

CMOS 8-BIT MICROCONTROLLER

TMP87CC70F, TMP87CH70F, TMP87CK70AF, TMP87CM70AF

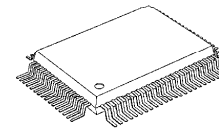
The 87CC70/H70/K70A/M70A are the high speed and high performance 8-bit single chip microcomputers. These MCU contain 6-bit A/D conversion inputs and a VFT (Vacuum Fluorescent Tube) driver on a chip.

| PART No. | ROM | RAM | PACKAGE | OTP MCU |
|-------------|-------------|-------------|--------------------|------------|
| TMP87CC70F | 12K × 8-bit | 512 × 8-bit | QFP80-P-1420-0.80B | TMP87PM70F |
| TMP87CH70F | 16K × 8-bit | | | |
| TMP87CK70AF | 24K × 8-bit | | | |
| TMP87CM70AF | 32K × 8-bit | | | |

FEATURES

- ◆ 8-bit single chip microcomputer TLCS-870 Series
- ◆ Instruction execution time : 0.5 μ s (at 8 MHz), 122 μ s (at 32.768 kHz)
- ◆ 412 basic instructions
 - Multiplication and Division (8bits × 8bits, 16bits ÷ 8bits)
 - Bit manipulations (Set/Clear/Complement/Move/Test/Exclusive or)
 - 16-bit data operations
 - 1-byte jump/subroutine-call (Short relative jump/ Vector call)
- ◆ 14 interrupt sources (External : 5, 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
- ◆ 10 Input/Output ports (73 pins)
 - Output : 1 port (8 pins)
 - Input/Output : 9 ports (65 pins)
- ◆ Two 16-bit Timer/Counters
 - Timer, Event counter 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 16 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
- ◆ 8-bit High Speed Serial Output (rate : max. 1 bit/ μ s)
- ◆ 6-bit A/D conversion input (6 channels)
- ◆ Vacuum Fluorescent Tube Driver (automatic display)
 - High breakdown voltage ports (max. 40 V × 32 bits)
- ◆ Key scanning function
 - Key-matrix constructed by segment outputs (1 to 16) and key inputs (1 to 8)
- ◆ 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 6 V at 4.19 MHz / 32.768 kHz,
4.5 to 6 V at 8 MHz / 32.768 kHz (87CK70A/M70A)
4.5 to 6 V at 8 MHz / 32.768 kHz (87CC70/H70, 87PM70)
- ◆ Emulation Pod : BM87CK70F0B (or BM87CK70F0A)

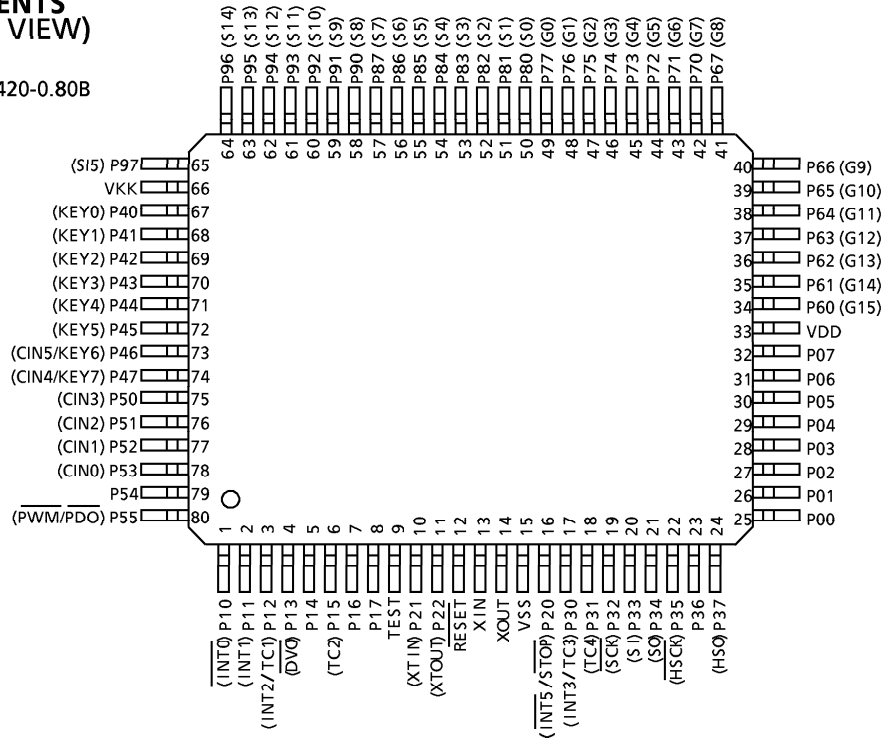
QFP80-P-1420-0.80B



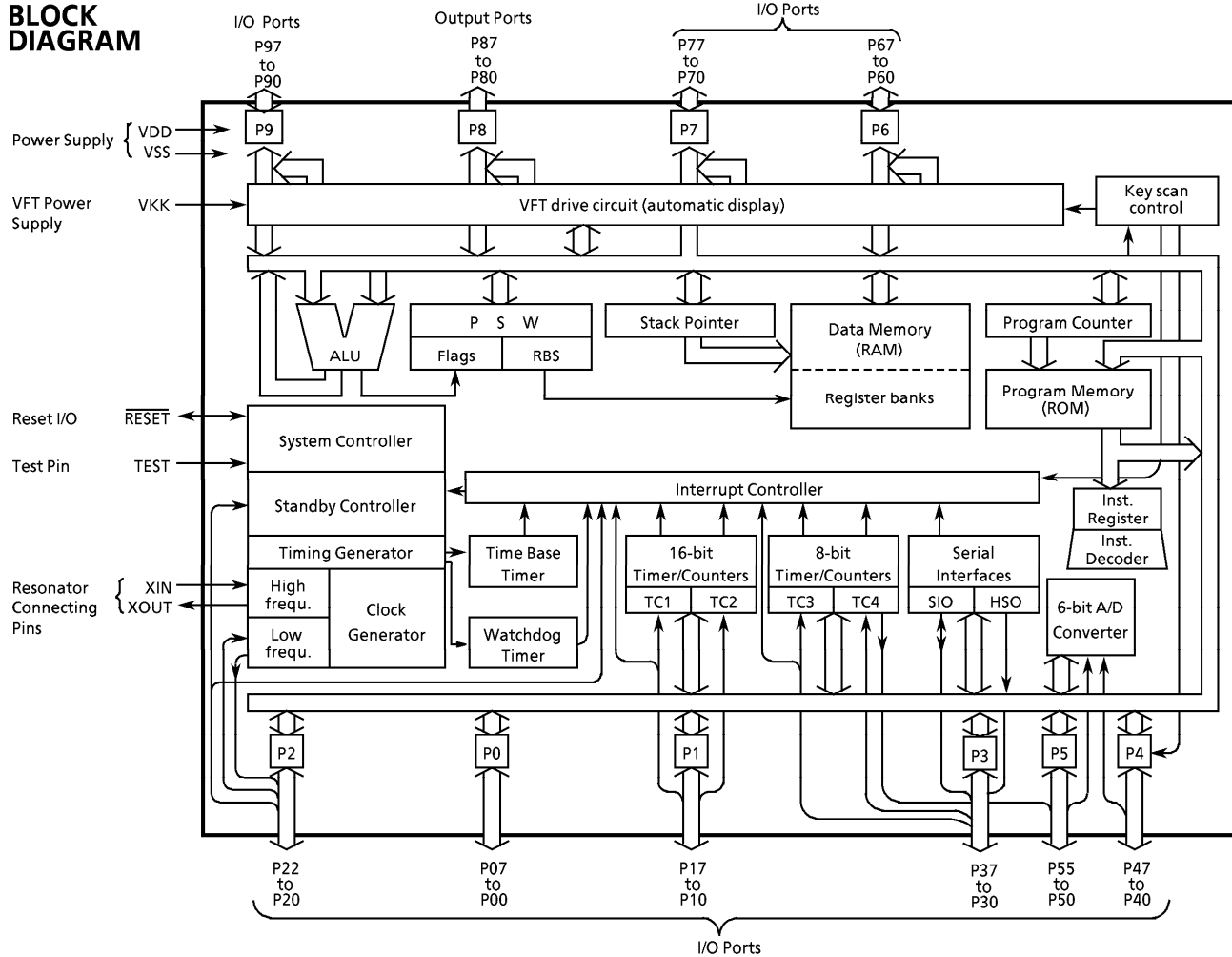
TMP87CC70F
 TMP87CH70F
 TMP87CK70AF
 TMP87CM70AF
 TMP87PM70F

**PIN ASSIGNMENTS
(TOP VIEW)**

QFP80-P-1420-0.80B



**BLOCK
DIAGRAM**



PIN FUNCTION

| PIN NAME | Input / Output | FUNCTION | | | |
|--|----------------|---|---|--|---|
| P07 to P00 | I/O | | | | |
| P17, P16, P14 | I/O | Two 8-bit programmable input/output ports (tri-state). | | | |
| P15 (TC2) | I/O (Input) | Each bit of these ports can be individually configured as an input or an output under software control. During reset, all bits are configured as inputs. When used as a divider output, the latch must be set to "1". | Timer/Counter 2 input | | |
| P13 (DVO) | I/O (Output) | | Divider output | | |
| P12 (INT2 / TC1) | | | External interrupt input 2 or Timer/Counter 1 input | | |
| P11 (INT1) | I/O (Input) | | External interrupt input 1 | | |
| P10 (INT0) | | | External interrupt input 0 | | |
| P22 (XTOUT) | I/O (Output) | 3-bit input/output port with latch. When used as an input port, 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) | | External interrupt input 5 or STOP mode release signal input | | |
| P20 (INT5 / STOP) | | | | | |
| P37 (HSO) | I/O (Output) | 8-bit input/output port with latch. When used as an input port, a HSO output, a SIO input/output, a timer/counter input, or an interrupt input, the latch must be set to "1". | HSO serial data output | | |
| P36 | I/O | | HSO serial clock output | | |
| P35 (HSCK) | I/O (Output) | | | | |
| P34 (SIO) | | | | SIO serial data output | |
| P33 (SIO) | I/O (Input) | | | SIO serial data input | |
| P32 (SCK) | I/O (I/O) | | | SIO serial clock input/output | |
| P31 (TC4) | I/O (Input) | | | Timer/Counter 4 input | |
| P30 (INT3 / TC3) | | | | External interrupt input 3 or Timer/Counter 3 input | |
| P47 (CIN4 / KEY7), P46 (CIN5 / KEY6) P45 (KEY5) to P40 (KEY0) | I/O (Input) | | | 8-bit input/output port with latch. When used as an input port, the latch must be set to "1". | Comparator inputs or Key scan inputs Key scan inputs |
| P55 (PWM / PDO) | I/O (Output) | | | 6-bit input/output port with latch. When used as an input port, a comparator input, or a PWM/PDO output, the latch must be set to "1". | 8-bit PWM output or 8-bit programmable divider output |
| P54 | I/O | | | | |
| P53 (CIN0) to P50 (CIN3) | I/O (Input) | Comparator inputs | | | |
| P67 (G8) to P60 (G15) P77 (G0) to P70 (G7) P97 (S15) to P90 (S8) | I/O (Output) | Three 8-bit high breakdown voltage I/O ports with the latch. When used as a VFT driver output, the latch must be cleared to "0". | VFT digit driver outputs | | |
| P87 (S7) to P80 (S0) | | | Output (Output) | VFT segment driver outputs (Key strobe outputs) | |
| | | | | | |
| XIN, XOUT | Input, Output | Resonator connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened. | | | |
| RESET | I/O | Reset signal input or watchdog timer output/address-trap-reset output/system-clock-reset output. | | | |
| 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 | | | |

OPERATIONAL DESCRIPTION

1. CPU CORE FUNCTIONS

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

1.1 Memory Address Map

The TLCS-870 Series is capable of addressing 64K bytes of memory. Figure 1-1 shows the memory address maps of the 87CC70/H70/K70A/M70A. 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.

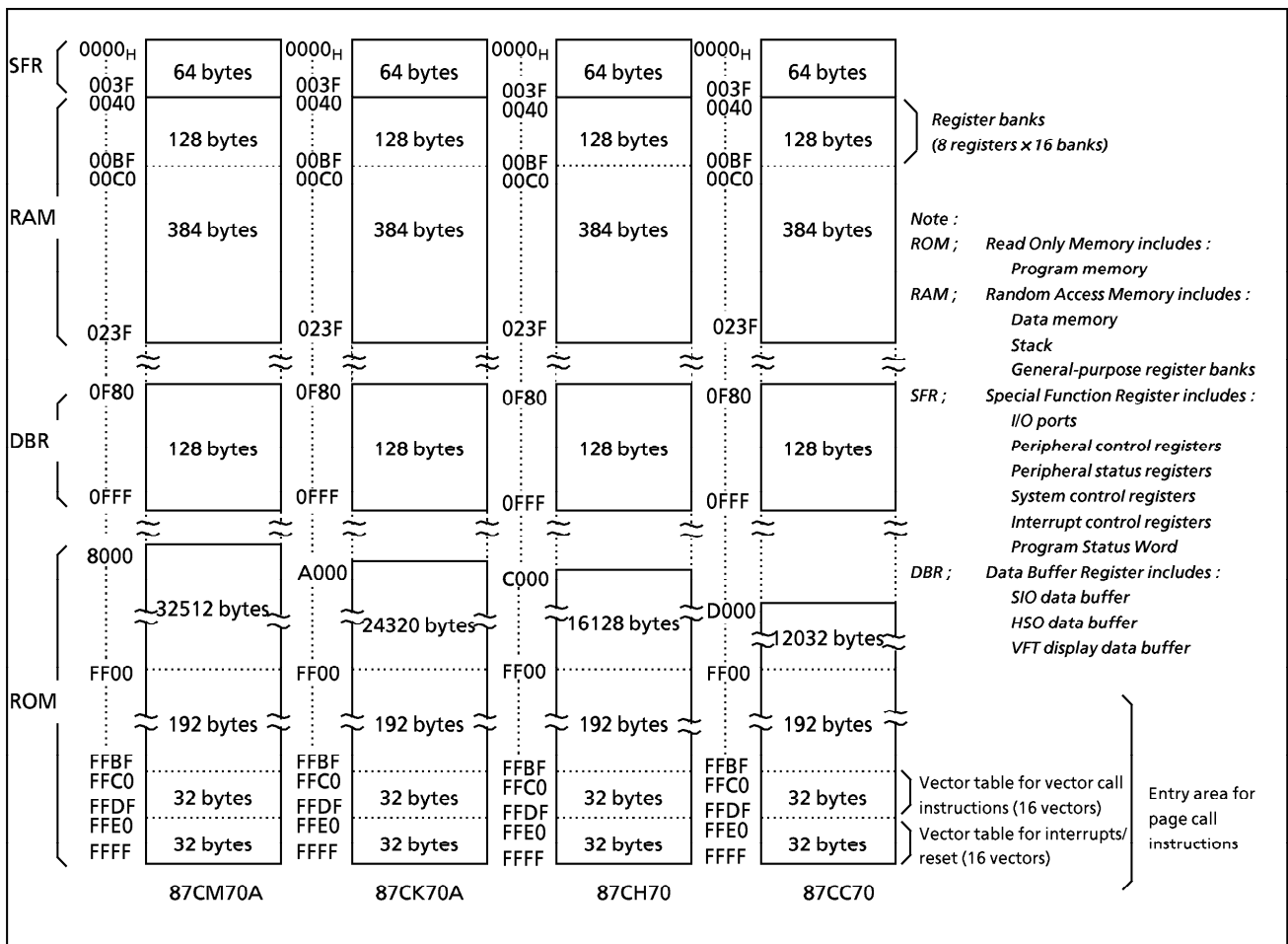


Figure 1-1. Memory Address Maps

1.2 Program Memory (ROM)

The 87CC70 has a 12K × 8-bit (addresses D000_H-FFFF_H), the 87CH70 has a 16K × 8-bit (addresses C000_H-FFFF_H), the 87CK70A has a 24K × 8-bit (addresses A000_H-FFFF_H), and the 87CM70A has a 32K × 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 15 interrupt vectors (2 bytes/vector). These vectors store a reset start address and interrupt service routine entry addresses.

(2) Vector table for **vector call** instructions (addresses FFC0_H-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.

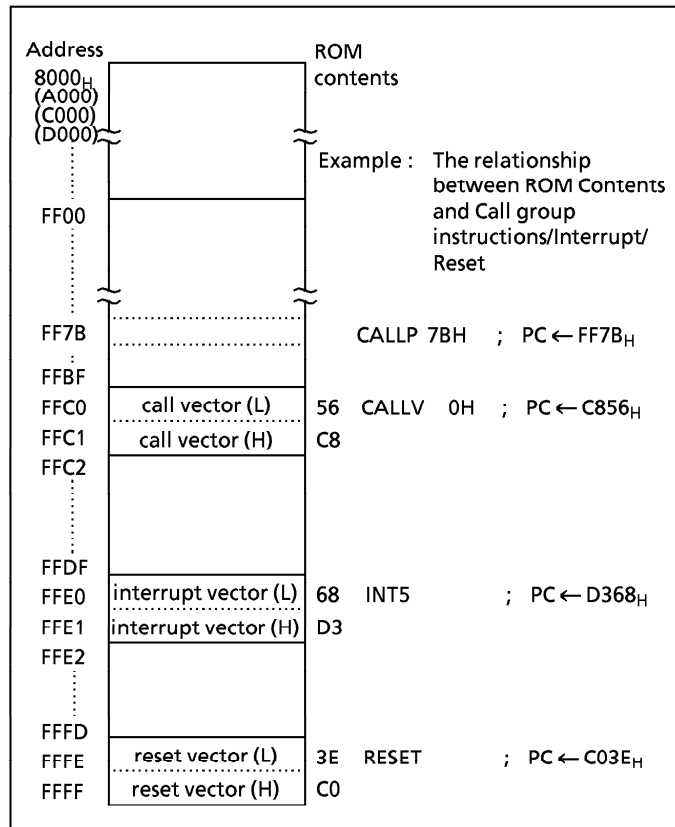


Figure 1-2. Program Memory Map

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 ($HL \geq 8000_H$):

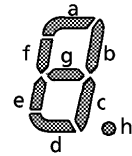
```
LD      A, (HL)          ; A ← ROM (HL)
```

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

```
ADD     A, TABLE - $ - 4    ; P3 ← ROM (TABLE + A)
LD      (P3), (PC + A)
JRS    T, SNEXT
```

```
TABLE : DB    0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0D8H, 80H, 98H
SNEXT :
```

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



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

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

```
SHLC   A                ; if A = 00_H then PC ← C234_H
JP      (PC + A)        ; if A = 01_H then PC ← C378_H
                          ; if A = 02_H then PC ← DA37_H
                          ; if A = 03_H then PC ← E1B0_H
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 $C125_H$.

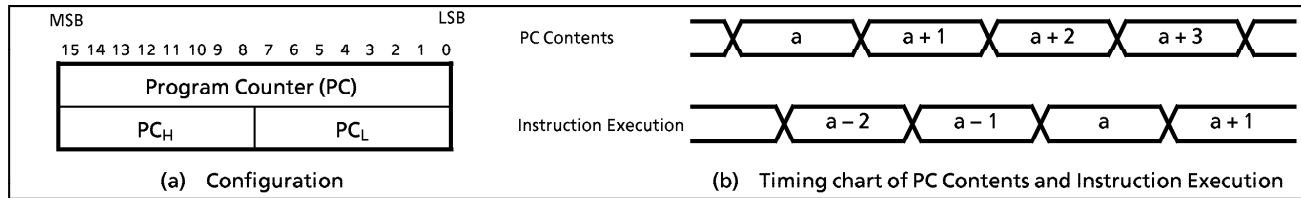


Figure 1-3. Program Counter

1.4 Data Memory (RAM)

The 87CC70/H70/K70A/M70A each have 512×8 -bits (addresses 0040_H - $023F_H$) 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 \times 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 87CC70/H70/K70A/M70A 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 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-purpose registers are mapped in the RAM ; therefore, *do not clear RAM at the current bank addresses.*

Example : Clears RAM to "00_H" except the bank 0:

```

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

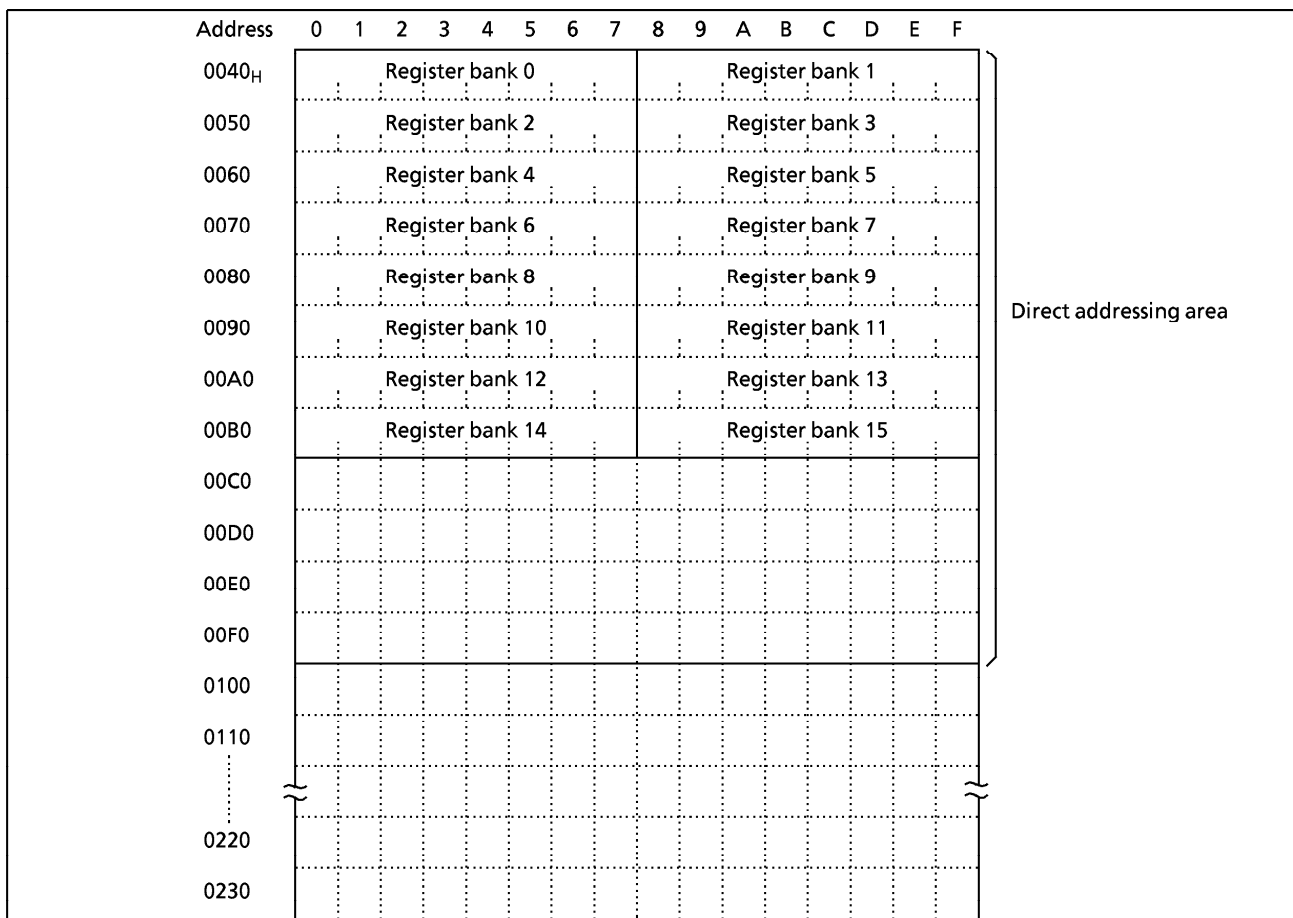


Figure 1-4. Data Memory Map

1.5 General-purpose Register Banks

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.

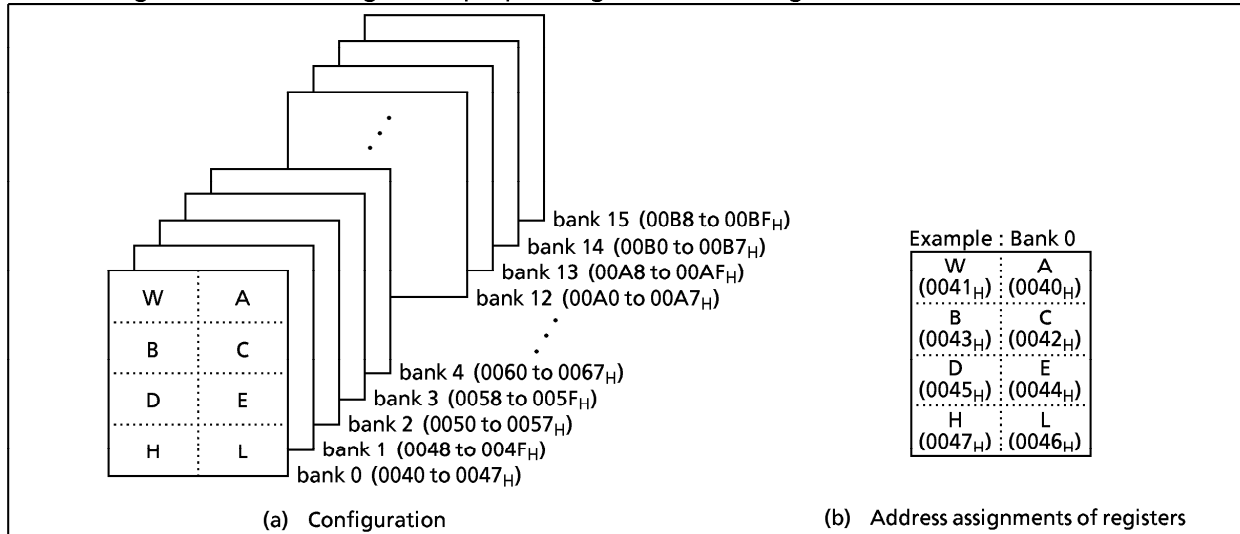


Figure 1-5. General-purpose Register Banks

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 1234_H 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 52_H 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 TLC8-870 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, n-1           ; Sets (number of bytes to transfer) - 1 to B
LD    HL, DSTA         ; Sets destination address to HL
LD    DE, SRCA         ; Sets source address to DE
SLOOP: LD  (HL), (DE)   ; (HL) ← (DE)
      INC  HL           ; HL ← HL + 1
      INC  DE           ; DE ← DE + 1
      DEC  B            ; B ← B - 1
      JRS  F, SLOOP     ; IF B ≥ 0 then loop

```

(3) B, C, BC

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

Example 1 : Repeat processing

```

LD    B, n           ; Sets n as the number of repetitions to B
SREPEAT: [ ..... ] ; (n + 1 times processing)
          [ 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 003FH 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 003FH 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".

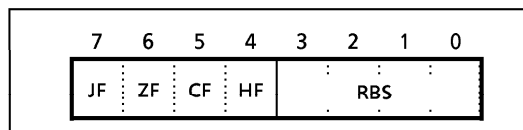


Figure 1-6. PSW (Flags, RBS) Configuration

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 00H (for 8-bit operations and data transfers)/0000H (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 00H during the multiplication instruction [MUL], and when 00H 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 00H (divided by zero error), or when the quotient is 100H or higher (quotient 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.

Example 1 : Bit manipulation

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

Example 2 : 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, CF = 0
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 |
| ROL A | 35 | 1 | 0 | 1 | 0 |
| ROR 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.

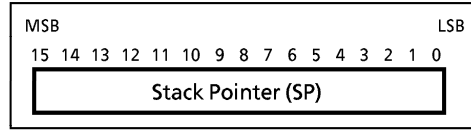


Figure 1-7. Stack Pointer

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←023FH
```

Example 2 : To read the SP

```
LD    HL, SP      ; HL←SP
```

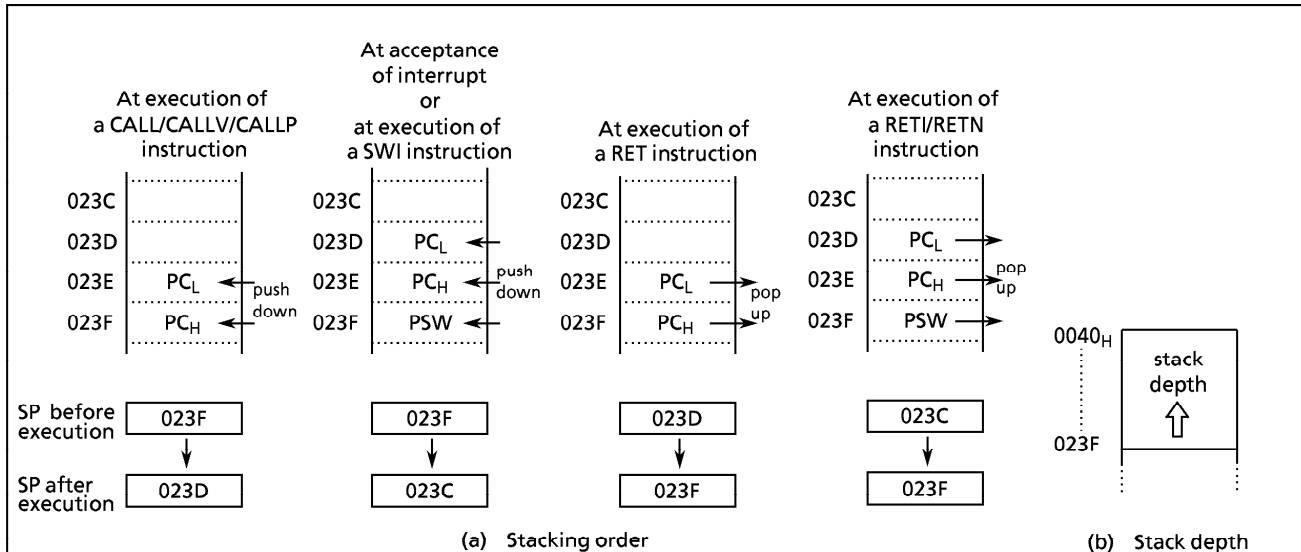


Figure 1-8. Stack

1.8 System Clock Controller

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

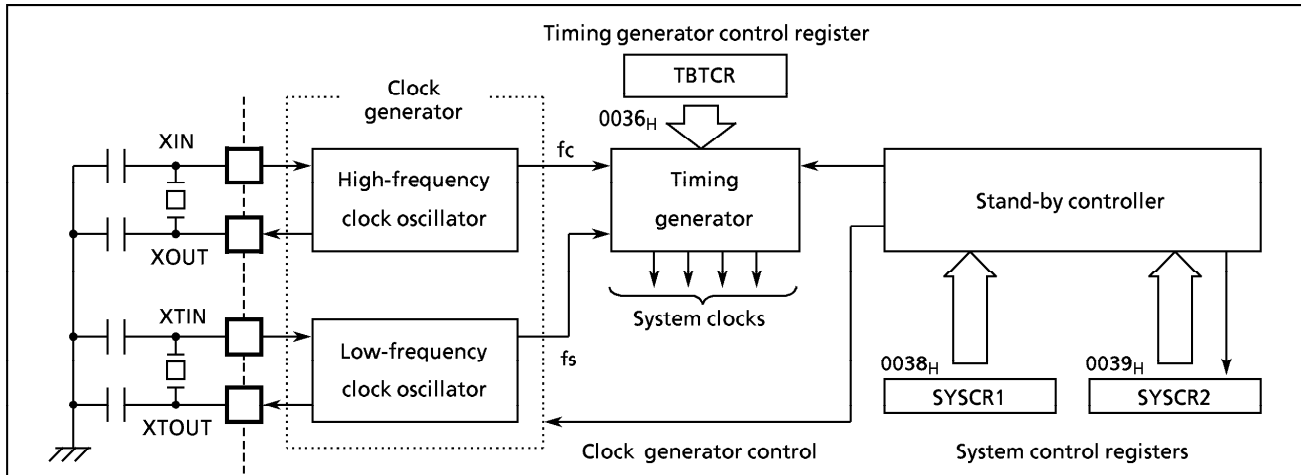


Figure 1-9. System Clock Controller

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 (f_c) and low-frequency (f_s) clocks can be easily obtained by connecting a resonator between the XIN/XOUT and XTIN/XTOUT pins, respectively. Clock input from an external oscillator is also possible. In this case, external clock is applied to the XIN/XTIN pin with the XOUT/XTOUT pin not connected. The 87CC70/H70/K70A/M70A are not provided an RC oscillation.

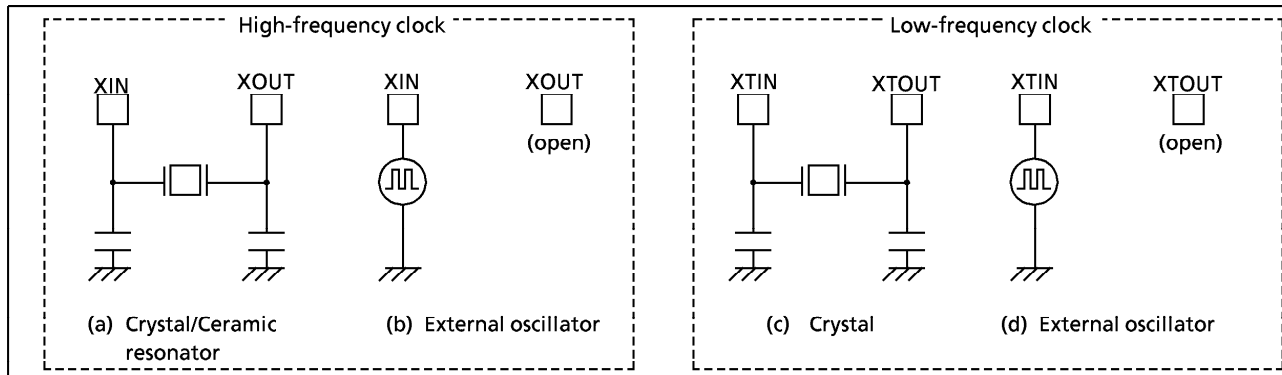
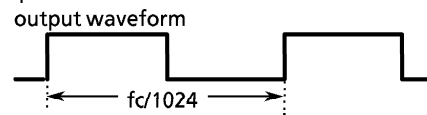


Figure 1-10. Examples of Resonator Connection

Note : *Accurate Adjustment of the Oscillation Frequency:*
 Although no hardware to externally and directly monitor the basic clock pulse is not provided, the oscillation frequency can be adjusted by 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 ($\overline{\text{DVO}}$) 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 ($\overline{\text{DVO}}$) 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.

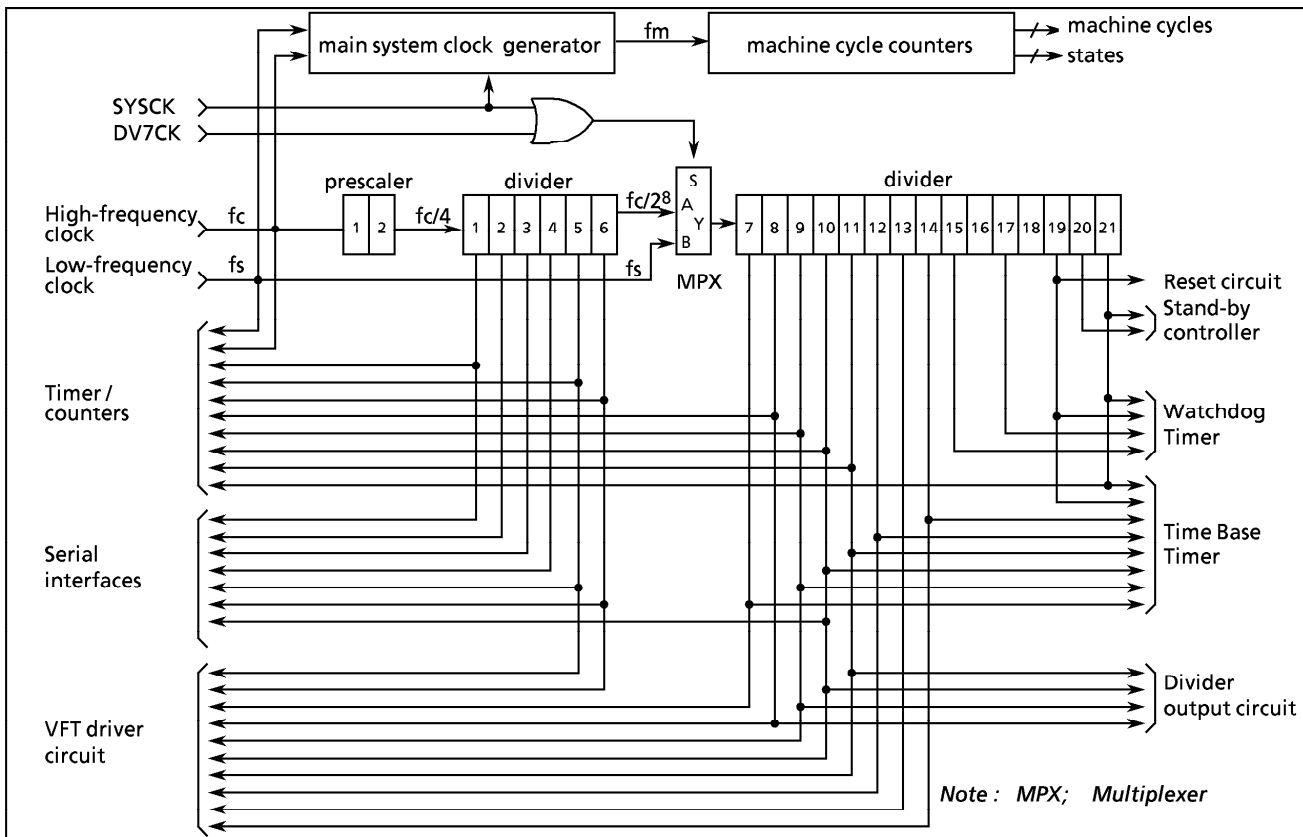


Figure 1-11. Configuration of Timing Generator

| | | | | | | | | | |
|------------------|---------|--|-------|---------|---------|---|-------------------------------|---|----------------------------|
| TBTCR (0036H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: 0**0 0***) |
| | (DVOEN) | (DVOCK) | DV7CK | (TBTEN) | (TBTCK) | | | | |
| | DV7CK | Selection of input clock to the 7th stage of the divider | | | | | 0 : $fc/2^8$ [Hz] 1 : fs | | write only |

Note 1 : 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 a "machine cycle". There are a total of 10 different types of instructions for the TLCS-870 Series: ranging from 1-cycle instructions which require one machine cycle for execution to 10-cycle instructions which require 10 machine cycles for execution.

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

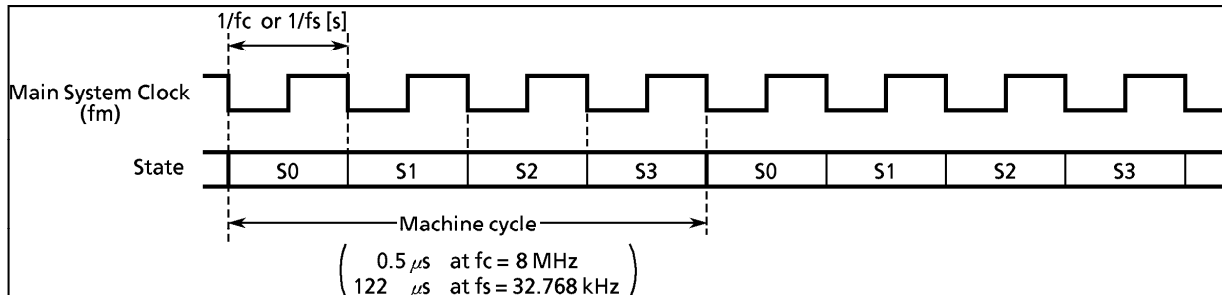


Figure 1-13. Machine Cycle

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 $4/f_c$ [s] ($0.5 \mu s$ at $f_c = 8 \text{ MHz}$).

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 87CC70/H70/K70A/M70A 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 $4/f_c$ [s] ($0.5 \mu\text{s}$ at $f_c = 8 \text{ MHz}$) in NORMAL2 and IDLE2 modes, and $4/f_s$ [s] ($122 \mu\text{s}$ at $f_s = 32.768 \text{ kHz}$) in SLOW and SLEEP modes. Note that *the 87PM70 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 87CC70/H70/K70A/M70A 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

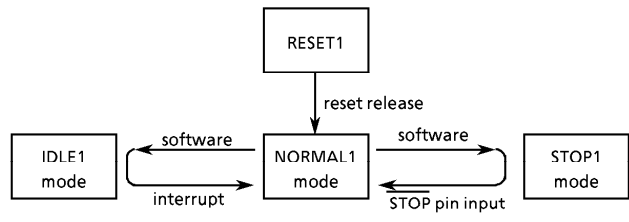
In this mode, the internal oscillation circuits remain active. The CPU and the watchdog timer are halted; however, on-chip peripherals operate using the high-frequency clock and/or the low-frequency clock. Starting and releasing of IDLE2 mode are the same as for IDLE1 mode, except that operation returns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

④ SLEEP mode

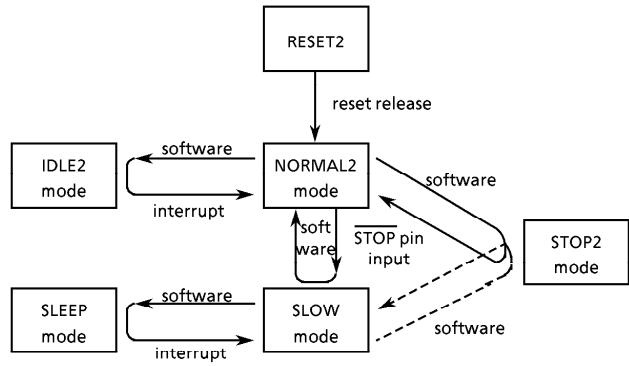
In this mode, the internal oscillation circuit of the low-frequency clock remains active. The CPU, the watchdog timer, and the internal oscillation circuit of the high-frequency clock are halted; however, on-chip peripherals operate using the low-frequency clock. Starting and releasing of SLEEP mode 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 : *NORMAL1 and NORMAL2 modes are generically called NORMAL; STOP1 and STOP2 are called STOP; and IDLE1, IDLE2 and SLEEP are called IDLE.*

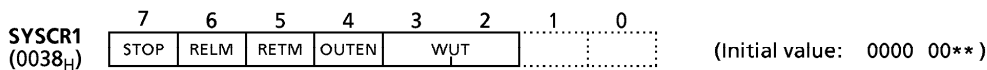
| Operating mode | | Frequency | | CPU core | On-chip Peripherals | Machine cycle time |
|----------------|---------|-------------------------|-------------------------|----------------|------------------------------------|--------------------|
| | | High-frequency | Low-frequency | | | |
| Single-Clock | RESET1 | turning on oscillation | turning off oscillation | reset | reset | 4/fc [s] |
| | NORMAL1 | | | operate | operate (Note 1) | |
| | IDLE1 | halt | | halt | | |
| | STOP1 | turning off oscillation | | — | | |
| 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 | Low-frequency | |
| | SLOW | turning off oscillation | | Low-frequency | Low-frequency (Note 2) | 4/fs [s] |
| | SLEEP | | | halt | halt | |
| | STOP2 | turning off oscillation | | halt | — | |

Note 1: *The High-Speed Serial Output (HSO) and the Comparator Inputs are halted in IDLE1/IDLE2 mode because CPU core is halted.*

Note 2: *The Vacuum Fluorescent Tube (VFT) driver circuit and the High Speed Serial Output (HSO) are halted. The Comparator inputs cannot be used in SLOW/SLEEP mode.*

Figure 1-14. Operating Mode Transition Diagram

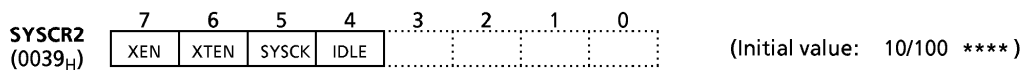
System Control Register 1



| | | | |
|-------|--|---|-----|
| 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 : $3 \times 2^{19} / f_c$ or $3 \times 2^{13} / f_s$ [s] 01 : $2^{19} / f_c$ or $2^{13} / f_s$ 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 RESET pin input, a return is made to NORMAL mode regardless of the RETM contents.
- Note 3 :** f_c ; high-frequency clock [Hz]
 f_s ; 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.

System Control Register 2



| | | | |
|-------|---|--|-----|
| 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 (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) |

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, 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
          JRS       F, SINT5           port P20 is at high
          LD        (SYSCR1), 01000000B ; Sets up the level-sensitive release mode.
          SET       (SYSCR1) . 7       ; Starts STOP mode
          LDW       (IL, 1111011101010111B ; IL11, 7, 5, 3←0 (clears interrupt latches)
SINT5 :   RETI
    
```

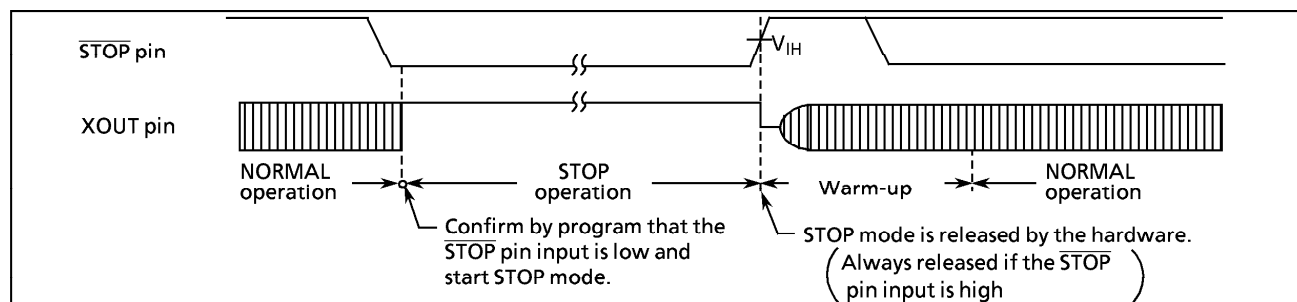


Figure 1-16. Level-sensitive Release Mode

Note 1 : After warm-up start, even if $\overline{\text{STOP}}$ pin input is low again, STOP mode does not restart.
 Note 2 : 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 $\overline{\text{STOP}}$ 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 $\overline{\text{STOP}}$ pin.

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

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

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

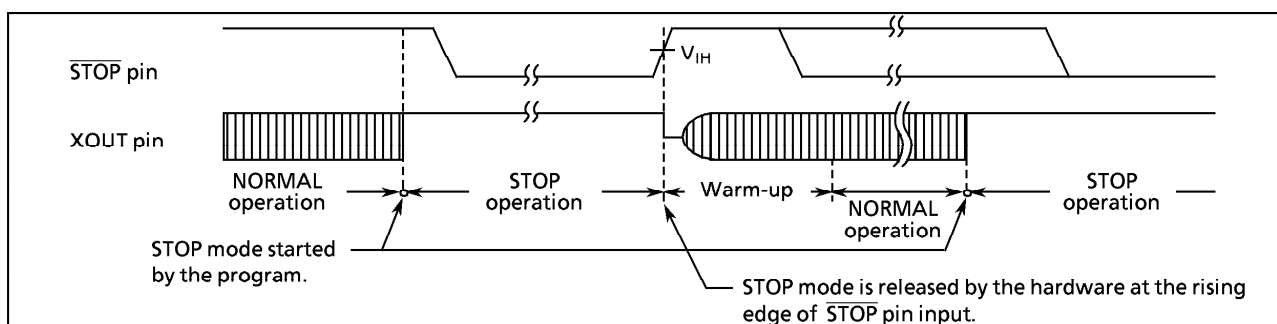


Figure 1-17. Edge-sensitive Release Mode

STOP mode is released by the following sequence:

- ① When returning to NORMAL2, both the high-frequency and low-frequency clock oscillators are turned on ; when returning to SLOW mode, only the low-frequency clock oscillator is turned on. When returning to 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 $f_c = 4.194304 \text{ MHz}$ | At $f_c = 8 \text{ MHz}$ | WUT | At $f_s = 32.768 \text{ kHz}$ |
| $3 \times 2^{19} / f_c$ [s] | 375 [ms] | 196.6 [ms] | $3 \times 2^{13} / f_s$ [s] | 750 [ms] |
| $2^{19} / f_c$ | 125 | 65.5 | $2^{13} / f_s$ | 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 $\overline{\text{RESET}}$ pin low, which immediately performs the normal reset operation.

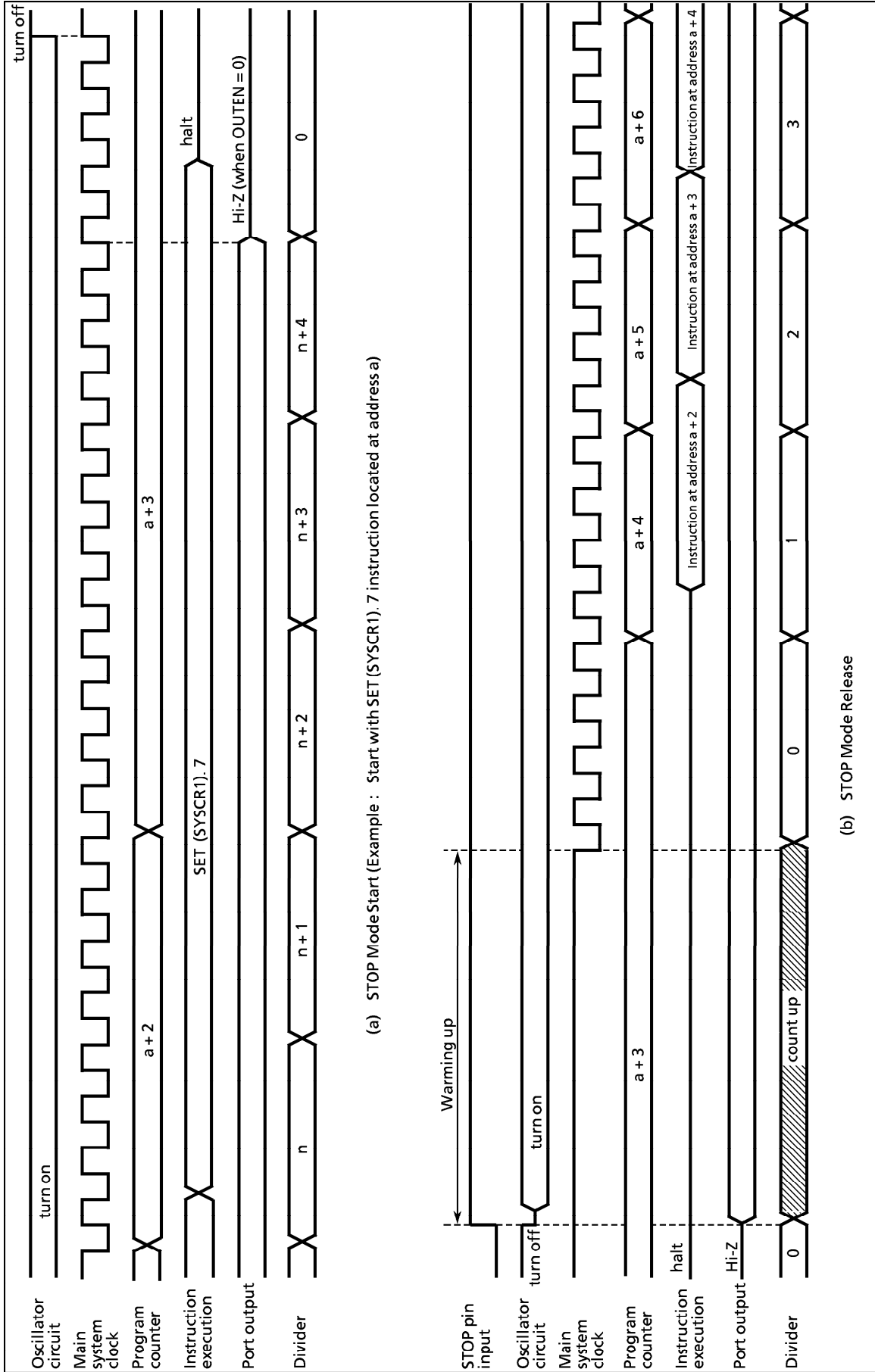


Figure 1-18. STOP Mode Start / Release

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.

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]).

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 87CC70/H70/K70A/M70A are placed in NORMAL mode.

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.

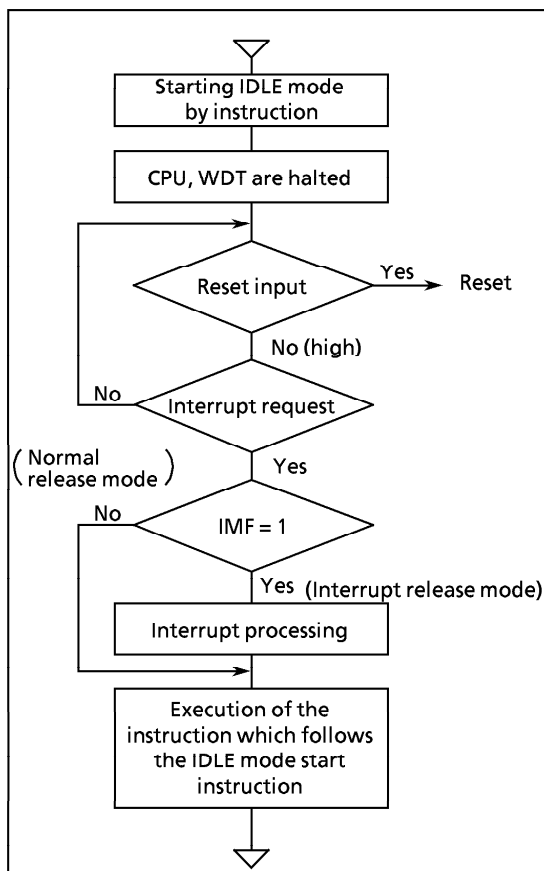


Figure 1-19. IDLE Mode

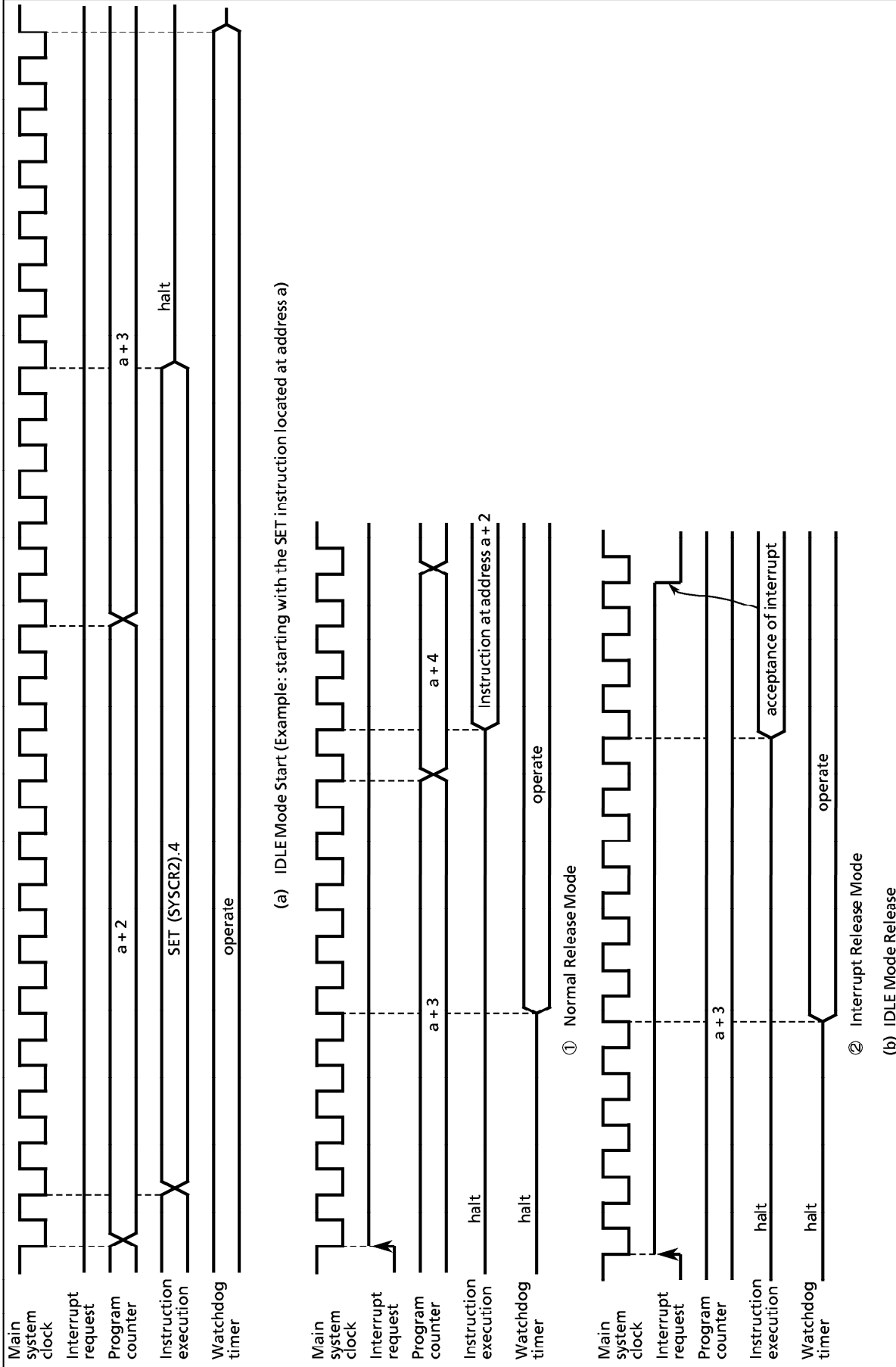


Figure 1-20. IDLE Mode Start/Release

(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)
LD      (TC2CR), 34H      ; Starts TC2
:
PINTTC2 : LD      (TC2CR), 10H ; Stops TC2
          SET      (SYSCR2) . 5 ; SYSCK←1
          CLR      (SYSCR2) . 7 ; XEN←0
          RETI
          :
VINTTC2 : DW      PINTTC2      ; INTTC2 vector table

```


b. Switching from SLOW mode to NORMAL2 mode

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

SLOW mode can also be released by setting the $\overline{\text{RESET}}$ pin low, which immediately performs the reset operation. After reset, the 87CC70/H70/K70A/M70A are placed in NORMAL mode.

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 $\overline{\text{RESET}}$ pin low, which immediately performs the reset operation. After reset, the 87CC70/H70/K70A/M70A are placed in NORMAL2 mode. (The PM70 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)
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 clock)
      RETI
      ∴
VINTTC2 : DW       PINTTC2 ; INTTC2 vector table
    
```

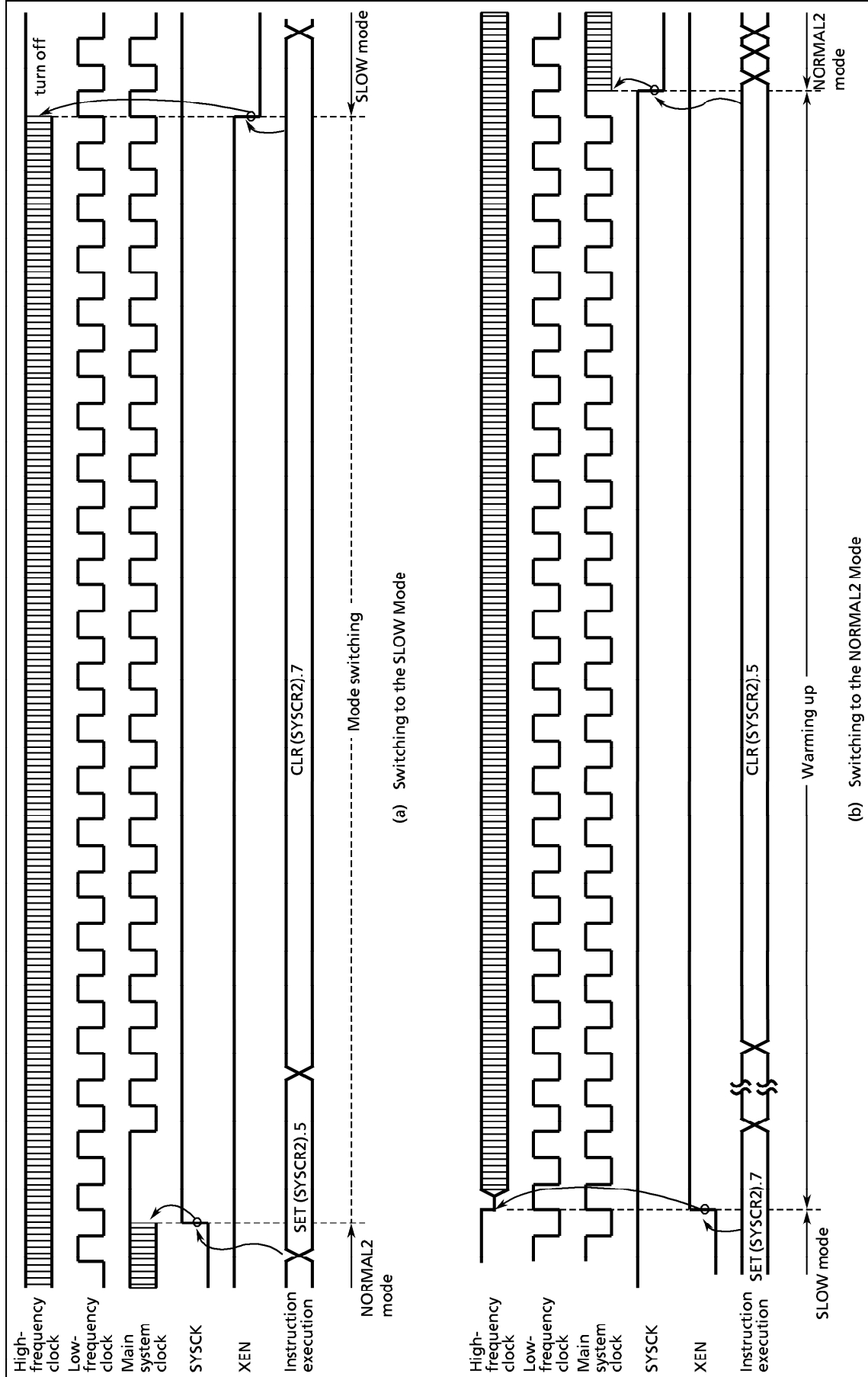


Figure 1-21. Switching between the NORMAL2 and SLOW Modes

1.9 Interrupt Controller

The 87CC70/H70/K70A/M70A each have a total of 14 interrupt sources: 5 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 | INTSIO (Serial 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 |
| Internal | INTKEY (Key scan interrupt) | IMF · EF ₁₂ = 1 | IL ₁₂ | FFE6 _H | 12 |
| reserved | | 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 (IL₁₅ to 2)

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

The interrupt latches are assigned to addresses 003C_H and 003D_H in the SFR. Each latch can be cleared to "0" individually by an instruction; however, *the read-modify-write instruction such as bit manipulation or operation instructions cannot be used (Do not clear the IL₂ for a watchdog timer interrupt 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 to IL6 ← 0
```

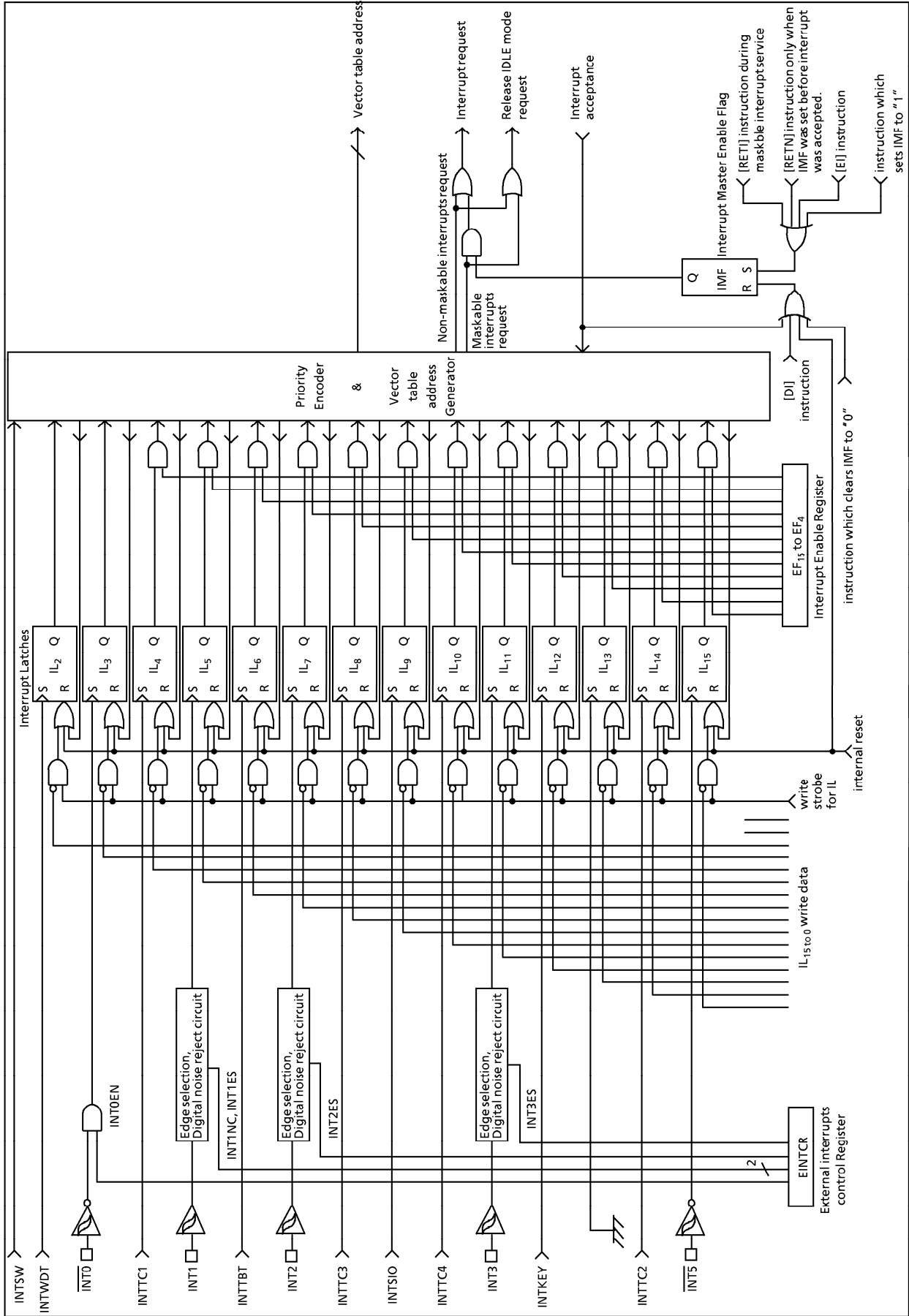


Figure 1-22. Interrupt Controller Block Diagram

Example 2 : Reads interrupt latches

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

Example 3: Tests an interrupt latch

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

(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 by 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 (EF₁₅ to EF₄)

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 to EF13, EF11, EF7, EF5, IMF←1
```

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

```
SET    (EIRH).4 ; EF12←1
```

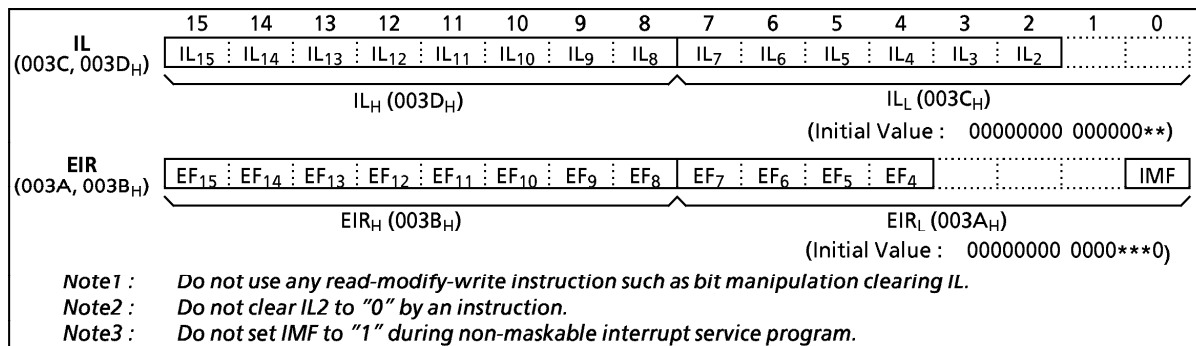


Figure 1-23. Interrupt Latch (IL) and Interrupt Enable Register (EIR)

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 @ 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).

Interrupt acceptance processing is as follows:

- 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.
- 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.

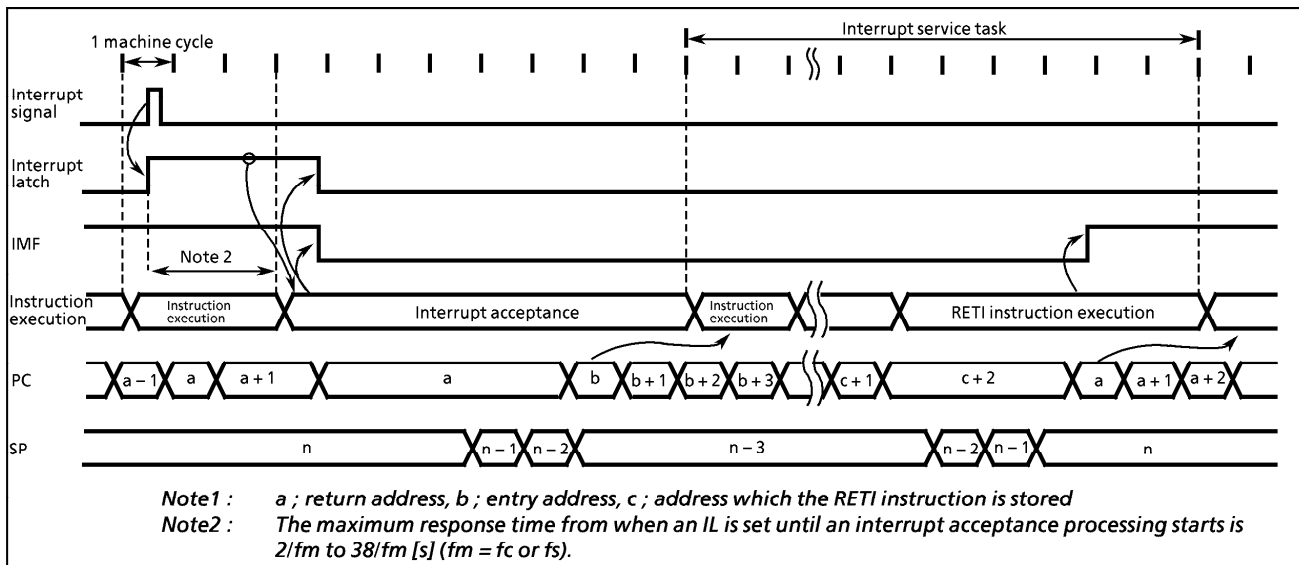
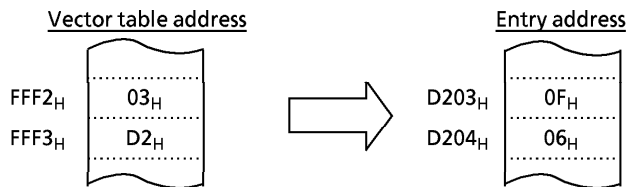


Figure 1-24. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

Example : Correspondence between vector table address for INTTB 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 INT0 pin must be disabled with the INTOEN in the external interrupt control register (EINTCR) or interrupt processing must be avoided by the program.

Example 1 : Disables an external interrupt 0 using INTOEN

```
LD (EINTCR), 0000000B ; INTOEN←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
```

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 @8 MHz)
        Interrupt processing
        RETI ; Restores bank and Returns
```

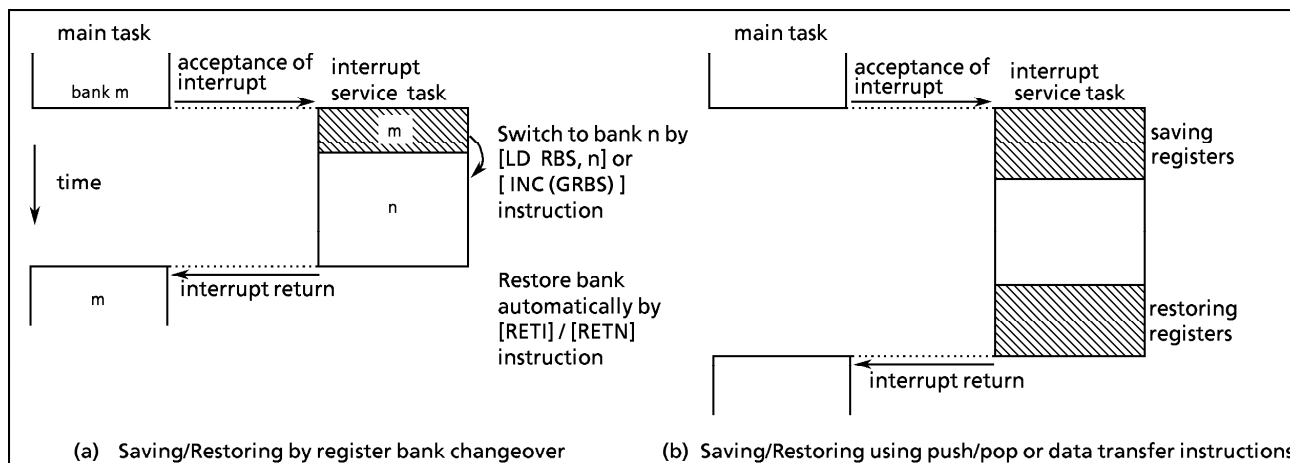
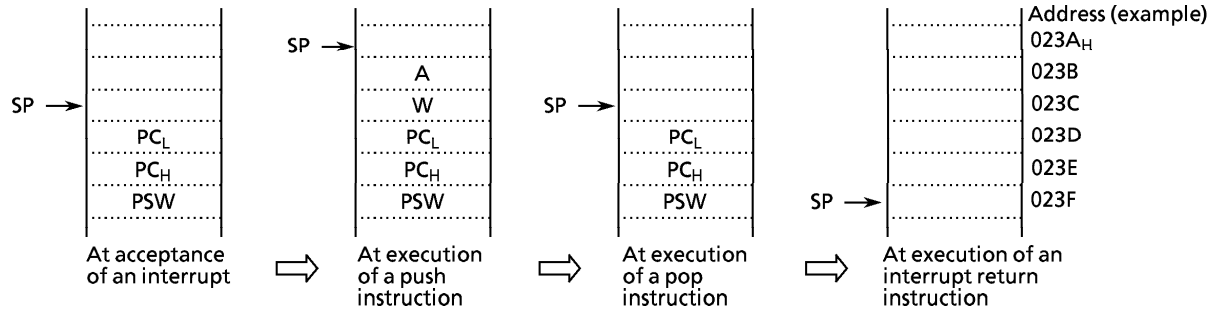


Figure 1-25. Saving/Restoring General-purpose Registers

- 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 External Interrupts

The 87CC70/H70/K70A/M70A each have five external interrupt inputs ($\overline{INT0}$, INT1, INT2, INT3, and $\overline{INT5}$). Three 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 and INT3.

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

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

Table 1-3. External Interrupts

| Source | Pin | Secondary function pin | Enable conditions | Edge | Digital noise rejection |
|--------|-------------------|------------------------|-------------------------|-----------------------------------|--|
| INT0 | $\overline{INT0}$ | P10 | $IMF = 1, INT0EN = 1$ | falling edge | — (hysteresis input) |
| INT1 | INT1 | P11 | $IMF \cdot EF_5 = 1$ | falling edge or rising edge | Pulses of less than $15/f_c$ or $63/f_c$ [s] are eliminated as noise. Pulse equal to or more than $48/f_c$ [s] or $192/f_c$ [s] are regarded as signals. |
| INT2 | INT2 | P12/TC1 | $IMF \cdot EF_7 = 1$ | | |
| INT3 | INT3 | P30/TC3 | $IMF \cdot EF_{11} = 1$ | | |
| INT5 | $\overline{INT5}$ | P20/STOP | $IMF \cdot EF_{15} = 1$ | falling edge | — (hysteresis input) |

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

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

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



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

- ① INT1 pin $49/f_c$ [s] ($INT1NC = 1$), $193/f_c$ [s] ($INT1NC = 0$)
- ② INT2,INT3 pins $25/f_c$ [s]

Note 5: When $INT0EN = 0$, interrupt latch IL_3 is not set even if a falling edge is detected for $\overline{INT0}$ pin input.

Note 6: When high-impedance is specified for port output in stop mode, port input is forcibly to low level internally. Thus, interrupt latches of external interrupt inputs except $\overline{INT5}$ (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),1111011101010111B  ; IL11,7,5,3=0 (clears interrupt latches)
EI      ; IMF=1 (enables interrupt service)
```

| EINTCR (0037 _H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value : 00*0 000*) |
|--------------------------------|----------------------------|------------|--|-------------|------------|------------|------------|---|-----------------------------|
| | INT1 NC | INT0 EN | | (TC4 ES) | INT3 ES | INT2 ES | INT1 ES | | |
| 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 | | write only | | | | |
| INT0EN | P10/INT0 pin configuration | | 0 : P10 input/output port 1 : INT0 pin (Port P10 should be set to an input mode) | | | | | | |
| INT3 ES INT2 ES INT1 ES | INT3 to INT1 edge select | | 0 : Rising edge 1 : Falling edge | | | | | | |

Note 1 : fc ; High-frequency clock [Hz] * ; don't care
Note 2 : Edge detection during switching edge selection is invalid.
Note 3 : Do not change EINTCR when IMF = 1. After changing EINTCR, interrupt latches of external interrupt inputs must be cleared to "0" using load instruction.
Note 4 : In order to change of external interrupt input by rewriting the contents of INT2ES, INT3ES and INT4ES during NORMAL1/2 mode, clear interrupt latches of external interrupt inputs (INT2, INT3 and INT4) after 8 machine cycles from the time of rewriting. During SLOW mode, 3 machine cycles are required.
Note 5 : In order to change an edge of timer counter input by rewriting the contents of INT2ES, INT3ES and INT4ES during NORMAL1/2 mode, rewrite the contents after timer counter is stopped (TC*s = 0), that is, interrupt disable state. Then, clear interrupt latches of external interrupt inputs (INT2, INT3 and INT4) after 8 machine cycles from the time of rewriting to change to interrupt enable state. Finally, start timer counter. During SLOW mode, 3 machine cycles are required.

Example : When changing TC1 pin inputs edge in external trigger timer mode from rising edge to falling edge. (example: TMP87CH00N)

```

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

```

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

Figure 1-26. External Interrupt Control Register

1.9.3 Software Interrupt (INTSW)

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

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 87CK70A/M70A, BF80_H to BFFF_H for 87CC70/H70 is not "FF_H".

- ② Debugging
Debugging efficiency can be increased by placing the SWI instruction at the software break point setting address.

1.10 Watchdog Timer (WDT)

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

The watchdog timer signal for detecting malfunction can be selected either 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

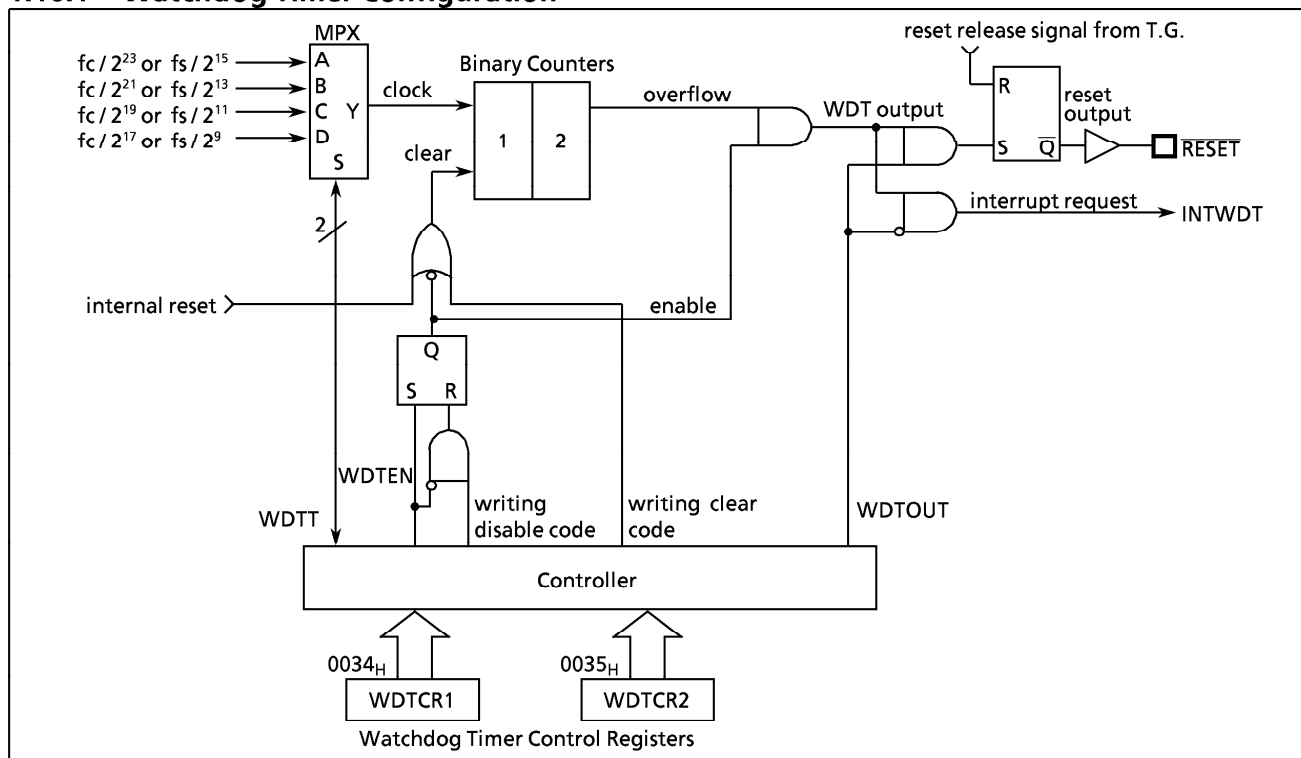


Figure 1-27. Watchdog Timer Configuration

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

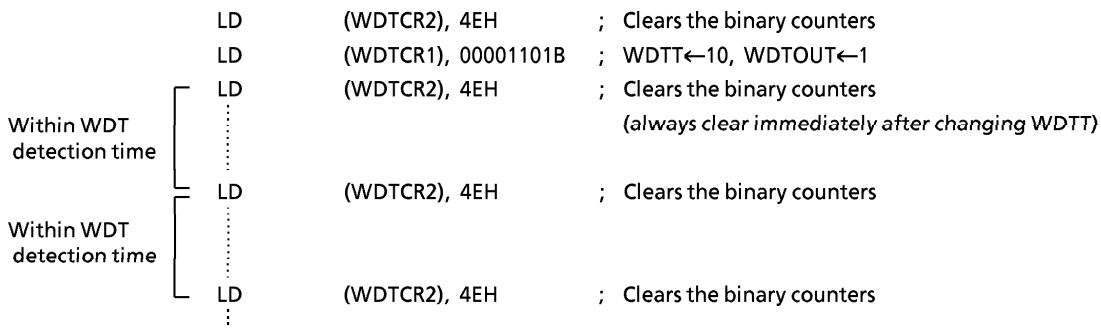
The CPU malfunction is detected as follows:

- 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 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 $2^{21}/f_c$ [s] and resets the CPU malfunction.



Watchdog Timer Control Register 1

WDTCR1 (0034_H)

| | | | | | | | |
|-----------------------------|---|---|---|--------|------|---------|-----------------------------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WDTCR1 (0034 _H) | | | | WDT EN | WDTT | WDT OUT | (Initial value : **** 1001) |

| | | | |
|--------|-------------------------------|--|------------|
| WDTCR1 | Watchdog timer enable/disable | 0 : Disable (It is necessary to write the disable code to WDTCR2) 1 : Enable | write only |
| WDTCR1 | Watchdog timer detection time | 00 : $2^{25}/f_c$ or $2^{17}/f_s$ [s] 01 : $2^{23}/f_c$ or $2^{15}/f_s$ 10 : $2^{21}/f_c$ or $2^{13}/f_s$ 11 : $2^{19}/f_c$ or $2^{11}/f_s$ | |
| WDTCR1 | 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 : f_c ; High-frequency clock [Hz] f_s ; 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 | |
| WDTCR2 (0035 _H) | | | | | | | | (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

Figure 1-28. Watchdog Timer Control Registers

Table 1-4. Watchdog Timer Detection Time

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

(2) Watchdog Timer Enable

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

Example : Enables watchdog timer

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

(3) Watchdog Timer Disable

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

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

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 $2^{20}/f_c$ [s] (131 ms at $f_c = 8$ MHz). The high-frequency clock oscillator also turns on when a watchdog timer reset is generated in SLOW mode.

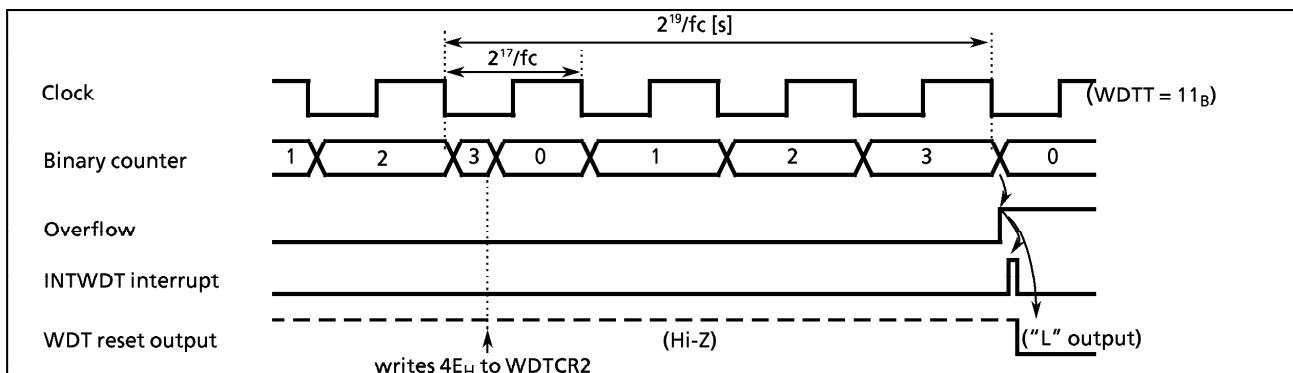


Figure 1-29. Watchdog Timer Interrupt / Reset

1.11 Reset Circuit

The 87CC70/H70/K70A/M70A 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 ($2^{20}/f_c$ [s]) (131 ms 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 $\overline{\text{RESET}}$ pin is held at low for at least 3 machine cycles ($12/f_c$ [s]) with the power supply voltage within the operating voltage range and oscillation stable, a reset is applied and the internal state is initialized.

When the $\overline{\text{RESET}}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses FFFE_H - FFFF_H. The $\overline{\text{RESET}}$ pin contains a Schmitt trigger (hysteresis) with an internal pull-up resistor. A simple power-on-reset can be applied by connecting an external capacitor and a diode.

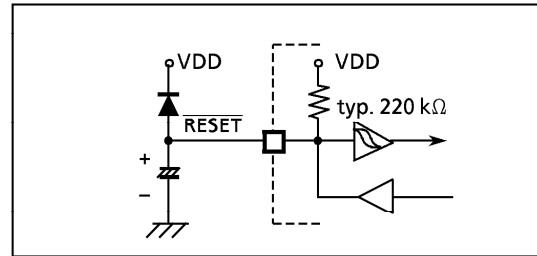


Figure 1-30. Simple Power-on-Reset Circuitry

1.11.2 Address Trap Reset

If a CPU malfunction occurs and an attempt is made to fetch an instruction from the RAM or the SFR area (addresses 0000_H - 023F_H), an address-trap-reset will be generated. Then, the $\overline{\text{RESET}}$ pin output will go low. The reset time is $2^{20}/f_c$ [s] (131 ms @ 8 MHz).

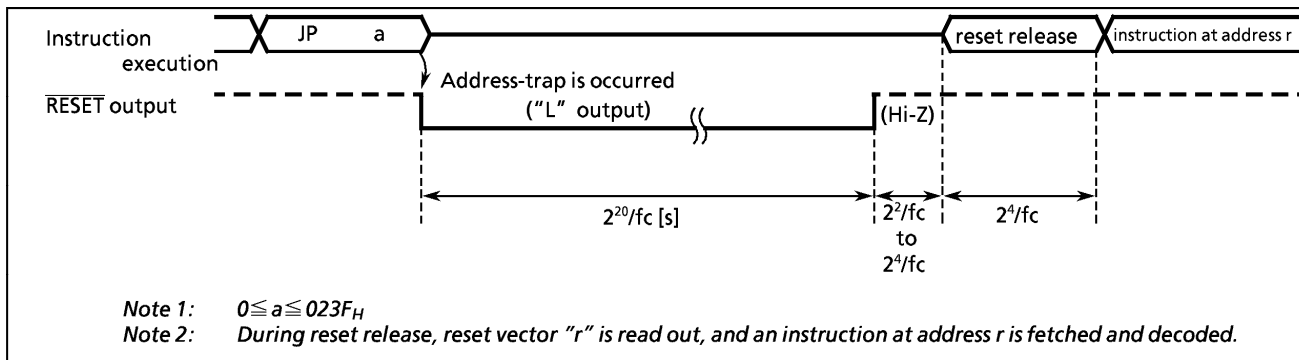


Figure 1-31. Address-Trap-Reset

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 $2^{20}/f_c$ [s] (131 ms 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 87CC70/H70/K70A/M70A SFRs and DBRs.

| Address | Read | Write | Address | Read | Write |
|-------------------|----------------------------|--|-------------------|---------------------------|--|
| 0000 _H | | P0 port | 0020 _H | SIOSR (SIO status) | SIOCR1 (SIO control) |
| 01 | | P1 Port | 21 | — | SIOCR2 |
| 02 | | P2 Port | 22 | | reserved |
| 03 | | P3 Port | 23 | | reserved |
| 04 | | P4 Port / KEYDR | 24 | HSOSR (HSO status) | HSOCR (HSO control) |
| 05 | | P5 Port | 25 | | reserved |
| 06 | | P6 Port | 26 | | reserved |
| 07 | | P7 Port | 27 | | reserved |
| 08 | | P8 Port | 28 | — | VFTCR1 (VFT control) |
| 09 | | P9 Port | 29 | VFTSR (VFT status) | VFTCR2 |
| 0A | — | POCR (P0 I/O control) | 2A | KEYSR (Key scan status) | — |
| 0B | — | P1CR (P1 I/O control) | 2B | | reserved |
| 0C | — | P4CR (P4 control) | 2C | | reserved |
| 0D | | reserved | 2D | | reserved |
| 0E | — | CMPCR (Comparator input control) | 2E | | reserved |
| 0F | | CMPDR (Comparator input data register) | 2F | | reserved |
| 10 | — | TREG1 _L (Timer register 1) | 30 | | reserved |
| 11 | — | TREG1 _H | 31 | | reserved |
| 12 | | reserved | 32 | | reserved |
| 13 | | reserved | 33 | | reserved |
| 14 | — | TC1CR (TC1 control) | 34 | — | WDTCR1 (WDT control) |
| 15 | — | TC2CR (TC2 control) | 35 | — | WDTCR2 |
| 16 | — | TREG2 _L (Timer register 2) | 36 | — | TBTCR (TBT / TG / DVO control) |
| 17 | — | TREG2 _H | 37 | — | EINTCR (Interrupt control) |
| 18 | | TREG3A (Timer register 3A) | 38 | | SYSCR1 (System control) |
| 19 | TREG3B (Timer register 3B) | — | 39 | | SYSCR2 |
| 1A | — | TC3CR (TC3 control) | 3A | | EIR _L (Interrupt enable register) |
| 1B | — | TREG4 (Timer register 4) | 3B | | EIR _H |
| 1C | — | TC4CR (TC4 control) | 3C | | IL _L (Interrupt latch) |
| 1D | | reserved | 3D | | IL _H |
| 1E | | reserved | 3E | | reserved |
| 1F | | reserved | 3F | PSW (Program status word) | RBS (Register bank selector) |

(a) Special Function Registers

| Address | Read | Write |
|-------------------|--------------------------------------|--------------------------|
| 0F80 _H | | |
| | VFT Display data buffer | |
| 0F9F | | |
| | reserved | |
| 0FF0 | | |
| F1 | | |
| F2 | | |
| F3 | SIO Transmit and receive data buffer | |
| F4 | | |
| F5 | | |
| F6 | | |
| F7 | | |
| 0FF8 | — | HSO transmit data buffer |
| | | reserved |
| 0FFF | | |

(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.)

Note 5 : KEYDR is a read-only register.

Figure 2-1. SFR & DBR

2.2 I/O Ports

The 87CC70/H70/K70A/M70A each have 10 parallel input/output ports (73pins) each as follows:

| | Primary Function | Secondary Functions |
|---------|-------------------|---|
| Port P0 | 8-bit I/O port | |
| Port P1 | 8-bit I/O port | External interrupt input, timer/counter input, 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 | 8-bit I/O port | Serial interface, external interrupt input, and timer/counter input |
| Port P4 | 8-bit I/O port | Key scan input and compartor input |
| Port P5 | 6-bit I/O port | Compartor input and timer/counter output |
| Port P6 | 8-bit I/O port | VFT digit driver output |
| Port P7 | 8-bit I/O port | VFT digit driver output |
| Port P8 | 8-bit output port | VFT segment driver output or key strobe output |
| Port P9 | 8-bit I/O port | VFT segment driver output or key strobe output |

Ports P1, P2, P3, P4, P5, P6, P7, P8 and P9 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.

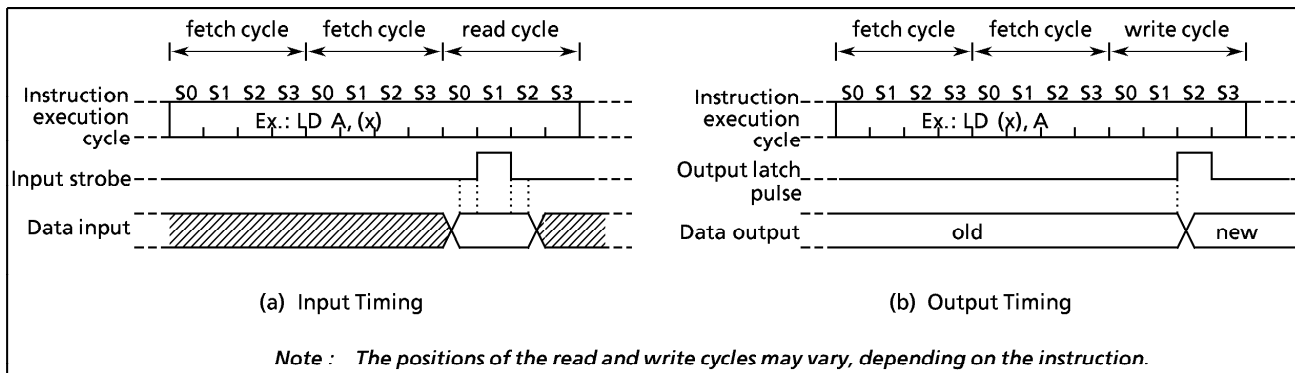


Figure 2-2. Input/Output Timing (Example)

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.

Note : The port set to an input mode reads pin input state. When used with input/output modes, output latch contents of the port set to an input mode may be placed by execution of bit operate instruction.

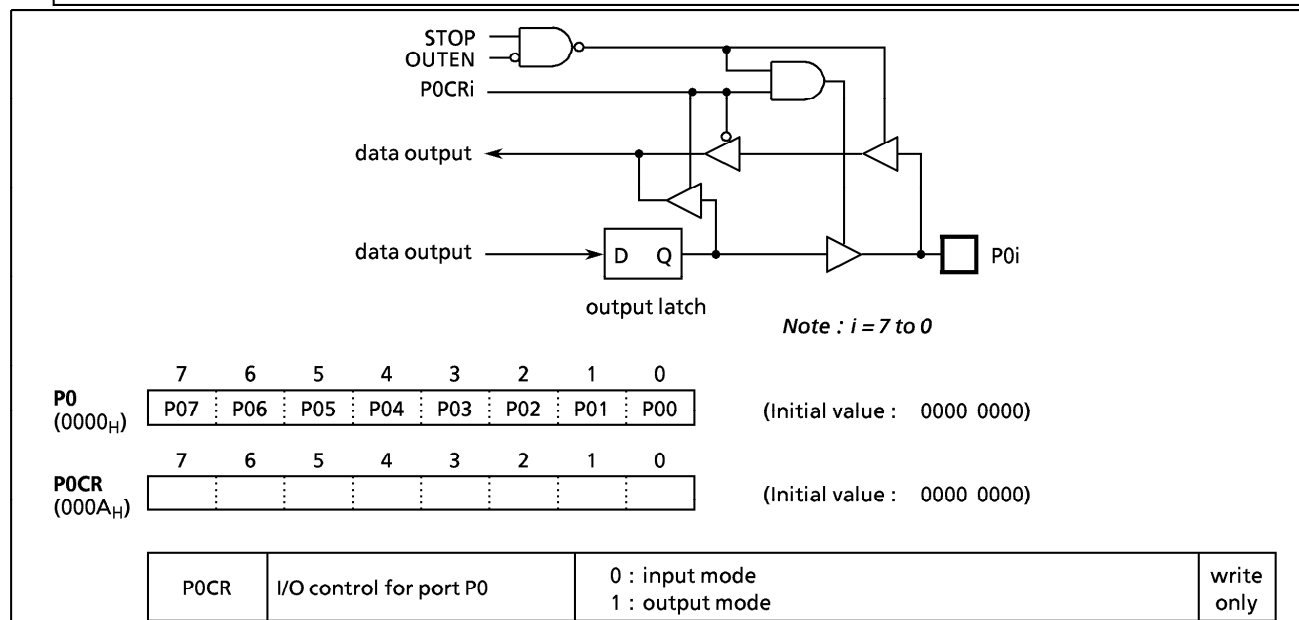


Figure 2-3. Port P0 and P0CR

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. Therefore initial output data should be written into the output latch before setting P1CR. Port P1 is also used as an external interrupt input, a timer/counter input, and a divider output. When used as 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 P12 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 ($\overline{\text{INT0}}$) 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 ($\overline{\text{INT0}}$) is configured as an input port P10.

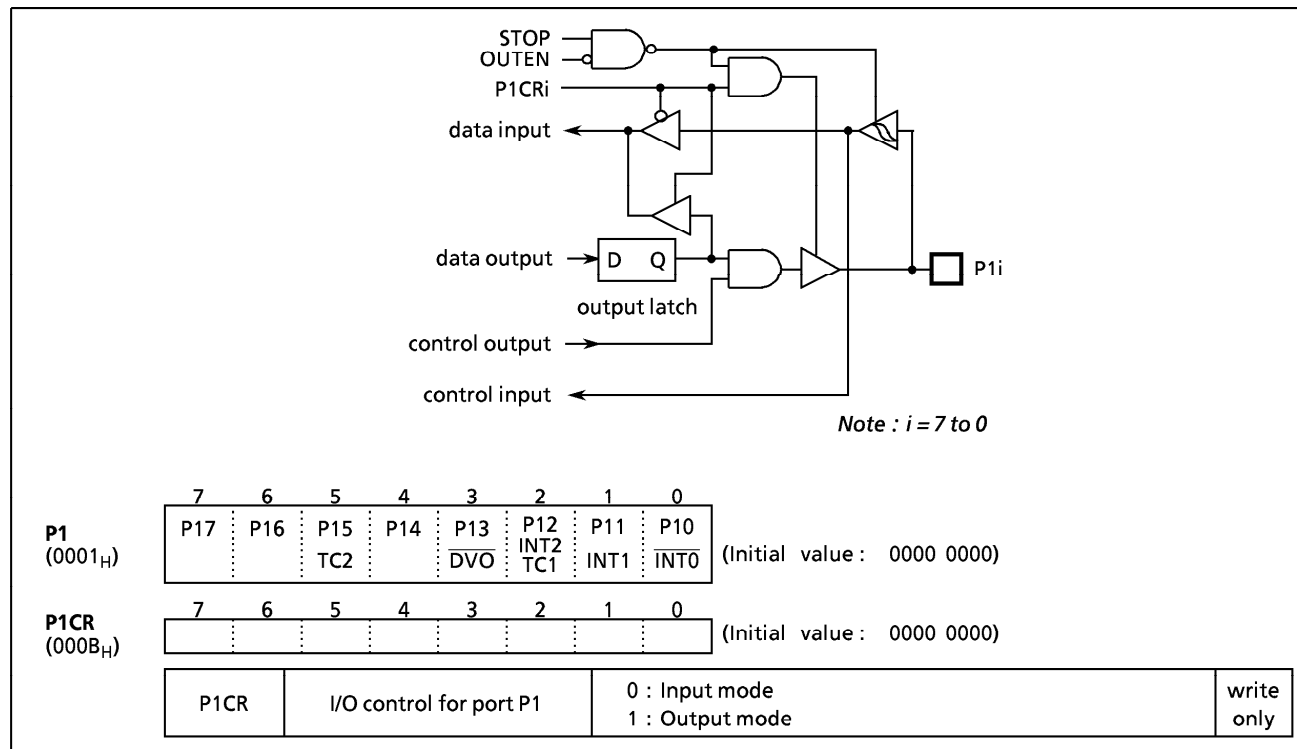


Figure 2-4. Port P1 and P1CR

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

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 87CC70/H70/K70A/M70A are placed in NORMAL2 mode. (The PM70 is placed in NORMAL1 mode)

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.

2.2.4 Port P3 (P37 - P30)

Port P3 is an 8-bit input/output port, and is also used as serial interface input/output, an external interrupt input, and a timer/counter input. When used as an input port or a secondary function pin, the output latch should be set to "1". The output latches are initialized to "1" during reset.

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

```
LD (P3), 5AH ; P3 ← 5AH
```

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

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

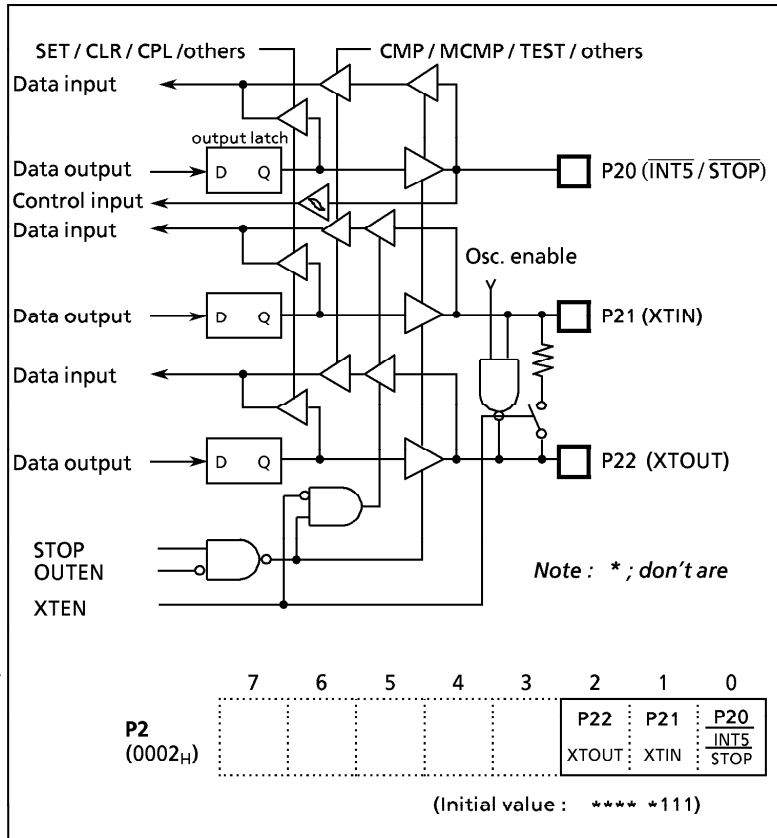


Figure 2-5. Port P2

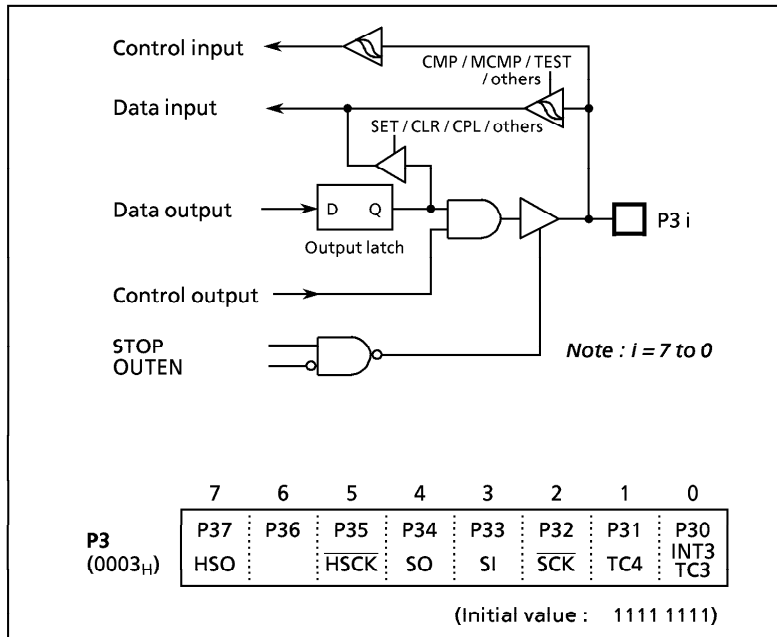


Figure 2-6. Port P3

2.2.5 Port P4 (P47 - P40)

Port P4 is an 8-bit input/output port, and is also used as a key scan input and a comparator input. When used as an input port or a secondary function pin, the output latch should be set to "1". The output latches are initialized to "1" during reset.

When used key scan function, all of the port P4 bits must be used as key scan inputs or comparator inputs. When used as both comparator inputs and key scan inputs, the P47 and P46 pins are used as only comparator inputs. Bits 7 and 6 are read in as "0" when a read instruction is executed for the key scan input latches (KEYDR). Built-in pull-down resistors can also be connected for each bit using the port P4 control register (refer to section "2.12.6 (2) Key scan input pins")

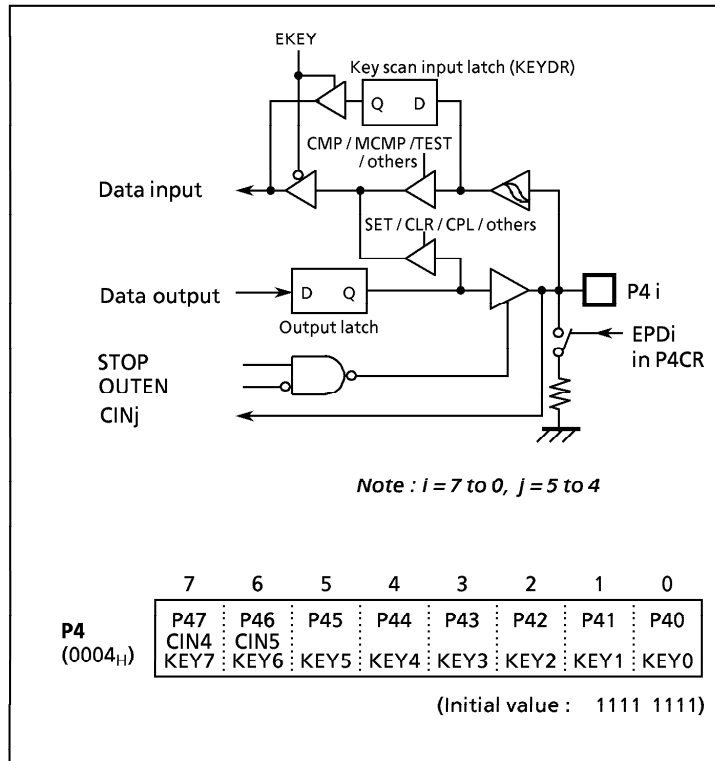


Figure 2-7. Port P4

2.2.6 Port P5 (P55 - P50)

Port P5 is a 6-bit input/output port, and is also used as a comparator input, an 8-bit PWM (Pulse Width Modulation) output and an 8-bit programmable divider output. When used as an input port or a secondary function pin, the output latch should be set to "1". The output latches are initialized to "1" during reset.

Bits 6 and 7 are read in as "1" when a read instruction is executed for the port P5.

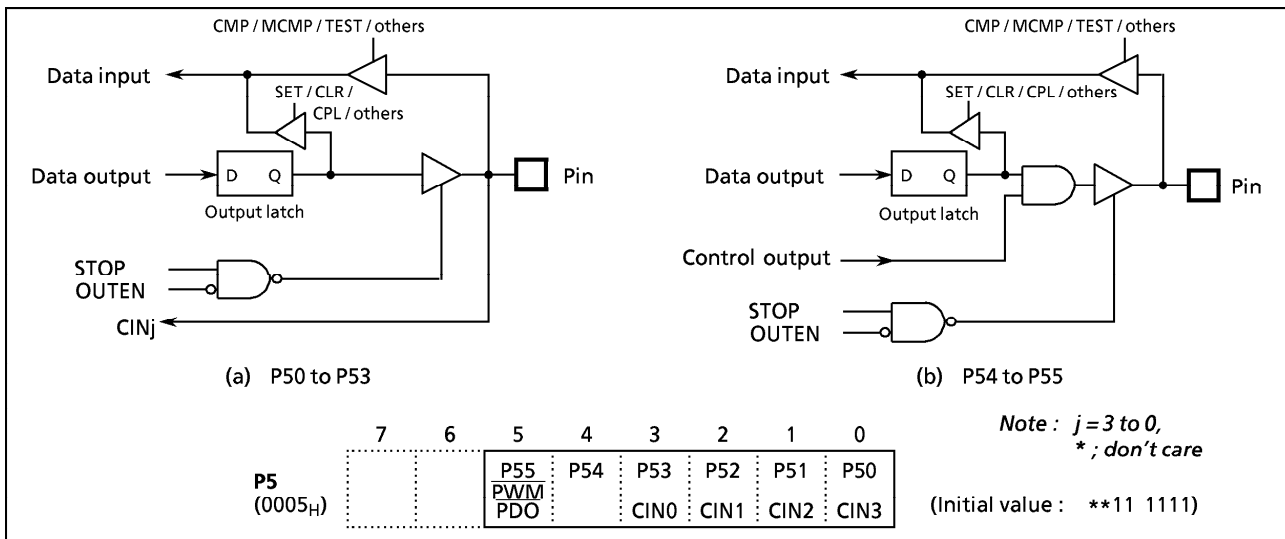


Figure 2-8. P5 Port

2.2.7 Ports P6 (P67 - P60) and P7 (P77 - P70)

Ports P6 and P7 are 8-bit high-breakdown voltage input/output ports, and are also used as digit outputs, which can directly drive vacuum fluorescent tube (VFT). When used as an input port or a digit output, the output latch should be cleared to "0". Pins which are not set for digit 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.

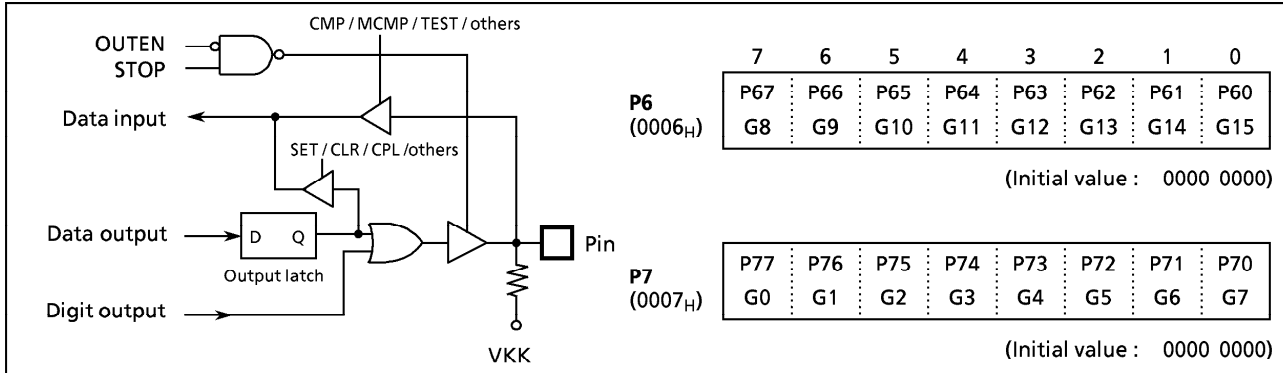


Figure 2-9. Ports P6 and P7

2.2.8 Port P8 (P87 - P80)

Port P8 is an 8-bit high-breakdown voltage output port, and also used as a segment output (or a key strobe output), which can directly drive vacuum fluorescent tube (VFT). When used as an input port or a segment (key strobe) output, the output latch should be cleared to "0". The output latches are initialized to "0" during reset.

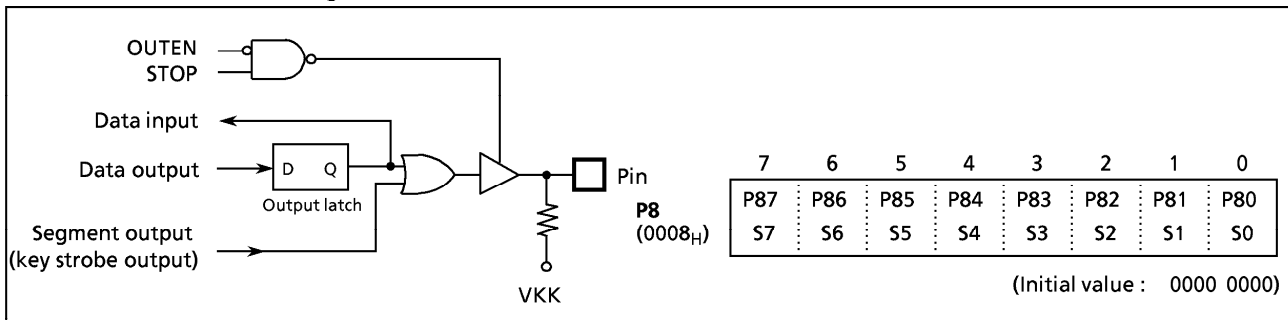


Figure 2-10. Port P8

2.2.9 Port P9 (P97 - P90)

Port P9 is an 8-bit high-breakdown voltage input/output port, and also used as a segment output (or a key strobe output), which can directly drive vacuum fluorescent tube (VFT). When used as an input port or a segment (key strobe) output, the output latch should be cleared to "0". Pins which are not set for segment 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.

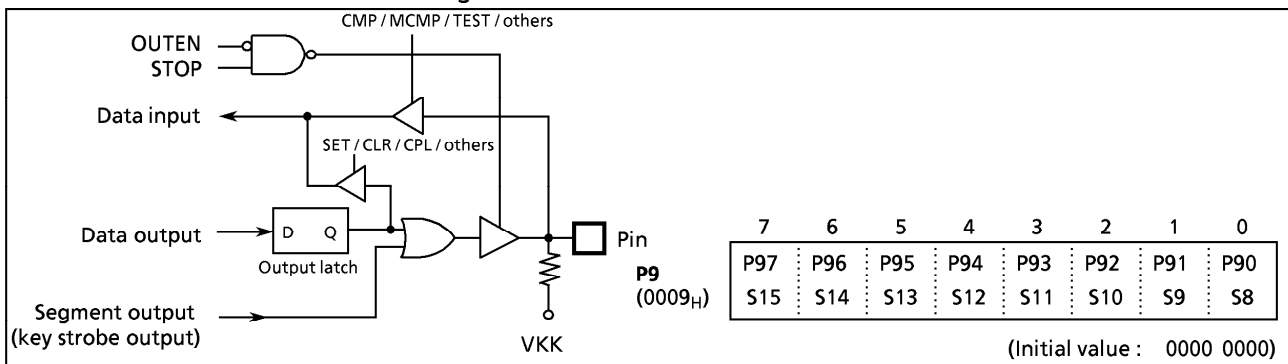


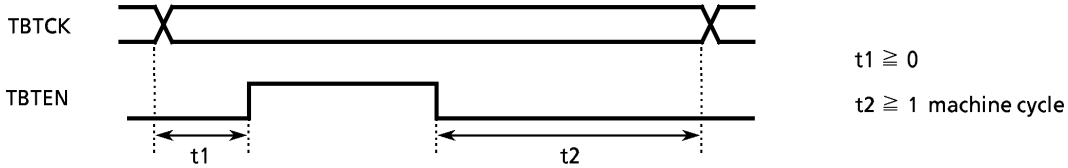
Figure 2-11. Port P9

2.3 Time Base Timer (TBT)

The time base timer generates time base for key scanning, dynamic displaying, etc. It also provides a time base timer interrupt (INTTBT). The time base timer is controlled by the control register (TBTCR) shown in Figure 2-13.

An INTTBT is generated on the first rising edge of source clock (the divider output of the timing generator) after the time base timer has been enabled. The divider is not cleared by the program; therefore, only the first interrupt may be generated ahead of the set interrupt period.

The interrupt frequency (TBTCCK) must be selected with the time base timer disabled (both frequency selection and enabling can be performed simultaneously).



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

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

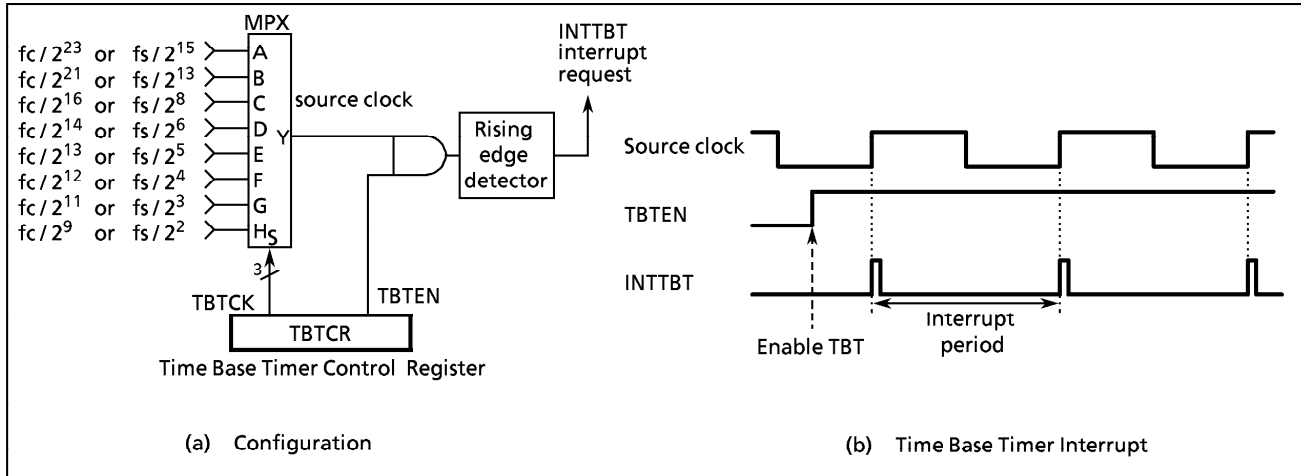


Figure 2-12. Time Base Timer

| | | | | | | | | | |
|-------------------------------|--|---------|---------|-------|--|---|---|---|-----------------------------|
| TBTCR (0036 _H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value : 0**0 0***) |
| | (DVOEN) | (DVQCK) | (DV7CK) | TBTEN | TBTCCK | | | | |
| TBTEN | Time base timer enable/disable | | | | 0 : Disable 1 : Enable | | | | write only |
| TBTCCK | 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 $fs/2$ | | | | |

Note1 : fc ; High-frequency clock [Hz], fs ; Low-frequency clock [Hz], * ; don't care
 Note2 : TBTCR is a write-only register and must not be used with any of the read-modify-write instructions.

Figure 2-13. 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 (TBTCCR) shown in Figure 2-14. Note that TBTCCR is a write-only register, and must not be used with any of the read-modify-write instructions.

| | | | | | | | | | |
|--------------------------------|--|-------|---|---------|---------|---------------|---|---|-----------------------------|
| TBTCCR (0036 _H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value : 0**0 0***) |
| | DVOEN | DVOCK | (DV7CK) | (TBTEN) | (TBTCK) | | | | |
| DVOEN | Divider output enable/disable | | 0 : Disable 1 : Enable | | | write only | | | |
| DVOCK | Divider output (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 2-14. 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       (TBTCCR), 10000000B ; DVOEN ← 1, DVOCK ← 00
```

Table 2-2. Frequency of Divider Output

| DVOCK | Frequency of Divider Output | At fc = 8 MHz | At fs = 32 kHz |
|-------|-----------------------------|---------------|----------------|
| 00 | $fc / 2^{13}$ or $fs / 2^5$ | 0.97 [kHz] | 1 [kHz] |
| 01 | $fc / 2^{12}$ or $fs / 2^4$ | 1.95 | 2 |
| 10 | $fc / 2^{11}$ or $fs / 2^3$ | 3.90 | 4 |
| 11 | $fc / 2^{10}$ or $fs / 2^2$ | 7.81 | 8 |

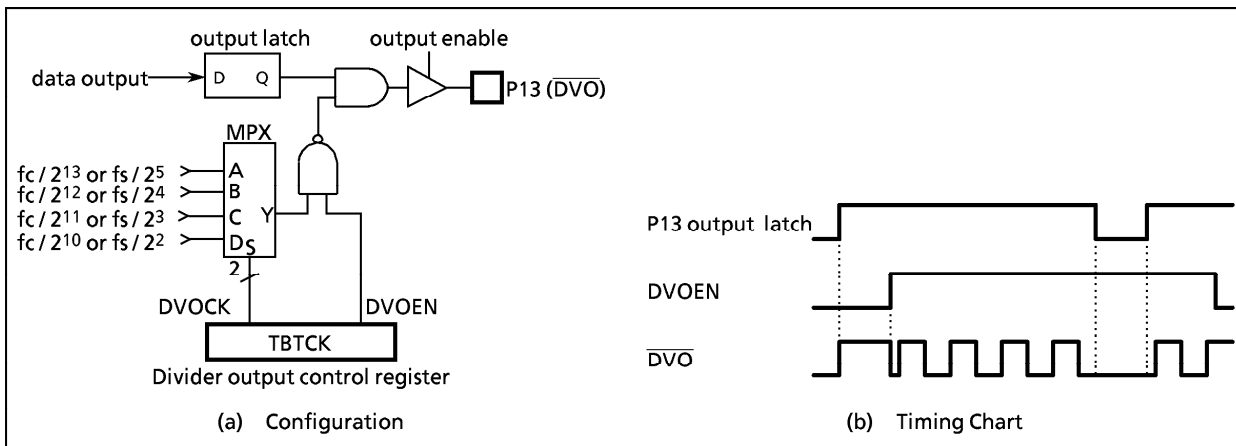


Figure 2-15. Divider Output

2.5 16-bit Timer/Counter 1 (TC1)

2.5.1 Configuration

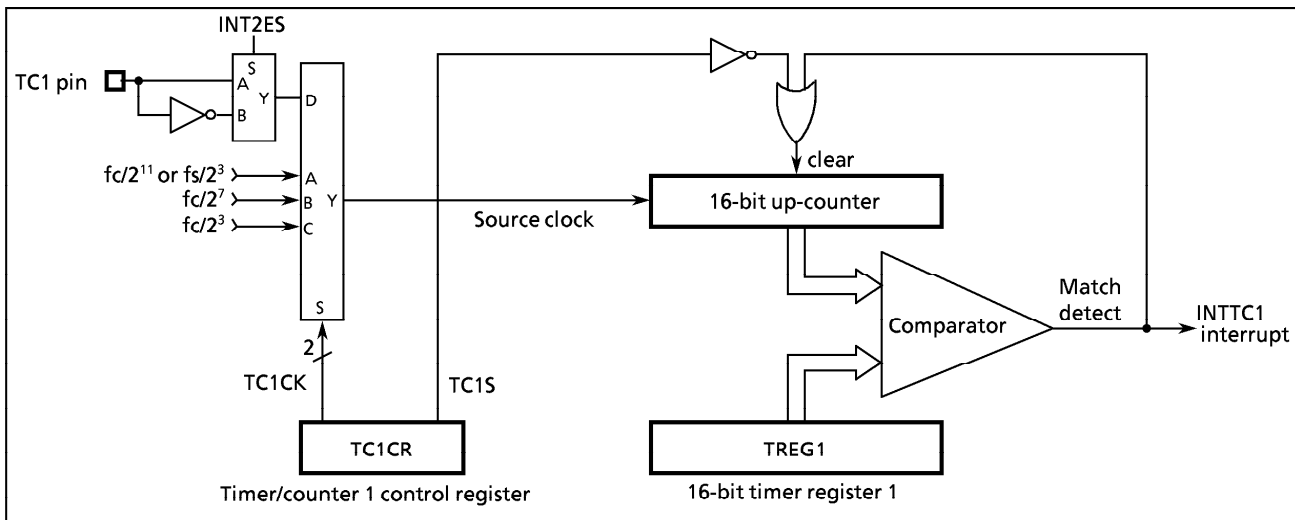


Figure 2-16. Timer/Counter 1

2.5.2 Control

The timer/counter 1 is controlled by a timer/counter 1 control register (TC1CR) and a 16-bit timer register (TREG1). Reset does not affect TREG1.

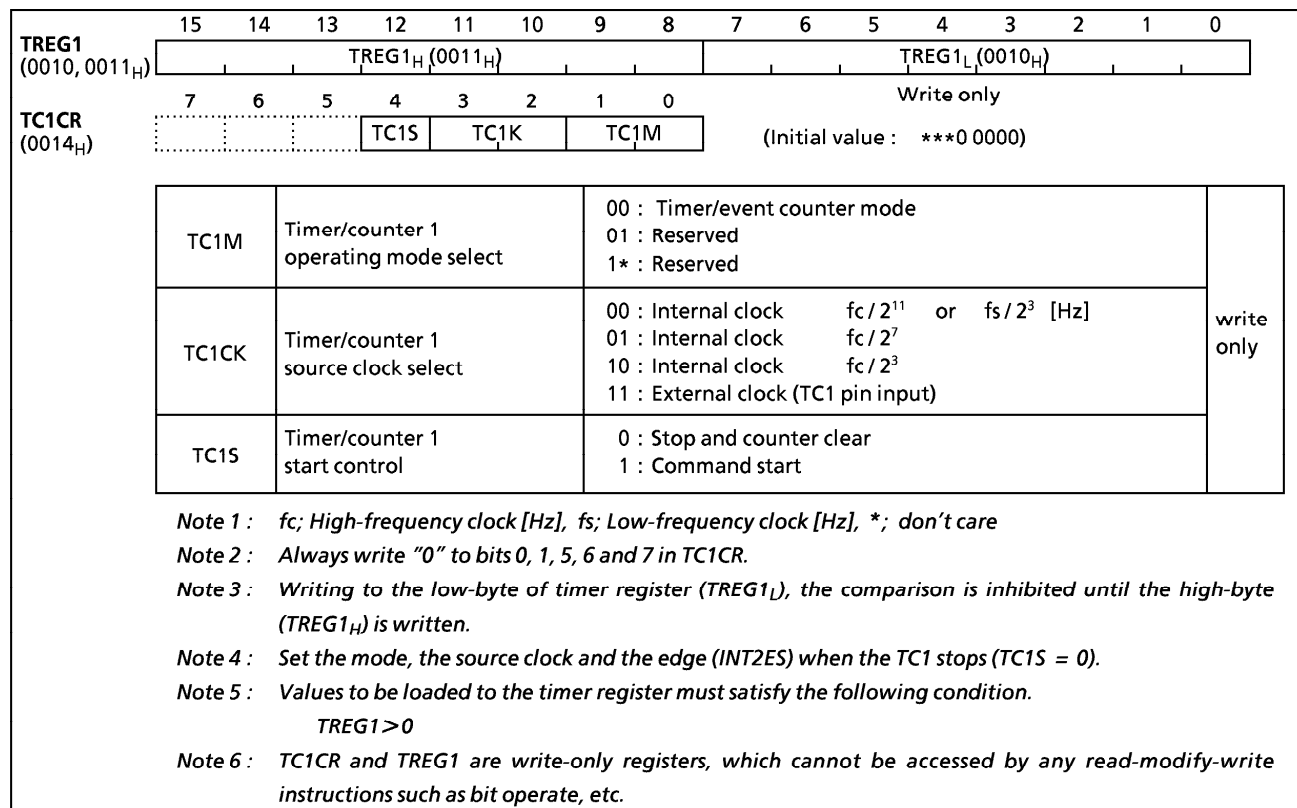


Figure 2-17. Timer Register 1 and TC1 Control Register

2.5.3 Function

Timer/counter 1 has two operating modes: timer mode and event counter mode.

(1) Timer Mode

In this mode, counting up is performed using the internal clock. The contents of TREG1 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 counter is cleared.

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

| Source clock | | SLOW, SLEEP modes | Resolution | | Maximum time setting | |
|--------------------------|-----------------|-------------------|-----------------|----------------------|----------------------|----------------------|
| NORMAL1/2, IDLE1/2 modes | | | At $fc = 8$ MHz | At $fs = 32.768$ kHz | At $fc = 8$ MHz | At $fs = 32.768$ kHz |
| DV7CK = 0 | DV7CK = 1 | | | | | |
| $fc / 2^3$ [Hz] | $fc / 2^3$ [Hz] | – | 1 μ s | | 65.5 ms | |
| $fc / 2^7$ | $fc / 2^7$ | – | 16 μ s | | 1.0 s | |
| $fc / 2^{11}$ | $fs / 2^3$ | $fs / 2^3$ [Hz] | 256 μ s | 244.14 μ s | 16.7 s | 16.0 s |

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

```
LD      (TC1CR), 00000000B      ; Sets the TC1 mode and source clock
LDW     (TREG1), 1000H          ; Sets the timer register ( $1\text{ s} \div 2^3 / f_s = 1000_{10}$ )
LD      (TC1CR), 00010000B      ; Starts TC1
```

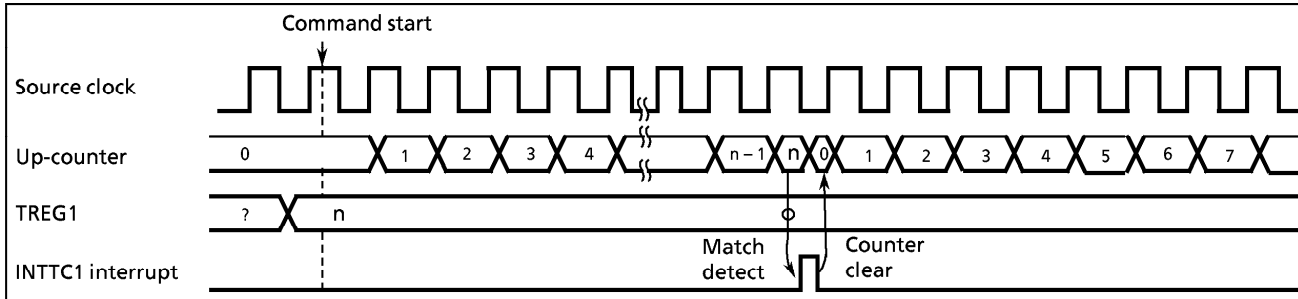


Figure 2-18. Timer Mode Timing Chart

(2) Event Counter Mode

In this mode, events are counted on the edge of the TC1 pin input. Either the rising or falling edge can be selected with INT2ES in EINTCR. The contents of TREG1 are compared with the contents of the up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. The maximum applied frequency is $f_c/2^4$ [Hz] in NORMAL1/2 or IDLE1/2 mode and $f_s/2^4$ [Hz] in SLOW or SLEEP mode.

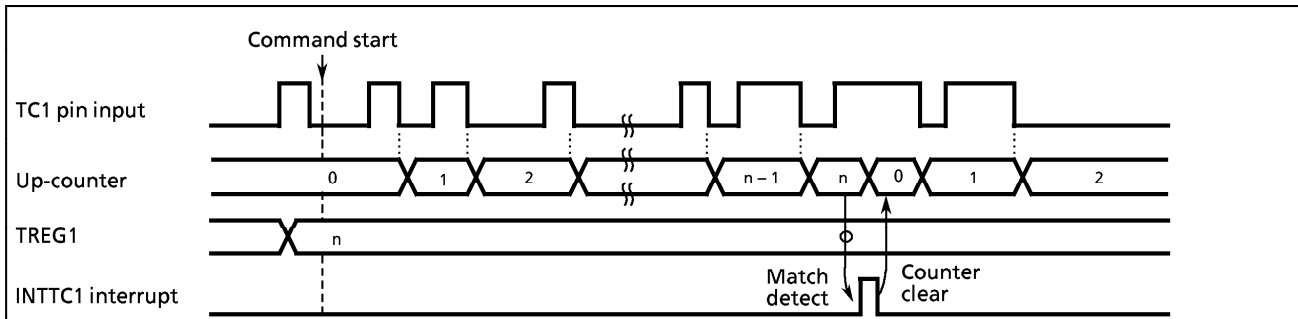


Figure 2-19. Event Counter Mode Timing Chart (INT2ES = 1)

2.6 16-bit Timer/Counter 2 (TC2)

2.6.1 Configuration

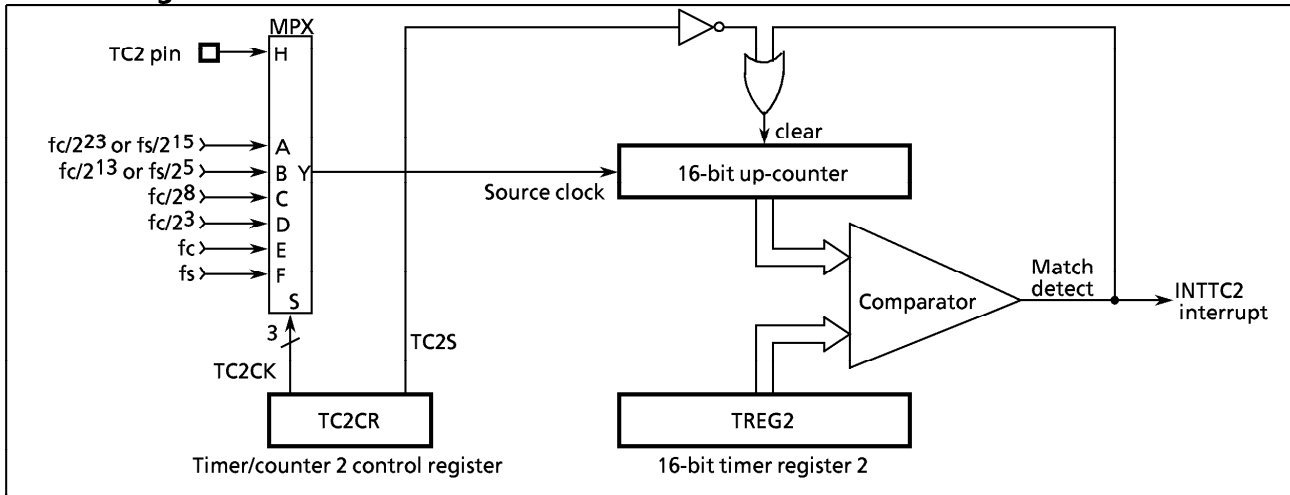


Figure 2-20. Timer/Counter 2

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 | (Initial value : **00 00*0) | | | | | | | |
| | | TC2S | | TC2CK | | | TC2M | | | | | | | | | | |
| TC2M | Timer/counter 2 operating mode select | 0 : Timer/event counter mode 1 : Reserved | | | | | | | | | | | | | | | |
| TC2CK | Timer/counter 2 source clock select | 000 : Internal clock $fc / 2^{23}$ or $fs / 2^{15}$ [Hz] 001 : Internal clock $fc / 2^{13}$ or $fs / 2^5$ 010 : Internal clock $fc / 2^8$ 011 : Internal clock $fc / 2^3$ 100 : Internal clock fc (Note 5) 101 : Internal clock fs 110 : Reserved 111 : External clock (TC2 pin input) | | | | | | | | | | | | | | | |
| TC2S | Timer/counter 2 start control | 0 : Stop and counter clear 1 : Start | | | | | | | | | | | | | | | |
| | | write only | | | | | | | | | | | | | | | |

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 stop (TC2S = 0).
 Note 4 : Values to be loaded to the timer register must satisfy the following condition.
 $TREG2 > 0$ (TREG2_{15~11} > 0 when warm-up).
 Note 5 : "fc" can be selected as the source clock only in the timer mode during 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 read-modify-write instructions.

Figure 2-21. Timer Register 2 and TC2 Control Register

2.6.3 Function

The timer/counter 2 has two operating modes: timer and event counter mode. 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 timer register 2 (TREG2) are compared with the contents of the 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 up-counter is cleared.

Also, when f_c 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 | At $f_c = 8$ MHz | At $f_s = 32.768$ kHz | At $f_c = 8$ MHz | At $f_s = 32.768$ kHz |
| DV7CK = 0 | DV7CK = 1 | | | | | | |
| $f_c / 2^{23}$ [Hz] | $f_s / 2^{15}$ [Hz] | $f_s / 2^{15}$ [Hz] | $f_s / 2^{15}$ [Hz] | 1.05 s | 1 s | 19.1 h | 18.2 h |
| $f_c / 2^{13}$ | $f_s / 2^5$ | $f_s / 2^5$ | $f_s / 2^5$ | 1.02 ms | 1 ms | 1.1 min | 1 min |
| $f_c / 2^8$ | $f_c / 2^8$ | - | - | 32 μ s | - | 2.1 s | - |
| $f_c / 2^3$ | $f_c / 2^3$ | - | - | 1 μ s | - | 65.5 ms | - |
| - | - | f_c (Note) | - | 125 ns | - | 7.9 ms | - |
| f_s | f_s | - | - | - | 30.5 μ s | - | 2 s |

Note : "fc" can be used only in the timer mode when switching from the SLOW mode to the NORMAL2 mode.

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

```
LD      (TC2CR), 00001100B      ; Sets the TC2 mode and source clock
LDW     (TREG2), 61A8H          ; Sets TREG2 (25 ms ÷ 23/fc = 61A8H)
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 $f_c/2^4$ [Hz] in NORMAL1/2 or IDLE1/2 mode, and $f_s/2^4$ [Hz] in SLOW or SLEEP mode.

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

```
LD      (TC2CR), 00011100B      ; Sets the TC2 mode
LDW     (TREG2), 0280H          ; Sets TREG2
LD      (TC2CR), 00111100B      ; Starts TC2
```

2.7 8-Bit Timer/Counter 3 (TC3)

2.7.1 Configuration

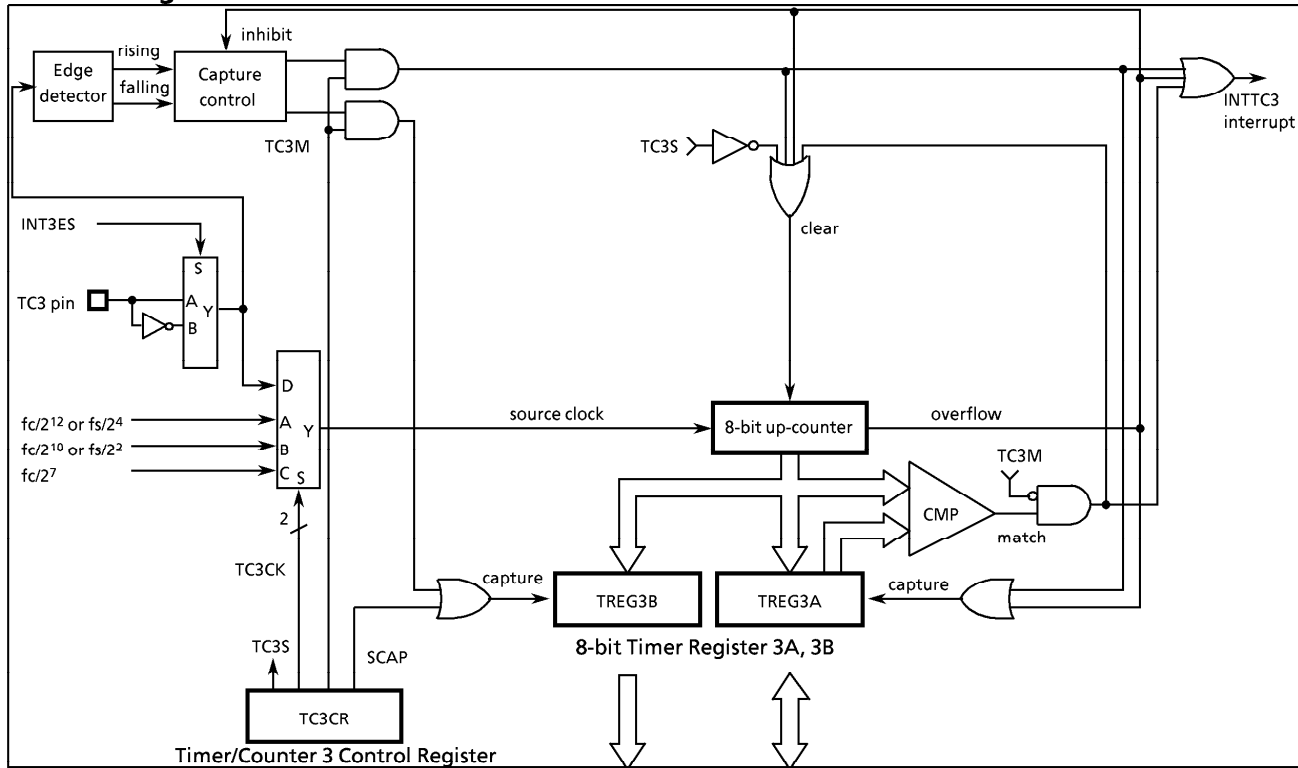


Figure 2-22. Timer/Counter 3

2.7.2 Control

| | | | |
|--------------------------------|-------------------------------------|--|-----------------------------|
| TREG3A (0018 _H) | 7 6 5 4 3 2 1 0 | | Read/Write |
| TREG3B (0019 _H) | 7 6 5 4 3 2 1 0 | | Read only |
| TC3CR (001A _H) | 7 6 5 4 3 2 1 0 | SCAP TC3S TC3CK TC3M | (Initial value : *0*0 00*0) |
| TC3M | Timer/counter 3 operation mode set | 0 : Timer/event counter 1 : Capture | write only |
| TC3CK | Timer/counter 3 source clock select | 00 : Internal clock $fc / 2^{12}$ or $fs / 2^4$ [Hz] 01 : Internal clock $fc / 2^{10}$ or $fs / 2^2$ 10 : Internal clock $fc / 2^7$ 11 : External clock (TC3 pin input) | |
| TC3S | Timer/counter 3 start select | 0 : Stop & clear 1 : Start | |
| SCAP | Software capture control | 0 : - 1 : Software capture | |

*Note 1 : fc ; High-frequency clock [Hz] fs ; Low-frequency clock [Hz] * ; don't care*
Note 2 : Set the mode, the source clock and the edge 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 2-23. Timer Register 3 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 timer register 3A (TREG3A) are compared with the contents of the up-counter. If a match is found, a timer/counter 3 interrupt (INTTC3) is generated, and the counter is cleared. Counting up resumes after the up-counter is cleared. The current contents of the up-counter are loaded to the 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 | $f_c = 8 \text{ MHz}$ | $f_s = 32.768 \text{ kHz}$ | $f_c = 8 \text{ MHz}$ | $f_s = 32.768 \text{ kHz}$ |
| $f_c / 2^{12}$ or $f_s / 2^4$ [Hz] | $f_s / 2^4$ [Hz] | 512 μs | 488.28 μs | 131.1 ms | 125 ms |
| $f_c / 2^{10}$ or $f_s / 2^2$ | – | 128 μs | 122.07 μs | 32.8 ms | 31.25 ms |
| $f_c / 2^7$ | – | 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 $f_c/2^4$ [Hz] in NORMAL1/2 or IDLE1/2 mode, and $f_s/2^4$ [Hz] in SLOW or SLEEP mode. Two or more machine cycles are required for both the high and low levels of the pulse width.

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

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

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

(3) Capture Mode

The pulse width, period and duty of the TC3 pin input are measured in this mode, which can be used in decoding the remote control signals, etc. The counter is free running by the internal clock. On the rising (falling) edge of the TC3 pin input, the current contents of the counter is loaded into TREG3A, then the 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 TREG3B. In this case, counting continues. On the next rising (falling) edge of the TC3 pin input, the current contents of the counter is loaded in to 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 into 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.

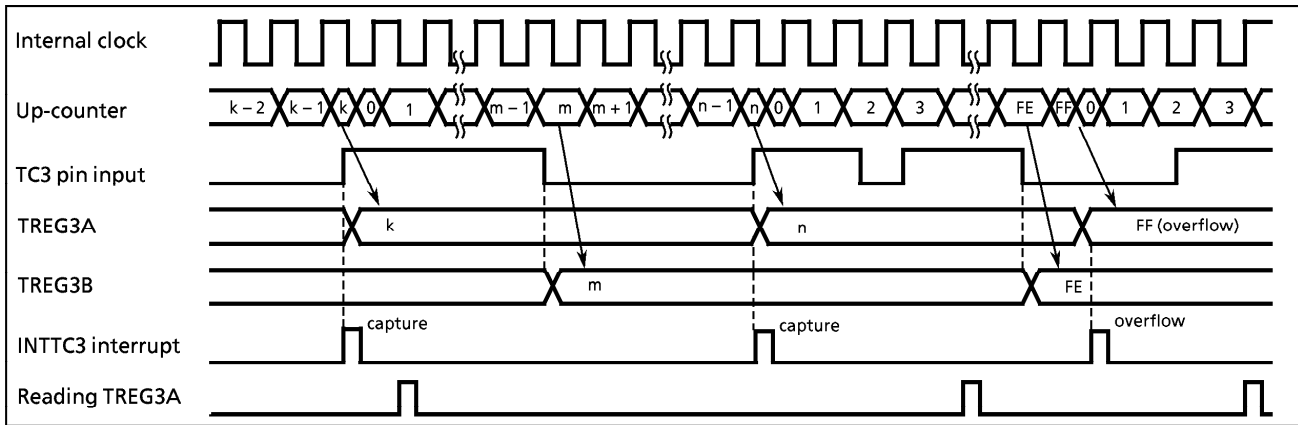


Figure 2-24. Timing Chart for Capture Mode (INT3ES = 0)

2.8 8-bit Timer/Counter (TC4)

2.8.1 Configuration

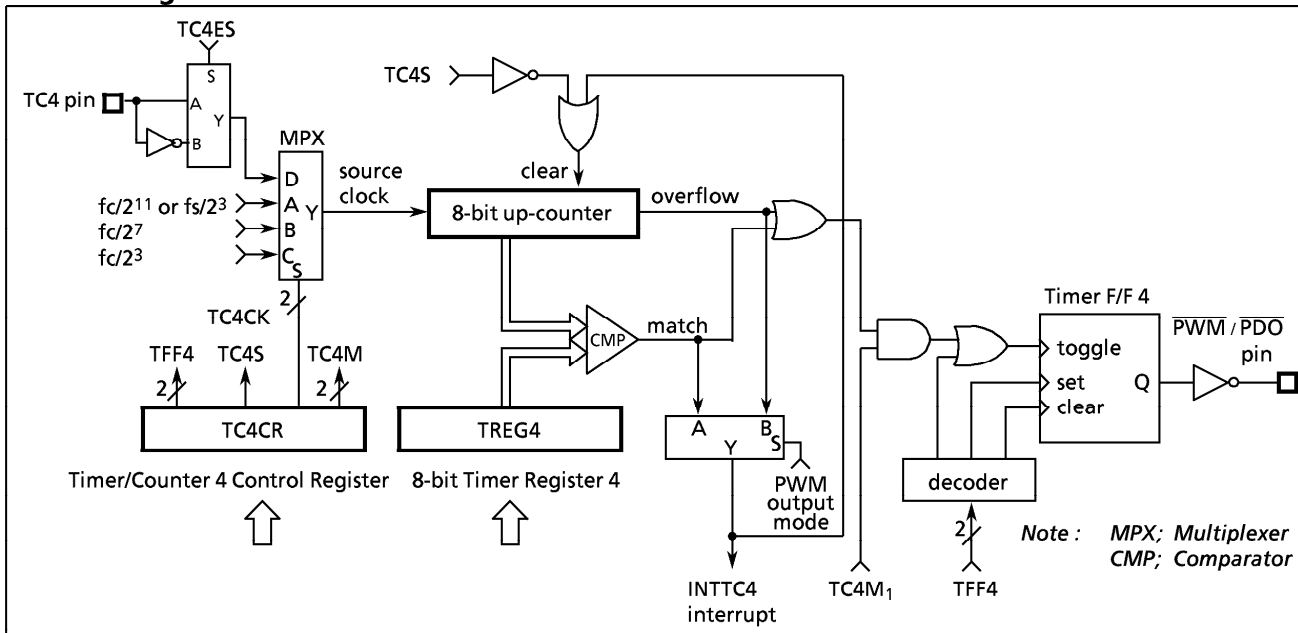


Figure 2-25. Timer/Counter 4

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.

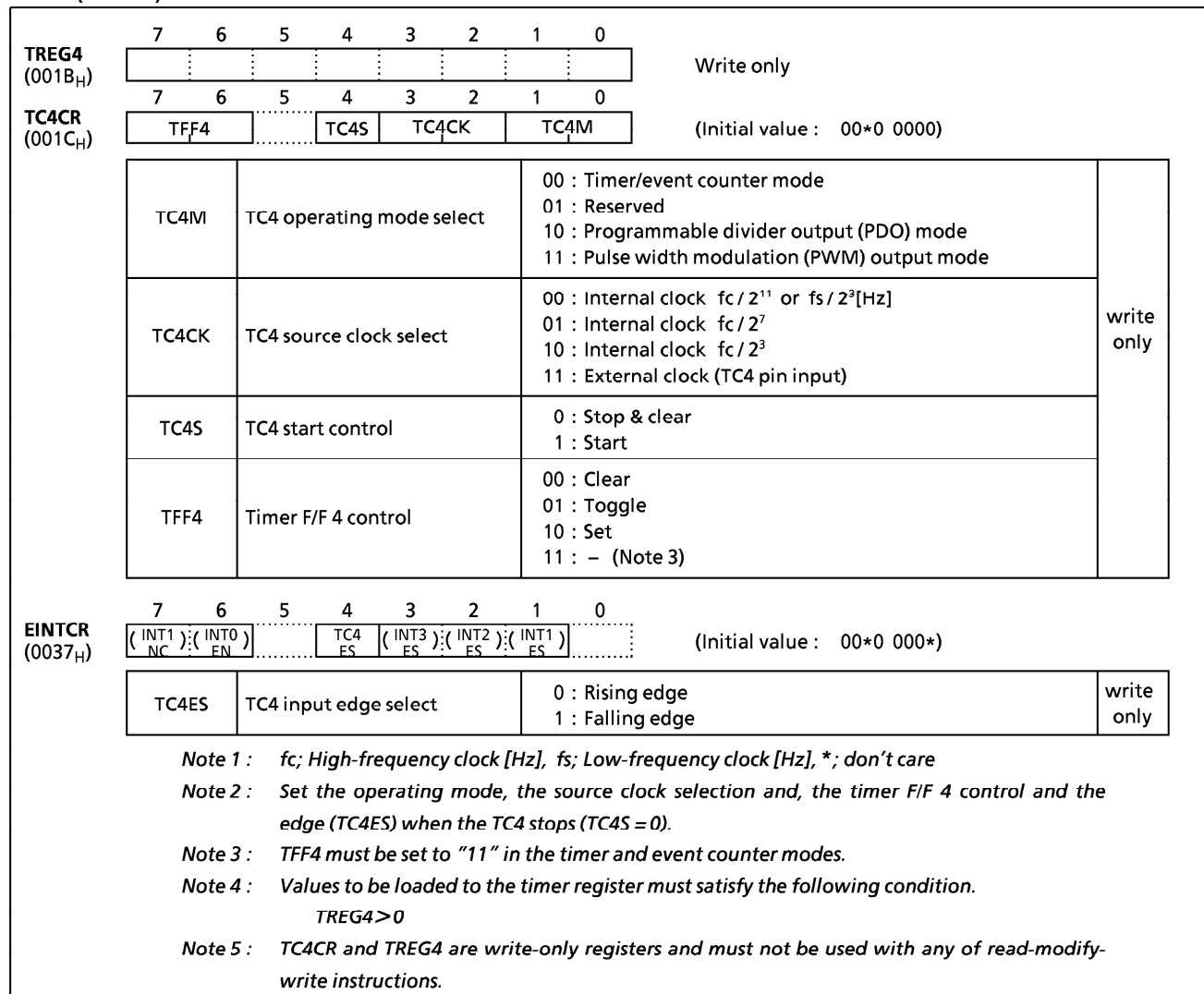


Figure 2-26. Timer Register 4 and TC4 Control Register

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 timer register 4 (TREG4) are compared with the contents of the up-counter. If a match is found, a timer/counter 4 interrupt (INTTC4) is generated and the counter is cleared. 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 | At $f_c = 8 \text{ MHz}$ | At $f_s = 32.768 \text{ kHz}$ | At $f_c = 8 \text{ MHz}$ | At $f_s = 32.768 \text{ kHz}$ |
| $f_c / 2^{11}$ or $f_s / 2^3$ [Hz] | $f_s / 2^3$ [Hz] | 256 μs | 244.14 μs | 65.536 ms | 62.5 ms |
| $f_c / 2^7$ | - | 16 μs | | 4.096 ms | |
| $f_c / 2^3$ | - | 1 μs | | 256 μs | |

(2) Event Counter Mode

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

(3) Programmable Divider Output (PDO) Mode

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

Example : Output a 1024 Hz pulse (At $f_c = 4.194304 \text{ MHz}$)

```
LD      (TC4CR), 00000010B      ; Initializes the TC4 mode, source clock and timer F/F 4.
LD      (TREG4), 20H            ; 1/1024 ÷ 27/fc = 20H
LD      (TC4CR), 00010010B      ; Starts TC4
```

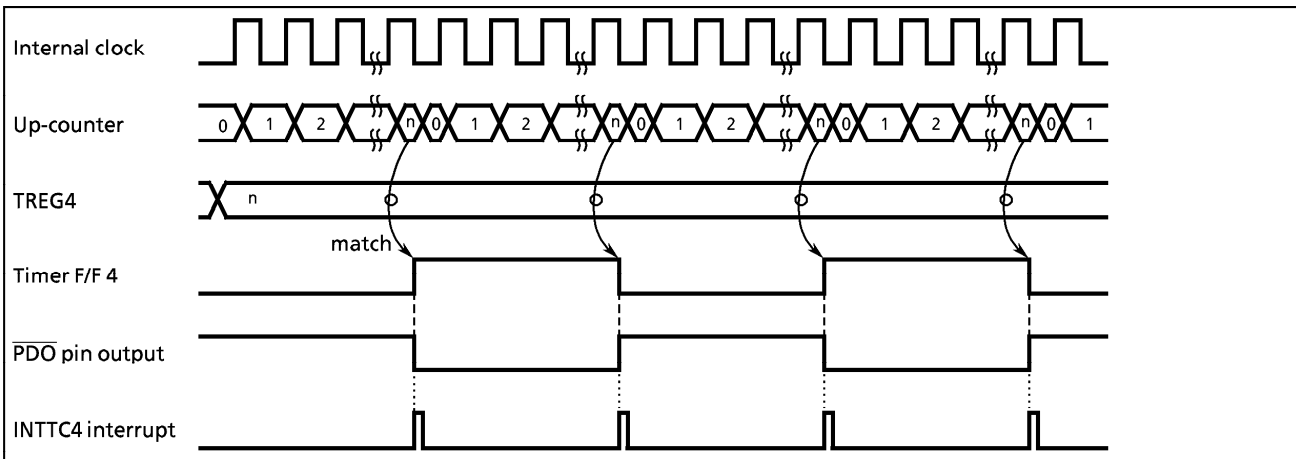


Figure 2-27. Timing Chart for PDO Mode

(4) Pulse Width Modulation (PWM) Output Mode

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

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

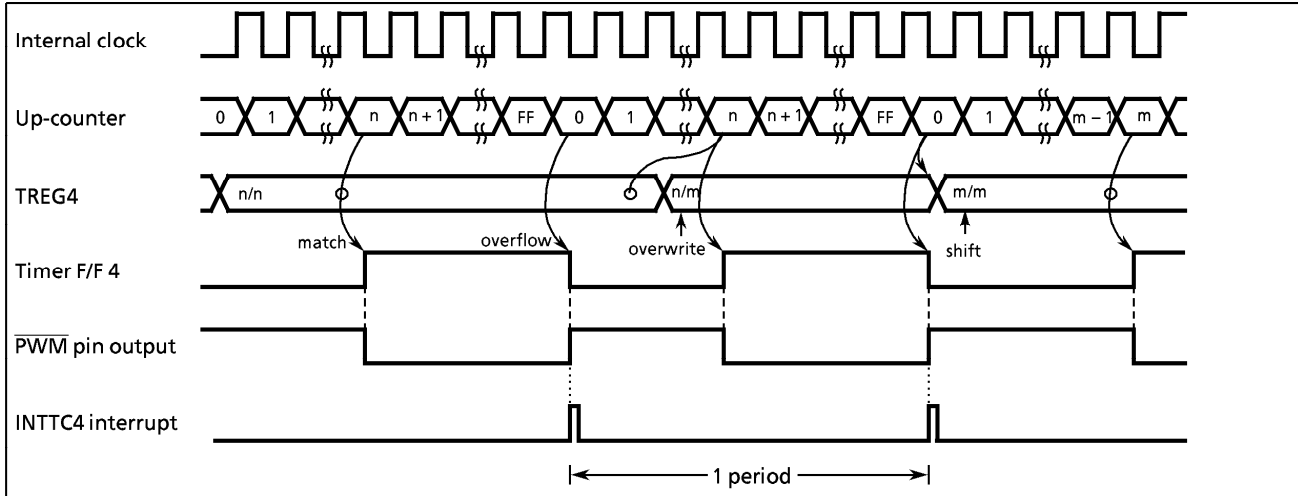


Figure 2-28. Timing Chart for PWM Mode

Table 2-7. PWM Output Mode

| Source clock | | Resolution | Repeat cycle | | | |
|-------------------------|------------------|--------------------------|-------------------------------|-------------------------------|-------------------------------|---------|
| NORMAL1/2, IDLE1/2 mode | SLOW, SLEEP mode | | At $f_c = 8 \text{ MHz}$ | At $f_s = 32.768 \text{ kHz}$ | | |
| DV7CK = 0 | DV7CK = 1 | At $f_c = 8 \text{ MHz}$ | At $f_s = 32.768 \text{ kHz}$ | At $f_c = 8 \text{ MHz}$ | At $f_s = 32.768 \text{ kHz}$ | |
| $f_c / 2^{11}$ [Hz] | $f_s / 2^3$ [Hz] | $f_s / 2^3$ [Hz] | 256 μs | 244.14 μs | 65.536 ms | 62.5 ms |
| $f_c / 2^7$ | $f_c / 2^7$ | - | 16 μs | | 4.096 ms | |
| $f_c / 2^3$ | $f_c / 2^3$ | - | 1 μs | | 256 μs | |

2.9 Serial Interface (SIO)

The 87CC70/H70/K70A/M70A each have a clocked-synchronous 8-bit serial interface (SIO). The SIO has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

The SIO is connected to external devices via pins P32 ($\overline{\text{SCK}}$), P33 (SI) and P34 (SO). The serial interface pins are also used as port P3. When used as serial interface pins, the P34-P32 output latches should be set to "1". In the transmit mode, pin P33 is used as a normal I/O port, and in the receive mode, pin P34 is used as a normal I/O port.

2.9.1 Configuration

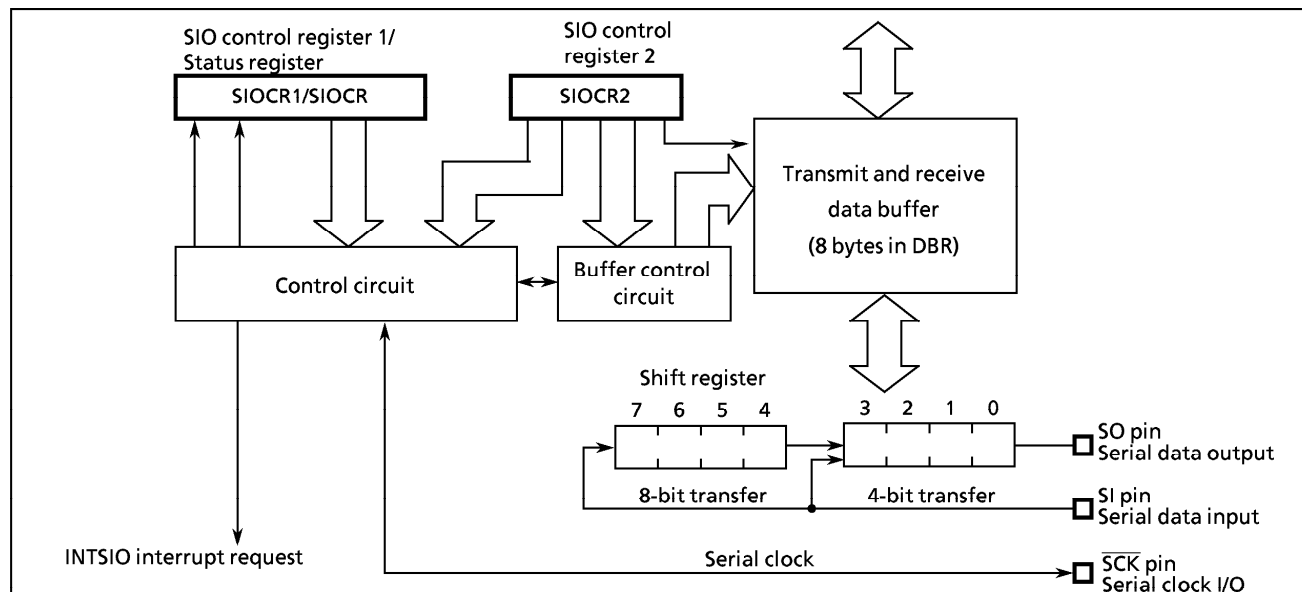


Figure 2-29. Serial Interface

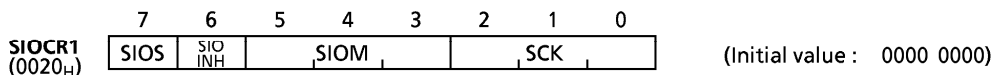
2.9.2 Control

The SIO is controlled by two SIO control registers (SIOCR1 and SIOCR2). The serial interface status can be determined by reading a SIO status register (SIOSR).

The transmit and receive data buffer is controlled by the SIOCR2. The data buffer is assigned to addresses 0FF0_H - 0FF7_H in the DBR (data buffer registers) 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 (INTSIO) 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 SIOCR2.

SIO Control Register 1



| | | | |
|-----------------------------------|------------------------------|---|------------|
| SIOS | Indicate transfer start/stop | 0 : Stop 1 : Start | write only |
| $\overline{\text{SIO}}\text{INH}$ | 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 or receive mode (Note 1) 110 : 4-bit receive mode **1 : reserved | |
| SCK | Serial clock select | 000 : Internal clock $f_c / 2^{13}$ or $f_s / 2^5$ [Hz] 001 : Internal clock $f_c / 2^8$ 010 : Internal clock $f_c / 2^9$ 011 : Internal clock $f_c / 2^5$ 111 : External clock (input from $\overline{\text{SCK}}$ pin) <div style="float: right; margin-left: 10px;"> } (Output on $\overline{\text{SCK}}$ pin) </div> | |

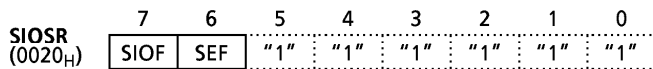
Note 1 : Transmit/receive or receive mode are selected with ERM in $\overline{\text{SIOC}}\text{R2}$.

Note 2 : Set SIOS to "0" and $\overline{\text{SIO}}\text{INH}$ to "1" when setting the transfer mode or serial clock.

Note 3 : f_c ; High-frequency clock [Hz], f_s ; Low-frequency clock [Hz]

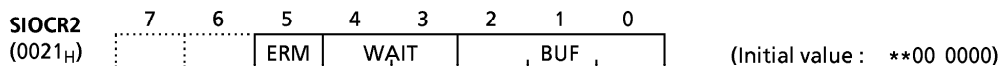
Note 4 : $\overline{\text{SIOC}}\text{R1}$ is write-only registers, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO Status Register



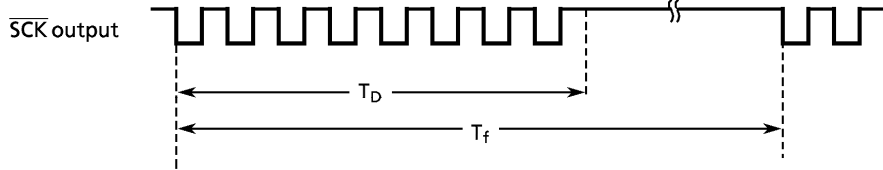
| | | | |
|------|--|--|-----------|
| SIOF | Serial transfer operating status monitor | 0 : Transfer terminated 1 : Transfer in process | read only |
| SEF | Shift operating status monitor | 0 : Shift operation terminated 1 : Shift operation in process | |

SIO Control Register 2



| | | | |
|------|-------------------------------------|--|------------|
| ERM | 8-bit transmit/receive mode control | 0 : Transmit/receive mode (Always set to "0" except in the 8-bit receive mode.) 1 : Receive mode | write only |
| WAIT | Wait control | 00 : $T_f = T_D$ (no waits) 01 : $T_f = 2T_D$ 10 : $T_f = 4T_D$ 11 : $T_f = 8T_D$ (waits) | |
| BUF | Number of transfer words | Buffer addresses used 000 : 1-word transfer OFF0 _H 001 : 2-word transfer OFF0 - OFF1 _H 010 : 3-word transfer OFF0 - OFF2 _H 011 : 4-word transfer OFF0 - OFF3 _H 100 : 5-word transfer OFF0 - OFF4 _H 101 : 6-word transfer OFF0 - OFF5 _H 110 : 7-word transfer OFF0 - OFF6 _H 111 : 8-word transfer OFF0 - OFF7 _H | |

- Note 1 : * ; don't care
- Note 2 : The 8-bit receive mode can be set by setting ERM to "1" after setting SIOM (bits 5 to 3 in SIOCR1) to "100".
- Note 3 : WAIT is effective only in the 8-bit transmit/receive mode and 8-bit receive mode. Always set to "00" except in these modes.
- Note 4 : T_f ; frame time (time required to transfer 1 word of data), T_D ; data transfer time.



- Note 5 : In the 4-bit transfers the lower 4 bits of each buffer are used. Zeros (0) are stored to the upper 4 bits when receiving.
- Note 6 : Transmission starts from the lowest buffer address. Received data are stored starting from the lowest buffer address.
- Note 7 : The value to be loaded to BUF is held after transfer is completed.
- Note 8 : SIOCR2 is write-only registers, which cannot access any of in read-modify-write instruction such as bit operate, etc.

Figure 2-30. SIO Control Registers and Status Register

(1) Serial Clock

a. Clock Source

SCK (bits 2 - 0 in SIOCR1) is able to select the following:

① Internal Clock

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

When data writing (in the transmit mode) or reading (in the receive mode and 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 2-8. Serial Clock Rate

| Serial clock | | | Maximum transfer rate | |
|-------------------------|------------------|------------------|-------------------------------|----------|
| NORMAL1/2, IDLE1/2 mode | | SLOW, SLEEP mode | At $f_c = 8 \text{ MHz}$ | |
| DV7CK = 0 | DV7CK = 1 | | At $f_s = 32.768 \text{ kHz}$ | |
| $f_c / 2^{13}$ [Hz] | $f_s / 2^5$ [Hz] | $f_s / 2^5$ [Hz] | 0.977 Kbit/s | 1 Kbit/s |
| $f_c / 2^8$ | $f_c / 2^8$ | - | 31.2 | |
| $f_c / 2^6$ | $f_c / 2^6$ | - | 125 | |
| $f_c / 2^5$ | $f_c / 2^5$ | - | 250 | |

Note : 1 Kbit = 1024bit

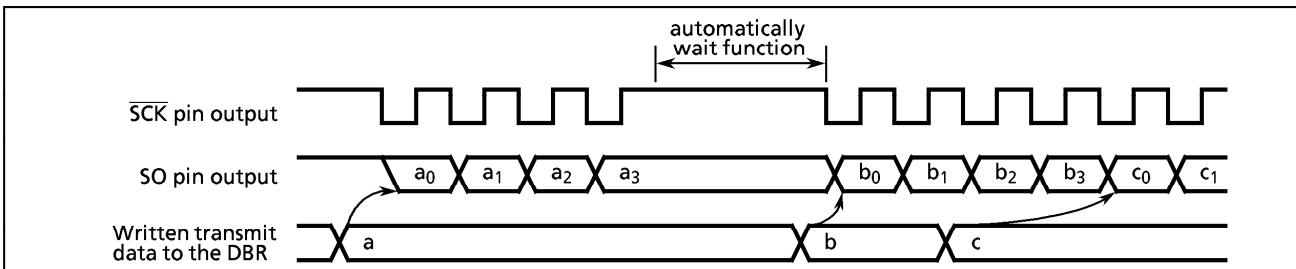
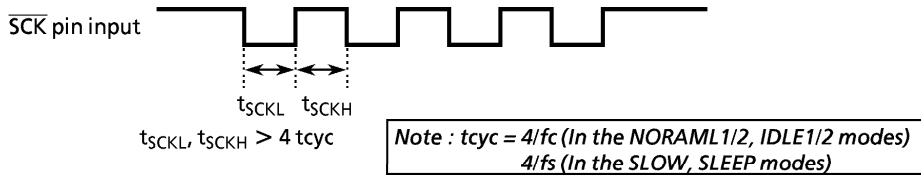


Figure 2-31. Clock Source (Internal Clock)

② External Clock

An external clock connected to the \overline{SCK} pin is used as the serial clock. In this case, the P32 (\overline{SCK}) 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 $f_c = 8$ MHz).



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{SCK} pin input/output).

② Trailing Edge

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

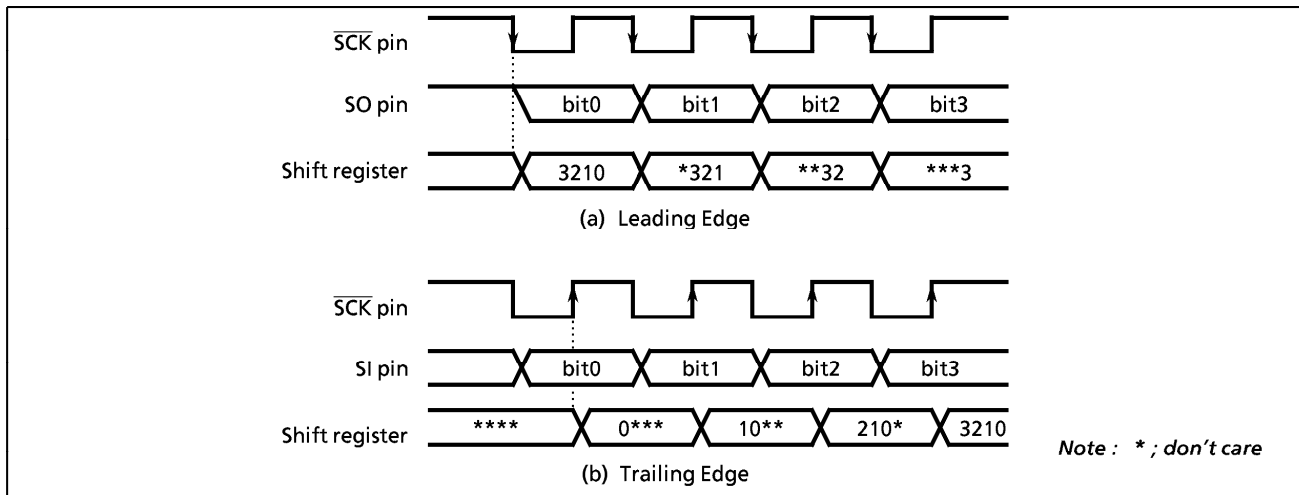


Figure 2-32. Shift Edge

(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 SIOCR2. 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.

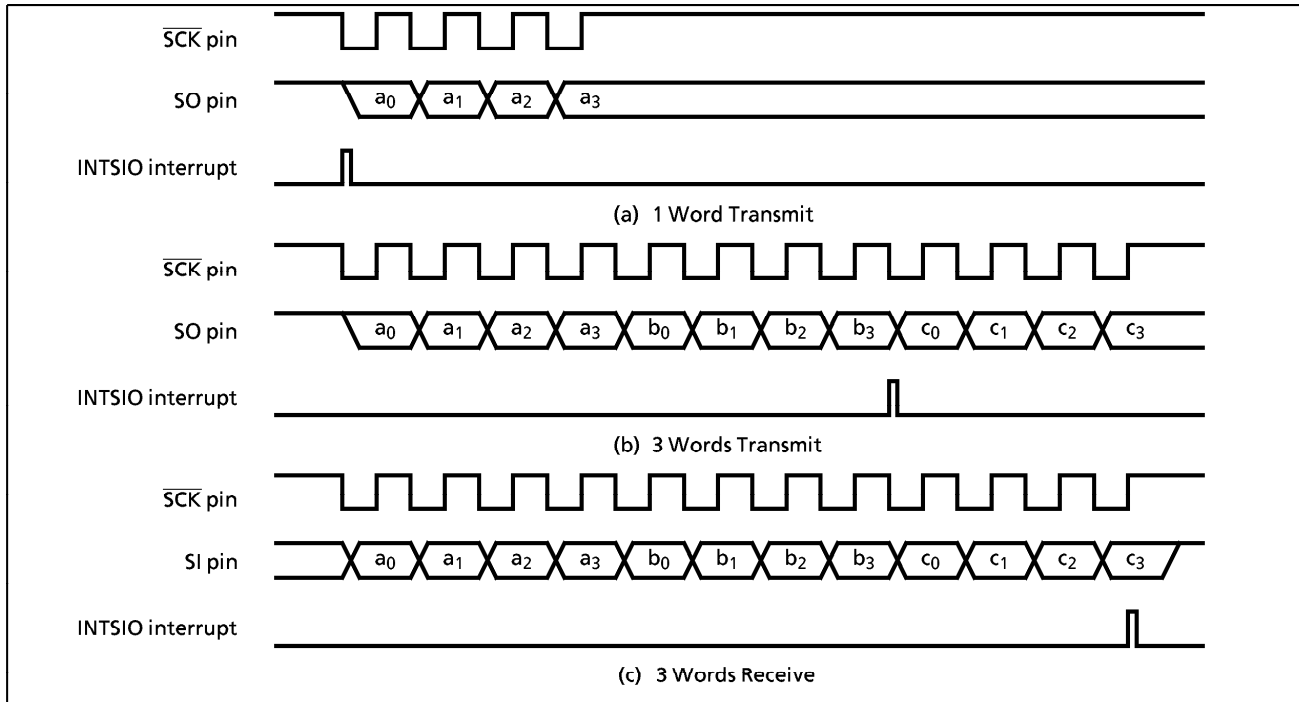


Figure 2-33. Number of Bits to Transfer (Example : 4-bit serial transfer)

2.9.3 Transfer Mode

SIOM (bits 5 - 3 in SIOCR1) and ERM (bit 5 in SIOCR2) are used to select the transmit, receive, or transmit/receive mode.

(1) 4-bit and 8-bit Transmit Modes

In these modes, SIOCR1 and SIOCR2 are 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 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 : The wait is 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{SCK} .

The transmission is ended by clearing SIOS to "0" at the time that the final bit of the data being shifted out has been transferred. That the transmission has ended can be determined from the status of SIOF (bit 7 in SIOSR) because SIOF is cleared to "0" when a transfer is completed.

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.

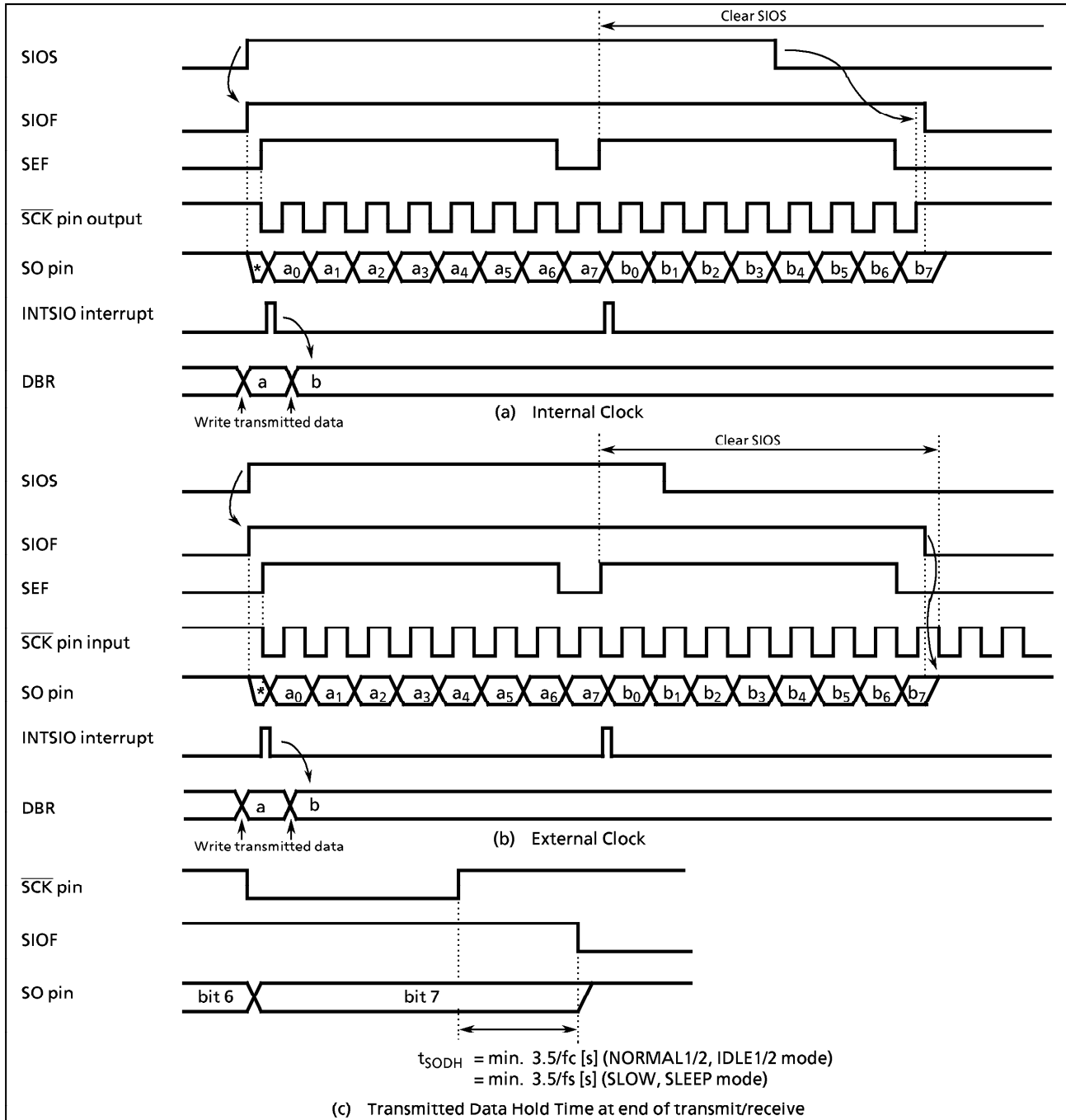


Figure 2-34. Transmit Mode (Example : 8 bit 1 Word Transmit)

Example : The way of the transfer stopped (8-bit Transmit mode, External clock).

```

STEST1 :   TEST      (SIOCR).SEF      ; If SEF = 1 then loop
           JRS       F,STEST1
STEST2 :   TEST      (P3).2          ; If SCK = 0 then loop
           JRS       T,STEST2
           LD        (SIOCR1),00000111B ; SIOS ← 0
    
```

Note : When to clear SIOS to "0", that it is also necessary to check SEF = 1 and SCK = 0.

(2) 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 is then transferred to the shift register via the SI pin 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 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 : The wait is 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.

Clear SIOS to "0" to end receiving. 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.

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.

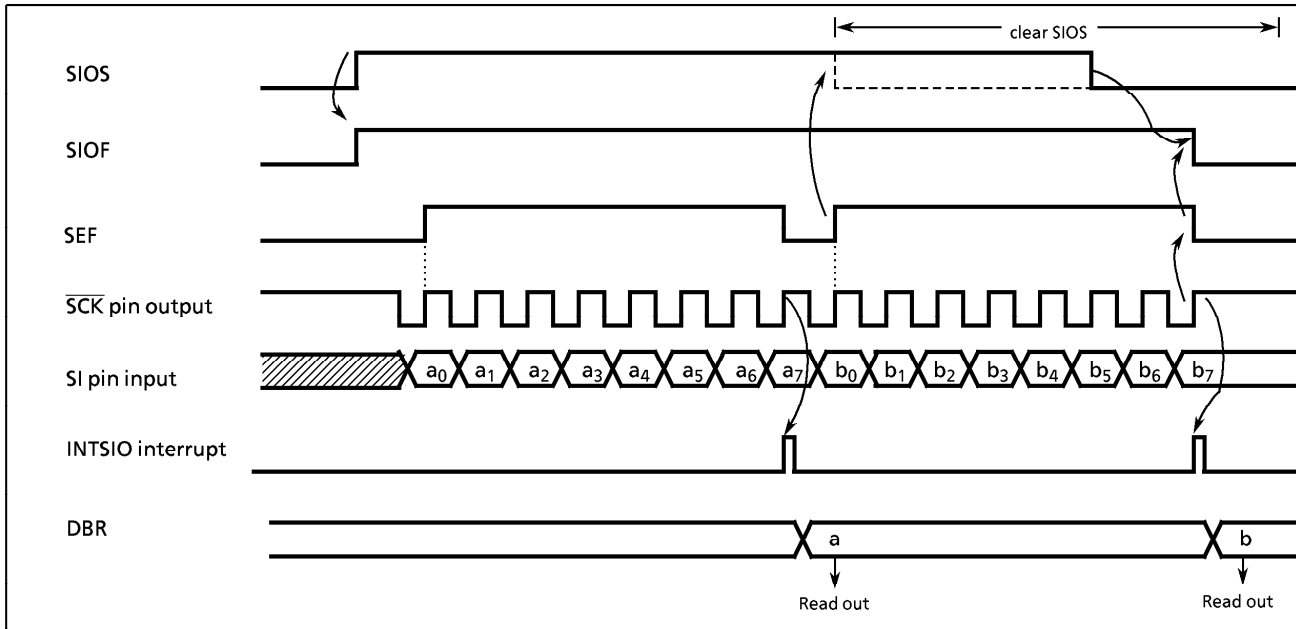


Figure 2-35. Receive Mode (Example : 8-bit, 1 word, internal clock)

(3) 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 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.

Note : The wait is also canceled by writing to a DBR not being used as a transmit data buffer registers; 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 from the SO pin holds final bit of the last data until falling edge of the SCK.

Clear SIOS to "0" to enable the transmit mode. 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.

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.

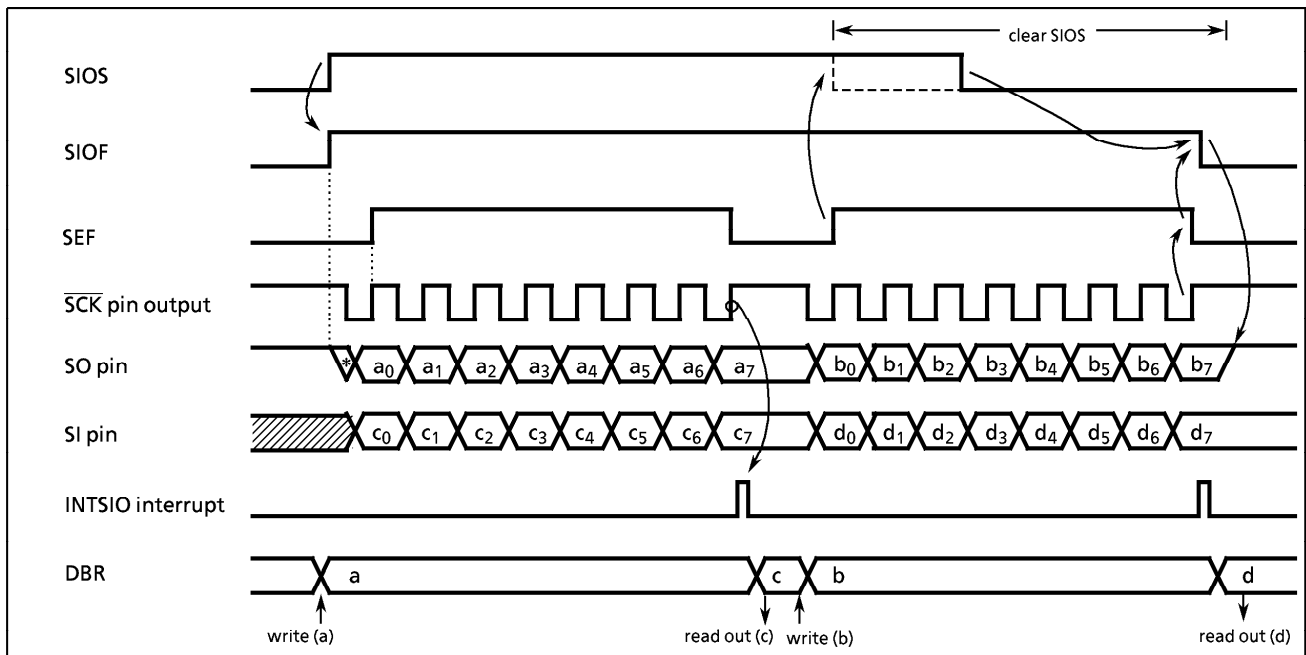


Figure 2-36. Transmit/Receive Mode (Example : 8-bit, 1word, internal clock)

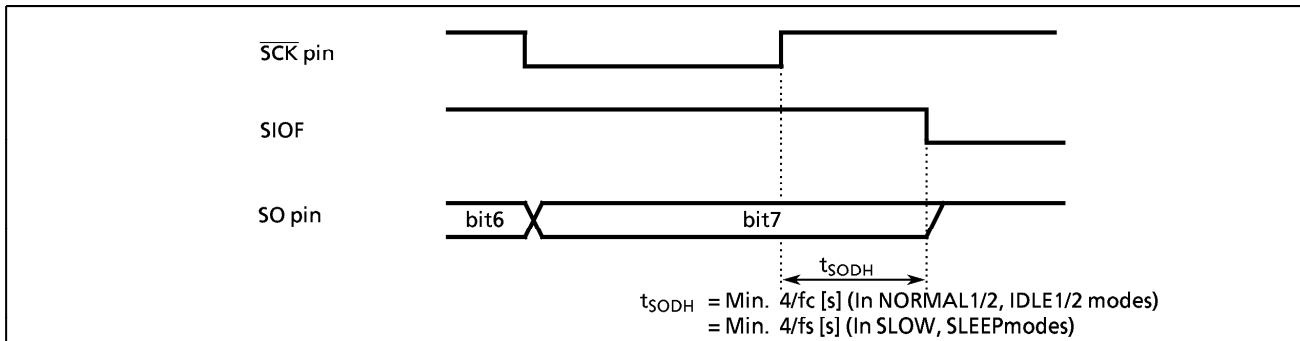


Figure 2-37. Transmitted Data Hold Time at end of transmit/receive

2.10 8-bit High-speed Serial Output (HSO)

The 87CC70/H70/K70A/M70A each have a clock-synchronous 8-bit serial output (HSO). The HSO has a 1-byte transmit data buffer register (HSODR). The HSODR is assigned to address 0FF8H in the DBR area. The HSO is connected to the external devices via pins P35 ($\overline{\text{HSCK}}$) and P37 (HSO). These pins are also used as the port P3. When used as pins $\overline{\text{HSCK}}$ /HSO, the P35/P37 output latches should be set to "1".

2.10.1 Configuration

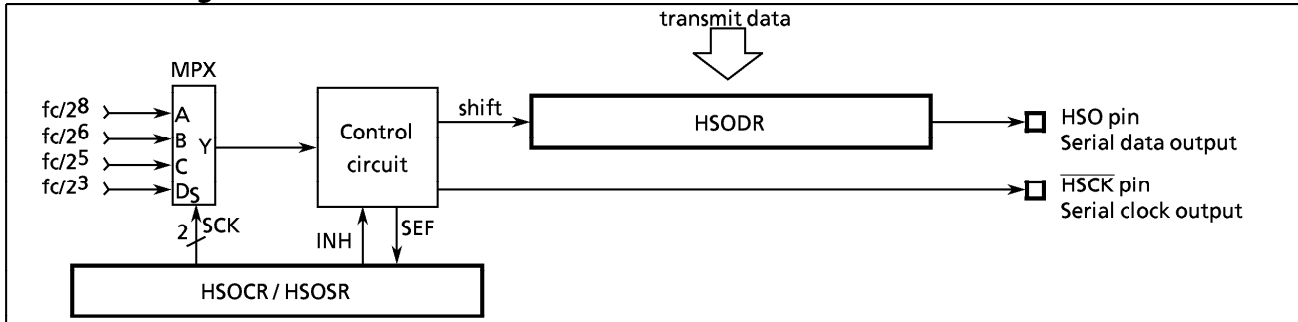


Figure 2-38. High-speed Serial Output

2.10.2 Control

The HSO is controlled by a HSO control register (HSOCR). The transfer status can be determined by reading a HSO status register (HSOSR).

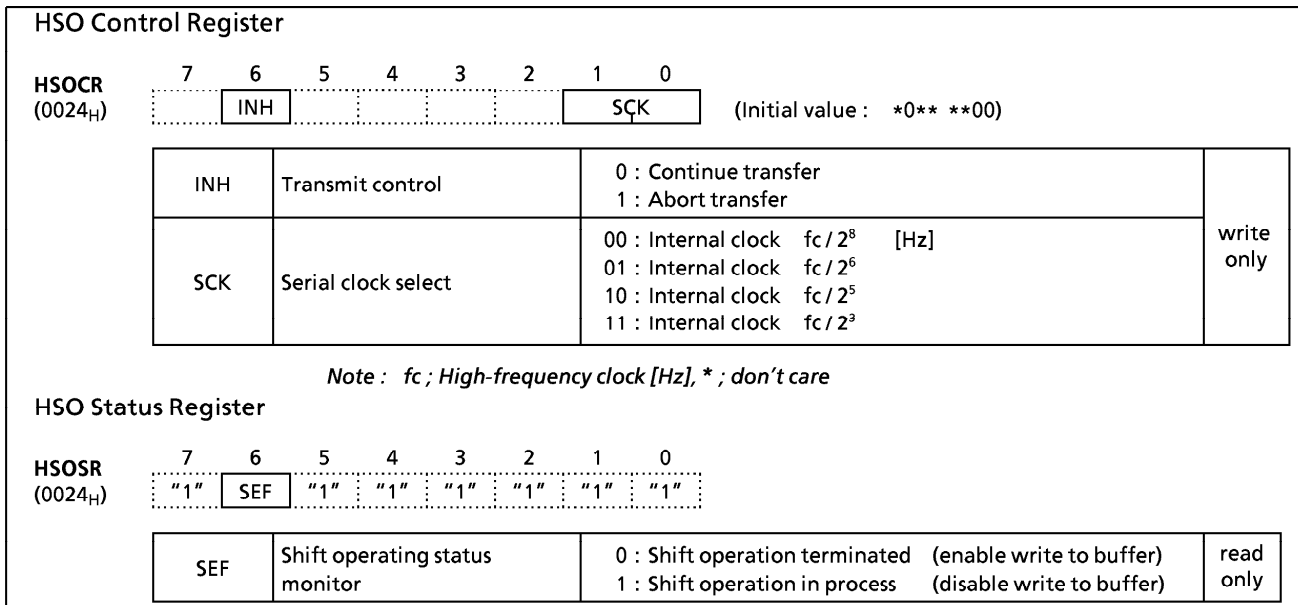


Figure 2-39. HSO Control Register and Status Register

2.10.3 Transmit Operations

SCK (bits 1 and 0 in HSOCR) is used to select the transfer rate. Transmission is started by writing one byte of data to the HSODR. The transmit data are output sequentially to the HSO pin in synchronized with the falling edges of the serial clock, starting with the least significant bit (LSB). Writing to the buffer is disabled by the hardware during data transfers. The shift register is empty after one byte of data has been transferred, so writing to the buffer is again enabled at that point. SEF (bit 6 in HSOSR) is set to "1" during transfers (write to buffer disabled) and is cleared to "0" when a transfer is completed (write to buffer enabled); therefore, whether or not a transfer has been completed can be confirmed with a program that reads SEF. The $\overline{\text{HSCK}}$ pin is raised to "high" at the start and end of transfers.

Note : To continue a transfer without sensing SEF, write the next data to be transferred after 9 cycles (11 cycles when $f_c/2^3$ [Hz] is selected only) at the transfer rate selected with SCK after writing to the buffer.

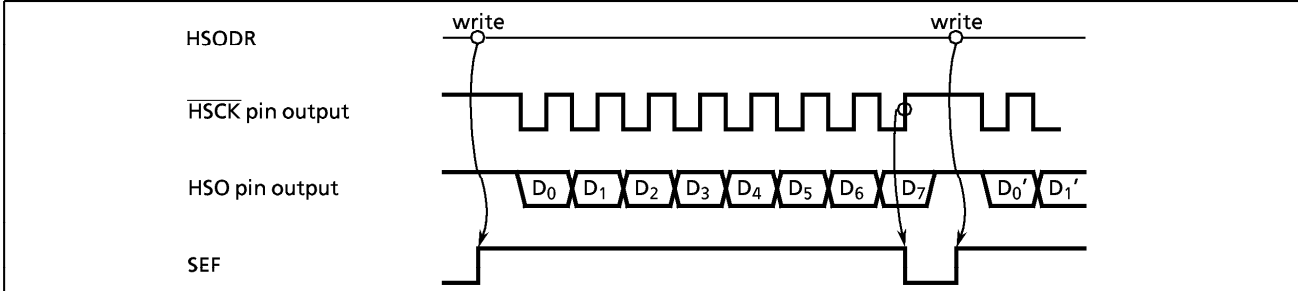


Figure 2-40. High-speed Serial Output Timing Chart

2.11 6-bit A/D Conversion (Comparator) Inputs

The comparator input is an analog input to discriminate key input or AFC (Auto Frequency Control) signal input, etc. The analog input voltage level (pins CIN5 - CIN0) can be detected as 64-stage by setting reference voltage.

The comparator input pins CIN5-CIN0 can also be used as ports P46, P47 and P53 - P50.

When used as a comparator input, the output latch should be set to "1". Note that the built-in pull-down resistances of the pins P47 and P46 must be cut off by clearing EPD (bits 7 and 6 in P4CR) to "0" when used as pins CIN4 and CIN5.

2.11.1 Configuration

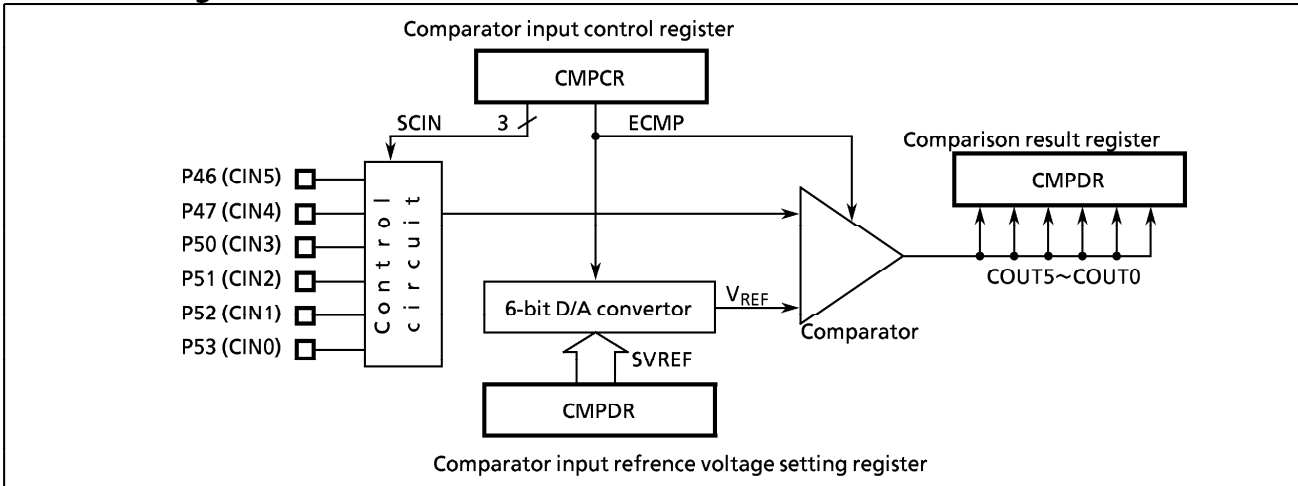


Figure 2-41. 6-bit A/D Conversion (Comparator) Input

2.11.2 Control

A/D conversion (comparator) inputs are controlled by a comparator input control register (CMPCR) and a comparator input data register (CMPDR). The CMPDR contains a reference voltage setting register (write-only) and a comparison result register (read-only).

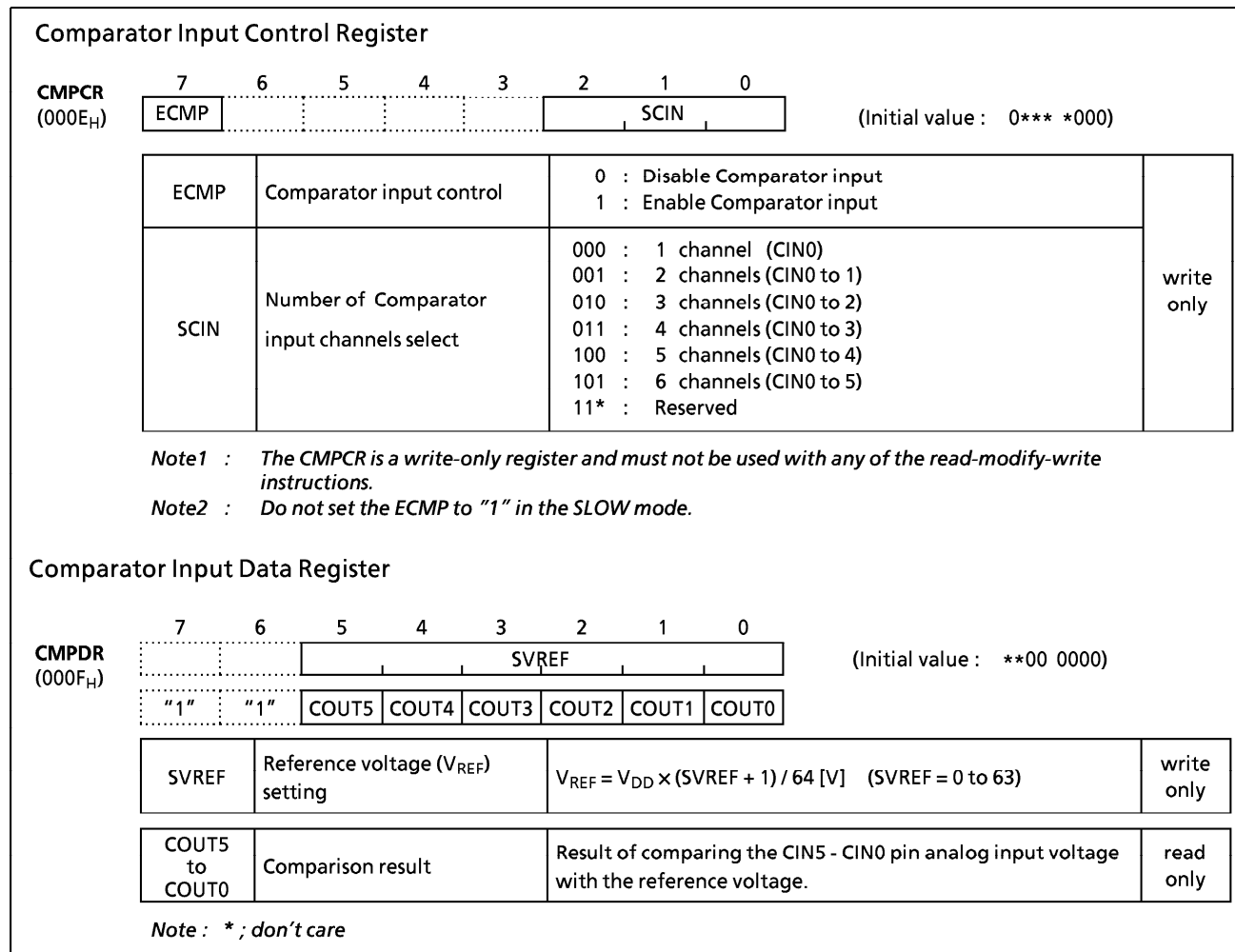


Figure 2-42. Comparator Input Control Register and Data Register

2.11.3 Function

Reference voltage (V_{REF}) is set with SVREF (bits 5 - 0 in CMPDR).

The number of comparator input channels is selected with SCIN (bits 2 - 0 in CMPPCR). Sequential comparison of the selected number of channels is started by setting ECMP (bit 7 in CMPPCR) to "1".

The comparison of one channel requires two machine cycles; therefore, the comparison result register (COUT5 - COUT0) should be read out at an interval equal to [number of channels × 2 machine cycles] after setting the reference voltage (V_{REF}). COUT5 - COUT0 are set to "1" if the input voltage (pins CIN5 - CIN0) is higher than the reference voltage (V_{REF}); otherwise those are cleared to "0".

Note 1 : When entering STOP and SLOW modes, ECMP is automatically cleared and SCIN/SVREF are held. And, COUT5 - COUT0 are always set to "1".

If a comparator is operated at a STOP or SLOW mode, Typ. 1.2 mA (V_{DD} = 5.0 V) of the supply current is used.

Note 2 : Any pins specified for comparator input with SCIN can no longer be used for normal digital input or key scan input, and are read out as "0".

Note 3 : COUT5 - COUT0 are read out as "1" when not used as a comparator input. For example, bits 7 to 3 in CMPDR are always read out as "1" when SCIN = 010_B.

Example : Compares the CIN3-CIN0 inputs with $V_{REF} = 2.5\text{ V}$ (At $V_{DD} = 5\text{ V}$).

LD (P5), 11111111B ; Sets port P5 output latches to "1".

LD (CMPDR), 00011111B ; Sets $V_{REF} = 2.5\text{ V}$

LD (CMPCR), 10000011B ; Sets SCIN to 4 channels and Enables comparator input

⋮

; 4ch × 2 machine cycles -2
= 6 machine cycles wait.

LD A, (CMPDR) ; Reads CMPDR (COUT0~COUT3).

Table 2-9. Reference Voltage (at $V_{DD} = 5\text{ V}$)

| SVREF | | | | | | | $V_{REF} [V]$ |
|-------|---|---|---|---|---|-------|---------------|
| 5 | 4 | 3 | 2 | 1 | 0 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0.078 | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0.156 | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0.234 | |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | |
| 1 | 1 | 1 | 1 | 0 | 1 | 4.844 | |
| 1 | 1 | 1 | 1 | 1 | 0 | 4.922 | |
| 1 | 1 | 1 | 1 | 1 | 1 | 5.000 | |

2.12 Vacuum Fluorescent Tube Driver Circuit

The 87CC70/H70/K70A/M70A each have 32 high-breakdown voltage output buffers that directly drive the Vacuum Fluorescent Tube (VFT) and its control circuit.

2.12.1 Configuration

Figure 2-43 shows the VFT driver circuitry configuration.

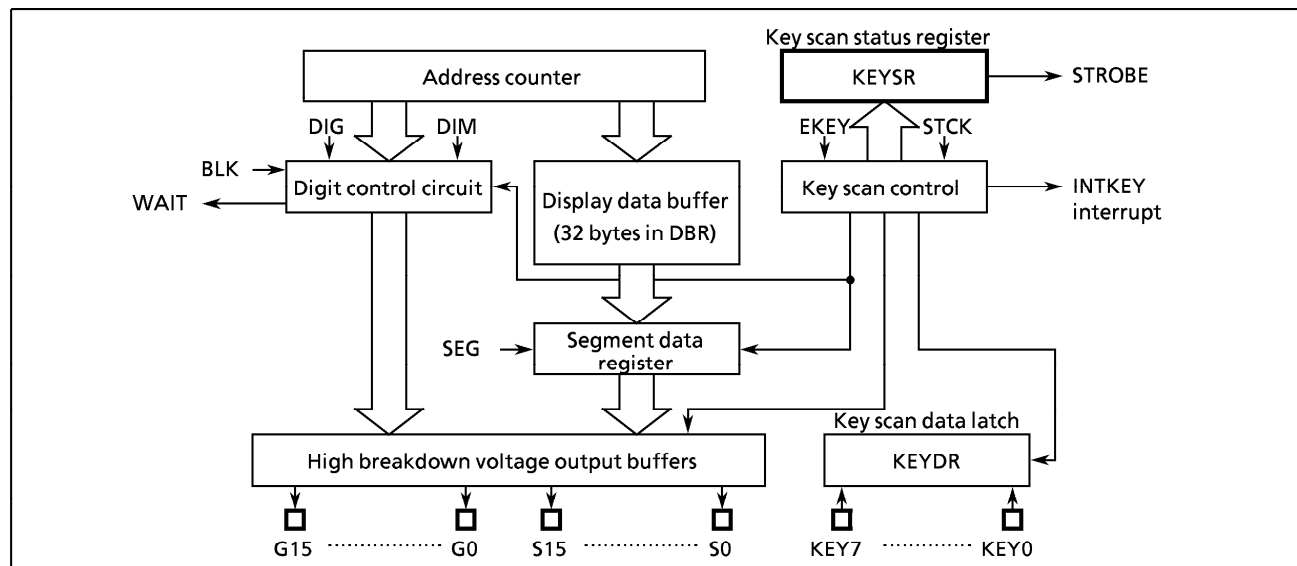


Figure 2-43. VFT Driver Circuitry

2.12.2 Function

- ① 32 high-breakdown voltage output buffers.
 - 16 digit output buffers (G0 to G15)
 - 16 segment output buffers (S0 to S15)

The VKK pin is provided for the VFT drive power supply.
- ② 8 - 16 segments × 1 - 16 digits can be selected by program using the dynamic display.
- ③ Pins not used for the VFT driver can be used as general-purpose ports.
- ④ The display data is automatically transferred to the high-breakdown voltage output buffers from the display data buffer memory (32 bytes in the DBR).
- ⑤ Brightness level adjustment (8 levels) is available with a dimmer function.
- ⑥ An automatic key scan function is available. It is possible to use the segment outputs for key strobe outputs.

2.12.3 Control

The VFT driver circuit is controlled by two VFT control registers (VFTCR1 and VFTCR2). The VFT operational status can be determined by reading a VFT status register (VFISR). Switching from NORMAL 1 or NORMAL 2 mode to SLOW or STOP mode, BLK is set to "1" and EKEY is cleared to "0"; therefore, the VFT driver circuit is placed into the blanking state. The contents of the VFT control registers (except BLK, EKEY) are held. Also segment outputs and digit outputs are cleared to "0"; therefore, the ports P6, P7, P8 and P9 function as normal I/O ports with pull-down.

| VFT Control Register 1 | | 7 6 5 4 3 2 1 0 | | | | | | | | (Initial value : 1000 0111) |
|--------------------------------|-------------------------------------|--|------|------|--|--|--|-----|--|-----------------------------|
| VFTCR1 (0028 _H) | | BLK | EKEY | STCK | | | | SEG | | |
| BLK | VFT display control | 0 : Enable 1 : Disable (blanking) | | | | | | | | write only |
| EKEY | Key scan function control | 0 : Disable 1 : Enable | | | | | | | | |
| STCK | Key scan strobe source clock select | 00 : $f_c / 2^{12}$ or $f_s / 2^5$ [Hz] (once every 4 t_{SEG}) 01 : $f_c / 2^{13}$ or $f_s / 2^6$ (once every 8 t_{SEG}) 10 : $f_c / 2^{14}$ or $f_s / 2^7$ (once every 16 t_{SEG}) 11 : $f_c / 2^{15}$ or $f_s / 2^8$ (once every 32 t_{SEG}) | | | | | | | | |
| SEG | Number of segments select | 0111 : 8-segment display mode (outputs S0 to S7) 1000 : 9-segment display mode (outputs S0 to S8) 1001 : 10-segment display mode (outputs S0 to S9) 1010 : 11-segment display mode (outputs S0 to S10) 1011 : 12-segment display mode (outputs S0 to S11) 1100 : 13-segment display mode (outputs S0 to S12) 1101 : 14-segment display mode (outputs S0 to S13) 1110 : 15-segment display mode (outputs S0 to S14) 1111 : 16-segment display mode (outputs S0 to S15) 0**0 : Reserved 0*0* : Reserved 00** : Reserved | | | | | | | | |

Note 1 : f_c ; High-frequency clock [Hz],
 f_s ; Low-frequency clock [Hz]
 t_{SEG} ; During NORMAL 1 or NORMAL 2 (DV7CK = 0) $2^{10} / f_c$ [s]
 During NORMAL 2 mode (DV7CK = 1) $2^3 / f_s$

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

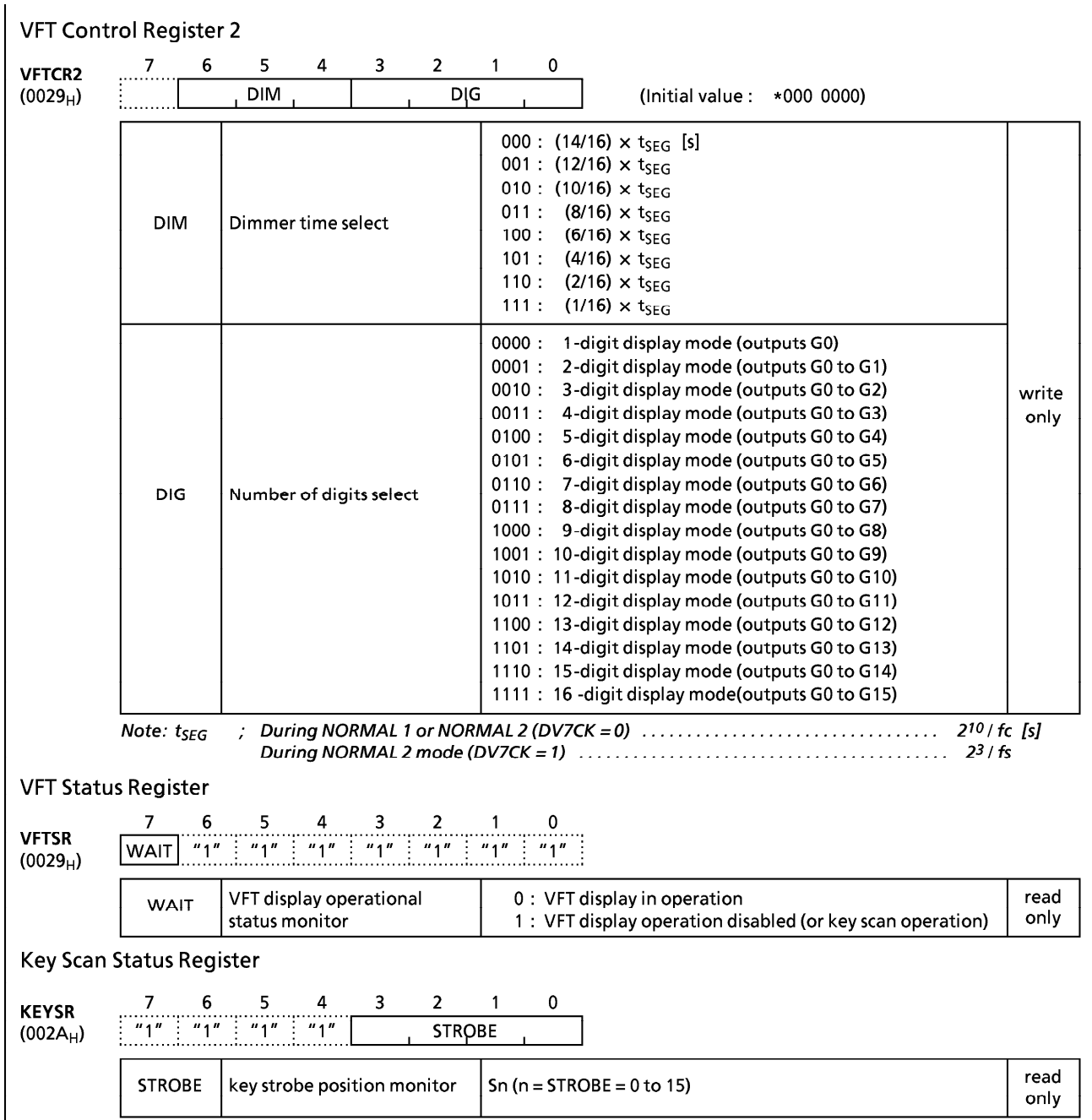


Figure 2-44. VFT Control Registers/Status Registers

(1) Display Mode Setting

The number of segments is selected with SEG (bits 3-0 in VFTCR1) and the number of digits is selected with DIG (bits 3-0 in VFTCR2) when BLK (bit 7 in VFTCR1) = 1 and EKEY (bit 6 in VFTCR1) = 0. The dimmer time (digit output time) is selected with DIM (bits 6-4 in VFTCR2).

Example : Sets display mode (8-seg. × 16-dig., (14/16) × t_{SEG})

LDW (VFTCR1), 0F87H ; BLK←1, EKEY←0, STCK←00, SEG←0111, DIM←000, DIG←1111

(2) Display Data Setting

Normally, the conversion of data to VFT display data is performed by instructions. The converted data which stored to the display data buffer memory (addresses 0F80_H - 0F9F_H in the DBR) is

automatically transferred to the VFT driver circuit and output to the high-breakdown voltage output buffers. Thus, display patterns can be varied by merely changing the data in the data buffer memory.

The VFT segments (dots) and display data area bits have a one-to-one correspondence. A "1" bit lights the corresponding segment and a "0" bit turns off the corresponding segment. Areas of the display data buffer not used for display purposes can be used as normal data memory. The display data buffer is assigned to the DBR area shown in Figure 2-45 (a) but, in the 8-segment display mode only, the assignment is as shown in Figure 2-45 (b) for more effective utilization of data memory.

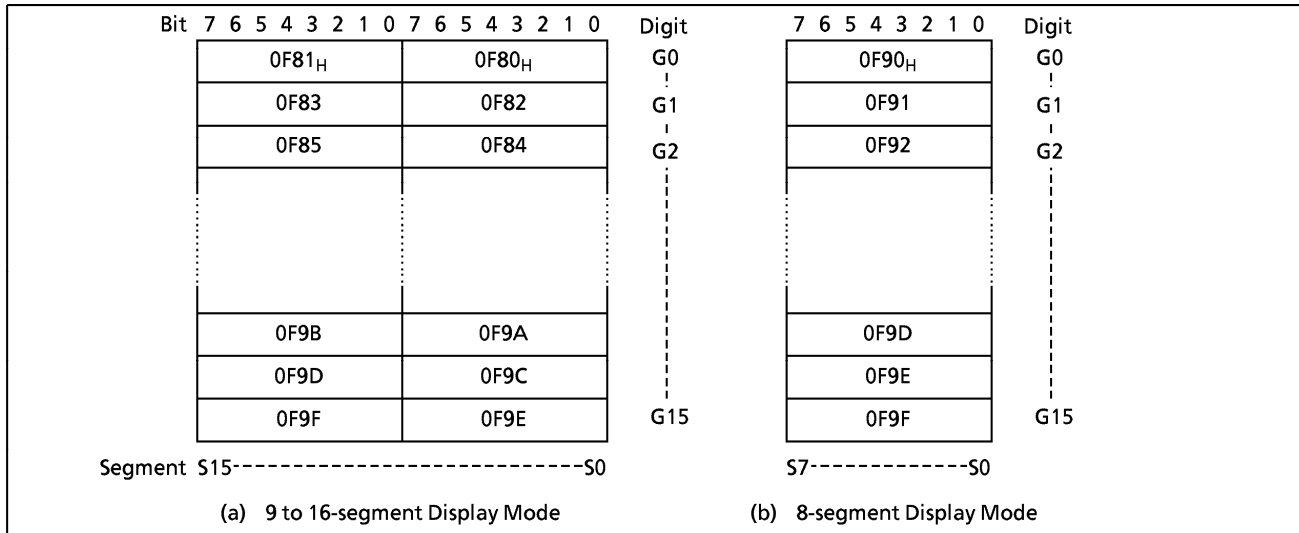


Figure 2-45. VFT Display Data Buffer Memory (DBR)

2.12.4 Display Operation

After setting the display mode and storing the display data, VFT display is started by clearing BLK to "0". Figure 2-46 shows the VFT drive waveform.

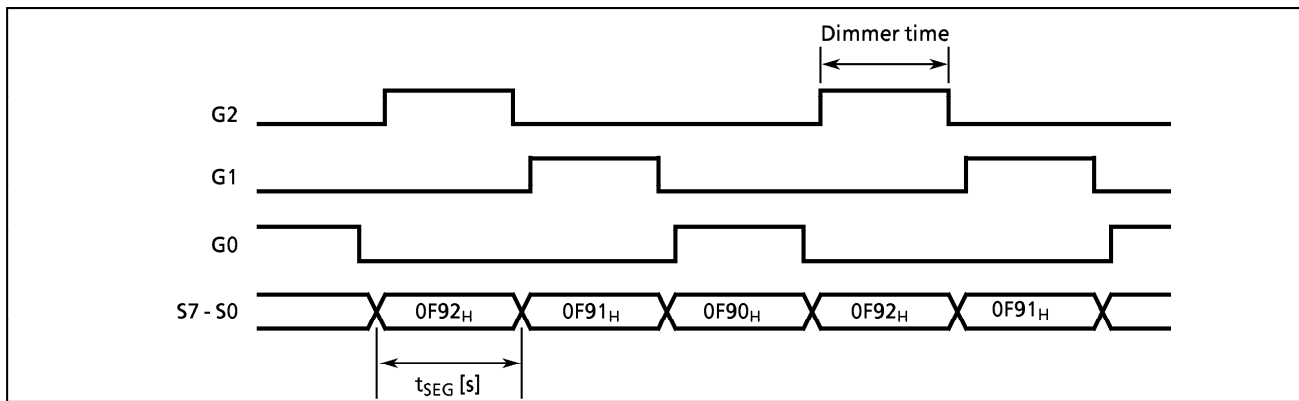


Figure 2-46. VFT Drive Waveform (Example : 8-segment x 3-digit display)

2.12.5 Key Scan Function

The key strobe signal can be output automatically. The output latch should be set to "1" to use the port P4 (KEY) for key scan input. Setting EKEY (bit 6 in VFTCR1) to "1", the key scan timing is inserted into the display operation synchronizes with the scan rate selected with STCK. A key strobe pulse is output at the key scan timing from the pin Sn (one of the scan segment output pins selected with SEG). The segment pin number "n" is decremented at each key scan timing. For example, in the 8-segment display mode, the output would be from the S7→S6→...→S0→S7 pins.

The KEY7 - KEY0 pin input data synchronized with the key strobe output are latched to the key scan input latches (KEYDR) ; therefore, key scan can be performed by configuring a key matrix with the segment output pins and KEY 7 - KEY0 pins.

The key scan interrupt request (INTKEY) is generated in synchronized with key data latching; therefore, the key strobe bit can be identified by sensing the key scan status register (002A_H) with an interrupt service routine and reading the stored key data (return code) from the key scan input latches (KEYDR).

The following key scan timing is canceled until KEYDR has been read out.

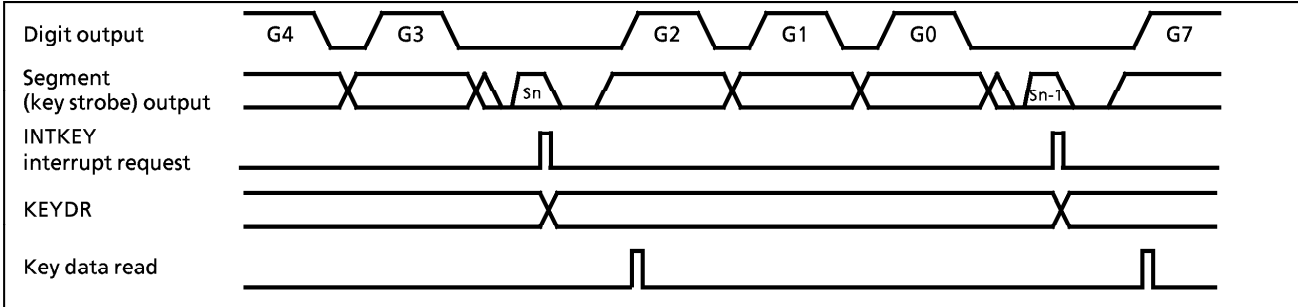


Figure 2-47. Key Scan Timing (8-digit Display)

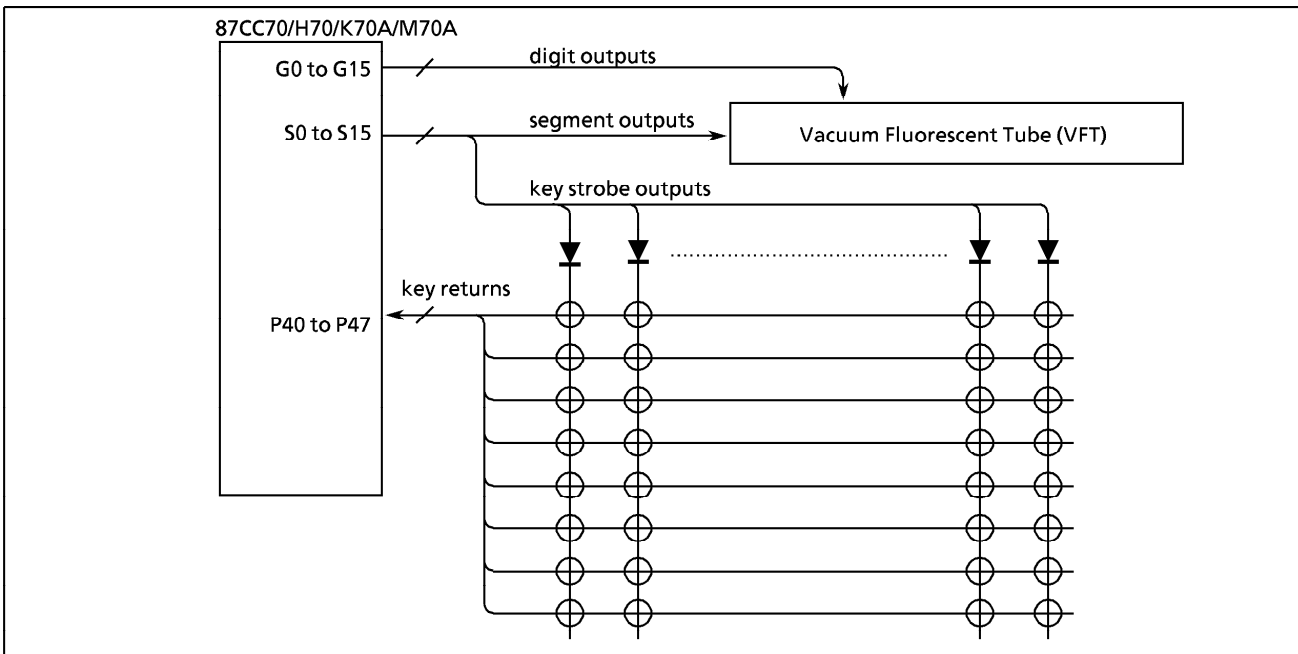


Figure 2-48. Key Matrix Example

2.12.6 Port Function

(1) High-Breakdown Voltage Buffer

When a VFT is being driven, the output latch is cleared to "0". The port output latch is initialized to "0" during reset).

When using as a normal input/output pin, caution is required because of being pulled down to the VKK pin voltage internally.

a. Output

The pins are brought to the V_{KK} pin voltage by the built-in pull-down resistor for "low" level output; consequently, as shown in Figure 2-49, diode grounding is necessary to prevent the V_{KK} pin voltage being applied to the external circuitry.

b. Input

The port output latch should be cleared to "0" when inputting external data. The input threshold level is same as for the other normal I/O port but, because of being pulled down to the V_{KK} pin voltage, R_K (typ. 80kΩ) must be fully driven.

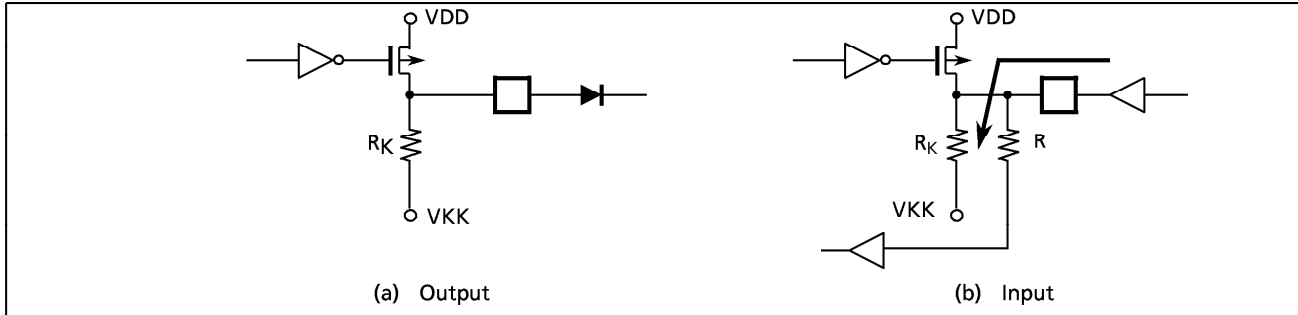


Figure 2-49. External Circuit Interface

(2) Key Scan Input Pins

The pins P47 (KEY7) to P40 (KEY0) are controlled by turning on and off the built-in pull-down resistance for each bit with a P4 port control register (P4CR).

Note 1: The key scan input latch (address 0004_H) is only effective when EKEY in VFTR1 is set to "1". The initial value is "0".
Note 2: When EKEY is set to "1", used for key scan input and cannot be used for normal input or bit operations.

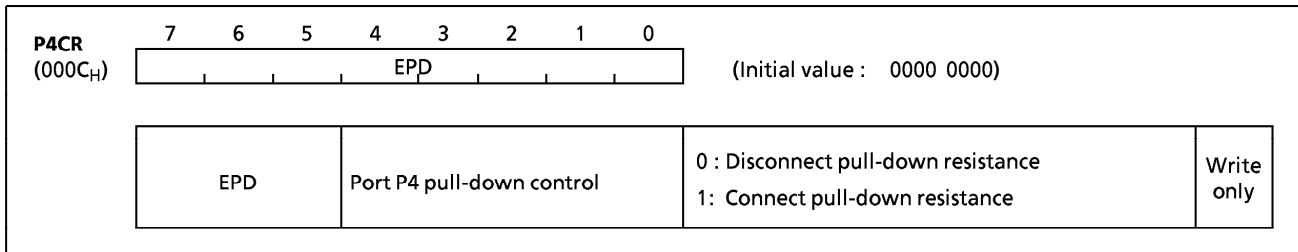


Figure 2-50. Port P4 Control Register

INPUT/OUTPUT CIRCUITRY

(1) Control pins

The input/output circuitries of the 87CC70/H70/K70A/M70A 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) $R_f = 1.2\text{ M}\Omega$ (typ.) $R_O = 1.5\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.) | | | | |
| XTIN (P21) XTOUT (P22) | Input Output | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; text-align: center;">NM1</td> <td style="width: 80%; text-align: center;">NM2</td> </tr> <tr> <td style="vertical-align: middle;">Refer to port P2</td> <td> </td> </tr> </table> | NM1 | NM2 | Refer to port P2 | | Resonator connecting pins (low-frequency) $R_f = 6\text{ M}\Omega$ (typ.) $R_O = 220\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.) |
| NM1 | NM2 | | | | | | |
| Refer to port P2 | | | | | | | |
| $\overline{\text{RESET}}$ | I/O | | Sink open drain output Hysteresis input Pull-up resistor $R_{IN} = 220\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.) | | | | |
| $\overline{\text{STOP}} / \overline{\text{INT5}}$ (P20) | Input | | Hysteresis input $R = 1\text{ k}\Omega$ (typ.) | | | | |
| TEST | Input | | Pull-down resistor $R_{IN} = 70\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.) | | | | |

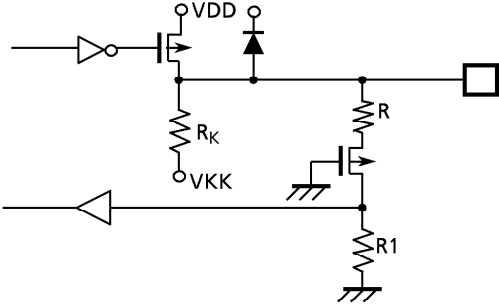
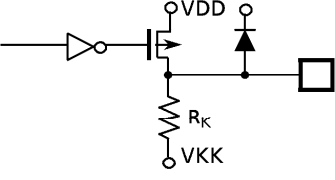
Note1 : The TEST pin of the 87PM70 does not have a pull-down resistor.
 Note2 : The 87PM70 is placed in the single-clock mode during reset, and the input/output circuitries are the code NM1 type.

(2) Input/Output Ports

The input/output circuitries of the 87CC70/H70/K70A/M70A input / output ports are shown below.
A mask option code is only "A".

| PORT | I/O | INPUT / OUTPUT CIRCUITRY (CODE A) | REMARKS |
|------|-----|-----------------------------------|---|
| P0 | I/O | <p>initial "Hi-Z"</p> | <p>Tri-state I/O</p> <p>R = 1 kΩ (typ.)</p> |
| P1 | I/O | <p>initial "Hi-Z"</p> | <p>Tri-state I/O</p> <p>Hysteresis input</p> <p>R = 1 kΩ (typ.)</p> |
| P2 | I/O | <p>initial "Hi-Z"</p> | <p>Sink open drain output</p> <p>R = 1 kΩ</p> |
| P3 | I/O | <p>initial "Hi-Z"</p> | <p>Sink open drain output</p> <p>Hysteresis input</p> <p>R = 1 kΩ (typ.)</p> |
| P4 | I/O | <p>initial "Hi-Z"</p> | <p>Sink open drain output</p> <p>Hysteresis input</p> <p>R_{IN} = 70 kΩ (typ.)</p> <p>R_{IN} is programable pull-down.</p> <p>R = 1 kΩ (typ.)</p> |
| P5 | I/O | <p>initial "Hi-Z"</p> | <p>Sink open drain output</p> <p>R = 1 kΩ (typ.)</p> |

To be continued.

| PORT | I/O | INPUT / OUTPUT CIRCUITRY (CODE A) | REMARKS |
|----------------|--------|---|--|
| P6 P7 P9 | I/O | <p>initial "Hi-Z"</p>  | <p>Source open drain output</p> <p>High-breakdown voltage $R_K = 80\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.)</p> |
| P8 | Output | <p>initial "Hi-Z"</p>  | <p>Source open drain output</p> <p>High-breakdown voltage $R_K = 80\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.)</p> |

ELECTRICAL CHARACTERISTICS

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

| PARAMETER | SYMBOL | CONDITIONS | RATINGS | UNIT |
|----------------------------------|---------------------|--------------------------|---|------|
| Supply Voltage | V _{DD} | | - 0.3 to 7 | V |
| Input Voltage | V _{IN} | | - 0.3 to V _{DD} + 0.3 | V |
| Output Voltage | V _{OUT1} | P2, P4, P5, XOUT, RESET | - 0.3 to V _{DD} + 0.3 | V |
| | V _{OUT2} | P3 | - 0.3 to 10 | |
| | V _{OUT3} | Source open drain ports | V _{DD} - 40 to V _{DD} + 0.3 | |
| Output Current (Per 1 pin) | I _{OUT1} | P0, P1, P2, P3, P4, P5 | 3.2 | mA |
| | I _{OUT3} | P8, P9 (segment outputs) | - 12 | |
| | I _{OUT4} | P6, P7 (digit outputs) | - 25 | |
| Output Current (Total) | Σ I _{OUT1} | P0, P1, P2, P3, P4, P5 | 120 | mA |
| | Σ I _{OUT2} | P6, P7, P8, P9 | - 120 | |
| Power Dissipation [Topr = 70 °C] | PD | | 350 | mW |
| Soldering Temperature (time) | Tsld | | 260 (10 s) | °C |
| Storage Temperature | Tstg | | - 55 to 125 | °C |
| Operating Temperature | Topr | | - 30 to 70 | °C |

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0 V, Topr = - 30 to 70 °C)

| PARAMETER | SYMBOL | PINS | CONDITIONS | Min. | Max. | UNIT | |
|--------------------|------------------|-------------------------|------------------------------|------------------------|------------------------|------|---|
| Supply Voltage | V _{DD} | | fc = 8 MHz | NORMAL1, 2 modes | 4.5 | 6.0 | V |
| | | | | IDLE1, 2 modes | | | |
| | | | fc = 4.2 MHz | NORMAL1, 2 modes | 2.7 | | |
| | | | | IDLE1, 2 modes | | | |
| | | | fs = 32.768 kHz | SLOW mode | 2.0 | | |
| SLEEP mode | | | | | | | |
| | | STOP mode | | | | | |
| Input High Voltage | V _{IH1} | Except hysteresis input | V _{DD} ≥ 4.5 V | V _{DD} × 0.70 | V _{DD} | V | |
| | V _{IH2} | Hysteresis input | | V _{DD} × 0.75 | | | |
| | V _{IH3} | | V _{DD} < 4.5 V | V _{DD} × 0.90 | | | |
| Input Low Voltage | V _{IL1} | Except hysteresis input | V _{DD} ≥ 4.5 V | 0 | V _{DD} × 0.30 | V | |
| | V _{IL2} | Hysteresis input | | | V _{DD} × 0.25 | | |
| | V _{IL3} | | V _{DD} < 4.5 V | V _{DD} × 0.10 | | | |
| Clock Frequency | fc | XIN, XOUT | V _{DD} = 4.5 to 6 V | 0.4 | 8.0 | MHz | |
| | | | V _{DD} = 2.7 to 6 V | | 4.2 | | |
| | fs | XTIN, XTOUT | | 30.0 | 34.0 | kHz | |

Note : The 87CC70/H70 are not guaranteed V_{DD} = 2.7 to 6 V in NORMAL1, 2 and IDLE1, 2 modes.

| D.C. CHARACTERISTICS | | (V _{SS} = 0 V, T _{opr} = -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} | RESET, STOP | | | | | | |
| Input Resistance | R _{IN1} | Port P4 with pull-down | | 30 | 70 | 150 | kΩ | |
| | R _{IN2} | RESET | | 100 | 220 | 450 | | |
| Pull-down Resistance | R _K | Source open drain ports | V _{DD} = 5.5 V, V _{KK} = -30 V | - | 80 | - | | |
| 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 | V _{DD} = 5.5 V, V _{OUT} = -32 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 | V _{DD} = 4.5 V, I _{OH} = -5 mA | 2.4 | - | - | | |
| Output Low Voltage | V _{OL} | Except XOUT | V _{DD} = 4.5 V, I _{OL} = 1.6 mA | - | - | 0.4 | V | |
| Output High current | I _{OH} | P6, P7 | V _{DD} = 4.5 V, V _{OH} = 2.4 V | - | -15 | - | mA | |
| Supply Current in NORMAL 1, 2 modes | I _{DD} | | V _{DD} = 5.5 V f _c = 8 MHz | 87CC70/H70 | - | 8 | 14 | mA |
| | | | | 87CK70A/M70A | - | 10 | 16 | |
| Supply Current in IDLE 1, 2 modes | | | | f _s = 32.768 kHz | 87CC70/H70 | - | 4.0 | 6 |
| | | | | V _{IN} = 5.3 V/0.2 V | 87CK70A/M70A | - | 4.5 | |
| Supply Current in SLOW mode | | | | V _{DD} = 3.0 V f _s = 32.768 kHz | | - | 30 | 60 |
| Supply Current in SLEEP mode | | V _{IN} = 2.8 V/0.2 V | | - | 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 T_{opr} = 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 : Typical current consumption during A/D conversion is 1.2 mA.

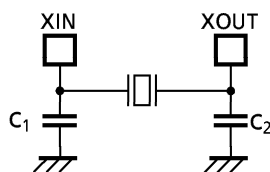
| A/D CONVERSION CHARACTERISTICS | | (V _{SS} = 0 V, V _{DD} = 4.5 to 6.0 V, T _{opr} = -30 to 70 °C) | | | | | |
|--------------------------------|------------------|--|-------------------------|-----------------|------|-----------------|------|
| PARAMETER | SYMBOL | PINS | CONDITIONS | Min. | Typ. | Max. | UNIT |
| Analog Input Voltage Range | V _{CIN} | CIN5 to CIN0 | | V _{SS} | - | V _{DD} | V |
| Conversion Error | | | V _{DD} = 5.0 V | - | - | ± 1.5 | LSB |

A.C. CHARACTERISTICS ($V_{SS} = 0\text{ V}$, $V_{DD} = 4.5\text{ to }6.0\text{ V}$, $T_{opr} = -30\text{ to }70\text{ }^{\circ}\text{C}$)

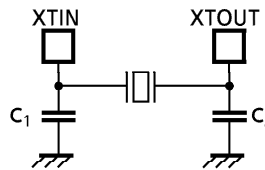
| PARAMETER | SYMBOL | CONDITIONS | Min. | Typ. | Max. | UNIT |
|------------------------------|-----------|--|-------|------|-------|---------------|
| Machine Cycle Time | t_{cy} | In NORMAL 1, 2 modes | 0.5 | - | 10 | μs |
| | | In IDLE 1, 2 modes | | | | |
| | | In SLOW mode | 117.6 | - | 133.3 | |
| | | In SLEEP mode | | | | |
| High Level Clock Pulse Width | t_{WCH} | For external clock operation (XIN input), $f_c = 8\text{ MHz}$ | 50 | - | - | ns |
| Low Level Clock Pulse Width | t_{WCL} | | | | | |
| High Level Clock Pulse Width | t_{WSH} | For external clock operation (XTIN input), $f_s = 32.768\text{ kHz}$ | 14.7 | - | - | μs |
| Low Level Clock Pulse Width | t_{WSL} | | | | | |

RECOMMENDED OSCILLATING CONDITIONS ($V_{SS} = 0\text{ V}$, $V_{DD} = 4.5\text{ to }6.0\text{ V}$, $T_{opr} = -30\text{ to }70\text{ }^{\circ}\text{C}$)

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



(1) High-frequency Oscillation



(2) Low-frequency Oscillation

Note : An electrical shield by metal shield plate on the surface of the 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.

