CMOS 8-Bit Microcontroller

TMP87CH29U/N, TMP87CK29U/N, TMP87CM29U/N

The TMP87CH29/K29/M29 are high-speed and high-performance 8-bit single chip microcomputers. These MCU contains CPU core, ROM, RAM, a LCD driver, multi-function timer/counters, an AD converter, two clock generators and a serial interface (UART) on a chip.

Part No.	ROM	RAM	Package	OTP MCU
TMP87CH29U	16 K × 8-bit	1 K × 8-bit	P-LQFP64-1010-0.50D	TMP87PM29U
TMP87CH29N			P-SDIP64-750-1.78	TMP87PM29N
TMP87CK29U	24 K × 8-bit		P-LQFP64-1010-0.50D	TMP87PM29U
TMP87CK29N			P-SDIP64-750-1.78	TMP87PM29N
TMP87CM29U	32 K × 8-bit		P-LQFP64-1010-0.50D	TMP87PM29U
TMP87CM29N			P-SDIP64-750-1.78	TMP87PM29N

Features

◆8-bit single chip microcomputer TLCS-870 Series

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

◆412 basic instructions

- Multiplication and Division (8 bits × 8 bits, 16 bits ÷ 8 bits)
- Bit manipulations (set/clear/complement/move/test /exclusive or)
- 16-bit data operations
- 1-byte jump/subroutine-call (Short relative jump/Vector call)

◆13 interrupt sources (External: 4, Internal: 9)

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

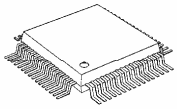
◆7 Input/Output ports (43 pins)

- High current output: 3 pins (typ. 20 mA)

◆18-bit Timer/Counter

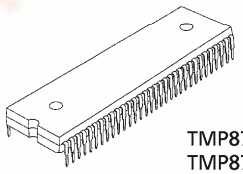
- Timer, Event counter, Pulse width measurement, Frequency measurement modes

P-LQFP64-1010-0.50D



TMP87CH29U
TMP87CK29U
TMP87CM29U
TMP87PM29U

P-SDIP64-750-1.78



TMP87CH29N
TMP87CK29N
TMP87CM29N
TMP87PM29N

000707EBP1

- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance / Handling Precautions.
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc..
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk.
- The products described in this document are subject to the foreign exchange and foreign trade laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

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

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

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

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

◆Universal asynchronous receiver and transmitter (UART)
- With 8-bit transmit/receive data buffer
- Transfer clock, Select of with/without parity bit.

◆LCD driver/Controller
- LCD direct drive capability (max. 12-digit display at 1/4 duty LCD).
- 1/4, 1/3, 1/2 duties or static drive are programmably selectable.
- With display memory.

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

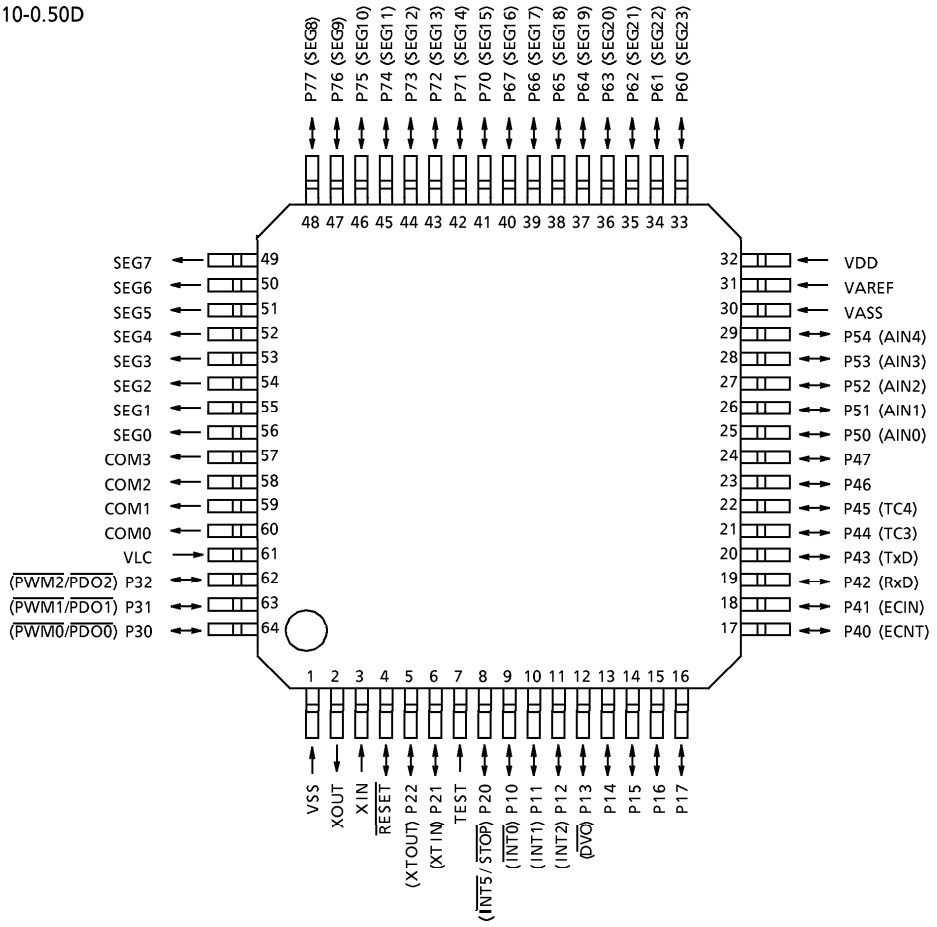
◆Five power saving operating modes
- STOP mode: Oscillation stops. Battery/Capacitor back-up. Port output hold/High-impedance.
- SLOW mode: Low power consumption operation using low-frequency clock (32.768 kHz).
- IDLE1 mode: CPU stops, and peripherals operate using high-frequency clock. Release by interrupts.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

◆Wide operating voltage: 2.7 to 5.5V at 4.19 MHz/32.768 kHz, 4.5 to 5.5 V at 8 MHz/32.768 kHz

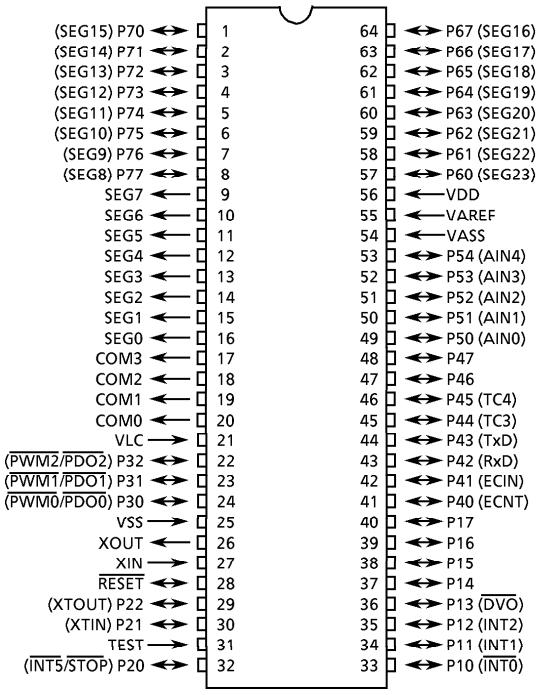
◆Emulation Pod: BM87CM29U0A

Pin Assignments (Top View)

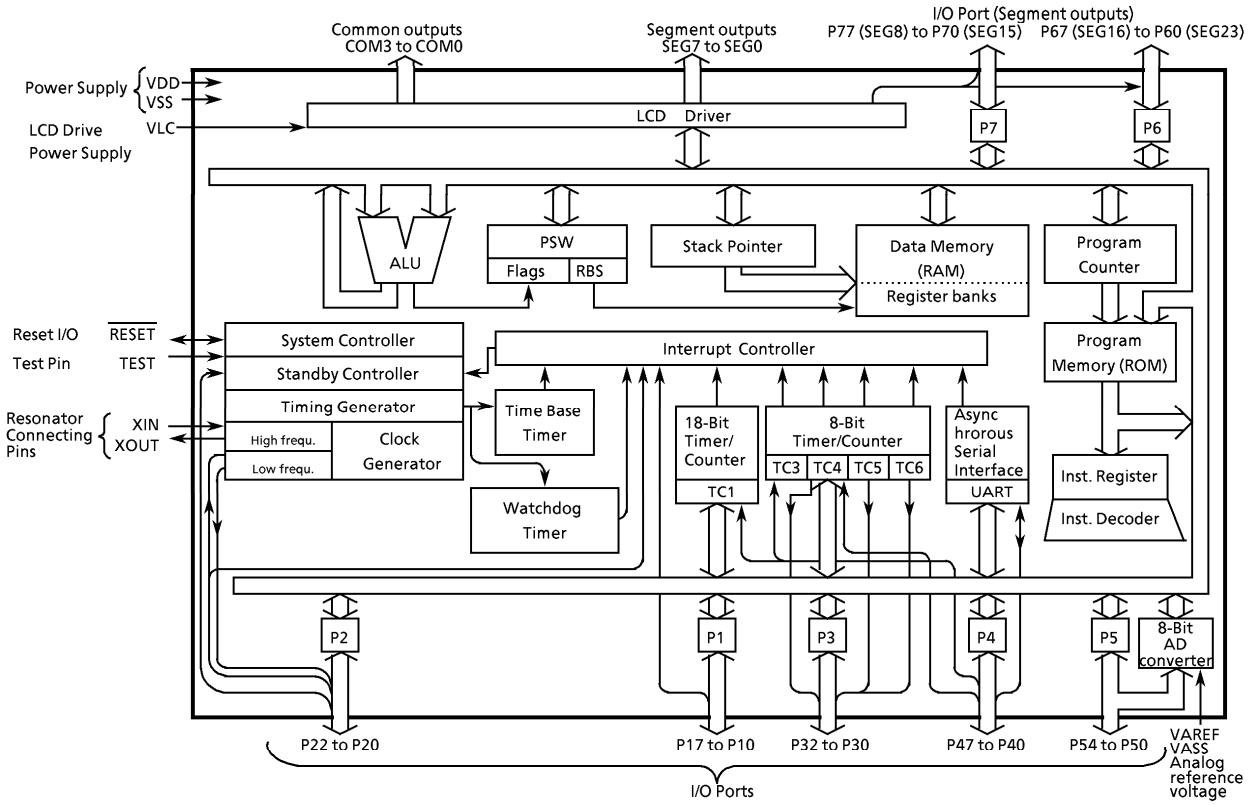
P-LQFP64-1010-0.50D



P-SDIP64-750-1.78



Block Diagram



Pin Function

Pin Name	Input/Output	Function	
P17 to P14	I/O	8-bit programmable input/output ports (tri-state) . Each bit of these ports can be individually configured as an input or an output under software control. During reset, all bits are configured as inputs. When used as a divider output, the latch must be set to "1".	
P13 ($\overline{DVO}$)	I/O (Output)		Divider output
P12 (INT2)	I/O (Input)		External interrupt input 2
P11 (INT1)			External interrupt input 1
P10 ($\overline{INT0}$)			External interrupt input 0
P22 (XTOUT)	I/O (Output)	3-bit input/output port with latch. When used as an input port, the latch must be set to "1".	Resonator connecting pins (32.768 kHz). For inputting external clock, XTIN is used and XTOUT is opened.
P21 (XTIN)	I/O (Input)		
P20 ($\overline{INT5}$/$\overline{STOP}$)			External interrupt input 5 or STOP mode release signal input
P32 ($\overline{PWM2}$/$\overline{PDO2}$)	I/O (Output)	3-bit input/output port (high current output) with latch. When used as an input port, a PWM output, or a PDO output, the latch must be set to "1".	8-bit $\overline{PWM2}$ output or 8-bit $\overline{PDO2}$ output
P31 ($\overline{PWM1}$/$\overline{PDO1}$)			8-bit $\overline{PWM1}$ output or 8-bit $\overline{PDO1}$ output
P30 ($\overline{PWM0}$/$\overline{PDO0}$)			8-bit $\overline{PWM0}$ output or 8-bit $\overline{PDO0}$ output
P47	I/O	8-bit input/output port with latch. Each bit of these ports can be individually configured as a sink open drain or a push-pull output under software control. During reset, all bits are configured as sink open drain outputs. When used as an input port, a timer/counter input, a PWM output, a PDO output, or a UART input/output, the latch must be set to "1".	
P46			
P45 (TC4)	I/O (Input)		Timer/Counter 4 input
P44 (TC3)			Timer/Counter 3 input
P43 (TxD)	I/O (Output)		UART data output
P42 (RxD)	I/O (Input)		UART data input
P41 (ECIN)			Timer/Counter 1 inputs
P40 (ECNT)			
P54 (AIN4)	I/O (Input)	5-bit programmble input/output ports (tri-state) . Each bit of these ports can be individually configured as an input or an output under software control.	AD converter analog inputs
P53 (AIN3)			
P52 (AIN2)			
P51 (AIN1)			
P50 (AIN0)			
P67 (SEG16) to P60 (SEG23)	I/O (Output)	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 a segment output, the P6 control register (P6CR) must be set to "1".
P77 (SEG8) to P70 (SEG15)	I/O (Output)	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 a segment output, the P7 control register (P7CR) must be set to "1".
SEG7 to SEG0	Output	LCD Segment outputs	
COM3 to COM0		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.	
$\overline{RESET}$	I/O	Reset signal input or watchdog timer output/address-trap-reset output/system-clock-reset output.	
TEST	Input	Test pin for out-going test. Be tied to low.	
VDD, VSS	Power Supply	+ 5 V, 0 V (GND)	
VAREF, VASS		Analog reference voltage inputs (High, Low)	
VLC		LCD drive power supply	

Operational Description

1. CPU Core Functions

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

1.1 Memory Address Map

The TLCS-870 Series is capable of addressing 64K bytes of memory. Figure 1-1 shows the memory address maps of the TMP87CH29/K29/M29.

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

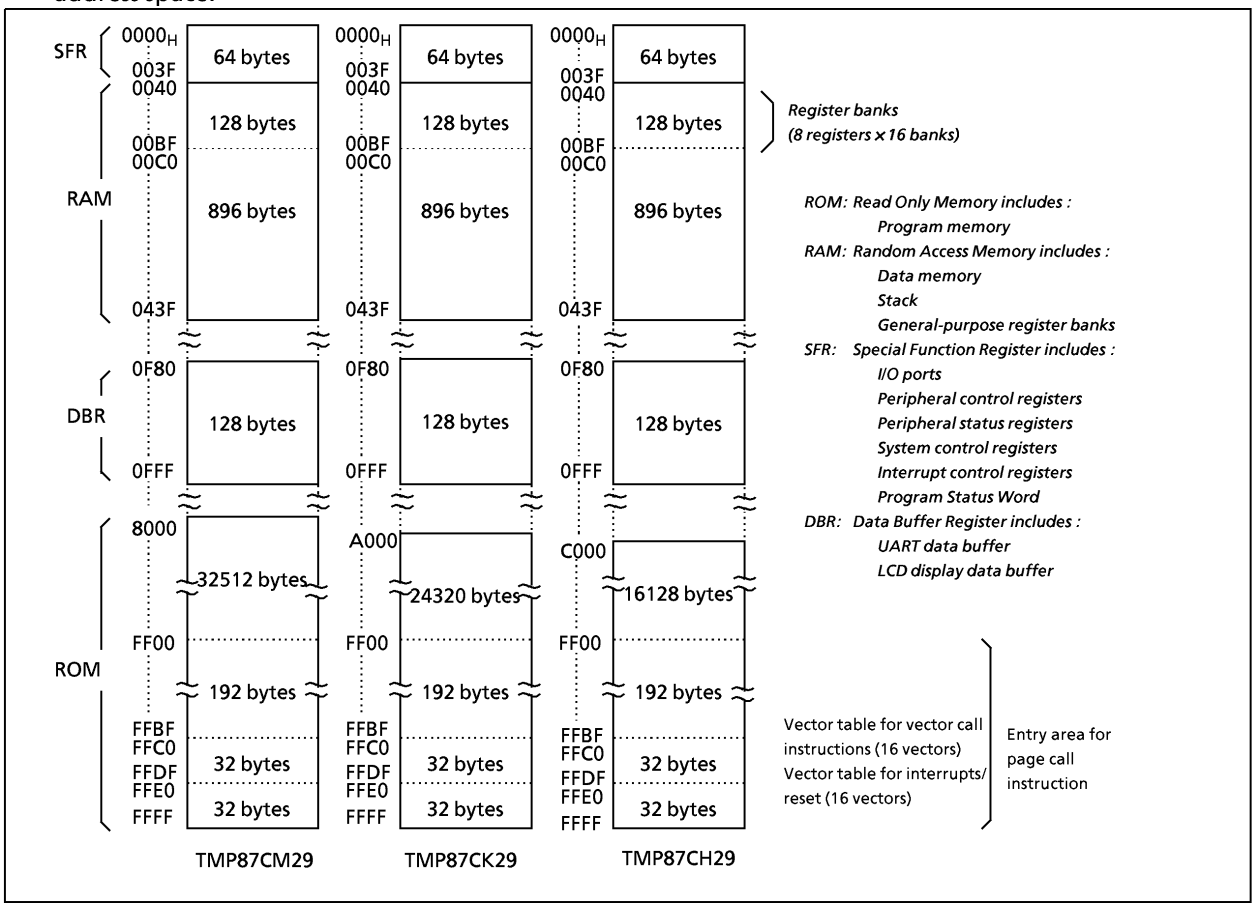


Figure 1-1. Memory Address Maps

1.2 Program Memory (ROM)

The TMP87CH29 has a 16K × 8-bit (addresses $C000_H$ to $FFFF_H$), the TMP87CK29 has a 24K × 8-bit (addresses $A000_H$ to $FFFF_H$), and the TMP87CM29 has a 32K × 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 addresses $FFC0_H$ 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 current contents of the PC. (The PC contains the address of the instruction being executed + 2; therefore, in this case, the PC contents are $E8C4_H + 2 = E8C6_H$.)

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

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

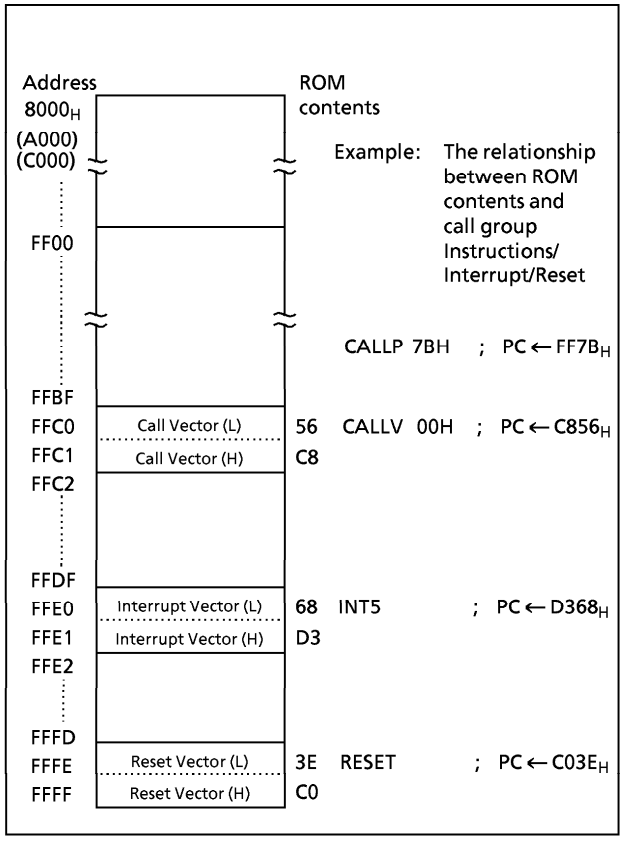


Figure 1-2. Program Memory Map

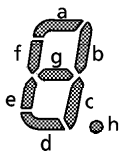
In the TLCS-870 Series, the same instruction used to access the data memory (e.g. [LD A, (HL)]) is also used to read out fixed data (ROM data) stored in the program memory. The register-offset-PC-relative-addressing (PC + A) instructions can also be used, and the code conversion, table look-up and n-way multiple 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 ≧ 8000H for TMP87CM29)

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

Example 2: Converts BCD to 7-segment code (commom anode LED). When A = 05_H, 92_H is output to port P4 after executing the following program:

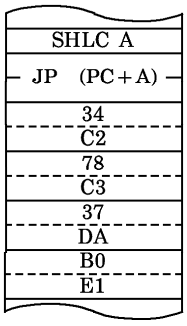
```
        ADD   A,TABLE – $ – 4      ;   P4 ← ROM (TABLE + A)
        LD    (P4), (PC + A)
        JRS   T, SNEXT
TABLE:  DB    0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0D8H, 80H, 98H
SNEXT:
```



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

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

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



Note: DW is a word data definition instruction.

1.3 Program Counter (PC)

The program counter (PC) is a 16-bit register which indicates the program memory address where the instruction to be executed next is stored. After reset, the user defined reset vector stored in the vector table (addresses $FFFF_H$ and $FFFE_H$) is loaded into the PC; therefore, program execution is possible from any desired address. For example, when $C0_H$ and $3E_H$ are stored at addresses $FFFF_H$ and $FFFE_H$, respectively, 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 $C125_H$.

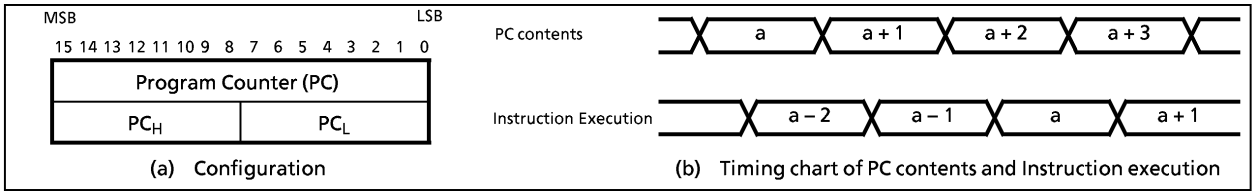


Figure 1-3. Program Counter

1.4 Data Memory (RAM)

The TMP87CH29/K29/M29 have 1K × 8-bit (addresses 0040_H to $043F_H$) 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 00C0H is "1", 00_H is writen to data memory at address $00E3_H$; otherwise, FF_H is written to the data memory at address 00E3.

```
        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: Incremonts the contents of data memory at address 00F5, and clears to 00 when 10_H is exceeded

```
INC     (00F5H)          ;  (00F5H) ← (00F5H) + 1
AND     (00F5H), 0FH     ;  (00F5H) Λ 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 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 ***TLCS-870 series***, programs in data memory cannot be executed. If the program counter indicates a data memory address, an address-trap-reset is generated due to bus error. (Output from the $\overline{\text{RESET}}$ pin goes low.)

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 general-purpose registers are mapped in the RAM; therefore, *do not clear RAM at the current bank address.*

Example 1: 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, 03F7H       ;  Sets number of byte to BC register pair
SRAMCLR:    LD      (HL+), A
            DEC     BC
            JRS     F, SRAMCLR
```

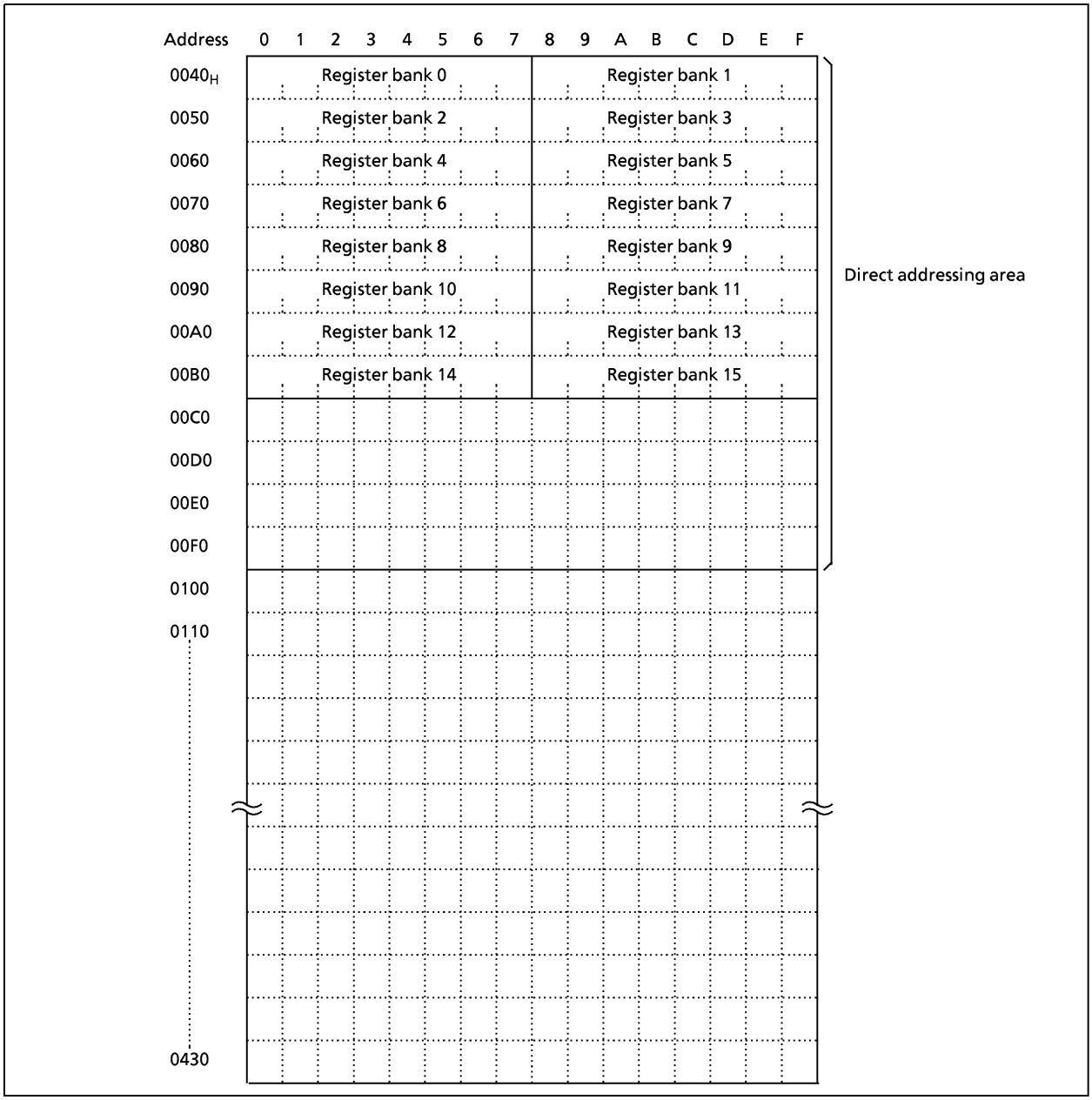


Figure 1-4. Data Memory Map

1.5 General-purpose Register Banks

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

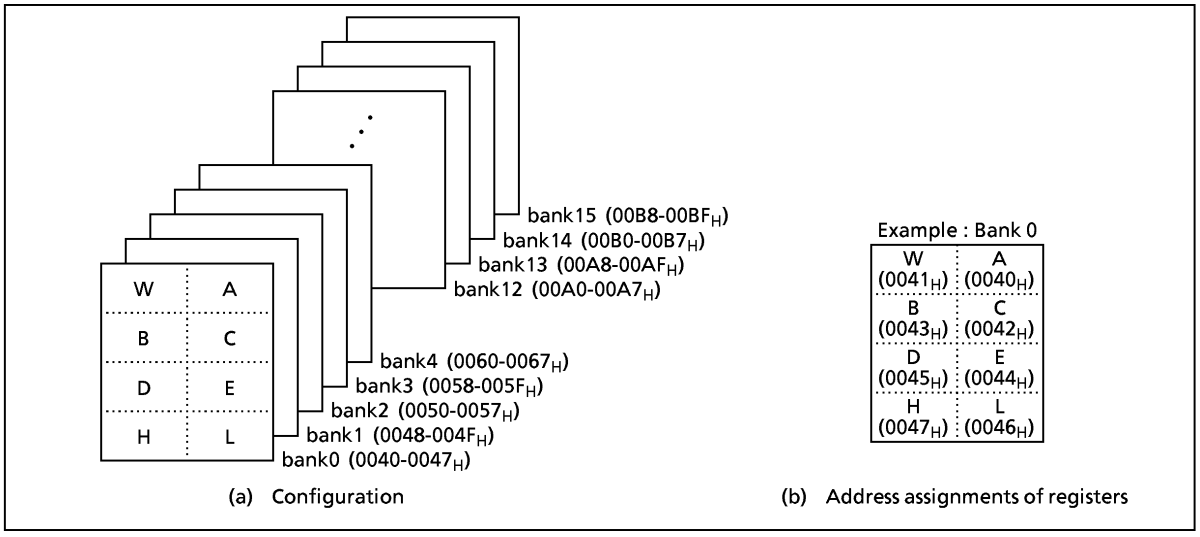


Figure 1-5. General-purpose Register Banks

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

(1) **A, WA**

The A register functions as an 8-bit accumulator and the WA 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 1234H from WA contents, and stores the result into WA.
```

(2) **HL, DE**

The HL and DE can 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 functions as a data pointer (DE). The HL also has an auto-post-increment and an auto-pre-decrement functions. This function simplifies multiple digit data processing, software LIFO (last-in first-out) processing, etc.

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

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

Example 2: Block transfer

```
        LD      B, m          ; m = n – 1 (n = Nunber of byte 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
        JRS     F, SLOOP
```

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

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

Example 1: Incrementing the RBS

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

Example 2: Reading the RBS

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

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

During interrupt, the PSW is automatically saved onto the stack. The bank used before the interrupt was accepted is restored automatically by executing an interrupt return instruction [RETI]/[RETN] ; therefore, there is no need for the RBS save/restore software processing. The TLCS-870 Series supports a maximum of 15 interrupt sources. One bank is assigned to the main task, and one bank can be assigned to each interrupt task. 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. 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. [LD RBS, n], [PUSH PSW] and [POP PSW] are the PSW access instructions.

1.6.1 Register Bank Selector (RBS)

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

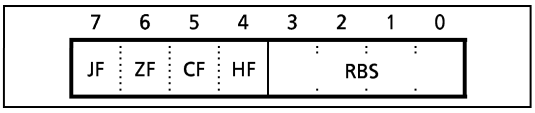


Figure 1-6. PSW (Flags, RBS) Configuration

1.6.2 Flags

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

cc	Means	Condition
T	True	JF = 1
F	False	JF = 0
Z	Zero	ZF = 1
NZ	Not Zero	ZF = 0
CS	Carry Set	CF = 1
CC	Carry Clear	CF = 0
EQ	Equal	ZF = 1
NE	Not Equal	ZF = 0
LT	Unsigned Less Than	CF = 1
GE	Unsigned Greater Than or Equal	CF = 0
LE	Unsigned Less Than or Equal	(CF$\vee$ZF) = 1
GT	Unsigned Greater Than	(CF$\vee$ZF) = 0

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

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

During the bit manipulation instruction [SET, CLR and CPL], it is also set to "1" if the contents of the specified bit is "0"; Otherwise it 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 (Quotient error); otherwise it is cleared. The CF is also affected during the shift/rotate instructions [SHLC, SHRC, ROLC and RORC]. The data shifted out from a register is set to the CF.

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

Example 1: Bit manipulation

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

Example 2: Arithmetic right shift

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

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

The HF is set to "1" when a carry occurred between bits 3 and 4 of the operation result during an 8-bit addition [ADD and ADDC], or when a borrow occurred from bit 4 into bit 3 of the result during an 8-bit subtraction[SUB, SUBB, CMP and MCMP]; 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).

Note that an undefined data is set to the HF during a 16-bit operation.

Example: BCD operation

(The a becomes 47_H after executing the following program when $A = 19_H$, $B = 28_H$)

```
ADD     A, B            ; A ← 41H, HF ← 1, CF = 0
DAA     A               ; A ← 41H + 06H = 47H (decimal-adjust)
```

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

Zero or carry information is set to the JF after operation (e.g. INC, ADD, CMP, TEST).

The JF provides the jump condition for conditional jump instructions [JRS T/F, $ + 2 + d], [JR T/F, $ + 2 + d] (T or F is a condition code). Jump is performed if the JF is "1" for a true condition (T), or the JF is "0" for a false condition (F).

The JF is normally 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: The accumulator and flags become as shown below after executing the following instructions when the WA register pair, the HL register pair, 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	Flags 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	Flags 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
SET A. 5	BA	1	1	1	0
ADD WA, 0F508H	16A2	1	0	1	0
MUL W, A	13DA	0	0	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 occupises three bytes.

When returning from the processing routine, executing a subroutine return instruction [RET] restores the contents of 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, a push, a software interrupt instruction is executed, or when an interrupt is accepted; and the SP is pre-incremented when a return or a pop instruction is executed. Figure 1-8 shows the stacking order.

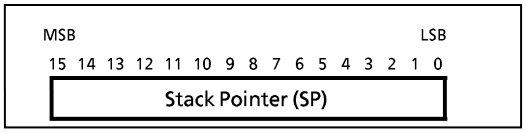


Figure 1-7. Stack Pointer

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

Example 1: To initialize the SP

```
LD      SP, 043FH       ; SP ← 043F_H
```

Example 2: To read the SP

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

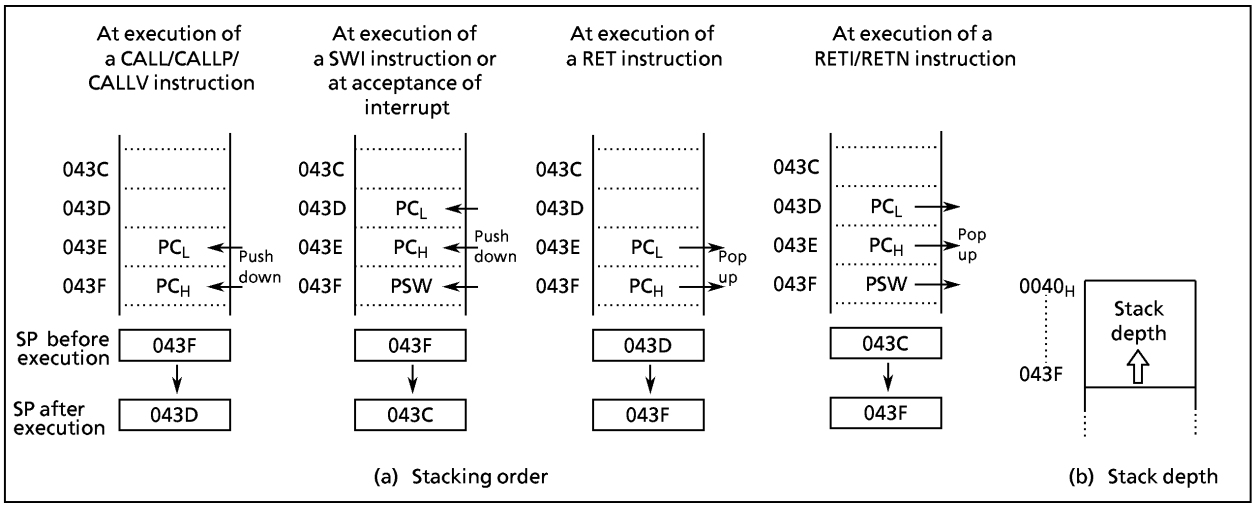


Figure 1-8. Stack

1.8 System Clock Controller

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

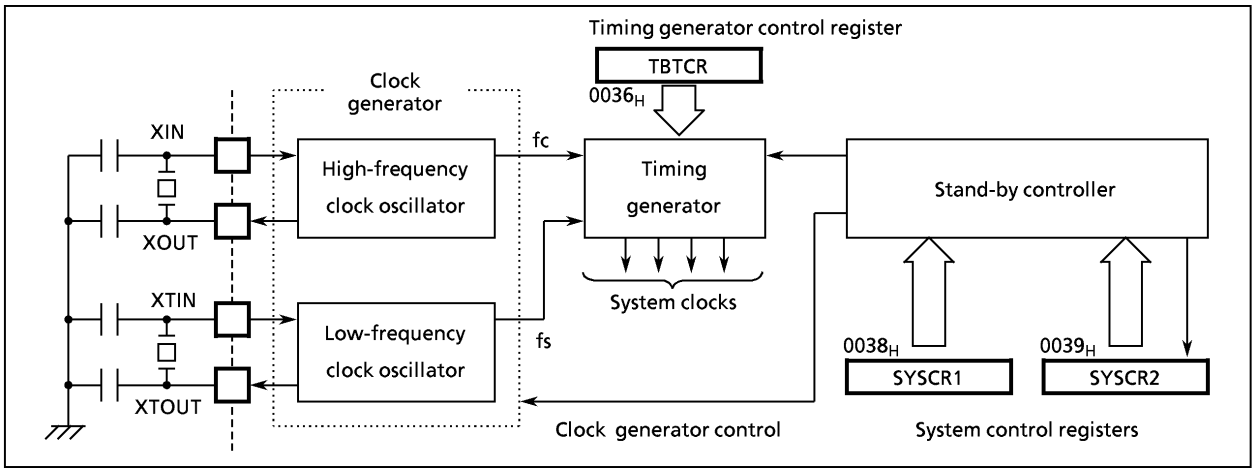


Figure 1-9. System Clock Controller

1.8.1 Clock Generator

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

The high-frequency (fc) and low-frequency (fs) clocks can be easily obtained by connecting a resonator between the XIN/XOUT and XTIN/XTOUT pins, respectively. Clock input from an external oscillator is also possible. In this case, external clock is applied to the XIN/XTIN pin with the XOUT/XTOUT pin not connected. The TMP87CH29/K29/M29 are not provided an RC oscillation.

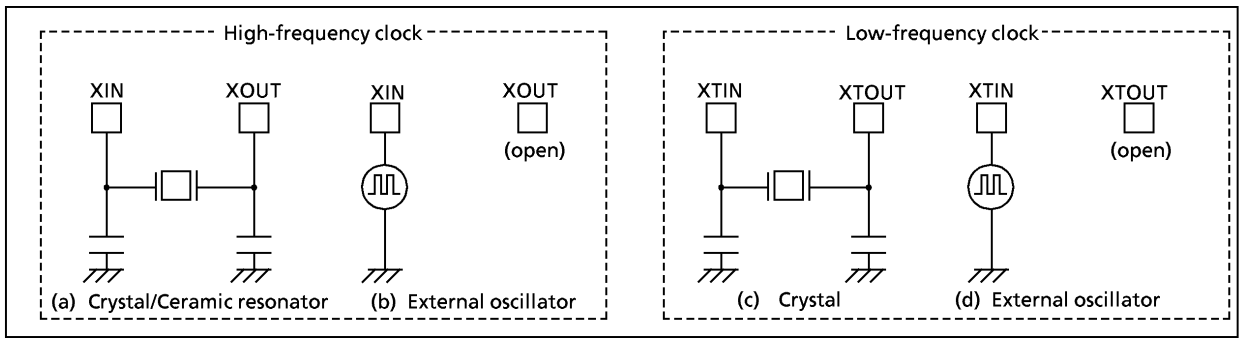


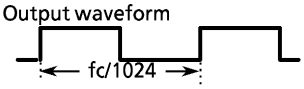
Figure 1-10. Examples of Resonator Connection

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

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

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

Output waveform



1.8.2 Timing Generator

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

① Generation of main system clock
② Generation of divider output ($\overline{DVO}$) pulses
③ Generation of source clocks for time base timer
④ Generation of source clocks for watchdog timer
⑤ Generation of internal source clocks for timer/counters (TC1, TC3 – TC6)
⑥ Generation of warm-up clocks for releasing STOP mode
⑦ Generation of base clocks for LCD driver/controller

(1) Configuration of timing generator

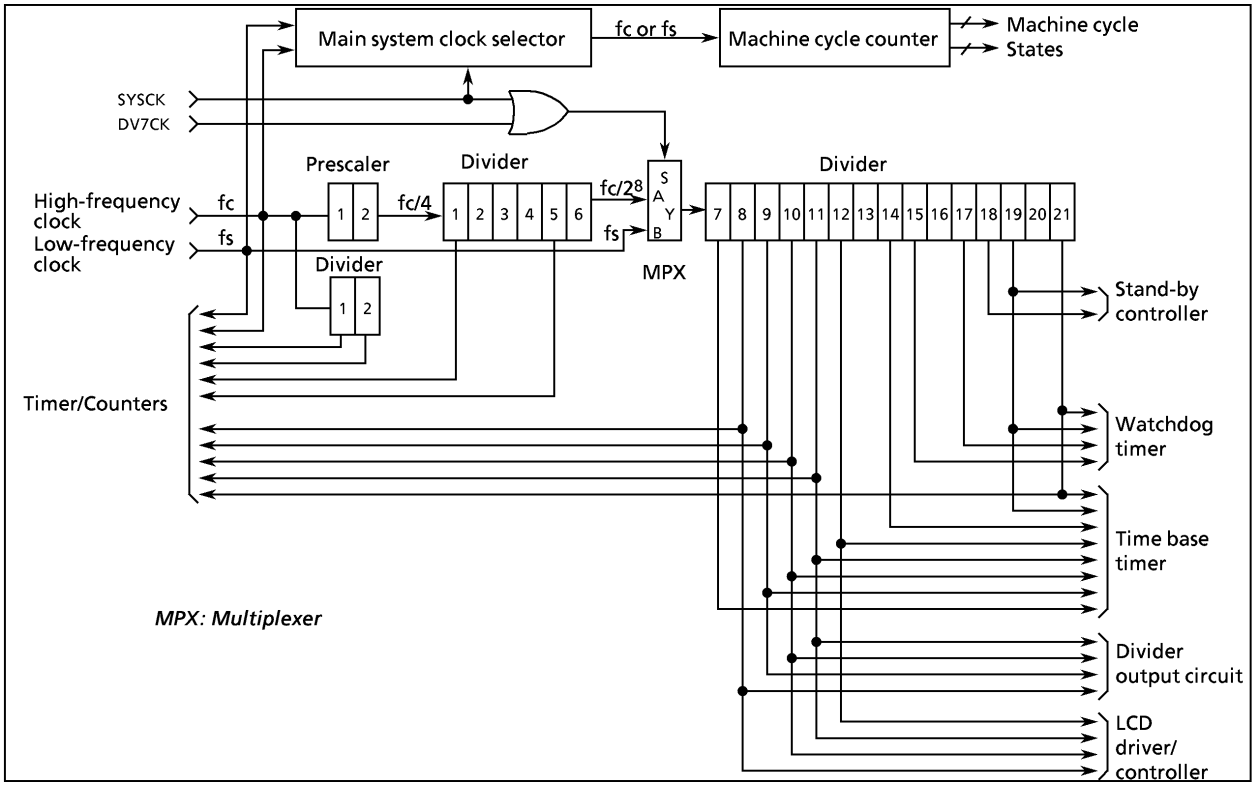


Figure 1-11. Configuration of Timing Generator

The timing generator consists of a 21-stage divider with a divided-by-2 prescaler, and a main system clock generator and machine cycle counters. An input clock to the 7th stage of the divider depends on the operating mode and DV7CK (bit 4 in TBTCR) shown in Figure 1-11 as follows.
During reset and upon releasing STOP mode, the divider is cleared to "0", however, the prescaler is not cleared.

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

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

TBTCR (0036_H)

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

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

DV7CK	Selection of input clock to the 7th stage of the divider	0: fc/2^8 [Hz] 1: fs	Write only

*Note1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care*
Note 2: Do not set DV7CK to "1" in the single-clock mode.
Note 3: Do not set DV7CK to "1" before low-frequency clock is stable in the dual-clock mode.
Note 4: TBTCR is a write-only register and must not be used with any of read-modify-write instructions.

Figure 1-12. Timing Generator Control Register

(2) Machine cycle

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

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

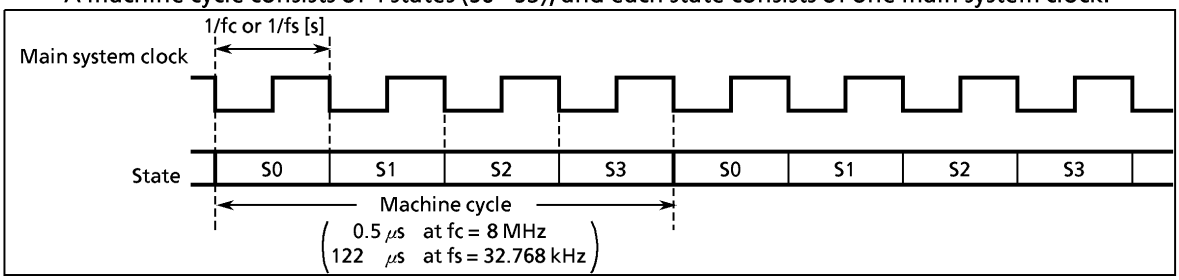


Figure 1-13. Machine Cycle

1.8.3 Stand-by Controller

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

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

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

(1) **Single-clock** mode

Only the oscillation circuit for the high-frequency clock is used, therefore 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 87CH29/K29/M29 are placed in this mode after reset.

② IDLE1 mode

In this mode, the internal oscillation circuit remains active. The CPU and the watchdog timer are halted; however, on-chip peripherals remain active (operate using the high-frequency clock). IDLE1 mode is started by setting IDLE bit in the system control register 2 (SYSCR2), and IDLE1 mode is released to NORMAL1 mode by an interrupt request from the on-chip peripherals or external interrupt inputs. When IMF (interrupt master enable flag) is "1" (interrupt enable), the execution will resume upon acceptance of the interrupt, and 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 the 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 input (either level-sensitive or edge-sensitive can be programmably selected) to the $\overline{\text{STOP}}$ pin. After the warming-up period is completed, the execution resumes with the next instruction which follows the STOP mode start instruction.

(2) **Dual-clock** mode

Both the high-frequency and low-frequency oscillation circuits are used in this mode. 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/IDLE2 modes, and 4/fs [s] (122μs at fc = 32.768 kHz) in SLOW/SLEEP modes.

*Note: The **TMP87PM29** is placed in the single-clock mode during reset. To use the dual-clock mode, the low-frequency oscillator should be turned on by executing **[SET (SYSCR2). XTEN]** instruction.*

① NORMAL2 mode

In this mode, the CPU core is operated using the high-frequency clock. The on-chip peripherals are operated using the high-frequency clock and/or low-frequency clock. In case that the dual-clock mode has been selected as an option, the TMP87CH29/K29/M29 are placed in this mode after reset.

② SLOW mode

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

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

③ IDLE2 mode

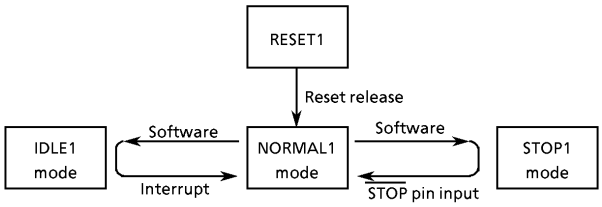
In this mode, the internal oscillation circuits remain active. The CPU and the watchdog timer are halted; however, on-chip peripherals operate using the high-frequency clock and/or low-frequency clock. Starting and releasing of IDLE2 mode are the same as for IDLE1 mode, except that operation returns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

④ SLEEP mode

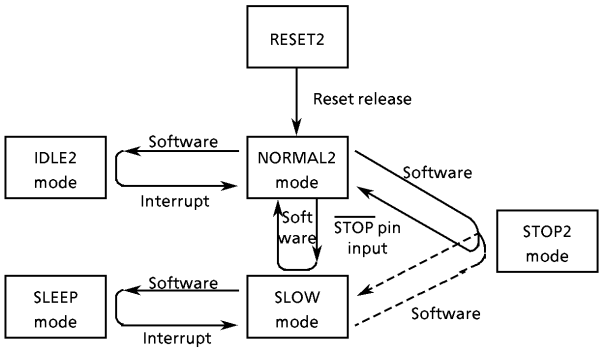
In this mode, the internal oscillation circuit of the low-frequency clock remains active. The CPU, the watchdog timer, and the internal oscillation circuit of the high-frequency clock are halted; however, on-chip peripherals operate using the low-frequency clock. Starting and releasing of SLEEP mode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

⑤ STOP2 mode

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



(a) Single-clock mode



(b) Dual-clock mode

Note 1: NORMAL1 and NORMAL2 modes are generically called NORMAL; STOP1 and STOP2 are called STOP; and IDLE1, IDLE2 and SLEEP are called IDLE.
Note 2: The TMP87PM29 doesn't have RESET2 mode.

Operating mode		Frequency		CPU core	On-chip Peripherals	Machine cycle time
		High-frequency	Low-frequency			
Single-Clock	RESET1	Turning on oscillation	Turning off oscillation	Reset	Reset	4/fc [s]
	NORMAL1			Operate	Operate	
	IDLE1			Halt		
	STOP1	Turning off oscillation			Halt	—
Dual-Clock	RESET2	Turning on oscillation	Turning on oscillation	Reset	Reset	4/fc [s]
	NORMAL2			High-frequency	Operate (High and/or Low)	
	IDLE2			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 (0038_H)

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

(Initial value: 0000 00**)

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

Note 1: Always set RETM to "0" when transiting from NORMAL1 mode to STOP1 mode and from NORMAL 2 mode to STOP 2 mode. Always set RETM to "1" when transiting from SLOW mode to STOP 2 mode.

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

Note 3: fc: *High-frequency clock* [Hz]
fs: *Low-frequency clock* [Hz]
*: *Don't care*

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

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

System Control Register 2

SYSCR2 (0039_H)

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

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

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

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

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

*Note 3 : WDT: Watchdog timer, *: Don't care*

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

Note 5 : An optional initial value can be selected for XTEN. Always specify when ordering ES (engineering sample).

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

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

Figure 1-15. System Control Registers

1.8.4 Operating Mode Control

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

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

① Oscillations are turned off, and all internal operations are halted.
② The data memory (except for DBR), registers and port output latches are all held in the status in effect before STOP mode was entered. The port output can be select either output hold or high-impedance by setting OUTEN (bit 4 in SYSCR1).
③ The divider of the timing generator is cleared to "0".
④ The program counter holds the address of the instruction following the instruction which started STOP mode.

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

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

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

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

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

Example: Starting STOP mode with an INT5 interrupt.

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

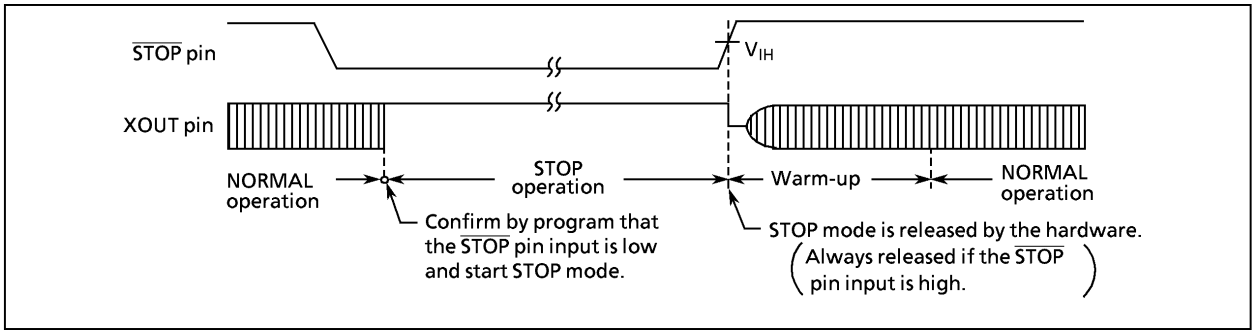


Figure 1-16. Level-sensitive Release Mode

Note 1: Even if the input of STOP pin is low level after starting warm-up operation, the STOP mode does not restart.

Note 2: When changing to the level-sensitive mode from the edge-sensitive release mode, the release mode is not switched until a rising edge of the $\overline{\text{STOP}}$ pin input is detected.

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

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

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

Example: Starting STOP mode operation in the edge-senstive release mode.

```
LD      (SYSCR1), 00000000B     ; OUTEN ← 0 (specifies high-impedance)
DI                              ; IMF ← 0 (disables interrupt service)
SET     (SYSCR1). STOP          ; STOP ← 1 (activates stop mode)
LDW     (IL), 1110011101010111B ; IL12, 11, 7, 5, 3 ← 0
                                  (clears interrupt latches)
EI                              ; IMF ← 1 (enables interrupt service)
```

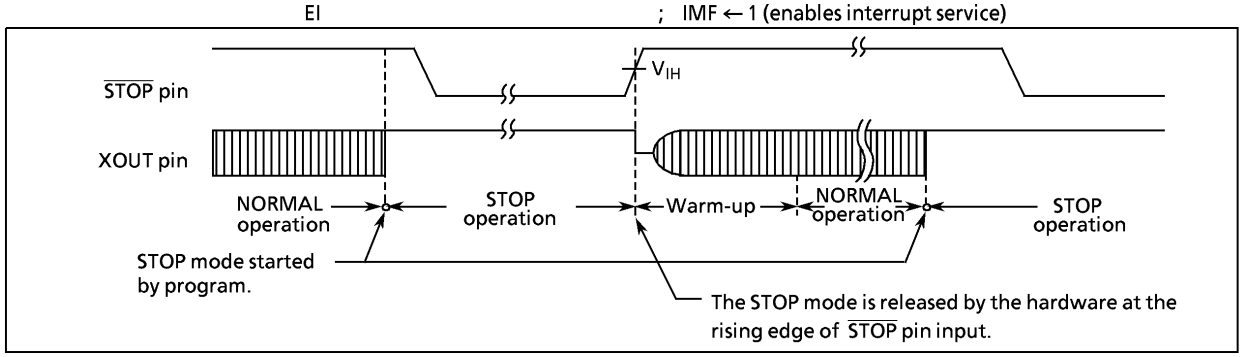


Figure 1-17. Edge-sensitive Release Mode

STOP mode is released by the following sequence:

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

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

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

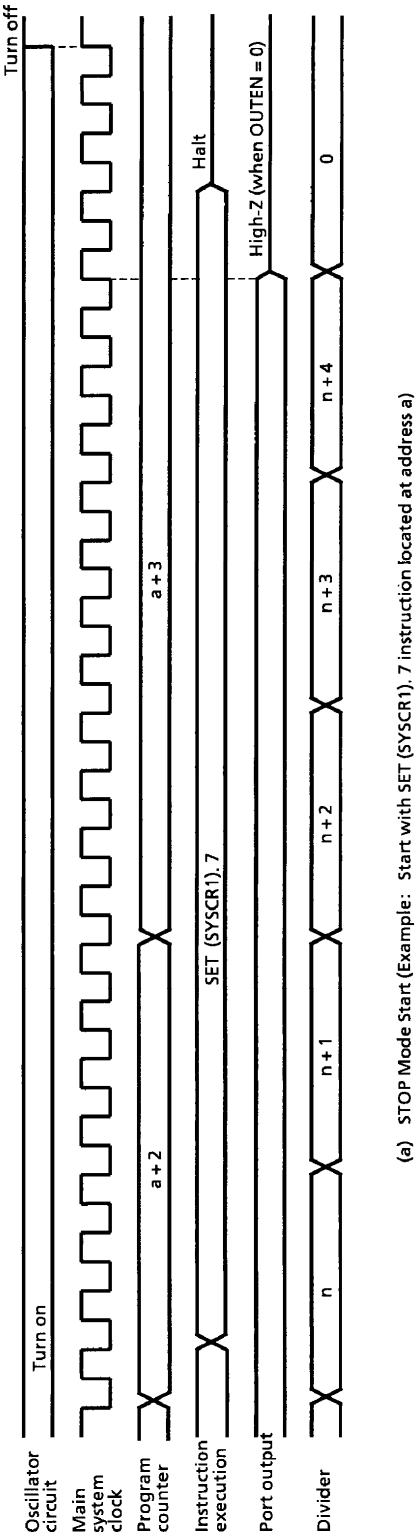
Table 1-1. Warming-up Time (Example)

Return to NORMAL 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

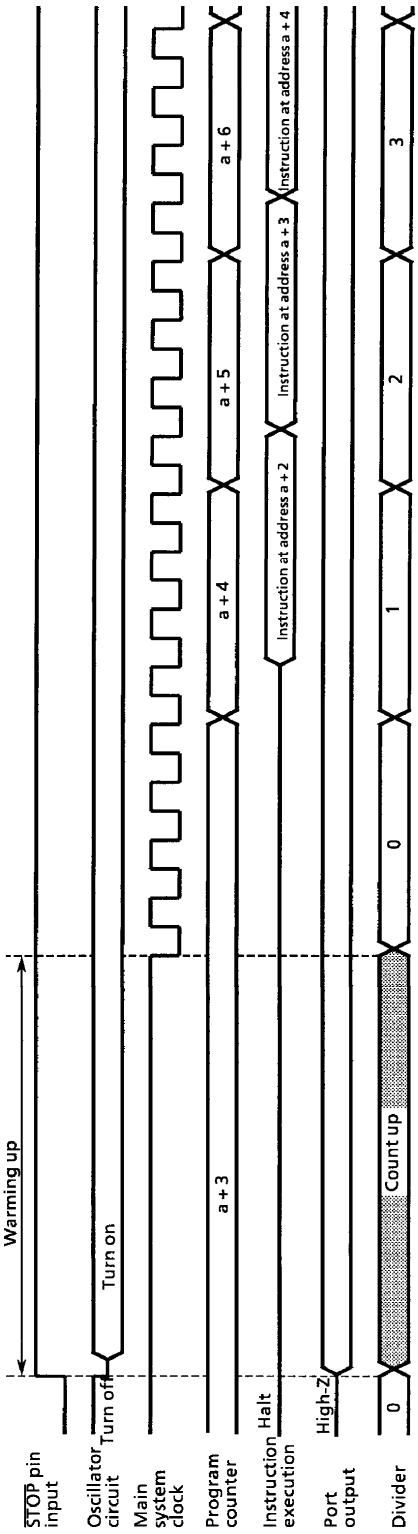
Note: The warming-up time is obtained by dividing the basic clock by the divider: therefore, the warming-up time may include a certain amount of error if there is any fluctuation of the oscillation frequency when STOP mode is released. Thus, the warming-up time must be considered an approximate value.

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. (If the initial XTEN of TMP87CH29/K29/M29 are set to "1" by mask option, they start from the NORMAL2 mode. In case of TMP87PM29 starts from NORMAL1 mode.)

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



(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

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

IDLE mode is controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following status is maintained during IDLE mode.

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

Example: Starting IDLE mode.

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

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

a. Normal release mode (IMF = "0")

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

b. Interrupt release mode (IMF = "1")

IDLE mode is released and interrupt processing is started by any interrupt source enabled by the individual interrupt enable flag (EF) or an external interrupt 0 ($\overline{INT0}$ pin) request. After the interrupt is processed, 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, the TMP87CH29/K29/M29 are placed in NORMAL mode.
The TMP87PM29 is placed in NORMAL1 mode after reset release.

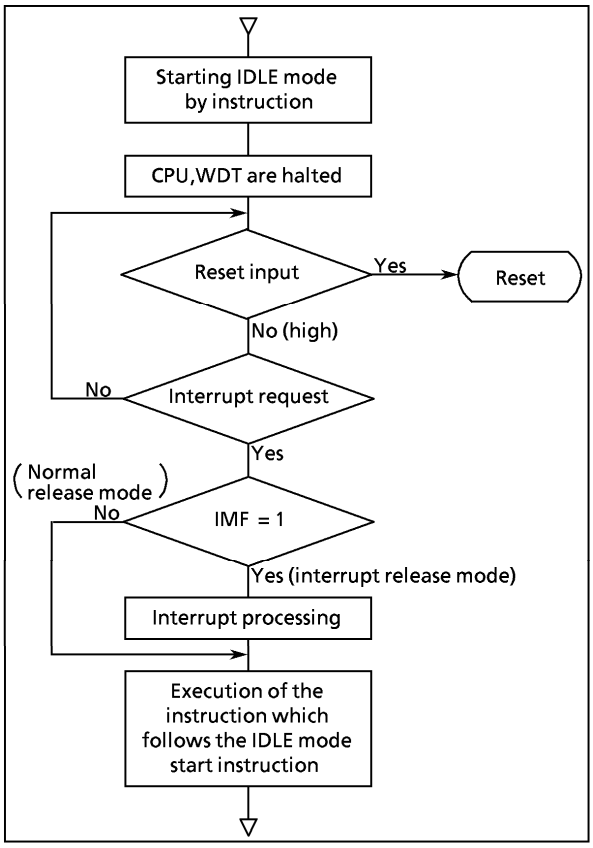


Figure1-19. IDLE Mode

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

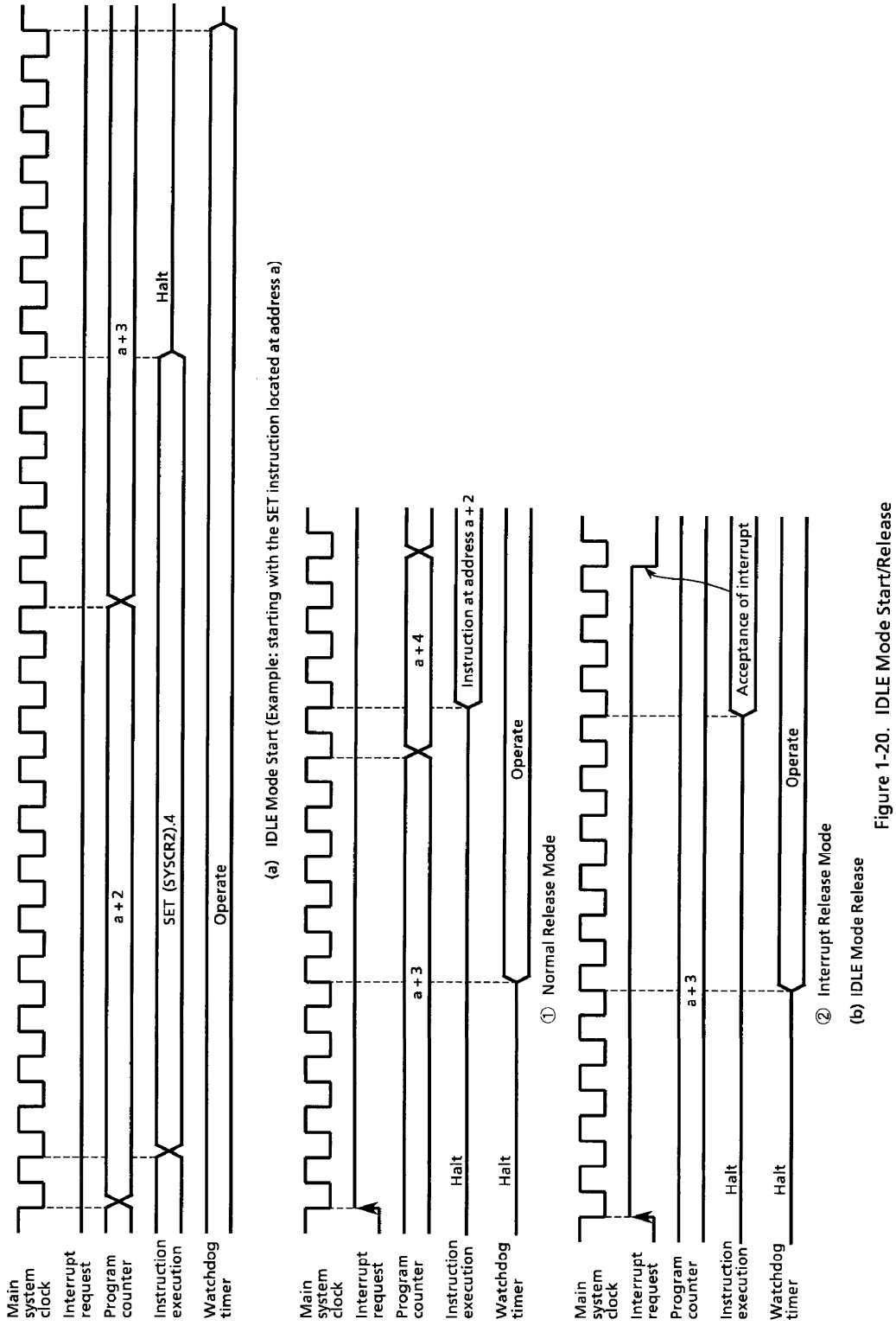


Figure 1-20. IDLE Mode Start/Release

(3) **SLOW** mode

SLOW mode is controlled by the system control register 2 (SYSCR2) and the timer/counter 1 (TC1).

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 from the high-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/counter1 (TC1) can conveniently be used to confirm that low-frequency clock oscillation has stabilized.

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

Example 1: 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)
```

Example 2: Switching to SLOW mode after low-frequency clock oscillation has stabilized.

```
        LD      (TC1CR) , 04H       ; Sets TC1 mode
                                      (timer mode, source clock: fs)
        LDW     (TREG1A$_L$) , 8000H  ; Sets warming-up time
                                      (according to Xtal characteristics)
        LD      (TREG1A$_H$) , 00H
        SET     (EIRH).EF8          ; Enables INTTC1
        LD      (TC1CR), 44H        ; Starts TC1
        :
PINTTC1: LD     (TC1CR), 04H        ; Stops TC1
        SET     (SYSCR2). 5         ; SYSCK ← 1
        CLR     (SYSCR2) . 7        ; XEN ← 0
        RETI
        :
VINTTC1: DW     PINTTC1             ; INTTC1 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 1 (TC1), clear SYSCK (bit 5 in SYSCR2) to switch the main system clock to the high-frequency clock.

Note 1: 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.

Note 2: SLOW mode can also be released by setting the $\overline{RESET}$ pin low, which immediately performs the reset operation. After reset, the TMP87CH29/K29/M29 are placed in NORMAL2 mode. (The TMP87PM29 is placed in NORMAL1 mode)

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

```
                SET     (SYSCR2) . 7          ;  XEN ← 1 (turns on high-frequency oscillation)
                LD      (TC1CR), 00H          ;  Sets the TC1 mode
                                              (Timer mode, source clock: fc)
                LDW     (TREG1AL), 0F800H     ;  Sets the warming up time
                                              (according to frequency and resonator characteristics).
                LD      (TREG1AH) , 00H
                SET     (EIRH).EF8            ;  Enables INTTC1
                LD      (TC1CR), 40H          ;  Starts TC1
                :
PINTTC1 :       LD      (TC1CR), 00H          ;  Stops TC1
                CLR     (SYSCR2) . 5          ;  SYSCK ← 0
                                              (switches the main system clock to the high-frequency clock)
                RETI
                :
VINTTC1 :       DW      PINTTC1               ;  INTTC1 vector table
```

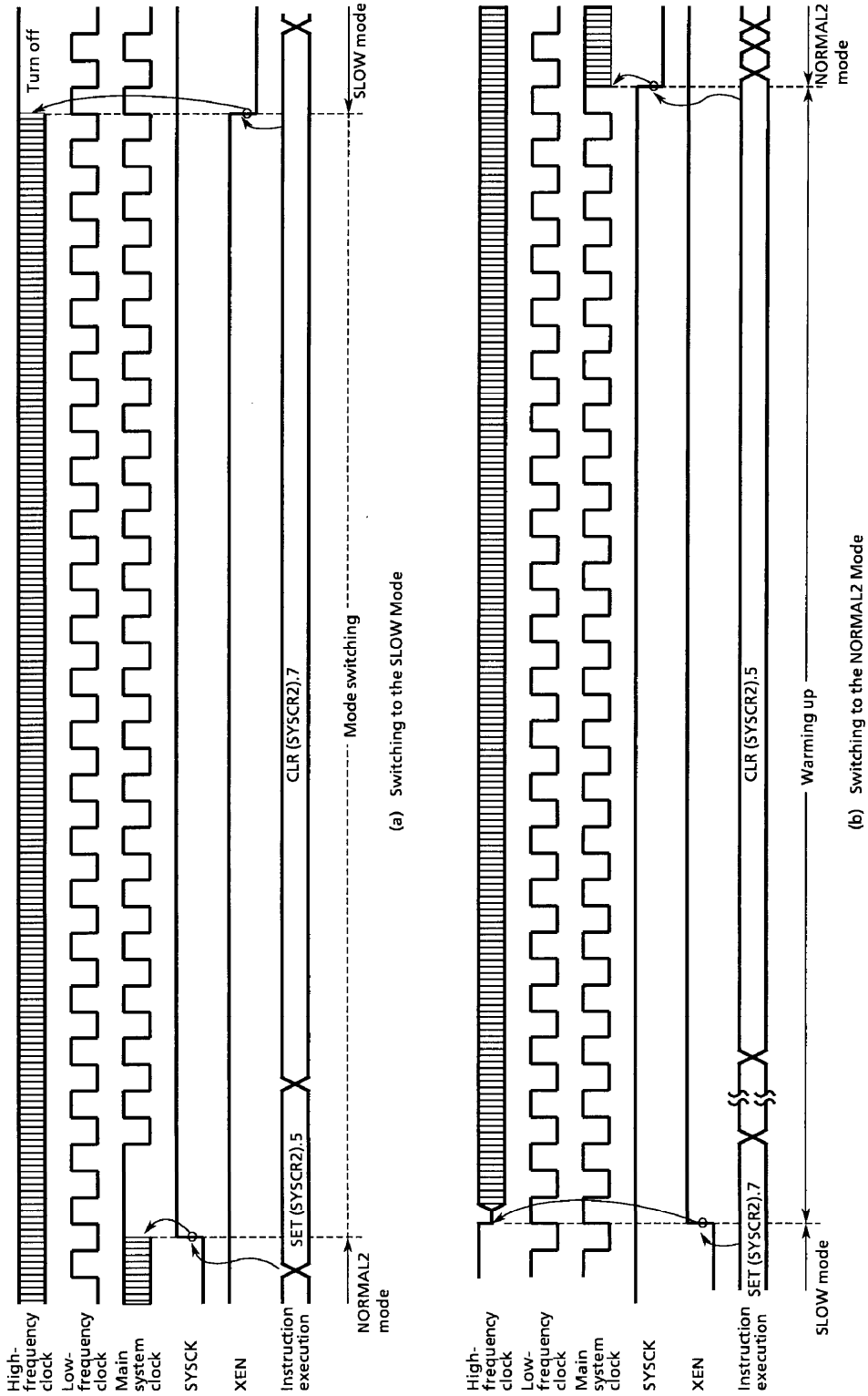


Figure 1-21. Switching between the NORMAL2 and SLOW Modes

1.9 Interrupt Controller

The TMP87CH29/K29/M29 each have a total of 13 interrupt sources: 4 externals and 9 internals. Nested interrupt control with priorities is also possible. Two of the internal sources are pseudo non-maskable interrupts; the remainder are all maskable interrupts.

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

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

Table 1-2. Interrupt Sources

Interrupt Source			Enable Condition	Interrupt Latch	VectorTable Address	Priority
Internal/ External	(Reset)		Non-Maskable	—	$FFFE_H$	High 0
Internal	INTSW	(Software Interrupt)	Pseudo non-maskable	—	$FFFC_H$	1
Internal	INTWDT	(Watchdog Timer Interrupt)		IL_2	$FFFA_H$	2
External	INT0	(External Interrupt 0)	IMF = 1, INT0EN = 1	IL_3	$FFF8_H$	3
Reserved			IMF · EF_4 = 1	IL_4	$FFF6_H$	4
External	INT1	(External Interrupt 1)	IMF · EF_5 = 1	IL_5	$FFF4_H$	5
Internal	INTTBT	(Time Base Timer Interrupt)	IMF · EF_6 = 1	IL_6	$FFF2_H$	6
External	INT2	(External Interrupt 2)	IMF · EF_7 = 1	IL_7	$FFF0_H$	7
Internal	INTTC1	(18-bit TC1 Interrupt)	IMF · EF_8 = 1	IL_8	$FFEE_H$	8
Internal	INTRX	(UART receive Interrupt)	IMF · EF_9 = 1	IL_9	$FFEC_H$	9
Internal	INTTX	(UART transmit Interrupt)	IMF · EF_{10} = 1	IL_{10}	$FFEA_H$	10
Internal	INTTC3	(8-bit TC3 Interrupt)	IMF · EF_{11} = 1	IL_{11}	$FFE8_H$	11
Internal	INTTC4	(8-bit TC4 Interrupt)	IMF · EF_{12} = 1	IL_{12}	$FFE6_H$	12
Internal	INTTC5	(8-bit TC5 Interrupt)	IMF · EF_{13} = 1	IL_{13}	$FFE4_H$	13
Reserved			IMF · EF_{14} = 1	IL_{14}	$FFE2_H$	14
External	INT5	(External Interrupt 5)	IMF · EF_{15} = 1	IL_{15}	$FFE0_H$	Low 15

(1) Interrupt Latches (IL_{15-2})

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

Interrupt latches are assigned to addresses $003C_H$ and $003D_H$ in the SFR. Each latch can be cleared to "0" individually by an instruction; however, the read-modify-write instructions 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: Clear interrupt latches

```
LDW   (IL), 1110100000111111B   ;   IL12, IL10 - IL6 ← 0
```

Example 2: Read interrupt latches

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

Example 3: Tests an interrupt latch

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

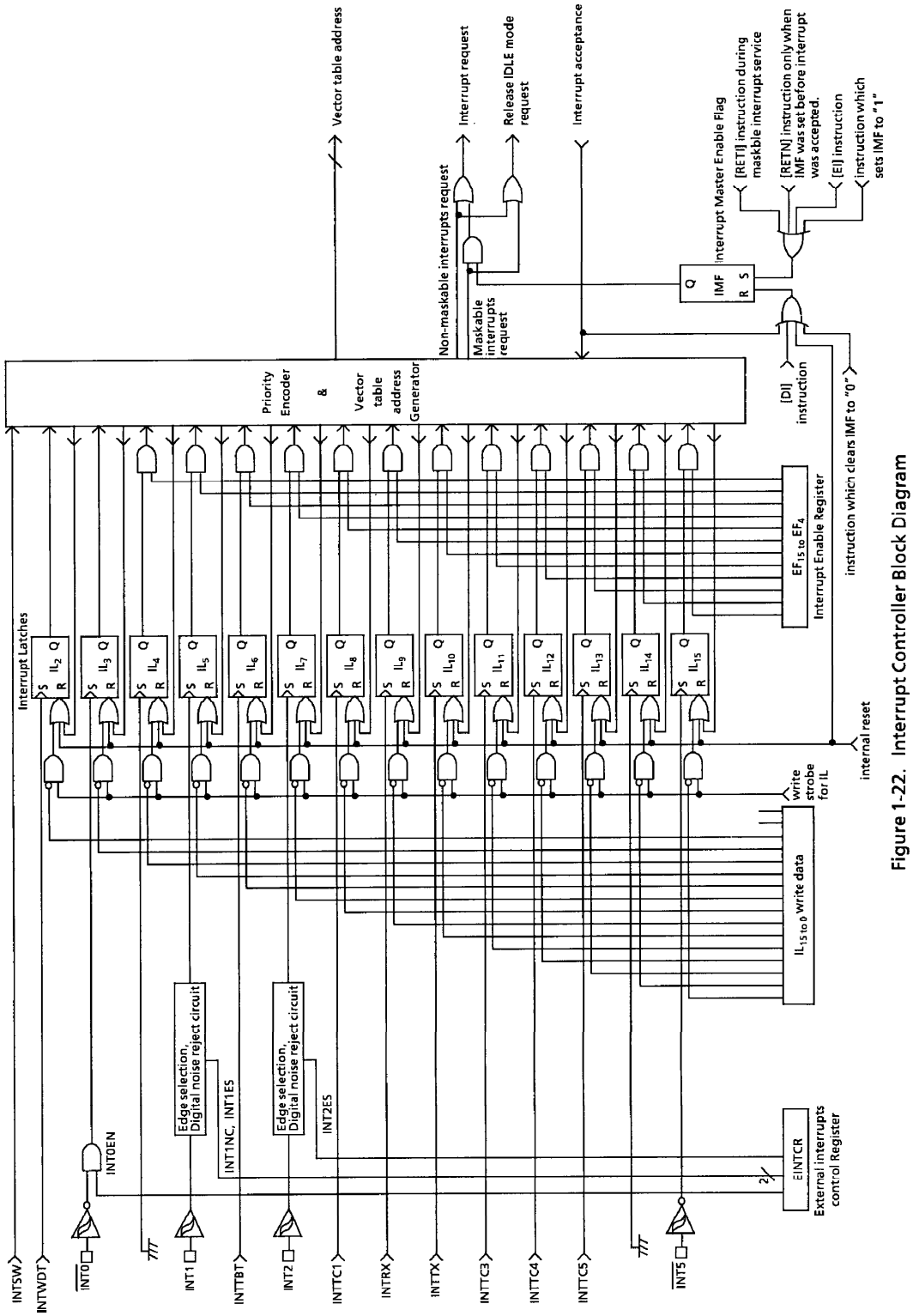


Figure 1-22. Interrupt Controller Block Diagram

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

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

The EIR consists of an interrupt master enable flag (IMF) and individual interrupt enable flags (EF). This 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).

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 other 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 occured, interrupt processing 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 processing is started with interrupt acceptance enabled (IMF = 1). Note that the IMF remains "0" when cleared in the interrupt service program.

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

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

② **Individual interrupt enable flags** (EF_{15} to EF_5)

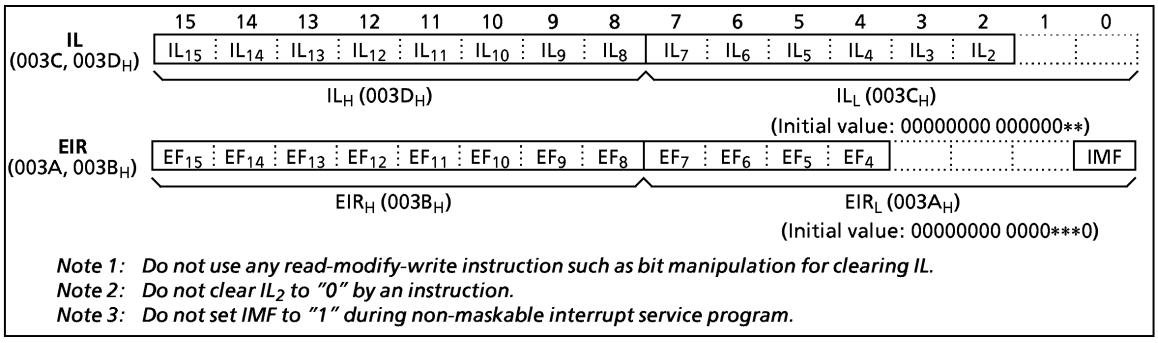
These flags enable and disable the acceptance of individual maskable interrupts, except for external interrupt 0. Setting the corresponding bit of an individual interrupt enable flag to "1" enables acceptance of an interrupt, setting the bit to "0" disables acceptance.

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

```
LDW    (EIR), 1010100010100001B      ;   EF15, EF13, EF11, EF7, EF5, IMF ← 1
```

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

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



Note 1: Do not use any read-modify-write instruction such as bit manipulation for clearing IL.
Note 2: Do not clear IL_2 to "0" by an instruction.
Note 3: Do not set IMF to "1" during non-maskable interrupt service program.

Figure 1-23. Interrupt Latch (IL) 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 the interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for pseudo non-maskable interrupts).

(1) Interrupt acceptance procesing

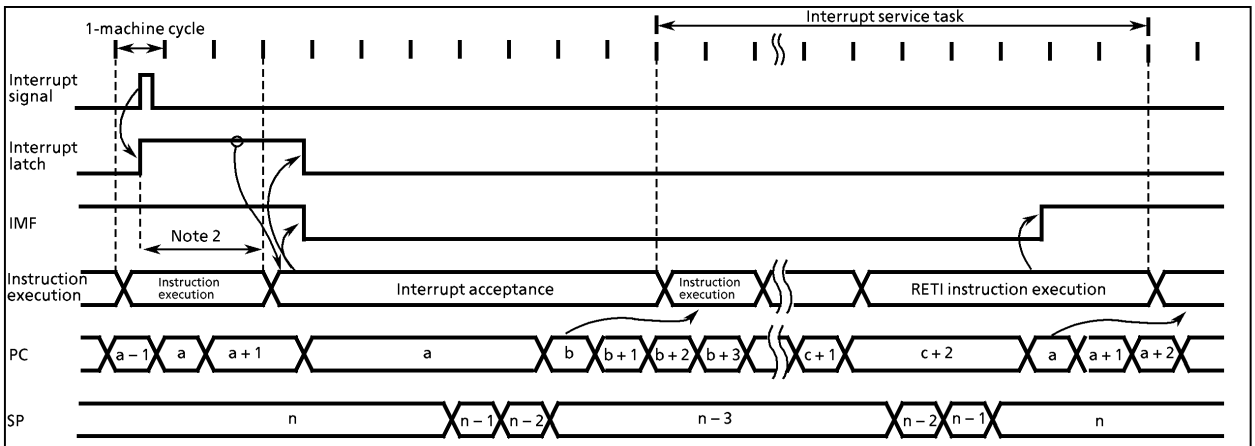
① The interrupt master enable flag (IMF) is cleared to "0" to temporarily disable the acceptance of any following maskable interrupts. When a non-maskable interrupt is accepted, the acceptance of any following interrupts is temporarily disabled.

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

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

④ The entry address of the interrupt service program is read from the vector table address for the interrupt source and load to the program counter.

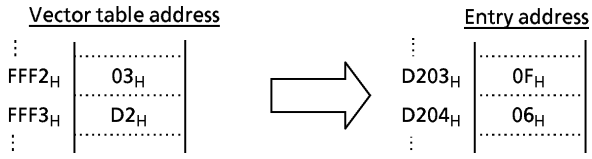
⑤ The instruction stored at the entry address of the interrupt service program is executed.



Note 1: a: Return address, b: Entry address, c: Address when the RETI instruction is stored
Note 2: The maximum response time from when an IL is set until an interrupt acceptance processing starts is 38/fc or 38/fs [s].

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

Example: Correspondence between vector addresses for INTTBT response processing 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 EF. However, an acceptance of external interrupt 0 cannot be disabled by the EF; therefore, if disablement is necessary, either the external interrupt function of the $\overline{\text{INT0}}$ pin must be disabled with INT0EN in the external interrupt control register or interrupt processing must be avoided by the program.
When INT0EN = 0, the interrupt latch IL3 is not set, therefore, the falling edge of the $\overline{\text{INT0}}$ pin input cannot be detected.

Example 1: Disables an external interrupt 0 using INT0EN

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

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

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

VINT0:   DW    PINT0
```

(2) General - purpose 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 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 switching register bank changeover:

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                ;   Switch to bank n (1 μs at 8 MHz)
         interrupt processing
         RETI                         ;   Bank restore and Return
```

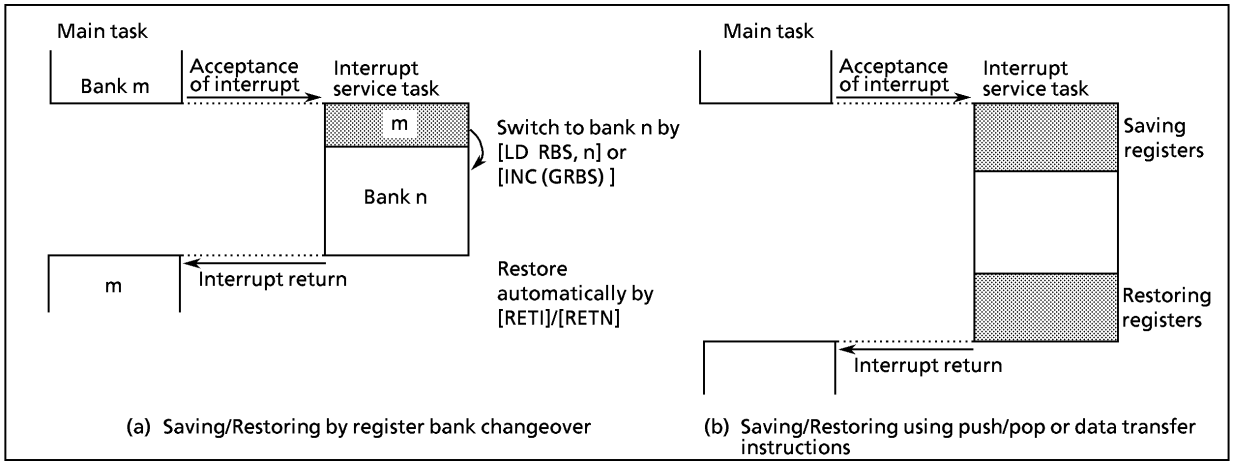
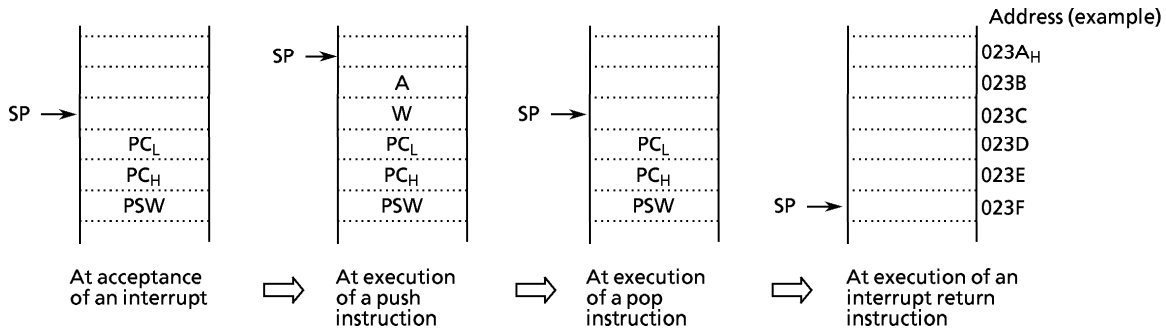


Figure 1-25. Saving/Restoring General-purpose Registers

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

Example: Register save using push and pop instructions

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



③ General-purpose registers save/restore using data transfer instructions:
Dara 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. ② The stack pointer is incremented 3 times. ③ The interrupt master enable flag is set to "1".	① The contents of the program counter and the program status word are restored from the stack. ② The stack pointer is incremented 3 times. ③ The interrupt master enable flag is set to "1" only when a non-maskable interrupt is accepted in interrupt enable status. However, the interrupt master enable flag remains at "0" when so clear by an interrupt service program.

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: If the interrupt processing time is longer than the time required to generate the interrupt request, only the interrupt service program may be executed 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 the development tool.

Use the [SWI] instruction only for detection of the address errors 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 (addresses 0040_H to $043F_H$) or SFR area (0000_H to $003F_H$).

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

② Debugging

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

1.9.3 External Interrupts

The TMP87CH29/K29/M29 each have four external interrupt inputs ($\overline{INT0}$, INT1, INT2, and $\overline{INT5}$). Two of these are equipped with digital noise rejection circuits (pulse inputs of less than a certain time are eliminated as noise).

Edge selection is also possible with INT1 and INT2. The $\overline{INT0}$/P10 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

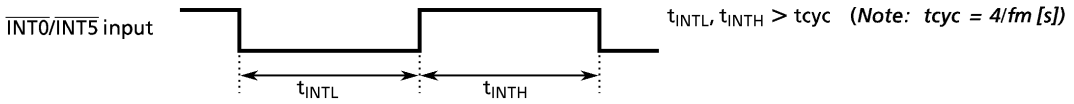
Edge selection, noise rejection control and $\overline{INT0}$/P10 pin function selection are performed by the external interrupt control register (EINTCR). When INT0EN = 0, the IL_3 will not be set even if the falling edge of $\overline{INT0}$ pin input is detected.

Table 1-3. External Interrupts

Source	Pin	Secondary function pin	Enable conditions	Edge	Digital noise reject
INT0	$\overline{INT0}$	P10	IMF = 1, INT0EN = 1	Falling edge	— (hysteresis input)
INT1	INT1	P11	IMF · EF_5 = 1	Falling edge or Rising edge	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses equal to or more than 48/fc [s] or 192/fc [s] are regarded as signals.
INT2	INT2	P12	IMF · EF_7 = 1		Pulses of less than 7/fc [s] are eliminated as noise. Pulses equal to or more than 24/fc [s] are regarded as signals.
INT5	$\overline{INT5}$	P20/$\overline{STOP}$	IMF · EF_{15} = 1	Falling edge	— (hysteresis input)

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

Note 2: The pulse width (both "H" and "L" level) for input to the $\overline{INT0}$ and $\overline{INT5}$ pins must be over 1 machine cycle.



Note 3: If a noiseless signal is input to the external interrupt pin in the NORMAL1/2 or IDLE1/2 mode, the maximum time from the edge of input signal until the IL is set is as follows :

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

② INT2 pin 25/fc [s]

Note 4: In INT0EN = 0, even if the falling edge of INT0 pin input is detected, interrupt latch IL3 is not set.

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

Example: Activating stop mode

```
LD  (SYSCR1),01000000B      ; OUTEN←0 (specifies high-impedance)
DI                          ; IMF←0 (disables interrupt service)
SET (SYSCR1).STOP           ; STOP←1 (activates stop mode)
LDW (IL),1111111101010111B ; IL7,5,3←0 (clears interrupt latches)
EI                          ; IMF←1 (enables interrupt service)
```

EINTCR (0037_H)

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

(Initial value: 00** *00*)

INT1NC	Noise reject time select	0: Pulses of less than 63/fc [s] are eliminated as noise 1: Pulses of less than 15/fc [s] are eliminated as noise	Write only
INT0EN	P10/$\overline{INT0}$ pin configuration	0: P10 input/output port 1: $\overline{INT0}$ pin (Set port P10 to input mode)	
INT2ES INT1ES	INT2 and INT1 edge select	0: Rising edge 1: Falling edge	

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

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

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

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

Note 5: If changing the contents of INT1ES during NORMAL1/2 mode, interrupt latch of external interrupt input INT1 must be cleared after 14 machine cycles (when INT1NC = 1) or 50 machine cycles (when INT1NC = 0) from the time of changing. During SLOW mode, 3 machine cycles are required.

Figure 1-26. External Interrupt Control Register

1.10 Watchdog Timer (WDT)

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

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

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

1.10.1 Watchdog Timer Configuration

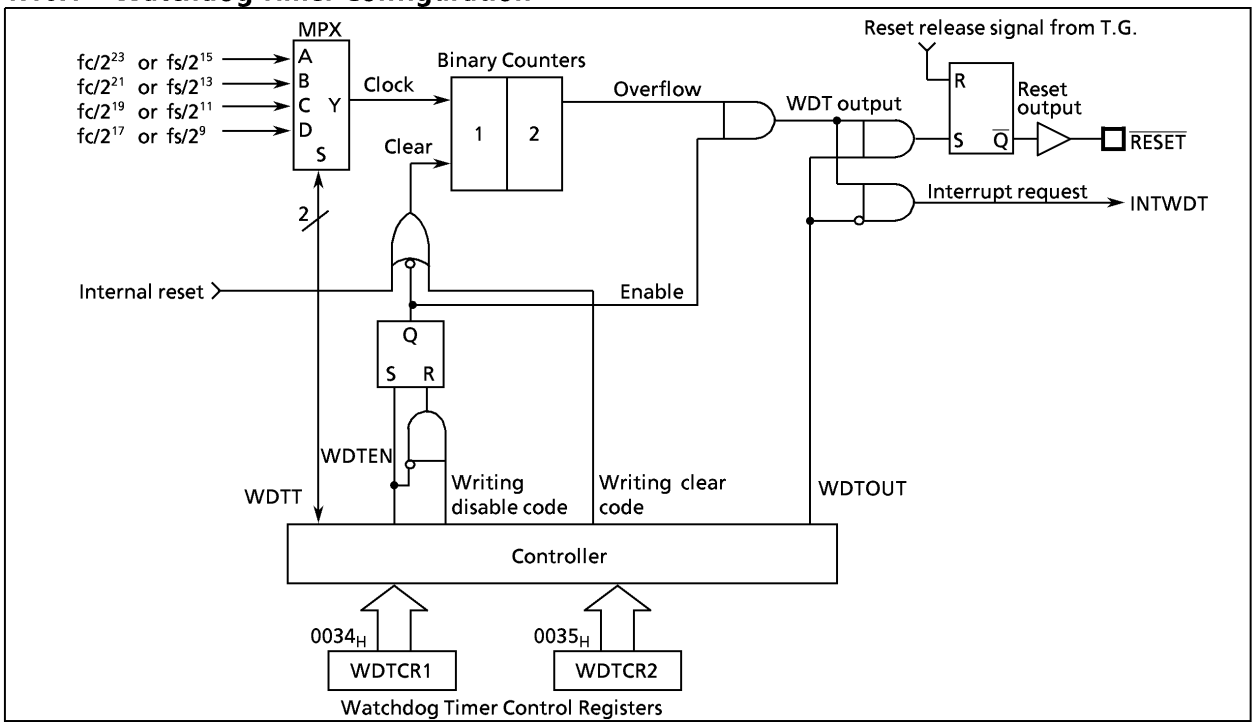


Figure 1-27. Watchdog Timer Configuration

1.10.2 Watchdog Timer Control

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

(1) Malfunction detection methods using the watchdog timer

The CPU malfunction is detected as follows:

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

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

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

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

Watchdog Timer Control Register 1

WDTCR1 (0034_H)

7	6	5	4	3	2	1	0
				WDT EN	WDTT		WDT OUT

(Initial value: **** 1001)

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

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

Watchdog Timer Control Register 2

WDTCR2 (0035_H)

7	6	5	4	3	2	1	0

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

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

Note 1: The disable code is invalid unless written when WDTEN = 0.
*Note 2: *: Don't care*
Note 3: Since WDTCR2 is a write-only register, read-modify-write instructions (e.g., bit manipulating instructions such as SET or CLR and arithmetic instructions such as AND or OR) cannot be used for read/write to this register.
Note 4: To clear binary counter doesn't initialize the source clock, therefore, it is recommended to clear binary counter within 3/4 of the detection period.

Figure 1-28. Watchdog Timer Control Registers

Table 1-4. Watchdog Timer Detection Time

Operating mode			Detection time	
NORMAL1	NORMAL2	SLOW	At fc = 8 MHz	At fs = 32.768 kHz
2^{25}/fc [s]	2^{25}/fc, 2^{17}/fs	2^{17}/fs	4.194 s	4 s
2^{23}/fc	2^{23}/fc, 2^{15}/fs	2^{15}/fs	1.048 s	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

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

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

(2) Watchdog timer enable

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

Example: Enables watchdog timer

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

(3) Watchdog timer disable

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

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

Example: Disables watchdog timer

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

1.10.3 Watchdog Timer Interrupt (INTWDT)

This is a pseudo non-maskable interrupt which can be accepted regardless of the contents of the EIR. If a watchdog timer interrupt or software interrupt is already accepted, however, the new watchdog timer interrupt waits until the previous 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:

```
LD    SP, 023FH             ;   Sets the stack pointer
LD    (WDTCR1), 00001000B   ;   WDTOUT ← 0
```

1.10.4 Watchdog Timer Reset

If the watchdog timer output becomes active, a reset is generated, which drives the $\overline{\text{RESET}}$ pin (sink open drain output) low to reset the internal hardware. The reset output time is 12/fc [s] (1.5 μs at fc = 8 MHz).

Note: The high-frequency clock oscillator also turns on when a watchdog timer reset is generated in SLOW mode so reset time is 8/fc to 24/fc [s]. When there is distortion on oscillation frequency at a start of high-frequency clock oscillation, the reset time includes measuring errors. It is recommended to be designated as an approximate value.

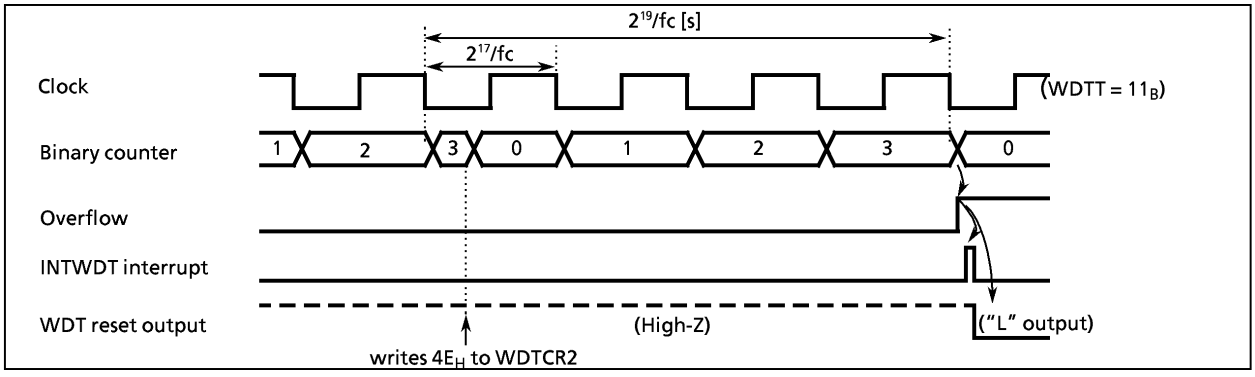


Figure 1-29. Watchdog Timer Interrupt/Reset

1.11 Reset Circuit

The TMP87CH29/K29/M29 each have four types of reset generation procedures: an external reset input, an address-trap- reset, a watchdog timer reset and a system-clock-reset. Table 1-5 shows on-chip hardware initialization by reset action. The internal source reset circuit (watchdog timer reset, address trap reset, and system clock reset) is not initialized when power is turned on. Thus, output from the $\overline{\text{RESET}}$ pin may go low 12/fc [s] (1.5 μs at 8 MHz), when power is turn on.

Table 1-5. Initializing Internal Status by Reset Action

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

1.11.1 External Reset Input

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

When the $\overline{RESET}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses $FFFE_H$ and $FFFF_H$.

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

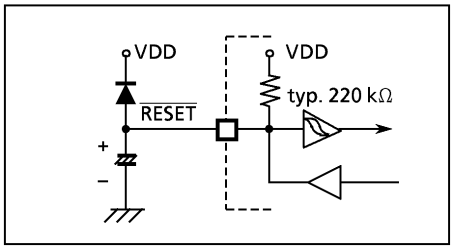
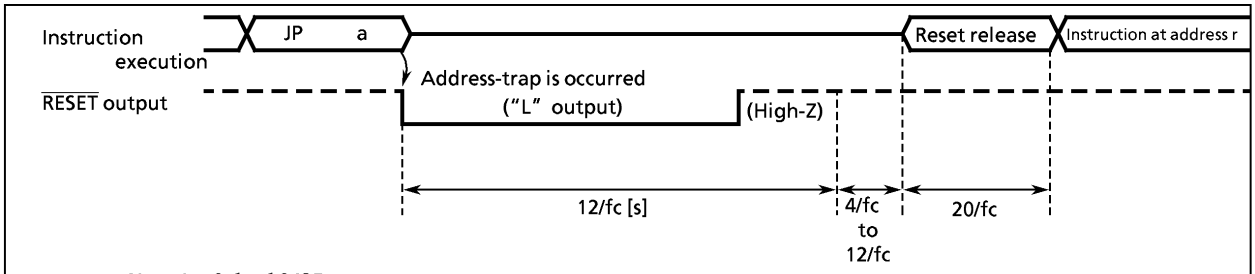


Figure 1-30. Simple Power-on-Reset Circuit

1.11.2 Address-Trap-Reset

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



Note 1: $0 \leqq a \leqq 043F_H$
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 "1.10 Watchdog Timer".

1.11.4 System-Clock-Reset

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

2. 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 TMP87CH29/K29/M29 SFRs and DBRs.

Address	Read	Write
0000_H	Reserved	
01	P1 Port	
02	P2 Port	
03	P3 Port	
04	P4 Port	
05	P5 Port	
06	P6 Port	
07	P7 Port	
08	Reserved	
09	Reserved	
0A	Reserved	
0B	–	P1CR (P1 I/O control)
0C	–	P4CR (P4 push-pull control)
0D	–	P5CR (P5 I/O control)
0E	ADCCR (AD converter control)	
0F	ADCDR (AD conv. result)	–
10	$TREG1A_L$	
11	$TREG1A_M$ (Timer register1A)	
12	$TREG1A_H$	
13	–	TREG1B (Timer register1B)
14	TC1SR (TC1 status)	TC1CR1 (TC 1 control)
15	–	TC1CR2 (TC 1 control)
16	–	TREG6 (Timer register6)
17	–	TC6CR (TC 6 control)
18	TREG3A (Timer register3A)	
19	TREG3B (Timer register3B)	–
1A	–	TC3CR (TC 3 control)
1B	–	TREG4 (Timer register4)
1C	–	TC4CR (TC 4 control)
1D	–	TREG5 (Timer register5)
1E	–	TC5CR (TC 5 control)
1F	Reserved	

Address	Read	Write
0020_H	UARTSR (UART status register)	UARTCR1 (UART control)
21	Reserved	UARTCR2
22	Reserved	
23	Reserved	
24	Reserved	
25	Reserved	
26	Reserved	
27	Reserved	
28	–	LCDCR (LCD control)
29	–	P6CR (P6 I/O control)
2A	–	P7CR (P7 I/O control)
2B	Reserved	
2C	Reserved	
2D	Reserved	
2E	Reserved	
2F	Reserved	
30	Reserved	
31	Reserved	
32	Reserved	
33	Reserved	
34	–	WDTCR1 (WDT control)
35	–	WDTCR2 (WDT control)
36	TBTCR (TBT control)	
37	EINTCR (external interrupt control)	
38	SYSCR1 (system control)	
39	SYSCR2 (system control)	
3A	EIR_L (interrupt enable register)	
3B	EIR_H (interrupt enable register)	
3C	IL_L (interrupt latch)	
3D	IL_H (interrupt latch)	
3E	Reserved	
3F	PSW (program status word)	RBS (register bank selector)

(a) Special Function Registers

Address	Read	Write
$0F80_H$	LCD display data buffer	
0F8B		
0F8C	Reserved	
0FEA		
0FEB	UART data buffer	
0FEC	Reserved	
0FFF		

(b) Data Buffer Registers

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.

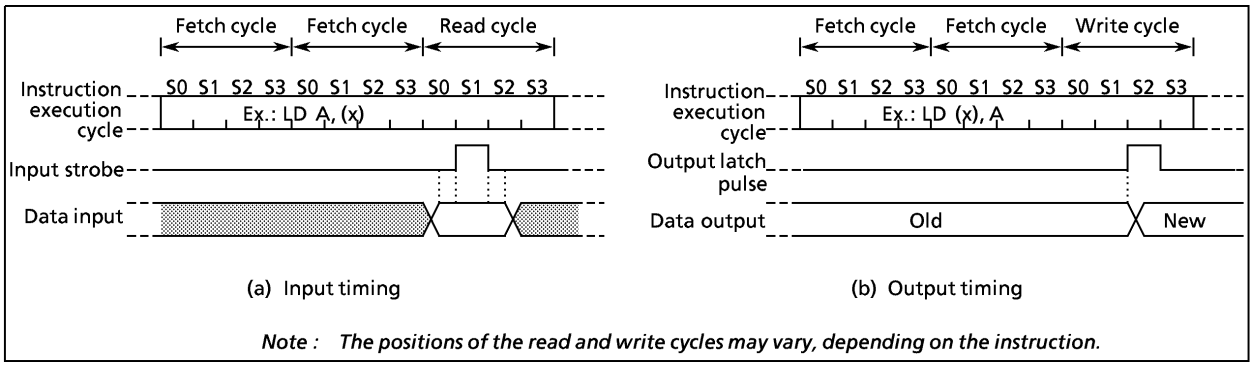
Figure 2-1. SFR and DBR

2.2 I/O Ports

The TMP87CH29/K29/M29 each have seven parallel input/output ports (43pins) each as follows:

	Primary Function	Secondary Functions
Port P1	8-bit I/O port	External interrupt input, and divider output.
Port P2	3-bit I/O port	Low-frequency resonator connection, external interrupt input, STOP mode release signal input.
Port P3	3-bit I/O port	Timer/counter output.
Port P4	8-bit I/O port	UART input and output, timer/counter input.
Port P5	5-bit I/O port	Analog input.
Port P6	8-bit I/O port	Segment output.
Port P7	8-bit I/O port	Segment output.

All output ports have latches, so output data are held by latching. All input ports have no latch, 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 read cycle during execution of the read instruction. This timing can not be recognized from outside, so that the transient input such as chattering must be processed by the program.
Data output is changed in the S2 state of write cycle during execution of the instruction which writes to an I/O port.



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)
- ② CLR/SET/CPL (src). b
- ③ CLR/SET/CPL (pp). g
- ④ LD (src). b, CF
- ⑤ LD (pp). b, CF
- ⑥ ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), n
- ⑦ (src) side of ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (HL)

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

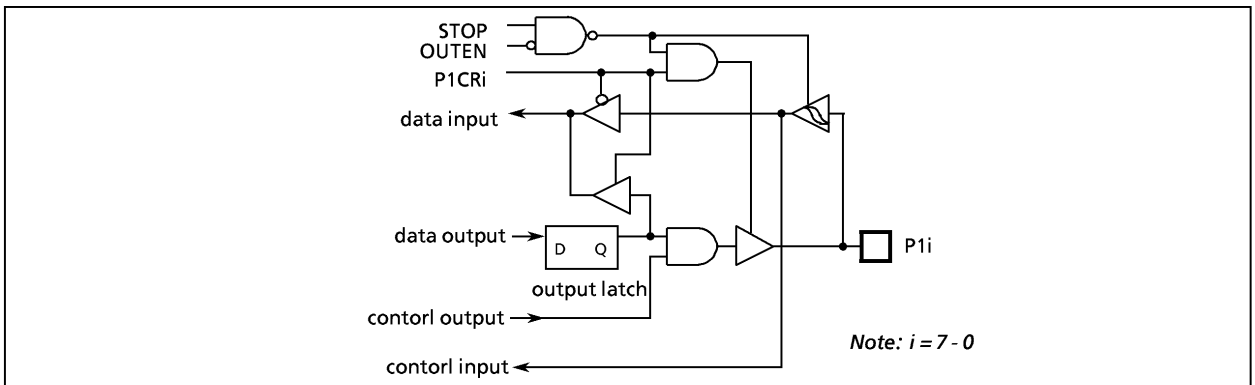
2.2.1 Port P1 (P17 to P10)

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, 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 an external interrupt input, and a divider output. When used as these secondary function pins, the input pins should be set to the input mode, and the output pin 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, or input ports. The interrupt latch is set on the rising or falling edge of the output when used as an output port. Pin P10 ($\overline{INT0}$) can be configured as either an I/O port or an external interrupt input with the INT0EN (bit 6 in EINTCR). During reset, pin P10 ($\overline{INT0}$) is configured as an input port P10.

Example: Sets P17, P16 as output ports, P15 and P14 as input ports, and the others as function pins. Internal output data is "1" for the P17, and "0" for the P16 pin.

```
LD      (EINTCR), 01000000B     ; INT0EN ← 1
LD      (P1), 10111111B         ; P17 ← 1, P16 ← 0, P13 ← 1
LD      (P1CR), 11001000B
```



	7	6	5	4	3	2	1	0
P1 (0001_H)	P17	P16	P15	P14	P13 / $\overline{DVO}$	P12 / INT2	P11 / INT1	P10 / $\overline{INT0}$

(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0
P1CR ($000B_H$)								

(Initial value: 0000 0000)

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

Note: P1CR is a write-only register and must not be used with any of the read-modify-write instructions.

Figure 2-3. Port P1 and P1CR

Note 1: Ports set to the input mode read the pin states. When input pin and output pin exist 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.)

2.2.2 Port P2 (P22 to P20)

Port P2 is a 3-bit input/output port. It is also used as an external interrupt input, STOP mode release signal input, and as low-frequency crystal connection pins. When used as an input port, or as 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 P20 output pulse. When a read instruction is executed for port P2, bits 7 to 3 in P2 are read in as undefined data.

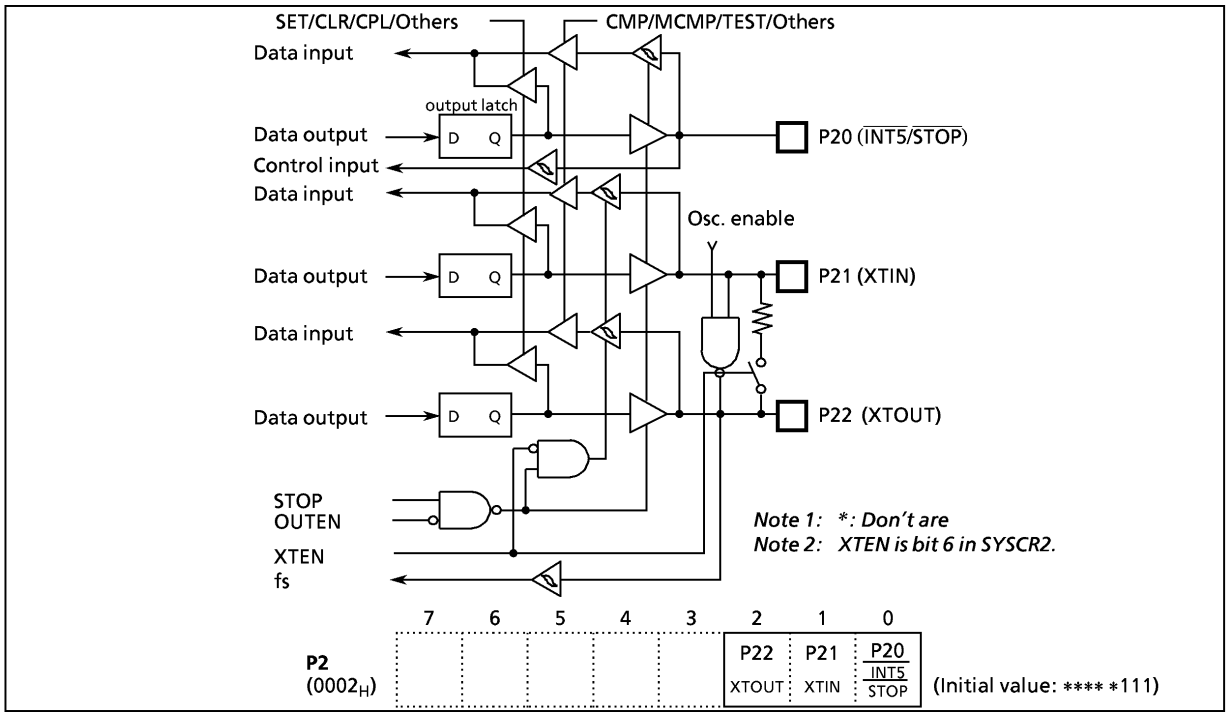


Figure 2-4. Port P2

2.2.3 Port P3 (P32 to P30)

Port P3 is a 3-bit input/output port. It is also used as a timer/counter output. Pins P30, P31 and P32 can be also used as an output of $\overline{PWM0/PDO0}$, $\overline{PWM1/PDO1}$ and $\overline{PWM2/PDO2}$ respectively, and only these pins are available high current output, so LEDs can be driven directly.

When used as an input or a secondary function pin, the output latch should be set to "1".

The output latches are initialized to "1" during reset.

Example 1: Outputs an immediate data FA_H to port P3.

```
LD    (P3), 0FAH          ;    P3 ← FAH
```

Example 2: Inverts the output of the upper 3 bits (P32 - P30) in port P3.

```
XOR   (P3), 00000111B     ;    P32 - P30 ← P32 - P30
```

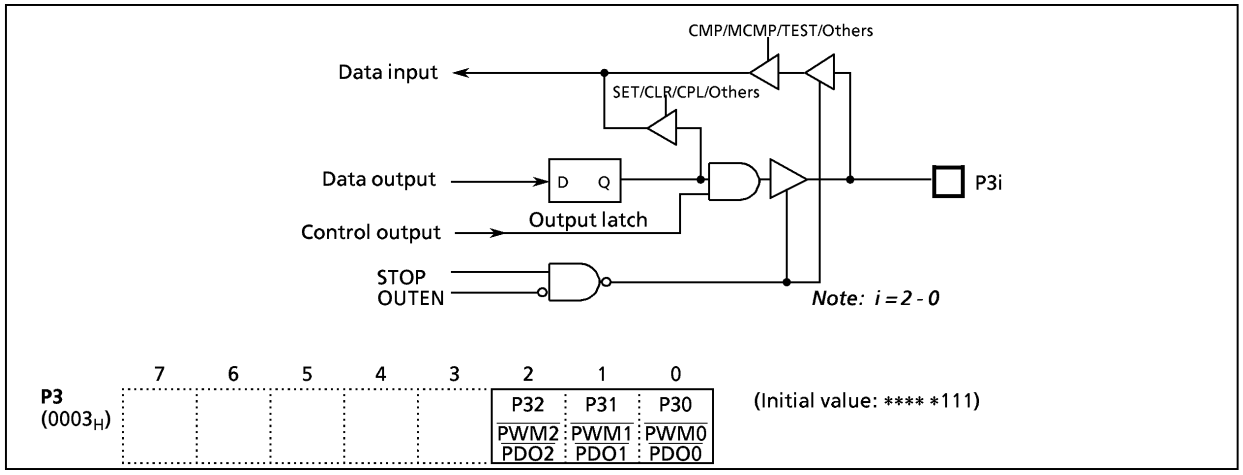


Figure 2-5. Port P3

2.2.4 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port, and it is also used as a timer/counter input/output and RxD/TxD pin for UART.

Output circuit is specfied by the corresponding bit in the port P4 push-pull control register (P4CR) . During reset, all bits of P4CR are initialized to "0", which configures port P4 as Nch open-drain output. The output latches are initialized to "1".

When used as an input or a secondary function pin, the output latch should be set to "1".

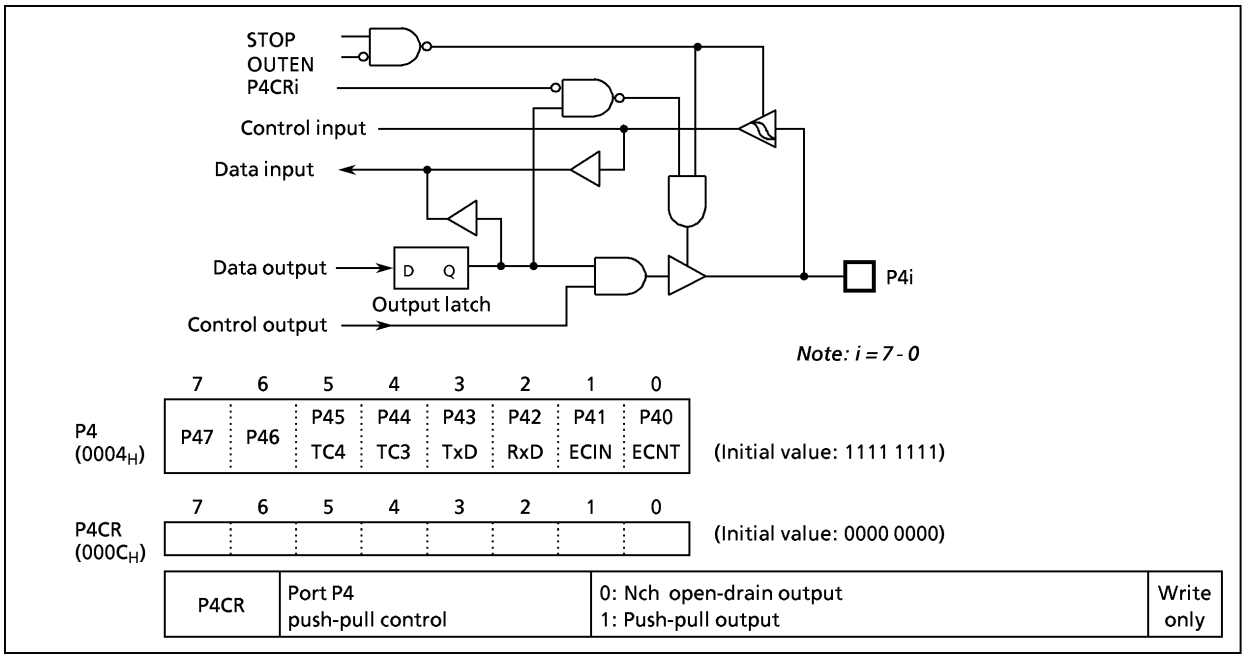
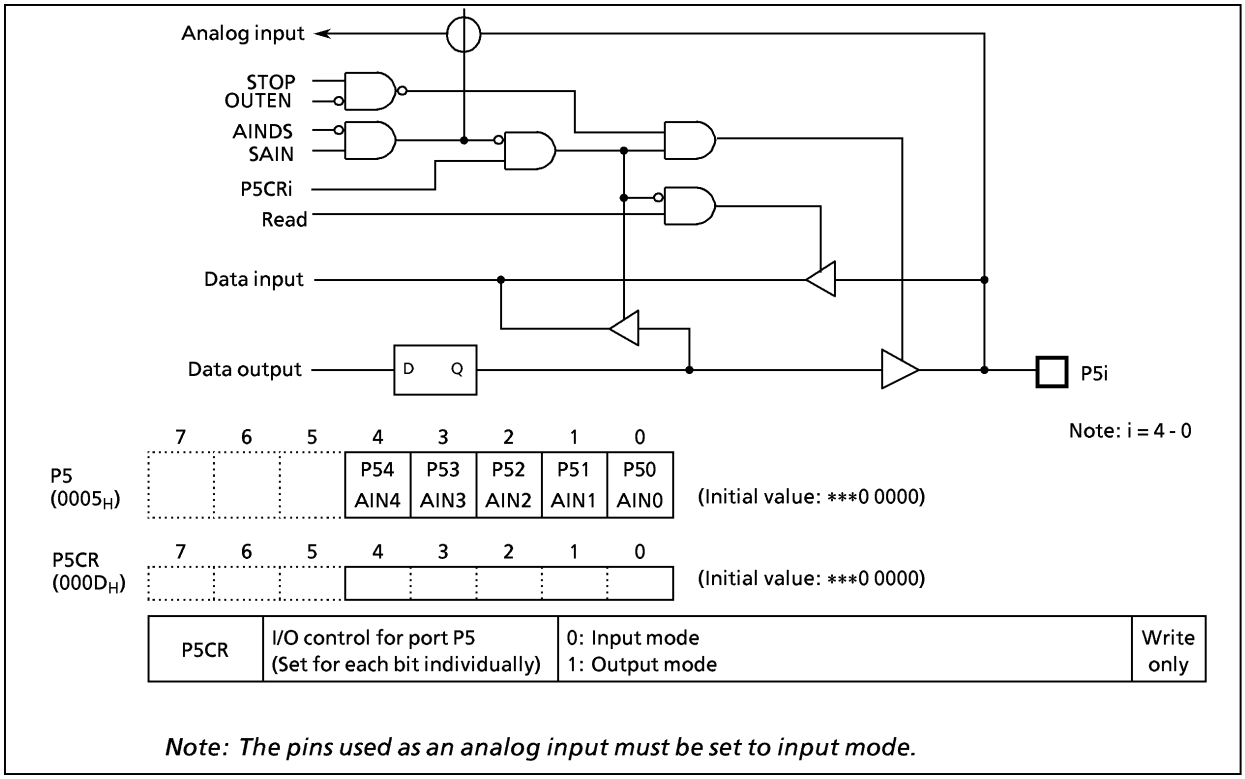


Figure 2-6. Port P4 and P4CR

2.2.5 Port P5 (P54 to P50)

Port P5 is a 5-bit input/output port. It is also used as an analog input.

Input or output is specified using the port P5 I/O control registe (P5CR) and AINDS (bit 4 in ADCCR) . At reset, P5CR is set to 0 and AINDS is cleared to 0. Thus, P5 becomes an analog input port. At the same time, the output latch of port P5 is initialized to 0. P5CR is a write-only register. Pins not used for analog input can be used as I/O ports. But do not execute the output instruction to keep the accuracy in AD conversion. Executing an input inistruction on port P5 when the AD converter is in use reads 0 at pins set for analog input ; 1 or 0 at pins not set for analog input, depending on the pin input level.



Note: The pins used as an analog input must be set to input mode.

Figure 2-7. Port P5 and P5CR

Note 1: Ports set to the input mode read the pin states. When input pin and output pin exist 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.)

2.2.6 Ports P6 (P67 to P60) and P7 (P77 to P70)

Port P6 and P7 are an 8-bit input/output ports and are also used as segment outputs of LCD driver. Input/output mode or segment output mode is specified by the corresponding bit in the P6, P7 port control register (P6CR, P7CR). During reset, P6CR and P7CR are initialized to "0", which configure port P6, P7 as input/output. When used as an input port, the output latch is set to "1". Port P6, P7 output latches are also initialized to "1". P6CR and P7CR can only be written. The output latches are initiazed to "1" during reset.

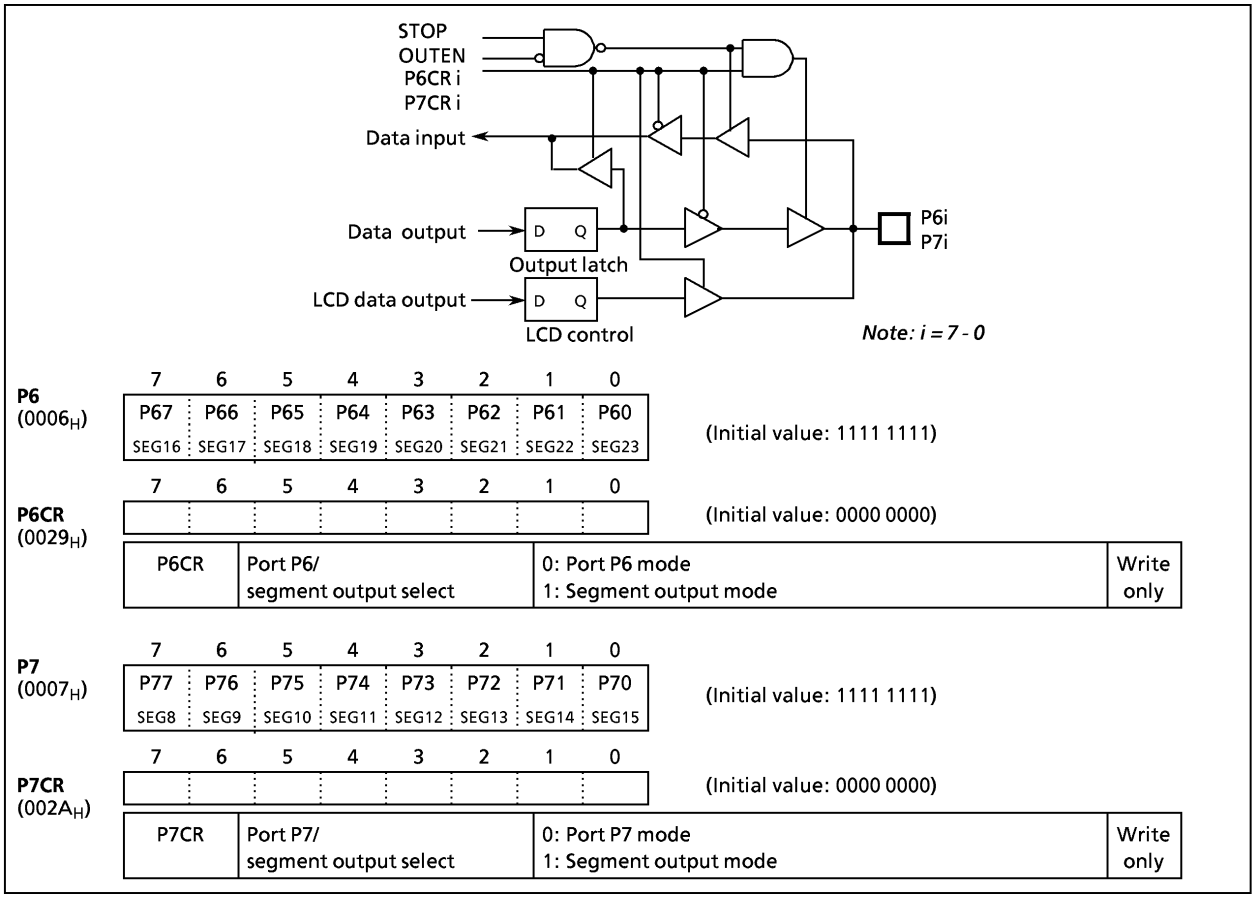


Figure 2-8. Port P6/P7 and P6CR/P7CR

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

```
LD    (P6CR), 11000000B
```

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 2-9 (b).)

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

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

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

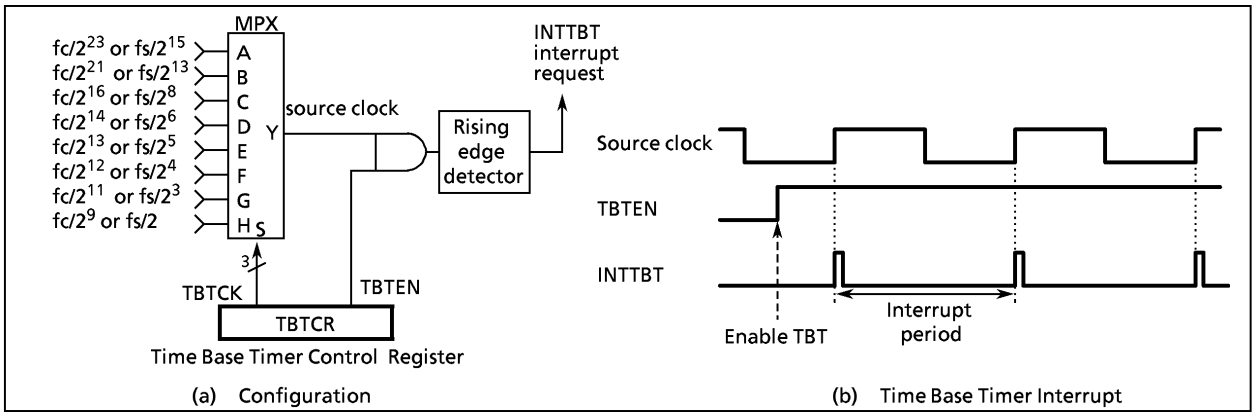


Figure 2-9. Time Base Timer

TBTCR (0036_H)

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

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

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

*Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care*
Note 2: TBTCR is a write-only register and must not be used with any of read-modify-write instructions.

Figure 2-10. Time Base Timer 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 2-11.

Note that the TBTCR is a write-only register, so a read-modify-write instruction cannot be used.

TBTCR (0036_H)

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

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

DVOEN	Divider output enable/disable	0: Disable 1: Enable	Write only
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 2-11. 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

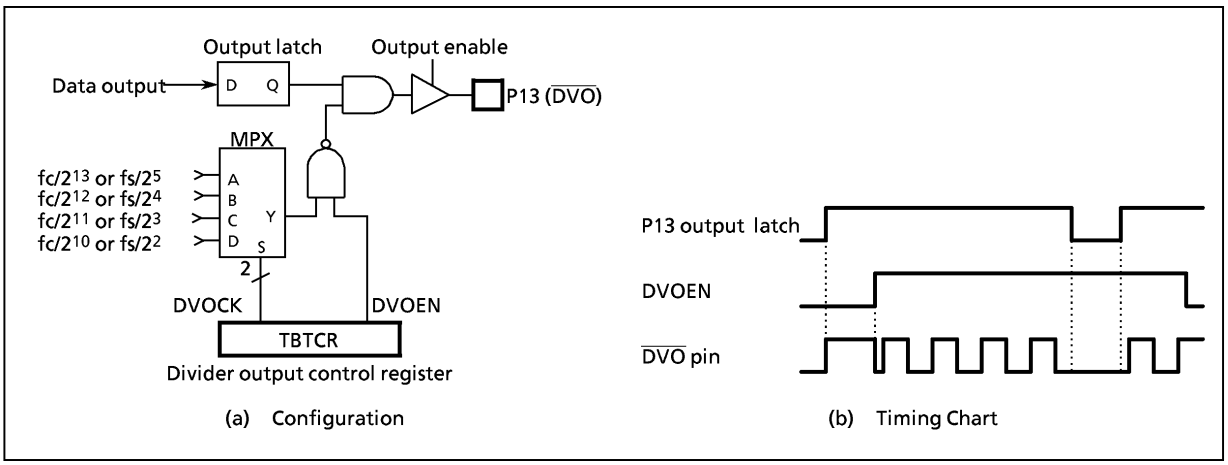


Figure 2-12. Divider Output

2.5 18-Bit Timer/Counter (TC1)

2.5.1 Configuration

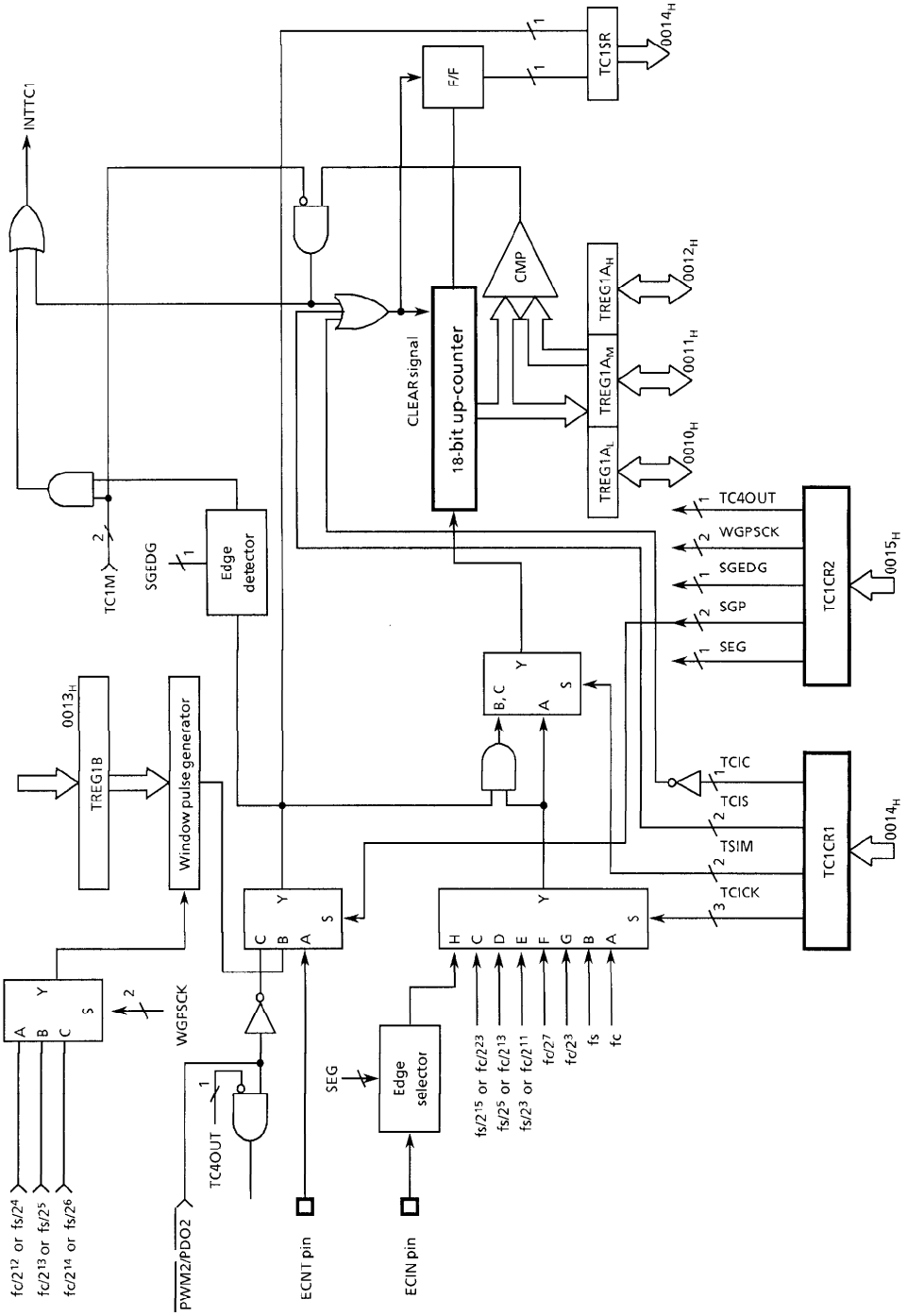


Figure 2-13. Timer/Counter 1

2.5.2 Control

The Timer/counter 1 is controlled by a timer/counter 1 control registers (TC1CR1/TC1CR2), an 18-bit timer register (TREG1A), and an 8-bit internal window gate pulse setting register (TREG1B).

Timer register

TREG1A (0010_H, 0011_H, 0012_H,)

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$TREG1A_H$ (0012_H)		$TREG1A_M$ (0011_H)								$TREG1A_L$ (0010_H)							

Read/Write

Internal window gate pulse setting register

TREG1B (0013_H)

7	6	5	4	3	2	1	0
Tb				Ta			

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

Ta	Setting "H" level period of the window gate pulse	$(16 - Ta) \times 2^{13}/fc$ or $(16 - Ta) \times 2^5/fs$ [s]	Write only
Tb	Setting "L" level period of the window gate pulse	$(16 - Tb) \times 2^{13}/fc$ or $(16 - Tb) \times 2^5/fs$ [s]	

Note : WGPSCK = 01

TC1CR1 (0014_H)

7	6	5	4	3	2	1	0
TC1C	TC1S			TC1CK		TC1M	

(Initial value: 1000 0000)

TC1M	TC1 mode select	00: Timer/event counter mode 01: reserved 10: Pulse width measurement mode 11: Frequency measurement mode	Write only
TC1CK	TC1 source clock select	000: Internal clock fc [Hz] 001: Internal clock fs 010: Internal clock $fc/2^{23}$ or $fs/2^{15}$ 011: Internal clock $fc/2^{13}$ or $fs/2^5$ 100: Internal clock $fc/2^{11}$ or $fs/2^3$ 101: Internal clock $fc/2^7$ 110: Internal clock $fc/2^3$ 111: External clock (ECIN pin input)	
TC1S	TC1 start control	00: Stop & counter clear 10: Start *1: Reserved	
TC1C		0: Counter/overflow flag clear request ("1" is automatically set after clearing.)	

TC1CR2 (0015_H)

7	6	5	4	3	2	1	0
SEG	SGP		SGEDG	WGPSCK		TC4OUT	

(Initial value: 0000 000*)

SEG	External input clock (ECIN) edge select	0: Counts at the falling edge 1: Counts at the falling/rising edges	Write only
SGP	Window gate pulse select	00: ECNT input 01: Internal window gate pulse (TREG1B) 10: $\overline{PWM2}/PDO2$ (TC4) output 11: Reserved	
SGEDG	Window gate pulse interrupt edge select	0: Interrupts at the falling edge 1: Interrupts at the falling/rising edges	
WGPSCK	Window gate pulse source clock select	00: Select $2^{12}/fc$ or $2^4/fs$ 01: Select $2^{13}/fc$ or $2^5/fs$ 10: Select $2^{14}/fc$ or $2^6/fs$ 11: Reserved	
TC4OUT	TC4 output (PWM2/PDO2) external output select	0: Output to P32 1: No output to P32	

TC1 Status register

TC1SR (0014_H)

7	6	5	4	3	2	1	0
HECF	HEOVF	"1"	"1"	"1"	"1"	"1"	"1"

HECF	Operating Status monitor	0: Stop (during Tb) or disable 1: Under counting (during Ta)	Read only
HEOVF	Counter overflow monitor	0: No overflow 1: Overflow status	

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

Note 2: Writing to the low-byte of the timer register 1A ($TREG1A_L$, $TREG1A_M$), the compare function is inhibited until the high-byte ($TREG1A_H$) is written.

Note 3: Set the mode, source clock, and edge (selection) when the TC1 stops (TC1S = 00).

Note 4: "fc" can be selected as the source clock only in the timer mode during SLOW mode and in the pulse width measurement mode during NORMAL 1/2 or IDLE 1/2 mode.

Note 5: TC1CR1, TC1CR2 and TREG1B are write-only registers and must not be used with any of the read-modify-write is instructions.

Figure 2-14. Timer Register/Window Gate Pulse Setting Register/Control Register of the TC1

2.5.3 Function

TC1 has four operating modes. The timer mode of the TC1 is used at warm-up when switching form SLOW mode to NORMAL2 mode.

(1) **Timer** mode

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

Table 2-3. Source Clock (internal clock) of Timer/Counter 1

Source clock				Resolution		Maximum time setting	
NORMAL1/2, IDLE1/2 mode		SLOW mode	SLEEP mode	fc = 8 MHz	fs = 32.768 kHz	fc = 8 MHz	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	76.5 h	72.8 h
$fc/2^{13}$	$fs/2^5$	$fs/2^5$	$fs/2^5$	1.02 ms	0.98 ms	4.5 min	4.3 min
$fc/2^{11}$	$fs/2^3$	$fs/2^3$	$fs/2^3$	256 μs	244 μs	1.1 min	1.07 min
$fc/2^7$	$fc/2^7$	—	—	16 μs	—	4.2 s	—
$fc/2^3$	$fc/2^3$	—	—	1 μs	—	262 ms	—
fc	fc	fc (Note)	—	125 ns	—	32.5 ms	—
fs	fs	—	—	—	30.5 μs	—	8 s

Note: "fc" can be used only in the timer mode (SLOW mode) and the pulse width measurement mode (NORMAL 1/2, IDLE 1/2 modes). When fc is selected for the source clock in SLOW mode, the lower bits 11 of TREG1A is invalid, and a match of the upper bits 7 makes interrupts.

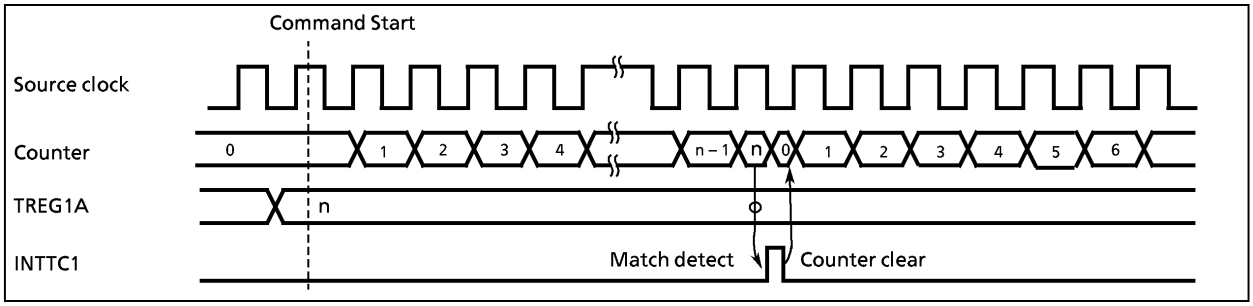


Figure 2-15. Timing Chart for Timer Mode

(2) **Event Counter** mode

It is a mode to count up at the falling edge of the ECIN pin input. Both edges can not be used. The countents 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 resums for ECIN pin input edge each after the counter is cleared. The maximum applied frequency is $fc/2^4$ [Hz] in NORMAL 1/2 or IDLE 1/2 mode and $fs/2^4$ [Hz] in SLOW or SLEEP mode.

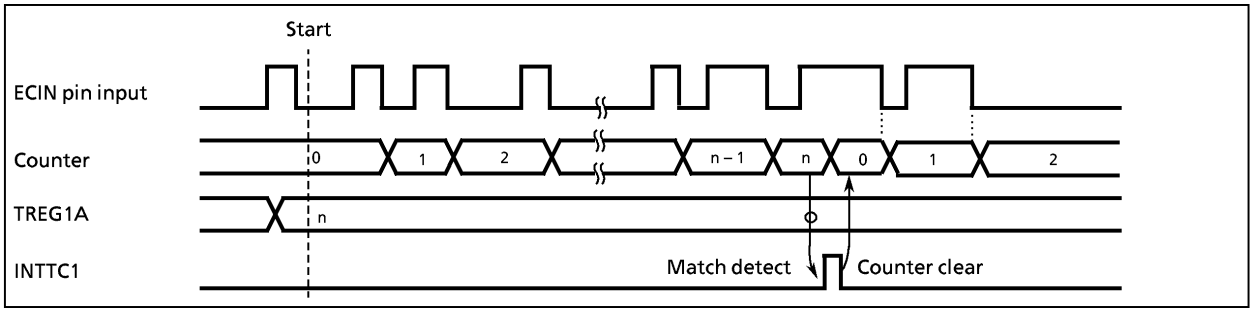


Figure 2-16. Event Counter Mode Timing Chart

(3) **Pulse Width Measurement** mode

In this mode, pulse widths are counted on the rising edge of logical AND-ed product between ECIN pin input (window pulse) and the internal clock. The internal clock is selected by TC1CK (bit 2, 3 and 4 in TC1CR1) . An INTTC1 interrupt is generated at the falling edge of the window pulse or both rising and falling edges of the window pulse, that can be selected by SGEDG (bit 4 in TC1CR2) . After reading out the contents of TREG1A by an interrupt service program, the counter is required to be cleared by TC1C (bit 7 in TC1CR1) . When the counter is not cleared, counting up resumes by starting count-up. The window pulse status can be monitored by HECF (bit 7 in TC1SR) of the status register. HEOVF (bit 6 in TC1SR) of the status register can monitor whether the binary counter overflows or not. HEOVF remains the old data until the counter is required to be cleared by TC1C.

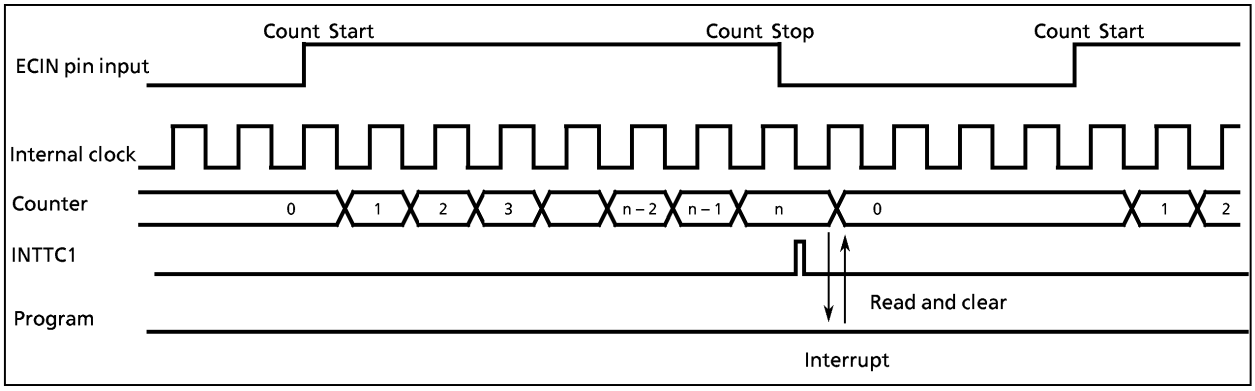
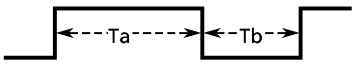


Figure 2-17. Pulse Width Measurement Mode Timing Chart

(4) **Frequency Measurement** mode

In This mode, the frequency of ECIN pin input pulse is measured. TC1CK is required to be set to the external clock. The edge of the input pulse is counted during "H" level of the window gate pulse selected by SGP (bit 5 and 6 in TC1CR2) . Whether the input pulse is counted on the falling edge or the both edges can be selected by SEG (bit 7 in TC1CR2). An INTTC1 interrupt is generated on the falling edge or both the rising/falling edges of the window gate pulse, that can be selected by SGEDG (bit 4 in TC1CR2). After reading out the contents of TREG1A by the interrupt service program, the counter is required be cleared by TC1C. When the counter is not cleared, counting up resumes by stating count-up. The window pulse status can be monitored by HECF of the status register. HEOVF of the status register can monitor whether the binary counter overflows or not. In the overflow flag status, a new data is not input until the counter clear requests.

- Using TC4 output ($\overline{PWM2}/PDO2$) for the window gate pulse can select whether $\overline{PWM2}/PDO2$ is output to the external port (P32) (initial output) in TC4OUT (bit 1 in TC1CR2).
- When the internal window gate pulse is selected, the window gate pulse is set as follows. The internal window gate pulse consists of "H" level period (Ta) that is counting time and "L" level period (Tb) that is counting stop time. Ta or Tb can be individually set by TREG1B. One cycle contains Ta + Tb.



The setting time fo Ta or Tb is shown as the following format. at (WGPSCK = 01).

$(16 - n) \times 2^{13}/fc[s]$ or

$(16 - n) \times 2^{5}/fs$ [s]

The setting time at fc = 4.194304 MHz is listed in a table menitioned right.

Tb set by the upper side (bits 7 to 4), and Ta is set by the lower side (bits 3 to 0) in TREG1B.

Figure 2-18. Window Gate Pulse Format

Tabale 2-4. Setting Ta and Tb

Setting value	Setting time	Setting value	Setting time
0	31.25 ms	8	15.63 ms
1	29.30 ms	9	13.67 ms
2	27.34 ms	A	11.72 ms
3	25.39 ms	B	9.77 ms
4	23.44 ms	C	7.81 ms
5	21.48 ms	D	5.86 ms
6	19.53 ms	E	3.91 ms
7	17.58 ms	F	1.95 ms

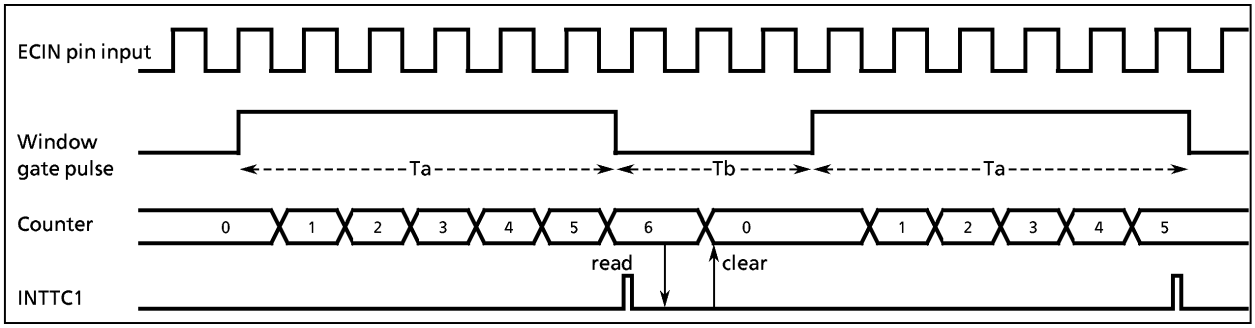


Figure 2-19. Timing Chart for the Frequency Measurement Mode
(ECIN falling edge count, window gate pulse falling interrupt)

2.6 8-Bit Timer/Counter 3 (TC3)

2.6.1 Configuration

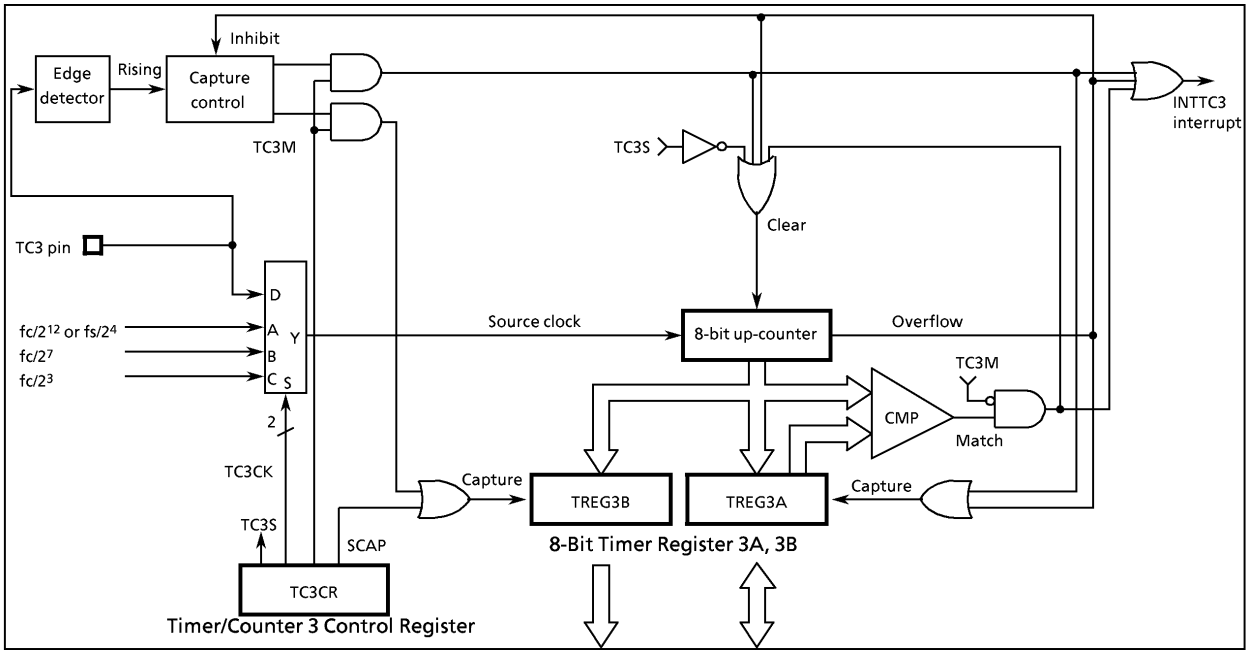


Figure 2-20. Timer/Counter 3

2.6.2 Control

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

TREG3A (0018_H) — Read/Write

TREG3B (0019_H) — Read only

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

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

TC3M	TC3 operating mode select	0: Timer/event counter 1: Capture	Write only
TC3CK	TC3 source clock select	00: Internal clock $fc/2^{12}$ or $fs/2^4$ [Hz] 01: Internal clock $fc/2^7$ 10: Internal clock $fc/2^3$ 11: External clock (TC3 pin input)	
TC3S	TC3 Start control	0: Stop and counter 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 operating mode, the source clock selection when the TC3 stops (TC3S = 0).
Note 3: Values to be loaded to timer register 3A must satisfy the following condition.
TREG3A > 0 (in the timer and event counter modes)
Note 4: The softcapture can be used only in timer/event counter mode.
Note 5: TC3CR is write-only register and must not be used with any of the read-modify-write instruction.

Figure 2-21. Timer Register 3 and TC3 Control Register

2.6.3 Function

The TC3 has three operating modes : timer, event counter, and capture mode.
Also TC3 is used as the transfer rate of UART. For more details on the transfer rate of UART, see section "2.10.4 Transfer Rate".

(1) **Timer** mode

In this mode, the internal clock shown in Table2-5 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 to "0". 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 to "0" after capturing.

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

Source clock			Resolution		Maximum setting time	
NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode	fc = 8MHz	At fs = 32.768 kHz	fc = 8MHz	At fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{12}$ [Hz]	$fs/2^4$ [Hz]	$fs/2^4$ [Hz]	512 μs	488.28 μs	131.1 ms	125 ms
$fc/2^7$	$fc/2^7$	–	16 μs	–	4.1 ms	–
$fc/2^3$	$fc/2^3$	–	1 μs	–	256 μs	–

Example: Software capture

```
LD    (TC3CR), 01011000B    ;    SCAP ← 1
LD    A, (TREG3B)           ;    Reads captured value
```

(2) **Event Counter** mode

It is a mode to count up at the rising edge of TC3 input (external clock) pulse. The contents of TREG3A are compared with the contents of up-conuter. 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 NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ [Hz] in SLOW or SLEEP mode. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.
The current contents of up-counter are loaded into TREG3B by setting SCAP (bit 6 in TC3CR) to "1". SCAP is automatically cleared after capturing.

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

```
LD    (TREG3A), 19H         ;    0.5 s ÷ 1/50 = 25 = 19H
LD    (TC3CR), 00011100B    ;    Start TC3
```

(3) **Capture** mode

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

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

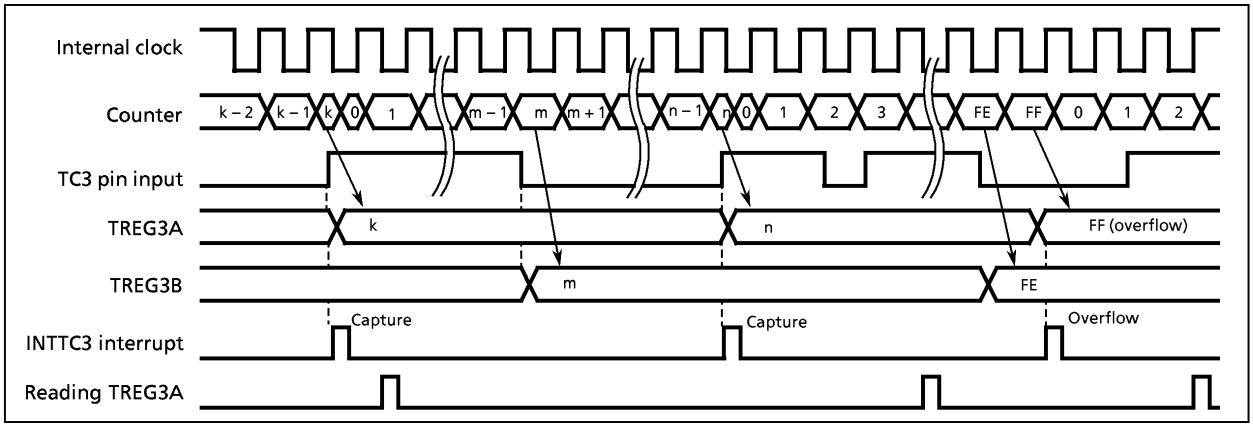


Figure 2-22. Capture Mode Timing Chart

2.7 8-Bit Timer/Counter (TC4)

2.7.1 Configuration

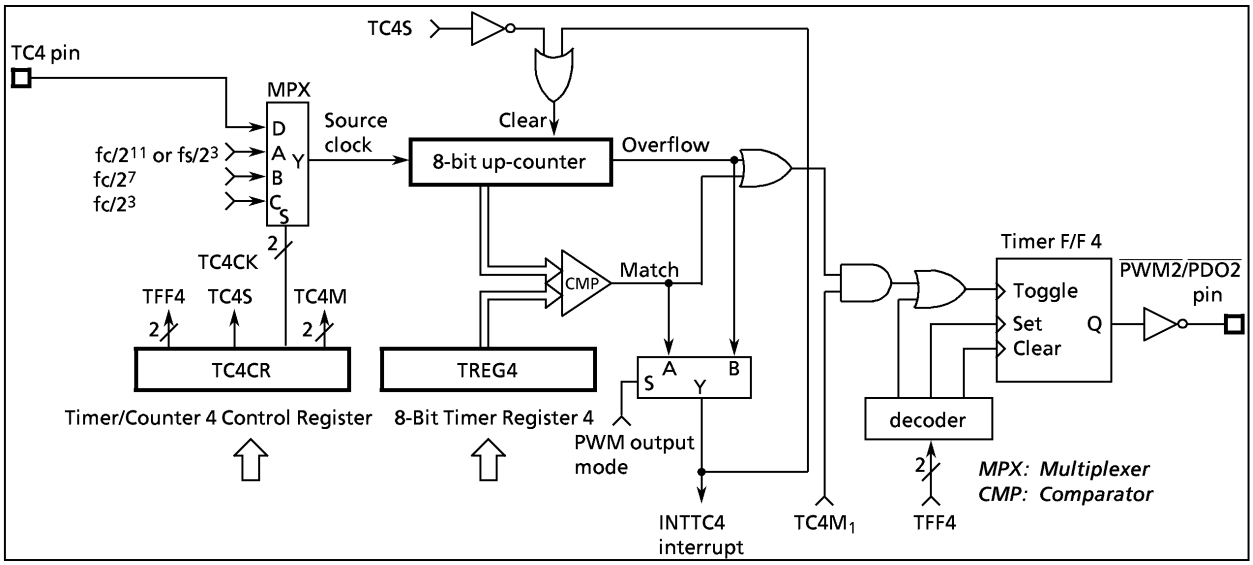


Figure 2-23. Timer/Counter 4

2.7.2 Control

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

TREG4 ($001B_H$)

7	6	5	4	3	2	1	0

Write only

TC4CR ($001C_H$)

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

(Initial value: 00*0 0000)

TC4M	TC4 operating mode select	00: Timer/event counter mode 01: Reserved 10: Programable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode	Write only
TC4CK	TC4 source clock select	00: Internal clock $fc/2^{11}$ or $fs/2^3$ [Hz] 01: Internal clock $fc/2^7$ 10: Internal clock $fc/2^3$ 11: External clock (TC4 pin input)	
TC4S	TC4 start control	0: Stop and counter clear 1: Start	
TFF4	Timer F/F4 control	00: Clear 01: Toggle 10: Set 11: –	

*Note 1 : fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care*
Note 2 : Set the operating mode, the source clock selection when the TC4 stops (TC4S = 0).
Note 3 : TFF4 must be set to "11" in the timer and event counter modes.
Note 4 : Values to be loaded to the timer register must satisfy the following condition.
TREG4 > 0
Note 5 : TC4CR and TREG4 are write-only registers and must not be used with any of the read-modify-write instructions.
Note 6 : "0" is always written at bit 5 of TC4CR.

Figure 2-24. Timer Register 4 and TC4 Control Register

2.7.3 Function

The TC4 has four operating modes: timer, event counter, programmable divider output, and pulse width modulation output mode. Also programmable divider output and pulse width modulation output mode are used as the window gate pulse of TC1. For more details on the window gate pulse of TC1, see section "2.5.3 Function (4) Frequency Measurement Mode".

(1) **Timer** mode

In this mode, the internal clock is used for counting up. The contents of TREG4 are compared with the contents of up-counter. Matching with TREG4 generates a timer/counter 4 interrupt (INTTC4) and clears the counter. Counting up resumes after the up-counter is cleared.

Table 2-6. Source Clock (Internal Clock) for TC4

Source clock			Resolution		Maximum setting time	
NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode	fc = 8MHz	At fs = 32.768 kHz	fc = 8MHz	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	65.5 ms	62.5 ms
$fc/2^7$	$fc/2^7$	–	16 μs	–	4.1 ms	–
$fc/2^3$	$fc/2^3$	–	1 μs	–	256 μs	–

(2) **Event Counter** mode

It is a mode to count up at the rising edge of TC4 input (external clock) pulse. Matching with TREG4 generates an INTTC4 interrupt and clears the counter. Counting up resumes after the up-counter is cleared. The maximum applied frequency is $fc/2^4$ [Hz] in NORMAL1, 2 or IDLE1, 2 mode, and $fs/2^4$ [Hz] in SLOW or SLEEP mode. Two or more machine cycles are required for both the high and low levels.

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

The internal clock is used for counting up. The contents of the TREG4 are compared with the contents of the up-counter. Timer F/F4 output is toggled and the counter is cleared each time a match is found. Timer F/F4 output is inverted and output to the $\overline{PDO2}$ (P32) pin. This mode can be used for 50% duty pulse output. Timer F/F4 can be initialized by program, and is initialized to "0" during reset. An INTTC4 interrupt is generated each time the $\overline{PDO2}$ output is toggled.

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

```
SET     (P3).2              ;  P32 output latch ← 1
LD      (TC4CR),  00000110B ;  Initializes the TC4 mode, source clock and timer F/F 4
LD      (TREG4),  10H       ;  (1/1024 ÷ 2^7/fc) ÷ 2 = 10H
LD      (TC4CR),  00010110B ;  Starts TC4
```

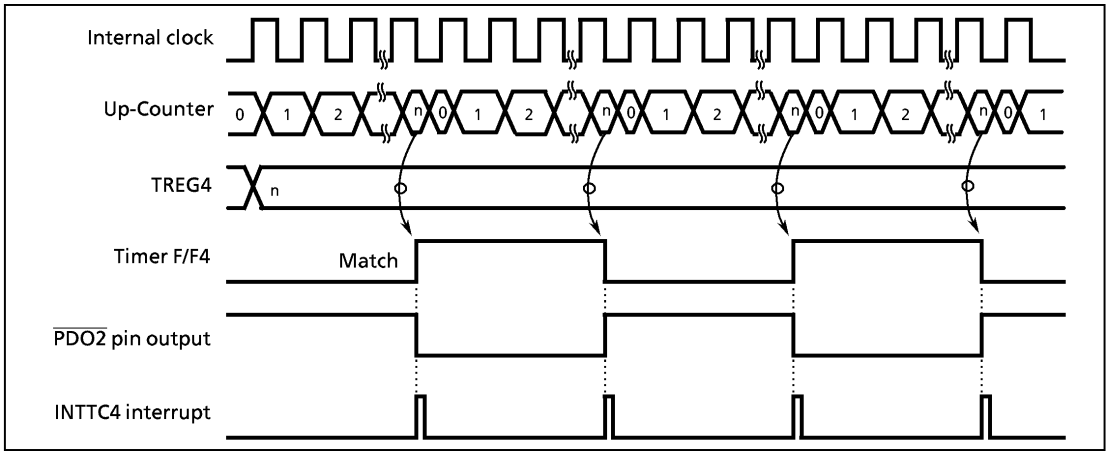


Figure 2-25. PDO Mode Timing Chart

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

PWM output with a resolution of 8 bits is possible. The internal clock is used for counting up. The contents of TREG4 is compared with the contents of up-counter. If a match is found, the timer F/F4 output is toggled. The counter continues counting and, when an overflow occurs, the timer F/F4 output is again toggled and the counter is cleared. Timer F/F4 output is inverted and output to the $\overline{\text{PWM2}}$ (P32) pin. An INTTC4 interrupt is generated when an overflow occurs.

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

Note 1: Do not overwrite TREG4 only when an INTTC4 interrupt is generated. Usually, TREG4 is overwritten in the routine of INTTC4 interrupt service.
Note 2: PWM output mode can be used only in the NORMAL 1, 2 and IDLE 1, 2 mode.

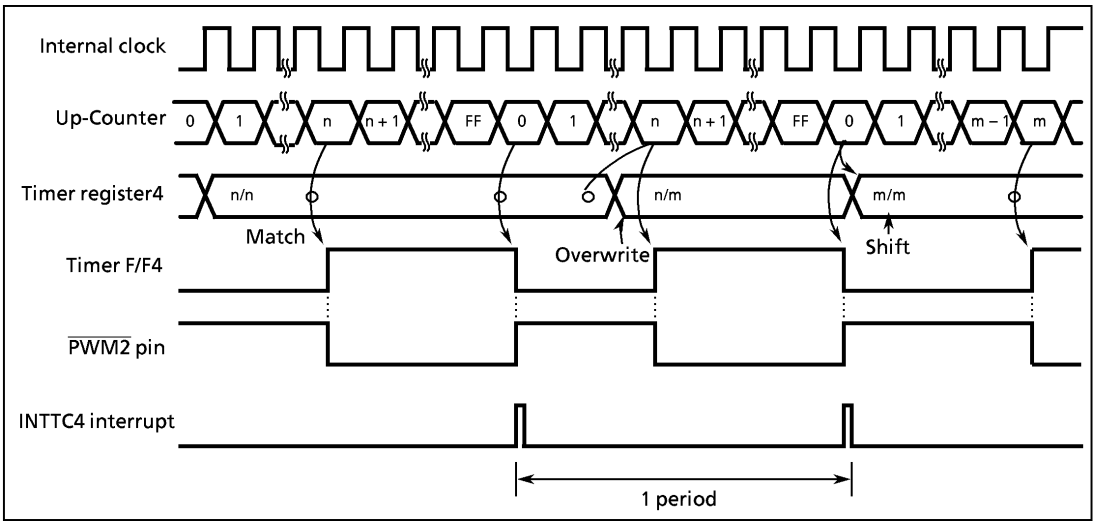


Figure 2-26. Timing Chart for PWM Mode

Table 2-7. PWM Output Mode

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

2.8 8-Bit Timer/Counter 5 (TC5)

2.8.1 Configuration

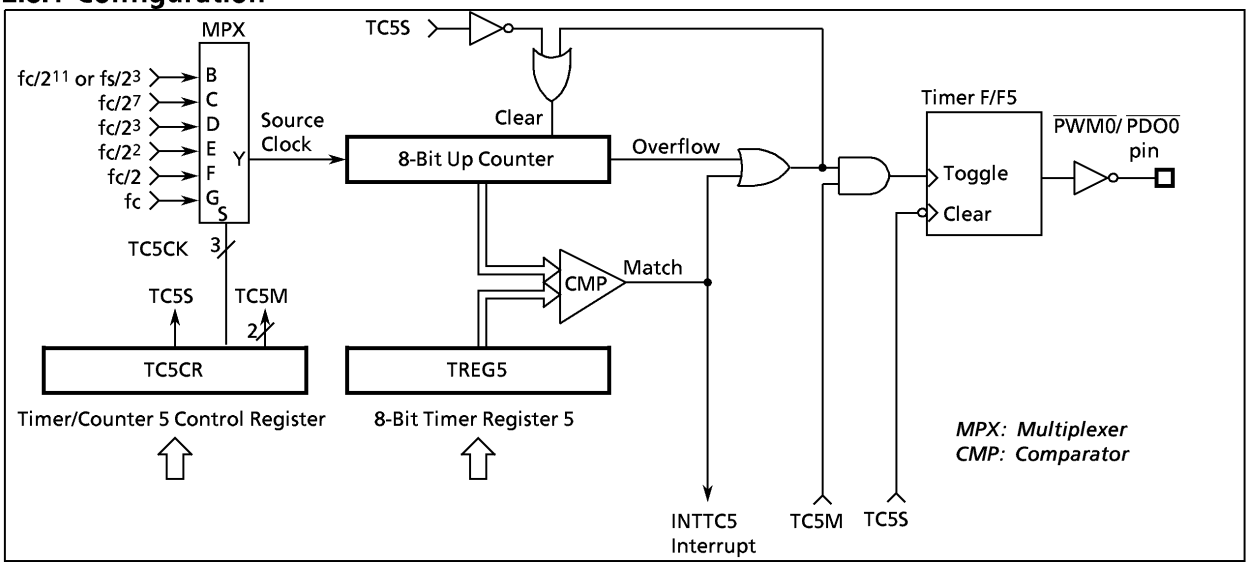


Figure 2-27. Timer/Counter 5 (TC5)

2.8.2 Control

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

TREG5 (001DH)

7	6	5	4	3	2	1	0

Write only

TC5CR (001EH)

7	6	5	4	3	2	1	0
"0"	"0"	TC5S	TC5CK			TC5M	

(Initial: **00 0000)

TC5M	TC5 Operating mode select	00: Timer mode 01: Reserved 10: Progrmmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode	Write only
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	
TC5S	TC5 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: 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 2-28. 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 2-8. 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	65.5 ms	62.5 ms
$fc/2^7$	$fc/2^7$	–	16 μs	–	4.1 ms	–
$fc/2^3$	$fc/2^3$	–	1 μs	–	256 μs	–

(2) **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 $\overline{PDO0}$ (P30) pin. This mode can be used for 50% duty pulse output. INTTC5 interrupt is generated each time the $\overline{PDO0}$ output is toggled.

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

```
LD      (TC5CR), 00001010B   ;  Sets to PDO modes and source clock (TC5M = 10, TC5CK = 010)
SET     (P3).0               ;  P30 output latch ← 1
LD      (TREG5), 10H         ;  1/2048 ÷ 2^7/fc = 10H
LD      (TC5CR), 00101010B   ;  Starts PDO1
```

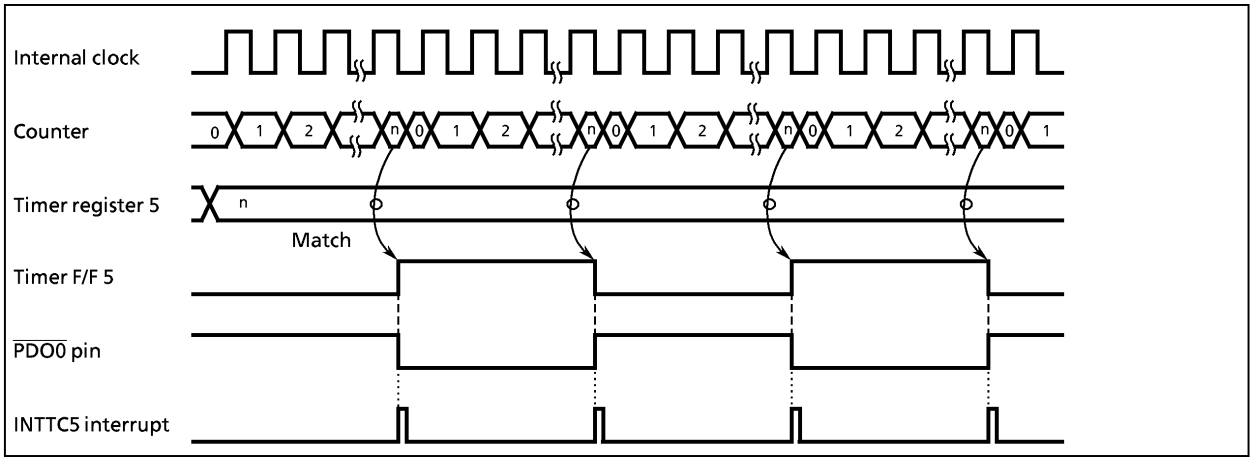


Figure 2-29. PDO Mode Timing Chart

(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 is again toggled and the counter is cleared. The timer F/F5 output is inverted and output to the $\overline{\text{PWM}}$ (P30) pin. 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: PWM output mode can be used only in NORMAL 1, 2 and IDLE 1, 2 modes.

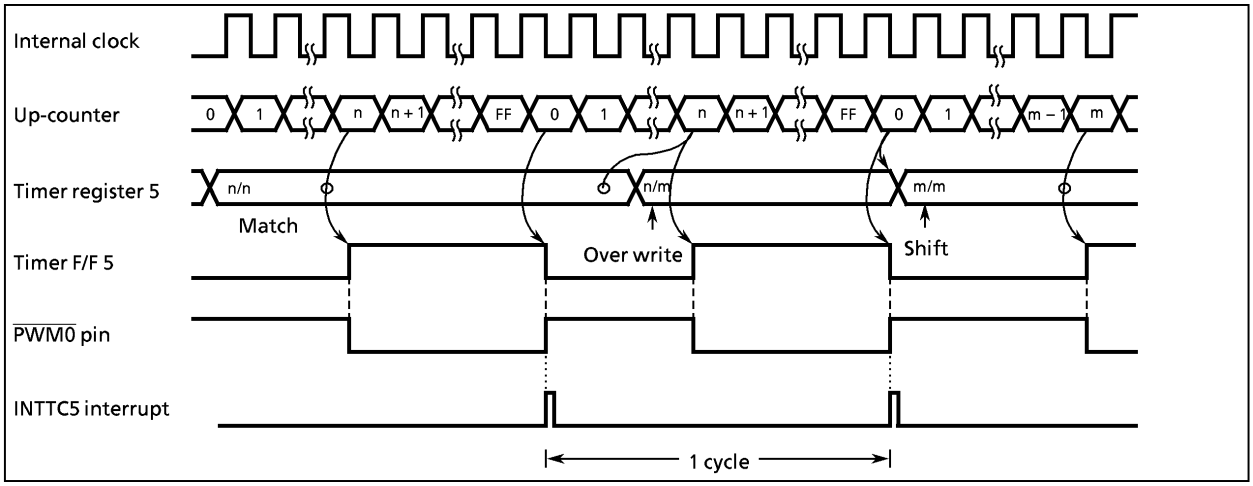


Figure 2-30. PWM Output Mode Timing Chart

Table 2-9. PWM Output Mode

Source clock		Resolution		Repeat cycle	
NORMAL1/2, IDLE1/2 mode		At fc = 8 MHz	At fc = 4.194304 MHz	At fc = 8 MHz	At fc = 4.194304 MHz
DV7CK = 0	DV7CK = 1				
$fc/2^2$ [Hz]		500 ns	953.7 ns	128 μs	244.14 μs
fc/2		250 ns	476.8 ns	64 μs	122.07 μs
fc		125 ns	238.4 ns	32 μs	61.04 μs

2.9 8-Bit Timer/Counter 6 (TC6)

Timer/counter 6 (TC6) can use only a PWM1/PDO1 mode.

2.9.1 Configuration

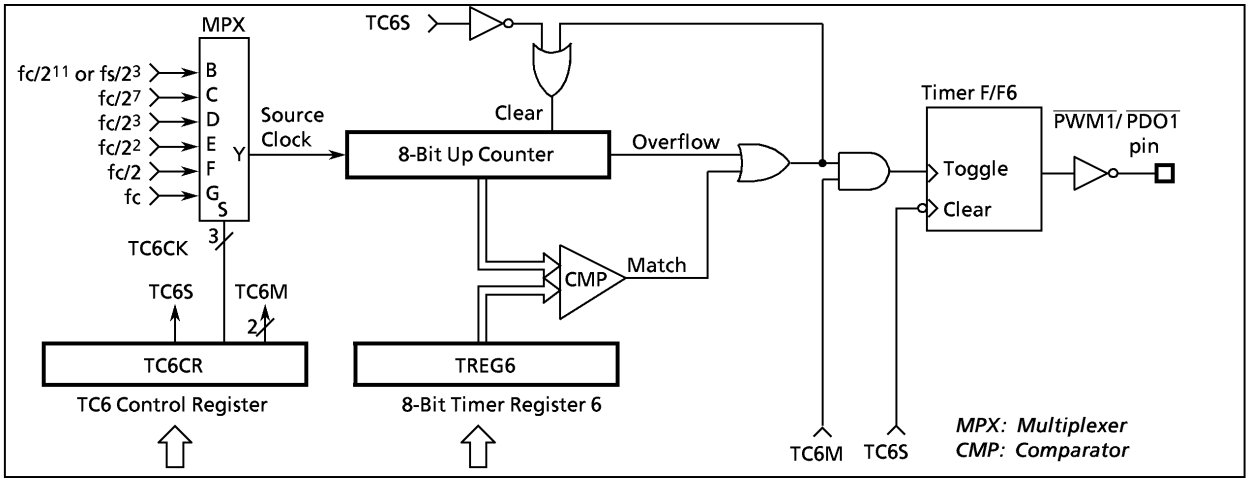


Figure 2-31. PWM1/PDO1

2.9.2 Control

The TC6 is controlled by a timer/counter6 control register (TC6CR) and an 8-bit timer register (TREG6).

TREG6 (0016H)

7	6	5	4	3	2	1	0

Write only

TC6CR (0017H)

7	6	5	4	3	2	1	0
"0"	"0"	TC6S		TC6CK		TC6M	

(Initial: **00 0000)

TC6M	PWM1/PDO1 Operating mode select	00: Reserved 01: Reserved 10: Progrmmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode	Write only
TC6CK	PWM1/PDO1 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	
TC6S	PWM1/PDO1 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: The set value of timer register must satisfy the following conditions.

(a) When in PWM output mode, 5<TREG6<251

(b) When in any other mode than PWM output mode, 0<TREG6

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 (TC6S = 0).

Figure 2-32. PWM1/PDO1 Control Register

2.9.3 Function

(1) **Programmable divider output** (PDO) mode

The internal clock is used for counting up. The contents of the TREG6 are compared with the contents of the up-counter. The timer F/F6 output is toggled and the counter is cleared each time a match is found. The timer F/F6 output is inverted and output to the $\overline{\text{PDO1}}$ (P31) pin. This mode can be used for 50% duty pulse output.

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

```
LD      (TC6CR), 00001010B   ;   Sets to PDO modes and source clock
SET     (P3).1               ;   P31 output latch ← 1
LD      (TREG6), 10H         ;   1/2048 ÷ 2^7/fc = 10H
LD      (TC6CR), 00101010B   ;   Starts PDO1
```

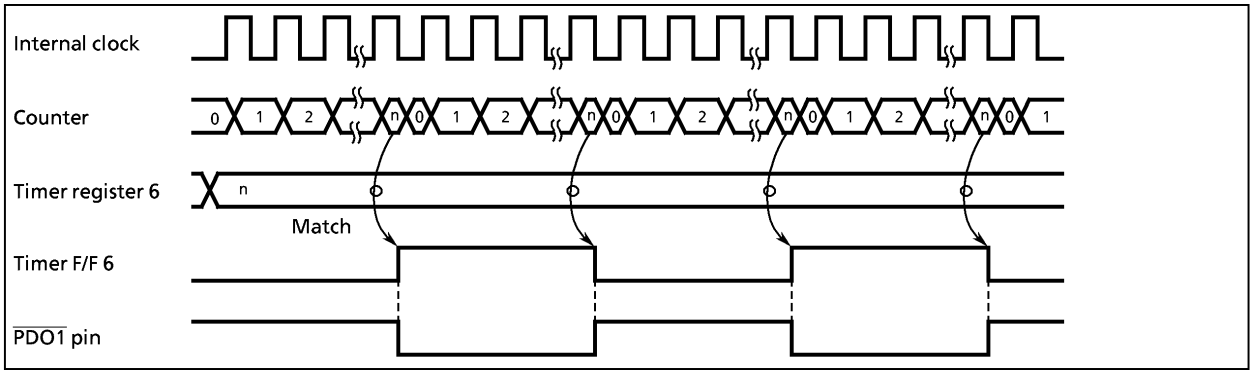


Figure 2-33. PDO Mode Timing Chart

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

PWM output with a resolution of 8-bit is possible. The internal clock is used for counting up. The contents of the TREG6 is compared with the contents of the up-counter. If a match is found, the timer F/F6 output is toggled. The counter continues counting and, when an overflow occurs, the timer is again toggled and the counter is cleared. The timer F/F6 output is inverted and output to the $\overline{\text{PWM}}$ (P31) pin.

TREG6 is configured a 2- stage shift register and, during output, will not switch until one output cycle is completed even if TREG6 is overwritten; therefore, output can be altered continuously. Also, the first time, TREG6 is shifted by setting TC6S (bit 5 in TC6CR) to "1" after data are loaded to TREG6.

Note: PWM output mode can be used only in NORMAL 1, 2 and IDLE 1, 2 modes.

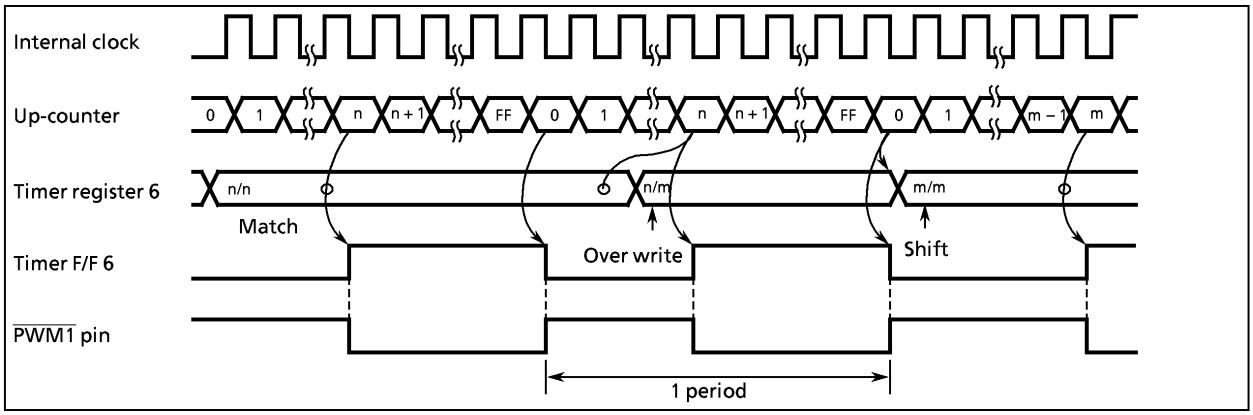


Figure 2-34. PWM Output Mode Timing Chart

Table 2-9. PWM Output Mode

Source clock		Resolution		Repeat cycle	
NORMAL1/2, IDLE1/2 mode		At fc = 8 MHz	At fc = 4.194304 MHz	At fc = 8 MHz	At fc = 4.194304 MHz
DV7CK = 0	DV7CK = 1				
$fc/2^2$ [Hz]		500 ns	953.7 ns	128 μs	244.14 μs
fc/2		250 ns	476.8 ns	64 μs	122.07 μs
fc		125 ns	238.4 ns	32 μs	61.04 μs

2.10 UART (Asynchronous serial interface)

The TMP87CH29/K29/M29 has 1 channel of UART (asynchronous serial interface) .

The UART is connected to external devices via RxD and TxD. RxD is also used as P42 ; TxD, as P43. To use P42 or P43 as the RxD or TxD pin, set P4 port output latches to 1.

2.10.1 Configuration

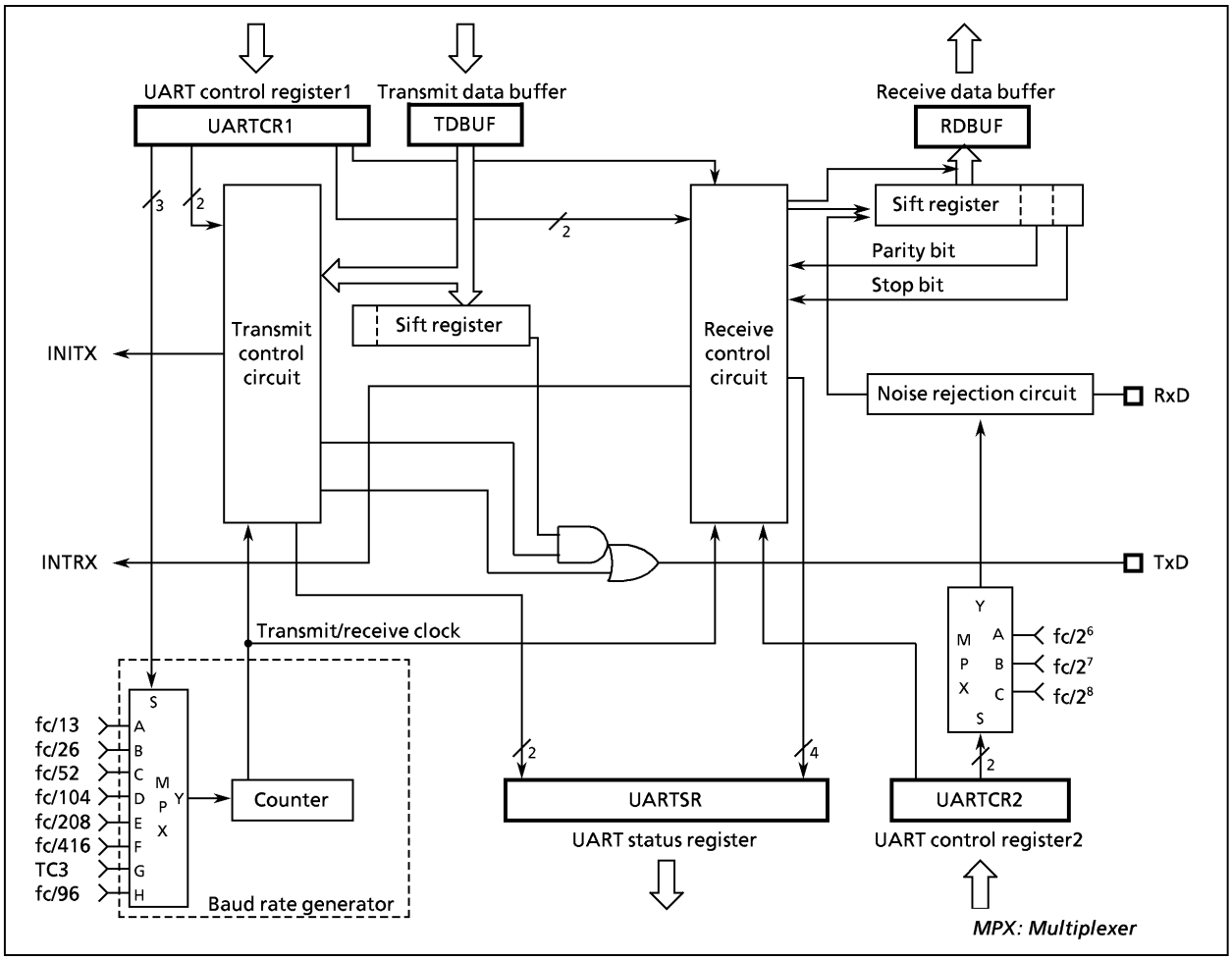


Figure 2-35. UART

2.10.2 Control

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

UART control register

UARTCR1 (0020_H)

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

(Initial value: 0000 0000)

BRG	Transmit clock select	000: fc/13 [Hz] 001: fc/26 010: fc/52 011: fc/104 100: fc/208 101: fc/416 110: TC3 111: f c/96	Write only
PE	Parity addition	0: No parity 1: Parity	
EVEN	Even-numbered parity	0: Odd-numbered parity 1: Even-numbered parity	
STBT	Transmit stop bit length	0: 1 bit 1: 2 bits	
RXE	Receive operation	0: Disable 1: Enable	
TXE	Transfer operation	0: Disable 1: Enable	

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

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

UARTCR2 (0021_H)

7	6	5	4	3	2	1	0
					RXDNC		STOPBR

(Initial value: **** *000)

STOPBR	Receive stop bit length	0: 1 bit 1: 2 bits	Write only
RxDNC	Selection of RxD input noise rejection time	00: No noise rejection (hysteresis input) 01: Rejects pulses shorter than 31/fc [s] as noise 10: Rejects pulses shorter than 63/fc [s] as noise 11: Rejects pulses shorter than 127/fc [s] as noise	

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

Figure 2-36. UART Control Register

UARTSR (0020_H)

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

(Initial value: 0000 11**)

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

UART receive data buffer (Initial value: 0000 0000) Read only

RDBUF ($0FEB_H$)

7	6	5	4	3	2	1	0

UART transmit data buffer (Initial value: 0000 000) Write only

TDBUF ($0FEB_H$)

7	6	5	4	3	2	1	0

Figure 2-37. UART Status Register and Data Buffer Registers

2.10.3 Transfer Data Format

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

Table 2-10.

PE	STBT	Frame length 1 2 3 ------- 8 9 10 11 12
0	0	Start Bit0 Bit1 ----- Bit6 Bit7 Stop1
0	1	Start Bit0 Bit1 ----- Bit6 Bit7 Stop1 Stop2
1	0	Start Bit0 Bit1 ----- Bit6 Bit7 Parity Stop1
1	1	Start Bit0 Bit1 ----- Bit6 Bit7 Parity Stop1 Stop2

2.10.4 Transfer Rate

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

Table 2-11.

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

When TC3 is used as the UART transfer rate (when BRG = 110), the transfer clock and transfer rate are detarmined as follows :

$$\text{Transfer clock} = \frac{\text{TC3 source clock}}{\text{TREG3 set value}}$$

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

2.10.5 Data Sampling

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

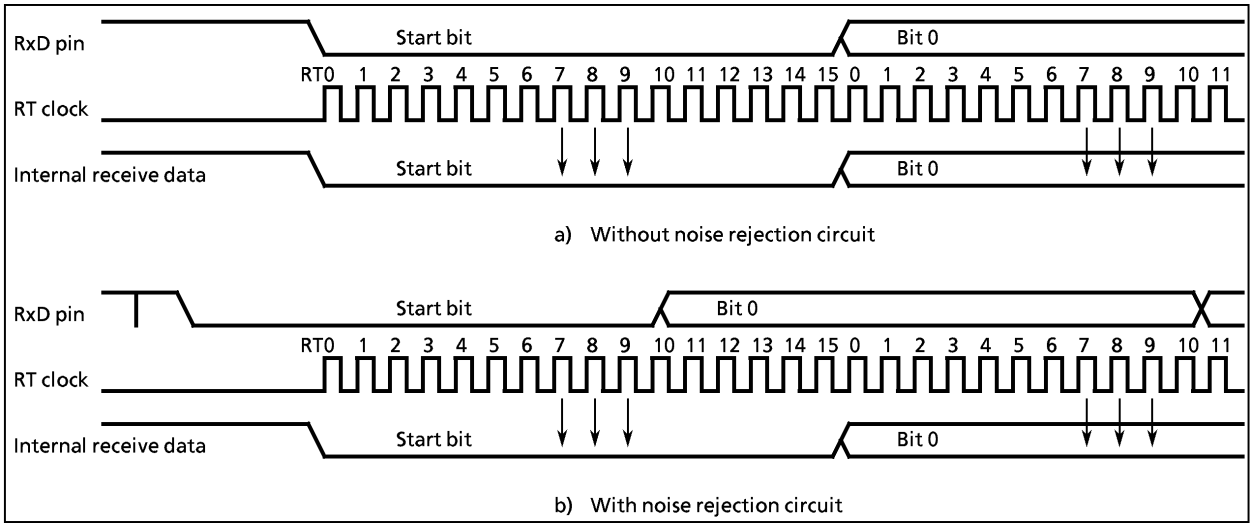


Figure 2-39. Data Samping

2.10.6 STOP Bit Length

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

2.10.7 Parity

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

2.10.8 Transmit/Receive

(1) Data transmit

Set TXE (bit 7 in UARTCR1) to 1. Read UARTSR to check TBEP = 1, then write data in TDBUF (transmit data buffer). Writing data in TDBUF zero-clears TBEP, transfers the data to the transmit shift register and the data are sequentially output from the TxD pin. The data output include a one-bit start bit, stop bits whose number is specified in STBT (bit 5 in UARTCR1) and a parity bit if parity addition is specified. Select the data transfer baud rate using bits 0 to 2 in UARTCR1. When data transmit starts, transmit buffer empty flag TBEP is set to 1 and an INTTX interrupt is generated.

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

(2) Data receive

Set RXE (bit 6 in UARTCR1) to 1. When data are received via the RxD pin, the receive data are transferred to RDBUF (receive data buffer). At this time, the data transmitted include a start bit and stop bit(s) and a parity bit if parity addition is specified. When stop bit(s) are received, data only are extracted and transferred to RDBUF (receive data buffer). Then the receive buffer full flag RBFL is set and an INTRX interrupt is generated. Select the data transfer baud rate using bits 0 to 2 in UARTCR1.

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

2.10.9 Status Flag/Interrupt Signal

(1) Parity error

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

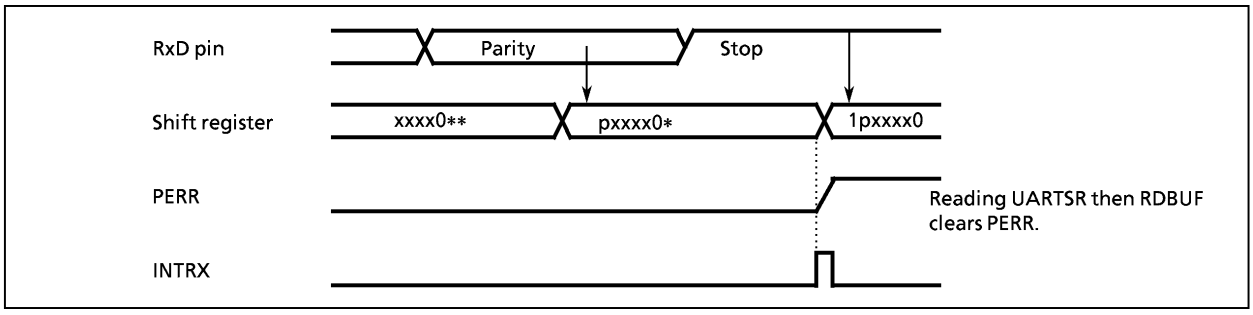


Figure 2-39. Generation of Parity Error

(2) Framing error

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

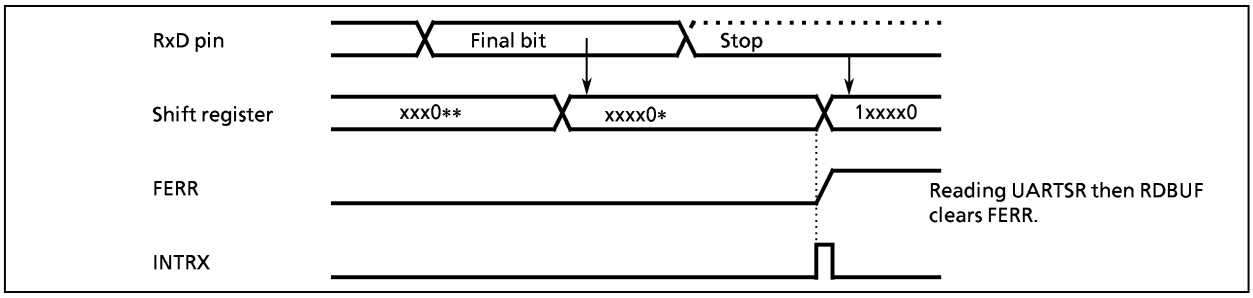


Figure 2-40. Generation of Framing Error

(3) Overrun error

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

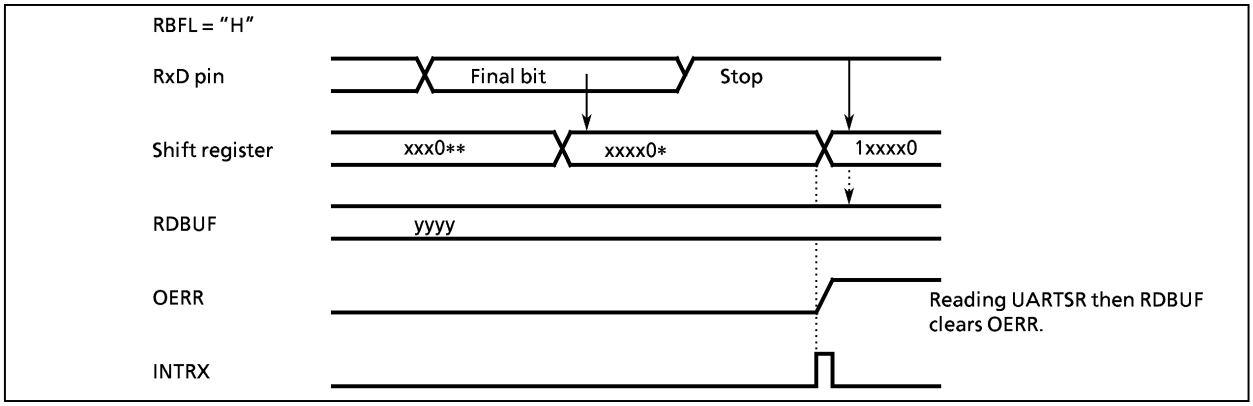


Figure 2-41. Generation of Overrun Error

(4) Receive data buffer full

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

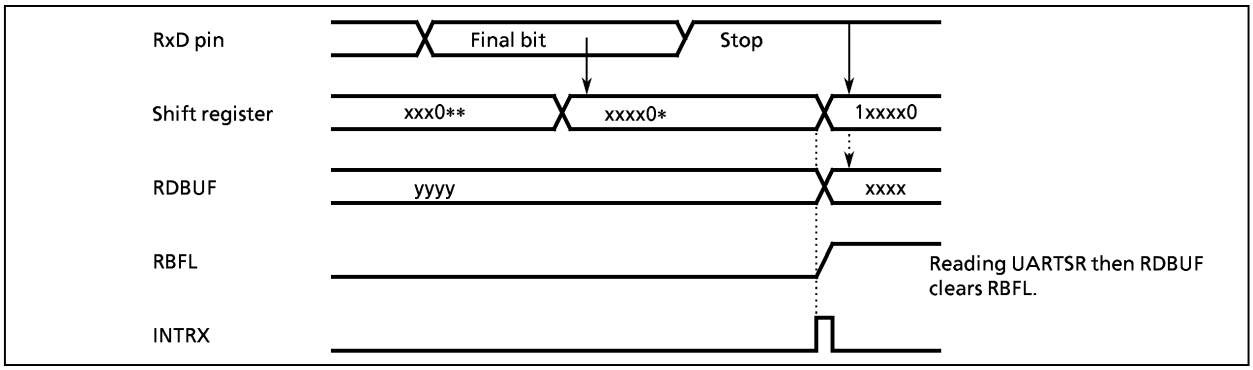


Figure 2-42. Generation of Receive Buffer Full

(5) Transmit data buffer empty

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

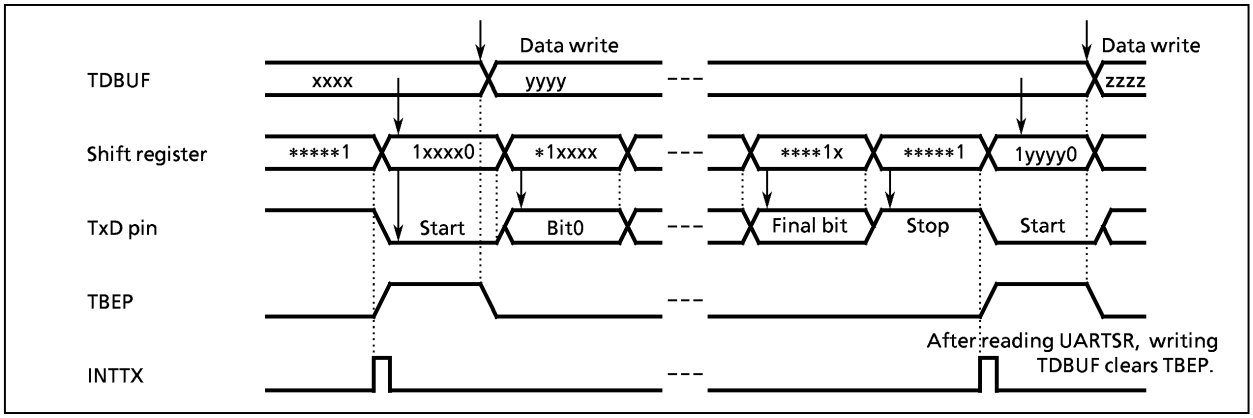


Figure 2-43. Generation of Transmit Buffer Empty

(6) Transmit end flag

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

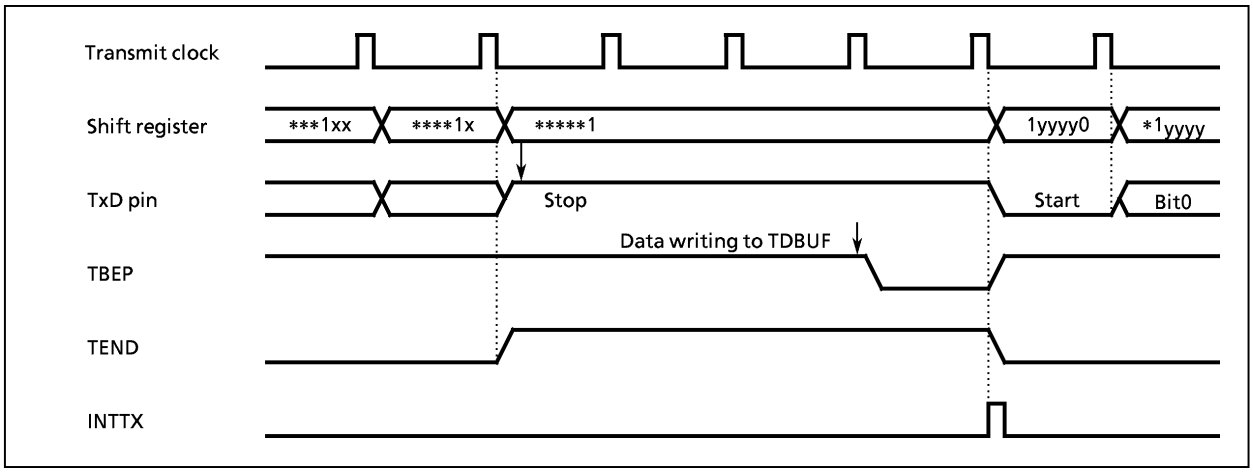


Figure 2-44. Generation of Transmit Buffer Empty

2.11 8-Bit AD Converter (ADC)

The TMP87CH29/K29/M29 each have a 5-channel multiplexed-input 8-bit successive approximate type AD converter with sample and hold.

2.11.1 Configuration

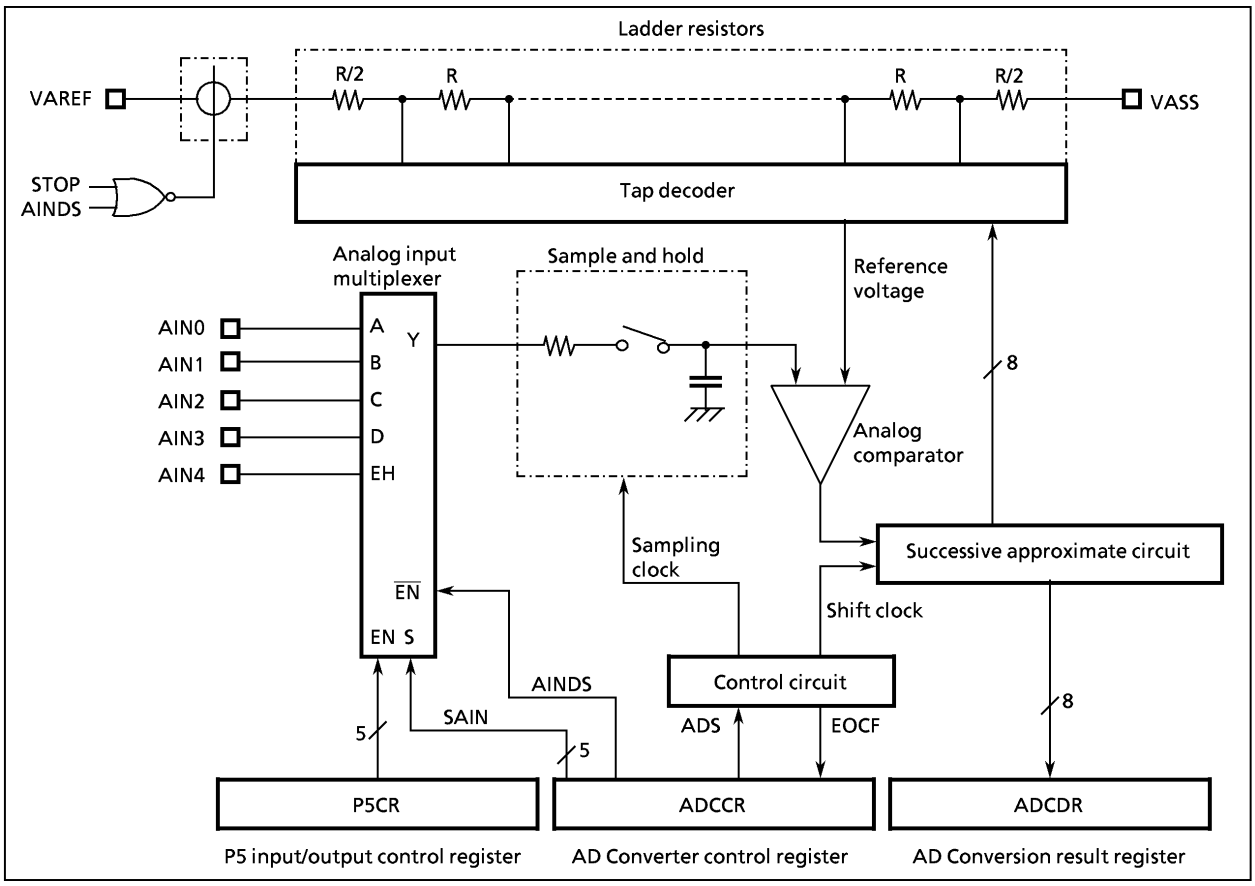


Figure 2-45. AD Converter

2.11.2 Control

The AD converter is controlled by an AD converter control register (ADCCR).

Reading EOCF of ADCCR acknowledges the operation state of AD converter, and reading AD conversion register (ADCDR) acknowledges AD conversion value.

AD Conversion Result Register

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

Figure 2-46. AD Conversion 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: Reserved 0110: Reserved 0111: Reserved 1***: Reserved	R/W
AINDS	Analog input control	0: Enable 1: Disable	
ACK	Conversion time select	0: 23 μs (at fc = 8 MHz) 1: 92 μs	
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	R

*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 2-47. AD Converter Control Register and AD Conversion Result Register

2.11.3 Operation

Applies the high side of analog reference voltage to VAREF pin and the low side to VASS pin. Splits the reference voltage between VAREF and VASS to a voltage corresponding to a bit by a ladder resistance. Comparing with an analog input executes AD conversion.

(1) Start of AD conversion

First, set the corressponding P5CR bit to "0" for input mode. Clear the AINDS (bit 4 in ADCCR) to "0" and select one of five analog input AIN4 to AIN0 with the SAIN (bits 3-0 in ADCCR).

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

AD conversion time is selected by setting the ACK (bits 5 in ADCCR).
AD conversion is started by setting the ADS (bit 6 in ADCCR) to "1".
When ACK = 0, conversion is accomplished in 46 machine cycles (184/fc [s]). When fc is 8 MHz, it needs 23 μs.
The EOCF (bit 7 in ADCCR) is set to "1" at end of conversion.
Setting ADS to "1" in AD conversion starts converting over again. An analog input voltage is sampled at intervals of four cycles after starting AD conversion.

> *Note: The sample hold circuit has a condenser of 12pF (typ.) via 5kΩ (typ.) resistor. It is necessary to charge to this condenser at intervals of four machine cycles.*

(2) Reading of AD conversion result

After the end of conversion (EOCF = 1), read the conversion result from the ADCDR.
The EOCF is automatically cleared to "0" when reading the ADCDR.
Undefined value is read in AD conversion.

(3) AD conversion in STOP mode

When the MCU places in the STOP mode during the AD conversion, the conversion is terminated and the ADCDR contents become indefinite. After returning from STOP mode, EOCF is kept to be cleared to "0".

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

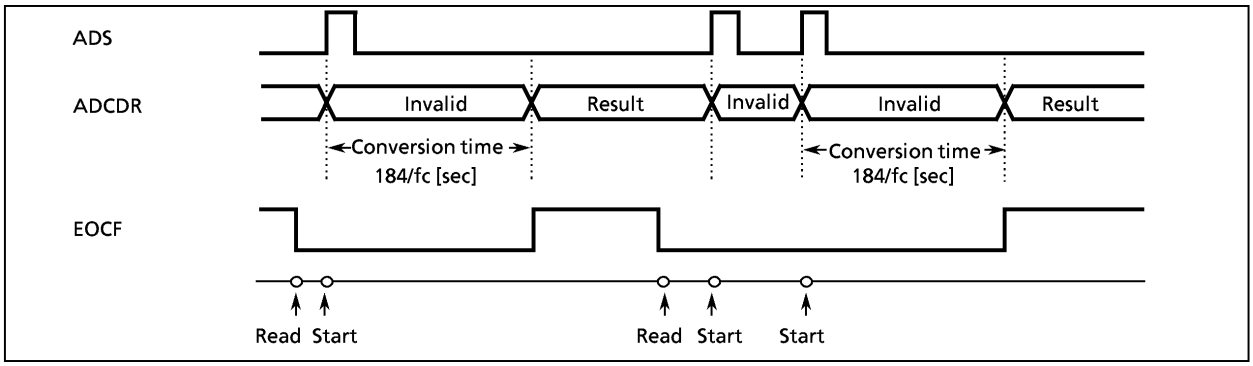


Figure 2-48. AD Conversion Timing Chart

Example 1:

```
        ; AIN SELECT
        LD      (ADCCR) , 00100100B     ;  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)
```

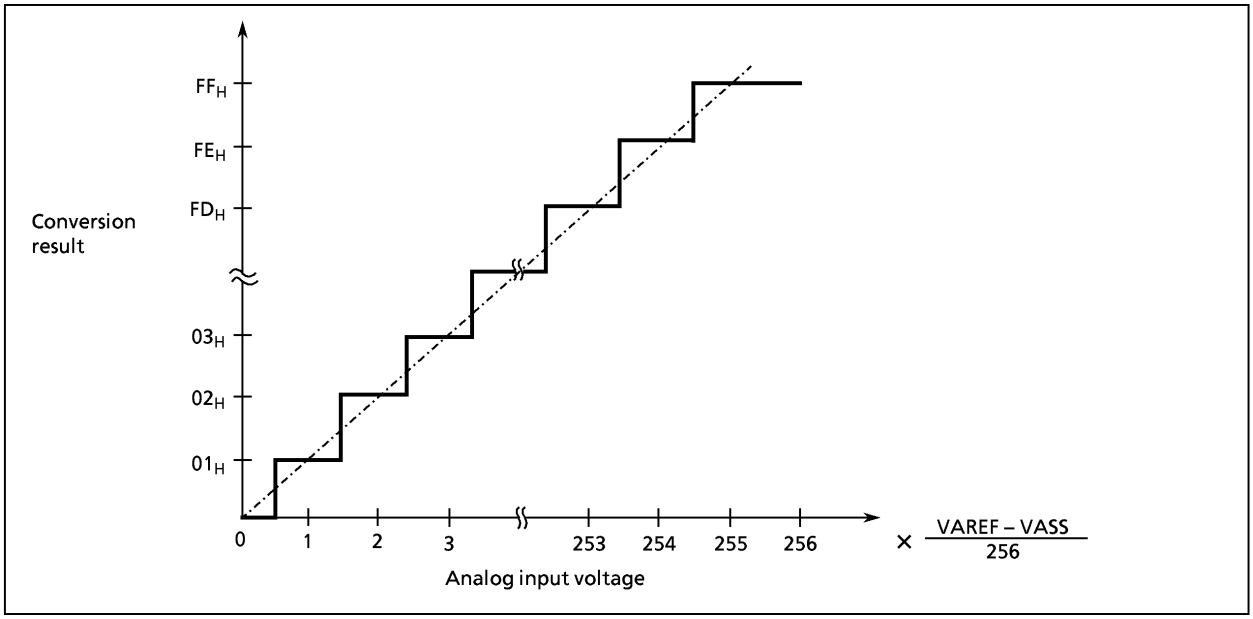


Figure2-49. Analog Input Voltage vs AD Conversion Result (typ.)

2.12 LCD Driver

The TMP87CH29/K29/M29 each have 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	8 pins	(SEG7 to SEG0)
② Segment output or P6, P7 input/output port	16 pins	(SEG23 to SEG8)
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 96 Segments (8-segment × 12 digits)
② 1/3 Duty	(1/3 Bias) LCD	Max 72 Segments (8-segment × 9 digits)
③ 1/3 Duty	(1/2 Bias) LCD	Max 72 Segments (8-segment × 9 digits)
④ 1/2 Duty	(1/2 Bias) LCD	Max 48 Segments (8-segment × 6 digits)
⑤ Static LCD	.....	Max 24 Segments (8-segment × 3 digits)

2.12.1 Configuration

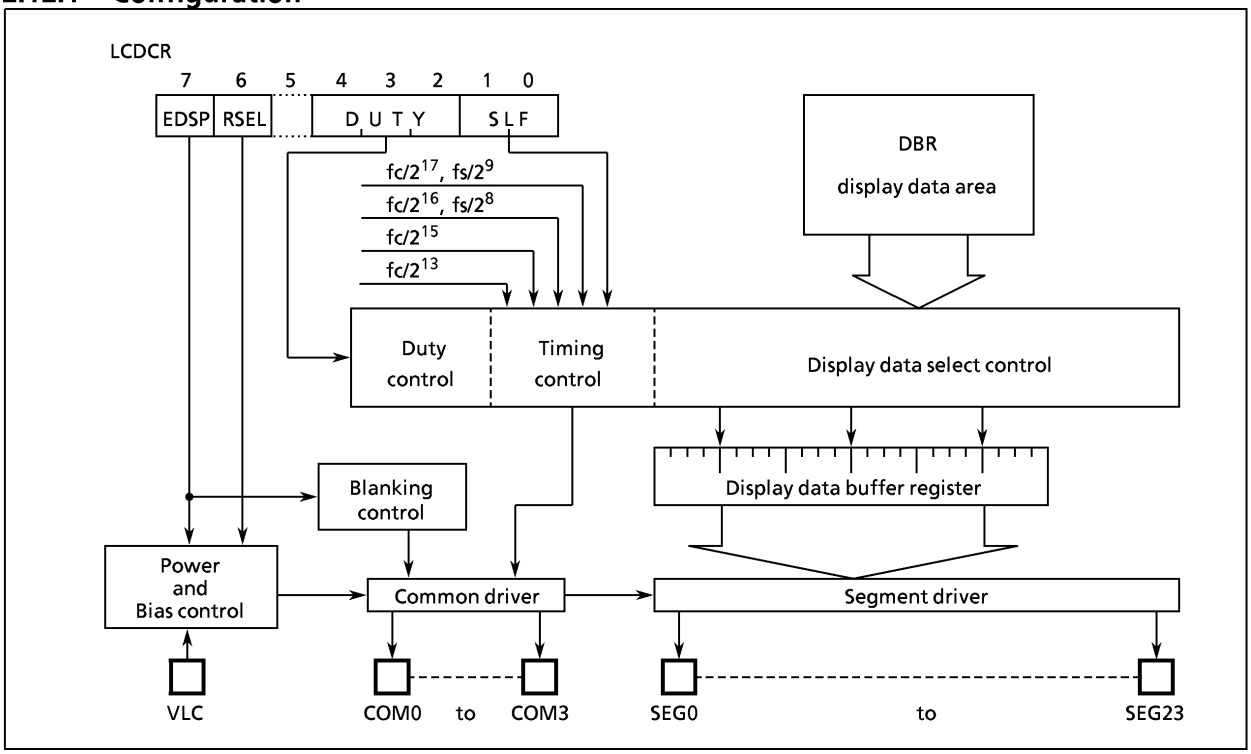


Figure 2-50. LCD Driver

2.12.2 Control

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

LCDCR (0028_H)

7	6	5	4	3	2	1	0
EDSP	RSEL		DUTY			SLF	

(Initial : 00*0 0000)

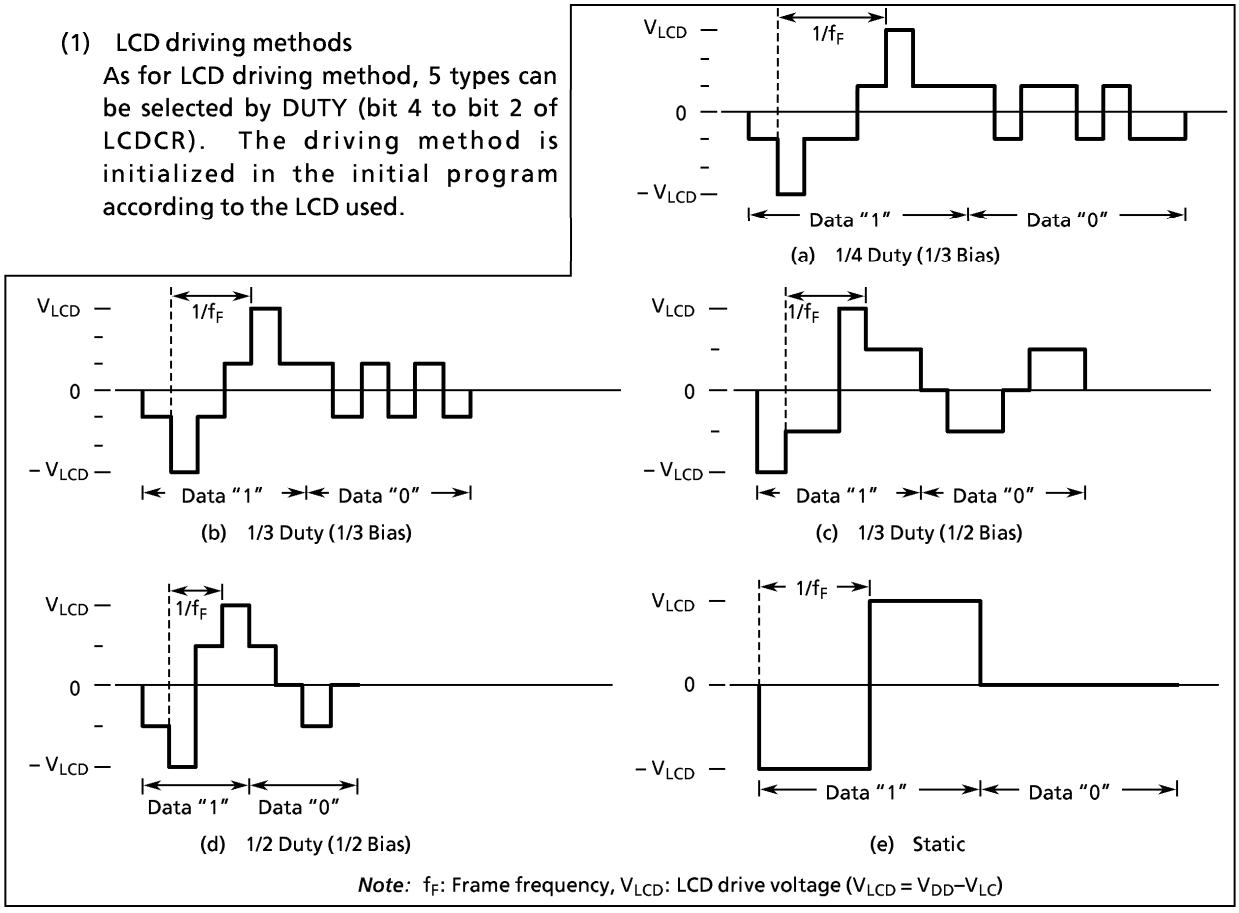
SLF	Selection of LCD driver frequency	00: $fc/2^{17}$ or $fs/2^9$ [Hz] 01: $fc/2^{16}$ or $fs/2^8$ 10: $fc/2^{15}$ 11: $fc/2^{13}$	Write only
DUTY	Selection of driving methods	000: 1/4 Duty (1/3 Bias) 001: 1/3 Duty (1/3 Bias) 010: 1/3 Duty (1/2 Bias) 011: 1/2 Duty (1/2 Bias) 100: Static 101: Reserved 110: Reserved 111: Reserved	
RSEL	Selection of LCD driver resistance	0: Divide resistance 0 (20/200 kΩ) 1: Divide resistance 1 (7/70 kΩ)	
EDSP	LCD Display Control	0: Blanking 1: Enables LCD display (Blanking is released)	

Note: fc: High-frequency clock, fs: Low-frequency clock

Figure 2-51. LCD Driver Control Register

(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.



Note: f_F: Frame frequency, V_{LCD}: LCD drive voltage ($V_{LCD} = V_{DD}-V_{LC}$)

Figure 2-52. 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 2-12. 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.

Table 2-12. Setting of LCD Frame Frequency

a. At the single clock mode

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

Note: fc: High-frequency clock [Hz]

b. At the dual clock mode

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

Note: fs: Low-frequency clock [Hz]

(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 $\pm V_{LCD}$. 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 V_{DD} 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.

(4) LCD output resistance

Selection of LCD output resistance can be selected by LCD control register (LCDCR).
The LCD output resistance is selected by RSEL (bit 6 of LCDCR) according to LCD used.
The LCD output resistance is selected 20/200kΩ (RSEL) during initial.

RSEL	Output low resistance		Output high resistance	
	Segment R_{OS1}	Common R_{OC1}	Segment R_{OS2}	Common R_{OC2}
0	20 kΩ		200 kΩ	
1	7 kΩ		70 kΩ	

Note: The output resistance shows Typ. values (Topr = 25°C, VDD = 5V)

Figure 2-53. Selection of LCD Output Resistance

2.12.3 LCD Display Operation

(1) Display data setting

Display data is stored to the display data area (assigned to address 0F80 to $0F8B_H$) 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 2-54 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 2-13)

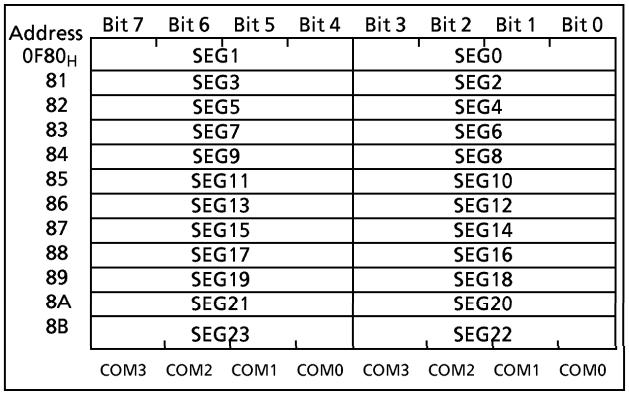


Figure 2-54. LCD Display Data Area (DBR)

Table 2-13. 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 $V_{LCD}/2$.

2.12.4 Control Method of LCD Driver

(1) Initial setting

Figure 2-55 shows the flowchart of initialization.

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

```
LD      (LCDCR),00000001B    ;  Sets LCD driving method and
                                frame frequency.
LD      (P7CR),0FFH          ;  Sets P7 port as segment
                                output .
  :       :                  ;  Sets the initial value of
                                display data.
LD      (LCDCR),10000001B    ;  Display enable
```

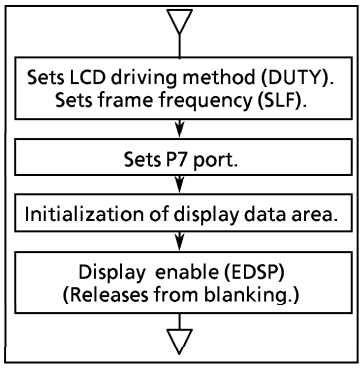


Figure 2-55. Initial Setting of LCD Driver

(2) Store of display data

Generally, display data are prepared as fixed data in program memory 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 2-56), display data become as shown in Table 2-14.

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

Note: DB is a byte data difinition instruction.

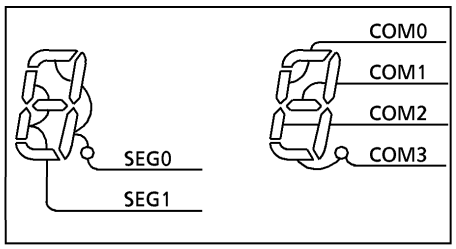
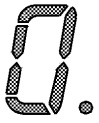
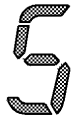
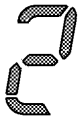
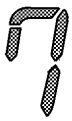
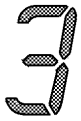
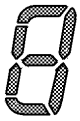
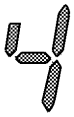
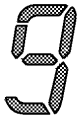


Figure 2-56. Example of COM, SEG Pin Connection (1/4 Duty)

Table 2-14. 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		00010111
3		10100111	8		11110111
4		00110110	9		10110111

Example 2: Table 2-15 shows an example of display data which are displayed using 1/2 duty LCD in the same way as Table 2-14. The connection between pins COM and SEG are the same as shown in Figure 2-57.

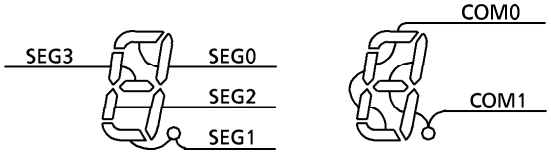


Figure 2-57. Example of COM, SEG Pin Connection

Table 2-15. 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

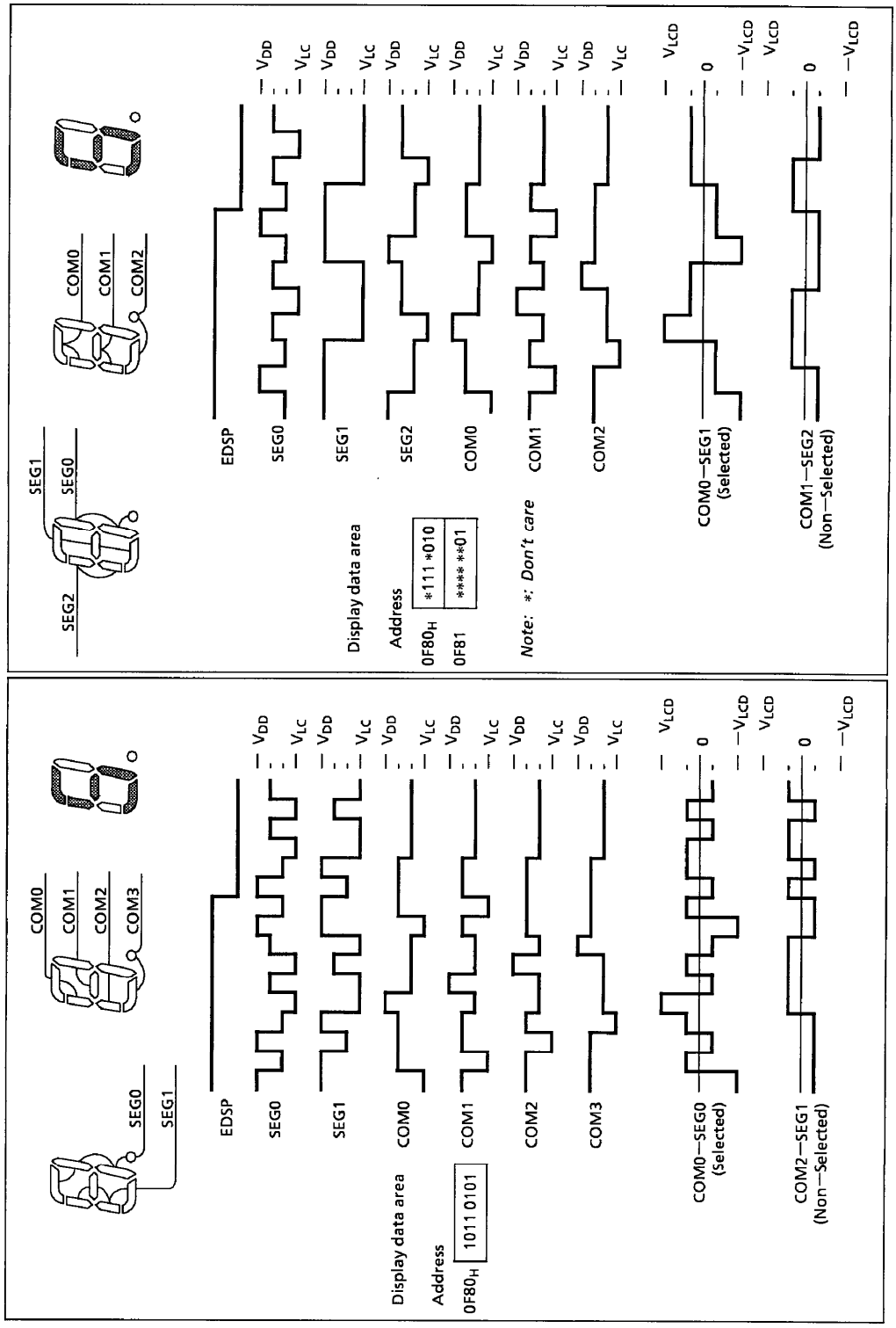


Figure 2-58. 1/4 Duty (1/3 Bias) Drive

Figure 2-59. 1/3 Duty (1/3 Bias) Drive

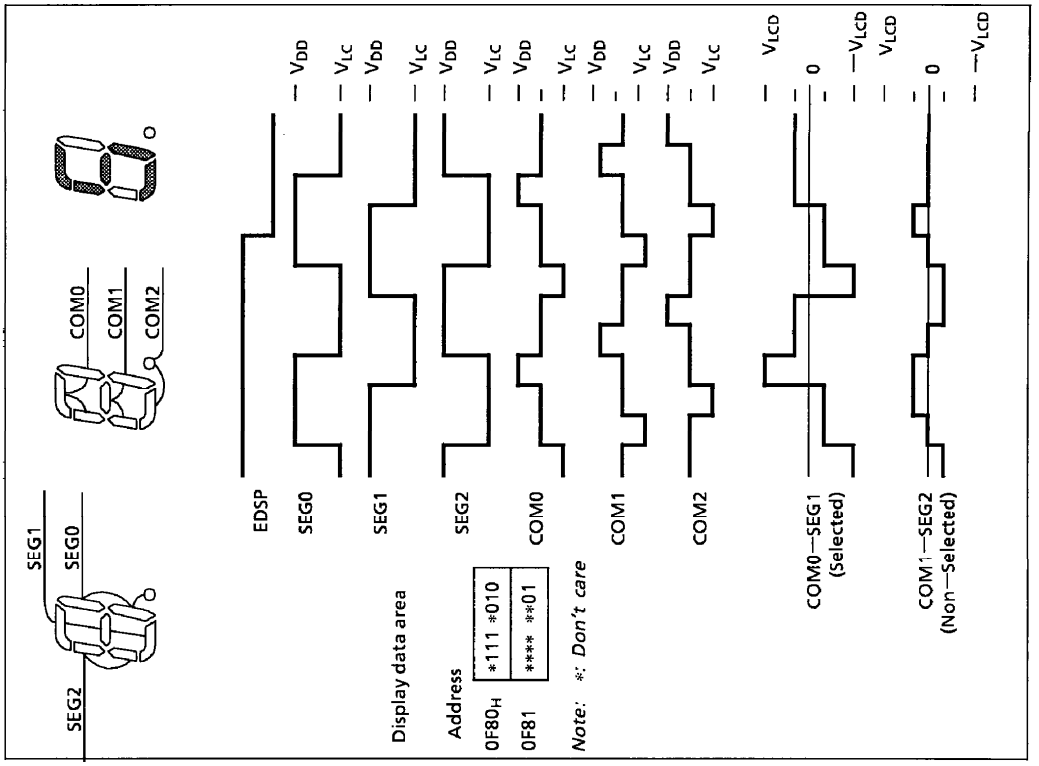


Figure 2-60. 1/3 Duty (1/2 Bias) Drive

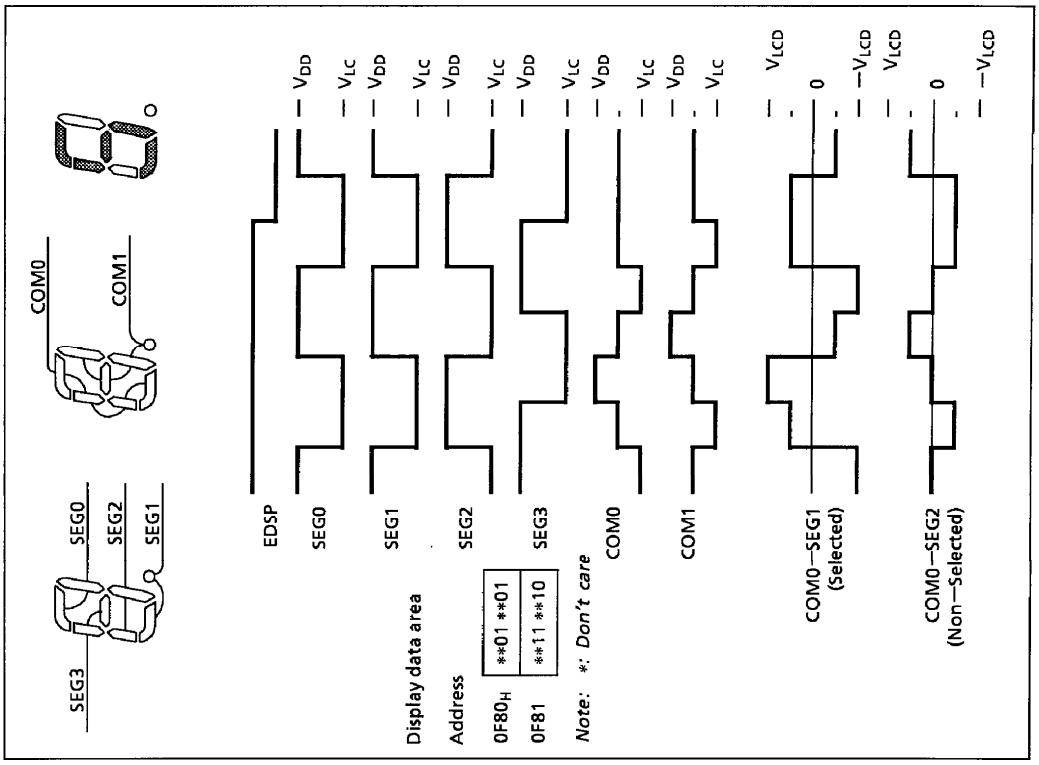


Figure 2-61. 1/2 Duty (1/2 Bias) Drive

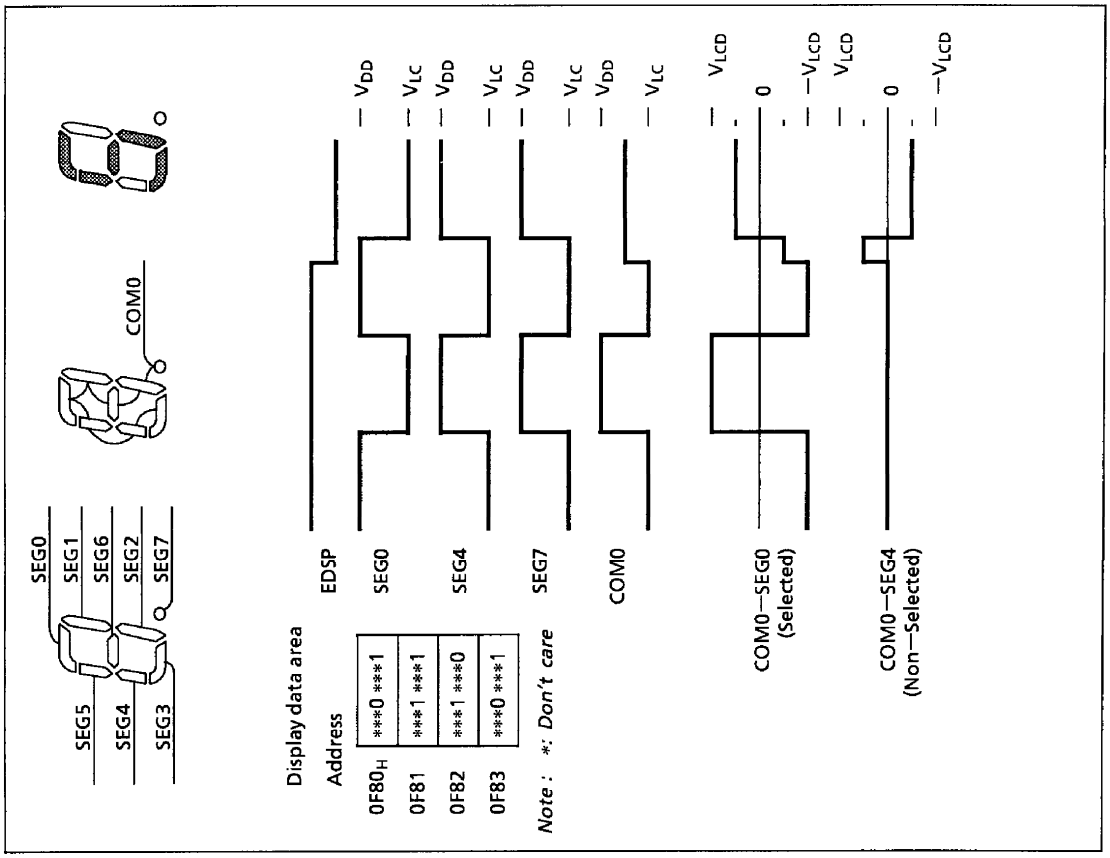


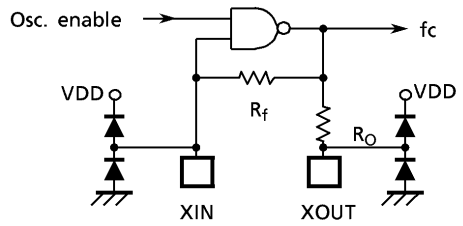
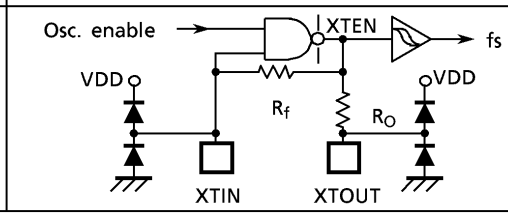
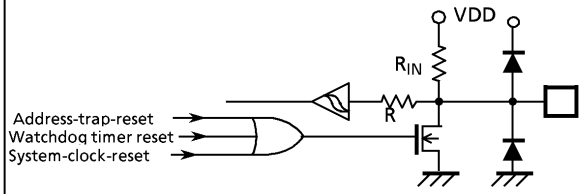
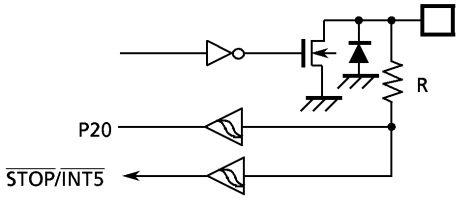
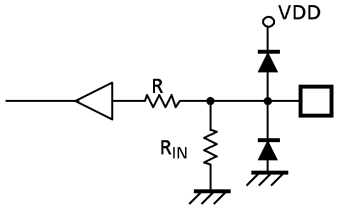
Figure 2-62. Static Drive

Input/Output Circuitry

(1) Control pins

The input/output circuitries of the TMP87CH29/K29/M29 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.

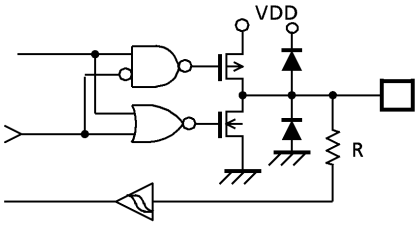
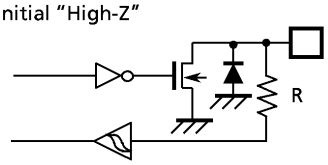
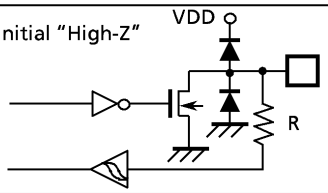
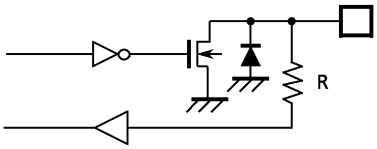
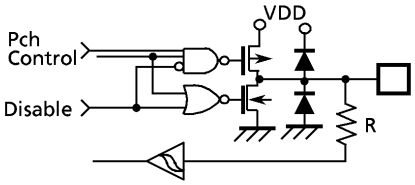
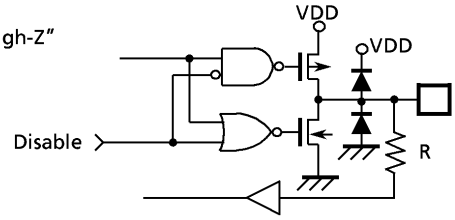
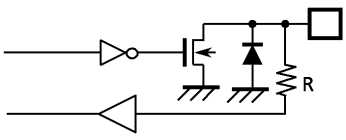
Control Pin	I/O	Input/Output Circuitry and CODE	Remarks
XIN XOUT	I/O		Resonator connecting pins (high-frequency) R_f = 1.2 MΩ (typ.) R_O = 1.5 kΩ (typ.)
XTIN (P21) XTOUT (P22)	I/O	NM1: Refer to port P2 NM2: 	Resonator connecting pins (low-frequency) R_f = 6 MΩ (typ.) R_O = 220 kΩ (typ.) In only dual-clock mode
$\overline{\text{RESET}}$	I/O		Sink open drain output Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 1 kΩ (typ.)
$\overline{\text{STOP/INT5}}$ (P20)	Input		Hysteresis input R = 1 kΩ (typ.)
TEST	Input		Pull-down resistor R_{IN} = 70 kΩ (typ.) R = 1 kΩ (typ.)

Note 1: The TEST pin of the TMP87PM29 does not have a pull-down resistor. Be sure to fix the TEST pin to low in MCU mode.

Note 2: The TMP87PM29 is placed in the single-clock mode during reset.

(2) Input/Output ports

The input/output circuitries of the TMP87CH29/K29/M29 input/output ports are shown below.
A mask option code is only "A".

Port	I/O	Input/Output Circuitry (CODE A)	Remarks
P1	I/O	Initial "High-Z" Disable 	Tri-state I/O Hysteresis input R = 1 kΩ (typ.)
P2	I/O	P20: Initial "High-Z"  P21, P22: initial "High-Z" 	Sink open drain output Hysterisis input R = 1 kΩ
P3	I/O	Initial "High-Z" 	Sink open drain output High current output (P3) R = 1 kΩ (typ.)
P4	I/O	Initial "High-Z" Pch Control Disable 	Sink open drain or push-pull output Programable port option Hysterisis input R = 1 kΩ (typ.)
P5	I/O	Initial "High-Z" Disable 	Tri-state I/O R = 1 kΩ (typ.)
P6 P7	I/O	Initial "High-Z" 	Sink open drain output R = 1 kΩ

Electrical Characteristics

(1) TMP87CH29/K29/M29

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

Parameter	Symbol	Conditions	Ratings	Unit
Supply Voltage	V_{DD}		– 0.3 to 6.5	V
Input Voltage	V_{IN}		– 0.3 to V_{DD} + 0.3	V
Output Voltage	V_{OUT1}	P21, P22, $\overline{RESET}$, Tri-state port, and Push-pull port	– 0.3 to V_{DD} + 0.3	V
	V_{OUT2}	P20, Port P3 and Segment port	– 0.3 to 5.5	
Output Current (Per 1 pin)	I_{OUT1}	Ports P1, P2, P4, P5, P6, P7	3.2	mA
	I_{OUT2}	Port P3	30	
Output Current (Total)	ΣI_{OUT1}	Ports P1, P2, P4, P5, P6, P7	120	mA
	ΣI_{OUT2}	Port P3	60	
Power Dissipation [Topr = 70°C]	PD	TMP87CH29N/CK29N/CM29N	600	mW
		TMP87CH29U/CK29U/CM29U	350	
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} = 0V, 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			
			fc = 4.2 MHz	NORMAL1, 2 mode	2.7		
				IDLE1, 2 mode			
			fs = 32.768 kHz	SLOW mode			
				SLEEP mode			
				STOP mode	2.0		
Input High Voltage	V_{IH1}	Except hysteresis input	$V_{DD} \geqq 4.5$ V		$V_{DD} \times 0.70$	V_{DD}	V
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 4.5$ V		$V_{DD} \times 0.90$		
Input Low Voltage	V_{IL1}	Except hysteresis input	$V_{DD} \geqq 4.5$ V		0	$V_{DD} \times 0.30$	V
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 4.5$ V			$V_{DD} \times 0.10$	
Clock Frequency	fc	XIN, XOUT	V_{DD} = 4.5 to 5.5 V		0.4	8.0	MHz
			V_{DD} = 2.7 to 5.5 V			4.2	
	fs	XTIN, XTOUT			30.0	34.0	kHz

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

Note 2: Clock frequency fc; The supply voltage range of the conditions shows the value in NORMAL 1, 2 modes and IDLE 1, 2 modes.

DCCharacteristics

(V_{SS} = 0 V, Topr = – 30 to 70°C)

Parameter	Symbol	Pins	Conditions		Min	Typ.	Max	Unit
Hysteresis Voltage	V_{HS}	Hysteresis input			—	0.9	—	V
Input Current	I_{IN1}	TEST	V_{DD} = 5.5 V V_{IN} = 5.5 V/0 V		—	—	± 2	μA
	I_{IN2}	Sink open drain port and tri-state port						
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$						
Input Low Current	I_{IL}	Push-pull port	V_{DD} = 5.5 V, V_{IN} = 0.4 V		—	—	– 2	mA
Input Resistance	R_{IN}	$\overline{RESET}$			100	220	450	kΩ
Output Leakage Current	I_{LO}	Sink open drain port and tri-state port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V		—	—	2	μA
Output High Voltage	V_{OH1}	Push-pull port	V_{DD} = 4.5 V, I_{OH} = – 200 μA		2.4	—	—	V
	Vo_{H2}	Tri- state port	V_{DD} = 4.5 V, I_{OH} = – 0.7 mA		4.1	—	—	
Output Low Voltage	V_{OL}	Except XOUT and port P3	V_{DD} = 4.5 V, I_{OL} = 1.6 mA		—	—	0.4	V
Output Low Current	I_{OL}	Only P30, P31, P32	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		—	10	16	mA
Supply Current in IDLE 1, 2 mode					—	4.5	6	mA
Supply Current in SLOW mode			V_{DD} = 3.0 V fs = 32.768 kHz V_{IN} = 2.8 V/0.2 V LCD driver is not enable		—	30	60	μA
Supply Current in SLEEP mode					—	15	30	μA
Supply Current in STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V		—	0.5	10	μA
Segment Output Low Resistance	R_{OS1}	SEG23 to SEG0 pins	V_{DD} = 5 V V_{DD} – V_{LC} = 3 V	RESL = 0 (Note 11)	—	20	—	kΩ
				RSEL = 1		7		
Common Output Low Resistance	R_{OC1}	COM3 to COM0 pins		RESL = 0		20		
				RSEL = 1		7		
Segment Output High Resistance	R_{OS2}	SEG23 to SEG0 pins		RESL = 0		200		
				RSEL = 1		70		
Common Output High Resistance	R_{OC2}	COM3 to COM0 pins		RESL = 0		200		
				RSEL = 1		70		
Segment /Common Output Voltage	$V_{O\ 2/3}$	SEG23 to SEG0 and COM3 to COM0 pins			3.8	4.0	4.2	V
	$V_{O\ 1/2}$				3.3	3.5	3.7	
	$V_{O\ 1/3}$				2.8	3.0	3.2	

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: IDD; Except for I_{REF}.
Note 4: Output resustance R_{OS} and R_{OC} indicate "on" when switching levels.
Note 5: $V_{O2/3}$ indicates an output current at the 2/3 level when operating in the 1/4 or 1/3 duty mode.
Note 6: $V_{O1/2}$ indicates an output current at the 1/2 level when operating in the 1/2 duty or static mode.
Note 7: $V_{O1/3}$ indicates an output current at the 1/3 level when operating in the 1/4 or 1/3 duty mode.
Note 8: When you use a liquid crystal display (LCD), it is necessary to give careful consideration to the value of the output resistor $R_{OS\ 1/2}$, $R_{OC\ 1/2}$.
Note 9: R_{OS1}, R_{OC1}: On time of the lower output resistor is 2^7/fc, 1/(2·fs) [s].
Note 10: R_{OS2}, R_{OC2}: On time of the higher output resistor is 1/(n·f_F). (1/n duty, f_F: frame frequency)
Note 11: RSEL ; Bit 6 in LCDCR

AD Conversion Characteristics

(V_{SS} = 0 V, V_{DD} = 2.7 to 5.5 V, Topr = – 30 to 70°C)

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Analog Reference Voltage	V_{AREF}	$V_{AREF} - V_{ASS} \geqq 2.5$ V	2.7	—	V_{DD}	V
	V_{ASS}		V_{SS}	—	1.5	
Analog Input Voltage	V_{AIN}		V_{ASS}	—	V_{AREF}	V
Analog Supply Current	I_{REF}	V_{AREF} = 5.5 V, V_{ASS} = 0.0 V	—	0.5	1.0	mA
Nonlinearity Error		V_{DD} = 5.0 V, V_{SS} = 0.0 V V_{AREF} = 5.000 V V_{ASS} = 0.000 V or V_{DD} = 2.7 V, V_{SS} = 0.0 V V_{AREF} = 2.700 V V_{ASS} = 0.000 V	—	—	± 1	LSB
Zero Point Error			—	—	± 1	
Full Scale Error			—	—	± 1	
			—	—	± 2	

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)

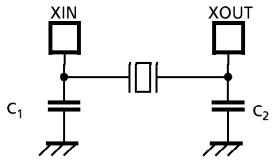
Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Machine Cycle Time	t_{cy}	In NORMAL 1, 2 mode	0.5	–	10	μs
		In IDLE 1, 2 mode				
		In SLOW mode	117.6	–	133.3	
		In SLEEP mode				
High Level Clock Pulse Width	t_{WCH}	For external clock operation (XIN input), fc = 8 MHz	50	–	–	ns
Low Level Clock Pulse Width	t_{WCL}					
High Level Clock Pulse Width	t_{WSH}	For external clock operation (XTIN input), fs = 32.768 kHz	14.7	–	–	μs
Low Level Clock Pulse Width	t_{WSL}					

(V_{SS} = 0 V, Topr = – 30 to 70°C)

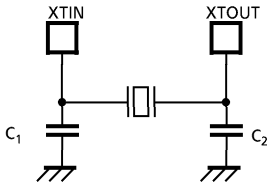
Parameter	Symbol	Conditions		Min	Typ.	Max	Unit
TC1 Input (ECIN input)	t_{TC1}	Frequency medsurement mode V_{DD} = 4.5 to 5.5 V	Single edge count	—	—	8	MHz
			Both edge count	—	—	4	
		Frequency medsurement mode V_{DD} = 2.7 to 5.5 V	Single edge count	—	—	4.2	
			Both edge count	—	—	3	

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

Parameter	Oscillator	Frequency	Recommended Oscillator		Recommended Condition	
					C_1	C_2
High-frequency	Ceramic Resonator	8 MHz	KYOCERA	KBR8.0M	30pF	30pF
		4 MHz	KYOCERA	KBR4.0MS		
			MURATA	CSA4.00MG		
	Crystal Oscillator	8 MHz	TOYOCOM	210B 8.0000	20pF	20pF
		4 MHz	TOYOCOM	204B 4.0000		
Low-frequency	Crystal Oscillator	32.768 kHz	NDK	MX-38T	15pF	15pF



(1) High-frequency



(2) Low-frequency

Note 1: When it is used in high electrical field, an electrical shield of the package is recommended to retain normal operations.

Note 2: The product numbers and specifications of the resonators by 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