSEL register.
Moreover, re-set an interrupt level as a desired level.

Symbol	NAME	address	7	6	5	4	3	2	1	0
INTEPAD	INTP0 & INTAD Enable	E4H	INTP0				INTAD			
			IP0C	IP0M2	IP0M1	IP0M0	IADC	IADM2	IADM1	IADM0
			R	R/W			R	R/W		
			0				0			
INTETC01	INTTC0 & INTTC1 Enable	F0H	INTTC1(DMA1)				INTTC0(DMA0)			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0				0			
INTETC23	INTTC2 & INTTC3 Enable	F1H	INTTC3(DMA3)				INTTC2(DMA2)			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0				0			
INTETC45	INTTC4 & INTTC5 Enable	F2H	INTTC5(DMA5)				INTTC4(DMA4)			
			ITC5C	ITC5M2	ITC5M1	ITC5M0	ITC4C	ITC4M2	ITC4M1	ITC4M0
			R	R/W			R	R/W		
			0				0			
INTETC67	INTTC6 & INTTC7 Enable	F3H	INTTC7(DMA7)				INTTC6(DMA6)			
			ITC7C	ITC7M2	ITC7M1	ITC7M0	ITC6C	ITC6M2	ITC6M1	ITC6M0
			R	R/W			R	R/W		
			0				0			
INTNMWDT	NMI & INTWDT Enable	EFH	NMI				INTWD			
			INCNM	-	-	-	ITCWD	-	-	-
			R				R			
			0	-	-	-	0	-	-	-

The state of an interrupt request flag

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request.
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 request.

Note 1: It is not set, even if it leads an interrupt request flag at the same time it inputted $\overline{NMI}$.
An interrupt request flag borrows from being set in X1 × 4 cycle.

(2) External interrupt control

Symbol	NAME	address	7	6	5	4	3	2	1	0
IIMC0	Interrupt Input mode Control 0	F6H (Prohibit RMW)								NMIREE
										R/W
										0
										$\overline{NMI}$ 0:Falling 1:Falling and Rising
IIMC1	Interrupt Input mode Control 1	FAH (Prohibit RMW)	I7LE	I6LE	I5LE	I4LE	I3LE	I2LE	I1LE	I0LE
			R/W							
			0	0	0	0	0	0	0	0
			INT7 0:Edge 1:Level	INT6 0:Edge 1:Level	INT5 0:Edge 1:Level	INT4 0:Edge 1:Level	INT3 0:Edge 1:Level	INT2 0:Edge 1:Level	INT1 0:Edge 1:Level	INT0 0:Edge 1:Level
IIMC2	Interrupt Input mode Control 2	FBH (Prohibit RMW)	I7EDGE	I6EDGE	I5EDGE	I4EDGE	I3EDGE	I2EDGE	I1EDGE	I0EDGE
			R/W							
			0	0	0	0	0	0	0	0
			INT7 0:Rising /High 1:Falling /Low	INT6 0:Rising /High 1:Falling /Low	INT5 0:Rising /High 1:Falling /Low	INT4 0:Rising /High 1:Falling /Low	INT3 0:Rising /High 1:Falling /Low	INT2 0:Rising /High 1:Falling /Low	INT1 0:Rising /High 1:Falling /Low	INT0 0:Rising /High 1:Falling /Low
IIMC3	Interrupt Input mode Control 3	10EH (Prohibit RMW)					IBLE	IALE	I9LE	I8LE
							R/W			
							0	0	0	0
							INTB 0:Edge 1:Level	INTA 0:Edge 1:Level	INT9 0:Edge 1:Level	INT8 0:Edge 1:Level
IIMC4	Interrupt Input mode Control 4	10FH (Prohibit RMW)					IBEDGE	IAEDGE	I9EDGE	I8EDGE
							R/W			
							0	0	0	0
							INTB 0:Rising /High 1:Falling /Low	INTA 0:Rising /High 1:Falling /Low	INT9 0:Rising /High 1:Falling /Low	INT8 0:Rising /High 1:Falling /Low

Note 1: Disable INT0 to INTB before changing INT0 to B pins mode from “level” to “edge”.

Setting example for case of INT0:

```
DI
LD      (IIMC2), XXXXXXX0B      ; change from “level” to “edge”.
LD      (INTCLR), 0AH           ; Clear interrupt request flag.
   NOP                          ; Wait EI execution.
NOP
   NOP
   EI
```

X = Don’t care; “-“ = No change.

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

Function Setting of External Interrupt Pin (1/2)

Interrupt Pin	Shared pin	Mode	Setting Method
INT0	PF0	 Rising edge	<I0LE> = 0,<I0EDGE> = 0
		 Falling edge	<I0LE> = 0, <I0EDGE> = 1
		 High level	<I0LE> = 1,<I0EDGE> = 0
		 Low level	<I0LE> = 1,<I0EDGE> = 1
INT1	PF2	 Rising edge	<I1LE> = 0,<I1EDGE> = 0
		 Falling edge	<I1LE> = 0,<I1EDGE> = 1
		 High level	<I1LE> = 1,<I1EDGE> = 0
		 Low level	<I1LE> = 1,<I1EDGE> = 1
INT2	PF4	 Rising edge	<I2LE> = 0,<I2EDGE> = 0
		 Falling edge	<I2LE> = 0,<I2EDGE> = 1
		 High level	<I2LE> = 1,<I2EDGE> = 0
		 Low level	<I2LE> = 1,<I2EDGE> = 1
INT3	PF6	 Rising edge	<I3LE> = 0,<I3EDGE> = 0
		 Falling edge	<I3LE> = 0,<I3EDGE> = 1
		 High level	<I3LE> = 1,<I3EDGE> = 0
		 Low level	<I3LE> = 1,<I3EDGE> = 1
INT4	PK0	 Rising edge	<I4LE> = 0,<I4EDGE> = 0
		 Falling edge	<I4LE> = 0,<I4EDGE> = 1
		 High level	<I4LE> = 1,<I4EDGE> = 0
		 Low level	<I4LE> = 1,<I4EDGE> = 1
INT5	PK1	 Rising edge	<I5LE> = 0,<I5EDGE> = 0
		 Falling edge	<I5LE> = 0,<I5EDGE> = 1
		 High level	<I5LE> = 1,<I5EDGE> = 0
		 Low level	<I5LE> = 1,<I5EDGE> = 1
INT6	PK2	 Rising edge	<I6LE> = 0,<I6EDGE> = 0
		 Falling edge	<I6LE> = 0,<I6EDGE> = 1
		 High level	<I6LE> = 1,<I6EDGE> = 0
		 Low level	<I6LE> = 1,<I6EDGE> = 1
INT7	PK3	 Rising edge	<I7LE> = 0,<I7EDGE> = 0
		 Falling edge	<I7LE> = 0,<I7EDGE> = 1
		 High level	<I7LE> = 1,<I7EDGE> = 0
		 Low level	<I7LE> = 1,<I7EDGE> = 1
INT8	PK4	 Rising edge	<I8LE> = 0,<I8EDGE> = 0
		 Falling edge	<I8LE> = 0,<I8EDGE> = 1
		 High level	<I8LE> = 1,<I8EDGE> = 0
		 Low level	<I8LE> = 1,<I8EDGE> = 1

Function Setting of External Interrupt Pin (2/2)

Interrupt Pin	Shared pin	Mode	Setting Method
INT9	PK5	 Rising edge	<I9LE> = 0,<I9EDGE> = 0
		 Falling edge	<I9LE> = 0,<I9EDGE> = 1
		High level	<I9LE> = 1,<I9EDGE> = 0
		 Low level	<I9LE> = 1,<I9EDGE> = 1
INTA	PK6	 Rising edge	<IALE> = 0,<IAEDGE> = 0
		 Falling edge	<IALE> = 0,<IAEDGE> = 1
		 High level	<IALE> = 1,<IAEDGE> = 0
		 Low level	<IALE> = 1,<IAEDGE> = 1
INTB	PK7	 Rising edge	<IBLE> = 0,<IBEDGE> = 0
		 Fallinf edge	<IBLE> = 0,<IBEDGE> = 1
		 High level	<IBLE> = 1,<IBEDGE> = 0
		 Low level	<IBLE> = 1,<IBEDGE> = 1

(3) Interrupt control

Symbol	NAME	address	7	6	5	4	3	2	1	0
INTSEL	Interruption combination selection	10CH (Prohibit RMW)		DP49SEL	DP48SEL	DP47SEL	DP39SEL	DP37SEL	DP26SEL	DP24SEL
				R/W						
				0	0	0	0	0	0	0
				0:INTTB50 Interruption is effective 1:INTTB51 Interruption is effective	0:INTTB40 Interruption is effective 1:INTTB41 Interruption is effective	0:INTTB30 Interruption is effective 1:INTTB31 Interruption is effective	0:INTB Interruption is invalid 1:INTB Interruption is effective	0:INTA Interruption is invalid 1:INTA Interruption is effective	0:INTTA7 Interruption is effective 1:INT9 Interruption is effective	0:INTTA5 Interruption is effective 1:INT8 Interruption is effective
INTST	Interruption generating flag	10DH (Prohibit RMW)			TBOF5ST	TBOF4ST	TBOF3ST	TBOF2ST	TBOF1ST	TBOF0ST
					R/W					
					0	0	0	0	0	0
					Read: 0:Interruption un-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption un-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption un-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption un-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption un-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption un-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care
SIMC	SIO Interrupt control	F5H (Prohibit RMW)	-				IR3LE	IR2LE	IR1LE	IR0LE
			W				R/W			
			0				1	1	1	1
			Note: Write "1"				0:INTRX3 edge mode 1:INTRX3 level mode	0:INTRX2 edge mode 1:INTRX2 level mode	0:INTRX1 edge mode 1:INTRX1 level mode	0:INTRX0 edge mode 1:INTRX0 level mode

Note 1: The default priorities 24, 26, 47 to 49 are making the interruption factor serve a double purpose. It is necessary to choose the interruption factor used in an interruption factor selection register. Therefore, interruption cannot be used simultaneously.

Note 2: The default priority 50 is making the interruption factor serve a double purpose. The interruption factor assigned to this default priority 50 can be used simultaneously. Of which interruption factor interruption occurred should interrupt, and please check it in a generating flag register.

Note 3: The interruption level setting register of combination interruption should clear an interruption demand flag in an INTCLR register, before changing an INTSEL register.
Moreover, re-set an interrupt level as a desired level.

(4) Interrupt request flag clear register

The interrupt request flag is cleared by writing the appropriate micro DMA start vector.

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 (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			0	0	0	0	0	0	0	0
			Interrupt vector							

(5) Micro DMA start vector registers

This register assigns micro DMA processing to which interrupt source. The interrupt source with a micro DMA start vector that matches the vector set in this register is assigned as the micro DMA start source.

When the micro DMA transfer counter value reaches “0”, 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, to continue micro DMA processing, set the micro DMA start vector register again during the 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 channel with the lowest number has a higher priority. Accordingly, if the same vector is set in the micro DMA start vector registers of two channels, the interrupt generated in the channel with the lower number is executed until micro DMA transfer is completed. If the micro DMA start vector for this channel is not set again, the next micro DMA is started for the channel with the higher number (Micro DMA chaining).

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

(6) 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 0. Setting any of the bits in the register DMAB which correspond to a micro DMA channel (as shown below) to 1 specifies that any micro DMA transfer on that channel will be a burst transfer.

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

(7) 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 (Note), 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 placed after a "DI" instruction. And in the case of setting an interrupt enable again by "EI" instruction after the execution of clearing instruction, execute "EI" instruction after clearing and more than 3-instructions (e.g., "NOP"× 1 times).

If placed "EI" instruction without waiting "NOP" instruction after execution of clearing instruction, interrupt will be enable before request flag is cleared. Thus, when be changed interrupt request level to "0", change it after cleared corresponding interrupt request by INTCLR instruction.

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

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

INT0 to INTB level mode	In level mode INT0 to INTB are not an edge-triggered interrupt. Hence, in level mode the interrupt request flip-flop for INT0 to INTB 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 INT x (x □0 to 7) going from "0" to "1", INTx must then be held at "1" until the interrupt response sequence has been completed. If INTx is set to Level mode so as to release a Halt state, INTx must be held at "1" from the time INTx 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 INTx to revert to "0" before the Halt state has been released.)
	When the mode changes from level mode to edge mode, interrupt request flags which were set in level mode will not be cleared. Interrupt request flags must be cleared using the following sequence.
	DI
	LD (IIMC2),00H ; Changes from level to edge.
	LD (INTCLR),0AH ; Clears interrupt request flag.
	NOP ; Wait EI execution.
	NOP
	NOP
	EI

INTRX0 to INTRX3	The interrupt request flip-flop can only be cleared by a reset or by reading the serial channel receive buffer. It cannot be cleared by an instruction.

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

INT0 to INT 7: Instructions which switch to level mode after an interrupt request has been generated in edge mode.

The pin input change from high to low after interrupt request has been generated in level mode. ("H" → "L", "L" →"H")

INTRX0 to INTRX2: Instruction which read the receive buffer.

(8) About combination of an interruption factor

About the following interruption factor, interruption is made to serve a double purpose. Cautions are needed when using it.

1)INT8/INTTA5

The interruption table / interruption level setting register is made to serve a double purpose.Therefore, it cannot be used simultaneously. Micro DMA starting is also that only which or one of the two can be used. In order to change the interruption factor to be used, it is necessary to set up INTSEL<DP24SEL>. It disappears, even if interruption of INTTA5(8-bit timer 5) will occur, if INTSEL<DP24SEL> is set as "1." It disappears, even if interruption of INT8(INT8 terminal input) will occur, if INTSEL<DP24SEL> is set as "0."

2)INT9/INTTA7

The interruption table / interruption level setting register is made to serve a double purpose.Therefore, it cannot be used simultaneously.Micro DMA starting is also that only which or one of the two can be used. In order to change the interruption factor to be used, it is necessary to set up INTSEL<DP26SEL>.It disappears, even if interruption of INTTA7(8-bit timer 7) will occur, if INTSEL<DP26SEL> is set as "1." It disappears, even if interruption of INT9(INT9 terminal input) will occur, if INTSEL<DP26SEL> is set as "0."

3)INTTB31/INTTB30

The interruption table / interruption level setting register is made to serve a double purpose.Therefore, it cannot be used simultaneously.Micro DMA starting is also that only which or one of the two can be used. In order to change the interruption factor to be used, it is necessary to set up INTSEL<DP47SEL>.It disappears, even if interruption of INTTB30(16-bit timer 3) will occur, if INTSEL<DP47SEL> is set as "1." It disappears, even if interruption of INTTB31(16-bit timer 3) will occur, if INTSEL<DP47SEL> is set as "0."

4)INTTB41/INTTB40

The interruption table / interruption level setting register is made to serve a double purpose.Therefore, it cannot be used simultaneously.Micro DMA starting is also that only which or one of the two can be used. In order to change the interruption factor to be used, it is necessary to set up INTSEL<DP48SEL>.It disappears, even if interruption of INTTB40(16-bit timer 4) will occur, if INTSEL<DP48SEL> is set as "1." It disappears, even if interruption of INTTB41(16-bit timer 4) will occur, if INTSEL<DP48SEL> is set as "0."

5)INTTB51/INTTB50

The interruption table / interruption level setting register is made to serve a double purpose.Therefore, it cannot be used simultaneously.Micro DMA starting is also that only which or one of the two can be used. In order to change the interruption factor to be used, it is necessary to set up INTSEL<DP49SEL>.It disappears, even if interruption of INTTB50(16-bit timer 5) will occur, if INTSEL<DP49SEL> is set as "1." It disappears, even if interruption of INTTB51(16-bit timer 5) will occur, if INTSEL<DP49SEL> is set as "0."

When you change an interruption factor, please change in the following procedures.

It interrupts, an interruption level setting register is set as the ban on a demand, and an interruption demand flag is cleared.It is set as the interruption factor which uses an interruption combination selection register. An interrupt level is set as an interrupt level setting register.

3.5 Function Ports

TMP92CM27 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. list I/O registers and their specifications.

Table 3.5.1 Port Function(R: PD = with programmable pull-down register, U = with pull-up register) (1/2)

Port Name	Pin Name	Number of Pins	I/O	R	I/O Setting	Pin Name for built-in function
Port 1	P10 to P17	8	I/O	–	Bit	D8 to D15
Port 6	P60 to P67	8	I/O	–	Bit	A16 to A23
Port 7	P71	1	I/O	U	Bit	$\overline{WRLL}$
	P72	1	I/O	U	Bit	$\overline{WRLU}$
	P73	1	I/O	–	Bit	R/$\overline{W}$
	P74	1	I/O	U	Bit	$\overline{SRWR}$
	P75	1	I/O	U	Bit	$\overline{SRLLB}$
	P76	1	I/O	U	Bit	$\overline{SRLUB}$
	P77	1	I/O	-	Bit	$\overline{WAIT}$
Port 8	P80	1	Output	–	(Fixed)	$\overline{CS0}$
	P81	1	Output	–	(Fixed)	$\overline{CS1}$
	P82	1	Output	–	(Fixed)	$\overline{CS2}$
	P83	1	Output	–	(Fixed)	$\overline{CS3}$ / $\overline{SDCS}$
	P84	1	Output	–	(Fixed)	$\overline{CS4}$
	P85	1	Output	–	(Fixed)	$\overline{CS5}$ / $\overline{WDTOUT}$
	P86	1	I/O	–	Bit	$\overline{BUSRQ}$
	P87	1	I/O	–	Bit	$\overline{BUSAK}$
Port 9	P90	1	Output	–	(Fixed)	$\overline{SDWE}$
	P91	1	Output	–	(Fixed)	$\overline{SDRAS}$
	P92	1	Output	–	(Fixed)	$\overline{SDCAS}$
	P93	1	Output	–	(Fixed)	SDLLDQM
	P94	1	Output	–	(Fixed)	SDLUDQM
	P95	1	Output	–	(Fixed)	SDCKE
	P96	1	Output	–	(Fixed)	SDCLK
Port A	PA0	1	I/O	–	Bit	RXD0
	PA1	1	I/O	–	Bit	TXD0
	PA2	1	I/O	–	Bit	SCLK0/ $\overline{CTS0}$
	PA3	1	I/O	–	Bit	RXD1
	PA4	1	I/O	–	Bit	TXD1
	PA5	1	I/O	–	Bit	SCLK1/ $\overline{CTS1}$
Port C	PC0	1	I/O	–	Bit	SO0/SDA0
	PC1	1	I/O	–	Bit	SI0/SCL0
	PC2	1	I/O	–	Bit	SCK0
	PC3	1	I/O	–	Bit	SO1/SDA1
	PC4	1	I/O	–	Bit	SI1/SCL1
	PC5	1	I/O	–	Bit	SCK1

Table 3.5.1 Port Function(R: PD = with programmable pull-down register, U = with pull-up register) (2/2)

Port Name	Pin Name	Number of Pins	I/O	R	I/O Setting	Pin Name for built-in function
Port D	PD0	1	I/O	–	Bit	HSSI0
	PD1	1	I/O	–	Bit	HSSO0
	PD2	1	I/O	–	Bit	HSCLK0
	PD3	1	I/O	–	Bit	RXD2
	PD4	1	I/O	–	Bit	TXD2
	PD5	1	I/O	–	Bit	SCLK2/ $\overline{\text{CTS2}}$
Port F	PF0	1	I/O	–	Bit	TA0IN/INT0
	PF1	1	I/O	–	Bit	TA1OUT
	PF2	1	I/O	–	Bit	TA2IN/INT1
	PF3	1	I/O	–	Bit	TA3OUT
	PF4	1	I/O	–	Bit	TA4IN/INT2
	PF5	1	I/O	–	Bit	TA5OUT
	PF6	1	I/O	–	Bit	TA6IN/INT3
Port J	PJ0	1	I/O	–	Bit	TB0OUT0
	PJ1	1	I/O	–	Bit	TB0OUT1
	PJ2	1	I/O	–	Bit	TB1OUT0
	PJ3	1	I/O	–	Bit	TB1OUT1
	PJ4	1	I/O	–	Bit	TB2OUT0/TB4OUT0
	PJ5	1	I/O	–	Bit	TB2OUT1/TB4OUT1
	PJ6	1	I/O	–	Bit	TB3OUT0/TB5OUT0
	PJ7	1	I/O	–	Bit	TB3OUT1/TB5OUT1
Port K	PK0	1	Input	–	(Fixed)	TB0IN0/INT4
	PK1	1	Input	–	(Fixed)	TB0IN1/INT5
	PK2	1	Input	–	(Fixed)	TB1IN0/INT6
	PK3	1	Input	–	(Fixed)	TB1IN1/INT7
	PK4	1	Input	–	(Fixed)	TB2IN0/INT8
	PK5	1	Input	–	(Fixed)	TB2IN1/INT9
	PK6	1	Input	–	(Fixed)	TB3IN0/INTA
	PK7	1	Input	–	(Fixed)	TB3IN1/INTB
Port L	PL0	1	I/O	–	Bit	PG00/RXD3
	PL1	1	I/O	–	Bit	PG01/TXD3
	PL2	1	I/O	–	Bit	PG02/SCLK3/ $\overline{\text{CTS3}}$
	PL3	1	I/O	–	Bit	PG03/TA7OUT
	PL4	1	I/O	–	Bit	PG10/HSSI1
	PL5	1	I/O	–	Bit	PG11/HSSO1
	PL6	1	I/O	–	Bit	PG12/HSCLK1
	PL7	1	I/O	–	Bit	PG13
Port M	PM0 to PM7	8	Input	–	(Fixed)	AN0 to AN7/KI0 to KI7
Port N	PN0 to PN2	3	Input	–	(Fixed)	AN8 to AN10
	PN3	1	Input	–	(Fixed)	AN11/ $\overline{\text{ADTRG}}$

Table 3.5.2 I/O Port and Specifications (1/7) X: Don't care

Port	Pin name	Specification	I/O register			
			Pn	PnCR	PnFC	PnFC2
Port 1	P10 to P17	Input Port	X	0	0	None
		Output Port	X	1		
		D8 to D15 bus	X	X	1	
Port 6	P60 to P67	Input Port	X	0	0	None
		Output Port	X	1		
		A16 to A23 output	X	X	1	
Port 7	P71	Input Port (without pull up)	0	0	0	None
		Input Port (with pull up)	1	0	0	
		Output Port	X	1	0	
		$\overline{WRLL}$	X	1	1	
	P72	Input Port (without pull up)	0	0	0	
		Input Port (with pull up)	1	0	0	
		Output Port	X	1	0	
		$\overline{WRLU}$	X	1	1	
	P73	Input Port	X	0	0	
		Output Port	X	1	0	
		R/$\overline{W}$	X	1	1	
	P74	Input Port (without pull up)	0	0	0	
		Input Port (with pull up)	1	0	0	
		Output Port	X	1	0	
		$\overline{SRWR}$	X	1	1	
	P75	Input Port (without pull up)	0	0	0	
		Input Port (with pull up)	1	0	0	
		Output Port	X	1	0	
		$\overline{SRLLB}$	X	1	1	
	P76	Input Port (without pull up)	0	0	0	
		Input Port (with pull up)	1	0	0	
		Output Port	X	1	0	
		$\overline{SRUB}$	X	1	1	
	P77	Input Port	X	0	0	
		Output Port	X	1	0	
		$\overline{WAIT}$	X	0	1	

Table 3.5.2 I/O Port and Specification (2/7) X: Don't care

Port	Pin name	Specification	I/O register			
			Pn	PnCR	PnFC	PnFC2
Port 8	P80	Output Port	X	None	0	None
		$\overline{CS0}$ output	X		1	
	P81	Output Port	X		0	
		$\overline{CS1}$ output	X		1	
	P82	Output Port	X		0	
		$\overline{CS2}$ output	X		1	
	P83	出力ポート	X		0	0
		$\overline{CS3}$ output	X		1	0
		$\overline{SDCS}$ output	X		1	1
		Reserved	X		0	1
	P84	Output Port	X		0	None
		$\overline{CS4}$ output	X		1	
	P85	Output Port	X		0	0
		$\overline{CS5}$ output	X		1	0
		$\overline{WDTOUT}$ output	X		1	1
		Reserved	X		0	1
	P86 to P87	Input Port	X	0	0	None
		Output Port	X	1	0	
	P86	$\overline{BUSRQ}$	X	0	1	
		Reserved	X	1	1	
	P87	$\overline{BUSAK}$	X	1	1	
		Reserved	X	0	1	
Port 9	P90 to P96	Output Port	X	None	0	None
	P90	$\overline{SDWE}$	X		1	
	P91	$\overline{SDRAS}$	X		1	
	P92	$\overline{SDCAS}$	X		1	
	P93	SDLLDQM	X		1	
	P94	SDLUDQM	X		1	
	P95	SDCKE	X		1	
	P96	SDCLK	X		1	

Table 3.5.2 I/O Port and Specifications (3/7) X: Don't care

Port	Pin name	Specification	I/O register			
			Pn	PnCR	PnFC	PnFC2
Port A	PA0	Input Port	X	0	0	None
		Output Port	X	1	0	
		RXD0 input	X	0	1	
	PA1	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TXD0 output	X	1	1	0
		TXD0 (open drain) output	X	1	1	1
	PA2	Input Port	X	0	0	None
		Output Port	X	1	0	
		SCLK0/ $\overline{\text{CTS0}}$ input	X	0	1	
		SCLK0 output	X	1	1	
	PA3	Input Port	X	0	0	None
		Output Port	X	1	0	
		RXD1 input	X	0	1	
	PA4	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TXD1 output	X	1	1	0
		TXD1 (open drain) output	X	1	1	1
	PA5	Input Port	X	0	0	None
		Output Port	X	1	0	
		SCLK1/ $\overline{\text{CTS1}}$ input	X	0	1	
		SCLK1 output	X	1	1	
Port C	PC0	Input Port	X	0	0	0
		Output Port	X	1	0	0
		SO0 output	X	0	1	0
		SDA0 I/O	X	1	1	0
		SO0 (open drain) output	X	0	1	1
		SDA0 (open drain) I/O	X	1	1	1
	PC1	Input Port	X	0	0	0
		Output Port	X	1	0	0
		SI0 input	X	0	1	0
		SCL0 I/O	X	0	1	1
		SCL0 (open drain) I/O	X	1	1	1
	PC2	Input Port	X	0	0	None
		Output Port	X	1	0	
		SCK0 input	X	0	1	
		SCK0 output	X	1	1	

Table 3.5.2 I/O Port and Specifications (4/7) X: Don't care

Port	Pin name	Specification	I/O register			
			Pn	PnCR	PnFC	PnFC2
Port C	PC3	Input Port	X	0	0	0
		Output Port	X	1	0	0
		SO1 output	X	0	1	0
		SDA1 I/O	X	1	1	0
		SO1 (open drain) output	X	0	1	1
		SDA1 (open drain) I/O	X	1	1	1
	PC4	Input Port	X	0	0	0
		Output Port	X	1	0	0
		SI1 input	X	0	1	0
		SCL1 I/O	X	0	1	1
		SCL1 (open drain) I/O	X	1	1	1
	PC5	Input Port	X	0	0	None
		Output Port	X	1	0	
		SCK1 input	X	0	1	
		SCK1 output	X	1	1	
Port D	PD0	Input Port	X	0	0	None
		Output Port	X	1	0	
		HSSI0 input	X	0	1	
	PD1	Input Port	X	0	0	None
		Output Port	X	1	0	
		HSSO0 output	X	1	1	
	PD2	Input Port	X	0	0	None
		Output Port	X	1	0	
		HSCLK0 output	X	1	1	
	PD3	Input Port	X	0	0	None
		Output Port	X	1	0	
		RXD2 input	X	0	1	
	PD4	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TXD2 output	X	1	1	0
		TXD2 (open drain) output	X	1	1	1
	PD5	Input Port	X	0	0	None
		Output Port	X	1	0	
		SCLK2/ $\overline{\text{CTS2}}$ input	X	0	1	
		SCLK2 output	X	1	1	

Table 3.5.2 I/O Port and Specifications (5/7) X: Don't care

Port	Pin name	Specification	I/O register			
			Pn	PnCR	PnFC	PnFC2
Port F	PF0	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TA0IN input	X	0	1	0
		INT0 input	X	0	1	1
	PF1	Input Port	X	0	0	None
		Output Port	X	1	0	
		TA1OUT output	X	1	1	
	PF2	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TA2IN input	X	0	1	0
		INT1 input	X	0	1	1
	PF3	Input Port	X	0	0	None
		Output Port	X	1	0	
		TA3OUT output	X	1	1	
	PF4	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TA4IN input	X	0	1	0
		INT2 input	X	0	1	1
	PF5	Input Port	X	0	0	None
		Output Port	X	1	0	
		TA5OUT output	X	1	1	
	PF6	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TA6IN input	X	0	1	0
		INT3 input	X	0	1	1
Port J	PJ0	Input Port	X	0	0	None
		Output Port	X	1	0	
		TB0OUT0 output	X	1	1	
	PJ1	Input Port	X	0	0	
		Output Port	X	1	0	
		TB0OUT1 output	X	1	1	
	PJ2	Input Port	X	0	0	
		Output Port	X	1	0	
		TB1OUT0 output	X	1	1	
	PJ3	Input Port	X	0	0	
		Output Port	X	1	0	
		TB1OUT0 output	X	1	1	

Table 3.5.2 I/O Port and Specifications (6/7) X: Don't care

Port	Pin name	Specification	I/O register			
			Pn	PnCR	PnFC	PnFC2
Port J	PJ4	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TB2OUT0 output	X	1	1	0
		TB4OUT0 output	X	1	1	1
	PJ5	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TB2OUT1 output	X	1	1	0
		TB4OUT1 output	X	1	1	1
	PJ6	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TB3OUT0 output	X	1	1	0
		TB5OUT0 output	X	1	1	1
	PJ7	Input Port	X	0	0	0
		Output Port	X	1	0	0
		TB3OUT1 output	X	1	1	0
		TB5OUT1 output	X	1	1	1
Port K	PK0	Input Port	X	None	0	0
		TB0IN0 input	X		1	0
		INT4 input	X		1	1
	PK1	Input Port	X		0	0
		TB0IN1 input	X		1	0
		INT5 input	X		1	1
	PK2	Input Port	X		0	0
		TB1IN0 input	X		1	0
		INT6 input	X		1	1
	PK3	Input Port	X		0	0
		TB1IN1 input	X		1	0
		INT7 input	X		1	1
	PK4	Input Port	X		0	0
		TB2IN0 input	X		1	0
		INT8 input	X		1	1
	PK5	Input Port	X		0	0
		TB2IN1 input	X		1	0
		INT9 input	X		1	1
	PK6	Input Port	X		0	0
		TB3IN0 input	X		1	0
		INTA input	X		1	1
	PK7	Input Port	X		0	0
		TB3IN1 input	X		1	0
		INTB input	X		1	1

Table 3.5.2 I/O Port and Specifications (7/7) X: Don't care

Port	Pin name	Specification	I/O register			
			Pn	PnCR	PnFC	PnFC2
Port L	PL0	Input Port	X	0	0	0
		Output Port	X	1	0	0
		PG00 output	X	1	1	0
		RXD3 input	X	0	1	0
	PL1	Input Port	X	0	0	0
		Output Port	X	1	0	0
		PG01 output	X	1	1	0
		TXD3 output	X	1	0	1
		TXD3 (open drain) output	X	1	1	1
	PL2	Input Port	X	0	0	0
		Output Port	X	1	0	0
		PG02 output	X	1	1	0
		SCLK3/ $\overline{\text{CTS3}}$ input	X	0	0	1
		SCLK3 output	X	1	0	1
	PL3	Input Port	X	0	0	0
		Output Port	X	1	0	0
		PG03 output	X	1	1	0
		TA7OUT	X	1	1	1
	PL4	Input Port	X	0	0	0
		Output Port	X	1	0	0
		PG10 output	X	1	1	0
		HSSI1 input	X	0	0	1
	PL5	Input Port	X	0	0	0
		Output Port	X	1	0	0
		PG11 output	X	1	1	0
		HSSO1 output	X	1	0	1
	PL6	Input Port	X	0	0	0
		Output Port	X	1	0	0
		PG12 output	X	1	1	0
		HSCLK1 output	X	1	0	1
	PL7	Input Port	X	0	0	0
		Output Port	X	1	0	0
		PG13 output	X	1	1	0
Port M	PM0 to PM7	Input Port/KEY IN input	X	None	0	None
		AN0 to AN7 input	X		1	
Port N	PN0 to PN2	Input Port	X	None	0	None
		AN8 to AN10 input	X		1	
	PN3	Input Port/ $\overline{\text{ADTRG}}$	X		0	
		AN11 input	X		1	

Input buffer state table (1/3)

Port name	Input Function name	Input buffer state											
		Reset state	CPU Operation state		HALT state							
					IDLE2		IDLE1		STOP		STOP	
									<DRVE> = 1		<DRVE> =0	
			At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup
D0 to D7	D0 to D7	OFF	ON by external read.	–	OFF	–	OFF	–	OFF	–	OFF	–
P10 to P17	D8 to D15	OFF		ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
P60 to P67	A16 to A23	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
P71 to P72 P74 to P76 (*1)	–	ON	–	ON	–	OFF	–	OFF	–	OFF	–	OFF
P73	–	ON	–	ON	–	OFF	–	OFF	–	OFF	–	OFF
P77	$\overline{WAIT}$	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
P80 to P85	–					–						
P86	$\overline{BUSRQ}$	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
P87	–	ON	–	ON	–	OFF	–	OFF	–	OFF	–	OFF
P90 to P96	–		Controls by P9DR.									
PA0	RXD0	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PA1	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PA2	SCLK0/ $\overline{CTS0}$	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PA3	RXD1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PA4	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PA5	SCLK1/ $\overline{CTS1}$	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PC0	SDA0	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PC1	SI0/SCL0	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PC2	SCK0	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PC3	SDA1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PC4	SI1/SCL1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PC5	SCK1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PD0	HSSI0	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PD1 to PD2	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PD3	RXD2	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PD4	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PD5	SCLK2/ $\overline{CTS2}$	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF

Input buffer state table (2/3)

Port name	Input Function name	Input buffer state										
		Reset state	CPU Operation state		HALT state							
					IDLE2		IDLE1		STOP		STOP	
									<DRVE> = 1		<DRVE> =0	
			At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup
PF0	TA0IN	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT0										ON	
PF1	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PF2	TA2IN	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT1										ON	
PF3	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PF4	TA4IN	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT2										ON	
PF5	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PF6	TA6IN	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT3										ON	
PJ0 to PJ7	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PK0	TB0IN0	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT4										ON	
PK1	TB0IN1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT4										ON	
PK2	TB1IN0	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT4										ON	
PK3	TB1IN1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT4										ON	
PK4	TB2IN0	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT4										ON	
PK5	TB2IN1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT4										ON	
PK6	TB3IN0	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT4										ON	
PK7	TB3IN1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
	INT4										ON	
PL0	RXD3	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PL1	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PL2	SCLK2/ $\overline{CTS2}$	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PL3	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PL4	HSSI1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	OFF
PL5 to PL7	–	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Input buffer state table (3/3)

Port name	Input Function name	Input buffer state											
		Reset state	CPU Operation state		HALT state							
					IDLE2		IDLE1		STOP		STOP	
									<DRVE> = 1		<DRVE> =0	
			At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup
PM0 to PM7	AN0 to AN7	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
	KEY0 to KEY7		ON		ON		ON		ON		ON	
PN0 to PN3	AN8 to AN11	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
PN3	$\overline{\text{ADTRG}}$	–	ON		ON		ON		ON		ON	

ON : The buffer is always turned on. A current flows the input buffer if the input pin is not driven.
OFF : The buffer is always turned off.
- : No applicable

*1 : Port having a pull-up/pull-down resistor.
*2 : AIN input does not cause a current to flow through the buffer.
*3 : It becomes an input port after reset and an input buffer turns on during reset at AM 0= 0 and AM1= 1.

Output buffer state table (1/3)

Port name	Output Function name	Output buffer state											
		Reset state	CPU Operation state		HALT state							
					IDLE2		IDLE1		STOP <DRVE> = 1		STOP <DRVE> =0	
			At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup
D0 to D7	D0 to D7	OFF	ON by external write.	–	ON	–	OFF	–	OFF	–	OFF	–
P10 to P17	D8 to D15	OFF		ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
P60 to P67	A16 to A23	ON	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
P71	$\overline{WRLL}$	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
P72	$\overline{WRLU}$											
P73	R/$\overline{W}$											
P74	$\overline{SRWR}$											
P75	$\overline{SRLLB}$											
P76	$\overline{SRLUB}$											
P77	–	OFF	–	ON	–	ON	–	ON	–	ON	–	OFF
P80	$\overline{CS0}$	ON	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
P81	$\overline{CS1}$	ON										
P82	$\overline{CS2}$	ON										
P83	$\overline{CS3}$ / $\overline{SDCS}$	ON										
P84	$\overline{CS4}$	ON										
P85	$\overline{CS5}$ / $\overline{WDTOUT}$	ON										
P86	–	OFF	–	ON	–	ON	–	ON	–	ON	–	OFF
P87	$\overline{BUSAK}$	OFF	ON	ON	ON	ON	ON	ON	On	ON	OFF	OFF
P90	$\overline{SDWE}$	ON	ON	ON	<PXDR>=1:ON <PXDR>=0:OFF							
P91	$\overline{SDRAS}$											
P92	$\overline{SDCAS}$											
P93	SDLLDQM											
P94	SDLUDQM											
P95	SDCKE											
P96	SDCLK											
PA0	–	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
PA1	TXD0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PA2	SCLK0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PA3	–	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
PA4	TXD1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PA5	SCLK1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF

Output buffer state table (2/3)

Port name	Output Function name	Output buffer state										
		Reset state	CPU Operation state		HALT state							
					IDLE2		IDLE1		STOP <DRVE> = 1		STOP <DRVE> =0	
			At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup
PC0	SO0/SDA0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PC1	SCL0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PC2	SCK0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PC3	SO1/SDA1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PC4	SCL1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PC5	SCK1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PD0	−	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
PD1	HSSO0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PD2	HSCLK	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PD3	−	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
PD4	TXD2	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PD5	SCLK2	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PF0	−	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
PF1	TA1OUT	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PF2	−	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
PF3	TA3OUT	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PF4	−	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
PF5	TA5OUT	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PF6	−	OFF	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF
PJ0	TB0OUT0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PJ1	TB0OUT1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PJ2	TB1OUT0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PJ3	TB1OUT1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PJ4	TB2OUT0/ TB4OUT0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PJ5	TB2OUT1/ TB4OUT1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PJ6	TB3OUT0/ TB5OUT0	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PJ7	TB3OUT1/ TB5OUT1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PK0 to PK7	−	−										

Output buffer state table (3/3)

Port name	Output Function name	Output buffer state										
		Reset state	CPU Operation state		HALT state							
					IDLE2		IDLE1		STOP		STOP	
									<DRVE> = 1		<DRVE> =0	
			At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup	At function setup	At input port setup
PL0	PG00	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PL1	PG01/ TXD3	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PL2	PG02/ SCLK3	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PL3	PG03/ TA7OUT	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PL4	PG10	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PL5	PG11/ HSSO1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PL6	PG12/ HSCLK1	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PL7	PG13	OFF	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
PM0 to PM7	−	−										
PN0 to PN3	−	−										

ON : The buffer is always turned on.
However, the output buffer of a specific terminal turns OFF at the time of bus release.
OFF : The buffer is always turned off.
- : No applicable
*1 : Port having a pull-up/pull-down resistor.

3.5.1 Port 1 (P10 to P17)

Port1 is 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P1CR and function register P1FC. In addition to functioning as a general-purpose I/O port, port1 can also function as a data bus (D8 to D15). Moreover, with the combination of AM1 and AM0 shown below, Port1 is set as the following function after reset release.

AM1	AM0	Function Setting after Reset is Released
0	0	Don't use this setting
0	1	Data bus (D8 to D15)
1	0	Input port (P10 to P17)
1	1	Don't use this setting

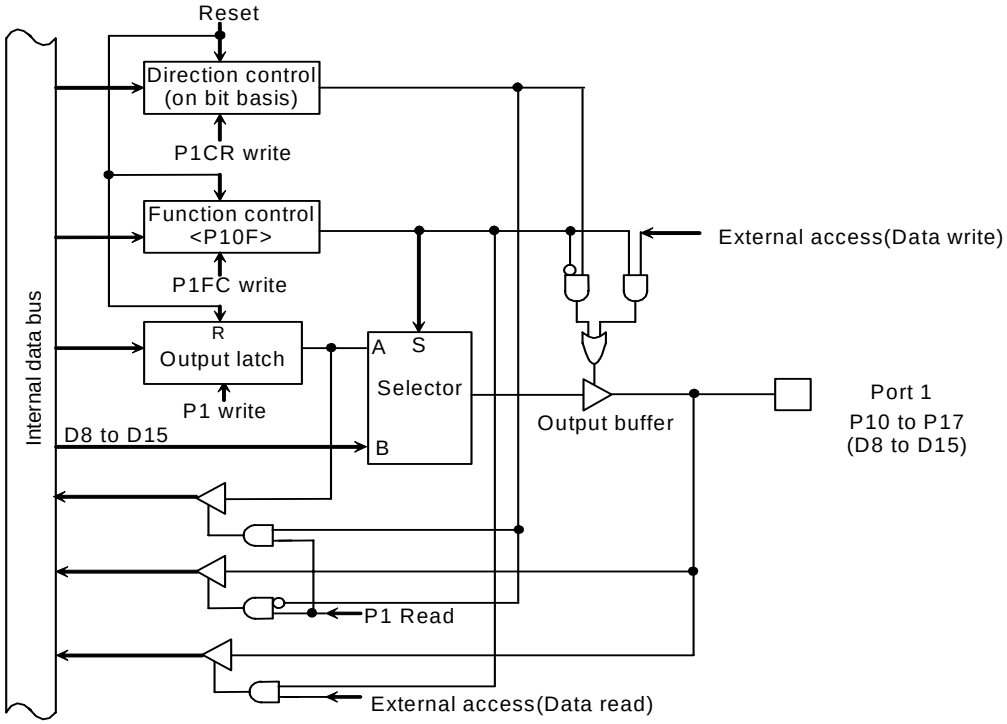


Figure 3.5.1 Port 1

Port 1 register

P1 (0004H)	7	6	5	4	3	2	1	0
bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R/W							
After reset	Data from external port(Output latch register is cleared to "0")							

Port 1 Control register

P1CR (0006H)	7	6	5	4	3	2	1	0
bit Symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port 1 Function register

P1FC (0007H)	7	6	5	4	3	2	1	0
bit Symbol								P1F
Read/Write	W							
After reset								0/1*
Function	Refer to following table							

Port 1 function setting

P1CR<P1xC> \ P1FC<P1F>	0	1
0	Input port	Data bus (D15 to D8)
1	Output port	Reserved

Note 1) Read-modify-write is prohibited for P1CR,P1FC.

Note 2) <P1xC> is bit X of P1CR register.

Note 3) At AM1=0 and AM0=1, it is after reset P1F="1".

At AM1=1 and AM0=0, it is after reset P1F="0".

Figure 3.5.2 Port 1 register

3.5.2 Port 6 (P60 to P67)

Port6 is 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P6CR and function register P6FC. In addition to functioning as a general-purpose I/O port, port6 can also function as an address bus (A16 to A23). Moreover, with the combination of AM1 and AM0 shown below, Port6 is set as the following function after reset release.

AM1	AM0	Function Setting after Reset is Released
0	0	Don't use this setting
0	1	Address bus(A16 to A23)
1	0	Address bus(A16 to A23)
1	1	Don't use this setting

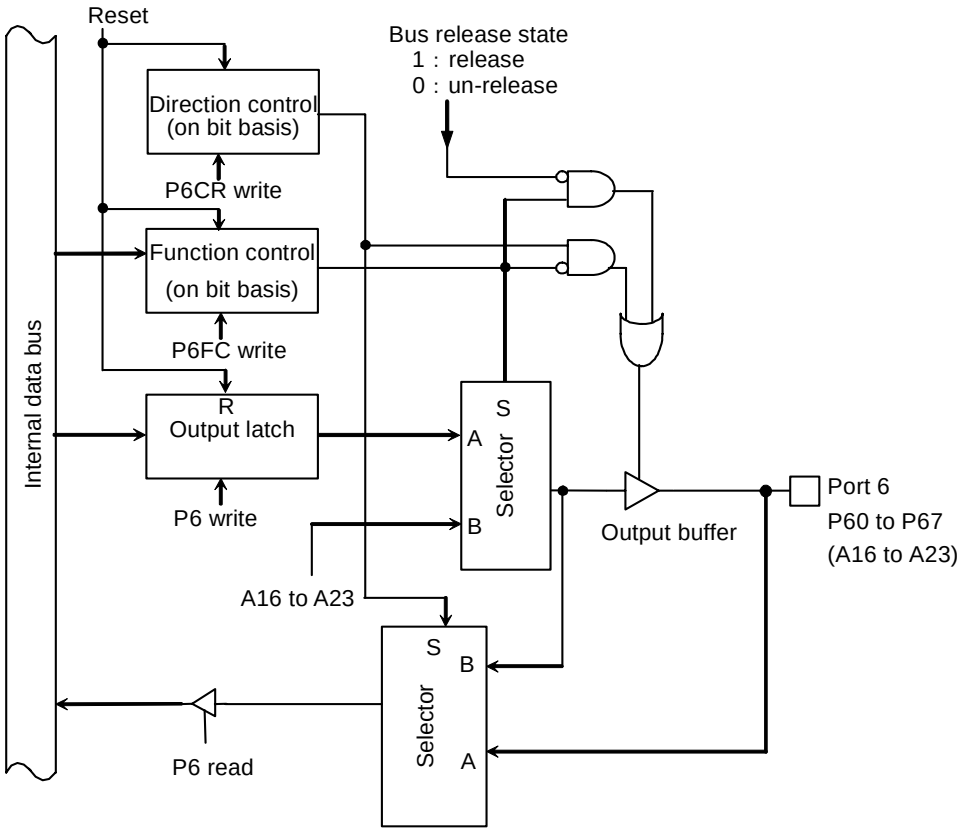


Figure 3.5.3 Port 6

Port 6 register

		7	6	5	4	3	2	1	0
P6 (0018H)	bit Symbol	P67	P66	P65	P64	P63	P62	P61	P60
	Read/Write	R/W							
	After reset	Data from external port(Output latch register is cleared to ”0”)							

Port 6 Control register

		7	6	5	4	3	2	1	0
P6CR (001AH)	bit Symbol	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:Input 1:Output							

Port 6 Function register

		7	6	5	4	3	2	1	0
P6FC (001BH)	bit Symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
	Read/Write	W							
	After reset	1	1	1	1	1	1	1	1
	Function	0:Port 1:Address bus(A16 to A23)							

Figure 3.5.4 Port 6 register

Note) Read-modify-write is prohibited for P6CR,P6FC.

3.5.3 Port 7 (P71 to P77)

Port 71 to P77 is a 7-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P7CR and function register P7FC. Moreover, P71, P72 and P74 to P76 are ports with pull-up resistance. There is an external memory interface function in addition to a general-purpose I/O port function. P71 to P77 become input mode after reset.

(1)P71,P72,P74,P75,P76

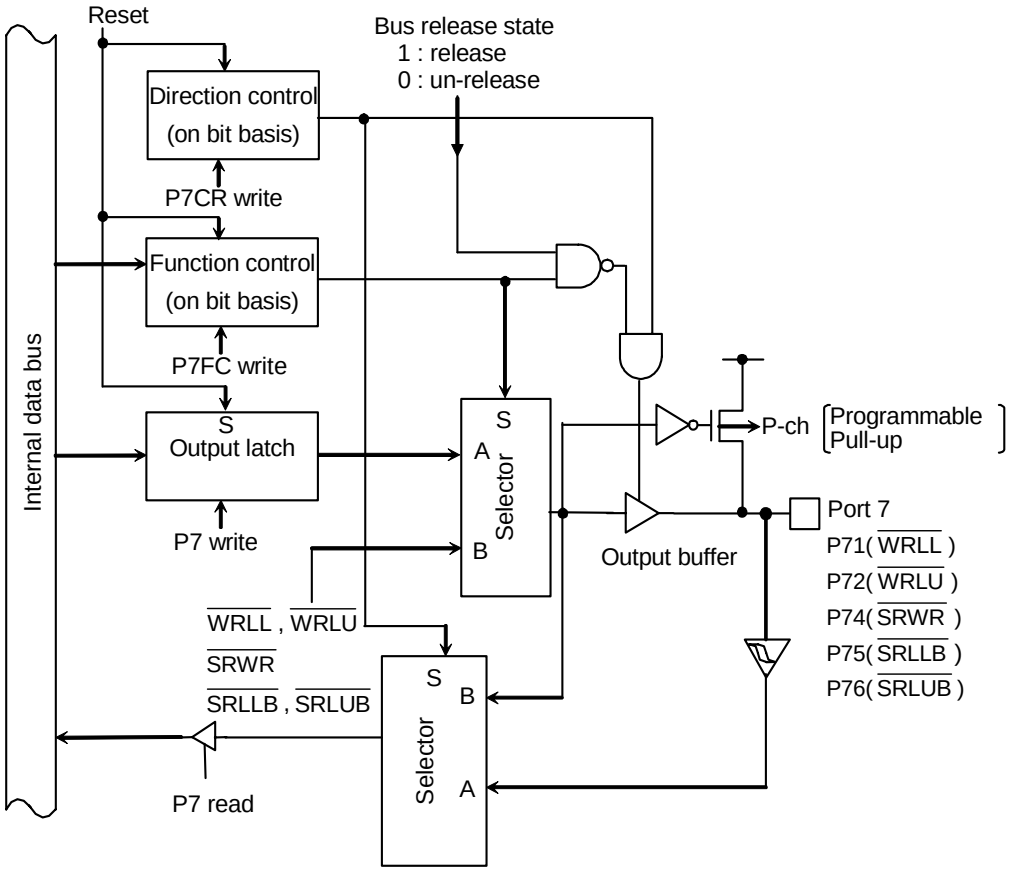


Figure 3.5.5 Port 7(P71,P72,P74,P75,P76)

Note) When a terminal is set as $\overline{\text{WRLL}}$, $\overline{\text{WRLU}}$, $\overline{\text{SRWR}}$, $\overline{\text{SRLLB}}$, $\overline{\text{SRLUB}}$ and $\overline{\text{WAIT}}$, at the time of bus release, an output buffer is not concerned with an output setup of control register P7CR, but is turned OFF.

(2) P73 (R/$\overline{W}$)

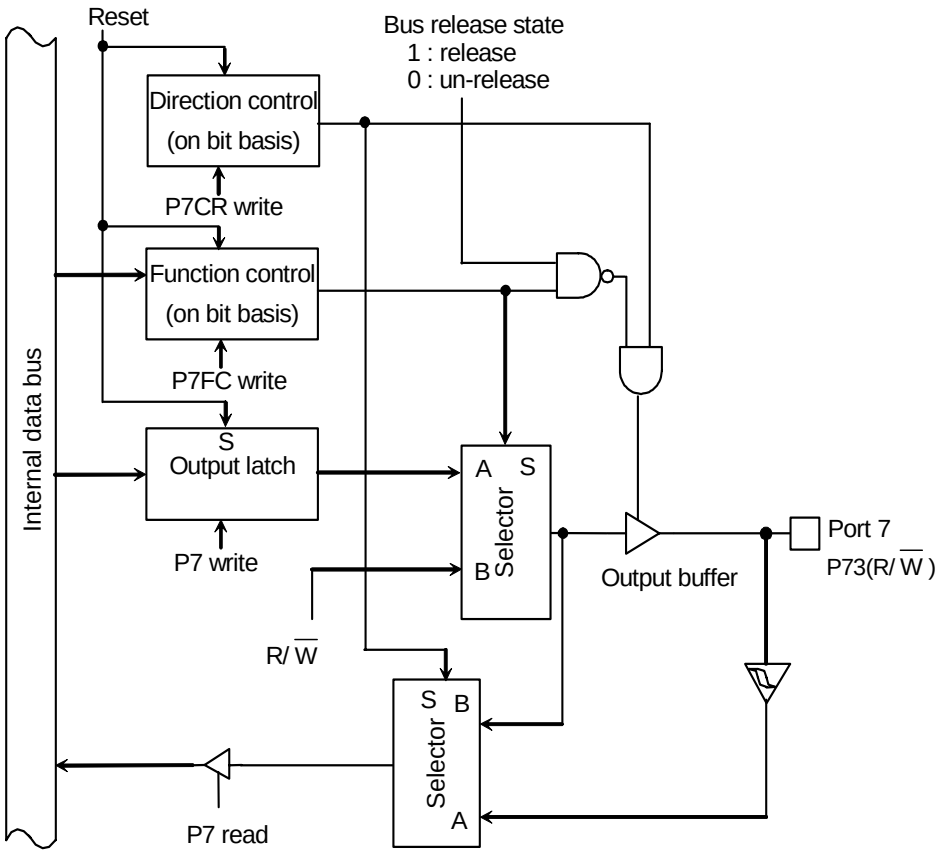


Figure 3.5.6 Port 7(P73)

Note) When a terminal is set as R/$\overline{W}$, at the time of bus release, an output buffer is not concerned with an output setup of control register P7CR, but is turned OFF.

(3) P77($\overline{\text{WAIT}}$)

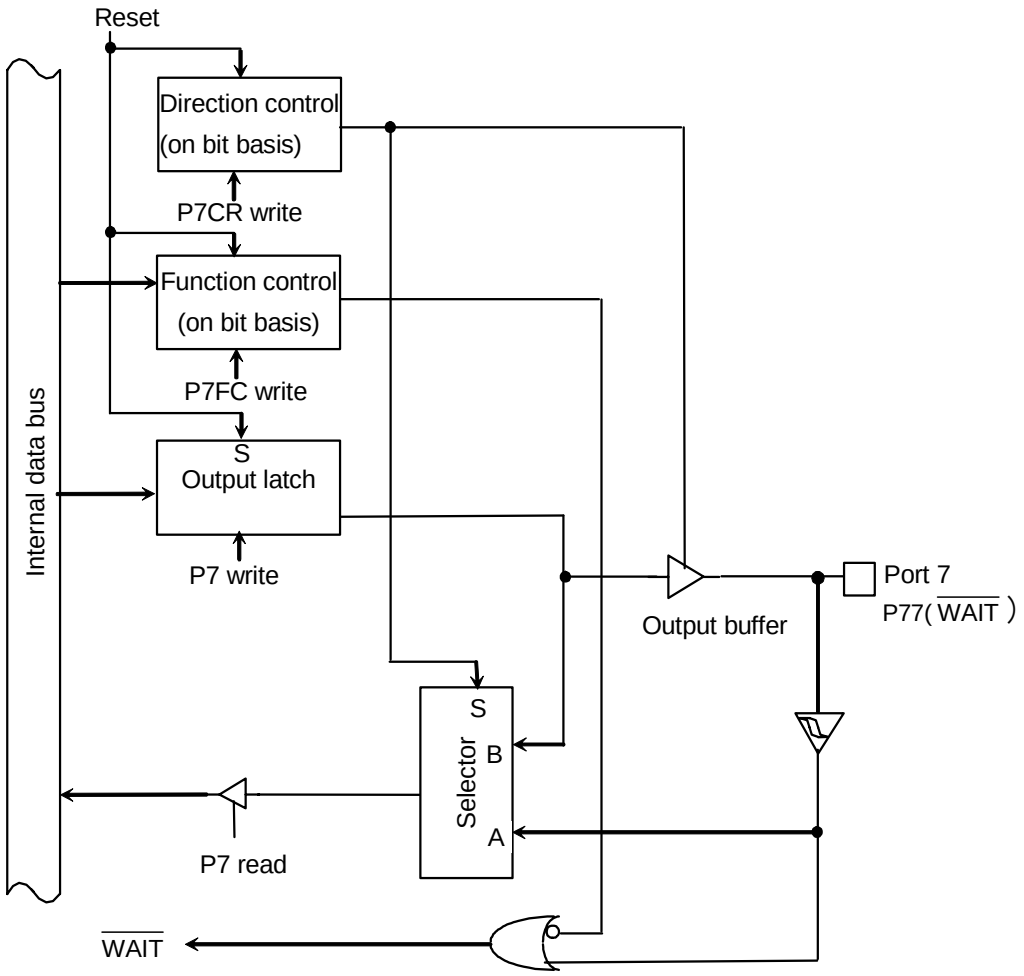


Figure 3.5.7 Port 7(P77)

Port 7 register

P7 (001CH)

	7	6	5	4	3	2	1	
bit Symbol	P77	P76	P75	P74	P73	P72	P71	
Read/Write	R/W							
After reset	Data from external port(Output latch register is set to ”1”)							
Function	-	0:Pull-up register OFF 1:Pull-up register ON			-	0:Pull-up register OFF 1:Pull-up register ON		-

Port 7 Control register

P7CR (001EH)

	7	6	5	4	3	2	1	
bit Symbol	P77C	P76C	P75C	P74C	P73C	P72C	P71C	
Read/Write	W							
After reset	0	0	0	0	0	0	0	-
	0: Input 1: Output							

Port 7 Function register

P7FC (001FH)

	7	6	5	4	3	2	1	
bit Symbol	P77F	P76F	P75F	P74F	P73F	P72F	P71F	
Read/Write	W							
After reset	0	0	0	0	0	0	0	-
Function	0: Port 1: $\overline{WAIT}$	0: Port 1: $\overline{SRLUB}$	0: Port 1: $\overline{SRLLB}$	0: Port 1: $\overline{SRWR}$	0: Port 1: R/$\overline{W}$	0: Port 1: $\overline{WRLU}$	0: Port 1: $\overline{WRLL}$	-

Port 7 function setting

<P7xF>	<P7xC>	P77	P76	P75	P74	P73	P72	P71	
0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	-
0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port	-
1	0	$\overline{WAIT}$	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	-
1	1	Reserved	$\overline{SRLUB}$	$\overline{SRLLB}$	$\overline{SRWR}$	R/$\overline{W}$	$\overline{WRLU}$	$\overline{WRLL}$	-

Note 1) When using P71,P72 and P74 to P76 in input mode, built-in pull-up resistance is controlled by port7 register. When using it, making input mode ot I/O mode intermingled, a Read-modigy-write is forbidden(When at least 1 bit of input terminals exists). A setup of built-in pull-up resistance may change according to the state of an input terminal.

Note 2) Read-modify-write is prohibited for P7CR and P7FC.

Note 3) In the case of a port function, about pull-up ON/OFF, it controls by the value of P7. When using it asafunction, it controls by the value of a function.

Figure 3.5.8 Port 7 register

3.5.4 Port 8 (P80 to P87)

P80 to P85 are a port only for outputs. P86 and P87 are general-purpose I/O ports.

There are the following functions in addition to an output and a general-purpose I/O port.

- The output function of a standard chip select signal($\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, $\overline{CS4}$, $\overline{CS5}$).
- The output function of the chip select signal for SDRAM($\overline{SDCS}$).
- The I/O function of a bus release function($\overline{BUSRQ}$, $\overline{BUSAK}$).
- The output function of a watchdog timer($\overline{WDTOUT}$).

These functions operate by setting the bit concerned of P8CR, P8FC and P8FC2 register. The value of each register of P8CR, P8FC, and P8FC2 is reset in "0" by the reset operation, P80 to P84 becomes an output port, P85 becomes $\overline{WDTOUT}$ output, and P86 and P87 become the input ports. Moreover, P82 is reset in "0" as for the output latch, and P80, P81, and P83 to P87 are set in "1".

(1) P80($\overline{CS0}$), P81($\overline{CS1}$), P84($\overline{CS4}$)

P80, P81, and P84 function as standard chip select signal output ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS4}$) besides the output port function.

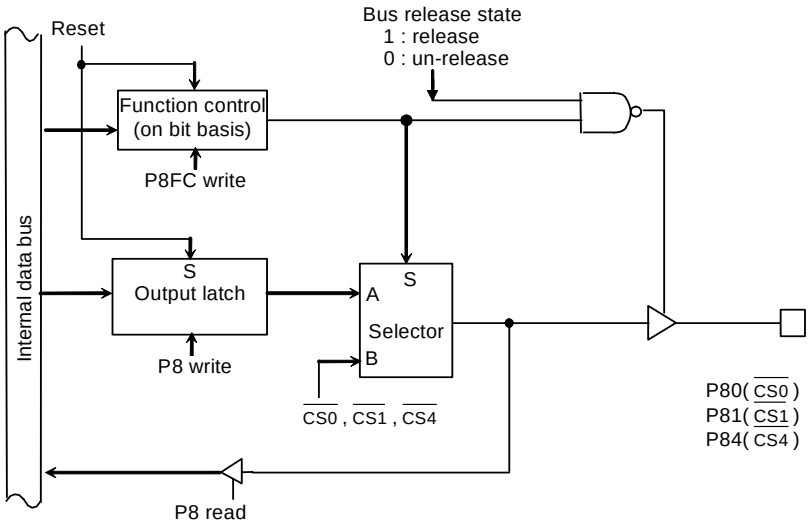


Figure 3.5.9 Port 8(P80,P81,P84)

(2) P82($\overline{CS2}$)

P82 functions as standard chip select signal output ($\overline{CS2}$) besides the output port function.

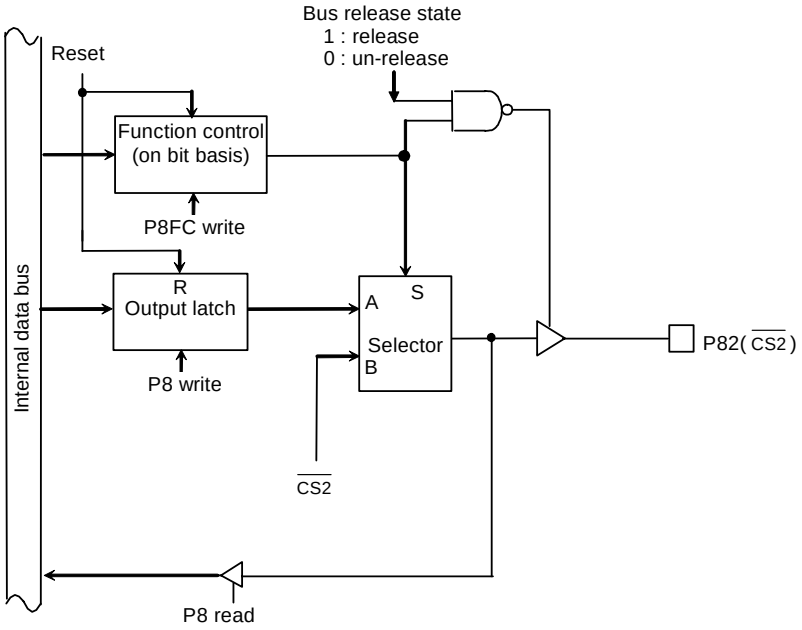


Figure 3.5.10 Port 8(P82)

(3) P83($\overline{CS3}$, $\overline{SDCS}$)

P83 functions as standard chip selection signal output ($\overline{CS3}$) and chip select signal output ($\overline{SDCS}$) for SDRAM besides the output port function.

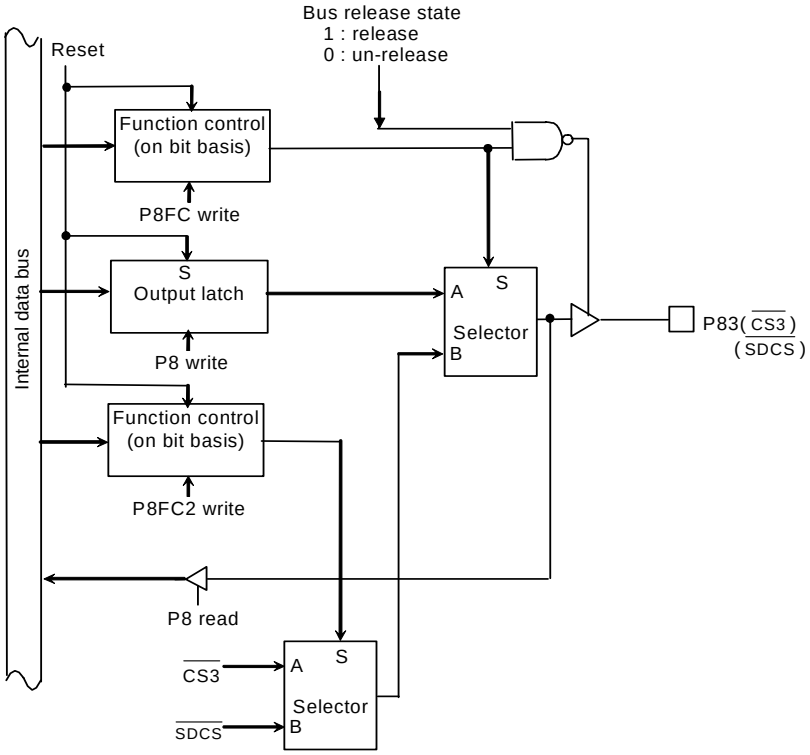


Figure 3.5.11 Port 8(P83)

(4)P85($\overline{CS5}$)

P85 functions as standard chip select signal output ($\overline{CS5}$) and watchdog timer signal output ($\overline{WDTOUT}$) besides the output port function.

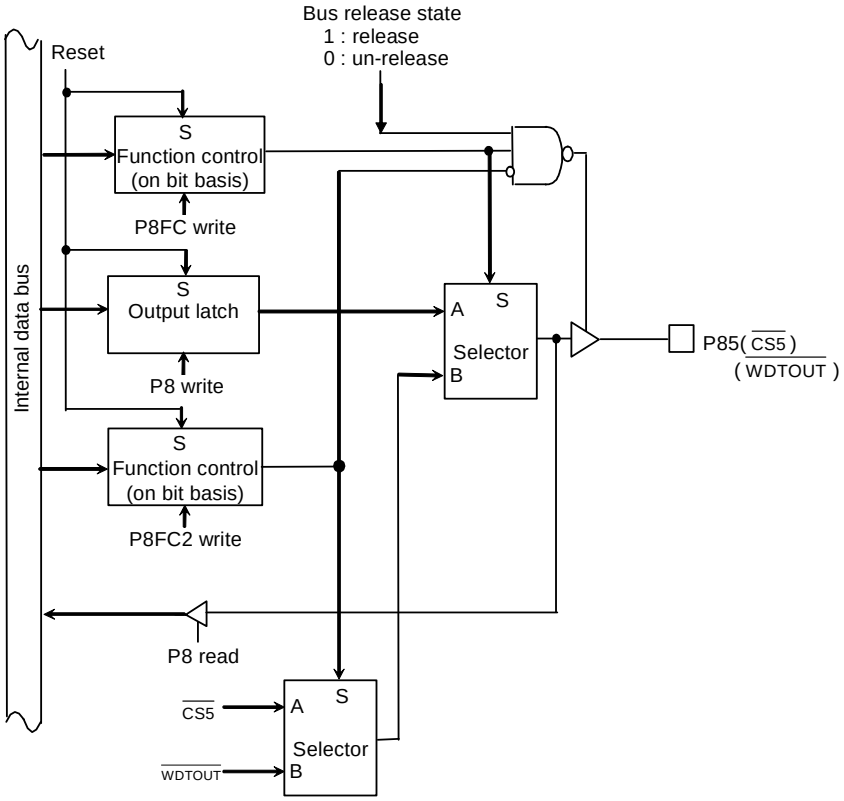


Figure 3.5.12 Port 8(P85)

(5)P86($\overline{\text{BUSRQ}}$)

P86 functions as input ($\overline{\text{BUSRQ}}$) of the function of bus open besides the I/O port function.

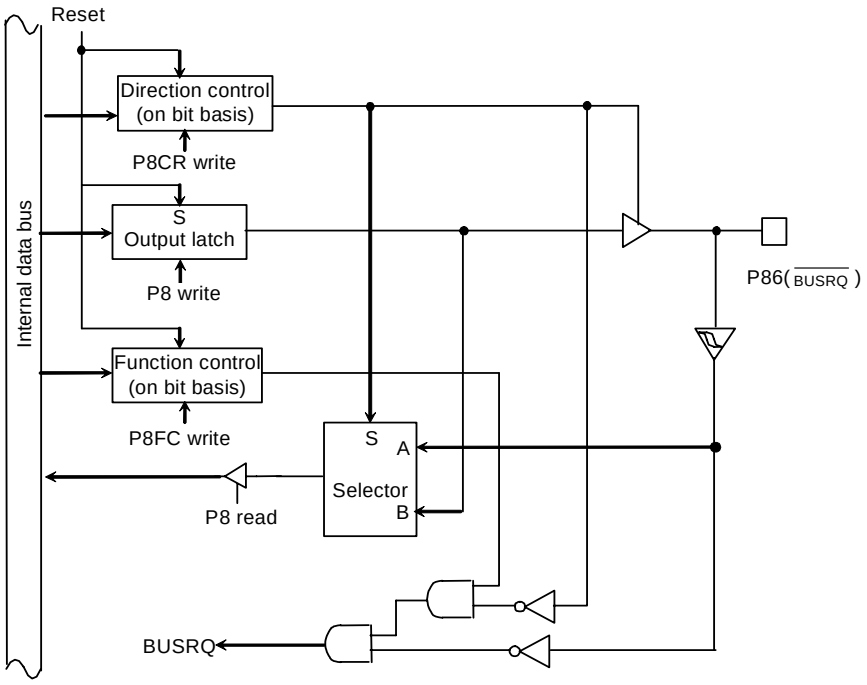


Figure 3.5.13 Port 8(P86)

(6)P87($\overline{\text{BUSAK}}$)

P87 functions as output ($\overline{\text{BUSAK}}$) of the function of bus open besides the I/O port function.

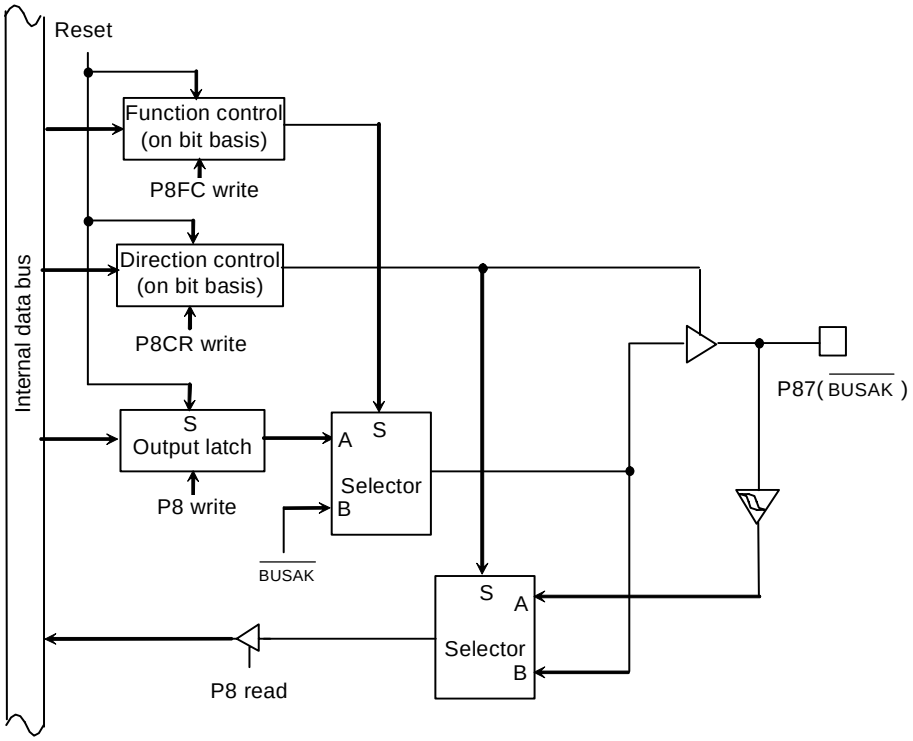


Figure 3.5.14 Port 8(P87)

Port 8 register

P8 (0020H)

	7	6	5	4	3	2	1	0
bit Symbol	P87	P86	P85	P84	P83	P82	P81	P80
Read/Write	R/W							
After reset	Data from external port (Output latch register is set to "1")		1	1	1	0	1	1

Port 8 Control register

P8CR (0021H)

	7	6	5	4	3	2	1	0
bit Symbol	P87C	P86C						
Read/Write	W							
After reset	0	0						
Function	0: Input 1: Output							

Port 8 Function register

P8FC (0022H)

	7	6	5	4	3	2	1	0
bit Symbol	P87F	P86F	P85F	P84F	P83F	P82F	P81F	P80F
Read/Write	W							
After reset	0	0	1	0	0	0	0	0
Function	0: Port 1: $\overline{\text{BUSAK}}$	0: Port 1: $\overline{\text{BUSRQ}}$	0: Port 1:<P85F2>	0: Port 1: $\overline{\text{CS4}}$	0: Port 1:<P83F2>	0: Port 1: $\overline{\text{CS2}}$	0: Port 1: $\overline{\text{CS1}}$	0: Port 1: $\overline{\text{CS0}}$

Port 8 Function register 2

P8FC2 (0023H)

	7	6	5	4	3	2	1	0
bit Symbol			P85F2		P83F2			
Read/Write			W		W			
After reset			1		0			
Function			0: $\overline{\text{CS5}}$ 1: $\overline{\text{WDTOUT}}$		0: $\overline{\text{CS3}}$ 1: $\overline{\text{SDCS}}$			

P85 function setting

<P85F> / <P85F2>	0	1
0	Port	$\overline{\text{CS5}}$
1	Reserved	$\overline{\text{WDTOUT}}$

P83 function setting

<P83F> / <P83F2>	0	1
0	Port	$\overline{\text{CS3}}$
1	Reserved	$\overline{\text{SDCS}}$

Note 1) Read-modify-write is prohibited for P8CR, P8FC and P8FC2.

Note 2) Don't do "1" to P8<P82> register in the write before setting P82 to $\overline{\text{CS2}}$ after releasing reset.
The period when (P8FC<P82F>=1) that sets the function register after the value of the output latch of P82 is made "1" (P8<P82>=1) and the output are not normally output exists and it is likely not to operate correctly.

Note 3) Use and set word instruction (LDW (P8FC),xxxxH) when you set P82 as $\overline{\text{CS2}}$.

Figure 3.5.15 Port 8 register

3.5.5 Port 9 (P90 to P96)

P90 to P96 are a port only for outputs.

There are the following functions in addition to an output port.

・The output function of a SDRAM controller

($\overline{\text{SDWE}}$, $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, SDLLDQM, SDLUDQM, SDCKE, SDCLK).

These functions operate by setting the bit concerned of P9FC register. The value of P9FC<P95:P90 > is reset in "0" by the reset operation, and P95 to P90 becomes an output port. The value of P9FC<P96F > is set in "1", and P96 becomes SDCLK function output. Moreover, all bits of the output latch are set in "1". Port 9 can bitting set the output in HALT. It sets it by the P9DR register.

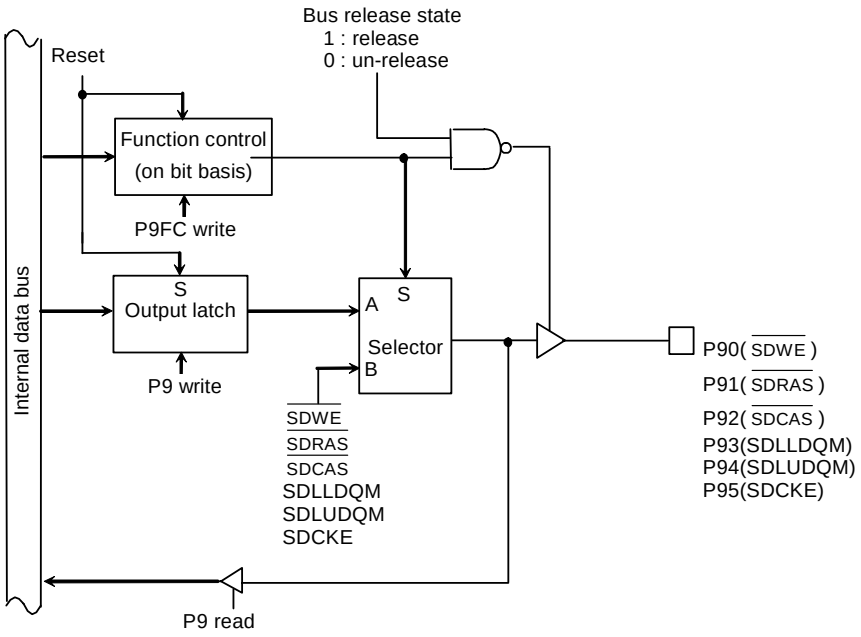


Figure 3.5.16 Port 9(P90 to P95)

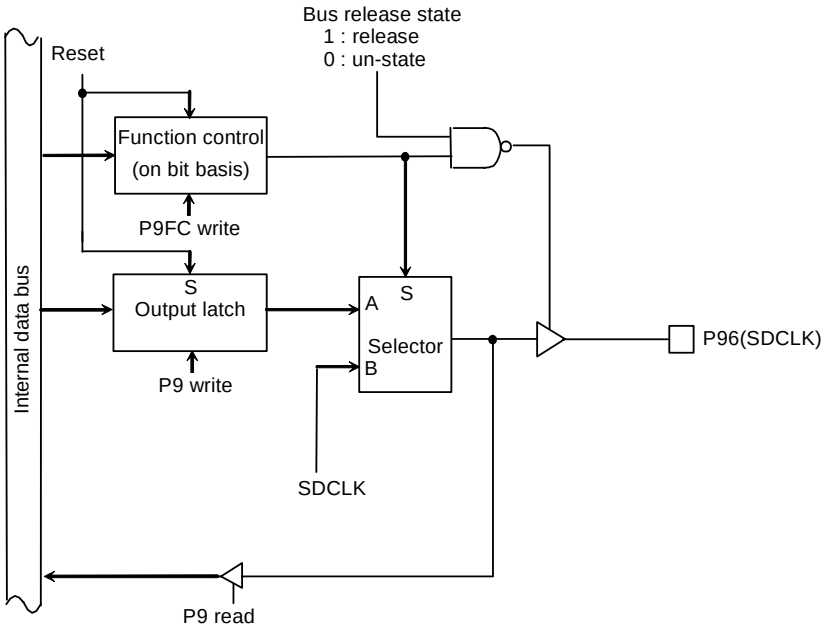


Figure 3.5.17 Port 9(P96)

Port 9 register

P9 (0024H)

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

Port 9 Function register

P9FC (0027H)

	7	6	5	4	3	2	1	0
bit Symbol		P96F	P95F	P94F	P93F	P92F	P91F	P90F
Read/Write		W						
After reset		1	0	0	0	0	0	0
Function		0:Port 1:SDCLK	0:Port 1:SDCKE	0:Port 1:SDLUDQM	0:Port 1:SDLLDQM	0:Port 1: $\overline{\text{SDCAS}}$	0:Port 1: $\overline{\text{SDRAS}}$	0:Port 1: $\overline{\text{SDWE}}$

Port 9 Drive register

P9DR (0025H)

	7	6	5	4	3	2	1	0
bit Symbol		P96D	P95D	P94D	P93D	P92D	P91D	P90D
Read/Write		R/W						
After reset		1	1	1	1	1	1	1
Function		0: The inside of HALT is high impedance、 1: The inside of HALT is also driven						

(The purpose of use and the usage)

- This register sets up the state of each pin at the time of standby mode.
- Set the state of the pin expected before the "HALT" command as a register. CPU serves as enable, after executing a "HALT" command.
- It becomes effective in all the standby modes that have three kinds.(IDLE2,IDLE1, or STOP mode)
- The state of I/O is shown in the following tables.

OE	P9nD	Output buffer	Input buffer
1	0	OFF	OFF
1	1	ON	OFF

Note 1) OE means the output enable signal before the mode of the standby.
Note 2) “n” of P9nD means the bit number of PORT9.

Note) Read-modify-write is prohibited for P9FC.

Figure 3.5.18 Port 9 register

3.5.6 Port A (PA0 to PA5)

Port A is an 6-bit general-purpose I/O port.

PA1 and PA4 can be set as an open drain output.

There are the following functions in addition to an I/O port.

- The I/O function of the serial cannel 0(RXD0, TXD0, SCLK0/$\overline{\text{CTS0}}$).
- The I/O function of the serial cannel 1(RXD1, TXD1, SCLK1/$\overline{\text{CTS1}}$).

These functions operate by setting the bit concerned of PACR, PAFC and PAFC2 register. All the bits of PACR, PAFC and PAFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

(1)PA0(RXD0),PA3 (RXD1)

PA0 and PA3 have a function as a RXD input of the serial channel 0 and 1 in addition to an I/O port.

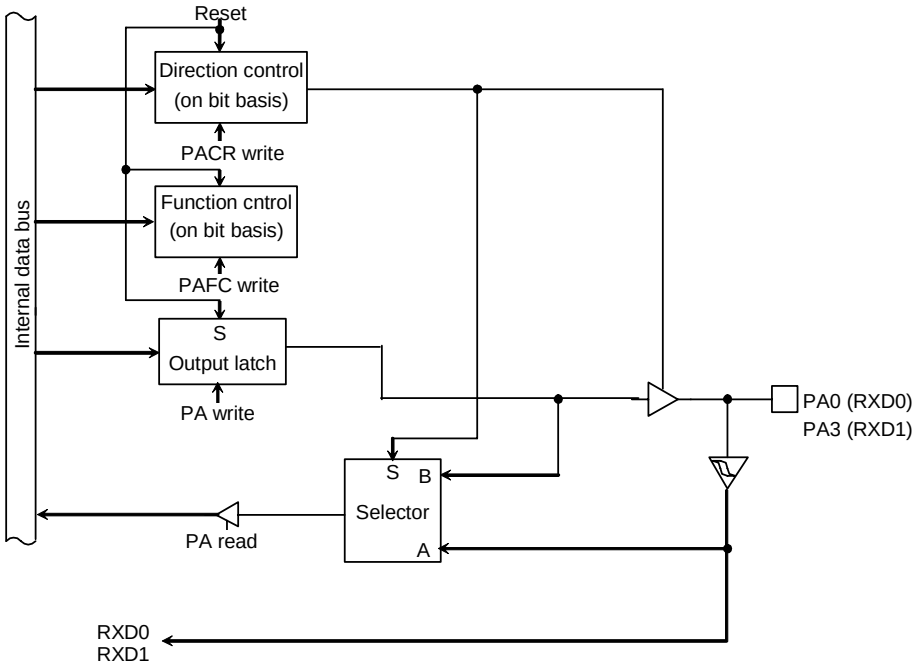


Figure 3.5.19 Port A(PA0,PA3)

(2)PA1(TXD0),PA4 (TXD1)

PA1 and PA4 have a function as a TXD output of the serial channel 0 and 1 in addition to an I/O port.

Moreover, when using it as an TXD output of the serial bus interfaces 0 and 1, the output buffer has the open drain function in which a program is possible. An open drain function can be set up by the PAFC<PA1F,PA4F> and PACR<PA1C,PA4C> register.

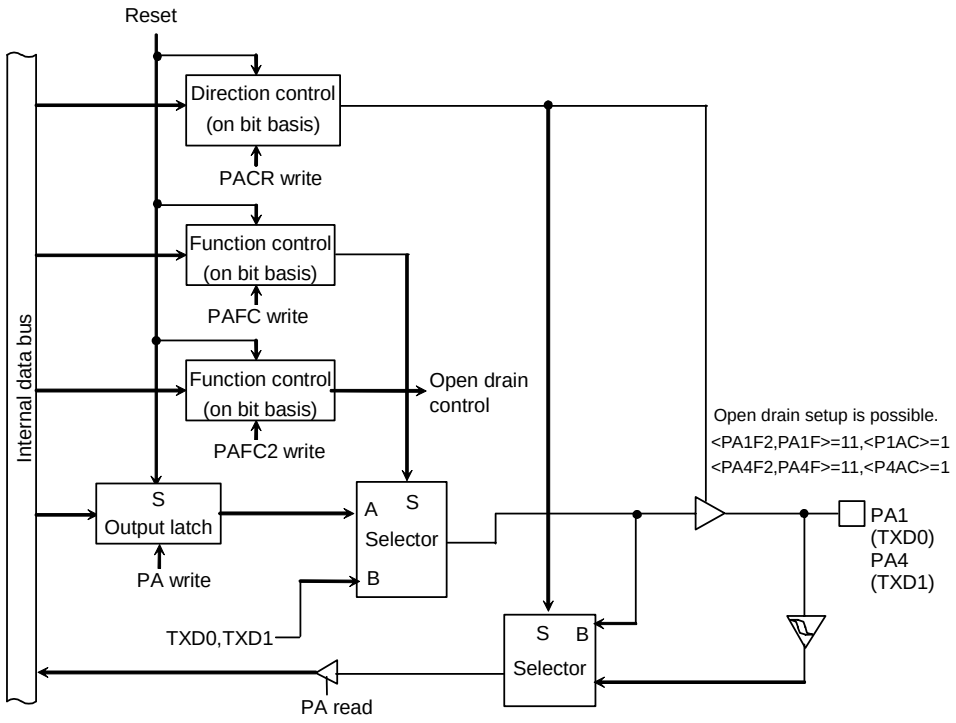


Figure 3.5.20 Port A(PA1,PA4)

(3)PA2($\overline{\text{CTS0}}$,SCLK0),PA5($\overline{\text{CTS1}}$,SCLK1)

PA2 and PA5 have a function as an $\overline{\text{CTS}}$ input or SCLK I/O in addition to the I/O port.

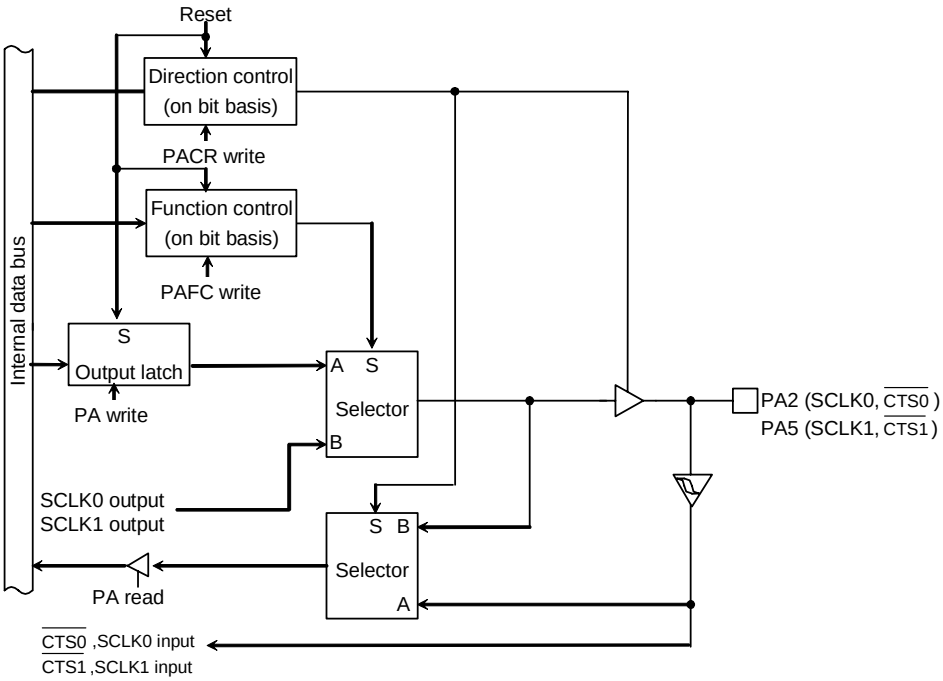


Figure 3.5.21 Port A(PA2,PA5)

Port A register

		7	6	5	4	3	2	1	0
PA (0028H)	bit Symbol			PA5	PA4	PA3	PA2	PA1	PA0
	Read/Write			R/W					
	After reset			Data from external port(Output latch register is set to “1”)					

Port A Control register

		7	6	5	4	3	2	1	0
PACR (002AH)	bit Symbol			PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
	Read/Write			W					
	After reset			0	0	0	0	0	0
				Refer to following table					

Port A Function register

		7	6	5	4	3	2	1	0
PAFC (002BH)	bit Symbol			PA5F	PA4F	PA3F	PA2F	PA1F	PA0F
	Read/Write			W					
	After reset			0	0	0	0	0	0
	Function			Refer to following table					

Port A Function register 2

	7	6	5	4	3	2	1	0
bit Symbol				PA4F2			PA1F2	
Read/Write				W			W	
After reset				0			0	
Function				Refer to following table			Refer to following table	

Port A function setting

<PAxF2>	<PAxF>	<PAxC>	PA5	PA4	PA3	PA2	PA1	PA0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	SCLK1/ $\overline{\text{CTS1}}$	Reserved	RXD1	SCLK0/ $\overline{\text{CTS0}}$	Reserved	RXD0
0	1	1	SCLK1	TXD1(O.D Dis)	Reserved	SCLK0	TXD0(O.D Dis)	Reserved
1	0	0		Reserved			Reserved	
1	0	1		Reserved			Reserved	
1	1	0		Reserved			Reserved	
1	1	1		TXD1(O.D Ena)			TXD0(O.D Ena)	

Note 1) Read-modify-write is prohibited for PACR, PAFC and PAFC2.

Note 2) RXD0/1, SCLK0/1, $\overline{\text{CTS0}}$ and $\overline{\text{CTS1}}$ input are inputted into the serial bus interface 0 and 1 irrespective of a functional setup of a port.

Note 3) PA1 and PA4 do not have a register for 3-state/open drain setup.
Moreover, there is no open drain function at the time of an output port.

Figure 3.5.22 Port A register

3.5.7 Port C(PC0 to PC5)

Port C is an 6-bit general-purpose I/O port.

PC0, PC1, PC3 and PC4 can be set as an open drain output.

There are the following functions in addition to an I/O port.

- The I/O function of the serial bus interface 0(SO0/SDA0, SI0/SCL0, SCK0).
- The I/O function of the serial bus interface 1(SO1/SDA1, SI1/SCL1, SCK1).

These functions operate by setting the bit concerned of PCCR, PCFC and PCFC2 register. All the bits of PCCR, PCFC and PCFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

(1)PC0(SO0/SDA0),PC3 (SO1/SDA1)

PC0 and PC3 have a function as I/O of the serial bus interface 0 and 1 in addition to an I/O port.

Moreover, when using it as an output of the serial bus interfaces 0 and 1, the output buffer has the open drain function in which a program is possible. An open drain function can be set up by the PCFC<PC0F,PC3F> and PCCR<PC0C,PC3C> register.

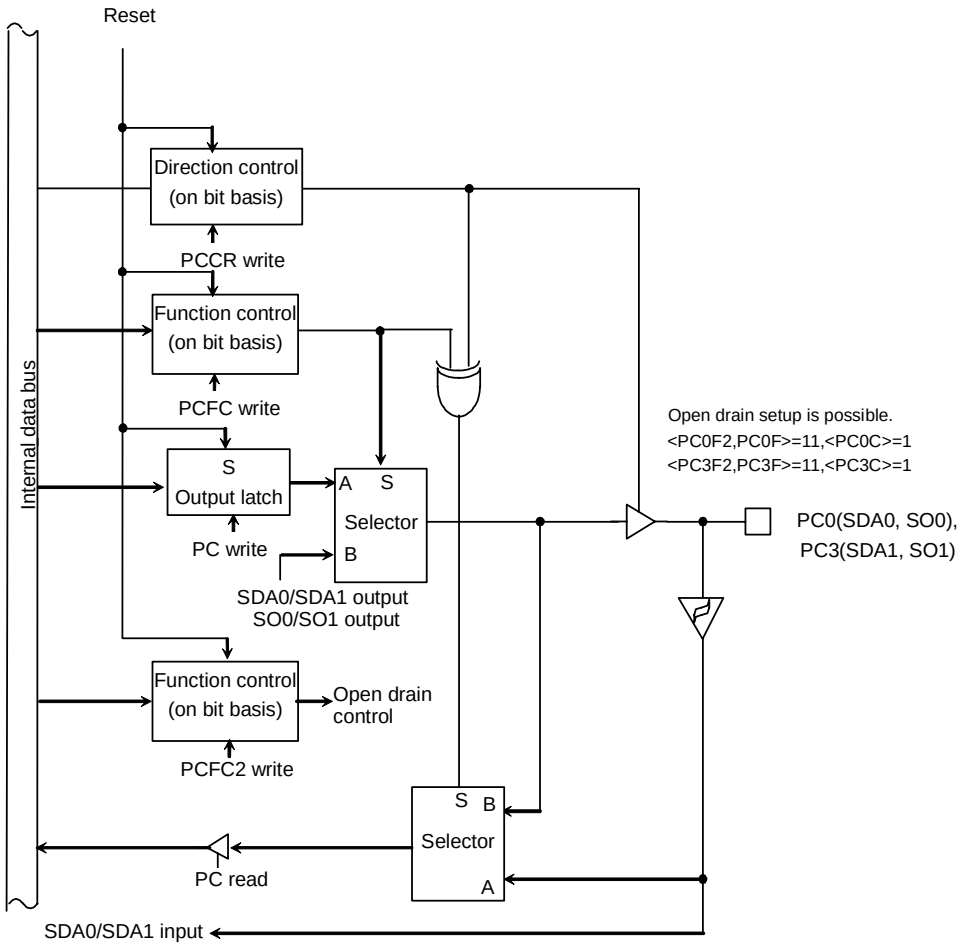


Figure 3.5.23 Port C(PC0,PC3)

(2)PC1(SI0/SCL0),PC4 (SI1/SCL1)

PC1 and PC4 have a function as I/O of the serial bus interface 0 and 1 in addition to an I/O port. Moreover, when using it as an output of the serial bus interfaces 0 and 1, the output buffer has the open drain function in which a program is possible. An open drain function can be set up by the PCFC<PC1F,PC4F> and PCCR<PC1C,PC4C> register.

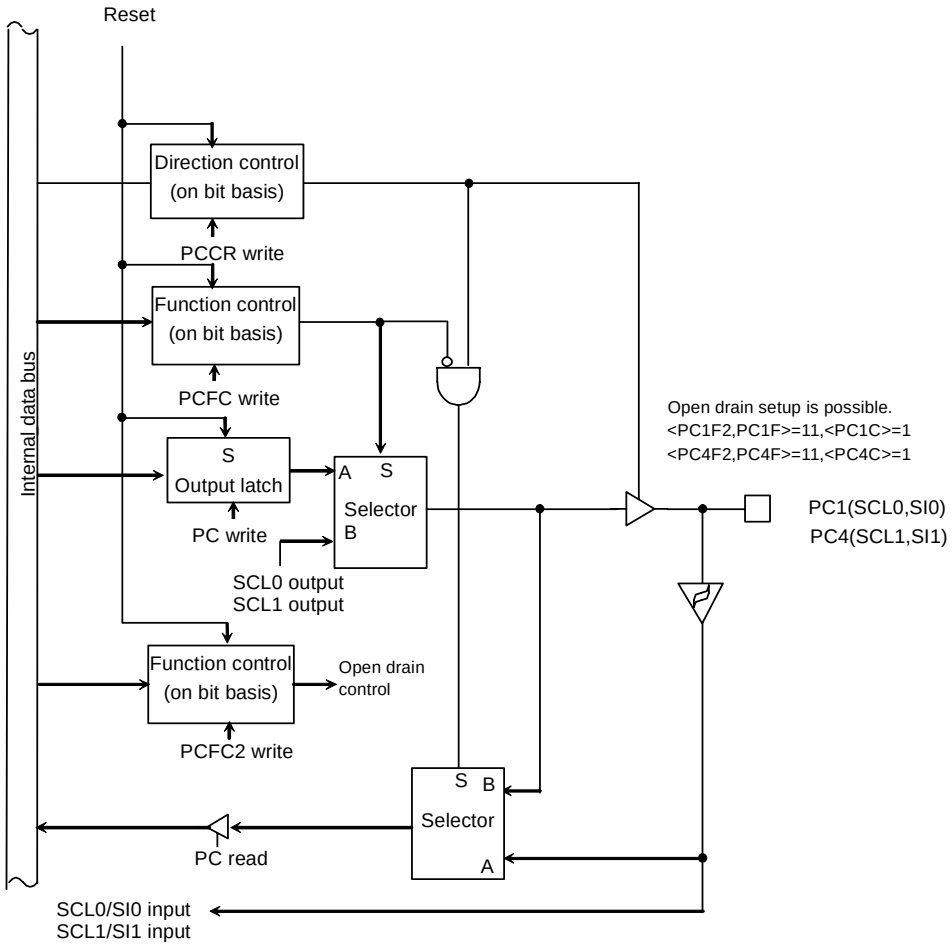


Figure 3.5.24 Port C(PC1,PC4)

(3)PC2(SCK0),PC5 (SCK1)

PC2 and PC5 have a function as I/O of the serial bus interface 0 and 1 in addition to an I/O port.

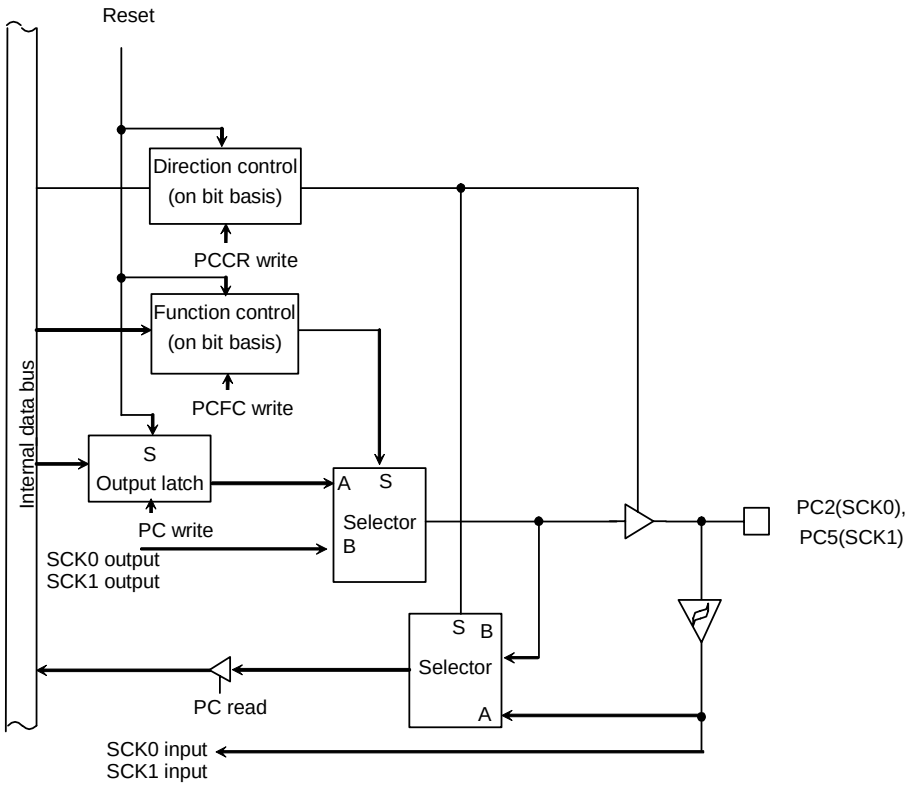


Figure 3.5.25 Port C(PC2,PC5)

Port C register

PC (0030H)

	7	6	5	4	3	2	1	0
bit Symbol			PC5	PC4	PC3	PC2	PC1	PC0
Read/Write			R/W					
After reset			Data from external port(Output latch register is set to “1”)					

Port C Control register

PCCR (0032H)

	7	6	5	4	3	2	1	0
bit Symbol			PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
Read/Write			W					
After reset			0	0	0	0	0	0
			Refer to following table					

Port C Function register

PCFC (0033H)

	7	6	5	4	3	2	1	0
bit Symbol			PC5F	PC4F	PC3F	PC2F	PC1F	PC0F
Read/Write			W					
After reset			0	0	0	0	0	0
Function			Refer to following table					

Port C Function register 2

PCFC2 (0031H)

	7	6	5	4	3	2	1	0
bit Symbol				PC4F2	PC3F2		PC1F2	PC0F2
Read/Write				W	W		W	W
After reset				0	0		0	0
Function				Refer to following table			Refer to following table	

Port C function setting

<PCxF2>	<PCxF>	<PCxC>	PC5	PC4	PC3	PC2	PC1	PC0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	SCK1 input	SI1 input	SO1 output(O.D Dis)	SCK0 input	SI0 input	SO0 output(O.D Dis)
0	1	1	SCK1 output	SCL1 I/O(O.D Dis)	SDA1 I/O(O.D Dis)	SCK0 output	SCL0 I/O(O.D Dis)	SDA0 I/O(O.D Dis)
1	0	0		Reserved	Reserved		Reserved	Reserved
1	0	1		Reserved	Reserved		Reserved	Reserved
1	1	0		Reserved	SO1 output(O.D Ena)		Reserved	SO0 出力(O.D Ena)
1	1	1		SCL1 I/O(O.D Ena)	SDA1 I/O(O.D Ena)		SCL0 I/O(O.D Ena)	SDA0 I/O(O.D Ena)

Note 1) Read-modify-write is prohibited for PCCR, PCFC and PCFC2.

Note 2) SDA0/1, SCL0/1, SI0/1 and SCK0/1 input are inputted into the serial bus interface 0 and 1 irrespective of a functional setup of a port.

Note 3) PC0, PC1, PC3 and PC4 do not have a register for 3-state/open drain setup.
Moreover, there is no open drain function at the time of an output port.

Figure 3.5.26 Port C register

3.5.8 Port D(PD0 to PD5)

Port D is an 6-bit general-purpose I/O port.

PD4 can be set as an open drain output.

There are the following functions in addition to an I/O port.

- The I/O function of the serial cannel 2(RXD2, TXD2, SCLK2/$\overline{\text{CTS2}}$)
- The I/O function of the high speed channel 0(HSSI0, HSSO0, HSCLK0)

These functions operate by setting the bit concerned of PDCR, PDFC and PDFC2 register. All the bits of PDCR, PDFC and PDFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

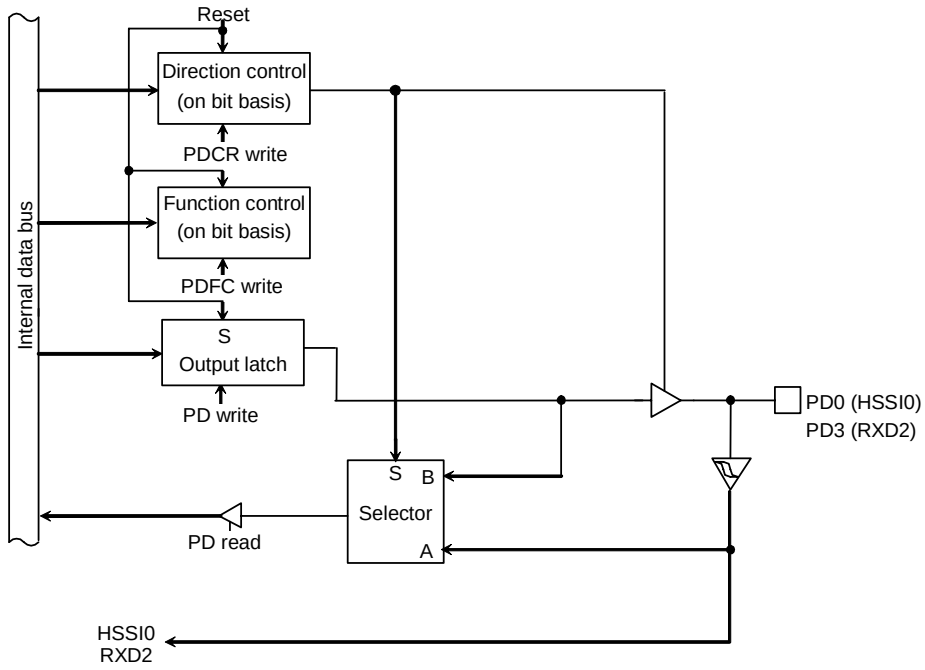


Figure 3.5.27 Port D(PD0,PD3)

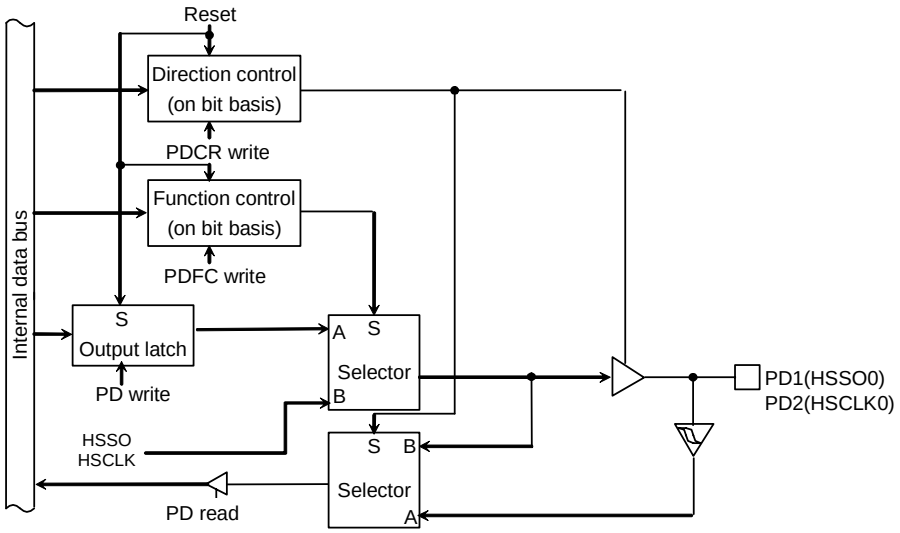


Figure 3.5.28 Port D(PD1,PD2)

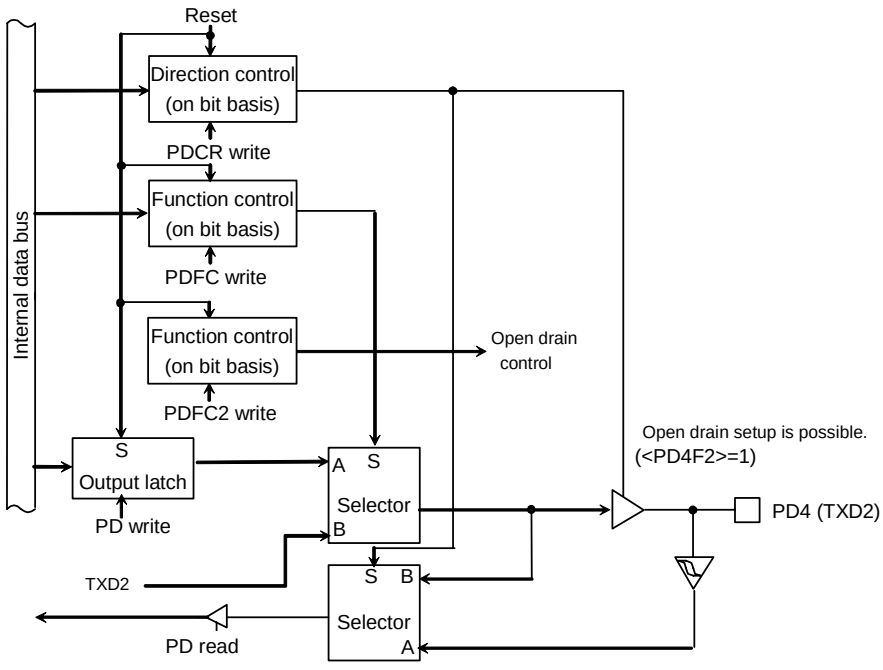


Figure 3.5.29 Port D(PD4)

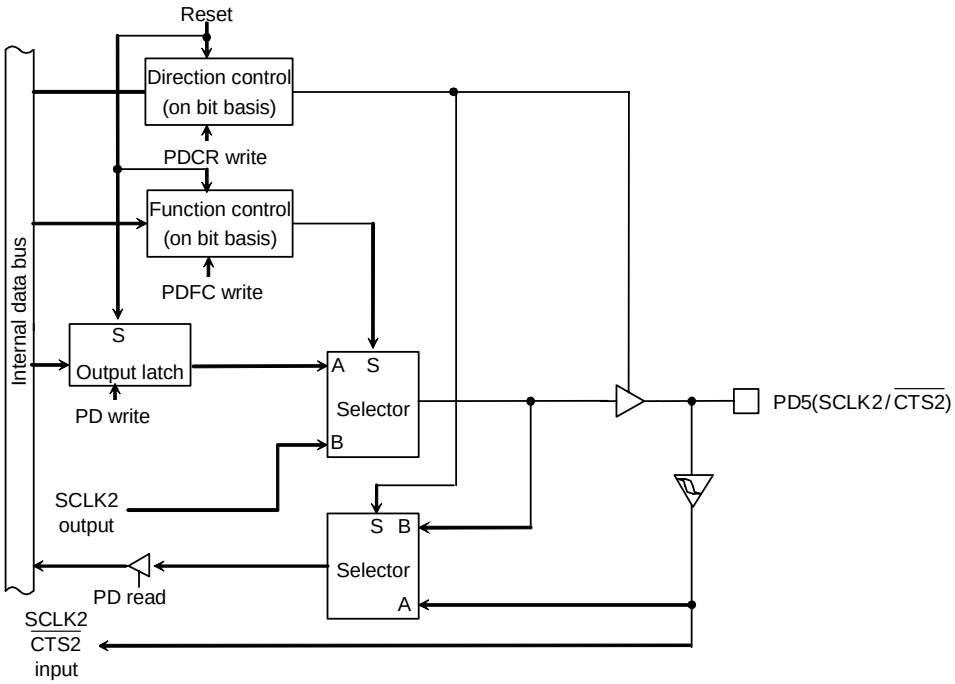


Figure 3.5.30 Port D(PD5)

Port D register

PD (0034H)

	7	6	5	4	3	2	1	0
bit Symbol			PD5	PD4	PD3	PD2	PD1	PD0
Read/Write			R/W					
After reset			Data from external port(Output latch register is set to “1”)					

Port D Control register

PDCR (0036H)

	7	6	5	4	3	2	1	0
bit Symbol			PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
Read/Write			W					
After reset			0	0	0	0	0	0
			Refer to following table					

Port D Function register

PDFC (0037H)

	7	6	5	4	3	2	1	0
bit Symbol			PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
Read/Write			W					
After reset			0	0	0	0	0	0
Function			Refer to following table					

Port D Function register 2

PDFC2 (0035H)

	7	6	5	4	3	2	1	0
bit Symbol				PD4F2				
Read/Write				W				
After reset				0				
Function				Refer to following table				

Port D function setting

<PDxF2>	<PDxF>	<PDxC>	PD5	PD4	PD3	PD2	PD1	PD0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	SCLK2/ $\overline{\text{CTS2}}$	Reserved	RXD2	Reserved	Reserved	HSSI0
0	1	1	SCLK2 output	TXD2(O.D Dis)	Reserved	HSCLK0	HSSO0	Reserved
1	0	0		Reserved				
1	0	1		Reserved				
1	1	0		Reserved				
1	1	1		TXD2(O.D Ena)				

Note 1) Read-modify-write is prohibited for PDCR, PDFC and PDFC2.

Note 2) RXD2, SCLK2 and $\overline{\text{CTS2}}$ input are inputted into the serial bus interface 0 irrespective of a functional setup of a port.

Note 3) HSSI0 input are inputted into the high speed serial channel 0 irrespective of a functional setup of a port.

Note 4) PD4 do not have a register for 3-state/open drain setup.
Moreover, there is no open drain function at the time of an output port.

Figure 3.5.31 Port D register

3.5.9 Port F (PF0 to PF6)

Port F is an 7-bit general-purpose I/O port.

There are the following functions in addition to an I/O port.

- The input function of 8-bit timer 0(TA0IN)
- The output function of 8-bit timer 1(TA1OUT)
- The input function of 8-bit timer 2(TA2IN)
- The output function of 8-bit timer 3(TA3OUT)
- The input function of 8-bit timer 4(TA4IN)
- The output function of 8-bit timer 5(TA5OUT)
- The input function of 8-bit timer 6(TA6IN)
- The input function of external interrupt(INT0 to INT3)

These functions operate by setting the bit concerned of PFCR, PFFC and PFFC2 register. All the bits of PFCR, PFFC and PFFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

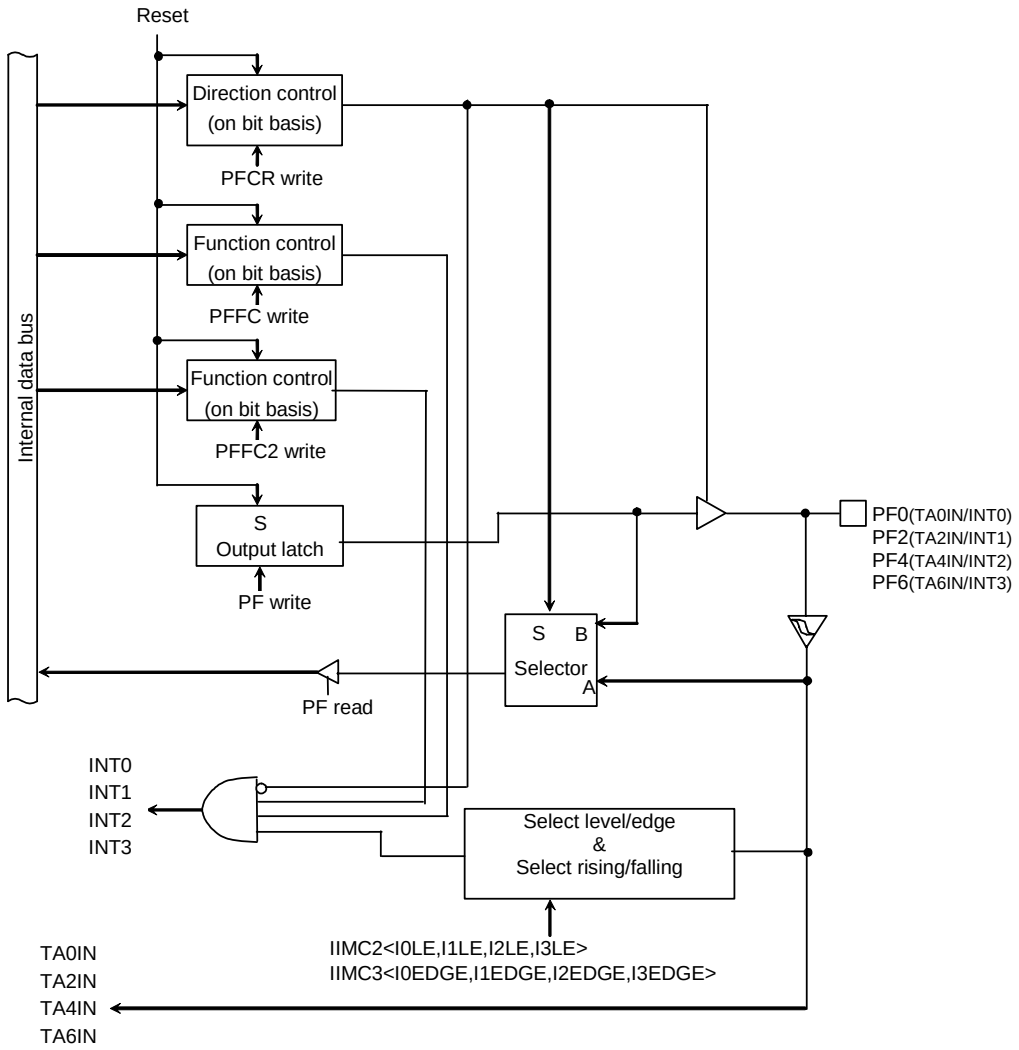


Figure 3.5.32 Port F(PF0,PF2,PF4,PF6)

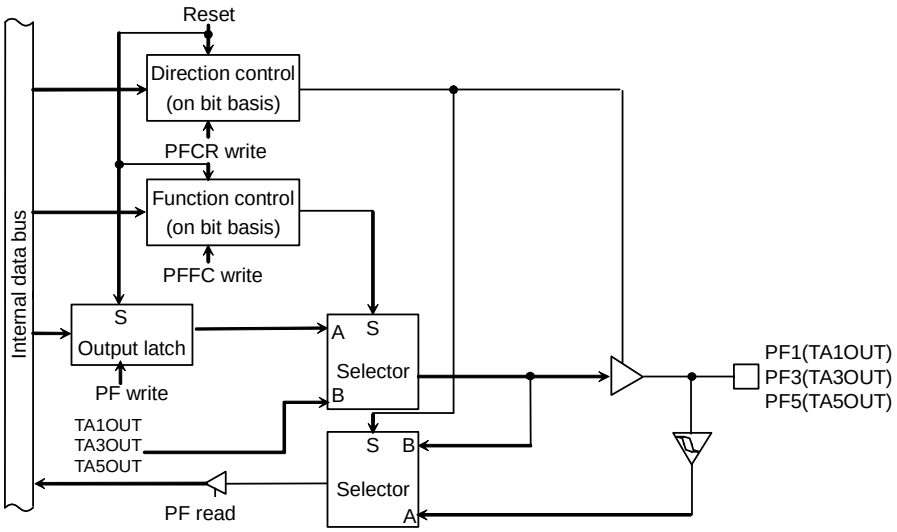


Figure 3.5.33 Port F(PF1,PF3,PF5)

Port F register

		7	6	5	4	3	2	1	0
PF (003CH)	bit Symbol		PF6	PF5	PF4	PF3	PF2	PF1	PF0
	Read/Write		R/W						
	After reset		Data from external port(Output latch register is set to “1”)						

Port F Control register

		7	6	5	4	3	2	1	0
PFCR (003EH)	bit Symbol		PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
	Read/Write		W						
	After reset		0	0	0	0	0	0	0
			Refer to following table						

Port F Function register

		7	6	5	4	3	2	1	0
PFFC (003FH)	bit Symbol		PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
	Read/Write		W						
	After reset		0	0	0	0	0	0	0
	Function		Refer to following table						

Port F Function register 2

		7	6	5	4	3	2	1	0
PFFC2 (003DH)	bit Symbol		PF6F2		PF4F2		PF2F2		PF0F2
	Read/Write		W		W		W		W
	After reset		0		0		0		0
	Function		Refer to following table		Refer to following table		Refer to following table		Refer to following table

Port F function setting

<PFx2>	<PFxF>	<PFxC>	PF6	PF5	PF4	PF3	PF2	PF1	PF0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	TA6IN	Reserved	TA4IN	Reserved	TA2IN	Reserved	TA0IN
0	1	1	Reserved	TA5OUT	Reserved	TA3OUT	Reserved	TA1OUT	Reserved
1	0	0	Reserved		Reserved		Reserved		Reserved
1	0	1	Reserved		Reserved		Reserved		Reserved
1	1	0	INT3		INT2		INT1		INT0
1	1	1	Reserved		Reserved		Reserved		Reserved

Note 1) Read-modify-write is prohibited for PFCR,PFFC and PFFC2.
Note 2) TA0IN, TA2IN, TA4IN and TA6IN input is inputted into the 8-bit timer TMRA0 to TMRA6 irrespective of a functional setup of a port.

Figure 3.5.34 Port F register

3.5.10 Port J (PJ0 to PJ7)

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

There are the following functions in addition to an I/O port.

- The output function of 16-bit timer 0(TB0OUT0, TB0OUT1)
- The output function of 16-bit timer 1(TB1OUT0, TB1OUT1)
- The output function of 16-bit timer 2(TB2OUT0, TB2OUT1)
- The output function of 16-bit timer 3(TB3OUT0, TB3OUT1)
- The output function of 16-bit timer 4(TB4OUT0, TB4OUT1)
- The output function of 16-bit timer 5(TB5OUT0, TB5OUT1)

These functions operate by setting the bit concerned of PJCR, PJFC and PJFC2 register. All the bits of PJCR, PJFC and PJFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

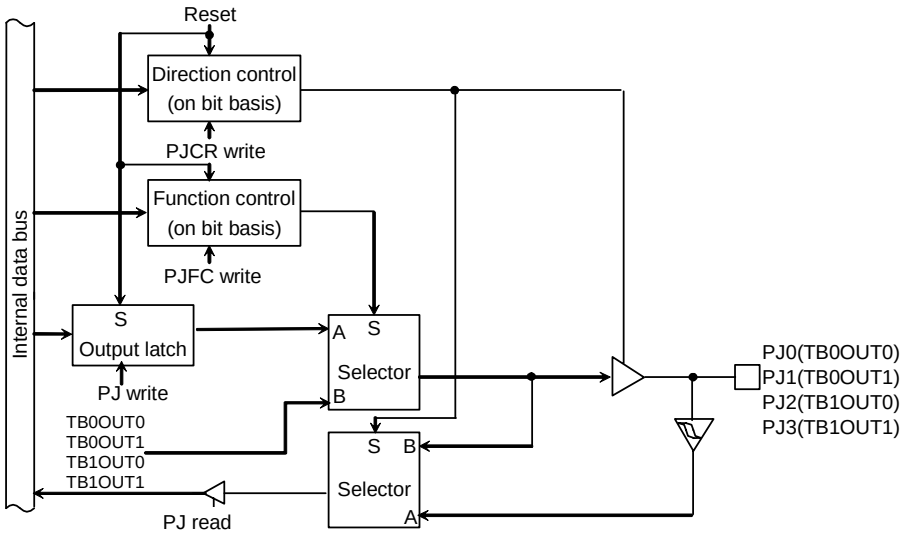


Figure 3.5.35 Port J(PJ0,PJ1,PJ2,PJ3)

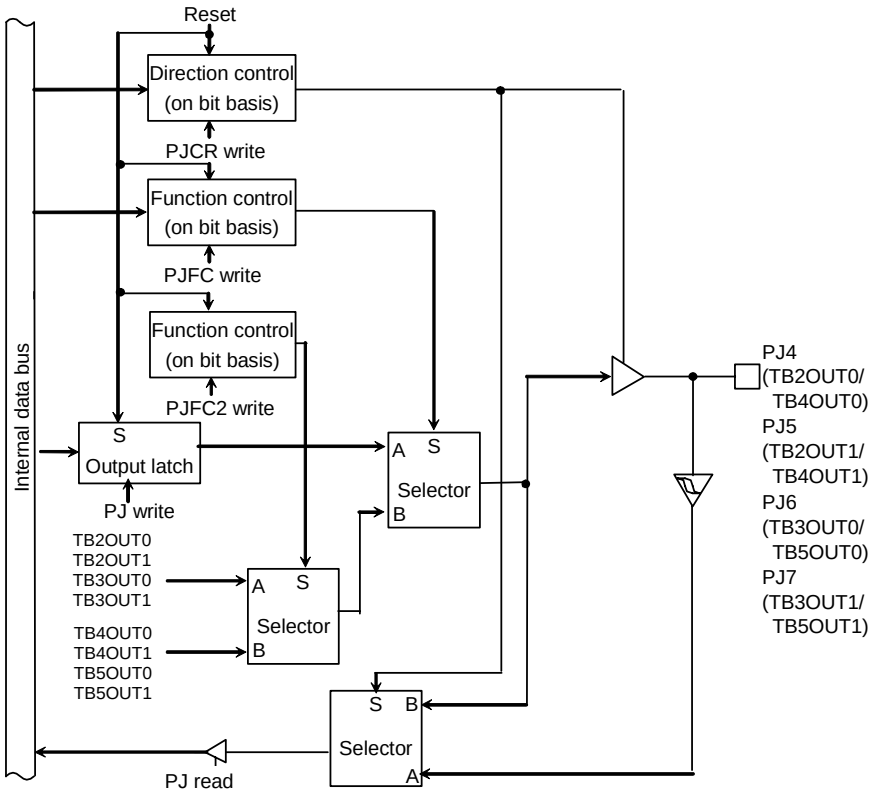


Figure 3.5.36 Port J(PJ4,PJ5,PJ6,PJ7)

Port J register

		7	6	5	4	3	2	1	0
PJ (004CH)	bit Symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
	Read/Write	R/W							
	After reset	Data from external port(Output latch register is set to “1”)							

Port J Control register

		7	6	5	4	3	2	1	0
PJCR (004EH)	bit Symbol	PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
	Read/Write	W							
	After reset	0							
		Refer to following table							

Port J Function register

		7	6	5	4	3	2	1	0
PJFC (004FH)	bit Symbol	PJ7F	PJ6F	PJ5F	PJ4F	PJ3F	PJ2F	PJ1F	PJ0F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Refer to following table							

Port J Function register 2

		7	6	5	4	3	2	1	0
PJFC2 (004DH)	bit Symbol	PJ7F2	PJ6F2	PJ5F2	PJ4F2				
	Read/Write	W							
	After reset	0	0	0	0				
	Function	Refer to following table							

Port J function setting

<PJx2>	<PJxF>	<PJxC>	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	1	1	TB3OUT1	TB3OUT0	TB2OUT1	TB2OUT0	TB1OUT1	TB1OUT0	TB0OUT1	TB0OUT0
1	0	0	Reserved	Reserved	Reserved	Reserved				
1	0	1	Reserved	Reserved	Reserved	Reserved				
1	1	0	Reserved	Reserved	Reserved	Reserved				
1	1	1	TB5OUT1	TB5OUT0	TB4OUT1	TB4OUT0				

Note) Read-modify-write is prohibited for PJCR,PJFC and PJFC2.

Figure 3.5.37 Port J register

3.5.11 Port K (PK0 to PK7)

Port K are a port only for inputs.

There are the following functions in addition to an input port.

- The input function of 16-bit timer 0(TB0IN0, TB0IN1)
- The input function of 16-bit timer 1(TB1IN0, TB1IN1)
- The input function of 16-bit timer 2(TB2IN0, TB2IN1)
- The input function of 16-bit timer 3(TB3IN0, TB3IN1)
- The input function of external interrupt(INT4 to INTB)

These functions operate by setting the bit concerned of PKFC and PKFC2 register. All the bits of PKFC and PKFC2 are cleared to "0" by the reset action, and all bits serve as an input port.

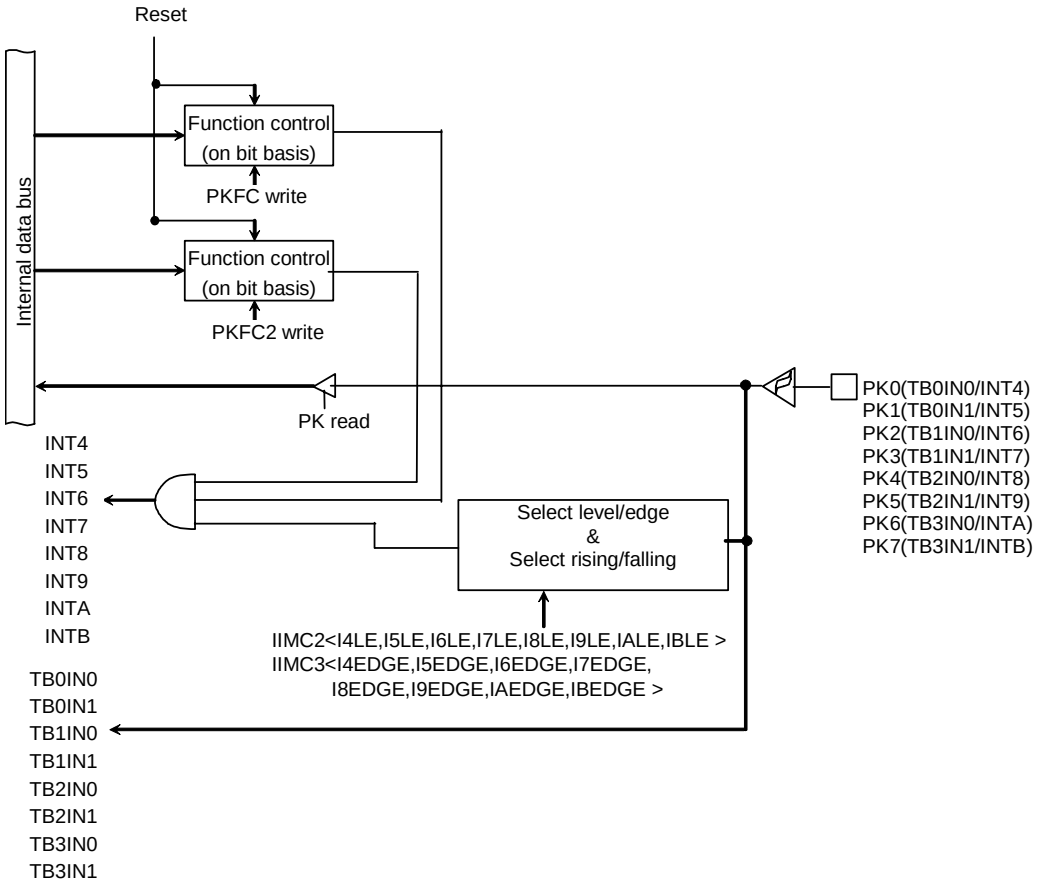


Figure 3.5.38 Port K(PK0 to PK7)

Port K register

PK (0050H)

	7	6	5	4	3	2	1	0
bit Symbol	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
Read/Write	R							
After reset	Data from external port							

Port K Function register

PKFC (0053H)

	7	6	5	4	3	2	1	0
bit Symbol	PK7F	PK6F	PK5F	PK4F	PK3F	PK2F	PK1F	PK0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port K Function register 2

PKFC2 (0051H)

	7	6	5	4	3	2	1	0
bit Symbol	PK7F	PK6F	PK5F	PK4F	PK3F	PK2F	PK1F	PK0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port K function setting

<PKxF2>	<PKxF>	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	1	TB3IN1	TB3IN0	TB2IN1	TB2IN0	TB1IN1	TB1IN0	TB0IN1	TB0IN0
1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1	1	INTB	INTA	INT9	INT8	INT7	INT6	INT5	INT4

Note 1) Read-modify-write is prohibited for PKFC and PKFC2.

Note 2) TB0IN0/1, TB1IN0/1, TB2IN0/1 and TB3IN0/1 input is inputted into the 8-bit timer TMRB0 to TMRB3 irrespective of a functional setup of a port.

Note 3) In setting (TB0IN0 to TB3IN1) of setting (INT4 to INTB) of <PKxF2>=1 and <PKxF>=1 and <PKxF2>=0 and <PKxF>=1, the operation of the HALT release is different.
Refer to Table 3.3.6 HALT release source of 3.3.6 standby control part and the operation of the HALT release for details.

Figure 3.5.39 Port K register

3.5.12 Port L (PL0 to PL7)

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

PL1 can be set as an open drain output.

There are the following functions in addition to an I/O port.

- The output function of pattern generator 0(PG00 to PG03)
- The output function of pattern generator 1(PG10 to PG13)
- The I/O function of the serial cannel 3(RXD3, TXD3, SCLK3/$\overline{\text{CTS3}}$)
- The output function of 8-bit timer 7(TA7OUT)
- The I/O function of the high speed channel 1(HSSI1, HSSO1, HSCLK1)

These functions operate by setting the bit concerned of PLCR, PLFC and PLFC2 register. All the bits of PLCR, PLFC and PLFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

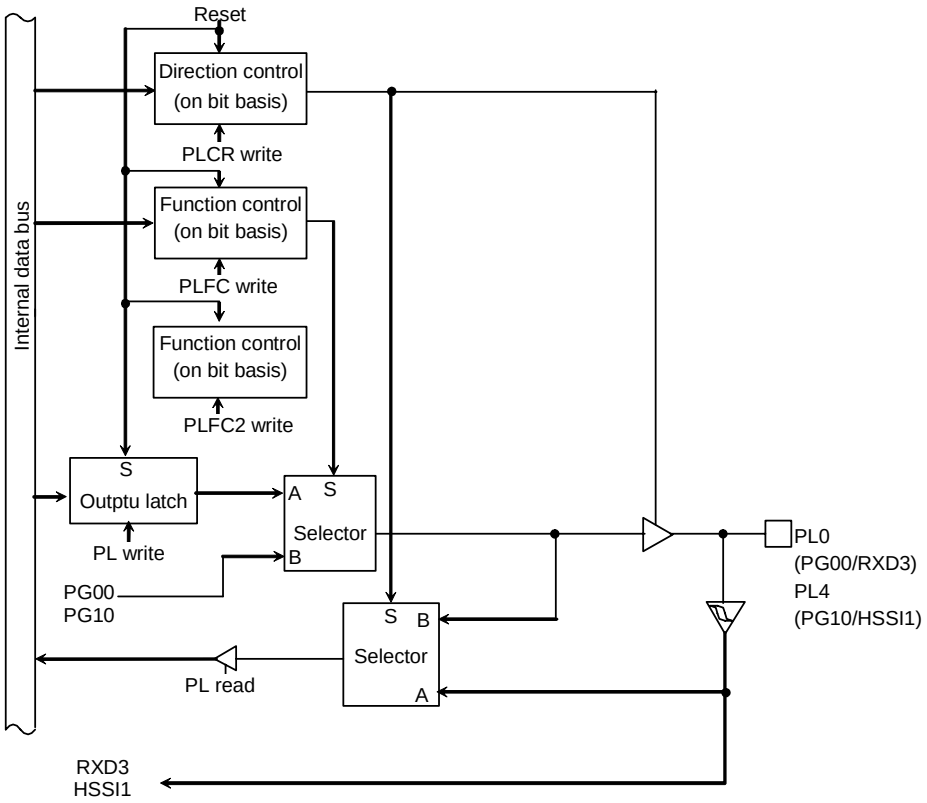


Figure 3.5.40 Port L(PL0,PL4)

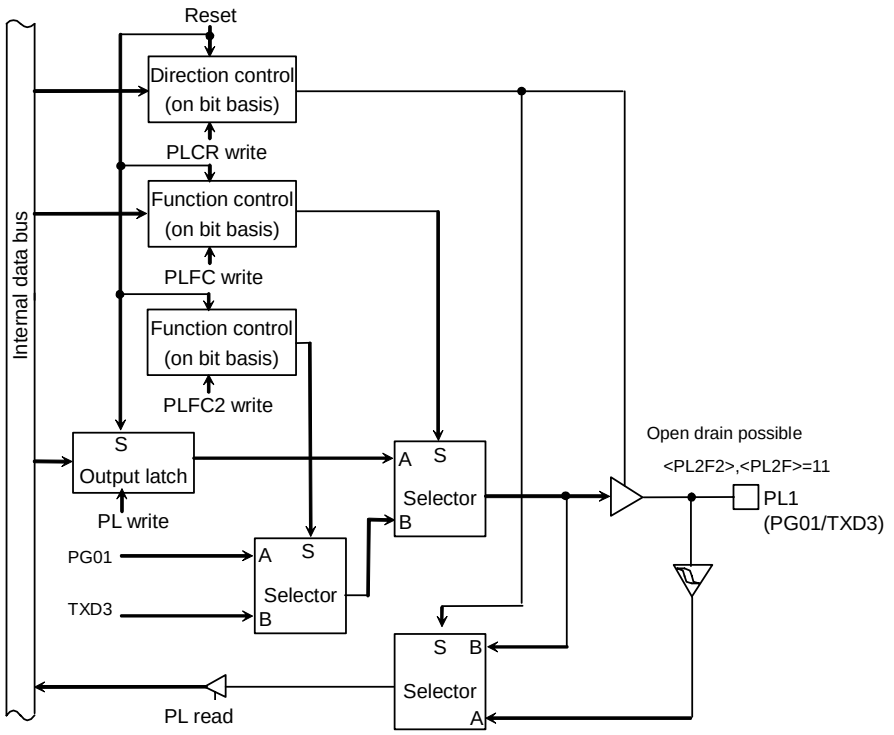


Figure 3.5.41 Port L(PL1)

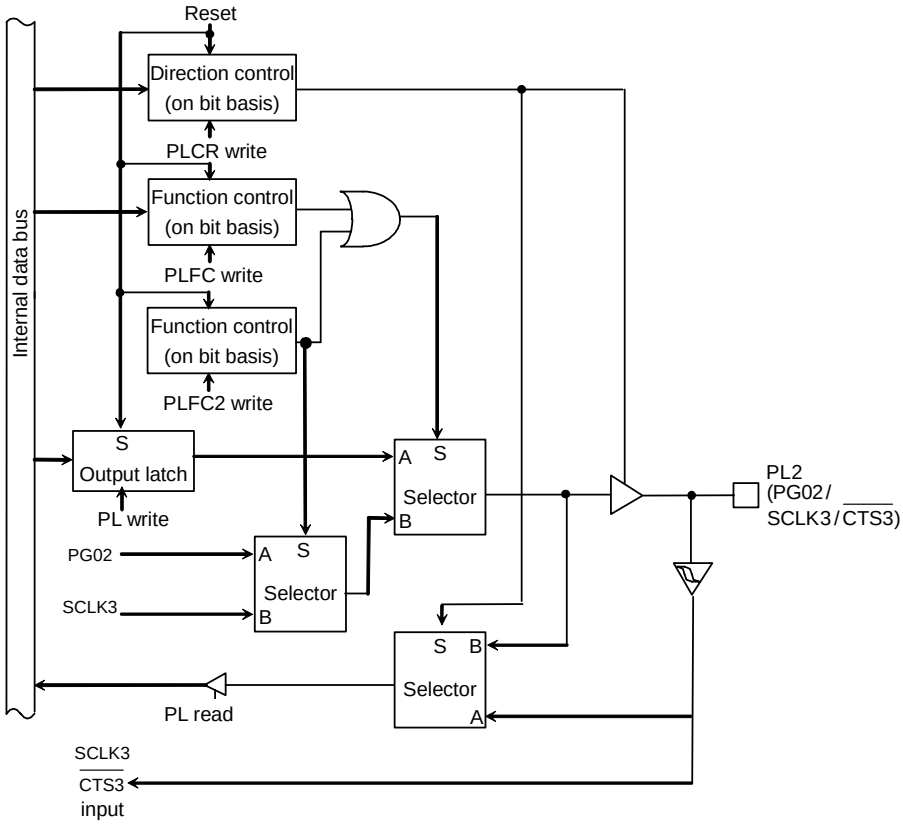


Figure 3.5.42 Port L(PL2)

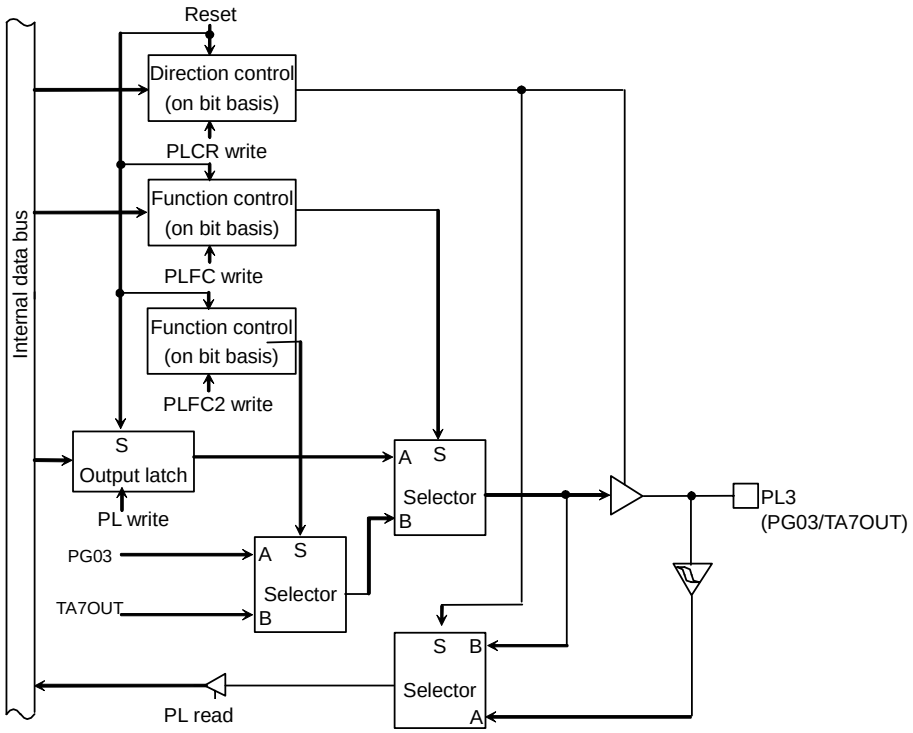


Figure 3.5.43 Port L(PL3)

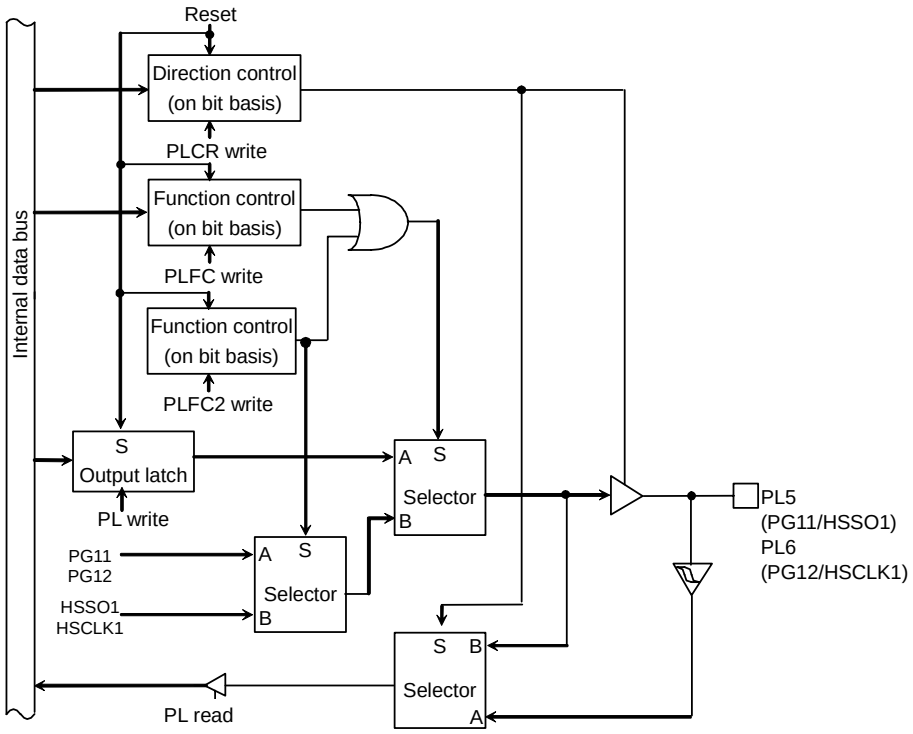


Figure 3.5.44 Port L(PL5,PL6)

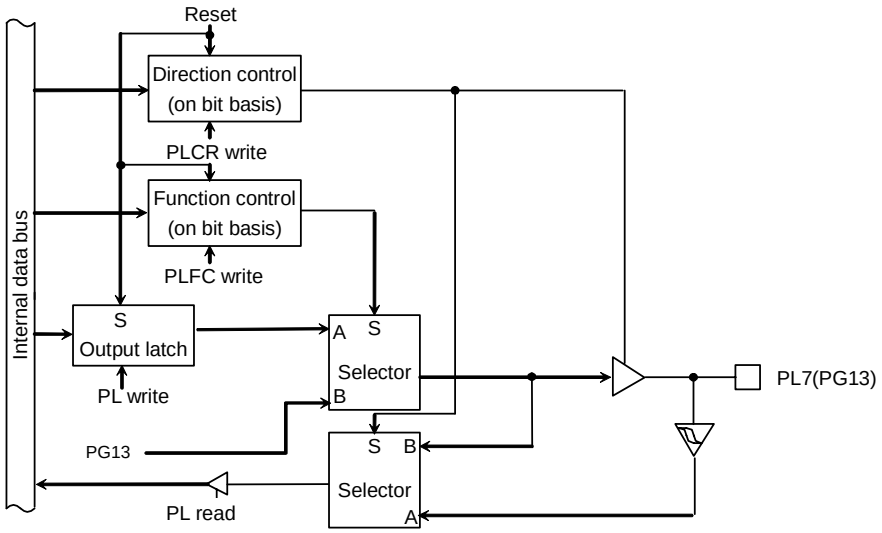


Figure 3.5.45 Port L(PL7)

Port L register

PL (0054H)

	7	6	5	4	3	2	1	0
bit Symbol	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
Read/Write	R/W							
After reset	Data from external port(Output latch register is set to “1”)							

Port L Control register

PLCR (0056H)

	7	6	5	4	3	2	1	0
bit Symbol	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
	Refer to following table							

Port L Function register

PLFC (0057H)

	7	6	5	4	3	2	1	0
bit Symbol	PL7F	PL6F	PL5F	PL4F	PL3F	PL2F	PL1F	PL0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port L Function register 2

PLFC2 (0055H)

	7	6	5	4	3	2	1	0
bit Symbol		PL6F2	PL5F2	PL4F2	PL3F2	PL2F2	PL1F2	PL0F2
Read/Write		W						
After reset		0	0	0	0	0	0	0
Function		Refer to following table						

Port L function setting

<PLxF2>	<PLxF>	<PLxC>	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	1	1	PG13	PG12	PG11	PG10	PG03	PG02	PG01	PG00
1	0	0		Reserved	Reserved	HSSI1	Reserved	SCLK3/ $\overline{\text{CTS3}}$	Reserved	RXD3
1	0	1		HSCLK1	HSSO1	Reserved	Reserved	SCLK3	TXD3 (O.D Dis)	Reserved
1	1	0		Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1	1	1		Reserved	Reserved	Reserved	TA7OUT	Reserved	TXD3 (O.D Ena)	Reserved

Note 1) Read-modify-write is prohibited for PLCR, PLFC and PLFC2.

Note 2) RXD3, SCLK3 and $\overline{\text{CTS3}}$ input are inputted into the serial bus interface 3 irrespective of a functional setup of a port.

Note 3) HSSI1 input are inputted into the high speed serial channel 0 irrespective of a functional setup of a port.

Note 4) PL1 does not have a register for 3-state/open drain setup.
Moreover, there is no open drain function at the time of an output port.

Figure 3.5.46 Port L register

3.5.8 Port D(PD0 to PD5)

Port D is an 6-bit general-purpose I/O port.

PD4 can be set as an open drain output.

There are the following functions in addition to an I/O port.

- The I/O function of the serial cannel 2(RXD2, TXD2, SCLK2/$\overline{\text{CTS2}}$)
- The I/O function of the high speed channel 0(HSSI0, HSSO0, HSCLK0)

These functions operate by setting the bit concerned of PDCR, PDFC and PDFC2 register. All the bits of PDCR, PDFC and PDFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

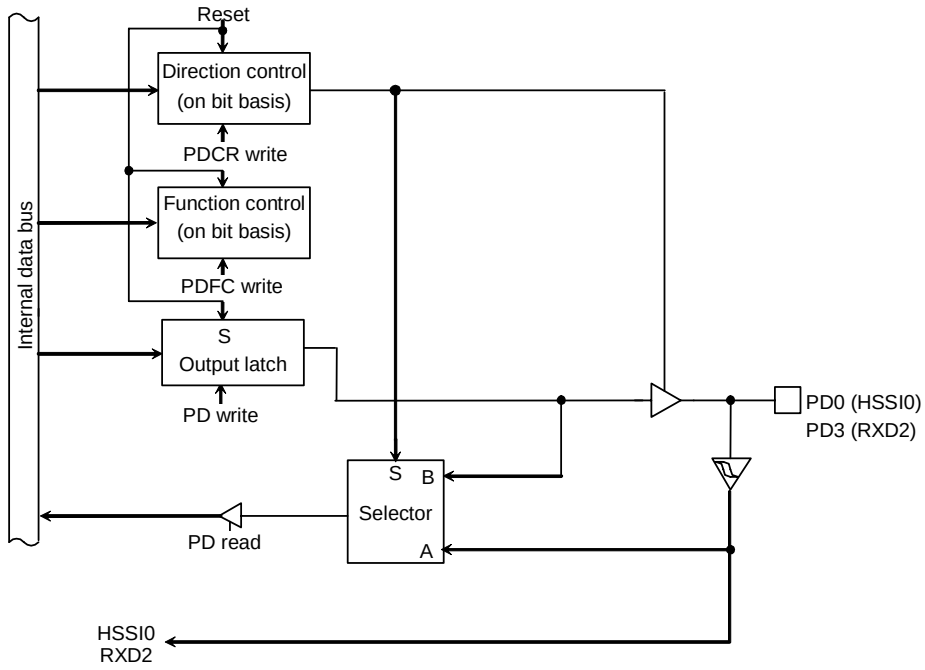


Figure 3.5.27 Port D(PD0,PD3)

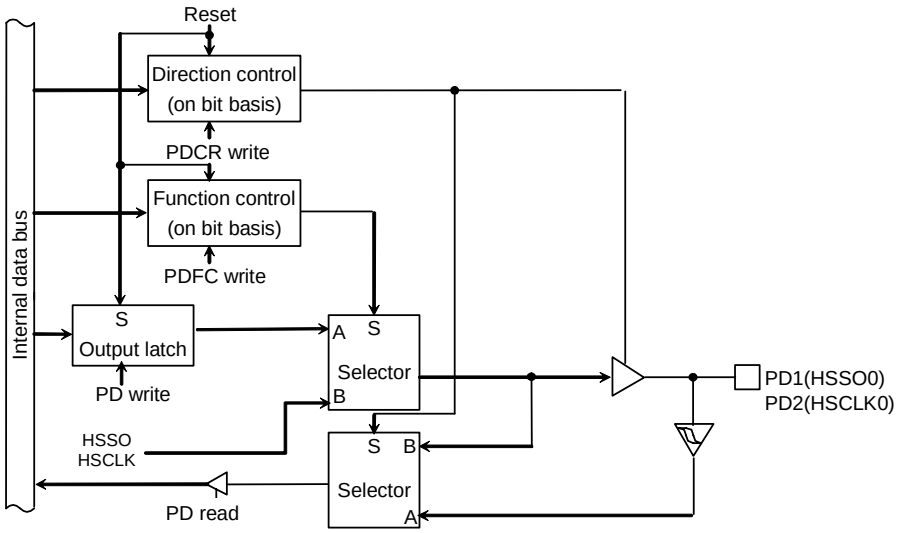


Figure 3.5.28 Port D(PD1,PD2)

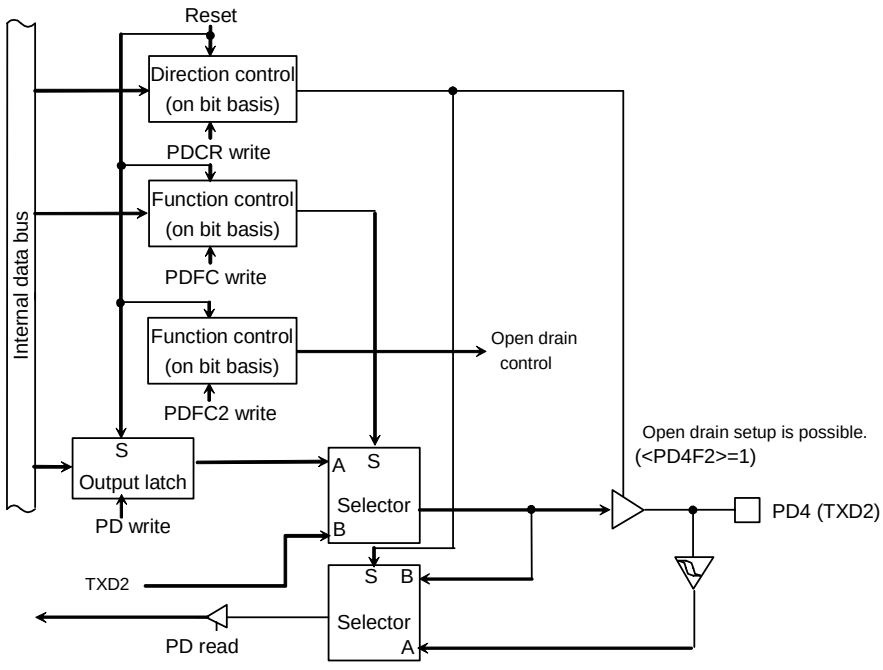


Figure 3.5.29 Port D(PD4)

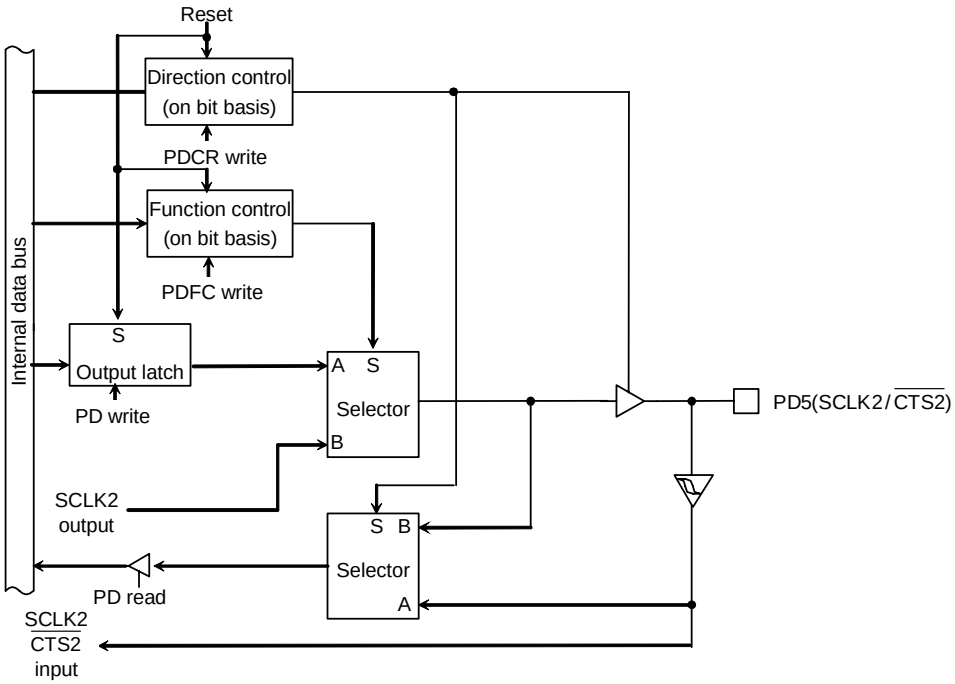


Figure 3.5.30 Port D(PD5)

Port D register

PD (0034H)

	7	6	5	4	3	2	1	0
bit Symbol			PD5	PD4	PD3	PD2	PD1	PD0
Read/Write			R/W					
After reset			Data from external port(Output latch register is set to “1”)					

Port D Control register

PDCR (0036H)

	7	6	5	4	3	2	1	0
bit Symbol			PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
Read/Write			W					
After reset			0	0	0	0	0	0
			Refer to following table					

Port D Function register

PDFC (0037H)

	7	6	5	4	3	2	1	0
bit Symbol			PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
Read/Write			W					
After reset			0	0	0	0	0	0
Function			Refer to following table					

Port D Function register 2

PDFC2 (0035H)

	7	6	5	4	3	2	1	0
bit Symbol				PD4F2				
Read/Write				W				
After reset				0				
Function				Refer to following table				

Port D function setting

<PDxF2>	<PDxF>	<PDxC>	PD5	PD4	PD3	PD2	PD1	PD0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	SCLK2/ $\overline{\text{CTS2}}$	Reserved	RXD2	Reserved	Reserved	HSSI0
0	1	1	SCLK2 output	TXD2(O.D Dis)	Reserved	HSCLK0	HSSO0	Reserved
1	0	0		Reserved				
1	0	1		Reserved				
1	1	0		Reserved				
1	1	1		TXD2(O.D Ena)				

Note 1) Read-modify-write is prohibited for PDCR, PDFC and PDFC2.

Note 2) RXD2, SCLK2 and $\overline{\text{CTS2}}$ input are inputted into the serial bus interface 0 irrespective of a functional setup of a port.

Note 3) HSSI0 input are inputted into the high speed serial channel 0 irrespective of a functional setup of a port.

Note 4) PD4 do not have a register for 3-state/open drain setup.
Moreover, there is no open drain function at the time of an output port.

Figure 3.5.31 Port D register

3.5.9 Port F (PF0 to PF6)

Port F is an 7-bit general-purpose I/O port.

There are the following functions in addition to an I/O port.

- The input function of 8-bit timer 0(TA0IN)
- The output function of 8-bit timer 1(TA1OUT)
- The input function of 8-bit timer 2(TA2IN)
- The output function of 8-bit timer 3(TA3OUT)
- The input function of 8-bit timer 4(TA4IN)
- The output function of 8-bit timer 5(TA5OUT)
- The input function of 8-bit timer 6(TA6IN)
- The input function of external interrupt(INT0 to INT3)

These functions operate by setting the bit concerned of PFCR, PFFC and PFFC2 register. All the bits of PFCR, PFFC and PFFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

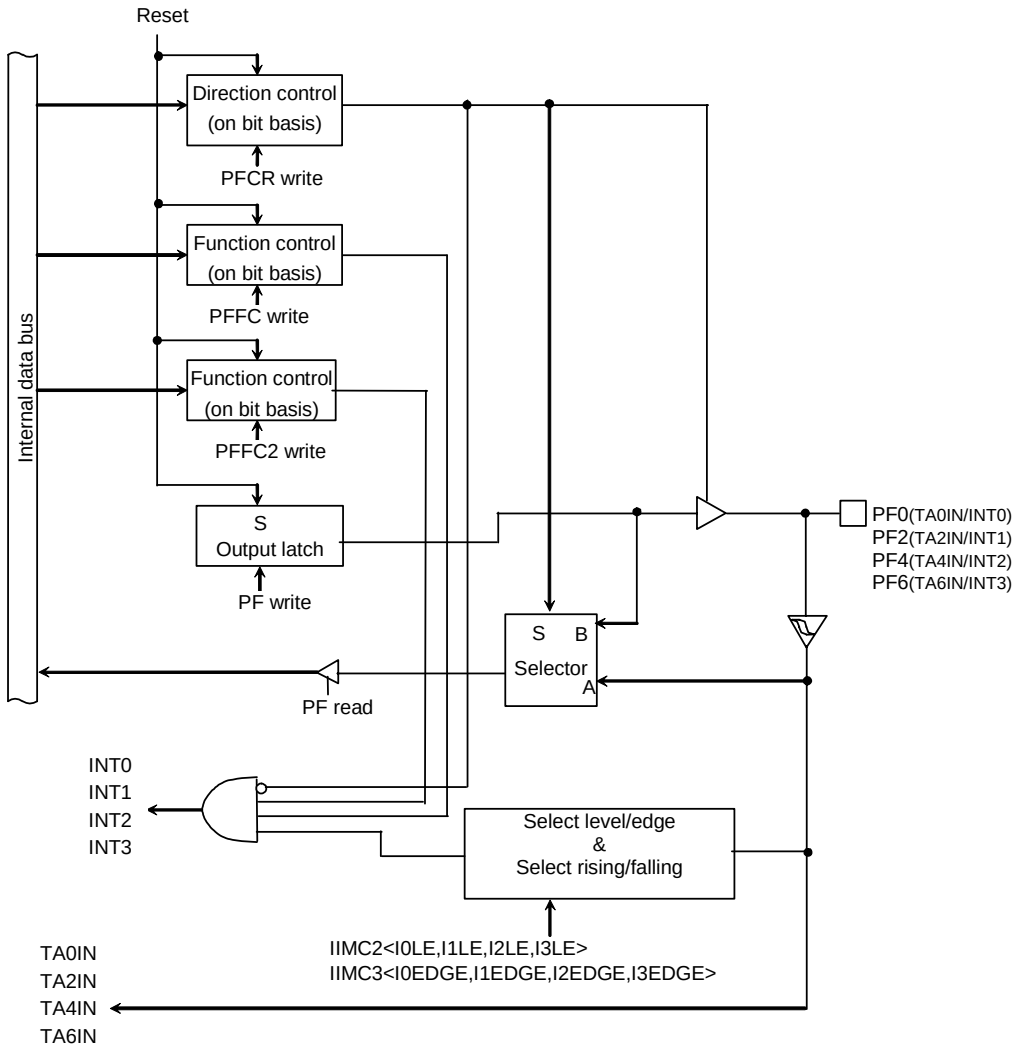


Figure 3.5.32 Port F(PF0,PF2,PF4,PF6)

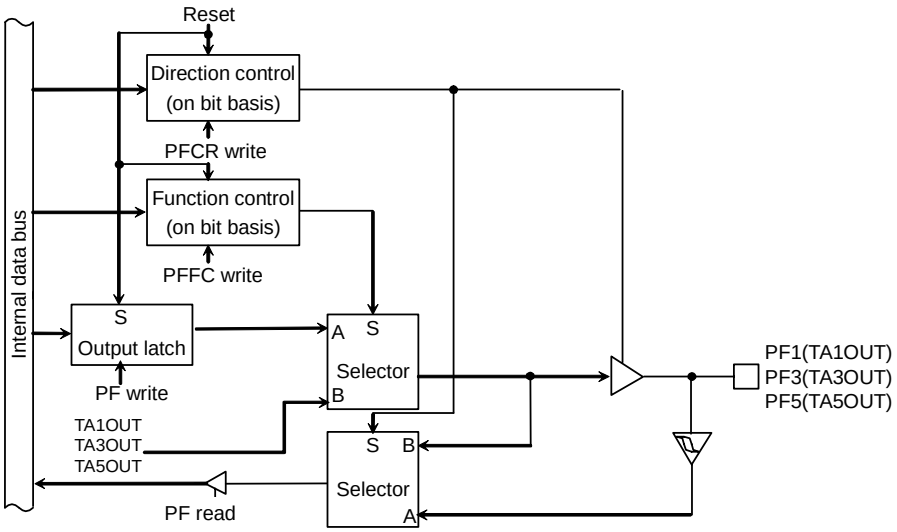


Figure 3.5.33 Port F(PF1,PF3,PF5)

Port F register

		7	6	5	4	3	2	1	0
PF (003CH)	bit Symbol		PF6	PF5	PF4	PF3	PF2	PF1	PF0
	Read/Write		R/W						
	After reset		Data from external port(Output latch register is set to “1”)						

Port F Control register

		7	6	5	4	3	2	1	0
PFCR (003EH)	bit Symbol		PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
	Read/Write		W						
	After reset		0	0	0	0	0	0	0
			Refer to following table						

Port F Function register

		7	6	5	4	3	2	1	0
PFFC (003FH)	bit Symbol		PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
	Read/Write		W						
	After reset		0	0	0	0	0	0	0
	Function		Refer to following table						

Port F Function register 2

		7	6	5	4	3	2	1	0
PFFC2 (003DH)	bit Symbol		PF6F2		PF4F2		PF2F2		PF0F2
	Read/Write		W		W		W		W
	After reset		0		0		0		0
	Function		Refer to following table		Refer to following table		Refer to following table		Refer to following table

Port F function setting

<PFx2>	<PFxF>	<PFxC>	PF6	PF5	PF4	PF3	PF2	PF1	PF0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	TA6IN	Reserved	TA4IN	Reserved	TA2IN	Reserved	TA0IN
0	1	1	Reserved	TA5OUT	Reserved	TA3OUT	Reserved	TA1OUT	Reserved
1	0	0	Reserved		Reserved		Reserved		Reserved
1	0	1	Reserved		Reserved		Reserved		Reserved
1	1	0	INT3		INT2		INT1		INT0
1	1	1	Reserved		Reserved		Reserved		Reserved

Note 1) Read-modify-write is prohibited for PFCR,PFFC and PFFC2.
Note 2) TA0IN, TA2IN, TA4IN and TA6IN input is inputted into the 8-bit timer TMRA0 to TMRA6 irrespective of a functional setup of a port.

Figure 3.5.34 Port F register

3.5.10 Port J (PJ0 to PJ7)

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

There are the following functions in addition to an I/O port.

- The output function of 16-bit timer 0(TB0OUT0, TB0OUT1)
- The output function of 16-bit timer 1(TB1OUT0, TB1OUT1)
- The output function of 16-bit timer 2(TB2OUT0, TB2OUT1)
- The output function of 16-bit timer 3(TB3OUT0, TB3OUT1)
- The output function of 16-bit timer 4(TB4OUT0, TB4OUT1)
- The output function of 16-bit timer 5(TB5OUT0, TB5OUT1)

These functions operate by setting the bit concerned of PJCR, PJFC and PJFC2 register. All the bits of PJCR, PJFC and PJFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

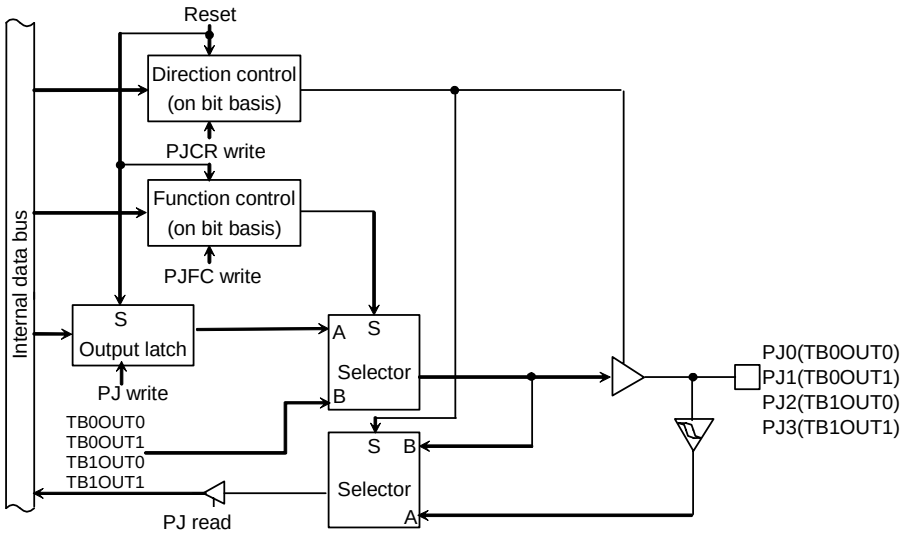


Figure 3.5.35 Port J(PJ0,PJ1,PJ2,PJ3)

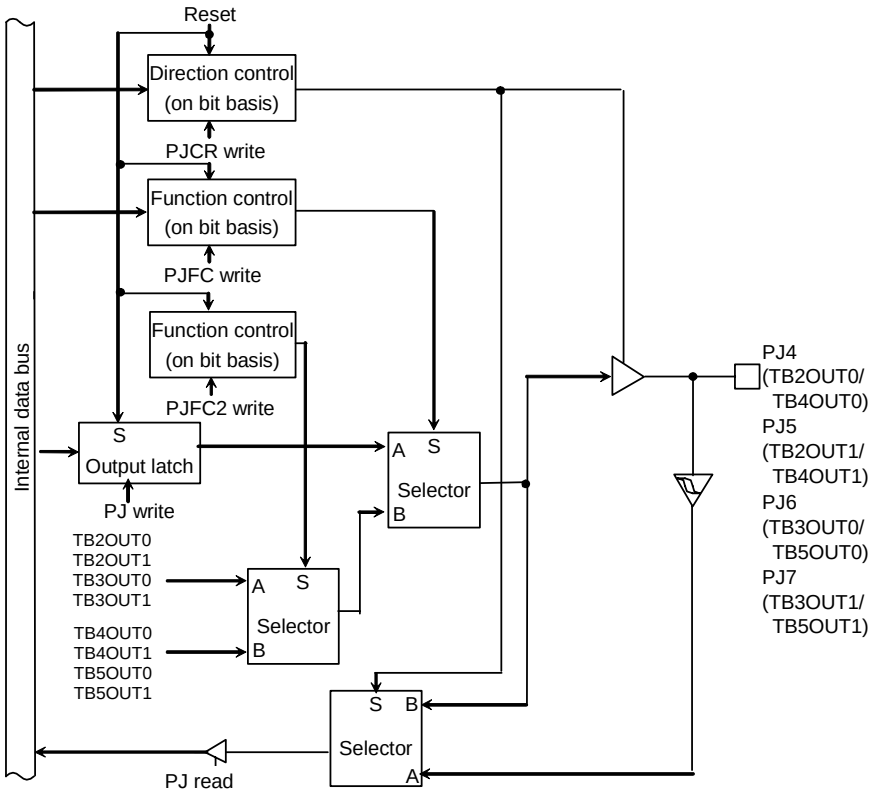


Figure 3.5.36 Port J(PJ4,PJ5,PJ6,PJ7)

Port J register

		7	6	5	4	3	2	1	0
PJ (004CH)	bit Symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
	Read/Write	R/W							
	After reset	Data from external port(Output latch register is set to “1”)							

Port J Control register

		7	6	5	4	3	2	1	0
PJCR (004EH)	bit Symbol	PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
	Read/Write	W							
	After reset	0							
		Refer to following table							

Port J Function register

		7	6	5	4	3	2	1	0
PJFC (004FH)	bit Symbol	PJ7F	PJ6F	PJ5F	PJ4F	PJ3F	PJ2F	PJ1F	PJ0F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Refer to following table							

Port J Function register 2

		7	6	5	4	3	2	1	0
PJFC2 (004DH)	bit Symbol	PJ7F2	PJ6F2	PJ5F2	PJ4F2				
	Read/Write	W							
	After reset	0	0	0	0				
	Function	Refer to following table							

Port J function setting

<PJx2>	<PJxF>	<PJxC>	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	1	1	TB3OUT1	TB3OUT0	TB2OUT1	TB2OUT0	TB1OUT1	TB1OUT0	TB0OUT1	TB0OUT0
1	0	0	Reserved	Reserved	Reserved	Reserved				
1	0	1	Reserved	Reserved	Reserved	Reserved				
1	1	0	Reserved	Reserved	Reserved	Reserved				
1	1	1	TB5OUT1	TB5OUT0	TB4OUT1	TB4OUT0				

Note) Read-modify-write is prohibited for PJCR,PJFC and PJFC2.

Figure 3.5.37 Port J register

3.5.11 Port K (PK0 to PK7)

Port K are a port only for inputs.

There are the following functions in addition to an input port.

- The input function of 16-bit timer 0(TB0IN0, TB0IN1)
- The input function of 16-bit timer 1(TB1IN0, TB1IN1)
- The input function of 16-bit timer 2(TB2IN0, TB2IN1)
- The input function of 16-bit timer 3(TB3IN0, TB3IN1)
- The input function of external interrupt(INT4 to INTB)

These functions operate by setting the bit concerned of PKFC and PKFC2 register. All the bits of PKFC and PKFC2 are cleared to "0" by the reset action, and all bits serve as an input port.

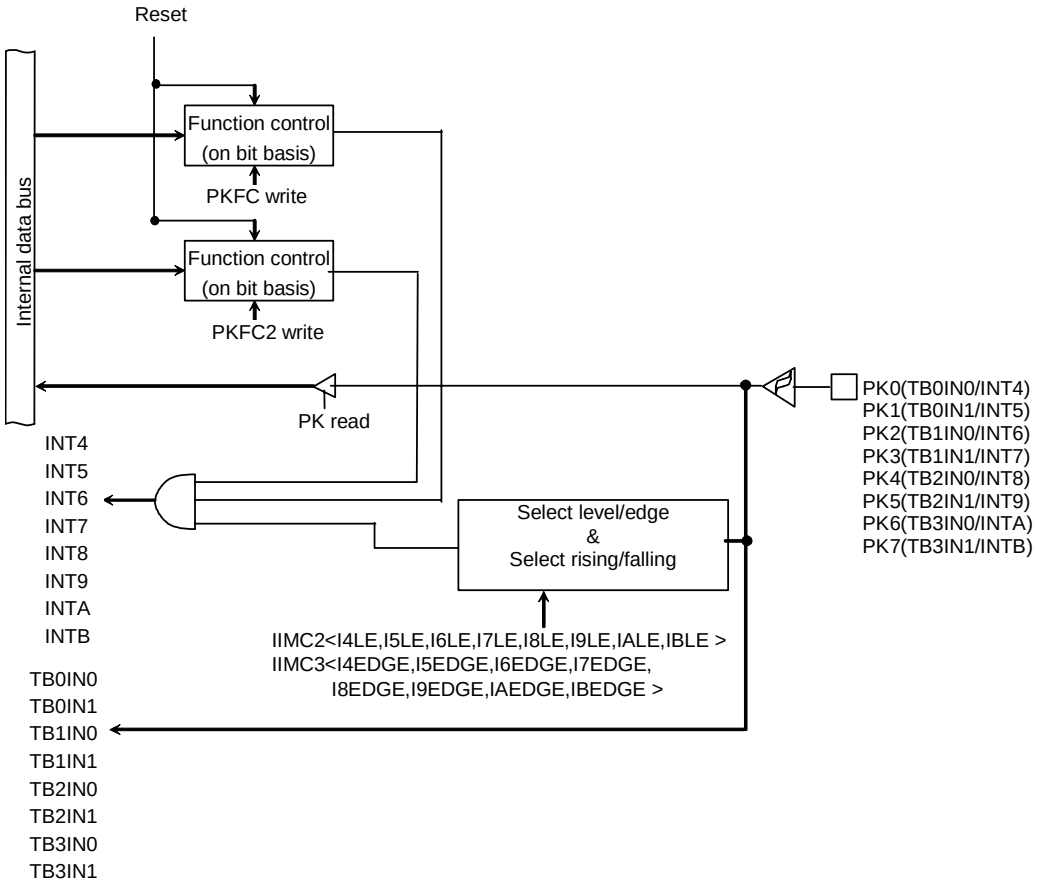


Figure 3.5.38 Port K(PK0 to PK7)

Port K register

PK (0050H)

	7	6	5	4	3	2	1	0
bit Symbol	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
Read/Write	R							
After reset	Data from external port							

Port K Function register

PKFC (0053H)

	7	6	5	4	3	2	1	0
bit Symbol	PK7F	PK6F	PK5F	PK4F	PK3F	PK2F	PK1F	PK0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port K Function register 2

PKFC2 (0051H)

	7	6	5	4	3	2	1	0
bit Symbol	PK7F	PK6F	PK5F	PK4F	PK3F	PK2F	PK1F	PK0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port K function setting

<PKxF2>	<PKxF>	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	1	TB3IN1	TB3IN0	TB2IN1	TB2IN0	TB1IN1	TB1IN0	TB0IN1	TB0IN0
1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1	1	INTB	INTA	INT9	INT8	INT7	INT6	INT5	INT4

Note 1) Read-modify-write is prohibited for PKFC and PKFC2.

Note 2) TB0IN0/1, TB1IN0/1, TB2IN0/1 and TB3IN0/1 input is inputted into the 8-bit timer TMRB0 to TMRB3 irrespective of a functional setup of a port.

Note 3) In setting (TB0IN0 to TB3IN1) of setting (INT4 to INTB) of <PKxF2>=1 and <PKxF>=1 and <PKxF2>=0 and <PKxF>=1, the operation of the HALT release is different.
Refer to Table 3.3.6 HALT release source of 3.3.6 standby control part and the operation of the HALT release for details.

Figure 3.5.39 Port K register

3.5.12 Port L (PL0 to PL7)

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

PL1 can be set as an open drain output.

There are the following functions in addition to an I/O port.

- The output function of pattern generator 0(PG00 to PG03)
- The output function of pattern generator 1(PG10 to PG13)
- The I/O function of the serial cannel 3(RXD3, TXD3, SCLK3/$\overline{\text{CTS3}}$)
- The output function of 8-bit timer 7(TA7OUT)
- The I/O function of the high speed channel 1(HSSI1, HSSO1, HSCLK1)

These functions operate by setting the bit concerned of PLCR, PLFC and PLFC2 register. All the bits of PLCR, PLFC and PLFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

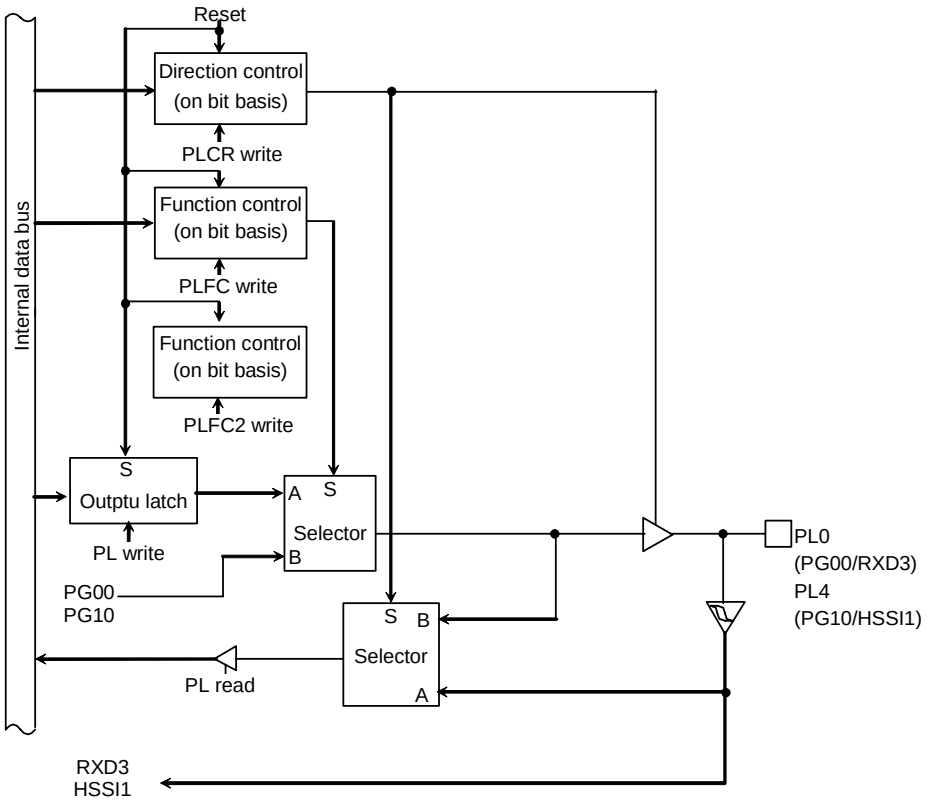


Figure 3.5.40 Port L(PL0,PL4)

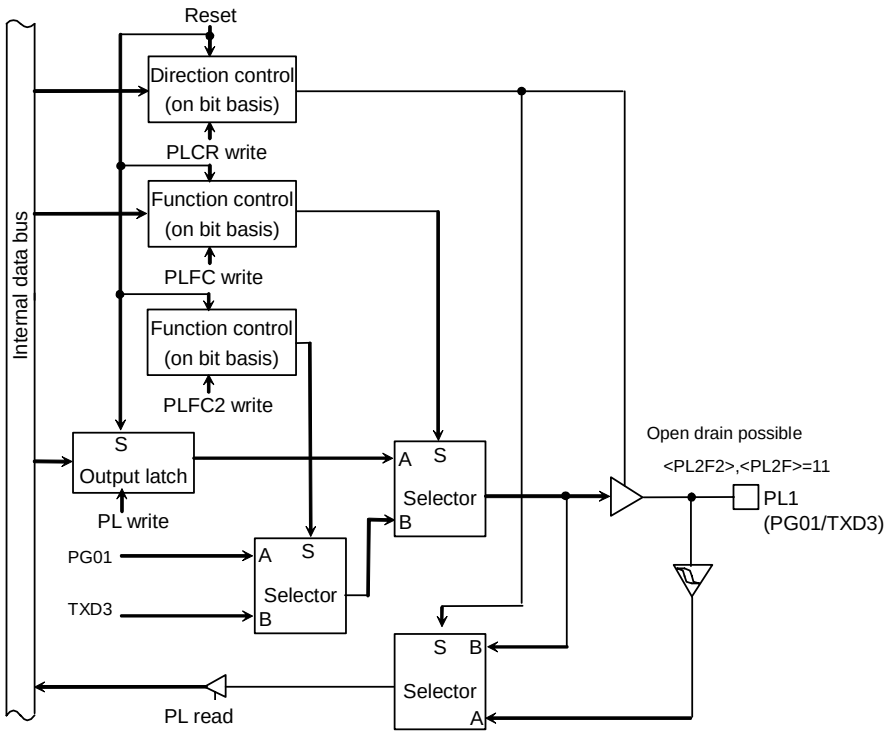


Figure 3.5.41 Port L(PL1)

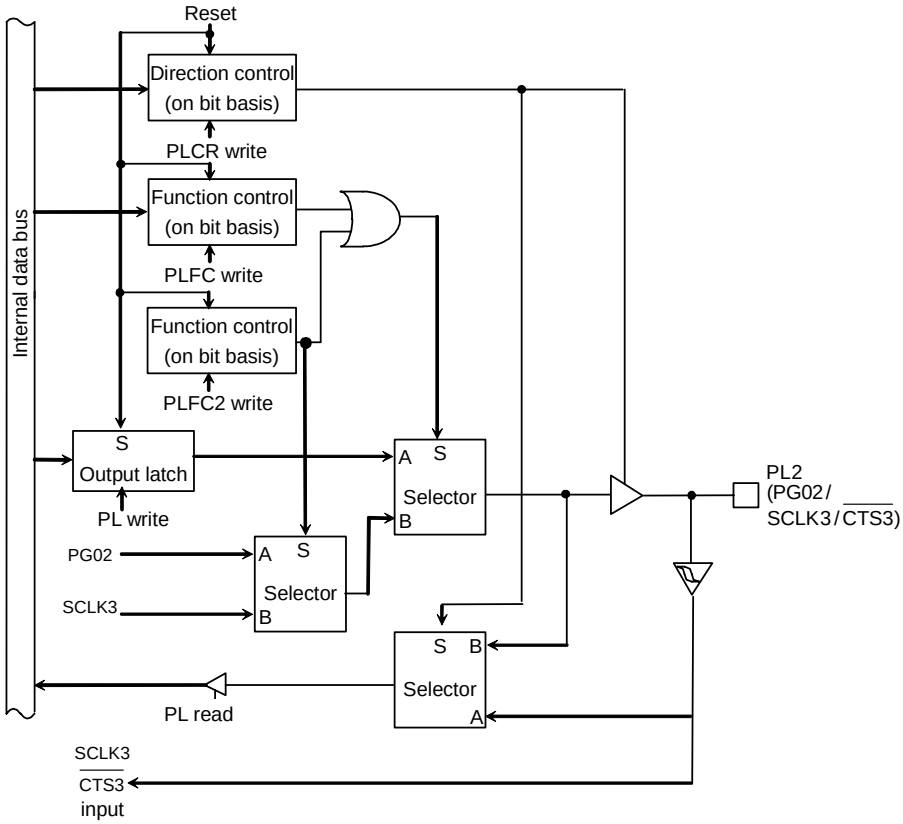


Figure 3.5.42 Port L(PL2)

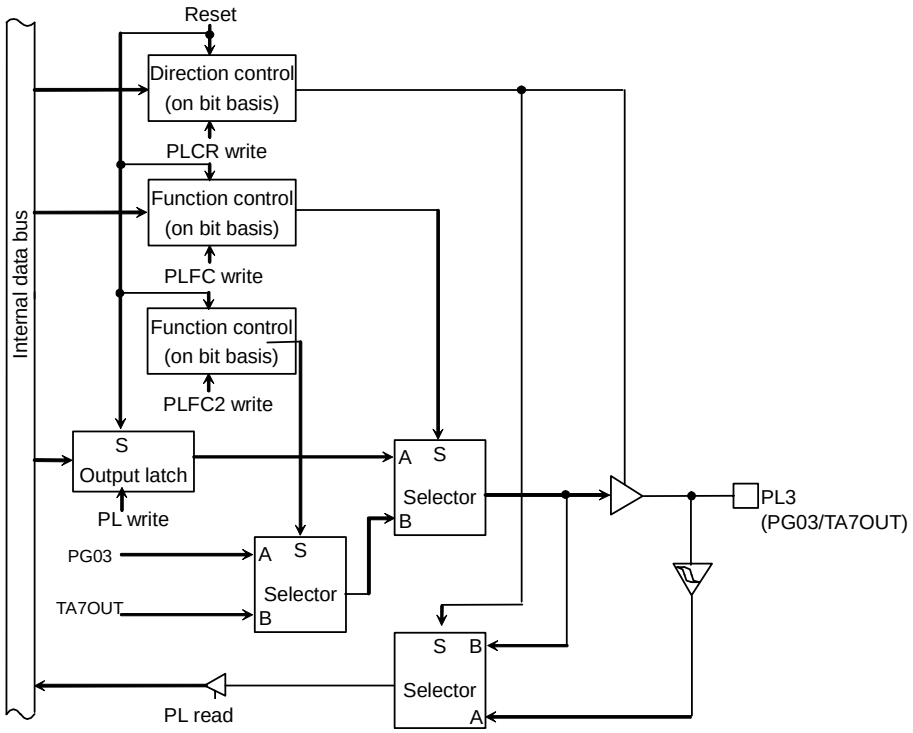


Figure 3.5.43 Port L(PL3)

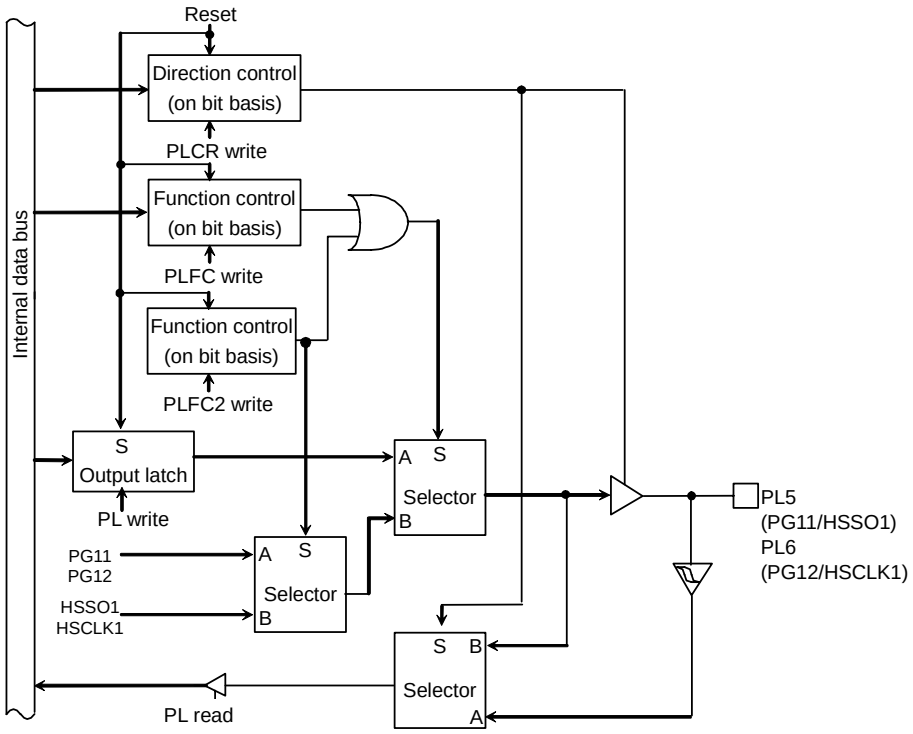


Figure 3.5.44 Port L(PL5,PL6)

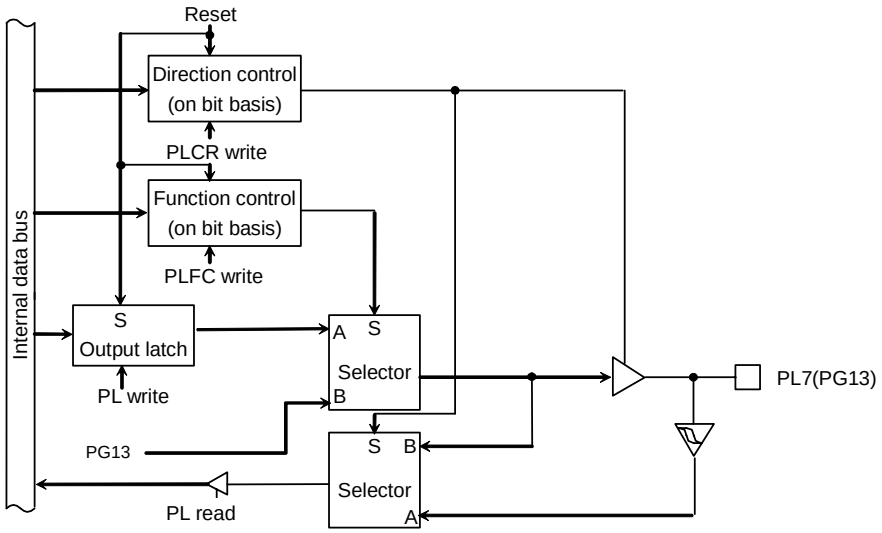


Figure 3.5.45 Port L(PL7)

Port L register

PL (0054H)

	7	6	5	4	3	2	1	0
bit Symbol	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
Read/Write	R/W							
After reset	Data from external port(Output latch register is set to “1”)							

Port L Control register

PLCR (0056H)

	7	6	5	4	3	2	1	0
bit Symbol	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
	Refer to following table							

Port L Function register

PLFC (0057H)

	7	6	5	4	3	2	1	0
bit Symbol	PL7F	PL6F	PL5F	PL4F	PL3F	PL2F	PL1F	PL0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port L Function register 2

PLFC2 (0055H)

	7	6	5	4	3	2	1	0
bit Symbol		PL6F2	PL5F2	PL4F2	PL3F2	PL2F2	PL1F2	PL0F2
Read/Write		W						
After reset		0	0	0	0	0	0	0
Function		Refer to following table						

Port L function setting

<PLxF2>	<PLxF>	<PLxC>	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	1	1	PG13	PG12	PG11	PG10	PG03	PG02	PG01	PG00
1	0	0		Reserved	Reserved	HSSI1	Reserved	SCLK3/ $\overline{\text{CTS3}}$	Reserved	RXD3
1	0	1		HSCLK1	HSSO1	Reserved	Reserved	SCLK3	TXD3 (O.D Dis)	Reserved
1	1	0		Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1	1	1		Reserved	Reserved	Reserved	TA7OUT	Reserved	TXD3 (O.D Ena)	Reserved

Note 1) Read-modify-write is prohibited for PLCR, PLFC and PLFC2.

Note 2) RXD3, SCLK3 and $\overline{\text{CTS3}}$ input are inputted into the serial bus interface 3 irrespective of a functional setup of a port.

Note 3) HSSI1 input are inputted into the high speed serial channel 0 irrespective of a functional setup of a port.

Note 4) PL1 does not have a register for 3-state/open drain setup.
Moreover, there is no open drain function at the time of an output port.

Figure 3.5.46 Port L register

3.5.8 Port D(PD0 to PD5)

Port D is an 6-bit general-purpose I/O port.

PD4 can be set as an open drain output.

There are the following functions in addition to an I/O port.

- The I/O function of the serial cannel 2(RXD2, TXD2, SCLK2/$\overline{\text{CTS2}}$)
- The I/O function of the high speed channel 0(HSSI0, HSSO0, HSCLK0)

These functions operate by setting the bit concerned of PDCR, PDFC and PDFC2 register. All the bits of PDCR, PDFC and PDFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

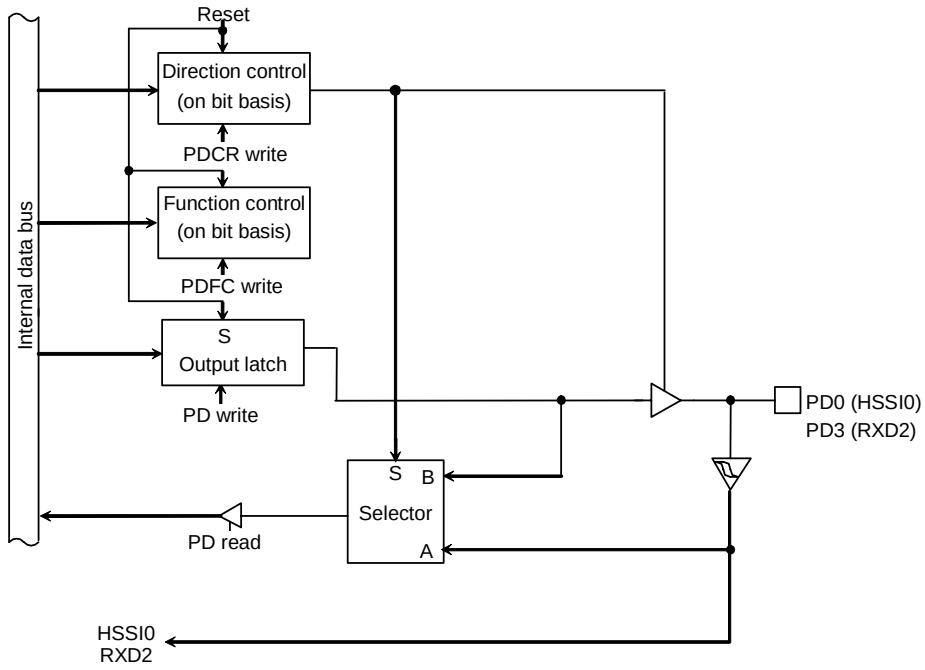


Figure 3.5.27 Port D(PD0,PD3)

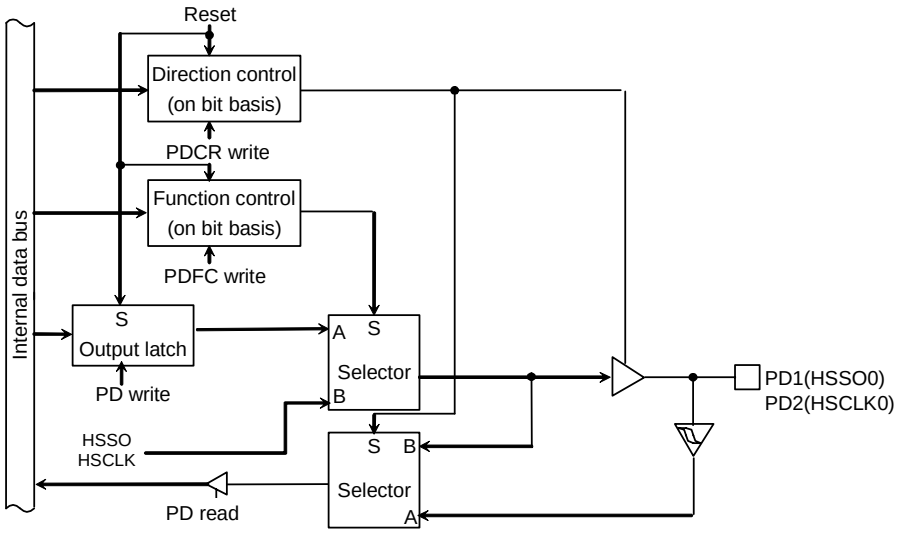


Figure 3.5.28 Port D(PD1,PD2)

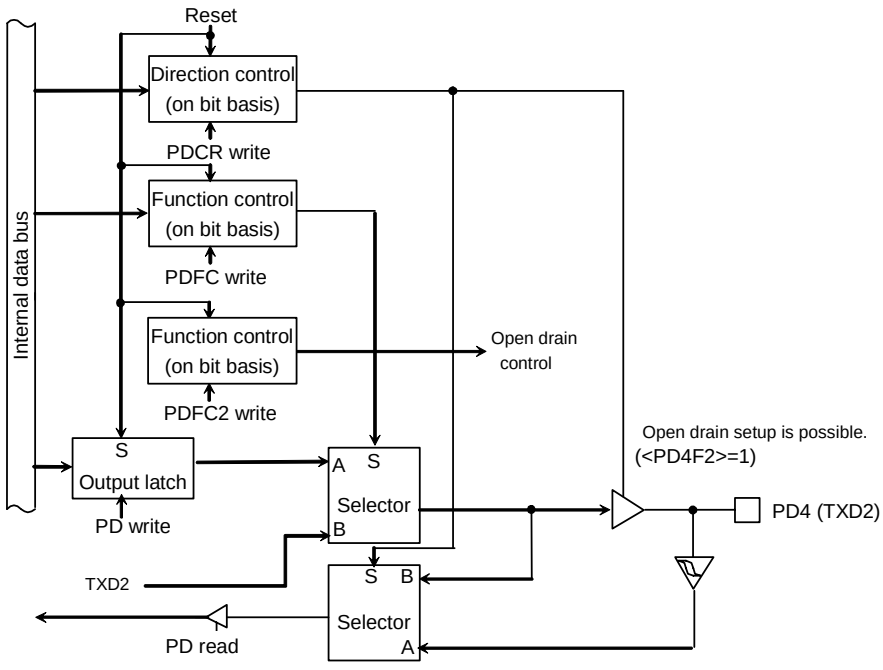


Figure 3.5.29 Port D(PD4)

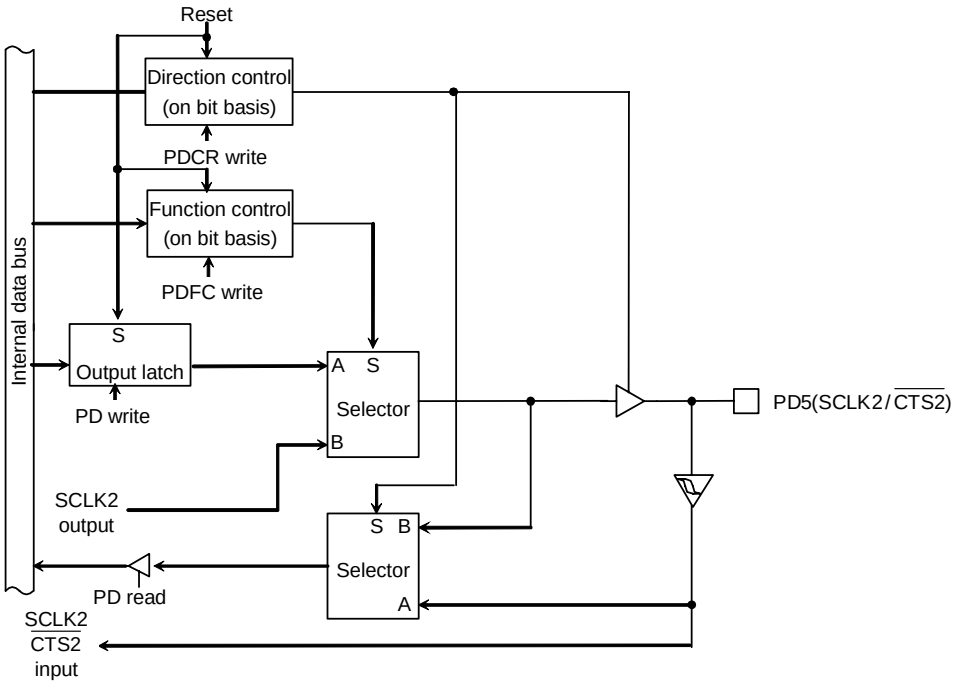


Figure 3.5.30 Port D(PD5)

Port D register

PD (0034H)

	7	6	5	4	3	2	1	0
bit Symbol			PD5	PD4	PD3	PD2	PD1	PD0
Read/Write			R/W					
After reset			Data from external port(Output latch register is set to “1”)					

Port D Control register

PDCR (0036H)

	7	6	5	4	3	2	1	0
bit Symbol			PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
Read/Write			W					
After reset			0	0	0	0	0	0
			Refer to following table					

Port D Function register

PDFC (0037H)

	7	6	5	4	3	2	1	0
bit Symbol			PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
Read/Write			W					
After reset			0	0	0	0	0	0
Function			Refer to following table					

Port D Function register 2

PDFC2 (0035H)

	7	6	5	4	3	2	1	0
bit Symbol				PD4F2				
Read/Write				W				
After reset				0				
Function				Refer to following table				

Port D function setting

<PDxF2>	<PDxF>	<PDxC>	PD5	PD4	PD3	PD2	PD1	PD0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	SCLK2/ $\overline{\text{CTS2}}$	Reserved	RXD2	Reserved	Reserved	HSSI0
0	1	1	SCLK2 output	TXD2(O.D Dis)	Reserved	HSCLK0	HSSO0	Reserved
1	0	0		Reserved				
1	0	1		Reserved				
1	1	0		Reserved				
1	1	1		TXD2(O.D Ena)				

Note 1) Read-modify-write is prohibited for PDCR, PDFC and PDFC2.

Note 2) RXD2, SCLK2 and $\overline{\text{CTS2}}$ input are inputted into the serial bus interface 0 irrespective of a functional setup of a port.

Note 3) HSSI0 input are inputted into the high speed serial channel 0 irrespective of a functional setup of a port.

Note 4) PD4 do not have a register for 3-state/open drain setup.
Moreover, there is no open drain function at the time of an output port.

Figure 3.5.31 Port D register

3.5.9 Port F (PF0 to PF6)

Port F is an 7-bit general-purpose I/O port.

There are the following functions in addition to an I/O port.

- The input function of 8-bit timer 0(TA0IN)
- The output function of 8-bit timer 1(TA1OUT)
- The input function of 8-bit timer 2(TA2IN)
- The output function of 8-bit timer 3(TA3OUT)
- The input function of 8-bit timer 4(TA4IN)
- The output function of 8-bit timer 5(TA5OUT)
- The input function of 8-bit timer 6(TA6IN)
- The input function of external interrupt(INT0 to INT3)

These functions operate by setting the bit concerned of PFCR, PFFC and PFFC2 register. All the bits of PFCR, PFFC and PFFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

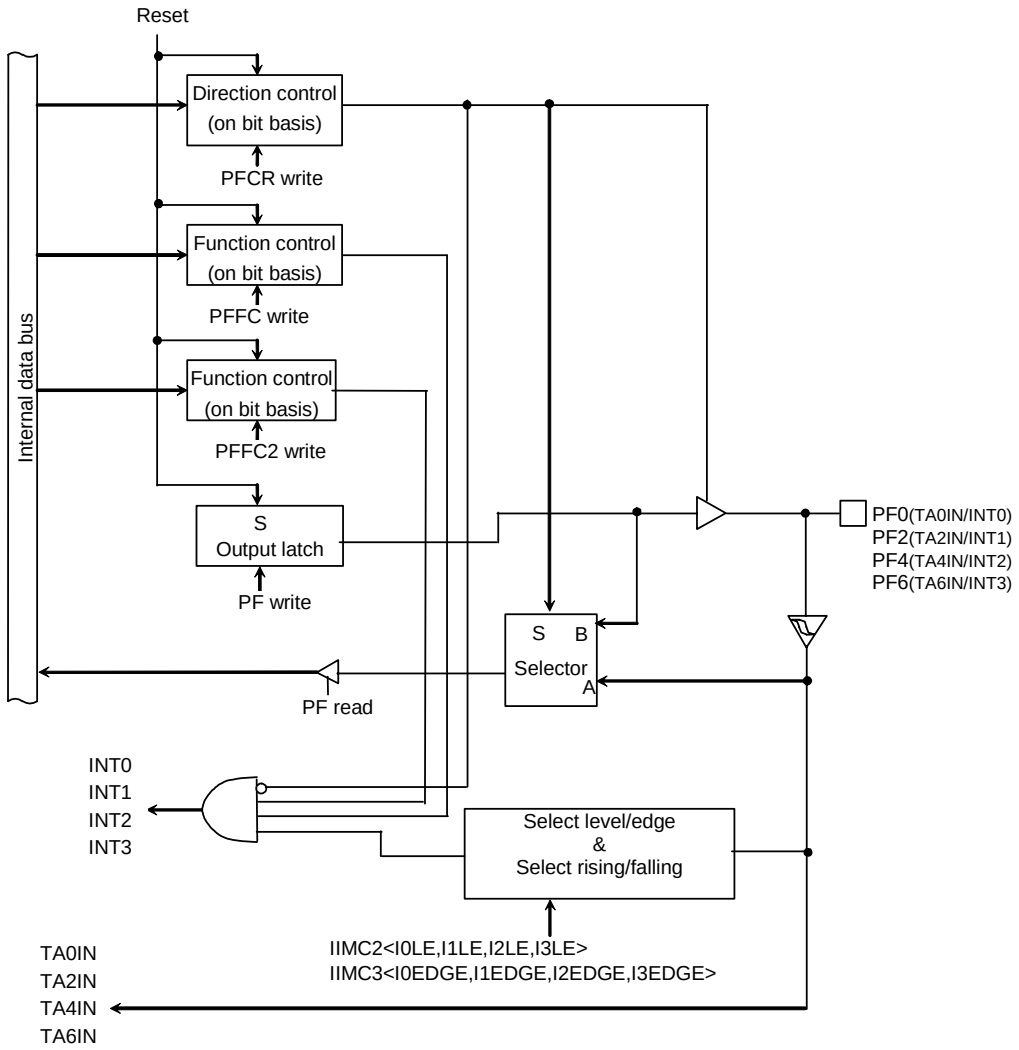


Figure 3.5.32 Port F(PF0,PF2,PF4,PF6)

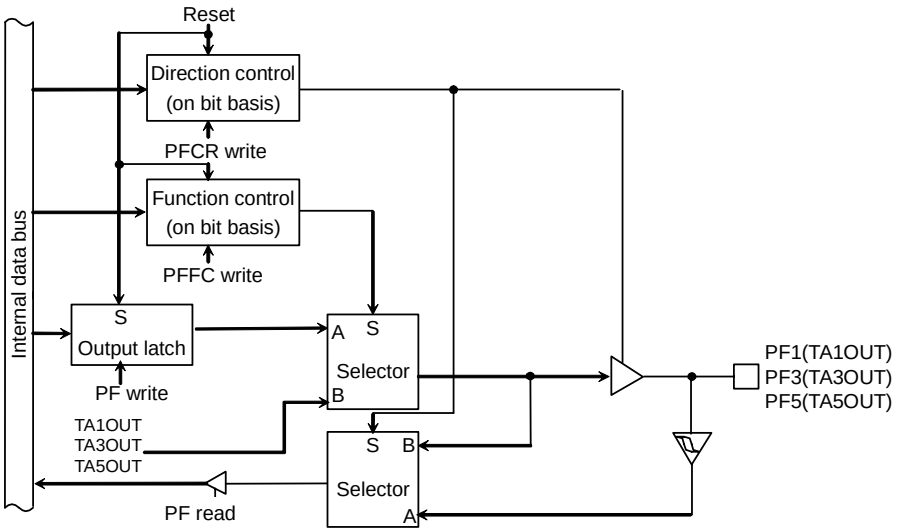


Figure 3.5.33 Port F(PF1,PF3,PF5)

Port F register

		7	6	5	4	3	2	1	0
PF (003CH)	bit Symbol		PF6	PF5	PF4	PF3	PF2	PF1	PF0
	Read/Write		R/W						
	After reset		Data from external port(Output latch register is set to “1”)						

Port F Control register

		7	6	5	4	3	2	1	0
PFCR (003EH)	bit Symbol		PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
	Read/Write		W						
	After reset		0	0	0	0	0	0	0
			Refer to following table						

Port F Function register

		7	6	5	4	3	2	1	0
PFFC (003FH)	bit Symbol		PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
	Read/Write		W						
	After reset		0	0	0	0	0	0	0
	Function		Refer to following table						

Port F Function register 2

		7	6	5	4	3	2	1	0
PFFC2 (003DH)	bit Symbol		PF6F2		PF4F2		PF2F2		PF0F2
	Read/Write		W		W		W		W
	After reset		0		0		0		0
	Function		Refer to following table		Refer to following table		Refer to following table		Refer to following table

Port F function setting

<PFx2>	<PFxF>	<PFxC>	PF6	PF5	PF4	PF3	PF2	PF1	PF0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	TA6IN	Reserved	TA4IN	Reserved	TA2IN	Reserved	TA0IN
0	1	1	Reserved	TA5OUT	Reserved	TA3OUT	Reserved	TA1OUT	Reserved
1	0	0	Reserved		Reserved		Reserved		Reserved
1	0	1	Reserved		Reserved		Reserved		Reserved
1	1	0	INT3		INT2		INT1		INT0
1	1	1	Reserved		Reserved		Reserved		Reserved

Note 1) Read-modify-write is prohibited for PFCR,PFFC and PFFC2.
Note 2) TA0IN, TA2IN, TA4IN and TA6IN input is inputted into the 8-bit timer TMRA0 to TMRA6 irrespective of a functional setup of a port.

Figure 3.5.34 Port F register

3.5.10 Port J (PJ0 to PJ7)

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

There are the following functions in addition to an I/O port.

- The output function of 16-bit timer 0(TB0OUT0, TB0OUT1)
- The output function of 16-bit timer 1(TB1OUT0, TB1OUT1)
- The output function of 16-bit timer 2(TB2OUT0, TB2OUT1)
- The output function of 16-bit timer 3(TB3OUT0, TB3OUT1)
- The output function of 16-bit timer 4(TB4OUT0, TB4OUT1)
- The output function of 16-bit timer 5(TB5OUT0, TB5OUT1)

These functions operate by setting the bit concerned of PJCR, PJFC and PJFC2 register. All the bits of PJCR, PJFC and PJFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

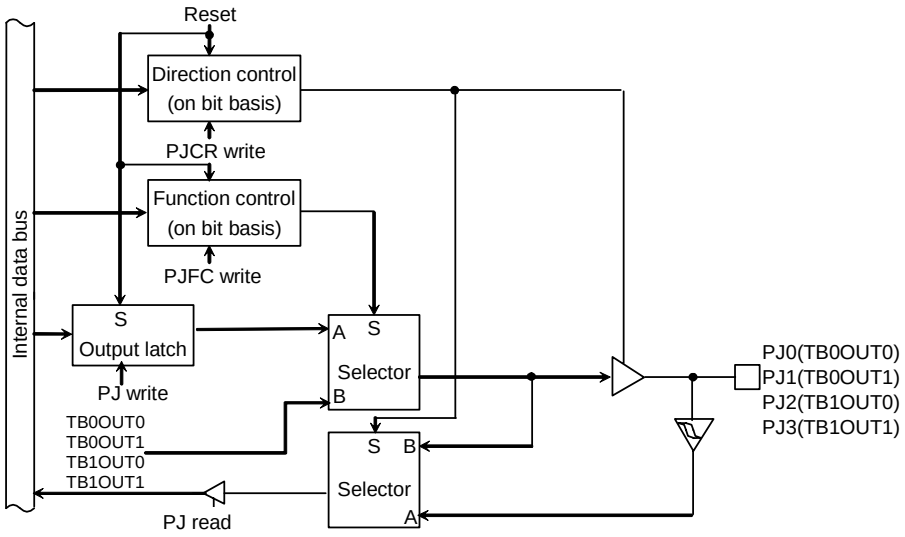


Figure 3.5.35 Port J(PJ0,PJ1,PJ2,PJ3)

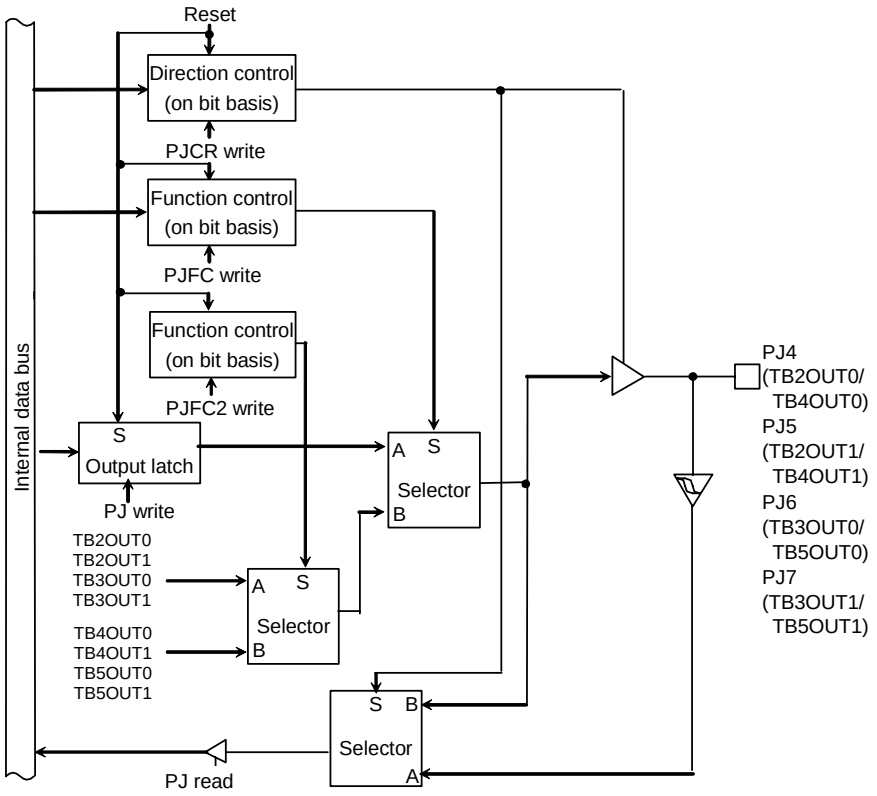


Figure 3.5.36 Port J(PJ4,PJ5,PJ6,PJ7)

Port J register

		7	6	5	4	3	2	1	0
PJ (004CH)	bit Symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
	Read/Write	R/W							
	After reset	Data from external port(Output latch register is set to “1”)							

Port J Control register

		7	6	5	4	3	2	1	0
PJCR (004EH)	bit Symbol	PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
	Read/Write	W							
	After reset	0							
		Refer to following table							

Port J Function register

		7	6	5	4	3	2	1	0
PJFC (004FH)	bit Symbol	PJ7F	PJ6F	PJ5F	PJ4F	PJ3F	PJ2F	PJ1F	PJ0F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Refer to following table							

Port J Function register 2

		7	6	5	4	3	2	1	0
PJFC2 (004DH)	bit Symbol	PJ7F2	PJ6F2	PJ5F2	PJ4F2				
	Read/Write	W							
	After reset	0	0	0	0				
	Function	Refer to following table							

Port J function setting

<PJx2>	<PJxF>	<PJxC>	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	1	1	TB3OUT1	TB3OUT0	TB2OUT1	TB2OUT0	TB1OUT1	TB1OUT0	TB0OUT1	TB0OUT0
1	0	0	Reserved	Reserved	Reserved	Reserved				
1	0	1	Reserved	Reserved	Reserved	Reserved				
1	1	0	Reserved	Reserved	Reserved	Reserved				
1	1	1	TB5OUT1	TB5OUT0	TB4OUT1	TB4OUT0				

Note) Read-modify-write is prohibited for PJCR,PJFC and PJFC2.

Figure 3.5.37 Port J register

3.5.11 Port K (PK0 to PK7)

Port K are a port only for inputs.

There are the following functions in addition to an input port.

- The input function of 16-bit timer 0(TB0IN0, TB0IN1)
- The input function of 16-bit timer 1(TB1IN0, TB1IN1)
- The input function of 16-bit timer 2(TB2IN0, TB2IN1)
- The input function of 16-bit timer 3(TB3IN0, TB3IN1)
- The input function of external interrupt(INT4 to INTB)

These functions operate by setting the bit concerned of PKFC and PKFC2 register. All the bits of PKFC and PKFC2 are cleared to "0" by the reset action, and all bits serve as an input port.

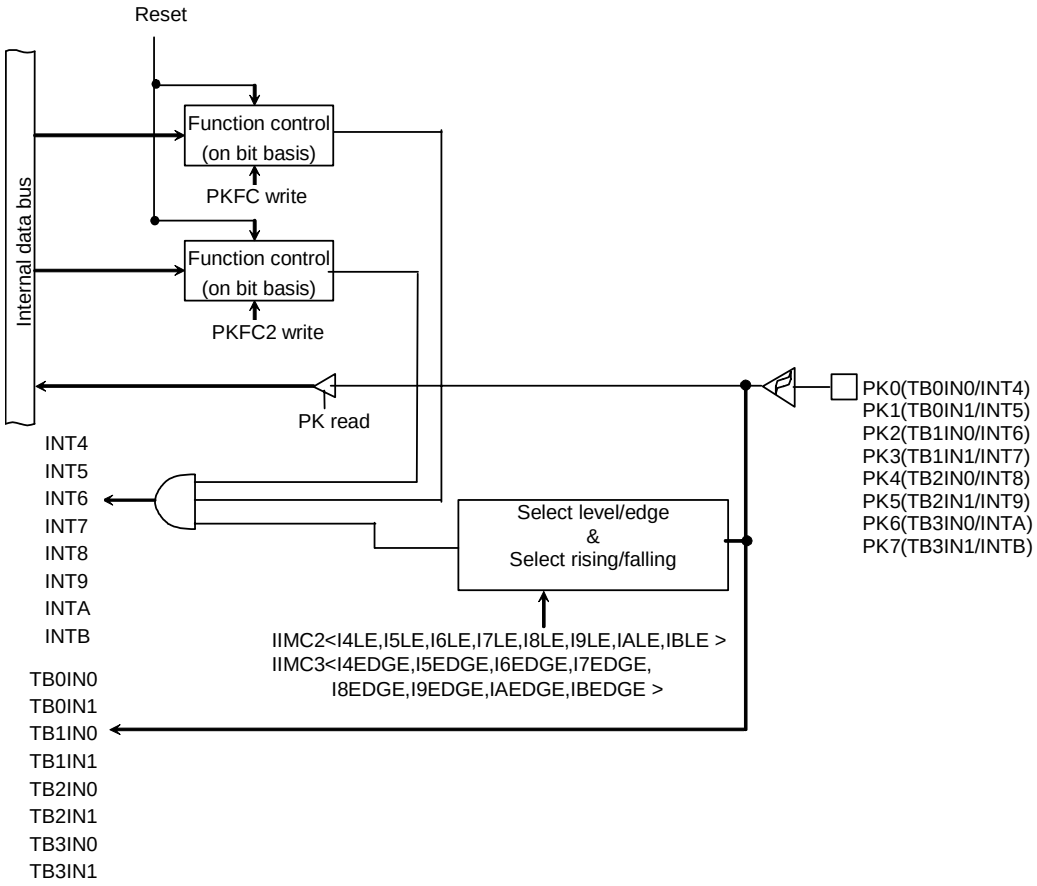


Figure 3.5.38 Port K(PK0 to PK7)

Port K register

PK (0050H)

	7	6	5	4	3	2	1	0
bit Symbol	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
Read/Write	R							
After reset	Data from external port							

Port K Function register

PKFC (0053H)

	7	6	5	4	3	2	1	0
bit Symbol	PK7F	PK6F	PK5F	PK4F	PK3F	PK2F	PK1F	PK0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port K Function register 2

PKFC2 (0051H)

	7	6	5	4	3	2	1	0
bit Symbol	PK7F	PK6F	PK5F	PK4F	PK3F	PK2F	PK1F	PK0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port K function setting

<PKxF2>	<PKxF>	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	1	TB3IN1	TB3IN0	TB2IN1	TB2IN0	TB1IN1	TB1IN0	TB0IN1	TB0IN0
1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1	1	INTB	INTA	INT9	INT8	INT7	INT6	INT5	INT4

Note 1) Read-modify-write is prohibited for PKFC and PKFC2.

Note 2) TB0IN0/1, TB1IN0/1, TB2IN0/1 and TB3IN0/1 input is inputted into the 8-bit timer TMRB0 to TMRB3 irrespective of a functional setup of a port.

Note 3) In setting (TB0IN0 to TB3IN1) of setting (INT4 to INTB) of <PKxF2>=1 and <PKxF>=1 and <PKxF2>=0 and <PKxF>=1, the operation of the HALT release is different.
Refer to Table 3.3.6 HALT release source of 3.3.6 standby control part and the operation of the HALT release for details.

Figure 3.5.39 Port K register

3.5.12 Port L (PL0 to PL7)

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

PL1 can be set as an open drain output.

There are the following functions in addition to an I/O port.

- The output function of pattern generator 0(PG00 to PG03)
- The output function of pattern generator 1(PG10 to PG13)
- The I/O function of the serial cannel 3(RXD3, TXD3, SCLK3/$\overline{\text{CTS3}}$)
- The output function of 8-bit timer 7(TA7OUT)
- The I/O function of the high speed channel 1(HSSI1, HSSO1, HSCLK1)

These functions operate by setting the bit concerned of PLCR, PLFC and PLFC2 register. All the bits of PLCR, PLFC and PLFC2 are cleared to "0" by the reset action, and all bits serve as an input port. Moreover, the output latch of all bit is set to "1".

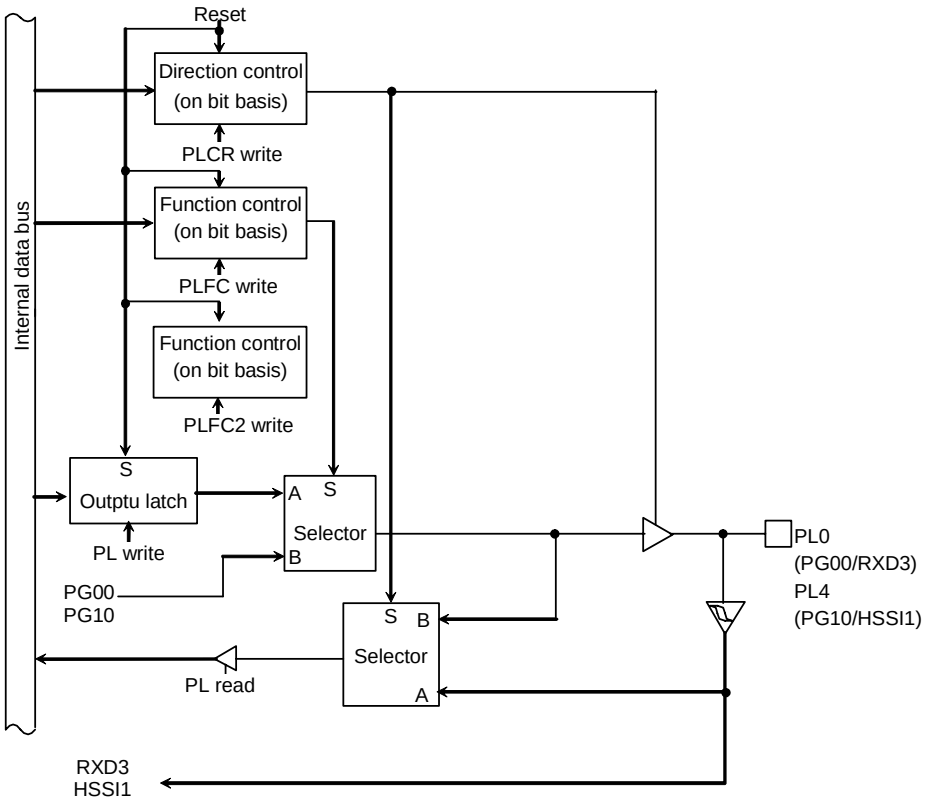


Figure 3.5.40 Port L(PL0,PL4)

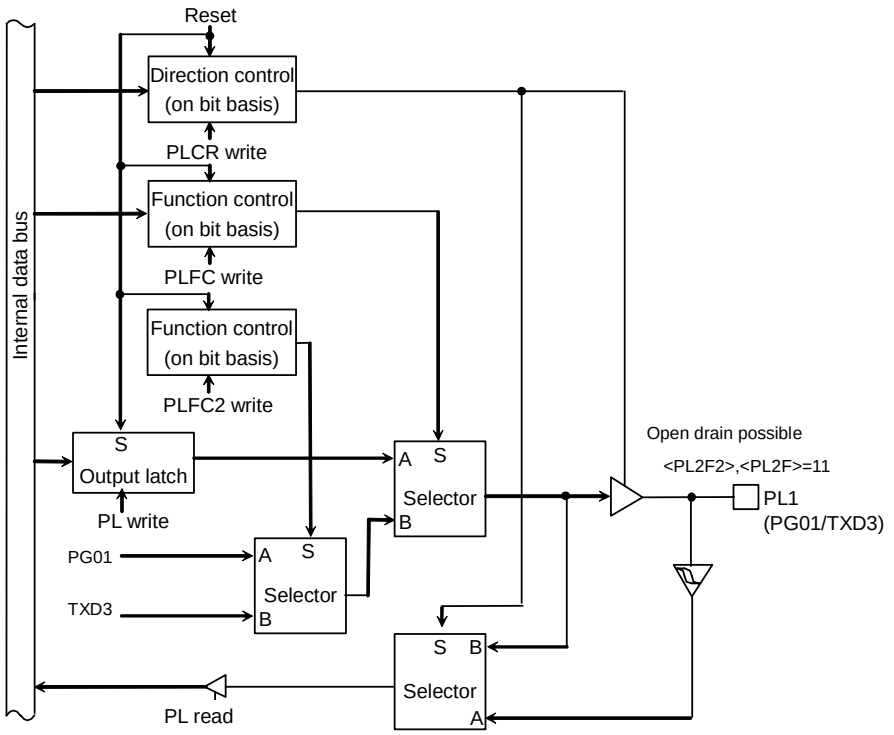


Figure 3.5.41 Port L(PL1)

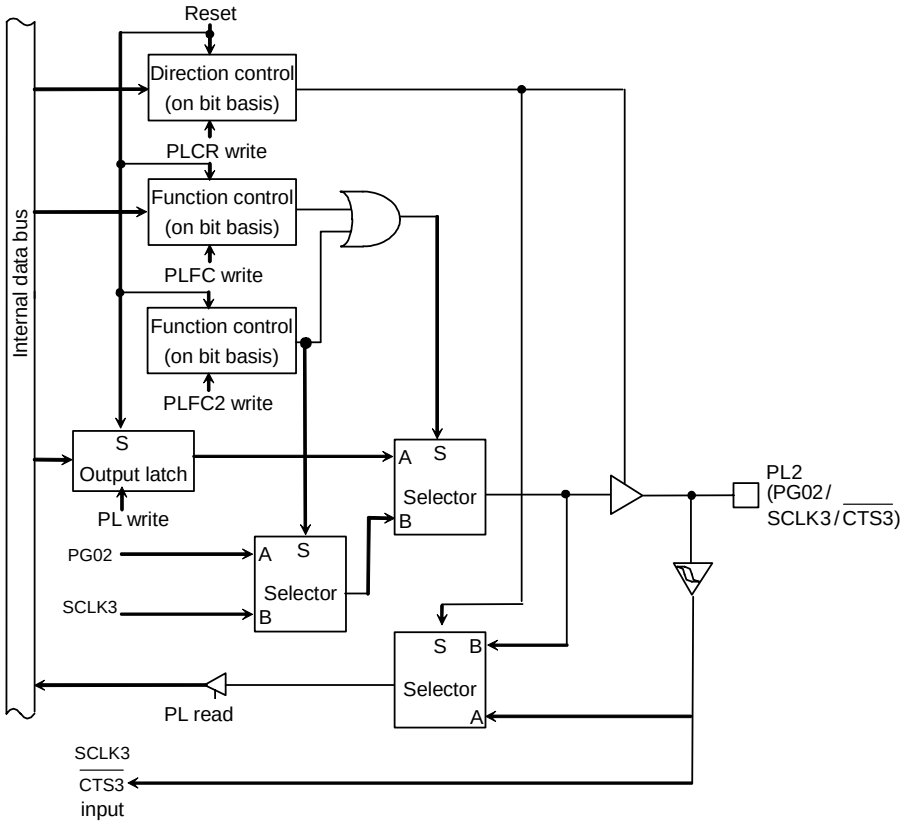


Figure 3.5.42 Port L(PL2)

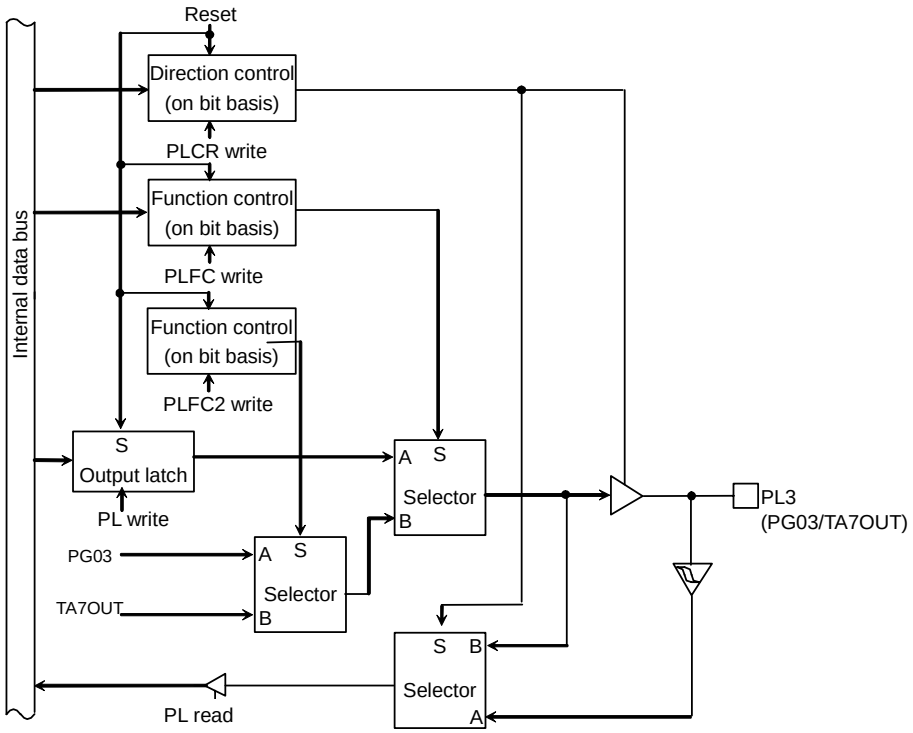


Figure 3.5.43 Port L(PL3)

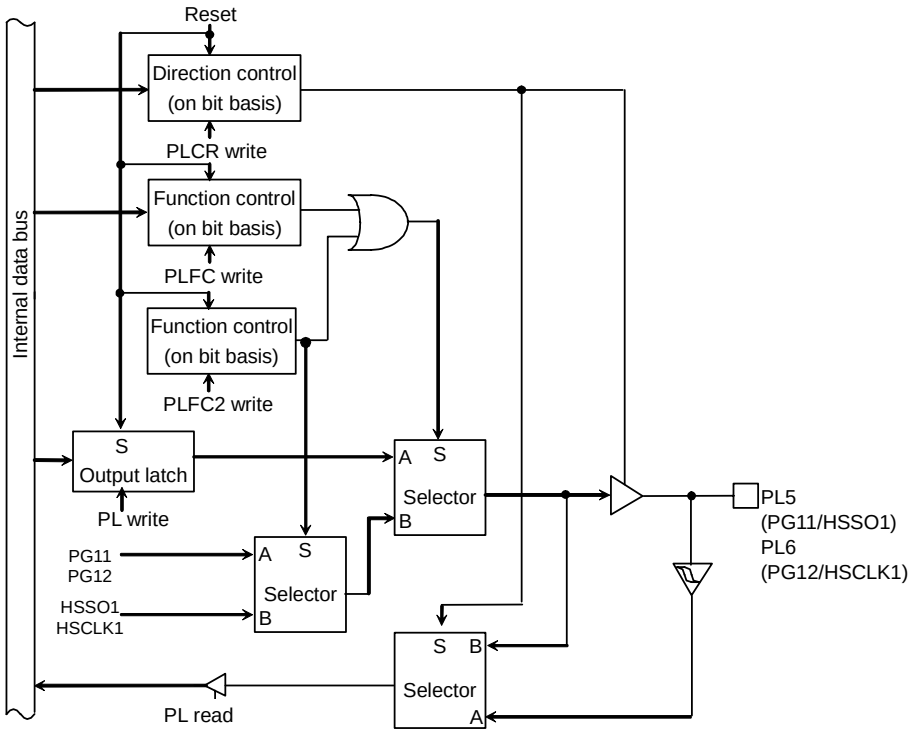


Figure 3.5.44 Port L(PL5,PL6)

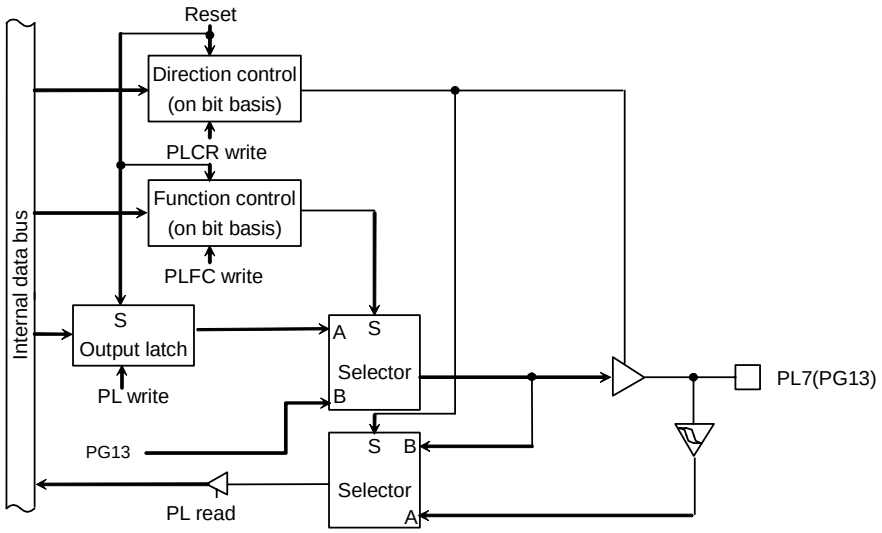


Figure 3.5.45 Port L(PL7)

Port L register

PL (0054H)

	7	6	5	4	3	2	1	0
bit Symbol	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
Read/Write	R/W							
After reset	Data from external port(Output latch register is set to “1”)							

Port L Control register

PLCR (0056H)

	7	6	5	4	3	2	1	0
bit Symbol	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
	Refer to following table							

Port L Function register

PLFC (0057H)

	7	6	5	4	3	2	1	0
bit Symbol	PL7F	PL6F	PL5F	PL4F	PL3F	PL2F	PL1F	PL0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to following table							

Port L Function register 2

PLFC2 (0055H)

	7	6	5	4	3	2	1	0
bit Symbol		PL6F2	PL5F2	PL4F2	PL3F2	PL2F2	PL1F2	PL0F2
Read/Write		W						
After reset		0	0	0	0	0	0	0
Function		Refer to following table						

Port L function setting

<PLxF2>	<PLxF>	<PLxC>	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
0	0	0	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
0	0	1	Output port	Output port	Output port	Output port	Output port	Output port	Output port	Output port
0	1	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	1	1	PG13	PG12	PG11	PG10	PG03	PG02	PG01	PG00
1	0	0		Reserved	Reserved	HSSI1	Reserved	SCLK3/ $\overline{\text{CTS3}}$	Reserved	RXD3
1	0	1		HSCLK1	HSSO1	Reserved	Reserved	SCLK3	TXD3 (O.D Dis)	Reserved
1	1	0		Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1	1	1		Reserved	Reserved	Reserved	TA7OUT	Reserved	TXD3 (O.D Ena)	Reserved

Note 1) Read-modify-write is prohibited for PLCR, PLFC and PLFC2.

Note 2) RXD3, SCLK3 and $\overline{\text{CTS3}}$ input are inputted into the serial bus interface 3 irrespective of a functional setup of a port.

Note 3) HSSI1 input are inputted into the high speed serial channel 0 irrespective of a functional setup of a port.

Note 4) PL1 does not have a register for 3-state/open drain setup.
Moreover, there is no open drain function at the time of an output port.

Figure 3.5.46 Port L register

3.5.13 Port M (PM0 to PM7)

Port M are a port only for inputs.

There are the following functions in addition to an input port.

- The input function of A/D converter(AN0 to AN7)
- The input function of Key input(KI0 to KI7)

These functions operate by setting the bit concerned of PMFC and KIEN register. PMFC is set in "1", is reset KIEN in "0" by the reset operation, and all bits become analog inputs.

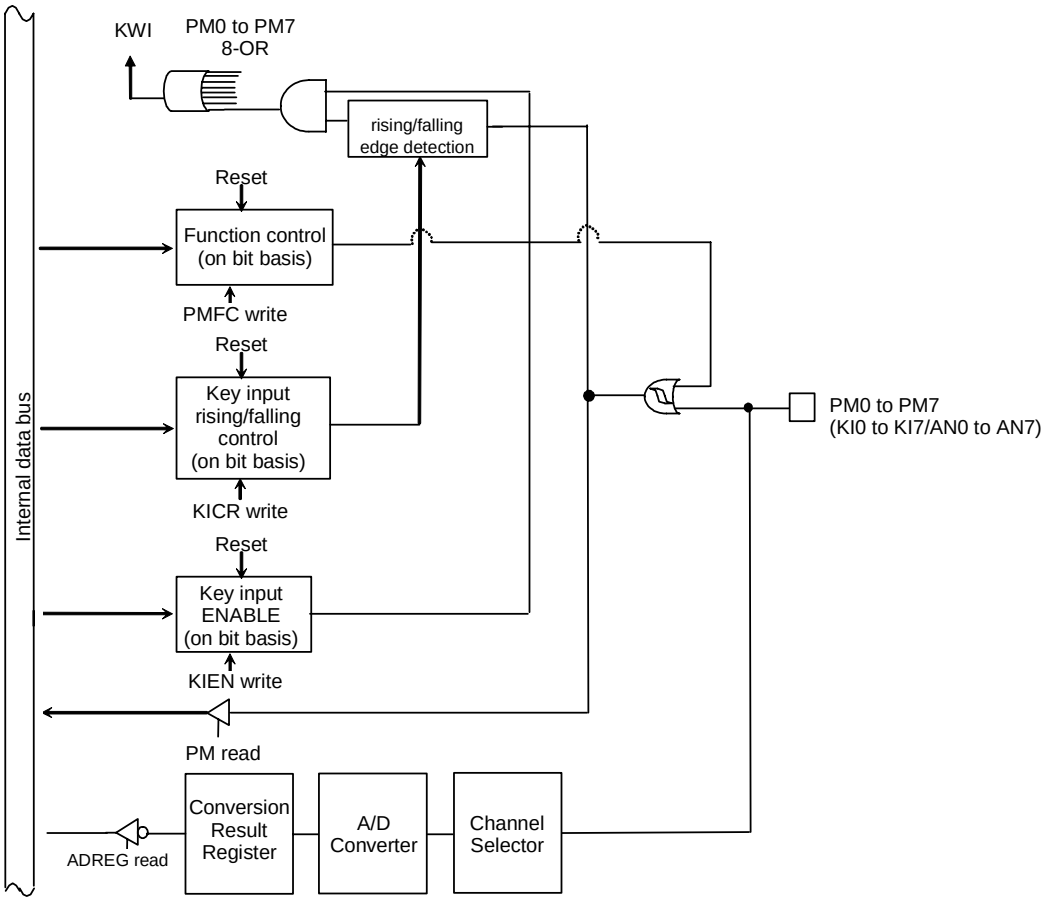


Figure 3.5.47 Port M(PM0 to PM7)

Port M register

PM (0058H)

	7	6	5	4	3	2	1	0
bit Symbol	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
Read/Write	R							
After reset	Data from external port							

Note) The input channel selection of the A/D converter is set by A/D converter mode register ADMOD1.

Port M Function register

PMFC (005BH)

	7	6	5	4	3	2	1	0
bit Symbol	PM7F	PM6F	PM5F	PM4F	PM3F	PM2F	PM1F	PM0F
Read/Write	W							
After reset	1	1	1	1	1	1	1	1
	0: Input port/Key input 1: Analog input							

Key input Enable register

KIEN (009EH)

	7	6	5	4	3	2	1	0
bit Symbol	KI7EN	KI6EN	KI5EN	KI4EN	KI3EN	KI2EN	KI1EN	KI0EN
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
	KI7 input 0: Disable 1: Enable	KI6 input 0: Disable 1: Enable	KI5 input 0: Disable 1: Enable	KI4 input 0: Disable 1: Enable	KI3 input 0: Disable 1: Enable	KI2 input 0: Disable 1: Enable	KI1 input 0: Disable 1: Enable	KI0 input 0: Disable 1: Enable

Key input Control register

KICR (009FH)

	7	6	5	4	3	2	1	0
bit Symbol	KI7EDGE	KI6EDGE	KI5EDGE	KI4EDGE	KI3EDGE	KI2EDGE	KI1EDGE	KI0EDGE
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
	KI7 edge 0: Rising 1: Falling	KI6 edge 0: Rising 1: Falling	KI5 edge 0: Rising 1: Falling	KI4 edge 0: Rising 1: Falling	KI3 edge 0: Rising 1: Falling	KI2 edge 0: Rising 1: Falling	KI1 edge 0: Rising 1: Falling	KI0 edge 0: Rising 1: Falling

Note) Read-modify-write is prohibited for PMFC, KIEN and KICR.

Figure 3.5.48 Port M register

3.5.14 Port N(PN0 to PN3)

Port N are a port only for inputs.

There are the following functions in addition to an input port.

- The input function of A/D converter(AN8 to AN10, AN11/ $\overline{\text{ADTRG}}$)
- The input function of Key input(KI0 to KI7)

These functions operate by setting the bit concerned of PNFC and KIEN register. PNFC is set in "1" by the reset operation, and all bits become analog inputs.

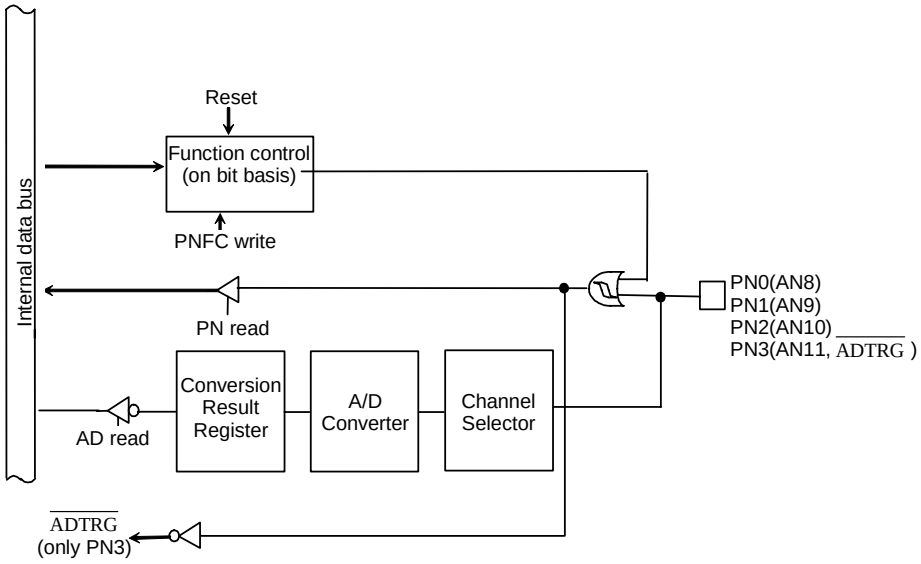


Figure 3.5.49 Port N(PN0 to PN3)

Port N register

PN (005CH)

	7	6	5	4	3	2	1	0
bit Symbol					PN3	PN2	PN1	PN0
Read/Write					R			
After reset					Data from external port			

Note) The input channel selection of the A/D converter is set by A/D converter mode register ADMOD1.
Moreover, the setting of AD trigger ($\overline{\text{ADTRG}}$) input permission is set by ADMOD2<ADTRGE >.

Port N Function register

PNFC (005FH)

	7	6	5	4	3	2	1	0
bit Symbol					PN3F	PN2F	PN1F	PN0F
Read/Write					W			
After reset					1	1	1	1
					0: Input port 1: Analog input			

Note) Read-modify-write is prohibited for PNFC.

Figure 3.5.50 Port N register

3.6 Memory Controller

3.6.1 Functions

TMP92CM27 has a memory controller with a variable 6-block address area that controls as follows.

(1) 6-block address area support

Specifies a start address and a block size for 6-block address area (block 0 to 5).

- SRAM or ROM : All CS blocks (CS0 to CS5) are supported.
- SDRAM : Only CS3 blocks are supported.
- Page ROM : Only CS2 blocks are supported.

(2) Connecting memory specifications

Specifies SRAM, ROM and SDRAM as memories that connect with the selected address areas.

(3) Data bus width selection

Whether 8 bits, 16 bits is selected as the data bus width of the respective block address areas.

(4) Wait control

Wait specification bit in the control register and $\overline{\text{WAIT}}$ input pin control the number of waits in the external bus cycle. Read cycle and write cycle can specify the number of waits individually.

The number of waits is controlled in 6 mode mentioned below.

0 waits, 1 wait,

2 waits, 3 waits, 4 waits

N waits (controls 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 after reset release state and necessary settings.

(1) Control register

The control registers of the memory controller are follows and Table 3.6.1 and Table 3.6.2.

- Control register: BnCSH/BnCSL (n = 0 to 5, EX)
  Sets the basic functions of the memory controller; the memory type that is connected, the number of waits whish is read and written.
- Memory start address register: MSARn (n = 0 to 5)
  Sets a start address in the selected address areas.
- Memory address mask register: MAMR (n = 0 to 5)
  Sets a block size in the selected address areas.
- Page ROM control register: PMEMCR
  Sets method of accessing page ROM.

Table 3.6.1 Control Register

		7	6	5	4	3	2	1	0
B0CSL (0140H)	Bit symbol		B0WW2	B0WW1	B0WW0		B0WR2	B0WR1	B0WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B0CSH (0141H)	Bit symbol	B0E	–	–	B0REC	B0OM1	B0OM0	B0BUS1	B0BUS0
	Read/Write	W							
	After reset	0	0 (Note)	0 (Note)	0	0	0	0	0
MAMR0 (0142H)	Bit symbol	M0V20	M0V19	M0V18	M0V17	M0V16	M0V15	M0V14 to M0V9	M0V8
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR0 (0143H)	Bit symbol	M0S23	M0S22	M0S21	M0S20	M0S19	M0S18	M0S17	M0S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
B1CSL (0144H)	Bit symbol		B1WW2	B1WW1	B1WW0		B1WR2	B1WR1	B1WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B1CSH (0145H)	Bit symbol	B1E	–	–	B1REC	B1OM1	B1OM0	B1BUS1	B1BUS0
	Read/Write	W							
	After reset	0	0 (Note)	0 (Note)	0	0	0	0	0
MAMR1 (0146H)	Bit symbol	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	M1V15 to M1V9	M1V8
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR1 (0147H)	Bit symbol	M1S23	M1S22	M1S21	M1S20	M1S19	M1S18	M1S17	M1S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
B2CSL (0148H)	Bit symbol		B2WW2	B2WW1	B2WW0		B2WR2	B2WR1	B2WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B2CSH (0149H)	Bit symbol	B2E	B2M	–	B2REC	B2OM1	B2OM0	B2BUS1	B2BUS0
	Read/Write	W							
	After reset	1	0	0 (Note)	0	0	0	0	0
MAMR2 (014AH)	Bit symbol	M2V22	M2V21	M2V20	M2V19	M2V18	M2V17	M2V16	M2V15
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR2 (014BH)	Bit symbol	M2S23	M2S22	M2S21	M2S20	M2S19	M2S18	M2S17	M2S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
B3CSL (014CH)	Bit symbol		B3WW2	B3WW1	B3WW0		B3WR2	B3WR1	B3WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B3CSH (014DH)	Bit symbol	B3E	–	–	B3REC	B3OM1	B3OM0	B3BUS1	B3BUS0
	Read/Write	W							
	After reset	0	0 (Note)	0 (Note)	0	0	0	0	0
MAMR3 (014EH)	Bit symbol	M3V22	M3V21	M3V20	M3V19	M3V18	M3V17	M3V16	M3V15
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR3 (014FH)	Bit symbol	M3S23	M3S22	M3S21	M3S20	M3S19	M3S18	M3S17	M3S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Note 1: Always write “0”.

Note 2: Read modify write is prohibited for BnCSL and BnCSH (n = 0 to 3) registers.

Table 3.6.2 Control Register

		7	6	5	4	3	2	1	0
B4CSL (0150H)	Bit symbol		B4WW2	B4WW1	B4WW0		B4WR2	B4WR1	B4WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B4CSH (0151H)	Bit symbol	B4E	B4M	–	B4REC	B4OM1	B4OM0	B4BUS1	B4BUS0
	Read/Write	W							
	After reset	0	0 (Note)	0 (Note)	0	0	0	0	0
MAMR4 (0152H)	Bit symbol	M4V22	M4V21	M4V20	M4V19	M4V18	M4V17	M4V16	M4V15
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR4 (0153H)	Bit symbol	M4S23	M4S22	M4S21	M4S20	M4S19	M4S18	M4S17	M4S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
B5CSL (0154H)	Bit symbol		B5WW2	B5WW1	B5WW0		B5WR2	B5WR1	B5WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B5CSH (0155H)	Bit symbol	B5E	–	–	B5REC	B5OM1	B5OM0	B5BUS1	B5BUS0
	Read/Write	W							
	After reset	0	0 (Note)	0 (Note)	0	0	0	0	0
MAMR5 (0156H)	Bit symbol	M5V22	M5V21	M5V20	M5V19	M5V18	M5V17	M5V16	M5V15
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR5 (0157H)	Bit symbol	M5S23	M5S22	M5S21	M5S20	M5S19	M5S18	M5S17	M5S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
BEXCSH (0159H)	Bit symbol					BEXOM1	BEXOM0	BEXBUS1	BEXBUS0
	Read/Write					W			
	After reset					0	0	0	0
BEXCSL (0158H)	Bit symbol		BEXWW2	BEXWW1	BEXWW0		BEXWR2	BEXWR1	BEXWR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
PMEMCR (0166H)	Bit symbol				OPGE	OPWR1	OPWR0	PR1	PR0
	Read/Write				R/W				
	After reset				0	0	0	1	0

Note 1: Always write “0”.

Note 2: Read modify write is prohibited for BnCSL, BnCSH(n = 4 to 5), BEXCSH and BEXCSL registers.

(2) Operation after reset release

The start data bus width is determined depending on state of AM1/AM0 pins just after reset release. Then, the external memory is accessed as follows

AM1	AM0	Start Mode
0	0	Don’t use this setting
0	1	Start with 16-bit data bus (Note)
1	0	Start with 8-bit data bus (Note)
1	1	Don’t use this setting

Note: A memory to be used as starting after reset is either NOR flash, masked ROM.
SDRAM can’t be used.

AM1/AM0 pins are valid only just after release reset. In the other cases, the data bus width is the value which is set to the control register <BnBUS1:0> .

By reset, only control register (B2CSH/B2CSL) of the block address area 2 becomes effective automatically (B2CSH<B2E> is set to “1” by reset).

The data bus width which is specified by AM1/AM0 pins are loaded to the bit for specification the bus width of the control register in the block address area 2.

The block address area 2 is set to 000000H to FFFFFFH address by reset (B2CSH<B2M> is reset to "0").

After release reset, the block address areas are specified by the memory start address register (MSARn) and the memory address mask register (MAMRn). Then the control register (BnCSH/L) is set.

Set the enable bit (BnCSH<BnE>) of the control register to "1" for enable the setting.

3.6.3 Basic Functions and Register Setting

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

(1) Block address area specification

The block address area is specified by two registers.

The memory start address register (MSARn) sets the start address of the block address areas. The memory controller compares between the register value and the address every bus cycles. The address bit which is masked by the memory address mask register (MAMRn) is not compared by the memory controller. The block address area size is determined by setting the memory address mask register. The value that is set to the register is compared with the block address area on the bus. If the compared result is a match, the memory controller sets the chip select signal (CSn) to "low".

(i) Memory start address register setting

The MS23 to 16 bits of the memory start address register correspond with addresses A23 to A16 respectively. The lower start addresses A15 to A0 are always set to address 0000H.

Therefore the start addresses of the block address area are set to addresses 000000H to FF0000H every 64 Kbytes.

(ii) Memory address mask registers setting

The memory address mask register sets whether an address bit is compared or not. In register setting, "0" is "compare", or "1" is "not compare".

The address bits that can set depend on the block address area.

Block address area 0: A20 to A8

Block address area 1: A21 to A8

Block address area 2 to 5: A22 to A15

The upper bits are always compared. The block address area size is determined by the compared result.

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

Size (bytes) / CS area	256	512	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M
CS0	○	○	○	○	○	○	○	○	○		
CS1	○	○		○	○	○	○	○	○	○	
CS2 to CS5			○	○	○	○	○	○	○	○	○

Note: After release reset, only the control register of the block address area 2 is valid. The control register of the block address area 2 has <B2M> bit. If <B2M> bit set to "0", the block address area 2 is set to addresses 000000H to FFFFFFH. (After release reset state is this state). If <B2M> bit set to "1", the start address and the address area size is set, as in the other block address area.

(iii) Example of register setting

To set the block address area 512bytes from address 110000H, set the register as follows.

MSAR1 Register

Bit	7	6	5	4	3	2	1	0
Bit symbol	M1S23	M1S22	M1S21	M1S20	M1S19	M1S18	M1S17	M1S16
Specified value	0	0	0	1	0	0	0	1

M1S23 to M1S16 bits of the memory start address register MSAR1 correspond with address A23 to A16.

A15 to A0 are set to "0". Therefore if MSAR1 is set to above values, the start address of the block address area is set to address 110000H.

MAMR1 Register

Bit	7	6	5	4	3	2	1	0
Bit symbol	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	M1V15 to M1V9	M1V8
Specified value	0	0	0	0	0	0	0	1

M1V21 to M1V16 and M1V8 bits of the memory address mask register MAMR1 set whether address A21 to A16 and A8 are compared or not. In register setting, "0" is "compare", or "1" is "not compare". M1V15 to M1V9 bits set whether address A15 to A9 are compared or not with 1 bit. A23 and A22 are always compared.

If it set to like an above setting, A23 to A9 is compared with the value that is set as the start addresses. Therefore 512 bytes (addresses 110000H to 1101FFH) are set as the block address area 1, and if it is compared with the addresses on the bus, the chip select signal CS1 is set to "low".

The other block address area sizes are specified like this.

A23 and A22 are always compared in the block address area 0. Whether A20 to A8 are compared or not is set to register.

Similarly, A23 is always compared in block address areas 2 to 5. Whether A22 to A15 are compared or not is set to register.

Note 1: When the set block address area overlaps with the built-in memory area, or both two address areas overlap, the block address area is processed according to priority as follows.

Built-in I/O > Built-in memory > Block address area 0 > 1 > 2 > 3 > 4 > 5

Note 2: If address area that is set in $\overline{CS0}$ to $\overline{CS5}$ was accessed, area is regarded as $\overline{CSEX}$ area. Therfore, wait number and data bus width controls becomes setting of $\overline{CSEX}$ (BEXCSH, BEXCSL register).

(2) Connection memory specification

Setting the <BnOM1:0> bit of the control register (BnCSH) specifies the memory type that is connected with the block address areas. The interface signal is outputted according to the set memory as follows.

<BnOM1:0> Bit (BnCSH Register)

<BnOM1>	<BnOM0>	Function
0	0	SRAM/ROM (Default)
0	1	(Reserved)
1	0	(Reserved)
1	1	SDRAM

Note 1: SDRAM should be set to block either 3.

(3) Data bus width specification

The data bus width is set for every block address area. The bus size is set by setting the control register (BnCSH)<BnBUS1:0> as follows.

<BnBUS1:0> bit (BnCSH Register)

BnBUS 1	BnBUS 0	Function
0	0	8-bit bus mode (Default)
0	1	16-bit bus mode
1	0	Don't use this setting
1	1	Don't use this setting

Note: SDRAM should be set to either "01" (16-bit bus).

This method of changing the data bus width depending on the accessing address is called "dynamic bus sizing". Part which data is outputted is changed by changing data size, bus width and start address.

Note: Since there is a possibility of abnormal writing/reading of the data if two memories with different bus width are put in consecutive address, do not execute a access to both memories with one command.

Operand Data Size (bit)	Operand Start Address	Memory Data Size (bit)	CPU Address	CPU Data	
				D15 to D8	D7 to D0
8	4n + 0	8/16	4n + 0	xxxxx	b7 to b0
	4n + 1	8	4n + 1	xxxxx	b7 to b0
	4n + 2	8/16	4n + 2	xxxxx	b7 to b0
	4n + 3	8	4n + 3	xxxxx	b7 to b0
		16	4n + 3	b7 to b0	xxxxx
16	4n + 0	8	(1) 4n + 0	xxxxx	b7 to b0
			(2) 4n + 1	xxxxx	b15 to b8
		16	4n + 0	b15 to b8	b7 to b0
	4n + 1	8	(1) 4n + 1	xxxxx	b7 to b0
			(2) 4n + 2	xxxxx	b15 to b8
		16	(1) 4n + 1	b7 to b0	xxxxx
			(2) 4n + 2	xxxxx	b15 to b8
	4n + 2	8	(1) 4n + 2	xxxxx	b7 to b0
			(2) 4n + 1	xxxxx	b15 to b8
		16	4n + 2	b15 to b8	b7 to b0
	4n + 3	8	(1) 4n + 3	xxxxx	b7 to b0
			(2) 4n + 4	xxxxx	b15 to b8
		16	(1) 4n + 3	b7 to b0	xxxxx
			(2) 4n + 4	xxxxx	b15 to b8
32	4n + 0	8	(1) 4n + 0	xxxxx	b7 to b0
			(2) 4n + 1	xxxxx	b15 to b8
			(3) 4n + 2	xxxxx	b23 to b16
			(4) 4n + 3	xxxxx	b31 to b24
		16	(1) 4n + 0	b15 to b8	b7 to b0
			(2) 4n + 2	b31 to b24	b23 to b16
	4n + 1	8	(1) 4n + 0	xxxxx	b7 to b0
			(2) 4n + 1	xxxxx	b15 to b8
			(3) 4n + 2	xxxxx	b23 to b16
			(4) 4n + 3	xxxxx	b31 to b24
		16	(1) 4n + 1	b7 to b0	xxxxx
			(2) 4n + 2	b23 to b16	b15 to b8
			(3) 4n + 4	xxxxx	b31 to b24
	4n + 2	8	(1) 4n + 2	xxxxx	b7 to b0
			(2) 4n + 3	xxxxx	b15 to b8
			(3) 4n + 4	xxxxx	b23 to b16
			(4) 4n + 5	xxxxx	b31 to b24
		16	(1) 4n + 2	b15 to b8	b7 to b0
			(2) 4n + 4	b31 to b24	b23 to b16
	4n + 3	8	(1) 4n + 3	xxxxx	b7 to b0
			(2) 4n + 4	xxxxx	b15 to b8
			(3) 4n + 5	xxxxx	b23 to b16
			(4) 4n + 6	xxxxx	b31 to b24
		16	(1) 4n + 3	b7 to b0	xxxxx
			(2) 4n + 4	b23 to b16	b15 to b8
			(3) 4n + 6	xxxxx	b31 to b24

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

(4) Wait control

The external bus cycle completes a wait of two states at least (100 ns at f_{SYS} = 20 MHz).

Setting the <BnWW2:0> and <BnWR2:0> of BnCSL specifies the number of waits in the read cycle and the write cycle. <BnWW2:0> is set with the same method as <BnWR2:0>.

<BnWW>/<BnWR> (BnCSL Register)

<BnWW2> <BnWR2>	<BnWW1> <BnWR1>	<BnWW0> <BnWR0>	Function
0	0	1	2 states (0 waits) access fixed mode
0	1	0	3 states (1 wait) access fixed mode (Default)
1	0	1	4 states (2 waits) access fixed mode
1	1	0	5 states (3 waits) access fixed mode
1	1	1	6 states (4 waits) access fixed mode
0	1	1	$\overline{\text{WAIT}}$ pin input mode
Others			(Reserved)

Note 1: For SDRAM, above setting is invalid. So, refer 3.13 SDRAM controller.

(i) Waits number fixed mode

The bus cycle is completed with the states which is set. The number of states is selected from 2 states (0 waits) to 6 states (4 waits).

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

This mode samples the $\overline{\text{WAIT}}$ input pins. And this mode inserts wait continuously in during signal is actived. The bus cycle is minimum 2 states. The bus cycle is completed if the wait signal is non active ("High" level) at 2 states. The bus cycle continue with that is extended if the wait signal is active at 2 states and more.

(5) Recovery (Data hold) cycle control

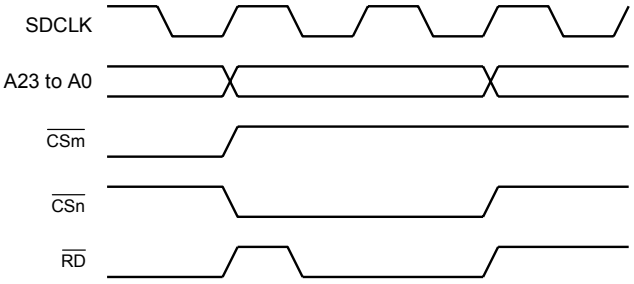
Some memory is defined an AC specification about data hold time by $\overline{CE}$ or $\overline{OE}$ for read cycle. Therefore, a data confliction problem may occur. To avoid this problem, 1-dummy cycle can be inserted after CSm-block access cycle by setting "1" to BmCSH<BmREC>.

This 1-dummy cycle is inserted when the next cycle is for another CS-block.

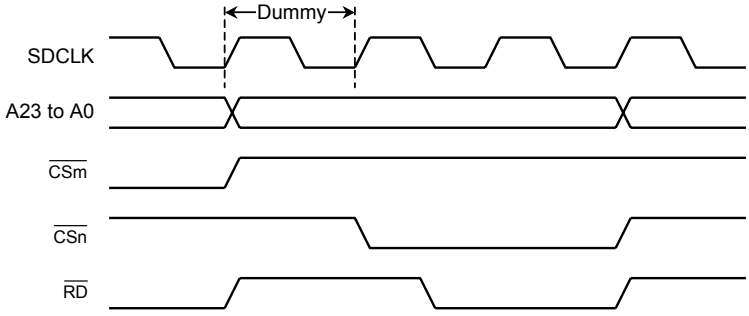
<BnREC> (BnCSH register)

0	No dummy cycle is inserted (Default).
1	Dummy cycle is inserted.

- When no inserting a dummy cycle (0 waits)

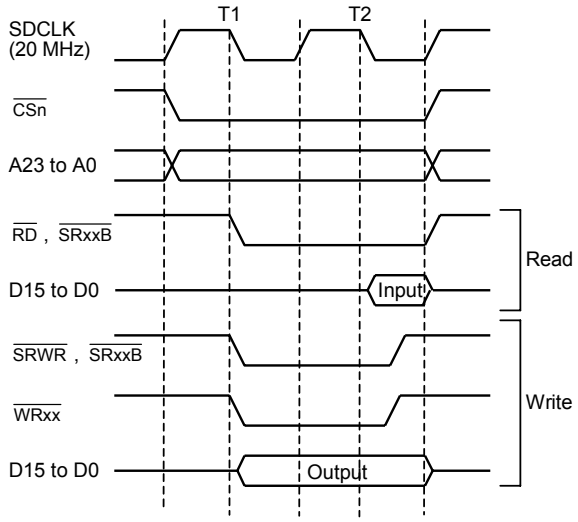


- When inserting a dummy cycle (0 waits)

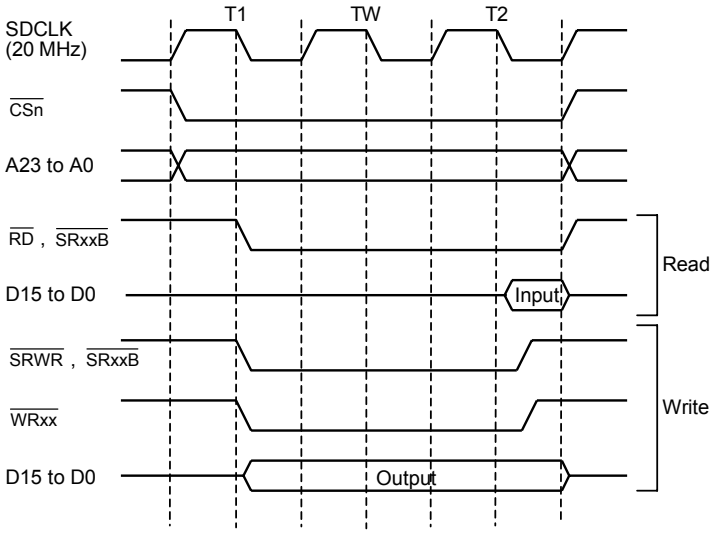


(6) Basic bus timing

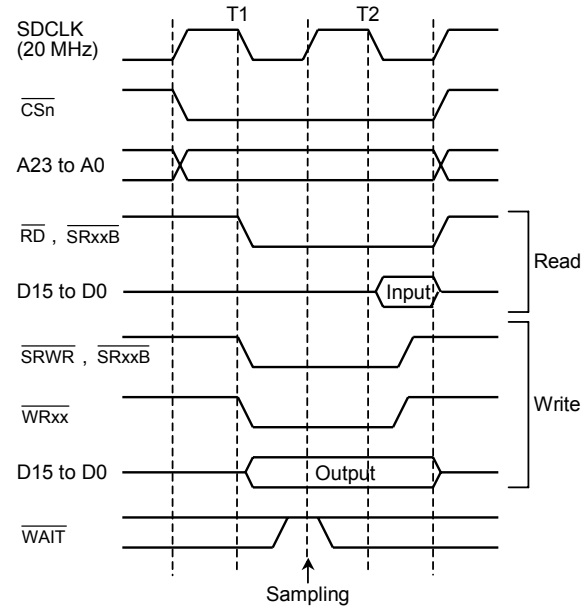
(a) External read/write cycle (0 waits)



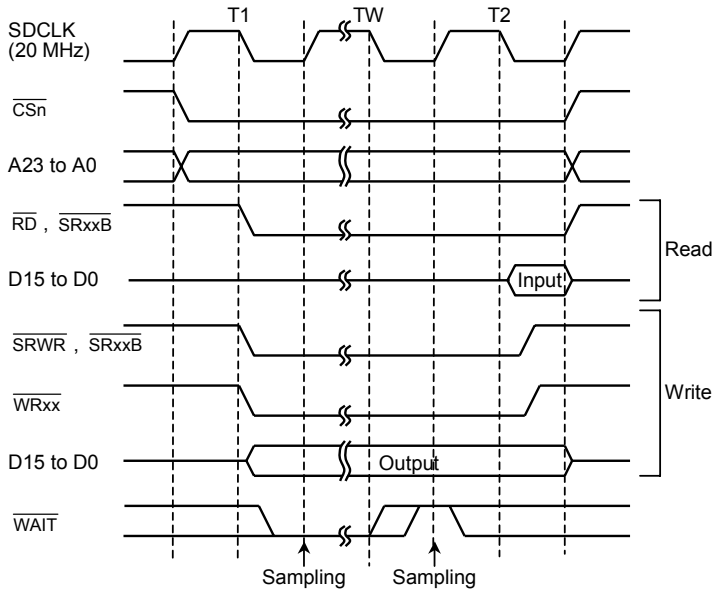
(b) External read/write cycle (1 wait)



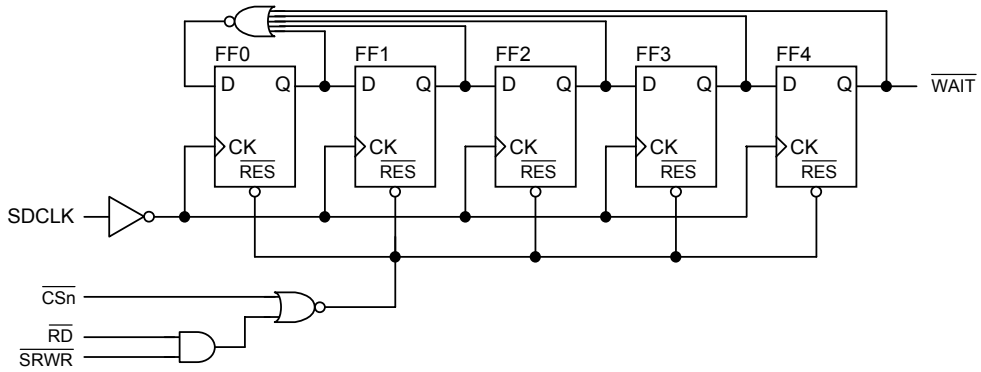
(c) External read/write cycle (0 waits at $\overline{\text{WAIT}}$ pin input mode)

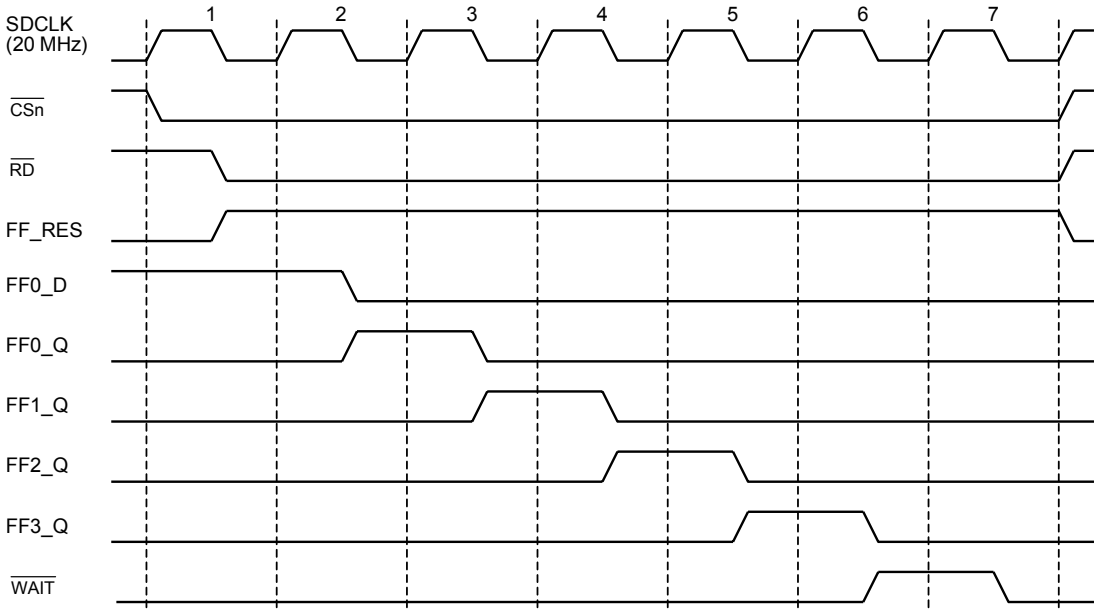


(d) External read/write cycle (n waits at $\overline{\text{WAIT}}$ pin input mode)



Example of wait input cycle (5 waits)





(7) Connecting external memory

Figure 3.6.1 shows an example of method of connecting external 16-bit SRAM and 16-bit NOR flash to the TMP92CM27.

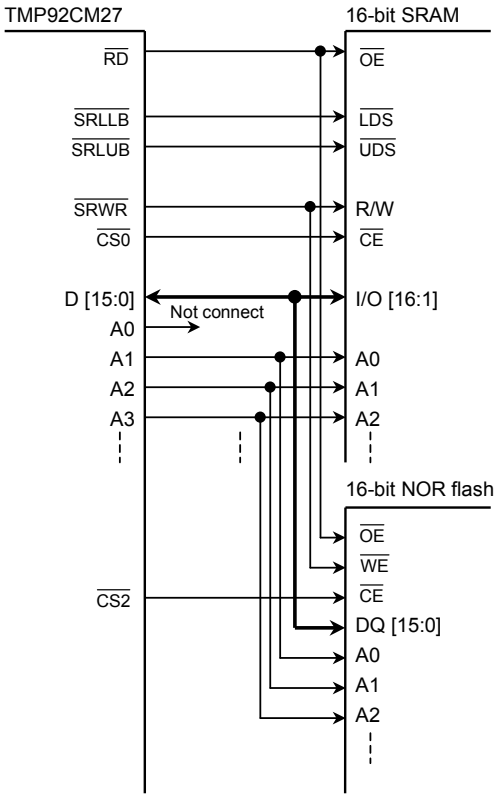


Figure 3.6.1 Example of External 16-Bit SRAM and NOR Flash Connection

3.6.4 ROM Control (Page mode)

This section describes ROM page mode accessing and how to set registers. ROM page mode is set by the page ROM control register.

(1) Operation and how to set the registers

TMP92CM27 supports ROM access of the page mode. The ROM access of the page mode is specified only in the block address area 2.

ROM page mode is set by the page ROM control register (PMEMCR). Setting <OPGE> of the PMEMCR register to "1" sets the memory access of the block address area to ROM page mode access.

The number of read cycles is set by the <OPWR1:0> of the PMEMCR register.

<OPWR1:0> (PMEMCR register)

<OPWR1>	<OPWR0>	Number of Cycle in a Page
0	0	1 state (n-1-1-1 mode) (n ≥ 2)
0	1	2 state (n-2-2-2 mode) (n ≥ 3)
1	0	3 state (n-3-3-3 mode) (n ≥ 4)
1	1	(Reserved)

Note: Set the number of waits "n" to the control register (BnCSL) in each block address area.

The page size (the number of bytes) of ROM in the CPU size is set to the <PR1:0> of the PMEMCR register. When data is read out until a border of the set page, the controller completes the page reading operation. The start data of the next page is read in the normal cycle. The following data is set to page read again.

<PR1:0> Bit (PMEMCR register)

<PR1>	<PR0>	ROM Page Size
0	0	64 bytes
0	1	32 bytes
1	0	16 bytes (Default)
1	1	8 bytes

For the signal timing pulse, see ROM read cycle in section 4.3.2.

3.6.5 Cautions

(1) Note the timing between $\overline{CS}$ and $\overline{RD}$

If the parasitic capacitance of the $\overline{RD}$ (Read signal) is greater than that of the $\overline{CS}$ (Chip select signal), it is possible that an unintended read cycle occurs due to a delay in the read signal. Such an unintended read cycle may cause a trouble as in the case of (a) in Figure 3.6.2.

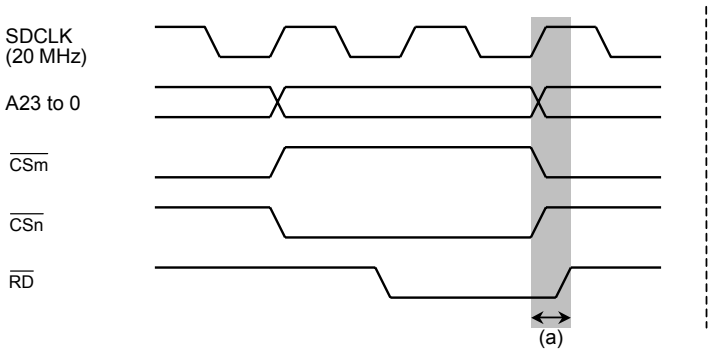


Figure 3.6.2 Read Signal Delay Read Cycle

Example: When using an externally connected NOR flash which users JEDEC standard commands, note that the toggle bit may not be read out correctly. If the read signal in the cycle immediately preceding the access to the NOR flash does not go high in time, as shown in Figure 3.6.3, an unintended read cycle like the one shown in (b) may occur.

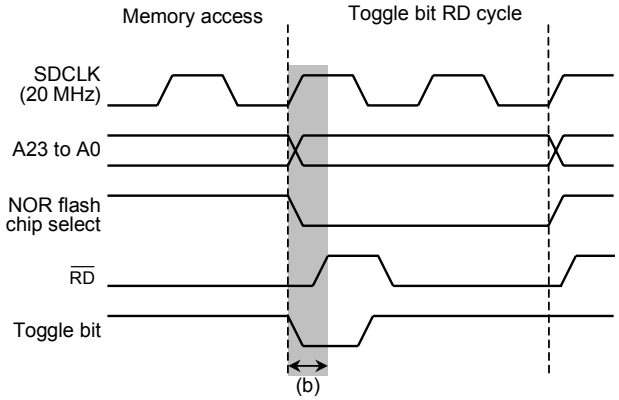


Figure 3.6.3 NOR Flash Toggle Bit Read Cycle

When the toggle bit reverse with this unexpected read cycle, CPU always reads same value of the toggle bit, and cannot read the toggle bit correctly. To avoid this phenomenon, the data polling function control is recommended.

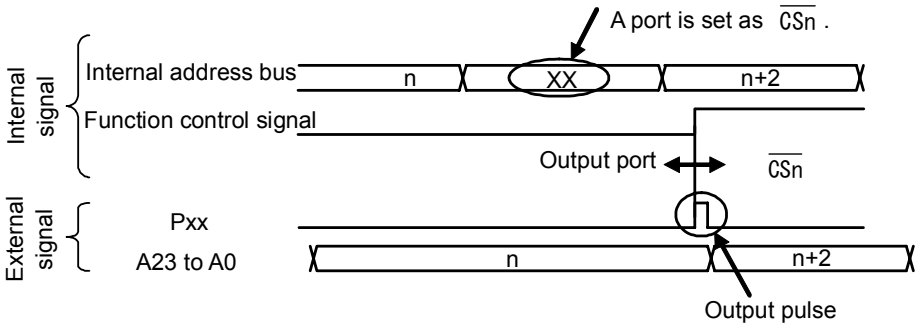
(2) The cautions at the time of the functional change of a $\overline{CSn}$.

A chip select signal output has the case of a combination terminal with a general-purpose port function. In this case, an output latch register and a function control register are initialized by the reset action, and an object terminal is initialized by the port output ("1" or "0") by it.

Functional change

Although an object terminal is changed from a port to a chip select signal output by setting up a function control register (PnFC register), the short pulse for several ns may be outputted to the changing timing. Although it does not become especially a problem when using the usual memory, it may become a problem when using a special memory.

※ XX is a function register address.(When an output port is initialized by "0")

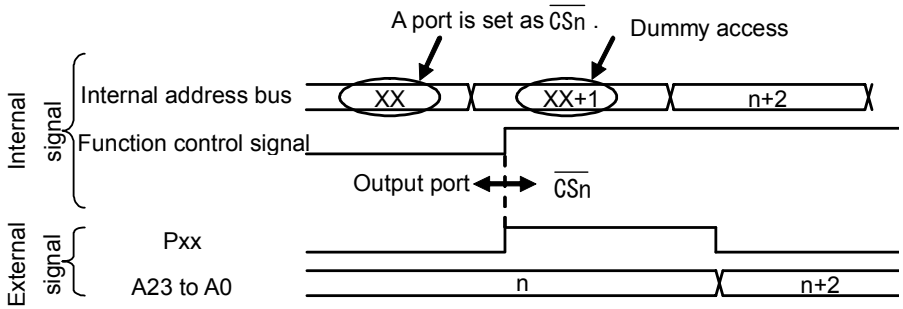


The measure by software

The countermeasures in S/W for avoiding this phenomenon are explained.

Since CS signal decodes the address of the access area and is generated, an unnecessary pulse is outputted by access to the object CS area immediately after setting it as a CSn function. Then, if internal area is accessed also immediately after setting a port as CS function, an unnecessary pulse will not output.

1. Prohibition of use of an NMI function
2. The ban on interruption under functional change (DI command)
3. A dummy command is added in order to carry out continuous internal access.
   (Access to a functional change register is corresponded by 16-bit command.
   (LDW command))



3.7 8-Bit Timers (TMRA)

The TMP92CM27 features 8 built-in 8-bit timers.
These timers are paired into four modules: TMRA01, TMRA23, TMRA45 and TMRA67. 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 cycle with variable period)
- 8-bit pulse width modulation output mode (PWM: Variable duty cycle with constant period)

Figure 3.7.1 to Figure 3.7.4 show block diagrams for TMRA01, TMRA23, TMRA45 and TMRA67.

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-byte controls SFR (Special-function registers).

Each of the four modules (TMRA01, TMRA23, TMRA45 and TMRA67) can be operated independently. All modules operate in the same manner; hence only the operation of TMRA01 is explained here.

The contents of this chapter are as follows.

Table 3.7.1 Registers and Pins for Each Module

Specification \ Module		TMRA01	TMRA23	TMRA45	TMRA67
External pin	Input pin for external clock	TA0IN (Shared with PF0)	TA2IN (Shared with PF2)	TA4IN (Shared with PF4)	TA6IN (Shared with PF6)
	Output pin for timer flip-flop	TA1OUT (Shared with PF1)	TA3OUT (Shared with PF3)	TA5OUT (Shared with PF5)	TA7OUT (Shared with PL3)
SFR (Address)	Timer RUN register	TA01RUN (1100H)	TA23RUN (1108H)	TA45RUN (1110H)	TA67RUN (1118H)
	Timer register	TA0REG (1102H) TA1REG (1103H)	TA2REG (110AH) TA3REG (110BH)	TA4REG (1112H) TA5REG (1113H)	TA6REG (111AH) TA7REG (111BH)
	Timer mode register	TA01MOD(1104H)	TA23MOD(110CH)	TA45MOD(1114H)	TA67MOD(111CH)
	Timer flip-flop control register	TA1FFCR(1105H)	TA3FFCR(110DH)	TA5FFCR(1115H)	TA7FFCR(111DH)

3.7.1 Block Diagrams

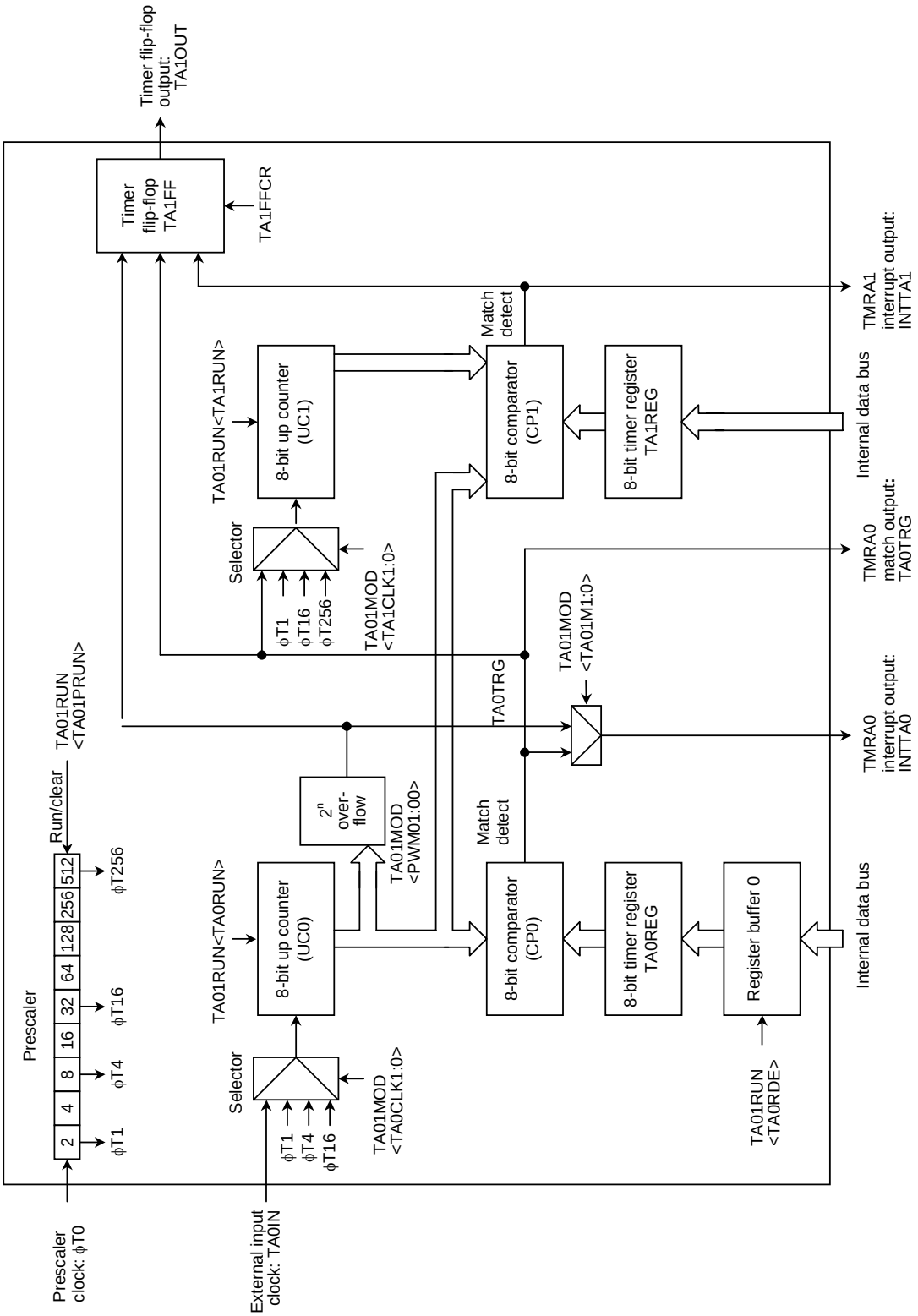


Figure 3.7.1 TMRA01 Block Diagram

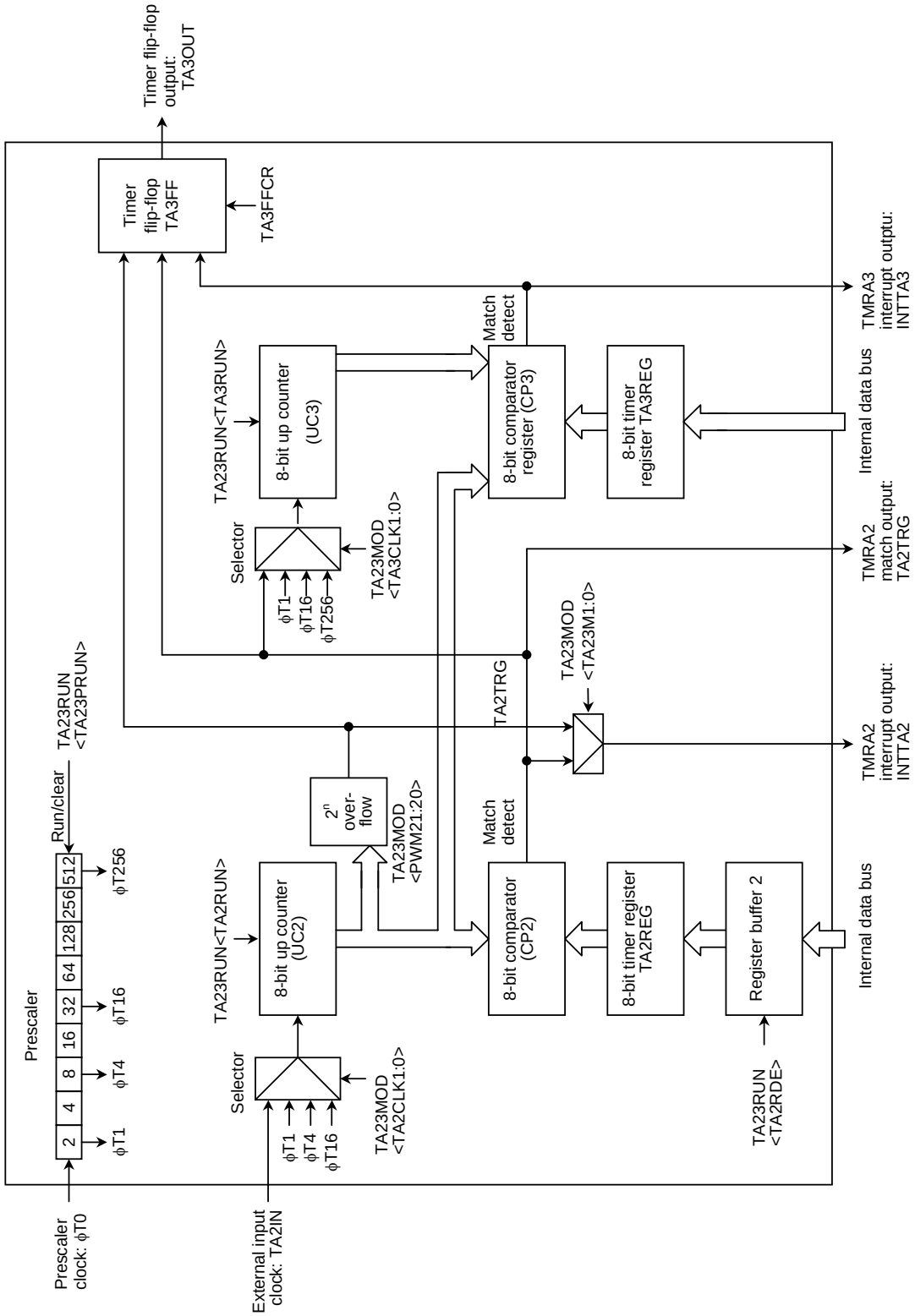


Figure 3.7.2 TMRA23 Block Diagram

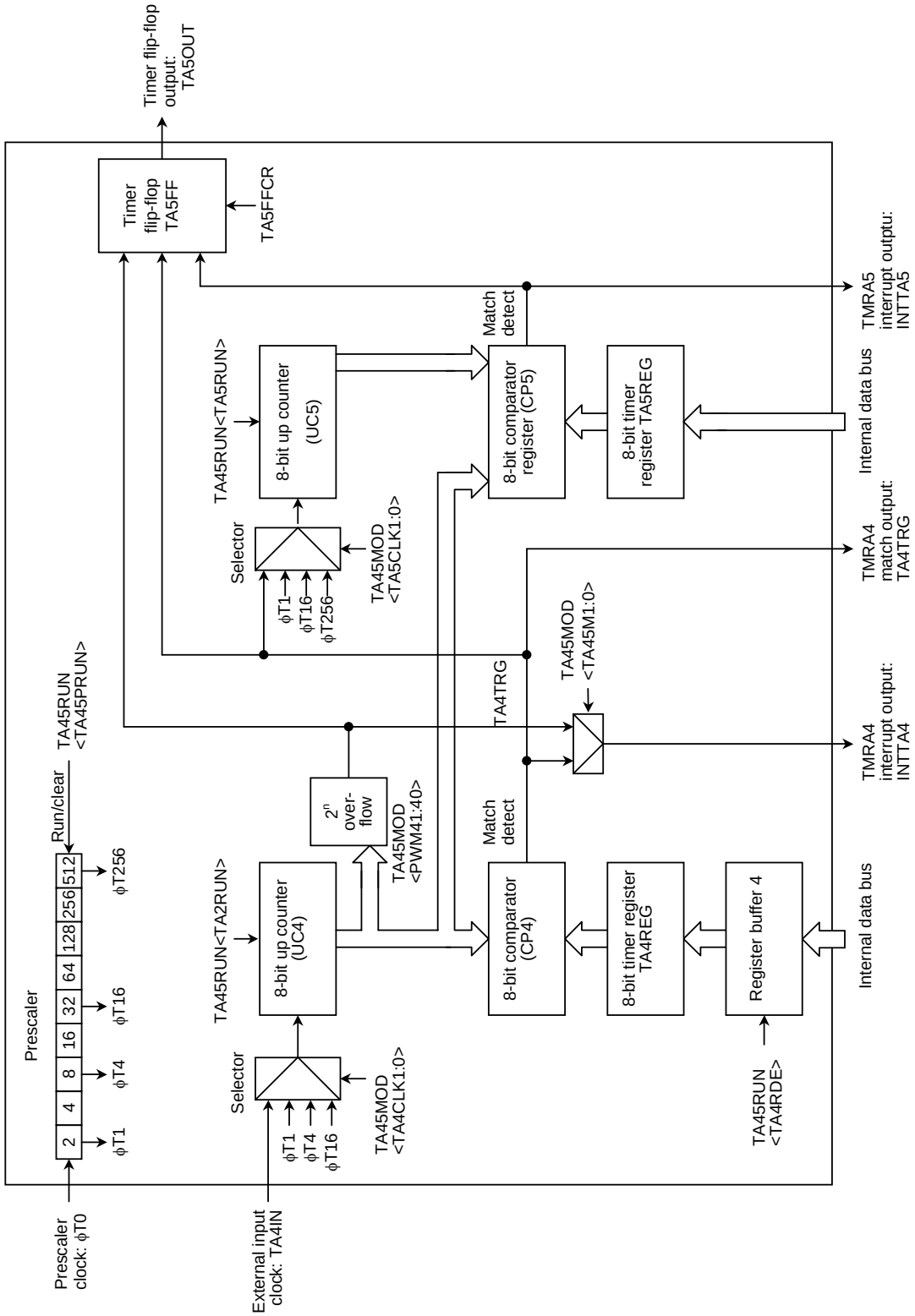


Figure 3.7.3 TMRA45 Block Diagram

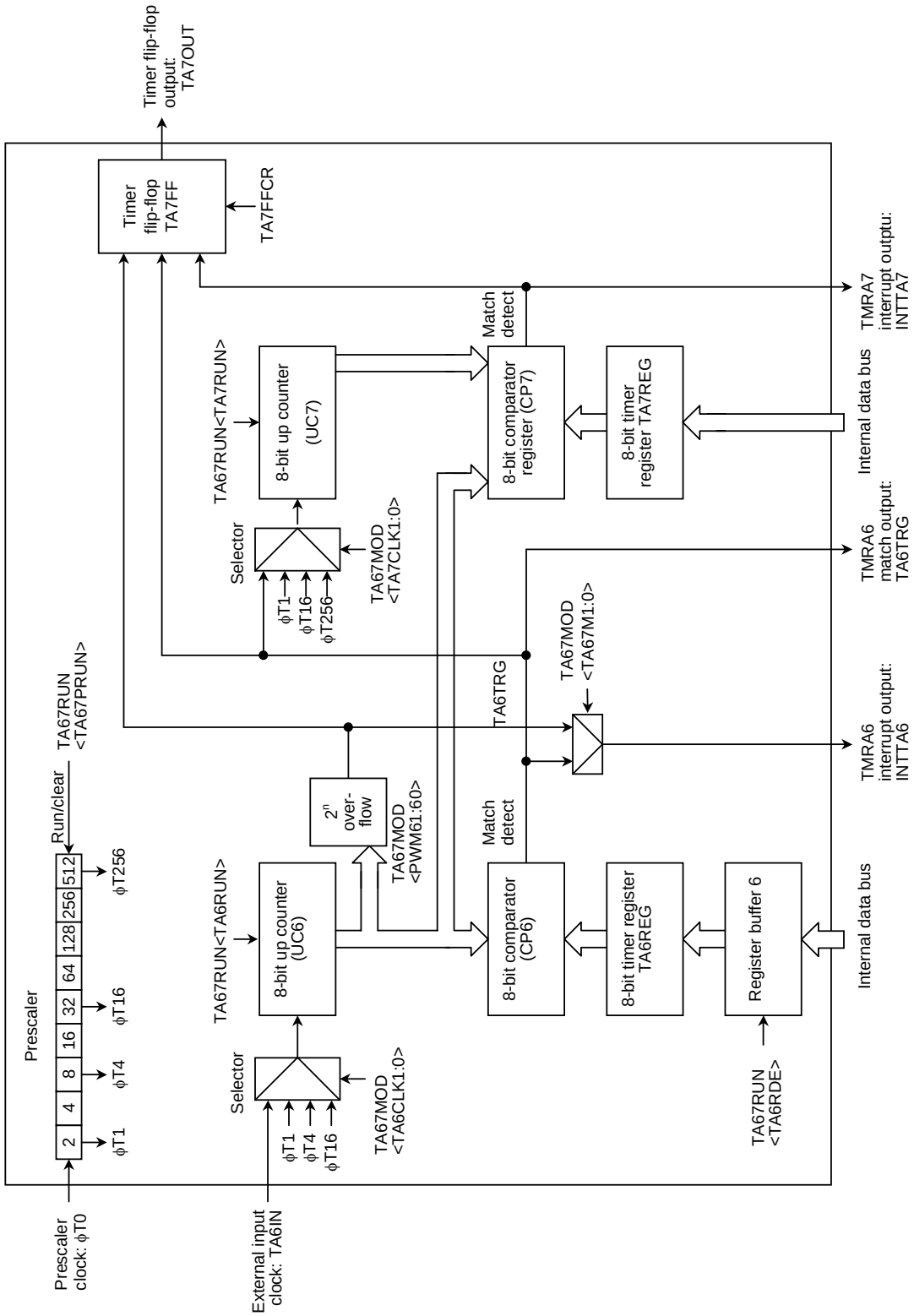


Figure 3.7.4 TMRA67 Block Diagram

3.7.2 Operation of Each Circuit

(1) Prescaler

A 9-bit prescaler generates the input clock to TMRA01.

The prescaler's operation can be controlled using TA01RUN <TA0PRUN> in the timer control register. Setting <TA0PRUN> to "1" starts the count; setting <TA0PRUN> to "0" clears the prescaler to zero and stops operation. Table 3.7.2 shows the various prescaler output clock resolutions.

Table 3.7.2 Prescaler Output Clock Resolution

at fc = 40 MHz

System clock selection <SYSCK>	Gear Value <GEAR2:0>	Cycle			
		φT1	φT4	φT16	φT256
0(fc)	000 (fc)	2^3/fc (0.2 μs)	2^5/fc (0.8 μs)	2^7/fc (3.2 μs)	2^{11}/fc (51.2 μs)
	001 (fc/2)	2^4/fc (0.4 μs)	2^6/fc (1.6 μs)	2^8/fc (6.4 μs)	2^{12}/fc (102.4 μs)
	010 (fc/4)	2^5/fc (0.8 μs)	2^7/fc (3.2 μs)	2^9/fc (12.8 μs)	2^{13}/fc (204.8 μs)
	011 (fc/8)	2^6/fc (1.6 μs)	2^8/fc (6.4 μs)	2^{10}/fc (25.6 μs)	2^{14}/fc (409.6 μs)
	100 (fc/16)	2^7/fc (3.2 μs)	2^9/fc (12.8 μs)	2^{11}/fc (51.2 μs)	2^{15}/fc (819.2 μs)

xxx: Don't care

(2) Up counters (UC0 and UC1)

These are 8-bit binary counters which count up the input clock pulses for the clock specified by TA01MOD.

The input clock for UC0 is selectable and can be either the external clock input via the TA0IN pin or one of the three internal clocks φT1, φT4, or φT16. The clock setting is specified by the value set in TA01MOD <TA0CLK1:0>.

The input clock for UC1 depends on the operation mode. In 16-bit timer mode, the overflow output from UC0 is used as the input clock. In any mode other than 16-bit timer mode, the input clock is selectable and can either be one of the internal clocks φT1, φT16, or φT256, or the comparator output (The match detection signal) from TMRA0.

For each interval timer the timer operation control register bits TA01RUN<TA0RUN> and TA01RUN<TA1RUN> can be used to stop and clear the up counters and to control their count. A reset releases both up counters, stopping the timers.

(3) Timer registers (TA0REG and TA1REG)

These are 8-bit registers, which can be used to set a time interval. When the value set in the timer register TA0REG or TA1REG 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 TA0REG are double buffer structure, each of which makes a pair with register buffer0.

The setting of the bit TA01RUN <TA0RDE> determines whether TA0REG's double buffer structure is enabled or disabled. It is disabled if <TA0RDE> = "0" and enabled if <TA0RDE> = "1".

When the double buffer is enabled, data is transferred from the register buffer0 to the timer register0 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 timer mode.

A reset initializes <TA0RDE> to "0", disabling the double buffer. To use the double buffer, write data to the timer register0, set <TA0RDE> to "1", and write the following data to the register buffer0 3.7.5 show the configuration of TA0REG.

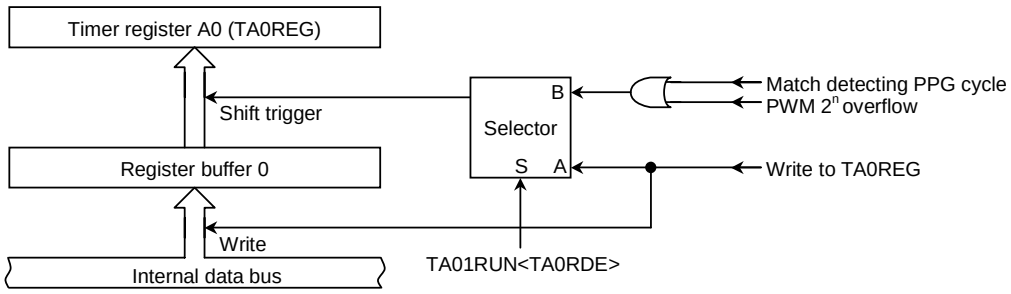


Figure 3.7.5 Timer Register A0 (TA0REG)

Note: The same memory address is allocated to TA0REG and the register buffer0. When <TA0RDE> = "0", the same value is written to the register buffer0 and TA0REG, when <TA0RDE> = 1, only the register buffer0 is written to.

The address of each timer register is as follows.

TA0REG: 001102H TA1REG: 001103H
TA2REG: 00110AH TA3REG: 00110BH
TA4REG: 001112H TA5REG: 001113H
TA6REG: 00111AH TA7REG: 00111BH

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

(4) Comparator (CP0, CP1)

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 0 and an interrupt signal (INTTA0 or INTTA1) is generated. If timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

Note) The timer causes the overflow when the value below the improvement counter value is written in the timer register while the timer is working, and the generation of interrupt by the expected value is not obtained.
(It is possible to operate normally if the changed set value is more than the improvement counter value.)
Moreover, the Compear circuit doesn't operate in writing only 8-bit subordinate position bits when operating in 16-bit mode.
Therefore, please write it in 16-bit in order in 8-bit subordinate position bits and 8-bit high rank bits.

(5) Timer flip-flop (TA1FF)

The timer flip-flop (TA1FF) is a flip-flop inverted by the match detects signal (8-bit comparator output) of each interval timer.

Whether inversion is enabled or disabled is determined by the setting of the bit TA1FFCR <TA1FFIE> in the timer flip-flops control register. A reset clears the value of TA1FF to "0". Programming "01" or "10" to TA1FFCR <TA1FFC1:0> sets TA1FF to 0 or 1. Programming "00" to these bits inverts the value of TA1FF. (This is known as software inversion.)

The TA1FF signal is output via the TA1OUT pin (which can also be used as PF1). When this pin is used as the timer output, the timer flip-flop should be set beforehand using the port F function register PFCR and PFFC.

Inversion of TA1FF by each mode
8-bit timer mode : Agreement of UC0 and TA0REG or agreements of UC1 and TA1REG.
16-bit timer mode: Agreement of UC0 and TA0REG and agreements of UC1 and TA1REG.
8-bit PWM mode : Agreement of overflow or UC0 and TA0REG.
8-bit PPG mode : Agreement of UC0 and TA0REG or agreements of UC0 and TA1REG.
Note) When the change request by inversion and the register setting with the timer is done at the same time, it is necessary to note it because it becomes the following operation by the state at that time.

- When inversion by the timer and inversion by register setup occur simultaneously.
  → Only once inversion.
- When inversion by the timer and "1" set by register setup occur simultaneously.
  → Set to "1".
- When inversion by the timer and "0" clear by register setup occur simultaneously.
  → Clear to "0".

3.7.3 SFRs

TMRA01 Run Register

TA01RUN (1100H)

	7	6	5	4	3	2	1	0
Bit symbol	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
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	TMRA01 prescaler	UC1	UC0
						0: Stop and clear 1: Run (Count up)		

TA0REG double buffer control

0	Disable
1	Enable

Count operation

0	Stop and clear
1	Count

Note: The values of bits 4 to 6 of TA01RUN are undefined when read.

TMRA23 Run Register

TA23RUN (1108H)

	7	6	5	4	3	2	1	0
Bit symbol	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
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	TMRA23 prescaler	UC3	UC2
						0: Stop and clear 1: Run (Count up)		

TA2REG double buffer control

0	Disable
1	Enable

Count operation

0	Stop and clear
1	Count

Note: The values of bits 4 to 6 of TA23RUN are undefined when read.

Figure 3.7.66666666 8-bit timer register(TA01RUN,TA23RUN)

TMRA45 Run Register

TA45RUN (1110H)

	7	6	5	4	3	2	1	0
Bit symbol	TA4RDE				I2TA45	TA45PRUN	TA5RUN	TA4RUN
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	TMRA45 prescaler	UC5	UC4
						0: Stop and clear 1: Run (Count up)		

TA4REG double buffer control

0	Disable
1	Enable

Count operation

0	Stop and clear
1	Count

Note: The values of bits 4 to 6 of TA45RUN are undefined when read.

TMRA67 Run Register

TA67RUN (1118H)

	7	6	5	4	3	2	1	0
Bit symbol	TA6RDE				I2TA67	TA67PRUN	TA7RUN	TA6RUN
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	TMRA67 prescaler	UC7	UC6
						0: Stop and clear 1: Run (Count up)		

TA6REG double buffer control

0	Disable
1	Enable

Count operation

0	Stop and clear
1	Count

Note: The values of bits 4 to 6 of TA67RUN are undefined when read.

Figure 3.7.7777777777777 8-bit timer register(TA45RUN,TA67RUN)

TMRA01 Mode Register

		7	6	5	4	3	2	1	0
TA01MOD (1104H)	Bit symbol	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
	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		TMRA1 source clock 00: TA0TRG 01: φT1 10: φT16 11: φT256		TMRA0 source clock 00: TA0IN pin input (Note) 01: φT1 10: φT4 11: φT16	

TMRA0 input clock

00	TA0IN (External input)
01	φT1 (Prescaler)
10	φT4 (Prescaler)
11	φT16 (Prescaler)

TMRA1 input clock

	TA01MOD <TA01M1:0> ≠ 01	TA01MOD <TA01M1:0> = 01
00	Matching output for TMRA0	Overflow output for TMRA0
01	φT1	
10	φT16	
11	φT256	(16-bit timer mode)

Select cycle in PWM mode

00	Reserved
01	2^6 × Clock source
10	2^7 × Clock source
11	2^8 × Clock source

Select operation mode for TMR01

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA0), 8-bit timer (TMRA1)

Note: When set TA0IN pin, set TA01MOD after set port F0.

Figure 3.7.8(1) 8-bit timer register8888(TA01MOD)

TMRA23 Mode Register

TA23MOD (110CH)

	7	6	5	4	3	2	1	0
Bit symbol	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
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		TMRA3 source clock 00: TA2TRG 01: φT1 10: φT16 11: φT256		TMRA2 source clock 00: TA2IN pin input (Note) 01: φT1 10: φT4 11: φT16	

TMRA2 input clock

00	TA2IN (External input)
01	φT1 (Prescaler)
10	φT4 (Prescaler)
11	φT16 (Prescaler)

TMRA3 input clock

	TA23MOD <TA23M1:0> ≠ 01	TA23MOD <TA23M1:0> = 01
00	Matching output for TMRA2	Overflow output for TMRA2 (16−bit timer mode)
01	φT1	
10	φT16	
11	φT256	

Select cycle in PWM mode

00	Reserved
01	2^6 × Clock source
10	2^7 × Clock source
11	2^8 × Clock source

Select operation mode for TMRA23

00	Two 8-bit timer
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA2), 8-bit timer (TMRA3)

Note: When set TA2IN pin, set TA23MOD after set port F2.

Figure 3.7.8(2)8 8-bit timer register(TA23MOD)

TMRA45 Mode Register

		7	6	5	4	3	2	1	0
TA45MOD (1114H)	Bit symbol	TA45M1	TA45M0	PWM41	PWM40	TA5CLK1	TA5CLK0	TA4CLK1	TA4CLK0
	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		TMRA5 source clock 00: TA4TRG 01: φT1 10: φT16 11: φT256		TMRA4 source clock 00: TA4IN pin input (Note) 01: φT1 10: φT4 11: φT16	

TMRA4 input clock

00	TA4IN (External input)
01	φT1 (Prescaler)
10	φT4 (Prescaler)
11	φT16 (Prescaler)

TMRA5 input clock

	TA45MOD <TA45M1:0>≠01	TA45MOD <TA45M1:0>=01
00	Matching output for TMRA4	Overflow output for TMRA4 (16−bit timer mode)
01	φT1	
10	φT16	
11	φT256	

Select cycle in PWM mode

00	Reserved
01	2^6 × Clock source
10	2^7 × Clock source
11	2^8 × Clock source

Select operation mode for TMRA45

00	Two 8-bit timer
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA4), 8-bit timer (TMRA5)

Note: When set TA4IN pin, set TA45MOD after set port F4.

Figure 3.7.9(3) 8-bit timer register(TA45MOD)

TMRA67 Mode Register

TA67MOD (111CH)	7	6	5	4	3	2	1	0
Bit symbol	TA67M1	TA67M0	PWM61	PWM60	TA7CLK1	TA7CLK0	TA6CLK1	TA6CLK0
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		TMRA7 source clock 00: TA6TRG 01: φT1 10: φT16 11: φT256		TMRA6 source clock 00: TA6IN pin input (Note) 01: φT1 10: φT4 11: φT16	

TMRA6 input clock

00	TA6IN (External input)
01	φT1 (Prescaler)
10	φT4 (Prescaler)
11	φT16 (Prescaler)

TMRA7 input clock

	TA67MOD <TA67M1:0>≠01	TA67MOD <TA67M1:0>=01
00	Matching output for TMRA6	Overflow output for TMRA6 (16−bit timer mode)
01	φT1	
10	φT16	
11	φT256	

Select cycle in PWM mode

00	Reserved
01	2^6 × Clock source
10	2^7 × Clock source
11	2^8 × Clock source

Select operation mode for TMRA67

00	Two 8-bit timer
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA6), 8-bit timer (TMRA7)

Note: When set TA6IN pin, set TA67MOD after set port F6.

Figure 3.7.10(4) 8-bit timer register(TA67MOD)

TMRA1 Flip Flop Control Register

		7	6	5	4	3	2	1	0
TA1FFCR (1105H)	Bit symbol					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
	Read/Write					R/W			
	After reset					1	1	0	0
Read-modify -write instruction is prohibited.	Function					00: Invert TA1FF 01: Set TA1FF to “1” 10: Clear TA1FF to “0” 11: Don’t care		TA1FF control for inversion 0: Disable 1: Enable	TA1FF Inversion signal select 0: TMRA0 1: TMRA1

Inversion signal for timer flip-flop 1 (TA1FF)
(Don’t care except in 8-bit timer mode)

0	Inversion by TMRA0
1	Inversion by TMRA1

TA1FF control for inversion

0	Disable inversion
1	Enable inversion

TFF1 control

00	Invert TA1FF
01	Set TA1FF to “1”
10	Clear TA1FF to “0”
11	Don’t care

Note: The values of bits 4 to 7 of TA1FFCR are undefined when read.

Figure 3.7.11(1) 8-bit timer register(TA1FFCR)

TMRA3 Flip-Flop Control Register

		7	6	5	4	3	2	1	0
TA3FFCR (110DH)	Bit symbol					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
	Read/Write					R/W			
	After reset					1	1	0	0
Read-modify -write instruction is prohibited.	Function					00: Invert TA3FF 01: Set TA3FF to “1” 10: Clear TA3FF to “0” 11: Don’t care		TA3FF control for inversion 0: Disable 1: Enable	TA3FF inversion select 0: TMRA2 1: TMRA3

Inverse signal for timer flip-flop 3 (TA3FF)
(Don’t care except in 8-bit timer mode)

0	Invert TMRA2
1	Invert TMRA3

TA3FF control for inversion

0	Disable inversion
1	Enable inversion

TA3FF control

00	Invert TA3FF
01	Set TA3FF to “1”
10	Clear TA3FF to “0”
11	Don’t care

Note: The values of bits 4 to 7 of TA3FFCR are undefined when read.

Figure 3.7.9(2) 8-bit timer register(TA3FFCR)

TMRA5 Flip-Flop Control Register

		7	6	5	4	3	2	1	0
TA5FFCR (1115H)	Bit symbol					TA5FFC1	TA5FFC0	TA5FFIE	TA5FFIS
	Read/Write					R/W			
	After reset					1	1	0	0
Read-modify -write instruction is prohibited.	Function					00: Invert TA5FF 01: Set TA5FF to “1” 10: Clear TA5FF to “0” 11: Don’t care		TA5FF control for inversion 0: Disable 1: Enable	TA5FF inversion select 0: TMRA4 1: TMRA5

Inverse signal for timer flip-flop 5 (TA5FF)
(Don’t care except in 8-bit timer mode)

0	Invert TMRA4
1	Invert TMRA5

TA5FF control for inversion

0	Disable inversion
1	Enable inversion

TA5FF control

00	Invert TA5FF
01	Set TA5FF to “1”
10	Clear TA5FF to “0”
11	Don’t care

Note: The values of bits 4 to 7 of TA5FFCR are undefined when read.

Figure 3.7.9(3) 8-bit timer register(TA5FFCR)

TMRA7 Flip-Flop Control Register

TA7FFCR (111DH)

	7	6	5	4	3	2	1	0
Bit symbol					TA7FFC1	TA7FFC0	TA7FFIE	TA7FFIS
Read/Write					R/W			
After reset					1	1	0	0
Function					00: Invert TA7FF 01: Set TA7FF to “1” 10: Clear TA7FF to “0” 11: Don’t care		TA7FF control for inversion 0: Disable 1: Enable	TA7FF inversion select 0: TMRA6 1: TMRA7

Read-modify-write instruction is prohibited.

Inverse signal for timer flip-flop 7 (TA7FF)
(Don’t care except in 8-bit timer mode)

0	Invert TMRA6
1	Invert TMRA7

TA7FF control for inversion

0	Disable inversion
1	Enable inversion

TA7FF control

00	Invert TA7FF
01	Set TA7FF to “1”
10	Clear TA7FF to “0”
11	Don’t care

Note: The values of bits 4 to 7 of TA7FFCR are undefined when read.

Figure 3.7.9(4) 8-bit timer register(TA7FFCR)

Timer Register (TA0REG to TA7REG)

Symbol	Address	7	6	5	4	3	2	1	0
TA0REG	1102H	–							
		W							
		Undefined							
TA1REG	1103H	–							
		W							
		Undefined							
TA2REG	110AH	–							
		W							
		Undefined							
TA3REG	110BH	–							
		W							
		Undefined							
TA4REG	1112H	–							
		W							
		Undefined							
TA5REG	1113H	–							
		W							
		Undefined							
TA6REG	111AH	–							
		W							
		Undefined							
TA7REG	111BH	–							
		W							
		Undefined							

Note: Read-modify-write instruction is prohibited for above registers.

Figure 3.7.10 8-bit timer register(TA0REG to TA7REG)

3.7.4 Operation in Each Mode

(1) 8-bit timer mode

Both TMRA0 and TMRA1 can be used independently as 8-bit interval timers.

When set function and count data, TMRA0 and TMRA1 should be stopped.

1. Generating interrupts at a fixed interval (using TMRA1)

To generate interrupts at constant intervals using TMRA1 (INTTA1), first stop TMRA1 then set the operation mode, input clock and a cycle to TA01MOD and TA1REG register, respectively. Then, enable the interrupt INTTA1 and start TMRA1 counting.

Example: To generate an INTTA1 interrupt every 40 μs at fc = 40 MHz, set each register as follows:

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← –	X	X	X	–	–	0	–	Stop TMRA1 and clear it to 0.
TA01MOD	← 0	0	X	X	0	1	–	–	Select 8-bit timer mode and select φT1 (0.2 μs at fc = 40 MHz) as the input clock.
TA1REG	← 1	1	0	0	1	0	0	0	Set 40 μs ÷ φT1 = 200 = C8H to TAREG.
INTETA01	← X	1	0	1	–	–	–	–	Enable INTTA1 and set it to Level 5.
TA01RUN	← –	X	X	X	–	1	1	–	Start TMRA1 counting.

X : Don't care, – : No change

Note: The input clocks for TMRA0 and TMRA1 differ as follows:

TMRA0: Uses TMRA0 input (TA0IN) and can be selected from φT1, φT4, or φT16.

TMRA1: Match output of TMRA0 (TA0TRG) and can be selected from φT1, φT16, φT256.

2. Generating a 50% duty ratio square wave pulse

The state of the timer flip-flop (TA1FF) is inverted at constant intervals and its status output via the timer output pin (TA1OUT).

Example: To output a 1.2 μs square wave pulse from the TA1OUT pin at f_{SYS} = 20 MHz, use the following procedure to make the appropriate register settings. This example uses TMRA1; however, either TMRA0 or TMRA1 may be used.

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← –	X	X	X	–	–	0	–	Stop TMRA1 and clear it to 0.
TA01MOD	← 0	0	X	X	0	1	–	–	Select 8-bit timer mode and select φT1 (0.2 μs at fc = 40 MHz) as the input clock.
TA1REG	← 0	0	0	0	0	0	1	1	Set the timer register to 1.2 μs ÷ φT1 ÷ 2 = 3.
TA1FFCR	← X	X	X	X	1	0	1	1	Clear TA1FF to 0 and set it to invert on the match detect signal from TMRA1.
PFCR	← X	–	–	–	–	–	1	–	Set PF1 to function as the TA1OUT pin
PFFC	← X	–	–	–	–	–	1	–	
TA01RUN	← –	X	X	X	–	1	1	–	Start TMRA1 counting.

X : Don't care, – : No change

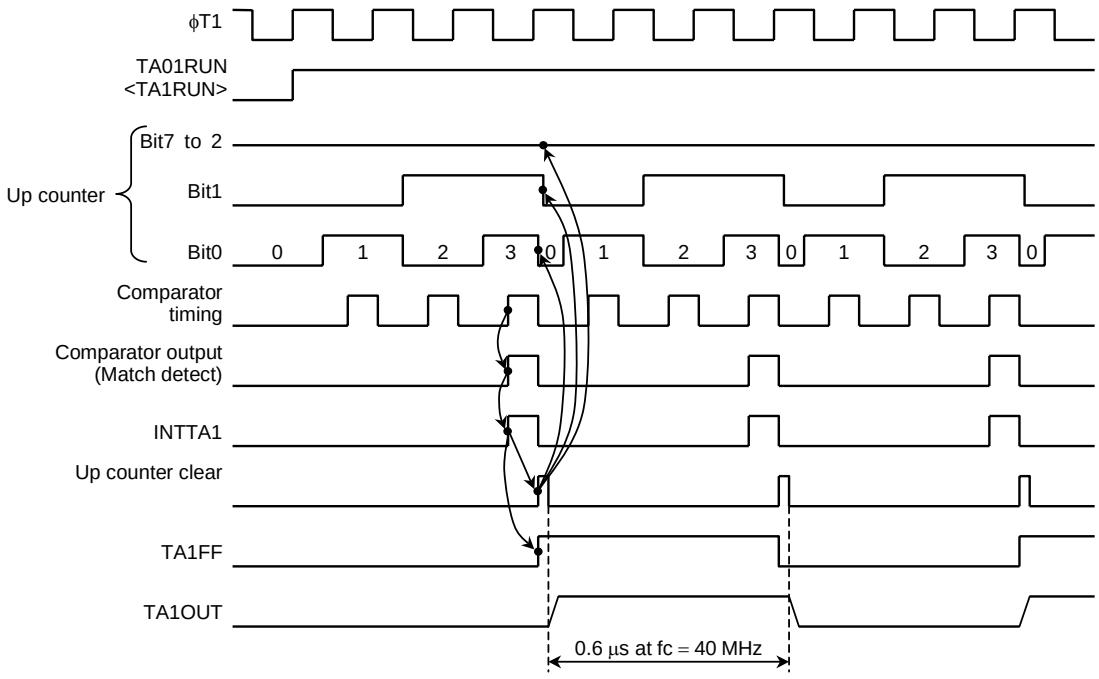


Figure 3.7.11 Square Wave Output Timing Chart (50% duty)

3. Making TMRA1 count up on the match signal from the TMRA0 comparator

Select 8-bit timer mode and set the comparator output from TMRA0 to be the input clock to TMRA1.

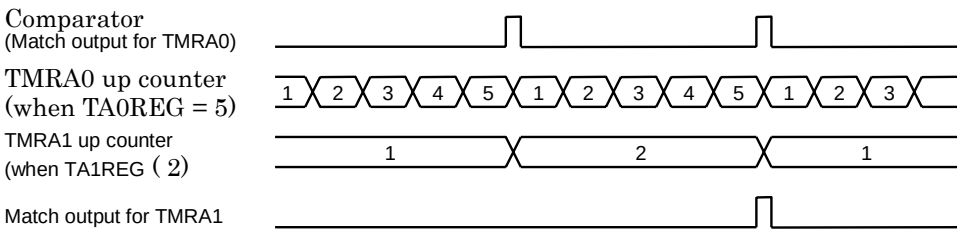


Figure 3.7.12 TMRA1 Count up on Signal from TMRA0

(2) 16-bit timer mode

A 16-bit interval timer is configured by pairing the two 8-bit timers TMRA0 and TMRA1.

To make a 16-bit interval timer, in which TMRA0 and TMRA1 are cascaded together, set TA01MOD <TA01M1:0> to "01".

In 16-bit timer mode, the overflow output from TMRA0 is used as the input clock for TMRA1, regardless of the value set in TA01MOD<TA1CLK1:0> Table 3.7.2 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 TA0REG and the upper eight bits in TA1REG. Be sure to set TA0REG first (As entering data in TA0REG temporarily disables the compare, while entering data in TA1REG starts the compare).

Example: To generate an INTTA1 interrupt every 0.2 s at fc = 40 MHz, set the timer registers TA0REG and TA1REG as follows:

If ϕT16 (3.2 μs at 40 MHz) is used as the input clock for counting, set the following value in the registers:

0.2 s ÷ 3.2 μs = 62500 = F424H;

e.g., set TA1REG to F4H and TA0REG to 24H.

The comparator match signal is output from TMRA0 each time the up-counter UC0 matches TA0REG, though the up-counter UC0 is not be cleared.

In the case of the TMRA1 comparator, the match detect signal is output on each comparator pulse on which the values in the up-counter UC1 and TA1REG match. When the match detect signal is output simultaneously from both the comparator TMRA0 and TMRA1, the up-counters UC0 and UC1 are cleared to "0" and the interrupt INTTA1 is generated. Also, if inversion is enabled, the value of the timer flip-flop TA1FF is inverted.

Example: When TA1REG = 04H and TA0REG = 80H

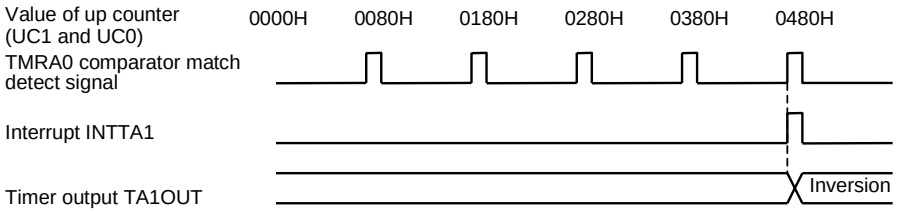


Figure 3.7.14 Timer Output by 16-Bit Timer Mode

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

Square wave pulses can be generated at any frequency and duty ratio by TMRA0. The output pulses may be active-low or active-high. In this mode TMRA1 cannot be used.

TMRA0 outputs pulses on the TA1OUT pin (Shared with PF1).

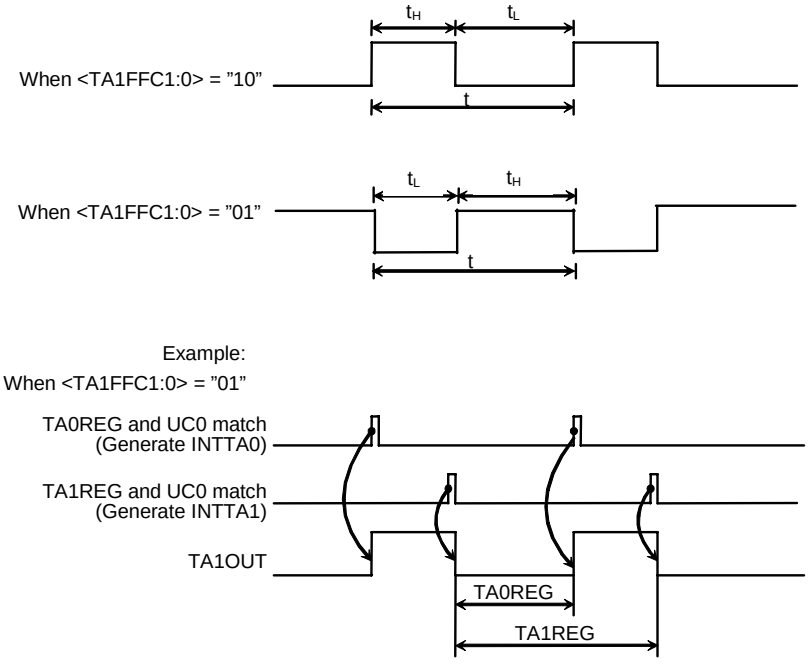


Figure 3.7.15 8-Bit PPG Output Waveforms

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 TA0REG or TA1REG.

The value set in TA0REG must be smaller than the value set in TA1REG.

Although the up-counter for TMRA1 (UC1) is not used in this mode, TA01RUN<TA1RUN> should be set to "1", so that UC1 is set for counting.

Figure 3.7.16 shows a block diagram representing this mode.

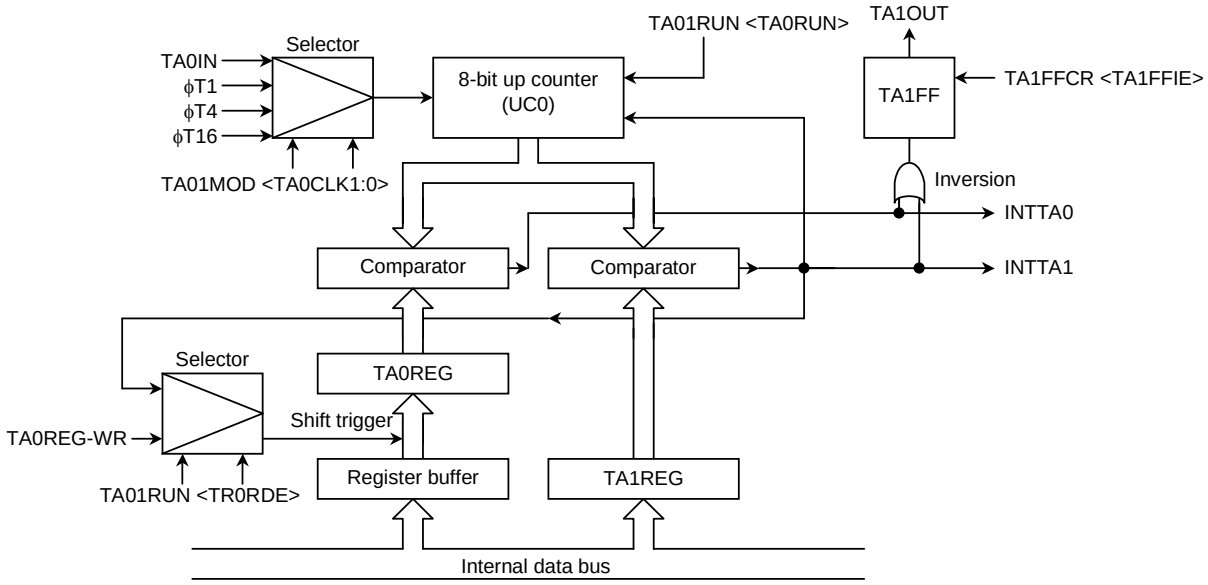


Figure 3.7.16 Block Diagram of 8-Bit PPG Output Mode

If the TA0REG double buffer is enabled in this mode, the value of the register buffer will be shifted into TA0REG each time TA1REG matches UC0.

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

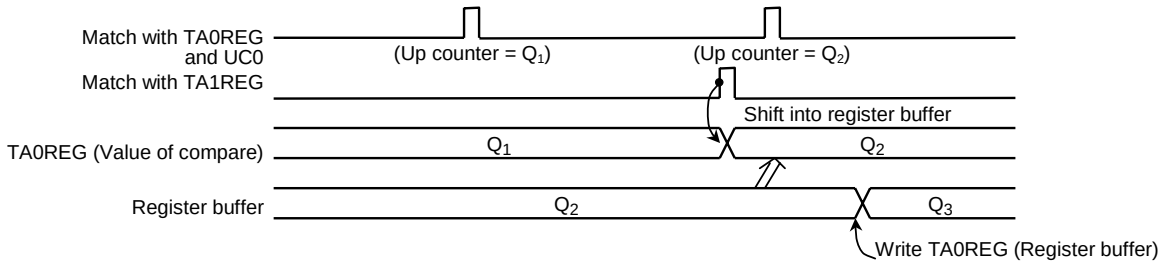
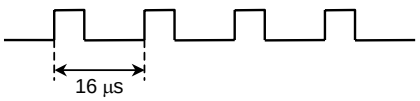


Figure 3.7.17 Operation of Register Buffer0

Example: To generate 1/4 duty 62.5 kHz pulses (at fc = 40 MHz):



Calculate the value that 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 = 0.2 μs (at fc = 40 MHz);

16 μs/0.2 μs = 80

Therefore set TA1REG = 80 = 50H

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

4 μs/0.2 μs = 20

Therefore, set TA0REG = 20 = 14H

		7	6	5	4	3	2	1	0	
TA01RUN	←	0	X	X	X	–	0	0	0	Stop TMRA0 and TMRA1 and clear it to “0”.
TA01MOD	←	1	0	X	X	X	X	0	1	Set the 8-bit PPG mode, and select ϕT1 as input clock.
TA0REG	←	0	0	0	1	0	1	0	0	Write 14H.
TA1REG	←	0	1	0	1	0	0	0	0	Write 50H.
TA1FFCR	←	X	X	X	X	0	1	1	X	Set TA1FF and set inversion to enable. Writing “10” provides negative logic pulse.
PFCR	←	X	–	–	–	–	–	–	1	Set PF1 to TA1OUT pin.
PFFC	←	X	–	–	–	–	–	–	1	
TA01RUN	←	1	X	X	X	–	1	1	1	Start TMRA0 and TMRA1 counting.

X : Don’t care, – : No change

(4) 8-bit PWM (Pulse width modulation) output mode

This mode is only valid for TMRA0. In this mode, a PWM pulse with the maximum resolution of 8 bits can be output.

When TMRA0 is used the PWM pulse is output on the TA1OUT pin (which is also used as PF1). TMRA1 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 TA0REG or when 2^n counter overflow occurs (n = 6, 7, or 8 as specified by TA01MOD <PWM01:00>). 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 TA0REG < Value of set for 2^n counter overflow

Value set in TA0REG ≠ 0

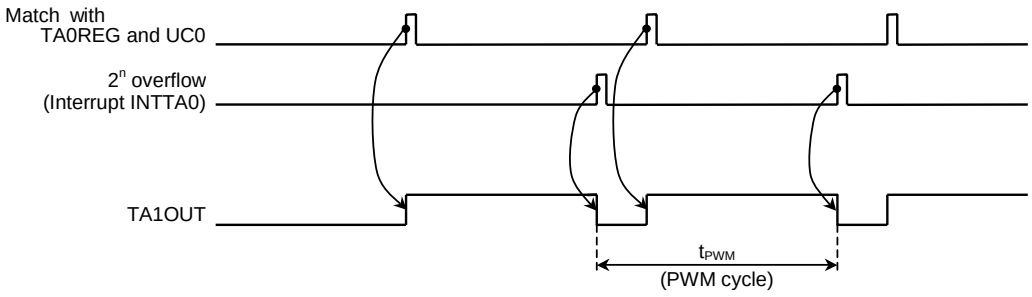


Figure 3.7.18 8-Bit Output Wave Form

Figure 3.7.19 shows a block diagram representing this mode.

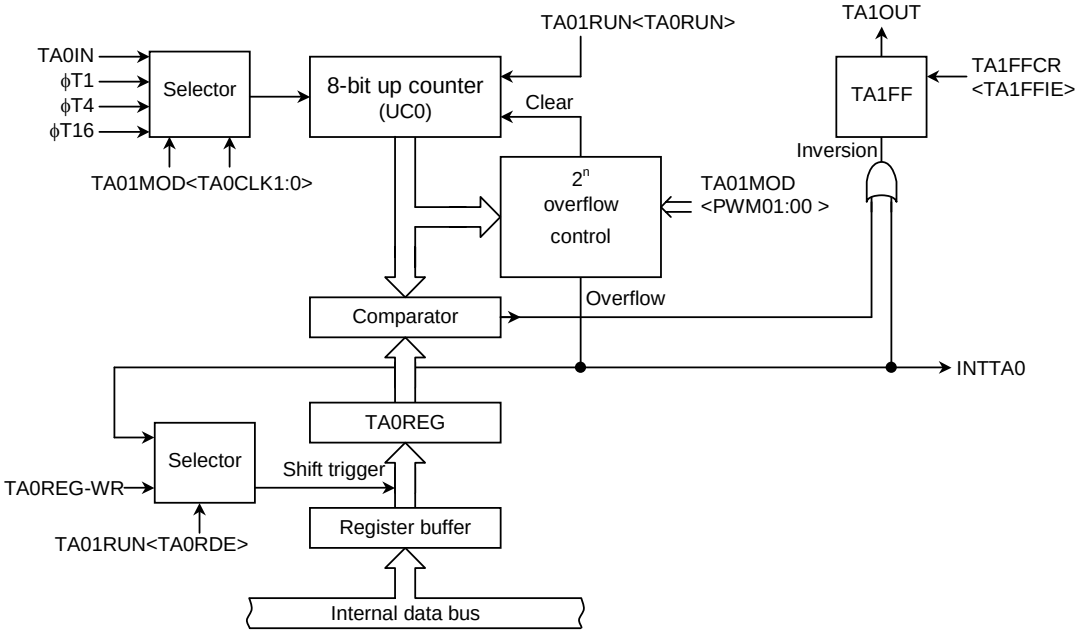


Figure 3.7.19 Block Diagram of 8-Bit PWM Output Mode

In this mode, the value of the register buffer will be shifted into TA0REG if 2^n overflow is detected when the TA0REG double buffer is enabled.

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

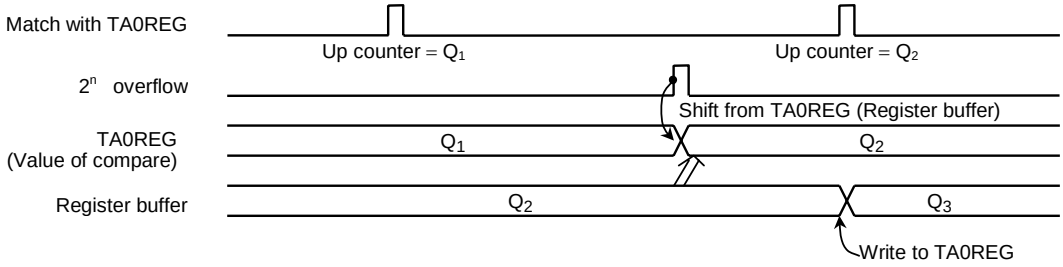
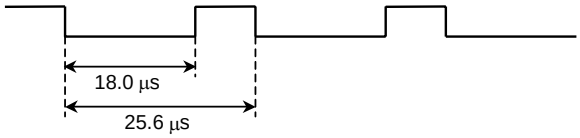


Figure 3.7.20 Operation of Register Buffer

Example: To output the following PWM waves on the TA1OUT pin at fc = 40 MHz:



To achieve a 25.6 μs PWM cycle by setting ϕT1 to 0.2 μs (at fc = 40 MHz):

25.6 μs/0.2 μs = 128 = 2^n

Therefore n should be set to 7.

Since the low-level period is 18.0 μs when ϕT1 = 0.2 μs,

set the following value for TA0REG:

18.0 μs/0.2 μs = 90 = 5AH

	MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	← –	X	X	X	–	–	–	0	Stop TMRA0 and clear it to 0.
TA01MOD	← 1	1	1	0	–	–	0	1	Select 8-bit PWM mode (cycle: 2^7) and select ϕT1 as the input clock.
TA0REG	← 0	1	0	1	1	0	1	0	Write 5AH.
TA1FFCR	← X	X	X	X	1	0	1	X	Clear TA1FF to 0; set inversion to enable.
PFCR	← X	–	–	–	–	–	–	1	Set PF1 to TA1OUT pin.
PFFC	← X	–	–	–	–	–	–	1	
TA01RUN	← 1	X	X	X	–	1	–	1	Start TMRA0 counting.

X : Don't care, – : No change

Table 3.7.3 Relationship of PWM Cycle and 2^n Counter

@fc = 40 MHz

System clock selection <SYSCK>	Clock gear value <GEAR2:0>	PWM Cycle								
		2^6			2^7			2^8		
		φT1	φT4	φT16	φT1	φT4	φT16	φT1	φT4	φT16
0 (fsys)	000 (fc)	12.8μs	51.2μs	204.8μs	25.6μs	102.4μs	409.6μs	51.2μs	204.8μs	819.2μs
	001 (fc/2)	25.6μs	102.4μs	409.6μs	51.2μs	204.8μs	819.2μs	102.4μs	409.6μs	1.63ms
	010 (fc/4)	51.2μs	204.8μs	819.2μs	102.4μs	409.6μs	1.63ms	204.8μs	819.2μs	3.27ms
	011 (fc/8)	102.4μs	409.6μs	1.63ms	204.8μs	819.2μs	3.27ms	409.6μs	1.63ms	6.55ms
	100 (fc/16)	204.8μs	819.2μs	3.27ms	409.6μs	1.63ms	6.55ms	819.2μs	3.27ms	13.1ms

XXX: Don't care

(5) Mode settings

Table 3.7.4 shows the SFR settings for each mode.

Table 3.7.4 Timer Mode Setting Registers

Register Name	TA01MOD				TA1FFCR
<Bit symbol>	<TA01M1:0>	<PWM01:00>	<TA1CLK1:0>	<TA0CLK1:0>	<TA1FFIS>
Function	Timer mode	PWM cycle	Upper timer input clock	Lower timer input clock	Timer F/F Inversion Signal select
8-bit timer × 2 channels	00	–	Lower timer match, φT1, φT16, φT256 (00, 01, 10, 11)	External, φT1, φT4, φT16 (00, 01, 10, 11)	0: Lower timer output 1: Upper timer output
16-bit timer mode	01	–	–	External, φT1, φT4, φT16 (00, 01, 10, 11)	–
8-bit PPG × 1 channel	10	–	–	External, φT1, φT4, φT16 (00, 01, 10, 11)	–
8-bit PWM × 1 channel	11	2^6, 2^7, 2^8 (01, 10, 11)	–	External, φT1, φT4, φT16 (00, 01, 10, 11)	–
8-bit timer × 1 channel	11	–	φT1, φT16, φT256 (01, 10, 11)	–	Output disable

– : Don't care

3.8 16-Bit Timer/Event Counters (TMRB)

The TMP92CM27 contains 6 channels 16-bit timer/event counter (TMRB0 to TMRB5) which have the following operation modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable square wave pulse generation output mode (PPG: Variable duty cycle with variable period)

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 diagram of TMRB0 to TMRB5. Each timer/event counter 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 11-byte control register (SFR).

Each of the six modules (TMRB0 to TMRB5) can be operated independently. All modules operate in the same manner; hence only the operation of TMRA01 is explained here.

Table 3.8.1 Pins and SFR of TMRB

Spec \ Channel		TMRB0	TMRB1	TMRB2	TMRB3	TMRB4	TMRB5
External pin	External clock/ Capture trigger input pin	TB0IN0 TB0IN1	TB1IN0 TB1IN1	TB2IN0 TB2IN1	TB3IN0 TB3IN1	None	None
	Timer flip-flop output pin	TB0OUT0 TB0OUT1	TB1OUT0 TB1OUT1	TB2OUT0 TB2OUT1	TB3OUT0 TB3OUT1	TB4OUT0 TB4OUT1	TB5OUT0 TB5OUT1
SFR	Timer run register	TB0RUN	TB1RUN	TB2RUN	TB3RUN	TB4RUN	TB5RUN
	Timer mode register	TB0MOD	TB1MOD	TB2MOD	TB3MOD	TB4MOD	TB5MOD
	Timer flip-flop control register	TB0FFCR	TB1FFCR	TB2FFCR	TB3FFCR	TB4FFCR	TB5FFCR
	Timer register	TB0RG0L	TB1RG0L	TB2RG0L	TB3RG0L	TB4RG0L	TB5RG0L
		TB0RG0H	TB1RG0H	TB2RG0H	TB3RG0H	TB4RG0H	TB5RG0H
		TB0RG1L	TB1RG1L	TB2RG1L	TB3RG1L	TB4RG1L	TB5RG1L
		TB0RG1H	TB1RG1H	TB2RG1H	TB3RG1H	TB4RG1H	TB5RG1H
	Capture register	TB0CP0L	TB1CP0L	TB2CP0L	TB3CP0L	TB4CP0L	TB5CP0L
		TB0CP0H	TB1CP0H	TB2CP0H	TB3CP0H	TB4CP0H	TB5CP0H
		TB0CP1L	TB1CP1L	TB2CP1L	TB3CP1L	TB4CP1L	TB5CP1L
		TB0CP1H	TB1CP1H	TB2CP1H	TB3CP1H	TB4CP1H	TB5CP1H
External signal	Capture trigger input signal	TA1OUT	TA1OUT	TA3OUT	TA3OUT	TA5OUT	TA5OUT
Interrupt	Timer interrupt	INTTB00	INTTB10	INTTB20	INTTB30	INTTB40	INTTB50
		INTTB01	INTTB11	INTTB21	INTTB31	INTTB41	INTTB51
	Timer overflow interrupt	INTTBOF0	INTTBOF1	INTTBOF2	INTTBOF3	INTTBOF4	INTTBOF5

Note 1) Since TB2OUT0/TB4OUT0, TB2OUT1/TB4OUT1, TB3OUT0/TB5OUT0, and TB3OUT1/TB5OUT1 are making the output terminal serve a double purpose, they cannot be used simultaneously.

Note 2) Since INTTB30/INTTB31,INTTB40/INTTB41 and INTTB50/INTTB51 are making the interruption factor serve a double purpose, they cannot be used simultaneously.

Note 3) Although INTTBOF0/INTTBOF1/INTTBOF2/INTTBOF3/INTTBOF4/INTTBOF5 is making the interruption factor serve a double purpose, it can be used simultaneously. Which interruption occurred should lead an INTST register.

This chapter consists of the following items:

3.8.1 Block Diagram

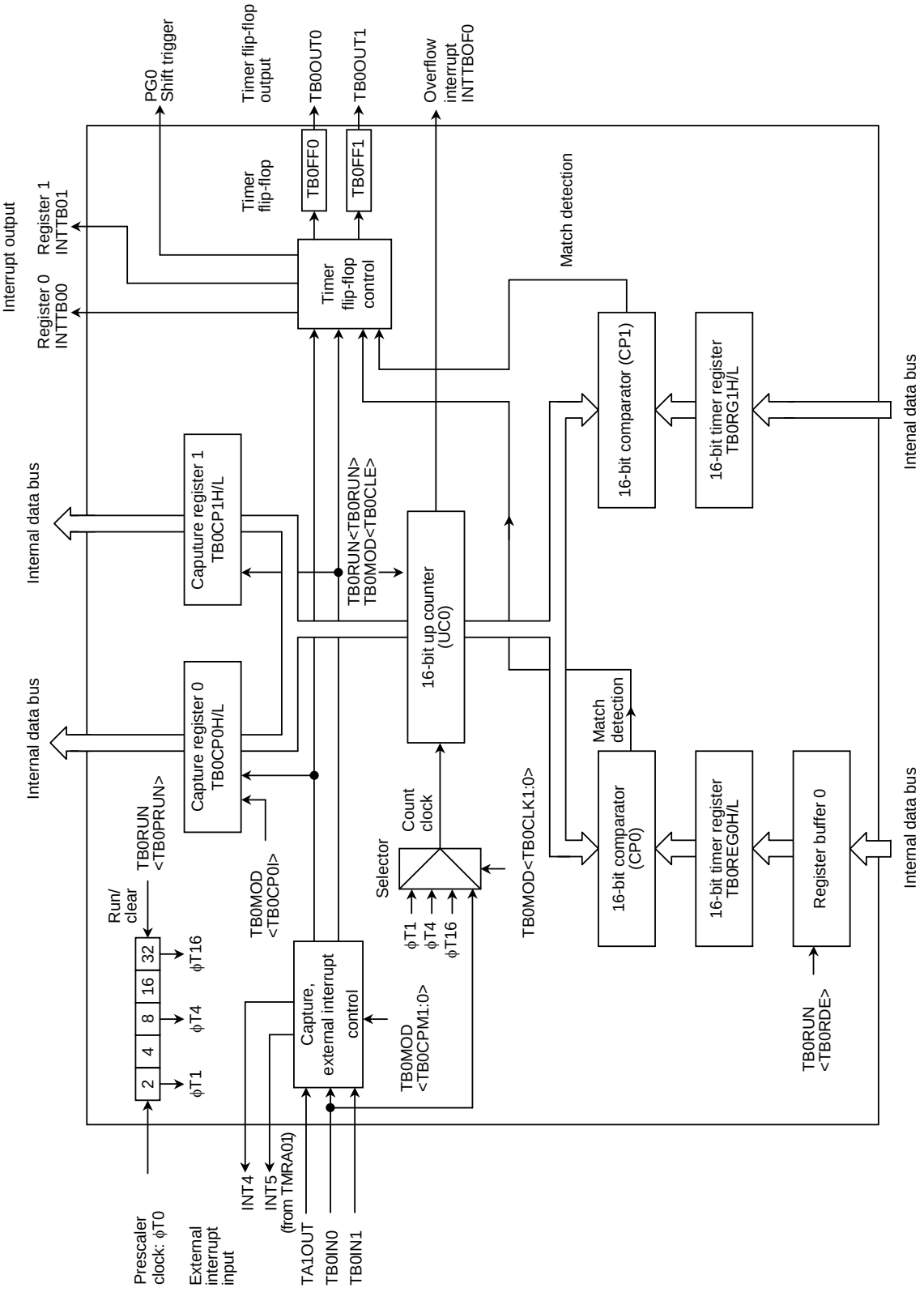


Figure 3.8.1 Block Diagram of TMRB0

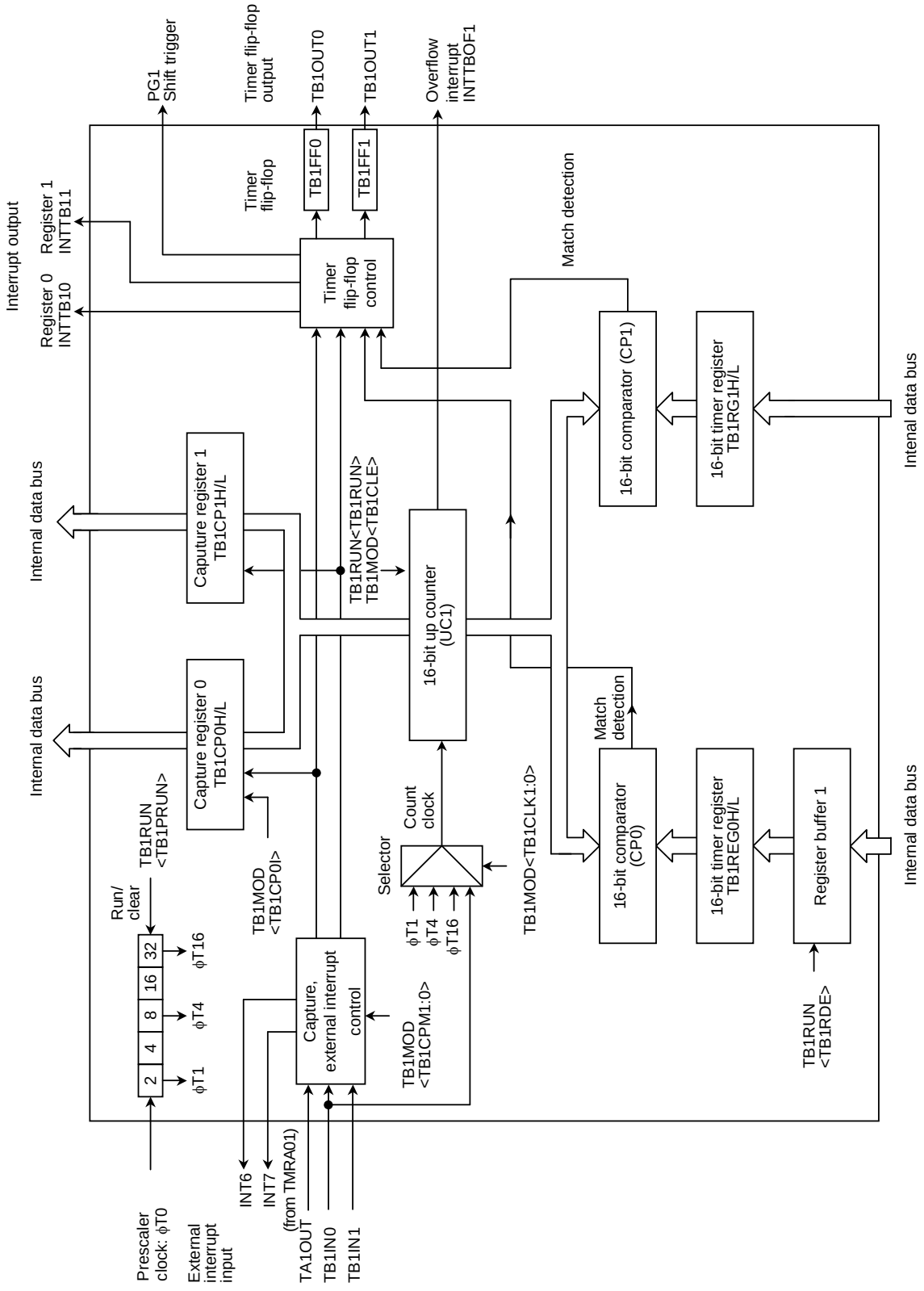


Figure 3.8.2 Block Diagram of TMRB1

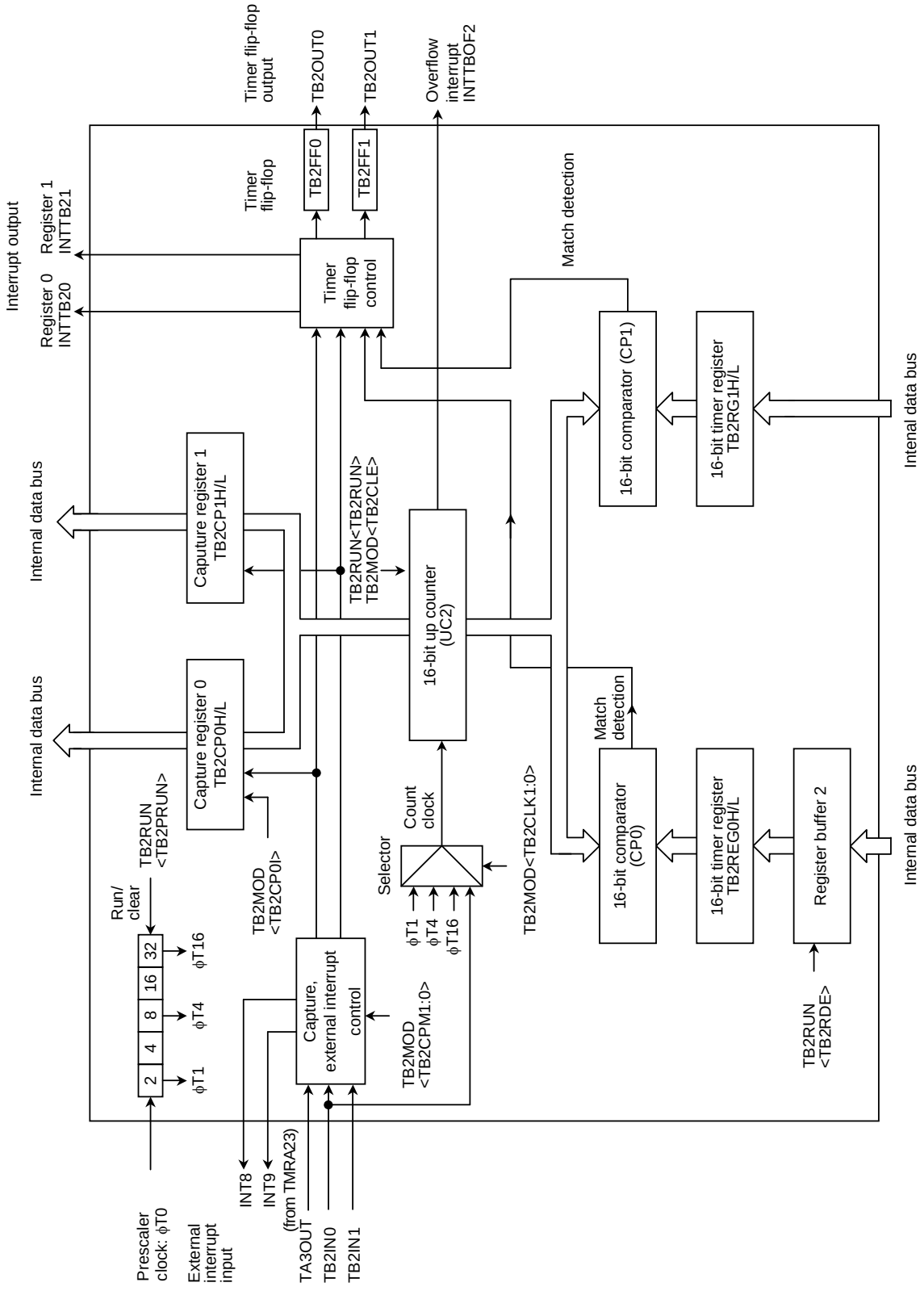


Figure 3.8.3 Block Diagram of TMRB2

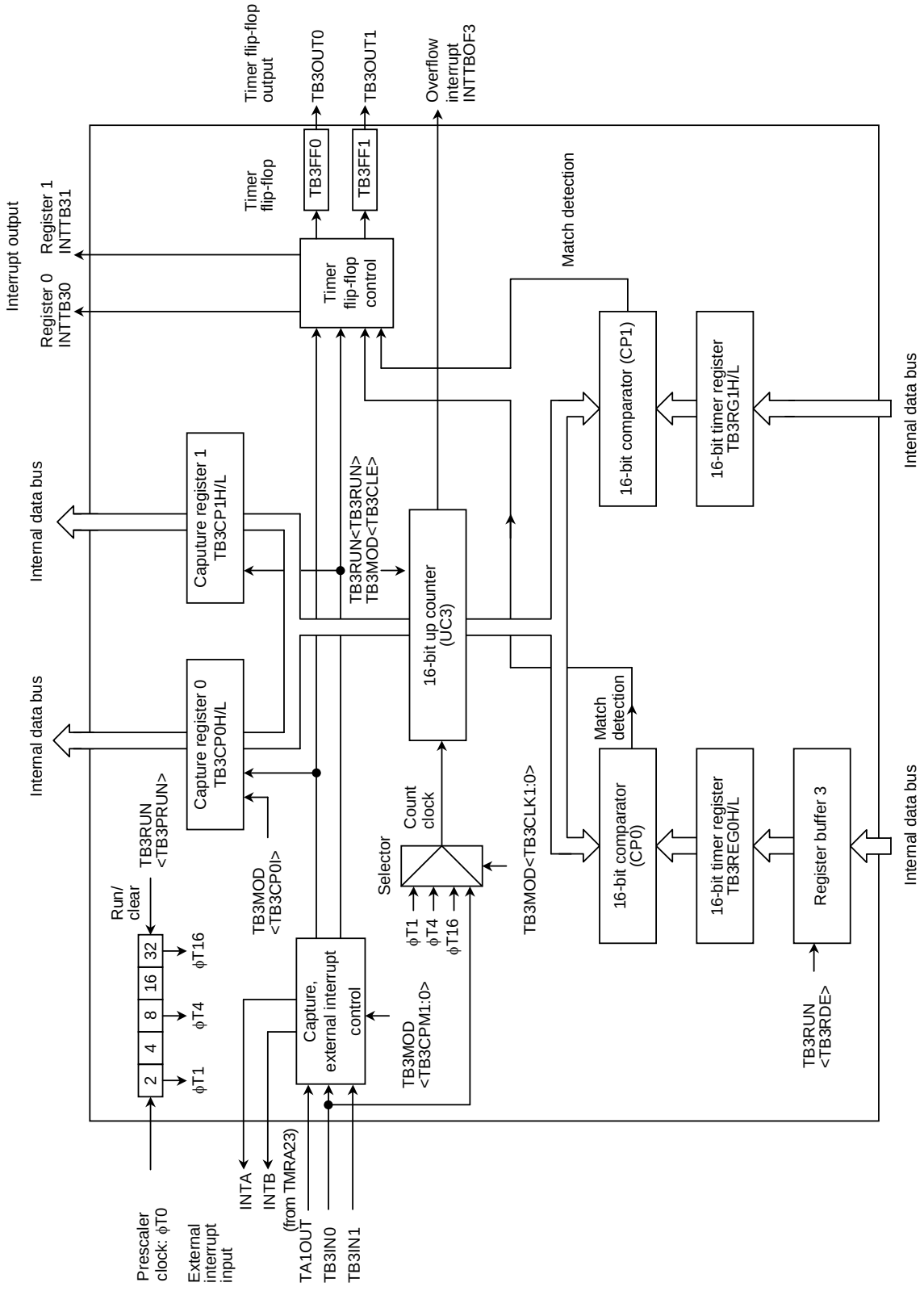


Figure 3.8.4 Block Diagram of TMRB3

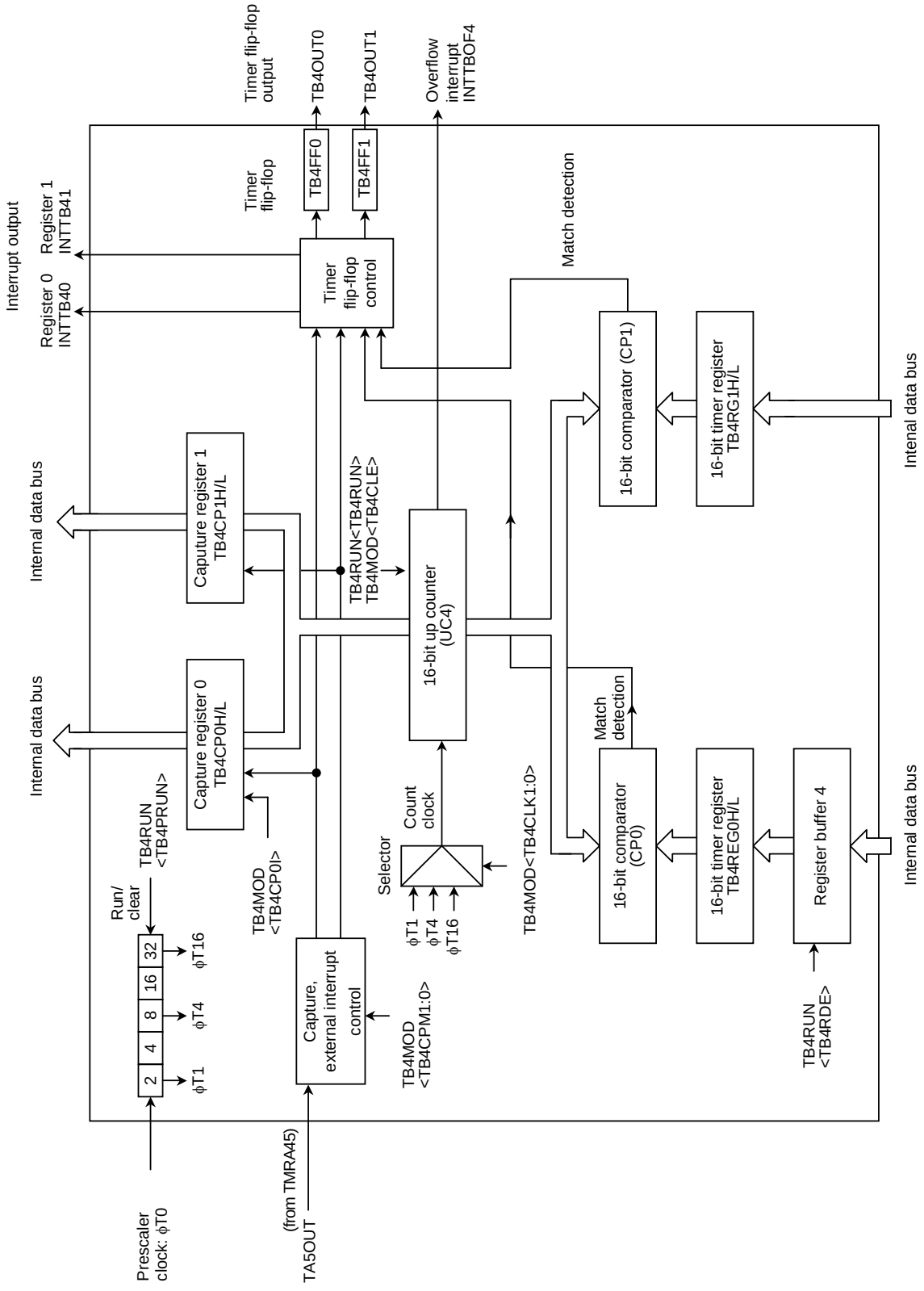


Figure 3.8.5 Block Diagram of TMRB4

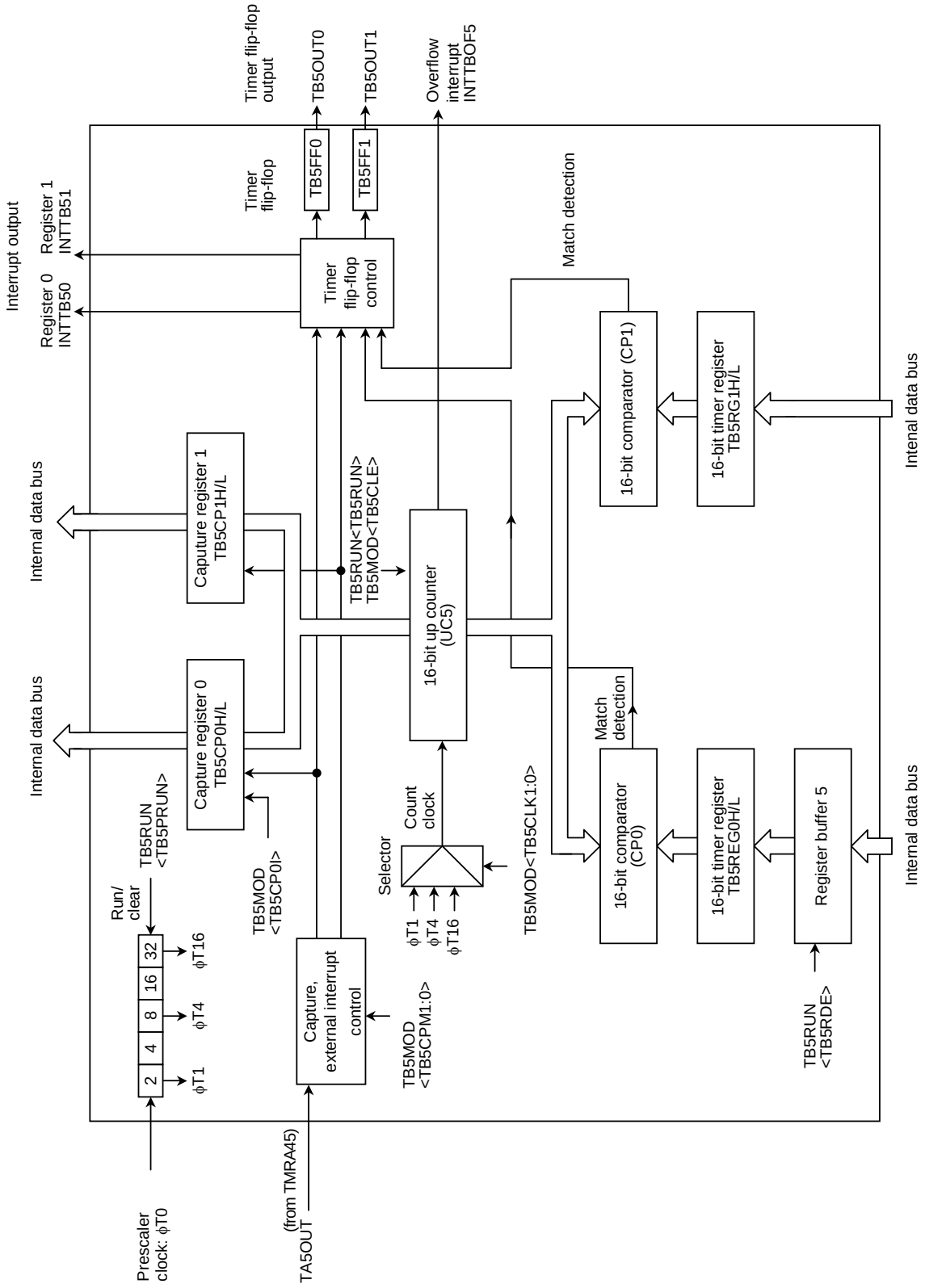


Figure 3.8.6 Block Diagram of TMRB5

3.8.2 Operation

(1) Prescaler

The 5-bit prescaler generates the source clock for TMRB1. Input clock ϕT0 to Priscara is a clock that was four dividing fFPH.

This prescaler can be started or stopped using TB0RUN<TB0PRUN>. Counting starts when <TB0PRUN> is set to 1; the prescaler is cleared to zero and stops operation when <TB0PRUN> is cleared to 0.

Table 3.8.2 show prescaler output clock resolution.

Table 3.8.2 Prescaler Output Clock Resolution

at fc = 40 MHz

Gear Value SYSCR1 <GEAR2:0>	Cycle		
	ϕT1	ϕT4	ϕT16
000 (fc)	2^3/fc (0.2 μs)	2^5/fc (0.8 μs)	2^7/fc (3.2 μs)
001 (fc/2)	2^4/fc (0.4 μs)	2^6/fc (1.6 μs)	2^8/fc (6.4 μs)
010 (fc/4)	2^5/fc (0.8 μs)	2^7/fc (3.2 μs)	2^9/fc (12.8 μs)
011 (fc/8)	2^6/fc (1.6 μs)	2^8/fc (6.4 μs)	2^{10}/fc (25.6 μs)
100 (fc/16)	2^7/fc (3.2 μs)	2^9/fc (12.8 μs)	2^{11}/fc (51.2 μs)

xxx: Don't care

(2) Up counter (UC0)

UC0 is a 16-bit binary counter that counts up according to input from the clock specified by TB0MOD<TB0CLK1:0> register.

As the input clock, one of the prescaler internal clocks ϕT1, ϕT4, and ϕT16 can be selected. Counting or stopping and clearing of the counter is controlled by timer operation control register TB0RUN<TB0RUN>. And an external clock from TB0IN0 pin can be selected in TB0MOD.

When clearing is enabled, the up counter UC0 will be cleared to zero each time its value matches the value in the timer register TB0RG1H/L. Clearing can be enabled or disabled using TB0MOD<TB0CLE>.

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

A timer overflow interrupt (INTTBO0) is generated when UC0 overflow occurs.

(3) Timer registers (TB0RG0H/L and TB0RG1H/L)

These two 16-bit registers are used to set the interval time. When the value in the up counter UC0 matches the value set in this timer register, the comparator match detect signal will go active.

Setting data for both upper and lower timer registers TB0RG0H/L and TB0RG1H/L is always needed. For example, either using 2-byte data transfer instruction or using 1-byte data transfer instruction twice for lower 8 bits and upper 8 bits in order.

The TB0RG0H/L timer register has a double-buffer structure, which is paired with register buffer 0. The value set in TB0RUN<TB0RDE> determines whether the double-buffer structure is enabled or disabled: It is disabled when <TB0RDE> = 0, and enabled when <TB0RDE> = 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 (UC0) and the timer register TB0RG1 match.

After a reset, TB0RG0H/L and TB0RG1H/L are undefined. If the 16-bit timer is to be used after a reset, data should be written to it beforehand.

On a reset <TB0RDE> is initialized to 0, disabling the double buffer. To use the double buffer, write data to the timer register, set <TB0RDE> to 1, then write data to the register buffer as shown below.

TB0RG0H/L and the register buffer both have the same memory addresses (001189H and 001188H) allocated to them. If <TB0RDE> = 0, the value is written to both the timer register and the register buffer. If <TB0RDE> = 1, the value is written to the register buffer only.

The addresses of the timer registers are as follows:

TMRB0

TB0RG0H/L		TB0RG1H/L	
Upper 8 bits (TB0RG0H)	Lower 8 bits (TB0RG0L)	Upper 8 bits (TB0RG1H)	Lower 8 bits (TB0RG1L)
1189H	1188H	118BH	118AH

TMRB1

TB1RG0H/L		TB1RG1H/L	
Upper 8 bits (TB1RG0H)	Lower 8 bits (TB1RG0L)	Upper 8 bits (TB1RG1H)	Lower 8 bits (TB1RG1L)
1199H	1198H	119BH	119AH

TMRB2

TB2RG0H/L		TB2RG1H/L	
Upper 8 bits (TB2RG0H)	Lower 8 bits (TB2RG0L)	Upper 8 bits (TB2RG1H)	Lower 8 bits (TB2RG1L)
11A9H	11A8H	11ABH	11AAH

TMRB3

TB3RG0H/L		TB3RG1H/L	
Upper 8 bits (TB3RG0H)	Lower 8 bits (TB3RG0L)	Upper 8 bits (TB3RG1H)	Lower 8 bits (TB3RG1L)
11B9H	11B8H	11BBH	11BAH

TMRB4

TB4RG0H/L		TB4RG1H/L	
Upper 8 bits (TB4RG0H)	Lower 8 bits (TB4RG0L)	Upper 8 bits (TB4RG1H)	Lower 8 bits (TB4RG1L)
11C9H	11C8H	11CBH	11CAH

TMRB5

TB5RG0H/L		TB5RG1H/L	
Upper 8 bits (TB5RG0H)	Lower 8 bits (TB5RG0L)	Upper 8 bits (TB5RG1H)	Lower 8 bits (TB5RG1L)
11D9H	11D8H	11DBH	11DAH

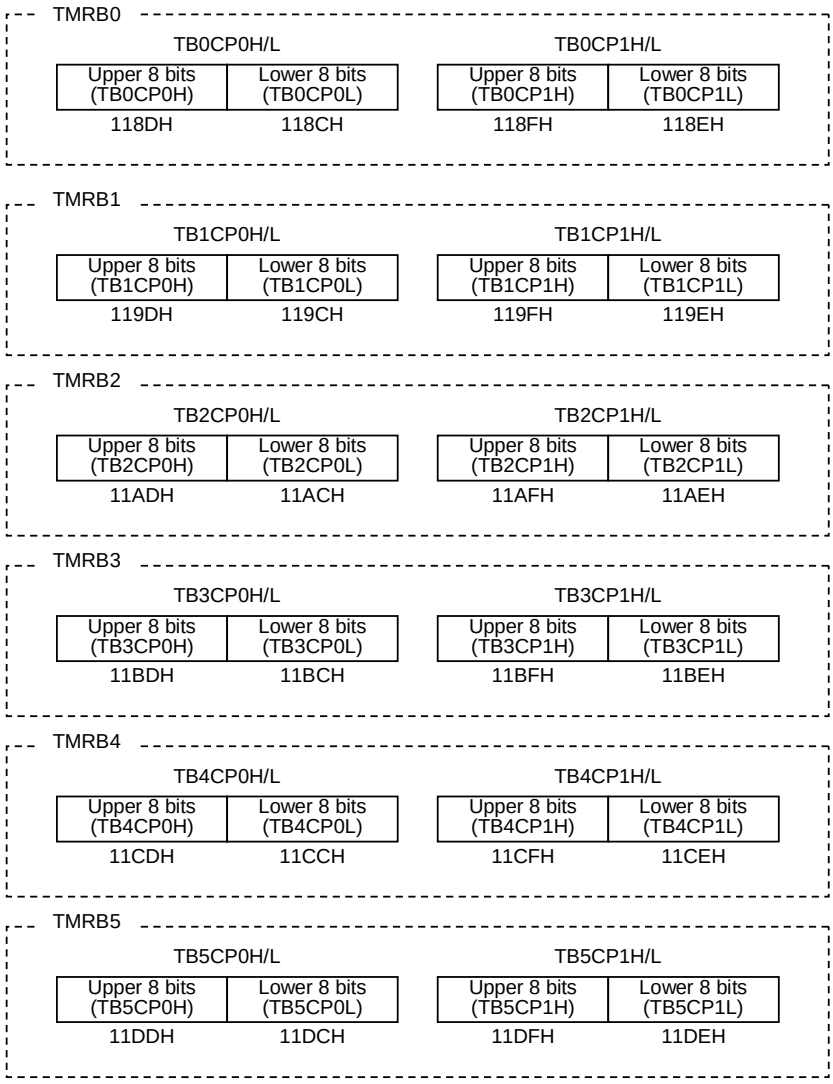
The timer registers are write-only registers and thus cannot be read.

(4) Capture registers

These 16-bit registers are used to latch the values in the up counters UC0.

Data in the capture registers should be read both upper and lower all 16 bits. For example, using 2-byte data transfer instruction or using 1-byte data transfer instruction twice for lower 8 bits and upper 8 bits in order.

The addresses of the capture registers are as follows:



The capture registers are read-only registers and thus cannot be written.

(5) Capture and external interrupt control

This circuit controls the timing to latch the value of up counter UC0 into TB0CP0H/L, TB0CP1H/L and generating for external interrupt.

Interrupt timing of capture register and selection edge of external interrupt are set by TB0MOD<TB0CPM1:0>. (TMRB4 and TMRB5 does not include the selection edge of external interrupt.)

External interrupt INT5 is fixed to the rising edge.

The value in the up counter (UC0) can be loaded into a capture register by software. Whenever 0 is programmed to TB0MOD<TB0CP0I>, the current value in the up counter is loaded into capture register TB0CP0H/L. It is necessary to keep the prescaler in Run mode (e.g., TB0RUN<TB0PRUN> must be held at a value of 1).

Note) External interrupt can be controlled with this control circuit by seeing when the port setting is set to input function (TB0IN0) of TMRB0. When the port setting is set to INT4, it controls by interrupt input mode control 1 and 2(IIMC1,IIMC2).

(6) Comparators (CP0 and CP1)

CP0 and CP1 are 16-bit comparators which compare the value in the up counter UC0 with the value set in TB0RG0H/L or TB0RG1H/L respectively, in order to detect a match. If a match is detected, the comparator generates an interrupt (INTTB00 or INTTB01 respectively).

(7) Timer flip-flop (TB0FF0 and TB0FF1)

These flip-flops (TB0FF0 and TB0FF1) 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 TB0FFCR<TB0C1T1, TB0C0T1, TB0E1T1, TB0E0T1>. Moreover, control of TB0FF0 and TB0FF1 is controllable by TB0MOD<TB0CT1, TB0ET1>.

After a reset the values of TB0FF0 and TB0FF1 are undefined. If “00” is programmed to TB0FFCR<TB0FF0C1:0> or <TB0FF1C1:0>, TB0FF0 will be inverted. If “01” is programmed to the capture registers, the value of TB0FF0 will be set to “1”. If “10” is programmed to the capture registers, the value of TB0FF0 will be cleared to “0”.

The values of TB0FF0 and TB0FF1 can be output via the timer output pins TB0OUT0 (which is shared with PJ0), TB0OUT1 (which is shard with PJ1). Because the timer output terminal of TMRB2/TMRB3 and TMRB4/TMRB5 uses the terminal combinedly, it is not possible to use it at the same time. Timer output should be specified using the port function register.

3.8.3 SFRs

TMRB0 Run Register

TB0RUN (1180H)

	7	6	5	4	3	2	1	0
Bit symbol	TB0RDE	–			I2TB0	TB0PRUN		TB0RUN
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	Always write "0".			IDLE2 0: Stop 1: Operate	Operation of prescaler 0: Stop and clear 1: Run (Count)		Operation of Up counter

Up counter operation for TMRB0

0	Stop & Clear
1	Count

Control of double buffer

0	Disable
1	Enable

Control at the time of the IDLE2 mode

0	Stop
1	Operate

Prescaler operation for TMRB0

0	Stop & Clear
1	Count

Note: The values of bits 1, 4 and 5 of TB0RUN are undefined when read

TMRB1 Run Register

TB1RUN (1190H)

	7	6	5	4	3	2	1	0
Bit symbol	TB1RDE	–			I2TB1	TB1PRUN		TB1RUN
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	Always write "0".			IDLE2 0: Stop 1: Operate	Operation of prescaler 0: Stop and clear 1: Run (Count)		Operation of Up counter

Up counter operation for TMRB1

0	Stop & Clear
1	Count

Control of double buffer

0	Disable
1	Enable

Control at the time of The IDLE2 mode

0	Stop
1	Operate

Prescaler operationfor TMRB1

0	Stop & Clear
1	Count

Note: The values of bits 1, 4 and 5 of TB1RUN are undefined when read

Figure 3.8.7 Register for TMRB (1)

TMRB2 Run Register

TB2RUN (11A0H)

	7	6	5	4	3	2	1	0
Bit symbol	TB2RDE	–			I2TB2	TB2PRUN		TB2RUN
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	Always write "0".			IDLE2 0: Stop 1: Operate	Operation of prescaler 0: Stop and clear 1: Run (Count)		Operation of Up counter

Up counter operation for TMRB2

0	Stop & Clear
1	Count

Control of double buffer

0	Disable
1	Enable

Control at the time of the IDLE2 mode

0	Stop
1	Operate

Prescaler operation for TMRB2

0	Stop & Clear
1	Count

Note: The values of bits 1, 4 and 5 of TB2RUN are undefined when read

TMRB3 Run Register

TB3RUN (11B0H)

	7	6	5	4	3	2	1	0
Bit symbol	TB3RDE	–			I2TB3	TB3PRUN		TB3RUN
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	Always write "0".			IDLE2 0: Stop 1: Operate	Operation of prescaler 0: Stop and clear 1: Run (Count)		Operation of Up counter

Up counter operation for TMRB3

0	Stop & Clear
1	Count

Control of double buffer

0	Disable
1	Enable

Control at the time of The IDLE2 mode

0	Stop
1	Operate

Prescaler operationfor TMRB3

0	Stop & Clear
1	Count

Note: The values of bits 1, 4 and 5 of TB3RUN are undefined when read

Figure 3.8.8 Register for TMRB (2)

TMRB4 Run Register

TB4RUN (11C0H)

	7	6	5	4	3	2	1	0
Bit symbol	TB4RDE	–			I2TB4	TB4PRUN		TB4RUN
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	Always write "0".			IDLE2 0: Stop 1: Operate	Operation of prescaler 0: Stop and clear 1: Run (Count)		Operation of Up counter

Control of double buffer

0	Disable
1	Enable

Control at the time of the IDLE2 mode

0	Stop
1	Oparate

Up counter operation for TMRB4

0	Stop & Clear
1	Count

Prescaler operation for TMRB4

0	Stop & Clear
1	Count

Note: The values of bits 1, 4 and 5 of TB4RUN are undefined when read

TMRB5 Run Register

TB5RUN (11D0H)

	7	6	5	4	3	2	1	0
Bit symbol	TB5RDE	–			I2TB5	TB5PRUN		TB5RUN
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	Always write "0".			IDLE2 0: Stop 1: Operate	Operation of prescaler 0: Stop and clear 1: Run (Count)		Operation of Up counter

Control of double buffer

0	Disable
1	Enable

Control at the time of The IDLE2 mode

0	Stop
1	Oparate

Up counter operation for TMRB5

0	Stop & Clear
1	Count

Prescaler operationfor TMRB5

0	Stop & Clear
1	Count

Note: The values of bits 1, 4 and 5 of TB5RUN are undefined when read

Figure 3.8.9 Register for TMRB (3)

TMRB0 Mode Register

TB0MOD (1182H)

Read-modify -write instruction is prohibited

	7	6	5	4	3	2	1	0
Bit symbol	TB0CT1	TB0ET1	TB0CP0I	TB0CPM1	TB0CPM0	TB0CLE	TB0CLK1	TB0CLK0
Read/Write	R/W		W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	TB0FF1 Inversion trigger 0: Trigger disable 1: Trigger enable		Software capture control 0: Software capturer 1: Undefined	Capture timing 00: Disable INT4 is rising edge 01: TB0N0 ↑ TB0IN1 ↑ INT4 is rising edge 10: TB0IN0 ↑ TB0IN0 ↓ INT4 is falling edge 11: TA1OUT ↑ TA1OUT ↓ INT4 is rising edge		Up counter control 0:disable 1:enable	TMRB0 source clock 00: TB0IN0 pin input 01: ϕT1 10: ϕT4 11: ϕT16	
	Invert when capture to capture register 1	Invert when match UC0 with TB0RG1H/L						

Input clock

00	TB0IN0 pin input
01	ϕT1
10	ϕT4
11	ϕT16

Clear up counter 0(UC0)

0	Clear disable
1	Clear by matching with TB0RG1H/L

Capture/interrupt timing

	Capture control	INT4 control
00	Capture disable	Generate INT4 by TB0IN0 rising
01	TB0CP0H/L by TB0IN0 rising TB0CP1H/L by TB0IN1 rising	
10	TB0CP0H/L by TB0IN0 rising TB0CP1H/L by TB0IN0 falling	Generate INT4 by TB0IN0 falling
11	TB0CP0H/L by TA1OUT rising TB0CP1H/L by TA1OUT falling	Generate INT4 by TB0IN0 rising

Software capture

0	Capture value of up counter to TB0CP0H/L
1	Undefined

Inversion trigger control of TB0FF1 when the UC0 match with TB0RG1H/L

0	Disable inversion
1	Enable inversion

Inversion trigger of TB0FF1 when the UC0 value is loaded in to TB0CP1H/L

0	Disable inversion
1	Enable inversion

Figure 3.8.10 Register for TMRB (4)

TMRB1 Mode Register

TB1MOD (1192H)

Read-modify -write instruction is prohibited

	7	6	5	4	3	2	1	0
Bit symbol	TB1CT1	TB1ET1	TB1CP0I	TB1CPM1	TB1CPM0	TB1CLE	TB1CLK1	TB1CLK0
Read/Write	R/W		W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	TB1FF1 Inversion trigger 0: Trigger disable 1: Trigger enable Invert when capture to capture register 1	Invert when match UC1 with TB1RG1H/L	Software capture control 0: Software capture 1:Undefined	Capture timing 00: Disable INT6 is rising edge 01: TB1N0 ↑ TB1IN1 ↑ INT6 is rising edge 10: TB1IN0 ↑ TB1IN0 ↓ INT6 is falling edge 11: TA1OUT ↑ TA1OUT ↓ INT6 is rising edge		Up counter control 0:disable 1:enable	TMRB1 source clock 00: TB1IN0 pin input 01: φT1 10: φT4 11: φT16	

Input clock

00	TB1IN0 pin input
01	φT1
10	φT4
11	φT16

Clear up counter (UC1)

0	Clear disable
1	Clear by matching with TB1RG1H/L

Capture/interrupt timing

	Capture control	INT6 control
00	Capture disable	Generate INT6 by TB1IN0 rising
01	TB1CP0H/L by TB1IN0 rising TB1CP1H/L by TB1IN1 rising	
10	TB1CP0H/L by TB1IN0 rising TB1CP1H/L by TB1IN0 falling	Generate INT6 by TB1IN0 falling
11	TB1CP0H/L by TA1OUT rising TB1CP1H/L by TA1OUT falling	Generate INT6 by TB1IN0 rising

Software capture

0	Capture value of up counter to TB1CP0H/L
1	Undefined

Inversion trigger control of TB1FF1 when the UC1 match with TB1RG1H/L

0	Disable inversion
1	Enable inversion

Inversion trigger of TB1FF1 when the UC1 value is loaded in to TB1CP1H/L

0	Disable inversion
1	Enable inversion

Figure 3.8.11 Register for TMRB (5)

TMRB2 Mode Register

TB2MOD (11A2H)

Read-modify -write instruction is prohibited

	7	6	5	4	3	2	1	0
Bit symbol	TB2CT1	TB2ET1	TB2CP0I	TB2CPM1	TB2CPM0	TB2CLE	TB2CLK1	TB2CLK0
Read/Write	R/W		W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	TB2FF1 Inversion trigger 0: Trigger disable 1: Trigger enable Invert when capture to capture register 2	Invert when match UC2 with TB2RG1H/L	Software capture control 0: Software capture 1:Undefined	Capture timing 00: Disable INT8 is rising edge 01: TB2N0 ↑ TB2IN1 ↑ INT8 is rising edge 10: TB2IN0 ↑ TB2IN0 ↓ INT8 is falling edge 11: TA3OUT ↑ TA3OUT ↓ INT8 is rising edge		Up counter control 0:disable 1:enable	TMRB2 source clock 00: TB2IN0 pin input 01: φT1 10: φT4 11: φT16	

Input clock

00	TB2IN0 pin input
01	φT1
10	φT4
11	φT16

Clear up counter (UC2)

0	Clear disable
1	Clear by matching with TB2RG1H/L

Capture/interrupt timing

	Capture control	INT8 control
00	Capture disable	Generate INT8 by TB2IN0 rising
01	TB2CP0H/L by TB2IN0 rising TB2CP1H/L by TB2IN1 rising	
10	TB2CP0H/L by TB2IN0 rising TB2CP1H/L by TB2IN0 falling	Generate INT8 by TB2IN0 falling
11	TB2CP0H/L by TA3OUT rising TB2CP1H/L by TA3OUT falling	Generate INT8 by TB2IN0 rising

Software capture

0	Capture value of up counter to TB2CP0H/L
1	Undefined

Inversion trigger control of TB2FF1 when the UC2 match with TB2RG1H/L

0	Disable inversion
1	Enable inversion

Inversion trigger of TB2FF1 when the UC2 value is loaded in to TB2CP1H/L

0	Disable inversion
1	Enable inversion

Figure 3.8.12 Register for TMRB (6)

TMRB3 Mode Register

TB3MOD (11B2H)

Read-modify -write instruction is prohibited

	7	6	5	4	3	2	1	0
Bit symbol	TB3CT1	TB3ET1	TB3CP0I	TB3CPM1	TB3CPM0	TB3CLE	TB3CLK1	TB3CLK0
Read/Write	R/W		W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	TB3FF1 Inversion trigger 0: Trigger disable 1: Trigger enable Invert when capture to capture register 3	Invert when match UC3 with TB3RG1H/L	Software capture control 0: Software capture 1:Undefined	Capture timing 00: Disable INTA is rising edge 01: TB3N0 ↑ TB3IN1 ↑ INTA is rising edge 10: TB3IN0 ↑ TB3IN0 ↓ INTA is falling edge 11: TA3OUT ↑ TA3OUT ↓ INTA is rising edge		Up counter control 0:disable 1:enable	TMRB3 source clock 00: TB3IN0 pin input 01: ϕT1 10: ϕT4 11: ϕT16	

Input clock

00	TB3IN0 pin input
01	ϕT1
10	ϕT4
11	ϕT16

Clear up counter (UC3)

0	Clear disable
1	Clear by matching with TB3RG1H/L

Capture/interrupt timing

	Capture control	INTA control
00	Capture disable	Generate INTA by TB3IN0 rising
01	TB3CP0H/L by TB3IN0 rising TB3CP1H/L by TB3IN1 rising	
10	TB3CP0H/L by TB3IN0 rising TB3CP1H/L by TB3IN0 falling	Generate INTA by TB3IN0 falling
11	TB3CP0H/L by TA3OUT rising TB3CP1H/L by TA3OUT falling	Generate INTA by TB3IN0 rising

Software capture

0	Capture value of up counter to TB3CP0H/L
1	Undefined

Inversion trigger control of TB3FF1 when the UC3 match with TB3RG1H/L

0	Disable inversion
1	Enable inversion

Inversion trigger of TB3FF1 when the UC3 value is loaded in to TB3CP1H/L

0	Disable inversion
1	Enable inversion

Figure 3.8.13 Register for TMRB (7)

TMRB4 Mode Register

	7	6	5	4	3	2	1	0
Bit symbol	TB4CT1	TB4ET1	TB4CP0I	TB4CPM1	TB4CPM0	TB4CLE	TB4CLK1	TB4CLK0
Read/Write	R/W		W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	TB4FF1 Inversion trigger 0: Trigger disable 1: Trigger enable Invert when capture to capture register 4	Invert when match UC4 with TB5RG1H/L	Software capture control 0: Software capture 1:Undefined	Capture timing 00: Disable 01: (Reserved) 10: (Reserved) 11: TA5OUT ↑ TA5OUT ↓		Up counter control 0:disable 1:enable	TMRB4 source clock 00: (Reserved) 01: φT1 10: φT4 11: φT16	

TB4MOD (11C2H)

Read-modify-write instruction is prohibited

Input clock

00	(Reserved)
01	φT1
10	φT4
11	φT16

Clear up counter (UC4)

0	Clear disable
1	Clear by matching with TB4RG1H/L

Capture/interrupt timing

	Capture control
00	Capture disable
01	(Reserved)
10	(Reserved)
11	TB4CP0H/L by TA1OUT rising TB4CP1H/L by TA1OUT falling

Software capture

0	Capture value of up counter to TB4CP0H/L
1	Undefined

Inversion trigger control of TB4FF1 when the UC4 match with TB4RG1H/L

0	Disable inversion
1	Enable inversion

Inversion trigger of TB4FF1 when the UC4 value is loaded in to TB4CP1H/L

0	Disable inversion
1	Enable inversion

Figure 3.8.14 Register for TMRB (8)

TMRB5 Mode Register

	7	6	5	4	3	2	1	0
Bit symbol	TB5CT1	TB5ET1	TB5CP0I	TB5CPM1	TB5CPM0	TB5CLE	TB5CLK1	TB5CLK0
Read/Write	R/W		W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	TB5FF1 Inversion trigger 0: Trigger disable 1: Trigger enable Invert when capture to capture register 5	Invert when match UC5 with TB5RG1H/L	Software capture control 0: Software capture 1:Undefined	Capture timing 00: Disable 01: (Reserved) 10: (Reserved) 11: TA5OUT ↑ TA5OUT ↓		Up counter control 0:disable 1:enable	TMRB5 source clock 00: (Reserved) 01: φT1 10: φT4 11: φT16	

TB5MOD (11D2H)

Read-modify -write instruction is prohibited

Input clock

00	(Reserved)
01	φT1
10	φT4
11	φT16

Clear up counter (UC5)

0	Clear disable
1	Clear by matching with TB5RG1H/L

Capture/interrupt timing

	Capture control
00	Capture disable
01	(Reserved)
10	(Reserved)
11	TB5CP0H/L by TA5OUT rising TB5CP1H/L by TA5OUT falling

Software capture

0	Capture value of up counter to TB5CP0H/L
1	Undefined

Inversion trigger control of TB5FF1 when the UC5 match with TB5RG1H/L

0	Disable inversion
1	Enable inversion

Inversion trigger of TB5FF1 when the UC5 value is loaded in to TB5CP1H/L

0	Disable inversion
1	Enable inversion

Figure 3.8.15 Register for TMRB (9)

TMRB0 Flip-flop Control Register

TB0FFCR (1183H)

Read-modify -write instruction is prohibited

	7	6	5	4	3	2	1	0
Bit symbol	TB0FF1C1	TB0FF1C0	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FFC1	TB0FFC0
Read/Write	W*		R/W				W*	
After reset	1	1	0	0	0	0	1	1
Function	TB0FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".		TB0FF0 inversion trigger 0: Trigger disable 1: Trigger enable				TB0FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".	
			Invert when the UC value is loaded in to TB0CP1H/L.	Invert when the UC value is loaded in to TB0CP0H/L.	Invert when the UC matches with TB0RG1H/L.	Invert when the UC matches with TB0RG0H/L.		

Timer flip-flop TB0 (TB0FF0) control

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

Inversion trigger of TB0FF0 when the UC match with TB0RG0H/L

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

Inversion trigger of TB0FF0 when the UC match with TB0RG1H/L

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

Inversion trigger of TB0FF0 When the UC value is loaded in to TB0CP0H/L

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

Inversion trigger of TB0FF0 When the UC value is loaded in to TB0CP1H/L

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

TB0FF1 control

00	Invert value of TB0FF1
01	Set TB0FF1 to "1".
10	Set TB0FF1 to "0".
11	Don't care

Figure 3.8.16 Register for TMRB (10)

TMRB1 Flip-flop Control Register

		7	6	5	4	3	2	1	0
TB1FFCR (1193H)	Bit symbol	TB1FF1C1	TB1FF1C0	TB1C1T1	TB1C0T1	TB1E1T1	TB1E0T1	TB1FFC1	TB1FFC0
	Read/Write	W*		R/W				W*	
	After reset	1	1	0	0	0	0	1	1
Read-modify -write instruction is prohibited	Function	TB1FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".		TB1FF0 inversion trigger 0: Trigger disable 1: Trigger enable				TB1FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".	
				Invert when the UC value is loaded in to TB1CP1H/L.	Invert when the UC value is loaded in to TB1CP0H/L.	Invert when the UC matches with TB1RG1H/L.	Invert when the UC matches with TB1RG0H/L.		

Timer flip-flop TB1 (TB1FF0) control

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

Inversion trigger of TB1FF0 when the UC match with TB1RG0H/L

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

Inversion trigger of TB1FF0 when the UC match with TB1RG1H/L

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

Inversion trigger of TB1FF0 When the UC value is loaded in to TB1CP0H/L

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

Inversion trigger of TB1FF0 When the UC value is loaded in to TB1CP1H/L

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

TB1FF1 control

00	Invert value of TB1FF1
01	Set TB1FF1 to "1".
10	Set TB1FF1 to "0".
11	Don't care

Figure 3.8.17 Register for TMRB (11)

TMRB2 Flip-flop Control Register

		7	6	5	4	3	2	1	0
TB2FFCR (11A3H)	Bit symbol	TB2FF1C1	TB2FF1C0	TB2C1T1	TB2C0T1	TB2E1T1	TB2E0T1	TB2FFC1	TB2FFC0
	Read/Write	W*		R/W				W*	
	After reset	1	1	0	0	0	0	1	1
Read-modify -write instruction is prohibited	Function	TB2FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".		TB2FF0 inversion trigger 0: Trigger disable 1: Trigger enable				TB2FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".	
				Invert when the UC value is loaded in to TB2CP1H/L.	Invert when the UC value is loaded in to TB2CP0H/L.	Invert when the UC matches with TB2RG1H/L.	Invert when the UC matches with TB2RG0H/L.		

Timer flip-flop TB2 (TB2FF0) control

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

Inversion trigger of TB2FF0 when the UC match with TB2RG0H/L

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

Inversion trigger of TB2FF0 when the UC match with TB2RG1H/L

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

Inversion trigger of TB2FF0 When the UC value is loaded in to TB2CP0H/L

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

Inversion trigger of TB2FF0 When the UC value is loaded in to TB2CP1H/L

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

TB2FF1 control

00	Invert value of TB2FF1
01	Set TB2FF1 to "1".
10	Set TB2FF1 to "0".
11	Don't care

Figure 3.8.18 Register for TMRB (12)

TMRB3 Flip-flop Control Register

		7	6	5	4	3	2	1	0
TB3FFCR (11B3H)	Bit symbol	TB3FF1C1	TB3FF1C0	TB3C1T1	TB3C0T1	TB3E1T1	TB3E0T1	TB3FFC1	TB3FFC0
	Read/Write	W*		R/W				W*	
	After reset	1	1	0	0	0	0	1	1
Read-modify -write instruction is prohibited	Function	TB3FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".		TB3FF0 inversion trigger 0: Trigger disable 1: Trigger enable				TB3FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".	
				Invert when the UC value is loaded in to TB3CP1H/L.	Invert when the UC value is loaded in to TB3CP0H/L.	Invert when the UC matches with TB3RG1H/L.	Invert when the UC matches with TB3RG0H/L.		

Timer flip-flop TB3 (TB3FF0) control

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

Inversion trigger of TB3FF0 when the UC match with TB3RG0H/L

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

Inversion trigger of TB3FF0 when the UC match with TB3RG1H/L

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

Inversion trigger of TB2FF0 When the UC value is loaded in to TB2CP0H/L

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

Inversion trigger of TB3FF0 When the UC value is loaded in to TB3CP1H/L

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

TB3FF1 control

00	Invert value of TB3FF1
01	Set TB3FF1 to "1".
10	Set TB3FF1 to "0".
11	Don't care

Figure 3.8.19 Register for TMRB (13)

TMRB4 Flip-flop Control Register

		7	6	5	4	3	2	1	0
TB4FFCR (11C3H)	Bit symbol	TB4FF1C1	TB4FF1C0	TB4C1T1	TB4C0T1	TB4E1T1	TB4E0T1	TB4FFC1	TB4FFC0
	Read/Write	W*		R/W				W*	
	After reset	1	1	0	0	0	0	1	1
Read-modify -write instruction is prohibited	Function	TB4FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".		TB4FF0 inversion trigger 0: Trigger disable 1: Trigger enable				TB4FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".	
				Invert when the UC value is loaded in to TB4CP1H/L.	Invert when the UC value is loaded in to TB4CP0H/L.	Invert when the UC matches with TB4RG1H/L.	Invert when the UC matches with TB4RG0H/L.		

Timer flip-flop TB4 (TB4FF0) control

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

Inversion trigger of TB4FF0 when the UC match with TB4RG0H/L

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

Inversion trigger of TB4FF0 when the UC match with TB4RG1H/L

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

Inversion trigger of TB4FF0 When the UC value is loaded in to TB4CP0H/L

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

Inversion trigger of TB4FF0 When the UC value is loaded in to TB4CP1H/L

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

TB4FF1 control

00	Invert value of TB4FF1
01	Set TB4FF1 to "1".
10	Set TB4FF1 to "0".
11	Don't care

Figure 3.8.20 Register for TMRB (14)

TMRB5 Flip-flop Control Register

		7	6	5	4	3	2	1	0
TB5FFCR (11D3H)	Bit symbol	TB5FF1C1	TB5FF1C0	TB5C1T1	TB5C0T1	TB5E1T1	TB5E0T1	TB5FFC1	TB5FFC0
	Read/Write	W*		R/W				W*	
	After reset	1	1	0	0	0	0	1	1
Read-modify -write instruction is prohibited	Function	TB5FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".		TB5FF0 inversion trigger 0: Trigger disable 1: Trigger enable				TB5FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".	
				Invert when the UC value is loaded in to TB5CP1H/L.	Invert when the UC value is loaded in to TB5CP0H/L.	Invert when the UC matches with TB5RG1H/L.	Invert when the UC matches with TB5RG0H/L.		

Timer flip-flop TB5 (TB5FF0) control

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

Inversion trigger of TB5FF0 when the UC match with TB5RG0H/L

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

Inversion trigger of TB5FF0 when the UC match with TB5RG1H/L

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

Inversion trigger of TB5FF0 When the UC value is loaded in to TB5CP0H/L

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

Inversion trigger of TB5FF0 When the UC value is loaded in to TB5CP1H/L

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

TB5FF1 control

00	Invert value of TB5FF1
01	Set TB5FF1 to "1".
10	Set TB5FF1 to "0".
11	Don't care

Figure 3.8.21 Register for TMRB (15)

Timer Register (TB0RG0H/L, TB0RG1H/L)

		7	6	5	4	3	2	1	0
TB0RG0L (1188H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB0RG0H (1189H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB0RG1L (118AH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB0RG1H (118BH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							

Read-modify-write instruction is prohibited

Capture Register (TB0CP0H/L, TB0CP1H/L)

		7	6	5	4	3	2	1	0
TB0CP0L (118CH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB0CP0H (118DH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB0CP1L (118EH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB0CP1H (118FH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							

Figure 3.8.22 Register for TMRB (16)

Timer Register (TB1RG0H/L, TB1RG1H/L)

		7	6	5	4	3	2	1	0
TB1RG0L (1198H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB1RG0H (1199H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB1RG1L (119AH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB1RG1H (119BH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							

Read-modify-write instruction is prohibited

Capture Register (TB1CP0H/L, TB1CP1H/L)

		7	6	5	4	3	2	1	0
TB1CP0L (119CH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB1CP0H (119DH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB1CP1L (119EH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB1CP1H (119FH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							

Figure 3.8.23 Register for TMRB (17)

Timer Register (TB2RG0H/L, TB2RG1H/L)

		7	6	5	4	3	2	1	0
TB2RG0L (11A8H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB2RG0H (11A9H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB2RG1L (11AAH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB2RG1H (11ABH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							

Read-modify-write instruction is prohibited

Capture Register (TB2CP0H/L, TB2CP1H/L)

		7	6	5	4	3	2	1	0
TB2CP0L (11ACH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB2CP0H (11ADH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB2CP1L (11AEH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB2CP1H (11AFH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							

Figure 3.8.24 Register for TMRB (18)

Timer Register (TB3RG0H/L, TB3RG1H/L)

		7	6	5	4	3	2	1	0
TB3RG0L (11B8H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB3RG0H (11B9H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB3RG1L (11BAH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB3RG1H (11BBH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							

Read-modify-write instruction is prohibited

Capture Register (TB3CP0H/L, TB3CP1H/L)

		7	6	5	4	3	2	1	0
TB3CP0L (11BCH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB3CP0H (11BDH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB3CP1L (11BEH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB3CP1H (11BFH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							

Figure 3.8.25 Register for TMRB (19)

Timer Register (TB4RG0H/L, TB4RG1H/L)

		7	6	5	4	3	2	1	0
TB4RG0L (11C8H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB4RG0H (11C9H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB4RG1L (11CAH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB4RG1H (11CBH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							

Read-modify-write instruction is prohibited

Capture Register (TB4CP0H/L, TB4CP1H/L)

		7	6	5	4	3	2	1	0
TB4CP0L (11CCH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB4CP0H (11CDH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB4CP1L (11CEH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB4CP1H (11CFH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							

Figure 3.8.26 Register for TMRB (20)

Timer Register (TB5RG0H/L, TB5RG1H/L)

		7	6	5	4	3	2	1	0
TB5RG0L (11D8H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB5RG0H (11D9H)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB5RG1L (11DAH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							
TB5RG1H (11DBH)	bit Symbol	–							
	Read/Write	W							
	After reset	undefined							

Read-modify-write instruction is prohibited

Capture Register (TB5CP0H/L, TB5CP1H/L)

		7	6	5	4	3	2	1	0
TB5CP0L (11DCH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB5CP0H (11DDH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB5CP1L (11DEH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							
TB5CP1H (11DFH)	bit Symbol	–							
	Read/Write	R							
	After reset	undefined							

Figure 3.8.27 Register for TMRB (21)

3.8.4 Operation in Each Mode

(1) 16-bit interval timer mode

Generating interrupts at fixed intervals in this example, the interval time is set the timer register TB0RG1H/L to generate the interrupt INTTB01.

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	X	–	0	X	0	Stop TMRB0.
INTETB0	←	X	1	0	0	X	0	0	0	Enable INTTB01 and set interrupt level 4. Disable INTTB00.
TB0FFCR	←	1	1	0	0	0	0	1	1	Disable the trigger.
TB0MOD	←	0	0	1	0	0	1	*	*	Set input clock to prescaler clock, and set capture function to disable. (** = 01, 10, 11)
TB0RG1H/L	←	*	*	*	*	*	*	*	*	Set the interval time (16 bits).
		*	*	*	*	*	*	*	*	
TB0RUN	←	0	0	X	X	–	1	X	1	Start TMRB0.

X : Don't care, – : No change

(2) 16-bit event counter mode

In 16-bit timer mode as described in above, the timer can be used as an event counter by selecting the external clock (TB0IN0 pin input) as the input clock.

Up counter counting up by rising edge of TB0IN0 pin input. And execution software capture and reading capture value enable reading count value.

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	X	–	0	X	0	Stop TMRB0.
PKFC	←	–	–	–	–	–	–	–	1	Set PK0 to TB0IN0 input mode.
PKFC2	←	–	–	–	–	–	–	–	0	
INTETB0	←	X	1	0	0	X	0	0	0	Set INTTB01 to enable (Interrupt level4). Set INTTB00 to disable.
TB0FFCR	←	1	1	0	0	0	0	1	1	Set trigger to disable.
TB0MOD	←	0	0	1	0	0	1	0	0	Set input clock to TB0IN0 pin input.
TB0RG1H/L	←	*	*	*	*	*	*	*	*	Set number of count. (16 bits)
		*	*	*	*	*	*	*	*	
TB0RUN	←	0	0	X	X	–	1	X	1	Start TMRB0.

X: Don't care, –: No change

Note: When used as an event counter, set the prescaler to "RUN" (TB0RUN<TB0PRUN> = "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 TB0FF0 that is to be enabled by the match of the up counter UC0 with timer register TB0RG0H/L or TB0RG1H/L and to be output to TB0OUT0. In this mode, the following conditions must be satisfied.

(Set value of TB0RG0H/L) < (Set value of TB0RG1H/L)

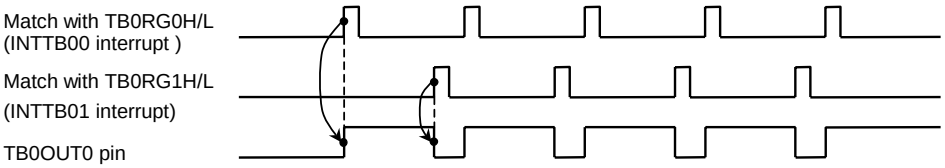


Figure 3.8.28 Programmable Pulse Generation (PPG) Output Waveform

When the TB0RG0H/L double buffer is enabled in this mode, the value of register buffer 0 will be shifted into TB0RG0H/L at match with TB0RG1H/L. This feature makes easy the handling of low-duty waves.

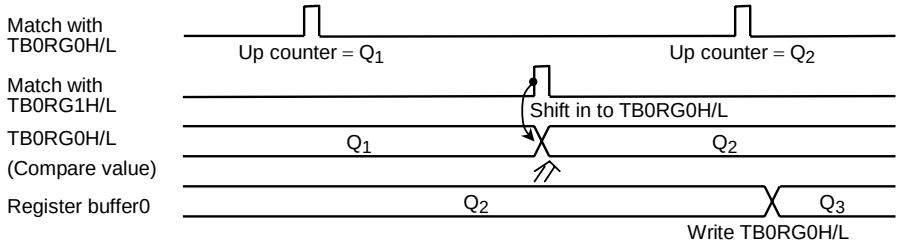


Figure 3.8.29 Operation of Register Buffer

The following block diagram illustrates this mode.

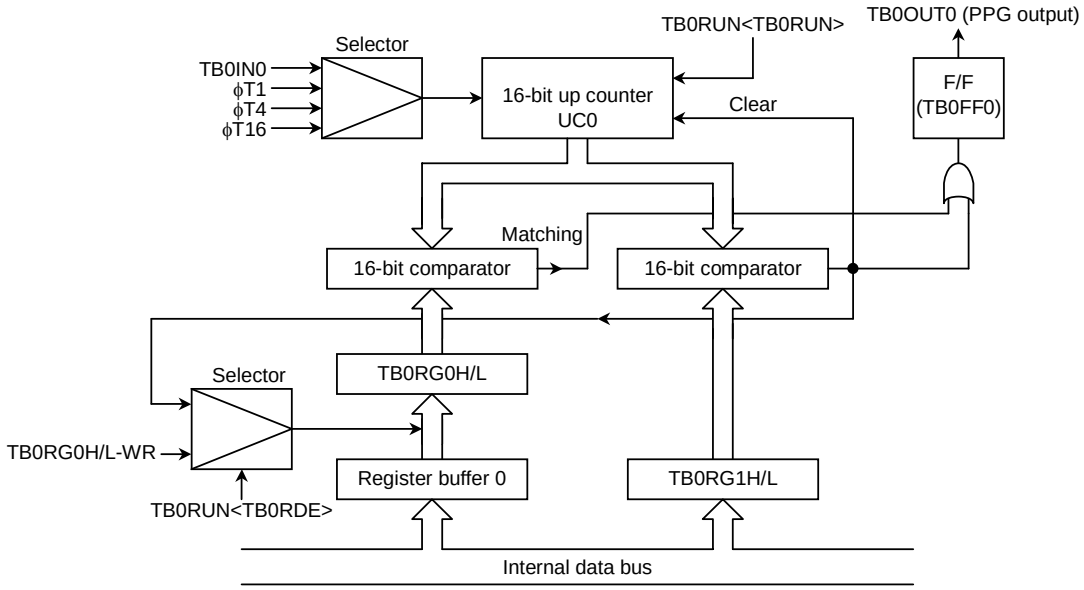


Figure 3.8.30 Block Diagram of 16-Bit PPG Mode

The following example shows how to set 16-bit PPG output mode:

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	X	–	0	X	0	Disable the TB0RG0H/L double buffer and stop TMRB0.
TB0RG0H/L	←	*	*	*	*	*	*	*	*	Set the duty ratio (16 bits).
		*	*	*	*	*	*	*	*	
TB0RG1H/L	←	*	*	*	*	*	*	*	*	Set the frequency (16 bits).
		*	*	*	*	*	*	*	*	
TB0RUN	←	1	0	X	X	–	0	X	0	Enable the TB0RG0H/L double buffer. (The duty and frequency are changed on an INTTB01 interrupt.)
TB0FFCR	←	X	X	0	0	1	1	1	0	Set the mode to invert TB0FF0 at the match with TB0RG0H/L, TB0RG1H/L. Clear TB0FF0 to 0.
TB0MOD	←	0	0	1	0	0	1	*	*	Set input clock to prescaler output clock and disable the capture function.
							(** = 01, 10, 11)			
PJFC	←	–	–	–	–	–	–	–	1	
PJCR	←	–	–	–	–	–	–	–	1	
TB0RUN	←	1	0	X	X	–	1	X	1	Start TMRB0.

X : Don't care, – : No change

(4) Capture function examples

Used capture function, they can be applicable in many ways, for example:

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

1. One-shot pulse output from external trigger pulse

Set the up counter UC0 in free-running mode with the internal input clock, input the external trigger pulse from TB0IN0 pin, and load the value of up counter into capture register TB0CP0H/L at the rise edge of external trigger pulse.

When the interrupt INT4 is generated at the rise edge of external trigger pulse, set the TB0CP0H/L value (c) plus a delay time (d) to TB0RG0H/L (= c + d), and set the above set value (c + d) plus a one-shot width (p) to TB0RG1H/L (= c + d + p). And, set "11" to timer flip-flop control register TB0FFCR<TB0E1T1, TB0E0T1>. Set to trigger enable for be inverted timer flip-flop TB0FF0 by UC0 matching with TB0RG0H/L and with TB0RG1H/L. When interrupt INTTB01 occurs, this inversion will be disabled after one-shot pulse is output.

The (c), (d), and (p) correspond to c, d, and p in Figure 3.8.31.

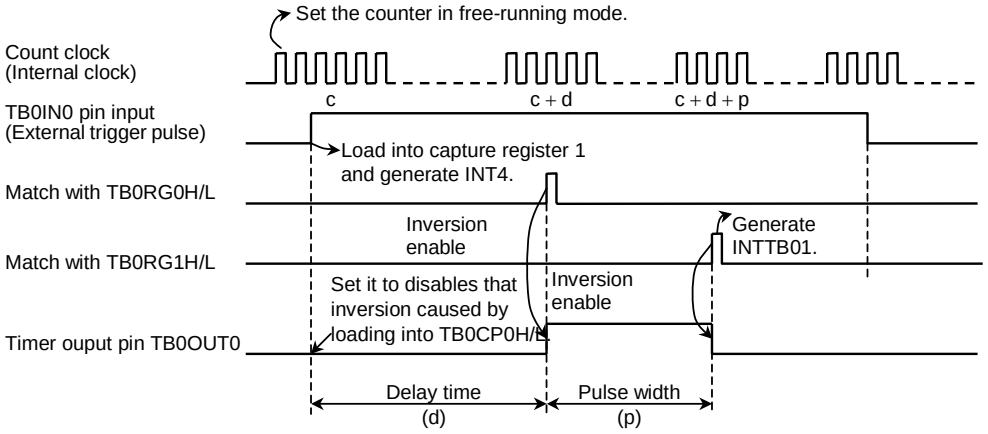
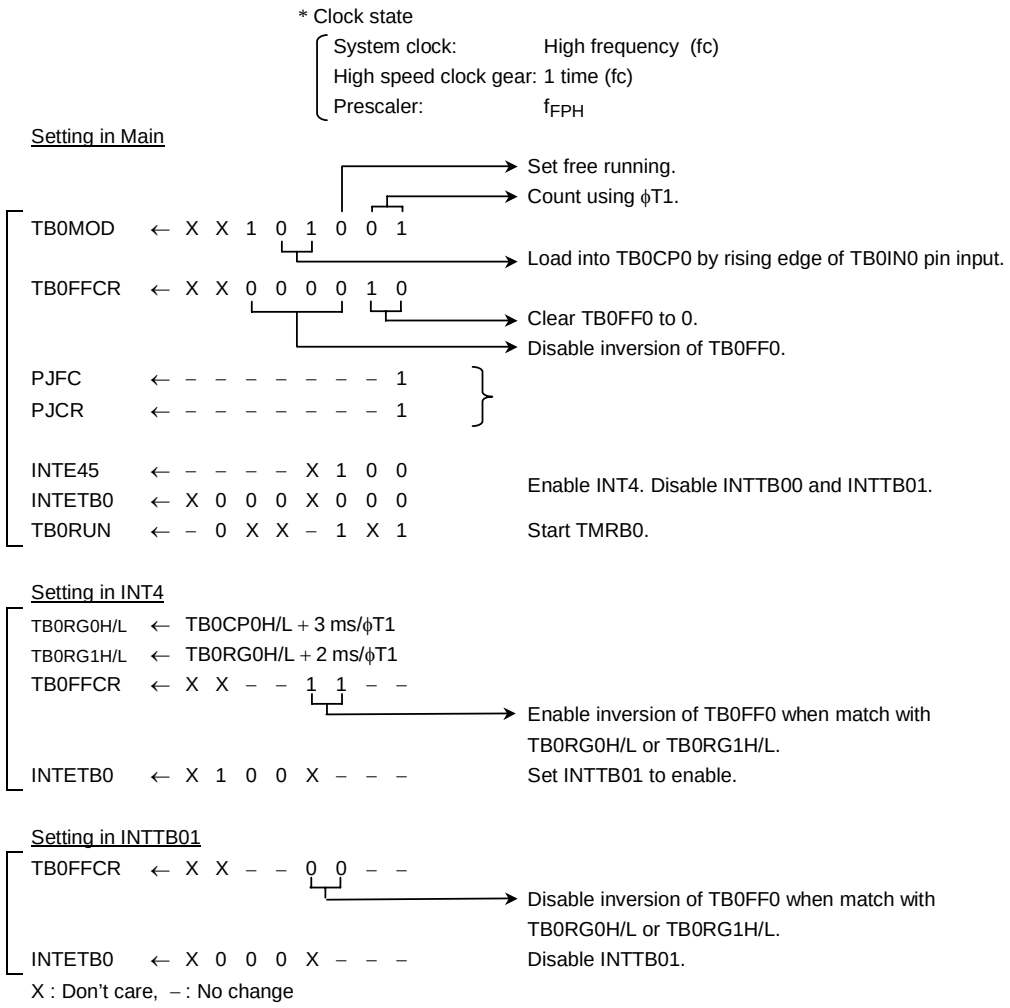


Figure 3.8.31 One-shot Pulse Output (with delay)

Example: To output a 2 [ms] one-shot pulse with a 3 [ms] delay to the external trigger pulse via the TB0IN0 pin.



When delay time is unnecessary, invert timer flip-flop TB0FF0 when up counter value is loaded into capture register (TB0CP0H/L), and set the TB0CP0H/L value (c) plus the one-shot pulse width (p) to TB0RG1H/L when the interrupt INT4 occurs. The TB0FF0 inversion should be enable when the up counter (UC0) value matches TB0RG1H/L, and disabled when generating the interrupt INTTB01.

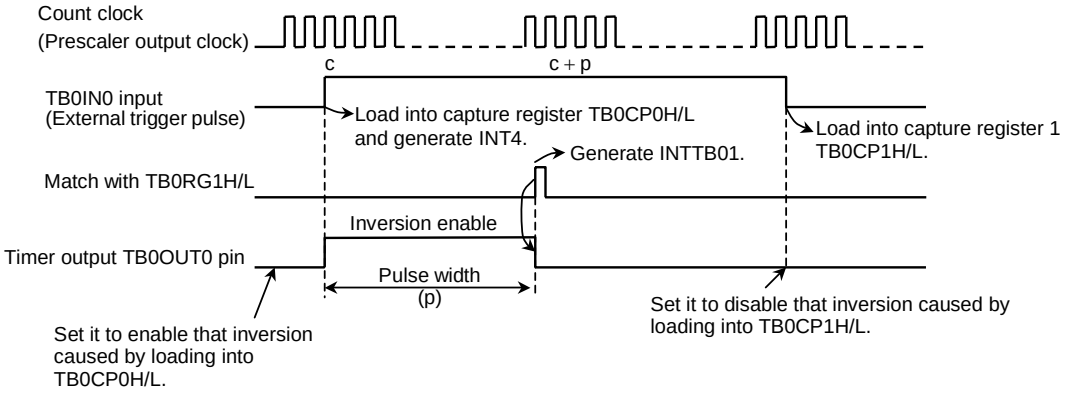


Figure 3.8.32 One-shot Pulse Output of External Trigger Pulse (without delay)

2. Frequency measurement

The frequency of the external clock can be measured in this mode. Frequency is measured by the 8-bit timers TMRA01 and the 16-bit timer/event counter.

TMRA01 is used to setting of measurement time by inversion TA1FF.

Counter clock in TMRB0 select TB0IN0 pin input, and count by external clock input. Set to TB0MOD<TB0CPM1:0> = "11". The value of the up counter (UC0) is loaded into the capture register TB0CP0H/L at the rise edge of the timer flip-flop TA1FF of 8-bit timers (TMRA01), and into TB0CP1H/L at its fall edge.

The frequency is calculated by difference between the loaded values in TB0CP0H/L and TB0CP1H/L when the interrupt (INTTA0 or INTTA1) is generates by either 8-bit timer.

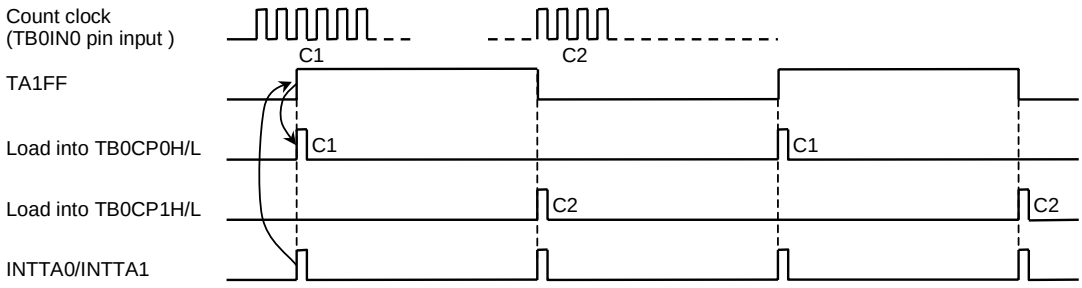


Figure 3.8.33 Frequency Measurement

For example, if the value for the level 1 width of TA1FF of the 8-bit timer is set to 0.5 s and the difference between the values in TB0CP0H/L and TB0CP1H/L is 100, the frequency is 100 ÷ 0.5 s = 200 Hz.

3. Pulse width measurement

This mode allows measuring the high level width of an external pulse. While keeping the 16-bit timer/event counter counting (Free running) with the prescaler output clock input, external pulse is input through the TB0IN0 pin. Then the capture function is used to load the UC0 values into TB0CP0H/L and TB0CP1H/L at the rising edge and falling edge of the external trigger pulse respectively. The interrupt INT4 occurs at the falling edge of TB0IN0.

The pulse width is obtained from the difference between the values of TB0CP0H/L and TB0CP1H/L and the internal clock cycle.

For example, if the prescaler output clock is 0.8 μs and the difference between TB0CP0H/L and TB0CP1H/L is 100, the pulse width will be 100 × 0.8 μs = 80 μs.

Additionally, the pulse width that is over the UC0 maximum count time specified by the clock source can be measured by changing software.

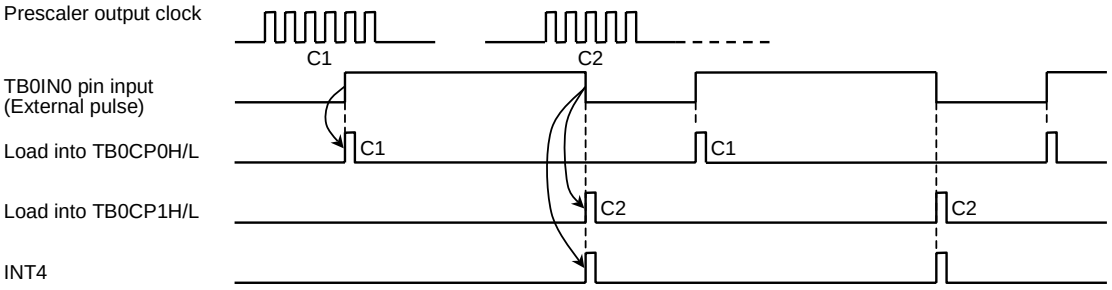


Figure 3.8.34 Pulse Width Measurement

Note: Pulse Width measure by setting “10” to TB0MOD<TB0CPM1:0>. The external interrupt INT4 is generated in timing of falling edge of TB0IN0 input. In other modes, it is generated in timing of rising edge of TB0IN0 input.

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

4. Measurement of difference time

This mode is used to measure the difference in time between the rising edges of external pulses input through TB0IN0 and TB0IN1.

Keep the 16-bit timer/event counter (TMRB0) counting (Free running) with the prescaler output clock, and load the UC0 value into TB0CP0H/L at the rising edge of the input pulse to TB0IN0. Then the interrupt INT4 is generated.

Similarly, the UC0 value is loaded into TB0CP1H/L at the rising edge of the input pulse to TB0IN1, generating the interrupt INT5.

The time difference between these pulses can be obtained by multiplying the value subtracted TB0CP0H/L from TB0CP1H/L and the internal clock cycle together at which loading the UC0 value into TB0CP0H/L and TB0CP1H/L has been done.

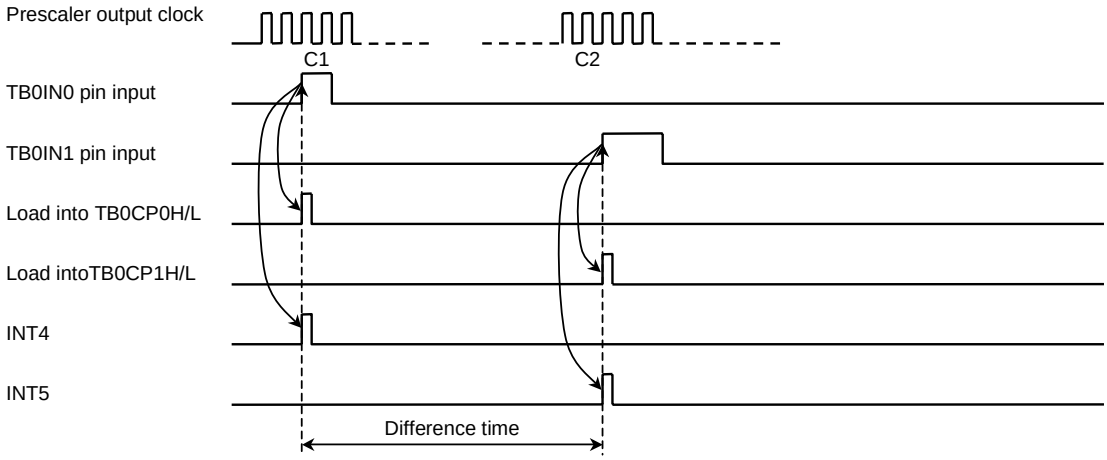


Figure 3.8.35 Measurement of Difference Time

3.9 Pattern Generator/Stepping Motor Control(PG)

The TMP92CM27 contains two 4-bit hardware pattern generator/stepping motor control channels, PG0 and PG1, (hereinafter called PG) which actuate in synchronization with the (8-bit/16-bit) timers. PG (PG0 and PG1) shares the 8-bit input/output port with PL.

The output on channel 0 (PG0) is updated in synchronization with the 8-bit timer 0, 1 (TMRA01) or 16-bit timer 0 (TMRB0). The output on channel 1 (PG1) is updated in synchronization with the 8-bit timer 2, 3 (TMRA23) or 16-bit timer 1 (TMRB1). Figure 3.9.1 show block diagram.

The PG ports are controlled by the control register (PG01CR) and can select either stepping motor control mode or pattern generation mode. Each bit of PL can be used for a PG port.

PG0 and PG1 can be used independently.

Since the two PG channels operate in the same manner, except for the following points, only the operation of PG0 will be explained below.

Differences between PG0 and PG1

	PG0	PG1
Trigger signal	8-bit timer 0,1 (TMRA01) or 16-bit timer 0 (TMRB0)	8-bit timer 2,3 (TMRA23) or 16-bit timer 1 (TMRB1)

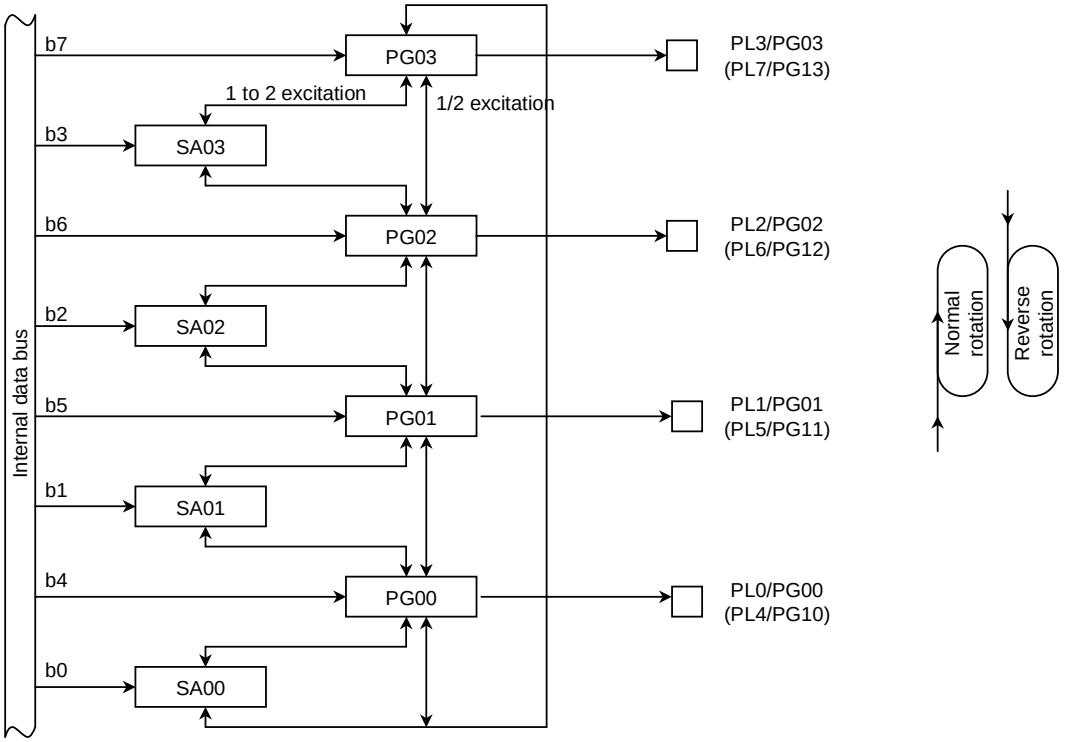


Figure 3.9.1 PG Block Diagram

PG01CR (1462H)

	7	6	5	4	3	2	1	0
Bit symbol	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
Read/Write	R/W				R/W			
After reset	0	0	0	0	0	0	0	0
Function	PG1 write mode 0: 8-bit write 1: 4-bit write	PG1 rotation direction 0: Normal rotation 1: Reverse rotation	PG1 mode (Excitation) 0: 1 step excitation or 2 step excitation 1: 1 to 2 step excitation	PG1 trigger input enable 0: Disable 1: Enable	PG0 write mode 0: 8-bit write 1: 4-bit write	PG0 rotation direction 0: Normal rotation 1: Reverse rotation	PG0 mode (Excitation) 0: 1 step excitation or 2 step excitation 1: 1 to 2 step excitation	PG0 trigger input enable 0: Disable 1: Enable

PG0 trigger input enable

0	Trigger input disable for PG0
1	Trigger input enable for PG0

Set the operation mode for PG0

0	1or 2 step excitation (Full step)
1	1 to 2 step excitation (Half step) /PG mode

PG0 (Stepping motor control) rotation direction control

0	Normal rotation/PG mode
1	Reverse rotation

Selecting PG0 write mode

0	8-bit write
1	4-bit write/PG mode (Only the shift alternate register can be written.)

Figure 3.9.2 Pattern Generation Control Register (PG01CR) (1/2)

PG01CR (1462H)

	7	6	5	4	3	2	1	0
Bit symbol	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
Read/Write	R/W				R/W			
After reset	0	0	0	0	0	0	0	0
Function	PG1 write mode 0: 8-bit write 1: 4-bit write	PG1 rotation direction 0: Normal rotation 1: Reverse rotation	PG1 mode (Excitation) 0: 1 step excitation or 2 step excitation 1: 1 to 2 step excitation	PG1 trigger input enable 0: Disable 1: Enable	PG0 write mode 0: 8-bit write 1: 4-bit write	PG0 rotation direction 0: Normal rotation 1: Reverse rotation	PG0 mode (Excitation) 0: 1 step excitation or 2 step excitation 1: 1 to 2 step excitation	PG0 trigger input enable 0: Disable 1: Enable

PG1 trigger input enable

0	Trigger input disable for PG1
1	Trigger input enable for PG1

Set the operation mode for PG1

0	1 or 2 step excitation (Full step)
1	1 to 2 step excitation (Half step) /PG mode

PG1 (Stepping motor control)
Rotation direction control

0	Normal rotation/PG mode
1	Reverse rotation

Selecting PG1 write mode

0	8-bit write
1	4-bit write / PG mode (Onlt the shift alternate register can be written.)

Figure 3.9.3 Pattern Generation Control Register (PG01CR) (2/2)

		7	6	5	4	3	2	1	0
PG0REG (1460H)	Bit symbol	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
	Read/Write	W				R/W			
Prohibit read-modify-write	After reset	0	0	0	0	Undefined			
	Function	Pattern generation 0 (PG0) output latch register (PG0 can be read by reading the port (PL) that is assigned to PG)				Shift alternate register 0 for the PG mode (4-bit write) register			

Figure 3.9.4 Pattern generation 0 register (PG0REG)

		7	6	5	4	3	2	1	0
PG1REG (1461H)	Bit symbol	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
	Read/Write	W				R/W			
	After reset	0	0	0	0	Undefined			
	Function	Pattern generation 1 (PG1) output latch register (PG1 can be read by reading the port (PL) that is assigned to PG)				Shift alternate register 1 for the PG mode (4-bit write) register			

Figure 3.9.5 Pattern generation 1 register (PG1REG)

PG01CR2 (1464H)

	7	6	5	4	3	2	1	0
Bit symbol							PG1T	PG0T
Read/Write							R/W	
After reset							0	0
Function							PG1shift trigger 0: 8-bit timer trigger (TMRA23) 1: 16-bit timer trigger (TMRB1)	PG0 shift trigger 0: 8-bit timer trigger (TMRA01) 1: 16-bit timer trigger (TMRB0)

Selecting PG0 shift trigger

0	8-bit timer trigger (TMRA01)
1	16-bit timer trigger (TMRB0)

Selecting PG1 shift trigger

0	8-bit timer trigger (TMRA23)
1	16-bit timer trigger (TMRB1)

Figure 3.9.6 Pattern Generation Control Register 2 (PG01CR2)

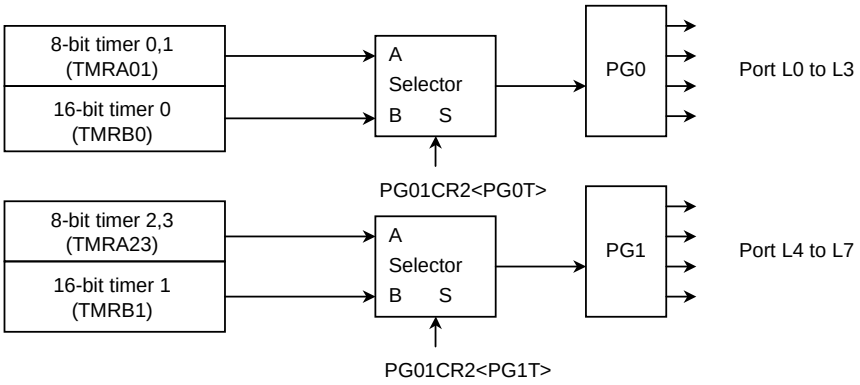


Figure 3.9.7 Connection between Timer and Pattern Generator

(1) Pattern generation mode

When PG01CR<PAT0> = "1", PG functions as a pattern generator. In this mode data is written from the CPU to the shift alternate register only. The pattern data is then written from the shift alternate register to the pattern generator register synchronized to the shift trigger interrupt from the timer.

In this mode, PG01CR<PG0M> should be set to "1", PG01CR<CCW0> to "0", and PG01CR<PG0TE> to "1".

The output from the pattern generator goes to port L; since port or functions can be switched by the bit settings in the port function control register (PLFC) and port function control register 2 (PLFC2), any port pin can be assigned to pattern generator output.

Figure 3.3.9 shows the block diagram for this mode.

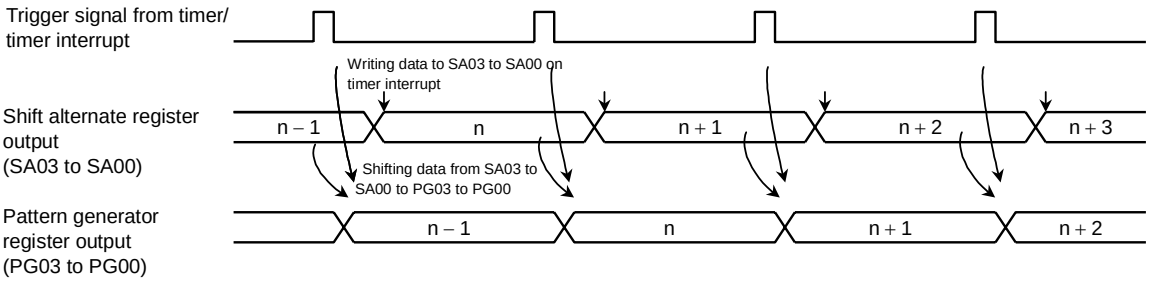


Figure 3.9.8 Example of Pattern Generation Mode

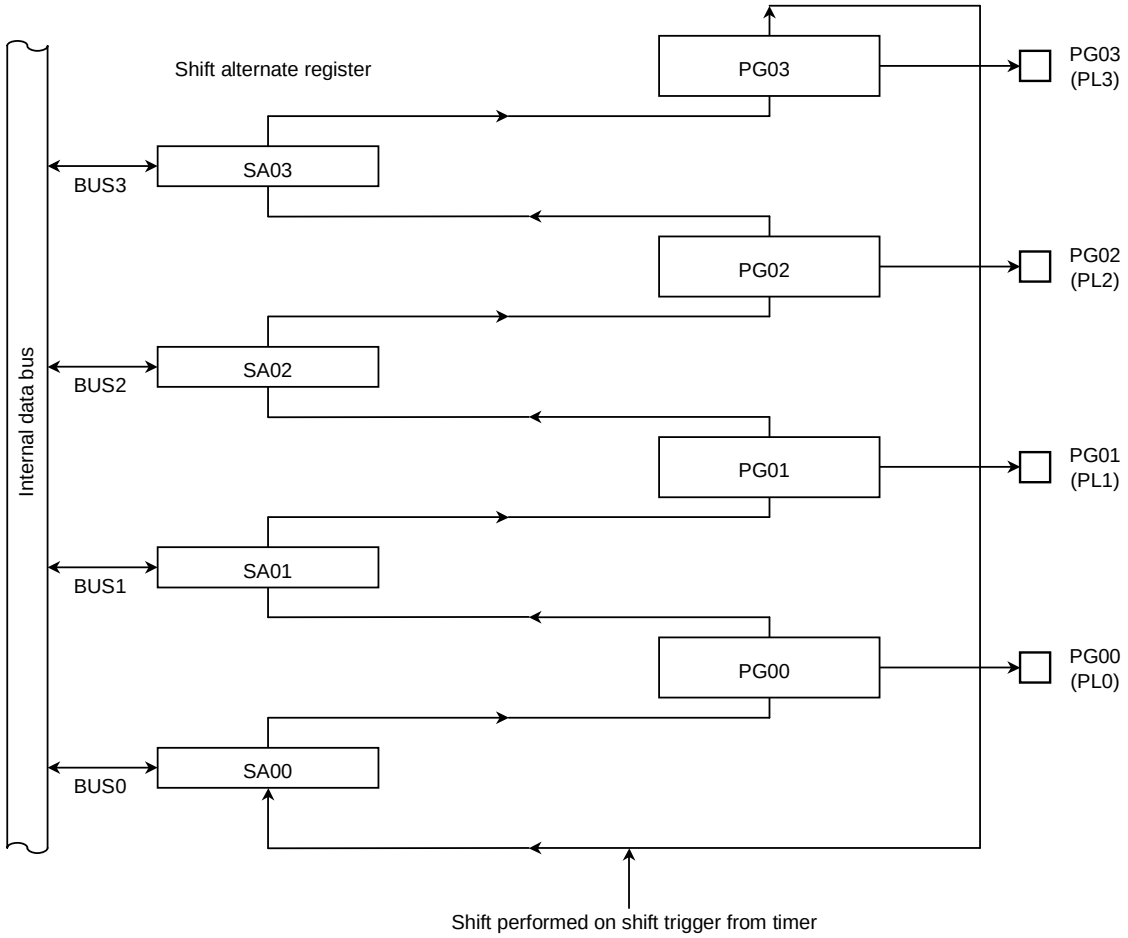


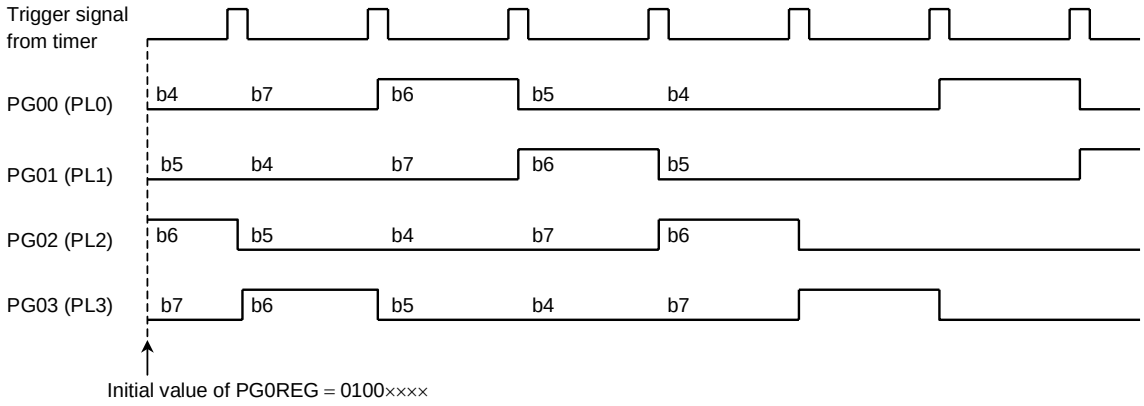
Figure 3.9.9 Pattern Generation Mode Block Diagram (PG0)

In pattern generation mode, only writing to the output latch can be disabled by hardware. All other functions behave in the same way as 1 to 2 step excitation in stepping motor control port mode. Hence, data shifted on the trigger signal from a timer must be written before the next trigger signal is output.

(2) Stepping motor control mode

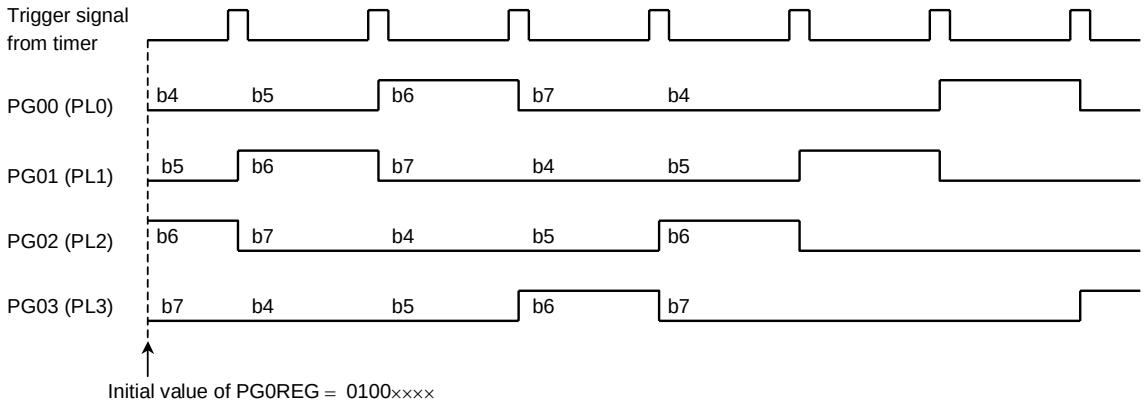
a. 4-phase 1-step/2-step excitation

Figure 3.9.10 and Figure 3.9.11 show the output waveforms for 4-phase 1 excitation and 4-phase 2 excitation respectively when channel 0 (PG0) is selected.



Note: bn indicates the initial value of PG0REG ← b7 b6 b5 b4 × × × ×

(1) Normal rotation



(2) Reverse rotation

Figure 3.9.10 Output Waveforms for 4-Phase 1-Step Excitation

(Normal rotation and Reverse rotation)

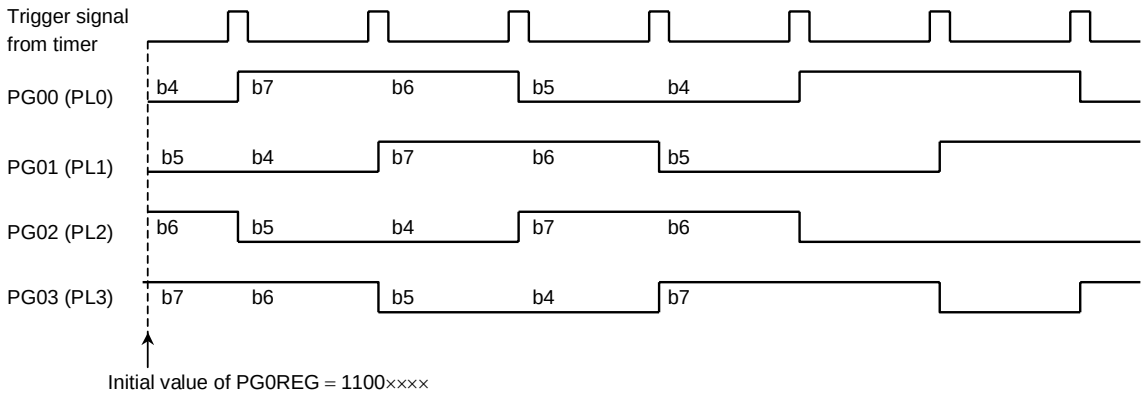


Figure 3.9.11 Output Waveforms for 4-Phase 2-Step Excitation (Normal rotation)

The output from PG0 (PL) is latched on the rising edge of the trigger signal from the timer.

The direction of shift is specified by the setting of PG01CR<CCW0>: Normal rotation (PG00→PG01→PG02→PG03) is selected when <CCW0> is set to "0"; reverse rotation (PG00←PG01←PG02←PG03) is selected when <CCW0> is set to "1". 4-phase 1-step excitation will be selected when only one bit is set to "1" during the initialization of PG, while 4-phase 2-step excitation will be selected when two consecutive bits are set to "1".

The value in the shift alternate registers are ignored when 4-phase 1-step/2-step excitation mode is selected.

Figure 3.9.12 shows the block diagram.

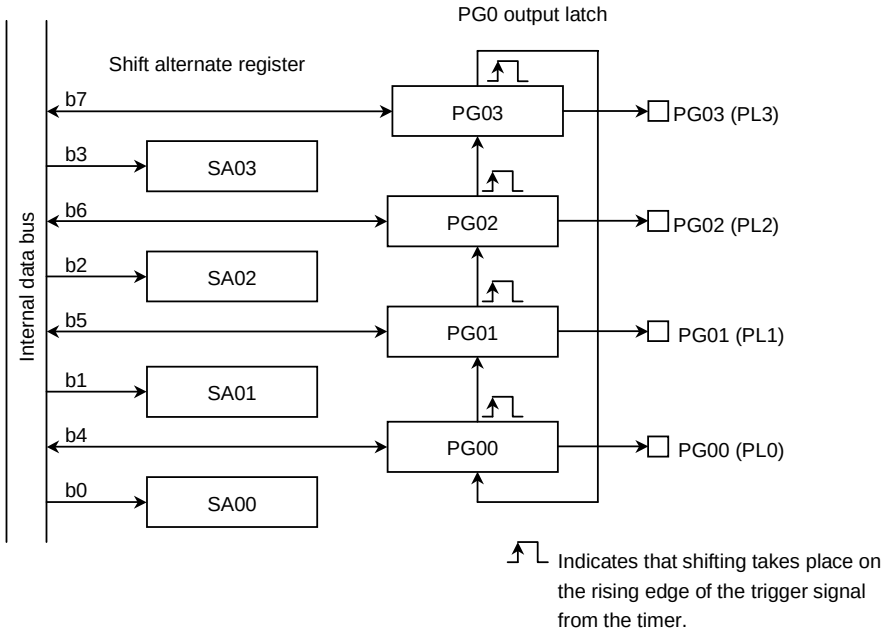
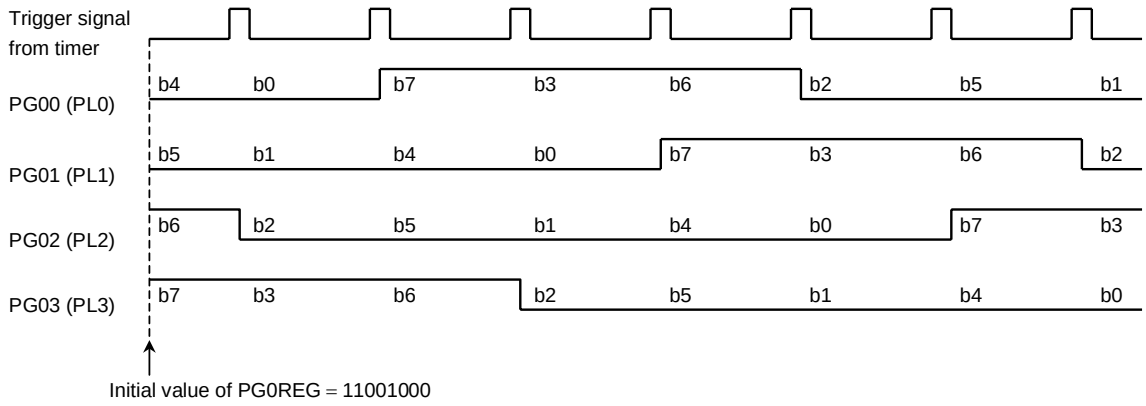


Figure 3.9.12 Block Diagram 4-Phase 1-step Excitation/2-step Excitation (Normal rotation)

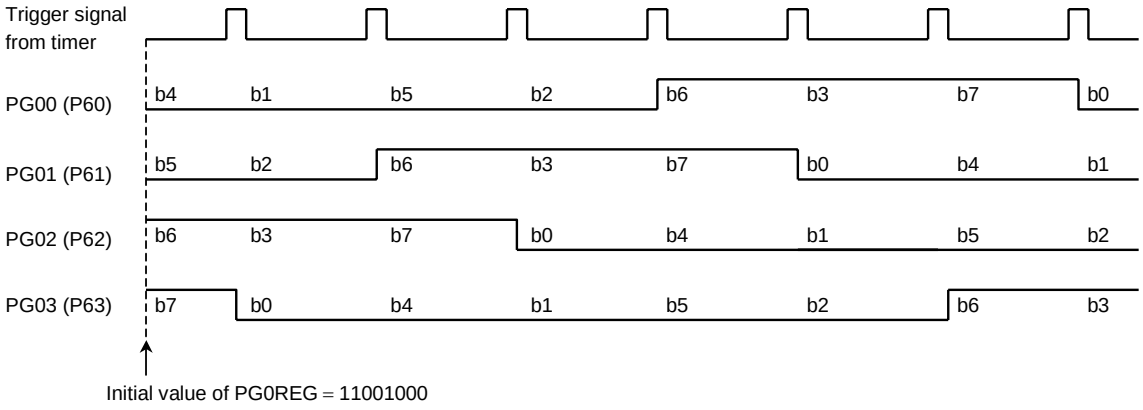
b. 4-phase 1 to 2 step excitation

FigureFigure 3.9.12 shows the output waveforms for 4-phase 1 to 2 step excitation.



Note: bn denotes the initial value PG0REG ← b7 b6 b5 b4 b3 b2 b1 b0

(1) Normal rotation



(2) Reverse rotation

Figure 3.9.12 Output Waveforms for 4-phase 1 to 2 step Excitation

(Normal rotation and reverse rotation)

The initialization sequence for 4-phase 1-2 step excitation is as follows.

By rearranging the initial value b7 b6 b5 b4 b3 b2 b1 b0 to b7 b3 b6 b2 b5 b1 b4 b0, three consecutive bits are set to 1 and the other bits are set to 0 (Positive logic).

For example, if b7, b3, and b6 are set to 1, the initial value becomes 11001000, producing the output waveforms shown in Figure 3.9.12.

To generate a negative logic output waveform, the 1's and 0's in the initial value must be inverted. For example, to change the output waveform shown in Figure 3.9.12 negative logic, change the initial value to 00110111.

The operation will be explained below for channel 0.

The output from PG0 (PL) and from the shift alternate register (SA0) for pattern generation is latched on the rising edge of the trigger signal from the timer. The shift direction is set by PG01CR<CCW0>.

Figure 3.9.13 shows the block diagram.

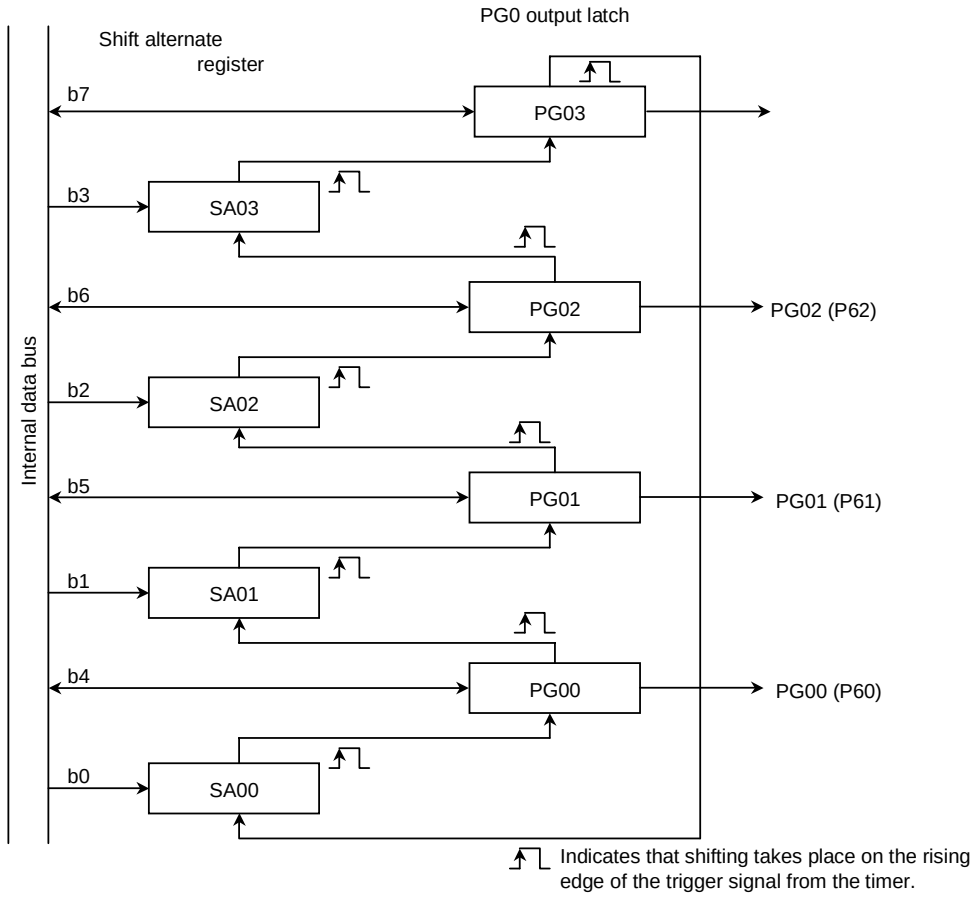


Figure 3.9.13 Block Diagram for 4-phase 1 to 2-step Excitation (Normal rotation)

Setting example: To drive channel 0 (PG0) using 4-phase 1 to 2-step excitation (Normal rotation) when timer 0 is selected, set each register as follows.

	7	6	5	4	3	2	1	0	
TA01RUN ←	0	X	X	X	–	0	0	0	Stop timer 0, and clear it to zero.
TA01MOD ←	0	0	0	0	–	–	0	1	Set 8-bit timer mode and select φT1 as the input clock.
TA1FFCR ←	X	X	X	X	1	0	1	0	Clear TA1FF to zero and enable the inversion trigger using timer 0.
TA0REG ←	*	*	*	*	*	*	*	*	Set the cycle in the timer register.
PLCR ←	–	–	–	–	1	1	1	1	Set bits PL0 to PL3 to PG0 output.
PLFC ←	–	–	–	–	1	1	1	1	
PLFC2 ←	–	–	–	–	0	0	0	0	
PG01CR ←	–	–	–	–	0	0	1	1	Select PG0 4-phase 1 to 2-step excitation mode and normal rotation.
PG0REG ←	1	1	0	0	1	0	0	0	Set an initial value.
TA01RUN ←	0	X	X	X	–	1	–	1	Start timer 0.

X: Don't care、–: No change

(3) Trigger signal from timer

The trigger signal from the timer used by PG is not the same as the trigger signal for the timer flip-flop (TA1FF, TA3FF, TB0FF0, TB0FF1, TB1FF0 and TB1FF1); they differ as shown in Table 3.9.1 depending on the operation mode of the timer.

Table 3.9.1 Trigger Signal Selection

	TA1FF Inversion	PG Shift
8-bit timer mode	Selected by TA1FFCR<TA1FFIS> when the up counter value matches TA0REG or TA1REG value.	Selected by TA1FFCR<TA1FFIS> when the up counter value matches TA0REG or TA1REG value.
16-bit timer mode	When the up counter value matches both TA0REG and TA1REG values (the value of up counter = TA1REG × 2^8 + TA0REG).	When the up counter value matches both TA0REG and TA1REG values (the value of up counter = TA1REG × 2^8 + TA0REG).
PPG output mode	When the up counter value matches both TA0REG and TA1REG.	When the up counter value matches TA1REG value (PPG cycle).
PWM output mode	When the up counter value matches TA0REG value and PWM cycle.	Trigger signal for PG is not generated.

Note: To shift PG, TA1FFCR<TA1FFIE> must be set to 1 to enable TA1FF inversion.

PG can be synchronized with the 16-bit timer timer 0/16-bit timer 1. In this case, the PG shift trigger signal from the 16-bit timer is output only when the up counter UC0/UC1 value matches TB0RG1H/L/TB1RG1H/L.

(4) Application of PG and timer output

As explained in the previous section trigger signal from timer, the timings for shifting PG and inverting TFF differ depending on the timer mode. An application which operates PG while operating an 8-bit timer in PPG mode is explained below.

To drive a stepping motor, a synchronizing signal is required for the excitation timing, in addition to the value of each phase (PG output). In this application, port L is used as a stepping motor control port to output a synchronizing signal to the TA1OUT pin (shared with PF1).

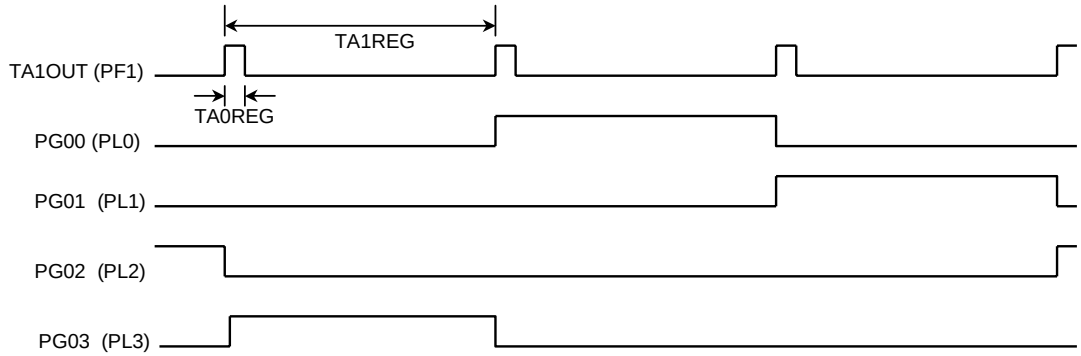


Figure 3.9.14 Output Waveforms for 4-phase 1-step Excitation

Setting example:

	7	6	5	4	3	2	1	0	
TA01RUN ←	0	X	X	X	–	0	0	0	Stop timer 0, 1 and clear it to zero.
TA01MOD ←	1	0	X	X	X	X	0	1	Set timer 0, 1 to PPG output mode and select ϕT1 as the input clock.
TA1FFCR ←	X	X	X	X	0	1	1	X	Enable TA1FF inversion and set TA1FF to “1”.
TA0REG ←	*	*	*	*	*	*	*	*	Set the duty of TA1OUT to TA0REG.
TA1REG ←	*	*	*	*	*	*	*	*	Set the cycle of TA1OUT to TA1REG.
PFCR ←	X	–	–	–	–	–	1	–	Assign PF1 as TA1OUT.
PFFC ←	X	–	–	–	–	–	1	–	
PLCR ←	–	–	–	–	1	1	1	1	Assign PL0 to PL3 as PG0.
PLFC ←	–	–	–	–	1	1	1	1	
PLFC2 ←	–	–	–	–	0	0	0	0	
PG01CR ←	–	–	–	–	0	0	0	1	Set PG0 to 4-phase 1-step excitation mode.
PG0REG ←	*	*	*	*	*	*	*	*	Set an initial value.
TA01RUN ←	0	X	X	X	–	1	1	1	Start timer 0, 1.

X: Don’t care、–: No change

3.10 Serial Channels (SIO)

TMP92CM27 includes 4 serial I/O channels. Each channel is called SIO0, SIO1, SIO2 and SIO3. For all 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 wakeup function for making the master controller start slave controllers via a serial link (Multi-controller system).

Figure 3.10.2 and Figure 3.10.3 are block diagrams for each channel. Each channel is structured in prescaler, serial clock generation circuit, receiving buffer and control circuit, and transfer buffer and control circuit.

Serial channels 0 to 3 can be used independently.

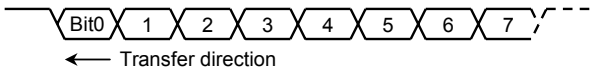
All channels operate in the same function except for the following points; hence only the operation of channel 0 is explained below.

Table 3.10.1 Differences between each Channels

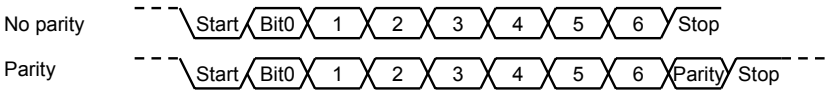
	Channel 0	Channel 1	Channel 2	Channel 3
Pin name	TXD0 (PA1) RXD0 (PA0) $\overline{CTS0}$ /SCLK0 (PA2)	TXD1 (PA4) RXD1 (PA3) $\overline{CTS1}$ /SCLK1 (PA5)	TXD2 (PD4) RXD2 (PD3) $\overline{CTS2}$ /SCLK2 (PD5)	TXD3 (PL1) RXD3 (PL0) $\overline{CTS3}$ /SCLK3 (PL2)
IrDA mode	Yes	Non	Non	Non

This chapter contains the following sections:

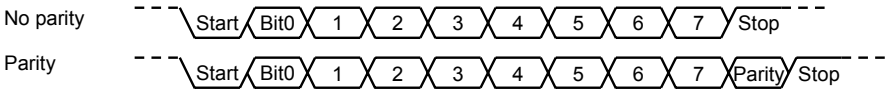
- Mode 0 (I/O interface mode)



- Mode 1 (7-bit UART mode)



- Mode 2 (8-bit UART mode)



- Mode 3 (9-bit UART mode)

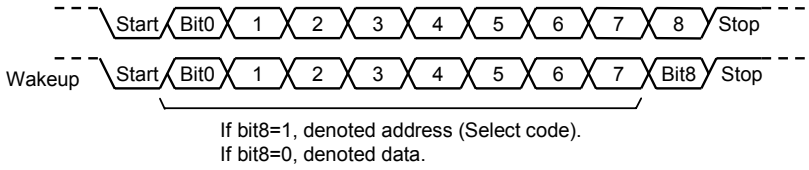


Figure 3.10.1 Data Format

3.10.1 Block Diagram

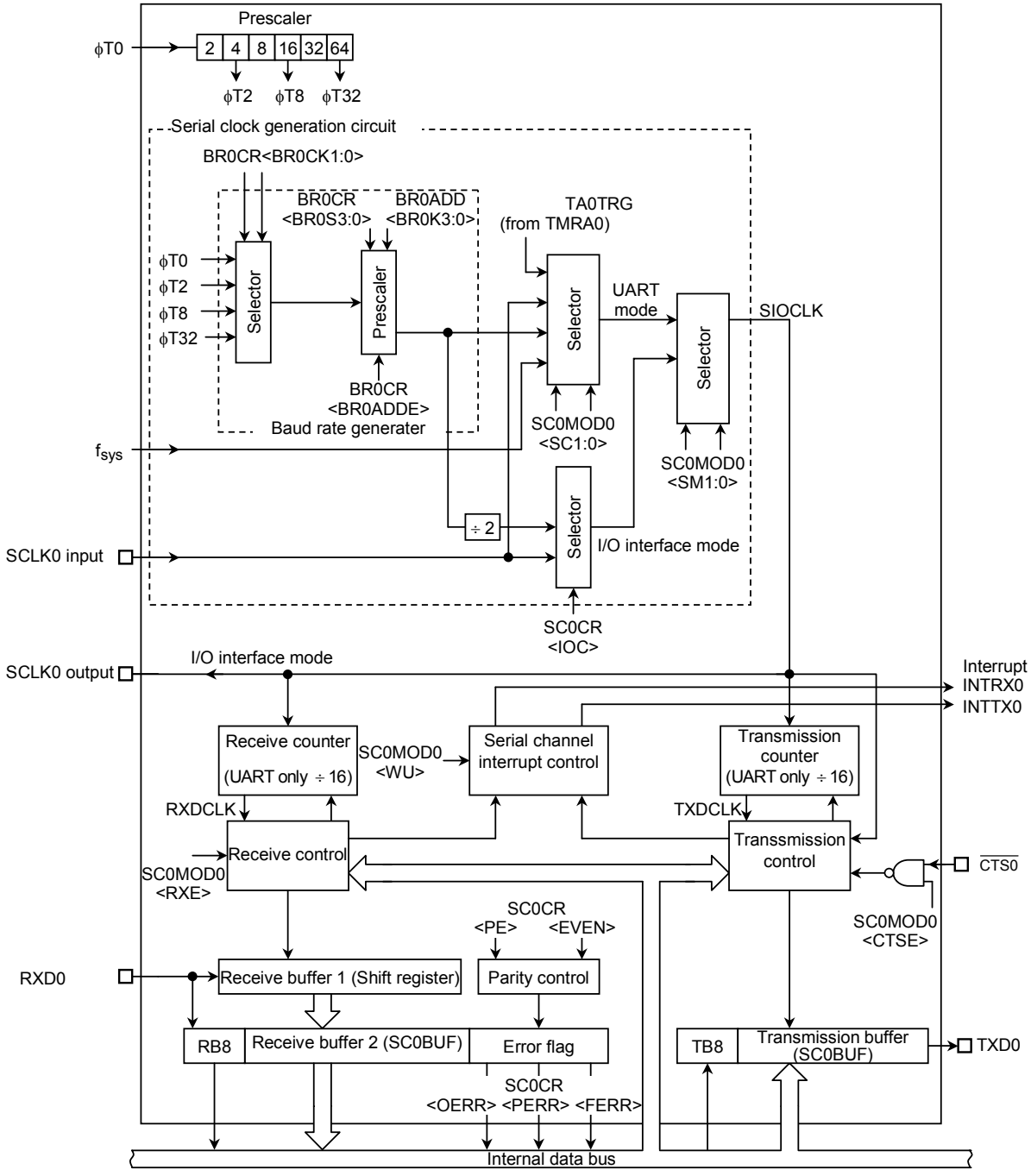


Figure 3.10.2 Block Diagram of SIO0

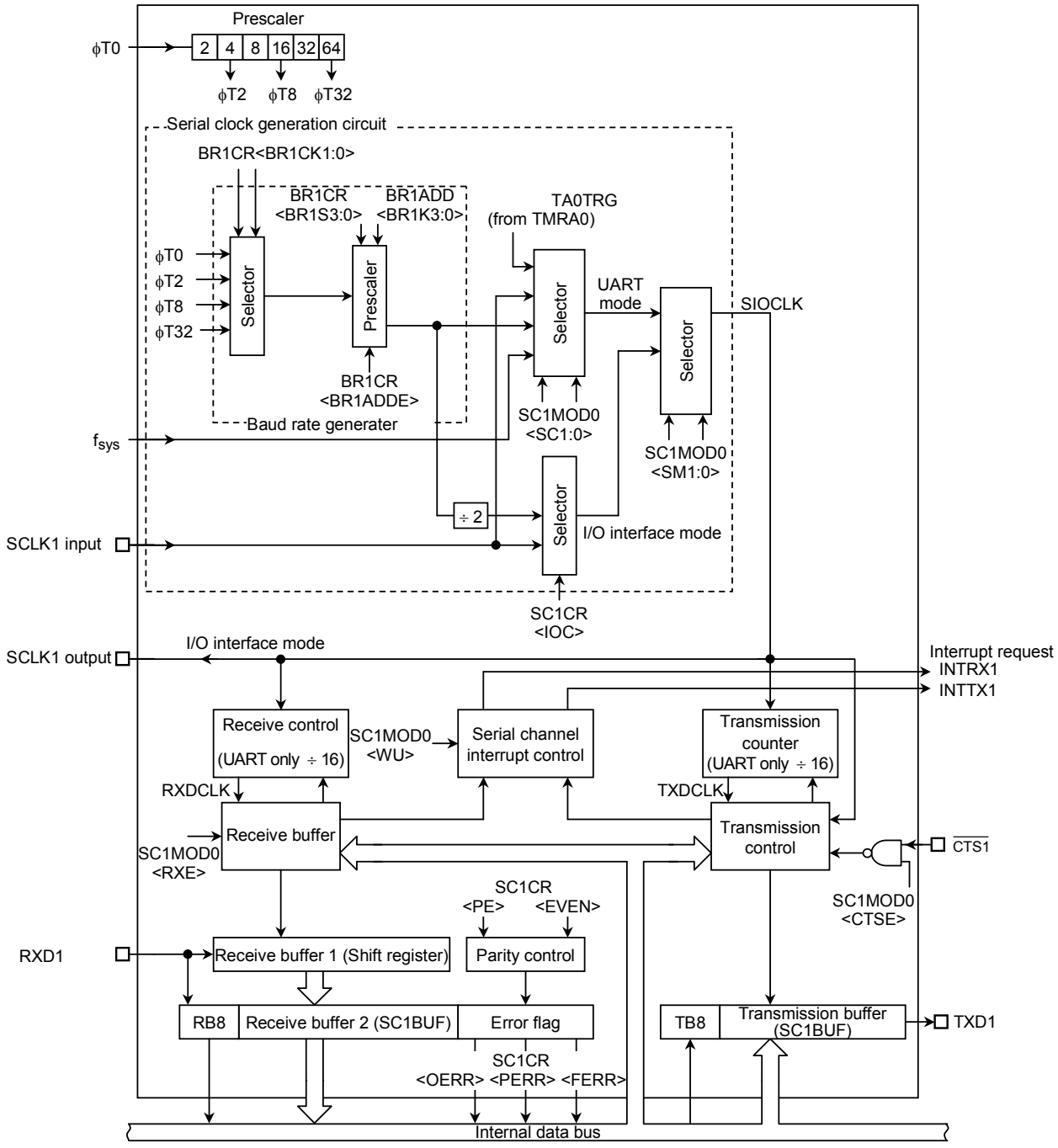


Figure 3.10.3 Block Diagram of SIO1

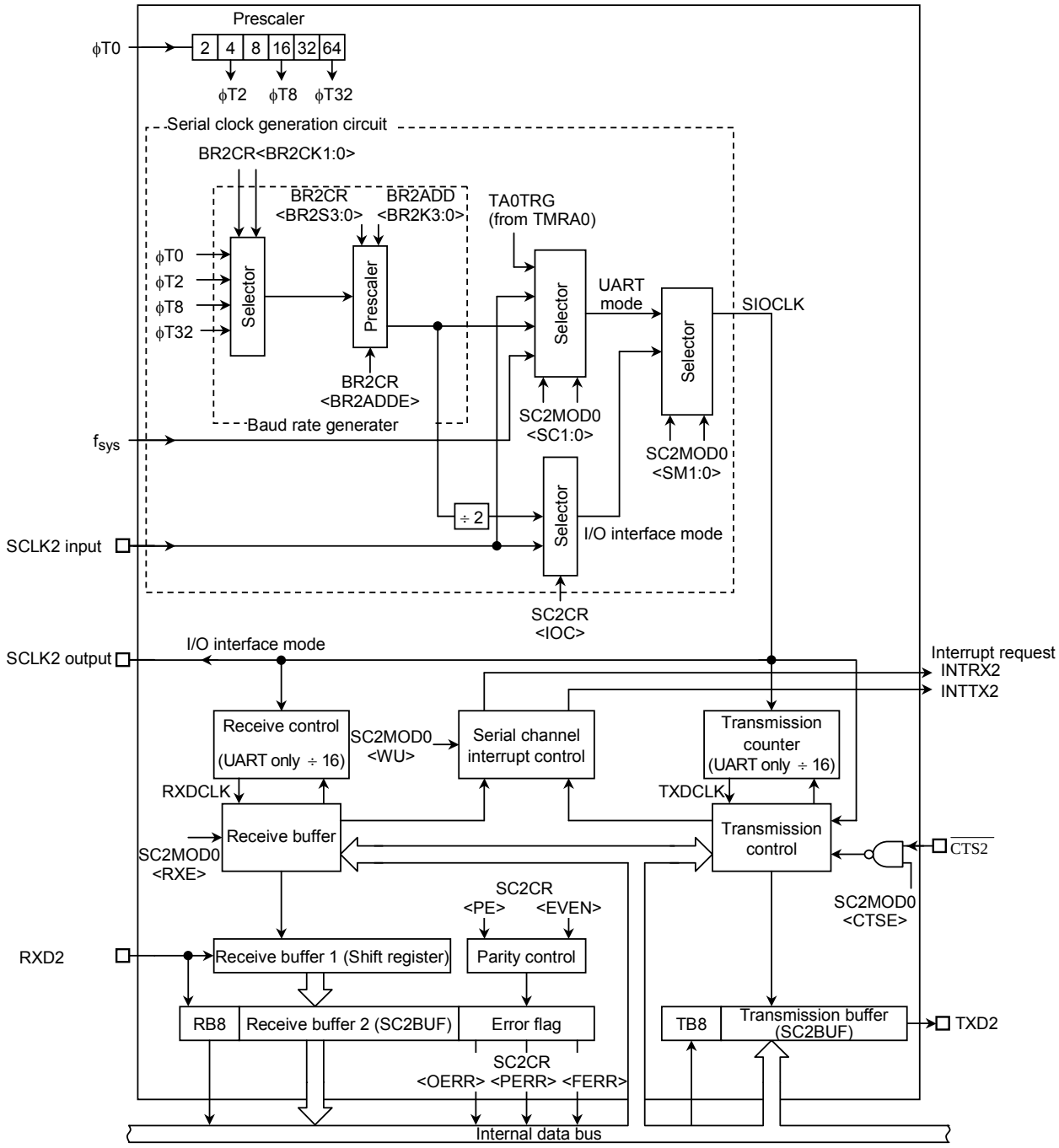


Figure 3.10.4 Block Diagram of SIO2

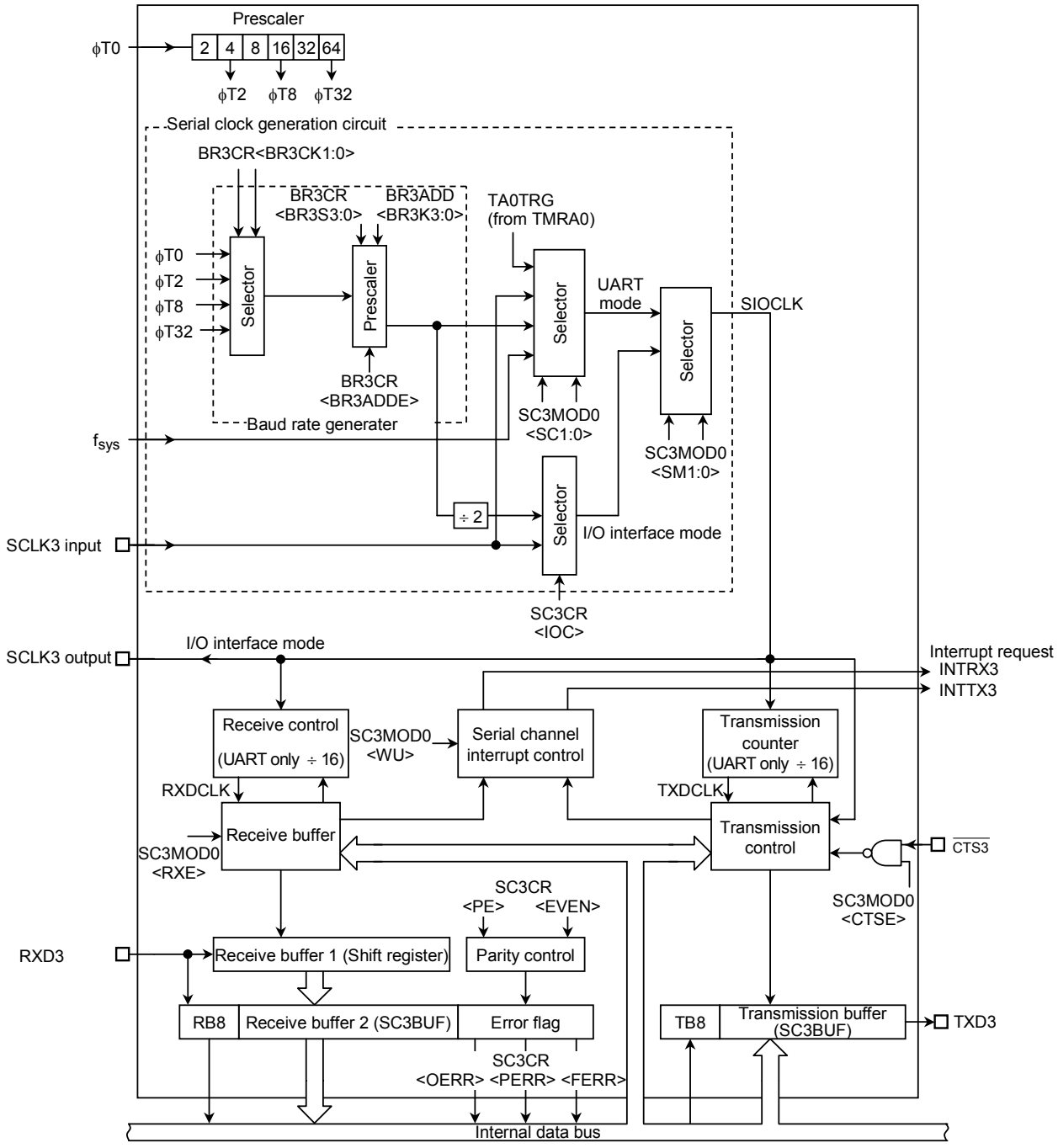


Figure 3.10.5 Block Diagram of SIO3

3.10.2 Operation of Each Circuit

(1) Prescaler

There is a 6-bit prescaler for generating a clock to SIO0. The prescaler can be run only case of selecting the baud rate generator as the serial transfer clock.

Table 3.10.2 shows prescaler clock resolution into the baud rate generator.

Table 3.10.2 Prescaler Clock Resolution to Baud Rate Generator

System Clock <SYSCK>	Clock Gear <GEAR2:0>	Clock Resolution			
		ϕT0	ϕT2	ϕT8	ϕT32
0 (fc)	000 (fc)	2^2/ fc	2^4/ fc	2^6/ fc	2^8/ fc
	001 ($^{fc}/_{2}$)	2^3/ fc	2^5/ fc	2^7/ fc	2^9/ fc
	010 ($^{fc}/_{4}$)	2^4/ fc	2^6/ fc	2^8/ fc	2^{10}/ fc
	011 ($^{fc}/_{8}$)	2^5/ fc	2^7/ fc	2^9/ fc	2^{11}/ fc
	100 ($^{fc}/_{16}$)	2^6/ fc	2^8/ fc	2^{10}/ fc	2^{12}/ fc

XXX:Don't care

The serial interface baud rate generator selects between 4 clock inputs: ϕT0, ϕT2, ϕT8, and ϕT32 among the prescaler outputs.

(2) Baud rate generator

The baud rate generator is a circuit that generates transmission and receiving clocks that determine the transfer rate of the serial channels.

The input clock to the baud rate generator, ϕT0, ϕT2, ϕT8, or ϕT32, is generated by the 6-bit prescaler which is shared by the timers. One of these input clocks is selected using the BR0CR<BR0CK1:0> field in the baud rate generator control register.

The baud rate generator includes a frequency divider, which divides the frequency by 1 or N + (16 – K)/16 to 16 values, determining the transfer rate.

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

- In UART mode

(1) When BR0CR<BR0ADDE> = 0

The settings BR0ADD<BR0K3:0> are ignored. The baud rate generator divides the selected prescaler clock by N (N = 1, 2, 3 ... 16), which is set in BR0CR<BR0S3:0>.

(2) When BR0CR<BR0ADDE> = 1

The N + (16 – K)/16 division function is enabled. The baud rate generator divides the selected prescaler clock by N + (16 – K)/16 using the value of N (N = 2, 3 ... 15) set in BR0CR<BR0S3:0> and the value of K (K = 1, 2, 3 ... 15) set in BR0ADD<BR0K3:0>.

Note: If N = 1 and N = 16, the N + (16 – K)/16 division function is disabled. Clear BR0CR<BR0ADDE> register to "0".

- In I/O interface mode

The N + (16 – K)/16 division function is not available in I/O interface mode. Clear BR0CR<BR0ADDE> to 0 before dividing by N.

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

- UART mode

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

- I/O interface mode

$$\text{Baud rate} = \frac{\text{Input clock of baud rate generator}}{\text{Frequency divider for baud rate generator}} \div 2$$

- Integer divider (N divider)

  For example, when the f_c = 12.288 MHz, the input clock frequency = ϕT2, the frequency divider N (BR0CR<BR0S3:0>) = 5, and BR0CR<BR0ADDE> = 0, the baud rate in UART mode is as follows:

  * Clock state

    System clock: High speed (fc)
    High speed gear: 1 time (fc)

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

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

Note: The N + (16 – K)/16 division function is disabled and setting BR0ADD<BR0K3:0> is invalid.

- N + (16 – K)/16 divider (UART mode only)

  Accordingly, when fc = 4.8 MHz, the input clock frequency = ϕT0, the frequency divider N (BR0CR<BR0S3:0>) = 7, K (BR0ADD<BR0K3:0>) = 3, and BR0CR<BR0ADDE> = 1, the baud rate is as follows:

  * Clock state

    System clock: High speed (fc)
    High speed gear: 1 time (fc)

  $$\text{Baud rate} = \frac{fc/4}{7 + \frac{(16 - 3)}{16}} \div 16$$

  $$= 4.8 \times 10^6 \div 16 \div (7 + \frac{13}{16}) \div 16 = 9600 \text{ (bps)}$$

  Table 3.10.3 and Table 3.10.4 show examples of UART mode transfer rates.

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

  - In UART mode

    Baud rate = External clock input frequency ÷ 16

    It is necessary to satisfy (External clock input cycle) ≥ 4/fc

  - In I/O interface mode

    Baud rate = External clock input frequency

    It is necessary to satisfy (External clock input cycle) ≥ 16/fc

Table 3.10.3 UART Baud Rate Selection

(when using baud rate generater and BR0CR<BR0ADDE> = 0) Unit (kbps)

fc [MHz]	Input Clock / Divider N (Set to BR0CR<BR0S3:0>)	φT0	φT2	φT8	φT32
9.830400	2	76.800	19.200	4.800	1.200
↑	4	38.400	9.600	2.400	0.600
↑	8	19.200	4.800	1.200	0.300
↑	0	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
↑	A	19.200	4.800	1.200	0.300
14.745600	2	115.200			
↑	3	76.800	19.200	4.800	1.200
↑	6	38.400	9.600	2.400	0.600
↑	C	19.200	4.800	1.200	0.300

Note 1: Transfer rates in I/O interface mode are eight times faster than the values given above.

Note 2: The values in this table are calculated for when f_{SYS} is selected as the system clock, f_{SYS} /1 is selected as the clock gear.

Table 3.10.4 UART Baud Rate Selection

(when using trigger output of TMRA0 and input clcok of TMRA0 is φT1.)

Unit (kbps)

fc / TA0REG0	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	96		76.8	62.5	48
2H	48		38.4	31.25	24
3H	32	31.25			16
4H	24		19.2		12
5H	19.2				9.6
8H	12		9.6		6
AH	9.6				4.8
10H	6		4.8		3
14H	4.8				2.4

Method for calculating the transfer rate (when TMRA0 is used):

$$\text{Transfer rate} = \frac{f_{FPH}}{\text{TA0REG} \times 2^{3} \times 16}$$

(When input clock of TMRA0 is φT1)

Note 1: The TMRA0 match detect signal cannot be used as the transfer clock in I/O interface mode.

Note 2: The values in this table are calculated for when f_c is selected as the system clock, f_c is selected as the clock gear.

(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<SC1:0> setting determines whether the baud rate generator clocks, the internal system clock f_{SYS}, the trigger output signal from TMRA0 or the external clock (SCLK0 pin) is used to generate the basic clock SIOCLK.

(4) Receiving counter

The receiving counter is a 4-bit binary counter used in UART mode that 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 are taken to be 0.

(5) Receiving control

- In I/O interface mode

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

  In SCLK input mode with the setting SC0CR<IOC> = 1, the RXD0 pin is sampled on the rising or falling edge of the SCLK input, according to the SC0CR<SCLKS> setting.

- In UART mode

  The receiving control block has a circuit that 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 reads receiving buffer 2 (SC0BUF), the received data can be stored in receiving buffer 1.

However, unless receiving buffer 2 (SC0BUF) is read before all bits of the next data are received by receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of receiving buffer 1 will be lost, although the contents of 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.

SIO interruption mode can be set up by the SIMC register.

(7) Transmission counter

The transmission counter is a 4-bit binary counter that 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.

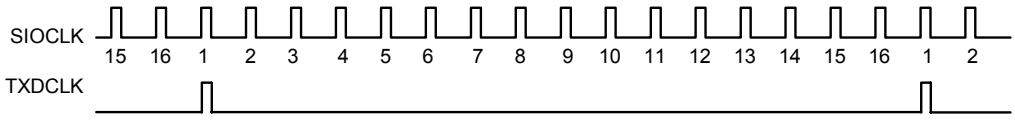


Figure 3.10.6 Generation of Transmission Clock

(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 or falling edge of the shift clock which is output on the SCLK0 pin, according to the SC0CR<SCLKS> setting.

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

- In UART mode

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

Handshake function

Serial channels 0 and 1 each has a $\overline{CTS}$ pin. Use of this pin allows data can be sent in units of one data format; thus, overrun errors can be avoided. The handshake functions is enabled or disabled by the SC0MOD0<CTSE> setting.

When the $\overline{CTS0}$ pin condition is high level, after completed the current data transmission, data transmission is halted until the $\overline{CTS0}$ pin state is low again. However, the INTTX0 interrupt is generated, it requests the next send data to the CPU. The next data is written in the transmission buffer and data transmission is halted.

Though there is no $\overline{RTS}$ pin, a handshake function can be easily configured by setting any port assigned to be the $\overline{RTS}$ function. The $\overline{RTS}$ should be output "High" to request send data halt after data receive is completed by software in the receive interrupt routine.

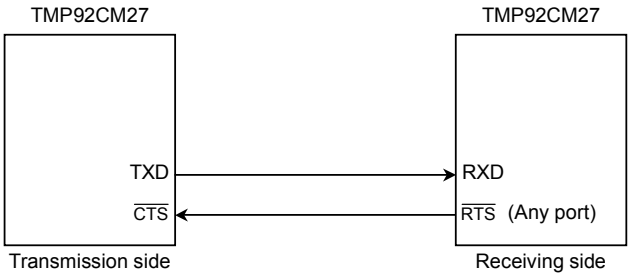
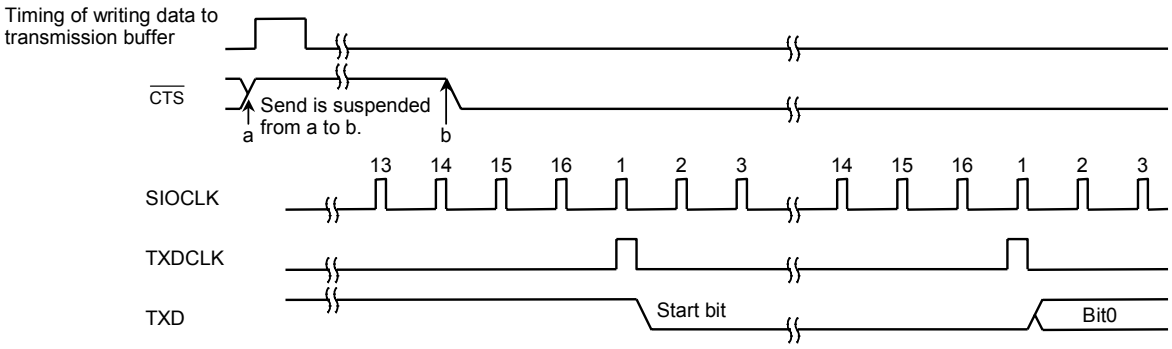


Figure 3.10.7 Handshake Function



Note 1: If the $\overline{CTS}$ signal goes high during transmission, will be stop next transmission data after completion of the current transmission.

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

Figure 3.10.8 $\overline{CTS}$ (Clear to send) Signal Timing

(9) Transmission buffer

The transmission buffer (SC0BUF) shifts out and sends the transmission data written from the CPU form the least significant bit in order. When all the bits are shifted out, the transmission buffer becomes empty and 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 recommended flow when the overrun-error is generated.

(INTRX interrupt routine)

1) Read receiving buffer
2) Read error flag
3) if <OERR> = "1"
   then
   a) Set to disable receiving (Program "0" to SC0MOD0<RXE>)
   b) Wait to terminate current frame
   c) Read receiving buffer
   d) Read error flag
   e) Set to enable receiving (Program "1" to SC0MOD0<RXE>)
   f) Request to transmit again
4) Other

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 Bits	8 Bits + Parity	8 Bits, 7 Bits + Parity, 7 Bits
Interrupt generation timing	Center of last bit (Bit8)	Center of last bit (Parity bit)	Center of stop bit
Framing error generation timing	Center of stop bit	Center of stop bit	Center of stop bit
Parity error generation timing	–	Center of last bit (Parity bit)	Center of stop bit
Overrun error generation timing	Center of last bit (Bit8)	Center of last bit (Parity bit)	Center of stop bit

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

Transmission

Mode	9 Bits	8 Bits + Parity	8 Bits, 7 Bits + Parity, 7 Bits
Interrupt generation timing	Just before stop bit is transmitted	Just before stop bit is transmitted	Just before stop bit is transmitted

2. In I/O interface mode

Transmission interrupt timing	SCLK output mode	Immediately after last bit data. (See Figure 3.10.31)
	SCLK input mode	Immediately after rise of last SCLK signal rising mode, or immediately after fall in falling mode. (See Figure 3.10.32)
Receiving interrupt timing	SCLK output mode	Timing used to transfer received to data receive buffer 2 (SC0BUF) (e.g., immediately after last SCLK). (See Figure 3.10.33)
	SCLK input mode	Timing used to transfer received data to receive buffer 2 (SC0BUF) (e.g., immediately after last SCLK). (See Figure 3.10.34)

3.10.3 SFRs

SC0MOD0 (1202H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data bit8	Handshake function control 0: CTS disable 1: CTS enable	Receive control 0: Receive disable 1: Receive enable	Wakeup 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: Timer A0 trigger 01: Baud rate generator 10: Internal clock f_{sys} 11: External clcok (SCLK0 input)	

Serial transmission clock source (UART)

00	TMRA0 trigger output signal
01	Baud rate generator
10	Internal clock f_{sys}
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 mode	7-bit mode
10		8-bit mode
11		9-bit mode

Wakeup 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 ($\overline{CTS}$ pin)

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

Figure 3.10.9 Serial Mode Control Register 0 (for SIO0 and SC0MOD0)

SC1MOD0 (120AH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data bit8	Handshake function control 0: CTS disable 1: CTS enable	Receive control 0: Receive disable 1: Receive enable	Wakeup 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: Timer A0 trigger 01: Baud rate generator 10: Internal clock f_{sys} 11: External clcok (SCLK1 input)	

Serial transmission clock source (UART)

00	TMRA0 trigger output signal
01	Baud rate generator
10	Internal clock f_{sys}
11	External clock (SCLK1 input)

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

Serial transmission mode

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

Wakeup 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 ($\overline{CTS}$ pin)

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

Figure 3.10.10 Serial Mode Control Register 0 (for SIO1 and SC1MOD0)

SC2MOD0 (1212H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data bit8	Handshake function control 0: CTS disable 1: CTS enable	Receive control 0: Receive disable 1: Receive enable	Wakeup 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: Timer A0 trigger 01: Baud rate generator 10: Internal clock f_{sys} 11: External clcok (SCLK2 input)	

Serial transmission clock source (UART)

00	TMRA0 trigger output signal
01	Baud rate generator
10	Internal clock f_{sys}
11	External clock (SCLK2 input)

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

Serial transmission mode

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

Wakeup 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 ($\overline{CTS}$ pin)

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

Figure 3.10.11 Serial Mode Control Register 0 (for SIO2 and SC2MOD0)

SC3MOD0 (121AH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data bit8	Handshake function control 0: CTS disable 1: CTS enable	Receive control 0: Receive disable 1: Receive enable	Wakeup 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: Timer A0 trigger 01: Baud rate generator 10: Internal clock f_{sys} 11: External clcok (SCLK3 input)	

Serial transmission clock source (UART)

00	TMRA0 trigger output signal
01	Baud rate generator
10	Internal clock f_{sys}
11	External clock (SCLK3 input)

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

Serial transmission mode

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

Wakeup 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 ($\overline{CTS}$ pin)

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

Figure 3.10.12 Serial Mode Control Register 0 (for SIO3 and SC3MOD0)

SC0CR (1201H)

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

I/O interface input clock selection

0	Baud rate generator
1	SCLK0 pin input

Edge selection for SCLK0 pin (I/O mode)

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 enables

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data bit8

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

Figure 3.10.13 Serial Control Register (for SIO0 and SC0CR)

SC1CR (1209H)

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

I/O interface input clock selection

0	Baud rate generator
1	SCLK1 pin input

Edge selection for SCLK1 pin (I/O mode)

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 enables

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data bit8

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

Figure 3.10.14 Serial Control Register (for SIO1 and SC1CR)

SC2CR (1211H)

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

I/O interface input clock selection

0	Baud rate generator
1	SCLK2 pin input

Edge selection for SCLK2 pin (I/O mode)

0	Transmits and receives data on rising edge of SCLK2. (↑)
1	Transmits and receives data on falling edge of SCLK2. (↓)

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

Parity addition enables

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data bit8

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

Figure 3.10.15 Serial Control Register (for SIO2 and SC2CR)

SC3CR (1219H)

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

I/O interface input clock selection

0	Baud rate generator
1	SCLK3 pin input

Edge selection for SCKL3 pin (I/O mode)

0	Transmits and receives data on rising edge of SCLK3.
1	Transmits and receives data on falling edge of SCLK3.

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

Parity addition enables

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data bit8

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

Figure 3.10.16 Serial Control Register (for SIO3 and SC3CR)

BR0CR (1203H)

	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	Always write "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 divisions enable

0	Disable
1	Enable

Setting the input clock of baud rate generator

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

BR0ADD (1204H)

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

Sets baud rate generator frequency divisor

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

Note1:Availability of +(16-K)/16 division function

N	UART mode	I/O mode
2 to 15	○	×
1 , 16	×	×

The baud rate generator can be set "1" in UART mode and disable +(16-K)/16 division function.Don't use in I/O interface mode.

Note2:Set BR0CR <BR0ADDE> to 1 after setting K (K = 1 to 15) to BR0ADD<BR0K3:0> when +(16-K)/16 division function is used.

Figure 3.10.17 Baud Rate Generator Control (for SIO0, BR0CR, and BR0ADD)

BR1CR (120BH)

	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	Always write "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Divided frequency setting			

+ (16 − K)/16 divisions 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 (120CH)

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

Note1:Availability of +(16-K)/16 division function

N	UART mode	I/O mode
2 to 15	○	×
1 , 16	×	×

The baud rate generator can be set "1" in UART mode and disable +(16-K)/16 division function.Don't use in I/O interface mode.

Note2:Set BR1CR <BR1ADDE> to 1 after setting K (K = 1 to 15) to BR1ADD<BR1K3:0> when +(16-K)/16 division function is used.

Figure 3.10.18 Baud Rate Generater Control (for SIO1, BR1CR, and BR1ADD)

BR2CR (1213H)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR2ADDE	BR2CK1	BR2CK0	BR2S3	BR2S2	BR2S1	BR2S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Always write "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Divided frequency setting			

+ (16 − K)/16 divisions 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

BR2ADD (120CH)

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

Baud rate generator frequency divisor setting

	BR2CR<BR2ADDE> = 1		BR2CR<BR2ADDE> = 0
BR2CR <BR2S3:0> / BR2ADD <BR2K3:0>	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) ~ 1111 (N = 15)	0001 (N = 1) (UART only) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	Divided by N
0001 (K = 1) ~ 1111 (K = 15)	Disable	Divided by N + (16 − K) / 16	

Note1:Availability of +(16-K)/16 division function

N	UART mode	I/O mode
2 to 15	○	×
1 , 16	×	×

The baud rate generator can be set "1" in UART mode and disable +(16-K)/16 division function.Don't use in I/O interface mode.

Note2:Set BR2CR <BR2ADDE> to 1 after setting K (K = 1 to 15) to BR2ADD<BR2K3:0> when +(16-K)/16 division function is used.

Figure 3.10.19 Baud Rate Generater Control (for SIO2, BR2CR, and BR2ADD)

BR3CR (121BH)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR3ADDE	BR3CK1	BR3CK0	BR3S3	BR3S2	BR3S1	BR3S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Always write "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Divided frequency setting			

+ (16 − K)/16 divisions 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

BR3ADD (121CH)

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

Baud rate generator frequency divisor setting

	BR3CR<BR3ADDE> = 1		BR3CR<BR3ADDE> = 0
BR3CR <BR3S3:0> / BR3ADD <BR3K3:0>	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) ~ 1111 (N = 15)	0001 (N = 1) (UART only) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	Divided by N
0001 (K = 1) ~ 1111 (K = 15)	Disable	Divided by N + (16 − K) / 16	

Note1:Availability of +(16-K)/16 division function

N	UART mode	I/O mode
2 to 15	○	×
1 , 16	×	×

The baud rate generator can be set "1" in UART mode and disable +(16-K)/16 division function.Don't use in I/O interface mode.

Note2:Set BR3CR <BR3ADDE> to 1 after setting K (K = 1 to 15) to BR3ADD<BR3K3:0> when +(16-K)/16 division function is used.

Figure 3.10.20 Baud Rate Generater Control (for SIO3, BR3CR, and BR3ADD)

SC0BUF (1200H)

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

Note: Prohibit read-modify-write for SC0BUF

Figure 3.10.21 Serial Transmission/Receiving Buffer Register (for SIO0 and SC0BUF)

SC0MOD1 (1205H)

	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 3.10.22 Serial Mode Control Regsiter 1 (for SIO0 and SC0MOD1)

SC1BUF (1208H)

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

Note: Prohibit read-modify-write for SC1BUF

Figure 3.10.23 Serial Transmission/Receiving Buffer Register (for SIO1 and SC1BUF)

SC1MOD1 (120DH)

	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 3.10.24 Serial Mode Control Regsiter 1 (for SIO1 and SC1MOD1)

SC2BUF (1210H)

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

Note: Prohibit read-modify-write for SC2BUF

Figure 3.10.25 Serial Transmission/Receiving Buffer Register (for SIO2 and SC2BUF)

SC2MOD1 (1215H)

	7	6	5	4	3	2	1	0
Bit symbol	I2S2	FDPX2						
Read/Write	R/W	R/W						
After reset	0	0						
Function	IDLE2 0: Stop 1: Run	Duplex 0: Half 1: Full						

Figure 3.10.26 Serial Mode Control Regsiter 1 (for SIO2 and SC2MOD1)

SC3BUF (1218H)

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

Note: Prohibit read-modify-write for SC3BUF

Figure 3.10.27 Serial Transmission/Receiving Buffer Register (for SIO3 and SC3BUF)

SC3MOD1 (121DH)

	7	6	5	4	3	2	1	0
Bit symbol	I2S3	FDPX3						
Read/Write	R/W	R/W						
After reset	0	0						
Function	IDLE2 0: Stop 1: Run	Duplex 0: Half 1: Full						

Figure 3.10.28 Serial Mode Control Regsiter 1 (for SIO3 and SC3MOD1)

3.10.4 Operation in 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 mode to input external synchronous clock SCLK.

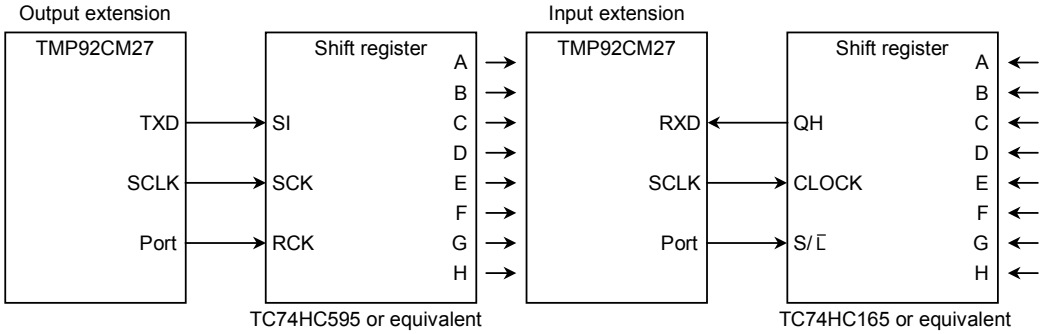


Figure 3.10.29 Example of SCLK Output Mode Connection

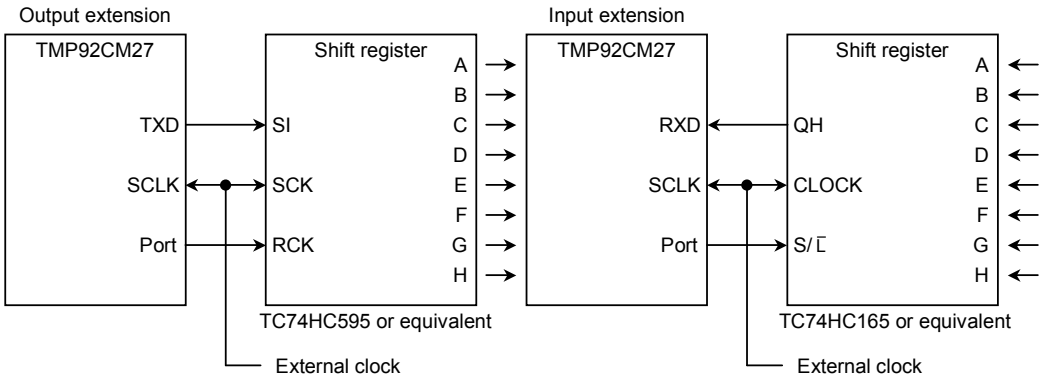


Figure 3.10.30 Example of SCLK Output Mode Connection

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 data is outputted, INTES0<ITX0C> will be set to generate the INTTX0 interrupt.

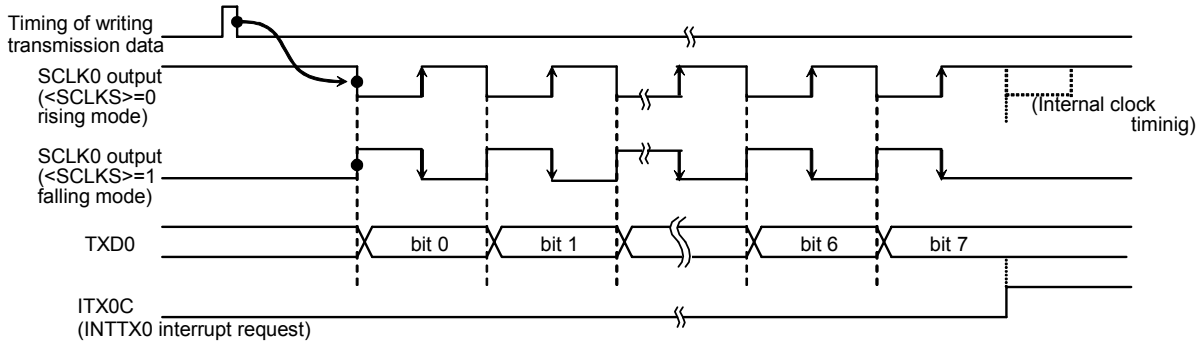


Figure 3.10.31 Transmission Operation in I/O Interface Mode (SCLK0 output mode)

In SCLK input mode, 8-bit data is output from the TXD0 pin when the SCLK0 input becomes active after the data has been written to the transmission buffer by the CPU.

When all data is outputted, INTES0<ITX0C> will be set to generate INTTX0 interrupt.

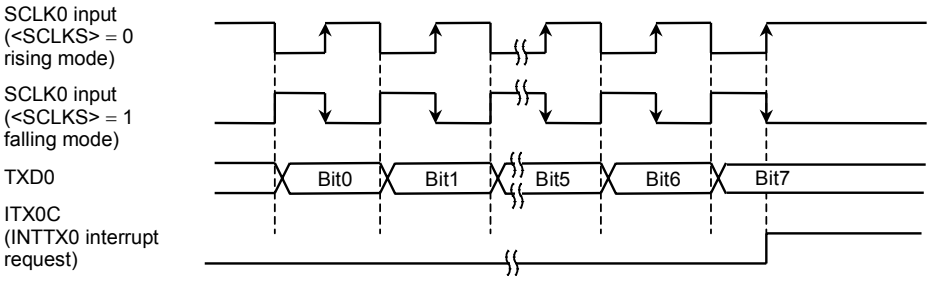


Figure 3.10.32 Transmission Operation in I/O Interface Mode (SCLK0 input mode)

2. Receiving

In SCLK output mode, the synchronous clock is outputted from SCLK0 pin and the data is shifted to receiving buffer 1. This starts when the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred to receiving buffer 2 (SC0BUF according to the timing shown below) and INTES0<IRX0C> will be set to generate INTRX0 interrupt.

The outputting for the first SCLK0 starts by setting SC0MOD0<RXE> to 1.

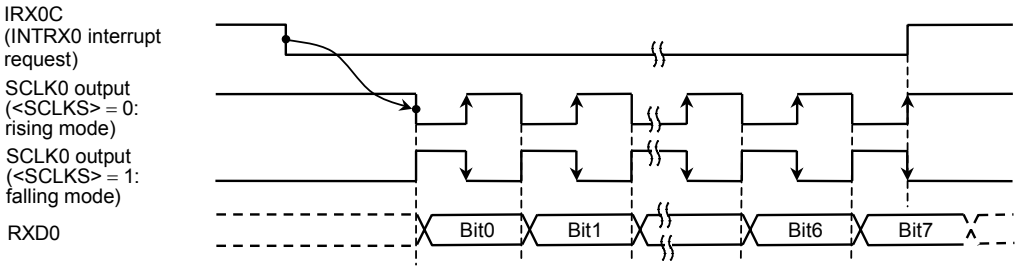


Figure 3.10.33 Receiving Operation in I/O Interface Mode (SCLK0 output mode)

In SCLK input mode, the data is shifted to receiving buffer 1 when the SCLK input becomes active after the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data is received, the data will be shifted to receiving buffer 2 (SC0BUF according to the timing shown below) and INTES0<IRX0C> will be set again to be generate INTRX0 interrupt.

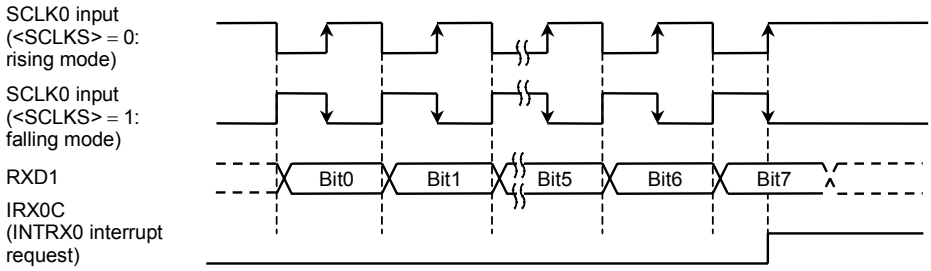


Figure 3.10.34 Receiving Operation in I/O Interface Mode (SCLK0 input mode)

Note: If receiving, set to the receive enable state (SC0MOD0<RXE> = 1) in both SCLK input mode and output mode.

3. Transmission and receiving (Full duplex mode)

When the full duplex mode is used, set the level of receive interrupt to "0" and set enable the interrupt level (1 to 6) to the transfer interrupts. In the transfer interrupt program, the receiving operation should be done like the below example before setting the next transfer data.

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

Main routine

	7	6	5	4	3	2	1	0	
INTES0	x	0	0	1	x	0	0	0	Set transmission interrupt level, and disable receiving interrupt.
PACR	–	–	–	–	–	1	1	0	Set to PA0 (RXD0), PA1 (TXD0), and PA2 (SCLK0).
PAFC	–	–	–	–	–	1	1	1	
PAFC2	x	x	–	–	–	0	–	–	
SC0MOD0	0	0	0	0	0	0	0	0	Set to I/O interface mode.
SC0MOD1	1	1	0	0	0	0	0	0	Set to full duplex mode.
SC0CR	0	0	0	0	0	0	0	0	Output SCLK, select rising edge.
BR0CR	0	0	1	1	0	1	0	0	Set to 9600 bps.
SC0MOD0	0	0	1	0	0	0	0	0	Set receive to enable.
SC0BUF	*	*	*	*	*	*	*	*	Set transmission data.

Transmission interrupt routine

Acc SC0BUF									Read receiving data.
SC0BUF	*	*	*	*	*	*	*	*	Set transmission data.

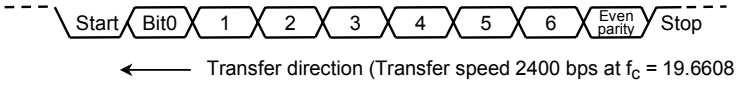
X: Don't care, –: No change

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

7-bit UART mode is selected by setting serial channel mode register SC0MOD0<SM1:0> 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).

Example: When transmitting data of the following format, the control registers should be set as described below. This explanation applies to channel 0.



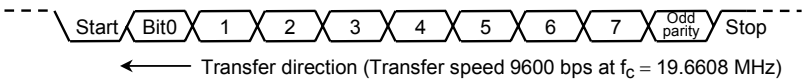
	7	6	5	4	3	2	1	0	
PACR	–	–	–	–	–	–	–	1	Set PA1 to as TXD0 pin.
PAFC	–	–	–	–	–	–	–	1	
PAFC2	X	X	X	–	X	X	0	–	
SC0MOD	X	0	–	X	0	1	0	1	Set to 7-bit UART mode.
SC0CR	X	1	1	X	X	X	0	0	Add even parity.
BR0CR	0	0	1	0	1	0	0	0	Set to 2400 bps.
INTES0	X	1	0	0	–	–	–	–	Set INTTX0 interrupt to enable, set to level 4.
SC0BUF	*	*	*	*	*	*	*	*	Set transmission data.

X : Don't care, – : No change

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

8-bit UART mode is selected by setting SC0MOD0<SM1:0> 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).

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



Main routine

		7	6	5	4	3	2	1	0	
PACR	←	–	–	–	–	–	–	–	0	Set PA0 (RXD0) to input pin.
PAFC	←	–	–	–	–	–	–	–	1	
SC0MOD	←	–	0	1	X	1	0	0	1	Set to 8-bit UART mode, set receives to enable.
SC0CR	←	X	0	1	X	X	X	0	0	Add odd parity.
BR0CR	←	0	0	0	1	1	0	0	0	Set to 9600 bps.
INTES0	←	X	–	–	–	X	1	0	0	Set INTTX0 interrupt to enable, set to level 4.

Interrupt routine processing

Acc	←	SC0CR AND 00011100	Check for error.
if Acc	≠	0 then ERROR	
Acc	←	SC0BUF	Read receiving data.

X : Don't care, – : No change

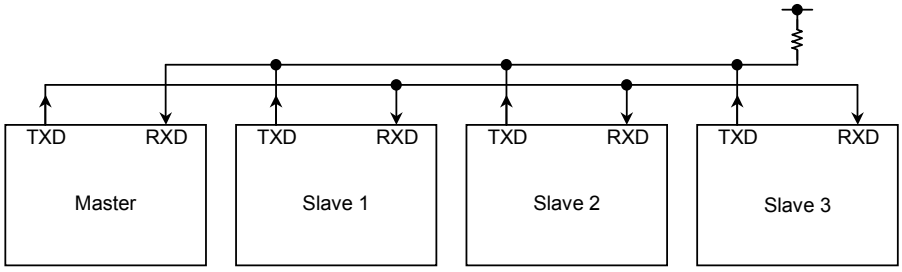
(4) Mode 3 (9-bit UART mode)

9-bit UART mode is selected by setting SC0MOD0<SM1:0> to 11. In this mode parity bit cannot be added.

In the case of transmission the MSB (9th bit) is programmed 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.

Wakeup function

In 9-bit UART mode, the wakeup function for slave controllers is enabled by setting SC0MOD0<WU> to 1. The interrupt INTRX0 occurs only when <RB8> = 1.

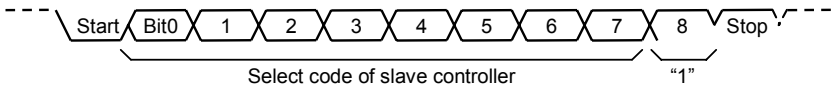


Note: The TXD pin of each slave controller must be in open-drain output mode.

Figure 3.10.35 Serial Link Using Wakeup Function

Protocol

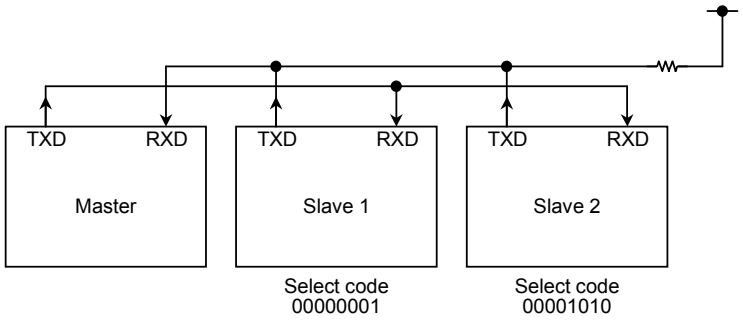
1. Select 9-bit UART mode on the master and slave controllers.
2. Set the SC0MOD0<WU> bit on each slave controller to 1 to enable data receiving.
3. The master controller transmits one-frame data including the 8-bit select code for the slave controllers. The MSB (Bit8) <TB8> is set to "1".



4. Each slave controller receives the above frame. If it matches with own select code, clears <WU> bit to "0".
5. The master controller transmits data to the specified slave controller whose SC0MOD0<WU> bit is cleared to "0". The MSB (Bit8) <TB8> is cleared to "0".

6. The other slave controllers (whose <WU> bits remain at 1) ignore the received data because their MSB (Bit8 or <RB8>) are set to "0", disabling INTRX0 interrupts. The slave controller (<WU> bit = "0") can transmit data to the master controller, and it is possible to indicate the end of data receiving to the master controller by this transmission.

Example: To link two slave controllers serially with the master controller using the system clock f_{SYS} as the transfer clock.



- Master controller setting

Main routine

		7	6	5	4	3	2	1	0	
PACR	←	–	–	–	–	–	–	1	0	Set PA0 to RXD0, and set PA1 to TXD0 pin.
PAFC	←	–	–	–	–	–	–	1	1	
PAFC2	←	X	X	X	–	X	X	0	X	
INTES0	←	X	1	0	0	X	1	0	1	Set INTTX0 to enable, and set interrupt level to level 4. Set INTRX0 to enable, and set interrupt level to level 5.
SC0MOD0	←	1	0	1	0	1	1	1	0	Set to 9-bit UART mode, and set transfer clock to f_{SYS}.
SC0BUF	←	0	0	0	0	0	0	0	1	Set select code of slave 1.

Interrupt routine (INTTX0)

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

- Slave setting

Main routine

		7	6	5	4	3	2	1	0	
PACR	←	–	–	–	–	–	–	1	0	Set PA0 to RXD0, and PA1 to TXD0 (open-drain output).
PAFC	←	–	–	–	–	–	–	1	1	
PAFC2	←	X	X	X	–	X	X	1	X	
INTES0	←	X	1	0	1	X	1	1	0	Set INTRX0 to enable, and set interrupt level to level 5. Set INTRX0 to enable, and set interrupt level to level 6
SC0MOD0	←	0	0	1	1	1	1	1	0	Set to <WU> = "1" in 9-bit UART mode transfer clock f_{SYS}.

Interrupt routine (INTRX0)

Acc ← SC0BUF
if Acc = Select code
Then SC0MOD0 ← – – – – 0 – – – – Clear to <WU> = "0".
X : Don't care, – : No change

3.10.5 Support for IrDA Mode

SIO0 includes support for the IrDA 1.0 infrared data communication specification.Figure 3.10.36 shows the block diagram.

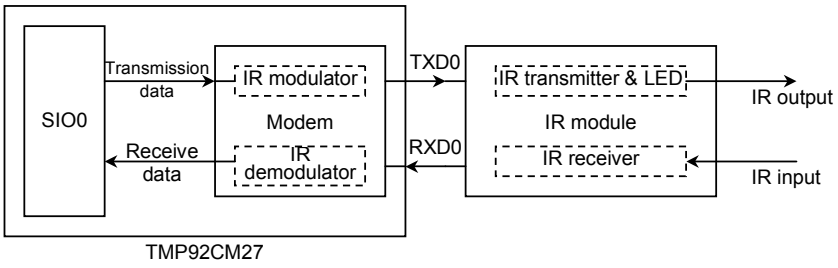


Figure 3.10.36 Block Diagram of IrDA (SIO0)

(1) Modulation of transmission data

When the transmission data is 0, output "H" level with either 3/16 or 1/16 times for width of baud-rate (Selectable in software). Moreover, pulse width is chosen in SIR0CR<PLSEL>.When data is "1", modem output "L" level.

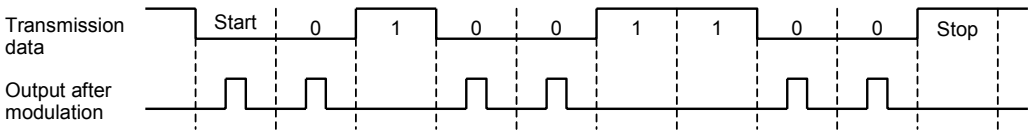


Figure 3.10.37 Example of Modulation of Transmission Data (SIO0)

(2) Modulation of receiving data

When the receive data has the effective high level pulse width (Software selectable), the modem outputs "0" to SIO0. Otherwise modem outputs "1" to SIO0. Effective pulse width is chosen in SIR0CR<SIR0WD3:0 >.

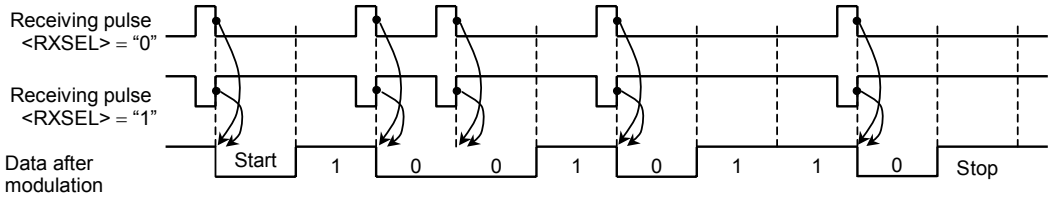


Figure 3.10.38 Example of Modulation of Receiving Data (SIO0)

(3) Data format

Format of transmission/receiving must set to data length 8-bit, without parity bit, 1 bit of stop bit.

Any other settings don't guarantee the normal operation.

(4) SFR

Figure 3.10.39 shows the control register SIR0CR. If change setting this register, must set it after set operation of transmission/receiving to disable (Both <TXEN> and <RXEN> of this register should be clear to 0).

Any changing for this register during transmission or receiving operation doesn't guarantee the normal operation.

The following example describes how to set this register:

1) SIO setting ; Set SIO side.
↓
2) LD (SIR0CR), 07H ; Set receiving effect pulse width to 16X+100ns.
3) LD (SIR0CR), 37H ; TXEN, RXEN enable the transmission and receiving.
↓
4) Transmission/receiving ; The modem operates as follows:
   - SIO0 starts transmitting.
   - IR receiver starts receiving.

(5) Notes

1. Making baud rate when using IrDA

   In baud rate during using IrDA, must set "01" to SC0MOD0<SC1:0> in SIO by using baud rate generator.

   TA0TRG, f_{SYS}, SCLK0 input of except for it can not using.

2. Output pulse width and baud rate generator during transmission IrDA

   As the IrDA 1.0 physical layer specification, the data transfer speed and infra-red pulse width is specified.

Table 3.10.5 Specification of Transfer Rate and Pulse Width

Transfer Rate	Modulation	Transfer Rate Tolerance (% of Rate)	Minimum of Pulse Width	Typical of Pulse Width 3/16	Maximum of Pulse Width
2.4 kbps	RZI	± 0.87	1.41 μs	78.13 μs	88.55 μs
9.6 kbps	RZI	± 0.87	1.41 μs	19.53 μs	22.13 μs
19.2 kbps	RZI	± 0.87	1.41 μs	9.77 μs	11.07 μs
38.4 kbps	RZI	± 0.87	1.41 μs	4.88 μs	5.96 μs
57.6 kbps	RZI	± 0.87	1.41 μs	3.26 μs	4.34 μs
115.2 kbps	RZI	± 0.87	1.41 μs	1.63 μs	2.23 μs

The infra-red pulse width is specified either baud rate T × 3/16 or 1.6 μs (1.6 μs is equal to T × 3/16 pulse width when baud rate is 115.2 kbps).

The TMP92CM27 has function which is selectable the transmission pulse width either 3/16 or 1/16. But T × 1/16 pulse width can be selected when the baud rate is equal or less than 38.4 kbps only. When 57.6 kbps and 115.2 kbps, the output pulse width should not be set to T × 1/16.

As the same reason, + (16 – K)/16 division function in the baud rate generator of SIO0 cannot be used to generate 115.2 kbps baud rate.

Also when the 38.4 kbps and 1/16 pulse width, + (16 – K)/16 division function cannot be used.

Table 3.10.6 shows baud rate and pulse width for (16 – K)/16 division function.

Table 3.10.6 Baud Rate and Pulse Width for (16 – K)/16 Division Function

Output Pulse Width	Baud Rate 115.2 kbps	57.6 kbps	38.4 kbps	19.2 kbps	9.6 kbps	2.4 kbps
T × 3/16	×	○	○	○	○	○
T × 1/16	–	–	×	○	○	○

○: Can be used (16 – K)/16 division function.
×: Cannot be used (16 – K)/16 division function.
–: Cannot be set to T × 1/16 pulse width.

SIR0CR (1207H)

	7	6	5	4	3	2	1	0
Bit symbol	PLSEL	RXSEL	TXEN	RXEN	SIR0WD3	SIR0WD2	SIR0WD1	SIR0WD0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Selection transmission pulse width 0: 3/16 1: 1/16	Receiving data logic 0: "H" pulse 1: "L" pulse	Transmission data 0: Disable 1: Enable	Receiving operation 0: Disable 1: Enable	Select receiving effective pulse width Set effective pulse width for equal or more than 2x × (Value + 1) + 100 ns Can be set: 1 to 14 Cannot be set: 0, 15			

Select receiving effective pulse width
Formula:
Receiving effective pulse width ≥ 2x × (Setting value + 1) + 100 ns
$x = 1/f_{FPH}$

0000	Cannot be set.
0001	Pulse width of equal or more than 4x + 100 ns is effective.
≀	
1110	Pulse width of equal or more than 30x + 100 ns is effective.
1111	Cannot be set.

Enable of receiving operation

0	Disable receiving operation. (Received input is ignored.)
1	Enable receiving operation.

Enable of transmission operation

0	Disable transmission operation (Input from SIO is ignored.)
1	Enable transmission operation.

Select transmission pulse width

0	Pulse width of 3/16
1	Pulse width of 1/16

Note: If a pulse width complying with the IrDA 1.0 standard (1.6μs min.) can be guaranteed with a low baud rate, setting this bit to "1" shortens the duration of infrared ray activation resulting in reduced power dissipation.

Figure 3.10.39 IrDA Control Register0 (for SIO0)

3.11 Serial Bus Interface (SBI)

The TMP92CM27 has 2-channel serial bus interface. Serial bus interface (SBI0, SBI1) include following 2 operation modes.

I²C bus mode (Multi master)

Clocked-synchronous 8-bit SIO mode

The serial bus interface is connected to an external device through PC0(SDA0), PC1(SCL0), PC3(SDA1) and PC4(SCL1) in the I²C bus mode; and through PC0(SO0), PC1(SI0), PC2(SCK0), PC3(SO1), PC4(SI1) and PC5(SCK1) in the clocked-synchronous 8-bit SIO mode.

Each of the channels can be operated independently. Since both SBI0 and SBI1 channels operate in the same manner, a channel explains only the case of SBI0.

Each pin is specified as follows: (SBI0)

	PCCR<PC2C, PC1C, PC0C>	PCFC<PC2F, PC1F, PC0F>	PCFC2<PC1F2, PC0F2>
I^2C bus mode	X11	X11	11
Clocked-synchronous 8-bit SIO mode	000(SCK input) 100(SCK output)	111	0n(Note)

X: Don't care

Note) Set PCFC2<PC0F2 > in the clocked-synchronous 8-bit SIO mode to "1" when the oped drain output is necessary.

Each pin is specified as follows: (SBI1)

	PCCR<PC5C, PC4C, PC3C>	PCFC<PC5F, PC4F, PC3F>	PCFC2<PC4F2, PC3F2>
I^2C bus mode	X11	X11	11
Clocked-synchronous 8-bit SIO mode	000(SCK input) 100(SCK output)	111	0n(Note)

X: Don't care

Note) Set PCFC2<PC3F2 > in the clocked-synchronous 8-bit SIO mode to "1" when the oped drain output is necessary.

3.11.1 Configuration

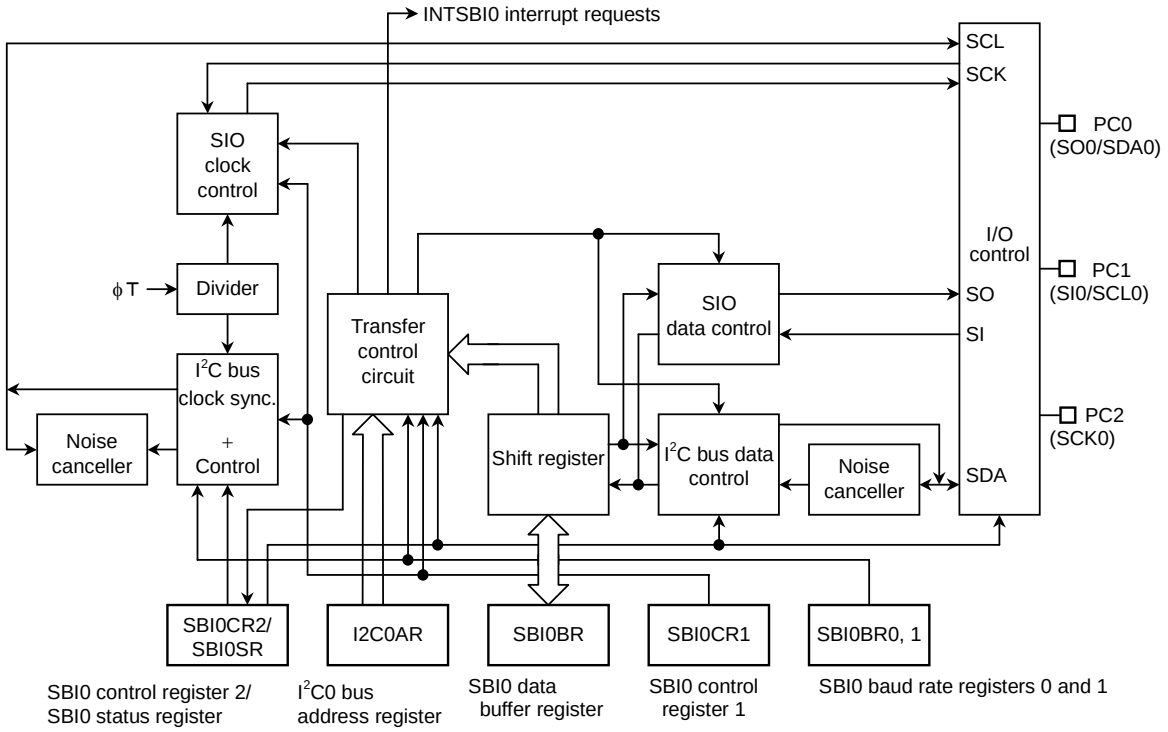


Figure 3.11.1 Serial Bus Interface (SBI0)

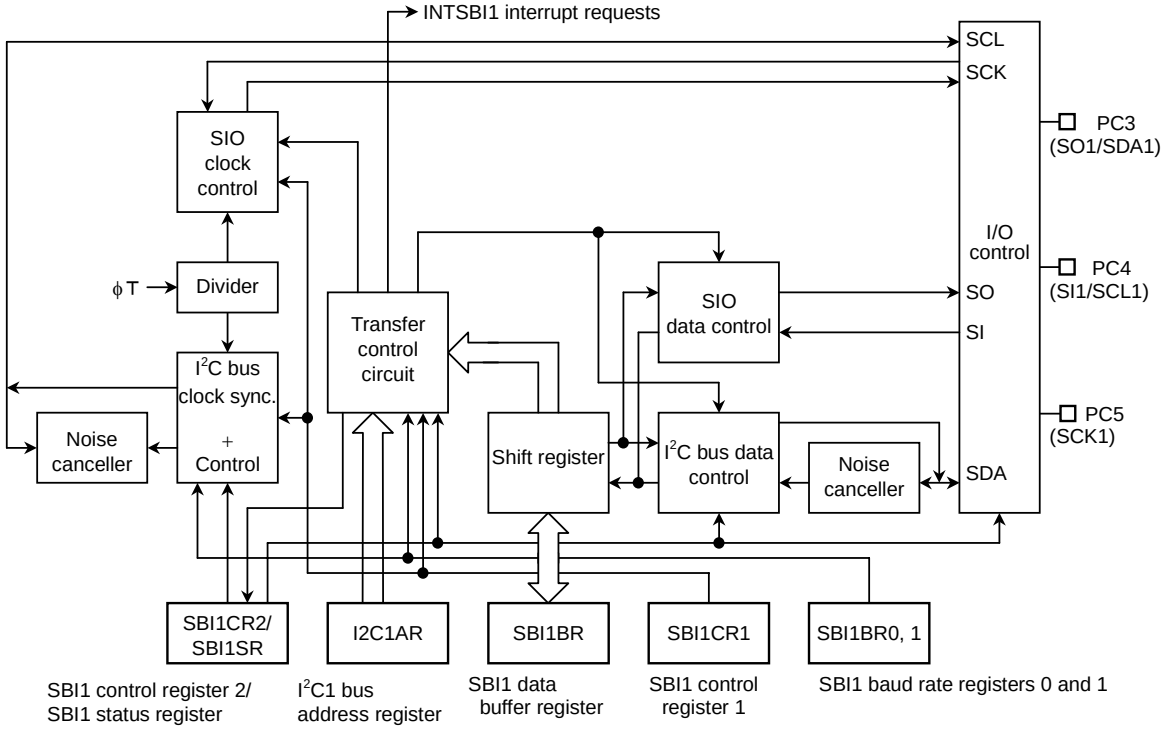


Figure 3.11.2 Serial Bus Interface (SBI1)

3.11.2 Control

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

- Serial bus interface control register 1 (SBI0CR1), (SBI1CR1)
- Serial bus interface control register 2 (SBI0CR2), (SBI1CR2)
- Serial bus interface data buffer register (SBI0DBR), (SBI1DBR)
- I²C bus address register (I2C0AR), (I2C1AR)
- Serial bus interface status register (SBI0SR), (SBI1SR)
- Serial bus interface baud rate register 0 (SBI0BR0), (SBI1BR0)
- Serial bus interface baud rate register 1 (SBI0BR1), (SBI1BR1)

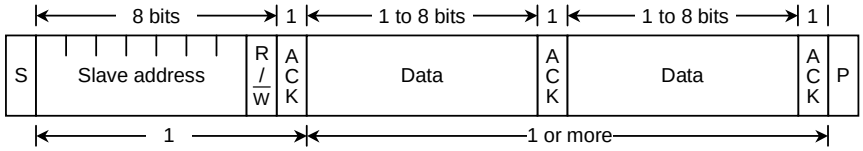
The above registers differ depending on a mode to be used.

Refer to Section 3.11.4 "I²C Bus Mode Control Register" and 3.11.7 "Clocked-synchronous 8-Bit SIO Mode Control".

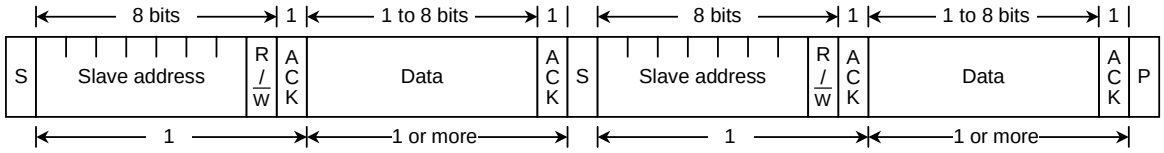
3.11.3 Data Format in I²C Bus Mode

Data format in I²C bus mode is shown Figure 3.11.3

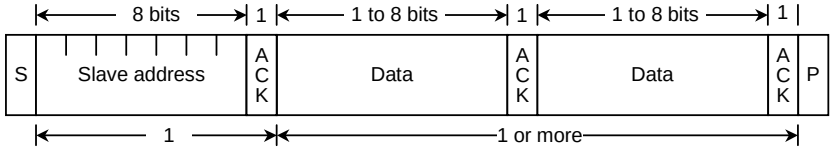
(a) Addressing format



(b) Addressing format (with restart)



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



S: Start condition

R/$\overline{W}$: Direction bit

ACK: Acknowledge bit

P: Stop condition

Figure 3.11.3 Data Format in I²C Bus Mode

3.11.4 I²C Bus Mode Control Register

The following registers are used to control and monitor the operation status when using the serial bus interface (SBI0, SBI1) in the I²C bus mode.

Serial Bus Interface Control Register 1

		7	6	5	4	3	2	1	0
SBI0CR1 (1240H)	Bit symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
	Read/Write	W			R/W		W		R/W
	After reset	0	0	0	0		0	0	0/1 (Note 3)
Read-modify-write instruction is prohibited.	Function	Select 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 <SCK2:0> at write

000	n = 5	– kHz (Note4)
001	n = 6	– kHz (Note4)
010	n = 7	– kHz (Note4)
011	n = 8	60.6 kHz
100	n = 9	30.8 kHz
101	n = 10	15.5 kHz
110	n = 11	7.78 kHz
111	(Reserved)	(Reserved)

System clock: f_c
Clock gear : fc/1
fc = 16 MHz (Output to SCL pin)
Frequency = $\frac{f_c}{2^n + 8}$ [Hz]

Software reset state monitor <SWRMON> at read

0	During software reset
1	Initial data

Acknowledge mode selection

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

Select number of bits transferred

<BC2:0>	<ACK> = 0		<ACK> = 1	
	Number of clock pulses	Data length	Number of clock pulses	Data length
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 <BC2:0> to "000" before switching to a clocked-synchronous 8-bit SIO mode.

Note 2: For the frequency of the SCL line clock, see section 3.11.5 (3) "Serial clock".

Note 3: Initial data of SCK0 is "0", SWRMON is "1".

Note 4: This I2C bus circuit does not support fast mode, it supports standard mode only. The fscl speed can be selected over 100kbps by fc and <SCK2:0>, however it's irregular operation.

Figure 3.11.4 Register for I²C Bus Mode (SBI0, SBI0CR1)

Serial Bus Interface Control Register 1

		7	6	5	4	3	2	1	0
SBI1CR1 (1248H)	Bit symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
	Read/Write	W			R/W		W		R/W
	After reset	0	0	0	0		0	0	0/1 (Note 3)
Read-modify-write instruction is prohibited.	Function	Select 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 <SCK2:0> at write

000	n = 5	– kHz (Note4)
001	n = 6	– kHz (Note4)
010	n = 7	– kHz (Note4)
011	n = 8	60.6 kHz
100	n = 9	30.8 kHz
101	n = 10	15.5 kHz
110	n = 11	7.78 kHz
111	(Reserved)	(Reserved)

System clock: f_c
Clock gear : fc/1
fc = 16 MHz (Output to SCL pin)
Frequency = $\frac{f_c}{2^n + 8}$ [Hz]

Software reset state monitor <SWRMON> at read

0	During software reset
1	Initial data

Acknowledge mode selection

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

Select number of bits transferred

<BC2:0>	<ACK> = 0		<ACK> = 1	
	Number of clock pulses	Data length	Number of clock pulses	Data length
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 <BC2:0> to “000” before switching to a clocked-synchronous 8-bit SIO mode.

Note 2: For the frequency of the SCL line clock, see section 3.11.5 (3) “Serial clock”.

Note 3: Initial data of SCK0 is “0”, SWRMON is “1”.

Note 4: This I2C bus circuit does not support fast mode, it supports standard mode only. The fscl speed can be selected over 100kbps by fc and <SCK2:0>, however it’s irregular operation.

Figure 3.11.5 Register for I^2C Bus Mode (SBI1, SBI1CR1)

Serial Bus Interface Control Register 2

SBI0CR2 (1243H)

Read-modify-write instruction is prohibited.

	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 condition generation	Release INTSBI0 interrupt request	Serial bus interface operation mode selection (Note 2) 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved)		Software reset control write “10” and “01” in order, then an internal software 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	I²C bus mode
11	(Reserved)

INTSBI0 interrupt request

0	–
1	Release interrupt request

Start/stop condition 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

Note 1: Reading this register function as SBI0SR register.

Note 2: Switch a mode to port mode after confirming that the bus is free.
Switch a mode between I²C bus mode and clocked-synchronous 8-bit SIO mode after confirming that input signals via port are high level.

Figure 3.11.6 Register for I²C Bus Mode (SBI0, SBI0CR2)

Serial Bus Interface Control Register 2

SBI1CR2 (124BH)

Read-modify-write instruction is prohibited.

	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 condition generation	Release INTSBI1 interrupt request	Serial bus interface operation mode selection (Note 2) 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved)		Software reset control write “10” and “01” in order, then an internal software 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	I²C bus mode
11	(Reserved)

INTSBI1 interrupt request

0	–
1	Release interrupt request

Start/stop condition 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

Note 1: Reading this register function as SBI1SR register.

Note 2: Switch a mode to port mode after confirming that the bus is free.
Switch a mode between I²C bus mode and clocked-synchronous 8-bit SIO mode after confirming that input signals via port are high level.

Figure 3.11.7 Register for I²C Bus Mode (SBI1, SBI1CR2)

Serial Bus Interface Status Register

SBI0SR (1243H)

Read-modify-write instruction is prohibited.

	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 selection monitor	Transmitter/ receiver status selection monitor	I^2C bus status monitor	INTSBI0 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

INTSBI0 interrupt request monitor

0	Interrupt requested
1	Interrupt released

I^2C bus status monitor

0	Free
1	Busy

Transmitter/receiver status monitor

0	Receiver
1	Transmitter

Master/slave status monitor

0	Slave
1	Master

Note: Writing in this register functions as SBI0CR2.

Figure 3.11.8 Register for I^2C Bus Mode (SBI0, SBI0SR)

Serial Bus Interface Status Register

SBI1SR (124BH)

Read-modify-write instruction is prohibited.

	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 selection monitor	Transmitter/ receiver status selection monitor	I^2C bus status monitor	INTSBI1 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

INTSBI1 interrupt request monitor

0	Interrupt requested
1	Interrupt released

I^2C bus status monitor

0	Free
1	Busy

Transmitter/receiver status monitor

0	Receiver
1	Transmitter

Master/slave status monitor

0	Slave
1	Master

Note: Writing in this register functions as SBI1CR2.

Figure 3.11.9 Register for I^2C Bus Mode (SBI1, SBI1SR)

Serial Bus Interface Baud Rate Register 0

SBI0BR0 (1244H)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	–	I2SBI0						
Read/Write	W	R/W						
After reset	0	0						
Function	Always write "0".	IDLE2 0: Stop 1: Run						

Operation during IDLE 2 mode

0	Stop
1	Run

Serial Bus Interface Baud Rate Register 1

SBI0BR1 (1245H)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Run	Always write "0".						

Internal baud rate circuit control

0	Stop
1	Run

Sirial Bus Interface Data Buffer Register

SBI0DBR (1241H)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receiving)/W (Transmission)							
After reset	Undefined							

Note 1: When writing transmission data, start from the MSB (Bit7). Receiving data is placed from LSB (Bit0).

Note 2: SBI0DBR can't be read the written data. Therefore read-modify-write instruction (e.g., "BIT" instruction) is prohibited.

Note 3: Written data in SBI0DBR is cleared by INTSBI0 signal.

I^2C Bus Address Register

I2C0AR (1242H)

Read-modify-write instruction is prohibited.

	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							Address recognition mode specification

Address recognition mode specification

0	Slave address recognition
1	Non slave address recognition

Figure 3.11.10 Register for I^2C Bus Mode (SBI0, SBI0BR0, SBI0BR1, SBI0DBR, I2C0AR)

Serial Bus Interface Baud Rate Register 0

SBI1BR0 (124CH)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	–	I2SBI0						
Read/Write	W	R/W						
After reset	0	0						
Function	Always write "0".	IDLE2 0: Stop 1: Run						

Operation during IDLE 2 mode

0	Stop
1	Run

Serial Bus Interface Baud Rate Register 1

SBI1BR1 (124DH)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Run	Always write "0".						

Internal baud rate circuit control

0	Stop
1	Run

Sirial Bus Interface Data Buffer Register

SBI1DBR (1249H)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receiving)/W (Transmission)							
After reset	Undefined							

Note 1: When writing transmission data, start from the MSB (Bit7). Receiving data is placed from LSB (Bit0).

Note 2: SBI1DBR can't be read the written data. Therefore read-modify-write instruction (e.g., "BIT" instruction) is prohibited.

Note 3: Written data in SBI1DBR is cleared by INTSBI1 signal.

I^2C Bus Address Register

I2C1AR (124AH)

Read-modify-write instruction is prohibited.

	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							Address recognition mode specification

Address recognition mode specification

0	Slave address recognition
1	Non slave address recognition

Figure 3.11.11 Register for I^2C Bus Mode (SBI1, SBI1BR0, SBI1BR1, SBI1DBR, I2C1AR)

3.11.5 Control in I^2C Bus Mode

(1) Acknowledge mode specification

Set the SBI0CR1<ACK> to 1 for operation in the acknowledge mode. The TMP92CM27 generates an additional clock pulse for an acknowledge signal when operating in master mode. In the transmitter mode during the clock pulse cycle, the SDA0 pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA0 pin is set to the low in order to generate the acknowledge signal.

Clear the <ACK> to 0 for operation in the non-acknowledge mode, the TMP92CM27 does not generate a clock pulse for the acknowledge signal when operating in the master mode.

(2) Select number of transfer bits

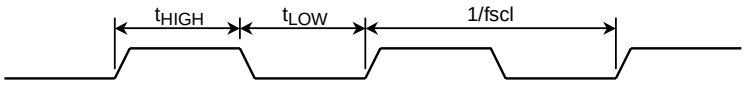
The SBI0CR1<BC2:0> is used to select a number of bits for next transmission/ receiving data.

Since the <BC2:0> is cleared to 000 as a start condition, a slave address and direction bit are transferred in 8 bits. Other than these, the <BC2:0> retains a specified value.

(3) Serial clock

1. Clock source

The SBI0CR1<SCK2:0> is used to select a maximum transfer frequency outputted on the SCL pin in master mode. Set the baud rates, which have been calculated according to the formula below, to meet the specifications of the I^2C bus, such as the smallest pulse width of t_{LOW}.



$t_{LOW} = 2^{n-1}/f_{SBI}$

$t_{HIGH} = 2^{n-1}/f_{SBI} + 8/f_{SBI}$

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

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

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

Note1: f_{SBI} shows f_{SYS}.

Note2: In a setup of prescaler of SYSCR0, the fc/16 mode cannot be used at the time of SBI circuit use.

Figure 3.11.12 Clock Source

2. Clock synchronization

In the I²C bus mode, in order to wired-AND a bus, a master device which pulls down a clock line to 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.

The TMP92CM27 has a clock synchronization function for normal data transfer even when more than one master exists on the bus.

The example explains the clock synchronization procedures when two masters simultaneously exist on a bus.

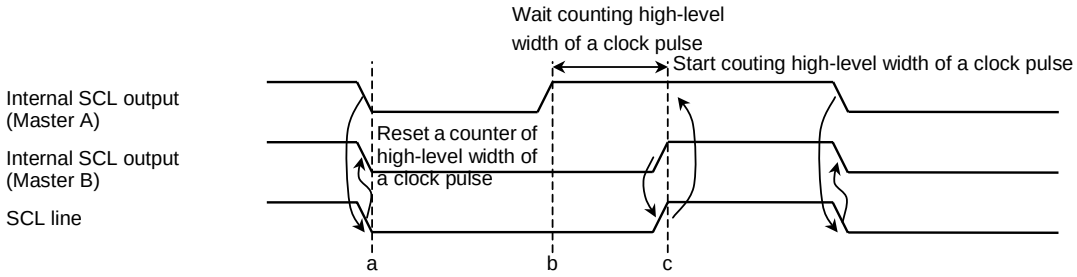


Figure 3.11.13 Clock Synchronization

As master A pulls down the internal SCL output to the low level at point "a", the SCL line of the bus becomes the low level. After detecting this situation, master B resets a counter of high-level width of an own clock pulse and sets the internal SCL output to the low level.

Master A finishes counting low-level width of an own clock pulse at point "b" and sets the internal SCL output to the high level. Since master B holds the SCL line of the bus at the low level, master A waits for counting high-level width of an own clock pulse. After master B finishes counting low-level width of an own clock pulse at point "c" and master A detects the SCL line of the bus at the high level, and starts counting high level of an own clock pulse. The clock pulse on the bus is 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 the TMP92CM27 is used as a slave device, set the slave address <SA6:0> and <ALS> to the I2C0AR. Clear the <ALS> to "0" for the address recognition mode.

(5) Master/slave selection

Set the SBI0CR2<MST> to "1" for operating the TMP92CM27 as a master device. Clear the SBI0CR2<MST> to "0" for operation as a slave device. The <MST> is cleared to "0" by the hardware after a stop condition on the bus is detected or arbitration is lost.

(6) Transmitter/receiver selection

Set the SBI0CR2<TRX> to "1" for operating the TMP92CM27 as a transmitter. Clear the <TRX> to "0" for operation as a receiver. In slave mode, when transfer data in addressing format, when received slave address is same value with setting value to I2C0AR, or GENERAL CALL is received (All 8-bit data are "0" after a start condition), the <TRX> is set to "1" by the hardware if the direction bit (R/$\overline{W}$) sent from the master device is "1", and <TRX> is cleared to "0" by the hardware if the bit is "0".

In the master mode, after an acknowledge signal is returned from the slave device, the <TRX> is cleared to "0" by the hardware if a transmitted direction bit is "1", and is set to "1" by the hardware if it is "0". When an acknowledge signal is not returned, the current condition is maintained.

The <TRX> is cleared to "0" by the hardware after a stop condition on the bus is detected or arbitration is lost.

(7) Start/stop condition generation

When programmed "1111" to SBI0CR2 <MST, TRX, BB, PIN> in during SBI0SR<BB> is "0", slave address and direction bit which are set to SBI0DBR and start condition are output on a bus. And it is necessary to set transmitted data to the data buffer register (SBI0DBR) and set "1" to <ACK> beforehand.

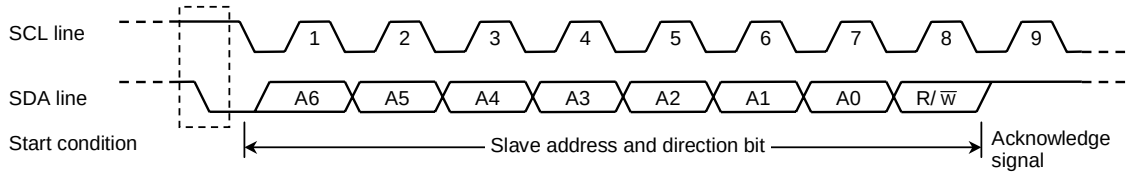


Figure 3.11.14 Generation of Start Condition and Slave Address

When programmed "0" to SBI0CR2<BB> and "111" to <MST, TRX, PIN> in during SBI0SR<BB> is "1", start a sequence of stop condition output. Do not modify the contents of <MST, TRX, BB, and PIN> until a stop condition is generated on a bus.

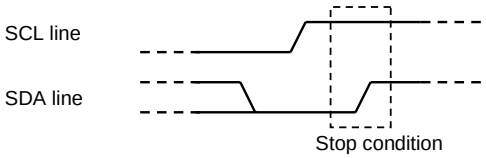


Figure 3.11.15 Generation of Stop Condition

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

In addition, since there is a restrictions matter about stop condition generating in master mode, please refer to 3.11.6.(4) " Stop condition generation ".

(8) Interrupt service requests and interrupt cancellation

When a serial bus interface interrupt request 0 (INTSBI0) occurs, the SBI0SR2 <PIN> is cleared to "0". During the time that the SBI0SR2<PIN> is "0", the SCL line is pulled down to the low level.

The <PIN> is cleared to "0" when end of transmission or receiving 1 word of data. And when writing data to SBI0DBR or reading data from SBI0DBR, <PIN> is set to "1".

The time from the <PIN> being set to "1" until the SCL line is released takes t_{LOW}.

In the address recognition mode (<ALS> = 0), <PIN> is cleared to "0" when the received slave address is the same as the value set at the I2C0AR or when a GENERAL CALL is received (All 8-bit data are "0" after a start condition). Although SBI0CR2<PIN> can be set to "1" by the program, the <PIN> is not clear it to "0" when it is programmed "0".

(9) Serial bus interface operation mode selection

SBI0CR2<SBIM1:0> is used to specify the serial bus interface operation mode. Set SBI0CR2<SBIM1:0> to "10" when the device is to be used in I²C bus mode after confirming pin condition of serial bus interface to "H".

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

(10) Arbitration lost detection monitor

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

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

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously 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 SDA line of the bus is wire-AND and the SDA line is pulled down to the low level by master A. When the SCL line of the bus is pulled up at point b, the slave device reads the data on the SDA line, that is, data in master A. A data transmitted from master B becomes invalid. The state in master B is called "ARBITRATION LOST". Master B device that loses arbitration releases the internal SDA 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.

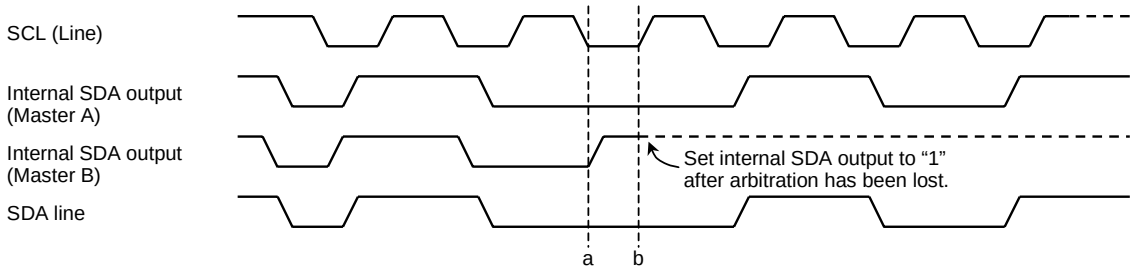


Figure 3.11.16 Arbitration Lost

The TMP92CM27 compares the levels on the bus's SDA line with those of the internal SDA output on the rising edge of the SCL line. If the levels do not match, arbitration is lost and SBI0SR<AL> is set to "1".

When SBI0SR<AL> is set to "1", SBI0SR<MST, TRX> are cleared to "00" and the mode is switched to slave receiver mode. Thus, clock output is stopped in data transfer after setting <AL> = "1".

SBI0SR <AL> is cleared to "0" when data is written to or read from SBI0DBR or when data is written to SBI0CR2.

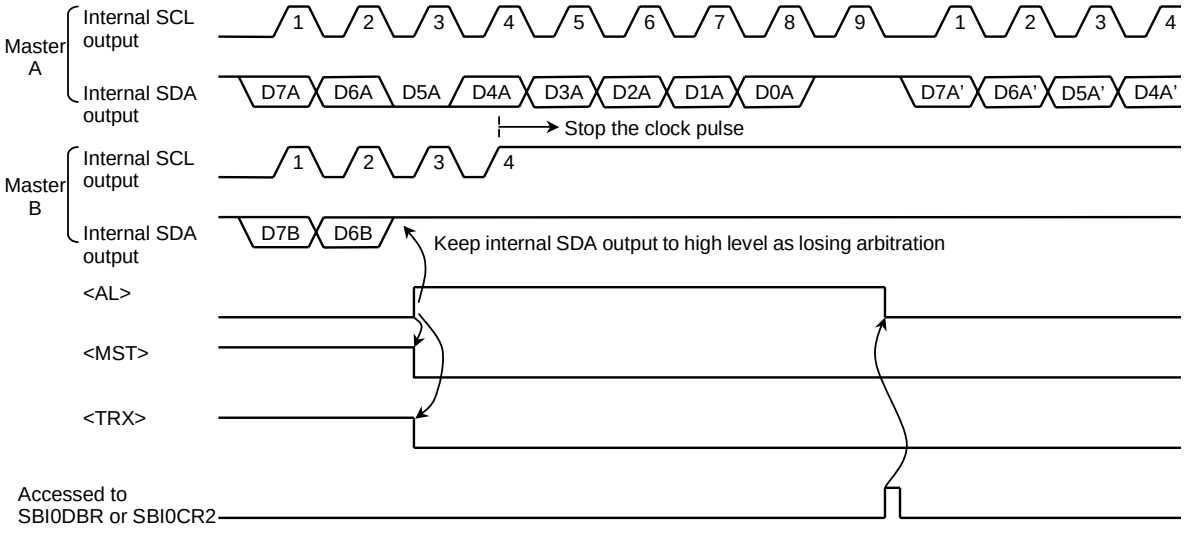


Figure 3.11.17 Example of when TMP92CM27 is a Master Device B (D7A = D7B, D6A = D6B)

(11) Slave address match detection monitor

SBI0SR<AAS> operates following in during slave mode; In address recognition mode (e.g., when I2C0AR<ALS> = "0"), when received GENERAL CALL or same slave address with value set to I2C0AR, SBI0SR<AAS> is set to "1". When <ALS> = "1", SBI0SR<AAS> is set to "1" after the first word of data has been received. SBI0SR<AAS> is cleared to "0" when data is written to SBI0DBR or read from SBI0DBR.

(12) GENERAL CALL detection monitor

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

(13) Last received bit monitor

The value on the SDA line detected on the rising edge of the SCL line is stored in the SBI0SR<LRB>. In the acknowledge mode, immediately after an INTSBI0 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.

When write first "10" next "01" to SBI0CR2<SWRST1:0>, reset signal is inputted to serial bus interface circuit, and circuit is initialized. All command registers except SBI0CR2<SBIM1:0> and status flag except SBI0CR2<SBIM1:0> are initialized to value of just after reset. SBI0CR1<SWRMON> is set to "1" automatically when completed initialization of serial bus interface.

(15) Serial bus interface data buffer register (SBI0DBR)

The received data can be read and transmission data can be written by reading or writing SBI0DBR.

In the master mode, after the slave address and the direction bit are set in this register, the start condition is generated.

(16) I^2C bus address register (I2C0AR)

I2C0AR<SA6:0> is used to set the slave address when the TMP92CM27 functions as a slave device.

The slave address outputted from the master device is recognized by setting the I2C0AR<ALS> to "0". And, the data format becomes the addressing format. When set <ALS> to "1", the slave address is not recognized, the data format becomes the free data format.

(17) Baud rate register (SBI0BR1)

Write "1" to baud rate circuit control register SBI0BR1<P4EN> before using I^2C bus.

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

SBI0BR0<I2SBI0> is the register setting operation/stop during IDLE2 mode. Therefore, setting <I2SBI0> is necessary before the HALT instruction is executed.

3.11.6 Data Transfer in I^2C Bus Mode

(1) Device initialization

In first, set the SBI0BR1<P4EN>, SBI0CR1<ACK, SCK2:0>. Set SBI0BR1<P4EN> to "1" and clear bits 7 to 5 and 3 in the SBI0CR1 to "0".

Next, set a slave address <SA6:0> and the <ALS> (<ALS> = "0" when an addressing format) to the I2C0AR.

And, write "000" to SBI0CR2<MST, TRX, BB>, "1" to <PIN>, "10" to <SBIM1:0> and "00" to <SWRST1:0>. Set initialization status to slave receiver mode by this setting.

(2) Start condition and slave address generation

1. Master mode

In the master mode, the start condition and the slave address are generated as follows.

In first, check a bus free status (when SBI0SR<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 SBI0SR<BB> = "0", the start condition are generated by writing "1111" to SBI0CR2<MST, TRX, BB, PIN>. Subsequently to the start condition, nine clocks are output from the SCL pin. While eight clocks are output, the slave address and the direction bit which are set to the SBI0DBR. At the 9th clock, the SDA line is released and the acknowledge signal is received from the slave device.

An INTSBI interrupt request generate at the falling edge of the 9th clock. The <PIN> is cleared to "0". In the master mode, the SCL pin is pulled down to the low level while <PIN> is "0". When an interrupt request is generated, the <TRX> is changed according to the direction bit only when an acknowledge signal is returned from the slave device.

2. Slave mode

In the slave mode, the start condition and the slave address are received.

After the start condition is received from the master device, while eight clocks are output from the SCL pin, the slave address and the direction bit that are output from the master device are received.

When a GENERAL CALL or the same address as the slave address set in I2C0AR is received, the SDA line is pulled down to the low level at the 9th clock, and the acknowledge signal is output.

An INTSBI0 interrupt request is generated on the falling edge of the 9th clock. The <PIN> is cleared to "0". In slave mode the SCL line is pulled down to the low level while the <PIN> = "0".

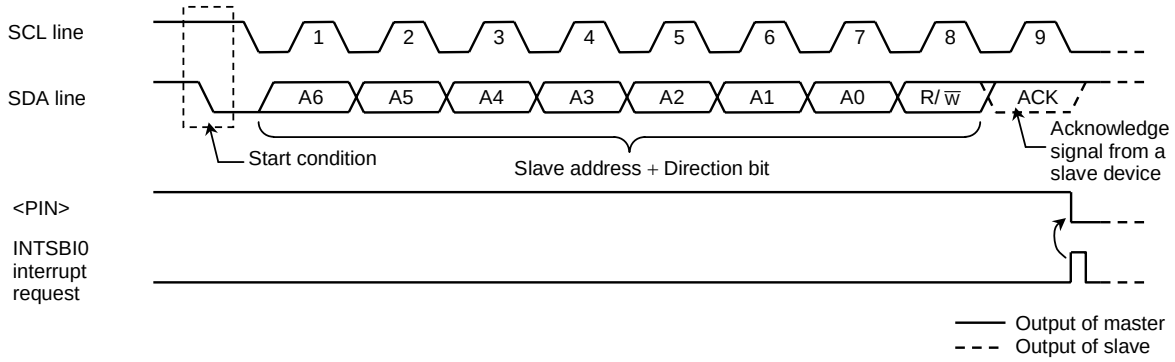


Figure 3.11.18 Start Condition and Slave Address Generation

(3) 1-word data transfer

Check the <MST> by the INTSBI0 interrupt process after the 1-word data transfer is completed, and determine whether the mode is a master or slave.

1. If <MST> = "1" (Master mode)

Check the <TRX> and determine whether the mode is a transmitter or receiver.

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

Check the <LRB>. When <LRB> is "1", a receiver does not request data. Implement the process to generate a stop condition (Refer to 3.11.6 (4)) and terminate data transfer.

When the <LRB> is "0", the receiver is requests new data. When the next transmitted data is 8 bits, write the transmitted data to SBI0DBR. When the next transmitted data is other than 8 bits, set the <BC2:0> <ACK> and write the transmitted data to SBI0DBR. After written the data, <PIN> becomes "1", a serial clock pulse is generated for transferring a new 1-word of data from the SCL0 pin, and then the 1-word data is transmitted. After the data is transmitted, an INTSBI0 interrupt request generates. The <PIN> becomes "0" and the SCL0 line is pulled down to the low level. If the data to be transferred is more than one word in length, repeat the procedure from the <LRB> checking above.

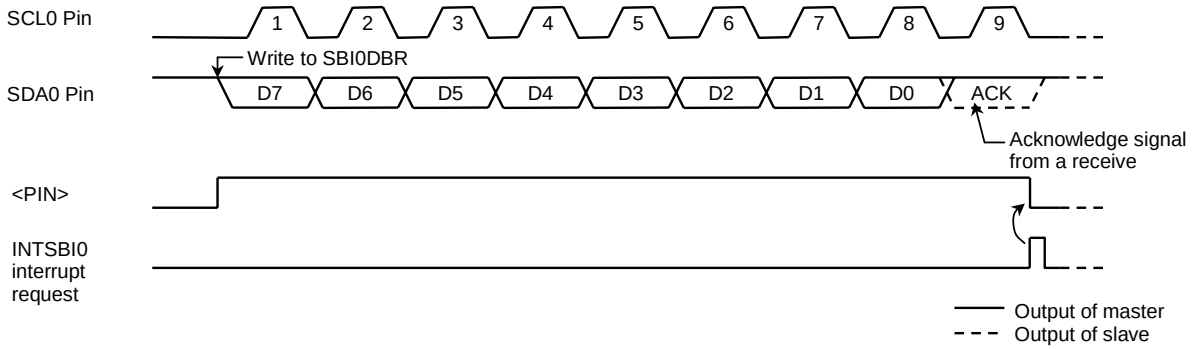


Figure 3.11.19 Example in which <BC2:0> = "000" and <ACK> = "1" (Transmitter mode)

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

When the next transmitted data is other than 8 bits, set <BC2:0> <ACK> and read the received data from SBI0DBR to release the SCL0 line (Data which is read immediately after a slave address is sent is undefined). After the data is read, <PIN> becomes "1". Serial clock pulse for transferring new 1 word of data is defined SCL and outputs "L" level from SDA0 pin with acknowledge timing.

An INTSBI0 interrupt request then generates and the <PIN> becomes "0", Then the TMP92CM27 pulls down the SCL pin to the low level. The TMP92CM27 outputs a clock pulse for 1 word of data transfer and the acknowledge signal each time that received data is read from the SBI0DBR.

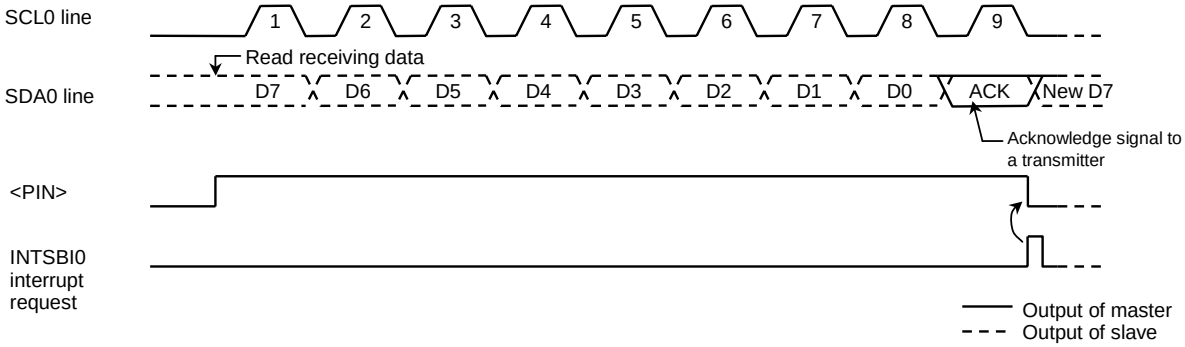


Figure 3.11.20 Example of when <BC2:0> = "000", <ACK> = "1" (Receiver mode)

In order to terminate the transmission of data to a transmitter, clear <ACK> to "0" before reading data which is 1 word before the last data to be received. The last data word does not generate a clock pulse as the acknowledge signal. After the data has been transmitted and an interrupt request has been generated, set <BC2:0> to "001" and read the data. The TMP92CM27 generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA0 line on the bus remains high. The transmitter receives the high signal as an ACK signal. The receiver indicates to the transmitter that the data transfer is completed.

After the one data bit has been received and an interrupt request has been generated, the TMP92CM27 generates a stop condition (See section 3.11.6 (4)) and terminates data transfer.

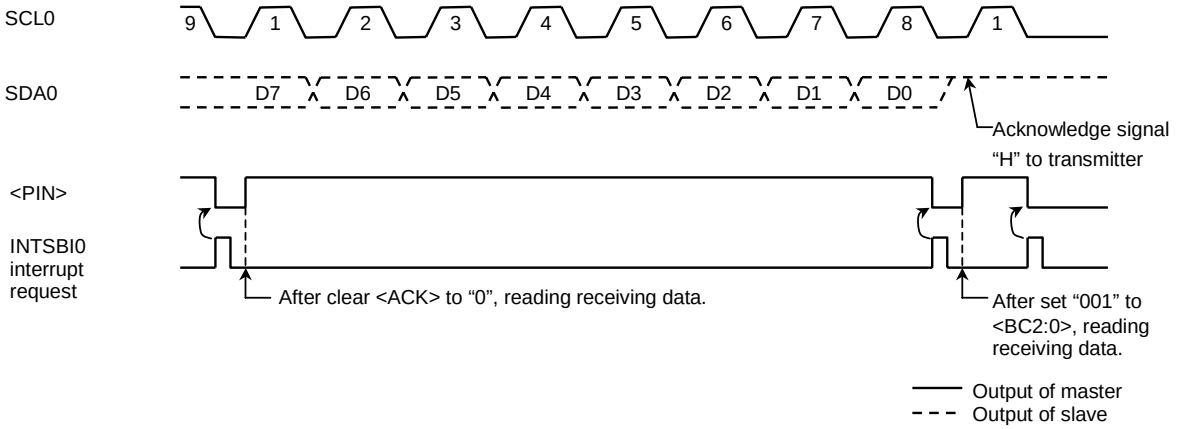


Figure 3.11.21 Termination of Data Transfer (Master receiver mode)

2. If <MST> = 0 (Slave mode)

In the slave mode the TMP92CM27 operates either in normal slave mode or in slave mode after losing arbitration.

In the slave mode, an INTSBI0 interrupt request generate when the TMP92CM27 receives a slave address or a GENERAL CALL from the master device, or when a GENERAL CALL is received and data transfer is completed, or after matching received address. In the master mode, the TMP92CM27 operates in a slave mode if it losing arbitration. An INTSBI0 interrupt request is generated when a word data transfer terminates after losing arbitration. When an INTSBI0 interrupt request is generated the <PIN> is cleared to "0" and the SCL pin is pulled down to the low level. Either reading/writing from/to the SBI0DBR or setting the <PIN> to "1" will release the SCL pin after taking t_{LOW} time.

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

Table 3.11.1 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	The TMP92CM27 detects arbitration lost when transmitting a slave address, and receives a slave address for which the value of the direction bit sent from another master is "1".	Set the number of bits of single word to <BC2:0>, and write the transmit data to SBI0DBR.
	0	1	0	In salve receiver mode, the TMP92CM27 receives a slave address for which the value of the direction bit sent from the master is "1".	
		0	0	In salve transmitter mode, transmission of data of single word is terminated.	Check the <LRB>, If <LRB> is set to "1", set <PIN> to "1", reset "0" to <TRX> and release the bus for the receiver no request next data. If <LRB> was cleared to "0", set bit number of single word to <BC2:0> and write the transmit data to SBI0DBR for the receiver requests next data.
0	1	1	1/0	The TMP92CM27 detects arbitration lost when transmitting a slave address, and receives a slave address or GENERAL CALL for 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	The TMP92CM27 detects arbitration lost when transmitting a slave address or data, and transfer of word terminates.	
	0	1	1/0	In slave receiver mode the TMP92CM27 receives a slave address or GENERAL CALL for which the value of the direction bit sent from the master is "0".	
		0	1/0	In slave receiver mode the TMP92CM27 terminates receiving word data.	Set bit number of single word to <BC2:0>, and read the receiving data from SBI0DBR.

(4) Stop condition generation

When SBI0SR<BB> = 1, the sequence for generating a stop condition is started by writing "111" to SBI0CR2<MST, TRX, PIN> and "0" to SBI0CR2<BB>. Do not modify the contents of SBI0CR2<MST, TRX, PIN, BB> until a stop condition has been generated on the bus. When the bus's SCL line has been pulled low by another device, the TMP92CM27 generates a stop condition when the other device has released the SCL line and SDA0 pin rising.

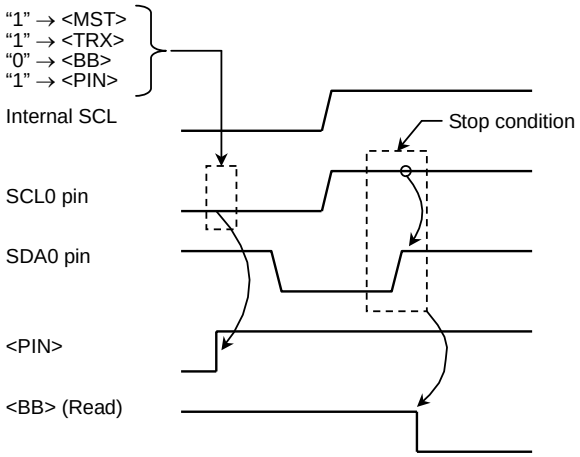


Figure 3.11.22 Stop Condition Generation (Single master)

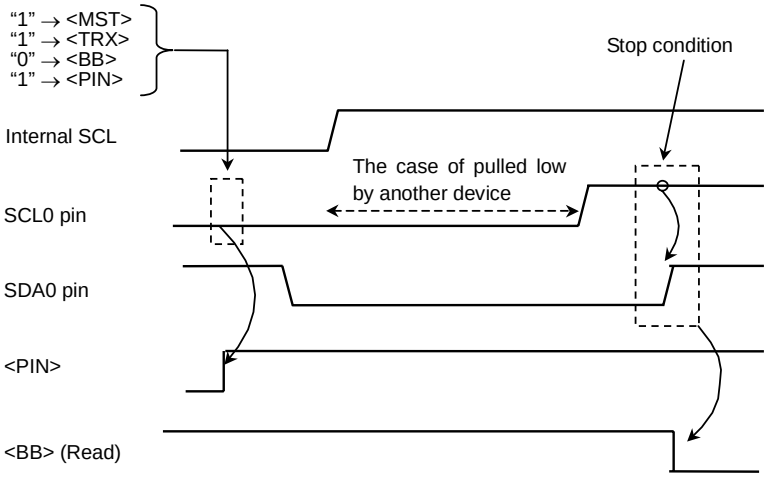


Figure 3.11.23 Stop Condition Generation (Multi master)

(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 SCL 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 3.11.6 (2).

In order to meet setup time when restarting, take at least 4.7 μs 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.

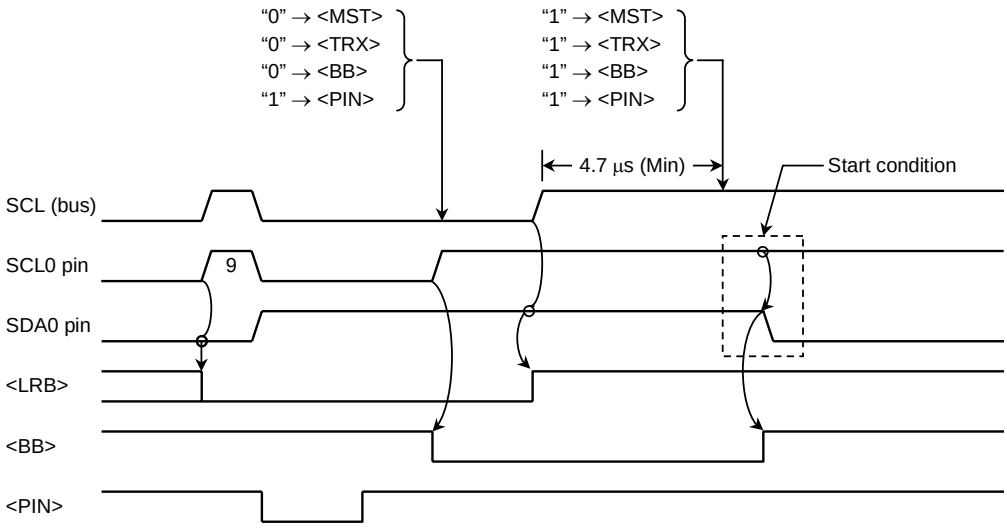


Figure 3.11.24 Timing Diagram when Restarting

3.11.7 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 (1240H)

Read-modify-write instruction is prohibited.

	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		0	0	0
Function	Transfer start 0: Stop 1: Start	Continue/ abort transfer 0: Continue transfer 1: Abort transfer	Transfer mode select 00: Transmit mode 01: (Reserved) 10: Transmit/receive mode 11: Receive mode			Serial clock selection and reset monitor		

Serial clock selection <SCK2:0> at write

000	n = 4	1 MHz
001	n = 5	500 kHz
010	n = 6	250 kHz
011	n = 7	125 kHz
100	n = 8	62.5 kHz
101	n = 9	31.25 kHz
110	n = 10	15.625 kHz
111	–	(External clock : SCK0)

System clock: f_c
f_c = 16 MHz (SCL output to SCK pin)
$fscl = \frac{f_c}{2^n}$ [Hz]

Transfer mode selection

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

Continue/abort transfer

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

Indicate transfer start/stop

0	Stop
1	Start

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

Serial Bus Interface 0 Data Buffer Register

SBI0DBR (1241H)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receiver)/W (Transfer)							
After reset	Undefined							

Figure 3.11.25 Register for the SIO Mode (SBI0, SBI0CR1, SBI0DBR)

Serial Bus Interface 1 Control Register 1

SBI1CR1 (1248H)

Read-modify-write instruction is prohibited.

	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		0	0	0
Function	Transfer start 0: Stop 1: Start	Continue/ abort transfer 0: Continue transfer 1: Abort transfer	Transfer mode select 00: Transmit mode 01: (Reserved) 10: Transmit/receive mode 11: Receive mode			Serial clock selection and reset monitor		

Serial clock selection <SCK2:0> at write

000	n = 4	1 MHz
001	n = 5	500 kHz
010	n = 6	250 kHz
011	n = 7	125 kHz
100	n = 8	62.5 kHz
101	n = 9	31.25 kHz
110	n = 10	15.625 kHz
111	–	(External clock : SCK0)

System clock: f_c

f_c = 16MHz (SCL output to SCK pin)

$fscl = \frac{f_c}{2^n}$ [Hz]

Transfer mode selection

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

Continue/abort transfer

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

Indicate transfer start/stop

0	Stop
1	Start

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

Serial Bus Interface 1 Data Buffer Register

SBI1DBR (1249H)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receiver)/W (Transfer)							
After reset	Undefined							

Figure 3.11.26 Register for the SIO Mode(SBI1, SBI1CR1, SBI1DBR)

Serial Bus Interface 0 Control Register 2

SBI0CR2 (1243H)

Read-modify-write instruction is prohibited.

	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)		(Note 2)	(Note 2)

Note 1: Set the SBI0CR1<BC2:0> to "000" before switching to a clocked-synchronous 8-bit SIO mode

Note 2: Please always write "00" to SBI0CR2<1:0>.

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)

Serial Bus Interface 0 Status Register

SBI0SR (1243H)

	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		

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in progress

Shift operation status monitor

0	Shift operation terminated
1	Shift operation in progress

Serial Bus Interface 0 Baud Rate Register 0

SBI0BR0 (1244H)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	–	I2SBI0						
Read/Write	W	R/W						
After reset	0	0						
Function	Always write "0".	IDLE2 0: Stop 1: Operate						

Note: Clocked-syncronous mode cannot operate in IDLE2 mode.

Operation in IDLE2 mode

0	Stop
1	Operate

Serial Bus Interface 0 Baud Rate Register 1

SBI0BR1 (1245H)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Operate	Always write "0".						

Baud rate clock control

0	Stop
1	Operate

Figure 3.11.27 Register for the SIO Mode (SBI0, SBI0CR2, SBI0SR, SBI0BR0, SBI0BR1)

Serial Bus Interface 1 Control Register 2

SBI1CR2 (124BH)

Read-modify-write instruction is prohibited.

	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)		(Note 2)	(Note 2)

Note 1: Set the SBI0CR1<BC2:0> to "000" before switching to a clocked-synchronous 8-bit SIO mode

Note 2: Please always write "00" to SBICR2<1:0>.

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)

Serial Bus Interface 1 Status Register

SBI1SR (124BH)

	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		

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in progress

Shift operation status monitor

0	Shift operation terminated
1	Shift operation in progress

Serial Bus Interface 1 Baud Rate Register 0

SBI1BR0 (124CH)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	–	I2SBI0						
Read/Write	W	R/W						
After reset	0	0						
Function	Always write "0".	IDLE2 0: Stop 1: Operate						

Note: Clocked-syncronous mode cannot operate in IDLE2 mode.

Operation in IDLE2 mode

0	Stop
1	Operate

Serial Bus Interface 1 Baud Rate Register 1

SBI1BR1 (124DH)

Read-modify-write instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Operate	Always write "0".						

Baud rate clock control

0	Stop
1	Operate

Figure 3.11.28 Register for the SIO Mode (SBI1, SBI1CR2, SBI1SR, SBI1BR0, SBI1BR1)

(1) Serial Clock

1. Clock source

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

Internal clock

In internal clock mode one of seven frequencies can be selected. The serial clock signal is output to the outside on the SCK0 pin. The SCK0 pin goes high when data transfer starts. When the device is writing (in transmit mode) or reading (in receive mode), data cannot follow the serial clock rate, so an automatic wait function is executed which automatically stops the serial clock and holds the next shift operation until reading or writing has been completed.

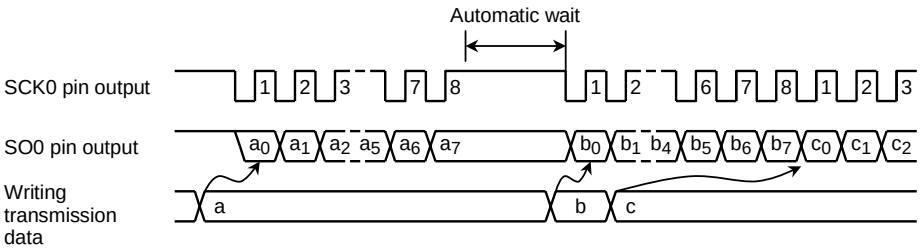


Figure 3.11.29 Automatic Wait Function

External clock (<SCK2:0> = "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 1MHz (when f_C = 16 MHz).

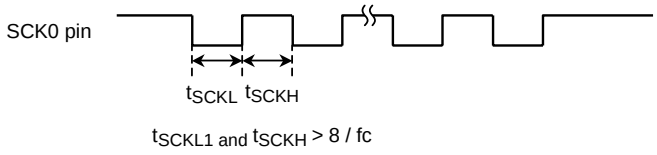


Figure 3.11.30 Maximum Data Transfer Frequency when External Clock Input

2. Shift edge

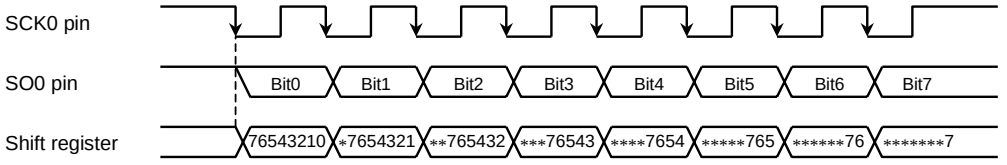
Data is transmitted on the leading edge of the clock and received on the trailing edge.

Leading edge shift

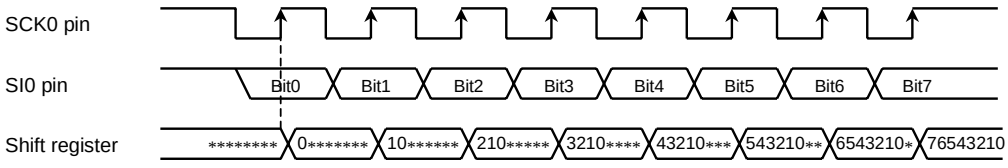
Data is shifted on the leading edge of the serial clock (on the falling edge of the SCK0 pin input/output).

Trailing edge shift

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



(a) Leading shift



(b) Trailing shift

*: Don't care

Figure 3.11.31 Shift Edge

(2) Transfer modes

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

1. 8-bit transmit mode

Set a control register to a transmit mode and write transmission data 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 INTSBI0 (Buffer empty) interrupt request is generated to request new data.

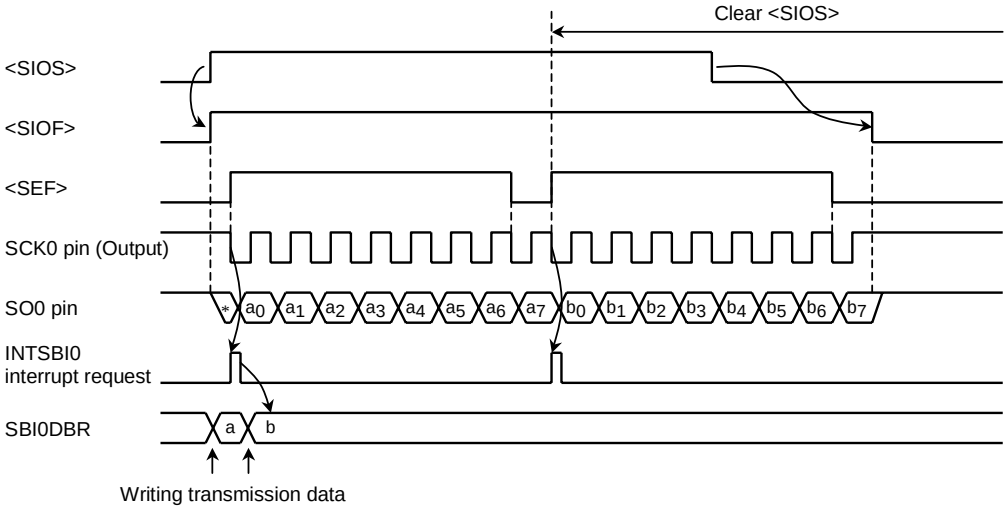
When the internal clock is used, the serial clock will stop and the automatic wait function will be initiated if new data is not loaded to the data buffer register after the specified 8-bit data is transmitted. When new transmission data is written, the automatic wait function is canceled.

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.

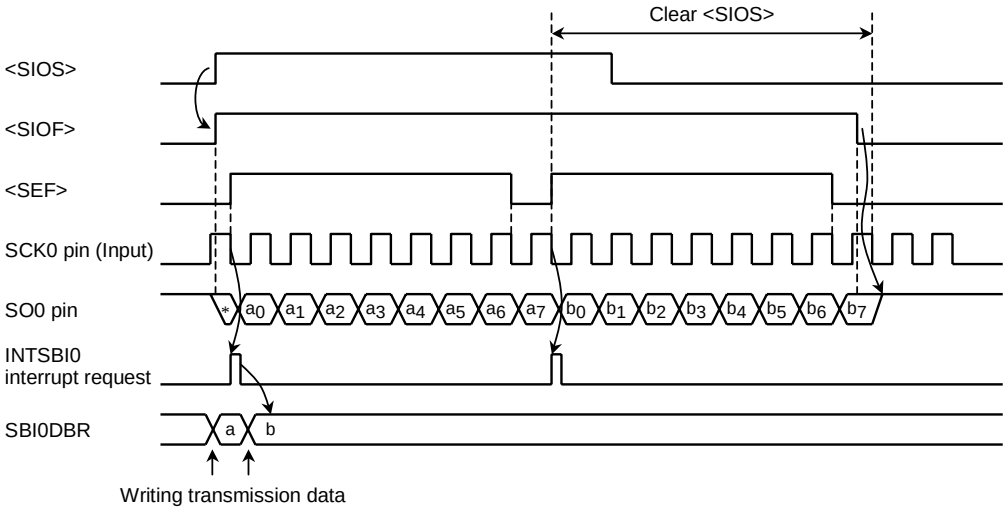
When the transmit is started, after the SBI0SR<SIOF> goes "1" output from the SO0 pin holds final bit of the last data until falling edge of the SCK.

For stopping data transmission, when the <SIOS> is cleared to "0" by the INTSBI0 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> 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.



(a) Internal clock



(b) External clock

Figure 3.11.32 Transmission Mode

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

```
STEST1  :   BIT   SEF, (SBI0SR)          ;  If <SEF> = 1 then loop.
            JR    NZ, STEST1
STEST2  :   BIT   0, (PN)                ;  If SCK = 0 then loop.
            JR    Z, STEST2
            LD    (SBI0CR1), 00000111B   ;  <SIOS> ← 0
```

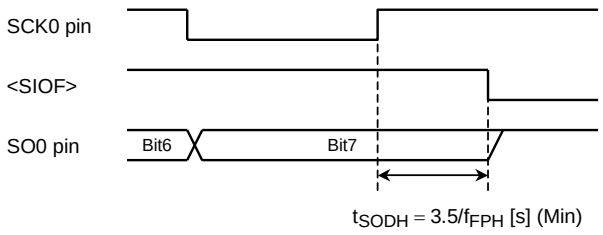


Figure 3.11.33 Transmission Data Hold Time at End Transmit

2. 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 INTSBI0 (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 INTSBI0 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 completed. 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 completed. 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: When the transfer mode is changed, the contents of the SBI0DBR will be lost. If the mode must be changed, conclude data receiving by clearing the <SIOS> to "0", read the last data, then change the mode.

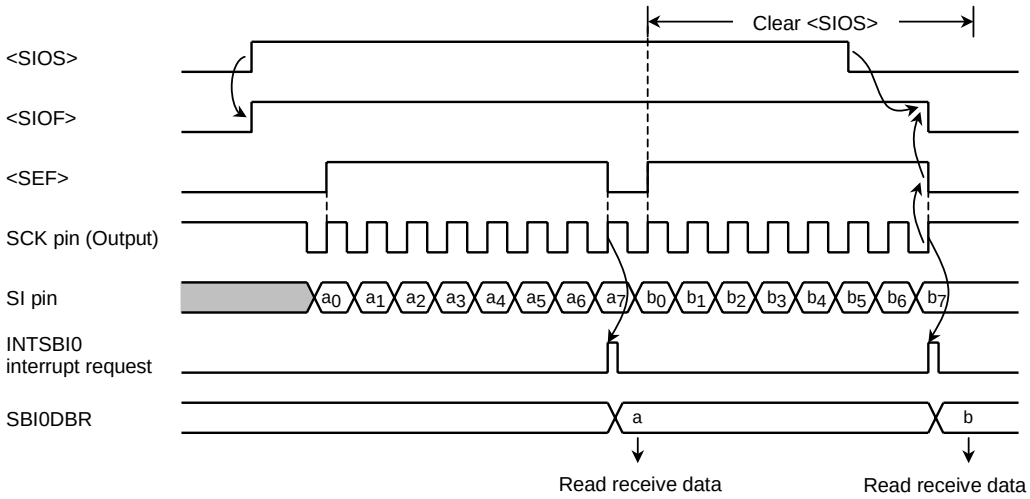


Figure 3.11.34 Receiver Mode (Example: Internal clock)

3. 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 SBI0CR1<SIOS> to "1" to start transmitting/receiving. When data is transmitted, the data is output from the SO0 pin, starting from the least significant bit (LSB) and synchronized with the leading edge of the serial clock signal. When data is received, the data is input via the SI0 pin on the trailing edge of the serial clock signal. 8-bit data is transferred from the shift register to the SBI0DBR and the INTSBI0 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 new 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 SCK.

Transmitting/receiving data ends when the <SIOS> is cleared to "0" by the INTSBI0 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 completed. In order to confirm whether data is being transmitted/received properly by the program, set the SBI0SR to be sensed. The <SIOF> is cleared 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, and then change the transfer mode.

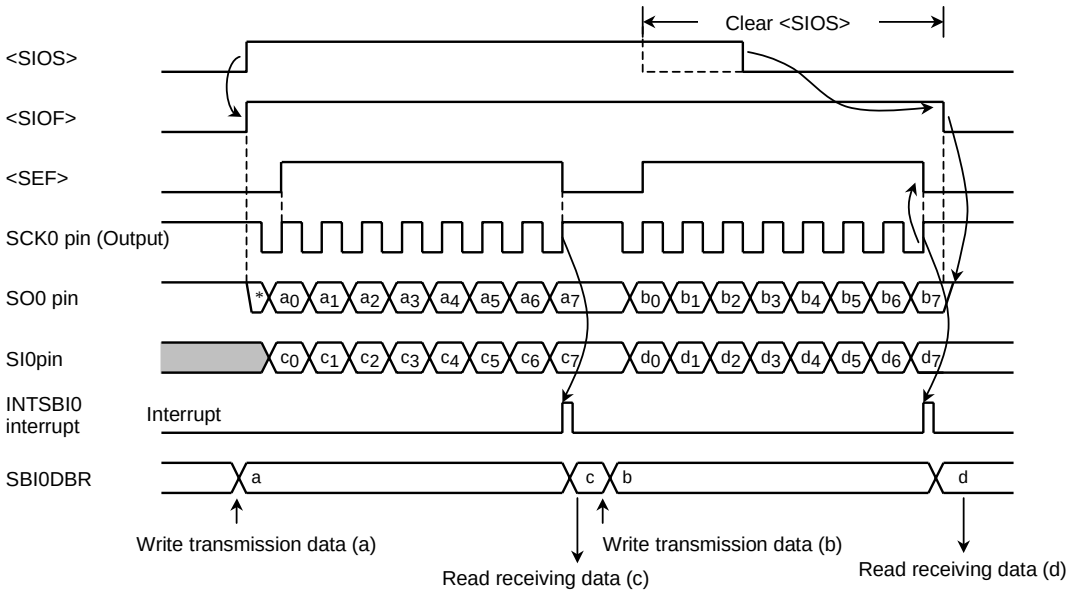


Figure 3.11.35 Transmission/Receiving Mode (when an external clock is used)

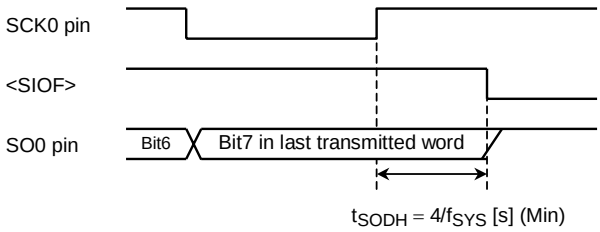


Figure 3.11.36 Transmission Data Hold Time at End of Transmission/Receiving (Transmission/receiving mode)

3.12 High Speed SIO (HSC)

TMP92CM27 includes 2 High Speed SIO channels. Each channel is called HSC0 and HSC1. Each channel supports only the master mode in I/O interface mode (synchronous transmission).

The features as follows.

1) Double buffer (Transmit/Receive)
2) Generate CRC7 and CRC16 of Transmit/Receive data
3) Baud Rate : 10Mbps max
4) MSB/LSB-first
5) 8/16bit data length
6) Clock Rising/Falling edge
7) The interruption function of each 1 channel : INTHSC0/INTHSC1

Read, Mask, Clear interrupt and Clear enable can control each 4 interrupts:

RFR0/1 (Receive buffer of HSC0RD/HSC1RD: Full),

RFW0/1 (Transmission buffer of HSC0TD/HSC1TD: Empty),

REND0/1 (Receive buffer of HSC0RS/HSC1RS: Full),

TEND0/1 (Transmission buffer of HSC0TS/HSC1TS: Empty).

RFR0/1, RFW0/1 can high-speed transaction by micro DMA.

High Speed SIO channels 0 to 1 can be used independently.

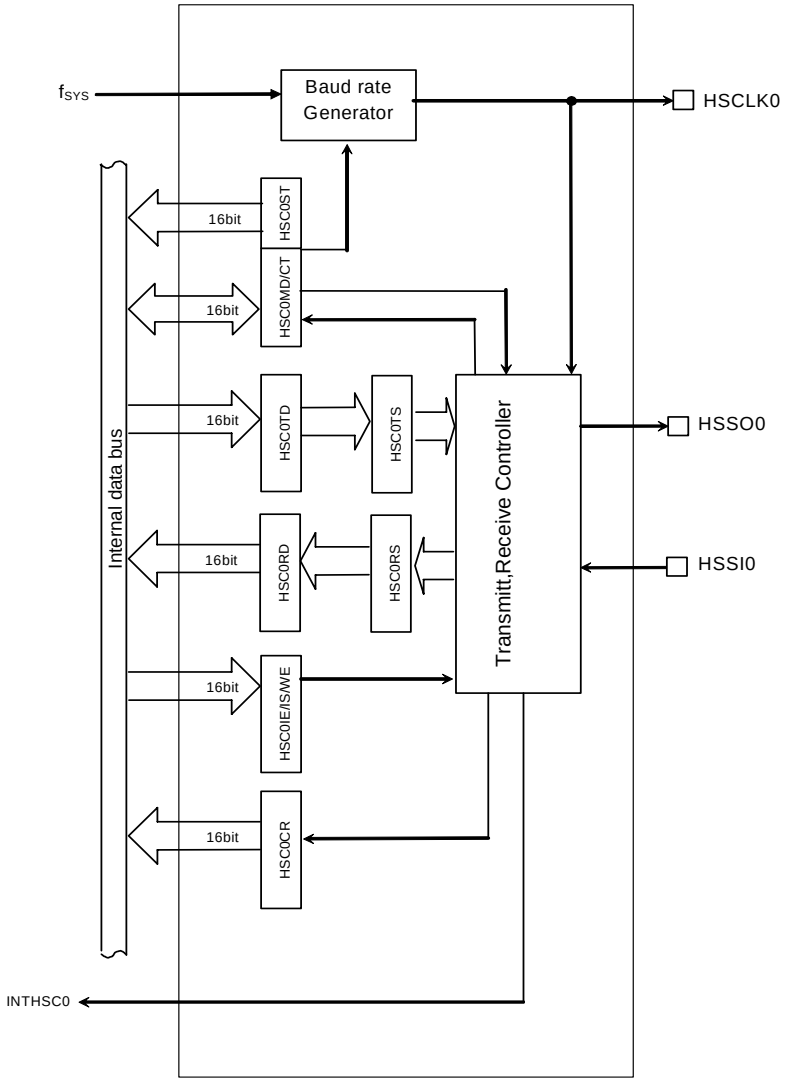
All channels operate in the same function except for the following points; hence only the operation of channel 0 is explained below.

Table 3.12.1 Differences between each Channels

	HSC0	HSC1
Pin name	HSSI0 (PD0) HSSO0 (PD1) HSSCLK0 (PD2)	HSSI1 (PL4) HSSO1 (PL5) HSCLK1 (PL6)
SFR (address)	HSC0MD (C00H/C01H) HSC0CT (C02H/C03H) HSC0ST (C04H/C05H) HSC0CR (C06H/C07H) HSC0IS (C08H/C09H) HSC0WE (C0AH/C0BH) HSC0IE (C0CH/C0DH) HSC0IR (C0EH/C0FH) HSC0TD (C10H/C11H) HSC0RD (C12H/C13H) HSC0TS (C14H/C15H) HSC0RS (C16H/C17H)	HSC1MD (C20H/C21H) HSC1CT (C22H/C23H) HSC1ST (C24H/C25H) HSC1CR (C26H/C27H) HSC1IS (C28H/C29H) HSC1WE (C2AH/C2BH) HSC1IE (C2CH/C2DH) HSC1IR (C2EH/C2FH) HSC1TD (C30H/C31H) HSC1RD (C32H/C33H) HSC1TS (C34H/C35H) HSC1RS (C36H/C37H)

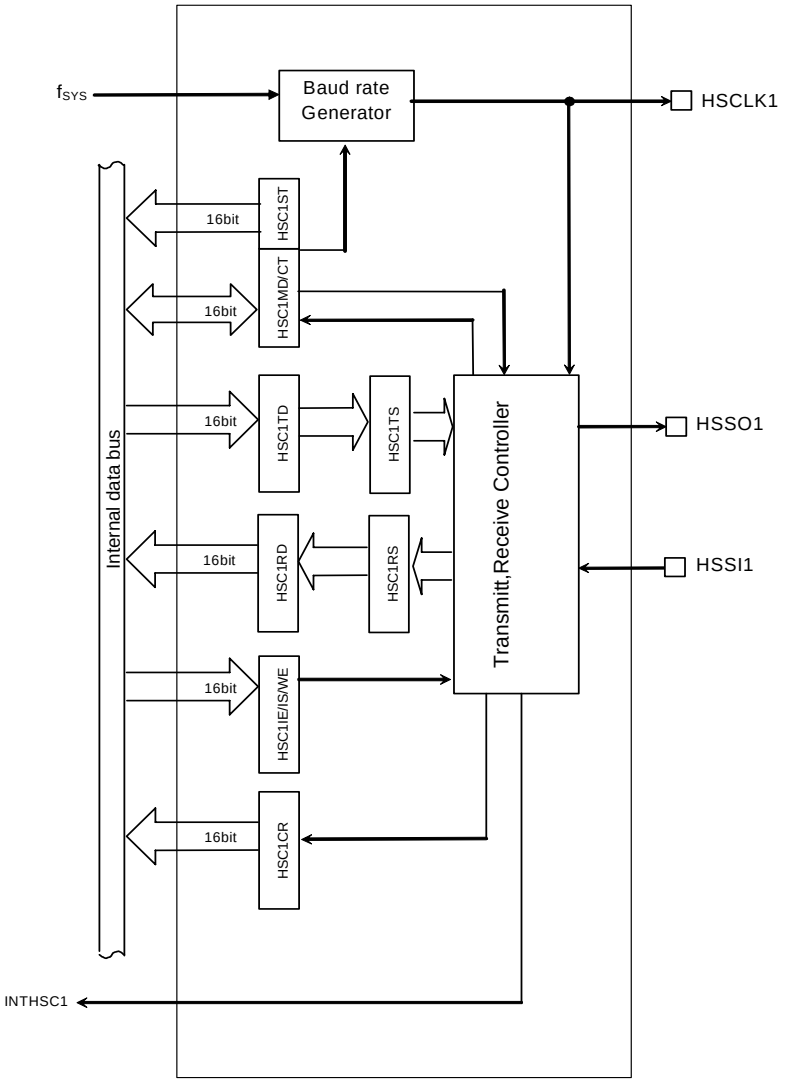
3.12.1 Block diagram

The block diagram of each channel is shown in the figure 3.12.1 and figure 3.12.2.



Note) By Reset, HSCLK0, HSSO0, HSSI0 pin are set to input port (PortD0, D1, D2) so that pull-up resister is needed.

Figure 3.12.1 HSC0 Block diagram



Note) By Reset, HSCLK1, HSSO1, HSSI1 pin are set to input port (PortL4, L5, L6) so that pull-up resister is needed.

Figure 3.12.2 HSC1 Block diagram

3.12.2 SFR

SFR is explained below. These are connected to CPU with 16bit data bus.

(1) Mode setting register

Register is for operation mode or clock etc.

HSC0MD Register

		7	6	5	4	3	2	1	0
HSC0MD (0C00H)	bit Symbol		XEN0				CLKSEL02	CLKSEL01	CLKSEL00
	Read/Write		R/W				R/W		
	After Reset		0				1	0	0
	Function		SYSCK 0: disable 1: enable				Select baud rate 000: Reserved 100: f_{SYS}/16 001: f_{SYS}/2 101: f_{SYS}/32 010: f_{SYS}/4 111: f_{SYS}/64 011: f_{SYS}/8 111:Reserved		
		15	14	13	12	11	10	9	8
(0C01H)	bit Symbol	LOOPBACK0	MSB1ST0	DOSTAT0		TCPOL0	RCPOL0	TDINV0	RDINV0
	Read/Write	R/W				R/W			
	After Reset	0	1	1		0	0	0	0
	Function	LOOPBACK test mode 0:disbale 1:enable	Start bit for transmit/receive 0:LSB 1:MSB	HSSO0 pin (no transmit) 0:fixed to "0" 1:fixed to "1"		Synchronous clock edge during transmitting 0: fall 1: rise	Synchronous clock edge during receiving 0: fall 1: rise	Invert data During transmitting 0: disable 1: enable	Invert data During receiving 0: disable 1: enable

Figure 3.12.3 HSC0MD Register

HSC1MD Register

		7	6	5	4	3	2	1	0
HSC1MD (0C20H)	bit Symbol		XEN1				CLKSEL12	CLKSEL11	CLKSEL10
	Read/Write		R/W				R/W		
	After Reset		0				1	0	0
	Function		SYSCK 0: disable 1: enable				Select baud rate 000: Reserved 100: f_{SYS}/16 001: f_{SYS}/2 101: f_{SYS}/32 010: f_{SYS}/4 111: f_{SYS}/64 011: f_{SYS}/8 111:Reserved		
		15	14	13	12	11	10	9	8
(0C21H)	bit Symbol	LOOPBACK1	MSB1ST1	DOSTAT1		TCPOL1	RCPOL1	TDINV1	RDINV1
	Read/Write	R/W				R/W			
	After Reset	0	1	1		0	0	0	0
	Function	LOOPBACK test mode 0:disbale 1:enable	Start bit for transmit/receive 0:LSB 1:MSB	HSSO1 pin (no transmit) 0:fixed to "0" 1:fixed to "1"		Synchronous clock edge during transmitting 0: fall 1: rise	Synchronous clock edge during receiving 0: fall 1: rise	Invert data During transmitting 0: disable 1: enable	Invert data During receiving 0: disable 1: enable

Figure 3.12.4 HSC1MD Register

(a) <LOOPBACK0>

Because Internal HSSO0 can be input to internal HSSI0, it can be used as test.
Please change the setting when transmitting/receiving is not in operation.

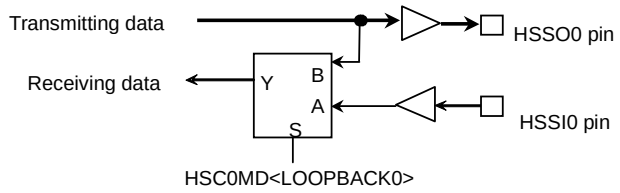


Figure 3.12.5 <LOOPBACK0> Register Function

(b) <MSB1ST0>

Select the start bit of transmit/receive data
Please change the setting when transmitting/receiving is not in operation.

(c) <DOSTAT0>

Set the status of HSSO0 pin during no transmitting (after transmitting or during receiving).
Please change the setting when transmitting/receiving is not in operation.

(d) <TCPOL0>

Select the edge of synchronous clock during transmitting.
Please change the setting during <XEN0> = "0". And set the same value of <RCPOL0>.

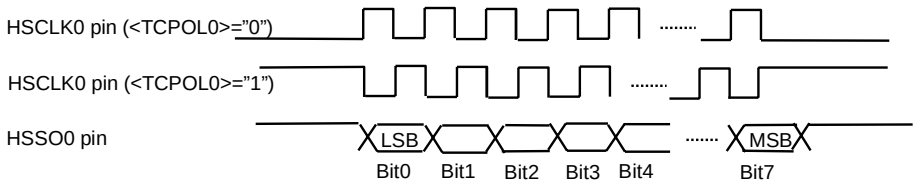


Figure 3.122.6 <TCPOL0> Register function

(e) <RCPOL0>

Select the edge of synchronous clock during receiving.
Please change the setting during <XEN0>= "0". And set the same value of <TCPOL0>.

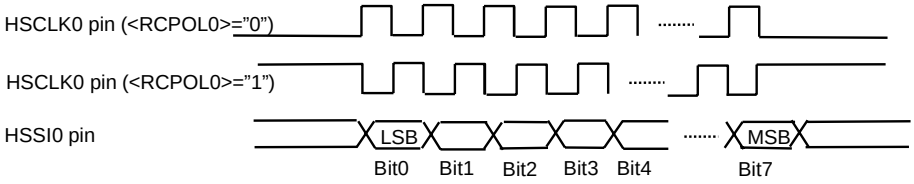


Figure 3.12.7 <TCPOL0> Register function

(f) <TDINV0>

Select logical invert/no invert when output transmitted data from HSSO0 pin.
Please change the setting when transmitting/receiving is not in operation.
Data that input to CRC calculation circuit is transmission data that is written to HSC0TD.
This input data is not corresponded to <TDINV0>.
<TDINV0> is not corresponded to <DOSTAT0>: it set condition of HSSO0 pin when it is not transferred.

(g) <RDINV0>

Select logical invert/no invert for received data from HSSI0 pin.
Please change the setting when transmitting/receiving is not in operation.
Data that input to CRC calculation circuit is selected by <RDINV0>.

(h) <XEN0>

Select the operation for the internal clock.

(i) <CLKSEL02:00>

Select baud rate. Baud rate is created from f_{SYS} and settings are in under table.
Please change the setting when transmitting/receiving is not in operation.

Table 3.12.2 Example of baud rate

<CLKSEL02:00>	Baud rate [Mbps]		
	f_{SYS} =12MHz	f_{SYS} =16MHz	f_{SYS} =20MHz
$f_{SYS}/2$	6	8	10
$f_{SYS}/4$	3	4	5
$f_{SYS}/8$	1.5	2	2.5
$f_{SYS}/16$	0.75	1	1.25
$f_{SYS}/32$	0.375	0.5	0.625
$f_{SYS}/64$	0.1875	0.25	0.3125

(2) Control Register

Register is for data length or CRC etc.

HSC0CT Register

		7	6	5	4	3	2	1	0
HSC0CT (0C02H)	bit Symbol	–	–	UNIT160			ALGNEN0	RXWEN0	RXUEN0
	Read/Write	R/W					R/W		
	After Reset	0	1	0			0	0	0
	Function	Always write “0”.	Always write “1”.	Data length 0: 8bit 1: 16bit			Full duplex alignment 0: disable 1: enable	Sequential receive 0: disable 1: enable	Receive UNIT 0: disable 1: enable
		15	14	13	12	11	10	9	8
(0C03H)	bit Symbol	CRC16_7_B0	CRCRX_TX_B0	CRCRESET_B0				DMAERFW0	DMAERFR0
	Read/Write	R/W						R/W	R/W
	After Reset	0	0	0				0	0
	Function	CRC select 0: CRC7 1: CRC16	CRC data 0: Transmit 1: Receive	CRC calculate register 0:Reset 1:Release Reset				Micro DMA 0: Disable 1: Enable	Micro DMA 0: Disable 1: Enable

Figure 3.12.8 HSC0CT Register

HSC1CT Register

		7	6	5	4	3	2	1	0
HSC1CT (C22H)	bit Symbol	–	–	UNIT161			ALGNEN1	RXWEN1	RXUEN1
	Read/Write	R/W					R/W		
	After Reset	0	1	0			0	0	0
	Function	Always write “0”.	Always write “1”.	Data length 0: 8bit 1: 16bit			Full duplex alignment 0: disable 1: enable	Sequential receive 0: disable 1: enable	Receive UNIT 0: disable 1: enable
		15	14	13	12	11	10	9	8
(C23H)	bit Symbol	CRC16_7_B1	CRCRX_TX_B1	CRCRESET_B1				DMAERFW1	DMAERFR1
	Read/Write	R/W						R/W	R/W
	After Reset	0	0	0				0	0
	Function	CRC select 0: CRC7 1: CRC16	CRC data 0: Transmit 1: Receive	CRC calculate register 0:Reset 1:Release Reset				Micro DMA 0: Disable 1: Enable	Micro DMA 0: Disable 1: Enable

Figure 3.12.9 HSC1CT Register

(a) <CRC16_7_B0>

Select CRC7 or CRC16 to calculate.

(b) <CRCRX_TX_B0>

Select input data to CRC calculation circuit.

(c) <CRCRESET_B0>

Initialize CRC calculate register.

The process that calculating CRC16 of transmits data and sending CRC next to transmit data is explained as follows.

1. Set HSC0CT<CRC16_7_B0> for select CRC7 or CRC16 and <CRCRX_TX_B0> for select calculating data.
2. For reset HSC0CR register, write “1” after set <CRCRESET_B0> to "0".
3. Write transmit data to HSC0TD register, and wait for finish transmission all data.
4. Read HSC0CR register, and obtain the result of CRC calculation.
5. Transmit CRC which is obtained in (4) by the same way as (3).

CRC calculation of receive data is the same process.

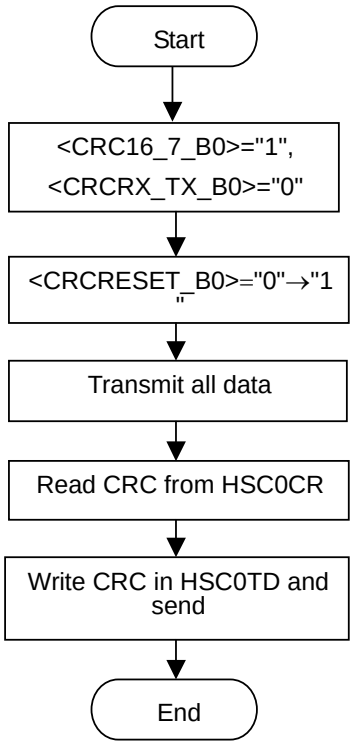


Figure 3.12.10 Flow chart of CRC calculation

(d) <DMAERFW0>

Set clearing interrupt in CPU to unnecessary because be supported RFW0 interrupt to Micro DMA. If write "1" to, it be set to one-shot interrupt, clearing interrupt by HSC0WE register become to unnecessary. HSC0ST<RFW0> flag generate 1-shot interrupt when change from "0" to "1"(Rising).

(e) <DMAERFR0>

Set clearing interrupt in CPU to unnecessary because be supported RFR0 interrupt to Micro DMA. If write "1" to, it be set to one-shot interrupt, clearing interrupt by HSC0WE register become to unnecessary. HSC0ST<RFR0> flag generate 1-shot interrupt when change from "0" to "1"(Rising).

(f) <UNIT160>

Select the length of transmit/receive data. Data length is described as UNIT downward. Please change the setting when transmitting/receiving is not in operation.

(g) <ALGNEN0>

Select whether using alignment function for transmit/receive per UNIT during full duplex.
Please change the setting when transmitting/receiving is not in operation.

(h) <RXWEN0>

Set enable/disable of sequential receiving.

(i) <RXUEN0>

Set enable/disable of receiving operation per UNIT. In case <RXWEN0> = "1", this bit is not valid.
Please change the setting when transmitting/receiving is not in operation.

[Transmit / receive operation mode]

It is supported 6 operation modes. They are selected in <ALGNEN0>, <RXWEN0> and <RXUEN0> registers.

Table 3.12.3 transmit/receive operation mode

Operation mode	Register setting			Note
	<ALGNEN0>	<RXWEN0>	<RXUEN0>	
(1) Transmit UNIT	0	0	0	Transmit written data per UNIT
(2) Sequential transmit	0	0	0	Transmit written data sequentially
(3) Receive UNIT	0	0	1	Receive data of only 1 UNIT
(4) Sequential receive	0	1	0	Receive automatically if buffer has space
(5) Transmit/Receive UNIT with alignment	1	0	1	Transmit/receive 1 UNIT with alignment per each UNIT
(6) Sequential Transmit/Receive UNIT with alignment	1	1	0	Transmit/receive sequentially with alignment per each UNIT

Difference between UNIT transmission and Sequential transmission

UNIT transmit mode is transmitted every 1 UNIT by writing data after confirmed HSC0ST<TEND0>=1.The written transmission data is shifted in turn. In hard ware, transmission is kept executing as long as data exists. If it transmit data sequentially, write next data when HSC0TD is empty and HSC0ST<REND0>=1.

UNIT transmission and sequential transmission depend on the way of using. Hardware doesn't depend on.

Figure 3.12.11 show Flow chart of UNIT transmission and Sequential transmission.

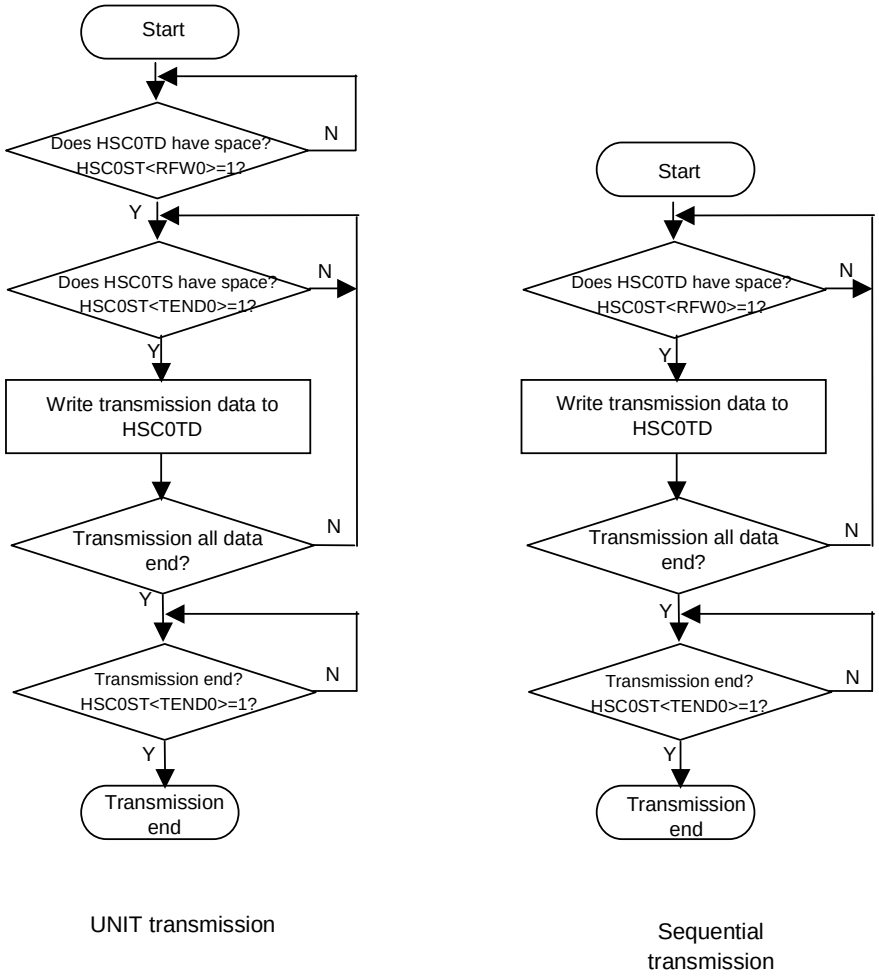


Figure 3.12.11 Flow chart of UNIT transmission and Sequential transmission

Difference between UNIT receive and Sequential receive

UNIT receive is the mode that receiving only 1 UNIT data.

By writing "1" to HSC0CT<RXUEN0>, receives 1UNIT data, and received data is loaded in receive data register (HSC0RD). When HSC0RD register is read, read it after wrote "0" to HSC0CT<RXUEN0>.

If data was read from HSC0RD with the condition HSC0CT<RXE0>= "1", 1 UNIT data is received again automatically. In hardware, this mode receives sequentially by Single buffer.

HSC0ST<REND0> is changed during UNIT receiving.

Sequential receive is the mode that receive data and automatically when receive FIFO has space.

Whenever buffer has space, next data is received automatically. Therefore, if data was read after data is loaded in HSC0RD, it is received sequentially every UNIT. In hardware, this mode receives sequentially by double buffer.

Figure 3.12.12 show Flow chart of UNIT receive and Sequential receive.

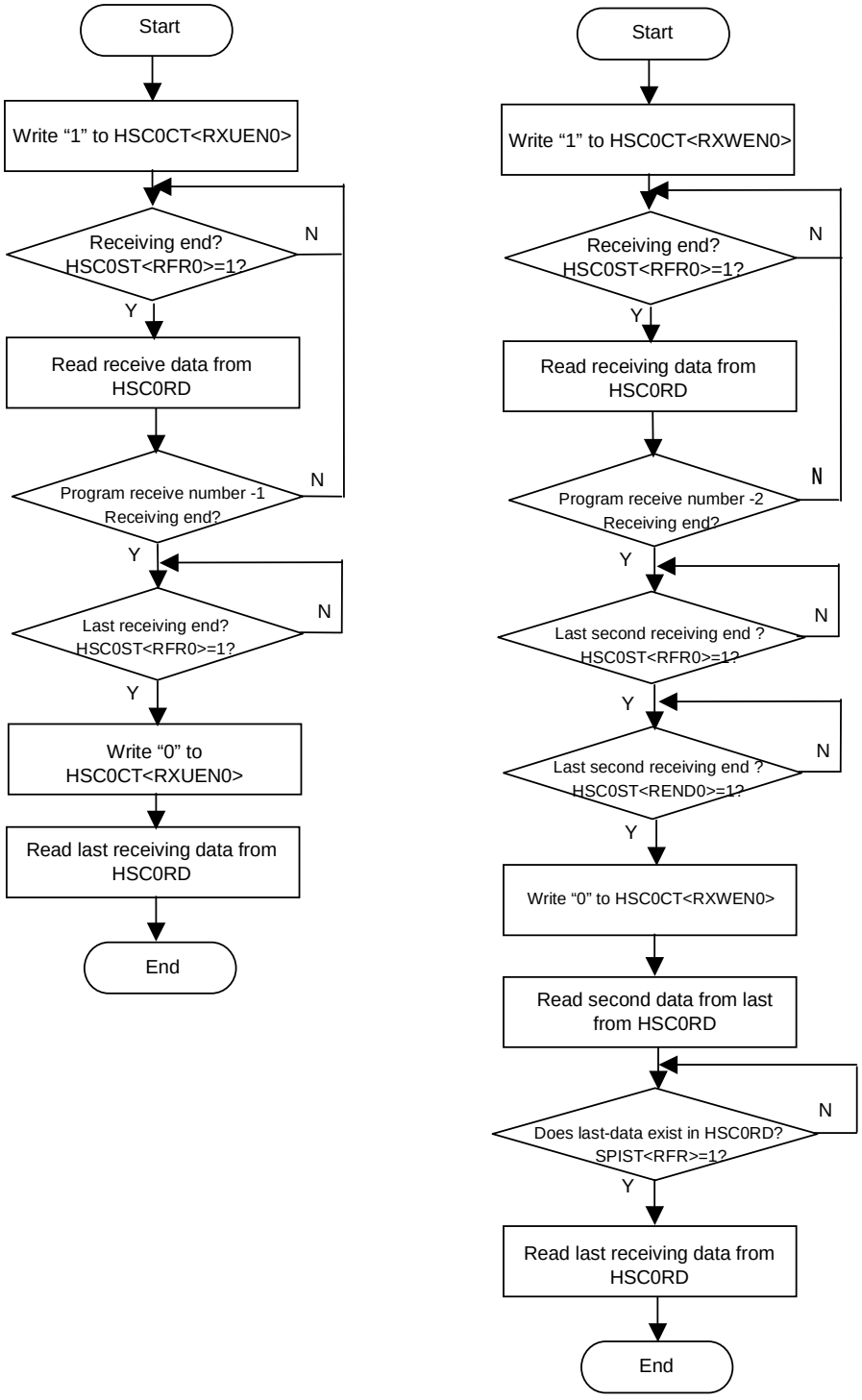


Figure 3.12.12 Flow chart of UNIT receive and Sequential receive

(3) Interrupt , Status register

Read of condition, Mask of condition, Clear interrupt and Clear enable can control each 4 interrupts; RFR0(HSC0RD receiving buffer is full), RFW0(HSC0TD transmission buffer is empty), REND0(HSC0RS receiving buffer is full), TEND0(HSC0TS transmission buffer is empty).

RFR0, RFW0 can high-speed transaction by micro DMA.

Following is description of Interrupt・status (example RFW0).

Status register HSC0ST<RFW0> show RFW0 (internal signal that show whether transmission data register exist or not). This register is "0" when transmission data exist. This register is "1" when transmission data doesn't exist. It can read internal signal directly. Therefore, it can confirm transmission data at any time.

Interrupt status register HSC0IS<RFWIS0> is set by rising edge of RFW0. This register keeps that condition until write "1" to this register and reset when HSC0WE<RFWWE0> is "1".

RFW0 interrupt generate when interrupt enable register HSC0IE<RFWIE0> is "1". When it is "0", interrupt is not generated.

Interrupt request register HSC0IR<RFWIR0> show whether interrupt is generating or not.

Interrupt status write enable register HSC0WE<RFWWE0> set that enables reset for reset interrupts status register by mistake.

Circuit config of transmission data shift register (HSC0TS), receiving register (HSC0RD), receiving data shift register (HSC0RS) are same with above register.

Control register HSC0CT<DMAERFW0>, HSC0CT<DMAERFR0> is register for using micro DMA. When micro DMA transfer is executed by using RFW0 interrupt, set "1" to <DMAERFW0>, and when it is executed by using RFR0 interrupt, set "1" to <DMAERFR0>, and prohibit other interrupt.

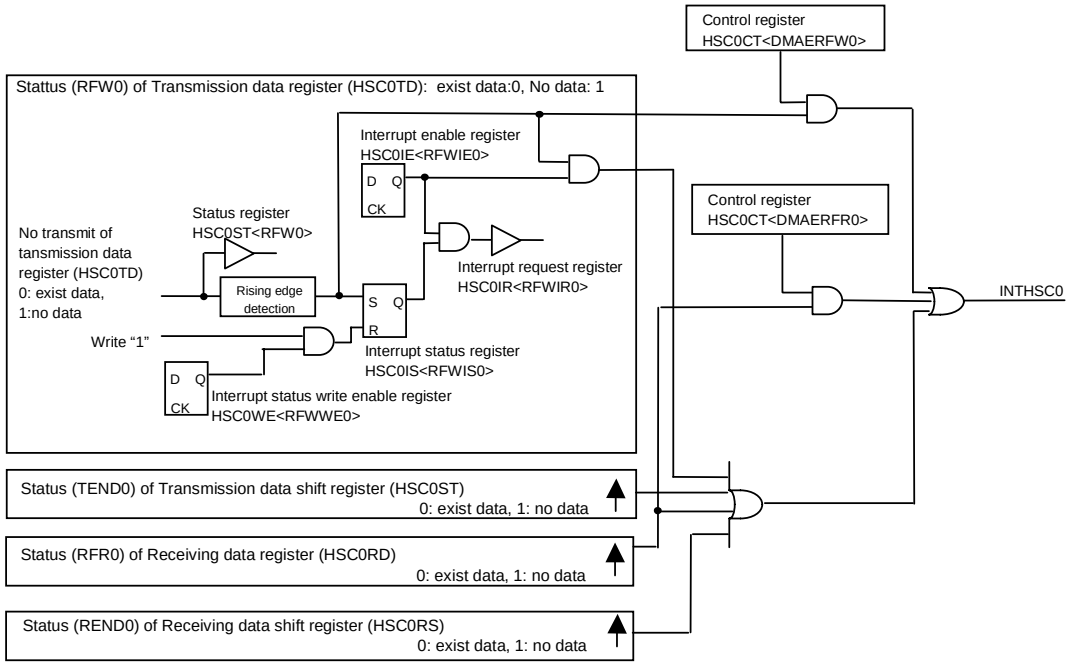


Figure 3.12.2 Figurer for interrupt, status

(3-1) Status register

Register shows 4 status.

HSC0ST Register

		7	6	5	4	3	2	1	0
HSC0ST (0C04H)	bit Symbol					TEND0	REND0	RFW0	RFR0
	Read/Write					R			
	After Reset					1	0	1	0
	Function					Receiving 0:operation 1: no operation	Receive Shift register 0: no data 1: exist data	Transmit buffer 0: untransmitted data exist 1: no untransmitted data	Receive buffer 0:no valid data 1:valid data exist
		15	14	13	12	11	10	9	8
(0C05H)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.3 HSC0ST Register

HSC1ST Register

		7	6	5	4	3	2	1	0
HSC1ST (0C24H)	bit Symbol					TEND1	REND1	RFW1	RFR1
	Read/Write					R			
	After Reset					1	0	1	0
	Function					Receiving 0:operation 1: no operation	Receive Shift register 0: no data 1: exist data	Transmit buffer 0: untransmitted data exist 1: no untransmitted data	Receive buffer 0:no valid data 1:valid data exist
		15	14	13	12	11	10	9	8
(0C25H)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.4 HSC1ST Register

(a) <TEND0>

This bit is set to “0” when valid data to transmit exists in the shift register for transmit. It is set to “1” when finish transmitting all the data.

(b) <REND0>

This bit is set to “0” when receiving is in operation or no valid data exist in receive shift register.

It is set to “1”, when valid data exist in receive read register and keep the data without shifting.

It is cleared to “0”, when CPU read the data and shift to receive read register.

(c) <RFW0>

After wrote the received data to receive data write register, shift the data to receive data shift register. It keeps “0” until all valid data has moved. And it is set to “1” when it can accept the next data with no valid data.

(d) <RFR0>

This bit is set to “1” when received data is shifted from received data shift register to received data read register and valid data exist. It is set to “0” when the data is read and no valid data.

(3-2) Interrupt status register

Register read 4 interrupt status and clear interrupt.

This register is cleared to "0" by writing "1" to applicable bit. Status of this register show interrupt source state. This register can confirm changing of interrupt condition, even if interrupt enable register is masked.

HSC0IS Register

		7	6	5	4	3	2	1	0
HSC0IS (0C08H)	bit Symbol					TENDIS0	RENDIS0	RFWIS0	RFRIS0
	Read/Write					R/W			
	After Reset					0	0	0	0
	Function					Read 0:no interrupt 1:interrupt Write 0:Don't care 1:clear	Read 0:no interrupt 1:interrupt Write 0:Don't care 1:clear	Read 0:no interrupt 1:interrupt Write 0:Don't care 1:clear	Read 0:nointerrupt 1:interrupt Write 0:Don't care 1:clear
		15	14	13	12	11	10	9	8
(0C09H)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.5 HSC0IS Register

HSC0IS Register

		7	6	5	4	3	2	1	0
HSC1IS (0C28H)	bit Symbol					TENDIS1	RENDIS1	RFWIS1	RFRIS1
	Read/Write					R/W			
	After Reset					0	0	0	0
	Function					Read 0:no interrupt 1:interrupt Write 0:Don't care 1:clear	Read 0:no interrupt 1:interrupt Write 0:Don't care 1:clear	Read 0:no interrupt 1:interrupt Write 0:Don't care 1:clear	Read 0:nointerrupt 1:interrupt Write 0:Don't care 1:clear
		15	14	13	12	11	10	9	8
(0C29H)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.6 HSC1IS Register

(a) <TENDIS0>

This bit read status of TEND interrupt and clear interrupt.
If write this bit, set “1” to HSC0WE<TENDWE0>.

(b) <REMDIS0>

This bit read status of REND interrupt and clear interrupt.
If write this bit, set “1” to HSC0WE<RENDWE0>.

(c) <RFWDIS0>

This bit read status of RFW interrupt and clear interrupt.
If write this bit, set “1” to HSC0WE<RFWWE0>.

(d) <RFRIS0>

This bit read status of RFR interrupt and clear interrupt.
If write this bit, set “1” to HSC0WE<RFRWE0>.

(3-3) Interrupt status write enable register

Register set clear enable for 4 interrupt stasus bit.

HSC0WE Register

		7	6	5	4	3	2	1	0
HSC0WE (0C0AH)	bit Symbol					TENDWE0	RENDWE0	RFWWE0	RFRWE0
	Read/Write					R/W			
	After Reset					0	0	0	0
	Function					Clear HSC0IS <TENDIS0> 0: disable 1: enable	Clear HSC0IS <RENDIS0> 0: disable 1: enable	Clear HSC0IS <TFWIS0> 0: disable 1: enable	Clear HSC0IS <RFRIS0> 0: disable 1: enable
		15	14	13	12	11	10	9	8
(0C0BH)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.20 HSC0WE Register

HSC1WE Register

		7	6	5	4	3	2	1	0
HSC1WE (0C2AH)	bit Symbol					TENDWE1	RENDWE1	RFWWE1	RFRWE1
	Read/Write					R/W			
	After Reset					0	0	0	0
	Function					Clear HSC1IS <TENDIS1> 0: disable 1: enable	Clear HSC1IS <RENDIS1> 0: disable 1: enable	Clear HSC1IS <TFWIS1> 0: disable 1: enable	Clear HSC1IS <RFRIS1> 0: disable 1: enable
		15	14	13	12	11	10	9	8
(0C2BH)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.21 HSC1WE Register

(a) <TENDWE0>

This bit set clear enable of HSC0IS<TENDIS0>.

(b) <RENDWE0>

This bit set clear enable of HSC0IS<RENDIS0>.

(c) <RFWWE0>

This bit set clear enable of HSC0IS<RFWIS0>.

(d) <RFRWE0>

This bit set clear enable of HSC0IS<RFRIS0>.

(3-4) Interrupt enable register

Register set output enable for 4 interrupt.

HSC0IE Register

		7	6	5	4	3	2	1	0
HSC0IE (0C0CH)	bit Symbol					TENDIE0	RENDIE0	RFWIE0	RFRIE0
	Read/Write					R/W			
	After Reset					0	0	0	0
	Function					TEND0 interrupt 0: Disable 1: Enable	REND0 interrupt 0: Disable 1: Enable	RFW0 interrupt 0: Disable 1: Enable	RFR0 interrupt 0: Disable 1: Enable
		15	14	13	12	11	10	9	8
(0C0DH)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.22 HSC0IE Register

HSC1IE Register

		7	6	5	4	3	2	1	0
HSC1IE (0C2CH)	bit Symbol					TENDIE1	RENDIE1	RFWIE1	RFRIE1
	Read/Write					R/W			
	After Reset					0	0	0	0
	Function					TEND1 interrupt 0: Disable 1: Enable	REND1 interrupt 0: Disable 1: Enable	RFW1 interrupt 0: Disable 1: Enable	RFR1 interrupt 0: Disable 1: Enable
		15	14	13	12	11	10	9	8
(0C2DH)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.23 HSC1IE Register

(a) <TENDIE0>

This bit set TEND0 interrupt enable.

(b) <RENDIE0>

This bit set REND0 interrupt enable.

(c) <RFWIE0>

This bit set RFW0 interrupt enable.

(d) <RFRIE0>

This bit set RFR0 interrupt enable.

(3-5) Interrupt request register

Register show generation condition for 4 interrupts.

This regiter read "0" (interrupt doesn't generate) always when Interrupt enable register is masked.

HSC0IR Register

		7	6	5	4	3	2	1	0
HSC0IR (0C0EH)	bit Symbol					TENDIR0	RENDIR0	RFWIR0	RFRIR0
	Read/Write					R			
	After Reset					0	0	0	0
	Function					TEND0 interrupt 0: none 1:generate	REND0 interrupt 0: none 1:generate	RFW0 interrupt 0: none 1:generate	RFR0 interrupt 0: none 1:generate
		15	14	13	12	11	10	9	8
(0C0FH)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.24 HSC0IR Register

HSC1IR Register

		7	6	5	4	3	2	1	0
HSC1IR (0C2EH)	bit Symbol					TENDIR1	RENDIR1	RFWIR1	RFRIR1
	Read/Write					R			
	After Reset					0	0	0	0
	Function					TEND1 interrupt 0: none 1:generate	REND1 interrupt 0: none 1:generate	RFW1 interrupt 0: none 1:generate	RFR1 interrupt 0: none 1:generate
		15	14	13	12	11	10	9	8
(0C2FH)	bit Symbol								
	Read/Write								
	After Reset								
	Function								

Figure 3.12.25 HSC1IR Register

(a) <TENDIR0>

This bit shows condition of TEND0 interrupt generation.

(b) <TENDIR0>

This bit shows condition of REND0 interrupt generation.

(c) <RFWIR0>

This bit shows condition of RFW0 interrupt generation.

(d) <RFRIR0>

This bit shows condition of RFR0 interrupt generation.

(4) HSC0CR (HSC0 CRC register)

Register load result of CRC calculation for transmission/receiving in it.

HSC0CR register

		7	6	5	4	3	2	1	0
HSC0CR (0C06H)	bit Symbol	CRCD007	CRCD006	CRCD005	CRCD004	CRCD003	CRCD002	CRCD001	CRCD000
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	CRC calculation result load register [7:0]							
		15	14	13	12	11	10	9	8
(0C07H)	bit Symbol	CRCD015	CRCD014	CRCD013	CRCD012	CRCD011	CRCD010	CRCD009	CRCD008
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	CRC calculation result load register [15:8]							

Figure 3.12.26 HSC0CR register

HSC1CR register

		7	6	5	4	3	2	1	0
HSC1CR (0C26H)	bit Symbol	CRCD107	CRCD106	CRCD105	CRCD104	CRCD103	CRCD102	CRCD101	CRCD100
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	CRC calculation result load register [7:0]							
		15	14	13	12	11	10	9	8
(0C27H)	bit Symbol	CRCD115	CRCD114	CRCD113	CRCD112	CRCD111	CRCD110	CRCD109	CRCD108
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	CRC calculation result load register [15:8]							

Figure 3.12.27 HSC1CR register

(a) <CRCD015:000>

The result that is calculated according to the setting; HSC0CT<CRC16_7_b0>, <CRCRX_TX_B0> and <CRCRESET_B0>, are loaded in this register.
In case CRC16, all bits are valid. In case CRC7, lower 7 bits are valid.
The flow will be showed to calculate CRC16 of received data for instance by flowchart.
Firstly, initialize CRC calculation register by writing <CRCRESET_B0> = "1" after set <CRC16_7_b0> = "1", <CRCRX_TX_B0> = "0", <CRCRESET_B0> = "0".
Next, finish transmitting all bits to calculate CRC by writing data in HSC0TD register.
Confirming whether receiving is finished or not use HSC0ST<TEND0>.
If HSC0CR register was read after finish, CRC16 of transmission data can read.

(5) Transmission data register

Register is register for write transmission data.

HSC0TD Register

		7	6	5	4	3	2	1	0
HSC0TD (0C10H)	bit Symbol	TXD007	TXD006	TXD005	TXD004	TXD003	TXD002	TXD001	TXD000
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	0	0
	Function	Transmission data register [7:0]							
		15	14	13	12	11	10	9	8
(0C11H)	bit Symbol	TXD015	TXD014	TXD013	TXD012	TXD011	TXD010	TXD009	TXD008
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	0	0
	Function	Transmission data register [15:8]							

Figure 3.12.28 HSC0TD Register

HSC1TD Register

		7	6	5	4	3	2	1	0
HSC1TD (0C30H)	bit Symbol	TXD0107	TXD106	TXD105	TXD104	TXD103	TXD102	TXD101	TXD100
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	0	0
	Function	Transmission data register [7:0]							
		15	14	13	12	11	10	9	8
(0C31H)	bit Symbol	TXD115	TXD114	TXD113	TXD112	TXD111	TXD110	TXD109	TXD108
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	0	0
	Function	Transmission data register [15:8]							

Figure 3.12.29 HSC1TD Register

(a) <TXD015:000>

This bit is bit for write transmission data. When read, the last written data is read.
The data is overwritten when next data was written with condition of this register does not empty. In this case, please write after checked the status of RFW0.
In case HSC0CT<UNIT160>= “1”, all bits are valid.
In case HSC0CT<UNIT160>= “0”, lower 7 bits are valid.

(6) Receiving data register

Register is register for read receiving data.

HSC0RD Register

		7	6	5	4	3	2	1	0
HSC0RD (0C12H)	bit Symbol	RXD007	RXD006	RXD005	RXD004	RXD003	RXD002	RXD001	RXD000
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Receive data register [7:0]							
		15	14	13	12	11	10	9	8
(0C13H)	bit Symbol	RXD015	RXD014	RXD013	RXD012	RXD011	RXD010	RXD009	RXD008
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Receive data register [15:8]							

Figure 3.12.30 HSC0RD Register

HSC1RD Register

		7	6	5	4	3	2	1	0
HSC1RD (0C32H)	bit Symbol	RXD107	RXD106	RXD105	RXD104	RXD103	RXD102	RXD101	RXD100
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Receive data register [7:0]							
		15	14	13	12	11	10	9	8
(0C33H)	bit Symbol	RXD115	RXD114	RXD113	RXD112	RXD111	RXD110	RXD109	RXD108
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Receive data register [15:8]							

Figure 3.12.31 HSC1RD Register

(a) <RXD015:000>

HSC0RD register is register for reading receiving data. Please read after checked status of RFK.

In case HSC0CT<UNIT160> = “1”, all bits are valid.

In case HSC0CT<UNIT160> = “0”, lower 7 bits are valid.

(7) Transmit data shift register

Register change transmission data to serial. This register is used for confirming changing condition when LSI test.

HSC0TS Register

		7	6	5	4	3	2	1	0
HSC0TS (0C14H)	bit Symbol	TSD007	TSD006	TSD005	TSD004	TSD003	TSD002	TSD001	TSD000
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Transmit data shift register [7:0]							
		15	14	13	12	11	10	9	8
	bit Symbol	TSD015	TSD014	TSD013	TSD012	TSD011	TSD010	TSD009	TSD008
(0C15H)	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Transmit data shift register [15:8]							

Figure 3.12.32 HSC0TS Register

HSC1TS Register

		7	6	5	4	3	2	1	0
HSC1TS (0C34H)	bit Symbol	TSD107	TSD106	TSD105	TSD104	TSD103	TSD102	TSD101	TSD100
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Transmit data shift register [7:0]							
		15	14	13	12	11	10	9	8
	bit Symbol	TSD115	TSD114	TSD113	TSD112	TSD111	TSD110	TSD109	TSD108
(0C35H)	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Transmit data shift register [15:8]							

Figure 3.12.33 HSC1TS Register

(a) <TSD015:000>

This register is register for reading the status of transmission data shift register.
In case HSC0CT<UNIT160>= “1”, all bits are valid.
In case HSC0CT<UNIT160>= “0”, lower 7 bits are valid.

(8) Receive data shift register

Register is register for reading receive data shift register.

HSC0RS Register

		7	6	5	4	3	2	1	0
HSC0RS (0C16H)	bit Symbol	RSD007	RSD006	RSD005	RSD004	RSD003	RSD002	RSD001	RSD000
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Receive data shift register [7:0]							
		15	14	13	12	11	10	9	8
(0C17H)	bit Symbol	RSD015	RSD014	RSD013	RSD012	RSD011	RSD010	RSD009	RSD008
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	function	Receive data shift register [15:8]							

Figure 3.12.34 HSC0RS Register

HSC1RS Register

		7	6	5	4	3	2	1	0
HSC1RS (0C36H)	bit Symbol	RSD107	RSD106	RSD105	RSD104	RSD103	RSD102	RSD101	RSD100
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function	Receive data shift register [7:0]							
		15	14	13	12	11	10	9	8
(0C37H)	bit Symbol	RSD115	RSD114	RSD113	RSD112	RSD111	RSD110	RSD109	RSD108
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	function	Receive data shift register [15:8]							

Figure 3.12.35 HSC1RS Register

(a) <RSD015:000>

This register is register for reading the status of receives data shift register.
In case HSC0CT<UNIT160>= “1”, all bits are valid.
In case HSC0CT<UNIT160>=“0”, lower 7 bits are valid.

3.12.3 Operation timing

Following examples show operation timing.

- Setting condition 1:
  Transmission in UNIT=8bit, LSB first

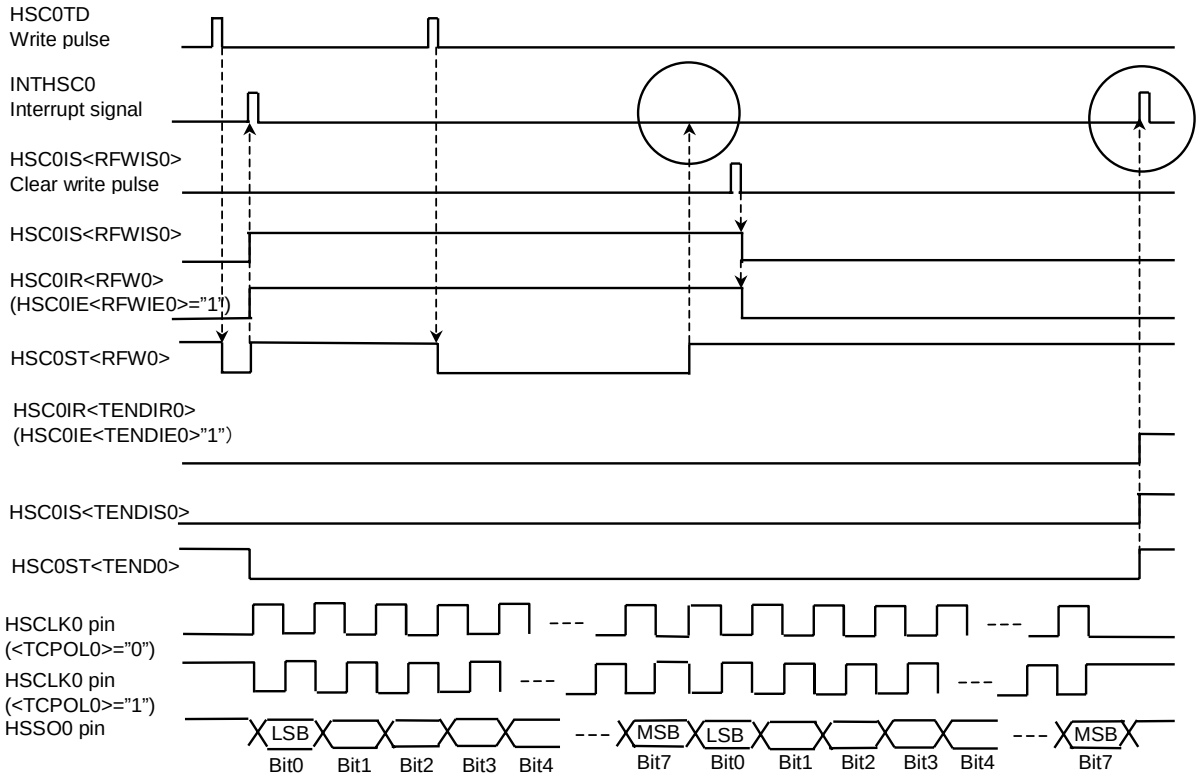


Figure 3.12.36 Transmission timing

In above condition, HSC0ST<RFW0> flag is set to "0" just after wrote transmission data. When data of HSC0TD register finish shifting to transmission register (HSC0TS), HSC0ST<RFW0> is set to "1", it is informed that can write next transmission data, start transmission clock and data from HSCLK0 pin and HSSO0 pin at same time with inform.

In this case, HSC0IS, HSC0IR change and INTHSC0 interrupt generate by synchronization to rising of HSC0ST<RFW0> flag. When HSC0IR register is setting to "1", interrupt is not generated even if HSC0ST<RFW0> was set to "1".

When finish transmission and lose data that must to transmit to HSC0TD register and HSC0TS register, transmission data and clock are stopped by setting "1" to HSC0ST<TEND0>, and INTHSC0 interrupt is generated at same time. In this case, if HSC0ST<TEND0> is set to "1" at different interrupt source, INTHSC0 is not generated. Therefore must to clear HSC0IS<RFW0> to "0".

- Setting condition 2:
  UNIT transmission in UNIT=8bit, LSB first

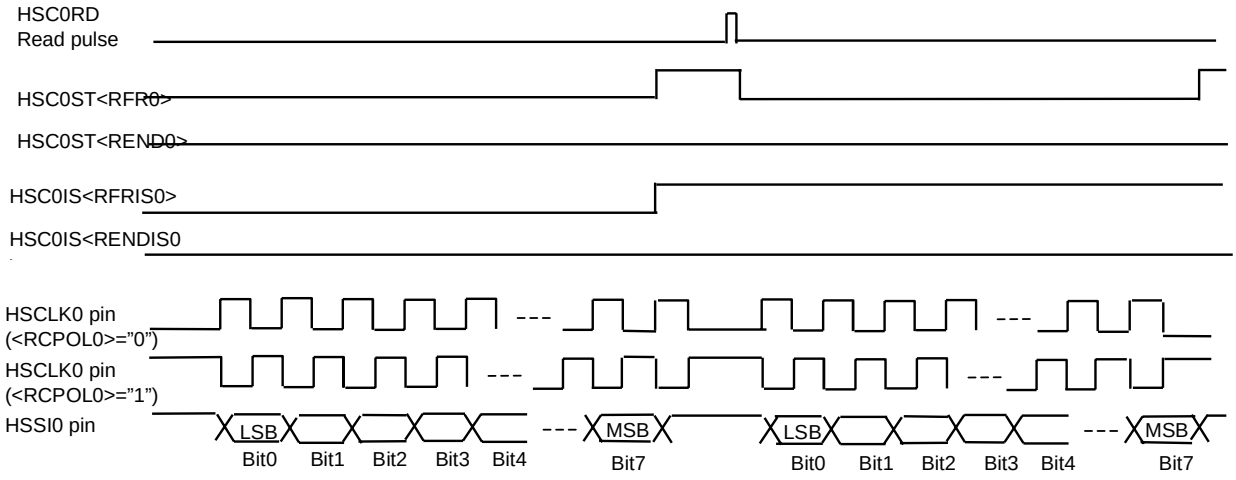


Figure 3.12.37 UNIT receiving (HSC0CT<RXUEN0>=1)

If set HSC0CT<RXUEN0> to "1" without valid receiving data to HSC0RD register (HSC0ST<RFR0>=0), UNIT receiving is started. When receiving is finished and stored receiving data to HSC0RD register, HSC0ST<RFR0> flag is set to "1", and inform that can read receiving data. Just after read HSC0RD register, HSC0ST<RFR0> flag is cleared to "0" and it start receiving next data automatically.

If be finished UNIT receiving, set HSC0CT<RXUEN0> to "0" after confirmed that HSC0ST<RFR0> was set to "1".

- Setting condition 3:
  Sequential receiving in UNIT=8 bit, LSB first

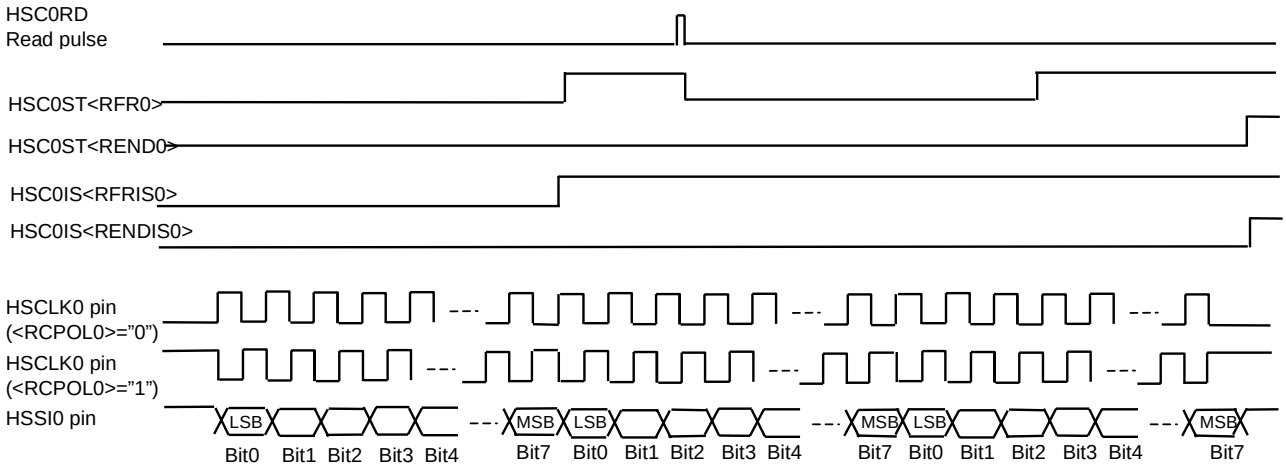


Figure 3.12.38 continuous receiving (HSC0CT<RXWEN0>=1)

If set HSC0CT<RXWEN0> to "1" without valid receiving data in HSC0RD register (HSC0ST<RFR0>=0), sequential receiving is started. When first receiving is finished and stored receiving data to HSC0RD register, HSC0ST<RFR0> flag is set to "1", and inform that can read receiving data. Sequential receiving is received until receiving data is stored to HSC0RD and HSC0RS registers If finished sequential receiving, set HSC0CT<RXWEN0> to "0" after confirmed that HSC0ST<REND0> was set to "1".

- Setting condition 4:
  Transmission by using micro DMA in UNIT=8bit, LSB first

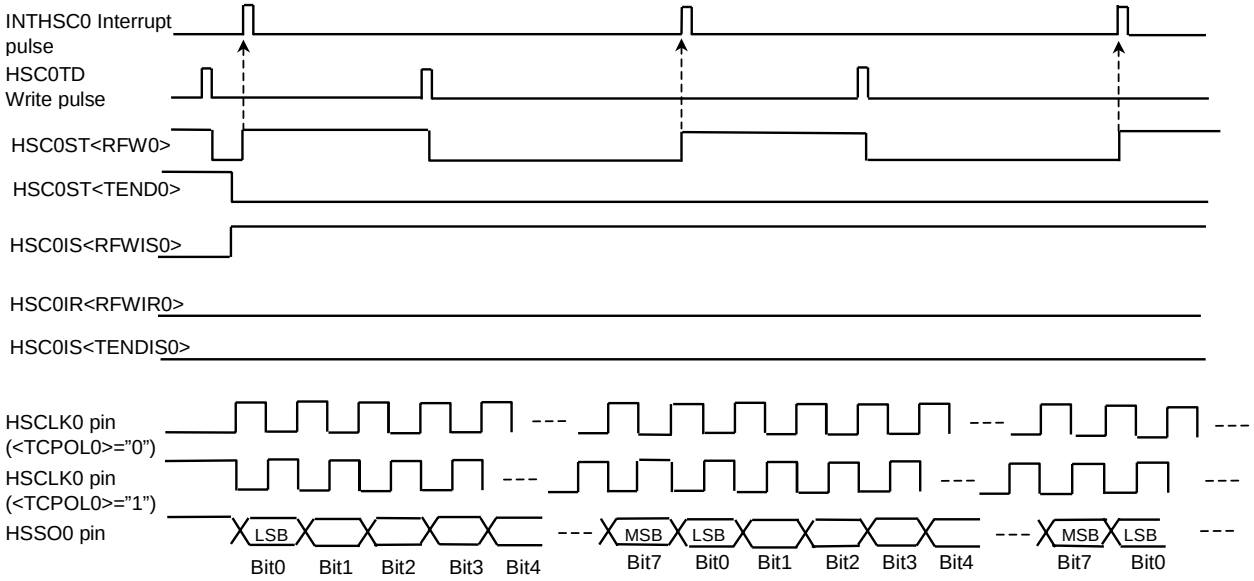


Figure 3.12.39 Micro DMA transmission (transmission)

If all bits of HSC0IE register are "0" and HSC0CT<DMAERFW0> is "1", transmission is started by writing transmission data to HSC0TD register.

If data of HSC0TD register is shifted to HSC0TS register and HSC0ST<RFW0> is set to "1" and can write next transmission data, INTHSC0 interrupt (RFW0 interrupt) is generated. By starting Micro DMA at this interrupt, can transmit sequential data automatically.

However, If transmit it at Micro DMA, set Micro DMA beforehand.

- Setting condition 5:
  Receiving by using micro DMA in UNIT=8bit, LSB first

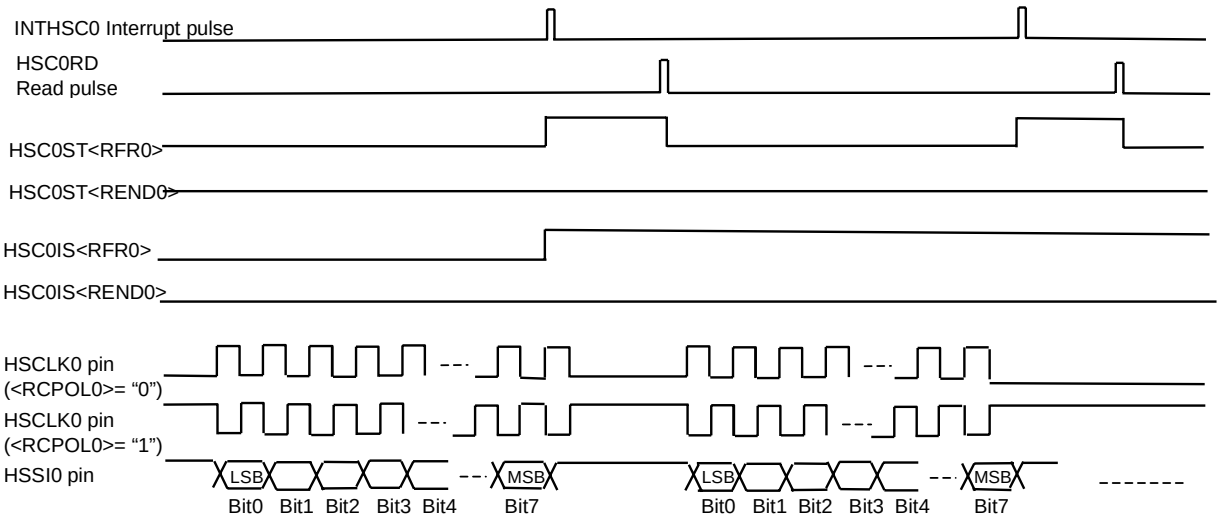


Figure 3.12.40 Micro DMA transmission (UNIT receiving (HSC0CT<RFUEN0>=1))

If all bits of HSC0IE register is "0" and HSC0CT<DMAERFR0> is "1", UNIT receiving is started by setting HSC0CT<RXUEN0> to "1". If receiving data is stored to HSC0RD register and can read receiving data, INTHSC0 interrupt (RFR0 interrupt) is generated. By starting Micro DMA at this interrupt, it can be received sequential data automatically.

However, If receive it at Micro DMA, set Micro DMA beforehand.

3.12.4 Example

Following is discription of HSC0 setting method.

(1) UNIT transmission

This example show case of transmission is executed by following setting, and it is generated INTHSC0 interrupt by finish transmission.

UNIT: 8bit
LSB first
Baud rate : $f_{SYS}/8$
Synchronous clock edge: Rising

Setting expample

```
    ld     (pdfc), 0x07          ; Port setting PD0: HSSI0, PD1: HSSO0, PK7: HSCLK0
    ld     (pdcr), 0x06          ; port setting PD0: HSSI0, PD1: HSSO0, PK7: HSCLK0

    ldw    (hsc0ct),0x0040       ; Set data length to 8bit
    ldw    (hsc0md),0x2c43       ; System clock enable, baud rate selection: fSYS/8
                                 ; LSB first, synchronous clock edge setting: set to Rising

    ld     (hsc0ie),0x08         ; Set to TEND0 interrupt enable
    ld     (inteahsc0),0x10      ; Set INTHSC0 interrupt level to 1
    ei                           ; Interrupt enable (iff=0)

loop                             ;Confirm that transmission data register doesn't have no transmission data
    bit    1,(hsc0st)            ; <RFW0>=1 ?
    jr     z,loop

    ld     (hsc0td),0x3a         ; Write Transmission data and Start transmission
    .
    .
    .
```

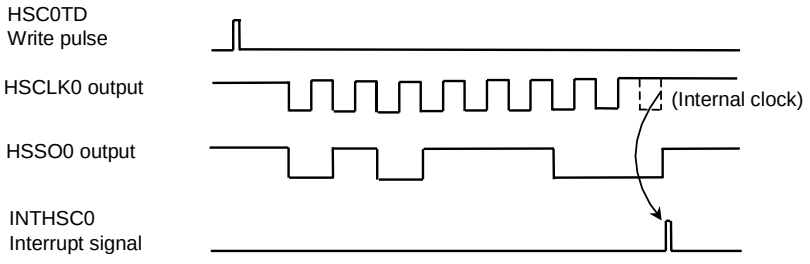


Figure 3.12.41 Example of UNIT transmission

(2) UNIT receiving

This example show case of receiving is executed by following setting, and it is generated INTHSC0 interrupt by finish receiving.

UNIT: 8bit
LSB first
Baud rate selection : $f_{SYS}/8$
Synchronous clock edge: Rising

Setting example

```
ld     (pdfc),0x07              ; Port setting  PD0:HSSI0, PD1:HSSO0, PK7:HSCLK0
ld     (pdcr),0x06              ; Port setting  PD0:HSSI0, PD1:HSSO0, PK7:HSCLK0

ldw    (hsc0ct),0x0040          ; Set data length to 8bit
ldw    (hsc0md),0x2c43          ; System clock enable, baud rate selection : fSYS/8
                                ; LSB first, synchronous clock edge setting: set to Rising

ld     (hsc0ie),0x01            ; Set to RFR0 interrupt enable
ld     (inteahsc0),0x10         ; Set INTHSC0 interrupt level to 1
ei                              ; Interrupt enable (iff=0)

set    0x0,(hsc0ct)             ; Start UNIT receiving
.
.
.
```

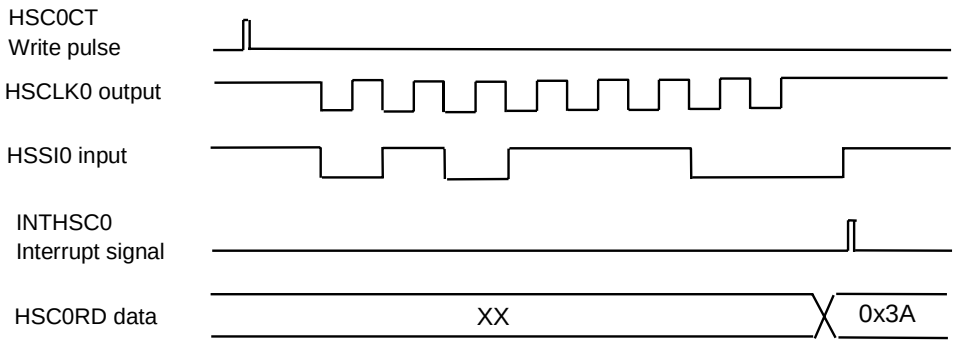


Figure 3.12.42 Example of UNIT receiving

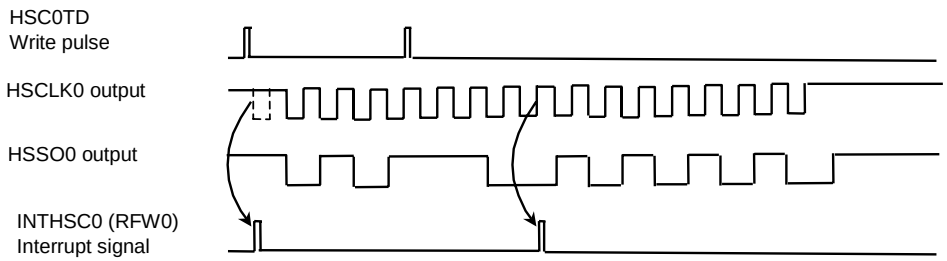
(3) Sequential transmission

This example show case of transmission is executed by following setting, and it is executed 2byte sequential transmission.

UNIT: 8bit
LSB first
Baud rate selection: $f_{SYS}/8$
Synchronous clock edge: Rising

Setting example

```
    ld    (pdfc),0x07          ; Port setting  PD0:HSSI0, PD1:HSSO0, PK7:HSCLK0
    ld    (pdcr),0x06          ; Port setting  PD0:HSSI0, PD1:HSSO0, PK7:HSCLK0

    ldw   (hsc0ct),0x0040      ; Set data length to 8bit
    ldw   (hsc0md),0x2c43      ; System clock enable, baud rate selection: fSYS/8
                               ; LSB first, synchronous clock edge setting: set to Rising

loop1:                         ; Confirm that transmission data register doesn't have no transmission data
    bit   1,(hsc0st)           ; <RFW0>=1 ?
    jr    z,loop1

    ld    (hsc0td),0x3a        ; Write transmission data of first byte and start transmission

loop2                          ; Confirm that transmission data register doesn't have no-transmission data
    bit   1,(hsc0st)           ; <RFW0>=1 ?
    jr    z,loop2

    ld    (hsc0td),0x55        ; Write transmission data of second byte

loop3:                         ; Confirm that transmission data register doesn't have no-transmission data
    bit   3,(hsc0st)           ; <TEND0>=1 ?
    jr    z,loop3
    .                          ; Finish transmission
    .
```



Note: Timing of this figure is an example. There is also that transmission interbal between first byte and sescond byte generate. (High baud rate etc.)

Figure 3.12.43 Example of sequential transmission

(4) Sequential receiving

This example show case of receiving is executed by following setting, and it is executed 2byte sequential receiving.

UNIT: 8bit
LSB first
Baud rate selection: $f_{SYS}/8$
Synchronous clock edge: Rising

Setting example

```
    ld    (pdfc),0x07          ; Port setting PD0:HSSI0, PD1:HSSO0, PD2:HSCLK0
    ld    (pdcr),0x06          ; Port setting PD0:HSSI0, PD1:HSSO0, PD2:HSCLK0

    ldw   (hsc0ct),0x0040      ; Set data length to 8bit
    ldw   (hsc0md),0x2c43      ; System clock enable, baud rate selection: fSYS/8
                               ; LSB first, synchronous clock edge setting: set to Rising

    set   0x01,(hsc0ct)        ; Start sequential receiving

loop1:                         ; Confirm that receiving data register has receiving data of first byte
    bit   0,(hsc0st)           ; <RFR0>=1 ?
    jr    z,loop1

loop2:                         ; Confirm that receiving data register has receiving data of second byte
    bit   2,(hsc0st)           ; <REND0>=1 ?
    jr    z,loop2

    res   0x01,(hsc0ct)        ; Sequential receiving disable

    ld    a,(hsc0rd)           ; Read receiving data of first byte

loop3:                         ; Confirm that receiving data of second byte is shifted from receiving data
                               ; shift register to receiving data register
    bit   0,(hsc0st)           ; <RFR0>=1 ?
    jr    z,loop3
    ld    w,(hsc0rd)           ; Read receiving data of second byte
```

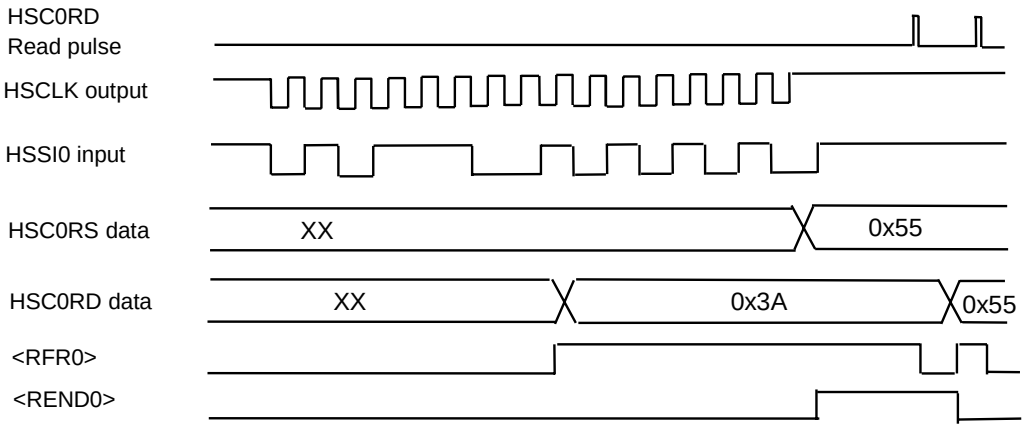


Figure 3.12.44 Example of sequential receiving

(5) Sequeintial Transmission by using micro DMA

This example show case of sequential transmission of 4byte is executed at using micro DMA by following setting.

UNIT: 8bit
LSB first
Baud rate : $f_{SYS}/8$
Synchronous clock edge: Rising

Setting example

Main routine

```
;-- micro DMA setting --
    ld     (dma0v),0x25          ; Set micro DMA0 to INTHSC0
    ld     wa,0x0003             ; Set number of micro DMA transmission  to that number -1 (third time)
    ldc    dmac0,wa
    ld     a,0x08                ; micro DMA mode setting: source INC mode, 1 byte transfer
    ldc    dmam0,a

    ld     xwa,0x806000          ; Set source address
    ldc    dmas0,xwa
    ld     xwa,0xC10             ; Set source address to HSC0TD register
    ldc    dmad0,xwa

;-- SPIC setting --
    ld     (pdfc),0x07           ; Port setting PD0:HSSI0, PD1:HSSO0, PK7:HSCLK0
    ld     (pdcr),0x06           ; Port setting PD0:HSSI0, PD1:HSSO0, PK7:HSCLK0

    ldw    (hsc0ct),0x0040       ; Set data length to 8bit
    ldw    (hsc0md),0x2c43       ; System clock enable, baud rate selection: fSYS/8
                                 ; LSB first, synchronous clock edge setting: set to Rising

    ld     (hsc0ie),0x00         ;Set to interrupt disable
    set    1,(hsc0ct+1)          ; Set micro DMA operation by RFW0 to enable
    ld     (intetc01),0x01       ; Set INTTC0 interrupt level to 1
    ei                           ; Interrupt enable (iff=0)

loop1:                           ; Confirm that transmission data register doesn't have no transmission data
    bit    1,(hsc0st)            ; <RFW0>=1 ?
    jr     z,loop1

    ld     (hsc0td),0x3a         ; Write Transmission data and Start transmission
```

Interrupt routine (INTTC0)

```
loop2:
    bit    1,(hsc0st)            ; <RFW0> = 1 ?
    jr     z,loop2
    bit    3,(hsc0st)            ; <TEND0> = 1 ?
    jr     z,loop2
    nop
```

(6) UNIT receiving by using micro DMA

This example show case of UNIT receiving sequentially 4byte is executed at using micro DMA by following setting.

UNIT: 8bit
LSB first
Baud rate : $f_{SYS}/8$
Synchronous clock edge: Rising

Setting example

Main routine

```
;-- micro DMA setting --
    ld     (dma0v),0x25          ; Set micro DMA0 to INTHSC0
    ld     wa,0x0003             ; Set number of micro DMA transmission to that number -1 (third time)
    ldc    dmac0,wa
    ld     a,0x00                ; micro DMA mode setting: source INC mode, 1 byte transfer
    ldc    dmam0,a

    ld     xwa,0xC12             ; Set source address to HSC0RD register
    ldc    dmas0,xwa
    ld     xwa,0x807000          ; Set source address
    ldc    dmad0,xwa

;-- SPIC setting --
    ld     (pdfc),0x07           ; Port setting PD0:HSSI0, PD1:HSSO0, PD2:HSCLK0
    ld     (pdcr),0x06           ; Port setting PD0:HSSI0, PD1:HSSO0, PD2:HSCLK0

    ldw    (hsc0ct),0x0040       ; Set data length to 8bit
    ldw    (hsc0md),0x2c43       ; System clock enable, baud rate selection: fSYS/8
                                 ; LSB first, synchronous clock edge setting: set to Rising

    ld     (hsc0ie),0x00         ; Set to interrupt disable
    set    0,(hsc0ct+1)          ; Set micro DMA operation by RFR0 to enable
    ld     (intetc01),0x01       ; Set INTTC0 interrupt level to 1
    ei                           ; Interrupt enable (iff=0)

    set    0x0,(hsc0ct)          ; Start UNIT receiving
```

Interrupt routine (INTTC0)

```
loop2:                           ; Wait receiving finish case of UNIT receiving
    bit    0,(hsc0st)            ; <RFR0> = 1 ?
    jr     z,loop2
    res    0,(hsc0ct)            ; UNIT receiving disable
    ld     a,(hsc0rd)            ; Read last receiving data
    nop
```

3.13 SDRAM Controller (SDRAMC)

TMP92CM27 includes SDRAM controller which supports SDRAM access by CPU.
The features are as follows.

(1) Support SDRAM

Data rate type:	Only SDR (Single data rate) type
Bulk of memory:	16/64 Mbits
Number of banks:	2/4 banks
Width of data bus:	16 bit
Read burst length:	1 word/full page
Write mode:	Single/burst

(2) Support Initialize sequence command

All banks precharge command
8 times auto refresh command
Mode Register setting command

(3) Access mode

	CPU Access
Read burst length	1 word/full page
Addressing mode	Sequential
CAS latency (clock)	2
Write mode	Single/burst

(4) Access cycle

CPU Access (Read/write)

Read cycle:	1 word– 4 states/full page – 1 state
Write cycle:	Single – 3 states/burst – 1 state
Data size:	8 bits/16 bits/32 bits

(5) Refresh cycle auto generate

- Auto refresh is generated during except SDRAM access.
- Refresh interval is programmable.
- Self refresh is supported

Note 1: Condition of SDRAM's area set by CS3 setting of memory controller.

3.13.1 Control Registers

Figure 3.13.1 shows the SDRAMC control registers. Setting these registers controls the operation of SDRAMC.

SDRAM Access Control Register 1

SDACR1 (0250H)

	7	6	5	4	3	2	1	0
Bit symbol	–	–	SMRD	SWRC	SBST	SBL1	SBL0	SMAC
Read/Write	R/W							
After reset	0	0	0	0	0	1	0	0
Function	Always write “0”	Always write “0”	Mode register recovery time 0: 1 clock 1: 2 clocks	Write recovery time 0: 1 clock 1: 2 clocks	Burst stop command 0: Precharge all 1: Burst stop	Select burst length (Note 1) 00: Reserved 01: Full-page read, burst write 10: 1-word read, single write 11: Full-page read, single write		SDRAM controller 0: Disable 1: Enable

Note 1: Execute the mode register setting command after changing <SBL1:0>. If change from “full-page read” to “1-word read”, take care setting. Please refer to “3.13.3 4) Limitation point to use SDRAM”.

SDRAM Access Control Register 2

SDACR2 (0251H)

	7	6	5	4	3	2	1	0
Bit symbol				SBS	SDRS1	SDRS0	SMUXW1	SMUXW0
Read/Write				R/W				
After reset				0	0	0	0	0
Function				Number of banks 0: 2 banks 1: 4 banks	Select ROW address size 00: 2048 rows (11 bits) 01: 4096 rows (12 bits) 10: 8192 rows (13 bits) 11: Reserved		Select address multiplex type 00: TypeA (A9-) 01: TypeB (A10-) 10: TypeC (A11-) 11: Reserved	

SDRAM Refresh Control Register

SDRCR (0252H)

	7	6	5	4	3	2	1	0
Bit symbol	–			SSAE	SRS2	SRS1	SRS0	SRC
Read/Write	R/W			R/W				
After reset	0			1	0	0	0	0
Function	Always write “0”.			SR Auto Exit function 0: Disable 1: Enable	Refresh interval 000: 47 states 001: 78 states 010: 97 states 011: 124 states 100: 156 states 101: 195 states 110: 249 states 111: 312 states			Auto refresh 0: Disable 1: Enable

SDRAM Command Register

SDCMM (0253H)

	7	6	5	4	3	2	1	0
Bit symbol						SCMM2	SCMM1	SCMM0
Read/Write						R/W		
After reset						0	0	0
Function						Command executing (Note 1) (Note 2) 000: Not execute 001: Execute initialize command a. Precharge all banks b. 8 times auto refresh c. Set mode register 100: Set mode register 101: Execute self refresh Entry 110: Execute self refresh EXIT Others: Reserved		

Note 1: <SCMM2:0> is cleared to “000” after a command is executed. But <SCMM2:0> is not cleared by executing the self refresh Entry command. It is cleared by executing the self refresh Exit command.

Note 2: When command except the self refresh Exit command is executed, write command after checking that <SCMM2:0> are “000”.

Figure 3.13.1 SDRAM Control Registers

3.13.2 Operation Description

(1) Memory access control

Access controller is enabled when SDACR1<SMAC> = 1. And then SDRAM control signals ($\overline{SDCS}$, $\overline{SDRAS}$, $\overline{SDCAS}$, $\overline{SDWE}$, SDLLDQM, SDLUDQM, SDCLK and SDCKE) are operating during the time CPU accesses CS3 area.

In the access cycle, outputs row/column multiplex address through A0 to A15 pin. And multiplex width is decided by setting SDACR2<SMUXW0:1>. The relation between multiplex width and row/column address is shown in Table.

Table 3.13.1 Address Multiplex

TMP92CM27 Pin Name	Address of SDRAM Access Cycle				
	Row Address			Column Address	
	TypeA <SMUXW> "00"	TypeB <SMUXW> "01"	TypeC <SMUXW> "10"	16-Bit Data Bus Width B1CSH<BnBUS> = "01"	32-Bit Data Bus Width B1CSH<BnBUS> = "10"
A0	A9	A10	A11	A1	A2
A1	A10	A11	A12	A2	A3
A2	A11	A12	A13	A3	A4
A3	A12	A13	A14	A4	A5
A4	A13	A14	A15	A5	A6
A5	A14	A15	A16	A6	A7
A6	A15	A16	A17	A7	A8
A7	A16	A17	A18	A8	A9
A8	A17	A18	A19	A9	A10
A9	A18	A19	A20	A10	A11
A10	A19	A20	A21	AP	AP
A11	A20	A21	A22	Row address	
A12	A21	A22	A23		
A13	A22	A23	EA24		
A14	A23	EA24	EA25		
A15	EA24	EA25	EA26		

Burst length of SDRAM read/write by CPU can be select by setting SDACR1<SBL1:0>.

SDRAM access cycle is shown in Table 3.13.2 and Table 3.13.3.

SDRAM access cycle number is not depending on B3CSL registers setting.

In the full page burst read/write cycle, a mode register set cycle and a precharge cycle are inserted automatically to cycle front and back.

(2) Instruction executing on SDRAM

CPU can be executed instructions that are asserted to SDRAM. However, below function is not operated.

a) Executing HALT instruction

b) Executing instructions that write to SDCMM register

When the above mentioned is operated, it is necessary to execute it by another memory such as built-in RAM.

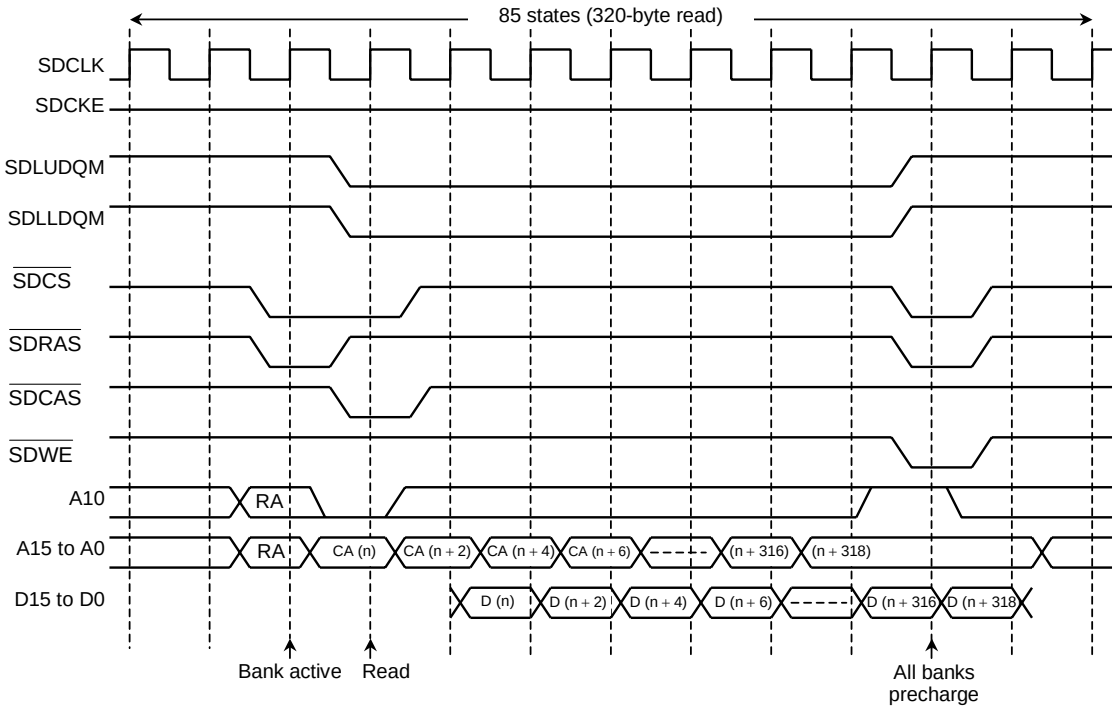


Figure 3.13.2 Timing of Burst Read Cycle

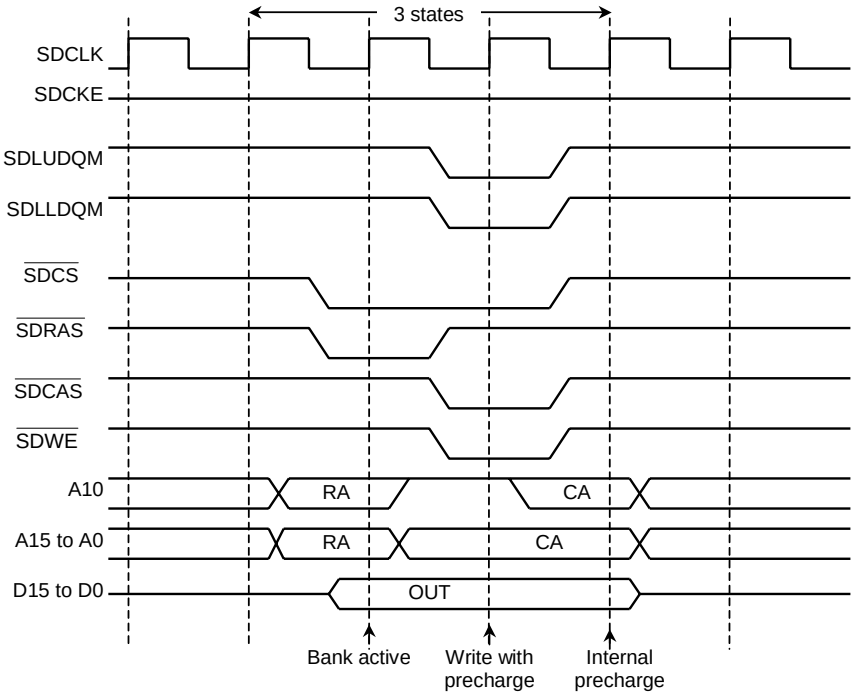


Figure 3.13.3 Timing of CPU Write Cycle

(Structure of Data Bus: 16 bits × 1, operand Size: 2 bytes, address: 2n + 0)

(3) Refresh control

This LSI supports two refresh commands of auto refresh and self refresh.

(a) Auto refresh

The auto refresh command is generated intervals that set to SDRCR<SRS2:0> automatically by setting SDRCR<SRC> to "1". The generation interval can be set between 47 to 312 states (2.4 μs to 15.6 μs at f_{SYS} = 20 MHz).

CPU operation (instruction fetch and execution) stops while performing the auto refresh command. The auto refresh cycle is shown in Figure 3.13.4 and the auto refresh generation interval is shown in Table 3.13.2. Auto self refresh doesn't operate at IDLE1 mode and STOP mode. It can be used only with CPU operation NORMAL mode or IDLE2 mode.

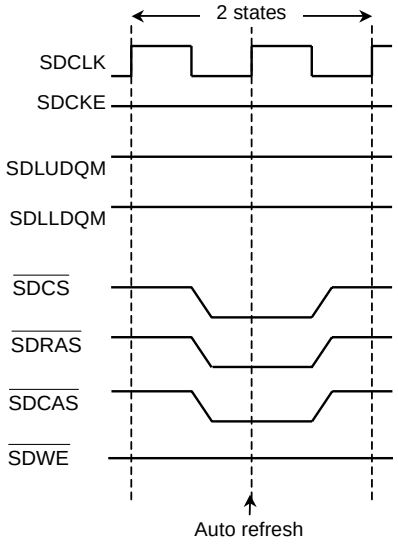


Figure 3.13.4 Timing of Auto Refresh Cycle

Table 3.13.2 Refresh Cycle Insertion Interval (Unit: μs)

SDRCR<SRS2:0>			Insertion Interval (State)	f_{SYS} Frequency (System clock)					
SRS2	SRS1	SRS0		6 MHz	10 MHz	12.5 MHz	15 MHz	17.5 MHz	20 MHz
0	0	0	47	7.8	4.7	3.8	3.1	2.7	2.4
0	0	1	78	13.0	7.8	6.2	5.2	4.5	3.9
0	1	0	97	16.2	9.7	7.8	6.5	5.5	4.9
0	1	1	124	20.7	12.4	9.9	8.3	7.1	6.2
1	0	0	156	26.0	15.6	12.5	10.4	8.9	7.8
1	0	1	195	32.5	19.5	15.6	13.0	11.1	9.8
1	1	0	249	41.5	24.9	19.9	16.6	14.2	12.4
1	1	1	312	52.0	31.2	25.0	20.8	17.8	15.6

(b) Self refresh

The self refresh command is generated by making it to SDCMM<SCMM2:0> to "101". The self refresh cycle is shown in Figure 3.13.5. During self refresh Entry, refresh is performed inside SDRAM (an auto refresh command is not needed).

Note 1: When stand-by mode is cancelled by a reset, the I/O registers are initialized, therefore, auto refresh is not performed.

Note 2: During self refresh Entry, it cannot be accessed to SDRAM.

Note 3: After the self refresh Entry command, shift CPU to IDLE1 or STOP mode. When during setting HALT instruction and SDCMM <SCMM2:0> to "101", execute NOP (more than 10 bytes) or another instructions.

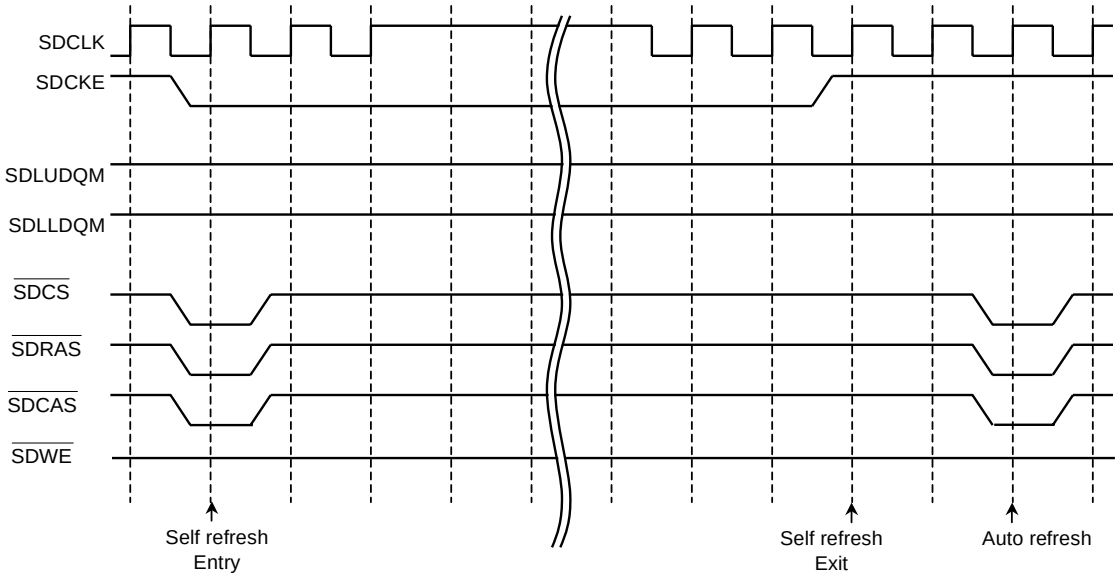


Figure 3.13.5 Timing of Self Refresh Cycle

Self-Refresh condition is released by executing Serf-Refresh command. Way to execute Self-Refresh EXIT command is 2 ways: write "110" to SDCMM<SCMM2:0>, or execute EXIT automatically by synchronizing to releasing HALT condition. Both ways, after it executes Auto-Refresh at once just after Self-Refresh EXIT, it executes Auto-Refresh at setting condition. When it became EXIT by writing "110" to <SCMM2:0>, <SCMM2:0> is cleared to "000".

EXIT command that synchronize to release HALT condition can be prohibited by setting SDRCR<SSAE> to "0". If don't set to EXIT automatically, set to prohibit. If using condition of SDRAM is satisfied by operation clock frequency (clock gear down, SLOW mode condition and so on) is falling, set to prohibit. Figure 3.13.6 shows execution flow in this case.

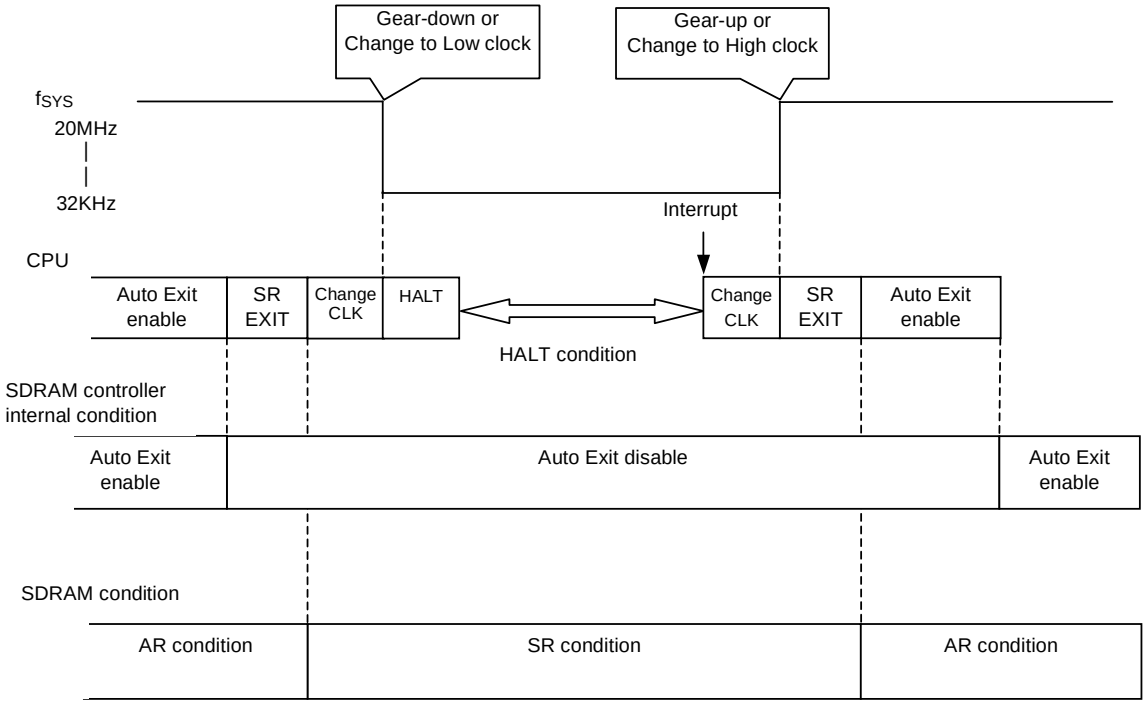


Figure 3.13.6 Execution flow example (Execute HALT instruction at low-speed clock).

```
; ********Sample program *********
LOOP1:
        LDB     A, (SDCMM)                  ;   Check the command register clear
        ANDB    A, 00000111B                ;
        J       NZ, LOOP1                   ;


        LDW     (SDRCR), 0000010100000011B  ;   Auto Exit disable→ Self refresh Entry

        NOP×10                              ;   Wait Self refresh Entry command executing
        LD      (SYSCR1), XXXXX001B         ;   fc/2
        HALT
        NOP                                 ;   Self refresh Exit (Internal signal only)


        LD      (SYSCR1), XXXXX000B         ;   fc
        LD      (SDCMM), 00000110B          ;   Self refresh Exit (command)
        LD      (SDRCR), 0001---1B          ;   Auto Exit enable
```

(4) SDRAM initialize

After released reset, it can generate the following cycle that is needed to SDRAM. The cycle is shown in Figure 3.13.7.

1. Precharge all banks
2. The auto refresh cycle of 8 cycles
3. Set a Mode register

The above cycle is generated by setting SDCMM<SCMM2:0> to "001".

While performing this cycle, operation (an instruction fetch, command execution) of CPU is stopped.

In addition, before execute an initialization cycle, set port as SDRAM control signal and an address signal (A0 to A15).

After the initialization cycle was finished, SDCMM<SCMM2:0> is set to "000" automatically.

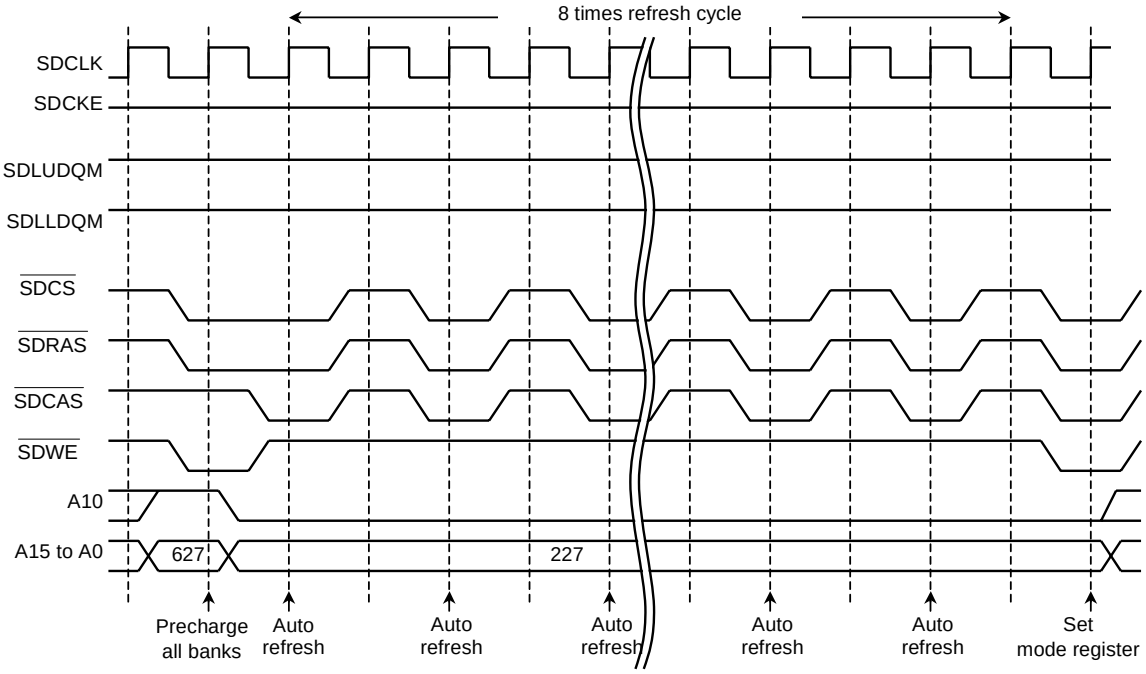


Figure 3.13.7 Timing of Initialization Cycle

(5) Connection example

The example of connection with SDRAM is shown in Table 3.13.3 and Figure 3.13.8.

Table 3.13.3 Connection with SDRAM

TMP92CM27 Pin Name	SDRAM Pin Name Data Bus Width: 16 Bits				
	16 M	64 M	128 M	256 M	512 M
A0	A0	A0	A0	A0	A0
A1	A1	A1	A1	A1	A1
A2	A2	A2	A2	A2	A2
A3	A3	A3	A3	A3	A3
A4	A4	A4	A4	A4	A4
A5	A5	A5	A5	A5	A5
A6	A6	A6	A6	A6	A6
A7	A7	A7	A7	A7	A7
A8	A8	A8	A8	A8	A8
A9	A9	A9	A9	A9	A9
A10	A10	A10	A10	A10	A10
A11	BS	A11	A11	A11	A11
A12	–	BS0	BS0	A12	A12
A13	–	BS1	BS1	BS0	BS0
A14	–	–	–	BS1	BS1
A15	–	–	–	–	–
$\overline{\text{SDCS}}$	CS	CS	CS	CS	CS
SDLUDQM	UDQM	UDQM	UDQM	UDQM	UDQM
SDLLDQM	LDQM	LDQM	LDQM	LDQM	LDQM
$\overline{\text{SDRAS}}$	RAS	RAS	RAS	RAS	RAS
$\overline{\text{SDCAS}}$	CAS	CAS	CAS	CAS	CAS
$\overline{\text{SDWE}}$	WE	WE	WE	WE	WE
SDCKE	CKE	CKE	CKE	CKE	CKE
SDCLK	CLK	CLK	CLK	CLK	CLK
SDACR <SMUXW>	00: TypeA	00: TypeA	01: TypeB	01: TypeB	10: TypeC

(An) : Row address

: Command address pin of SDRAM

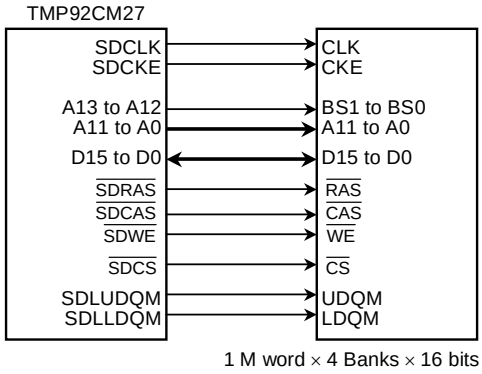


Figure 3.13.8 Connection with SDRAM (4 M word × 16 bits)

3.13.3 Limitation point to use SDRAM

There are some points to notice when using SDRAMC. Please refer to the section under below and please be careful.

1. WAIT access

   When it uses SDRAM, some limitation is added if it access to memory except SDRAM. In N-WAIT setting of this LSI, if setting time is inserted as external WAIT, set time less than Auto Refresh cycle (Auto Refresh function that is controlled by SDRAM controller) × 8190.

2. Execution of SDRAM command before HALT instruction (SR (Self refresh)-Entry, Initialize, Mode-set)

   When command that SDRAM controller has (SR-Entry, Initialize and Mode-set) is executed, execution time is needed few states.

   Therefore, when HALT instruction is executed after the SDRAM command, please insert NOP more than 10 bytes or other 10 instructions before executing HALT instruction.

3. AR (Auto Refresh) interval time

   When using SDRAM, set CPU clock that satisfy minimum operation frequency for SDRAM and minimum refresh cycle.

   When SLOW mode is used by using SDRAM or it use system that clock gear may become down, consider AR cycle for SDRAM.

   When AR cycle is changed, set to disable by writing "0" to SDRCR<SRC>.

4. Note of when changing access mode

   If changing access mode from "full page read" to "1 word read", execute following program. This program must not execute on the SDRAM.

```
di                                          ; Interrupt Disable (Added)
ld      a,(optional external memory address) ; Dummy read instruction (Added)
ld      (sdacr1),00001101b                  ; Change to "1-word read"
ld      (sdcmm),0x04                        ; Execute MRS (mode register setting)
ei                                          ; Interrupt enable (Added)
```

3.14 Analog/Digital Converter

The TMP92CM27 incorporates a 10-bit successive approximation-type analog/digital converter (AD converter) with 12-channel analog input.

Figure 3.14.1 is a block diagram of the AD converter. The 12-channel analog input pins (AN0 to AN11) are shared with the input-only port M and port N so they can be used as an input port.

Note: When IDLE2, IDLE1, or STOP mode is selected, as to reduce the power, with some timings the system may enter a standby mode even though the internal comparator is still enabled. Therefore be sure to check that AD converter operations are halted before a HALT instruction is executed.

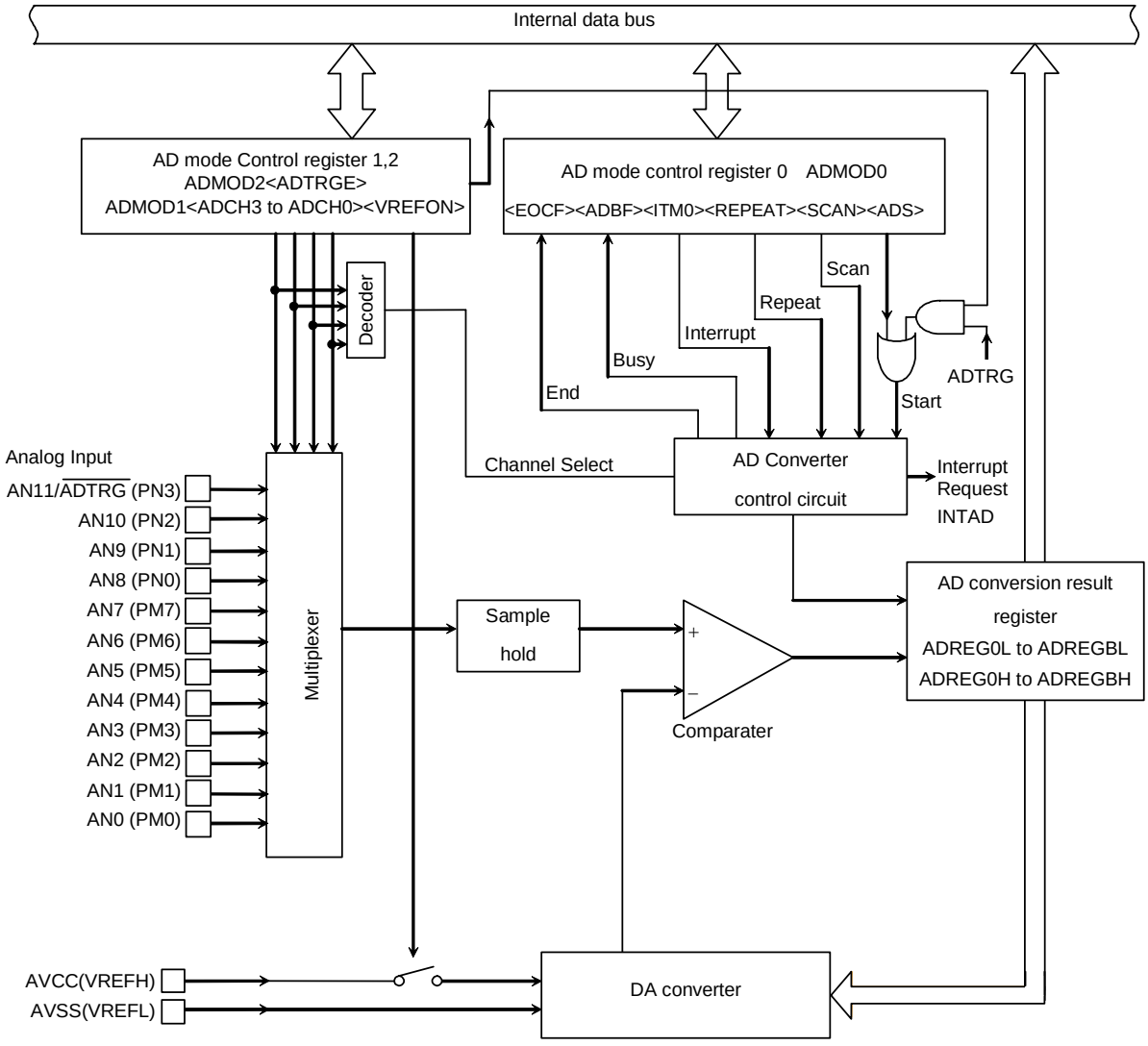


Figure 3.14.1 Block Diagram of AD Converter

3.14.1 Analog/Digital Converter Registers

The AD converter is controlled by the three AD mode control registers: ADMOD0, ADMOD1, and ADMOD2. The 24 AD conversion data result registers (ADREG0H/L to ADREGBH/L) store the results of AD conversion.

Figure 3.14.2 to Figure 3.14.6 shows the registers related to the AD converter.

AD Mode Control Register 0

ADMOD0 (12B8H)

	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	AD conversion busy flag 0: Conversion stopped 1: Conversion in progress	Always write "0".	Always write "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 in progress

0	Before or during AD conversion
1	AD conversion complete

Figure 3.14.2 Register for AD Converter (1)

AD Mode Control Register 1

		7	6	5	4	3	2	1	0
ADMOD1 (12B9H)	bit Symbol	VREFON	I2AD	−	−	ADCH3	ADCH2	ADCH1	ADCH0
	Read/Write	R/W	R/W	R/W		R/W			
	リセット後	0	0	0	0	0	0	0	0
	Function	VREF application control 0: OFF 1: ON	IDLE2 0: Stop 1: Operate	Always write "0".	Always write "0".	Analog input channel selection			

Analog input channel selection

ADMOD0<SCAN> / <ADCH3:0>	0 (Channel fixed)	1 (Channel scanned)
0000	AN0	AN0
0001	AN1	AN0 → AN1
0010	AN2	AN0 → AN1 → AN2
0011	AN3	AN0 → AN1 → AN2 → AN3
0100	AN4	AN0 → AN1 → AN2 → AN3 → AN4
0101	AN5	AN0 → AN1 → AN2 → AN3 → AN4 → AN5
0110	AN6	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6
0111	AN7	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7
1000	AN8	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8
1001	AN9	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8 → AN9
1010	AN10	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8 → AN9 → AN10
1011	AN11	AN0 → AN1 → AN2 → AN3 → AN4 → AN5 → AN6 → AN7 → AN8 → AN9 → AN10 → AN11
1100～1111	Please do not set up.	

IDLE2 control

0	Stop
1	Operate

Control of application of reference voltage to AD converter

0	OFF
1	ON

Note: As pin AN11 also functions as the $\overline{\text{ADTRG}}$ input pin, do not set ADMOD1<ADCH3:0> ="1011" when using $\overline{\text{ADTRG}}$ with ADMOD2<ADTRGE> set to "1".

Figure 3.14.3 Register for AD Converter (2)

AD Mode Control Register 2

		7	6	5	4	3	2	1	0
ADMOD2 (12BAH)	Bit symbol	–	–	–	–	–	–	–	ADTRGE
	Read/Write								R/W
	After reset	0							0
	Function	All ways write “0”.							AD conversion trigger start control 0: Disable 1. Enable

AD conversion start control by external trigger ($\overline{\text{ADTRG}}$ input)

0	Disabled
1	Enabled

Figure 3.14.4 Register for AD Converter (3)

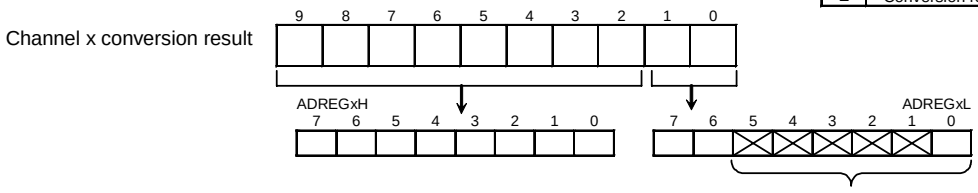
AD conversion result register Low

		7	6	5	4	3	2	1	0
ADREG0L (12A0H)	bit Symbol	ADR01	ADR00						ADR0RF
	Read/Write	R							R
	After reset	Undefined							0
ADREG1L (12A2H)	bit Symbol	ADR11	ADR10						ADR1RF
	Read/Write	R							R
	After reset	Undefined							0
ADREG2L (12A4H)	bit Symbol	ADR21	ADR20						ADR2RF
	Read/Write	R							R
	After reset	Undefined							0
ADREG3L (12A6H)	bit Symbol	ADR31	ADR30						ADR3RF
	Read/Write	R							R
	After reset	Undefined							0
ADREG4L (12A8H)	bit Symbol	ADR41	ADR40						ADR4RF
	Read/Write	R							R
	After reset	Undefined							0
ADREG5L (12AAH)	bit Symbol	ADR51	ADR50						ADR5RF
	Read/Write	R							R
	After reset	Undefined							0
ADREG6L (12ACH)	bit Symbol	ADR61	ADR60						ADR6RF
	Read/Write	R							R
	After reset	Undefined							0
ADREG7L (12AEH)	bit Symbol	ADR71	ADR70						ADR7RF
	Read/Write	R							R
	After reset	Undefined							0
ADREG8L (12B0H)	bit Symbol	ADR81	ADR80						ADR8RF
	Read/Write	R							R
	After reset	Undefined							0
ADREG9L (12B2H)	bit Symbol	ADR91	ADR90						ADR9RF
	Read/Write	R							R
	After reset	Undefined							0
ADREGAL (12B4H)	bit Symbol	ADRA1	ADRA0						ADRARF
	Read/Write	R							R
	After reset	Undefined							0
ADREGBL (12B6H)	bit Symbol	ADRB1	ADRB0						ADRBRF
	Read/Write	R							R
	After reset	Undefined							0

Stores lower 2 bits of AD conversion result

AD conversion data storage flag

1	Conversion result stored



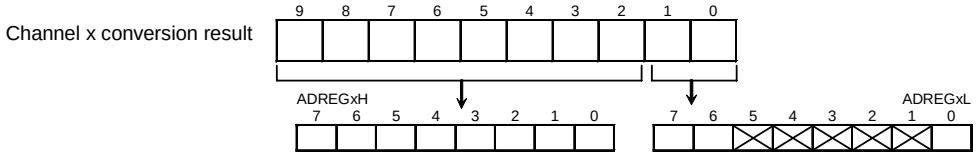
- Bits 5 to 1 are always read as 1.
- Bit0 is the AD conversion data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.144.5 Register for AD Converter (4)

AD conversion result register High

		7	6	5	4	3	2	1	0
ADREG0H (12A1H)	bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
	Read/Write	R							
	After reset	Undefined							
ADREG1H (12A3H)	bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
	Read/Write	R							
	After reset	Undefined							
ADREG2H (12A5H)	bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
	Read/Write	R							
	After reset	Undefined							
ADREG3H (12A7H)	bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
	Read/Write	R							
	After reset	Undefined							
ADREG4H (12A9H)	bit Symbol	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
	Read/Write	R							
	After reset	Undefined							
ADREG5H (12ABH)	bit Symbol	ADR59	ADR58	ADR57	ADR56	ADR55	ADR54	ADR53	ADR52
	Read/Write	R							
	After reset	Undefined							
ADREG6H (12ADH)	bit Symbol	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
	Read/Write	R							
	After reset	Undefined							
ADREG7H (12AFH)	bit Symbol	ADR79	ADR78	ADR77	ADR76	ADR75	ADR74	ADR73	ADR72
	Read/Write	R							
	After reset	Undefined							
ADREG8H (12B1H)	bit Symbol	ADR89	ADR88	ADR87	ADR86	ADR85	ADR84	ADR83	ADR82
	Read/Write	R							
	After reset	Undefined							
ADREG9H (12B3H)	bit Symbol	ADR99	ADR98	ADR97	ADR96	ADR95	ADR94	ADR93	ADR92
	Read/Write	R							
	After reset	Undefined							
ADREGAH (12B5H)	bit Symbol	ADRA9	ADRA8	ADRA7	ADRA6	ADRA5	ADRA4	ADRA3	ADRA2
	Read/Write	R							
	After reset	Undefined							
ADREGBH (12B7H)	bit Symbol	ADRB9	ADRB8	ADRB7	ADRB6	ADRB5	ADRB4	ADRB3	ADRB2
	Read/Write	R							
	After reset	Undefined							

Stores Higher 8 bits of AD conversion result



- Bits 5 to 1 are always read as 1.

- Bit0 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 3.14.6 Register for AD Converter (5)

3.14.2 Description of Operation

(1) Analog reference voltage

A high-level analog reference voltage is applied to the AVCC pin; a low-level analog reference voltage is applied to the AVSS pin. To perform AD conversion, the reference voltage, the difference between AVCC and AVSS, 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 AVCC and AVSS, program 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 f_c), 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 to AN11 as the input channel.
- In analog input channel scan mode (ADMOD0<SCAN> = 1)
  Setting ADMOD1<ADCH3:0> selects one of the 12 scan modes.

Table 3.14.1 illustrates analog input channel selection in each operation mode.

On a reset, ADMOD0<SCAN> is cleared to "0" and ADMOD1<ADCH3:0> 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 3.14.1 Analog Input Channel Selection

<ADCH3 to 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, program "1" to ADMOD0<ADS> in AD mode control register 0, or ADMOD2<ADTRGE> in AD mode control register 1 and input falling edge on $\overline{\text{ADTRG}}$ pin. 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<REPEAT> and ADMOD0<SCAN> settings in AD mode control register 0 determine the AD mode setting.

Completion of AD conversion triggers an INTAD conversion end interrupt request. Also, ADMOD0<EOCF> will be set to "1" to indicate that AD conversion has been completed.

a. Channel fixed single conversion mode

Setting ADMOD0<REPEAT> 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.

b. Channel scan single conversion mode

Setting ADMOD0<REPEAT> 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.

c. Channel fixed repeat conversion mode

Setting ADMOD0<REPEAT> 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>.

Clearing <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.

d. Channel scan repeat conversion mode

Setting ADMOD0<REPEAT> 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 (e.g., in cases c and d), program a "0” to ADMOD0<REPEAT>. 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 (e.g., in cases c and d), when the halt is released, conversion restarts from the beginning. In single conversion modes (e.g., in cases a and b), conversion does not restart when the halt is released (The converter remains stopped).

Table 3.14.2 shows the relationship between the AD conversion modes and interrupt requests.

Table 3.14.2 Relationship between the AD Conversion Modes and Interrupt Requests AD

Mode	Interrupt Request Generation	ADMOD0		
		<ITM0>	<REPEAT>	<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

99 states (4.95 μs at f_{sys} = 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 to AN11 conversion results are stored in ADREG0H/L to ADREGBH/L respectively.

Table 3.14.3 shows the correspondence between the analog input channels and the registers which are used to hold the results of AD conversion.

Table 3.14.3 Correspondence between Analog Input Channel and AD Conversion Result Register

Analog Input Channel (Port G / Port L)	AD Conversion Result Register	
	Conversion Modes Other than at Right	Channel Fixed Repeat Conversion Mode (ADMOD0<ITM0>= "1")
AN0	ADREG0H/L	ADREG0H/L → ADREG1H/L → ADREG2H/L → ADREG3H/L → ADREG0H/L
AN1	ADREG1H/L	
AN2	ADREG2H/L	
AN3	ADREG3H/L	
AN4	ADREG4H/L	
AN5	ADREG5H/L	
AN6	ADREG6H/L	
AN7	ADREG7H/L	
AN8	ADREG8H/L	
AN9	ADREG9H/L	
AN10	ADREGAH/L	
AN11	ADREGBH/L	

<ADRxRF>, bit0 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".

Example:

1. Convert the analog input voltage on the AN3 pin and write the result, to memory address 2800H using the AD interrupt (INTAD) processing routine.

Setting of main routine

		7	6	5	4	3	2	1	0	
INTEPAD	←	X	–	–	–	X	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 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	←	ADREG3H/L	Read value of ADREG3L, ADREG3H to general purpose register WA (16 bits).
WA	> > 6		Shift contents read into WA six times to right and zero-fill upper bits.
(2800H)	←	WA	Write contents of WA to memory address 2800H.

2. Converts repeatedly the analog input voltages on the three pins AN0, AN1, and AN2, using channel scan repeat conversion mode.

		7	6	5	4	3	2	1	0	
INTEPAD	←	X	–	–	–	X	0	0	0	Disable INTAD.
ADMOD1	←	1	1	0	0	0	0	1	0	Set pins AN0 to AN2 to be the analog input channels.
ADMOD0	←	X	X	0	0	0	1	1	1	Start conversion in channel scan repeat conversion mode.

X : Don't care, – : No change

3.15 Digital/Analog Converter

8-bit resolution D/A converter of 2 channels is built into and it has the following features.

- 8-bit resolution D/A converter with two internal channels.
- A full range Buffer AMP is built in each channel.
- The standby can be set to each channel by the control register.

3.15.1 Operation

Control register 0 DACnCNT0<OPn><REFONn> is set to "11". Output CODE is set to output register DACnREG. And, the output voltage corresponding to CODE appears to output pin DAOUTn by doing "1" to Control register 1 DACnCNT1<VALIDn> in write. When <VALIDn> is not set, the value of the output register is not reflected in DAOUTn. Therefore, set DACnCNT1<VALIDn> after the data of eight bits is updated without fail in DACnREG when you renew CODE. When "1" is written to <VALIDn>, the data of DACnREG takes in to a DA converter as 8 bit data, and recognizes as CODE. Moreover, DACnCNT0<OPn> output DAOUTn becomes High-Z by setting it as "0". Iref is cut by setting DACnCNT0<REFONn > to "0", and current consumption can be reduced. The setting of DACnCNT0<OPn><REFONn > is needed before the HALT instruction is executed because the output voltage corresponding to CODE is output from output terminal DAOUTn after the HALT instruction is executed.

FigureFigure 3.15.1 is block diagram if the D/A converter.

Note: From DAOUTn, "1" is outputted from from immediately after setting DACnCNT0 <OPn> as "1."
Then, the value set up by DACnREG is outputted from DAOUTn.

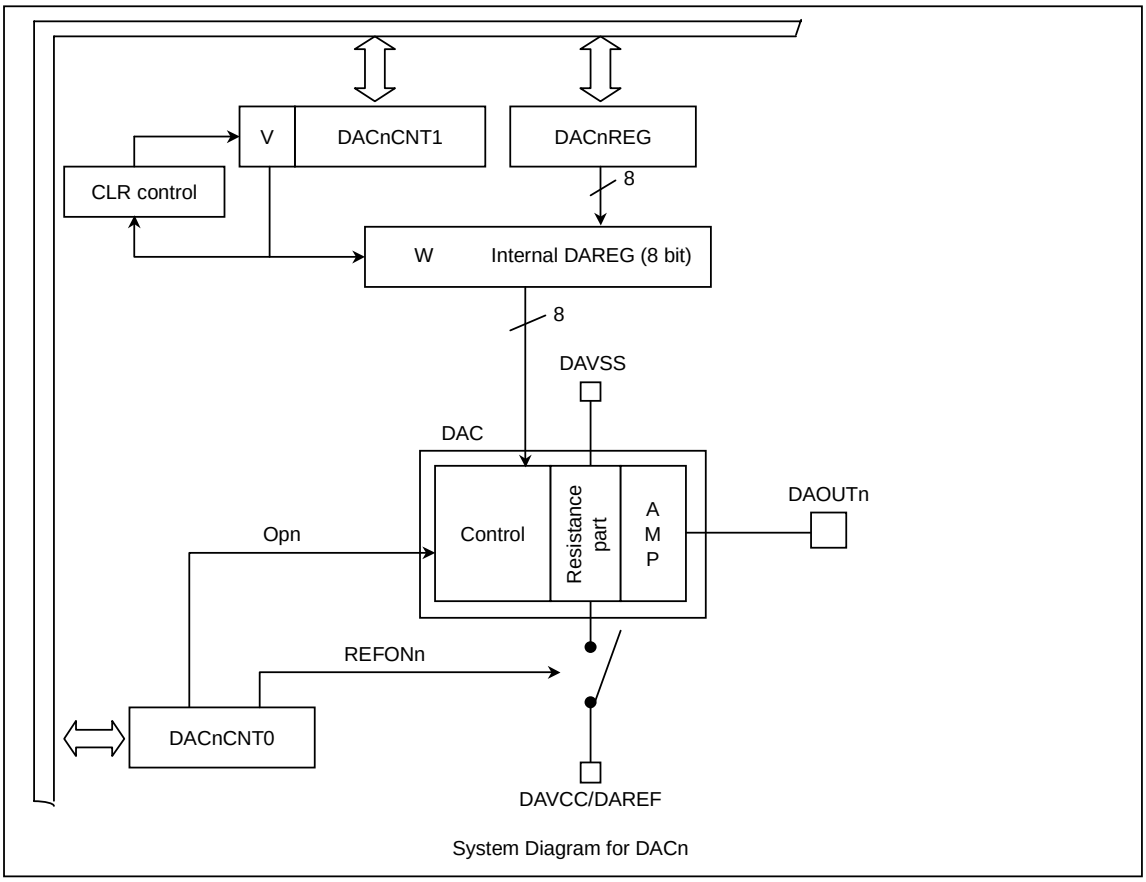


Figure 3.15.1 D/A Converter Block Diagram

Control register 0 DAC0CNT0 register

		7	6	5	4	3	2	1	0
DAC0CNT0 (12E3H)	Bit Symbol							REFON0	OP0
	Read/Write							R/W	R/W
	After reset							0	0
	Function							0: Ref off 1: Ref on	0: Output High-Z 1: Output

Control register 0 DAC1CNT0 register

		7	6	5	4	3	2	1	0
DAC1CNT0 (12E7H)	Bit Symbol							REFON1	OP1
	Read/Write							R/W	R/W
	After reset							0	0
	Function							0: Ref off 1: Ref on	0: Output High-Z 1: Output

Control register 1 DAC0CNT1

DAC0CNT1 (12E1H)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	–				VALID0
Read/Write	R/W	R/W	R/W	R/W				W
After reset	0	0	0	0				0
Function	Always write “0”	Always write “0”	Always write “0”	Always write “0”				0: Don’t care 1: Output CODE valid

Output register DAC0REG

DAC0REG (12E0H)

	7	6	5	4	3	2	1	0
Bit Symbol	DAC07	DAC06	DAC05	DAC04	DAC03	DAC02	DAC01	DAC00
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function								

Note: Write digital data and VALID in order of DAC0REG → DAC0CNT1.

Control register 1 DAC1CNT1

DAC1CNT1 (12E5H)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	–				VALID1
Read/Write	R/W	R/W	R/W	R/W				W
After reset	0	0	0	0				0
Function	Always write “0”	Always write “0”	Always write “0”	Always write “0”				0: Don’t care 1: Output CODE valid

Output register DAC1REG

DAC1REG (12E4H)

	7	6	5	4	3	2	1	0
Bit Symbol	DAC17	DAC16	DAC15	DAC14	DAC13	DAC12	DAC11	DAC10
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function								

Note: Write digital data and VALID in order of DAC1REG → DAC1CNT1.

3.16 Watchdog Timer (Runaway detection timer)

The TMP92CM27 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, and outputs "0" from the watchdog timer out pin $\overline{\text{WDTOUT}}$ to notify peripheral devices of the malfunction.

Connecting the watchdog timer output to the reset pin internally forces a reset.

3.16.1 Configuration

Figure 3.16.1 is a block diagram of the watchdog timer (WDT).

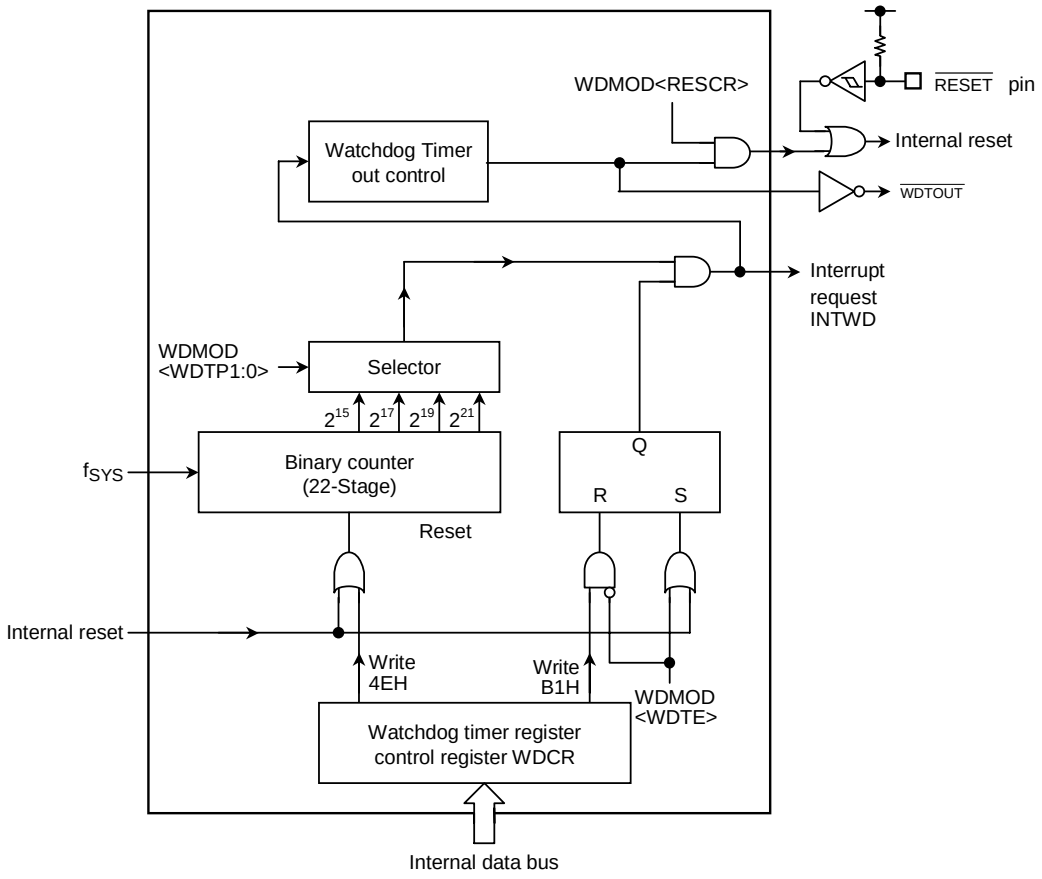


Figure 3.16.1 Block Diagram of Watchdog Timer

The watchdog timer consists of a 22-stage binary counter which uses the clock f_{SYS} as the input clock. The binary counter can output $2^{15}/f_{SYS}$, $2^{17}/f_{SYS}$, $2^{19}/f_{SYS}$, and $2^{21}/f_{SYS}$. Selecting one of the outputs using WDMOD<WDTP1:0> generates a watchdog timer interrupt and output watchdog timer out when an overflow generate as shown in Figure 3.16.2.

Since the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs "0" when there is a watchdog timer overflow, the peripheral devices can be reset. Clearing the watchdog timer (writing the clear code (4EH) to the WDCR register) sets the WDTOUT pin to "1". In normal mode, the $\overline{\text{WDTOUT}}$ pin continually outputs "0" until the clear code is written to the WDCR register.

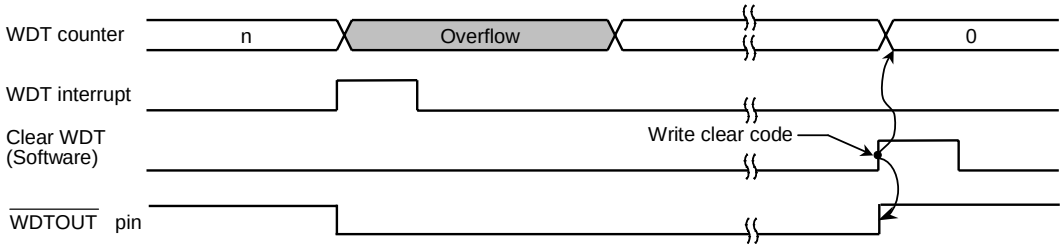


Figure 3.16.2 Normal Mode

The runaway detection result can also be connected to the reset pin internally.
In this case, the reset time will be between 22 to 29 system clocks (2.2 to 2.9 μs at f_{SYS} = 20 MHz) as shown in Figure 3.16.3.

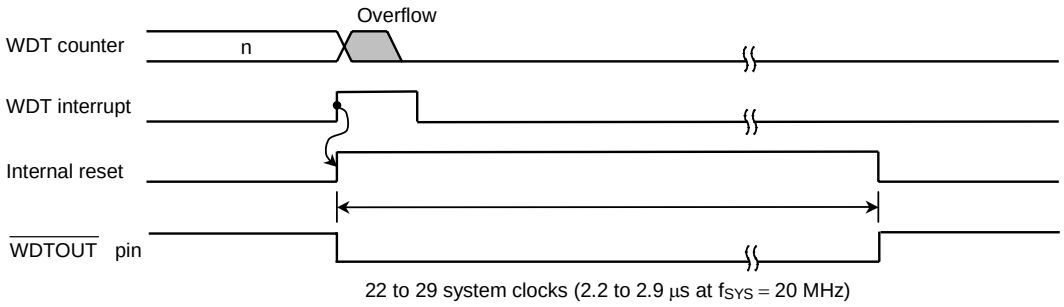


Figure 3.16.3 Reset Mode

3.16.2 Control Registers

The watchdog timer WDT is controlled by two control registers WDMOD and WDCR .

(1) Watchdog timer mode register (WDMOD)

1. Setting the detection time for the watchdog timer in <WDTP1:0>

   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<WDTP1:0> = "00".

   The detection time of the watch dog timer is shown in Figure 3.16.4.

2. Watchdog timer enable/disable control register <WDTE>

   At reset, the WDMOD<WDTE> is initialized to 1, enabling the watchdog timer.

   To disable the watchdog timer, it is necessary to clear this bit to 0 and to write the disable code (B1H) to the watchdog timer control register WDCR. This makes it difficult for the watchdog timer to be disabled by runaway.

   However, it is possible to return the watchdog timer from the disabled state to the enabled state merely by setting <WDTE> to "1".

3. Watchdog timer out reset connection <RESCR>

   This register is used to connect the output of the watchdog timer with the RESET terminal internally. Since WDMOD<RESCR> is initialized to "0" at reset, a reset by the watchdog timer will not be performed.

(2) Watchdog timer control register (WDCR)

This register is used to disable and clear the binary counter for the watchdog timer.

- Disable control

The watchdog timer can be disabled by clearing WDMOD<WDTE> to 0 and then writing the disable code (B1H) to the WDCR register.

```
WDMOD  ← 0 – – – – – – –   Clear WDMOD<WDTE> to "0".
WDCR   ← 1 0 1 1 0 0 0 1   Write the disable code (B1H).
```

- Enable control

Set WDMOD<WDTE> to "1".

- Watchdog timer clear control

To clear the binary counter and cause counting to resume, write the clear code (4EH) to the WDCR register.

```
WDCR   ← 0 1 0 0 1 1 1 0   Write the clear code (4EH).
```

WDMOD (1300H)

	7	6	5	4	3	2	1	0
Bit symbol	WDTE	WDTP1	WDTP0	–	–	I2WDT	RESCR	–
Read/Write	R/W					R/W		
After reset	1	0	0	0	0	0	0	0
Function	WDT control 0: Disable 1: Enable	WDT detection time 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Always write “0”.	Always write “0”.	IDLE2 0: Stop 1: Operation	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 at fc = 40MHz

Gear <GEAR2:0>	Watch Dog Timer detection time WDMOD<WDTP1:0>			
	00	01	10	11
000 (fc)	1.638 ms	6.554 ms	26.214 ms	104.857 ms
001 ($fc_{/2}$)	3.277 ms	13.107 ms	52.429 ms	209.715 ms
010 ($fc_{/4}$)	6.554 ms	26.214 ms	104.857 ms	419.430 ms
011 ($fc_{/8}$)	13.107 ms	52.429 ms	209.715ms	838.860 ms
100 ($fc_{/16}$)	26.214 ms	104.857 ms	419.430 ms	1.677 s

Watchdog timer enable/disable control

0	Disable
1	Enable

Figure 3.16.4 Watchdog Timer Mode Register

WDCR (1301H)

Read-modify-write instruction is prohibited

	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 3.16.5 Watchdog Timer Control Register

3.16.3 Operation

After the detection time set by the WDMOD<WDTP1:0> register is reached, the watchdog timer generates interrupt INTWD and outputs a low signal to the watchdog timer out pin $\overline{\text{WDTOUT}}$. The binary counter for the watchdog timer must be cleared to 0 by software (Instruction) before INTWD is generated. If the CPU malfunctions (Runaway) due to causes such as noise and does not execute an instruction to clear the binary counter, the binary counter overflows and generates INTWD.

The CPU interprets INTWD as a malfunction detection signal, which can be used to start the malfunction recovery program to return the system to normal. A CPU malfunction can also be fixed by connecting the watchdog timer output to a reset pin for peripheral devices.

The watchdog timer begins operating immediately on release of the watchdog timer reset.

The watchdog timer is reset and halted in IDLE1 or STOP modes. The watchdog counter continues counting during bus release ($\overline{\text{BUSAK}}$ = Low).

When the device is in IDLE2 mode, the operation of WDT depends on the WDMOD<I2WDT> setting. Ensure that WDMOD<I2WDT> is set before the device enters IDLE2 mode.

Example: 1. Clear the binary counter.

```
WDCR    ← 0 1 0 0 1 1 1 0        Write the clear code (4EH).
```

2. Set the watchdog timer detection time to $2^{17}/f_{SYS}$.

```
WDMOD   ← 1 0 1 X 0 – – –
```

3. Disable the watchdog timer.

```
WDMOD   ← 0 – – X 0 – – –        Clear <WDTE> bit to 0.
WDCR    ← 1 0 1 1 0 0 0 1        Write the disable code (B1H).
```

3.17 External bus release function

TMP92CM27 have external bus release function that can connect bus master to external. Bus release request ($\overline{\text{BUSRQ}}$), bus release answer ($\overline{\text{BUSAK}}$) pin is assigned to Port 86 and 87. And, it become effective by setting to P8CR and P8FC.

Figure 3.17.1 shows operation timing. Time that from $\overline{\text{BUSRQ}}$ pin inputted "0" until busis released ($\overline{\text{BUSAK}}$ is set to "0") depend on instruction that CPU execute at that time.

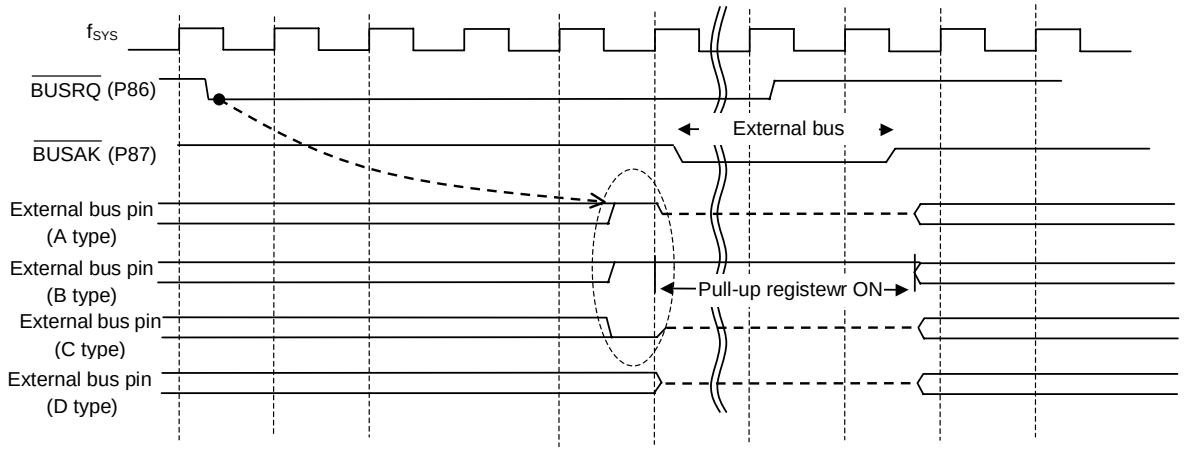


Figure 3.17.1 Bus release function operation timing

3.17.1 Non release pin

If it received bus release request, CPU release bus to external by setting $\overline{\text{BUSAK}}$ pin to "0" without start next bus. In this case, pin that is released have 4 types (A, B, C and D). Eve operation that set to high impedance (HZ) is different in 4 types.(Note) Table 3.17.1 shows support pin for 4 types. Any pin become non release pin only case of setting to that function by setting port. Therefore, if pin set to output port and so on, it is not set non relase pin, and it hold previous condition.

Table 3.17.1 Non release pin

Type	Eve operation that set to HZ	Support function (Pin name)
A	Drive "1"	A23 to A16(P67 to P60), A15 to A8, A7 to A0, $\overline{\text{CS0}}$ (P80), $\overline{\text{CS1}}$ (P81), $\overline{\text{CS2}}$ (P82), $\overline{\text{CS3}}$ (P83), $\overline{\text{SDCS}}$ (P83), $\overline{\text{CS4}}$ (P84), $\overline{\text{CS5}}$ (P85), $\overline{\text{SDWE}}$ (P90), $\overline{\text{SDRAS}}$ (P91), $\overline{\text{SDCAS}}$ (P92), SDLLDQM(P93), SDLUDQM(P94), SDCLK(P96)
B	Drive "1"	$\overline{\text{RD}}$, $\overline{\text{WRLL}}$ (P71), $\overline{\text{WRLU}}$ (P72), R/$\overline{\text{W}}$ (P73), $\overline{\text{SRWR}}$ (P74), $\overline{\text{SRLLB}}$ (P75), $\overline{\text{SRLUB}}$ (P76)
C	Drive "0"	SDCKE(P95)
D	None operation	D15 to D8(P17 to P10), D7 to D0

Note) Although the output buffer of $\overline{\text{RD}}$, $\overline{\text{WRLL}}$ (P71), $\overline{\text{WRLU}}$ (P72), R/$\overline{\text{W}}$ (P73), $\overline{\text{SRWR}}$ (P74), $\overline{\text{SRLLB}}$ (P75) and $\overline{\text{SRLUB}}$ (P76) is turned off at the time of bus release, a pull-up will be turned on and it will not become high impedance (HZ).

3.17.2 Connection example

Figure 3.17.2 show connection example.

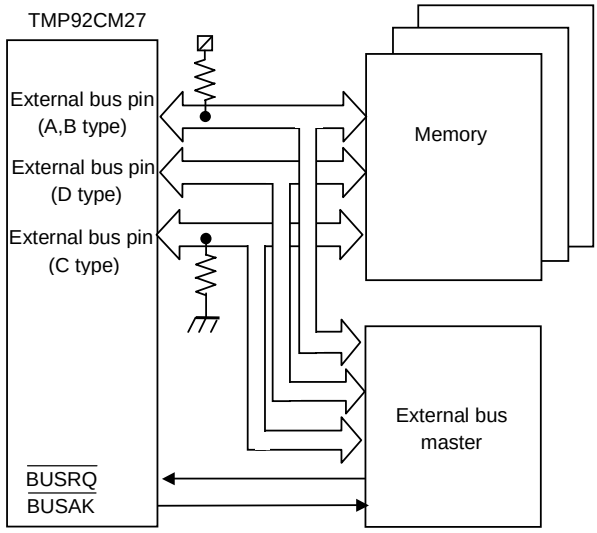


Figure 3.17.2 Connection example

3.17.3 Note

If use bus release function, be careful following notes.

1) Prohibit using this function together SDRAM controller

Prohibitalso SDRAMC basically, but if external bus master use SDRAM, set SDRAM to SR (self refresh) condition before bus release request. And, when finish bus release, release SR condition. In this case, confirm each condition by handshake of general purpose port.

2) Support standby mode

The condition that can receive this function is only CPU operationg condition and during IDLE2 mode. During IDLE1 and STOP condition don't receive. (Bus release function is ignored).

3) Internal resource access disable

External bus master cannnot access to internal memory and interhal I/O of TMP92CM27. Internal I/O operation during bus releasing.

4) Internal I/O operation during bus releasing

Internal I/O continue operation during bus releasing, please be careful. And, if set the watchdog timer, set runaway time by consider bus release time.

4. Electrical Characteristics

4.1 Maximum Ratings

Parameter	Symbol	Rating	Unit
Power Supply Voltage	VCC	-0.5 to 4.0	V
Input Voltage	V_{IN}	-0.5 to VCC+0.5	V
Output Current (1 pin)	IOL	2	mA
Output Current (1 pin)	IOH	-2	mA
Output Current (total)	ΣI_{OL}	80	mA
Output Current (total)	ΣI_{OH}	-80	mA
Power Dissipation (Ta=85℃)	P_D	600	mW
Soldering Temperature (10 s)	T_{SOLDER}	260	℃
Storage Temperature	T_{STG}	-65 to 150	℃
Operation Temperature	T_{OPR}	-40 to 85	℃

Note: The 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 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 maximum rating value will ever be exceeded.

Point of note about solderability of lead free products (attach “G” to package name)

Test parameter	Test condition	Note
Solderability	Use of Sn-63Pb solder Bath Solder bath temperature = 230°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: solderability rate until forming ≥ 95%
	Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux (use of lead free)	

4.2 DC Electrical Characteristics

VCC = 3.3 ± 0.3V / X1 = 4 to 40MHz / Ta = −40 to 85°C

Symbol	Parameter	Min	Typ	Max	Unit	Condition
VCC	Power Supply Voltage (DVCC=AVCC=DAVCC) (DVSS=AVSS=DAVSS=0V)	3.0		3.6	V	X1 = 6 to 10MHz (Note 1) X1 = 4 to 40MHz (Note 2)
VIL0	Input Low Voltage for D0 to D7 P10 to P17(D8 to D15)	−0.3		0.6	V	
VIL1	Input Low Voltage for PC0 to PC1, PC3 to PC4, PD0, PL4			0.3 × VCC		
VIL2	Input Low Voltage for P71 to P77, P86, P87, PA0 to PA5, PC2, PC5, PD1 to PD5, PF0 to PF7, PJ0 to PJ7, PK0 to PK7, PL0 to PL3, PL5 to PL7, PM0 to PM7, PN0 to PN3, $\overline{NMI}$, $\overline{RESET}$			0.25 × VCC		
VIL3	Input Low Voltage for AM0 to AM1			0.3		
VIL4	Input Low Voltage for X1			0.2 × VCC		
VIH0	Input High Voltage for D0 to D7 P10 to P17(D8 to D15)	2.0		VCC + 0.3	V	
VIH1	Input High Voltage for PC0 to PC1, PC3 to PC4, PD0, PL4	0.7 × VCC				
VIH2	Input High Voltage for P71 to P77, P86, P87, PA0 to PA5, PC2, PC5, PD1 to PD5, PF0 to PF7, PJ0 to PJ7, PK0 to PK7, PL0 to PL3, PL5 to PL7, PM0 to PM7, PN0 to PN3, $\overline{NMI}$, $\overline{RESET}$	0.75 × VCC				
VIH3	Input High Voltage for AM0 to AM1	VCC − 0.3				
VIH4	Input High Voltage for X1	0.8 × VCC				

Note 1) At the time of PLL use.
Note 2) At the time of PLL un-use.

Symbol	Parameter	Min	Typ	Max	Unit	Condition
VOL	Output Low Voltage			0.45	V	IOL = 1.6mA
VOL2	Output Low Voltage for PC0 to PC1, PC3 to PC4			0.45		IOL = 3.0mA
VOH	Output High Voltage	2.4				IOH = −400 μA
ILI	Input Leakage Current		0.02	±5	μA	0.0 ≤ Vin ≤ VCC
ILO	Output Leakage Current		0.05	±10	μA	0.2 ≤ Vin ≤ VCC−0.2V
VSTOP	Power Down Voltage at STOP (for initernal RAM back-up)	1.8		3.6	V	VIL2 = 0.2*VCC, VIH2 = 0.8*VCC
RRST	Pull Up Resister for $\overline{\text{RESET}}$	80		500	KΩ	
RKH	Programmable Pull Up Resister for P70 to P72, P74 to P76					
CIO	Pin Capacitance			10	pF	fc=1MHz
VTH	Schmitt Width for P71 to P77, P86, P87, PA0 to PA5, PC2, PC5, PD1 to PD5, PF0 to PF7, PJ0 to PJ7, PK0 to PK7, PL0 to PL3, PL5 to PL7, PM0 to PM7, PN0 to PN3, $\overline{\text{NMI}}$, $\overline{\text{RESET}}$	0.4	1.0		V	
VTH2	Schmitt Width for PC0 to PC1, PC3 to PC4	0.2			V	
ICC	NORMAL (Note 2)		50.0	60.0	mA	VCC=3.6V, fc=40MHz(fsys=20MHz)
	IDLE2		25.0	31.5		
	IDLE1		7.5	11.5		
	STOP		0.2	50	μA	VCC=3.6V

Note 1: Typical values are for when Ta = 25°C, Vcc = 3.3 V unless otherwise noted.

Note 2: ICC NORMAL measurement conditions:

All functions are operational; output pins except bus pin are open, and input pins are fixed. Bus pin CL=30pF.

4. 3 AC Characteristics

4.3.1 Basic Bus Cycle

Read cycle

VCC = 3.3 ± 0.3V / fc = 4 to 40MHz / Ta = −40 to 85°C

No.	Parameter	Symbol	Variable		fc=40MHz fsys=20MHz	fc=27MHz fsys=13.5MHz	Unit
			Min	Max			
1	OSC period (X1/X2)	t_{OSC}	25	250	25	37.0	ns
2	System Clock period (=T)	t_{CYC}	50	500	50	74.0	
3	SDCLK Low Width	t_{CL}	0.5T−15		10	22	
4	SDCLK High Width	t_{CH}	0.5T−15		10	22	
5-1	A0 to A23 Valid → D0 to D15 Input at 0WAIT	t_{AD}		2.0T−50	50	--	
5-2	A0 to A23 Valid → D0 to D15 Input at 1WAIT	t_{AD3}		3.0T−50	100	--	
6-1	$\overline{RD}$ Fall → D0 to D15 Input at 0WAIT	t_{RD}		1.5T−45	30	66	
6-2	$\overline{RD}$ Fall → D0 to D15 Input at 1WAIT	t_{RD3}		2.5T−45	80	140	
7-1	$\overline{RD}$ Low Width at 0WAIT	t_{RR}	1.5T−20		55	91	
7-2	$\overline{RD}$ Low Width at 1WAIT	t_{RR3}	2.5T−20		105	165	
8	A0 to A23 Valid → $\overline{RD}$ Fall	t_{AR}	0.5T−20		5	17	
9	$\overline{RD}$ Fall → SDCLK Rise	t_{RK}	0.5T−20		5	17	
10	A0 to A23 Valid → D0 to D15 Hold	t_{HA}	0		0	0	
11	$\overline{RD}$ Rise → D0 to D15 Hold	t_{HR}	0		0	0	
12	$\overline{WAIT}$ Set-up Time	t_{TK}	20		20	20	
13	$\overline{WAIT}$ Hold Time	t_{KT}	5		5	5	
14	Data Byte Control Access Time for SRAM	t_{SBA}		1.5T−45	40	66	
15	$\overline{RD}$ High Width	t_{RRH}	0.5T−15		10	22	

Write cycle

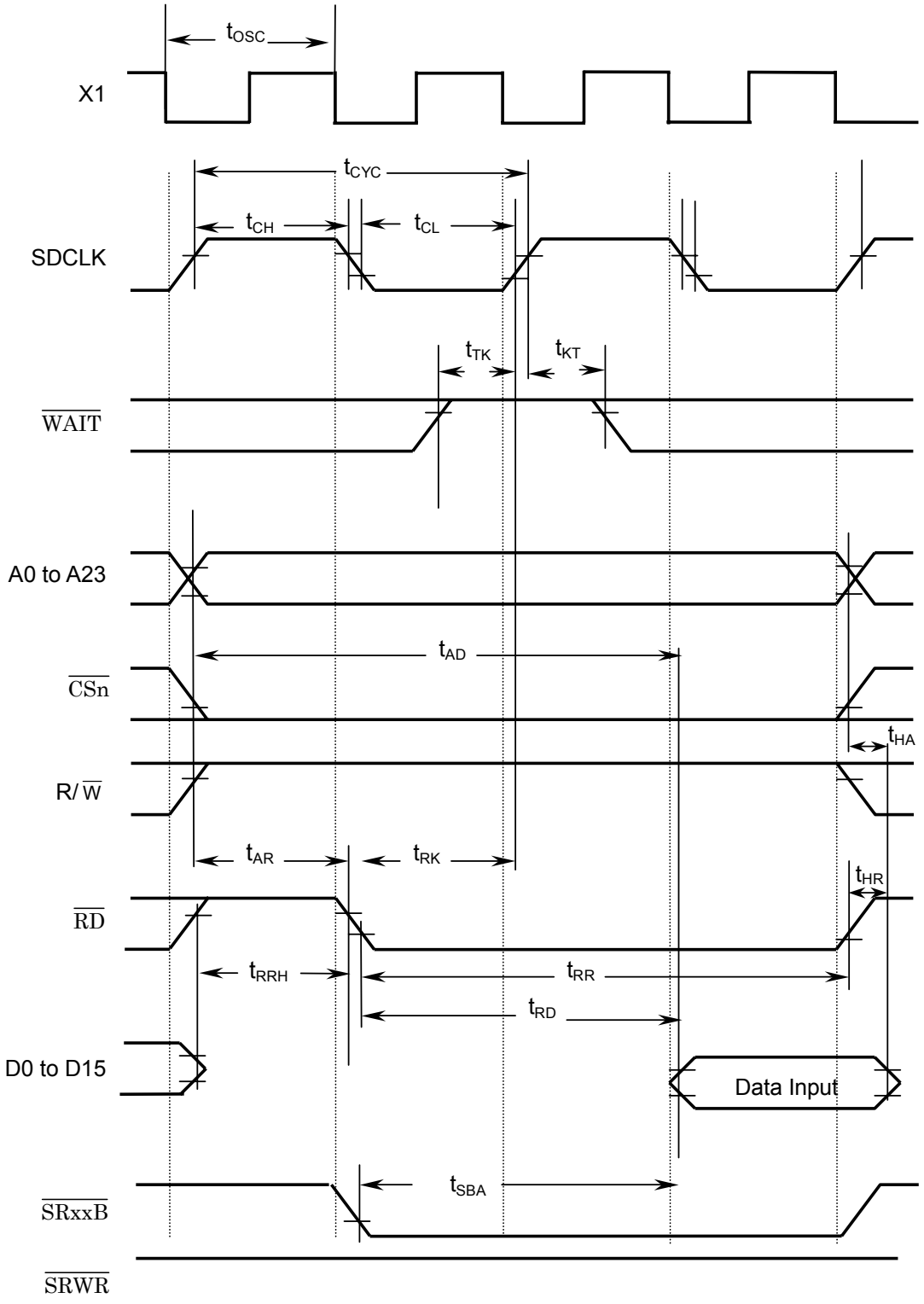
VCC = 3.3 ± 0.3V / fc = 4 to 40MHz / Ta = −40 to 85°C

No.	Parameter	Symbol	Variable		fc=40MHz fsys=20MHz	fc=27MHz fsys=13.5MHz	Unit
			Min	Max			
16-1	D0 to D15 Valid → $\overline{WRxx}$ Rise at 0WAIT	t_{DW}	1.25T−35		27.5	57.5	ns
16-2	D0 to D15 Valid → $\overline{WRxx}$ Rise at 1WAIT	t_{DW3}	2.25T−35		77.5	131.5	
17-1	$\overline{WRxx}$ Low Width at 0WAIT	t_{WW}	1.25T−30		32.5	62.5	
17-2	$\overline{WRxx}$ Low Width at 1WAIT	t_{WW3}	2.25T−30		82.5	136.5	
18	A0 to A23 Valid → $\overline{WR}$ Fall	t_{AW}	0.5T−20		5	17	
19	$\overline{WRxx}$ Fall → SDCLK Rise	t_{WK}	0.5T−20		5	17	
20	$\overline{WRxx}$ Rise → A0 to A23 Hold	t_{WA}	0.25T−5		7.5	13.5	
21	$\overline{WRxx}$ Rise → D0 to D15 Hold	t_{WD}	0.25T−5		7.5	13.5	
22	$\overline{RD}$ Rise → D0 to D15 Output	t_{RDO}	0.5T−5		20	--	
23	Write Pulse Width for SRAM	t_{SWP}	1.25T−30		32.5	62.5	
24	Data Byte Control to End of Write for SRAM	t_{SBW}	1.25T−30		32.5	62.5	
25	Address Setup Time for SRAM	t_{SAS}	0.5T−20		5	17	
26	Write Recovery Time for SRAM	t_{SWR}	0.25T−5		7.5	13.5	
27	Data Setup Time for SRAM	t_{SDS}	1.25T−35		27.5	57.5	
28	Data Hold Time for SRAM	t_{SDH}	0.25T−5		7.5	13.5	

AC Measuring Condition

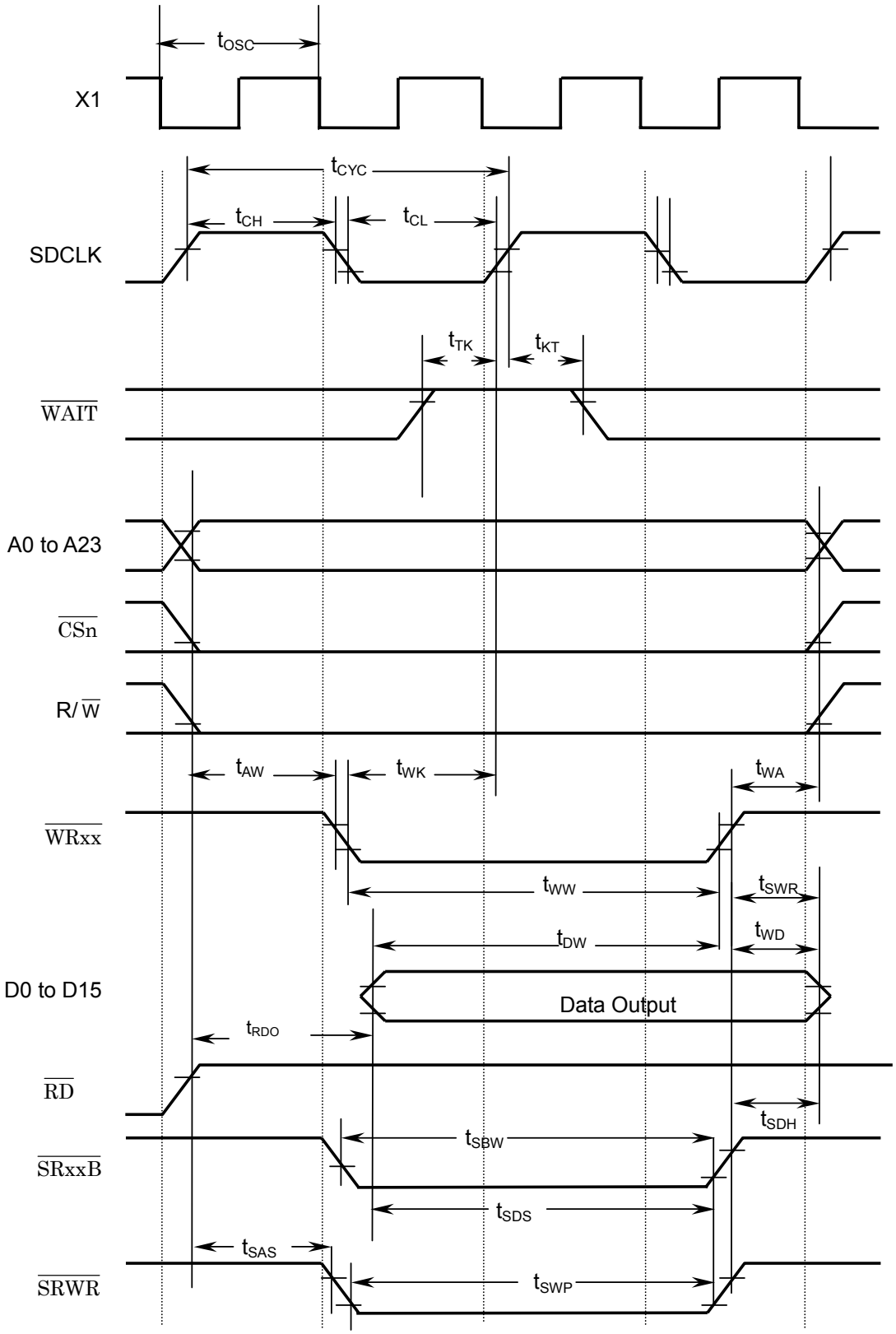
●Output level : High = 0.7Vcc, Low = 0.3Vcc, CL = 50pF ●Input level : High = 0.9Vcc, Low = 0.1Vcc

(1) Read cycle (0 wait, fc=f_{OSCH}, f_{FPH}=fc/1)



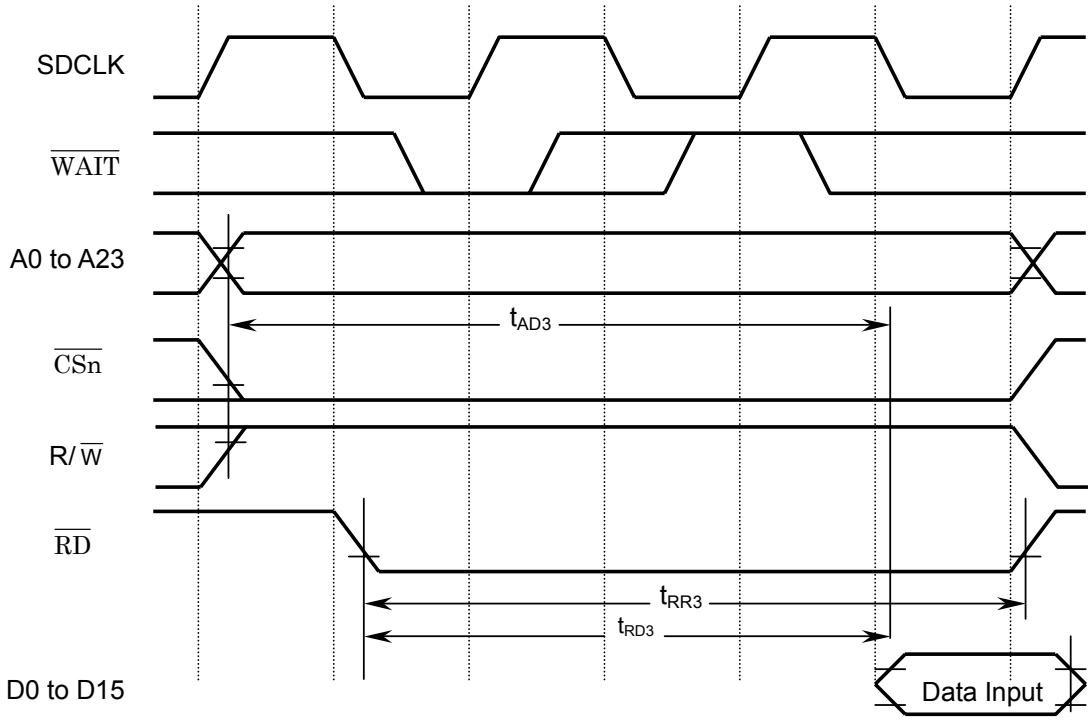
Note: The phase relation between X1 input signal and the other signals is undefined.
The above timing chart is an example.

(2) Write cycle (0 wait, fc=f_{OSCH}, f_{FPH}=fc/1)

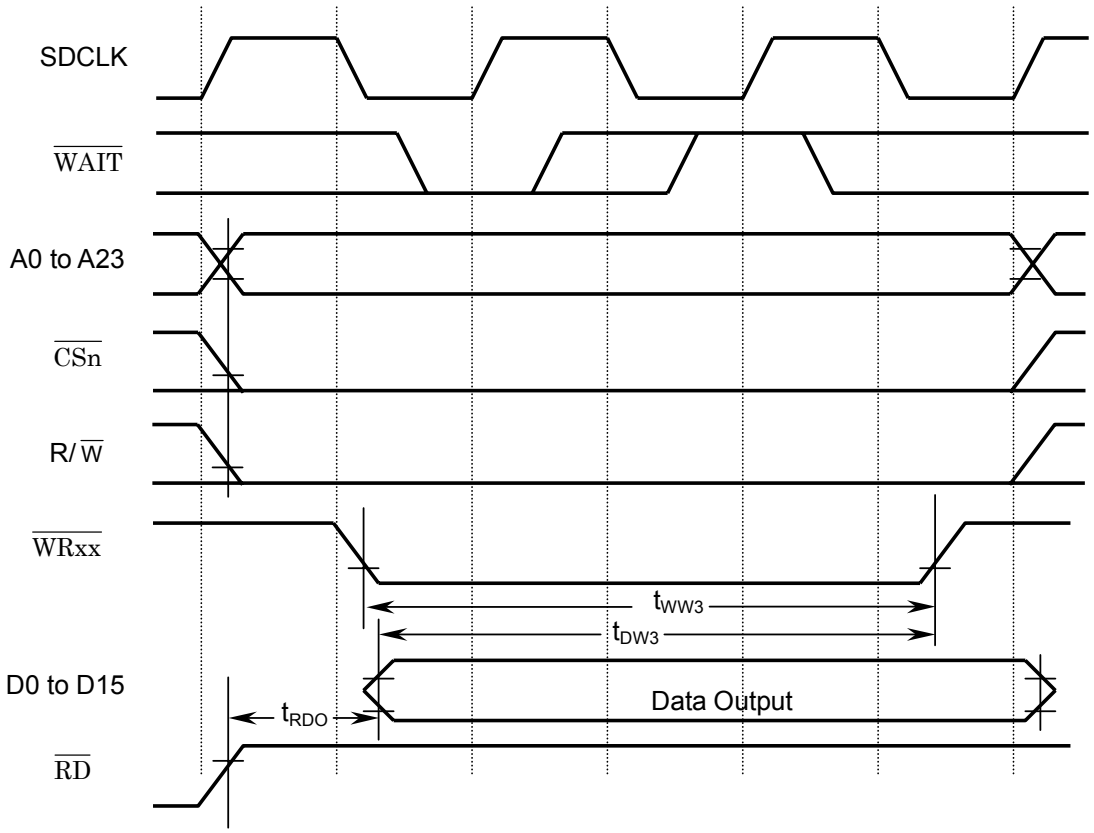


Note: The phase relation between X1 input signal and the other signals is undefined.
The above timing chart is an example.

(3) Read cycle (1 wait,fc=f_{OSCH},f_{FPH}=fc/1)



(4) Write cycle (1 wait,fc=f_{OSCH},f_{FPH}=fc/1)

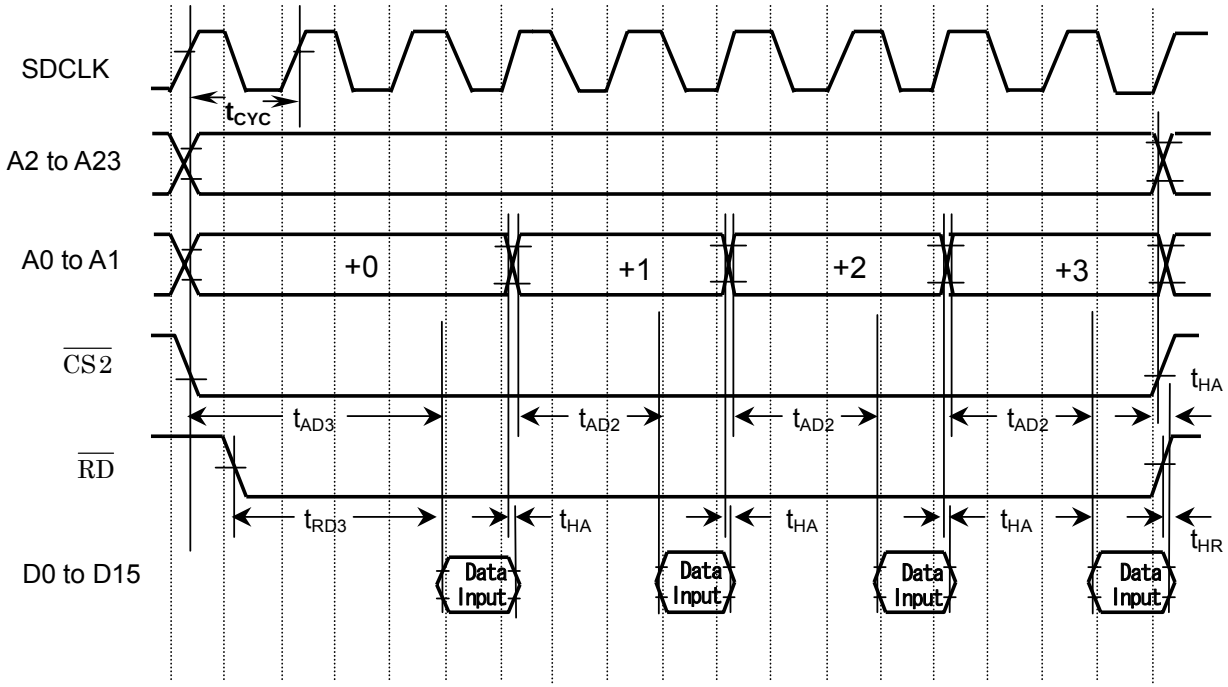


4.3.2 Page ROM read cycle

(1) Page ROM Read Cycle (3-2-2-2 mode)

No	Symbol	Parameter	Variable		40MHz	27MHz	Unit
			Min	Max			
1	t_{CYC}	System Clock Period (=T)	50	166.7	50	74	ns
2	t_{AD2}	A0,A1 → D0 to D15 Input		2.0T−50	50	98	
3	t_{AD3}	A2 to A23 → D0 to D15 Input		3.0T−50	100	172	
4	t_{RD3}	$\overline{RD}$ Fall → D0 to D15 Input		2.5T−45	80	140	
5	t_{HA}	A0 to A23 Invalid → D0 to D15 Hold	0		0	0	
6	t_{HR}	$\overline{RD}$ Rise → D0 to D15 Hold	0		0	0	

AC Measuring Condition
- ●Output level: High = 0.7Vcc, Low = 0.3Vcc, CL = 50pF
- ●Input level: High = 0.9Vcc, Low = 0.1Vcc



4.3.3 SDRAM Controller AC Characteristics

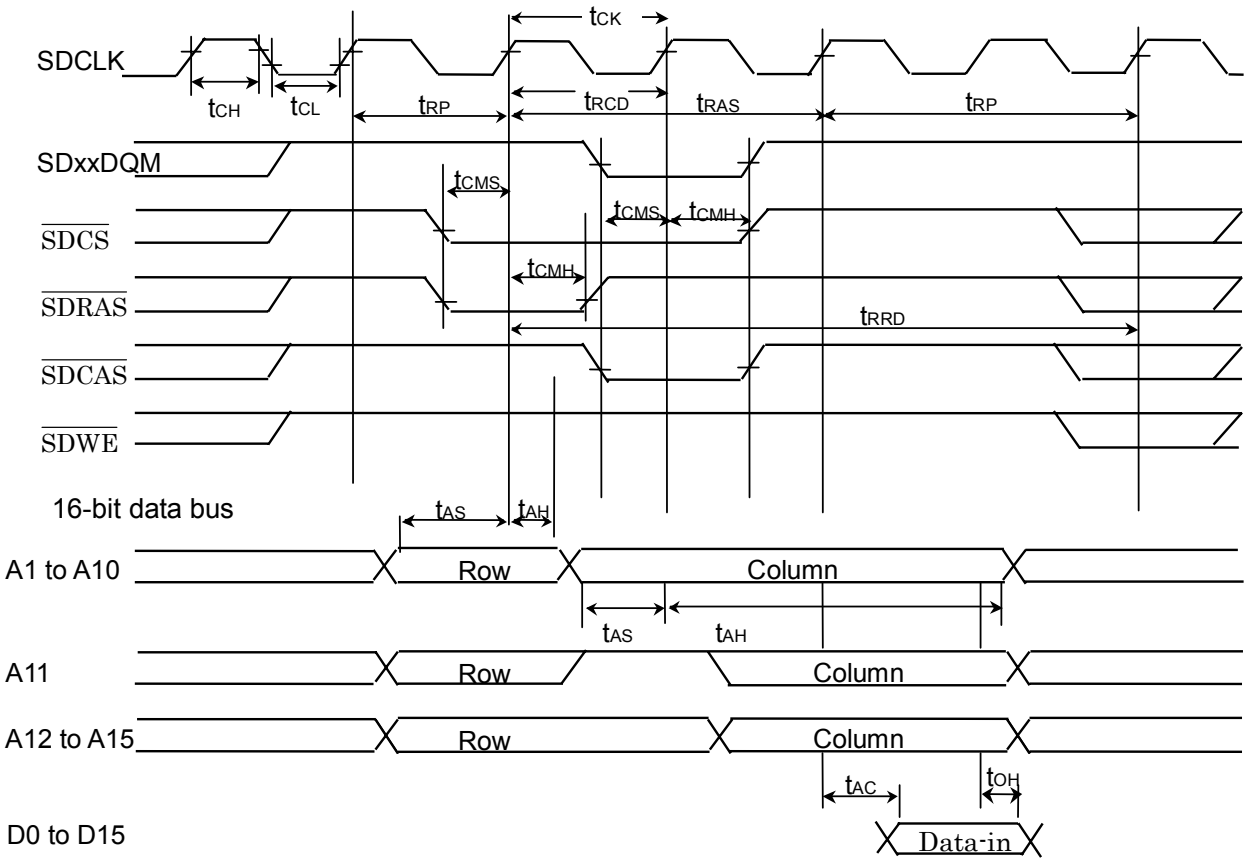
No	Symbol	Parameter	Variable		40MHz	27MHz	Unit
			Min	Max			
1	t_{RC}	Ref/Active to Ref/Active Command Period	2T		100	148	ns
2	t_{RAS}	Active to Precharge Command Period	2T	12210	100	148	
3	t_{RCD}	Active to Read/Write Command Delay Time	T		50	74	
4	t_{RP}	Precharge to Active Command Period	T		50	74	
5	t_{RRD}	Active to Active Command Period	3T		150	222	
6	t_{WR}	Write Recovery Time(CL*=2)	T		50	74	
7	t_{CK}	CLK Cycle Time(CL*=2)	T		50	74	
8	t_{CH}	CLK High Level Width	0.5T-15		10	22	
9	t_{CL}	CLK Low Level Width	0.5T-15		10	22	
10	t_{AC}	Access Time from CLK(CL*=2)		T-30	20	44	
11	t_{OH}	Output Data Hold Time	0		0	0	
12	t_{DS}	Data-in Set-up Time	0.5T-10		15	27	
13	t_{DH}	Data-in Hold Time	T-15		35	59	
14	t_{AS}	Address Set-up Time	0.75T-30		7.5	25.5	
15	t_{AH}	Address Hold Time	0.25T-9		3.5	9.5	
16	t_{CKS}	CKE Set-up Time	0.5T-15		10	22	
17	t_{CMS}	Command Set-up Time	0.5T-15		10	22	
18	t_{CMH}	Command Hold Time	0.5T-15		10	22	
19	t_{RSC}	Mode Register Set Cycle Time	T		50	74	

CL*: CAS latency.

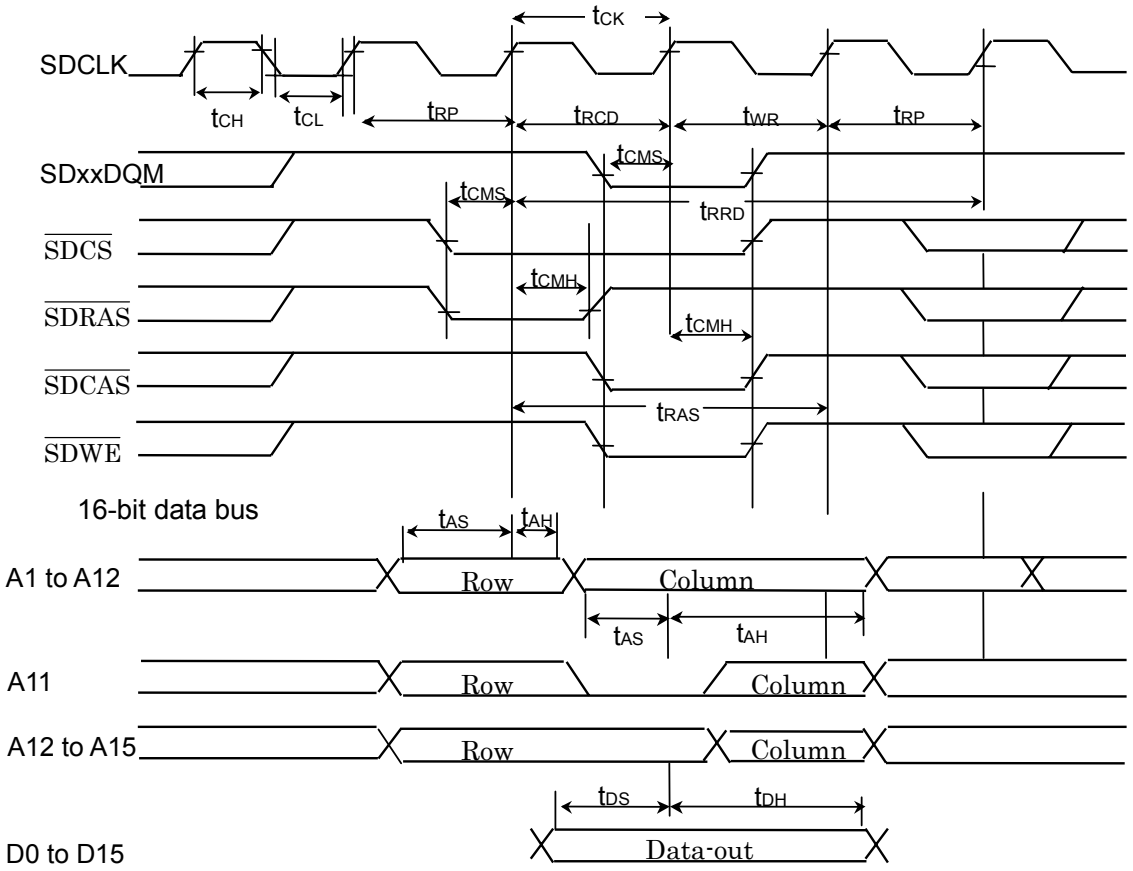
AC measuring conditions

- Output level: High = 0.7 Vcc, Low = 0.3 Vcc, CL = 50 pF
- Input level: High = 0.9 Vcc, Low = 0.1 Vcc.

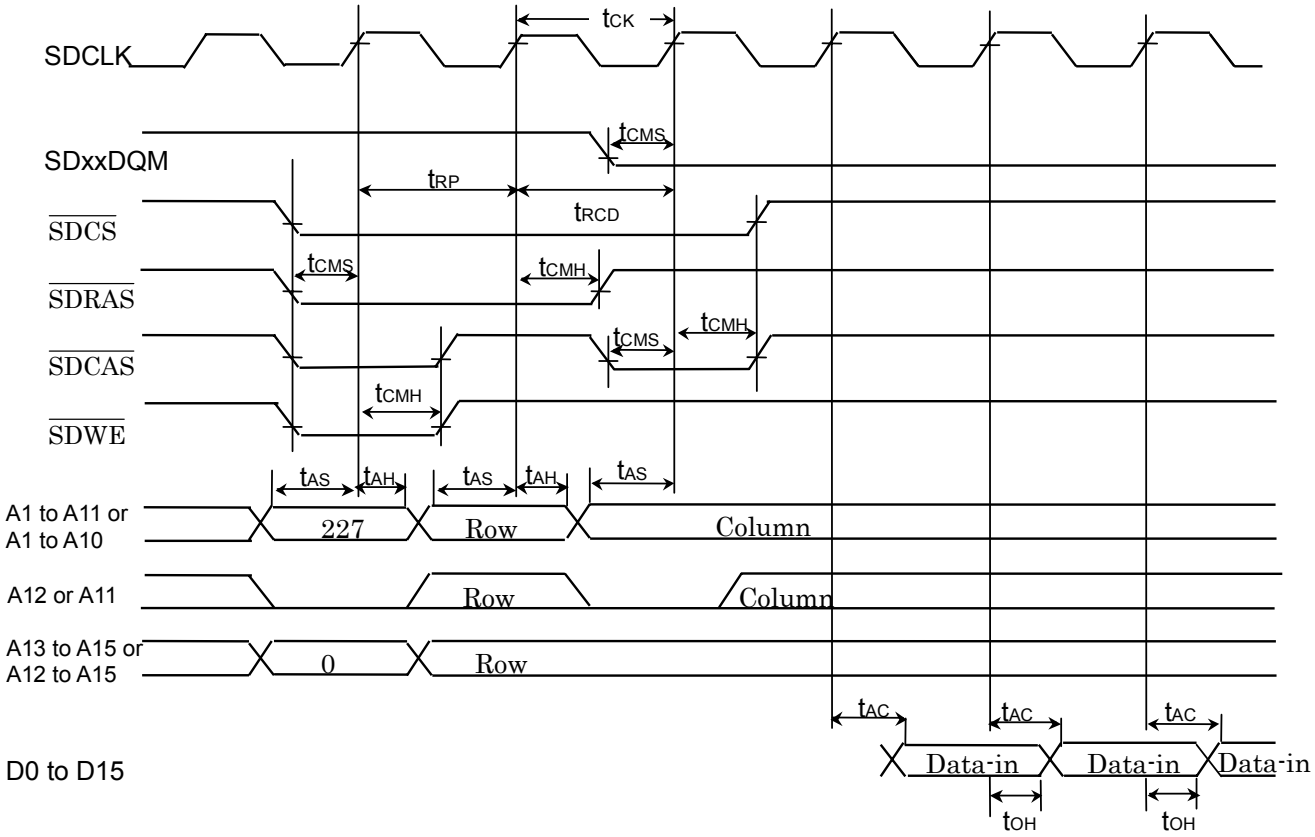
(1) SDRAM read timing (CPU access)



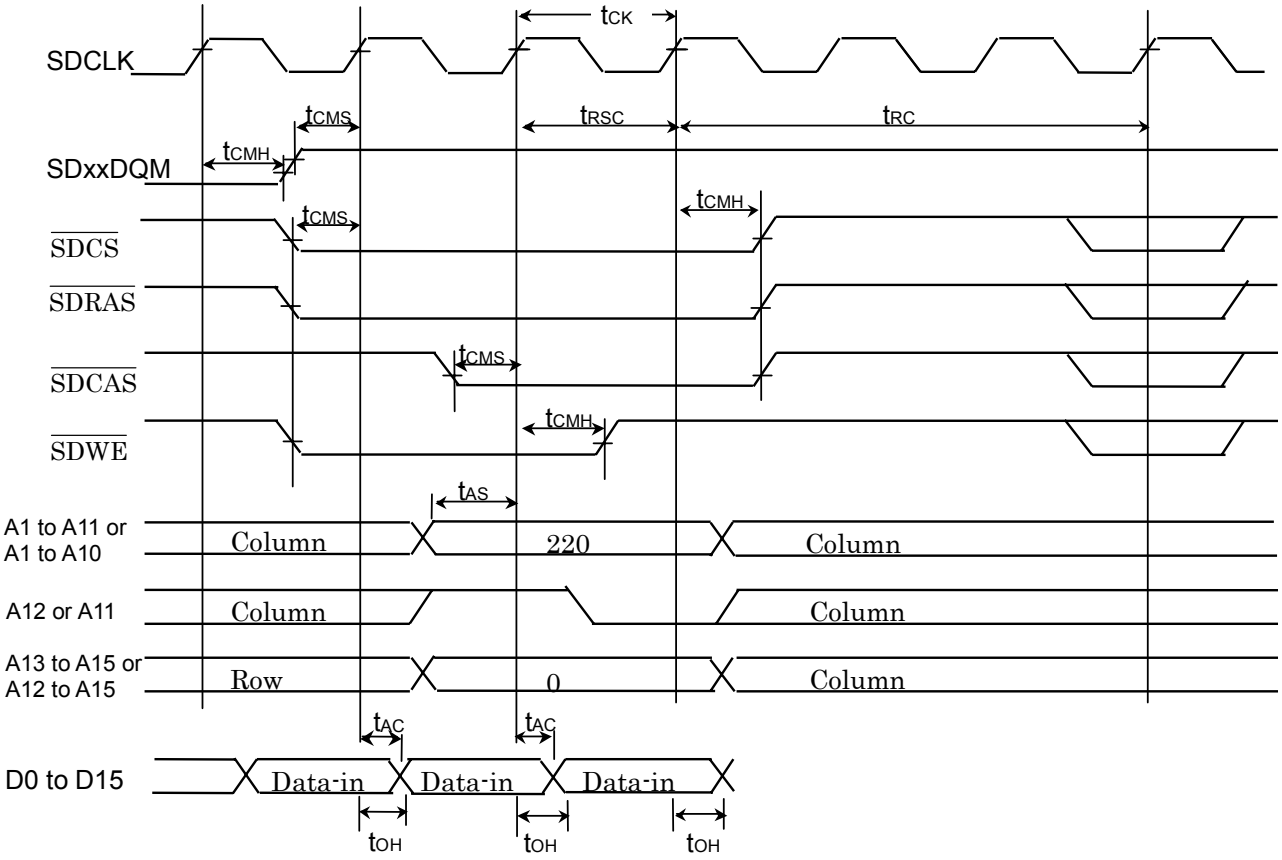
(2) SDRAM write timing (CPU access)



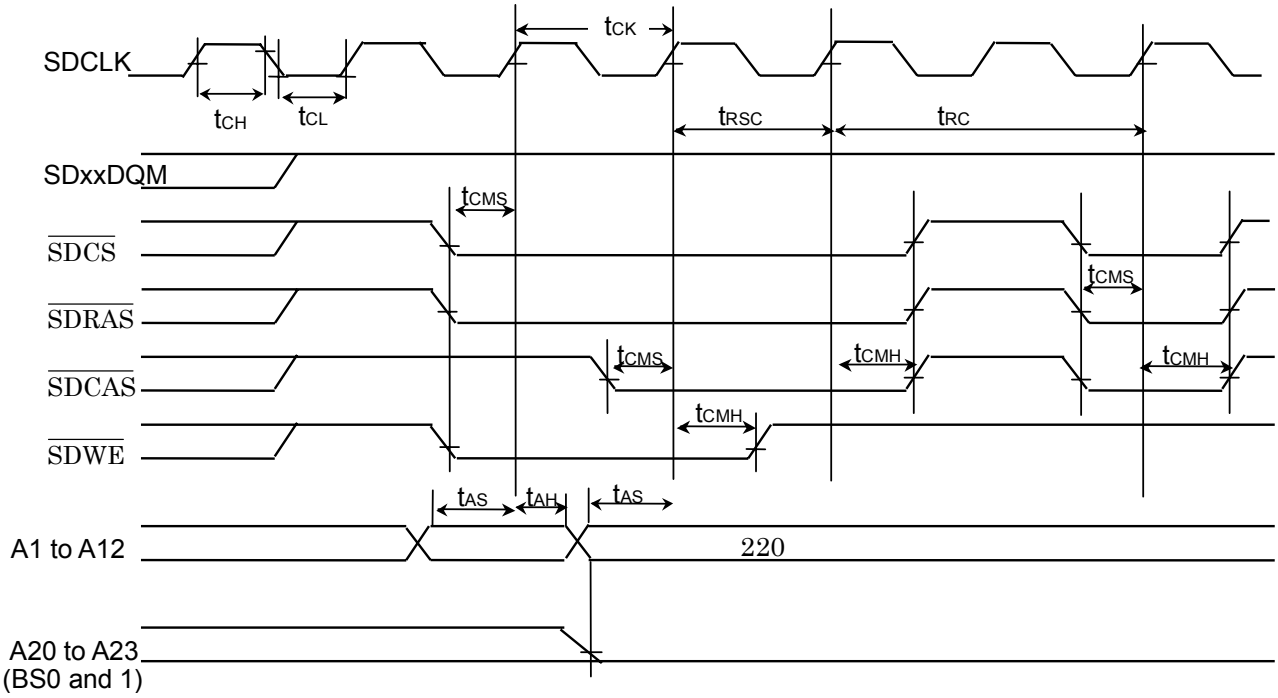
(3) SDRAM burst read timing (Start of burst cycle)



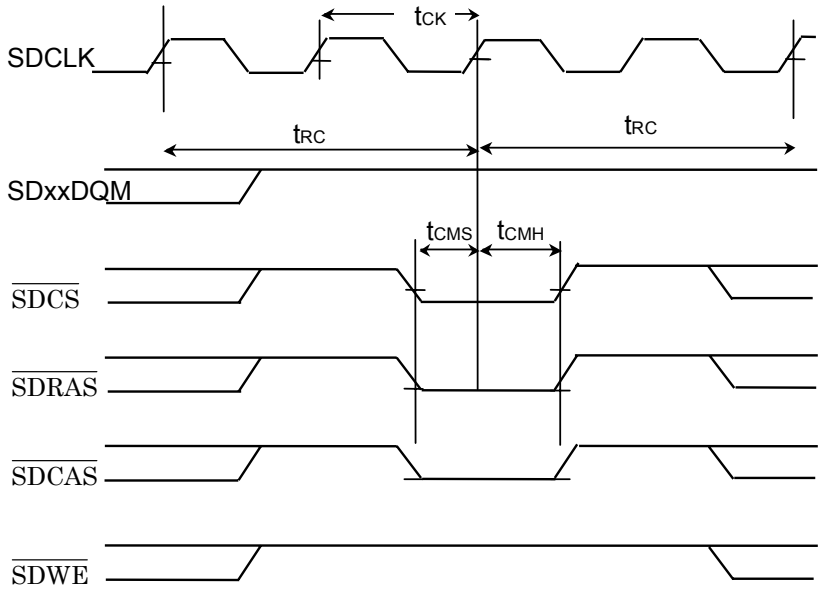
(4) SDRAM burst read timing (End of burst cycle)



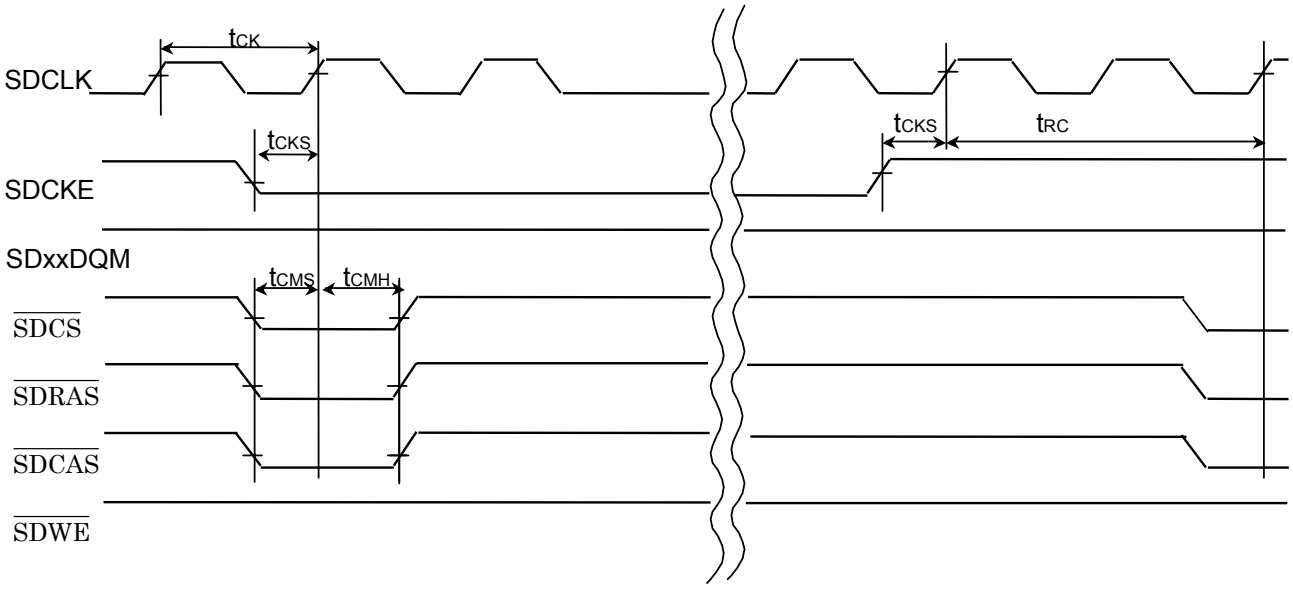
(5) SDRAM initialize timing



(6) SDRAM refresh timing



(7) SDRAM self refresh timing



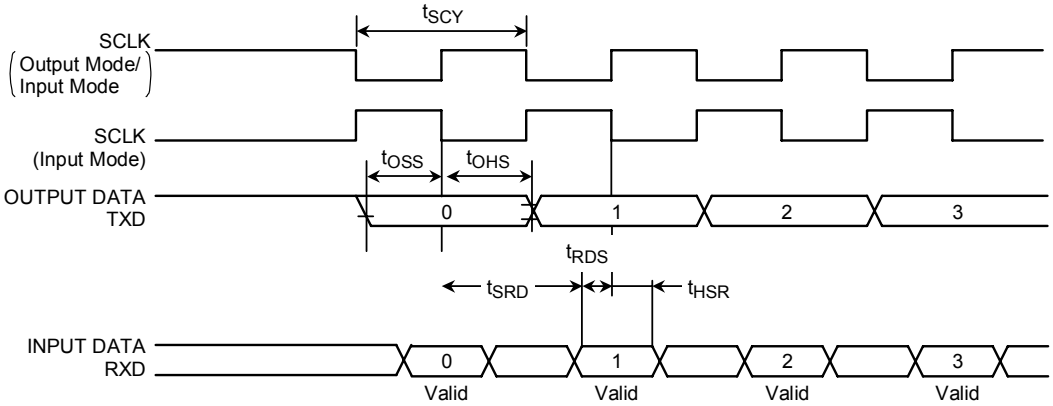
4.3.4 Serial Channel Timing

(1) SCLK input mode (I/O interface mode)

Parameter	Symbol	Variable		fc=40MHz fsys=20MHz		fc=27MHz fsys=13.5MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Cycle (Programmable)	t_{SCY}	16X		0.4		0.59		μs
Output Data → SCLK Rise/Fall	t_{OSS}	t_{SCY}/2-4X-90		10		58		ns
SCLK Rise/Fall → Output Data Hold	t_{OHS}	t_{SCY}/2+2X+0		250		370		
SCLK Rise/Fall → Input Data Hold	t_{HSR}	3X+10		85		121		
SCLK Rise/Fall → Input Data Hold	t_{SRD}		t_{SCY}-0		400		592	
Input Data Valid → SCLK Rise/Fall	t_{RDS}	0		0		0		

(2) SCLK output mode (I/O interface mode)

Parameter	Symbol	Variable		fc=40MHz fsys=20MHz		fc=27MHz fsys=13.5MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle (Programmable)	t_{SCY}	16X	8192X	0.4	204	0.59	303	μs
Output Data → SCLK Rise/Fall	t_{OSS}	t_{SCY}/2-40		160		256		ns
SCLK Rise/Fall → Output Data Hold	t_{OHS}	t_{SCY}/2-40		160		256		
SCLK Rise/Fall → Input Data Hold	t_{HSR}	0		0		0		
SCLK Rise/Fall → Input Data Valid	t_{SRD}		t_{SCY}-1X-180		195		375	
Input Data Valid → SCLK Rise/Fall	t_{RDS}	1X+180		205		217		



4.3.5 Interrupts

Parameter	Symbol	Variable		fc=40MHz fsys=20MHz		fc=27MHz fsys=13.5MHz		Unit
		Min	Max	Min	Max	Min	Max	
INT0 to INTB, $\overline{NMI}$ low level width	t_{INTAL}	4T+40		240		336		ns
INT0 to INTB, $\overline{NMI}$ high level width	t_{INTAH}	4T+40		240		336		ns

4.3.6 AD Conversion Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
AVCC	AD Converter Power Supply Voltage	VCC	VCC	VCC	V
AVSS	AD Converter Ground	VSS	VSS	VSS	
AVIN	Analog Input Voltage	AVSS		AVCC	
E_T	Total error (Quantize error of ±0.5LSB is included)		±1.0	±4.0	LSB

Note 1: 1LSB = (AVCC − AVSS)/1024 [V]

Note 2: Minimum frequency for operation
Clock frequency which is selected by clock is over than 4MHz, operation is guaranteed.

Note 3: The value for I_{CC} includes the current which flows through the AVCC pin.

4.3.7 DA Conversion Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Unit
DAOUT	Output voltage range	RL = 3.6 KΩ	DAVSS+0.3		DAVCC-0.3	V
E_T	Total error	RL = 3.6 KΩ		±1.0	±4.0	LSB
RL	Resistive load	DAVSS+0.3 ≤ DAOUT ≤ DAVCC-0.3	3.6			KΩ

Note 1: 1LSB = (DAVCC − DAVSS)/256 [V]

Note 2: The value for I_{CC} includes the current which flows through the DVCC pin.

4.3.8 Event Counter (TA0IN, TA2IN, TA4IN, TA6IN, TB0IN0, TB0IN1, TB1IN0, TB1IN1, TB2IN0, TB2IN1, TB3IN0, TB3IN1)

Parameter	Symbol	Variable		fc = 40MHz fsys = 20MHz		fc = 27MHz fsys = 13.5MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock period	t_{VCK}	8X+100		300		396		ns
Clock low level width	t_{VCKL}	4X+40		140		188		ns
Clock high level width	t_{VCKH}	4X+40		140		188		ns

Note: Symbol x in the above table means the period of clock f_{FPH}, it's half period of the system clock f_{SYS} for CPU core. The period of f_{FPH} depends on the clock gear setting.

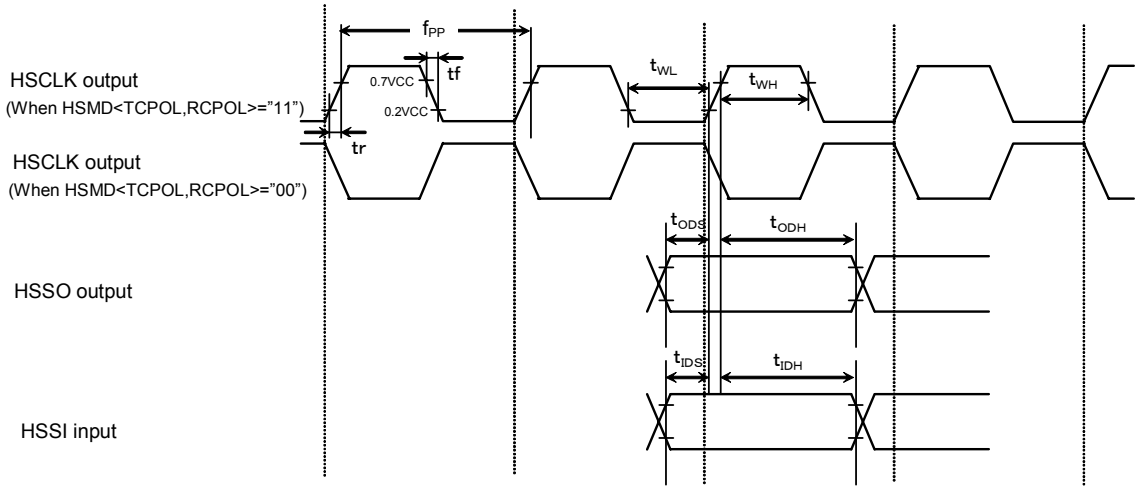
4.3.9 High Speed SIO Timing

Symbol	Parameter	Variable		40 MHz	36 MHz	27MHz	Unit
		Min	Max				
f_{PP}	HSCLK frequency (= 1/X)		10	10	9	6.75	MHz
t_r	HSCLK rising timing		8	8	8	8	ns
t_f	HSCLK falling time		8	8	8	8	
t_{WL}	HSCLK Low pulse width	0.5X-8		42	47	66	
t_{WH}	HSCLK High pulse width	0.5X-16		34	39	58	
t_{ODS1}	Output data valid → HSCLK rise	0.5X-18		32	37	56	
t_{ODS2}	Output data valid → HSCLK fall	0.5X-23		27	32	51	
t_{ODH}	HSCLK rise/fall → Output data hold	0.5X-10		40	45	64	
t_{IDS}	Input data valid → HSCLK rise/fall	0X+20		20	20	20	
t_{IDH}	HSCLK rise/fall → Input data hold	0X+5		5	5	5	

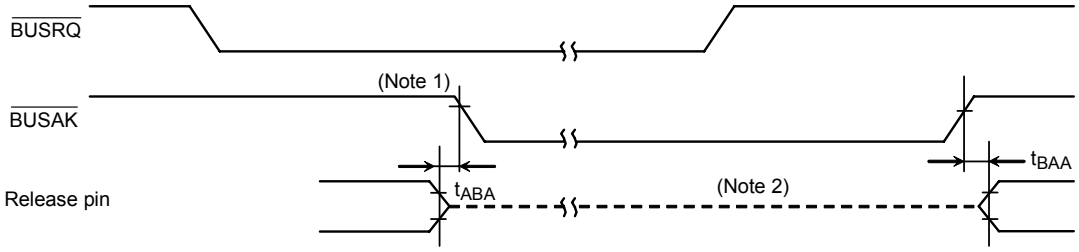
AC measuring conditions

Output level : High = 0.7 VCC, Low = 0.2 VCC, CL = 25 pF

Input level : High = 0.9 VCC, Low = 0.1 VCC



4.3.10 External bus release function



Parameter	Symbol	Variable		fc=40 MHz fsys=20MHz		Unit
		Min	Max	Min	Max	
Floating $\overline{BUSAK}$ falling	t_{ABA}	0	30	0	30	ns
Floating $\overline{BUSAK}$ rising	t_{BAA}	0	30	0	30	ns

Note 1: Even if the $\overline{BUSRQ}$ signal goes low, the bus will not be released while the $\overline{WAIT}$ signal is low. The bus will only be released when $\overline{BUSRQ}$ goes low while $\overline{WAIT}$ is high.

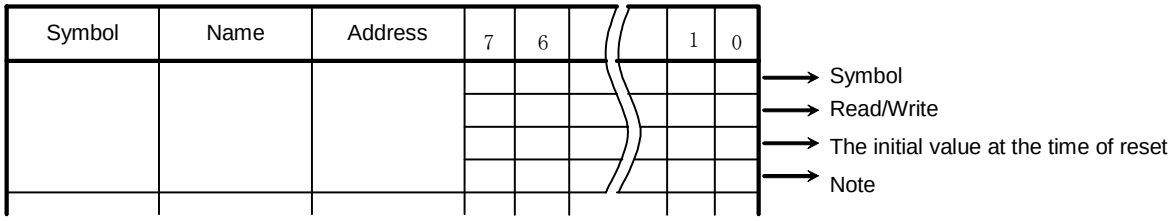
Note 2: This line shows only that the output buffer is in the off state. It does not indicate that the signal level is fixed. Just after the bu is released, the signal level set before the bus was released is maintained dynamically by the external capacitance. Therefore, to fix the signal level using an external resistor during bus release, careful design is necessary, as fixing of the level is delayed. The internal programmable pull-up/pull-down resistor is switched between the active and non-active states by the internal signal.

5 . Special Function Register

Special function register(SFR) is control of an input-and-output port and the control register of a circumference part, and it is assigned to 8 K bytes of address area of 000000H to 001FFFH.

(1) Input-and-Output port
(2) Interrupt control
(3) DMA controller
(4) Memory controller
(5) Clock control / PLL
(6) SDRAM controller
(7) 8-bit timer
(8) 16-bit timer
(9) Pattern Generator
(10) High speed serial channels
(11) UART mode / Serial channels
(12) I^2CBUS mode / Serial channels
(13) AD converter
(14) DA converter
(15) Watch dog timer
(16) Key-on wake up

Composition of a table



Note1 : "Prohibit RMW" of a table shows that it do not support read-modify-write operation for the register.
Example) When only bit 0 of a P1CR register is set to "1",usually “SET 0,(0006H)”, but It is necessary to write in this register to a 8-bit register by the "LD" (transfer) command for "Prohibit RMW”.

The meaning of a sign

R/W	:Read/Write enable
R	:Only Read enable
W	:Only Write enable
W*	:Read Write enable (However, always read as “1”)
Prohibit RMW	:Read-modify-write instruction is Prohibit ed (EX, ADD, ADC, BUS, SBC, INC,DEC,AND, OR, XOR, STCF, RES, SET, CHG, TEST, RLC, RRC, RL, RR, SLA,SRA, SLL, SRL, RLD, RRD instruction is disable)。
Prohibit RMW*	:Read-modify-write instruction is Prohibit ed in the case of pull-up control of the port.

Table 5. I/O Register Map

[1] Input-and-Output port

address	register name
0000H	
1H	
2H	
3H	
4H	P1
5H	
6H	P1CR
7H	P1FC
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

address	register name
0010H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P6
9H	
AH	P6CR
BH	P6FC
CH	P7
DH	
EH	P7CR
FH	P7FC

address	register name
0020H	P8
1H	P8CR
2H	P8FC
3H	P8FC2
4H	P9
5H	P9DR
6H	
7H	P9FC
8H	PA
9H	PAFC2
AH	PACR
BH	PAFC
CH	
DH	
EH	
FH	

address	register name
0030H	PC
1H	PCFC2
2H	PCCR
3H	PCFC
4H	PD
5H	PDFC2
6H	PDCR
7H	PDFC
8H	
9H	
AH	
BH	
CH	PF
DH	PFFC2
EH	PFCR
FH	PFFC

address	register name
0040H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PJ
DH	PJFC2
EH	PJCR
FH	PJFC

address	register name
0050H	PK
1H	PKFC2
2H	
3H	PKFC
4H	PL
5H	PLFC2
6H	PLCR
7H	PLFC
8H	PM
9H	
AH	
BH	PMFC
CH	PN
DH	
EH	
FH	PNFC

Note) Do not access address to which Register Name is not assigned. register is not assigned to the address.

[2] Interrupt control

address	register name
00D0H	INTE01
1H	INTE23
2H	INTE45
3H	INTE67
4H	INTETA01
5H	INTETA23
6H	INTE8TA45
7H	INTE9TA67
8H	INTES0
9H	INTES1
AH	INTES2
BH	INTES3
CH	INTESB0
DH	INTESB1
EH	INTEAHSC0
FH	INTEBHSC1

address	register name
00E0H	INTETB0
1H	
2H	INTETB1
3H	
4H	INTEPAD
5H	INTETB2
6H	INTETB3
7H	INTETB4
8H	INTETB5
9H	INTETBOX
AH	
BH	
CH	
DH	
EH	
FH	INTNMWDT

[3] DMA controller

address	register name
00F0H	INTETC01
1H	INTETC23
2H	INTETC45
3H	INTETC67
4H	
5H	SIMC
6H	IIMC0
7H	
8H	INTCLR
9H	
AH	IIMC1
BH	IIMC2
CH	BECSL
DH	BECSH
EH	EMUCR
FH	MSAREMU

address	register name
0100H	DMA0V
1H	DMA1V
2H	DMA2V
3H	DMA3V
4H	DMA4V
5H	DMA5V
6H	DMA6V
7H	DMA7V
8H	DMAB
9H	DMAR
AH	
BH	
CH	INTSEL
DH	INTST
EH	IIMC3
FH	IIMC4

[4] Memory controller

address	register name
0140H	B0CSL
1H	B0CSH
2H	MAMR0
3H	MSAR0
4H	B1CSL
5H	B1CSH
6H	MAMR1
7H	MSAR1
8H	B2CSL
9H	B2CSH
AH	MAMR2
BH	MSAR2
CH	B3CSL
DH	B3CSH
EH	MAMR3
FH	MSAR3

address	register name
0150H	B4CSL
1H	B4CSH
2H	MAMR4
3H	MSAR4
4H	B5CSL
5H	B5CSH
6H	MAMR5
7H	MSAR5
8H	BEXCSL
9H	BEXCSH
AH	
BH	
CH	
DH	
EH	
FH	

[5] Clock control / PLL

address	register name
0160H	
1H	
2H	
3H	
4H	
5H	
6H	PMEMCR
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

address	register name
10E0H	SYSCR0
1H	SYSCR1
2H	SYSCR2
3H	EMCCR0
4H	EMCCR1
5H	EMCCR2
6H	
7H	
8H	PLLCR0
9H	PLLCR1
AH	
BH	
CH	
DH	
EH	
FH	

[6] SDRAM controller

address	register name
0250H	SDACR1
1H	SDACR2
2H	SDRCR
3H	SDCMM
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[7] 8-bit timer

address	register name
1100H	TA01RUN
1H	
2H	TA0REG
3H	TA1REG
4H	TA01MOD
5H	TA1FFCR
6H	
7H	
8H	TA23RUN
9H	
AH	TA2REG
BH	TA3REG
CH	TA23MOD
DH	TA3FFCR
EH	
FH	

address	register name
1110H	TA45RUN
1H	
2H	TA4REG
3H	TA5REG
4H	TA45MOD
5H	TA5FFCR
6H	
7H	
8H	TA67RUN
9H	
AH	TA6REG
BH	TA7REG
CH	TA67MOD
DH	TA7FFCR
EH	
FH	

[8] 16-bit timer

address	register name
1180H	TB0RUN
1H	
2H	TB0MOD
3H	TB0FFCR
4H	
5H	
6H	
7H	
8H	TB0RG0L
9H	TB0RG0H
AH	TB0RG1L
BH	TB0RG1H
CH	TB0CP0L
DH	TB0CP0H
EH	TB0CP1L
FH	TB0CP1H

address	register name
1190H	TB1RUN
1H	
2H	TB1MOD
3H	TB1FFCR
4H	
5H	
6H	
7H	
8H	TB1RG0L
9H	TB1RG0H
AH	TB1RG1L
BH	TB1RG1H
CH	TB1CP0L
DH	TB1CP0H
EH	TB1CP1L
FH	TB1CP1H

address	register name
11A0H	TB2RUN
1H	
2H	TB2MOD
3H	TB2FFCR
4H	
5H	
6H	
7H	
8H	TB2RG0L
9H	TB2RG0H
AH	TB2RG1L
BH	TB2RG1H
CH	TB2CP0L
DH	TB2CP0H
EH	TB2CP1L
FH	TB2CP1H

address	register name
11B0H	TB3RUN
1H	
2H	TB3MOD
3H	TB3FFCR
4H	
5H	
6H	
7H	
8H	TB3RG0L
9H	TB3RG0H
AH	TB3RG1L
BH	TB3RG1H
CH	TB3CP0L
DH	TB3CP0H
EH	TB3CP1L
FH	TB3CP1H

address	register name
11C0H	TB4RUN
1H	
2H	TB4MOD
3H	TB4FFCR
4H	
5H	
6H	
7H	
8H	TB4RG0L
9H	TB4RG0H
AH	TB4RG1L
BH	TB4RG1H
CH	TB4CP0L
DH	TB4CP0H
EH	TB4CP1L
FH	TB4CP1H

address	register name
11D0H	TB5RUN
1H	
2H	TB5MOD
3H	TB5FFCR
4H	
5H	
6H	
7H	
8H	TB5RG0L
9H	TB5RG0H
AH	TB5RG1L
BH	TB5RG1H
CH	TB5CP0L
DH	TB5CP0H
EH	TB5CP1L
FH	TB5CP1H

[9] Pattern Generator

address	register name
1460H	PG0REG
1H	PG1REG
2H	PG01CR
3H	
4H	PG01CR2
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[10] High speed serial channels

address	register name
0C00H	HSC0MD
1H	HSC0MD
2H	HSC0CT
3H	HSC0CT
4H	HSC0ST
5H	HSC0ST
6H	HSC0CR
7H	HSC0CR
8H	HSC0IS
9H	HSC0IS
AH	HSC0WE
BH	HSC0WE
CH	HSC0IE
DH	HSC0IE
EH	HSC0IR
FH	HSC0IR

address	register name
0C10H	HSC0TD
1H	HSC0TD
2H	HSC0RD
3H	HSC0RD
4H	HSC0TS
5H	HSC0TS
6H	HSC0RS
7H	HSC0RS
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

address	register name
0C20H	HSC1MD
1H	HSC1MD
2H	HSC1CT
3H	HSC1CT
4H	HSC1ST
5H	HSC1ST
6H	HSC1CR
7H	HSC1CR
8H	HSC1IS
9H	HSC1IS
AH	HSC1WE
BH	HSC1WE
CH	HSC1IE
DH	HSC1IE
EH	HSC1IR
FH	HSC1IR

address	register name
0C30H	HSC1TD
1H	HSC1TD
2H	HSC1RD
3H	HSC1RD
4H	HSC1TS
5H	HSC1TS
6H	HSC1RS
7H	HSC1RS
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[11] UART/Serial channels

address	register name
1200H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	
7H	SIR0CR
8H	SC1BUF
9H	SC1CR
AH	SC1MOD0
BH	BR1CR
CH	BR1ADD
DH	SC1MOD1
EH	
FH	

address	register name
1210H	SC2BUF
1H	SC2CR
2H	SC2MOD0
3H	BR2CR
4H	BR2ADD
5H	SC2MOD1
6H	
7H	
8H	SC3BUF
9H	SC3CR
AH	SC3MOD0
BH	BR3CR
CH	BR3ADD
DH	SC3MOD1
EH	
FH	

[12] I^2CBUS/Serial channels

address	register name
1240H	SBI0CR1
1H	SBI0DBR
2H	I2C0AR
3H	SBI0CR2/SBI0SR
4H	SBI0BR0
5H	SBI0BR1
6H	
7H	
8H	SBI1CR1
9H	SBI1DBR
AH	I2C1AR
BH	SBI1CR2/SBI1SR
CH	SBI1BR0
DH	SBI1BR1
EH	
FH	

[13] AD converter

address	register name
12A0H	ADREG0L
1H	ADREG0H
2H	ADREG1L
3H	ADREG1H
4H	ADREG2L
5H	ADREG2H
6H	ADREG3L
7H	ADREG3H
8H	ADREG4L
9H	ADREG4H
AH	ADREG5L
BH	ADREG5H
CH	ADREG6L
DH	ADREG6H
EH	ADREG7L
FH	ADREG7H

address	register name
12B0H	ADREG8L
1H	ADREG8H
2H	ADREG9L
3H	ADREG9H
4H	ADREGAL
5H	ADREGAH
6H	ADREGBL
7H	ADREGBH
8H	ADMOD0
9H	ADMOD1
AH	ADMOD2
BH	
CH	
DH	
EH	
FH	

[14] DA converter

address	register name
12E0H	DAC0REG
1H	DAC0CNT1
2H	
3H	DAC0CNT0
4H	DAC1REG
5H	DAC1CNT1
6H	
7H	DAC1CNT0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[15] Watch Dog Timer

address	register name
1300H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[16] Key-on wake up

address	register name
0090H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	KIEN
FH	KICR

(1) I/O port (1/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1	Port 1	0004H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Data from external port (Output latch register is cleared to “0”)							
P6	Port 6	0018H	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			Data from external port (Output latch register is cleared to “0”)							
P7	Port 7	001CH	P77	P76	P75	P74	P73	P72	P71	
			R/W							
			Data from external port (Output latch register is set to “1”)							
				Pull-up register 0:OFF 1:ON				Pull-up register 0:OFF 1:ON		
P8	Port 8	0020H	P87	P86	P85	P84	P83	P82	P81	P80
			R/W							
			Data from external port (Output latch register is set to “1”)		1	1	1	0	1	1
P9	Port 9	0024H		P96	P95	P94	P93	P92	P91	P90
				R/W						
				1	1	1	1	1	1	1
PA	Port A	0028H			PA5	PA4	PA3	PA2	PA1	PA0
					R/W					
					Data from external port (Output latch register is set to “1”)					
PC	Port C	0030H			PC5	PC4	PC3	PC2	PC1	PC0
					R/W					
					Data from external port (Output latch register is set to “1”)					
PD	Port D	0034H			PD5	PD4	PD3	PD2	PD1	PD0
					R/W					
					Data from external port (Output latch register is set to “1”)					
PF	Port F	003CH		PF6	PF5	PF4	PF3	PF2	PF1	PF0
				R/W						
				Data from external port (Output latch register is set to “1”)						
PJ	Port J	004CH	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
			R/W							
			Data from external port (Output latch register is set to “1”)							
PK	Port K	0050H	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
			R/W							
			Data from external port							
PL	Port L	0054H	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
			R/W							
			Data from external port (Output latch register is set to “1”)							
PM	Port M	0058H	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
			R							
			Input disable							
PN	Port N	005CH					PN3	PN2	PN1	PN0
							R			
							Input disable			

I/O port (2/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1CR	Port 1 control register	0006H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
P1FC	Port 1 function register	0007H (Prohibit RMW)								P1F
										W
										0
										0:Port 1:Data bus (D8 to D15)
P6CR	Port 6 control regiser	001AH (Prohibit RMW)	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			0	0	0	0	0	0	0	0
			0:Input 1:Output							
P6FC	Port 6 function register	001BH (Prohibit RMW)	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
			W							
			1	1	1	1	1	1	1	1
			0:Port 1:Address bus (A16 to A23)							
P7CR	Port 7 control register	001EH (Prohibit RMW)	P77C	P76C	P75C	P74C	P73C	P72C	P71C	
			W							
			0	0	0	0	0	0	0	
			0:Input 1:Output							
P7FC	Port 7 function register	001FH (Prohibit RMW)	P77F	P76F	P75F	P74F	P73F	P72F	P71F	
			W							
			0	0	0	0	0	0	0	
			0:Port 1: $\overline{WAIT}$	0:Port 1: $\overline{SRLUB}$	0:Port 1: $\overline{SRLLB}$	0:Port 1: $\overline{SRWR}$	0:Port 1:R/ $\overline{W}$	0:Port 1: $\overline{WRLU}$	0:Port 1: $\overline{WRLL}$	
P8CR	Port 8 control register	0021H (Prohibit RMW)	P87C	P86C						
			W							
			0	0						
			0:Input 1:Ouput							
P8FC	Port 8 control register	0022H (Prohibit RMW)	P87F	P86F	P85F	P84F	P83F	P82F	P81F	P80F
			W							
			0	0	0	0	0	0	0	0
			0:Port 1: $\overline{BUSAK}$	0:Port 1: $\overline{BUSRQ}$	0:Port 1:<P85F2>	0:Port 1: $\overline{CS4}$	0:Port 1:<P83F2>	0:Port 1: $\overline{CS2}$	0:Port 1: $\overline{CS1}$	0:Port 1: $\overline{CS0}$
P8FC2	Port 8 function register 2	0023H (Prohibit RMW)			P85F2		P83F2			
					W		W			
					0		0			
					0: $\overline{CS5}$ 1: $\overline{WDOUT}$		0: $\overline{CS3}$ 1: $\overline{SDCS}$			
P9DR	Port 9 drive register	0025H (Prohibit RMW)		P96D	P95D	P94D	P93D	P92D	P91D	P90D
				R/W						
				1	1	1	1	1	1	1
				0:The inside of HALT is figh impedance 1:The inside of HALT is also driven						
P9FC	Port 9 function register	0027H (Prohibit RMW)		P96F	P95F	P94F	P93F	P92F	P91F	P90F
				W						
				0	0	0	0	0	0	0
				0:Port 1:SDCLK	0:Port 1:SDCKE	0:Port 1:SDLUDQM	0:Port 1:SDLLDQM	0:Port 1: $\overline{SDCAS}$	0:Port 1: $\overline{SDRAS}$	0:Port 1: $\overline{SDWE}$

I/O Port (3/6)

PAFC2	Port A function register 2	0029H (Prohibit RMW)				PA4F2			PA1F2		
						W			W		
						0			0		
						<Refer to PAFC>			<Refer to PAFC>		
PACR	Port A control register	002AH (Prohibit RMW)			PA5C	PA4C	PA3C	PA2C	PA1C	PA0C	
					W						
					0	0	0	0	0	0	
					<Refer to PAFC>						
PAFC	Port A function register	002BH (Prohibit RMW)			PA5F	PA4F	PA3F	PA2F	PA1F	PA0F	
					W						
					0	0	0	0	0	0	

<PAxF2,PAxF,PAxC>	PA5	PA4	PA3	PA2	PA1	PA0
000	Input port	Input port	Input port	Input port	Input port	Input port
001	Output port	Output port	Output port	Output port	Output port	Output port
010	SCLK1/ $\overline{\text{CTS1}}$ input	Reserved	RXD1 input	SCLK0/ $\overline{\text{CTS0}}$ input	Reserved	RXD0 input
011	SCLK1 output	TXD1 output (Open Drain Disable)	Reserved	SCLK0 output	TXD0 output (Open Drain Disable)	Reserved
100		Reserved			Reserved	
101		Reserved			Reserved	
110		Reserved			Reserved	
111		TXD1 output (Open Drain Enable)			TXD0 output (Open Drain Enable)	

PCFC2	Port C function register 2	0031H (Prohibit RMW)				PC4F2	PC3F2			PC1F2	PC0F2
						W				W	
						0	0			0	0
						<Refer to PCFC>				<Refer to PCFC>	
PCCR	Port C control register	0032H (Prohibit RMW)			PC5C	PC4C	PC3C	PC2C	PC1C	PC0C	
					W						
					0	0	0	0	0	0	
					<Refer to PCFC>						
PCFC	Port C function register	0033H (Prohibit RMW)			PC5F	PC4F	PC3F	PC2F	PC1F	PC0F	
					W						
					0	0	0	0	0	0	

<PCxF2,PCxF,PCxC>	PC5	PC4	PC3	PC2	PC1	PC0
000	Input port	Input port	Input port	Input port	Input port	Input port
001	Output port	Output port	Output port	Output port	Output port	Output port
010	SCK1 input	SI1 input	SO1 output (Open Drain Disable)	SCK0 input	SI0 input	SO0 output (Open Drain Disable)
011	SCK1 output	SCL1 I/O (Open Drain Disable)	SDA1 I/O (Open Drain Disable)	SCK0 output	SCL0 I/O (Open Drain Disable)	SDA0 I/O (Open Drain Disable)
100		Reserved	Reserved		Reserved	Reserved
101		Reserved	Reserved		Reserved	Reserved
110		Reserved	SO1 output (Open Drain Enable)		Reserved	SO0 output (Open Drain Enable)
111		SCL1 I/O (Open Drain Enable)	SDA1 I/O (Open Drain Enable)		SCL0 I/O (Open Drain Enable)	SDA0 I/O (Open Drain Enable)

I/O Port (4/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PDFC2	Port D function register 2	0035H (Prohibit RMW)				PD4F2				
						W				
						0				
						<Refer to PDFC >				
PDCR	Port D control register	0036H (Prohibit RMW)			PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
						W				
					0	0	0	0	0	0
					<Refer to PDFC >					
PDFC	Port D function register	0037H (Prohibit RMW)			PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
					W					
					0	0	0	0	0	0

<PDxF2,PDxF,PDxC>	PD5	PD4	PD3	PD2	PD1	PD0
000	Input port	Input port	Input port	Input port	Input port	Input port
001	Output port	Output port	Output port	Output port	Output port	Output port
010	SCLK2/ $\overline{\text{CTS2}}$ input	Reserved	RXD2 input	Reserved	Reserved	HSSI0 input
011	SCLK2 output	TXD2 output (Open Drain Disable)	Reserved	HSCLK0 output	HSSO0 output	Reserved
100		Reserved				
101		Reserved				
110		Reserved				
111		TXD2 output (Open Drain				

PFFC2	Port F function register 2	003DH (Prohibit RMW)		PF6F2		PF4F2		PF2F2		PF0F2
				W		W		W		W
				0		0		0		0
				<Refer to PFFC>		<Refer to PFFC >		<Refer to PFFC >		<Refer to PFFC>
PFCR	Port F control register	003EH (Prohibit RMW)		PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
				W						
				0	0	0	0	0	0	0
				<Refer to PFFC >						
PFFC	Port F function register	003FH (Prohibit RMW)		PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
				W						
				0	0	0	0	0	0	0

<PFxF2,PFxF,PFxC>	PF6	PF5	PF4	PF3	PF2	PF1	PF0
000	Input port	Input port	Input port	Input port	Input port	Input port	Input port
001	Output port	Output port	Output port	Output port	Output port	Output port	Output port
010	TA6IN input	Reserved	Reserved	RXD2 input	Reserved	Reserved	HSSI0 input
011	Reserved	TA5OUT output	Reserved	TA3OUT output	Reserved	TA1OUT output	Reserved
100	Reserved		Reserved		Reserved		Reserved
101	Reserved		Reserved		Reserved		Reserved
110	INT3 input		INT2 input		INT1 input		INT0 input
111	Reserved		Reserved		Reserved		Reserved

I/O Port (5/6)

PJFC2 — Port J function register 2 — 004DH (Prohibit RMW)

PJ7F	PJ6F	PJ5F	PJ4F				
W							
0	0	0	0				
<Refer to PJFC >							

PJCR — Port J control register — 004EH (Prohibit RMW)

PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
W							
0	0	0	0	0	0	0	0
<Refer to PJFC >							

PJFC — Port J function register — 004FH (Prohibit RMW)

PJ7F	PJ6F	PJ5F	PJ4F	PJ3F	PJ2F	PJ1F	PJ0F
W							
0	0	0	0	0	0	0	0

<PJxF2,PJxF,PJxC>	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
000	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
001	Output port	Output port	Output port	Output port	Output port	Output port	Output port	Output port
010	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
011	TB3OUT1 output	TB3OUT0 output	TB2OUT1 output	TB2OUT0 output	TB1OUT1 output	TB1OUT0 output	TB0OUT1 output	TB0OUT0 output
100	Reserved	Reserved	Reserved	Reserved				
101	Reserved	Reserved	Reserved	Reserved				
110	Reserved	Reserved	Reserved	Reserved				
111	TB5OUT1 output	TB5OUT0 output	TB4OUT1 output	TB4OUT0 output				

PKFC2 — Port K function register 2 — 0051H (Prohibit RMW)

PK7F2	PK6F2	PK5F2	PK4F2	PK3F2	PK2F2	PK1F2	PK0F2
W							
0	0	0	0	0	0	0	0
<Refer to PKFC>							

PKFC — Port K function register — 0053H (Prohibit RMW)

PK7F	PK6F	PK5F	PK4F	PK3F	PK2F	PK1F	PK0F
W							
0	0	0	0	0	0	0	0

<PKxF2,PKxF>	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
00	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
01	TB3IN1 input	TB3IN0 input	TB2IN1 input	TB2IN0 input	TB1IN1 input	TB1IN0 input	TB0IN1 input	TB0IN0 input
10	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
11	INTB input	INTA input	INT9 input	INT8 input	INT7 input	INT6 input	INT5 input	INT4 input

I/O ポート(6/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PLFC2	Port L function register 2	0055H (Prohibit RMW)		PL6F2	PL5F2	PL4F2	PL3F2	PL2F2	PL1F2	PL0F2
				W						
				0	0	0	0	0	0	0
				<Refer to PLFC>						
PLCR	Port L function register	0056H (Prohibit RMW)	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
			W							
			0	0	0	0	0	0	0	0
			<Refer to PLFC >							
PLFC	Port L function register	0057H (Prohibit RMW)	PL7F	PL6F	PL5F	PL4F	PL3F	PL2F	PL1F	PL0F
			W							
			0	0	0	0	0	0	0	0

<PLxF2,PLxF,PLxC>	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
000	Input port	Input port	Input port	Input port	Input port	Input port	Input port	Input port
001	Output port	Output port	Output port	Output port	Output port	Output port	Output port	Output port
010	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
011	PG13 output	PG12 output	PG11 output	PG10 output	PG03 output	PG02 output	PG01 output	PG00 output
100		Reserved	Reserved	HSSI1 input	Reserved	SCLK3/ $\overline{\text{CTS3}}$ input	Reserved	RXD3 input
101		HSCLK1 output	HSSO1 output	Reserved	Reserved	SCLK3 output	TXD3 output (Open Drain Disable)	Reserved
110		Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
111		Reserved	Reserved	Reserved	TA7OUT output	Reserved	TXD3 output (Open Drain Enable)	Reserved

PMFC	Port M function register	005BH (Prohibit RMW)	PM7F	PM6F	PM5F	PM4F	PM3F	PM2F	PM1F	PM0F
			W							
			1	1	1	1	1	1	1	1
			0:Input port / Key input 1:Analog input							
PNFC	Port N function register	005FH (Prohibit RMW)					PN3F	PN2F	PN1F	PN0F
							W			
							1	1	1	1
							0:Input port 1:Analog input			

(2) Interrupt control (1/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE01	INT0 & INT1 Enable	00D0H	INT1				INT0			
			I1C	I1M2	I1M1	I1M0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INT1	Interrupt request level			1:INT0	Interrupt request level		
INTE23	INT2 & INT3 Enable	00D1H	INT3				INT2			
			I3C	I3M2	I3M1	I3M0	I2C	I2M2	I2M1	I2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INT3	Interrupt request level			1:INT2	Interrupt request level		
INTE45	INT4 & INT5 Enable	00D2H	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INT5	Interrupt request level			1:INT4	Interrupt request level		
INTE67	INT6 & INT7 Enable	00D3H	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INT7	Interrupt request level			1:INT6	Interrupt request level		
INTETA01	INTTA0 & INTTA1 Enable	00D4H	INTTA1 (TMRA1)				INTTA0 (TMRA0)			
			ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTA1	Interrupt request level			1:INTTA0	Interrupt request level		
INTETA23	INTTA2 & INTTA3 Enable	00D5H	INTTA3 (TMRA3)				INTTA2 (TMRA2)			
			ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTA3	Interrupt request level			1:INTTA2	Interrupt request level		
INTE8TA45	INTTA4 & INT8/INTTA5 Enable	00D6H	INT8/INTTA5 (TMRA5)				INTTA4 (TMRA4)			
			ITA5C	ITA5M2	ITA5M1	ITA5M0	ITA4C	ITA4M2	ITA4M1	ITA4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INT8/ INTTA5	Interrupt request level			1:INTTA4	Interrupt request level		
INTE9TA67	INTTA6 & INT9/INTTA7 Enable	00D7H	INT9/INTTA7 (TMRA7)				INTTA6 (TMRA6)			
			ITA7C	ITA7M2	ITA7M1	ITA7M0	ITA6C	ITA6M2	ITA6M1	ITA6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INT9/ INTTA7	Interrupt request level			1:INTTA6	Interrupt request level		

Interrupt control (2/5)

INTES0	INTRX0 & INTTX0 Enable	00D8H	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTX0	Interrupt request level			1:INTRX0	Interrupt request level		
INTES1	INTRX1 & INTTX1 Enable	00D9H	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTX1	Interrupt request level			1:INTRX1	Interrupt request level		
INTES2	INTRX2 & INTTX2 Enable	00DAH	INTTX2				INTRX2			
			ITX2C	ITX2M2	ITX2M1	ITX2M0	IRX2C	IRX2M2	IRX2M1	IRX2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTX2	Interrupt request level			1:INTRX2	Interrupt request level		
INTES3	INTRX3 & INTTX3 Enable	00DBH	INTTX3				INTRX3			
			ITX3C	ITX3M2	ITX3M1	ITX3M0	IRX3C	IRX3M2	IRX3M1	IRX3M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTX3	Interrupt request level			1:INTRX3	Interrupt request level		
INTESB0	INTSBE0 Enable	00DCH	-				INTSBE0			
			-	-	-	-	ISBE0C	ISBE0M2	ISBE0M1	ISBE0M0
							R	R/W		
							0	0	0	0
			Always write “0”				1:INTSBE0	Interrupt request level		
INTESB1	INTSBE1 Enable	00DDH	-				INTSBE1			
			-	-	-	-	ISBE1C	ISBE1M2	ISBE1M1	ISBE1M0
							R	R/W		
							0	0	0	0
			Always write “0”				1:INTSBE1	Interrupt request level		
INTEAHSC0	INTA & INTHSC0 Enable	00DEH	INTHSC0				INTA			
			IHSC0C	IHSTX0M2	IHSTX0M1	IHSTX0M0	IAC	IAM2	IAM1	IAM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTHSC0	Interrupt request level			1:INTA	Interrupt request level		
INTEBHSC1	INTB & INTHSC1 Enable	00DFH	INTHSC1				INTB			
			IHSC1C	IHSTX1M2	IHSTX1M1	IHSTX1M0	IBC	IBM2	IBM1	IBM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTHSC1	Interrupt request level			1:INTB	Interrupt request level		

Interrupt control (3/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTETB0	INTTB00 & INTTB01 Enable	00E0H	INTTB01 (TMRB0)				INTTB00 (TMRB0)			
			ITB01C	ITB01M2	ITB01M1	ITB01M0	ITB00C	ITB00M2	ITB00M1	ITB00M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTB01	Interrupt request level			1:INTETB00	Interrupt request level		
INTETB1	INTTB10 & INTTB11 Enable	00E2H	INTTB11 (TMRB1)				INTTB10 (TMRB1)			
			ITB11C	ITB11M2	ITB11M1	ITB11M0	ITB10C	ITB10M2	ITB10M1	ITB10M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTB11	Interrupt request level			1:INTTB10	Interrupt request level		
INTETB2	INTTB20 & INTTB21 Enable	00E5H	INTTB21 (TMRB2)				INTTB20 (TMRB2)			
			ITB21C	ITB21M2	ITB21M1	ITB21M0	ITB20C	ITB20M2	ITB20M1	ITB20M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTB21	Interrupt request level			1:INTTB20	Interrupt request level		
INTETB3	INTTB30 & INTTB31 Enable	00E6H					INTTB31/INTTB30 (TMRB3)			
							ITB3XC	ITB3XM2	ITB3XM1	ITB3XM0
							R	R/W		
							0	0	0	0
				Always write “0”			1:INTTB31/30	Interrupt request level		
INTETB4	INTTB40 & INTTB41 Enable	00E7H					INTTB41/INTTB40 (TMRB4)			
							ITB4XC	ITB4XM2	ITB4XM1	ITB4XM0
							R	R/W		
							0	0	0	0
				Always write “0”			1:INTTB41/40	Interrupt request level		
INTETB5	INTTB50 & INTTB51 Enable	00E8H					INTTB51/INTTB50 (TMRB5)			
							ITB5XC	ITB5XM2	ITB5XM1	ITB5XM0
							R	R/W		
							0	0	0	0
				Always write “0”			1:INTTB51/50	Interrupt request level		
INTETBOX	INTTBOX (Overflow) Enable	00E9H					INTTBOX			
							ITBOXC	ITBOXM2	ITBOXM1	ITBOXM0
							R	R/W		
							0	0	0	0
				Always write “0”			1:INTTBOX	Interrupt request level		

Interrupt control (4/5)

INTEPAD	INTP0 & INTAD Enable	00E4H	INTP0				INTAD			
			IP0C	IP0M2	IP0M1	IP0M0	IADC	IADM2	IADM1	IADM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTP0	Interrupt request level			1:INTAD	Interrupt request level		
INTNMWDT	NMI & INTWDT Enable	00EFH	NMI				INTWDT			
			INCNM	-	-	-	INCWD	-	-	-
			R				R			
			0				0			
			1:NMI				1:INTWDT			
INTETC01	INTTC0 & INTTC1 Enable	00F0H	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
			1:INTTC1	Interrupt request level			1:INTTC0	Interrupt request level		
INTETC23	INTTC2 & INTTC3 Enable	00F1H	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
			1:INTTC3	Interrupt request level			1:INTTC2	Interrupt request level		
INTETC45	INTTC4 & INTTC5 Enable	00F2H	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
			1:INTTC5	Interrupt request level			1:INTTC4	Interrupt request level		
INTETC67	INTTC6 & INTTC7 Enable	00F3H	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
			1:INTTC7	Interrupt request level			1:INTTC6	Interrupt request level		

Interrupt control (5/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SIMC	SIO Interrupt Mode control	00F5 (Prohibit RMW)	-				IR3LE	IR2LE	IR1LE	IR0LE
			W				R/W			
			0				1	1	1	1
			Always write "1"				INTRX3 0: edge mode 1: level mode	INTRX2 0: edge mode 1: level mode	INTRX1 0: edge mode 1: level mode	INTRX0 0: edge mode 1: level mode
IIMC0	Interrupt Input mode control 0	00F6H (Prohibit RMW)								NMIREE
										R/W
										0
										$\overline{\text{NMI}}$ 0:Falling 1:Falling and Rising
IIMC1	Interrupt Input mode control 1	00FAH (Prohibit RMW)	I7LE	I6LE	I5LE	I4LE	I3LE	I2LE	I1LE	I0LE
			R/W							
			0	0	0	0	0	0	0	0
			INT7 0:Edge 1:Level	INT6 0:Edge 1:Level	INT5 0:Edge 1:Level	INT4 0:Edge 1:Level	INT3 0:Edge 1:Level	INT2 0:Edge 1:Level	INT1 0:Edge 1:Level	INT0 0:Edge 1:Level
IIMC2	Interrupt Input mode control 2	00FBH (Prohibit RMW)	I7EDGE	I6EDGE	I5EDGE	I4EDGE	I3EDGE	I2EDGE	I1EDGE	I0EDGE
			R/W							
			0	0	0	0	0	0	0	0
			INT7 0: Rising /High 1: Falling /Low	INT6 0: Rising /High 1: Falling /Low	INT5 0: Rising /High 1: Falling /Low	INT4 0: Rising /High 1: Falling /Low	INT3 0: Rising /High 1: Falling /Low	INT2 0: Rising /High 1: Falling /Low	INT1 0: Rising /High 1: Falling /Low	INT0 0: Rising /High 1: Falling /Low
IIMC3	Interrupt Input mode control 3	010EH (Prohibit RMW)					IBLE	IALE	I9LE	I8LE
							R/W			
							0	0	0	0
							INTB 0:Edge 1:Level	INTA 0:Edge 1:Level	INT9 0:Edge 1:Level	INT8 0:Edge 1:Level
IIMC4	Interrupt Input mode control 4	010FH (Prohibit RMW)					IBEDGE	IAEDGE	I9EDGE	I8EDGE
							R/W			
							0	0	0	0
							INTB 0: Rising /High 1: Falling /Low	INTA 0: Rising /High 1: Falling /Low	INT9 0: Rising /High 1: Falling /Low	INT8 0: Rising /High 1: Falling /Low
INTCLR	Interrupt Clear Control	00F8H (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			0	0	0	0	0	0	0	0
			Clear the interrupt request flag by the writing of a micro DMA starting vector							
INTSEL	Interruption combination selection	010CH (Prohibit RMW)	-	DP49SEL	DP48SEL	DP47SEL	DP39SEL	DP37SEL	DP26SEL	DP24SEL
				R/W						
				0	0	0	0	0	0	0
				0:INTTB50 Interruption is effective 1:INTTB51 Interruption is effective	0:INTTB40 Interruption is effective 1:INTB41 Interruption is effective	0:INTTB30 Interruption is effective 1:INTTB31 Interruption is effective	0:INTB Interruption is invalid 1:INTB Interruption is effective	0:INTA Interruption is invalid 1:INTA Interruption is effective	0:INTTA7 Interruption is effective 1:INT9 Interruption is effective	0:INTTA5 Interruption is effective 1:INT8 Interruption is effective
INTST	Interruption generating flag	010DH (Prohibit RMW)			TBOF5ST	TBOF4ST	TBOF3ST	TBOF2ST	TBOF1ST	TBOF0ST
					R/W					
					0	0	0	0	0	0
					Read: 0:Interruption in-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption in-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption in-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption in-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption in-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care	Read: 0:Interruption in-generating 1:Interruption generating Write: 0:"0" clear 1:Don't care

(3) DMA controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA0 Start Vector	0100H			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
					DMA0 Start Vector					
DMA1V	DMA1 Start Vector	0101H			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
					DMA1 Start Vector					
DMA2V	DMA2 Start Vector	0102H			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
					DMA2 Start Vector					
DMA3V	DMA3 Start Vector	0103H			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
					0	0	0	0	0	0
					DMA3 Start Vector					
DMA4V	DMA4 Start Vector	0104H			DMA4V5	DMA4V4	DMA4V3	DMA4V2	DMA4V1	DMA4V0
					R/W					
					0	0	0	0	0	0
					DMA4 Start Vector					
DMA5V	DMA5 Start Vector	0105H			DMA5V5	DMA5V4	DMA5V3	DMA5V2	DMA5V1	DMA5V0
					R/W					
					0	0	0	0	0	0
					DMA5 Start Vector					
DMA6V	DMA6 Start Vector	0106H			DMA6V5	DMA6V4	DMA6V3	DMA6V2	DMA6V1	DMA6V0
					R/W					
					0	0	0	0	0	0
					DMA6 Start Vector					
DMA7V	DMA7 Start Vector	0107H			DMA7V5	DMA7V4	DMA7V3	DMA7V2	DMA7V1	DMA7V0
					R/W					
					0	0	0	0	0	0
					DMA7 Start Vector					
DMAB	DMA Burst	0108H	DBST7	DBST6	DBST5	DBST4	DBST3	DBST2	DBST1	DBST0
			R/W							
			0	0	0	0	0	0	0	0
			1: DMA request on Burst Mode							
DMAR	DMA Request	0109H (Prohibit RMW)	DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
			R/W							
			0	0	0	0	0	0	0	0
			1:DMA request in software							

(4) Memory controller (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CSL	BLOCK 0 MEMC control register Low	0140H (Prohibit RMW)		B0WW2	B0WW1	B0WW0		B0WR2	B0WR1	B0WR0
				W					W	
				0	1	0		0	1	0
				Write waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved				Read waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved		
B0CSH	BLOCK 0 MEMCT control register High	0141H (Prohibit RMW)	B0E	-	-	B0REC	B0OM1	B0OM0	B0BUS1	B0BUS0
			W				W			
			0			0	0	0	0	0
			CS select 0:Disable 1:Enable	Always write "0"	Always write "0"	0:Not insert a dummy cycle 1:Insert a dummy cycle	00:ROM/SRAM 01:Reserved 10:Reserved 11:Reserved		Data Bus width 00:8bit 01:16bit 10:Reserved 11:Reserved	
B1CSL	BLOCK 1 MEMC control register Low	0144H (Prohibit RMW)		B1WW2	B1WW1	B1WW0		B1WR2	B1WR1	B1WR0
				W					W	
				0	1	0		0	1	0
				Write waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved				Read waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved		
B1CSH	BLOCK 1 MEMC control register High	0145H (Prohibit RMW)	B1E	-	-	B1REC	B1OM1	B1OM0	B1BUS1	B1BUS0
			W				W			
			0			0	0	0	0	0
			CS select 0:Disable 1:Enable	Always write "0"	Always write "0"	0: Not insert a dummy cycle 1: Insert a dummy cycle	00:ROM/SRAM 01:Reserved 10:Reserved 11:Reserved		Data Bus width 00:8bit 01:16bit 10:Reserved 11:Reserved	
B2CSL	BLOCK 2 MEMC control register Low	0148H (Prohibit RMW)		B2WW2	B2WW1	B2WW0		B2WR2	B2WR1	B2WR0
				W					W	
				0	1	0		0	1	0
				Write waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved				Read waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved		
B2CSH	BLOCK 2 MEMC control register High	0149H (Prohibit RMW)	B2E	B2M	-	B2REC	B2OM1	B2OM0	B2BUS1	B2BUS0
			W			W				
			1	0		0	0	0	0/1	0/1
			CS select 0:Disable 1:Enable	0:16MB 1:Sets area	Always write "0"	0: Not insert a dummy cycle 1: Insert a dummy cycle	00:ROM/SRAM 01:Reserved 10:Reserved 11:Reserved		Data Bus width 00:8bit 01:16bit 10:Reserved 11:Reserved	
B3CSL	BLOCK 3 MEMC control register Low	014CH (Prohibit RMW)		B3WW2	B3WW1	B3WW0		B3WR2	B3WR1	B3WR0
				W					W	
				0	1	0		0	1	0
				Write waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved				Read waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved		
B3CSH	BLOCK 3 MEMC control register High	014DH (Prohibit RMW)	B3E	-	-	B3REC	B3OM1	B3OM0	B3BUS1	B3BUS0
			W				W			
			0			0	0	0	0	0
			CS select 0:Disable 1:Enable	Always write "0"	Always write "0"	0: Not insert a dummy cycle 1: Insert a dummy cycle	00:ROM/SRAM 01:Reserved 10:Reserved 11:SDRAM		Data Bus width 00:8bit 01:16bit 10:Reserved 11:Reserved	

Note1: A value is set to B2CSH <B2BUS1:0> according to the state of AM[1:0] terminal at the time of reset release.

Memory controller (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B4CSL	BLOCK 4 MEMC control register Low	0150H (Prohibit RMW)		B4WW2	B4WW1	B4WW0		B4WR2	B4WR1	B4WR0
				W					W	
				0	1	0		0	1	0
				Write waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved				Read waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved		
B4CSH	BLOCK 4 MEMC control register High	0151H (Prohibit RMW)	B4E	-	-	B4REC	B4OM1	B4OM0	B4BUS1	B4BUS0
			W			W				
			0			0	0	0	0	0
			CS select 0:Disable 1:Enable	Always write "0"	Always write "0"	0: Not insert a dummy cycle 1: Insert a dummy cycle	00:ROM/SRAM 01:Reserved 10:Reserved 11:Reserved		Data Bus width 00:8bit 01:16bit 10:Reserved 11:Reserved	
B5CSL	BLOCK 4 MEMC control register Low	0154H (Prohibit RMW)		B5WW2	B5WW1	B5WW0		B5WR2	B5WR1	B5WR0
				W					W	
				0	1	0		0	1	0
				Write waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved				Read waits 001:0WAIT 010:1WAIT 101:2WAIT 110:3WAIT 111:4WAIT 011: $\overline{WAIT}$ pin Others:Reserved		
B5CSH	BLOCK 4 MEMC control register High	0155H (Prohibit RMW)	B5E	-	-	B5REC	B5OM1	B5OM0	B5BUS1	B5BUS0
			W			W				
			0			0	0	0	0	0
			CS select 0:Disable 1:Enable	Always write "0"	Always write "0"	0: Not insert a dummy cycle 1: Insert a dummy cycle	00:ROM/SRAM 01:Reserved 10:Reserved 11:Reserved		Data Bus width 00:8bit 01:16bit 10:Reserved 11:Reserved	
BEXCSL	BLOCK EX MEMC control register Low	0158H		BEXWW2	BEXWW1	BEXWW0		BEXWR2	BEXWR1	BEXWR0
				W					W	
				0	1	0		0	1	0
				Write waits 001:2WAIT 010:1WAIT 101:2WAIT 110:2WAIT 011:1+NWAIT Others:Reserved				Read waits 001:2WAIT 010:1WAIT 101:2WAIT 110:2WAIT 011:1+NWAIT Others:Reserved		
BEXCSH	BLOCK EX MEMC control register High	0159H					BEXOM1	BEXOM0	BEXBUS1	BEXBUS0
							W			
			-	0	0	0	0	0	0	0
			-	Always write "0"	Always write "0"	Always write "0"	00:ROM/SRAM 01:Reserved 10:Reserved 11:Reserved		Data Bus width 00:8bit 01:16bit 10:Reserved 11:Reserved	
PMEMCR	Page ROM control register	0166H	-	-	-	OPGE	OPWR1	OPWR0	PR1	PR0
							R/W			
						0	0	0	1	0
						ROM page access 0:Disable 1:Enable	Wait number on page 00: 1state (n-1-1-1 mode) 01: 2state (n-2-2-2 mode) 10: 3state (n-3-3-3 mode) 11: Reserved		Byte number in a page 00: 64byte 01: 32byte 10: 16byte 11: 8byte	

Memory controller (3/3)

MAMR0	Memory mask register 0	0142H	M0V20	M0V19	M0V18	M0V17	M0V16	M0V15	M0V14-9	M0V8
			R/W							
			1	1	1	1	1	1	1	1
			0:Compare enable 1:Compare disable							
MSAR0	Memory start address register 0	0143H	M0S23	M0S22	M0S21	M0S20	M0S19	M0S18	M0S17	M0S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
MAMR1	Memory mask register 1	0146H	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	MV15-9	M1V8
			R/W							
			1	1	1	1	1	1	1	1
			0:Compare enable 1:Compare disable							
MSAR1	Memory start address register 1	0147H	M1S23	M1S22	M1S21	M1S20	M1S19	M1S18	M1S17	M1S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
MAMR2	Memory mask register 2	014AH	M2V22	M2V21	M2V20	M2V19	M2V18	M2V17	M2V16	M2V15
			R/W							
			1	1	1	1	1	1	1	1
			0:Compare enable 1:Compare disable							
MSAR2	Memory start address register 2	014BH	M2S23	M2S22	M2S21	M2S20	M2S19	M2S18	M2S17	M2S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
MAMR3	Memory mask register 3	014EH	M3V22	M3V21	M3V20	M3V19	M3V18	M3V17	M3V16	M3V15
			R/W							
			1	1	1	1	1	1	1	1
			0:Compare enable 1:Compare disable							
MSAR3	Memory start address register 3	014FH	M3S23	M3S22	M3S21	M3S20	M3S19	M3S18	M3S17	M3S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
MAMR4	Memory mask register 4	0152H	M4V22	M4V21	M4V20	M4V19	M4V18	M4V17	M4V16	M4V15
			R/W							
			1	1	1	1	1	1	1	1
			0:Compare enable 1:Compare disable							
MSAR4	Memory start address register 4	0153H	M4S23	M4S22	M4S21	M4S20	M4S19	M4S18	M4S17	M4S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
MAMR5	Memory mask register 5	0156H	M5V22	M5V21	M5V20	M5V19	M5V18	M5V17	M5V16	M5V15
			R/W							
			1	1	1	1	1	1	1	1
			0:Compare enable 1:Compare disable							
MSAR5	Memory start address register 5	0157H	M5S23	M5S22	M5S21	M5S20	M5S19	M5S18	M5S17	M5S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							

(5) Clock control / PLL (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SYSCR0	System Clock Control 0	10E0H						-		
								R/W		
								0	0	0
								Always write "0"		
SYSCR1	System Clock Control 1	10E1H						GEAR2	GEAR1	GEAR0
								R/W		
								1	0	0
								Select gear value of high frequency (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (Reserved) 110: (Reserved) 111: (Reserved)		
SYSCR2	System Clock Control 2	10E2H	-		WUPTM1	WUPTM0	HALTM1	HALTM0		DRVE
			R/W		R/W					R/W
			0		1	0	1	1		0
			Always write "0"		Warm-up timer 00: Reserved 01: 2^8/input frequency 10: 2^{14}/input frequency 11: 2^{16}/input frequency		HALT mode 00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode			Pin state control in STOP mode 0: I/O off 1: Remains the state before HALT
PLLCR0	PLL Control 0	10E8H		FCSEL	LWUPFG					
				R/W	R					
				0	0					
				Select fc clock 0: f_{OSCH} 1: f_{PLL}	Lock up timer status flag 0: not end 1: end					
PLLCR1	PLL Control 1	10E9H	PLLON							
			R/W							
			0							
			Control on/off 1: ON 0: OFF							

Clock control / PLL (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
EMCCR0	EMC control register 0	10E3H	PROTECT					EXTIN	DRVOSCH	
			R					R/W	R/W	
			0					0	1	
			Protect flag 0: OFF 1: ON					1: External fc clock	fc oscillator driver abillity 1: NORMAL 0: WEAK	
EMCCR1	EMC control register 1	10E4H	Switching the protect ON/OFF by write to following 1st-KEY, 2nd-KEY 1st-KEY: EMCCR1 = 5AH, EMCCR2 = A5H in succession write 2nd-KEY: EMCCR1 = A5H, EMCCR2 = 5AH in succession write							
EMCCR2	EMC control register 2	10E5H								

(6) SDRAM Controller

Symbol	Name	address	7	6	5	4	3	2	1	0
SDACR1	SDRAM Access Control Register1	0250H	-	-	SMRD	SWRC	SBST	SBL1	SBL0	SMAC
			R/W							
			0	0	0	0	0	1	0	0
			Always write “0”	Always write “0”	Mode register recovery time 0: 1clock 1: 2clock	Write recovery time 0: 1clock 1: 2clock	Burst stop command 0: Precharge all 1: Burst stop	Select burst length 00: Reserved 01:Full-page read, burst write 10:1-word read, single write 11:Full-page read, single write		SDRAM controller 0: Disable 1: Enable
SDACR2	SDRAM Access Control Register2	0251H				SBS	SDRS1	SDRS0	SMUXW1	SMUXW0
						R/W				
						0	0	0	0	0
						Number of banks 0: 2 banks 1: 4 banks	Select ROW address size 00: 2048rows (11bits) 01: 4096rows (12bits) 10: 8192rows (13bits) 11: Reserved		Select address multiplex 00: TypeA (A9-) 01: TypeB (A10-) 10: TypeC (A11-) 11: Reserved	
SDRCR	SDRAM Refresh Control Register	0252H				SSAE	SRS2	SRS1	SRS0	SRC
						R/W				
						1	0	0	0	0
						SR Auto Exit 0:Disable 1:Enable	Refresh interval 000: 47state 100:156state 001: 78state 101:295state 010: 97state 110:249state 011:124state 111:312state			Auto refresh 0:Disable 1:Enable
SDCMM	SDRAM Commandl Register	0253H						SCMM2	SCMM1	SCMM0
								R/W		
								0	0	0
								Command executing 000: Not execute 001: Excute initialize command a. Precharge all banks b. 8 times auto refresh c. Set mode register 100: Set mode register 101: Execute self refresh Entry 110: Execute self refresh EXIT Others: Reserved		

(7) 8-bit timer (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA01RUN	TMRA01 RUN register	1100H	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
			R/W				R/W			
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TA0REG	TMRA0 register	1102H (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			Undefined							
TA1REG	TMRA1 register	1103H (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			Undefined							
TA01MOD	TMRA01 MODE register	1104H	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			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		TMRA1 source clock 00: TA0TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA0 source clock 00: TA0IN input 01: ϕT1 10: ϕT4 11: ϕT16	
TA1FFCR	TMRA01 Flip-Flop Conttol register	1105H					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA1FF 01: Set TA1FF 10: Clear TA1FF 11: Don't care		TA1FF Control for inversion 0: Disable 1: Enable	TA1FF Inversion select 0: TMRA0 1: TMRA1
TA23RUN	TMRA23 RUN register	1108H	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
			R/W				R/W			
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TA2REG	TMRA2 register	110AH (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			Undefined							
TA3REG	TMRA3 register	110BH (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			Undefined							
TA23MOD	TMRA23 MODE register	110CH	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			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		TMRA3 source clock 00: TA2TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA2 source clock 00: TA2IN input 01: ϕT1 10: ϕT4 11: ϕT16	
TA3FFCR	TMRA23 Flip-Flop Conttol register	110DH					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA3FF 01: Set TA3FF 10: Clear TA3FF 11: Don't care		TA3FF Control for inversion 0: Disable 1: Enable	TA3FF Inversion select 0: TMRA2 1: TMRA3

8-bit timer (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA45RUN	TMRA45 RUN register	1110H	TA4RDE				I2TA45	TA45PRUN	TA5RUN	TA4RUN
			R/W				R/W			
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TA4REG	TMRA4 register	1112H (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			Undefined							
TA5REG	TMRA5 register	1113H (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			Undefined							
TA45MOD	TMRA45 MODE register	1114H	TA45M1	TA45M0	PWM41	PWM40	TA5CLK1	TA5CLK0	TA4CLK1	TA4CLK0
			R/W							
			0	0	0	0	0	0	0	0
			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		TMRA5 source clock 00: TA4TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA4 source clock 00: TA4IN input 01: ϕT1 10: ϕT4 11: ϕT16	
TA5FFCR	TMRA45 Flip-Flop Conttol register	1115H					TA5FFC1	TA5FFC0	TA5FFIE	TA5FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA5FF 01: Set TA5FF 10: Clear TA5FF 11: Don't care		TA5FF Control for inversion 0: Disable 1: Enable	TA5FF Inversion select 0: TMRA4 1: TMRA5
TA67RUN	TMRA67 RUN register	1118H	TA6RDE				I2TA67	TA67PRUN	TA7RUN	TA6RUN
			R/W				R/W			
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TA6REG	TMRA6 register	111AH (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			Undefined							
TA7REG	TMRA7 register	111BH (Prohibit RMW)	-	-	-	-	-	-	-	-
			W							
			Undefined							
TA67MOD	TMRA67 MODE register	111CH	TA67M1	TA67M0	PWM61	PWM60	TA7CLK1	TA7CLK0	TA6CLK1	TA6CLK0
			R/W							
			0	0	0	0	0	0	0	0
			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		TMRA7 source clock 00: TA6TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA6 source clock 00: TA6IN input 01: ϕT1 10: ϕT4 11: ϕT16	
TA7FFCR	TMRA67 Flip-Flop Conttol register	111DH					TA7FFC1	TA7FFC0	TA7FFIE	TA7FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA7FF 01: Set TA7FF 10: Clear TA7FF 11: Don't care		TA7FF Control for inversion 0: Disable 1: Enable	TA7FF Inversion select 0: TMRA6 1: TMRA7

(8) 16-bit timer (1/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB0RUN	TMRB0 RUN register	1180H	TB0RDE	–			I2TB0	TB0PRUN		TB0RUN
			R/W	R/W			R/W	R/W		R/W
			0	0			0	0		0
			Double Buffer 0: Disable 1: Enable	Always write "0"			IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TB0MOD	TMRB0 MODE register	1182H (Prohibit RMW)	TB0CT1	TB0ET1	TB0CP0I	TB0CPM1	TB0CPM0	TB0CLE	TB0CLK1	TB0CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TB0FF1 inversion trigger 0: Disable 1: Enable		Software capture control 0: Software capture 1: Undefind	Capture timing 00: Disable INT4 is rising edge 01: TB0IN0 ↑, TB0IN1 ↑ INT4 is rising edge 10: TB0IN0 ↑, TB0IN0 ↓ INT4 is falling edge 11: TA1OUT ↑, TA1OUT ↓ INT4 is rising edge		Up counter control 0: Disable 1: Enable	TMRB0 source clock 00:TB0IN0 pin input 01: φT1 10: φT4 11: φT16	
			Invert when capture to capture register 1	Invert when match UC0 with TB0RG1H/L						
TB0FFCR	TMRB0 Flip-Flop control register	1183H (Prohibit RMW)	TB0FF1C1	TB0FF1C0	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
			W*						W*	
			1	1	0	0	0	0	1	1
			TB0FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11"		TB0FF0 inversion trigger 0: Disable 1: Enable				TB0FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11"	
					Invert when the UC value is loaded in to TB0CP1H/L	Invert when the UC value is loaded in to TB0CP0H/L	Invert when the UC matches with TB0RG1H/L	Invert when the UC matches with TB0RG0H/L		
TB0RG0L	TMRB0 register 0 Low	1188H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB0RG0H	TMRB0 register 0 High	1189H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB0RG1L	TMRB0 register 1 Low	118AH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB0RG1H	TMRB0 register 1 High	118BH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB0CP0L	TMRB0 Capture register 0 Low	118CH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB0CP0H	TMRB0 Capture register 0 High	118DH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB0CP1L	TMRB0 Capture register 1 Low	118EH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB0CP1H	TMRB0 Capture register 1 High	118FH	–	–	–	–	–	–	–	–
			R							
			Undefined							

16-bit timer (2/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB1RUN	TMRB1 RUN register	1190H	TB1RDE	–			I2TB1	TB1PRUN		TB1RUN
			R/W	R/W			R/W	R/W		R/W
			0	0			0	0		0
			Double Buffer 0: Disable 1: Enable	Always write "0"			IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TB1MOD	TMRB1 MODE register	1192H (Prohibit RMW)	TB1CT1	TB1ET1	TB1CP0I	TB1CPM1	TB1CPM0	TB1CLE	TB1CLK1	TB1CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TB1FF1 inversion trigger 0: Disable 1: Enable		Software capture control 0: Software capture 1: Undefined	Cature timing 00: Disable INT6 is rising edge 01:TB1IN0 ↑ , TB1IN1 ↑ INT6 is rising edge 10: TB1IN0 ↑ , TB1IN0 ↓ INT6 is falling edge 11: TA1OUT ↑ , TA1OUT ↓ INT6 is rising edge		Up counter control 0: Disable 1: Enable	TMRB1 source clock 00: TB1IN0 pin input 01: φT1 10: φT4 11: φT16	
			Invert when capture to capture register 1	Invert when match UC1 with TB1RG1H/L						
TB1FFCR	TMRB1 Flip-Flop control register	1193H (Prohibit RMW)	TB1FF1C1	TB1FF1C0	TB1C1T1	TB1C0T1	TB1E1T1	TB1E0T1	TB1FF0C1	TB1FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			TB1FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11"		TB1FF0 inversion trigger 0: Disable 1: Enable				TB1FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11"	
					Invert when the UC value is loaded in to TB1CP1H/L	Invert when the UC value is loaded in to TB1CP0H/L	Invert when the UC matches with TB1RG1H/L	Invert when the UC matches with TB1RG0H/L		
TB1RG0L	TMRB1 register 0 Low	1198H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB1RG0H	TMRB1 register 0 High	1199H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB1RG1L	TMRB1 register 1 Low	119AH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB1RG1H	TMRB1 register 1 High	119BH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB1CP0L	TMRB1 Capture register 0 Low	119CH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB1CP0H	TMRB1 Capture register 0 High	119DH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB1CP1L	TMRB1 Capture register 1 Low	119EH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB1CP1H	TMRB1 Capture register 1 High	119FH	–	–	–	–	–	–	–	–
			R							
			Undefined							

16-bit timer (3/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB2RUN	TMRB2 RUN register	11A0H	TB2RDE	–			I2TB2	TB2PRUN		TB2RUN
			R/W	R/W			R/W	R/W		R/W
			0	0			0	0		0
			Double Buffer 0: Disable 1: Enable	Always write "0"			IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TB2MOD	TMRB2 MODE register	11A2H (Prohibit RMW)	TB2CT1	TB2ET1	TB2CP0I	TB2CPM1	TB2CPM0	TB2CLE	TB2CLK1	TB2CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TB2FF1 inversion trigger 0: Disable 1: Enable		Software capture control 0: Software capture 1: Undefined	Capture timing 00: Disable INT8 is rising edge 01: TB2IN0 ↑ , TB2IN1 ↑ INT8 is rising edge 10: TB2IN0 ↑ , TB2IN0 ↓ INT8 is falling edge 11: TA3OUT ↑ , TA3OUT ↓ INT8 is rising edge		Up counter control 0: Disable 1: Enable	TMRB2 source clock 00: TB2IN0 pin input 01: φT1 10: φT4 11: φT16	
			Invert when capture to capture register 2	Invert when match UC2 with TB2RG1H/L						
TB2FFCR	TMRB2 Flip-Flop control register	11A3H (Prohibit RMW)	TB2FF1C1	TB2FF1C0	TB2C1T1	TB2C0T1	TB2E1T1	TB2E0T1	TB2FF0C1	TB2FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			TB2FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11"		TB2FF0 inversion trigger 0: Disable 1: Enable				TB2FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11"	
					Invert when the UC value is loaded in to TB2CP1H/L	Invert when the UC value is loaded in to TB2CP0H/L	Invert when the UC matches with TB2RG1H/L	Invert when the UC matches with TB2RG0H/L		
TB2RG0L	TMRB2 register 0 Low	11A8H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB2RG0H	TMRB2 register 0 High	11A9H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB2RG1L	TMRB2 register 1 Low	11AAH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB2RG1H	TMRB2 register 1 High	11ABH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB2CP0L	TMRB2 Capture register Low	11ACH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB2CP0H	TMRB2 Capture register 0 High	11ADH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB2CP1L	TMRB2 Capture register 1 Low	11AEH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB2CP1H	TMRB2 Capture register 1 High	11AFH	–	–	–	–	–	–	–	–
			R							
			Undefined							

16-bit timer (4/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB3RUN	TMRB3 RUN register	11B0H	TB3RDE	–			I2TB3	TB3PRUN		TB3RUN
			R/W	R/W			R/W	R/W		R/W
			0	0			0	0		0
			Double Buffer 0: Disable 1: Enable	Always write “0”			IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TB3MOD	TMRB3 MODE register	11B2H (Prohibit RMW)	TB3CT1	TB3ET1	TB3CP0I	TB3CPM1	TB3CPM0	TB3CLE	TB3CLK1	TB3CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TB3FF1 inversion trigger 0: Disable 1: Enable		Software capture control 0: Software capture 1: Undefined	Capture timing 00: Disable INTA is rising edge 01:TB3IN0 ↑ , TB3IN1 ↑ INTA is rising edge 10: TB3IN0 ↑ , TB3IN0 ↓ INTA is falling edge 11: TA3OUT ↑ , TA3OUT ↓ INTA is riseing edge		Up counter control 0: Disable 1: Enable	TMRB3 source clock 00: TB3IN0 pin input 01: ϕT1 10: ϕT4 11: ϕT16	
			Invert when capture to capture register 3	Invert when match UC3 with TB3RG1H/L						
TB3FFCR	TMRB3 Flip-Flop control register	11B3H (Prohibit RMW)	TB3FF1C1	TB3FF1C0	TB3C1T1	TB3C0T1	TB3E1T1	TB3E0T1	TB3FF0C1	TB3FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			TB3FF1 control 00: Invert 01: Set 10: Clear 11: Don’t care * Always read as “11”		TB3FF0 inversion trigger 0: Disable 1: Enable				TB3FF0 control 00: Invert 01: Set 10: Clear 11: Don’t care * Always read as “11”	
					Invert when the UC value is loaded in to TB3CP1H/L	Invert when the UC value is loaded in to TB3CP0H/L	Invert when the UC matches with TB3RG1H/L	Invert when the UC matches with TB3RG0H/L		
TB3RG0L	TMRB3 register 0 Low	11B8H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB3RG0H	TMRB3 register 0 High	11B9H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB3RG1L	TMRB3 register 1 Low	11BAH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB3RG1H	TMRB3 register 1 High	11BBH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB3CP0L	TMRB3 Capture register 0 Low	11BCH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB3CP0H	TMRB3 Capture register 0 High	11BDH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB3CP1L	TMRB3 Capture register 1 Low	11BEH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB3CP1H	TMRB3 Capture register 1 High	11BFH	–	–	–	–	–	–	–	–
			R							
			Undefined							

16-bit timer (5/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB4RUN	TMRB4 RUN register	11C0H	TB4RDE	–			I2TB4	TB4PRUN		TB4RUN
			R/W	R/W			R/W	R/W		R/W
			0	0			0	0		0
			Double Buffer 0: Disable 1: Enable	Always write “0”			IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TB4MOD	TMRB4 MODE register	11C2H (Prohibit RMW)	TB4CT1	TB4ET1	TB4CP0I	TB4CPM1	TB4CPM0	TB4CLE	TB4CLK1	TB4CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TB4FF1 inversion trigger 0: Disable 1: Enable		Software capture control 0: Software capture 1: Undefined	Capture timing 00: Disable 01: Reserved 10: Reserved 11: TA5OUT ↑, TA5OUT ↓		Up counter control 0: Disable 1: Enable	TMRB4 source clock 00: Reserved 01: φT1 10: φT4 11: φT16	
			Invert when capture to capture register 4	Invert when match UC4 with TB5RG1H/L						
TB4FFCR	TMRB4 Flip-Flop control register	11C3H (Prohibit RMW)	TB4FF1C1	TB4FF1C0	TB4C1T1	TB4C0T1	TB4E1T1	TB4E0T1	TB4FF0C1	TB4FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			TB4FF1 control 00: Invert 01: Set 10: Clear 11: Don’t care * Always read as “11”		TB4FF0 inversion trigger 0: Disable 1: Enable				TB4FF0 control 00: Invert 01: Set 10: Clear 11: Don’t care * Always read as “11”	
					Invert when the UC value is loaded in to TB4CP1H/L	Invert when the UC value is loaded in to TB4CP0H/L	Invert when the UC matches with TB4RG1H/L	Invert when the UC matches with TB4RG0H/L		
TB4RG0L	TMRB4 register 0 Low	11C8H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB4RG0H	TMRB4 register 0 High	11C9H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB4RG1L	TMRB4 register 1 Low	11CAH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB4RG1H	TMRB4 register 1 High	11CBH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB4CP0L	TMRB4 Capture register 0 Low	11CCH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB4CP0H	TMRB4 Capture register 0 High	11CDH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB4CP1L	TMRB4 Capture register 1 Low	11CEH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB4CP1H	TMRB4 Capture register 1 High	11CFH	–	–	–	–	–	–	–	–
			R							
			Undefined							

16-bit timer (6/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB5RUN	TMRB5 RUN register	11D0H	TB5RDE	–			I2TB5	TB5PRUN		TB5RUN
			R/W	R/W			R/W	R/W		R/W
			0	0			0	0		0
			Double Buffer 0: Disable 1: Enable	Always write "0"			IDLE2 0: Stop 1: Operate	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TB5MOD	TMRB5 MODE register	11D2H (Prohibit RMW)	TB5CT1	TB5ET1	TB5CP0I	TB5CPM1	TB5CPM0	TB5CLE	TB5CLK1	TB5CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TB5FF1 inversion trigger 0: Disable 1: Enable		Software capture control 0: Software capture 1: Undefined	Capture timing 00: Disable 01: Reserved 10: Reserved 11: TA5OUT ↑, TA5OUT ↓		Up counter control 0: Disable 1: Enable	TMRB5 source clock 00: Reserved 01: φT1 10: φT4 11: φT16	
			Invert when capture to capture register 5	Invert when match UC5 with TB5RG1H/L						
TB5FFCR	TMRB5 Flip-Flop control register	11D3H (Prohibit RMW)	TB5FF1C1	TB5FF1C0	TB5C1T1	TB5C0T1	TB5E1T1	TB5E0T1	TB5FF0C1	TB5FF0C0
			W*		R/W				W*	
			1	1	0	0	0	0	1	1
			TB5FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11"		TB5FF0 inversion trigger 0: Disable 1: Enable				TB5FF0 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11"	
					Invert when the UC value is loaded in to TB5CP1H/L	Invert when the UC value is loaded in to TB5CP0H/L	Invert when the UC matches with TB5RG1H/L	Invert when the UC matches with TB5RG0H/L		
TB5RG0L	TMRB5 register 0 Low	11D8H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB5RG0H	TMRB5 register 0 High	11D9H (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB5RG1L	TMRB5 register 1 Low	11DAH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB5RG1H	TMRB5 register 1 High	11DBH (Prohibit RMW)	–	–	–	–	–	–	–	–
			W							
			Undefined							
TB5CP0L	TMRB5 Capture register 0 Low	11DCH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB5CP0H	TMRB5 Capture register 0 High	11DDH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB5CP1L	TMRB5 Capture register 1 Low	11DEH	–	–	–	–	–	–	–	–
			R							
			Undefined							
TB5CP1H	TMRB5 Capture register 1 High	11DFH	–	–	–	–	–	–	–	–
			R							
			Undefined							

(9) Pattern Generator

PG0REG	PG0 register	1460H (Prohibit RMW)	PG03	PG02	PG01	PG00	SA03	SA02	SA01	SA00
			W				R/W			
			0	0	0	0	Undefined			
			Pattern generation 0 (PG0) output latch register (PG0 can be read by reading the port (PL) that is assigned to PG				Shift alternate register 0 for the PG mode (4-bit write) register			
PG1REG	PG1 register	1461H (Prohibit RMW)	PG13	PG12	PG11	PG10	SA13	SA12	SA11	SA10
			W				R/W			
			0	0	0	0	Undefined			
			Pattern generation 1 (PG1) output latch register (PG1 can be read by reading the port (PL) that is assigned to PG				Shift alternate register 1 for the PG mode (4-bit write) register			
PG01CR	PG0,1 Control register	1462H	PAT1	CCW1	PG1M	PG1TE	PAT0	CCW0	PG0M	PG0TE
			R/W							
			0	0	0	0	0	0	0	0
			PG1 write mode 0: 8-bit write 1: 4-bit write	PG1 rotation direction 0: Normal rotation 1: Reverse rotation	PG1 mode (Excitation) 0: 1 step excitation or 2 step excitation 1: 1 to 2 step excitation	PG1 trigger input enable 0: Disable 1: Enable	PG0 write mode 0: 8-bit write 1: 4-bit write	PG0 write mode 0: 8-bit write 1: 4-bit write	PG0 mode (Excitation) 0: 1 step excitation or 2 step excitation 1: 1 to 2 step excitation	PG0 trigger input enable 0: Disable 1: Enable
PG01CR2	PG0,1 Control2 register	1464H							PG1T	PG0T
									R/W	
									0	0
									PG1 shift trigger 0:8-bit timer trigger (TMRA23) 1:16-bit timer trigger (TMRB1)	PG0 shift trigger 0:8-bit timer trigger (TMRA01) 1:16-bit timer trigger (TMRB0)

(10) High Speed SIO (1/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
HSC0MD	High Speed Serial Channel 0 mode setting register	C00H		XEN0				CLKSEL02	CLKSEL01	CLKSEL00
				R/W				R/W		
				0				1	0	0
				SYSCK 0:disable 1:enable				Select baud rate 000:Reserved 100:fsys/16 001:fsys/2 101:fsys/32 010:fsys/4 110:fsys/64 011:fsys/8 111:Reserved		
		C01H	LOOPBACK0	MSB1ST0	DOSTAT0		TCPOL0	RCPOL0	TDINV0	RDINV0
			R/W				R/W			
			0	1	1		0	0	0	0
			LOOPBACK teset mode 0:disable 1:enable	Start bit for transmit/receive 0:LSB 1:MSB	HSSO0 pin (no transmit) 0:fixed to "0" 1:fixed to "1"		Synchronous clock edge during transmitting 0: fall 1: rise	Synchronous clock edge during receiving 0: fall 1: rise	Invert data During transmitting 0: disable 1: enable	Invert data During receiving 0: disable 1: enable
HSC0CT	High Speed Serial Channel 0 control register	C02H	-	-	UNIT160			ALGNEN0	RXWEN0	RXUEN0
			R/W					R/W		
			0	1	0			0	0	0
			Always write "0"	Always write "1"	Data length 0: 8bit 1: 16bit			Full duplex allgnment 0:disable 1:enable	Sequential receive 0:disable 1:enable	Receive UNIT 0:disable 1:enable
		C03H	CRC16_7_B0	CRCRX_TX_B0	CRCREST_B0				DMAERFW0	DMAERFR0
			R/W						R/W	
			0	0	0				0	0
			CRC select 0:CRC7 1:CRC16	CRC data 0:Transmit 1:Receive	CRC calculate register 0:Reset 1:Release Reset				Micro DMA 0: Disable 1: Enable	Micro DMA 0: Disable 1: Enable
HSC0ST	High Speed Serial Channel 0 status register	C04H					TEND0	REND0	RFW0	RFR0
							R			
							1	0	1	0
							Transmitting 0: operation 1: no operation	Receive Shift register 0: no data 1: exist data	Transmit buffer 0: untransmitted data exist 1: no untransmitted data	Receive buffer 0: no valid data 1: valid data exist
		C05H								
HSC0CR	High Speed Serial Channel 0 CRC register	C06H	CRCD007	CRCD006	CRCD005	CRCD004	CRCD003	CRCD002	CRCD001	CRCD000
			R							
			0	0	0	0	0	0	0	0
			CRC calculation result load register [7:0]							
		C07H	CRCD015	CRCD014	CRCD013	CRCD012	CRCD011	CRCD010	CRCD009	CRCD008
			R							
			0	0	0	0	0	0	0	0
			CRC calculation result load register [15:8]							

High Speed SIO (2/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
HSC0IS	High Speed Serial Channel 0 interrupt status register	C08H					TENDIS0	RENDIS0	RFWIS0	RFRIS0
							R/W			
							0	0	0	0
							Read 0:no interrupt 1:interrupt Write 0: Don't care 1: clear	Read 0:no interrupt 1:interrupt Write 0: Don't care 1: clear	Read 0:no interrupt 1:interrupt Write 0: Don't care 1: clear	Read 0:no interrupt 1:interrupt Write 0: Don't care 1: clear
		C09H								
HSC0WE	High Speed Serial Channel 0 interrupt status write enable register	C0AH					TENDWE0	RENDWE0	RFWWE0	RFRWE0
							R/W			
							0	0	0	0
							Clear HSC0IS <TENDIS0> 0:disable 1:enable	Clear HSC0IS <RENDIS0> 0:disable 1:enable	Clear HSC0IS <RFWIS0> 0:disable 1:enable	Clear HSC0IS <RFRIS0> 0:disable 1:enable
		C0BH								
HSC0IE	High Speed Serial Channel 0 interupt enable register	C0CH					TENDIE0	RENDIE0	RFWIE0	RFRIE0
							R/W			
							0	0	0	0
							TEND0 interrupt 0:Disable 1:Enable	REND0 interrupt 0:Disable 1:Enable	RFW0 interrupt 0:Disable 1:Enable	RFR0 interrupt 0:Disable 1:Enable
		C0DH								
HSC0IR	High Speed Serial Channel 0 interupt request register	C0EH					TENDIR0	RENDIR0	RFWIR0	RFRIR0
							R			
							0	0	0	0
							TEND0 interrupt 0:none 1:generate	REND0 interrupt 0:none 1:generate	RFW0 interrupt 0:none 1:generate	RFR0 interrupt 0:none 1:generate
		C0FH								

High Speed SIO (3/6)

HSC0TD	High Speed Serial Channel 0 transmission data register	C10H	TXD007	TXD006	TXD005	TXD004	TXD003	TXD002	TXD001	TXD000
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data register [7:0]							
		C11H	TXD015	TXD014	TXD013	TXD012	TXD011	TXD010	TXD009	TXD008
			R							
			0	0	0	0	0	0	0	0
			Transmission data register [15:8]							
HSC0RD	High Speed Serial Channel 0 receiving data register	C12H	RXD007	RXD006	RXD005	RXD004	RXD003	RXD002	RXD001	RXD000
			R							
			0	0	0	0	0	0	0	0
			Receive data register [7:0]							
		C13H	RXD015	RXD014	RXD013	RXD012	RXD011	RXD010	RXD009	RXD008
			R							
			0	0	0	0	0	0	0	0
			Receive data register [15:8]							
HSC0TS	High Speed Serial Channel 0 transmit data shift register	C14H	TSD007	TSD006	TSD005	TSD004	TSD003	TSD002	TSD001	TSD000
			R							
			0	0	0	0	0	0	0	0
			Transmit data shift register [7:0]							
		C15H	TSD015	TSD014	TSD013	TSD012	TSD011	TSD010	TSD009	TSD008
			R							
			0	0	0	0	0	0	0	0
			Transmit data shift register [15:8]							
HSC0RS	High Speed Serial Channel 0 receive data shift register	C16H	RSD007	RSD006	RSD005	RSD004	RSD003	RSD002	RSD001	RSD000
			R							
			0	0	0	0	0	0	0	0
			Receive data shift register [7:0]							
		C17H	RSD015	RSD014	RSD013	RSD012	RSD011	RSD010	RSD009	RSD008
			R							
			0	0	0	0	0	0	0	0
			Receive data shift register [15:8]							

High Speed SIO (4/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
HSC1MD	High Speed Serial Channel 1 mode setting register	C20H		XEN1				CLKSEL12	CLKSEL11	CLKSEL10
				R/W				R/W		
				0				1	0	0
				SYSCK 0:disable 1:enable				Select baud rate 000:Reserved 100:fsys/16 001:fsys/2 101:fsys/32 010:fsys/4 110:fsys/64 011:fsys/8 111:Reserved		
		C21H	LOOPBACK1	MSB1ST1	DOSTAT1		TCPOL1	RCPOL1	TDINV1	RDINV1
			R/W				R/W			
			0	1	1		0	0	0	0
			LOOPBACK test mode 0:disable 1:enable	Start bit for transmit/receive 0:LSB 1:MSB	HSSO1 pin (no transmit) 0:fixed to "0" 1:fixed to "1"		Synchronous clock edge during transmitting 0: fall 1: rise	Synchronous clock edge during receiving 0: fall 1: rise	Invert data During transmitting 0: disable 1: enable	Invert data During receiving 0: disable 1: enable
HSC1CT	High Speed Serial Channel 1 control register	C22H	-	-	UNIT161			ALGNEN1	RXWEN1	RXUEN1
			R/W					R/W		
			0	1	0			0	0	0
			Always write "0"	Always write "1"	Data length 0: 8bit 1: 16bit			Full duplex alignment 0:disable 1:enable	Sequential receive 0:disable 1:enable	Receive UNIT 0:disable 1:enable
		C23H	CRC16_7_B1	CRCRX_TX_B1	CRCREST_B1				DMAERFW1	DMAERFR1
			R/W						R/W	
			0	0	0				0	0
			CRC select 0:CRC7 1:CRC16	CRC data 0:Transmit 1:Receive	CRC calculate register 0:Reset 1:Release Reset				Micro DMA 0: Disable 1: Enable	Micro DMA 0: Disable 1: Enable
HSC1ST	High Speed Serial Channel 1 status register	C24H					TEND1	REND1	RFW1	RFR1
							R			
							1	0	1	0
							Transmitting 0: operation 1: no operation	Receive shift register 0: no data 1: exist data	Transmit buffer 0: untransmitted data exist 1: no untransmitted data	Receive buffer 0: no valid data 1: valid data exist
		C25H								
HSC1CR	High Speed Serial Channel 1 CRC register	C26H	CRCD107	CRCD106	CRCD105	CRCD104	CRCD103	CRCD102	CRCD101	CRCD100
			R							
			0	0	0	0	0	0	0	0
			CRC calculation result load register [7:0]							
		C27H	CRCD115	CRCD114	CRCD113	CRCD112	CRCD111	CRCD110	CRCD109	CRCD108
			R							
			0	0	0	0	0	0	0	0
			CRC calculation result load register [15:8]							

High Speed SIO (5/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
HSC1IS	High Speed Serial Channel 1 interrupt status register	C28H					TENDIS1	RENDIS1	RFWIS1	RFRIS1
							R/W			
							0	0	0	0
							Read 0:no interrupt 1:interrupt Write 0: Don't care 1: clear	Read 0:no interrupt 1:interrupt Write 0: Don't care 1: clear	Read 0:no interrupt 1:interrupt Write 0: Don't care 1: clear	Read 0:no interrupt 1:interrupt Write 0: Don't care 1: clear
		C29H								
HSC1WE	High Speed Serial Channel 1 interrupt status write enable	C2AH					TENDWE1	RENDWE1	RFWWE1	RFRWE1
							R/W			
							0	0	0	0
							Clear HSC1IS <TENDIS1> 0:Disable 1:Enable	Clear HSC1IS <RENDIS1> 0:Disable 1:Enable	Clear HSC1IS <RFWIS1> 0:Disable 1:Enable	Clear HSC1IS <RFRIS1> 0:Disable 1:Enable
		C2BH								
HSC1IE	High Speed Serial Channel 1 interrupt enable register	C2CH					TENDIE1	RENDIE1	RFWIE1	RFRIE1
							R/W			
							0	0	0	0
							TEND1 interupt 0:Disable 1:Enable	REND1 interrupt 0:Disable 1:Enable	RFW1 interupt 0:Disable 1:Enable	RFR1 interrupt 0:Disable 1:Enable
		C2DH								
HSC1IR	High Speed Serial Channel 1 interrupt request register	C2EH					TENDIR1	RENDIR1	RFWIR1	RFRIR1
							R			
							0	0	0	0
							TEND1 interrupt 0:none 1:generate	REND1 interrupt 0:none 1:generate	RFW1 interrupt 0:none 1:generate	RFR1 interrupt 0:none 1:generate
		C2FH								

High Speed SIO (6/6)

HSC1TD	High Speed Serial Channel 1 transmission data register	C30H	TXD107	TXD106	TXD105	TXD104	TXD103	TXD102	TXD101	TXD100
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data register [7:0]							
		C31H	TXD115	TXD114	TXD113	TXD112	TXD111	TXD110	TXD109	TXD108
			R							
			0	0	0	0	0	0	0	0
			Transmission data register [15:8]							
HSC1RD	High Speed Serial Channel 1 Receive data register	C32H	RXD107	RXD106	RXD105	RXD104	RXD103	RXD102	RXD101	RXD100
			R							
			0	0	0	0	0	0	0	0
			Receive data register [7:0]							
		C33H	RXD115	RXD114	RXD113	RXD112	RXD111	RXD110	RXD109	RXD108
			R							
			0	0	0	0	0	0	0	0
			Receive data register [15:8]							
HSC1TS	High Speed Serial Channel 1 transmit data shift register	C34H	TSD107	TSD106	TSD105	TSD104	TSD103	TSD102	TSD101	TSD100
			R							
			0	0	0	0	0	0	0	0
			Transmit data shift register [7:0]							
		C35H	TSD115	TSD114	TSD113	TSD112	TSD111	TSD110	TSD109	TSD108
			R							
			0	0	0	0	0	0	0	0
			Transmit data shift register [15:8]							
HSC1RS	High Speed Serial Channel 1 receive data shift register	C36H	RSD107	RSD106	RSD105	RSD104	RSD103	RSD102	RSD101	RSD100
			R							
			0	0	0	0	0	0	0	0
			Receive data shift register [7:0]							
		C37H	RSD115	RSD114	RSD113	RSD112	RSD111	RSD110	RSD109	RSD108
			R							
			0	0	0	0	0	0	0	0
			Receive data shift register [15:8]							

(11)UART / Serial Channels (1/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer レジスタ	1200H (Prohibit 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	1201H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Clear 0 by reading)			R/W	
			0	0	0	0	0	0	0	0
			Received data bit8	Parity 0:Odd 1:Even	Parity addition 0:Disable 1:Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0:SCLK0 ↑ 1:SCLK0 ↓	I/O interface Input clock selection 0: Baud Rate Generator 1:SCLK0 input
SC0MOD0	Serial Channel 0 Mode 0 register	1202H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transfer data bit8	Handshake function control 0: Disable 1: Enable	Receive control 0: Disable 1: Enable	Wake-up function 0:Disable 1:Enable	Serial transmission mode 00: I/O Interface Mode 01: 7bit UART Mode 10: 8bit UART Mode 11: 9bit UART Mode		Serial transmission clock (UART) 00: TA0TRG (TMRA01) 01: Baud Rate Generator 10: Internal clock fsys 11: External clock (SCLK0 input)	
BR0CR	Serial Channel 0 Baud Rate control register	1203H	-	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write "0"	(16-K)/16 division 0:Disable 1:Enable	Baud Rate Generator Input clock selection 00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the devided frequency 0 to F			
BR0ADD	Serial Channel 0 K setup register	1204H					BR0K3	BR0K2	BR0K1	BR0K0
							R/W			
							0	0	0	0
							Sets frequency divisor "K" (Divided by N+(16-K)/16) 1 to F			
SC0MOD1	Serial Channel 0 Mode 1 register	1205H	I2S0	FDPX0						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface mode 0: Half 1: Full						
SIR0CR	Serial Channel 0 IrDA control register	1207H	PLSEL	RXSEL	TXEN	RXEN	SIR0WD3	SIR0WD2	SIR0WD1	SIR0WD0
			R/W							
			0	0	0	0	0	0	0	0
			Selection transmission pulse width 0: 3/16 1: 1/16	Receiving data logic 0: "H" pulse 1: "L" pulse	Transmission data 0: disable 1: enable	Receiving operation 0: disable 1: enable	Select receiving effective pulse width Set effective pulse width for equal or more than 2x × (Value+1)+100ns Can be set: 1 to 14 Cannot be set: 0, 15			

UART / Serial Channels (2/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC1BUF	Serial Channel 1 Buffer register	1208H (Prohibit 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	1209H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Clear 0 by reading)			R/W	
			0	0	0	0	0	0	0	0
			Receive data bit8	Parity 0:Odd 1:Even	Parity addition 0:Disable 1:Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0:SCLK1 ↑ 1:SCLK1 ↓	I/O interface Input clock selection 0: Baud Rate Generator 1:SCLK1 input
SC1MOD0	Serial Channel 1 Mode 0 register	120AH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transfer data bit8	Handshake function control 0: Disable 1: Enable	Receive control 0: Disable 1: Enable	Wake-up function 0:Disable 1:Enable	Serial transmission mode 00: I/O Interface Mode 01: 7bit UART Mode 10: 8bit UART Mode 11: 9bit UART Mode		Serial transmission clock (UART) 00: TA0TRG (TMRA01) 01: Baud Rate Generator 10: Internal clock fsys 11: External clock (SCLK1 input)	
BR1CR	Serial Channel 1 Baud Rate control register	120BH	-	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write "0"	(16-K)/16 division 0:Disable 1:Enable	Baud Rate Generator Input clock selection 00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the devided frequency 0 to F			
BR1ADD	Serial Channel 1 K setup register	120CH					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Sets frequency divisor "K" (Divided by N+(16-K)/16) 1 to F			
SC1MOD1	Serial Channel 1 Mode 1 register	120DH	I2S1	FDPX1						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface mode 0: Half 1: Full						

UART / Serial Channels (3/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC2BUF	Serial Channel 2 Buffer register	1210H (Prohibit RMW)	RB7 / TB7	RB6 / TB6	RB5 / TB5	RB4 / TB4	RB3 / TB3	RB2 / TB2	RB1 / TB1	RB0 / TB0
			R (Receiving) / W (Transmission)							
			Undefined							
SC2CR	Serial Channel 2 control register	1211H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Clear 0 by reading)			R/W	
			0	0	0	0	0	0	0	0
			Received data bit8	Parity 0:Odd 1:Even	Parity addition 0:Disable 1:Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0:SCLK2 ↑ 1:SCLK2 ↓	I/O interface Input clock selection 0: Baud Rate Generator 1:SCLK2 input
SC2MOD0	Serial Channel 2 Mode 0 register	1212H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transfer data bit8	Handshake function control 0: Disable 1: Enable	Receive control 0: Disable 1: Enable	Wake-up function 0:Disable 1:Enable	Serial transmission mode 00: I/O Interface Mode 01: 7bit UART Mode 10: 8bit UART Mode 11: 9bit UART Mode		Serial transmission clock (UART) 00: TA0TRG (TMRA01) 01: Baud Rate Generator 10: Internal clock fsys 11: External clock (SCLK2 input)	
BR2CR	Serial Channel 2 Baud Rate control register	1213H	-	BR2ADDE	BR2CK1	BR2CK0	BR2S3	BR2S2	BR2S1	BR2S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write "0"	(16-K)/16 division 0:Disable 1:Enable	Baud Rate Generator Input clock selection 00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the devided frequency 0 to F			
BR2ADD	Serial Channel 2 K setup register	1214H					BR2K3	BR2K2	BR2K1	BR2K0
							R/W			
							0	0	0	0
							Sets frequency divisor "K" (Divided by N+(16-K)/16) 1 to F			
SC2MOD1	Serial Channel 2 Mode 1 register	1215H	I2S2	FDPX2						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface mode 0: Half 1: Full						

UART / Serial Channels (4/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC3BUF	Serial Channel 3 Buffer register	1218H (Prohibit RMW)	RB7 / TB7	RB6 / TB6	RB5 / TB5	RB4 / TB4	RB3 / TB3	RB2 / TB2	RB1 / TB1	RB0 / TB0
			R (Receiving) / W (Transmission)							
			Undefined							
SC3CR	Serial Channel 3 control register	1219H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Clear 0 by reading)			R/W	
			0	0	0	0	0	0	0	0
			Received data bit8	Parity 0:Odd 1:Even	Parity addition 0:Disable 1:Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0:SCLK3 ↑ 1:SCLK3 ↓	I/O interface input clock selection 0: Baud Rate Generator 1:SCLK3 input
SC3MOD0	Serial Channel 3 Mode 0 register	121AH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transfer data bit8	Handshake function control 0: Disable 1: Enable	Receive control 0: Disable 1: Enable	Wake-up function 0:Disable 1:Enable	Serial transmission mode 00: I/O Interface Mode 01: 7bit UART Mode 10: 8bit UART Mode 11: 9bit UART Mode		Serial transmission clock (UART) 00: TA0TRG (TMRA01) 01: Baud Rate Generator 10: Internal clock fsys 11: External clock (SCLK3 input)	
BR3CR	Serial Channel 3 Baud Rate control register	121BH	-	BR3ADDE	BR3CK1	BR3CK0	BR3S3	BR3S2	BR3S1	BR3S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write "0"	(16-K)/16 division 0:Disable 1:Enable	Baud Rate Generator Input clock selection 00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the devided frequency 0 to F			
BR3ADD	Serial Channel 3 K setup Register	121CH					BR3K3	BR3K2	BR3K1	BR3K0
							R/W			
							0	0	0	0
							Sets frequency divisor "K" (Divided by N+(16-K)/16) 1 to F			
SC3MOD1	Serial Channel 3 Mode 1 register	121DH	I2S3	FDPX3						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface mode 0: Half 1: Full						

(12) I^2CBUS/Serial Channel(1/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI0CR1	SBI0 control register 1	1240H (no RMW) I^2C mode	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
			W			R/W		W		R/W
			0	0	0	0		0	0	0
			Number of transfer bits 000:8 001:1 010:2 011:3 100:4 101:5 110:6 111:7			Acknowledge mode 0:Disable 1:Enable		Setting of the divide value "n" 000:5 001:6 010:7 011:8 100:9 101:10 110:11 111:Reserved		
		1240H (no RMW) SIO mode	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
			W					W		W
			0	0	0	0		0	0	0
			Transfer 0:Stop 1:Start	Transfer 0:Continue 1:Abort	Transfer mode 00:8bit transmit 10:8bit ransmit/receive 11:8bit receive			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	1241H (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	1242H (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	1243H (no RMW) I^2C 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	INTSBI0 interrupt 0:Request 1:Cancel	Operation mode selection 00:Port mode 10: SIO mode 01: I^2C mode 11: Reserved		Software reset generate write "10" and "01", then an internal reset signal is generated.	
		1243H (no RMW) SIO mode					SBIM1	SBIM0	-	-
							W		W	W
							0	0	0	0
							Operation mode selection 00:Port mode 10:SIO mode 01:I^2C mode 11:Reserved		Always write "0"	Always write "0"
SBI0SR	SBI0 Status register	1243H (no RMW) I^2C 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	INTSBI0 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"
		1243H (no RMW) SIO mode					SIOF	SEF		
							R			
							0	0		
							Transfer status 0:Stopped 1:In progress	Shift status 0:Stopped 1:In progress		

I²CBUS/Serial Channel(2/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI0BR0	SBI0 Baud rate register 0	1244H	-	I2SBI0						
				R/W						
			-	0						
			Always write “0”	IDLE2 0:Abort 1:Operate						
SBI0BR1	SBI0 Baud rate register 1	1245H	P4EN	-						
			R/W	W						
			0	0						
			Clock control 0:Stop 1:Operate	Always write “0”						

I²CBUS/Serial Channel(3/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI1CR1	SBI1 control register 1	1248H (no RMW) I²C mode	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
			W			R/W		W		R/W
			0	0	0	0		0	0	0
			Number of transfer bits 000:8 001:1 010:2 011:3 100:4 101:5 110:6 111:7			Acknowledge mode 0:Disable 1:Enable		Setting of the divide value "n" 000:5 001:6 010:7 011:8 100:9 101:10 110:11 111:Reserved		
		1248H (no RMW) SIO mode	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
			W					W		W
			0	0	0	0		0	0	0
			Transfer 0:Stop 1:Start	Transfer 0:Continue 1:Abort	Transfer mode 00:8bit transmit 10:8bit ransmit/receive 11:8bit receive			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	SBI0 Buffer register	1249H (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	124AH (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	124BH (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	INTSBI1 interrupt 0:Request 1:Cancel	Operation mode selection 00:Port mode 10: SIO mode 01: I²Cmode 11: Reserved		Software reset generate write "10" and "01", then an internal reset signal is generated.	
		124BH (no RMW) SIO mode					SBIM1	SBIM0	-	-
							W		W	W
							0	0	0	0
							Operation mode selection 00:Port mode 10:SIO mode 01:I²C mode 11:Reserved		Always write "0"	Always write "0"
SBI1SR	SBI1 Status register	124BH (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	INTSBI1 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"
		124BH (no RMW) SIO mode					SIOF	SEF		
							R			
							0	0		
							Transfer status 0:Stopped 1:In progress	Shift status 0:Stopped 1:In progress		

I²CBUS/Serial Channel(4/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI1BR0	SBI1 Baud rate register 0	124CH	-	I2SBI1						
				R/W						
			-	0						
			Always write “0”	IDLE2 0:Abort 1:Operate						
SBI1BR1	SBI1 Baud rate register 1	124DH	P4EN	-						
			R/W	W						
			0	0						
			Clock control 0:Stop 1:Operate	Always write “0”						

(13) AD converter (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD0	AD Mode control register 0	12B8H	EOCF	ADBF	-	-	ITM0	REPEAT	SCAN	ADS
			R		R/W		R/W			
			0	0	0	0	0	0	0	0
			AD conversion end flag 0: Conversion in progress 1: Conversion complete	AD conversion busy flag 0:Conversion stopped 1:Conversion in progress	Always write “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.
ADMOD1	AD Mode control register 1	12B9H	VREFON	I2AD	-	-	ADCH3	ADCH2	ADCH1	ADCH0
			R/W	R/W	R/W		R/W			
			0	0	0	0	0	0	0	0
			VREF applicatio n control 0: OFF 1: ON	DLE2): Stop L: Operate	Always write “0”		Analog input channel selection fixed / 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			
ADMOD2	AD Mode control register 2	12BAH	-	-	-	-	-	-	-	ADTRGE
										R/W
			0	0	0	0	0	0	0	0
			Always write “0”	Always write “0”	Always write “0”	Always write “0”	Always write “0”	Always write “0”	Always write “0”	AD conversion trigger start control 0: disable 1: enable

AD converter (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADREG0L	AD Result register 0 Low	12A0H	ADR01	ADR00						ADR0RF
			R							R
			Undefined							0
ADREG0H	AD Result register 0 High	12A1H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
ADREG1L	AD Result register 1 Low	12A2H	ADR11	ADR10						ADR1RF
			R							R
			Undefined							0
ADREG1H	AD Result register 1 High	12A3H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
ADREG2L	AD Result register 2 Low	12A4H	ADR21	ADR20						ADR2RF
			R							R
			Undefined							0
ADREG2H	AD Result register 2 High	12A5H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
ADREG3L	AD Result register 3 Low	12A6H	ADR31	ADR30						ADR3RF
			R							R
			Undefined							0
ADREG3H	AD Result register 3 High	12A7H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							
ADREG4L	AD Result register 4 Low	12A8H	ADR41	ADR40						ADR4RF
			R							R
			Undefined							0
ADREG4H	AD Result register 4 High	12A9H	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
			R							
			Undefined							
ADREG5L	AD Result register 5 Low	12AAH	ADR51	ADR50						ADR5RF
			R							R
			Undefined							0
ADREG5H	AD Result register 5 High	12ABH	ADR59	ADR58	ADR57	ADR56	ADR55	ADR54	ADR53	ADR52
			R							
			Undefined							
ADREG6L	AD Result register 6 Low	12ACH	ADR61	ADR60						ADR6RF
			R							R
			Undefined							0
ADREG6H	AD Result register 6 High	12ADH	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
			R							
			Undefined							
ADREG7L	AD Result register 7 Low	12AEH	ADR71	ADR70						ADR7RF
			R							R
			Undefined							0
ADREG7H	AD Result register 7 High	12AFH	ADR79	ADR78	ADR77	ADR76	ADR75	ADR74	ADR73	ADR72
			R							
			Undefined							

AD converter (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADREG8L	AD Result register 8 Low	12B0H	ADR81	ADR80						ADR8RF
			R							R
			Undefined							0
ADREG8H	AD Result register 8 High	12B1H	ADR89	ADR88	ADR87	ADR86	ADR85	ADR84	ADR83	ADR82
			R							
			Undefined							
ADREG9L	AD Result register 9 Low	12B2H	ADR91	ADR90						ADR9RF
			R							R
			Undefined							0
ADREG9H	AD Result register 9 High	12B3H	ADR99	ADR98	ADR97	ADR96	ADR95	ADR94	ADR93	ADR92
			R							
			Undefined							
ADREGAL	AD Result register A Low	12B4H	ADRA1	ADRA0						ADRARF
			R							R
			Undefined							0
ADREGAH	AD Result register A High	12B5H	ADRA9	ADRA8	ADRA7	ADRA6	ADRA5	ADRA4	ADRA3	ADRA2
			R							
			Undefined							
ADREGBL	AD Result register B Low	12B6H	ADRB1	ADRB0						ADRBRF
			R							R
			Undefined							0
ADREGBH	AD Result register B High	12B7H	ADRB9	ADRB8	ADRB7	ADRB6	ADRB5	ADRB4	ADRB3	ADRB2
			R							
			Undefined							

(14) DA converter

Symbol	Name	Address	7	6	5	4	3	2	1	0
DAC0REG	DA 0 register	12E0H	DAC07	DAC06	DAC05	DAC04	DAC03	DAC02	DAC01	DAC00
			R/W							
			0	0	0	0	0	0	0	0
DAC0CNT1	DA 0 control register 1	12E1H	–	–	–	–				VALID
			R/W							W
			0	0	0	0				0
			Always write "0"	Always write "0"	Always write "0"	Always write "0"				0:Don't care 1:Output CODE valid
DAC0CNT0	DA 0 control register 0	12E3H							REFON0	OP0
									R/W	
									0	0
									0:Ref off 1:Ref on	0:Output High-Z 1:Output
DAC1REG	DA 1 register	12E4H	DAC17	DAC16	DAC15	DAC14	DAC13	DAC12	DAC11	DAC10
			R/W							
			0	0	0	0	0	0	0	0
DAC1CNT1	DA 1 control register 1	12E5H	–	–	–	–				VALID
			R/W							W
			0	0						0
			Always write "0"	Always write "0"	Always write "0"	Always write "0"				0:Don't care 1:Output CODE valid
DAC1CNT0	DA 1 control register 0	12E7H							REFON1	OP1
									R/W	
									0	0
									0:Ref off 1:Ref on	0:Output High-Z 1:Output

(15) Watch dog timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	WDT Mode register	1300H	WDTE	WDTP1	WDTP0	–	–	I2WDT	RESCR	–
			R/W	R/W				R/W	R/W	
			1	0	0	0	0	0	0	0
			WDT control 1: enable	WDT detection time 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Always write "0"		IDLE2 0: Stop 1: Operate	1:Internally connects WDT out to the reset pin	Always write "0"
WDCR	WDT control register	1301H	–							
			W							
			–							
			B1H: WDT disable code 4E: WDT clear code							

(16) Key-on wake up

Symbol	Name	Address	7	6	5	4	3	2	1	0
KIEN	KeyInput Enable setup register	009EH (Prohibit RMW)	KI7EN	KI6EN	KI5EN	KI4EN	KI3EN	KI2EN	KI1EN	KI0EN
			W							
			0	0	0	0	0	0	0	0
			KI7 input 0:Disable 1:Enable	KI6 input 0:Disable 1:Enable	KI5 input 0:Disable 1:Enable	KI4 input 0:Disable 1:Enable	KI3 input 0:Disable 1:Enable	KI2 input 0:Disable 1:Enable	KI1 input 0:Disable 1:Enable	KI0 input 0:Disable 1:Enable
KICR	Key Input control register	009FH (Prohibit RMW)	KI7EDGE	KI6EDGE	KI5EDGE	KI4EDGE	KI3EDGE	KI2EDGE	KI1EDGE	KI0EDGE
			W							
			0	0	0	0	0	0	0	0
			KI7 edge 0:Rising 1:Falling	KI6 edge 0:Rising 1:Falling	KI5 edge 0:Rising 1:Falling	KI4 edge 0:Rising 1:Falling	KI3 edge 0:Rising 1:Falling	KI2 edge 0:Rising 1:Falling	KI1 edge 0:Rising 1:Falling	KI0 edge 0:Rising 1:Falling

6. Points of Note and Restrictions

(1) Notation

a. The notation for built-in/ I/O registers is as follows register symbol <Bit symbol> (e.g., TA01RUN <TA0RUN> denotes bit TA0RUN of register TA01RUN).

b. Read-modify-write instructions

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, (TA01RUN) ... Set bit 3 of TA01RUN.

Example 2: INC 1, (100H) ... Increment the data at 100H.

- Examples of read-modify-write instructions on the TLCS-900

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)

c. fc, fs, f_{FPH}, f_{SYS} and one state

The clock frequency input on X1 and 2 is called f_{OSCH}. The clock selected by PLLCR0<FCSEL> is called fc.

The clock selected by SYSCR1<SYSCK> is called f_{FPH}. The clock frequency give by f_{FPH} divided by 2 is called f_{SYS}.

One cycle of f_{SYS} is referred to as one state.

(2) Points of note

a. AM0 and AM1 pins

This pin is connected to the V_{CC} or the V_{SS} pin. Do not alter the level when the pin is active.

b. Reserved address areas

Since the 16 byte area of FFFFF0H ～ FFFFFFH is reserved as internal area, use of it is impossible. Moreover, when using an emulator, since it is used for control of an emulator, 64K bytes with arbitrary 16M byte area of use cannot be performed.

c. Warm-up counter

The warm-up counter operates when STOP mode is released, even if the system is using an external oscillator. As a result a time equivalent to the warm-up time elapses between input of the release request and output of the system clock.

d. Watchdog timer

The watchdog timer starts operation immediately after a reset is released. When the watchdog timer is not to be used, disable it.

e. AD converter

The string resistor between the VREFH and VREFL pins can be cut by a program so as to reduce power consumption. When STOP mode is used, disable the resistor using the program before the HALT instruction is executed.

f. CPU (Micro DMA)

Only the “LDC cr, r” and “LDC r, cr” instructions can be used to access the control registers in the CPU (e.g., the transfer source address register (DMASn)).

g. Undefined SFR

The value of an undefined bit in an SFR is undefined when read.

h. POP SR instruction

Please execute the POP SR instruction during DI condition.

i. Interrupt

When you use interruption, be sure to set "1" as the bit 7 of a SIMC register.

7. Package Dimensions

Package Name: P-LQFP144-1616-0.40C

Unit:mm