CMOS 8-Bit Microcontroller

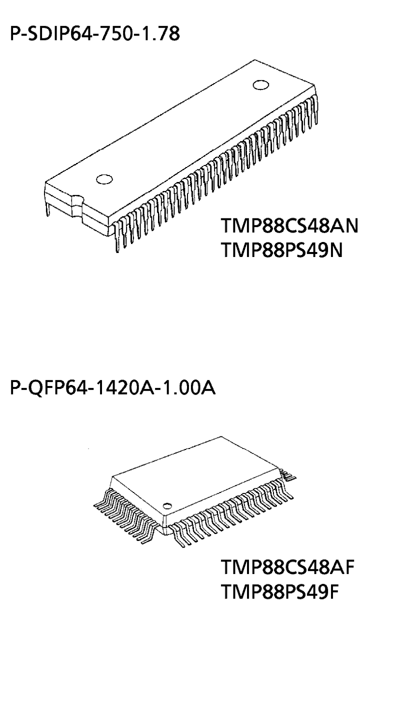
TMP88CS48AN, TMP88CS48AF

TMP88CS48A is high-speed and high-function 8-bit single-chip microcomputers whose built-in features include large-capacity RAM, multi-function timer/counter, and 10-bit AD converter, serial interface (UART/I^{2}C bus). They are equipped with 3 phase brushless DC sensorless/sensor motor control, and AC motor inverter control.

Part No.	ROM	RAM	Package	OTP MCU
TMP88CS48AN	64 Kbytes	2 Kbytes	P-SDIP64-750-1.78	TMP88PS49N
TMP88CS48AF			P-QFP64-1420A-1.00A	TMP88PS49F

Features

- ◆8-bit single-chip microcomputer TLCS-870/X series microcomputer
- ◆Interrupt sources: 25 (6 external, 19 Internal)
- ◆I/O ports: 56 pins
  - ● Large-current output: 8 pins (typ. 20 mA), LED direct drive
- ◆16-bit timer/counter: 2 channels
  - ● Timer, event counter, programmable pulse generator (PPG) output, pulse width measurement, external trigger timer, window mode
- ◆8-bit Timer/Counter: 3 channels
  - ● Timer, event counter, capture (pulse width/duty measurement), pulse width modulation (PWM) output, programmable divider output (PDO) mode
- ◆Time base timer (interrupt frequency: 1 to 16384 Hz)
- ◆Watchdog timer
- ◆Divider output function (frequency: 1 to 8 kHz)
- ◆Programmable motor driver (PMD): 1 channel
  - ● Rotor position: minimum resolution of 250 ns for detecting rotor position
  - ● Motor control timer, timer capture function
  - ● Overload protection function
    DC overload protection function
    AC overload protection function (Can halt counter in 3-phase PWM output circuit)
  - ● Protection circuit for malfunction (urgent halt)
  - ● Automatic direction change, automatic position detection start
- ◆High-speed PWM output: 2 channel
  - ● Cycle: 32 kHz, 64 kHz, 128 kHz (at 8 MHz operation)
  - ● Resolution: 8-bit, 7-bit, 6-bit mode selectable



980910EBP1

- ● For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance / Handling Precautions.
- ● TOSHIBA is continually working to improve the quality and the reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.
- ● The products described in this document are subject to the foreign exchange and foreign trade laws.
- ● The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- ● The information contained herein is subject to change without notice.

Purchase of TOSHIBA I^2 C components conveys a license under the Philips I^2 C Patent Rights to use these components in an I^2 C system, provided that the system conforms to the I^2 C Standard Specification as defined by Philips.

◆Serial interface

- 8-bit SIO/I^{2}C bus
- Universal asynchronous receiver transmitter (UART)

◆10-bit successive approximation type AD converter

- Analog input: 16 channels
- Conversion time: 46 μs (at 16 MHz operation)

◆Low power dissipation operation (2 modes)

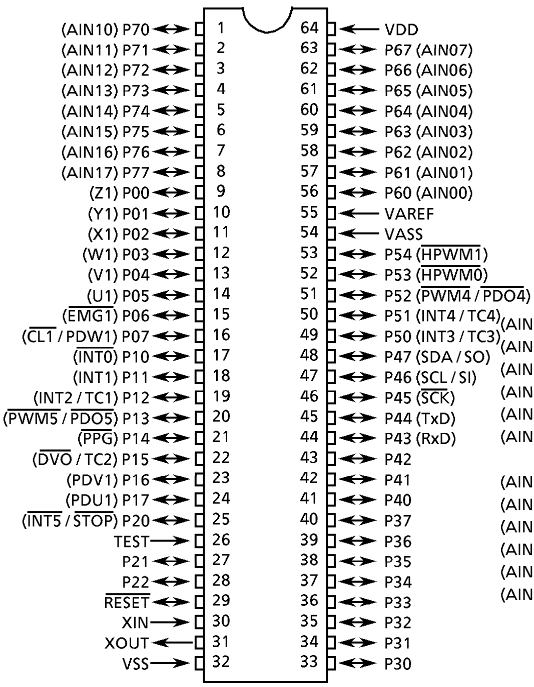
- STOP mode:Stops oscillation (battery or capacitor backup). Port output hold or high impedance selectable
- IDLE mode:Stops CPU but continues operation of peripheral hardware. Released by interrupt (restarts CPU)

◆Operating voltage: 4.5 to 5.5 V at 16 MHz operation

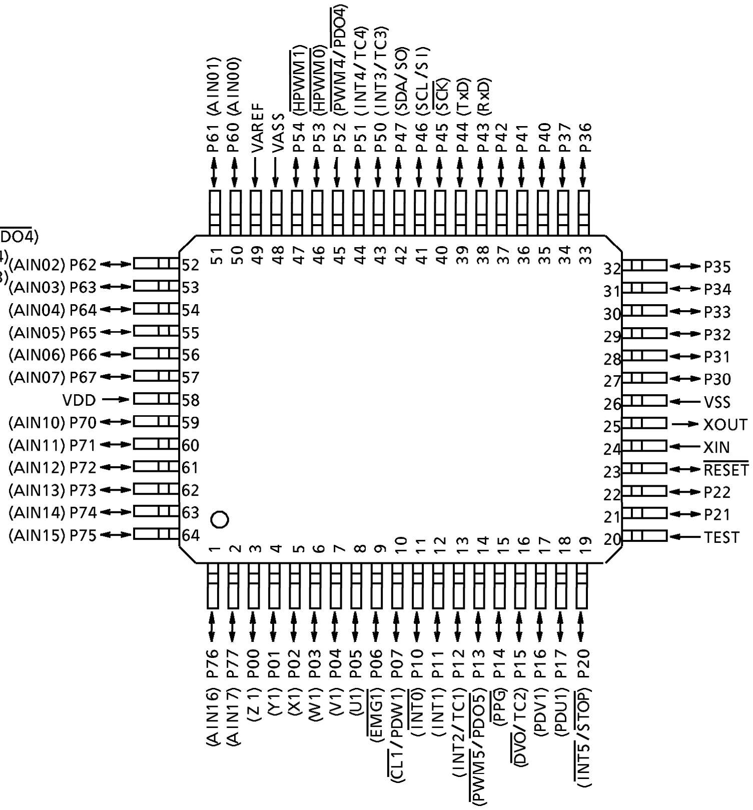
◆Emulation pod: BM88CM49N0A

Pin Assignments

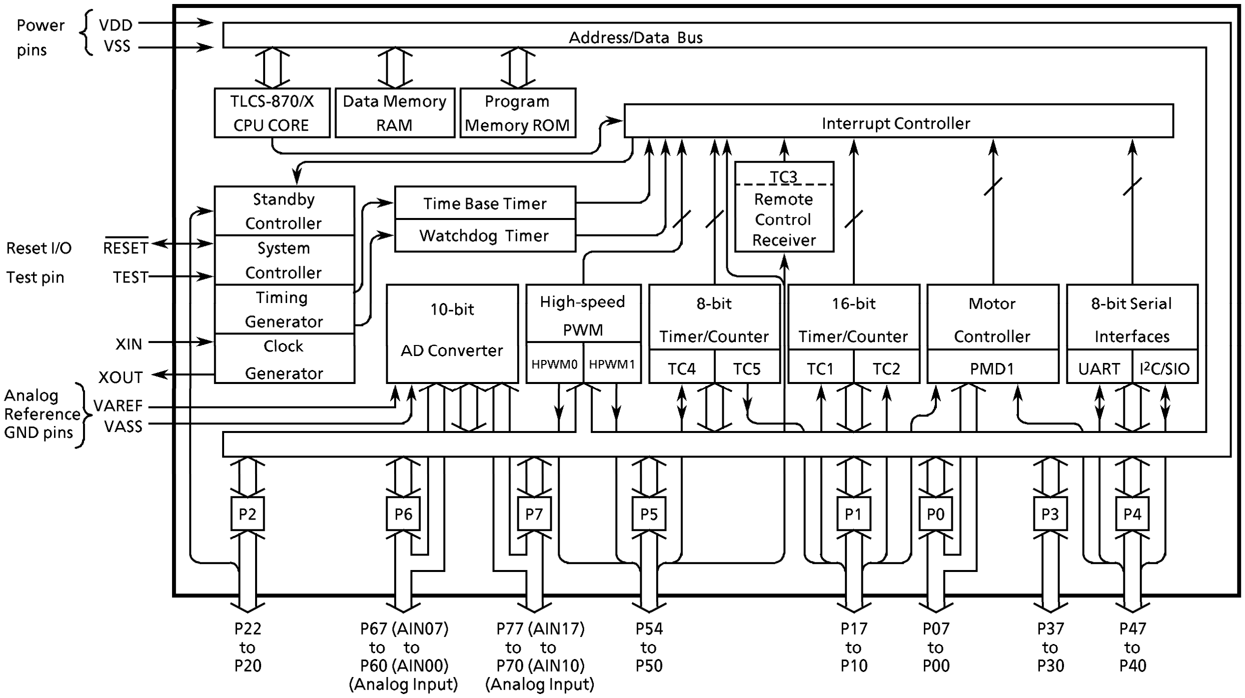
P-SDIP64-750-1.78



P-QFP64-1420-1.00A



Block Diagram



Pin Function

Pin Name	I/O	Function	
P07 ($\overline{\text{CL1}}$/PDW1)	I/O (Input)	8-bit programmable I/O port (tri state) Input or output specifiable in units of bits. When using pins for motor control circuit, set accordingly using P0CR, then MDCR to 1.	Overload protection input 1/motor control circuit W1-phase position detection input
P06 ($\overline{\text{EMG1}}$)			Motor control circuit malfunction detection input 1
P05 (U1)	I/O (Output)		Motor control circuit U1-/V1-/W1-phase output
P04 (V1)			
P03 (W1)			
P02 (X1)	I/O (Output)		Motor control circuit X1-/ Y1-/Z1-phase output
P01 (Y1)			
P00 (Z1)			
P17 (PDU1)	I/O (Input)	8-bit programmable I/O port (tri state) Input or output specifiable units of bits. When using pins for motor control circuit, timer/counter input, or external interrupt input, set them to input mode. When using pins for PPG output, divider output, or PWM output/PDO output, set them to output mode.	Motor control circuit U1-phase position detection input
P16 (PDV1)			Motor control circuit V1-phase position detection input
P15 ($\overline{\text{DVO}}$/TC2)	II/O (Output/Input)		Divider output or Timer/Counter 2 input
P14 ($\overline{\text{PPG}}$)	I/O (Output)		Programmable pulse generator output
P13 ($\overline{\text{PWM5}}$/$\overline{\text{PDO5}}$)			8-BIT PWM output 5 or 8-BIT programmable divider output 5
P12 (INT2/TC1)	I/O (Input)		External interrupt input 2 or Timer/Counter 1 input
P11 (INT1)			External interrupt input 1
P10 ($\overline{\text{INT0}}$)			External interrupt input 0
P22	I/O	3-bit I/O port When using pins for input port, external interrupt input, or STOP mode release input, set output latches to 1.	—
P21			—
P20 ($\overline{\text{INT5}}$/$\overline{\text{STOP}}$)	I/O (Input)		External interrupt input 5 or STOP mode release signal input
P37	I/O (Output)	8-bit I/O port (large-current output) When using pins for motor control circuit input, set output latches to 1, then MDCR2 to 1.	—
P36			
P35			
P34			
P33			
P32			
P31			
P30			
P47 (SDA/SO)	I/O (I/O/Output)	8-bit I/O port When using pins for motor control circuit input, UART/I²C/SIO, set output latches to 1.	I²C/SIO I/O
P46 (SCL/SI)	I/O (I/O/Input)		
P45 ($\overline{\text{SCK}}$)	I/O (I/O)		
P44 (TxD)	I/O (Input)		UART data input
P43 (RxD)	I/O (Output)		UART data output
P42	I/O		—
P41			
P40			
P54 ($\overline{\text{HPWM1}}$)	I/O (Output)	5-bit input/output port with latch. When using pins for input port, HPWM output, PWM output/PDO output, external interrupt input, or timer/counter input, set output latches to 1.	high-speed PWM output
P53 ($\overline{\text{HPWM0}}$)			
P52 ($\overline{\text{PWM4}}$/$\overline{\text{PDO4}}$)			8-BIT PWM output 4 or, 8-bit programmable divider output 4
P51 (INT4/TC4)	I/O (Input)		External interrupt 4 input or timer / counter 4 input
P50 (INT3/TC3)			External interrupt 3 input or timer / counter 3 input

Pin Name	Input/Output	Function	
P67 (AIN07) to P60 (AIN00)	I/O (Input)	8-bit programmable I/O port (tri state) Input or output specifiable in units of bits. When using pins for analog input, set to input mode using P6CR and ADCCR.	AD converter analog input
P77 (AIN17) to P70 (AIN10)	I/O (Input)	8-bit programmable I/O port (tri state) Input or output specifiable in units of bits. When using pins for analog input, set to input mode using P7CR and ADCCR.	AD converter analog input
XIN, XOUT	Input, Output	High-frequency oscillator connecting pins. For external clock input, input to XIN and leave XOUT open.	
$\overline{\text{RESET}}$	I/O	Reset signal input, watchdog timer output, address trap reset output, system clock reset output	
TEST	Input	Shipment test pin, fix to "L" level.	
VDD, VSS	Power Supply	+ 5 V, 0 V (GND)	
VAREF, VASS		Analog reference voltage for AD conversion. Reference GND.	

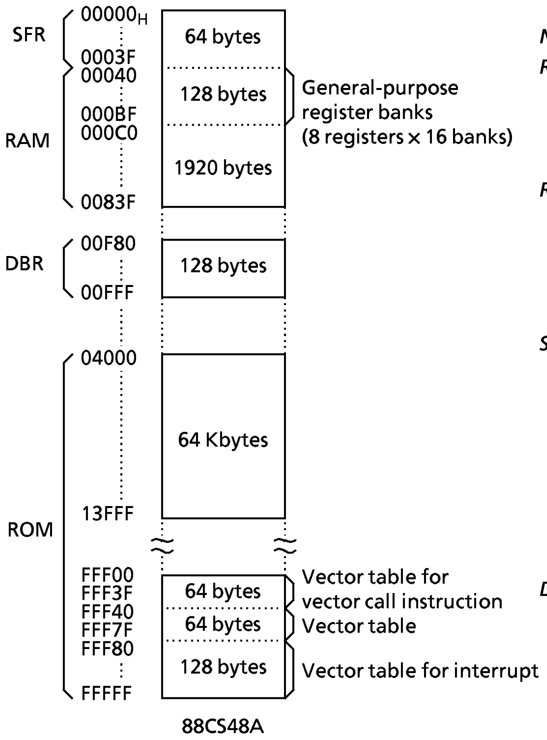
Operation

1. CPU Core Functions

The CPU core consists of the CPU, system clock control circuit, and interrupt control circuit.
This chapter describes the CPU core, program memory, data memory and the reset circuit.

1.1 Memory Address Map

The TMP88CS48A memory consists of four blocks: ROM, RAM, special function registers (SFR) and Data buffer registers (DBR). They are all mapped to a 1 Mbyte address space. Figure 1-1 shows the TMP88CS48A memory address map. There are 16 general-purpose registers mapped to the RAM address space.



Note:
ROM: Read Only Memory includes
Program memory
Vector tables

RAM: Random Access Memory includes
Data memory
Stack
General-purpose register banks

SFR: Special Function Register includes
I/O ports
Peripheral control registers
Peripheral status registers
System control registers
Interrupt control registers
Program Status Word

DBR: Data Buffer Registers includes
Motor control register
UART control registers
UART status registers

Figure 1-1. Memory Address Maps

1.2 Program Memory (ROM)

TMP88CS48A contains a 64 Kbyte program memory (mask ROM) at addresses from 04000 to 13EFF$_H$ and a 256-byte program memory (mask ROM) at addresses from FFF00 to FFFFF$_H$.

1.3 Data Memory (RAM)

TMP88CS48A contains a 2 Kbyte RAM at addresses from 00040 to 0083F$_H$. The first 128 bytes in RAM (00040 to 000BF$_H$) are also used as a general-purpose register bank.
Since the data in data memory become undefined at power on, initialize the RAM using the initialize routine.

Example: Clear RAM (zero-clear the whole RAM except bank 0).

```
            LD      HL, 00048H        ;  Sets start address.
            LD      A, H              ;  Sets initialization data (00H).
            LD      BC, 07F7H         ;  Sets number of bytes ( – 1).
SRAMCLR:    LD      (HL + ), A
            DEC     BC
            JRS     F, SRAMCLR
```

Note: Since the general-purpose registers are allocated to RAM, do not clear RAM at addresses in the current bank. Therefore, in the above example, RAM except bank 0 is cleared.

1.4 System Clock Control Circuit

The system clock control circuit consists of a clock generator, timing generator, and standby control circuit.

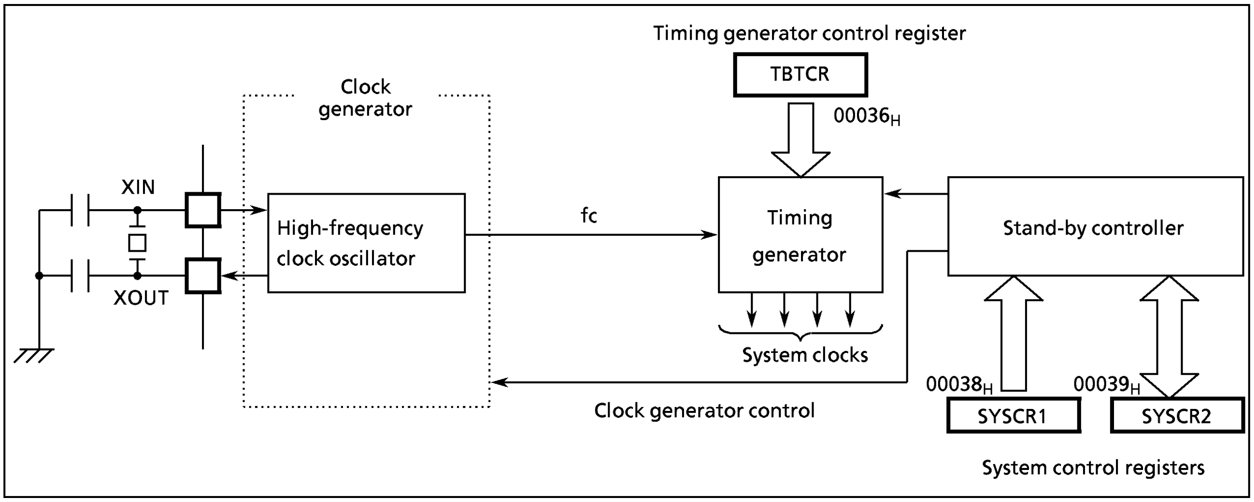


Figure 1-2. System Clock Controller

1.4.1 Clock Generator

The clock generator is an oscillator circuit which generates the basic clock pulse used as the system clock supplied to the CPU core and peripheral hardware.

A high-frequency (frequency: fc) clock can be easily obtained by connecting the oscillator to the XIN and XOUT pins. A clock can also be input externally. In this case, input the clock from the XIN pin and leave the XOUT pin open.

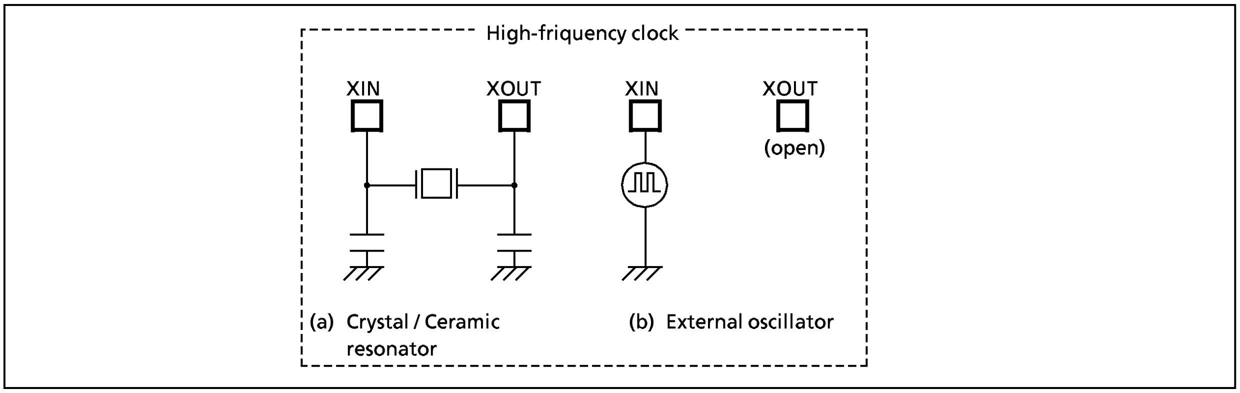


Figure 1-3. Example of Resonator Connection

Adjustment of oscillator frequency

Note: Although the basic clock pulse cannot be directly monitored externally by hardware, it is possible to do so by software. That is, output pulses at a certain frequency to the port (for example, $\overline{\text{DVO}}$), with interrupts and the watchdog timer disabled, then monitor pulses so that the frequency can be adjusted. For a system which requires the oscillator frequency to be adjusted, write a program for the required frequency adjustment.

1.4.2 Timing Generator

The timing generator is a circuit used to generate, based on the basic clock pulse, system clocks supplied to the CPU core and peripheral hardware. The timing generator functions to generate the following:

(1) main system clock
(2) divider output ($\overline{\text{DVO}}$) pulse
(3) source clock for time base timer
(4) source clock for watchdog timer
(5) internal source clock for timer/counter
(6) warm-up clock at STOP mode release

(1) Configuration of timing generator

The timing generator consists of a 2-step prescaler, 21-step divider, and machine cycle counter. At reset, when STOP mode is entered or released, the prescaler and divider are zero-cleared.

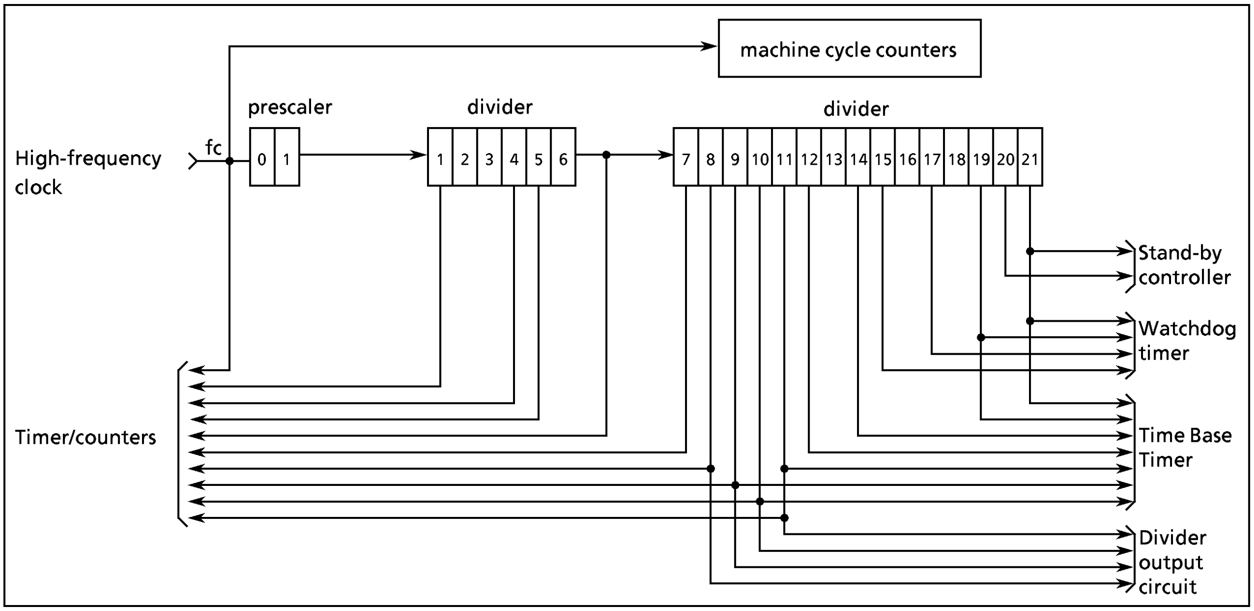


Figure 1-4. Configuration of Timing Generator

TBTCR (00036_H)	7	6	5	4	3	2	1	0	
	(DVOEN)	(DVOCK)		DV7CK	(TBTEN)	(TBTCK)			(Initial value: 0**0 0***)

DV7CK	Selection of input clock to the 7th stage of the divider.	0: 2^8/fc 1: Reserved	R/W

Note 1: Do not set DV7CK to 1.
*Note 2: fc ; High-frequency clock [Hz] * ; Don't care*

Figure 1-5. Timing Generator Control Register

(2) Machine cycle

Execution of instructions and operation of internal hardware are performed in sync with the system clock.

The minimum unit for instruction execution is called the machine cycle. TLCS-870/X series instructions are classified into 15 types: from 1-cycle instructions to 15-cycle instructions.

A machine cycle consists of four states (S0 to S3). Each state consists of one main system clock.

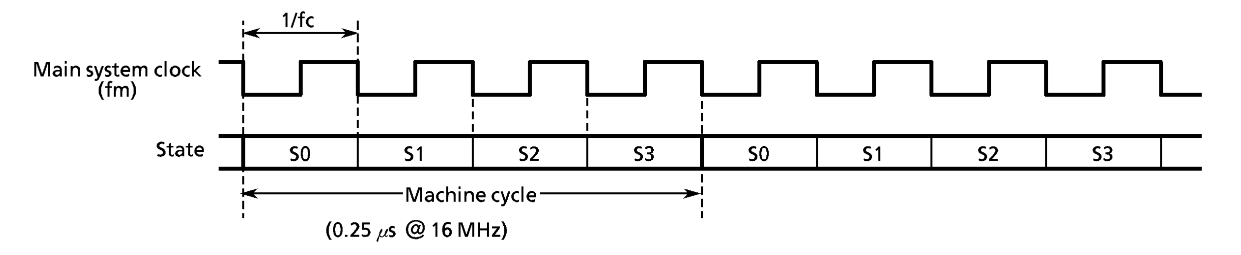


Figure 1-6. Machine Cycle

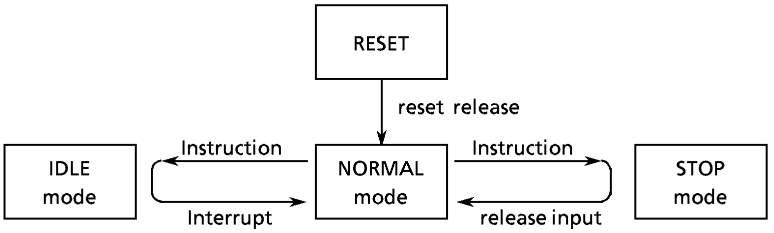
1.4.3 Standby Control Circuit

The standby control circuit operates or stops the oscillator circuit for the high-frequency clock. Control the operating mode using the system control register (SYSCR1, SYSCR2). Figure 1-7 is operating mode transition; Figure 1-8 is the control register.

1) NORMAL mode
Operates the CPU core and peripheral hardware using the high-frequency clock.

2) IDLE mode
Stops the CPU and watchdog timer but operates peripheral hardware using the high-frequency clock. To start IDLE mode, use system control register 2. Interrupts from the peripheral hardware or external interrupts release IDLE mode and return to NORMAL mode. When IMF (interrupt master enable flag) is set to 1 (interrupt enabled), normal operation returns after interrupt processing. When IMF is set to 0 (interrupt disabled), the system restarts execution of the instruction following the instruction which started IDLE mode.

3) STOP mode
Stops all operation including the oscillator circuit but holds the internal states immediately before the stop at low power dissipation.
To start STOP mode, use system control register 1. Input to the $\overline{\text{STOP}}$ pin (level or edge selectable) releases STOP mode. After warm-up time elapses, the system restarts execution of the instruction following the instruction which started STOP mode.



Operating mode	Oscillator circuit	CPU core	Peripheral circuit	Machine cycle
	High-frequency			
RESET	turning on oscillation	reset	reset	4/fc [s]
NORMAL		operate	operate	
IDLE		halt		
STOP	turning off oscillation		halt	—

Figure 1-7. Operation Mode Transition Diagram

System Control Register 1

SYSCR1
(00038$_H$)

7	6	5	4	3	2	1	0
STOP	RELM	RETM	"1"	WUT			

(Initial value: 0001 00**)

STOP	STOP mode start	0: Runs CPU core and peripheral hardware. 1: Stops operation of CPU core and peripheral hardware (stats STOP mode)	R/W
RELM	Release method for STOP mode	0: Releases STOP mode at rising edge of input to $\overline{\text{STOP}}$ pin. 1: Releases STOP mode when input to $\overline{\text{STOP}}$ pin is at "H" level.	
RETM	Operating mode after STOP mode	0: Returns to NORMAL mode. 1: reserved	
WUT	Warming-up time at releasing STOP mode [s]	00: $3 \times 2^{16}/fc$ 01: $2^{16}/fc$ 10: $3 \times 2^{14}/fc$ 11: $2^{14}/fc$	

Note 1: Always set RETM to "0" when transiting from NORMAL mode to STOP mode. Always set RETM to "1" when transiting from SLOW mode to STOP mode.
Note 2: When STOP mode is released with $\overline{\text{RESET}}$ pin input, a return is made to NORMAL mode regardless of the RETM contents.
*Note 3: fc ; High-frequency [Hz] * ; Don't care*
Note 4: Bits 1 and 0 in SYSCR1 are read in as undefined data when a read instruction is executed.
Note 5: Always set bit 4 in SYSCR1 to "1" when STOP mode is started.

System Control Register 2

SYSCR2
(00039$_H$)

7	6	5	4	3	2	1	0
XEN			IDLE				

(Initial value: 1**0 ****)

XEN	High-frequency oscillator control	0 : Turn off oscillation. 1 : Turn on oscillation.	R/W
IDLE	IDLE modes start	0 : Operates CPU and watchdog timer. 1 : Stops operation of CPU and watchdog timer. (Starts IDLE mode.)	

Note 1: Always write "1" in bit 7 of SYSCR 2 and "0" in bits 6 and 5.
Note 2: Writing "0" in bit 7 of SYSCR2 generates a reset (sets output from $\overline{\text{RESET}}$ pin to "L" level).
*Note 3: * ; Don't care*
Note 4: Bits 3 to 0 in SYSCR2 are always read in as "1".

Figure 1-8. System Control Registers

1.4.4 Operating Mode Control

(1) STOP mode

Stop mode is controlled by system control register 1 (SYSCR1) and input to the $\overline{\text{STOP}}$ pin. The $\overline{\text{STOP}}$ pin is also used as P20 or $\overline{\text{INT5}}$ (external interrupt input 5). Setting STOP (bit 7 in SYSCR1) to 1 activates STOP mode. During STOP mode, the following states are held.

① Stops oscillation and stops all internal operation.
② The states of data memory, registers, program status word, and port output latches just before entering STOP mode are held.
③ Zero-clears the timing generator prescaler and divider.
④ The program counter indicates the address for the second instruction from the instruction which stared STOP mode.
For example, [SET (SYSCR1). 7]

STOP mode has two further modes, level and edge. Select one using RELM in system control register 1 (bit 6 in SYSCR1).

a. Level-sensitive release mode (RELM = 1)

Releases STOP mode using "H" level input to the $\overline{\text{STOP}}$ pin. Used for capacitor backup at main power off or during long-time battery backup.

When an instruction is executed which uses a "H" level input to the $\overline{\text{STOP}}$ pin to activate STOP mode, processing does not enter STOP mode, but immediately moves to the release sequence (warm-up). To enter STOP mode in level-sensitive release mode, use the program to check that input to the $\overline{\text{STOP}}$ pin is "L" level. Checking can be done in either of the following two ways:

① Check P20
② Use INT5. (Interrupt is generated at the falling edge of input to the $\overline{\text{INT5}}$ pin.)

Example 1: Starts STOP mode from NORMAL mode checking port 20.

```
            LD      (SYSCR1), 01010000B  ;  Sets to level-sensitive release mode.
SSTOPH:     TEST    (P2). 0              ;  Waits until input to STOP pin becomes
                                            "L" level.
            JRS     F, SSTOPH
            SET     (SYSCR1). 7          ;  Starts STOP mode.
```

Example 2: Starts STOP mode from NORMAL mode using INT5.

```
PINT5:      TEST    (P2). 0              ;  Does not enter STOP mode if input to P20
            JRS     F, SINT5                "H" level due to noise rejection.
            LD      (SYSCR1), 01010000B  ;  Sets to level-sensitive release mode.
            SET     (SYSCR1). 7          ;  Starts stop mode.
SINT5:      RETI
```

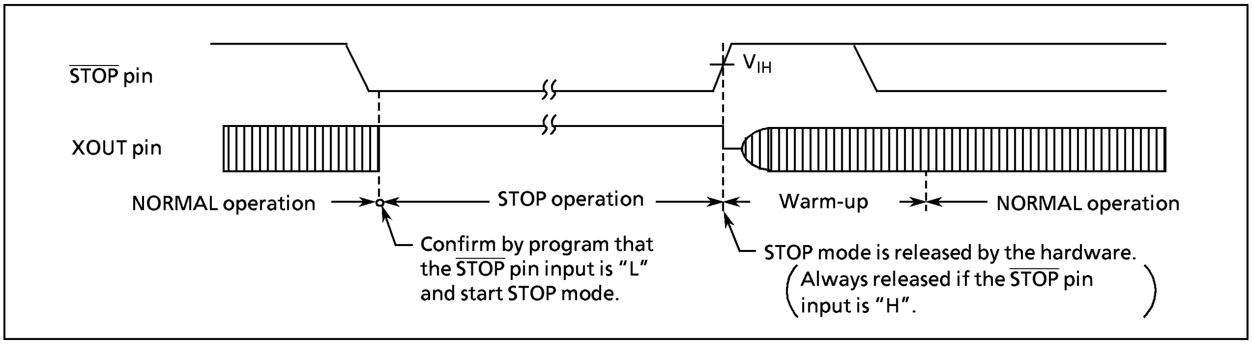

$\overline{\text{STOP}}$ pin
XOUT pin
V_{IH}
NORMAL operation
STOP operation
Warm-up
NORMAL operation
Confirm by program that the $\overline{\text{STOP}}$ pin input is "L" and start STOP mode.
STOP mode is released by the hardware.
(Always released if the $\overline{\text{STOP}}$ pin input is "H".)

Figure 1-9. Level-sensitive Release Mode

Note 1: Even if the $\overline{\text{STOP}}$ pin input is low after warming up start, the STOP mode is not restarted.
Note 2: In this case of changing to the level-sensitive release mode from the edge-sensitive release mode, the release mode is not switched until a rising edge of the $\overline{\text{STOP}}$ pin input is detected.

b. Edge-sensitive release mode (RELM = 0)

Releases STOP mode at the rising edge of input to the $\overline{\text{STOP}}$ pin. Used to repeat a relatively short program at a certain cycle. Input this cycle signal (eg, a clock from the low power dissipation oscillator) to the $\overline{\text{STOP}}$ pin. In edge-sensitive release mode, STOP mode is entered even if input to the $\overline{\text{STOP}}$ pin is "H" level.

Example: Enter STOP mode from NORMAL mode.

```
LD      (SYSCR1), 10010000B  ;  Starts STOP mode with edge-sensitive
                                 release mode set.
```

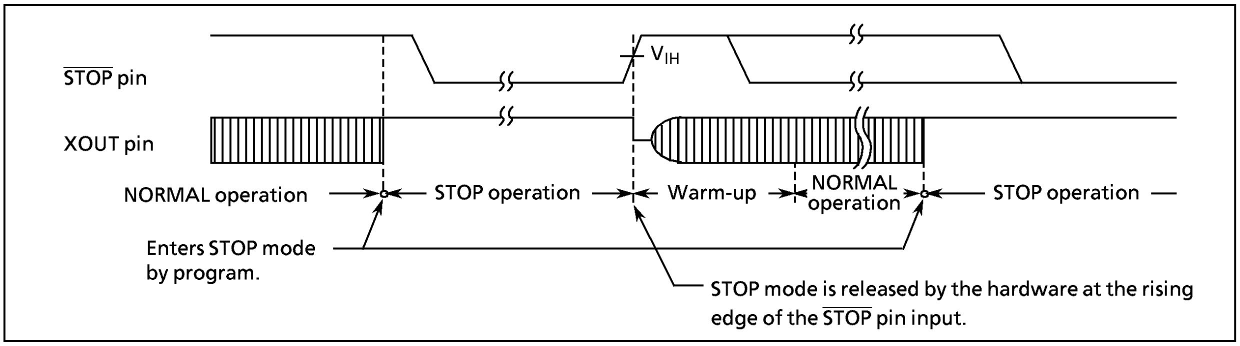

STOP pin
XOUT pin
V_{IH}
NORMAL operation
STOP operation
Warm-up
NORMAL operation
STOP operation
Enters STOP mode by program.
STOP mode is released by the hardware at the rising edge of the STOP pin input.

Figure 1-10. Edge-sensitive Release Mode

STOP mode is released in the flowing sequence:

(1) Starts oscillation.
(2) To wait until oscillation stabilizes, performs warm-up. During warm-up, internal operation remain stopped. Select one of the four warm-up times using WUT (bits 3 and 2 in SYSCR1) depending on the characteristics of the oscillator.
(3) After warm-up time elapses, the instruction following the instruction which activated STOP mode is executed. At this time, the timing generator prescaler and divider are zero-cleared.

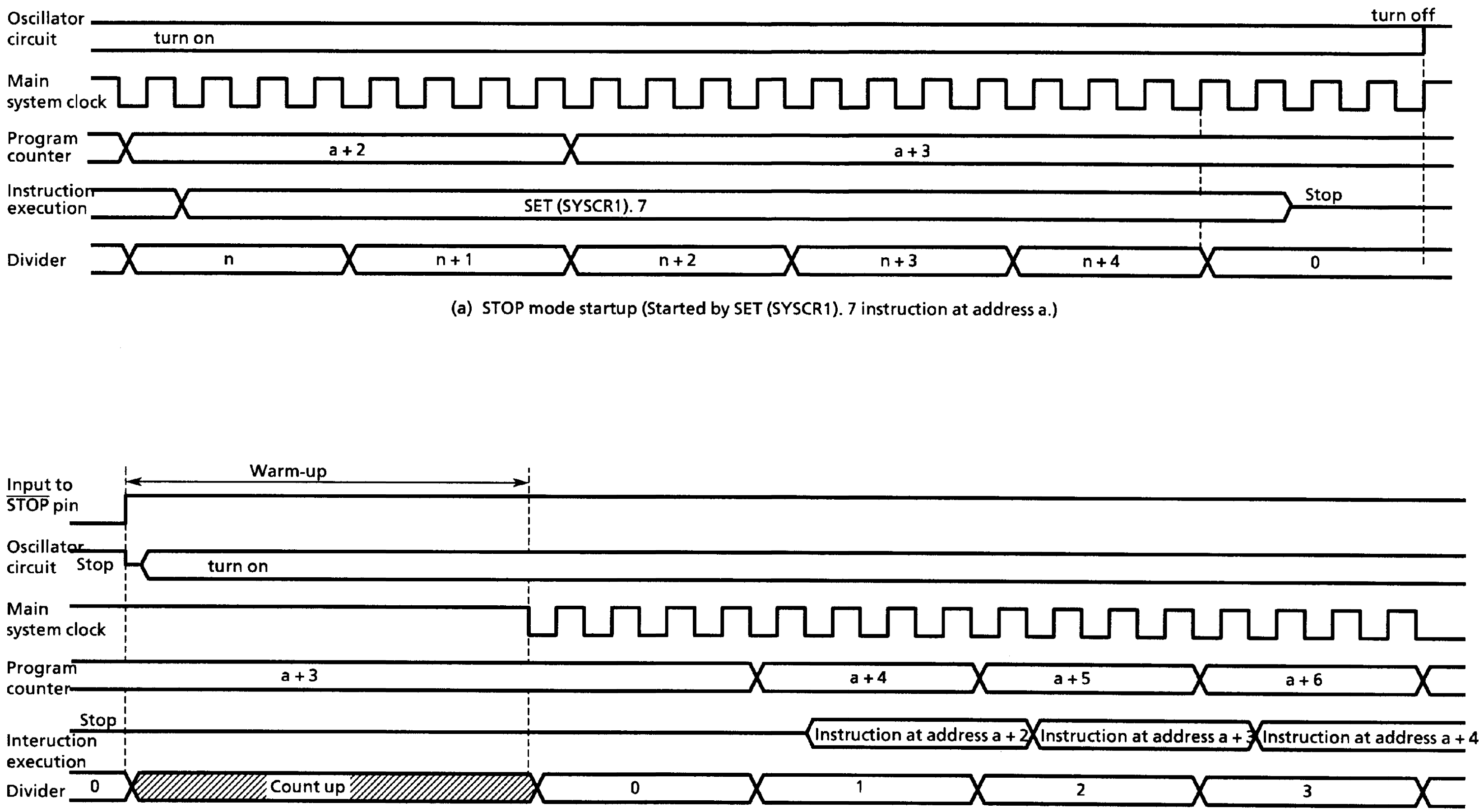
Table 1-1. Warming-up Time Example (at fc = 16 MHz)

WUT	Warming-up Time [ms]
00	$3 \times 2^{16}/fc$ (12.288 m)
01	$2^{16}/fc$ (4.096 m)
10	$3 \times 2^{14}/fc$ (3.072 m)
11	$2^{14}/fc$ (1.024 m)

Note: The warming-up time is obtained by dividing the basic clock by the divider: therefore, the warming-up time may include a certain amount of error if there is any fluctuation of the oscillation frequency when STOP mode is released. Thus, the warming-up time must be considered an approximate value.

For immediate ordinary reset, STOP mode can also be released by setting the RESET pin to "L" level.

Note: STOP mode should not be released when voltage is low. Before releasing STOP mode, increase the supply voltage to the operating voltage. At this time, the RESET pin is at "H" level, which increases along with the supply voltage. If a time constant circuit is externally connected, the increase in the voltage of the input to the RESET pin starts later than that of the supply voltage. If the input voltage drops below the non-inverted "H" level input voltage of the RESET pin (hysteresis input), a reset may be generated.



(a) STOP mode startup (Started by SET (SYSCR1). 7 instruction at address a.)

(b) STOP mode release

Figure 1-11. STOP Mode Start / Release

(2) **IDLE Mode**

IDLE mode is controlled by system control register 2 (SYSCR2) and maskable interrupts. During IDLE mode, the following states are held:

① CPU and watchdog timer operation stop, but operation of peripheral hardware continues.
② The states of data memory, registers, program status word, port output latch immediately before entering IDLE mode are held.
③ The program counter holds the address fo the second instruction after the instruction which started IDLE mode.

```
Example :   Start IDLE mode.
                 SET         (SYSCR2), 4
```

IDLE mode is released by normal release mode or interrupt release mode. Select one using the interrupt master enable flag (IMF). After IDLE mode is released, the mode returns to NORMAL.

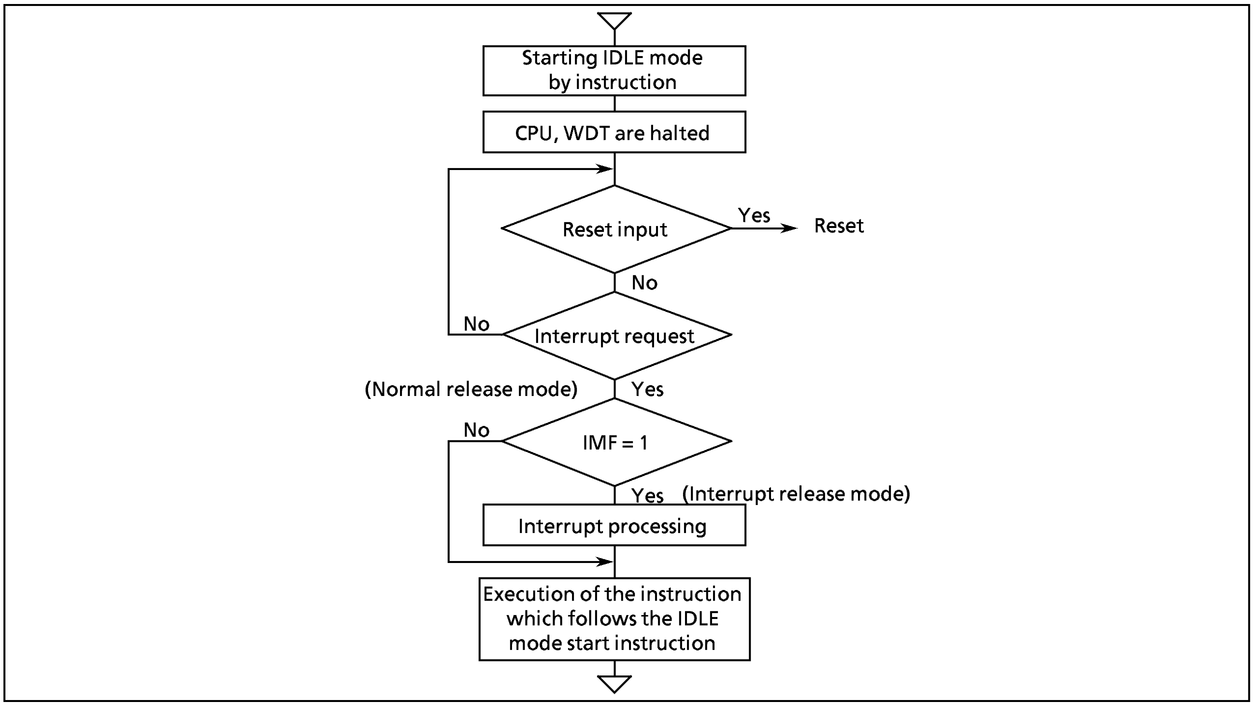


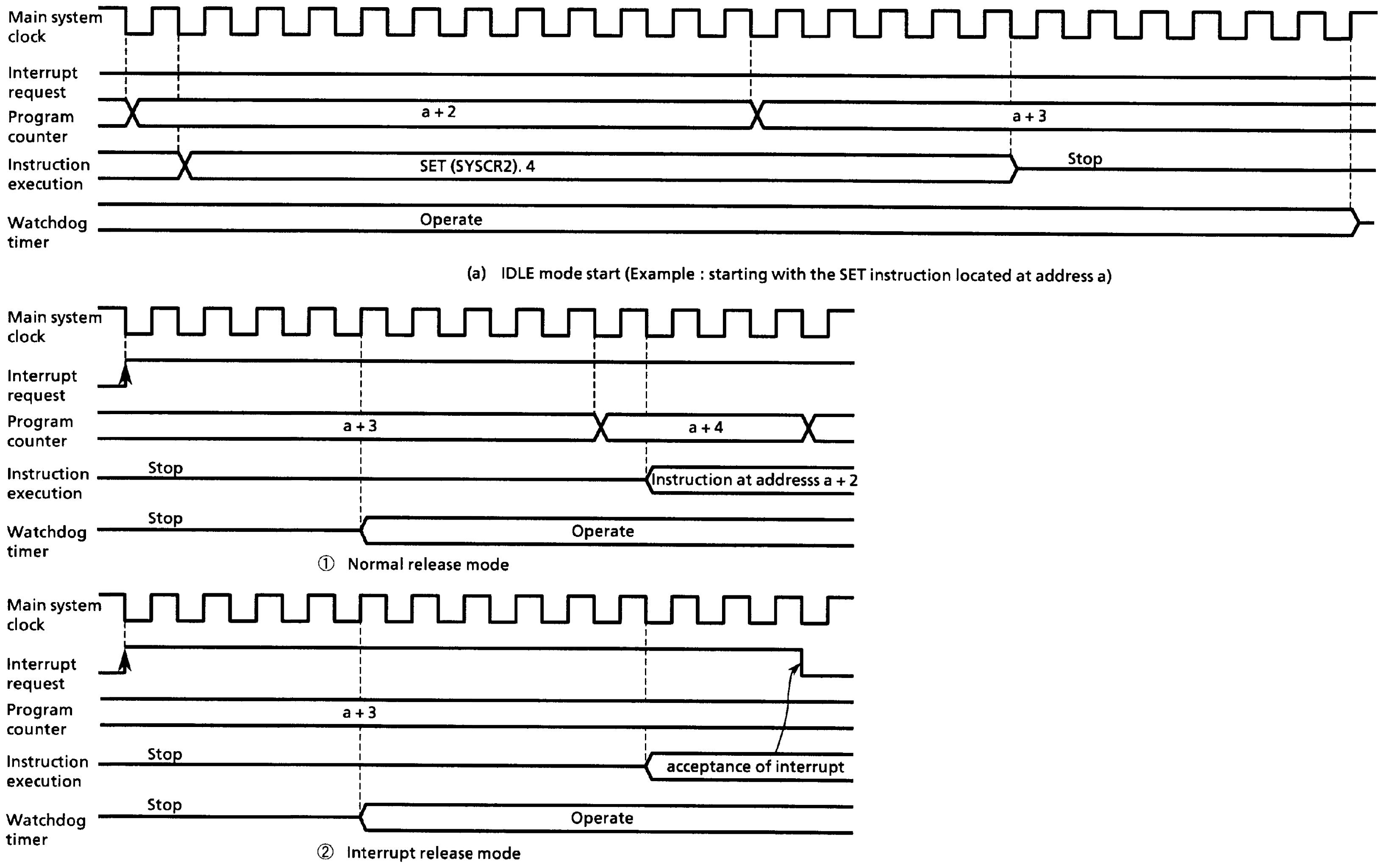
Figure 1-12. IDLE Mode

a. Normal release mode (when IMF = 0)
IDLE mode is released either by an interrupt source enabled by the interrupt enable flag (EF) or by an external interrupt 0 ($\overline{\text{INT0}}$) request. Instruction execution restarts from the instruction following the instruction which started IDLE mode. Usually, the interrupt latch (IL) for the interrupt source used for IDLE mode release must be zero-cleared using the load instruction.

b. Interrupt release mode (when IMF = 1)
IDLE mode is released either by an interrupt source enabled by the interrupt enable flag (EF) or by an external interrupt 0 ($\overline{\text{INT0}}$) request, and interrupt processing starts. After interrupt processing, the instruction following the instruction which started IDLE mode is executed.

For immediate ordinary reset, IDLE mode can also be released by setting the $\overline{\text{RESET}}$ pin to “L” level.

Note: When a watchdog timer interrupt is generated immediately before IDLE mode is started, the watchdog timer interrupt will be processed but IDLE mode will not be started.



(a) IDLE mode start (Example : starting with the SET instruction located at address a)

① Normal release mode

② Interrupt release mode

(b) IDLE mode release

Figure 1-13. IDLE Mode Startup / Release

1.5 Interrupt Control Circuit

TMP88CS48A supports 25 interrupt sources enabling multiple interrupts with priorities. Two internal interrupts are pseudo non-maskable, the rest are all maskable interrupts.

Interrupt sources are provided with interrupt latches (IL), which hold interrupt requests, and independent vectors. Interrupt latches are set to 1 by the generation of interrupt requests which request the CPU to accept the interrupts. Interrupts are enabled or disabled by program using the interrupt master enable flag (IMF) and interrupt enable flag (EF). If more than one interrupt is generated simultaneously, interrupts are accepted in order of the priority specified by hardware.

Figure 1-14 shows the interrupt control circuit.

Table 1-2. Interrupt Sources

Interrupt source			Enable condition	Interrupt latch	Vector table address	Priority
Internal	INTSW	(Software interrupt)	Pseudo non-maskable	—	FFFF8$_H$	1
Internal	INTWDT	(Watchdog timer interrupt)		IL$_2$	FFFF4$_H$	2
External	INT0	(External interrupt)	IMF = 1, INT0EN = 1	IL$_3$	FFFF0$_H$	3
Internal	INTEMG1	(CH1 malfunction detect interrupt)	IMF · EF$_4$ = 1	IL$_4$	FFFEC$_H$	4
Internal	INTT01	(CH1 mode timer overflow/capture range overwrite)	IMF · EF$_6$ = 1	IL$_6$	FFFE4$_H$	5
Internal	INTCLM1	(CH1 current limit interrupt)	IMF · EF$_8$ = 1	IL$_8$	FFFDC$_H$	6
Internal	INTPDC1	(CH1 position detect interrupt)	IMF · EF$_{10}$ = 1	IL$_{10}$	FFFD4$_H$	7
Internal	INTPWM1	(CH1 waveform generator interrupt)	IMF · EF$_{12}$ = 1	IL$_{12}$	FFFCC$_H$	8
Internal	INTTC3	(8-bit TC3 interrupt)	IMF · EF$_{14}$ = 1	IL$_{14}$	FFFC4$_H$	9
External	INT5	(External interrupt)	IMF · EF$_{15}$ = 1	IL$_{15}$	FFFC0$_H$	10
Internal	INTTC1	(16-bit TC1 interrupt)	IMF · EF$_{16}$ = 1	IL$_{16}$	FFFBC$_H$	11
Internal	INTTC2	(16-bit TC2 interrupt)	IMF · EF$_{17}$ = 1	IL$_{17}$	FFFB8$_H$	12
Internal	INTTMR11	(CH1 timer 1 interrupt)	IMF · EF$_{18}$ = 1	IL$_{18}$	FFFB4$_H$	13
Internal	INTTMR21	(CH1 timer 2 interrupt)	IMF · EF$_{20}$ = 1	IL$_{20}$	FFFAC$_H$	14
Internal	INTTMR31	(CH1 timer 3 interrupt)	IMF · EF$_{22}$ = 1	IL$_{22}$	FFFA4$_H$	15
Internal	INTTMR41	(CH1 timer 4 interrupt)	IMF · EF$_{24}$ = 1	IL$_{24}$	FFF9C$_H$	16
External	INT1	(External interrupt)	IMF · EF$_{26}$ = 1	IL$_{26}$	FFF94$_H$	17
External	INT2	(External interrupt)	IMF · EF$_{27}$ = 1	IL$_{27}$	FFF90$_H$	18
External	INT3	(External interrupt)	IMF · EF$_{28}$ = 1	IL$_{28}$	FFF8C$_H$	19
External	INT4	(External interrupt)	IMF · EF$_{29}$ = 1	IL$_{29}$	FFF88$_H$	20
Internal	INTRX	(UART receive interrupt)	IMF · EF$_{30}$ = 1	IL$_{30}$	FFF84$_H$	21
Internal	INTTX	(UART transmit interrupt)	IMF · EF$_{31}$ = 1	IL$_{31}$	FFF80$_H$	22
Internal	INTSBI	(Serial bus interface interrupt)	IMF · EF$_{32}$ = 1	IL$_{32}$	FFF3C$_H$	23
Internal	INTTBT	(Timer base timer interrupt)	IMF · EF$_{33}$ = 1	IL$_{33}$	FFF38$_H$	24
Internal	INTTC5	(8-bit TC5 interrupt)	IMF · EF$_{34}$ = 1	IL$_{34}$	FFF34$_H$	25

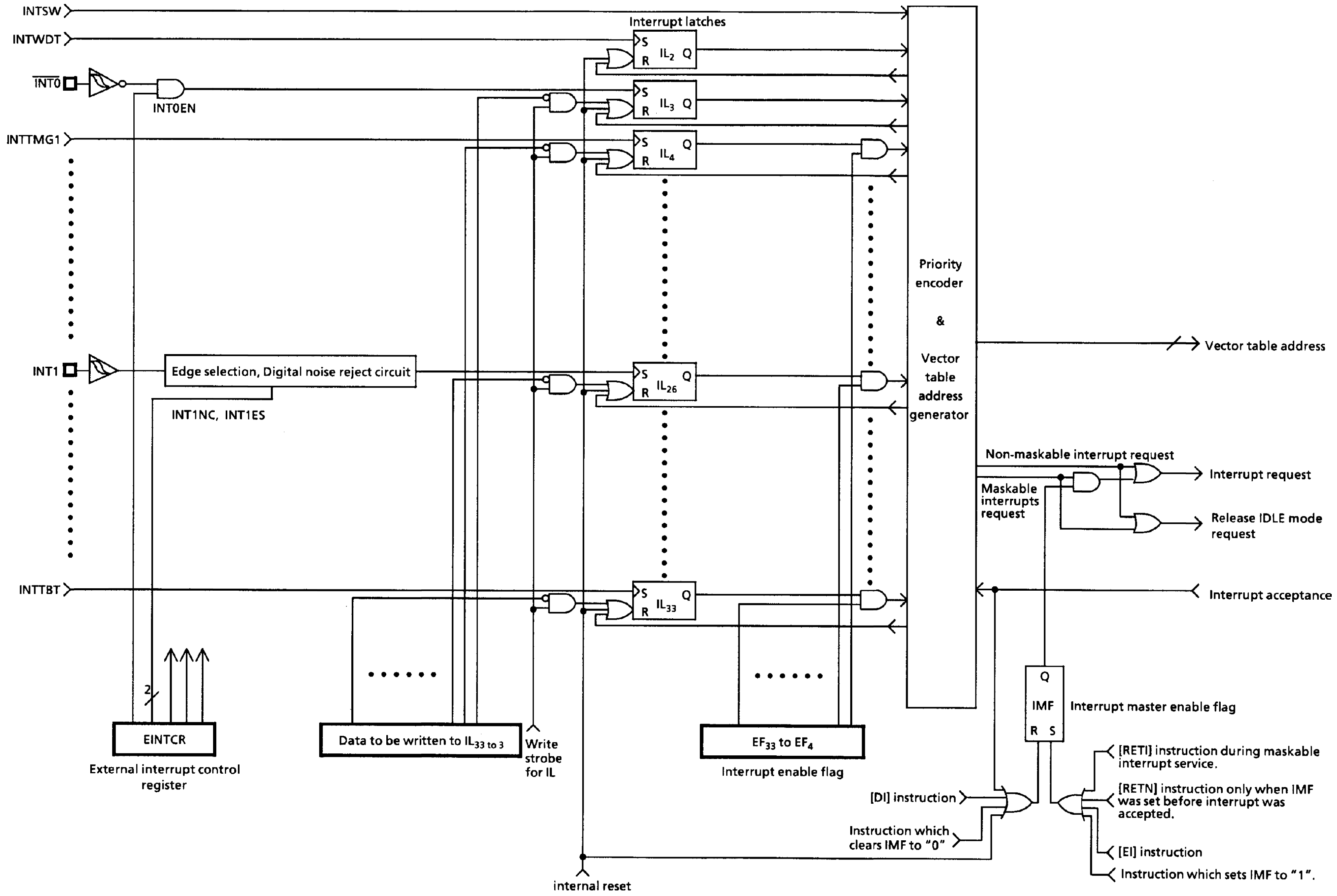

INTSW
INTWDT
INT0
INTTMG1
INT1
INTTBT
INT0EN
Edge selection, Digital noise reject circuit
INT1NC, INT1ES
Interrupt latches
S
R
IL2
Q
IL3
IL4
IL26
IL33
Priority encoder & Vector table address generator
Vector table address
Non-maskable interrupt request
Maskable interrupts request
Interrupt request
Release IDLE mode request
Interrupt acceptance
Q
IMF
R S
Interrupt master enable flag
[RETI] instruction during maskable interrupt service.
[RETN] instruction only when IMF was set before interrupt was accepted.
[EI] instruction
Instruction which sets IMF to "1".
[DI] instruction
Instruction which clears IMF to "0"
internal reset
2
EINTCR
External interrupt control register
Data to be written to IL33 to 3
Write strobe for IL
EF33 to EF4
Interrupt enable flag

Figure 1-14. Interrupt Controller Block Diagram

(1) **Interrupt latches (IL_{33} to IL_2)**

Latches are provided for all interrupt sources except software interrupts. Generation of an interrupt request sets the latch to 1. If the interrupt is enabled, the CPU is requested to accept the interrupt. Immediately the interrupt is accepted, the interrupt latch is zero-cleared. At reset, interrupt latches are all initialized to 0.

The interrupt latches are allocated to addresses 0003C, 0003D, 0002B, 0002E, and $0002F_H$ in SFR. An interrupt latch can be cleared independently by instructions (other than read-modify-write instructions such as bit manipulation and arithmetic instructions). Interrupt requests can be canceled or initialized by program. Note that interrupt latches cannot be set by instruction. Since interrupt latch data can be read, interrupt requests can be tested by software.

```
Example 1:  Clear interrupt latch.
            LDW     (ILL), 1110100000111111B    ;  IL12, IL10 to IL6←0
            LDW     (ILE), 1111111111111000B    ;  IL18 to IL16←0
            LD      (ILC), 00010111B            ;  IL34 to IL33←0
Example 2:  Read interrupt latch.
            LD      WA, (ILL)                   ;  W←ILH, A←ILL
            LD      BC, (ILE)                   ;  B←ILD, C←ILE
            LD      D, (ILC)                    ;  D←ILC
Example 3:  Test interrupt latch.
            TEST    (ILL). 6                    ;  If IL6 = 1, jump.
            JR      F, SSET
```

(2) **Interrupt enable register (EIR)**

Enables or disables interrupt sources except pseudo non-maskable interrupts (software interrupt and watchdog timer interrupt). Pseudo non-maskable interrupts are accepted regardless of the setting in the interrupt enable register. Note that more than one pseudo non-maskable interrupt must not be generated simultaneously.

The interrupt enable register consists of the interrupt master enable flag (IMF) and the interrupt enable flag (EF). EIR is allocated to addresses 0003A, 0003B, 0002B, 0002C, and $0002D_H$ in SFR. Data can be read from or written to the interrupt enable register by instructions (including read-modify-write instructions such as the bit manipulation instruction.)

① Interrupt master enable flag (IMF)

Enables or disables all maskable interrupt requests. If zero-cleared, all maskable interrupt requests are disabled; if set to 1, enabled.

Once an interrupt request is accepted, the interrupt master enable flag is zero-cleared, disabling all subsequent maskable interrupt requests. After interrupt processing is executed, the flag is set to 1 by the maskable interrupt return instruction (RETI) so that interrupts are enabled again. That is, if an interrupt request is already generated, interrupt processing starts immediately after execution of the RETI instruction.

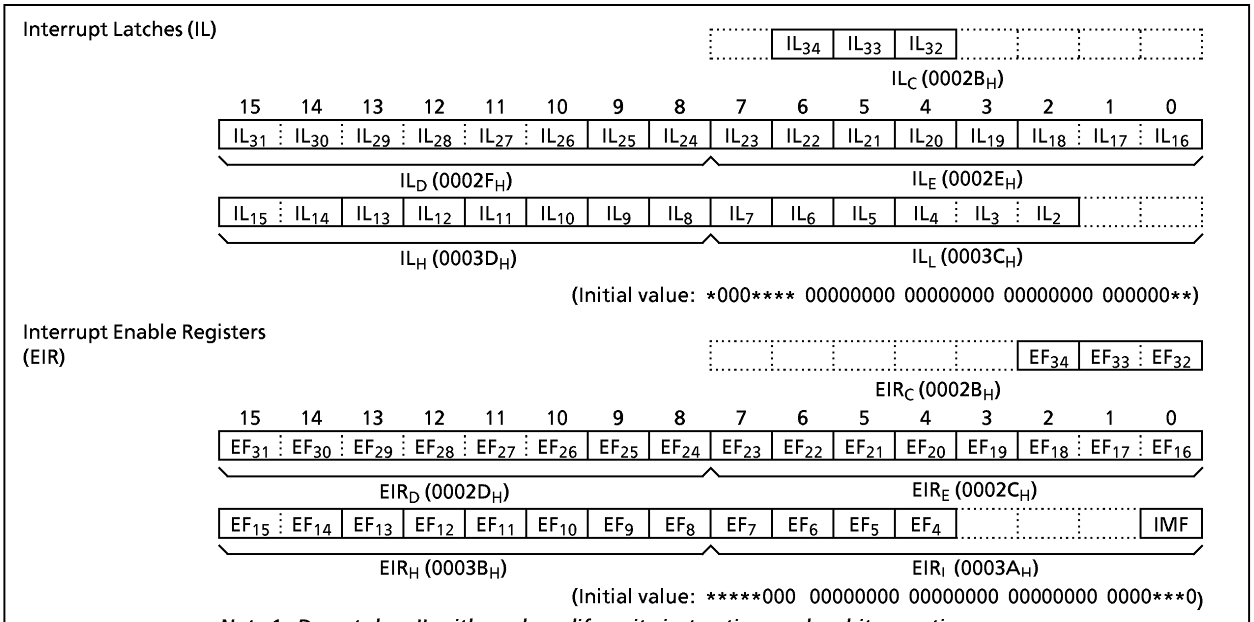
With pseudo non-maskable interrupts, the non-maskable interrupt return instruction (RETN) is used to return. Only when the pseudo non-maskable interrupt processing is entered with interrupt request enabled (IMF = 1), the interrupt master enable flag is set to 1. If the interrupt master enable flag is zero-cleared during the interrupt service program, the interrupt master enable flag remains as 0.

The interrupt master enable flag is allocated to bit 0 in EIR_L (at address $0003A_H$ in SFR). Data can be read from or written to the flag using an instruction. Usually, the flag is set or cleared using the EI or the DI instruction. At reset, the flag is initialized to 0.

② Interrupt enable flag (EF_{34} to EF_4)

Independently enables or disables maskable interrupt requests except external interrupt 0. If set to 1, enables interrupt requests; to 0, disables requests.

```
Example 1:  Set interrupt requests independently and set IMF.
            LD     (EIRC), 01110111B           ;  EF34 to EF32←1
            LDW    (EIRE), 1110011000000001B   ;  EF31 to EF29, EF26, EF25, EF16←1
            LDW    (EIRL), 1110100010100001B   ;  EF15 to EF13, EF11, EF7, EF5, IMF←1
Example 2:  Set interrupt requests independently.
            SET    (EIRH). 4                   ;  EF12←1
```



Interrupt Latches (IL)
IL34 IL33 IL32
IL_C (0002B_H)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
IL31 IL30 IL29 IL28 IL27 IL26 IL25 IL24 IL23 IL22 IL21 IL20 IL19 IL18 IL17 IL16
IL_D (0002F_H)
IL_E (0002E_H)
IL15 IL14 IL13 IL12 IL11 IL10 IL9 IL8 IL7 IL6 IL5 IL4 IL3 IL2
IL_H (0003D_H)
IL_L (0003C_H)
(Initial value: *000**** 00000000 00000000 00000000 000000**)
Interrupt Enable Registers (EIR)
EF34 EF33 EF32
EIR_C (0002B_H)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
EF31 EF30 EF29 EF28 EF27 EF26 EF25 EF24 EF23 EF22 EF21 EF20 EF19 EF18 EF17 EF16
EIR_D (0002D_H)
EIR_E (0002C_H)
EF15 EF14 EF13 EF12 EF11 EF10 EF9 EF8 EF7 EF6 EF5 EF4 IMF
EIR_H (0003B_H)
EIR_l (0003A_H)
(Initial value: *****000 00000000 00000000 00000000 0000***0)

Note 1: Do not clear IL with read-modify-write instructions such as bit operations.
Note 2: Do not set IMF to "1" during non-maskable interrupt service program.
Note 3: Bits 1 and 0 in IL_L are read in as undefined data when a read instruction is executed.
Note 4: Please set IL_{32} to IL_{34} when operate EF_{32} to EF_{34} because IL_C and EIR_C are the same address.
*Note 5: * ; Don't care*

Figure 1-15. Interrupt Latches (IL) and Interrupt Enable Registers (EIR)

1.5.1 Interrupts Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is zero-cleared by reset or instruction. After execution of the current instruction is complete, the interrupt is accepted in 12 machine cycles (3 μs, @ 16 MHz). The interrupt service program ends after executing an interrupt return instruction: RETI for maskable interrupts and RETN for pseudo non-maskable interrupts. Figure 1-16 shows interrupt accept processing timing.

(1) **Interrupt acceptance**

Interrupts are automatically accepted as follows:

① Zero-clears the interrupt master flag (IMF). This temporarily disables subsequent maskable interrupt requests. If a non-maskable interrupt request is accepted, also temporarily disables the subsequent non-maskable interrupt requests.

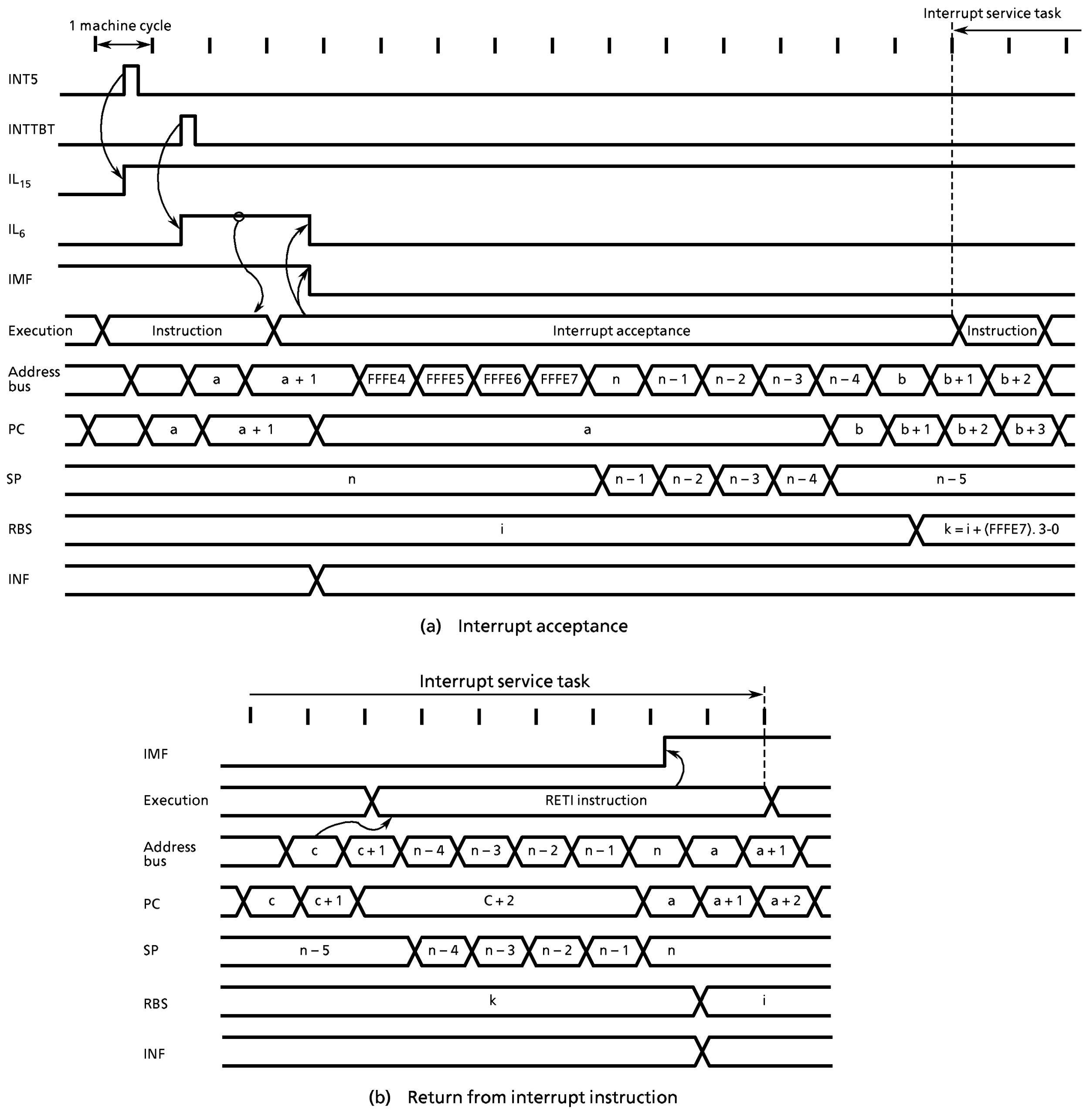
② Zero-clears the latch of the accepted interrupt request.

③ Saves the data in the program counter (PC) and program status word (PSW) to the stack. (Pushes down from PSW_H, PSW_L, PC_E, PC_H, PC_L.) The stack pointer (SP) is decremented five times.

④ Reads the entry address (interrupt vector) of the interrupt service program from the vector table address corresponding to the interrupt source and sets the read entry address in the program counter.

⑤ Reads the register bank selector (RBS) control code from the vector table and adds the lower 4 bits to RBS.

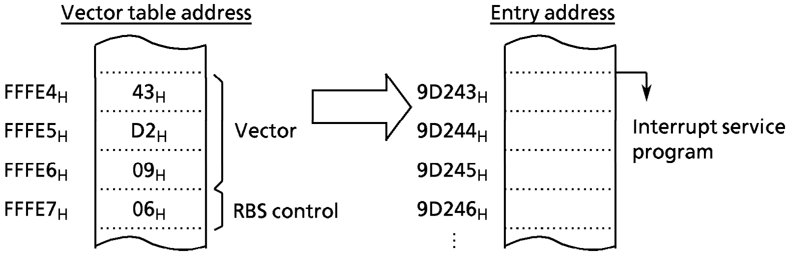
⑥ The instruction stored at the entry address of the interrupt service program is executed.



Note 1: a: return address b: entry address c: address which the RETI instruction is stored.
Note 2: The maximum response time from when an IL is set until an interrupt acceptance processing start is 62/fc [s] with interrupt enabled.

Figure 1-16. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

Example: Correspondence between vector table address for INTTBT processing and interrupt service program entry address



Even when a maskable interrupt source is generated whose priority is higher than the interrupt currently being processed, the later interrupt is not accepted until the interrupt master enable flag is set to "1". Thus, to process multiple interrupt requests, set the interrupt master enable flag to "1" using the interrupt service program. At the same time, enable interrupt requests which are acceptable using the interrupt enable flag. Note that external interrupt 0 cannot be disabled using the interrupt enable flag. Either disable external interrupts (while INT0EN = 0, IL3 is not set, thus the falling edge of input to the $\overline{\text{INT0}}$ pin cannot be detected), or disable interrupt processing by software.

Example 1: Disable external interrupt 0 using the external interrupt control register.

```
        LD      (EINTCR), 00000000B  ;  INT0EN←0
```

Example 2: Disable interrupt processing for external interrupt 0 by software. (The interrupt processing disable switch is bit 0 at address 000F0H.)

```
PINT0:  TEST    (000F0H). 0          ;  If (000F0H) 0 = 1, Returns without processing
                                        interrupt.
        JRS     T, SINT0
        RETI
SINT0:  Interrupt processing
        RETI
          ⋮
VINT0:  DL      PINT0
```

(2) **Saving / Restoring general-purpose registers**

Processing for accepting interrupts automatically saves data from the program counter and the program status word to the stack but not data from the accumulator and other registers, which can be saved by program if necessary. For multiple interrupts, be careful not to overlap data memory when saving the above data.

Data from general-purpose registers are saved in one of the following four ways.

① General-purpose register save / restore by automatic register bank changeover.
Switching to an unused register bank saves data from general-purpose registers at high speed.
Usually, bank 0 is allocated to main tasks ; banks 1 to 15, to interrupt service tasks.
Executing an interrupt return instruction (RETI or RETN) automatically restores banks. Therefore, RBS need not be saved by program.

Example: Switch register banks.

```
PINTxx:  Interrupt processing
         RETI                     ;  Restores bank and returns.
          ⋮
VINTxx:  DP         PINTxx        ;  Interrupt service routine entry address
         DB         1             ;  RBS←RBS + 1
```

② General-purpose register save / restore by register bank changeover
Switching to an unused register bank saves data from general-purpose registers at high speed.
Usually, bank 0 is allocated to main tasks ; banks 1 to 15, to interrupt service tasks. To improve data memory efficiency, allocate the common bank to the non-multiple interrupt sources.
Executing an interrupt return instruction (RETI or RETN) automatically restores banks. Therefore, RBS need not be saved by program.

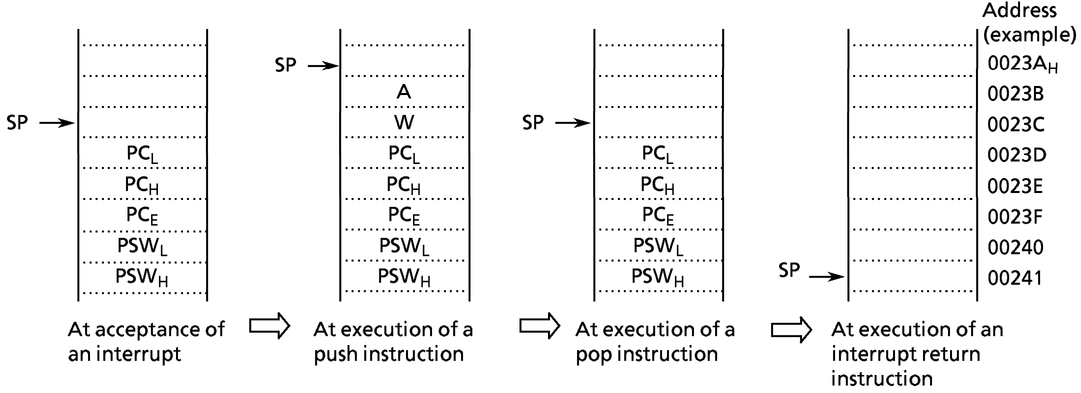
Example: Switch register banks.

```
PINTxx:  LD         RBS, n           Interrupt processing
         RETI                     ;  Restores bank and returns.
          ⋮
VINTxx:  DP         PINTxx        ;  Interrupt service routine entry address
         DB         0             ;
```

③ General-purpose registers save / restore using push and pop instructions
To save data from a specific register or to generate more than one interrupt request for the same interrupt source, save or restore data from the general-purpose register using the push and pop instructions.

Example: Save and restore register using the push and pop instructions.

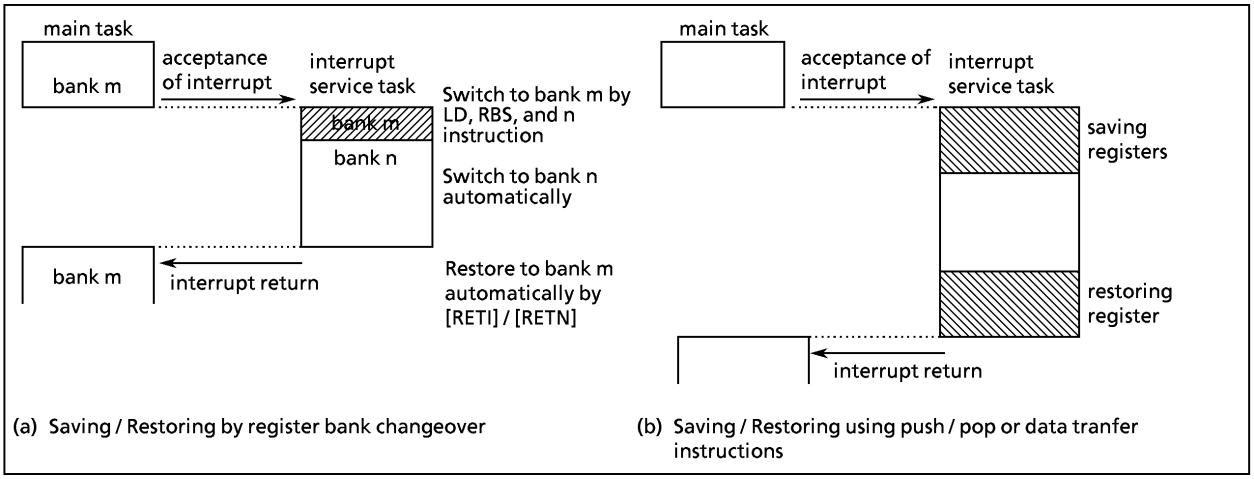
```
PINTxx:  PUSH        WA            ;  Saves WA register pair to stack.
         Interrupt processing
         POP         WA            ;  Restores WA register pair from stack.
         RETI                      ;  Returns.
```



④ General-purpose registers save / restore using data transfer instructions
For non-multiple interrupt processing, to save data from a specific register, save and restore data between the register and data memory using the load instruction.

Example: Save and restore data between register and data memory using the load instruction

```
PINTxx:  LD          (GSAVA), A    ;  Saves register A.
         Interrupt processing
         LD          A, (GSAVA)    ;  Restores register A.
         RETI                      ;  Returns.
```



(a) Saving / Restoring by register bank changeover

(b) Saving / Restoring using push / pop or data tranfer instructions

Figure 1-17. Saving/Restoring General-Purpose Registers

(3) **Interrupt return**

The interrupt return instructions [RETI] / [RETN] perform the following operations.

[RETI] Maskable interrupt return	[RETN] Non-maskable interrupt return
① The contents of the program counter and the program status word are restored from the stack. ② The stack pointer is incremented 5 times. ③ The interrupt master enable flag is set to "1". ④ The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.	① The contents of the program counter and program status word are restored from the stack. ② The stack pointer is incremented 5 times. ③ The interrupt master enable flag is set to "1" only when a non-maskable interrupt is accepted in interrupt enable status. However, the interrupt master enable flag remains at "0" when so clear by an interrupt service program. ④ The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.

Interrupt requests are sampled during the final cycle of the instruction being executed. Thus, the next interrupt can be accepted immediately after the interrupt return instruction is executed.

Note: When the interrupt processing time is longer than the interrupt request generation time, the interrupt service task is performed but not the main task.

1.5.2 Software Interrupt (INTSW)

Executing the SWI instruction generates a software interrupt and immediately starts interrupt processing (highest priority). However, if a non-maskable interrupt is already being processed, executing the SWI instruction does not generate a software interrupt and processing is the same as for the NOP instruction. Do not use the SWI instruction for other than detecting address errors or debugging as described below:

① Address error detection

If the CPU fetches an instruction from an address not found in memory for some reason (eg, noise), FF_H is read. Code FF_H is the SWI instruction; thus, a software interrupt is generated so that an address error is detected. Filling the unused program memory area with FF_H widens the range for detecting address errors. If the CPU fetches an instruction from RAM or the SFR area, an address trap is generated.

② Debugging

Allocating the SWI instruction to the software break point set address improves debugging efficiency.

1.5.3 External Interrupts

TMP88CS48A supports six external interrupts out of which four are provided with a digital noise rejection circuit (rejects pulse input shorter than the specified time as noise).

Edge can be selected for pins INT1 to INT4. The $\overline{\text{INT0}}$/P10 pin can be specified for external interrupt input or as an I/O port. At reset, the pin is set to input port.

To select edge, control noise rejection, or select $\overline{\text{INT0}}$/P10 pin function, use the external interrupt control register.

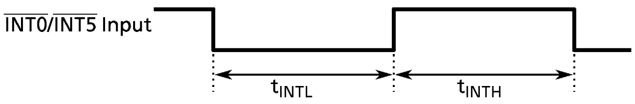
Table 1-3. External Interrupts

Source	Pin	Secondary function pin	Enable conditions	Edge	Digital noise reject
INT0	$\overline{\text{INT0}}$	P10	IMF = 1, INT0EN = 1	falling edge	— (hysteresis input)
INT1	INT1	P11	IMF · EF_{26} = 1	falling edge or rising edge	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 48/fc or 192/fc [s] or more are considered to be signals.
INT2	INT2	P12/TC1	IMF · EF_{27} = 1		pulses of less than 7/fc [s] are eliminated as noise. Pulses of 24/fc [s] or more are considered to be signals.
INT3	INT3	P50/TC3	IMF · EF_{28} = 1		
INT4	INT4	P51/TC4	IMF · EF_{29} = 1		
INT5	$\overline{\text{INT5}}$	P20/$\overline{\text{STOP}}$	IMF · EF_{15} = 1	falling edge	— (hysteresis input)

Note 1: The noise reject function is turned off in SLOW and SLEEP modes. Also, the noise reject times are not constant for pulses input while transiting between operating modes .

Note 2: The noise reject function is also affected for timer / counter input (TC1 pin).

Note 3: The pulse width (both "H"and "L" level) for input to the $\overline{\text{INT0}}$ and $\overline{\text{INT5}}$ pins must be over 2 machine cycle.



t_{INTL}, t_{INTH} > 2tcyc
NB: tcyc = 4/fc [s] (In NORMAL or IDLE mode)

Note 4: If a noiseless signal is input to the external interrupt pin in the NORMAL 1/2 or IDLE 1/2 mode, the maximum time from the edge of input signal until the IL is set is as follows:

① INT1 pin 49/fc [s] (INT1NC = 1), 193/fc [s] (INT1NC = 0)
② INT2 to INT4 pin 25/fc [s]

Note 5: Even if the falling edge of $\overline{\text{INT0}}$ pin input is detected at INT0EN = 0, the interrupt latch IL_3 is not set.

EINTCR
(00037_H)

7	6	5	4	3	2	1	0
INT1 NC	INT0 EN		INT4 ES	INT3 ES	INT2 ES	INT1 ES	

(Initial value: 00** *00*)

INT1NC	Noise reject time select	0: Pulses of less than 63/fc [s] are eliminated as noise 1: Pulses of less than 15/fc [s] are eliminated as noise	R/W
INT0EN	P10/$\overline{\text{INT0}}$ pin configuration	0: P10 input / output port 1: $\overline{\text{INT0}}$ pin (Port P10 should be set to an input mode)	
INT4 ES INT3 ES INT2 ES INT1 ES	INT4 to INT1 edge select	0: Rising edge 1: Falling edge	

*Note: fc ; High-frequency clock [Hz] * ; Don't care*

Figure 1-18. External Interrupt Control Register

1.6 Reset Circuit

TMP88CS48A supports four types of reset generation: external reset input, address trap reset output, watchdog timer reset output, and system clock reset output.

Table 1-3 shows initialization of the internal hardware by reset.

At power on, the internal reset output circuits (watchdog timer reset, address trap reset, and system clock reset) are not initialized. Thus, at power on, the $\overline{\text{RESET}}$ pin may output "L" level for up to 24/fc [s] (1.5 μs, @ 16 MHz).

Table 1-4. Initialization of Internal Hardware by Reset

Internal hardware	Initial value	Internal hardware	Initial value
Program counter (PC)	(FFFFC$_H$ to FFFFE$_H$)	Prescaler and Divider of timing generator	0
Stack pointer (SP)	not initialized.		
General-purpose register (WABCDEHL)	not initialized.		
Register bank selector (RBS)	0	Watchdog timer	Enable
Jump status flag (JF)	1		
Zero flag (ZF)	not initialized.	Output latches I/O ports	Refer to I/O port circuitry
Carry flag (CF)	not initialized.		
Half carry flag (HF)	not initialized.		
Sign flag (SF)	not initialized.		
Overflow flag (VF)	not initialized.		
Interrupt master enable flag (IMF)	0		
Interrupt individual enable flag (EF)	0	Control registers	Refer to each of control register
Interrupt latches (IL)	0		
		RAM	not initialized.

1.6.1 External Reset Input

The $\overline{\text{RESET}}$ pin has a pull-up resistor for hysteresis input. When the supply voltage is within the operating voltage and oscillation is stabilized, holding the $\overline{\text{RESET}}$ pin at "L" level for a minimum of 3 machine cycles (12/fc [s]) generates reset and initializes the internal states.

When the $\overline{\text{RESET}}$ pin input reaches "H" level, reset is released. Execution of the program starts from the vector address stored at addresses FFFFC to FFFFE$_H$.

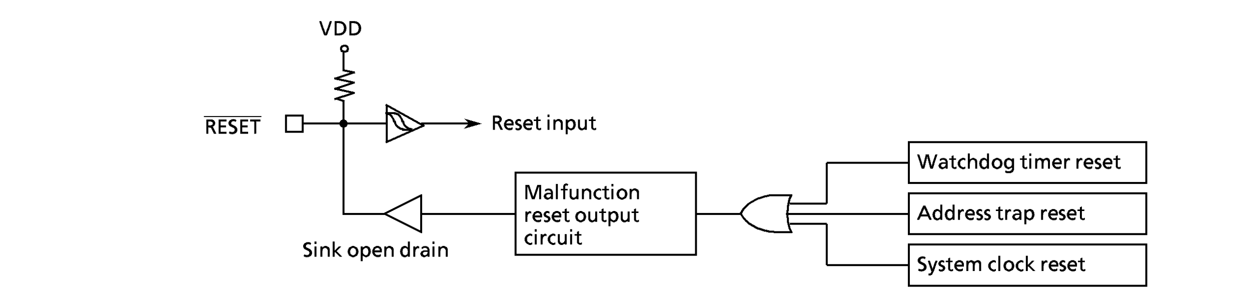
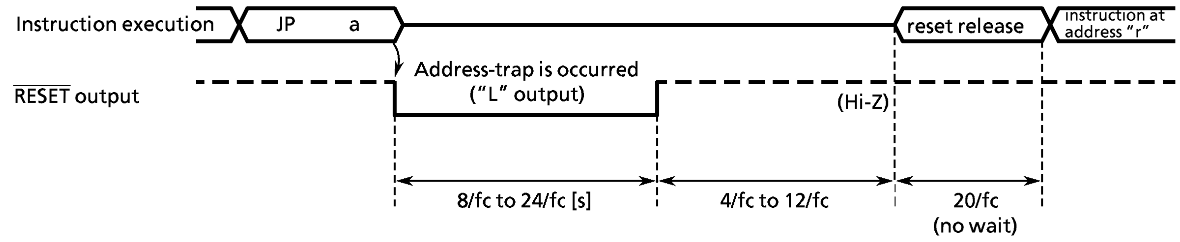


Figure 1-19. Reset Circuit

1.6.2 Address-Trap-Reset

When CPU runaway is caused by noise, for example, and an attempt is made to fetch an instruction from on-board RAM or the SFR, an internal reset is generated and a reset signal ("L" level) is output from the $\overline{\text{RESET}}$ pin. The reset signal is output for 8/fc to 24/fc [s] (0.5-1.5 μs @ 16 MHz)


Instruction execution
JP a
reset release
instruction at address "r"
$\overline{\text{RESET}}$ output
Address-trap is occurred ("L" output)
(Hi-Z)
8/fc to 24/fc [s]
4/fc to 12/fc
20/fc (no wait)

Note 1: Address "a" is in the SFR or on-chip RAM space.
Note 2: During reset release, reset vector "r" is read out, and an instruction at address "r" is fetched and decoded.

Figure 1-20. Address Trap Reset

1.6.3 System-Clock-Reset

Zero-clearing bits 7 and 6 in SYSCR2 stops the system clock and causes MCU deadlock. To avoid deadlock, detecting that bits 7 and 6 in SYSCR2 are both set to 0 automatically generates a reset signal and continues oscillation. Reset signal is output from the $\overline{\text{RESET}}$ pin. Reset signal is output for 8/fc to 24/fc [s] (0.5 to 1.5 μs @ 16 MHz)

2. On-Chip Peripheral Functions

2.1 Special Function Registers (SFR) and Data Buffer Registers (DBR)

The TLCS-870/X series uses memory mapped I/O: peripheral hardware control and data transfer are performed using the special function registers (SFR) and data buffer registers (DBR).

SFRs are mapped to addresses 00000 to 0003F$_H$; DBR, to addresses 00F80 to 00FBF$_H$ and 00FFB to 00FFF$_H$.

Figure 2-1 shows the TMP88CS48A SFRs and DBRs.

Address	Read	Write
00000$_H$	Port P0	
01	Port P1	
02	Port P2	
03	Port P3	
04	Port P4	
05	Port P5	
06	Port P6	
07	Port P7	
08	reserved	
09	reserved	
0A	–	P0CR (Port P0 I/O control)
0B	–	P1CR (Port P1 I/O control)
0C	–	P6CR (Port P6 I/O control)
0D	–	P7CR (Port P7 I/O control)
0E	ADCCR (AD converter control)	
0F	ADCDR1 (The upper 8 bits of AD conv. result)	
10	–	TREG1A$_L$ (Timer register 1A)
11	–	TREG1A$_H$
12	TREG1B$_L$ (Timer register 1B)	
13	TREG1B$_H$	
14	–	TC1CR (TC1 control)
15	–	TC2CR (TC2 control)
16	–	TREG2$_L$ (Timer register 2)
17	–	TREG2$_H$
18	TREG3A (Timer register 3A)	
19	TREG3B (Timer register 3B)	–
1A	–	TC3CR (TC3 control)
1B	–	TREG4 (Timer register 4)
1C	–	TC4CR (TC4 control)
1D	–	TREG5 (Timer register 5)
1E	–	TC5CR (TC5 control)
1F	RCCR (Remote control receive control)	

Address	Read	Write
00020$_H$	–	SBICR1 (SBI control 1)
21	SBIDBR (SBI data buffer)	
22	–	I2CAR (I^{2}C bus address)
23	SBISR (SBI status)	SBICR2 (SBI control 2)
24	ADCDR2L (The lower 8 bits of AD conv. result)	
25	ADCDR2H (The lower 2 bits of AD conv. result)	
26	HPWMCR (HPWM control)	
27	–	HPWMDR0 (HPWM0 data)
28	–	HPWMDR1 (HPWM1 data)
29	reserved	
2A	reserved	
2B	EIRC (Extended interrupt enable register/extended interrupt latch)	
2C	EIRE (Extended interrupt enable register lower)	
2D	EIRD (Extended interrupt enable register upper)	
2E	ILE (Extended interrupt latch lower)	
2F	ILD (Extended interrupt latch upper)	
30	reserved	
31	reserved	
32	reserved	
33	reserved	
34	–	WDTCR1 (WDT control 1)
35	–	WDTCR2 (WDT control 2)
36	TBTCR (TBT/TG/DVO control)	
37	EINTCR (External interrupt control)	
38	SYSCR1 (System control 1)	
39	SYSCR2 (System control 2)	
3A	EIRL (Interrupt enable register)	
3B	EIRH	
3C	ILL (Interrupt latch)	
3D	ILH	
3E	PSWL (Program status word)	
3F	PSWH	

Figure 2-1. SFR & DBR (1/2)

PMD

Address	Read	Write
$00F80_H$	PDCRA (Position detect control register)	
81	PDCRB	
82	SDREG (Sampling delay control register)	
83	CMPCR (Compare control register)	
84	MTCRA (Mode timer control register)	
85	MTCRB	
86	MCAPA (Mode capture register)	–
87	MCAPB	–
88	CMP1A (Compare register 1)	
89	CMP1B	
8A	CMP2A (Compare register 2)	
8B	CMP2B	
8C	CMP3A (Compare register 3)	
8D	CMP3B	
8E	CMP4A (Compare register 4)	
8F	CMP4B	
90	EMGCRA (EMG control register)	
91	EMGCRB	
92	DTR (Dead time register)	
93	MDCR (PMD control register)	
94	MDOUTA (PWD output register)	
95	MDOUTB	
96	MDCNTA (PMD counter)	EMGREL (EMG release register)
97	MDCNTB	–
98	CMPUA (PMD compare register)	
99	CMPUB	
9A	CMPVA (PMD compare register)	
9B	CMPVB	
9C	CMPWA (PMD compare register)	
9D	CMPWB	
9E	MDPRDA (PMD period register)	
9F	MDPRDB	

UART, TC6

Address	Read	Write
$00FFB_H$	RDBUF (UART receive buffer)	TDBUF (UART transmit buffer)
C	UARTSR (UART status register)	UARTCR1 (UART control 1)
D	–	UARTCR2 (UART control 2)
E	–	TREG6 (Timer register 6)
F	–	TC6CR (TC6 control)

Figure 2-1. SFR & DBR (2/2)

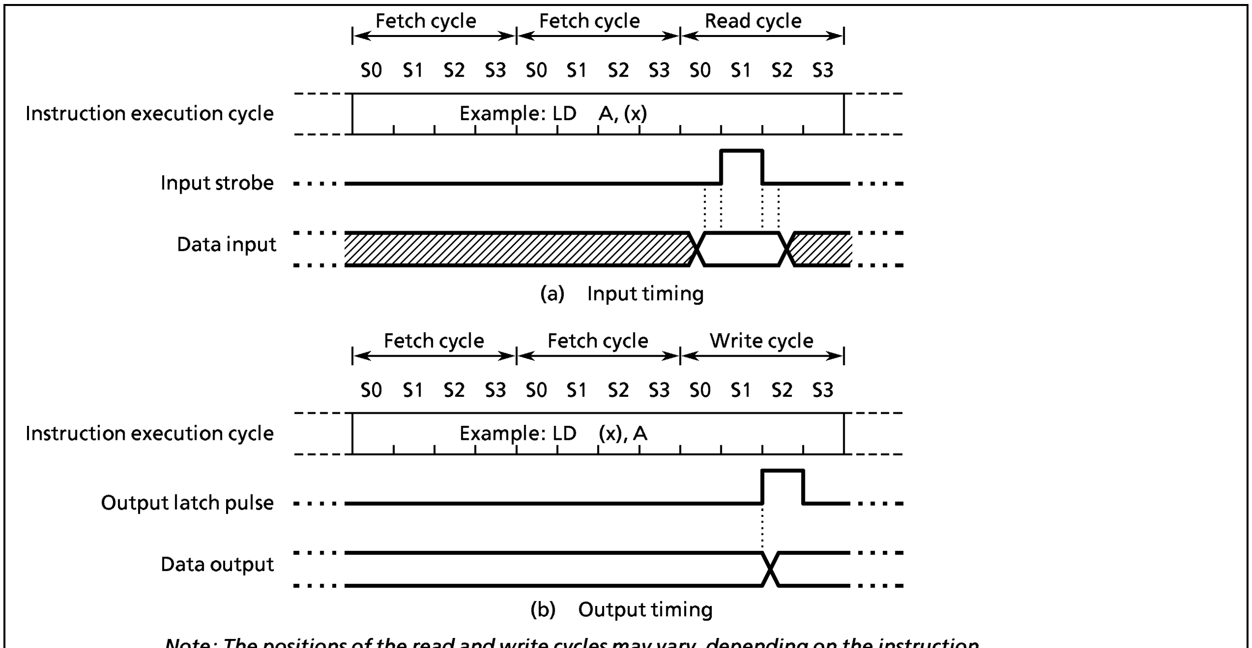
2.2 I/O Ports

TMP88CS48A features eight built-in I/O ports (56 pins).

	Primary Function	Secondary Function
Port P0	8-bit I/O port	motor control circuit I/O
Port P1	8-bit I/O port	external interrupt, timer/counter I/O, divider output, and motor control circuit input
Port P2	3-bit I/O port	external interrupt and STOP mode release signal input
Port P3	8-bit I/O port	–
Port P4	8-bit I/O port	serial interface I/O, and motor control circuit input
Port P5	5-bit I/O port	external interrupt input and timer/counter I/O
Port P6	8-bit I/O port	analog input
Port P7	8-bit I/O port	analog input

All output ports have built-in latches in which data are held. Input ports do not have latches; thus, data should be held externally until read externally, or held after more than one read. Figure 2-2 shows data input and output timings.

External data are read from I/O ports at the S1 state of the read cycle at instruction execution. This timing cannot be checked externally; thus, transient input data such as chattering should be handled by program. Data are output to I/O ports at the S2 state of the write cycle at instruction execution.



Note: The positions of the read and write cycles may vary, depending on the instruction.

Figure 2-2. Input / Output Timing (Example)

When data are read from an I/O port other than a programmable I/O port, whether pin input values or output latch data are read depends on the instruction as follows.

(1) **Instructions which read output latch data:**

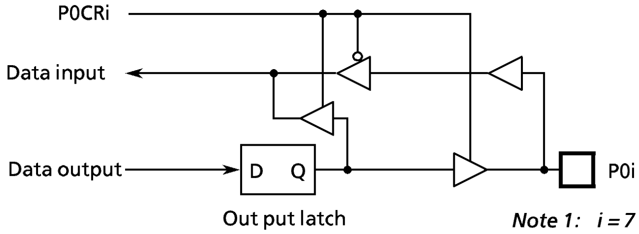
① XCH r, (src)	⑤ LD (pp). b,CF
② SET/CLR/CPL (src). b	⑥ XCH CF, (src).b
③ SET/CLR/CPL (pp). g	⑦ ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), n
④ LD (src). b, CF	⑧ ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (src)-side for (HL) instruction.
	⑨ MXOR (src), m

(2) **Instructions which read pin input values:**
Instructions other than (1) above, and ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (HL)-side for (HL) instructions.

2.2.1 Port P0 (P07 to P00)

Port P0 is an 8-bit general-pupose I/O port whose input or output is specifiable in units of bits. Specify input or output using the Port P0 I/O control register (P0CR). At reset, P0CR is initialized to 0, port P0 is set to input mode, and the port P0 output latches are initialized to 0.

> *Note: Input mode port reads the state of input pin. When input / output mode is used to mixed, the contents of output latch setting to the input mode port may be overwritten by executing bit manipulation instructions.*



Note 1: i = 7 to 0
Note 2: STOP is bit 7 in SYSCR1 ; OUTEN, bit 4 in SYSCR1.

P0 (00000$_H$) (Initial value: 0000 0000)

7	6	5	4	3	2	1	0
P07 $\overline{CL1}$ PDW1	P06 $\overline{EMG1}$	P05 U1	P04 V1	P03 W1	P02 X1	P01 Y1	P00 Z1

P0CR (0000A$_H$) (Initial value: 0000 0000)

7	6	5	4	3	2	1	0

P0CR	I/O control for port P0 (specified for each bit)	0: Input mode 1: Output mode	Write only

Figure 2-3. Port P0 and P0CR

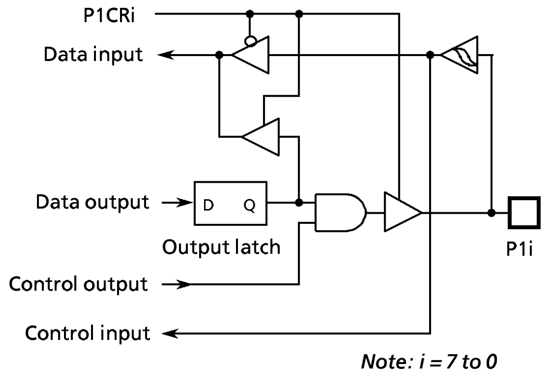
Example: Set upper 4 bits of port P0 to input mode and lower 4 bits to output mode. The initial output value is 1010B.

```
LD      (P0), 00001010B     ;  Sets initial value for port P0 output latches.
LD      (P0CR), 00001111B   ;  Sets I/O mode for port P0.
```

2.2.2 Port P1 (P17 to P10)

Port P1 is an 8-bit port whose input or output is specifiable in units of bits. Specify input or output using the Port P1 I/O control register (P1CR). At reset, P1CR is initialized to 0, port P1 is set to input mode, and the port P1 output latches are initialized to 0.

To use port P1 pins as function pins, set the input pins to input mode. For output mode, set the output latches for the output pins to 1. We recommend you use the P11 and P12 pins for external interrupt input, timer/counter input or as an input port. (If used as an output port, the interrupt latches are set by a rising or falling edge.) The P10 pin can be used for either I/O port or external interrupt input according to the external interrupt control register (INT0EN). At reset, the P10 pin is set to an input port.



Note: i = 7 to 0

	7	6	5	4	3	2	1	0	
P1 (00001_H)	P17 PDU1	P16 PDV1	P15 $\overline{\text{DVO}}$ TC2	P14 $\overline{\text{PPG}}$	P13 $\overline{\text{PWM5}}$ $\overline{\text{PDO5}}$	P12 INT2 TC1	P11 INT1	P10 $\overline{\text{INT0}}$	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P1CR ($0000B_H$)									(Initial value: 0000 0000)

P1CR	I/O control for port P1. (specified for each bit.)	0: Input mode 1: Output mode	Write only

Figure 2-4. Port P1 and P1CR

Example: Set P17 and P16 to an output port, P13 and P11 to an input port, the other pins as function pins. P17 outputs 1 and P16 outputs 0.

```
LD      (EINTCR), 01000000B     ;   INT0EN←1
LD      (P1), 10111111B         ;   P17←1, P14←1, P16←0
LD      (P1CR), 11010000B
```

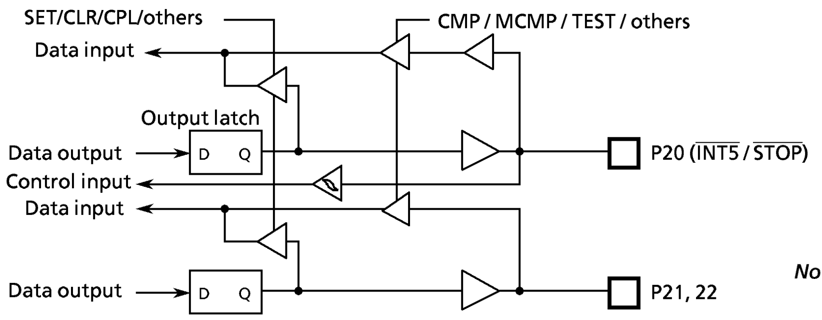
Note: Input mode port reads the state of input pin. When input / output mode is used to mixed, the contents of output latch setting to the input mode port may be overwritten by executing bit manipulation instructions.

2.2.3 Port P2 (P22 to P20)

Port P2 is a 3-bit I/O port. It is also used for external interrupt input, and STOP mode release signal. To use port P2 as function pins or an input port, set the output latches to 1. At reset, the output latches are initialized to 1.

We recommend that the P20 pin be used for external interrupt input, STOP mode release signal input, or input port. (If used as an output port pin, the interrupt latch is set at a falling edge.)

When the read instruction is executed for port P2, undefined values are read from bits 7 to 3.



*Note : * ; Don't care*

	7	6	5	4	3	2	1	0
P2 (00002_H)						P22	P21	P20 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$

(Initial value: **** *111)

Figure 2-5. Port P2

2.2.4 Port P3 (P37 to P30)

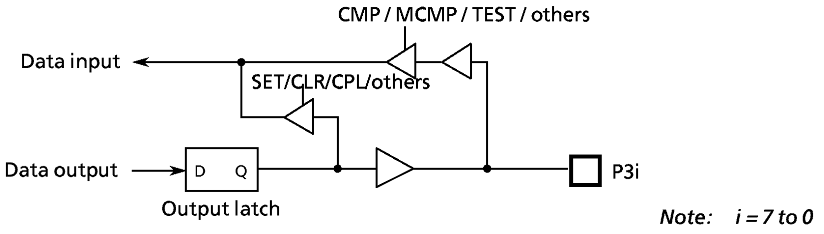
Port P3 is an 8-bit I/O port whose input or output is specifiable in units of bits. The port can handle a large output current to enable direct drive of LEDs. At reset, the port P3 output latches are initialized to 1.

Example 1: Output immediate value 5AH from port P3.

```
        LD      (P3), 5AH          ;   P3←5AH
```

Example 2: Invert output or upper 4 bits (P37 to P34) in port P3.

XOR (P3), 11110000B ; P37 to P34←$\overline{P37}$ to $\overline{P34}$



Note: i = 7 to 0

P3 (00003$_H$) (Initial value: 1111 1111)

7	6	5	4	3	2	1	0
P37	P36	P35	P34	P33	P32	P31	P30

Figure 2-6. Port P3

2.2.5 Port P4 (P47 and P40)

To use port P4 as an input port or as function pins, set the output latches to 1. At reset, the port P4 output latches are initialized to 1.

2.2.6 Port P5 (P54 to P50)

Port P5 is a 5-bit I/O port. To use port P5 as input pins or function pins, set the output latches to 1. At reset, the port P5 output latches are initialized to 1.

When the read instruction is executed for port P5, 1s are read from bits 7 to 5.

Example: Clear the P53 pin ("L" level output)

```
        CLR     (P5). 3       ;   P53←0
```

P4 (00004$_H$)

7	6	5	4	3	2	1	0
P47 SDA SO	P46 SCL SI	P45 $\overline{SCK}$	P44 TxD	P43 RxD	P42	P41	P40

(Initial value: 1111 1111)

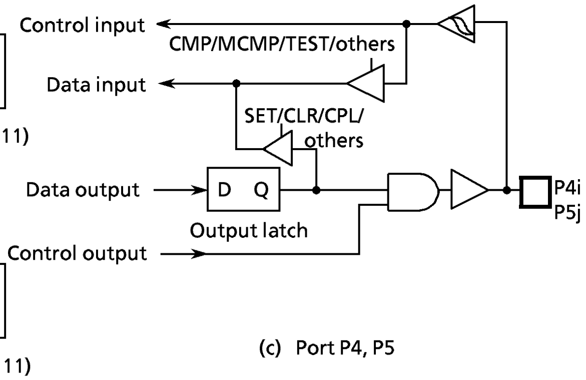
(a) Port P4

P5 (00005$_H$)

7	6	5	4	3	2	1	0
"1"	"1"	"1"	P54 $\overline{HPWM1}$	P53 $\overline{HPWM0}$	P52 $\overline{PWM4}$ $\overline{PDO4}$	P51 INT4 TC4	P50 INT3 TC3

(Initial value: ***1 1111)

(b) Port P5



(c) Port P4, P5

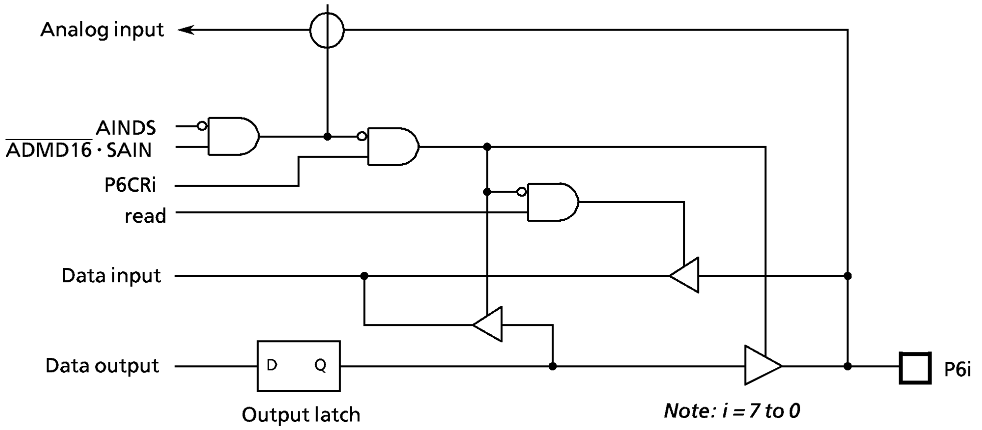
*Note: i = 7 to 0, j = 4 to 0 * ; Don't care*

Figure 2-7. Port P4 and P5

2.2.7 Port P6 (P67 to P60)

Port P6 is an 8-bit I/O port whose input or output is specifiable in units of bits. Port P6 is also used for analog input. Specify input or output using the port P6 I/O control register (P6CR) and AINDS (bit 4 in ADCCR).

At reset, P6CR is set to 1; AINDS is cleared to 0 ; thus, port P6 is set to analog input. At reset, the port P6 output latches are initialized to 0. P6CR is a write-only register. When using the AD converter, the port P6 pins not used for analog input can be used as input port pins.



Note: i = 7 to 0

	7	6	5	4	3	2	1	0	
P6 (00006$_H$)	P67 AIN07	P66 AIN06	P65 AIN05	P64 AIN04	P63 AIN03	P62 AIN02	P61 AIN01	P60 AIN00	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P6CR (0000C$_H$)									(Initial value: 0000 0000)

P6CR	I/O control for port P6 (specified for each bit)	0: 1: } As listed below	Write only

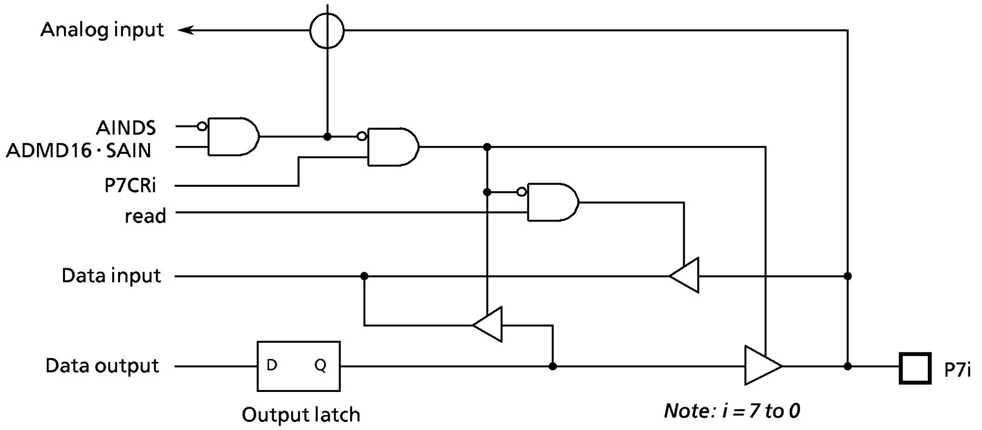
ADCCR		P6CR	
ADMD16	AINDS	0	1
0	0	Input port except for the analog input selected by SAIN	Output port except for the analog input selected by SAIN
1	0	Input port	Output port
0	1	Input port	Output port
1	1	Input port	Output port

Figure 2-8. Port P6 and Port P6CR

Note: Input mode port reads the state of input pin. When input / output mode is used to mixed, the contents of output latch setting to the input mode port may be overwritten by executing bit manipulation instructions.

2.2.8 Port P7 (P77 to P70)

Port P7 is an 8-bit I/O port whose input or output is specifiable in units of bits. Port P7 is also used for analog input. Specify input or output by port P7 I/O control register (P7CR) and AINDS (bit 4 in ADCCR). At reset, P7CR is cleared to 1, AINDS to 0. At reset, the port P7 output latches are initialized to 0. P7CR is a write-only register. When using the AD converter, the port pins not used for analog input can only be used as input port pins.



	7	6	5	4	3	2	1	0	
P7 (00007$_H$)	P77 AIN17	P76 AIN16	P75 AIN15	P74 AIN14	P73 AIN13	P72 AIN12	P71 AIN11	P70 AIN10	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P7CR (0000C$_H$)									(Initial value: 0000 0000)

P7CR	I/O control for port P7 (specified for each bit)	0: } As listed below 1:	Write only

ADCCR		P7CR	
ADMD16	AINDS	0	1
0	0	Input port	Output port
1	0	Input port except for the analog input selected by SAIN	Output port except for the analog input selected by SAIN
0	1	Input port	Output port
1	1	Input port	Output port

Figure 2-9. Port P7 and P7CR

Note: Input mode port reads the state of input pin. When input / output mode is used to mixed, the contents of output latch setting to the input mode port may be overwritten by executing bit manipulation instructions.

Example: Set the lower 4 bits of port P7 to an output port and the remaining bits to an input port.

```
        LD        (P7CR), 0FH        ;    P7CR←00001111
```

2.3 Time Base Timer (TBT)

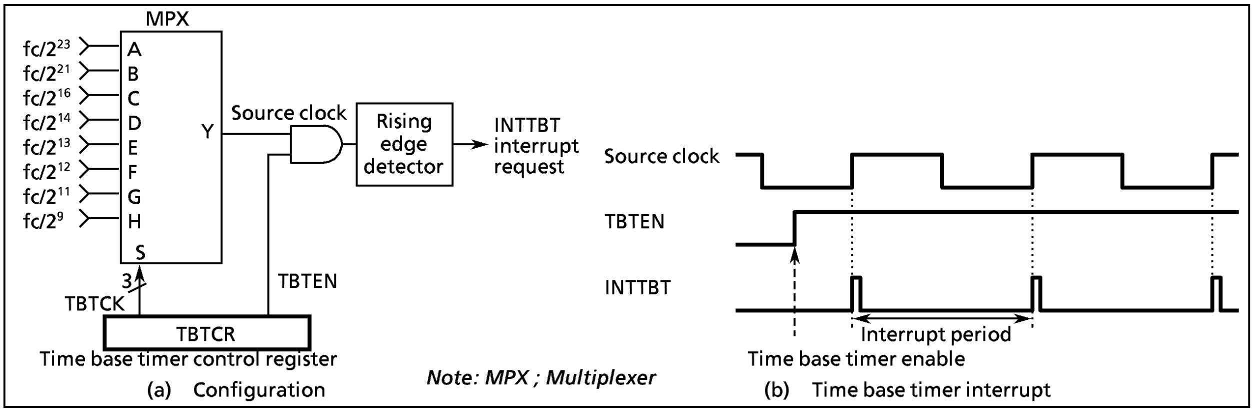
The time base timer is the reference time generation timer for key scan and dynamic display processing; it generates the time base timer interrupt (INTBT) at a fixed cycle.

After the time base timer is enabled, time base timer interrupts are generated at the first rising edge of the source clock (select timing generator divider output using TBTCK). Sine the divider is not cleared by program, only the first interrupt may be generated before the set interrupt cycle. (Figure 2-10 (b))

Select the interrupt frequency with the time base timer disabled. (When switching to timer disabled from timer enabled, do not change the set interrupt frequency.) Note that selecting the frequency and enabling the time base timer can be performed simultaneously.

Example: Set time base timer interrupt frequency to fc/2^{16} [Hz] and enable interrupts.

```
LD      (TBTCR), 00001010B
SET     (EIRL). 6
```



(a) Configuration

Note: MPX ; Multiplexer

(b) Time base timer interrupt

Figure 2-10. Time Base Timer

TBTCR
(00036H)

7	6	5	4	3	2	1	0
(DVOEN)	(DVOCK)		(DV7CK)	TBTEN	TBTCK		

(Initial value: 0**0 0***)

TBTEN	Time base timer enable / disable	0: Disable 1: Enable	R/W
TBTCK	Time base timer interrupt frequency select [Hz]	000: fc/2^{23} 001: fc/2^{21} 010: fc/2^{16} 011: fc/2^{14} 100: fc/2^{13} 101: fc/2^{12} 110: fc/2^{11} 111: fc/2^{9}	

*Note: fc: High frequency clock [Hz] * ; Don't care*

Figure 2-11. Time Base Timer Control Register

Table 2-1. Time Base Timer Interrupt Frequency (Example ; fc = 16 MHz)

TBTCK	Time base timer interrupt frequency [Hz]
000	1.91
001	7.63
010	244.14
011	976.56
100	1953.12
101	3906.25
110	7812.50
111	31250

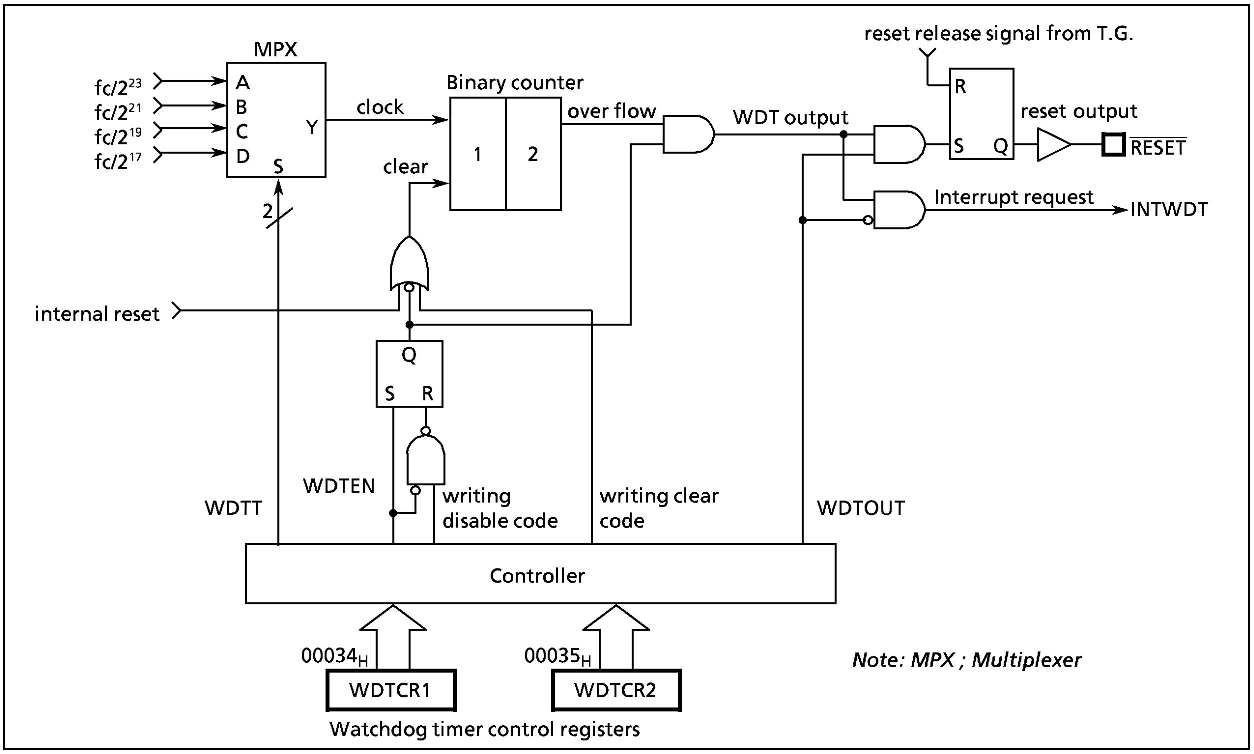
2.4 Watchdog Timer (WDT)

The watchdog timer is a fail-safe function, which detects misoperation (runaway) of the CPU caused by, for example, noise or deadlock, and returns it to normal operation.

Runaway signals detected by the watchdog timer, selectable by program, can be either reset output or pseudo non-maskable interrupt request. Note that this selection can be mode only once. At reset release, initialized to reset output.

When the watchdog timer is not used for detecting runaway, it can be used as a timer which generates interrupts at a fixed cycle.

2.4.1 Watchdog Timer Configuration



Note: MPX ; Multiplexer

Figure 2-12. Watchdog Timer Configuration

2.4.2 Watchdog Timer Control

Figure 2-13 shows the watchdog timer control registers. Releasing reset enables the watchdog timer.

(1) **How to detect runaway using the watchdog timer**

CPU runaway is detected as follows:

① Set detection time, select output, and clear the binary counter.

② Repeat clearing of the binary counter within every set detection time.

If runaway or deadlock occurs, and the binary counter is not cleared, the binary counter overflows causing the watchdog timer output (WDTOUT) to be active. At this time, if WDTOUT = 1, reset signals are output from the $\overline{\text{RESET}}$ pin and the internal hardware is reset. If WDTOUT = 0, a watchdog timer interrupt (INTWDT) is generated.

In STOP (including warm up) or IDLE mode, the watchdog timer temporarily stops counting. Releasing STOP or IDLE mode automatically restarts the watchdog timer.

Example: Set watchdog timer detection time to 2^{21}/fc [s] and resets runaway detection.

```
                      LD    (WDTCR2), 4EH          ; Clears binary counter.
                      LD    (WDTCR1), 00001101B    ; WDTT←10, WDTOUT←1
                    ┌ LD    (WDTCR2), 4EH          ; Clears binary counter.
     Within WDT     │  :                             (Always clears immediately before and
     detection      │  :                             after WDTT change.)
     time           │  :
                    ├ LD    (WDTCR2), 4EH          ; Clears binary counter.
     Within WDT     │  :
     detection      │  :
     time           │  :
                    └ LD    (WDTCR2), 4EH          ; Clears binary counter.
                       :
```

Watchdog timer control register 1

	7	6	5	4	3	2	1	0	
WDTCR1 (00034_H)					WDT EN	WDTT		WDT OUT	(Initial value: **** 1001)

WDTEN	Watchdog timer enable/disable	0: Disable (It is necessary to write the disable code to WDTCR2.) 1: Enable	Write only
WDTT	Watchdog timer detection time [s]	00: 2^{25}/fc 01: 2^{23}/fc 10: 2^{21}/fc 11: 2^{19}/fc	
WDTOUT	Watchdog timer output select	0: Interrupt request 1: Reset output	

Note 1: WDTOUT cannot be set to "1" by program after clearing WDTOUT to "0".
*Note 2: fc ; High-frequency clock [Hz], *; Don't care*
Note 3: WDTCR1 is a write-only register and must not be used with any of read-modify-write instructions.
Note 4: The watchdog timer must be disabled or the counter must be cleared immediately before entering to the STOP mode. When the counter is cleared, the counter must be cleared again immediately after releasing the STOP mode.

Watchdog timer control register 2

	7	6	5	4	3	2	1	0	
WDTCR2 (00035_H)									(Initial value: **** ****)

WDTCR2	Writes watchdog timer control code.	$4E_H$: Watchdog timer binary counter clear. (clear code) $B1_H$: Watchdog timer disable. (disable code) others: Invalid	Write only

Note 1: The disable code is invalid unless written when WDTEN = 0.
*Note 2: * ; Don't care*
Note 3: The binary counter of the watchdog timer must not be cleared by the interrupt task.

Figure 2-13. Watchdog Timer Control Registers

(2) **Watchdog timer enable**

Setting 1 in WDTEN (bit 3 in WDTCR1) enables the watchdog timer. Resetting initializes WDTEN to 1; thus, releasing reset immediately starts the watchdog timer.

(3) **Watchdog timer disable**

Zero-clearing WDTEN (bit 3 in WDTCR1) and writing disable code ($B1_H$) in WDTCR2 disables the watchdog timer. Note that first writing disable code in WDTCR2 then zero-clearing WDTEN do not disable the watchdog timer. While the watchdog timer is disabled, the watchdog timer binary counter is cleared.

Example: Disable watchdog timer.

```
LDW     (WDTCR1), 0B101H        ;   WDTEN←0, WDTCR2←disable code
```

Table 2-2. Watchdog Timer Detection Time (Example: fc = 16 MHz)

WDTT	Watchdog Timer Detection Time [s]
00	2.097
01	524.288 m
10	131.072 m
11	32.768 m

2.4.3 Watchdog Timer Interrupt (INTWDT)

Watchdog timer interrupts are pseudo non-maskable. They are accepted regardless of the contents in the interrupt enable register. However, if another watchdog timer interrupt or software interrupt is being processed, the new watchdog timer interrupt must wait until processing of the current interrupt is complete (RETN instruction execution completes).

Set the stack pointer before setting watchdog timer output as the interrupt source in WDTOUT.

Example: Set Watchdog timer interrupt.

```
LD      SP, 0043FH              ;   Sets SP.
LD      (WDTCR1), 00001000B     ;   WDTOUT←0
```

2.4.4 Watchdog Timer Reset

"L" level is output from the $\overline{\text{RESET}}$ pin and, at the same time, internal hardware is reset. Reset timer is 8/fc to 24/fc [s] (0.5 to 1.5 μs at fc = 16 MHz). The $\overline{\text{RESET}}$ pin is sink open-drain I/O with a pull-up resistor.

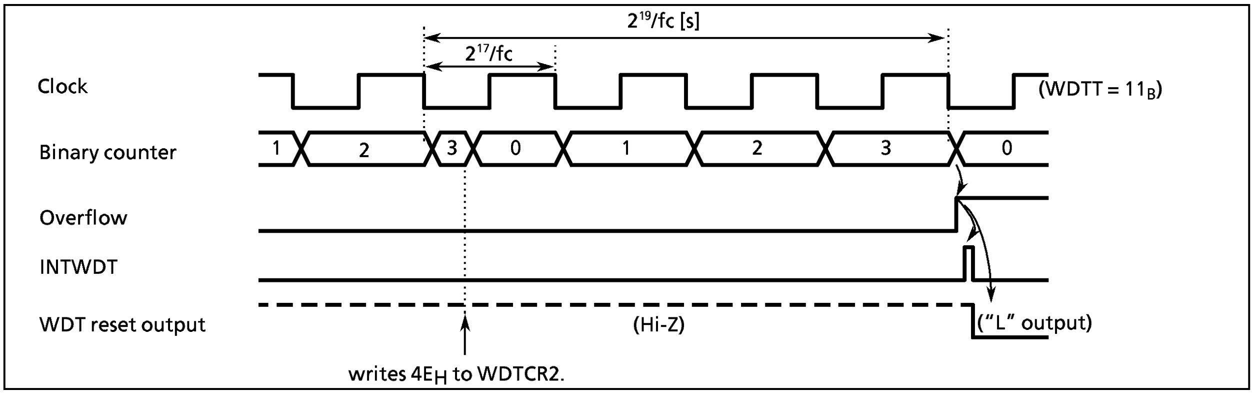


Figure 2-14. Watchdog Timer Interrupt / Reset

2.5 Divider Output ($\overline{\text{DVO}}$)

The timing generator divider outputs approx. 50 %-duty pulses, which can be used for driving a buzzer. The pulses are output to the P15 ($\overline{\text{DVO}}$) pin. First set the output latch for P15 to 1, then set it to output mode.

TBTCR (00036$_H$)	7	6	5	4	3	2	1	0	
	DVOEN	DVOCK		(DV7CK)	(TBTEN)	(TBTCK)			(Initial value: 0**0 0***)

DVOEN	Divider output enable/disable	0: Disable 1: Enable	R/W
DVOCK	Divider output ($\overline{\text{DVO}}$) frequency selection [Hz]	00: fc/2^{13} 01: fc/2^{12} 10: fc/2^{11} 11: fc/2^{10}	

*Note: fc ; High-frequency clock [Hz] * ; Don't care*

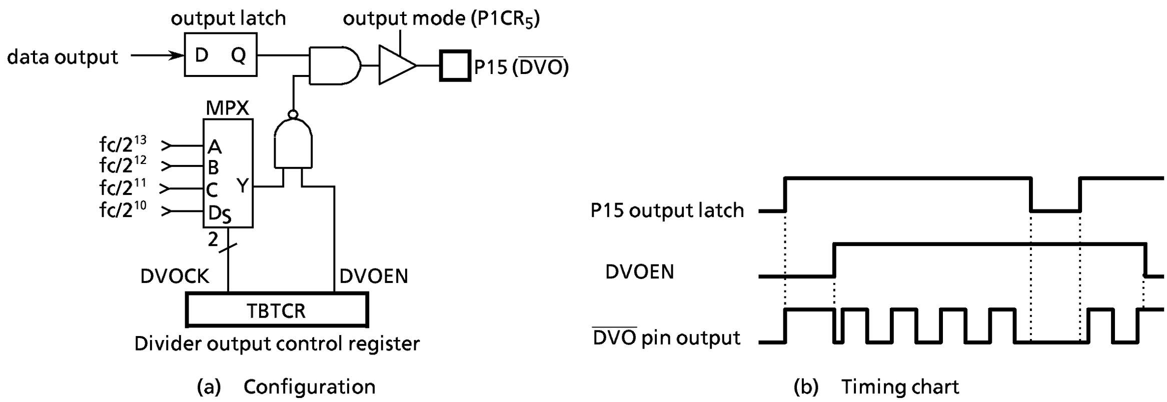
Figure 2-15. Divider Output Control Register

Example: Output 1.953 kHz pulses (when fc = 16 MHz).

```
SET    (P1). 5               ;  Sets P15 output latch to 1.
LD     (P1CR), 00100000B     ;  Sets P15 to output mode.
LD     (TBTCR), 10000000B    ;  DVOEN←1, DVOCK←00
```

Table 2-3. Divider Output Frequency (Example: fc = 16 MHz)

DVOCK	Divider output frequency [kHz]
00	1.953
01	3.906
10	7.812
11	15.625



(a) Configuration

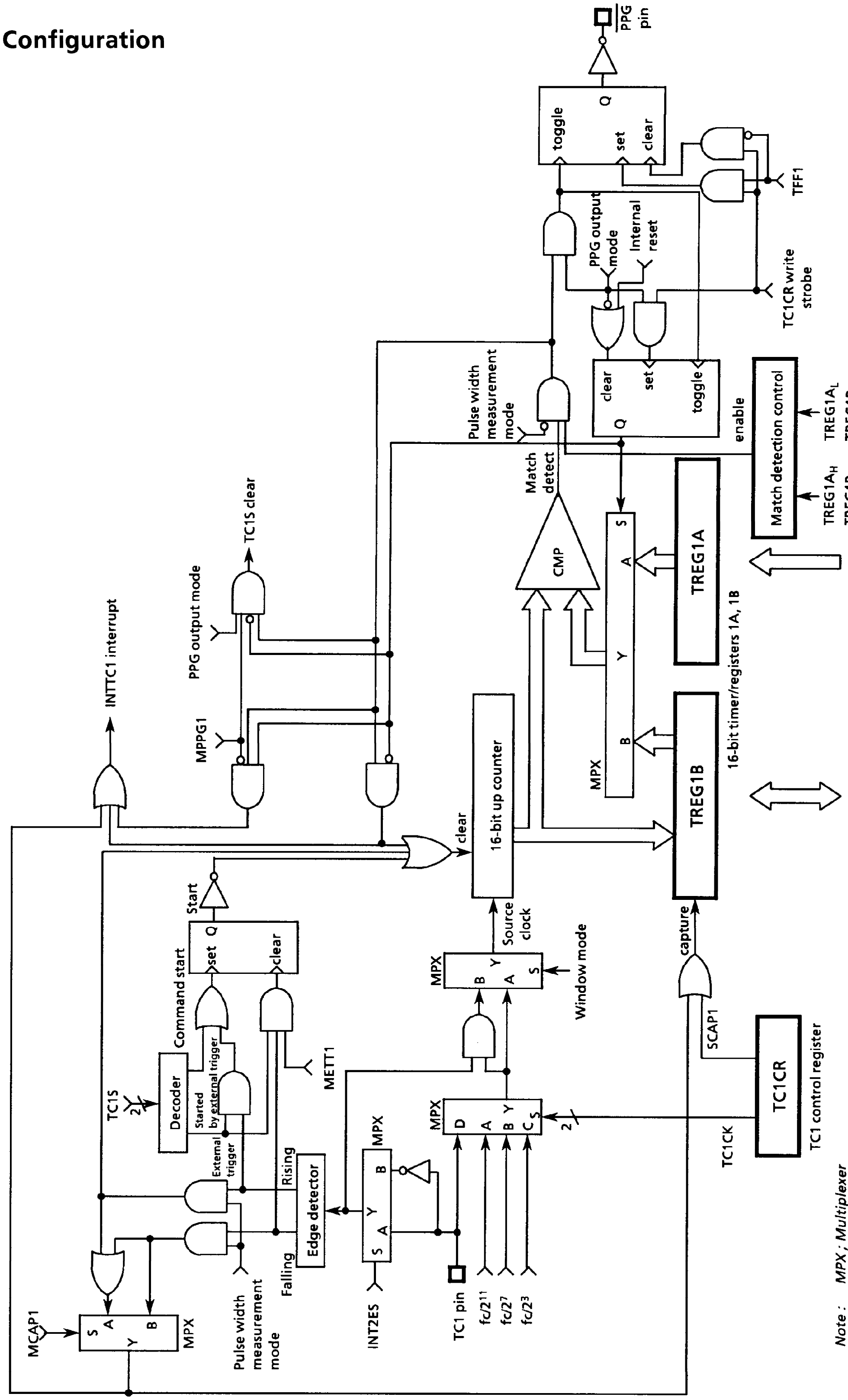
(b) Timing chart

Note: MPX ; Multiplexer

Figure 2-16. Divider Output

2.6 16-bit Timer/Counter 1 (TC1)

2.6.1 Configuration



Note : MPX ; Multiplexer
CMP ; Comparator

Figure 2-17. Timer/Counter 1 (TC1)

2.6.2 Control

Timer/counter 1 is controlled by the timer counter 1 control register (TC1CR) and two 16-bit timer registers (TREG1A and TREG1B).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREG1A (00010, 00011$_H$)	TREG1A$_H$ (0011$_H$)								TREG1A$_L$ (0010$_H$)							
									Write only							
TREG1B (00012, 00013$_H$)	TREG1B$_H$ (0013$_H$)								TREG1B$_L$ (0012$_H$)							

Read/Write (Writing is capable only when PPG output mode)

	7	6	5	4	3	2	1	0
TC1CR (00014$_H$)	TFF1	SCAP1 MCAP1 METT1 MPPG1	TC1S		TC1CK		TC1M	

(Initial value: 0000 0000)

TC1M	TC1 operating mode select	00: Timer/external trigger timer/event counter mode 01: Window mode 10: Pulse width measurement mode 11: PPG (Programmable pulse generate) output mode	Write only
TC1CK	TC1 source clock select [Hz]	00: Internal clock fc/2^{11} [Hz] 01: 〃 fc/2^7 10: 〃 fc/2^3 11: External clock (TC1 pin input)	
TC1S	TC1 start control	00: Stop and clears counter. 01: Command start 10: Reserved 11: Started by external trigger	
SCAP1	Software capture control	0: — 1: Software capture trigger	
MCAP1	Pulse width measurement mode control	0: Double edge capture 1: Single edge capture	
METT1	External trigger timer mode control	0: Trigger start 1: Trigger start & stop	
MPPG1	PPG output control	0: Pulse 1: Single	
TFF1	Timer F/F1 control for PPG output mode	0: Clear 1: Set	

Note 1: fc ; High-frequency clock [Hz]

Note 2: Writing to the lower byte of the timer registers (TREG1A$_L$, TREG1B$_L$), the comparison is inhibited until the upper byte (TREG1A$_H$, TREG1B$_H$) is written. Only the lower byte of the timer registers can not be changed. After writing to the upper byte, any match during 1 machine cycle (instruction execution cycle) is ignored.

Note 3: Set the mode, source clock, edge (including INT2ES), PPG control and timer F/F control when TC1 stops (TC1S = 00).

Note 4: Software capture can be used in only timer and event counter modes. SCAP1 is automatically cleared to "0" after capturing.

Note 5: Values to be loaded to timer registers must satisfy the following condition.
TREG1A>TREG1B>0 (PPG output mode), TREG1A>0 (others)

Note 6: Always write "0" to TFF1 except PPG output mode.

Note 7: TREG1B cannot be written after setting to PPG output mode.

Note 8: TC1R and TREG1A are write-only registers and must not be used with any of the read-modify-write instructions such as SET, CLR, etc.

Figure 2-18. Timer Registers and TC1 Control Register

2.6.3 Function

Timer/counter 1 supports six operating modes: timer, external trigger timer, event counter, window, pulse width measurement, and programmable pulse generator output.

(1) **Timer mode**

TC1 counts up using the internal clock. A match between the counter value and the value set in timer register 1A (TREG1A) generates a timer/counter 1 interrupt (INTTC1) and clears the counter. After the counter is cleared, it continues counting. Setting SCAP1 (bit 6 in TC1CR) to 1 loads the data in the up-counter to timer register 1B (TREG1B) (software capture). After capture, SCAP1 is automatically zero-cleared.

Table 2-4. Source Clock (internal clock) for Timer / Counter 1

Source clock	Resolution	Maximum setting time
NORMAL or IDLE mode	When fc = 16 MHz	
$fc/2^3$ [Hz]	0.5 μs	32.75 ms
$fc/2^7$	8 μs	0.5 s
$fc/2^{11}$	128 μs	8.4 s

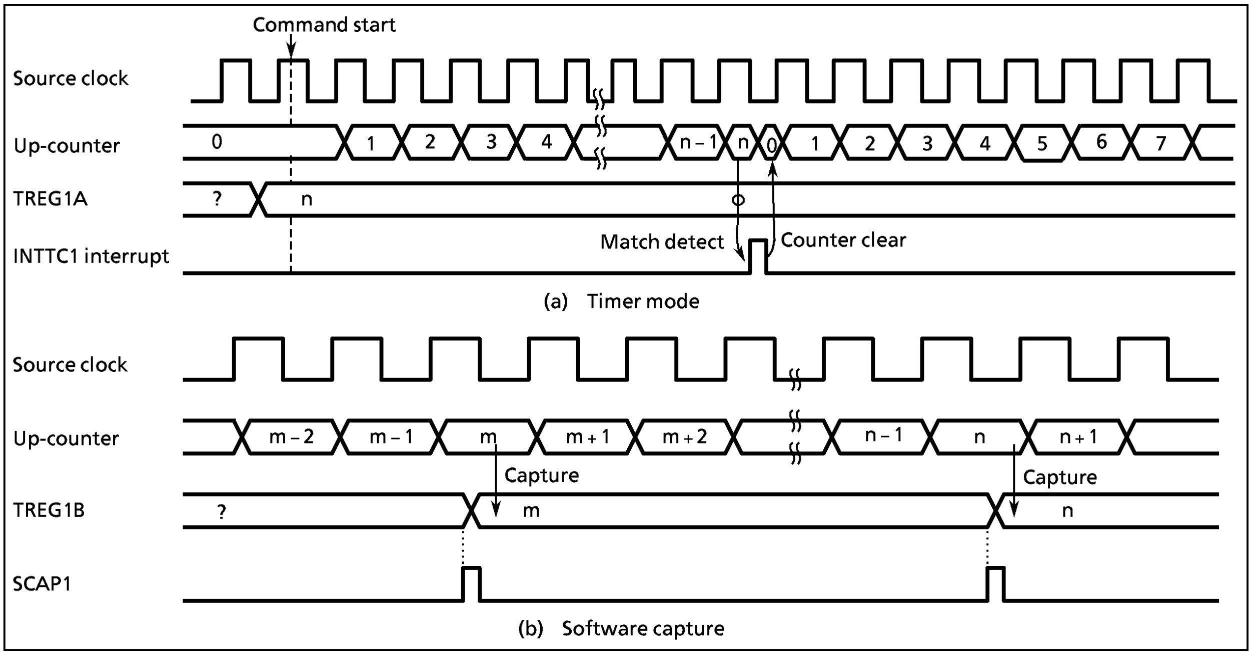
Example 1: Set to timer mode with source clock $fs/2^{11}$ [Hz] and generate an interrupt after 1s (when fc = 16MHz).

```
LDW      (TREG1A), 1E84H       ;   Sets timer register.  (1s ÷ 2^11/fc = 1E84H)
SET      (EIRL). EF4           ;   Enables INTTC1.
EI
LD       (TC1CR), 00010000B    ;   Starts TC1
```

Note: TC1R is a write-only register and must not be used with [SET (TC1CR). 4] instruction.

Example 2: Software capture

```
LD       (TC1CR), 01010000B    ;   SCAP1←1
LD       WA, (TREG1B)          ;   Reads capture value
```



(a) Timer mode

(b) Software capture

Figure 2-19. Timer Mode Timing Chart

(2) External trigger timer mode

TC1 starts counting at the input edge of the TC1 pin (rising or falling edge selection also selects the INT2 pin edge). (source clock: internal clock) A match between the counter value and the value set in TREG1A generates INTTC1, clears and stops the counter. At the TC1 pin input edge, counting resumes. When METT1 (bit 6 in TC1CR) is set to 1, inputting the inverse of the trigger edge used for starting the counter clears, then stops the counter. In this mode, an interrupt of fixed pulse width can be generated When METT1 is set to 0, input of the inverse edge is ignored. TC1 pin input before match detect is also ignored.

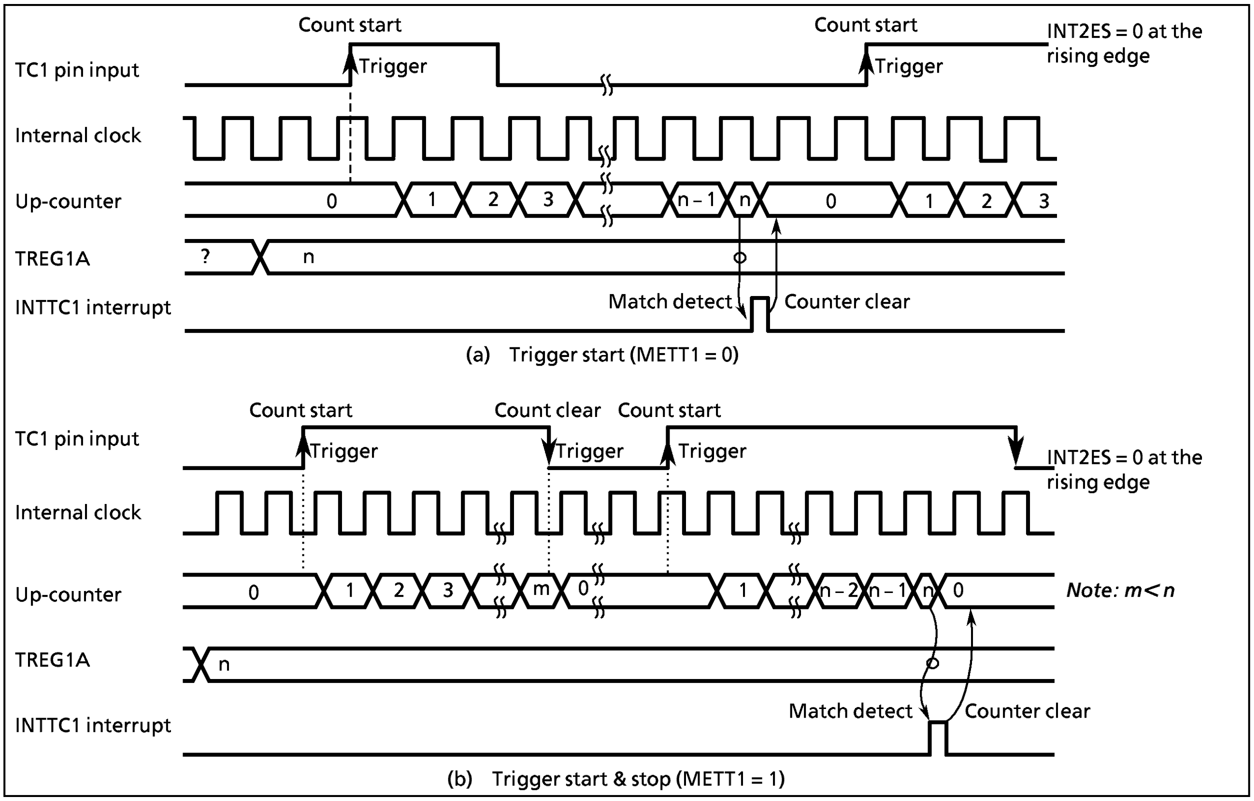
Since the TC1 pin also has a noise rejection circuit the same as the INT2 pin, in NORMAL or IDLE mode, pulses with a width of 7 / fc [s] or less are rejected as noise. For reliable edge detection, pulses must have a width of 24 / fc [s] or more. In SLEEP or SLOW mode, the noise rejection circuit is turned off; pulses must have a width of 1 machine cycle or more.

Example 1: Generate an interrupt 100 μs after TC1 pin input rising edge. (when fc = 16 MHz)

```
LD      (EINTCR), 00000000B  ;  INT2ES←0 (rising edge)
LDW     (TREG1A), 00C8H      ;  100 μs ÷ 2^3 / fc = C8H
SET     (EIRE). EF16         ;  Enables INTTC1.
EI
LD      (TC1CR), 00111000B   ;  Starts TC1 by external trigger.  METT = 0
```

Example 2: Generate an interrupt when a pulse with a width of 4 ms or more is input to the TC1 pin at "L" level. (when fc = 16 MHz)

```
LD      (EINTCR), 00000100B  ;  INT2ES←1 ("L" level)
LDW     (TREG1A), 01F4H      ;  4 ms ÷ 2^7 / fc = 1F4H
SET     (EIRE). EF16         ;  Enables INTTC1.
EI
LD      (TC1CR), 01110100B   ;  Starts TC1 by external trigger.  METT = 1
```



(a) Trigger start (METT1 = 0)

(b) Trigger start & stop (METT1 = 1)

Figure 2-20. External Trigger Timer Mode Timing Chart

(3) **Event counter mode**

TC1 starts counting at the input edge of the TC1 pin (rising or falling edge selection also selects the INT2 pin edge). A match between the counter value and the value set in TREG1A generates INTTC1 and clears the counter. After the counter is cleared, it continues counting at every input edge of the TC1 pin. The applicable maximum frequency is fc/2^4 [Hz] (in NORMAL or IDLE mode).
Setting SCAP1 to 1 loads the data in the up-counter to TREG1B (software capture).

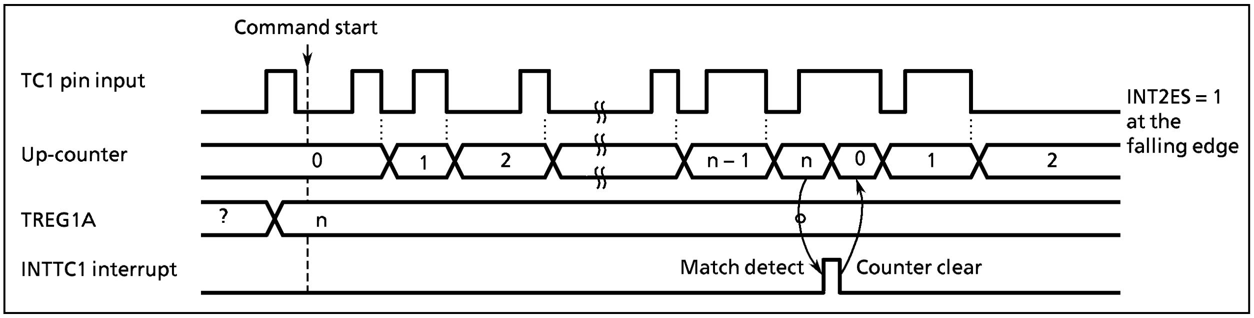
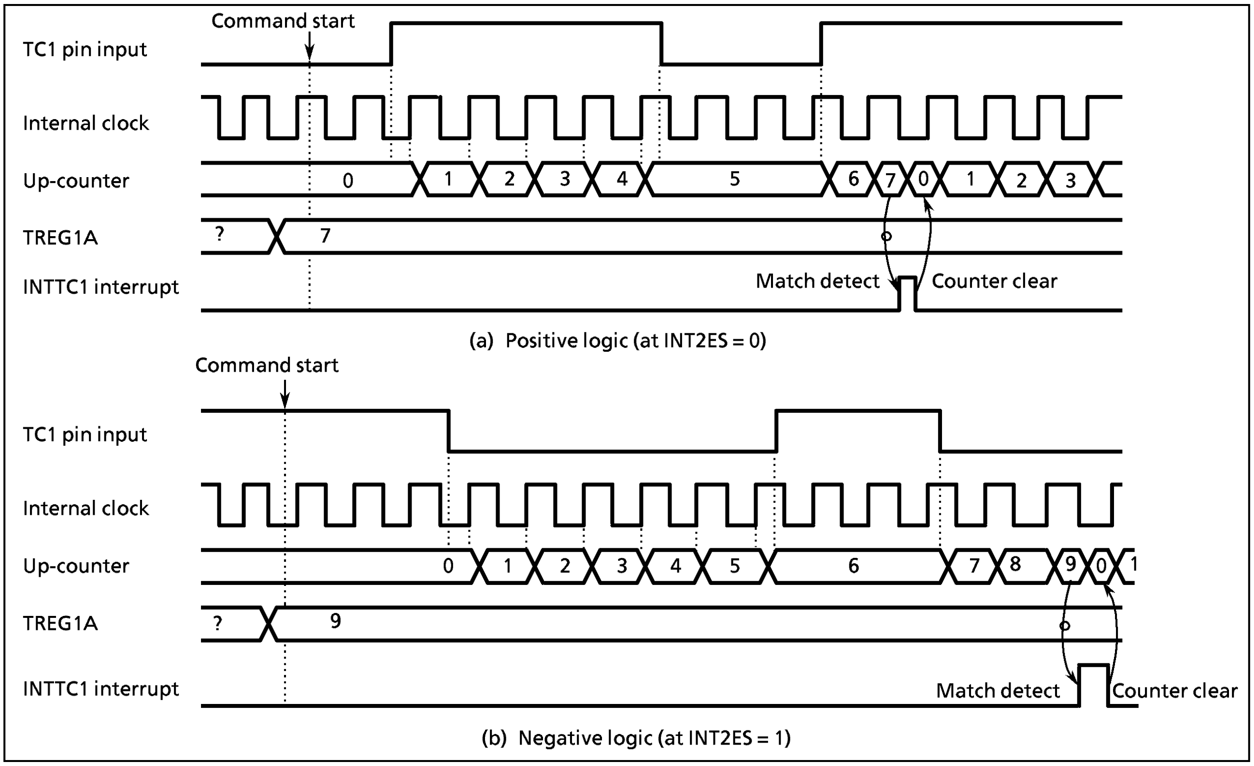


Figure 2-21. Event Counter Mode Timing Chart

(4) **Window mode**

TC1 starts counting at the rising edge of the AND pulse of the input to the TC1 pin (window pulse) and the internal clock. A match between the counter value and the value set in TREG1A generates INTTC1 and clears the counter. Positive or negative AND is selectable for TC1 pin input (rising or falling edge selection also selects the INT2 pin edge).
The applicable maximum frequency must be a frequency whose count value can be analyzed by program. That is, the frequency must be much slower than the set internal clock.



(a) Positive logic (at INT2ES = 0)

(b) Negative logic (at INT2ES = 1)

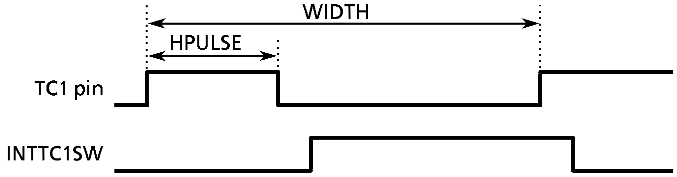
Figure 2-22. Window Mode Timing Chart

(5) **Pulse width measurement mode**

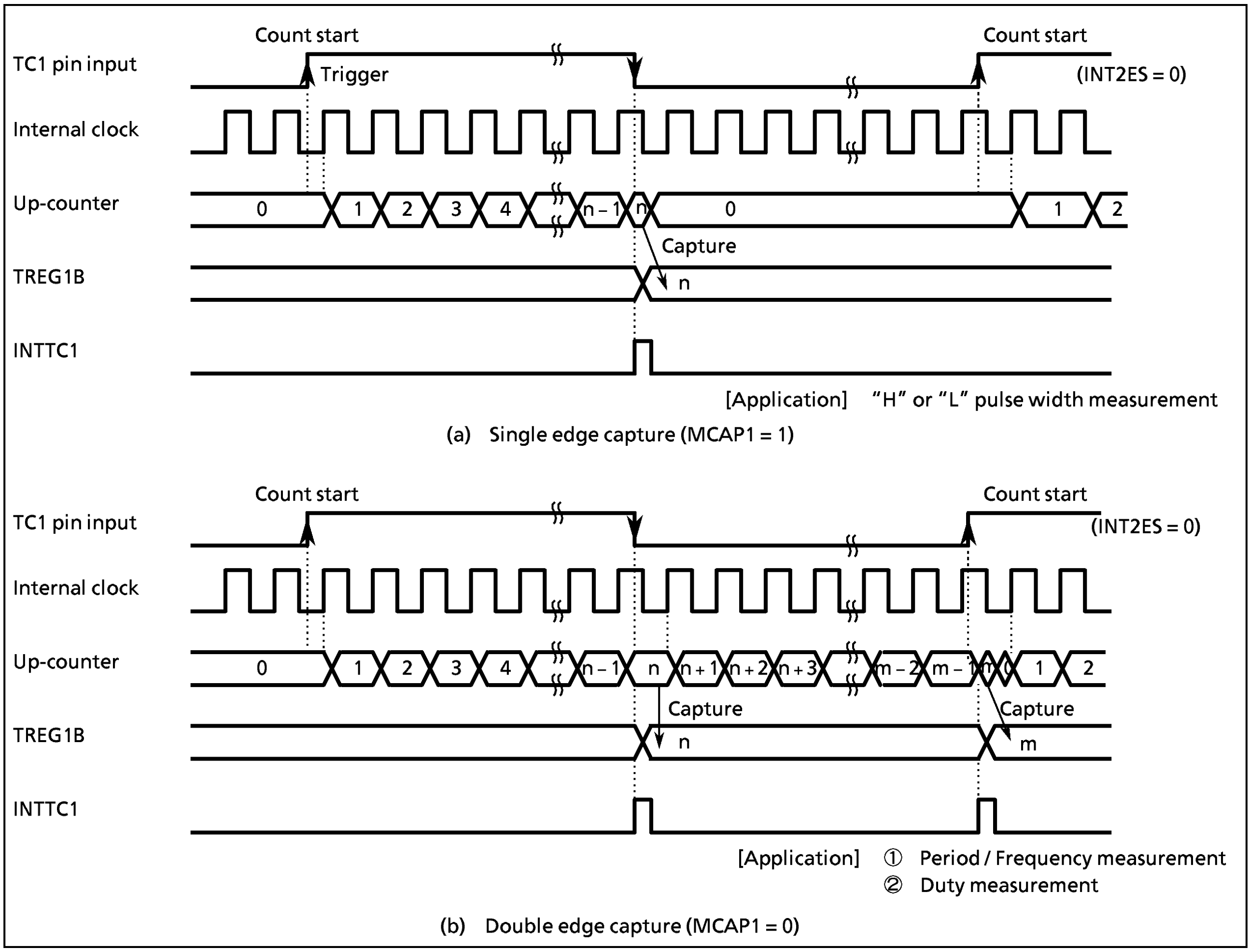
TC1 starts counting triggered by the rising (falling) edge of input to the TC1 pin (set start by external trigger in TC1CR). The source clock is the internal clock. At the next falling (rising) edge, the counter value is loaded to TREG1B and an interrupt is generated. If one-edge capture is set, the counter is cleared. If both-edge capture is set, the counter continues counting; at the next rising (falling) edge, the counter value is loaded to TREG1B. If a capture value at a falling (rising) edge is required, data in TREG1B must be read before a rising (falling) edge is detercted. Select rising or falling edge using INT2ES; select one- or both-edge capture using MCAP1 (bit 6 in TC1CR).

Example: Measure duty (when resolution fc / 2^7 [Hz]).

```
           CLR   (INTTC1SW). 0            ;  Initializes INTTC1 service switch.
           LD    (EINTCR), 00000000B      ;  Sets INT2ES to rising edge.
           LD    (TC1CR), 00000110B       ;  Sets TC1 mode and source clock.
           SET   (EIRE). EF16             ;  Enables INTTC1.
           EI
           LD    (TC1CR), 00110110B       ;  Starts TC1 by external trigger with
                                             MCAP1 = 0.
            ⋮
PINTTC1:   CPL   (INTTC1SW). 0            ;  Inverts / tests INTTC1 service switch.
           JRS   F, SINTTC1
           LD    (HPULSE), (TREG1BL)      ;  Reads TREG1B ("H" level pulse width).
           LD    (HPULSE + 1), (TREG1BH)
           RETI
SINTTC1:   LD    (WIDTH), (TREG1BL)       ;  Reads TREG1B.  (cycle)
           LD    (WIDTH + 1), (TREG1BH)
            ⋮                             ;  Calculates duty.
           RETI
            ⋮
VINTTC1:   DL    PINTTC1
```



In the width measurement mode, when $fc/2^3$ is selected as the source clock, the least significant bit (bit 0) of the read counter value (TREG1B) is set to 0. The other source clocks read the counter values according to counts.



[Application] "H" or "L" pulse width measurement

(a) Single edge capture (MCAP1 = 1)

[Application] ① Period / Frequency measurement
② Duty measurement

(b) Double edge capture (MCAP1 = 0)

Figure 2-23. Pulse Width Measurement Mode Timing Chart

(6) **Programmable Pulse Generate (PPG) output mode**

TC1 starts counting at a rising (falling) edge input to the TC1 pin (rising or falling edge selection also selects the INT2 pin edge) or by a command. The source clock is the internal clock. A match between the counter value and the value set in TREG1B inverts timer F/F1. Continuous output (MPPG1 = 0) generates INTTC1. The next match between the counter value and the value set in TREG1A inverts timer F/F1 again and clears TC1. At the same time, INTTC1 is generated. Timer F/F1 output is inverted then connected to the P14 ($\overline{\text{PPG}}$) pin. For PPG output, set the P14 output latch to 1 for output mode. Reset zero-clears timer F/F1. Since the timer F/F1 value can be set in TFF1 (bit 7 in TC1CR), either positive or negative AND pulses can be output. Set to PPG output mode first; otherwise, data cannot be written to TREG1B.

Example: Output "H" level 800-µs pulse and "L" level 200 µs pulse (when fc = 8 MHz).

```
SET     (P1). 4                 ;  Sets P14 output latch to 1.
LD      (P1CR), 00010000B       ;  Sets P14 to output mode.
LD      (TC1CR), 10001011B      ;  Sets to PPG output mode.
LDW     (TREG1A), 03E8H         ;  Sets cycle. (1 ms ÷ 1 µs = 03E8H)
LDW     (TREG1B), 00C8H         ;  Sets "L" level pulse width.
                                   (200 µs ÷ 1 µs = 00C8H)
LD      (TC1CR), 10011011B      ;  Starts TC1.
```

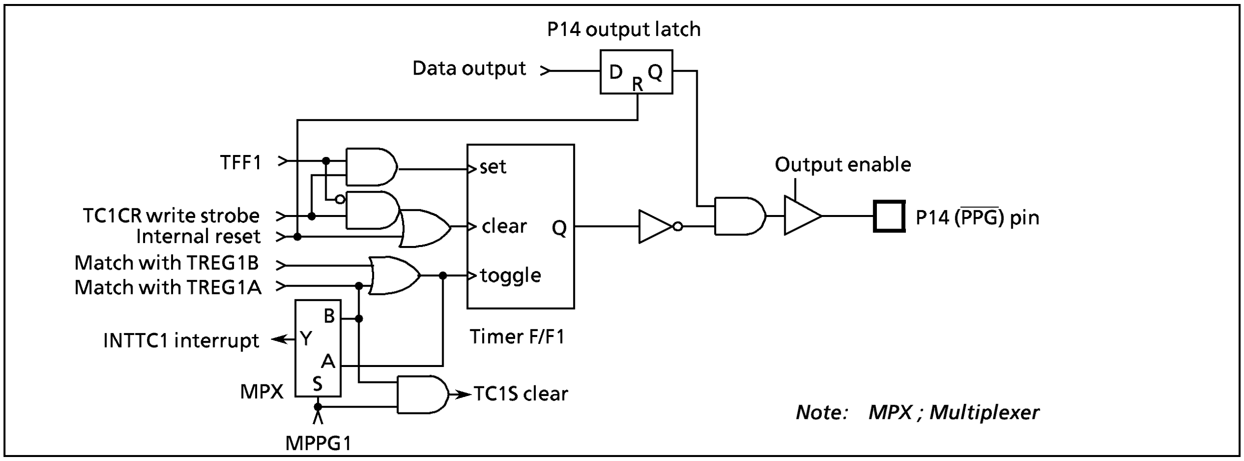
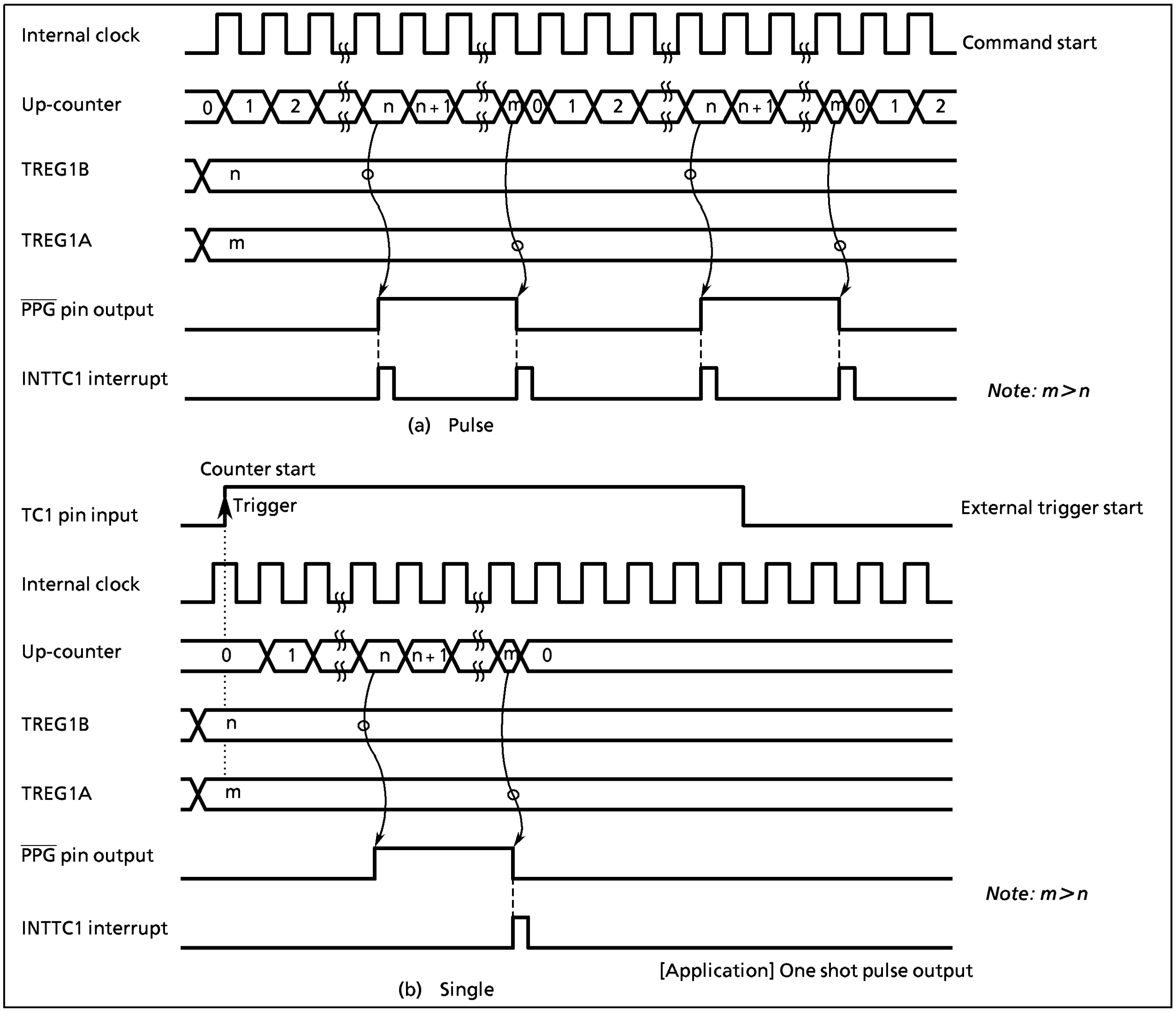


Figure 2-24. $\overline{\text{PPG}}$ Output



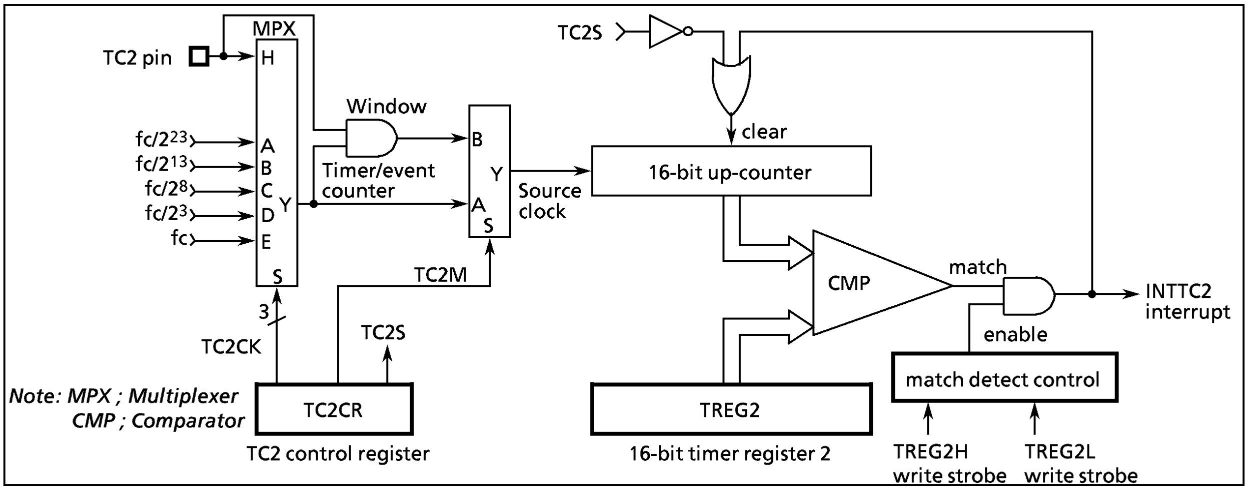
(a) Pulse

(b) Single

Figure 2-25. PPG Output Mode Timing Chart

2.7 16-bit Timer/Counter 2 (TC2)

2.7.1 Configuration



Note: MPX ; Multiplexer
CMP ; Comparator

Figure 2-26. Timer/Counter 2 (TC2)

2.7.2 Control

Timer/counter 2 is controlled by timer/counter 2 control register (TC2CR) and 16-bit timer register 2 (TREG2).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREG2 (00016, 00017$_H$)	TREG2$_H$ (0017$_H$)								TREG2$_L$ (0016$_H$)							

Write only

	7	6	5	4	3	2	1	0
TC2CR (00015$_H$)			TC2S	TC2CK				TC2M

(Initial value: **00 00*0)

TC2M	TC2 operating mode select	0: Timer/event counter mode 1: Window mode 0: Stops and counter clear 1: Start	Write only
TC2CK	TC2 source clock select [Hz]	000 : Internal clock fc/2^{23} [Hz] 001 : 〃 fc/2^{13} 010 : 〃 fc/2^8 011 : 〃 fc/2^3 100 : reserved 101 : reserved 110 : reserved 111 : External clock (TC2 pin input)	
TC2S	TC2 start control	0: Stops and counter clear 1: Start	

*Note 1: fc ; High-frequency clock [Hz] * ; Don't care*

Note 2: Writing to the lower byte of timer register 2 (TREG2$_L$), the comparison is inhibited until the upper byte (TREG2$_H$) is written. After writing to the upper byte, any match during 1 machine cycle (instructon execution cycle) is ignored.

Note 3: Set the mode and source clock when the TC2 stops (TC2S = 0).

Note 4: Values to be loaded to the timer register must satisfy the following condition.
TREG2 > 0 (TREG2$_{15\ to\ 11}$ > 0 at warm-up)

Note 5: TC2CR and TREG2 are write-only registers and must not be used with any of the read-modify-write instructions such as SET, CLR, etc.

Figure 2-27. Timer Register 2 and TC2 Control Register

2.7.3 Function

Timer/counter 2 supports three operating modes: timer, event counter, and window. When switching modes from SLOW to NORMAL, use timer/counter 2 in timer mode.

(1) **Timer mode**

TC2 counts up using the internal clock. A match between the counter value and the value set in timer register 2 (TREG2) generates INTTC2 and clears the counter. After the counter is cleared, it continues counting.

Table 2-5. Soure Clock internal clock for Timer/Counter 2

Source clock	Resolution	Maximum setting time
NORMAL, IDLE mode	When fc = 16 MHz	
$fc/2^{23}$ [Hz] $fc/2^{13}$ $fc/2^{8}$ $fc/2^{3}$	524 ms 512 μs 16 μs 0.5 μs	9.5 ms 33.6 min 1.05 ms 32.75 ms

Example: Set to timer mode with source clock = $fc/2^3$ [Hz] and generate interrupts every 25 ms (fc = 16 MHz)

```
LDW    (TREG2), C350H       ;  Sets TREG2. (25 ms ÷ 2^3/fc = C350_H)
SET    (EIRE). EF17         ;  Enables INTTC2.
EI
LD     (TC2CR), 00101100B   ;  Starts TC2.
```

(2) **Event counter mode**

TC2 counts up at the rising edge of input to the TC2 pin. A match between the counter value and the value set in TREG2 generates INTTC2 and clears the counter. The applicable maximum frequency for the TC2 pin is $fc/2^4$ [Hz] (in NORMAL or IDLE mode). For both "H" and "L" levels, the pulse width must be 2 machine cycles or more.

Example: Set to event counter mode and generate INTTC2 after 640 counts.

```
LDW    (TREG2), 640         ;  Sets TREG2.
SET    (EIRE). EF17         ;  Enables INTTC2.
EI
LD     (TC2CR), 00111100B   ;  Starts TC2.
```

(3) **Window mode**

TC2 counts up using the internal clock while input to the TC2 external pin (window pulse) is at "H" level. A match between the counter value and the value set in TREG2 generates INTTC2 and clears the counter. The applicable maximum frequency for the TC2 pin must be sufficiently slower than the set internal clock.

Example: When a "H" level pulse of 120 ms or longer is input, generate an interrupt (when fc = 16 MHz).

```
LDW    (TREG2), 00EAH       ;  Sets TREG2. (120 ms ÷ 2^13/fc = 00EA_H)
SET    (EIRE). EF17         ;  Enables INTTC2.
EI
LD     (TC2CR), 00100101B   ;  Starts TC2.
```

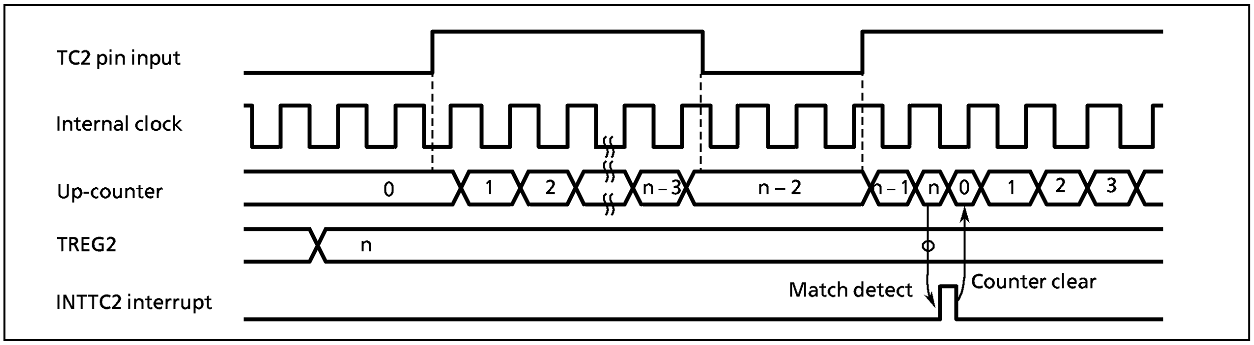
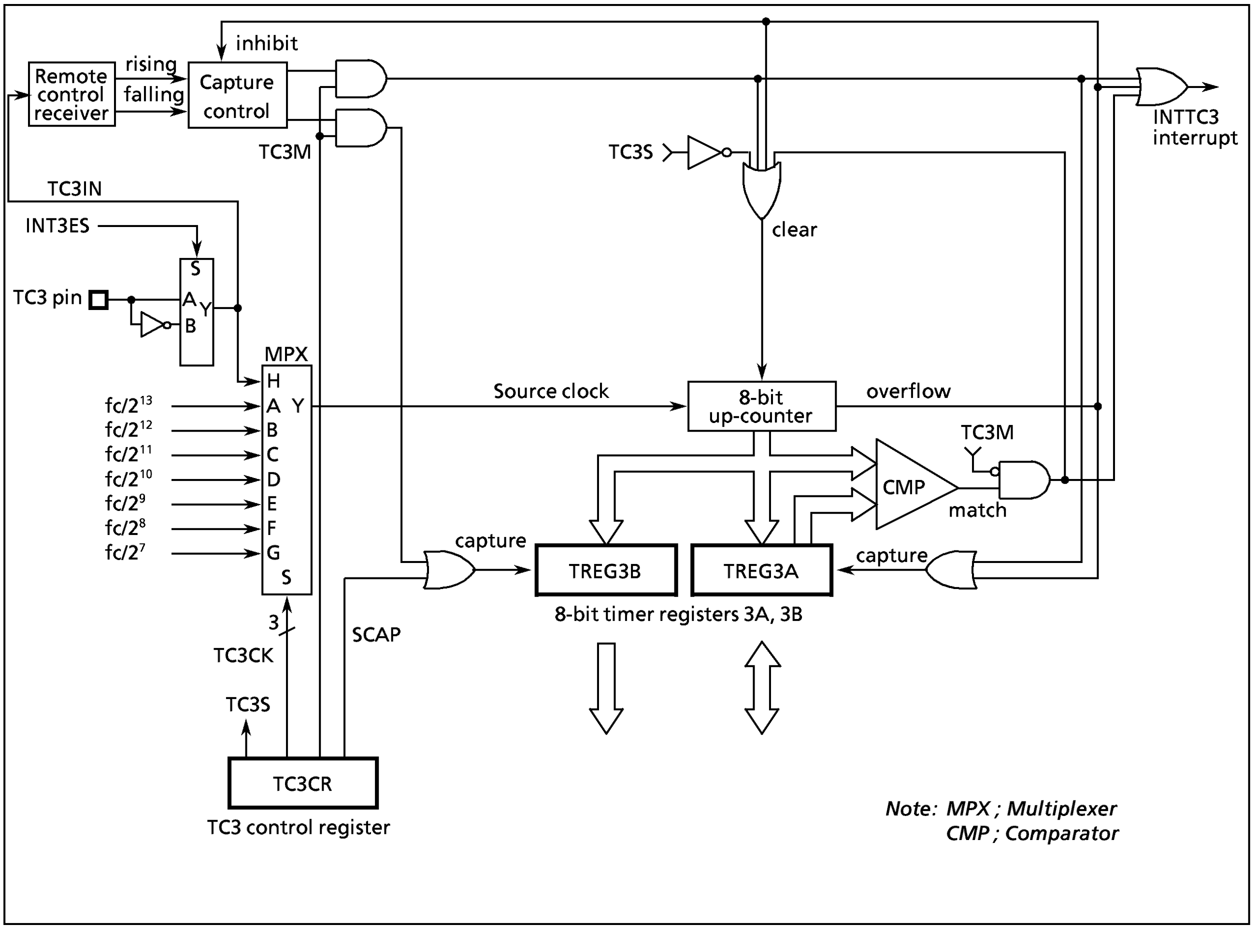


Figure 2-28. Window Mode Timing Chart

2.8 8-bit Timer/Counter 3 (TC3)

2.8.1 Configuration



Note: MPX ; Multiplexer
CMP ; Comparator

Figure 2-29. Timer/Counter 3 (TC3)

2.8.2 Control

Timer/counter 3 is controlled by timer/counter 3 control register (TC3CR) and tow 8-bit timer registers (TREG3A and TREG3B).

	7	6	5	4	3	2	1	0	
TREG3A (00018_H)									Read/Write
TREG3B (00019_H)									Read only

	7	6	5	4	3	2	1	0	
TC3CR ($0001A_H$)		SCAP		TC3S	TC3CK			TC3M	(Initial value: *0*0 0000)

TC3M	TC3 operation mode set	0: Timer/event counter 1: Capture	Write only
TC3CK	TC3 source clock select [Hz]	000: $fc/2^{13}$ [Hz] 001: $fc/2^{12}$ 010: $fc/2^{11}$ 011: $fc/2^{10}$ 100: $fc/2^{9}$ 101: $fc/2^{8}$ 110: $fc/2^{7}$ 111: External clock (TC3 pin input)	
TC3S	TC3 start select	0: Stops & clear 1: Starts	
SCAP	Software capture control	0: – 1: Software capture trigger	

*Note 1: fc ; High-frequency clock [Hz], * ; Don't care*

Note 2: Set the mode, the source clock and the edge selection (INT3ES) when the TC3 stops (TC3S = 0).

Note 3: Values to be loaded into timer register 3A must satisfy the following condition.
TREG3A > 0 (in the timer and event counter mode)

Note 4: Software capture can be used only in the timer and event counter mode. SCAP is automatically cleared to "0" after capturing.

Note 5: TC3CR is a write-only register and must not be used with any of read -modify-write instructions such as SET, CLR, etc.

Figure 2-30. Timer Register 3 and TC3 Control Register

2.8.3 Function

Timer/counter 3 supports three operating modes: timer, event counter, and window. In capture mode, noise rejection time can be set for input to TC3 input using the remote receive control register.

(1) **Timer mode**

TC3 counts up using the internal clock. A match between the counter value and the value set in timer register 3A (TREG3A) generates INTTC3 and clears the counter. Setting SCAP (bit 6 in TC3CR) to 1 loads the data in the up-counter to timer register 3B (TREG3B). SCAP is automatically zero-cleared.

Table 2-6. Source Clock (internal clock) for Timer / Counter 3
(Example: at fc = 16 MHz)

TC3CK	NORMAL, IDLE mode	
	Resolution [μs]	Maximum setting time [ms]
000	512	131.1
001	256	65.3
010	128	32.6
011	64	16.3
100	32	8.2
101	16	4.1
110	8	2.0

(2) **Event counter mode**

TC3 starts counting at the input edge of the TC3 pin (rising or falling edge selection also selects the INT3 pin edge). A match between the counter value and the value set in TREG3A generates INTTC3 and clears the counter.

The applicable maximum frequency is as shown in Table 2-7. For both "H" and "L" levels, the pulse width must be 2 machine cycles or more.

Setting SCAP to 1 loads the data in the up-counter to TREG3B (software capture). After software capture, SCAP is automatically zero-cleared.

Example: Input a pulse of 50 Hz to the TC3 pin and generate an interrupt every 0.5 s.

```
LD      (TREG3A), 19H        ;   0.5 s ÷ 1/50 = 25 = 19H
LD      (TC3CR), 00011110B   ;   Starts TC3.
```

Table 2-7. Source Clock (external clock) for Timer/Counter 3

Maximum applied frequency [Hz]
NORMAL, IDLE mode
$fc/2^4$

(3) **Capture mode**

Capture mode is used to measure pulse width, cycle, and duty of input to the TC3 pin. It is used for decoding remote control signals and identifying 50/60 Hz AC. the counter free-runs using the internal clock. The counter value is loaded to TREG3A at the rising (falling) edge of input to the TC3 pin (rising of falling edge selection also selects the INT3 pin edge), the counter is cleared, then an interrupt is generated. The counter value is also loaded to TREG3B at the falling (rising) edge of input to the TC3 pin. In this case, the counter continues counting, the counter value is loaded to TREG3A, the counter is cleared, and an interrupt is generated. If the counter overflows before edge detection, FF_H is set in TREG3A and INTTC3 is generated simultaneously. Reading TREG3A with the interrupt processing determines overflow or no overflow. After the interrupt is generated (capture to TREG3A or overflow detect), capture or overflow detect is disabled until TREG3A is read. The counter continues counting. Reading TREG3A resumes capture or overflow detect ; TREG3B is usually read first.

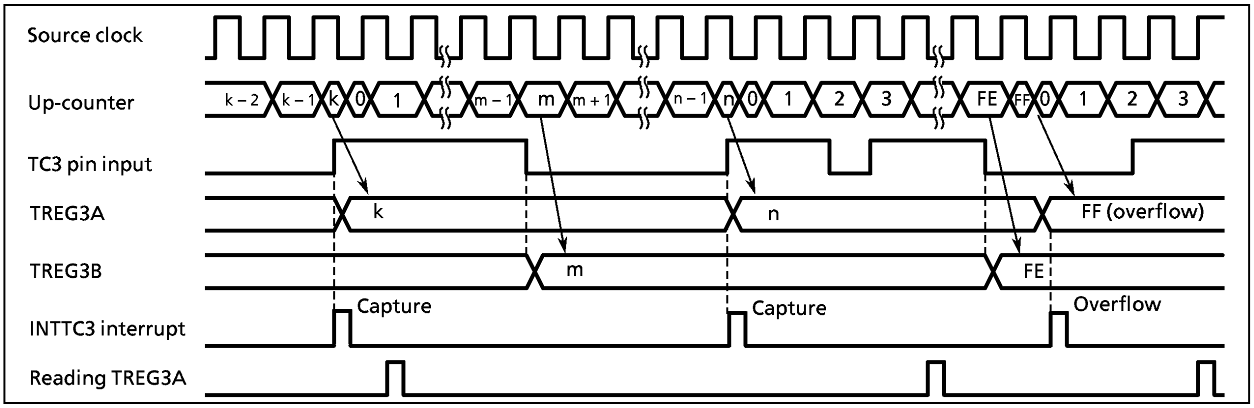


Figure 2-31. Capture Mode Timing Chart (at INT3ES = 0)

The edge of input to the TC3 pin is detected using the remote control receive circuit shared by the noise rejection circuit. The remote control receive circuit is controlled by the remote control register (RCCR). The remote control status register (RCSR) is also used as the remote control register; it monitors polarity selection and noise rejection circuit status.

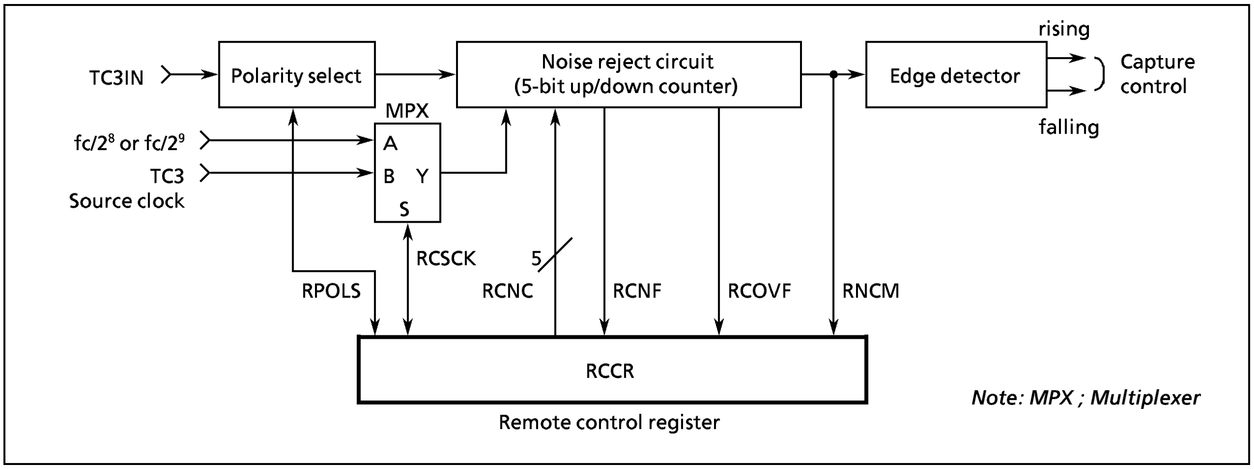


Figure 2-32. Remote Control Receive Circuit

RCCR
($0001F_H$)

	RPOLS	RCSCK	RCNC				

(Initial value: *001 1111)

RPOLS	Remote control signal polarity select	0: Positive 1: Negative	R/W
RCSCK	Noise reject circuit Source clock select	0: 2^8/fc 1: TC3CK	
RCNC	Noise reject time select 02_H ≦ RCNC ≦ $1F_H$	(source clock) × (RCNC–1) [s]	Write only

Note 1: Set RPOLS and RCSCK when the timer / counter stops (TC3S = 0)
Note 2: Source clock of timer / counter 3
*Note 3: fc ; High-frequency clock [Hz], * ; Don't care*
Note 4: RCCR is a write-only register and must no be used with any of read-modify-write instructions such as SET, CLR. etc.

RCSR
($0001F_H$)

RCNF	RPOLS	RCSCK	RCOVF	RNCM			

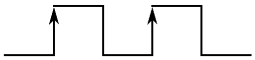
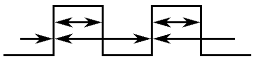
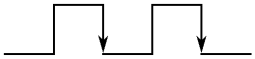
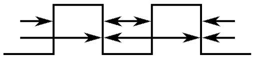
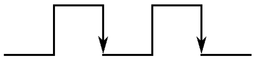
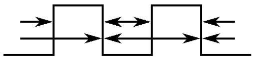
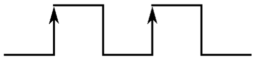
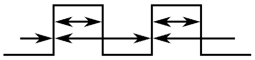
(Initial value: 0000 0***)

<table>
<tr><td>RCNF</td><td>Remote control signal monitor after noise rejecter</td><td colspan="3">0: without noise
1: with noise</td><td>Read only</td></tr>
<tr><td>RPOLS</td><td>Remote control signal polarity select</td><td colspan="3">0: Positive
1: Negative</td><td rowspan="2">R/W</td></tr>
<tr><td>RCSCK</td><td>Noise reject circuit
Source clock select</td><td colspan="3">0: 2^8/fc
1: TC3CK</td></tr>
<tr><td>RCOVF</td><td>Noise reject circuit Overflow flag</td><td colspan="3">0: Determines as signal, because noise rejection time RCNC is overwritten.
1: Other than above</td><td rowspan="4">Read only</td></tr>
<tr><td rowspan="3">RNCM</td><td rowspan="3">Remote control signal monitor after noise rejecter</td><td></td><td>INT3ES = 0</td><td>INT3ES = 1</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>

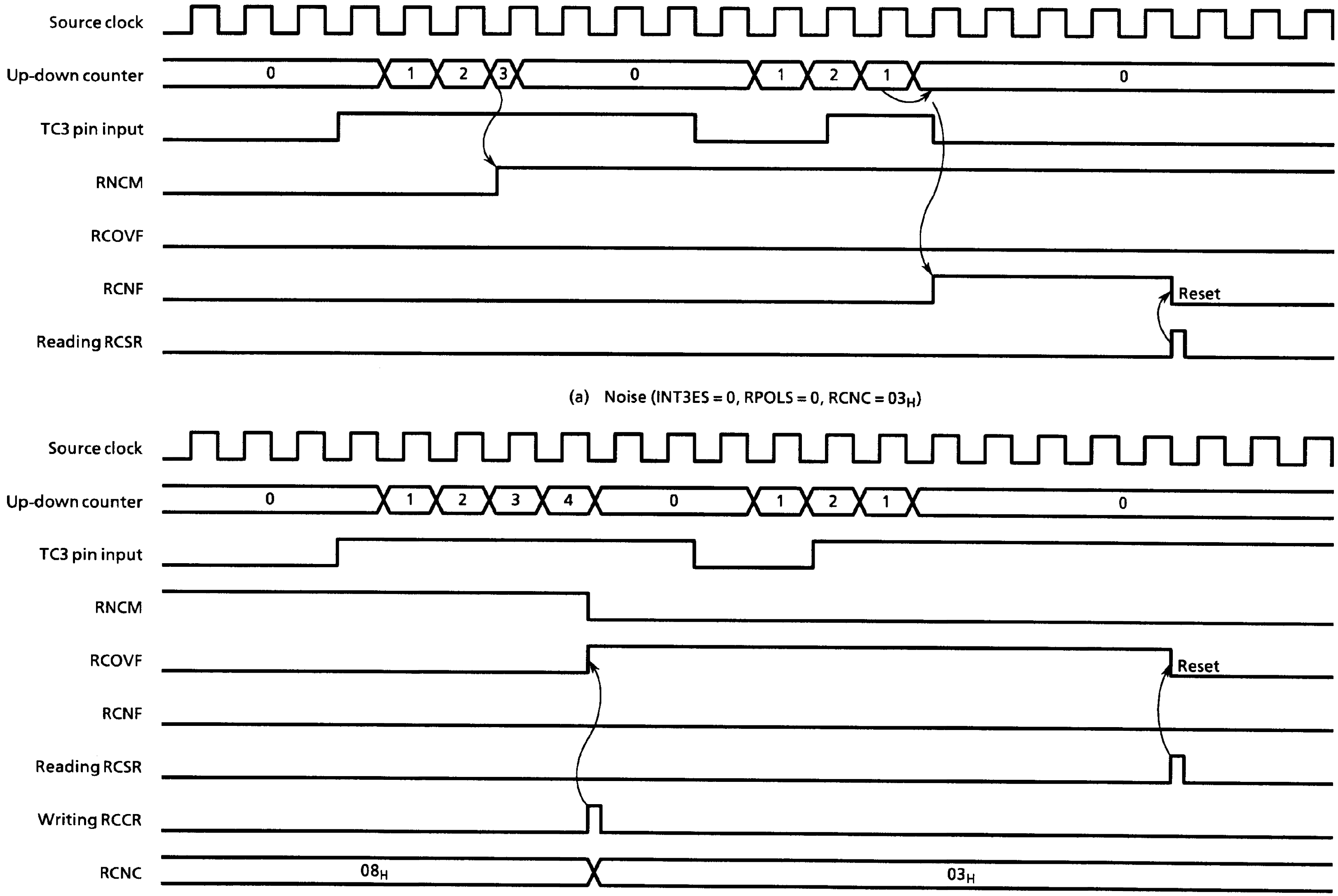
Note 1: Reading out the register RCSR resets RCNF and RCOVF.
Note 2: Source clock of timer / counter 3
Note 3: When a 5-bit up-down counter counts down to "0" after counting up, the RCNF defines to be noise.
*Note 4: fc; High-frequency clock [Hz], * ; Don't care*

Figure 2-33. Remote Control Receive Control Register and Remote Control Receive Status Register

Table 2-8. Combination between The Porality and The Edge Selection

RPOLS	INT3ES	TC3 pin input pulse (Interrupt occurrence is shown as arrow.)	Measurement
0	0		
	1		
1	0		
	1		

Note 1: When TC3CK is used in RCSCK, do not select an external clock to the TC3CK.
Note 2: Starting remote control receive detects an interrupt occurrence edge first.



(a) Noise (INT3ES = 0, RPOLS = 0, RCNC = 03_H)

(b) Noise rejection circuit overflow flag (INT3ES = 0, RPOLS = 1, RCNC = $08_H \rightarrow 03_H$)

Figure 2-34. Remote Control Receive Circuit Timing Chart

2.9 8-bit Timer/Counter 4 (TC4)

2.9.1 Configuration

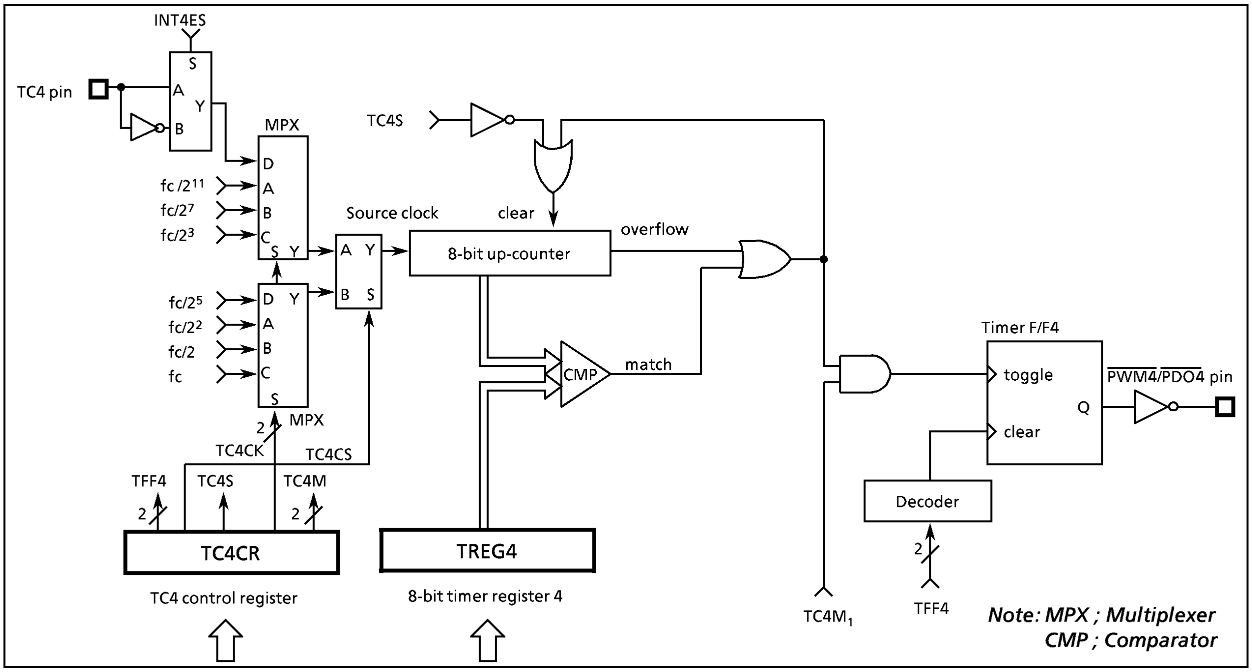


Figure 2-35. Timer/Counter 4 (TC4)

2.9.2 Control

Timer/counter 4 is controlled by timer/counter 4 control register (TC4CR) and timer register 4 (TREG4).

	7	6	5	4	3	2	1	0	
TREG4 (0001B$_H$)									Write only

	7	6	5	4	3	2	1	0	
TC4CR (0001C$_H$)	TFF4		TC4CS	TC4S	TC4CK		TC4M		(Initial value: 0000 0000)

TC4M	TC4 operating mode select	0*: reserved 10: Programmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode		Write only
TC4CK	TC4 source clock select [Hz]	A mode (TC4CS = 0)	B mode (TC4CS = 1)	
		00: $fc/2^{11}$ 01: $fc/2^7$ 10: $fc/2^3$ 11: reserved	00: $fc/2^5$ 01: $fc/2^2$ 10: fc/2 11: fc	
TC4S	TC4 start control	0: Stop & clear 1: Starts		
TC4CS	TC4 source clock mode select	0: A mode 1: B mode		
TFF4	Timer F/F4 control	00: Clear 11: reserved		

*Note 1: fc ; High-frequency clock [Hz], * ; Don't care*

Note 2: Set the operating mode, the source clock selection, the edge selection (INT4ES) and timer F/F4 control when the TC4 stops (TC4S = 0)

Note 3: Set TFF4 to "11" in the B mode.

Note 4: Values to be loaded to the timer register must satisfy the following condition.
(a) 5< TREG4< 251 in PWM output mode
(b) 0< TREG4 in others

Note 5: The source clock $fc/2^2$, fc/2 and fc must be used in only PWM output mode.

Note 6: TC4CR is a write-only register and must not be used with any of the read-modify-write instructions such as SET, CLR. etc.

Figure 2-36. Timer Register 4 and TC4 Control Register

2.9.3 Function

Timer/counter 4 supports two operating modes: programmable divider output and pulse width modulation output.

(1) **Programmable Divider Output (PDO) mode**

TC4 counts up using the internal clock. A match between the counter value and the value set in TREG4 inverts the timer F/F4 output and clears TC4. The inverted timer F/F4 output is output to the P52 ($\overline{\text{PDO4}}$) pin. For programmable divider output, set P52 output latches to 1. PDO mode can be used for approx. 50 % duty pulse output. Timer F/F4 can be initialized by program. At reset, timer F/F4 is initialized to 0.

```
Example: Output 1024 Hz pulse (fc = 8 MHz)
                    SET     (P5). 2               ;   Sets P52 output latch to 1.
                    LD      (TREG4), 1EH          ;   1/2048 ÷ 2^7/fc = 1EH
                    LD      (TC4CR), 00010110B    ;   Starts TC4.
```

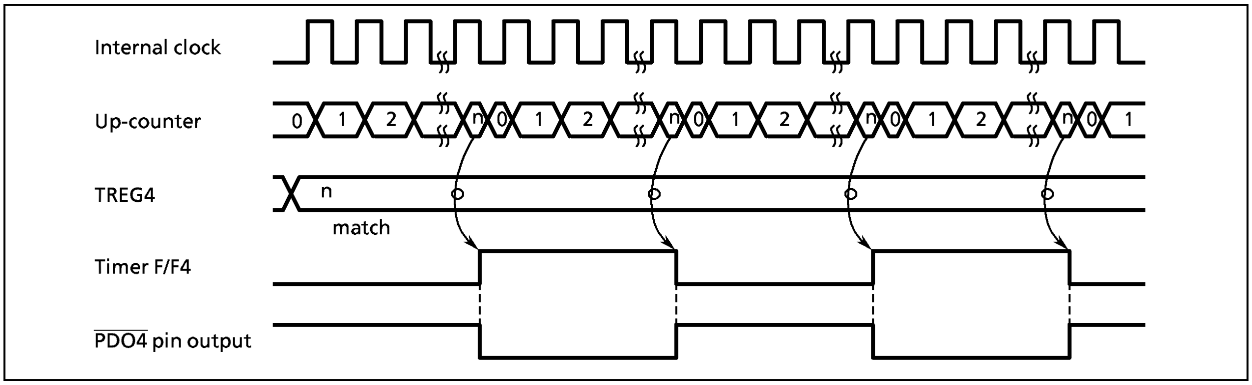


Figure 2-37. PDO Mode Timing Chart

(2) **Pulse Width Modulation (PWM) output mode**

Resolution of 8-bit PWM output can be performed. TC4 counts up using the internal clock. A match between the counter value and the value set in TREG4 inverts timer F/F4 output. TC4 continues counting up; an overflow inverts timer F/F4 output again, and clears TC4. The inverted timer F/F4 output is output to the P52 ($\overline{\text{PWM4}}$) pin. For PWM output, set the P52 output latch to 1.

TREG4 is a shift register (2 steps). Overwriting TREG4 during PWM output does not change output until output for one cycle is completed; thus, output can be changed continuously. First data are set in TREG4, then shifted at the start by TC4CR.

If data are overwritten during the cycle, the latest value is valid.

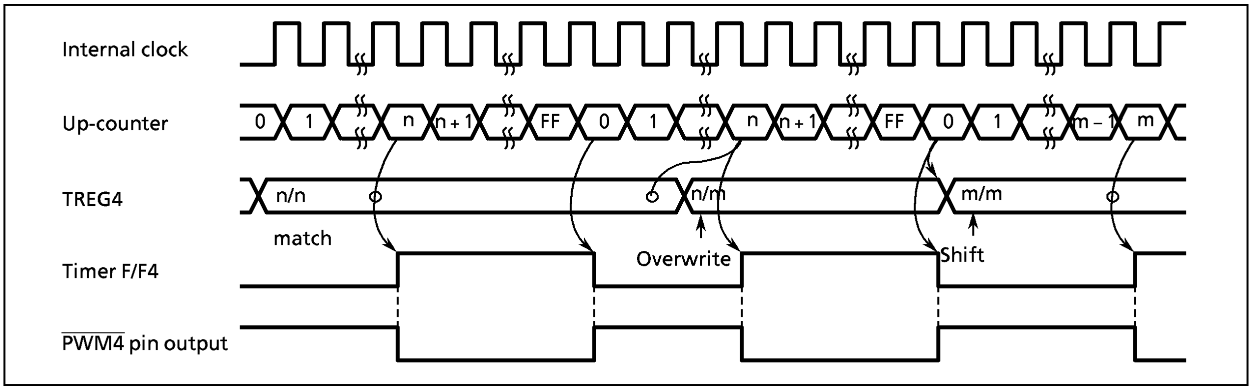


Figure 2-38. PWM Output Mode Timing Chart

Table 2-9. PWM Output Mode

Source Clock		Resolution	Repeat Cycle
Mode	NORMAL,IDLE Mode	When fc = 16 MHz	
A	fc / 2^{11} [Hz] fc / 2^7 fc / 2^3	128 μs 8 μs 0.5 μs	32.8 ms 2 ms 128 μs
B	fc / 2^5 [Hz] fc / 2^2 fc / 2 fc	2 μs 250 ns 125 ns 62.5 ns	512 μs 64 μs 32 μs 16 μs

2.10 8-bit Timer/Counter 5 (TC5)

2.10.1 Configuration

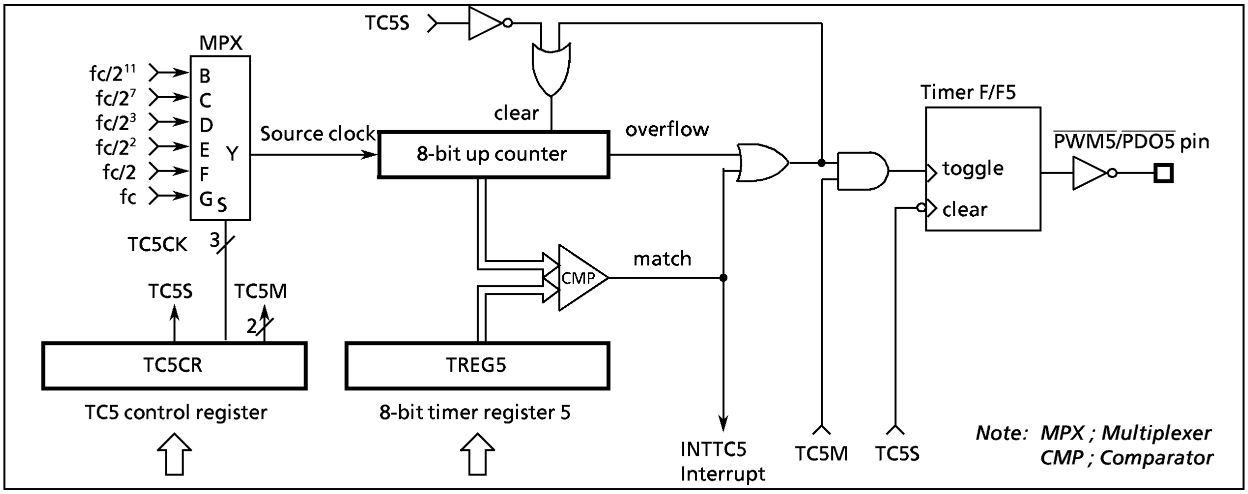


Figure 2-39. Timer/Counter 5 (TC5)

2.10.2 Control

Timer/counter 5 is controlled by timer/counter 5 control register (TC5CR) and timer register 5 (TREG5).

	7	6	5	4	3	2	1	0	
TREG5 (001D$_H$)									Write only

	7	6	5	4	3	2	1	0	
TC5CR (001E$_H$)			TC5S	TC5CK			TC5M		(Initial value: **00 0000)

TC5M	TC5 operating mode select	00: Timer mode 01: reserved 10: Programmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode	Write only
TC5CK	TC5 source clock select	000: reserved 001: Initial clock fc/2^{11} or fs/2^3 [Hz] 010: 〃 fc/2^7 011: 〃 fc/2^3 100: 〃 fc/2^2 101: 〃 fc/2 110: 〃 fc 111: reserved	
TC5S	TC5 start control	0: Stop & clear 1: Start	

*Note 1: fc ; High-frequency clock [Hz], * ; Don't care*

Note 2: Values to be loaded to the timer register must satisfy the followiing condition.

(a) 5< TREG5< 251 in PWM output mode

(b) 0< TREG5 in others

Note 3: The source clock fc/2^2, fc/2 and fc must be used in only PWM output mode.

Note 4: Set the mode and source clock when timer / counter stops (TC5S = 0).

Figure 2-40. Timer Register 5 and TC5 Control Register

2.10.3 Function

Timer/counter 5 supports three operating modes: timer, programmable divider output, and pulse width modulation output.

(1) **Timer mode**

TC5 counts up using the internal clock. A match between the counter value and the value set in TREG5 generates INTC5 and clears the counter. After the counter is cleared, it continues counting.

Table 2-10. Source Clock (internal clock) for Timer/Counter 5

Source clock	Resolution	Maximum setting time
NORMAL or IDLE mode	When fc = 16 MHz	When fc = 16 MHz
$fc/2^{11}$ [Hz]	128 μs	32.6 ms
$fc/2^7$	8 μs	2.0 ms
$fc/2^3$	0.5 μs	127.5 μs

(2) **Programmable Divider Output (PDO) mode**

TC5 counts up using the internal clock. A match between the counter value and the value set in timer registers 5 (TREG5) inverts timer F/F5 and clears TC5. The inverted timer F/F5 output is output to the P13 ($\overline{\text{PDO5}}$) pin. For programmable divider output, set P13 to output. PDO mode can be used for approx. 50% duty pulse output. INTTC5 is generated every inversion of $\overline{\text{PDO5}}$ output.

Example: Output 1024 Hz pulse (when fc = 4.194304 MHz)

```
        LD     (TC5CR), 00001010B     ;  Sets to PDO mode. (TC5M = 10, TC5CK = 010)
        SET    (P1). 3                ;  Sets P13 output latch to 1.
        LD     (TREG5), 10H           ;  1/1024 × 1/2 ÷ 2^7/fc = 10H
        LD     (TC5CR), 00101010B     ;  Starts TC5.
```

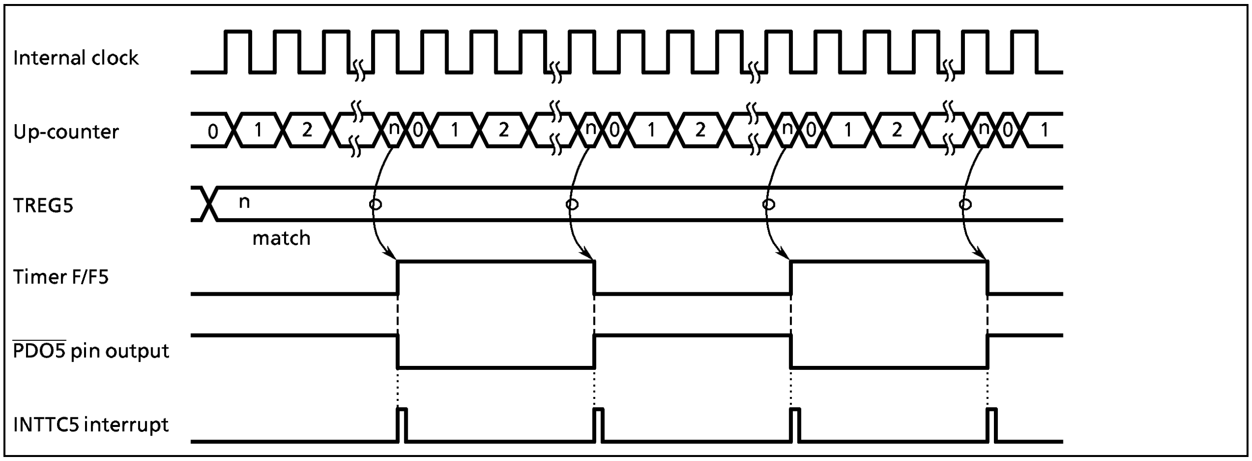


Figure 2-41. PDO Mode Timing Chart

(3) **Pulse Width Modulation (PWM) output mode**

Resolution of 8-bit PWM output can be performed. TC5 counts up using the internal clock. A match between the counter value and the value set in TREG5 inverts timer F/F5 output. TC5 continues counting up; an overflow inverts timer F/F5 output again, and clears TC5. The inverted timer F/F5 output is output to the P13 ($\overline{\text{PWM5}}$) pin. For PWM output, set the P13 output latch to 1.

TREG5 is a shift register (2 steps). Overwriting TREG5 during PWM output does not change output until output for one cycle is completed; thus, output can be changed continuously. First data are set in TREG5, then shifted at the start by TC5CR.

Note: The PWM output mode can be used in NORMAL and IDLE modes.

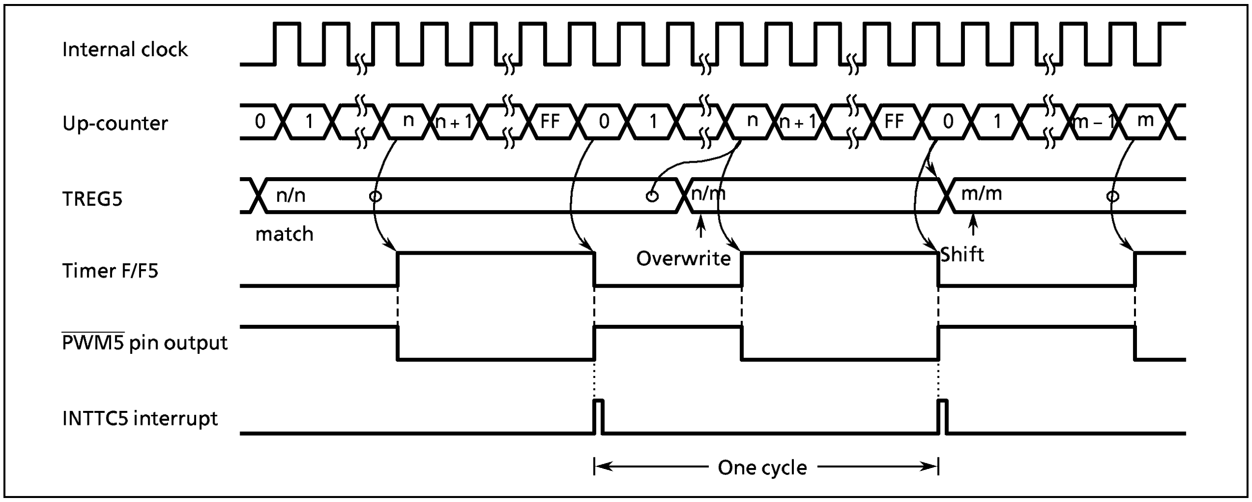


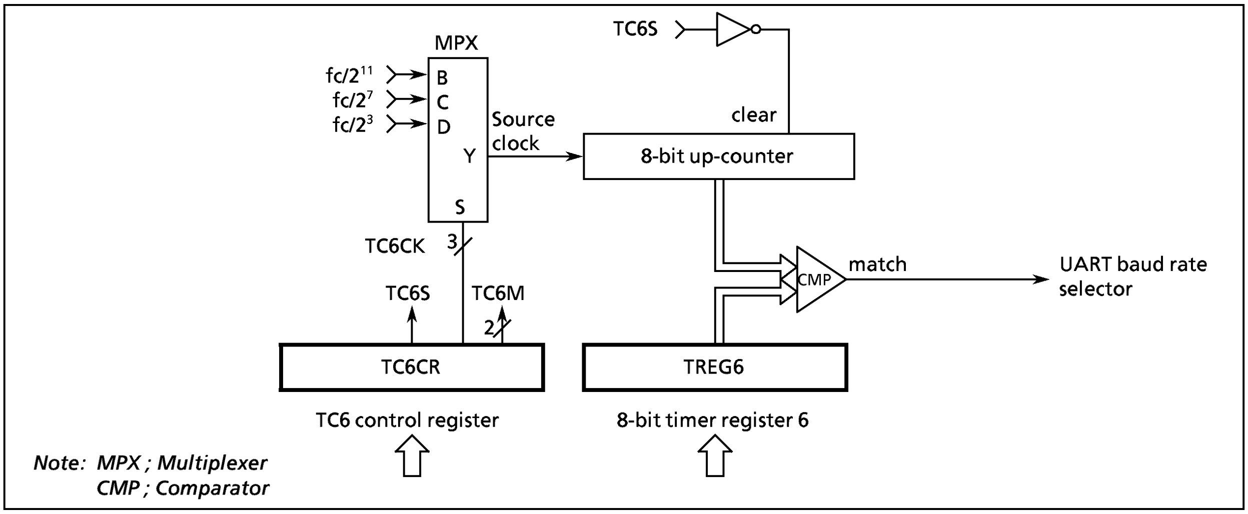
Figure 2-42. PWM Output Mode Timing Chart

Table 2-11. PWM Output Mode

Source clock	Resolution	Repeat cycle
NORMAL, IDLE mode	When fc = 16 MHz	When fc = 16 MHz
fc/2^2 [Hz]	250 ns	64 μs
fc/2	125 ns	32 μs
fc	62.5 ns	16 μs

2.11 8-Bit Timer/Counter 6 (TC6)

2.11.1 Configuration



Note: MPX ; Multiplexer
CMP ; Comparator

Figure 2-43. Timer/Counter 6 (TC6)

2.11.2 Control

Timer/counter 6 is controlled by timer/counter 6 control register (TC6CR) and timer register 6 (TREG6).

	7	6	5	4	3	2	1	0	
TREG6 (00FFE$_H$)									Write only

	7	6	5	4	3	2	1	0	
TC6CR (00FFF$_H$)			TC6S	TC6CK					(Initial value: **00 00**)

TC6CK	TC6 source clock select [Hz]	000 : reserved 001 : fc/2^{11} 010 : fc/2^7 011 : fc/2^3 1** : reserved	Write only
TC6S	TC6 start control	0: Stop & clear 1: Starts	

*Note 1: fc ; High-frequency clock [Hz], * ; Don't care*

Note 2: Values to be loaded to the timer register must satisfy the following condition. TREG6>0

Note 3: Set the source clock when TC6 stops (TC6S = 0).

Figure 2-44. Timer Register 6 and TC6 Cotrol Register

2.11.3 Function

TMP88CS48A timer/counter 6 can only be used as the UART baud rate generator.

(1) **UART baud generator mode**

TC6 counts up using the internal clock. A match between the counter value and the value set in timer register 6 can be used as the UART transfer clock.

When using UART baud rate generator mode, set BRG in the UART control register (UARTCR) to "110".

The UART transfer clock and transfer rate are determined as follows:

$$\text{Transfer clock} = \frac{\text{TC6 source clock}}{\text{TREG6 specified value}}$$

$$\text{Baud rate} = \frac{\text{Transfer clock}}{16}$$

2.12 Motor Control Circuit (PMD: Programmable Motor Driver)

The TMP88CS48A contains one channel of motor control circuits with the same function. This control circuit can control brushless DC motors and AC motors with or without sensors. Its primary functions shown below are included in hardware, allowing to reduce software burden and materialize motor control easily.

(1) Rotor position detecting function

- Position detection is possible for motors with and without sensors.
- To prevent erroneous detection, position detection can be set to be ascertained when multiple detected positions match.
- A position detection disabled period after PWM-on can be set.

(2) Independent timer/timer capture functions for motor control

- Two channels of magnitude comparison timers and matching comparison timers which both operate synchronously with position detection are incorporated.
- Interrupt can be generated when overflow occurs.
- Interrupt can be generated when capture is overwritten.

(3) PWM waveform generating function

- 16-bit PWM with 125 ns resolution can be generated.
- PWM interrupt frequency can be set.
- Dead time when PWM-on can be set.

(4) Protective function

- Protective function can be actuated by overload protective input.

(5) Emergency stop function in case of failure

- Operation can be emergency stopped by EMG input or timer overflow.
- This function cannot be easily reset by software runway.

(6) Automatic commutation/automatic position detection starting functions

- A double-buffer structure allows for position detection- or timer-synchronized automatic commutation.
- Position detection can be automatically started by setting a desired position detection period with a timer.

2.12.1 Outline of Motor Control

This section describes how to control brushless DC motors with square wave drive. In brushless DC motors, decision must be made of the stator winding to which current is flowed from the rotor's magnetic pole, and conducting windings must be switched over according to rotation of the rotor. The position of the rotor's magnetic pole is detected using a hall IC or other like sensor, and for sensorless, by detecting a polarity change point (zero crossing) of the voltage induced in the motor winding. For sensorless motors, the induced voltage is detected by conducting current in two phases, and not in the remaining other phase. For two-phase conduction, there are six conduction patterns as shown in Table 2-12, and these patterns are switched over synchronously with rotor phases. In this two-phase conduction, the conducting time in each phase is 120-degree phases with regard to 180-degree phases of the inducted voltage.

Table 2-12. Conduction Patterns

Conduction Pattern	Upper Transistor			Lower Transistor			Conducting Winding
	u	*v*	*w*	*x*	*y*	*z*	
Mode 0	ON	OFF	OFF	OFF	ON	OFF	U→V
Mode 1	ON	OFF	OFF	OFF	OFF	ON	U→W
Mode 2	OFF	ON	OFF	OFF	OFF	ON	V→W
Mode 3	OFF	ON	OFF	ON	OFF	OFF	V→U
Mode 4	OFF	OFF	ON	ON	OFF	OFF	W→U
Mode 5	OFF	OFF	ON	OFF	ON	OFF	W→V

One of the upper or lower transistors is PWM-controlled.

Brushless motors control revolutions by an applied voltage, and the applied voltage is controlled by PWM. At this time, conducting windings must be switched over synchronously with the phases of the voltage induced by revolutions. Figure 2-46 shows control timing when switchover of conducting windings is performed by sensorless control. Because zero-crossing in three-phase motors occurs six times in one period (360 degrees of electric angle) of the induced voltage, 60 degrees of electric angle exists between one zero-crossing to the next. Assuming this period to be one mode, the rotor position can be divided into six modes by zero-crossing. Each mode corresponds one for one to the six conduction patterns described above. The conduction pattern switchover (commutation) timing is 30 degrees of electric angle out of phase with respect to position detection by the induced voltage.

The mode duration can be found by detecting zero-crossing at given timing and counting the elapsed time from the previous zero-crossing. Because the mode duration is equivalent to 60 degrees of electric angle, in the case of Figure 2-46, timings can be calculated as shown below:

① Conducting winding switchover (commutation) timing

30 degrees of electric angle = mode duration / 2

② Position detection start timing

45 degrees of electric angle = mode duration × 3 / 4

③ Failure detection timing

120 degrees of electric angle = mode duration × 2

The position detection start timing is set to prevent erroneous detection of the induced voltage due to the fact that even after conduction is turned off, current continues flowing for a while for reasons of the motor reactance.

Control can be synchronized to the phases of the motor's induced voltage by successively calculating the above timing every zero-crossing that is detected six times in 360 degrees of electric angle and then processing commutation, position detection start, etc. in synchronism with the calculated timing.

The timing required for motor control as in this example can be set as desired using the internal timer of the PMD.

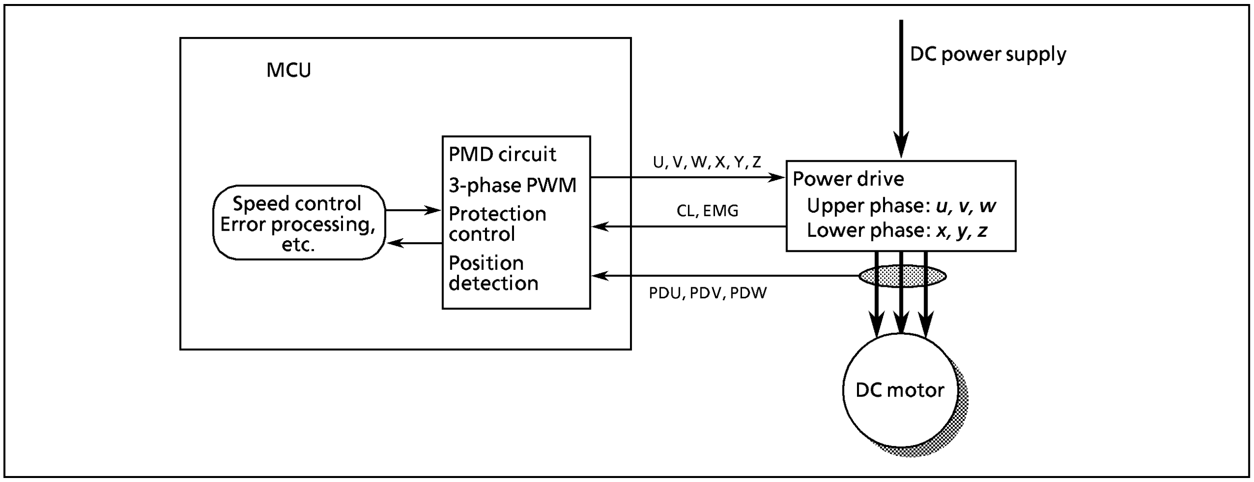


Figure 2-45. Conceptual Diagram of DC Motor Control

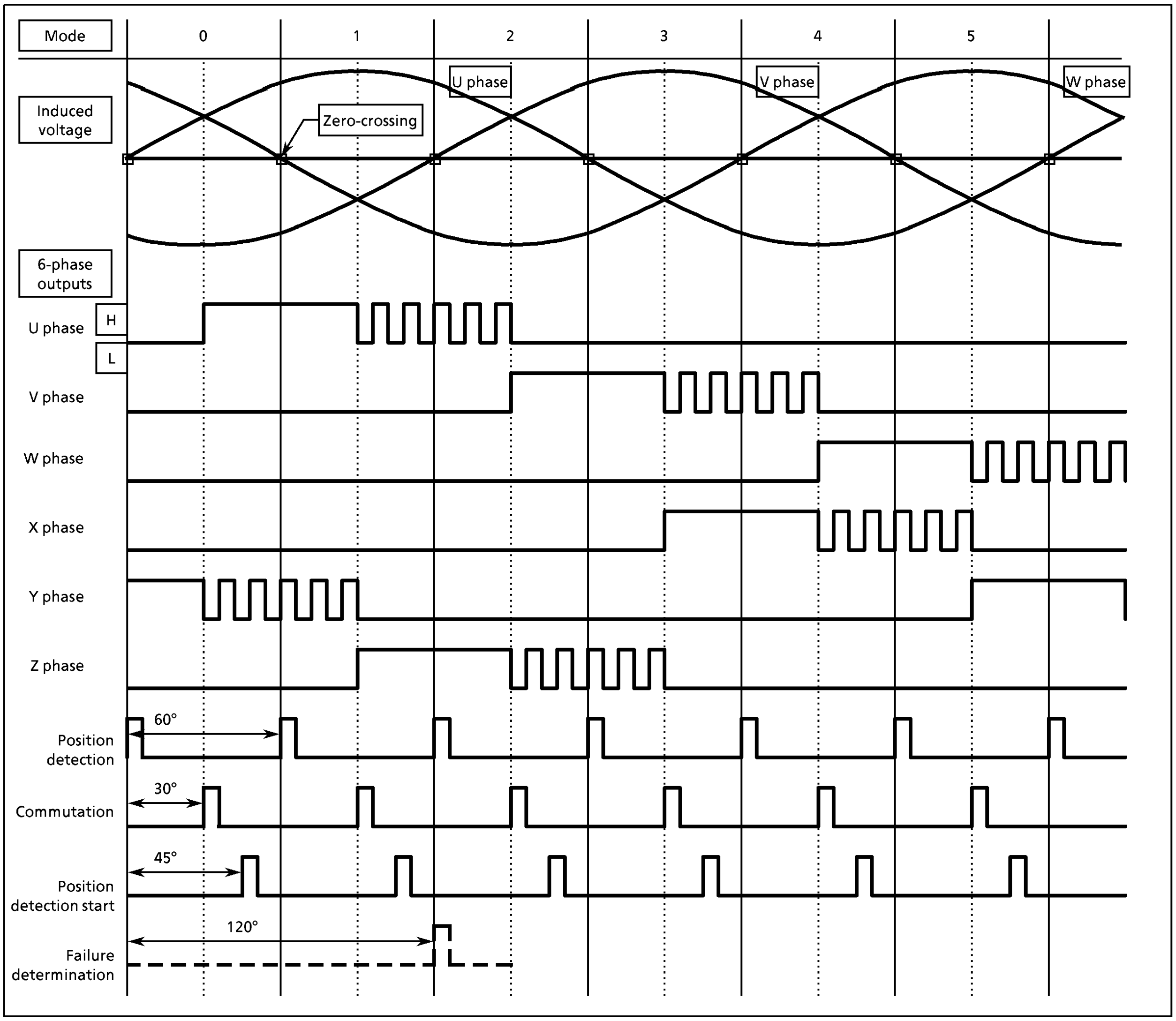


Figure 2-46. Example Timing Chart for Sensorless DC Motor Control

2.10.2 Structure of the Motor Control Circuit

The motor control circuit consists of a position detection unit to detect zero-crossing of the induced voltage or position sensor signal, a timer unit to generate events from position detection at four types of electric angle timing, and a 3-phase PWM output unit to generate PWM waves comprised of three phases. The input/output unit is configured as shown in Figure 2-48. When using ports for the PMD function, set input/output control (P0CRi) for input ports to "0". For output ports, set the data latch (P0i) to "1" and then input/output control to "1". Although Figure 2-48 shows the structure of only port P0, set the PMD function for input or output for all other input/output ports in the same way.

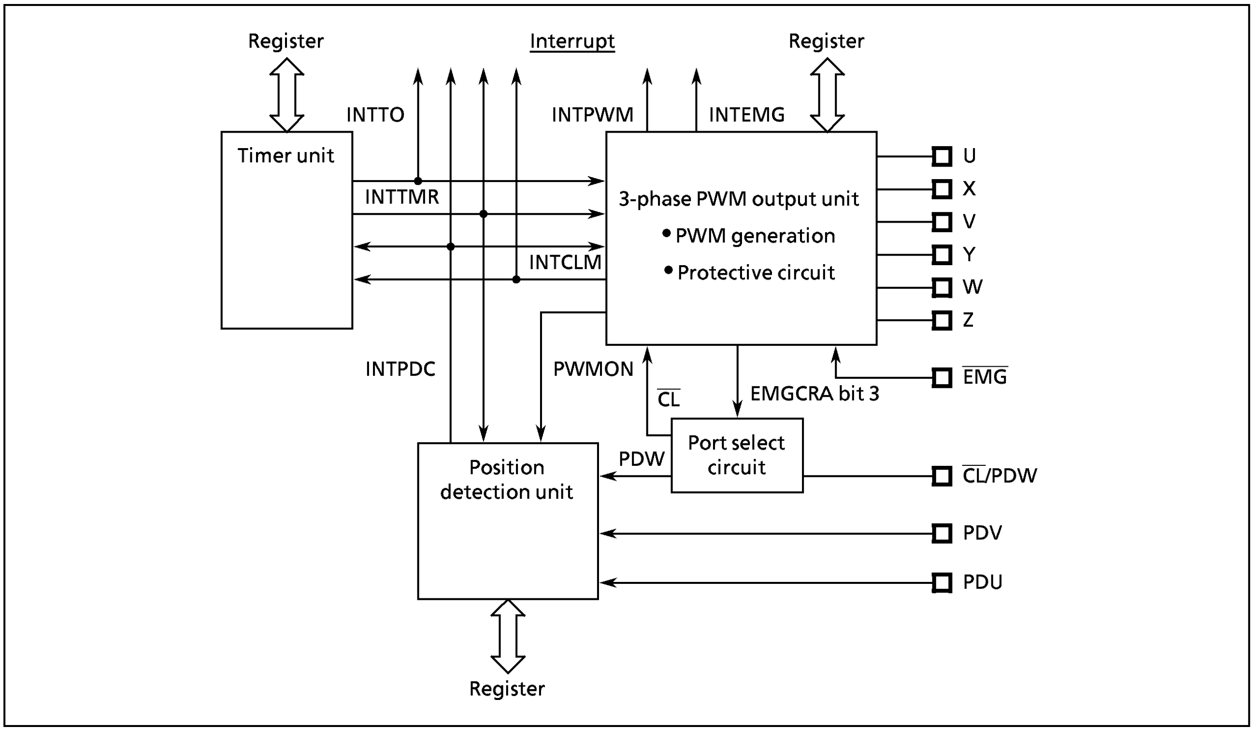
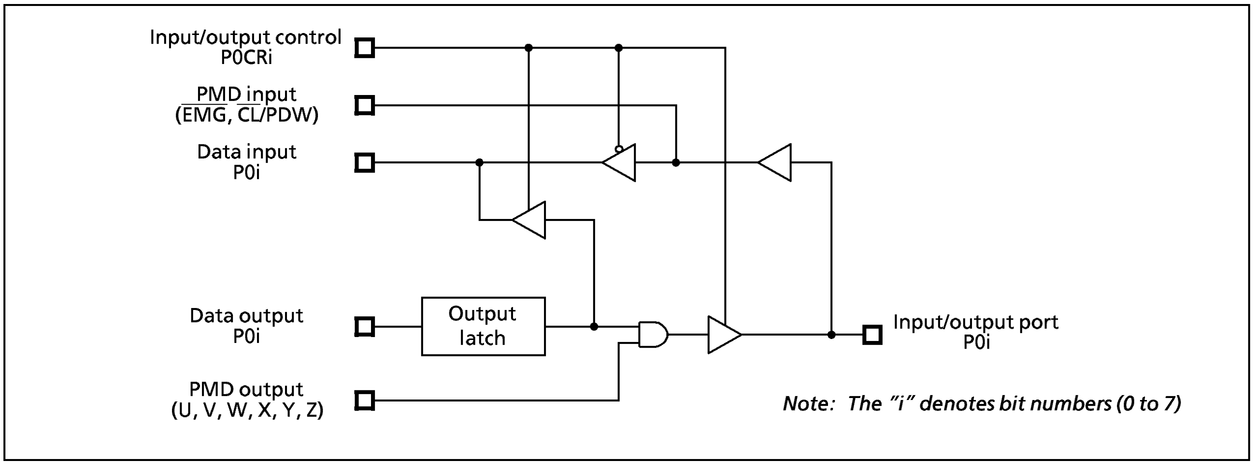


Figure 2-47. Block Diagram of the Entire Motor Control Circuit



Note: The "i" denotes bit numbers (0 to 7)

Figure 2-48. Structure of the Input/Output Unit

2.12.3 Position Detection Unit

The position detection unit identifies the rotor position of the motor from the input pattern on the position signal input port. The position signal input port has as its input the potential state of the motor winding in the case of sensorless DC motors or the Hall element signal in the case of sensor-mounted DC motors. A pattern of expected values corresponding to specific rotor positions is preset in the PMD output register (MDOUT), so that a position detection interrupt (INTPDC) is generated when the rotor turns and the input position signal and the expected value match. For three-phase brushless motors, when tabulated from the timing chart in Figure 2-49, there are six patterns of position signals for each mode. Once a predicted position signal pattern is set in MDOUT, a position detection interrupt is generated the moment the position signal input port goes to mode indicated by this expected value. The position signal in each phase shown in the diagram is an internal signal and cannot be observed from outside.

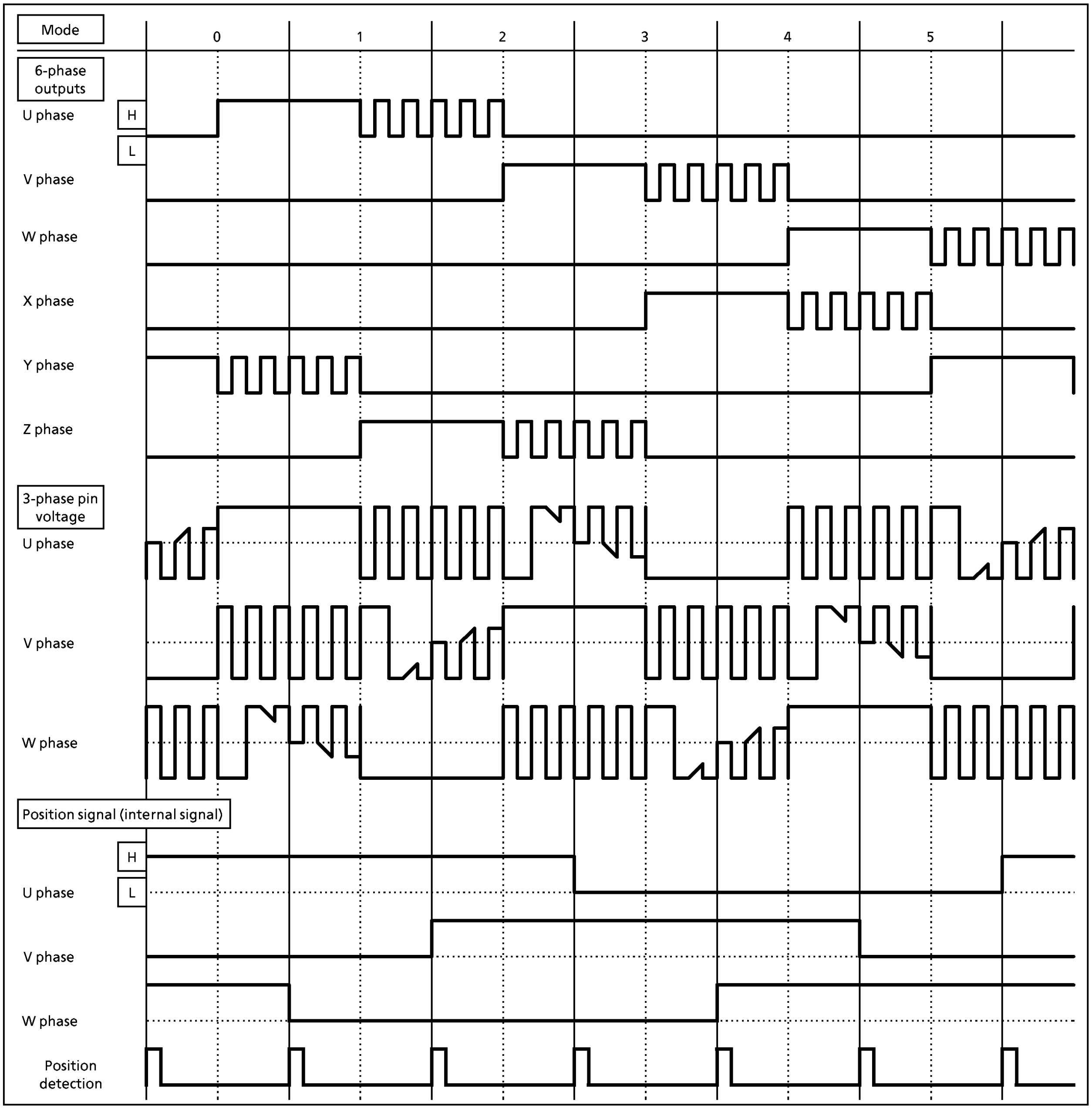


Figure 2-49. Example of Position Detection Timing Chart

Table 2-13. Position Signal Input Patterns

Position Detection Mode	U Phase	V Phase	W Phase
Mode 0	H	L	H
Mode 1	H	L	L
Mode 2	H	H	L
Mode 3	L	H	L
Mode 4	L	H	H
Mode 5	L	L	H

(1) Structure of the position detection unit

The position detection unit consists of a comparison circuit that compares the expected value for position signal with the input position signal, an erroneous detection prevention circuit that does not recognize matching detection as valid unless it is sampled a specified number of times, and a sampling control circuit that controls the sampling clock and sampling delay. The position detection unit is controlled by position detection control registers (PDCRA/B). A delay value can be set in the sampling delay setup register (SDREG) to set a sampling inhibit period every PWM-on. Position detection is accomplished by generating a matching detection signal when the expected value set in the PMD output register (MDOUT) matches the input position signal.

The expected value for position signal is updated by a latch synchronously with the output timing (MDOUTsync) from the commutation control circuit.

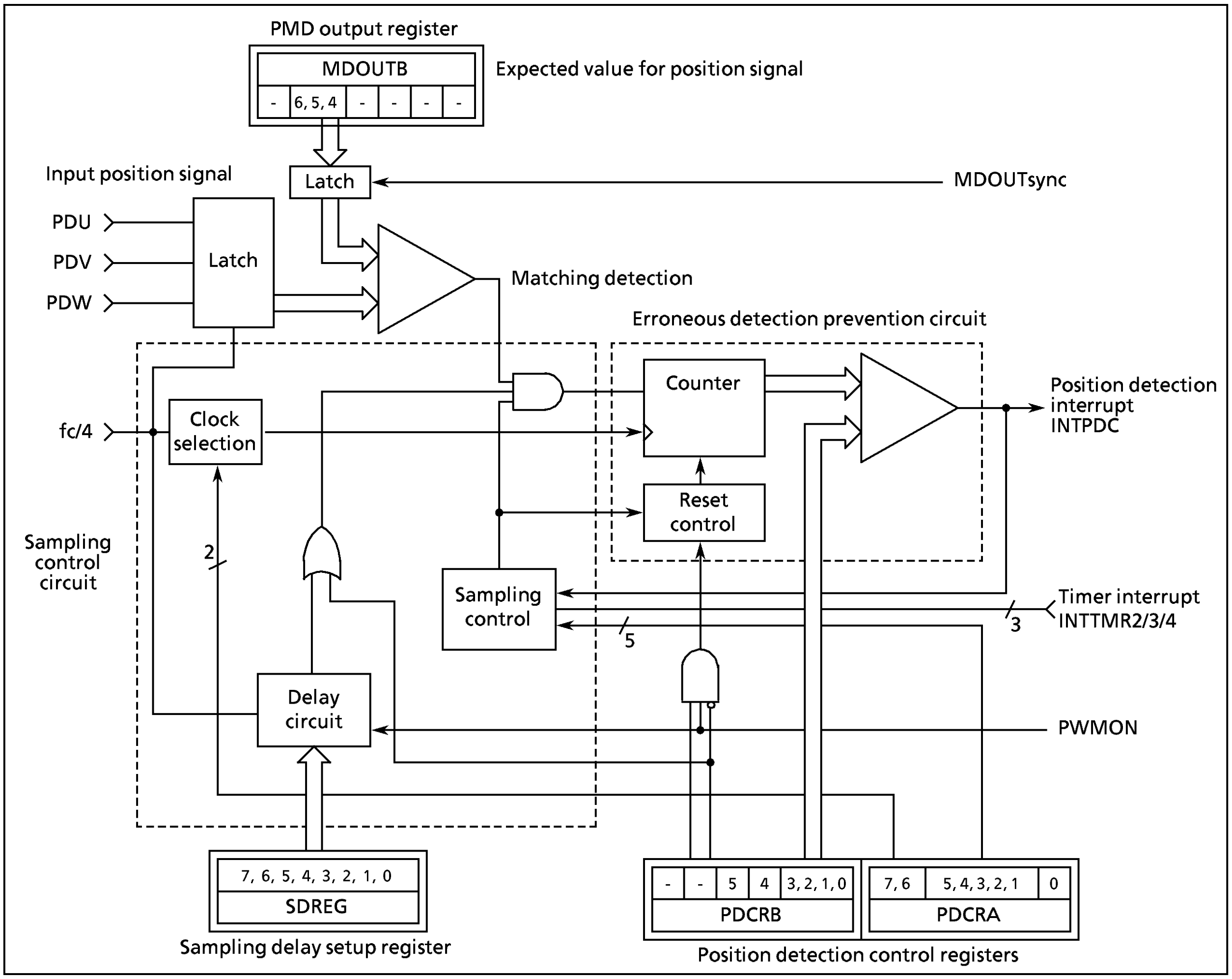


Figure 2-50. Position Detection Circuit

(2) Operation of the position detection unit

① Function list

Function	Command/Status		Operation
	Register Name	Bit Number	
Recount matching counts when PWM signal is on	PDCRB	5	Setting this bit to "1" clears matching counts of the erroneous detection prevention circuit every time PWM signal turns on and starts counting over again.
Select sampling method		4	Setting this bit to "1" selects analog sampling. The input signal is successively sampled.
			Setting this bit to "0" selects digital sampling. The input signal is sampled only when PWM is on.
Prevent erroneous position detection	PDCRA	3 to 0	Generates a position detection interrupt (INTPDC) after counting occurrences of matching 1 to 15 times as set.
Sampling period		7, 6	Selects a sampling period from 250 ns, 500 ns, 1 μs, or 2 μs (@fc = 16 MHz).
Stop sampling by timer		5, 4	Selects one of timers 2 to 4 and stops sampling when triggered by the timer.
Start sampling by timer		3, 2	Selects one of timers 2 to 4 and starts sampling when triggered by the timer.
Start/stop sampling by software		1	Setting this bit to "1" starts sampling.
			Setting this bit to "0" stops sampling.
Monitor sampling status			When this bit = "0", sampling remains idle.
			When this bit = "1", sampling is under way.
Enable/disable position detection function		0	Setting this bit to "1" enables position detection function. Sampling is ready to start.
			Setting this bit to "0" disables position detection function. Sampling also stops.
Set position signal expected value data	MDOUTB	6 to 4	When the expected-value pattern of the rotor position set in this register matches the input position signal, a position detection interrupt (INTPDC) is generated. If overload protection input is enabled, the W-phase expected value must be "0".
Set sampling delay	SDREG	7 to 0	Disables sampling for a given period immediately after PWM has turned on. Effective for only digital sampling, this period can be set in the range of 0 to 64 μs. When set in 8 bits, the resolution is 250 ns (@fc = 16 MHz).
Select CL/PDW port function	EMGCRA	3	When set to "0", this bit functions as an overload protection input pin. The W-phase position signal input is held "0".
			When set to "1", this bit functions as a W-phase position signal input pin. The overload protection input is held "1".
Automatically stop sampling	–	–	When a position detection interrupt (INTPDC) occurs, sampling is automatically stopped.

② Register list

PDCRB
(00F81$_H$)

7	6	5	4	3	2	1	0

(Initial value: 0000 0001)

5	Set recount of matching counts when PWM is on	0: Counts continuously. 1: Recounts every time PWM is on. Effective only when digital sampling is selected.	R/W
4	Select sampling method (note)	0: Samples only when PWM is on (digital sampling). 1: Successively samples (analog sampling).	
3, 2, 1, 0	Matching count of position signal	0001 to 1111: Generates position detection interrupt (INTPDC) when detection matches 1 to 15 times (to prevent erroneous detection). Counts are updated with sampling period.	

Note: When you chose digital sampling, be careful about the order in which to set the output circuit.
To enable the PWM control circuit, set MDCR bit 0 to 1 and then PDCRA bit 0 to 1.
To disable the PWM control circuit, set PDCRA bit 0 to 0 and then MDCR bit 0 to 0.

PDCRA
(00F80$_H$)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

7, 6	Select position detection circuit's sampling input clock	00: $fc/2^2$ [Hz] 01: $fc/2^3$ 10: $fc/2^4$ 11: $fc/2^5$	R/W
5, 4	Select sampling stop condition timer	00: Does not use timers for stop condition. 01: Timer 2 (INTTMR2) 10: Timer 3 (INTTMR3) 11: Timer 4 (INTTMR4)	
3, 2	Select sampling start condition timer	00: Does not use timers for start condition. 01: Timer 2 (INTTMR2) 10: Timer 3 (INTTMR3) 11: Timer 4 (INTTMR4)	
1	Command to start/stop sampling	0: Stops sampling. 1: Starts sampling.	W
		0: Sampling inactive 1: Sampling in progress	R
0	Enable/disable position detection function	0: Disables position detection function, with sampling stopped. 1: Enables position detection function, with sampling ready to start.	R/W

Note: Except for bits 0 and 1, be sure to set a pair of bits, and not a single bit.

Figure 2-51. Position Detection Control Register

SDREG
(00F82$_H$)

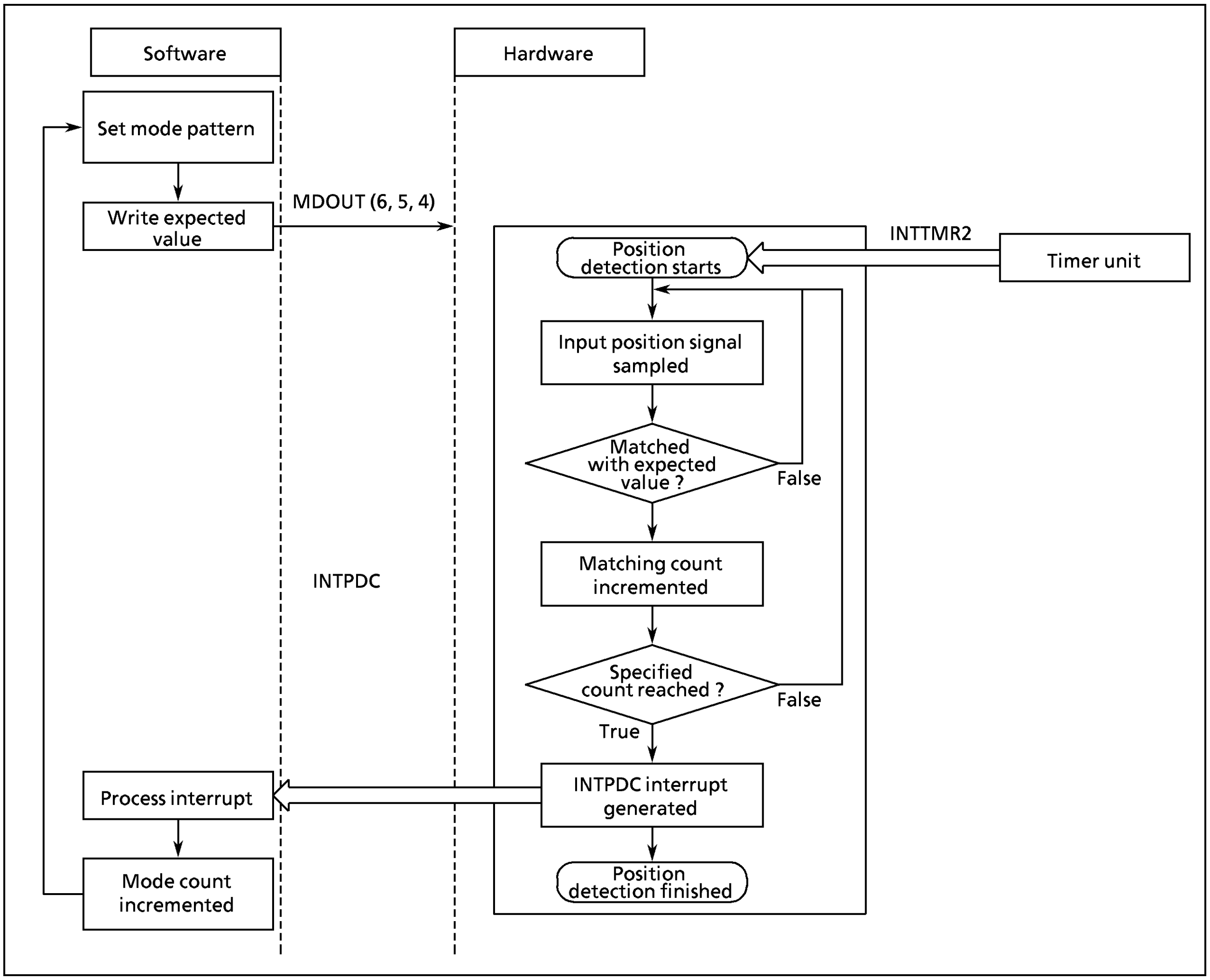
7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

7 to 0	Set sampling delay time	0000 0000 to 1111 1111: 0 to 255 Resolution = 1 / (fc / 22)	R/W

Figure 2-52. Sampling Delay Setup Register

(3) Outline of processing by the position detection unit


Software
Hardware
Set mode pattern
Write expected value
MDOUT (6, 5, 4)
INTTMR2
Timer unit
Position detection starts
Input position signal sampled
Matched with expected value ?
False
Matching count incremented
INTPDC
Specified count reached ?
False
True
Process interrupt
INTPDC interrupt generated
Mode count incremented
Position detection finished

2.12.4 Timer Unit

The timer unit has an up-counter (mode timer) which is cleared by a position detection interrupt (INTPDC), so that four types of timing interrupts (INTTMR1 to 4) referenced to it can be created. This timer function allows to generate a commutation trigger, a position detection start trigger, etc. Furthermore, the mode timer has a capture function which is capable of automatic capturing synchronously with position detection or overload protection. This makes it possible to calculate motor revolutions by measuring position detection intervals.

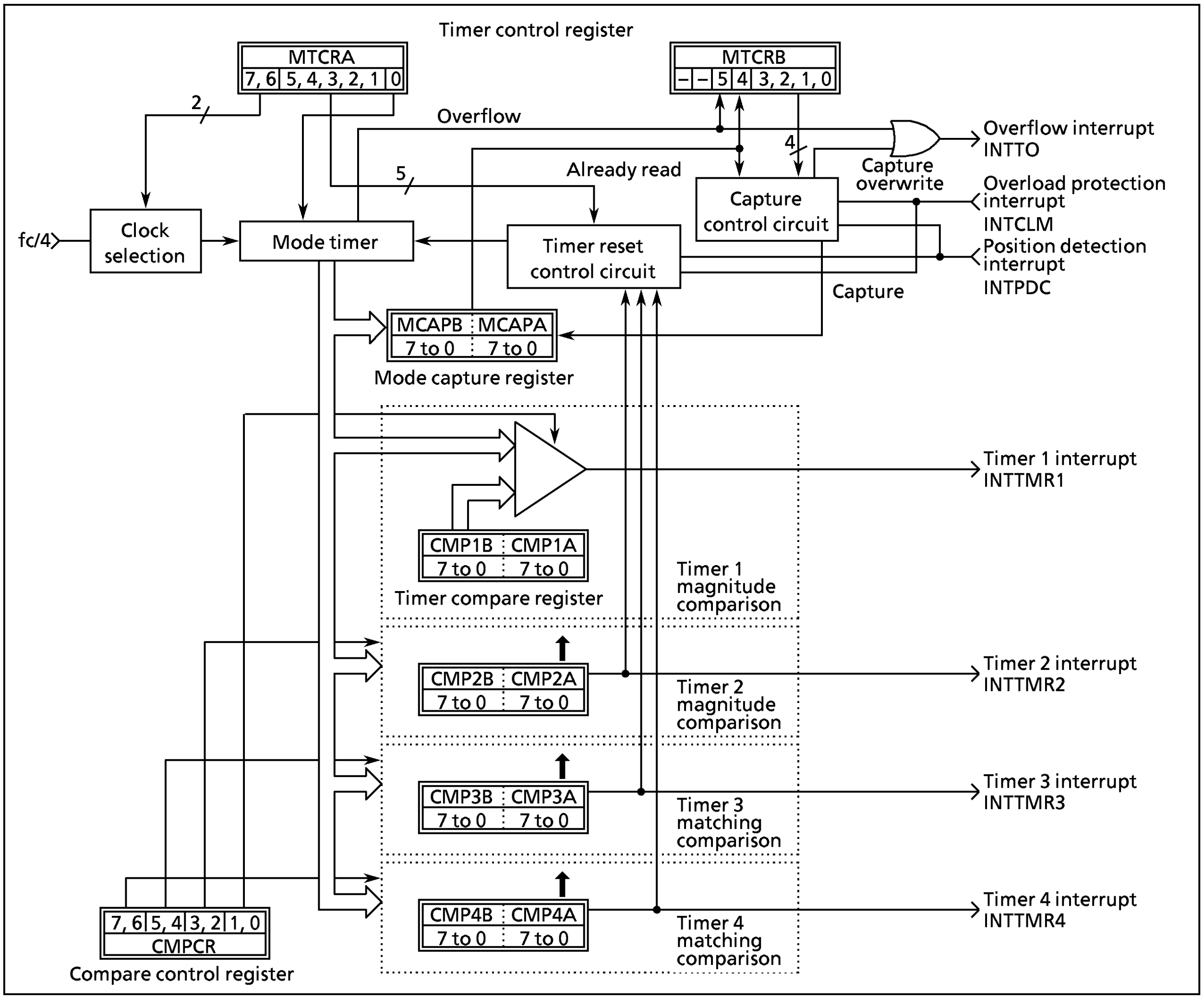


Figure 2-53. Timer Unit Circuit Configuration

(1) Structure of the timer unit

The timer unit consists of a mode timer, four timers, and a mode capture register, and is controlled by timer control registers and a compare control register.

The mode timer is an up-counter which is cleared by position detection, and functions as a reference counter for the timer unit. When it overflows, an overflow interrupt (INTTO) is generated.

Timers 1 to 4 compare the mode timer's count value with the values set in timer compare registers 1 to 4 to generate timer interrupts 1 to 4.

Timers 1, 2 generate an interrupt when the count value is greater than the values of compare registers 1, 2. Timers 3, 4 generate an interrupt when the count value matches the values of compare registers 3, 4.

Comparison is started by writing data to the compare registers while timers 1 to 4 have their compare functions enabled, and is stopped when the condition is met (holds true). When automatic updating is selected, comparison is automatically started by resetting the mode timer.
The mode capture register can be updated automatically in synchronism with position detection or overload protection interrupt or by a software command. If capture operation is activated again before reading the capture register, an overflow interrupt (INTTO) is generated.

(2) Operation of the timer unit

① Function list

Function		Command/Status		Operation
		Register Name	Bit Number	
Check mode timer overflow status		MTCRB	5	Generates INTTO when the mode timer overflows, in which case the bit is set to "1". This bit is set to "0" when the mode timer is reset.
Check capture read			4	Already read when this bit = "0". Not read after capture when this bit = "1".
Capture mode timer			3	Setting this bit to "1" captures timer value when overload protection is activated.
			2	Captures timer value when it is set to "1".
			1	Setting this bit to "1" captures timer value when position is detected.
Enable/disable capture overwrite			0	Setting this bit to "1" enables capture overwrite. Setting this bit to "0" disables capture overwrite and generates INTTO when capture overwrite occurs. Capture is not overwritten.
Select mode timer clock		MTCRA	7, 6	Selects 250 ns, 500 ns, 1 μs, or 2 μs clock (@fc = 16 MHz).
Reset mode timer			5, 4	Mode timer is reset by a trigger from the selected timer.
			3	Setting this bit to "1" resets the mode timer upon activation of overload protection.
			2	Setting this bit to "1" resets the mode timer immediately.
			1	Setting this bit to "1" resets the mode timer when position is detected.
Enable/disable timer function			0	Setting this bit to "1" enables timer function. Setting this bit to "0" disables timer function.
Enable compare function	Timer 4	CMPCR	6	Setting this bit to "1" enables matching comparison. Comparison starts upon writing data to the compare register and stops when condition is met (holds true).
	Timer 3		4	
	Timer 2		2	Setting this bit to "1" enables magnitude comparison. Comparison starts upon writing data to the compare register and stops when condition is met (holds true).
	Timer 1		0	
Automatically update comparison start	Timer 4		7	When this bit is set to "1", comparison stops when compare condition is met and automatically restarts when the When this bit is set to "0", comparison stops when compare condition is met and is restarted by writing to the compare register.
	Timer 3		5	
	Timer 2		3	
	Timer 1		1	

② Register list

MTCRB ($00F85_H$)

7	6	5	4	3	2	1	0

(Initial value: 1100 0000)

5	Mode timer overflow status	0: No overflow. 1: Overflow occurred.	R
4	Check capture read	0: Already read. 1: Not read yet.	
3	Enable overload protection as mode timer capture condition	0: Does not use overload protection as capture condition. 1: Captures timer value when overload protection starts.	R/W
2	Mode timer capture command	Always "0" 0: No operation. 1: Captures immediately upon writing to bit.	R W
1	Enable position detection as mode timer capture condition	0: Does not use position detection as capture condition. 1: Captures timer value when position is detected.	R/W
0	Enable/disable capture overwrite	0: Disables capture overwrite. INTTO is generated when overwrite occurs. Capture is not overwritten. 1: Enables capture overwrite when not read yet.	

MTCRA ($00F84_H$)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

7, 6	Select mode timer clock	00: $fc/2^2$ [Hz] 01: $fc/2^3$ 10: $fc/2^4$ 11: $fc/2^5$	R/W
5, 4	Select timer for mode timer reset condition	00: Does not use timer as reset condition 01: Timer 2 (INTTMR2) 10: Timer 3 (INTTMR3) 11: Timer 4 (INTTMR4)	
3	Enable overload protection as mode timer reset condition	0: Does not use overload protection as capture condition. 1: Resets mode timer when overload protection starts (INTCLM).	
2	Mode timer reset command	Always "0" 0: No operation. 1: Resets mode timer.	R W
1	Enable position detection as mode timer reset condition	0: Does not use position detection as capture condition. 1: Resets mode timer when position is detected.	R/W
0	Enable/disable timer function	0: Disables timer function. 1: Enables timer function. Enables up-count.	

Note: Before selecting clock (bits 4, 5), be sure to disable timer function (bit 0).

Figure 2-54. Timer Control Register

CMPCR (00F83$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

7	Automatically update comparison start for compare 4	0: Comparison stops when condition is met. 1: After being stopped, comparison starts automatically when the mode timer is reset.	R/W
6	Enable compare 4	0: Disables compare operation. 1: Enables compare operation. Comparison starts upon writing data to timer 4 compare register and stops when timer 4 compare register matches with the mode timer as compared, at which time an interrupt is generated.	
5	Automatically update comparison start for compare 3	0: Comparison stops when condition is met. 1: After being stopped, comparison starts automatically when the mode timer is reset.	
4	Enable compare 3	0: Disables compare operation. 1: Enables compare operation. Comparison starts upon writing data to timer 3 compare register and stops when timer 3 compare register matches with the mode timer as compared, at which time an interrupt is generated.	
3	Automatically update comparison start for compare 2	0: Comparison stops when condition is met. 1: After being stopped, comparison starts automatically when the mode timer is reset.	
2	Enable compare 2	0: Disables compare operation. 1: Enables compare operation. Comparison starts upon writing data to timer 2 compare register and stops when timer 2 compare register matches with the mode timer as compared, at which time an interrupt is generated.	
1	Automatically update comparison start for compare 1	0: Comparison stops when condition is met. 1: After being stopped, comparison starts automatically when the mode timer is reset.	
0	Enable compare 1	0: Disables compare operation. 1: Enables compare operation. Comparison starts upon writing data to timer 1 compare register and stops when timer 1 compare register matches with the mode timer as compared, at which time an interrupt is generated.	

Figure 2-55. Timer Compare Control Register

MCAPB (00F87$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
MCAPA (00F86$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

Mode capture Upper: MCAPB, Lower: MCAPA	0000$_H$ to FFFF$_H$: Sets mode timer value during capture operation.	R

Figure 2-56. Mode Capture Register

CMP1B (00F89$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
CMP1A (00F88$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

Timer 1 Upper: CMP1B, Lower: CMP1A	0000$_H$ to FFFF$_H$: Sets value to compare with the mode timer.	R/W

CMP2B (00F8B$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
CMP2A (00F8A$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

Timer 2 Upper: CMP2B, Lower: CMP2A	0000$_H$ to FFFF$_H$: Sets value to compare with the mode timer.	R/W

CMP3B (00F8D$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
CMP3A (00F8C$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

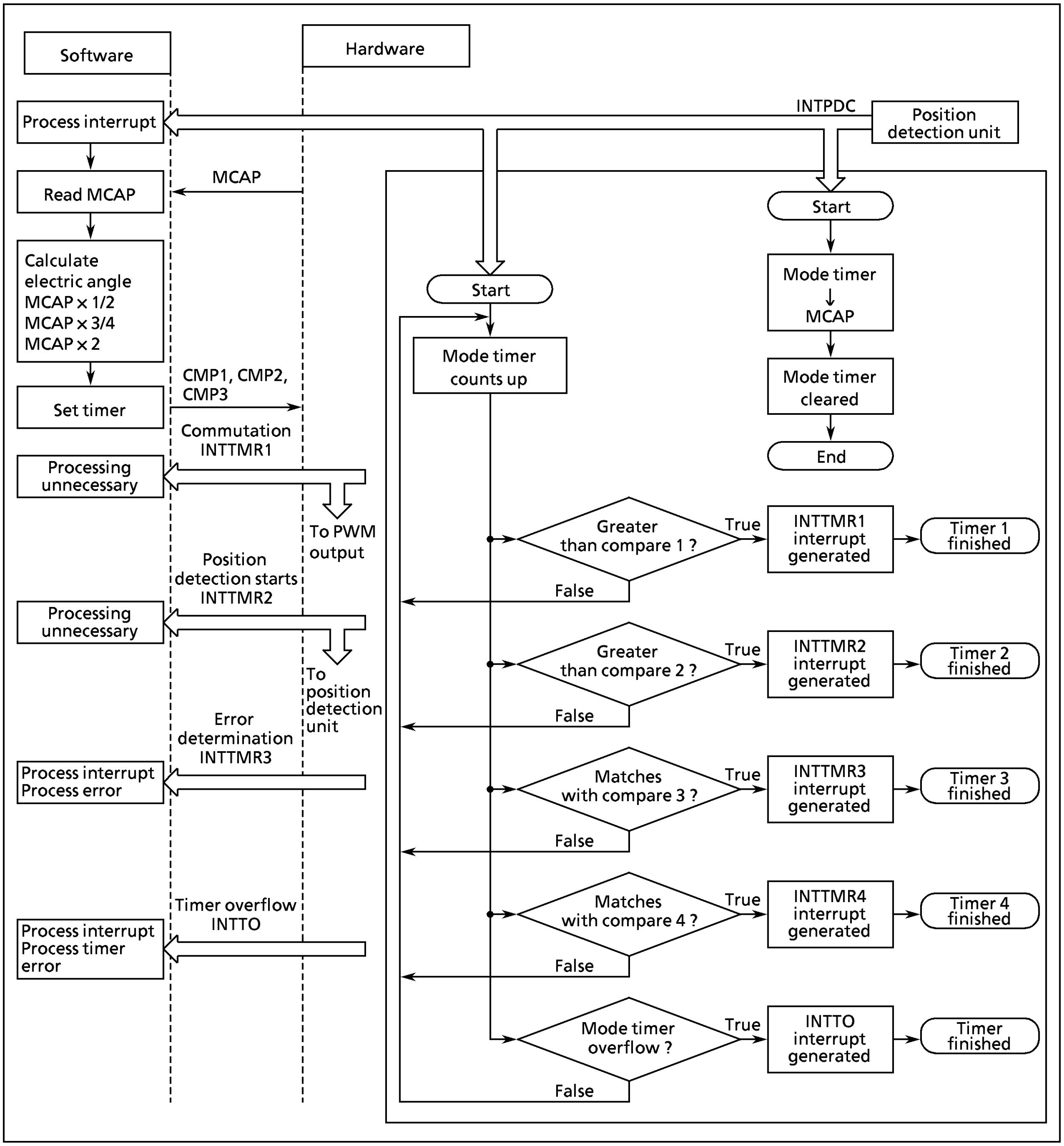
Timer 3 Upper: CMP3B, Lower: CMP3A	0000$_H$ to FFFF$_H$: Sets value to compare with the mode timer.	R/W

CMP4B (00F8F$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
CMP4A (00F8E$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

Timer 4 Upper: CMP4B, Lower: CMP4A	0000$_H$ to FFFF$_H$: Sets value to compare with the mode timer.	R/W

Figure 2-57. Timer Compare Registers

(3) Outline processing by the timer unit



2.12.5 Three-phase PWM Output Unit

The three-phase PWM output unit has the function to generate three-phase PWM waves in a given pulse width and the commutation function capable of controlling brushless DC motors. It also has protective functions for overload protection, emergency stop, etc. to protect the power drive unit, and a dead time addition function to prevent the upper/lower transistors matched in phase from turning on simultaneously and getting shorted when they are switched over.

(1) Structure of the three-phase PWM output unit

The three-phase PWM output unit consists of a pulse width modulation circuit, a commutation control circuit, a protective circuit (emergency stop/overload), and a dead time control circuit.

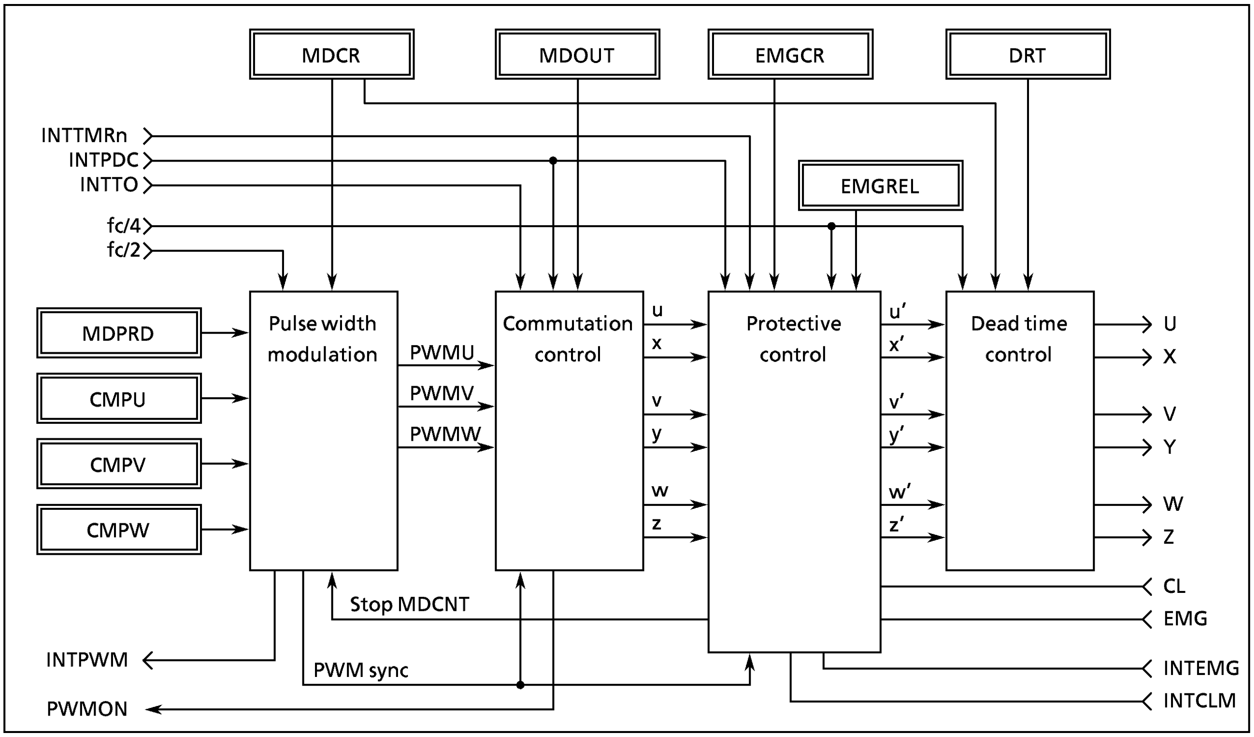


Figure 2-58. Structure of the Three-phase PWM Output Unit

① Pulse width modulation circuit (PWM wave generating unit)

This circuit generates independent three-phase PWM waves that are equal in PWM frequency. PWM waveform mode can be selected from triangular wave modulation and sawtooth wave modulation by setting PMD control register (MDCR) bit 1. The PWM frequency is set by PMD period register (MDPRD). The relationship between register values and frequencies is shown below.

Sawtooth wave PWM: MDPRD register set value = External oscillation frequency [Hz] / PWM frequency [Hz] × 2

Triangular wave PWM: MDPRD register set value = External oscillation frequency [Hz] / PWM frequency [Hz] × 4

The PMD period register is configured with double-buffers, with the comparator input updated at PWM period. If the PWM interrupt period is set to a half period, the comparator input is updated at a half PWM period.

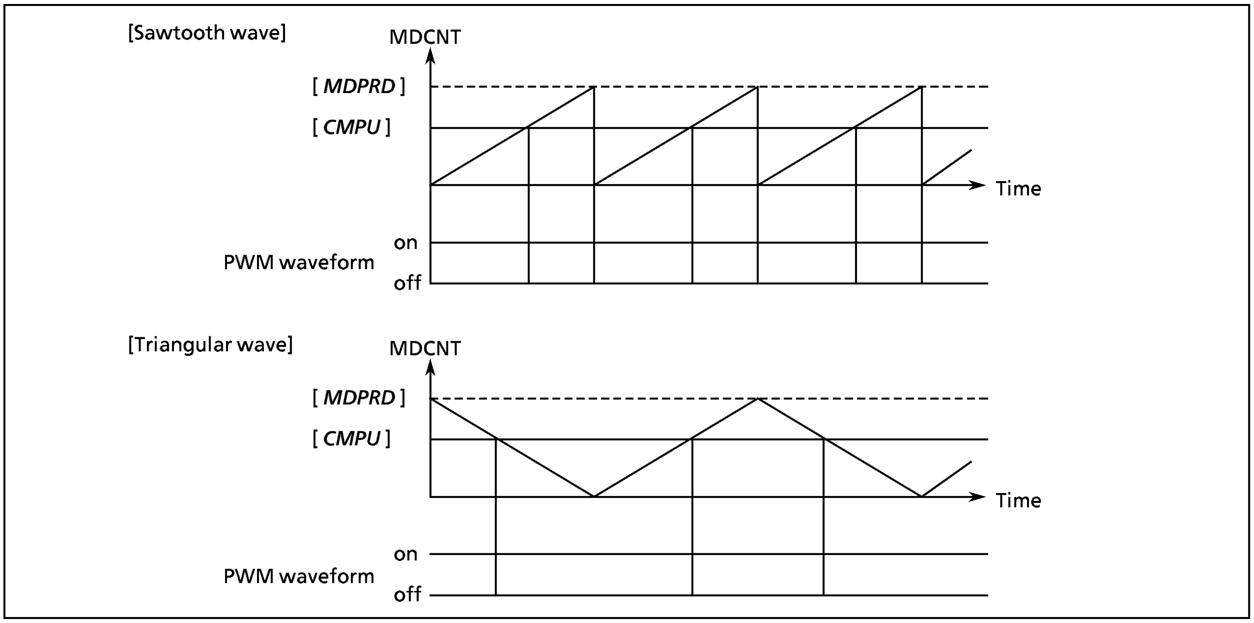


Figure 2-59. PWM Waveforms

The pulse width modulation circuit compares PWM compare register (CMPU/V/W) value and the carrier wave generated by PMD counter (MDCNT) with a comparator to determine which is larger than the other as it generates PWM waveforms.

In three-phase output control, one of two methods for generating three-phase PWM can be set.

i) Three-phase independent mode:
Values are set independently in three-phase PMD compare registers to generate independent three-phase PWM waveforms. This mode is used to generate any drive waveform such as a sinusoidal wave.

ii) Three-phase common mode:
A value is set in only U-phase PMD compare register to generate identical three-phase PWM waveforms using the U-phase set value. This mode is used for square wave drive of DC motors.

The PMD compare register in each phase has a compare register, and is configured with double-buffers. The PMD compare register value is loaded into the compare register synchronously with PWM period. If the PWM interrupt period is set to a half period, the register content is updated at a half PWM period.

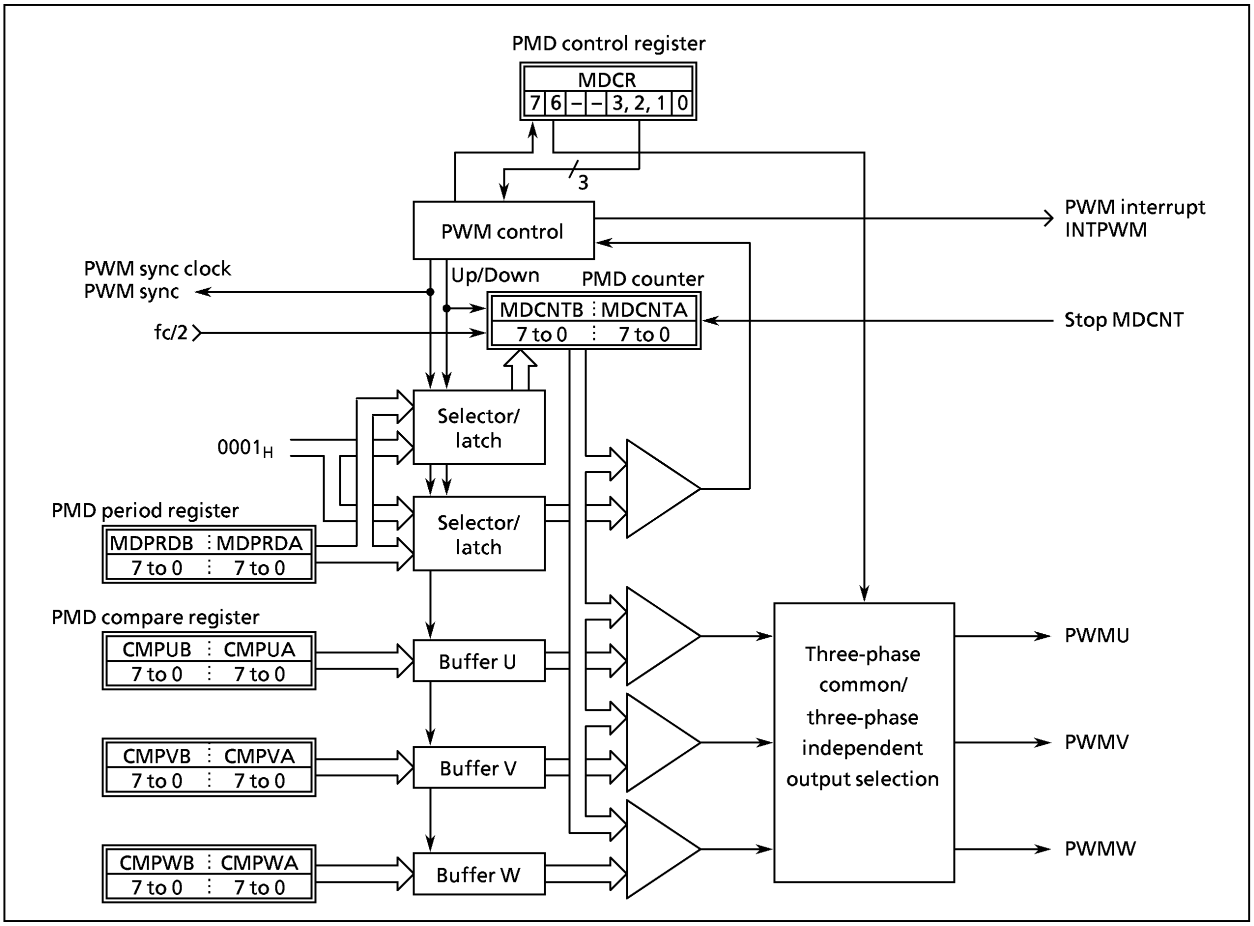


Figure 2-60. Pulse Width Modulation Circuit

② Commutation control circuit

This circuit controls output ports according to the contents set in PMD output registers (MDOUTA/B). The setup contents are divided between sync signal section during port output and settings for port output. The sync signal select bit of MDOUT takes effect immediately after a write. Except this bit, the register is configured with double-buffers, and is updated by a selected sync signal.

Output settings for six ports are asserted or deasserted independently by MDOUTA bits 5 to 0. Furthermore, for each of three U, V, and W phases, selection is made between PWM output and "H"/"L" output by setting MDOUTB bits 1, 2, and 3. When PWM output is selected, PWM waveform is obtained. When "H"/"L" output is selected, a waveform is obtained that is fixed "H" or "L". Port output settings by MDOUT and the pin outputs obtained by PMD control register (MDCR) polarity settings are shown in Table 2-14.

MDOUTB bits 6 to 4 set the expected value for position signal of the position detection circuit.

When writing to MDOUTA/B registers, use 16-bit access instructions.

Table 2-14. Example of Pin Output Settings

Polarity: Active high (MDCR bits 4, 5 = 1)

MDOUTA Output Control		MDOUTB Bits 2, 1, 0 H/L or PWM Output Selection			
		0: H/L output		1: PWM output	
Bits 5, 3, 1 Upper phase	Bits 4, 2, 0 Lower phase	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	L	L	L	L
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{\text{PWM}}$

Polarity: Active low (MDCR bits 4, 5 = 0)

MDOUTA Output Control		MDOUTB Bits 2, 1, 0 H/L or PWM Output Selection			
		0: H/L output		1: PWM output	
Bits 5, 3, 1 Upper phase	Bits 4, 2, 0 Lower phase	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	H	H	H
0	1	H	L	H	$\overline{\text{PWM}}$
1	0	L	H	$\overline{\text{PWM}}$	H
1	1	L	L	$\overline{\text{PWM}}$	PWM

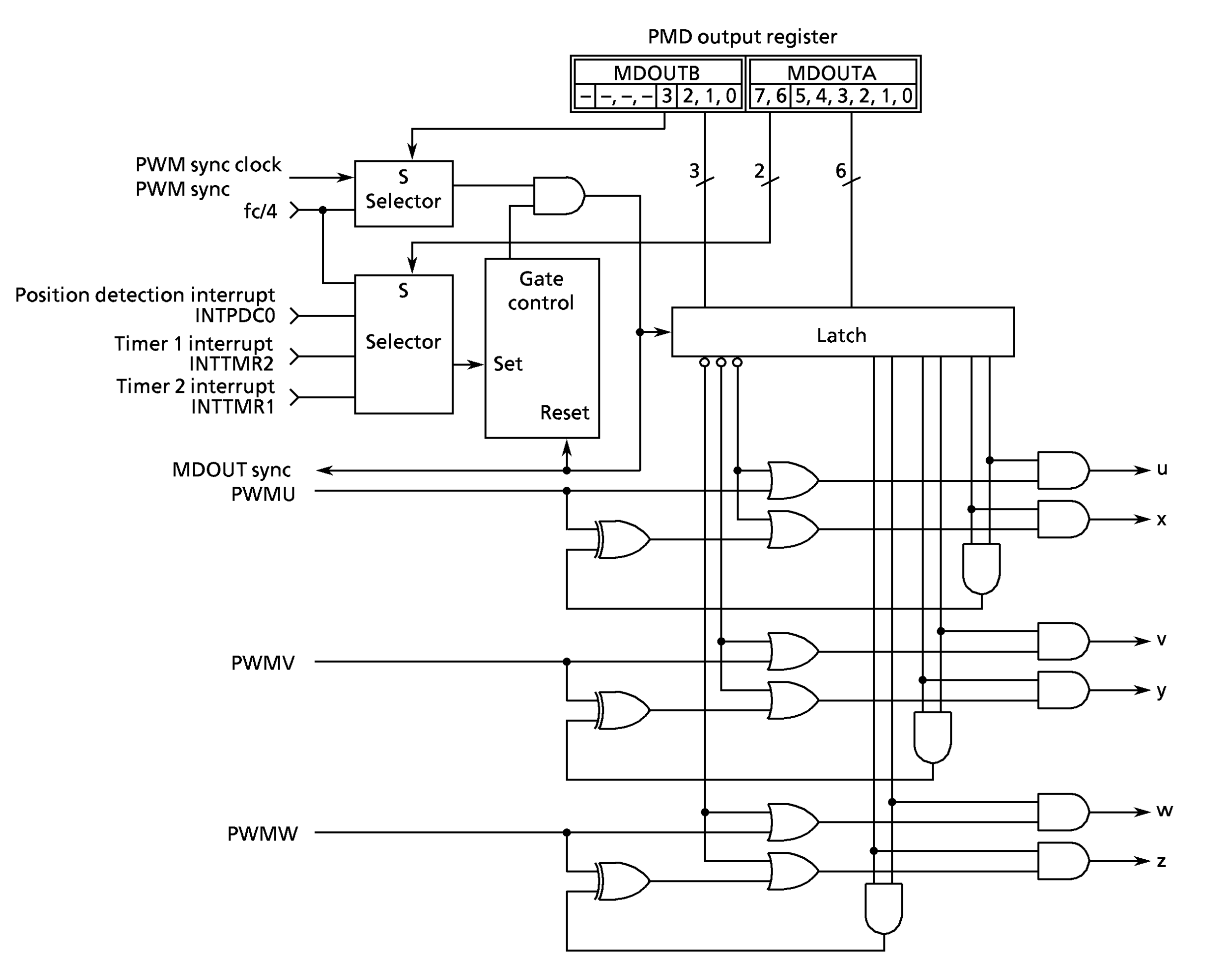


Figure 2-61. Commutation Control Circuit

③ Protective circuit

This circuit consists of an EMG protective circuit and an overload protective circuit. These circuits are actuated by assertion of their respective port input.

a) EMG protective circuit

This protective circuit is used for emergency stop, so that when an active transition (high to low) occurs on EMG input port, six lines of port output are immediately disabled and an EMG interrupt (INTEMG) is generated.

EMG protection is set by EMG control register (EMGCRA). When EMGCRA bit 1 indicates a "1" when read, it means that EMG protective circuit is in operation. When EMG protective circuit is operating, EMG protection can be restored by deasserting all port outputs and then setting EMGCRA bit 1 to "0". To disable EMG function, set $5A_H$ and $A5_H$ sequentially in EMG release register (EMGREL) and then EMGCRA bit 0 to "0".

b) Overload protective circuit

The overload protective circuit is set by EMG control registers (EMGCRA/B). To enable overload protection, set port CL/PDW for overload protection input by using EMGCRA bit 3 and then enable the overload protective circuit by using EMGCRB bit 0. This circuit is actuated by assertion ("H" → "L") of overload protection input.

Overload protection can be disabled by timer (EMGCRB bits 5, 4), by PWM sync (EMGCRB bit 6), or manually (EMGCRB bit 7). This, whichever method may be used, is effective only when overload protection input has been released back "H".

A sampling count can be set for overload protection input by using EMGCRA bits 7 to 4. The sampling count can be set in the range of 0 to 15 times a 250 ns period (@fc = 16 MHz).

To set the output-cut phases during overload protection, use EMGCRB bits 3 and 2. These bits allow selection of all phases, all upper/all lower phases, or no cut phase. When selection is made of all upper/all lower phases, port output is determined by the conducting state immediately before cut. If two or more upper phases were active, all upper phases turn on and all lower phases turn off; when less than two upper phases were active, all upper phases turn off and all lower phases turn on.

When port CL/PDW is set for W-phase position signal input (PDW) by EMGCRA bit 3, overload protection input is held "H".

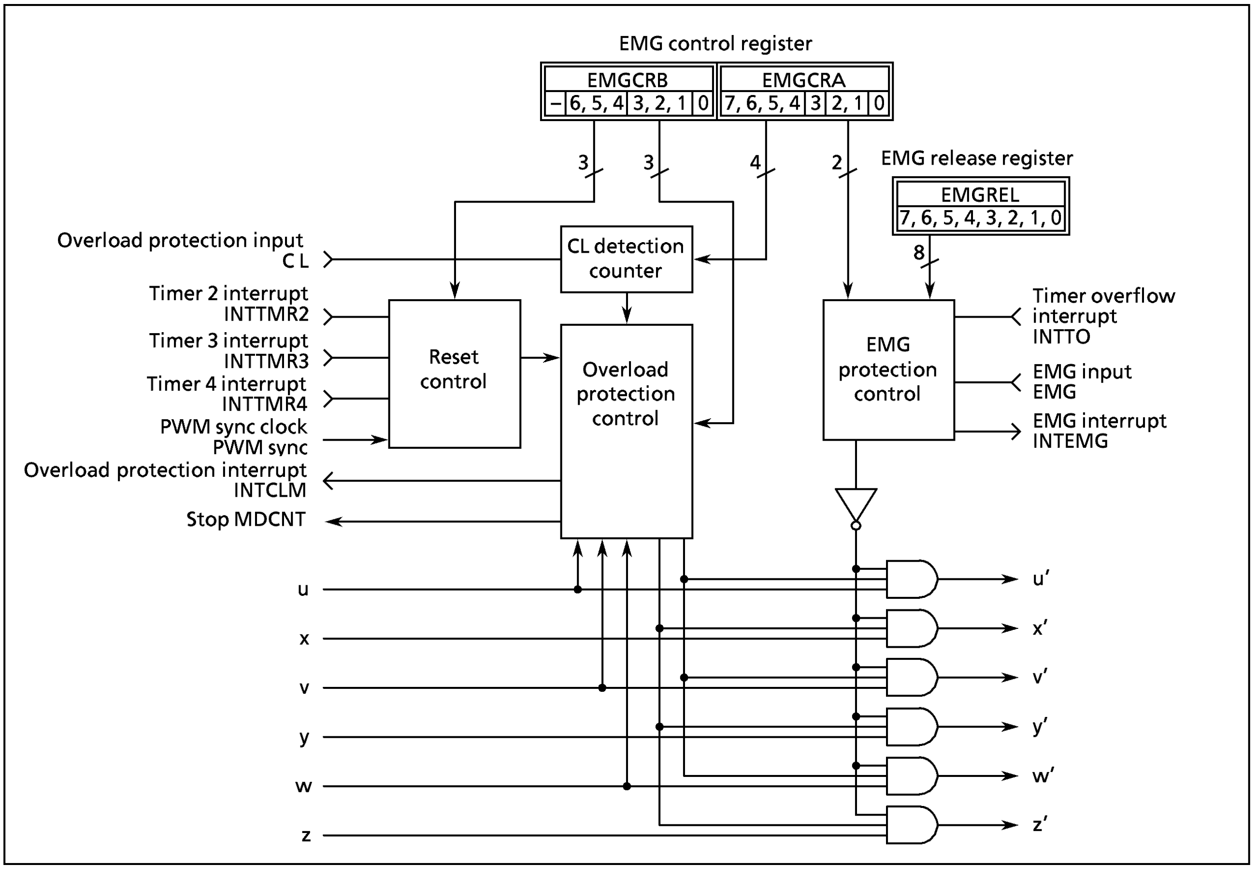


Figure 2-62. Protective Circuit

④ Dead time circuit

The dead time circuit consists of a dead time unit and an output polarity select unit.

The dead time counter has its turn-on time delayed for reasons of a possibility that when upper and lower phases reverse in any U, V, or W phase, upper and lower phases may be shorted. The delay time is set in increments of 250 ns (@fc = 16 MHz) by writing an 8-bit value to the dead time register (DTR).

The output polarity select circuit can set polarity (= active high or active low) independently for upper and lower phases by using PMD control register (MDCR) bits 4 and 5.

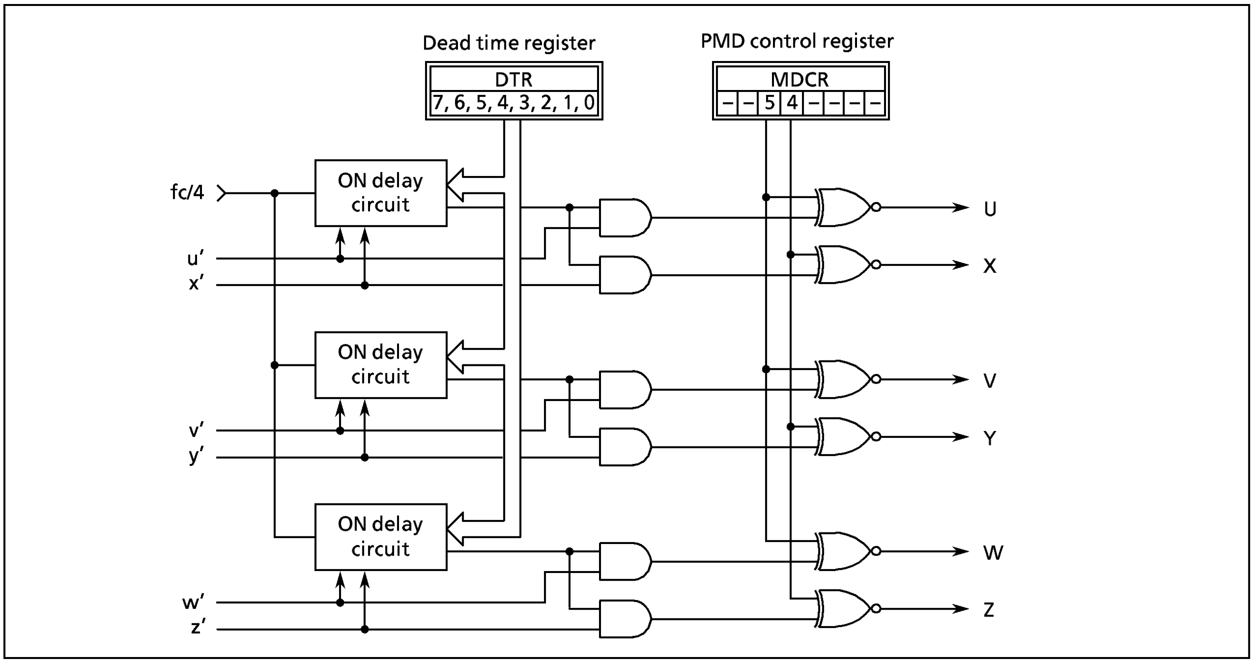


Figure 2-63. Dead Time Circuit

(2) Operation of the three-phase PWM output circuit

① Function list

Function	Command/Status		Operation
	Register Name	Bit Number	
Read PMD counter	MDCNTB MDCNTA	7 to 0 7 to 0	16-bit up/down counter with 125 ns (@fc = 16 MHz) resolution
Set PWM period	MDPRDB MDPRDA	7 to 0 7 to 0	Determines PWM period in 16 bits. Configured with double-buffers and updated at PWM period.
Set PWM pulse width	CMPUB CMPUA	7 to 0 7 to 0	16-bit register to set U-phase pulse width. Configured with double-buffers and updated at PWM period.
	CMPVB CMPVA	7 to 0 7 to 0	16-bit register to set V-phase pulse width. Configured with double-buffers and updated at PWM period.
	CMPWB CMPWA	7 to 0 7 to 0	16-bit register to set W-phase pulse width. Configured with double-buffers and updated at PWM period.
PMD counter up/ down state	MDCR	7	0: Up-count in progress 1: Down-count in progress
Select 3-phase output mode		6	Selects 3-phase independent mode or 3-phase common mode using U phase.
Output port polarity		5, 4	Sets port output polarity independently for upper and lower phases.
Select PWM interrupt generation period		3, 2	Selects INTPWM generation period from half or 1, 2, or 4 times the PWM period.
Select PWM mode		1	Selects sawtooth wave modulation or triangular wave modulation.
Enable/disable 3-phase PWM output circuit		0	Enables or disables 3-phase PWM output. When disabled, output ports are placed in the high-impedance state.
Mode compare register	MDOUTB	6, 5, 4	Sets the data to be compared with position detection input.
Set port output period		3	Selects whether or not synchronized to PWM period.
	MDOUTA	7, 6	Selects sync signal for port output from timer sync,position detection sync, or asynchronous.
Set dead time	DTR	7 to 0	Sets a delay in 250 ns increments (@fc = 16 MHz) using 8 bits.

Function	Command/Status		Operation
	Register Name	Bit Number	
Initial state of output port	–	–	• Ports are initially set for the motor control circuit. • When reset, ports go to a high-impedance state and remain in this state until the motor control circuit is enabled.
Restore from overload protection state	EMGCRB	7	Restored immediately by resetting this bit (= 0) while overload protection input = "H".
		6	Enables restoration by PWM sync when overload protection input = "H".
		5, 4	Enables restoration by timer when overload protection input = "H".
Select overload protection output disabled phase		3, 2	Selects output-disabled phase.
Stop PMD counter during overload protection		1	Stops PMD counter when overload protection is on.
Enable/disable overload protection		0	When enabled, INTCLM is generated by overload protection input. As EMG protection circuit is disabled, so is the overload protective function.
Set sampling count for overload protection input	EMGCRA	7 to 4	Can be set in the range of 0 to 15 times a 250 ns sampling period (@fc = 16 MHz). $125 + 250 \times (n - 1) <$ sampling time $< 125 + 250 \times n$
Select CL/PDW port function		3	Selects W-phase position signal input or overload protection input. When W-phase position signal input is selected, overload protection input is held "H".
Emergency stop for timer error		2	• INTTO is generated when the mode timer overflows or when capture overwrite is attempted, thereby actuating the protective circuit. • To determine the cause of output stopped, read EMGCR and MTCRB bits 4, 5.
EMG protection state		1	This bit is set to "1" during EMG protection.
Restore from EMG protection state	MDCR	0	Restored by setting MDOUTA bits 0 to 5 to 0 to deassert output and then setting EMGCRA bit 1 to "0".
Enable/disable EMG (emergency stop)	EMGCRA	0	• Enables or disables EMG protective circuit. However, before the circuit can be disabled, key code must be written to EMGREL. • Disables output in all phases when EMG signal is input, in which case output ports go to a high-impedance state.
	EMGREL	7 to 0	Made ready to get disabled by writing data in order of $5A_H$ and $A5_H$ and then disabled by setting EMGCRA bit 0 to "0". The key code is cleared when EMG has been disabled.

② Register list

MDCR (00F93$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

7	PMD counter status	0: Up-count in progress 1: Down-count in progress	R
6	Select 3-phase mode	0: 3-phase command mode. PWM waves common to three phases are generated using U-phase data. 1: 3-phase independent mode. PWM waves are generated independently for three phases.	R/W
5	Upper phase output port polarity (U/V/W output)	0: Active low 1: Active high	
4	Lower phase output port polarity (X/Y/Z output)	0: Active low 1: Active high	
3, 2	Select PWM interrupt generation period	00: PWM half period (valid for only PWM mode 1) 01: PWM × 1 period 10: PWM × 2 period 11: PWM × 4 period	
1	Select PWM mode	0: PWM mode 0 (sawtooth wave modulation) 1: PWM mode 1 (triangular wave modulation)	
0	Enable/disable 3-phase PWM output circuit	0: Disables 3-phase PWM output circuit. 1: Enables 3-phase PWM output circuit.	

Note: When PDCRB bit 4 has been set to "0" (sampled only when PWM is on), pay attention to the order in which to enable or disable the 3-phase PWM output circuit:

To enable, set MDCR bit 0 to 1 and then PDCRA bit 0 to 1.

To disable, set PDCRA bit 0 to 0 and then MDCR bit 0 to 0.

Figure 2-64. PMD Control Register

MDCNTB (00F97$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
MDCNTA (00F96$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0001)

16-bit up/down-counter for PWM Upper: MDCNTB, Lower: MDCNTA	0001$_H$ to FFFF$_H$: 16-bit up/down-counter with 125 ns resolution (@fc = 16 MHz)	R

Note: The register at address 00F96$_H$ varies depending on read or write.

Read: MDCNTA

Write: EMGREL

Figure 2-65. PMD Counter

CMPUB (00F99$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
CMPUA (00F98$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

Compare register to set U-phase pulse width Upper: CMPUB, Lower: CMPUA	0000$_H$ to FFFF$_H$: Configured with double-buffers and updated at PWM period.	R/W

CMPVB (00F9B$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
CMPVB (00F9A$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

Compare register to set V-phase pulse width Upper: CMPVB, Lower: CMPVA	0000$_H$ to FFFF$_H$: Configured with double-buffers and updated at PWM period.	R/W

CMPWB (00F9D$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
CMPWA (00F9C$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

Compare register to set W-phase pulse width Upper: CMPWB, Lower: CMPWA	0000$_H$ to FFFF$_H$: Configured with double-buffers and updated at PWM period.	R/W

Figure 2-66. PMD Compare Register

MDPRDB (00F9F$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)
MDPRDA (00F9E$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0001 0000)

Set PWM Upper: MDPRDB, Lower: MDPRDA	0010$_H$ to FFFF$_H$: Sets PWM period.	R/W

Figure 2-67. PMD Period Register

MDOUTB (00F95$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

6, 5, 4	Compare register for position detection	Set the expected values for U, V, and W-phase position signal input using bits 4, 5, and 6, respectively. When overload protection input has been enabled, set W phase to "0".	R/W
3	Select port output update clock	Selects port output updating clock. 0: fc/4 1: PWM synchronous	
2	W-phase PWM output	0: H/L output 1: PWM waveform output	
1	V-phase PWM output	0: H/L output 1: PWM waveform output	
0	U-phase PWM output	0: H/L output 1: PWM waveform output	

MDOUTA (00F94$_H$)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

7, 6	Select port output updating trigger	Selects the trigger to update port output after writing to PMD output register. 00: Enables updating immediately. 01: Enables updating by position detection. 10: Enables updating by timer 1. 11: Enables updating by timer 2.	R/W
5, 4	Control W-phase output	0: Output deasserted. 1: Output asserted.	
3, 2	Control V-phase output	0: Output deasserted. 1: Output asserted.	
1, 0	Control U-phase output	0: "L" 1: "H" (PWM output when MDOUTB (0) = 1) This applies when active-high.	

Note: When writing to MDOUTA/B registers, use 16-bit access instructions.

Figure 2-68. PMD Output Register

EMGCRB
($00F91_H$)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

7	Restore from overload protection state (Note 1)	1: Protective operation at work.	R
		0: Restores from protection state. 1: No operation.	W
6	Enable/disable restoration from overload protection state by PWM sync	0: Disables restoration. 1: Enables restoration.	R/W
5, 4	Restore from overload protection state by timer	00: Not restored by timer. 01: Restored by timer 2. 10: Restored by timer 3. 11: Restored by timer 4.	
3, 2	Select output-disabled phase when overload protection is on	00: No output-disabled phase. 01: Disables output in all phases. 10: Reserved. 11: All upper/all lower phases (Note 2)	
1	Stop PMD counter when overload protection is on	0: Counter does not stop. 1: Counter stops.	
0	Enable/disable overload protective circuit function	0: Disables circuit function. 1: Enables circuit function.	

Note 1: Before setting the register to restore from the overload protection state, wait until the CL detection time elapses after deasserting CL input.

Note 2: When two or more upper phases are on, all upper phases turn on and all lower phases turn off; when less than two upper phases are on, all upper phases turn off and all lower phases turn on.

EMGCRA
($00F90_H$)

7	6	5	4	3	2	1	0

(Initial value: 0001 0001)

7, 6, 5, 4	Overload protection sampling count	INTCLM is generated when consecutive occurrences are detected as many as the sampling counts set (0 to 15). 250 ns (@fc = 16 MHz) period.	R/W
3	Select CL/PDW port function	0: Sets the port for overload protection pin. In this case, W-phase position detection input is held "L". 1: Sets the port for W-phase position detection input pin.	
2	Stop output for mode timer error	0: Output not stopped by INTTO. 1: Enables protective circuit operation by INTTO.	
1	Restore from EMG protection state	1: Protective operation at work.	R
		0: Restores from protection state. 1: No operation.	W
0	Enable/disable EMG protective circuit function	0: Disables circuit function. 1: Enables circuit function.	R/W

EMGREL
($00F96_H$)

7	6	5	4	3	2	1	0

Key code input for exiting EMG protection	Can be exited by writing $5A_H$ and $A5_H$.	W

Note: The register at address $00F96_H$ varies depending on read or write.

Read: MDCNTA
Write: EMGREL

Figure 2-69. EMG Control Register and EMG Release Register

DTR
(00F92$_H$)

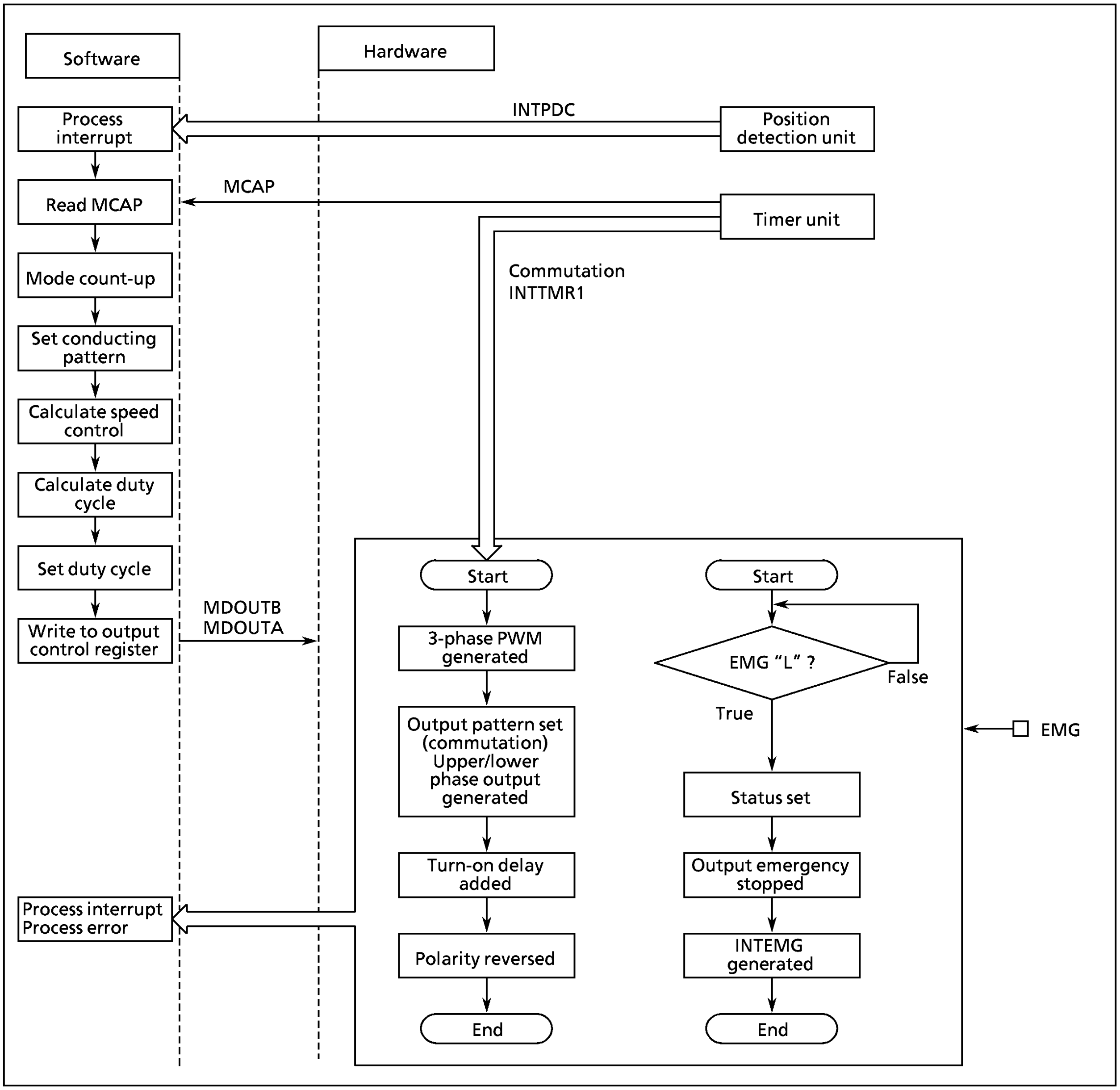
7	6	5	4	3	2	1	0

(Initial value: 1111 1111)

Set dead time	0000 0000 to 1111 1111: Sets an output turn-on delay. This can be set in increments of fc/4 clock using 8 bits.	R/W

Figure 2-70. Dead Time Register

(3) Outline of processing by the PWM output unit



2.12.6 Motor Control Processing

The following shows how to set registers for motor control by using a simple control flow as an example.

For sensorless control of DC motors, the rotor position is detected by an induced voltage. However, no sufficient induced voltage is generated until the rotor revolution reaches a certain speed. Startup requires a special procedure. First, the rotor position is identified by a d.c. current conduction. Current is flowed in conduction mode corresponding to the identified position, which is forcibly commutated to the next mode after a certain time by a timer interrupt. This is repeated until position detection becomes possible.

This control flow is created under conditions below:

Target:	Brushless DC motor
Drive method:	Square wave 120-degree conduction
Control method:	Sensorless control (by detecting induced voltage in motor)

Example of using internal PMD timers

Timer 1: Used for commutation timing, set to 30 degrees.

Timer 2: Used for position detection start timing, set to 45 degrees.

Timer 3: Used for forced commutation/error determination, set to maximum timer value ($FFFF_H$).

Timer 4: Unused.

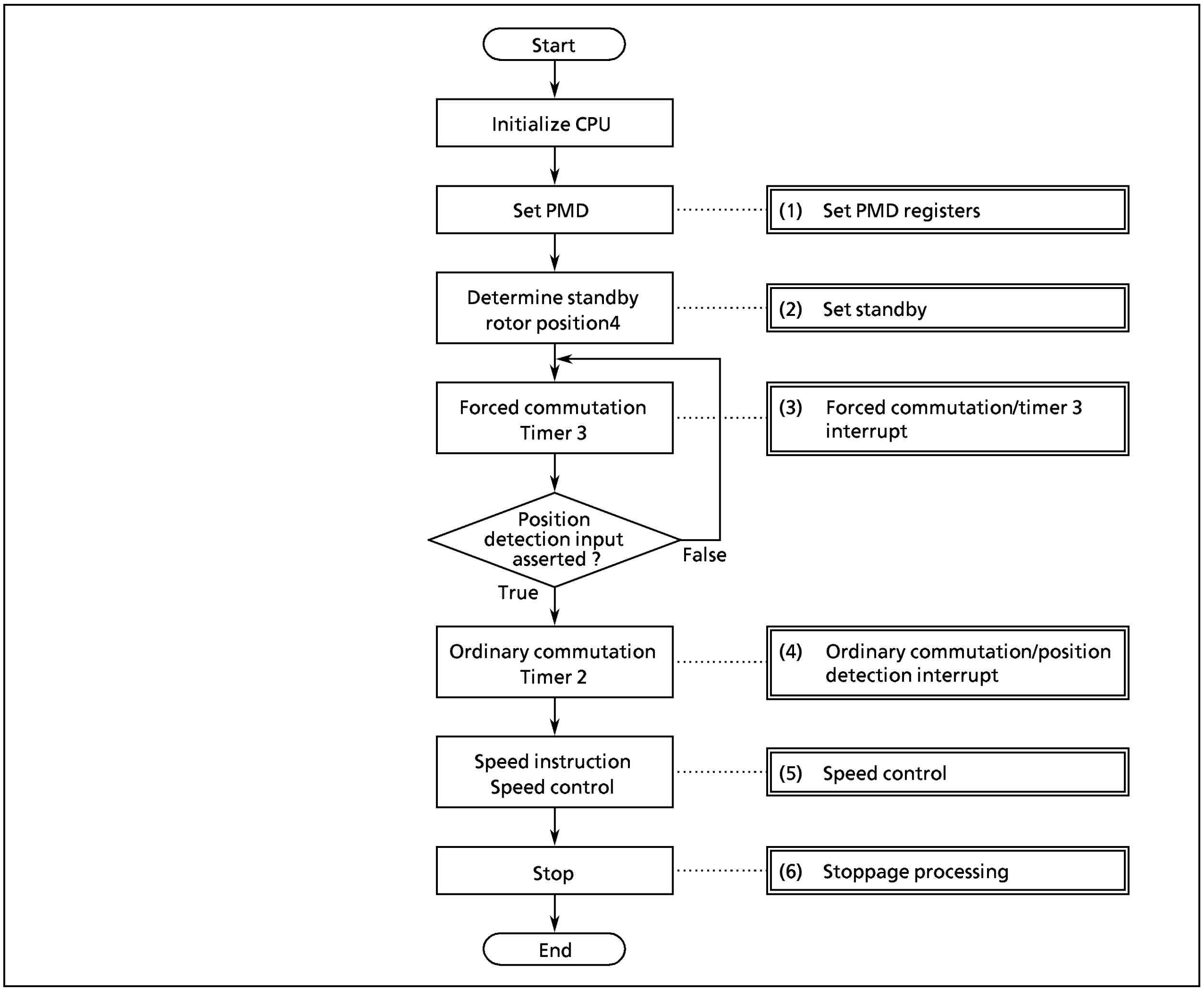

Start
Initialize CPU
Set PMD
(1) Set PMD registers
Determine standby rotor position4
(2) Set standby
Forced commutation Timer 3
(3) Forced commutation/timer 3 interrupt
Position detection input asserted ?
False
True
Ordinary commutation Timer 2
(4) Ordinary commutation/position detection interrupt
Speed instruction Speed control
(5) Speed control
Stop
(6) Stoppage processing
End

Figure 2-71. Motor Control Flow

(1) Setting PMD registers

① Set 0000$_H$ in PMD output register (MDOUT)

② Set position detection sampling delay, sampling delay register (SDREG)

③ Set mode timer control, mode timer control registers (MTCRA, MTCRB)

④ Set compare control, compare control register (CMPCR)

⑤ Set error detection timer, compare register 3 (CMP3)

⑥ Initialize PWM drive, PMD control register (MDCR)

⑦ Set dead time, dead time register (DTR)

⑧ Initialize EMG, EMG control register (EMGCR)

⑨ Set PWM carrier period, PMD period register (MDPRD)

⑩ Set pulse width, PMD compare register 2 (CMPU)

⑪ Set position detection control, position detection mode timer control register (PDCR)

(2) Setting standby

① Initialize speed buffer (set in RAM by software)
② Set 0000_H in PMD output register (MDOUT)
③ Reset EMG, clear EMG control register (EMGCR) bit 1
④ Select initial output pattern, PMD output register (MDOUT)
⑤ Set commutation timing $FFFF_H/2 = 8000_H$, set compare register (CMP1)
⑥ Set position detection start timing $FFFF_H \times 3/4 = C000_H$, set compare register (CMP2)
⑦ Set PWM carrier period, set maximum period in PMD period register (MDPRD)
⑧ Set PWM pulse width, set forced commutation initial pulse width in PMD compare register (CMPU)
⑨ Read mode capture register (MCAP) (dummy read)
⑩ Enable mode timer, set mode timer control register (MTCRA) bit 0

(3) Forced commutation/timer 3 interrupt

① Save buffer
② Update conduction and position detection patterns, set PMD output register (MDOUT)
③ Set commutation timing 8000_H, set compare register 1 (CMP1)
④ Set position detection start timing $C000_H$, set compare register 2 (CMP2)
⑤ Set PWM carrier wave period, set PMD period register (MDPRD)
⑥ Set PWM pulse width, set PMD compare register (CMPU)

(4) Ordinary commutation/position detection interrupt

① Read mode capture register (MCAP)
② Save mode capture value in speed buffer
③ Update conduction and position detection patterns, set PMD output register (MDOUT)
④ Calculate commutation timing, set compare register 1 (CMP1)
⑤ Calculate position detection start timing, set compare register 2 (CMP2)

(5) Speed control

① Calculate average period (speed) from speed buffer value
② Calculate difference with target period (speed)
③ Calculate pulse width
④ MIN/MAX limits
⑤ Select optimum values for pulse width and PWM period
⑥ Set PWM carrier period, set period in PMD period register (MDPRD)
⑦ Set PWM pulse width, set it in PMD compare register (CMPU)

(6) Stoppage processing

① Set 0000_H in PMD output register (MDOUT)
② Set 0000_H in PMD compare register (CMPU)
③ Set 00_H in mode timer control register (MTCRA)
④ Set $FFFF_H$ for speed control

2.12.7 List of PMD Control Related Registers

(1) Input/Output Pins and Input/Output Control Registers

PMD1 Input/Output Pins (P0, P1) and Port Input/Output Control Registers (P0CR, P1CR)

Name	Address	Bit	R or W	Description
P0	00000_H	7	Read	Overload protection/W1 position detection input ($\overline{\text{CL1}}$/PDW1).
		6	Read	EMG input ($\overline{\text{EMG1}}$).
		5 to 0	R/W	U1/V1/W1/X1/Y1/Z1 outputs.
P1	00001_H	7, 6	Read	U1/V1 position signal inputs (PDU1, PDV1).
P0CR	$0000A_H$	7 to 0	Write	P0 port input/output control (can be specified bitwise). 0: Input mode; 1: Output mode.
P1CR	$0000B_H$	7, 6	Write	P1 port input/output control (can be specified bitwise). 0: Input mode; 1: Output mode.

(2) Control Registers of Motor Control Circuit

Position Detection Control Registers (PDCRA/B) and Sampling Delay Register (SDREG).

Name	Address	Bit	R or W	Description
PDCRB	$00F81_H$	5	R/W	Enables/disables matching count recount when PWM is on. 0: Disable, 1: Enable
		4	R/W	Switch between analog and digital sampling. 0: Digital (valid only when PWM signal is on) 1: Analog (successive sampling)
		3 to 0	R/W	Position signal matching count (1 to 15)
PDCRA	$00F80_H$	7, 6	R/W	Selects sampling input clock [Hz]. 00: $fc/2^2$, 01: $fc/2^3$ 10: $fc/2^4$, 11: $fc/2^5$.
		5, 4	R/W	Sampling stop condition: Selected from timers below. 00: Disable, 01: Timer 2 10: Timer 3, 11: Timer 4.
		3, 2	R/W	Sampling start condition: Selected from timers below. 00: Disable, 01: Timer 2 10: Timer 3, 11: Timer 4.
		1	Write	Starts or stops sampling. 0: Stops sampling. 1: Starts sampling.
			Read	0: Idle, 1: Sampling in progress.
		0	R/W	Enables/disables position detection function. 0: Disable, 1: Enable.
SDREG	$00F82_H$	7 to 0	R/W	Sampling delay register. Count period 250 ns @fc = 16 MHz.

Mode Timer Control Register (MTCR), Compare Control Register (CMPCR), Mode Capture Register (MCAP), Compare Registers (CMP1, CMP2, CMP3, CMP4)

Name	Address	Bit	R or W	Description
MTCRB	00F85$_H$	5	Read	Mode timer overflow. 0: Not overflowed, 1: Overflowed.
		4	Read	Mode timer capture status. 0: Already read, 1: Not read yet.
		3	R/W	Mode timer capture condition: Overload protection 0: Disable, 1: Enable.
		2	Write	Captures mode timer. 1: Capture. 0: No operation.
			Read	Always “0” when read.
		1	R/W	Mode timer capture condition: Position detection 0: Disable, 1: Enable.
		0	R/W	Reenables/disables capture automatically. (Capture register overwrite) 0: Disable, 1: Enable.
MTCRA	00F84$_H$	7, 6	R/W	Selects mode timer clock [Hz]. 00: $fc/2^2$, 01: $fc/2^3$ 10: $fc/2^4$, 11: $fc/2^5$.
		5, 4	R/W	Mode timer reset condition: Timers below. 00: Inhibited, 01: Timer 2 10: Timer 3, 11: Timer 4.
		3	R/W	Mode timer reset condition: Overload protection 0: Disable, 1: Enable.
		2	Write	Resets mode timer. 1: Reset. 0: No operation.
			Read	Always “0” when read.
		1	R/W	Mode timer reset condition: Position detection 0: Disable, 1: Enable.
		0	R/W	Enables/disables mode timer. 0: Disable, 1: Enable.
CMPCR	00F83$_H$	7	R/W	Reenables/disables compare register 4. 0: Disable, 1: Reenable.
		6	R/W	Enables/disables compare register 4. 0: Disable, 1: Enable.
		5	R/W	Reenables/disables compare register 3.
		4	R/W	Enables/disables compare register 3.
		3	R/W	Reenables/disables compare register 2.
		2	R/W	Enables/disables compare register 2.
		1	R/W	Reenables/disables compare register 1.
		0	R/W	Enables/disables compare register 1.
MCAP	00F87$_H$ to 00F86$_H$	7 to 0	Read	Mode capture register.
CMP1	00F89$_H$ to 00F88$_H$	7 to 0	R/W	Compare register 1.
CMP2	00F8B$_H$ to 00F8A$_H$	7 to 0	R/W	Compare register 2.
CMP3	00F8D$_H$ to 00F8C$_H$	7 to 0	R/W	Compare register 3.
CMP4	00F8F$_H$ to 00F8E$_H$	7 to 0	R/W	Compare register 4.

PMD Control Register (MDCR), PMD Output Register (MDOUT), Dead Time Register (DTR)

Name	Address	Bit	R or W	Description
MDCR	00F93$_H$	7	Read	PMD counter up/down status. 0: Up-count, 1: Down-count.
		6	R/W	AC/DC modes (3-phase independent, 3-phase common modes). 0: DC mode, 1: AC mode.
		5	R/W	Upper phase (U, V, W) output port polarity. 0: Active low, 1: Active high.
		4	R/W	Lower phase (X, Y, Z) output port polarity. 0: Active low, 1: Active high.
		3, 2	R/W	Selects PWM interrupt (INTPWM1) generation period. 00: PWM half period (valid for only PWM mode 1). 01: PWM x1 period. 10: PWM x2 period. 11: PWM x4 period.
		1	R/W	PWM mode (sawtooth wave/triangular wave modulation). 0: PWM mode 0, 1: PWM mode 1.
		0	R/W	PMD Output ENABLE. 0: Disables PWM output circuit, 1: Enables.
MDOUTB	00F95$_H$	6 to 4	R/W	Comparison register for position detection. 6: W, 5: V, 4: U.
		3	R/W	Select PWM sync (port output sync signal selection 2). 0: Asynchronous to PWM period, 1: Synchronized.
		2 to 0	R/W	W, V, U-phase PWM outputs. 0: H/L output, 1: PWM waveform output.
MDOUTA	00F94$_H$	7, 6	R/W	Selects port output sync signal. 00: Asynchronous (directly output). 01: Synchronized to position detection. 10: Synchronized to timer 1. 11: Synchronized to timer 2.
		5 to 0	R/W	W, Z, V, Y, U, X output control.
DTR	00F92$_H$	7 to 0	R/W	Dead time register count period. Unit: 250 ns @fc = 16 MHz.

PMD Counter (MDCNT), PMD Period Register (MDPRD), PMD Compare Registers (CMPU, CMPV, CMPW)

Name	Address	Bit	R or W	Description
MDCNT	00F97$_H$ to 00F96$_H$	7 to 0	Read	16-bit up/down-counter for PWM.
MDPRD	00F9F$_H$ to 00F9E$_H$	7 to 0	R/W	Register to set PWM period.
CMPU	00F99$_H$ to 00F98$_H$	7 to 0	R/W	Register to set U-phase PWM duty cycle.
CMPV	00F9B$_H$ to 00F9A$_H$	7 to 0	R/W	Register to set V-phase PWM duty cycle.
CMPW	00F9D$_H$ to 00F9C$_H$	7 to 0	R/W	Register to set W-phase PWM duty cycle.

EMG Release Register (EMGREL), EMG Control Register (EMGCR)

Name	Address	Bit	R or W	Description
EMGREL	00F96$_{H}$	7 to 0	Write	Key code input for exiting EMG protection.
EMGCRB	00F91$_{H}$	7 to 0	Write	Restores from current limiting state. 0: Restores from current limiting state. 1: No operation.
			Read	1: Current liming at work.
		6	R/W	Condition for restoring from current limiting state: PWM sync. 0: Disable, 1: Enable.
		5, 4	R/W	Condition for restoring from current limiting state: Timers below. 00: Inhibited, 01: Timer 2 10: Timer 3, 11: Timer 4.
		3, 2	R/W	Select output-disabled phase when overload protection is on. 00: No output-disabled phase. 01: Disables output in all phases. 10: Reserved. 11: All upper/all lower phases.
		1	R/W	Stops PMD counter (MDCNT) when overload protection is on. 0: Does not stop, 1: Stops.
		0	R/W	Enables/disables overload protection circuit function. 0: Disables circuit function, 1: Enables.
EMGCRA	00F90$_{H}$	7 to 4	R/W	Overload protection sampling counts (1 to 15). Sampling clock 250 ns @fc = 16 MHz.
		3	R/W	Selects $\overline{\text{CL}}$/PDW pin function. 0: Overload protection ($\overline{\text{CL}}$). 1: Position detection (PDW).
			R/W	EMG protective function by mode timer error. 0: Disable, 1: Enable.
		1	Write	Resets EMG protection state. 0: Resets EMG protection state. 1: No operation.
			Read	1: Protection at work.
		0	R/W	EMG protective function by $\overline{\text{EMG}}$ pin. 0: Disable, 1: Enable. (Before this function can be disabled, key code 5A$_{H}$, A5$_{H}$ must be written to EMGREL1.)

2.13 UART (Asynchronous serial interface)

TMP88CS48A features a built-in channel for UART (asynchronous serial interface). UART is connected to external devices via RxD and TxD. RxD is also used as P43; TxD, as P44. To use P43 or P44 as the RxD or TxD pin, set P4 port output latches to 1.

2.13.1 Configuration

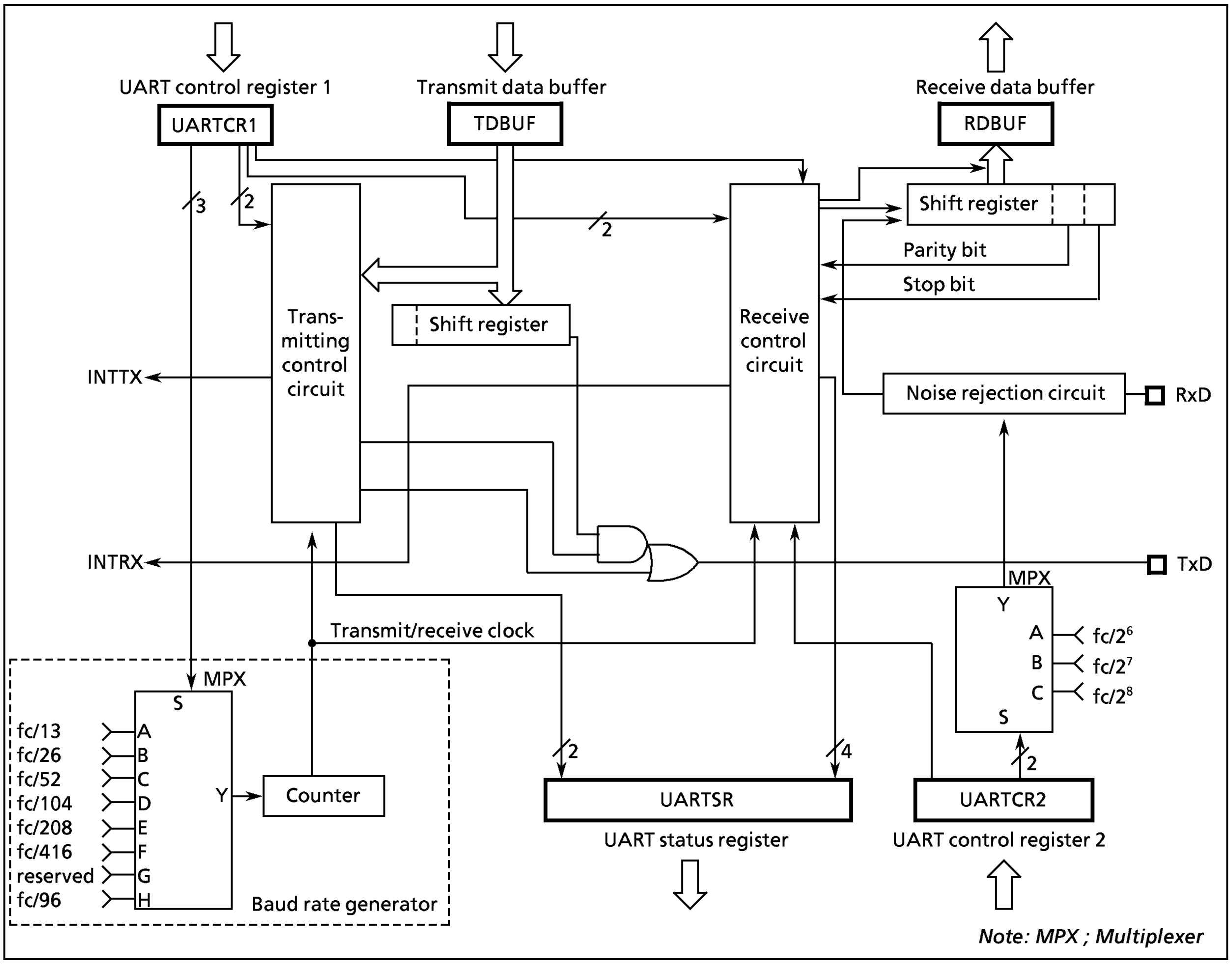


Figure 2-72. UART (Asynchronous serial interface)

2.13.2 Control

UART is controlled by the UART control register (UARTCR). The operating status can be monitored using the UART status register (UARTSR).

UARTCR1
(00FFC$_H$)

7	6	5	4	3	2	1	0
TXE	RXE	STBT	EVEN	PE	BRG		

(Initial value: 0000 0000)

BRG	Transfer clock select	000: fc/13 [Hz] 001: fc/26 010: fc/52 011: fc/104 100: fc/208 101: fc/416 110: reversed 111: fc/96	Write only
PE	Parity addition	0: Without parity 1: Parity added	
EVEN	Even parity	0: Odd parity 1: Even parity	
STBT	Transmit stop bit length	0: 1 bit 1: 2 bits	
RXE	Receive operation	0: Disable 1: Enable	
TXE	Transmit operation	0: Disable 1: Enable	

Note 1: When transmitting or receiving operation is implemented, setting TXE and RXE bit to "0" inhibits the operation. When transmitting is disabled, the transfer data stored in the transfer data buffer is not output. Even if the transmit enable is set, transmitting is not operated until writing new data.

Note 2: The transfer clock and the parity are used for transmitting and receiving.

UARTCR2
(00FFD$_H$)

7	6	5	4	3	2	1	0
					RXDNC		STOPBR

(Initial value: **** *000)

STOPBR	Receive stop bit length	0: 1 bit 1: 2 bits	Write only
RXDNC	Selection of RXD input noise rejection time	00: No noise cancellation (hysteresis input) 01: Cancels less than 31/fc [s] as noise. 10: Cancels less than 63/fc [s] as noise. 11: Cancels less than 127/fc [s] as noise.	

Note 3: 96/fc or more when RXDNC = 01, 192/fc or more when RXDNC = 10, 384/fc or more when RXDNC = 11 are regarded as signals.

Figure 2-73. UART Control Register

UART status register

	7	6	5	4	3	2	1	0	
UARTSR (00FFC$_H$)	PERR	FERR	OERR	RBFL	TEND	TBEP			(Initial value: 0000 11**)

TBEP	Transmit data buffer empty flag	0: – 1: Transmit buffer empty	Read only
TEND	Transmit end flag	0: Under transmission 1: End of transmission	
RBFL	Receive buffer full flag	0: – 1: Receive buffer full	
OERR	Overrun error flag	0: Without overrun error 1: Overrun error occurrence	
FERR	Framing error flag	0: Without framing error 1: Framing error occurrence	
PERR	Parity error flag	0: Without parity error 1: Parity error occurrence	

UART receive data buffer

	7	6	5	4	3	2	1	0	
RDBUF (00FFB$_H$)									Read only (Initial value: 0000 0000)

UART transmit buffer

	7	6	5	4	3	2	1	0	
TDBUF (00FFB$_H$)									Write only (Initial value: 0000 0000)

Figure 2-74. UART Status Register and Data Buffer Registers

2.13.3 Transfer Data Format

With UART, a one-bit start bit ("L" level), stop bit (bit length selectable at "H" level, in STBT), and parity (select parity in PE; even- or odd-numbered parity in EVEN) are added to the data to be transferred. Shown below are the transfer data formats.

Table 2-15. Transfer Data Formats

PE	STBT	Frame length 1	2	3	-------	8	9	10	11	12
0	0	Start	bit 0	bit 1	-----	bit 6	bit 7	Stop 1		
0	1	Start	bit 0	bit 1	-----	bit 6	bit 7	Stop 1	Stop 2	
1	0	Start	bit 0	bit 1	-----	bit 6	bit 7	Parity	Stop 1	
1	1	Start	bit 0	bit 1	-----	bit 6	bit 7	Parity	Stop 1	Stop 2

2.13.4 Baud Rate

Set the UART transfer rate (baud rate) in BRG (bits 0, 1, and 2 in UARTCR). Listed below are the transfer rates.

Table 2-16. Baud Rate (example)

BRG	Source clock		
	16 MHz	8 MHz	4 MHz
000	76800 [baud]	38400 [baud]	19200 [baud]
001	38400	19200	9600
010	19200	9600	4800
011	9600	4800	2400
100	4800	2400	1200
101	2400	1200	600

2.13.5 Data Sampling Method

The UART receiver keeps sampling input until a start bit is detected in the RxD pin input, using the clock selected by BRG (bits 0, 1, and 2 in UARTCR). RT clock detects the falling edge of the RxD pin and start. Once a start bit is detected, the start bit, data bits, stop bit (s), and parity bit are sampled three times at 7RT, 8RT, and 9RT during one receiver clock interval (RT1 clock). (RT0 is the position where the bit supposedly starts.) Bits are determined according to majority rule (the data are the same twice or more out of three samplings).

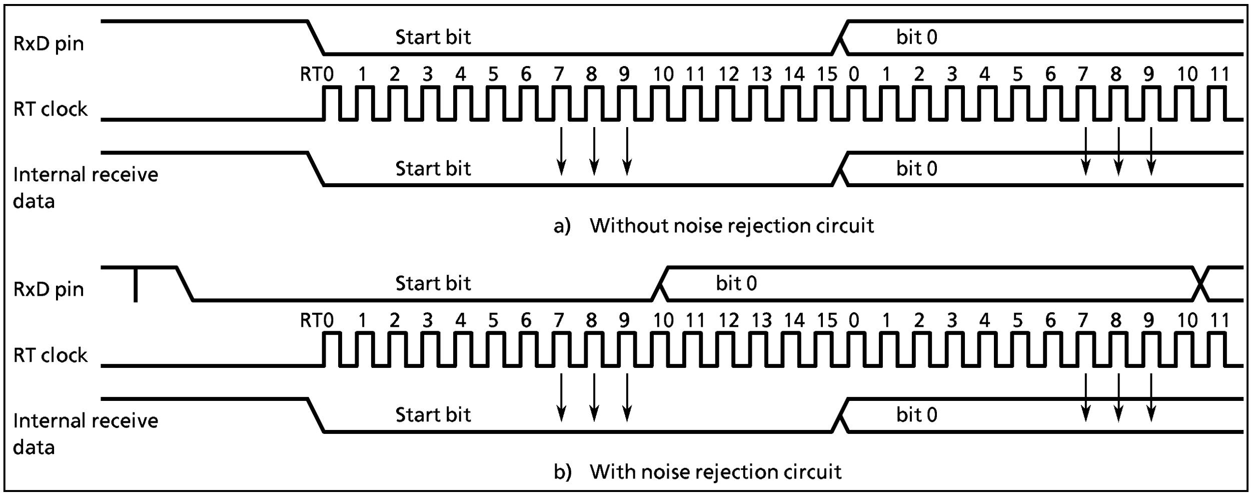

RxD pin
Start bit
bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
bit 0
a) Without noise rejection circuit
RxD pin
Start bit
bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
bit 0
b) With noise rejection circuit

Figure 2-75. Data Sampling Method

2.13.6 STOP Bit Length

Select a transmit stop bit length (1 or 2 bits) in STBT (bit 5 in UARTCR).

2.13.7 Parity

Set parity/no parity in PE; set parity type (odd- or even-numberred) in EVEN (bit 4 in UARTCR).

2.13.8 Transmit/Receive Operation

(1) **Data transmit operation**

Set TXE (bit 7 in UARTCR) to 1. Read UARTSR to check TBEP = 1, then write data in TDBUF (transmit data buffer). Writing data in TDBUF zero-clears TBEP, transfers the data to the transmit shift register and the data are sequentially output from the TxD pin. The data output include a one-bit start bit, stop bits whose number is specified in STBT (bit 5 in UARTCR), and a parity bit if parity addition is specified. Select the data transfer baud rate using bits 0 to 2 in UARTCR. When data transmit starts, transmit buffer empty flag TBEP is zero-cleared and interrupt INTTX is generated.

When transmitting data, first read UARTSR, then write data in TDBUF. Otherwise, TBEP is not zero-cleared and transmit does not start.

(2) **Data receive operation**

Set RXE (bit 6 in UARTCR) to 1. When data are received via the RxD pin, the receive data are transferred to RDBUF (receive data buffer). At this time, the data transmitted include a start bit and stop bit (s), and a parity bit if parity bit if parity addition is specified. When stop bit (s) are received, data only are extracted and transferred to RDBUF (receive data buffer). Then the receive buffer full flag RBFL is set and interrupt INTRX is generated. Select the data transfer baud rate using BRG (bits 0 to 2 in UARTCR).

If an overrun error (OERR) occurs when data are received, the data are not transferred to RDBUF (receive data buffer) but discarded; data in the RDBUF are not affected.

2.13.9 Status Flag

(1) **Parity error**

When parity determined using the receive data bits differs from the received parity bit, parity error flag PERR is set in UARTSR. Reading UARTSR the RDBUF clears PERR.

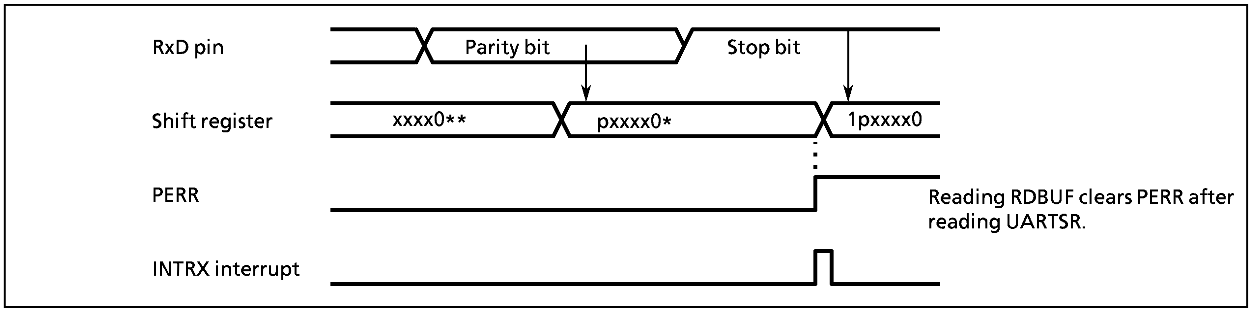


Figure 2-76. Parity Error Occurrence

(2) **Framing error**

When 0 is sampled as the stop bit in the receive data, framing error flag FERR is set. Reading UARTSR then RDBUF clears FERR.

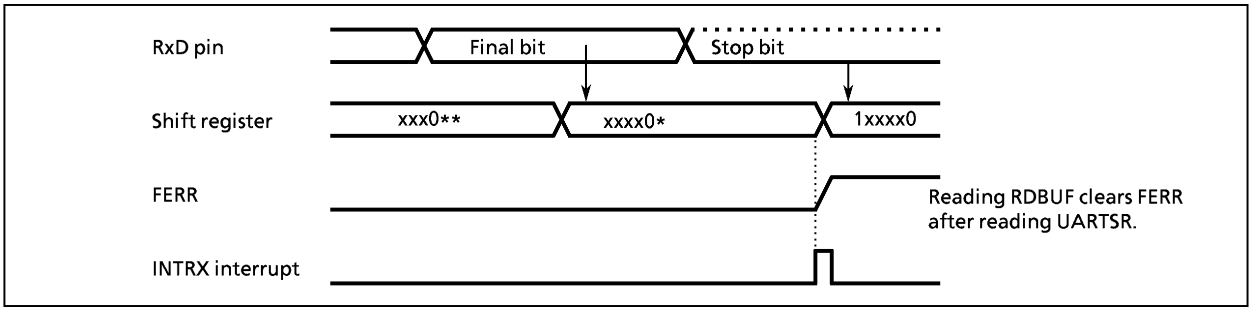


Figure 2-77. Framing Error Occurrence

(3) **Overrun error**

When all bits in the next data are received while unread data are still in RDBUF, overrun error flag OERR is set. The receive data are discarded; data in RDBUF are not affected. Reading UARTSR then RDBUF clears OERR.

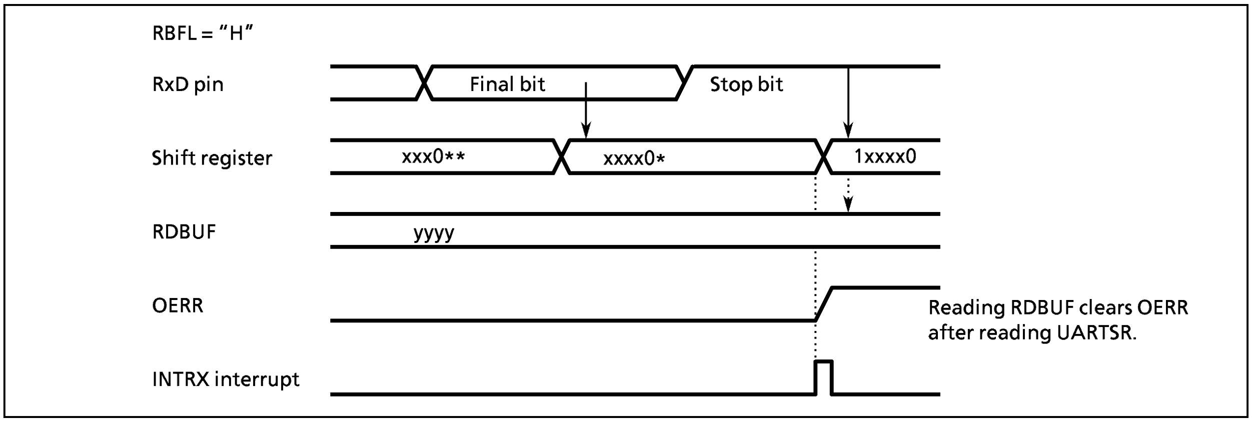


Figure 2-78. Overrun Error Occurrence

(4) **Receive buffer full**

Loading the receive data in RDBUF sets receive data buffer full flag RBFL. Reading UARTSR then RDBUF clears RBFL.

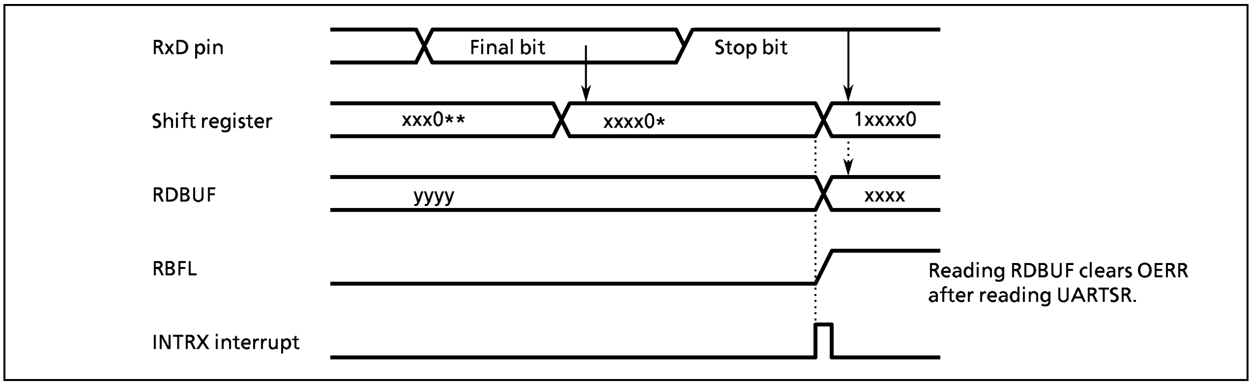


Figure 2-79. Receive Buffer Full Occurrence

(5) **Transmit data buffer empty**

When there are no data in transmit data buffer TDBUF, that is, when data in TDBUF are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag TBEP is set. Reading UARTSR then writing the data to TDBUF clears TBEP.

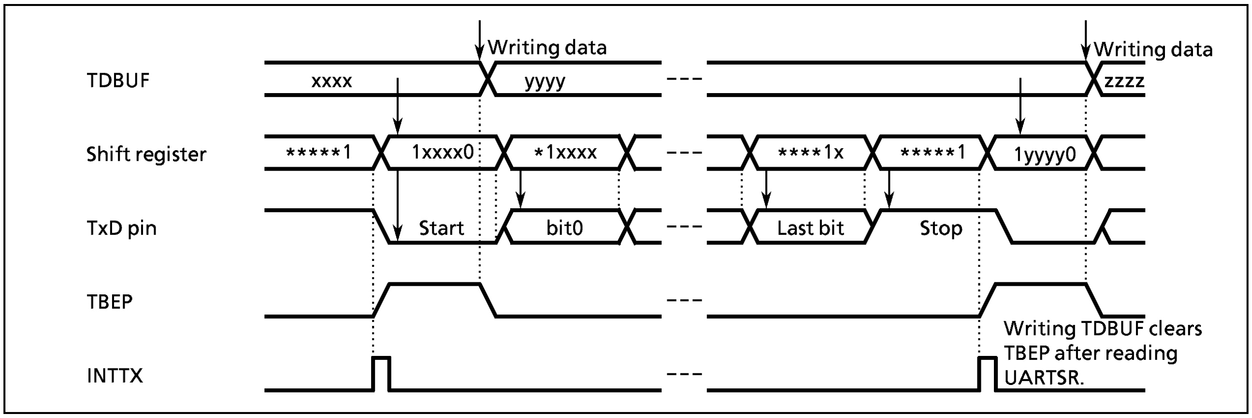


Figure 2-80. Transmit Buffer Empty Occurrence

(6) **Transmit end flag**

When data are transmitted and there are no data in TDBUF (TBEP = 1), transmit end flag TEND is set. Writing data to TDBUF then starting data transmit clears TEND.

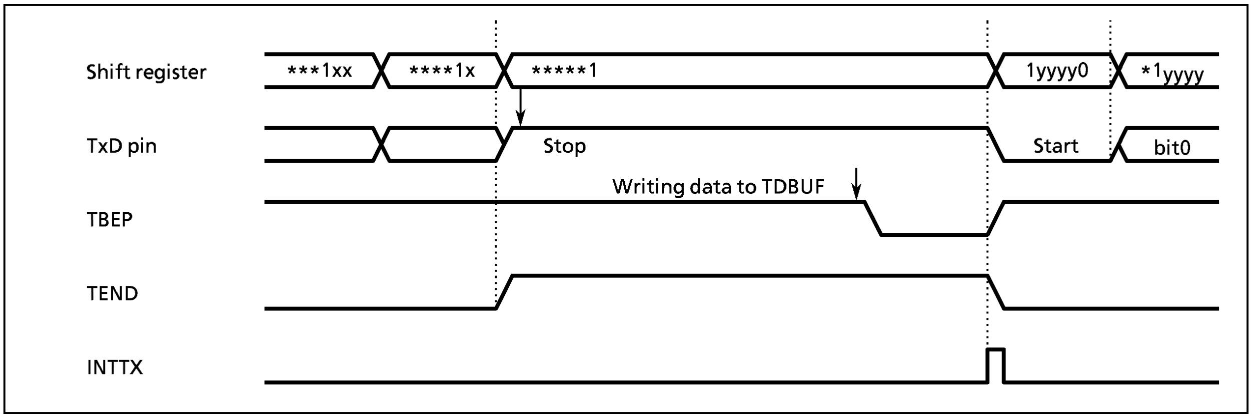


Figure 2-81. Transmit End Flag

2.14 Serial Bus Interface (SBI-ver.A)

The 88CS48A has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit serial bus interface and an I^{2}C bus.

The serial bus interface is connected to an external device through P47 (SDA) and P46 (SCL) in the I^{2}C bus mode; and through P45 ($\overline{\text{SCK}}$), P47 (SO) and P46 (SI) in the clocked-synchronous 8-bit SIO mode.

The serial bus interface pins are also used as the port. When used as serial bus interface pins, set the P4 output latches of these pins to "1". When not used as serial bus interface pins, the P4 port is used as a normal I/O port.

I^{2}C bus has no an arbitration function which is necessary when two or more master devices scramble for the bus control. In master mode, other devices which are connected on the same bus need be slave devices. (single master)

Note: When a multi master I^{2}C bus system operates in I^{2}C bus mode of this serial bus interface circuit, there is a possibility that the following problems raise. I^{2}C bus mode of this serial bus interface circuit should be used by a single master I^{2}C bus system.

1. *The SCL line is fixed to "L" level and transferring stops by the serial bus interface circuit. The other devices can not run on the SCL line. Thus the bus locks.*
2. *The SCL pin is pulled down to "L" level regardless of the state of the SCL line by the serial bus interface circuit. A period of high-level SCL clock pulse which other devices output is shortened. The minimum value of which the SCL clock holds "H" level is not satisfied, which is specified with the I^{2}C bus standard.*

2.14.1 Configuration

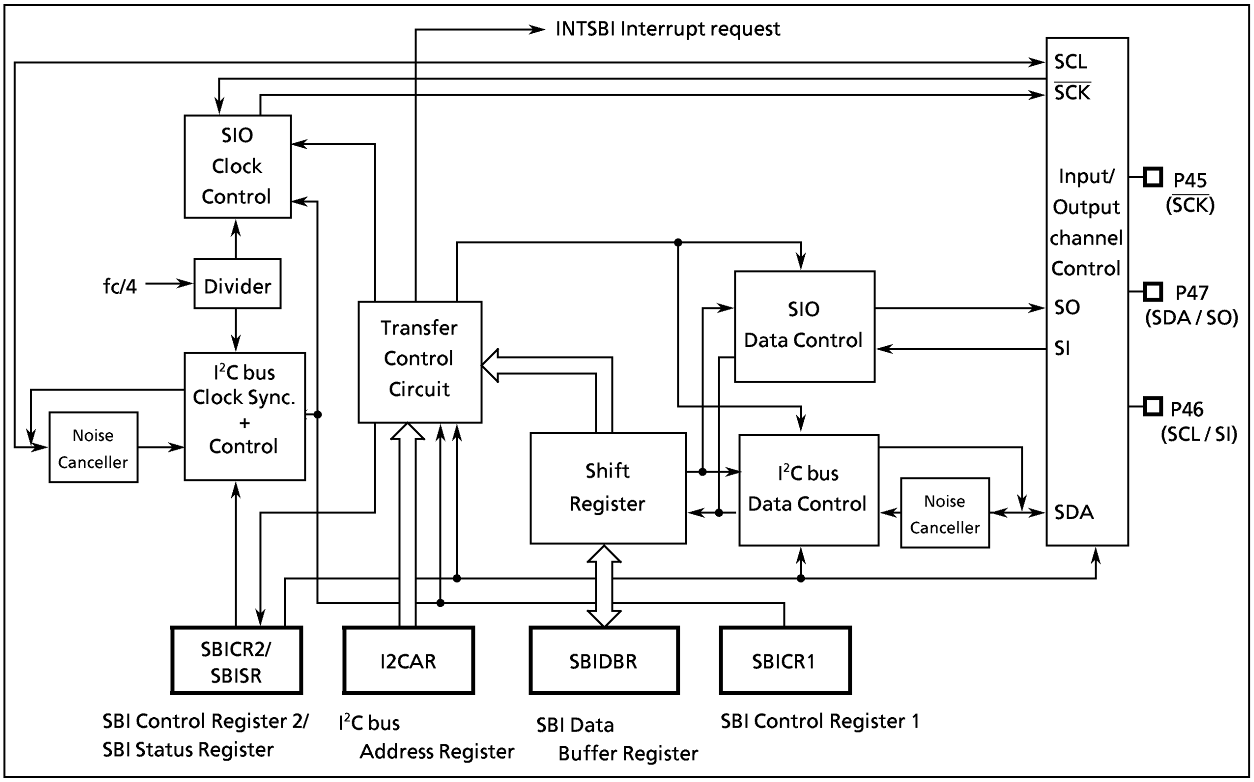


Figure 2-82. Serial Bus Interface (SBI-ver.A)

2.14.2 Serial Bus Interface (SBI-ver.A) Control

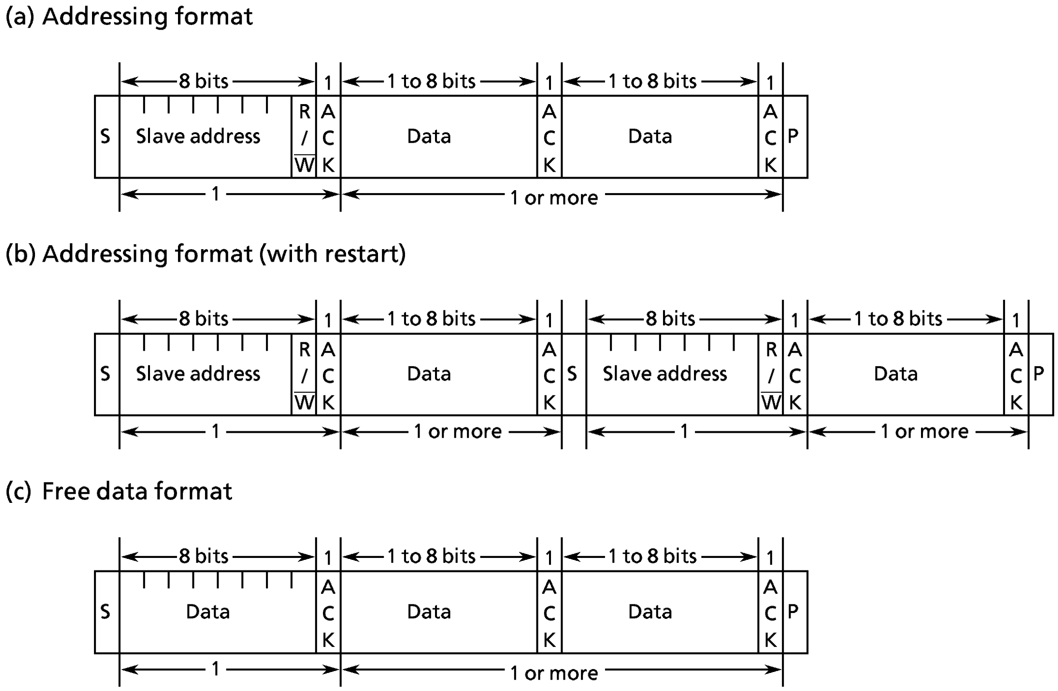
The following registers are used for control and operation status monitoring when using the serial bus interface (SBI-ver.A).

- Serial bus interface control register 1 (SBICR1)
- Serial bus interface control register 2 (SBICR2)
- Serial bus interface data buffer register (SBIDBR)
- I^{2}C bus address register (I2CAR)
- Serial bus interface status register (SBISR)

The above registers differ depending on a mode to be used.
Refer to Section "2.14.4 I^{2}C bus Mode Control" and "2.14.6 Clocked-synchronous 8-bit SIO Mode Control".

2.14.3 The Data Formats in the I^{2}C bus Mode

The data formats when using the serial bus interface circuit in the I^{2}C bus mode are shown below.


(a) Addressing format

(b) Addressing format (with restart)

(c) Free data format

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

Figure 2-83. Data Format

2.14.4 I^2C bus Mode Control

The following registers are used for control and operation status monitoring when using the serial bus interface (SBI-ver.A) in the I^2C bus mode.

Serial Bus Interface Control Register 1

SBICR1 (0020_H)

7	6	5	4	3	2	1	0
BC			ACK		SCK		

(Initial value: 0000 *000)

<table>
<tr><td rowspan="10">BC</td><td rowspan="10">Number of transferred bits</td><td rowspan="2">BC</td><td colspan="2">ACK = 0</td><td colspan="2">ACK = 1</td><td rowspan="10">Write only</td></tr>
<tr><td>Number of Clock</td><td>Bits</td><td>Number of Clock</td><td>Bits</td></tr>
<tr><td>000</td><td>8</td><td>8</td><td>9</td><td>8</td></tr>
<tr><td>001</td><td>1</td><td>1</td><td>2</td><td>1</td></tr>
<tr><td>010</td><td>2</td><td>2</td><td>3</td><td>2</td></tr>
<tr><td>011</td><td>3</td><td>3</td><td>4</td><td>3</td></tr>
<tr><td>100</td><td>4</td><td>4</td><td>5</td><td>4</td></tr>
<tr><td>101</td><td>5</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>110</td><td>6</td><td>6</td><td>7</td><td>6</td></tr>
<tr><td>111</td><td>7</td><td>7</td><td>8</td><td>7</td></tr>
<tr><td>ACK</td><td>Acknowledge mode specification</td><td colspan="5">0: Does not generate clock pulse for acknowledgment. (master mode) / Does not count clock pulse for acknowledgment. (slave mode)
1: Generates clock pulse for acknowledgment. (master mode) / Counts clock pulse for acknowledgment. (slave mode)</td><td>R/W</td></tr>
<tr><td>SCK</td><td>Serial clock selection</td><td colspan="5">000: 181.8 kHz
001: 105.3 kHz
010: 57.1 kHz
011: 29.9 kHz
100: 15.3 kHz
101: 7.72 kHz
110: 3.88 kHz
111: reserved
} at fc = 8 MHz (Output on SCL pin)</td><td>Write only</td></tr>
</table>

*Note 1: fc ; high-frequency clock [Hz], * ; Don't care*
Note 2: Set the BC to "000" before switching to a clock-synchronous 8-bit SIO mode.
Note 3: SBICR1 has write-only register bits, which cannot access any of in read-modify-write instructions such as bit operate, etc.
Note 4: Clear "0" to bit 3 in SBICR1.

Serial Bus Interface Data Buffer Register

SBIDBR (0021_H)

7	6	5	4	3	2	1	0

(Initial value: **** ****) Read / Write

Note 1: When writing transmitted data, start from the MSB.
Note 2: Cannot read the data which was written into SBIDBR, since a write data buffer and a read data buffer are independent in SBIDBR. Therefore, cannot access it any of in read-modify-write instructions such as bit operate, etc.
Note3: A value written to SBIDBR is cleared to "0" by INTSBI interrupt request signal.
*Note4: * ; Don't care*

I^2C-bus Address Register

I2CAR (0022_H)

7	6	5	4	3	2	1	0
Slave address							ALS
SA6	SA5	SA4	SA3	SA2	SA1	SA0	

(Initial value: 0000 0000)

SA	88CK48/M48 slave address selection		Write only
ALS	Address recognition mode specification	0: Slave address recognition 1: Non slave address recognition	

Note: I2CAR is a write-only register, which cannot access any of in read-modify-write instructions such as bit operate, etc.

Figure 2-84. Serial Bus Interface Control Register 1 / Serial Bus Interface Data Buffer Register/ I^2C bus Address Register in the I^2C bus Mode

Serial Bus Interface Control Register 2

SBICR2
(0023$_H$)

7	6	5	4	3	2	1	0
MST	TRX	BB	PIN	SBIM		"0"	"0"

(Initial value 0001 00**)

MST	Master / slave selection	0: Slave 1: Master	Write only
TRX	Transmitter / receiver selection	0: Receiver 1: Transmitter	
BB	Start / stop generation	0: Generate the stop condition when the MST, TRX, and PIN are "1". 1: Generate the start condition when the MST, TRX, and PIN are "1".	
PIN	Cancel interrupt service request	0: – 1: Cancel interrupt service request	
SBIM	Serial bus interface operating mode selection	00: Port mode (serial bus interface output disable) 01: SIO mode 10: I^{2}C bus mode 11: Reserved	

*Note 1: * ; Don't care*
Note 2: Switch a mode to port mode after confirming that the bus is free.
Note 3: Swich a mode to I^{2}C bus mode after confirming that input signals via port are "H" level.
Note 4: SBICR2 has write-only register bits, which can not access any of in read-modify-write instructions such as bit operate, etc.
Note 5: Clear bits 1 and 0 in SBICR2 to "0".

Serial Bus Interface Status Register

SBISR
(0023$_H$)

7	6	5	4	3	2	1	0
MST	TRX	BB	PIN	AL	AAS	AD0	LRB

(Initial value: 0001 0000)

MST	Master / Slave selection status monitor	0: Slave 1: Master	Read only
TRX	Transmitter / Receiver selection status monitor	0: Receiver 1: Transmitter	
BB	Bus status monitor	0: Bus free 1: Bus busy	
PIN	Interrupt service request status monitor	0: Requesting interrupt service 1: Releasing interrupt service request	
AL	Noise detection monitor	0: Does not detect noise 1: Detects noise	
AAS	Slave address match detection monitor	0: Does not detect slave address match or "GENERAL CALL" 1: Detects slave address match or "GENERAL CALL"	
AD0	"GENERAL CALL" detection monitor	0: Does not detect "GENERAL CALL" 1: Detects "GENERAL CALL"	
LRB	Last received bit monitor	0: Last received bit is "0" 1: Last received bit is "1"	

Figure 2-85. Serial Bus Interface Control Register 2 / Serial Bus Interface Status Register in the I^{2}C bus Mode

(1) **Acknowledgment mode specification**

Set the ACK (bit 4 in SBICR1) to "1" for operation in acknowledgment mode. When the serial bus interface circuit is the master mode, an additional clock pulse is generated for an acknowledge signal. In the transmitter mode during this additional clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during this additional clock pulse cycle, the SDA pin is set to "L" level generating the acknowledge signal.

Clear the ACK to "0" for operation in the non-acknowledgment mode. When the serial bus interface circuit is the master mode, a clock pulse for the acknowledge signal is not generated.

In the acknowledgment mode, when the serial bus interface circuit is the slave mode, clocks are counted for the acknowledge signal. During the clock for the acknowledge signal, when a received slave address matches to a slave address set to the I2CAR or a "GENERAL CALL" is received, the SDA pin is set to "L" level generating an acknowledge signal.

After a received slave address matches to a slave address set to the I2CAR and a "GENERAL CALL" is received, in the transmitter mode during the clock for the acknowledge signal, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode, the SDA pin is set to "L" level generating an acknowledge signal.

In the non-acknowledgment mode, when the serial bus interface circuit is the slave mode, clocks for the acknowledge signal are not counted.

(2) **Number of transfer bits**

The BC (bits 7 to 5 in the SBICR1) is used to select a number of bits for next transmitting and receiving data.

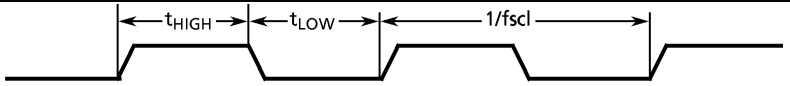
Since the BC is cleared to "000" by a start condition, a slave address and direction bit transmissions are executed in 8 bits. Other than these, the BC retains a specified value.

(3) **Serial clock**

a. Clock source

The SCK (bits 2 to 0 in the SBICR1) is used to select a maximum transfer frequency outputed on the SCL pin in the master mode.

Four or more machine cycles are required for both the "H" and "L" levels of the pulse width of a clock which is input externally in both the master and slave mode.



$t_{LOW} = 2^n/fc$

$t_{HIGH} = 2^n/fc + 12/fc$

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

Note: fc ; high-frequency clock

SCK (bits2 to 0 in the SBICR1)	n
000	4
001	5
010	6
011	7
100	8
101	9
110	10

Figure 2-86. Clock Source

b. Clock synchronization

The I^{2}C bus has a clock synchronization function to meet the transfer speed to a slow processing device when a transfer is performed between devices which have different process speed.

The clock synchronization functions when the SCL pin is "H" level and the SCL line of the bus is "L" level in the serial bus interface circuit. The serial bus interface circuit waits counting a clock pulse in "H" level until the SCL line of the bus is "H" level. When the SCL line of the bus is "H" level, the serial bus interface circuit starts counting during "H" level. The clock synchronization function holds clocks which are output from the serial interface circuit to be "H" level.

The slave device can stop the clock output of the master device on one word or one bit basis. Additionally, the transfer speed by the master device matches to the process speed of the slave device.

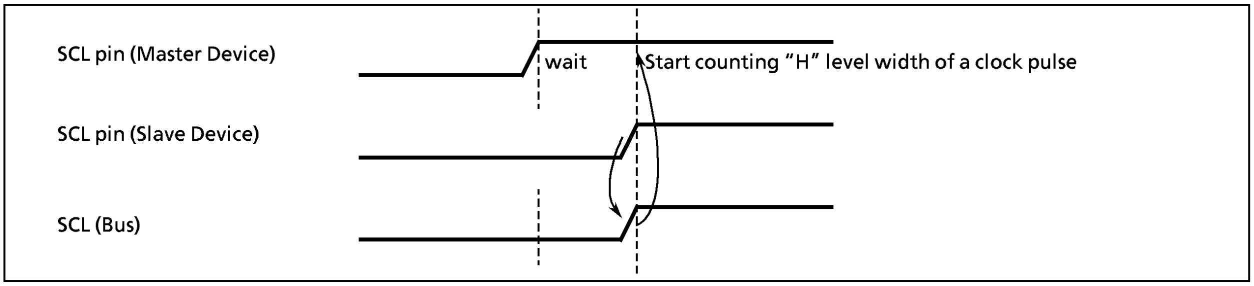


Figure 2-87. Clock Synchronization

(4) **Slave address and address recognition mode specification**
To operate the serial bus interface circuit in the addressing format which recognizes the slave address, clear the ALS (bit 0 in I2CAR) to "0" and set the slave address to the SA (bits 7 to 1 in I2CAR). To operate the serial bus interface circuit in the free data format which does not recognize the slave address, set the ALS to "1". When the serial bus interface circuit is used in the free data format, the slave address and the direction bit are not recognized. They are handled as data just after generation of start conditions.

(5) **Master/slave selection**
Set the MST (bit 7 in the SBICR2) to "1" for operating the serial bus interface as a master device. Clear the MST to "0" for operation as a slave device. The MST is cleared to "0" by the hardware after a stop condition on a bus is detected or the noise is detected.

(6) **Transmitter / receiver selection**
Set the TRX (bit 6 in the SBICR2) to "1" for operating the serial bus interface circuit as a transmitter. Clear the TRX to "0" for operation as a receiver. When data with an addressing format is transferred in the slave mode, the TRX is set to "1" by the hardware if the direction bit (R/$\overline{W}$) sent from the master device is "1", and 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 the noise is detected.

The following shows TRX change conditions in each mode and TRX after changing.

Mode	Direction bit	Change condition	TRX after changing
Slave mode	0	A received slave address is the same as a value set to I2CAR.	0
	1		1
Master mode	0	ACK signal is returned.	1
	1		0

When the serial bus interface circuit operates in the free data format, the slave address and the direction bit are not recognized. They are handled as data just after generating a start condition. The TRX was not changed by the hardware.

(7) **Start/stop condition generation**

When the BB (bit 5 in the SBICR2) is "0", the slave address and the direction bit which are set to the SBIDBR are output on a bus after generating a start condition by writing "1" to the MST, TRX, BB, and PIN. It is necessary to set transmitted data to the data buffer register (SBIDBR) and set "1" to ACK beforehand.

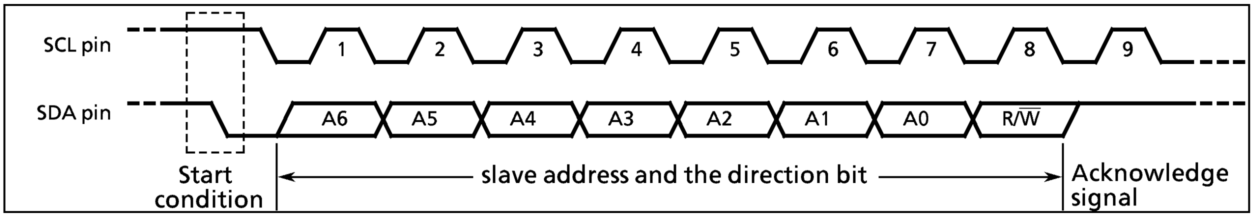


Figure 2-88. Start Condition Generation and Slave Address Generation

When the BB is "1", a sequence of generating a stop condition is started by writing "1" to the MST, TRX, and PIN, and "0" to the BB. Do not modify the contents of MST, TRX, BB and PIN until a stop condition is generated on a bus.

When a stop condition is generated and the SCL line on the bus is set to "L" level by another device, a stop condition is generated after releasing the SCL line.

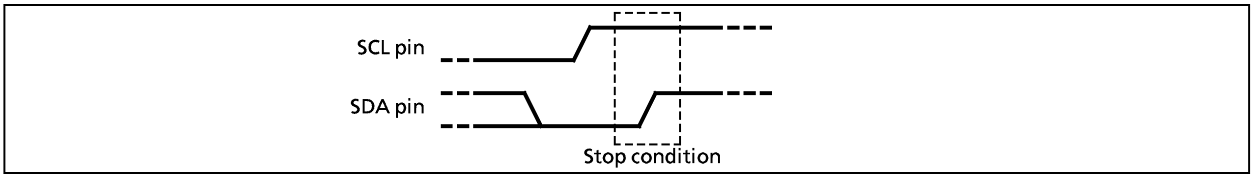


Figure 2-89. Stop Condition Generation

The bus condition can be indicated by reading the contents of the BB (bit 5 in the SBISR). The BB is set to "1" when a start condition on a bus is detected,and is cleared to "0" when a stop condition is detected on a bus.

(8) **Interrupt service request and cancel**

When the serial bus interface circuit is the master mode and transferring a number of clocks set by the BC and the ACK is complete, a serial bus interface interrupt request (INTSBI) is generated.

In the slave mode, the INTSBI is generated when the received slave address is the same as the value set to the I2CAR and an acknowledge signal is output, when a "GENERAL CALL" is received and an acknowledge signal is output, or when transferring / receiving data is complete after the received slave address is the same as the value set to the I2CAR and a "GENERAL CALL" is received.

When the serial bus interface interrupt request occurs, the PIN (bit 4 in the SBISR) is cleared to "0". During the time that the PIN is "0", the SCL pin is set to "L" level.

Either writing or reading data to or from the SBIDBR sets the PIN to "1".

The time from the PIN being set to "1" until the SCL pin is released takes t_{LOW}.

Although the PIN (bit 4 in the SBICR2) can be set to "1" by the program, the PIN is not cleared to "0" when it is written "0".

(9) **Serial bus interface operating mode selection**

The SBIM (bits 3 and 2 in the SBICR2) is used to specify the serial bus interface operation mode.

Set the SBIM to "10" when used in the I^2C-bus mode after confirming that the serial bus interface pin is "H" level. Switch a mode to port after confirming that the bus is free.

(10) **Noise detection monitor**

The I²C bus is easy to be affected by noise, because the bus is driven by the open drain and the pull-up resistor.

With the serial bus interface circuit, the SDA pin output and the SDA line level are compared at a rise of the SCL line on the bus, and whether data are output correctly on the bus is detected only in the master transmitter mode.

When the SDA pin output differs from the SDA line level, the AL (bit 3 in the SBISR) is set to "1". When the AL is set to "1", the SDA pin is released and the MST and the TRX are cleared to "0" by the hardware. The serial bus interface circuit changes to the slave receiver mode, and the serial bus interface circuit continues outputting clocks until transferring data when the AL was set to "1" is completed.

Either writing or reading data to or from the SBIDBR, or writing data to the SBICR2 clears to the AL to "0".

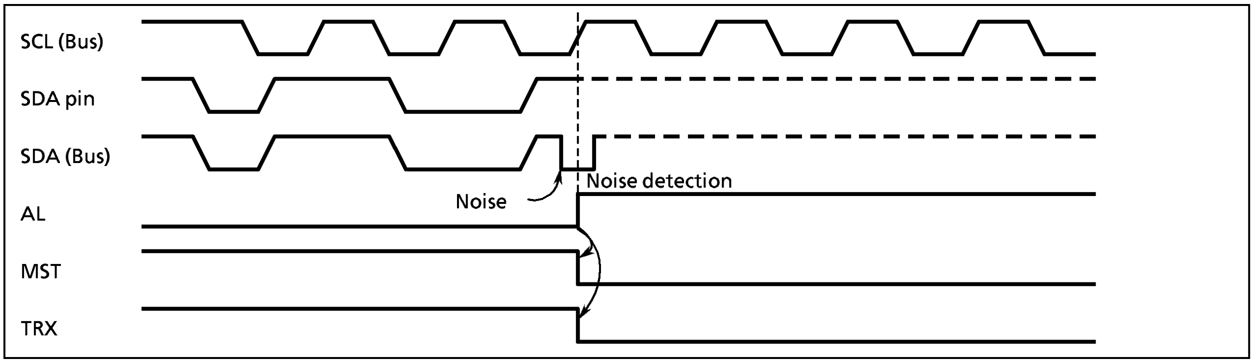


Figure 2-90. Noise Detection Monitor

(11) **Slave address match detection monitor**

The AAS (bit 2 in the SBISR) is set to "1" in the slave mode, in the address recognition mode (ALS = 0), when receiving "GENERAL CALL" or a slave address with the same value that is set to the I2CAR. When the ALS is "1", the AAS is set to "1" after receiving the first 1-word of data. The AAS is cleared to "0" by writing / reading data to / from a data buffer register.

(12) **GENERAL CALL detection monitor**

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

(13) **Last received bit monitor**

The SDA value stored at the rising edge of the SCL is set to the LRB (bit 0 in the SBISR). In the acknowledge mode, immediately after an INTSBI interrupt request is generated, an acknowledge signal is read by reading the contents of the LSB.

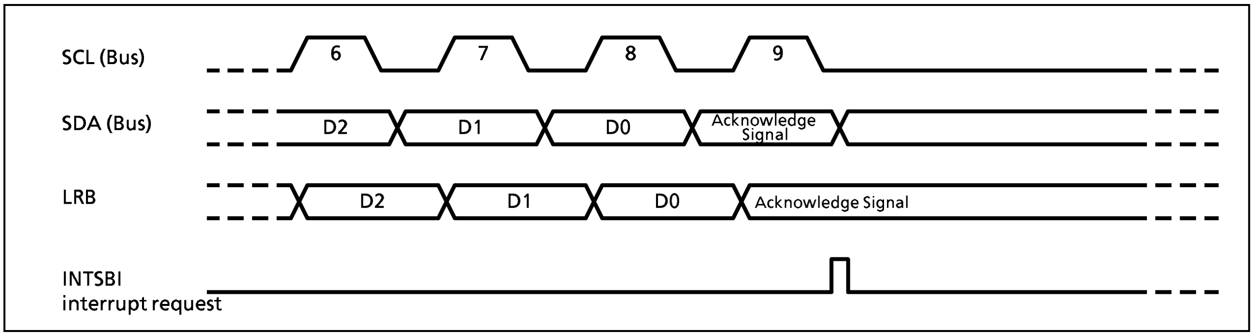


Figure 2-91. Last Received bit Monitor

2.14.5 Data Transfer in I^2C bus Mode

(1) **Device Initialization**

Set the ACK in the SBICR1 to "1", and the BC to 000. Specify the data length to 8 bits to count clocks for acknowledge. Set a transfer frequency to the SCK.

Subsequently, set a slave address to the SA in the I2CAR and clear the ALS to "0" to set an addressing format.

After confirming that the serial bus interface pin is "H" level, for specifying the default setting to a slave receiver mode, clear "0" to the MST, TRX, and BB in the SBICR2, set "1" to the PIN, "10" to the SBIM, and "0" to bits 1 and 0,

Note: The initialization of the serial bus interface circuit must be complete within the time from all devices which are connected to the bus have initialized to any device does not generate a start condition. If not, there is a possibility that another device starts transferring before an end of the initialization of the serial bus interface circuit. Data can not be received correctly.

(2) **Start Condition and Slave Address Generation**

Confirm a bus free status (when BB = 0).

Set the ACK to "1" and specify a slave address and a direction bit to be transmitted to the SBIDBR.

When the BB is "0", the start condition are generated and the slave address and the direction bit which are set to the SBIDBR are output on a bus by writing "1" to the MST, TRX, BB and PIN. An INTSBI interrupt request occurs at the 9th falling edge of the SCL clock cycle, and the PIN is cleared to "0". The SCL pin is pulled down to the "L" level while the PIN is "0". When an interrupt request occurs, the TRX changes by the hardware according to the direction bit only when an acknowledge signal is returned from the slave device.

*Note 1: Do not write a slave address to be output to the **SBIDBR** while data are transferred. If data is written to the **SBIDBR**, data to been outputting may be destroyed.*

*Note 2: Do not start transferring due to another master from writing a slave address to be output to the **SBIDBR** to writing a start condition generation command to the **SBICR2**. The serial bus interface circuit malfunctions because it has not an arbitration function.*

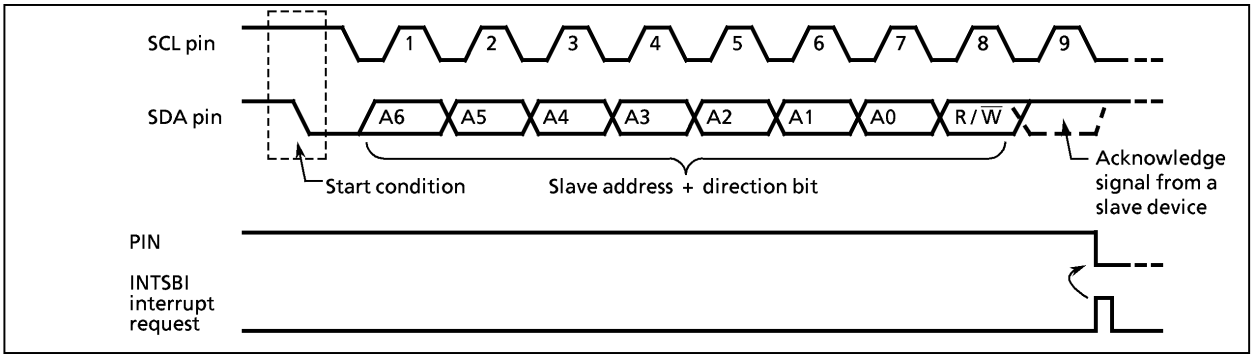


Figure 2-92. Start Condition Generation and Slave Address Transfer

(3) **1-word Data Transfer**

Check the MST by the INTSBI interrupt process after an 1-word data transfer is completed, and determine whether the mode is a master or slave.

a. When the MST is "1" (Master mode)

Check the TRX and determine whether the mode is a transmitter or receiver.

① When the TRX is "1" (Master mode)

Test the LRB. When the LRB is "1", a receiver does not request data. Implement the process to generate a stop condition (described later) and terminate data transfer.

When the LRB is "0", the receiver requests new data. When the next transmitted data is other than 8 bits, set the BC, set the ACK to "1", and write the transmitted data to the SBIDBR. After writing the data, the PIN becomes "1", a serial clock pulse is generated for transferring a new 1-word of data from the SCL pin, and then the 1-word data is transmitted. After the data is transmitted, and an INTSBI interrupt request occurs. The PIN becomes "0" and the SCL pin is set to "L" level. If the data to be transferred is more than one word in length, repeat the procedure from the LRB test above.

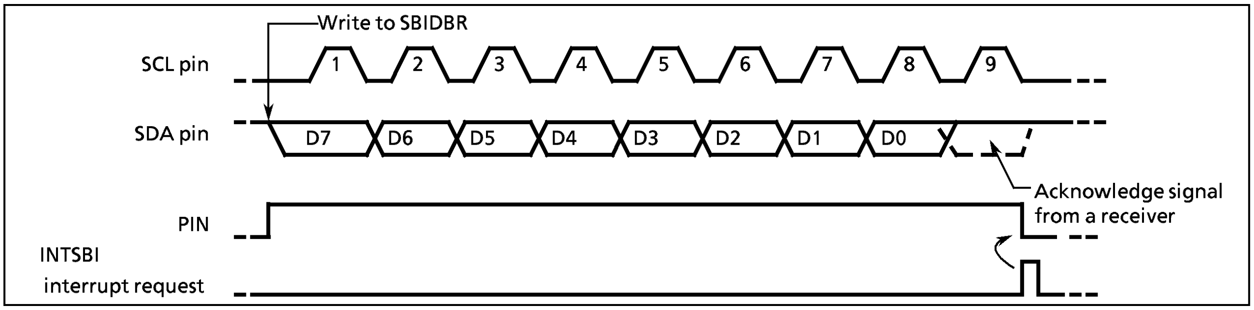


Figure 2-93. Example when BC = "000", ACK = "1" in Transmitter Mode

② When the TRX is "0" (Receiver mode)

When the next transmitted data is other than 8 bits, set the BC again. Set the ACK to "1" and read the received data from the SBIDBR (data which is read immediately after a slave address is sent is undefined). After the data is read, the PIN becomes "1". The serial bus interface circuit outputs a serial clock pulse to the SCL to transfer new 1-word of data and sets the SDA pin to "0" at the acknowledge signal timing.

An INTSBI interrupt request occurs and the PIN becomes "0". Then the serial bus interface circuit pulls down the SCL pin to the "L" level. The serial bus interface circuit outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBIDBR.

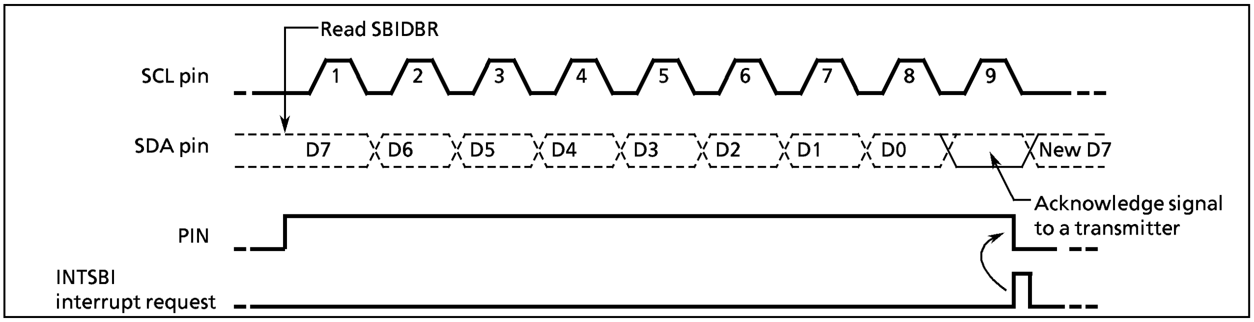


Figure 2-94. Example when BC = "000", ACK = "1" in Receiver Mode

In order to terminate transmitting data to a transmitter, clear the ACK to "0" before reading data which is 1 word before the last data to be received. The last data does not generate a clock pulse for the acknowledge signal. After the data is transmitted and an interrupt request has occurred, set the BC to "001" and read the data. The serial bus interface circuit generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA line on a bus keeps the "H" level. The transmitter receives the "H" level signal as an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, the serial bus interface circuit generates a stop condition (Refer to 2.14.5. (4)) and terminates data transfer.

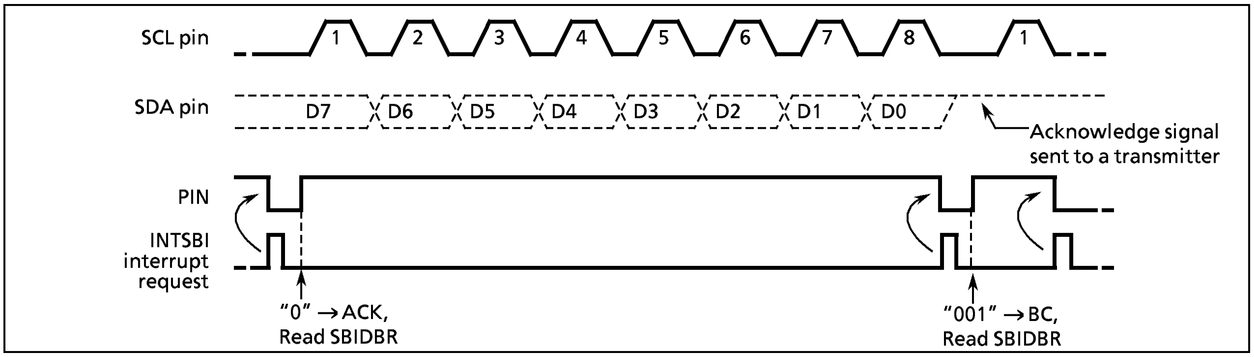


Figure 2-95. Termination of Data Transfer in Master Receiver Mode

b. When the MST is "0" (Slave mode)

In the slave mode, the serial bus interface circuit operates either in normal slave mode or in recovery process after a noise detection.

In the slave mode, an INTSBI interrupt request occurs when the serial bus interface circuit receives a slave address or a "GENERAL CALL" from the master device, or when a "GENERAL CALL" is received and data transfer is complete after matching a received slave address. In the master mode, the serial bus interface circuit operates in a slave mode if a noise is detected. An INTSBI interrupt request occurs when word data transfer terminates after a noise detection. When an INTSBI interrupt request occurs, the PIN (bit 4 in the SBICR2) is reset, and the SCL pin is set to "L" level. Either reading or writing from or to the SBIDBR or setting the PIN to "1" releases the SCL pin after taking t_{LOW} time. The serial bus interface circuit tests the AL (bit 3 in the SBISR), the TRX (bit 6 in the SBISR), the AAS (bit 2 in the SBISR), and the AD0 (bit 1 in the SBISR) and implements processes according to conditions listed in the next table.

Table 2-17. Operation in the Slave Mode

TRX	AL	AAS	AD0	Conditions	Process
1	0	1	0	In the slave receiver mode, the serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "1".	Set the number of bits in 1-word to the BC and write transmitted data to the SBIDBR.
		0	0	In the slave transmitter mode, 1-word data is transmitted.	Check the LRB. If the LRB is set to "1", set the PIN to "1" since the receiver does not request next data. Then, clear the TRX to "0" release the bus. If the LRB is cleared to "0", set the number of bits in a word to the BC and write transmitted data to the SBIDBR since the receiver requests next data.
0	1	0	0	The serial bus interface circuit detects the noise when transmitting a slave address or data and terminates transferring word data.	There is a possibility that a serial bus interface circuit does not receive data normally. The recovery process such as a data re-transfer, etc. is needed.
	0	1	1/0	In the slave receiver mode, the serial bus interface circuit receives a slave address or GENERAL CALL of which the value of the direction bit sent from the master is "0".	Read the SBIDBR for setting the PIN to "1" (reading dummy data) or set the PIN to "1".
		0	1/0	In the slave receiver mode, the serial bus interface circuit terminates receiving of 1-word data.	Set the number of bits in a word to the BC and read received data from the SBIDBR.

(4) **Stop Condition Generation**

When the BB is "1", a sequence of generating a stop condition is started by setting "1" to the MST, TRX and PIN, and "0" to the BB. Do not modify the contents of the MST, TRX, BB, PIN until a stop condition is generated on a bus. When a SCL line of bus is pulled down by other devices, the serial bus interface circuit generates a stop condition after they release a SCL line.

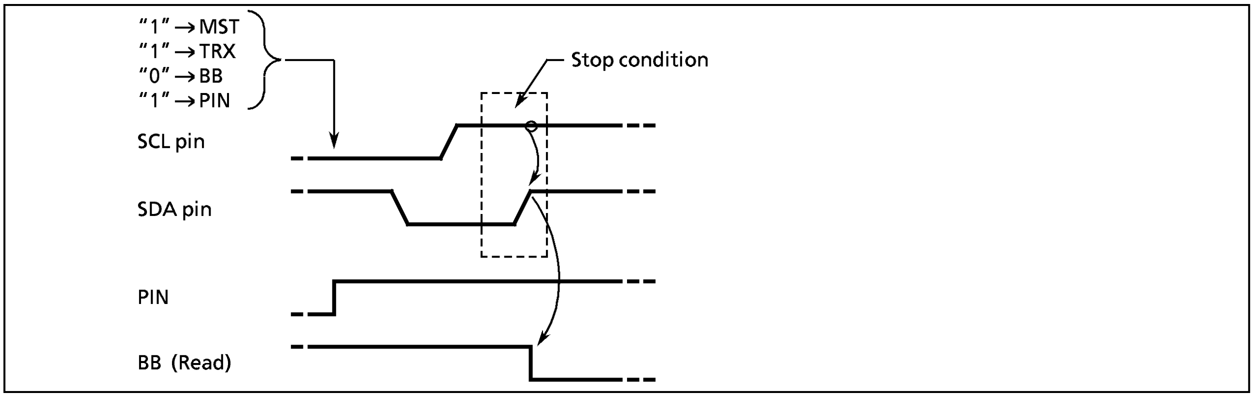


Figure 2-96. Stop Condition Generation

(5) **Restart**

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart the serial bus interface circuit.

Clear "0" to the MST, TRX, and BB and set "1" to the PIN. The SDA pin retains the "H" level and the SCL pin is released. Since a stop condition is not generated on the bus, the bus is assumed to be in a busy state from other devices. Test the BB until it becomes "0" to check that the SCL pin of the serial bus interface circuit is released. Test the LRB until it becomes "1" to check that the SCL line of the bus is not set to "L" level by other devices. After confirming that the bus stays in a free state, generate a start condition with procedure (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.

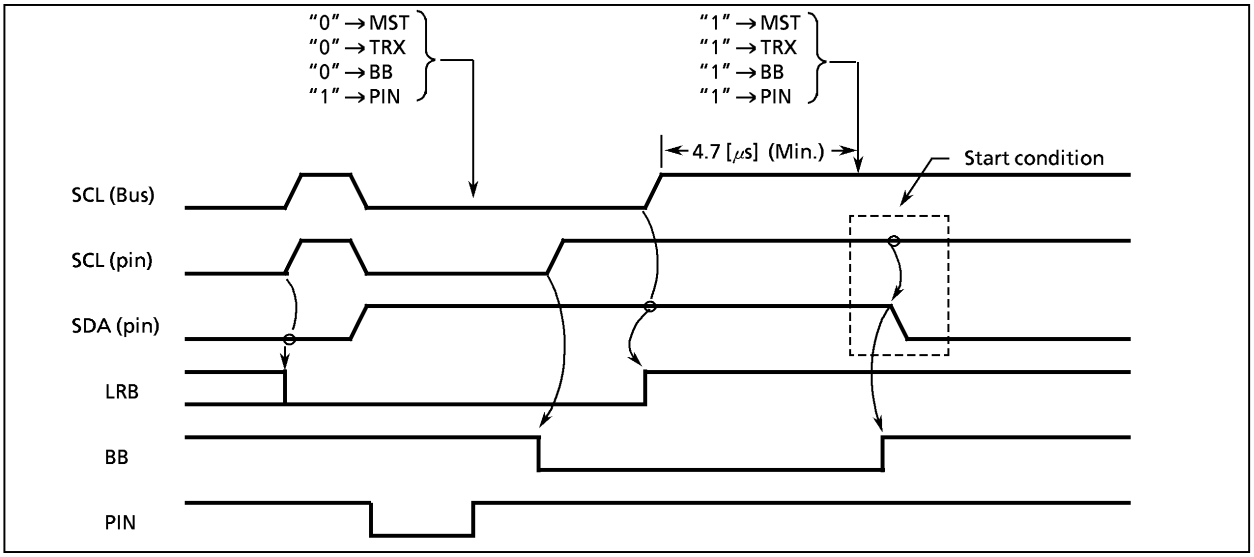


Figure 2-97. Timing Diagram when Restarting

2.14.6 Clocked-synchronous 8-bit SIO Mode Control

The following registers are used for control and operation status monitoring when using the serial bus interface (SBI-ver.A) in the clocked-synchronous 8-bit SIO mode.

Serial Bus Interface Control Register 1

SBICR1 (0020_H)

7	6	5	4	3	2	1	0
SIOS	SIO INH	SIOM			SCK		

(Initial value: 0000 0000)

SIOS	Indicate transfer start/stop	0: Stop 1: Start	Write only
SIOINH	Continue/abort transfer	0: Continue transfer 1: Abort transfer (automatically cleared after abort)	
SIOM	Transfer mode select	00: 8-bit transmit mode 01: reserved 10: 8-bit transmit/receive mode 11: 8-bit receive mode	
SCK	Serial clock select	000: fc/2^5 (250 kHz) 001: fc/2^6 (125 kHz) 010: fc/2^7 (62.5 kHz) 011: fc/2^8 (31.25 kHz) 100: fc/2^9 (15.62 kHz) 101: fc/2^{10} (7.81 kHz) 110: fc/2^{11} (3.90 kHz) } at fc = 8 MHz (Output on $\overline{\text{SCK}}$ pin) 111: External clock (input from $\overline{\text{SCK}}$ pin)	

Note 1: fc ; high-frequency clock [Hz]
Note 2: Clear the SISO to "0" and set the SIOINH to "1" when setting the transfer mode and serial clock.
Note 3: SBICR1 is a write-only register, which cannot access any of in read-modify-write instructions such as bit operate, etc.
Note 4: Clear bit 3 to "0" in SBICR1.

Serial Bus Interface Data Buffer Register

SBIDBR (0021_H)

7	6	5	4	3	2	1	0

(Initial value: **** ****) Read / Write

Note 1: Cannot read the data which was written into SBIDBR, since a write data buffer and a read data buffer are independent in SBIDBR. Therefore, cannot access it any of in read-modify-write instructions such as bit operate, etc.
Note 2: don't care

Serial Bus Interface Control Register 2

SBICR2 (0023_H)

7	6	5	4	3	2	1	0
"0"	"0"	"0"	"1"	SBIM		"0"	"0"

(Initial value: **** 00**)

SBIM	Serial bus interface operation mode selection	00: Port mode (serial bus interface output disable) 01: SIO mode 10: I^2C bus mode 11: reserved	Write only

*Note 1: * ; Don't care*
Note 2: Switch a mode to port after data transfer is complete.
Note 3: Switch a mode to SIO mode after confirming that input signals via port are high level.
Note 4: SBICR2 is a write-only regiter, which cannot access any of in read-modify-write instructions such as bit operate, etc.
Note 5: Clear bits 7 to 5 in the SBICR2 to "0", and set bit 4 to "1".

Figure 2-98-1. Serial Bus Interface Control Register 1 / Serial Bus Interface Data Buffer Register / Serial Bus Interface Control Register 2 in SIO Mode

Serial Bus Interface Status Register

SBISR (0023$_H$)	7	6	5	4	3	2	1	0	
	"1"	"1"	"1"	"1"	SIOF	SEF	"1"	"1"	(Initial value: 1111 0011)

SIOF	Serial transfer operating status monitor	0: Transfer terminated 1: Transfer in process	Read only
SEF	Shift operating status monitor	0: Shift operation terminated 1: Shift operation in process	

Figure 2-98-2. Serial Bus Interface Status Register in SIO Mode

(1) Serial Clock

<u>a. Clock source</u>

The SCK (bit 2 to 0 in the SBICR1) is used to select the following functions.

① Internal Clock

In an internal clock mode, any of seven frequencies can be selected. The serial clock is output to the outside on the $\overline{\text{SCK}}$ pin. The $\overline{\text{SCK}}$ pin becomes a "H" level when data transfer starts. When writing (in the transmit mode) or reading (in the receive mode) data cannot follow the serial clock rate, an automatic-wait function is executed to stop the serial clock automatically and hold the next shift operation until reading or writing is complete.

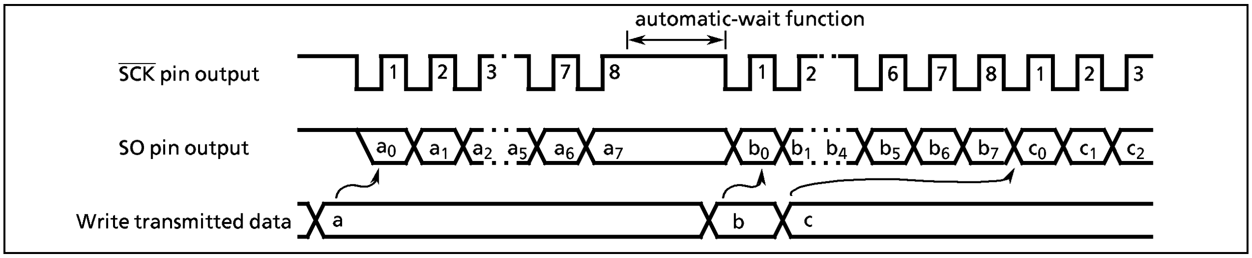
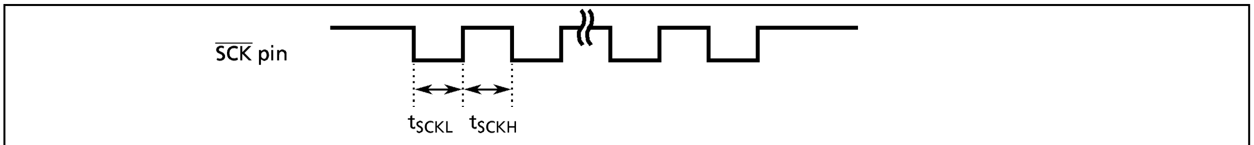

automatic-wait function
$\overline{\text{SCK}}$ pin output
1 2 3 7 8 1 2 6 7 8 1 2 3
SO pin output
a_0 a_1 a_2 a_5 a_6 a_7 b_0 b_1 b_4 b_5 b_6 b_7 c_0 c_1 c_2
Write transmitted data
a
b
c

Figure 2-99. Automatic-wait Function

② External clock (SCK = "111")

An external clock supplied to the $\overline{\text{SCK}}$ pin is used as the serial clock. In order to ensure shift operation, a pulse width of at least 4 machine cycles is required for both "H" level and "L" level in the serial clock. The maximum data transfer frequency is 250 kHz (when fc = 8 MHz).



t_{SCKL}, t_{SCKH} > 4 tcyc *Note: tcyc = 4/fc (in NORMAL mode, IDLE mode)*

Figure 2-100. Maximum Data Transfer Frequency When External Clock Input

b. Shift edge

The leading edge is used to transmit data, and the trailing edge is used to receive data.

① Leading edge shift

Data is shifted on the leading edge of the serial clock (at a falling edge of the $\overline{\mathrm{SCK}}$ pin input/output).

② Trailing edge shift

Data is shifted on the trailing edge of the serial clock (at a rising edge of the $\overline{\mathrm{SCK}}$ pin input/output).

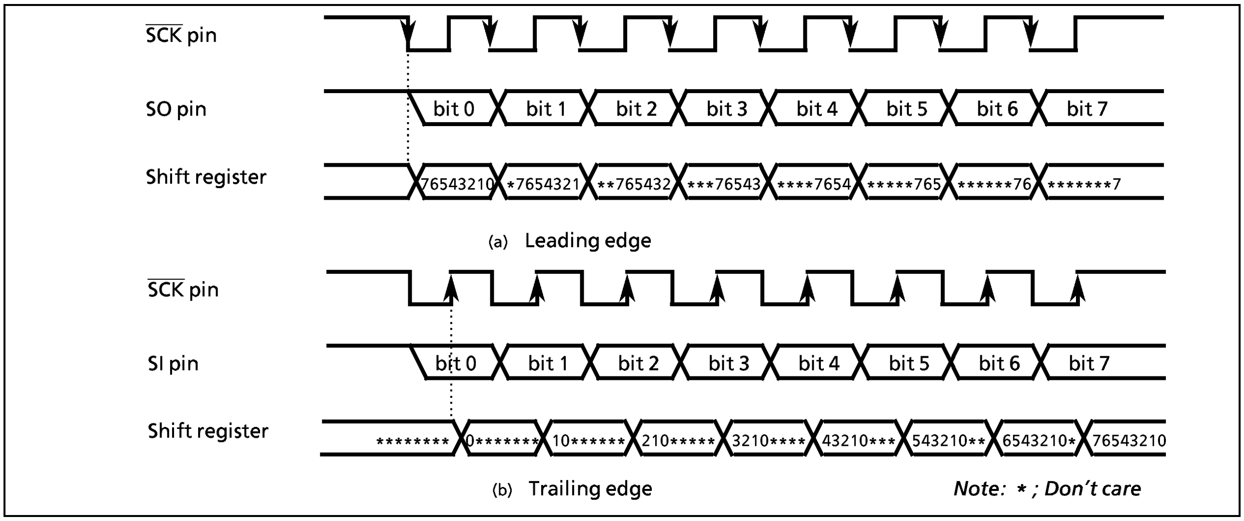


Figure 2-101. Shift Edge

(2) Transfer mode

The SIOM (bit 5 and 4 in the SBICR1) is used to select a transmit, receive, or transmit/receive mode.

a. 8-bit transmit mode

Set a control register to a transmit mode and write transmit data to the SBIDBR.

After the transmit data is written, set the SIOS to "1" to start data transfer. The transmitted data is transferred from the SBIDBR to the shift register and output to the SO pin in synchronous with the serial clock, starting from the least significant bit (LSB). When the transmit data is transferred to the shift register, the SBIDBR becomes empty. The INTSBI (buffer empty) interrupt request is generated to request new data.

When the internal clock is used, the serial clock will stop and 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 transmit data is written, automatic-wait function is canceled.

When the external clock is used, data should be written to the SBIDBR 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 SBIDBR by the interrupt service program.

When the transmit is started, after the SIOF goes "1" output from the SO pin holds final bit of the last data until falling edge of the $\overline{\mathrm{SCK}}$.

Transmitting data is ended by clearing the SIOS to "0" by the buffer empty interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, the transmitted mode ends when all data is output. In order to confirm if data is surely transmitted by the program, set the SIOF (bit 3 in the SBISR) to be sensed. The SIOF is cleared to "0" when transmitting is complete. When the SIOINH is set, transmitting data stops. The SIOF turns "0".

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

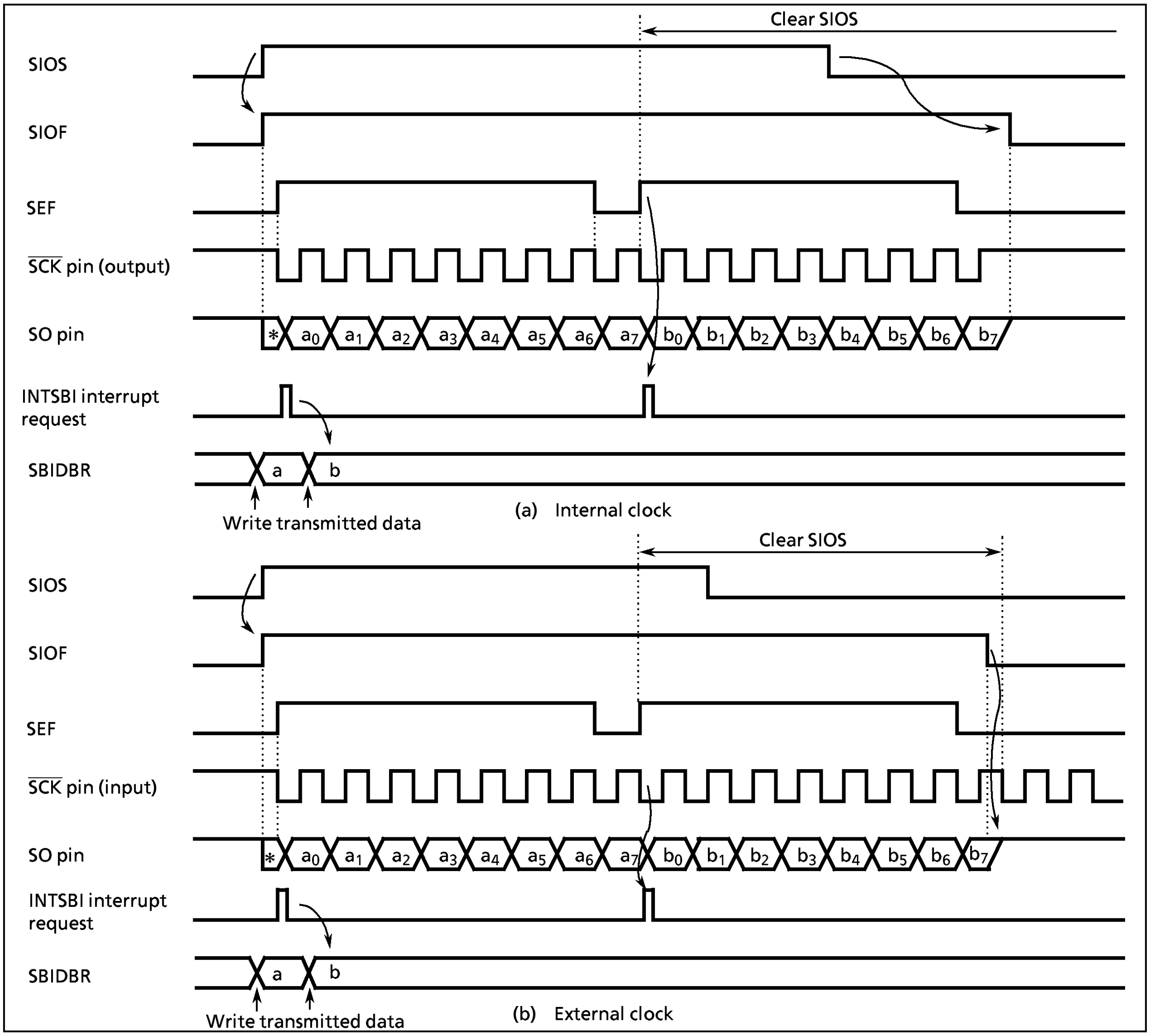


Figure 2-102. Transfer Mode

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

```
STEST1:  TEST  (SBISR) . SEF          ; If SEF = 1 then loop
         JRS  F , STEST1
STEST2:  TEST  (P3) . 6               ; If SCK = 0 then loop
         JRS  T , STEST2
         LD   (SBICR1), 00000111B ; SIOS ← 0
```

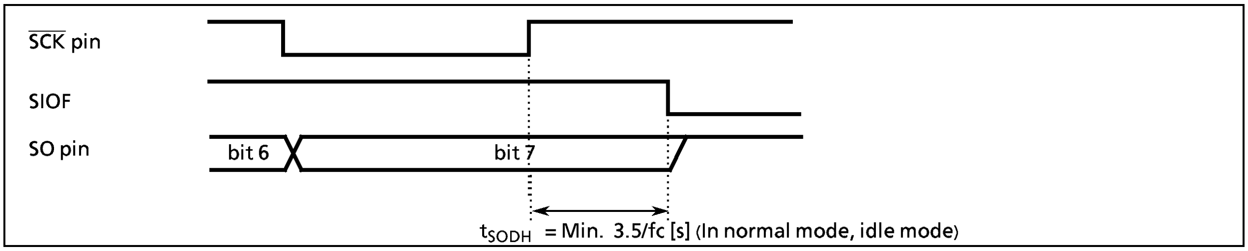


Figure 2-103. Transmitted Data Hold Time at End of Transmit

b. 8-bit Receive Mode

Set the control register to receive mode and the SIOS to "1" for switching to receive mode. Data is received from the SI pin to the shift register in synchronous 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 SBIDBR. The INTSBI (buffer full) interrupt request is generated to request of reading the received data. The data is then read from the SBIDBR by the interrupt service program.

When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated until the received data is read from the SBIDBR.

When the external clock is used, since shift operation is synchronized with the clock pulse provided externally, the received data should be read from the SBIDBR before next serial clock is input. If the received data is not read, further data to be received is canceled. 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 when received data is read.

Receiving data is ended by clearing the SIOS to "0" by the buffer full interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, received data is transferred to the SBIDBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm if data is surely received by the program, set the SIOF (bit 3 in the SBIDBR) to be sensed. The SIOF is cleared to "0" when receiving is complete. After confirming that receiving has ended, the last data is read. When the SIOINH is set, receiving data stops. The SIOF turns "0" (the received data becomes invalid, therefore no need to read it).

*Note: When the transfer mode is switched, the **SBIDBR** contents are lost. In case that the mode needs to be switched, receiving data is concluded by clearing the **SIOS** to "0", read the last data, and then switch the mode.*

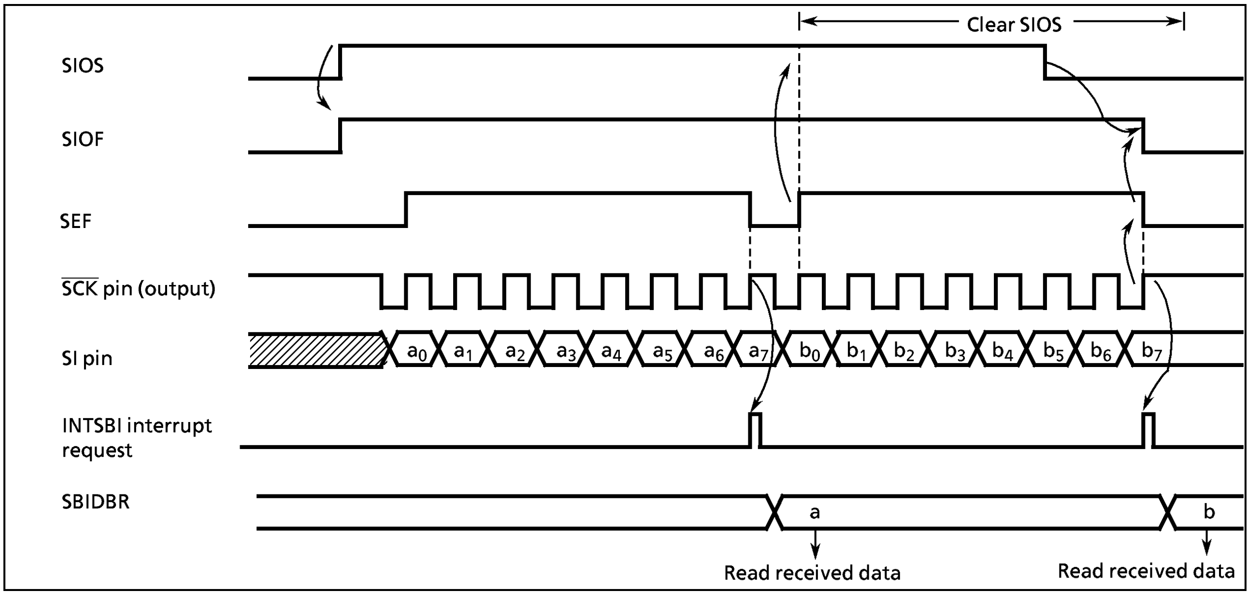


Figure 2-104. Receive Mode (Example: Internal clock)

c. 8-bit Transmit / Receive Mode

Set a control register to a transmit / receive mode and write data to the SBIDBR. After the data is written, set the SIOS to "1" to start transmitting / receiving. When transmitting, the data is output from the SO pin on the leading edges in synchronous with the serial clock, starting from the least significant bit (LSB). When receiving, the data is input to the SI pin on the trailing edges of the serial clock. 8-bit data is transferred from the shift register to the SBIDBR, and the INTSBI interrupt request occurs. The interrupt service program reads the received data from the data buffer register and writes data to be transmitted. The SBIDBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

When the internal clock is used, automatic-wait function is initiated until received data is read and next data is written.

When the external clock is used, since the shift operation is synchronized with the external clock, received data is read and transmitted data is written before 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 when received data is read and transmitted data is written.

When the transmit is started, after the SIOF goes "1" output from the SO pin holds final bit of the last data until falling edge of the $\overline{\text{SCK}}$.

Transmitting / receiving data is ended by clearing the SIOS to "0" by the INTSBI interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, received data is transferred to the SBIDBR in complete blocks. The transmit / receive mode ends when the transfer is complete. In order to confirm if data is surely transmitted / received by the program, set the SIOF (bit3 in the SBISR) to be sensed. The SIOF becomes "0" after transmitting / receiving is complete. When the SIOINH is set, transmitting / receiving data stops. The SIOF turns "0".

Note: When the transfer mode is switched, the SBIDBR contents are lost. In case that the mode needs to be switched, conclude transmitting / receiving data by clearing the SIOS to "0", read the last data, and then switch the transfer mode.

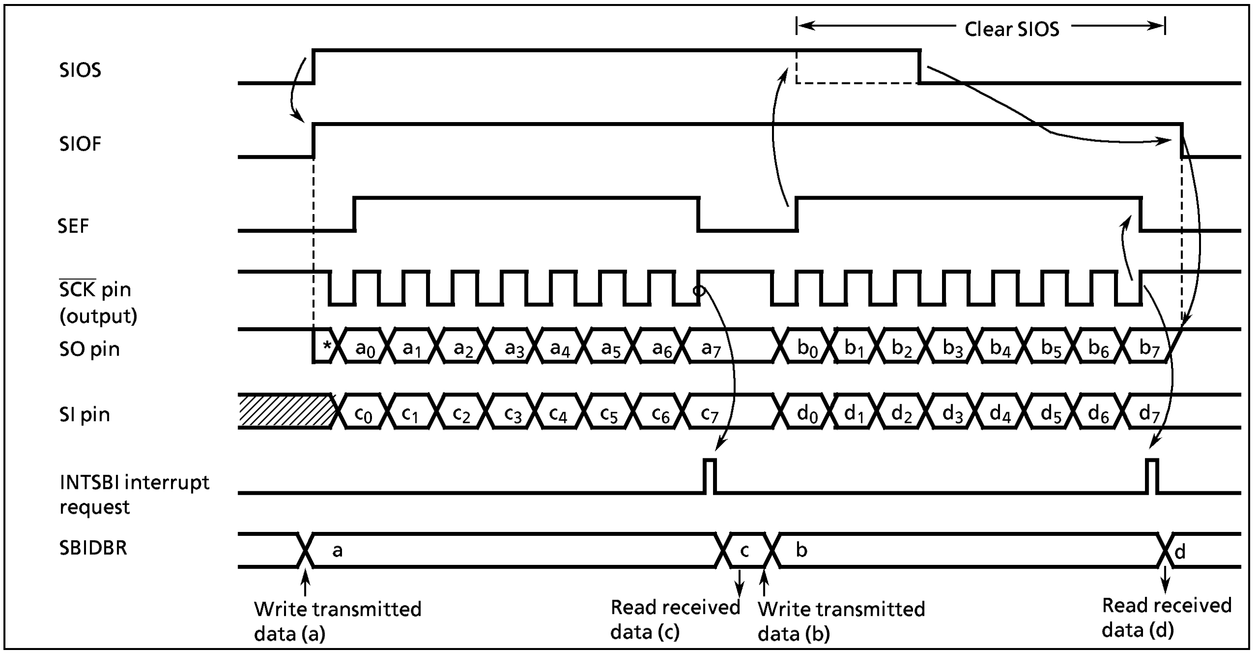


Figure 2-105. Transmit / Receive Mode (Example: Internal clock)

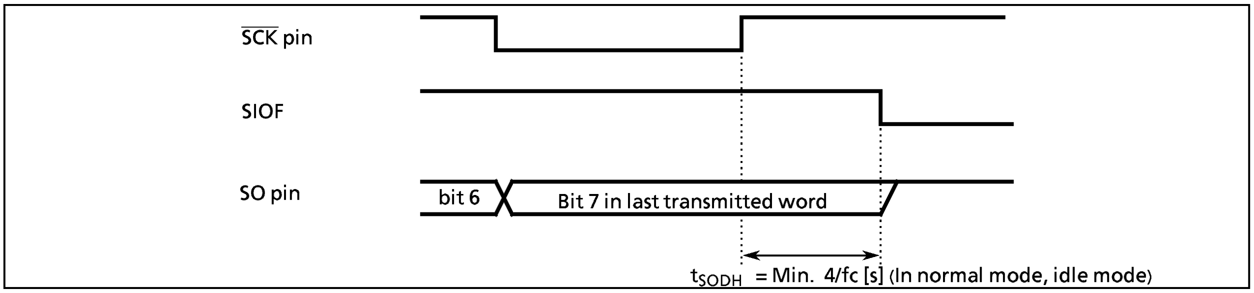

SCK pin
SIOF
SO pin
bit 6
Bit 7 in last transmitted word
t_{SODH} = Min. 4/fc [s] (In normal mode, idle mode)

Figure 2-106. Transmitted Data Hold Time at End of Transmit / Receive

2.15 10-bit AD Converter (ADC)

2.15.1 Configuration

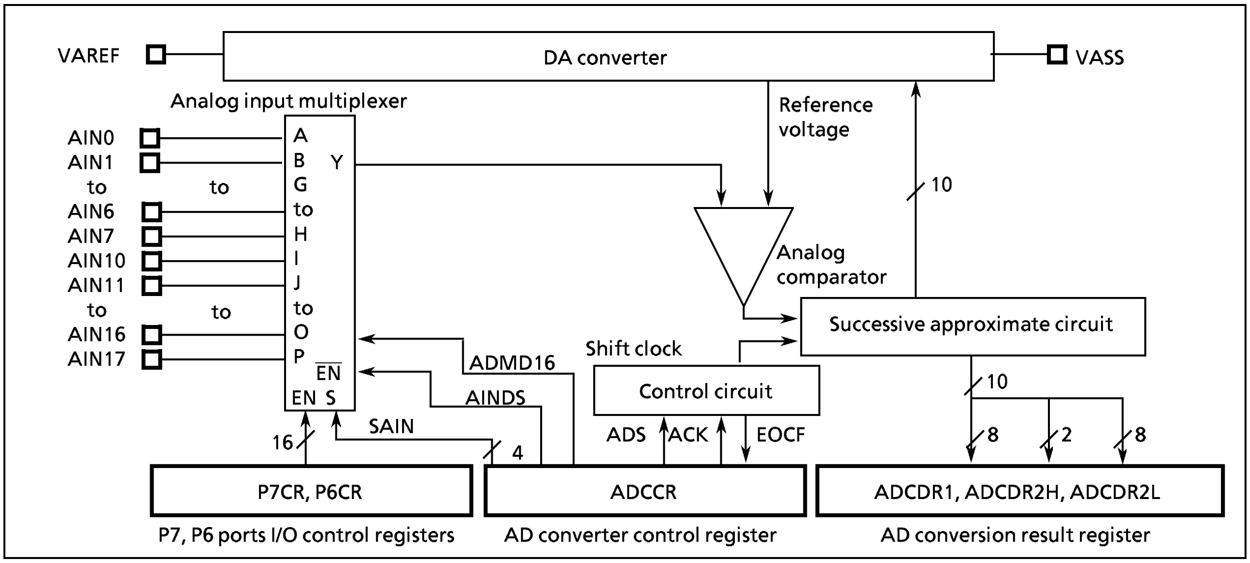


Figure 2-107. AD Converter (ADC)

2.15.2 Control

The AD converter is controlled using the AD converter control register (ADCCR), port P6 I/O control register (P6CR), and port P7 I/O control register (P7CR). Reading EOCF in ADCCR detects the AD converter operating status; reading AD conversion data register (ADCDR1) or (ADCDR2H), (ADCDR2L) detects AD conversion value.

AD converter control register

	7	6	5	4	3	2	1	0	
ADCCR ($000E_H$)	EOCF/ ADMD16	ADS	ACK	AINDS	SAIN				(Initial value: 0000 0000)

SAIN	Analog input channel select	0000: Selects AIN00. 1000: Selects AIN10. 0001: Selects AIN01. 1001: Selects AIN11. 0010: Selects AIN02. 1010: Selects AIN12. 0011: Selects AIN03. 1011: Selects AIN13. 0100: Selects AIN04. 1100: Selects AIN14. 0101: Selects AIN05. 1101: Selects AIN15. 0110: Selects AIN06. 1110: Selects AIN16. 0111: Selects AIN07. 1111: Selects AIN17.	R/W
AINDS	Analog input control	0: Analog input enable 1: Analog input disable	
ACK	Conversion time select	0: reserved 1: 736/fc (46 μs at 16 MHz)	
ADS	AD conversion start	0: – 1: AD conversion start	
EOCF	AD conversion end flag	0: Under conversion or before conversion 1: End of conversion	Read only
ADMD16	Number of analog channels switch	0: Selects 8 channels. (AIN00 to 07) 1: Selects 16 channels. (AIN10 to 17)	Write only

Note 1: The analog input channel must be selected when the AD conversion is stopped.
Note 2: The ADS is automatically cleared to "0" after starting the AD conversion.
Note 3: The EOCF is cleared to "0" by reading the AD conversion registers such as ADCDR1, ADCDR2H, or ADCDR2L.
Note 4: The EOCF is read-only.
Note 5: ADMD16 is write-only ; cannot be read.
Note 6: To select AIN10 to AIN17, switch to 16 channels. (ADMD16 = 1).
Note 7: fc ; High-frequency clock [Hz]

Figure 2-108. AD Converter Control Register

AD conversion data registers

	7	6	5	4	3	2	1	0	
ADCDR1 (000F$_H$)	DATA9	DATA8	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	Read only
ADCDR2H (0025$_H$)	1	1	1	1	1	1	DATA9	DATA8	Read only
ADCDR2L (0024$_H$)	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	Read only

Port P6 I/O control register

	7	6	5	4	3	2	1	0	
P6CR (000C$_H$)									(Initial value: 0000 0000)

P6CR	I/O control for port P6 (specified for each bit)	0: 1: } As listed below	Write only

ADCCR		P6CR	
ADMD16	AINDS	0	1
0	0	Input port except for the analog input selected by SAIN	Output port except for the analog input selected by SAIN
1	0	Input port	Output port
0	1	Input port	Output port
1	1	Input port	Output port

Port P7 I/O control register

	7	6	5	4	3	2	1	0	
P7CR (000D$_H$)									(Initial value: 0000 0000)

P7CR	I/O control for port P7 (specified for each bit)	0: 1: } As listed below	Write only

ADCCR		P7CR	
ADMD16	AINDS	0	1
0	0	Input port	Output port
1	0	Input port except for the analog input selected by SAIN.	Output port except for the analog input selected by SAIN.
0	1	Input port	Output port
1	1	Input port	Output port

Figure 2-109. AD Conversion Register and Ports P6, P7 I/O Control Register

2.15.3 AD Converter operation

Apply the analog reference voltage high side to the VAREF pin; apply the analog reference voltage low side to the VASS pin. AD conversion is performed by dividing, using a ladder resistor, the reference voltage between VAREF and VASS into voltages corresponding to bits and comparing the divided voltage with analog input voltage.

Note: VAREF ≦ VDD Please meet this condition by all means.

(1) **Starting AD conversion Example with port P6 (AIN00 to AIN07)**

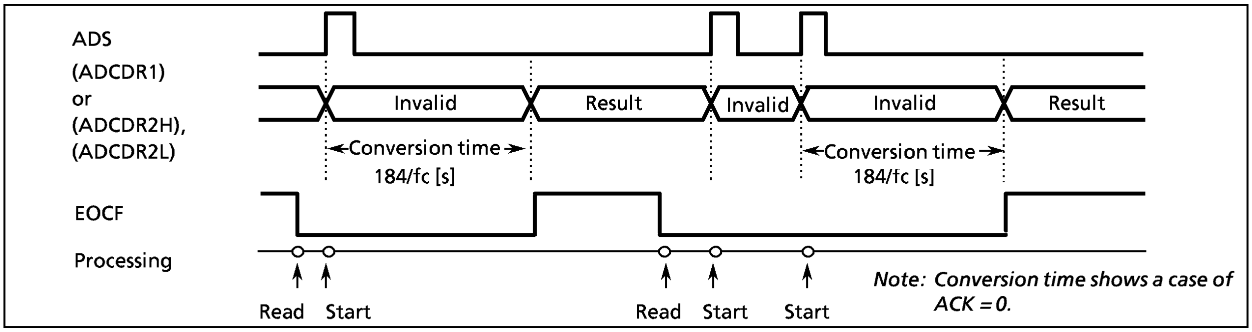
Before AD conversion, select a pin among analog input channel pins (AIN7 to AIN0) using SAIN (bits 3 to 0 in ADCCR). Zero-clear AINDS (bit 4 in ADCCR) and sets a channel to be used for analog input to 1 using port P6 I/O control register (P6CR).

Note: The pin that is not used as an analog input can be used as regular input/output pins. During conversion, do not perform output instruction to maintain a precision for all of the pins.

Sets the AD conversion time using ACK (bit 5 in ADCCR).
The AD conversion is started by setting 1 in ADS (bit 6 in ADCCR).
When ACK = 0, a minimum of 184/fc [s] (46 machine cycles) is required from AD conversion start to conversion result set in ADCDR1 or ADCDR2H/ADCDR2L. For example, when fc = 8 MHz, the AD conversion time is 23 μs. When AD conversion ends, EOCF (bit 7 in ADCCR) is set to 1 indicating conversion end.
Setting ADS to 1 during AD conversion initializes and starts conversion from the beginning again.

(2) **Reading AD converted value**

Read the conversion value stored in the AD conversion data register ADCDR1 or ADCDR2H/ADCDR2L after checking conversion end (EOCF = 1). Reading the conversion value automatically zero-clears EOCF. If EOCF is read during AD conversion, an undefined value is read.



Note: Conversion time shows a case of ACK = 0.

Figure 2-110. AD Conversion Timing Chart

(3) **AD conversion in STOP mode**

Entering STOP mode during AD conversion aborts AD conversion; the AD conversion value becomes undefined. Thus, after return from STOP mode, EOCF remains 0. Entering STOP mode after AD conversion end (EOCF = 1), the AD conversion value and EOCF status are retained.

Example: After the AIN04 pin in selected an analog input channel, perform AD conversion. Check EOCF, read the conversion value, store upper 2 bits at address 009E$_H$ in RAM; lower 8 bits at address 009F$_H$.

```
        ; AIN SELECT
        LD          (ADCCR), 00100100B      ;   Selects conversion time and AIN04.
        ; AD CONVERT START
        LD          (ADCCR), 01100100B
SLOOP:  TEST        (ADCCR). 7              ;   EOCF = 1 ?
        JRS         T, SLOOP
        ; RESULT DATA READ
        LD          (9EH), (ADCDR2H)
        LD          (9FH), (ADCDR2L)
```

(4) Notes for the current consumption on the stop mode when using an AD converter

Note 1: Current consumption value (I_{DD}) on stop mode on D. C. Characteristics chart is not including the value between $V_{AREF} – V_{ASS}$ (I_{REF}). TMP87CS48A does not have function to cut current between $V_{AREF} – V_{ASS}$ (I_{REF}). To cut IREF on stop mode, maintain V_{AREF} on open condition by external circuit, or same electrical potential of VASS.

Note 2: Turning to stop mode during the process of AD conversion (ADCCR EOCF = 0) aborts the operation though it does not cut electricity on analog comparator sometimes. Before turning to stop mode, check AD conversion end flag is "1". Moreover check EOCF after AD conversion is finished, and when EOCF terns to "1", read AD conversion values (ADCDR1, ADCR2H, ADCR2L) and turn to stop mode. Or if it has been turned to stop mode without reading AD conversion value, read them after stop mode has released since the values are maintained. Refer to flowchart 2-111 (a).

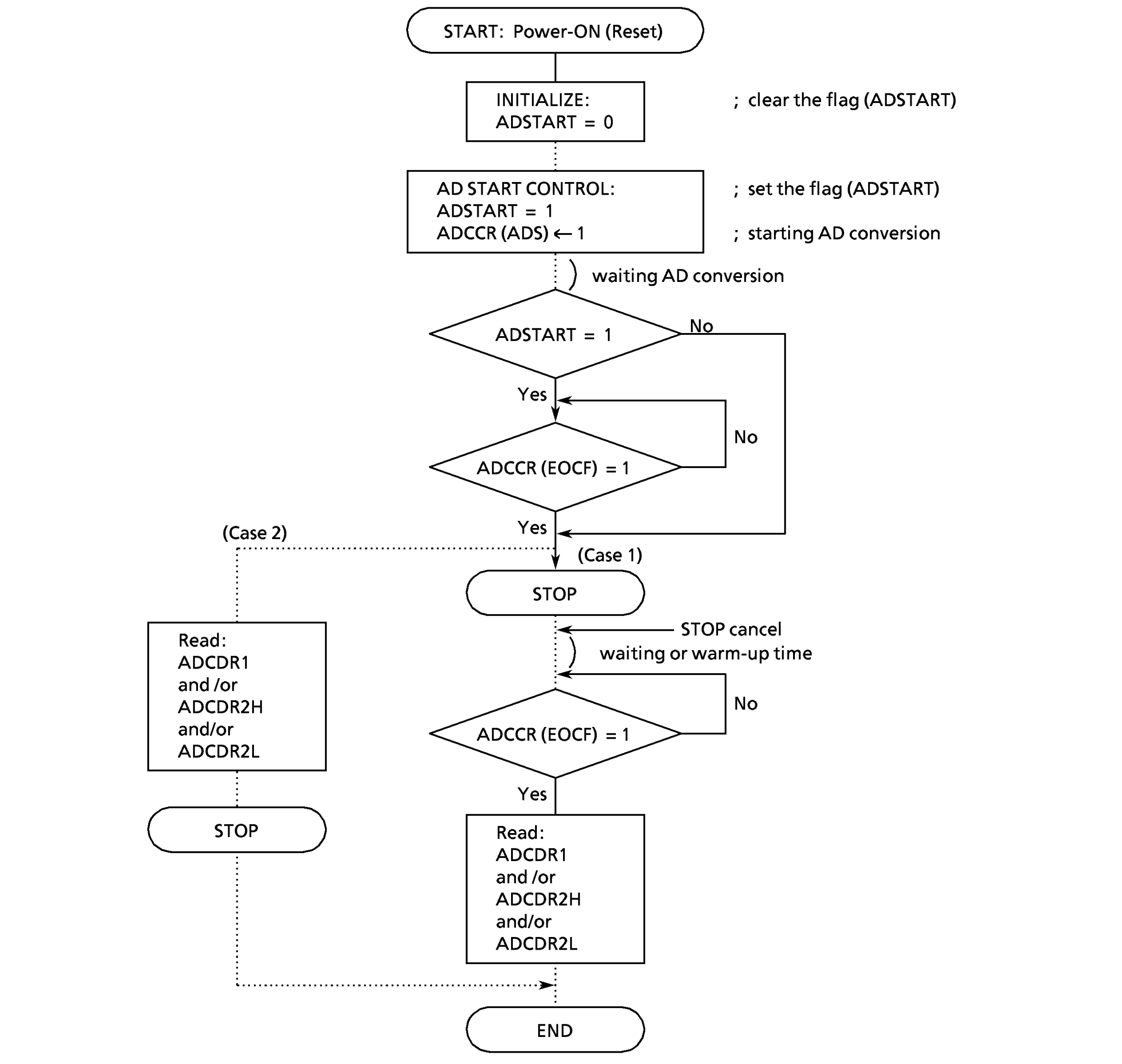
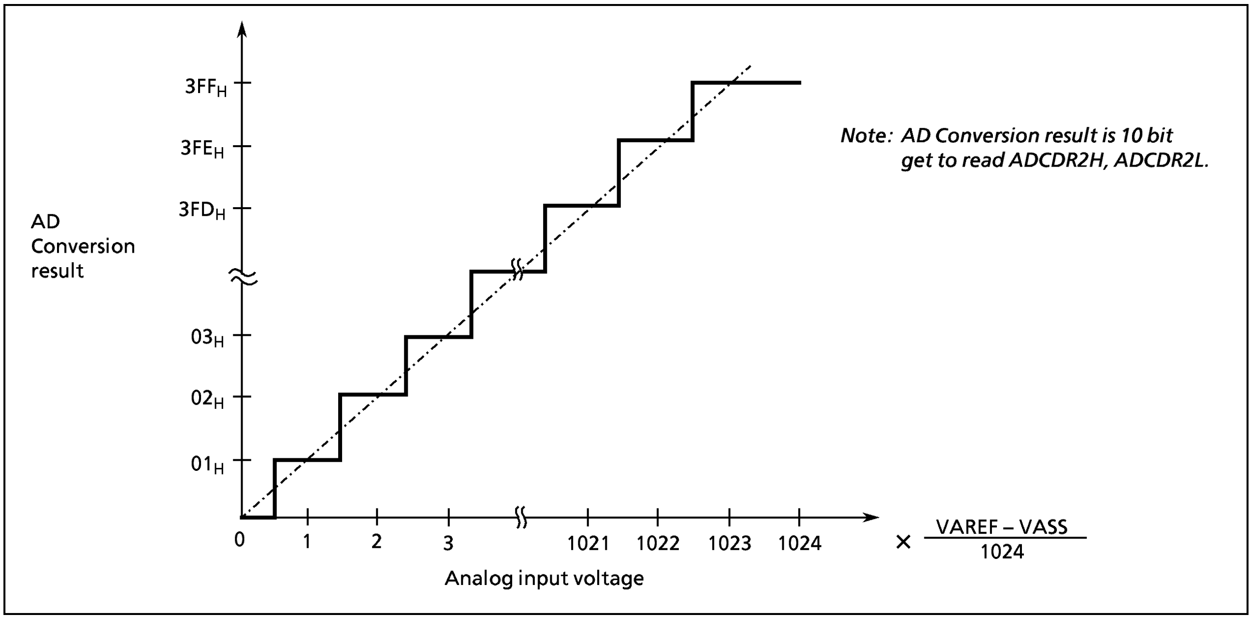


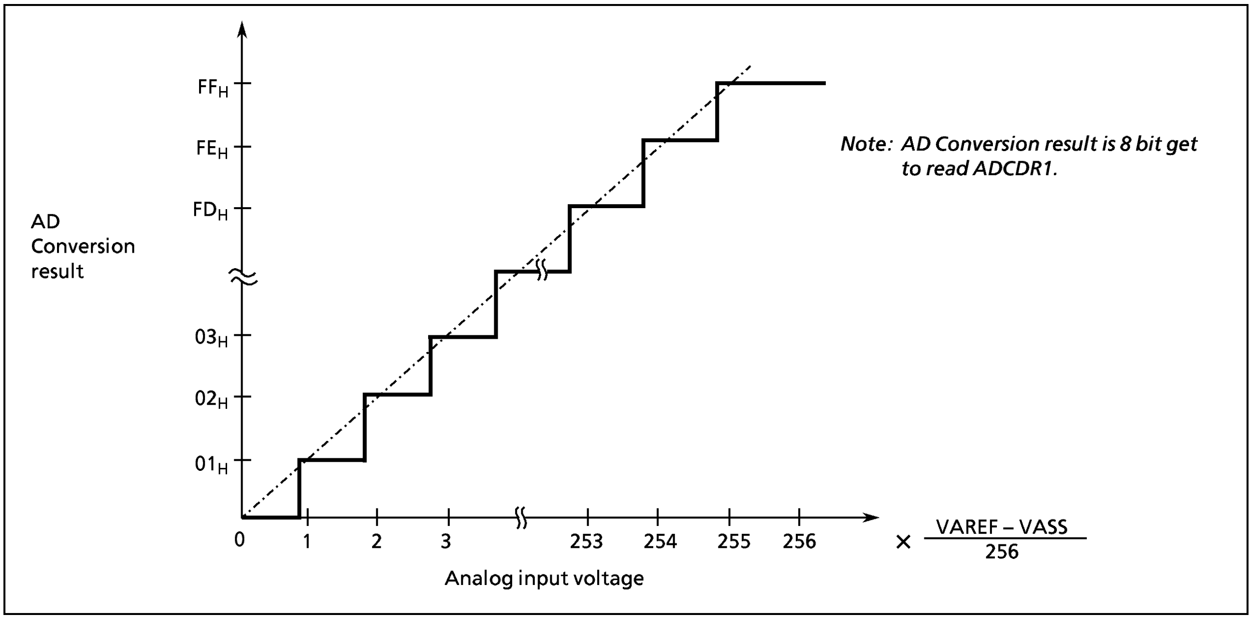
Figure 2-111 (a). Example Flow Chart for STOP Mode Control in the AD Converter System

(5) The relation between Analog Input Voltage and AD Conversion Result



Note: AD Conversion result is 10 bit get to read ADCDR2H, ADCDR2L.

Figure 2-111 (b). Analog Input Voltage vs AD Conversion Result (typ.)



Note: AD Conversion result is 8 bit get to read ADCDR1.

Figure 2-111 (c). Analog Input Voltage vs AD Conversion Result (typ.)

2.16 8-bit High-speed PWM ($\overline{\text{HPWM0}}$, $\overline{\text{HPWM1}}$)

TMP88CS48A features two channels for high-speed PWM. High-speed PWM outputs different waveforms by writing data in the each channel data register.
High-speed PWM uses P53 ($\overline{\text{HPWM0}}$) and P54 ($\overline{\text{HPWM1}}$). To use these pins for high-speed PWM, set the P53 and P54 output latches to 1.

2.16.1 Configuration

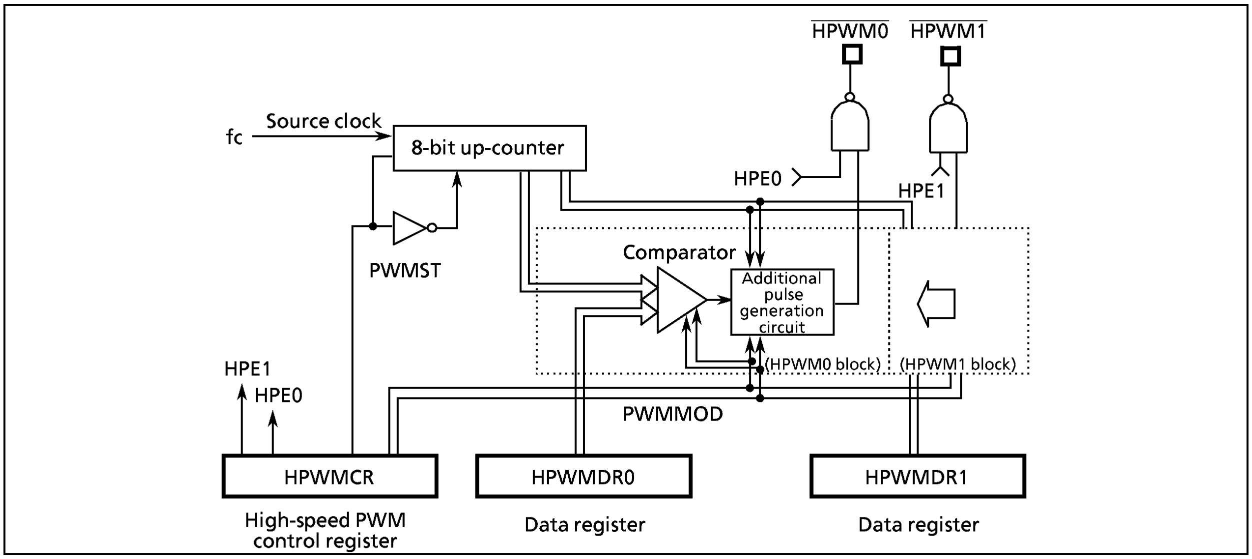


Figure 2-112. High-Speed PWM ($\overline{\text{HPWM0}}$ and $\overline{\text{HPWM1}}$)

2.16.2 Control

Control register

HPWMCR (00026$_H$)

7	6	5	4	3	2	1	0
HPE1	HPE0			PWMST		PWMMOD	

Write only (Initial value: 00** 0000)

PWMMOD	PWM mode select	00: Mode 0 (8 bits) 01: Mode 1 (7 bits) 10: Mode 2 (6 bits) 11: Unused	Write only
PWMST	RUN/STOP of 8-bit up-counter	0: STOP 1: RUN	
HPE0	HPWM0 output control	0: Disable 1: Enable	
HPE1	HPWM1 output control	0: Disable 1: Enable	

Note: bit 2 to 0

Data registers

HPWMDR0 (00027$_H$)

7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Write only (Initial value: **** ****)

HPWMDR1 (00028$_H$)

7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Write only (Initial value: **** ****)

Figure 2-113. Registers of High-Speed PWM

2.16.3 Operation

High-speed PWM is controlled by the control register (HPWMCR) and data registers (HPWMR0, 1). To write to these registers, set HPWMCR<PWMST> = 1 to enable setting. Setting HPWMCR<PWMST> = 0 resets the control registers, resetting high-speed PWM by software.

(1) **Operation mode**

High-speed PWM supports following three operating modes:

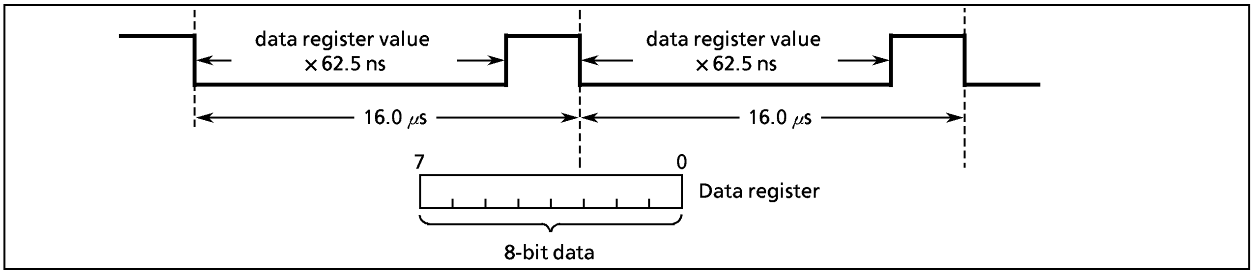
- 8-bit mode: (T = 2^8 × clock cycle, f≒32 kHz, f≒*64 kHz)
- 7-bit mode: (T = 2^7 × clock cycle, f≒64 kHz)
- 6-bit mode: (T = 2^6 × clock cycle, f≒128 kHz)

*Note: * indicates the value when the source clock (X1) operates at 16 MHz*

Set the operating mode using HPWMCR<PWMMOD>. The operating mode is common to the channels: two operating modes cannot be set simultaneously.

① 8-bit mode

8-bit mode generates a pulse with 16.0 μs cycle at a frequency of approximately 64 kHz (X1 = 16 MHz).



The minimum pulse width is 62.5 ns (data "1") and the maximum pulse width is 15.0 μs (data "F0").

Pulse width = 8-bit data × 62.5 ns

A wave cycle example is shown the figure 2-114. (The value is when X1 = 16 MHz)

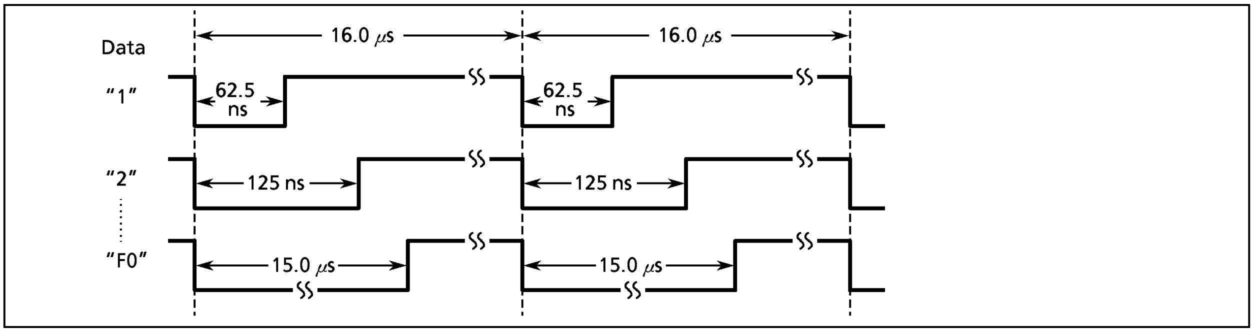
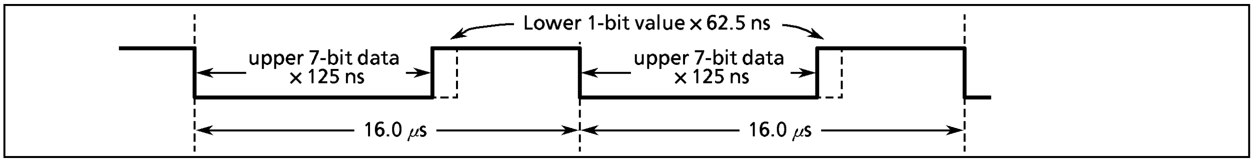


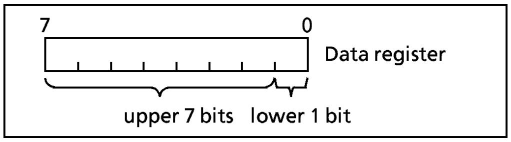
Figure 2-114. 8-bit mode

② 7-bit mode

7-bit mode generates a pulse with 16.0 μs cycle at a frequency of approximately 64 kHz (X1 = 16 MHz).


Lower 1-bit value × 62.5 ns
upper 7-bit data × 125 ns
upper 7-bit data × 125 ns
16.0 μs
16.0 μs

7-bit mode has 7-bits for a cycle (2^7 × 125 ns/cycle) and the lower 1 bit for a 62.5 ns resolution (1/2 cycle of source clock (X1)). When the lower 1 bit is "1", the additional 62.5 ns pulse is output. The minimum pulse width is 62.5 ns (data "1") and the maximum pulse width is 15.06255 μs (data "F1").



Pulse width = (Upper 7-bit data × 125 ns) + (Lower 1-bit data × 62.5 ns)

A wave cycle example is shown the Figure 2-115. (The value is when X1 = 8 MHz.)

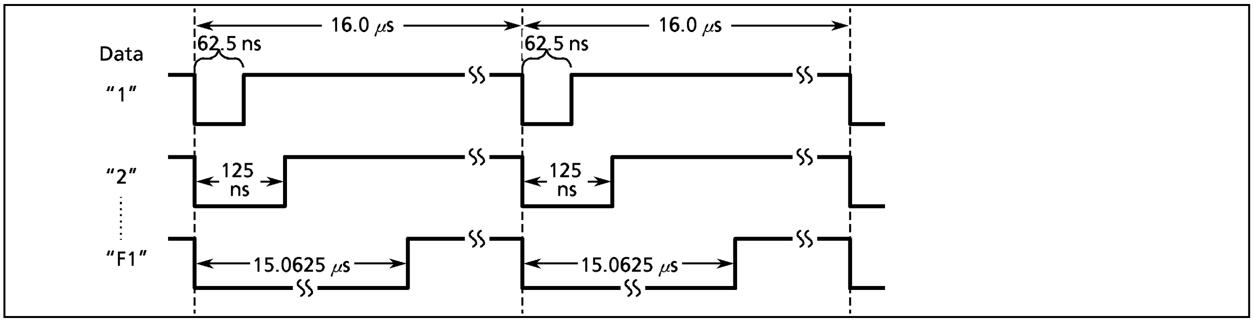
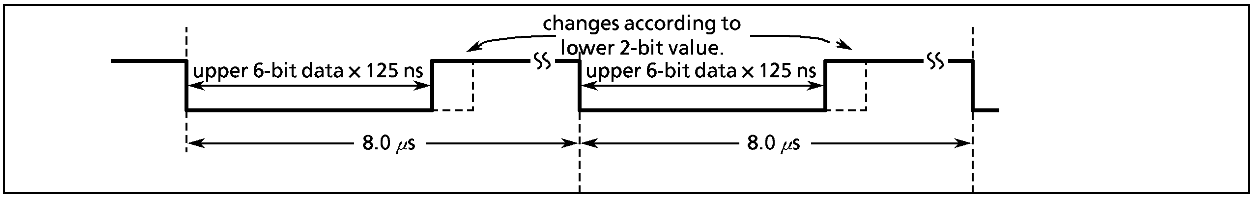


Figure 2-115. 7-bit mode

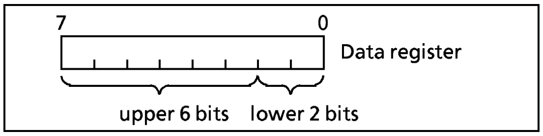
③ 6-bit mode

6-bit mode generates a pulse with 8.0 μs cycle at a frequency of approximately 128 kHz (X1 = 8 MHz).



6-bit mode has 6 bits for a cycle (2^6 × 125 ns/cycle) and simulates a 31.25 ns resolution by 2 cycle of the lower 2 bits.

The minimum pulse width is 31.25 ns (data "1") and the maximum pulse width is 7.625 μs (data "F3").



Pulse width = (Upper 6-bit data × 125 ns) + (* Lower 2-bit data)

* Equivalent time added for lower 2-bit data is shown below.

2-bit data	Equivalent time added
0 0	0 ns
0 1	31.25 ns
1 0	62.5 ns
1 1	93.75 ns

A wave cycle example is shown the Figure 2-116. (The value is when X1 = 8 MHz.)

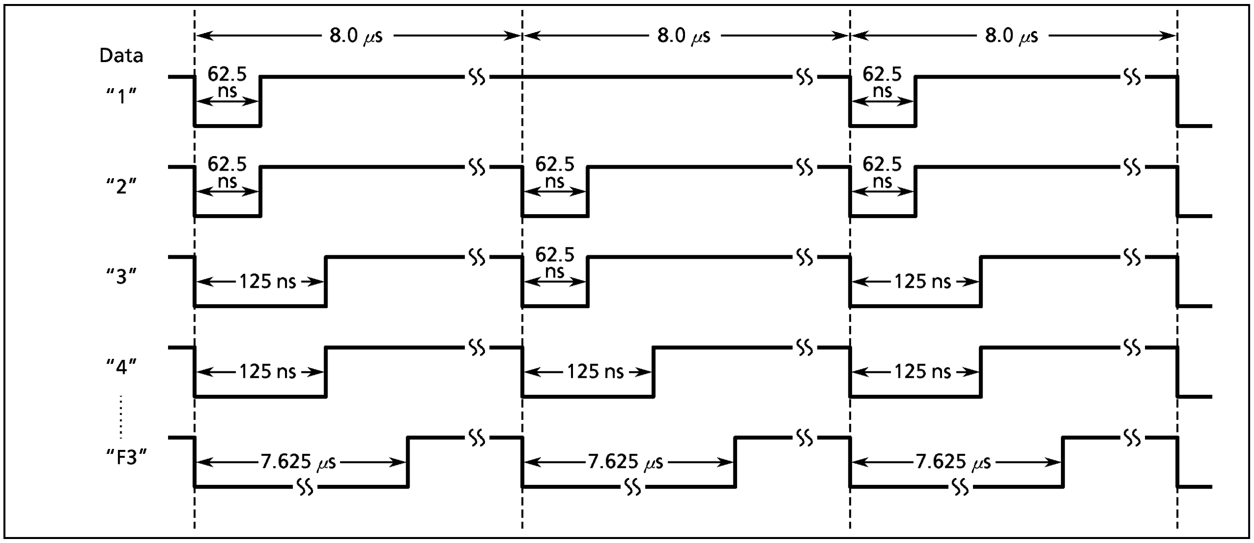
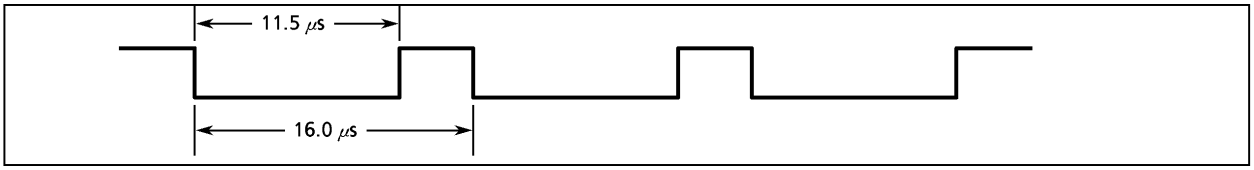


Figure 2-116. 6-bit mode

(2) **Output data setting**

Output data is set by writing to the data registers (HPWMDR0, 1).

Example: To output 11.5 μs wave with $\overline{\text{HPWM0}}$ in 7-bit mode with a source clock (X1) = 8 MHz.

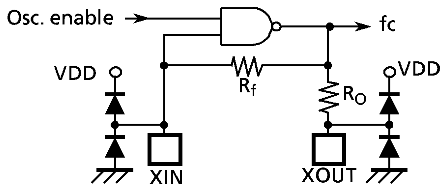
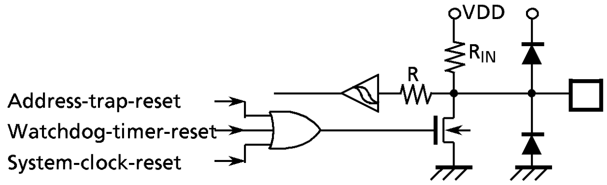
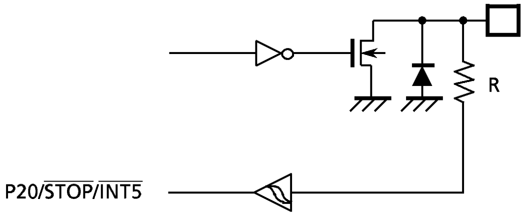
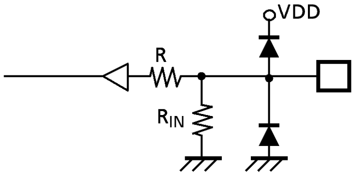


When the resolution in 7-bit mode is 62.5 ns, a 11.5 μs pulse is output by setting the following to HPWMDR0.

11.5 μs ÷ 62.5 ns = 184 = B8H

Input / Output Circuit

(1) Control pins
The input / output circuits of the 88CS48A are shown below.

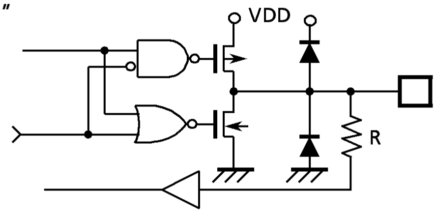
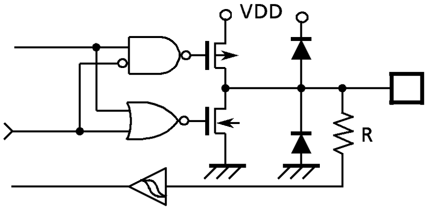
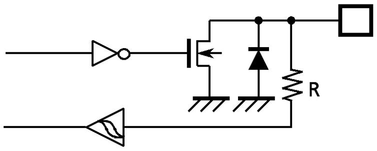
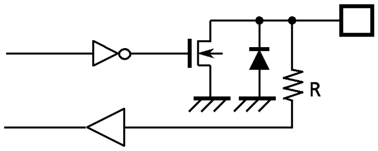
Control pin	I/O	Input / Output circuitry and Code	Remarks
XIN XOUT	Input Output	 Osc. enable fc VDD R_f R_O VDD XIN XOUT	Resonator connecting pins (high-frequency) R_f = 1.2 MΩ (typ.) R_O = 1.5 kΩ (typ.)
$\overline{\text{RESET}}$	I/O	 VDD R_{IN} R Address-trap-reset Watchdog-timer-reset System-clock-reset	Sink opendrain output Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 1 kΩ (typ.)
$\overline{\text{STOP}}$/INT5	Input		Hysteresis input R = 1 kΩ (typ.)
TEST	Input		Pull-down resistor R_{IN} = 70 kΩ (typ.) R = 1 kΩ (typ.)

Note 1: The TMP88PS49 does not have a pull-down resistor for TEST pin. Must be fixed to "L" level.

Note 2: Insert a protection diode between V_{SS} or V_{DD} as close to the package as possible.

(2) Input / Output ports

The input/output circuitries of the 88CS48A input/output ports are shown below.

Port	I/O	Input/Output circuitry and Code	Remarks
P0 P6 P7	I/O	initial "Hi-Z" 	Tri-state I/O R = 1 kΩ (typ.)
P1	I/O	initial "Hi-Z" 	Tri-state I/O Hysteresis input R = 1 kΩ (typ.)
P2 P4 P5	I/O	initial "Hi-Z" 	Sink open drain output R = 1 kΩ (typ.)
P3	I/O	initial "Hi-Z" 	Sink open drain output High current output R = 1 kΩ (typ.)

Note: Insert a protection diode between V_{SS} or V_{DD} as close to the package as possible.

Electrical Characteristics

Absolute Maximum Ratings

(V_{SS} = 0 V)

Parameter	Symbol	Pins	Ratings	Unit
Supply Voltage	V_{DD}		– 0.3 to 6.5	V
Input Voltage	V_{IN}		– 0.3 to V_{DD} + 0.3	V
Output Voltage	V_{OUT1}	Port P21, P22, $\overline{\text{RESET}}$, Tri-state port	– 0.3 to V_{DD} + 0.3	V
	V_{OUT2}	Port P20, Sink open drain port	– 0.3 to 5.5	V
Output Current	I_{OUT1}	Ports P1, P2, P4, P5, P6, P7	3.2	mA
	I_{OUT2}	Port P0	20	
	I_{OUT3}	Port P3	30	
Output Current	ΣI_{OUT1}	Ports P1, P2, P4, P5, P6, P7	120	mA
	ΣI_{OUT2}	Port P0	60	
	ΣI_{OUT3}	Port P3	120	
Power Dissipation [Topr = 70°C]	PD	TMP88CS48AN	600	mW
		TMP88CS48AF	350	
Soldering Temperature (time)	Tsld		260 (10 s)	°C
Storage Temperature	Tstg		– 55 to 125	°C
Operating Temperature	Topr		– 40 to 85	°C

Note: The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

Recommended Opeating Conditions

(V_{SS} = 0 V, Topr = – 40 to 85°C)

Parameter	Symbol	Pins	Conditions		Min	Max	Unit
Supply Voltage	V_{DD}		fc = 16 MHz	NORMAL mode	4.5	5.5	V
				IDLE mode			
				STOP mode			
Input High Voltage	V_{IH1}	Except hysteresis input	$V_{DD} \geqq 4.5$ V		$V_{DD} \times 0.70$	V_{DD}	V
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 4.5$ V		$V_{DD} \times 0.90$		
Input Low Voltage	V_{IL1}	Except hysteresis input	$V_{DD} \geqq 4.5$ V		0	$V_{DD} \times 0.30$	V
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 4.5$ V			$V_{DD} \times 0.10$	
Clock Frequency	fc	XIN, XOUT	V_{DD} = 4.5 to 5.5 V		8.0	16.0	MHz

Note 1: The recommended operating conditions for a device are operating conditions under which it can be guaranteed that the device will operate as specified. If the device is used under operating conditions other than the recommended operating conditions (supply voltage, operating temperature range, specified AC/DC values etc.), malfunction may occur. Thus, when designing products which include this device, ensure that the recommended operating conditions for the device are always adhered to.

Note 2: Clock frequency fc: The condition of supply voltage range is the value in NORMAL and IDLE modes.

DC Characteristics

(V_{SS} = 0 V, Topr = − 40 to 85°C)

Parameter	Symbol	Pins	Conditions	Min	Typ.	Max	Unit
Hysteresis Voltage	V_{HS}	Hysteresis inputs		–	0.9	–	V
Input Current	I_{IN1}	TEST	V_{DD} = 5.5 V V_{IN} = 5.5 V/0 V	–	–	± 2	μA
	I_{IN2}	Sink open drain, Tri-state ports					
	I_{IN3}	$\overline{\text{RESET}}$, $\overline{\text{STOP}}$					
Input Resistor (*)	R_{IN}	TEST with pull-down		20	70	170	kΩ
		$\overline{\text{RESET}}$		90	220	510	
Output Leakage Current	I_{OL}	Sink open drain, Tri-state ports	V_{DD} = 5.5 V, V_{OUT} = 5.5 V/0 V	–	–	± 2	μA
Output High Voltage	V_{OH}	Tri-state ports	V_{DD} = 4.5 V, I_{OH} = − 0.7 mA	4.1	–	–	V
Output Low Current	I_{OL1}	Except XOUT, Ports P0, P3.	V_{DD} = 4.5 V, V_{OL} = 0.4 V	–	1.6	–	mA
	I_{OL2}	Port P0	V_{DD} = 4.5 V, V_{OL} = 1.0 V	6	10	–	
	I_{OL3}	Port P3		12	20	–	
Supply Current in NORMAL Mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V fc = 16.0 MHz	–	32	40	mA
Supply Current in IDLE Mode				–	24	30	mA
Supply Current in STOP Mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V	–	0.5	20	μA

Note 1: Typical values show those at Topr = 25℃, V_{DD} = 5 V.

Note 2: Input Current I_{IN1}, I_{IN3}; The current through registor is not included, when the input resistor (pull-up or pull-down) is contained.

Note 3: IDD except I_{REF}.

AD Conversion Characteristics

(Topr = − 40 to 85°C)

Parameter	Symbol	Conditions	Min	Typ.	Max		Unit
					ADCDR1	ADCDR2	
						ACK = 1	
Analog Reference Voltage	V_{AREF}	$V_{AREF} - V_{ASS} \geqq 3.5$ V	V_{DD} − 1.0	—	V_{DD}		V
	V_{ASS}		V_{SS}	—	1.0		
Analog Input Voltage	V_{AIN}		V_{ASS}	—	V_{AREF}		V
Analog Supply Current	I_{REF}	V_{AREF} = 5.5 V, V_{ASS} = 0.0 V	—	0.5	1.0		mA
Non-Linearity Error		V_{DD} = 5.0 V, V_{SS} = 0.0 V V_{AREF} = 5.000 V V_{ASS} = 0.000 V	—	—	± 1	± 2	LSB
Zero Point Error			—	—	± 1	± 2	
Full Scale Error			—	—	± 1	± 2	
Total Error			—	—	± 2	± 4	

Note 1: ADCDR1: 8-bit AD conversion result (1LSB = ΔV_{AREF}/256)
ADCDR2: 10-bit AD conversion result (1LSB = ΔV_{AREF}/1024)

Note 2: Total error includes all errors except quantization error.

AC Characteristics

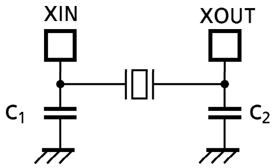
(V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = – 40 to 85°C)

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Machine Cycle Time	tcy	NORMAL mode	0.25	–	0.5	μs
		IDLE mode				
High Level Clock Pulse Width	t_{WCH}	For external clock operation	31.25	–	62.5	ns
Low Level Clock Pulse Width	t_{WCL}	(XIN input)				

Recommended Oscillating Conditions

(V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = – 40 to 85°C)

Parameter	Oscillator	Oscillation Frequency	Recommended Oscillator	Recommended Constant	
				C_1	C_2
High-frequency Oscillation	Ceramic Resonator	16 MHz	MURATA CSA16.00MXZ	5 pF	5 pF
			MURATA CST16.00MXW	built-in 5 pF	built-in 5 pF



High-frequency Oscillation

Note: An electrical shield by metal shield on the surface of IC package should be recommendable in order to prevent the device from the high electric fieldstress applied from CRT (Cathode Ray Tube) for continuous reliable operation.