TOSHIBA

TOSHIBA Original CMOS 8-bit Microcontroller

TLCS-870 Series

TMP87EP26

Databook

6th Edition

TOSHIBA CORPORATION

CMOS 8-bit Microcontroller

TMP87EP26F

The TMP87EP26 is the high speed and high performance 8-bit single chip microcomputers. This MCU contains, ROM, RAM, E²PROM, input/output ports, LCD driver, a 8-bit AD converter, four multi-function timer/counters, two serial interfaces, and two clock generators on chip.
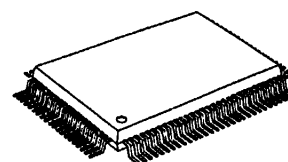
Product No.	ROM	RAM	E²PROM	Package	Serial PROM Board
TMP87EP26F	32 K x 8-bit	2 K x 8-bit	16 K x 8-bit (Program memory) 512 x 8-bit (Data memory)	P-QFP100-1420-0.65A	BM11185* (CE marking is not supported)

*Under Development

Features

◆ 8-bit single chip microcomputer TLCS-870 Series
◆ Instruction execution time: 0.5 μs (at 8 MHz), 122 μs (at 32.768 kHz)
◆ 129 types and 412 basic instructions
   ● Multiplication and Division
     (8 bits x 8 bits , 16 bits ÷ 8 bits): Execution time 3.5 μs (at 8 MHz)
   ● Bit manipulations
     (Set/Clear/Complement/Load/Store/Test/Exclusive OR)
   ● 16-bit data operations
   ● 1-byte jump/call (Short relative jump / Vector call)
◆ 15 interrupt sources (External:5, Internal:10)
   ● All sources have independent latches each,
     and nested interrupt control is available
   ● 4 edge-selectable external interrupts with noise reject
   ● High-speed task switching by register bank changeover
◆ 11 Input/Output ports (Max 73 pins)
◆ Two 16-bit Timer/Counters
   ● Timer, Event counter, Frequency measurement,
     Pulse width measurement modes
◆ Two 8-bit Timer/Counters
   ● Timer, Event counter, Capture (Pulse width/duty measurement),
   PWM output, PDO modes

P-QFP100-1420-0.65A

TMP87EP26F

000707EBP1

◆Time Base Timer (Interrupt frequency: 1 Hz to 16384 Hz)
◆Divider output function (frequency: 1 kHz to 8 kHz)
◆Watchdog Timer
◆Two 8-bit Serial Interfaces
   ● Each 8 bytes transmit/receive data buffer
   ● Internal/external serial clock, and 4/8-bit mode
◆LCD driver
   ● With displaymemory (20 bytes)
   ● LCD direct drive capability (Max. 40 seg. × 4 com.)
   ● 1/4, 1/3, 1/2 duty or static drive are programmably selectable
◆8-bit successive approximate type AD converter with sample and hold
   ● 8 analog inputs
   ● Conversion time: 23 μs / 92 μs (at 8 MHz)
◆Dual clock operation (optional)
◆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.
◆Operating Voltage: 4.5 to 5.5 V at 8 MHz / 32.768 kHz
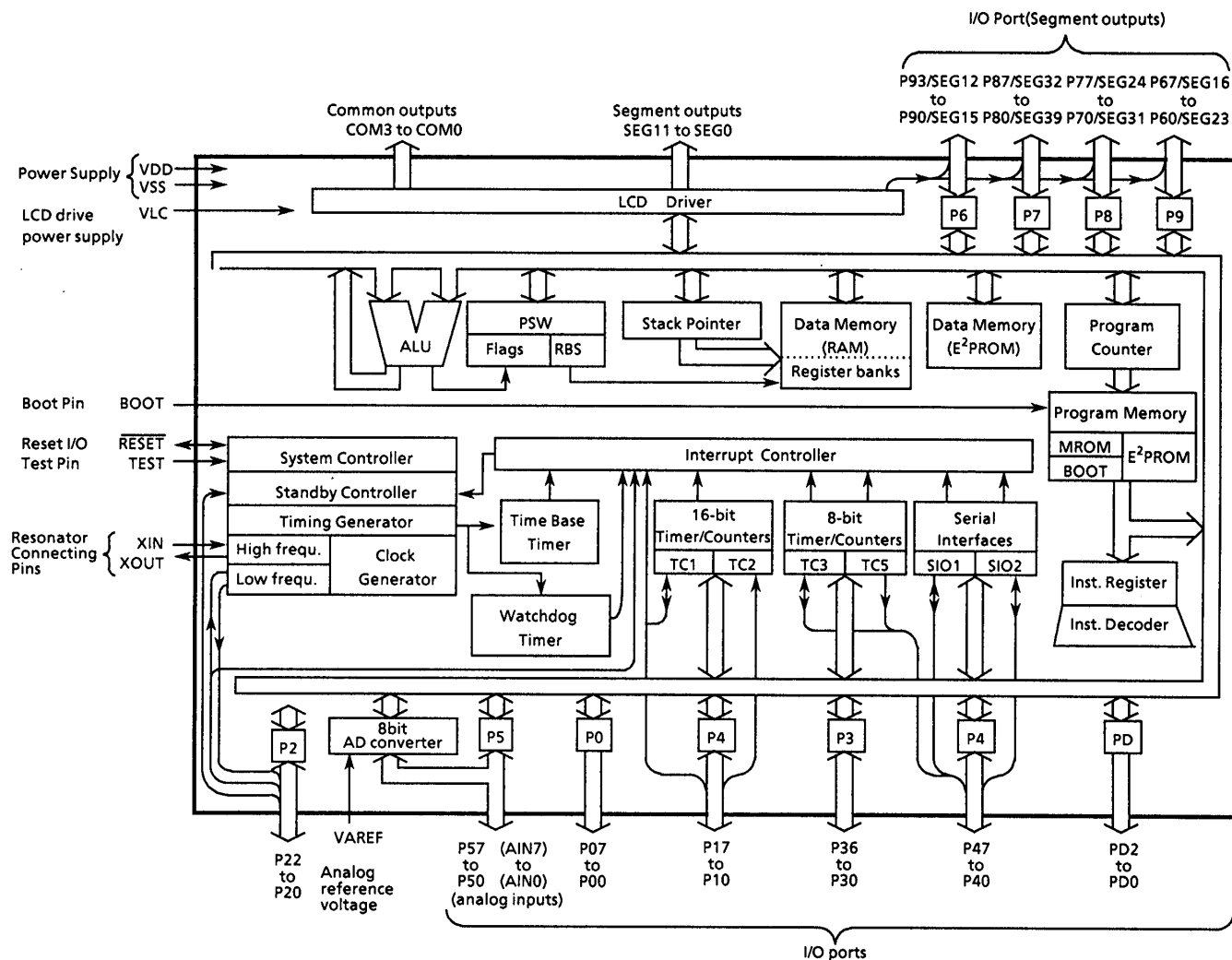◆Emulation Pod: BM87EP26F0A

Pin Assignments (Top View)

P-QFP100-1420-0.65A
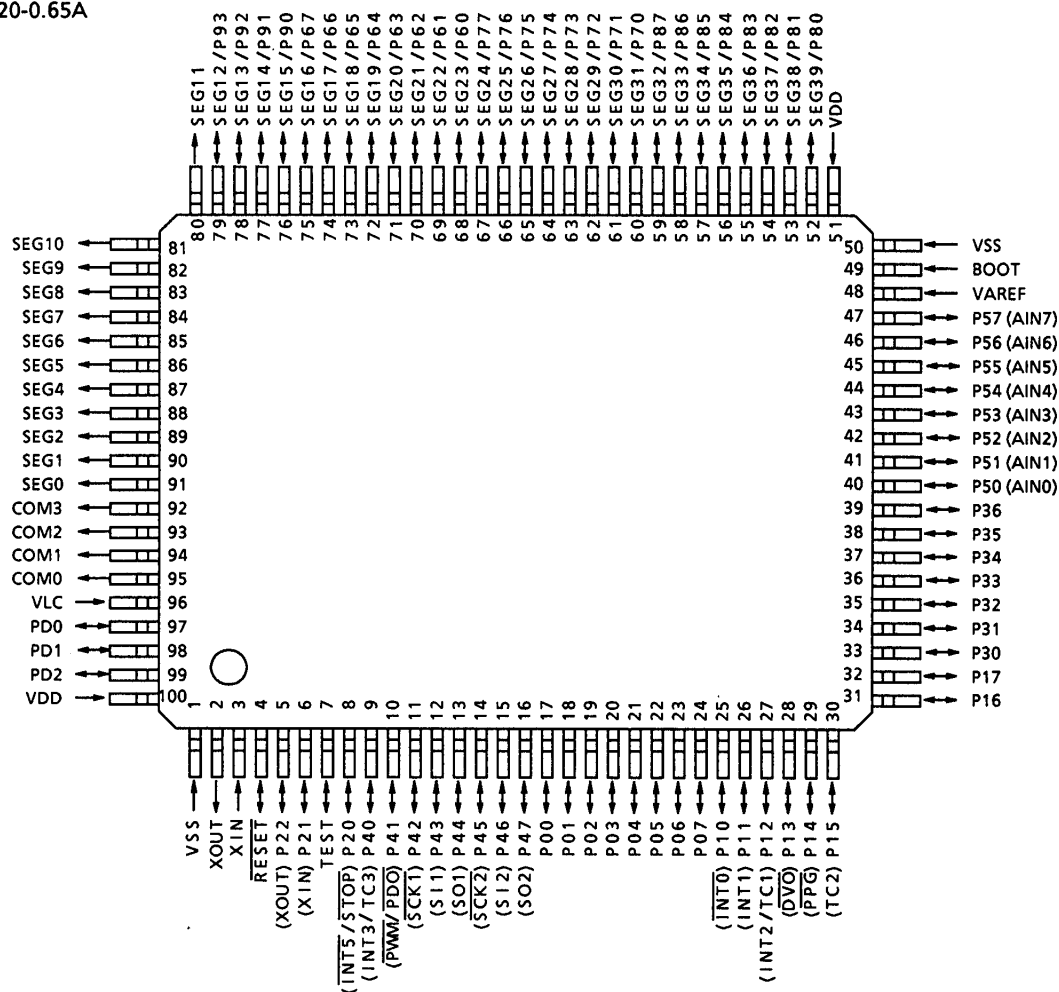
Note: TMP87EP26 has two VDD pins and two VSS pins for power supply.
All VDD(VSS) pins should be connected externally for keeping the same voltage level.

Block Diagram

Pin Function

Pin Name	Input / Output	Function	
P07 to P00	I/O	8-bit programmable input/output ports (tri-state).	
P17, P16	I/O		
P15 (TC2)	I/O (Input)	Each bit of these ports can be individually configured as an input or an output under software control. When used as timer/counter input or external interrupt input, the input mode should be set. When used as PPG output or divider output, output mode should be set and the latch must be set to "1".	Timer/Counter 2 input
P14 ($\overline{\text{PPG}}$)	I/O (Output)		Programmable pulse generator output
P13 ($\overline{\text{DVO}}$)			Divider output
P12 (INT2 / TC1)			External interrupt 2 input or Timer/Counter 1 input
P11 (INT1)	I/O (Input)		External interrupt 1 input
P10 ($\overline{\text{INT0}}$)			External interrupt 0 input
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 (32.768kHz). For inputting external clock, XTIN is used and XTOUT is opened.
P21 (XTIN)	I/O (Input)		
P20 ($\overline{\text{INT5}}$/$\overline{\text{STOP}}$)			External interrupt 5 input or STOP mode release signal input
P36 to P30	I/O	7-bit input/output port with latch. When used as input port, the latch must be set to "1".	
P47 (SO2)	I/O (Output)	8-bit input/output port with latch. When used as input port, a SIO input/output port or timer/counter input/output, the latch must be set to "1".	SIO2 serial data output
P46 (SI2)	I/O (Input)		SIO2 serial data input
P45 ($\overline{\text{SCK2}}$)	I/O (I/O)		SIO2 serial clock input/output
P44 (SO1)	I/O (Output)		SIO1 serial data output
P43 (SI1)	I/O (Input)		SIO1 serial data input
P42 ($\overline{\text{SCK1}}$)	I/O (I/O)		SIO1 serial clock input/output
P41 ($\overline{\text{PWM}}$/$\overline{\text{PDO}}$)	I/O (Output)		8-bit PWM output, 8-bit programmable divider output
P40 (INT3/TC3)	I/O (Input)		External interrupt 3 input, Timer/Counter 3 input
P57 (AIN7) to P50 (AIN0)	I/O (Input)	8-bit programmable input/output port (tri-state). Each bit of the port can be individually configured as an input or an output under software control. When used as analog input, input mode should be set.	AD converter analog inputs
PD2 to PD0	I/O	3-bit input/output port with latch. When used as an input port, the latch must be set to "1".	
SEG39 (P80) to SEG32 (P87)	Output (I/O)	8-bit input/output port with latch. When used as an input port, the latch must be set to "1".	LCD segment outputs. When used as segment output, the control register of P6, P7, P8 and P9 must be set to "1".
SEG31 (P70) to SEG24 (P77)	Output (I/O)		
SEG23 (P60) to SEG16 (P67)	Output (I/O)		
SEG15 (P90) to SEG12 (P93)	Output (I/O)	4-bit input/output port with latch. When used as an input port, the latch must be set to "1".	
SEG11 to SEG0	Output	LCD segment outputs	
COM3 to COM0	Output	LCD common outputs	
XIN, XOUT	Input, Output	Resonator connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened.	
$\overline{\text{RESET}}$	I/O	Reset signal input or watchdog timer output/address-trap-reset output	
TEST	Input	Test pin for out-going test. Be fixed to low.	
VDD, VSS	Power Supply	+ 5 V, 0 V (GND)	
VAREF		Analog reference voltage inputs (High)	
VLC	LCD drive power supply.		
BOOT	Input	Control input for writing E²PROM of program area. When the E²PROM program area is written, be fixed to high level. Usually, be fixed to low level.	

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 TMP87EP26 is capable of addressing 64 Kbytes of memory. Figure 1-1 shows the memory address maps. The memory is organized 5 address spaces (ROM, E²PROM, 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.
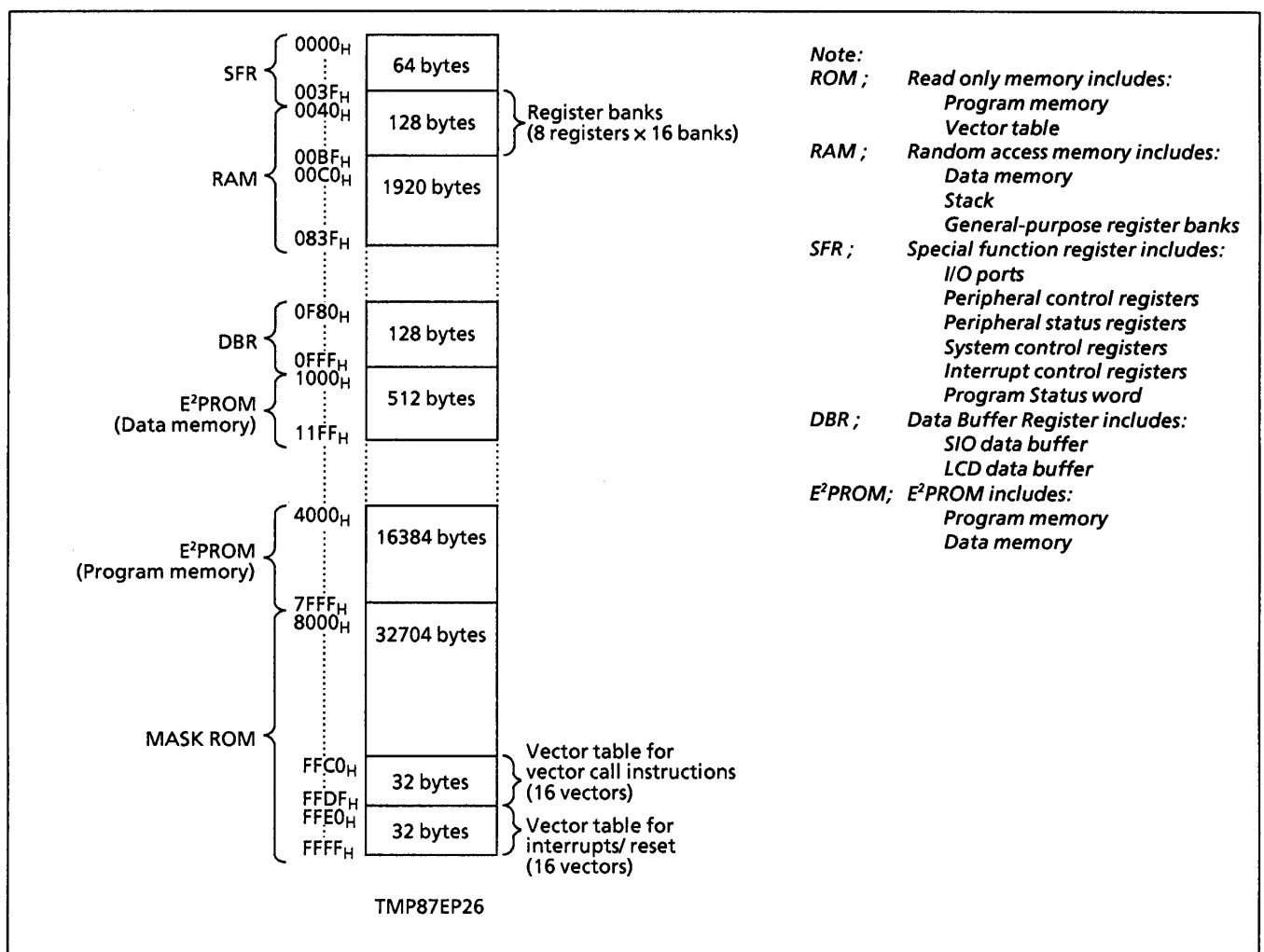

Figure 1-1. Memory Address Maps

1.2   Program Memory (ROM)

The TMP87EP26 has a 32 K×8-bit (addresses 8000$_H$ to FFFF$_H$) of program memory (mask programmed ROM).

Addresses FF00$_H$ to FFFF$_H$ in the program memory can also be used for special purposes.

(1) **Interrupt/ Reset** vector table (addresses FFE0$_H$ to FFFF$_H$)
This table consists of a reset vector and 16 interrupt vectors (2 bytes/vector). These vectors store a reset start address and interrupt service routine entry addresses.

(2) Vector table for **vector call** instructions (addresses FFC0$_H$ to FFDF$_H$)
This table stores call vectors (subroutine entry address, 2 bytes/vector) for the vector call instructions [CALLV n]. There are 16 vectors. The CALLV instruction increases memory efficiency when utilized for frequently used subroutine calls (called from 3 or more locations).

(3) Entry area (addresses FF00$_H$ to FFFF$_H$) for **page call** instructions
This is the subroutine entry address area for the page call instructions [CALLP n]. Addresses FF00$_H$ to FFBF$_H$ are normally used because address FFC0$_H$ to FFFF$_H$ are used for the vector tables.

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

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

① 5-bit PC-relative jump [JRS cc, $ + 2 + d]
E8C4H: JRS   T, $ + 2 + 08H
When JF = 1, the jump is made to E8CE$_H$, which is 08$_H$ added to the contents of the PC. (The PC contains the address of the instruction being executed + 2; therefore, in this case, the PC contents are E8C4$_H$ + 2 = E8C6$_H$.)

② 8-bit PC-relative jump [JR cc, $ + 2 + d]
E8C4H: JR   Z, $ + 2 + 80H
When ZF = 1, the jump is made to E846$_H$, which is FF80$_H$ (−128) added to the current contents of the PC.

③ 16-bit absolute jump [JP   a]
E8C4H: JP   0C235H
An unconditional jump is made to address C235$_H$. The absolute jump instruction can jump anywhere within the entire 64K-bytes space.
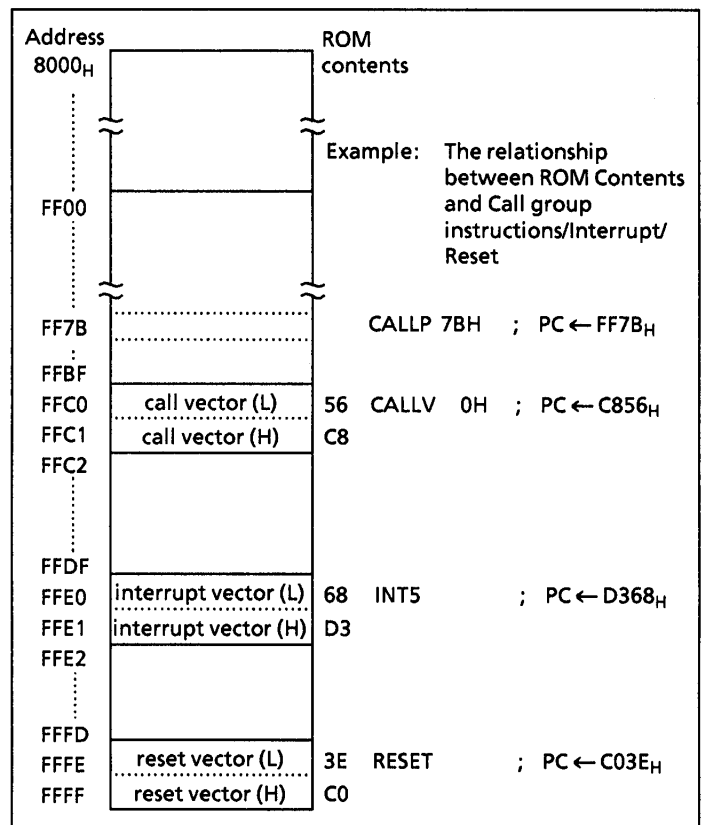
Figure 1-2. Program Memory Map

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

Example 1: Loads the ROM contents at the address specified by the HL register pair contents into the accumulator (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 P6 after executing the following program:

         ADD   A, TABLE – $ – 4   ; P6 ←ROM (TABLE + A)
         LD   (P6), (PC + A)
         JRS   T, SNEXT   ; Jump to SNEXT
TABLE: DB   0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0D8H, 80H, 98H
SNEXT:

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

Example 3: N-way multiple jump in accordance with the contents of accumulator (0 $\leq$ 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$_H$.
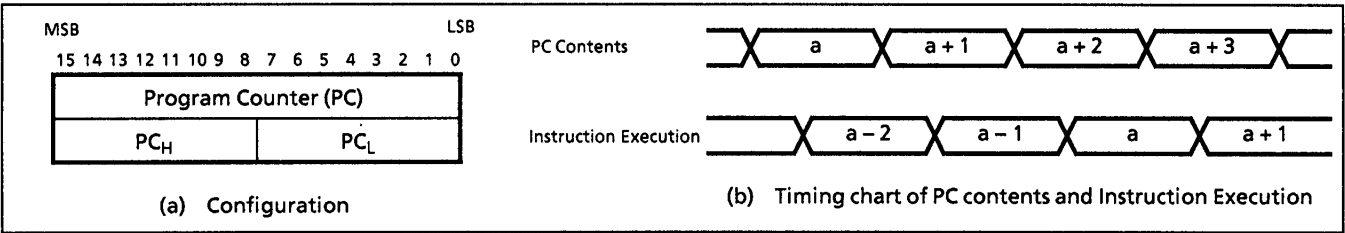
Figure 1-3. Program Counter

1.4 Data Memory (RAM)

The TMP87EP26 has a 2 K×8-bit (address 0040_H to $083F_H$) of data memory (static RAM). Figure 1-4 shows the data memory map.

Addresses 0000_H to $00FF_H$ are used as a direct addressing area to enhance instructions which utilize this addressing mode; therefore, addresses 0040_H to $00FF_H$ in the data memory can also be used for user flags or user counters.

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
```

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

With the TMP87EP26, programs in data memory cannot be executed. If the program counter indicates a specific data memory address (addresses 0040_H to $083F_H$), an address-trap-reset is generated due to due to bus error. (Output from the $\overline{RESET}$ pin goes low.)

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, 07F7H       ; Sets number of byte to BC register pair
SRAMCLR:    LD      (HL +), A
            DEC     BC
            JRS     F, SRAMCLR
```

> Note: The data memory contents become unstable when the power supply is turned on; therefore, the data memory should be initialized by an initialization routine. Note that the general-purpuse registers are mapped in the RAM ; therefore, do not clear RAM at the current bank addresses.

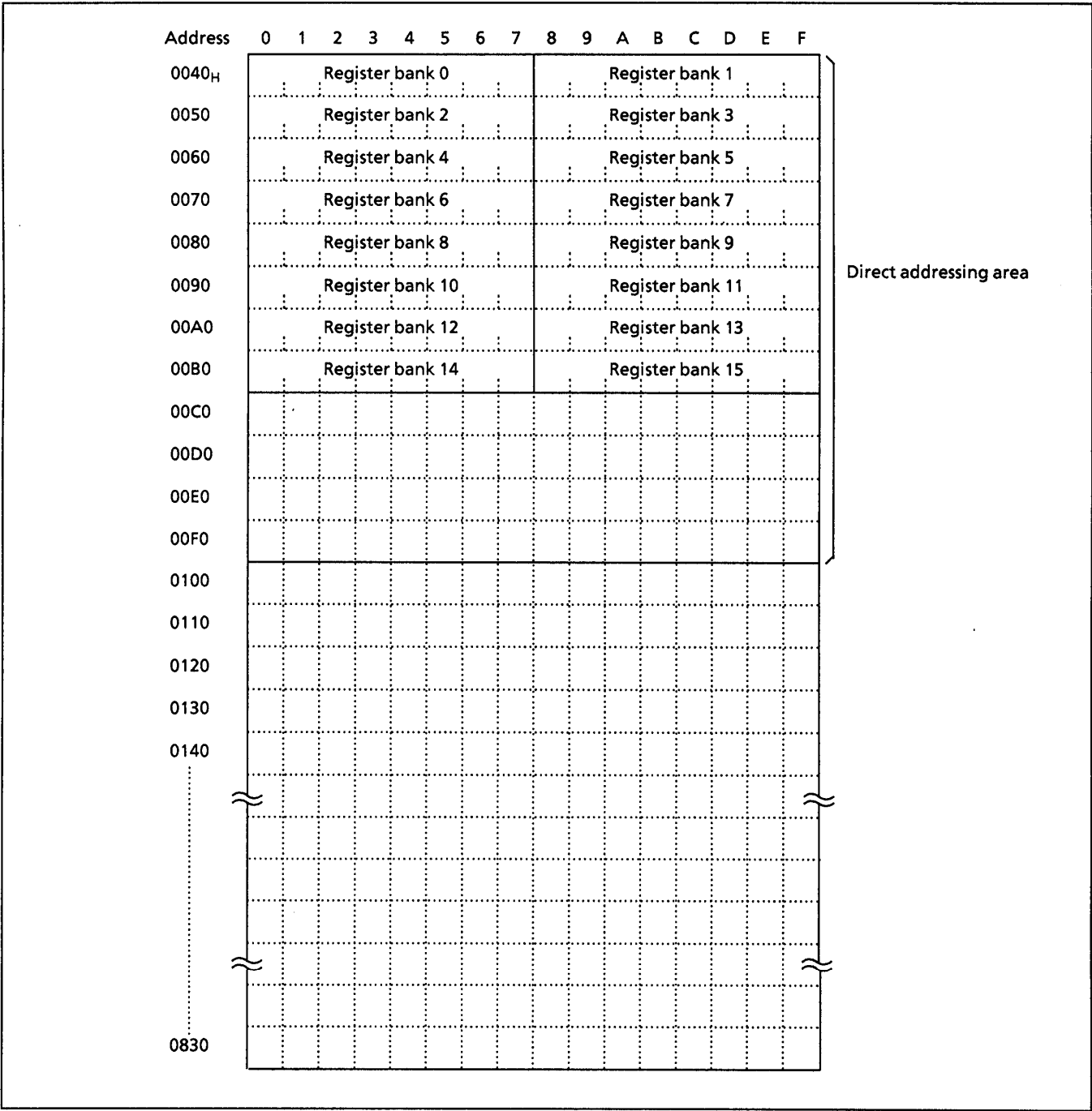| Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0040$_H$			Register bank 0								Register bank 1						
0050			Register bank 2								Register bank 3						
0060			Register bank 4								Register bank 5						
0070			Register bank 6								Register bank 7						
0080			Register bank 8								Register bank 9						
0090			Register bank 10								Register bank 11						
00A0			Register bank 12								Register bank 13						
00B0			Register bank 14								Register bank 15						Direct addressing area
00C0																	
00D0																	
00E0																	
00F0																	
0100																	
0110																	
0120																	
0130																	
0140																	
0830																	

Figure 1-4. Data Memory Map

1.5 General-purpose Register Banks

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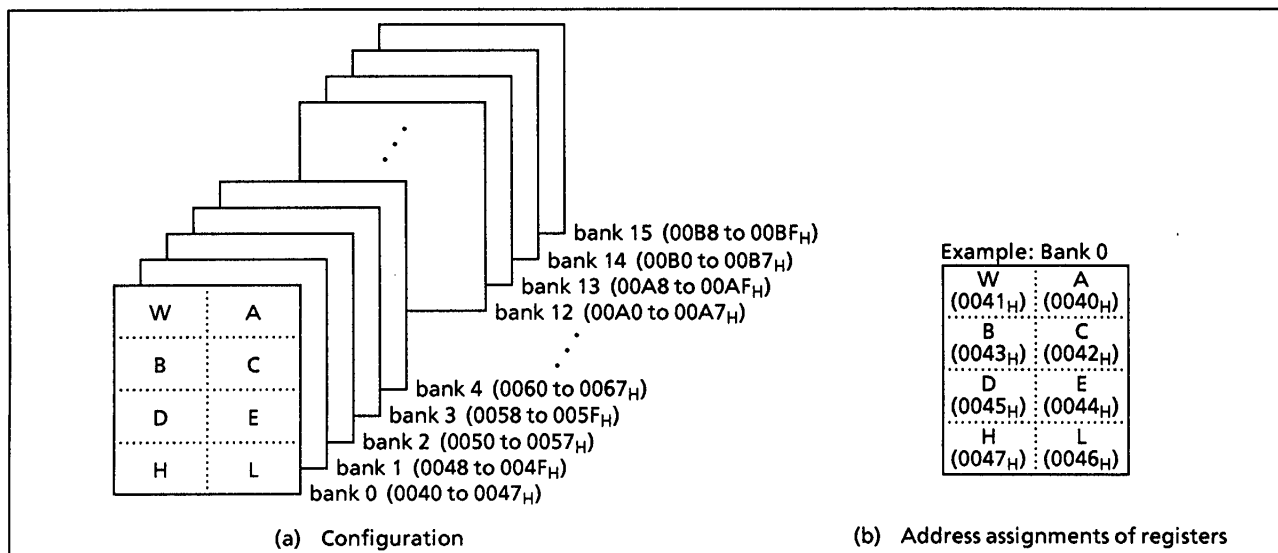
Figure 1-5. General-purpose Register Banks

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

(1) A, WA

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

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

Example 2: Block transfer

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

(3) B, C, BC

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

Example 1: Repeat processing

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

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

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

The general-purpose register banks are selected by the 4-bit register bank selector (RBS). During reset, the RBS is initialized to "0". The bank selected by the RBS is called the current bank.

Together with the flag, the RBS is assigned to address $003F_H$ in the SFR as the program status word (PSW). There are 3 instructions [LD RBS, n], [PUSH PSW], [POP PSW] to access the PSW. The PSW can be also operated by the memory access instruction.

Example 1: **Incrementing the RBS**

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

Example 2: **Reading the RBS**

```
            LD    A, (003FH)    ; A ← PSW (A_{3 to 0} ← RBS, A_{7 to 4} ← Flags)
```

Highly efficient programming and high-speed task switching are possible by using bank changeover to save registers during interrupt and to transfer parameters during subroutine processing.

During interrupt, the PSW is automatically saved onto the stack. The bank used before the interrupt was accepted is restored automatically by executing an interrupt return instruction [RETI]/[RETN] ; therefore, there is no need for the RBS save/restore software processing.

The TLCS-870 Series supports a maximum of 15 interrupt sources. One bank is assigned to the main program, and one bank can be assigned to each source. Also, to increase the efficiency of data memory usage, assign the same bank to interrupt sources which are not nested.

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

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

1.6 Program Status Word (PSW)

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

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

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

1.6.1 Register Bank Selector (RBS)

The register bank selector (RBS) is a 4-bit register used to select general-purpose register banks. For example, when RBS = 2, bank 2 is currently selected. During reset, the RBS is initialized to "0".
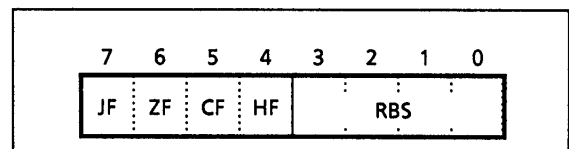
7	6	5	4	3	2	1	0
JF	ZF	CF	HF		RBS		

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

1.6.2 Flags

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

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

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

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

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

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

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

This flag is also a 1-bit register (a boolean accumulator) for the bit manipulation instructions. Set/clear/complement are possible with the CF manipulation instructions.

Example: Bit manipulation (The result of exclusive-OR between bit 5 content of address 07$_H$ and bit 0 content of address 9A$_H$ is written to bit 2 of address 01$_H$.)

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

(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 ← 41_H, HF ← 1	
DAA	A	; A ← 41_H + 06_H = 47_H (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 [JR T/F, $ + 2 + d], [JRS T/F, $ + 2 + d] (T or F is a condition code). Jump is performed if the JF is "1" for a true condition (T), or the JF is "0" for a false condition (F).

The JF is set to "1" after executing the load/exchange/swap/nibble rotate/jump instruction, so that [JRS T, $ + 2 + d] and [JR T, $ + 2 + d] can be regarded as an unconditional jump instruction.

Example: Jump status flag and conditional jump instruction

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

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

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

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

1.7 Stack and Stack Pointer

1.7.1 Stack

The stack provides the area in which the return address or status, etc. are saved before a jump is performed to the processing routine during the execution of a subroutine call instruction or the acceptance of an interrupt. On a subroutine call instruction [CALL a] / [CALLP n] / [CALLV n], the contents of the PC (the return address) is saved; on an interrupt acceptance, the contents of the PC and the PSW are saved (the PSW is pushed first, followed by PC_H and PC_L). Therefore, a subroutine call occupies two bytes on the stack; an interrupt occupies three bytes.

When returning from the processing routine, executing a subroutine return instruction [RET] restores the contents to the PC from the stack; executing an interrupt return instruction [RETI] / [RETN] restores the contents to the PC and the PSW (the PC_L is popped first, followed by PC_H and PSW).

The stack can be located anywhere within the data memory space except the register bank area, therefore the stack depth is limited only by the free data memory size.

1.7.2 Stack Pointer (SP)

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

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

MSB   LSB

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Stack Pointer (SP)

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, 083FH   ; SP←083F$_H$
Example 2: To read the SP
            LD   HL, SP   ; HL←SP
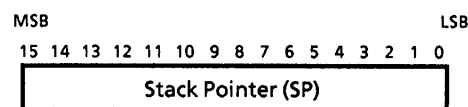

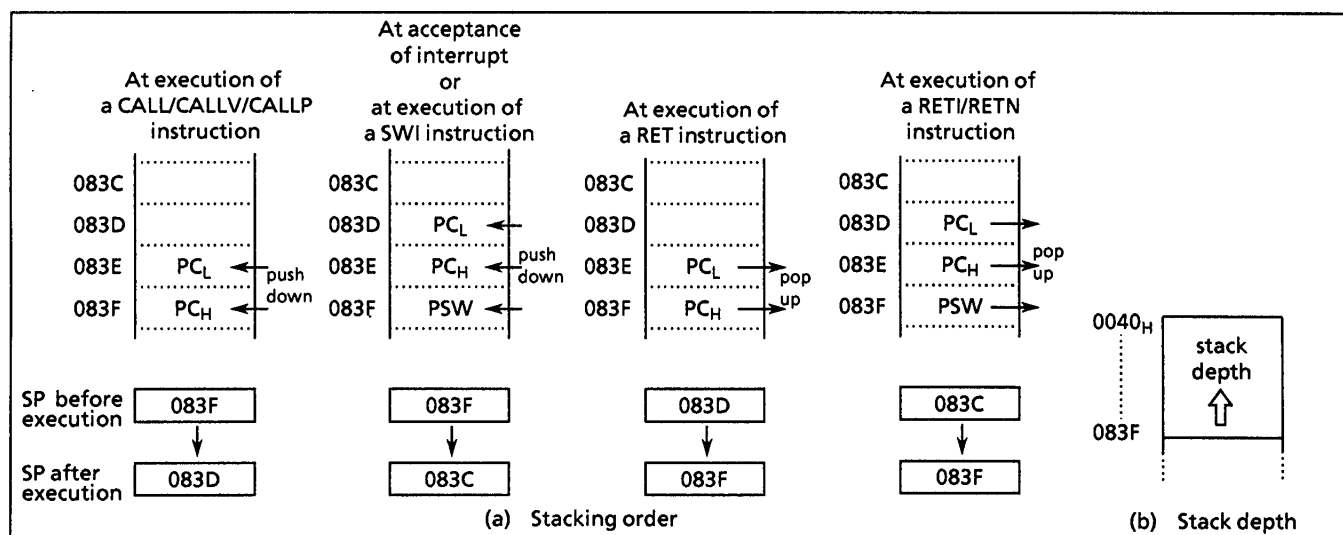(a) Stacking order   (b) Stack depth

Figure 1-8. Stack

1.8   System Clock Controller

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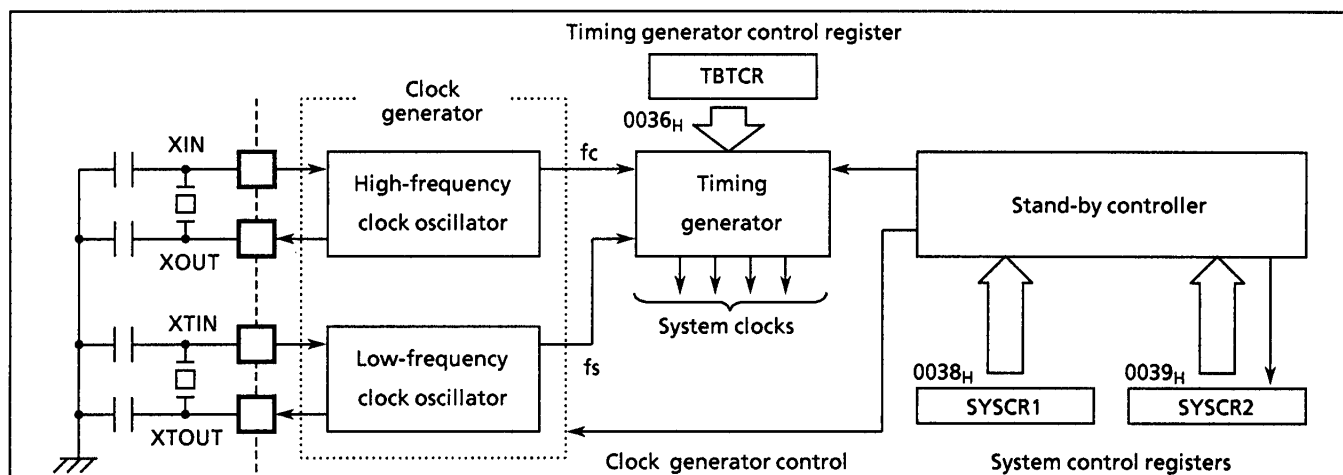

Figure 1-9. System Clock Controller

1.8.1 Clock Generator

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

The high-frequency (fc) and low-frequency (fs) clocks can be easily obtained by connecting a resonator between the XIN/XOUT and XTIN/XTOUT pins, respectively. Clock input from an external oscillator is also possible. In this case, external clock is applied to the XIN/XTIN pin with the XOUT/XTOUT pin not connected. The TMP87EP26 is not provided an RC oscillation.
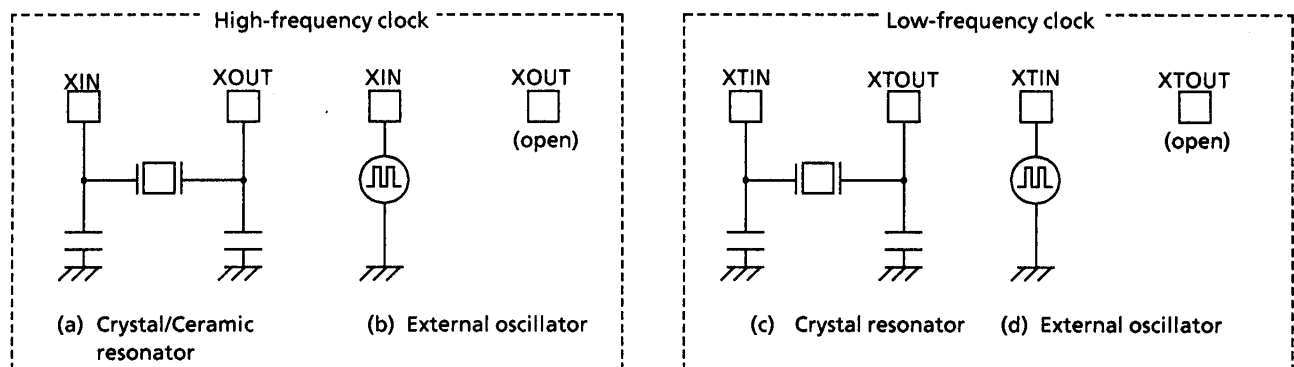
Figure 1-10. Examples of Resonator Connection

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

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 to TC3, TC5
⑥ Generation of internal clocks for serial interfaces SIO1 and SIO2
⑦ Generation of warm-up clocks for releasing STOP mode
⑧ Generation of a clock for releasing reset output

(1)   Configuration of Timing Generator

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

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

① In the single-clock mode

A divided-by-256 of high-frequency clock (fc/2⁸) is input to the 7th stage of the divider.
Do not set DV7CK to "1" in the single-clock mode.

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/2⁸" 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.

Note : The E²PROM write time (RAM emulation time) of the BM87EP26F0A differs from that of the TMP878EP26.

BM87EP26F0A write time   : 2¹⁵/fc[s]   (TBTCR<DV7CK> = "0")

                            : 2⁷/fs [s]   (TBTCR<DV7CK> = "1")

TMP87EP26 write time   : 2¹⁵/fc[s]   (Always the same regardless of TBTCR<DV7CK>)
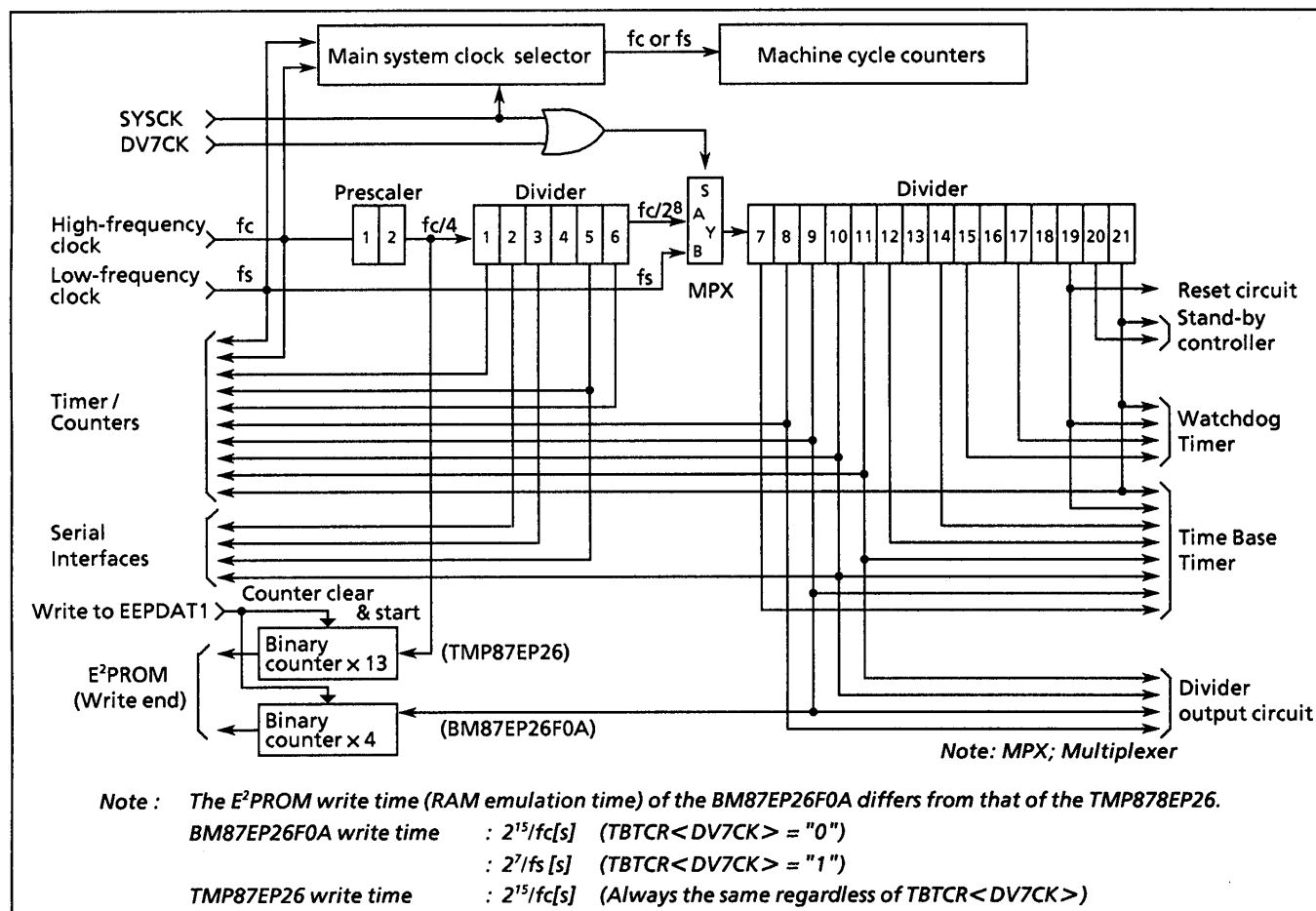
Figure 1-11. Configuration of Timing Generator
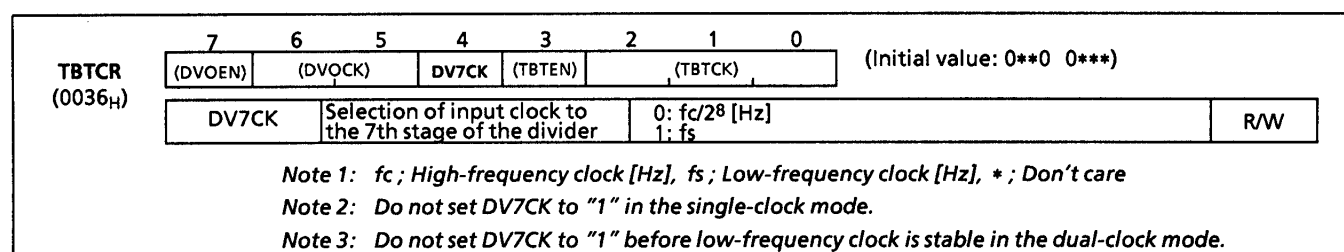
Figure 1-12. Timing Generator Control Register

(2) Machine Cycle

Instruction execution and peripheral hardware operation are synchronized with the main system clock. The minimum instruction execution unit is called 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 to S3), and each state consists of one main system clock.
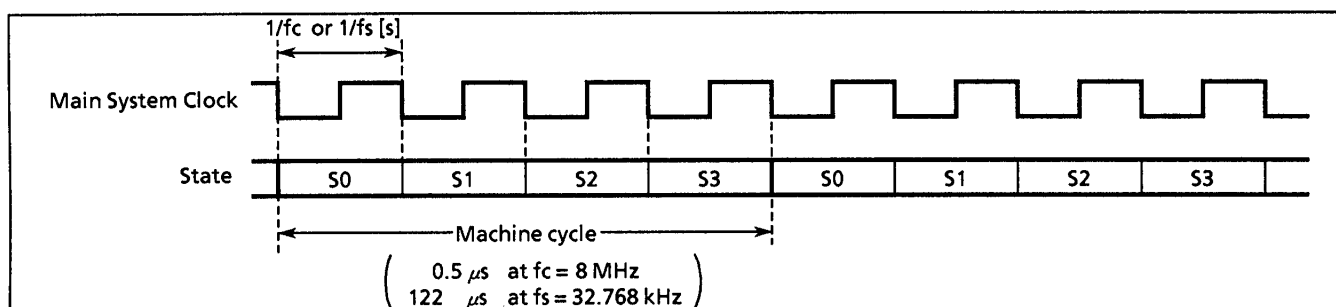
Figure 1-13. Machine Cycle

1.8.3 Stand-by Controller

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

Figure 1-14 show the operating mode transition diagram and Figure 1-15 shows the system control registers.

Either the single-clock or the dual-clock mode can be selected by an option during reset.

(1) **Single-clock mode**

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

① NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock. In the case where the single-clock mode has been selected as an option, the TMP87EP26 are placed in this mode after reset.

② IDLE1 mode

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

③ STOP1 mode

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

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

(2) **Dual-clock mode**

Both high-frequency and low-frequency oscillation circuits are used in this mode. Pins P21 (XTIN) and P22 (XTOUT) cannot be used as input/output ports. The main system clock is obtained from the high-frequency clock in NORMAL2 and IDLE2 modes, and is obtained from the low-frequency clock in SLOW and SLEEP modes. The machine cycle time is 4/fc [s] (0.5 μs at fc = 8 MHz) in NORMAL2 and IDLE2 modes, and 4/fs [s] (122 μs at fs = 32.768 kHz) in SLOW and SLEEP modes. To use the dual-clock mode, the low-frequency oscillator should be turned on by executing [SET (SYSCR2).XTEN] instruction.

① NORMAL2 mode

In this mode, the CPU core operates using the high-frequency clock. On-chip peripherals operate using the high-frequency clock and/or low-frequency clock. In case that the dual-clock mode has been selected by an option, the TMP87EP26 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 operate using the low-frequency clock.

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

③ IDLE2 mode

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

④ SLEEP mode

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

⑤ STOP2 mode

As in STOP1 mode, all system operations are halted in this mode.
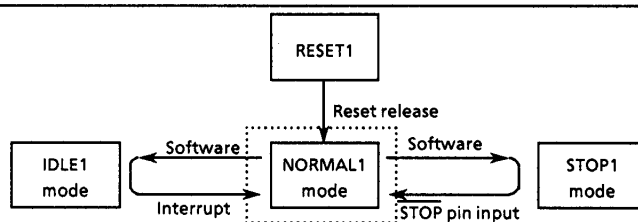
(a) Single-clock mode
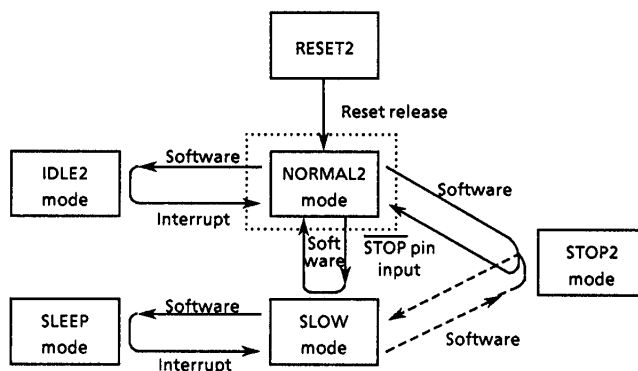
(b) Dual-clock mode

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

Note 2: The inside of dotted line operate E²PROM.

Operating mode		Frequency		CPU core	On-chip Peripherals	E²PROM		Machine cycle time
		High-frequency	Low-frequency			Program area	Data area	
Single-Clock	RESET1	Turning on oscillation	Turning off oscillation	Reset	Reset	halt		4/fc [s]
	NORMAL1			Operate	Operate	Operate		
	IDLE1			Halt		Halt	Halt	
	STOP1	Turning off oscillation			Halt			–
Dual-Clock	RESET2	Turning on oscillation	Turning on oscillation	Reset	Reset			4/fc [s]
	NORMAL2			High-frequency	Operate	Operate		
	IDLE2			Halt		Halt	Halt	
	SLOW	Turning off oscillation		Low-frequency	Low-frequency			4/fs [s]
	SLEEP			Halt				
	STOP2		Turning off oscillation		Halt			–

Figure 1-14. Operating Mode Transition Diagram

System Control Register 1

SYSCR1
(0038H)

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

(Initial value: 0000 00**)

STOP	STOP mode start	0: CPU core and peripherals remain active 1: CPU core and peripherals are halted (start STOP mode)	
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	R/W
OUTEN	Port output control during STOP mode	0: High-impedance 1: Remain unchanged	
WUT	Warming-up time at releasing STOP mode	00: 3×2^{19} / fc or 3×2^{13} / fs [s] 01: 2^{19} / fc or 2^{13} / fs 1*: Reserved	

Note 1: Always set RETM to "0" when transiting from NORMAL1 mode to STOP1 mode and from NOMAL2 mode to STOP2 mode. Always set RETM to "1" when transiting from SLOW mode to STOP2 mode.

Note 2: When STOP mode is released with $\overline{\text{RESET}}$ pin input, a return is made to NORMAL mode regardless of the RETM contents.

Note 3: fc ; high-frequency clock [Hz]
      fs ; low-frequency clock [Hz]
      * ; Don't care

Note 4: Bits 1 and 0 in SYSCR1 are read in as undefined data when a read instruction is executed.

Note 5: When the STOP mode is started by specifying OUTEN = "0", the internal input of port is fixed to "0" and the interrupt of the falling edge may be set.

System Control Register 2

SYSCR2
(0039H)

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

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

XEN	High-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation	
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	R/W
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)

Note 6: Do not use E^2PROM, when SYSCK is "1" or IDLE is "1".

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 $\overline{STOP}$ pin input. The $\overline{STOP}$ pin is also used both as a port P20 and an $\overline{INT5}$ (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 (except for DBR) and port output latches are all held in the status in effect before STOP mode was entered. The port output can be select either output hold or high-impedance by setting OUTEN (bit 4 in SYSCR1).

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

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

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

<u>a. Level-sensitive release mode (RELM = 1)</u>

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

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

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

Example: Starting STOP mode with an INT5 interrupt.

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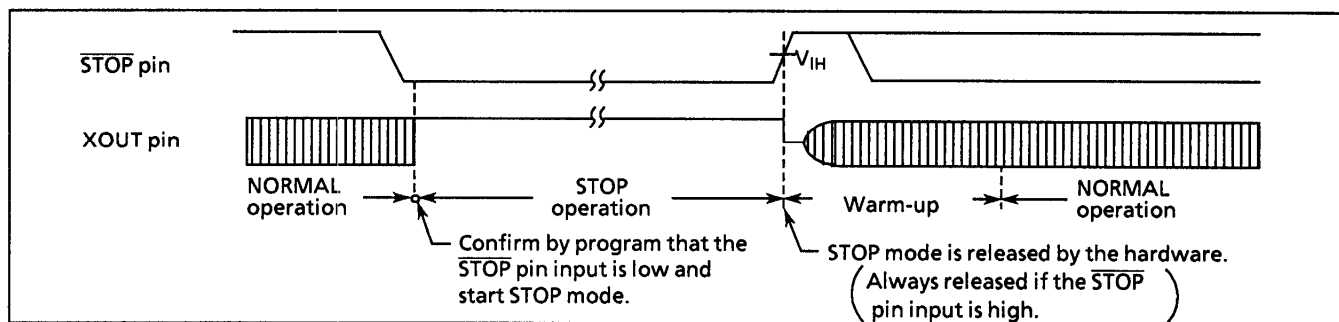

Figure 1-16. Level-sensitive Release Mode

Note 1: After warm-up start, even if $\overline{STOP}$ 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 $\overline{STOP}$ 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), 10000000B
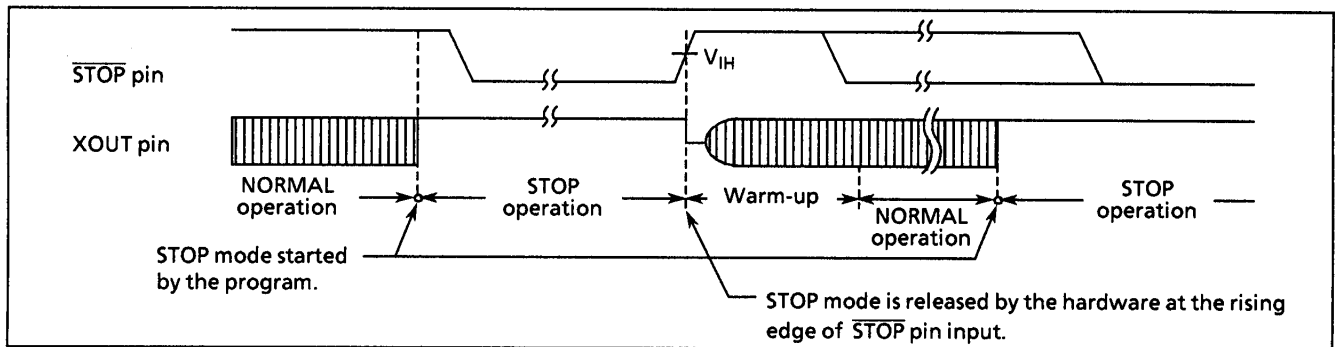
Figure 1-17. Edge-sensitive Release Mode

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 fc = 4.194304 MHz	At fc = 8 MHz	WUT	At fs = 32.768 kHz
$3 \times 2^{19} / fc$ [s]	375 [ms]	196.6 [ms]	$3 \times 2^{13} / fs$ [s]	750 [ms]
$2^{19} / fc$	125	65.5	$2^{13} / fs$	250

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

STOP mode can also be released by setting the $\overline{\text{RESET}}$ pin low, which immediately performs the normal reset operation. In this case, even if the setting is to return to the SLOW mode, it starts from the NORMAL mode. After reset, if the initial XTEN (bit 6 in SYSCR2) is set to "1" by mask option, TMP87EP26 starts from NORMAL2 mode.
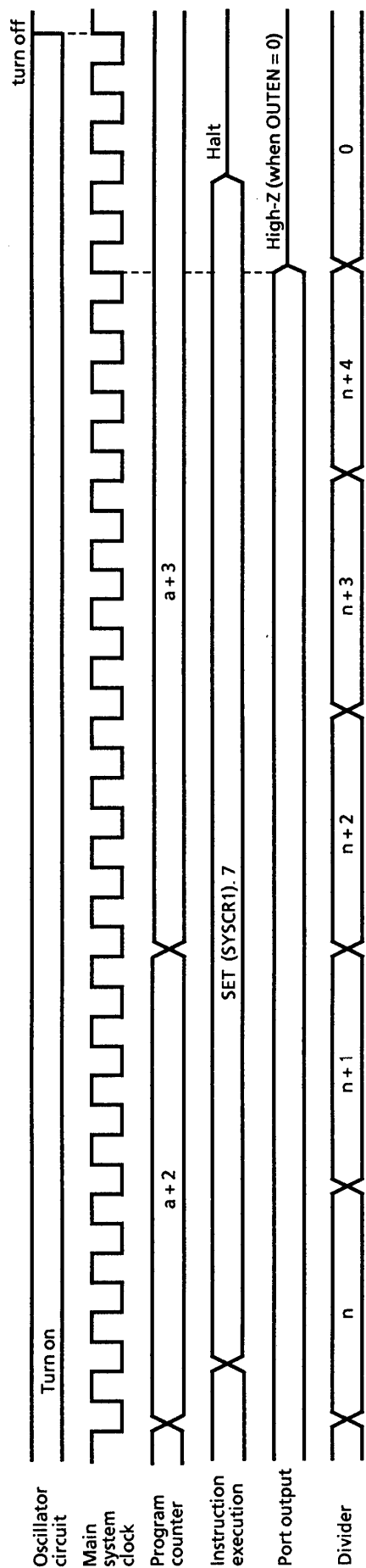
(a) STOP Mode Start (Example : Start with SET (SYSCR1). 7 instruction located at address a)
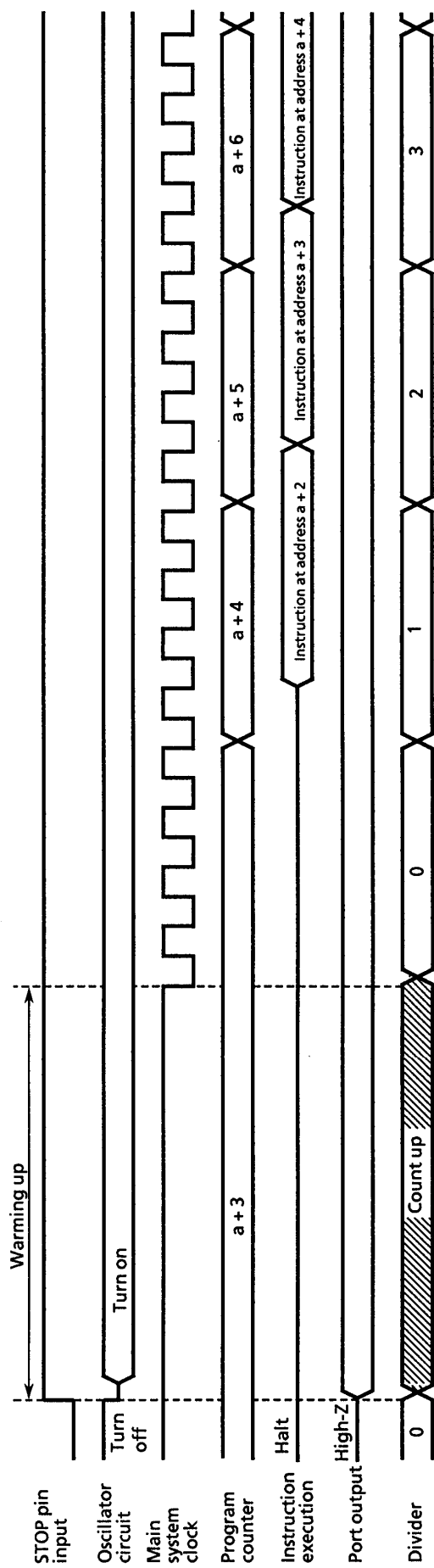
(b) STOP-Mode Release

Figure 1-18. STOP Mode Start / Release

Note: When STOP mode is released with a low hold voltage, the following cautions must be observed. The power supply voltage must be at the operating voltage level before releasing STOP mode. The $\overline{RESET}$ 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 $\overline{RESET}$ 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 $\overline{RESET}$ pin drops below the non-inverting high-level input voltage (hysteresis input).

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

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

① Operation of the CPU and watchdog timer is halted. On-chip peripherals continue to operate.

② The data memory, CPU registers and port output latches are all held in the status in effect before IDLE mode was entered.

③ The program counter holds the address of the instruction following the instruction which started IDLE mode.

Example: Starting IDLE mode.
        SET (SYSCR2) . 4

IDLE 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 ($\overline{INT0}$ pin) request. Execution resumes with the instruction following the IDLE mode start instruction. The interrupt latch (IL) of the interrupt source for releasing the IDLE mode must be cleared to "0" by load instruction.

Figure 1-19. IDLE Mode

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 ($\overline{INT0}$ 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 $\overline{RESET}$ pin low, which immediately performs the reset operation. After reset, if the initial XTEN (bit 6 in SYSCR2) is set to "1" by mask option, TMP87EP26 starts from NORMAL2 mode.

Note: When a watchdog timer interrupt is generated immediately before the IDLE mode is started, the watchdog timer interrupt will be processed but IDLE mode will not be started.
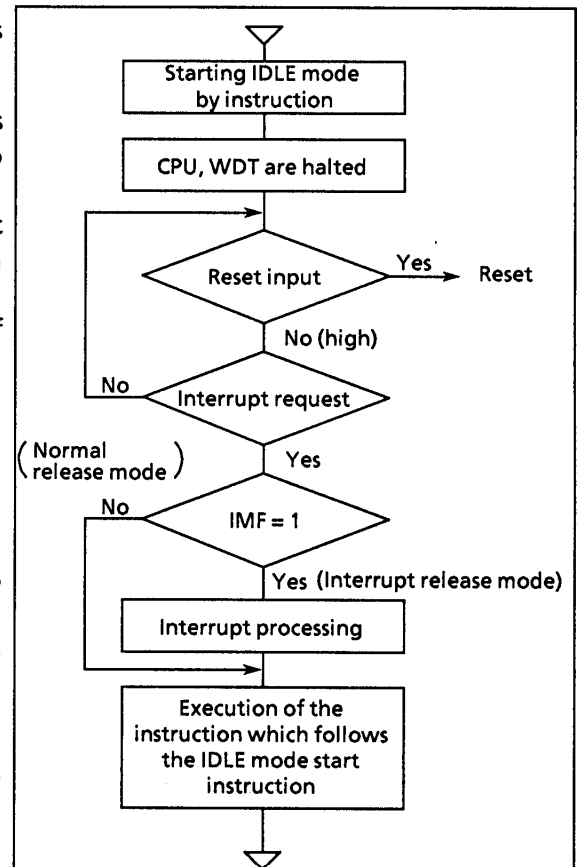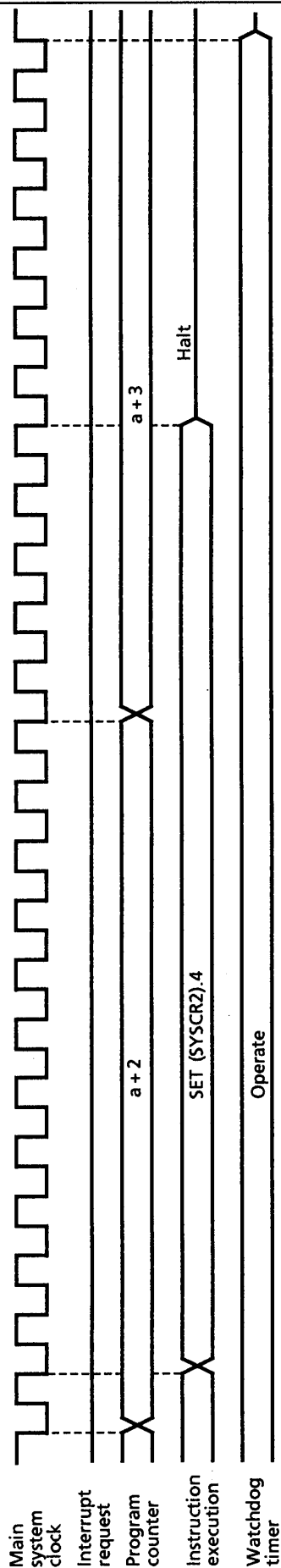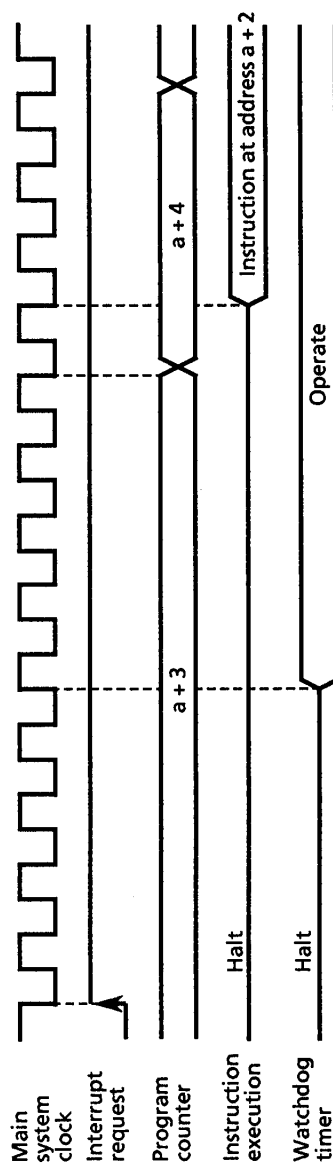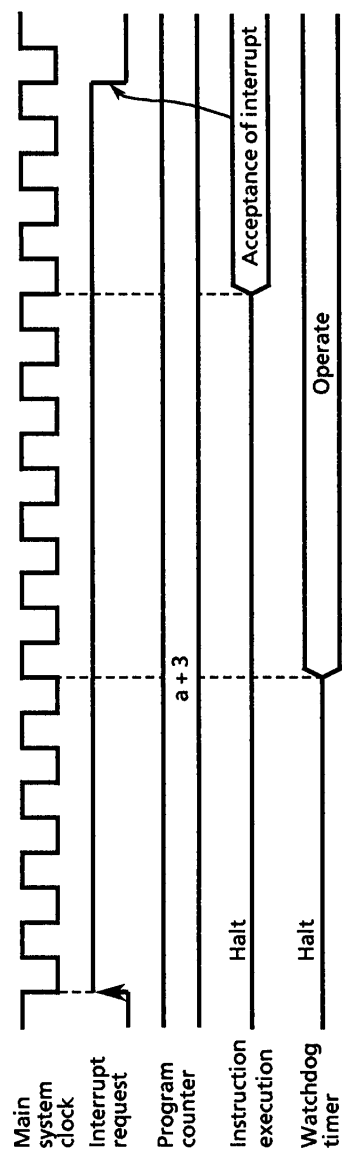
Figure 1-20. IDLE Mode Start / Release

(3) **SLOW** mode

SLOW mode is controlled by the system control register 2 (SYSCR2) and the timer/counter 2 (TC2).

a. Switching from NORMAL2 mode to SLOW mode

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

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

> *Note: 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.*

Example1: Switching from NORMAL2 mode to SLOW mode.

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

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

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

b. Switching from SLOW mode to NORMAL2 mode

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

> *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 $\overline{RESET}$ pin low, which immediately performs the reset operation. After reset, when an initial value of XTEN (bit 6 in SYSCR2) is set to "1" by mask option, TMP87EP26 is placed in NORMAL2 mode. When the initial value of XTEN is set to "0" by mask option, TMP87EP26 is placed in NORMAL1 mode.*

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

```
                SET     (SYSCR2) . 7      ; XEN←1  (turns on high-frequency oscillation)
                LD      (TC2CR), 10H      ; Sets TC2 mode
                                            (timer mode, source clock: fc)
                LD      (TREG2 + 1), 0F8H ; Sets the warming-up time
                                            (according to frequency and resonator
                                            characteristics)
                SET     (EIRH). EF14      ; Enable INTTC2
                LD      (TC2CR), 30H      ; Starts TC2
                :
PINTTC2:        LD      (TC2CR), 10H      ; Stops TC2
                CLR     (SYSCR2) . 5      ; SYSCK←0  (Switches the main system clock to the
                                            high-frequency clcok)
                RETI
                :
VINTTC2:        DW      PINTTC2           ; INTTC2 vector table
```
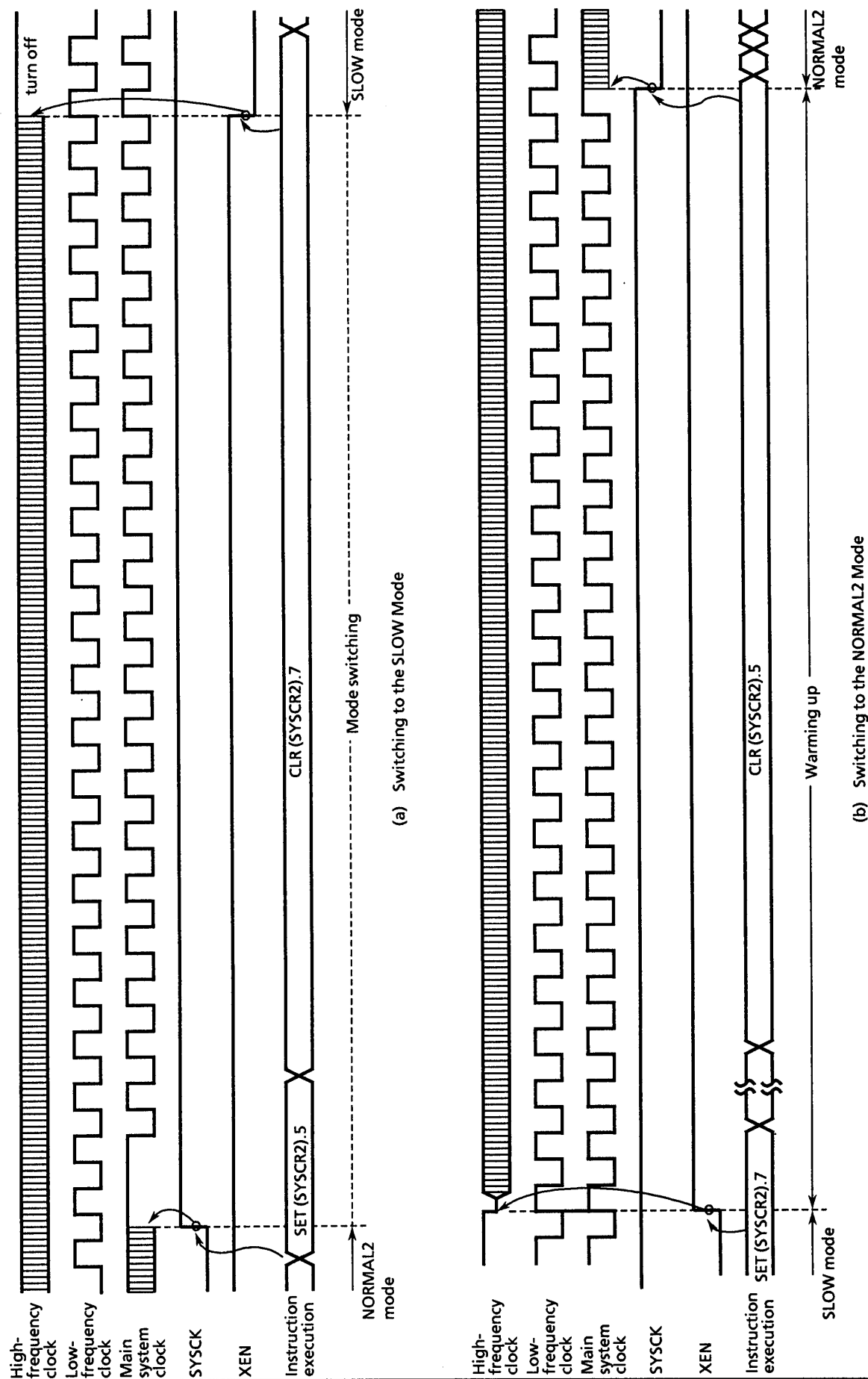
Figure 1-21. Switching between the NORMAL2 and SLOW Modes

1.9 Interrupt Controller

The TMP87EP26 has a total of 15 interrupt sources: 5 externals and 10 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	—	$FFFC_H$		1
Internal	INTWDT (Watchdog Timer interrupt)	non-maskable	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)	$IMF \cdot EF_4 = 1$	IL_4	$FFF6_H$		4
External	INT1 (External interrupt 2)	$IMF \cdot EF_5 = 1$	IL_5	$FFF4_H$		5
Internal	INTTBT (Time Base Timer interrupt)	$IMF \cdot EF_6 = 1$	IL_6	$FFF2_H$		6
External	INT2 (External interrupt 2)	$IMF \cdot EF_7 = 1$	IL_7	$FFF0_H$		7
Internal	INTTC3 (8-bit TC3 interrupt)	$IMF \cdot EF_8 = 1$	IL_8	$FFEE_H$		8
Internal	INTSIO1 (Serial Interface 1 interrupt)	$IMF \cdot EF_9 = 1$	IL_9	$FFEC_H$		9
Internal	INTTC5 (8-bit TC5 interrupt)	$IMF \cdot EF_{10} = 1$	IL_{10}	$FFEA_H$		10
External	INT3 (External interrupt 3)	$IMF \cdot EF_{11} = 1$	IL_{11}	$FFE8_H$		11
Internal	INTEEP (E^2PROM write-end interrupt)	$IMF \cdot EF_{12} = 1$	IL_{12}	$FFE6_H$		12
Internal	INTSIO2 (Serial Interface 2 interrupt)	$IMF \cdot EF_{13} = 1$	IL_{13}	$FFE4_H$		13
Internal	INTTC2 (16-bit TC2 interrupt)	$IMF \cdot EF_{14} = 1$	IL_{14}	$FFE2_H$		14
External	INT5 (External interrupt 5)	$IMF \cdot EF_{15} = 1$	IL_{15}	$FFE0_H$	Low	15

(1) Interrupt Latches (IL_{15} to IL_2)

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

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

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

Example 1: Clears interrupt latches

```
        DI                              ; IMF←"0"
        LDW     (IL), 1110100000111111B  ; IL12, IL10 to IL6←"0"
        EI                              ; IMF←"1"
```
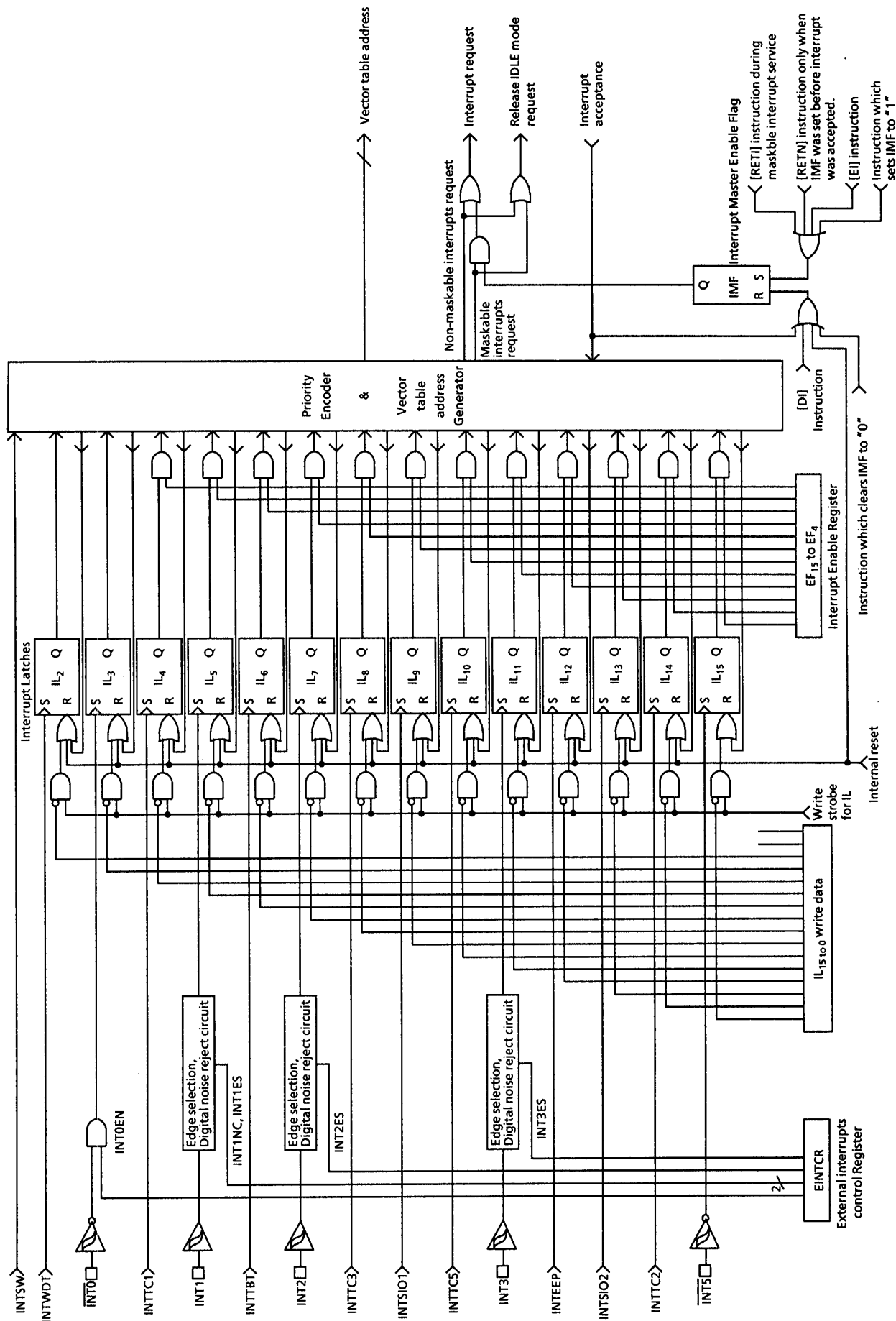
Figure 1-22. Interrupt Controller Block Diagram

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

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

① Interrupt Master enable Flag (IMF)

The interrupt master enable flag (IMF) enables and disables the acceptance of all interrupts, except for pseudo non-maskable interrupts. Clearing this flag to "0" disables the acceptance of all maskable interrupts. Setting to "1" enables the acceptance of interrupts. When an interrupt is accepted, this flag is cleared to "0" to temporarily disable the acceptance of maskable interrupts. After execution of the interrupt service program, this flag is set to "1" by the maskable interrupt return instruction [RETI] to again enable the acceptance of interrupts. If an interrupt request has already been occurred, interrupt service starts immediately after execution of the [RETI] instruction.

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

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

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".
```
         DI                                   ; IMF←"0"
         LDW    (EIR), 1110100010100001B      ; EF15 to EF13, EF11, EF7, EF5, IMF←"1"
         EI                                   ; IMF←"1"
```
Example 2: Sets an individual interrupt enable flag to "1".
```
         DI                                   ; IMF←"0"
         SET    (EIRH).4                      ; EF12←"1"
         EI                                   ; IMF←"1"
```


	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IL (003C, 003D$_H$)	IL$_{15}$	IL$_{14}$	IL$_{13}$	IL$_{12}$	IL$_{11}$	IL$_{10}$	IL$_9$	IL$_8$	IL$_7$	IL$_6$	IL$_5$	IL$_4$	IL$_3$	IL$_2$		

IL$_H$ (003D$_H$)   IL$_L$ (003C$_H$)
(Initial Value: 00000000 000000**)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIR (003A, 003B$_H$)	EF$_{15}$	EF$_{14}$	EF$_{13}$	EF$_{12}$	EF$_{11}$	EF$_{10}$	EF$_9$	EF$_8$	EF$_7$	EF$_6$	EF$_5$	EF$_4$				IMF

EIR$_H$ (003B$_H$)   EIR$_L$ (003A$_H$)
(Initial Value: 00000000 0000***0)

Note1: Do not use any read-modify-write instruction such as bit manipulation for clearing IL.
Note2: Do not clear IL$_2$ to "0" by an instruction.
Note3: Do not set IMF to "1" during non-maskable interrupt service program.
Note4: When manipulating IL or EF, clear IMF (to disable interrupts) beforehand.

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

1.9.1 Interrupt Sequence

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

(1) Interrupt acceptance processing

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

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

③ The contents of the program counter (PC) and the program status word (PSW) are saved (pushed) onto the stack. (pushed down in order of PSW, PC_H, PC_L). The stack pointer (SP) is three decrements.

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

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

Note1: a ; return address, b ; entry address, c ; address when the RETI instruction is stored
Note2: The maximum response time from when an IL is set until an interrupt acceptance processing starts is 38/fc or 38/fs [s]. It equals to setting the IL on the first machine cycle in 10 cycles instruction execution.
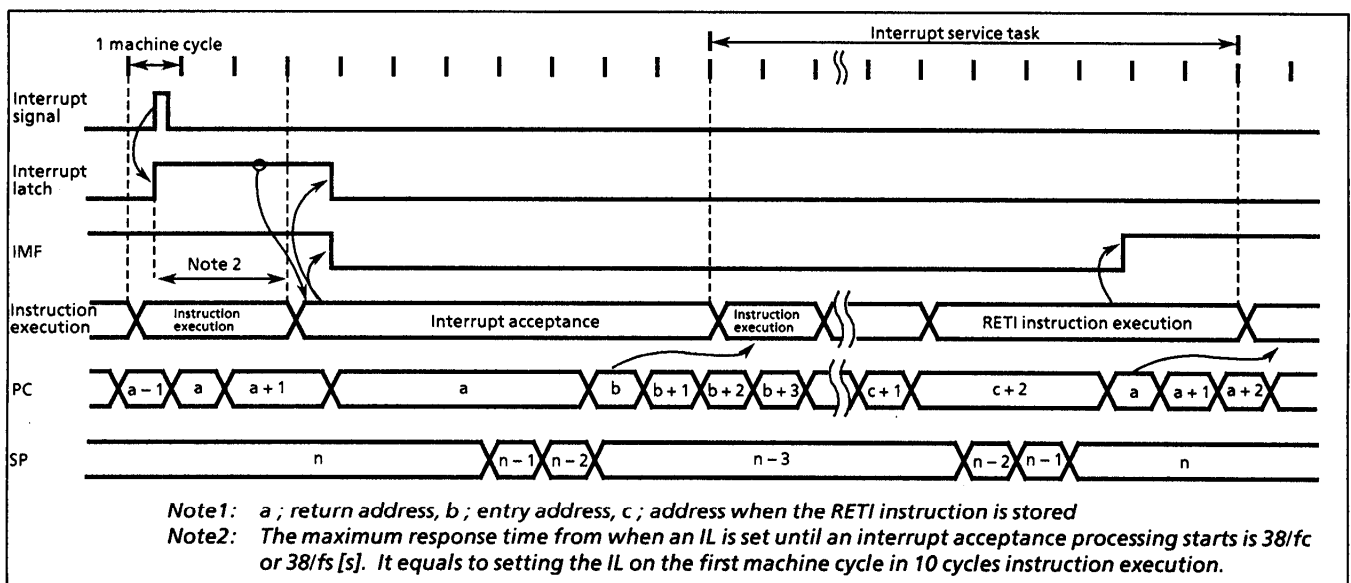
Figure 1-24. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

Example: Correspondence between vector table address for INTTBT and the entry address of the interrupt service program.
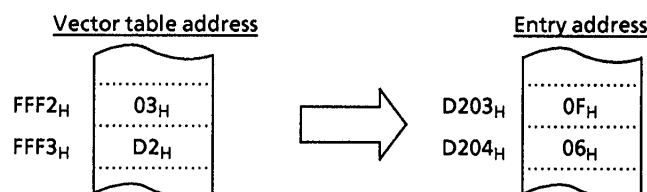
A maskable interrupt is not accepted until the IMF is set to "1" even if a maskable interrupt of higher priority than that of the current interrupt being serviced.

When nested interrupt service is necessary, the IMF is set to "1" in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags. However, an acceptance of external interrupt 0 cannot be disabled by the EF; therefore, if disablement is necessary, either the external interrupt function must be disabled the external interrupt control register (INTOEN) or interrupt processing must be avoided by the program. (When INTOEN = 0, the interrupt latch IL_3 is not set, therefore, the falling edge of the INT0 pin input cannot be detected.)

> Example 1: Disables an external interrupt 0 using INTOEN:
>
>   CLR   (EINTCR),INTOEN ; 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   ; Returns without interrupt processing if $(00F0_H)_0 = 1$
>
>   JRS   T, SINTO
>
>   RETI
>
> SINTO: Interrupt processing
>
>   RETI
>
> VINT0:   DW   PINT0

(2) **General-purpose register save/restore processing**

During interrupt acceptance processing, the program counter and the program status word are automatically saved on the stack, but not the accumulator and other registers. These registers are saved by the program if necessary. Also, when nesting multiple interrupt services, it is necessary to avoid using the same data memory area for saving registers.

The following method is used to save/restore the general-purpose registers:

① General-purpose register save/restore by register bank changeove:

General-purpose registers can be saved at high-speed by switching to a register bank that is not in use. Normally, bank 0 is used for the main task and banks 1 to 15 are assigned to interrupt service tasks. To increase the efficiency of data memory utilization, the same bank is assigned for interrupt sources which are not nested.

The switched bank is automatically restored by executing an interrupt return instruction [RETI] or [RETN]. Therefore, it is not necessary for a program to save the RBS.

> Example: Register Bank Changeover
>
> PINTxx: LD   RBS, n   ; Switches to bank n (1 μs at 8 MHz)
>
>   interrupt processing
>
>   RETI   ; Restores bank and Returns

(a) Saving/Restoring by register bank changeover   (b) Saving/Restoring using push/pop or data transfer instructions
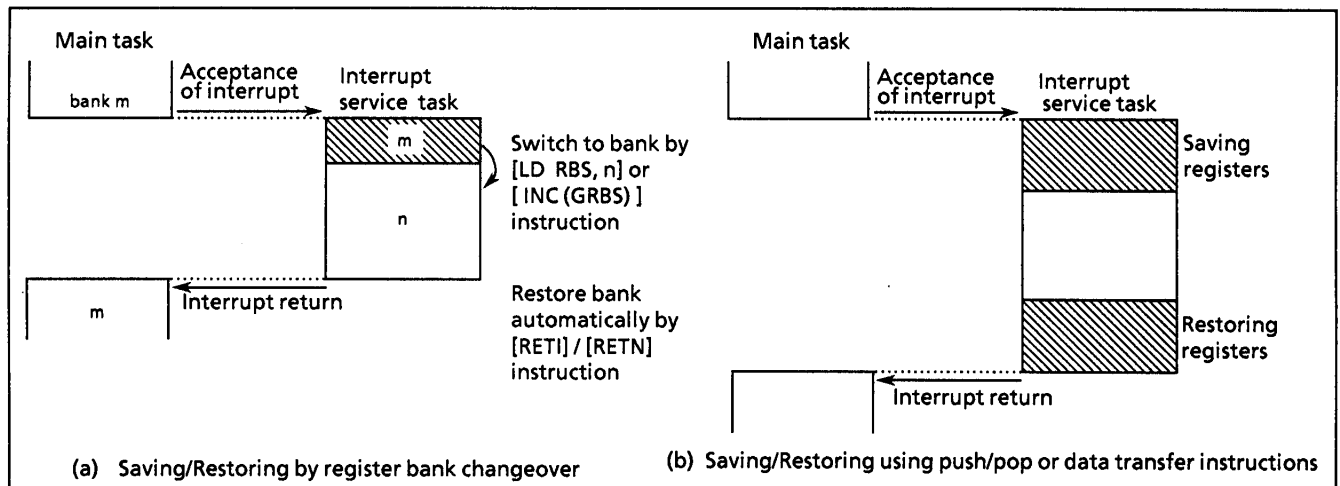
Figure 1-25. Saving/Restoring General-purpose Registers

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
          PUSH......HL..........; Save HL register pair
         :Interrupt processing :
          POP       HL          ; Restore HL register pair
          POP       WA          ; Restore WA register pair
          RETI                  ; Return
```
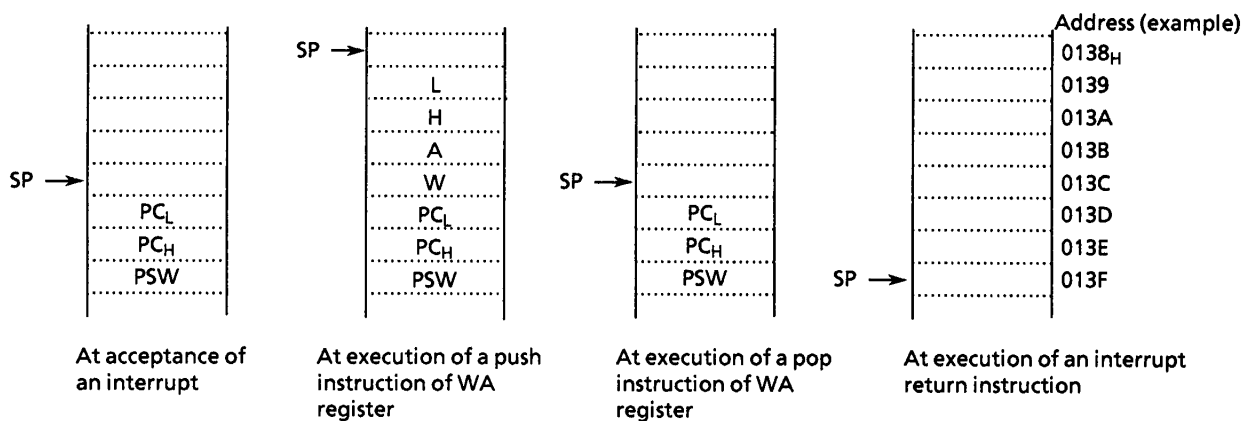
③ General-purpose registers save/restore using data transfer instruction:
Data transfer instructions can be used to save only a specific general-purpose register during processing of a single interrupt.

Example: Saving/restoring a register using data transfer instructions
    PINTxx: LD   (GSAVA), A   ; Save A register
                 :interrupt processing :
                 LD   A, (GSAVA)   ; Restore A register
                 RETI   ; Return

(3)   Interrupt return instructions
The interrupt return instructions [RETI] / [RETN] perform the following operations.

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

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

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

1.9.2 Software Interrupt (INTSW)

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

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

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

① Address Error Detection
FF_H is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address. Code FF_H is the SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing FF_H to unused areas of the program memory. Address trap reset is generated for instruction fetch from a part of RAM area (address 0040_H to $083F_H$) or SFR area (0000_H to $003F_H$).

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

1.9.3 External Interrupts

The TMP87EP26 has five external interrupts (INT0 to INT5: $\overline{INT0}$, INT1, INT2, INT3, $\overline{INT5}$). Three of these (INT1, INT2, INT3) have digital noise cancellation circuits (pulse inputs of less than a fixed time are cancelled as noise). Edge selection is possible with pins INT1, INT2, and INT3.

The $\overline{INT0}$/P10 pin can be selected either as an external interrupt input pin or as an I/O port. At reset, it is initialized as an input port.

Edge selection, noise cancellation control, and $\overline{INT0}$/P10 pin function selection are performed by the external interrupt control register (#0037H: EINTCR).

The both-edge detect function of the INT3 pin is selected by the external interrupt control register (#0037H:EINTCR and #001FH: EINT3).

Table 1-3 lists enable conditions, edge select, noise cancellation conditions. The following are notes on the usage of external interrupts:

Notes on usage of external interrupts:

Note 1: When INT0 to INT5 ($\overline{INT0}$, INT1, INT2, INT3, $\overline{INT5}$) are used in SLOW or SLEEP mode, the noise cancellation function is disabled. Noise cancellation time for a pulse input during operating mode transition is indeterminate.

Note 2: Input pulse width for $\overline{INT0}$ and $\overline{INT5}$ must be one machine cycle or more at both high and low levels.

$\overline{INT0}$, $\overline{INT5}$ input pulse

t_{INTL} t_{INTH}

$t_{INTL}, t_{INTH} > t_{cyc}$
$t_{cyc} = 4/fc$ [s] (at NORMAL 1/2 and IDLE 1/2 modes)
     $4/fs$ [s] (at SLOW and SLEEP modes)
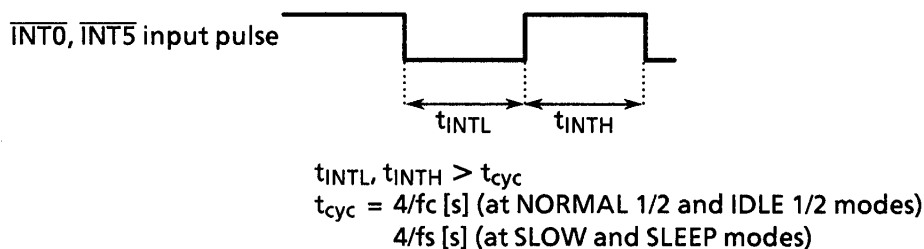
Note 3: If a signal without noise is input to the external interrupt pin in NORMAL 1/2 or IDLE 1/2 mode, the maximum times from input signal edge to input latch set are as described below:

① INT1 pin   49/fc [s] (when INT1NC = 1)
     193/fc [s] (when INT1NC = 0)

② INT2 pin   25/fc [s]
③ INT3 pin   25/fc [s] (when #0037H: INT3W = 0, falling or rising edge)
     25/fc [s] (when #0037H: INT3W = 1 and #001FH: NCS (0, 0, 0))
     $(2^6/fc) \times 8.5 + 19/fc$ [s] (when #0037H: INT3W = 1 and #001FH: NCS (0, 0, 1))
     $(2^7/fc) \times 8.5 + 19/fc$ [s] (when #0037H: INT3W = 1 and #001FH: NCS (0, 1, 0))
     $(2^8/fc) \times 8.5 + 19/fc$ [s] (when #0037H: INT3W = 1 and #001FH: NCS (0, 1, 1))
     $(2^9/fc) \times 8.5 + 19/fc$ [s] (when #0037H: INT3W = 1 and #001FH: NCS (1, 0, 0))
     $(2^{10}/fc) \times 8.5 + 19/fc$ [s] (when #0037H: INT3W = 1 and #001FH: NCS (1, 0, 1))
     $(2^{11}/fc) \times 8.5 + 19/fc$ [s] (when #0037H: INT3W = 1 and #001FH: NCS (1, 1, 0))
     $(2^{12}/fc) \times 8.5 + 19/fc$ [s] (when #0037H: INT3W = 1 and #001FH: NCS (1, 1, 1))

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

① TC1 pin: Less than 7/fc [s] (noise cancellation) and 24/fc [s] or more (pulse receive)
② TC3 pin: When INT3W = 0, less than 7/fc [s] (noise cancellation) and 24/fc [s] or more (pulse receive).
     For when INT3W = 1, see Table 1-4.

Note 5: When INT0EN = 0, interrupt latch IL3 is not set even if a falling edge is detected for $\overline{INT0}$ pin input.

Note 6: *Change EINTCR only when IMF = 0. After changing EINTCR, interrupt latches of external interrupt inputs must be cleared to "0" using load instruction.*

Example: Changes INT2 edge selection from rising edge to falling edge

DI		; IMF ← 0 (disables interrupt service)
LD	(EINTCR), 10000110B	; INT2ES ← 1 (changes edge selection)
LD	(ILL), 01111111B	; IL7 ← 0 (clears interrupt latch)
EI		; IMF ← 1 (enables interrupt service)

Note 7: *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.*

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

Note 9: *In order to change an edge of timer counter input by rewriting the contents of INT2ES and INT3ES during NORMAL1/2 mode, rewrite the contents after timer counter is stopped (TC*s = 0), that is, interrupt disable state. Then, clear interrupt laches of external interrupt inputs (INT2 and INT3) after 8 machine cycles from the time of rewriting to change to interrupt enable state. Finally, start timer counter. During SLOW mode, 3 machine cycles are required.*

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

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

Note 10: *When high-impedance is specified for port output in stop mode, port input is forcibly fixed to low level internally. Thus, interrupt latches of external interrupt inputs except $\overline{INT5}$ (P20/$\overline{STOP}$) which are also used as ports may be set to "1". To specify high-impedance for port output in stop mode, first disable interrupt service (IMF = 0), activate stop mode. After releasing stop mode, clear interrupt latches using load instruction, then, enable interrupt service.*

Example: Activating stop mode

LD	(SYSCR1), 01000000B	; OUTEN ← 0 (specifies high-impedance)
DI		; IMF ← 0 (disables interrupt service)
SET	(SYSCR1). STOP	; STOP ← 1 (activates stop mode)
LDW	(IL), 1111011101010111B	; IL11, 7, 5, 3 ← 0 (clears interrupt latches)
EI		; IMF ← 1 (enables interrupt service)

Table 1-3 (a). External Interrupts

SOURCE	Pin	Secondary function	Enable Condition	Edge			Digital noise reject
				Rising	Falling	Both	
INT0	INT0̄	P10	IMF = 1, INT0EN = 1	—	○	—	— (hysteresis input)
INT1	INT1	P11	IMF · EF$_5$ = 1	INT1ES = 0	INT1ES = 1	—	Pulses less than 15/fc [s] or 63/fc [s] are cancelled as noise. Pulses equal to or more than 48/fc [s] or 192/fc [s] are regarded as signals.
INT2	INT2	P12 / TC1	IMF · EF$_7$ = 1	INT2ES = 0	INT2ES = 1	—	Pulses less than 7/fc [s] are cancelled as noise. Pulses equal to or more than 24/fc [s] are regarded as signals. Same applies to pin TC1.
INT3	INT3	P40 / TC3	IMF · EF$_{11}$ = 1, INT3W = 0	INT3ES = 0	INT3ES = 1	—	For falling or rising edge, pulses less than 7/fc [s] are cancelled as noise. Pulses equal to or more than 24/fc [s] are regarded as signals. Same applies to pin TC3 (at one edge).
			IMF · EF$_{11}$ = 1, INT3W = 1	—	—	INT3W = 1 Note)	Noise cancellation conditions are as listed in Table 1-3 (b). They are applied to the INT3 pin when it is used for both-edge interrupts.
INT5	INT5̄	P20 / STOP̄	IMF · EF$_{15}$ = 1	—	○	—	— (hysteresis input)

Note: To detect the edge at which an interrupt is generated, read bit 7 (INTEDT) in EINT3CR (#001F$_H$), that is, at the beginning of the interrupt processing routine.
INTEDT is valid only for both-edge interrupts (INT3W = 1). INTEDT is set to 1 by an interrupt as the non-selected edge; cleared to 0 after read automatically.
For both-edge interrupts, rising or falling edge is selected by setting/modifying bit 3 (INT3ES) in EINTCR (#0037$_H$).
When rising edge is selected (INT3ES = 0), bit 7 in INTEDT (#001F$_H$) is set to 1 when a falling edge is detected at the INT3 pin. (That is, remains 0 if rising edge is detected.)
When falling edge is selected (INT3ES = 1), bit 7 in INTEDT: #001F$_H$ is set to 1 when a rising edge is detected at the INT3 pin. (That is, remains 0 at falling edge.)

Table 1-3 (b). Noise reject condition for INT3 (both-edge interrupt)

#001FH			Max. pulse width for noise reject	Min. pulse width for immediate signal
NCS2	NCS1	NCS0		
0	0	0	- (histeresis input)	
0	0	1	$(2^6/fc) \times 7 - 6/fc$	$(2^6/fc) \times 8 + 5/fc$
0	1	0	$(2^7/fc) \times 7 - 6/fc$	$(2^7/fc) \times 8 + 5/fc$
0	1	1	$(2^8/fc) \times 7 - 6/fc$	$(2^8/fc) \times 8 + 5/fc$
1	0	0	$(2^9/fc) \times 7 - 6/fc$	$(2^9/fc) \times 8 + 5/fc$
1	0	1	$(2^{10}/fc) \times 7 - 6/fc$	$(2^{10}/fc) \times 8 + 5/fc$
1	1	0	$(2^{11}/fc) \times 7 - 6/fc$	$(2^{11}/fc) \times 8 + 5/fc$
1	1	1	$(2^{12}/fc) \times 7 - 6/fc$	$(2^{12}/fc) \times 8 + 5/fc$

Note: In SLOW mode, set (NCS2,1,0) = (0, 0, 0).
In SLOW mode, the digital noise filter in the above table is disabled.

External interrupt Control Register 1

EINTCR
(0037H)

7	6	5	4	3	2	1	0
INT1 NC	INT0 EN			INT3 ES	INT2 ES	INT1 ES	INT3W

(Initial value: 00** 0000)

INT1NC	Noise reject time select	0: Pulses of less than 63 / fc [s] are eliminated as noise 1: Pulses of less than 15 / fc [s] are eliminated as noise	R/W
INT0EN	P10 / $\overline{INT0}$ pin configuration	0: P10 input/output 1: $\overline{INT0}$ pin (port P10 should be set to an input mode)	
INT3ES INT2ES INT1ES	INT3 to INT1 edge select	0: Rising edge 1: Falling edge	
INT3W	INT3 both edge selection	0: Refer to INT3ES 1: Both edge detection	

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

Figure 1-26 (a). External Interrupt Control Register

External interrupt Control Register 2

EINT3CR
(001F$_H$)

7	6	5	4	3	2	1	0
INT EDT		NCS		INT3 DET			

(initial value: 0000 0***)

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

Note: INTEDT and NCS are valid only when the INT3W bit in EINTCR (#0037$_H$) is set to 1.
Therefore, when INT3W = 0, the digital noise filter set by the NCS bit is disabled.

Figure 1-26 (b). External Interrupt Control Register 2
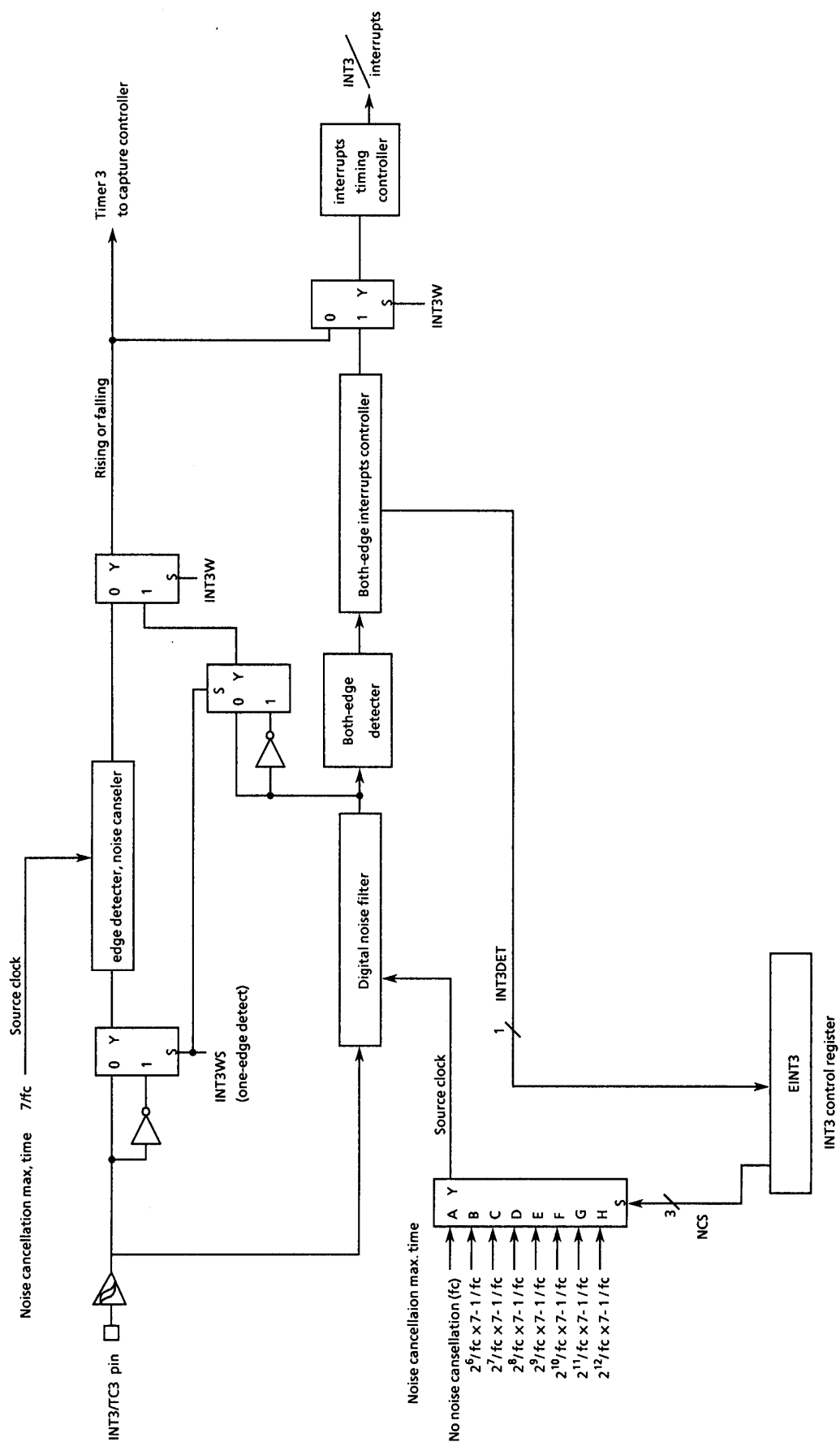
Figure1-26 (c). Bother One Edge Detictor of INT3/TC3 Pin

Notes on the usage of INT3 pin (external interrupt)

1. In the case of using the INT3 pin for one edge (either rising or falling).

> *Note: In order to set/rewrite external interrupt control register(EINTCR), set / rewirte external interrupt register in the interrupt disable state (IMF = 0). Then, enable interrupt acceptance after interrupt latch cleared.*

2. In the case of using the INT3 pin for both edge (rising and falling).

> *Note 1: When using the INT3 pin for both edges (rising and falling), set bit 0 (INT3W) in EINTCR ($\#0037_H$) to 1.*
>
> *Note 2: To detect the edge at which an interrupt is generated, read bit 7 (INTEDT) in EINT3CR ($\#001F_H$), that is, at the beginning of the interrupt processing routine.*
>
> *Note 3: INTEDT is valid only for both-edge interrupts (INT3W = 1). INTEDT is set to 1 by an interrupt as the non-selected edge; cleared to 0 after read automatically.*
> *When rising edge is selected (INT3ES = 0), bit 7 in INTEDT ($\#001F_H$) is set to 1 when a falling edge is detected at the INT3 pin. (That is, remains 0 if rising edge is detected.)*
> *When falling edge is selected (INT3ES = 1), bit 7 in INTEDT: $\#001F_H$ is set to 1 when a rising edge is detected at the INT3 pin. (That is, remains 0 at falling edge.)*
>
> *Note 4: In order to set/rewrite external interrupt control register(EINTCR), set / rewirte external interrupt register in the interrupt disable state (IMF = 0). Then, enable interrupt acceptance after interrupt latch cleared.*

<u>Operation description for INT3 (both-edge interrupt) in use:</u>

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

   For both-edge interrupts, rising edge is selected (INT3ES = 0) and fixed.

   1) Case1: When the initial state of the INT3 pin is high after reset:
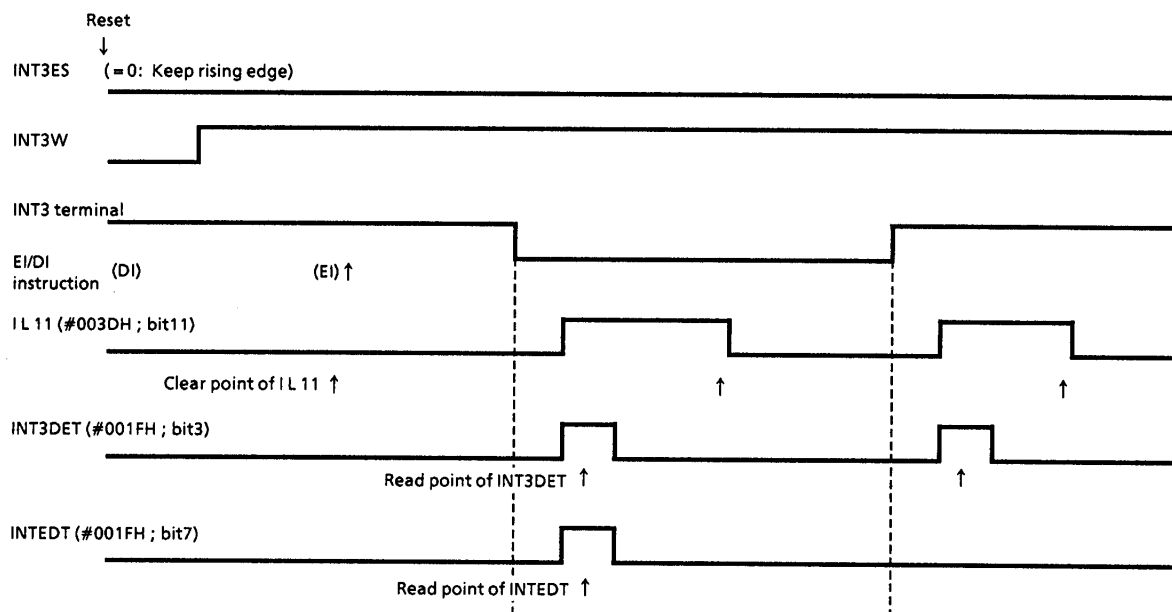
   2) Case2: When the initial state of the INT3 pin is low after reset:
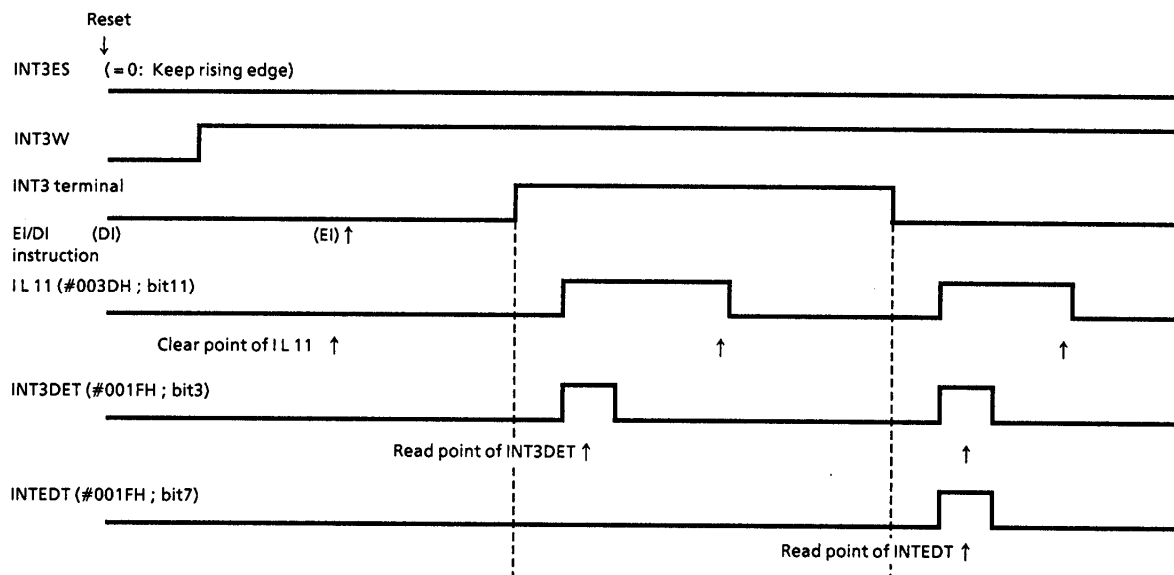
2. Operation with setting/modifying external interrupt control register (EINTCR) after reset:

    1) Case3: When the initial state of the INT3 pin is low after reset/low at edge switchover from rising to falling:
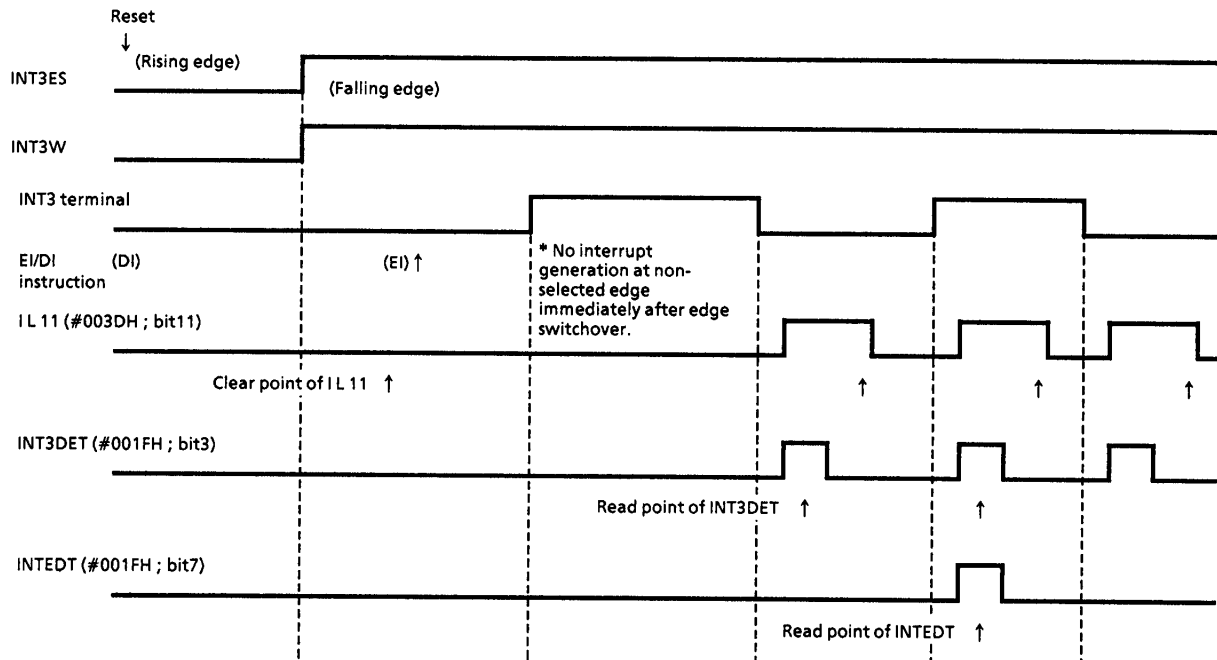
    2) Case4: When the initial state of the INT3 pin is high after reset/high at edge switchover from rising to falling:
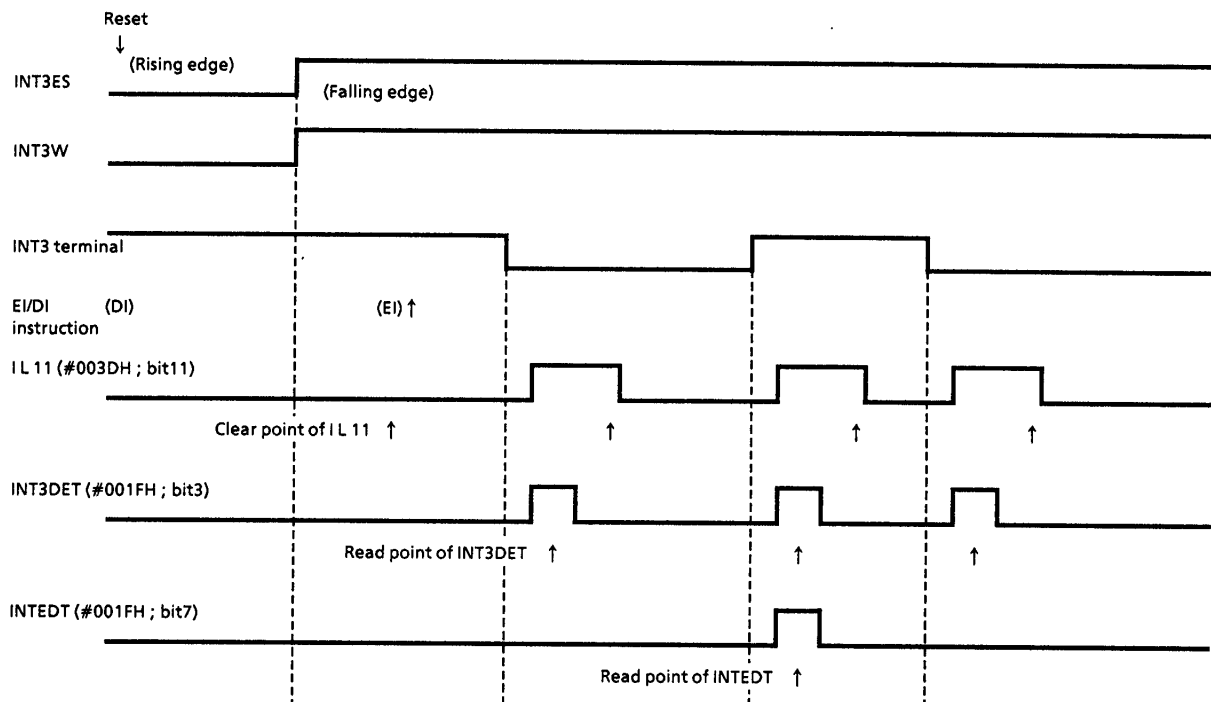
3) Case5:　　Case 5: When the initial state of the INT3 pin is high after reset/low at edge switchover from rising to falling:
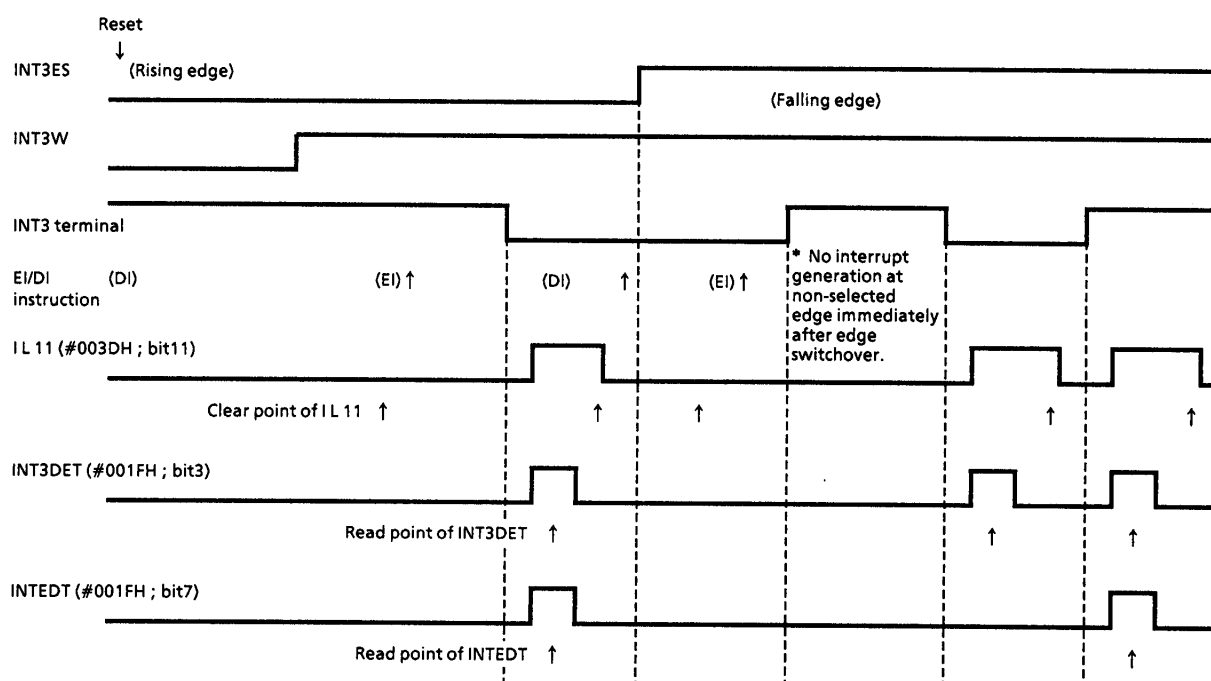
4) Case6:　　When the initial state of the INT3 pin is low after reset/high at edge switchover from rising to falling:
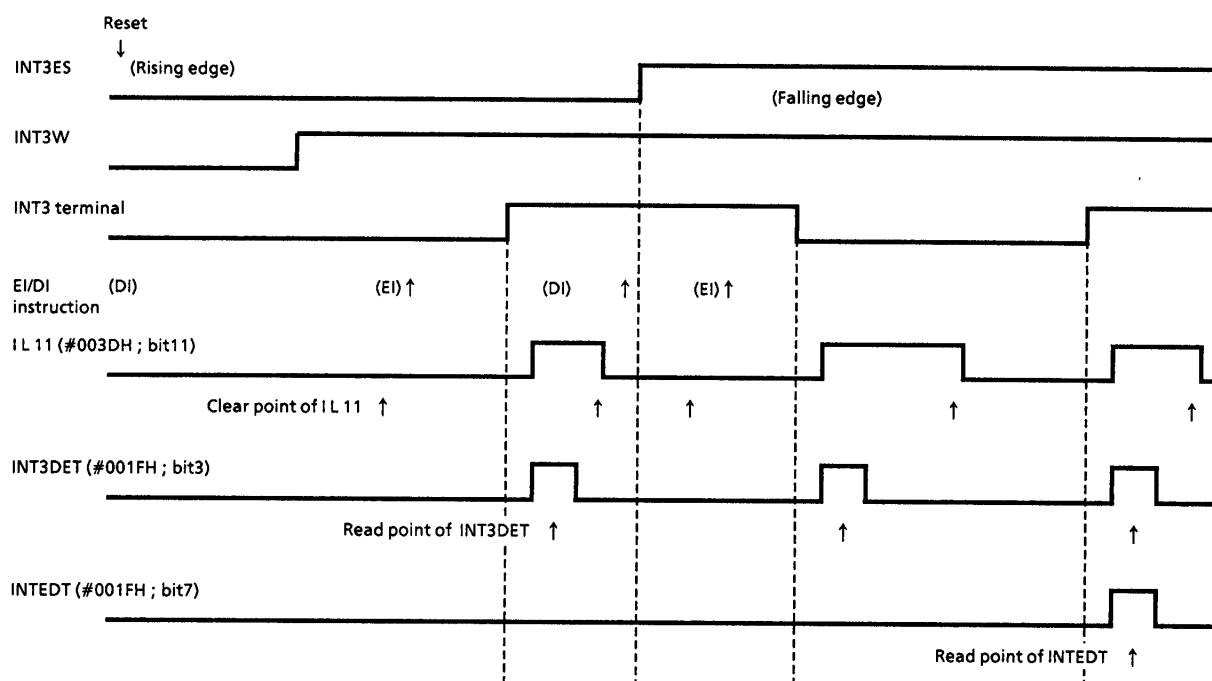
1.10 Watchdog Timer (WDT)

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

The watchdog timer signal for detecting malfunction can be selected either 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.

> Note: Care must be given in system design so as to protect the Watchdog Timer from disturbing noise. Otherwise the Watchdog Timer may not fully exhibit its functionality.

1.10.1 Watchdog Timer Configuration
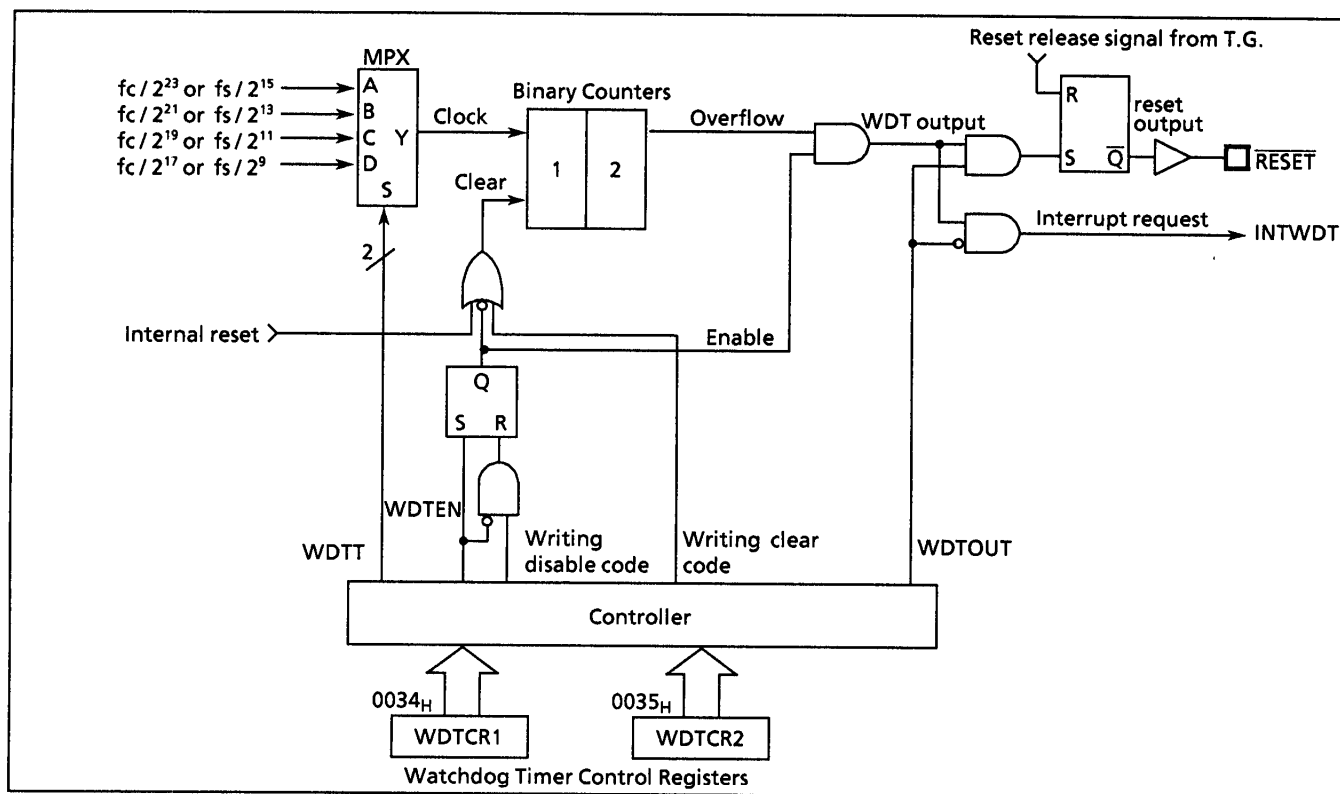
Figure 1-27. Watchdog Timer Configuration

1.10.2 Watchdog Timer Control

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

(1) Malfunction detection methods using the watchdog timer

The CPU malfunction is detected as follows.

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

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

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

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

> Note: The watchdog timer consists of an internal divider and a two-stage binary counter. When clear code $4E_H$ is written, only the binary counter is cleared, not the internal divider. Depending on the timing at which clear code $4E_H$ is written on the WDTCR2 register, the overflow time of the binary counter may be at minimum 3/4 of the time set in WDTCR1 <WDTT>. Thus, write the clear code using a shorter cycle than 3/4 of the time set in WDTCR1 <WDTT>.

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

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

Watchdog Timer Control Register 1

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

WDTEN	Watchdog timer enable/disable	0: Disable (It is necessary to write the disable code to WDTCR2)	Write-only
WDTT	Watchdog timer detection time	00: $2^{25}/fc$ or $2^{17}/fs$ [s] 01: $2^{23}/fc$ or $2^{15}/fs$ 10: $2^{21}/fc$ or $2^{13}/fs$ 11: $2^{19}/fc$ or $2^{11}/fs$	
WDTOUT	Watchdog timer output select	0: Interrupt request 1: Reset output	

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

Watchdog Timer Control Register 2

WDTCR2 (0035$_H$)	7 6 5 4 3 2 1 0	(Initial value: **** ****)

WDTCR2	Watchdog timer control code write register	$4E_H$: Watchdog timer binary counter clear (clear code) $B1_H$: Watchdog timer disable (disable code) others: Invalid	Write-only

Note 1: The disable code is invalid unless written when WDTEN = 0.
Note 2: * ; Don't care
Note 3: Since WDTCR2 is a write-only register, read-modify-write instructions (e.g., bit manipulating instructions such as SET or CLR and arithmetic instructions such as AND or OR) cannot be used for read/write to this register.
Note 4: Write clear code $4E_H$ within 3/4 of the time set in WDTCR1<WDTT>.

Figure 1-28. Watchdog Timer Control Registers

Table 1-4. Watchdog Timer Detection Time

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

(2) Watchdog Timer Enable

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

Example: Enables watchdog timer
        LD   (WDTCR1), 00001000B ; WDTEN←1

(3) Watchdog Timer Disable

The watchdog timer is disabled by writing the disable code (B1$_H$) to WDTCR2 after clearing WDTEN (bit 3 in WDTCR1) to "0". The watchdog timer is not disabled if this procedure is reversed and the disable code is written to WDTCR2 before WDTEN is cleared to "0".
During disabling the watchdog timer, the binary counters are cleared to "0".

Example: Disables watchdog timer
        LDW   (WDTCR1) , 0B101H   ; WDTEN←0, WDTCR1←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, 043FH   ; 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 $\overline{RESET}$ pin low to reset the internal hardware. The reset output time is $2^{20}/fc$ [s] (131 ms at fc = 8 MHz). The $\overline{RESET}$ pin is sink open drain input / output with pull-up resistor.

> Note: The high-frequency clock oscillator also turns on when a watchdog timer reset is generated in SLOW mode. Thus, the reset output time is $2^{20}/fc$.
> The reset output time include a certain amount of error if there is any fluctuation of the oscillation frequency when the high-frequency clock oscillator turns on. Thus, the reset output time must be considered approximate value.
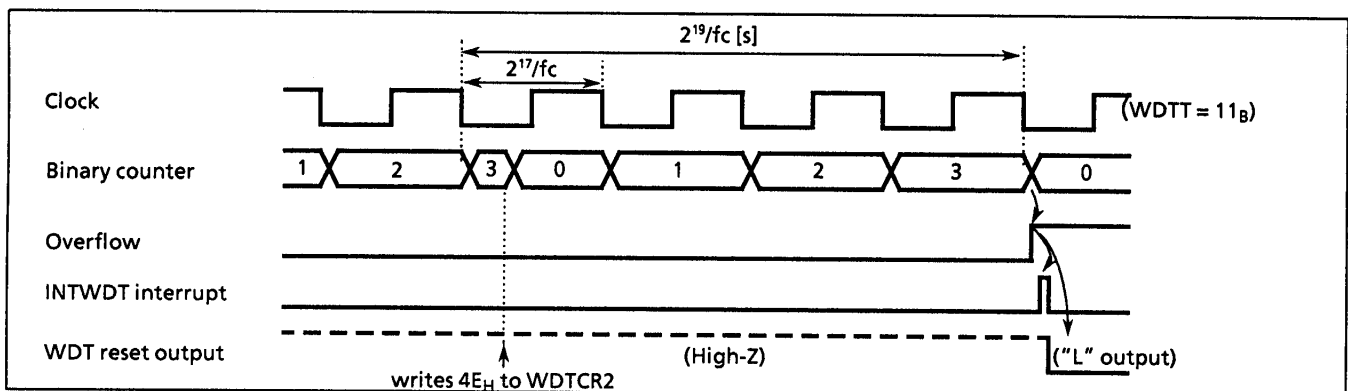
Figure 1-29. Watchdog Timer Interrupt / Reset

1.11 Reset Circuit

The TMP87EP26 has 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 $\overline{RESET}$ pin may go low ($2^{20}/fc$ [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)	$(FFFF_H) \cdot (FFFE_H)$	Divider of Timing generator	0
Register bank selector	(RBS)	0		
Jump status flag	(JF)	1	Watchdog timer	Enable
Interrupt master enable flag	(IMF)	0	Output latches of I/O ports	Refer to I/O port circuitry
Interrupt individual enable flags	(EF)	0		
Interrupt latches	(IL)	0	Control registers	Refer to each of control register

1.11.1 External Reset Input

When the $\overline{\text{RESET}}$ pin is held at low for at least 3 machine cycles (12/fc [s]) with the power supply voltage within the operating voltage range and oscillation stable, a reset is applied and the internal state is initialized.

When the $\overline{\text{RESET}}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses FFFE$_H$ - FFFF$_H$.

The $\overline{\text{RESET}}$ pin contains a Schmitt trigger (hysteresis) with an internal pull-up resistor. A simple power-on-reset can be applied by connecting an external capacitor and a diode.
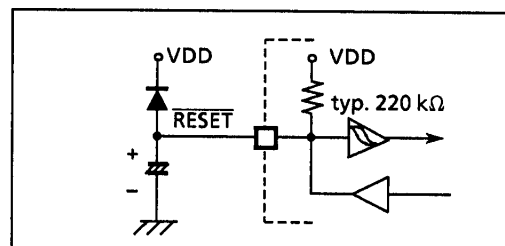
Figure 1-30. Simple Power-on-Reset Circuitry

1.11.2 Address-Trap-Reset

An address-trap-reset is one of fail-safe function that detects CPU malfunction such as endless looping caused by noise or the like, and returns the CPU to the normal state. If the CPU attempts to fetch an instruction from RAM or SFRs (address 0000$_H$ to 083F$_H$), an address-trap-reset will be generated. Then, the $\overline{\text{RESET}}$ pin output will go low. The reset time is 2^{20}/fc [s] (131 ms at 8 MHz) .

Note 1: $0 \leqq a \leqq 083F_H$
Note 2:   During reset release, reset vector "r" is read out, and an instruction at address r is fetched and decoded.
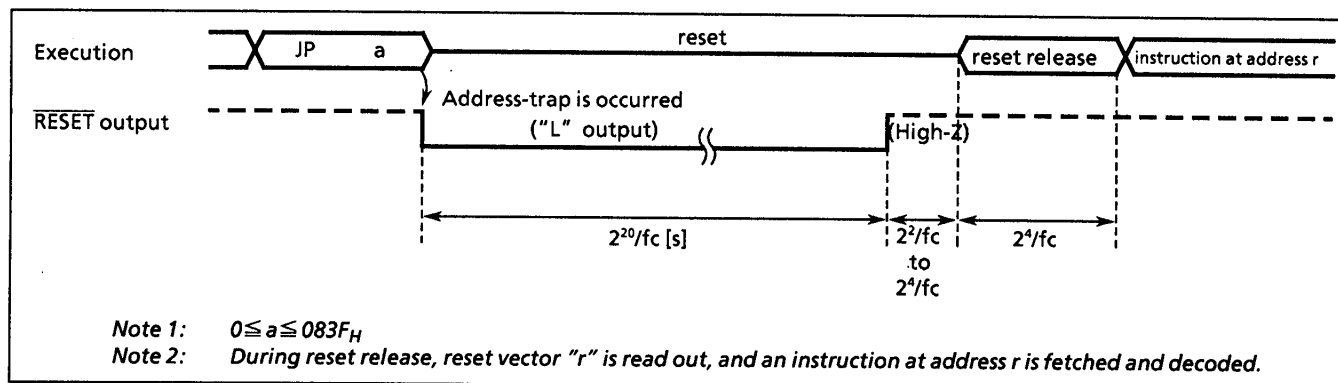
Figure 1-31. Address-Trap-Reset

1.11.3 Watchdog Timer Reset

Refer to Section "1.10 Watchdog Timer".

1.11.4 System-Clock-Reset

Clearing both XEN and XTEN (bits 7 and 6 in SYSCR2) to "0" stops both high-frequency and low-frequency oscillation, and causes the MCU to deadlock. This can be prevented by automatically generating a reset signal whenever XEN = XTEN = 0 is detected to continue the oscillation. Then, the $\overline{\text{RESET}}$ pin output goes low from high-impedance. The reset time is 2^{20}/fc [s] (131 ms at 8 MHz).

2. Peripheral Hardware 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 peripherals control and data transfers are performed through the special function registers (SFR) and data buffer registers (DBR).

The SFR are mapped to addresses 0000_H to $003F_H$ and the DBR to addresses $0F80_H$ to $0FFF_H$.

Figure 2-1 shows the TMP87EP26 SFRs and DBRs.

Address	Read	Write	Address	Read	Write
0000_H	P0 Port		0020_H	SIO1SR (SIO1 status)	SIO1CR1 (SIO1 control)
01	P1 Port		21	−	SIO1CR2
02	P2 Port		22	SIO2SR (SIO2 status)	SIO2CR1 (SIO2 control)
03	P3 Port		23	−	SIO2CR2
04	P4 Port		24	EEPADR1$_L$	(E^{2}PROM address register 1)
05	P5 Port		25	EEPADR1$_H$	
06	P6 Port		26	EEPDAT1	(E^{2}PROM data register 1)
07	P7 Port		27	EEPCR1	(E^{2}PROM control register 1)
08	P8 Port		28	−	LCDCR (LCD control)
09	P9 Port		29	−	P6CR (P6 I/O control)
0A	−	P0CR (P0 I/O control)	2A	−	P7CR (P7 I/O control)
0B	−	P1CR (P1 I/O control)	2B	−	P8CR (P8 I/O control)
0C	−	P4CR1 (P4 I/O control)	2C	−	P9CR (P9 I/O control)
0D	−	P5CR (P5 I/O control)	2D	−	P4CR2
0E	ADCCR (AD converter control)		2E	reserved	
0F	ADCDR (AD conv. result)	−	2F	reserved	
10	−	TREG1A$_L$ (Timer register 1A)	30	reserved	
11	−	TREG1A$_H$	31	reserved	
12	TREG1B$_L$ (Timer register 1B)		32	reserved	
13	TREG1B$_H$		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 (External 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	reserved		3B	EIR$_H$	
1C	PD port		3C	IL$_L$ (interrupt latch)	
1D	−	TREG5 (Timer register5)	3D	IL$_H$	
1E	−	TC5CR (TC5 control)	3E	reserved	
1F	EINT3CR		3F	PSW (program status word)	RBS (register bank selector)

(a)　Special Function Registers

Address	Read	Write
$0F80_H$ – 0F93	LCD display data buffer	
0F94 – 0FEF	reserved	
0FF0 – 0FF7	SIO1 transmit and receive data buffer	
0FF8 – 0FFF	SIO2 transmit and receive data buffer	

Note 1: Do not access reserved areas by the program.

Note 2: When defining address $003F_H$ with assembler symbols, use GPSW and GRBS.

Note 3: − ; Do not access.

Note 4: Operations specified to writing registers and interrupt latches by read modifying write instructions (bit operation instructions such as SET, CLR, etc. , or operation instructions such as AND, OR, etc.) are not effective.

(b)　Data Buffer Registers

Figure 2-1.　SFR and DBR

2.2   I/O Ports

The TMP87EP26 has 11 parallel input/output ports (73 pins) each as follows:

	Primary Function	Secondary Functions
Port P0	8-bit I/O port	——
Port P1	8-bit I/O port	External interrupt input, timer/counter input/output, and divider output
Port P2	3-bit I/O port	Low-frequency resonator connections, external interrupt input, and STOP mode release signal input
Port P3	7-bit I/O port	——
Port P4	8-bit I/O port	Serial interface, external interrupt input, timer/counter input/output
Port P5	8-bit I/O port	Analog input
Port P6	8-bit I/O port	Segment Output
Port P7	8-bit I/O port	Segment Output
Port P8	8-bit I/O port	Segment Output
Port P9	4-bit I/O port	Segment Output
Port PD	3-bit I/O port	——

Each output port contains a latch, which holds the output data. All input ports do not have latches, so the external input data should either be held externally until read or reading should be performed several times before processing. Figure 2-2 shows input/output timing examples.

External data is read from an I/O port in the S1 state of the read cycle during execution of the read instruction. This timing can not be recognized from outside, so that transient input such as chattering must be processed by the program.

Output data output changes in the S2 state of the write cycle during execution of the instruction which writes to an I/O port.

(a) Input Timing   (b) Output Timing

Note: The positions of the read and write cycles may vary, depending on the instruction.
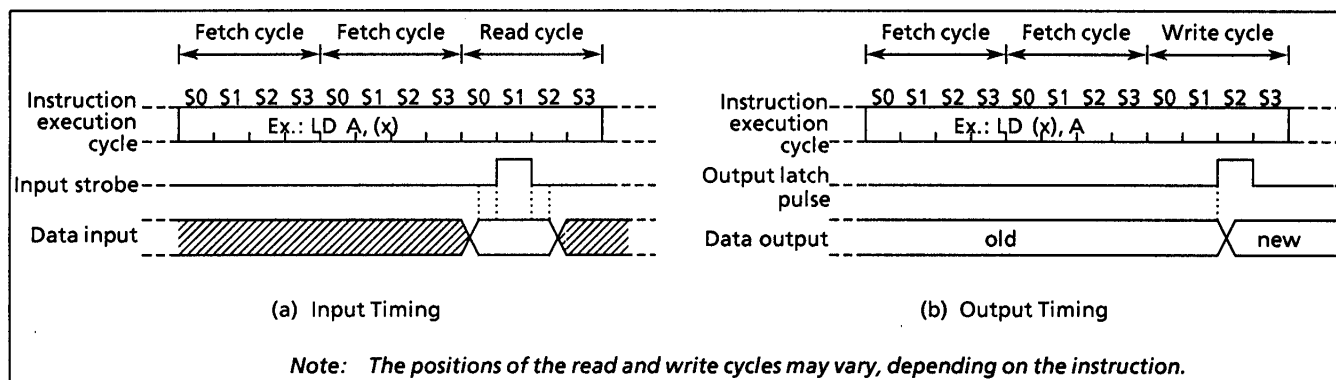
Figure 2-2.   Input/Output Timing (Example)

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

(1)   Instructions that read the output latch contents

    ① XCH   r, (src)   ⑤ LD   (pp) . b, CF

    ② CLR/SET/CPL (src).b   ⑥ ADD/ADDC/SUB/SUBB/AND/OR/XOR   (src), n

    ③ CLR/SET/CPL (pp).g   ⑦ (src) side of ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (HL)

    ④ LD   (src).b, CF

(2)   Instructions that read the pin input data

    ① Instructions other than the above (1)

    ② (HL) side of ADD/ADDC/SUB/SUBB/AND/OR/XOR   (src), (HL)

2.2.1 Port P0 (P07 to P00)

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

During reset, P0CR is initialized to "0", which configures port P0 as input. The P0 output latches are also initialized to "0".

> Note 1: Ports set to the input mode read the pin states. When input pin and output pin exist in port P0 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.
>
> Note 2: The P0CR is a write-only register. It can not be operated by the read-modify instruction (Bit manipulation instructions of SET, CLR, etc. and Arithmetic instructions of AND, OR, etc.)

	7	6	5	4	3	2	1	0	
P0 (0000H) R/W	P07	P06	P05	P04	P03	P02	P01	P00	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P0CR (000AH)									(Initial value: 0000 0000)

POCR	I/O control for port P0 (Set for each bit individually)	0: Input mode 1: Output mode	Write-only

Figure 2-3. Port P0 and P0CR

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

```
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 to P10)

Port P1 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P1 input/output control register (P1CR). Port P1 is configured as an input if its corresponding P1CR bit is cleared to "0", and as an output if its corresponding P1CR bit is set to "1". During reset, the P1CR is initialized to "0", which configures port P1 as an input. The P1 output latches are also initialized to "0".

Port P1 is also used as an external interrupt input, a timer/counter input/output, and a divider output. When used as secondary function pin, the input pins should be set to the input mode, and the output pins should be set to the output mode and beforehand the output latch should be set to "1".

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

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

P1 (0001$_H$) R/W	7	6	5	4	3	2	1	0	
	P17	P16	P15 TC2	P14 $\overline{\text{PPG}}$	P13 $\overline{\text{DVO}}$	P12 INT2 TC1	P11 INT1	P10 $\overline{\text{INT0}}$	(Initial value: 0000 0000)

P1CR (000B$_H$)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000)

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

Figure 2-4. Port P1 and P1CR

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

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

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

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

2.2.3 Port P2 (P22 to P20)

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

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

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

When a read instruction is executed for port P2, bits 7 to 3 read in as indefinite.
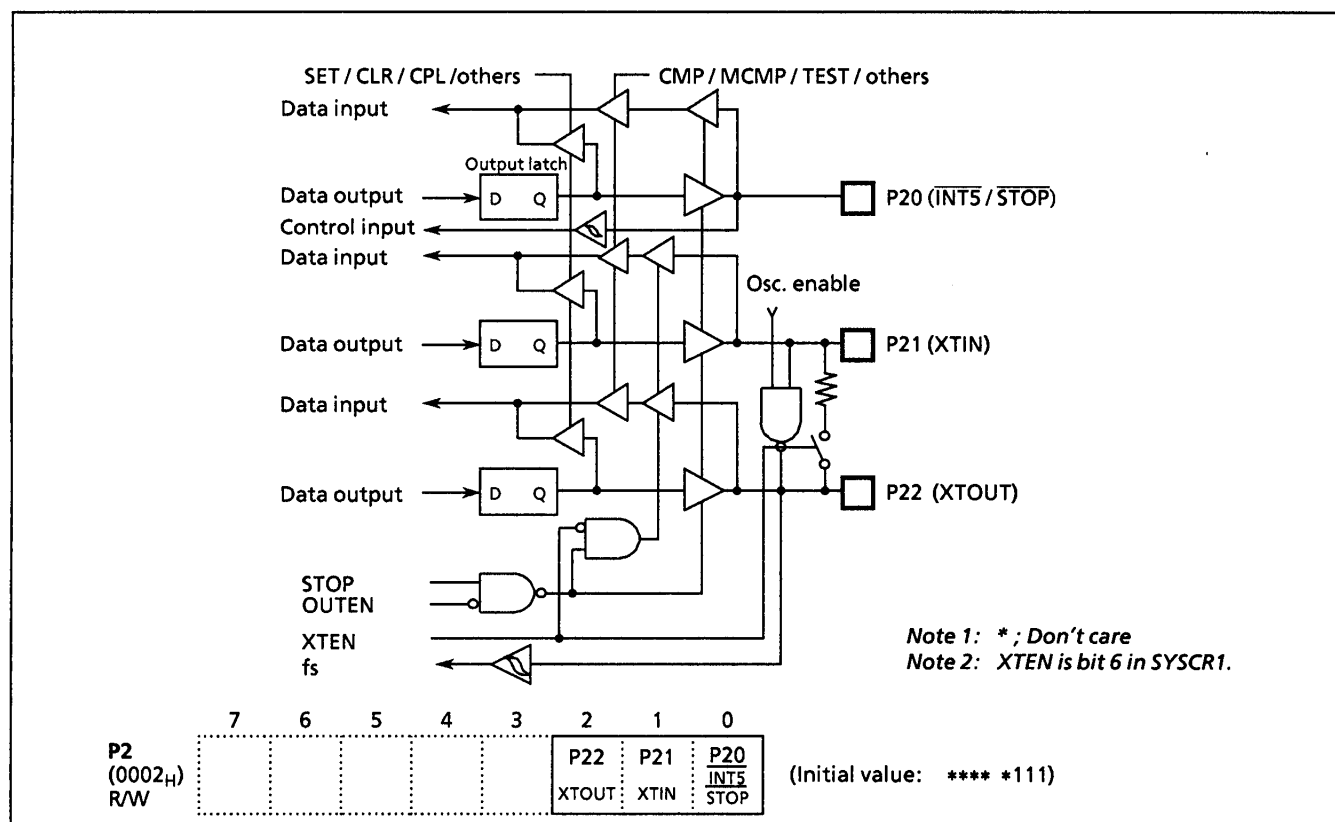
Figure 2-5. Port P2

2.2.4 Port P3 (P36 to P30)

Port P3 is an 7-bit input/output port. When used as an input port, the output latch should be set to "1". The output latches are initialized to "1" during reset.

When a read instruction is executed for port P3, bit 7 reads in as indefinite.

    Example 1: Output the immediate data 5A$_H$ to the P3 port.
        LD   (P3), 5AH   ; P3←5A$_H$
    Example 2: Inverts the output of the upper 3bits (P36 to P34) of the P3 port.
        XOR   (P3), *1110000B
                    ; P36 to P34←$\overline{P36}$ to $\overline{P34}$
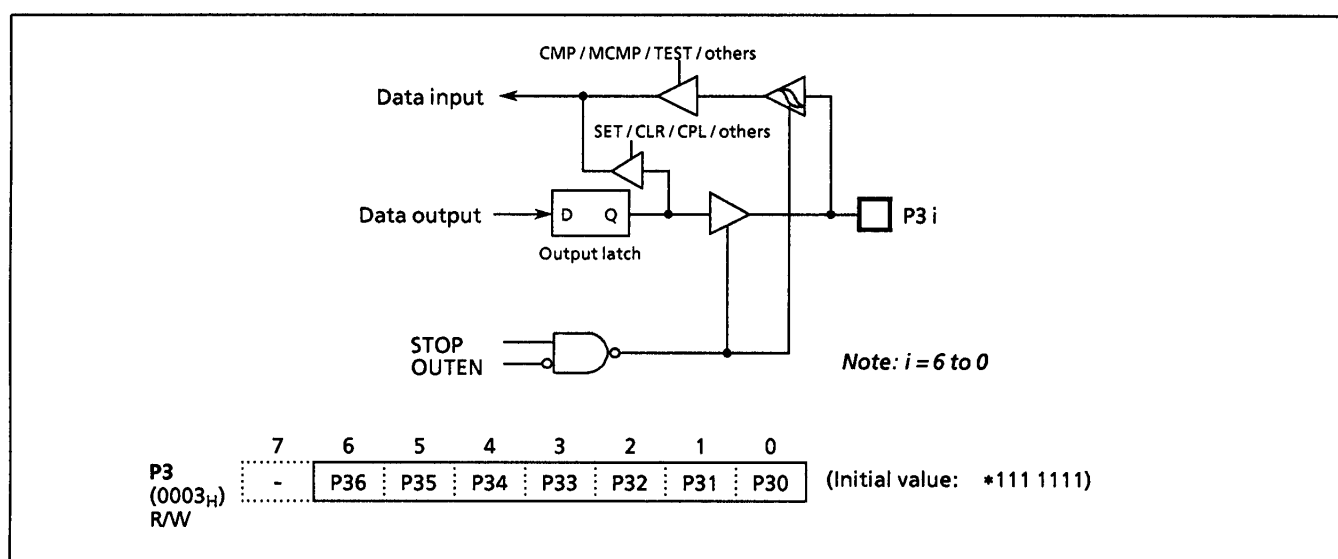
Figure 2-6.   Port P3

2.2.5 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port, and is also used as an external interrupt input, a timer/counter input/output and a serial interface input/output. Input/output mode is specified by the corresponding bit in the port P4 input/output control register (P4CR1). It can be selected whether output circuit of P4 port is Push-pull port or Sink open drain individually, by setting P4CR2.

When used as a timer/counter output and serial interface output, respective P4CR1 should be set to "1" after P4 output latch is set to "1".

When used as an input port, external interrupt input, timer/counter input and serial interface input, respective P4CR1 should be set to "0" after P4CR2 is set to "0".

During reset, the P4CR1 is initialized to "1", and configures port P4 as an output mode. Also, port P4 output latch is initialized to "1".

Note: i = 7 to 0 ; Don't care

	7	6	5	4	3	2	1	0	
P4 (0004$_H$) R/W	P47 SO2	P46 SI2	P45 $\overline{SCK2}$	P44 SO1	P43 SI1	P42 $\overline{SCK1}$	P41 $\overline{PWM}$ PDO	P40 INT3 TC3	(Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
P4CR1 (000C$_H$)									(Initial value: 1111 1111)

P4CR1	Port 4 / I/O select (Set for each bit individually)	0: Input mode 1: Output mode	Write-only

	7	6	5	4	3	2	1	0	
P4CR2 (002D$_H$)									(Initial value: 0000 0000)

P4CR2	Port 4 / Output circuit control (Set for each bit individually)	0: Open-drain output 1: Push-pull output	Write - only

Figure 2-7. Port P4

Note 1: Ports set to the input mode read the pin states. When input pin and output pin exist in port P4 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.

Note 2: The P4CR1 and P4CR2 are a write-only register. It can not be operated by the read-modify instruction (Bit manipulation instructions of SET, CLR, etc. and Arithmetic instructions of AND, OR, etc.)
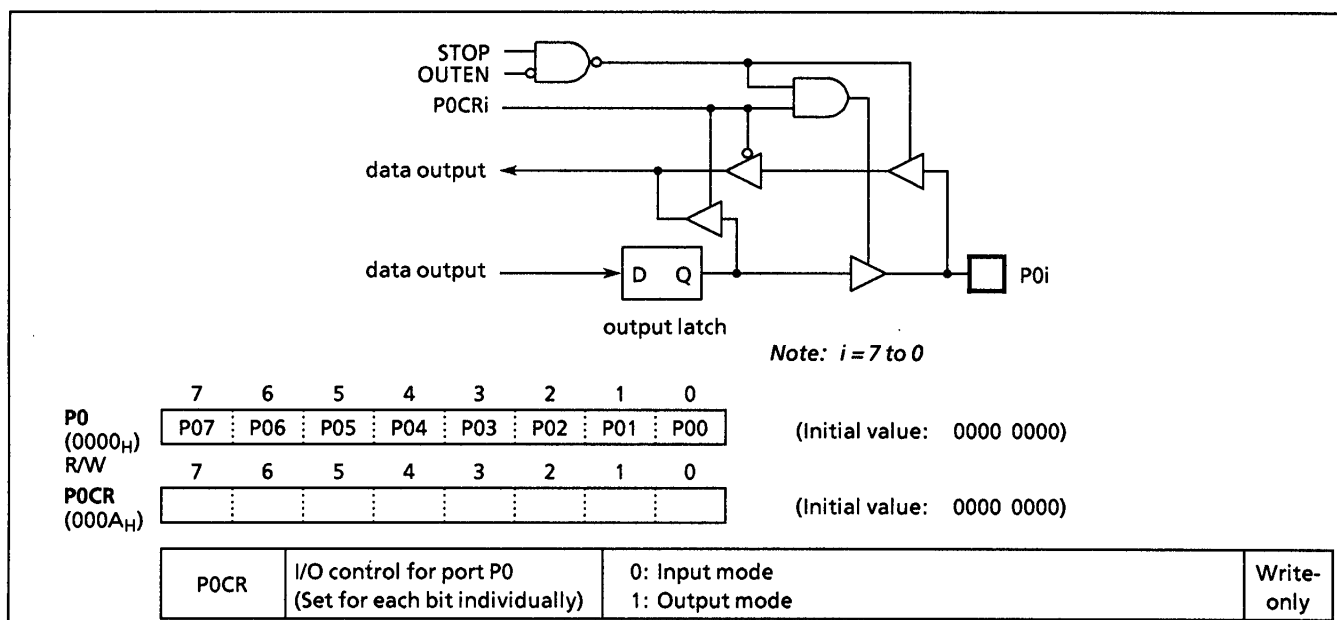
2.2.6 Port P5 (P57 to P50)

Port P5 is a general-purpose 8-bit I/O port that can be specified bitwise. It is also used for analog input. Specify input or output using the P5 I/O control register (P5CR) and AINDS (bit 4 of ADCCR). At reset, the P5CR is set to 0; AINDS, to 0, setting port P5 to analog input. At reset, the output latch of port P5 is initialized to "0". The P5CR is write-only register. The pins of port P5 not specified for analog input can be used as an I/O port; to maintain accuracy, do not use them for output instructions during AD conversion. While the AD converter is operating, if a read instruction is executed for port P5, read data of port selected to analog input is "1".

Note: i = 7 to 0

	7	6	5	4	3	2	1	0	
P5 (0005H) R/W	P57 AIN7	P56 AIN6	P55 AIN5	P54 AIN4	P53 AIN3	P52 AIN2	P51 AIN1	P50 AIN0	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P5CR (000DH)									(Initial value: 0000 0000)

P5CR	I/O control for port P5 (Set for each bit individually)	0: Input mode 1: Output mode	Write-only

Note 1: P5CR is a write-only register and must not be used with any of the read-modify-write instruction.
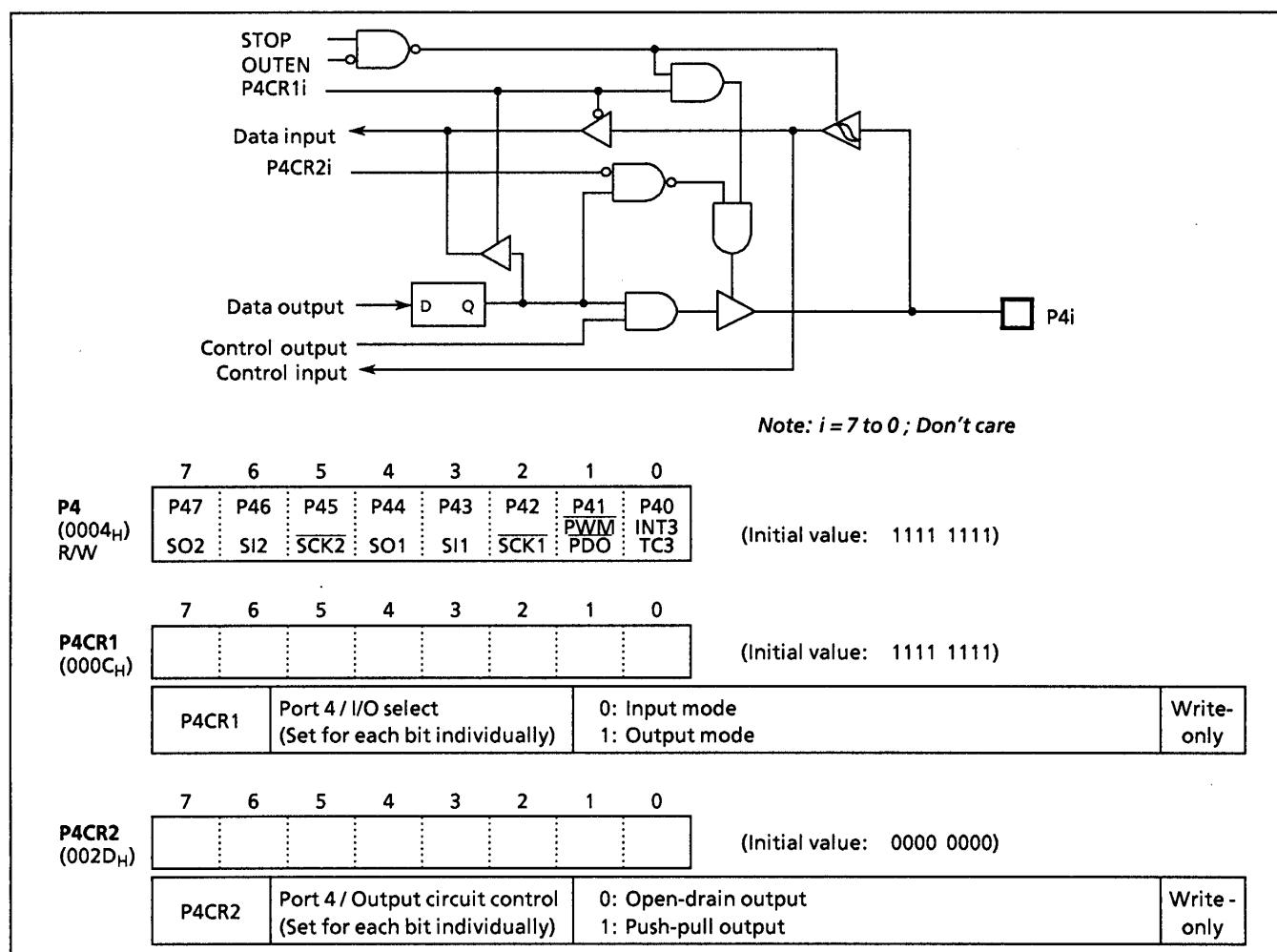Note 2: Unused analog input pins cannot be configured as output mode when AINDS = 0.

Figure 2-8. Port P5

Note 1: Ports set to the input mode read the pin states. When input pin and output pin exist in port P5 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.

Note 2: The P5CR is a write-only register. It can not be operated by the read-modify instruction (Bit manipulation instructions of SET, CLR, etc. and Arithmetic instructions of AND, OR, etc.)

2.2.7 Ports P6 (P67 to P60) Port P7 (P77 to P70) Port P8 (P87 to P80) Port P9 (P93 to P90)

Port P6, P7, P8 and P9 are an 8-bit input / output ports and are also used as the segment output port.
Input / output mode or segement output mode is specified by the corresponding bit in the Px port control register (PxCR). During reset, PxCR is initialized to "0", which configure port Px as input / output. Port Px output latches are also initialized to "1". PxCR can only be written.
When a read instruction is executed for port P9, bits 7 to 4 read in as indefinite.   Note: x = 6,7,8,9

Example: Setting the upper 2 bits of port P6 as a segment output port, and the others as input/output port.

LD   (P6CR), 11000000B

P6 (0006H) R/W

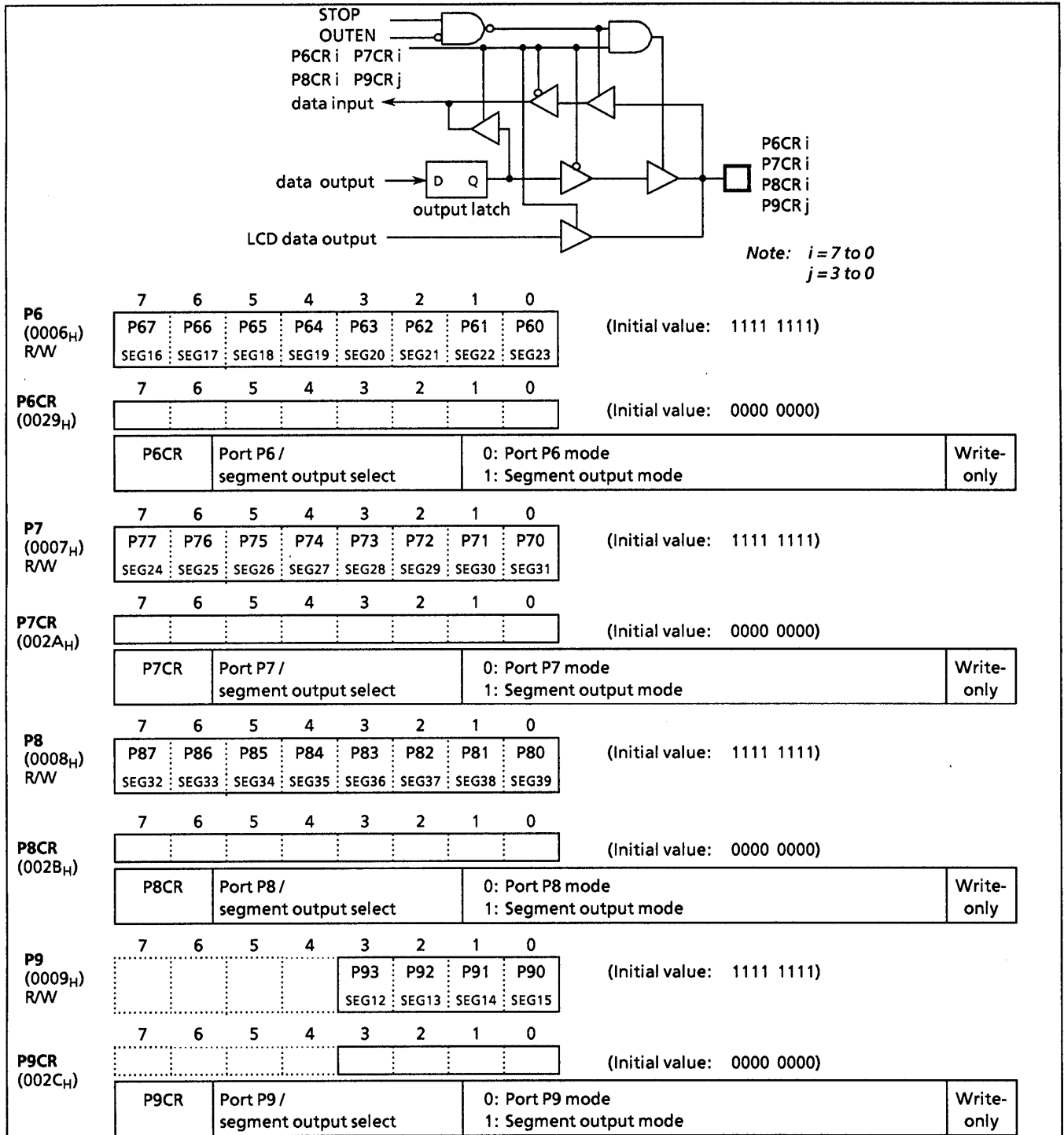| 7	6	5	4	3	2	1	0
P67	P66	P65	P64	P63	P62	P61	P60
SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23

(Initial value: 1111 1111)

P6CR (0029H)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P6CR	Port P6 / segment output select	0: Port P6 mode 1: Segment output mode	Write-only

P7 (0007H) R/W

7	6	5	4	3	2	1	0
P77	P76	P75	P74	P73	P72	P71	P70
SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31

(Initial value: 1111 1111)

P7CR (002AH)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P7CR	Port P7 / segment output select	0: Port P7 mode 1: Segment output mode	Write-only

P8 (0008H) R/W

7	6	5	4	3	2	1	0
P87	P86	P85	P84	P83	P82	P81	P80
SEG32	SEG33	SEG34	SEG35	SEG36	SEG37	SEG38	SEG39

(Initial value: 1111 1111)

P8CR (002BH)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P8CR	Port P8 / segment output select	0: Port P8 mode 1: Segment output mode	Write-only

P9 (0009H) R/W

7	6	5	4	3	2	1	0
				P93	P92	P91	P90
				SEG12	SEG13	SEG14	SEG15

(Initial value: 1111 1111)

P9CR (002CH)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P9CR	Port P9 / segment output select	0: Port P9 mode 1: Segment output mode	Write-only

Figure 2-9. Port P6, P7, P8, P9

> *Note: The P6CR, P7CR, P8CR and P9CR are write-only register. It can not be operated by the read-modify instruction (Bit manipulation instructions of SET, CLR, etc. and Arithmetic instructions of AND, OR, etc.)*

2.2.8 Port PD (PD2 to PD0)

Port PD is an 3-bit input/output port. When used as an input port, the output latch should be set to "1". The output latches are initialized to "1" during reset.

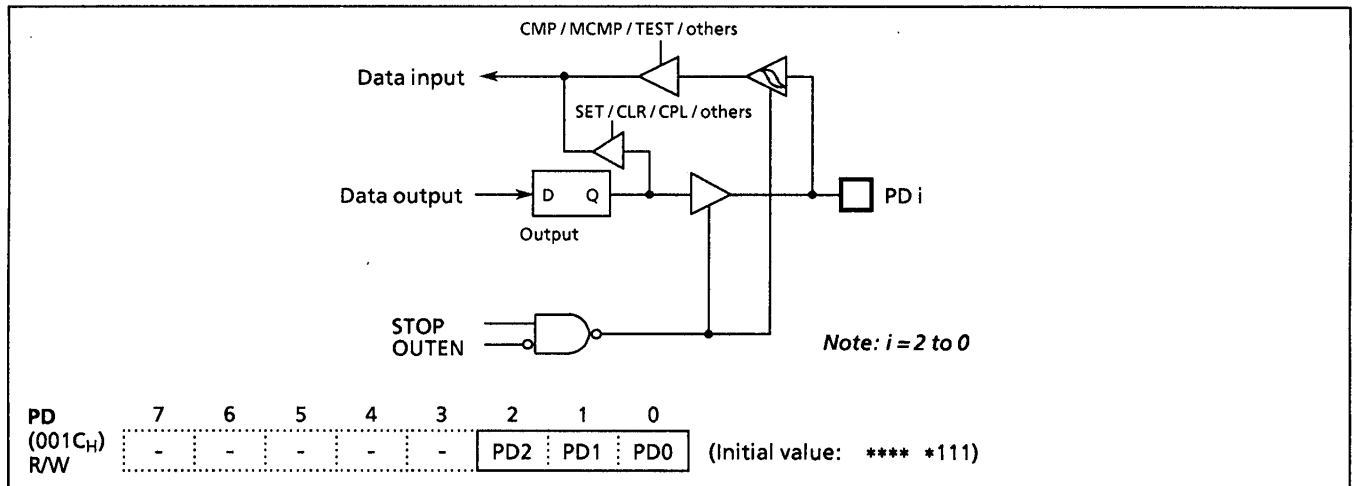When a read instruction is executed for port PD, bits 7 to 3 read in as indefinite.

Figure 2-10. Port PD

2.3   Time Base Timer (TBT)

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

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

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

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

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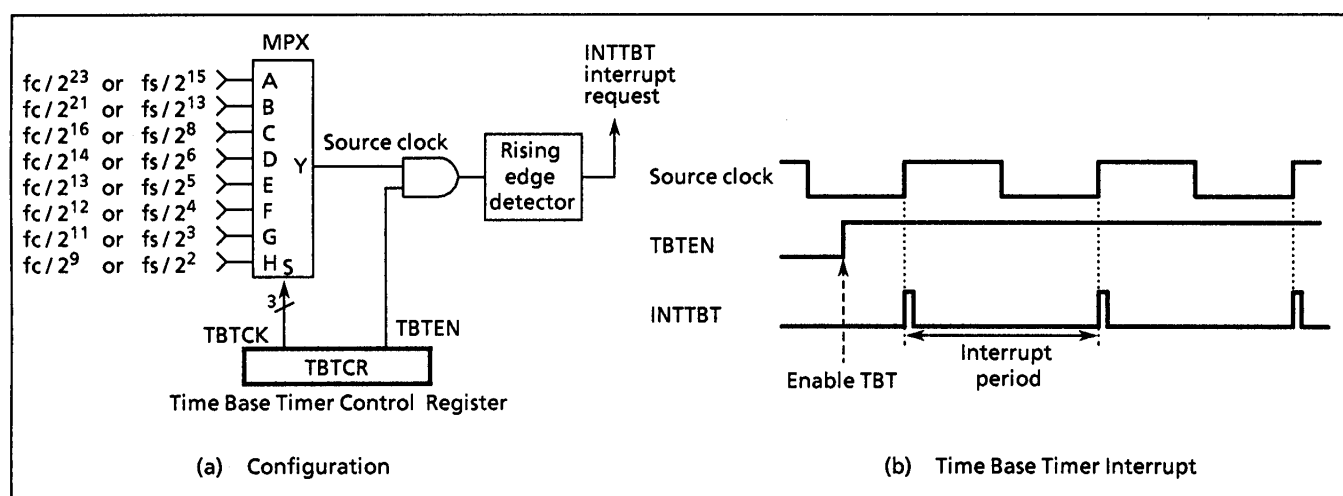   SET   (EIRL). 6

(a) Configuration   (b)   Time Base Timer Interrupt

Figure 2-11. Time Base Timer

TBTCR (0036_H)	7	6	5	4	3	2	1	0	(Initial value: 0**0 0***)
	(DVOEN)	(DV6CK)		(DV7CK)	TBTEN		TBTCK		

TBTEN	Time base timer enable/disable	0: Disable 1: Enable	
TBTCK	Time base timer interrupt frequency select	000:$fc/2^{23}$ or $fs/2^{15}$ [Hz] 001:$fc/2^{21}$ or $fs/2^{13}$ 010:$fc/2^{16}$ or $fs/2^{8}$ 011:$fc/2^{14}$ or $fs/2^{6}$ 100:$fc/2^{13}$ or $fs/2^{5}$ 101:$fc/2^{12}$ or $fs/2^{4}$ 110:$fc/2^{11}$ or $fs/2^{3}$ 111:$fc/2^{9}$ or $fs/2$	R/W

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

Figure 2-12. Time Base Timer and Divider Output Control Register

Table 2-1. Time Base Timer Interrupt Frequency

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

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

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

Divider output circuit is controlled by the control register (TBTCR) shown in Figure 2-13.

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

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

Figure 2-13. Divider Output Control Register

Example: 1 kHz pulse output (at fc = 8 MHz)
```
            SET        (P1).3                ; P13 output latch ←1
            LD         (P1CR), 00001000B     ; Configures P13 as an output mode
            LD         (TBTCR), 10000000B    ; DVOEN←1, DVOCK←00
```

Table 2-2. Frequency of Divider Output

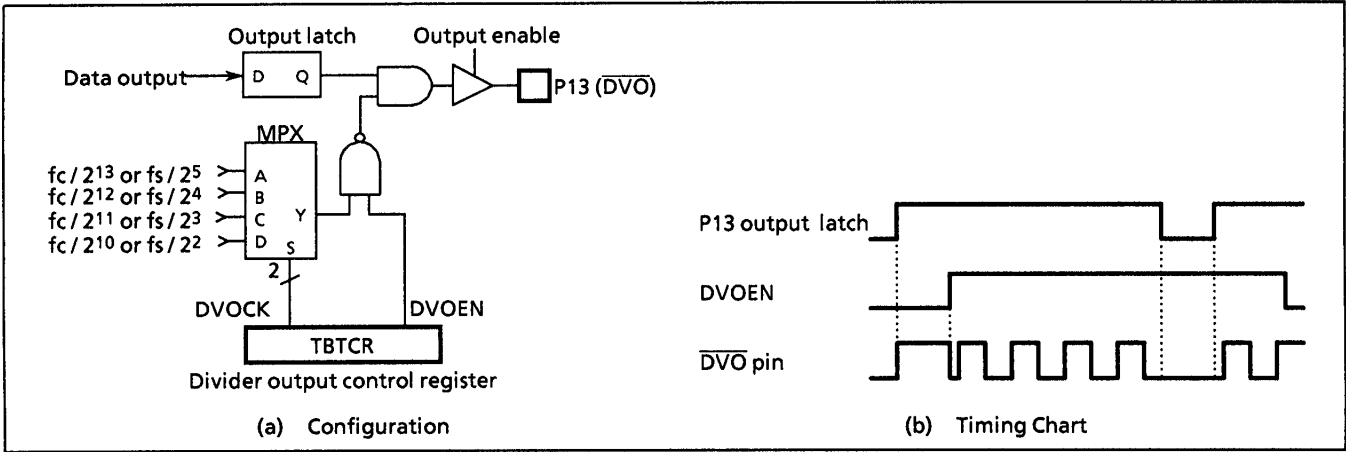| DVOCK	Frequency of Divider Output		At fc = 4.194304 MHz	At fc = 8 MHz	At fs = 32.768 kHz
00	$fc/2^{13}$ or	$fs/2^{5}$	0.512 [kHz]	0.976 [kHz]	1.024 [kHz]
01	$fc/2^{12}$	$fs/2^{4}$	1.024	1.953	2.048
10	$fc/2^{11}$	$fs/2^{3}$	2.048	3.906	4.096
11	$fc/2^{10}$	$fs/2^{2}$	4.096	7.812	8.192

Figure 2-14. Divider Output

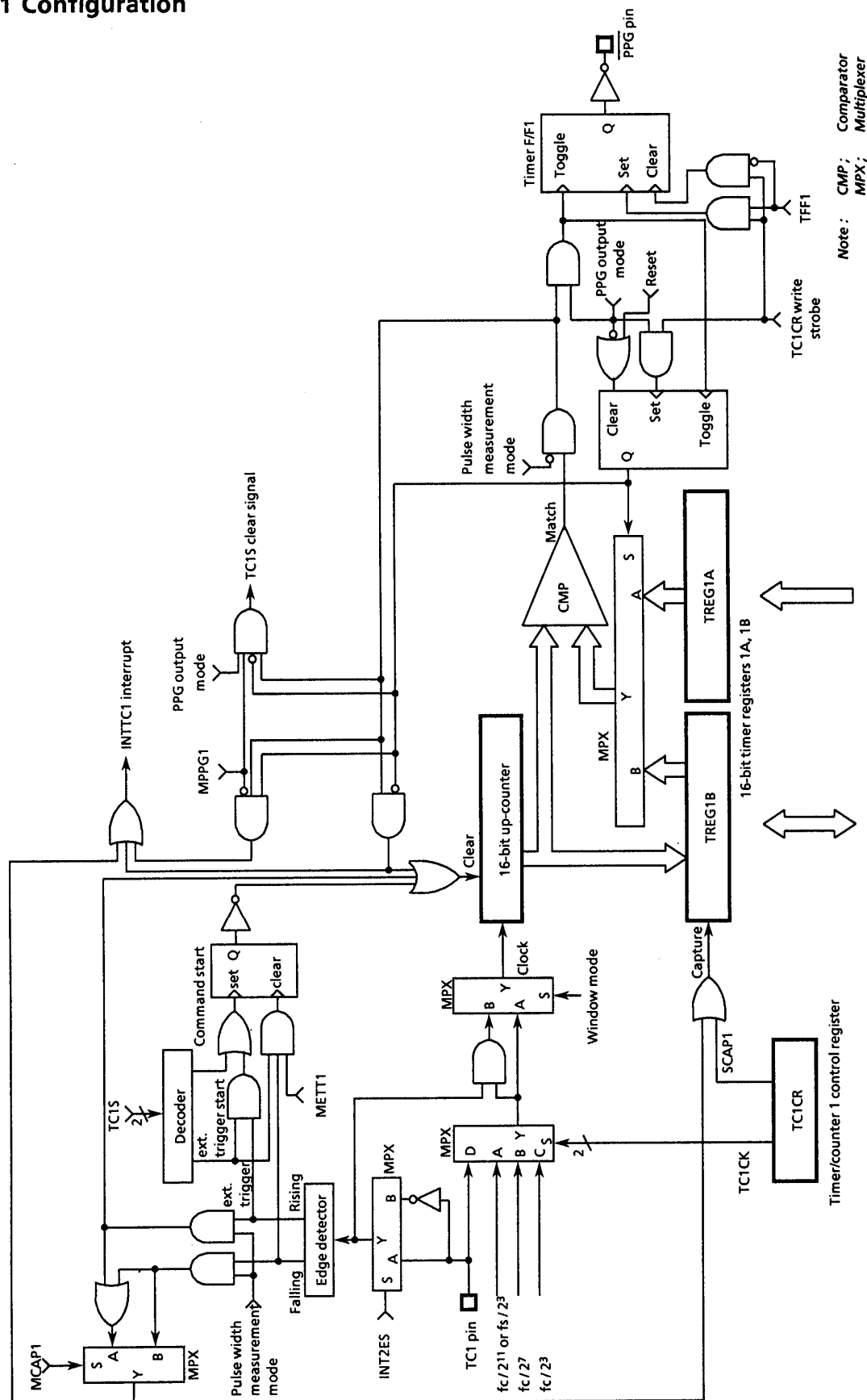## 2.5 16-bit Timer/Counter 1 (TC1)

2.5.1 Configuration

Figure 2-15. Timer/Counter 1

2.5.2 Control

The timer/counter 1 is controlled by a timer/counter 1 control register (TC1CR) and two 16-bit timer registers (TREG1A and TREG1B). Reset does not affect TREG1A and TREG1B.
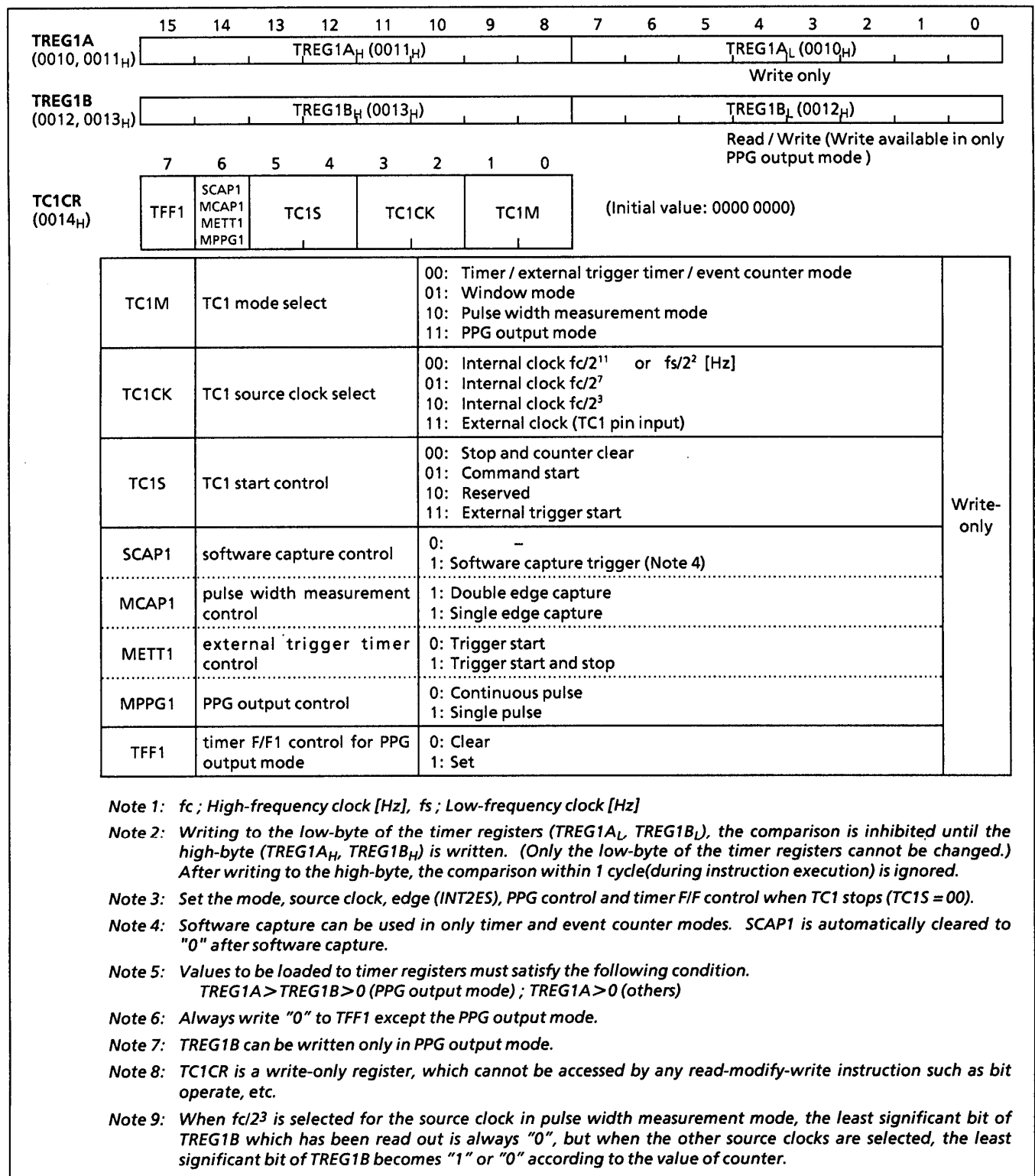
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREG1A ($0010, 0011_H$)				$TREG1A_H$ (0011_H)								$TREG1A_L$ (0010_H)				

Write only

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREG1B ($0012, 0013_H$)				$TREG1B_H$ (0013_H)								$TREG1B_L$ (0012_H)				

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

	7	6	5	4	3	2	1	0	
TC1CR (0014_H)	TFF1	SCAP1 MCAP1 METT1 MPPG1	TC1S		TC1CK		TC1M		(Initial value: 0000 0000)

TC1M	TC1 mode select	00: Timer / external trigger timer / event counter mode 01: Window mode 10: Pulse width measurement mode 11: PPG output mode	
TC1CK	TC1 source clock select	00: Internal clock $fc/2^{11}$ or $fs/2^2$ [Hz] 01: Internal clock $fc/2^7$ 10: Internal clock $fc/2^3$ 11: External clock (TC1 pin input)	
TC1S	TC1 start control	00: Stop and counter clear 01: Command start 10: Reserved 11: External trigger start	Write- only
SCAP1	software capture control	0: – 1: Software capture trigger (Note 4)	
MCAP1	pulse width measurement control	1: Double edge capture 1: Single edge capture	
METT1	external trigger timer control	0: Trigger start 1: Trigger start and stop	
MPPG1	PPG output control	0: Continuous pulse 1: Single pulse	
TFF1	timer F/F1 control for PPG output mode	0: Clear 1: Set	

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

Note 2:   Writing to the low-byte of the timer registers ($TREG1A_L$, $TREG1B_L$), the comparison is inhibited until the high-byte ($TREG1A_H$, $TREG1B_H$) is written. (Only the low-byte of the timer registers cannot be changed.) After writing to the high-byte, the comparison within 1 cycle(during instruction execution) is ignored.

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

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

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

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

Note 7:   TREG1B can be written only in PPG output mode.

Note 8:   TC1CR is a write-only register, which cannot be accessed by any read-modify-write instruction such as bit operate, etc.

Note 9:   When $fc/2^3$ is selected for the source clock in pulse width measurement mode, the least significant bit of TREG1B which has been read out is always "0", but when the other source clocks are selected, the least significant bit of TREG1B becomes "1" or "0" according to the value of counter.

Figure 2-16. Timer Registers and TC1 Control Register

2.5.3 Function

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

(1) **Timer** Mode

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

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

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

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

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

Note: TC1CR is a write-only register, which cannot start by [SET(TC1CR).4] instruction.

Example 2: Software capture

LD	(TC1CR), 01010000B	; SCAP1←1 (Captures)
LD	WA, (TREG1B)	; Reads captured value
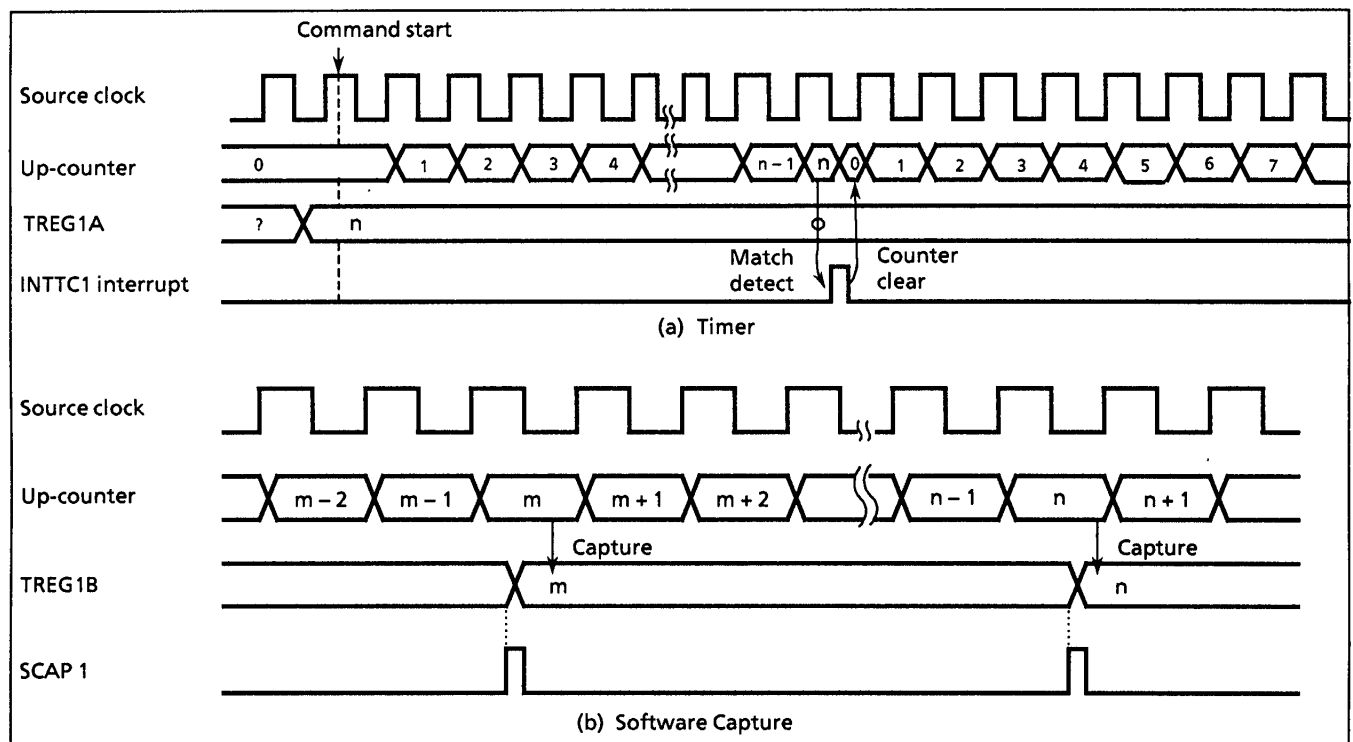
Figure 2-17. Timer Mode Timing Chart

(2) External Trigger Timer mode

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

When the edge input is opposite to the edge input way of the count start trigger at METTI (bit 6 in TC1CR) = 1, the counter is cleared, and count stops. In this mode, pulse input with a constant pulse width generates interrupt. When METT1 is "0", the opposite edge input is ignored. The edge of TC1 pin input before match detection is also ignored.

The TC1 pin input has the same noise rejection as the INT2 pin; therefore, pulses of 7/fc [s] or less are rejected as noise. A pulse width of 24/fc [s] or more is required for edge detection in NORMAL1/2 or IDLE1/2 mode. The noise rejection circuit is turned off in SLOW and SLEEP modes. But, a pulse width of 1 machine cycle or more is required.

Example 1:Generates interrupt after 100 μs from TC1 pin input rising edge (at fc = 8 MHz).

```
        LD      (EINTCR), 00000000B    ; INT2ES←0 (rising edge)
        LDW     (TREG1A), 0064H        ; 100 μs ÷ 2³ / fc = 64H
        SET     (EIRL).EF4             ; Enables INTTC1 interrupt
        EI
        LD      (TC1CR), 00111000B     ; Starts TC1 external trigger, METT = 0
```

Example 2: When "L" level pulses of 4ms or more is input to TC1 pin, generates interrupt.

(at fc = 8 MHz)

```
LD      (EINTCR), 00000100B      ; INT2ES←1 ("L" level)
LDW     (TREG1A), 00FAH          ; 4 ms ÷ 2⁷ / fc = FAH
SET     (EIRL).EF4               ; Enables INTTC1 interrupt
EI
LD      (TC1CR), 01110100B       ; Starts TC1 external trigger, METT = 1
```


(a) Trigger Start (METT = 0)

(b) Trigger Start and Stop (METT1 = 1)
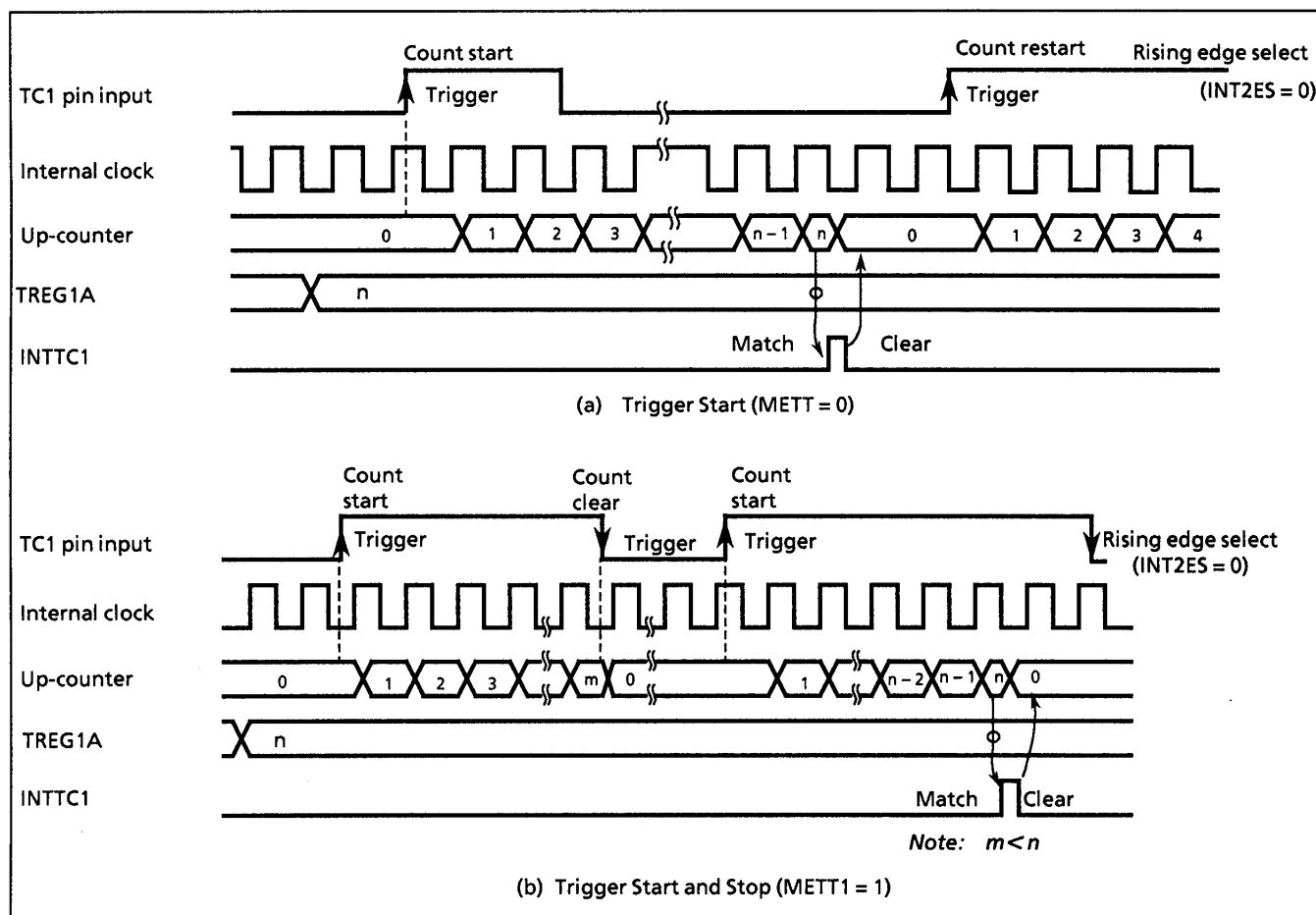
Figure 2-18. External Trigger Timer Mode Timing Chart

(3) **Event Counter** Mode

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

Setting SCAP1 to "1" transferres the current contents of up-counter to TREG1B (software capture function). SCAP is automatically cleared after capturing.
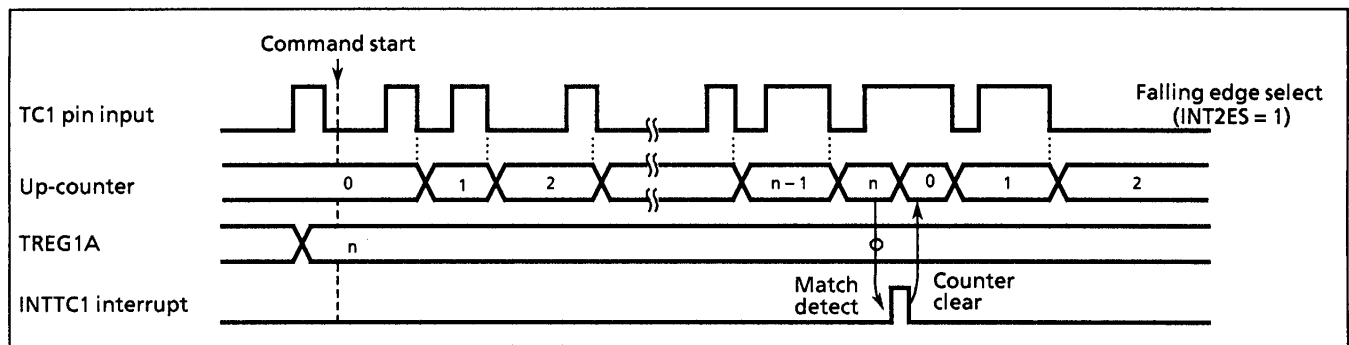
Figure 2-19. Event Counter Mode Timing Chart (INT2ES = 1)

(4) Window mode

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

(a) Positive Logic (INT2ES = 0)

(b) Negative Logic (INT2ES = 1)
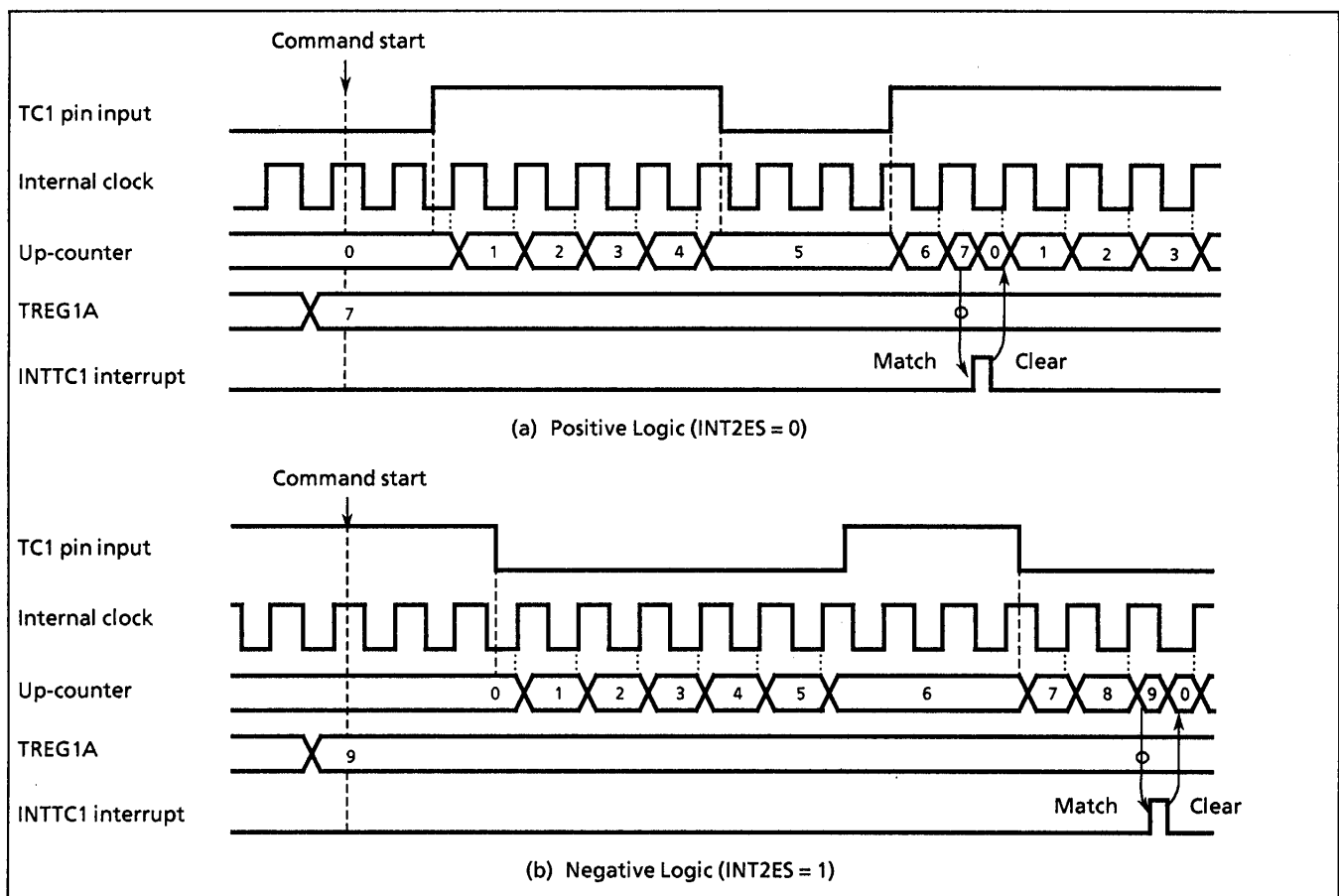
Figure 2-20. Window Mode Timing Chart

(5) Pulse width measurement mode

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

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

Example: Duty measurement (Resolution fc/2⁷ [Hz])

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

$$\text{Resolution} = fc/2^7 \text{ [Hz]}$$

(a) Single Edge Capture

[Applications]   High or low pulse width measurement

(b) Double Edge Capture

[Applications] ① Period / Frequency measurement
                ② Duty measurement
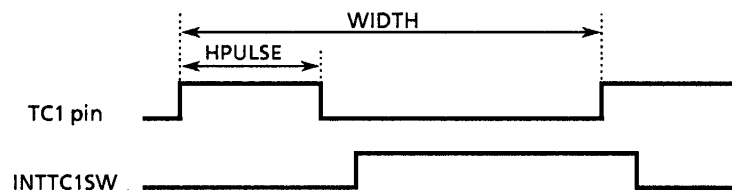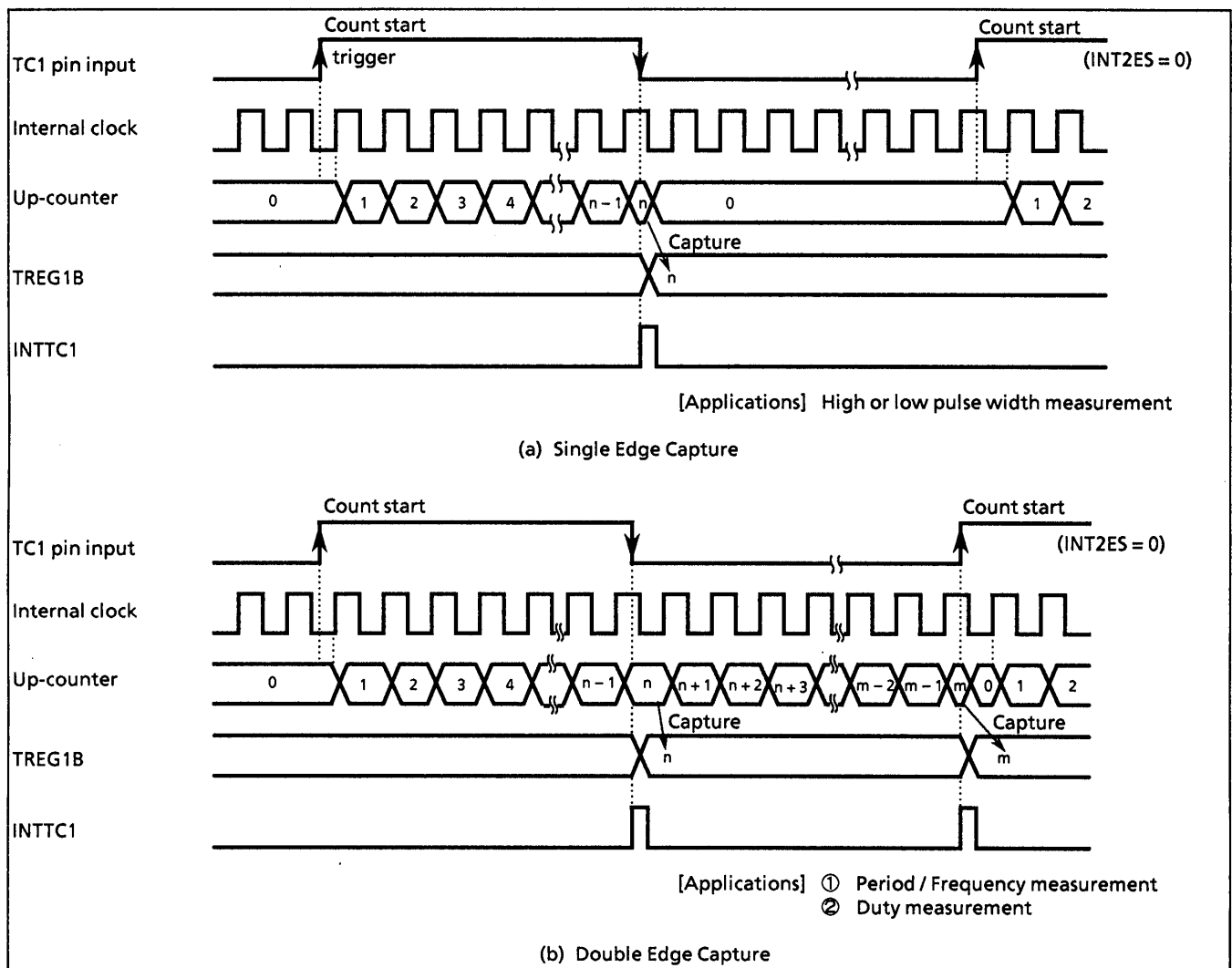
Figure 2-21. Pulse Width Measurement Mode Timing Chart

(6) Programmable Pulse Generate (PPG) output mode

Counting is started by an edge of the TC1 pin input (either the rising or falling edge can be selected) or by a command. The source clock is used an internal clock. First, the contents of TREG1B are compared with the contents of the up-counter. If a match is found, timer F/F1 output is toggled. Next, timer F/F1 is again toggled and the counter is cleared by matching with TREG1A. An INTTC1 interrupt is generated at this time. Timer F/F output is connected to the P14 ($\overline{PPG}$) pin. In the case of $\overline{PPG}$ output, set the P14 output latch to "1" and configure as an output with P1CR4. Timer F/F1 is cleared to "0" during reset. The timer F/F1 value can also be set by program and either a positive or negative logic pulse output is available. Also, writing to the TREG1B is not possible unless the timer / counter 1 is set to the PPG output mode.

Example: "H" level 800 μs, "L"" level 200 μs pulse output at fc = 8 MHz

```
            SET    (P1).4                ; P14 output latch←1
            LD     (P1CR), 00010000B     ; Sets P14 to an output mode
            LD     (TC1CR), 10001011B    ; Sets PPG output mode
            LDW    (TREG1A), 03E8H       ; Sets a period (1 ms ÷ 1 μs = 03E8H)
            LDW    (TREG1B), 00C8H       ; Sets "L" level pulse width (200 μs ÷ 1 μs = 00C8H)
            LD     (TC1CR), 10010011B    ; Start
```

Figure 2-22. $\overline{PPG}$ Output

(a) Pulse

(b) Single
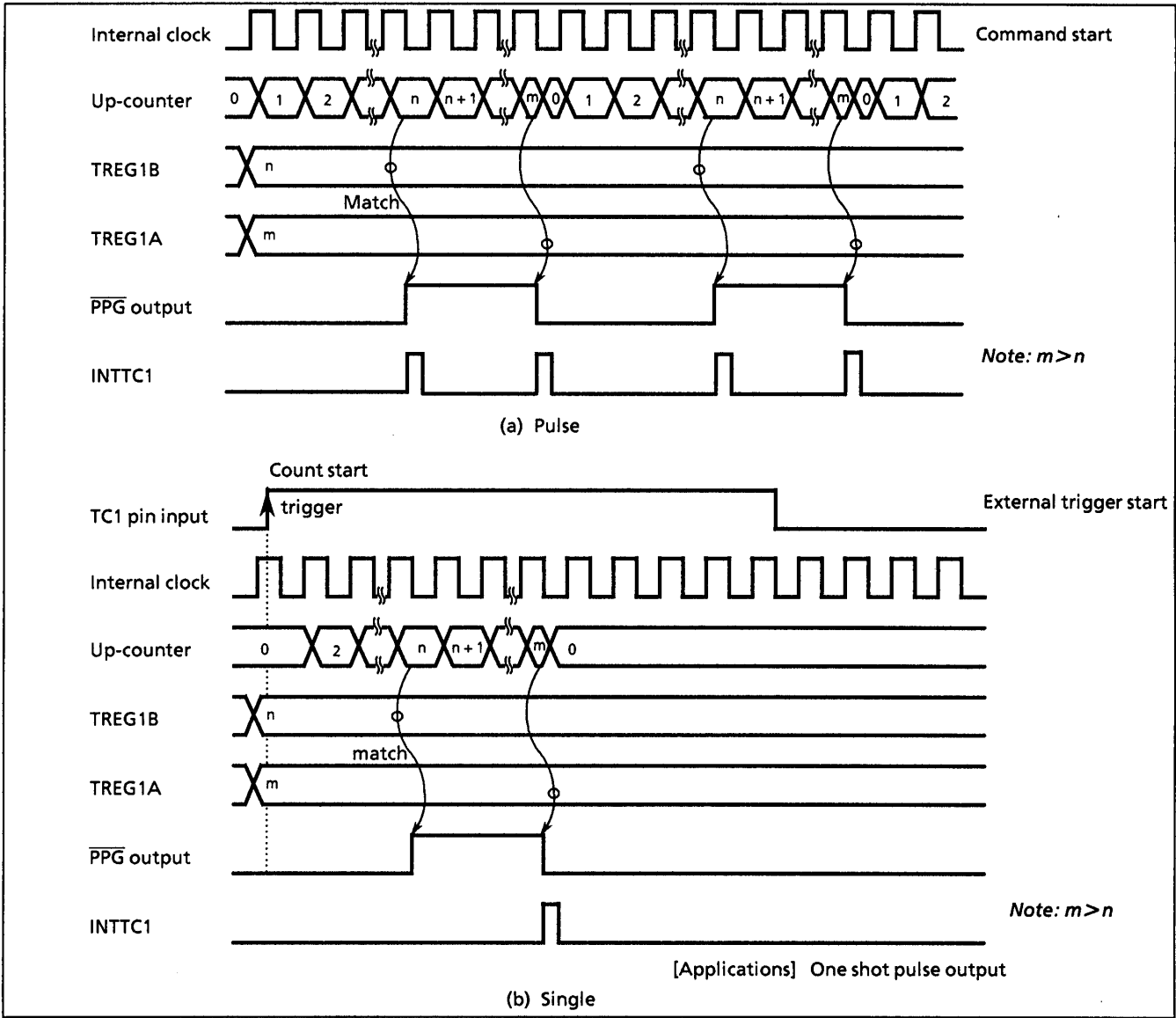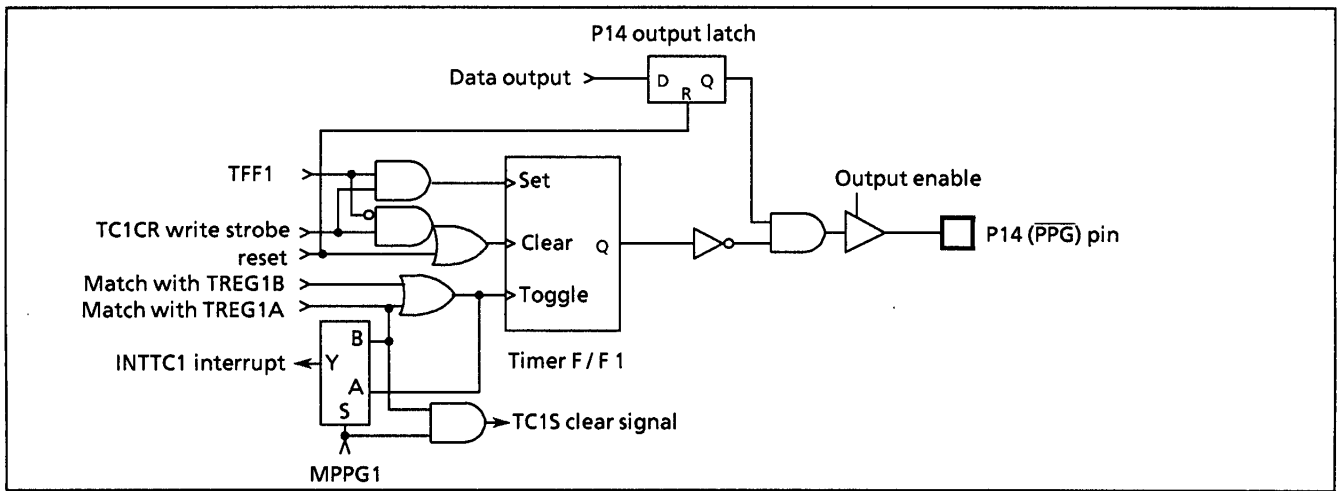
Figure 2-23. PPG Output Mode Timing Chart

2.6   16-bit Timer/Counter 2 (TC2)

2.6.1 Configuration
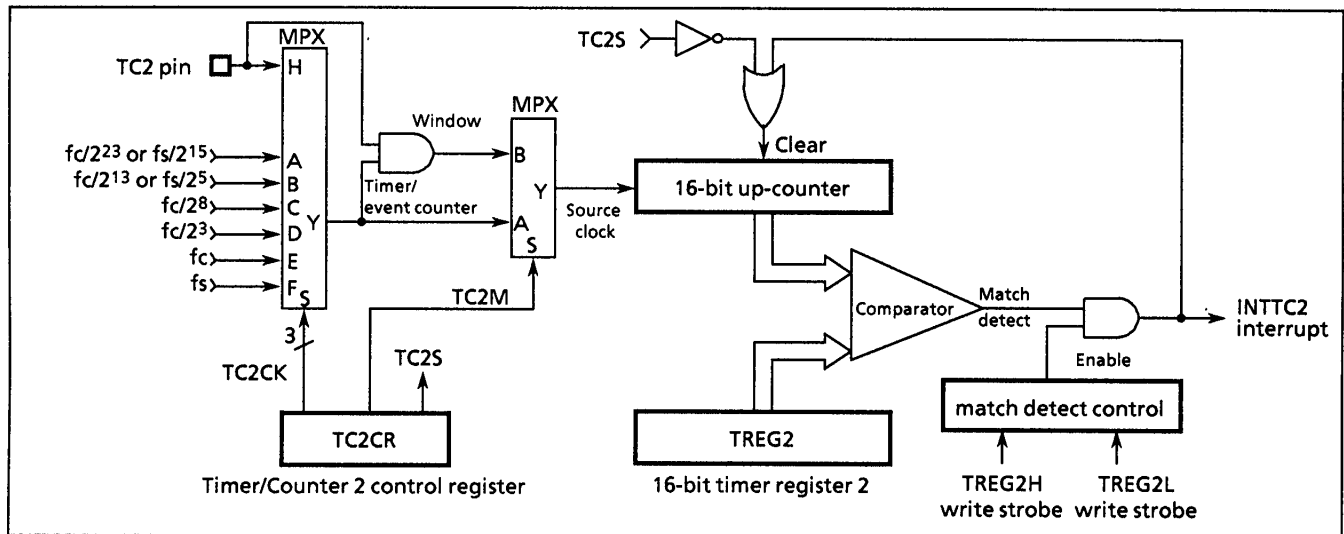
Figure 2-24. Timer/Counter 2 (TC2)

2.6.2 Control

The timer/counter 2 is controlled by a timer/counter 2 control register (TC2CR) and a 16-bit timer register 2 (TREG2). Reset does not affect TREG2.
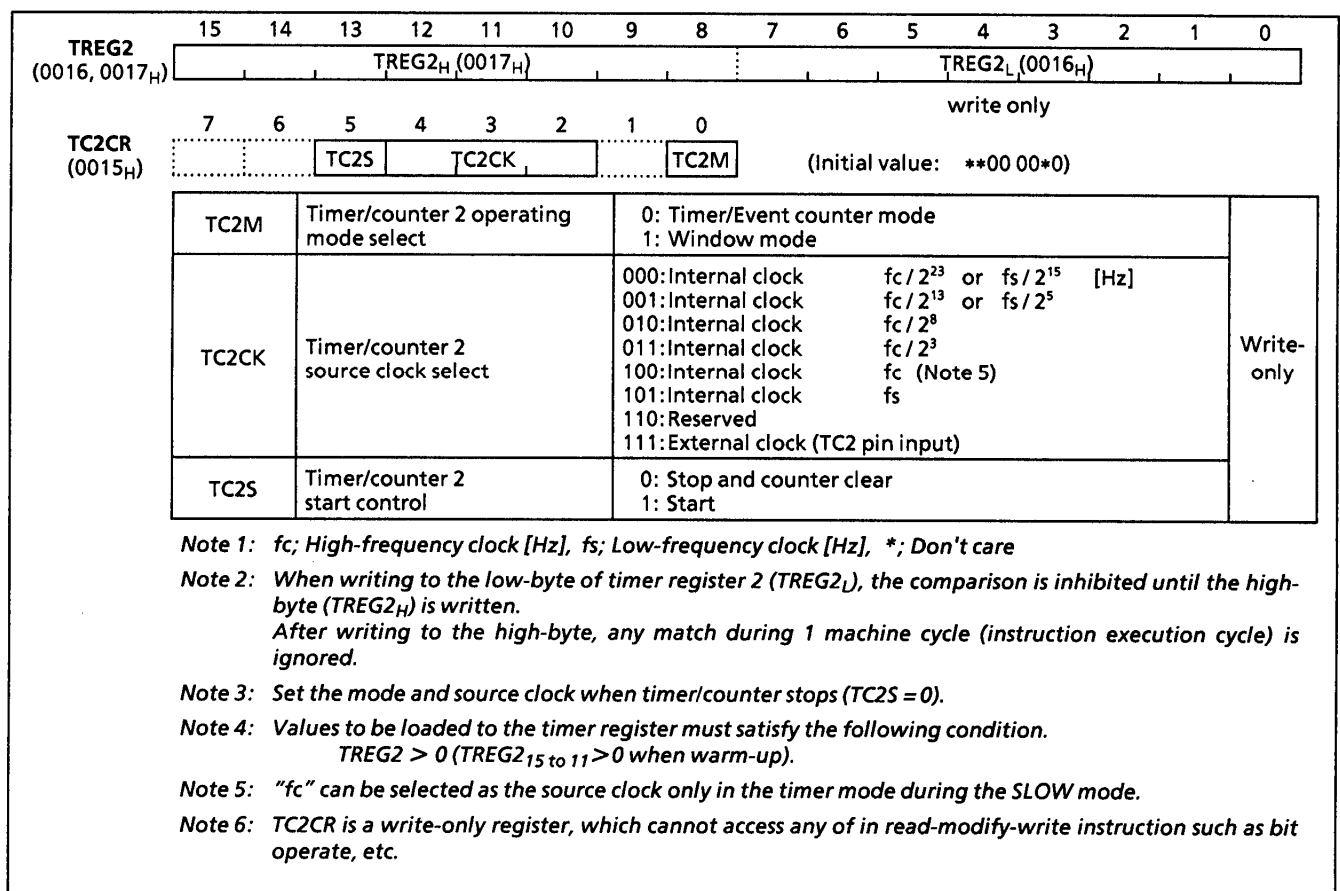
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREG2 (0016, 0017H)				$TREG2_H$ (0017H)								$TREG2_L$ (0016H)			

write only

	7	6	5	4	3	2	1	0
TC2CR (0015H)			TC2S		TC2CK			TC2M

(Initial value: **00 00*0)

TC2M	Timer/counter 2 operating mode select	0: Timer/Event counter mode 1: Window mode	Write-only
TC2CK	Timer/counter 2 source clock select	000:Internal clock $fc/2^{23}$ or $fs/2^{15}$ [Hz] 001: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)	Write-only
TC2S	Timer/counter 2 start control	0: Stop and counter clear 1: Start	

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

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

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

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

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

Note 6: TC2CR is a write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

Figure 2-25. Timer Register 2 and TC2 Control Register

2.6.3 Function

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

(1) **Timer Mode**

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

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

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

Source clock				Resolution		Maximum time setting	
NORMAL1/2, IDLE1/2 mode		SLOW mode	SLEEP mode	At fc = 8 MHz	At fs = 32.768 kHz	At fc = 8 MHz	At fs = 32.768 kHz
DV7CK = 0	DV7CK = 1						
$fc/2^{23}$[Hz]	$fs/2^{15}$[Hz]	$fs/2^{15}$ [Hz]	$fs/2^{15}$ [Hz]	1.05 [s]	1 [s]	19.1 [h]	18.2 [h]
$fc/2^{13}$	$fs/2^5$	$fs/2^5$	$fs/2^5$	1.02 [ms]	1 [ms]	1.1 [min]	1 [min]
$fc/2^8$	$fc/2^8$	–	–	32 [μs]	–	2.1 [s]	–
$fc/2^3$	$fc/2^3$	–	–	1 [μs]	–	65.5 [ms]	–
–	–	fc (Note)	–	125 [ns]	–	7.936 [ms]	–
fs	fs	–	–	–	30.5 [μs]	–	2 [s]

> Note: "fc" can be used only in the timer mode. This is used for warm up when switching from SLOW mode to NORMAL2 mode.

Example: Sets the timer mode with source clock fc/2³ [Hz] and generates an interrupt every 25 ms (at fc = 8 MHz).

```
LD      (TC2CR), 00001100B      ; Sets the TC2 mode and source clock
LDW     (TREG2), 61A8H          ; Sets TREG2 (25 ms ÷ 2³/fc = 61A8H)
SET     (EIRH).EF14             ; Enable INTTC2
EI
LD      (TC2CR), 00101100B      ; Starts TC2
```

(2) **Event Counter Mode**

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

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

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

(3) Window Mode

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

Example: Inputs "H" level pulse of 120 ms or more and generates interrupt. (at fc = 8 MHz).

LDW	(TREG2), 0078H	; Sets TREG2 (120 ms $\div$ 2^{13}/fc = 0078$_H$)
SET	(EIRH).EF14	; Enables INTTC2 interrupt
EI		
LD	(TC2CR), 00100101B	; Starts TC2
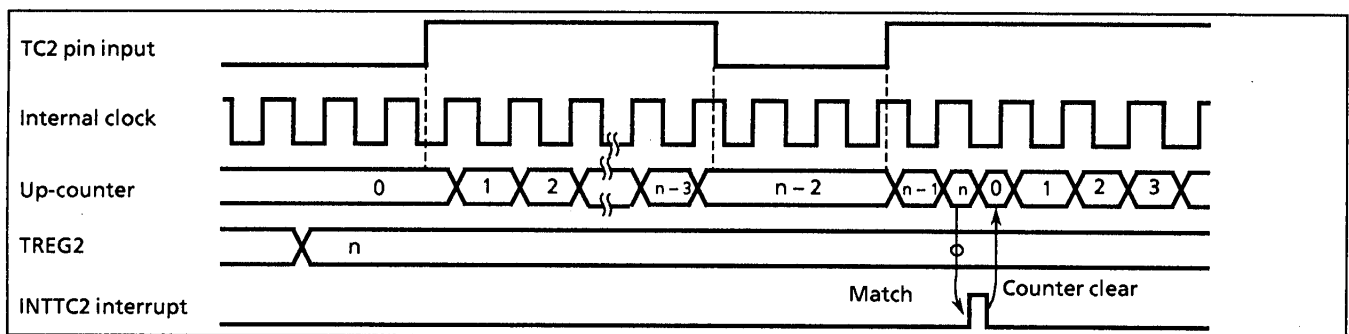
Figure 2-26. Window Mode Timing Chart

2.7   8-Bit Timer/Counter 3 (TC3)

2.7.1 Configuration
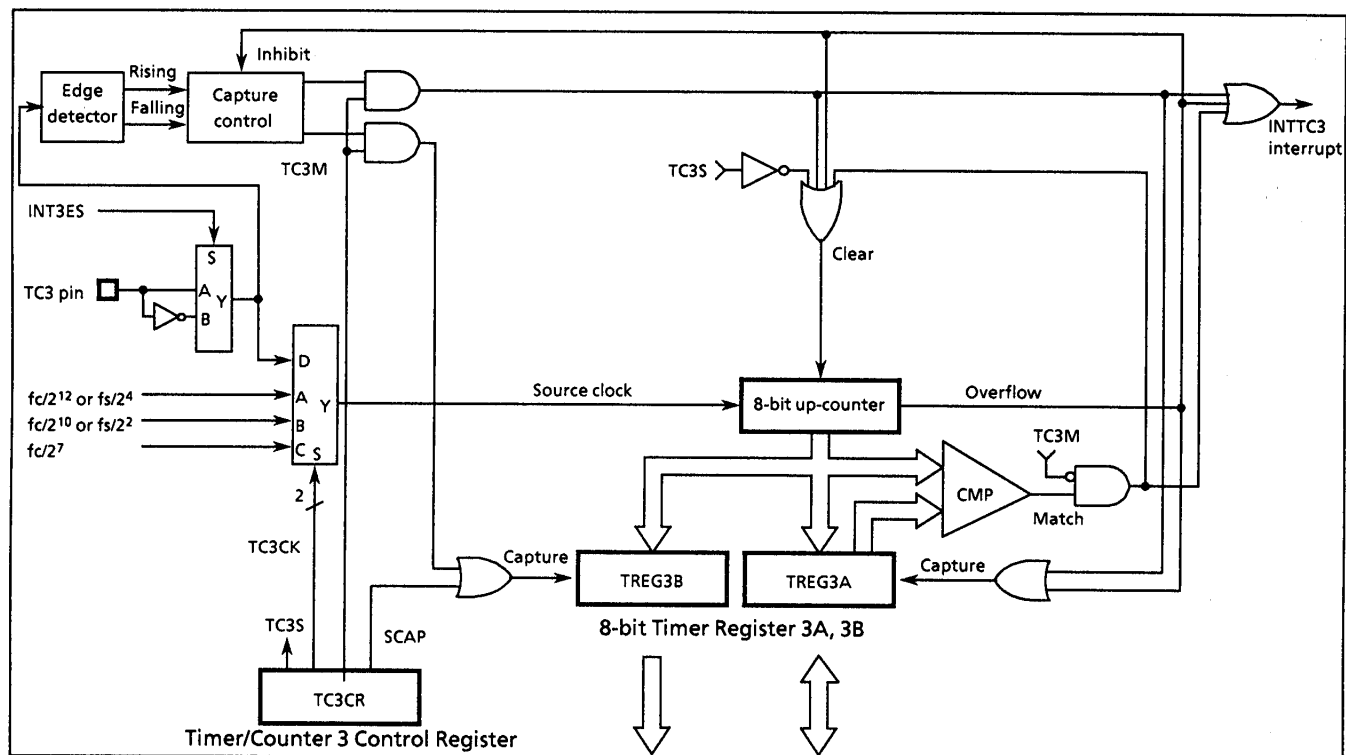
Figure 2-27. Timer/Counter 3

2.7.2 Control

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

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

TREG3B
(0019$_H$)
         Read only

TC3CR
(001A$_H$)
   7 6 5 4 3 2 1 0 | SCAP | | TC3S | TC3CK | | TC3M | (Initial value: *0*0 00*0)

			Write-only
TC3M	Timer/counter 3 operation mode set	0: Timer/event counter 1: Capture	
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 and clear 1: Start	
SCAP	Software capture control	0: – 1: Software capture	

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

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

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

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

Note 5: TC3CR is a write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

Figure 2-28. Timer Register 3A/3B and TC3 Control Register

2.7.3 Function

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

(1) **Timer** Mode

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

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

Source clock			Resolution		Maximum time setting	
NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode	At fc = 8 MHz	At fs = 32.768 kHz	At fc = 8 MHz	At fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
fc/2^{12} [Hz] fc/2^{10} [Hz] fc/2^7 [Hz]	fs/2^4 [Hz] fs/2^2 [Hz] fc/2^7 [Hz]	fs/2^4 [Hz] – –	512 [μs] 128 [μs] 16 [μs]	488.28 [μs] 122.07 [μs] –	130.6 [ms] 32.6 [ms] 4.1 [ms]	124.5 [ms] 31.1 [ms] –

(2) **Event Counter** Mode

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

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

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

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

(3) **Capture** Mode

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

When the TREG3A has been read out, capture and overflow detection resumes.

Thus, it is general to read out TREG3B before reading out TREG3A.
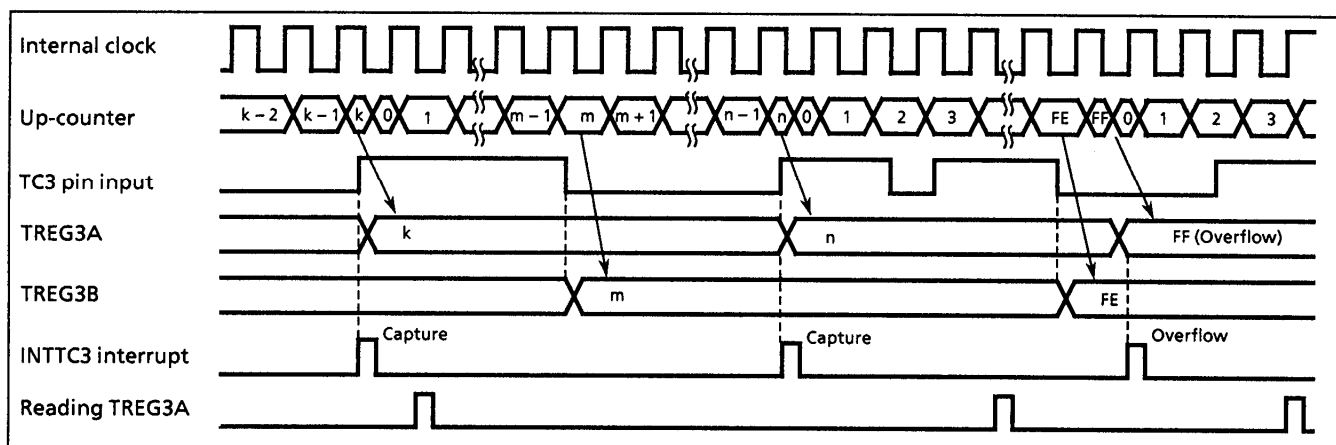
Figure 2-29. Timing Chart for Capture Mode (INT3ES = 0)

2.8   8-bit Timer/Counter 5 (TC5)

2.8.1 Configuration
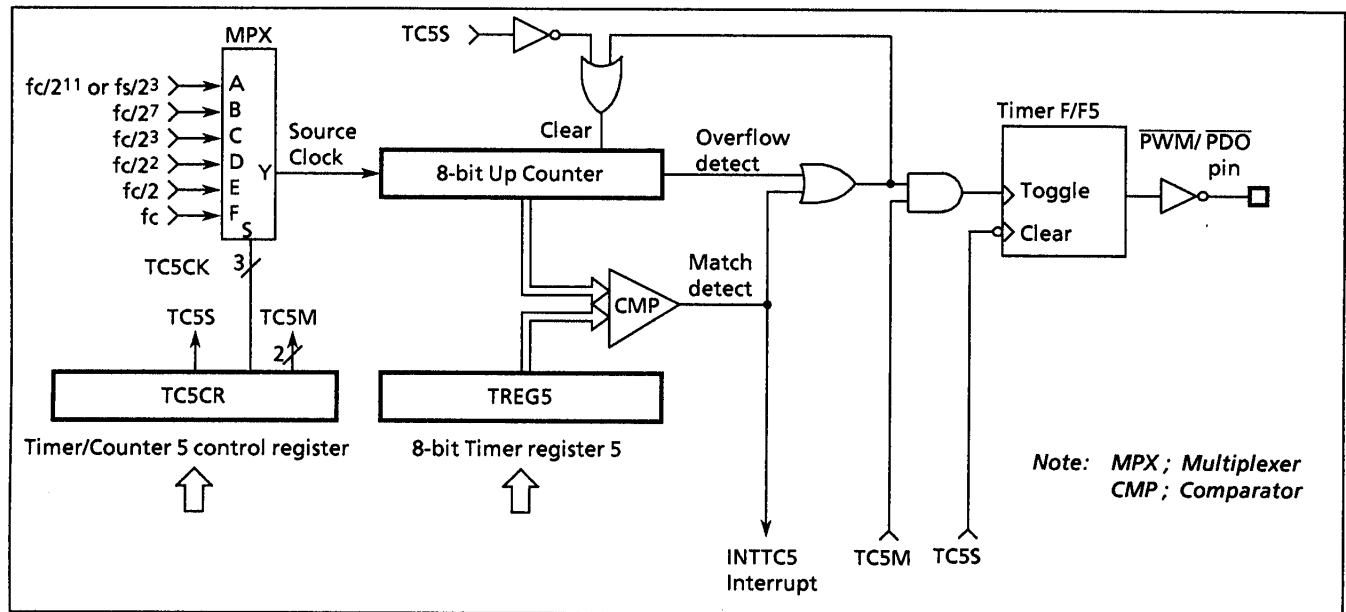
Figure 2-30. Timer/Counter 5 (TC5)

2.8.2 Control

The TC5 is controlled by a timer/counter 5 control register (TC5CR) and an 8-bit timer register 5 (TREG5).

TC5M	TC5 Operating mode select	00: Timer mode 01: Reserved 10: Progrmmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode	
TC5CK	TC5 Source clock select	000: Reserved 001: Internal clock $fc/2^{11}$ or $fs/2^3$ [Hz] 010: Internal clock $fc/2^7$ 011: Internal clock $fc/2^3$ 100: Internal clock $fc/2^2$ 101: Internal clock $fc/2$ 110: Internal clock fc 111: Reserved	Write-only
TC5S	TC5 Start control	0: Stop and clear 1: Start	

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

Note 2:   The set value of timer register must satisfy the following conditions.
   (a) When in PWM output mode, $5 < TREG5 < 251$
   (b) When in any other mode than PWM output mode, $0 < TREG5$

Note 3:   Source clock $fc/2^2$, $fc/2$, and fc cannot be used except in PWM output mode.

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

Figure 2-31. Timer/Counter 5 Timer register, Control register

2.8.3 Function

TC5 has 3 operating modes: timer, programmable divider output, and pulse width modulation output mode.

(1) Timer mode

In this mode, the internal clock is used for counting up. The contents of the timer register 5 (TREG5) is compared with the contents of the up-counter. Matching with TREG5 generates a timer/counter 5 interrupt (INTTC5) and clears the counter. Counting up resumes after the counter is cleared.

Table 2-6. Source Clock (Internal clock) for TC5

Source clock			Resolution		Maximum setting time	
NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode	fc = 8 MHz	fs = 32.768 kHz	fc = 8 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$ [Hz]	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	256 [μs]	244.14 [μs]	65.3 [ms]	62.3 [ms]
$fc/2^7$	$fc/2^7$	–	16 [μs]	–	4.1 [ms]	–
$fc/2^3$	$fc/2^3$	–	1 [μs]	–	255 [μs]	–

(2) Programmable divider output (PDO) mode

The internal clock is used for counting up. The contents of the TREG5 are compared with the contents of the up-counter. The timer F/F5 output is toggled and the counter is cleared each time a match is found. The timer F/F5 output is inverted and output to the $\overline{PDO}$ (P41) pin. In the case of $\overline{PDO}$ output, set the P41 output latch to "1" and configure as an output with P4CR1. This mode can be used for 50% duty pulse output. INTTC5 interrupt is generated each time the $\overline{PDO}$ output is toggled.

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

```
SET    (P4).1                ;  P41 output latch←1
LD     (TC5CR), 00000010B    ;  Sets to TC5 modes and source clock
LD     (TREG5), 10H          ;  (1/1024 ÷ 2⁷/fc) ÷ 2 = 10H
LD     (TC5CR), 00010010B    ;  Starts TC5
```
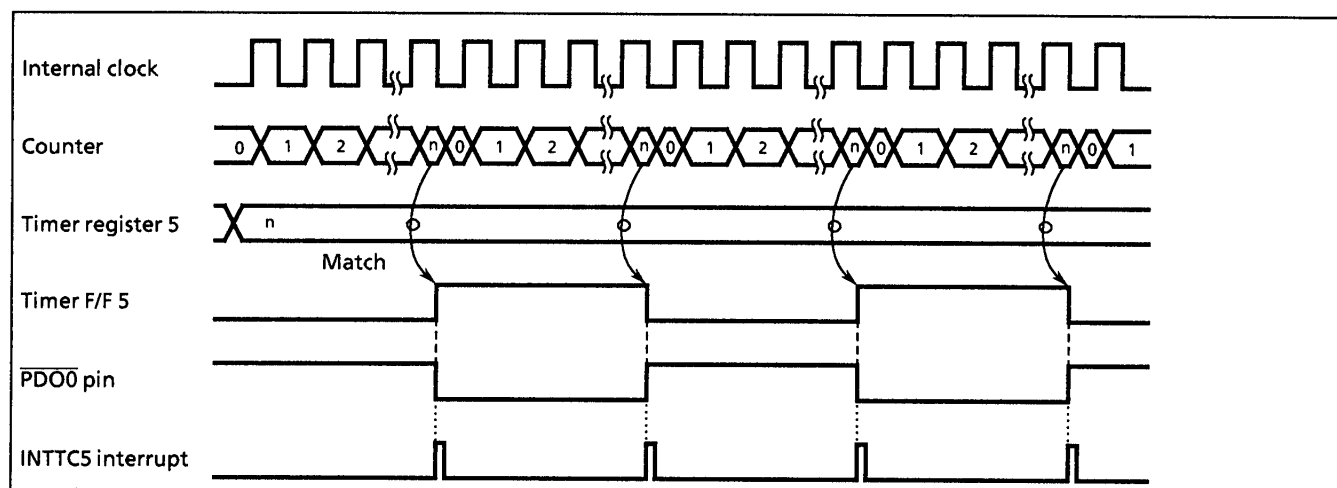

Figure 2-32. PDO Mode Timing Chart

(3) **Pulse width modulation (PWM) output mode**

PWM output with a resolution of 8-bits is possible. The internal clock is used for counting up. The contents of the TREG5 is compared with the contents of the up-counter. If a match is found, the timer F/F5 output is toggled. The counter continues counting and, when an overflow occurs, the timer F/F5 is again toggled and the counter is cleared. The timer F/F5 output is inverted and output to the $\overline{PWM}$ (P41) pin. In the case of $\overline{PWM}$ output, set the P41 output latch to "1" and configure as an output with P4CRI. An INTTC5 interrupt is generated when an overflow occurs.

TREG5 is configured a 2-stage shift register and, during output, will not switch until one output cycle is completed even if TREG5 is overwritten; therefore, output can be altered continuously. Also, the first time, TREG5 is shifted by setting TC5S (bit 5 in TC5CR) to "1" after data are loaded to TREG5.

Note 1:   *Do not overwrite TREG5 only when an INTTC5 interrupt is generated. Usually, TREG5 is overwritten in the routine of INTTC5 interrupt service.*

Note 2:   *PWM output mode can be used only in the NORMAL 1, 2 and IDLE1, 2 mode.*
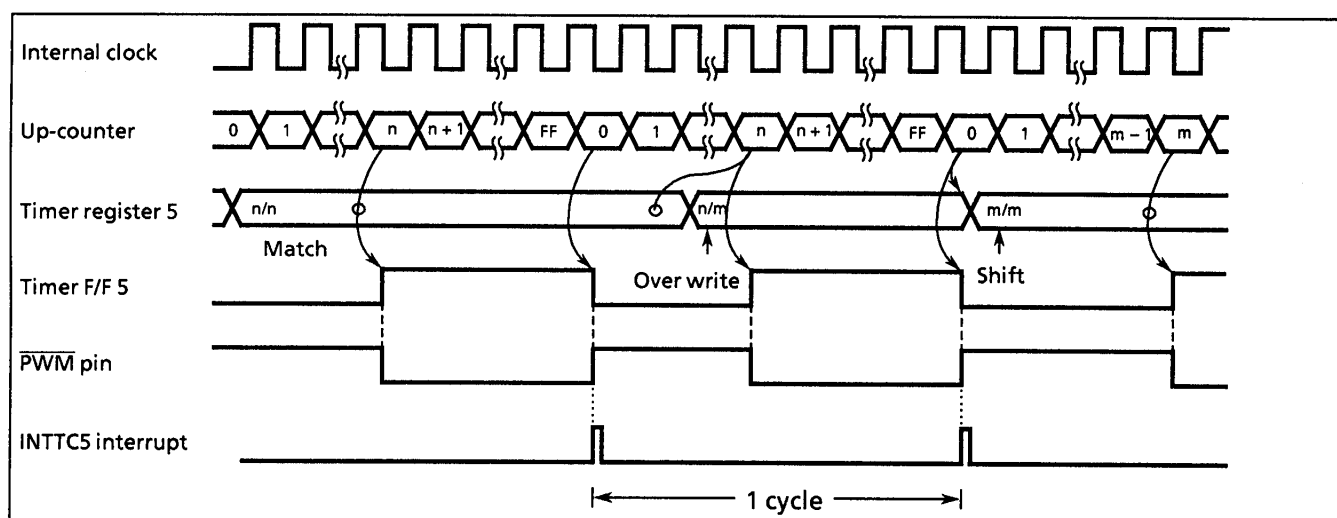
Figure 2-33. PWM Output Mode Timing Chart

Table 2-7. PWM Output Mode

Source clock		Resolution		Repeat cycle	
NORMAL1/2, IDLE1/2 mode		At fc = 8MHz	At fc = 4.194304 MHz	At fc = 8MHz	At fc = 4.194304 MHz
DV7CK = 0	DV7CK = 1				
$fc/2^2$ [Hz]		500 [ns]	953.7 [ns]	128 [μs]	244 [μs]
fc/2		250 [ns]	476.8 [ns]	64 [μs]	122 [μs]
fc		125 [ns]	238.4 [ns]	32 [μs]	61 [μs]

2.9 Serial Interface (SIO1, SIO2)

The TMP87EP26 has two clocked-synchronous 8-bit serial interfaces (SIO1 and SIO2). Each serial interface has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

The serial interfaces are connected to external devices via pins P44 (SO1), P43 (SI1), P42 ($\overline{SCK1}$) for SIO1 and P47 (SO2), P46 (SI2), P45 ($\overline{SCK2}$) for SIO2. The serial interface pins are also used as port P4. When used as serial interface pins, the output latches of these pins should be set to "1". In the transmit mode, pins P43 and P46 can be used as normal I/O ports, and in the receive mode, the pins P44 and P47 can be used as normal I/O ports.

2.9.1 Configuration

The SIO1 and SIO2 have the same configuration, except for the addresses/bit positions of the control/status registers and buffer registers.
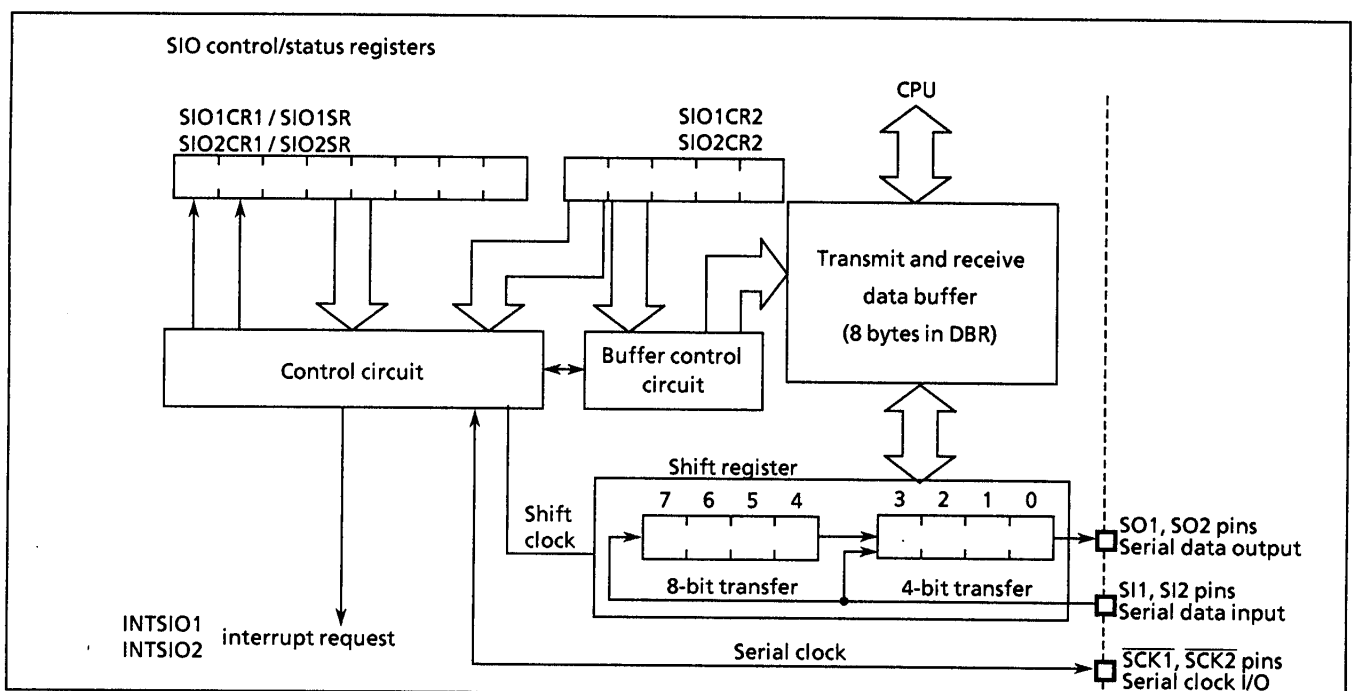
Figure 2-34. Serial Interfaces

2.9.2 Control

The serial interfaces are controlled by SIO control registers (SIO1CR1/SIO1CR2 or SIO2CR1/SIO2CR2). The serial interface status can be determined by reading SIO status registers (SIO1SR or SIO2SR).

The transmit and receive data buffer is controlled by the BUF (bits 2-0 in SIO1CR2/SIO2CR2). The data buffer is assigned to addresses 0FF0$_H$ to 0FF7$_H$ for SIO1 or 0FF8$_H$ to 0FFF$_H$ for SIO2 in the DBR area, and can continuously transfer up to 8 words (bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO1 or INTSIO2) is generated.

When the internal clock is used as the serial clock in the 8-bit receive mode and the 8-bit transmit/receive mode, a fixed interval wait can be applied to the serial clock for each word transferred. Four different wait times can be selected with WAIT (bits 4 and 3 in SIO1CR2/SIO2CR2).

SIO1, SIO2 Control Registers 1

SIO1CR1 (0020H) SIO2CR1 (0022H)					
7 6 5 4 3 2 1 0					
SIOS	SIO INH	SIOM		SCK	(Initial value: 0000 0000)

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

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

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

Note 3: SIO1CR1/SIO2CR1 are write-only registers, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO1, SIO2 Status Registers

SIO1SR (0020H) SIO2SR (0022H)			
7 6 5 4 3 2 1 0			
SIOF	SEF	"1" "1" "1" "1" "1" "1"	

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

SIO1, SIO2 Control Registers 2

SIO1CR2 (0021H) SIO2CR2 (0023H)			
7 6 5 4 3 2 1 0			
	WAIT	BUF	(Initial value: ***0 0000)

WAIT	Wait control	Always sets "00" except 8-bit transmit/receive mode. 00: $T_f = T_D$ (non-wait) 01: $T_f = 2T_D$ 10: $T_f = 4T_D$ (wait) 11: $T_f = 8T_D$	Write-only
BUF	Number of transfer words	Buffer address used $\qquad$ SIO1 $\qquad$ SIO2 000: 1 word transfer $\quad$ 0FF0H $\qquad$ 0FF8H 001: 2 words transfer $\quad$ 0FF0 to 0FF1H $\quad$ 0FF8 to 0FF9H 010: 3 words transfer $\quad$ 0FF0 to 0FF2H $\quad$ 0FF8 to 0FFAH 011: 4 words transfer $\quad$ 0FF0 to 0FF3H $\quad$ 0FF8 to 0FFBH 100: 5 words transfer $\quad$ 0FF0 to 0FF4H $\quad$ 0FF8 to 0FFCH 101: 6 words transfer $\quad$ 0FF0 to 0FF5H $\quad$ 0FF8 to 0FFDH 110: 7 words transfer $\quad$ 0FF0 to 0FF6H $\quad$ 0FF8 to 0FFEH 111: 8 words transfer $\quad$ 0FF0 to 0FF7H $\quad$ 0FF8 to 0FFFH	

Note 1:   T_f;   frame time,   T_D;   data transfer time

$\overline{SCK}$ pin

T_D

T_f
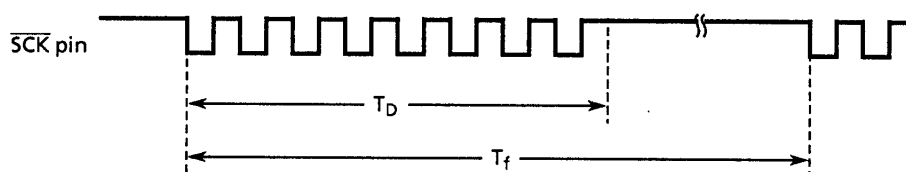
Note 2:   The lower 4 bits of each buffer are used during 4-bit transfers. Zeros (0) are stored to the upper 4 bits when receiving.

Note 3:   Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. For example, in the case of SIO1, the first buffer address transmitted is $0FF0_H$.

Note 4:   The value to be loaded to BUF is held after transfer is completed.

Note 5:   SIO1CR2/SIO2CR2 must be set when the serial interface is stopped (SIOF = 0).

Note 6:   SIO1CR2/SIO2CR2 are write-only registers, which cannot access any of in read-modify-write instruction such as bit operate, etc.

Note 7:   *; Don't care

Figure 2-35. SIO Control Registers and Status Registers

(1) Serial Clock

a. Clock Source

SCK (bits 2 to 0 in SIO1CR1/SIO2CR1) is able to select the following:

① Internal Clock

Any of four frequencies can be selected. The serial clock is output to the outside on the $\overline{SCK1}$ / $\overline{SCK2}$ pin. The $\overline{SCK}$ pin goes high when transfer starts.

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

Table 2-8. Serial Clock Rate

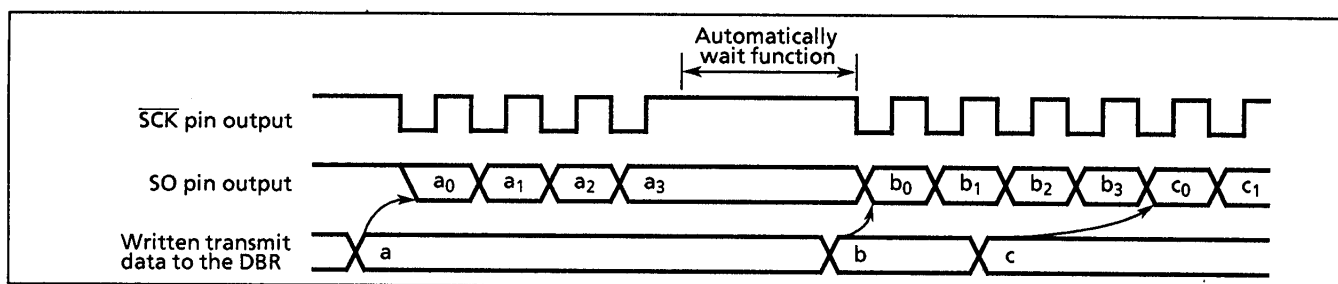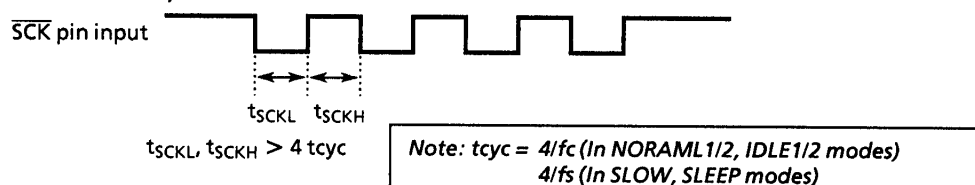| Serial clock			Maximum transfer rate	
NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode	At fc = 8 MHz	At fs = 32.768 kHz
DV7CK = 0	DV7CK = 1			
$fc/2^{13}$ [Hz]	$fs/2^5$ [Hz]	$fs/2^5$ [Hz]	0.954 Kbit/s	1 Kbit/s
$fc/2^8$	$fc/2^8$	–	30.5	–
$fc/2^6$	$fc/2^6$	–	122	–
$fc/2^5$	$fc/2^5$	–	244	–

Note: 1 Kbit = 1024 bit

Figure 2-36. Clock Source (Internal Clock)

② External Clock

An external clock connected to the $\overline{SCK1}$ / $\overline{SCK2}$ pin is used as the serial clock. In this case, the P42 ($\overline{SCK1}$) / P45 ($\overline{SCK2}$)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 244 Kbit/s. (at fc = 8 MHz).

$t_{SCKL}, t_{SCKH} > 4\,tcyc$

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

b. Shift edge

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

① Leading Edge

Transmitted data are shifted on the leading edge of the serial clock (falling edge of the $\overline{SCK}$ 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).
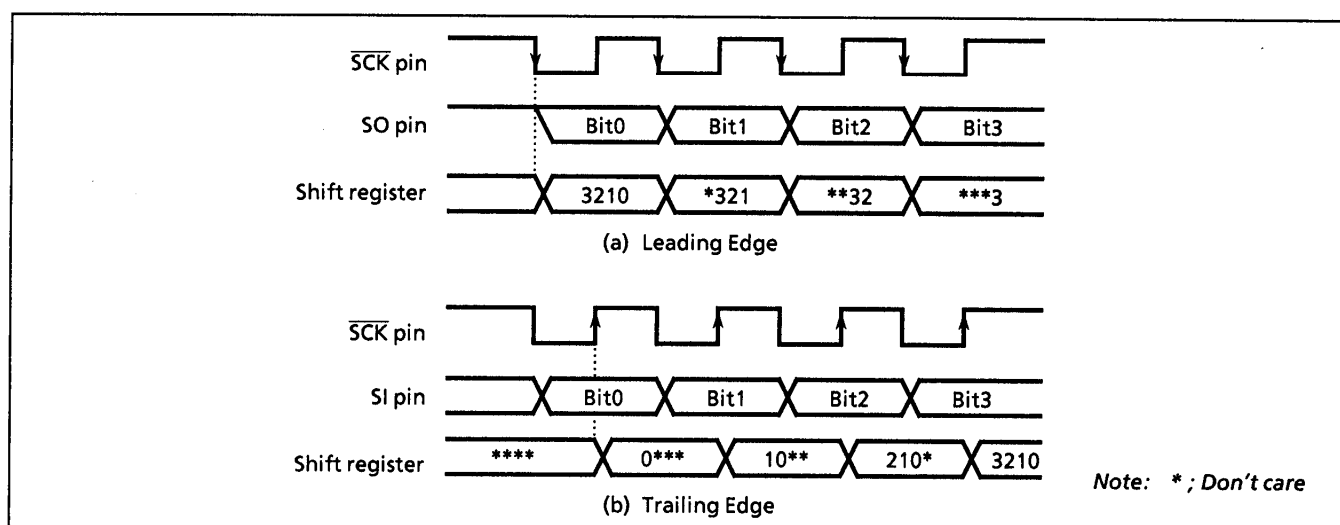

Note: * ; Don't care

Figure 2-37. 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.
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. The number of words can be changed during automatic-wait operation of an internal clock. In this case, the serial interface is not required to be stopped.
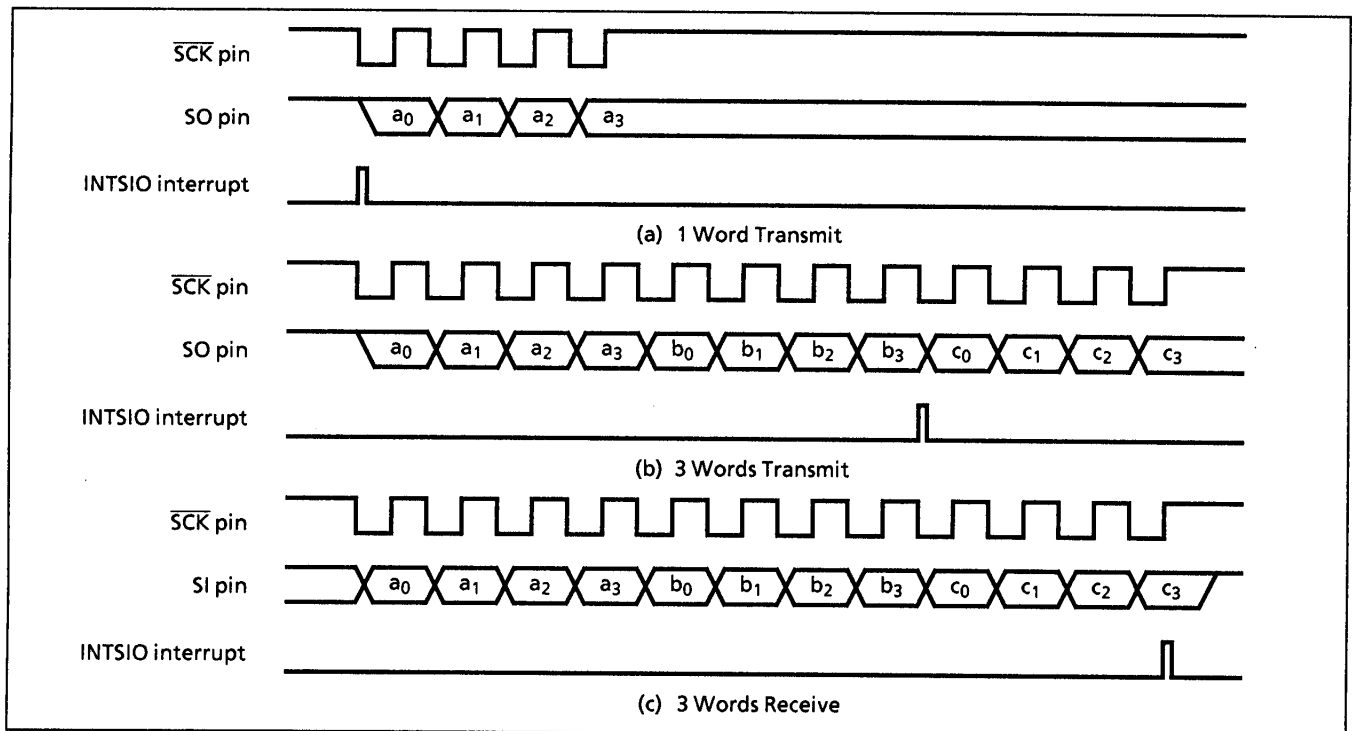

Figure 2-38. Number of Bits to Transfer (Example: 4-bit serial transfer)

(4) Transfer Mode

SIOM (bits 5 to 3 in SIO1CR1/SIO2CR1) is used to select the transmit, receive, or transmit/receive mode.

a. 4-bit and 8-bit Transmit Modes

In these modes, the SIO1CR1/SIO2CR1 is set to the transmit mode and then the data to be transmitted first are written to the data buffer registers (DBR). After the data are written, the transmission is started by setting SIOS to "1". The data are then output sequentially to the SO pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO (buffer empty) interrupt is generated to request the next transmitted data.

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

> Note: Automatic-waits are also canceled by writing to a DBR not being used as a transmit data buffer register; therefore, during SIO do not use such DBR for other applications. For example, when 3 words are transmitted, do not use the DBR of the remained 5 words.

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 "high" output from the SO pin holds final bit of the last data until falling edge of the $\overline{SCK}$.

The transmission is ended by clearing SIOS to "0" or setting SIOINH to "1" in buffer empty interrupt service program. That the transmission has ended can be determined from the status of SIOF (bit 7 in SIO1SR/SIO2SR) because SIOF is cleared to "0" when a transfer is completed.

When SIOHIN is set, the transmission is immediately ended and SIOF is cleared to "0".

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

When an external clock is used, it is also necessary to clear SIOS to "0" before shifting the next data; otherwise, dummy data will be transmitted and the operation will end.
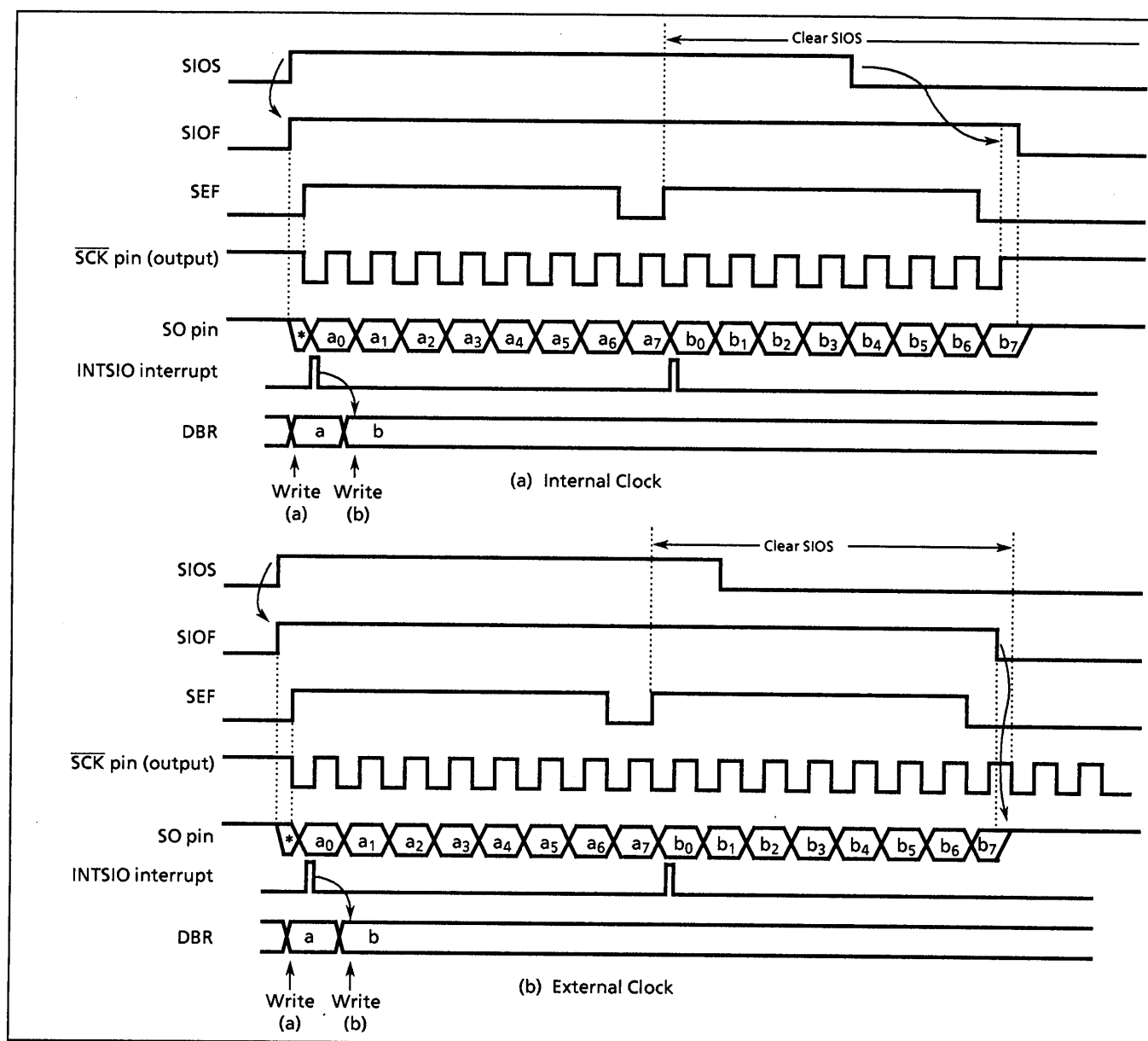
Figure 2-39. Transfer Mode (Example: 8-bit, 1 Word Transfer)

t_{SODH} = Min. 3.5/fc [s] (In the NORMAL1/2, IDLE1/2 modes)
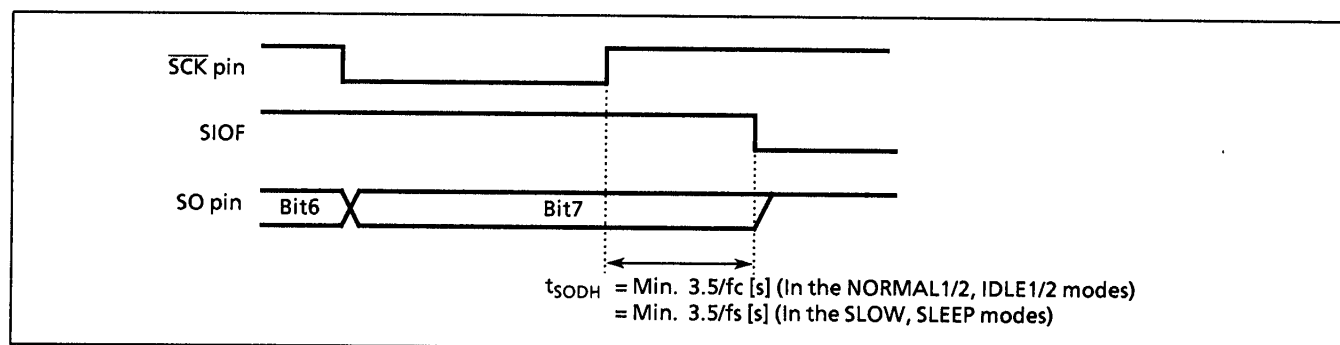= Min. 3.5/fs [s] (In the SLOW, SLEEP modes)

Figure 2-40. Transmitted Data Hold Time at end of transmit

b.   4-bit and 8-bit Receive Modes
After setting the control registers to the receive mode, set SIOS to "1" to enable receiving. The data are then transferred to the shift register via the SI pin in synchronous with the serial clock. When one word of data has been received, it is transferred from the shift register to the data buffer register (DBR). When the number of words specified with the BUF has been received, an INTSIO (buffer full) interrupt is generated to request that these data be read out. The data are then read from the data buffer registers by the interrupt service program.

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

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

When an external clock is used, the shift operation is synchronized with the external clock; therefore, the previous data are read before the next data are transferred to the data buffer register. If the previous data have not been read, the next data will not be transferred to the data buffer register and the receiving of any more data will be canceled. When an external clock is used, the maximum transfer speed is determined by the delay between the time when the interrupt request is generated and when the data received have been read.

The receiving is ended by clearing SIOS to "0" or setting SIOINH to "1" in buffer full interrupt service program. When SIOINH is set, the receiving is immediately ended SIOF is cleared to "0". When SIOS is cleared, the current data are transferred to the buffer. After SIOS cleaned, the transmissions is ended at the time that the final bit of the data being shifted has been output. That the transmission has ended can be determined from the status of SIOF (bit 7 in SIOSR). SIOF is cleared to "0" when the transmission is ended. After confirmed the receiving termination, the final receiving data is read. When SIOINH is set, the transmission is immediately ended and SIOF is cleared to "0". (The received data is ignored, and it is not required to be read out.) If the number of words is to be changed during transfer, SIOS must be cleared to "0" and BUF is rewritten after SIOF is determined to be cleared to "0" during automatic-wait operation of an external clock operation. The number of words can be changed in an internal clock. In this case, BUF must be rewritten before the received data is read out.

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

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

> *Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIOS to "0", read the last data and then switch the transfer mode.*
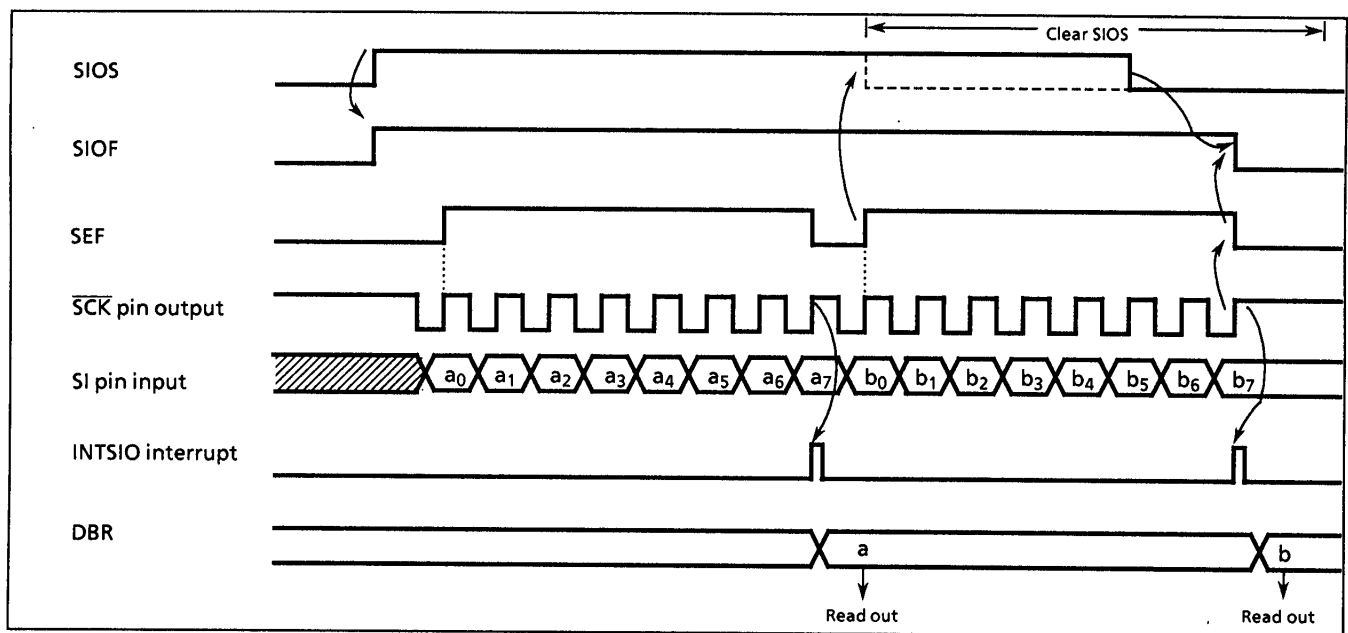
Figure 2-41. Receive Mode (Example: 8-bit, 1 word, internal clock)

c.   8-bit Transmit/Receive Mode

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

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

A wait will not be initialed if even one data ward has been written.

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 receive is started, after the SIOF goes "high"output from the SO pin holds final bit of the last data until falling edge of the SCK.

The transmit/receive operation is ended by clearing SIOS to "0" or setting SIOINH to "1" in interrupt service program.

When SIOS is cleared, the current data are transferred to the data buffer register. The transmit/receive is ended at the time that the final bit of the data being shifted has been output. The end of transmit/receive can be determined from the status of SIOF (bit 7 in SIOSR). SIOF is cleared to "0" when the transmit/receive is ended. When SIOINH is set, the transmit/receive is immediately ended and SIOF is cleared to "0".

If the number of words is to be changed during transfer, SIOS must be cleared to "0" and BUF is rewritten after confirmed that SIOF clearing to "0". The number of words can be changed during automatic-wait operation of an internal clock. In this case, BUF must be rewritten before the final transmitted/received data is read out.

When SIO1NH is set, the transmit/receive operation is immediately ended and SIOF is cleared to "0".

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

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

> Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIOS to "0", read the last data and then switch the transfer mode.
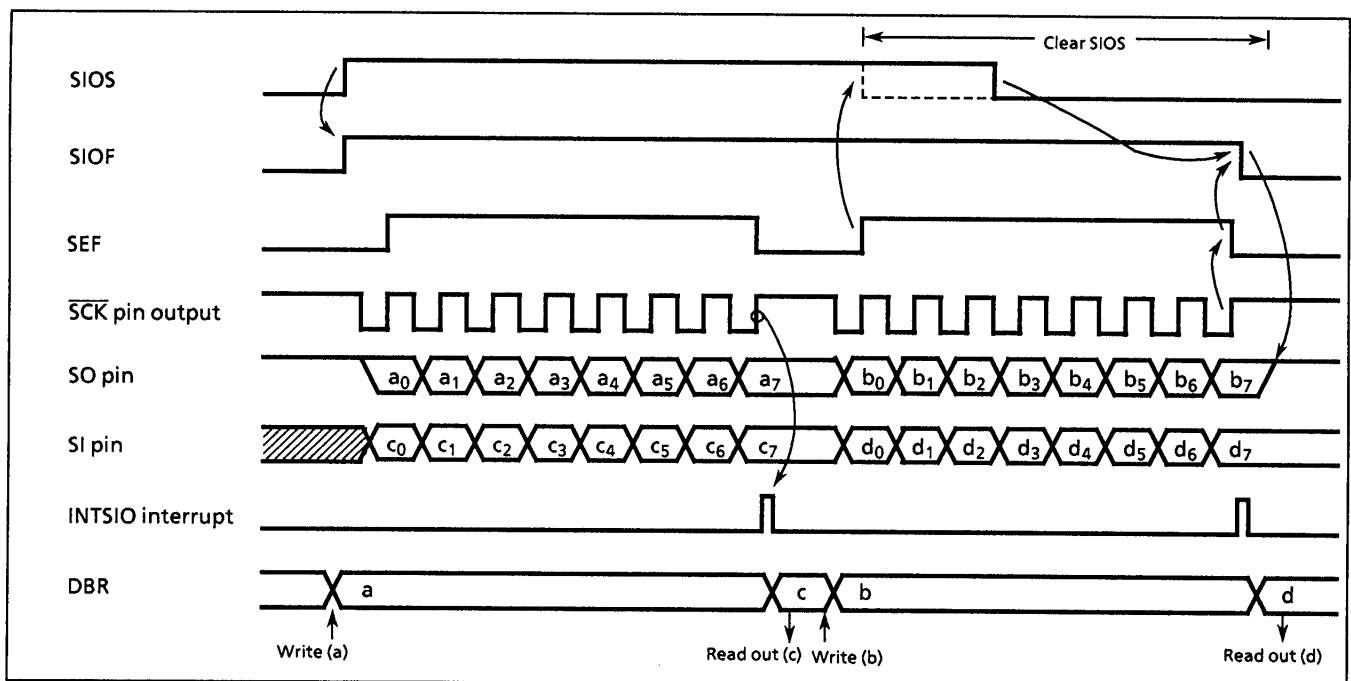
Figure 2-42. Transmit/Receive Mode (Example: 8-bit, 1word, internal clock)

t_{SODH} = Min. 4/fc [s] (In the NORMAL1/2, IDLE1/2 modes)
= Min. 4/fs [s] (In the SLOW, SLEEPmodes)
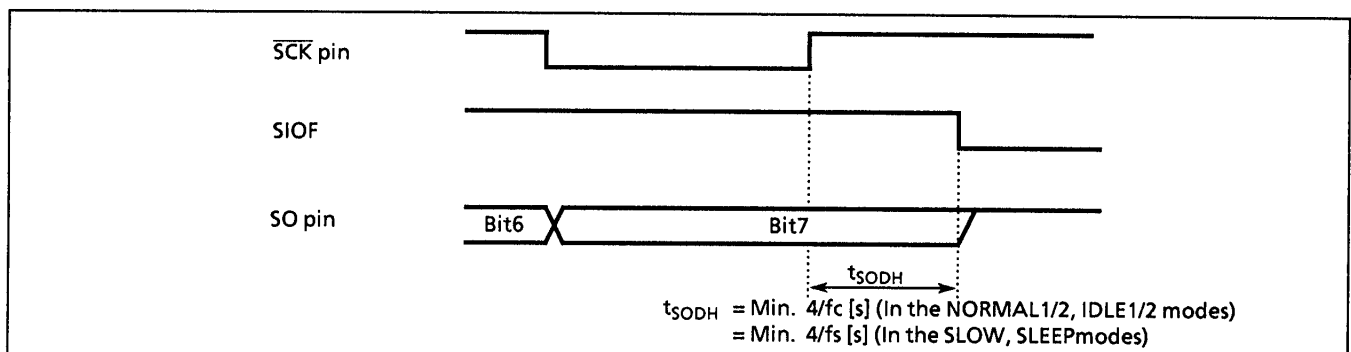
Figure 2-43. Transmitted Data Hold Time at end of transmit/receive

2.10   LCD Driver

The TMP87EP26 has a driver and control circuit to directly drive the liquid crystal device (LCD). The pins to be connected to LCD are as follows:

① Segment output port   12 pins (SEG11 to SEG0)
② Segment output or P6, P7, P8, P9 input / output port   28 pins (SEG39 to SEG12)
③ Common output port   4 pins (COM3 to COM0)

In addition, VLC pin is provided as the drive power pins

The devices that can be directly driven is selectable from LCD of the following drive methods:

① 1 / 4 Duty   (1 / 3 Bias) LCD .... Max 160 Segments (8-segment × 20 digits)
② 1 / 3 Duty   (1 / 3 Bias) LCD .... Max 120 Segments (8-segment × 15 digits)
③ 1 / 3 Duty   (1 / 2 Bias) LCD .... Max 120 Segments (8-segment × 15 digits)
④ 1 / 2 Duty   (1 / 2 Bias) LCD .... Max 80 Segments (8-segment × 10 digits)
⑤ Static LCD   .... Max 40 Segments (8-segment × 5 digits)

2.10.1   Configuration
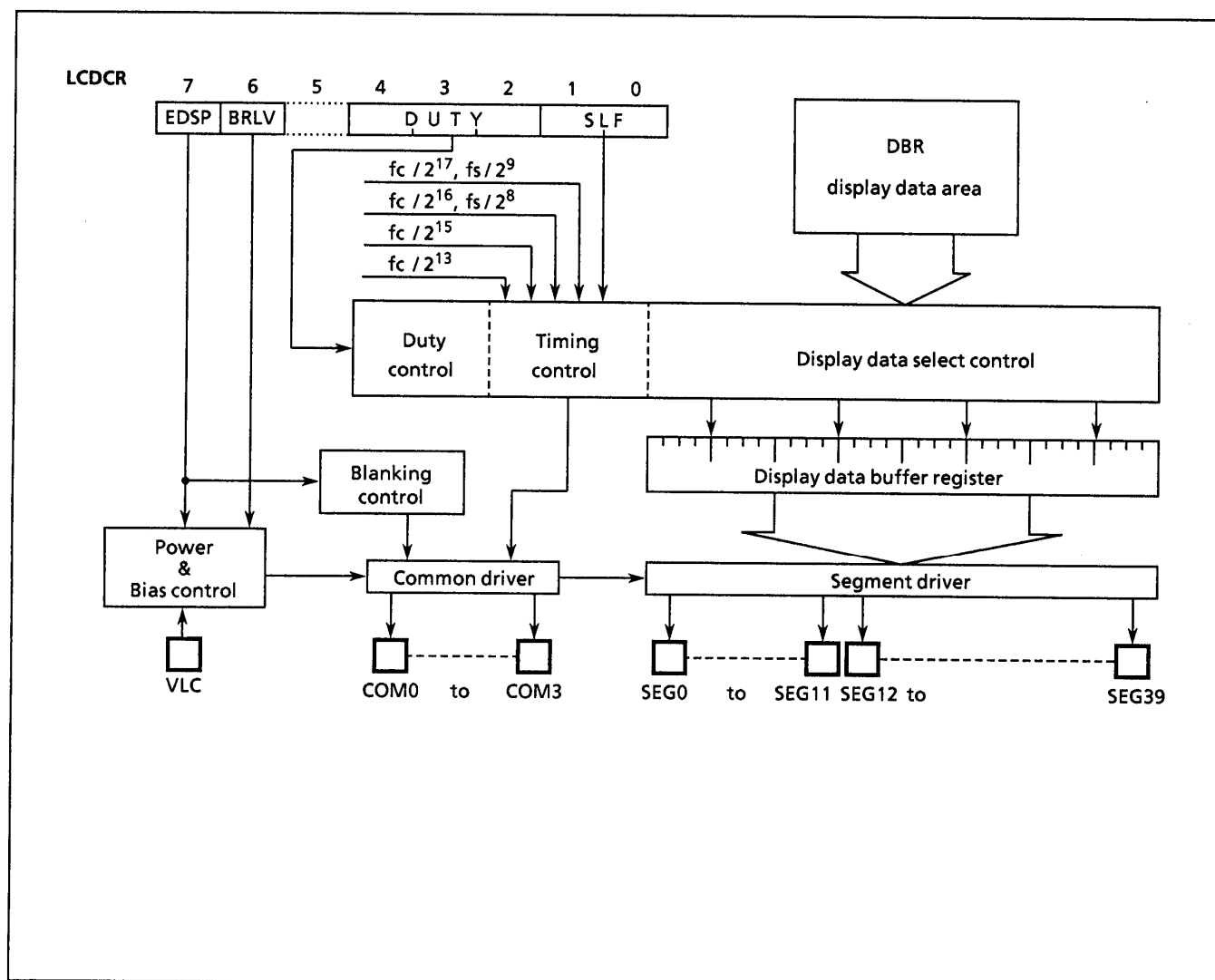
Figure 2-44. LCD driver

2.10.2 Control

The LCD driver is controlled by the LCD control register (LCDCR).

7	6	5	4	3	2	1	0	
EDSP	BRLV		D U T Y			S L F		(Initial value: 00*0 0000)

LCDCR (0028H)

EDSP	LCD Display Control	0: Blanking 1: Enables LCD display (Blanking is released)	
BRLV	Selection of LCD driver resistance	0: Low resistance (20 kΩ) 1: Low and high resistance (20/200 kΩ)	Write-only
DUTY	Selection of driving methods	000: 1 / 4 Duty (1 / 3 Bias) 001: 1 / 3 Duty (1 / 3 Bias) 010: 1 / 3 Duty (1 / 2 Bias) 011: 1 / 2 Duty (1 / 2 Bias) 100: Static 101: reserved 11*: reserved	
SLF	Selection of LCD base frequency	00: $fc / 2^{17}$ or $fs / 2^9$ [Hz] 01: $fc / 2^{16}$ or $fs / 2^8$ 10: $fc / 2^{15}$ 11: $fc / 2^{13}$	

Note 1: fc: high-frequency clock, fs: low-frequency clock
Note 2: Set BRLV in accordance with LCD panel.
When BRLV is set to "0", strong driving power is obtained by low resistance (typ. 20 kΩ), but a driver current consumption is increased. When BRLV is set to "1", a driver current consumption is restrained by high resistance (typ. 200 kΩ) and low resistance (typ. 20 kΩ). However, if a LCD panel requiring strong driving power is connected, the quality of display may not reach the required level. For details, please refer to Figure 2-46.

Figure 2-45. LCD driver control register

Note: R_L: Low resistance, R_H: High resistance

Figure 2-46. Selecting split resistance with BRLV setting (1/3 bias)

(1) LCD driving methods

As for LCD driving method, 5 types can be selected by DUTY (bit 4 to bit 2 of LCDCR). The driving method is initialized in the initial program according to the LCD used.
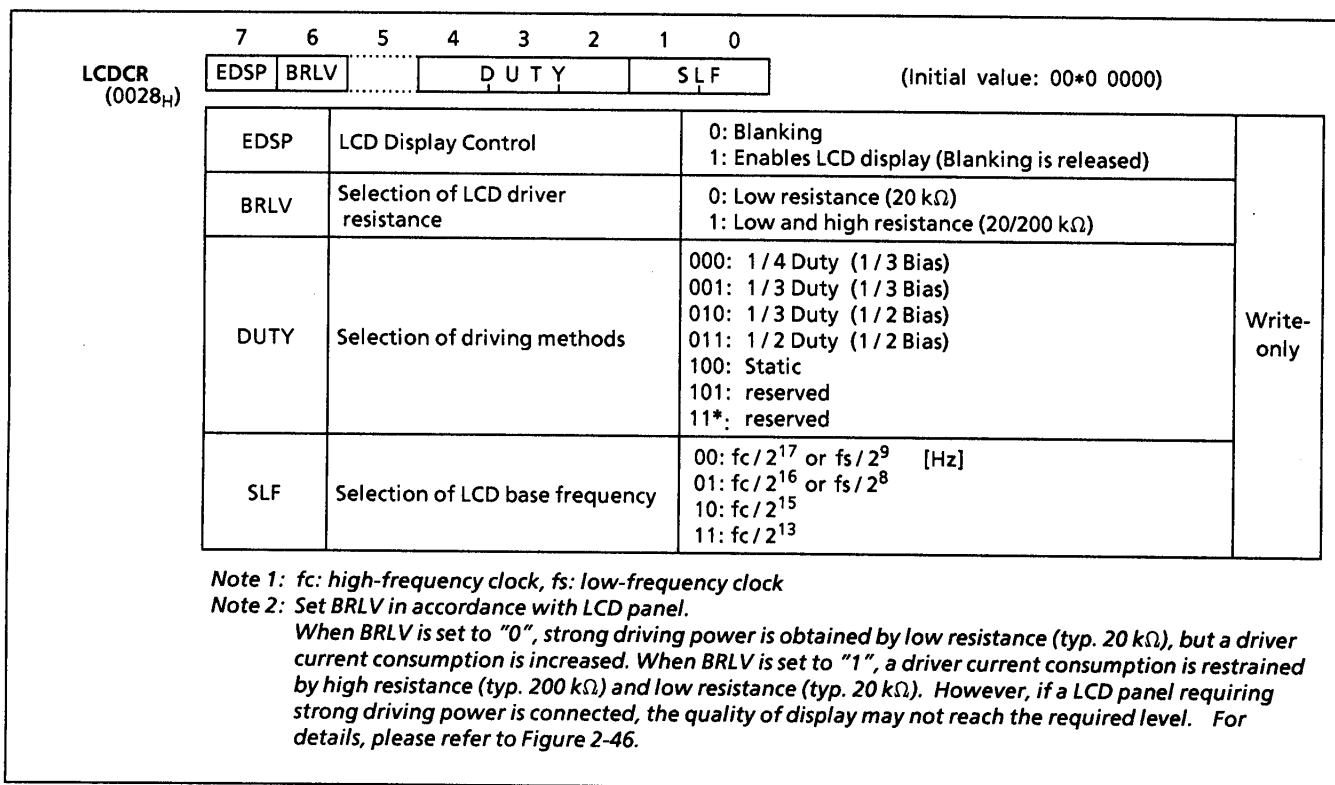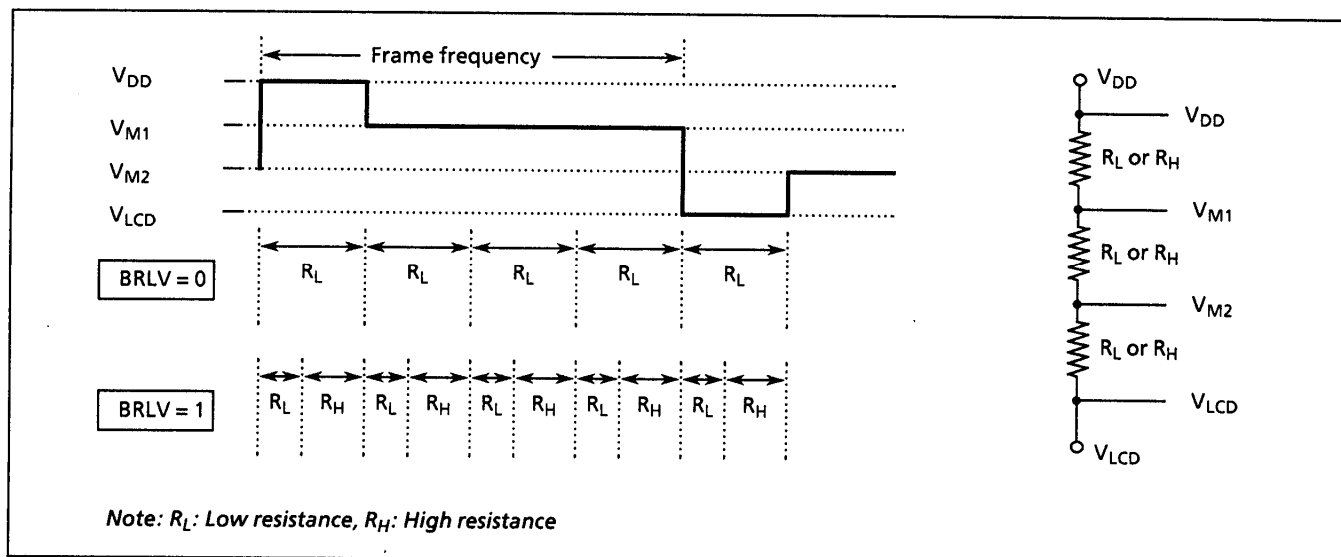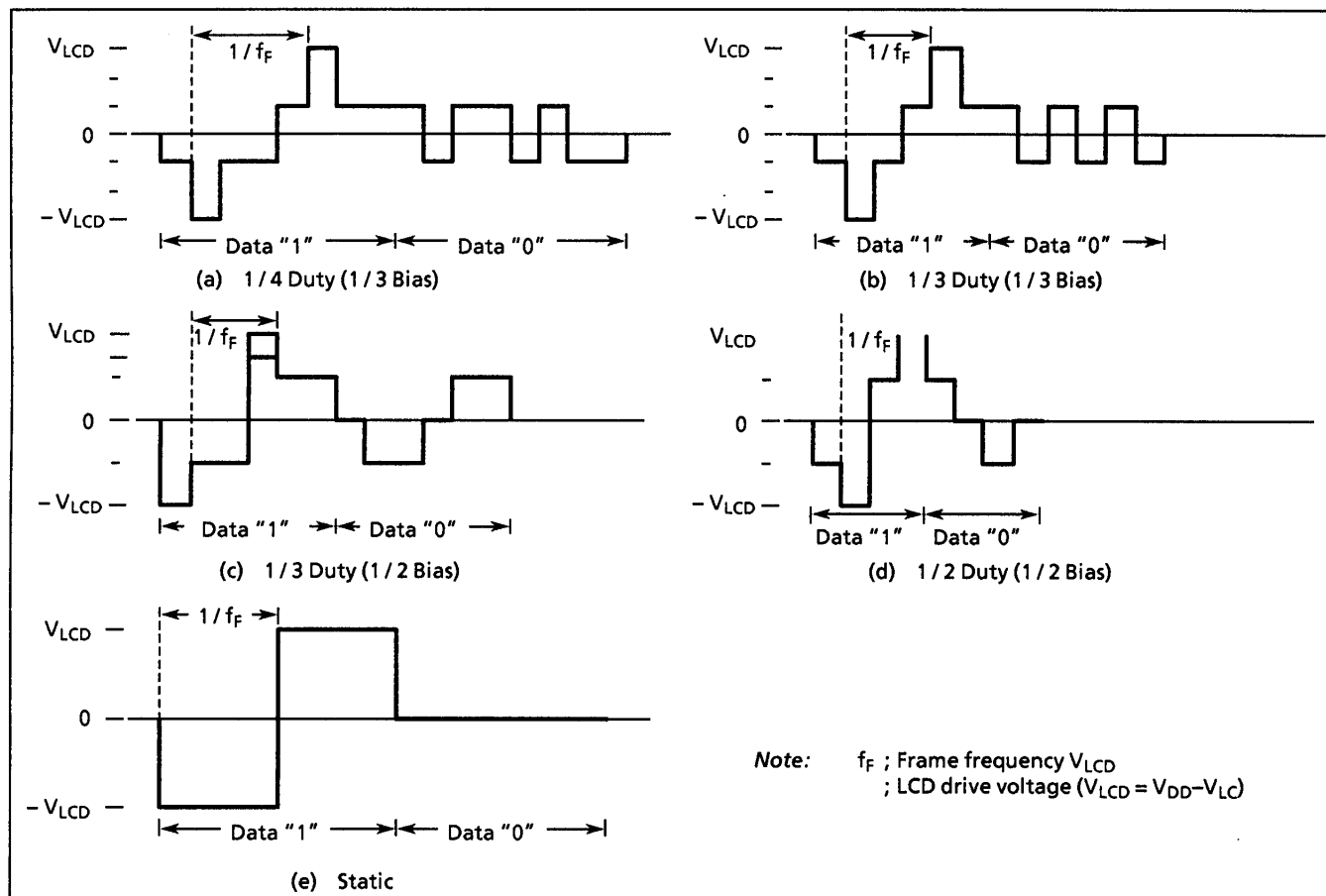

Figure 2-47. LCD drive waveform (COM – SEG pins)

(2) Frame frequency

Frame frequency (f$_F$) is set according to driving method and base frequency as shown in the following Table 2-9. The base frequency is selected by SLF (Lower two bits of command register) according to the frequency fc and fs of the basic clock to be used.

a. At the single clock mode or at the dual clock mode with DVCK7 = 0

SLF	Base frequency [Hz]	Frame frequency [Hz]			
		1 / 4 Duty	1 / 3 Duty	1 / 2 Duty	Static
00	$\dfrac{fc}{2^{17}}$	$\dfrac{fc}{2^{17}}$	$\dfrac{4}{3}\cdot\dfrac{fc}{2^{17}}$	$\dfrac{4}{2}\cdot\dfrac{fc}{2^{17}}$	$\dfrac{fc}{2^{17}}$
	(fc = 8 MHz)	61	81	122	61
01	$\dfrac{fc}{2^{16}}$	$\dfrac{fc}{2^{16}}$	$\dfrac{4}{3}\cdot\dfrac{fc}{2^{16}}$	$\dfrac{4}{2}\cdot\dfrac{fc}{2^{16}}$	$\dfrac{fc}{2^{16}}$
	(fc = 4 MHz)	61	81	122	61
10	$\dfrac{fc}{2^{15}}$	$\dfrac{fc}{2^{15}}$	$\dfrac{4}{3}\cdot\dfrac{fc}{2^{15}}$	$\dfrac{4}{2}\cdot\dfrac{fc}{2^{15}}$	$\dfrac{fc}{2^{15}}$
	(fc = 4 MHz)	122	162	244	122
11	$\dfrac{fc}{2^{13}}$	$\dfrac{fc}{2^{13}}$	$\dfrac{4}{3}\cdot\dfrac{fc}{2^{13}}$	$\dfrac{4}{2}\cdot\dfrac{fc}{2^{13}}$	$\dfrac{fc}{2^{13}}$

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

b. At the dual clock mode with DVCK7 = 1 or with SYSCK = 1

SLF	Base frequency [Hz]	Frame frequency [Hz]			
		1 / 4 Duty	1 / 3 Duty	1 / 2 Duty	Static
00	$\dfrac{fs}{2^{9}}$	$\dfrac{fs}{2^{9}}$	$\dfrac{4}{3}\cdot\dfrac{fs}{2^{9}}$	$\dfrac{4}{2}\cdot\dfrac{fs}{2^{9}}$	$\dfrac{fs}{2^{9}}$
	(fs = 32.768 kHz)	64	85	128	64
01	$\dfrac{fs}{2^{8}}$	$\dfrac{fs}{2^{8}}$	$\dfrac{4}{3}\cdot\dfrac{fs}{2^{8}}$	$\dfrac{4}{2}\cdot\dfrac{fs}{2^{8}}$	$\dfrac{fs}{2^{8}}$
	(fs = 32.768 kHz)	128	171	256	128

Note: fs ; *Low-frequency clock [Hz]*

Table 2-9. Setting of LCD Frame Frequency

(3) LCD drive voltage

LCD driving voltage V$_{LCD}$ is given as potential difference V$_{DD}$ – V$_{LC}$ between pins VDD and VLC. Therefore, when the CPU voltage and LCD drive voltage are the same, VLC pin will be connected to VSS pin. The LCD lights when the potential difference between segment output and common output is ± V$_{LCD}$. Otherwise it turns off.

During reset, the power switch of LCD driver is automatically turned off, shutting off the VLC voltage. At the same time, both segment outputs and common outputs become at VDD level, turning off the LCD. The power switch is turned on to supply VLC voltage to LCD driver by setting

with EDSP (bit 7 in LCDCR) to "1". After that, the power switch will not turned off even during blanking (clearing EDSP to "0") and the VLC voltage continues flow. When STOP mode starts, the power switch will be turned off. Therefore, LCD light out, and stop operation is executed at low power consumption. When STOP mode is released the status in effect immediately before the STOP operation is reinstated.

> Note: During reset, the LCD segment outputs (SEG0 to SEG11) and LCD common outputs (COM3 to COM0) are fixed "0" level. But the multiplex terminal (P6, P7, P8 and P9 ports) of input / output port and LCD segment output becomes high impedance. Therefore, when the reset input is long remarkably, ghost problem may appear in LCD display.

2.10.3 LCD Display Operation

(1) Display data setting

Display data is stored to the display data area (assigned to address 0F80 to 0F93$_H$) in the DBR. The display data which are stored in the display data area is automatically read out and sent to the LCD driver by the hardware. The LCD driver generates the segment signal and common signal according to the display data and driving method. Therefore, display patterns can be changed by only over writing the contents of display data area by the program. Figure 2-48 shows the correspondence between the display data area and SEG/COM pins.

LCD light when display data is "1" and turn off when "0". According to the driving method of LCD, the number of pixels which can be driven becomes different, and the number of bits in the display data area which is used to store display data also becomes different.

Therefore, the bits which are not used to store display data as well as the data buffer which corresponds to the addresses not connected to LCD can be used to store general user process data (see Table 2-10.)

address	bit 7 bit 6 bit 5 bit 4	bit 3 bit 2 bit 1 bit 0
0F80$_H$	SEG1	SEG0
81	SEG3	SEG2
82	SEG5	SEG4
83	SEG7	SEG6
84	SEG9	SEG8
85	SEG11	SEG10
86	SEG13	SEG12
87	SEG15	SEG14
88	SEG17	SEG16
89	SEG19	SEG18
8A	SEG21	SEG20
8B	SEG23	SEG22
8C	SEG25	SEG24
8D	SEG27	SEG26
8E	SEG29	SEG28
8F	SEG31	SEG30
90	SEG33	SEG32
91	SEG35	SEG34
92	SEG37	SEG36
93	SEG39	SEG38
	COM3 COM2 COM1 COM0	COM3 COM2 COM1 COM0

Figure 2-48. LCD display data area (DBR)

Table 2-10. Driving method and bit for display Data

Driving methods	bit 7/3	bit 6/2	bit 5/1	bit 4/0
1 / 4 Duty	COM3	COM2	COM1	COM0
1 / 3 Duty	–	COM2	COM1	COM0
1 / 2 Duty	–	–	COM1	COM0
Static	–	–	–	COM0

Note: – ; This bit is not used for display data

(2) Blanking

Blanking is enabled when EDSP is cleared to "0".

Blanking turns off LCD through outputting a non-selective level to COM pin. A signal level is continuously output to SEG pin according to display data and driving method. For static drive, lights-out by data (clearing display data to "0") does not apply any voltage between pins COM and SEG. On the other hand, lights-out by blanking makes the output to COM pin at a constant $V_{LCD}/2$ level, so that the part between pins COM and SEG becomes in the state driven by $V_{LCD}/2$.

2.10.4 Control Method of LCD Driver

(1) Initial setting

Figure 2-49 shows the flowchart of initialization.

Example: To operate a 1/4 duty LCD of 40 segments × 4 commons at frame frequency $fc/2^{16}$ [Hz]

LD	(LCDCR), 00000001B;	Sets LCD driving method and frame frequency.
LD	(P6CR), 0FFH	; Sets P6, P7, P8, P9 port as
LD	(P7CR), 0FFH	segment output .
LD	(P8CR), 0FFH	; Sets the initial value of .
LD	(P9CR), 0FFH	display data.
LD	(LCDCR),10000001B	; Display enable
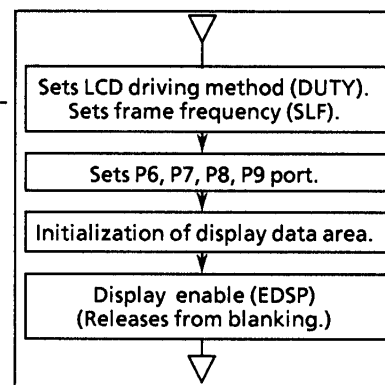
Figure 2-49. Initial Setting of LCD Driver

(2) Store of display data

Generally, display data are prepared as fixed data in program memory (ROM) and stored in display data area by load command.

Example 1:   To display using 1/4 duty LCD a numerical value which corresponds to the LCD data stored in data memory at address 80_H (when pins COM and SEG are connected to LCD as in Figure 2-50), display data become as shown in Table 2-11.

```
        LD      A, (80H)
        ADD     A, TABLE - $ - 5
        LD      HL,0F80H
        LD      (HL), (PC + A)
        RET
TABLE:  DB      11011111B, 00000110B,
                11100011B, 10100111B,
                00110110B, 10110101B,
                11110101B, 00010111B,
                11110111B, 10110111B
SNEXT:
```


Figure 2-50. Example of COM, SEG Pin Connection (1/4 Duty)

Note: DB is a byte data difinition instruction.

Table 2-11. Example of Display Data (1/4 Duty)

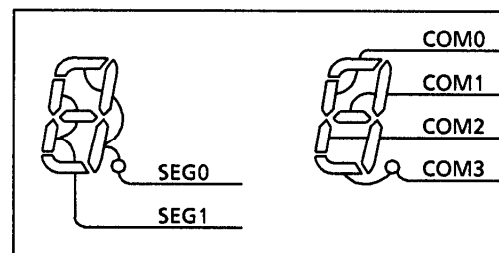| No.	Display	Display data	No.	Display	Display data
0		11011111	5		10110101
1		00000110	6		11110101
2		11100011	7		00000111
3		10100111	8		11110111
4		00110110	9		10110111

Example 2 :   Table 2-12 shows an example of display data which are displayed using 1/2 duty LCD in the same way as Table 2-11. The connection between pins COM and SEG are the same as shown in Figure 2-51.
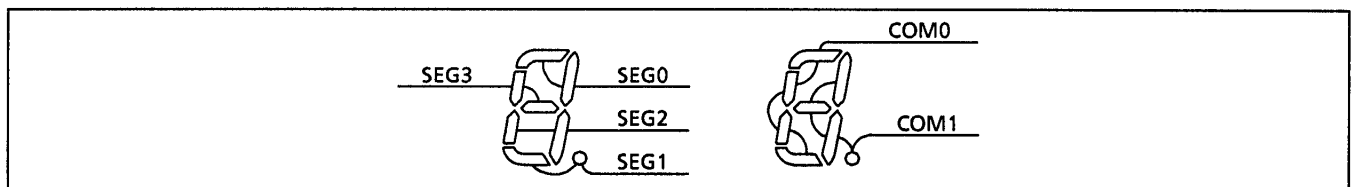
Figure 2-51. Example of COM, SEG Pin Connection

Table 2-12. Example of Display Data (1/2 Duty)

| Number | Display data | | Number | Display data | |
	High order address	Low order address		High order address	Low order address
0	**01**11	**01**11	5	**11**10	**01**01
1	**00**10	**00**10	6	**11**11	**01**01
2	**10**01	**01**11	7	**01**10	**00**11
3	**10**10	**01**11	8	**11**11	**01**11
4	**11**10	**00**10	9	**11**10	**01**11

Note: * ; Don't care

(3) Example of LCD drive output

Figure 2-53. 1/3 Duty (1/3 Bias) Drive

Figure 2-52. 1/4 Duty (1/3 Bias) Drive
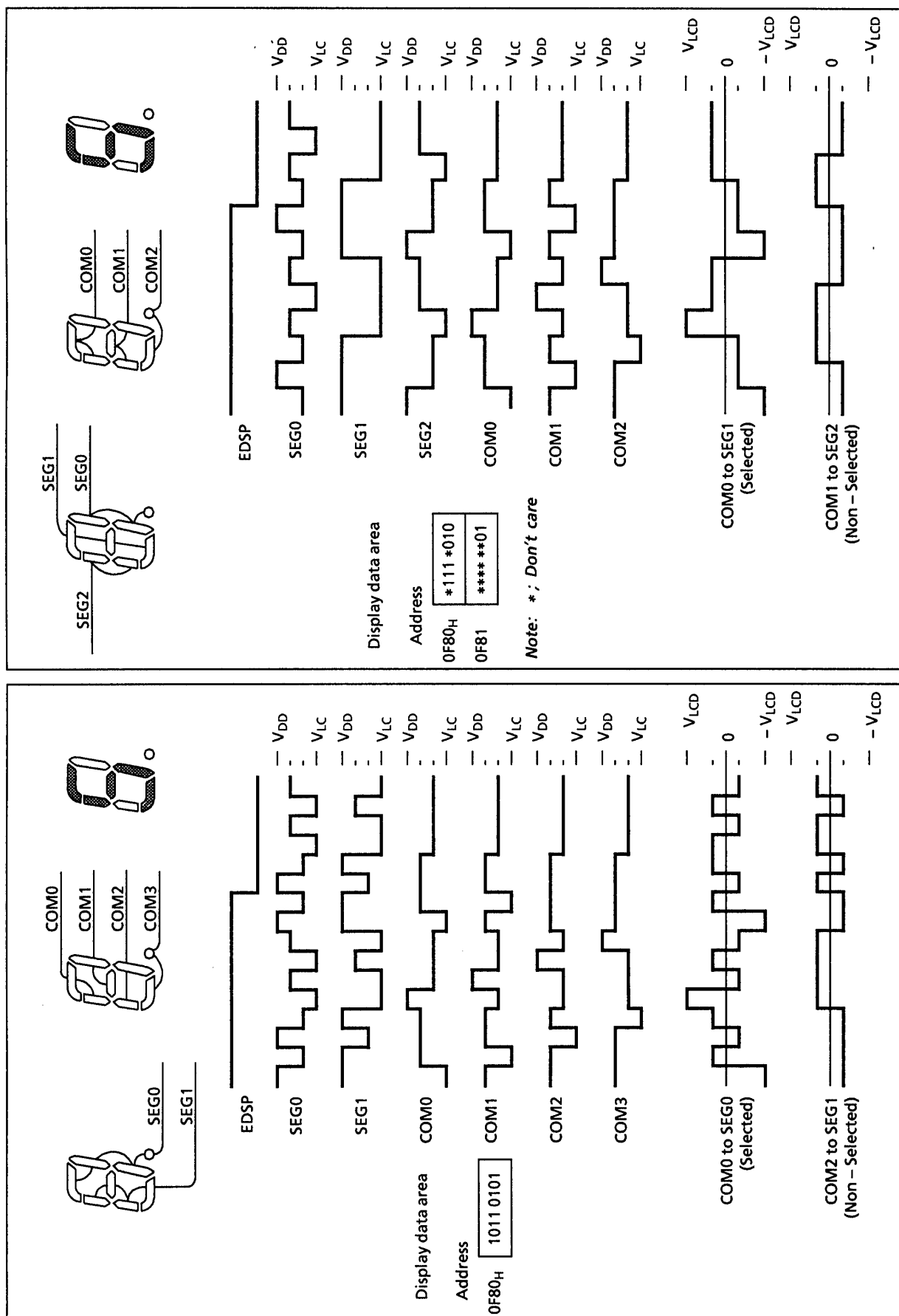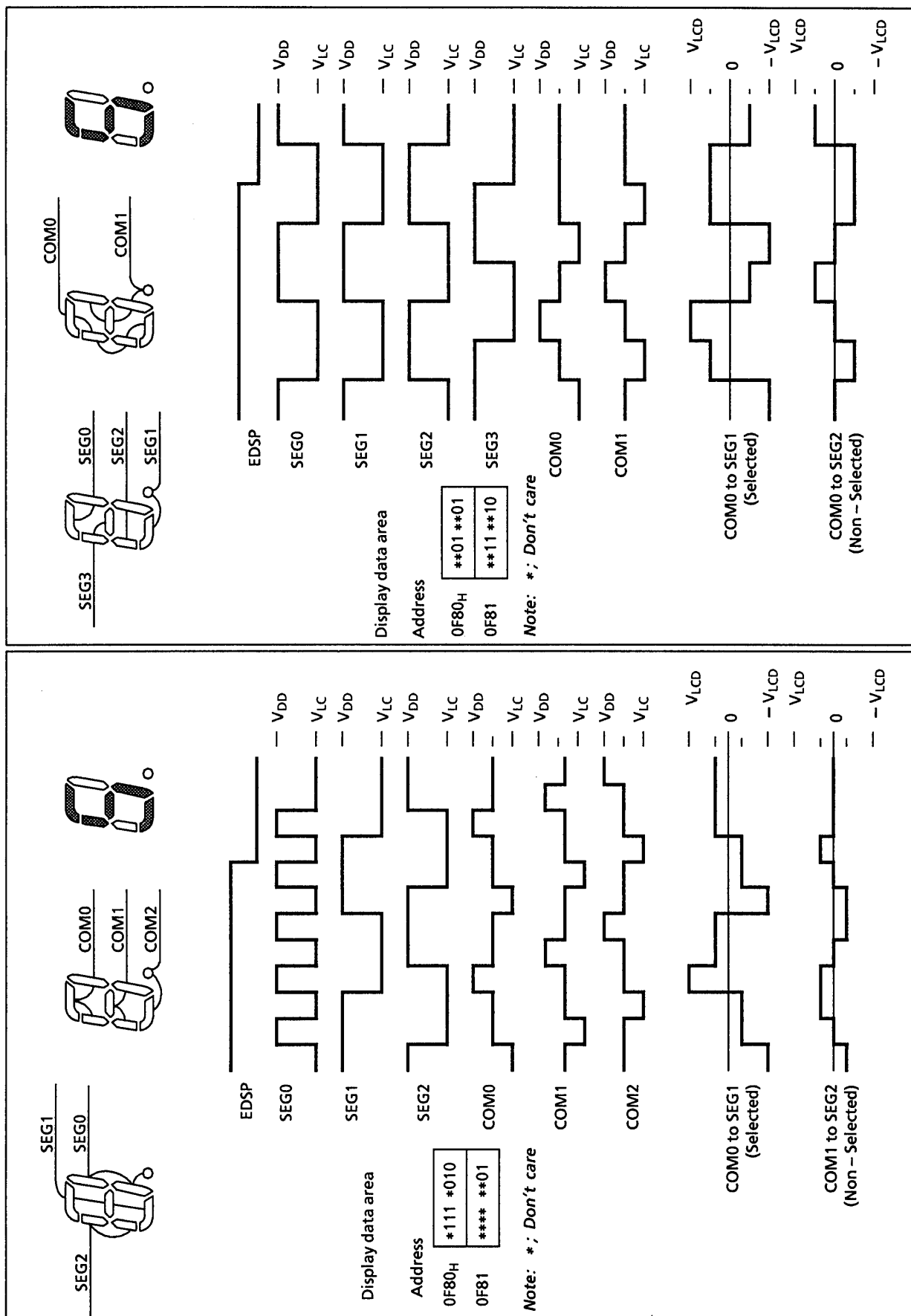
Figure 2-55. 1/2 Duty (1/2 Bias) Drive

Figure 2-54. 1/3 Duty (1/2 Bias) Drive
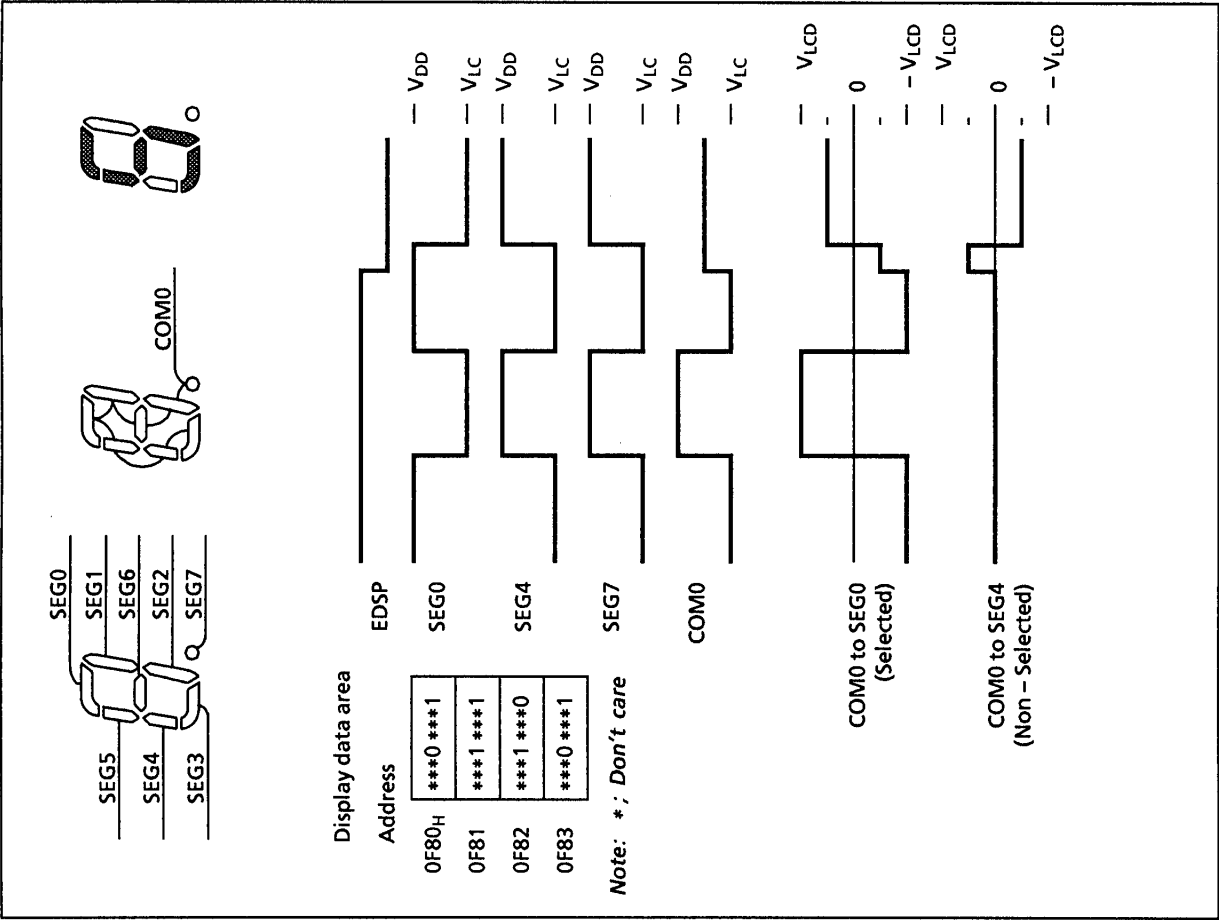
Figure 2-56. Static Drive

2.11   8-bit AD converter (ADC)

The TMP87EP26 has an 8-channel multiplexed-input 8-bit successive approximate type AD converter with sample and hold.

2.11.1   Configuration

Figure 2-57. AD converter

2.11.2   Control

The AD converter is controlled by the AD converter control register (ADCCR). Reading EOCF in ADCCR determines the AD converter operating state; reading the AD conversion value register (ADCDR) determines the AD conversion value.

Figure 2-58. AD converter Result Register

AD converter Control Register

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

SAIN	Analog input selection	0000: AIN0 0001: AIN1 0010: AIN2 0011: AIN3 0100: AIN4 0101: AIN5 0110: AIN6 0111: AIN7 1***: reserved	R/W
AINDS	Analog input control	0: Enable 1: Disable	
ACK	Conversion time selection	0: Conversion time = 23μs (at fc = 8MHz) 1: Conversion time = 92μs	
ADS	AD conversion start	0: – 1: AD conversion start	
EOCF	End of AD conversion flag	0: Under conversion or Before conversion 1: End of conversion	

Note 1: * ; Don't care
Note 2: Select analog input when AD converter stops.
Note 3: The ADS is automatically cleared to "0" after starting conversion.
Note 4: The EOCF is cleared to "0" when reading the ADCDR.
Note 5: The EOCF is read-only.
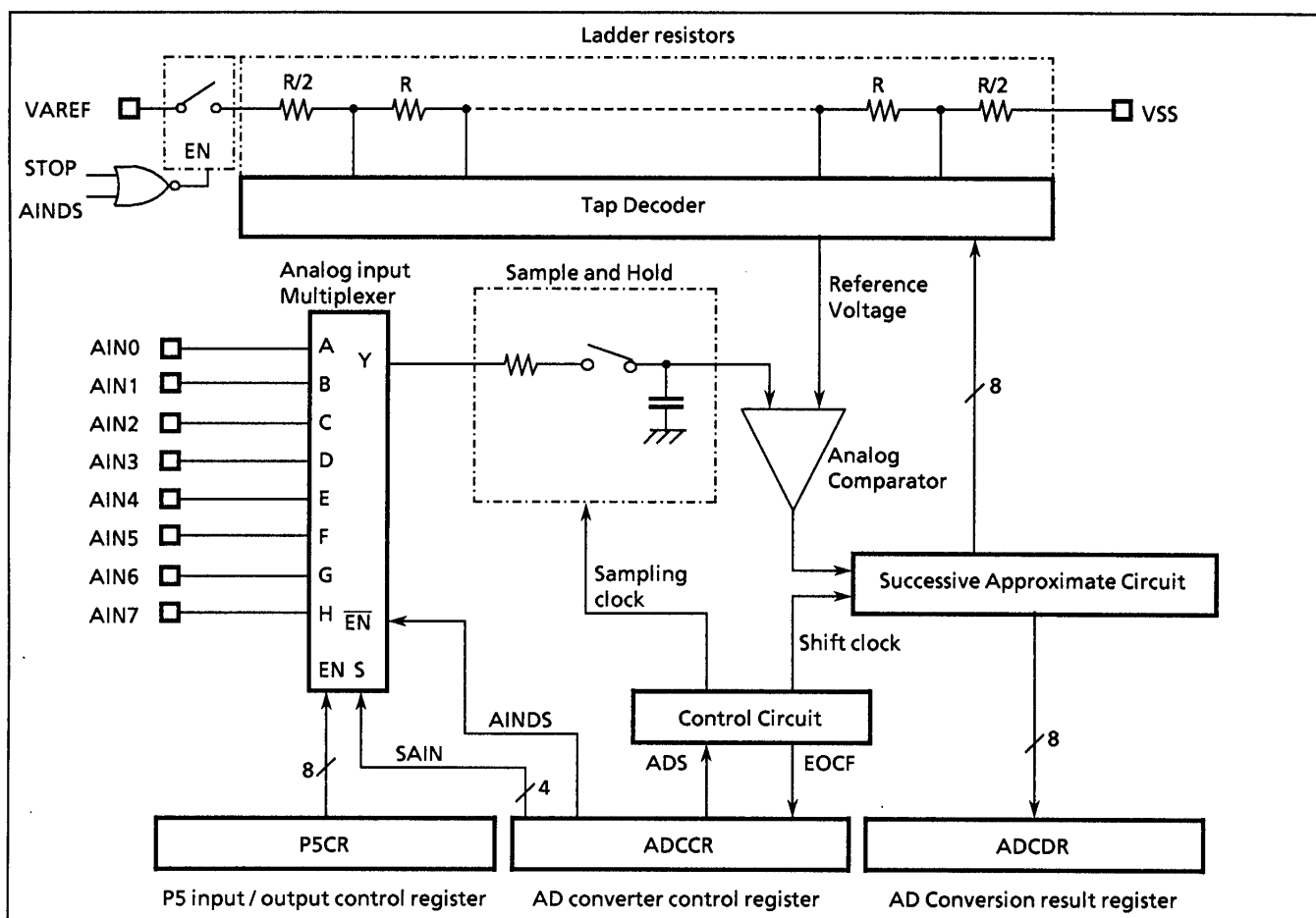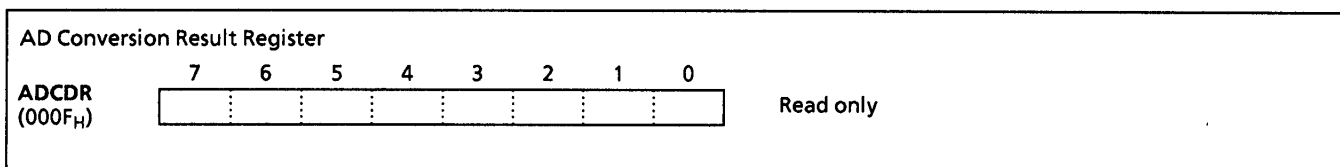
Figure 2-59. AD converter control register and AD conversion result register

2.11.3 Operation

The high side of an analog reference voltage is applied to VAREF pin, and the low side is applied to VASS pin. The reference voltage between VAREF and VASS is divided into the voltage corresponding with bits by ladder resistance. The reference voltage is compared with an analog input voltage and AD conversion is performed.

(1) Start of AD conversion

Prior to AD conversion start, select one pin among analog input channels (AIN7 to AIN0) using the SAIN (bit 3 tp 0 in ADCCR). Clear AINDS (bit 4 in ADDCCR) to 0 and clear the channel to be used for analog input using the P5 I/O control register (P5CR).

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

Set AD conversion time using the ACK (bit 5 in ADCCR).
To start AD conversion, set AD conversion to "1" using the ADS (bit 6 in ADCCR).
AD conversion time is from AD conversion start to AD conversion result being set in ADCDR. When ACK = 0, 184/fc [s] (46 machine cycles) is necessary. That is, when fc = 8 MHz, the AD conversion time is 23 μs. After AD conversion, the EOCF (bit 7 in ADCCR) is set to "1" indicating end of conversion. Setting the ADS to "1" during AD conversion resumes conversion from the beginning.
The analog input voltage is sampled every 4 machine cycles after AD conversion start.

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

(2) Reading of AD conversion result

After the end of conversion, read the conversion result from the ADCDR.

The EOCF is automatically cleared to "0" when reading the ADCDR. When the conversion result is read out during AD conversion, the invalid value is read out.

(3) AD conversion in STOP mode

When the MCU places in the STOP mode during the AD conversion, the conversion is terminated and the AD conversion value become indefinite. Thus EOCF is maintained to "0" after returned from the STOP mode.

However, if the STOP mode is started after the end of conversion (EOCF = 1), the ADCDR contents are held.
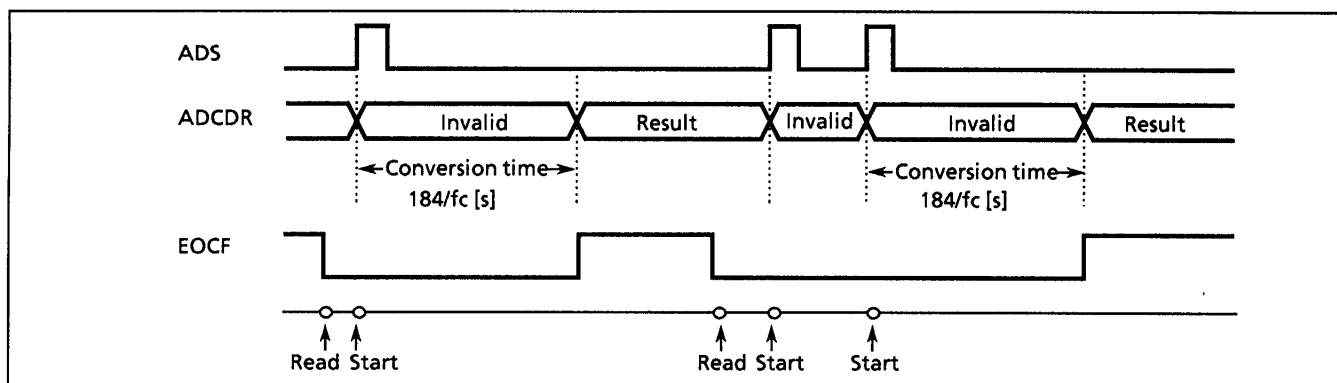
Figure 2-60.   AD conversion Timing chart

Example: After AIN pin 4 is selected as an analog input channel, AD conversion is started. EOCF is confirmed and the converted result is read out. It is saved to address 009E$_H$ in RAM.

```
          ; AIN SELECT
          LD        (ADCCR), 00000100B    ;  selects AIN4
          ; AD CONVERT START
          SET       (ADCCR). 6            ;  ADS = 1
SLOOP:    TEST      (ADCCR). 7            ;  EOCF = 1 ?
          JRS       T, SLOOP
          ; RESULT DATA READ
          LD        (9EH), (ADCDR)
```

Figure 2-61 shows the relationship between An analog input voltage and AD converted 8-bit digital value.
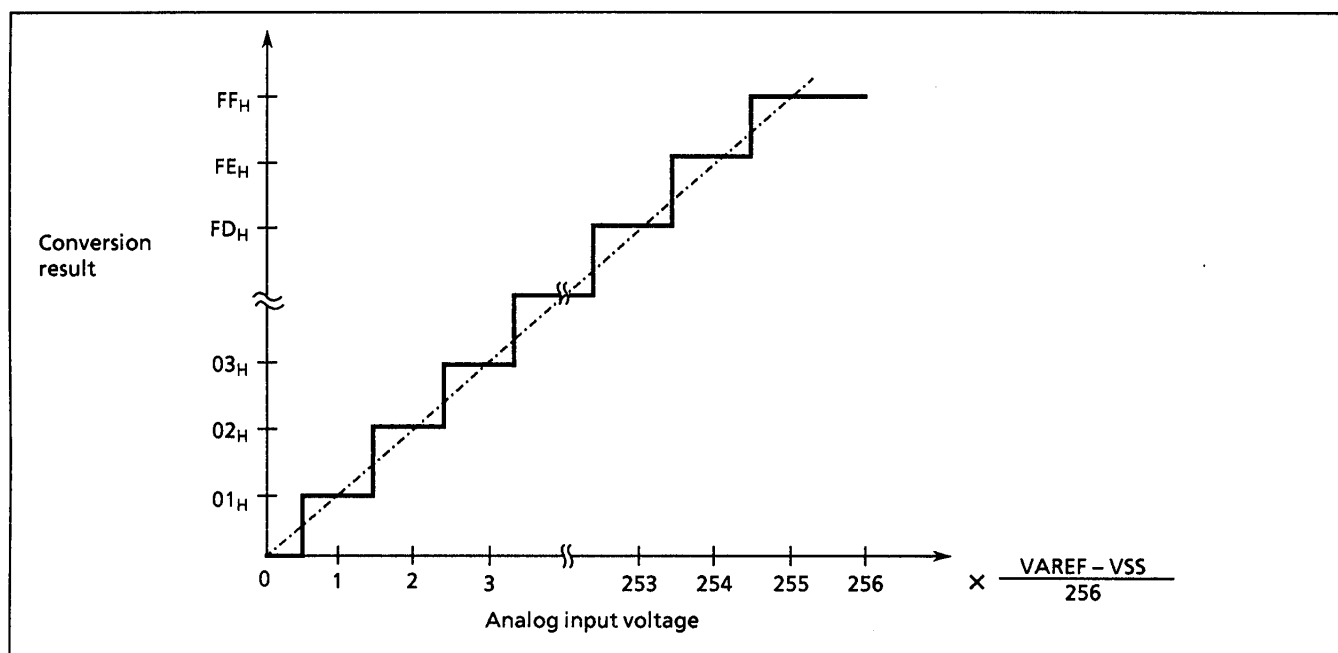
Figure2-61.   Analog input voltage vs AD conversion result (typ.)

2.12   E²PROM (Data memory area)

The TMP87EP26 incorporates a 16-Kbyte (4000$_H$ to 7FFF$_H$) E²PROM program memory and a 512-byte (1000$_H$ to 11FF$_H$) E²PROM data memory.

2.12.1 Configuration
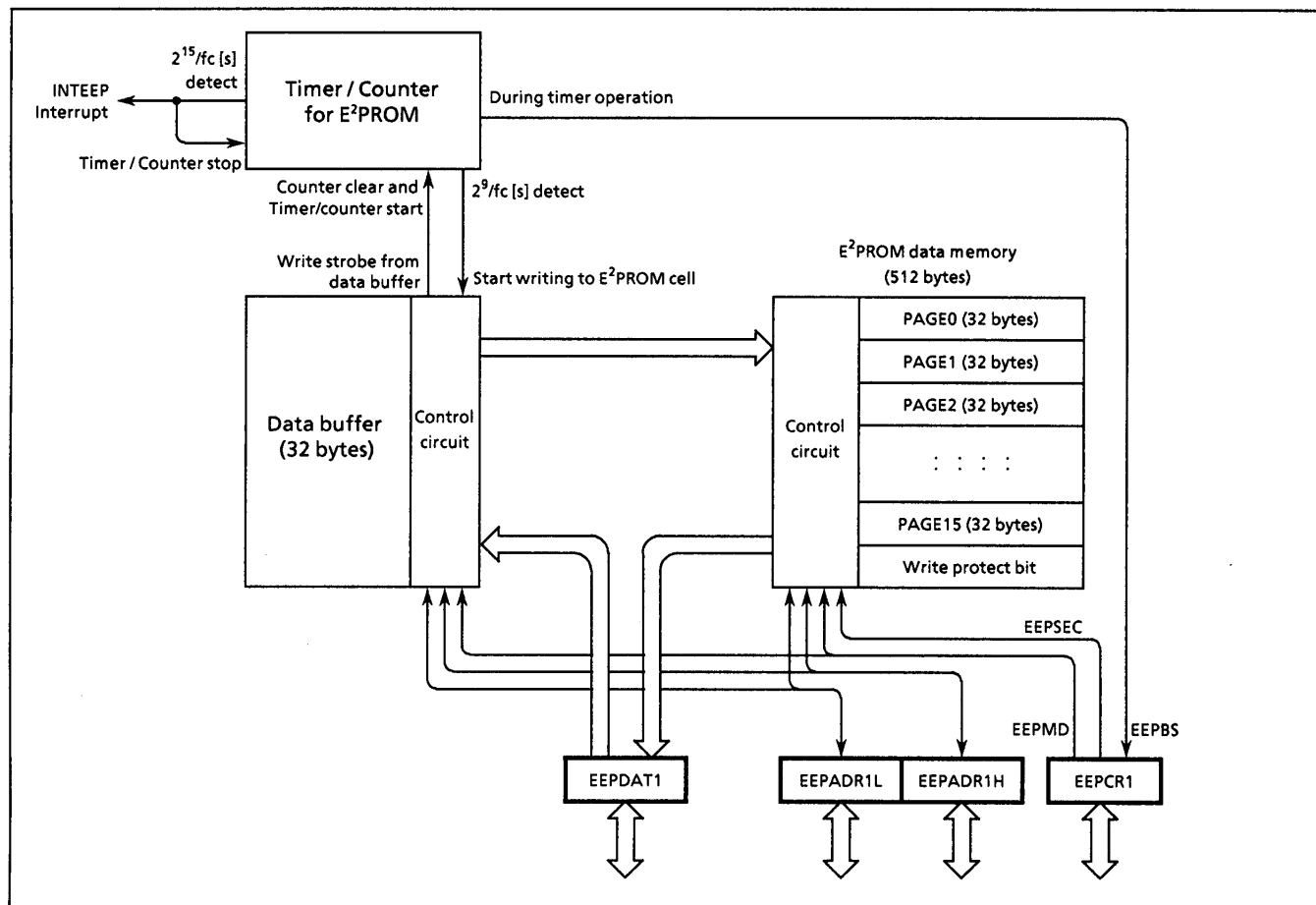
Figure 2-62. E²PROM Data Memory

2.12.2   Control

The following registers are used to access E^{2}PROM data memory area.
- E^{2}PROM Control register (EEPCR1)
- E^{2}PROM Address Control register (EEPADR1)
- E^{2}PROM Data Buffer register (EEPDAT1)

E^{2}PROM control register 1

	7	6	5	4	3	2	1	0	
EEPCR1 (0027$_H$)			EEPSEC				EEPBS	EEPMD	(Initial value: 1100 **00)

		1100: Write protect control	
EEPSEC	E^{2}PROM access control	⌈ Bit7 of EEPDAT1 is "0":Setting write protect ⌉ ⌊ Bit7 of EEPDAT1 is "1":Releasing write protect ⌋ 0101: Normal Read/Write mode Others: reserved	R/W
EEPBS	E^{2}PROM Write Busy Flag	0: Write complete or buffer free 1: Data buffer transfer in progress or E^{2}PROM write in progress	R
EEPMD	E^{2}PROM R/W Mode Select	0: Read mode 1: Write mode	R/W

Note 1: EEPBS is set to "1" immediately after the first data are written to EEPDAT1 and cleared to "0" after write to E^{2}PROM completes.

Note 2: To set/release write protect, set dummy address to EEPADR1 after setting EEPSEC to "1100" and then write data to EEPDAT1.

Note 3: When main system clock is low-frequency, E^{2}PROM should not be accessed (read from/written to).

Note 4: "reserved" cannot be accessed.

Note 5: Release write protect before writing of E^{2}PROM.

E^{2}PROM address control register 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EEPADR1 (0024$_H$, 0025$_H$) R/W	"0"	"0"	"0"	"1"	"0"	"0"	"0"			EEPADR1$_H$ (0025$_H$), EEPADR1$_L$ (0024$_H$)						

(Initial value: **** ***0 00000000)

Note 1: Write "1" to bit 12 of EEPADR1. Write "0" to bit 15 to 13 and bit 11 to 9.

Note 2: Read from E^{2}PROM is triggered by setting data in EEPADR1$_H$ (higher register). Thus, for every read, a 16-bit address must be specified first in EEPADR1$_L$, then RRPADR1$_H$.

Note 3: To set/release write protect, set any one byte between 1000$_H$ and 11FF$_H$ in EEPADR1 before data of EEPDAT1 is set.

E^{2}PROM data buffer register 1

	7	6	5	4	3	2	1	0	
EEPDAT1 (0026$_H$) R/W									(Initial value: 0000 0000)

Note 1: In E^{2}PROM Write mode, to write two or more bytes, the next data must be written to EEPDAT1 within 2^9/fc [s] (64 μs at fc = 8 MHz).

Note 2: In E^{2}PROM Write mode, transfer to the E^{2}PROM data buffer is triggered by setting data in EEPDAT1. Thus, the address to which data are to be written must be set in EEPADR1L and EEPADR1H beforehand.

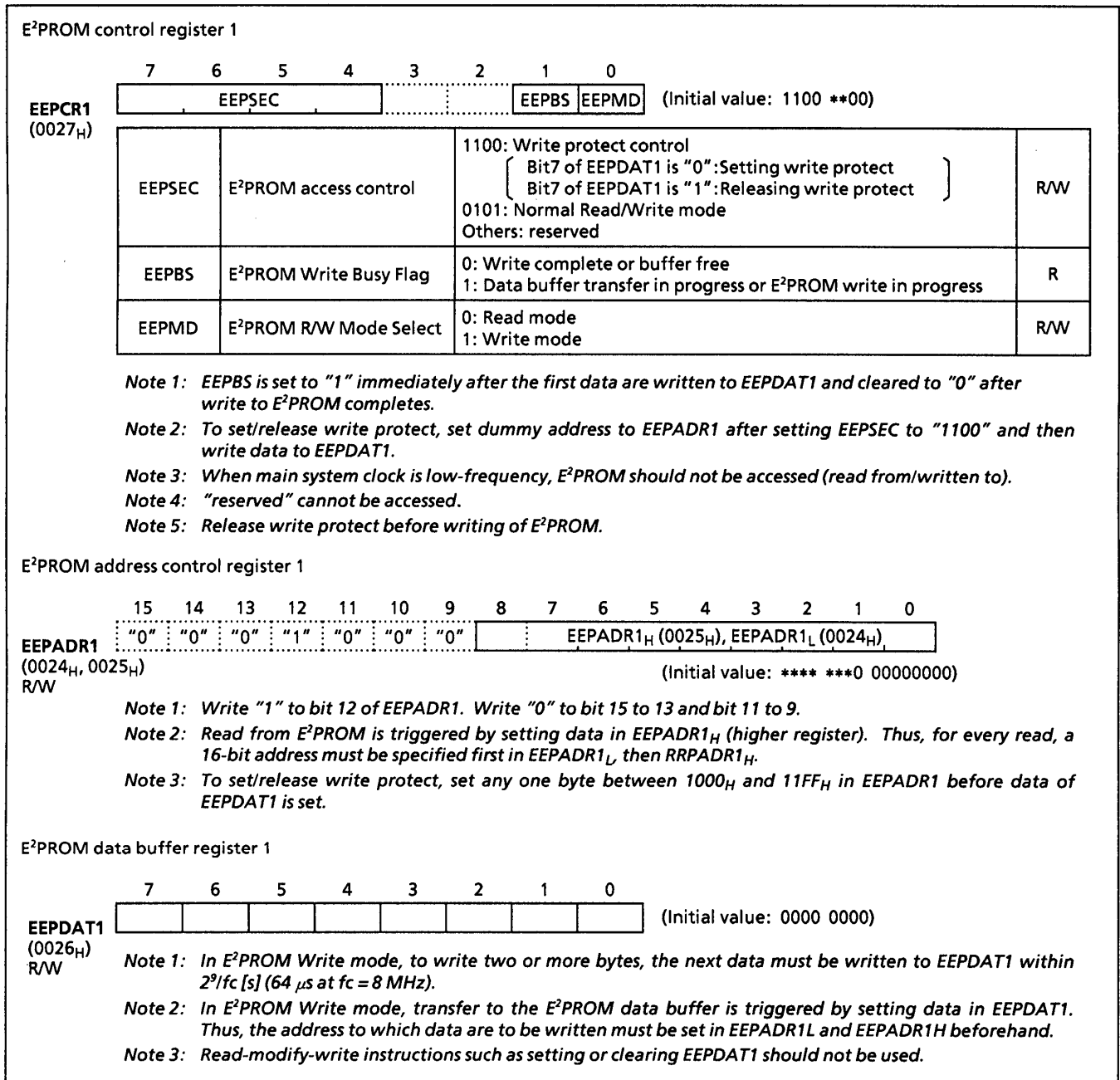*Note 3: Read-modify-write instructions such as setting or clearing EEPDAT1 should not be used.*

Figure 2-63.   E^{2}PROM control registers (Data memory)

2.12.3   Page configuration

The E^{2}PROM data memory area consists of 512 bytes (16 pages comprises 32 bytes in each page).

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1000h															PAGE 0																	
1020h															PAGE 1																	
1040h															PAGE 2																	
1060h															PAGE 3																	
1080h															PAGE 4																	
10A0h															PAGE 5																	
10C0h															PAGE 6																	
10E0h															PAGE 7																	
1100h															PAGE 8																	
1120h															PAGE 9																	
1140h															PAGE 10																	
1160h															PAGE 11																	
1180h															PAGE 12																	
11A0h															PAGE 13																	
11C0h															PAGE 14																	
11E0h															PAGE 15																	

Figure 2-64.   Data memory area of E^{2}PROM (512 byte)

2.12.4 Functions

Control of E²PROM data memory area
The E²PROM data memory area (1000$_H$ to 11FF$_H$) is accessed via the SFR (Special Function Register) while running a program in the MROM (32 KB) or E²PROM program area (16 KB).

(1) Read from data memory area
In reading from the data memory area, data are read from the SFR (EEPDAT1) through the E²PROM cell.
(a) Setting registers and reading data
To read data from the data memory area, set EEPCR1<EEPMD> to Read mode and set the address in EEPADR1L, then EEPADR1H. By reading EEPDAT1, the data at the specified address can be read.

> Note 1: *Because storing data in EEPDAT1 is triggered by writing to EEPADR1H, make sure to set the address first in EEPADR1L and then EEPADR1H.*
> Note 2: *To continue reading data, set an address in EEPADR1L, EEPADR1H in this order for each data to be read.*
> Note 3: *When EEPCR1<EEPBS> is "1", do not read data from EEPDAT1.*

Example: Read data from data memory area (1000$_H$ to 11FF$_H$) and write to RAM (0100$_H$ to 02FF$_H$)

```
        LD    (EEPCR1),01010000B   ; set to Read mode
        LD    HL,0100H             ; set destination RAM address
        LD    BC,1FFH              ; number of bytes to be transferred (512 bytes)
        LD    DE,1000H             ; set address
LOOP:
        LD    (EEPADR1L),DE        ; set address
        LD    A,(EEPDAT1)          ; read data
        LD    (HL+),A
        INC   DE
        DEC   BC
        JRS   F,LOOP
```

(2) Write to data memory area
In writing to the data memory area, data are written to the E²PROM cell from the SFR (EEPDAT1) via the data buffer. The data buffer consists of 32 bytes and the E²PROM cell consists of sixteen 32-byte pages. As data are transferred from the data buffer to the E²PROM cell one page at a time, up to 32 bytes can be written at a time.

(a) Setting registers
To write data to the data memory area, set EEPCR1<EEPMD> to Write mode, set an address in EEPADR1L, EEPADR1H, and set write data in EEPDAT1. To write data more than one, it is necessary to specify an address for each write data.

(b) Storing write data in data buffer
Setting data in EEPDAT1 automatically sets EEPCR1<EEPBS> to "1" and stores the data in the data buffer. As the data buffer consists of 32 bytes, up to 32 bytes can be stored at a time.

> *Note 1:* *The page in the E²PROM cell to which data are written is fixed with the address of the first byte stored in the data buffer. Therefore, the second and later bytes must be stored on the same page as the first byte. Do not write to the same address more than once and do not store data exceeding 32 bytes.*
>
> *Note 2:* *The data address to be stored in the data buffer is the value set in EEPADR1 when data are written to EEPDAT1. Therefore, set an address in EEPADR1 before writing data in EEPDAT1.*
>
> *Note 3:* *Before storing the first data in the data buffer, make sure that EEPCR1<EEPBS> is "0". (For details, see the example that follows.)*
>
> *Note 4:* *If continuous data store to the data buffer is interrupted by an interrupt of another function, data may not be written properly. Therefore, before storing data in the data buffer, disable interrupts by the IMF (interrupt master enable flag). However, pseudo non-maskable interrupts cannot be disabled. In the case of such an interrupt, when the interrupt service task reaches its end, return from this task after confirming that EEPCR1<EEPBS> is "0". (For details, see the example that follows.)*

(c) Write to E²PROM cell from data buffer
When 2^9/fc[s] (64 μs at fc = 8 MHz) elapses after data are stored in the data buffer, the data are automatically written to the E²PROM cell. To write two or more bytes of data, therefore, the time interval for writing to EEPDAT1 must be within 2^9/fc[s] (64 μs at fc = 8 MHz). The amount of data that can be written at a time is 1 to 32 bytes on the same page. Data are written to the E²PROM cell one page at a time. If data written to the data buffer are less than 32 bytes, only available data are written to the E²PROM cell.
A write to the E²PROM cell requires 2^{15}/fc[s](4.096 ms at fc = 8 MHz) after the last data are stored in the data buffer.
When the write completes, EEPCR1<EEPBS> is cleared to "0" and the data buffer is initialized and then an INTEEP interrupt is generated.

> *Note 1:* *To continuously write data to a different page, wait until an INTEEP interrupt is generated or EEPCR1<EEPBS> becomes "0".*
>
> *Note 2:* *Any write to EEPDAT1 after start of writing to the E²PROM buffer from the data buffer until EEPCR1<EEPBS> becomes "0" is invalid.*
>
> *Note 3:* *After writing the last data to EEPDAT1, do not overwrite the values of EEPCR1, EEPADRIL, EEPADRIH, and EEPDAT1 until the EEPCR1<EEPBS> becomes "0".*

Example: Write data "00" to data memory in units of 32 bytes

<E²PROM write task>

```
WTASK:
        TEST    (EEPCR1).1      ; monitor EEPCR1<EEPBS>
        JRS     T,WSTART        ; If EEPCR1<EEPBS> = "0" then WSTART
        RET                     ; return
WSTART:
        SET     (EEPCR1).0      ; set to Write mode
        LD      HL,1000H        ; set address
        DI                      ; IMF←"0"
LOOP1:
        LD      (EEPADR1L),HL   ; set address
        LD      (EEPDAT1),00H   ; set write data
        INC     HL
        LD      A,L
        AND     A,1FH
        CMP     A,00H
        JRS     F,LOOP1
LOOP2:
        TEST    (EEPCR1).1
        JRS     F,LOOP2
        CMP     HL,1200H
        JRS     F,LOOP1         ; If HL = 1200H then LOOP1
        EI                      ; IMF←"1"
        RET                     ; return
```

<Pseudo non-maskable interrupt task>

```
NMKTS:
        :    :
        (Interrupt task)
        :    :
LOOP3:
        TEST    (EEPCR1).1      ; monitor EEPCR1<EEPBS>
        JRS     F,LOOP3         ; If EEPCR1<EEPBS> = "1" then LOOP3
        RETN                    ; return
```

(3) Write protect function for data memory (Write disable protection)

Write disable (overwrite) protection is provided for the data memory. To use the write protection function, write data in the E²PROM cell specifically provided for this purpose. The write time is 2^{15}/fc[s] (4.096 ms at fc = 8 MHz) as in the case of writing to the normal E²PROM cell.

(a) Setting registers and writing data

To set or release write protection, write "1100" in EEPCR1<EEPSEC> and set EEPCR1<EEPMD> to "1". Then, set a dummy address (any one byte between 1000H and 11FFH) in EEPADR1 and set data in bit 7 of EEPDAT1. Writing data in EEPDAT1 transfers the data to the data buffer. When 2^9/fc[s] (64 μs at fc = 8 MHz) elapses, a write to the write protect bit automatically starts. Writing "0" in bit 7 of EEPDAT1 sets write protection (write disabled) while writing "1" releases write protection (write enabled). When the write completes, the EEPCR1<EEPBS> flag is cleared to "0" and an INTEEP interrupt is generated.

After the write has completed, set EEPCR1<EEPSEC> to "0101" to set Normal Read/Write mode.

When write protection is released, data can be read from or written to the data memory. When write protection is set, data can only be read from the data memory. Even if a write is attempted in this state, no data are written.

> Note 1: At shipment, write protection is set.
> Note 2: The write protect bit shares the same structure as the E²PROM cell in the data memory. Attention should be paid so that the write protect bit is not excessively overloaded compared with the data memory.

Example 1: Release write protect

```
        LD   (EEPCR1),0C1H      ; Write Protect mode
        LD   (EEPADRL),00H      ; set dummy address in lower register
        LD   (EEPADRH),10H      ; set dummy address in higher register
        LD   (EEPDAT1),80H      ; release write protection
LOOP:
        TEST (EEPCR1).1         ; write completion check
        JRS  F,LOOP
```

Example 2: Set write protect

```
        LD   (EEPCR1),0C1H      ; Write Protect mode
        LD   (EEPADRL),00H      ; set dummy address in lower register
        LD   (EEPADRH),10H      ; set dummy address in higher register
        LD   (EEPDAT1),00H      ; release write protection
LOOP:
        TEST (EEPCR1).1         ; write completion check
        JRS  F,LOOP
```

2.12.5 Notes on using BM87EP26F0A (development tool)

The data memory area of the BM87EP26F0A uses emulation RAM instead of E^2PROM for reliability purposes. Therefore, the BM87EP26F0A and the TMP87EP26 are not exactly equivalent in operation and the following points should be taken into account when using the BM87EP26F0A.

Note 1: If the power to the BM87EP26F0A is cut off, the contents of the data memory area (1000_H to $11FF_H$) and the write protect bit become undefined. Make sure to perform initial setting after the power is turned on.

Example: Initial setting for write protection at power-up

```
WPINIT:
        LD     (EEPCR1),11000000B      ; set to Read mode
        LD     (EEPADR1L),00000000B    ; set dummy address (lower)
        LD     (EEPADR1H),00010000B    ; set dummy address (higher)
        TEST   (EEPDAT1).7             ; write protection check (0: set,1: release)
        JRS    F,SEC_END
        LD     (EEPCR1),11000010B      ; set to Write mode
        LD     (EEPADR1L),00000000B    ; set dummy address (lower)
        LD     (EEPADR1H),00010000B    ; set dummy address (higher)
        LD     (EEPCR1),10000000B      ; release write protection
SEC_REL:
        TEST   (EEPCR1).1              ; write completion check
        JRS    F,SEC_REL
SEC_END:
              :    :
```

Note 2: Data cannot be written to E^2PROM when using the BM87EP26F0A.

Note 3: The E^2PROM write time (RAM emulation time) of the BM87EP26F0A differs from that of the TMP87EP26.

BM87EP26F0A write time	: 2^{15}/fc[s] (TBTCR<DV7CK> = "0")
	: 2^7/fs [s] (TBTCR<DV7CK> = "1")
TMP87EP26 write time	: 2^{15}/fc[s] (Always the same regardless of TBTCR<DV7CK>)

Note 4: In the BM87EP26F0A the data buffer is not used in writing data to the data memory and a write to emulation RAM is performed immediately after data is written in EEPDAT1. Therefore, in writing 2 or more bytes of data on the same page at a time, if the EEPDAT1 write interval exceeds 2^9/fc[s] (64 μs at fc = 8 MHz), the EEPDAT1 data that is set after a period of 2^9/fc[s] is valid in the BM87EP26F0A while it is invalid in the TMP87EP26.
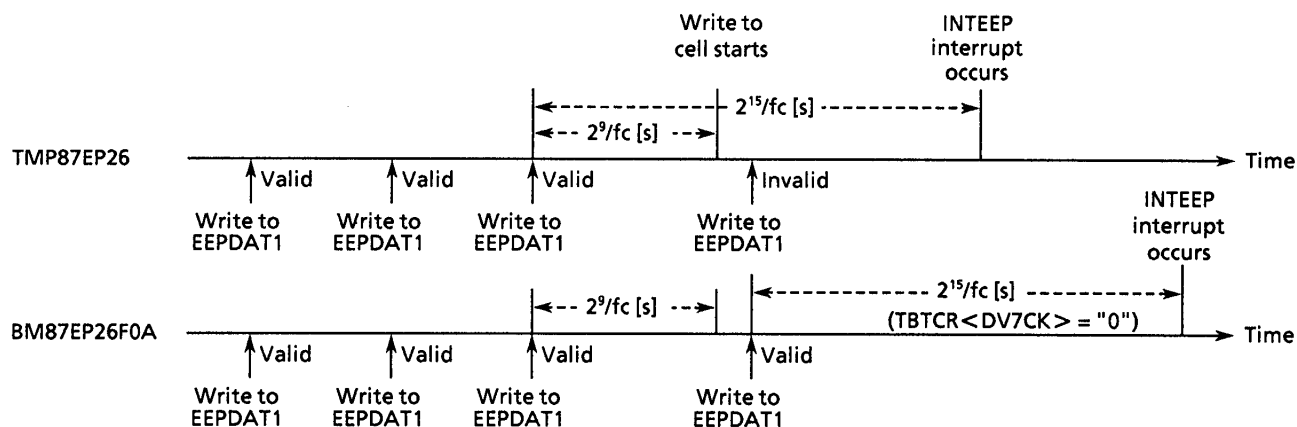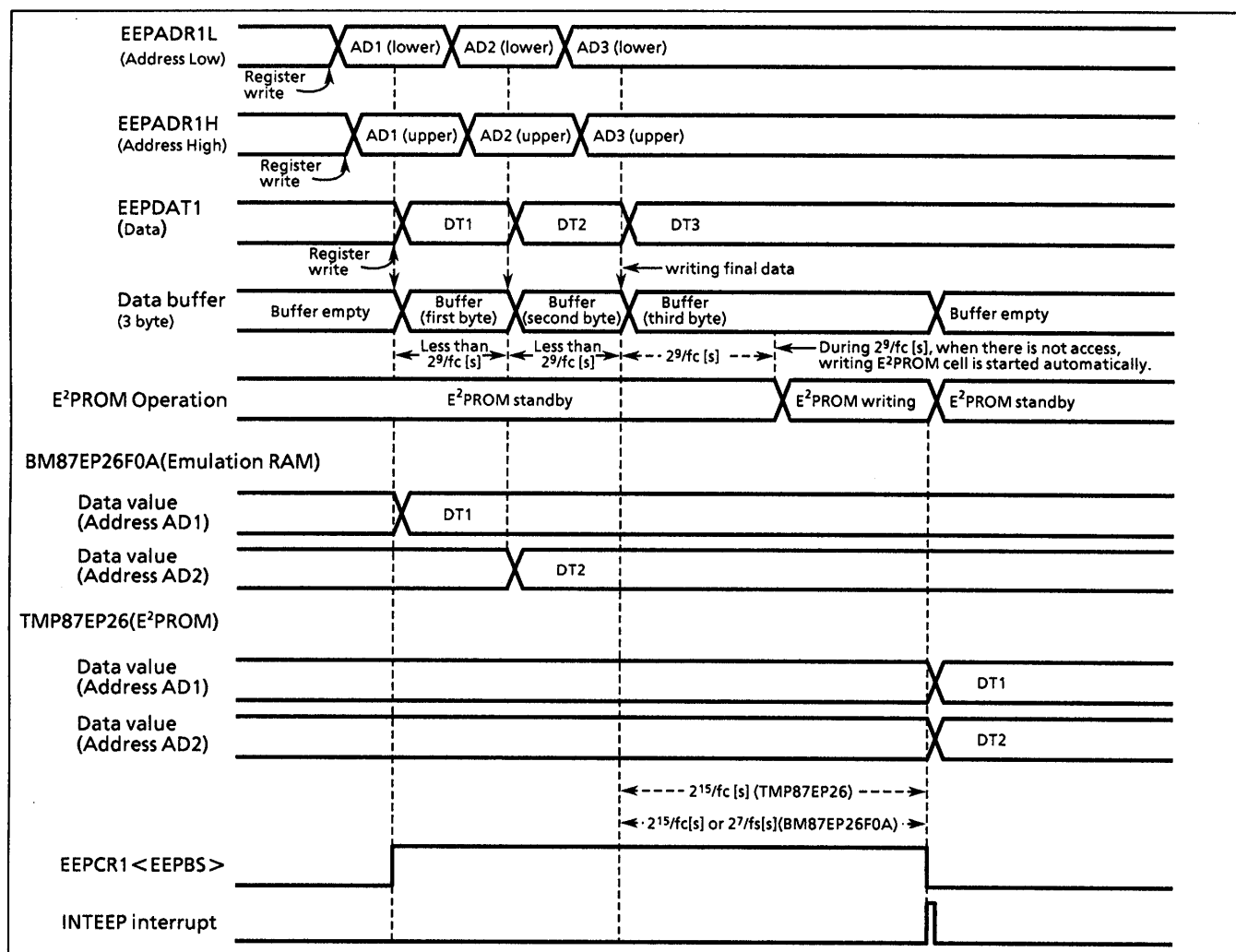

Figure 2-65. Write timing of BM87EP26F0A and TMP87EP26 (In case of 3 byte transfer)

2.13   E²PROM (Program memory area)
The TMP87EP26 incorporates E²PROM program memory of 16 KB (4000$_H$ to 7FFF$_H$).
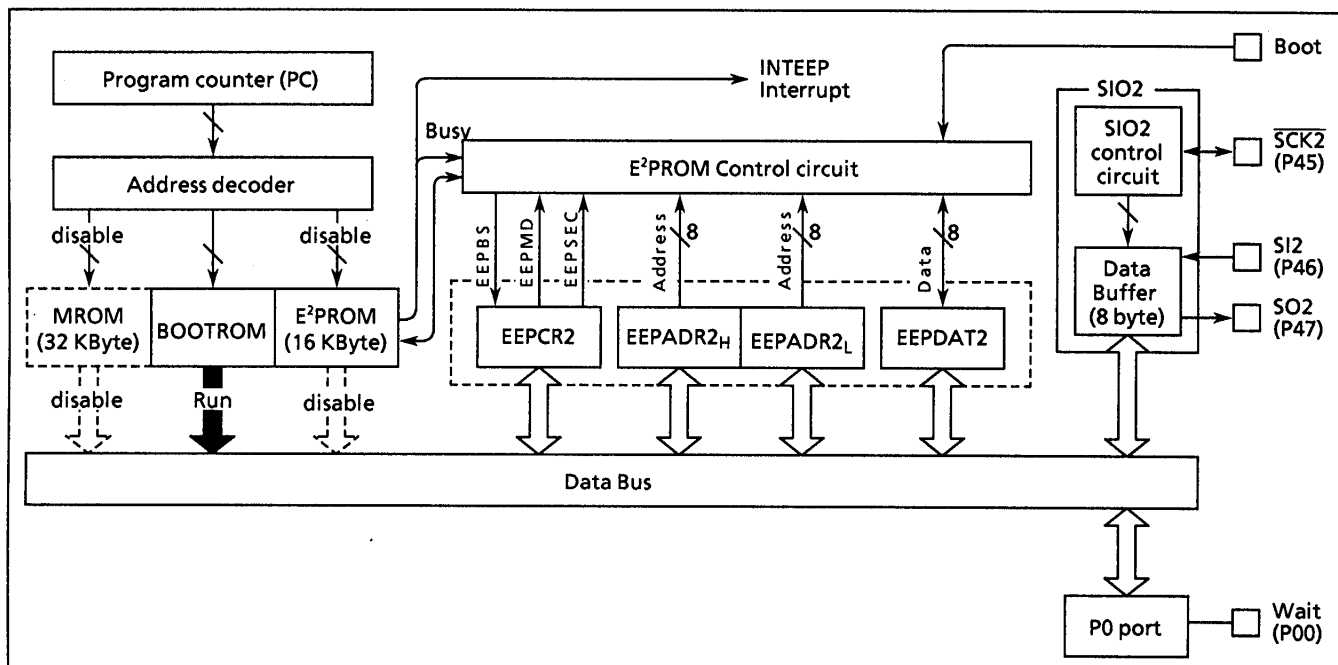
2.13.1   Configuration

Figure 2-66.   Program memory area of E²PROM (Serial PROM mode)

To write data to the program memory area (4000$_H$ to 7FFF$_H$), activate Serial PROM mode by releasing RESET while inputting "1" to the BOOT pin. When Serial PROM mode is activated, the built-in BOOT-ROM is mapped onto the ROM area and the E²PROM write program is executed. Data are transferred via the serial interface and are written using commands. The serial interface is controlled by an external clock and the maximum transfer speed is 244 Kbps (at fc = 8 MHz). Writing 1 page (32 bytes) in the program memory area requires 2^{15}/fc[s] (4.096 ms at fc = 8 MHz) as in the case of the data memory. The program memory consists of 512 pages (1 page = 32 bytes), so the time required for writing 16 KB is as follows:

> EPROM write time   :4.096[ms](write time per page) × 512(pages) = 2.048[s]
> Serial transfer time   :37(command + address + data) × 8(bits) × 512(pages) × (1/244K) = 0.606[s]
> (VDD = 5V, fc = 8MHz, excluding instruction execution time and external interface wait time)

> *Note 1: Data can be written to the program memory area only in Serial PROM mode.*
> *Note 2: In Serial PROM mode, data cannot be read from the program memory area.*
> *Note 3: In Serial PROM mode, the data memory area cannot be accessed.*

2.13.2 Memory Composition

The E²PROM program memory area comprises 16 KB (512 pages each comprising 32 bytes).

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4000h															PAGE 0																	
4020h															PAGE 1																	
4040h															PAGE 2																	
4060h															PAGE 3																	
4080h															PAGE 4																	
40A0h															PAGE 5																	
40C0h															PAGE 6																	
40E0h															PAGE 7																	
4100h															PAGE 8																	
4120h															PAGE 9																	
4140h															PAGE 10																	
4160h															PAGE 11																	
: :															: :																	
7FA0h															PAGE 509																	
7FC0h															PAGE 510																	
7FE0h															PAGE 511																	

図2-67. E²PROM Program Memory Area (16 Kbytes)

2.13.3 Notes on using BM87EP26F0A (development tool)

Take note of the following points for emulating the E²PROM program area.

Note 1: *Develop software on the BM87EP26F0A in the same way as for 48 Kbytes mask ROM products.*
Note 2: *Data cannot be written to E²PROM when using the BM87EP26F0A.*
Note 3: *If the reference destination of the reset vector is specified directly in the E²PROM area, mask ROM data may be read by a third party overwriting E²PROM. It is recommended that security measures be taken such as first decoding a specified code written in E²PROM within the mask ROM area and then starting the program instead of setting the reset vector to directly refer to E²PROM.*

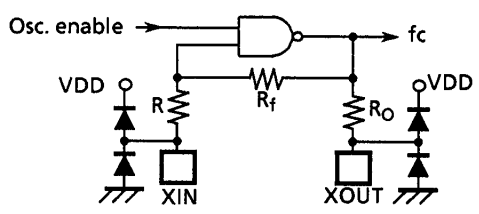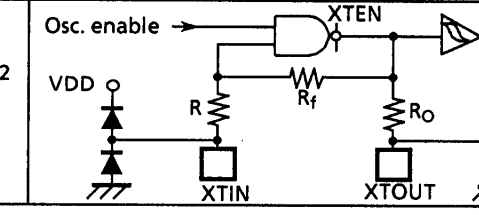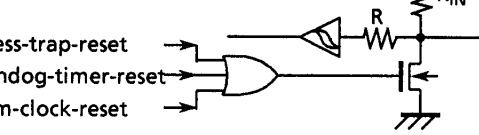## 2.14   Product comparison list
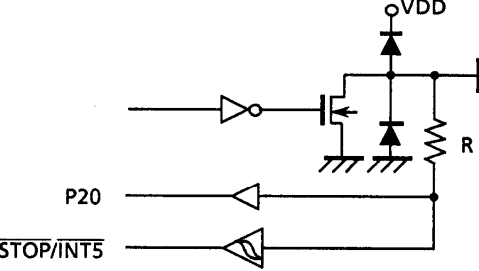
Table 2-13. Product comparison list
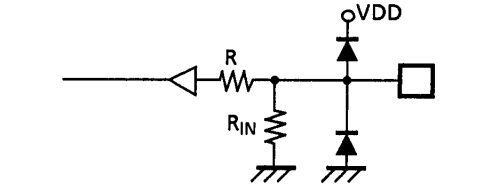
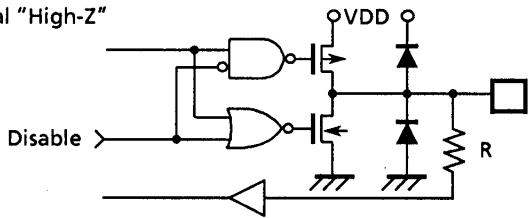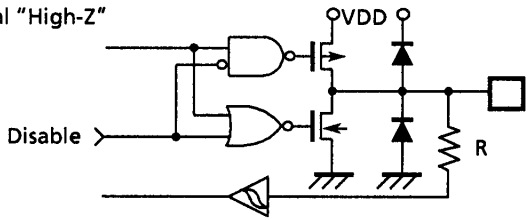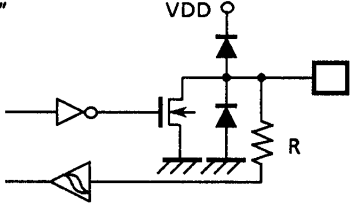| Product	TMP87CP24A	TMP87CP23	TMP87EP26
ROM	48 KB (MROM)		32 KB (MROM) + 16 KB (E²PROM) + 512 B (E²PROM)
RAM	2 kB		
I/O ports	69 ports	70 ports	73 ports
Package	P-QFP100-1414-0.50C	P-QFP100-1420-0.65A	
Supply Voltage	2.7 V to 5.5 V at 4.2 MHz / 32.768 kHz 4.5 V to 5.5 V at 8 MHz / 32.768 kHz		4.5 V to 5.5 V at 8 MHz / 32.768 kHz
AD	8 ch (Total error ± 2 LSB)		8 ch (Total error ± 4 LSB)
Timer/Counter	16 bit Timer / Counter × 2 ch 8 bit Timer / Counter × 2 ch		
SIO	2 ch (8 Byte buffer × 2)		
LCD	Voltage booster	Bleeder resistor	Bleeder resistor (Selectable LCD drive ability)
Electrical characteristics	–		In all I/O ports, absolute maximum rating of input voltage is " – 0.3 to VDD to 0.3" volts.

Input / Output Circuitry
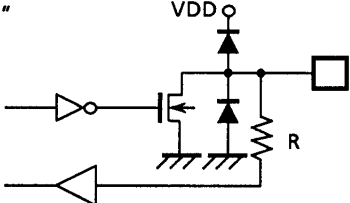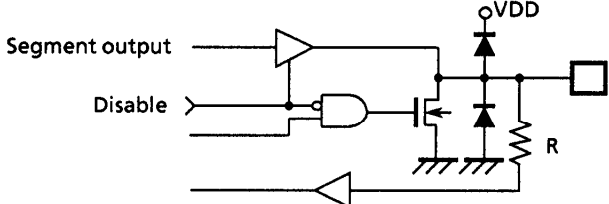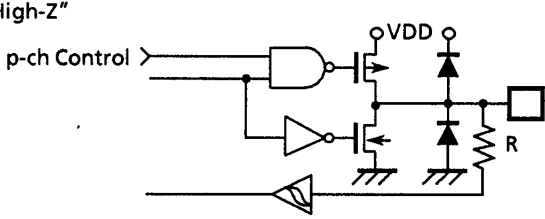
(1) Control pins
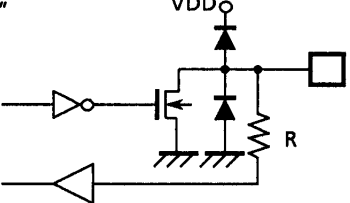
The input / output circuitries of the TMP87EP26 control pins are shown below.
Please specify either the single-clock mode or the dual-clock mode by a code (NM1 or NM2) as an option for an operating mode during reset.

Control Pin	I/O	Input / Output Circuitry and Code	Remarks
XIN XOUT	Input Output		Resonator connecting pins (high-frequency) $R_f = 1.2\ M\Omega$ (typ.) $R_O = 1.5\ k\Omega$ (typ.) $R = 1\ k\Omega$ (typ.)
XTIN XTOUT	Input Output	NM1: Refer to port P2 NM2:	Resonator connecting pins (low-frequency) $R_f = 6\ M\Omega$ (typ.) $R_O = 220\ k\Omega$ (typ.) $R = 1\ k\Omega$ (typ.)
$\overline{RESET}$	I/O		Sink open drain output Hysteresis input Pull-up resistor $R_{IN} = 220\ k\Omega$ (typ.) $R = 1\ k\Omega$ (typ.)
$\overline{STOP}$/INT5	Input		Hysteresis input $R = 1\ k\Omega$ (typ.)
TEST BOOT	Input		Pull-down resistor $R_{IN} = 70\ k\Omega$ (typ.) $R = 1\ k\Omega$ (typ.)

(2)　Input/Output Ports
　　　The input/output circuitries of the TMP87EP26 input/output ports are shown below.

Port	I/O	Input / Output Circuitry and Code	Remarks
P0 P5	I/O	Initial "High-Z" Disable	Tri-state I/O R = 1 kΩ (typ.)
P1	I/O	Initial "High-Z" Disable	Tri-state I/O Hysteresis input R = 1 kΩ (typ.)
P3	I/O	Initial "High-Z"	Sink open drain output Hysteresis input R = 1 kΩ (typ.)
P2	I/O	Initial "High-Z"	Sink open drain output R = 1 kΩ (typ.)
P6 P7 P8 P9	I/O	Segment output Disable	Sink open drain or Segment output R = 1 kΩ (typ.)
P4	I/O	Initial "High-Z" p-ch Control	Sink open drain or push-pull output P41 Nch High current output Hysteresis input
PD	I/O	Initial "High-Z"	Sink open drain output R = 1 kΩ (typ.)

Electrical Characteristics

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

Parameter	Symbol	Pins	Ratings	Unit
Supply Voltage	V_{DD}		– 0.3 to 6.5	V
Input Voltage	V_{IN}		– 0.3 to V_{DD} + 0.3	V
Output Voltage	V_{OUT}		– 0.3 to V_{DD} + 0.3	V
Output Current (Per 1 pin)	I_{OUT1}	All I/O ports except P41 port	3.2	mA
	I_{OUT2}	P41	30	
Output Current (Total)	$\Sigma\,I_{OUT1}$	All I/O ports except P41 port	120	mA
	$\Sigma\,I_{OUT2}$	P41	30	
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

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

Recommended 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 mode	4.5	5.5	V
				IDLE1, 2 mode			
			fs = 32.768 kHz	SLOW mode			
				SLEEP mode			
				STOP mode			
Input High Voltage	V_{IH1}	Except hysteresis input			$V_{DD} \times 0.70$	V_{DD}	V
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
Input Low Voltage	V_{IL1}	Except hysteresis input			0	$V_{DD} \times 0.30$	V
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
Clock Frequency	fc	XIN, XOUT			4.0	8.0	MHz
	fs	XTIN, XTOUT			30.0	34.0	kHz

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

Note 2: Clock frequency fc: Supply voltage range is specified in NORMAL1/2 mode and IDLE1/2 mode.

| DC Characteristics | | | $(V_{SS} = 0\ V,\ Topr = -30\ to\ 70°C)$ |

Parameter	Symbol	Pins	Conditions	Min	Typ.	Max	Unit
Hysteresis Voltage	V_{HS}	Hysteresis inputs		–	0.9	–	V
Input Current	I_{IN1}	TEST	$V_{DD} = 5.5\ V$, $V_{IN} = 5.5\ V\,/\,0\ V$	–	–	±2	μA
	I_{IN2}	Open drain ports and tri-state ports					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
Input Low Current	I_{IL}	Push-pull ports	$V_{DD} = 5.5\ V$, $V_{IN} = 0.4\ V$	–	–	– 2	mA
Input Resistance	R_{IN2}	$\overline{RESET}$		100	220	450	$k\Omega$
Output Leakage Current	I_{LO1}	Open drain ports	$V_{DD} = 5.5\ V$, $V_{OUT} = 5.5\ V$	–	–	2	μA
	I_{LO2}	Tri-state ports	$V_{OUT} = 5.5\ V\,/\,0\ V$	–	–	±2	
Segment Output Low Resistance	R_{OS1}	SEG39 to SEG0		–	20	–	$k\Omega$
Common Output Low Resistance	R_{OC1}	COM3 to COM0					
Segment Output High Resistance	R_{OS2}	SEG39 to SEG0	$V_{DD} = 5\ V$, $V_{DD} - V_{LC} = 3\ V$	–	200	–	$k\Omega$
Common Output High Resistance	R_{OC2}	COM3 to COM0					
Segment/Common Output Voltage	$V_{O\,2/3}$	SEG39 to SEG0 and COM3 to COM0		3.8	4.0	–	V
	$V_{O\,1/2}$			3.3	3.5	–	
	$V_{O\,1/3}$			2.8	3.0	–	
Output High Voltage	V_{OH1}	Push-pull ports P4 ports	$V_{DD} = 4.5\ V$, $I_{OH} = -200\ \mu A$	2.4	–	–	V
	V_{OH2}	Tri-state ports P0, P1, P5 ports	$V_{DD} = 4.5\ V$, $I_{OH} = -0.7\ mA$	4.1	–	–	
Output Low Voltage	V_{OL}	Except XOUT and P41	$V_{DD} = 4.5\ V$, $I_{OL} = 1.6\ mA$	–	–	0.4	V
Output Low Current	I_{OL3}	P41	$V_{DD} = 4.5\ V$, $V_{OL} = 1.0\ V$	–	20	–	mA

Supply Current in NORMAL 1, 2 mode — I_{DD} — $V_{DD} = 5.5\ V$, $fc = 8\ MHz$, $fs = 32.768\ kHz$, $V_{IN} = 5.3\ V\,/\,0.2\ V$

	Data area	Prog. area	Min	Typ.	Max	Unit
EEPROM operate	–		–	7	12	mA
	RD	–	–	9	15	
	WR		–	8	13.5	
	–	RD	–	9	15	
	RD		–	11	18	
	WR		–	10	16.5	
Serial PROM mode			–	9	15	

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Supply Current in IDLE 1, 2 mode	I_{DD}		–	4.5	7.5	mA
Supply Current in SLOW mode		$VDD = 5.5\ V$, $fs = 32.768\ kHz$, $VIN = 5.3\ V\,/\,0.2\ V$, LCD driver is not enable	–	55	110	μA
Supply Current in SLEEP mode			–	27.5	55	μA
Supply Current in STOP mode		$V_{DD} = 5.5\ V$, $V_{IN} = 5.3\ V\,/\,0.2\ V$	–	0.5	10	μA

Note 1: Typical values show those at Topr = 25°C, $V_{DD} = 5\ V$.
Note 2: Input Current ; The current through pull-up or pull-down resistor is not included.
Note 3: I_{DD} ; Except for I_{REF}
Note 4: Output resistors Ros, Roc indicate "on" when switching levels.
Note 5: $V_{O2/3}$ indicates an output voltage at the 2/3 level when operating in the 1/4 or 1/3 duty mode.
Note 6: $V_{O1/2}$ indicates an output voltage at the 1/2 level when operating in the 1/2 duty or static mode.
Note 7: $V_{O1/3}$ indicates an output voltage at the 1/3 level when operating in the 1/4 or 1/3 duty mode.
Note 8: When using LCD, it is necessary to consider values of Ros1/2 and Rbc1/2.
Note 9: Times for SEG/COM output switching on: Ros1, Roc1: $2^6/fc$, $2/fc$ (s)
*   Ros2, Roc2: 1/(n, f_F)*
*   (1/n: duty, f_F: frame frequency)*

AD Conversion Characteristics   (V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = − 30 to 70°C)

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Analog Reference Voltage	V_{AREF}		4.5	−	V_{DD}	V
Analog Input Voltage	V_{AIN}		0	−	V_{AREF}	V
Analog Supply Current	I_{REF}	V_{AREF} = 5.5 V	−	0.5	1.0	mA
Nonlinearity Error			−	−	± 2	LSB
Zero Point Error		V_{DD} = 5.0 V, V_{SS} = 0.0 V V_{AREF} = 5.000 V	−	−	± 2	
Full Scale Error			−	−	± 2	
Total Error			−	−	± 4	

Note: Quantizing error is not contained in those errors.

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

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Machine Cycle Time	t_{cy}	In NORMAL 1, 2 mode	0.5	−	1	μs
		In IDLE 1, 2 mode				
		In SLOW mode	117.6	−	133.3	
		In SLEEP mode				
High Level Clock Pulse Width	t_{WCH}	For external clock operation (XIN input), fc = 8 MHz	62.5	−	−	ns
Low Level Clock Pulse Width	t_{WCL}					
High Level Clock Pulse Width	t_{WSH}	For external clock operation (XTIN input), fs = 32.768 kHz	14.7	−	−	μs
Low Level Clock Pulse Width	t_{WSL}					

E²PROM Characteristics				
Parameter	Conditions	Ta = 25°C	Ta = 70°C	Unit
Number of Guaranteed Writes to E²PROM Data Memory	VDD = 5 V, fc = 8 MHz	10^5	2.5×10^4	times
Number of Guaranteed Writes to E²PROM Write Protect bit				
Number of Guaranteed Writes to E²PROM Program Memory		10	10	times

Note 1:   Number of Guaranteed Writes indicates that at least Writes up to that number of times will be performed properly and does not indicate that further Writes will necessarily fail.

Note 2:   Number of Writes is counted on a per-cell basis; The number does not refer to the number of Writes on E²PROM as a whole.

Recomended Oscillating Condition

The product numbers and specifications of the resonatorsby Murata Manufacturing Co., Ltd. are subject to change.
For up-to-date information, please refer to the following URL;
http://www.murata.co.jp/search/index.html

(1)   High-frequency   (2)   Low-frequency
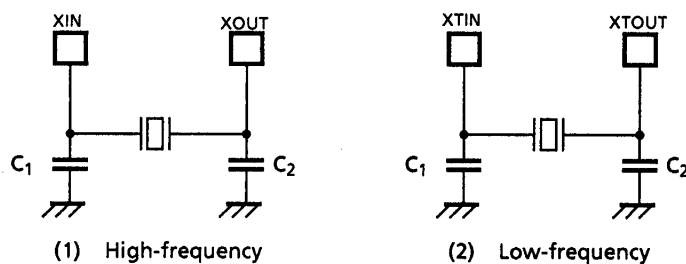
Note: When used in high electric field such as a picture tube, the package is recommended to be electrically shielded to maintain a regular operation.