CMOS 8-bit microcomputer

TMP87CH48U / DF

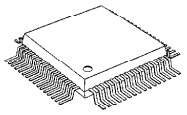
TMP87CH48 is a low power, high-speed and high-performance 8-bit single chip microcomputer, including large capacity ROM/RAM, input/output ports, a multiple timer counter, serial interfaces (UART, I2C bus, and SIO), four 12-bit PWM outputs, a 10-bit A/D converter and two oscillators.

PART No.	ROM	RAM	Package	OTP
TMP87CH48U	16 K bytes	512 bytes	LQFP64-P-1010-0.50	TMP87PH48U
TMP87CH48DF			QFP64-P-1414-0.80A	TMP87PH48DF

FEATURES

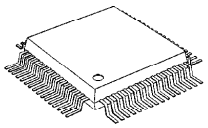
- 8-bit single chip microcomputer TLCS-870 series
- Minimum instruction execution time: 0.5 μs (at 8 MHz), 122 μs (at 32.768 kHz)
- 412 basic machine instructions: 129 types
- 15 interrupt sources (External: 6, Internal: 9)
  - All sources have independent latches each, and nested interrupt control is available.
  - Edge-selectable external interrupts with noise reject.
  - High-speed task switching by register bank changeover
- Input/output ports (56 pins)
  - High current output: 8 pins (typ.20 mA), LED direct drive
- 16-bit timer counters : 2 channels
  - Timer, Event counter, PPG (Programmable Pulse Generator) output, Pulse width measurement, External trigger timer, Window modes
- 8-bit timer counters : 2 channels
  - Timer, Event counter, Capture (Pulse width/duty measurement) PWM (Changeable pulse width) output, PDO (ProgrammableDivider Output)
- Time base timer (Interrupt frequency : 1 to 16384 Hz)
- Divider output functions (Frequency : 1 to 8 kHz)
- Watchdog timer
  - Interrupt/Reset output (programmable)
- D/A conversion (changeable pulse width) output
  - 12-bit resolution : 4 channels
- UART : 1 channel (parity·framing·overrun error detection)
- Serial bus interface (SBI-ver. B) 1 channel (I2C bus or clock synchronous SIO)
- 10-bit successive approximation type A/D converter
  - Analog input : 16 channels
  - Conversion time : 24.5 μs or 98 μs (at 8 MHz)
- Clock oscillation circuit : Two circuits
  - Single/Dual clock modes (Initial mode is always set to a single clock mode.)
- Low consumption power (Five modes)
  - STOP mode : Oscillation stop (Battery/Capacitor back-up). Port output hold/high-impedance.
  - SLOW mode : Low consumption power operation by low-frequency clock
  - IDEL1 mode : CPU stops, and only peripheral hardware operates using high-frequency clock. Release by interrupts (CPU restarts).
  - IDEL2 mode : CPU stops, and only peripheral hardware operates using high or low-frequency clock). Release by interrupts.
  - SLEEP mode : CPU stops, and only peripheral hardware operates using low-frequency clock. Release by interrupts.
- Operation voltage : 2.7 to 5.5 V at 4.2 MHz/32.768 kHz, 4.5 to 5.5 V at 8 MHz/32.768 kHz
- Emulation pod : BM87CH48U0A

LQFP64-P-1010-0.50



TMP87CH48U
TMP87PH48U

QFP64-P-1414-0.80A

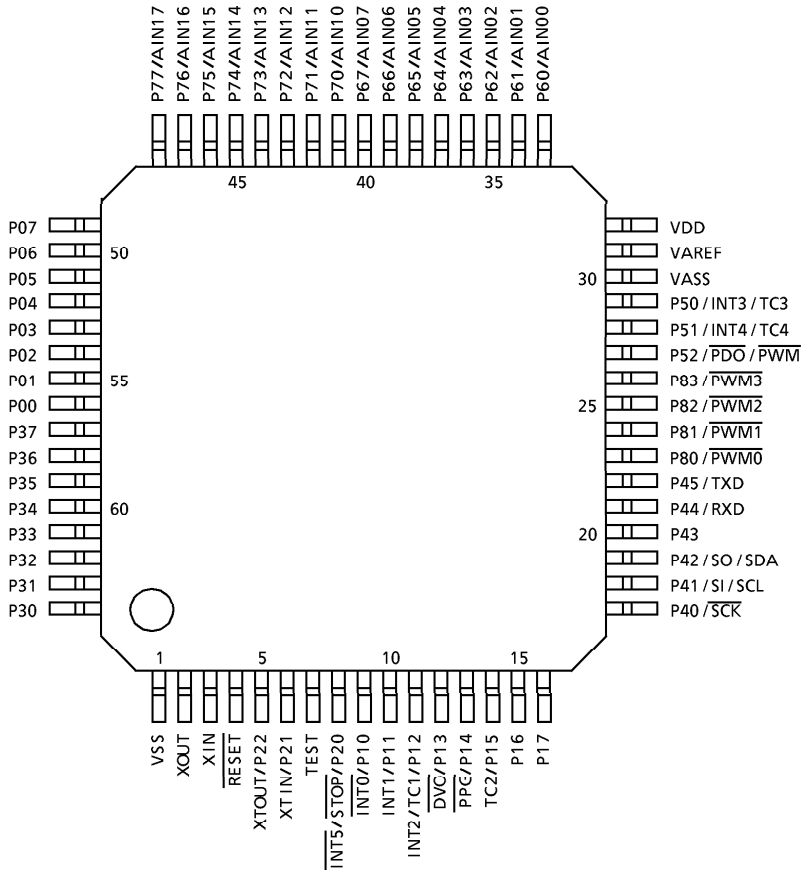


TMP87CH48DF
TMP87PH48DF

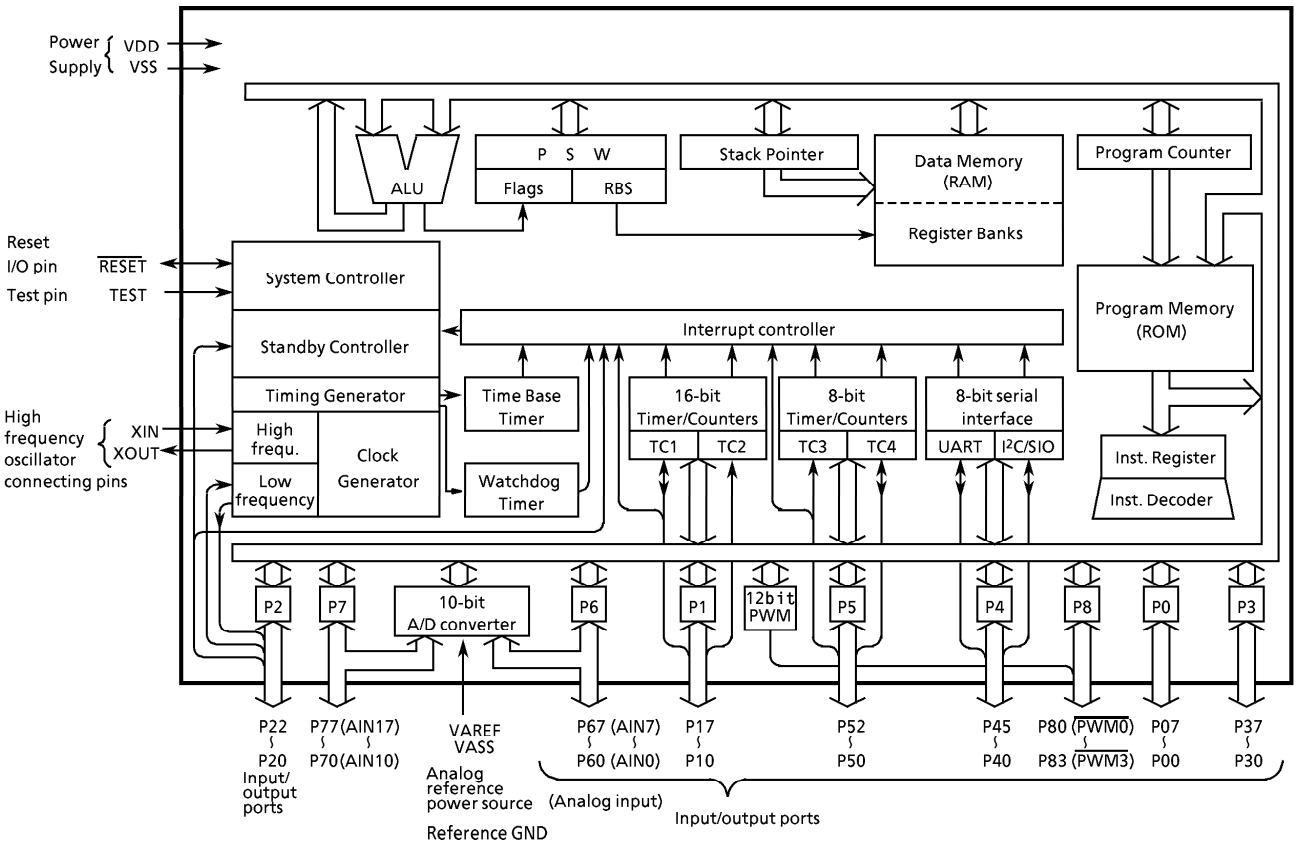
Purchase of TOSHIBA I² C components conveys a license under the Philips I² C Patent Rights to use these components in an I² C system, provided that the system conforms to the I² C Standard Specification as defined by Philips.

PIN ASSIGNMENTS (TOP VIEW)

LQFP64-P-1010-0.50
QFP64-P-1414-0.80A



BLOCK DIAGRAM



PIN FUNCTION

PIN NAME	Input / Output	FUNCTIONS	
P07 to P00	I/O	8-bit programmable input/output port (tri-state). Each bit of these ports can be individually configured as an input or an output under software control. When used as an external interrupt input or a timer counter input, the latch must be set to input mode. When used as PPG output or a divider output, the output latch must be set to "1".	
P17, P16	I/O		
P15 (TC2)	I/O (Input)		Timer counter 2 input
P14 ($\overline{PPG}$)	I/O (Output)		Programmable pulse generator output
P13 ($\overline{DVO}$)			Divider output
P12 (INT2 / TC1)	I/O (Input)		External interrupt input 2 or Timer counter 1 input
P11 (INT1)			External interrupt input 1
P10 ($\overline{INT0}$)			External interrupt input 0
P22 (XTOUT)	I/O (Output)	3-bit input/output port. When used as an input port, an oscillator connecting pin, an external interrupt input or STOP mode release input of P20, the output latch must be set to "1".	Low frequency oscillator connecting pins (32.768 kHz). For inputting external clock, XTIN is used and XTOUT is opened.
P21 (XTIN)	I/O (Input)		
P20 ($\overline{INT5}$ / $\overline{STOP}$)			External interrupt input 5 or STOP mode release signal input
P37 to P30	I/O	8-bit input/output port (high current output). When used as an input port, the output latch must be set to "1".	
P45 (TxD)	I/O (Output)	8-bit input/output port. When used as an input port, a serial interface pin, the output latch must be set to "1".	UART serial data output (send)
P44 (RxD)	I/O (Input)		UART serial data output (receive)
P43	I/O		
P42 (SO / SDA)	I/O (Output, I/O)		SIO serial data output or I^2C bus data input/output
P41 (SI / SCL)	I/O (Input, I/O)		SIO serial data output or I^2C bus clock input/output
P40 ($\overline{SCK}$)	I/O (I/O)		SIO serial clock input/output
	I/O		
P52 ($\overline{PWM}$ / $\overline{PDO}$)	I/O (Output)	3-bit input/output port. When used as an input port, PWM output, high-speed PWM output, a programmable divider output, an external interrupt input or timer counter input, the output latch must be set to "1".	8-bit PWM output or 8-bit programmable divider output
P51 (INT4 / TC4)	I/O (Input)		External interrupt input 4 or Timer counter 4 input
P50 (INT3 / TC3)			External interrupt input 3 or Timer counter 3 input
P67 (AIN7) to P60 (AIN0)	I/O	8-bit programmable input/output port (tri-state). Each bit of these ports can be individually configured as an input or an output under software control. When used as an analog input, the latch must be set to an analog input mode by P6CR and P7CR.)	A/D converter analog input
P77 (AIN17) to P70 (AIN10)			
P83 ($\overline{PWM3}$) to P80 ($\overline{PWM0}$)	I/O (Output)	4-bit programmable input/output port (tri-state). Each bit of the port can be individually configured as an input or an output under software control. An input or an output is determined by setting P8CR.	DA conversion (PWM) output ($\overline{PWM3}$ to $\overline{PWM0}$)
XIN, XOUT	Input, Output	Oscillator connecting pins for high frequency clock. For inputting external clock, XIN is used and XOUT is opened.	
$\overline{RESET}$	I/O	Reset signal input or watchdog timer output/address-trap-reset output/system-clock-reset output.	
TEST	Input	Test pin for outgoing test. Be externally tied to low.	
VDD, VSS	Power Supply	+ 5 V, 0 V (GND)	
VAREF, VASS		A/D conversion analog reference voltage, Reference GND.	

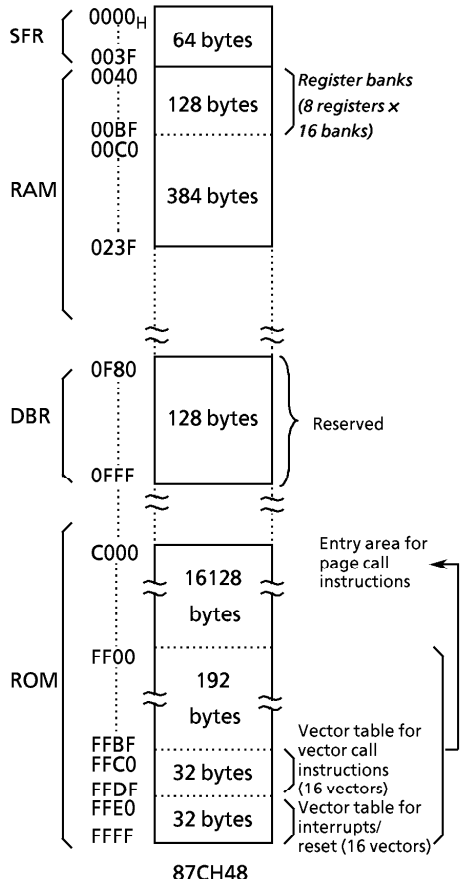
OPERATIONAL DESCRIPTION

1. CPU CORE FUNCTIONS

The CPU core consists of a CPU, a system clock controller, an interrupt controller, and a watchdog timer. This section provides a description of the CPU core, the program memory (ROM), the data memory (RAM), and the reset circuit.

1.1 Memory Address Map

The TLCS-870 Series is capable of addressing 64K bytes of memory. Figure 1-1 shows the memory address maps of the 87CH48. In the TLCS-870 Series, the memory is organized 4 address spaces (ROM, RAM, SFR, and DBR). It uses a memory mapped I/O system, and all I/O registers are mapped in the SFR/DBR address spaces. There are 16 banks of general-purpose registers. The register banks are also assigned to the first 128 bytes of the RAM address space.



Note :

ROM ; Read Only Memory includes :
- *Program memory*

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 Register includes :
- *SIO data buffer*

Figure 1-1. Memory Address Maps

1.2 Program Memory (ROM)

The 87CH48 has a 16K × 8-bit (addresses $C000_H$ to $FFFF_H$) of program memory (mask programmed ROM). Addresses $FF00_H$ to $FFFF_H$ in the program memory can also be used for special purposes.

(1) **Interrupt / Reset** vector table (addresses $FFE0_H$ to $FFFF_H$)
This table consists of a reset vector and 15 interrupt vectors (2 bytes/vector). These vectors store a reset start address and interrupt service routine entry addresses.

(2) Vector table for **vector call** instructions (addresses $FFC0_H$ to $FFDF_H$)
This table stores call vectors (subroutine entry address, 2 bytes/vector) for the vector call instructions [CALLV n]. There are 16 vectors. The CALLV instruction increases memory efficiency when utilized for frequently used subroutine calls (called from 3 or more locations).

(3) Entry area (addresses $FF00_H$ to $FFFF_H$) for **page call** instructions
This is the subroutine entry address area for the page call instructions [CALLP n]. Addresses $FF00_H$ to $FFBF_H$ are normally used because address $FFC0_H$ to $FFFF_H$ are used for the vector tables.

Programs and fixed data are stored in the program memory. The instruction to be executed next is read from the address indicated by the current contents of the program counter (PC). There are relative jump and absolute jump instructions. The concepts of page or bank boundaries are not used in the program memory concerning any jump instruction.

Example: The relationship between the jump instructions and the PC.

① 5-bit PC-relative jump [JRS cc, $ + 2 + d]
E8C4H: JRS T, $ + 2 + 08H
When JF = 1, the jump is made to $E8CE_H$, which is 08_H added to the contents of the PC. (The PC contains the address of the instruction being executed + 2; therefore, in this case, the PC contents are $E8C4_H + 2 = E8C6_H$.)

② 8-bit PC-relative jump [JR cc, $ + 2 + d]
E8C4H : JR Z, $ + 2 + 80H
When ZF = 1, the jump is made to $E846_H$, which is $FF80_H$ (– 128) added to the current contents of the PC.

③ 16-bit absolute jump [JP a]
E8C4H : JP 0C235H
An unconditional jump is made to address $C235_H$. The absolute jump instruction can jump anywhere within the entire 64K-bytes space.

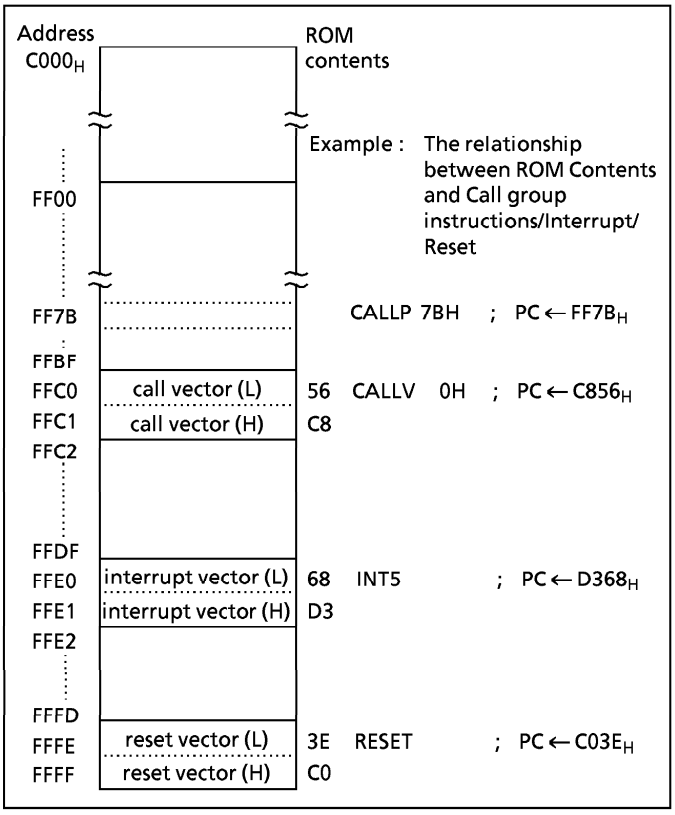


Figure 1-2. Program Memory Map

In the TLCS-870 Series, the same instruction used to access the data memory (e.g. [LD A, (HL)]) is also used to read out fixed data (ROM data) stored in the program memory. The register-offset PC-relative addressing (PC + A) instructions can also be used, and the code conversion, table look-up and n-way multiple jump processing can easily be programmed.

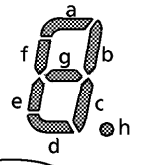
Example 1 : Loads the ROM contents at the address specified by the HL register pair contents into the accumulator (87CH48 : HL≧C000$_H$):

```
LD      A, (HL)          ; A←ROM (HL)
```

Example 2 : Converts BCD to 7-segment code (common anode LED). When A = 05$_H$, 92$_H$ is output to port P3 after executing the following program:

```
        ADD     A, TABLE – $ – 4     ; P3 ←ROM (TABLE + A)
        LD      (P3), (PC + A)
        JRS     T, SNEXT
TABLE : DB      0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0D8H, 80H, 98H
SNEXT :
```

Notes : "$" is a header address of ADD instruction.
DB is a byte data difinition instruction.



Example 3 : N-way multiple jump in accordance with the contents of accumulator (0≦A≦3):

```
SHLC    A            ; if A = 00H then PC←C234H
JP      (PC + A)       if A = 01H then PC←C378H
                       if A = 02H then PC←DA37H
                       if A = 03H then PC←E1B0H
DW      0C234H, 0C378H, 0DA37H, 0E1B0H
```

Note : DW is a word data definition instruction.

```
SHLC A
JP (PC + A)
34
C2
78
C3
37
DA
B0
E1
```

1.3 Program Counter (PC)

The program counter (PC) is a 16-bit register which indicates the program memory address where the instruction to be executed next is stored. After reset, the user defined reset vector stored in the vector table (addresses FFFF$_H$ and FFFE$_H$) is loaded into the PC ; therefore, program execution is possible from any desired address. For example, when C0$_H$ and 3E$_H$ are stored at addresses FFFF$_H$ and FFFE$_H$, respectively, the execution starts from address C03E$_H$ after reset.

The TLCS-870 Series utilizes pipelined processing (instruction pre-fetch); therefore, the PC always indicates 2 addresses in advance. For example, while a 1-byte instruction stored at address C123$_H$ is being executed, the PC contains C125$_H$.

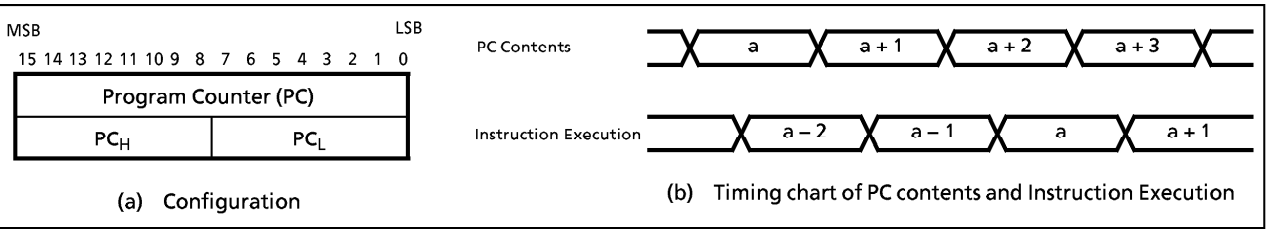


Figure 1-3. Program Counter

1.4 Data Memory (RAM)

The 87CH48 has a 512 × 8-bit (addresses 0040_H to $023F_H$) of data memory (static RAM). Figure 1-4 shows the data memory map.

Addresses 0000_H to $00FF_H$ are used as a direct addressing area to enhance instructions which utilize this addressing mode; therefore, addresses 0040_H to $00FF_H$ in the data memory can also be used for user flags or user counters. General-purpose register banks (8 registers × 16 banks) are also assigned to the 128 bytes of addresses 0040_H to $00BF_H$. Access as data memory is still possible even when being used for registers. For example, when the contents of the data memory at address 0040_H is read out, the contents of the accumulator in the bank 0 are also read out. The stack can be located anywhere within the data memory except the register bank area. The stack depth is limited only by the free data memory size. For more details on the stack, see section "1.7 Stack and Stack Pointer".

With the CH48, programs in data memory cannot be executed. If the program counter indicates a data memory address addresses 0040_H to $023F_H$ for 87CH48 an address-trap-reset is generated due to bus error. (Output from the $\overline{RESET}$ pin goes low.)

Example 1 : If bit 2 at data memory address $00C0_H$ is "1", 00_H is written to data memory at address $00E3_H$; otherwise, FF_H is written to the data memory at address $00E3_H$.

```
        TEST    (00C0H).2       ;  if (00C0H)2 = 0 then jump
        JRS     T,SZERO
        CLR     (00E3H)         ;  (00E3H) ← 00H
        JRS     T,SNEXT
SZERO : LD      (00E3H), 0FFH   ;  (00E3H) ← FFH
SNEXT :
```

Example 2 : Increments the contents of data memory at address $00F5_H$, and clears to 00_H when 10_H is exceeded.

```
        INC     (00F5H)         ;  (00F5H) ← (00F5H) + 1
        AND     (00F5H), 0FH    ;  (00F5H) ← (00F5H)∧0FH
```

The data memory contents become unstable when the power supply is turned on; therefore, the data memory should be initialized by an initialization routine.

> Note that the general-purpuse registers are mapped in the RAM ; therefore, *do not clear RAM at the current bank addresses.*

Example 1 : Clears RAM to "00_H" except the bank 0 (87CH48)

```
        LD      HL, 0048H       ;  Sets start address to HL register pair
        LD      A, H            ;  Sets initial data (00H) to A register
        LD      BC, 01F7H       ;  Sets number of byte to BC register pair
SRAMCLR : LD    (HL+), A
        DEC     BC
        JRS     F, SRAMCLR
```

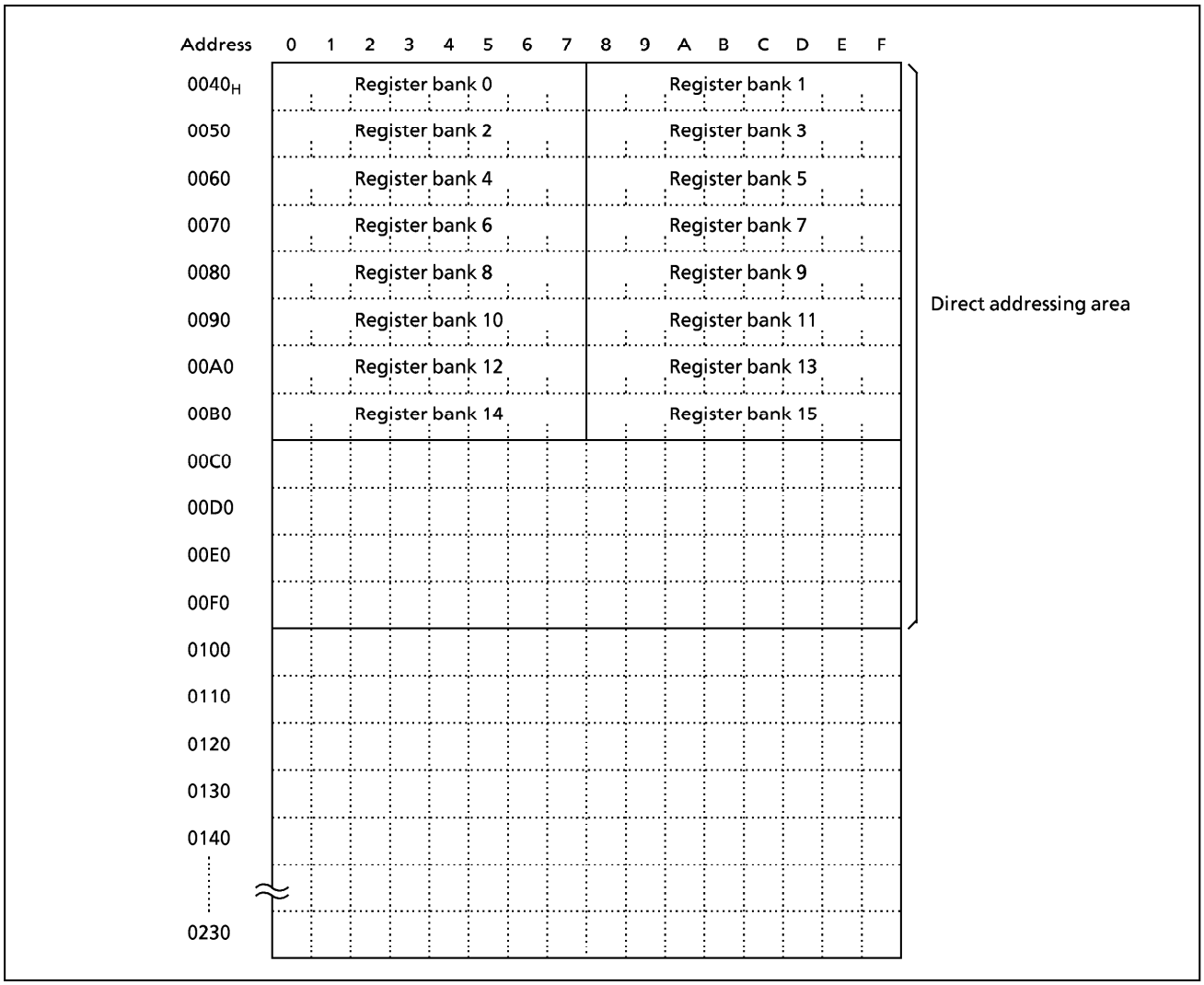


Figure 1-4. Data Memory Map

1.5 General-purpose Register Banks

General-purpose registers are mapped into addresses 0040_H to $00BF_H$ in the data memory as shown in Figure 1-4. There are 16 register banks, and each bank contains eight 8-bit registers W, A, B, C, D, E, H, and L. Figure 1-5 shows the general-purpose register bank configuration.

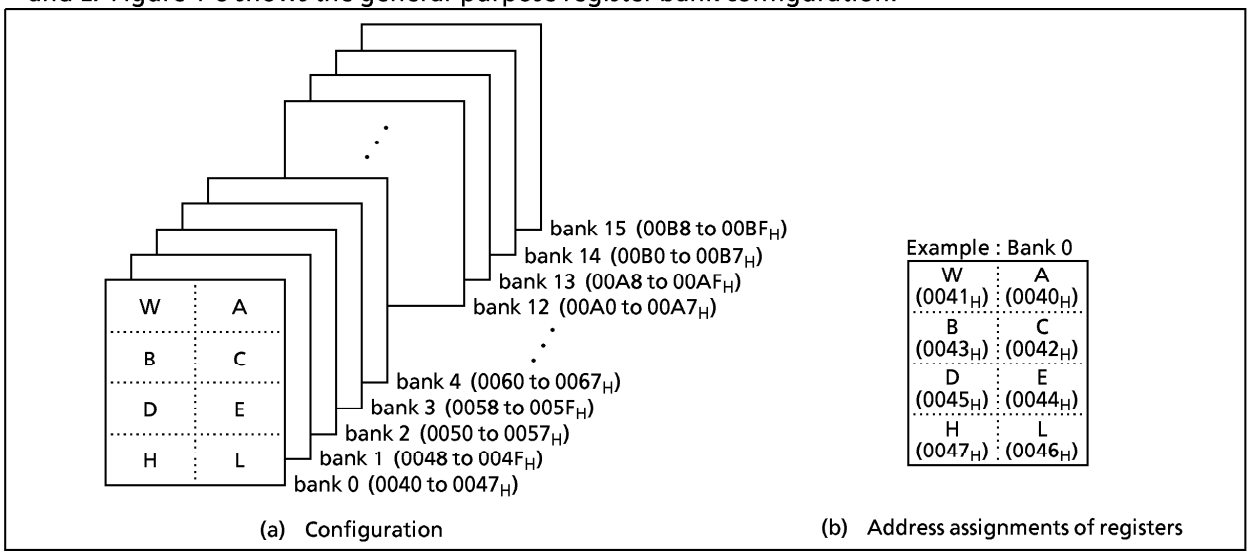


Figure 1-5. General-purpose Register Banks

In addition to access in 8-bit units, the registers can also be accessed in 16-bit units as the register pairs WA, BC, DE, and HL. Besides its function as a general-purpose register, the register also has the following functions:

(1) **A, WA**

The A register functions as an 8-bit accumulator and WA the register pair functions as a 16-bit accumulator (W is high byte and A is low byte). Registers other than A can also be used as accumulators for 8-bit operations.

```
Examples :  ①   ADD  A, B          ; Adds B contents to A contents and stores the result into A.
            ②   SUB  WA, 1234H     ; Subtracts 1234H from WA contents and stores the result into WA.
            ③   SUB  E, A          ; Subtracts A contents from E contents, and stores the result into E.
```

(2) **HL, DE**

The HL and DE specify a memory address. The HL register pair functions as data pointer (HL) /index register (HL + d) /base register (HL + C), and the DE register pair function as a data pointer (DE). The HL also has an auto-post- increment and auto-pre-decrement functions. This function simplifies multiple digit data processing, software LIFO (last-in first-out) processing, etc.

```
Example 1 :  ①   LD   A, (HL)       ; Loads the memory contents at the address specified by HL into A.
             ②   LD   A, (HL + 52H) ; Loads the memory contents at the address specified by the value
                                      obtained by adding 52H to HL contents into A.
             ③   LD   A, (HL + C)   ; Loads the memory contents at the address specified by the value
                                      obtained by adding the register C contents to HL contents into A.
             ④   LD   A, (HL + )    ; Loads the memory contents at the address specified by HL into A.
                                      Then increments HL.
             ⑤   LD   A, ( – HL)    ; Decrements HL. Then loads the memory contents at the address
                                      specified by new HL into A.
```

The TLCS-870 Series can transfer data directly memory to memory, and operate directly between memory data and memory data. This facilitates the programming of block processing.

Example 2 : Block transfer

```
        LD      m           ; m = n – 1 (n : Number of bytes to transfer)
        LD      HL, DSTA    ; Sets destination address to HL
        LD      DE, SRCA    ; Sets source address to DE
SLOOP : LD      (HL), (DE)  ; (HL) ← (DE)
        INC     HL          ; HL ← HL + 1
        INC     DE          ; DE ← DE + 1
        DEC     B           ; B ← B – 1
        JRS     F, SLOOP    ; if B ≧ 0 then loop
```

(3) **B, C, BC**

Registers B and C can be used as 8-bit buffers or counters, and the BC register pair can be used as a 16-bit buffer or counter. The C register functions as an offset register for register-offset index addressing (refer to example 1 ③ above) and as a divisor register for the division instruction [DIV gg, C].

Example 1 : Repeat processing

```
        LD      B, n        ; Sets n as the number of repetitions to B
SREPEAT : processing        ;     (n + 1 times processing)
        DEC     B
        JRS     F, SREPEAT
```

Example 2 : Unsigned integer division (16-bit ÷ 8-bit)

```
        DIV     WA, C       ; Divides the WA contents by the C contents, places the
                            ; quotient in A and the remainder in W.
```

The general-purpose register banks are selected by the 4-bit register bank selector (RBS). During reset, the RBS is initialized to "0". The bank selected by the RBS is called the current bank.
Together with the flag, the RBS is assigned to address $003F_H$ in the SFR as the program status word (PSW). There are 3 instructions [LD RBS, n], [PUSH PSW], [POP PSW] to access the PSW. The PSW can be also operated by the memory access instruction.

Example 1 : Incrementing the RBS

```
        INC     (003FH)     ; RBS ← RBS + 1
```

Example 2 : Reading the RBS

```
        LD      A, (003FH)  ; A ← PSW (A3-0 ← RBS, A7-4 ← Flags)
```

Highly efficient programming and high-speed task switching are possible by using bank changeover to save registers during interrupt and to transfer parameters during subroutine processing.
During interrupt, the PSW is automatically saved onto the stack. The bank used before the interrupt was accepted is restored automatically by executing an interrupt return instruction [RETI]/[RETN] ; therefore, there is no need for the RBS save/restore software processing.
The TLCS-870 Series supports a maximum of 15 interrupt sources. One bank is assigned to the main program, and one bank can be assigned to each source. Also, to increase the efficiency of data memory usage, assign the same bank to interrupt sources which are not nested.

Example: Saving /restoring registers during interrupt task using bank changeover.

```
PINT1 : LD      RBS, n      ; RBS ← n (Bank changeover)
        Interrupt processing
        RETI                ; Maskable interrupt return (Bank restoring)
```

1.6 Program Status Word (PSW)

The program status word (PSW) consists of a register bank selector (RBS) and four flags, and the PSW is assigned to address $003F_H$ in the SFR.

The RBS can be read and written using the memory access instruction (e. g. [LD A, (003FH)], [LD (003FH), A], however the flags can only be read. When writing to the PSW, the change specified by the instruction is made without writing data to the flags. For example, when the instruction [LD (003FH), 05H] is executed, "5" is written to the RBS and the JF is set to "1", but the other flags are not affected.

[PUSH PSW] and [POP PSW] are the PSW access instructions.

1.6.1 Register Bank Selector (RBS)

The register bank selector (RBS) is a 4-bit register used to select general-purpose register banks. For example, when RBS = 2, bank 2 is currently selected. During reset, the RBS is initialized to "0".

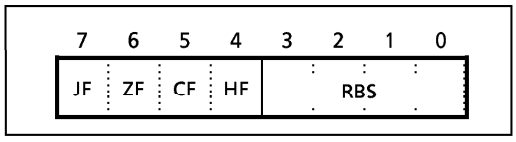


Figure 1-6. PSW (Flags, RBS) Configuration

1.6.2 Flags

The flags are configured with the upper 4 bits : a zero flag, a carry flag, a half carry flag and a jump status flag. The flags are set or cleared under conditions specified by the instruction. These flags except the half carry flag are used as jump condition "cc" for conditional jump instructions [JR cc, $ + 2 + d]/[JRS cc, $ + 2 + d]. After reset, the jump status flag is initialized to "1", other flags are not affected.

(1) **Zero flag (ZF)**

The ZF is set to "1" if the operation result or the transfer data is 00_H (for 8-bit operations and data transfers)/0000_H (for 16-bit operations); otherwise the ZF is cleared to "0".

During the bit manipulation instructions [SET, CLR, and CPL], the ZF is set to "1" if the contents of the specified bit is "0"; otherwise the ZF is cleared to "0".

This flag is set to "1" when the upper 8 bits of the product are 00_H during the multiplication instruction [MUL], and when 00_H for the remainder during the division instruction [DIV]; otherwise it is cleared to "0".

(2) **Carry flag (CF)**

The CF is set to "1" when a carry out of the MSB (most significant bit) of the result occurred during addition or when a borrow into the MSB of the result occurred during subtraction; otherwise the CF is cleared to "0". During division, this flag is set to "1" when the divisor is 00_H (divided by zero error), or when the quotient is 100_H or higher (overflow error); otherwise it is cleared. The CF is also affected during the shift/rotate instructions [SHLC, SHRC, ROLC, and RORC]. The data shifted out from a register is set to the CF.

This flag is also a 1-bit register (a boolean accumulator) for the bit manipulation instructions.

Set/clear/complement are possible with the CF manipulation instructions.

Example1 : Bit manipulation

```
LD      CF, (0007H) . 5      ; (0001H)2 ← (0007H)5 ∀ (009AH)0
XOR     CF, (009AH) . 0
LD      (0001H) . 2, CF
```

Example2 : Arithmetic right shift

```
LD      CF, A . 7            ; A ← A / 2
RORC    A
```

(3) **Half carry flag (HF)**

The HF is set to "1" when a carry occurred between bits 3 and 4 of the operation result during an 8-bit addition, or when a borrow occurred from bit 4 into bit 3 of the result during an 8-bit subtraction; otherwise the HF is cleared to "0". This flag is useful in the decimal adjustment for BCD operations (adjustments using the [DAA r], or [DAS r] instructions).

Example : BCD operation
(The A becomes 47_H after executing the following program when A = 19_H, B = 28_H)

```
ADD     A, B        ; A←41H, HF←1
DAA     A           ; A←41H + 06H = 47H (decimal-adjust)
```

(4) **Jump status flag (JF)**

Zero or carry information is set to the JF after operation (e. g. INC, ADD, CMP, TEST).
The JF provides the jump condition for conditional jump instructions [JRS T/F, $ + 2 + d], [JR T/F, $ + 2 + d] (T or F is a condition code). Jump is performed if the JF is "1" for a true condition (T), or the JF is "0" for a false condition (F).
The JF is set to "1" after executing the load/exchange/swap/nibble rotate/jump instruction, so that [JRS T, $ + 2 + d] and [JR T, $ + 2 + d] can be regarded as an unconditional jump instruction.

Example : Jump status flag and conditional jump instruction

```
INC     A
JRS     T, SLABLE1      ; Jump when a carry is caused by the immediately
 ⋮                        preceding operation instruction.
LD      A, (HL)
JRS     T, SLABLE2      ; JF is set to "1" by the immediately preceding
 ⋮                        instruction, making it an unconditional jump
                          instruction.
```

Example : The accumulator and flags become as shown below after executing the following instructions when the WA register pair, the HL register pair, the data memory at address $00C5_H$, the carry flag and the half carry flag contents being "$219A_H$", "$00C5_H$", "$D7_H$", "1" and "0", respectively.

Instruction	Acc. after execution	Flag after execution			
		JF	ZF	CF	HF
ADDC A, (HL)	72	1	0	1	1
SUBB A, (HL)	C2	1	0	1	0
CMP A, (HL)	9A	0	0	1	0
AND A, (HL)	92	0	0	1	0
LD A, (HL)	D7	1	0	1	0
ADD A, 66H	00	1	1	1	1

Instruction	Acc. after execution	Flag after execution			
		JF	ZF	CF	HF
INC A	9B	0	0	1	0
ROLC A	35	1	0	1	0
RORC A	CD	0	0	0	0
ADD WA, 0F508H	16A2	1	0	1	0
MUL W, A	13DA	0	0	1	0
SET A.5	BA	1	1	1	0

1.7 Stack and Stack Pointer

1.7.1 Stack

The stack provides the area in which the return address or status, etc. are saved before a jump is performed to the processing routine during the execution of a subroutine call instruction or the acceptance of an interrupt. On a subroutine call instruction [CALL a] / [CALLP n] / [CALLV n], the contents of the PC (the return address) is saved; on an interrupt acceptance, the contents of the PC and the PSW are saved (the PSW is pushed first, followed by PC_H and PC_L). Therefore, a subroutine call occupies two bytes on the stack; an interrupt occupies three bytes.
When returning from the processing routine, executing a subroutine return instruction [RET] restores the contents to the PC from the stack; executing an interrupt return instruction [RETI] / [RETN] restores the contents to the PC and the PSW (the PC_L is popped first, followed by PC_H and PSW).
The stack can be located anywhere within the data memory space except the register bank area, therefore the stack depth is limited only by the free data memory size.

1.7.2 Stack Pointer (SP)

The stack pointer (SP) is a 16-bit register containing the address of the next free locations on the stack.

The SP is post-decremented when a subroutine call or a push instruction is executed, or when an interrupt is accepted; and the SP is pre-incremented when a return or a pop instruction is executed. Figure 1-8 shows the stacking order.

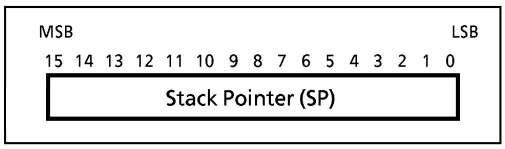


Figure 1-7. Stack Pointer

The SP is not initialized hardware-wise but requires initialization by an initialize routine (sets the highest stack address). [LD SP, mn], [LD SP, gg] and [LD gg, SP] are the SP access instructions (mn ; 16-bit immediate data, gg ; register pair).

Example 1 : To initialize the SP

```
LD      SP, 013FH       ; SP←013F_H
```

Example 2 : To read the SP

```
LD      HL, SP          ; HL←SP
```

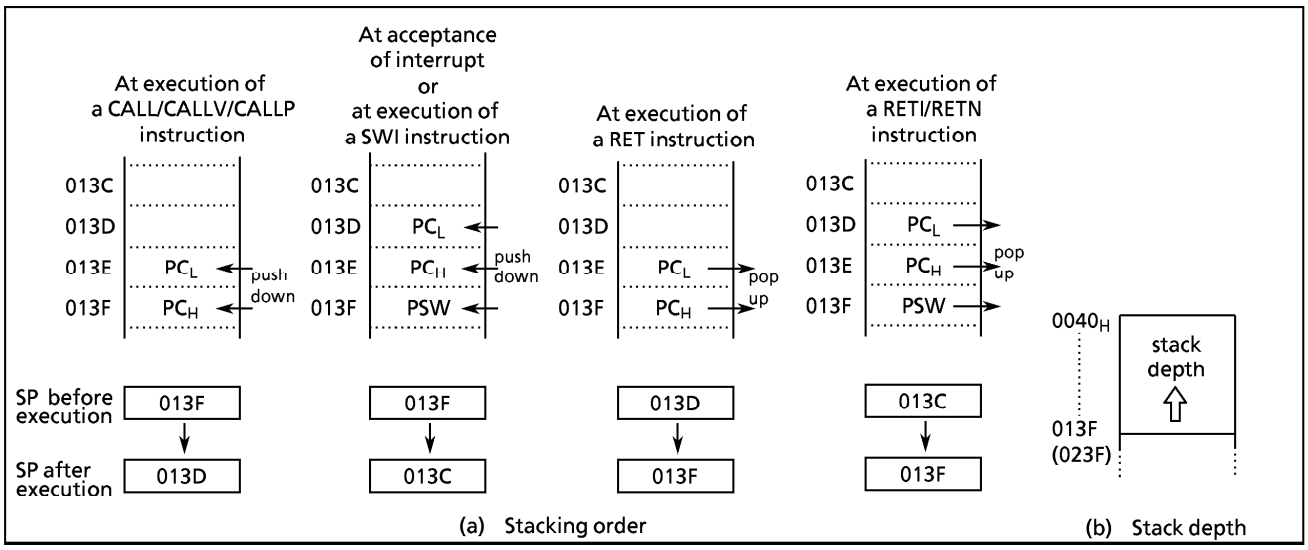


Figure 1-8. Stack

1.8 System Clock Controller

The system clock controller consists of a clock generator, a timing generator, and a stand-by controller.

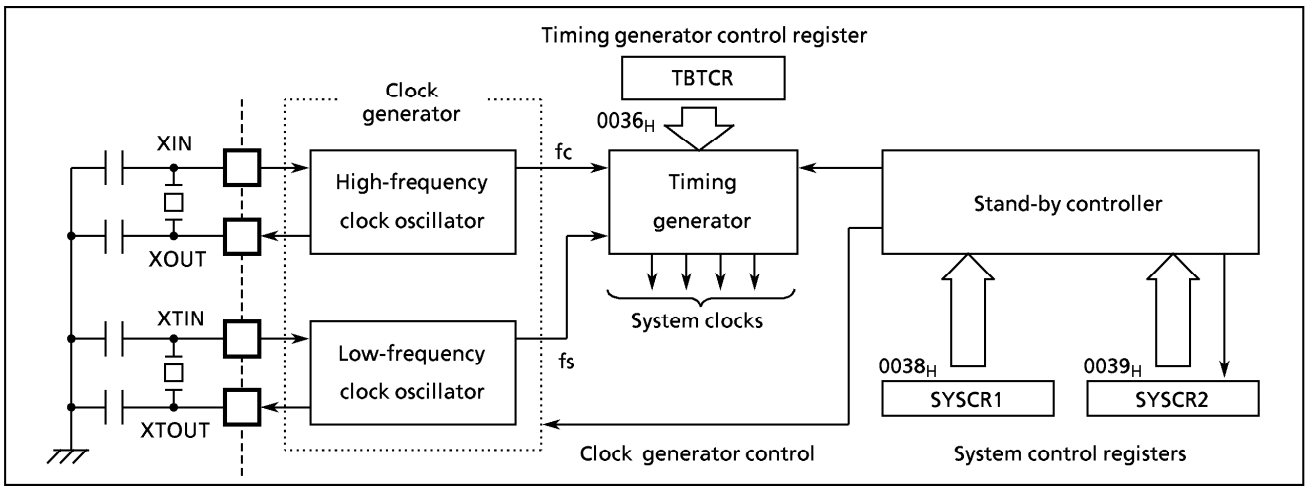


Figure 1-9. System Clock Controller

1.8.1 Clock Generator

The clock generator generates the basic clock which provides the system clocks supplied to the CPU core and peripheral hardware. It contains two oscillation circuits: one for the high-frequency clock and one for the low-frequency clock. Power consumption can be reduced by switching of the system clock controller to low-power operation based on the low-frequency clock.

The high-frequency (fc) and low-frequency (fs) clocks can be easily obtained by connecting a resonator between the XIN/XOUT and XTIN/XTOUT pins, respectively. Clock input from an external oscillator is also possible. In this case, external clock is applied to the XIN/XTIN pin with the XOUT/XTOUT pin not connected. The 87CH48 is not provided an RC oscillation.

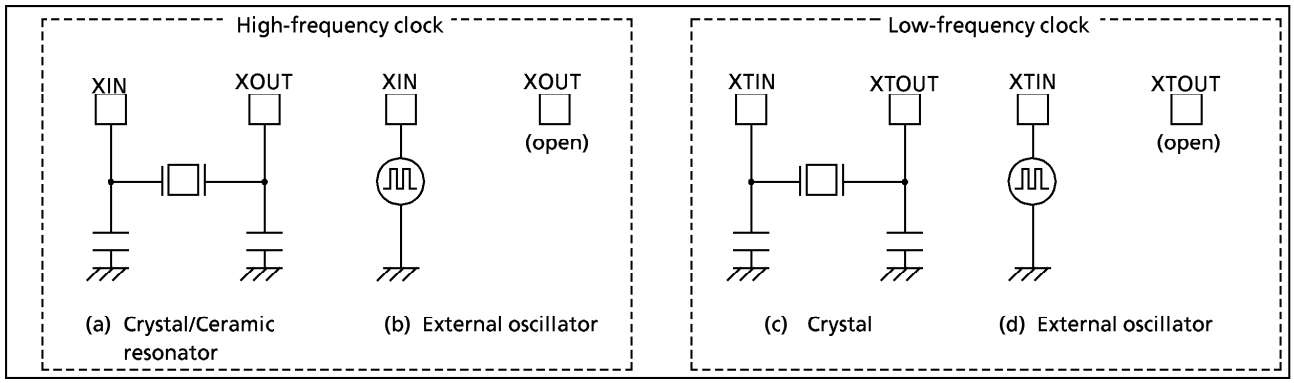
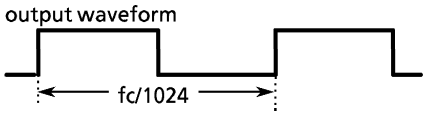


Figure 1-10. Examples of Resonator Connection

Note : Accurate Adjustment of the Oscillation Frequency:
Although no hardware to externally and directly monitor the basic clock pulse is not provided, the oscillation frequency can be adjusted by making the program to output fixed frequency pulses to the port while disabling all interrupts and monitoring this pulse. With a system requiring adjustment of the oscillation frequency, the adjusting program must be created beforehand.

Example: To output the high-frequency oscillation frequency adjusting monitor pulse to P13 ($\overline{DVO}$) pin.

```
SFCCHK:  LD   (P1CR), 00001000B  ; Configures port P13 as an output
         SET  (P1).3             ; P13 output latch ← 1
         LD   (TBTCR), 11100000B ; Enables divider output
         JRS  T, $               ; Loops endless
```



1.8.2 Timing Generator

The timing generator generates from the basic clock the various system clocks supplied to the CPU core and peripheral hardware. The timing generator provides the following functions :

① Generation of main system clock
② Generation of divider output ($\overline{DVO}$) pulses
③ Generation of source clocks for time base timer
④ Generation of source clocks for watchdog timer
⑤ Generation of internal source clocks for timer/counters TC1 – TC4
⑥ Generation of internal clocks for serial interfaces SIO1 and SIO2
⑦ Generation of warm-up clocks for releasing STOP mode
⑧ Generation of a clock for releasing reset output

(1) Configuration of Timing Generator

The timing generator consists of a 21-stage divider with a divided-by-4 prescaler, a main system clock generator, and machine cycle counters. An input clock to the 7th stage of the divider depends on the operating mode and DV7CK (bit 4 in TBTCR) shown in Figure 1-11 as follows.

During reset and at releasing STOP mode, the divider is cleared to "0", however, the prescaler is not cleared.

① In the single-clock mode
A divided-by-256 of high-frequency clock (fc/2^8) is input to the 7th stage of the divider.

② In the dual-clock mode
During NORMAL2 or IDLE2 mode (SYSCK = 0), an input clock to the 7th stage of the divider can be selected either "fc/2^8" or "fs" with DV7CK.
During SLOW or SLEEP mode (SYSCK = 1), fs is automatically input to the 7th stage. To input clock to the 1st stage is stopped ; output from the 1st to 6th stages is also stopped.

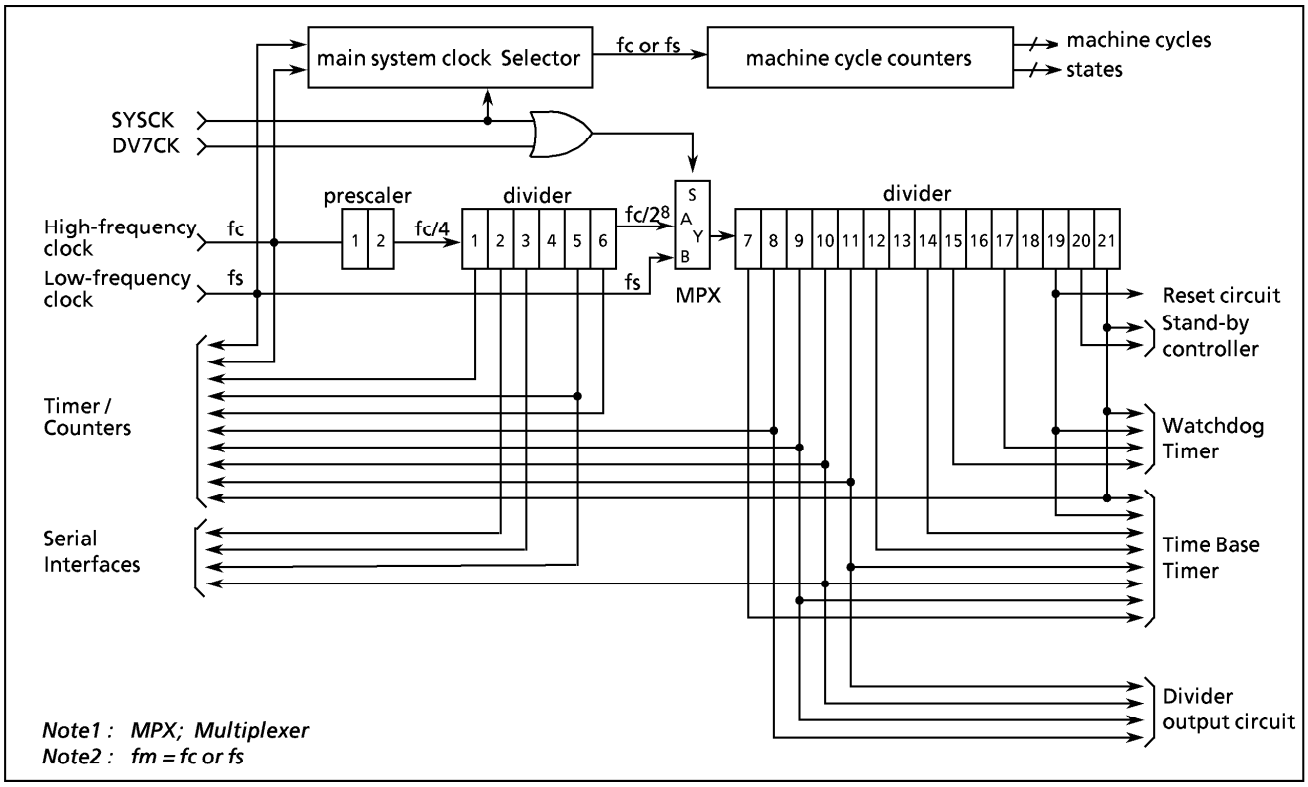


Figure 1-11. Configuration of Timing Generator

TBTCR (0036_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 : fc/2^8 [Hz] 1 : fs	R/W

*Note1 : fc ; high-frequency clock [Hz], fs ; low-frequency clock [Hz], * ; don't care*
Note 2 : Do not set DV7CK to "1" in the single-clock mode.
Note 3 : Do not set DV7CK to "1" before low-frequency clock is stable in the dual-clock mode.

Figure 1-12. Timing Generator Control Register

(2) Machine Cycle

Instruction execution and peripheral hardware operation are synchronized with the main system clock. The minimum instruction execution unit is called an "machine cycle". There are a total of 10 different types of instructions for the TLCS-870 Series: ranging from 1-cycle instructions which require one machine cycle for execution to 10-cycle instructions which require 10 machine cycles for execution.

A machine cycle consists of 4 states (S0 to S3), and each state consists of one main system clock.

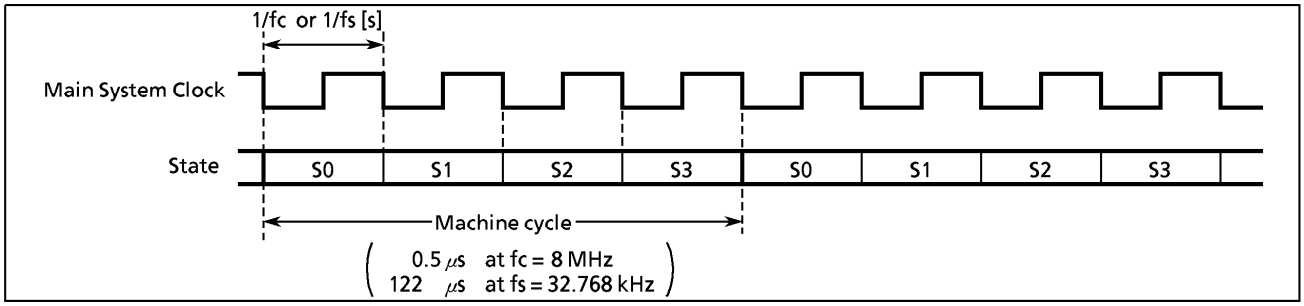


Figure 1-13. Machine Cycle

1.8.3 Stand-by Controller

The stand-by controller starts and stops the oscillation circuits for the high-frequency and low-frequency clocks, and switches the main system clock. There are two operating modes: single-clock and dual-clock. These modes are controlled by the system control registers (SYSCR1, SYSCR2).

Figure 1-14 shows the operating mode transition diagram and Figure 1-15 shows the system control registers. Either the single-clock or the dual-clock mode can not be selected by an option during reset, because the 87CH48 /87PH48 don't have option. The 87CH48/87PH48 are placed in the single-clock mode during reset.

(1) **Single-clock mode**

Only the oscillation circuit for the high-frequency clock is used, and P21 (XTIN) and P22 (XTOUT) pins are used as input/output ports. In the single-clock mode, the machine cycle time is 4/fc [s] (0.5 μs at fc = 8 MHz).

① NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock. In the case where the single-clock mode has been selected as an option, the 87CH48 are placed in this mode after reset.

② IDLE1 mode

In this mode, the internal oscillation circuit remains active, and the CPU and the watchdog timer are halted; however, on-chip peripherals remain active (operate using the high-frequency clock). IDLE1 mode is started by setting IDLE bit in the system control register 2 (SYSCR2), and IDLE1 mode is released to NORMAL1 mode by an interrupt request from on-chip peripherals or external interrupt inputs. When IMF (interrupt master enable flag) is "1" (interrupt enable), the execution will resume upon acceptance of the interrupt, and the operation will return to normal after the interrupt service is completed. When IMF is "0" (interrupt disable), the execution will resume with the instruction which follows IDLE mode start instruction.

③ STOP1 mode

In this mode, the internal oscillation circuit is turned off, causing all system operations to be halted. The internal status immediately prior to the halt is held with the lowest power consumption during this mode. The output status of all output ports can be set to either output hold or high-impedance under software control.

STOP1 mode is started by setting STOP bit in the system control register 1 (SYSCR1), and STOP1 mode is released by an input (either level-sensitive or edge-sensitive can be programmably selected) to the $\overline{\text{STOP}}$ pin. After the warming-up period is completed, the execution resumes with the next instruction which follows the STOP mode start instruction.

(2) **Dual-clock mode**

Both high-frequency and low-frequency oscillation circuits are used in this mode. Pins P21 (XTIN) and P22 (XTOUT) cannot be used as input/output ports. The main system clock is obtained from the high-frequency clock in NORMAL2 and IDLE2 modes, and is obtained from the low-frequency clock in SLOW and SLEEP modes. The machine cycle time is 4/fc [s] (0.5 μs at fc = 8 MHz) in NORMAL2 and IDLE2 modes, and 4/fs [s] (122 μs at fs = 32.768 kHz) in SLOW and SLEEP modes.

Note that the 87CH48/87PH48 are placed in the single-clock mode during reset. To use the dual-clock mode, the low-frequency oscillator should be turned on by executing [SET (SYSCR2).XTEN] instruction.

① NORMAL2 mode

In this mode, the CPU core operates using the high-frequency clock. On-chip peripherals operate using the high-frequency clock and/or low-frequency clock.

② SLOW mode

This mode can be used to reduce power-consumption by turning off oscillation of the high-frequency clock. The CPU core and on-chip peripherals operate using the low-frequency clock.

Switching back and forth between NORMAL2 and SLOW modes is performed by the system control register 2.

③ IDLE2 mode

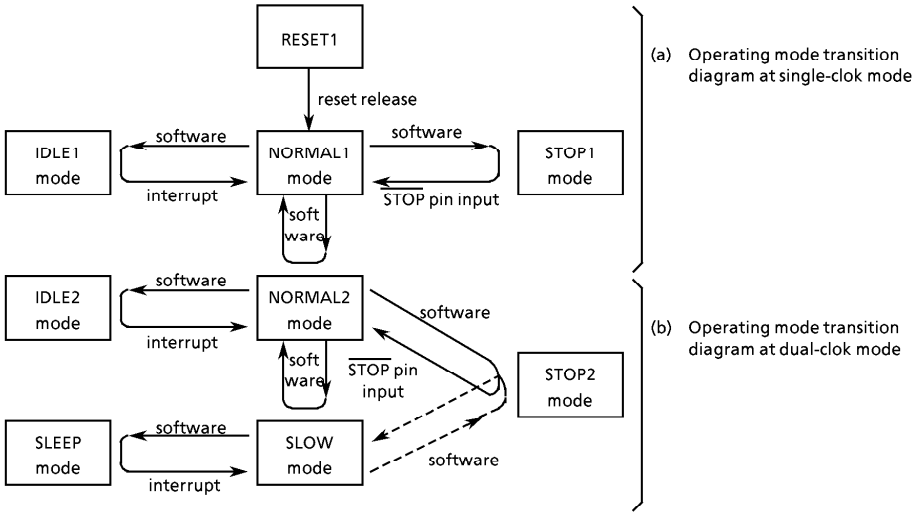
In this mode, the internal oscillation circuits remain active. The CPU and the watchdog timer are halted; however, on-chip peripherals remain active (operate using the high-frequency clock and/or the low-frequency clock). Starting and releasing of IDLE2 mode are the same as for IDLE1 mode, except that operation returns to NORMAL2 mode.

④ SLEEP mode

In this mode, the internal oscillation circuit of the low-frequency clock remains active. The CPU, the watchdog timer, and the internal oscillation circuit of the high- frequency clock are halted; however, on-chip peripherals remain active (operate using the low-frequency clock). Starting and releasing of SLEEP mode is the same as for IDLE1 mode, except that operation returns to SLOW mode.

⑤ STOP2 mode

As in STOP1 mode, all system operations are halted in this mode.



Note : NORMAL1 and NORMAL2 modes are generically called NORMAL; STOP1 and STOP2 are called STOP; and IDLE1, IDLE2 and SLEEP are called IDLE.

Operating mode		Frequency		CPU core	On-chip Peripherals	Machine cycle time
		High-frequency	Low-frequency			
Single-Clock	RESET1	turning on oscillation	turning off oscillation	reset	reset	4/fc [s]
	NORMAL1			operate	operate	
	IDLE1			halt		
	STOP1	turning off oscillation			halt	—
Dual-Clock	NORMAL2	turning on oscillation	turning on oscillation	High-frequency	operate (High and/or Low)	4/fc [s]
	IDLE2			halt		
	SLOW	turning off oscillation		Low-frequency	Low-frequency	
	SLEEP			halt		4/fs [s]
	STOP2		turning off oscillation		halt	—

Figure 1-14. Operating Mode Transition Diagram

System Control Register 1

SYSCR1 (0038_H)

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

(Initial value: 0000 00**)

STOP	STOP mode start	0 : CPU core and peripherals remain active 1 : CPU core and peripherals are halted (start STOP mode)	R/W
RELM	Release method for STOP mode	0 : Edge-sensitive release 1 : Level-sensitive release	
RETM	Operating mode after STOP mode	0 : Return to NORMAL mode 1 : Return to SLOW mode	
OUTEN	Port output control during STOP mode	0 : High-impedance 1 : Remain unchanged	
WUT	Warming-up time at releasing STOP mode	00 : 3×2^{19} / fc or 3×2^{13} / fs [s] 01 : 2^{19} / fc or 2^{13} / fs 1* : Reserved	

Note 1 : Always set RETM to "0" when transiting from NORMAL1 mode to STOP1 mode and from NOMAL2 mode to STOP2 mode. Always set RETM to "1" when transiting from SLOW mode to STOP2 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 clock* [Hz]
fs ; *low-frequency clock* [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 : When the STOP mode is started by specifying OUTEN = "0", the internal input of port is fixed to "0" and the interrupt of the falling edge may be set.

System Control Register 2

SYSCR2 (0039_H)

7	6	5	4	3	2	1	0
XEN	XTEN	SYSCK	IDLE				

(Initial value: 1000 ****)

XEN	High-frequency oscillator control	0 : Turn off oscillation 1 : Turn on oscillation	R/W
XTEN	Low-frequency oscillator control	0 : Turn off oscillation 1 : Turn on oscillation	
SYSCK	Main system clock select (write)/main system clock monitor (read)	0 : High-frequency clock 1 : Low-frequency clock	
IDLE	IDLE mode start	0 : CPU and watchdog timer remain active 1 : CPU and watchdog timer are stopped (start IDLE mode)	

Note 1 : A reset is applied ($\overline{\text{RESET}}$ pin output goes low) if both XEN and XTEN are cleared to "0".

Note 2 : Do not clear XEN to "0" when SYSCK = 0, and do not clear XTEN to "0" when SYSCK = 1.

*Note 3 : WDT; watchdog timer, * ; don't care*

Note 4 : Bits 3 - 0 in SYSCR2 are always read in as "1" when a read instruction is executed.

Note 5 : An optional initial value can't be selected for XTEN. In case of 87CH48/87PH48, initial value of XTEN is "0".

XTEN	operating mode after reset
0	Single-clock mode (NORMAL1)

Note 6 : The instruction for specifying Masking Option (Operating Mode) is ES Order Sheet is described in ADDITIONAL INFORMATION "Notice for Masking Option of TLCS-870 series" section 8.

Figure 1-15. System Control Registers

1.8.4 Operating Mode Control

(1) **STOP** mode (STOP1, STOP2)

STOP mode is controlled by the system control register 1 (SYSCR1) and the $\overline{\text{STOP}}$ pin input. The $\overline{\text{STOP}}$ pin is also used both as a port P20 and an $\overline{\text{INT5}}$ (external interrupt input 5) pin. STOP mode is started by setting STOP (bit 7 in SYSCR1) to "1". During STOP mode, the following status is maintained.

① Oscillations are turned off, and all internal operations are halted.

② The data memory (except for DBR), registers and port output latches are all held in the status in effect before STOP mode was entered. The port output can be select either output hold or high-impedance by setting OUTEN (bit 4 in SYSCR1).

③ The divider of the timing generator is cleared to "0".

④ The program counter holds the address of the instruction following the instruction which started the STOP mode.

STOP mode includes a level-sensitive release mode and an edge-sensitive release mode, either of which can be selected with RELM (bit 6 in SYSCR1).

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

In this mode, STOP mode is released by setting the $\overline{\text{STOP}}$ pin high. This mode is used for capacitor back-up when the main power supply is cut off and long term battery back-up.

When the $\overline{\text{STOP}}$ pin input is high, executing an instruction which starts the STOP mode will not place in STOP mode but instead will immediately start the release sequence (warm-up). Thus, to start STOP mode in the level-sensitive release mode, it is necessary for the program to first confirm that the $\overline{\text{STOP}}$ pin input is low. The following method can be used for confirmation:

- Using an external interrupt input $\overline{\text{INT5}}$ ($\overline{\text{INT5}}$ is a falling edge-sensitive input).

Example : Starting STOP mode with an INT5 interrupt.

```
PINT5 :   TEST   (P2) . 0                  ;  To reject noise, the STOP mode does not start if
          JRS    F, SINT5                      port P20 is at high
          LD     (SYSCR1), 01000000B       ;  Sets up the level-sensitive release mode.
          SET    (SYSCR1) . 7              ;  Starts STOP mode
          LDW    (IL) , 1110011101010111B  ;  IL12, 11, 7, 5, 3 ← 0 (Clears interrupt latches)
SINT5 :   RETI
```

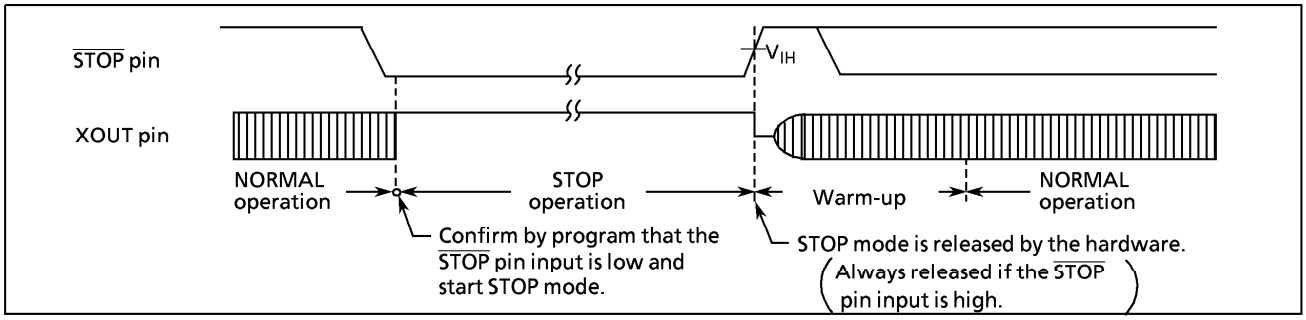


Figure 1-16. Level-sensitive Release Mode

Note 1 : When 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)

In this mode, STOP mode is released by a rising edge of the $\overline{STOP}$ pin input. This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is input to the $\overline{STOP}$ pin.

In the edge-sensitive release mode, STOP mode is started even when the $\overline{STOP}$ pin input is high.

Example : Starting STOP mode operation in the edge-sensitive release mode

```
LD      (SYSCR1), 10000000B   ;  OUTEN ← 0 (specifies high-impedance)
DI                            ;  IMF ← 0 (disables interrupt service)
SET     (SYSCR1).STOP         ;  STOP ← 1 (activates stop mode)
LDW     (IL),1110011101010111B ;  IL12, 11, 7, 5, 3 ← 0
                                  (clears interrupt latches)
EI                            ;  IMF ← 1 (enables interrupt service)
```

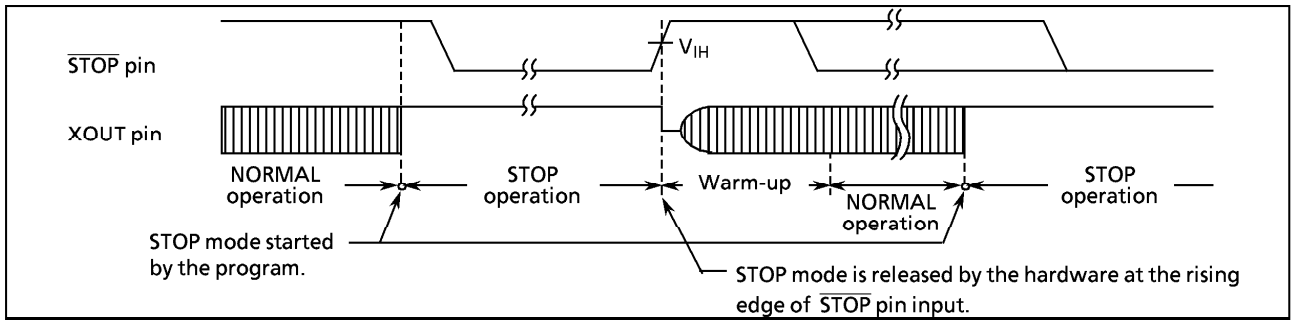


Figure 1-17. Edge-sensitive Release Mode

STOP mode is released by the following sequence:

① When returning to NORMAL2, both the high-frequency and low-frequency clock oscillators are turned on ; when returning to SLOW mode, only the low-frequency clock oscillator is turned on. When returning to NORMAL1, only the high-frequency clock oscillator is turned on.

② A warming-up period is inserted to allow oscillation time to stabilize. During warm-up, all internal operations remain halted. Two different warming-up times can be selected with WUT (bits 2 and 3 in SYSCR1) as determined by the resonator characteristics.

③ When the warming-up time has elapsed, normal operation resumes with the instruction following the STOP mode start instruction (e.g. [SET (SYSCR1). 7]). The start is made after the divider of the timing generator is cleared to "0".

Table 1-1. Warming-up Time example

Return to NORMAL1 mode			Return to SLOW mode	
WUT	At fc = 4.194304 MHz	At fc = 8 MHz	WUT	At fs = 32.768 kHz
3×2^{19} / fc [s]	375 [ms]	196.6 [ms]	3×2^{13} / fs [s]	750 [ms]
2^{19} / fc	125	65.5	2^{13} / fs	250

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.

STOP mode can also be released by setting the $\overline{RESET}$ pin low, which immediately performs the normal reset operation.

In this case, even if the setting is to return to the SLOW mode, it starts from the NORMAL mode. (If the initial XTEN of 87CH48 is set to "1" by mask option, they start from the NORMAL2 mode. In case of 87PH48, starts from NORMAL1 mode.)

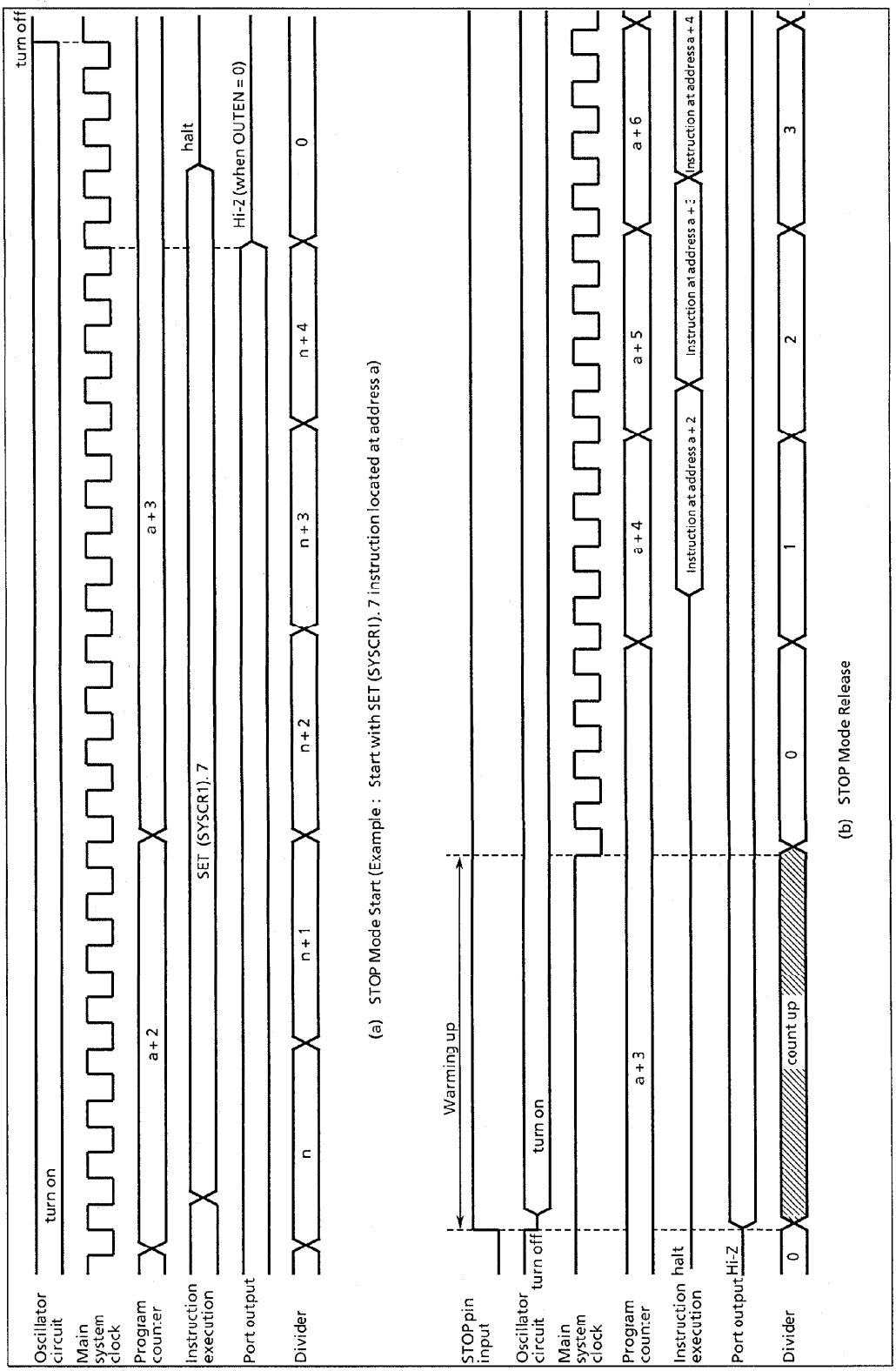


Figure 1-18. STOP Mode Start / Release

Note : When STOP mode is released with a low hold voltage, the following cautions must be observed.
The power supply voltage must be at the operating voltage level before releasing STOP mode. The $\overline{\text{RESET}}$ pin input must also be high, rising together with the power supply voltage. In this case, if an external time constant circuit has been connected, the $\overline{\text{RESET}}$ pin input voltage will increase at a slower rate than the power supply voltage. At this time, there is a danger that a reset may occur if input voltage level of the $\overline{\text{RESET}}$ pin drops below the non-inverting high-level input voltage (hysteresis input).

(2) **IDLE** mode (IDLE1, IDLE2, SLEEP)

IDLE mode is controlled by the system control register 2 and maskable interrupts. The following status is maintained during IDLE mode.

① Operation of the CPU and watchdog timer is halted. On-chip peripherals continue to operate.
② The data memory, CPU registers and port output latches are all held in the status in effect before IDLE mode was entered.
③ The program counter holds the address of the instruction following the instruction which started IDLE mode.

Example : Starting IDLE mode.

```
SET        (SYSCR2) . 4        ;  IDLE←1
```

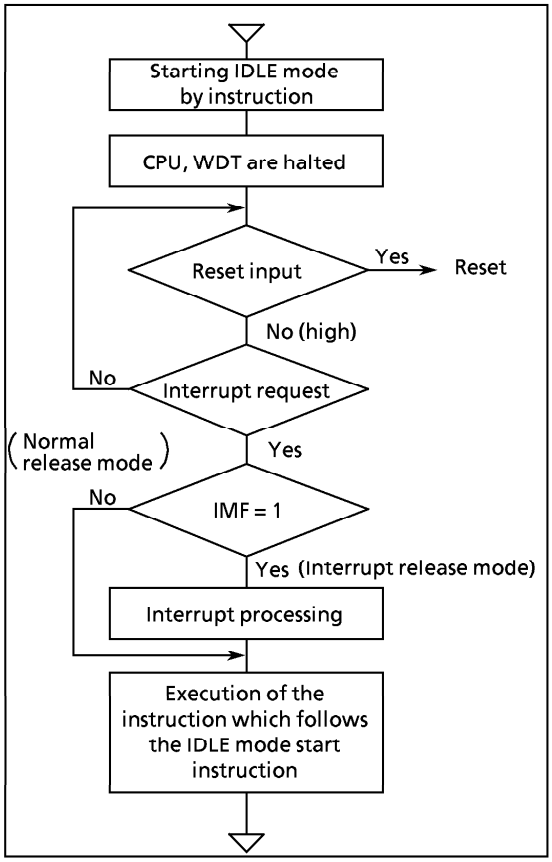


Figure 1-19. IDLE Mode

IDLE mode includes a normal release mode and an interrupt release mode. Selection is made with the interrupt master enable flag (IMF). Releasing the IDLE mode returns from IDLE1 to NORMAL1, from IDLE2 to NORMAL2, and from SLEEP to SLOW mode.

a. Normal release mode (IMF = "0")
IDLE mode is released by any interrupt source enabled by the individual interrupt enable flag (EF) or an external interrupt 0 ($\overline{\text{INT0}}$ pin) request. Execution resumes with the instruction following the IDLE mode start instruction (e.g. [SET (SYSCR2).4]).
The interrupt latch (IL) of the interrupt source for releasing the IDLE mode must be cleared to "0" by load instruction.

b. Interrupt release mode (IMF = "1")
IDLE mode is released and interrupt processing is started by any interrupt source enabled with the individual interrupt enable flag (EF) or an external interrupt 0 ($\overline{\text{INT0}}$ pin) request. After the interrupt is processed, the execution resumes from the instruction following the instruction which started IDLE mode.

IDLE mode can also be released by setting the $\overline{\text{RESET}}$ pin low, which immediately performs the reset operation. After reset, the 87CH48/PH48 are placed in NORMAL mode.
The 87PM48 is placed in NORMAL1 mode after reset release.

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

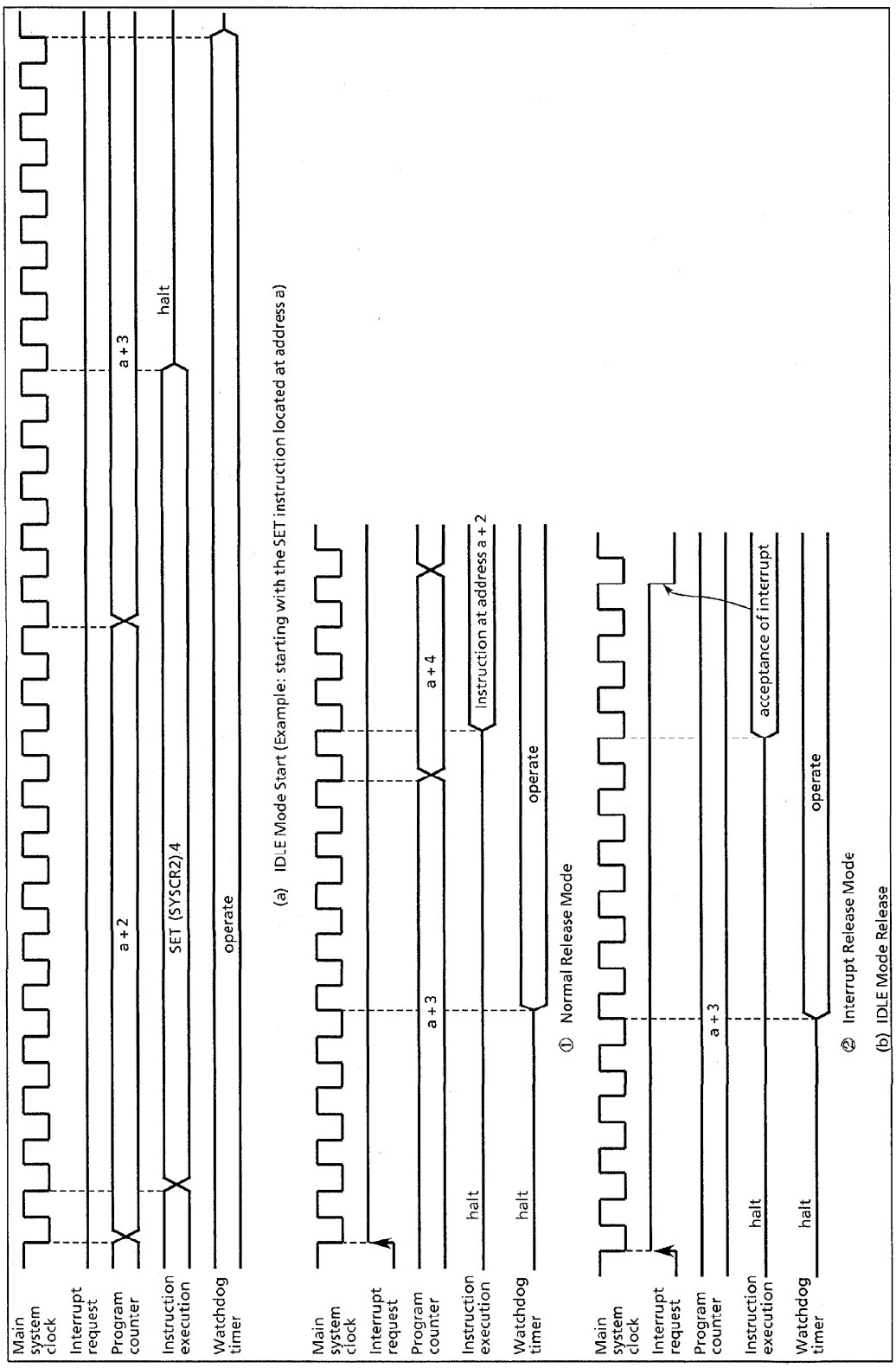


Figure 1-20. IDLE Mode Start/Release

(3) **SLOW** mode

SLOW mode is controlled by the system control register 2 and the timer/counter 2.

a. Switching from NORMAL2 mode to SLOW mode

First, set SYSCK (bit 5 in SYSCR2) to switch the main system clock to the low-frequency clock. Next, clear XEN (bit 7 in SYSCR2) to turn off high-frequency oscillation.

When the low-frequency clock oscillation is unstable, wait until oscillation stabilizes before performing the above operations. The timer/counter 2 (TC2) can conveniently be used to confirm that low-frequency clock oscillation has stabilized.

Note : The high frequency clock can be continued oscillation in order to return to NORMAL2 mode from SLOW mode quickly. Always turn off oscillation of high frequency clock when switching from SLOW mode to STOP mode.

Example1 : Switching from NORMAL2 mode to SLOW mode.

```
        SET     (SYSCR2) . 5        ; SYSCK←1   (Switches the main system clock to the
                                                low-frequency clock)
        CLR     (SYSCR2) . 7        ; XEN←0     (turns off high-frequency oscillation)
```

Example2 : Switching to SLOW mode after low-frequency clock oscillation has stabilized.

```
        LD      (TC2CR), 14H        ; Sets TC2 mode
                                      (timer mode, source clock : fs)
        LDW     (TREG2), 8000H      ; Sets warming-up time
                                      (according to Xtal characteristics)
        SET     (EIRH). EF14        ; Enable INTTC2
        LD      (TC2CR), 34H        ; Starts TC2
         ⋮
PINTTC2 : LD    (TC2CR), 10H        ; Stops TC2
        SET     (SYSCR2) . 5        ; SYSCK←1
        CLR     (SYSCR2) . 7        ; XEN←0
        RETI
         ⋮
VINTTC2 : DW    PINTTC2             ; INTTC2 vector table
```

b. Switching from SLOW mode to NORMAL2 mode

First, set XEN (bit 7 in SYSCR2) to turn on the high-frequency oscillation. When time for stabilization (warm-up) has been taken by the timer/counter 2 (TC2), clear SYSCK (bit 5 in SYSCR2) to switch the main system clock to the high-frequency clock.

SLOW mode can also be released by setting the $\overline{\text{RESET}}$ pin low, which immediately performs the reset operation. After reset, the 87CH48/PH48 are placed in NORMAL mode.

Example : Switching from SLOW mode to NORMAL2 mode (fc = 8 MHz, warming-up time is about 7.9 ms).

```
                SET     (SYSCR2) . 7        ; XEN←1     (turns on high-frequency oscillation)
                LD      (TC2CR), 10H        ; Sets TC2 mode
                                            (timer mode, source clock: fc)
                LD      (TREG2 + 1), 0F8H   ; Sets the warming-up time
                                            (according to frequency and resonator characteristics)
                SET     (EIRH). EF14        ; Enable INTTC2
                LD      (TC2CR), 30H        ; Starts TC2
                 ⋮
PINTTC2 :       LD      (TC2CR), 10H        ; Stops TC2
                CLR     (SYSCR2) . 5        ; SYSCK←0   (Switches the main system clock to the
                                                         high-frequency clcok)
                RETI
                 ⋮
VINTTC2 :       DW      PINTTC2             ; INTTC2 vector table
```

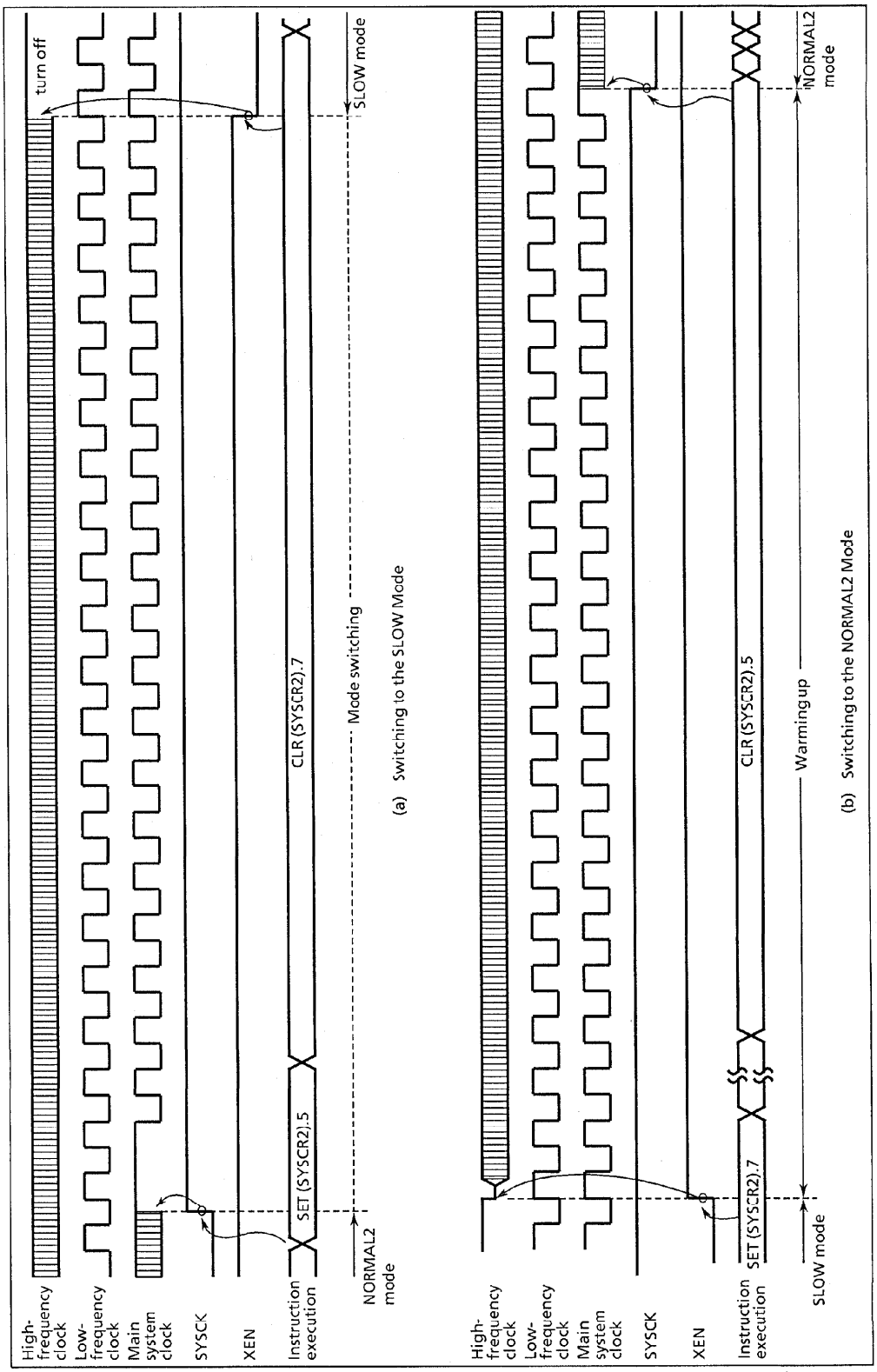


Figure 1-21. Switching between the NORMAL2 and SLOW Modes

1.9 Interrupt Controller

The 87CH48 each have a total of 15 interrupt sources: 6 externals and 9 internals. Nested interrupt control with priorities is also possible. Two of the internal sources are pseudo non-maskable interrupts; the remainder are all maskable interrupts.

Interrupt latches (IL) that hold the interrupt requests are provided for interrupt sources. Each interrupt vector is independent.

The interrupt latch is set to "1" when an interrupt request is generated and requests the CPU to accept the interrupt. The acceptance of maskable interrupts can be selectively enabled and disabled by the program using the interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). When two or more interrupts are generated simultaneously, the interrupt is accepted in the highest priority order as determined by the hardware. Figure 1-22 shows the interrupt controller.

Interrupt Source			Enable Condition	Interrupt Latch	Vector Table Address	Priority
Internal/ External	(Reset)		Non-Maskable	—	$FFFE_H$	High 0
Internal	INTSW	(Software interrupt)	Pseudo non-maskable	—	$FFFC_H$	1
Internal	INTWDT	(Watchdog Timer interrupt)		IL_2	$FFFA_H$	2
External	INT0	(External interrupt 0)	IMF = 1, INT0EN = 1	IL_3	$FFF8_H$	3
Internal	INTTC1	(16-bit TC1 interrupt)	IMF · EF_4 = 1	IL_4	$FFF6_H$	4
External	INT1	(External interrupt 2)	IMF · EF_5 = 1	IL_5	$FFF4_H$	5
Internal	INTTBT	(Time Base Timer interrupt)	IMF · EF_6 = 1	IL_6	$FFF2_H$	6
External	INT2	(External interrupt 2)	IMF · EF_7 = 1	IL_7	$FFF0_H$	7
Internal	INTTC3	(8-bit TC3 interrupt)	IMF · EF_8 = 1	IL_8	$FFEE_H$	8
Internal	INTSBI	(Serial BUS Interface interrupt)	IMF · EF_9 = 1	IL_9	$FFEC_H$	9
External	INT3	(External interrupt 3)	IMF · EF_{10} = 1	IL_{10}	$FFEA_H$	10
External	INT4	(External interrupt 4)	IMF · EF_{11} = 1	IL_{11}	$FFE8_H$	11
Internal	INTRX	(UART receive interrupt)	IMF · EF_{12} = 1, INTS = 0	IL_{12}	$FFE6_H$	12
	INTTC4	(8-bit TC4 interrupt)	IMF · EF_{12} = 1, INTS = 1			
Internal	INTTX	(UART transmit interrupt)	IMF · EF_{13} = 1	IL_{13}	$FFE4_H$	13
Internal	INTTC2	(16-bit TC2 interrupt)	IMF · EF_{14} = 1	IL_{14}	$FFE2_H$	14
External	INT5	(External interrupt 5)	IMF · EF_{15} = 1	IL_{15}	$FFE0_H$	Low 15

(1) Interrupt Latches ($IL_{15\ to\ 2}$)

Interrupt latches are provided for each source, except for a software interrupt. The latch is set to "1" when an interrupt request is generated, and requests the CPU to accept the interrupt. The latch is cleared to "0" just after the interrupt is accepted. All interrupt latches are initialized to "0" during reset.

The interrupt latches are assigned to addresses $003C_H$ and $003D_H$ in the SFR. Each latch can be cleared to "0" individually by an instruction; however, the read-modify-write instruction such as bit manipulation or operation instructions cannot be used (Do not clear the IL_2 for a watchdog timer interrupt to "0"). Thus, interrupt requests can be cancelled and initialized by the program. Note that interrupt latches cannot be set to "1" by any instruction.

The contents of interrupt latches can be read out by an instruction. Therefore, testing interrupt requests by software is possible.

```
Example 1 : Clears interrupt latches
            LDW     (IL), 1110100000111111B     ; IL12, IL10 to IL6←0
Example 2 : Reads interrupt latches
            LD      WA, (IL)                    ; W←ILH, A←ILL
Example 3: Tests an interrupt latch
            TEST    (IL).7                      ; if IL7 = 1 then jump
            JR      F, SSET
```

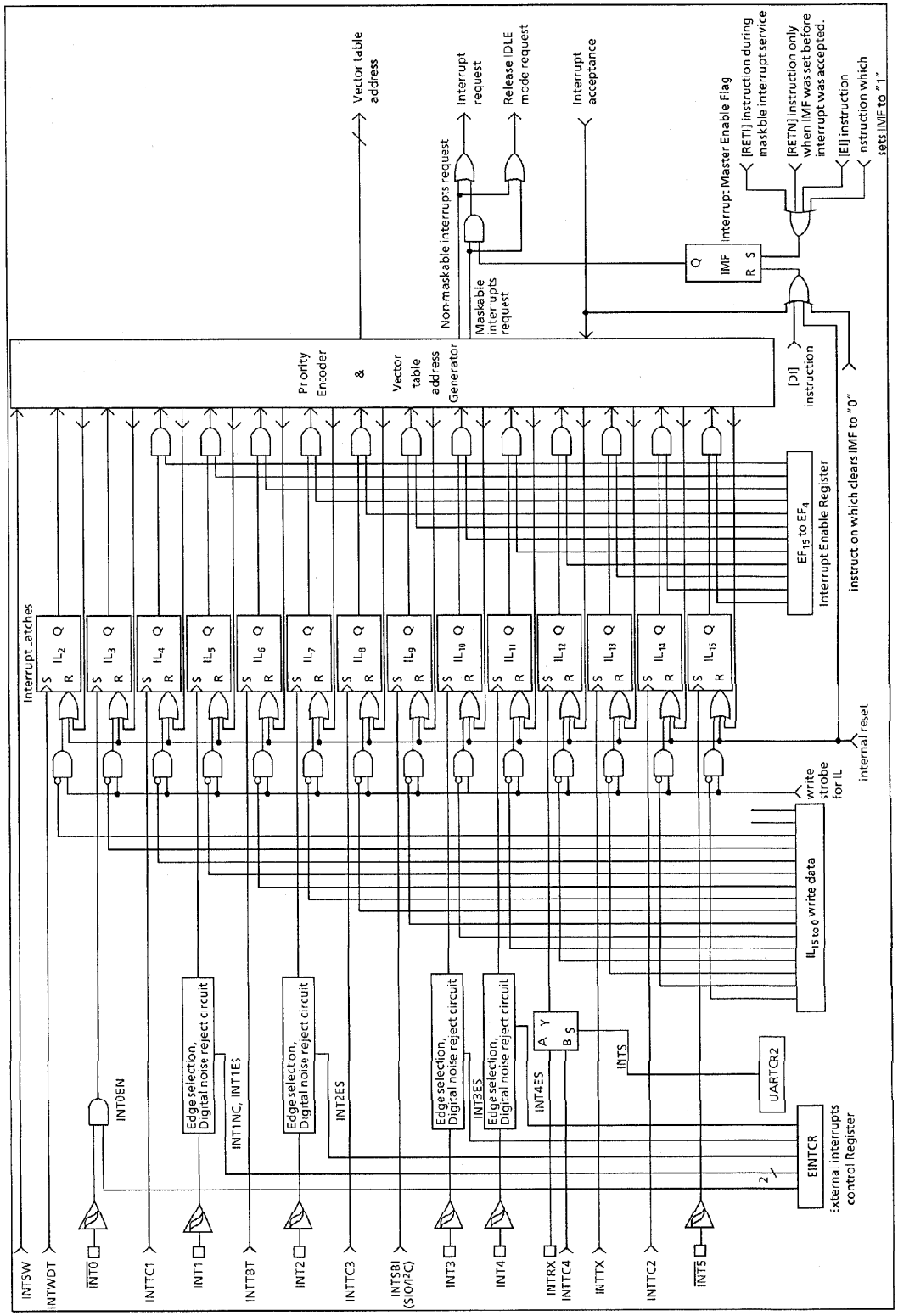


Figure 1-22. Interrupt Controller Block Diagram

(2) **Interrupt Enable Register** (EIR)

The interrupt enable registers (EIR) enable and disable the acceptance of interrupts except for the pseudo non-maskable interrupts (software and watchdog timer interrupts). Pseudo non-maskable interrupts are accepted regardless of the contents of the EIR; however, the pseudo non-maskable interrupts cannot be nested more than once at the same time. For example, the watchdog timer interrupt is not accepted during the software interrupt service.

The EIR consists of an interrupt master enable flag (IMF) and individual interrupt enable flags (EF). These registers are assigned to addresses $003A_H$ and $003B_H$ in the SFR, and can be read and written by an instruction (including read-modify-write instructions such as bit manipulation instructions).

① **Interrupt Master enable Flag** (IMF)

The interrupt master enable flag (IMF) enables and disables the acceptance of all interrupts, except for pseudo non-maskable interrupts. Clearing this flag to "0" disables the acceptance of all maskable interrupts. Setting to "1" enables the acceptance of interrupts.

When an interrupt is accepted, this flag is cleared to "0" to temporarily disable the acceptance of maskable interrupts. After execution of the interrupt service program, this flag is set to "1" by the maskable interrupt return instruction [RETI] to again enable the acceptance of interrupts. If an interrupt request has already been occurred, interrupt service starts immediately after execution of the [RETI] instruction.

Pseudo non-maskable interrupts are returned by the [RETN] instruction. In this case, the IMF is set to "1" only when pseudo non-maskable interrupt service is started with interrupt acceptance enabled (IMF = 1). Note that IMF remains "0" when cleared in the interrupt service program.

The IMF is assigned to bit 0 at address $003A_H$ in the SFR, and can be read and written by an instruction. IMF is normally set and cleared by the [EI] and [DI] instructions, and the IMF is initialized to "0" during reset.

Note : Do not set IMF to "1"during non-maskable interrupt service programs.

② **Individual interrupt Enable Flags** (EF_{15} to EF_4)

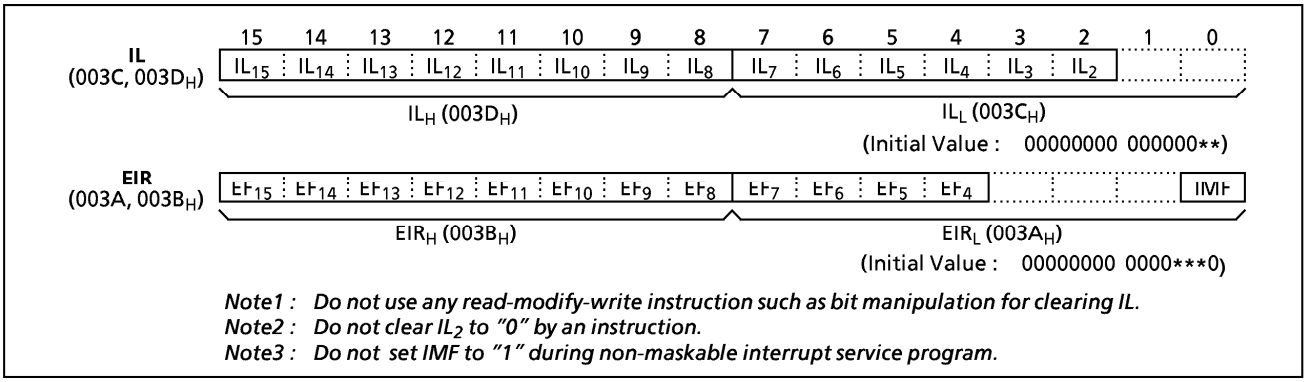
These flags enable and disable the acceptance of individual maskable interrupts, except for an external interrupt 0. Setting the corresponding bit of an individual interrupt enable flag to "1" enables acceptance of an interrupt, setting the bit to "0" disables acceptance.

Example 1 : Sets EF for individual interrupt enable, and sets IMF to "1".

```
LDW     (EIR), 1110100010100001B    ;  EF15~EF13, EF11, EF7, EF5, IMF←1
```

Example 2 : Sets an individual interrupt enable flag to "1".

```
SET     (EIRH).4                    ;  EF12←1
```



Note1 : Do not use any read-modify-write instruction such as bit manipulation for clearing IL.
Note2 : Do not clear IL_2 to "0" by an instruction.
Note3 : Do not set IMF to "1" during non-maskable interrupt service program.

Figure 1-23. Interrupt Latch (IL) and Interrupt Enable Register (EIR)

1.9.1 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to "0" by a reset or an instruction. Interrupt acceptance sequence requires 8 machine cycles (4 μs at fc = 8 MHz in NORMAL mode) after the completion of the current instruction execution. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for pseudo non-maskable interrupts).

(1) Interrupt acceptance processing

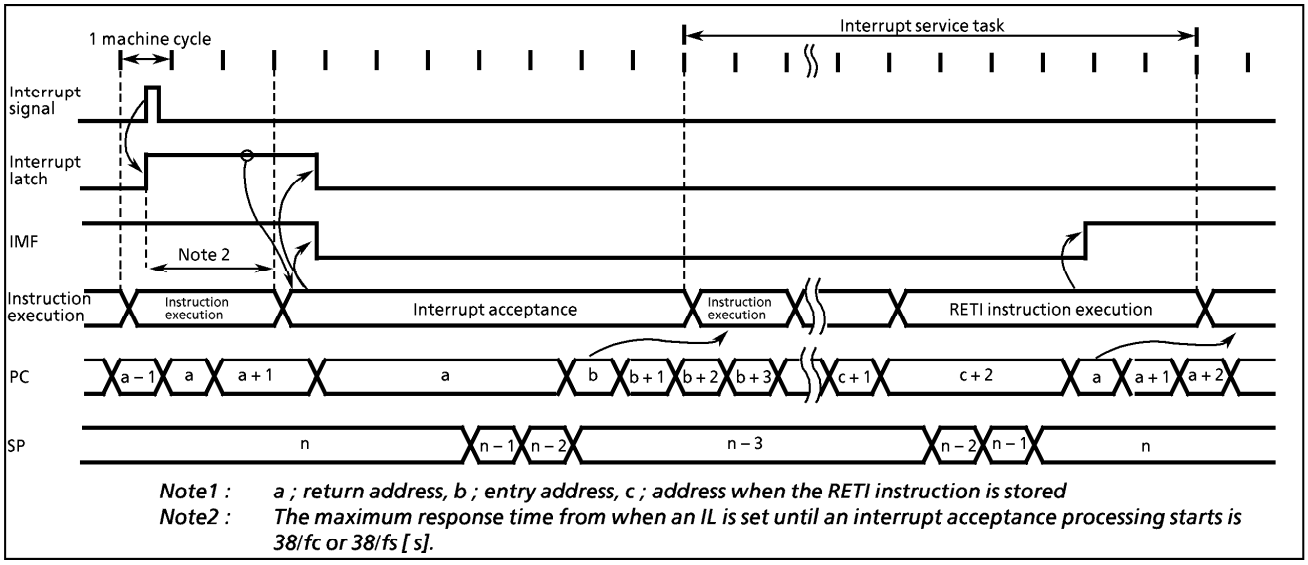
① The interrupt master enable flag (IMF) is cleared to "0" to temporarily disable the acceptance of any following maskable interrupts. When a non-maskable interrupt is accepted, the acceptance of any following interrupts is temporarily disabled.

② The interrupt latch (IL) for the interrupt source accepted is cleared to "0".

③ The contents of the program counter (return address) and the program status word are saved (pushed) onto the stack. The contents of Stack Pointer is decreased by 3.

④ The entry address of the interrupt service program is read from the vector table address, and the entry address is loaded to the program counter.

⑤ The instruction stored at the entry address of the interrupt service program is executed.

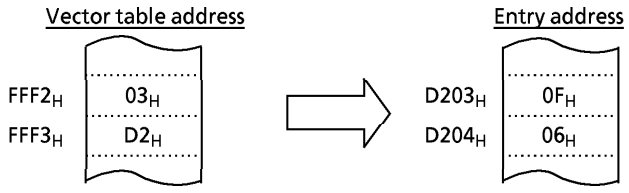


Note1 : a ; return address, b ; entry address, c ; address when the RETI instruction is stored

Note2 : The maximum response time from when an IL is set until an interrupt acceptance processing starts is 38/fc or 38/fs [s].

Figure 1-24. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

Example : Correspondence between vector table address for INTTBT and the entry address of the interrupt service program.



A maskable interrupt is not accepted until the IMF is set to "1" even if a maskable interrupt of higher priority than that of the current interrupt being serviced.

When nested interrupt service is necessary, the IMF is set to "1" in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags. However, an acceptance of external interrupt 0 cannot be disabled by the EF; therefore, if disablement is necessary, either the external interrupt function of the $\overline{INT0}$ pin must be disabled with INT0EN in the external interrupt control register (EINTCR) or interrupt processing must be avoided by the program.

When INT0EN = 0, the interrupt latch IL3 is not set, therefore, the falling edge of the $\overline{INT0}$.

Example 1 : Disables an external interrupt 0 using INT0EN:

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

Example 2 : Disables the processing of external interrupt 0 under the software control (using bit 0 at address $00F0_H$ as the interrupt processing disable switch):

```
PINT0 :   TEST    (00F0H) . 0     ;  Returns without interrupt processing if (00F0H)0 = 1
          JRS     T, SINT0
          RETI
SINT0 :   Interrupt processing
          RETI
            :
VINT0 :   DW      PINT0
```

(2) General-Purpose register save / restore processing

During interrupt acceptance processing, the program counter and the program status word are automatically saved on the stack, but not the accumulator and other registers. These registers are saved by the program if necessary. Also, when nesting multiple interrupt services, it is necessary to avoid using the same data memory area for saving registers.
The following method is used to save/restore the general-purpose registers:

① General-purpose register save/restore by register bank changeove:
General-purpose registers can be saved at high-speed by switching to a register bank that is not in use. Normally, bank 0 is used for the main task and banks 1 to 15 are assigned to interrupt service tasks. To increase the efficiency of data memory utilization, the same bank is assigned for interrupt sources which are not nested.
The switched bank is automatically restored by executing an interrupt return instruction [RETI] or [RETN]. Therefore, it is not necessary for a program to save the RBS.

Example : Register Bank Changeover

```
PINTxx :  LD      RBS, n          ;  Switches to bank n (1μs at 8MHz)
          interrupt processing
          RETI                    ;  Restores bank and Returns
```

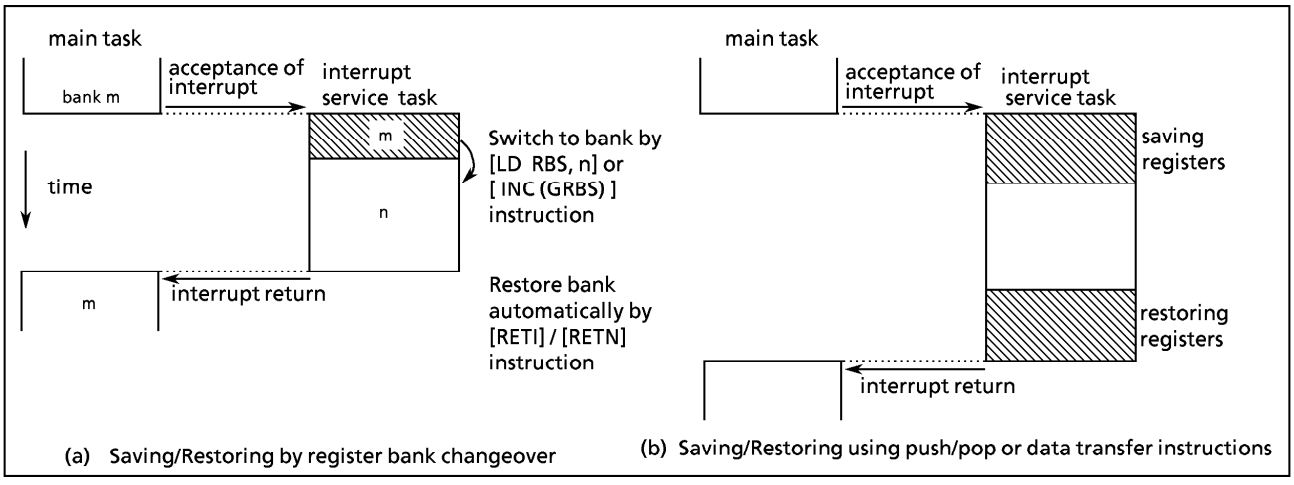
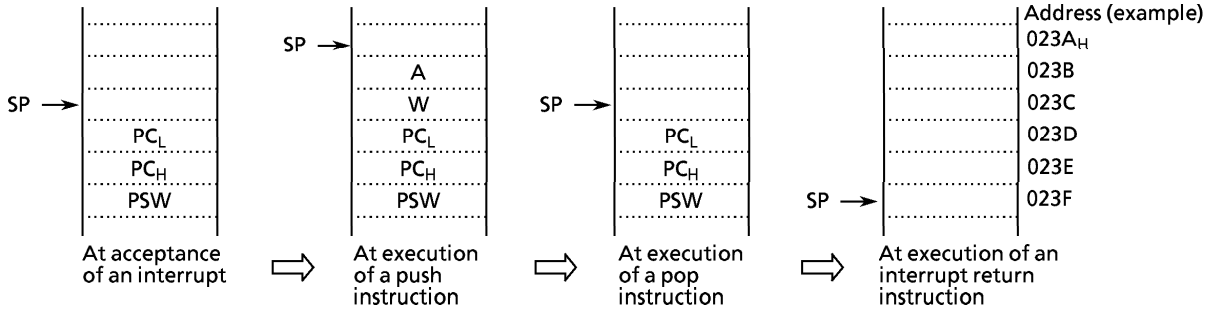


Figure 1-25. Saving/Restoring General-purpose Registers

② General-purpose register save/restore using push and pop instructions:
To save only a specific register, and when the same interrupt source occurs more than once, the general-purpose registers can be saved/restored using push/pop instructions.

Example : Register save using push and pop instructions

```
PINTxx :    PUSH    WA          ; Save WA register pair
            interrupt processing
            POP     WA          ; Restore WA register pair
            RETI                ; Return
```



③ General-purpose registers save/restore using data transfer instruction:
Data transfer instructions can be used to save only a specific general-purpose register during processing of a single interrupt.

Example : Saving/restoring a register using data transfer instructions

```
PINTxx :    LD      (GSAVA), A  ; Save A register
            interrupt processing
            LD      A, (GSAVA)  ; Restore A register
            RETI                ; 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.	① The contents of the program counter and program status word are restored from the stack.
② The stack pointer is incremented 3 times.	② The stack pointer is incremented 3 times.
③ The interrupt master enable flag is set to "1".	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.

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.

1.9.2 External Interrupts

The 87CH48 each have six external interrupt inputs ($\overline{INT0}$, INT1, INT2, INT3, INT4, and $\overline{INT5}$). Four of these are equipped with digital noise rejection circuits (pulse inputs of less than a certain time are eliminated as noise). Edge selection is also possible with INT1, INT2, INT3 and INT4.

The $\overline{INT0}$/P10 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

Edge selection, noise rejection control and $\overline{INT0}$/P10 pin function selection are performed by the external interrupt control register (EINTCR). When INT0EN = 0, the IL_3 will not be set even if the falling edge of $\overline{INT0}$ pin input is detected.

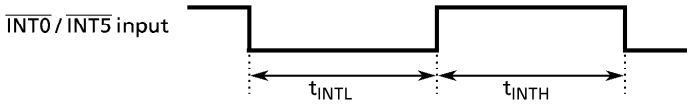
Table 1-3. External Interrupts

Source	Pin	Secondary function pin	Enable conditions	Edge	Digital noise reject
INT0	$\overline{INT0}$	P10	IMF = 1, INT0EN = 1	falling edge	— (hysteresis input)
INT1	INT1	P11	IMF · EF_5 = 1	falling edge or rising edge	Pulses less than 15/fc [s] or 63/fc [s] are cancelled as noise. Pulses equal to or more than 48/fc [s] or 192/fc [s] are regarded as signals.
INT2	INT2	P12/TC1	IMF · EF_7 = 1		Pulses less than 7/fc [s] are cancelled as noise. Pulses equal to or more than 24/fc [s] are regarded as signals. Same applies to pins TC1, TC3 and TC4.
INT3	INT3	P50/TC3	IMF · EF_{10} = 1		
INT4	INT4	P51/TC4	IMF · EF_{11} = 1		
INT5	$\overline{INT5}$	P20/$\overline{STOP}$	IMF · EF_{15} = 1	falling edge	— (hysteresis input)

Note 1 : The noise rejection function is turned off in the SLOW and SLEEP modes. Also, the noise reject times are not constant for pulses input while transiting between operating modes (NORMAL2↔SLOW)

Note 2 : The noise rejection function is also affected for timer/counter input (TC1 and TC3 pins).

Note 3 : The pulse width (both "H" and "L" level) for input to the $\overline{INT0}$ and $\overline{INT5}$ pins must be over 1 machine cycle.



t_{INTL}, t_{INTH} > tcyc

Note : tcyc = 4/fc [s] (at Normal 1/2 and IDLE 1/2 modes)
4/fs [s] (at SLOW and SLEEP modes)

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,INT3, INT4 pins 25/fc [s]

Note 5 : When high-impedance is specified for port output in stop mode, port input is forcibly fixed to low level internally. Thus, interrupt latches of external interrupt inputs except $\overline{INT5}$ (P20/$\overline{STOP}$) which are also used as ports may be set to "1". To specify high-impedance for port output in stop mode, first disable interrupt service (IMF = 0) , activate stop mode. After releasing stop mode, clear interrupt latches using load instruction, then, enable interrupt service.

Example : Activating stop mode.

```
LD   (SYSCR1),01000000B      ; OUTEN ← 0 (specifies high impedance)
DI                           ; IMF ← 0 (disables interrupt service)
SET (SYSCR1).STOP            ; STOP ← 1 (activates stop mode)
LDW (IL),1110011101010111B   ; IL12,11,7,5,3 ← 0 (clears interrupt latches)
EI                           ; IMF ← 1 (enables interrupt service)
```

EINTCR (0037_H)

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

(Initial value : 00*0 000*)

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/$\overline{INT0}$ pin configuration	0 : P10 input/output port 1 : $\overline{INT0}$ 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 1 : *fc ; High-frequency clock [Hz] * ; don't care*

Note 2 : *Edge detection during switching edge selection is invalid.*

Note 3 : *Do not change EINTCR when IMF = 1. After changing EINTCR, interrupt latches of external interrupt inputs must be cleared to "0" using load instruction.*

Note 4 : *In order to change of external interrupt input by rewriting the contents of INT2ES, INT3ES and INT4ES during NORMAL1/2 mode, clear interrupt latches of external interrupt inputs (INT2, INT3 and INT4) after 8 machine cycles from the time of rewriting. During SLOW mode, 3 machine cycles are required.*

Note 5 : *In order to change an edge of timer counter input by rewriting the contents of INT2ES, INT3ES and INT4ES during NORMAL1/2 mode, rewrite the contents after timer counter is stopped (TC*s = 0), that is, interrupt disable state. Then, clear interrupt laches of external interrupt inputs (INT2, INT3 and INT4) after 8 machine cycles from the time of rewriting to change to interrupt enable state. Finally, start timer counter. During SLOW mode, 3 machine cycles are required.*

Example : When changing TC1 pin inputs edge in external trigger timer mode from rising edge to falling edge.

```
        LD (TC1CR),01001000B    ; TC1S ← 00 (stop TC1)
        DI                      ; IMF ← 0 (disable interrupt service)
        LD (EINTCR),00000100B   ; INT2ES ← 1 (change edge selection)
        NOP
        to
        NOP                     } 8 machine cycles
        LD (ILL),01111111B      ; IL7 ← 0 (clear interrupt latch)
        EI                      ; IMF ← 1 (enable interrupt service)
        LD (TC1CR),01111000B    ; TC1S ← 11 (start TC1)
```

Note 6 : *If changing the contents of INT1ES during NORMAL1/2 mode, interrupt latch of external interrupt input INT1 must be cleared after 14 machine cycles (when INT1NC = 1) or 50 machine cycles (when INT1NC = 0) from the time of changing. During SLOW mode, 3 machine cycles are required.*

Figure 1-26. External Interrupt Control Register

1.9.3 Software Interrupt (INTSW)

Executing the [SWI] instruction generates a software interrupt and immediately starts interrupt processing (INTSW is highest prioritized interrupt). However, if processing of a non-maskable interrupt is already underway, executing the SWI instruction will not generate a software interrupt but will result in the same operation as the [NOP] instruction. Thus, the [SWI] instruction behaves like the [NOP] instruction.

Note : Software interrupt generates during non-maskable interrupt processing to use SWI instruction for software break in a development tool.

Use the [SWI] instruction only for detection of the address error or for debugging.

① Address Error Detection

FF_H is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address. Code FF_H is the SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing FF_H to unused areas of the program memory. Address trap reset is generated for instruction fetch from a specific address (0000 to $043F_H$).

Note : The fetch data from addresses $BF80_H$ to $BFFF_H$ (test ROM area) for 87CH48/PH48 is not "FF_H".

② Debugging

Debugging efficiency can be increased by placing the SWI instruction at the software break point setting address.

1.10 Watchdog Timer (WDT)

The watchdog timer rapidly detects the CPU malfunction such as endless looping caused by noise or the like, and resumes the CPU to the normal state.

The watchdog timer signal for detecting malfunction can be selected either a reset output or a non-maskable interrupt request. However, selection is possible only once after reset. At first the reset output is selected.

When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals.

1.10.1 Watchdog Timer Configuration

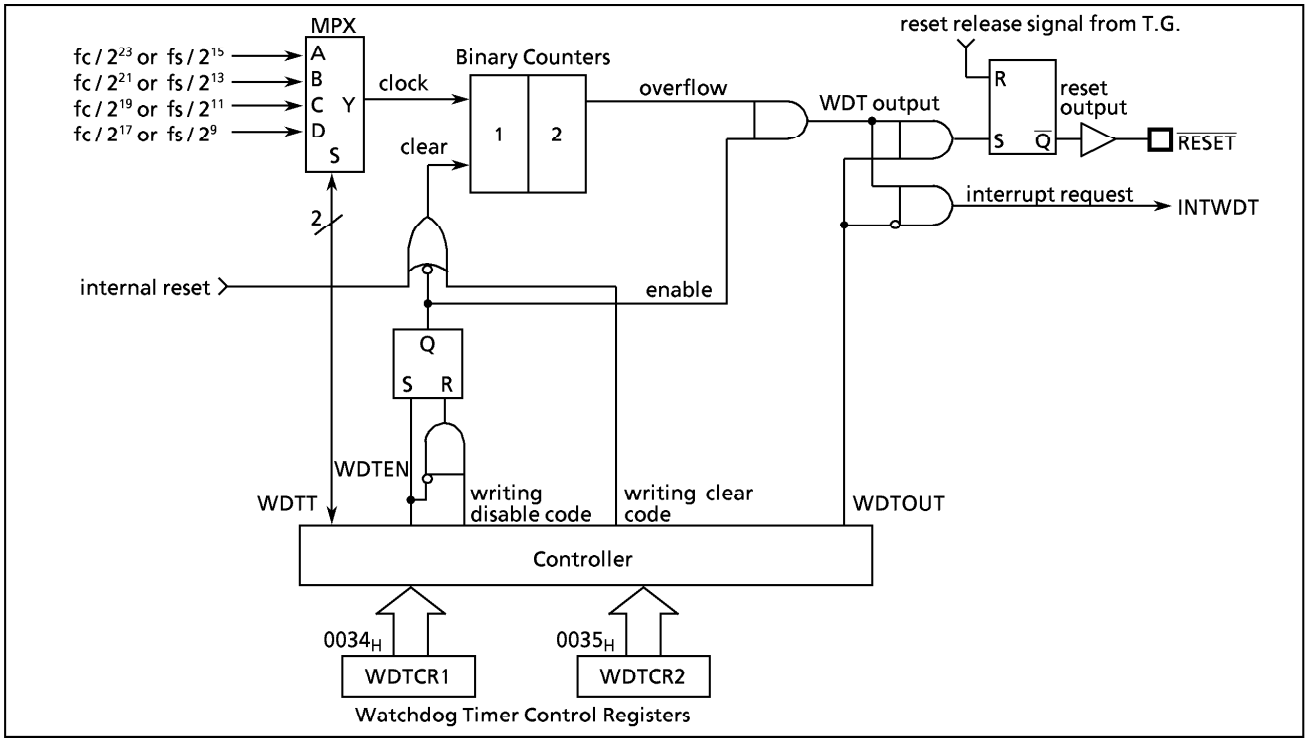


Figure 1-27. Watchdog Timer Configuration

1.10.2 Watchdog Timer Control

Figure 1-28 shows the watchdog timer control registers (WDTCR1, WDTCR2). The watchdog timer is automatically enabled after reset.

(1) Malfunction detection methods using the watchdog timer

The CPU malfunction is detected as follows.

① Setting the detection time, selecting output, and clearing the binary counter.

② Repeatedly clearing the binary counter within the setting detection time.

If the CPU malfunction occurs for any cause, the watchdog timer output will become active at the rising of an overflow from the binary counters unless the binary counters are cleared. At this time, when WDTOUT = 1 a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the internal hardware and the external circuits. When WDTOUT = 0, a watchdog timer interrupt (INTWDT) is generated.

The watchdog timer temporarily stops counting in the STOP mode including warm-up or IDLE mode, and automatically restarts (continues counting) when the STOP/IDLE mode is released.

Example : Sets the watchdog timer detection time to 2^{21}/fc [s] and resets the CPU malfunction.

```
        LD      (WDTCR1), 00001101B   ; WDTT←10, WDTOUT←1
        LD      (WDTCR2), 4EH         ; Clears the binary counters
Within WDT                              (always clear immediately after changing WDTT)
detection time
        LD      (WDTCR2), 4EH         ; Clears the binary counters
Within WDT
detection time
        LD      (WDTCR2), 4EH         ; Clears the binary counters
```

Watchdog Timer Control Register 1

WDTCR1 (0034_H)

7	6	5	4	3	2	1	0
				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	00 : 2^{25} / fc or 2^{17} / fs [s] 01 : 2^{23} / fc or 2^{15} / fs 10 : 2^{21} / fc or 2^{13} / fs 11 : 2^{19} / fc or 2^{11} / fs	
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] fs ; Low-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 : Disable the watchdog timer or clear the counter just before switching to STOP mode.
When the counter is cleared just before switching to STOP mode, clear the counter again subsequently to releasing STOP mode.

Watchdog Timer Control Register 2

WDTCR2 (0035_H)

7	6	5	4	3	2	1	0

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

WDTCR2	Watchdog timer control code write register	$4E_H$: 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 : Since WDTCR2 is a write-only register, read-modify-write instructions (e.g., bit manipulating instructions such as SET or CLR and arithmetic instructions such as AND or OR) cannot be used for read / write to this register.
Note 4 : To clear binary counter doesn't initialize the source clock, therefore, it is recommended to clear binary counter within 3/4 of the detection period.

Figure 1-28. Watchdog Timer Control Registers

Table 1-4. Watchdog Timer Detection Time

Operating mode			Detection time	
NORMAL1	NORMAL2	SLOW	At fc = 8 MHz	At fs = 32.768 kHz
2^{25} / fc [s]	2^{25} / fc, 2^{17} / fs	2^{17} / fs	4.194 s	4 s
2^{23} / fc	2^{23} / fc, 2^{15} / fs	2^{15} / fs	1.048 ms	1 s
2^{21} / fc	2^{21} / fc, 2^{13} / fs	——	262.1 ms	250 ms
2^{19} / fc	2^{19} / fc, 2^{11} / fs	——	65.5 ms	62.5 ms

(2) Watchdog Timer Enable

The watchdog timer is enabled by setting WDTEN (bit 3 in WDTCR1) to "1". WDTEN is initialized to "1" during reset, so the watchdog timer operates immediately after reset is released.

Example : Enables watchdog timer

```
LD      (WDTCR1), 00001000B     ; WDTEN←1
```

(3) Watchdog Timer Disable

The watchdog timer is disabled by writing the disable code ($B1_H$) to WDTCR2 after clearing WDTEN (bit 3 in WDTCR1) to "0". The watchdog timer is not disabled if this procedure is reversed and the disable code is written to WDTCR2 before WDTEN is cleared to "0". The watchdog timer is halted temporarily in STOP mode (including warm-up) and IDLE mode, and restarts automatically after STOP or IDLE mode is released.

During disabling the watchdog timer, the binary counters are cleared to "0".

Example : Disables watchdog timer

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

1.10.3 Watchdog Timer Interrupt (INTWDT)

This is a pseudo non-maskable interrupt which can be accepted regardless of the contents of the EIR. If a watchdog timer interrupt or a software interrupt is already accepted, however, the new watchdog timer interrupt waits until the previous interrupt processing is completed (the end of the [RETN] instruction execution).

The stack pointer (SP) should be initialized before using the watchdog timer output as an interrupt source with WDTOUT.

Example : Watchdog timer interrupt setting up.

```
LD      SP, 013FH               ; Sets the stack pointer
LD      (WDTCR1) , 00001000B    ; WDTOUT←0
```

1.10.4 Watchdog Timer Reset

If the watchdog timer output becomes active, a reset is generated, which drives the $\overline{\text{RESET}}$ pin (sink open drain output) low to reset the internal hardware and the external circuits. The reset output time is 12/fc [s] to 16/fc [s] (1.5 to 2.0 μs at 8MHz). The high-frequency clock oscillator also turns on when a watchdog timer reset is generated in SLOW mode.

Note : The high-frequency clock oscillator also turns on when a watchdog timer reset is generated in SLOW mode. Thus, the reset output time is 2^{20}/fc.
The reset output time include a certain amount of error if there is any function of the oscillation frequency when the high-frequency clock oscillator turns on. Thus, the reset, the reset output time must be considered approximate value.

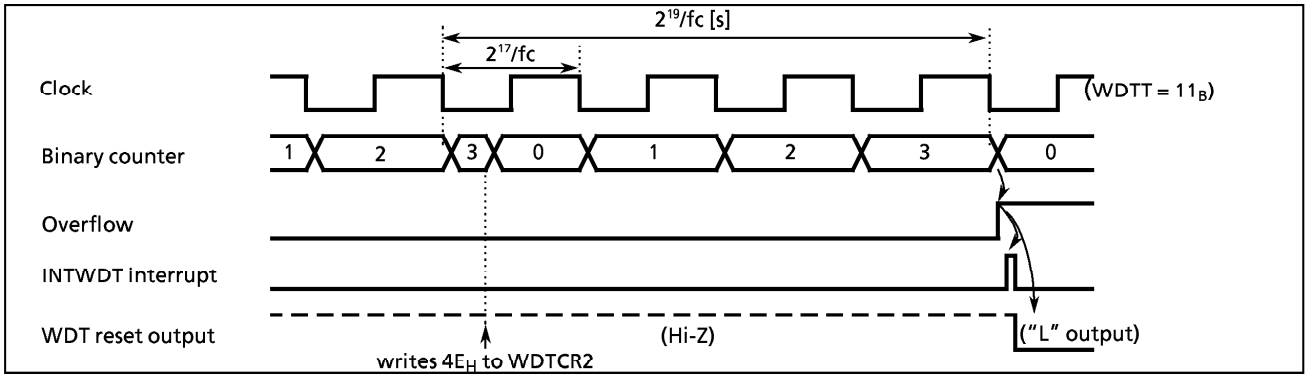


Figure 1-29. Watchdog Timer Interrupt / Reset

1.11 Reset Circuit

The 87CH48 each have four types of reset generation procedures: an external reset input, an address trap reset, a watchdog timer reset and a system clock reset. Table 1-5 shows on-chip hardware initialization by reset action. The internal source reset circuit (watchdog timer reset, address trap reset, and system clock reset) is not initialized when power is turned on. Thus, output from the $\overline{\text{RESET}}$ pin may go low (16/fc [s] (3 μs at 8MHz) when power is turned on.

Note : On the emulator, output from the $\overline{\text{RESET}}$ pin may go Low 2^{20}/fc [s] (131 ms at 8MHz) when power is turned on.

Table 1-5. Initializing Internal Status by Reset Action

On-chip Hardware		Initial Value	On-chip Hardware	Initial Value
Program counter	(PC)	$(FFFF_H) \cdot (FFFE_H)$	Divider of Timing generator	0
Register bank selector	(RBS)	0	Watchdog timer	Enable
Jump status flag	(JF)	1	Output latches of I/O ports	Refer to I/O port circuitry
Interrupt master enable flag	(IMF)	0	Control registers	Refer to each of control register
Interrupt individual enable flags	(EF)	0		
Interrupt latches	(IL)	0		

1.11.1 External Reset Input

When the $\overline{\text{RESET}}$ pin is held at low for at least 3 machine cycles (12/fc [s]) with the power supply voltage within the operating voltage range and oscillation stable, a reset is applied and the internal state is initialized.
When the $\overline{\text{RESET}}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses $FFFE_H$ to $FFFF_H$.
The $\overline{\text{RESET}}$ pin contains a Schmitt trigger (hysteresis) with an internal pull-up resistor. A simple power-on-reset can be applied by connecting an external capacitor and a diode,

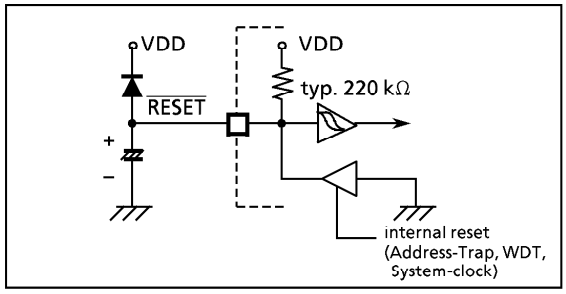
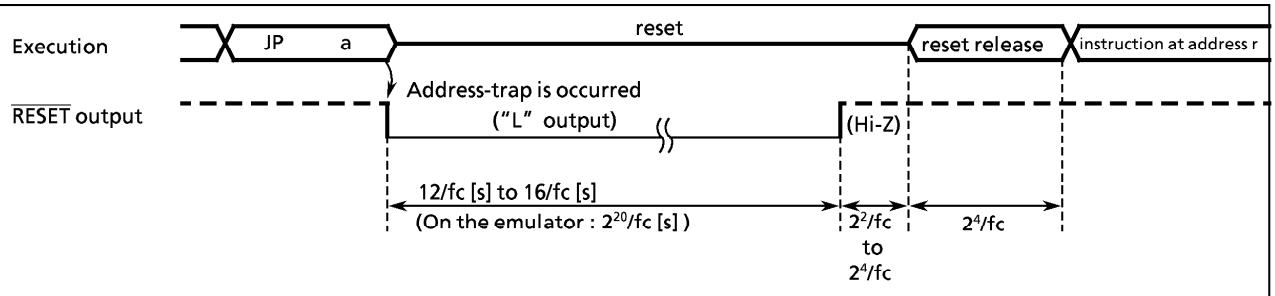


Figure 1-30. Simple Power-on-Reset Circuitry

1.11.2 Address-Trap-Reset

An address-trap-reset is one of fail-safe function that detects CPU malfunction such as endless looping caused by noise or the like, and returns the CPU to the normal state. If the CPU attempts to fetch an instruction from a specific address (0000 to $023F_H$), an internal reset (called address-trap-reset) will be generated. Then, the $\overline{\text{RESET}}$ pin output will go low. The reset time is 12/fc [s] to 16/fc [s] (1.5 μs to 2.0 μs at 8 MHz) .

Note : On the emulator, output from the $\overline{\text{RESET}}$ pin may go Low 2^{20}/fc [s] (131 ms at 8MHz) when address-trap-reset is occurred.



Note 1: $0 \leqq a \leqq 023F_H$
Note 2: *During reset release, reset vector "r" is read out, and an instruction at address r is fetched and decoded.*

Figure 1-31. Address-Trap-Reset

1.11.3 Watchdog Timer Reset

Refer to Section "1.10 Watchdog Timer".

1.11.4 System-Clock-Reset

Clearing both XEN and XTEN (bits 7 and 6 in SYSCR2) to "0" stops both high-frequency and low-frequency oscillation, and causes the MCU to deadlock. This can be prevented by automatically generating a reset signal whenever XEN = XTEN = 0 is detected to continue the oscillation. Then, the $\overline{\text{RESET}}$ pin output goes low from high-impedance. The reset time is 12/fc [s] to 16/fc [s] (1.5 μs to 2.0 μs at 8 MHz) .

Note : On the emulator, output from the $\overline{\text{RESET}}$ pin may go Low 2^{20}/fc [s] (131 ms at 8MHz) when system-clock-reset is occurred.

2. ON-CHIP PERIPHERALS FUNCTIONS

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

The TLCS-870 Series uses the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function registers (SFR) and data buffer registers (DBR).
The SFR are mapped to addresses 0000_H to $003F_H$, and the DBR to addresses $0F80_H$ to $0FFF_H$.
Figure 2-1 shows the 87CH48 SFRs and DBRs.

Address	Read	Write
0000_H	P0 port	
01	P1 Port	
02	P2 Port	
03	P3 Port	
04	P4 Port	
05	P5 Port	
06	P6 Port	
07	P7 Port	
08	P8 Port	
09		P8CR (P8 I/O control)
0A	–	P0CR (P0 I/O control)
0B	–	P1CR (P1 I/O control)
0C	–	P6CR (P6 I/O control)
0D	–	P7CR (P7 I/O control)
0E	ADCCR (A/D converter control)	
0F	ADCDR1 (A/D conv. upper 8-bit 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	reserved	
1E	reserved	
1F	reserved	

Address	Read	Write
0020_H	–	SBICR1 (SBI control)
21	SBIDBR (SBI data buffer)	
22	–	I^2CAR (I^2C Bus address)
23	SBISR (SBI Status)	SBICR2 (SBI control)
24	ADCDR2L (A/D conv. lower 8-bit result)	–
25	ADCDR2H (A/D conv. upper 2-bit result)	–
26	PWMSR (PWM Status register)	PWMCR (PWM control)
27		PWMDR (PWM data register)
28	reserved	
29	RDBUF (UART recieve data)	FDBUF (UART transmit data)
2A	UARTSR (UART Status)	UARTCR1 (UART control 1)
2B	–	UARTCR2 (UART control 2)
2C	reserved	
2D	reserved	
2E	reserved	
2F	reserved	
30	reserved	
31	reserved	
32	reserved	
33	reserved	
34	–	WDTCR1 (WDT control)
35	–	WDTCR2
36	TBTCR	(TBT / TG / DVO control)
37	EINTCR	(External interrupt control)
38	SYSCR1	(System control)
39	SYSCR2	
3A	EIR_L	(Interrupt enable register)
3B	EIR_H	
3C	IL_L	(Interrupt latch)
3D	IL_H	
3E	reserved	
3F	PSW (Program status word)	RBS (Register bank selector)

(a) Special Function Registers

Address	Read	Write
$0F80_H$	reserved	
	reserved	
0FEF	reserved	
0FF0	reserved	
F1	reserved	
F2	reserved	
F3	reserved	
F4	reserved	
F5	reserved	
F6	reserved	
F7	reserved	
0FF8	reserved	
F9	reserved	
FA	reserved	
FB	reserved	
FC	reserved	
FD	reserved	
FE	reserved	
FF	reserved	

(b) Data Buffer Registers

Note 1 : Do not access reserved areas by the program.
Note 2 : – : Cannot be accessed.
Note 3 : When defining address $003F_H$ with assembler symbols, use GPSW and GRBS.
Note 4 : Write-only registers and interrupt latches cannot use the read-modify-write instructions (bit manipulation instructions such as SET, CLR, etc. and logical operation instructions such as AND, OR, etc.)

Figure 2-1. SFR & DBR

2.2 I/O Ports

The 87CH48 has 9 parallel input/output ports (56pins) each as follows:

	Primary Function	Secondary Functions
Port P0	8-bit I/O port	———
Port P1	8-bit I/O port	external interrupt input, timer/counter input/output, and divider output
Port P2	3-bit I/O port	low-frequency resonator connections, external interrupt input, and STOP mode release signal input
Port P3	6-bit I/O port	———
Port P4	3-bit I/O port	serial interface
Port P5	5-bit I/O port	external interrupt input, and timer/counter input/output.
Port P6	8-bit I/O port	analog input
Port P7	8-bit I/O port	analog input
Port P8	4-bit I/O port	12 bit PWM output

Each output port contains a latch, which holds the output data. All input ports do not have latches, so the external input data should either be held externally until read or reading should be performed several times before processing. Figure 2-2 shows input/output timing examples.
External data is read from an I/O port in the S1 state of the read cycle during execution of the read instruction. This timing can not be recognized from outside, so that transient input such as chattering must be processed by the program.
Output data output changes in the S2 state of the write cycle during execution of the instruction which writes to an I/O port.

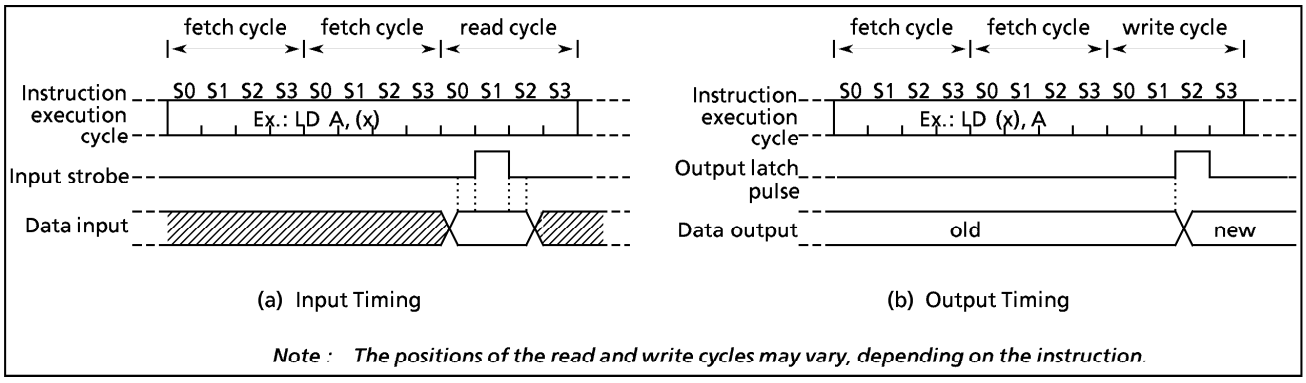


Figure 2-2. Input/Output Timing (Example)

When reading an I/O port except programmable I/O ports, whether the pin input data or the output latch contents are read depends on the instructions, as shown below:

(1) Instructions that read the output latch contents
- ① XCH r, (src)
- ② CLR/SET/CPL (src).b
- ③ CLR/SET/CPL (pp).g
- ④ LD (src).b, CF
- ⑤ LD (pp) . b, CF
- ⑥ ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), n
- ⑦ (src) side of ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (HL)

(2) Instructions that read the pin input data
- ① Instructions other than the above (1)
- ② (HL) side of ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (HL)

2.2.1 Port P0 (P07 to P00)

Port P0 is an 8-bit general-purpose input/output port which can be configured as either an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P0 input/output control register (P0CR). Port P0 is configured as an input if its corresponding P0CR bit is cleared to "0", and as an output if its corresponding P0CR bit is set to "1".

During reset, P0CR is initialized to "0", which configures port P0 as input. The P0 output latches are also initialized to "0". Data is written into the output latch regardless of the P0CR contents. Therefore initial output data should be written into the output latch before setting P0CR.

Note 1 : Ports set to the input mode read the pin states. When input pin and output pin exist in port P0 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.

Note 2 : The P0CR is a write-only register. It can not be operated by the read-modify instruction (Bit manipulation instructions of SET, CLR, etc. and Arithmetic instructions of AND, OR, etc.)

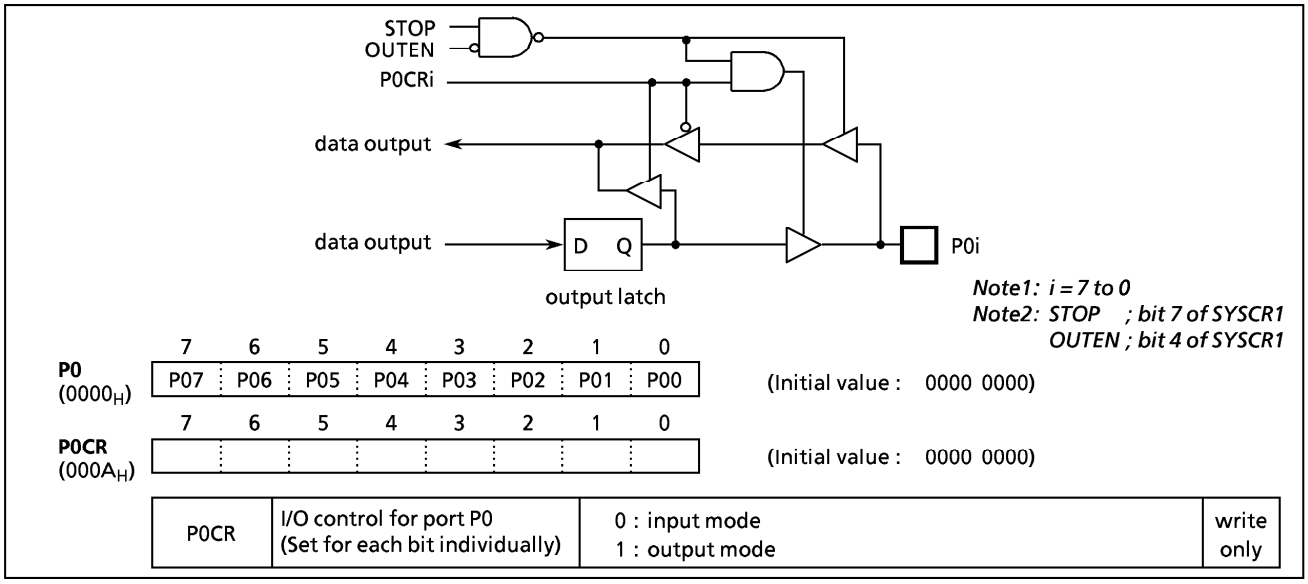


Figure 2-3. Port P0 and P0CR

Example : Setting the upper 4 bits of port P0 as an input port and the lower 4 bits as an output port (Initial output data are 1010_B).

```
LD      (P0), 00001010B     ;  Sets initial data to P0 output latches
LD      (P0CR), 00001111B   ;  Sets the port P0 input/output mode
```

2.2.2 Port P1 (P17 - P10)

Port P1 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P1 input/output control register (P1CR). Port P1 is configured as an input if its corresponding P1CR bit is cleared to "0", and as an output if its corresponding P1CR bit is set to "1". During reset, the P1CR is initialized to "0", which configures port P1 as an input . The P1 output latches are also initialized to "0". Data is written into the output latch regardless of P1CR contents. Therfore initial output data should be written into the output latch before setting P1CR. Port P1 is also used as an external interrupt input, a timer/counter input, and a divider output. When used as secondary function pin, the input pins should be set to the input mode, and the output pins should be set to the output mode and beforehand the output latch should be set to "1".

It is recommended that pins P11 and P12 should be used as external interrupt inputs, timer/counter input, or input ports. The interrupt latch is set at the rising or falling edge of the output when used as output ports.

Pin P10 ($\overline{INT0}$) can be configured as either an I/O port or an external interrupt input with INT0EN (bit 6 in EINTCR). During reset, pin P10 ($\overline{INT0}$) is configured as an input port P10.

Note 1 : Ports set to the input mode read the pin states. When input pin and output pin exist in port P1 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.

Note 2 : The P1CR is a write-only register. It can not be operated by the read-modify instruction (Bit manipulation instructions of SET, CLR, etc. and Arithmetic instructions of AND, OR, etc.)

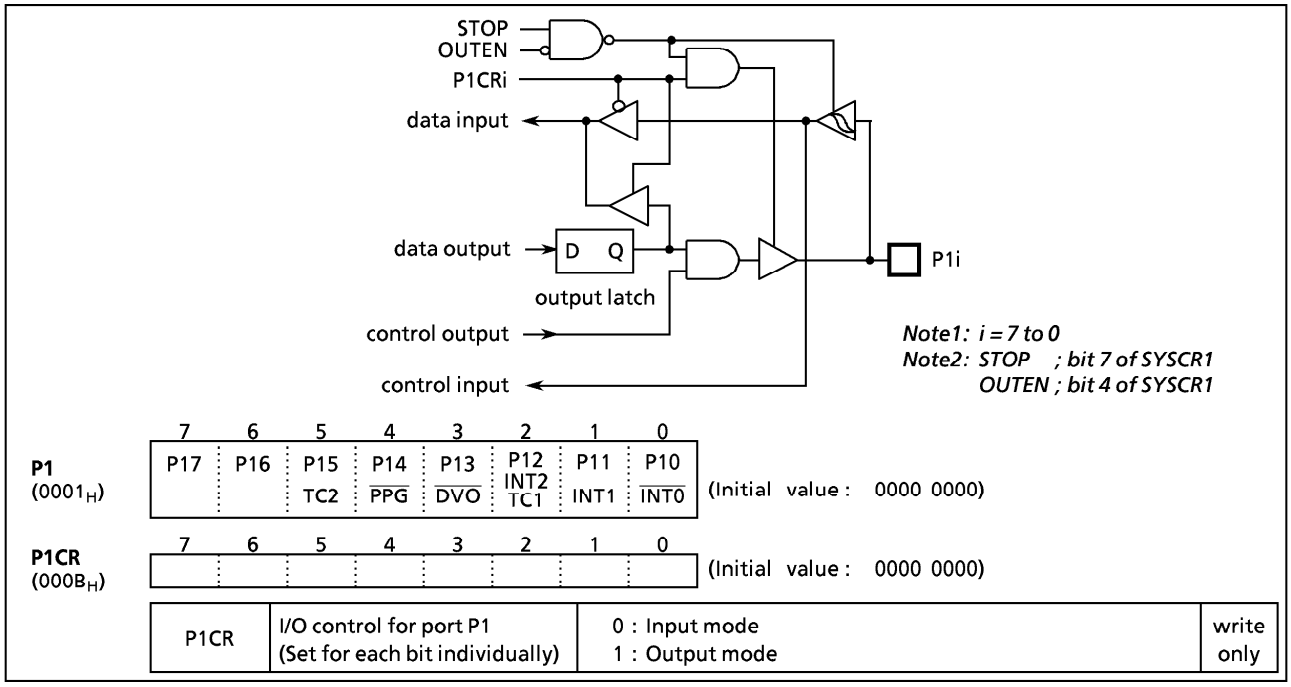


Figure 2-4. Port P1 and P1CR

Example : Sets P17, P16 and P14 as output ports, P13 and P11 as input ports, and the others as function pins. Internal output data is "1" for the P17 and P14 pins, and "0" for the P16 pin.

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

2.2.3 Port P2 (P22 to P20)

Port P2 is a 3-bit input/output port. It is also used as an external interrupt input, and low-frequency crystal connection pins When used as an input port, or a secondary function pin, the output latch should be set to "1". During reset , the output latches are initialized to "1".

A low-frequency crystal (32.768 kHz) is connected to pins P21 (XTIN) and P22 (XTOUT) in the dual-clock mode. In the single-clock mode, pins P21 and P22 can be used as normal input/output ports.

It is recommended that pin P20 should be used as an external interrupt input, a STOP mode release signal input, or an input port. If used as an output port, the interrupt latch is set on the falling edge of the output pulse.

When a read instruction is executed for port P2, bits 7 to 3 read in as "1".

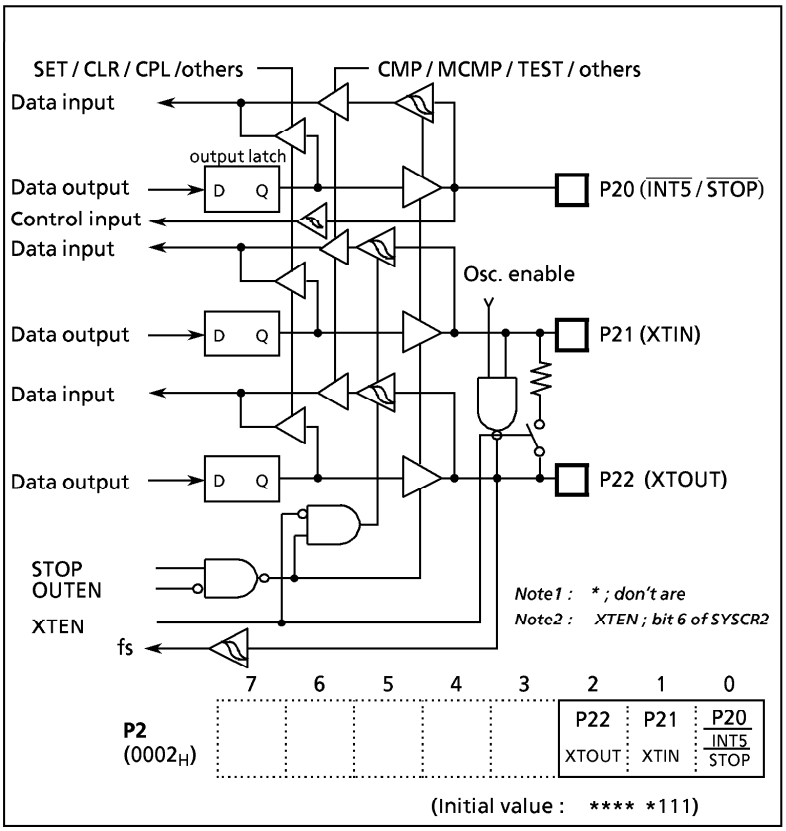


Figure 2-5. Port P2

2.2.4 Port P3 (P37 to P30)

Port P3 is an 8-bit input/output port. High current output is available so LEDs can be driven directly. When used as an input port, the output latch should be set to "1". The output latches are initialized to "1" during reset.

Example 1: Output the immediate data $5A_H$ to the P3 port.

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

Example 2: Inverts the output of the upper 4bits (P37 - P34) of the P3 port.

```
XOR   (P3), 11110000B
      ; P37 to P34←P37 to P34
```

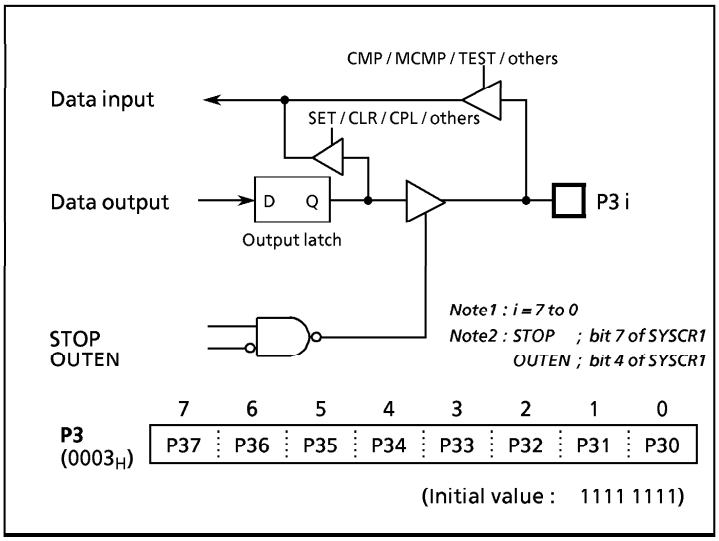


Figure 2-6. Port P3

2.2.5 Port P4 (P45 to P40)

Port P4 is an 6-bit input/output port, and is also used as serial interface (SIO/SBI, UART) input/output. When used as an input port or a secondary function pin, the output latch should be set to "1". The output latches are initialized to "1" during reset. Bits 7-6 are read in as "1" when a read instruction is executed for the port P4.

2.2.6 Port P5 (P52 to P50)

Port P5 is a 3-bit input/output port, and is also used as an external interrupt input, a timer/counter input/output. When used as an input port or a secondary function pin, the output latch should be set to "1". The output latches are initialized to "1" during reset.

Bits 7-3 are read in as "1" when a read instruction is executed for the port P5.

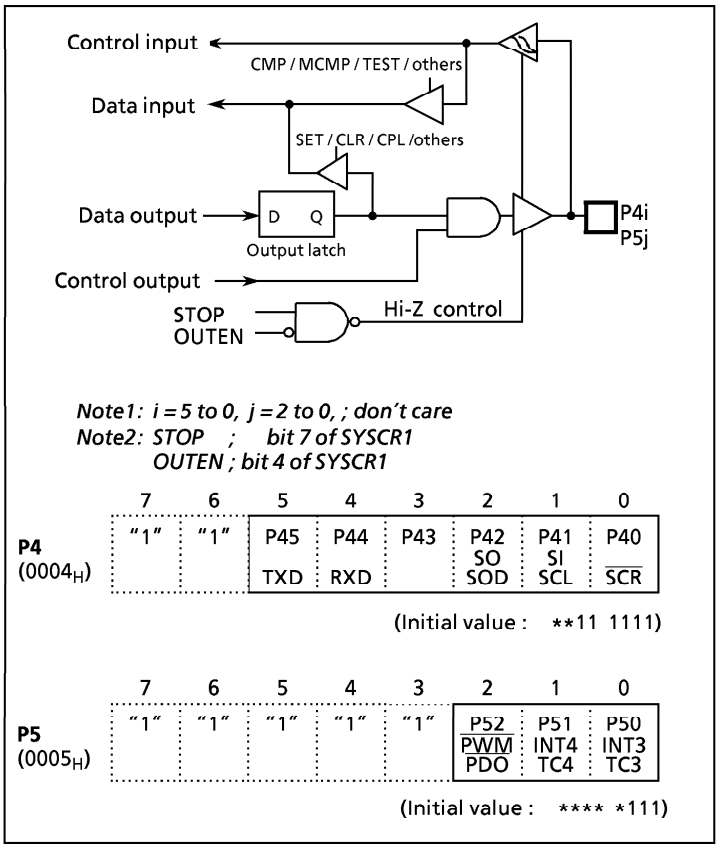
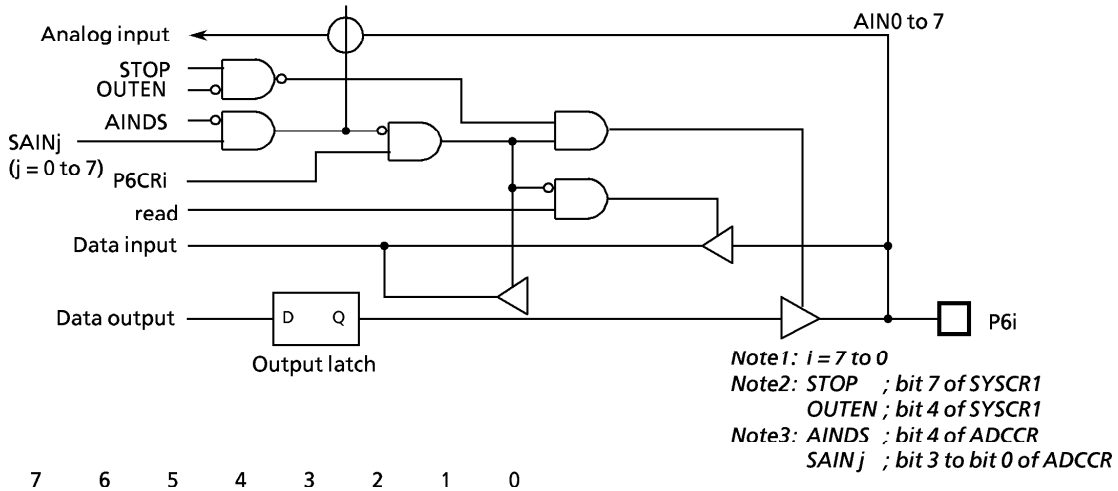


Figure 2-7. Ports P4 and P5

2.2.7 Port P6 (P67 to P60)

Port P6 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P6 input/output control register (P6CR) and AINDS (bit 4 tof ADCCR).

During reset, all bits of P6 CR and the P6 output latches are initialized to "0", AINDS is initialized to "0" and SAIN (bit 3 to bit 0 of ADCCR) is initialized to "0", therefore P60 terminal become analog input.
Nonuse pins as analog input are configured as an input or an output in one-bit unit, but during A/D conversion, please don't change the output data for about nonuse pins as analog input of P6 and the other ports to keeping the stability of A/D conversion.
During A/D conversion (at AINDS = 0), an analog input pin (selected by P6CR (i) = 0 and SAIN (j) = 1 or P6CR (i) = 1 and SAIN (j) = 1) is read in as "1", when a read instruction is executed for the port P6.
During A/D conversion (at AINDS = 0), nonuse pins as analog input configured as input port (by P6CR (i) = 0 and SAIN (j) = 0) are read in as "1" or "0" under the terminal input level, when a read instruction is executed for the port P6.
During A/D conversion (at AINDS = 0), nonuse pins as analog input configured as output port (by P6CR (i) = 1, and SAIN (j) = 0) and read in as "1" or "0" under the output latches, when a read instruction is executed for the port P6.
To use P6 port as the output pins, data is written into the output latch regardless of the P6CR contents. Therefore initial output data should be written into the output latch before setting P6CR.



P6 (0006_H)

7	6	5	4	3	2	1	0
P67 AIN7	P66 AIN6	P65 AIN5	P64 AIN4	P63 AIN3	P62 AIN2	P61 AIN1	P60 AIN0

(Initial value : 0000 0000)

P6CR ($000C_H$)

7	6	5	4	3	2	1	0

(Initial value : 0000 0000) write only

	P6CR (i) = 0			P6CR (i) = 1		
	AINDS = 0		AINDS = 1	AINDS = 0		AINDS = 1
	SAIN (j) = 1	SAIN (j) = 0	–	SAIN (j) = 1	SAIN (j) = 0	–
function of P6i	Analog input	Input port	Input port	Analog input	Output port	Output port
value at read	"1"	Terminal level	Terminal level	"1"	Output latch	Output latch

Note 1 : Ports set to the input mode read the pin states. When input pin and output in exist in port P6 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.

Note 2 : The P6CR is a write-only register. If can not be operated by the read-modify instruction (Bit manipulation instruction of SET, CLR, etc. and Arithmetic instructions of AND, OR, etc.)

Note 3 : When used as an analog input, please refer the description of A/D converter.

Figure 2-8. Port P6

2.2.8 Port P7 (P77 to P70)

Port P7 is an 8-bit input/output port which can be configured as either input or output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P7 input/output control register (P7CR) and AINDS (bit 4 of ADCCR).

During reset, all bits of P7CR and the P7 output latches are initialized to "0", AINDS is initialized to "0" and SAIN (bit 3 to bit 0 of ADCCR) is initialized to "0", therefore port P7 become analog input after setting the SAIN (bit 3 to bit 0 of ADCCR) to '1000 to 1111'.

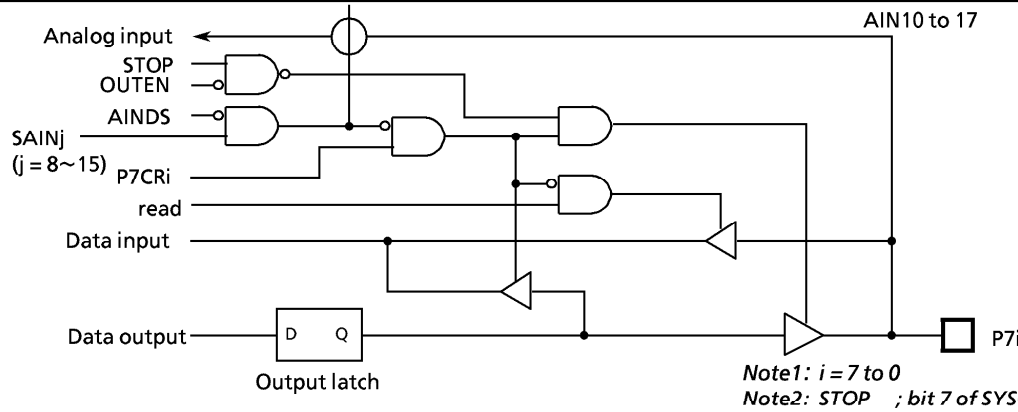
Nonuse pins as analog input are configured as input or an output in one-bit unit, but during A/D conversion, please don't change the output data for about nonuse pins as analog input of P7 and the other ports to keeping the stability of A/D conversion.

During A/D conversion (at AINDS = 0), an analog input pin (selected by P7CR (i) = 0 and SAIN (j) = 1 or P7CR (i) = 1 and SAIN (j) = 1) is read in as "1", when a read instruction is executed for the port P7.

During A/D conversion (at AINDS = 0), nonuse pins as analog input configured as input port (by P7CR (i) = 0 and SAIN (j) = 0) are read in as "1" or "0" under the terminal input level, when a read instruction is executed for the port P7.

During A/D conversion (at AINDS = 0), nonuse pins as analog input configured as output port (by P7CR (i) = 1, and SAIN (j) = 0) and read in as "1" or "0" under the output latches, when a read instruction is executed for the port P7.

To use P7 port as the output pins, data is written into the output latch regardless of the P7CR contents. Therefore initial output data should be written into the output latch before setting P7CR.



Note1: i = 7 to 0
Note2: STOP ; bit 7 of SYSCR1
OUTEN ; bit 4 of SYSCR1
Note3: AINDS ; bit 4 of ADCCR
SAIN j ; bit 3 to bit 0 of ADCCR

P7 (0007_H)

7	6	5	4	3	2	1	0
P77 AIN17	P76 AIN16	P75 AIN15	P74 AIN14	P73 AIN13	P72 AIN12	P71 AIN11	P70 AIN10

(Initial value 0000 0000)

P7CR ($000D_H$)

7	6	5	4	3	2	1	0

(Initial value 0000 0000) Write Only

	P7CR (i) = 0			P7CR (i) = 1		
	AINDS = 0		AINDS = 1	AINDS = 0		AINDS = 1
	SAIN (j) = 1	SAIN (j) = 0	–	SAIN (j) = 1	SAIN (j) = 0	–
function of P7i	Analog input	Input port	Input port	Analog input	Output port	Output port
value at read	"1"	Terminal level	Terminal level	"1"	Output latch	Output latch

Note 1 : Ports set to the input mode read the pin states. When input pin and output in exist in port P7 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.

Note 2 : The P7CR is a write-only register. If can not be operated by the read-modify instruction (Bit manipulation instruction of SET, CLR, etc. and Arithmetic instructions of AND, OR, etc.)

Note 3 : In the case of being used as analog input, refer to A/D converter (2.10).

Figure 2.9 (a) Port P7 and P7CR

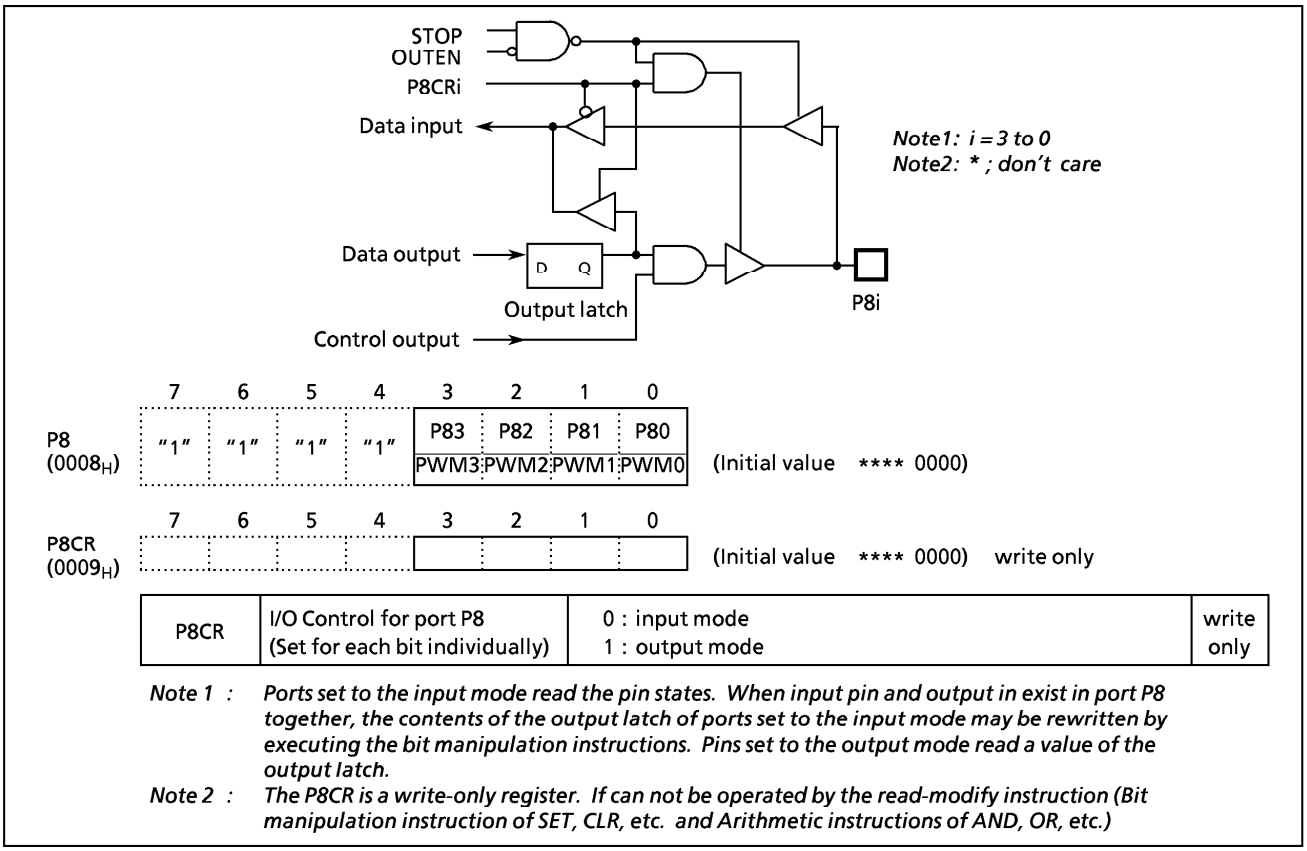
2.2.9 Port P8 (P83 to P80)

Port P8 is an 4-bit general-purpose input/output port which can be configured as either an input or an output in one-bit unit under software control. Input/output mode is specified as an input if its corresponding P8CR bit is cleared to "0", and as an output if its corresponding P8CR bit is set to "1".

During reset, P8CR is initialized to "0", which configures port P8 as input. The P8 output latches are also initialized to "0". To use P8 port as the output pins, data is written into the output latch regardless of the P8CR contents. Therefore initial output data should be written into the output latch before setting P8CR.

Bits 7-4 are read in as "1" when a read instruction is executed for the port P8.

To use P8 port as the PWM0-3 pin, set port output latches to 1.



Note 1 : Ports set to the input mode read the pin states. When input pin and output in exist in port P8 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.

Note 2 : The P8CR is a write-only register. If can not be operated by the read-modify instruction (Bit manipulation instruction of SET, CLR, etc. and Arithmetic instructions of AND, OR, etc.)

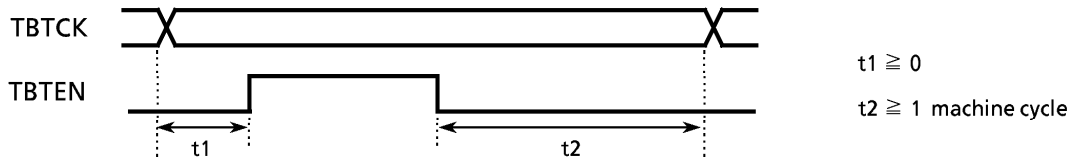
Figure 2.9 (b) Port P8 and P8CR

2.3 Time Base Timer (TBT)

The time-base timer is used to generate the base time for key scan and dynamic display processing. For this purpose, it generates a time-base timer interrupt (INTTBT) at predetermined intervals.

This interrupt is generated beginning with the first rising edge of the source clock (the timing generator's divider output selected by TBTCK) after the time-base timer is enabled. Note that since the divider cannot be cleared by a program, the first interrupt only may occur earlier than the set interrupt period. (See Figure 2-10. (b).)

When selecting the interrupt frequency, make sure the time-base timer is disabled. (Do not change the selected interrupt frequency when disabling the active timer either.) However, you can select the interrupt frequency simultaneously when enabling the timer.



Example : Sets the time base timer frequency to fc/2^{16} [Hz] and enables an INTTBT interrupt.

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

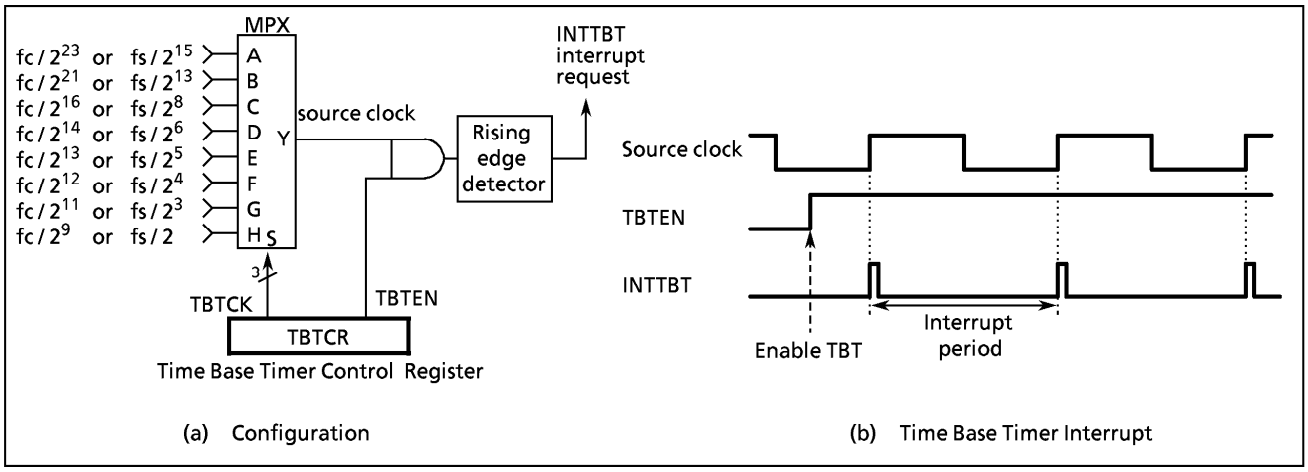


Figure 2-10. Time Base Timer

TBTCR (0036_H)

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	000 : fc / 2^{23} or fs / 2^{15} [Hz] 001 : fc / 2^{21} or fs / 2^{13} 010 : fc / 2^{16} or fs / 2^{8} 011 : fc / 2^{14} or fs / 2^{6} 100 : fc / 2^{13} or fs / 2^{5} 101 : fc / 2^{12} or fs / 2^{4} 110 : fc / 2^{11} or fs / 2^{3} 111 : fc / 2^{9} or fs / 2	

*Note : fc ; High-frequency clock [Hz], fs ; Low-frequency clock [Hz], * ; don't care*

Figure 2-11. Time Base Timer and Divider Output Control Register

Table 2-1. Time Base Timer Interrupt Frequency

TBTCK	NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode	Interrupt Frequency	
	DV7CK = 0	DV7CK = 1		At fc = 8 MHz	At fs = 32.768 kHz
000	fc / 2^{23}	fs / 2^{15}	fs / 2^{15}	0.95 Hz	1 Hz
001	fc / 2^{21}	fs / 2^{13}	fs / 2^{13}	3.81	4
010	fc / 2^{16}	fs / 2^{8}	–	122.07	128
011	fc / 2^{14}	fs / 2^{6}	–	488.28	512
100	fc / 2^{13}	fs / 2^{5}	–	976.56	1024
101	fc / 2^{12}	fs / 2^{4}	–	1953.12	2048
110	fc / 2^{11}	fs / 2^{3}	–	3906.25	4096
111	fc / 2^{9}	fs / 2	–	15625	16384

2.4 Divider Output ($\overline{DVO}$)

A 50 % duty pulse can be output using the divider output circuit, which is useful for piezo-electric buzzer drive. Divider output is from pin P13 ($\overline{DVO}$). The P13 output latch should be set to "1" and then the P13 should be configured as an output mode.

Divider output circuit is controlled by the control register (TBTCR) shown in Figure 2-12.

TBTCR (0036_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 ($\overline{DVO}$) frequency selection	00 : $fc/2^{13}$ or $fs/2^5$ [Hz] 01 : $fc/2^{12}$ or $fs/2^4$ 10 : $fc/2^{11}$ or $fs/2^3$ 11 : $fc/2^{10}$ or $fs/2^2$	

*Note : fc ; High-frequency clock [Hz], fs ; Low-frequency clock [Hz], * ; don't care*

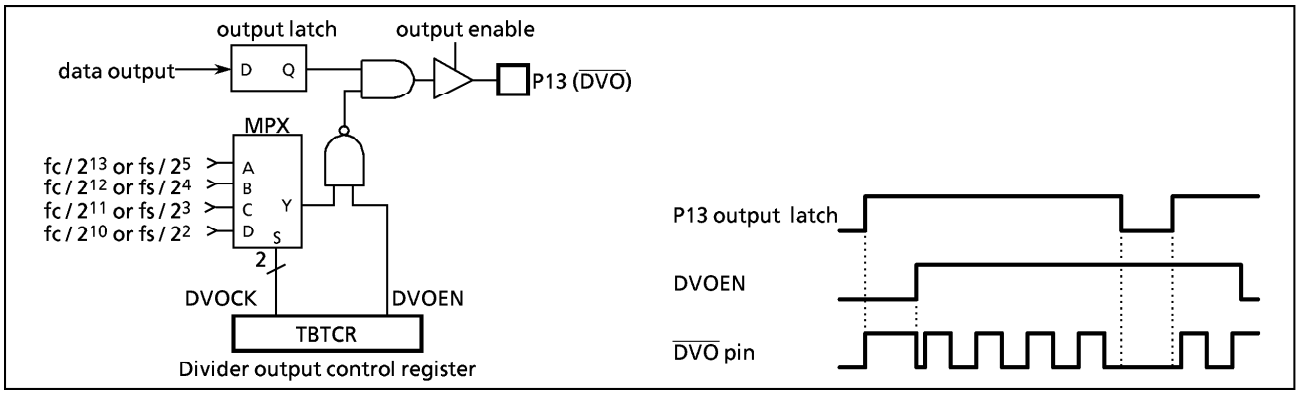
Figure 2-12. Divider Output Control Register

Example : 1 kHz pulse output (at fc = 8 MHz)

```
SET     (P1).3              ; P13 output latch ←1
LD      (P1CR), 00001000B   ; Configures P13 as an output mode
LD      (TBTCR), 10000000B  ; DVOEN←1, DVOCK←00
```

Table 2-2. Frequency of Divider Output

DVOCK	Frequency of Divider Output	At fc = 8 MHz	At fs = 32.768 kHz
00	$fc/2^{13}$ or $fs/2^5$	0.976 [kHz]	1.024 [kHz]
01	$fc/2^{12}$ $fs/2^4$	1.953	2.048
10	$fc/2^{11}$ $fs/2^3$	3.906	4.096
11	$fc/2^{10}$ $fs/2^2$	7.812	8.192



(a) Configuration (b) Timing Chart

Figure 2-13. Divider Output

2.5 16-bit Timer/Counter 1 (TC1)

2.5.1 Configuration

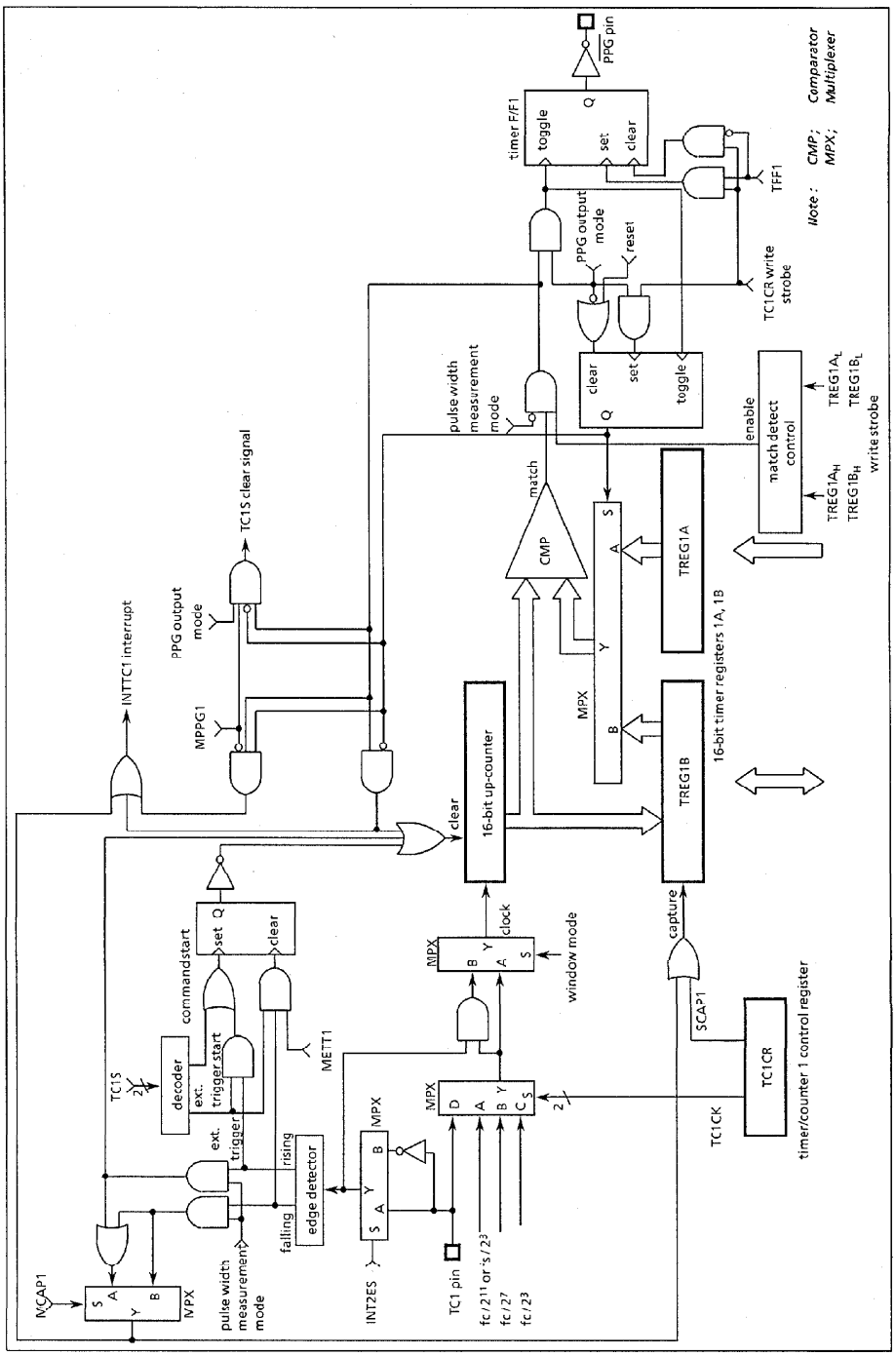


Figure 2-14. Timer/Counter 1

2.5.2 Control

The timer/counter 1 is controlled by a timer/counter 1 control register (TC1CR) and two 16-bit timer registers (TREG1A and TREG1B). Reset does not affect TREG1A and TREG1B.

TREG1A (0010, 0011_H)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$TREG1A_H$ (0011_H)								$TREG1A_L$ (0010_H)							

Write only

TREG1B (0012, 0013_H)

$TREG1B_H$ (0013_H)								$TREG1B_L$ (0012_H)							

Read / Write (Write available in only PPG output mode)

TC1CR (0014_H)

7	6	5	4	3	2	1	0
TFF1	SCAP1 MCAP1 METT1 MPPG1	TC1S		TC1CK		TC1M	

(Initial value : 0000 0000)

TC1M	TC1 mode select	00 : timer / external trigger timer / event counter mode 01 : window mode 10 : pulse width measurement mode 11 : PPG output mode	Write only
TC1CK	TC1 source clock select	00 : internal clock fc/2^{11} or fs/2^2 [Hz] 01 : internal clock fc/2^7 10 : internal clock fc/2^3 11 : external clock (TC1 pin input)	
TC1S	TC1 start control	00 : stop & counter clear 01 : command start 10 : reserved 11 : external trigger srart	
SCAP1	software capture control	0 : – 1 : software capture trigger (Note 3)	
MCAP1	pulse width measurement control	1 : double edge capture 1 : single edge capture	
METT1	external trigger timer control	0 : trigger srart 1 : trigger start & stop	
MPPG1	PPG output control	0 : cotinuous pulse 1 : single pulse	
TFF1	timer F/F1 control for PPG output mode	0 : clear 1 : set	

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

Note 2 : Writing to the low-byte of the timer registers ($TREG1A_L$, $TREG1B_L$), the comparison is inhibited until the high-byte ($TREG1A_H$, $TREG1B_H$) is written.

Note 3 : Set the mode, source clock, edge (INT2ES), PPG control and timer F/F control when TC1 stops (TC1S = 00). After writing to the high-byte, the comparison of 1 machine cycle (during instruction execution) is ignored.

Note 4 : Software capture can be used in only timer and event counter modes.

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 the PPG output mode.

Note 7 : TC1CR is a write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

Note 8 : TREG1B can be written only in PPG output mode.

Figure 2-15. Timer Registers and TC1 Control Register

2.5.3 Function

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

(1) **Timer** Mode

In this mode, counting up is performed using the internal clock. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to"0". Counting up resumes after the counteriscleared. The current contents of up-counter can be transfered to TREG1B by setting SCAP1 (bit 6 in TC1CR) to "1" (software capture function). SCAP1 is automatically cleared to "0" after capaturing.

Table 2-3. Timer/Counter 1 Source Clock (Internal Clock)

Source clock			Resolution		Maximum time setting	
NORMAL1/2, IDLE1/2 modes		SLOW, SLEEP modes				
DV7CK = 0	DV7CK = 1		At fc = 8 MHz	At fs = 32.768 kHz	At fc = 8 MHz	At fs = 32.768 kHz
$fc/2^3$ [Hz]	$fc/2^3$ [Hz]	–	1 μs	–	65.5 ms	–
$fc/2^7$	$fc/2^7$	–	16 μs	–	1.0 s	–
$fc/2^{11}$	$fs/2^3$	$fs/2^3$ [Hz]	256 μs	244.14 μs	16.8 s	16.0 s

Example 1 : Sets the timer mode with source clock $fs/2^3$ [Hz] and generates an interrupt 1s later (at fs = 32.768 kHz).

```
LD      (TC1CR), 00000000B      ; Sets the TC1 mode and source clock
LDW     (TREG1A), 1000H         ; Sets the timer register (1 s ÷ 2^3 / fs = 1000H)
SET     (EIRL).EF4              ; enable INTTC1
EI
LD      (TC1CR), 00010000B      ; Starts TC1
```

Example 2 : Software capture

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

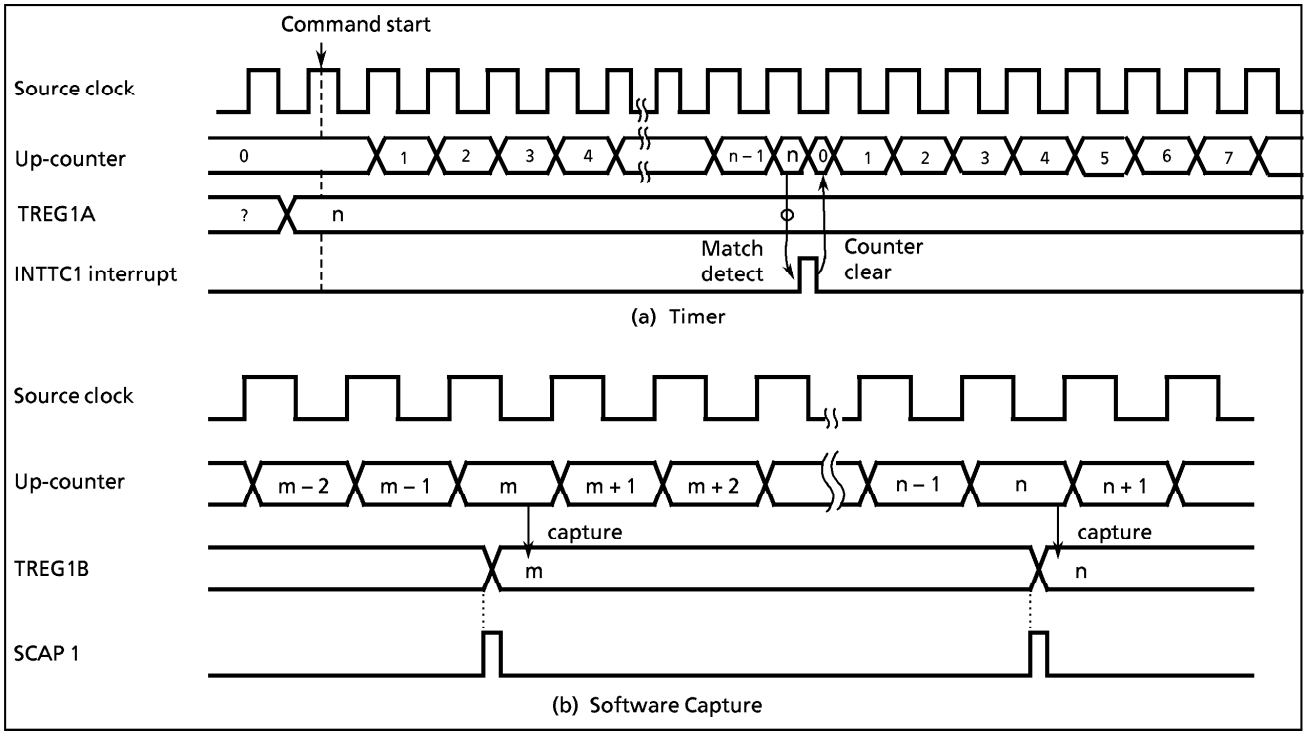


Figure 2-16. Timer Mode Timing Chart

(2) **External Trigger Timer** mode

In this mode, counting up is started by an external trigger. This trigger is the edge of the TC1 pin input. Either the rising or falling edge can be selected with INT2ES. Edge selection is the same as for the external interrupt input INT2 pin. Source clock is used an internal clock selected with TC1CK. The contents of TREG1A is compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to"0" and halted. The counter is restarted by the selected edge of the TC1 pin input.

The TC1 pin input has the same noise rejection as the INT2 pin; therefore, pulses of 7/fc [s] or less are rejected as noise. A pulse width of 24/fc [s] or more is required for edge detection in NORMAL1/2 or IDLE1/2 mode. The noise rejection circuit is turned off in SLOW and SLEEP modes. But, a pulse width of 4/fs [s] or more is required.

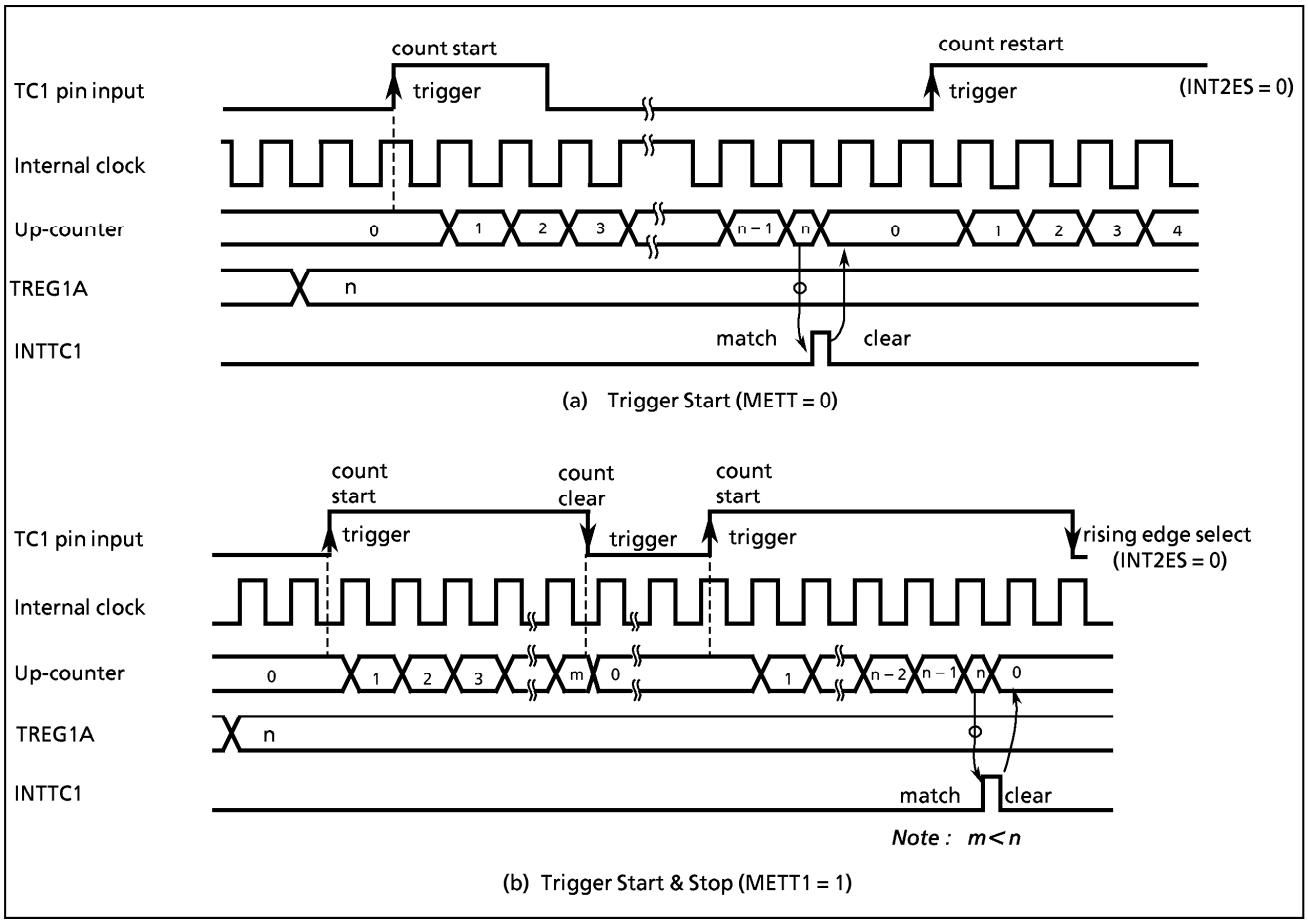


Figure 2-17. External Trigger Timer Mode Timing Chart

(3) **Event Counter** Mode

In this mode, events are counted on the edge of the TC1 pin input. Either the rising or falling edge can be selected with INT2ES in EINTCR. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. The maximum applied frequency is fc/2^4 [Hz] in NORMAL1/2 or IDLE1/2 mode and fs/2^4 [Hz] in SLOW or SLEEP mode.

Setting SCAP1 to "1" transferres the current contents of up-counter to TREG1B (software capture function). SCAP is automatically cleared after capturing.

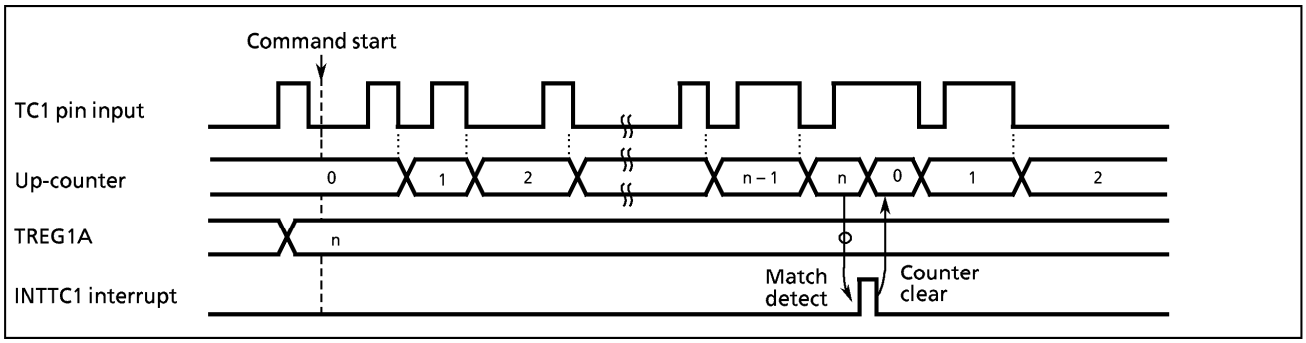


Figure 2-18. Event Counter Mode Timing Chart (INT2ES = 1)

(4) **Window** mode

Counting up is performed on the rising edge of the pulse that is the logical AND-ed product of the TC1 pin input (window pulse) and an internal clock. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Positive or negative logic for the TC1 pin input can be selected with INT2ES. Setting SCAP1 to "1" transferes the current contents of up-counter to TREG1B. It is necessary that the maximum applied frequency (TC1 input) be such that the counter value can be analyzed by the program. That is, the frequency must be considerably slower than the selected internal clock.

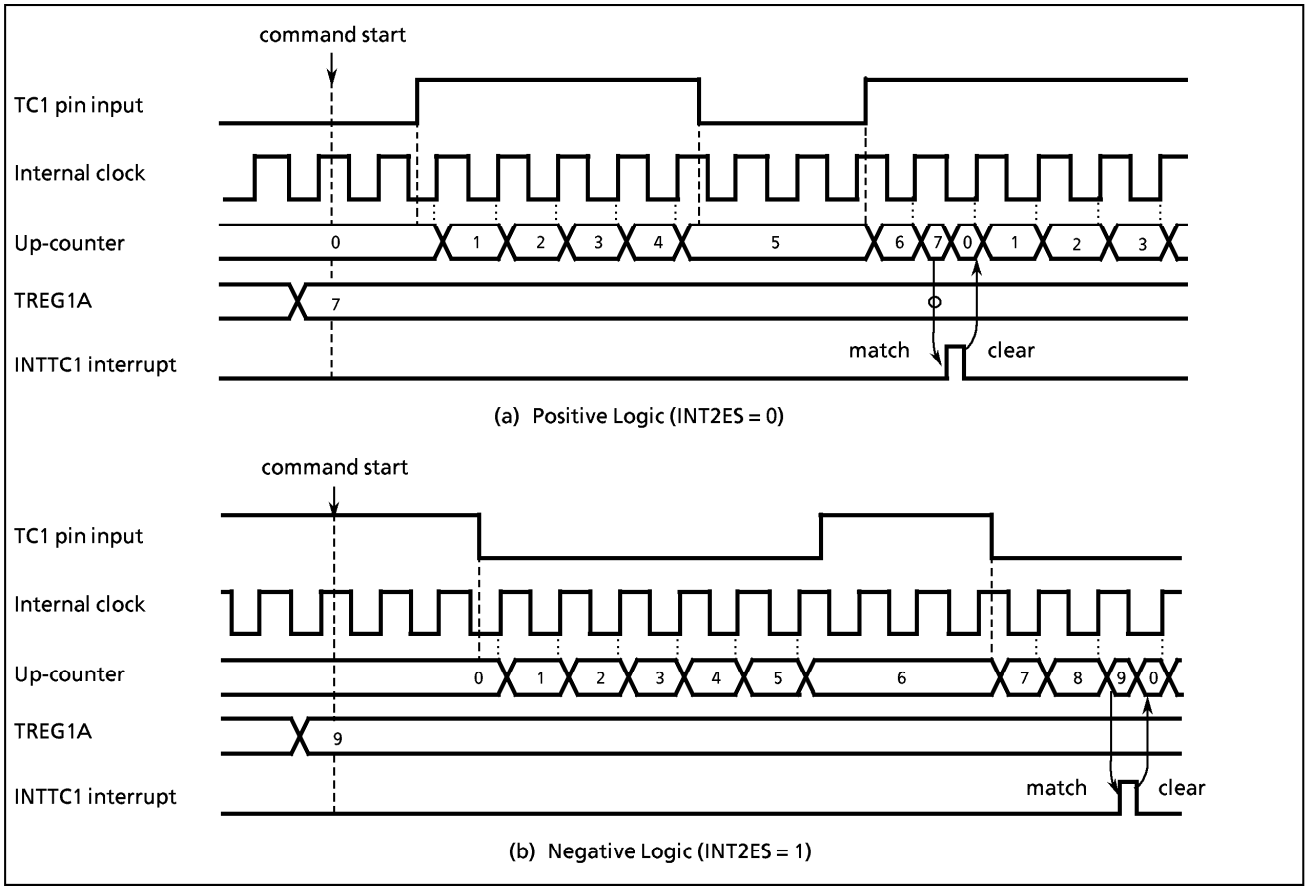


Figure 2-19. Window Mode Timing Chart

(5) **Pulse width measurement** mode

Counting is started by the external trigger (set to external trigger start by TC1S). The trigger can be selected either the rising or falling edge of the TC1 pin input. The source clock is used an internal clock. On the next falling (rising) edge, the counter contents are transferred to TREG1B and an INTTC1 interrupt is generated. The counter is cleared when the single edge capture mode is set. When double edge capture is set, the counter continues and, at the next rising (falling) edge, the counter contents are again transferred to TREG1B. If a falling (rising) edge capture value is required, it is necessary to read out TREG1B contents until a rising (falling) edge is detected. Falling or rising edge is selected with INT2ES, and single edge or double edge is selected with MCAP1 (bit 6 in TC1CR).

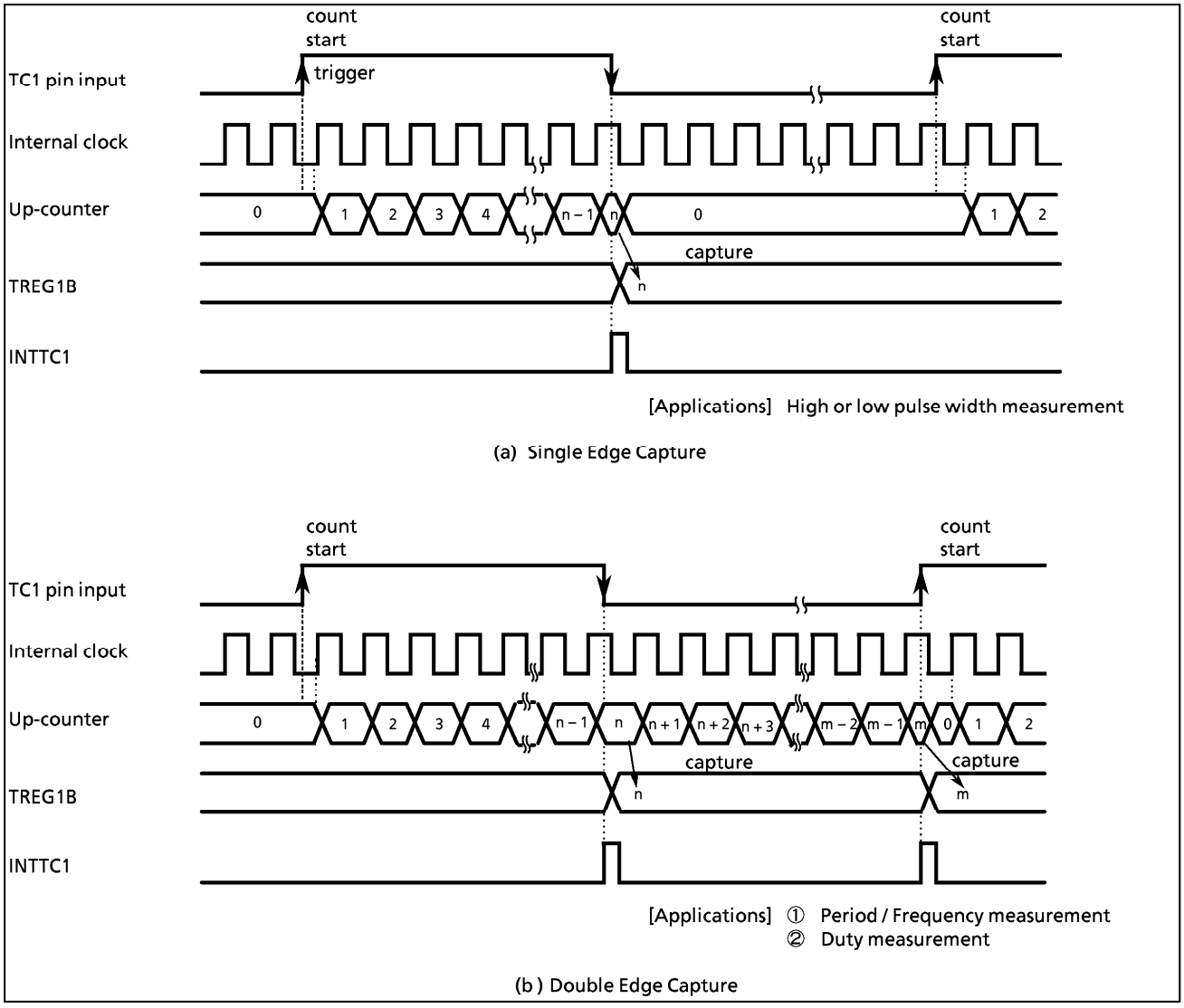


Figure 2-20. Pulse Width Measurement Mode Timing Chart

Example : Duty measurement (Resolution fc/2^7 [Hz])

```
                CLR     (INTTC1SW). 0           ;   INTTC1 service switch initial setting
                LD      (EINTCR), 00000000B     ;   Sets the rise edge at the INT2 edge
                LD      (TC1CR), 00000110B      ;   Sets the TC1 mode and source clock
                SET     (EIRL). 4               ;   Enables INTTC1
                EI
                LD      (TC1CR), 00110110B      ;   Starts TC1 with an external trigger
                 :
PINTTC1 :       CPL     (INTTC1SW). 0           ;   Complements INTTC1 service switch
                JRS     F, SINTTC1
                LD      (HPULSE), (TREG1BL)     ;   Reads TREG1B
                LD      (HPULSE + 1), (TREG1BH)
                RETI
SINTTC1 :       LD      (WIDTH), (TREG1BL)      ;   Reads TREG1B (Period)
                LD      (WIDTH + 1), (TREG1BH)
                 :
                RETI
                 :
VINTTC1 :       DW      PINTTC1
```

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

Counting is started by an edge of the TC1 pin input (either the rising or falling edge can be selected) or by a command. The source clock is used an internal clock. First, the contents of TREG1B are compared with the contents of the up-counter. If a match is found, timer F/F1 output is toggled. Next, timer F/F1 is again toggled and the counter is cleared by matching with TREG1A. An INTTC1 interrupt is generated at this time. Timer F/F output is connected to the P14 ($\overline{PPG}$) pin. In the case of $\overline{PPG}$ output, set the P14 output latch to "1" and configure as an output with $P1CR_4$. Timer F/F1 is cleared to "0" during reset. The timer F/F1 value can also be set by program and either a positive or negative logic pulse output is available. Also, writing to the TREG1B is not possible unless the timer / counter 1 is set to the PPG output mode with TC1M.

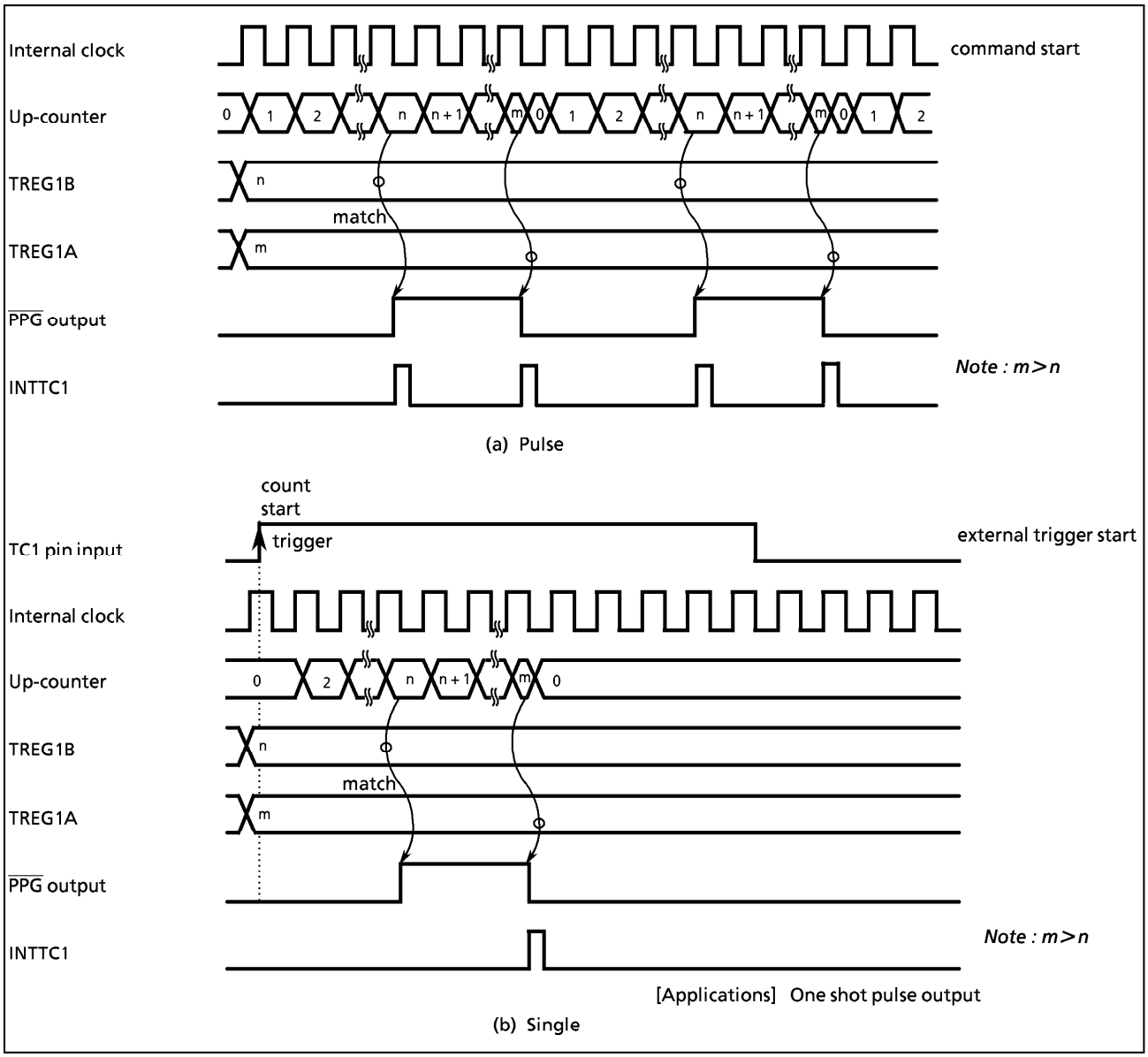


Figure 2-21. PPG Output Mode Timing Chart

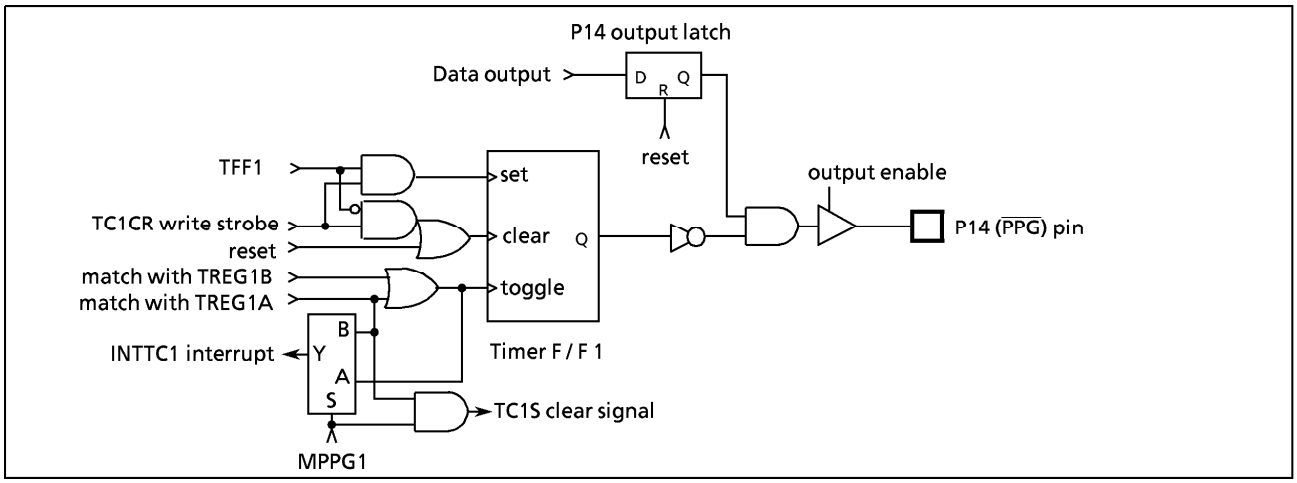


Figure 2-22. PPG Output

2.6 16-bit Timer/Counter 2 (TC2)

2.6.1 Configuration

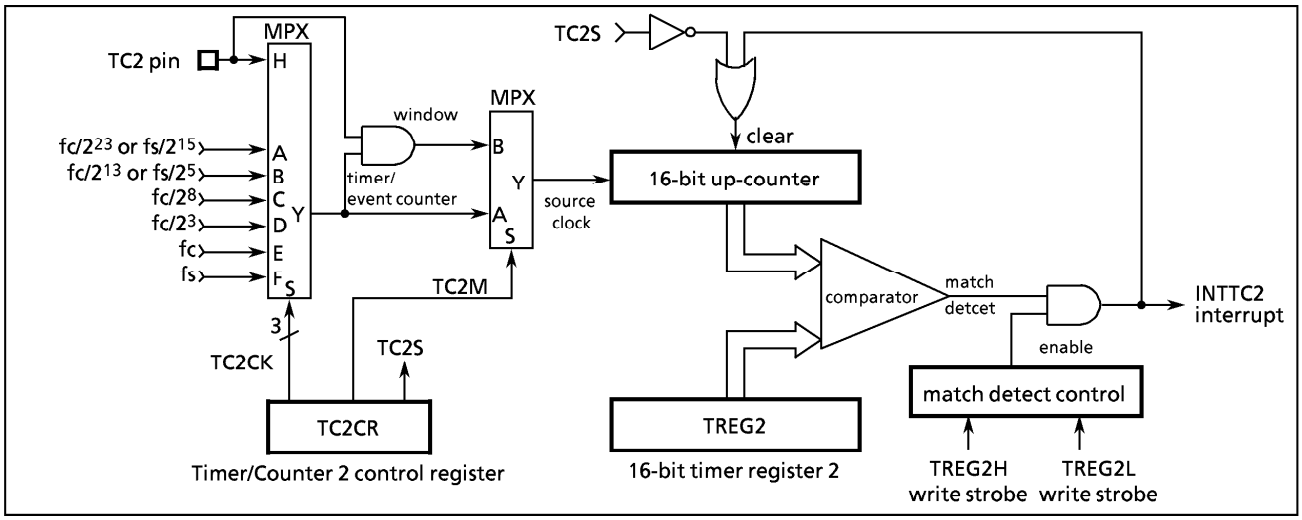


Figure 2-23. Timer/Counter 2 (TC2)

2.6.2 Control

The timer/counter 2 is controlled by a timer/counter 2 control register (TC2CR) and a 16-bit timer register 2 (TREG2). Reset does not affect TREG2.

TREG2 (0016, 0017$_H$)

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

write only

TC2CR (0015$_H$)

7	6	5	4	3	2	1	0
		TC2S	TC2CK				TC2M

(Initial value : **00 00*0)

TC2M	Timer/counter 2 operating mode select	0 : Timer/Event counter mode 1 : Window mode	write only
TC2CK	Timer/counter 2 source clock select	000 : Internal clock fc / 2^{23} or fs / 2^{15} [Hz] 001 : 〃 fc / 2^{13} or fs / 2^5 010 : 〃 fc / 2^8 011 : 〃 fc / 2^3 100 : 〃 fc (Note 5) 101 : 〃 fs 110 : Reserved 111 : External clock (TC2 pin input)	
TC2S	Timer/counter 2 start control	0 : Stop and counter clear 1 : Start	

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

Note 2 : When writing to the low-byte of timer register 2 (TREG2$_L$), the comparison is inhibited until the high-byte (TREG2$_H$) is written.
After writing to the high-byte, any match during 1 machine cycle (instruction execution cycle) is ignored.

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

Note 4 : Values to be loaded to the timer register must satisfy the following condition.
TREG2 > 0 (TREG2$_{15\sim11}$>0 when warm-up).

Note 5 : "fc" can be selected as the source clock only in the timer mode during the SLOW mode.

Note 6 : Always write "0" to bit 0 in TC2CR.

Note 7: TC2CR and TREG2 are write-only registers and must not be used with any of the read-modify-write instructions.

Figure 2-24. Timer Register 2 and TC2 Control Register

2.6.3 Function

The timer/counter 2 has three operating modes: timer, event counter and window modes. Also timer/counter 2 is used for warm-up when switching from SLOW mode to NORMAL2 mode.

(1) **Timer** Mode

In this mode, the internal clock is used for counting up. The contents of TREG2 are compared with the contents of up-counter. If a match is found, a timer/ counter 2 interrupt (INTTC2) is generated, and the counter is cleared. Counting up is resumed after the counter is cleared.

Also, when fc is selected as the source clock during SLOW mode, the lower 11 bits of TREG2 are ignored and an INTTC2 interrupt is generated by matching the upper 5 bits. Thus, in this case, only the $TREG2_H$ setting is necessary.

Table 2-4. Source Clock (Internal Clock) for Timer/Counter 2

Source clock				Resolution		Maximum time setting	
NORMAL1/2, IDLE1/2 mode		SLOW mode	SLEEP mode				
DV7CK = 0	DV7CK = 1			At fc = 8 MHz	At fs = 32.768 kHz	At fc = 8 MHz	At fs = 32.768 kHz
$fc / 2^{23}$[Hz]	$fs / 2^{15}$[Hz]	$fs / 2^{15}$ [Hz]	$fs / 2^{15}$ [Hz]	1.05 s	1 s	19.1 hour	18.2 hour
$fc / 2^{13}$	$fs / 2^{5}$	$fs / 2^{5}$	$fs / 2^{5}$	1.02 ms	1 ms	1.1 min	1 min
$fc / 2^{8}$	$fc / 2^{8}$	–	–	32 μs	–	2.1 s	–
$fc / 2^{3}$	$fc / 2^{3}$	–	–	1 μs	–	65.5 ms	–
–	–	fc (Note)	–	125 ns	–	7.9 ms	–
fs	fs	–	–	–	30.5 μs	–	2 s

Note : "fc" can be used only in the timer mode.

Example : Sets the timer mode with source clock $fc/2^3$ [Hz] and generates an interrupt every 25ms (at fc = 8 MHz).

```
LD      (TC2CR), 00001100B      ; Sets the TC2 mode and source clock
LDW     (TREG2), 61A8H          ; Sets TREG2 (25 ms ÷ 2^3/fc = 61A8H)
SET     (EIRH).EF14             ; Enable INTTC2
EI
LD      (TC2CR), 00101100B      ; Starts TC2
```

(2) **Event Counter** Mode

In this mode, events are counted on the rising edge of the TC2 pin input. The contents of TREG2 are compared with the contents of the up-counter. If a match is found, an INTTC2 interrupt is generated, and the counter is cleared. The maximum frequency applied to the TC2 pin is $fc/2^4$ [Hz] in NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ [Hz] in SLOW or SLEEP mode.

Example : Sets the event counter mode and generates an INTT2 interrupt 640 counts later.

```
LD      (TC2CR), 00011100B      ; Sets the TC2 mode
LDW     (TREG2), 640            ; Sets TREG2
SET     (EIRH).EF14             ; Enable INTTC2
EI
LD      (TC2CR), 00111100B      ; Starts TC2
```

(3) **Window** Mode

In this mode, counting up is performed on the rising edge of the pulse that is the logical AND-ed product of the TC2 pin input (window pulse) and an internal clock. The internal clock is selected with TC2CK. The contents of TREG2 are compared with the contents of up-counter. If a match is found, an INTTC2 interrupt is generated, and the up-counter is cleared to "0". It is necessary that the maximum applied frequency (TC2 input) be such that the counter value can be analyzed by the program. That is, the frequency must be considerably slower than the selected internal clock.

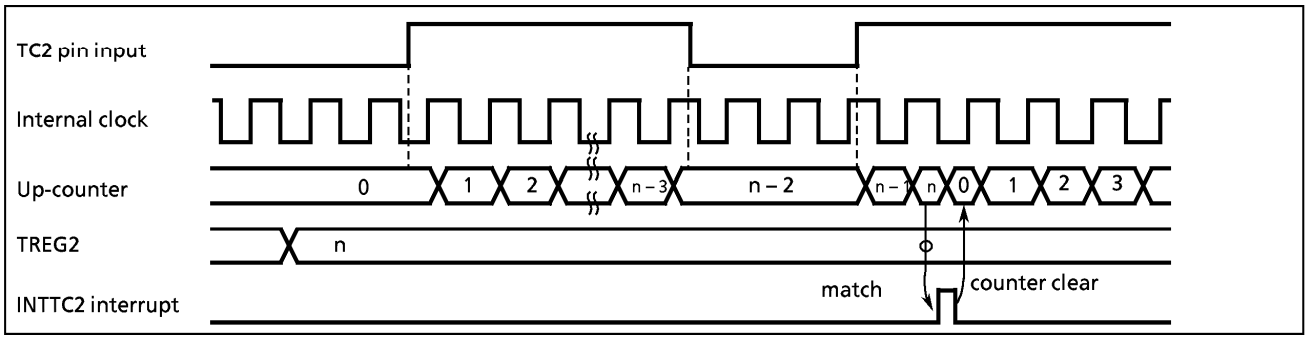


Figure 2-25. Window Mode Timing Chart

2.7 8-Bit Timer/Counter 3 (TC3)

2.7.1 Configuration

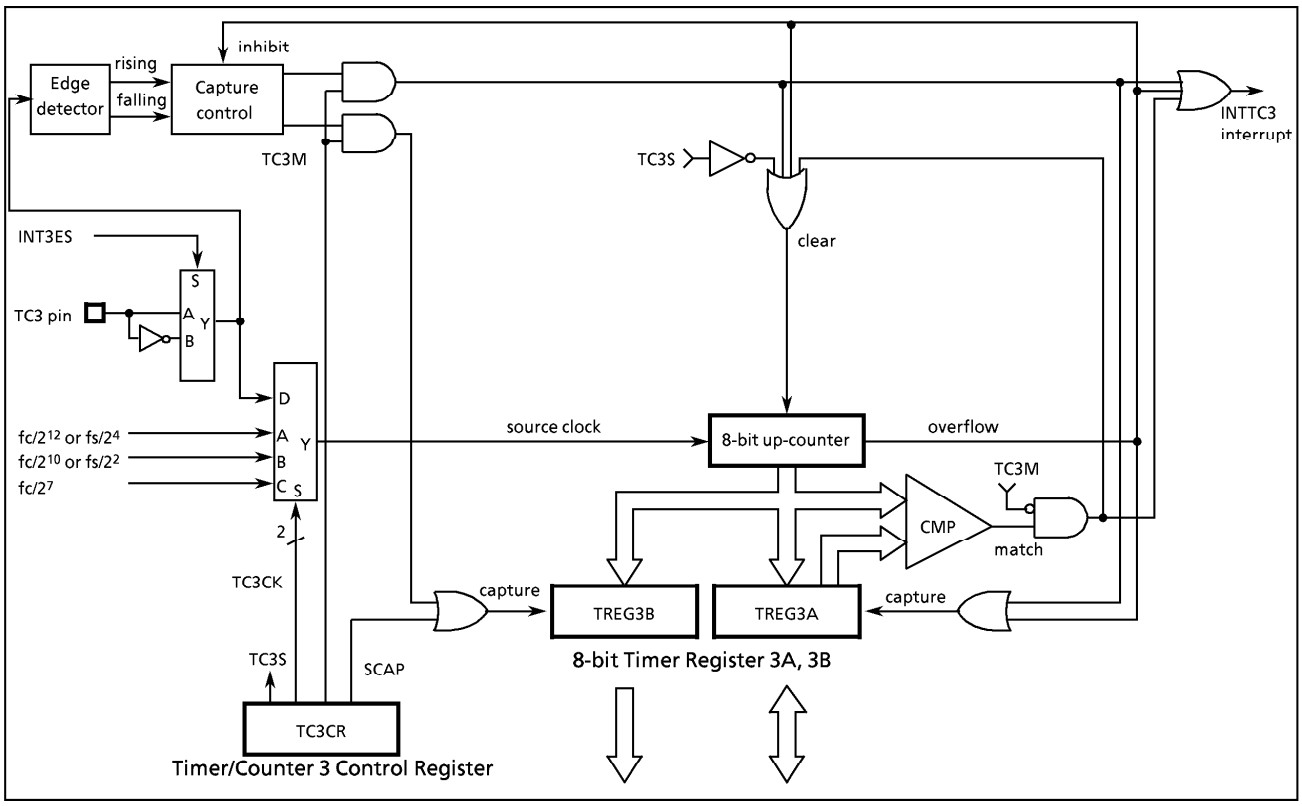


Figure 2-26. Timer/Counter 3

2.7.2 Control

The timer/counter 3 is controlled by a timer/counter 3 control register (TC3CR) and two 8-bit timer registers (TREG3A and TREG3B). Reset does not affect these timer registers.

TREG3A (0018_H) — 7 6 5 4 3 2 1 0 — Read/Write

TREG3B (0019_H) — Read only

TC3CR ($001A_H$) — 7 6 5 4 3 2 1 0 — (Initial value : *0*0 00*0)

TC3M	Timer/counter 3 operation mode set	0 : Timer/event counter 1 : Capture	Write only
TC3CK	Timer/counter 3 source clock select	00 : Internal clock $fc/2^{12}$ or $fs/2^4$ [Hz] 01 : Internal clock $fc/2^{10}$ or $fs/2^2$ 10 : Internal clock $fc/2^7$ 11 : External clock (TC3 pin input)	
TC3S	Timer/counter 3 start select	0 : Stop & clear 1 : Start	
SCAP	Software capture control	0 : – 1 : Software capture	

*Note 1 : fc ; High-frequency clock [Hz] fs ; Low-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/event counter mode)
Note 4 : TC3CR is a write-only register and must not be used with any of read-modify-write instructions.

Figure 2-27. Timer Register 3A/3B and TC3 Control Register

2.7.3 Function

The timer/counter 3 has three operating modes : timer, event counter, and capture mode.

(1) **Timer** Mode

In this mode, the internal clock is used for counting up. The contents of TREG3A are compared with the contents of up-counter. If a match is found, a timer/counter 3 interrupt (INTTC3) is generated, and the up-counter is cleared. Counting up resumes after the up-counter is cleared. The current contents of up-counter are loaded into TREG3B by setting SCAP (bit 6 in TC3CR) to "1". SCAP is automatically cleared after capturing.

Table 2-5. Source Clock (Internal Clock) for Timer Counter 3

Source clock		Resolution		Maximum setting time	
NORMAL1/2, IDLE1/2 mode	SLOW, SLEEP mode	fc = 8 MHz	fs = 32.768 kHz	fc = 8 MHz	fs = 32.768 kHz
$fc/2^{12}$ or $fs/2^4$ [Hz]	$fs/2^4$ [Hz]	512 μs	488.28 μs	131.1 ms	125 ms
$fc/2^{10}$ or $fs/2^2$	–	128 μs	122.07 μs	32.8 ms	31.25 ms
$fc/2^7$	–	16 μs	–	4.1 ms	–

(2) **Event Counter** Mode

In this mode, the TC3 pin input pulses are used for counting up. Either the rising or falling edge can be selected with INT3ES (bit 3 in EINTCR). The contents of TREG3A are compared with the contents of the up-counter. If a match is found, an INTTC3 interrupt is generated and the counter is cleared. The maximum applied frequency is $fc/2^4$ [Hz] in the NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ [Hz] in SLOW or SLEEP mode. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

The current contents of up-counter are loaded into TREG3B by setting SCAP (bit 6 in TC3CR) to "1". SCAP is automatically cleared after capturing.

Example : Generates an interrupt every 0.5 s, inputing 50 Hz pulses to the TC3 pin.

```
LD (TC3CR), 00001100B   ;  Sets TC3 mode and source clock
LD (TREG3A), 19H        ;  0.5 s ÷ 1 / 50 = 25 = 19H
LD (TC3CR), 00011100B   ;  Start TC3
```

(3) **Capture** Mode

The pulse width, period and duty of the TC3 pin input are measured in this mode, which can be used in decoding the remote control signals, etc. The counter is free running by the internal clock. On the rising (falling) edge of the TC3 pin input, the current contents of counter is loaded into TREG3A, then the up-counter is cleared and an INTTC3 interrupt is generated. On the falling (rising) edge of the TC3 pin input, the current contents of the counter is loaded into the TREG3B. In this case, counting continues. At the next rising (falling) edge of the TC3 pin input, the current contents of counter are loaded into TREG3A, then the counter is cleared again and an interrupt is generated. If the counter overflows before the edge is detected, FF_H is set to the TREG3A and an overflow interrupt (INTTC3) is generated. During interrupt processing, it can be determined whether or not there is an overflow by checking whether or not the TREG3A value is FF_H. Also, after an interrupt (capture to TREG3A, or overflow detection) is generated, capture and overflow detection are halted until TREG3A has been read out; however, the counter continues.

After TREG3A has been read out, capture and overflow detection are resumed, usually, TREG3B is read out first.

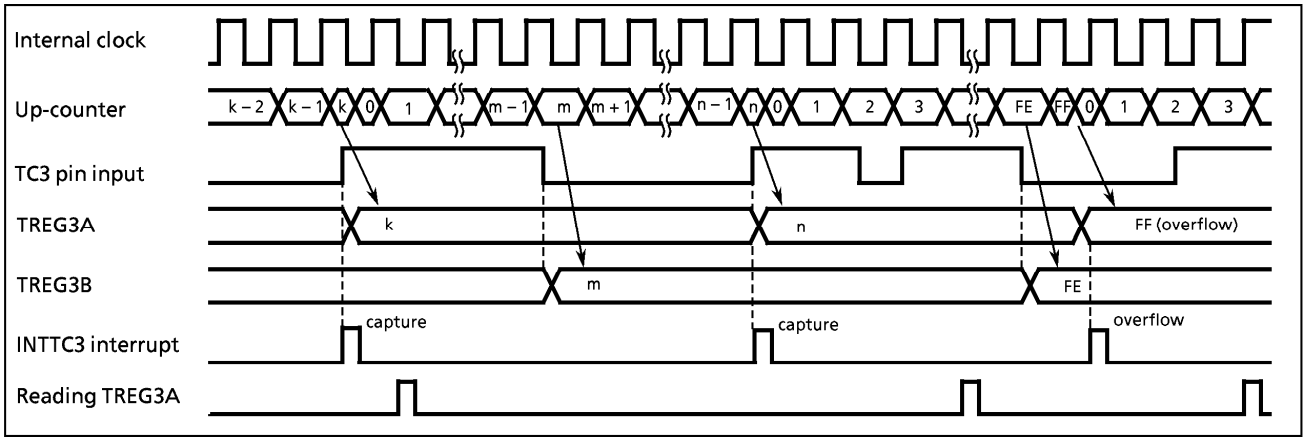


Figure 2-28. Timing Chart for Capture Mode (INT3ES = 0)

2.8 8-bit Timer/Counter (TC4)

2.8.1 Configuration

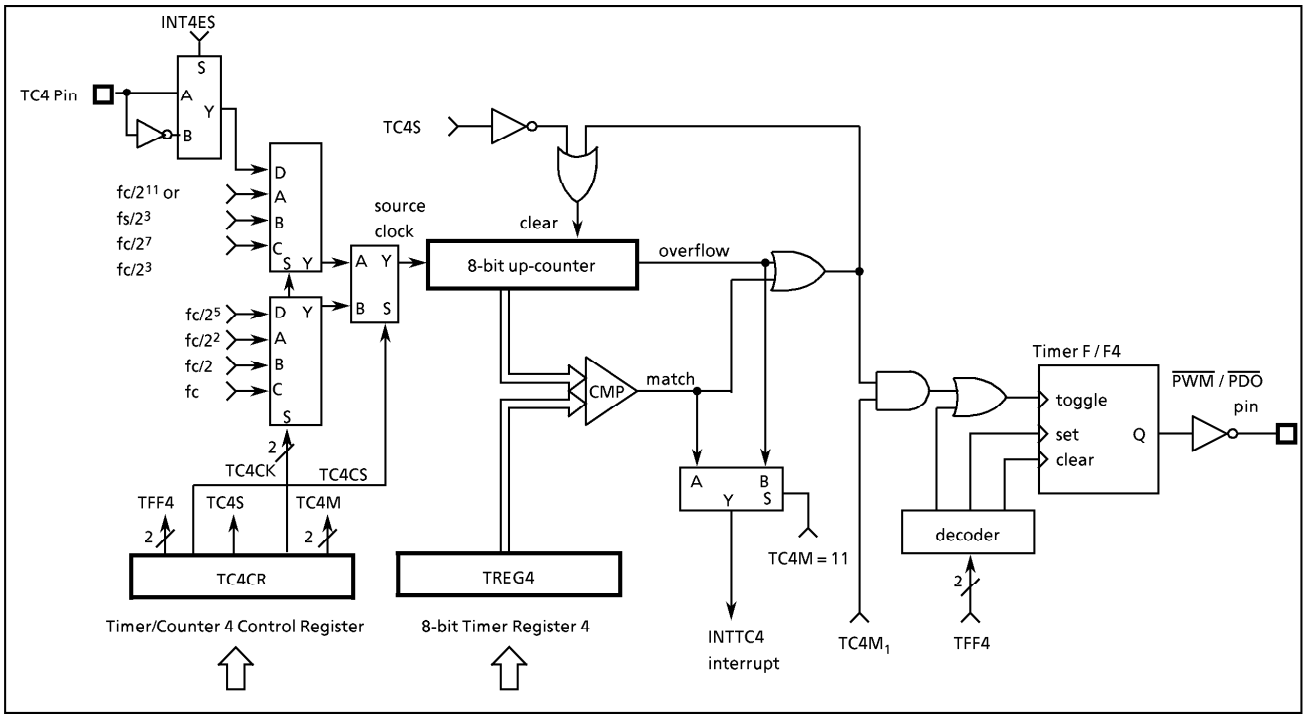


Figure 2-29. Timer/Counter 4

2.8.2 Control

The timer/counter 4 is controlled by a timer/counter 4 control register (TC4CR) and an 8-bit timer register 4 (TREG4). Reset does not affect TREG4.

TREG4 ($001B_H$)

7	6	5	4	3	2	1	0

Write only

TC4CR ($001C_H$)

7	6	5	4	3	2	1	0
TFF4		TC4CS	TC4S	TC4CK		TC4M	

(Initial value : 0000 0000)

TC4M	TC4 operating mode select	00 : Timer/event counter mode 01 : Reserved 10 : Programmable divider output (PDO) mode 11 : Pulse width modulation (PWM) output mode		write only
TC4CK	TC4 source clock select	mode A (TC4CS = 0)	mode B (TC4CS = 1)	
		00 : fc / 2^{11} or fs / 2^3 01 : fc / 2^7 10 : fc / 2^3 11 : External clock (TC4 pin input)	00 : fc / 2^5 01 : fc / 2^2 10 : fc / 2 11 : fc	
TC4S	TC4 start control	0 : Stop & clear 1 : Start		
TC4CS	TC4 mode select	0 : mode A 1 : mode B		
TFF4	Timer F/F 4 control	00 : Clear 01 : Toggle 10 : Set 11 : – (Note 3)		

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

Note 2 : Set the operating mode, the source clock selection, the timer F/F 4 control and the edge selection (INT4ES) when the TC4 stops (TC4S = 0).

Note 3 : TFF4 must be set to "11" in the timer, event counter modes and B mode.

Note 4 : Values to be loaded to the timer register must satisfy the following condition.

(a) When in PWM output mode, 5 < TREG4 < 251

(b) When any other mode than PWM output mode, 0 < TREG4

Note 5 : Source clock fc/2^2, fc/2, and fc cannot be used except in PWM output mode.

Note 6 : TC4CR and TREG4 are write-only registers and must not be used with any of read-modify-write instructions.

Note7 : TFF4 should be clear "00", when TC4S is changed from "1" to "0".
Also $\overline{PWM}/\overline{PDO}$ terminal become 'High' after changing TC4S from "1" to "0".

Figure 2-30. Timer Register 4 and TC4 Control Register

2.8.3 Function

The timer/counter 4 has four operating modes : timer, event counter, programmable divider output, and PWM output mode.

(1) **Timer** Mode

In this mode, the internal clock is used for counting up. The contents of TREG4 are compared with the contents of up-counter. If a match is found, a timer/counter 4 interrupt (INTTC4) is generated and the up-counter is cleared to "0". Counting up resumes after the up- counter is cleared.

Table 2-6. Source Clock (Internal Clock) for Timer/Counter 4

Source clock			Resolution		Maximum setting time	
Mode	NORMAL1/2, IDLE1/2 mode	SLOW, SLEEP mode	At fc = 8 MHz	At fs = 32.768 kHz	At fc = 8 MHz	At fs = 32.768 kHz
A	$fc/2^{11}$ or $fs/2^3$ [Hz]	$fs/2^3$ [Hz]	256 μs	244.14 μs	65.3 ms	62.2 ms
	$fc/2^7$	–	16 μs	–	4.1 ms	–
	$fc/2^3$	–	1 μs	–	255 μs	–
B	$fc/2^5$	–	4 μs	–	1020 μs	

(2) **Event Counter** Mode

In this mode, the TC4 pin input (external clock) pulse is used for counting up. Either the rising or falling edge can be selected with TC4ES (bit 4 in EINTCR). The contents of the TREG4 are compared with the contents of the up-counter. If a match is found, an INTTC4 interrupt is generated and the counter is cleared. The maximum applied frequency is $fc/2^4$ [Hz] in NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ [Hz] in SLOW or SLEEP mode. Two or more machine cycles are required for both the high and low levels of the pulse width.

(3) **Programmable Divider Output** (PDO) Mode

The internal clock is used for counting up. The contents of TREG4 are compared with the contents of the up-counter. Timer F/F 4 output is toggled and the counter is cleared each time a match is found. Timer F/F 4 output is inverted and output to the $\overline{PDO}$ (P55) pin. This mode can be used for 50% duty pulse output. Timer F/F 4 can be initialized by program, and it is initialized to "0" during reset. An INTTC4 interrupt is generated each time the $\overline{PDO}$ output is toggled.

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

```
LD      (TC4CR), 00000010B      ; Initializes the TC4 mode, source clock and timer F/F 4.
LD      (TREG4), 10H            ; (1/1024 ÷ 2^7/fc) ÷ 2 = 10H
LD      (TC4CR), 00010010B      ; Starts TC4
```

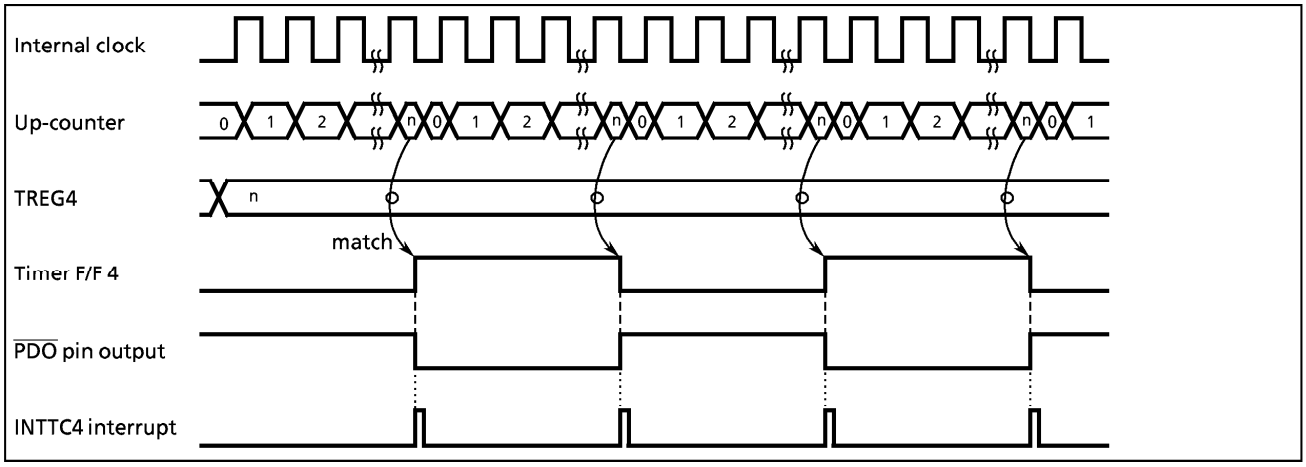


Figure 2-31. Timing Chart for PDO Mode

(4) **Pulse Width Modulation** (PWM) **Output** Mode

PWM output with a resolution of 8 bits is possible. The internal clock is used for counting up. The contents of TREG4 are compared with the contents of up-counter. If a match is found, the timer F/F 4 output is toggled. The counter continues counting. And, when an overflow occurs, the timer F/F4 output is again toggled and the counter is cleared. Timer F/F 4 output is inverted and output to the $\overline{\text{PWM}}$ (P55) pin. An INTTC4 interrupt is generated when an overflow occurs.

TREG4 is configured a 2-stage shift register and, during output, will not switch until one output cycle is completed even if TREG4 is overwritten; therefore, output can be altered continuously. Also, the first time, TREG4 is shifted by setting TC4S (bit 4 in TC4CR) to "1" after data are loaded to TREG4.

Note 1 : Do not overwrite TREG4 only when an INTTC4 interrupt is generated. Usually, TREG4 is overwritten in the routine of INTTC4 interrupt service.

Note 2 : PWM output mode can be used only in the NORMAL1, 2, and IDLE1, 2 mode.

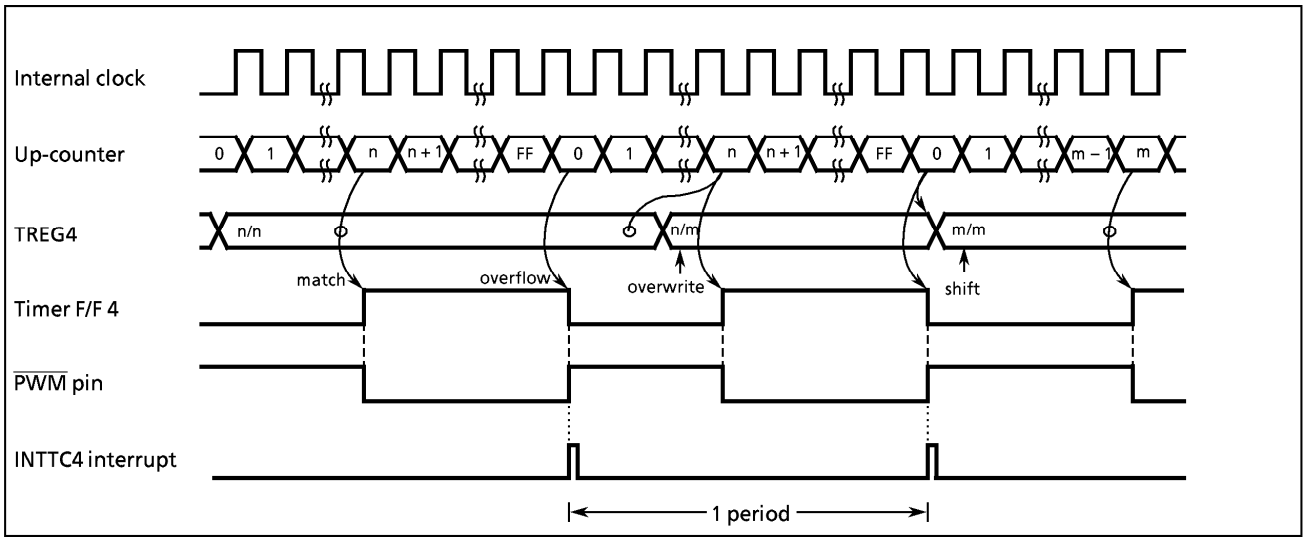


Figure 2-32 (a). Timing Chart for PWM Mode

Table 2-7. PWM Output Mode

Mode	Source clock: NORMAL1/2, IDLE1/2 mode DV7CK = 0	Source clock: NORMAL1/2, IDLE1/2 mode DV7CK = 1	Source clock: SLOW, SLEEP mode	Resolution: At fc = 8 MHz	Resolution: At fs = 32.768 kHz	Repeat cycle: At fc = 8 MHz	Repeat cycle: At fs = 32.768 kHz
A	$fc / 2^{11}$ [Hz]	$fs / 2^3$ [Hz]	$fs / 2^3$ [Hz]	256 μs	244.14 μs	65.3 ms	62.2 ms
	$fc / 2^7$	–	–	16 μs	–	4.1 ms	–
	$fc / 2^3$	–	–	1 μs	–	255 μs	–
B	$fc / 2^5$ [Hz]	–	–	4 μs	–	1024 μs	–
	$fc / 2^2$	–	–	500 ns	–	128 μs	–
	fc / 2	–	–	250 ns	–	64 μs	–
	fc	–	–	125 ns	–	32 μs	–

2.9 Serial Bus Interface (SBI-ver. B)

The 87CH48 has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit serial bus interface and an I²C bus. (a bus system by philips)

The serial bus interface is connected to an external device through P42 (SDA) and P41 (SCL) in the I²C bus mode; and through P40 ($\overline{SCK}$), P42 (SO), and P41 (SI) in the clocked-synchronous 8-bit SIO mode.

The serial bus interface pins are also used as the P4 port. When used for 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.

2.9.1 Configuration

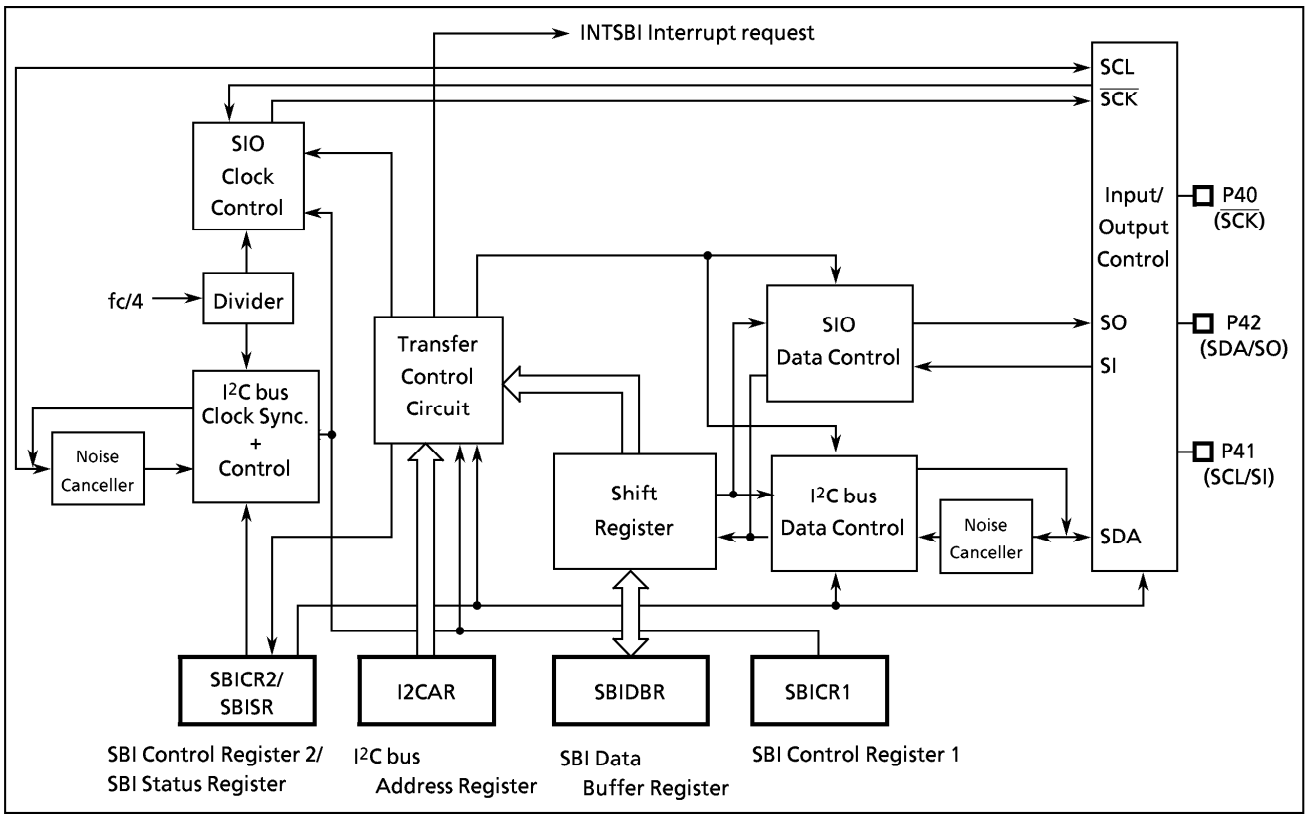


Figure 2-32 (b). Serial Bus Interface (SBI-ver. B)

2.9.2 Serial Bus Interface (SBI-ver. B) Control

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

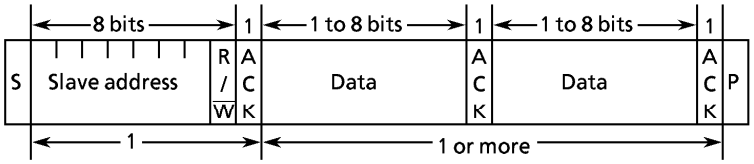
- Serial bus interface control register 1 (SBICR1)
- Serial bus interface control register 2 (SBICR2)
- Serial bus interface data buffer register (SBIDBR)
- I^2C bus address register (I2CAR)
- Serial bus interface status register (SBISR)

The above registers differ depending on a mode to be used.
Refer to Section "2.9.4 I2C bus Mode Control" and "2.9.6 Clocked-synchronous 8-bit SIO Mode Control".

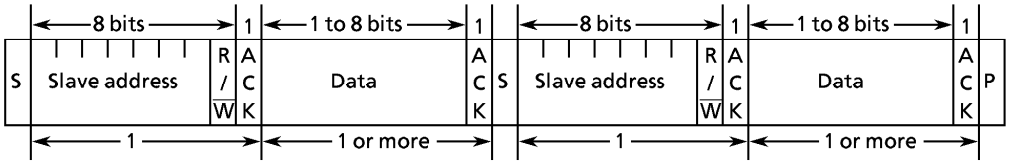
2.9.3 The Data Formats in the I^2C bus Mode

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

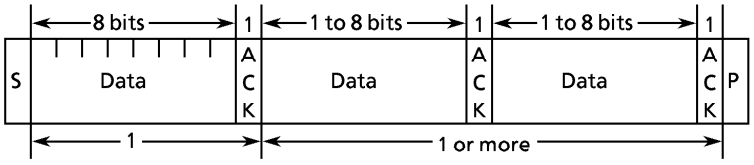
(a) Addressing format



(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 2-33. Data Format at I^2C bus mode

2.9.4 I²C Bus Mode Control

The following registers are used for control the serial bus interface (SBI-ver. B) and monitor the operation status in the I²C bus mode.

Serial Bus Interface Control Register 1

SBICR1 (0020_H)

7	6	5	4	3	2	1	0
BC			ACK	"0"	SCK		

(Initial value 0000 *000)

		BC	ACK = 0		ACK = 1		
			Number of Clock	Bits	Number of Clock	Bits	
BC	Number of transferred bits	000	8	8	9	8	Write only
		001	1	1	2	1	
		010	2	2	3	2	
		011	3	3	4	3	
		100	4	4	5	4	
		101	5	5	6	5	
		110	6	6	7	6	
		111	7	7	8	7	
ACK	Acknowledge mode specification	0 : Not generate clock pulse for acknowledge signal (in master mode) / Not count clock pulse for acknowledge signal (in slave mode) 1 : Generate clock pulse for acknowledge signal (in master mode) / Count clock pulse for acknowledge signal (in slave mode)					Read/ Write
SCK	Serial clock selection	000 : 200.0 kHz 001 : 111.1 kHz 010 : 58.8 kHz 011 : 30.3 kHz 100 : 15.4 kHz 101 : 7.75 kHz 110 : 3.89 kHz 111 : reserved	at fc = 8 MHz (Output on SCL pin)				Write only

*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 bus 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 : * ; don't care*

Serial Bus Interface Data Buffer Register

SBIDBR (0021_H)

7	6	5	4	3	2	1	0

Read / Write (Initial value 0000 0000)

Note 1 : When writing transmitted data, start from the MSB (bit7).
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.
Note 3 : The data which was written into SBIDBR is cleared to "0" when INTSBI is generated.

I²C 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	87CH48 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 2-34. Serial Bus Interface Control Register 1/Serial Bus Interface Data Buffer Register/ I²C bus Address Register in the I²C 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 : – (cannot be cleared to "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^2C 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 : Switch a mode to I2Cbus mode after confirming that input signals via port are high level.
Note 4 : SBICR2 has write-only register bits, which cannot access any of in read-modify-write instructions such as bit operate, etc.
Note 5 : Write "0" to bit 1, 0 in the SBICR2.

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 status monitor	0 : Slave 1 : Master	Read only
TRX	Transmitter/receiver status monitor	0 : Receiver 1 : Transmitter	
BB	I^2C bus status monitor	0 : Bus free 1 : Bus busy	
PIN	Interrupt service request status monitor	0 : INTSBI occurs 1 : INTSBI does not occur	
AL	Arbitration loss detection monitor	0 : Arbitration loss undetected 1 : Arbitration loss detected	
AAS	Slave address match detection monitor	0 : Slave address unmatch or "GENERAL CALL" undetected 1 : Slave address match or "GENERAL CALL" detected	
AD0	"GENERAL CALL" detection monitor	0 : "GENERAL CALL" undetected 1 : "GENERAL CALL" detected	
LRB	Last received bit monitor	0 : Last received bit "0" 1 : Last received bit "1"	

Figure 2-35. Serial Bus interface Control Register 2/Serial Bus interface status register in the I^2Cbus Mode

(1) **Acknowledge mode specification**

Set the ACK (bit 4 in SBICR1) to "1" for operation in the acknowledge mode. The 87CH48 generates an additional clock pulse for an acknowledge signal when operating in the master mode. In the transmitter mode during the clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low level in order to generate the acknowledge signal.

Reset the ACK for operation in the non-acknowledge mode. The 87CH48 does not generate a clock pulse for the acknowledge signal when operating in the master mode.

In the acknowledge mode, the 87CH48 counts a clock pulse for the acknowledge signal when operating in the slave mode. During the clock pulse, when the received slave address is the same as the value set at the I2CAR or when a GENERAL CALL is received, the SDA pin is set to the low level in order to generate the acknowledge signal.

In the transmitter mode during the clock pulse cycle after matching the slave addresses or receiving a GENERAL CALL, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low level in order to generate the acknowledge signal.

In non-acknowledge mode, the 87CH48 does not count a clock pulse for the acknowledge signal when operating in the slave mode.

(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.

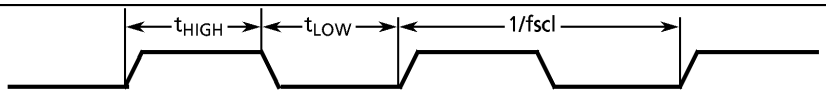
Since the BC is cleared to "000" as a start condition, a slave address and direction bit transmissions are executed in 8 bits. Other than these, the BC retains a specified value.

(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.

In both master mode and slave mode, a pulse width of at least 4 machine cycles is require for both high and low levels.



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

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

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

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 2-36. Clock Source

b. Clock synchronization

In the I^2C bus mode, in order to drive a bus with a wired-AND, a master device which pulls down a clock line to low-level, in the first place, invalidate a clock pulse of another master device which generates a high-level clock pulse. The master device with a high-level clock pulse needs to detect the situation and implement the following procedure.

The 87CH48 have a clock synchronization function for normal data transfer even when more than one master exists on a bus.

The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

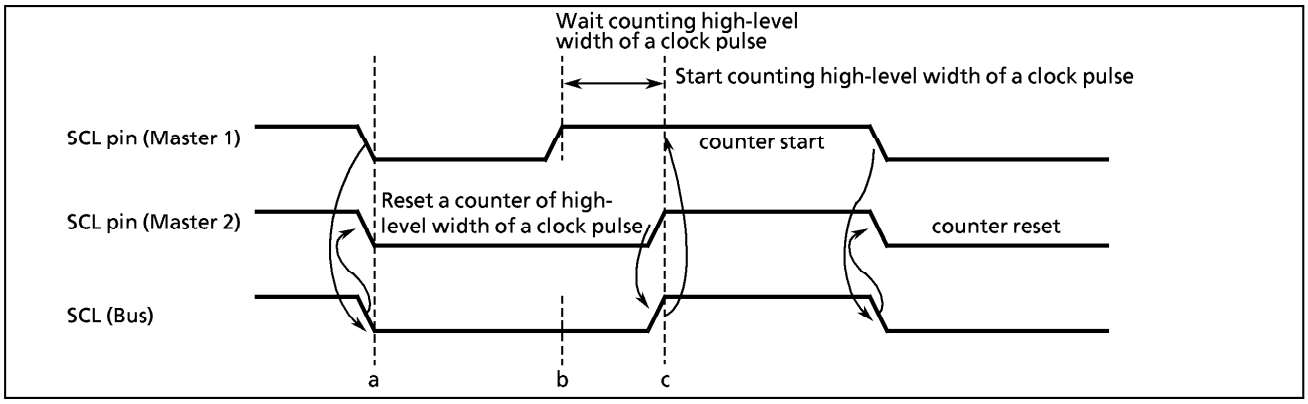


Figure 2-37. Clock Synchronization

As Master 1 pulls down the SCL pin to the low level at point "a", the SCL line of the bus becomes the low-level. After detecting this situation, Master 2 resets counting a clock pulse in the high level and sets the SCL pin to the low-level.

Master 1 finishes counting a clock pulse in the low level at point "b" and sets the SCL pin to the high level. Since Master 2 holds the SCL line of the bus at the low-level, Master 1 waits for counting a clock pulse in the high level. After Master 2 sets a clock pulse to the high level at point "c" and detects the SCL line of the bus at the high-level, Master 1 starts counting a clock pulse in the high level.

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

(4) **Slave address and Address recognition mode specification**

When the serial bus interface circuit is used with an addressing format to recognize the slave address, clear the ALS (bit 0 in I2CAR) to "0", and set the SA (bits 7 to 1 in I2CAR) to the slave address. When the serial bus interface circuit is used with a free data format not to recognize the slave address, set the ALS to "1". With a free data format, the slave address and the direction bit are not recognized, and they are processed as data from immediately after start condition.

(5) **Master/slave selection**

Set the MST (bit 7 in the SBICR2) to "1" for operating the 87CH48 as a masterdevice. Reset 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 arbitration lost is detected.

(6) **Transmitter / receiver selection**

Set the TRX (bit 6 in SBICR2) to "1" for operating the 87CH48 as a transmitter. Reset the TRX for operation as a receiver. When data with an addressing format is transferred in the slave mode, the TRX is set to "1" if the direction bit (R/$\overline{W}$) sent from the master device is "1", and is cleared to "0" if the bit is "0". In the master mode, after an acknowledge signal is returned from the slave device with the hardware, the TRX is set to "0" if a transmitted direction bit is "1", and set to "1" if it is "0". When an acknowledge signal is not returned, the current condition is maintained.

The TRX is cleared to "0" by the hardware after a stop condition on the bus is detected or arbitration is lost.

The following table shows TRX changing conditions and TRX value after changing.

Mode	Direction bit	Conditions	TRX after changing
Slave mode	0	When the received slave address is the same as I2CAR	0
	1		1
Master mode	0	When the ACK signal is returned	1
	1		0

When the serial bus interface circuit is used with a free data format, the TRX is not changed by hardware since the slave address and the direction bit are not recognized, and they are processed as data from immediately after start condition.

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

A start condition and 8-bit data are output on the bus by writing "1" to the MST, TRX and BB when the BB (bit 5 in SBICR2) is "0". It is necessary to set the transmitting data to the data buffer register and "1" to ACK beforehand.

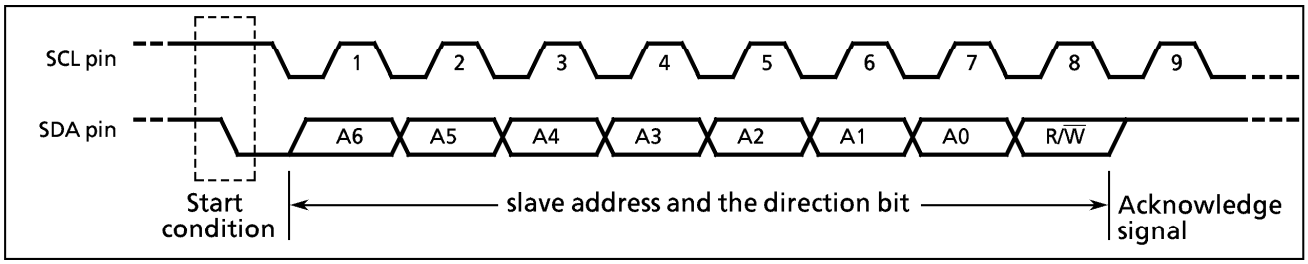


Figure 2-38. Start Condition Generation and Slave Address Generation

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.

Note : When a stop condition is generated and bus SCL line is set to Low level by the other devices, a stop condition is not started normally. Write "1" to the MST, TRX, and PIN, and "0" to the BB to generate a stop condition after releasing the SLC line.

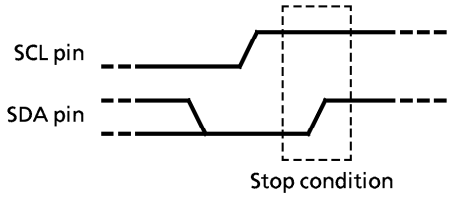


Figure 2-39. Stop Condition Generation

Note : When a stop condition is generated, a time to rise the SCL line should not exceed tr = 2^n/fc – 3.5×4/fc (s). (n depends on the SCK) If the rising time of the SCL line exceeds the above value, there is a probability that a stop condition is not started normally.

SCK	n	tr (Max., fc = 8 MHz)	tr (Max., fc = 4 MHz)
000	4	0.25 us	0.50 us
001	5	2.25 us	4.50 us
010	6	6.25 us	12.5 us
011	7	14.2 us	28.5 us
100	8	30.2 us	60.5 us
101	9	62.5 us	124.5 us
110	10	126.25 us	252.5 us

fc ; High frequency [Hz]

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 cancel**

In the master mode, a serial bus interface interrupt request (INTSBI) occurs after the number of clocks which is specified by the BC and ACK has been transmitted.

In the slave mode, when the received slave address is the same as the value set at the I2CAR, after outputting the acknowledge signal when a GENERAL CALL is received, or when data transfer is complete after matching the slave addresses or receiving a GENERAL CALL, an INTSBI interrupt request occurs.

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

Either writing / reading data to / 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 SBICR2) can be set to "1" by the program, the PIN is not set to "0" when "0" is written.

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

The SBIM (bits 3 , 2 in SBICR2) is used to specify the serial bus interface operation mode. Set the SBIM to "10" after confirming that the serial bus interface pin is set to high level when used in the I^2C bus mode.

Switch a mode to port after making sure that a bus is free.

(10) **Arbitration lost detection monitor**

Since more than one master device can exist simultaneously on a bus in the I^2C bus mode, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the I^2C bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on the bus. Master 1 and Master 2 output the same data until point "a". After Master 1 outputs "1" and Master 2, "0", the SDA line of the bus is wired AND and the SDA line is pulled down to the low level by Master 2. When the SCL line of the bus is pulled up at point "b", the slave device reads data on the SDA line, that is, data in Master 2. Data transmitted from Master 1 becomes invalid. The state in Master 1 is called "arbitration lost". A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

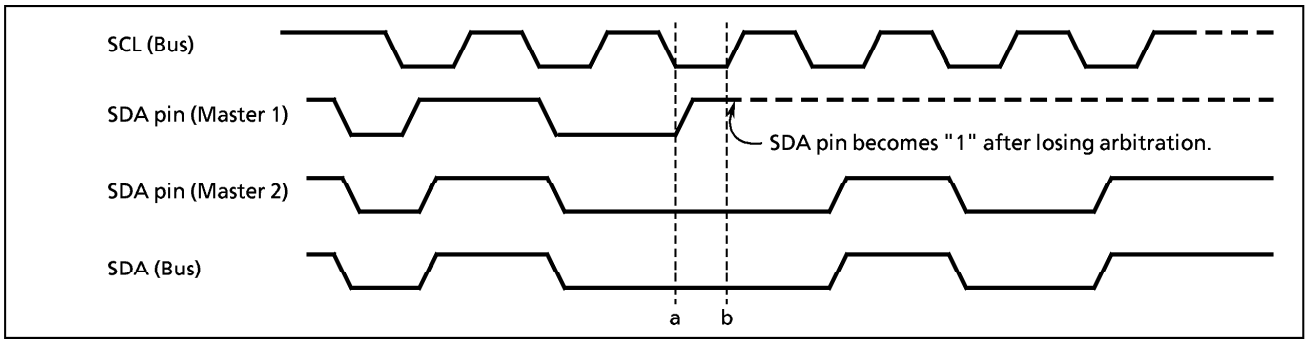


Figure 2-40. Arbitration Lost

The 87CH48 compares levels of the SDA line of the bus with those of the 87CH48 SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and the AL (bit 3 in SBISR) is set to "1".

When the AL is set to "1", the MST and TRX are reset to "0" and the mode is switched to a slave receiver mode.

The AL is reset to "0" by writing/reading data to / from the SBIDBR or writing data to the SBICR2.

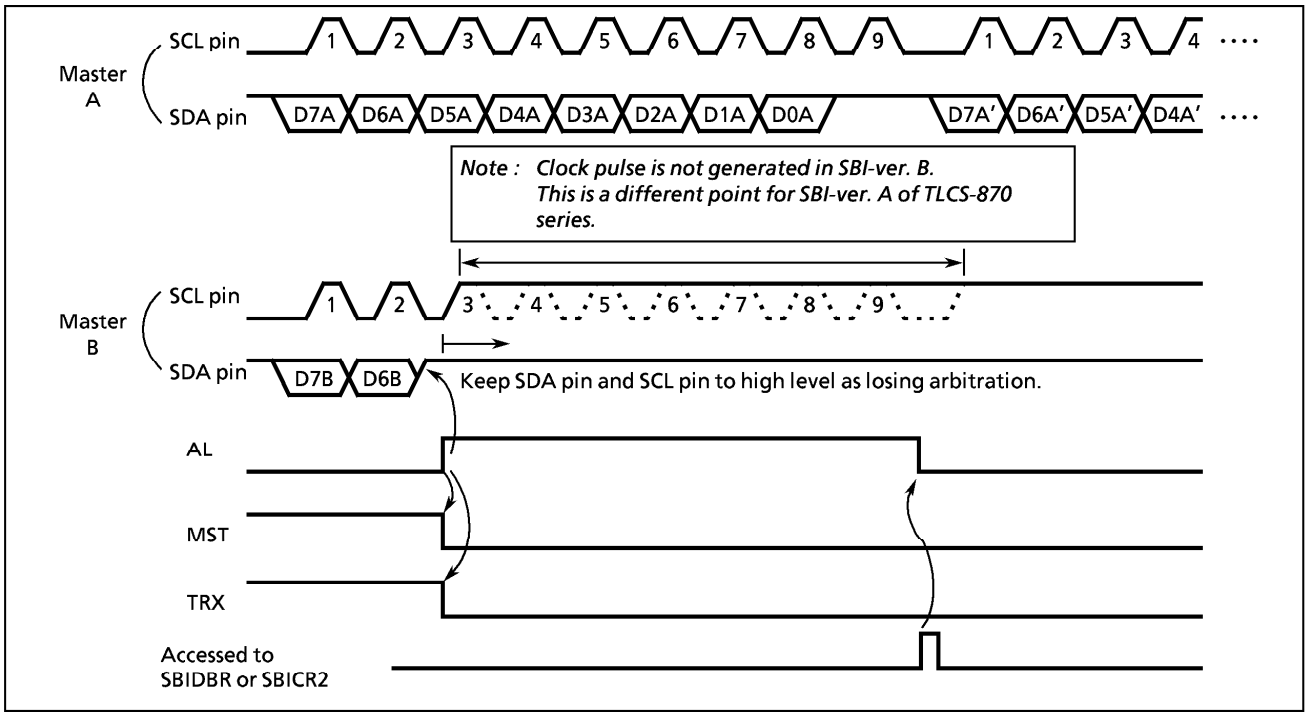


Figure 2-41. Example of Arbitration Lost of when 87CH48 is a Master Device B

(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 after writing/reading data to/from a data buffer register.

(12) **GENERAL CALL detection monitor**

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

(13) **Last received bit monitor**

The SDA value stored at the rising edge of the SCL line is set to the LRB (bit 0 in SBISR). When the contents of the LRB are read immediately after an INTSBI interrupt request is generated in the acknowledge mode, and ACK signal is read.

2.9.5 Data Transfer in I^2C bus Mode

(1) **Device Initialization**

First, set the ACK in the SBICR1 to "1", the BC to "000", and the data length to 8-bit to count a clock pulse for the acknowledge signal. In addition, set the transmit frequency to the SCK.
Next, set the slave address to the SA in the I2CAR. Clear the ALS to "0" to set the addressing format.
After confirming that the serial bus interface pin is high level, for specifying the default setting to a slave receiver mode, clear "0" to the MST, TRX, and BB in the SBICR2; "1" to the PIN; "10" to the SBIM; and "0" to bits 1 and 0.

Note : To initialize the serial bus interface circuit, a constant period that the start conditions are not generated for any device is required after all devices which are connected to the bus are initialized. Then, the initialization must be completed during the period. If not, other devices may start transmitting data before the serial bus interface circuit has been initialized. Thus, data can not be normally received.

(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 low 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 : The slave address to be output to the SBIDBR must be set after the bus free is detected by software. If setting of the slave address is executed before detection bus free, the current output data may be corrupted.
Note 2 : The bus free must be confirmed by software within 98.0 μs (the shortest transmitting time according to the I2C bus standard) after setting of the slave address to be output. Only when the bus free is confirmed, set "1" to the MST, TRX, BB, and PIN to generate the start conditions. If the start conditions are generated without writing "1" to them, transferring may be executed by other masters between the time when the slave address to be output to the SBIDBR is written and the time when "1" is written to the MST, TRX, BB, and PIN in the SBICR2. Thus, the slave address may be corrupted.

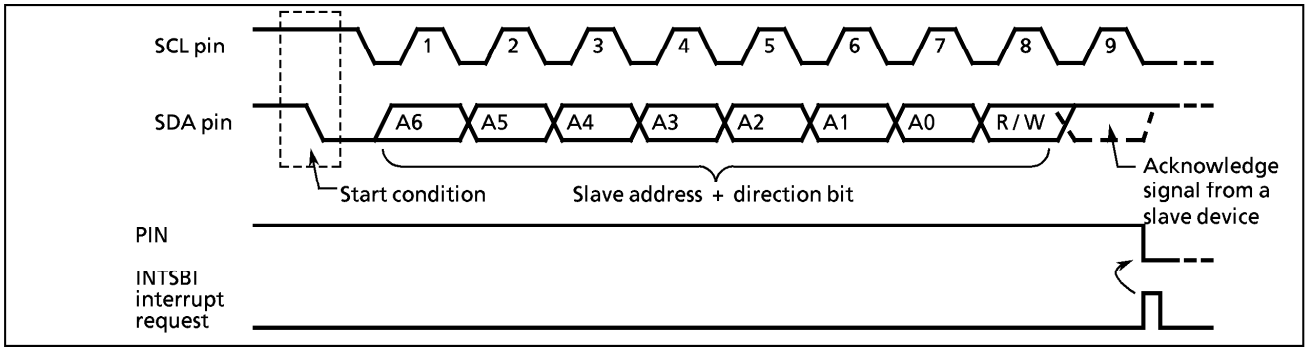


Figure 2-42. Start Condition Generation and Slave Address Transfer

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

Check the MST by the INTSBI interrupt process after a 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" (Transmitter mode)

Check the LRB. When the LRB is "1", a receiver does not request data. Implement the process to generate a stop condition (Refer to 2.9.5. (4)) 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 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, an INTSBI interrupt request occurs. The PIN becomes "0" and the SCL pin is pulled down to the low-level. If the data to be transferred is more than one word in length, repeat the procedure from the LRB checking above.

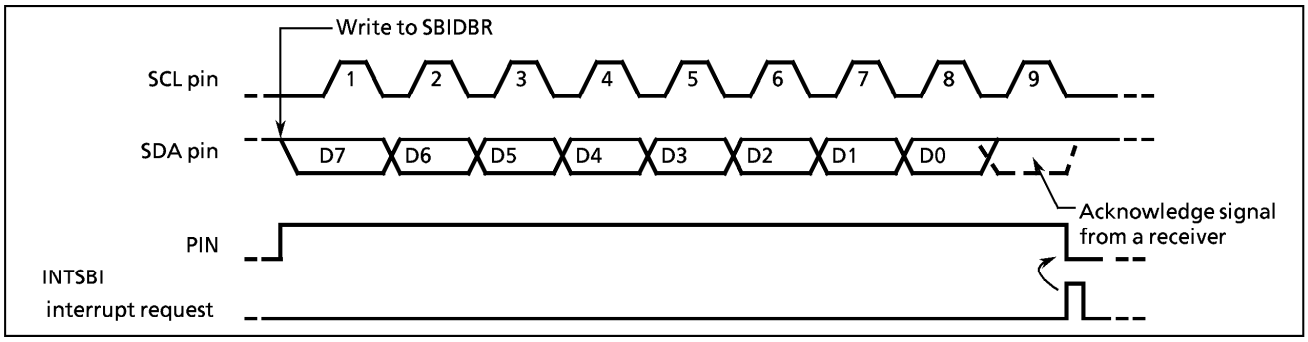


Figure 2-43. Example when BC = "000", ACK = "1" in Transmitter Mode

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 87CH48 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 then occurs and the PIN becomes "0". Then the 87CH48 pulls down the SCL pin to the low level. The 87CH48 outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBIDBR.

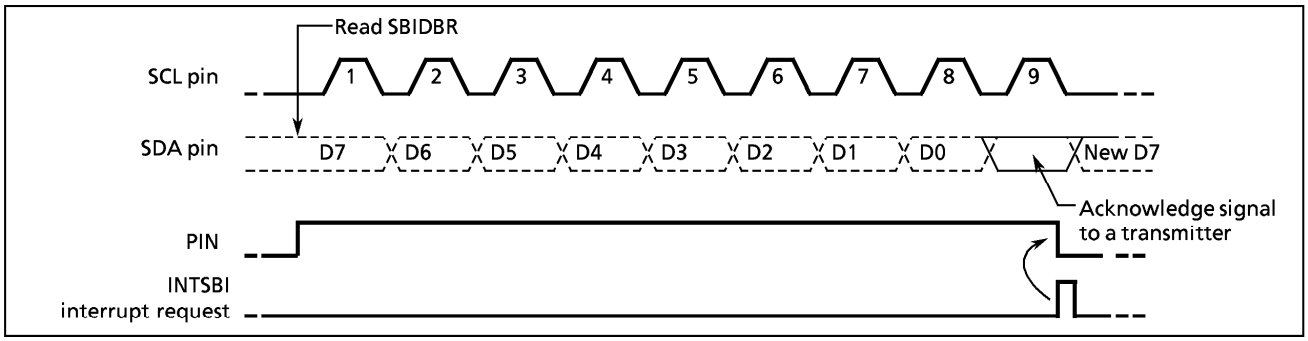


Figure 2-44. Example of when BC = "000", ACK = "1" in Receiver Mode

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 87CH48 generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA line of the bus keeps the high level. The transmitter receives the high-level signal as an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, the 87CH48 generates a stop condition (Refer to 2.9.5 (4)) and terminates data transfer.

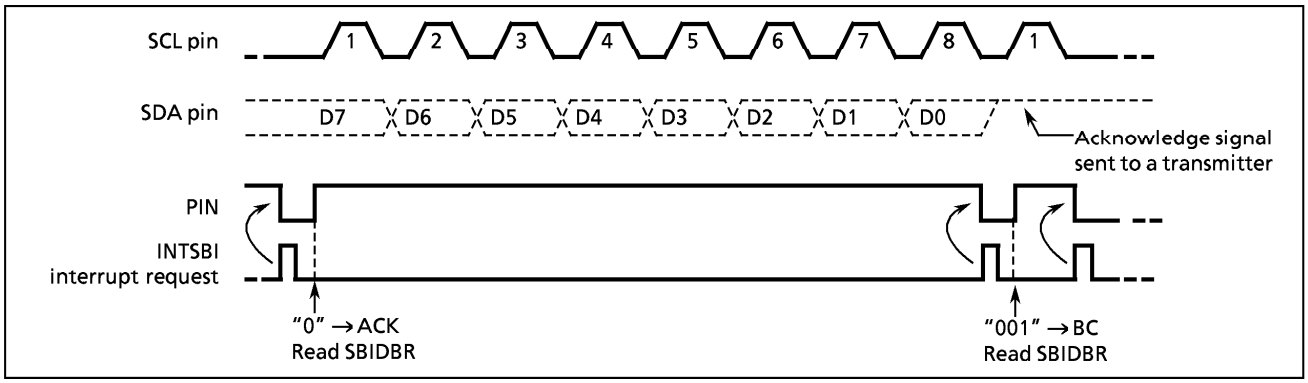


Figure 2-45. Termination of Data Transfer in Master Receiver Mode

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

In the slave mode, the 87CH48 operates either in normal slave mode or in slave mode after losing arbitration.

In the slave mode, an INTSBI interrupt request occurs when the 87CH48 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 87CH48 operates in a slave mode if it is losing arbitration. An INTSBI interrupt request occurs when word data transfer terminates after losing arbitration. When an INTSBI interrupt request occurs, the PIN (bit 4 in the SBICR2) is reset, and the SCL pin is pulled down to the low level. Either reading/writing from/to the SBIDBR or setting the PIN to "1" releases the SCL pin after taking t_{LOW} time.

Check 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 2-12. Operation in the Slave Mode

TRX	AL	AAS	AD0	Conditions	Process
1	1	1	0	The 87CH48 loses arbitration when transmitting a slave address and receives a slave address of which the value of the direction bit sent from another master is "1".	Set the number of bits in 1 word to the BC and write transmitted data to the SBIDBR.
	0	1	0	In the slave receiver mode, the 87CH48 receives a slave address of which the value of the direction bit sent from the master is "1".	
		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	1	1/0	The 87CH48 loses arbitration when transmitting a slave address and receives a slave address or GENERAL CALL of which the value of the direction bit sent from another master is "0".	Read the SBIDBR for setting the PIN to "1" (reading dummy data) or write "1" to the PIN.
		0	0	The 87CH48 loses arbitration when transmitting a slave address or data and terminates transferring word data.	
	0	1	1/0	In the slave receiver mode, the 87CH48 receives a slave address or GENERAL CALL of which the value of the direction bit sent from the master is "0".	
		0	1/0	In the slave receiver mode, the 87CH48 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 a stop condition is generated and a bus SCL line is set to low level by the other devices, a stop condition is not started normally.

Write "1" to the MST, TRX, and PIN, and "0" to the BB to generate a stop condition after releasing the SCL line.

Note : When a stop condition is generated, a time to rise the SCL line should not exceed tr = 2^n/fc – 3.5 × 4/fc (2). (n depends on the SCK.)
If the rising time of the SCL line exceeds the above value, there is a probability that a stop condition is not started normally.

SCK	n	tr (Max., fc = 8 MHz)	tr (Max., fc = 4 MHz)
000	4	0.25 us	0.50 us
001	5	2.25 us	4.50 us
010	6	6.25 us	12.5 us
011	7	14.2 us	28.5 us
100	8	30.2 us	60.5 us
101	9	62.5 us	124.5 us
110	10	126.25 us	252.5 us

fc ; High frequency [Hz]

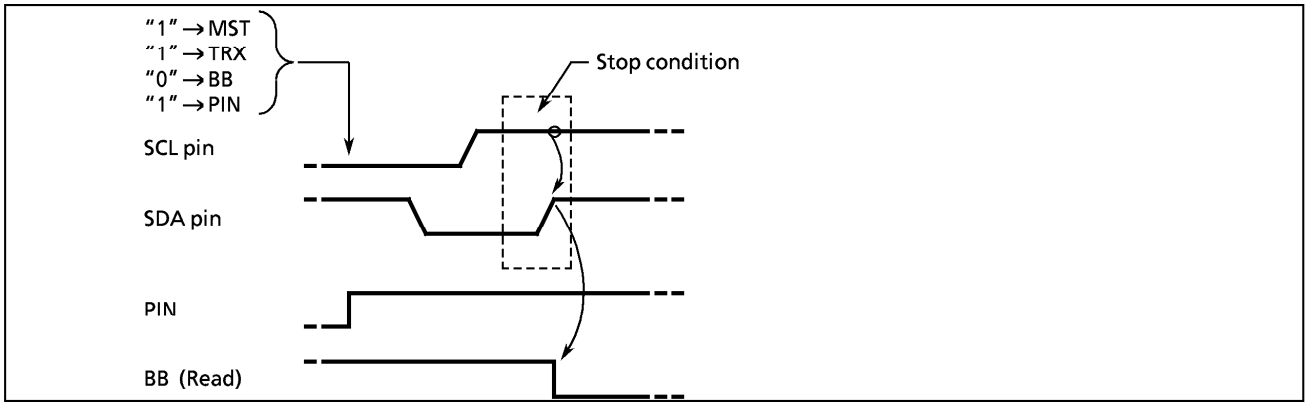


Figure 2-46. Stop Condition Generation

(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 when the 87CH48 is in the master mode.

Clear "0" to the MST, TRX, and BB and set "1" to the PIN and release the bus. The SDA pin retains the high level and the SCL pin is released. Since a stop condition is not generated on a bus, a bus is assumed to be in a busy state from other devices. Check the BB until it becomes "0" to check that the SCL pin of the 87CH48 is released. Check the LRB until it becomes "1" to check that the SCL line of a bus is not pulled down to the low level by other devices. After confirming that a bus stays in a free state, generate a start condition with procedure 2.9.5. (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.

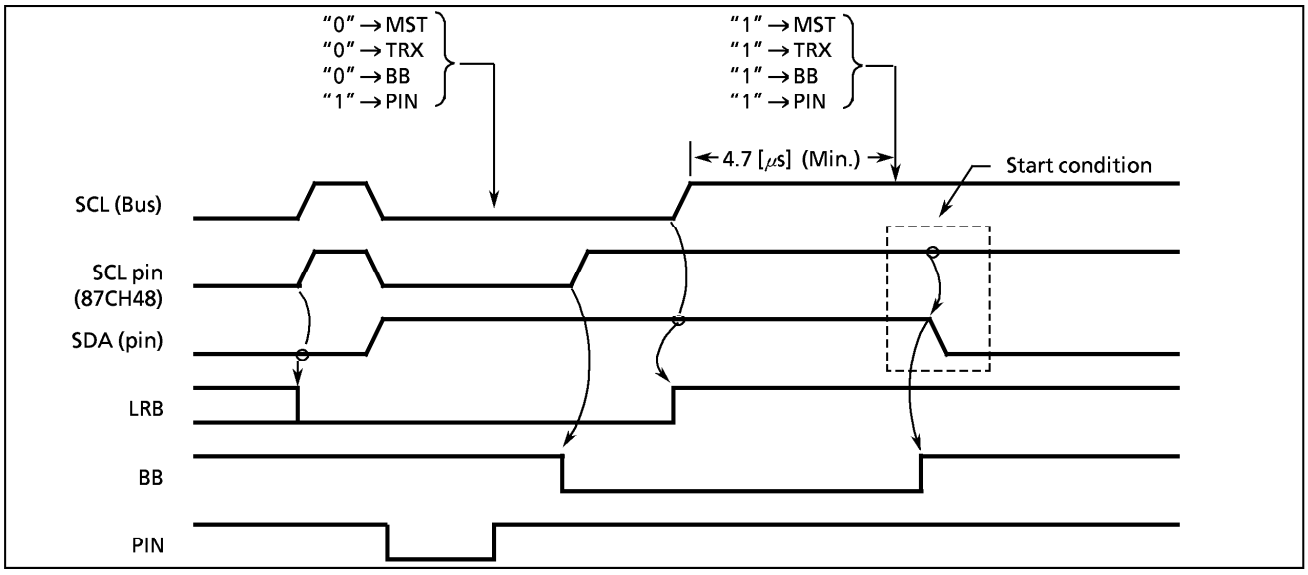


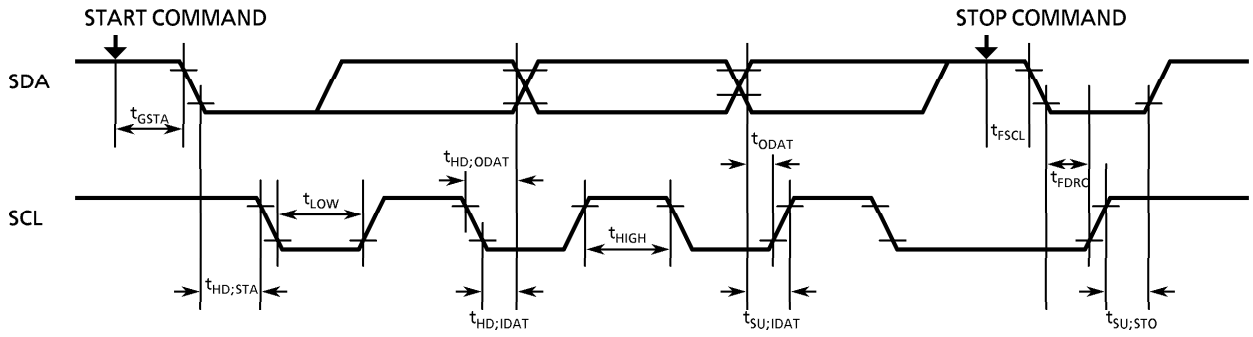
Figure 2-47. Timing Diagram when Restarting the 87CH48

AC Timing for SBI-Ver. B (I2C-BUS)

Parameter	Symbol	Min.	Typ.	Max	Unit
Hold time (repeated) START condition. After this period, the first clock pulse is generated.	$t_{HD;STA}$	2^n/fc	–		s
HIGH period of the SCL clock	t_{HIGH}	2^n/fc + 8/fc	–	–	s
LOW period of the SCL clock	t_{LOW}	2^n/fc	–	–	s
Data hold time (input)	$t_{HD;IDAT}$	0	–	–	ns
Data set-up time (input)	$t_{SU;IDAT}$	250	–	–	ns
Data hold time (output)	$t_{HD;ODAT}$	3/fc	–	7/fc	s
Data output time before rising SCL clock.	t_{ODAT}	–	2^n/fc – $t_{HD;ODAT}$	–	s
Set-up time for STOP condition	$t_{SU;STO}$	2^n/fc + 4/fc	–	–	s
The period of generating a start condition when writing START command.	t_{GSTA}	3/fc	–	–	s
The period of falling SCL clock when writing STOP command.	t_{FSCL}	3/fc	–	–	s
The period between falling edge of SDA and rising edge of SCL when generation a STOP condition.	t_{FDRC}	2^n/fc	–	–	s

Note n:

SCK (bit 2 to 0 in the SBICR1)	n
000	4
001	5
010	6
011	7
100	8
101	9
110	10



2.9.6 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) 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		"0"	SCK		

(Initial value 0000 *000)

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) 111 : External clock (input from $\overline{SCK}$ pin) } at fc = 8 MHz (Output on $\overline{SCK}$ pin)	

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

Serial Bus Interface Data Buffer Register

SBIDBR (0021_H)

7	6	5	4	3	2	1	0

(Initial value 0000 0000) Read / Write

Note : 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.

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.

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 (After SIOS is cleared to "0", SIOF is cleared to "0" at the termination of transfer or setting of SIOINH.)	Read only
SEF	Shift operating status monitor	0 : Shift operation terminated 1 : Shift operation in process	

Figure 2-48.Serial Bus Interface Control Register 1/Serial Bus Interface Data Buffer Register/Serial Bus Interface Control Register 2/Serial Bus Interface Status Register in SIO mode

(1) Serial Clock

a. Clock source

The SCK (bits 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 $\overline{SCK}$ pin. The $\overline{SCK}$ pin becomes a high 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.

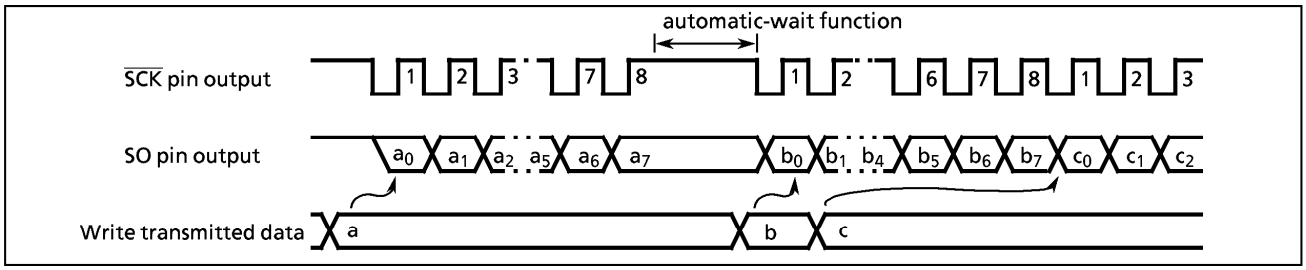


Figure 2-49. Automatic-wait Function

② External clock (SCK = "111")

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

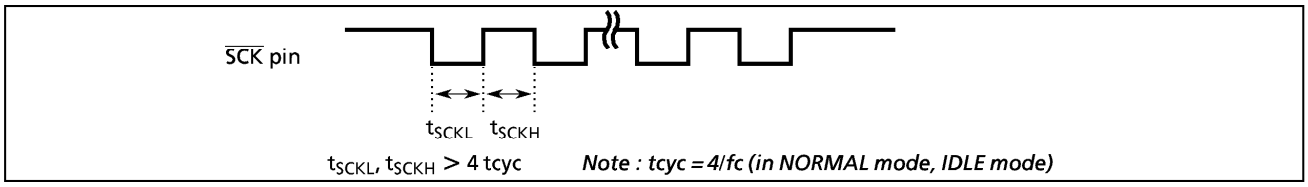


Figure 2-50. Maximum Data Transfer Frequency When External Clock Input

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 ofthe $\overline{SCK}$ pin input/output).

② Trailing edge shift
Data is shifted on the trailing edge of the serial clock (at a rising edge ofthe $\overline{SCK}$ pin input/output).

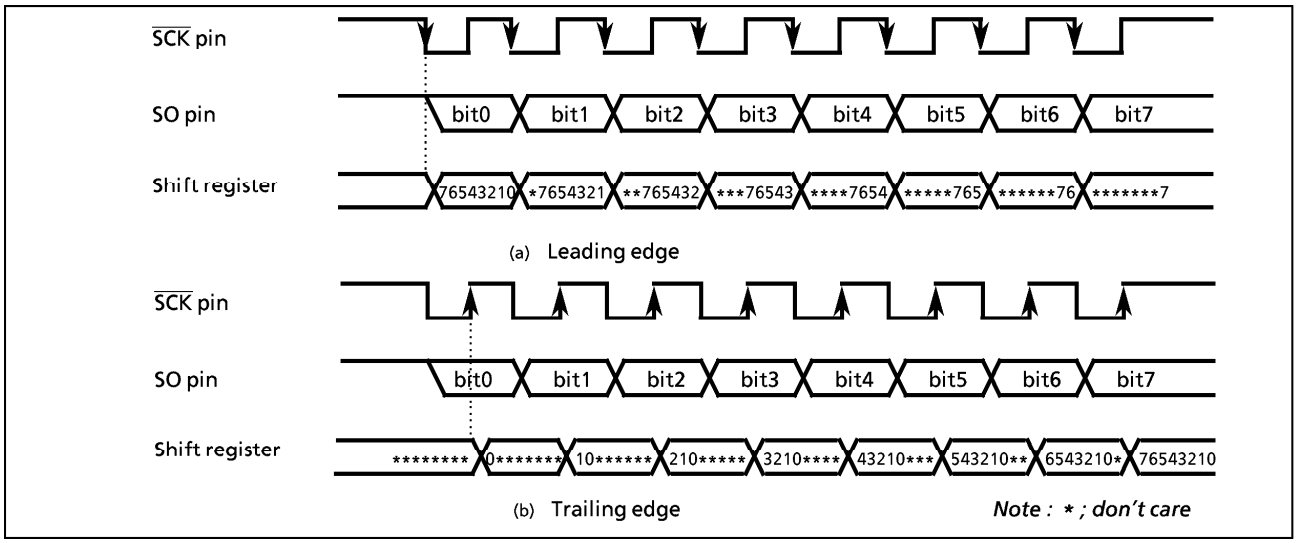


Figure 2-51. Shift Edge

(2) Transfer mode

The SIOM (bits 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 $\overline{SCK}$.

The transmission is ended by clearing SIOS to "0" or setting SIOINH to "1" in buffer empty interrupt service program. 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.

When SIOINH is set, the transmission is immediately ended and SIOF is cleared to "0".
If it is necessary to change the number of words, SIOS should be cleared to "0", then BUF must be rewritten after confirming that SIOF has been cleared to "0".

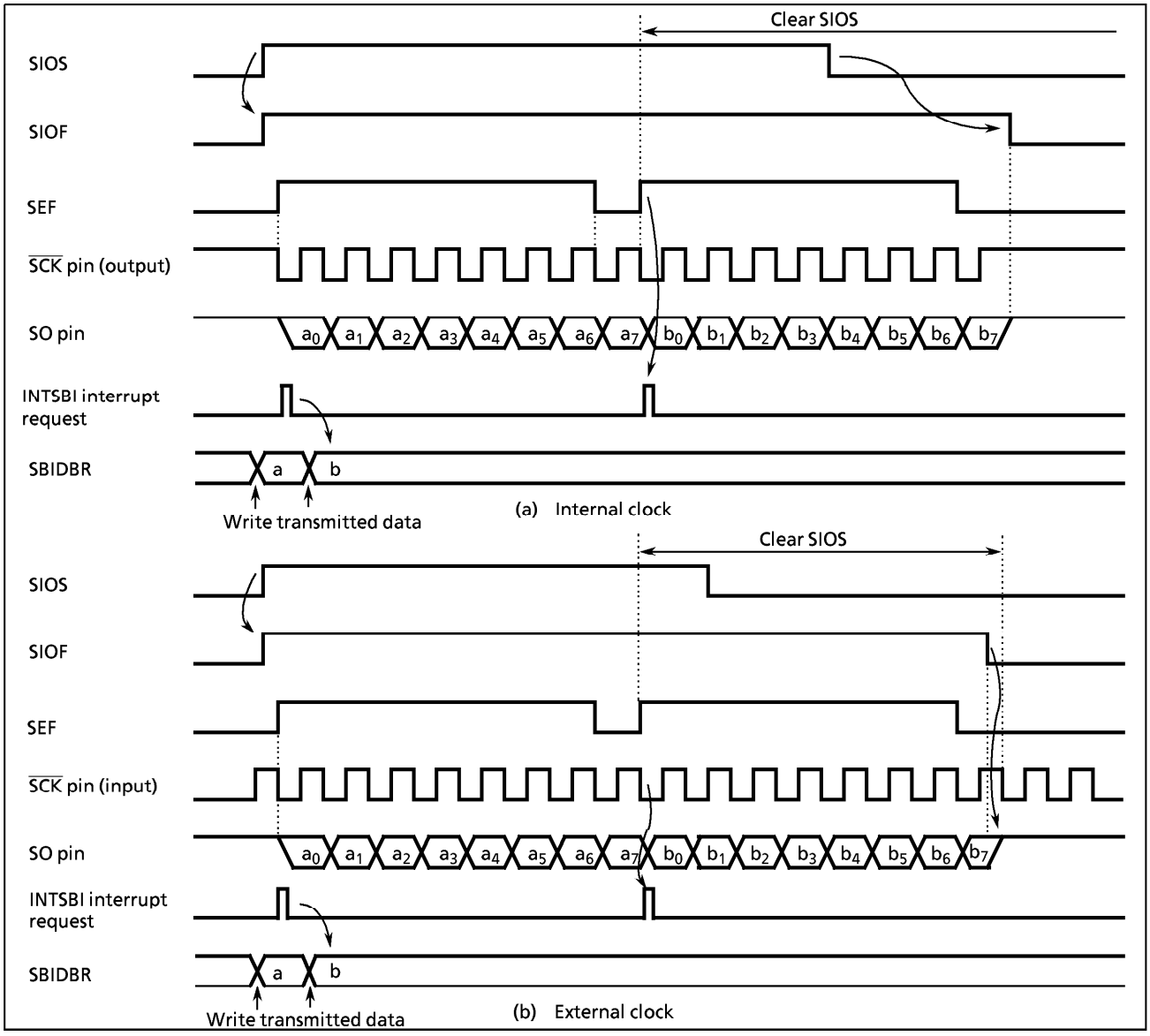


Figure 2-52. Transfer Mode

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  (P4) . 0              ; If SCK = 0 then loop
         JRS   T , STEST2
         LD    (SBICR1) , 00000111B  ; SIOS ← 0
```

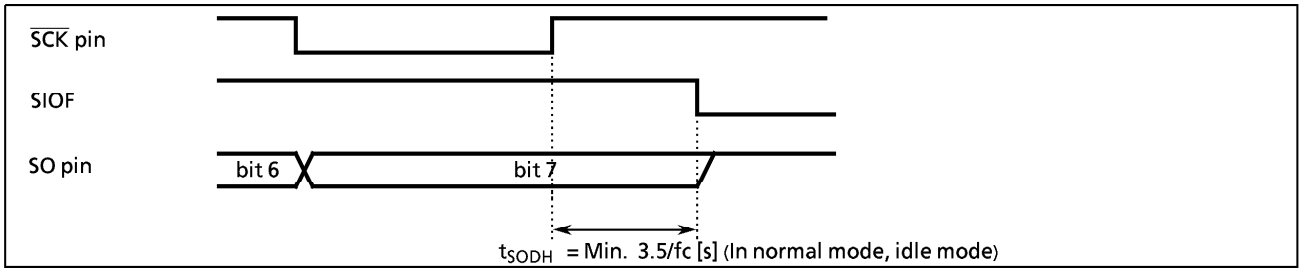


Figure 2-53. Transmitted Data Hold Time at End of Transmit

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.

The receiving is ended by clearing SIOS to "0" or setting SIOINH to "1" in buffer full interrupt service program. When SIOINH is set, the receiving is immediately ended and SIOF is cleared to "0". 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).

If it is necessary to change the number of words in external clock operation, SIOS should be cleared to "0" then BUF must be rewritten after confirming that SIOF has been cleared to "0".

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of data receive, BUF must be rewritten before the received data is read out.

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.

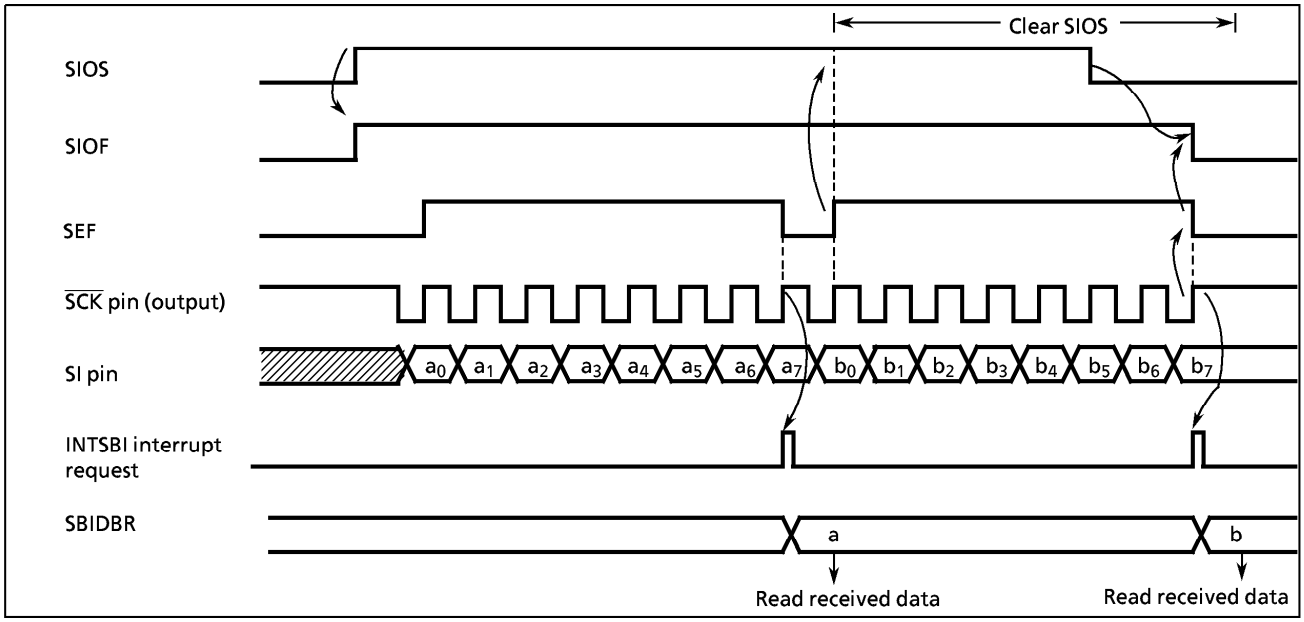


Figure 2-54. Receive Mode (Example : Internal clock)

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 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 $\overline{SCK}$.

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.

The transmission is ended by clearing SIOS to "0" or setting SIOINH to "1" in buffer empty interrupt service program. 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 SIOINH is set, the transmit/receive operation is immediately ended and SIOF is cleared to "0".

If it is necessary to change the number of words in external clock operation, SIOS should be cleared to "0", then BUF must be rewritten after confirming that SIOF has been cleared to "0".

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of transmit / receive operation, BUF must be rewritten before reading and writing of the receive/transmit data.

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.

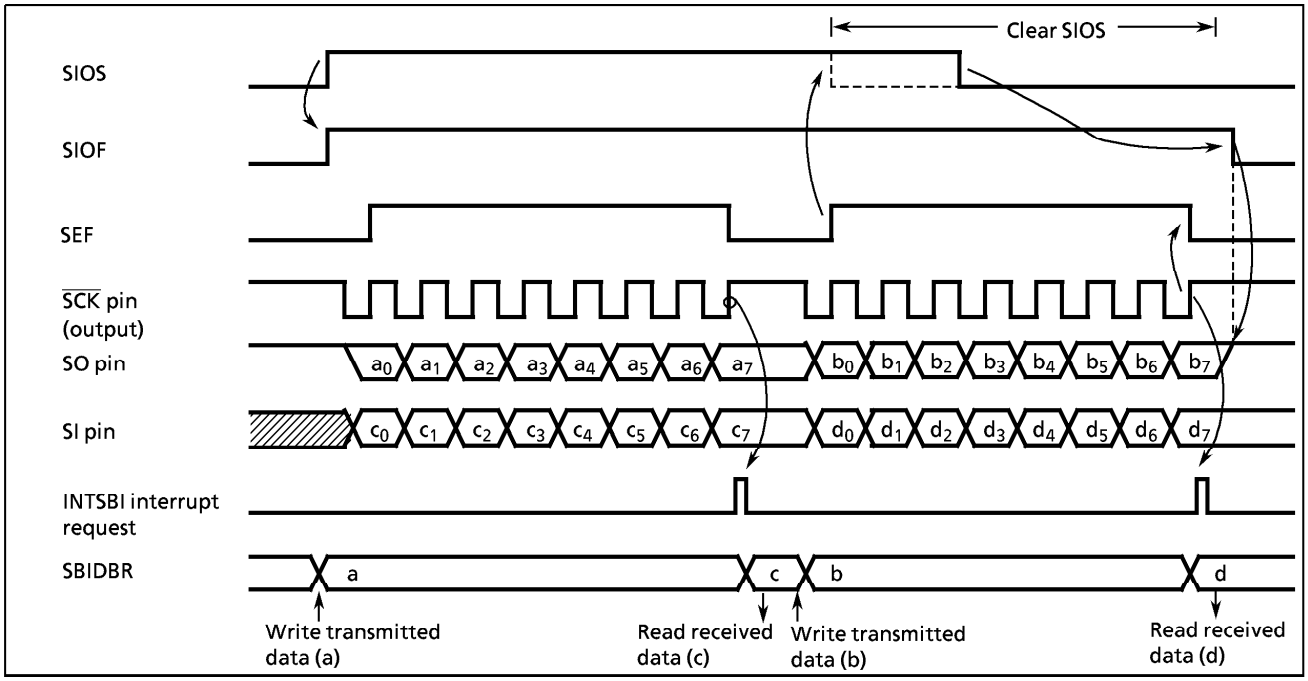


Figure 2-55. Transmit/Receive Mode (Example : Internal clock)

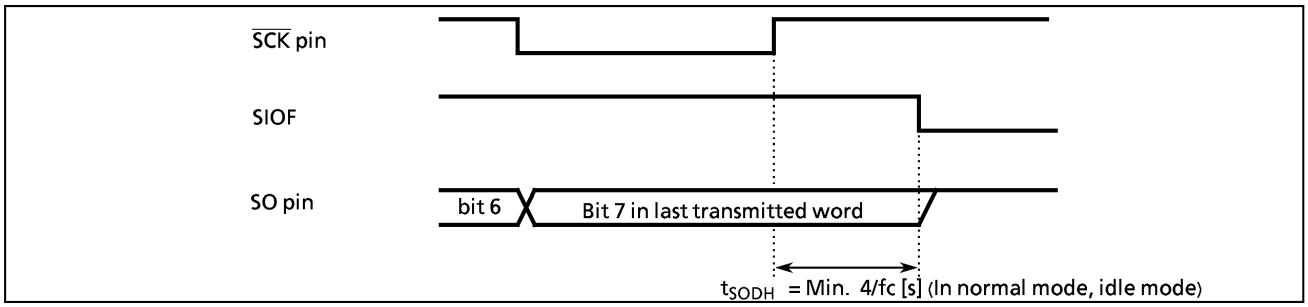


Figure 2-56. Transmitted Data Hold Time at End of Transmit/Receive

2.10 UART (Asynchronous serial interface)

The 87CH48 has 1 channel of UART (asynchronous serial interface) .

The UART is connected to external devices via RxD and TxD. RxD is also used as P44 ; TxD, as P45. To use P44 or P45 as the RxD or TxD pin, set P4 port output latches to 1.

2.10.1 Configuration

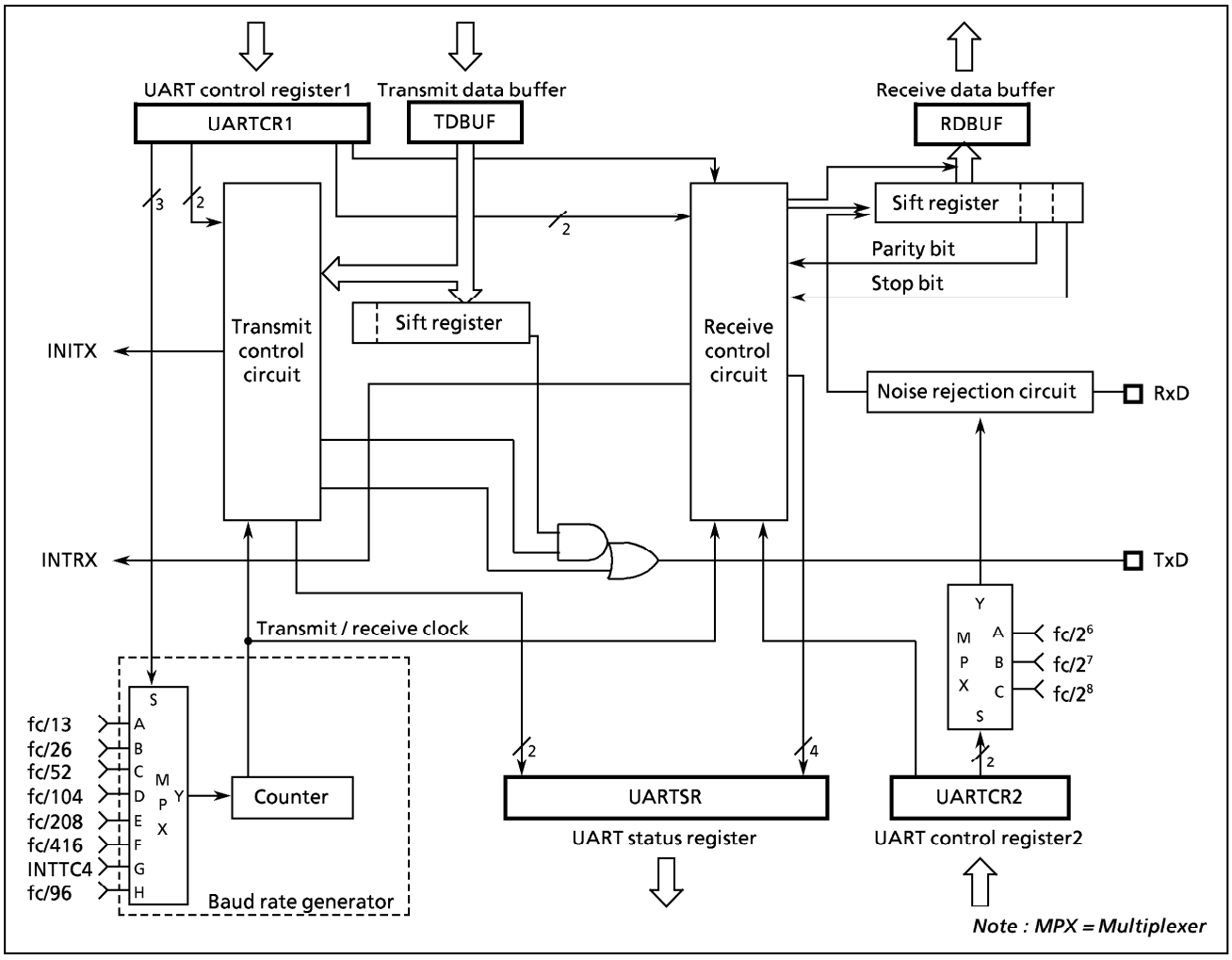


Figure 2-57. UART

2.10.2 Control

UART is controlled by the UART control registers (UARTCR1, UARTCR2). The operating status can be monitored using the UART status register (UARTSR).

UART control register

UARTCR1 ($002A_H$)

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

(Initial value 0000 0000)

BRG	Transmit clock select	000 : fc / 13 [Hz] 001 : fc / 26 010 : fc / 52 011 : fc / 104 100 : fc / 208 101 : fc / 416 110 : INTTC4 (Interrupt from TC4) 111 : f c / 96	write only
PE	Parity addition	0 : No parity 1 : Parity	
EVEN	Even-numbered parity	0 : Odd-numbered parity 1 : Even-numbered parity	
STBT	Transmit stop bit length	0 : 1 bit 1 : 2 bit	
RXE	Receive operation	0 : Disable 1 : Enable	
TXE	Transfer operation	0 : Disable 1 : Enable	

Note 1 : When operations are disabled by setting TXE and RXE bit to "0", the setting becomes valid when data transmit or receive complete. When the transmit data is stored in the transmit data buffer, the data are not transmitted. Even if data transmit is enabled, until new data are written to the transmit data buffer, the current data are not transmitted.

Note 2 : The transmit clock and the parity are common to transmit and receive.

UARTCR2 ($002B_H$)

7	6	5	4	3	2	1	0
INTS					RXDNC		STOPBR

(Initial value 0*** *000)

STOPBR	Receive stop bit length	0 : 1 bit 1 : 2 bit	write only
RxDNC	Selection of RxD input noise rejection time	00 : No noise rejection (hysteresis input) 01 : Rejects pulses shorter than 31 / fc [s] as noise 10 : Rejects pulses shorter than 63 / fc [s] as noise 11 : Rejects pulses shorter than 127 / fc [s] as noise	
INTS	IL12 selection	0 : INTRX (UART receive interrupt) 1 : INTTC4 (8-bit TC4 interrupt)	

Note 3 : When RxDNC = 01, pulses longer than 96/fc [s] are always regarded as signals ; when RxDNC = 10, longer than 192/fc [s] ; and when RxDNC = 11, longer than 384/fc [s]

Figure 2-58. UART Control Register

UARTSR ($002A_H$)

7	6	5	4	3	2	1	0
PERR	FERR	OERR	RBFL	TEND	TBEP		

(Initial value 0000 11**)

TBEP	Transmit data buffer empty flag	0 : – 1 : Transmit data buffer empty	read only
TEND	Transmit end flag	0 : Transmitting 1 : Transmit end	
RBFL	Receive data buffer full flag	0 : – 1 : Receive data buffer full	
OERR	Overrun error flag	0 : No overrun error 1 : Overrun error	
FERR	Framing error flag	0 : No framing error 1 : Framing error	
PERR	Parity error flag	0 : No parity error 1 : Parity error	

UART receive data buffer (Initial value 0000 0000) read only

RDBUF (0029_H)

7	6	5	4	3	2	1	0

UART transmit data buffer (Initial value 0000 0000) write only

TDBUF (0029_H)

7	6	5	4	3	2	1	0

Figure 2-59. UART Status Register and Data Buffer Registers

2.10.3 Transfer Data Format

In UART, a one-bit start bit (low level) , stop bit (bit length selectable at high level, by STBT), and parity (select parity in PE ; even-or odd-numbered parity by EVEN) are added to the transfer data. The transfer data formats are shown as follow.

Table 2-9.

PE	STBT	Frame length: 1 2 3 ------- 8 9 10 11 12
0	0	Start bit0 bit1 ----- bit6 bit7 Stop1
0	1	Start bit0 bit1 ----- bit6 bit7 Stop1 Stop2
1	0	Start bit0 bit1 ----- bit6 bit7 Parity Stop1
1	1	Start bit0 bit1 ----- bit6 bit7 Parity Stop1 Stop2

2.10.4 Transfer Rate

The baud rate of UART is set of BRG (bit 0, 1, and 2 in UARTCR1) . The example of the baud rate shown as follows.

Table 2-10.

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

When TC4 is used as the UART transfer rate (when BRG = 110), the transfer clock and transfer rate are detarmined as follows :

$$\text{Transfer clock} = \frac{\text{TC4 source clock}}{\text{TREG4 set value}}$$

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

2.10.5 Data Sampling

The UART receiver keeps sampling input using the clock selected by BRG (bit 0, 1, and 2 in UARTCR1) until a start bit is detected in RxD pin input. RT clock starts at the falling edge of the RxD pin. Once a start bit is detected, the start bit, data bits, stop bit(s), and parity bit are sampled at three times of RT7, RT8, and RT9 during one receiver clock interval (RT clock). (RT0 is the position where the bit supposedly starts) . Bit is determined according to majority rule (the data are the same twice or more out of three samplings) .

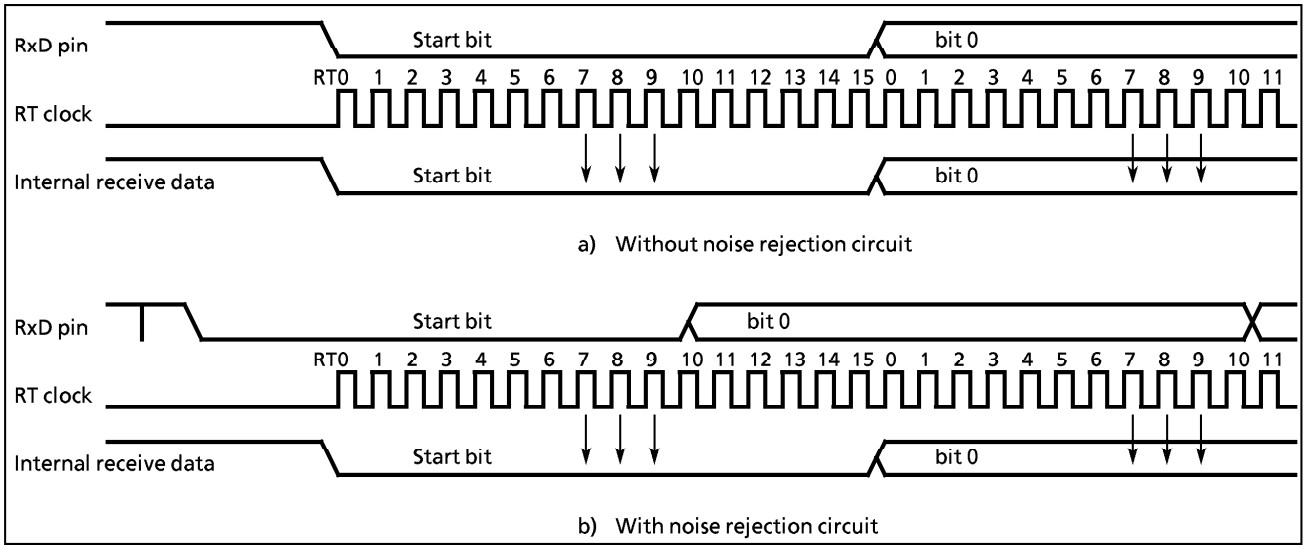


Figure 2-60. Data Samping

2.10.6 STOP Bit Length

Select a transmit stop bit length (1 or 2 bits) by STBT (bit5 in UARTCR1)

2.10.7 Parity

Set parity/no parity by PE ; set parity type (odd-or even-numbered) by EVEN (bit 4 in UARTCR1).

2.10.8 Transmit / Receive

(1) Data transmit

Set TXE (bit 7 in UARTCR1) 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 UARTCR1) and a parity bit if parity addition is specified. Select the data transfer baud rate using bits 0 to 2 in UARTCR1. When data transmit starts, transmit buffer empty flag TBEP is set to 1 and an INTTX interrupt 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

Set RXE (bit 6 in UARTCR1) 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 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 an INTRX interrupt is generated. Select the data transfer baud rate using bits 0 to 2 in UARTCR1. 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.

2.10.9 Status Flag / Interrupt Signal

(1) Parity error

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

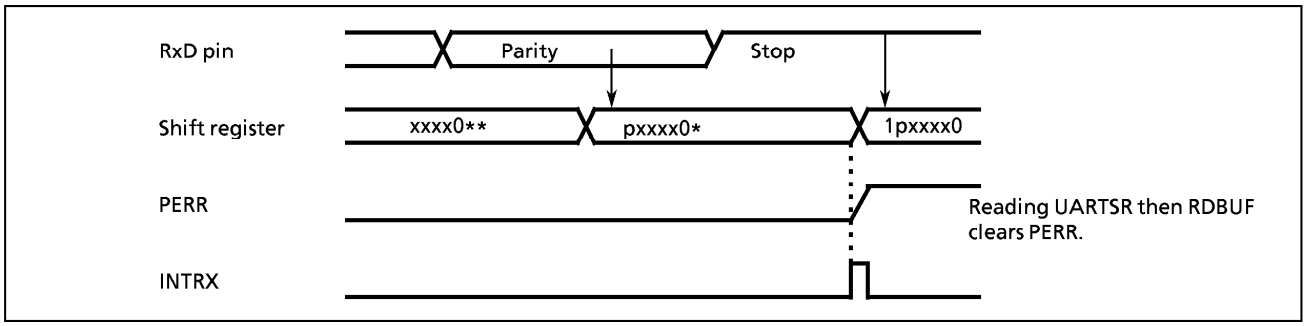


Figure 2-61. Generation of Parity Error

(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.

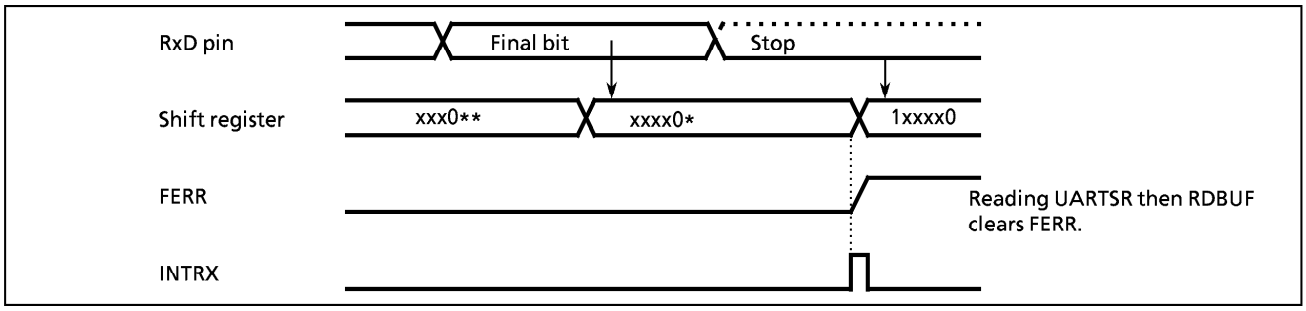


Figure 2-62. Generation of Framing Error

(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. In this case, the receive data is discarded ; data in RDBUF are not affected. Reading UARTSR then RDBUF clears OERR.

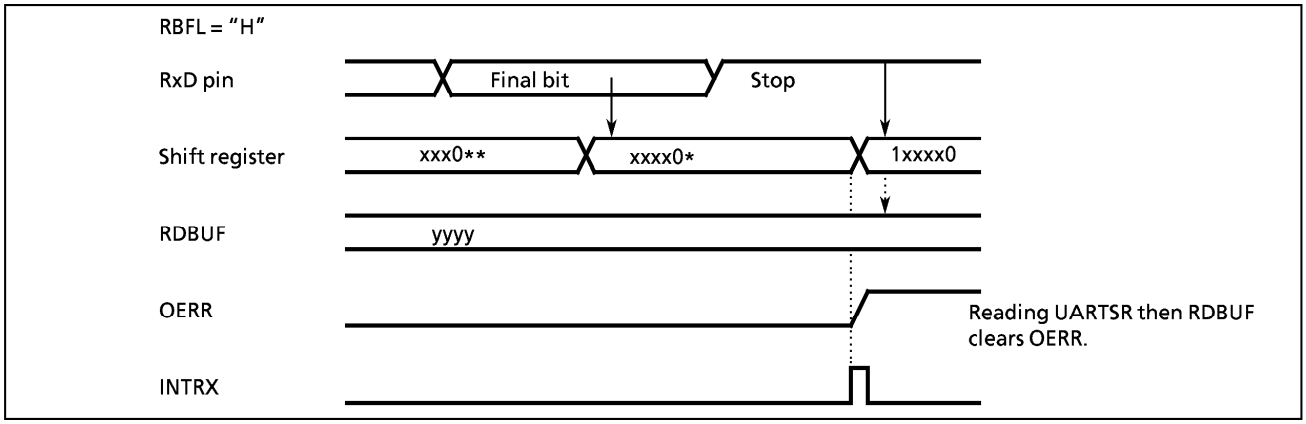


Figure 2-63. Generation of Overrun Error

(4) Receive data buffer full

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

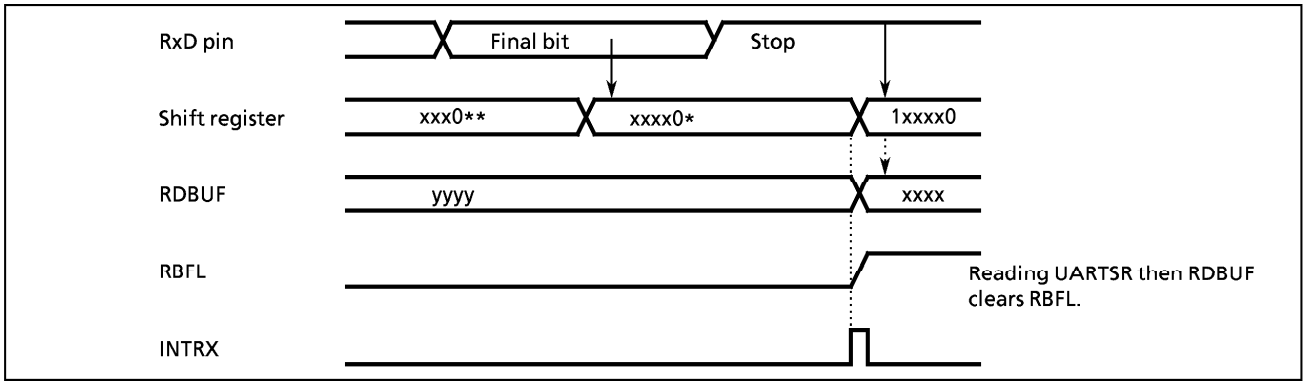


Figure 2-64. Generation of Receive Buffer Full

(5) Transmit data buffer empty

When no data is in the transmit buffer TDBUF, TBEP is set, 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.

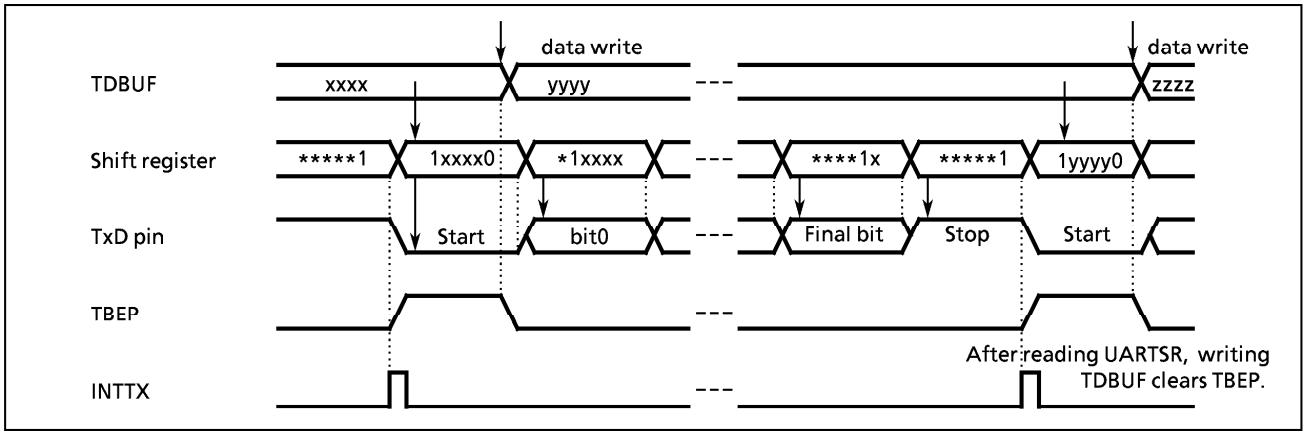


Figure 2-65. Generation of Transmit Buffer Empty

(6) Transmit end flag

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

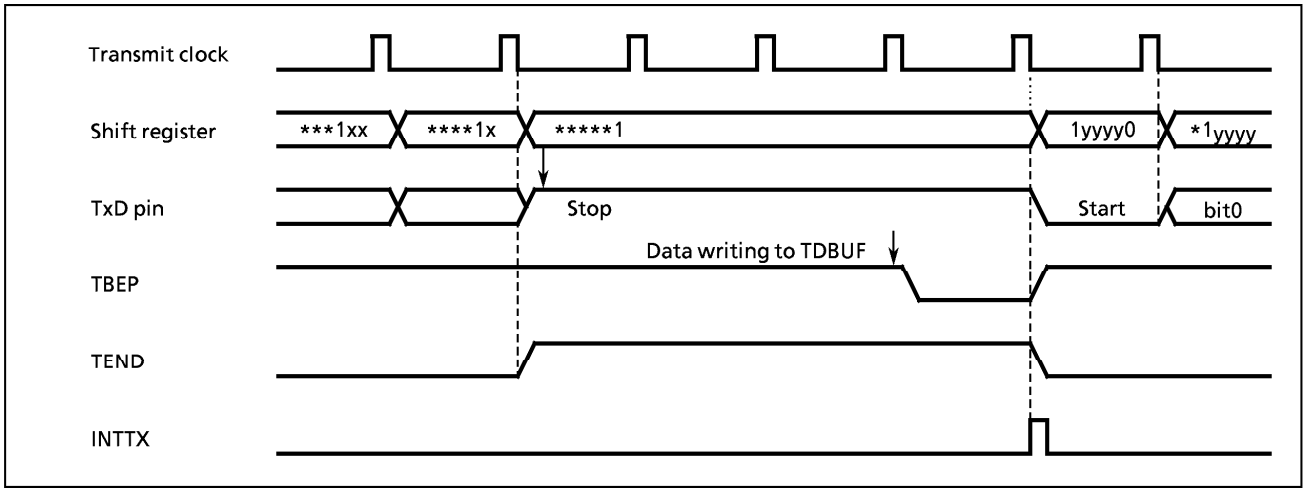


Figure 2-66. Generation of Transmit Buffer Empty

2.11 12-bit PWM (Pulse Width Modulation) Output Circuit

87CH48 has four built-in pulse width modulation (PWM) channels which are 12-bit resolution. D/A converter output can easily be obtained by connecting an external low-pass filter. PWM outputs are output from P80 (PWM0) to P83 (PWM3) with negative logic (low active). When P80 to P83 are used as PWM outputs, the output latch of P8 (#0008_H) must be set to "1", subsequently P8CR (#0009_H) must be set to the output mode. P80 to P83 become Hz during and just after reset. When P80 to P83 are used as PWM output, the pull-up resistor is externally added. PWM outputs can output low level pulse in only NORMAL1 (or IDLE1) and NORMAL2 (IDLE2) modes. When P80 to P83 are used as PWM outputs and the operation mode is transferred from NORMAL1 mode to STOP1 mode or from NORMAL2 mode to STOP2 mode (or SLOW mode, SLEEP mode), the PWM pin always outputs "High". Consequently, the lower level pulse can not be output. In addition, when the operation mode is transferred from STOP1 mode to NORMAL1 mode, or from STOP2 (or SLOW, SLEEP mode) to NORMAL2 mode, the registers (PWMCR, PWMDBR) necessary for PWM output control must be initialized to output the PWM pulse again, which is the same settlings after reset.

2.11.1 Configuration

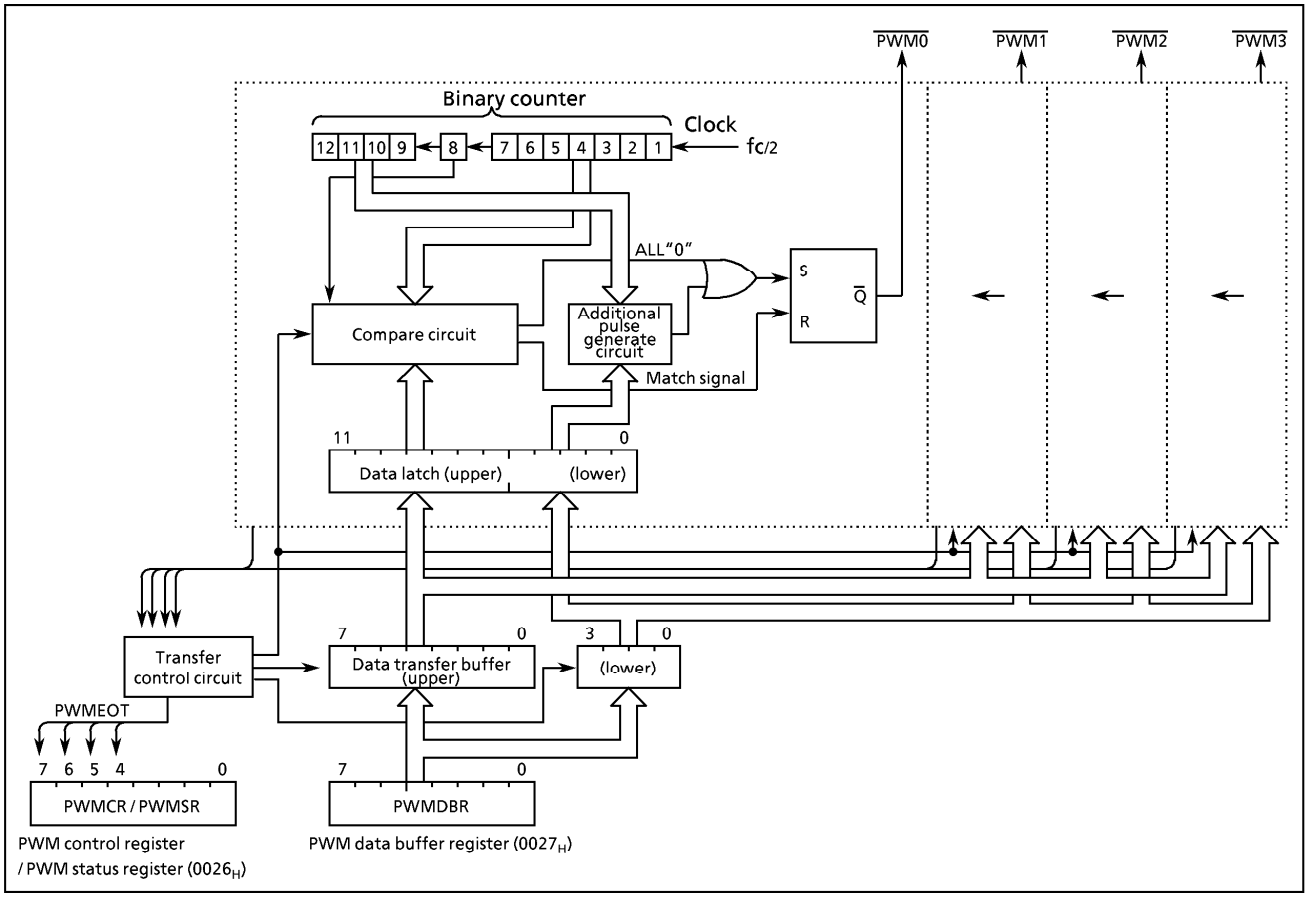


Figure 2-67. Pulse Width Modulation Circuit

2.11.2 Control

PWM output is controlled by the PWM control register (PWMCR) PWM data buffer register (PWMDBR). Whether writing to the PWM data buffer register is enable or not is acknowledged by reading EOT flag of the PWM status register (PWMSR).

(1) Control register

PWM control register

PWMCR (0026_H)

7	6	5	4	3	2	1	0
–	–	–	–	P	W M	D L	C

(Initial value **** 0000)

PWMDLC	Pointer select o the respective PWMDBR channels	0000 : Lower 4-bit for channel 0 (PWMDBR) 0001 : Upper 8-bit for channel 0 (PWMDBR) 0010 : Lower 4-bit for channel 1 (PWMDBR) 0011 : Upper 8-bit for channel 1 (PWMDBR) 0100 : Lower 4-bit for channel 2 (PWMDBR) 0101 : Upper 8-bit of channel 2 (PWMDBR) 0110 : Lower 4-bit of channel 3 (PWMDBR) 0111 : Upper 8-bit of channel 3 (PWMDBR)	Write Only
	Data transfer request command of the respective channels	1000 : Data transfer request command for channel 0 1001 : Data transfer request command for channel 1 1010 : Data transfer request command for channel 2 1011 : Data transfer request command for channel 3	

PWM status register

PWMSR (0026_H)

7	6	5	4	3	2	1	0
EOT3	EOT2	EOT1	EOT0	"1"	"1"	"1"	"1"

(Initial value 0000 ****)

EOT3 EOT2 EOT1 EOT0	End of transfer	0 : Enables writing to PWMDBR 1 : Disables writing to PWMDBR	Read Only

PWM data buffer register

PWMDBR (0027_H)

7	6	5	4	3	2	1	0

Write Only

(Initial value 0000 0000)

Note1 : Writing either 1 or 0 to bit 7-4 of PWMCR dose not affect PWM control.

Note2 : "1" is always read out in bit 3-0 of PWMSR

Note3 : The initial value of the respective registers means the value after reset or cancel of SLOW, SLEEP and STOP.

Figure 2-68 (1). 12-bit PWM Control Register

(2) Control Procedures

Figure 2-68 (2) shows a program control flow chart for the respective channels.

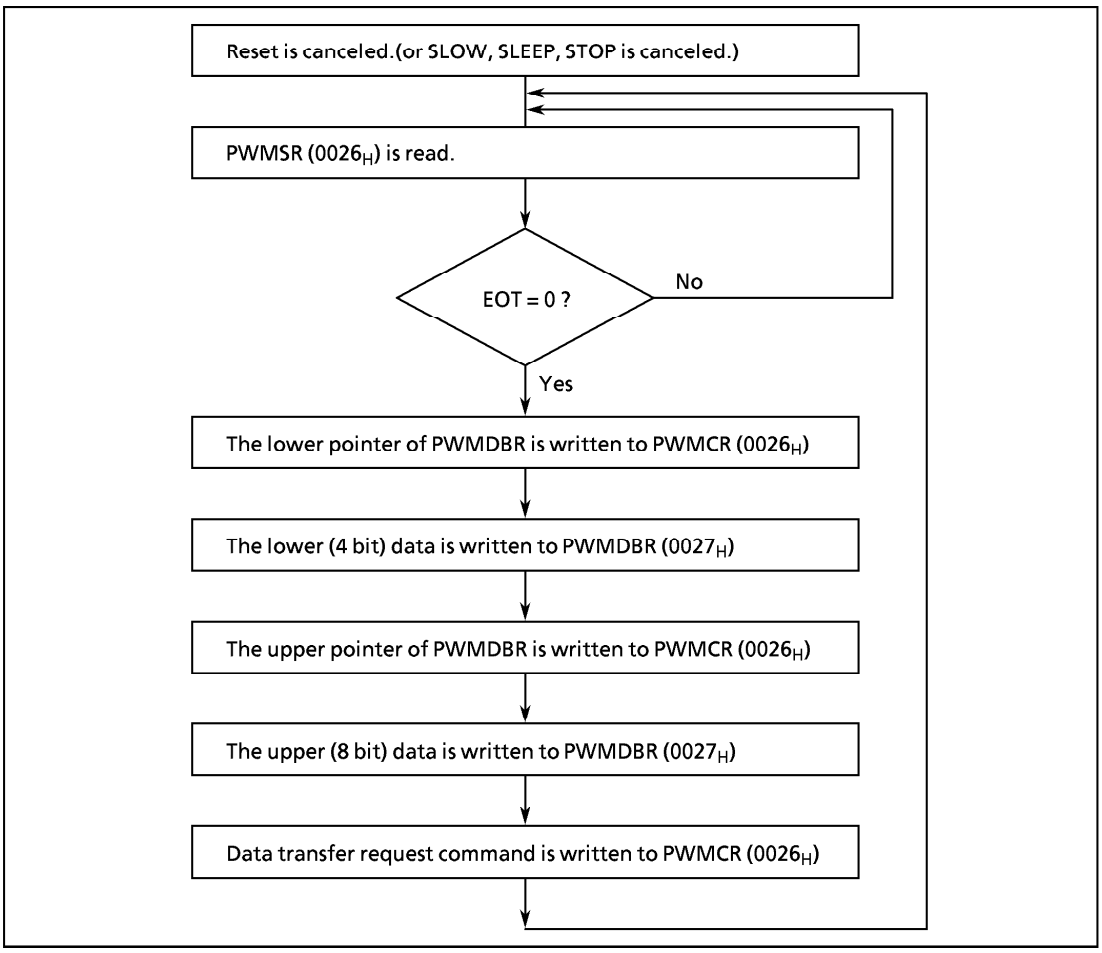


Figure 2-68 (2). Program Control Procedures for channels

Example : Outputs 32 μs(at fc = 8MHz) without additional pulse to pin PWM0.

```
WAIT0   : TEST  (PWMSR),   4        ;  EOT0 = 0?
          JRS   F, WAIT0
          LD    (PWMCR),   00H      ;  Selects the lower of PWMDBR for channel 0
          LD    (PWMDBR),  00H      ;  Writes 00H without additional pulse to the lower 4 bit of
                                       PWMDBR for channel 0
          LD    (PWMCR),   01H      ;  Selects the lower of PWMDBR for channel 0
          LD    (PWMDBR),  80H      ;  Writes 80H ( = 32μs ÷ 2/fc) to the upper 8 bit of PWMDBR
                                       for channel 0.
          LD    (PWMCR),   08H      ;  Writes the data transfer request command for channel 0.
```

2.11.3 Functions

(1) PWMSR (EOT flag) operation

The EOT flag becomes "1" when the upper 8 bit and the lower 4 bit data are transferred from the PWM data buffer (PWMDBR) to the PWM data latch (up to 2^{13}/fc), and "0" when the data transfer from PWMDBR to the PWM data latch is completed.

Writing to PWMDBR (0027_H) must be executed when the EOT flag is "0". If PWMDBR data is rewritten at the EOT flag = "1", the PWM output pulse may be not correctly switched.

(2) The duration time until PWM output starts

In NORMAL mode subsequent to canceling reset or STOP (or SLOW, SLEEP mode), the maximum time form the point at which the data transfer request command is written until the PWM pulse output to the PWM pin is 2^{13}/fc after writing to the lower and the upper of PWMDBR is completed according to the procedures mentioned in figure 2-68 (2).

Additionally, when the output pulse is modulated in the middle of a output, the maximum time from the point at which the data transfer request command is written until the output pulse is switched is 2^{13}/fc.

(3) Low level pulse width of PWM output

This is 12-bit resolution PWM output and one period is $T_M = 2^{13}$/fc [s].

The upper 8 bit of PWMDBR controls the low level pulse width of the pulse output with a period of Ts ($Ts = T_M/16$).When the upper 8 bit of the PWM data latch is n (n = 0 to 255), this low level pulse width becomes n x to (to = 2/fc).

The lower 4 bit of PWMDBR controls the output position of the additional pulse of which the width is to between Ts (1) to Ts (15) out of 16 periods contained in Ts period.

When the lower 4 bit data of PWMDBR is m (m = 0 to 15), the additional pulse is generated in each of m periods between Ts (1) to Ts (15). (The additional pulse is not generated at Ts (0).)

The output position of the additional pulse is controlled by setting some bits in the lower 4 bit of PWMDBR to "1". An example to set a position of the additional pulse referred to table 2-11 as mentioned below.

Figure 2-68 (3) shows an example of output pulse.

Table2-11. Setting example of the additional pulse output position

	Setting of the lower 4 bit of PWM data latch				Relative position when the additional pulse is output between Ts (1) to Ts (15) in T_M period
	bit 3	bit 2	bit 1	bit 0	
a)	0	0	0	0	without additional pulse
b)	0	0	0	1	8
c)	0	0	1	0	4,12
d)	0	1	0	0	2, 6, 10, 14
e)	1	0	0	0	1, 3, 5, 7, 9, 11, 13, 15

Note : It is possible to combine a) to e). The position where the additional pulse is output can be set with sixteen kinds.

(4) Pin operation by reset

PWMCR, PWMSR and PWMDBR are initialized by reset. In addition, the internal 12-bit counter, the data transfer buffer and the PWM data latch are initialized. P80 to P83 pins become "Hz" during and just after reset. When the PWM pin is used as PWM output, the pull-up resistor must be externally added to retain the pin state to "H" during reset and after reset cancel.

After reset, when P80 to P83 pins are used as PWM output, the output latch of P8 (#0008_H) must be set to "1", subsequently P8CR (#0009_H) must be set to the output mode.

(5) PWM output in STOP, SLOW, SLEEP mode

PWM outputs can output low level pulse in only NORMAL1 (or IDLE1) and NORMAL2 (or IDLE2) modes. When P80 to P83 are used as PWM outputs and the operation mode is transferred NORMAL1 mode to STOP1 mode or from NORMAL2 mode to STOP2 mode (or SLOW mode, SLEEP mode), the PWM pin always output "High".

Consequently, the lower level pulse can not be output. (PWMCR, PWMSR and PWMDBR are initialized in STOP, SLOW and SLEEP mode, additionally, the internal 12-bit counter, the data transfer buffer and the PWM data latch are initialized.)

In addition, when the operation mode is transferred from STOP1 mode to NORMAL1 mode, or from STOP2 (or SLOW, SLEEP mode) to NORMAL2 mode, the registers necessary for PWM output control must be initialized to output PWM pulse again, which is the same settings after reset. The PWM control register must be initialized according to the procedures mentioned in figure 2-68 (2).

(6) PWM output pulse stop

To output always "H" after T_M period (Ts (0) to Ts (15)) of the PWM pulse which is being written is completed, "00_H" must be written to the lower and the upper of PWMDBR according to the procedures mentioned in figure 2-68 (2) and the data transfer request command must be written. Note that the maximum time from the point at which "00_H" is written until PWM pulse is always output by "H" is 2^{13}/fc.

(7) PWM output pulse abort ("H" is always output.)

To abort PWM output in the middle of PWM pulse output ("H" is always output), P8CR input mode must be set. Consequently, the PWM pin always outputs "H" by the pull-up resistor which is externally added.

When P8CR is set to the input mode without the external pull-up resistor, note that the pin becomes "Hz".

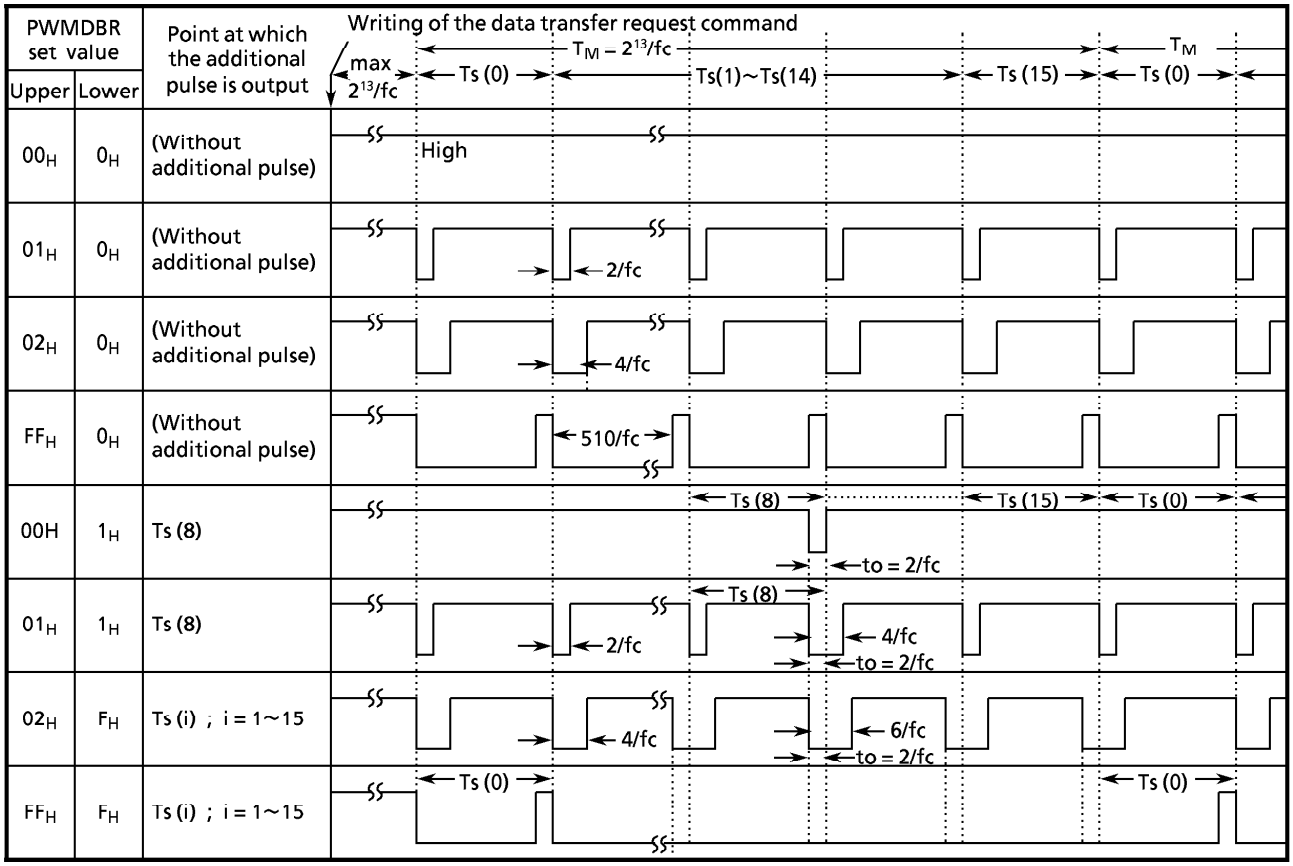


Figure 2-68 (3). PWM Output Pulse

2.12 10-bit A/D Converter (ADC)

The 87CH48 each have an 16-channel multiplexed-input 10-bit successive approximate type A/D converter.

2.12.1 Configuration

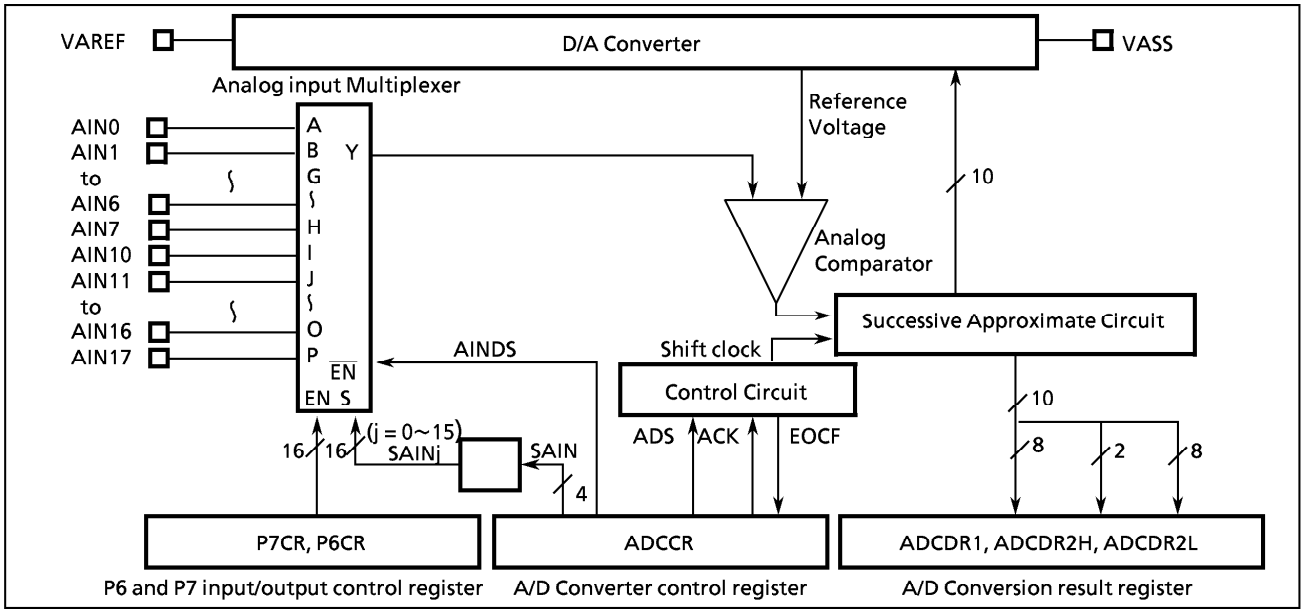


Figure 2-69. A/D Converter

2.12.2 Control

The A/D converter is controlled by an A/D converter control register (ADCCR), a port P6 input/output control register (P6CR) and a port P7 input/output control register (P7CR).

A/D Converter Control Register

ADCCR ($000E_H$)

7	6	5	4	3	2	1	0
EOCF	ADS	ACK	AINDS	SAIN			

(Intial value : 0000 0000)

SAIN	Analog input selection	0000 : AIN0 0001 : AIN1 0010 : AIN2 0011 : AIN3 0100 : AIN4 0101 : AIN5 0110 : AIN6 0111 : AIN7 1000 : AIN10 1001 : AIN11 1010 : AIN12 1011 : AIN13 1100 : AIN14 1101 : AIN15 1110 : AIN16 1111 : AIN17	R/W
AINDS	Analog input control	0 : Enable 1 : Disable	
ACK	Conversion time	0 : 196/fc : (at 8 MHz 24.5 μs). (at 4.2 MHz 46.7 μs) 1 : 784/fc : (at 8 MHz 98 μs). (at 4.2 MHz 187 μs)	
ADS	A/D convertion start	0 : – 1 : A/D convertion start	
EOCF	End of A/D conversion flag	0 : Under conversion or before conversion 1 : End of conversion	Read only

Note 1 : Select analog input when A/D converter stops.
Note 2 : The ADS is automatically cleared to "0" after starting conversion.
Note 3 : The EOCF is cleared to "0" when reading the ADCDR1 or the ADCDR2H, ADCDR2L.
Note 4 : The EOCF is read-only.

Figure 2-70. A/D Converter Control Register

A/D Conversion Result Register

ADCDR1 ($000F_H$) — Read only

7	6	5	4	3	2	1	0
DATA9	DATA8	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2

ADCDR2H (0025_H) — Read only

7	6	5	4	3	2	1	0
"1"	"1"	"1"	"1"	"1"	"1"	DATA9	DATA8

ADCDR2L (0024_H) — Read only

7	6	5	4	3	2	1	0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

P6 input/output control register

P6CR ($000C_H$) — (Initial value : 0000 0000) Write only

7	6	5	4	3	2	1	0

P6CR (i) = 0			P6CR (i) = 1		
AINDS = 0		AINDS = 1	AINDS = 0		AINDS = 1
SAIN (j) = 1	SAIN (j) = 0	–	SAIN (j) = 1	SAIN (j) = 0	–
analig input	input port	input port	analig input	output port	output port

P7 input/output control register

P7CR ($000D_H$) — (Initial value : 0000 0000) Write only

7	6	5	4	3	2	1	0

P7CR (i) = 0			P7CR (i) = 1		
AINDS = 0		AINDS = 1	AINDS = 0		AINDS = 1
SAIN (j) = 1	SAIN (j) = 0	–	SAIN (j) = 1	SAIN (j) = 0	–
analig input	input port	input port	analig input	output port	output port

Figure 2-71. A/D Converter Result Register and P6, P7 Input/Output Control Register

2.12.3 Operation

Apply analog reference voltage to pins VAREF and VASS.

Note : The voltage value of VAREF should be kept the condition as below,
$VAREF \leqq VDD$

(1) Start of A/D conversion

First, select one of sixteen analog inputs AIN17 to AIN10, AIN7 to AIN0 with the SAIN (bits 3 to 0 in ADCCR), clear the AINDS (bit 4 in ADCCR) to "0". Other ports, not be selected as an used as output ports, be careful not to execute the output instruction for any port during the conversion in order to keep the accuracy of conversion.

A/D conversion time is set with the ACK (bit 5 in ADCCR).

A/D conversion is started by setting the ADS (bit 6 in ADCCR) to "1". The conversion is accomplished in 49 machine cycles (196/fc [s], ACK = 0).

The EOCF (bit 7 in ADCCR) is set to "1" at the end of the conversion.

If the ADS (bit 6 in ADCCR) is set to "1" during the conversion, the operation is initialized and the conversion is started again.

(2) Reading of A/D conversion result

After the end of the conversion, read the conversion result from the ADCDR1 or the ADCDR2H, ADCDR2L.

The EOCF is automatically cleared to "0" when reading the ADCDR1 or the ADCDR2H, ADCDR2L.

Reading the conversion result during A/D conversion, an unexpected value is given.

(3) A/D conversion in STOP mode

When the MCU places in the STOP mode during the A/D conversion, the conversion is terminated and the ADCDR1 or ADCDR2H, ADCDR2L contents become indefinite.

However, if the STOP mode is started after the end of conversion (EOCF = 1), the ADCDR1 or the ADCDR2H, ADCDR2L contents are held.

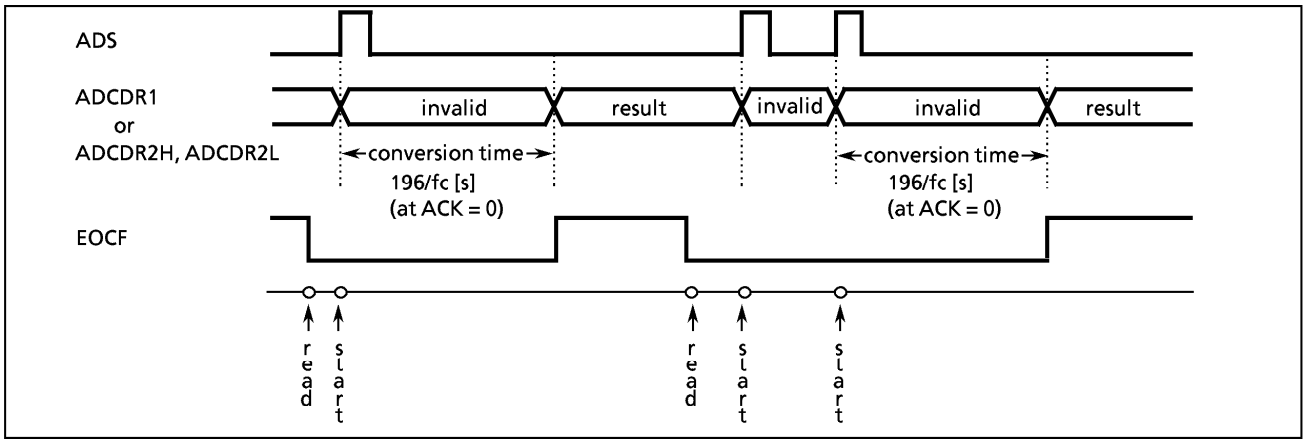


Figure 2-72. A/D Conversion Timing Chart

Example:

```
        ; AIN SELECT
        LD      (ADCCR) , 00100100B     ;   selects conversion time and AIN4
        ; A/D CONVERT START
        LD      (ADCCR) . 01100100B     ;   ADS = 1
SLOOP : TEST    (ADCCR) . 7             ;   EOCF = 1 ?
        JRS     T, SLOOP
        ; RESULT DATA READ
        LD      (9EH), (ADCDR2H)
        LD      (9DH), (ADCDR2L)
```

(4) Notes for the current consumption on the stop mode when using an A/D 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}). TMP87CH48 and TMP87PH48 do 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 2-73 (a).

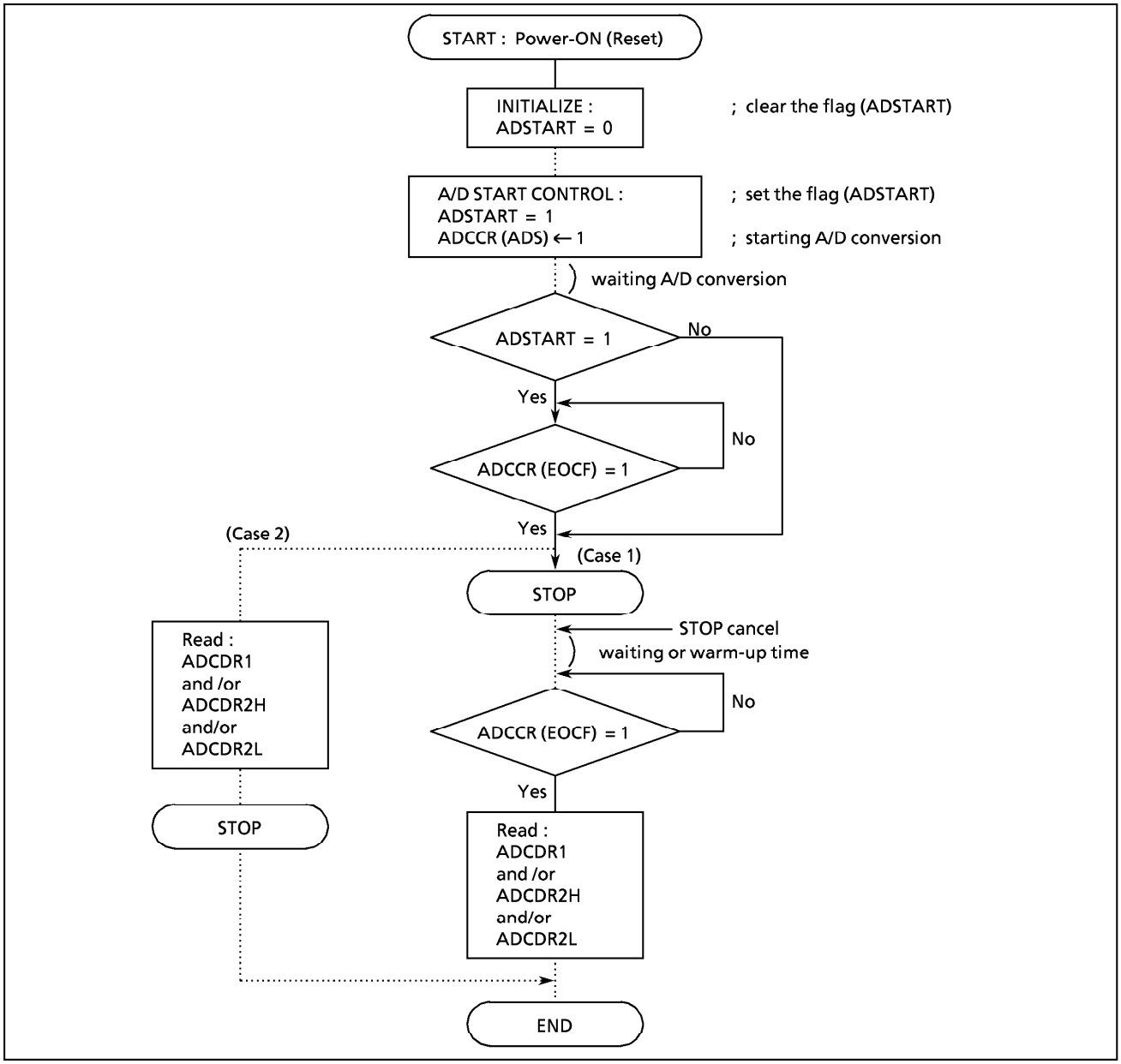


Figure 2-73 (a). Example Flow Chart for STOP Mode Control in the A/D Converter System

(5) The relation between Analog Input Voltage and A/D Conversion Result

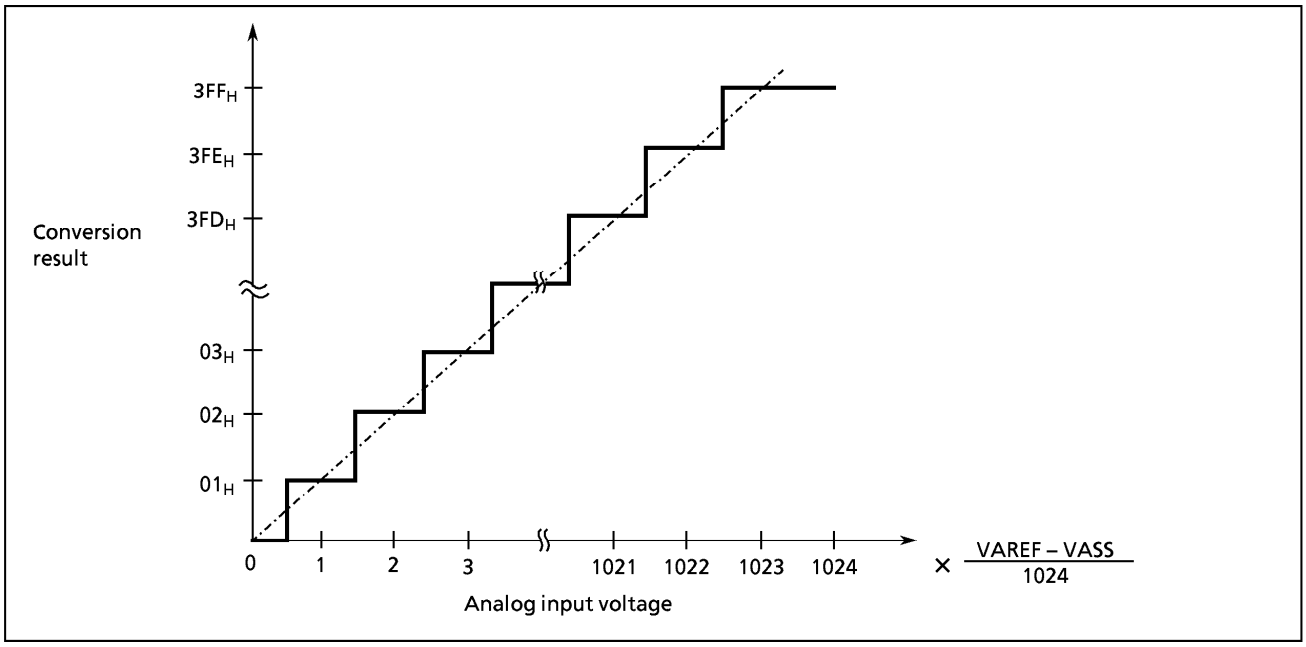


Figure2-73 (b). Analog Input Voltage vs A/D Conversion Result (typ.)

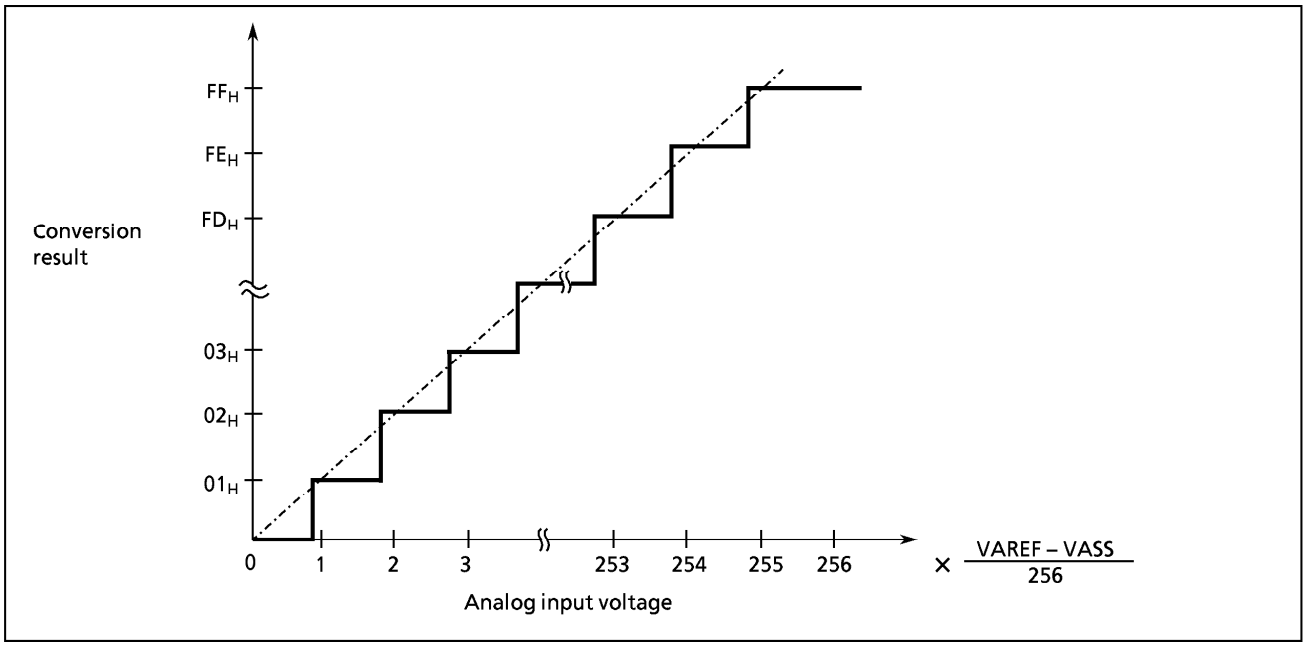


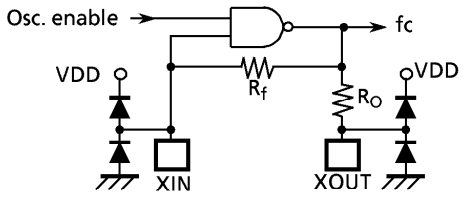
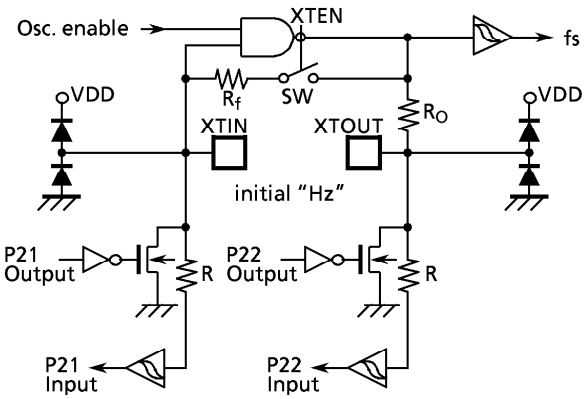
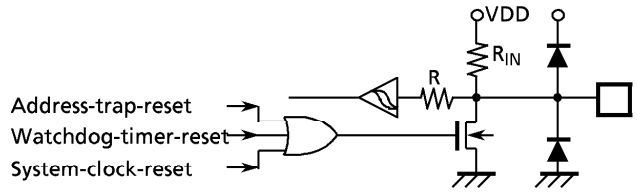
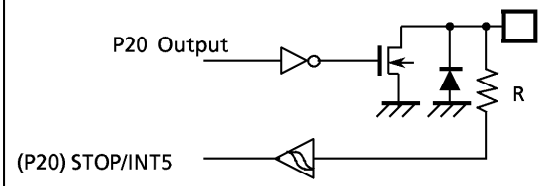
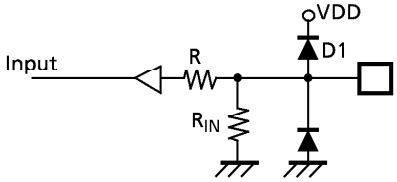
Figure2-73 (c). Analog Input Voltage vs A/D Conversion Result (typ.)

INPUT/OUTPUT CIRCUITRY

The instruction for specifying Masking Option (I/O code) in ES Order Sheet is described in ADDITIONAL INFORMATION "Notice for Masking Option of TLCS-870 series" section 8.

(1) Control pins

The input/output circuitries of the 87CH48 control pins are shown below.

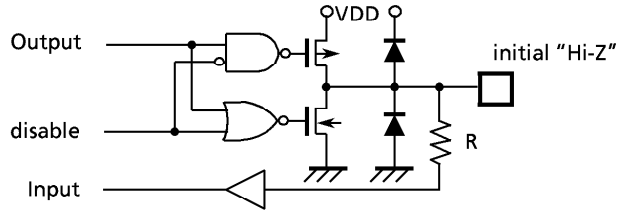
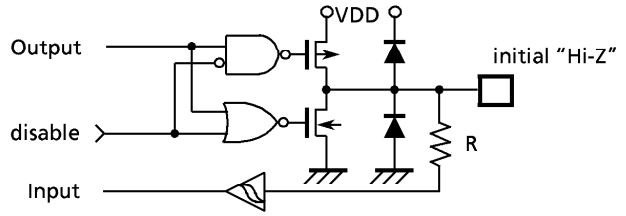
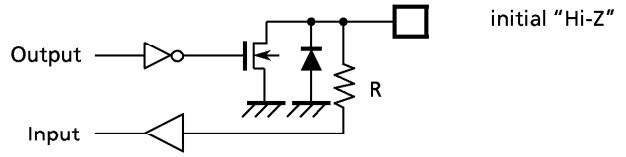
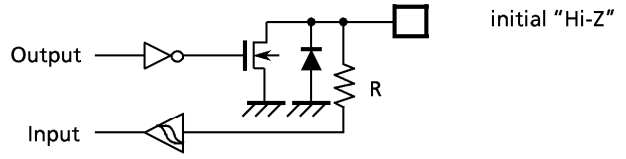
CONTROL PIN	I/O	INPUT/OUTPUT CIRCUITRY	REMARKS
XIN XOUT	Input Output		Resonator connecting pins (high-frequency) R_f = 1.2MΩ (typ.) R_O = 1.5kΩ (typ.)
XTIN XTOUT P21 P22	Input Output I/O I/O		XTIN, XTOUT Resonator connecting pins (low-frequency) R_f = 6MΩ (typ.) R_O = 220kΩ (typ.) XTEN (Initial : 0) SW (XTEN = 0 : OFF) (XTEN = 1 : ON) P21, P22 Sink open drain output Hysteresis input R = 1kΩ (typ.)
RESET	I/O		Sink open drain output Hysteresis input Pull-up resistor R_{IN} = 220kΩ (typ.) R = 1kΩ (typ.)
STOP/INT5 (P20)	I/O		Hysteresis input R = 1kΩ (typ.)
TEST	Input		Pull-down registor R_{IN} = 70kΩ (typ.) R = 1kΩ (typ.)

Note 1 : The 87PH48 does not have a pull-down resistor (R_{IN}) and a diode (D_1) for TEST pin.

Note 2 : The 87PH48/CH48 are placed in the single-clock mode during reset.

(2) I nput/Output Ports

The input/output circuitries of the 87CH48 input/output ports are shown below.

PORT	I/O	INPUT/OUTPUT CIRCUITRY (I/O CODE : A)	REMARKS
P0 P6 P7 P8	I/O		Tri-state I/O R = 1kΩ (typ.)
P1	I/O		Tri-state I/O Hysteresis input R = 1kΩ (typ.)
P3	I/O		Hligh current output only P3 Sink open drain output R = 1kΩ (typ.)
P4 P5	I/O		Sink open drain output Hysteresis input R = 1kΩ (typ.)

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (V_{SS} = 0 V)

PARAMETER	SYMBOL	CONDITIONS	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_{OUT}		– 0.3 to V_{DD} + 0.3	V
Output Current (Per 1 pin)	I_{OUT1}	Ports P0, P1, P2, P4, P5, P6, P7, P8	3.2	mA
	I_{OUT2}	Port P3	30	
Output Current (Total)	Σ I_{OUT1}	Ports P0, P1, P2, P4, P5, P6, P7, P8	120	mA
	Σ I_{OUT2}	Port P3	120	
Power Dissipation	PD		350	mW
Soldering Temperature (time)	Tsld		260 (10 s)	°C
Storage Temperature	Tstg		– 55 to 125	°C
Operating Temperature	Topr		– 30 to 70	°C

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0 V, Topr = – 40 to 85 °C)

PARAMETER	SYMBOL	PINS	CONDITIONS		Min.	Max.	UNIT
Supply Voltage	* V_{DD}		fc = 8 MHz	NORMAL1, 2 mode	4.5	5.5	V
				IDLE1, 2 mode			
			fc = 4.2 MHz	NORMAL1, 2 mode	2.7		
				IDLE1, 2 mode			
			fs = 32.768 kHz	SLOW mode			
				SLEEP mode			
				STOP mode	2.0		
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		0.4	8.0	MHz
			V_{DD} = 2.7 to 5.5 V			4.2	
	fs	XTIN, XTOUT			30.0	34.0	kHz

D.C. CHARACTERISTICS

(V_{SS} = 0 V, Topr = – 40 to 85 °C)

PARAMETER	SYMBOL	PINS	CONDITIONS	Min.	Typ.	Max.	UNIT
Hysteresis Voltage	V_{HS}	Hysteresis inputs	VDD = 5.0 V	–	0.9	–	V
Input Current	I_{IN1}	TEST	VDD = 5.5 V V_{IN} = 5.5 V / 0 V	–	–	± 2	μA
	I_{IN2}	Open drain ports, Tri-state ports					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
Input Resistance	R_{IN2}	$\overline{RESET}$	VDD = 5.0 V	100	220	450	kΩ
Output Leakage Current	I_{LO}	Sink open drain ports	V_{DD} = 5.5 V, V_{OUT} = 5.5 V	–	–	2	μA
		Tri-state ports	V_{DD} = 5.5 V, V_{OUT} = 5.5/0 V	–	–	± 2	
Output High Voltage	V_{OH2}	Tri-state ports	V_{DD} = 4.5 V, I_{OH} = – 0.7 mA	4.1	–	–	V
Output Low Voltage	V_{OL}	Except for XOUT and P3	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.4	mA
Output Low current	I_{OL3}	P3	V_{DD} = 4.5 V, V_{OL} = 1.0 V	–	20	–	mA
Supply Current in NORMAL 1, 2 modes	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3 V / 0.2 V fc = 8 MHz fs = 32.768 kHz	–	4.5	5.5	mA
Supply Current in IDLE 1, 2 modes				–	2.5	4.0	mA
Supply Current in NORMAL 1, 2 modes			V_{DD} = 3.0 V, V_{IN} = 2.8V/0.2V V_{IN} = 4.19 MHz fs = 32.768 kHz	–	1.75	3.0	mA
Supply Current in IDLE 1, 2 modes				–	1.25	2.0	mA
Supply Current in SLOW mode			V_{DD} = 3.0 V V_{IN} = 2.8 V / 0.2 V fs = 32.768 kHz	–	20	30	μA
Supply Current in SLEEP mode				–	10	20	μA
Supply Current in STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V / 0.2 V	–	0.5	10	μA

Note 1 : Typical values show those at Topr = 25 °C
Note 2 : Input Current I_{IN1},I_{IN3} ; The current through resistor is not included, when the input resistor (pull-upor pull-down) is contained.
Note 3 : IDD except for I_{REF}.

A/D CONVERSION CHARACTERISTICS

(V_{SS} = 0V, V_{DD} = 2.7 to 5.5V, Topr = – 40 to 85 °C)

PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.			UNIT
					ADCDR1	ADCDR2		
						ACK = 0	ACK = 1	
Analog Reference Voltage	V_{AREF}	V_{AREF} – V_{ASS} ≧ 2.5 V	2.7	–	V_{DD}			V
	V_{ASS}		V_{SS}	–	1.5			
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	1.2			mA
Nonlinearity Error		V_{DD} = 5.0, V_{SS} = 0.0 V V_{AREF} = 5.000 V V_{ASS} = 0.000 V or V_{DD} = 2.7, V_{SS} = 0.0 V V_{AREF} = 2.700 V V_{ASS} = 0.000 V	–	–	1.0			LSB
Zero Point Error			–	–	± 1	± 3	± 2	
Full Scale Error			–	–	± 1	± 3	± 2	
Total Error			–	–	± 2	± 6	± 4	

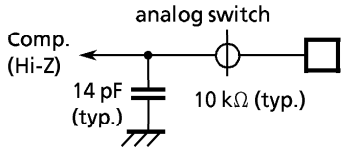
Note 1 : $\triangle V_{AREF} = V_{AREF} - V_{ASS}$
ADCDR1 ; 8 bit – A/D conversion result (1LSB = $\triangle V_{AREF}$ / 256)
ADCDR2 ; 10 bit – A/D conversion result (1LSB = $\triangle V_{AREF}$ / 1024)
Note 2 : Quantizing error is not contained in those errors.

A.C. CHARACTERISTICS

(V_{SS} = 0 V, Topr = – 40 to 85 °C)

PARAMETER	SYMBOL	CONDITIONS	V_{DD}	Min.	Typ.	Max.	UNIT
Machine Cycle Time	t_{cy}	In NORMAL 1, 2 mode	4.5 to 5.5V	0.5	–	10	μs
		In IDLE 1, 2 mode					
		In SLOW mode	2.7 to 5.5V	117.6	–	133.3	
		In SLEEP mode					
High Level Clock Pulse Width	t_{WCH}	For external clock operation (XIN input), fc = 8 MHz	4.5 to 5.5V	62.5	–	–	ns
Low Level Clock Pulse Width	t_{WCL}						
High Level Clock Pulse Width	t_{WSH}	For external clock operation (XTIN input), fs = 32.768 kHz	2.7 to 5.5V	14.7	–	–	μs
Low Level Clock Pulse Width	t_{WSL}						
A/D Conversion Time	t_{ADC}	ADCCR bit 4 ; ACK = 0	–	–	49 tcy	–	ns
		ADCCR bit 4 ; ACK = 1	–	–	196 tcy	–	

AIN (i) internal circuit



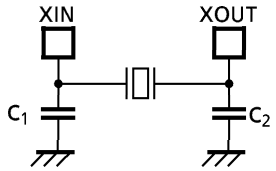
Note 1 : V_{AIN} must be kept the voltage level during A/D conversion period (t_{ADC})

Note 2 : i = 17 to 10, 07 to 00

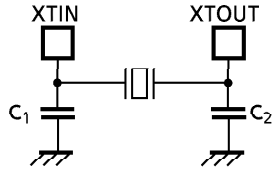
Timing of A/D Conversion

RECOMMENDED OSCILLATING CONDITIONS (V_{SS} = 0 V, V_{DD} = 2.7 / 4.5 to 5.5 V, Topr = – 40 to 85 °C)

PARAMETER	Oscillator	Oscillation Frequency	VDD	Recommended Oscillator	Recommended Constant	
					C_1	C_2
High-frequency Oscillation	Ceramic Resonator	8 MHz	4.5 to 5.5V	KYOCERA KBR8.0M	30 pF	30 pF
		4 MHz	2.7 to 5.5V	KYOCERA KBR4.0MS		
				MURATA CSA4.00MG		
	Crystal Oscillator	8 MHz	4.5 to 5.5V	TOYOCOM 210B 8.0000	20 pF	20 pF
		4 MHz	2.7 to 5.5V	TOYOCOM 204B 4.0000		
Low-frequency Oscillation	Crystal Oscillator	32.768 kHz	2.7 to 5.5V	NDK MX-38T	15 pF	15 pF



(1) High-frequency Oscillation



(2) Low-frequency Oscillation

Note : When it is used in high electrical field, an electrical shield of the package is recommended to retain normal operations.