CMOS 8-Bit Microcontroller

TMP87CH74AF, TMP87CM74AF

The 87CH74A/CM74A are the high speed and high performance 8-bit single chip microcomputers. These MCU contain 8-bit A/D conversion inputs and a VFT (Vacuum Fluorescent Tube) driver on a chip.

Part No.	ROM	RAM	Package	OTP MCU
TMP87CH74AF	16 K × 8-bit	512 × 8-bit	P-QFP80-1420-0.80B	TMP87PM74F
TMP87CM74AF	32 K × 8-bit	1 K × 8-bit		

Features

◆8-bit single chip microcomputer TLCS-870 Series

◆Instruction execution time: 0.5 μs (at 8 MHz), 122 μs (at 32.768 kHz)

◆412 basic instructions

- Multiplication and Division (8 bits × 8 bits , 16 bits ÷ 8 bits)
- Bit manipulations
  (Set/Clear/Complement/Load/Store/Test/Exclusive OR)
- 16-bit data operations
- 1-byte jump/subroutine-call (Short relative jump/ Vector call)

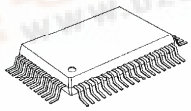
◆15 interrupt sources (External: 6, Internal: 9)

- All sources have independent latches each,
  and nested interrupt control is available.
- 3 edge-selectable external interrupts with noise reject
- High-speed task switching by register bank changeover

◆11 Input/Output ports (71 pins)

- High current output: 16 pins (typ. 20 mA)

P-QFP80-1420-0.80B



TMP87CH74AF
TMP87CM74AF
TMP87PM74F

030619EBP1

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.

◆Two 16-bit Timer/Counters
- Timer, Event counter, programmable pulse generator output, Pulse width measurement, External trigger timer, Window modes.

◆Two 8-bit Timer/Counters
- Timer, Event counter, Capture (Pulse width/duty measurement), PWM output, Programmable divider output modes

◆Time Base Timer (Interrupt frequency: 1 Hz to 16384 kHz)

◆Divider output function (frequency: 1 kHz to 8 kHz)

◆Watchdog Timer
- Interrupt source/reset output (programmable)

◆8-bit Serial Interface
- With 8 bytes transmit/receive data buffer
- Internal/external serial clock, and 4/8-bit mode

◆Serial bus Interface
- I^2C-bus, 8-bit SIO modes

◆8-bit successive approximate type A/D converter with sample and hold
- 12 analog inputs
- Conversion time: 23 μs at 8 MHz

◆Vacuum Fluorescent Tube Driver (automatic display)
- Programmable grid scan
- High breakdown voltage ports (max. 40 V × 37 bits)

◆Dual clock operation
- Single/Dual-clock mode (option)

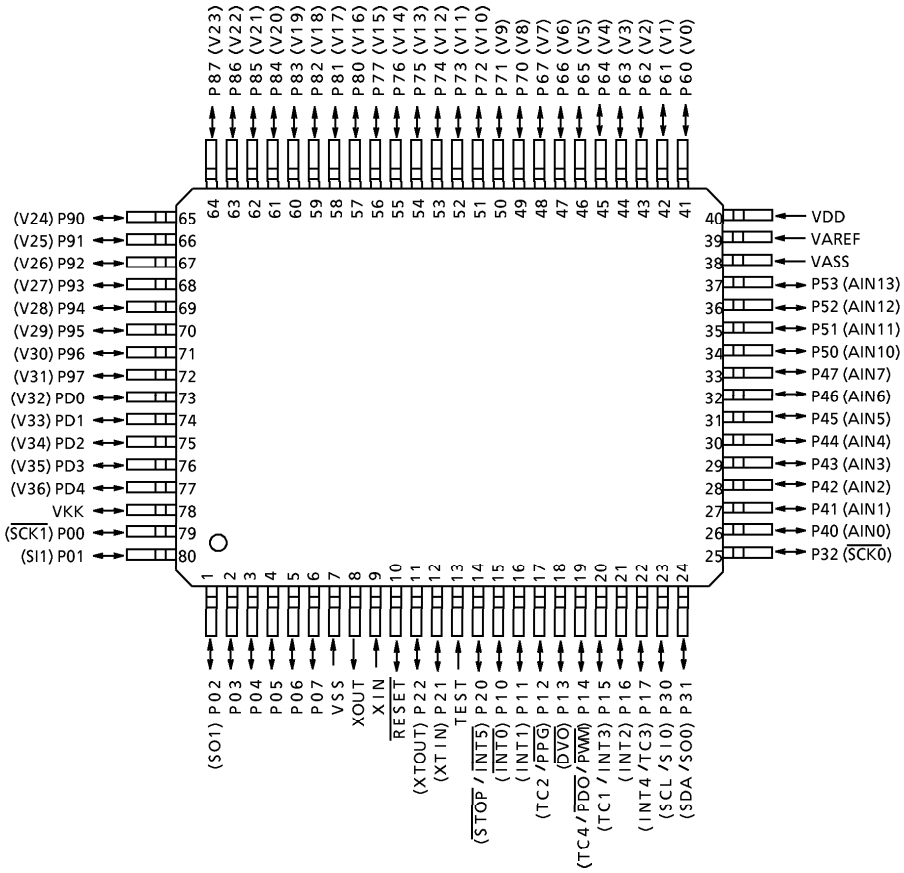
◆Five Power saving operating modes
- STOP mode: Oscillation stops. Battery/Capacitor back-up. Port output hold/High-impedance.
- SLOW mode: Low power consumption operation using low-frequency clock (32.768 kHz).
- IDLE1 mode: CPU stops, and Peripherals operate using high-frequency clock. Release by interrupts.
- IDLE2 mode: CPU stops, and Peripherals operate using high-and low-frequency clock. Release by interrupts.
- SLEEP mode: CPU stops, and Peripherals operate using low-frequency clock. Release by interrupts.

◆Wide operating voltage: 2.7 to 5.5 V at 32.768 kHz, 4.5 to 5.5 V at 8 MHz / 32.768 kHz

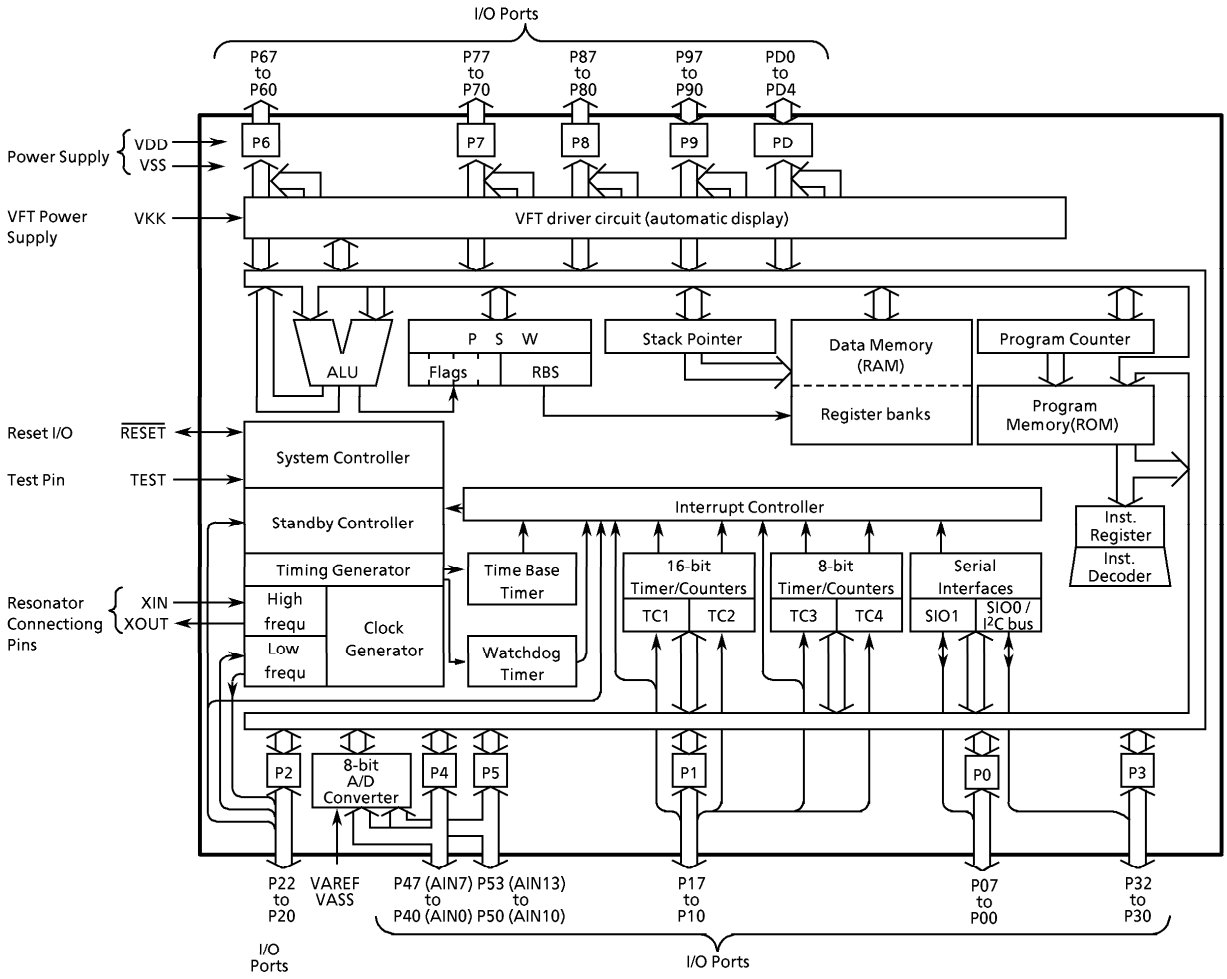
◆Emulation Pod: BM87CM75F0A

Pin Assignments (Top View)

P-QFP80-1420-0.80B



Block Diagram



Pin Function

Pin Name	Input / Output	Function	
P07 to P03	I/O	Two 8-bit programmable input/output ports (tri-state). Each bit of these ports can be individually configured as an input or an output under software control. When used as a SIO input/output, an External interrupt input, a timer/counter input, the latch must be set to "0". When used as a PPG output or divider output, the latch must be set to "1".	
P02 (SO1)	I/O (Output)		SIO1 serial data Output
P01 (SI1)	I/O (Input)		SIO1 serial data Input
P00 ($\overline{SCK1}$)	I/O (I/O)		SIO1 serial clock input/output
P17 (INT4/TC3)	I/O (Input)		External interrupt input 4 or Timer/Counter 3 input
P16 (INT2)			External interrupt input 2
P15 (INT3/TC1)			External interrupt input 3 or Timer/Counter 1 input
P14 (TC4/$\overline{PDO}$/$\overline{PWM}$)	I/O (I/O)		Timer counter 4 input or 8-bit programmable divider output or 8-bit PWM output
P13 ($\overline{DVO}$)	I/O (Output)		Divider output
P12 (TC2/$\overline{PPG}$)	I/O (I/O)		Timer counter 2 input or Programmable pulse generator output
P11 (INT1)	I/O (Input)		External interrupt input 1
P10 ($\overline{INT0}$)			External interrupt input 0
P22 (XTOUT)	I/O (Output)	3-bit input/output port with latch. When used as input port, or external interrupt input, STOP mode release signal input, the latch must be set to "1".	Resonator 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
P32 ($\overline{SCK0}$)	I/O (I/O)	3-bit programmable input/output ports (Sink open drain). Each bit of these ports can be individually configured as an input or an output under software control. When used as a I^2C input/output, the latch must be set to "1".	SIO0 serial clock input/output
P31 (SDA/SO0)	I/O (I/O/Output)		I^2Cbus serial data input/output or SIO0 serial data output
P30 (SCL/SI0)	I/O (I/O/Input)		I^2Cbus serial clock input/output or SIO0 serial data Input
P47 (AIN7) to P40 (AIN0)	I/O (Input)	8-bit programmable input/output ports (tri-state). Each bit of these ports can be individually configured as an input or an output under software control. When used as a analog input, the P4CR must be set to "0".	A/D converter analog inputs
P53 (AIN13) to P50 (AIN10)	I/O (Input)	4-bit programmable input/output ports (tri-state). Each bit of these ports can be individually configured as an input or an output under software control. When used as a analog input, the P5CR must be set to "0".	A/D converter analog inputs
P67 (V7) to P60 (V0)	I/O (Output)	Four 8-bit high brackdown voltage output ports with the latch. When used as a VFT driver output, the latch must be cleared to "0".	VFT driver outputs
P77 (V15) to P70 (V8)			
P87 (V23) to P80 (V16)			
P97 (V31) to P90 (V24)			
PD4 (V36) toPD0 (V32)	I/O (Output)	5-bit high breakdown voltage output ports with the latch. When used as a VFT driver output, the latch must be cleared to "0".	

Pin Name	Input / Output	Function
XIN, XOUT	Input, Output	Resonator connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened.
$\overline{\text{RESET}}$	I/O	Reset signal input or watchdog timer output/address-trap-reset output/system-clock-reset outputted.
TEST	Input	Test pin for out-going test. Be tied to low.
VDD, VSS	Power Supply	+ 5 V, 0 V (GND)
VKK		VFT driver power supply
VAREF, VASS		Analog reference voltage inputs (High, Low)

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 64 Kbytes of memory. Figure 1-1 shows the memory address maps of the 87CH74A/M74A. 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.

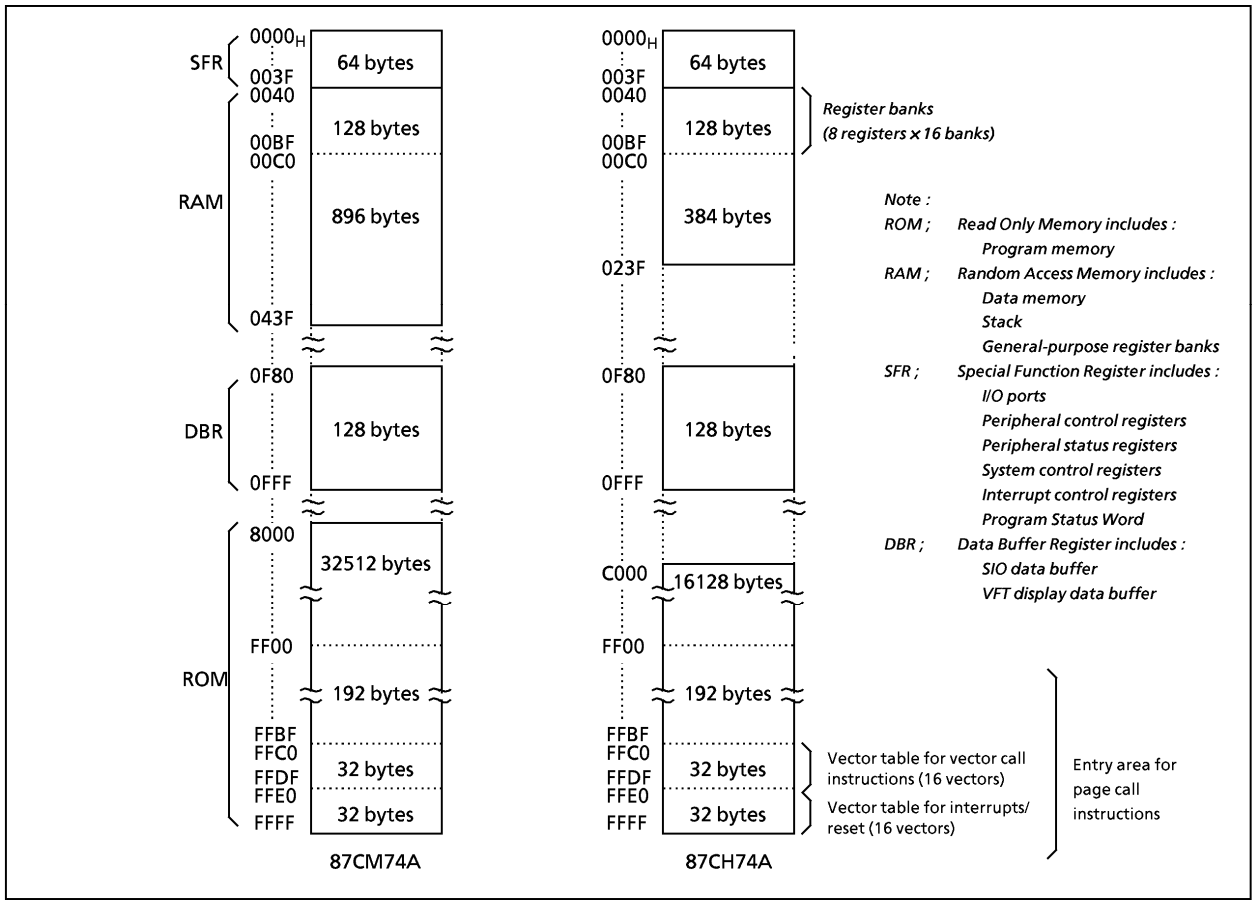


Figure 1-1. Memory Address Maps

1.2 Program Memory (ROM)

The 87CH74A has a 16 K×8-bit (address $C000_H$-$FFFF_H$), the 87CM74A has a 32 K×8-bit (address 8000_H-$FFFF_H$) of program memory (mask programmed ROM).
Addresses $FF00_H$-$FFFF_H$ in the program memory can also be used for special purposes.

(1) **Interrupt / Reset** vector table (addresses $FFE0_H$-$FFFF_H$)
This table consists of a reset vector and 16 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 $FFC0_H$-$FFDF_H$)
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 $FF00_H$-$FFFF_H$) for **page call** instructions
This is the subroutine entry address area for the page call instructions [CALLP n]. Addresses $FF00_H$-$FFBF_H$ are normally used because address $FFC0_H$-$FFFF_H$ 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-byte space.

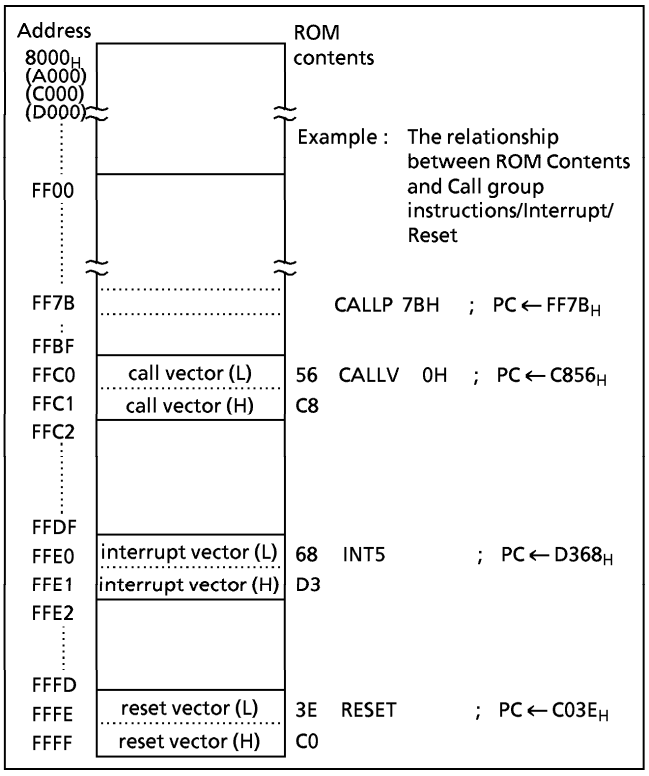


Figure 1-2. Program Memory Map

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-direction jump processing can easily be programmed.

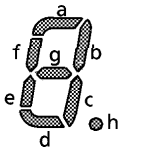
Example 1 : Loads the ROM contents at the address specified by the HL register pair contents into the accumulator (HL≧A000$_H$):

```
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 P0 after executing the following program:

```
        LD      (P0), (PC + A)
        ADD     A, TABLE – $ – 4      ; P0 ←ROM (TABLE + A)
        JRS     T, SNEXT
TABLE : DB      0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0D8H, 80H, 98H
SNEXT :
```

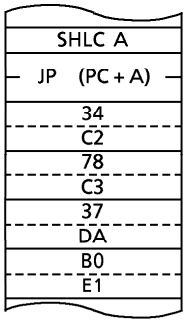
Notes : "$" is a header address of ADD instruction.
DB is a byte data difinition instruction.



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.



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$_H$.

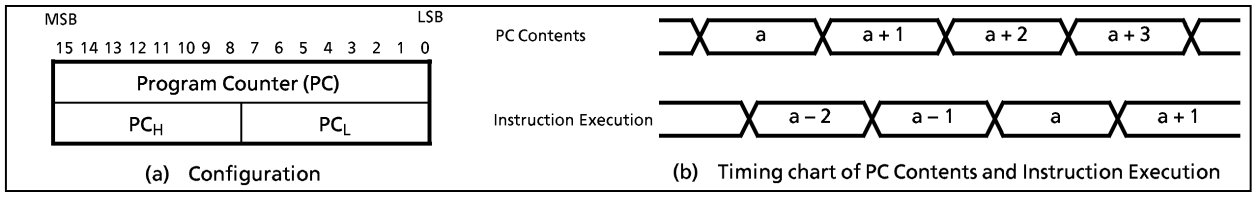


Figure 1-3. Program Counter

1.4 Data Memory (RAM)

The 87CH74A have a 512 × 8-bits (addresses 0040$_H$-023F$_H$), the 87CM74A has a 1K × 8-bits (address 0040$_H$-043F$_H$) of data memory (static RAM). Figure 1-4 shows the data memory map.

Addresses 0000$_H$-00FF$_H$ are used as a direct addressing area to enhance instructions which utilize this addressing mode; therefore, addresses 0040$_H$-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$_H$-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".

The 87CH74A/M74A cannot execute programs placed in the data memory. When the program counter indicates a data memory address, a bus error occurs and an address-trap-reset applies. The $\overline{\text{RESET}}$ pin goes low during the address-trap-reset.

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.*

Example1 : Clears RAM to "00_H" except the bank 0: (87CH74A)

```
        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
```

Example2 : Clears RAM to "00_H" except the bank 0: (87CM74A/PM74)

```
        LD      HL, 0048H       ;  Sets start address to HL register pair
        LD      A, H            ;  Sets initial data (00H) to A register
        LD      BC, 03F7H       ;  Sets number of byte to BC register pair
SRAMCLR : LD    (HL+), A
        DEC     BC
        JRS     F, SRAMCLR
```

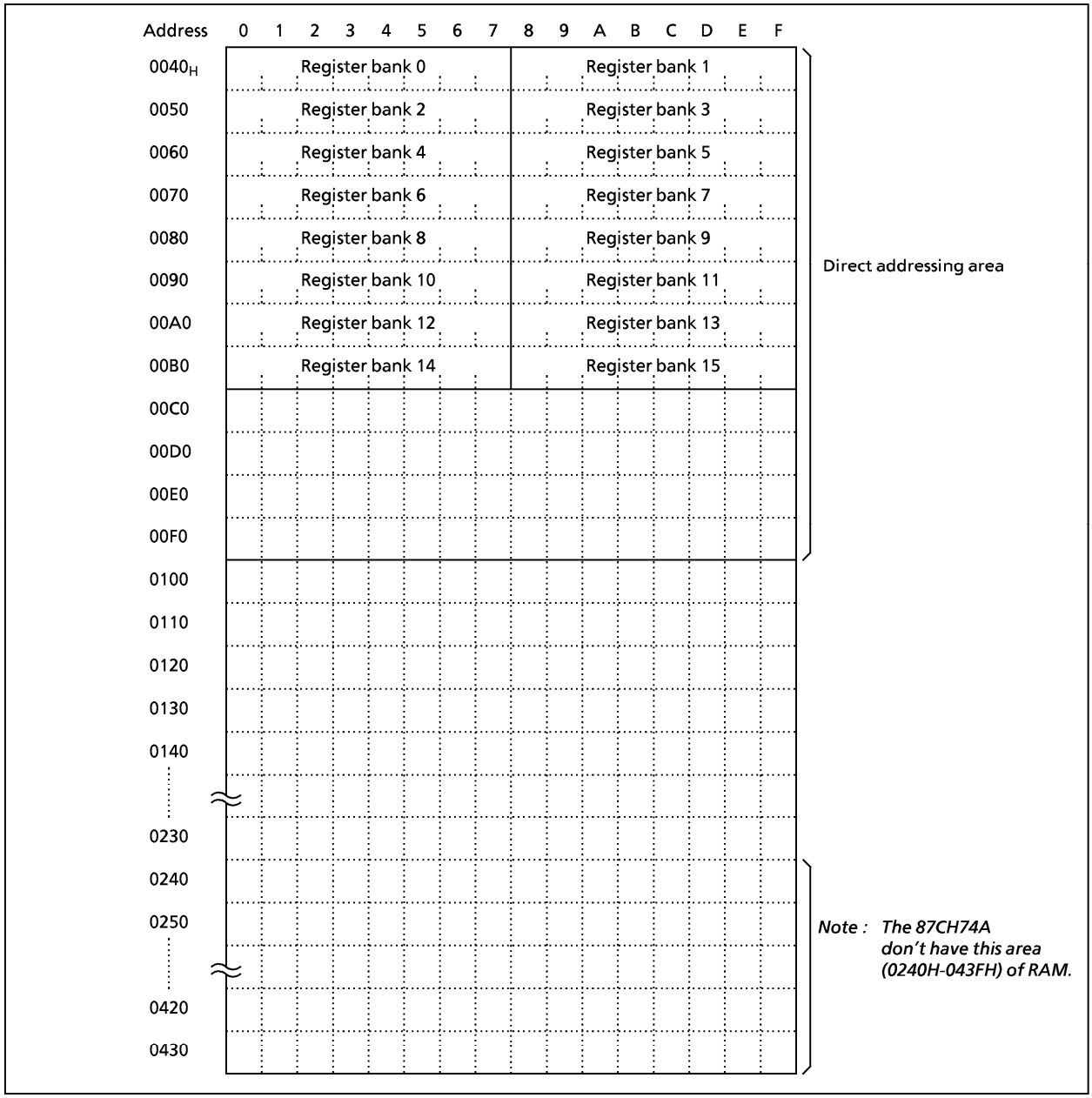


Figure 1-4. Data Memory Map

1.5 General-purpose Register Banks

The general-purpose registers are mapped into addresses 0040_H-$00BF_H$ 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.

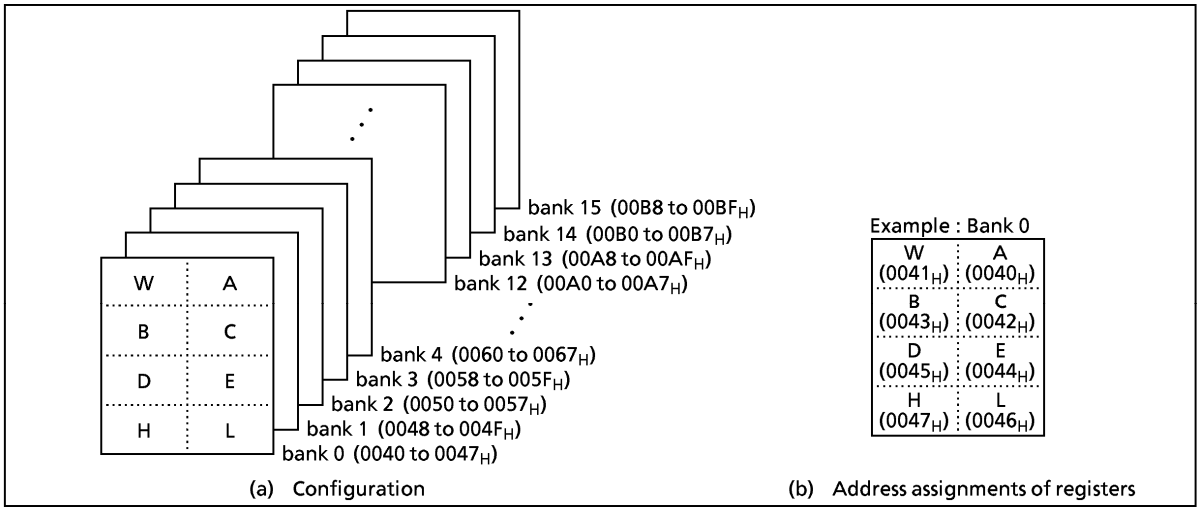


Figure 1-5. General-purpose Register Banks

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      B, 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
        INC     DE
        DEC     B
        JRS     F, SLOOP
```

(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] and [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".

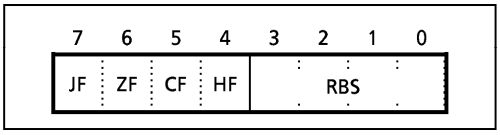


Figure 1-6. PSW (Flags, RBS) Configuration

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 instruction [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
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 will 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, 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 postdecremented when a subroutine call or a push instruction is executed, or when an interrupt is accepted; and the SP is preincremented when a return or a pop instruction is executed. Figure 1-8 shows the stacking order.

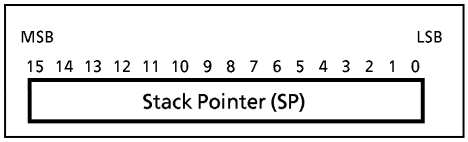


Figure 1-7. Stack Pointer

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, 023FH       ; SP←023F_H
```

Example 2 : To read the SP

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

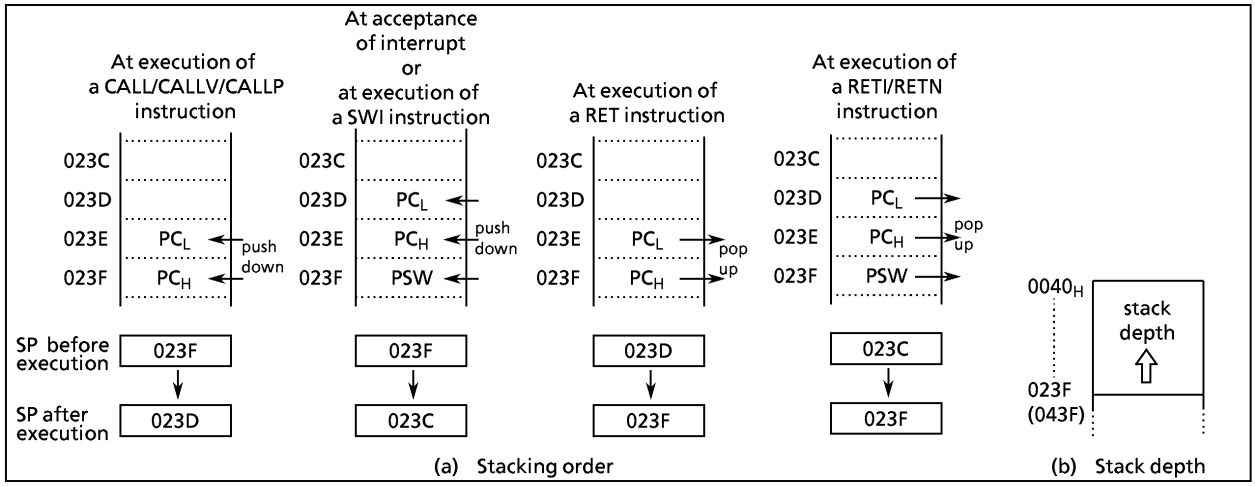


Figure 1-8. Stack

1.8 System Clock Controller

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

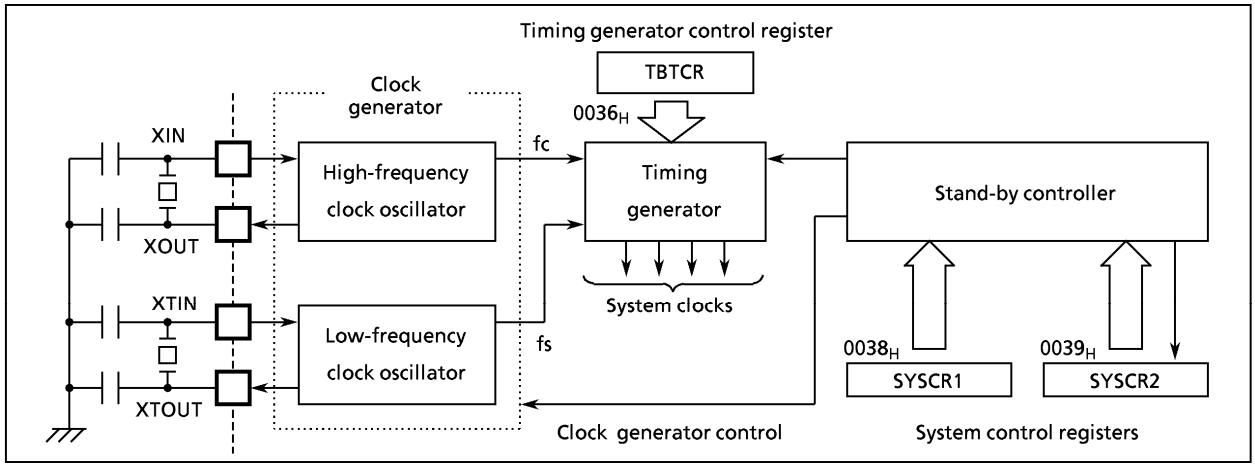


Figure 1-9. System Clock Controller

1.8.1 Clock Generator

The clock generator generates the basic clock which provides the system clocks supplied to the CPU core and on-chip 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.

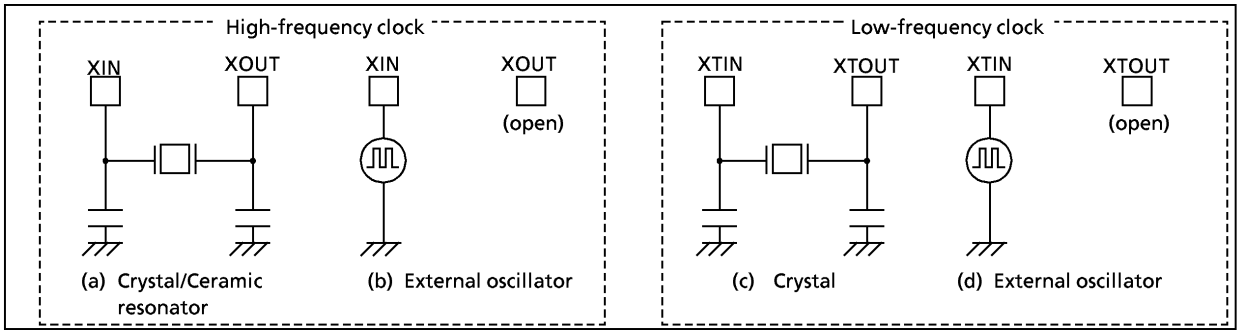
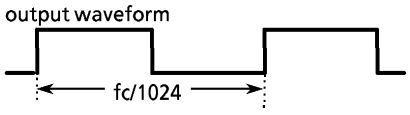


Figure 1-10. Examples of Resonator Connection

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 providing a 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 SIO and HSO
⑦ Generation of source clocks for VFT driver circuit
⑧ Generation of warm-up clocks for releasing STOP mode
⑨ 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 upon 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.

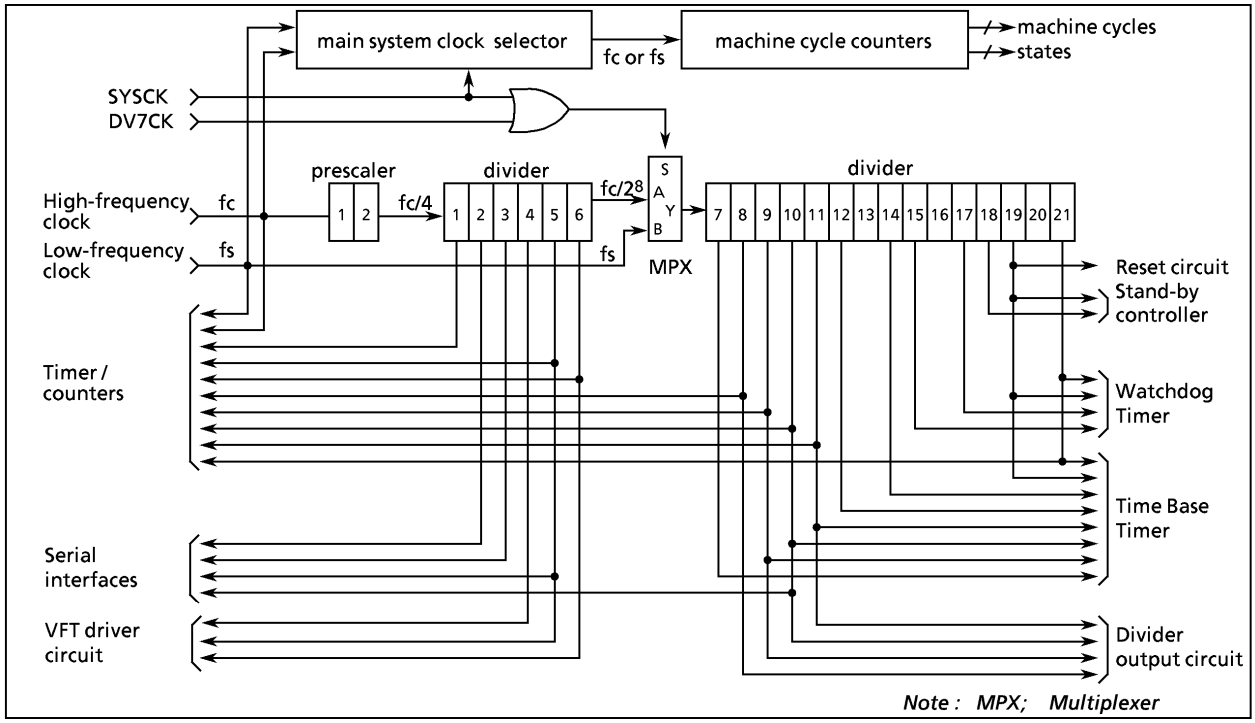


Figure 1-11. Configuration of Timing Generator

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 on-chip 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 forexecution.

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

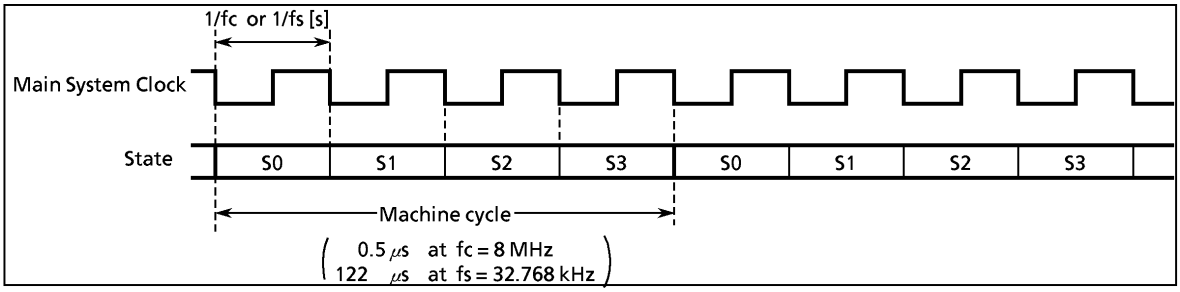


Figure 1-13. Machine Cycle

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 be selected by an option 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 87CH74A/M74A are placed in this mode after reset.

② IDLE1 mode

In this mode, the internal oscillation circuit remains active. 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 the 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 the high-frequency and low-frequency oscillation circuits are used in this mode. P21 (XTIN) and P22 (XTOUT) pins 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 87PM74 is 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 is operated using the high-frequency clock. The on-chip peripherals are operated on the high-frequency clock and/or low-frequency clock. In case that the dual-clock mode has been selected as an option, the 87CH74A/M74A are placed in this mode after reset.

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 are operated 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

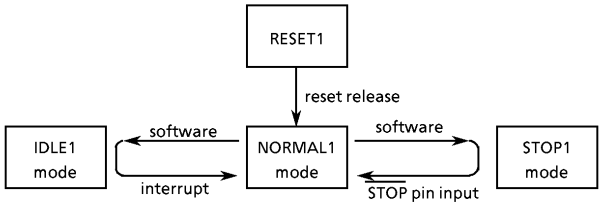
In this mode, the internal oscillation circuits remain active. The CPU and the watchdog timer are halted; however, on-chip peripherals operate using the high-frequency clock and/or the low-frequency clock. Starting and releasing of IDLE2 mode are the same as for IDLE1 mode, except that operation returns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

④ SLEEP mode

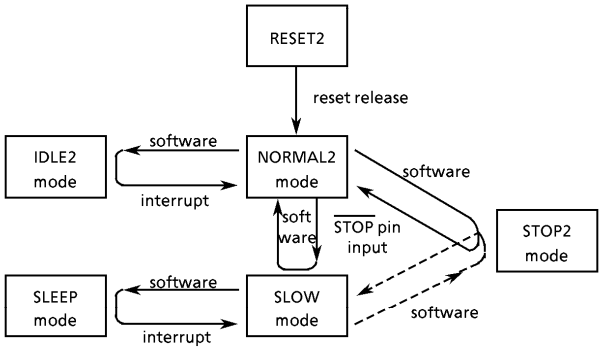
In this mode, the internal oscillation circuit of the low-frequency clock remains active. The CPU, the watchdog timer, and the internal oscillation circuit of the high- frequency clock are halted; however, on-chip peripherals operate using the low-frequency clock. Starting and releasing of SLEEP mode 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.



(a) Single-clock mode



(b) Dual-clock mode

Note 1: NORMAL1 and NORMAL2 modes are generically called NORMAL; STOP1 and STOP2 are called STOP; and IDLE1, IDLE2 and SLEEP are called IDLE.

Note 2: The 87PM74 doesn't have RESET2 mode

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 (Note 1)	
	IDLE1			halt		
	STOP1	turning off oscillation			halt	—
Dual-Clock	RESET2	turning on oscillation	turning on oscillation	reset	reset	4/fc [s]
	NORMAL2			High-frequency	operate (High and/or Low) (Note 1)	
	IDLE2			halt		
	SLOW	turning off oscillation		Low-frequency	Low-frequency (Note 2)	4/fs [s]
	SLEEP			halt		
	STOP2		turning off oscillation		halt	—

Note 1: The Vacuum Fluorescent Tube (VFT) driver circuit are halted.

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 Normal2 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 by set.

System Control Register 2

SYSCR2 (0039_H)

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

(Initial value: 10/100 ****)

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 be selected for XTEN. Always specify when ordering ES (engineering sample).

XTEN	operating mode after reset
0	Single-clock mode (NORMAL1)
1	Dual-clock mode (NORMAL2)

Note 6 : The instruction for specifying Masking Option (Operating Mode) in 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 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 for 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 the STOP mode but instead will immediately start the release sequence (warm-up). Thus, to start the 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, STOP mode does not start if
                                              port P20 is at high
          JRS    F, SINT5
          LD     (SYSCR1), 01000000B       ;  Sets up the level-sensitive release mode.
          SET    (SYSCR1) . 7              ;  Starts STOP mode
          LDW    (IL), 1111011101010111B   ;  IL11, 7, 5, 3 ← 0 (Clears interrupt latehes)
SINT5 :   RETI
```

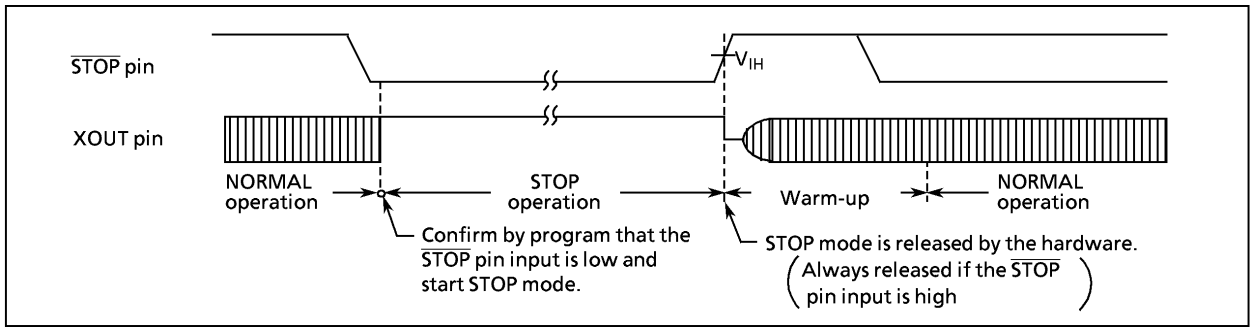


Figure 1-16. Level-sensitive Release Mode

Note : 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

```
PINT5 :   LD    (SYSCR1) , 00000000B          ;  OUTEN ← 0 (specifies high-impedance)
          DI                                    ;  IMF ← 0 (disables interrupt service)
          SET   (SYSCR1). STOP                  ;  STOP ← (activates stop mode)
          LDW   (IL), 1111011101010111B         ;  IL11, 7, 5, 3 ← 0 (Clears interrupt latehes)
          EI                                    ;  IMF ← 1 (enables interrupt service)
```

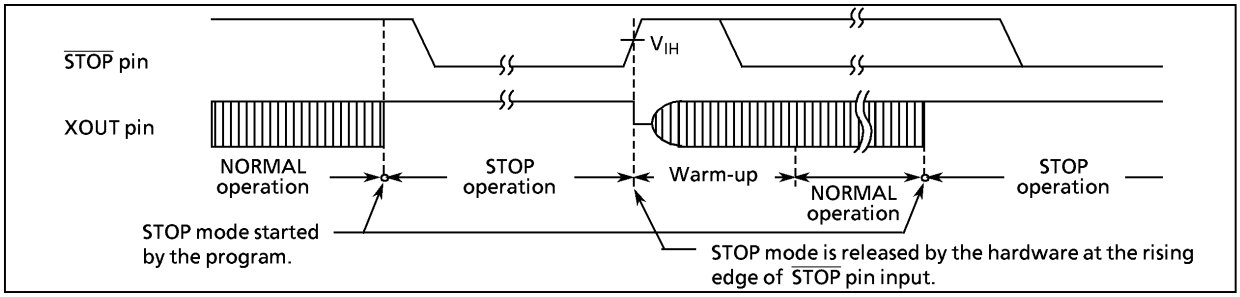


Figure 1-17. Edge-sensitive Release Mode

STOP mode is released by the following sequence:

① In the dual-clock mode. 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 Normal 1, 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 form the NORMAL mode. (If the initial XTEN of 87CH48 is set to "1" by mask option, they start form the NORMAL2 mode In case of 87PH48, starts form NORMAL1 mode.)

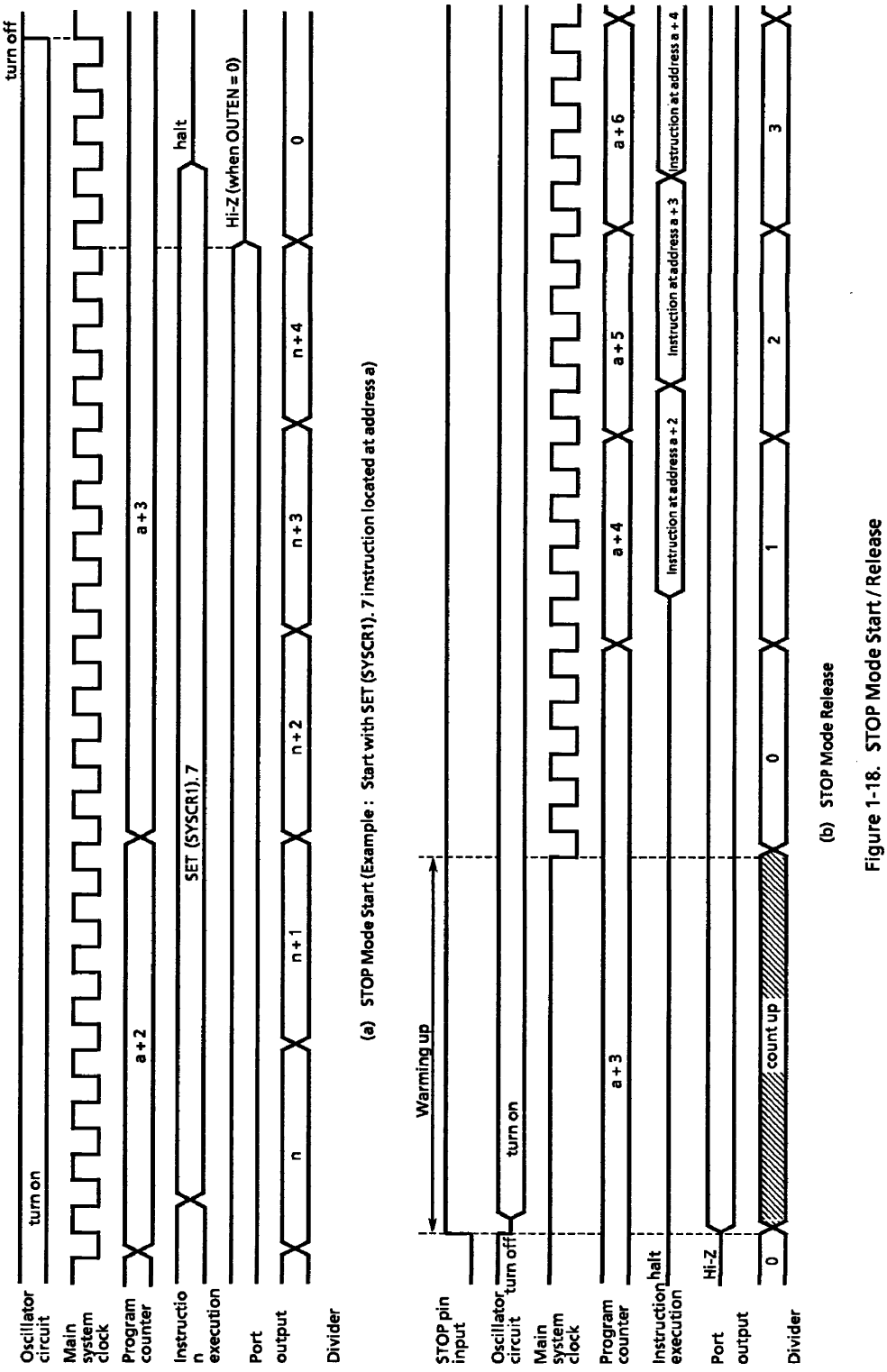


Figure 1-18. STOP Mode Start / Release

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 the 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. The 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
```

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.

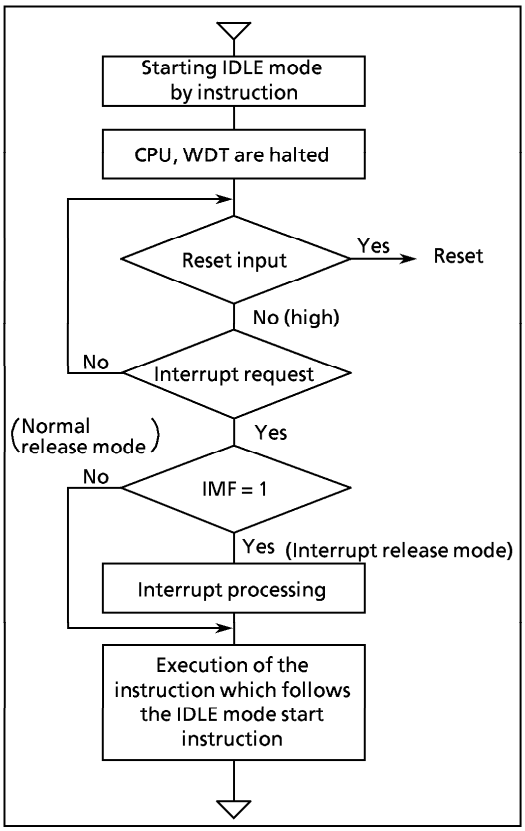


Figure 1-19. IDLE Mode

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 87CH74A/M74A are placed in NORMAL mode.
The 87PM74 is placed in NORMAL1 mode after reset release.

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

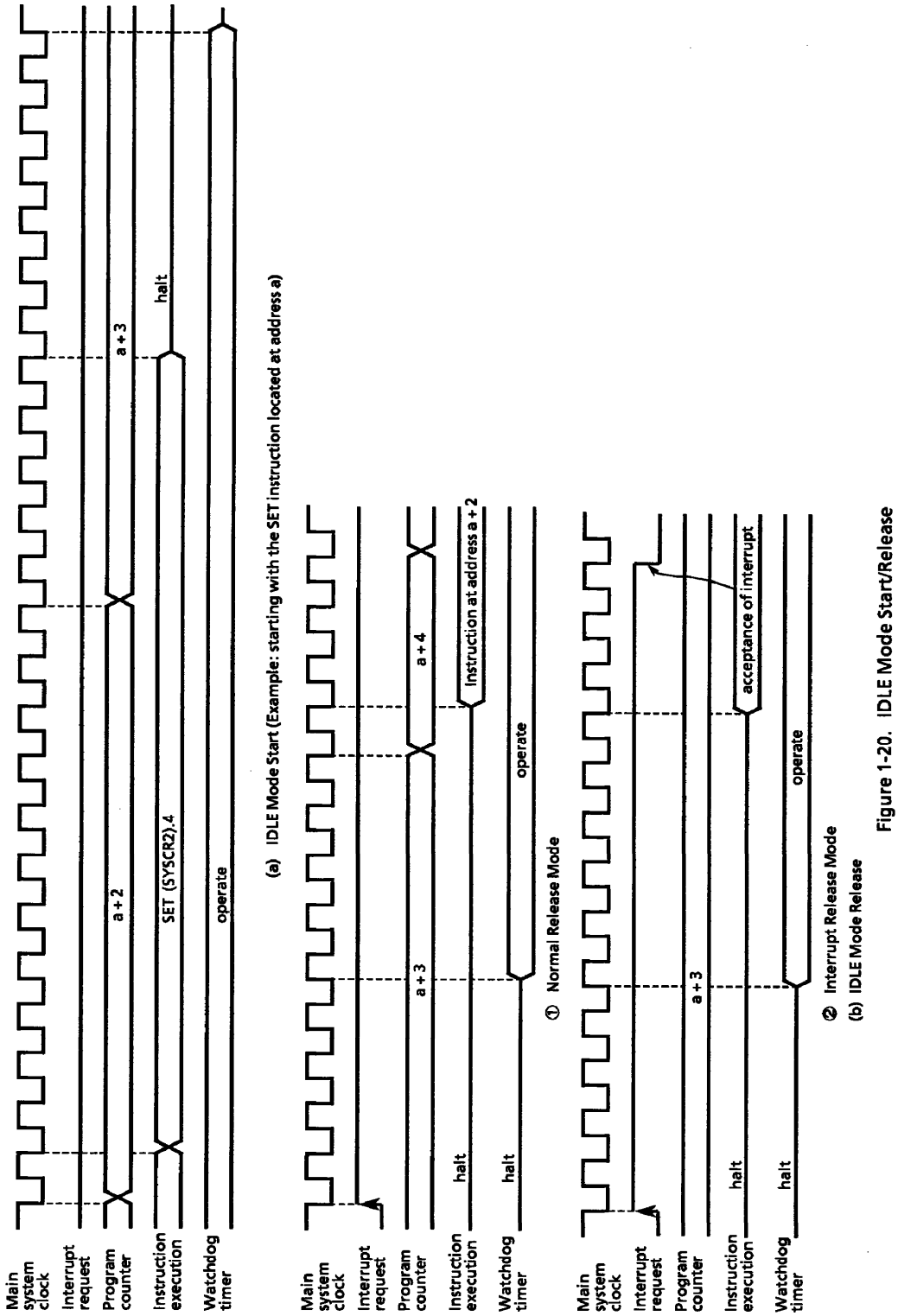


Figure 1-20. IDLE Mode Start/Release

(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.

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.

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.

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 interrupt
        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.

Note1 : After the SYSCK is cleared to "0", the CPU core operate using low frequency clock when the main system clock is switching from low frequency clock to high frequency clock.

Note2 : SLOW mode can also be released by setting the RESET pin low, which immediately performs the reset operation. After reset, the 87CH74A/M74A is placed in NORMAL2 mode. (The PM74 is placed in NORMAL1 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 interrupt
        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
```

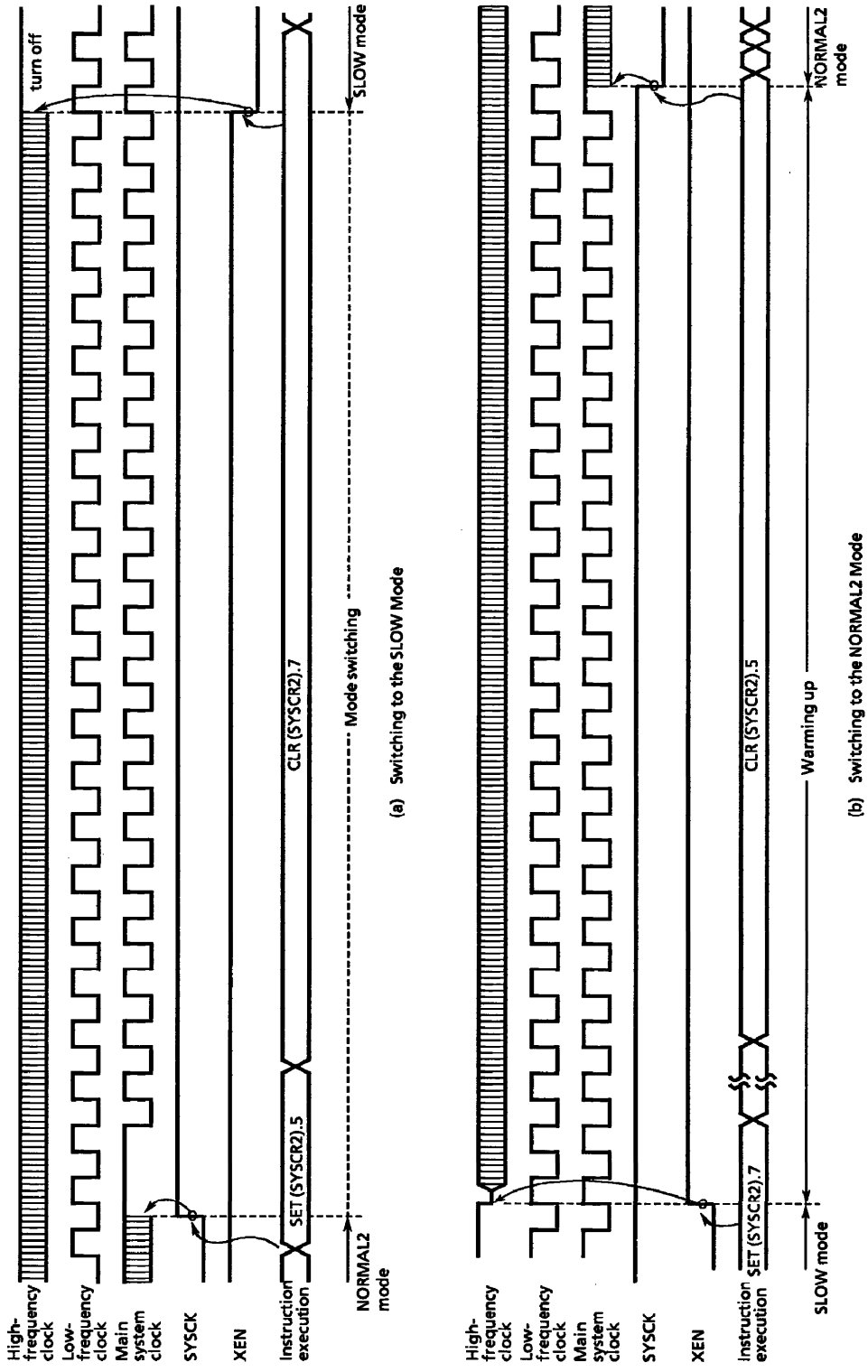


Figure 1-21. Switching between the NORMAL2 and SLOW Modes

1.9 Interrupt Controller

The 87CH74A/M74A 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.

Table 1-2. Interrupt Sources

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
Internal	INTTC4	(8-bit TC4 interrupt)	IMF · EF_{10} = 1	IL_{10}	$FFEA_H$	10
External	INT3	(External interrupt 3)	IMF · EF_{11} = 1	IL_{11}	$FFE8_H$	11
External	INT4	(External interrupt 4)	IMF · EF_{12} = 1	IL_{12}	$FFE6_H$	12
Internal	INTSIO1	(Serial interface1 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_{15-2})

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.

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 intlerrupt to "0").* Thus, interrupt requests can be canceled 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 – IL6←0
```

Example 2 : Reads interrupt latches

```
LD      WA, (IL)                    ; W←ILH, A←ILL
```

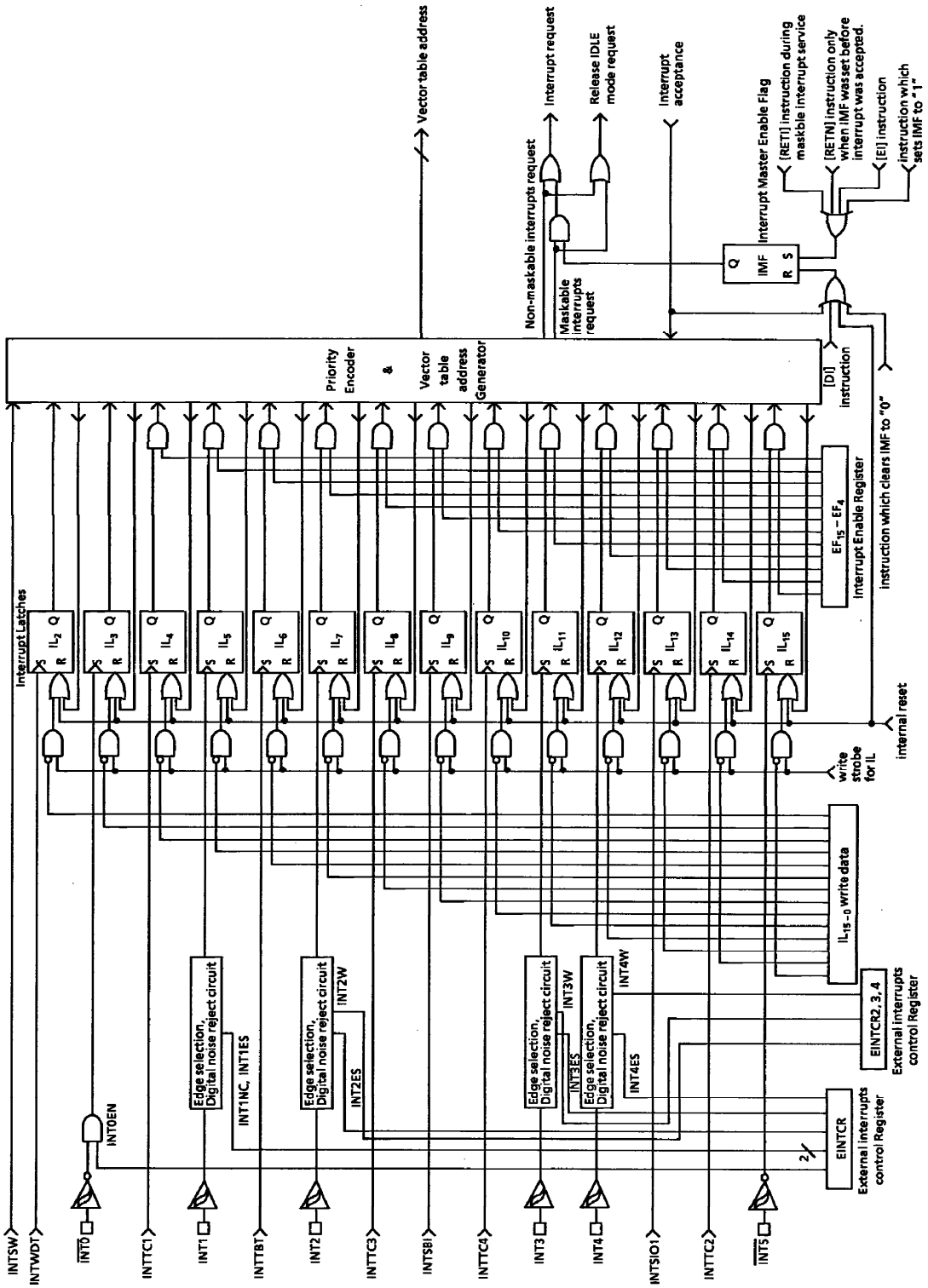


Figure 1-22. Interrupt Controller Block Diagram

Example 3: Tests an interrupt latch

```
TEST    (IL).7          ; if IL7 = 1 then jump
JR      F, SSET
```

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

The interrupt enable register (EIR) enables and disables 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 the individual interrupt enable flags (EF). This register is 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 the 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. The 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} – 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
```

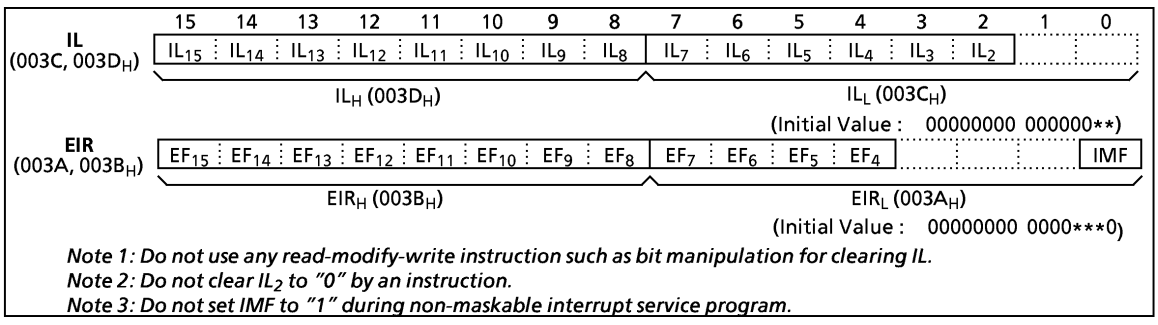


Figure 1-23. Interrupt Latch (IL) and Interrupt Enable Register (EIR)

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 the 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) on the stack. The contents of stack pointer is clearesed by 3.
④ The entry address of the interrupt service program is read from the vector table, 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.

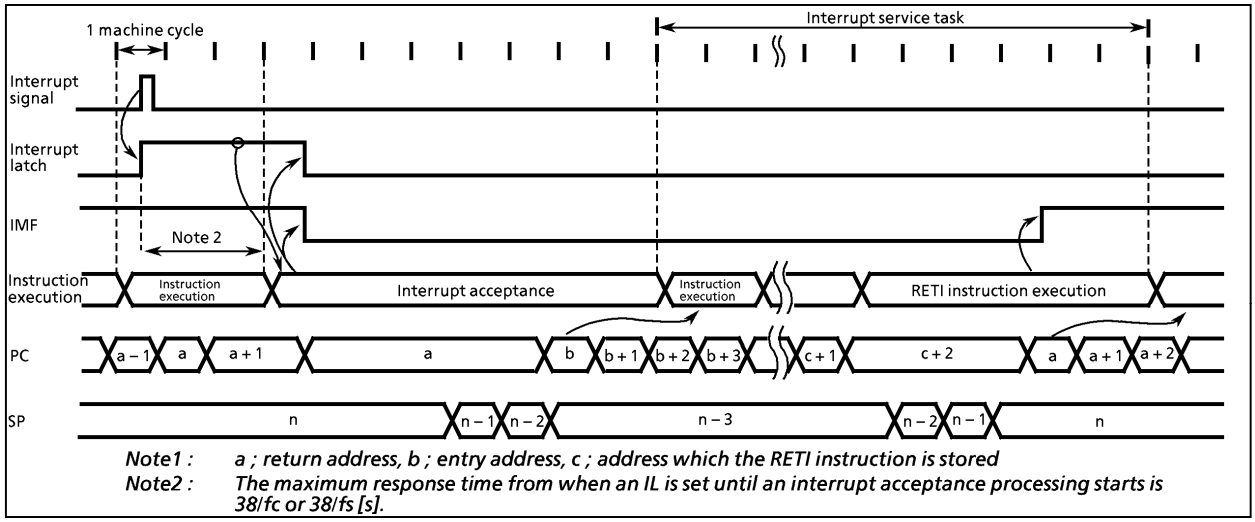
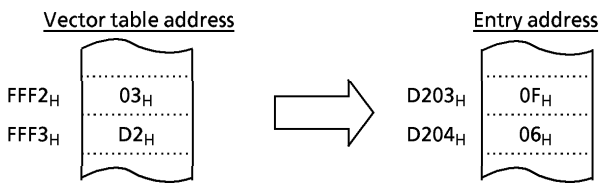


Figure 1-24. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

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 $\overline{\text{INT0}}$ pin must be disabled with the 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 $\overline{\text{INT0}}$ pin input cannot be detected.

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

(2) General-purpose registers 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 changeover:

The 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 (1 μs at 8 MHz)
          interrupt processing
          RETI                    ; Restores bank and Returns
```

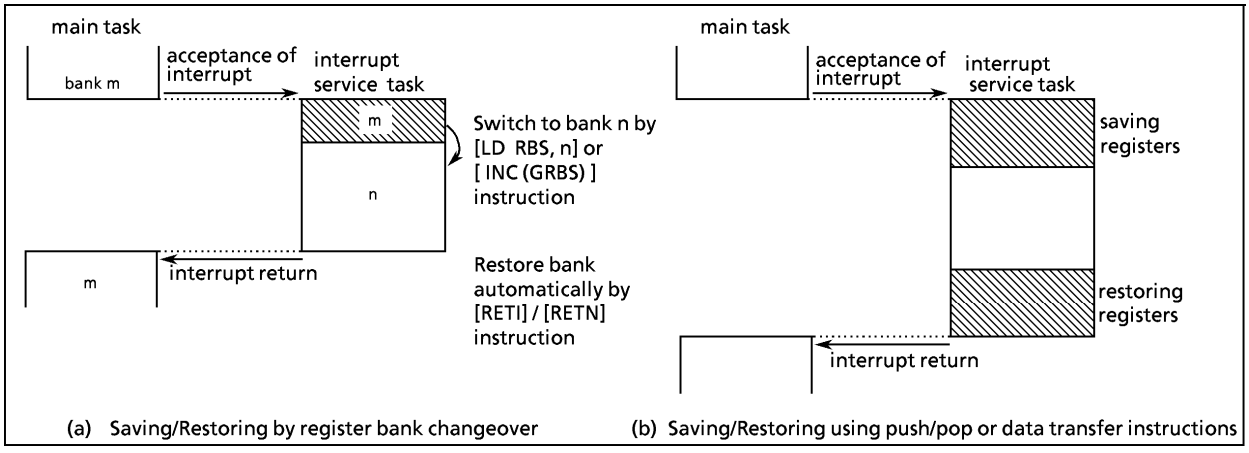


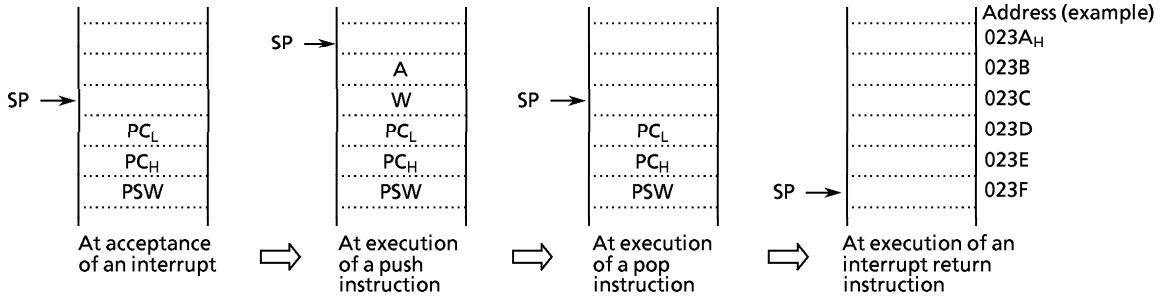
Figure 1-25. Saving/Restoring General-purpose Registers

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 instructions:
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 from interrupt service
```

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 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.

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. the address trap reset is generated in case that an instruction is fetched from RAM or SFR areas.

Note : The fetch data from addresses $7F80_H$ to $7FFF_H$ (test ROM area) for 87CH74A/M74A is not "FF_H".

② Debugging

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

1.9.3 External Interrupts

The 87CH74A/M74A 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 (a). External Interrupts

Source	Pin	Secondary function pin	Enable conditions	Edge			Secondary function pin
				rising	falling	both	
INT0	$\overline{INT0}$	P10	IMF = 1 INT0EN = 1	–	○	–	– (hysteresis input)
INT1	INT1	P11	IMF · EF_5 = 1	INT1ES = 0	INT1ES = 1	–	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of equal to or more than 48/fc or 192/fc [s] are regarded as be signals.
INT2	INT2	P16	IMF · EF_7 = 1 INT2W = 0	INT2ES = 0	INT2ES = 1	–	For falling or rising edge, pulses of less than 7/fc [s] are eliminated as noise. Pulses of equal to or more than 24/fc [s] are regarded as signals.
			IMF · EF_7 = 1 INT2W = 1	–	–	INT2W = 1 Note 2)	Noise cancellation conditions are as listed in Table 1.4. They are applied to the INT2 pin when it is used for both edge interrupts.
INT3	INT3	P15/TC1	IMF · EF_{11} = 1 INT3W = 0	INT3ES = 0	INT3ES = 1	–	For falling or rising edge, 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 pin TC1 (at one edge)
			IMF · EF_{11} = 1 INT3W = 1	–	–	INT3W = 1 Note 2)	Noise cancellation conditions are as listed in Table 1.4. They are applied to the INT3 pin when it is used for both edge interrupts.
INT4	INT4	P17/TC3	IMF · EF_{11} = 1 INT4W = 0	INT4ES = 0	INT4ES = 1	–	For falling or rising edge, 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 pin TC3 (at one edge)
			IMF · EF_{11} = 1 INT4W = 1	–	–	INT4W = 1 Note 2)	Noise cancellation conditions are as listed in Table 1.4. They are applied to the INT4 pin is used for both edge interrupts. To detect remote control signals using timer 3 in capture mode, the INT4 pin is used for both edge interrupts.
INT5	$\overline{INT5}$	$\overline{P20/STOP}$	IMF · EF_{15} = 1	–	○	–	– (hysteresis input)

Note 1 : *The noise rejection function is turned off for $\overline{INT0}$, INT1, INT2, INT3, INT4, $\overline{INT5}$ used in SLOW and SLEEP modes. Also, the noise reject times are not constant for pulses input while transiting between operating modes (NORMAL2↔SLOW)*

Note 2 : *To detect the edge at which an interrupt is generated, read bit 6 (INT2EDT) in EINTCR2 (#0025$_H$), bit 6 (INT3EDT) in EINTCR3 (#0026$_H$), bit 6 (INT4EDT) in EINTCR4 (#0024$_H$), that is, at the beginning of the interrupt processing routine. INT2EDT, INT3EDT, INT4EDT is valid only for both-edge interrupts (INT2W = 1, INT3W = 1, INT4W = 1). INT2EDT, INT3EDT, INT4EDT is set to 1 by an interrupt as the non-selected edge; cleared to 0 after read automatically.*

For both-edge interrupts, rising or falling edge is selected by setting/modifying bit 2 (INT2ES), bit 3 (INT3ES), bit 4 (INT4ES) in EINTCR (#0037$_H$).

When rising edge is selected (INT3ES = 0), bit 6 in INT3EDT (#0026$_H$) is set to 1 when a falling edge is detected at the INT3 pin. (That is, remains 0 if rising edge is detected.)

When falling edge is selected (INT3ES = 1), bit 6 in INT3EDT; #0026$_H$ is set to 1 when a rising edge is detected at the INT3 pin. (That is, remains 0 at falling edge.)

Note 3 : *The noise rejection function is also affected for timer/counter input (TC1 and TC3 pins).*

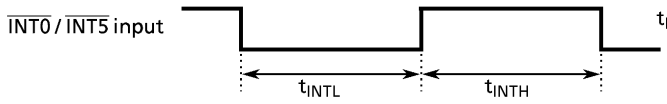
Note 4 : *Noise cancellation/pulse receive conditions for timer/counter are as described below:*

① TC1 pin : When the P15 pin is used for TC1 input , INT3W must be cleared to "0". Do not change INT3W to "1".

② TC3 pin : When INTSW = 0, less than 7/fc [s] (noise cancellation) and 24/fc [s] or more (pulse receive). For when INT3W = 1, see Table 1.4.

Note 5 : *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] (when INT1NC = 1)
   - 193/fc [s] (when INT1NC = 0)
2. INT2 pin
   - 25/fc [s] (when #0025$_H$: INT2W = 0, falling or rising edge)
   - 25/fc [s] (when #0025$_H$: INT2W = 1, and NCS (0, 0, 0))
   - $(2^6/fc) \times 8.5 + 19/fc$ [s] (when #0025$_H$: INT2W = 1, and NCS (0, 0, 1))
   - $(2^7/fc) \times 8.5 + 19/fc$ [s] (when #0025$_H$: INT2W = 1, and NCS (0, 1, 0))
   - $(2^8/fc) \times 8.5 + 19/fc$ [s] (when #0025$_H$: INT2W = 1, and NCS (0, 1, 1))
   - $(2^9/fc) \times 8.5 + 19/fc$ [s] (when #0025$_H$: INT2W = 1, and NCS (1, 0, 0))
   - $(2^{10}/fc) \times 8.5 + 19/fc$ [s] (when #0025$_H$: INT2W = 1, and NCS (1, 0, 1))
   - $(2^{11}/fc) \times 8.5 + 19/fc$ [s] (when #0025$_H$: INT2W = 1, and NCS (1, 1, 0))
   - $(2^{12}/fc) \times 8.5 + 19/fc$ [s] (when #0025$_H$: INT2W = 1, and NCS (1, 1, 1))
3. INT3 pin
   - 25/fc [s] (when #0026$_H$: INT3W = 0, falling or rising edge)
   - 25/fc [s] (when #0026$_H$: INT3W = 1, and NCS (0, 0, 0))
   - $(2^6/fc) \times 8.5 + 19/fc$ [s] (when #0026$_H$: INT3W = 1, and NCS (0, 0, 1))
   - $(2^7/fc) \times 8.5 + 19/fc$ [s] (when #0026$_H$: INT3W = 1, and NCS (0, 1, 0))
   - $(2^8/fc) \times 8.5 + 19/fc$ [s] (when #0026$_H$: INT3W = 1, and NCS (0, 1, 1))
   - $(2^9/fc) \times 8.5 + 19/fc$ [s] (when #0026$_H$: INT3W = 1, and NCS (1, 0, 0))
   - $(2^{10}/fc) \times 8.5 + 19/fc$ [s] (when #0026$_H$: INT3W = 1, and NCS (1, 0, 1))
   - $(2^{11}/fc) \times 8.5 + 19/fc$ [s] (when #0026$_H$: INT3W = 1, and NCS (1, 1, 0))
   - $(2^{12}/fc) \times 8.5 + 19/fc$ [s] (when #0026$_H$: INT3W = 1, and NCS (1, 1, 1))
4. INT4 pin
   - 25/fc [s] (when #0024$_H$: INT4W = 0, falling or rising edge)
   - 25/fc [s] (when #0024$_H$: INT4W = 1, and NCS (0, 0, 0))
   - $(2^6/fc) \times 8.5 + 19/fc$ [s] (when #0024$_H$: INT4W = 1, and NCS (0, 0, 1))
   - $(2^7/fc) \times 8.5 + 19/fc$ [s] (when #0024$_H$: INT4W = 1, and NCS (0, 1, 0))
   - $(2^8/fc) \times 8.5 + 19/fc$ [s] (when #0024$_H$: INT4W = 1, and NCS (0, 1, 1))
   - $(2^9/fc) \times 8.5 + 19/fc$ [s] (when #0024$_H$: INT4W = 1, and NCS (1, 0, 0))
   - $(2^{10}/fc) \times 8.5 + 19/fc$ [s] (when #0024$_H$: INT4W = 1, and NCS (1, 0, 1))
   - $(2^{11}/fc) \times 8.5 + 19/fc$ [s] (when #0024$_H$: INT4W = 1, and NCS (1, 1, 0))
   - $(2^{12}/fc) \times 8.5 + 19/fc$ [s] (when #0024$_H$: INT4W = 1, and NCS (1, 1, 1))



t_{INTL}, t_{INTH} > tcyc (*Note : tcyc = 4/fm [s]*)

Note 6 : The pulse width (both "H" and "L" level) for input to the INT0 and INT5 pins must be over 1 machine cycle.

Note 7 : When INT0EN = 0, interrupt latch IL_3 is not set even if a falling edge is detected for INT0 pin input.

Note 8 : 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 INT5 (P20/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)
```

Table 1.3 (b) Noise reject condition for INT2, INT3, INT4 (both-edge interrupt)

#0024, #0025, #0026			max. pulse width for noise reject	min. pulse width for immediate signal
NCS2 × 2	NCS1 × 1	NCS0 × 0		
0	0	0	– (histeresis input)	
0	0	1	$(2^6/fc) \times 7 - 6/fc$	$(2^6/fc) \times 8 + 5/fc$
0	1	0	$(2^7/fc) \times 7 - 6/fc$	$(2^7/fc) \times 8 + 5/fc$
0	1	1	$(2^8/fc) \times 7 - 6/fc$	$(2^8/fc) \times 8 + 5/fc$
1	0	0	$(2^9/fc) \times 7 - 6/fc$	$(2^9/fc) \times 8 + 5/fc$
1	0	1	$(2^{10}/fc) \times 7 - 6/fc$	$(2^{10}/fc) \times 8 + 5/fc$
1	1	0	$(2^{11}/fc) \times 7 - 6/fc$	$(2^{11}/fc) \times 8 + 5/fc$
1	1	1	$(2^{12}/fc) \times 7 - 6/fc$	$(2^{12}/fc) \times 8 + 5/fc$

Note : In SLOW mode, set (NCS × 2, 1, 0) = (0, 0, 0) .
In SLOW mode, the digital noise filter in the above table is disabled.

EINTCR (0037_H)

7	6	5	4	3	2	1	0
INT1NC	INT0EN		INT4ES	INT3ES	INT2ES	INT1ES	

(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 only when IMF = 0. 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 and INT3ES, INT4ES during NORMAL1/2 mode, clear interrupt latches of external interrupt inputs (INT2 and INT3, 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 and INT3ES, 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 and INT3) after 8 machine cycles from the time of rewriting to change to interrupt enable state. Finally, state 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                       ↑ 8 machine cycles ↓
NOP
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 (a). External Interrupt Control Register (1)

External Interrupt Control Register 2

EINTCR2 (0025_H)

7	6	5	4	3	2	1	0
INT2W	INT2 EDT	NCS2			INT2 EDT		

(initial value 0000 00**)

INT2W	INT2 both edge selection	0 : Refer to INT2ES 1 : Both edge detection	R/W
INT2EDT	Flag indicating an interrupt at selected edge/non-selected edge, when INT2W = 1 (for both-edge interrupts)	0 : Interrupt at selected edge or no interrupt 1 : Interrupt at non-selected edge	R
NCS2	Noise cancellation time select for INT2 digital noise filter (valid only when INT2W = 1)	000 : No noise cancellation 001 : Cancels (2^6/fc × 7 – 6/fc) as noise. 010 : Cancels (2^7/fc × 7 – 6/fc) as noise. 011 : Cancels (2^8/fc × 7 – 6/fc) as noise. 100 : Cancels (2^9/fc × 7 – 6/fc) as noise. 101 : Cancels (2^{10}/fc × 7 – 6/fc) as noise. 110 : Cancels (2^{11}/fc × 7 – 6/fc) as noise. 111 : Cancels (2^{12}/fc × 7 – 6/fc) as noise.	R/W
INT2DET	INT2 interrupt detection flag	0 : No interrupt 1 : Interrupt	R

Note 1 : INT2EDT and NCS2 are valid only when the INT2W bit in EINTCR2 (#0025_H) is set to 1. Therefore, when INT2W = 0, the digital noise filter set by the NCS2 bit is disabled.

Note 2 : Do not changing the contents of INT2ES (bit 2 in #0037_H) when INT2W is set to 1 (both-edge detention). If changing the contents of INT2ES during INT2W is set to 1, according to Note 3/4/5 at Figure 1-26 (a).

External Interrupt Control Register 3

EINTCR3 (0026_H)

7	6	5	4	3	2	1	0
INT3W	INT3 EDT	NCS3			INT3 EDT		

(initial value 0000 00**)

INT3W	INT3 both edge selection	0 : Refer to INT3ES 1 : Both edge detection	R/W
INT3EDT	Flag indicating an interrupt at selected edge/non-selected edge, when INT3W = 1 (for both-edge interrupts)	0 : Interrupt at selected edge or no interrupt 1 : Interrupt at non-selected edge	R
NCS3	Noise cancellation time select for INT3 digital noise filter (valid only when INT3W = 1)	000 : No noise cancellation 001 : Cancels (2^6/fc × 7 – 6/fc) as noise. 010 : Cancels (2^7/fc × 7 – 6/fc) as noise. 011 : Cancels (2^8/fc × 7 – 6/fc) as noise. 100 : Cancels (2^9/fc × 7 – 6/fc) as noise. 101 : Cancels (2^{10}/fc × 7 – 6/fc) as noise. 110 : Cancels (2^{11}/fc × 7 – 6/fc) as noise. 111 : Cancels (2^{12}/fc × 7 – 6/fc) as noise.	R/W
INT3DET	INT3 interrupt detection flag	0 : No interrupt 1 : Interrupt	R

Note 1 : INT3EDT and NCS3 are valid only when the INT3W bit in EINTCR3 (#0026_H) is set to 1. Therefore, when INT3W = 0, the digital noise filter set by the NCS3 bit is disabled.

Note 2 : Do not changing the contents of INT3ES (bit 3 in #0037_H) when INT3W is set to 1 (both-edge detention). If changing the contents of INT3ES during INT2W is set to 1, according to Note 3/4/5 at Figure 1-26 (a).

External Interrupt Control Register 4

EINTCR4 (0024_H)

7	6	5	4	3	2	1	0
INT4W	INT4 EDT	NCS4			INT4 EDT		

(initial value 0000 00**)

INT4W	INT4 both edge selection	0 : Refer to INT4ES 1 : Both edge detection	R/W
INT4EDT	Flag indicating an interrupt at selected edge/non-selected edge, when INT4W = 1 (for both-edge interrupts)	0 : Interrupt at selected edge or no interrupt 1 : Interrupt at non-selected edge	R
NCS4	Noise cancellation time select for INT4 digital noise filter (valid only when INT4W = 1)	000 : No noise cancellation 001 : Cancels (2^6/fc × 7 – 6/fc) as noise. 010 : Cancels (2^7/fc × 7 – 6/fc) as noise. 011 : Cancels (2^8/fc × 7 – 6/fc) as noise. 100 : Cancels (2^9/fc × 7 – 6/fc) as noise. 101 : Cancels (2^{10}/fc × 7 – 6/fc) as noise. 110 : Cancels (2^{11}/fc × 7 – 6/fc) as noise. 111 : Cancels (2^{12}/fc × 7 – 6/fc) as noise.	R/W
INT4DET	INT4 interrupt detection flag	0 : No interrupt 1 : Interrupt	R

Note 1 : INT4EDT and NCS4 are valid only when the INT4W bit in EINTCR4 (#0024_H) is set to 1. Therefore, when INT4W = 0, the digital noise filter set by the NCS4 bit is disabled.

Note 2 : Do not changing the contents of INT4ES (bit 4 in #0037_H) when INT4W is set to 1 (both-edge detention). If changing the contents of INT4ES during INT4W is set to 1, according to Note 3/4/5 at Figure 1-26 (a).

Figure 1-26 (b). External Interrupt Control Register (2)

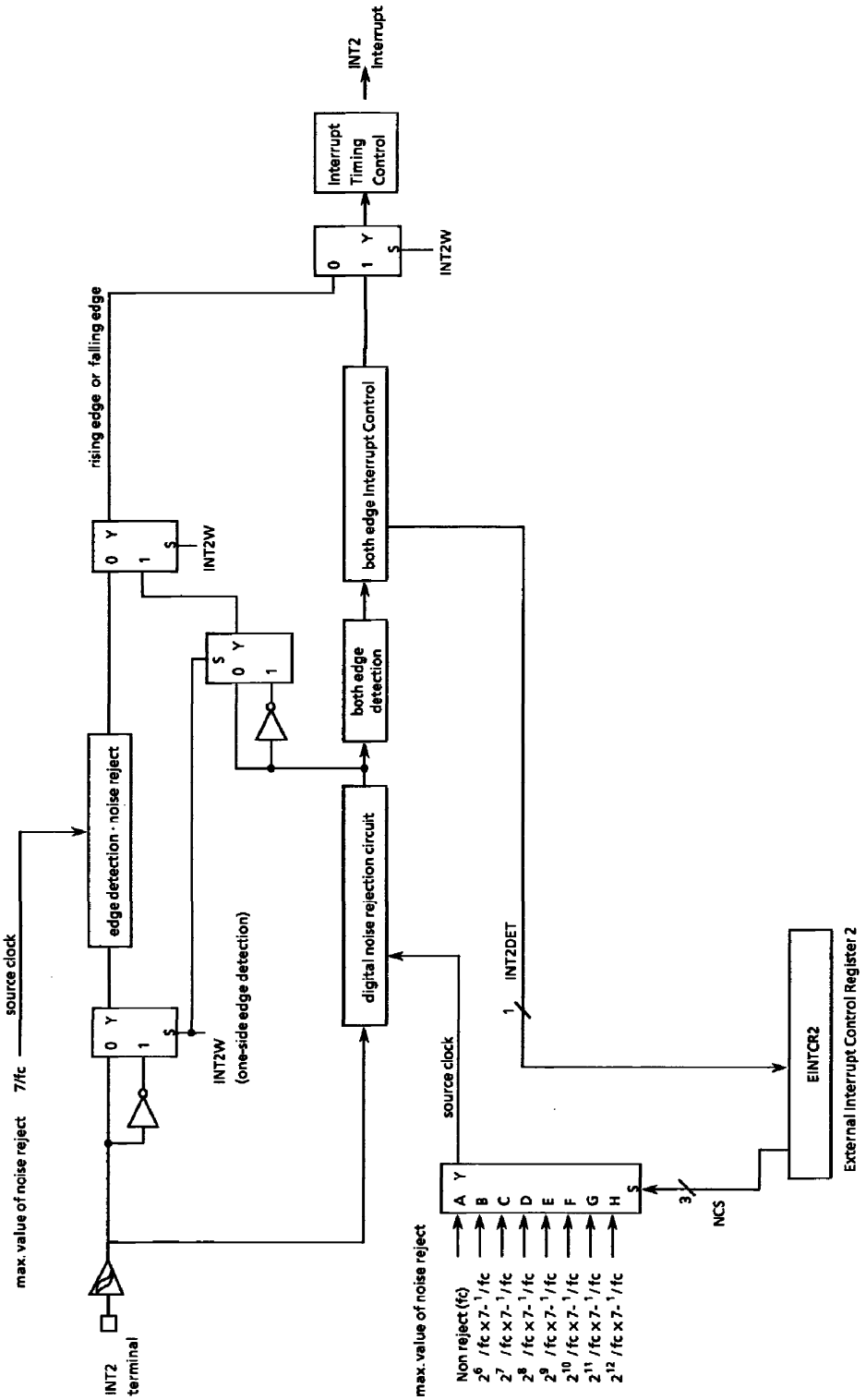


Figure1-26 (C) INT2 both edge detection / one-side edge detection

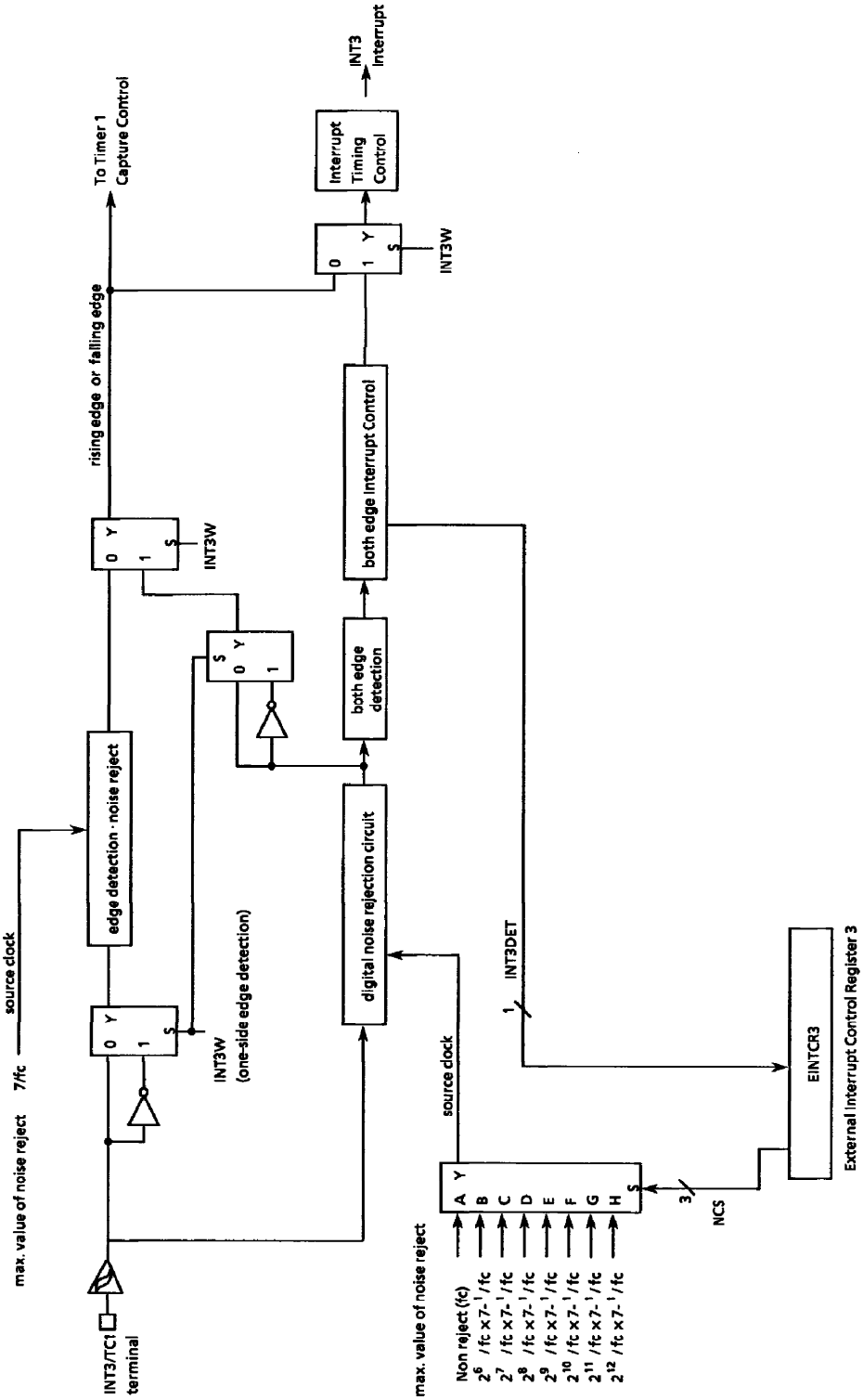


Figure1-26 (d) INT3/TC1 both edge detection / one-side edge detection

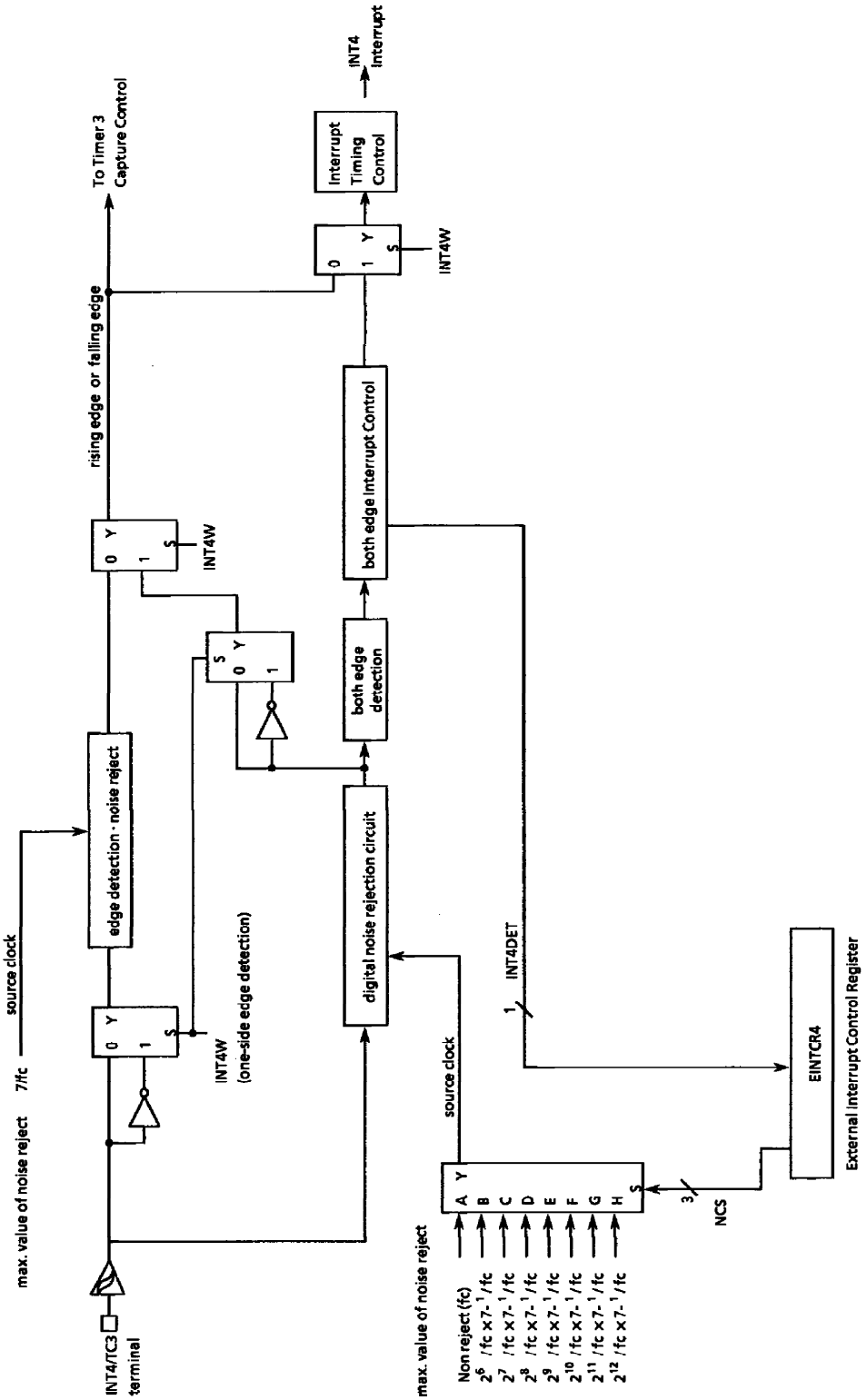


Figure1-26 (e) INT4/TC3 both edge detection / one-side edge detection

Notes on the usage of INT2, INT3, INT4 pin (external interrupt)
(The functions of only INT2 are described. INT3 and INT4 have the same functions as INT2)

When using the INT2 pin for one edge (either rising or falling)

Note 1 : An interrupt generated from the INT2 pin can be detected by reading the interrupt latch (IL7).

Note 2 : The greatest care must be taken in setting or rewriting the external interrupt control register (EINTCR : 0037H).
For details, see Figure 1-26. (a) note 2, 3, 4, 5 and 6 in accordance with the using instructions.

When using the INT2 pin for both edges (rising and falling)

Note 1 : When using the INT2 pin for both edges (rising or falling), set bit 7 (INT3W) in EINTCR2 ($\#0025_H$) to 1.

Note 2 : To detect the edge at which an interrupt is generated, read bit 6 (INT2EDT) in EINTCR2 ($\#0025_H$), that is, at the beginning of the interrupt processing routine.

Note 3 : INT2EDT is valid only for both-edge interrupts (INT2W = 1). INT2EDT is set to 1 by an interrupt as the non-selected edge; cleared to 0 after read automatically.
For both-edge interrupts, rising or falling edge is selected by setting/modifying bit 2 (INT2ES) in EINTCR ($\#0037_H$). When rising edge is selected (INT2ES = 0), bit 6 in INT2EDT ($\#0025_H$) is set to 1 when a falling edge is detected at the INT2 pin. (That is, remains 0 if rising edge is detected.) When falling edge is selected (INT2ES = 1), bit 6 in INT2EDT: $\#0025_H$ is set to 1 when a rising edge is detected at the INT2 pin. (That is, remains 0 at falling edge).

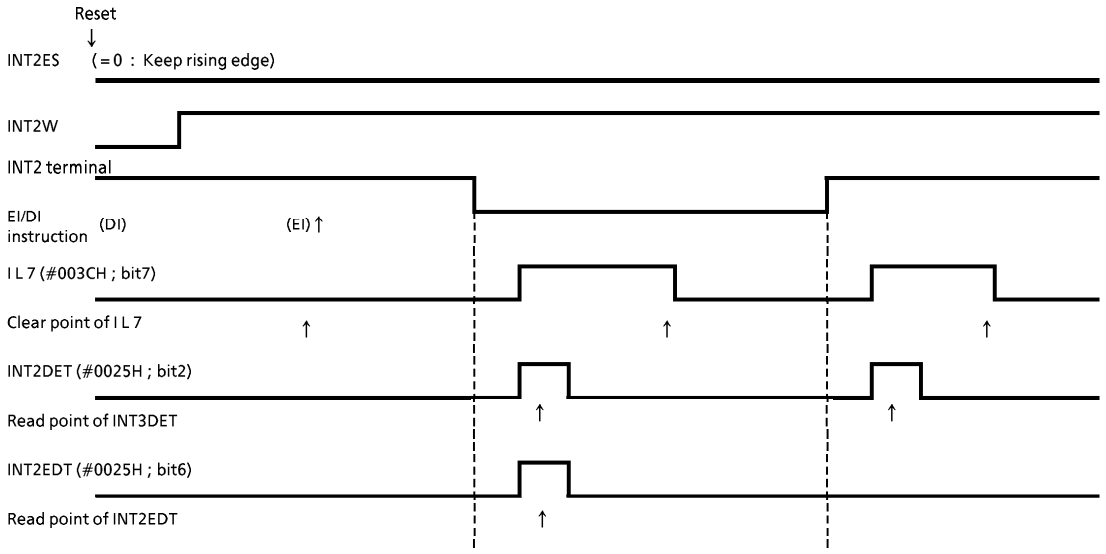
Note 4 : The greatest care must be taken in setting or rewriting the external interrupt control register (EINTCR : 0037H).
For details, see Figure 1-26. (a) note 2, 3, 4, 5 and 6 in accordance with the using instructions.

Operation description for INT2 (both-edge interrupt) in use:

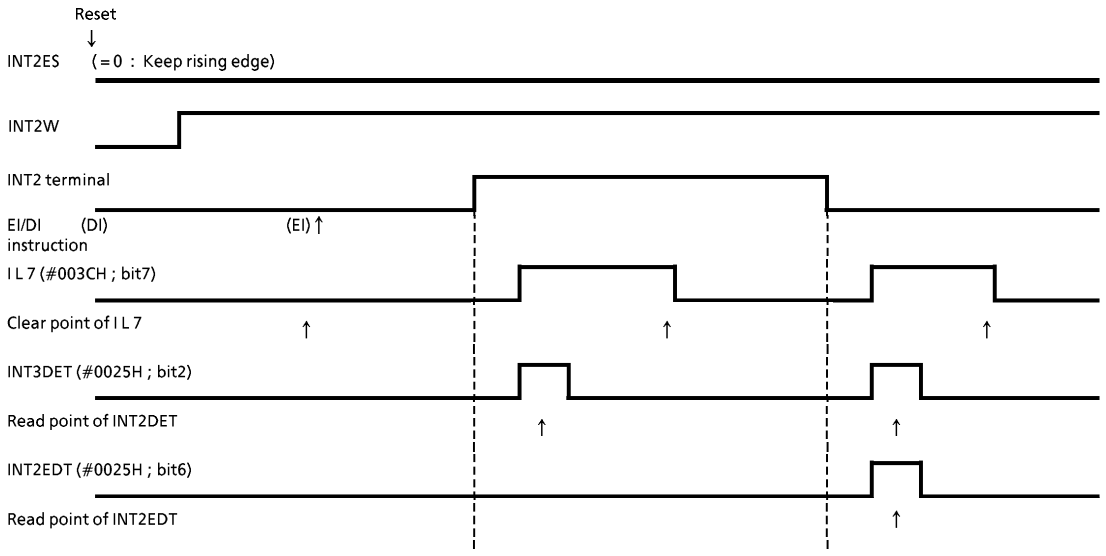
1. Operation without setting/modifying external interrupt control register (EINTCR) after reset:

   For both-edge interrupts, rising edge is selected (INT2ES = 0) and fixed.

   1) Case1 : When the initial state of the INT2 pin is high after reset:

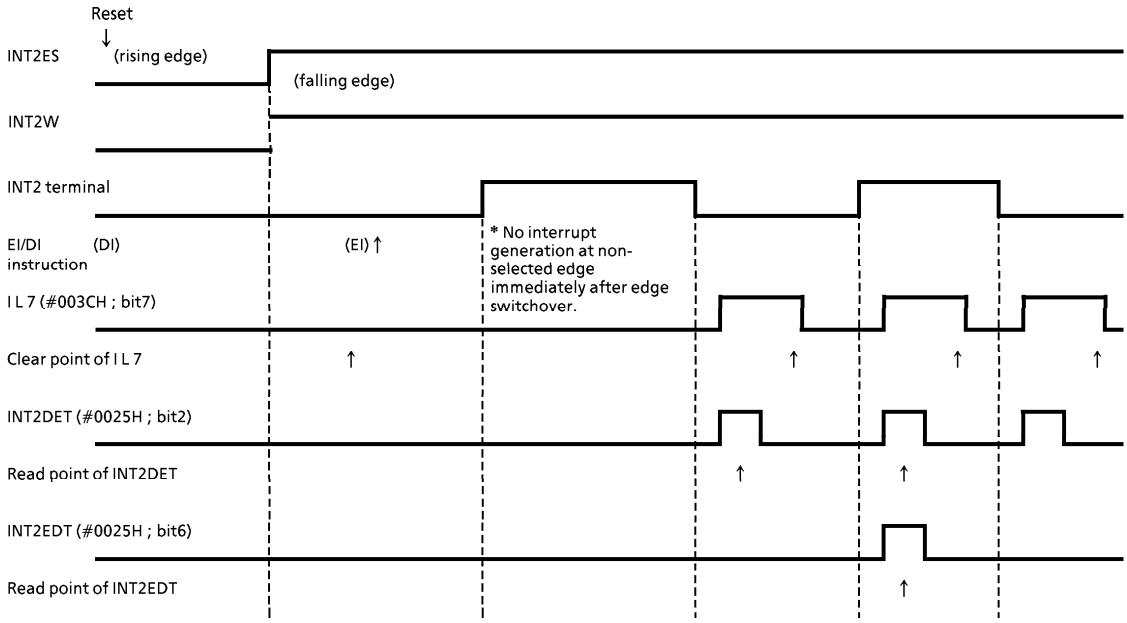


   2) Case2 : When the initial state of the INT2 pin is low after reset:

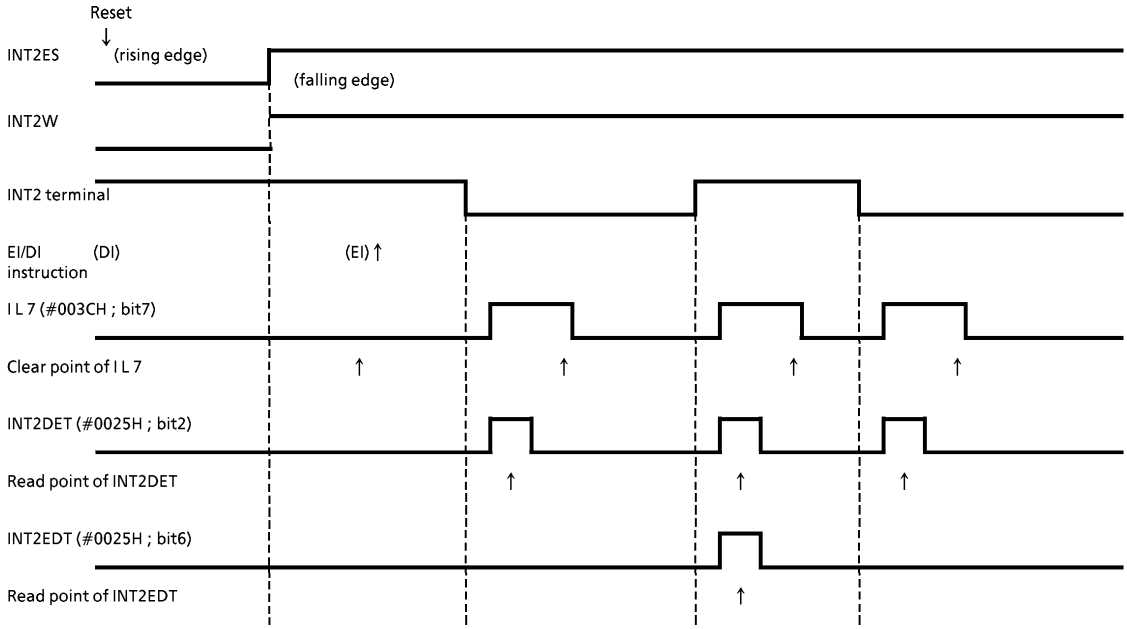


2. Operation with setting/modifying external interrupt control register (EINTCR) after reset:

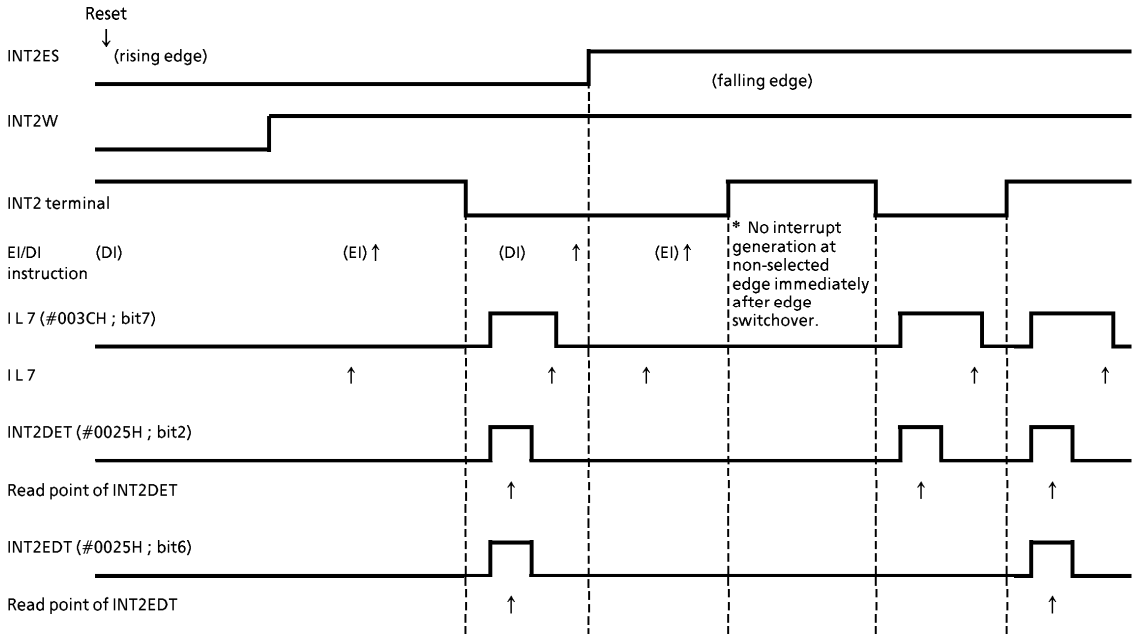
1) Case3 : When the initial state of the INT2 pin is low after reset/low at edge switchover from rising to falling:



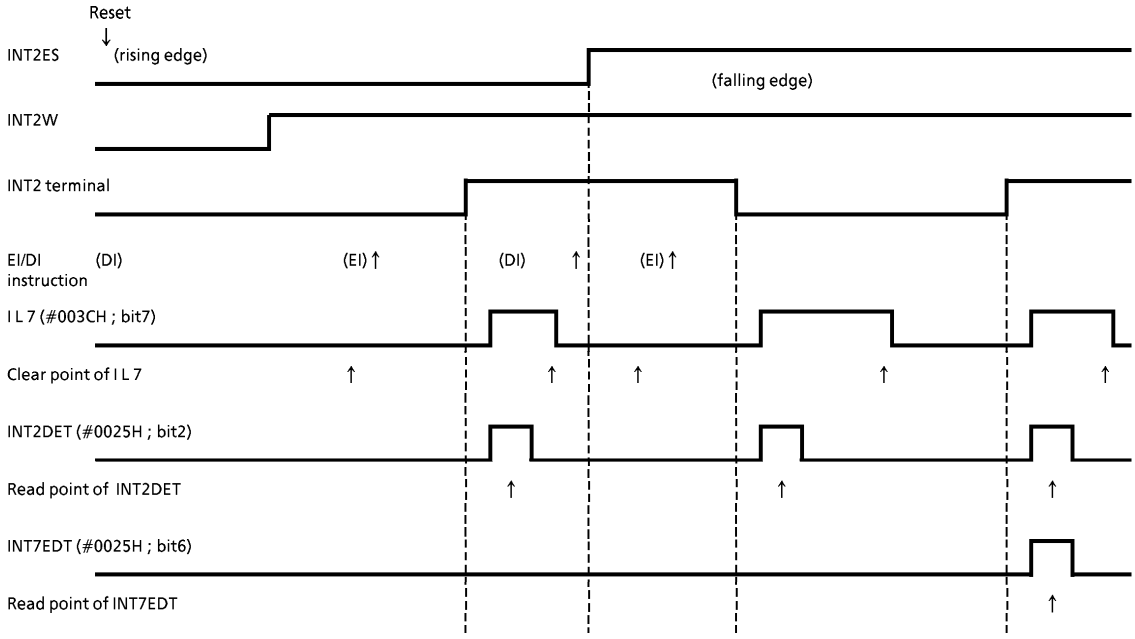
2) Case4 : When the initial state of the INT2 pin is high after reset/high at edge switchover from rising to falling:



3) Case5 : Case 5: When the initial state of the INT2 pin is high after reset/low at edge switchover from rising to falling:



4) Case6 : When the initial state of the INT2 pin is low after reset/high at edge switchover from rising to falling:



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 as 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

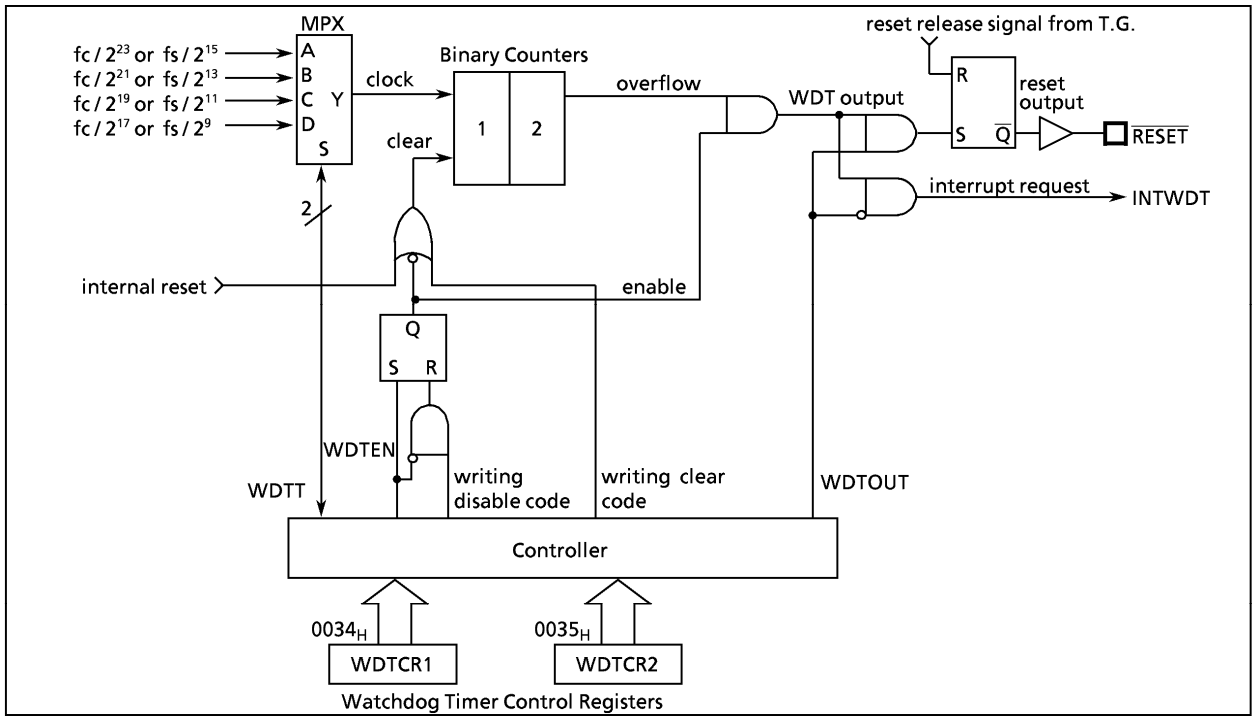


Figure 1-27. Watchdog Timer Configuration

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 a CPU malfunction occurs for any cause, the watchdog timer output will become active on the rise 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 STOP mode (including warm-up) or IDLE mode, and automatically restarts (continues counting) when STOP/IDLE mode is released.

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

```
        LD      (WDTCR2), 4EH       ; Clears the binary counters
        LD      (WDTCR1), 00001101B ; WDTT←10, WDTOUT←1
        LD      (WDTCR2), 4EH       ; Clears the binary counters
Within 3/4 of WDT                   (always clear immediately after changing WDTT)
detection time
        LD      (WDTCR2), 4EH       ; Clears the binary counters
Within 3/4 of 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 the 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 automaticallyafter STOP or IDLE mode is released.

During disabling the watchdog timer, the binary counters are cleared.

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, 023FH               ; 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] (1.5 μs at fc = 8 MHz). The high-frequency clock oscillator also turns on when a watchdog timer reset is generated in SLOW mode.

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

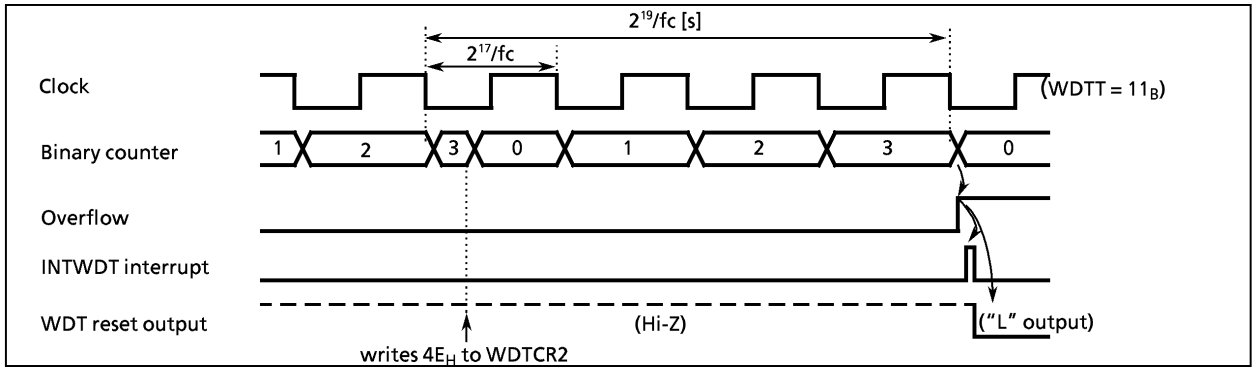


Figure 1-29. Watchdog Timer Interrupt / Reset

1.11 Reset Circuit

The 87CH74A/M74A 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 (12/fc [s.] (1.5 μs at 8 MHz) 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)	($FFFF_H$) - ($FFFE_H$)	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 $\overline{RESET}$ 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 $\overline{RESET}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses $FFFE_H$ - $FFFF_H$.

The $\overline{RESET}$ 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.

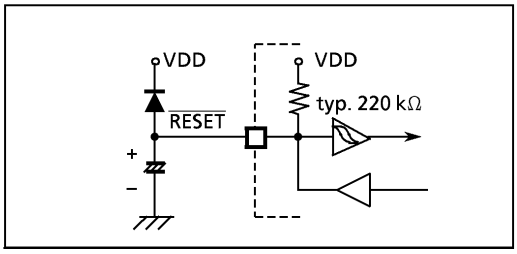
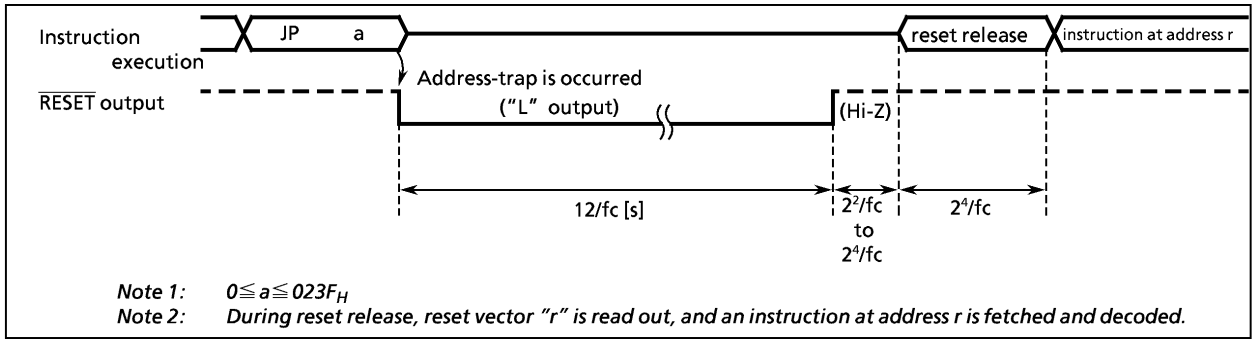


Figure 1-30. Simple Power-on-Reset Circuitry

1.11.2 Address Trap Reset

If a CPU malfunction occurs and an attempt is made to fetch an instruction from the RAM or the SFR area (addresses 87CH74A/M74A: 0040_H - $023F_H$, 87CM74A : 0040_H-$043F_H$, an address-trap-reset will be generated. Then, the $\overline{RESET}$ pin output will go low. The reset time is 12/fc [s] (1.5 μs at 8 MHz).



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 $\overline{RESET}$ pin output goes low from high-impedance. The reset time is 12/fc [s] (1.5 μs at 8 MHz).

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 – $003F_H$, and the DBR to addresses $0F80_H$ – $0FFF_H$.
Figure 2-1 shows the 87CH74A/M74A SFRs and DBRs.

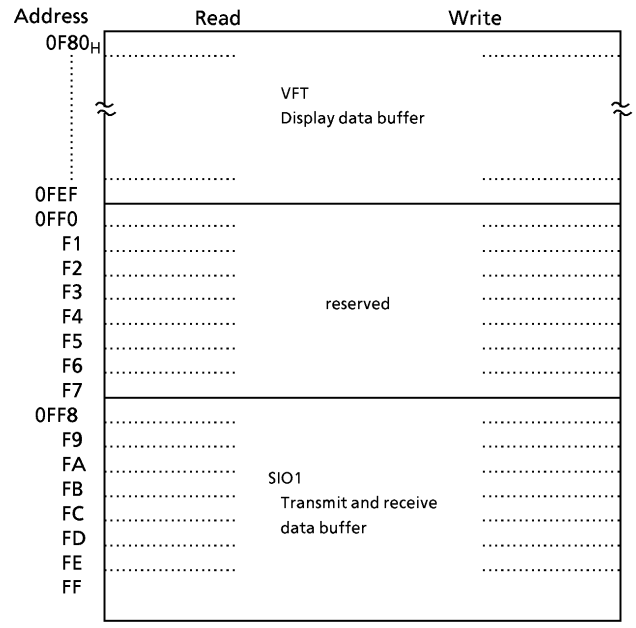
Address	Read	Write
0000_H	P0 port	
01	P1 Port	
02	P2 Port	
03	P3 Port	
04	P4 Port	
05	P5 Port	
06	P6 Port	
07	P7 Port	
08	P8 Port	
09	P9 Port	
0A	–	P0CR (P0 I/O control)
0B	–	P1CR (P1 I/O control)
0C	–	P4CR (P4 I/O control)
0D	–	P5CR (P5 I/O control)
0E	ADCCR (A/D converter control)	
0F	ADCDR (A/D conv. register)	
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	PD port	
1E	–	
1F	–	

Address	Read	Write
0020_H		SBICR1 (SBI control 1)
21	SBIDBR (SBI Data Buffer)	
22		I^2CAR (I^2C bus address)
23	SBISR (SBI status)	SBICR2 (SBI control 2)
24	EINTCR4 (External interrupt control 4)	
25	EINTCR2 (External interrupt control 2)	
26	EINTCR3 (External interrupt control 3)	
27	SIO1SR (SIO status)	SIO1CR1 (SIO1 control 1)
28		SIO1CR2 (SIO1 control 2)
29	VFTSR (VFT status)	VFTCR1 (VFT control 1)
2A		VFTCR2 (VFT control 2)
2B		P3CR (P3 I/O control)
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 (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

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. (a) SFR & DBR



(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. (b) SFR & DBR

2.2 I/O Ports

The 87CH74A/M74A each have 13 parallel input/output ports (89pins) each as follows:

	Primary Function	Secondary Functions
Port P0	8-bit I/O port	Serial port input/output
Port P1	8-bit I/O port	External interrupt input, timer/counter input, 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	4-bit I/O port	Serial bus interface
Port P4	8-bit I/O port	Analog input
Port P5	8-bit I/O port	Analog input
Port P6	8-bit I/O port	VFT output
Port P7	8-bit Outoput port	VFT output
Port P8	8-bit output port	VFT output
Port P9	8-bit I/O port	VFT output
Port PD	5-bit I/O port	VFT output

Ports P0, P1, P2, P3, P4, P5, P6, P7, P8, P9 and PD can also use secondary function.
Each output port contains a latch, which holds the output data. Input ports excluding P4 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 changes in the S2 state of the write cycle during execution of the instruction which writes to an I/O port.

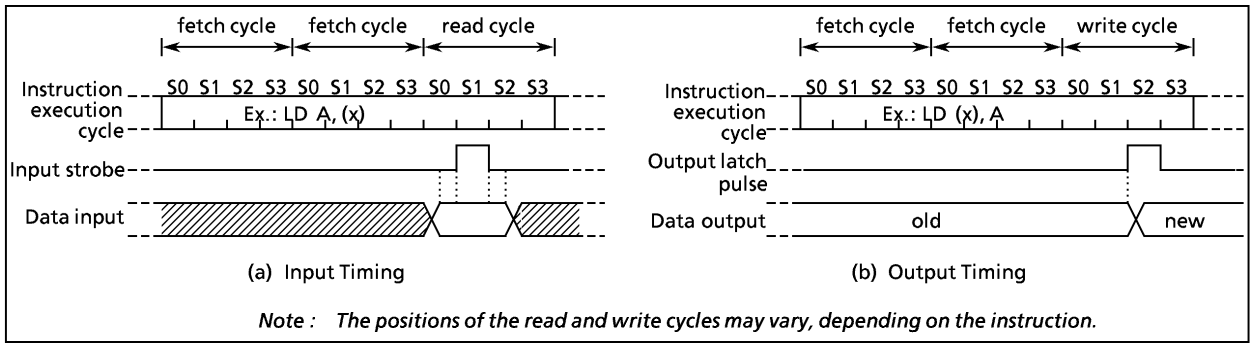


Figure 2-2. Input/Output Timing (Example)

When reading an I/O port except programmable I/O ports P0 and P1, 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)

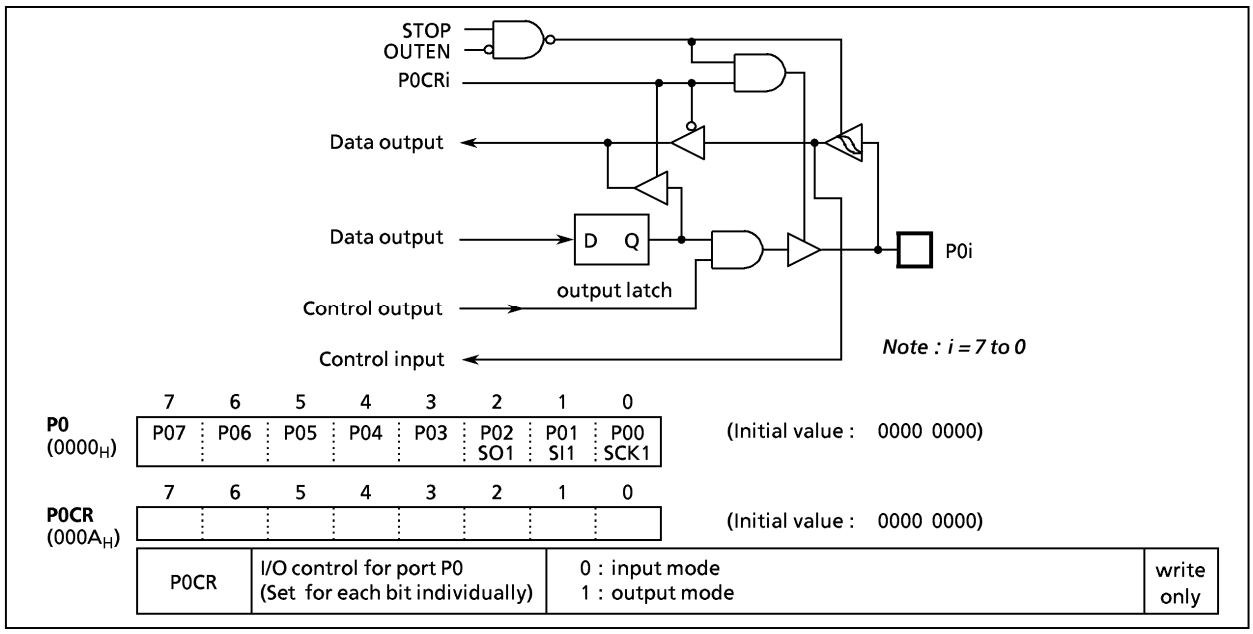
2.2.1 Port P0 (P07 - P00)

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 P0CR contents. Therefore initial output data should be written into the output latch before setting P0CR.

Note1 : Ports set to the input mode read the pin states. When input pin and output in 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.

Note2 : 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.)



	7	6	5	4	3	2	1	0	
P0 (0000_H)	P07	P06	P05	P04	P03	P02 SO1	P01 SI1	P00 SCK1	(Initial value : 0000 0000)
P0CR ($000A_H$)									(Initial value : 0000 0000)

P0CR	I/O control for port P0 (Set for each bit individually)	0 : input mode 1 : output mode	write only

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, 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/output, and a divider output. When used as a 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 P15, P16, P17 should be used as external interrupt inputs, timer/counter input, or input ports. The interrupt latch is set on 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, the pin P10 ($\overline{INT0}$) is configured as an input port P10.

Note1 :	*Ports set to the input mode read the pin states. When input pin and output in 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.*
Note2 :	*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.)*

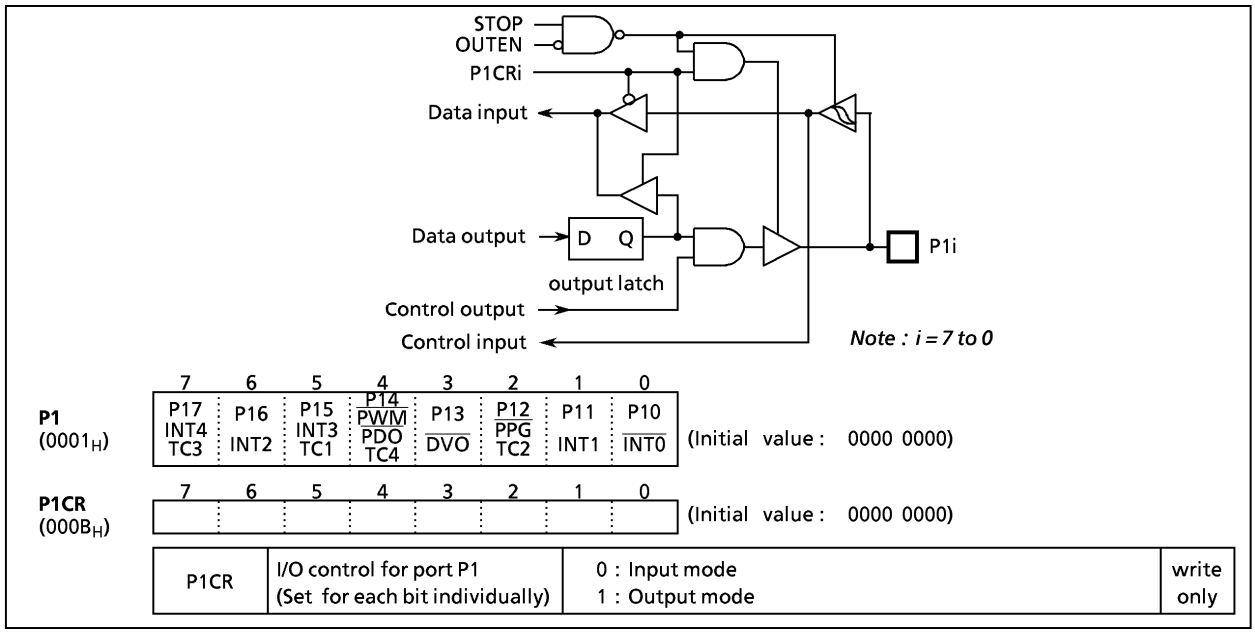


Figure 2-4. Port P1 and P1CR

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 - 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 the 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 the P20 pin 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 for port P2 is executed, bits 7 to 3 in P2 read in as undefined data.

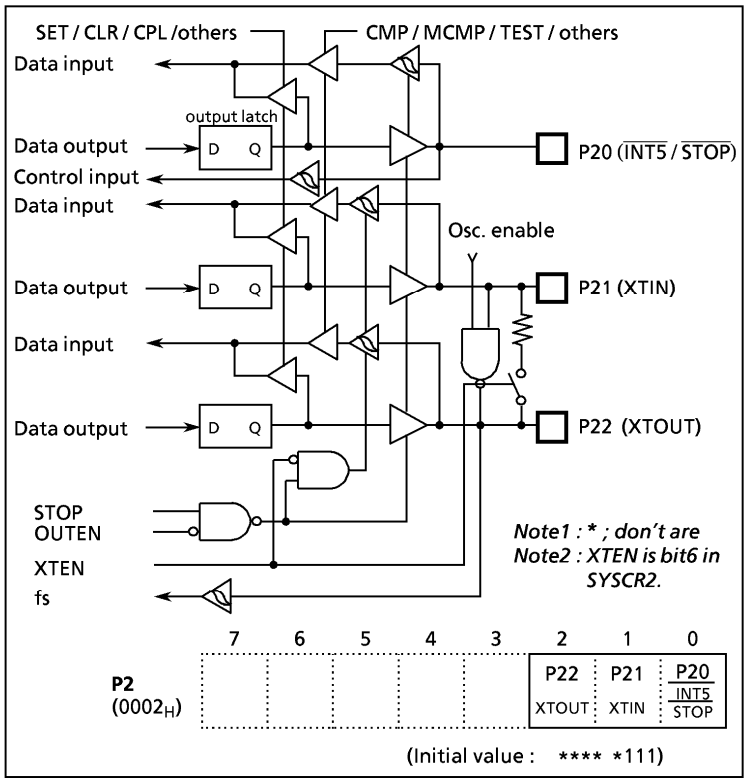


Figure 2-5. Port P2

2.2.4 Port P3 (P32 - P30)

Port P3 is an 3-bit input/output port, and is also used as serial bus interface (SBI) input/output. Input/output mode is specified by the corresponding bit in the port P3 input/output control register (P3CR). Port P3 is configured as an input if its corresponding P3CR bit is cleared to "0", and as an output if its corresponding P3CR bit is set to "1". During reset, P3CR is initialized to "0", which configures port P3 as input. The P3 output latches are also initialized to "0". Port P3 is also used as a serial bus interface input/output. When used as a secondary function pin, set P3 to the output mode using P3 port input/output control register (P3CR) to control input/output by the output data. (P3 is sink open drain with input/output control. At "1" of the output data, a pin is set to Hi-z and enabled to input.)
When a read instruction for port P3 is executed bit 7 to 3 in P3 read in as undefined data.

Note1 : Ports set to the input mode read the pin states. When input pin and output in exist in port P3 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.

Note2 : The P3CR 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.)

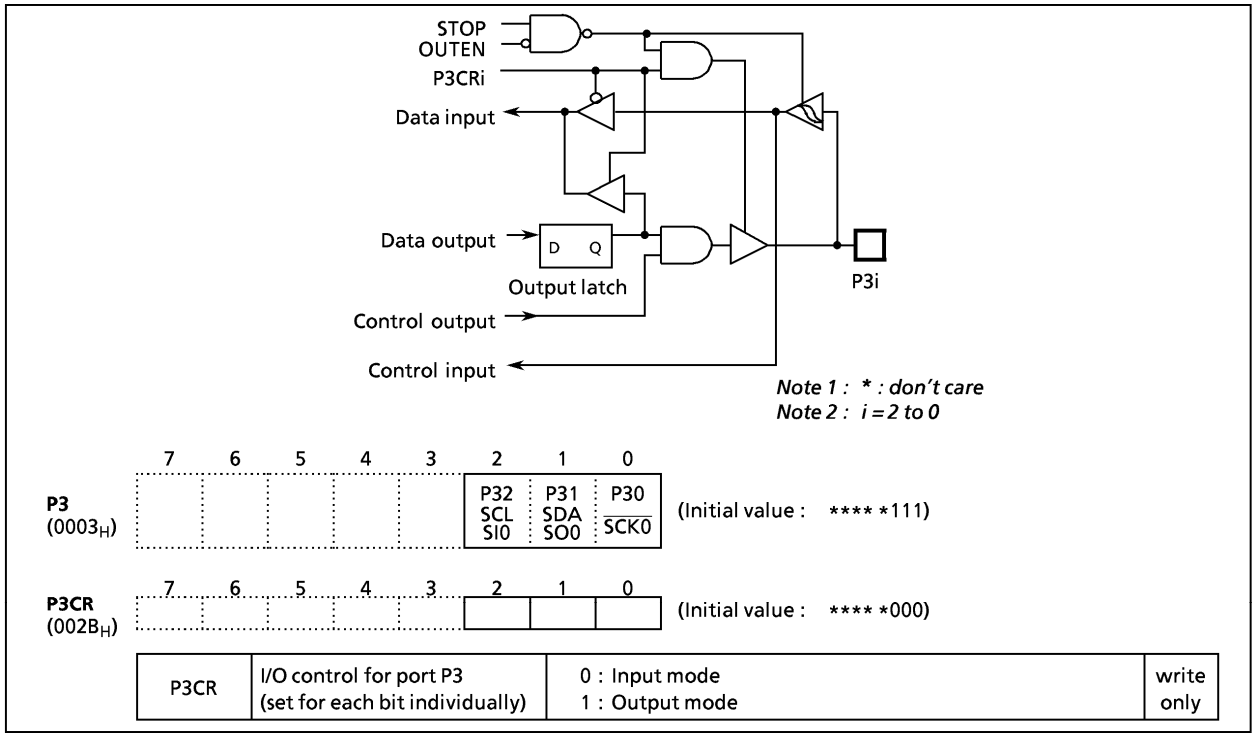


Figure 2-6. Port P3

2.2.5 Port P4 (P47 - P40)

Ports P4 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 P4 input/output control register (P4CR).

At reset, P4CR is set to 0 and AINDS is cleared to 0. Thus, P4 becomes an analog input port . At the same time, the output latch of port P4 is initialized to 0. P4CR is a write-only register. Pins not used for analog input can be used as I/O ports. But do not execute the output instruction to keep the accuracy in A/D conversion. Executing an input instruction on port P4 when the A/D converter is in use reads 0 at pins set for analog input : 1 or 0 at pins not set for analog input, depending on the pin input level.

Note1 : Ports set to the input mode read the pin states. When input pin and output in exist in port P4 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.

Note2 : The P4CR 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.)

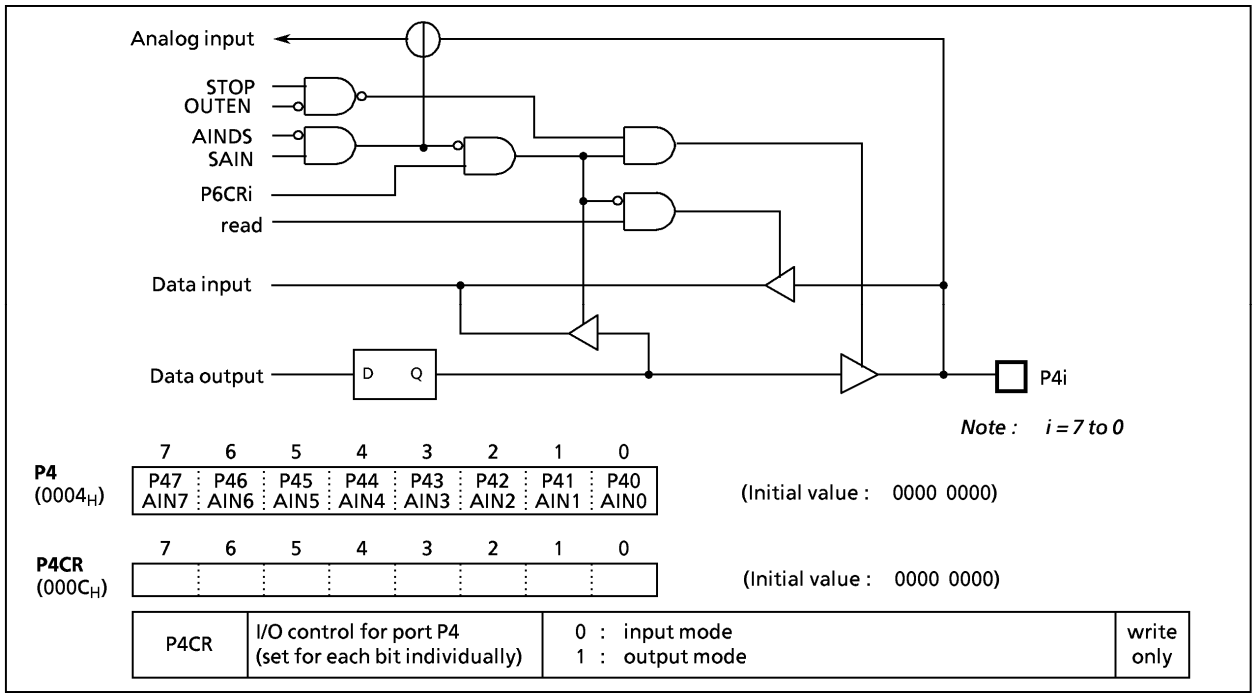


Figure 2-7. Port P4

2.2.6 Port P5 (P57 - P50)

Ports P5 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 P5 input/output control register (P5CR).

At reset, P5CR is set to 0 and AINDS is cleared to 0. Thus, P5 becomes an analog input port . At the same time, the output latch of port P5 is initialized to 0. P5CR is a write-only register. Pins not used for analog input can be used as I/O ports. But do not execute the output instruction to keep the accuracy in A/D conversion. Executing an input instruction on port P5 when the A/D converter is in use reads 0 at pins set for analog input : 1 or 0 at pins not set for analog input, depending on the pin input level.

Note1 : Ports set to the input mode read the pin states. When input pin and output in exist in port P5 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.

Note2 : The P5CR 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.)

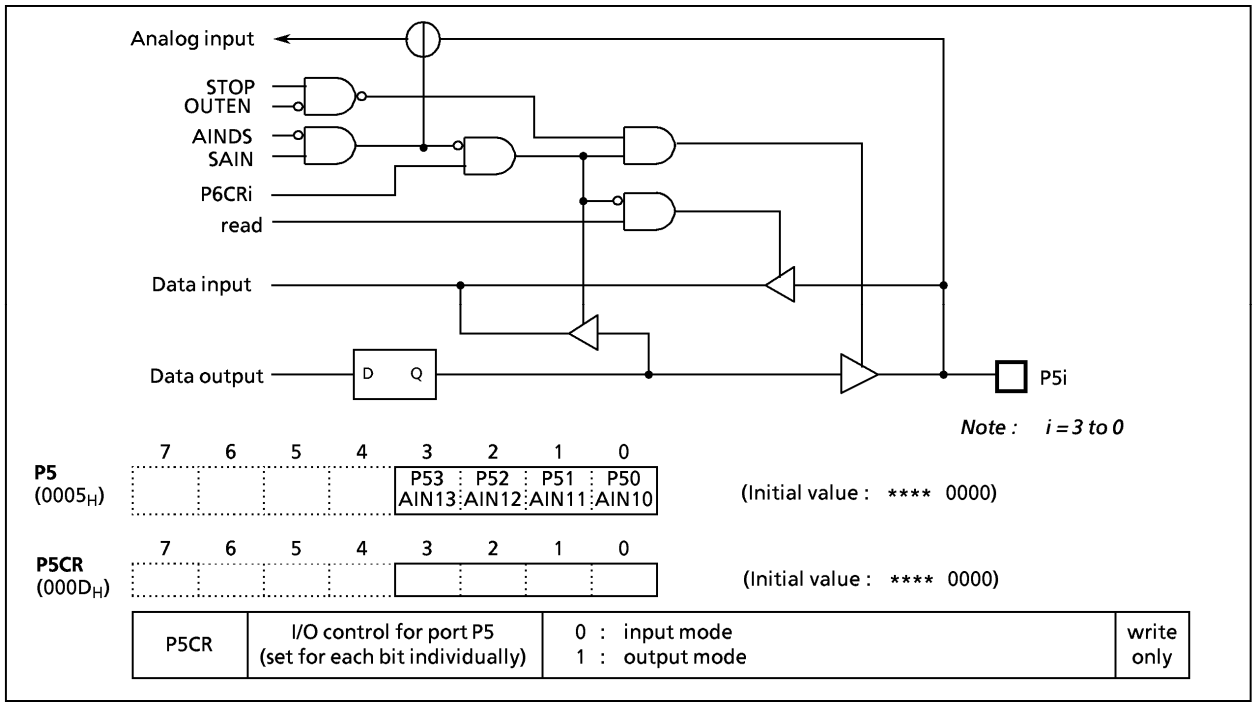
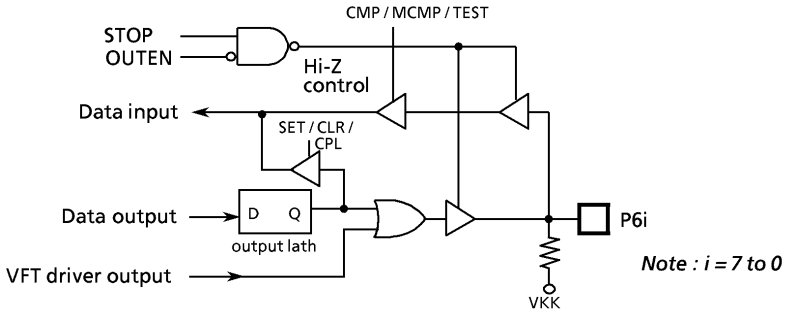


Figure 2-8. Port P5

2.2.7 Port P6 (P67 - P60)

Port P6 are 8-bit high-breakdown voltage input/output ports, and are also used as VFT driver outputs, which can directly drive vacuum fluorescent tube (VFT). When used as an input port or a VFT driver output, the output latch should be cleared to "0". The output latches are initialized to "0" during reset. Pins which are not set for VFT driver output can be used as normal I/O port (refer to section "2.12.6 Port Function"). It is recommended that pins P67 to P60 should be used as VFT driver output.



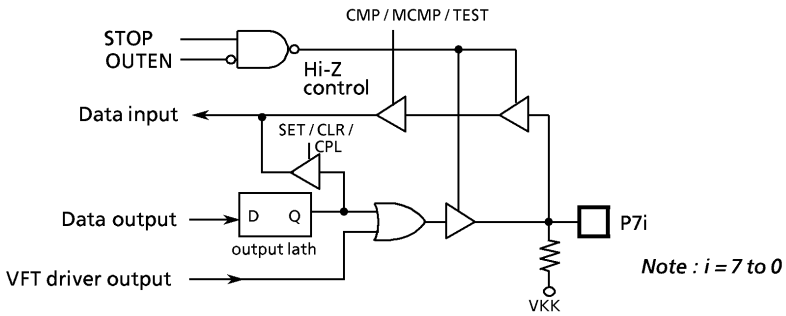
	7	6	5	4	3	2	1	0
P6 (0006_H)	P67	P66	P65	P64	P63	P62	P61	P60
	V7	V6	V5	V4	V3	V2	V1	V0

(Initial value : 0000 0000)

Figure 2-9. P6 Port

2.2.8 Port P7 (P77 - P70)

Port P7 are 8-bit high-breakdown voltage input/output ports, and are also used as VFT driver outputs, which can directly drive vacuum fluorescent tube (VFT). When used as an input port or a VFT driver output, the output latch should be cleared to "0". The output latches are initialized to "0" during reset. Pins which are not set for VFT driver output can be used as normal I/O port (refer to section "2.12.6 Port Function"). It is recommended that pins P77 to P70 should be used as VFT driver output.



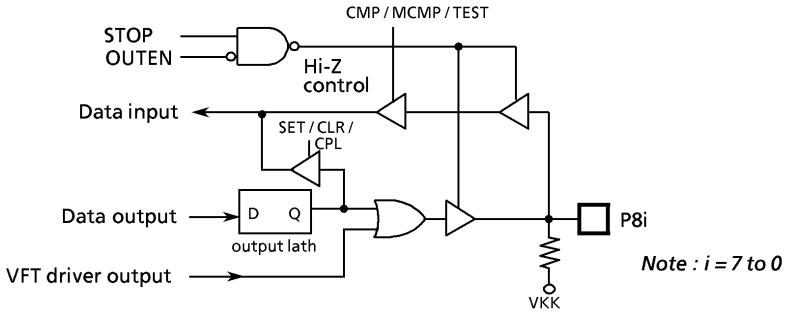
	7	6	5	4	3	2	1	0
P7 (0007_H)	P77	P76	P75	P74	P73	P72	P71	P70
	V15	V14	V13	V12	V11	V10	V9	V8

(Initial value : 0000 0000)

Figure 2-10. P7 Port

2.2.9 Port P8 (P87 - P80)

Port P8 are 8-bit high-breakdown voltage input/output ports, and are also used as VFT driver outputs, which can directly drive vacuum fluorescent tube (VFT). When used as an input port or a VFT driver output, the output latch should be cleared to "0". The output latches are initialized to "0" during reset. Pins which are not set for VFT driver output can be used as normal I/O port (refer to section "2.12.6 Port Function"). It is recommended that pins P87 to P80 should be used as VFT driver output.



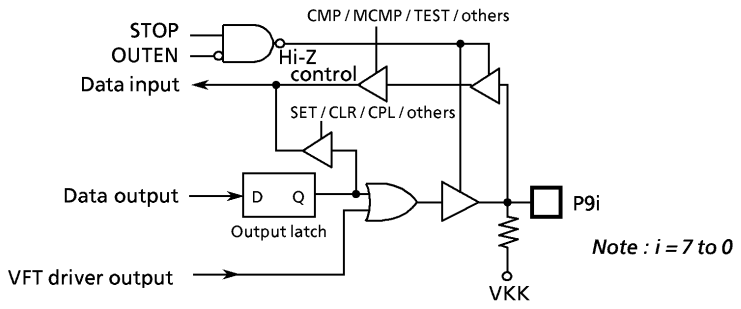
	7	6	5	4	3	2	1	0
P8 (0008_H)	P87	P86	P85	P84	P83	P82	P81	P80
	V23	V22	V21	V20	V19	V18	V17	V16

(Initial value : 0000 0000)

Figure 2-11. P8 Port

2.2.10 Port P9 (P97 - P90)

Port P9 is an 8-bit high-breakdown voltage input/output port, and also used as a VFT driver output, which can directly drive vacuum fluorescent tube (VFT). When used as an input port or a VFT driver output, the output latch should be cleared to "0". The output latches are initialized to "0" during reset. Pins which are not set for VFT driver output can be used as normal I/O port (refer to section "2.12.6 Port Function"). It is recommended that pins P97 to P90 should be used as segment output.



	7	6	5	4	3	2	1	0
P9 (0009_H)	P97	P96	P95	P94	P93	P92	P91	P90
	V31	V30	V29	V27	V26	V25	V23	V22

(Initial value : 0000 0000)

Figure 2-12. Port P9

2.2.11 PD (PD4 - PD0)

Ports PD are high-breakdown voltage input/output ports, and are also used as VFT driver outputs, which can directly drive vacuum fluorescent tube (VFT). The segment and the input/output port are specified by VSEL (bit 5 to 0) of VFT driver control register (VFTCR1). At reset, VSEL is cleared to "0" and used as the input/output port. When used as an input port or a VFT driver output, the output latch should be cleared to "0". Pins which are not set for VFT driver output can be used as normal I/O port (refer to section"2.12.6 Port Function"). The output latches are initialized to "0" during reset. When a read instruction for port PD is executed bit 7 to 5 in PD read in as undefined data.

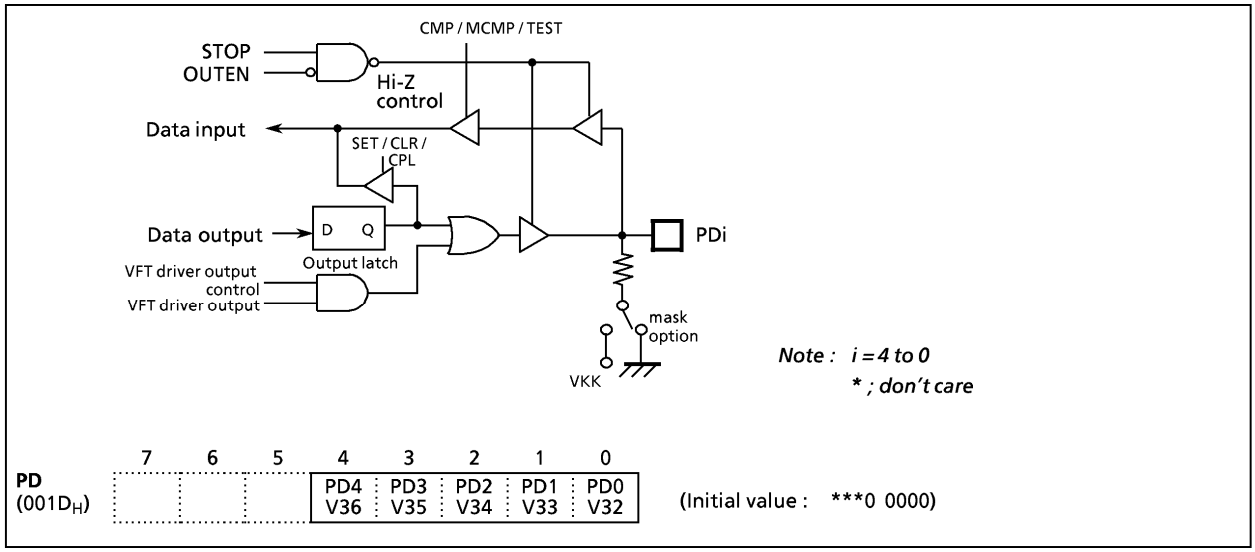


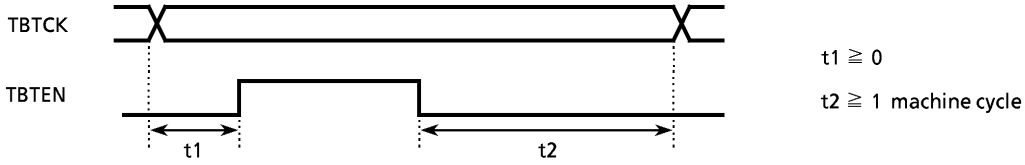
Figure 2-13. PD, PE, PF Ports

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-bace 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 2-14 (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
```

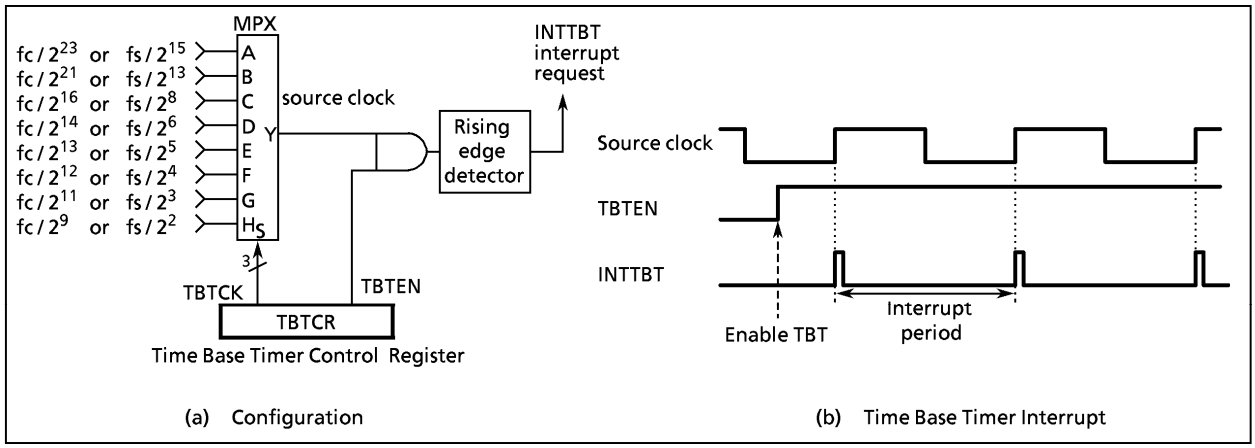


Figure 2-14. Time Base Timer

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 2-15. 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)$_L$		

(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 2-16. 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

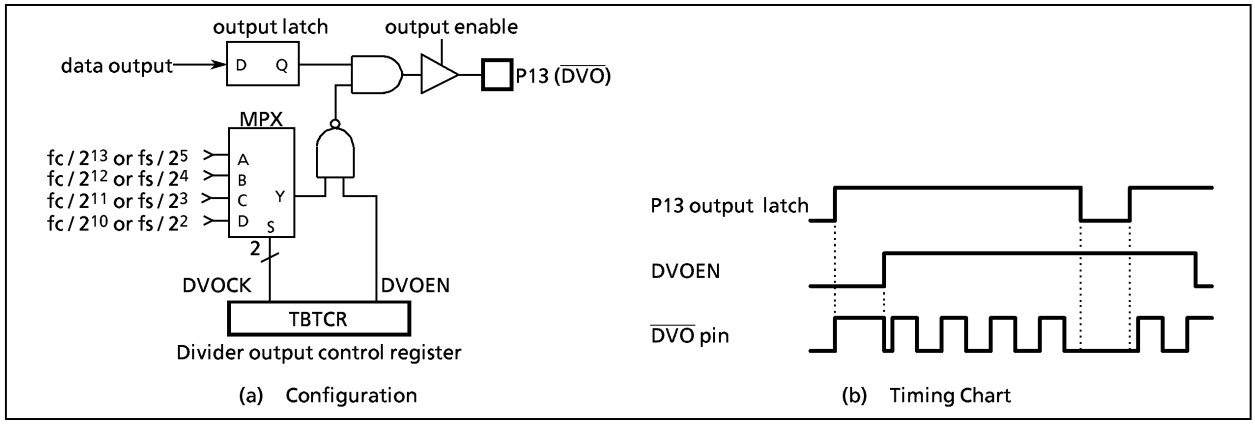


Figure 2-17. Divider Output

2.5 16-bit Timer/Counter 1 (TC1)

2.5.1 Configuration

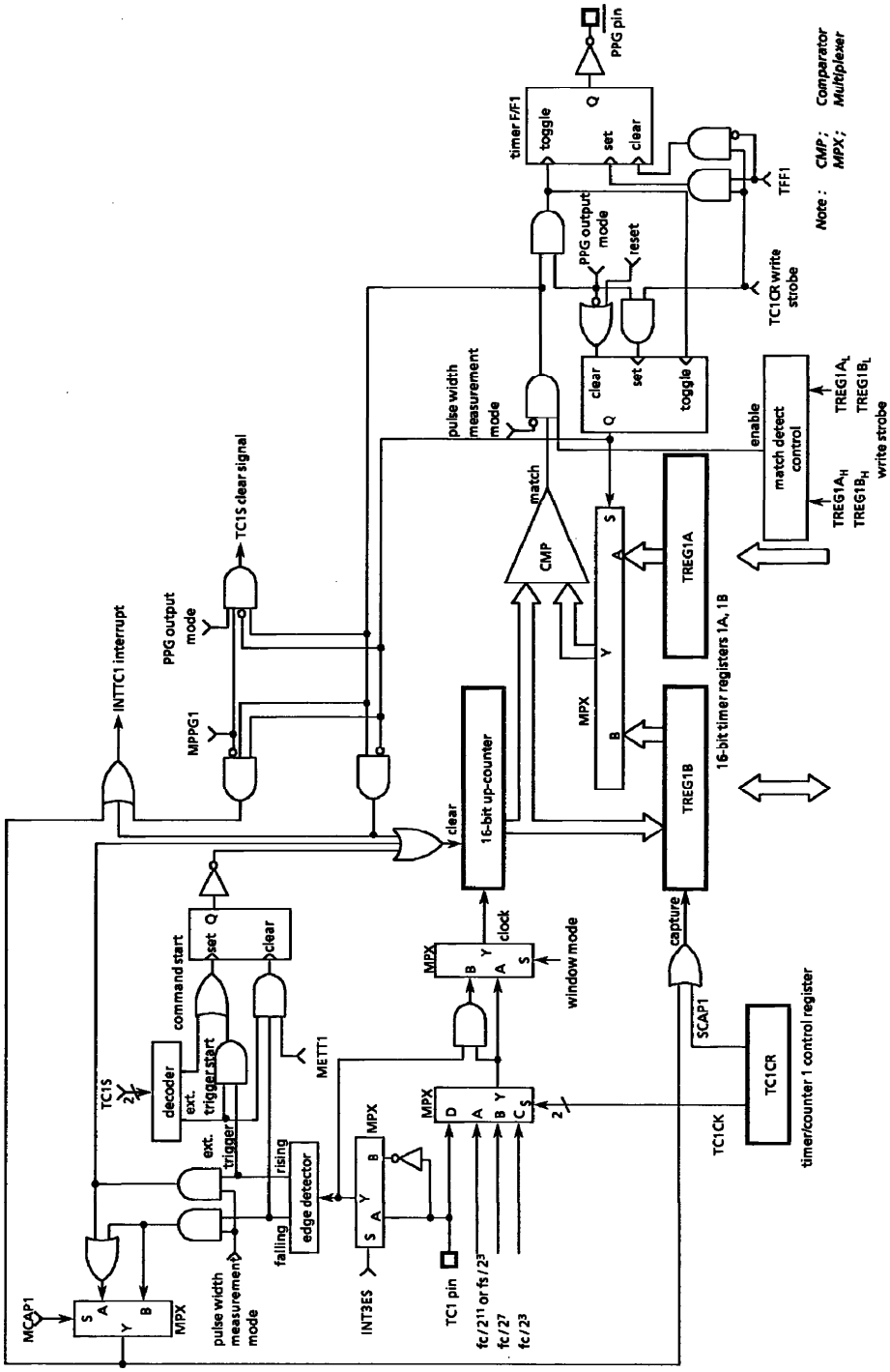
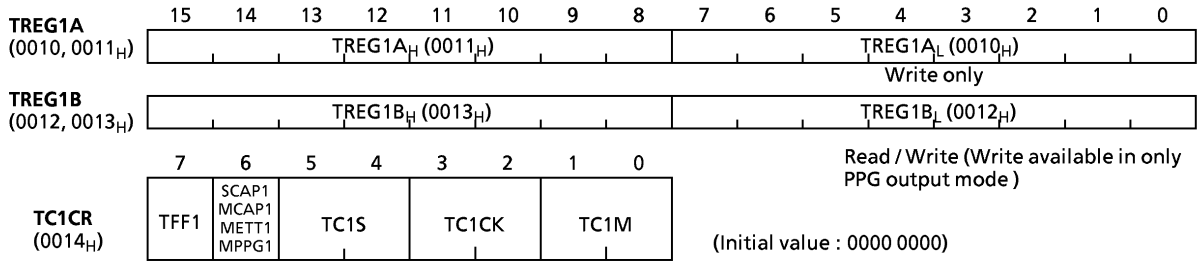


Figure 2-18. Timer/Counter 1

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.



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/2^3 [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 start	
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 start 1 : trigger start & stop	
MPPG1	PPG output control	0 : continuous 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. After writing to the high-byte, the comparison of 1 machine cycle (during instruction execution) is ignored.

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

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

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 be accessed by any read-modify-write instruction such as bit operate, etc.

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

Note9 : When fc/2^3 is selected for the source clock in pulse width measurement mode, the least significant bit of TREG1B which has been read out is always "0", but when the other source clocks are selected, the least significant bit of TREG1B becomes "1" or "0" according to the value of counter.

Figure 2-19. 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 1 s later (at fs = 32.768 kHz).

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

Example 2 : Software capture

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

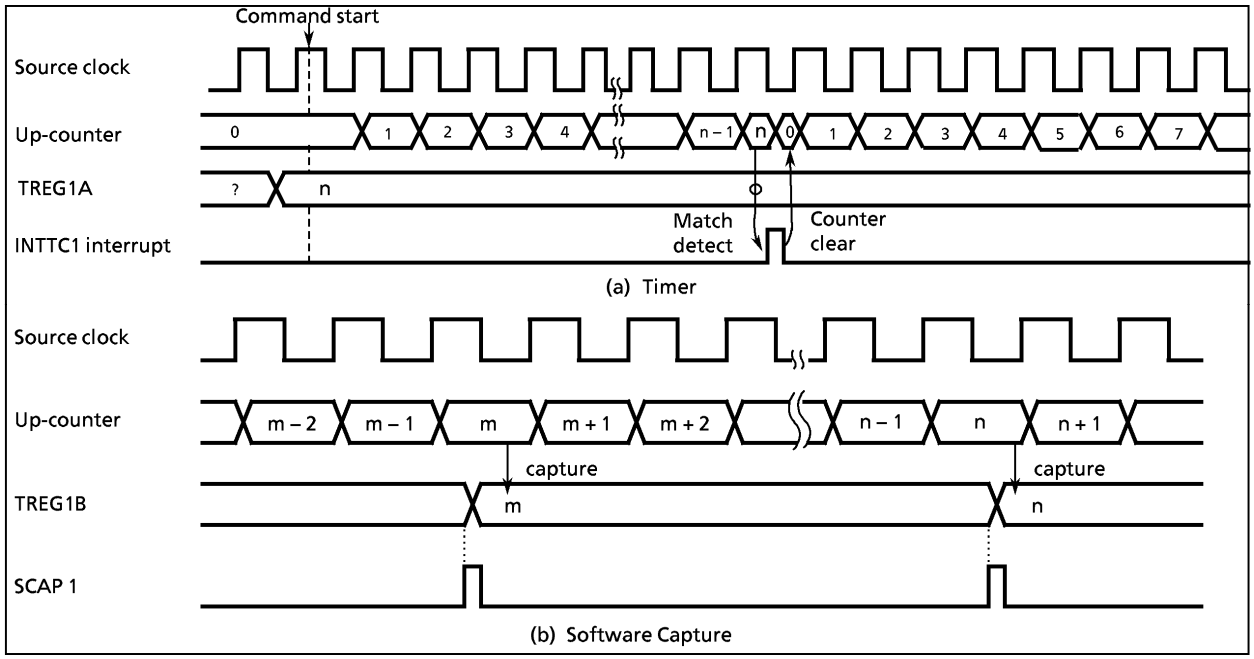


Figure 2-20. Timer Mode Timing Chart

(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 INT3ES. Edge selection is the same as for the external interrupt input INT3 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 INT3 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.

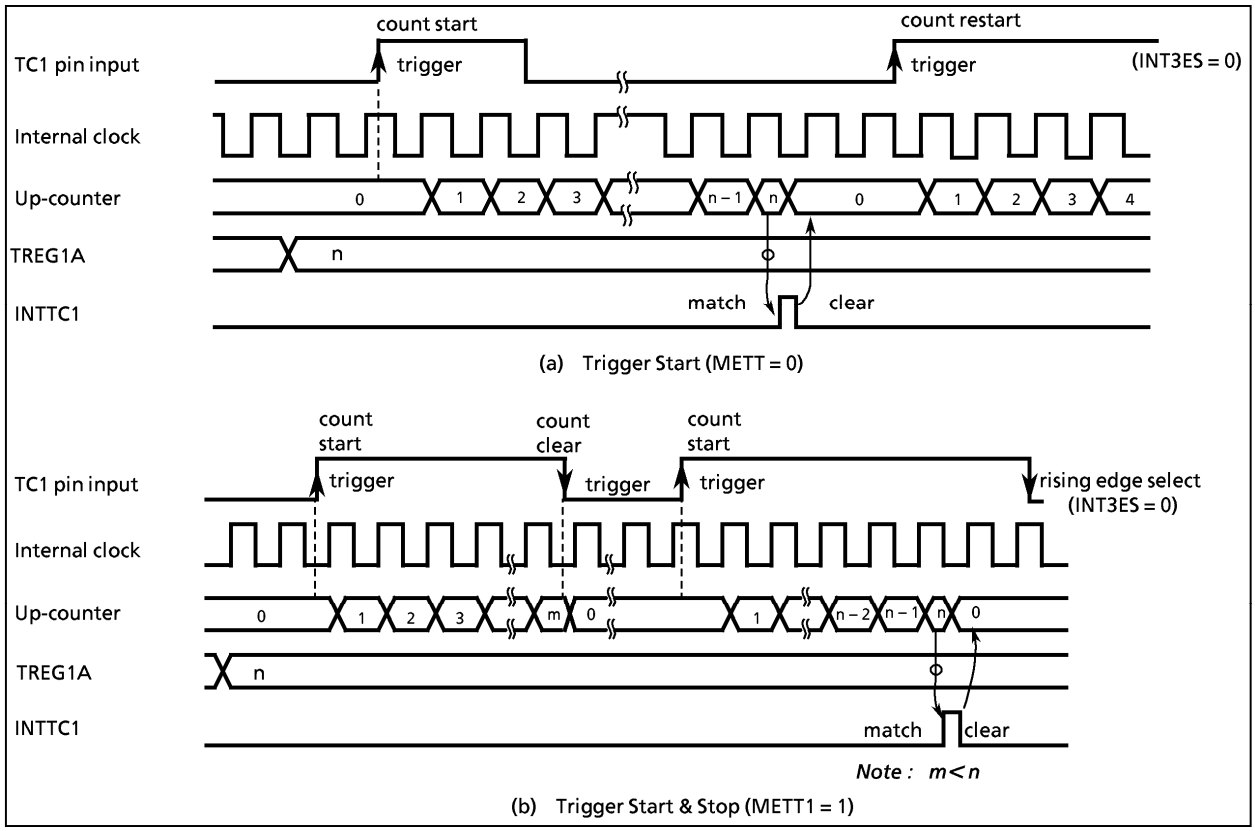


Figure 2-21. External Trigger Timer Mode Timing Chart

(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 INT3ES 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.

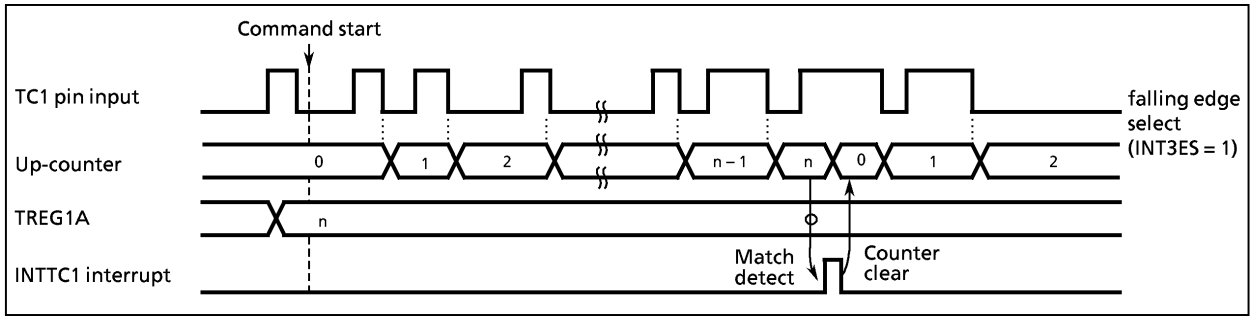


Figure 2-22. Event Counter Mode Timing Chart (INT3ES = 1)

(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 INT3ES. 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.

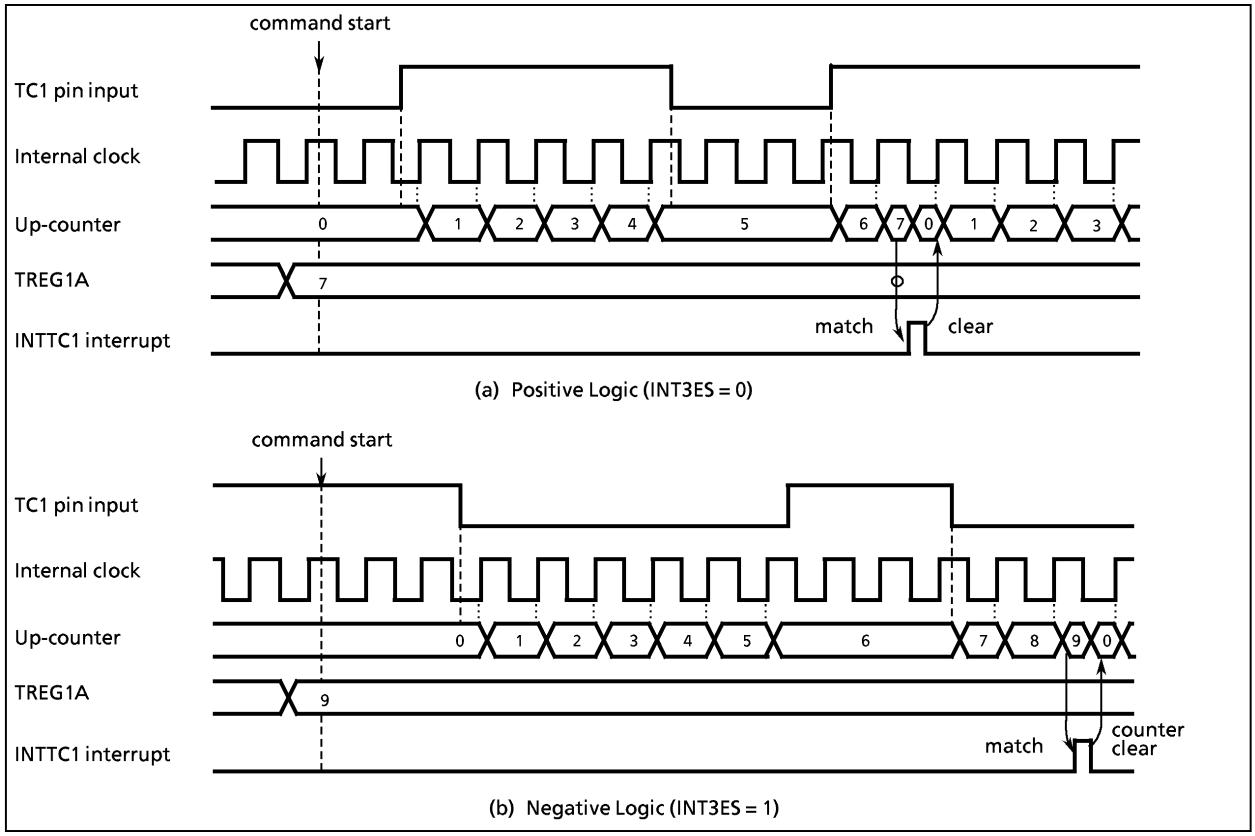


Figure 2-23. Window Mode Timing Chart

(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 INT3ES, and single edge or double edge is selected with MCAP1 (bit 6 in TC1CR).

Note : When fc/23 is selected for the source clock in pulse width measurement mode, the least significant bit of TREG1B which has been read out is always "0", but when the other source clocks are selected, the least significant bit of TREG1B becomes "1" or "0" according to the value of counter.

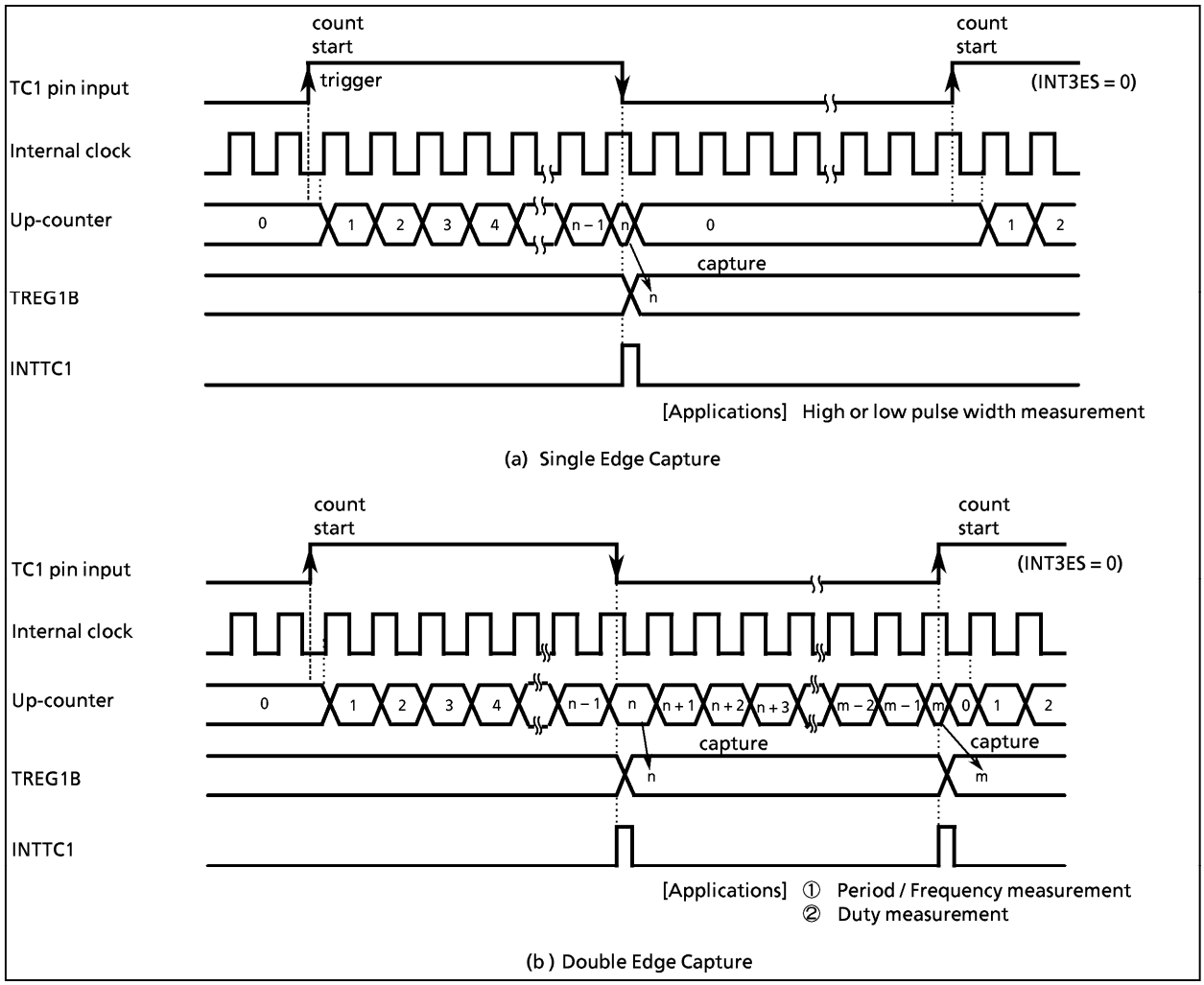


Figure 2-24. Pulse Width Measurement Mode Timing Chart

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

```
                CLR     (INTTC1C). 0            ;   INTTC1 service switch initial setting
                LD      (EINTCR), 00000000B     ;   Sets the rise edge at the INT3 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     (INTTC1C). 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 P12 ($\overline{PPG}$) pin. In the case of $\overline{PPG}$ output, set the P12 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.

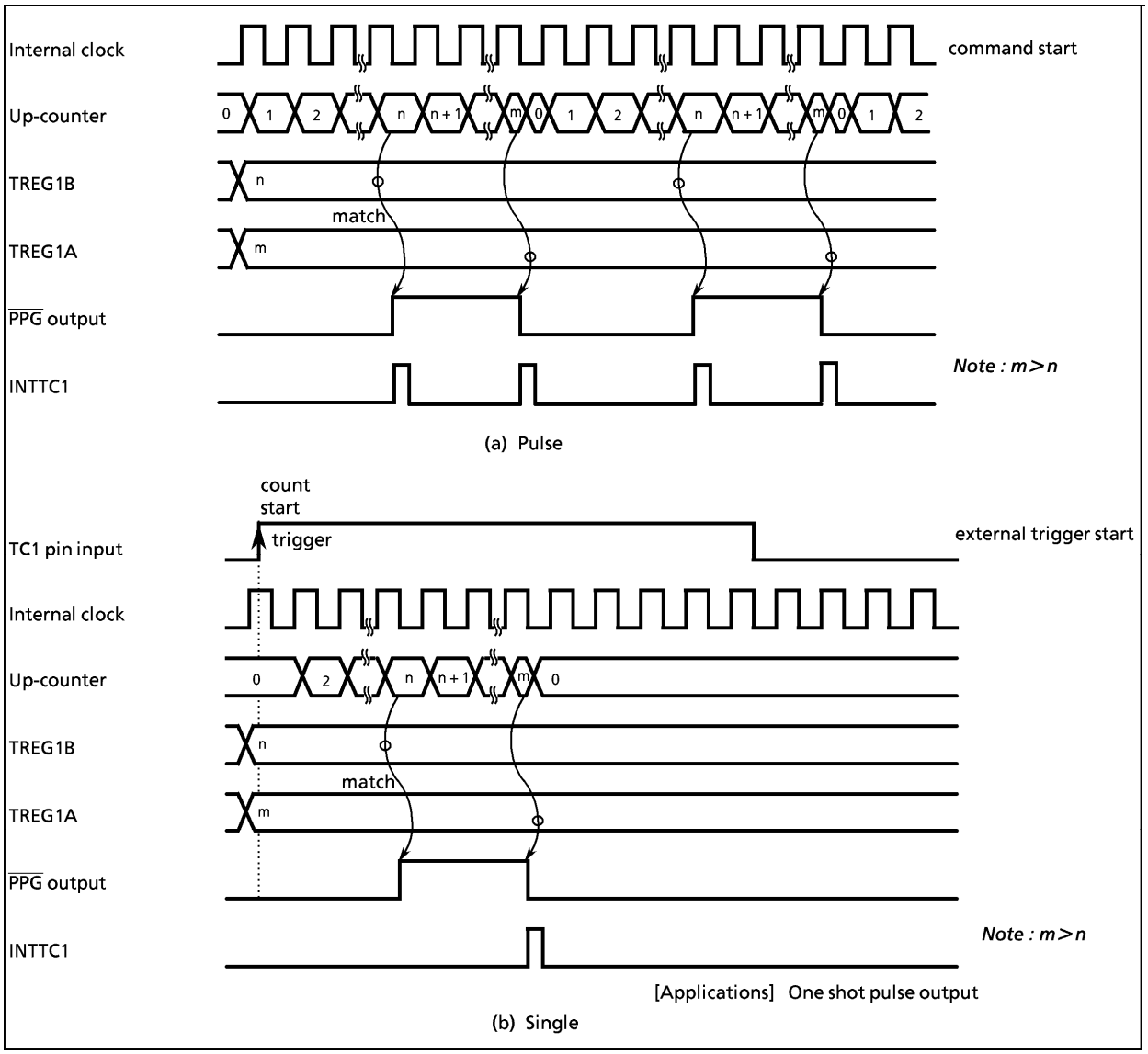


Figure 2-25. PPG Output Mode Timing Chart

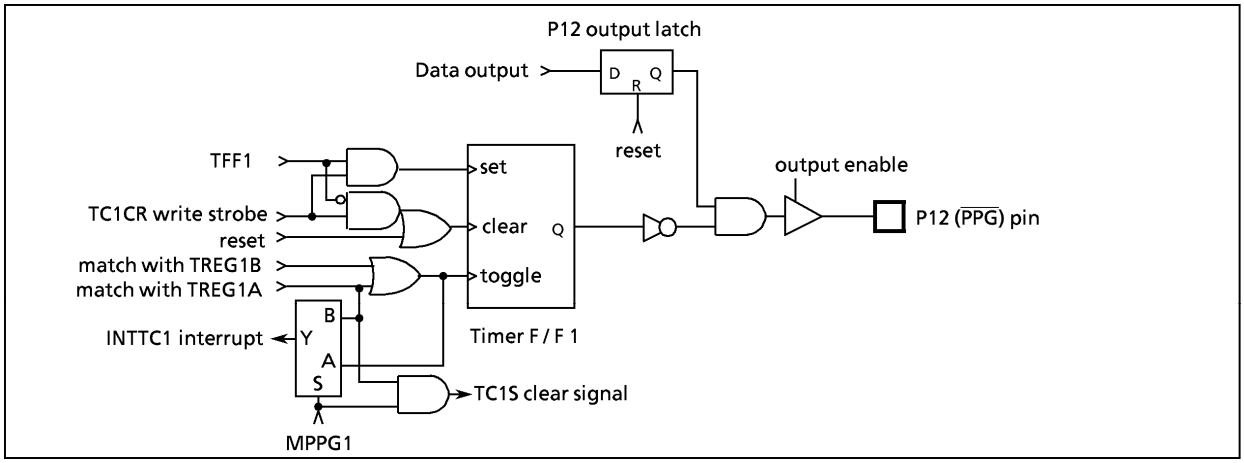


Figure 2-26. PPG Output

2.6 16-bit Timer/Counter 2 (TC2)

2.6.1 Configuration

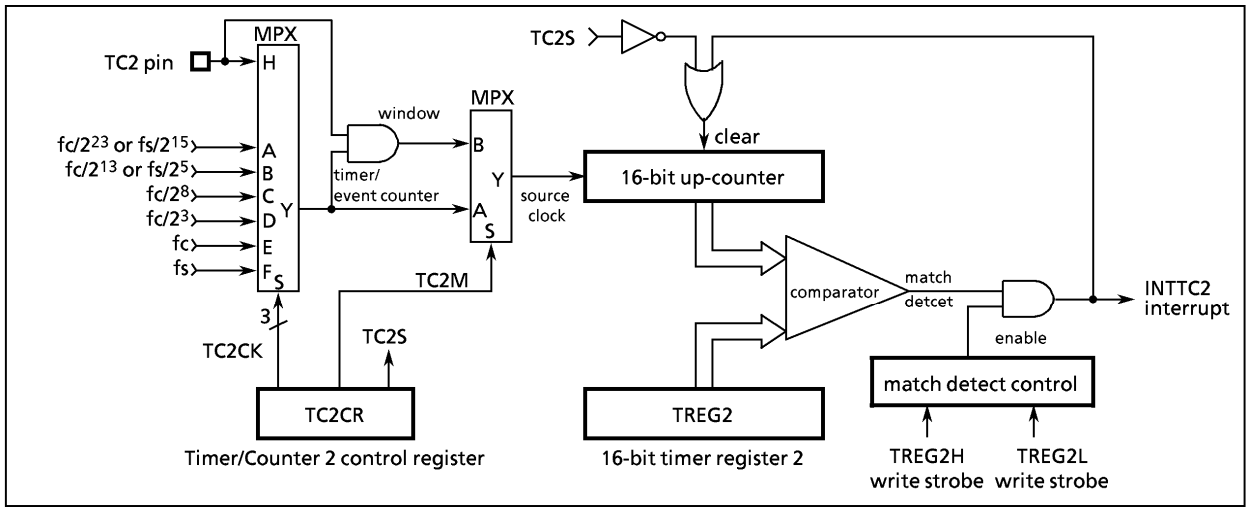


Figure 2-27. Timer/Counter 2 (TC2)

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
"0"	"0"	TC2S	TC2CK			"0"	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 2-28. 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 h	18.2 h
$fc/2^{13}$	$fs/2^{5}$	$fs/2^{5}$	$fs/2^{5}$	1.02 ms	0.98 ms	1.1 min	1.07 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		8.2 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 25 ms (at fc = 8 MHz).

```
LDW     (TREG2), 61A8H          ; Sets TREG2 (25 ms ÷ 2^3/fc = 61A8H)
SET     (EIRH). EF14            ; INTTC2 interrupt enable
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.

```
LDW     (TREG2), 0280H          ; Sets TREG2
SET     (EIRH). EF14            ; INTTC2 interrupt enable
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.

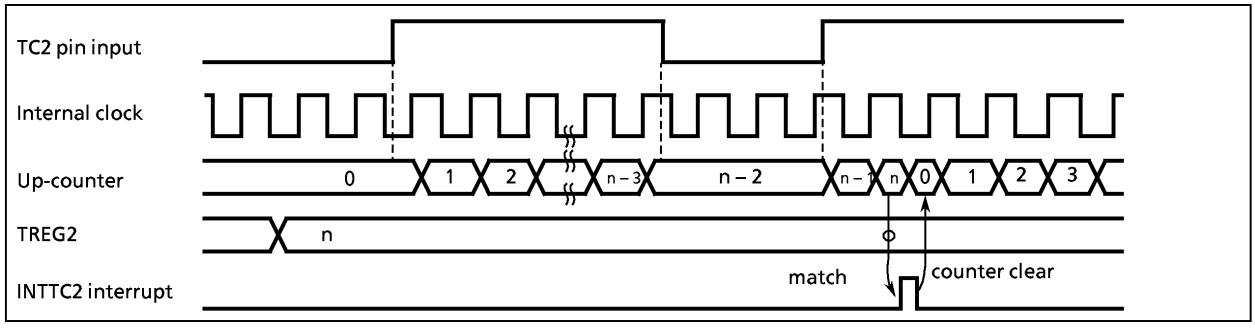


Figure 2-29. Window Mode Timing Chart

2.7 8-Bit Timer/Counter 3 (TC3)

2.7.1 Configuration

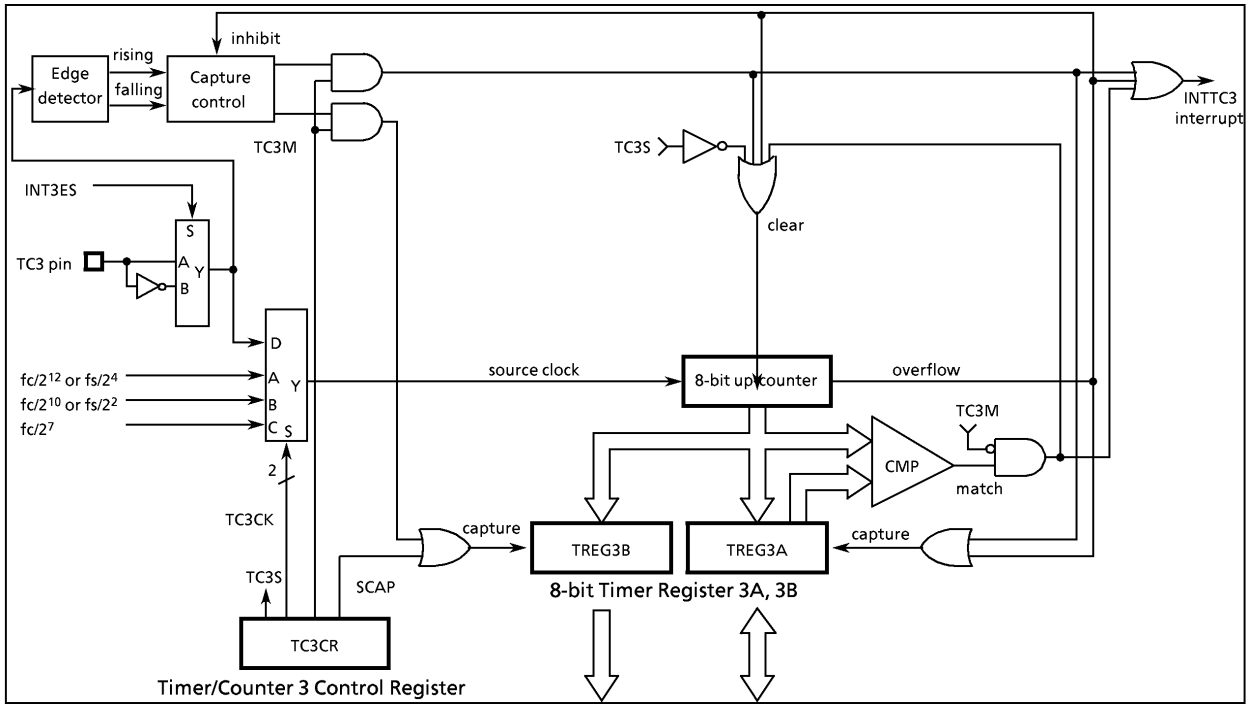


Figure 2-30. Timer/Counter 3

2.7.2 Control

TREG3A (0018_H) — Read/Write

TREG3B (0019_H) — Read only

TC3CR ($001A_H$) — (Initial value : *0*0 00*0)

7	6	5	4	3	2	1	0
"0"	SCAP	"0"	TC3S	TC3CK		"0"	TC3M

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 2-31. Timer Register 3A/3B and TC3 Control Register

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.

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				
DV7CK = 0	DV7CK = 1		fc = 8 MHz	fs = 32.768 kHz	fc = 8 MHz	fs = 32.768 kHz
$fc / 2^{12}$ [Hz]	$fs / 2^4$ [Hz]	$fs / 2^4$ [Hz]	512 μs	488.28 μs	131.1 ms	124.5 ms
$fc / 2^{10}$	$fs / 2^2$	–	128 μs	122.07 μs	32.6 ms	31.1 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 50Hz pulses to the TC3 pin.

```
LD (TC3CR), 00001100    ;  Sets TC3 mode, 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.

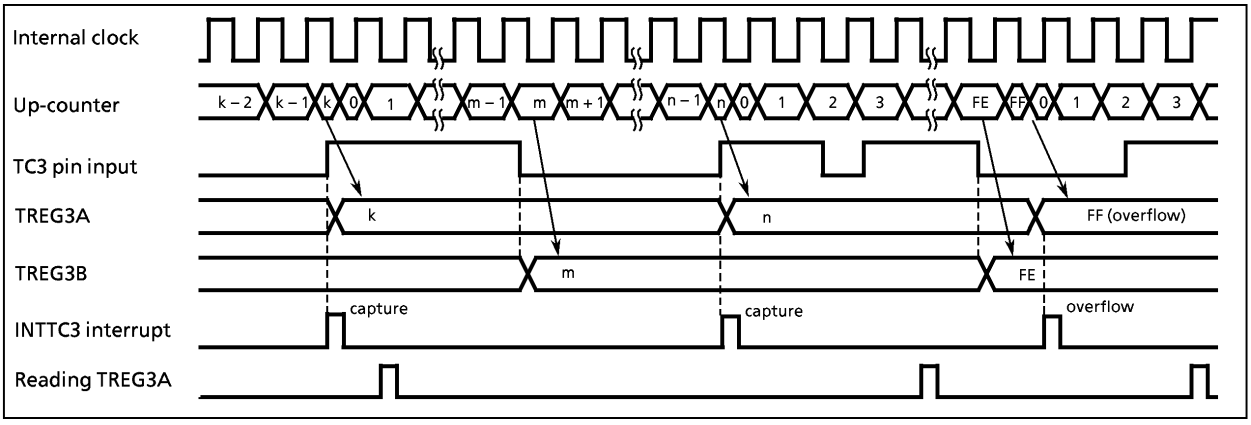


Figure 2-32. Timing Chart for Capture Mode (INT3ES = 0)

INT4/TC3 input pulse width measurement (for detecting remote control receive waveform)

To detect and measure the low or high level width of waveforms input from INT4 or TC3, set timer 3 to capture mode and INT4/TC3 input edge detect to both edges.

Figure 2-33. is a timing chart of when timer 3 is used in capture mode. Numbers ① to ⑱ in Figure 2-33. are described below:

① Set INT4/TC3 edge detect to both edges.
In Figure 2-33. INT4ES = 1 (falling edge is selected) and INT4W = 1 (both-edge detect enable).
Change INT4ES and INT4W bits at $\#0024_H$ only when IMF = 0. After changing EINTCR, interrupt latches of external interrupts must be cleared to "0" using load instruction.

② Wake up timer 3 and enable timer 3 soft capture. Then the timer 3 counter starts free running. Also, enable timer 3 interrupts (EF_8 = 1).

③~⑤ At the selected (falling) edge of the INT4/TC3 input pin, the current counter value (K) is fetched to TREG3A and the counter is zero-cleared. At the same time, a timer 3 interrupt is generated.

⑥ The interrupt processing routine for the timer 3 interrupt sets EF_{12} in the interrupt enable register (EIR) to 1 and clears EF_8 to 0 so that INT4 can be detected at the non-selected (rising) edge of INT4/TC3 input.

⑦ TREG3B and TREG3A are read next, because after a timer 3 interrupt by capture to TREG3A is generated, capture/overflow detect is halted until the next TREG3A read. Reading TREG3A by the interrupt processing routine resumes capture/overflow detect.

⑧~⑩ Timer count continues. The counter value (m) is fetched to TREG3B at the next non-selected (rising) edge.
At this time, INT4 is generated.

⑪ Simultaneously, bit 6 in INT4EDT (#0024) is set to 1.

⑫ The interrupt processing routine for INT4 enables timer 3 interrupts and disables INT4 interrupts to detect the next edge (selected edge at ⑮).

⑬~⑭ Same as ⑦, resumes next capture/overflow detect. TREG3B value (m) read at ⑭ is necessary to determine T_WL in Figure 2-33.
At ⑬ #0024H is read: bit 6 (INT4EDT) and bit 2 (INT4EDT) are set to 1. After read, bits 6 and 2 are both cleared.

⑮~⑯ Timer 3 count continues. The counter value (n) is fetched to TREG3A at the next selected (rising) edge. The counter is zero-cleared.
At the same time, INT4 is generated.

⑱ Interrupt processing for INT4 resumes at the next capture/overflow detect as at ⑦. T_WH is determined by TREG3A value (n) and TREG3B value (m) read at ⑭.

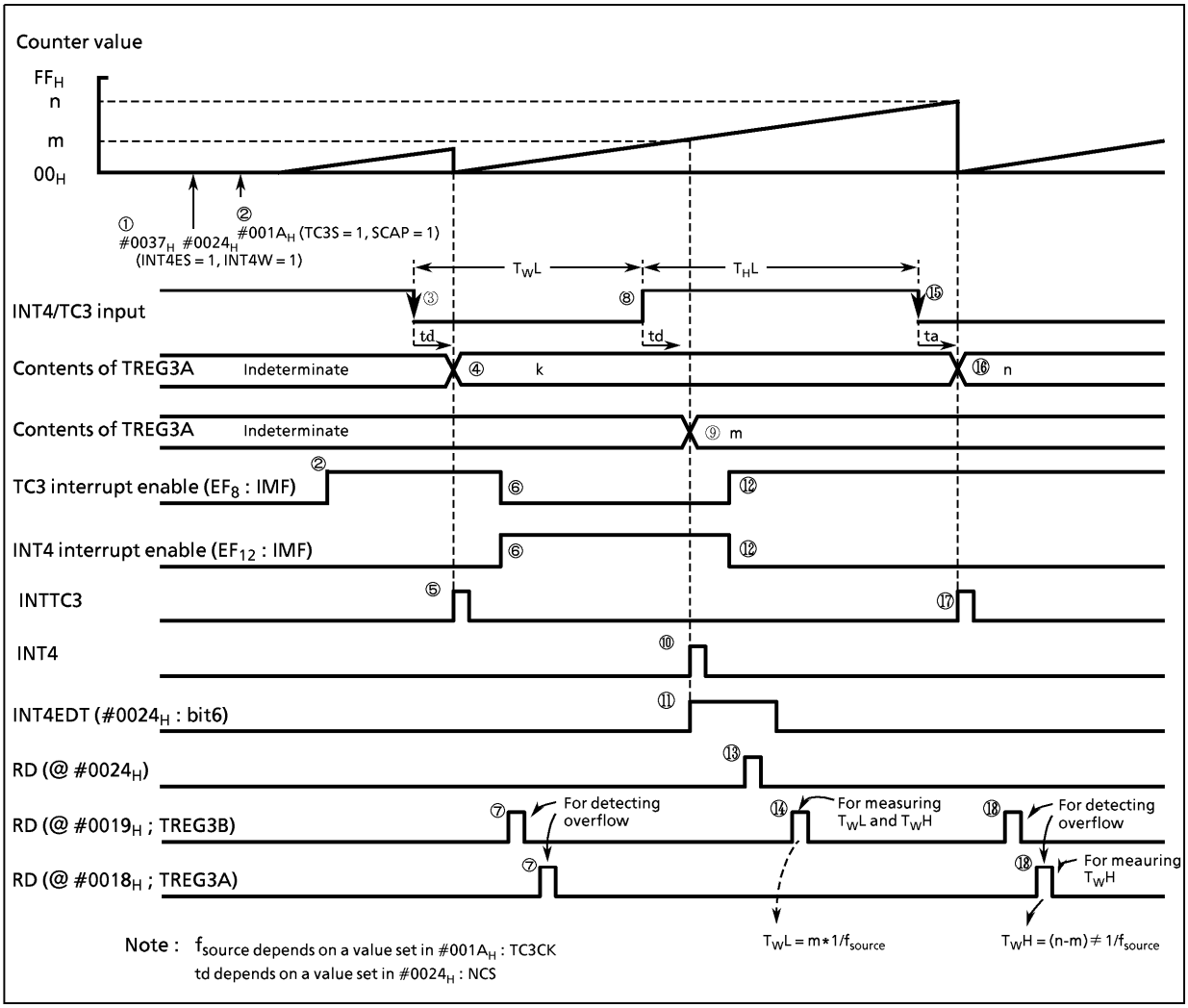


Figure 2-33. Example of remote control waveform by timer 3 (in capture mode)

2.8 8-bit Timer/Counter (TC4)

2.8.1 Configuration

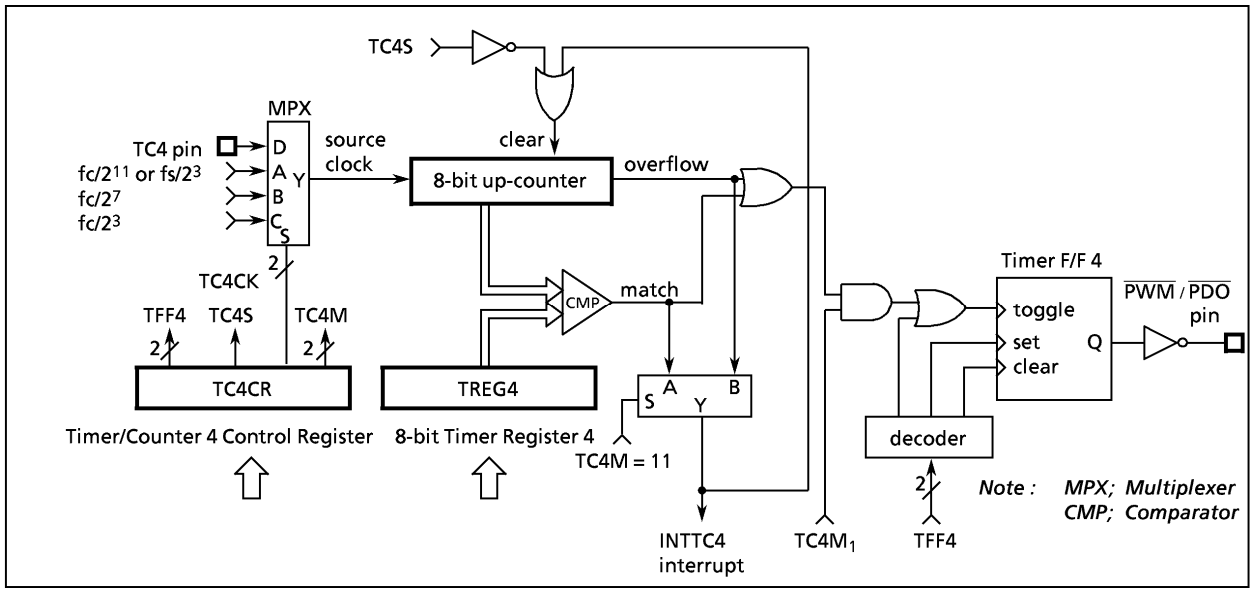


Figure 2-34. Timer/Counter 4

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		"0"	TC4S	TC4CK		TC4M	

(Initial value : 00*0 0000)

TC4M	TC4 operating mode select	00 : Timer mode 01 : Reserved 10 : Programmable divider output (PDO) mode 11 : Pulse width modulation (PWM) output mode	Write only
TC4CK	TC4 source clock select	00 : Internal clock fc / 2^{11} or fs / 2^3[Hz] 01 : Internal clock fc / 2^7 10 : Internal clock fc / 2^3 11 : External clock (TC4 pin input)	
TC4S	TC4 start control	0 : Stop & clear 1 : Start	
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 and event counter modes.

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

$TREG4 > 0$

Note 5 : TC4CR and TREG4 are write-only registers and must not be used with any of read-modify-write instructions.

Figure 2-35. 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	
NORMAL1 / 2, IDLE1 / 2 mode		SLOW, SLEEP mode				
DV7CK = 0	DV7CK = 1		fc = 8 MHz	fs = 32.768 kHz	fc = 8 MHz	fs = 32.768 kHz
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	–

(2) **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 $\overline{PDO}$ (P44) 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)

```
SET     (P1). 4                 ;  P14 output latch ← 1
LD      (P1CR), 00010000BH      ;  Sets P14 (output mode)
LD      (TREG4), 10H            ;  1/2048 ÷ 2^7/fc = 10H
LD      (TC4CR), 00010010B      ;  Starts TC4
```

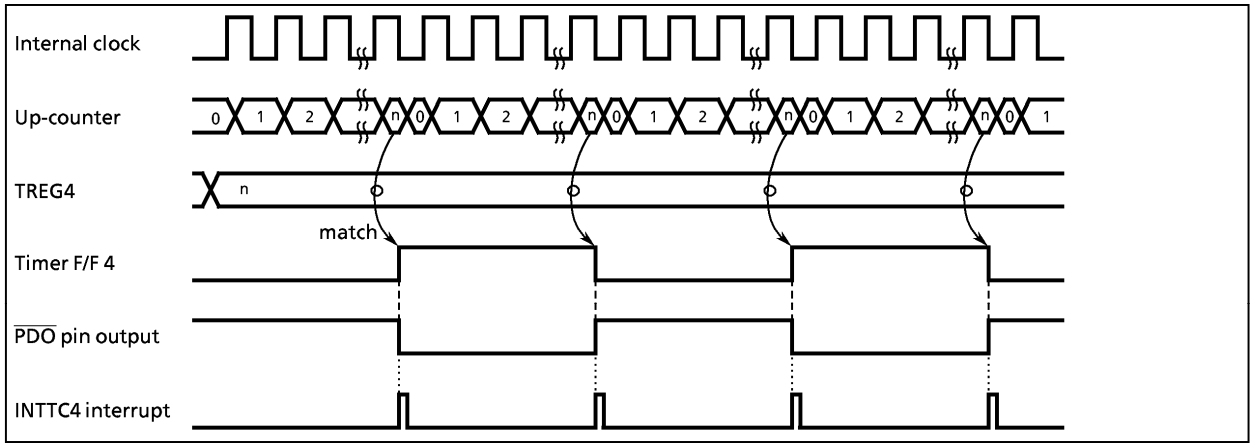


Figure 2-36. Timing Chart for PDO Mode

(3) **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/F 4 output is again toggled and the counter is cleared. Timer F/F 4 output is inverted and output to the $\overline{PWM}$ (P44) 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.

Note1 : Do not overwrite TREG4 only when an INTTC4 interrupt is generated. Usually, TREG4 is overwritten in the routine of INTTC4 interrupt service.

Note2 : PWM output mode can be used only in the NORMAL1, 2 and IDLE 1,2 mode.

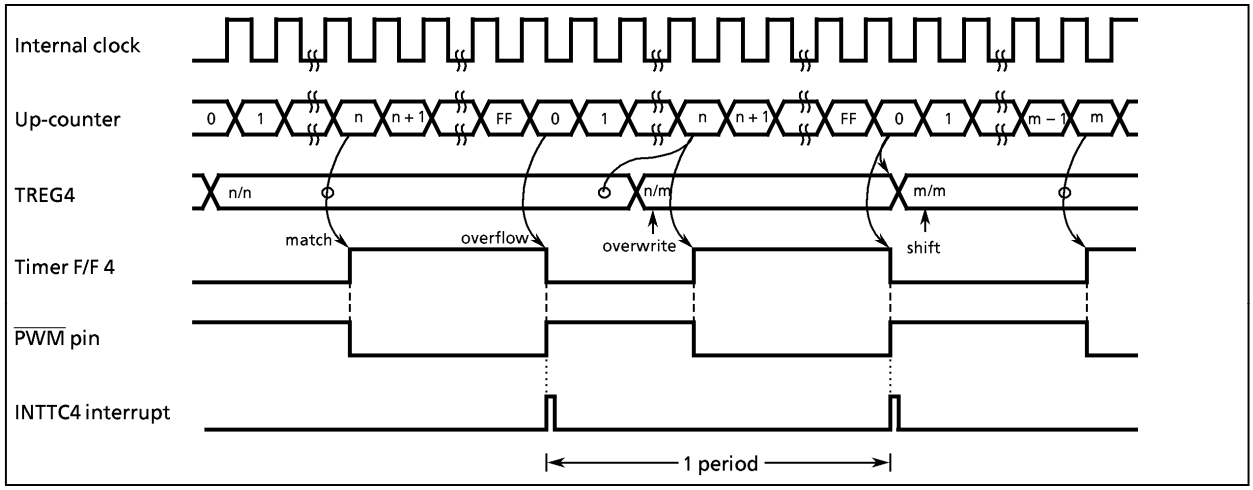


Figure 2-37. Timing Chart for PWM Mode

Table 2-7. PWM Output Mode

Source clock			Resolution		Maximum setting time	
NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode				
DV7CK = 0	DV7CK = 1		fc = 8 MHz	fs = 32.768 kHz	fc = 8 MHz	fs = 32.768 kHz
$fc / 2^{11}$ [Hz]	$fs / 2^{3}$ [Hz]	$fs / 2^{3}$ [Hz]	256 μs	244.14 μs	65.5 ms	62.5 ms
$fc / 2^{7}$	$fc / 2^{7}$	–	16 μs		4.1 ms	
$fc / 2^{3}$	$fc / 2^{3}$	–	1 μs		256 μs	

2.9 Serial Bus Interface (SBI-ver.A)

The 87CH74A/M74A has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit serial bus interface and an I²C bus.

The serial bus interface is connected to an external device through P31 (SDA) and P30 (SCL) in the I²C bus mode; and through P32 ($\overline{SCK0}$), P31 (SO0) and P30 (SI0) in the clocked-synchronous 8-bit SIO mode.

The serial bus interface pins are also used as the port. When used as serial bus interface pins, set the P3 output latches of these pins to "1". When not used as serial bus interface pins, the P3 port is used as a normal I/O port.

I²C bus has no an arbitration function which is necessary when two or more master devices scramble for the bus control. In master mode, other devices which are connected on the same bus need be slave devices. (single master)

Note : When a multi master I²C bus system operates in I²C bus mode of this serial bus interface circuit, there is a possibility that the following problems raise. I²C bus mode of this serial bus interface circuit should be used by a single master I²C bus system.

1. *The SCL line is fixed to low level and transferring stops by the serial bus interface circuit. The other devices can not run on the SCL line. Thus the bus locks.*
2. *The SCL pin is pulled down to low level regardless of the state of the SCL line by the serial bus interface circuit. A period of high-level SCL clock pulse which other devices output is shortened. The minimum value of which the SCL clock holds high level is not satisfied, which is specified with the I2C bus standard.*

2.9.1 Configuration

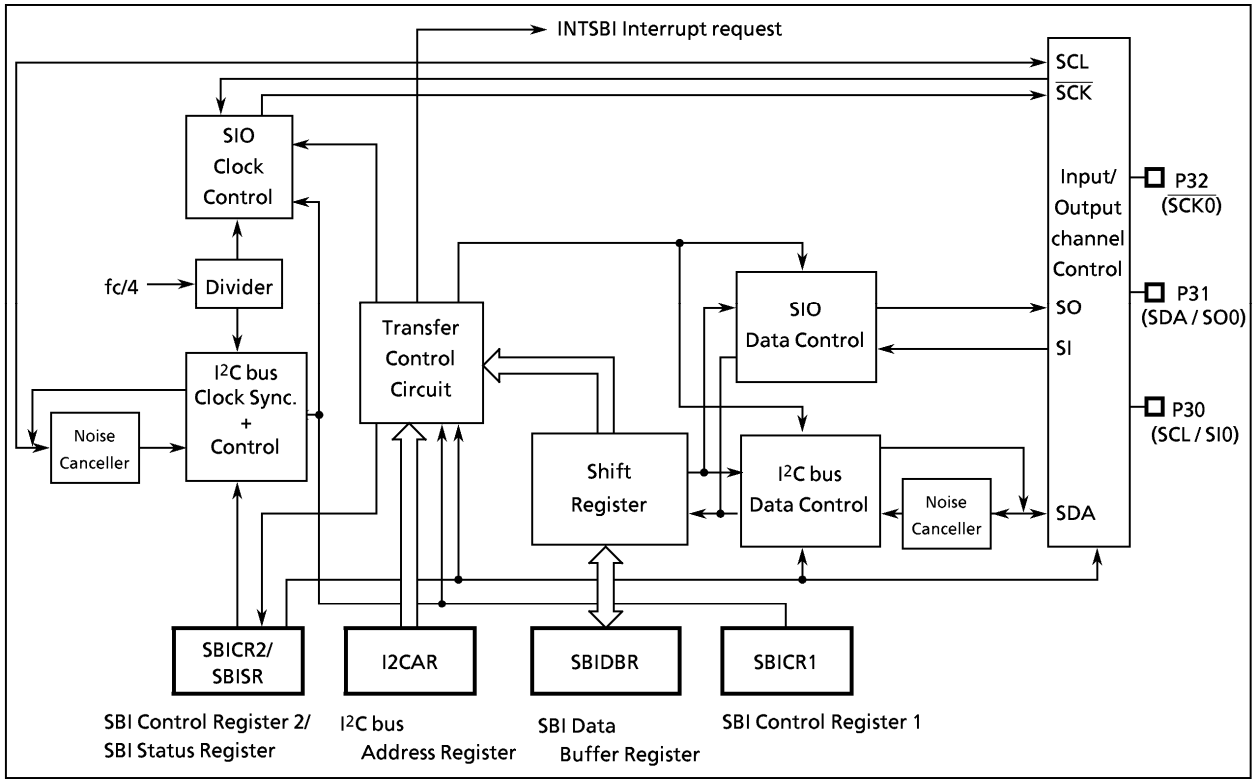


Figure 2-38. Serial Bus Interface (SBI-ver.A)

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

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

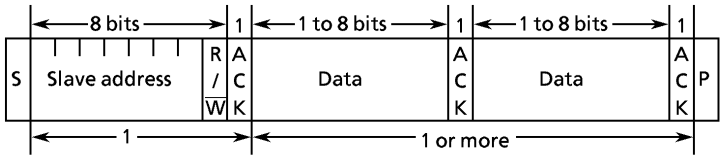
- Serial bus interface control register 1 (SBICR1)
- Serial bus interface control register 2 (SBICR2)
- Serial bus interface data buffer register (SBIDBR)
- I^2C bus address register (I2CAR)
- Serial bus interface status register (SBISR)

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".

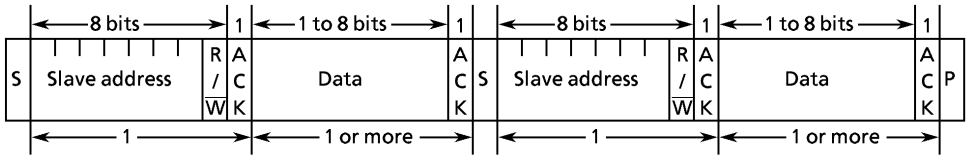
2.9.3 The Data Formats in the I^2C bus Mode

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

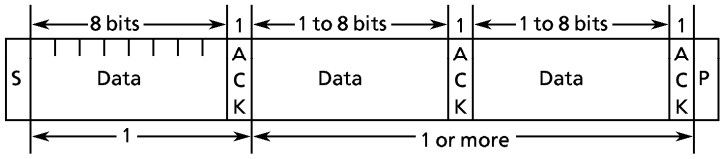
(a) Addressing format



(b) Addressing format (with restart)



(c) Free data format



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

Figure 2-39. Data Format

2.9.4 I²C Bus Mode Control

The following registers are used for control and operation status monitoring when using the serial bus interface (SBI-ver.A) in the I²C bus mode.

Serial Bus Interface Control Register 1

SBICR1 (0020_H)

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

(Initial value 0000 0000)

		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 : Does not generate clock pulse for acknowledgment. (master mode) / Does not count clock pulse for acknowledgment. (slave mode) 1 : Generates clock pulse for acknowledgment. (master mode) / Counts clock pulse for acknowledgment. (slave mode)					Read/ Write
SWRST	Initiate a internal of SBI	0 : – 1 : Initialized (clearing "0" after initialized)					
SCK	Serial clock selection	000 : Reserved (Note 4) 001 : Reserved (Note 4) 010 : 57.1 [kHz] 011 : 29.9 [kHz] 100 : 15.3 [kHz] 101 : 7.72 [kHz] 110 : 3.88 [kHz] 111 : Reserved			at fc = 8 MHz (Output on SCL pin)		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 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: This I²C bus circuit does not support the Fast mode. It supports the Standard mode only. Although the I²C bus circuit itself allows the setting of a baud rate over 100 kbps, the compliance with the I²C specification is not guaranteed in that case.

Serial Bus Interface Data Buffer Register

SBIDBR (0021_H)

7	6	5	4	3	2	1	0

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

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

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	87CH74A/M74A 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 2-40. 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 : – 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 : Swich a mode to I2Cbus mode after confirming that input signals via port are high level.
Note 4 : SBICR2 has write-only register bits, which can not access any of in read-modify-write instructions such as bit operate, etc.
Note 5 : Clear bits 1 and 0 in SBICR2 to "0".

Serial Bus Interface Status Register

SBISR (0023_H)

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

(Initial value 0001 0000)

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

Figure 2-41. Serial Bus Interface Control Register 2 / Serial Bus Interface Status Register in the I^2Cbus Mode

(1) **Acknowledgment mode specification**

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

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

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

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

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

(2) **Number of transfer bits**

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

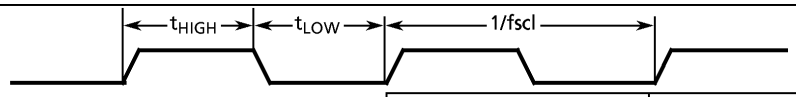
Since the BC is cleared to "000" by a start condition, a slave address and direction bit transmissions are executed in 8 bits. Other than these, the BC retains a specified value.

(3) **Serial clock**

a. Clock source

The SCK (bits 2 to 0 in the SBICR1) is used to select a maximum transfer frequency outputed on the SCL pin in the master mode. Set a communication baud rate that meets the I^2C bus specification, such as the shortest pulse width of t_{LOW}, based on the equations shown below.

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



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

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

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

Note : fc ; high-frequency clock

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

Figure 2-42. Clock Source

b. Clock synchronization

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

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

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

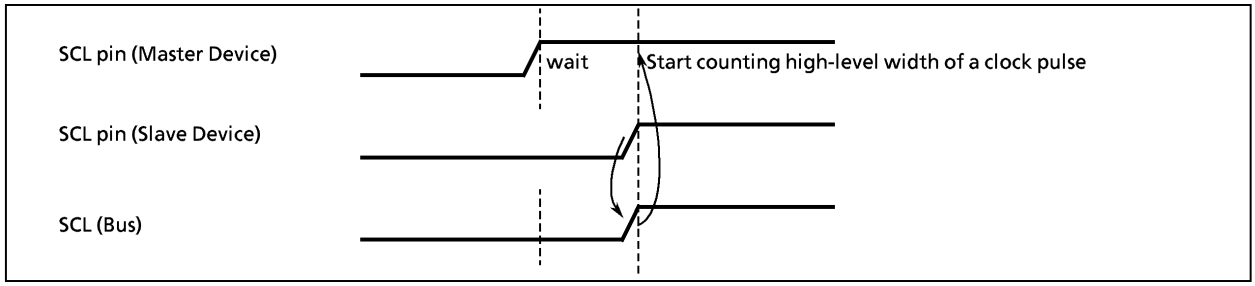


Figure 2-43. Clock Synchronization

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

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

(6) **Transmitter / receiver selection**
Set the TRX (bit 6 in the SBICR2) to "1" for operating the serial bus interface circuit as a transmitter. Clear the TRX to "0" for operation as a receiver. When data with an addressing format is transferred in the slave mode, the TRX is set to "1" by the hardware if the direction bit (R/W) sent from the master device is "1", and is cleared to "0" by the hardware if the bit is "0". In the master mode, after an acknowledge signal is returned from the slave device, the TRX is cleared to "0" by the hardware if a transmitted direction bit is "1", and is set to "1" by the hardware if it is "0". When an acknowledge signal is not returned, the current condition is maintained.
The TRX is cleared to "0" by the hardware after a stop condition on the bus is detected or the noise is detected.

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

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

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

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

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

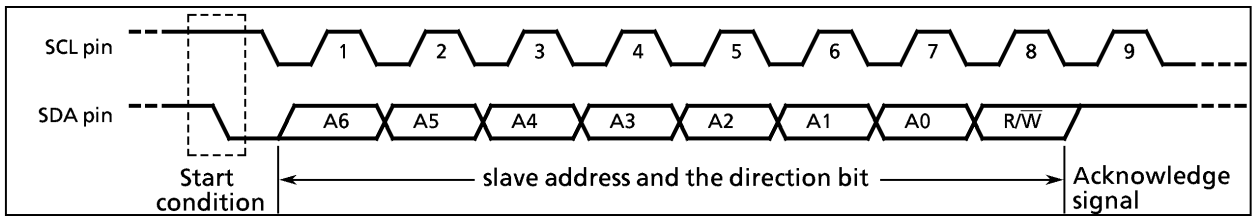


Figure 2-44. Start Condition Generation and Slave Address Generation

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

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

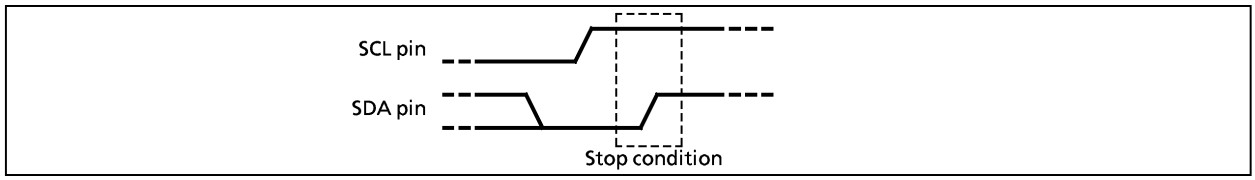


Figure 2-45. Stop Condition Generation

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

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

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

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

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

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

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

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

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

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

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

(10) **Noise detection monitor**

The I^2C bus is easy to be affected by noise, because the bus is driven by the open drain and the pull-up resistor.

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

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

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

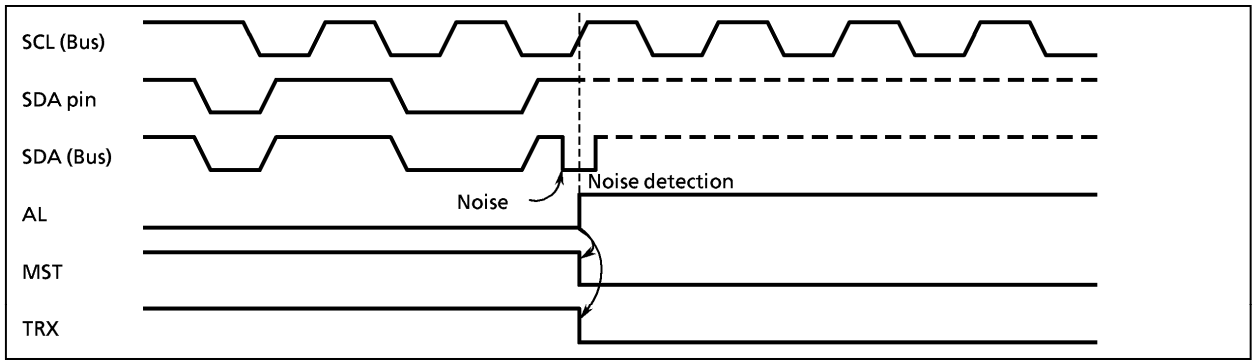


Figure 2-46. Noise Detection Monitor

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

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

(12) **GENERAL CALL detection monitor**

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

(13) **Last received bit monitor**

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

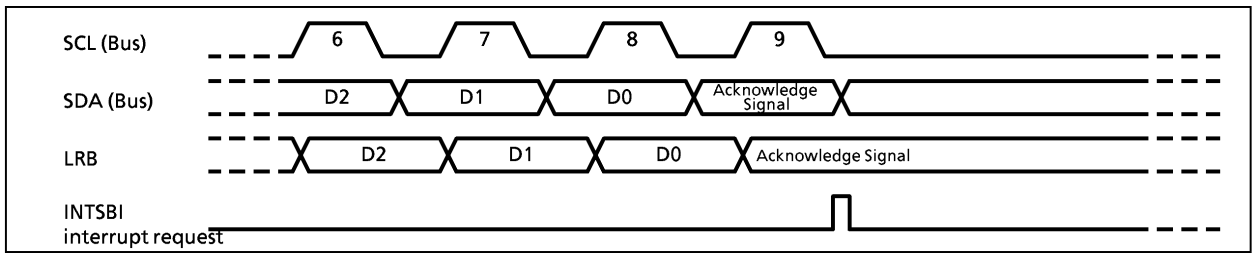


Figure 2-47. Last Received bit Monitor

(14) **Software reset function**

Software reset function is used to initialize SBI, when SBI is locked by external noise, etc. SWRST is set to "1, internal reset signal pulse is generated and inputted into SBI circuit.

All command registers and status registers are initialized to an initial value.

SWRST is automatically cleared to "0" after initialize SBI circuit.

2.9.5 Data Transfer in I^2C bus Mode

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

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

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

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

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

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

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

When the BB is "0", the start condition are generated and the slave address and the direction bit which are set to the SBIDBR are output on a bus by writing "1" to the MST, TRX, BB and PIN. An INTSBI interrupt request occurs at the 9th falling edge of the SCL clock cycle, and the PIN is cleared to "0". The SCL pin is pulled down to the 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 : Do not write a slave address to be output to the SBIDBR while data are transferred. If data is written to the SBIDBR, data to been outputting may be destroyed.

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

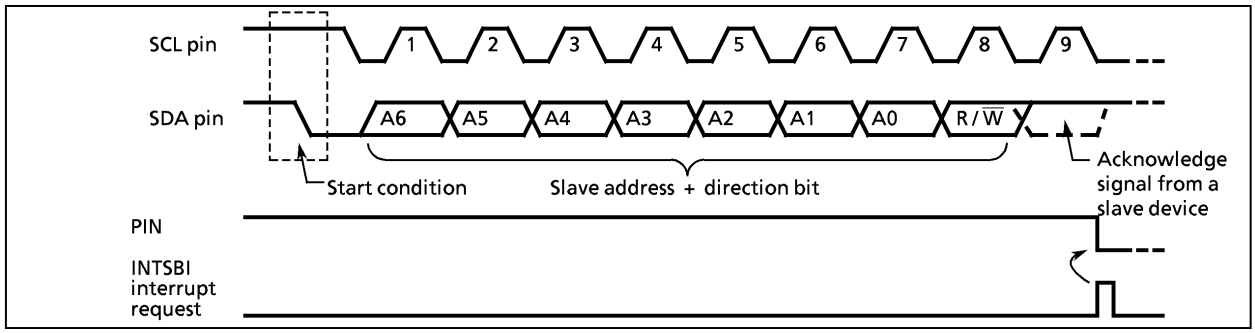


Figure 2-48. Start Condition Generation and Slave Address Transfer

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

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

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

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

① When the TRX is "1" (Master mode)

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

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

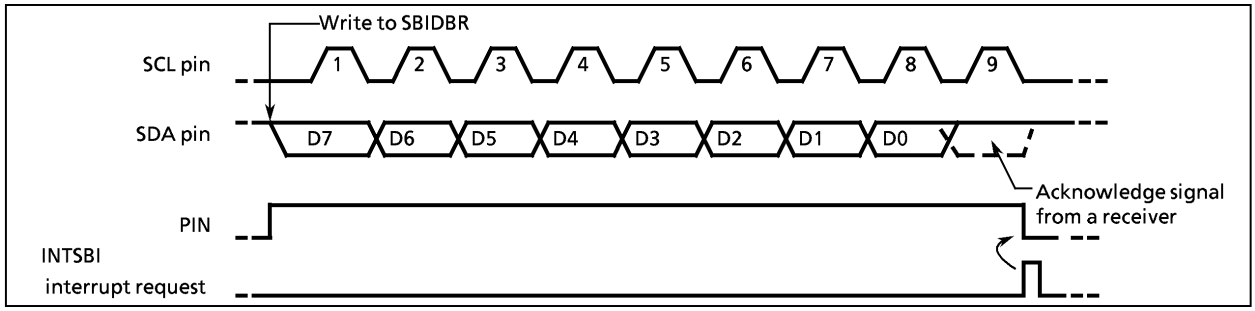


Figure 2-49. Example when BC = "000", ACK = "1" in Transmitter Mode

② When the TRX is "0" (Receiver mode)

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

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

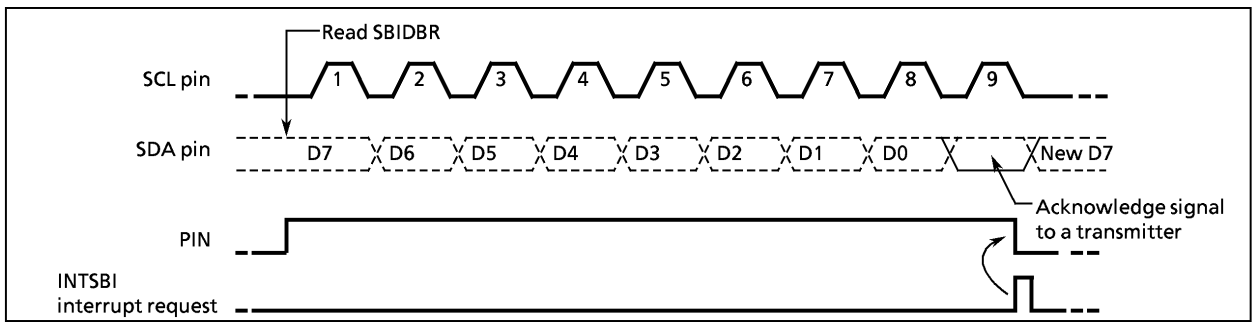


Figure 2-50. Example when BC = "000", ACK = "1" in Receiver Mode

In order to terminate transmitting data to a transmitter, clear the ACK to "0" before reading data which is 1 word before the last data to be received. The last data does not generate a clock pulse for the acknowledge signal. After the data is transmitted and an interrupt request has occurred, set the BC to "001" and read the data. The serial bus interface circuit generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA line on a bus keeps the 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 serial bus interface circuit generates a stop condition (Refer to 2.9.5. (4)) and terminates data transfer.

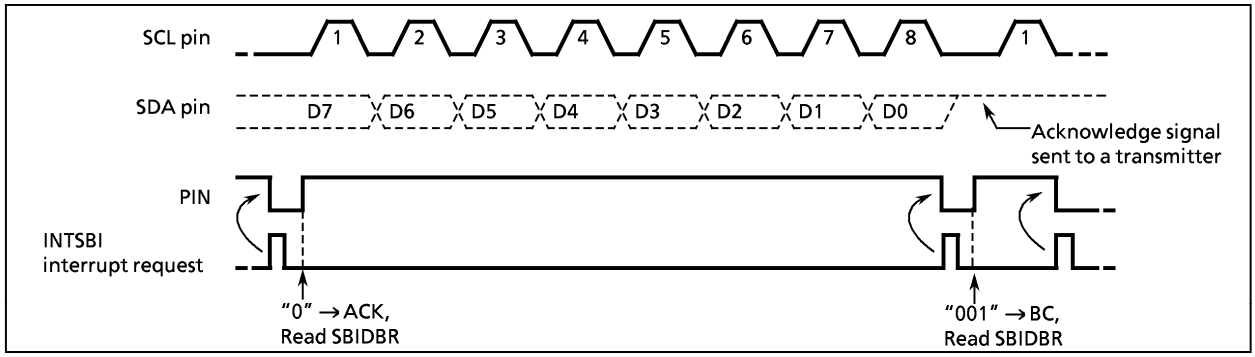


Figure 2-51. Termination of Data Transfer in Master Receiver Mode

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

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

Table 2-8. Operation in the Slave Mode

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

(4) **Stop Condition Generation**

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

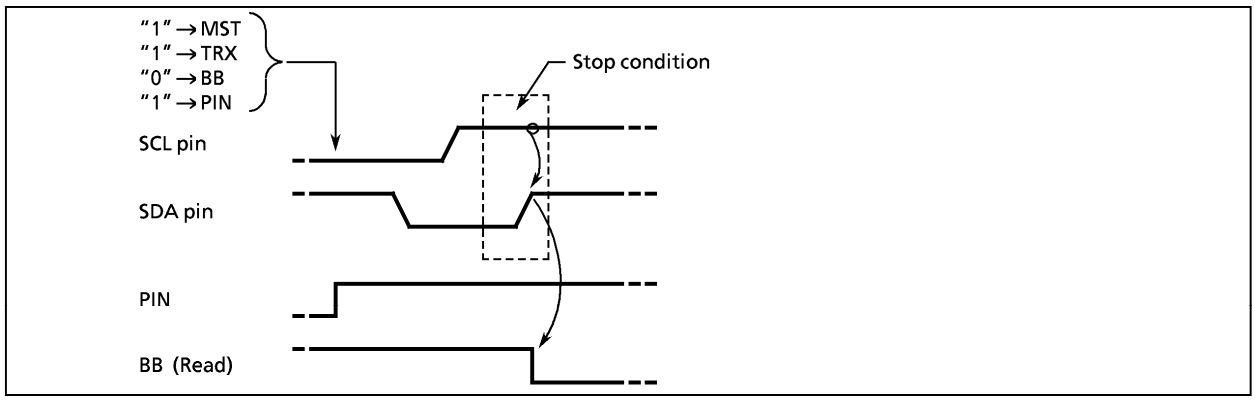


Figure 2-52. Stop Condition Generation

(5) **Restart**

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

Clear "0" to the MST, TRX, and BB and set "1" to the PIN. The SDA pin retains the high level and the SCL pin is released. Since a stop condition is not generated on the bus, the bus is assumed to be in a busy state from other devices. Test the BB until it becomes "0" to check that the SCL pin of the serial bus interface circuit is released. Test the LRB until it becomes "1" to check that the SCL line of the bus is not set to low level by other devices. After confirming that the bus stays in a free state, generate a start condition with procedure (2).

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

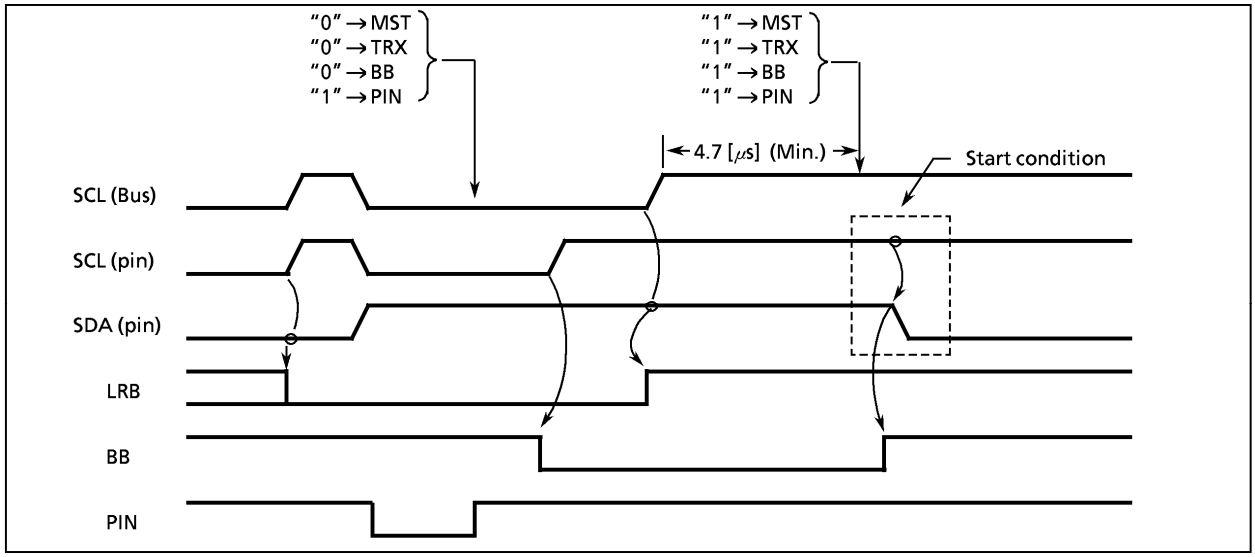


Figure 2-53. Timing Diagram when Restarting

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

*Note 1 : * ; don't care*
Note 2 : Set the SIOS to "0" when setting the transfer mode or serial clock.
Note 3 : SBICR1 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

Serial Bus Interface Data Buffer Register

SBIDBR (0021_H)

7	6	5	4	3	2	1	0

Read / Write

Note : Cannot read the data which was written into SBIDBR, since a write data buffer and a read buffer are independent in SBIDBR. Therefore, cannot access it any of in read-modify-write instruction 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 signal via port is high level.
Note 4 : SBICR2 is write-only register, which cannot access any of in read-modify-write instruction 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"

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

Figure 2-54. Control Register / Data Buffer Register / Status Register in SIO Mode

(1) Serial clock

a. Clock source

The SCK(bits 2 to 0 in 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 $\overline{SCK0}$ pin. The $\overline{SCK0}$ 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.

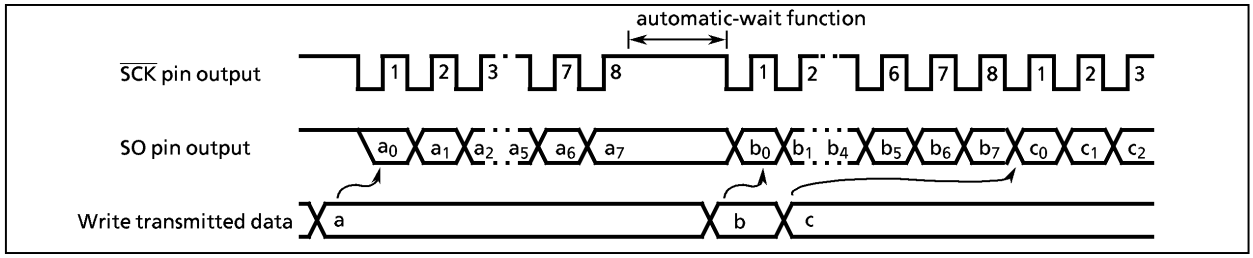


Figure 2-55. Automatic Wait Function

② External clock (SCK = "111")

An external clock supplied to the $\overline{SCK0}$ pin is used as the serial clock. In order to ensure shift operation, a pulse width of at least 4 machine cycles is required for both high and low levels in the serial clock. The maximum data transfer frequency is 250 kHz (fc = 8 MHz).

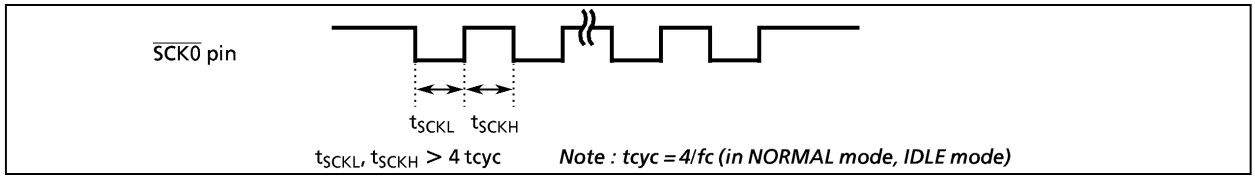


Figure 2-56. The Maximum Data Transfer Frequency in The External Clock Input

b. Shift edge

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

① Leading edge

Data is shifted on the leading edge of the serial clock (at a falling edge of the $\overline{SCK0}$ pin input / output).

② Trailing edge

Data is shifted on the trailing edge of the serial clock (at a rising edge of the $\overline{SCK0}$ pin input / output).

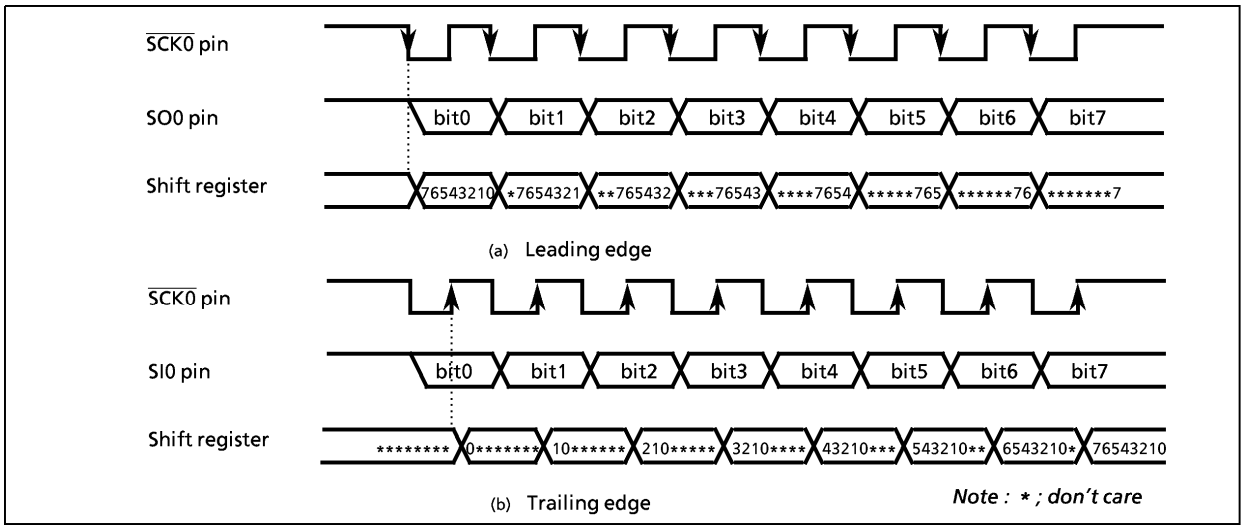


Figure 2-57. Shift Edge

(2) Transfer mode

The SIOM (bits 5 and 4 in SBICR) 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 data to the SBIDBR.

After the 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 SO0 pin in synchronous with the serial clock, starting from the least significant bit (LSB). When the 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 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 $\overline{SCK}$.

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

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

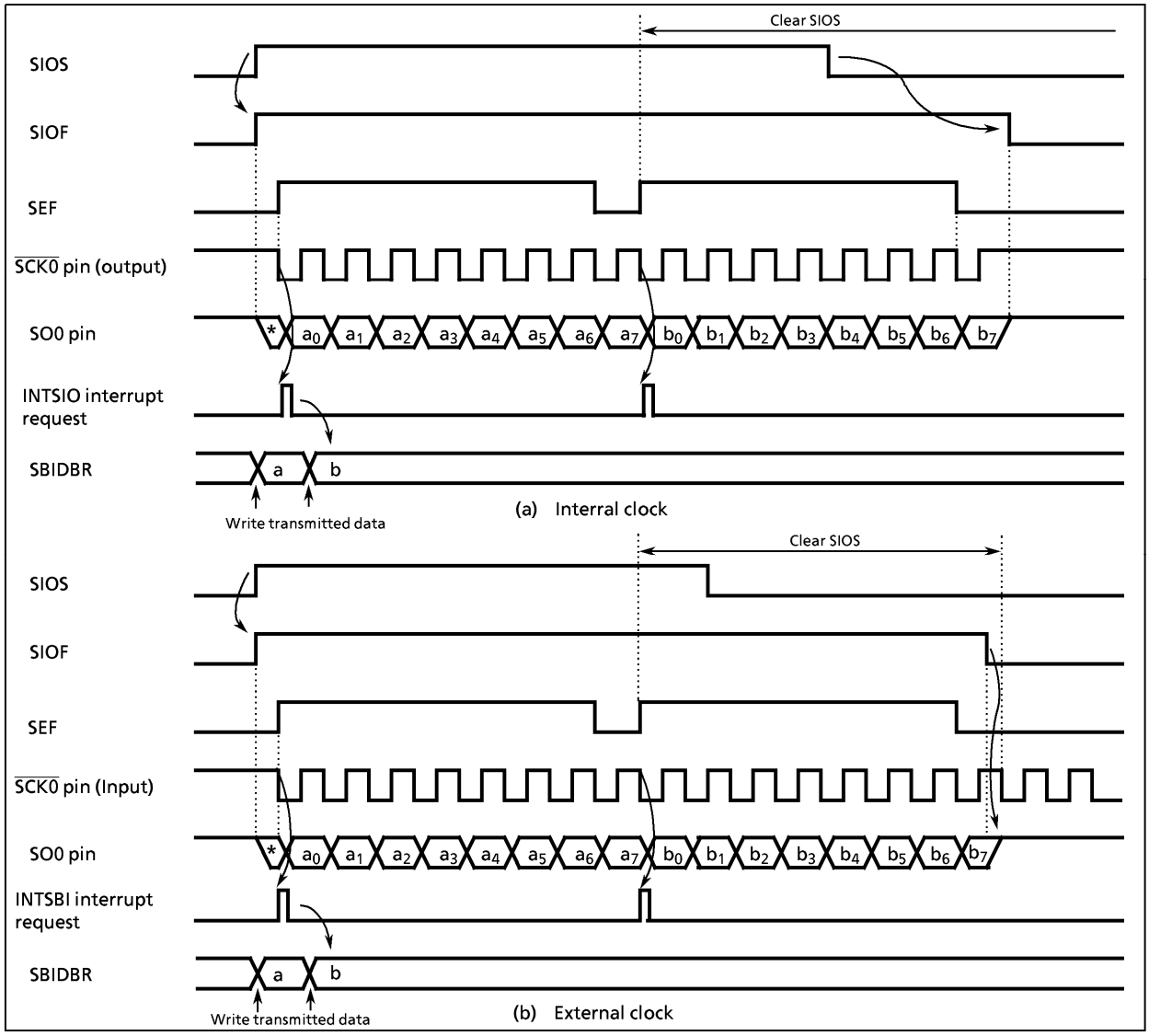


Figure 2-58. Transfer Mode

```
Example:  SIO0 transfer end command (External clock)
STEST1  :  TEST   (SBISR) . SEF          ;  If SEF = 1   then   loop
           JRS F , STEST1
STEST2  :  TEST   (P3) . 6               ;  If SCK = 0   then   loop
           JRS T , STEST2
           LD  (SBICR1) , 00000111B      ;  SIOS ← 0
```

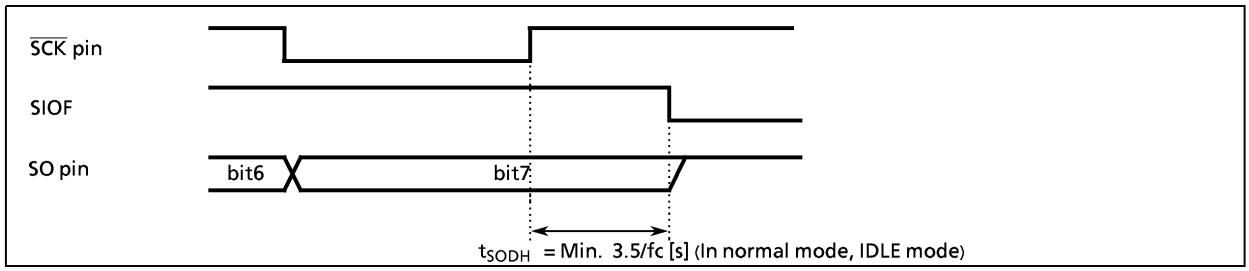


Figure 2-59. Transmitted Data Hold Time at End of Transmit

b. 8-bit receive mode

Set a control register to a receive mode and the SIOS to "1" for switching to a 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 before new data is transferred to the SBIDBR. 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.

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 $\overline{SCK}$.

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

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

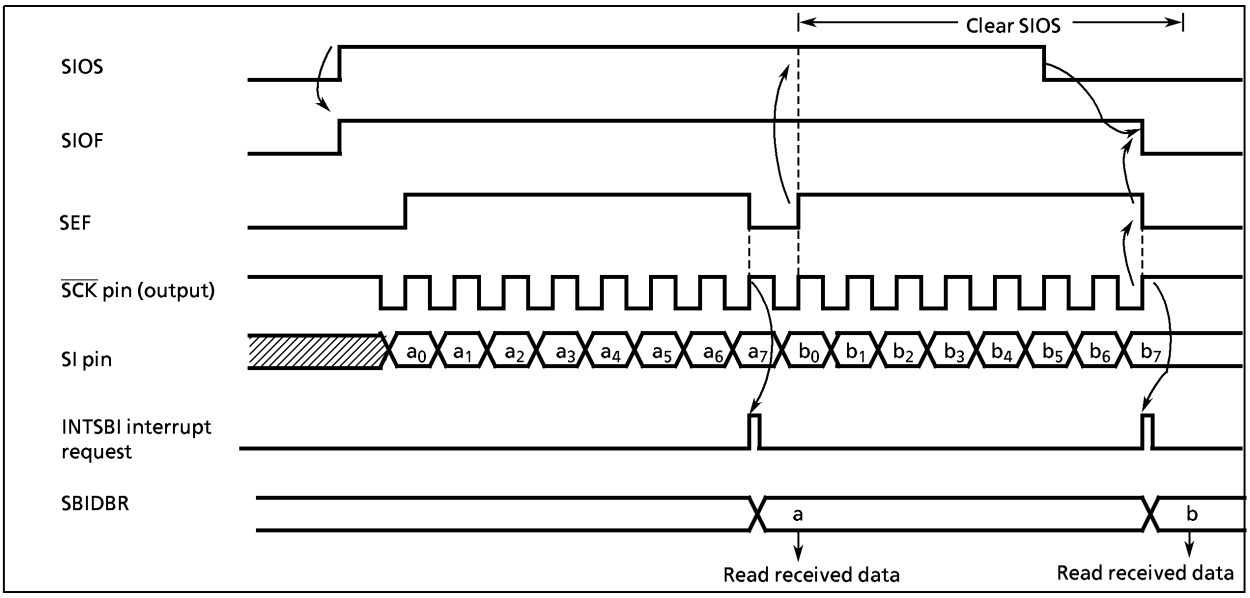


Figure 2-60. Receive Mode (Example: Internal clock)

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 shift register to the SBIDBR, and the INTSBI interrupt request occurs. The interrupt service program reads the received data from the data buffer register and writes data to be transmitted. The SBIDBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

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

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

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

Transmitting / receiving data is ended by cleaning the SIOS to "0" by the INTSBI interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, received data is transferred to the SBIDBR in complete blocks. The transmit / receive mode ends when the transfer is complete. In order to confirm if data is surely transmitted / received by the program, set the SIOF (bit 3 in SBISR) to be sensed. The SIOF becomes "0" after transmitting / receiving is complete. When the SIOINH is set, transmitting / receiving data stops. The SIOF turns "0".

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

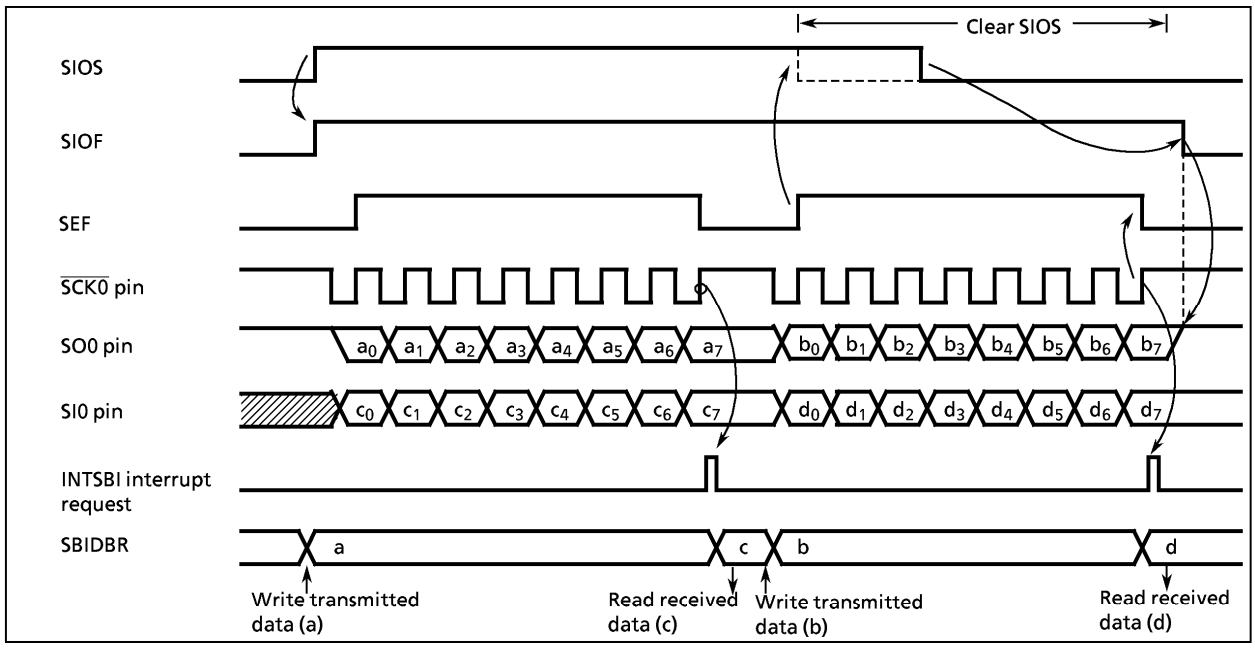


Figure 2-61. Transmit / Receive Mode (Example: Internal clock)

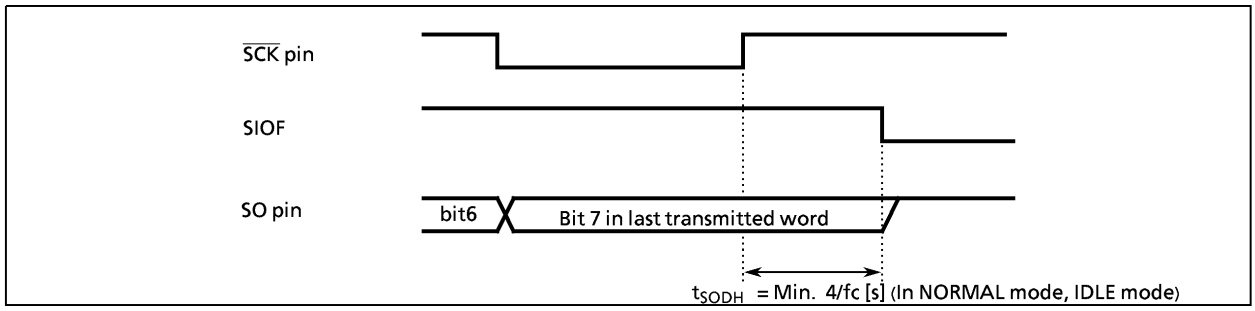


Figure 2-62. Transmitted Data Hold Time at End of Transmit / Receive

2.10 Serial Interface (SIO1)

The 87CH74A each have one clocked-synchronous 8-bit serial interface (SIO1). Each serial interface has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

The serial interfaces are connected to external devices via pins P02 (SO1), P01 (SI1), P00 ($\overline{SCK1}$) for SIO1. The serial interface pins are also used as port P0. When used as serial interface pins, the output latches of these pins should be set to "1". In the transmit mode, pins P01 can be used as normal I/O ports, and in the receive mode, the pins P02 can be used as normal I/O ports.

2.10.1 Configuration

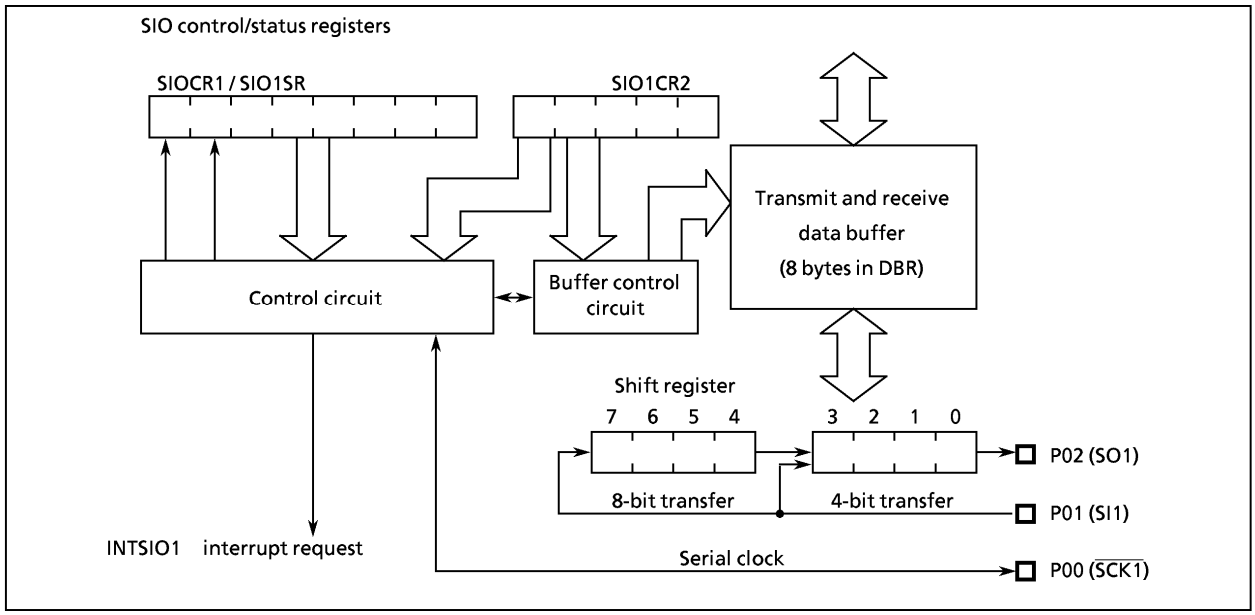


Figure 2-63. Serial Interfaces

2.10.2 Control

The serial interfaces are controlled by SIO1 control registers (SIO1CR1/SIO1CR2). The serial interface status can be determined by reading SIO status registers (SIO1SR).

The transmit and receive data buffer is controlled by the BUF (bits 2-0 in SIO1CR2). The data buffer is assigned to addresses $0FF8_H$ - $0FFF_H$ for SIO1 in the DBR area, and can continuously transfer up to 8 words (bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO1) is generated.

When the internal clock is used as the serial clock in the 8-bit receive mode and the 8-bit transmit/receive mode, a fixed interval wait can be applied to the serial clock for each word transferred. Four different wait times can be selected with WAIT (bits 4 and 3 in SIO1CR2).

SIO Control Registers 1

SIO1CR1 (0027_H)

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

(Initial value : 0000 0000)

SIOS	Indicate transfer start/stop	0 : Stop 1 : Start	write only
SIOINH	Continue/abort transfer	0 : Continue transfer 1 : Abort transfer (automatically cleared after abort)	
SIOM	Transfer mode select	000 : 8-bit transmit mode 010 : 4-bit transmit mode 100 : 8-bit transmit/receive mode 101 : 8-bit receive mode 110 : 4-bit receive mode	
SCK	Serial clock select	000 : Internal clock fc / 2^{13} or fs / 2^5 [Hz] 001 : Internal clock fc / 2^8 010 : Internal clock fc / 2^6 011 : Internal clock fc / 2^5 } (Output on $\overline{SCK}$ pin) 111 : External clock (input from $\overline{SCK}$ pin)	

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

Note 2 : Set SIOS to "0" and SIOINH to "1" when setting the transfer mode or serial clock.

Note 3 : SIO1CR1 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO1 Control Registers 2

SIO1CR2 (0028_H)

7	6	5	4	3	2	1	0
			WAIT		BUF		

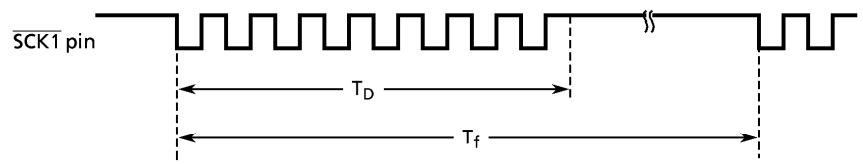
(Initial value: ***0 0000)

WAIT	Wait control	00 : $T_f = T_D$ (no-WAIT) 01 : $T_f = 2T_D$ 10 : $T_f = 4T_D$ 11 : $T_f = 8T_D$ } (WAIT)	Write only
BUF	Number of transfer words	Buffer address used SIO 000 : 1 word transfer — $0FF8_H$ 001 : 2 words transfer — $0FF8 - 0FF9_H$ 010 : 3 words transfer — $0FF8 - 0FFA_H$ 011 : 4 words transfer — $0FF8 - 0FFB_H$ 100 : 5 words transfer — $0FF8 - 0FFC_H$ 101 : 6 words transfer — $0FF8 - 0FFD_H$ 110 : 7 words transfer — $0FF8 - 0FFE_H$ 110 : 8 words transfer — $0FF8 - 0FFF_H$	

*Note 1 : *; don't care*

Note 2 : WAIT is valid only in the 8-bit transmit / receive and 8-bit receive modes.

Note 3 : T_f; frame time, T_D ; data transfer time



Note 4 : The lower 4 bits of each buffer are used during 4-bit transfers. Zeros (0) are stored to the upper 4bits when receiving.

Note 5 : Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. For example, in the case of SIO, the first buffer address transmitted is $0FF8_H$.

Note 6 : The value to be loaded to BUF is held after transfer is completed.

Note 7 : SIO1CR2 are write-only registers, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO1, SIO2 Status Registers

SIO1SR (0020_H)

7	6	5	4	3	2	1	0
SIOF	SEF	"1"	"1"	"1"	"1"	"1"	"1"

SIOF	Serial transfer operating status monitor	0 : Transfer terminated 1 : Transfer in process (After SIOS is cleared "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 2-64. SIO Control Registers and Status Registers

(1) Serial Clock

a. Clock Source

SCK (bits 2 - 0 in SIO1CR1) is able to select the following:

① Internal Clock

Any of four frequencies can be selected. The serial clock is output to the outside on the $\overline{SCK1}$ pin. The $\overline{SCK}$ pin goes high when transfer starts.

When data writing (in the transmit mode) or reading (in the receive mode or the transmit/receive mode) cannot keep up with the serial clock rate, there is a wait function that automatically stops the serial clock and holds the next shift operation until the read/write processing is completed.

Table 2-9. Serial Clock Rate

Serial clock			Maximum transfer rate	
NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode		
DV7CK = 0	DV7CK = 1		At fc = 8 MHz	At fs = 32.768 kHz
$fc / 2^{13}$ [Hz]	$fs / 2^5$ [Hz]	$fs / 2^5$ [Hz]	0.95 Kbit/s	1 Kbit/s
$fc / 2^8$	$fc / 2^8$	–	30.5	
$fc / 2^6$	$fc / 2^6$	–	122	
$fc / 2^5$	$fc / 2^5$	–	244	

Note : 1K bit = 1024 bit

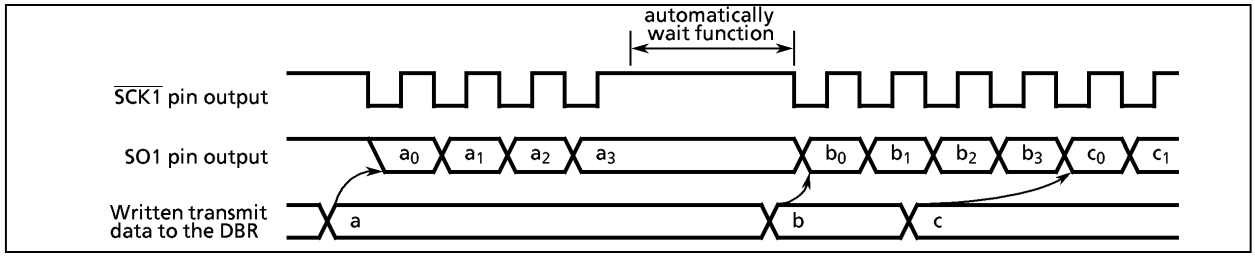
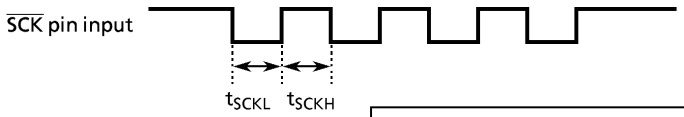


Figure 2-66. Clock Source (Internal Clock)

② External Clock

An external clock connected to the $\overline{SCK1}$ pin is used as the serial clock. In this case, the P00 ($\overline{SCK1}$) output latch must be set to "1". To ensure shifting, a pulse width of at least 4 machine cycles is required. Thus, the maximum transfer speed is 244K-bit/s. (at fc = 8 MHz).



Note : tcyc = 4/fc (In NORAML1/2, IDLE1/2 modes)
4/fs (In SLOW, SLEEP modes)

b. Shift edge

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

① Leading Edge

Transmitted data are shifted on the leading edge of the serial clock (falling edge of the $\overline{SCK1}$ pin input/output).

② Trailing Edge

Received data are shifted on the trailing edge of the serial clock (rising edge of the $\overline{SCK1}$ pin input/output).

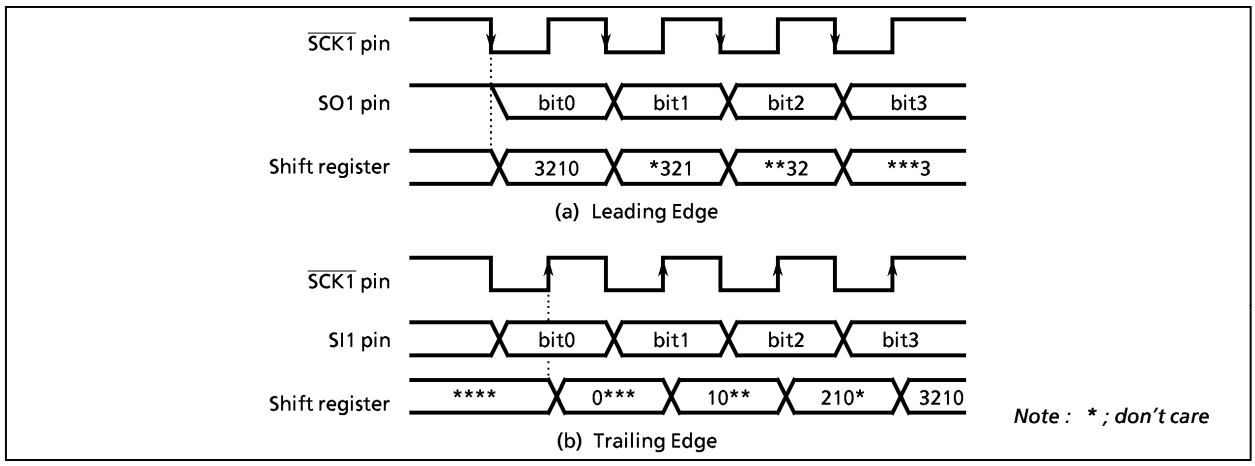


Figure 2-67. Shift Edge

(2) Number of Bits to Transfer

Either 4-bit or 8-bit serial transfer can be selected. When 4-bit serial transfer is selected, only the lower 4 bits of the transmit/receive data buffer register are used. The upper 4 bits are cleared to "0" when receiving.

The data is transferred in sequence starting at the least significant bit (LSB).

(3) Number of Words to Transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred is loaded to BUF in SIOBCR. An INTSIO interrupt is generated when the specified number of words has been transferred. If the number of words is to be changed during transfer, the serial interface must be stopped before making the change.

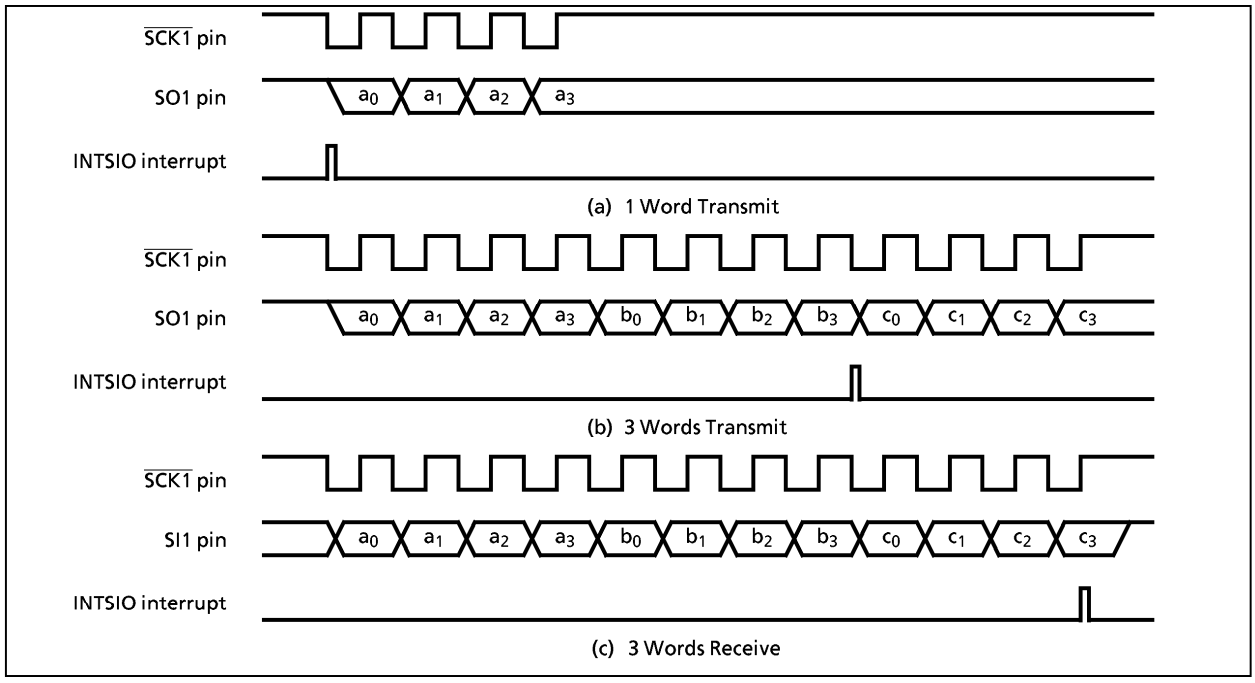


Figure 2-68. Number of Bits to Transfer (Example : 4-bit serial transfer)

(4) Transfer Mode

SIOM (bits 5 - 3 in SIO1CR1) is used to select the transmit, receive, or transmit/receive mode.

a. **4-bit and 8-bit Transmit** Modes

In these modes, the SIO1CR1 is set to the transmit mode and then the data to be transmitted first are written to the data buffer registers (DBR). After the data are written, the transmission is started by setting SIOS to "1". The data are then output sequentially to the SO pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO (buffer empty) interrupt is generated to request the next transmitted data.

When the internal clock is used, the serial clock will stop and an automatic-wait will be initiated if the next transmitted data are not loaded to the data buffer register by the time the number of data words specified with the BUF has been transmitted. Writing even one word of data cancels the automatic-wait; therefore, when transmitting two or more words, always write the next word before transmission of the previous word is completed.

Note : Waits are also canceled by writing to a DBR not being used as a transmit data buffer register; therefore, during SIO do not use such DBR for other applications.

When an external clock is used, the data must be written to the data buffer register before shifting next data. Thus, the transfer speed is determined by the maximum delay time from the generation of the interrupt request to writing of the data to the data buffer register 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 $\overline{SCK1}$.
The transmission is ended by clearing SIOS to "0" or setting SIOINH to "1" buffer empty interrupt service program. That the transmission has ended can be determined from the status of SIOF (bit 7 in SIO1SR) because SIOF is cleared to "0" when a transfer is completed.
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".
When an external clock is used, it is also necessary to clear SIOS to "0" before shifting the next data; otherwise, dummy data will be transmitted and the operation will end.

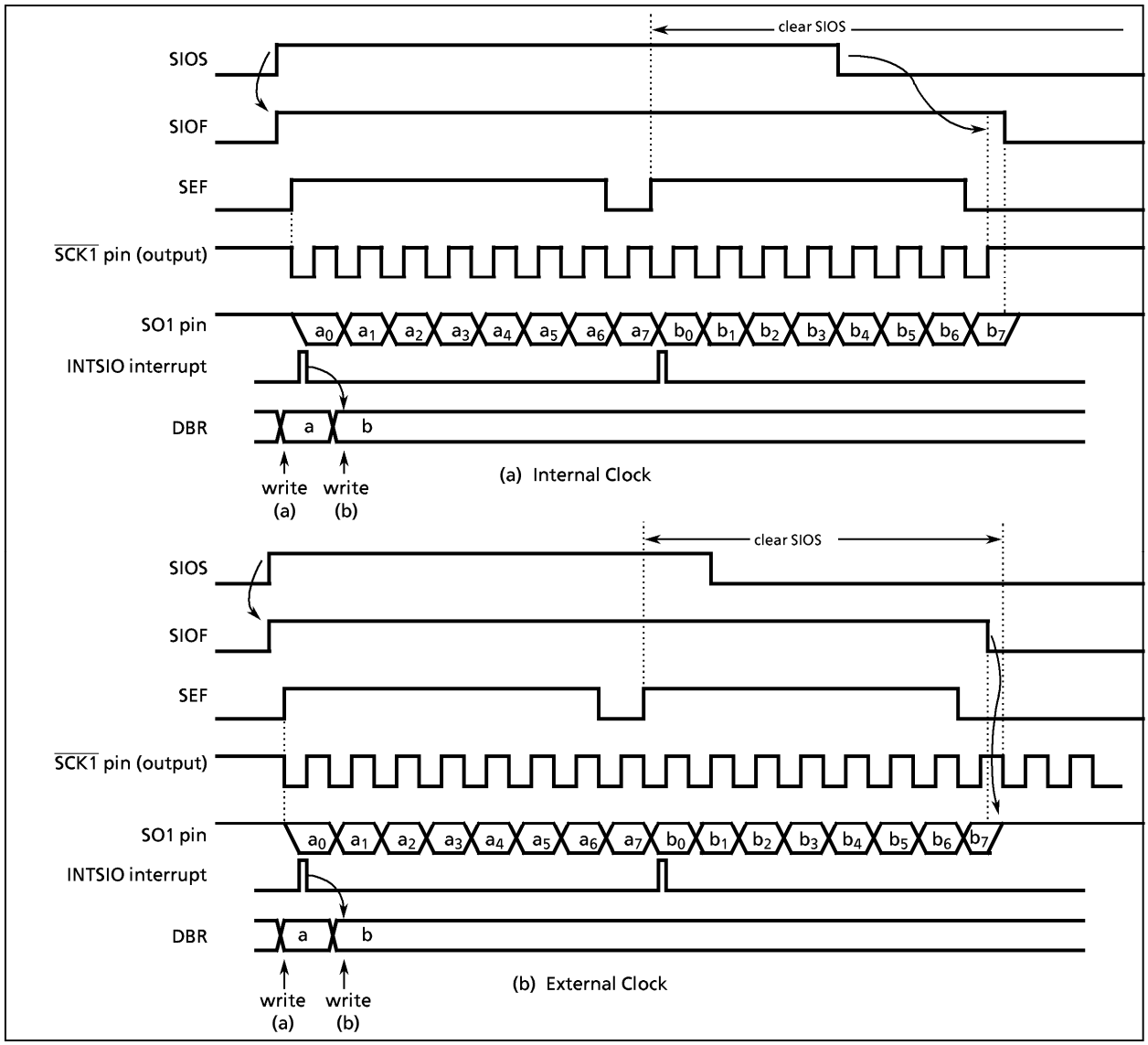


Figure 2-69. Transfer Mode (Example: 8-bit, 1 Word Transfer)

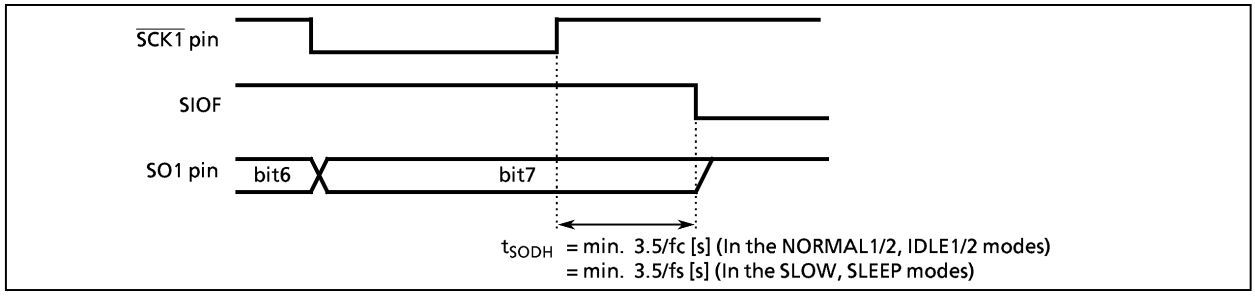


Figure 2-70. Transmitted Data Hold Time at End of Transmit

b. **4-bit and 8-bit Receive** Modes

After setting the control registers to the receive mode, set SIOS to "1" to enable receiving. The data are then transferred to the shift register via the SI1 pin in synchronous with the serial clock. When one word of data has been received, it is transferred from the shift register to the data buffer register (DBR). When the number of words specified with the BUF has been received, an INTSIO (buffer full) interrupt is generated to request that these data be read out. The data are then read from the data buffer registers by the interrupt service program.

When the internal clock is used, and the previous data are not read from the data buffer register before the next data are received, the serial clock will stop and an automatic-wait will be initiated until the data are read. A wait will not be initiated if even one data word has been read.

Note : Waits are also canceled by reading a DBR not being used as a received data buffer register is read; therefore, during SIO do not use such DBR for other applications.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, the previous data are read before the next data are transferred to the data buffer register. If the previous data have not been read, the next data will not be transferred to the data buffer register and the receiving of any more data will be canceled. When an external clock is used, the maximum transfer speed is determined by the delay between the time when the interrupt request is generated and when the data received have been 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 SIOS is cleared, the current data are transferred to the buffer in 4-bit or 8-bit blocks. The receiving mode ends when the transfer is completed. SIOF is cleared to "0" when receiving is ended and thus can be sensed by program to confirm that receiving has ended.

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 : The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIOS to "0", read the last data and then switch the transfer mode.

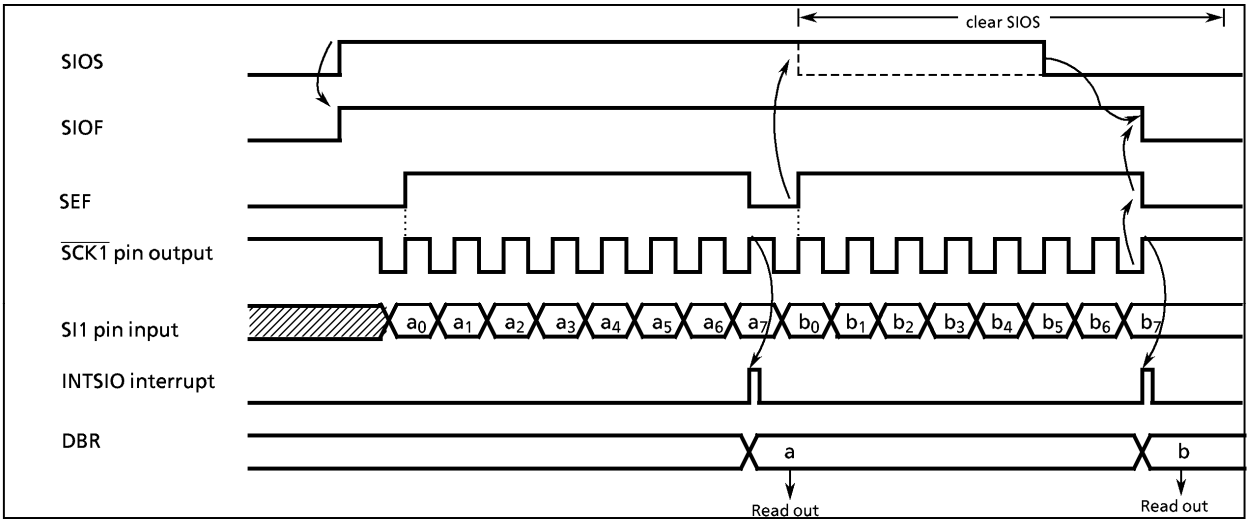


Figure 2-71. Receive Mode (Example : 8-bit, 1 word, internal clock)

c. **8-bit Transmit/Receive** Mode

After setting the control registers to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable transceiving by setting SIOS to "1". When transmitting, the data are output from the SO pin at leading edges of the serial clock. When receiving, the data are input to the SI pin at the trailing edges of the serial clock. 8-bit data are transferred from the shift register to the data buffer register. An INTSIO interrupt is generated when the number of data words specified with the BUF has been transferred. The interrupt service program reads the received data from the data buffer register and then writes the data to be transmitted. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the received data.

When the internal clock is used, a wait is initiated until the received data is read and the next data is written. A wait will not be initiated if even one data word has been written.

Note : The wait is also canceled by writing to a DBR not being used as a transmit data buffer resisters; therefore, during SIO do not use such DBR for other applications.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, it is necessary to read the received data and write the data to be transmitted next before starting the next shift operation. When an external clock is used, the transfer speed is determined by the maximum delay between generation of an interrupt request and the received data are read and the data to be transmitted next are written.

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

The transmit/receive operation is ended by clearing SIOS to "0" or setting SIOINH to "1" in interrupt service program. When SIOS is cleared, the current data are transferred to the data buffer register in 8-bit blocks. The transmit mode ends when the transfer is completed. SIOF is cleared to "0" when receiving is ended and thus can be sensed by program to confirm that receiving has ended.

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 : The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIOS to "0", read the last data and then switch the transfer mode.

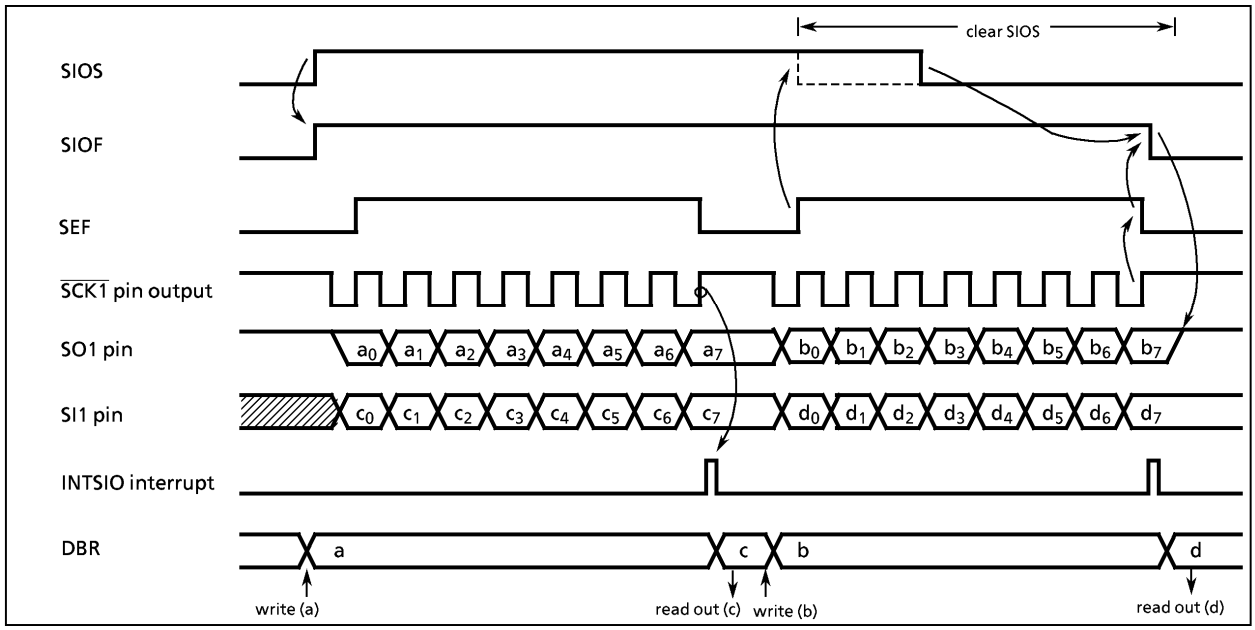


Figure 2-72. Transmit/Receive Mode (Example : 8-bit, 1word, internal clock)

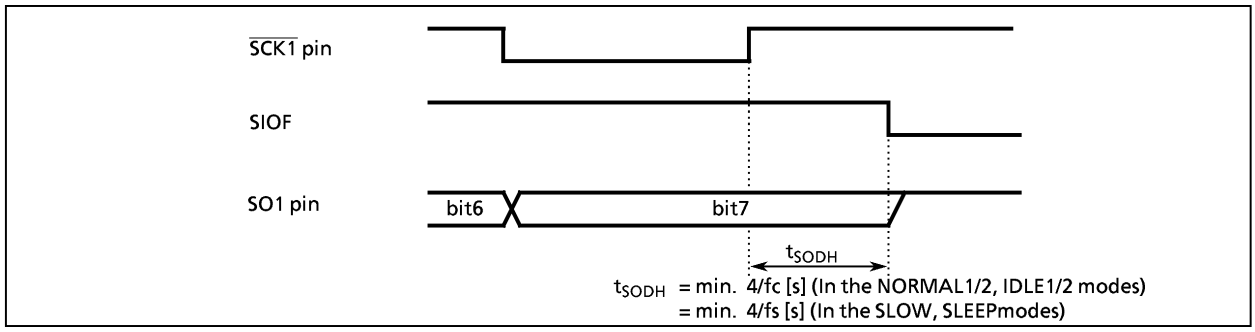


Figure 2-73. Transmitted Data Hold Time at End of Transmit/receive

2.11 8-bit A/D Converter (ADC)

The 87CH74A/M74A each have an 8-channel multiplexed-input 8-bit successive approximate type A/D converter with sample and hold.

2.11.1 Configuration

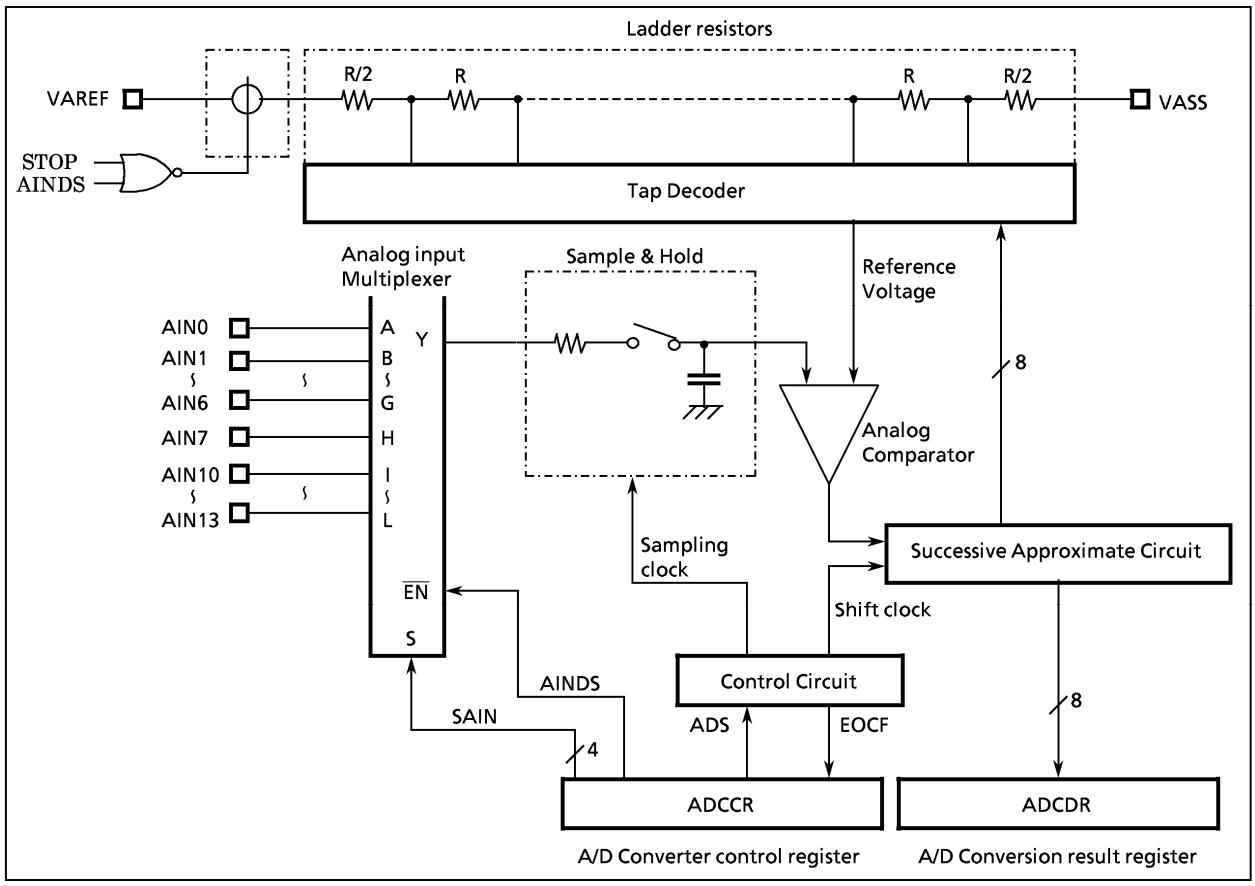


Figure 2-74. A/D Converter

2.11.2 Control

The A/D converter is controlled by an A/D converter control register (ADCCR).

A/D Conversion Result Register

	7	6	5	4	3	2	1	0	
ADCDR ($000F_H$)									Read only

A/D Converter Control Register

ADCCR ($000E_H$)

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

(Initial 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	R/W
AINDS	Analog input control	0 : Enable 1 : Disable	
ACK	Conversion time select	0 : 23 μs (at fc = 8 MHz) 1 : 92 μs (at fc = 8 MHz)	
ADS	A/D conversion start	0 : – 1 : A/D conversion start	
EOCF	End of A/D conversion flag	0 : Under conversion or Before conversion 1 : End of conversion	R

*Note 1 : * ; don't care*
Note 2 : Select analog input when A/D converter stops.
Note 3 : The ADS is automatically cleared to "0" after starting conversion.
Note 4 : The EOCF is cleared to "0" when reading the ADCDR.
Note 5 : The EOCF is read-only.

Figure 2-76. A/D converter control register and A/D conversion result register

2.11.3 Operation

Apply analog reference voltage to pins VAREF and VASS.

(1) Start of A/D conversion

First, set the corressponding P4CR and P5CR bit to "0" for analog input. Clear the AINDS (bit 4 in ADCCR) to "0" and select one of eight analog input AIN13-AIN0 with the SAIN (bits 3-0 in ADCCR).

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

A/D conversion is started by setting the ADS (bit 6 in ADCCR) to "1".
Conversion is accomplished in 46 machine cycles (184/fc [s]). When fc is 8 MHz, it needs 23 μs.
The EOCF (bit 7 in ADCCR) is set to "1" at end of conversion.
Setting ADS to "1" in A/D conversion starts converting over again. An analog input voltage is sampled at intervals of four cycles after starting A/D conversion.

Note : To keep same level of an analog input during 4 Machine Cycle Time is necessary for charging the electron to the sample hold circuit which a resistor (typ. 5k) and a capacitor (typ. 12pF).

(2) Reading of A/D conversion result

After the end of conversion, read the conversion result from the ADCDR.
The EOCF is automatically cleared to "0" when reading the ADCDR.
Undefined value is read in A/D conversion.

(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 ADCDR contents become indefinite. After returning from STOP mode, EOCF is kept to be cleared to "0".

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

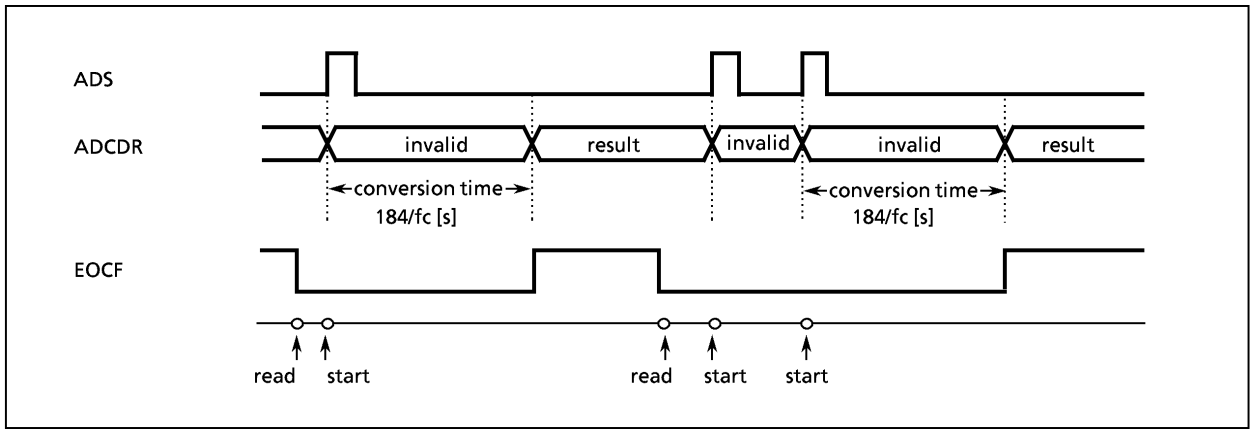


Figure 2-77. A/D Conversion Timing Chart

Example:

```
                ; AIN SELECT
                LD          (ADCCR) , 00100100B     ;   selects AIN4
                ; A/D CONVERT START
                SET         (ADCCR) . 6             ;   ADS = 1
SLOOP    :      TEST        (ADCCR) . 7             ;   EOCF = 1 ?
                JRS         T, SLOOP
                ; RESULT DATA READ
                LD          (9EH), (ADCDR)
```

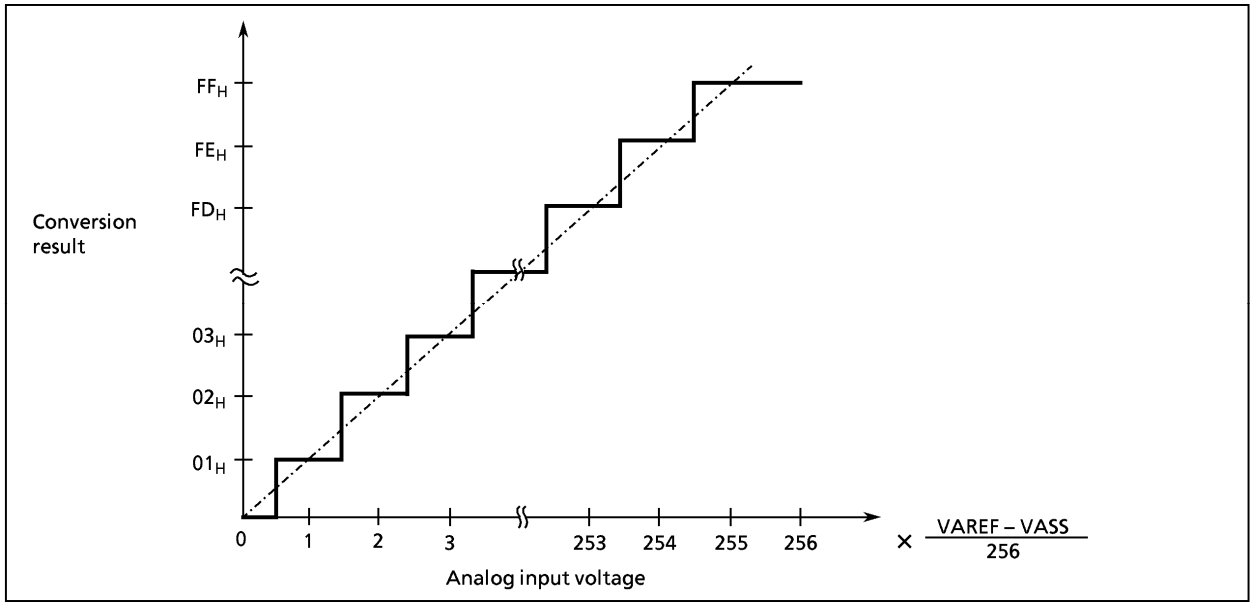


Figure2-78. Analog Input Voltage vs A/D Conversion Result (typ.)

2.12 Vacuum Fluorescent Tube (VFT) Driver Circuit

The 87CH74A/M74A features built-in high-breakdown voltage output buffers for directly driving fluorescent tubes, and a display control circuit used to automatically transfer display data to the output port.

The segment and the digit, as it is the VFT drive circuit which included in the usual products, are not allocated. The segment and the digit can be freely allocated in the timing (T0 to T15) which is specified according to the display tube types and the layout.

2.12.1 Functions

(1) 37 high-breakdown voltage output buffers built-in.
- Large current output pin (typ. 20mA) 16 (V0 to V15)
- Middle current output pin (typ. 8mA) 21 (V16 to V36)

There is also the VKK pin used for the VFT drive power supply.

(2) The dynamic lighting system makes it possible to select 1 to 16 digits (T0 to T15) by program.

(3) Pins not used for VFT driver can be used as general-purpose ports.
- Pins can be selected using the VSEL (bits 4 to 0) in VFT control register1 bit by bit.

(4) Display data (80 bytes in DBR) are automatically transferred to the VFT output pin.

(5) Brightness level can be adjusted in 8 steps using the dimmer function.

(6) Four types (fc/2^{12} to fc/2^{9}) of display time can be selected.

2.12.2 Configuration

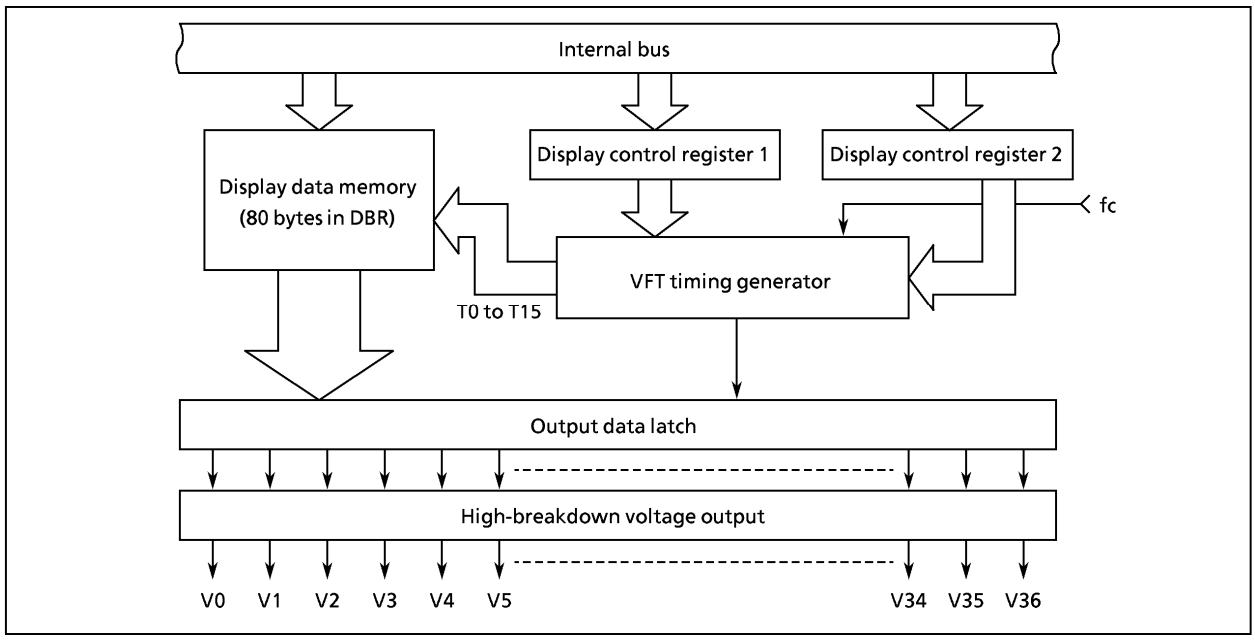


Figure 2-51.

2.12.3 Control

The VFT driver circuit is controlled by the VFT control registers (VFTCR1, VFTCR2). Reading VFTSR determines the VFT operating status.

Switching the mode from NORMAL1/2 to SLOW or STOP puts the VFT driver circuit into blanking state (BLK is set to "1" and EXEY is cleared to "0" ; values set in the VFT control registers except BLK and EKEY are maintained), and sets segment outputs and digit outputs are cleared to "0". Thus, ports P6 to P9, and PD function as general-purpose output ports with pull-down.

VFT control register 1

VFTCR1 (0029_H)

7	6	5	4	3	2	1	0
BLK	SDT				VSEL		

(initial value: 1000 0000)

BLK	VFT display control	0 : Display enable 1 : Disable	write only
SDT	Display time select (tdisp) (Display time of 1 digit)	00 : 2^9/fc 01 : 2^{10}/fc 10 : 2^{11}/fc 11 : 2^{12}/fc	
VSEL	Automatic display select (When using VFT driver (automatic display), V31 to V0 are only used to output VFT.) Pins which are not selected by the output pins other than the above-mentioned pins can be used as general-purpose input/output pins. (When using as a general-purpose input/output pin, the display data which corresponds to the pin must be set to "0")	00000 : 32 (V31 to V0) 00001 : 33 (V32 to V0) 00010 : 34 (V33 to V0) 00011 : 35 (V34 to V0) 00100 : 36 (V35 to V0) 00101 : 37 (V36 to V0)	

Note 1 : fc ; high frequency clock

Note 2 : VFTCR1 is write-only register, which cannot use any of in read-modify-write instruction such as bit operate, etc.

Figure 2-80. VFT Control Register 1

VFT control register 2

VFTCR2 ($002A_H$)

7	6	5	4	3	2	1	0
DIM				STA			

(initial value: 000* 0000)

DIM	Dimmer time select	000 : (15/16) × tdisp (s) 001 : (14/16) × tdisp (s) 010 : (12/16) × tdisp (s) 011 : (10/16) × tdisp (s) 100 : (8/16) × tdisp (s) 101 : (6/16) × tdisp (s) 110 : (4/16) × tdisp (s) 111 : (2/16) × tdisp (s)	write only
STA	Number of state (display)	0000 : 1 display mode (T0) 0001 : 2 display mode (T1 to T0) 0010 : 3 display mode (T2 to T0) 0011 : 4 display mode (T3 to T0) 0100 : 5 display mode (T4 to T0) 0101 : 6 display mode (T5 to T0) 0110 : 7 display mode (T6 to T0) 0111 : 8 display mode (T7 to T0) 1000 : 9 display mode (T8 to T0) 1001 : 10 display mode (T9 to T0) 1010 : 11 display mode (T10 to T0) 1011 : 12 display mode (T11 to T0) 1100 : 13 display mode (T12 to T0) 1101 : 14 display mode (T13 to T0) 1110 : 15 display mode (T14 to T0) 1111 : 16 display mode (T15 to T0)	

Note 1 : VFTCR2 is write-only register, which cannot use any of in read-modify-write instruction such as bit operate, etc.

Note 2 : Even if a number of the display digit is set a pin which is equal to the digit dose not output. It is necessary to write data to the data buffer which corresponds to the digit according to the display timing (T0 to T15).

*Note 3 : * ; don't care*

VFTSR (0029_H)

7	6	5	4	3	2	1	0
WAIT							

WAIT	VFT operational status monitor	0 : VFT display in operation 1 : VFT display operation disabled	read only

Figure 2-81. VFT control Register 2, VFT status register

(1) Setting of Display mode

VFT display mode is set by VFT control register 1 (VFTCR1) and VFT control register 2 (VFTCR2). VFT control register 1 (VFTCR1) sets 1 display time (tdisp) and the number of display lines (VSEL), and VFT control register 2 (VFTCR2) sets dimmer timer (DIM) and state (STA). (BLK of VFTCR1 must be set to "1".) The segments and the digits are not fixed, so that they can be freely allocated. However the number of states must be specified according to the number of digits of VFT which you use. Thought the layout of VFT display mode is freely allocated, the followings are recommended; usually, large current output (V0 to V15) is used for a digit, and middle current output (V16 to V36) is used for a segment.

(See Display operation in section 2.12.4 for display timing and data setting procedures.)

(2) Display data setting

Data are converted into VFT display data by instructions. The converted data stored in the display data buffer (addresses 0F80 to 0FCF in DBR) are automatically transferred to the VFT driver circuit, then transferred to the high-breakdown voltage output buffer. Thus, to change the display pattern, just change the data in the display data buffer.

Bits in the VFT segment (dot) and display data area correspond one to one. When data are set to 1, the segments corresponding to the bits light. The display data buffer is assigned to the DBR area shown in Figure 2-82. (The display data buffer can not be used as data memory)

bit	0 to 7	0 to 7	0 to 7	0 to 7	0 to 4	Timing
	$0F80_H$	$0F90_H$	$0FA0_H$	$0FB0_H$	$0FC0_H$	T0
	$0F81_H$	$0F91_H$	$0FA1_H$	$0FB1_H$	$0FC1_H$	T1
	$0F82_H$	$0F92_H$	$0FA2_H$	$0FB2_H$	$0FC2_H$	T2
	$0F83_H$	$0F93_H$	$0FA3_H$	$0FB3_H$	$0FC3_H$	T3
	$0F84_H$	$0F94_H$	$0FA4_H$	$0FB4_H$	$0FC4_H$	T4
	$0F85_H$	$0F95_H$	$0FA5_H$	$0FB5_H$	$0FC5_H$	T5
	$0F86_H$	$0F96_H$	$0FA6_H$	$0FB6_H$	$0FC6_H$	T6
	$0F87_H$	$0F97_H$	$0FA7_H$	$0FB7_H$	$0FC7_H$	T7
	$0F88_H$	$0F98_H$	$0FA8_H$	$0FB8_H$	$0FC8_H$	T8
	$0F89_H$	$0F99_H$	$0FA9_H$	$0FB9_H$	$0FC9_H$	T9
	$0F8A_H$	$0F9A_H$	$0FAA_H$	$0FBA_H$	$0FCA_H$	T10
	$0F8B_H$	$0F9B_H$	$0FAB_H$	$0FBB_H$	$0FCB_H$	T11
	$0F8C_H$	$0F9C_H$	$0FAC_H$	$0FBC_H$	$0FCC_H$	T12
	$0F8D_H$	$0F9D_H$	$0FAD_H$	$0FBD_H$	$0FCD_H$	T13
	$0F8E_H$	$0F9E_H$	$0FAE_H$	$0FBE_H$	$0FCE_H$	T14
	$0F8F_H$	$0F9F_H$	$0FAF_H$	$0FBF_H$	$0FCF_H$	T15
output pin	V0 to V7	V8 to V15	V16 to V23	V24 to V31	V32 to V36	

Figure 2-82. VFT Display Data Buffer Memory (DBR)

2.12.4 Display Operation

As the above-mentioned, the segment and the digit are not allocated. After setting of the display timing for the number of digits according to the using VFT and storing the segment and digit data according to the respective timings, clearing BLK in VFTCR1 to 0 starts VFT display.

Figure 2-83. shows the VFT drive pulse and Figure 2-84, 85 show the display operation.

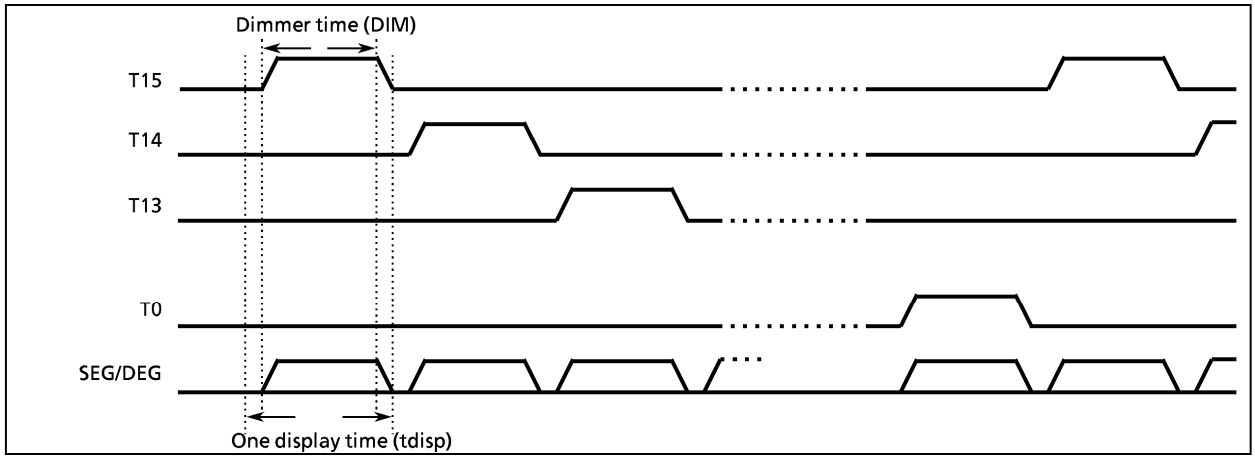


Figure 2-83. VFT Drive Waveform and Display timing

2.12.5 Example of Display operation

(1) For Conventional type VFT

When using the conventional type VFT, the output timing of the digits is specified to output 1 digit for 1 timing. Data must be set to output the pins which are specified to the digit in sequence. The following figure shows a data allocation of the display data buffer (DBR) and the output timing when VFT of 10 digits is used and V0 to V9 pins are allocated as the digit outputs. (When data is first written by the data buffer which corresponds to the digit pin, it is unnecessary to rewrite the data later.)

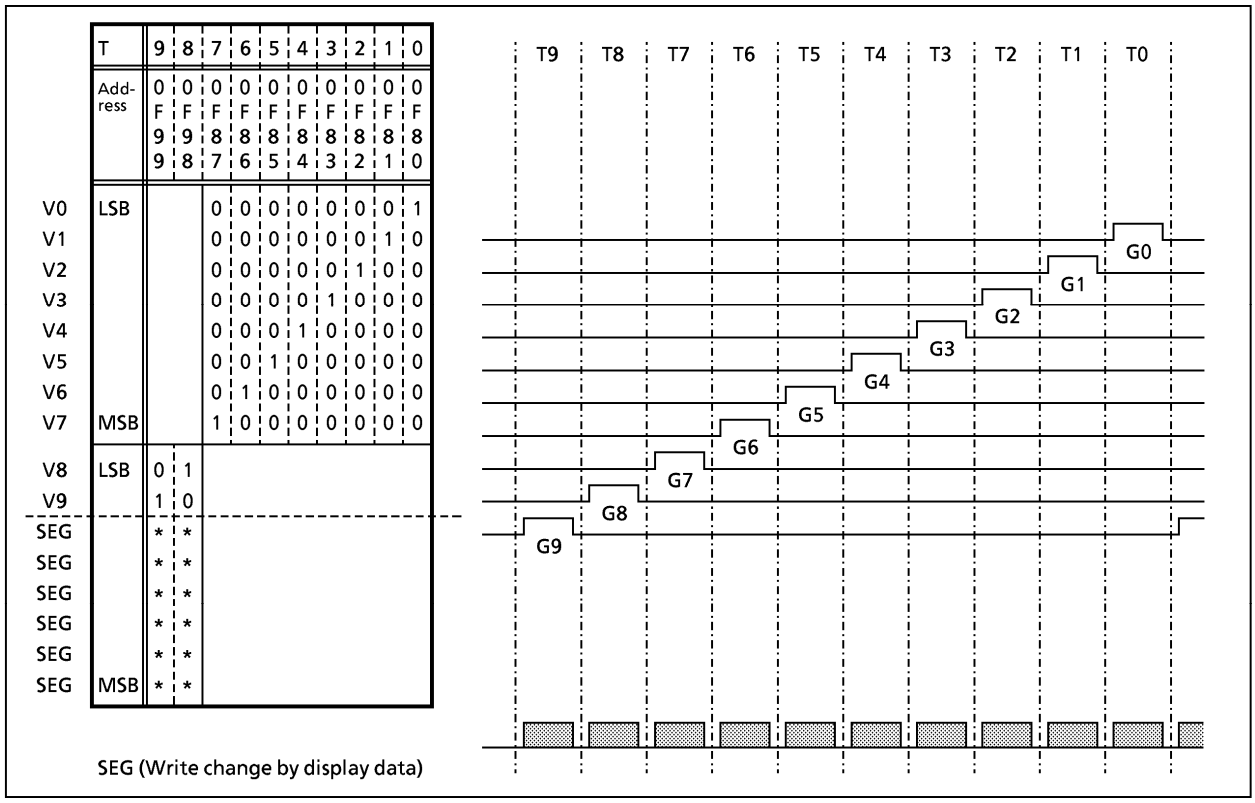


Figure 2-84. Example of Conventional type VFT driver pulse

(2) For Grid scan type VFT

When using the grid scan type VFT, two or more grids must be simultaneously selected to turn the display pattern which contains two or more grids on. Additionally, the timing and the data must be determined to set the grid scan mode as follows.

- When the display pattern which is fully set in the respective grids is turned on, only the grids which correspond as ever must be scanned in sequence to turn on the display pattern. (timing of T8 to T3 in the following figure)
- When the display pattern which contains two or more grids is turned on, two or more corresponding grids are simultaneously selected to turn on the display pattern. (timing of T2 to T0 in the following figure)

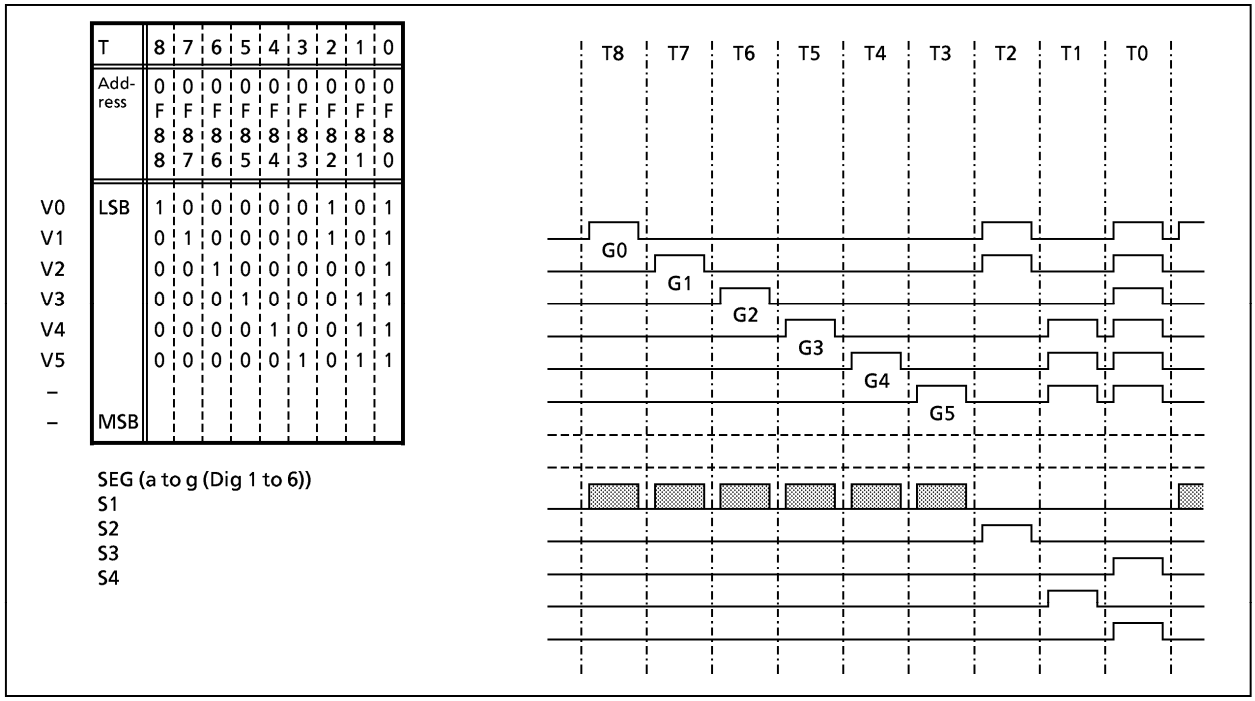


Figure 2-85. Grid Scan Type Display Vacuum Fluorescent Tube Ware

2.12.6 Port Function

(1) High-breakdown voltage buffer

To drive fluorescent display tube, clears the port output latch to "0". The port output latch is initialized to 0 at reset.

It is recommended that ports P6, P7, P8 and P9 should be used as VFT driver output. Precaution for using as general-purpose I/O pins are follows.

Note : When not using a pin which is pulled down to pin V_{KK} (RK = typ. 80 kΩ), it must be set to open. It is necessary to clear the port output latch and the data buffer memory (DBR) to "0".

① Ports P6 to P9

When a part of P6 to P9 is used as the input/output pin (VFT driver in operation), the data buffer memory (DBR) of the segment which is also used as the input/output pin must be cleared to "0".

② Port PD

VFT output and usual input/output are controlled by VSEL of VFT control register in bits. When a pin which is pulled down to pin V_{KK} is used as usual output or input, the following cautions are required.

(a) When outputting

When level "L" is output, a port which is pulled down to pin V_{KK} is pin V_{KK} voltage. Such processes as clamping with the diode as shown in figure 2-86. (a) are necessary to prevent pin V_{KK} voltage applying to the external circuit.

(b) When inputting

When the external data is input, the port output latch is cleared to "0".

The input threshold is the same as that of the other usual input/output port. However it is necessary to drive RK (typ. 80 kΩ) sufficiently because of pulled down to pin VKK.

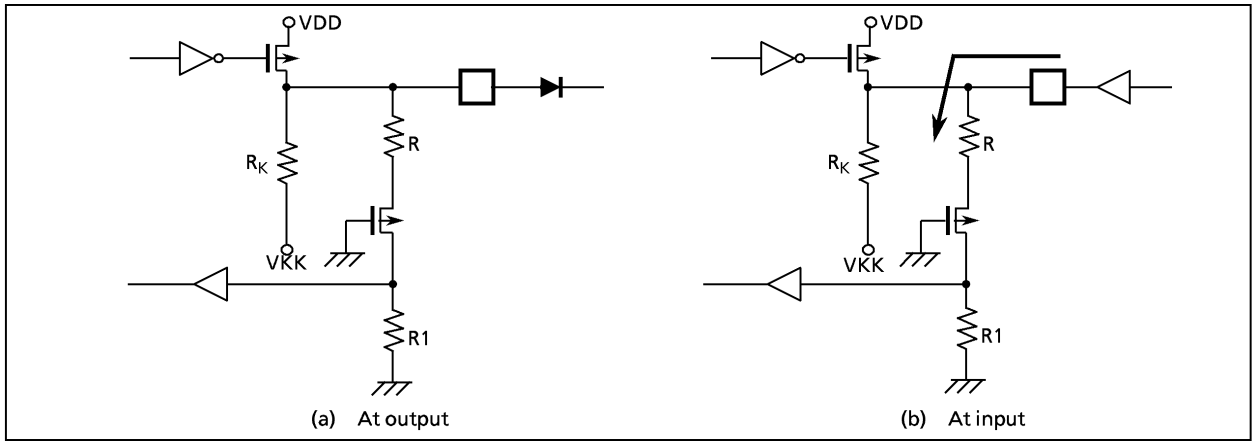


Figure 2-86. External Circuit Interface

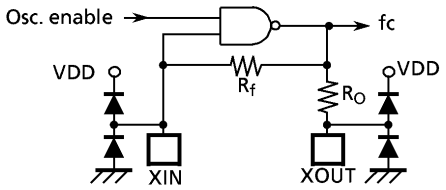
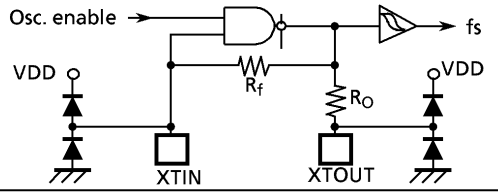
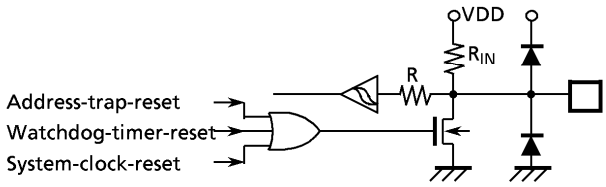
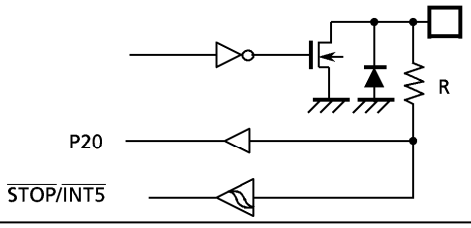
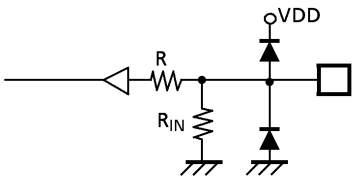
INPUT/OUTPUT CIRCUITRY

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 87CH74A/M74A control pins are shown below.

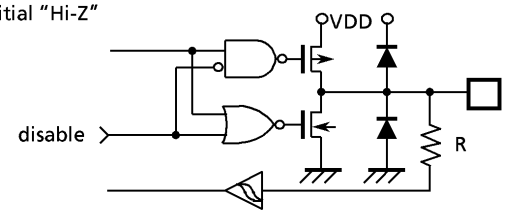
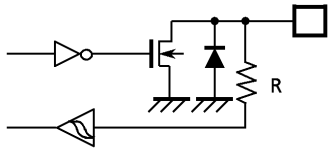
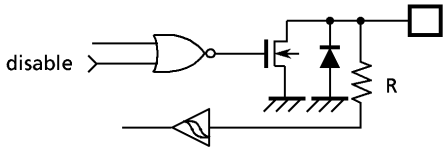
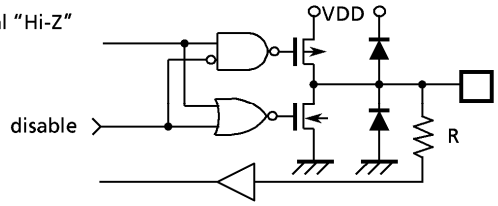
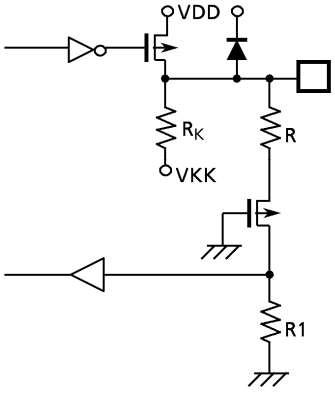
Please specify either the single-clock mode (oscillation only XIN/XOUT) or the dual-clock mode (oscillation both XIN/XOUT and XTIN/XTOUT) by a code (NM1 or NM2) as an option for an operating mode during reset.

CONTROL PIN	I/O	INPUT/OUTPUT CIRCUITRY and code	REMARKS
XIN XOUT	Input Output		Resonator connecting pins (high-frequency) R_f = 1.2 MΩ (typ.) R_O = 1.5 kΩ (typ.)
XTIN XTOUT	Input Output	NM1: Refer to port P2 NM2: 	Resonator connecting pins (low-frequency) R_f = 6 MΩ (typ.) R_O = 220 kΩ (typ.)
$\overline{\text{RESET}}$	I/O		Sink open drain output Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 1 kΩ (typ.)
$\overline{\text{STOP}}$ / $\overline{\text{INT5}}$	Input		Hysteresis input R = 1 kΩ (typ.)
TEST	Input		Pull-down resistor R_{IN} = 70 kΩ (typ.) R = 1 kΩ (typ.)

Note1: The TEST pin of the 87PM74 does not have a pull-down resistor. Be sure to fix the TEST pin to low in MCU mode.
Note2: The 87PM74 is placed in the single-clock mode during reset, and the input/output circuitries are the code NMI type.

(2) - ① Input/Output Ports

The input/output circuitries of the 87CH74A/M74A input / output ports are shown below, any one of the circuitries can be chosen by a code (A or D) as a mask option.

PORT	I/O	INPUT / OUTPUT CIRCUITRY	REMARKS
P0 P1	I/O	initial "Hi-Z" 	Tri-state I/O Hysteresis input High current output (P0, P10 to P14) R = 1 kΩ (typ.)
P2	I/O	initial "Hi-Z" 	Sink open drain output High current output R = 1 kΩ
P3	I/O	initial "Hi-Z" 	Sink open drain output Hysteresis input R = 1 kΩ (typ.)
P4 P5	I/O	initial "Hi-Z" 	Tri-state I/O R = 1 kΩ (typ.)
P6 P7 P8 P9	I/O	initial "Hi-Z" 	Source open drain output High-breakdown voltage RK = 80 kΩ (typ.) R = 1 kΩ (typ.) R1 = 200 kΩ (typ.)

Note: The input/output circuitries of the 87PM74 I/O ports are the code A type.

(2) - ② Input/Output Ports

PORT	I/O	INPUT/OUTPUT CIRCUITRY and CODE	REMARKS
PD	I/O	A: initial "Hi-Z", VDD, R, R1 D: initial "Hi-Z", VDD, R_K, VKK, R, R1	Source open drain output High-breakdown voltage R_K = 80 kΩ (typ.) R = 1 kΩ (typ.) R1 = 200 kΩ (typ.)

Electrical Characteristics

Absolute Maximum Ratings (V_{SS} = 0 V)

Parameter	Symbol	Pins	Ratings	Unit
Supply Voltage	V_{DD}		– 0.3 to 6.5	V
Input Voltage	V_{IN}		– 0.3 to V_{DD} + 0.3	V
Output Voltage	V_{OUT1}	P2, P3, P4, P5, XOUT, $\overline{RESET}$	– 0.3 to V_{DD} + 0.3	V
	V_{OUT3}	Source open drain ports	V_{DD} – 40 to V_{DD} + 0.3	
Output Current (Per 1 pin)	I_{OUT1}	P15 to P17, P3, P4, P5	3.2	mA
	I_{OUT2}	P0, P10 to P14, P2	30	
	I_{OUT3}	P8, P9, PD	– 12	
	I_{OUT4}	P6, P7	– 25	
Output Current (Total)	ΣI_{OUT1}	P15 to P17, P3, P4, P5	60	mA
	ΣI_{OUT2}	P0, P10 to P14, P2	160	
	ΣI_{OUT3}	P6, P7, P8, P9, PD	– 200	
Power Dissipation [Topr = 25°C]	PD	Note 2	1200	mW
Soldering Temperature (time)	Tsld		260 (10 s)	°C
Storage Temperature	Tstg		– 55 to 125	°C
Operating Temperature	Topr		– 30 to 70	°C

Note 1: The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

Note 2: Power Dissipation (PD) ; For PD, it is necessary to decrease 14.3 mw/°C.

Recommended Operating Conditions (V_{SS} = 0 V, Topr = – 30 to 70°C)

Parameter	Symbol	Pins	Conditions		Min	Max	Unit
Supply Voltage	V_{DD}		fc = 8 MHz	NORMAL 1, 2 modes	4.5	5.5	V
				IDLE1, 2 modes			
			fs = 32.768 kHz	SLOW mode	2.7		
				SLEEP mode			
				STOP mode	2.0		
Output Voltage	V_{OUT3}	Source open drain ports			V_{DD} – 38	V_{DD}	V
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 V to 5.5 V		0.4	8.0	MHz
			V_{DD} = 2.7 V to 5.5 V			4.2	
	fs	XTIN, XTOUT			30.0	34.0	kHz

Note 1: The recommended operating conditions for a device are operating conditions under which it can be guaranteed that the device will operate as specified. If the device is used under operating conditions other than the recommended operating conditions (supply voltage, operating temperature range, specified AC/DC values etc.), malfunction may occur. Thus, when designing products which include this device, ensure that the recommended operating conditions for the device are always adhered to.

Note 2: Clock frequency fc: Supply voltage range is specified in NORMAL 1/2 mode and IDLE 1/2 mode.

How to calculate power consumption.

With the TMP87CH74A/M74A, a pull-down resistor (Rk = 80 kΩ typ.) can be built into a VFT driver using mask option. The share of VFT driver loss (VFT driver output loss + pull-down resistor (Rk) loss) in power consumption Pmax is high. When using a fluorescent display tube with a large number of segments, the maximum power consumption Pd must not be exceeded.

power consumption Pmax = operating power consumption + normal output port loss + VFT driver loss

Where,
operating power consumption: VDD x IDD
LED output loss : I_{OL3} x V_{OL}
VFT driver loss : VFT driver output loss + pull-down resistor (Rk) loss

Example:
When Ta = 10 to 50°C (When using a fluorescent display tube with a grid scan type which can use two or more grid outputs.) and a fluorescent display tube with segment output = 3 mA, digit output = 15 mA, Vxx = -25 V is used.
Operating conditions: VDD = 5 V ± 10 %, fc = 8 MHz, VFT dimmer time (DIM) = (14/16) x tseg,
Digit outputs = two pin

Power consumption Pmax = (1) + (2) + (3)
Where,
(1) Operating power consumption: $V_{DD} \times I_{DD}$ = 5.5 V × 14 mA = 77 mW
(2) LED output : 10 mA × 1.0 V × 4 = 40 mW (when using four LED)
(3) VFT driver loss : segment pin = 3 mA x 2 V x number of segments X = 6 mW x X
digit pin = 15 mA x 2 V x 14/16 (DIM) x number of digits Y = 52.5 mW
Rk loss = (5.5 + 25 V)^2 / 50 kΩ x (number of segments X + number of digits Y) = 18.605 mW x (X + 2)

Therefore, Pmax = 77 mW + 40 mW + 6 mW × X + 52.5 mW + 18.605 mWx (X + 2) = 206.71 mW + 24.605X...

Maximum power consumption Pd when Ta = 50°C is determined by the following equation:
PD = 1200 mW-(14.3 x 25) = 842.5 mW
The number of segments X which can be lit is:
PD > Pmax
842.5 mW > 206.71 + 24.605 X
25.8 > X

Thus, a fluorescent display tube with less than 25 segments can be used. If a fluorescent display tube with 25 segments or more is used, either a pull-down resistor must be attached externally , or the number of segments to be lit must be kept to less than 25 by software.

D.C. Characteristics

(V_{SS} = 0 V, Topr = – 30 to 70°C)

Parameter	Symbol	Pins	Conditions	Min	Typ.	Max	Unit
Hysteresis Voltage	V_{HS}	Hysteresis input		–	0.9	–	V
Input Current	I_{IN1}	TEST	V_{DD} = 5.5 V V_{IN} = 5.5 V/0 V	–	–	± 2	μA
	I_{IN2}	Open drain ports, Tri-state ports					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
	I_{IN4}	PD ports (Note3)		–	–	80	
Input Resistance	R_{IN2}	$\overline{RESET}$		100	220	450	kΩ
Pull-down Resistance	R_K	Source open drain ports	V_{DD} = 5.5 V, V_{KK} = – 30 V	50	80	110	kΩ
Output Leakage Current	I_{LO1}	Sink open drain ports	V_{DD} = 5.5 V, V_{OUT} = 5.5 V	–	–	2	μA
	I_{LO2}	Source open drain ports and tri-state ports	V_{DD} = 5.5 V, V_{OUT} = – 32 V	–	–	– 2	
	I_{LO3}	Tri-state ports	V_{DD} = 5.5 V, V_{OUT} = 5.5 V / 0 V	–	–	± 2	
Output High Voltage	V_{OH2}	Tri-state ports	V_{DD} = 4.5 V, I_{OH} = – 0.7 mA	4.1	–	–	V
	V_{OH3}	P8, P9, PD	V_{DD} = 4.5 V, I_{OH} = – 8 mA	2.4	–	–	
Output Low Voltage	V_{OL}	Except XOUT, P0, P10 to P14, P2	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.4	V
Output Low Current	I_{OL3}	P0, P10 to P14, P2	V_{DD} = 4.5 V, V_{OL} = 1.0 V	–	20	–	mA
Output High Current	I_{OH}	P6, P7	V_{DD} = 4.5 V, V_{OH} = 2.4 V	–	– 20	–	mA
Supply Current in NORMAL 1, 2 modes	I_{DD}		V_{DD} = 5.5 V fc = 8 MHz fs = 32.768 kHz V_{IN} = 5.3 V/0.2 V	–	10	14	mA
Supply Current in IDLE 1, 2 modes				–	6	9	
Supply Current in SLOW mode			V_{DD} = 3.0 V fs = 32.768 kHz V_{IN} = 2.8 V / 0.2 V	–	30	60	μA
Supply Current in SLEEP mode				–	15	30	
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 , V_{DD} = 5 V.

Note 2: Input Current I_{IN1},I_{IN3} ; The current through resistor is not included, when the input resistor (pull-up/pull-down) is contained.

Note 3: Input Current I_{IN4} ; The current when the pull-down register (Rk) is not connected by the mask option.

A/D Conversion Characteristics

(V_{SS} = 0 V, V_{DD} = 4.5 to 6.0 V, Topr = – 30 to 70°C)

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Analog Reference Voltage	V_{AREF}	$V_{AREF} - V_{ASS} \geqq 2.5$ V	V_{DD} – 1.5	–	V_{DD}	V
	V_{ASS}		V_{SS}			
Analog Input Voltage	V_{AIN}		V_{ASS}	–	V_{AREF}	V
Analog Supply Current	I_{REF}	V_{AREF} = 5.5 V, V_{ASS} = 0.0 V		0.5	1.0	mA
Nonlinearity Error		V_{DD} = 5.0 V, V_{SS} = 0.0 V V_{AREF} = 5.000 V V_{ASS} = 0.000 V	–	–	± 1	LSB
Zero Point Error			–	–	± 1	
Full Scale Error			–	–	± 1	
Total Error			–	–	± 2	

Note: Total errors includes all errors, except quantization error.

A.C. Characteristics

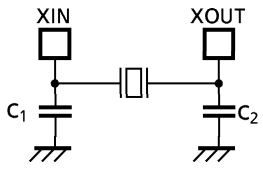
(V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = – 30 to 70°C)

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Machine Cycle Time	t_{cy}	In NORMAL1, 2 modes	0.5	–	10	μs
		In IDLE 1, 2 modes				
		In SLOW mode	117.6	–	133.3	
		In SLEEP mode				
High Level Clock Pulse Width	t_{WCH}	For external clock operation (XIN input), fc = 8 MHz	50	–	–	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	14.7	–	–	μs
Low Level Clock Pulse Width	t_{WSL}					

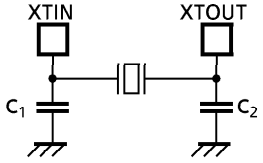
Recommended Oscillating Conditions

(V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = – 30 to 70°C)

Parameter	Oscillator	Oscillation Frequency	Recommended Oscillator		Recommended Constant C_1	Recommended Constant C_2
High-frequency Oscillation	Ceramic Resonator	8 MHz	KYOCERA	KBR8.0M	30pF	30pF
		4 MHz	KYOCERA	KBR4.0MS		
			MURATA	CSA 4.00MG		
	Crystal Oscillator	8 MHz	TOYOCOM	210B 8.0000	20pF	20pF
		4 MHz	TOYOCOM	204B 4.0000		
Low-frequency Oscillation	Crystal Oscillator	32.768 KHz	NDK	MX-38T	15pF	15pF



(1) High-frequency Oscillation



(2) Low-frequency Oscillation

Note: An electrical shield by metal shield plate on the surface of IC package should be recommendable in order to prevent the device from the high electric fieldstress applied from CRT (Cathode Ray Tube) for continuous reliable operation.